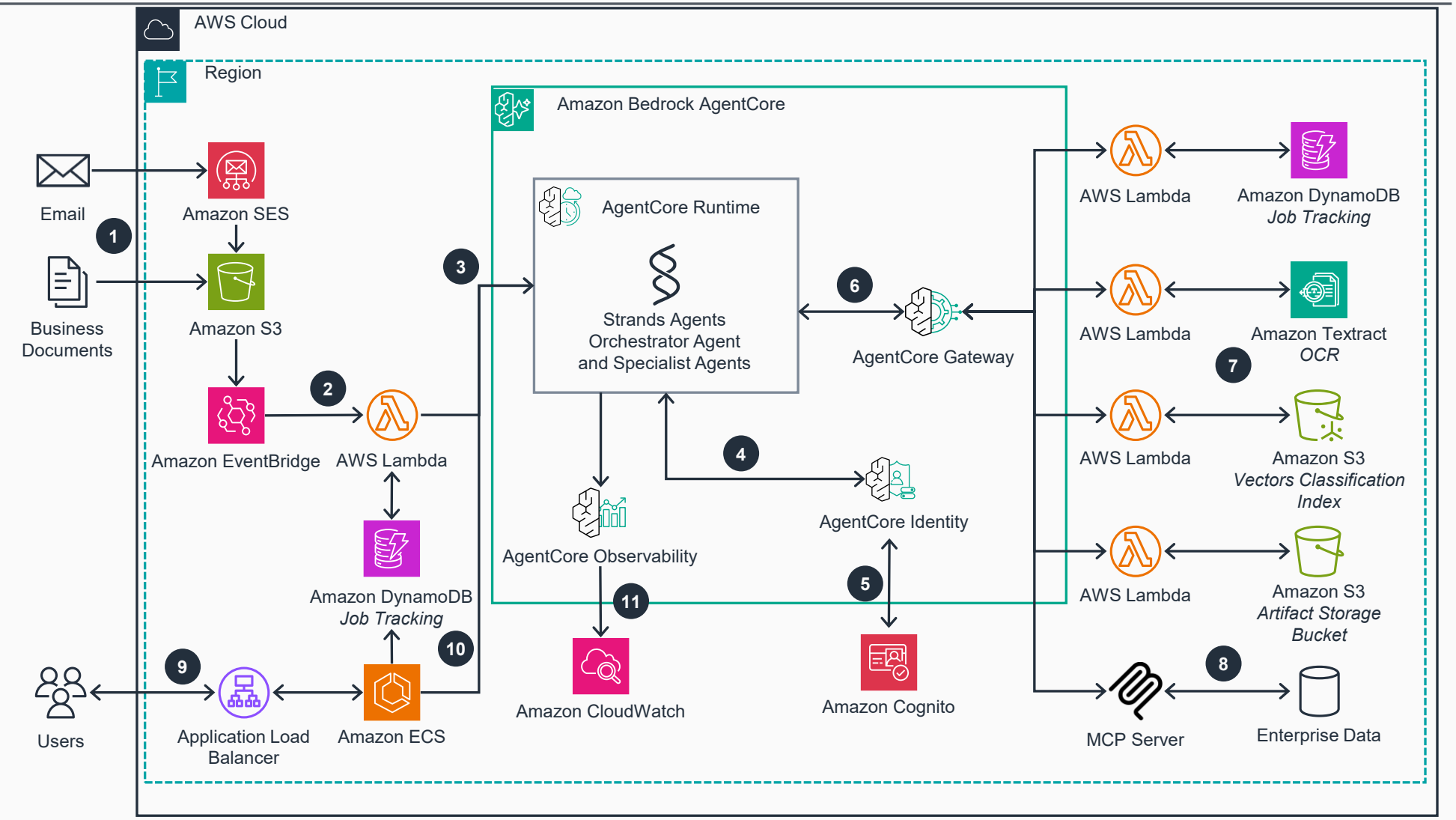
Guidance for Agentic Intelligent Document Processing on AWS

This architecture implements an intelligent document processing system using Amazon Bedrock AgentCore, where multiple specialized agents collaborate through graph-based workflows to automatically identify, extract, validate, and learn from business documents with minimal human intervention. This slide shows Steps 1-6.



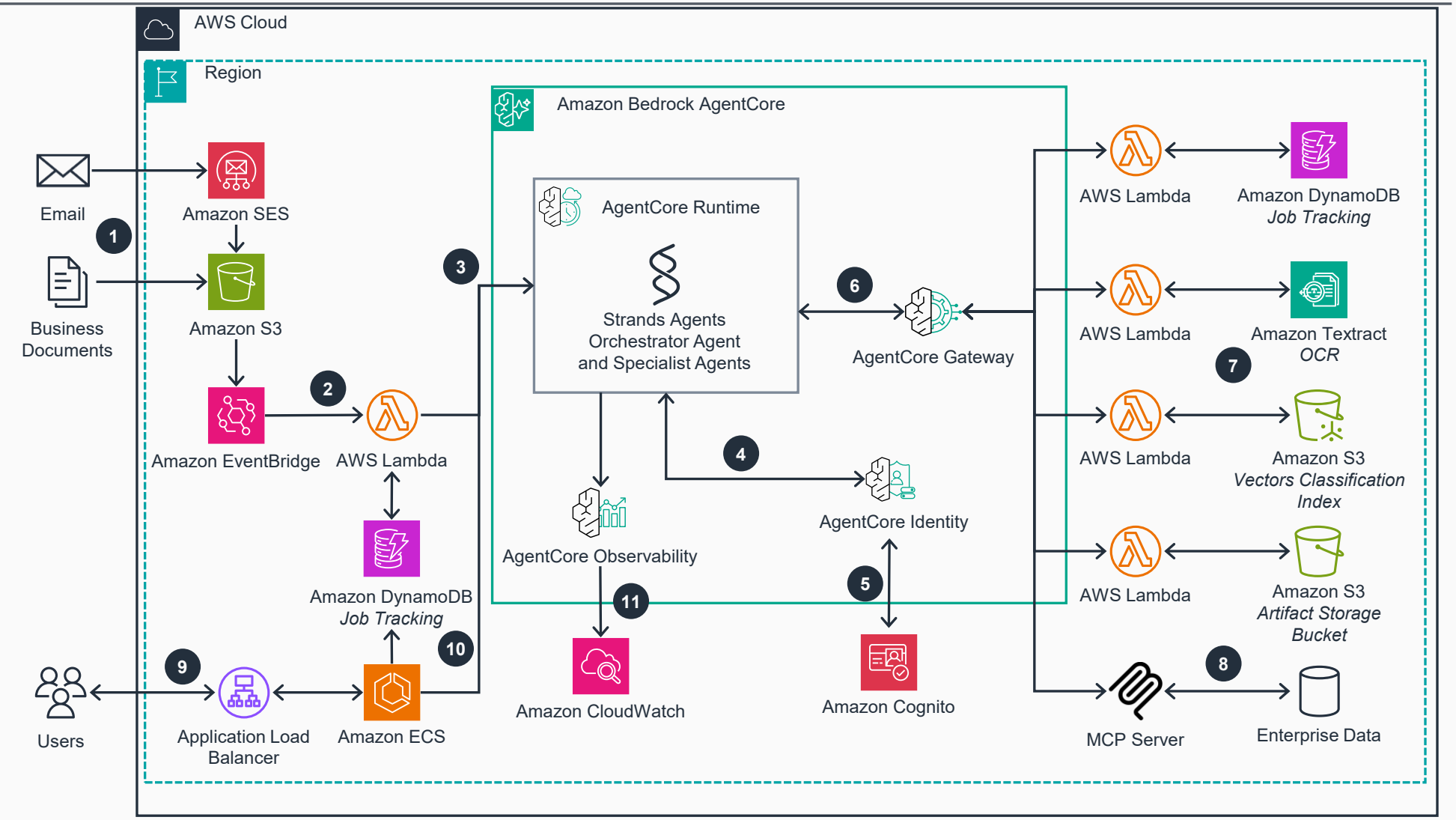
1. Business documents including purchase orders, invoices, shipping tickets, and other transactional records arrive via **Amazon Simple Email Service (Amazon SES)** or direct upload to **Amazon Simple Storage Service (Amazon S3).**
2. **Amazon EventBridge** invokes an **AWS Lambda** function when new documents arrive in **Amazon S3**.
3. The **Lambda** function creates a tracking job in **Amazon DynamoDB** and invokes **Amazon Bedrock AgentCore Runtime** to orchestrate document processing through **Strands Agents**.
4. **Amazon Bedrock AgentCore Runtime** queries **Amazon Bedrock AgentCore Identity** for machine-to-machine credentials, enabling secure access to AWS resources and third-party services. **Amazon Bedrock AgentCore Identity** retrieves a workload access token from **Amazon Cognito**.
5. Using this token, **Amazon Bedrock AgentCore Runtime** discovers and accesses tools from **Amazon Bedrock AgentCore Gateway**. The agent selects tools based on document type and processing requirements.
6. **Amazon Bedrock AgentCore Gateway** transforms existing APIs and Lambda functions into agent-compatible tools with minimal code. It provides a searchable Model Context Protocol (MCP) interface for AWS resources, external tools, and databases, enabling secure discovery and communication.

Reviewed for technical accuracy November 17, 2025

AWS Reference Architecture

Guidance for Agentic Intelligent Document Processing on AWS

This architecture implements an intelligent document processing system using Amazon Bedrock AgentCore, where multiple specialized agents collaborate through graph-based workflows to automatically identify, extract, validate, and learn from business documents with minimal human intervention. This slide shows Steps 7-11.



7. Agents extract text from documents using **Amazon Textract**, classify entities using vector embeddings in **Amazon S3**, and determine next processing steps.
8. Agents access Enterprise Data via MCP servers to validate extracted content against business rules.
9. Users monitor status and perform administrative actions through a chat-enabled dashboard hosted on **Amazon Elastic Container Service (Amazon ECS)**.
10. The dashboard invokes **Amazon Bedrock AgentCore Runtime**, enabling users to troubleshoot document processing issues in real-time.
11. **Amazon Bedrock AgentCore Observability** traces, debugs, and monitors agent performance by automatically logging telemetry data to **Amazon CloudWatch**. This provides detailed visualizations of each workflow step, enabling inspection of execution paths and identification of performance bottlenecks.

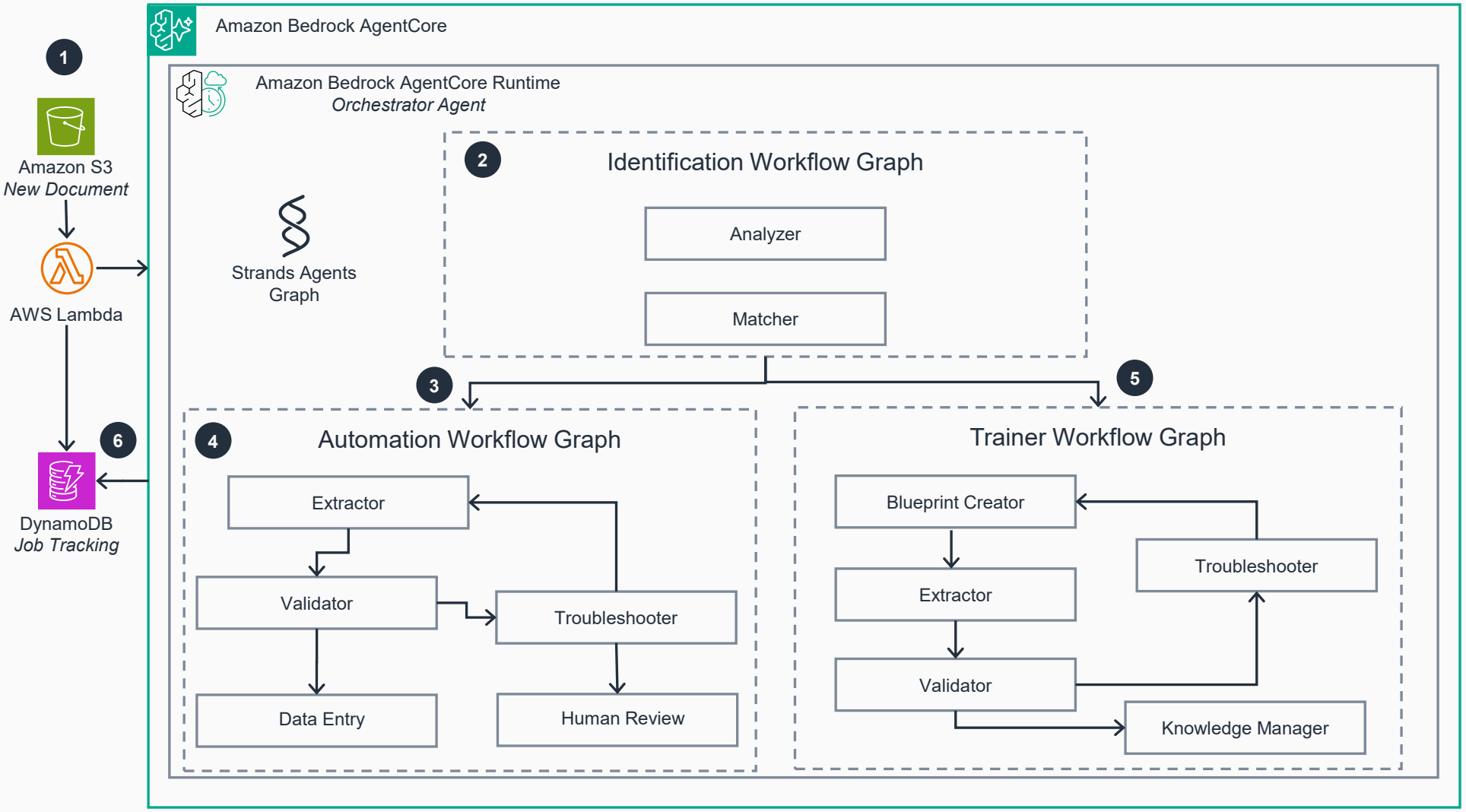
Reviewed for technical accuracy November 17, 2025

AWS Reference Architecture

Guidance for Agentic Intelligent Document Processing on AWS

Multi-Agent Orchestration

This diagram details the multi-agent workflow running within Amazon Bedrock AgentCore Runtime from the previous architecture, showing how specialized AI agents collaborate through graph-based orchestration to automatically process documents, with dual workflows handling known document types through automated extraction and validation, while unknown documents trigger blueprint creation with human oversight for continuous system learning.



1. **Amazon Bedrock AgentCore** orchestrates document processing through a multi-agent system built with **Strands Agents**, using graph-based workflows to automate processing tasks.
2. When a new document arrives in **Amazon S3**, an **AWS Lambda** function creates a job record and asynchronously invokes the Orchestrator Agent to begin processing. The Orchestrator first identifies the document type and sender, then queries the knowledge base to determine if this document type has existing processing rules.
3. If a match exists, the Orchestrator initiates the Automation Workflow to process the document using the stored processing rules.
4. The Automation Workflow extracts and validates data according to processing rules. Valid data is automatically sent downstream systems. Invalid data triggers the Troubleshooter agent, which reviews errors against the source document and rules, then sends corrective instructions to the Extractor. After three failed attempts, the document is routed for human review.
5. If no match is found, the Orchestrator routes the document to the Trainer Workflow. This workflow creates a new processing blueprint by extracting and validating data from the document. Upon successful extraction, the workflow adds the new blueprint to the knowledge base and triggers human review before enabling automated processing for this document type.
6. Throughout processing, the Orchestrator tracks document progress and updates job status in **Amazon DynamoDB**. Upon completion, processed documents are stored in **Amazon S3** for downstream processing and integration. Users can monitor processing status, review extraction results, and access completed documents through the web dashboard interface.

Reviewed for technical accuracy November 17, 2025

AWS Reference Architecture