

ARCHITETTURE WEB N-TIER BASATE SU CLIENT-SERVER

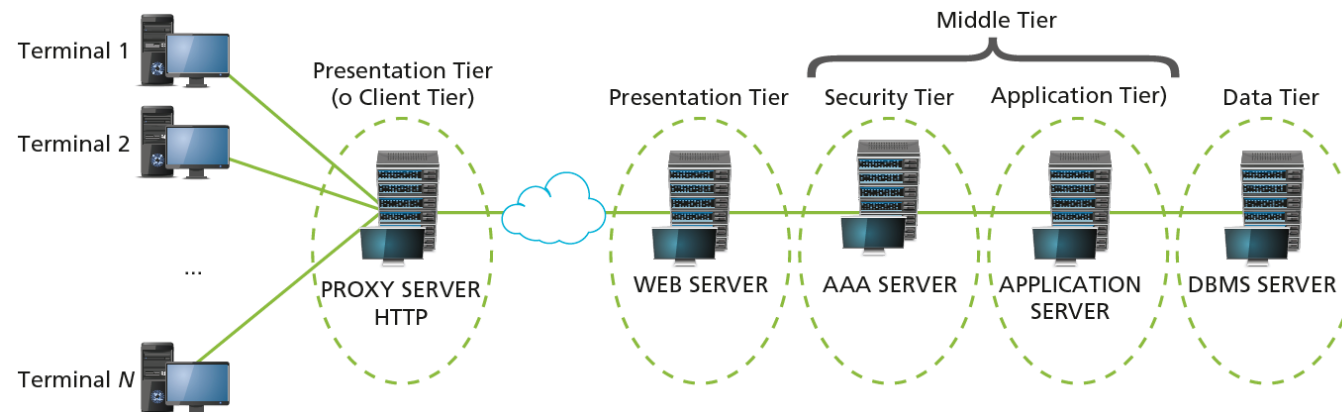
Sistemi e reti

1



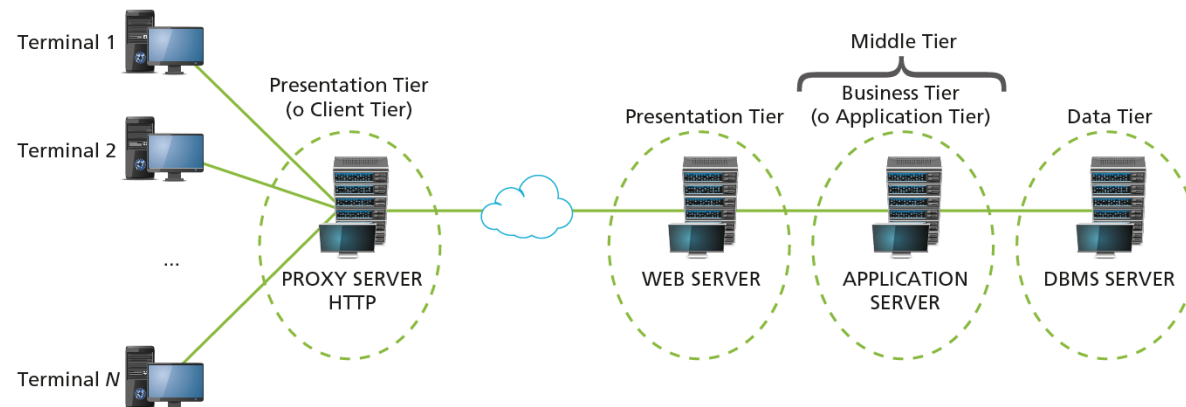
ARCHITETTURE N-TIER

- Nell'ambito dei sistemi per il web, le architetture vengono, come spesso accade in informatica, modellizzate a livelli (tier).
- Un'architettura N-Tier è un sistema modellizzato utilizzando N strati.
- Vantaggio: è possibile intervenire su un livello senza che questo condizioni i livelli adiacenti.



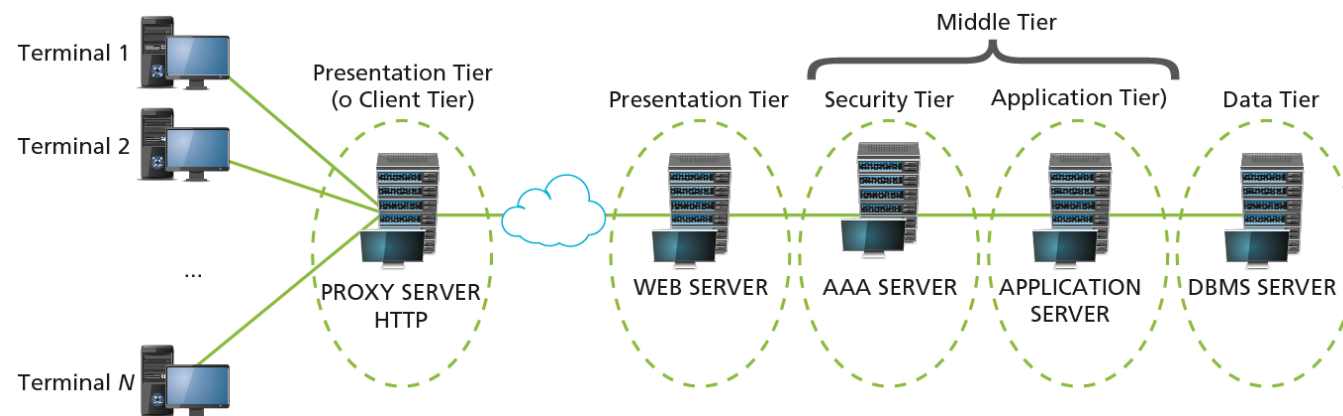
I TIER PRINCIPALI

- I tier principali in un'architettura web lato server sono tre:
 - **Presentation (o client) Tier:** gestisce le modalità di comunicazione coi client. Rappresenta il front-end dell'applicazione web;
 - **Business Tier:** elabora le richieste dei client e produce i risultati da inoltrare come risposta;
 - **Data Tier:** gestisce l'accesso ai dati necessari alle applicazioni web.



ALTRI TIER COMUNI

- Le architetture attuali sono spesso progettate con ulteriori tier, i più comuni sono:
 - **Application Tier:** gestisce le applicazioni web;
 - **Security Tier:** esegue il monitoraggio e la gestione degli accessi alla rete;
 - **Networking Tier:** garantisce la connettività e la privacy del client;
 - **Data access Tier:** gestisce le autorizzazioni per i dati sensibili;
 - **Data storage Tier:** esegue le operazioni sul database.



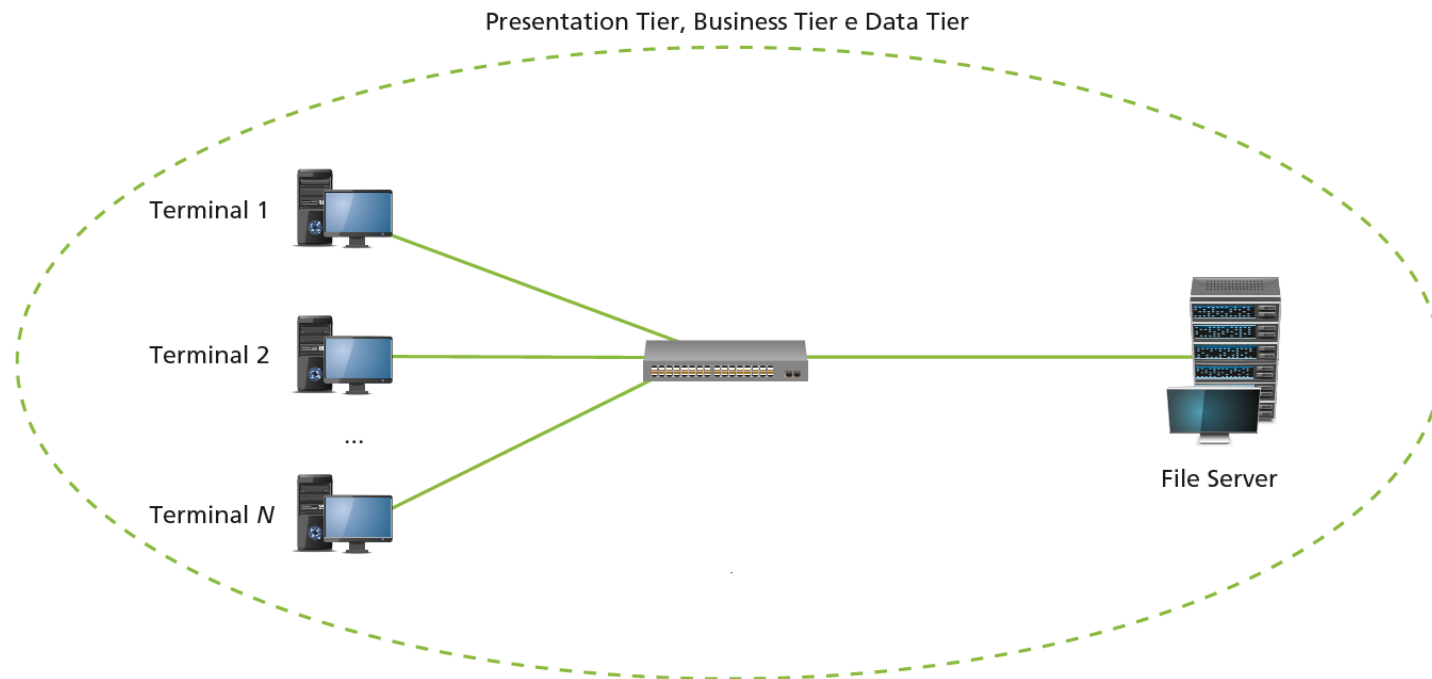
COME SI REALIZZANO I TIER

- I tier vengono realizzati mediante macchine server che svolgono il compito loro assegnato.
- I server principalmente utilizzati sono:
 - **Web Server**: realizza le funzioni di **Presentation Tier**;
 - **Application Server**: realizza le funzioni di **Application Tier**, talvolta coincidenti con quelle di Business Tier;
 - **DBMS Server**: realizza funzioni di **Data Tier**, talvolta le funzioni sono suddivise tra Data access Tier e Data storage Tier;
 - **Proxy Server**: realizza funzioni di **Presentation Tier**, ma anche di Security e Networking Tier;
 - **AAA Server**: realizza funzioni di **Security Tier**.



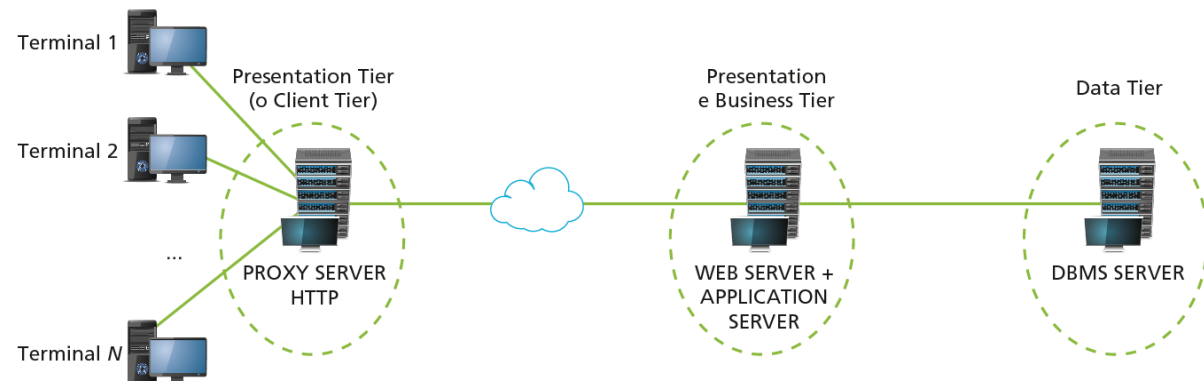
ARCHITETTURA 1-TIER

- Tutte le componenti fanno parte di un unico tier.
- I client sono dei semplici terminali che visualizzano i dati elaborati dal mainframe.
- Problemi di:
 - scalabilità;
 - portabilità;
 - aggiornamento;
 - flessibilità.



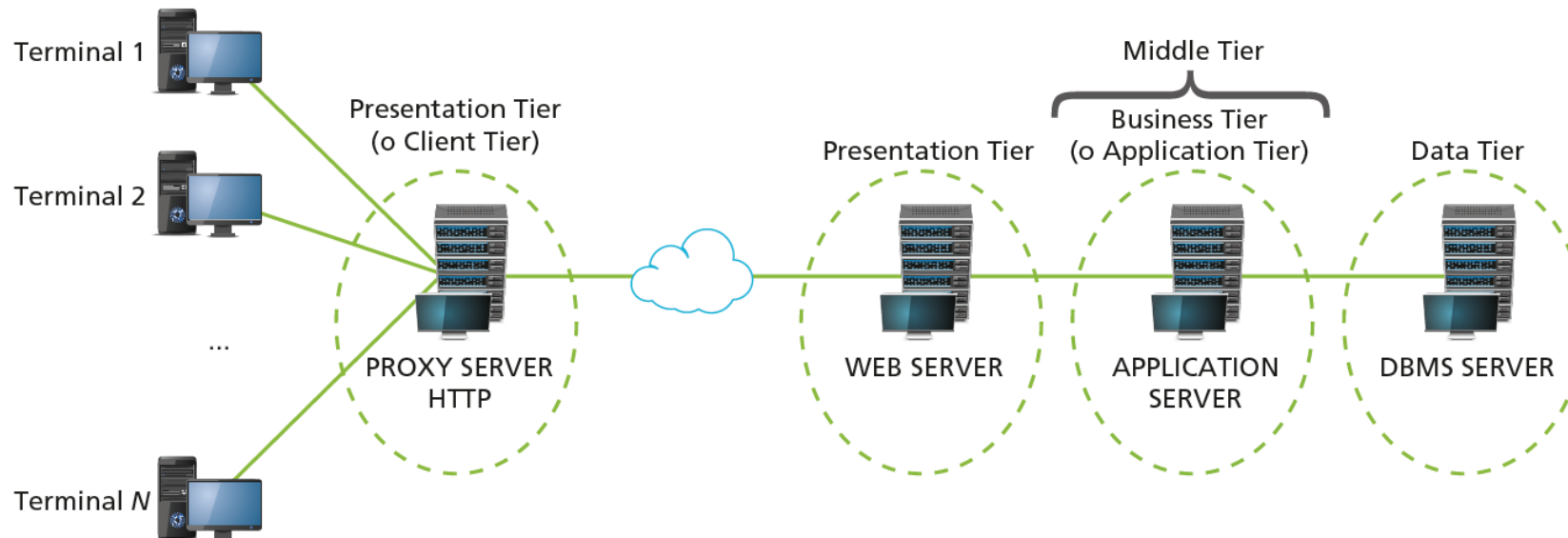
ARCHITETTURA 2-TIER

- E' il modello client-server che è diventato il paradigma di internet.
- Le applicazioni ed i dati possono risiedere in remoto ed essere collocate su macchine diverse.
- I client non sono più dei terminali, ma dei veri e propri host in grado di elaborare i dati ed effettuare richieste ai server.
- Presenta ancora molte delle problematiche dell'architettura 1-tier.
- Rende indipendente il Data Tier.



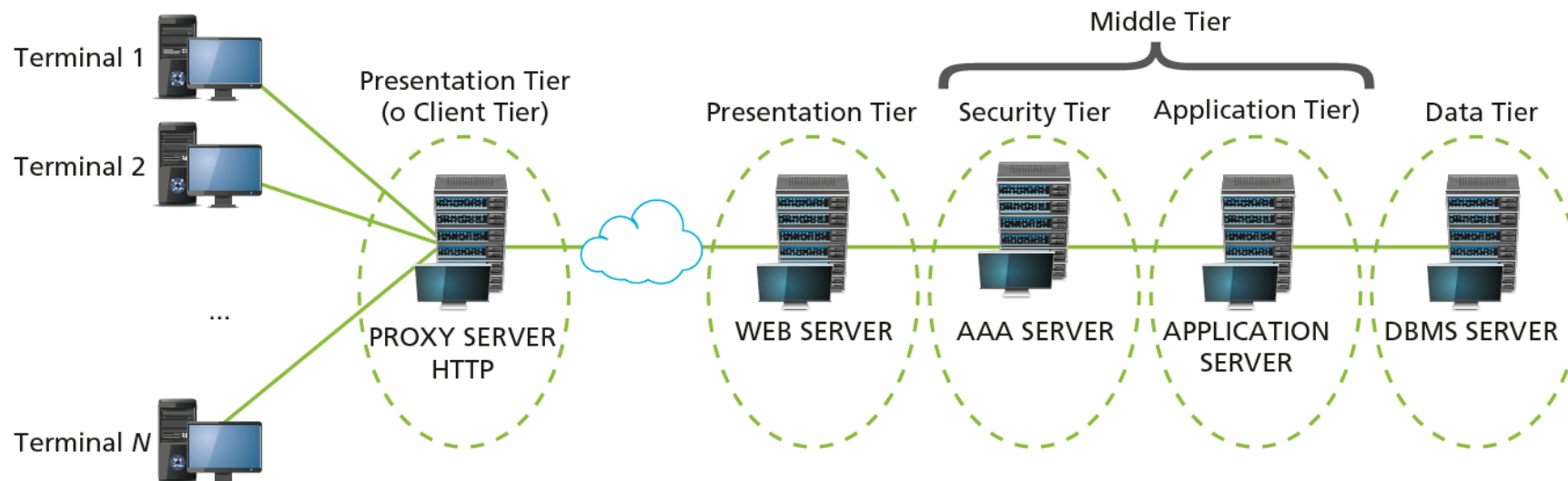
ARCHITETTURA 3-TIER

- Viene aggiunto un tier intermedio (il Business Tier) che, pur mantenendo il paradigma client-server, consente di rendere completamente indipendenti i tre tier principali.
- Consente di superare le problematiche segnalate nei due casi precedenti.



ARCHITETTURA N-TIER

- Prevedono un'ulteriore suddivisione interna al Business Tier.
- Consentono di migliorare lo sviluppo delle applicazioni web in un contesto di sicurezza.



SCALE-OUT VS SCALE-UP

- La **scalabilità** è la capacità di un sistema di aumentare o diminuire di scala in funzione delle necessità e delle disponibilità.
- Quanto un determinato sistema sia scalabile dipende dalla sua architettura software e/o hardware.
- In questo contesto si può definire una:
 - **scalabilità orizzontale (scale-out)**: si ottiene con un aumento/diminuzione del numero dei nodi che consentono al sistema di gestire in parallelo il carico di lavoro;
 - **scalabilità verticale (scale-up)**: si ottiene aumentando/diminuendo le risorse di un singolo nodo.

