

System Design In DevOps

Understanding the Role of DevOps in System Design

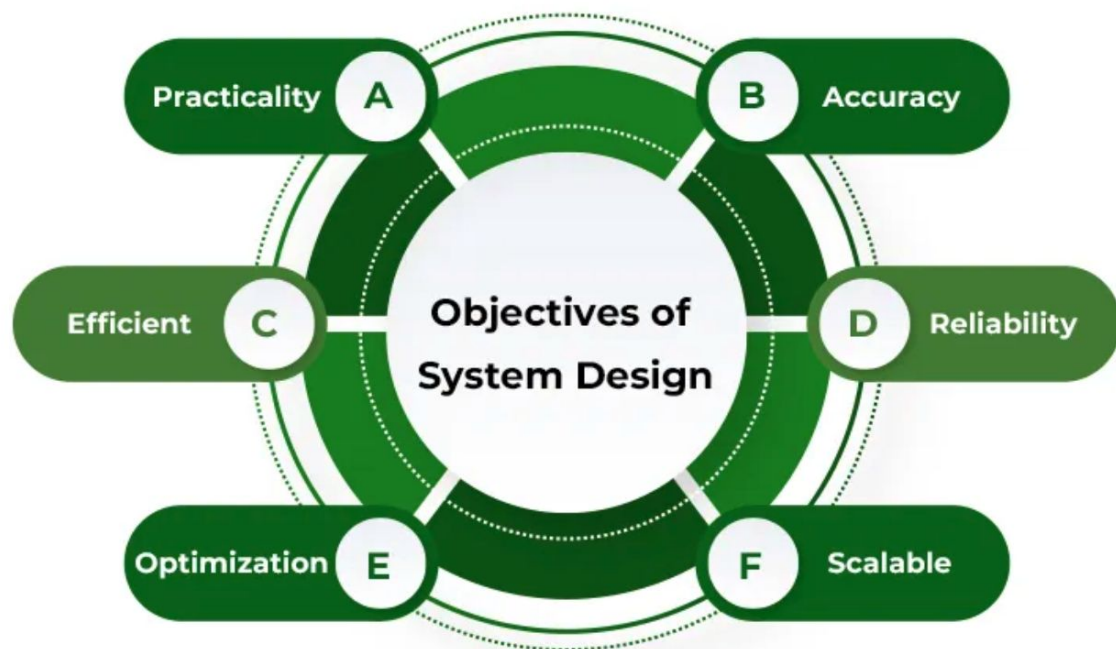
Principles of Scalable and Reliable System Architecture

Continuous Integration and Deployment (CI/CD) for Scalability

Infrastructure as Code (IaC) for Reliable System Provisioning

Monitoring and Logging for Scalability and Reliability





1. **Scaling of Applications & Databases and types of Scaling.**
2. **Load balancing, Proxies, Service Mesh**
3. **High-level design of Monolithic and Microservices architecture(Stateless and Stateful)**
4. **Caching and Content Delivery Network**
5. **Networking & Security**
6. **Database Sharding, CAP Theorem, Database schema Design, SQL and NoSQL Databases**
7. **Concurrency, Rate Limiting, Circuit Breakers**
8. **API Gateways, HTTPS Vs Rest Vs GraphQL,**
9. **Containerisation and Orchestration**
10. **Cost Optimization and Planning**
11. **High Availability, Resiliency, and Reliability of every component in the system**
12. **Shell Scripting or Python**
13. **Monitoring and Observability**
14. **Disaster Recovery**



scalable and reliable system architecture:

A scalable and reliable system architecture is built on several key principles, including modularity, redundancy, fault tolerance, and elasticity.

Modularity involves breaking down complex systems into smaller, manageable components that can be independently scaled and maintained.

Redundancy ensures that critical components have backups in place to prevent single points of failure, while fault tolerance enables the system to continue functioning even in the face of failures.

Elasticity allows the system to dynamically adjust its resources based on workload fluctuations, ensuring optimal performance and cost-efficiency.

In the context of scalability, CI/CD enables teams to rapidly deploy updates and scale resources based on demand, ensuring that the system can handle increasing workloads without manual intervention.

IaC allows teams to define infrastructure configurations in code, track changes in version control, and automate the provisioning of resources, leading to more reliable and efficient system deployments.



Monitoring and Logging

Effective monitoring and logging are essential for maintaining the scalability and reliability of systems.

Monitoring tools provide real-time insights into system performance, resource utilization, and potential bottlenecks, enabling teams to proactively address issues before they impact users.

Logging, on the other hand, captures detailed information about system events, errors, and user interactions, helping teams troubleshoot issues, conduct root cause analysis, and improve system reliability over time.



NATIVE COMPONENTS

APPLICATION - BUSINESS INTELLIGENCE COMPONENTS



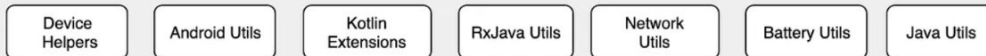
APPLICATION CORE



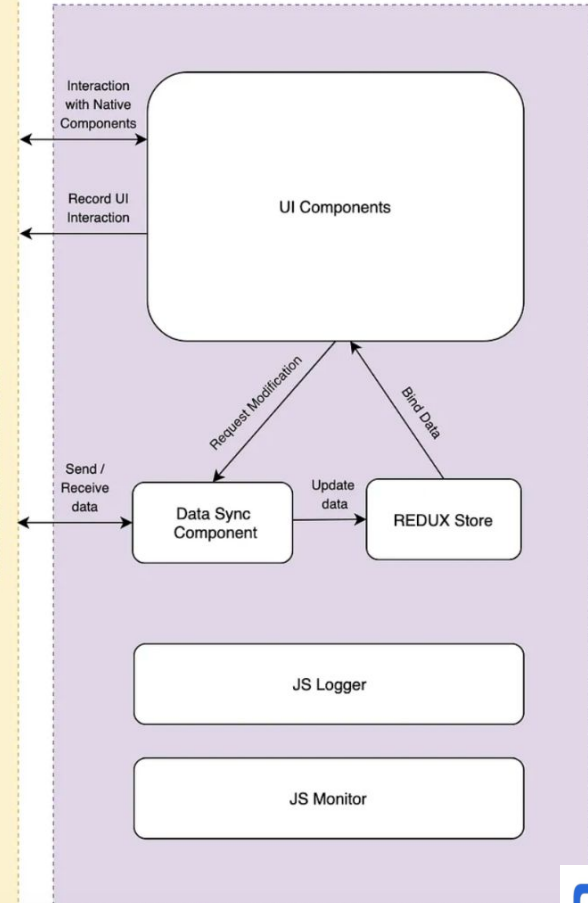
FRAMEWORKS / MODULES



COMMONS



JAVASCRIPT COMPONENTS



REACT NATIVE BRIDGE

