

## Unità D1

### Architetture di rete

### Obiettivi

- Comprendere il concetto di architettura di rete
- Conoscere l'architettura stand-alone (vantaggi, difetti e contesti applicativi)
- Conoscere l'architettura terminal server, comprenderne il funzionamento e l'importanza nella storia dell'informatica
- Conoscere l'architettura client/server, la più utilizzata e diffusa
- Conoscere l'architettura file server, e la sua importanza nella gestione delle informazioni aziendali

### Concetti generali

- Un'**architettura di rete** è un modello adottato da una particolare configurazione di sistemi.
  - architettura **stand-alone**;
  - architettura **terminal server**;
  - architettura **client/server**.
  - architettura **file server**.
- In una rete di calcolatori possono coesistere varie architetture.

### Architettura stand-alone

- L'**architettura stand-alone** è la più semplice e comprende una sola macchina, che spesso in inglese viene denominata *workstation*.
- La singola macchina racchiude in sé tutte le funzionalità cui essa è preposta (non dipende da altre macchine per servizi o risorse di cui non dispone).
- In genere architetture di questo tipo sono utilizzate in contesti in cui i servizi richiesti sono molto specifici.
- Alcuni ambiti applicativi in cui possono essere adeguatamente utilizzate le architetture stand-alone sono il multimedia, il CAD/CAM o il semplice utilizzo domestico.

### Un esempio di workstation



### Nota

- Sebbene le workstation siano sistemi sostanzialmente isolati, sarebbe sbagliato pensare che un'architettura stand-alone non possa esistere in un ambiente di rete.
- In molti contesti lavorativi può capitare di trovare workstation che condividono la connessione a Internet o che magari utilizzano la rete per la condivisione di file.
- Parliamo ancora di architettura stand-alone poiché la rete **non è un mezzo indispensabile** allo svolgimento dei compiti cui il sistema è preposto.

## L'architettura stand-alone nei database

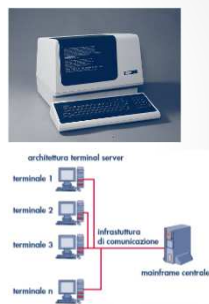
- Nel contesto dei database, l'architettura **stand-alone** è implementata mediante sistemi in grado di gestire insiemi di dati di **piccole dimensioni**, solitamente a uso personale.
- Un database stand-alone risiede interamente sulla macchina che lo ospita e garantisce l'accesso ai dati al solo utente della workstation.
- Un esempio di questo tipo di sistema è Access di Microsoft.

## DBMS stand-alone

- Da un punto di vista strutturale, i database stand-alone si compongono di due elementi:
  - la logica di funzionamento del database;
  - i dati.
- Nel caso di Access, la logica è contenuta interamente nell'applicazione che viene installata sulla workstation, mentre i dati sono contenuti in uno o più file (con estensione .mdb), memorizzati sul disco fisso.
- Le funzionalità di questi database sono limitate e non risultano quindi adatte a un utilizzo professionale.
- Il vantaggio è un costo contenuto.

## Architettura terminal server

- L'architettura **terminal server** è l'opposto dell'architettura stand-alone.
- Le risorse (dati, risorse di calcolo, software) sono interamente concentrate in una macchina centrale.
- L'utente ha accesso a queste risorse per mezzo di un terminale che non dispone di alcuna risorsa e si comporta esclusivamente da "tramite" tra l'utente e il mainframe.

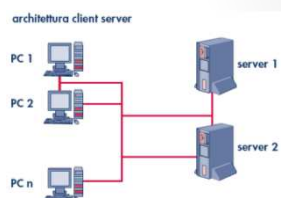


## L'architettura terminal server nei database

- Nei database, l'architettura terminal server è stata ed è tuttora sfruttata ampiamente.
- Il vantaggio è la sua natura centralizzata.
- I dati e le procedure di gestione del database si trovano sul mainframe e gli utenti, tramite i terminali, possono effettuare interrogazioni o immettere nuove informazioni.
- Estremamente diffusa in passato (banche, pubbliche amministrazioni)
- Esistono tuttora soluzioni di questo tipo.
- Le soluzioni disponibili in passato e ancora presenti sul mercato sono molteplici. (DB2 di IBM per AS400, Oracle...).

## Architettura client/server

- "Evoluzione" dell'architettura terminal server.
- Un **server** è una macchina che mette a disposizione uno o più servizi attraverso un'infrastruttura di rete, che funziona da sistema di comunicazione tra il server e i client.
- Un **client** è un sistema hardware o software che usufruisce di uno o più servizi.



## Funzionamento dell'architettura client/server

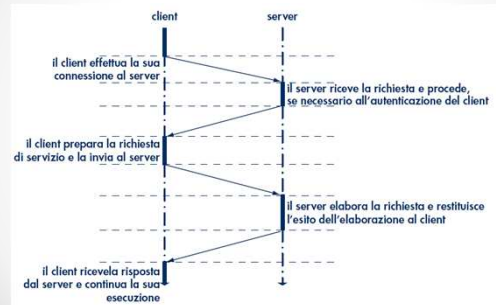
- Le fasi della comunicazione tra client e server sono in genere tre:
  1. instaurazione di un **contatto**;
  2. **richiesta** del client;
  3. **risposta** da parte del server.
- Nella prima fase, si stabilisce un canale di comunicazione tra client e server che può prevedere un processo di **autenticazione** per garantire la sicurezza.
- Una volta stabilita la connessione, il client effettua la **richiesta** del servizio e rimane in **attesa** di una risposta da parte del server.
- Il server riceve la richiesta del client, determina se è in grado, o meno, di evaderla e "risponde" al client con il servizio richiesto o con un eventuale messaggio d'errore.

## Multitier



- Il server può a sua volta necessitare di altri server per soddisfare la richiesta.
- Esempio: web server che richiede un servizio a un database server.

## Le fasi



## Terminal server e client/server

	Terminal server	Clientserver
Funzionalità	Difficilmente ampliabili.	Facilmente ampliabili sia tramite l'installazione di software sia affiancando un server dedicato.
Aumento della potenza di calcolo	Difficile e costosa, a causa della scarsa scalabilità dei mainframe.	Più semplice, i server di nuova generazione sono facilmente potenziabili.
Costi di manutenzione	Elevati sia in termini di effettive ore di manodopera sia di ore di produttività perse.	Contenuti.

© 2007 SEI-Società Editrice Internazionale, Apogeo

## File Server

- Per l'organizzazione dei file in un ambiente di rete esistono due paradigmi:
- **Dispersione:** i file sono memorizzati nelle postazioni degli utenti
  - Vantaggi: semplicità di gestione, facile reperibilità
  - Svantaggi: scarsa disponibilità se computer non raggiungibile, problematico backup generale
- **Centralizzazione:** i file sono memorizzati nel server (file server)
  - Vantaggi svantaggi speculari. Necessaria "politica" di gestione diritti di accesso.
- Il server può essere un computer (dedicato o meno) o un Network Attached Storage (NAS), cioè un apparecchio specificatamente studiato e costruito allo scopo.

## NAS



- Un Network Attached Storage (NAS) è un dispositivo collegato ad una rete di computer la cui funzione è quella di condividere tra gli utenti della rete una Area di memoria.
- I NAS sono generalmente, ma non necessariamente, dei computer ridotti con a bordo il minimo necessario per poter comunicare via rete. Oggi giorno i NAS più diffusi sono in pratica dei PC dotati solitamente di sistema operativo Linux (comunque non visibile dall'utente), e numerosi hard disk per l'immagazzinamento dati. Questa architettura ha il vantaggio di rendere disponibili i file contemporaneamente su diverse piattaforme come ad esempio Linux, Windows e Unix (o Mac OS X), in quanto il sistema operativo implementa i server di rete per tutti gli standard più diffusi quali ad esempio FTP, Network File System (NFS) e Samba per le reti Windows.

Wikipedia

## RAID



- Un Redundant Array of Independent Disks ("insieme ridondante di dischi indipendenti", RAID) è un sistema informatico che usa un insieme di dischi rigidi per condividere o replicare le informazioni.
- I benefici del RAID sono di aumentare l'integrità dei dati, la tolleranza ai guasti e le prestazioni, rispetto all'uso di un disco singolo.
- RAID 1 (Mirroring)
  - Il sistema RAID 1 crea una copia esatta (mirror) di tutti i dati su due o più dischi.
  - È utile nei casi in cui la ridondanza è più importante che usare tutti i dischi alla loro massima capacità; infatti il sistema può avere una capacità massima pari a quella del disco più piccolo. Poiché ogni disco può essere gestito autonomamente nel caso l'altro si guasti, l'affidabilità aumenta linearmente al numero di dischi presenti. RAID-1 aumenta anche le prestazioni in lettura, visto che molte implementazioni possono leggere da un disco mentre l'altro è occupato.

Wikipedia