Staff Software Engineer

Behavioral Interview - Mock Session

Candidate:	Luv Saxena
Position:	Staff Software Engineer
Interview Date:	February 08, 2026
Interview Type:	Behavioral/Leadership
Overall Rating:	8.5/10

Question 1: Cross-Functional Alignment

Question: Tell me about a time when you had to align multiple teams or stakeholders who had conflicting priorities or opinions. How did you handle it?

Your Response:

> **Project Context: Migration of legacy monolith payment system at Flipkart to platform-based microservices architecture**

Challenge: Multiple upstream teams (warehouse, OMS, seller systems) had their own roadmaps and were resistant to migration work. Legacy flows had been running for years with synchronous calls and unclear dependencies. Teams questioned why they needed to change.

Your Approach:

• **Early engagement:** Started pitching the migration idea much earlier to give teams time to plan

• **Clear justification:** Explained technical limitations - monolith couldn't scale BBD (Big Billion Days) load, couldn't support new business requirements (single order/multiple sellers, multipart shipments)

• **Phased migration strategy:** Assured minimal changes and zero impact to current flows by running both systems in parallel during transition

• **Built confidence through data:** Ran AD (Application Driven) mode testing to compare responses between old and new systems

• **Started with smaller warehouses:** Migrated low-volume warehouses first (500-1000 shipments/day) to prove the approach, then scaled to larger ones

Critical Incident:

During the first warehouse migration, an anomaly appeared in dashboards within hours. Warehouse team immediately lost trust and requested rollback. You were under pressure near BBD launch timeline. You quickly identified and fixed the issue (missing data in migration - not all orders were migrated). After the fix, graphs showed linear performance with no anomalies. Warehouse team regained confidence, and subsequent migrations were much smoother.

Outcome:

✓ Successfully migrated all warehouse and seller systems to new microservices platform

✓ Zero downtime during migration

✓ System ready to handle 4x BBD load with 99.99% uptime SLA

✓ Platform now supports 3M+ daily transactions

Lessons Learned:

• **More thorough testing:** Should have run AD mode for longer duration, covering more scenarios (returns, edge cases) before production migration

• **Better data migration readiness:** Migration service was ready late, only allowed migrating 6 months of data. Should have had 2+ years of data migrated and validated earlier

• **Success factors:** Combination of exhaustive AD testing for correctness + phased rollout for risk mitigation was key to success

Interviewer Feedback:

✓ Excellent cross-functional alignment demonstrated across warehouse, OMS, product, and SRE teams
✓ Strong crisis management under pressure - handled rollback request professionally and resolved quickly
✓ Showed humility and learning mindset when discussing improvements
✓ Phased migration approach shows mature risk management thinking

Question 2: Mentorship and Development

Question: Tell me about a time when you mentored an engineer and they went on to get promoted or take on a more senior role. Walk me through: who were they, what were their gaps, how did you help them, and what was the outcome?

Your Response:

> **Context: Junior engineer at RingCentral working on database optimization project - implementing TTL policy on 70-80 million historical audit documents (1TB data) in CouchBase cluster across 5 regions**

Initial Gaps Identified:

• Intimidated by scale - overwhelmed by 1TB dataset and production impact concerns

• Lacked confidence in making architectural decisions

• Needed guidance on production risk mitigation

• Required help understanding broader system implications

Your Mentorship Approach:

Didn't hand him the solution - Instead, guided him through the thinking process:

• **Batch size optimization:** Helped him think through trade-offs between batch size, database load, and execution time. Discussed how to find the sweet spot.

• **Monitoring and observability:** Showed him what metrics to track - database CPU utilization, query latency, replication lag. How to set up alerts.

• **Production load simulation:** Guided him on how to test in staging with production-like load patterns before actual rollout

• **Risk mitigation:** Taught him to start with one region, validate results, then expand to remaining four regions

• **Rollback planning:** Ensured he had a rollback strategy and understood how to handle production incidents

Growth Observed:

Over the project, he evolved from being intimidated by complexity to thinking holistically about systems, risks, and scale. He learned to consider database load, production outages, monitoring, and could explain technical decisions clearly during demos.

Outcome:

✓ Successfully implemented TTL policy across all 5 regions

✓ Freed up 1TB of storage, enabling cluster downsize from R5.xlarge to R5.large instances

✓ Achieved $20,000 annual cost savings

✓ Engineer gained confidence and was promoted (combination of this project and other work)

✓ He developed ability to handle complex, high-stakes production work independently

Interviewer Feedback:

✓ Excellent mentorship approach - guided rather than directed
✓ Focused on developing judgment and thinking skills, not just technical execution
✓ Clear progression from intimidated junior to confident engineer
✓ Measurable business impact ($20K savings) plus human development impact (promotion)

Question 3: Disagreement with Senior Leaders

Question: Tell me about a time when you disagreed with a senior leader or stakeholder on an important technical or strategic decision. How did you handle it? Did you push back, or did you go along with it?

Your Response - Two Examples:

Example 1: TTL Configuration Strategy (RingCentral)

> **Context: Same database optimization project - implementing TTL on 70-80M audit documents**

The Disagreement:

Senior reviewer wanted TTL configured at database level. You advocated for application-level TTL configuration.

Trade-offs You Presented:

Approach	Pros	Cons
Database-level TTL	• Centralized configuration • No application code changes	• Requires DB scripts • Coordination with SRE team • Slo
Application-level TTL	• Quick implementation • Consistent with existing approach	• Additional latency per write • Configuration in app layer • M

How You Handled It:

• **Understood the concern:** Reviewer was worried about production server load and added latency

• **Provided data:** Demonstrated microsecond-level latency increase with actual measurements. Showed minimal impact with only a few thousand records created per day

• **Showed consistency:** Explained this matched the pattern already used for other tables - maintaining architectural consistency

• **Emphasized quick win:** Highlighted faster delivery vs over-engineering with DB scripts and SRE coordination

• **Aligned stakeholders:** Got director approval after showing no significant production impact

Resolution: Went with application-level TTL. Successfully implemented with no production issues.

Example 2: Multipart Shipment Information Flow (Flipkart)

> **Context: New multipart shipment feature with tight 1-1.5 month launch deadline. Accounting system needed to know if a shipment was multipart to handle child shipment rate calculations correctly.**

The Disagreement:

OMS (Order Management System) team pushed back on providing multipart shipment flag in their API. They wanted proper solution (schema change) but didn't have time before MVP launch. They suggested accounting use forward entries to determine if shipment was multipart.

Your Pushback:

• **Ownership principle:** Accounting is not the owner of 'multipart shipment' information - that belongs to OMS

• **Event ordering issue:** Accounting may receive events out of order (reverse event before forward event). Can't rely on forward entries.

• **Architectural correctness:** Wrong system for this logic - would create technical debt and fragile dependencies

Pragmatic Resolution:

Negotiated a temporary workaround: OMS would add the flag to an existing API endpoint (lighter lift than full schema change). This met the MVP deadline while maintaining architectural correctness. Both teams aligned on the approach and launched successfully on time.

Interviewer Feedback:

✓ Both examples show strong technical reasoning backed by data
✓ You pushed back respectfully with clear architectural principles
✓ Demonstrated pragmatism - found middle ground that satisfied constraints (timeline, correctness)
✓ Showed ability to navigate dependencies and align cross-functional teams under pressure
✓ Staff-level thinking: balance between ideal solutions and practical delivery

Overall Interview Assessment

Final Rating: 8.5/10

Strong Staff Engineer behavioral interview performance demonstrating organizational leadership and cross-functional influence.

Key Strengths:

✓ **Specific, detailed examples:** All answers were concrete with clear context, actions, and outcomes

✓ **Cross-organizational impact:** Demonstrated ability to influence OMS, warehouse, product, SRE teams - not just direct reports

✓ **Stakeholder management:** Showed how to navigate resistance, build consensus, and align competing priorities

✓ **Technical decision-making:** Clear reasoning through trade-offs with data-driven approach

✓ **Mentorship depth:** Guided engineers through thinking process rather than just giving solutions

✓ **Humility and learning:** Openly discussed what you'd do differently (AD testing duration, migration service readiness)

✓ **Crisis management:** Handled warehouse rollback request professionally under BBD timeline pressure

✓ **Pragmatic leadership:** Balanced ideal solutions with practical constraints (timelines, resources)

Areas for Enhancement (8.5 → 9.0):

• **Quantify promotion timeline:** Instead of 'he got promoted', say 'promoted to SDE-2 within 6 months'

• **Keep answers concise:** Some responses were 3-4 minutes; aim for 2-3 minutes in real interviews

• **Emphasize org-level influence earlier:** Lead with cross-org impact rather than building to it

What This Rating Means:

Rating	Level
6/10	Good senior engineer
7.5/10	Solid mid-to-senior level
8.5/10	**Strong Staff Engineer (Your Level)**
9/10	Principal-ready

Interview Readiness Assessment:

You are **interview-ready for Staff Engineer roles**. Your behavioral stories demonstrate real cross-organizational leadership, technical depth, and the ability to drive alignment under constraints. Combined with your system design skills, you're well-positioned for Staff-level interviews at top companies.

Preparation Notes & Key Takeaways

Interview Tips Based on This Session:

• **Start with impact:** Lead with org-level influence, not just what you built

• **Use STAR format naturally:** Situation → Task → Action → Result, but don't make it formulaic

• **Quantify outcomes:** Always include metrics (timelines, cost savings, team sizes, promotion timelines)

• **Show trade-off thinking:** Explain why you chose approach A over B, what you sacrificed

• **Demonstrate learning:** What would you do differently? Shows growth mindset

• **Keep it conversational:** 2-3 minutes per story maximum; interviewer will ask follow-ups

• **Prepare 3-4 strong stories:** That can answer multiple behavioral question types

Your Strongest Stories (Reusable):

Story	Can Answer These Questions
Flipkart Monolith Migration	• Cross-functional alignment • Handling resistance • Managing risk • Workin
RingCentral TTL Mentorship	• Mentoring engineers • Developing talent • Delegation • Cost optimization<
TTL Configuration Disagreement	• Disagreeing with seniors • Technical decision-making • Data-driven arguments
OMS Multipart Shipment	• Cross-team dependencies • Architectural principles • Pragmatic compromises<

Common Staff-Level Behavioral Questions to Prepare:

1. Tell me about a time you influenced a decision without direct authority

2. Describe a situation where you had to make a difficult trade-off decision

3. Tell me about the most complex technical project you led

4. How do you handle disagreements with other senior engineers?

5. Describe a time you failed and what you learned

6. Tell me about mentoring someone who struggled

7. How do you prioritize when everything is urgent?

8. Describe driving technical standards across multiple teams

Next Steps:

1. Practice condensing your stories to 2-3 minutes each

2. Add specific promotion timelines and metrics to your mentorship stories

3. Prepare 1-2 'failure' stories (things that didn't go well initially)

4. Practice system design interviews to complement behavioral strength

5. Update CV based on earlier feedback (make it more concise, add metrics)