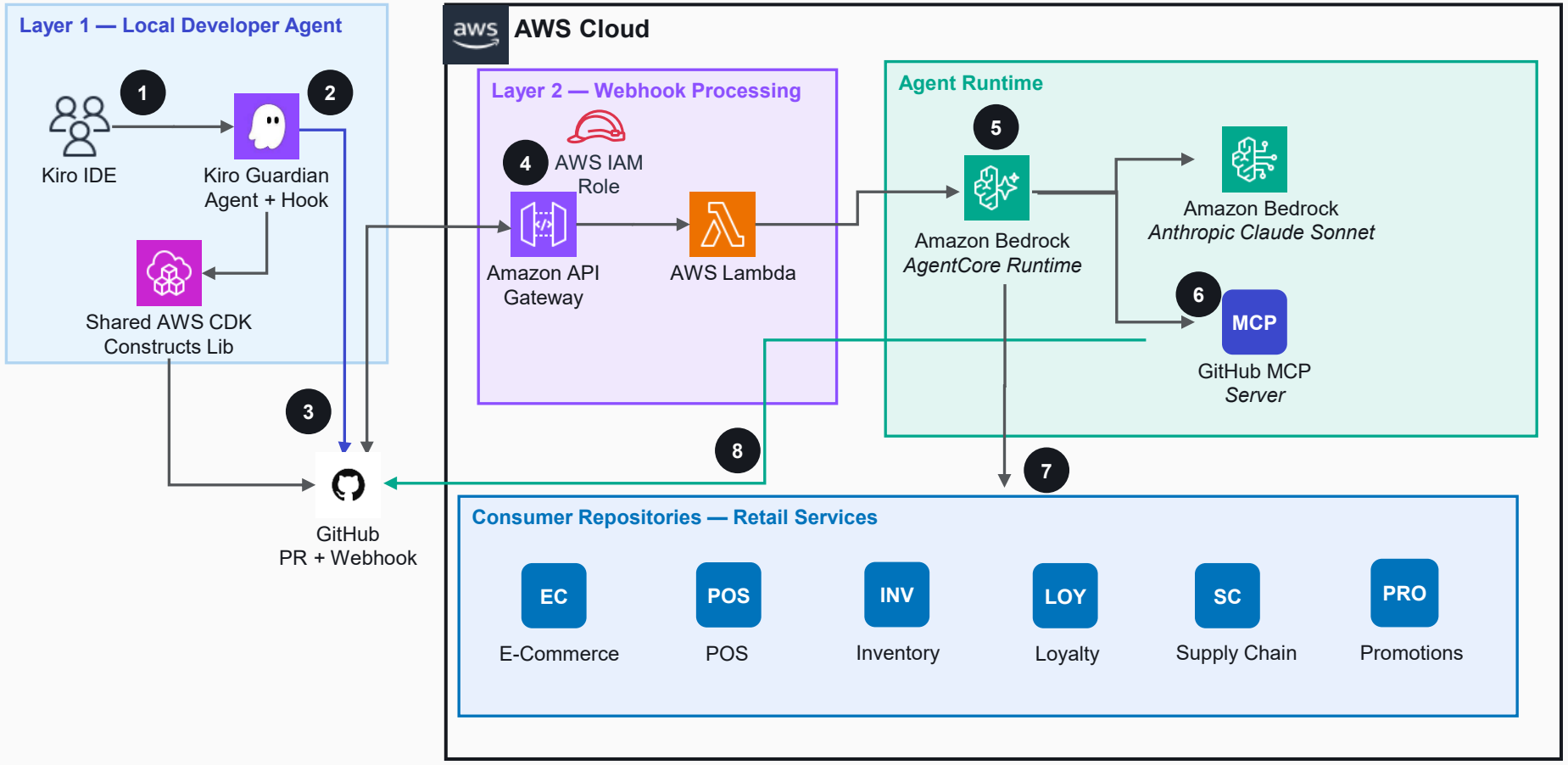
Guidance for CDK Construct Compatibility Testing using Amazon Bedrock AgentCore

This architecture diagram shows how two AI agent layers protect shared CDK construct libraries from breaking downstream retail services. Amazon CloudWatch and AWS X-Ray provide observability across the pipeline.



1. A platform engineer edits a shared **AWS Cloud Development Kit (CDK)** construct in Kiro IDE. The hook triggers the local guardian agent on file save.
2. The local agent authenticates to GitHub via a scoped credential, validates the construct against standards, and searches for consumer repos. It surfaces blast radius feedback to the developer in Kiro IDE.
3. The developer opens a pull request on the shared constructs library hosted on GitHub.
4. GitHub sends a webhook to **Amazon API Gateway**, secured by a resource policy scoped to GitHub's IP ranges. **AWS Lambda** verifies the HMAC signature and assumes a least-privilege **AWS Identify and Access Management (IAM)** role.
5. **AWS Lambda** invokes the Strands agent on **Amazon Bedrock AgentCore** Runtime—a managed execution environment for AI agents. The agent calls **Amazon Bedrock** for LLM inference via Anthropic's Claude Sonnet.
6. The agent discovers all consumer repos (e-commerce, POS, inventory, loyalty, supply chain, promotions) via the GitHub MCP Server—an externally hosted MCP-compatible interface to the GitHub API.
7. The agent clones each consumer repository via the GitHub MCP Server, resolves the shared construct to the PR branch version, and runs **AWS CDK** synth and tests to detect breaking changes.
8. The agent posts a structured review comment on the GitHub PR via the GitHub MCP Server, summarizing breaking changes and blast radius impact across all retail services.

Reviewed for technical accuracy July 21, 2026

AWS Reference Architecture