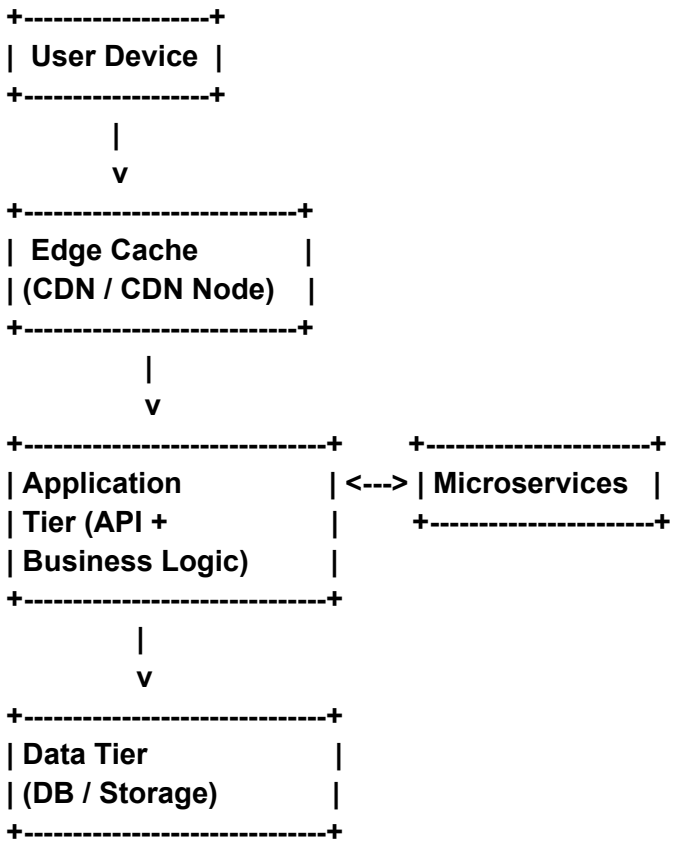
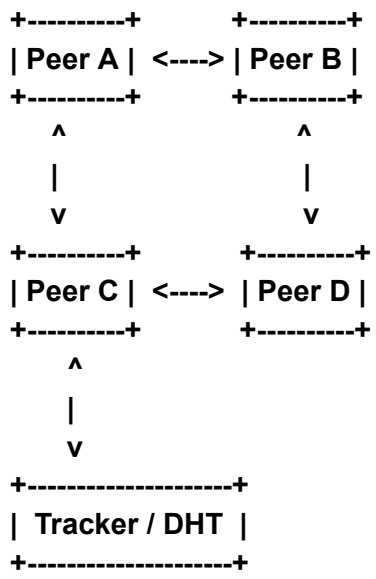
DISTRIBUTED SYSTEMS ARCHITECTURES

1.NETFLIX ARCHITECTURE(Multi-tier + Edge Caching)



2.BitTorrent Architecture (Pure Peer - to - Peer)



3.Explain how each architecture handles load, failure, and scaling

Netflix (Multi-tier + Edge Caching):

- **Load Handling:**
    - Edge caching reduces server load
    - Load balancers distribute requests across multiple servers
- **Failure Handling:**
    - Redundant servers ensure service continuity
    - Failover mechanisms in case of server failure
    - Cached content at edge nodes helps maintain availability
- **Scaling:**
    - Horizontal scaling of application servers
    - Adding more CDN/edge nodes to handle traffic spikes

BitTorrent (Pure P2P):

- **Load Handling:**
    - Load is shared across all peers
    - More peers improve download speed and resource availability
- **Failure Handling:**
    - Peer redundancy ensures files remain available
    - Decentralized storage prevents single points of failure
- **Scaling:**
    - Capacity grows naturally as more peers join
    - Each new peer contributes storage and bandwidth