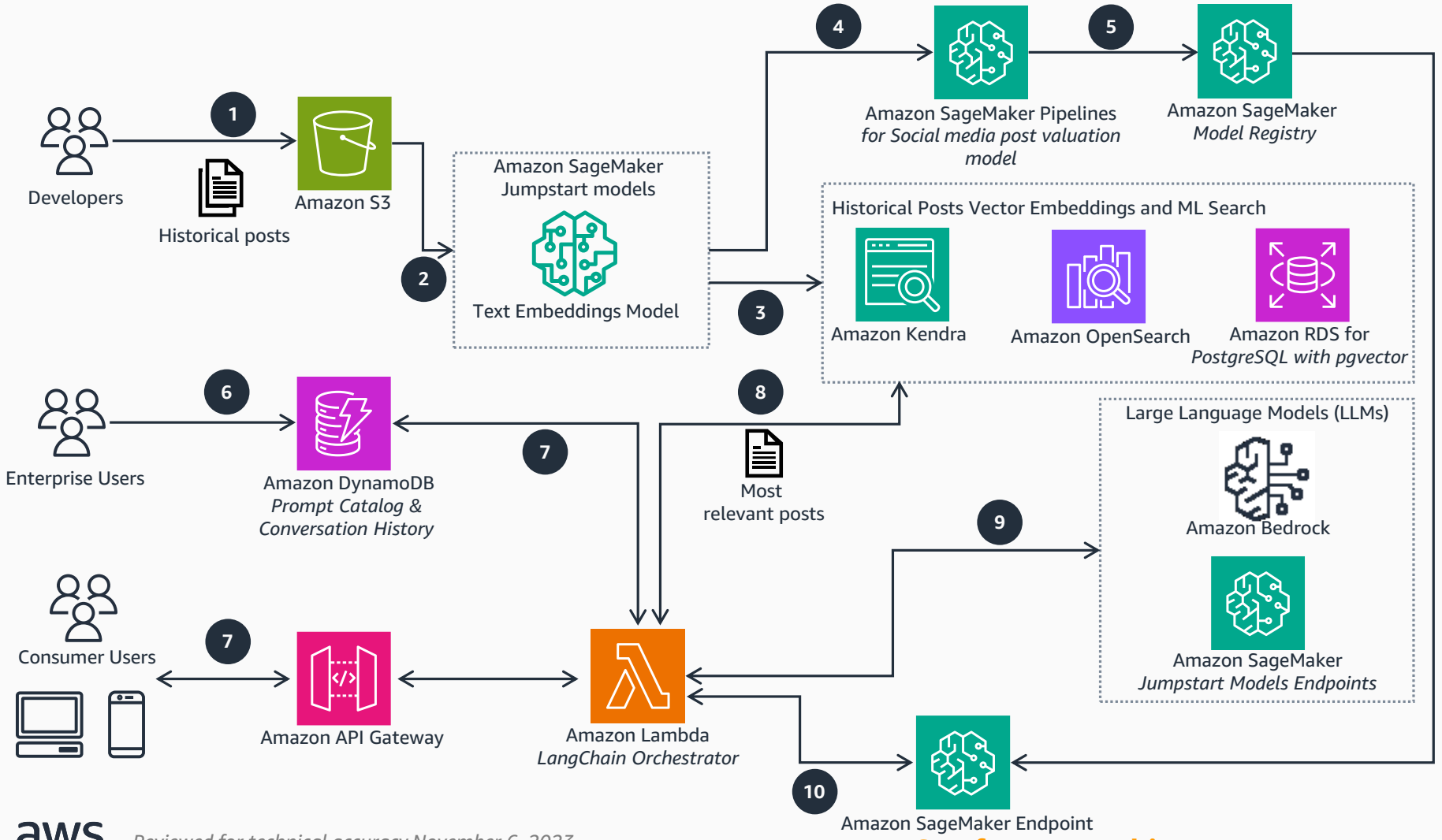
Guidance for Creating Dynamic Content with Brand Intelligence on AWS

This architecture diagram shows how you can design a brand intelligence system to evaluate social media posts using machine learning (ML). It uses Amazon SageMaker to build ML models, including foundation models (FMs), to create dynamic content for social media that is engaging and conforming to your brand guidelines.

Steps 1-7 are outlined here. For descriptions about Steps 8-10, go to slide 2.



1. Developers ingest social media historical data through the social media's API and store them in an **Amazon Simple Storage Service (Amazon S3)** bucket.

2. **Amazon SageMaker Jumpstart** models are used to enrich the raw data by extracting topic information, post sentiment, and embeddings from text and images.

3. Text embeddings are stored in a vector database to enable searchability through Retrieval Augmented Generation (RAG). The vector database options include **Amazon Kendra**, **Amazon OpenSearch Service**, or **Amazon RDS for PostgreSQL** with pgvector.

4. Data scientists process **SageMaker** pipelines, which use historical data, embeddings, and engineered features to build a machine learning (ML) model to predict a user post's engagement rate.

5. Best performing ML models are automatically saved on the **SageMaker** model registry.

6. Enterprise users create a prompt catalog with different styles to choose from and brand guidelines to be enforced. The prompt catalog is stored in **Amazon DynamoDB**.

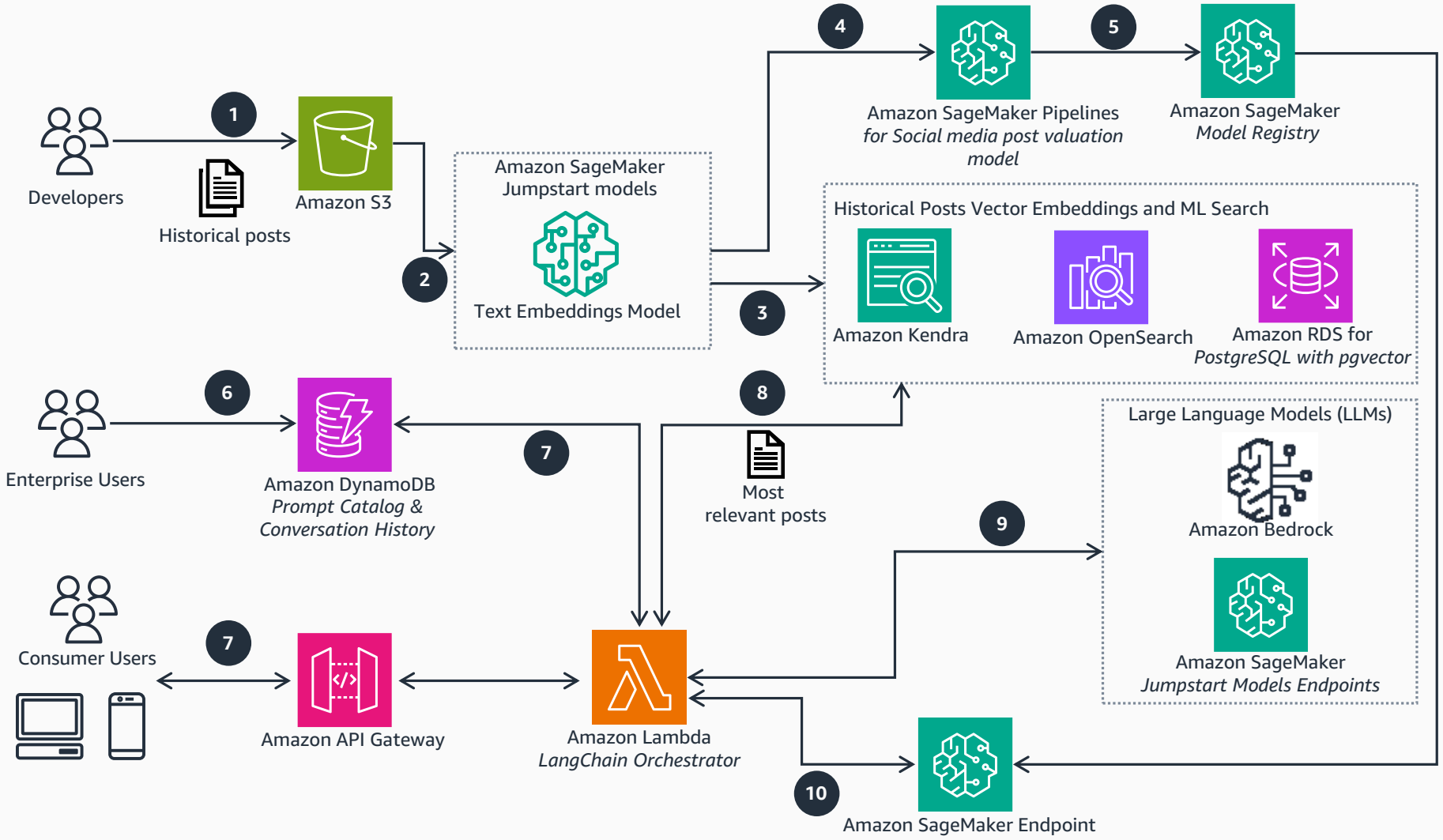
7. The consumer utilizes an app to generate a request for a social media post by providing a description and selecting a specific style (equivalent to a prompt) that aligns with brand guidelines. The consumer app invokes **AWS Lambda**, which contains a LangChain orchestrator accessed through **Amazon API Gateway**. This **Lambda** function processes the input request and prepares the prompt retrieved from the prompt catalog, while the conversation history is also stored in **DynamoDB** for future requests.

Reviewed for technical accuracy November 6, 2023

AWS Reference Architecture

Guidance for Creating Dynamic Content with Brand Intelligence on AWS

Steps 8-10 are outlined here.



8 The LangChain orchestrator **Lambda** function uses RAG to identify similar posts within the historical post data that is stored in vector-embedding databases, and retrieves the most relevant posts.
The retrieved relevant posts are used to create a custom prompt together with a post description and the desired style.

9 The LangChain orchestrator uses the custom prompt and invokes **Amazon Bedrock** and **SageMaker JumpStart** foundation models (FMs) for text and image generation to generate a selection of personalized posts.

10 Each draft will receive a predicted engagement rate using the model trained earlier from **SageMaker**. User chooses the draft having highest engagement rate and publishes to social media.

Reviewed for technical accuracy November 6, 2023

AWS Reference Architecture