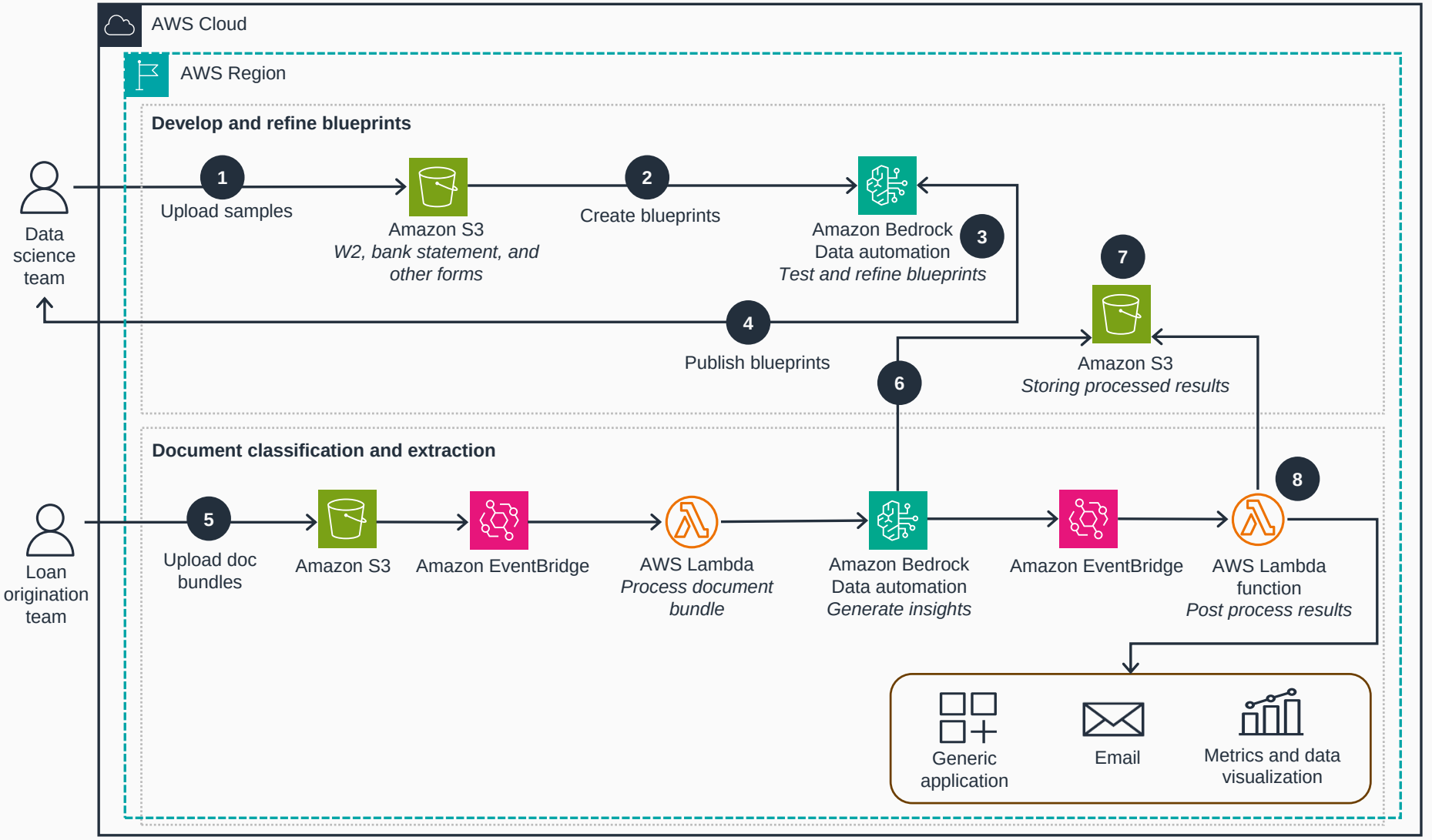
Guidance for Multimodal Data Processing Using Amazon Bedrock Data Automation

Intelligent document processing

This architecture diagram shows how to perform document classification and extraction using a loan origination processing example for a financial services company.



Develop and refine blueprints

1. The data science team uploads sample documents to an **Amazon Simple Storage Service (Amazon S3)** bucket.
2. The data science team uses provided blueprints and creates new custom blueprints for each document class: W2, Pay Slip, Drivers License, 1099, and Bank Statement. Each sample is processed, and generative AI prompts extract fields (such as first and last name, gross pay, capital gains, and closing balance).
3. The blueprints are tested and refined. Key normalizations, transformations, and validations are added.
4. The blueprints are managed and stored in the Amazon Bedrock Data Automation feature.

Document classification and extraction

5. Using an “Object Created” event, **Amazon EventBridge** triggers an **AWS Lambda** function when documents are uploaded to **Amazon S3**. This **Lambda** function then uses the **Amazon Bedrock** Data Automation feature to process the uploaded documents.
6. The processing workflow in the **Amazon Bedrock** Data Automation feature includes document splitting based on logical boundaries, with each split containing up to 20 pages. Each page is classified into a specific document type and matched to appropriate blueprints. The corresponding blueprint is then invoked for each page, executing key normalizations, transformations, and validations. This entire process operates asynchronously, allowing for efficient handling of multiple documents and large data volumes.
7. **Amazon Bedrock** Data Automation stores the results in a **Amazon S3** bucket for later processing and triggers **EventBridge**.
8. **EventBridge** triggers the **Lambda** function to process the JSON results of **Amazon Bedrock** Data Automation. The processing results are sent to downstream processing systems.

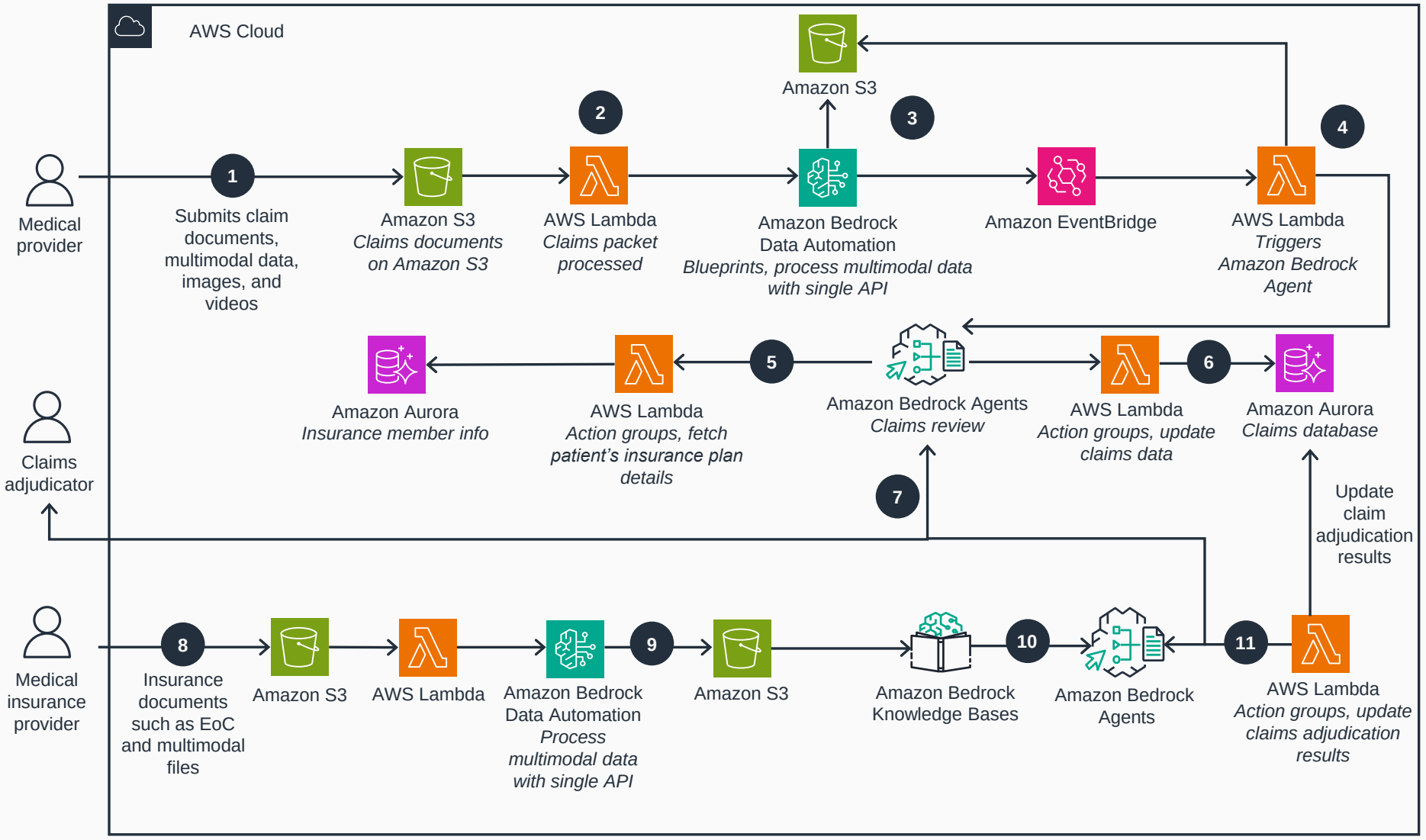
Reviewed for technical accuracy November 14, 2024

AWS Reference Architecture

Guidance for Multimodal Data Processing Using Amazon Bedrock Data Automation

Medical claims processing

This architecture diagram shows how to automate medical claims processing with multimodal input data and processing to improve efficiency and accuracy.



Reviewed for technical accuracy November 14, 2024

AWS Reference Architecture

Document submission

1. Providers submit claims documents, images, and videos to **Amazon S3**.
2. A workflow is triggered in **Amazon Bedrock** Data Automation.

Claims review

3. Developers create blueprints in **Amazon Bedrock** Data Automation to extract relevant data.
4. **Amazon Bedrock** Data Automation processes documents, images, and videos by extracting text, tables, objects, transcripts; normalizing structuring the data; and flagging low-confidence items for review. **Amazon Bedrock** Data Automation stores the data in **Amazon S3** and triggers **EventBridge**.
5. EventBridge triggers Lambda, which retrieves the **Amazon Bedrock** Data Automation output from the **S3** bucket.
6. **Amazon Bedrock** Agents uses the **Lambda** function to fetch the patient's insurance plan details from **Amazon Aurora**.
7. **Amazon Bedrock** Agents then updates the claims database in **Aurora**.
8. Adjudicators verify important fields and focus on low-confidence items.

Eligibility determination

9. Explanation of Coverage (EoC) documents, images, and videos are stored in **Amazon S3**. **Amazon Bedrock** Data Automation processes multimodal data with a single API and stores it in **Amazon S3**. It is then processed, embedded, and stored in a vector collection for **Amazon Bedrock** Knowledge Bases.
10. **Amazon Bedrock** Agents calculates eligibility using extracted data and indexed information.
11. **Amazon Bedrock** Agents updates the claims database and notifies the adjudicator. The adjudicator reviews and approves or adjusts the claim efficiently.