

Part 4: Suppose we have a microservice architecture with many different services. We also have an app that gets data from these different services. The app runs on both Android and IOS. The problem we face is that when we start this app, the first page on the app needs to show data that comes from 15 different microservices. The time to retrieve all this data from all 15 microservices takes 10 seconds, which is too slow and not acceptable. The app should get all the required data as fast as possible. It should not take more than 2.5 seconds to retrieve all necessary data. How can we solve this problem?

To reduce the data retrieval time from 15 microservices to under 2.5 seconds, consider these strategies:

- **Parallel Requests:** Make all requests to the microservices concurrently rather than sequentially.
- **Caching:** Implement caching to store frequently accessed data, reducing the need for repeated calls.
- **Aggregation Service:** Create a service that aggregates data from all microservices and provides a single endpoint for the app.
- **Preloading Data:** Pre-fetch and cache data before it's needed, such as when the app is in the background.
- **GraphQL:** Use GraphQL to request specific data from multiple services in a single call.
- **Optimize Microservices:** Improve the performance of individual microservices using tools like Zipkin for tracing and optimizations like database indexing.
- **Load Balancing:** Ensure microservices are load-balanced and have redundancy to handle traffic efficiently.
- **Fallback Strategies:** Implement circuit breakers and fallback methods to handle slow or failed services gracefully.