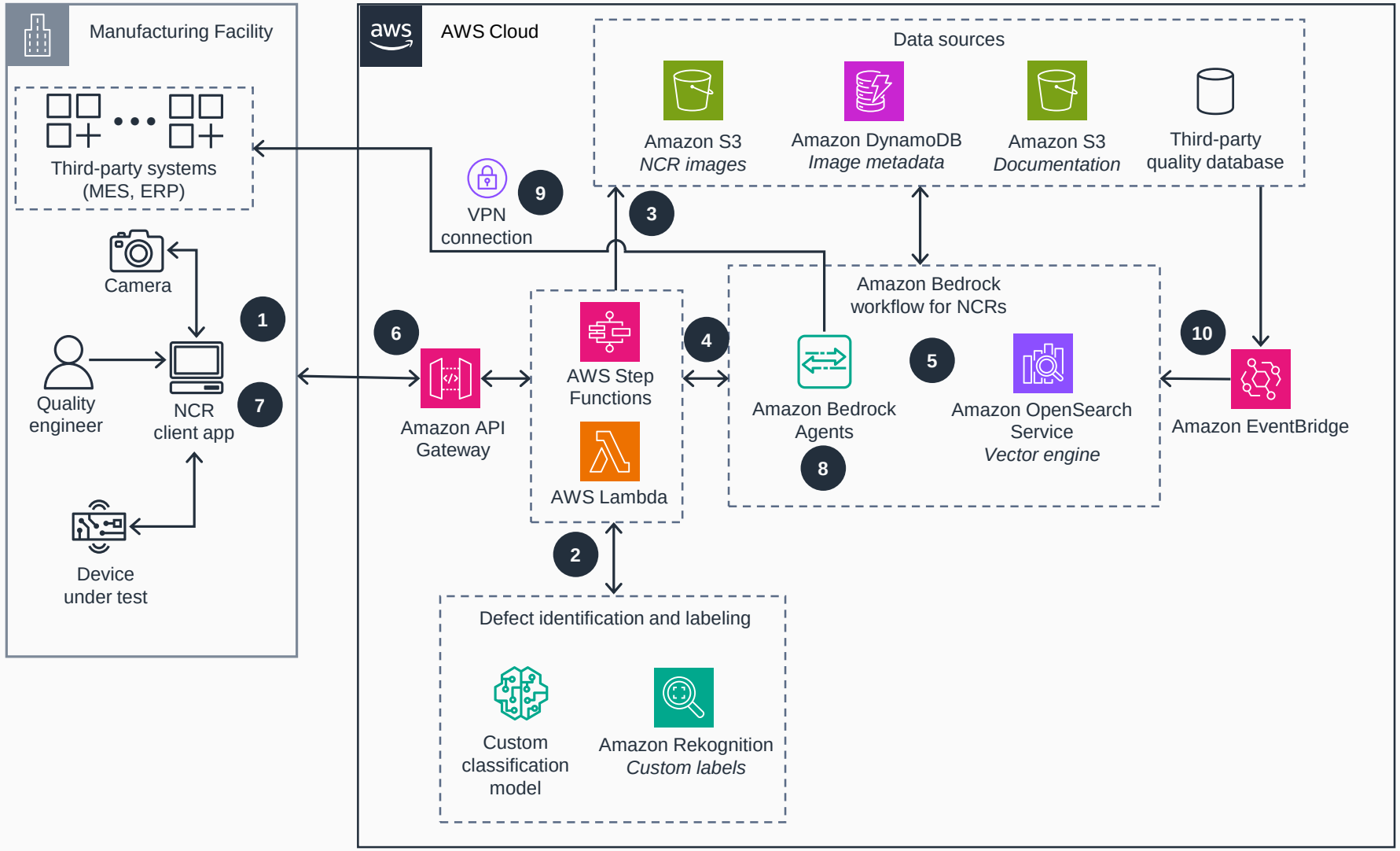
Guidance for Automating Non-Conformance Reviews on AWS

This architecture diagram demonstrates the use of generative AI to automate recommendations for NCR disposition based on natural language and image capture similarity in existing quality ticketing systems.



1. Enter description of non-conformance into quality management interface, with images, part #, and other details. The App sends NCR information to AWS through **Amazon API Gateway**.
2. Process NCR API requests automatically using **AWS Lambda** functions (sequenced by **AWS Step Functions**). **Lambda** invokes computer vision services for defect identification using **Amazon Rekognition Custom Labels** or third-party models.
3. Record image and metadata to **Amazon Simple Storage Service (Amazon S3)** and **Amazon DynamoDB**.
4. Query your large language model (LLM) for recommendation on disposition for the NCR using the **Amazon Bedrock** workflow for NCRs.
5. Find the most relevant historical NCRs using **Amazon Bedrock** native integration to **Amazon OpenSearch Service** vector engine. The LLM uses this info as context to synthesize the best recommendation.
6. Return recommendations to the user through an **API Gateway** response. Recommendations include most relevant past NCRs with associated images.
7. Select final disposition in the UI, taking recommendations into account. This selection is sent to AWS through **API Gateway**.
8. Automate entry of final NCR into a ticketing system using **Amazon Bedrock Agents**.
9. Automate other workflows through **Amazon Bedrock Agents** using APIs to third-party systems over a secure **AWS Virtual Private Network (AWS VPN)**. For example, if the NCR requires rework, queue up work orders in your manufacturing execution system (MES) or replacement part orders in your enterprise resource planning (ERP.)
10. Continuously update your knowledge bases based on new NCR feedback, using **Amazon EventBridge** to trigger scheduled or on-demand re-indexing. Data mutations trigger **EventBridge** events, which coordinate idempotent re-indexing through **Step Functions** and **Lambda**, while images use versioning or timestamp-based naming.

Reviewed for technical accuracy May 15, 2025

AWS Reference Architecture