

System Design Interview Questions for Engineering Managers

This document contains a comprehensive collection of system design interview questions specifically curated for Engineering Manager positions. These questions are organized by the new 4-category framework and focus on both technical depth and leadership aspects.

Table of Contents

- [1. Question Categories](#)
- [2. Distributed Systems & Infrastructure](#)
- [3. Data & AI/ML Systems](#)
- [4. Real-time & Communication Systems](#)
- [5. Product & Platform Systems](#)
- [6. Interview Approach](#)
- [7. EM-Specific Considerations](#)

Question Categories

The system design questions are organized into 4 main categories that cover the breadth of systems an Engineering Manager might encounter:

1. Distributed Systems & Infrastructure (4 questions)

Focus on large-scale distributed systems, infrastructure design, content delivery networks, and monitoring systems.

2. Data & AI/ML Systems (10 questions)

Emphasis on data platforms, AI/ML systems, search engines, recommendation systems, and data processing pipelines.

3. Real-time & Communication Systems (5 questions)

Real-time systems, messaging platforms, streaming services, social networks, and collaboration tools.

4. Product & Platform Systems (3 questions)

E-commerce platforms, startup system architectures, content management systems, and geospatial systems.

Distributed Systems & Infrastructure

1. Design a Content Delivery Network (CDN)

Complexity: High

Time: 45-60 minutes

Key Focus Areas: Global distribution, caching strategies, edge computing

Leadership Aspects: Infrastructure team coordination, vendor management, cost optimization

2. Design Meta's News Feed system

Complexity: High

Time: 45-60 minutes

Key Focus Areas: Feed generation, ranking algorithms, real-time updates

Leadership Aspects: Cross-team collaboration, scalability planning, performance optimization

3. Design a cloud monitoring service (like Azure Monitor)

Complexity: Medium

Time: 45 minutes

Key Focus Areas: Metrics collection, alerting, dashboard visualization

Leadership Aspects: SRE team management, incident response processes, tool selection

4. Design a distributed training pipeline for large AI models

Complexity: High

Time: 60 minutes

Key Focus Areas: Distributed computing, model parallelism, resource management

Leadership Aspects: Research team coordination, infrastructure scaling, cost management

Data & AI/ML Systems

1. Design a system to serve a large language model (e.g., GPT-4) to millions of users with low latency

Complexity: High

Time: 60 minutes

Key Focus Areas: Model serving, caching, load balancing

Leadership Aspects: AI team management, infrastructure scaling, cost optimization

2. Design Netflix's content recommendation system

Complexity: High

Time: 45-60 minutes

Key Focus Areas: Machine learning pipelines, real-time recommendations, A/B testing

Leadership Aspects: Data science team coordination, product collaboration, experimentation

3. Design TikTok's For You Page algorithm

Complexity: High

Time: 45-60 minutes

Key Focus Areas: Content ranking, user engagement, real-time processing

Leadership Aspects: Algorithm team management, content policy, global scaling

4. Design Airbnb's search and ranking system

Complexity: Medium

Time: 45 minutes

Key Focus Areas: Search indexing, ranking algorithms, personalization

Leadership Aspects: Search team coordination, product requirements, performance optimization

5. Design LinkedIn's professional network feed algorithm

Complexity: Medium

Time: 45 minutes

Key Focus Areas: Professional content ranking, network effects, engagement optimization

Leadership Aspects: Product team collaboration, user experience, business metrics

6. Design a system for large-scale data labeling and annotation

Complexity: Medium

Time: 45 minutes

Key Focus Areas: Workflow management, quality control, human-in-the-loop systems

Leadership Aspects: Operations team management, vendor coordination, quality assurance

7. Design a cloud-native data warehouse with automatic scaling

Complexity: Medium

Time: 45 minutes

Key Focus Areas: Data storage, query optimization, auto-scaling

Leadership Aspects: Data engineering team, cost optimization, performance monitoring

8. Design a system for safe AI model deployment with constitutional AI principles

Complexity: High

Time: 60 minutes

Key Focus Areas: AI safety, model governance, deployment pipelines

Leadership Aspects: AI safety team, compliance, risk management

9. Design Reddit's content ranking and moderation system

Complexity: Medium

Time: 45 minutes

Key Focus Areas: Content scoring, moderation workflows, community management

Leadership Aspects: Content team coordination, policy enforcement, community relations

10. Design a scalable e-commerce product page with real-time inventory and recommendations

Complexity: Medium

Time: 45 minutes

Key Focus Areas: Inventory management, recommendation engines, page performance

Leadership Aspects: E-commerce team coordination, product management, user experience

Real-time & Communication Systems

1. Design Uber's real-time ride matching system

Complexity: High

Time: 45-60 minutes

Key Focus Areas: Real-time matching, geospatial algorithms, demand prediction

Leadership Aspects: Marketplace team coordination, algorithm optimization, operational efficiency

2. Design Meta Messenger or a similar large-scale chat system

Complexity: High

Time: 45-60 minutes

Key Focus Areas: Real-time messaging, presence systems, message delivery

Leadership Aspects: Messaging team management, privacy considerations, global scaling

3. Design a global video streaming platform with adaptive bitrate

Complexity: High

Time: 60 minutes

Key Focus Areas: Video encoding, CDN optimization, adaptive streaming

Leadership Aspects: Media team coordination, infrastructure scaling, user experience

4. Design YouTube or a large-scale video streaming platform

Complexity: High

Time: 60 minutes

Key Focus Areas: Video processing, content delivery, recommendation systems

Leadership Aspects: Platform team management, creator relations, monetization

5. Design a system for real-time document co-authoring (like Office 365)

Complexity: Medium

Time: 45 minutes

Key Focus Areas: Operational transformation, conflict resolution, real-time sync

Leadership Aspects: Collaboration tools team, user experience, enterprise features

Product & Platform Systems

1. Design a scalable system architecture for a fast-growing startup

Complexity: Medium

Time: 45 minutes

Key Focus Areas: Scalability planning, technology choices, cost optimization

Leadership Aspects: Technical strategy, team scaling, architectural decisions

2. Design a minimum viable product (MVP) architecture that can scale

Complexity: Medium

Time: 45 minutes

Key Focus Areas: MVP design, scalability planning, technical debt management

Leadership Aspects: Startup engineering leadership, resource allocation, growth planning

3. Design Google Maps

Complexity: High

Time: 60 minutes

Key Focus Areas: Geospatial data, routing algorithms, real-time traffic

Leadership Aspects: Maps team coordination, data partnerships, global infrastructure

Interview Approach

1. Clarify Requirements (5-10 minutes)

- **Functional Requirements:** What features need to be supported?
- **Non-functional Requirements:** Scale, performance, availability expectations
- **Constraints:** Budget, timeline, existing systems
- **Success Metrics:** How will success be measured?

EM Focus: Show ability to ask business-relevant questions and understand stakeholder needs.

2. High-Level Design (10-15 minutes)

- Start with simple, high-level architecture
- Identify major components and data flow
- Focus on core functionality first
- Avoid premature optimization

EM Focus: Demonstrate ability to communicate technical concepts clearly and make architectural decisions.

3. Deep Dive (15-20 minutes)

- Choose 1-2 critical components to elaborate
- Discuss data models and algorithms
- Address specific technical challenges
- Show technical depth

EM Focus: Balance technical details with leadership considerations like team structure and implementation planning.

4. Scale the Design (10-15 minutes)

- Identify bottlenecks and scaling challenges
- Discuss horizontal vs vertical scaling
- Add caching, load balancing, data partitioning
- Consider global distribution

EM Focus: Show understanding of operational challenges and team coordination needed for scaling.

5. Address Edge Cases (5-10 minutes)

- Failure scenarios and recovery
- Data consistency and conflict resolution
- Monitoring and alerting
- Security considerations

EM Focus: Demonstrate operational thinking and risk management skills.

EM-Specific Considerations

Technical Leadership

- **Decision Making:** How would you make technology choices and architectural decisions?
- **Technical Communication:** How would you explain technical concepts to different audiences?
- **Code Quality:** How would you ensure engineering best practices across teams?
- **Technical Debt:** How would you balance feature development with technical debt?

Team and Process Design

- **Team Structure:** How would you organize teams around this system?
- **Development Process:** What development methodologies and practices would you implement?
- **Skill Development:** What skills would your team need and how would you develop them?
- **Cross-team Coordination:** How would you manage dependencies between teams?

Risk Management

- **Technical Risks:** What are the biggest technical risks and how would you mitigate them?
- **Operational Risks:** How would you ensure system reliability and handle incidents?

- **Business Risks:** How would you validate assumptions and measure success?
- **Timeline Risks:** How would you manage project timelines and deliverables?

Stakeholder Management

- **Product Collaboration:** How would you work with product managers on requirements?
- **Executive Communication:** What would you communicate to leadership about progress and challenges?
- **Customer Impact:** How would you ensure the system meets customer needs?
- **Resource Planning:** How would you plan and justify resource requirements?

Preparation Tips

Technical Preparation

1. **Review System Design Fundamentals:** Scalability, reliability, consistency, availability
2. **Study Real-world Systems:** Read engineering blogs from major tech companies
3. **Practice Drawing:** Get comfortable with whiteboard/digital drawing tools
4. **Learn Common Patterns:** Load balancing, caching, database sharding, microservices

Leadership Preparation

1. **Prepare Team Examples:** Have stories about building and managing engineering teams
2. **Practice Communication:** Explain technical concepts to non-technical audiences
3. **Think About Process:** Be ready to discuss development methodologies and practices
4. **Consider Trade-offs:** Practice discussing technical and business trade-offs

During the Interview

1. **Ask Clarifying Questions:** Understand the scope and requirements clearly
2. **Think Out Loud:** Explain your thought process and reasoning
3. **Show Leadership:** Discuss team coordination and process considerations
4. **Be Pragmatic:** Balance ideal solutions with practical constraints
5. **Engage the Interviewer:** Ask for feedback and iterate on your design

This document contains system design questions from the EM Interview Prep platform. For detailed solutions, frameworks, and additional guidance, please refer to the full platform.