



Armen Melkumyan • 1st

Technical / Solutions Architect

1yr •

...

How to design system like Telegram Part 3: Networking and Real-time Communication

Message Queues and Caching

◆ Message Queues:

What: Message queues manage the flow of messages and requests.

Why: Decoupling components, managing load, orderly processing.

Use Cases: Handling high volumes, notifications, asynchronous tasks.

◆ Caching:

What: Temporarily storing frequently accessed data.

Why: Reduces database load, improves response time.

Use Cases: Storing session data, chat histories, contact lists.

WebSocket Protocol

What: Full-duplex communication over TCP.

Why: Real-time communication, reduced latency, efficiency.

◆ Load Balancing and Networking

Load Balancers: Distribute network traffic across servers.

Why: Prevent overloads, reliability, availability.

Use Cases: Distributing requests, managing traffic spikes.

◆ Content Delivery Network (CDN):

What: Network of servers for efficient content delivery.

Why: Speed, efficiency for media content.

Use Cases: Delivering multimedia content quickly.

Challenges and Solutions

High Volume and Scalability: Scalable queues, caching strategies.

Latency and Efficiency: WebSockets, CDNs.

⚙️ Load Distribution: Load balancing, routing strategies.

Conclusion

💡 In summary, the networking and real-time communication aspect is key to Telegram's functionality, involving message queues, caching, WebSockets, load balancing, and CDNs for a robust and efficient messaging app.

👍 4
