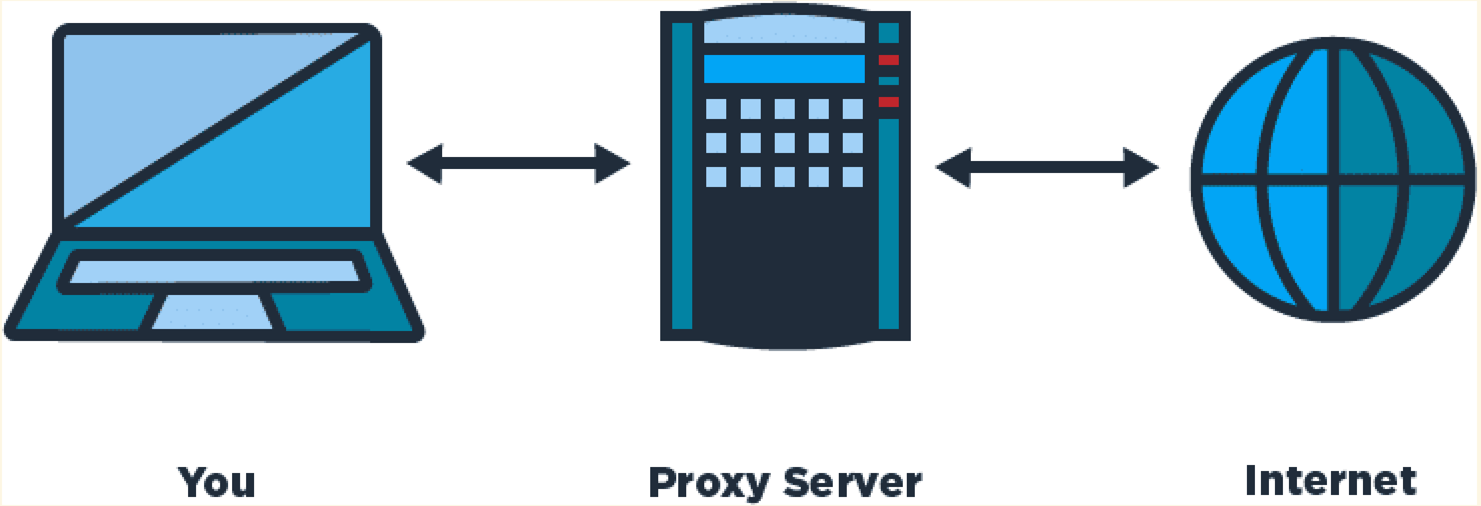
REVERSE PROXYING

Architecture & Deployment

WHAT IS A PROXY?



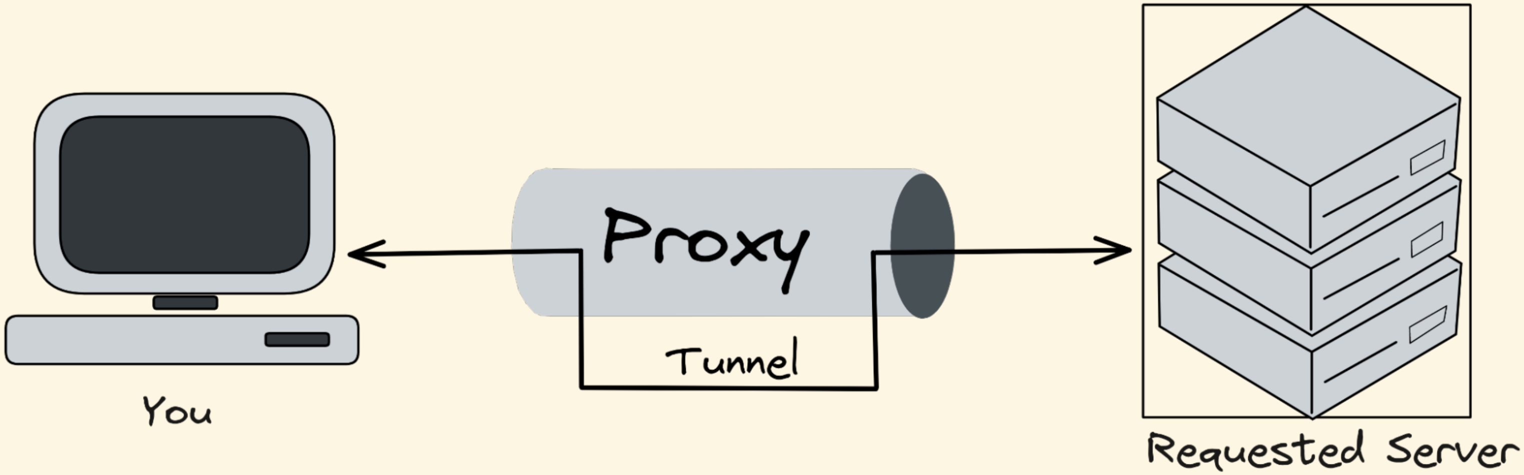
A **proxy server** is a computer or application that acts as an **intermediary** for requests from clients seeking resources from other servers.

TYPES OF PROXY SERVERS

There are 3 main kinds:

- **Tunneling proxy** or **gateway**
- **Forward proxy**
- **Reverse proxy**

TUNNELING PROXY



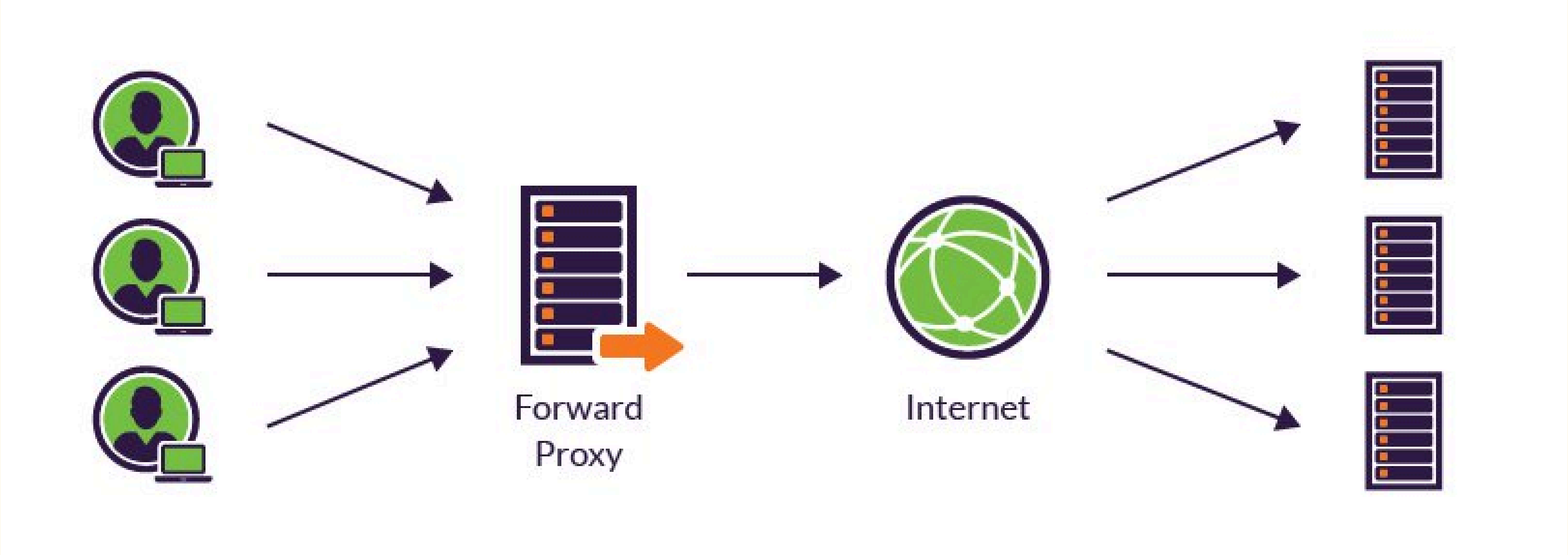
A tunneling proxy can pass **unmodified requests and responses** from one network to another.

Speaker notes

It can also be used to encapsulate a protocol into another, such as running IPv6 over IPv4. For example, an SSH connection may be relayed by a proxy server to a different target server. The proxy server simply passes the packets through, with no ability to compromise the security of the communication.

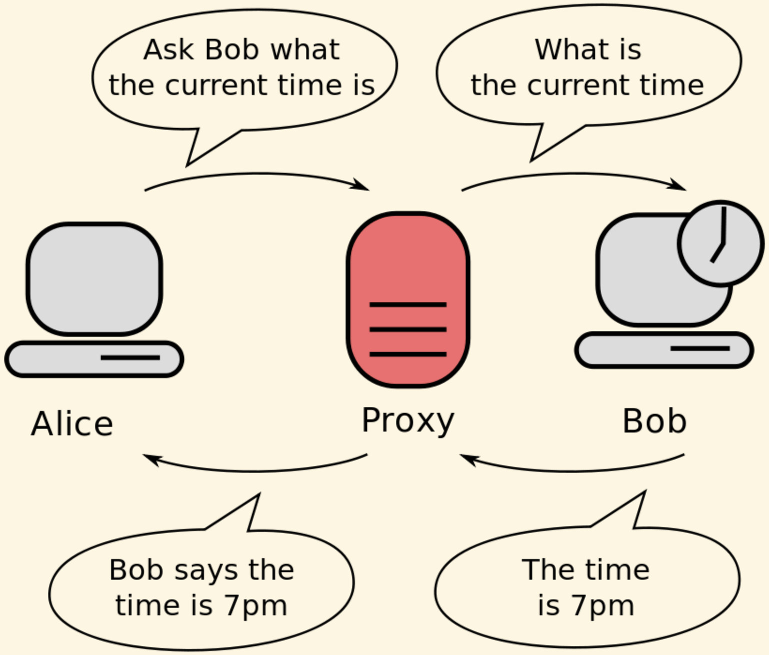
Virtual private networks (VPN) use tunneling protocols, usually with an additional layer of encryption.

FORWARD PROXY



A **forward proxy** retrieves data from a server on behalf of a client. It is said to be **open** if it is accessible by any Internet user.

ANONYMOUS FORWARD PROXY

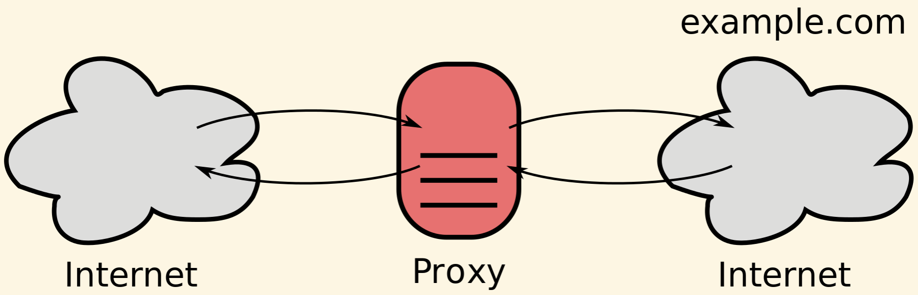


Conceals the identity of the client.

Speaker notes

An **anonymous forward proxy** reveals its identity as a server but conceals that of the client. Since the target server does not know who the original client is, it can be used to protect privacy. VPNs are often used in combination with this type of proxy server.

TRANSPARENT FORWARD PROXY



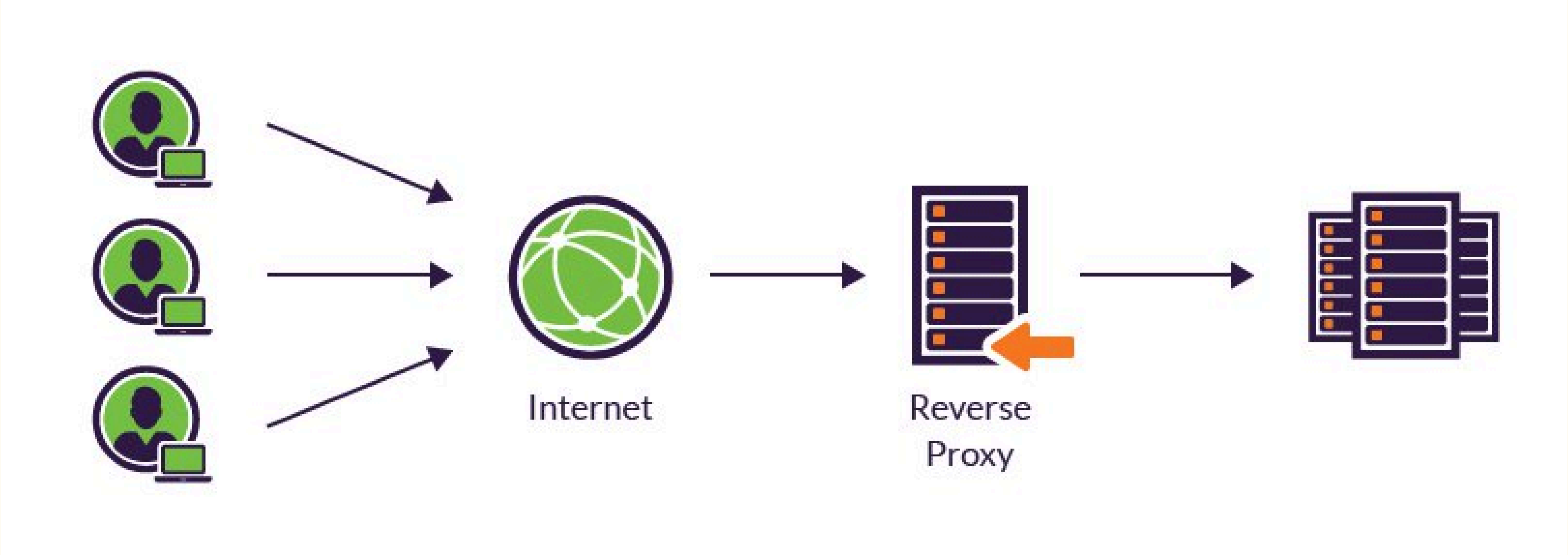
Identifies both itself and the original client.

Speaker notes

A **transparent forward proxy** identifies both itself and the original client through the use of HTTP headers. It can be used to cache websites. Schools often use this kind of proxy to restrict access to particular websites (e.g. Facebook).

When many clients go through the same forward proxy server, its IP address may get banned, since the target server only sees one computer (the proxy) making too many requests at the same time.

REVERSE PROXY

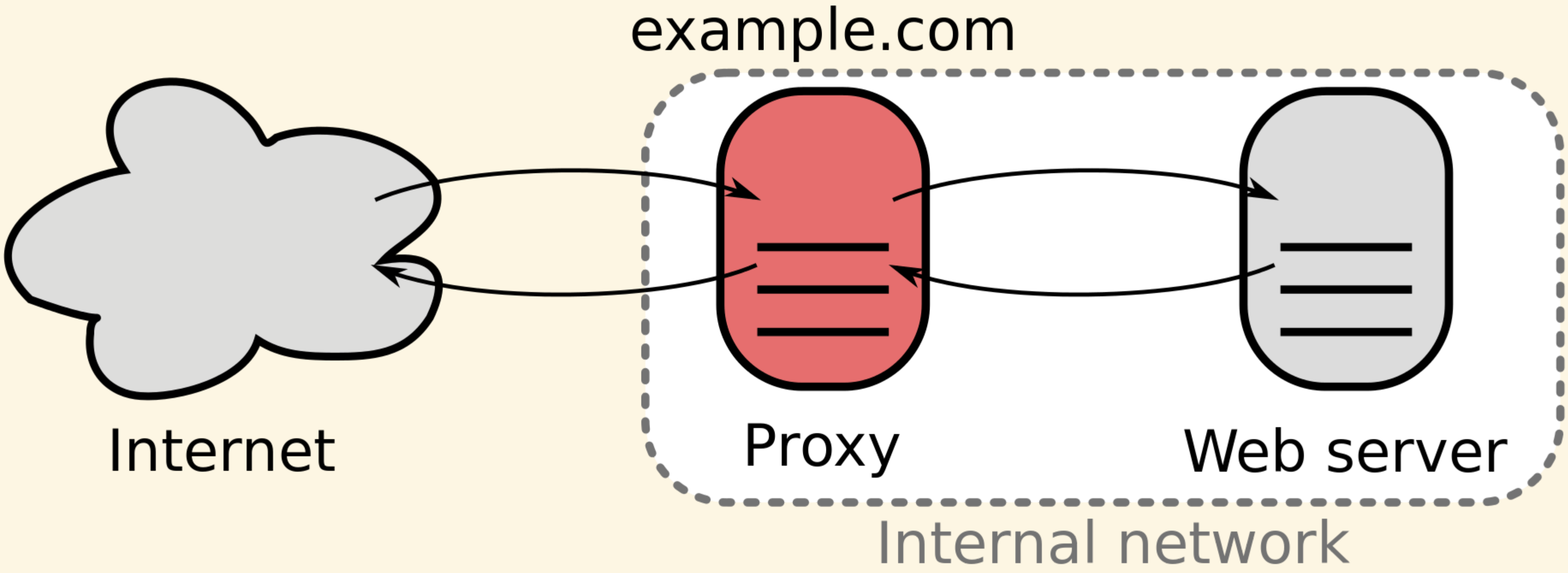


A **reverse proxy** is an **internal-facing proxy** used to control and protect access to servers in a **private network**.

Speaker notes

A reverse proxy **appears to clients to be an ordinary server**, but actually **transmit their requests to** one or more **other servers in an internal private network** which handle the requests.

HIDING INTERNAL ARCHITECTURE



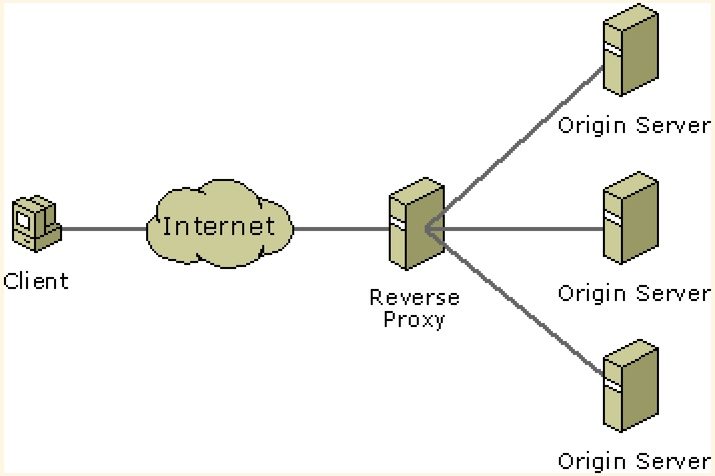
The client only sees the proxy server.

Speaker notes

The response from the private server is returned as if it was coming from the proxy server itself, leaving the client with no knowledge of the structure of the internal network.

WHY USE A REVERSE PROXY?

HIDING INTERNAL ARCHITECTURE



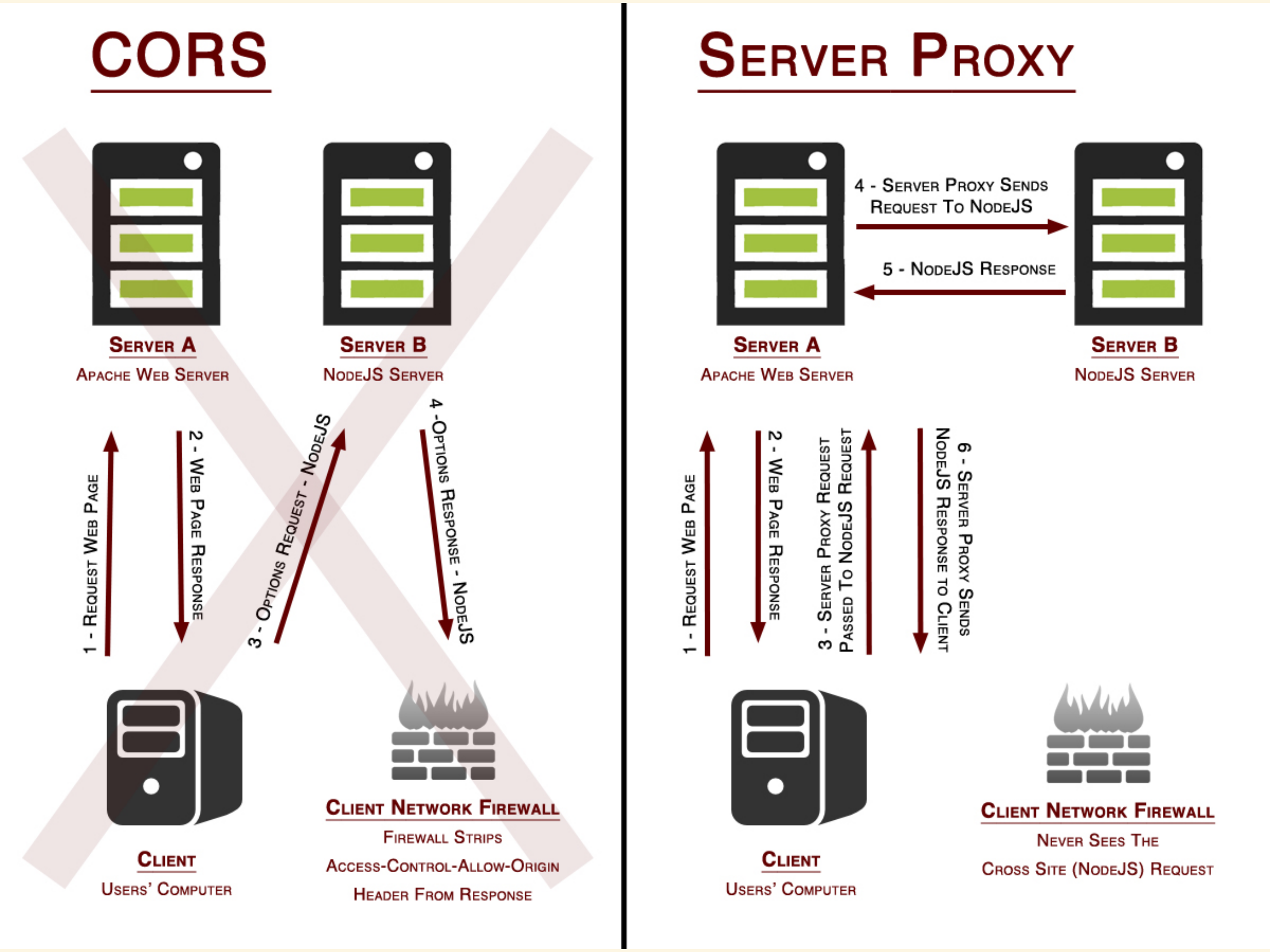
Reverse proxies can **hide the existence and characteristics of an internal network's private servers**.

Speaker notes

Since the client only sees the proxy server, it is unaware of the complexity of the internal architecture and does not have to worry about it.

In a scenario where you have only a single public IP address available, a reverse proxy allows you to make multiple private servers accessible on that IP address through the proxy server.

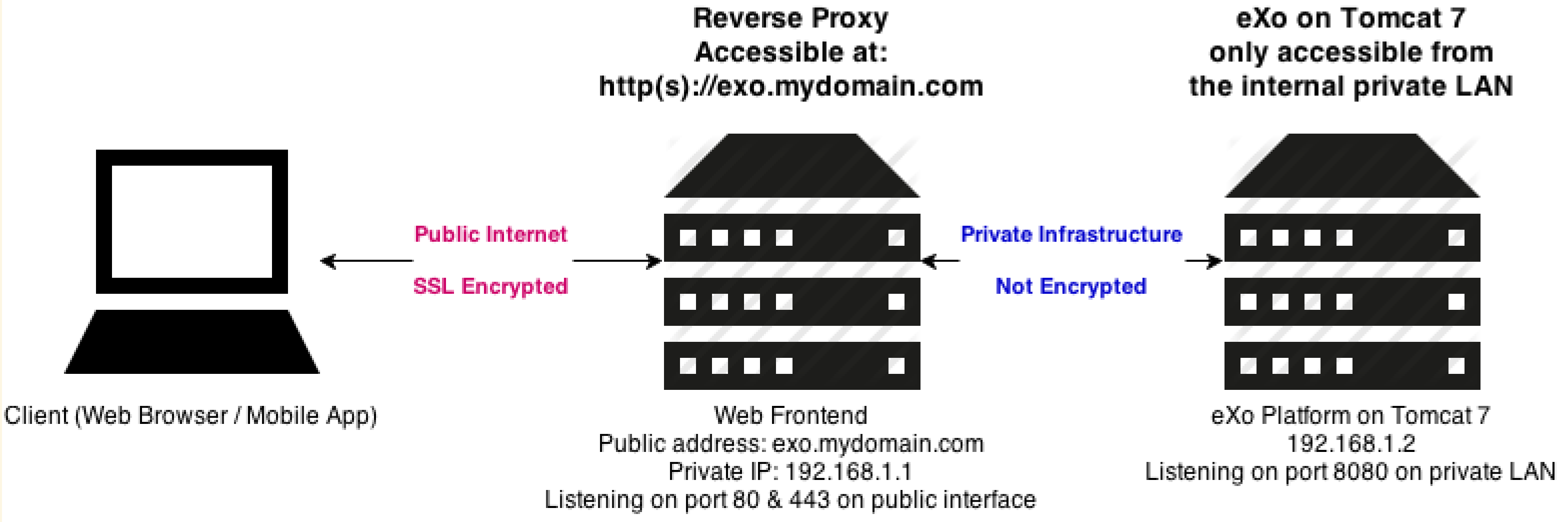
HIDING MULTI-COMPONENT WEBSITES



Speaker notes

Modern websites can be complex applications, often with a **separate frontend and backend** developed by different teams with different technologies. Putting a reverse proxy in front can make it **appear as one single website** on a single domain name, avoiding CORS issues.

SSL TERMINATION OR AUTHENTICATION



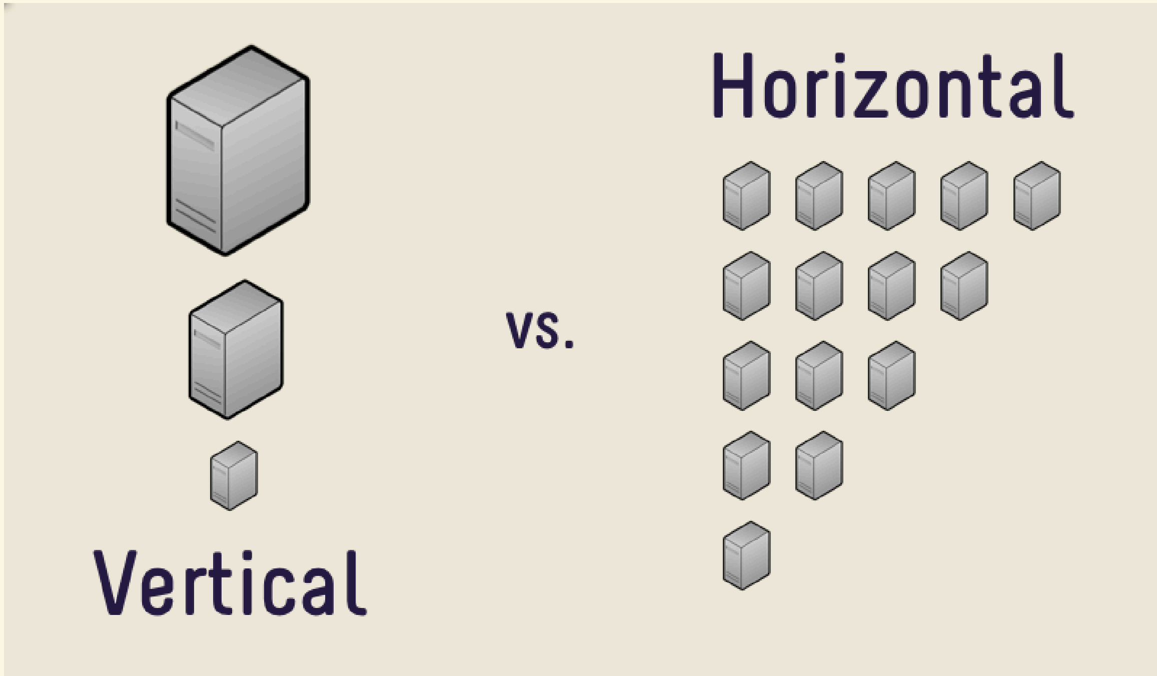
A reverse proxy can be the **secure endpoint** with all the SSL certificates, then **forwarding unencrypted requests** to other servers in the private network.

Speaker notes

Managing SSL certificates to provide websites over HTTPS is rather complex. It can be hard to configure some frameworks or tools to ensure they are only using secure communications.

Similarly, a reverse proxy could also require **authentication** before letting a client access an insecure application, adding security without having to modify the application itself.

SCALABILITY



Scalability is the capability of a computer system to handle a growing amount of work (e.g. client requests).

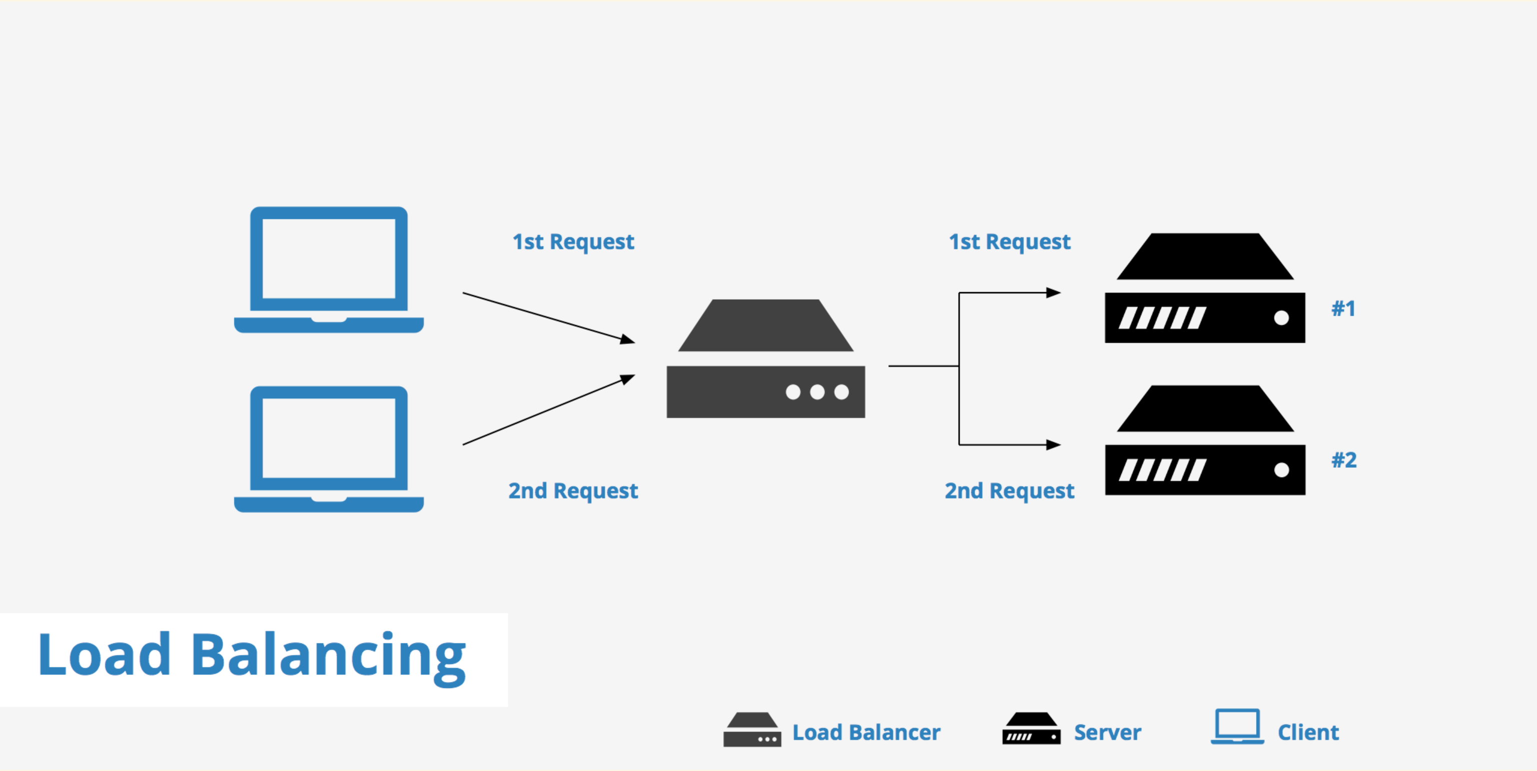
Speaker notes

There are 2 broad ways of adding more resources for a particular application.

Vertical scaling consists in using a more powerful computer, with more CPU, RAM, throughput, etc. The added power will allow the server to serve more clients.

Horizontal scaling means adding more computers to handle the same work. For example, 3 instances of a web application can probably handle 3 times as many clients at the same time. Computers or applications can be combined in clusters to improve performance.

LOAD BALANCING



A common function of reverse proxies is to perform **load balancing**, i.e. the distribution of workloads across multiple servers to achieve **horizontal scaling**.

Speaker notes

As multiple clients arrive simultaneously, the reverse proxy will distribute requests to different servers, spreading the load between them.

OTHER USES

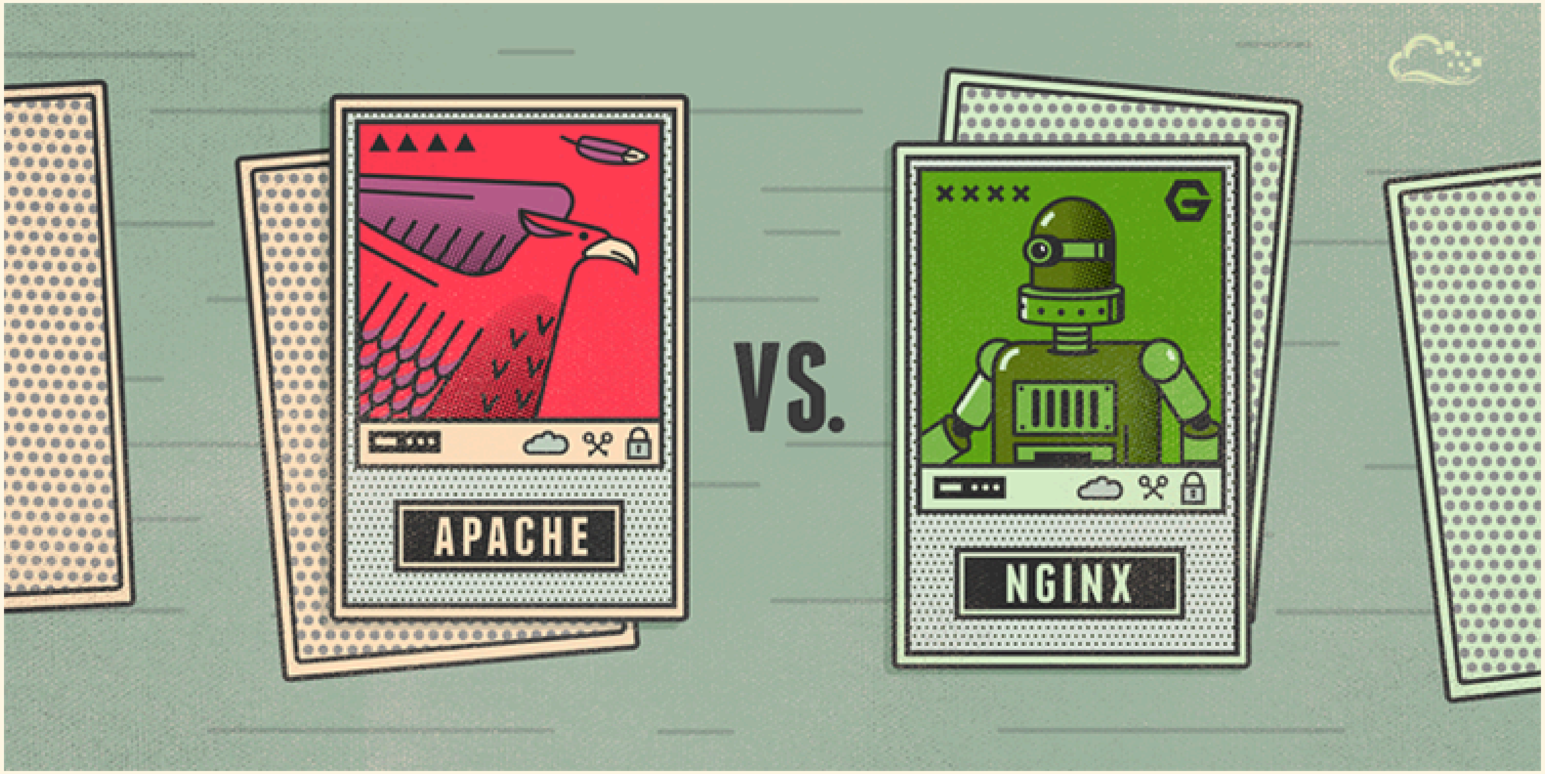
- Web acceleration
- On-the-fly compression
- Spoon-feeding
- Denial-of-service (DoS) protection
- A/B testing

Speaker notes

- Web acceleration is the caching of static content and dynamic content to reduce load on internal servers.
- Optimize content by transparently compressing it to speed up loading times.
- *Spoon-feeding* is a technique where the reverse proxy temporarily stores a dynamically generated page, then serves it to the client a little bit at a time. This avoids the internal server having to wait for slow clients such as mobile applications.
- And more

NGINX

WHAT IS NGINX?

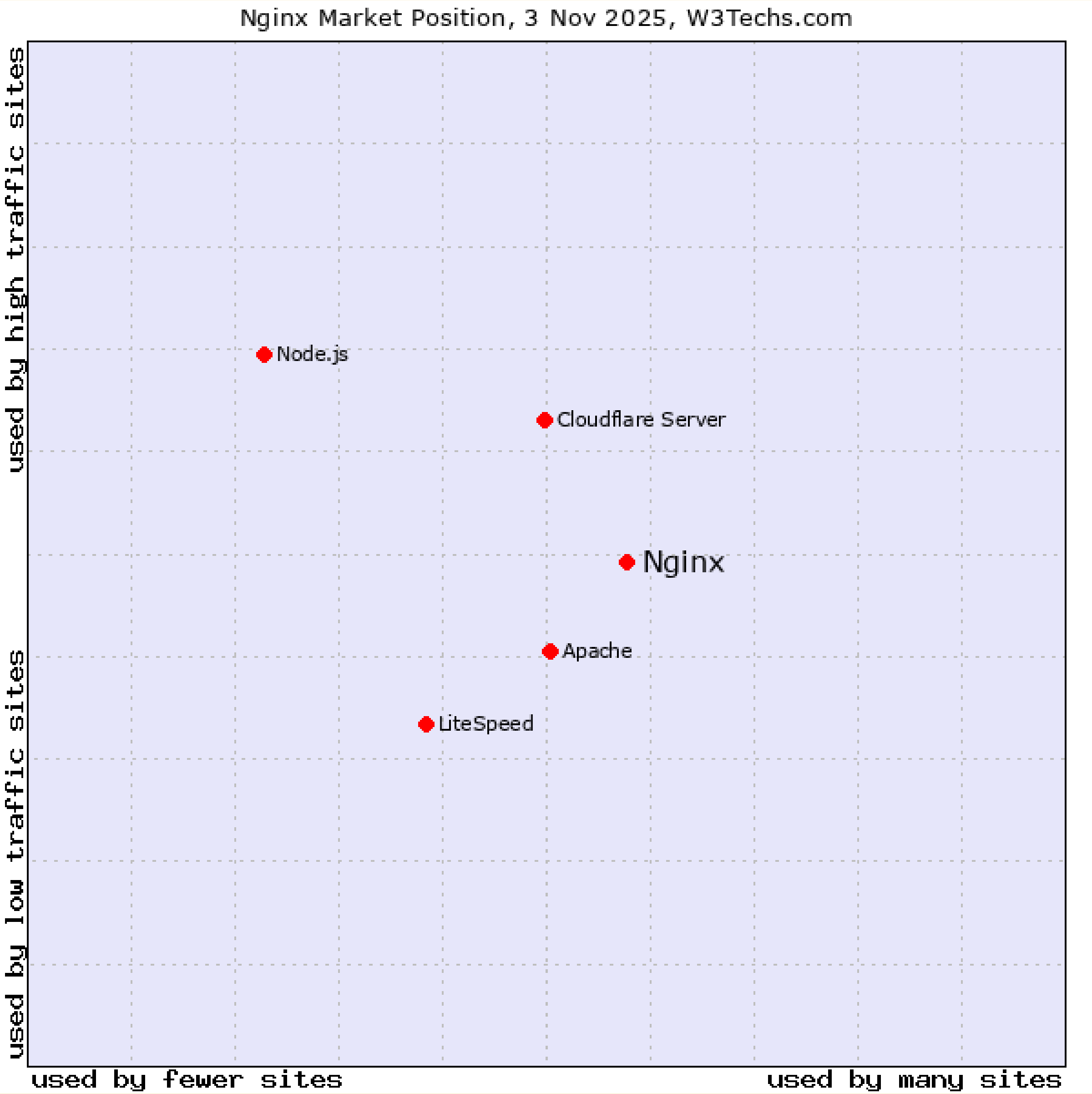


nginx is a concurrent of the well-known Apache HTTP server developed in 2004 to solve the C10k problem.

Speaker notes

nginx is an HTTP and reverse proxy server used by more than 25% of the busiest sites in December 2018. It was developed to solve the C10k problem, i.e. the capability of a computer system to handle ten thousand concurrent connections, thanks to its event-driven architecture. It also has many other features to serve modern web applications.

APACHE VS. NGINX


Nginx Market Position, 3 Nov 2025, W3Techs.com
used by high traffic sites
used by low traffic sites
used by fewer sites
used by many sites
Node.js
Cloudflare Server
Nginx
Apache
LiteSpeed

Speaker notes

Although Apache is still used to serve more websites, nginx leads the pack in web performance, and is used more for the busiest websites.