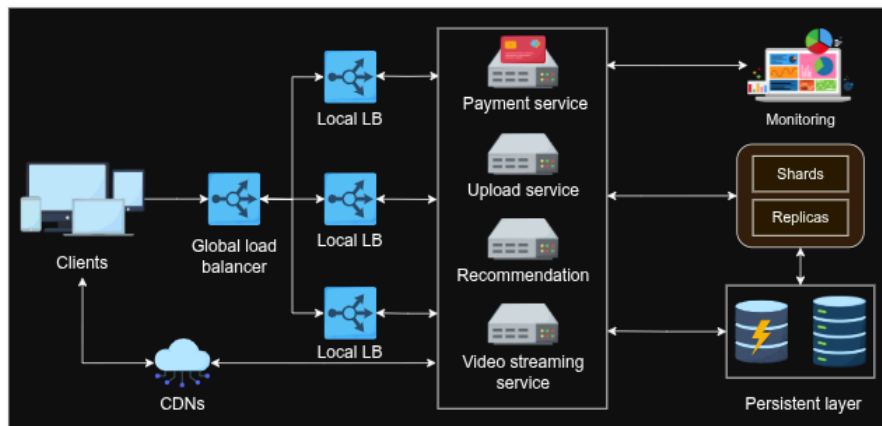


Netflix vs BitTorrent: Distributed System Architectures

1. Netflix Architecture (Multi-tier + Edge Caching)

Netflix uses a **hierarchical, multi-tier distributed architecture** combined with **edge caching** through its Open Connect CDN.

Diagram: Netflix Architecture



A high-level design of fault-tolerant streaming service

Handling Load

Netflix reduces backend load by serving most video content directly from edge caches located inside ISPs. Only lightweight API calls (login, recommendations) reach central servers, allowing massive concurrent streaming.

Handling Failure

Failures are isolated by tier. If an edge cache fails, traffic is rerouted to another nearby cache or region. Backend services are replicated across availability zones to ensure resilience.

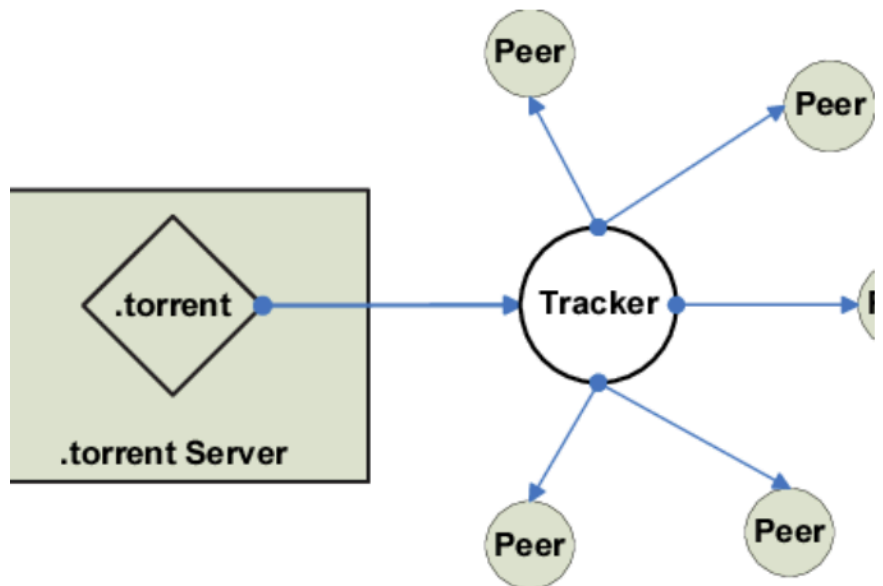
Handling Scaling

Netflix scales horizontally. New users are handled by adding more edge servers and cloud instances. This makes scaling predictable and centrally controlled.

BitTorrent Architecture (Pure Peer-to-Peer)

BitTorrent is a **fully decentralized peer-to-peer (P2P) system** where users both consume and provide resources.

(Each peer uploads and downloads file chunks)



Load is distributed across peers. As more users join a torrent, total available bandwidth increases, making the system more efficient under heavy demand.

Handling Failure

The system is highly fault tolerant. If one peer leaves or fails, others continue sharing file chunks. No single point of failure exists.

Handling Scaling

BitTorrent scales **organically**. More users improve performance instead of degrading it, making it extremely scalable for large file distribution.

Aspect	Netflix	BitTorrent
Control	Centralized	Fully decentralized
Load	Edge caching	Shared by peers
Failure	Redundancy & rerouting	Peer replacement
Scaling	Planned, provider-driven	User-driven, self-scaling