Lesson 5 -> Reverse Proxy

What is a Reverse Proxy?

A reverse proxy is a **server that sits between clients and your backend servers**.

Clients send requests to the reverse proxy.
The reverse proxy forwards those requests to one or more backend services and returns the response.

The client **never directly talks to backend services**.

Think of it as:

> a single public entry point for multiple internal services.

Why reverse proxy exists (core problem it solves)

Without a reverse proxy:

- Clients must know where every service lives
- Backend IPs are exposed
- Security, SSL, load balancing must be implemented by each service

At scale, this becomes **unmanageable**.

Reverse proxy centralizes these concerns.

What does a Reverse Proxy actually do?

1. **Request routing**
   Routes incoming requests to the correct backend.

Example:

- /login → Auth service
- /products → Product service
- /recommend → ML service

This allows you to add or move services **without changing clients**.

---

2. **Load balancing**
   Distributes traffic across multiple instances of the same service.

If you have 5 instances of an ML inference service, the reverse proxy ensures:

- No single instance gets overloaded
- Traffic is spread evenly

This is critical for real-time ML.

---

3. **Security & isolation**

- Backend servers are not exposed to the internet
- Reverse proxy can block malicious traffic
- Rate limiting and IP filtering are applied once, centrally

Your ML models stay protected.

---

4. **SSL termination**
   Reverse proxy handles HTTPS:

- Decrypts incoming requests
- Forwards plain HTTP to backend services

This removes SSL complexity from every service.

---

5. **Caching**
   Frequently requested responses can be cached:

- Product lists
- Static recommendations
- Metadata APIs

This reduces backend load dramatically.

---

6. **Observability & control**
   Because all traffic passes through one point:

- Centralized logging

- Metrics collection
- Traffic shaping
- Canary or A/B routing

This is extremely useful for ML experiments.

Reverse Proxy in a Real ML System

Typical flow:
Client
→ Reverse Proxy / API Gateway
→ User service
→ Recommendation service (ML)
→ Feature store

Why ML systems need it:

- ML inference is expensive
- Traffic spikes are common
- You must protect model endpoints
- You may want to route 5% traffic to a new model

All of this is controlled at the reverse proxy.

Advantages of Using a Reverse Proxy

1. Simplifies system architecture
   Clients only know one endpoint.
2. Improves scalability
   Backend services can scale independently.
3. Improves security
   Backends are hidden and protected.
4. Improves reliability
   If one instance fails, traffic is routed elsewhere.
5. Enables advanced deployment strategies
   Canary, blue-green, shadow deployments.
6. Reduces duplication
   No need to implement auth, SSL, rate limiting in every service.

Does a Reverse Proxy have downsides? Yes.

This is important.

---

1) Single point of failure (if poorly designed)

If you run:

- One reverse proxy
- No redundancy

Then:
If it goes down → entire system is down.

Mitigation:

- Multiple proxy instances
- Load balancer in front of proxy

---

2) Added latency

Reverse proxy adds:

- One extra network hop
- TLS termination overhead

In practice:

- Usually a few milliseconds
- Acceptable for most systems
- Critical systems optimize heavily

---

3) Configuration complexity

As systems grow:

- Routing rules
- Security policies
- Rate limits
- Timeouts

Misconfiguration can cause outages.

This is why reverse proxy configs are treated as **production code**.

---

4) Debugging becomes indirect

When something fails:

- Is it the proxy?
- Is it routing?
- Is it backend?

You need good observability to debug effectively.

---

5) Not ideal for all traffic

For:

- Very high-throughput internal traffic
- Low-latency service meshes

Sometimes direct service-to-service communication or gRPC mesh is better.

Reverse Proxy vs “Not using one”

Small systems can survive without it.
Large systems **cannot**.

Once you have:

- Multiple services
- Multiple teams
- ML inference at scale

A reverse proxy becomes **mandatory infrastructure**.