



UNIVERSITÀ  
degli STUDI  
di CATANIA



**Docente: Fabrizio Messina**

# SISTEMI CENTRALI - LEZIONE 1

## PRIMA PARTE

- Obiettivi del corso
- Il piano del corso
- Modalità di esame

## SECONDA PARTE

- Modelli Elaborativi:
  1. Centralizzati
  2. Client server
  3. Distribuiti
  4. in Cloud



UNIVERSITÀ  
degli STUDI  
di CATANIA





UNIVERSITÀ  
degli STUDI  
di CATANIA



# PRIMA PARTE

# SISTEMI CENTRALI - OBIETTIVI DEL CORSO

Obiettivi del corso:

- 1) Trattare le **principali architetture di elaborazione** correntemente in uso nei sistemi di Information Technology.
- 2) Comprendere i **requisiti del sistema** tra cui **disponibilità e carico di lavoro**, e **relative soluzioni architetture e tecnologiche**.
- 3) Presentare i criteri generali per
  - **dimensionamento** dell'hardware
  - **definizione della topologia** del sistema
- 4) I concetti della **virtualizzazione**
- 5) Le architetture ed i servizi in **Cloud**



UNIVERSITÀ  
degli STUDI  
di CATANIA



# SISTEMI CENTRALI - PIANO DEL CORSO

## 1. Modelli Elaborativi

- Centralizzati
- Client server
- Distribuiti
- in Cloud

## 2. Architettura a tre livelli



UNIVERSITÀ  
degli STUDI  
di CATANIA



## SISTEMI CENTRALI - PIANO DEL CORSO

### 3. Continuità Operativa e Scalabilità

- **Requisito:** Alta affidabilità (High Availability o HA)
  - **Soluzione:** Clusters
- **Requisito:** Scalabilità
  - **Soluzione:** Bilanciatori di carico (load balancing) e politiche di bilanciamento
- **Scalabilità** orizzontale vs scalabilità verticale
- **Ridondanza dei dischi** (RAID)
- Altri punti di vulnerabilità (**single point of failure**)
- **Backup and recovery**
- **Disaster Recovery**



UNIVERSITÀ  
degli STUDI  
di CATANIA



# SISTEMI CENTRALI - PIANO DEL CORSO

## 4. Dimensionamento del sistema e topologia

### PRIMA PARTE: LA NECESSITA' DEL DIMENSIONAMENTO

- Stima del dimensionamento
- La stima per il budget di progetto
- L'acquisto dell'infrastruttura
- **Ciclo di sviluppo: ambienti richiesti**

### SECONDA PARTE: METODOLOGIA DEL DIMENSIONAMENTO

- Requisiti di scalabilità del software
- Stima delle CPU e della memoria RAM
- Contingency e think time
- Throughput del processore
- Il dimensionamento del Database
- Il dimensionamento del file system
- Crescita dei volumi e capacity plan
- Lo stress test



UNIVERSITÀ  
degli STUDI  
di CATANIA



## SISTEMI CENTRALI - PIANO DEL CORSO

### 4. Dimensionamento del sistema e topologia

#### TERZA PARTE: DISEGNO DELLA TOPOLOGIA

- La definizione della topologia
- Fattori che influenzano la topologia
- Un approccio euristico

### 5. Virtualizzazione

#### PRIMA PARTE: COMPRENDERE LA VIRTUALIZZAZIONE

- Il concetto della virtualizzazione
- L'importanza della virtualizzazione
- I trend ed il cloud computing



UNIVERSITÀ  
degli STUDI  
di CATANIA





## SISTEMI CENTRALI - PIANO DEL CORSO

### 5. Virtualizzazione

SECONDA PARTE: VIRTUALIZZAZIONE DEL SISTEMA

- ✓ Hypervisor e virtual machine
- ✓ Gestione CPU, RAM e storage della VM
- ✓ Copia e clone di una VM

TERZA PARTE: APPLICARE LA VIRTUALIZZAZIONE

- ✓ **Disponibilità** del sistema virtualizzato
- ✓ Le **applicazioni** informatiche in ambiente virtualizzato
- ✓ Fornitori di software di virtualizzazione
- ✓ Cenni sui **containers**



UNIVERSITÀ  
degli STUDI  
di CATANIA



## SISTEMI CENTRALI - PIANO DEL CORSO

### 6. Architettura in Cloud

#### PRIMA PARTE: CARATTERISTICHE DEL CLOUD COMPUTING

- ✓ Caratteristiche del cloud computing
- ✓ Modelli di servizi Cloud.
- ✓ Deployment models: benefici e svantaggi

#### SECONDA PARTE: CLOUD COMPUTING PER LE AZIENDE

- ✓ Considerazioni sui costi
- ✓ Dipartimento IT
- ✓ Bilancio e cash flow
- ✓ **Service Level Agreement (SLA)**
- ✓ Elasticità e scalabilità
- ✓ Continuità operativa e DR (Disaster Recovery) in cloud
- ✓ Ambienti e cicli di sviluppo dei sistemi software
- ✓ Update automatici, P2T e T2T
- ✓ Interazione tra ambienti SaaS ed on premise



UNIVERSITÀ  
degli STUDI  
di CATANIA



# SISTEMI CENTRALI - PIANO DEL CORSO

**Lettura consigliata in bibliografia.**

- Sistemi Centrali: i «mainframe»
  - Introduzione e Storia dei Sistemi Centrali
  - Introduzione all' Architettura dei Calcolatori (z/Architecture)



UNIVERSITÀ  
degli STUDI  
di CATANIA



## SISTEMI CENTRALI - MODALITA' ESAME

- 1. Test con domande a risposte multiple, 30 minuti
- +
- 2. Prova orale su argomento a piacere



UNIVERSITÀ  
degli STUDI  
di CATANIA



## SISTEMI CENTRALI - ESEMPIO DI TEST

Come si definisce la situazione di un servizio che non può essere riavviato sullo stesso nodo di un cluster e viene riavviato su un altro nodo?

- A. Scalabilità
- B. Failover
- C. Distribuzione di carico
- D. Data mirroring

La risposta corretta è: B. Failover



UNIVERSITÀ  
degli STUDI  
di CATANIA



# SECONDA PARTE (modelli elaborativi)



UNIVERSITÀ  
degli STUDI  
di CATANIA



# Sistemi centrali

## Sistema centrale o mainframe [1]

- Calcolatore dotato di **elevata capacità elaborativa**
- Gestisce **grandi volumi di dati** con un elevato livello di sicurezza ed affidabilità
- Acceduto da un alto numero di utenti contemporaneamente
- Gestisce un **carico di lavoro misto**, ad esempio 'a lotti' (batch) e transazionale (online)



UNIVERSITÀ  
degli STUDI  
di CATANIA



## Sistemi centrali

- Il sistema centrale e' identificato anche come «mainframe»
- I **terminali**, impiegati per l'accesso alle risorse del mainframe, non dispongono di capacità elaborative
- Assolve tre funzioni fondamentali:
  1. Database server
  2. Application server
  3. Presentation server
- Possibile l'**integrazione in una rete di elaboratori eterogenea**, poichè ad oggi adottano standard IT:
  - TCP/IP, Linux, POSIX, Java, XML

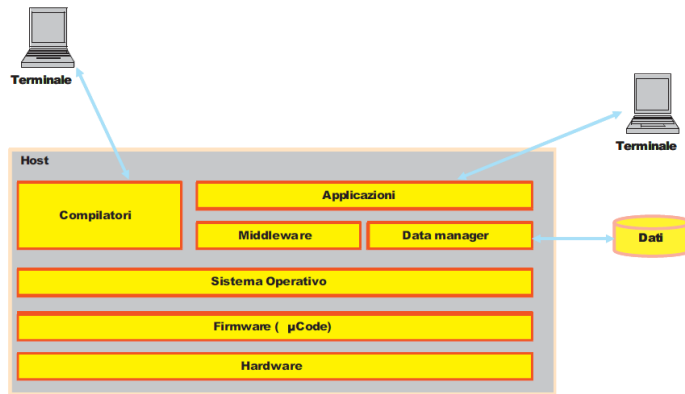


UNIVERSITÀ  
degli STUDI  
di CATANIA





## Sistemi centrali: Mainframe



Modello applicativo del sistema centrale



UNIVERSITÀ  
degli STUDI  
di CATANIA



Si definisce Sistema Centrale un **calcolatore dotato di elevata capacità elaborativa** usato per gestire **grandi volumi di dati** con un elevato livello di sicurezza ed affidabilità. Tale sistema è acceduto da un **alto numero di utenti contemporaneamente**. Esso gestisce un **carico di lavoro misto** eseguendo attività tra loro differenti per caratteristiche e impegno elaborativo.

Queste **attività vengono eseguite in contemporanea** mantenendo **priorità** assegnate ed armonizzando le varie necessità elaborative.

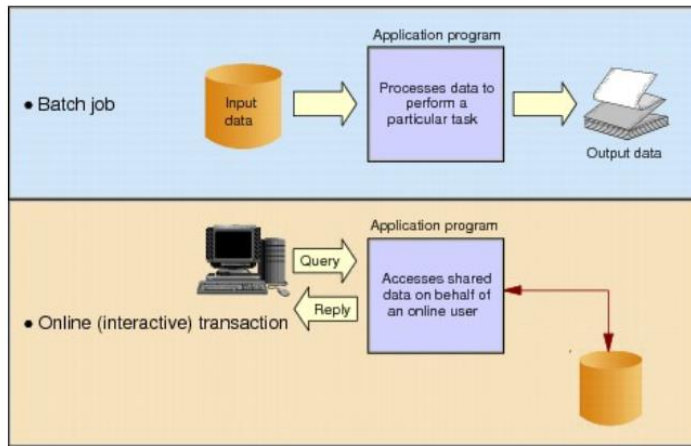
Il Sistema Centrale si contrappone a server dedicati , dedicati cioè svolgere un solo tipo di workload (firewall, router, file server, print server, ecc.).

E' simile ai server applicativi (si vedano slide seguenti) che sono in grado di far girare applicazioni di vario tipo.

**Lo scenario in cui il sistema centrale si inserisce è cambiato nel tempo.** Inizialmente il sistema centrale accentrava su di sé tutte le funzioni informatiche (controllo degli accessi, elaborazioni con calcoli numerici complessi – ad esempio scientifiche – gestione delle reti ed altro).

Oggi il **sistema centrale è collocato in una configurazione di server** in cui una serie di **funzioni sono demandate a server applicativi** poiché questa infrastruttura è quella più efficiente dal punto di vista dei costi. Possiamo dire che **nei sistemi centrali sono rimaste tre funzioni fondamentali: Data base server, Application server e Presentation server.** Questi concetti verranno sviluppati in seguito. Questa cooperazione è stata resa possibile poiché i sistemi centrali hanno implementato tutti gli standard adottati e richiesti dal mercato IT, siano essi de jure o de facto (TCP/IP, Linux, POSIX, Java, XML ecc.); in questo modo viene resa possibile l'integrazione dei sistemi centrali all'interno di una rete di elaboratori eterogenea

## Sistemi centrali: tipi di lavoro



Elaborazione batch (a lotti) e transazionale (online)



UNIVERSITÀ  
degli STUDI  
di CATANIA



Questi concetti saranno sviluppati ulteriormente nella parte dedicata ai mainframe. Si veda [1] per un riassunto pag 20 e segg.

**1. Batch job.** Nell'elaborazione batch si ha la sovrapposizione completa nel tempo tra le operazioni di input/output (I/O) con l'elaborazione vera e propria. I programmi sorgente vengono posti in una coda, prima dell'elaborazione. Nella pratica, sono applicazioni che girano senza interazione utente. Un batch job è sottmessso sul computer, legge e scrive quantità elevatissime di dati – anche TeraByte e produce report in output, quali:

- 1.1 situazioni finanziarie
- 1.2 estratti conto bancari/carte di credito
- 1.3 fatture

Le caratteristiche dell'elaborazione batch sono:

- a. Lettura e scrittura di **grandi quantità di dati**
- b. La risposta non deve essere contestuale o sincrona alla richiesta. Tuttavia devono poter girare in una finestra temporale limitata, quindi **l'elaborazione parallela è un requisito fondamentale**, anche in contemporanea ad elaborazioni online, senza interazioni reciproche negative
- c. Le informazioni sono generate a partire da un **grande numero di entità in input**
- d. Un job batch può contenere **centinaia o migliaia di passi**

2. **Online o interactive processing.** Si identifica con la sigla OLTP (online transaction processing), e indicano transazioni eseguite in modo interattivo: richiesta e risposta si susseguono immediatamente, con interazione utente. Un utente esegue una **funzione di business tramite una sequenza di transazioni atomiche**. L'elaborazione deve essere in grado di supportare carichi variabili e transitori di carico superiori alla norma. Nel caso di applicazioni **mission-critical** si devono garantire: continuous **availability, high performance, data protection and integrity**

# Sistemi client server

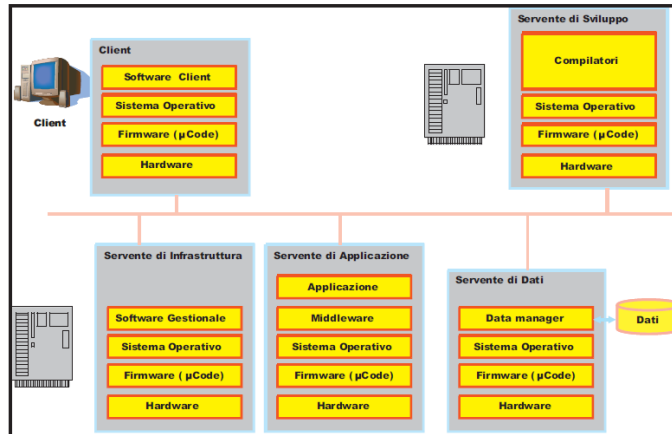
- Supera il modello centrale.
- Computer dotati di potenza di calcolo: i **personal computer (i clienti)**.
- A partire dagli anni 1980.
- I **client** mandano richieste di servizio.
- I server soddisfano la richiesta ed inviano dati ai clienti
- I server sono DB Server, compilatori, applicativi
- **L'interfaccia diventa grafica**



UNIVERSITÀ  
degli STUDI  
di CATANIA



# Sistemi client/server



Modello applicativo client/server



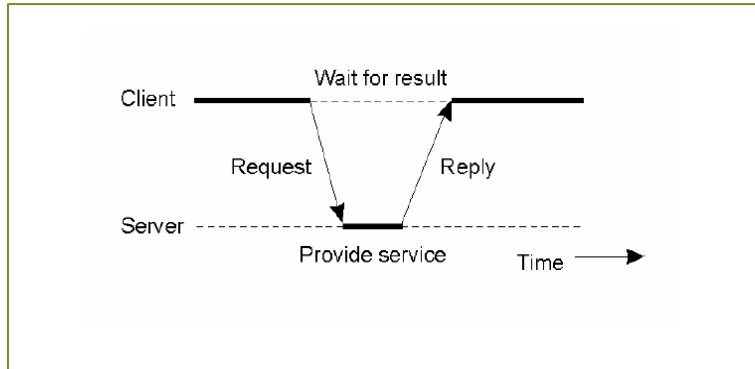
UNIVERSITÀ  
degli STUDI  
di CATANIA



.. In accordo con questo modello, un sistema monolitico viene progressivamente suddiviso in processi client (cioè richiedenti) e processi server (cioè fornitori di servizio) allocati su unità elaborative logicamente (e spesso anche fisicamente) distinte. L'elemento centrale del sistema è la rete. Tipici server sono i server dati (DBMS), i server di applicazioni, i server di sviluppo, dove ad esempio vengono eseguiti i compilatori; l'utenza, non è più necessariamente preidentificata e utilizza interfacce grafiche (GUI) per accedere alle applicazioni...

(IBM MAINFRAME 2010)

# Sistemi client server



## Interazione generale tra client e server



UNIVERSITÀ  
degli STUDI  
di CATANIA



### Client – Funzioni principali

Detto anche **front-end**, esegue un'applicazione in grado di:

1. Presentare **un'interfaccia all'utente**
2. Convertire le richieste dei dati in un formato appropriato, di solito SQL, ad es.  
`select name from studenti where studenti.name=«marco»`
3. **Visualizzare i dati ricevuti dal server**
4. Esecuzione di **logiche applicative**

### Server – Funzioni principali

Detto anche back-end, Il software di gestione del DB esegue operazioni quali:

1. **Aggiornamento, creazione e cancellazione di record** – Il server restituisce solo i record che soddisfano la richiesta del client
2. **Stored procedures, routine di elaborazione già scritte e a disposizione del client per l'esecuzione.** Eseguite sul server riducono il traffico di rete e facilitano la protezione perchè restringono la necessità di protezione di accesso

Un server in rete deve gestire

1. Richieste multiple e simultanee
2. Protezione dei dati
3. Amministrazione della rete e del dominio

Risorse condivise di rete

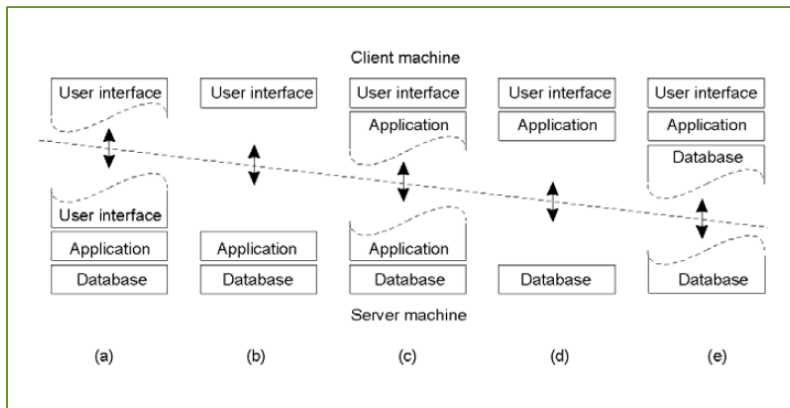
Altre risorse possono essere utilizzate in comune dai client dell'ambiente client/server quali:

- Stampanti, fax
- Directory su dischi remoti

Queste risorse sono installate dall'Amministratore di rete e condivise, cioè rese accessibili agli utenti con privilegi specifici (ad esempio lettura, scrittura, ownership)



# Sistemi client server (\*)



Architetture alternative tra client e server: a - e



UNIVERSITÀ  
degli STUDI  
di CATANIA



- Il server esegue la logica, accede al dato (DB) e prepara la visualizzazione. Il client visualizza i dati (rendering) ed invia le richieste di servizio
- Il server esegue la logica e accede al dato (DB). Il client prepara la visualizzazione e visualizza i dati ed invia le richieste di servizio
- Il server accede al dato (DB). Il client esegue la logica prepara la visualizzazione e visualizza i dati ed invia le richieste di servizio
- Il client dispone anche di un DB locale che viene sincronizzato con il DB sul server. Un esempio attuale di questa modalità con un DB su entrambe le piattaforme sono le applicazioni mobili delle forze di vendita. L'agente – lavorando in mobilità, cioè disconnesso dalla rete Internet - raccoglie gli ordini e i pagamenti del cliente e li salva sul DB locale. Al rientro in ufficio o quando può connettersi alla rete i dati saranno sincronizzati con il DB sul server. Questa soluzione è quella adottata dalle applicazioni di tipo «mobile», che hanno a bordo un sottoinsieme dei dati, di pertinenza dell'operatore che possiede il device

- a. **(\*) PROPOSTA DI APPROFONDIMENTO: DESCRIVERE, PER OGNI VARIANTE MOSTRATA IN FIGURA, UN ESEMPIO CONCRETO (applicazione di uso comune, esempio presente in letteratura, etc).**

# Sistemi client server

- Il client inizia la connessione al server tramite:
  1. un indirizzo IP o un lookup DNS
  2. Un port number (multiplexing per i differenti servizi).
- Uno o più port number sono assegnati ad un servizio. Ad esempio:
  - 80 - HTTP
  - 443 - HTTPS
  - 25, 465, 587, 2525 - SMTP
  - 110 POP3
  - 23 - TELNET
- Un applicativo può accedere al DB server tramite ODBC, una API per esporre servizi del database



UNIVERSITÀ  
degli STUDI  
di CATANIA



## DNS

Il **Domain Name System (DNS)** è un servizio di «directory server» che fornisce una mappatura tra un nome *amichevole* di facile memorizzazione (quasi sempre) ed un indirizzo Internet Protocol o IP (ad es. 125.3.8.217). Questo caso rappresenta un *name service*.

Ha un'organizzazione gerarchica e può fornire anche altri servizi, quali informazioni sugli utenti, mailing list e facilitare il traffico email.

Il meccanismo di mappa rappresenta una soluzione client/server: un client che vuole collegarsi ad un computer o Servizio server, interroga il server DNS che restituisce l'indirizzo di destinazione.

## File HOSTS

Il file hosts è un residuo del passato, essendo un file utilizzato per collegare in **modo statico** nomi di dominio a determinati indirizzi IP, ovvero per risolverli. La gestione dei file host in una grande azienda non è possibile perché occorre modificare in massa tutte le workstation utente. Si usa l'accesso al DNS per risolvere il nome

Consiste cioè in un (lungo) elenco di nomi host e dei relativi indirizzi.

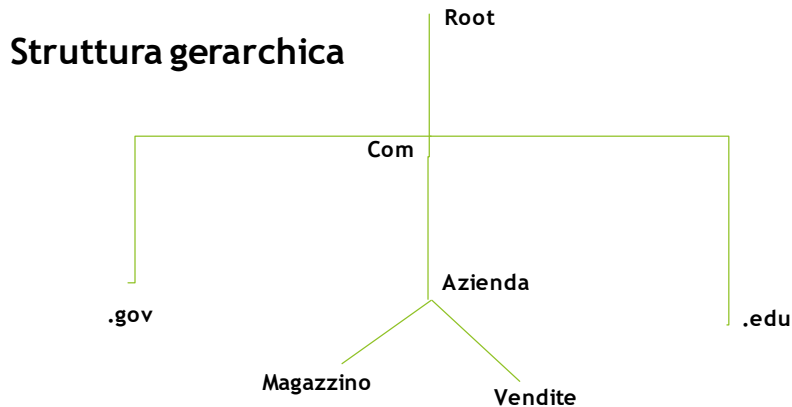
In windows si trova in:

C:\Windows\System32\drivers\etc\hosts

In Linux: /etc/hosts

Esempio di una riga nel file HOSTS  
UNICTDMI 128.1.3.4 (indirizzo fittizio)

# Sistemi client server: DNS



UNIVERSITÀ  
degli STUDI  
di CATANIA



## DNS

Il DNS è il Domain Name System. L'obiettivo del DNS è quella di tradurre i nomi di dominio negli indirizzi IP.

Il DNS è una rete mondiale che costituisce un unico database di nomi di dominio e indirizzi IP.

L'indirizzo IP non è facile da ricordare, quindi sono stati creati nomi di dominio per facilitare l'accesso. Il DNS è responsabile della traduzione di questi nomi di dominio negli indirizzi IP.

## Server DNS

Un server DNS è web server (si vedano pagine seguenti). Il suo ruolo è quello di interagire con il database DNS. Questi server DNS traducono il nome di dominio immesso nell'area URL di un browser Web nell'indirizzo IP corrispondente. Ci sono migliaia di server DNS in tutto il mondo che formano il Domain Name System

## Fully Qualified Domain Name (FQDN)

Fully Qualified Domain Name (FQDN) è il nome di dominio che specifica la sua posizione esatta nella gerarchia DNS.

Specifica tutti i livelli di dominio, incluso il dominio di primo livello e la root. Si compone di due parti, il nome host e il nome di dominio.

Un esempio di FQDN in un server di posta è “mail.mydomain.com” dove “mail” è il nome host e “mydomain.com” è il nome di dominio.

Un nome di dominio completo non ha ambiguità. FQDN è anche chiamato nome di dominio assoluto.

### **DNS Hierarchy (Gerarchia DNS)**

La gerarchia DNS comprende i seguenti livelli:

- 1) Root Level
- 2) Top Level Domains (TLD)
- 3) Second Level Domains (SLD)
- 4) Sub-Domain
- 5) Host

### **DNS Root Zone**

È il livello più alto nell'albero della gerarchia DNS.

I root nameserver sono molto importanti perché sono il primo passo nella risoluzione di un nome di dominio. Questi server contengono l'elenco globale dei domini di primo livello. La root zone contiene i due seguenti elementi:

1. Gerarchia organizzativa come .com, .net, .org.
2. Gerarchie geografiche come .uk, .fr, .pe.

I server DNS radice sono gestiti da 12 diverse organizzazioni.

### **Top Level Domains**

Sono il secondo livello e sono costituiti da due categorie:

- Gerarchie organizzative
- Gerarchie geografiche

La struttura è un albero che parte da una root (radice) e arriva alle foglie. Il nome di un dominio è la concatenazione dei nomi di tutti i nodi – che rappresentano i domini - separati da un punto, dal basso – l'host - fino alla radice (root)

Nell'esempio il nome di un dominio valido DNS è: <https://www.interno.gov.it>

Gerarchie organizzative:

- |       |                                    |
|-------|------------------------------------|
| • com | Commercial organizations           |
| • edu | Educational institutions           |
| • gov | Government institutions            |
| • mil | Military groups                    |
| • net | Major network support centers      |
| • org | Nonprofit organizations and others |
| • int | International organizations        |

La gerarchia geografica è un punto seguito da due lettere per indicare la nazione: .uk, .tk., .it

è.

Esempio: <https://translate.google.co.uk>, <https://www.interno.gov.it>

Dettagli si possono trovare a:

<https://www.interserver.net/tips/kb/dns-dns-hierarchy/>

# Sistemi distribuiti

- Computer collegati in rete, detti nodi (ES: Sistemi Grid)
- Ospitano programmi che condividono
  - Dati
  - Risorse
- Le reti possono essere locali (LAN)
  - Oppure reti di reti, ovvero reti geografiche, le Wide Area Network (WAN)

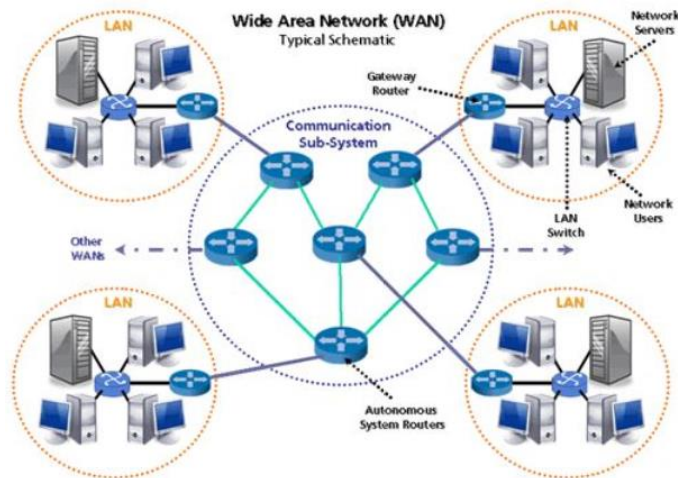


UNIVERSITÀ  
degli STUDI  
di CATANIA





## Sistemi distribuiti



UNIVERSITÀ  
degli STUDI  
di CATANIA



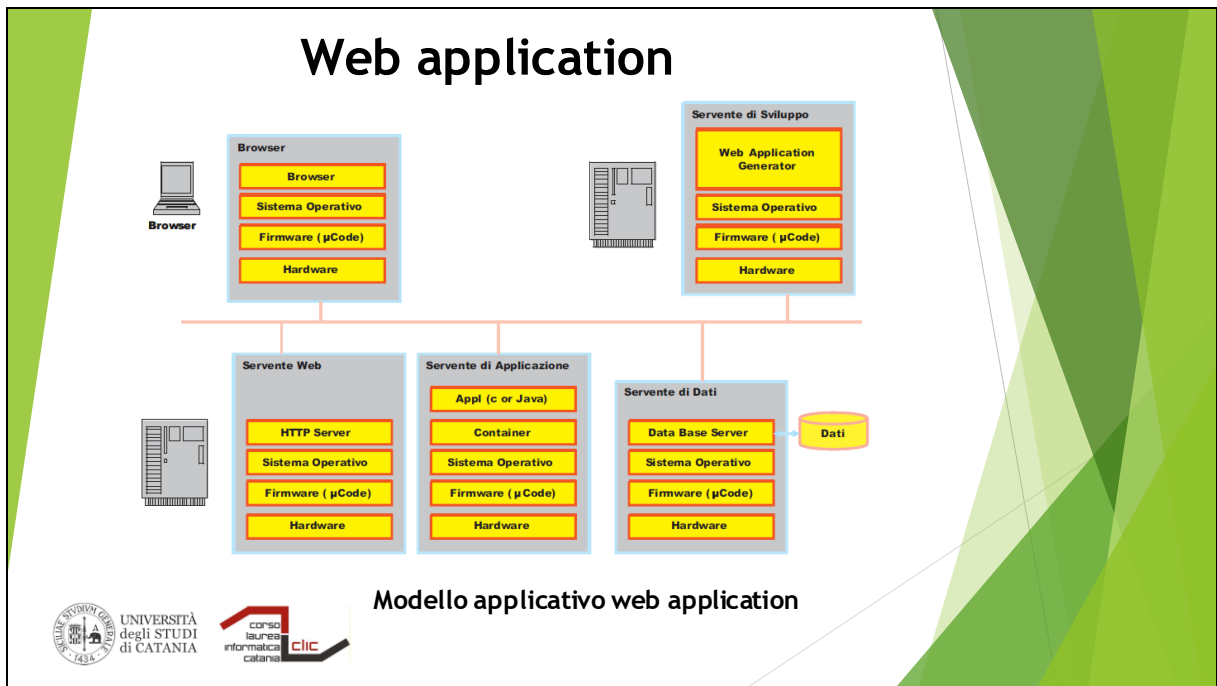
**Classless Inter-Domain Routing (CIDR)** è un metodo per l'allocazione di indirizzi IP e per il routing IP. L'Internet Engineering Task Force ha introdotto CIDR nel 1993 per sostituire la precedente architettura di indirizzamento di rete basata su classi di sottorete (subnet class) su Internet. Il suo obiettivo era rallentare la crescita delle tabelle di routing dei router su Internet e contribuire a rallentare il rapido esaurimento degli indirizzi IPv4.

La notazione CIDR è una rappresentazione compatta di un indirizzo IP e della sua maschera di rete associata. La notazione è stata inventata da Phil Karn negli anni '80. La notazione CIDR specifica un indirizzo IP, un carattere barra ("/") e un numero decimale. Il numero decimale è il conteggio di bit iniziali nella maschera di rete. Il numero può anche essere pensato come la larghezza (in bit) del prefisso di rete.

L'indirizzo IP nella notazione CIDR è rappresentato secondo la notazione standard per IPv4 o IPv6.

Esempio 192.168.100.14/24 rappresenta l'indirizzo IPv4 192.168.100.14 e il prefisso di rete associato 192.168.100.0, o equivalentemente, la sua subnet mask 255.255.255.0, che ha 24 bit iniziali.

[https://en.wikipedia.org/wiki/Classless\\_Inter-Domain\\_Routing](https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing)



Un passo avanti rispetto al modello Client/Server è rappresentato dal modello **Web Application**.

In questo modello il client utilizza in genere come interfaccia un programma "internet browser" di internet (o un applet Java) ed i servizi di connettività applicativa sono forniti da un programma di tipo "web server". In pratica si standardizza l'accesso alle applicazioni in maniera sempre più integrata con Internet e gli standard collegati. I protocolli di comunicazione TCP/IP e http sono alla base di questo modello.

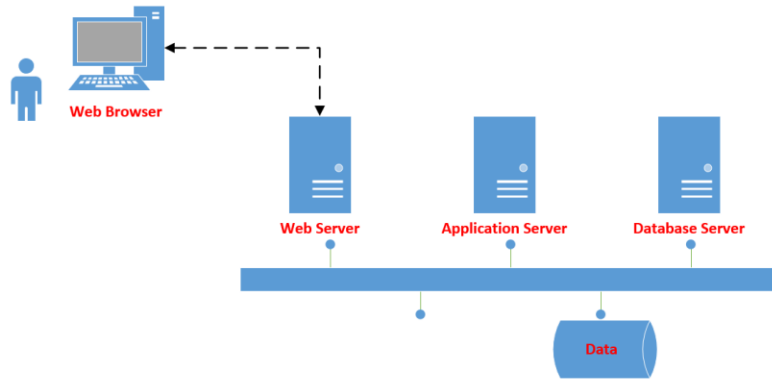
Nel modello, un importante ruolo è rivestito dai

“contenitori” applicativi (Web Application Server) in grado di ospitare applicazioni scritte con linguaggi di programmazione ad oggetti (ad es. Java).

(rif. IBM MAINFRAME 2010)

--> Web 2.0 - pagine web "interattive" codificate in linguaggi come javascript (AJAX), interpretato dal browser. Elaborazione si sposta sul browser

# Architettura a tre livelli



UNIVERSITÀ  
degli STUDI  
di CATANIA



Evoluzione del modello Client/Server.

**Client** --> "internet browser"

**Servizi di connettività** --> web server

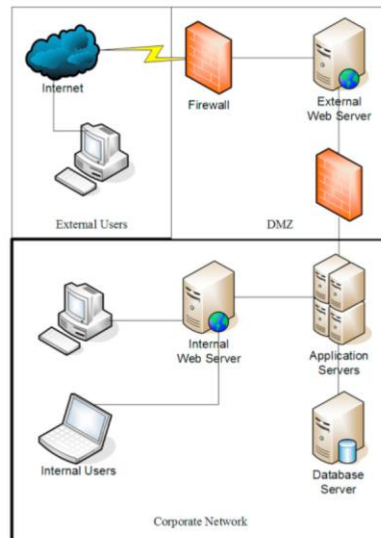
**Elaborazione delle informazioni** --> server applicativo

**Database server** --> mantiene i dati per l'accesso da parte del server applicativo .

**In pratica si standardizza l'accesso alle applicazioni in maniera sempre più integrata con Internet e gli standard de facto.**

**I protocolli di comunicazione TCP/IP e HTTP sono alla base di questo modello.**

## ARCHITETTURA DI ELABORAZIONE



UNIVERSITÀ  
degli STUDI  
di CATANIA



Gli utenti possono accedere dalla rete interna direttamente al Web Server della **LAN aziendale**.

Oppure gli utenti sono dei clienti oppure dei partner (rivenditori, agenti,...) oppure dei dipendenti che lavorano in mobilità. In questo caso gli accessi avvengono tramite Internet e sono protetti da **firewall che delimitano la capacità di accesso ai sistemi interni**

DMZ = DEMILITARIZED ZONE

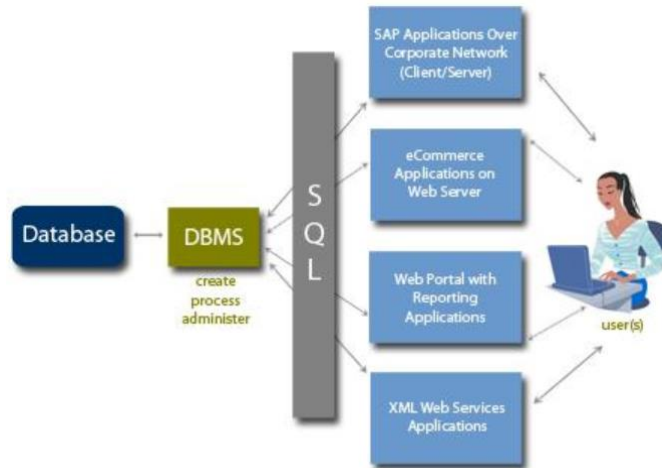
Il firewall consentono di isolare la rete interna da internet, consentendo di accedere solo ad alcuni dati.

Si distinguono tre categorie principali:

- 1. Filtri a pacchetto o senza stati** – Funzionano filtrando i pacchetti per indirizzi e port sorgenti e destinazione. Non considerano la storia del pacchetto
- 2. Filtri a circuito** – Il filtro è in grado di ricordare un insieme limitato di dati della storia del pacchetto e decide l'inoltro anche in base a questa
- 3. Filtri a livello di applicazione** – Il firewall costituisce una rete privata alla quale si collegano i client che non hanno conoscenza del fatto che si stanno collegando ad un server interno – detto anche proxy applicativo. Sarà questo proxy a decidere se i pacchetti possono essere inoltrati e a collegarsi all'esterno. Analogamente le richieste

che giungono dall'esterno vedono una rete che non è quella interna, disaccoppiando in questo modo la rete dal mondo esterno. Uno svantaggio potenziale è che il firewall può introdurre rallentamenti nelle prestazioni comportandosi come un collo di bottiglia (bottleneck)

## IL DBMS E LE APPLICAZIONI



UNIVERSITÀ  
degli STUDI  
di CATANIA



### DDL = DATA DEFINITION LANGUAGE

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
)
```

### DML = DATA MANAGEMENT LANGUAGE

```
Select *  
Form ACCOUNT  
Where SURNAME='Rossi'»
```

## Grid computing (prevalentemente calcolo scientifico)

- ▶ Il modello del **Grid Computing** [9] emerge nella seconda meta' degli anni 90.
- ▶ **Condivisione** di risorse su **larga scala**.
- ▶ Applicazioni **innovative, emergenti** grazie alla condivisione di risorse.
- ▶ High performance / high throughput
- ▶ Grid problem: flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources— what is referred to as **virtual organizations**.
- ▶ Architettura Grid, aperta, modulare, estensibile.
  - ▶ Definizione di **servizi di base**;
  - ▶ **Interfacce per le applicazioni**
  - ▶ **Protocolli di interoperabilita' delle risorse**
- ▶ Sviluppo di middleware (e.g. globus toolkit[10], Glite[11]) che permettono la condivisione di risorse geograficamente distribuite, quindi:
  - ▶ Autenticazione
  - ▶ Autorizzazione per l'uso di risorse
  - ▶ Resource discovery



## ELABORAZIONE IN CLOUD

**Definizione secondo U.S. National Institute of Standards and Technology (NIST) [8]**

**“Cloud è un modello di accesso via rete da qualsiasi luogo, comodo e on-demand, ad un insieme di risorse condivise - reti, server, dispositivi di memorizzazione, applicazioni e servizi - che possono essere forniti e rilasciati rapidamente, con minimo costo di gestione o interazione con il fornitore del Servizio”**

**Nelle note è riportato il testo originale in inglese.**



UNIVERSITÀ  
degli STUDI  
di CATANIA



National Institute of Standards and Technology (NIST):

*“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”*

## ELABORAZIONE IN CLOUD

Caratteristiche dell'elaborazione in Cloud - vedi Note

1. On **demand** (variazione di risorse "agile")
2. Shared resource pooling (risorse condivise)
3. Rapid **elasticity** (variazione di risorse "immediata")
4. **Measured service** (misura uso risorse / fatturaz. A consumo)
5. **Broad network access**
6. High availability, disaster recovery (SLA)



UNIVERSITÀ  
degli STUDI  
di CATANIA



Si può leggere il link [5] in aggiunta - opzionale

1. On demand – La **variazione delle risorse necessarie non richiede la procedura di acquisizione e installazione e configurazione di nuovo hardware o software**, ma basta richiedere al fornitore (Provider) l'aggiunta (o diminuzione) della capacità elaborativa
2. Shared resource pooling – Le risorse sono localizzate su **piattaforme condivise** che vengono **assegnate in modo dinamico** in silos dedicati e sicuri che realizzano la suddivisione delle aziende
3. Rapid elasticity – La variazione delle risorse è (pressocchè) immediata. Questo consente di **fronteggiare picchi di richieste**. Esempi tipici: un ciclo di test di un'applicazione in fase di rilascio; la variazione di carico prevista nei periodi di picco, come Natale o l'Estate per le compagnie telefoniche (Telco)
4. Measured service – Il fornitore **misura i consumi di risorse ed il numero di utenti per fatturare il costo del servizio (a consumo)**
5. Broad network access – Accesso da qualsiasi parte del mondo (in accordo ai requisiti di sicurezza del cliente), e con diversi device: workstation, laptop, mobile phone,..
6. High availability, disaster recovery – Il servizio cloud deve mostrare valori di disponibilità elevatissimi – superiori al 95% - e resistere ad eventi catastrofici. Il livello di servizio viene stabilito nel contratto come un **Service Level Agreement**

**(SLA).** Questi concetti saranno chiariti nel seguito del corso

# ELABORAZIONE IN CLOUD

Modelli infrastrutturali [5] - vedi Note

1. Public cloud (forma più diffusa, accessibile a tutti)
2. Private cloud (servizi personalizzati, maggiore sicurezza)
3. Hybrid cloud



UNIVERSITÀ  
degli STUDI  
di CATANIA

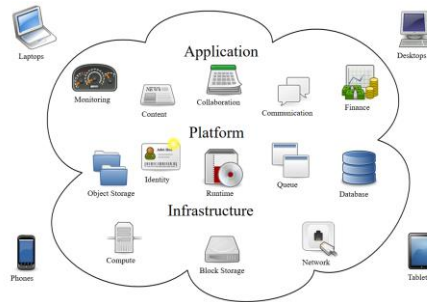


1. **Public cloud** – E' la forma canonica del Cloud. Il provider mantiene i dati ed i programmi nei propri centri di elaborazione e vende il servizio al cliente. Il cliente accede ai servizi tramite browser. La gestione dell'infrastruttura e del software (patching) è a carico del provider.
2. **Private cloud** – Indica servizi di calcolo offerti solo a utenti selezionati e non al pubblico generale. Il cloud privato offre alle aziende molti dei vantaggi di un cloud pubblico, ad esempio **funzionalità self-service, scalabilità ed elasticità, con controllo e personalizzazione aggiuntivi forniti da risorse dedicate tramite un'infrastruttura di elaborazione ospitata in locale**. I cloud privati, inoltre, offrono un livello elevato di sicurezza e privacy tramite firewall aziendali e hosting interno per garantire che le operazioni e i dati sensibili non siano accessibili a provider di terze parti.
3. **Hybrid cloud** – Una combinazione delle precedenti.

## Modelli (\*).

### Cloud Service Models

1. SaaS - Software as a service
  2. PaaS - Platform as a service
  3. IaaS - Infrastructure as a service
- Vedi Note



UNIVERSITÀ  
degli STUDI  
di CATANIA



**ESERCIZIO:** Nel corso degli anni, dalla comparsa del cloud, sono stati proposti differenti ulteriori modelli X-as-a-service (ES: DaaS – Data as a Service). Svolgere una breve indagine in letteratura al fine di elencare ulteriori servizi X-as-a-service.

**IaaS (Infrastructure as a Service)** consiste nel servizio di fornitura di risorse hardware quali server, rete, memorizzazione, archivio e backup. La caratteristica dello IaaS è che le risorse possono essere istanziate al momento in cui una piattaforma ne ha bisogno. Questa opzione offre la massima capacità di controllo da parte del cliente

**Platform as a service (PaaS)** consiste nel servizio di fornitura di piattaforme di elaborazione, quali middleware, web server, DB Server, ambienti di sviluppo Java o Python), autenticazione. E' un livello superiore all'infrastruttura (IaaS) e quindi offre minore capacità di controllo da parte del cliente

Una panoramica di servizi PaaS fornito da Oracle è disponibile a <https://www.oracle.com/uk/cloud/>

**Software as a service (SaaS)** il provider fornisce l'accesso ad un'applicazione web che mette a disposizione dei propri clienti via Internet. Il software può essere un custom o un package (si vedrà nelle prossime slide). Tutta l'infrastruttura è gestita dal provider. Update di versioni del software di base, tuning, copie e salvataggi dei dati sono trasparenti per il

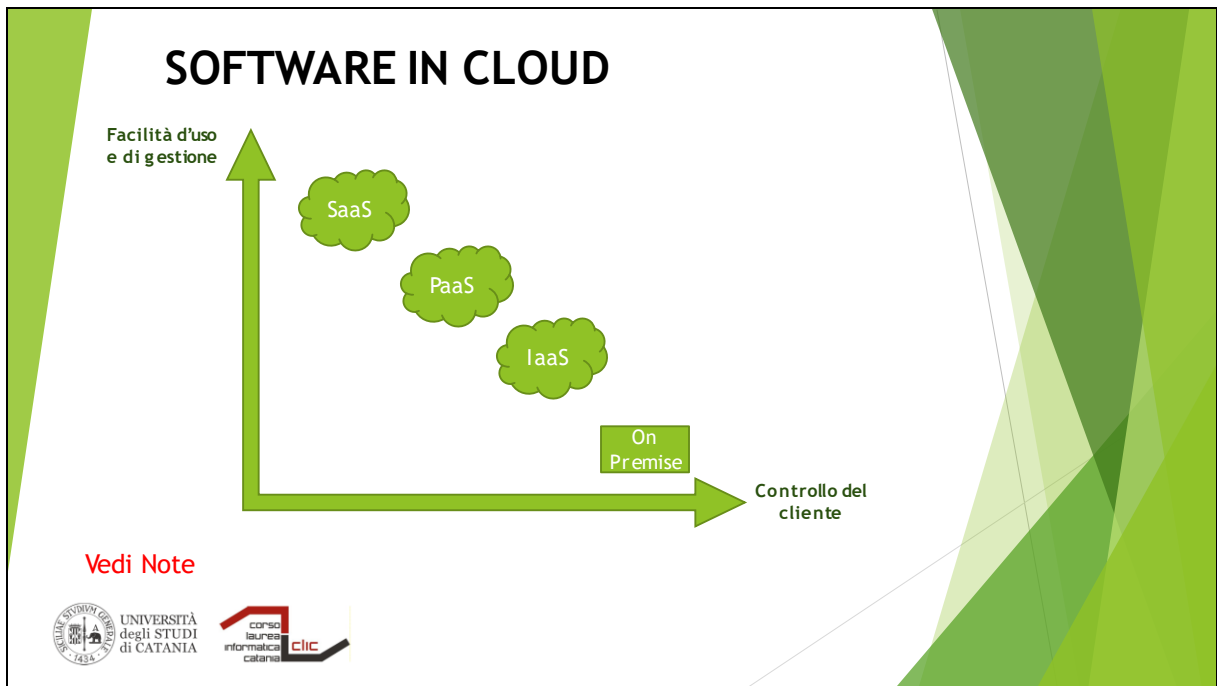
cliente – Canone in abbonamento – Questa opzione offre la minima capacità di controllo da parte del cliente

Una componente PaaS che sta assumendo una vendita separate come servizio è il **DBaaS**. Il DBMS è offerto in cloud e quindi sono fornite tutte le caratteristiche del servizio in Cloud:

1. On demand
2. Shared resource pooling
3. Rapid elasticity
4. Measured service
5. Broad network access
6. High availability, disaster recovery

Si veda anche [7] – Opzionale

Ulteriori letture opzionali: [https://it.wikipedia.org/wiki/Cloud\\_computing](https://it.wikipedia.org/wiki/Cloud_computing)



Si veda anche [6] per IaaS, PaaS, SaaS.

Come si vede il cammino verso il Cloud supera le "vecchie" abitudini di avere tutto all'interno della compagnia.

Tutto ciò avviene grazie allo sviluppo delle **tecnologie per la virtualizzazione**.

**Il cloud supera il problema dei costi di acquisto e manutenzione dell'hardware, con relative rapida obsolescenza.**

L'upgrade del software di base (sistemi operativi, DBMS, Middleware, storage).

Inoltre sposta le spese relative all'Information technology (IT) dagli investimenti in conto capitale (**CAPEX – Capital EXpenditure**) a costi operativi (**OPEX – Operational Expenditures**)

Gli OPEX sono dedotti dal fatturato ai fini del calcolo delle imposte.

I CAPEX sono soggetti ad **ammortamenti** (di solito triennali per le infrastrutture informatiche) e quindi si trasformano anch'essi in costi ricorrenti ma con una maggiore

complessità e minore possibilità di detrazioni dalle imposte

Si può consultare il Cap 4.0.8 del manuale IBM in bibliografia che tuttavia è molto più orientato alla valutazione del costo di acquisizione e gestione



## SOFTWARE CUSTOM

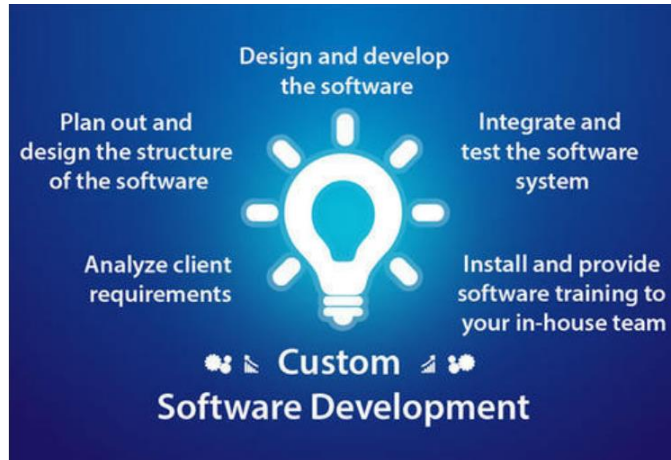
- Il software custom (anche detto tailor-made software) è **sviluppato appositamente per un unico utilizzatore** o cliente sulla base di requisiti specifici
- Esempi tipici sono:
  - i programmi Cobol dei mainframe (es. Banche)
  - le applicazioni client server o di tipo web sviluppate per clienti finali o dall'IT interno oppure da consulenti esterni



UNIVERSITÀ  
degli STUDI  
di CATANIA



# SOFTWARE CUSTOM



Vedi Note



UNIVERSITÀ  
degli STUDI  
di CATANIA



Il cliente che sviluppa un software custom deve tenere in considerazione tutte le fasi ed i costi ad esse associati:

- Disegno
- Sviluppo
- Test
- Installazione e porting tra ambienti
- Manutenzione evolutiva
- Gestione operativa

Il ciclo di sviluppo può essere qualsiasi: **waterfall, agile, scrum**

## SOFTWARE PACKAGED

- **Software** sviluppato per essere venduto a molti clienti (il software package)
- Le **funzionalità** del sistema sono **comuni** per tutti i clienti
- Le specifiche del software sono tipiche del mercato, del settore merceologico, del tipo di utente finale
- Può essere configurabile tramite **parametrizzazione**, aggiunta di moduli, tramite tool grafici o aggiunta di codice

Vedi Note



UNIVERSITÀ  
degli STUDI  
di CATANIA



Come si vede può essere un applicativo «blindato» come il foglio di calcolo, il **CAD** o l'editor grafico.

Oppure a livello aziendale sono software che coprono aree aziendali critiche – **contabilità, logistica o CRM** – e che ammettono la modifica o personalizzazione delle funzionalità (anche detta customizzazione). Questa personalizzazione deve essere limitata per non modificare in modo negativo la stabilità e la performance del package

Un software che include funzionalità standard comuni ad un settore merceologico – telecomunicazioni, energia, automobilistico – viene spesso indicato come «verticalizzazione».

## SOFTWARE PACKAGED

Mercato aziendale o Enterprise (Esempi)

- Enterprise resource planning (ERP)
- Customer Relationship management (CRM) / Customer Experience CX
- Supply Chain Management SCM
- Applicazioni di Marketing e Business Intelligence

Mercato al consumo o mass market (Esempi)

- Excel
- Word



UNIVERSITÀ  
degli STUDI  
di CATANIA



Ad esempio I software utilizzati dalle compagnie telefoniche nei loro call center rientrano nella categoria dei CRM

## SOFTWARE ON PREMISE

Il software on-premise consiste nell'installazione ed esecuzione del software su architettura locale presso il cliente esclusivamente raggiungibile dall'interno della rete aziendale.

- Vs off premise (server remoto)
- Vs in Cloud: SaaS (software as a service). SaaS ha gradualmente preso il posto della soluzione in oggetto a partire Dal 2005.

L'utente può accedere alla rete aziendale anche tramite la Internet pubblica, fatto salvo il criterio di sicurezza previsto (es: VPN)



UNIVERSITÀ  
degli STUDI  
di CATANIA



Opzionale: [https://it.wikipedia.org/wiki/On-premises\\_software](https://it.wikipedia.org/wiki/On-premises_software)

## SOFTWARE ON PREMISE

- (+) Controllo esclusivo su sistemi e dati
- (+) Gestione interna di dati sensibili e core business
- (+) Flessibilità nella gestione, ad esempio il ciclo di patch
- (-) Necessita di uno staff IT dedicato
- (-) E' un costo di investimento (CAPEX)
- (-) Costo di gestione complessivo elevato (Total Cost of Ownership - TCO):
  - costi iniziali di acquisto di software, hardware e di implementazione
  - + costi ricorrenti di gestione: supporto, costi energia, ammortamenti..
- Si può fare una stima iniziale su un orizzonte di tre anni per capire quanto si spenderà in totale per l'investimento



UNIVERSITÀ  
degli STUDI  
di CATANIA



# SOFTWARE IN CLOUD

Installazione ed esecuzione del **software** su di una **architettura in Cloud** fornita da un provider

Raggiungibile dalla **rete pubblica**, fatto salvo il criterio di sicurezza previsto.

In una modalità SaaS:

- **il cliente può installare in cloud il proprio software, quindi disegnato e sviluppato in proprio (in house).**
- **Oppure è il provider stesso del servizio cloud che mette a disposizione un software, che il cliente personalizza ('customizza').**

Quindi il provider è allo stesso tempo il fornitore dell'infrastruttura (IaaS, PaaS) e installa e gestisce il proprio software, personalizzato dal cliente, realizzando quindi un servizio SaaS.



UNIVERSITÀ  
degli STUDI  
di CATANIA



## SOFTWARE IN CLOUD (vantaggi)

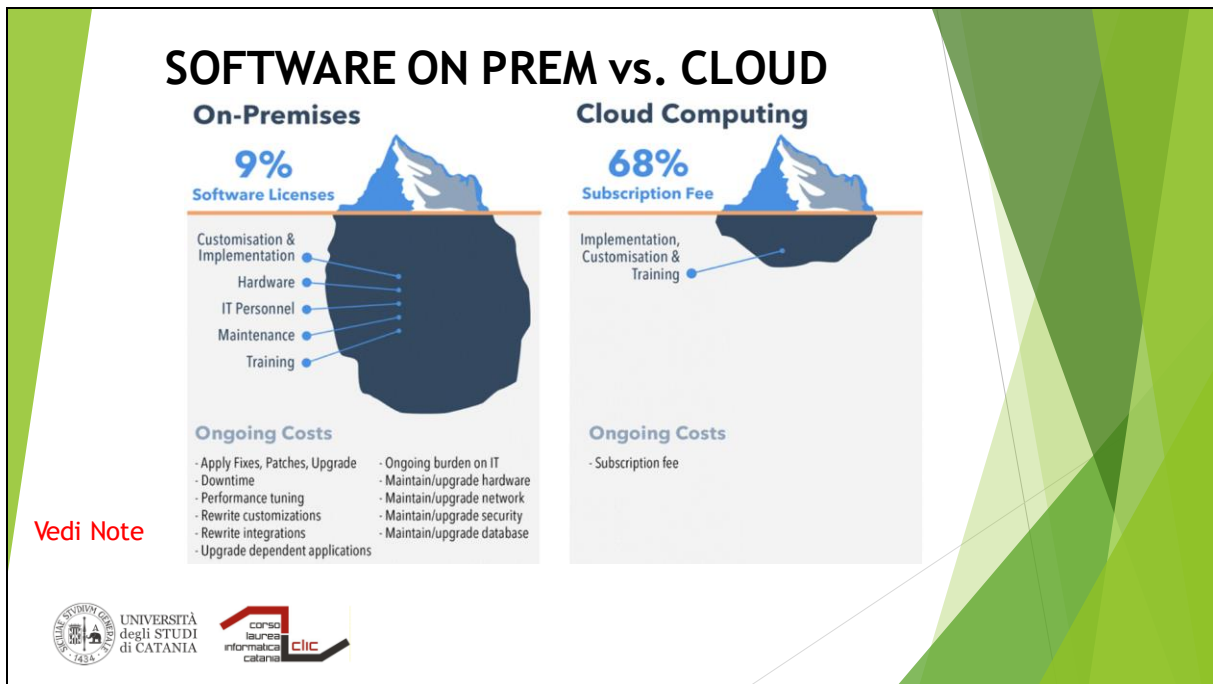
- Controllo e gestione dei sistemi e dati delegato al provider
- Non necessita di uno staff IT dedicato
- E' un costo operativo OpEx (vs CapEx)
- Libera da necessità di gestione dell'infrastruttura
- La gestione del software dipende dal tipo: custom o package
- Costo di gestione inferiore rispetto al modello on premise (Total Cost of Ownership - TCO).



UNIVERSITÀ  
degli STUDI  
di CATANIA







La figura mostra un paragone teorico tra i costi totali di acquisizione e di gestione – **il TCO** - del software on premise ed in Cloud.

Per essere perfettamente applicabile **il confronto deve tenere in considerazione alcuni elementi principali:**

- Il costo dello **sviluppo** del software e' **altamente variabile**: packaged o custom
- Nel caso di un package, l'approccio in cloud deve essere più possibile "agile" ovvero superare il ciclo di sviluppo a cascata (waterfall)
  - Si deve ragionare in funzione di *solution driven* invece che di *requirement driven*. Ovvero è il cliente che si deve adeguare ai processi implementati nativamente (out of the box) dal package, anzichè forzare personalizzazioni dello stesso
  - Il ciclo di sviluppo deve essere di tipo agile: una serie di workshop – tre o quattro – dove si prospettano agli utenti le funzionalità del package e si raccolgono le richieste di modifica – NON I REQUIREMENT
  - Nei workshop successivi si mostrano le modifiche apportate (possibilmente limitate)
  - Il workshop finale convalida la soluzione concordata

Approcci diversi, ancora legati all'approccio waterfall aumenteranno inevitabilmente I costi di sviluppo implementazione

## LINK NOTEVOLI / bibliografia

1. Autori Vari - Il Mainframe 2010 - Terza Edizione - IBM (scaricare il pdf)

[https://www.dmi.unict.it/messina/didat/SC/MAINFRAME\\_LR.pdf](https://www.dmi.unict.it/messina/didat/SC/MAINFRAME_LR.pdf)

2. ODBC

<https://docs.microsoft.com/en-us/sql/odbc/reference/what-is-odbc?view=sql-server-ver15>

3. Port Numbers

<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

4. Public SaaS for Dummies - 2<sup>nd</sup> Edition - Contiene un glossario riepilogativo (pag. 61)

<https://www.oracle.com/a/ocom/docs/applications/saas-for-dummies-3rdedition.pdf>



UNIVERSITÀ  
degli STUDI  
di CATANIA



## LINK NOTEVOLI / bibliografia

5. Descrizione semplice dell'elaborazione in cloud

<https://community.spiceworks.com/cloud/articles/2502-what-is-the-cloud-cloud-computing-for-dummies>

6. IaaS, PaaS, SaaS

<https://community.spiceworks.com/cloud/articles/2504-iaas-paas-saas-differences-between-cloud-service-models>

7. DbaaS

<https://www.ibm.com/cloud/learn/dbaas#toc-what-is-db-XJUdAzg0>

8. Nist definition of Cloud Computing

<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>

9. Foster, I., Kesselman, C. and Tuecke, S., 2001. The anatomy of the grid: Enabling scalable virtual organizations. *The International Journal of High Performance Computing Applications*, 15(3), pp.200-222.

<https://arxiv.org/pdf/cs/0103025.pdf>



UNIVERSITÀ  
degli STUDI  
di CATANIA



## LINK NOTEVOLI / bibliografia

[10] Globus middleware (retired): <https://toolkit.globus.org/>

[11] Glite (retired): <https://en.wikipedia.org/wiki/GLite>