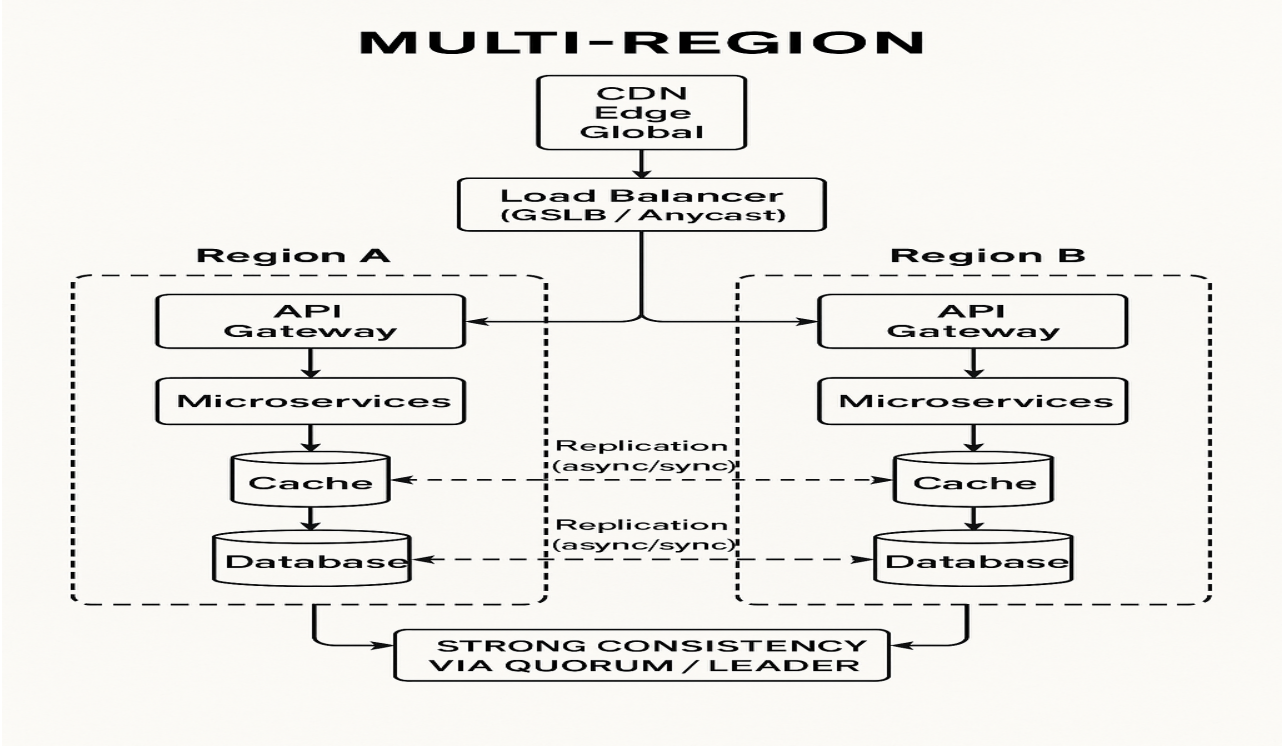
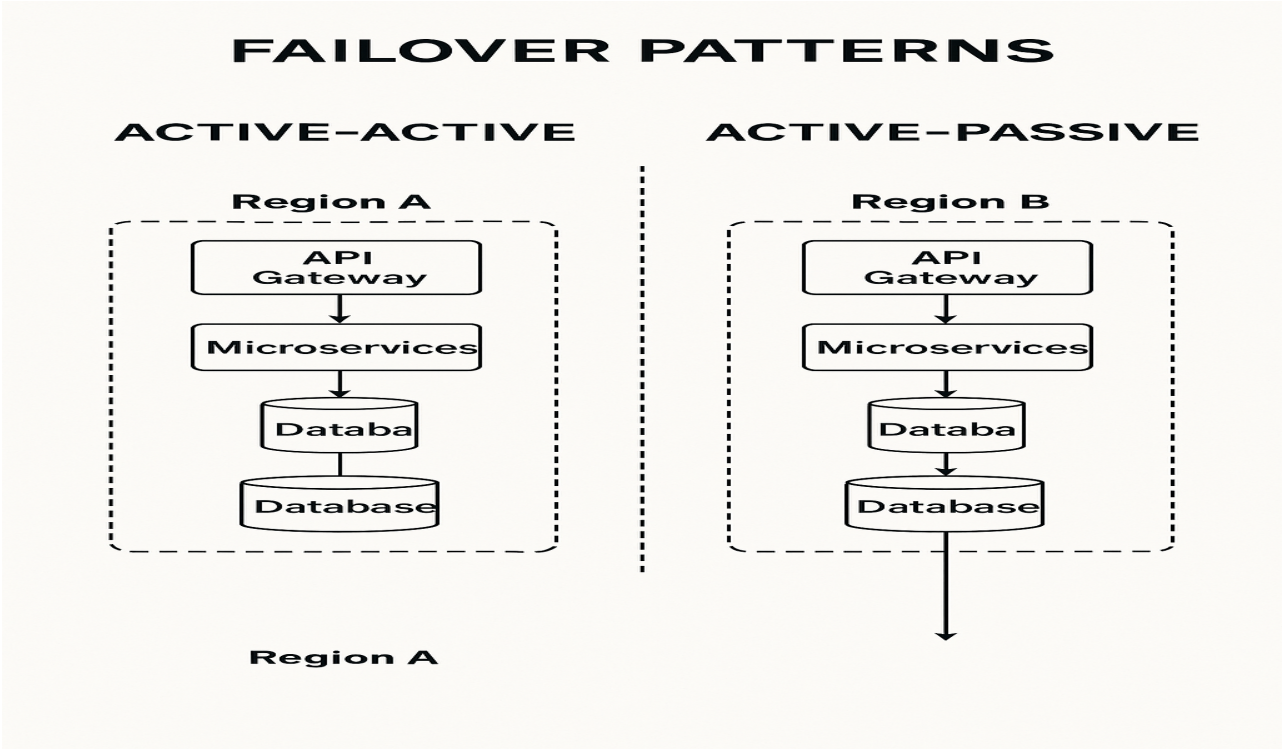
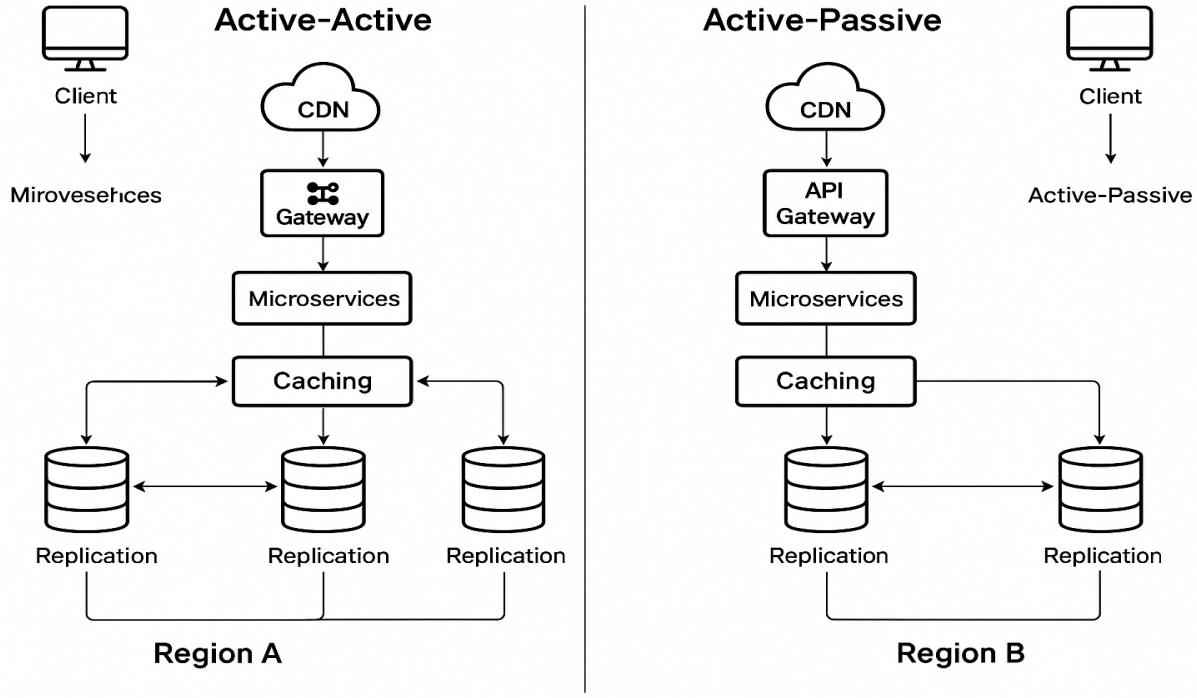
API Gateway vs CDN, Replication Architectures, and GraphQL Overview

Part 1: API Gateway vs CDN
Comparison of API Gateway and CDN roles and how they complement each other in cloud systems.

Part 2: Multi-Region and Failover Architectures
Includes replication mechanisms, active-active vs active-passive failover models, and consistency management.

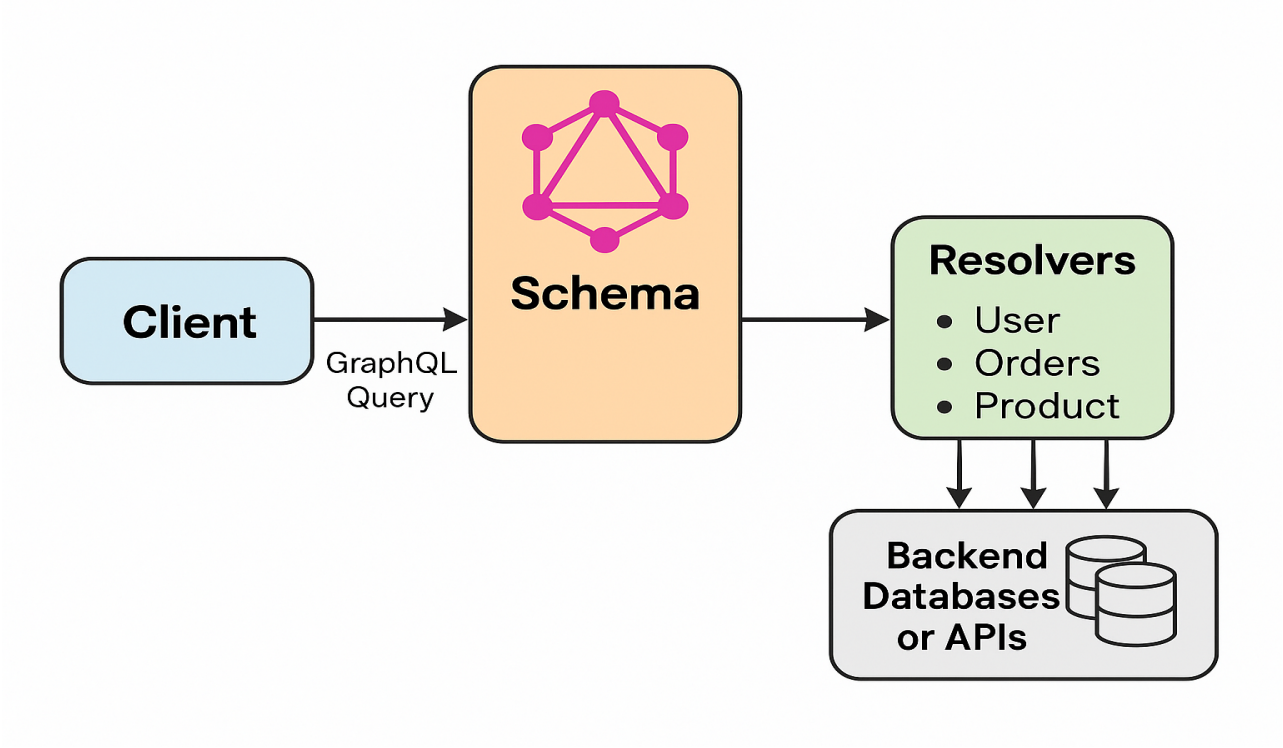



FAILOVER PATTERNS
ACTIVE–ACTIVE
ACTIVE–PASSIVE
Region A
API Gateway
Microservices
Databa
Database
Region B
API Gateway
Microservices
Databa
Database
Region A


Active-Active
Client
Miroveseнces
CDN
Gateway
Microservices
Caching
Replication
Replication
Replication
Region A
Active-Passive
CDN
API Gateway
Microservices
Caching
Client
Active-Passive
Replication
Replication
Region B

Part 3: GraphQL Overview

GraphQL is a query language and runtime for APIs that allows clients to request exactly the data they need. It was developed by Facebook (Meta) to make APIs more flexible, efficient, and developer-friendly. Unlike REST, GraphQL uses a single endpoint and structured queries to minimize over-fetching and under-fetching.



1. Core Architecture
GraphQL consists of three core components:
- **Schema:** Defines available types and data relationships.
- **Query:** Specifies exactly what data the client wants.
- **Resolver:** Fetches data from the appropriate source.

2. Data Types
GraphQL defines types such as Query, Mutation, Subscription, Object, Enum, and Scalar types. Queries retrieve data, Mutations modify data, and Subscriptions handle real-time updates.

3. Execution Flow
Client sends query → Server validates against schema → Resolvers fetch data → Aggregated JSON response returned.

4. Benefits
- Reduced API round-trips.
- Strong typing and schema-based validation.
- Unified access across multiple backend sources.
- No need for API versioning.

5. Performance Considerations
Challenges like deep nested queries or N+1 problems can be mitigated using query depth limits, batching (DataLoader), and caching layers.

6. Common Implementations
GraphQL is used by GitHub, Netflix, Shopify, Airbnb, and AWS AppSync. Popular frameworks include Apollo Server, Hasura, and Express-GraphQL.

7. REST vs GraphQL
| Feature | REST | GraphQL | |----------|------|----------| | Endpoints | Multiple | Single | | Data shape | Fixed | Client-defined | | Over-fetching | Common | None | | Versioning | Required | Not needed | | Efficiency | High for simple data | High for complex nested data |

8. When to Use GraphQL
Best suited for frontend-heavy, data-intensive, or multi-source aggregation applications.
Avoid when data is flat, caching is critical, or backend services are simple RESTful APIs.

9. Example Lifecycle
1. Client sends query
2. Server validates schema
3. Resolvers fetch data
4. Response is assembled into JSON
5. Returned to client with exactly the requested fields.