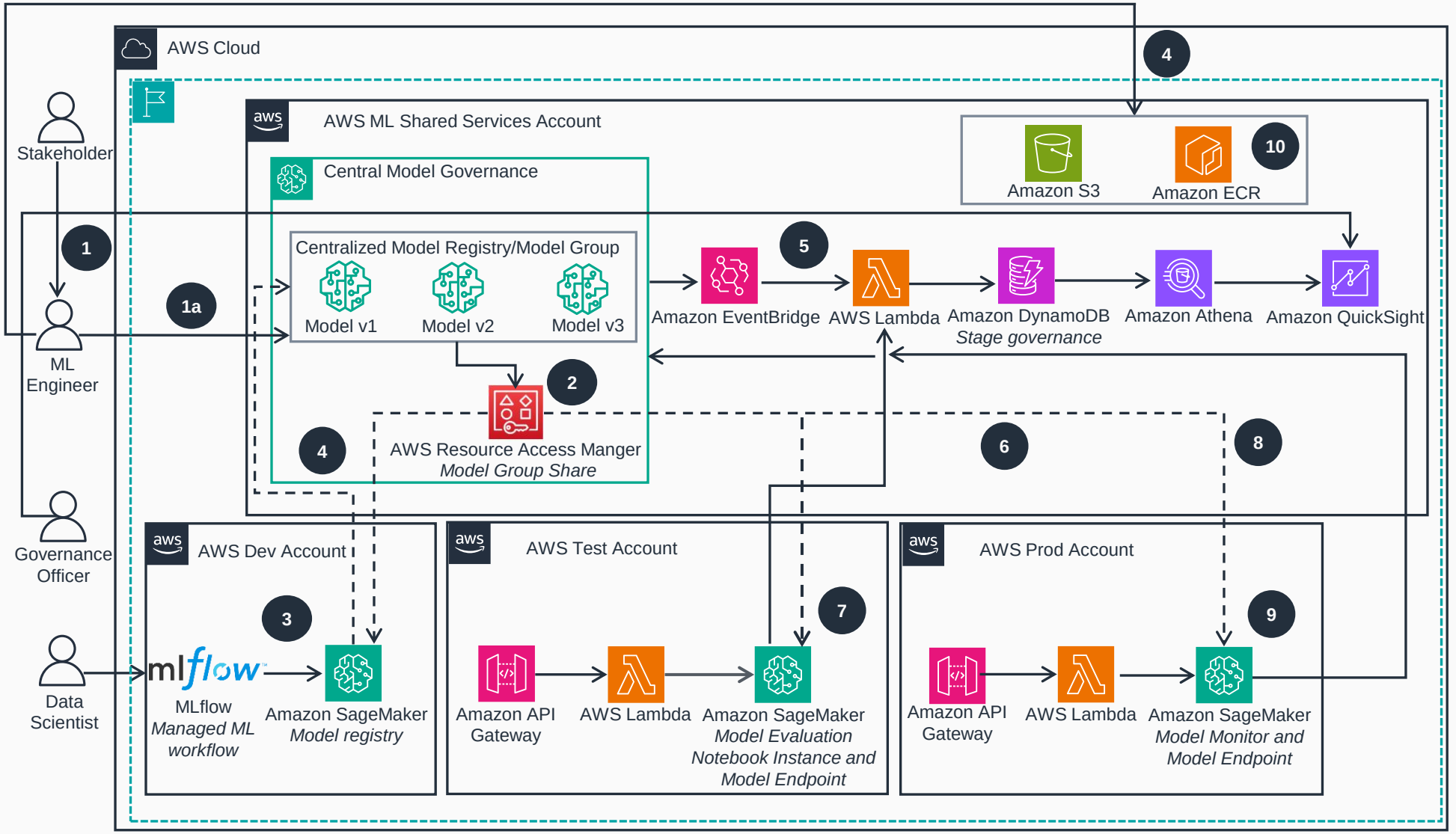
Guidance for Multi-Account Machine Learning Model Governance on AWS

This architecture diagram shows a centralized model governance approach using AWS Resource Access Manager (AWS RAM). With AWS RAM, you can share models using the Amazon SageMaker Model Package Group, a fundamental construct within the Amazon SageMaker Model Registry where Model Package Versions are registered. Steps 1-4 out of 10 are shown here; steps 5-10 are shown on the next slide.



1. The stakeholder, specifically the Data Scientist (DS) team lead, receives a request from the business leader to develop an AI use case, such as a credit risk model.

   1a. The Machine Learning Engineer (MLE) is notified to establish a model group for the development of a new model. The MLE then creates the necessary infrastructure pipeline to set up the new model package group.

2. The MLE sets up the pipeline to share the model group with the necessary permissions (create, describe, update model version) to the ML project team's development account. Optionally, the package group can also be shared with the test and production accounts if local account access to model versions is required.

3. The DS uses MLflow, an open-source platform for managing the end-to-end machine learning lifecycle. MLflow is used within **Amazon SageMaker Studio** to construct model experiments, select a candidate model, and register the model version within the shared model group in the local **Amazon SageMaker** Model Registry.

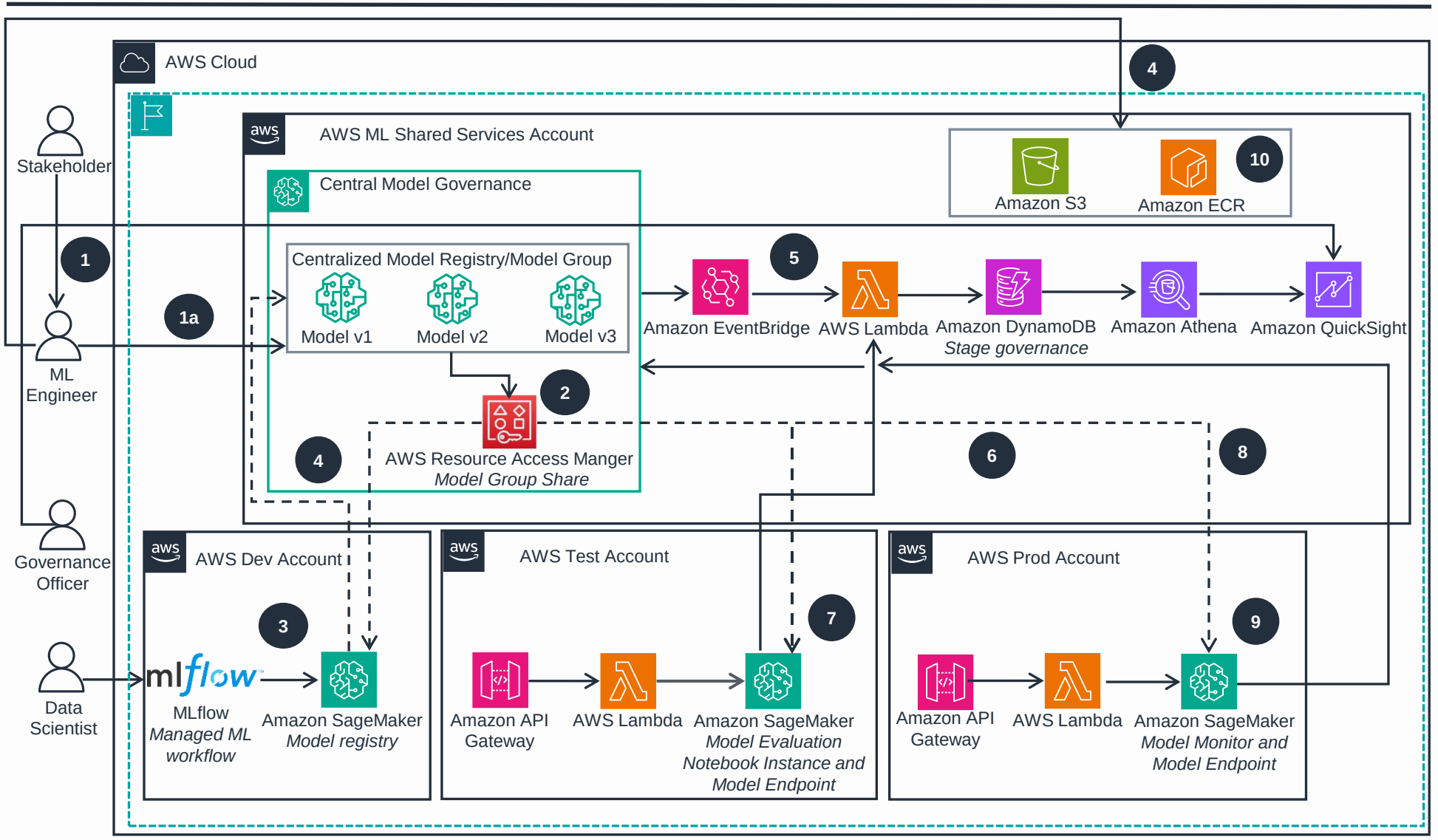
4. Since this is a shared model group, the model version metadata will be recorded in the Centralized Model Registry, and a corresponding link will be maintained in the development account. The MLE can set up **Amazon Simple Storage Service** (Amazon S3) and **Amazon Elastic Container Registry** (Amazon ECR) in the shared account, allowing the DS to store the model artifacts from the development account. The DS is granted the necessary permissions to access the model artifacts in **Amazon S3** and **Amazon ECR** within the shared services account.

Reviewed for technical accuracy November 5, 2024

AWS Reference Architecture

Guidance for Multi-Account Machine Learning Model Governance on AWS

Steps 5-10



5 The Centralized Model Registry triggers an **Amazon EventBridge** rule, which in turn invokes an **AWS Lambda** function that writes the relevant data to an **Amazon DynamoDB** table. The model versions are synchronized with the Model Stage Governance table using **DynamoDB**, which records attributes such as Model Group, Model Version, Model Stage (for example: Dev, Test, Prod), Model Status (pending, approved, rejected), and Model Metrics. **DynamoDB** provides storage for registering models from diverse sources beyond **Amazon SageMaker**, enabling a consolidated view of all enterprise models and metadata. The **DynamoDB** table is the central model governance system that integrates with both use case and model lifecycle stages. It also augments metadata and approvals along with **SageMaker** model registry attributes, and centralizes model governance and performance metrics from production inference endpoints.

6 The model version is approved for deployment into the testing stage and is subsequently deployed into the test account. It's deployed with the necessary infrastructure for invoking the model, such as **Amazon API Gateway** and **Lambda**.

7 The model undergoes integration testing in the test environment, and the quality assurance (QA) model evaluation metrics are updated in the Centralized Model Registry, which are then written to the **DynamoDB** table using a **Lambda** function.

8 The model test results are validated, and the model version is approved for deployment into the production stage. The model is then deployed into the production account, along with the necessary infrastructure for invoking the model, such as **API Gateway** and **Lambda**.

9 The model undergoes A/B testing in the production environment, and the model production metrics are updated in the **DynamoDB** (Model Stage Governance) table. Once satisfactory production results are achieved, the model version is promoted in the production environment. Additionally, model monitoring is enabled at the model endpoint.

10 The Model Governance or Compliance Officer uses the Governance dashboard within **Amazon QuickSight** to execute model governance functions, including reviewing the model for compliance validation and monitoring for risk mitigation.

Reviewed for technical accuracy November 5, 2024

AWS Reference Architecture