AWS State, Local, and Education Learning Days

Salt Lake City, UT

3:15pm – 4:15pm

300 level

Designing modern applications in AWS

Unlock Serverless Potential: Reduce Costs, Boost Scalability, and Enhance Security with Cloud-Native Architectures.

aws Learning Days
State, Local, and Education

Designing modern applications in AWS

Cade McClellan (he/him)

Solutions Architect
Amazon Web Services
cademc@amazon.com

Ekta Vasa (she/her)

Senior Solutions Architect
Amazon Web Services
ekvasa@amazon.com

aws Learning Days
State, Local, and Education

Voice of the Customer

- What's your biggest challenge when modernizing legacy applications?
- What keeps you up at night about your current application architecture?
- How many of you have experience with Infrastructure as Code?

Architecting on AWS is different

Legacy	Modern (AWS)
Monolith (app + DB tightly coupled)	Modular (microservices, event, SOA) architectures
Vertical scaling (big VMs)	Automated scaling
One language/stack	Polyglot, fit-for-purpose technology
Manual deployments	Automated pipelines (CI/CD)
Fragile updates	Independent service delivery
Owned infrastructure	Managed services, focus on delivery business value

Architecting on AWS is different

- It’s not just about stringing together services, but about building scalable, elastic, resilient, secure, reliable and cost-efficient solutions using managed cloud-native services
- Emphasis on composable architectures of distributed, modular & reusable components; generally, service-oriented
- Infrastructure-as-code: infrastructure as cattle, not pets
- Expanded integration and orchestration pathways (asynchronous, event-driven, “everything-as-an-API” decoupling)

Principles of Modern Cloud System Architecture

Systems

Build flexibly for the future with loosely coupled services and component-based architecture

Require auto-scaling and load balancing

Use purpose-built services

Govern architecture across the enterprise

Ensure performance and skill alignment

Offer seamless storage functionality

Decouple infrastructure & experience

Work backwards from business needs

Leverage automation and containers

Principles of Modern Cloud System Architecture

Experience

Deliver holistically with user-centered design, accessibility, and reusable components

Require scalable public and worker interfaces

Segment and personalize

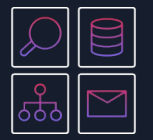
Develop a unified design approach

Ensure performance and skill alignment

Use seamless data to make decisions

Decouple data and channels

Work backwards from user needs



Leverage reusable components

What is the AWS Well-Architected Framework?

Pillars & Lenses

Design principles

Questions

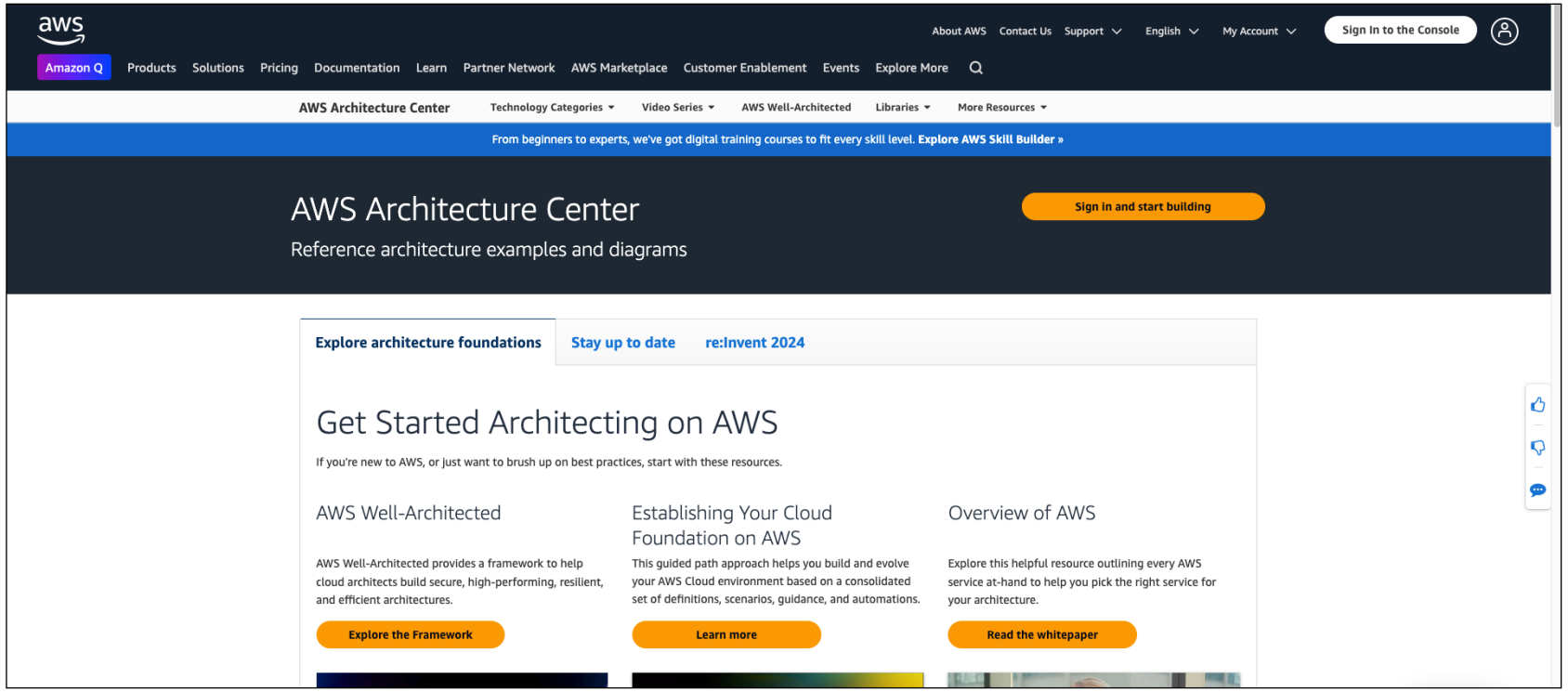
Best Practices

Pillars of the AWS Well-Architected Framework

AWS Architecture Center

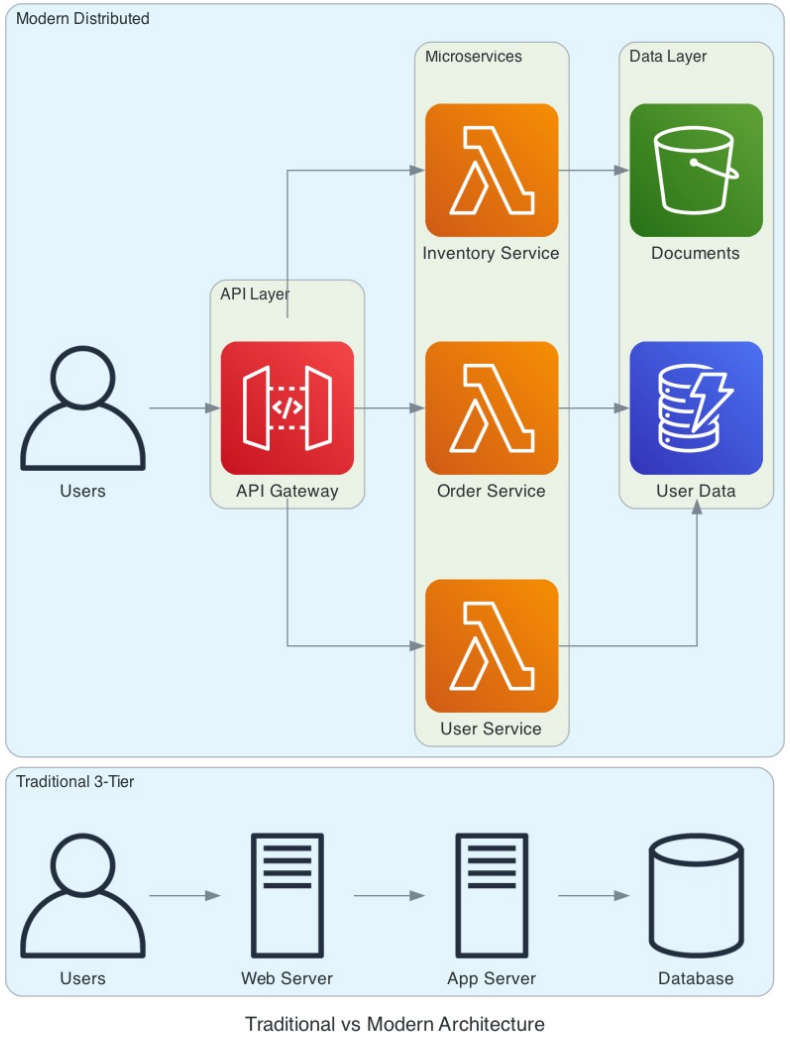
- Library of content including
  - Patterns
  - Reference Architectures
  - Guidance
  - Solutions, and more
- Links and other resources for architecting on AWS
- Video Series like ‘This is My Architecture’ and ‘How to Build This’
- Architecture Best Practices

https://aws.amazon.com/architecture



Let’s get out the chalk

Traditional vs Modern

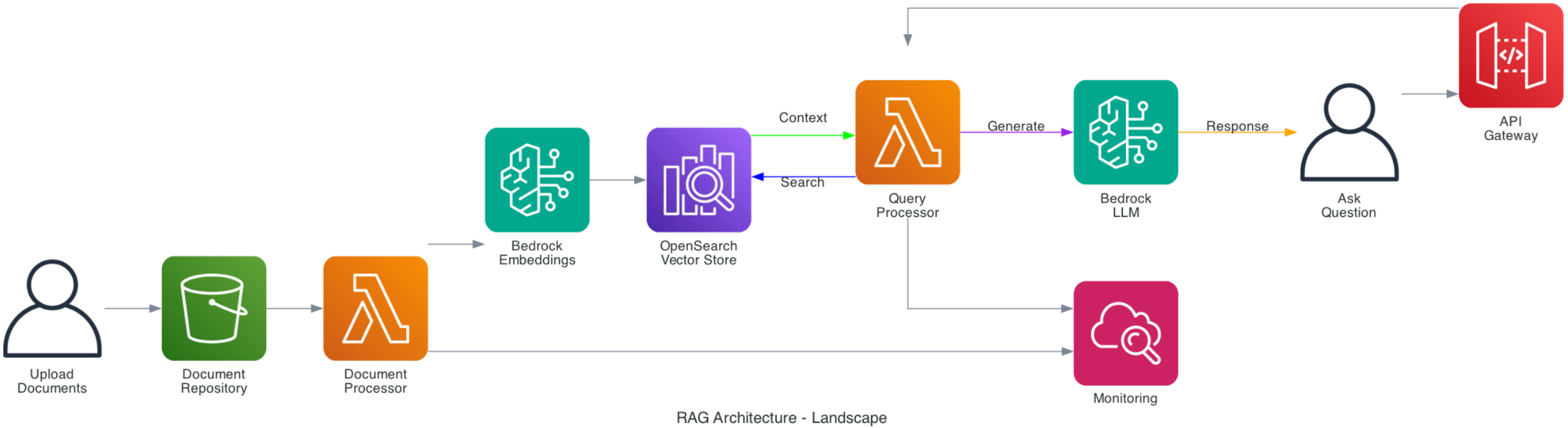


Traditional vs Modern Architecture

Modern Web App

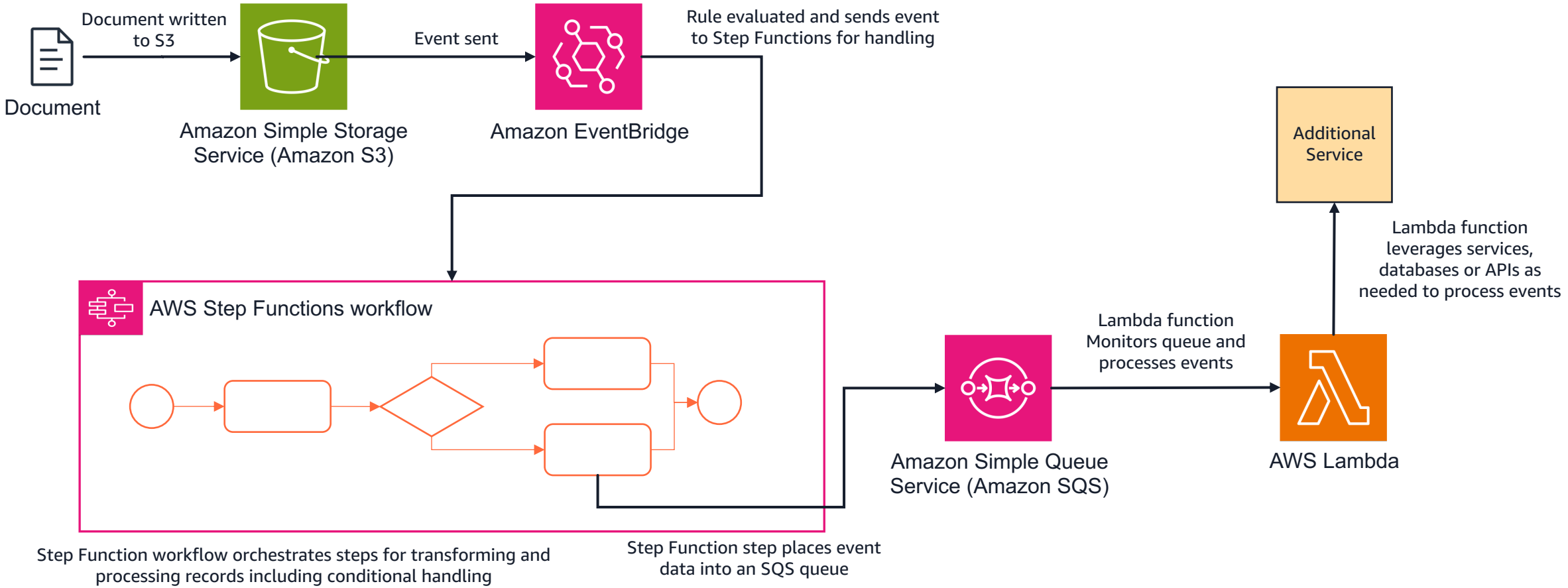
Modern Web Application - Step by Step Build

Generative AI RAG Architecture



RAG Architecture - Landscape

Event-driven processing



Thank you!

Cade McClellan (he/him)

Solutions Architect
Amazon Web Services
cademc@amazon.com

Ekta Vasa (she/her)

Senior Solutions Architect
Amazon Web Services
lekvasa@amazon.com

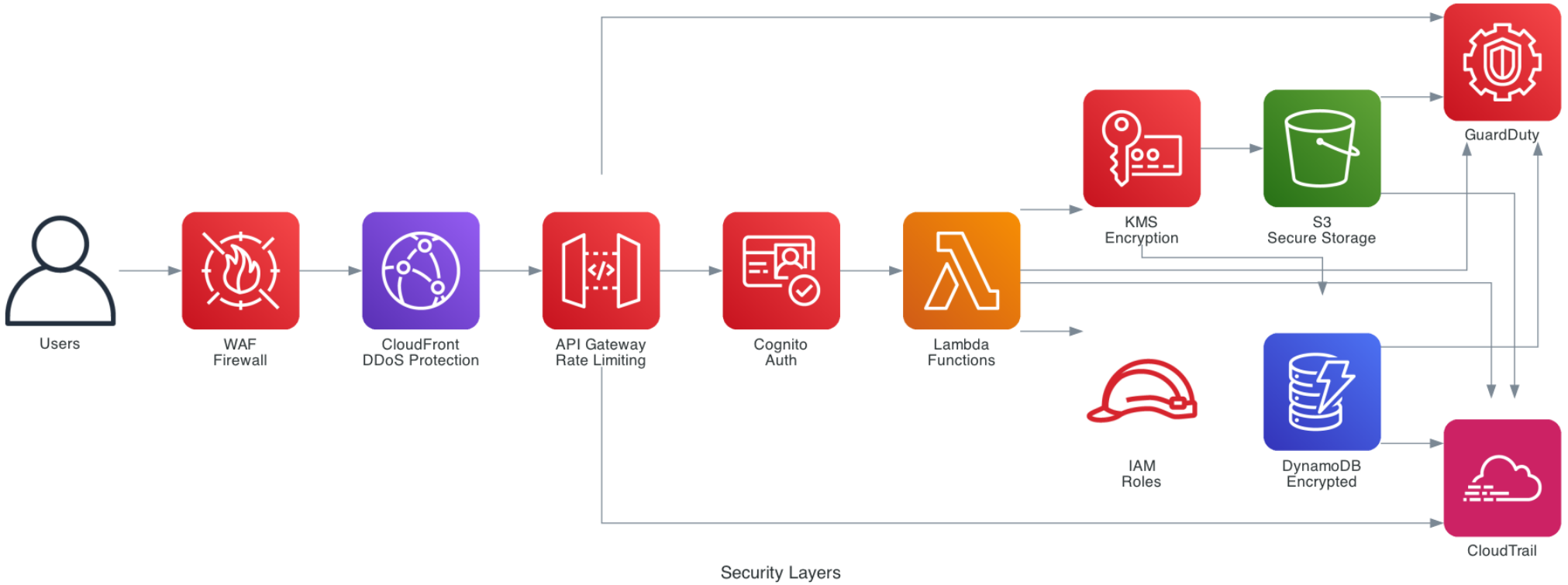
Please complete the survey for this session

Track: Cloud Fundamentals

Session: Designing modern applications in AWS

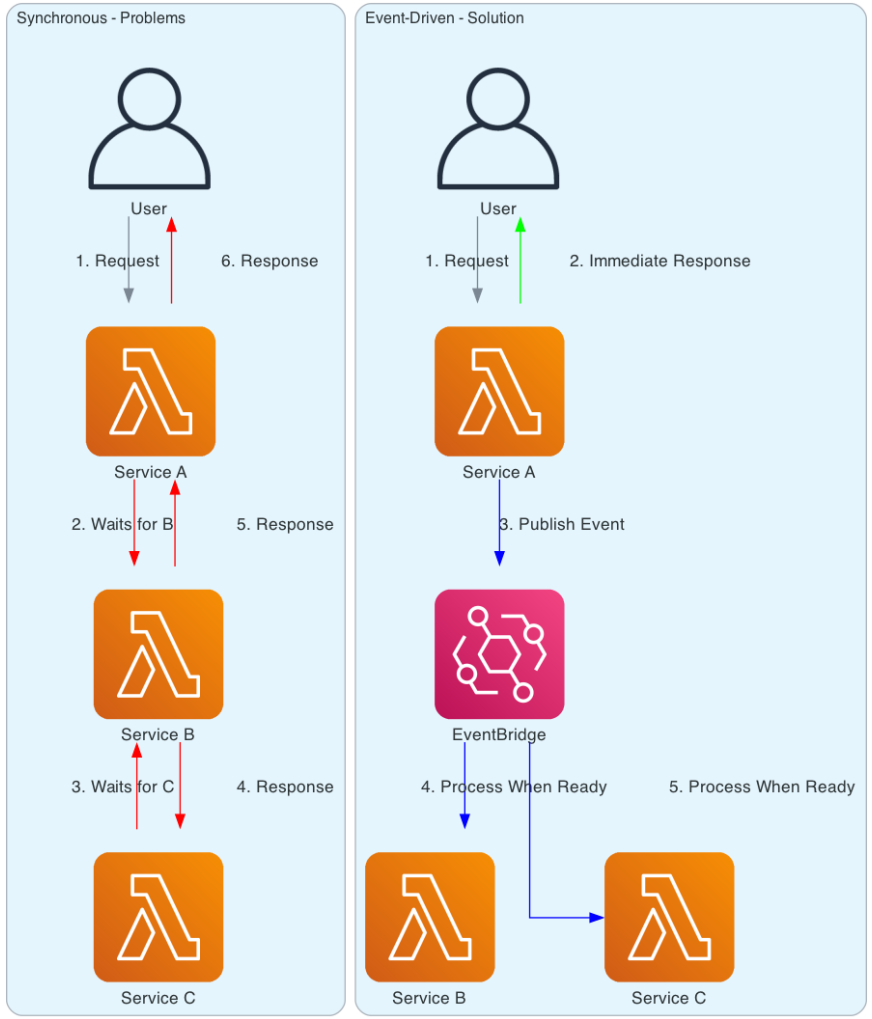
aws Learning Days
State, Local, and Education

Security Architecture



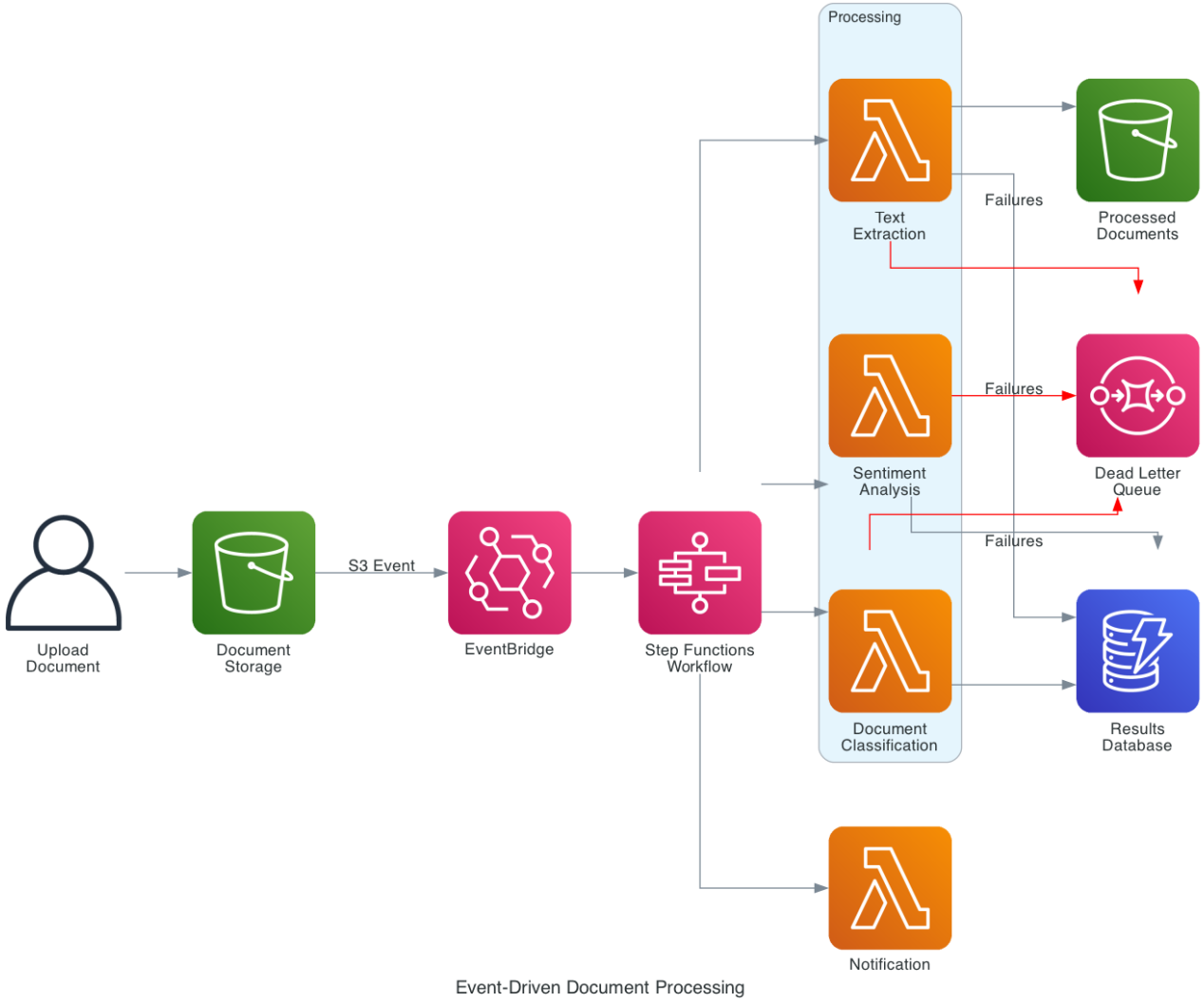
Appendix

Synchronous vs Event Driven

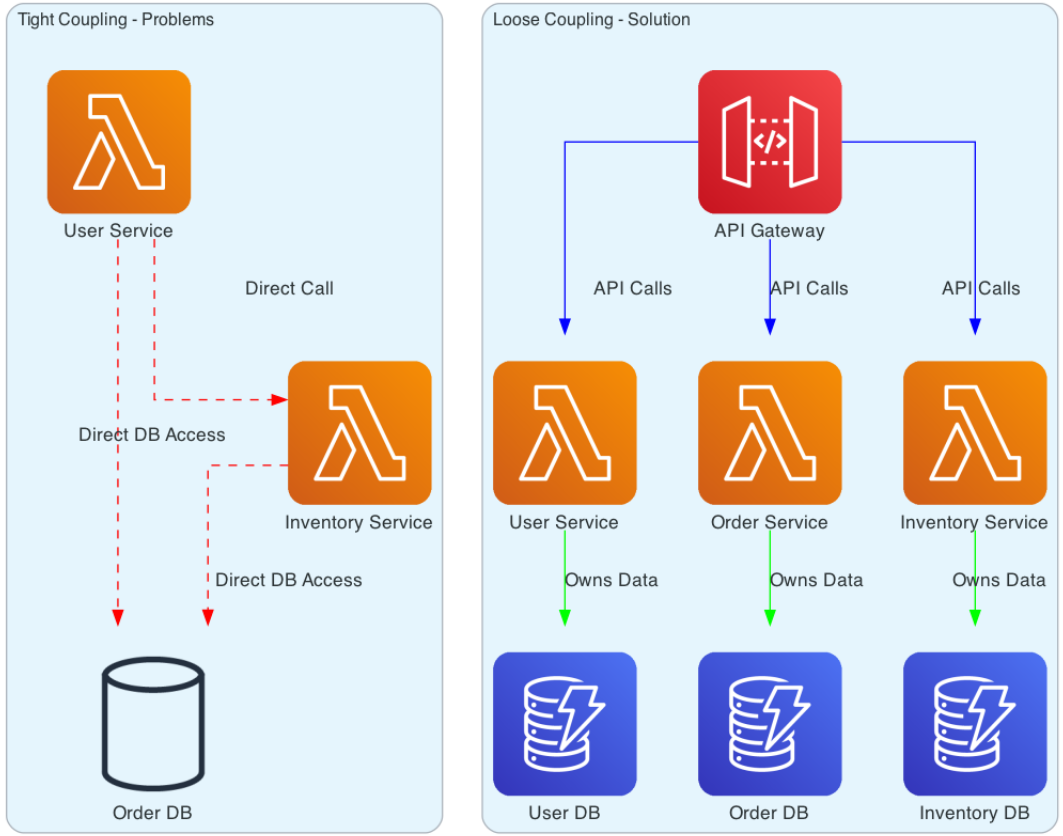


Event-Driven vs Synchronous

Chalk #8 Event Driven Document

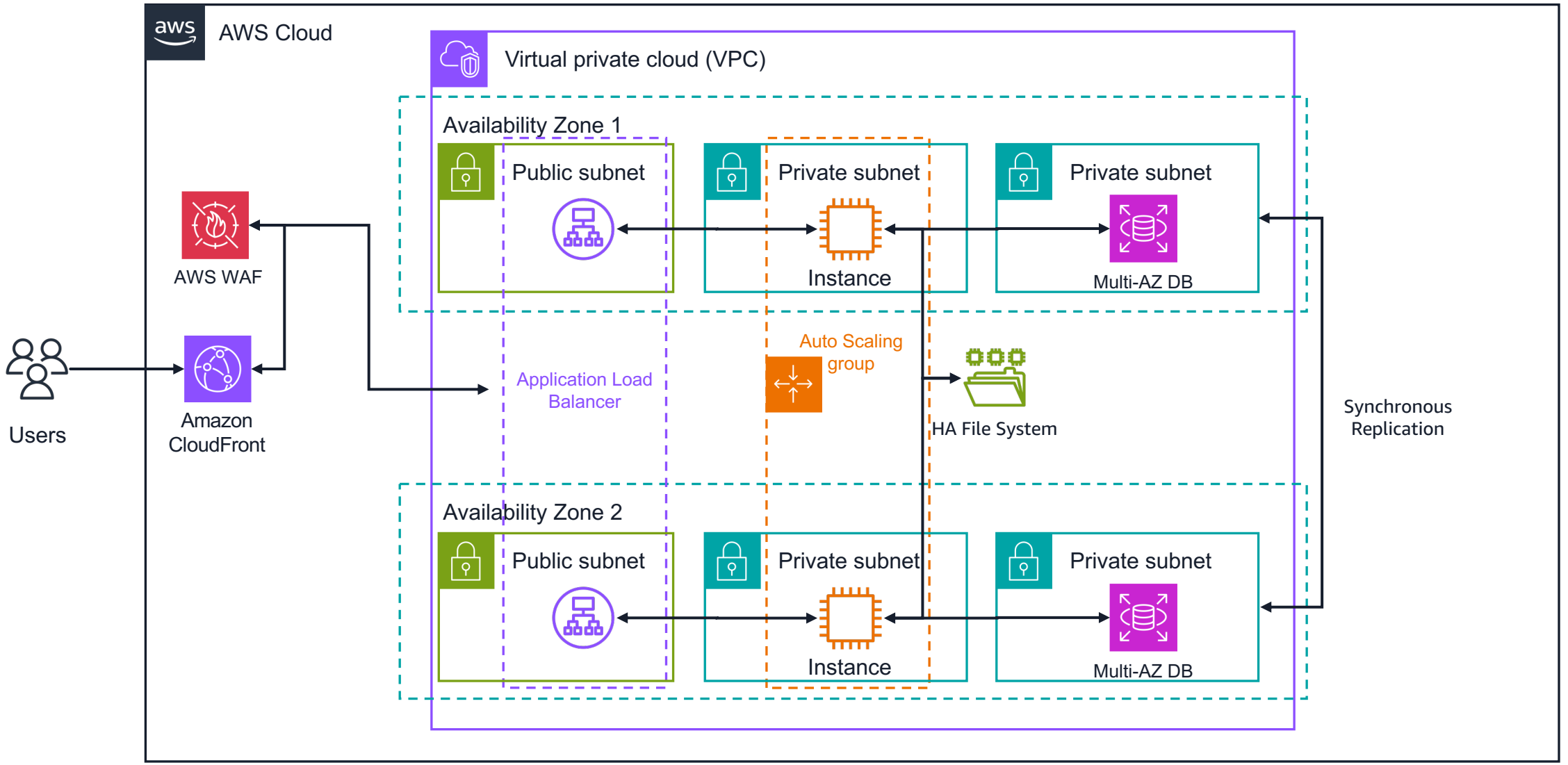


Chalk #6 Loose Coupling

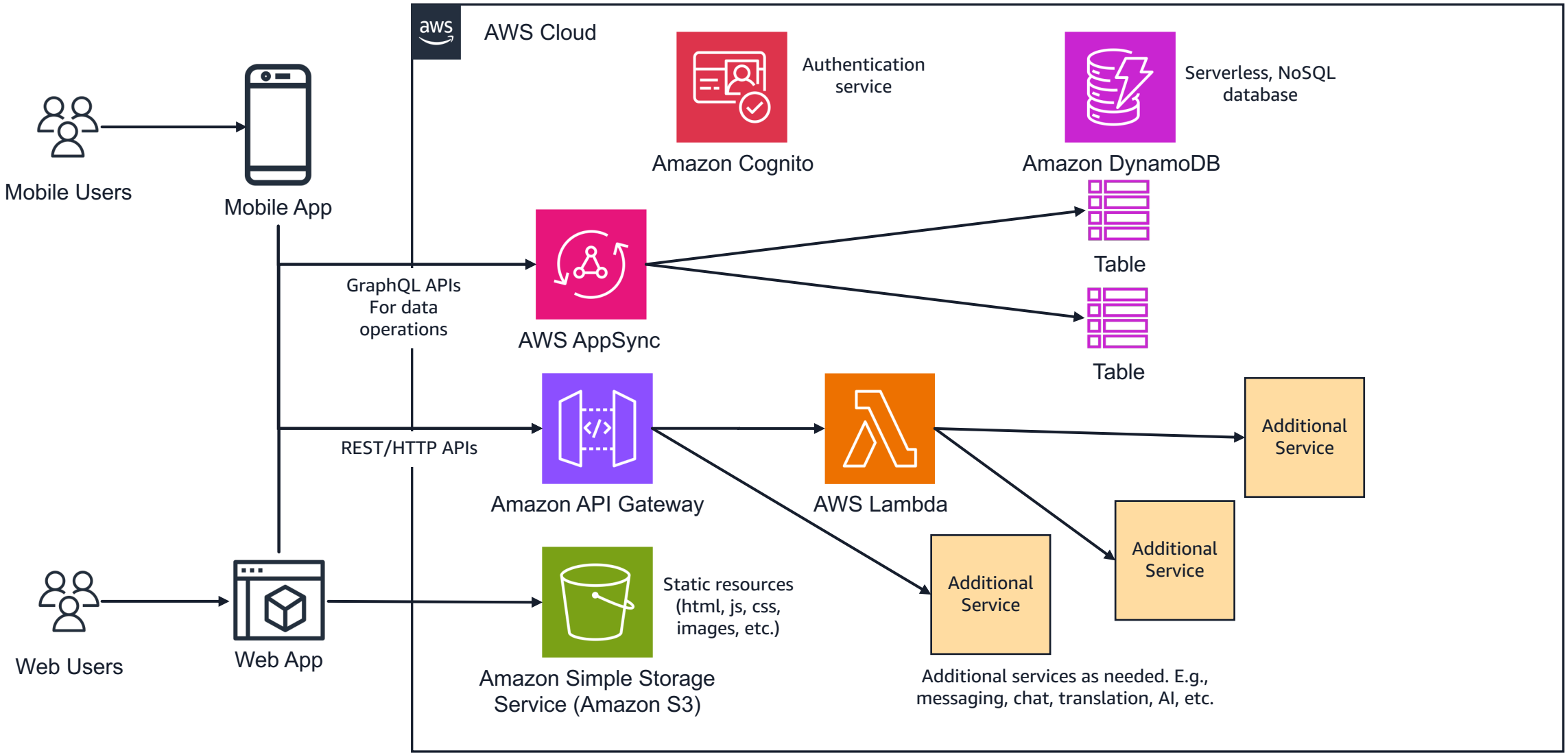


Loose Coupling - Before and After

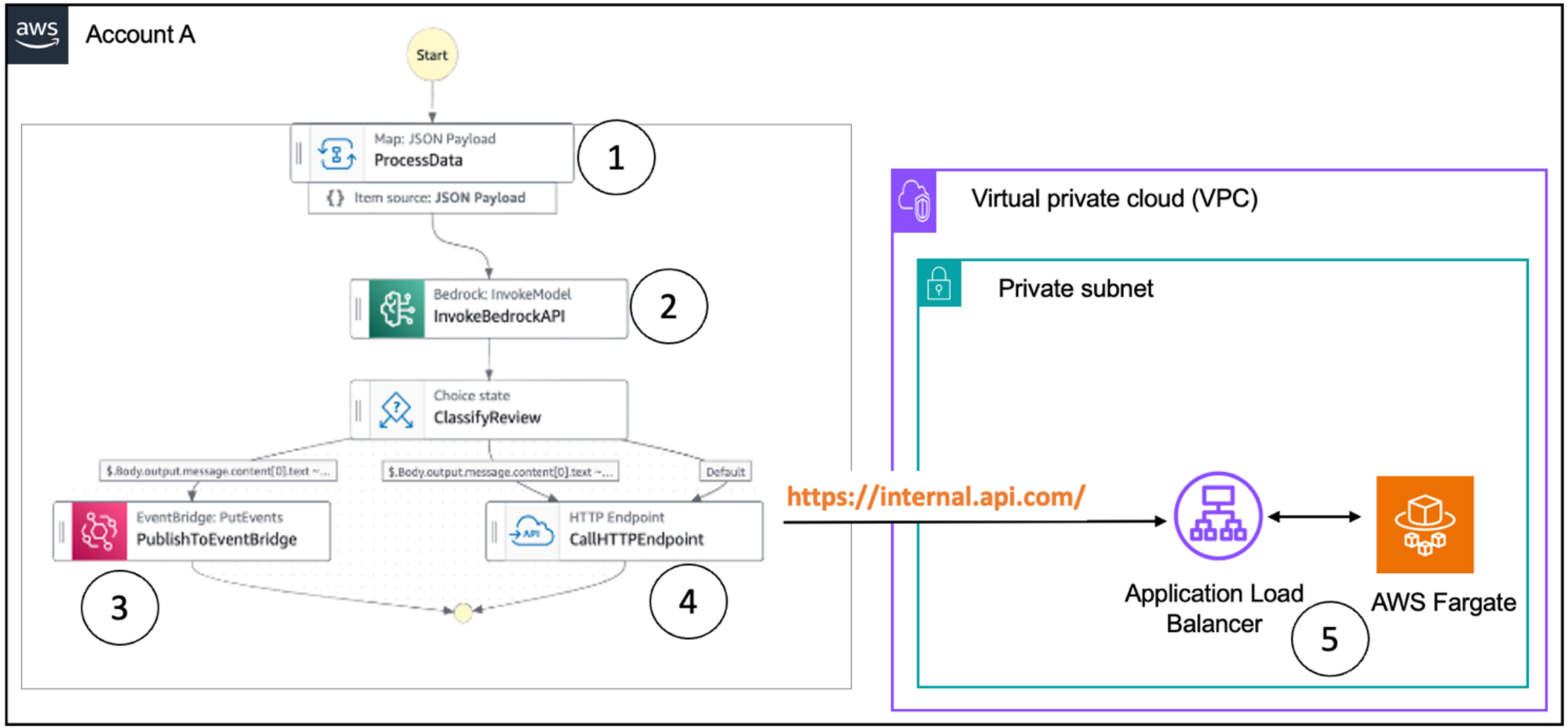
Highly Available 3-Tier Application



Serverless Web/Mobile Application with APIs

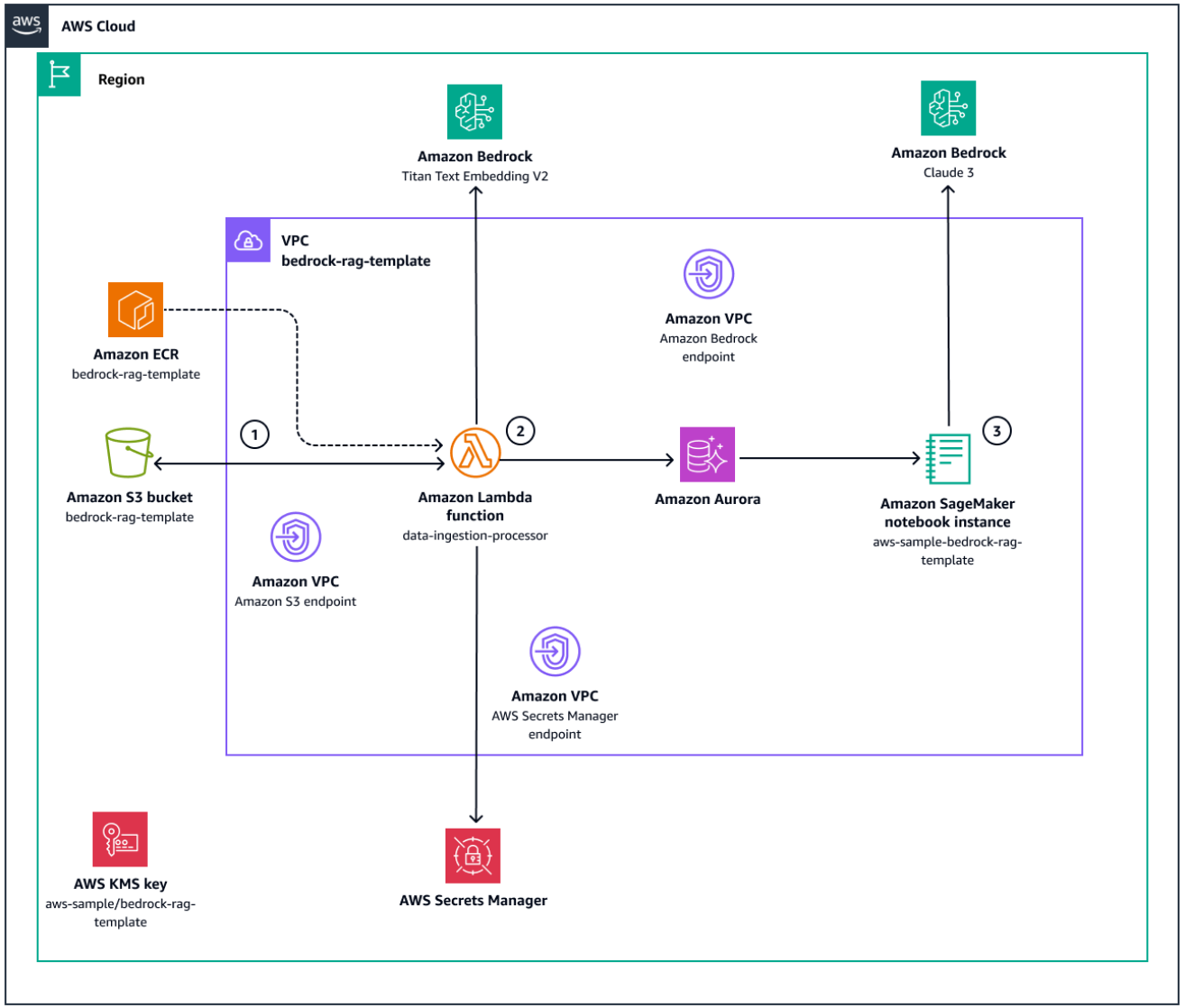


Orchestrated Asynchronous Processing



Source: https://aws.amazon.com/blogs/compute/simplifying-private-api-integrations-with-amazon-eventbridge-and-aws-step-functions-2/

Generative AI RAG Architecture



Source: https://docs.aws.amazon.com/prescriptive-guidance/latest/patterns/deploy-rag-use-case-on-aws.html