**🎯 Interview Focus: Architecture Depth + Strategic Thinking**

**Duration**: ~45 minutes  
**Candidate**: Amarnath (25+ years of experience, AWS-certified, heavy enterprise cloud & modernization background)  
**Goal**: Evaluate thinking process, ability to navigate ambiguity, and how experience maps to SuperApp’s needs.

**1. Architecting at Scale in Regulated Industries**

**Question**: You've led major architecture efforts in banking and pharma. How do you approach building a scalable, secure, and compliant service architecture in a highly regulated environment like healthcare?

* ✅ Good Answer: Discusses layered security, HIPAA-style controls, shared responsibility model, auditing, architecture patterns like zero-trust, and compliance gates in CI/CD.
* ❌ Bad Answer: Focuses only on tech stacks (e.g., "use AWS and encrypt everything") without acknowledging regulatory nuance or architectural enforcement.
* 🔁 Follow-ups:
  + How do you adapt this for teams not trained in regulatory domains?
  + What’s your strategy to balance velocity with compliance?

🧩 Alignment: Direct overlap with SuperApp HIPAA/ADA/API concerns and experience in financial and pharma domains.

**2. Multi-Org API Contract Negotiation**

**Question**: When integrating APIs between two large organizations like Aetna and CVS, how do you ensure agreement on API contracts and data semantics?

* ✅ Good Answer: Mentions anti-corruption layers, shared domain models, canonical schemas, experience APIs, or governance bodies.
* ❌ Bad Answer: Says "we define the API" or ignores cross-org negotiation entirely.
* 🔁 Follow-ups:
  + When would you accept schema divergence?
  + How would you avoid over-engineering a shared contract?

🧩 Alignment: Amarnath’s work in cross-entity consolidation (pharma, banking) and API ecosystem governance at CITI is highly relevant.

**3. Experience APIs vs. GraphQL**

**Question**: For a complex, consumer-facing app like SuperApp, when would you use an experience API layer instead of GraphQL?

* ✅ Good Answer: Weighs operational simplicity, caching, synthetic object complexity, and access control granularity.
* ❌ Bad Answer: Says “REST is simpler” or “GraphQL is better for everything.”
* 🔁 Follow-ups:
  + Would you use GraphQL for pharmacy search? Why/why not?
  + What performance pitfalls have you hit in federated models?

🧩 Alignment: He’s delivered multi-channel platforms at Citi and Accenture—this is a natural decision space for him.

**4. Handling Mid-Project Architectural Change**

**Question**: Can you describe a time when a major architectural direction shifted mid-project? How did you navigate that?

* ✅ Good Answer: Discusses stakeholder alignment, technical debt trade-offs, and iterative re-architecture.
* ❌ Bad Answer: Blames “the business,” says “we just redid it,” or avoids real conflict resolution.
* 🔁 Follow-ups:
  + How would you avoid such shifts upfront?
  + When is starting over justified?

🧩 Alignment: Amarnath has “rescued” multiple stalled projects. Looking for maturity in handling design pivots.

**5. API Aggregation Strategy**

**Question**: If you're aggregating multiple backend services with different SLAs and auth methods, what’s your architectural strategy?

* ✅ Good Answer: Circuit breakers, retries, token exchange, proxy layers, API Gateway or service mesh.
* ❌ Bad Answer: “Just call each service from the frontend” or no plan for error isolation.
* 🔁 Follow-ups:
  + What latency optimization techniques would you use?
  + When would you use GraphQL federation instead?

🧩 Alignment: Leverage his microservices and SOA experience (TCS, Accenture, Citi).

**6. SuperApp-Specific Latency Reduction**

**Question**: SuperApp requires chaining multiple APIs. What’s your approach to minimizing latency and handling failures?

* ✅ Good Answer: Fan-out/fan-in discussion, async via SNS/SQS/Kinesis, resilience design, partial fallbacks.
* ❌ Bad Answer: Only mentions retries or vague “optimize the code.”
* 🔁 Follow-ups:
  + When is fan-in too expensive?
  + How would you decouple identity verification in chained flows?

🧩 Alignment: This is close to his Pega + ML service orchestration and BPM tooling experience at Citi and Fed Reserve.

**7. CI/CD Enforcement of Architecture Principles**

**Question**: How would you enforce architectural principles like security, DDD, and compliance in a multi-team CI/CD pipeline?

* ✅ Good Answer: OPA, static analysis, shared IaC templates, SAST/DAST, GitOps practices.
* ❌ Bad Answer: “Train everyone,” “have a checklist,” or purely manual reviews.
* 🔁 Follow-ups:
  + What has failed for you in this space?
  + Would you centralize or federate enforcement?

🧩 Alignment: Directly relevant to his CI/CD enforcement at Citi, and portfolio rationalization efforts at TCS.

**8. Designing for Elastic Spikes**

**Question**: During open enrollment, how would you architect a SuperApp feature like Benefits Lookup to scale elastically?

* ✅ Good Answer: Mentions serverless (Lambda/Dynamo), queue buffering, auto-scaling groups, caching tiers, observability.
* ❌ Bad Answer: Just says “scale up EC2” or misses burst-read/cold-start issues.
* 🔁 Follow-ups:
  + What are Lambda’s limits in this case?
  + How would you layer cache vs. real-time data?

🧩 Alignment: His cloud migration work for 3000 apps and performance tuning under load makes this a strong fit.

**9. Inter-Org Governance Strategy**

**Question**: How would you create and maintain architectural coherence across CVS and Aetna teams building the same SuperApp?

* ✅ Good Answer: Governance councils, shared architecture runway, contract-first APIs, architecture playbooks.
* ❌ Bad Answer: “We make decisions together” without explaining process or governance.
* 🔁 Follow-ups:
  + How would you measure architectural drift?
  + When would you allow local deviations?

🧩 Alignment: Relevant to his experience turning around cross-vendor projects (pharma, energy, Accenture).

**10. Mentorship & Technical Influence**

**Question**: You’ve mentored architects into “thought leaders.” How do you mentor effectively while still delivering complex projects?

* ✅ Good Answer: Gives examples of coaching with context, modeling design reviews, and setting architecture strategy frameworks.
* ❌ Bad Answer: Just says “I give them freedom” or “I delegate.”
* 🔁 Follow-ups:
  + How do you help mid-level engineers think like system architects?
  + What anti-patterns do you see in technical mentorship?

🧩 Alignment: Aligns with SuperApp’s expectation for technical influence and his ongoing coaching role at Citi.