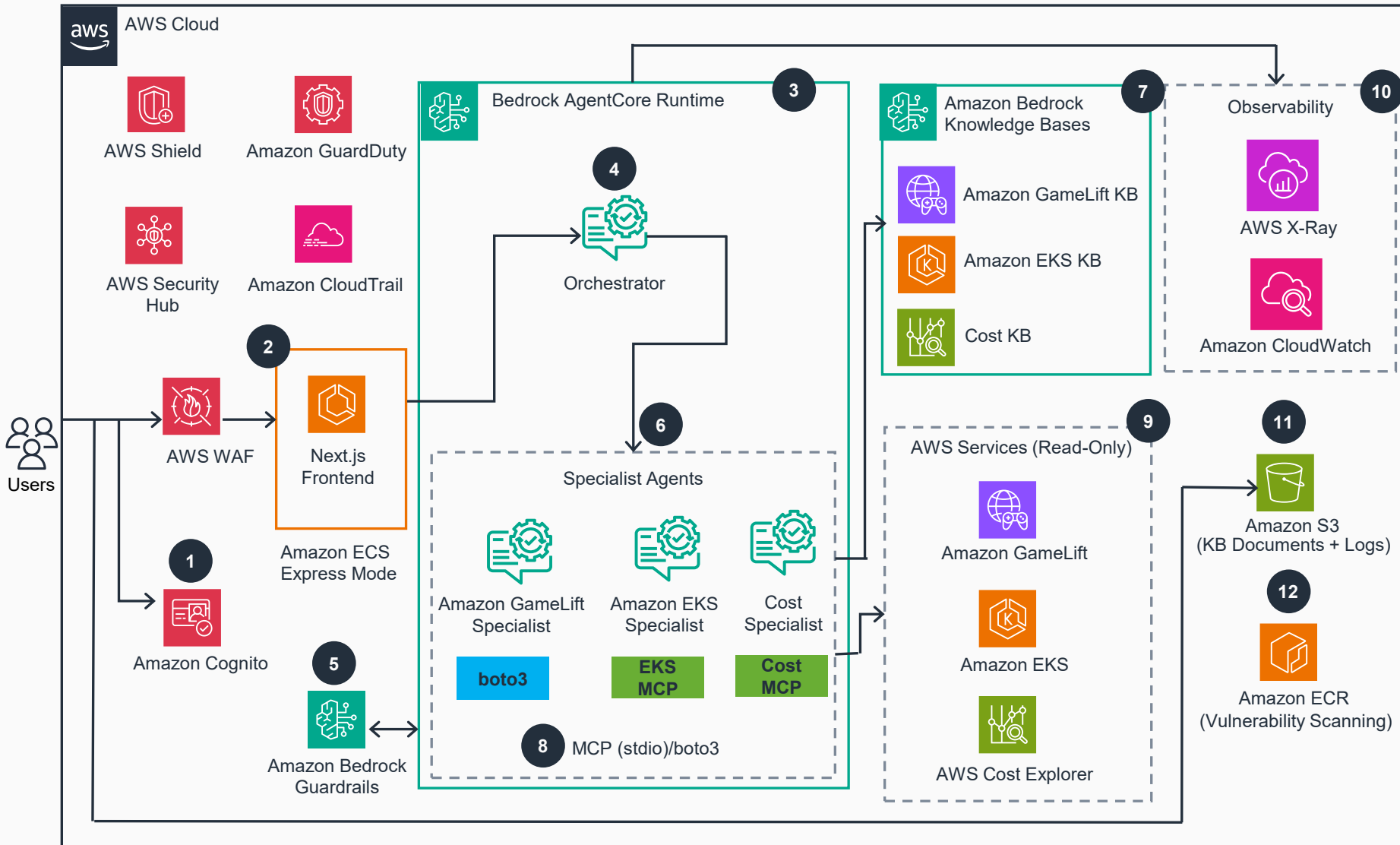


Guidance for Game Backend and Infrastructure Agentic Workflows

This architecture diagram shows how to build specialized AI agents with Amazon Bedrock AgentCore for conversational management of game infrastructure running on Amazon GameLift and Amazon EKS infrastructure.



- 1 Authenticate with **Amazon Cognito** User Pool. JWT tokens stored in HttpOnly cookies.
- 2 Send query through Next.js frontend on **Amazon Elastic Container (Amazon ECS)** Express Mode. CopilotKit provides the chat UI.
- 3 Invoke **Amazon Bedrock AgentCore** Runtime via **AWS SDK** with SigV4 authentication.
- 4 Route to Orchestrator agent which classifies query intent.
- 5 **Amazon Bedrock** Guardrails filter input/output. Detects prompt injection, blocks PII exposure.
- 6 Delegate to Specialist Agent (**Amazon GameLift**, **Amazon Elastic Kubernetes Service (EKS)**, or Cost) based on query.
- 7 Query **Amazon Bedrock Knowledge Base** for domain context. RAG with Titan embeddings.
- 8 Invoke MCP Servers (**Amazon EKS**, Cost) or boto3 (**Amazon GameLift**) for AWS API integration.
- 9 Read-only API calls to AWS Services: **Amazon GameLift** fleets, **Amazon EKS** clusters, AWS Cost Explorer.
- 10 **Amazon CloudWatch** Logs and X-Ray traces capture observability data.
- 11 KB documents stored in **Amazon S3** with encryption and access logging.
- 12 Container images in **Amazon Elastic Container Repository (ECR)** with vulnerability scanning enabled.

