Generative AI with Amazon Bedrock

Workshop

Abhi Sodhani

Sr. AI/ML Specialist Solutions Architect

Agenda

- What is Generative AI
- Amazon Bedrock
- How to access Bedrock
- Architecture patterns
- Handson lab

What is Generative AI?

What is generative AI?

AI that can produce original content close enough to human generated content for real-world tasks

Powered by foundation models pre-trained on large sets of data with several hundred billion parameters

Tasks can be customized for specific domains with minimal fine-tuning

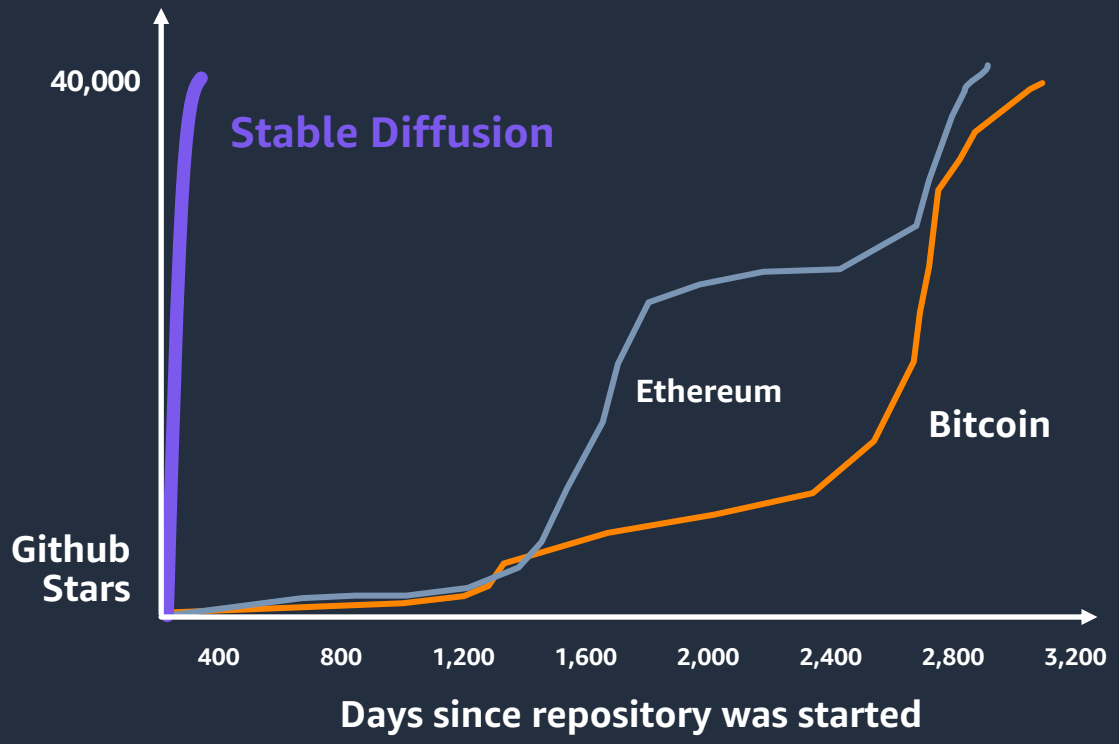
Applicable to many use cases like text summarization, question answering, digital art creation, code generation, etc.

Reduces time and cost to develop ML models and innovate faster

Generative AI is the fastest growing trend in AI

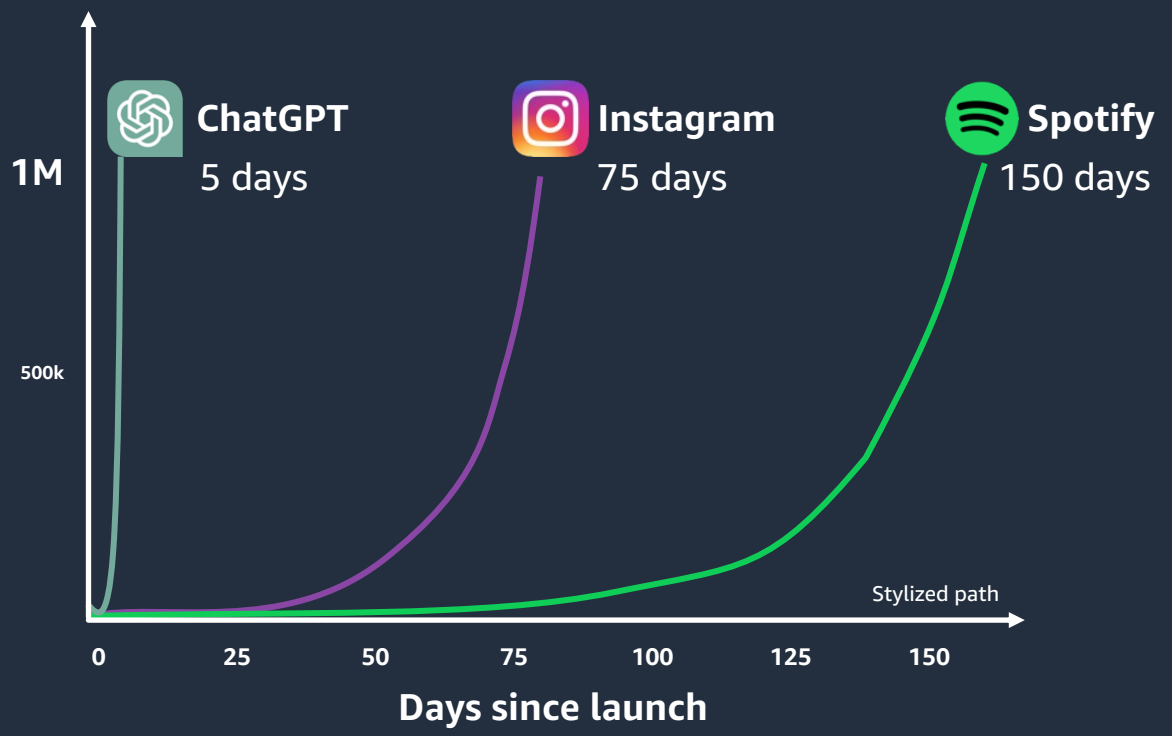
Developer adoption

Stable Diffusion accumulated 40k stars on GitHub in its first 90 days



Consumer adoption

ChatGPT reached the 1 million users mark in just 5 days



Generative AI is emerging across a range of domains ...

Enhance customer experience

CHATBOTS

VIRTUAL ASSISTANTS

AI-POWERED CONTACT CENTER

PERSONALIZATION

Boost employee productivity

CONVERSATIONAL SEARCH

SUMMARIZATION

CODE GENERATION

DATA TO INSIGHTS

Creativity and content creation

WRITING

MEDIA

DESIGN

MODELING

Improve business operations

DOCUMENT PROCESSING

PROCESS OPTIMIZATION

CYBERSECURITY

DATA AUGMENTATION

Building generative AI applications is challenging

Accessing multiple FMs and newer versions

Customizing FMs is not easy

Data privacy and security

Getting FMs to execute tasks

Connecting to data sources

Difficult to manage infrastructure

Amazon Bedrock

THE EASIEST WAY TO BUILD AND SCALE GENERATIVE AI APPLICATION WITH FMS

Access a range of leading FMs via a single API

Privately customize FMs with your own data

Enable data security and compliance

Build agents that execute complex business tasks by dynamically invoking APIs

Extend the power of FMs with your data using retrieval augmented generation (RAG)

Get the best price performance without managing infrastructure

How Amazon Bedrock works

Amazon Bedrock

Build generative AI applications using FMs through a serverless API service

Choose a FM

Use the playground to experiment with FMs and select the one that suits your needs

Use as is or customize

Fine-tune FMs as needed; Bedrock will automatically deploy the FM for inference

Send prompt

Use Bedrock API to send your prompts to the model

Receive response

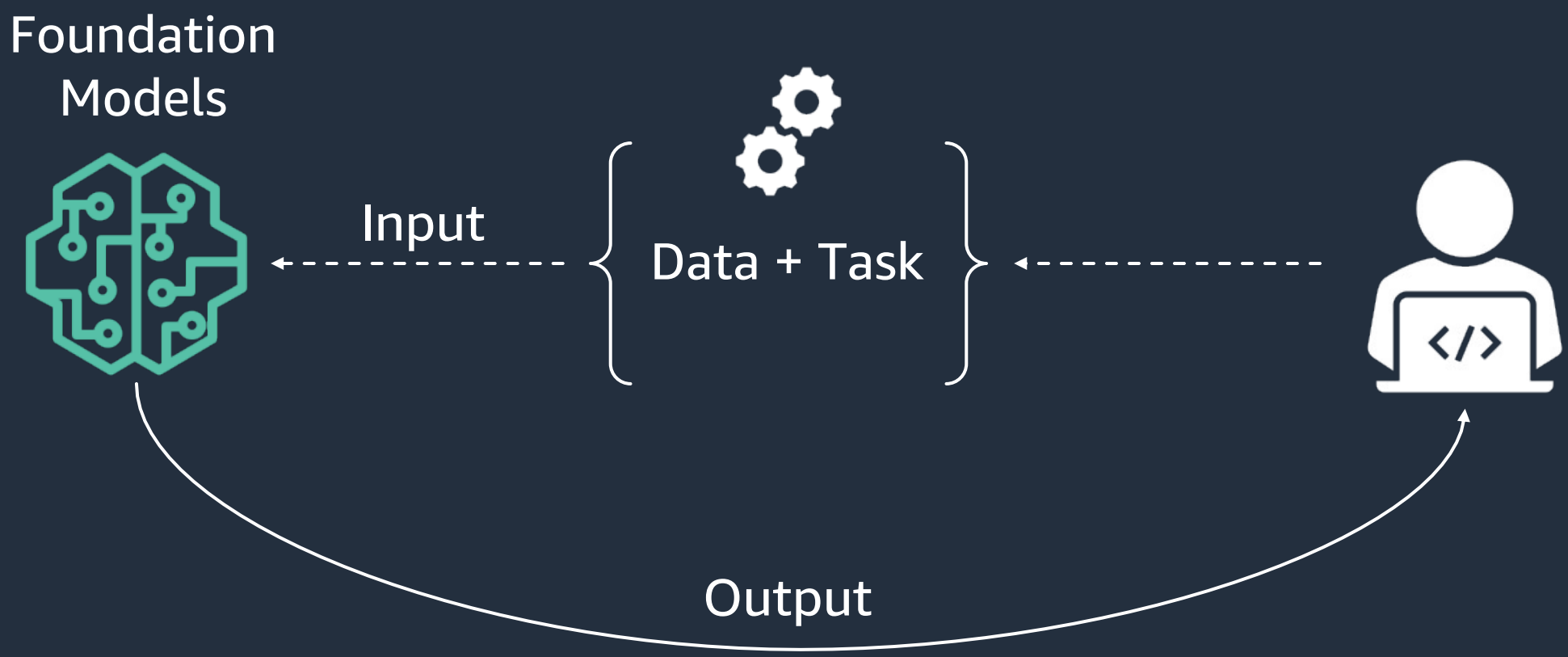
Receive model response in your application

Benefits

- Choice of leading foundation models
- Easy model customization
- Fully managed agents to execute tasks
- Native support for RAG
- Security and compliance

Prompt engineering

Prompt engineering, new way of using ML!



Elements of a Prompt

Instruction

Task description or instruction on how the model should perform

Context

Additional/external information to steer the model performance

Input Data

The input/question that the model needs to provide output for

Output Indicator

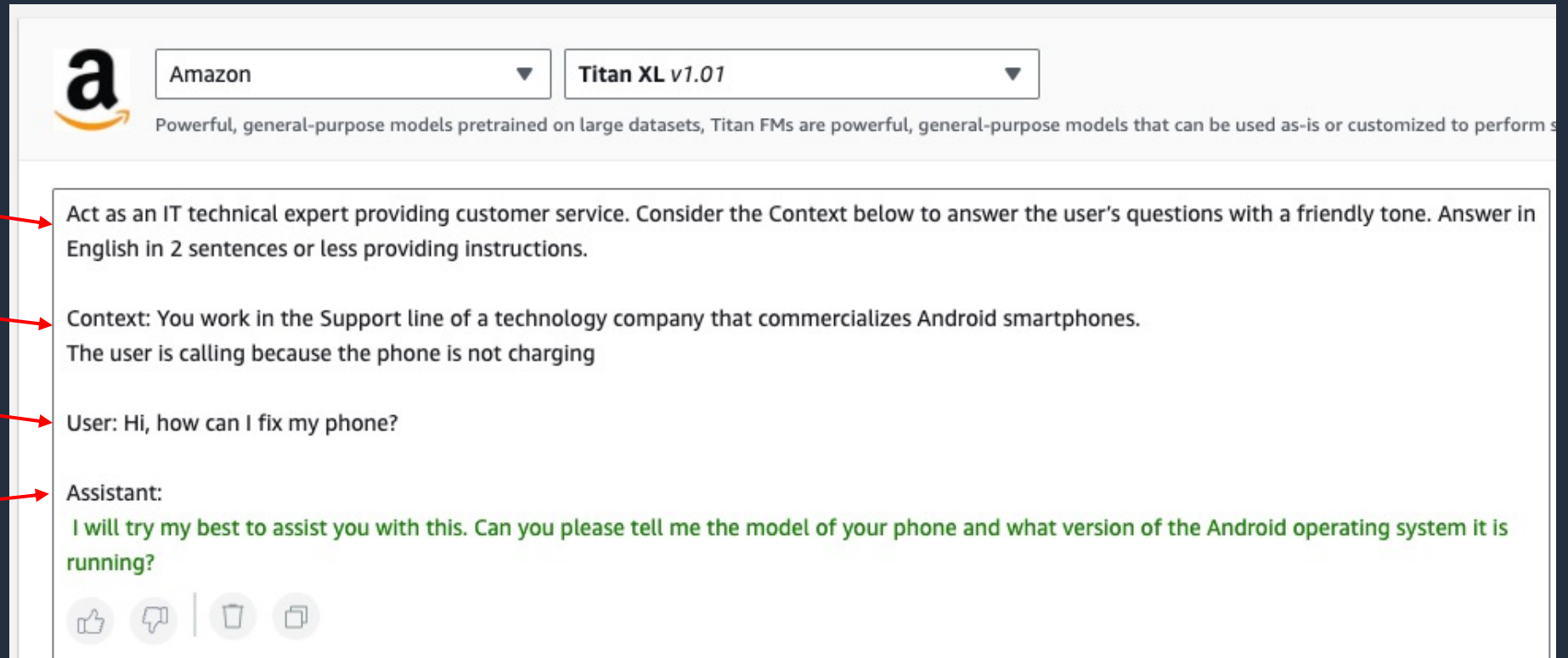
The indicator/format the model needs to provide output with

Instructions

Context

User Input

Output Indicator



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 adjust inference configuration parameters and then re-run your prompt.

Amazon Bedrock API

Single API to build with generative AI

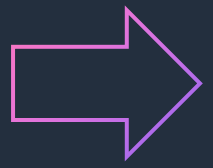
Single API to build with generative AI

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

Bedrock core API: InvokeModel

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

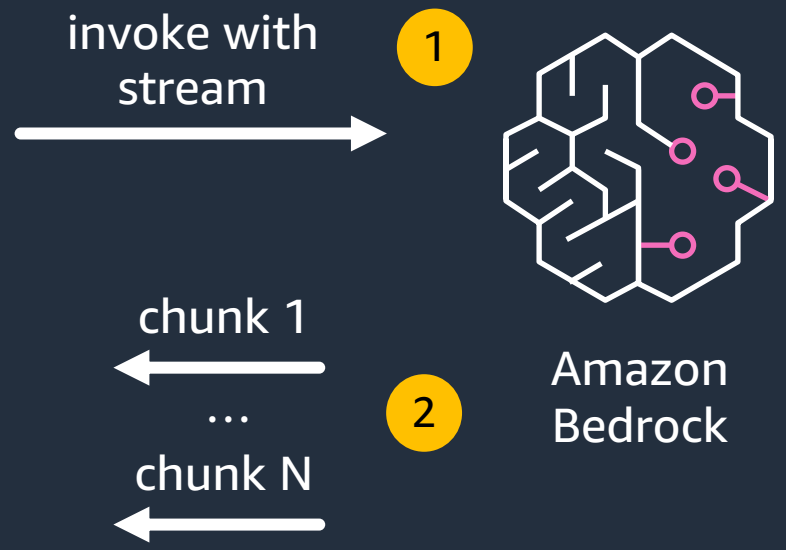


Access foundation models

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

Bedrock core API: Streaming responses

```
response = bedrock.invoke_model_with_response_stream(
      modelId = model_id, body = body)
stream = response.get('body')
if stream:
     for event in stream:
          chunk = event.get('chunk’)
          if chunk:
               print(json.loads(chunk.get('bytes').decode()))
```



- Users can start reading the response as soon as the first chunk is available
- Initially supported for Amazon Titan models; Claude and J2 models coming soon

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)

Integrated with LangChain

```
pip install langchain
```

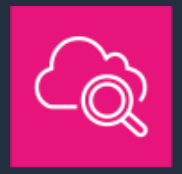
```
from langchain import Bedrock
from langchain.embeddings
import BedrockEmbeddings

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

CloudWatch metrics



Amazon CloudWatch

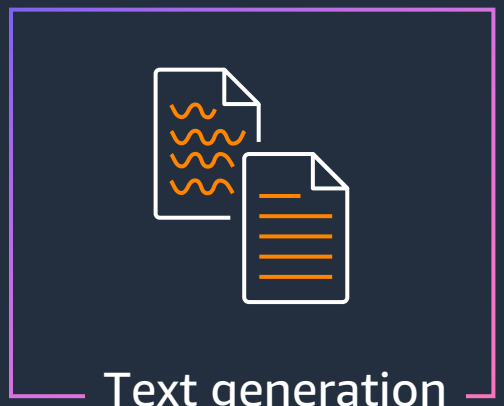
CloudWatch metrics now supported:

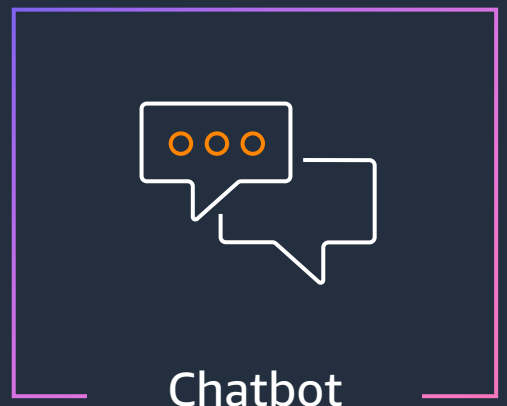
- Number of model invocations
- Latency of invocation
- Error metrics include number of invocations with:
  - Client side errors
  - Server side errors
  - Throttling

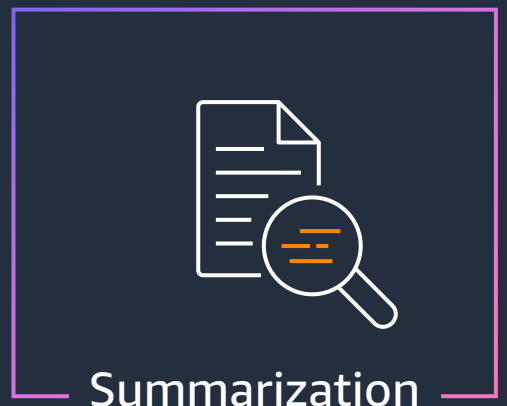
“AWS/Bedrock” namespace, and each metric is per model (“ModelId” dimension)

Architecture patterns

Architecture patterns in:







Text Generation

WITH SIMPLE PROMPT

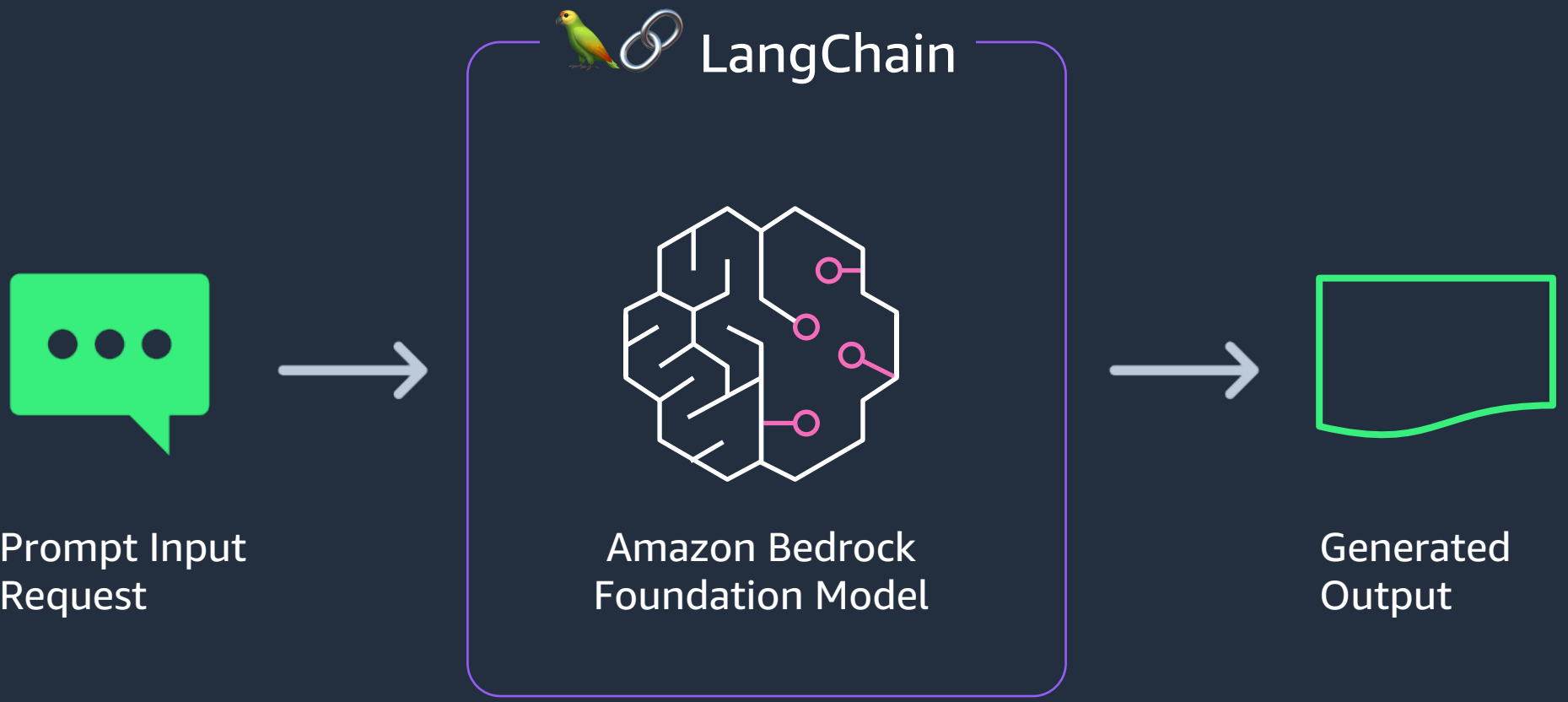
Prompt Input
Request

Amazon Bedrock
Foundation Model

Generated
Output

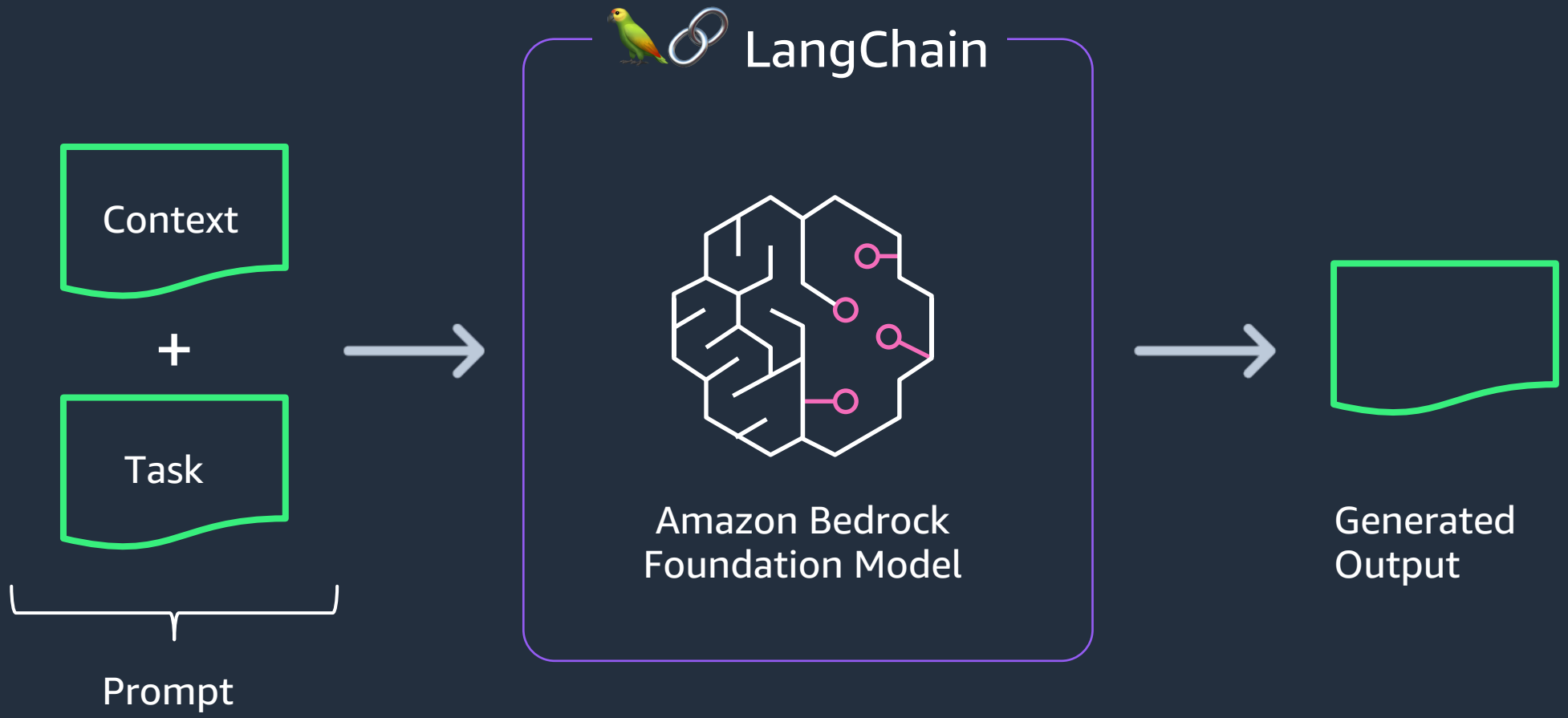
Text Generation

WITH LANGCHAIN



Text Generation

WITH CONTEXT AND LANGCHAIN



Text Summarization

WITH SMALL FILES

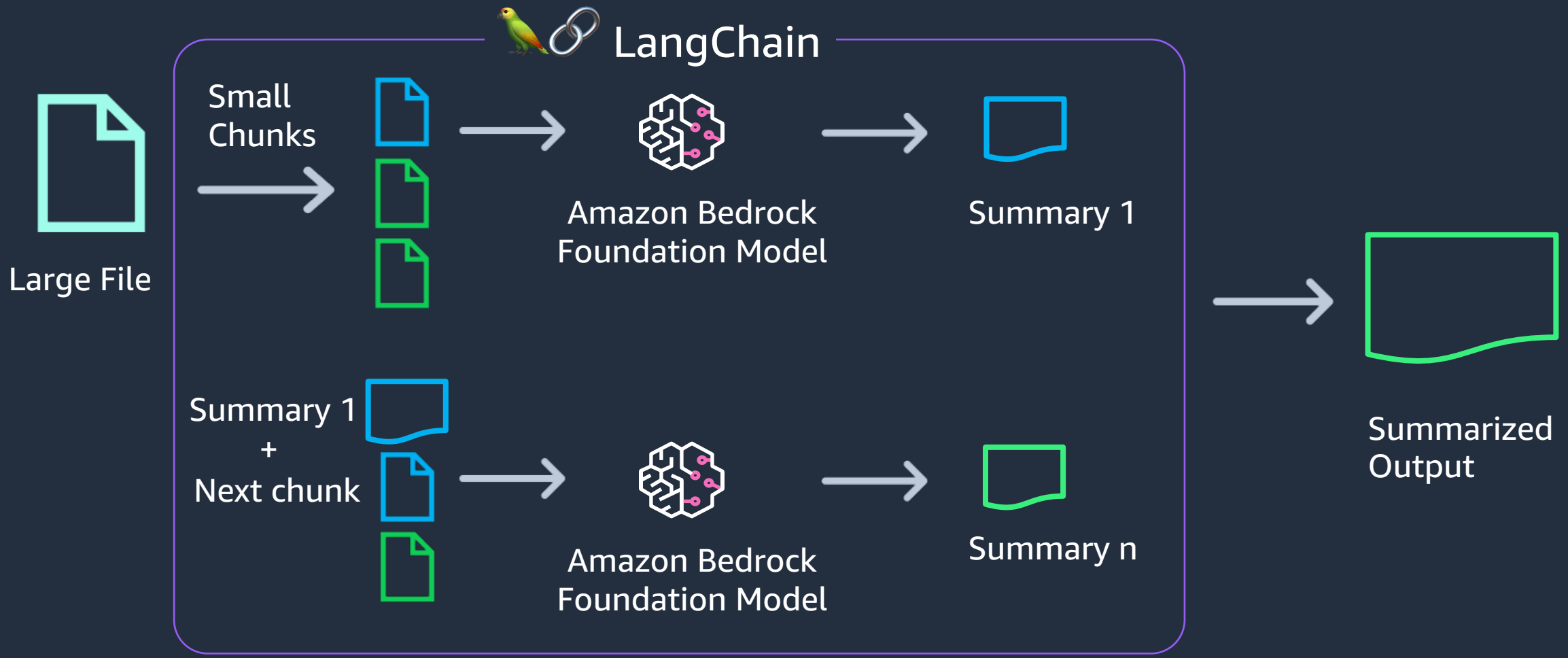
Small file

Amazon Bedrock
Foundation Model

Summarized
Output

Text Summarization

WITH LARGE FILES AND LANGCHAIN



Question Answering

WITH SIMPLE PROMPT

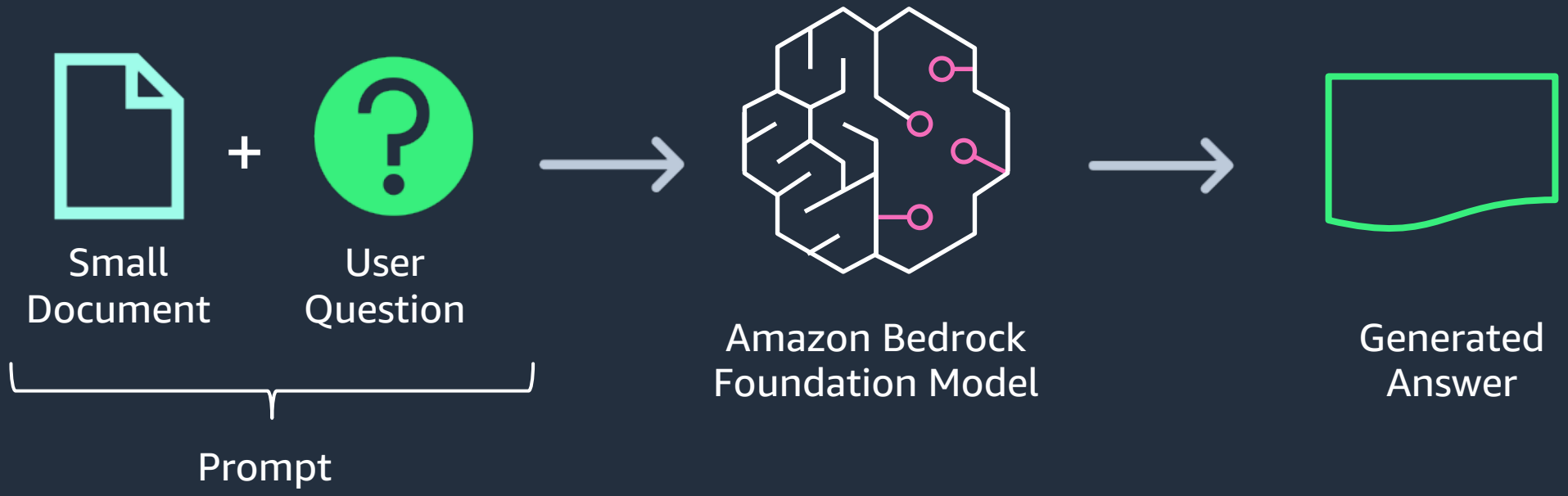
User Question

Amazon Bedrock Foundation Model

Generated Answer

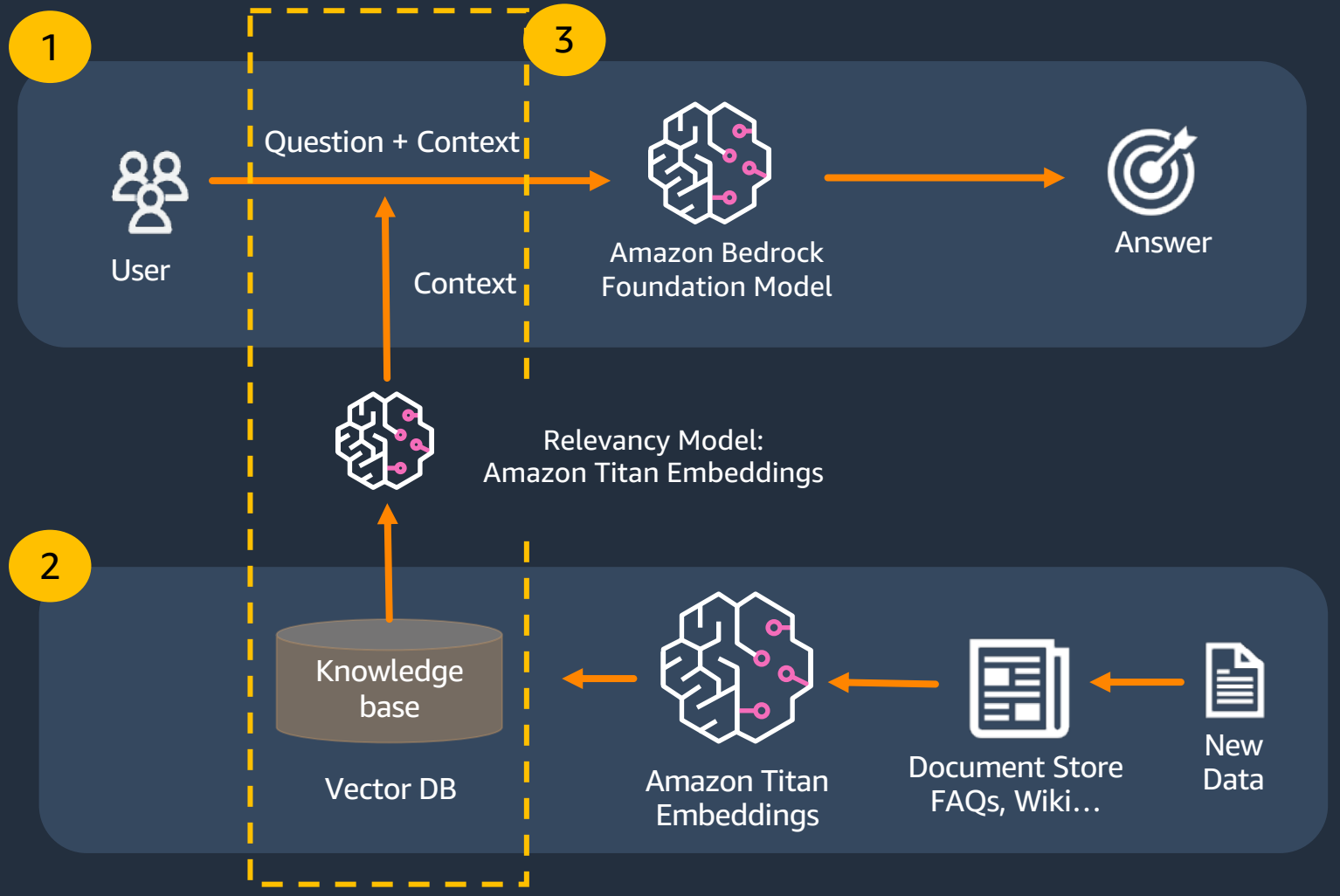
Question Answering

WITH CONTEXT



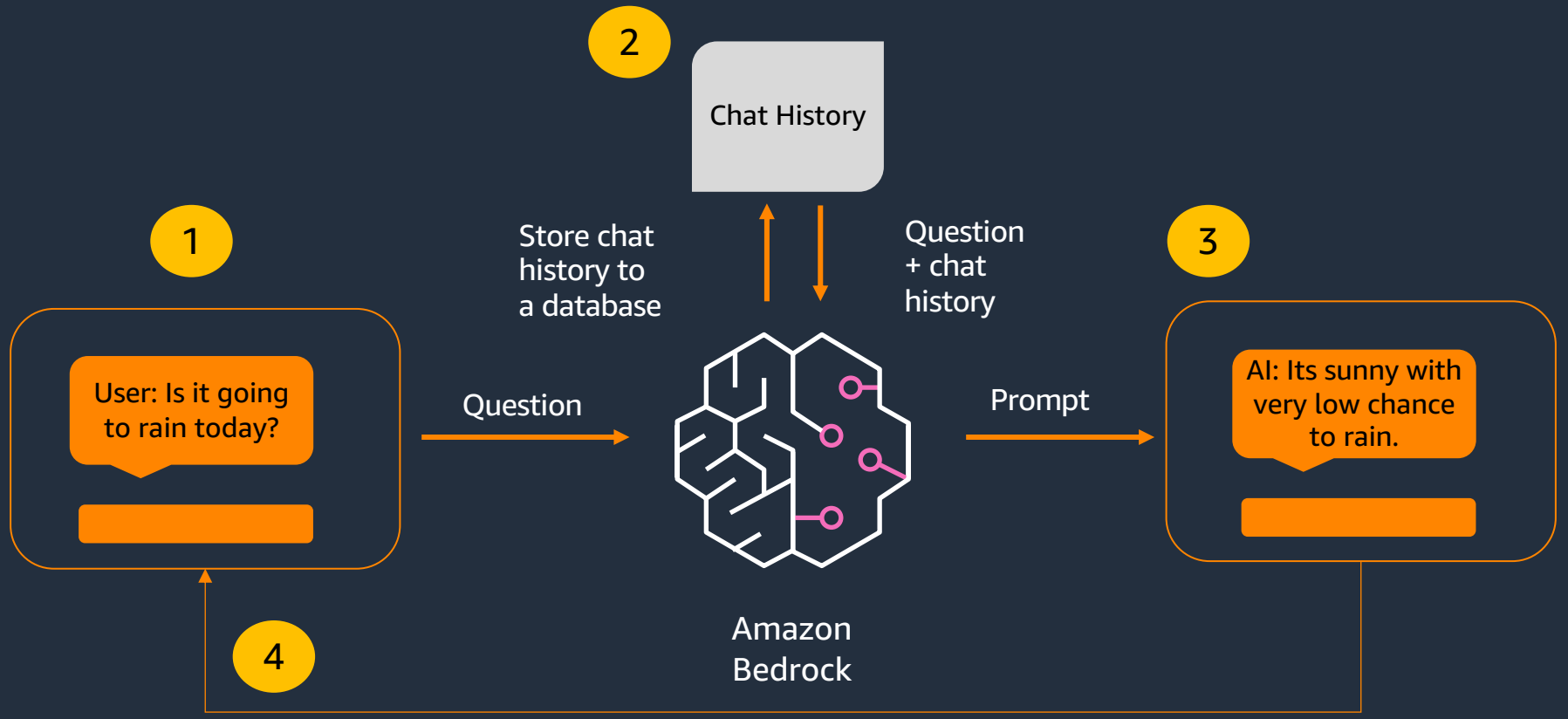
Question Answering

WITH RETRIEVAL-AUGMENTED GENERATION



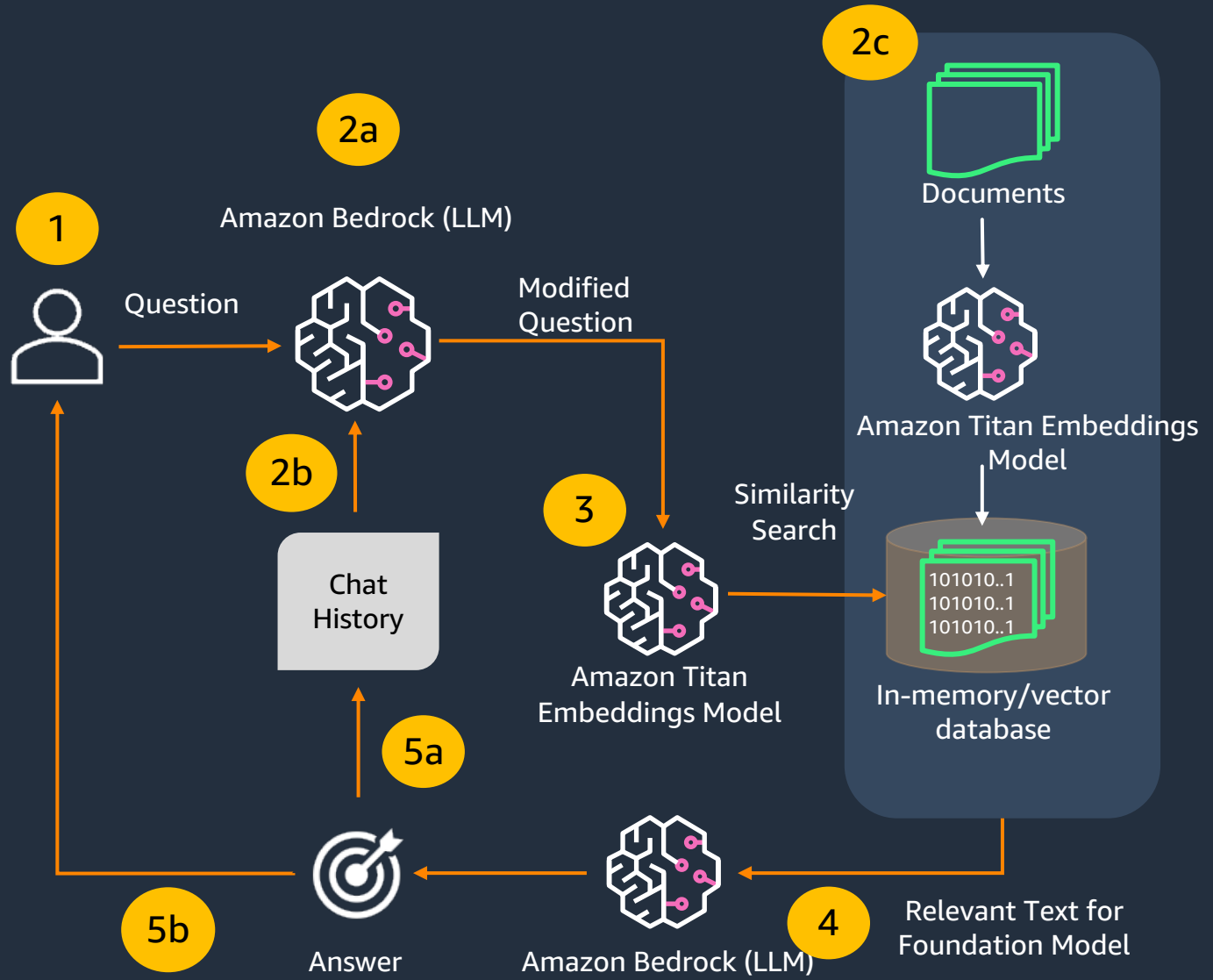
Chatbot

BASIC



Chatbot

WITH CONTEXT



Text to Image

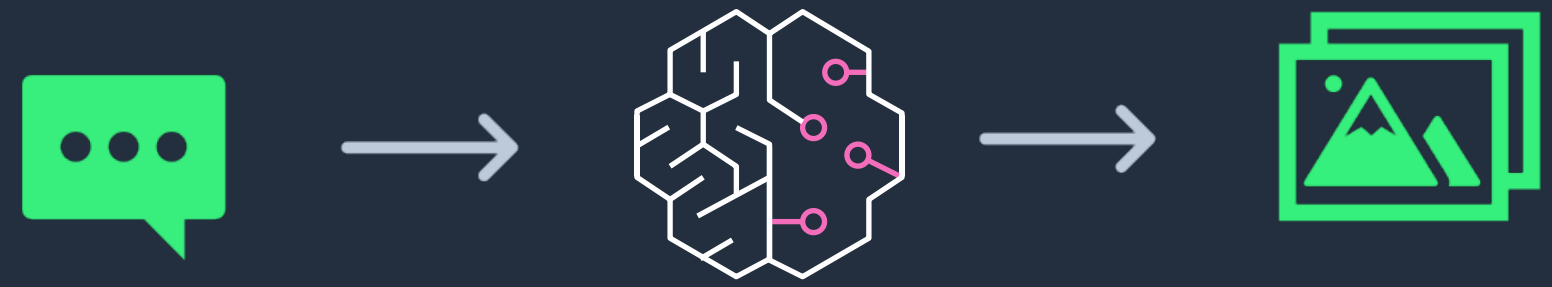
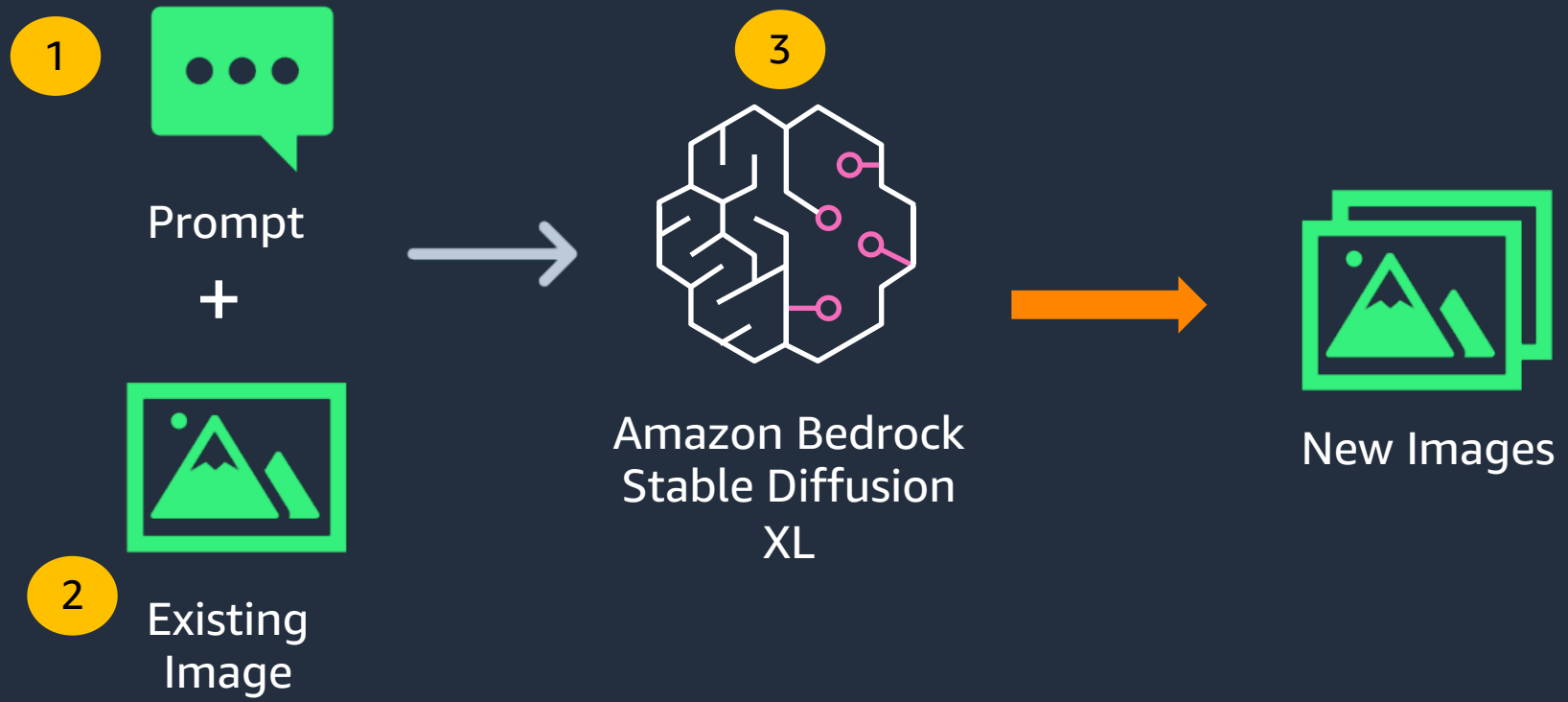
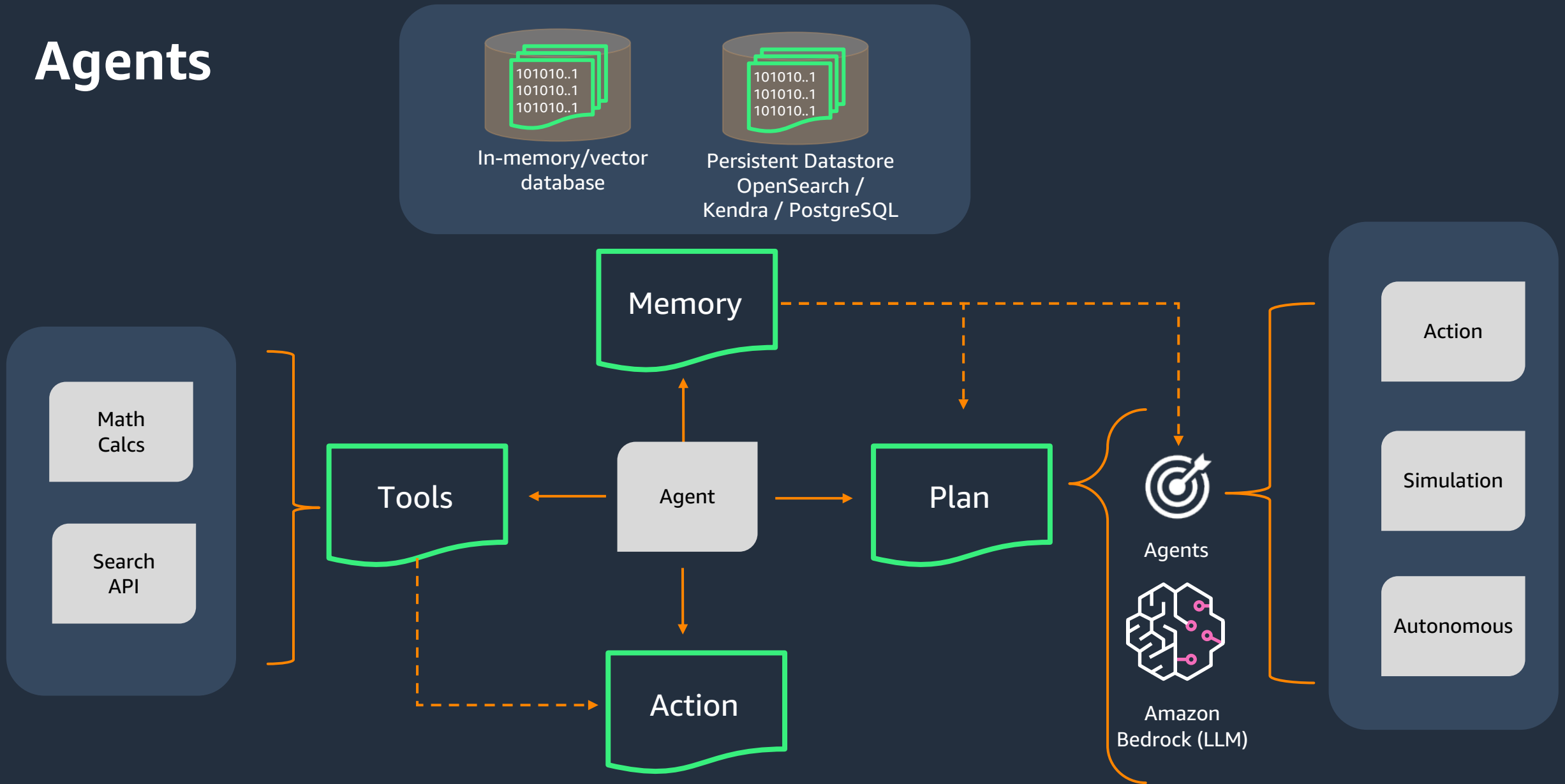


Image to Image (In-painting)



Agents


101010..1
101010..1
101010..1
In-memory/vector database
101010..1
101010..1
101010..1
Persistent Datastore OpenSearch / Kendra / PostgreSQL
Memory
Math Calcs
Search API
Tools
Agent
Plan
Action
Agents
Amazon Bedrock (LLM)
Action
Simulation
Autonomous

Security

Data privacy

You are always in control of your data

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

Data security

You are always in control of your data

- Support for **AWS PrivateLink** so customers can establish private connectivity between virtual private clouds (VPCs) and the Bedrock service using VPC Endpoints
- Integration with AWS Identity and Access Management Service (IAM) to manage inference access, deny access for specific models, and enable Console access
- You can use CloudTrail to monitor API activity and troubleshoot issues as you build solutions
- Fine-tuned (customized) models are encrypted and stored using service managed keys; only you have access to your customized models through an endpoint
- Support for **Customer Managed Keys (CMK)** so customers can create and control keys to encrypt fine-tuned models
- Support for VPC configuration of fine-tuning jobs

Data protection

ENCRYPTION

CLIENT

AMAZON BEDROCK

HTTPS/TLS1.2/TLS1.3

SSE-BEDROCK (Amazon Bedrock managed keys)

1 Client-side

2 In transit

3 At rest

On-demand vs. provisioned compute capacity

Provisioned capacity compute

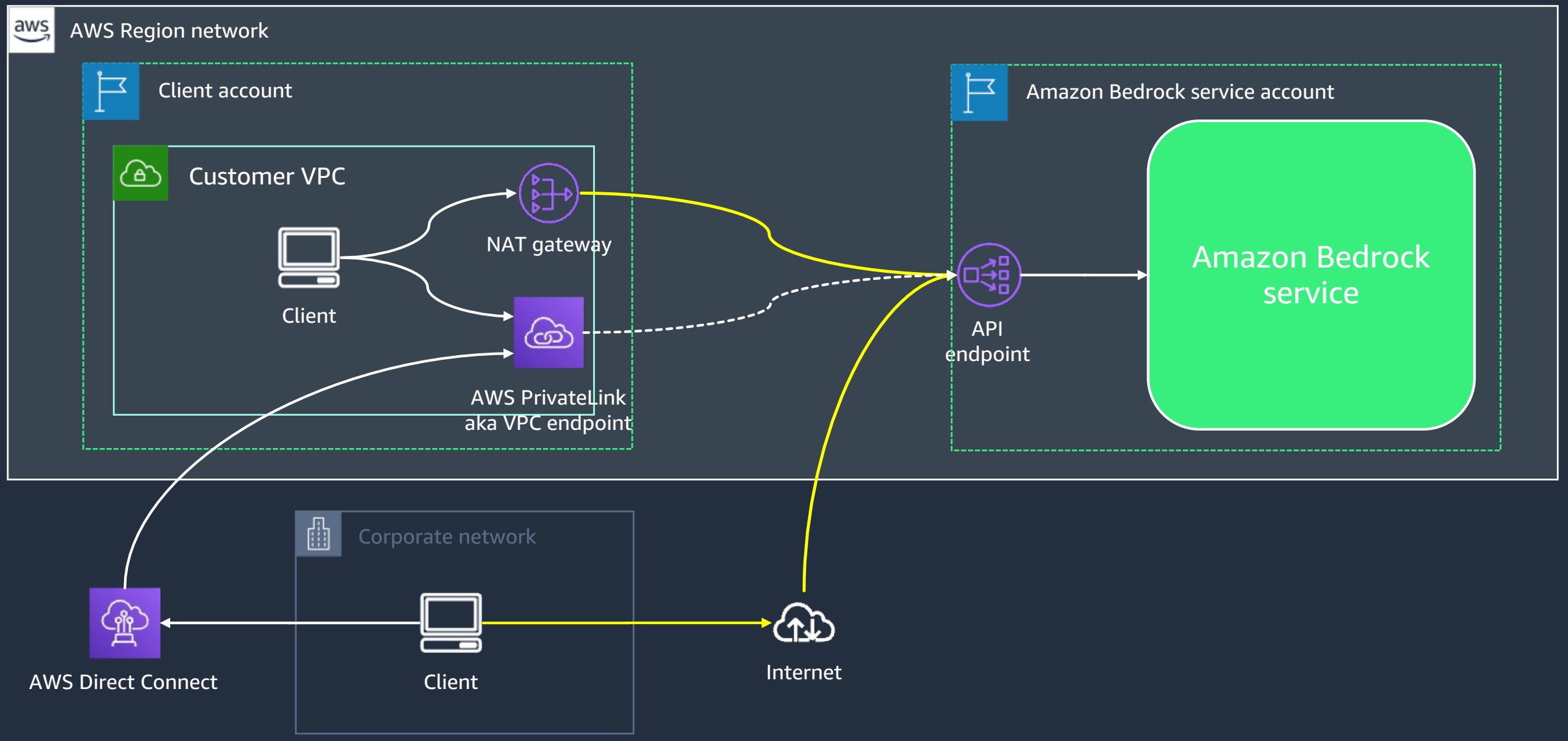
1. Deployment available to a single customer
2. Holds a private copy of a baseline model that may have been fine-tuned by a customer

On-demand compute

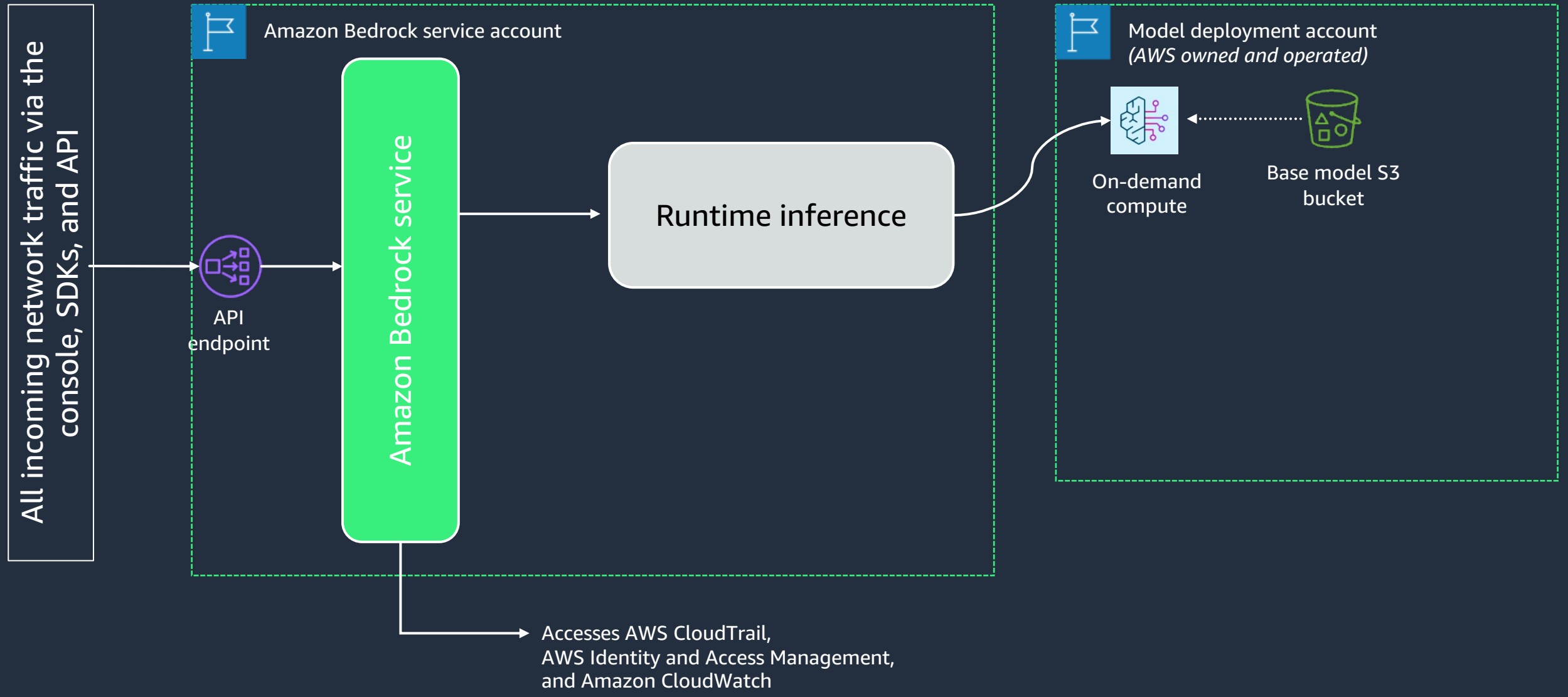
1. Deployment available to all customers
2. Holds a baseline version of a supported model

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

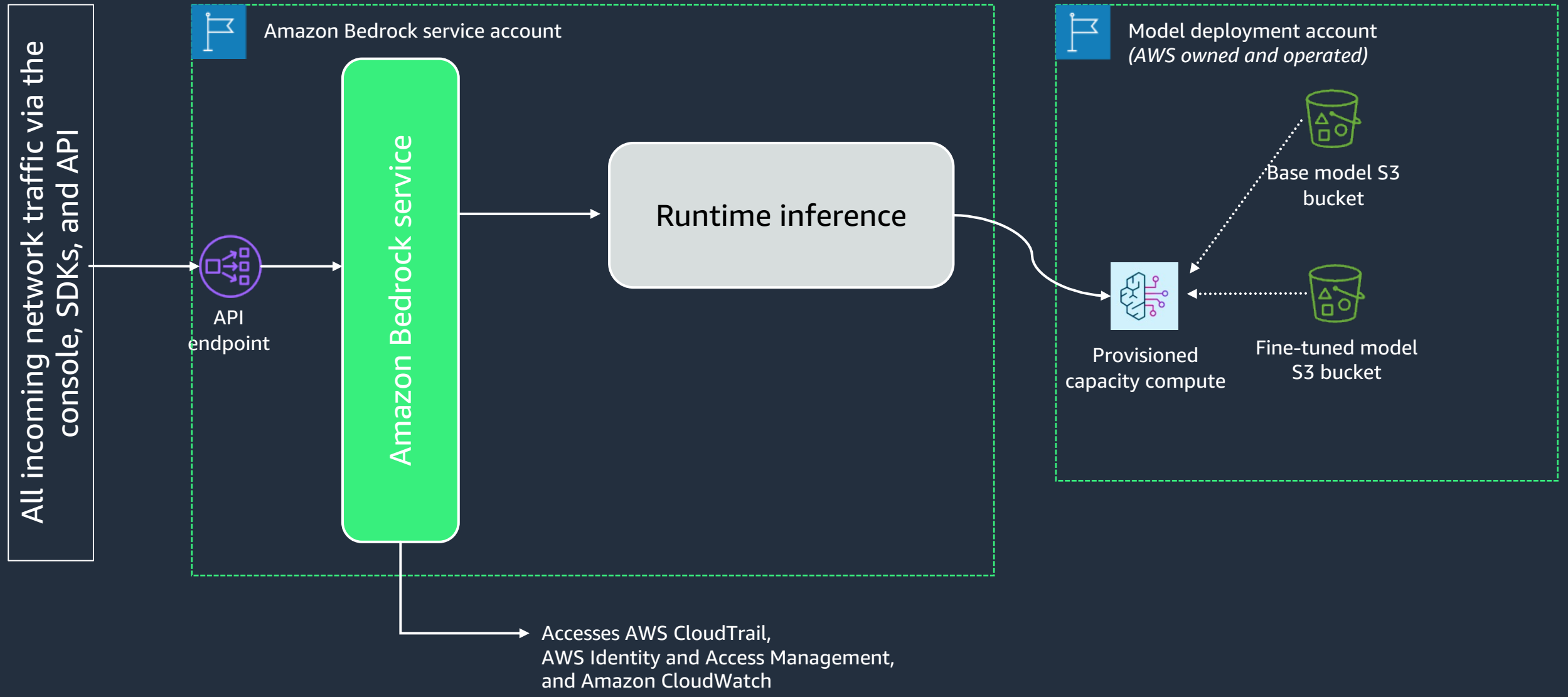
Client connectivity



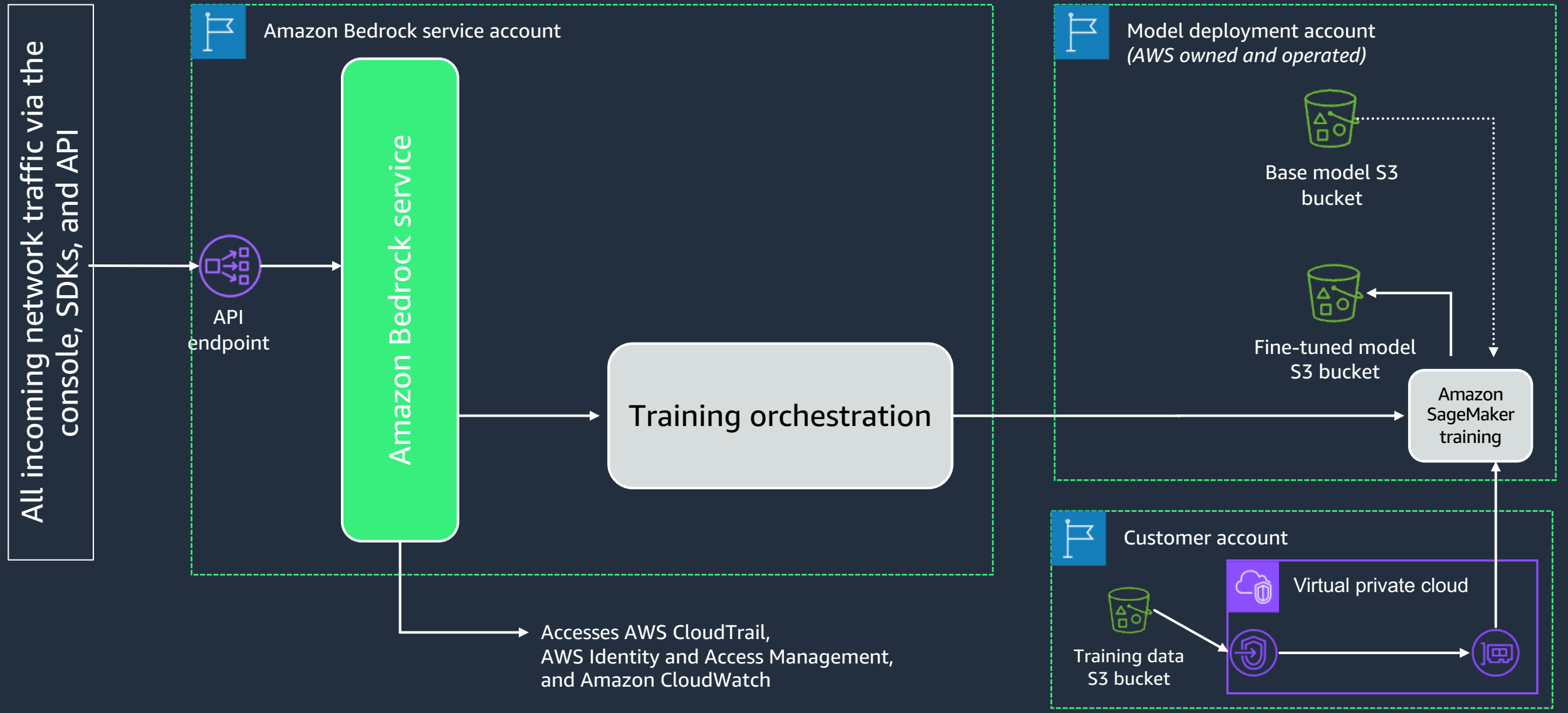
On-demand compute architecture overview



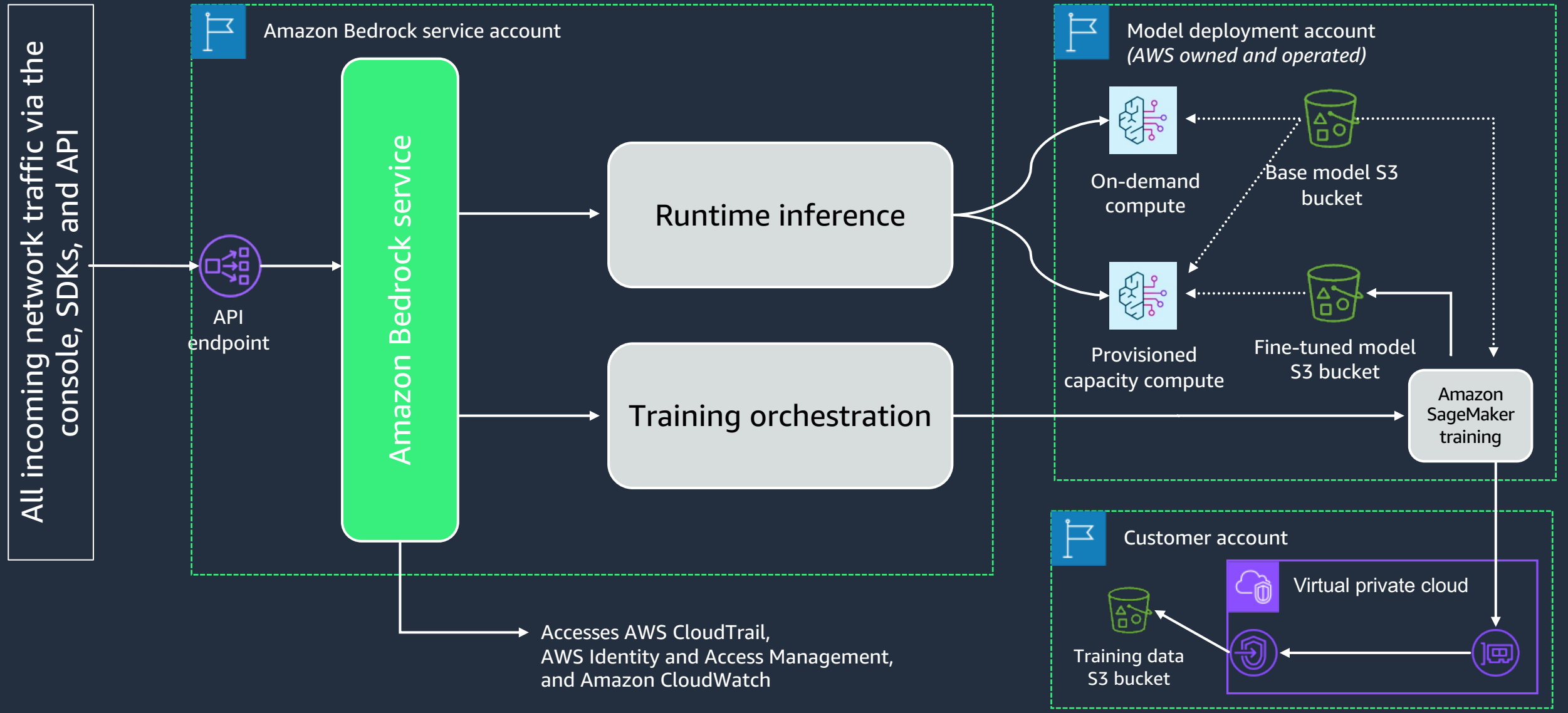
Provisioned capacity architecture overview



Model fine-tuning architecture overview



Complete architecture overview



AWS Identity and Access Management

IAM

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

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>"
    }
}
```

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