

Architetture N-tier

Con **multi-tier** o architettura multi-strato (spesso definita con l'espressione inglese **n-tier architecture**) si intende un'architettura software di tipo **client-server** per sistemi distribuiti, in cui le varie funzionalità del software sono logicamente separate ovvero suddivise su più strati o livelli software differenti in comunicazione tra loro.

1-tier

L'architettura **1-tier** è un modello in cui tutte le funzionalità del software risiedono su un **unico livello** o strato. In altre parole, tutte le componenti del sistema, inclusa l'interfaccia utente, la logica di business e il database, sono integrate in un'unica applicazione o processo.

Questa è spesso considerata la forma più **semplice** di architettura software ed è adatta per applicazioni di piccole dimensioni o prototipi in cui la complessità e i requisiti di scalabilità sono limitati. Tuttavia, questa architettura può diventare difficile da gestire e mantenere in grandi progetti o in sistemi che richiedono aggiornamenti frequenti o modifiche.

Quest'architettura comporta infatti problemi di:

- **Scalabilità**
- **Portabilità**
- **Aggiornamento**
- **Flessibilità**

2-tiers

L'architettura **2-tier** (modello Client-Server) è un modello in cui l'applicazione è suddivisa in due livelli principali:

- Livello di **presentazione** (o interfaccia utente, ovvero il client): Questo è il livello con cui l'utente interagisce direttamente. Include l'interfaccia utente grafica o text-based (GUI o CLI), che consente agli utenti di interagire con l'applicazione. Questo livello gestisce l'input dell'utente e visualizza i risultati.
- Livello di **dati** (o backend, ovvero il server): Questo livello è responsabile della gestione dei dati e della logica di business dell'applicazione. Include il database in cui vengono memorizzati e recuperati i dati, nonché la logica di elaborazione dei dati, come la validazione dei dati, le regole di business e altre elaborazioni.

Il client richiede i servizi al server, comunicando **direttamente** con esso senza nessun'altra entità intermedia.

L'architettura 2-tier è una scelta comune per molte applicazioni, poiché offre un buon compromesso tra semplicità, scalabilità e manutenibilità.

3-tiers

L'architettura a 3 livelli, o **3-tier**, è un modello di progettazione software che suddivide l'applicazione in tre livelli distinti:

- **Primo livello** (client). La responsabilità principale di questo livello è quella di ricevere gli eventi degli utenti, controllare l'interfaccia utente e rappresentare i dati sull'interfaccia.
- **Secondo livello** (application/proxy server). La parte complessa della logica dell'applicazione è all'interno di questo livello ed è disponibile per il livello client, quando questo la richiede. Tale livello costituisce l'entità centrale per la soluzione dei problemi presenti nel modello a due livelli, anche proteggendo l'accesso diretto di dati.
- **Terzo livello** (core server). Questo livello è responsabile della memorizzazione dei dati, spesso tramite un File Server e un gestore di basi di dati.

Vantaggi (rispetto all'architettura 1-tier o 2-tiers):

- **Sviluppo più rapido:** ciascun tier può essere sviluppato contemporaneamente da team differenti, velocizzando lo sviluppo dell'applicazione.
- **Scalabilità migliorata:** qualsiasi tier può essere scalato indipendentemente dagli altri come necessario.
- **Affidabilità migliorata:** la probabilità che un'interruzione in un tier abbia impatto sulle prestazioni degli altri tier sarà minore.
- **Sicurezza migliorata:** poiché il tier presentazione e il tier dati non possono comunicare direttamente, un tier applicazione ben progettato può funzionare come una sorta di firewall interno, impedendo attacchi di vario tipo.