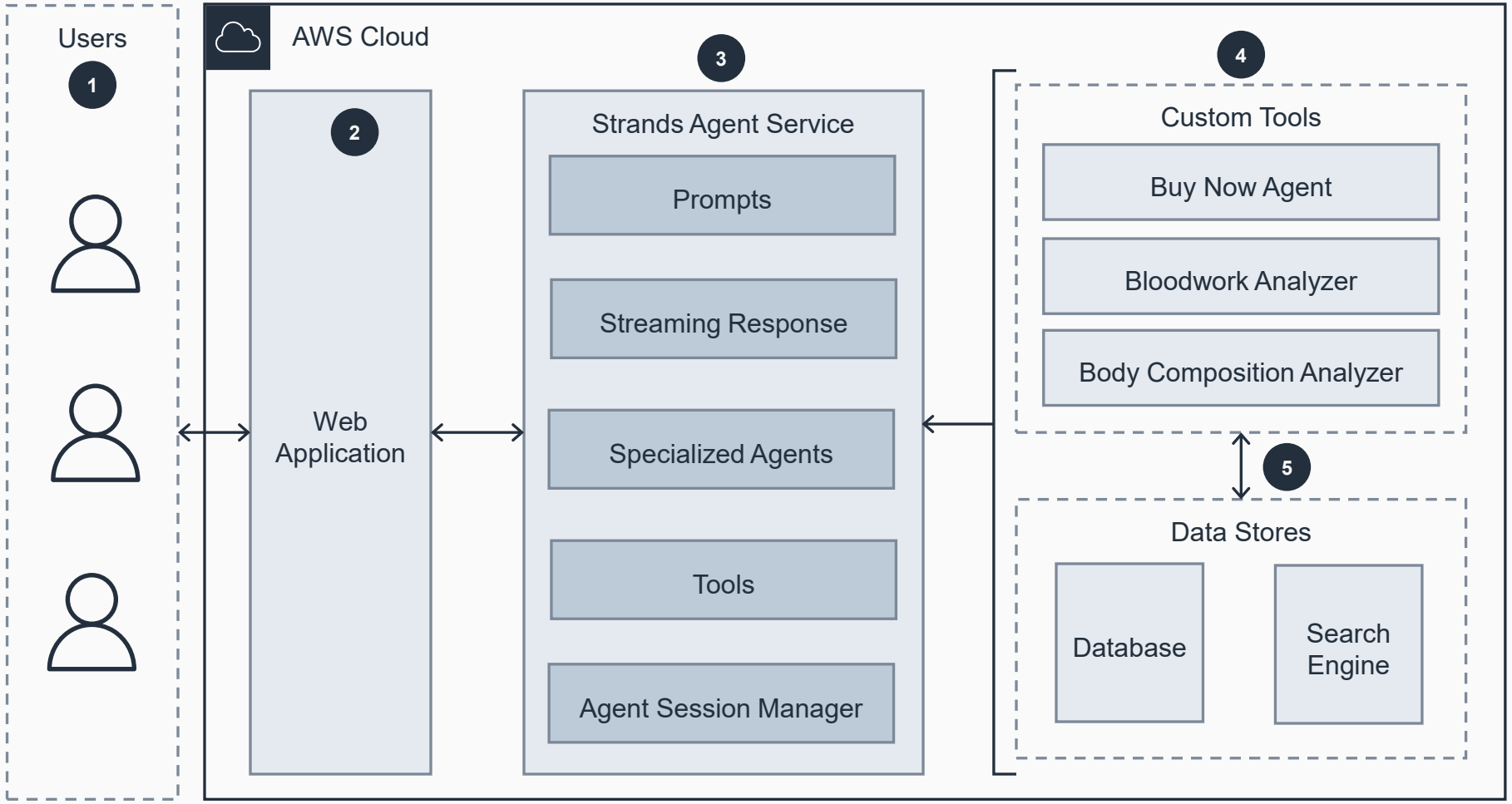
Guidance for Building Agentic AI-Powered Hyper-Personalized Customer Experience on AWS

Solution Overview

This architecture diagram illustrates an AI-powered product recommendation solution that provides personalized product suggestions based on customer profiles, hyper-personal customer data, and intelligent search capabilities.



1. Multiple users access the platform simultaneously, each receiving individualized product recommendations and health insights based on their unique profiles and preferences.
2. Users engage with the system through a web application via token-based authentication.
3. Strands Agents coordinates how various specialized AI agents initialize, cache information, stream responses, and adapt to task context while maintaining user sessions.
4. The custom tools include a Buy Now agent, which can execute product purchases, as well as domain-specific agents. The Bloodwork Analyzer interprets laboratory results, identifies biomarker patterns, and correlates nutritional deficiencies with targeted supplement recommendations, and the Body Composition Analyzer evaluates fitness metrics, tracks body composition trends, and generates personalized nutrition and exercise plans aligned with individual health goals.
5. The data sources layer includes structured databases for profile and catalog management and a search engine with semantic search capabilities that enable intelligent product discovery and contextual retrieval. Data is encrypted at rest and secure access patterns are used for queries.

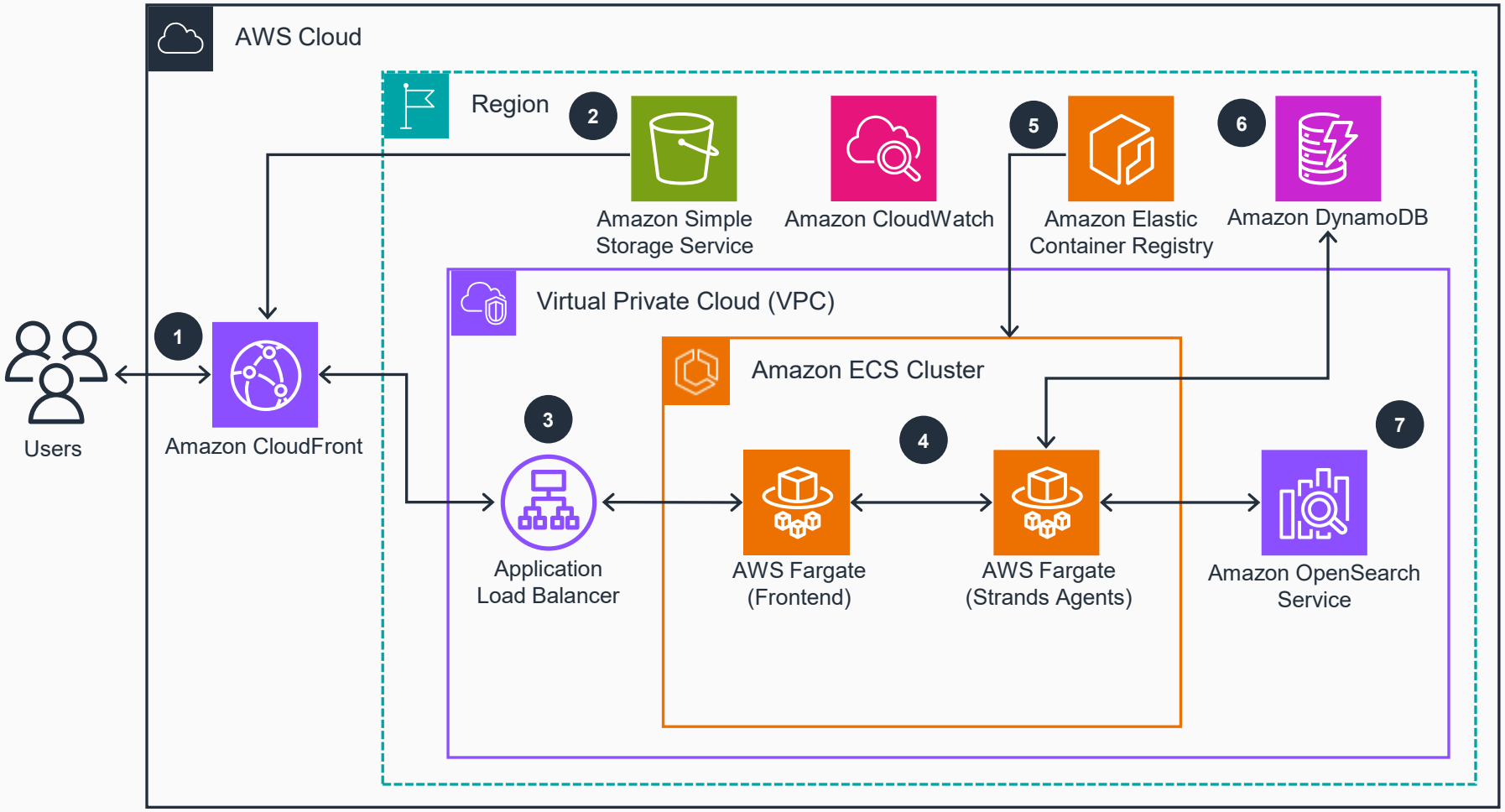
Reviewed for technical accuracy December 18, 2025

AWS Reference Architecture

Guidance for Building Agentic AI-Powered Hyper-Personalized Customer Experience on AWS

Infrastructure Architecture

This architecture diagram illustrates the key infrastructure components for the web application and data sources used to deploy the solution on AWS.



1. Users access the platform through **Amazon CloudFront**, a global content delivery network that provides low-latency access to the application for users worldwide.
2. The web application is served from **Amazon Simple Storage Service (Amazon S3)**, that hosts product images and static assets for the application.
3. The Application Load Balancer distributes incoming traffic across the instances of the frontend and Strands services.
4. **AWS Fargate** runs the serverless frontend and backend containers that provide the web application user interface and backend functionality, enabling product search, chat interaction, and customer profile management.
5. **Amazon Elastic Container Registry (Amazon ECR)** serves as the container registry that stores and manages Docker images.
6. **Amazon DynamoDB** provides the database layer storing customer profiles, product catalog, search history, and application data with low-latency access for near real-time operations. Data is encrypted at rest and secure access patterns are maintained.
7. **Amazon OpenSearch Service** powers the search and analytics engine that enables keyword search, semantic search with embeddings, and intelligent product discovery capabilities for personalized recommendations. Data is encrypted at rest and secure access patterns are maintained.

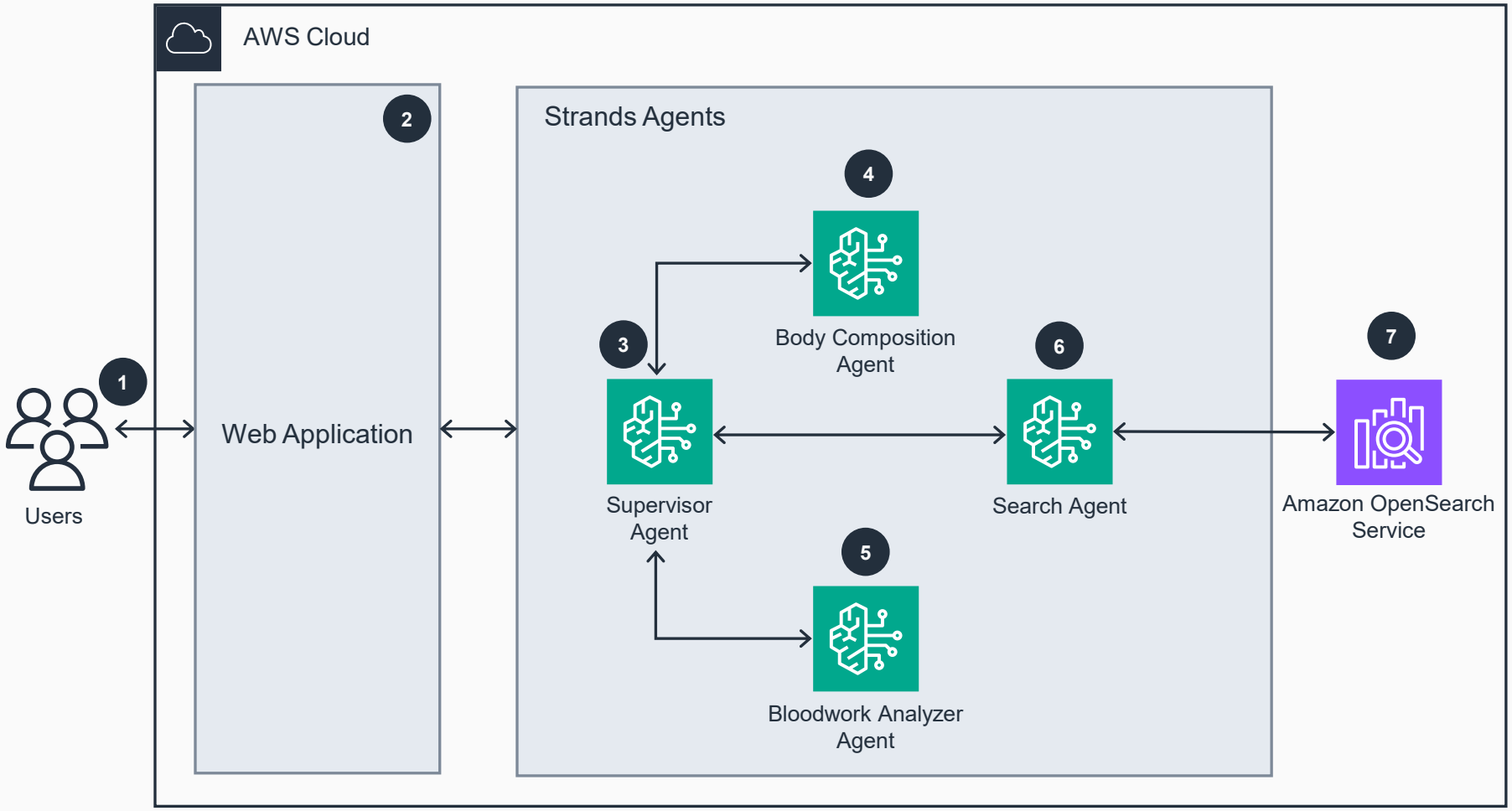
Reviewed for technical accuracy December 18, 2025

AWS Reference Architecture

Guidance for Building Agentic AI-Powered Hyper-Personalized Customer Experience on AWS

Agent Collaboration Framework

This architecture diagram illustrates the Amazon Bedrock multi-agent collaboration using the Strands Agents SDK for the entire customer experience workflow.



1. Users interact submitting health data, product queries, and receiving personalized recommendations based on their unique profiles and preferences.
2. The Strands service coordinates how specialized AI agents initialize, cache information, stream responses, and adapt to both user tasks and context while maintaining user sessions throughout the interaction.
3. The supervisor agent orchestrates the workflow by delegating tasks to specialized agents, coordinating their responses, and synthesizing insights to provide comprehensive personalized recommendations.
4. The body composition agent gets user input and evaluates fitness metrics, tracks body composition trends, and generates personalized nutrition and exercise plans aligned with individual health goals.
5. The bloodwork analyzer agent interprets laboratory results from user input, identifies biomarker patterns, and correlates nutritional deficiencies with targeted supplement recommendations.
6. The search agent queries the product catalog, applies personalized filters based on user preferences and health data, and retrieves relevant products.
7. **Amazon OpenSearch Service** serves as the data layer containing the product catalog with semantic search capabilities that enable intelligent product discovery and contextual retrieval.

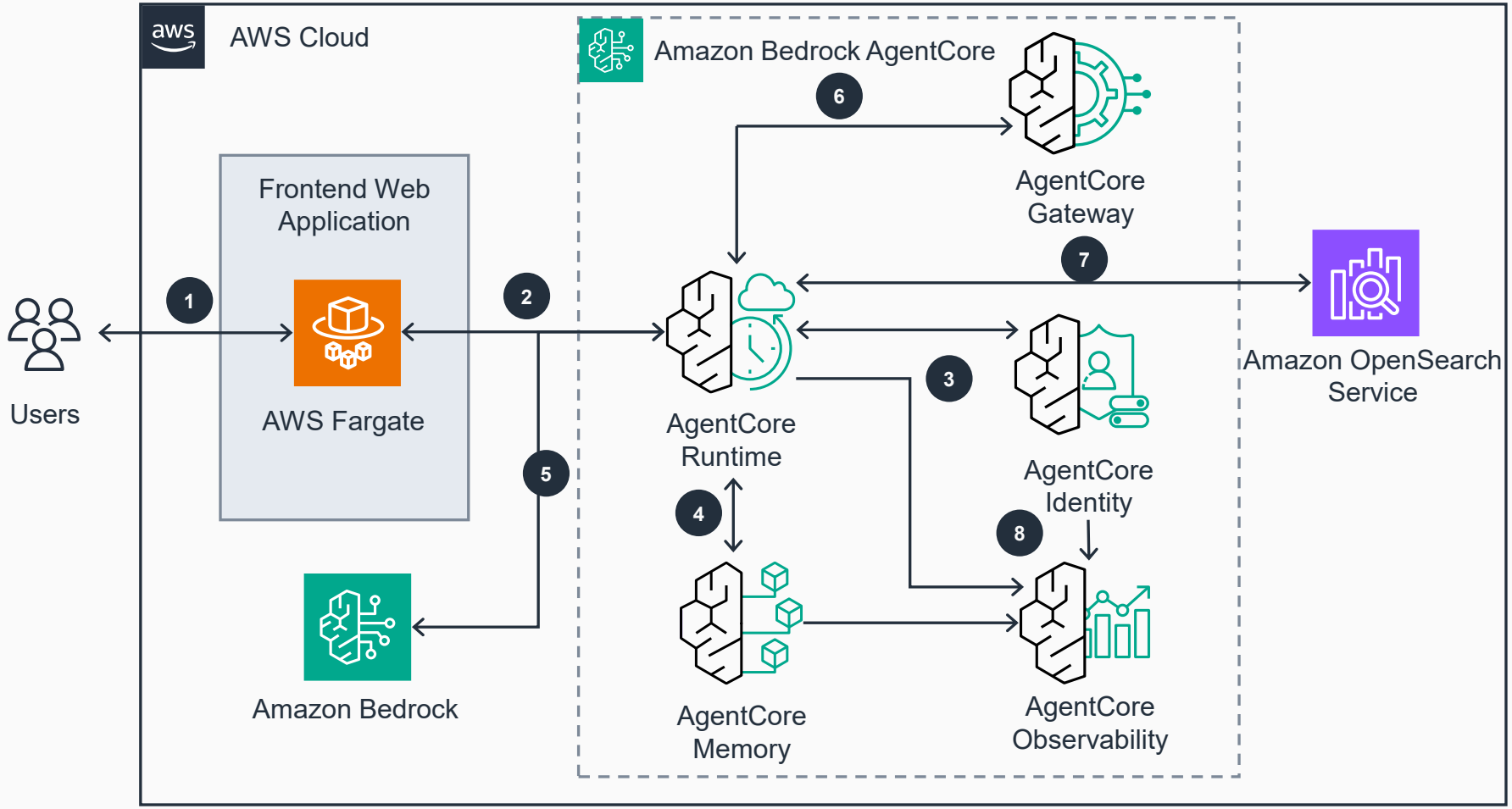
Reviewed for technical accuracy December 18, 2025

AWS Reference Architecture

Guidance for Building Agentic AI-Powered Hyper-Personalized Customer Experience on AWS

AgentCore Implementation

This architecture diagram shows alternative agent hosting on Amazon Bedrock AgentCore. Deploy secure, scalable AI agents on AWS with Amazon Bedrock AgentCore's purpose-built infrastructure, enabling complex workflows across tools and data sources while eliminating infrastructure management overhead.



1. Users interact with the chat interface hosted on **AWS Fargate**.
2. Execute agent code, tools, and instructions in **Amazon Bedrock AgentCore** Runtime's serverless environment, supporting multiple frameworks and 8-hour sessions.
3. Secure agent operations with **AgentCore** Identity, managing authentication and access controls across all interactions.
4. Build context-aware agents with **AgentCore** Memory, maintaining both short-term and long-term knowledge across interactions.
5. Access **Amazon Bedrock** for foundation models, enabling flexible use of various LLMs through a unified API.
6. Transform REST APIs into Model Context Protocol (MCP) servers through **AgentCore** Gateway, enabling reusable tool sharing across agents.
7. Connect to AWS services like **Amazon OpenSearch Service** to enrich context and perform tasks.
8. Monitor agent performance through **AgentCore** Observability, tracking key metrics and ensuring operational excellence.

Reviewed for technical accuracy December 18, 2025

AWS Reference Architecture