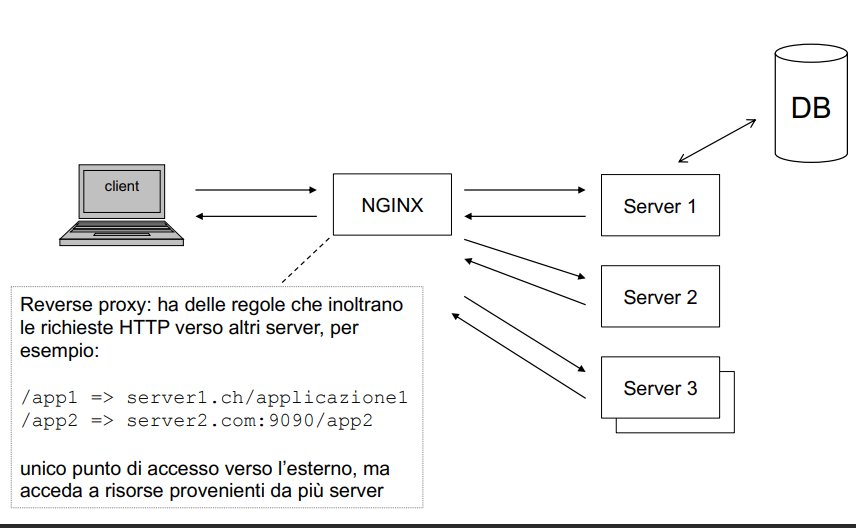
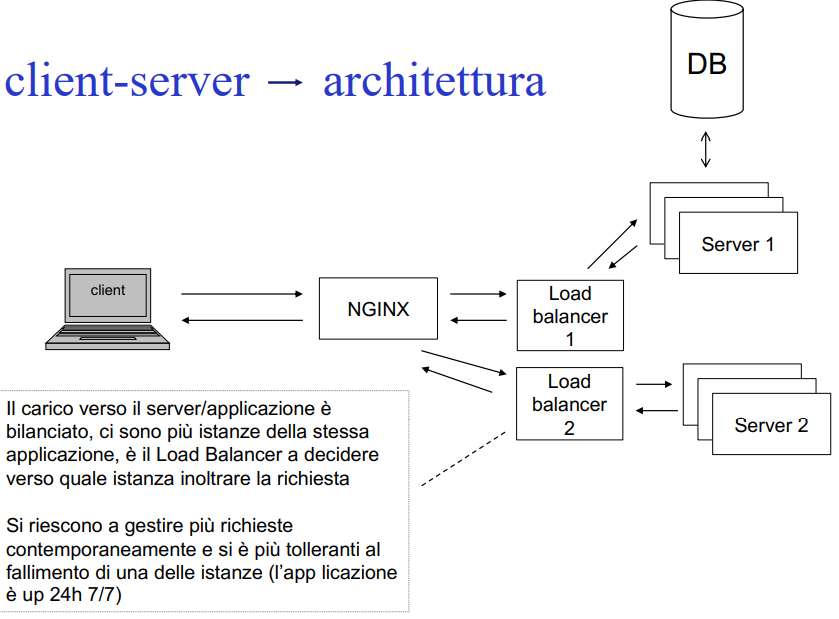
**Architetture web**

Il web funziona principalmente con architettura client-server

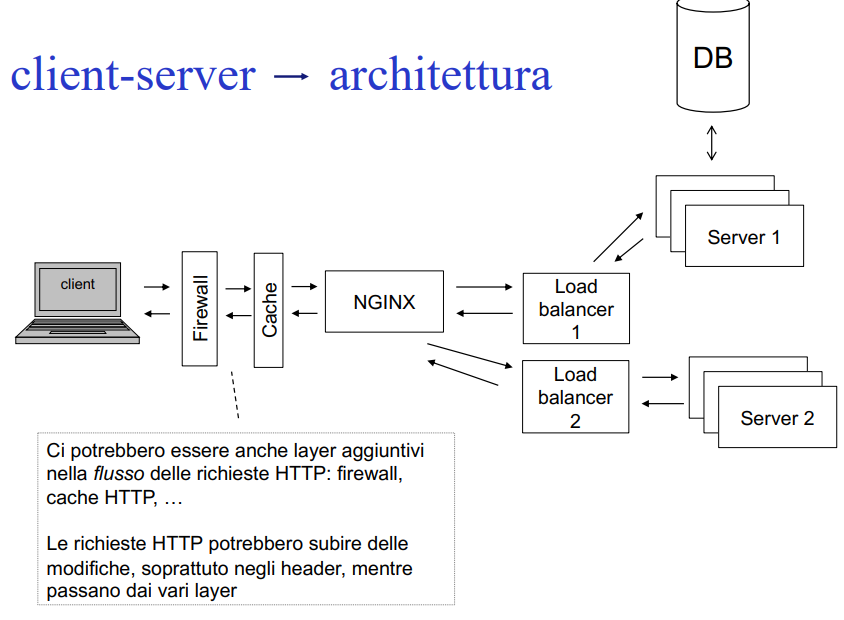
Client(request)🡺Server | Client🡸(response) Server



Con un reverse proxy come nginx si possono avere più server che forniscono un servizio

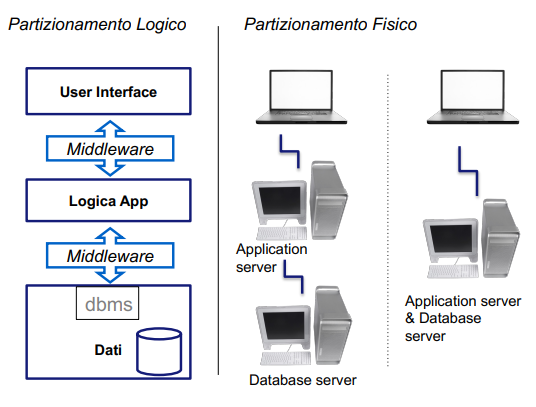


Il load balancer in base alle richieste bilancia il carico sui server, round robbin è la soluzione più semplice.



Quest’ultima è l’architettura più completa di una web app

Le architetture web sono a N livelli (tier) ed hanno 3 tipi separati di funzionalità

* D :Gestione dei dati (Data layer) 🡺 gestisce e accede ai dati al database
* L :Logica di applicazione (Business layer) 🡺 gestisce le richieste dell’utente e determina a quali dati accedere e quale logica applicare per fornire le risposte
* P: Presentazione (Presentation layer) 🡺 parte che consente all’utente di interagire (front-end)

Middleware sono le componenti (interfacce) che fanno comunicare i vari livelli, garantisce il disaccoppiamento, permette lo sviluppo indipendente per ogni componente, semplifica la riconfigurazione del sistema.

L’applicazione per l’utente è vista come una unica.

**Vantaggi di un’architettura N-tier**

* Scalabilità
* Manutenzione
* Riusabilità
* Sicurezza
* Tolleranza ai guasti

**Svantaggi**

* Complessità
* Tracciabilità
* Latenza