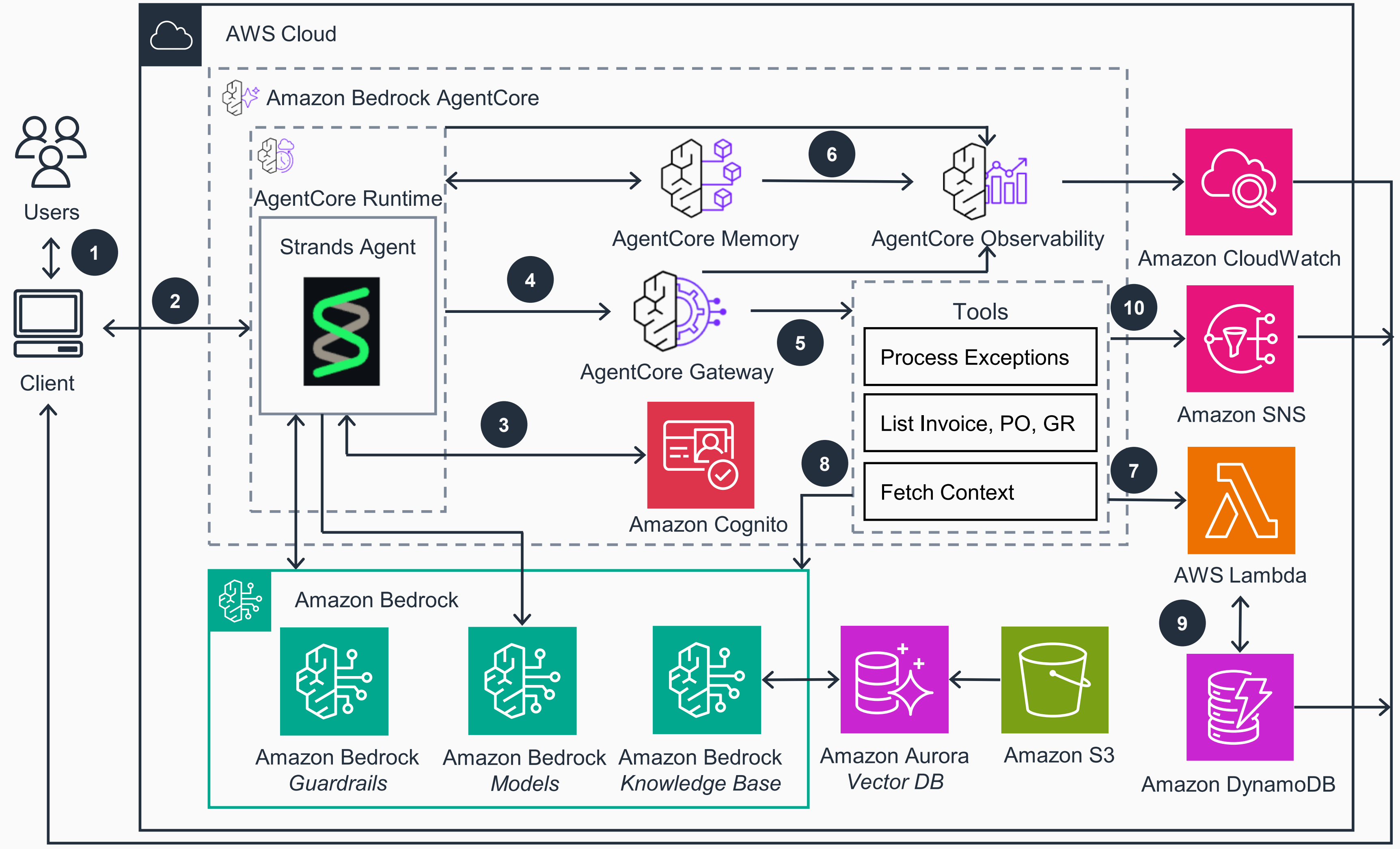
Guidance for Agentic ERP Accounts Payable and Receivable Exception Handling on AWS

Accounts Payable – Strands SDK

This architecture diagram provides a customizable solution to accelerate developers implementing SAP procure-to-pay (P2P) exception handling using intelligent AI agents powered by Strand Agents SDK and AWS Bedrock AgentCore.



1. Users access the application through a web client.
2. The client communicates with the **Amazon Bedrock AgentCore** System, passing natural language user queries and questions related to escalated invoices and exceptions to the **Strands Agent** hosted on **AgentCore Runtime.**
3. The **Strands Agent** authenticates credentials with **Amazon Cognito**.
4. **Strands Agent** initializes the model context protocol (MCP) client that connects to the **AgentCore Gateway** endpoint.
5. **AgentCore Gateway** discovers and invokes SAP procure-to-pay tools across AWS services and MCP-compatible APIs to update exceptions and related financial documentation.
6. **AgentCore Observability** captures logging, metrics, and traces to monitor agent performance in production environments. Users can view collected data from **Amazon CloudWatch**.
7. The **AWS Lambda** function processes the P2P exception called from **AgentCore Gateway**, enabling intelligent routing, escalation workflows, and database actions.
8. **AWS Lambda** retrieves related invoice documents, goods receipts and customer information from the **Amazon Bedrock Knowledge Base** that uses **Amazon Aurora** with vector extension for vector storage and **Amazon Simple Storage Service (Amazon S3)** for document storage to form a context of the exception.
9. **AWS Lambda** retrieves and processes customer and vendor data from SAP backend systems stored in **Amazon DynamoDB**.
10. **Amazon Simple Notification Service (Amazon SNS)** delivers automated email notifications to vendors, financial analysts, and clients to keep them informed of status updates and required actions.

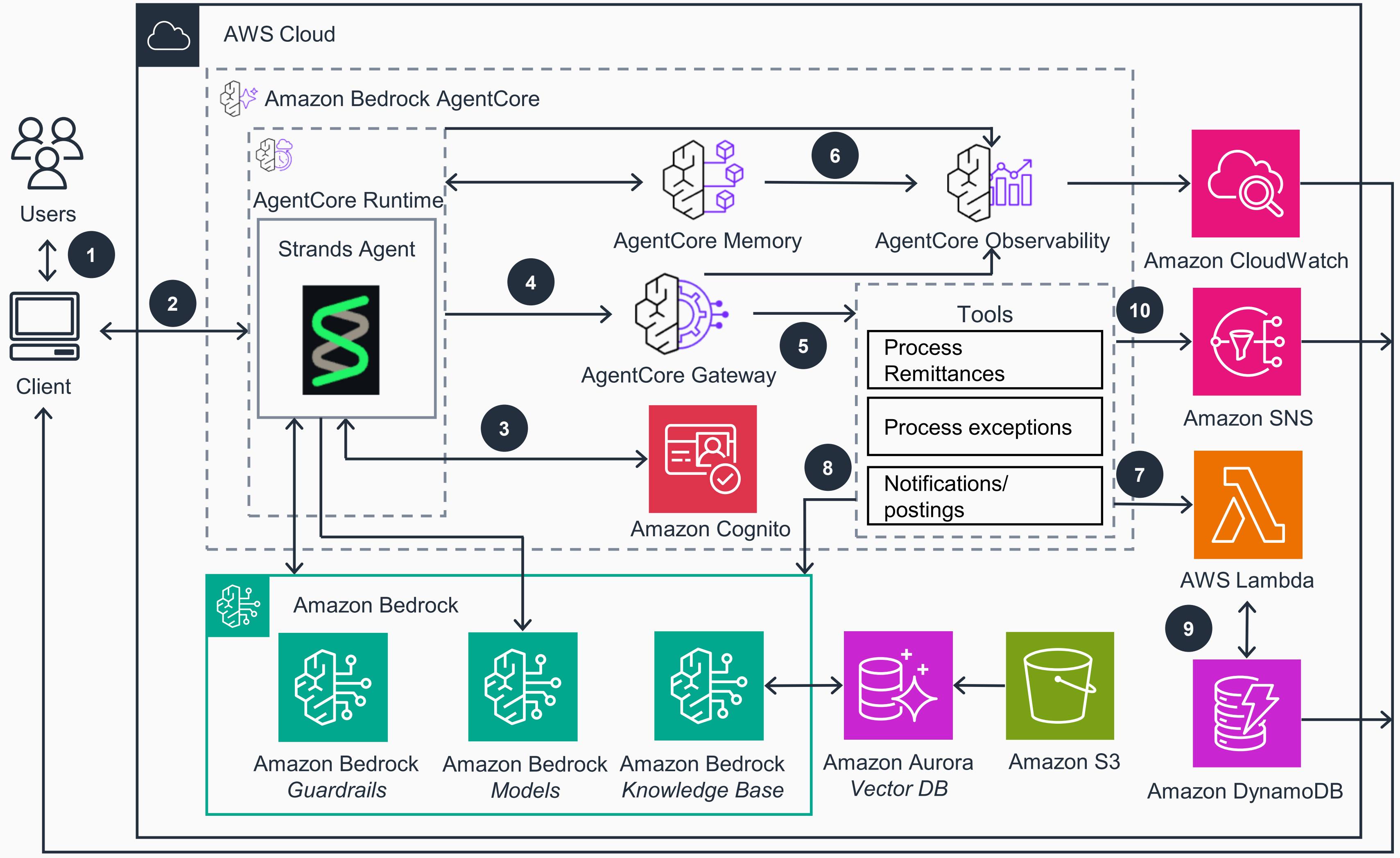
Reviewed for technical accuracy November 19, 2025

AWS Reference Architecture

Guidance for Agentic ERP Accounts Payable and Receivable Exception Handling on AWS

Accounts Receivable – Strands SDK

This architecture diagram provides a customizable solution to accelerate developers implementing SAP order-to-cash (OTC) exception handling using intelligent AI agents powered by Strand Agents SDK and AWS Bedrock AgentCore.



1. Users access the application through a web client.
2. The client communicates with the **Amazon Bedrock AgentCore** System, passing natural language user queries and questions related to escalated invoices and exceptions to the **Strands Agent** hosted on **AgentCore Runtime.**
3. The **Strands Agent** authenticates credentials with **Amazon Cognito**.
4. **Strands Agent** initializes the model context protocol (MCP) client that connects to **AgentCore Gateway** endpoint.
5. **AgentCore Gateway** discovers and invokes SAP order-to-cash tools across AWS services and MCP-compatible APIs to update exceptions and related financial documentation.
6. **AgentCore Observability** captures logging, metrics, and traces to monitor agent performance in production environments. Users can view collected data from **Amazon CloudWatch**.
7. The **AWS Lambda** function processes the order-to-cash exception called from **AgentCore Gateway**, enabling intelligent routing, escalation workflows, and database actions.
8. **AWS Lambda** retrieves related invoice documents, goods, receipts, and customer information from the **Amazon Bedrock Knowledge Base** that uses **Amazon Aurora** with vector extension for vector storage and **Amazon Simple Storage Service (Amazon S3)** for document storage to form a context of the exception.
9. **AWS Lambda** retrieves and processes customer and vendor data from SAP backend systems stored in **Amazon DynamoDB**.
10. **Amazon Simple Notification Service (Amazon SNS)** delivers automated email notifications to vendors, financial analysts, and clients to keep them informed of status updates and required actions.

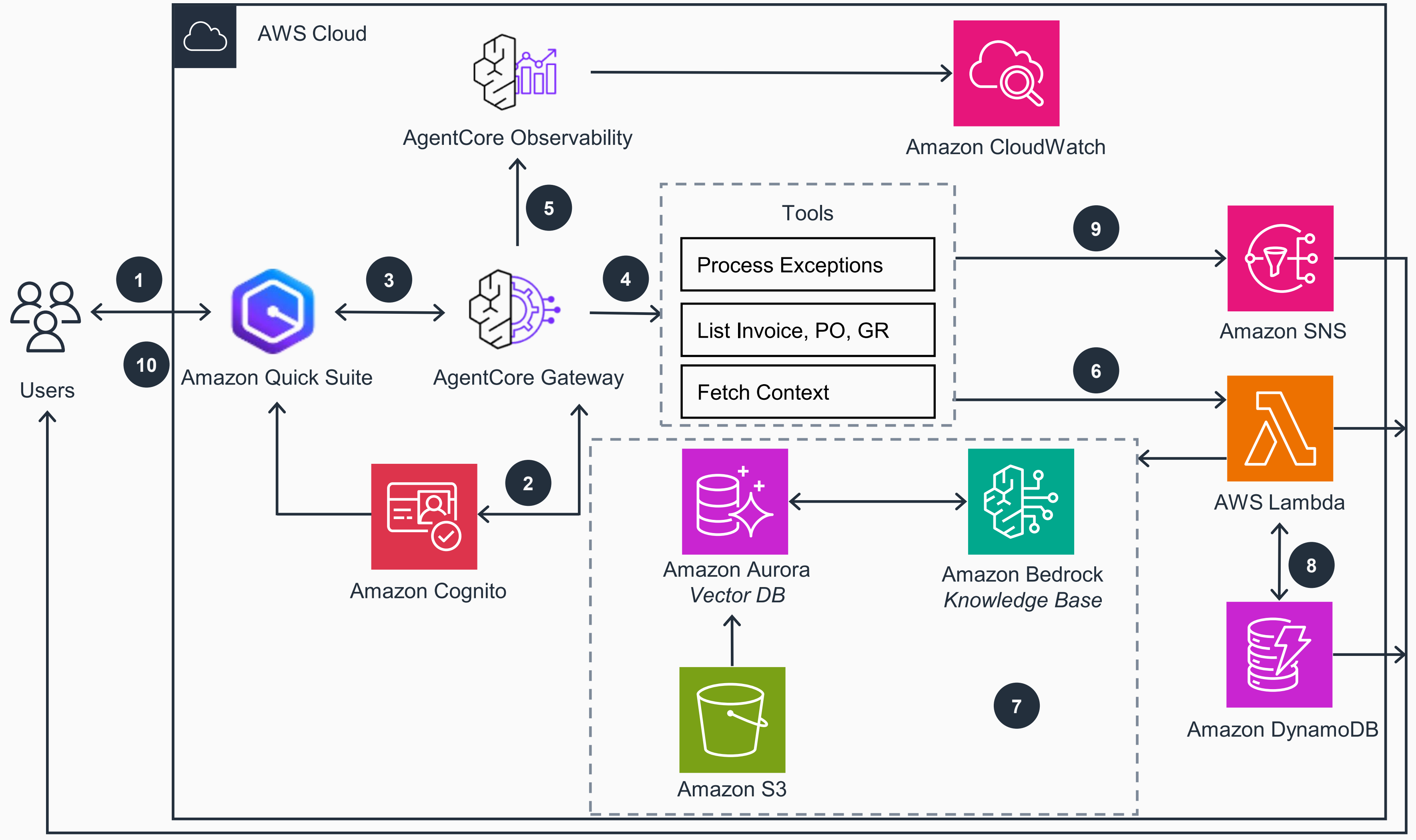
Reviewed for technical accuracy November 19, 2025

AWS Reference Architecture

Guidance for Agentic ERP Accounts Payable and Receivable Exception Handling on AWS

Accounts Payable - Quick Suite

This architecture diagram provides a customizable solution to accelerate developers implementing SAP procure-to-pay (P2P) exception handling using intelligent AI agents powered by Amazon Quick Suite and AWS Bedrock AgentCore.



AWS Reference Architecture

1. Users authenticate to **Amazon Quick Suite** through **Amazon Cognito** User Pools using single sign-on credentials.
2. Integrate with **AgentCore Gateway** through **Amazon Cognito** inbound authentication using authenticated credentials.
3. **Amazon Quick Suite Flows** initializes a model context protocol (MCP) client that connects to **AgentCore Gateway** endpoint.
4. **AgentCore Gateway** discovers and invokes SAP procure-to-pay tools across AWS services and MCP-compatible APIs to update exceptions and related financial documentation.
5. **AgentCore Observability** captures logging, metrics, and traces to monitor agent performance in production environments. Users can view collected data from **Amazon CloudWatch.**
6. The **AWS Lambda** function processes the P2P exception called from **AgentCore Gateway**, enabling intelligent routing, escalation workflows, and database actions.
7. **AWS Lambda** retrieves related invoice documents, goods receipts and customer information from the **Amazon Bedrock Knowledge Base** that uses **Amazon Aurora** with vector extension for vector storage and **Amazon Simple Storage Service (Amazon S3)** for document storage to form a context of the exception.
8. **AWS Lambda** retrieves and processes customer and vendor data from ERP backend systems stored in **Amazon DynamoDB**.
9. **Amazon Simple Notification Service (Amazon SNS)** delivers automated email notifications to vendors, financial analysts, and clients to keep them informed of status updates and required actions.
10. **Amazon Quick Suite** provides recommendations for resolving exceptions and returns the results of updated invoices and purchase orders with database operation status updates in natural language format.