



---

Università degli Studi di Parma  
Facoltà di Scienze MM. FF. NN.  
Corso di Laurea in Informatica

**Ingegneria del Software**

**Architetture Software**

**Giulio Destri**



Ing. del Sw: Architettura - 1

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Scopo del modulo

---

**Definire**

**L'architettura moderna delle  
applicazioni software**

Ing. del Sw: Architettura - 2

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Argomenti

---

- Le componenti del sistema
- Applicazioni monolitiche e client-server
- L'approccio Model-View-Controller (MVC)

Ing. del Sw: Architettura - 3 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## Struttura base di un'applicazione

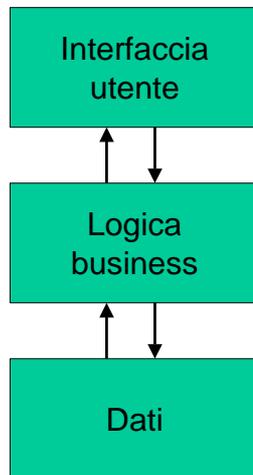
---

- **Interfaccia utente** (grafica):  
presentazione dei dati e interazione con l'utente
- **Regole funzionali** (logica business):  
le procedure che compiono le operazioni in base ai comandi ricevuti dal livello precedente
- **Dati**: su cui si deve agire e che devono essere memorizzati (durano oltre i programmi)

Ing. del Sw: Architettura - 4 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## La struttura dell'applicazione

---



Ing. del Sw: Architettura - 5 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

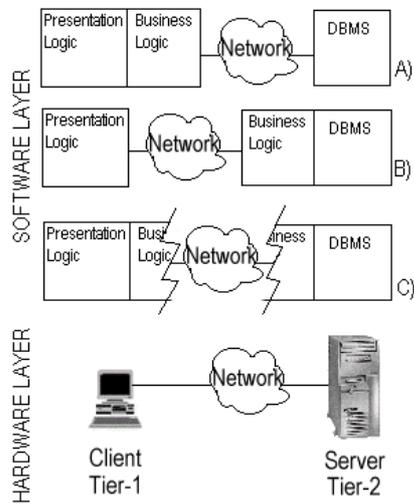
## Il modello client-server

---

- Un **programma server** opera su un **computer server (host)**
- Un **programma client** opera su una **postazione client** (workstation)
- L'utente interagisce col programma client
- Il programma client dialoga col server **via rete**

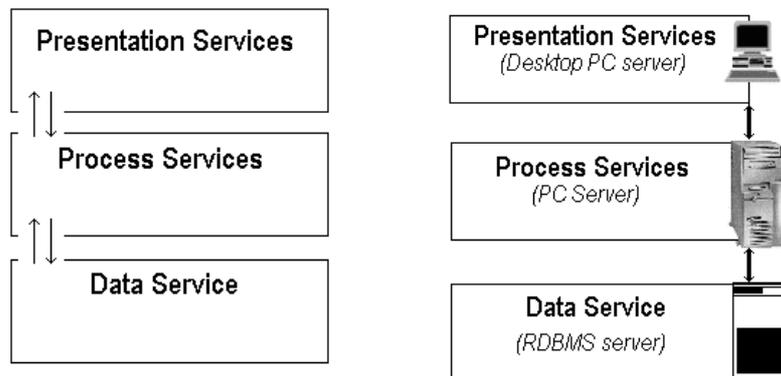
Ing. del Sw: Architettura - 6 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## 2-Tier Client/Server



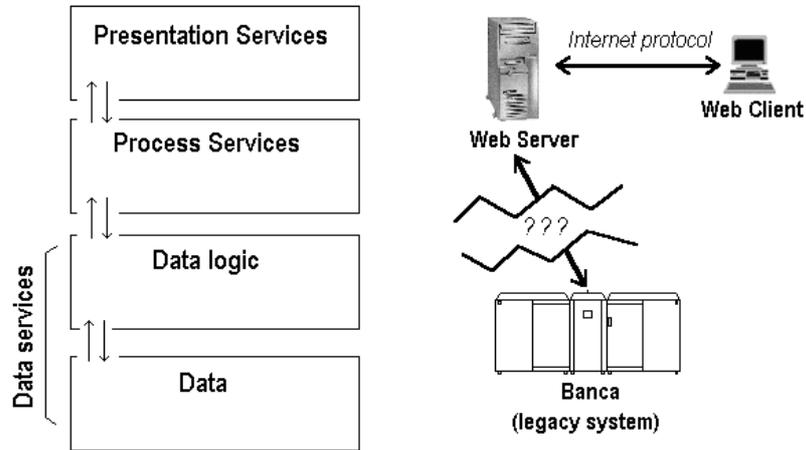
Ing. del Sw: Architettura - 7 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## 3-Tier Client/Server



Ing. del Sw: Architettura - 8 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

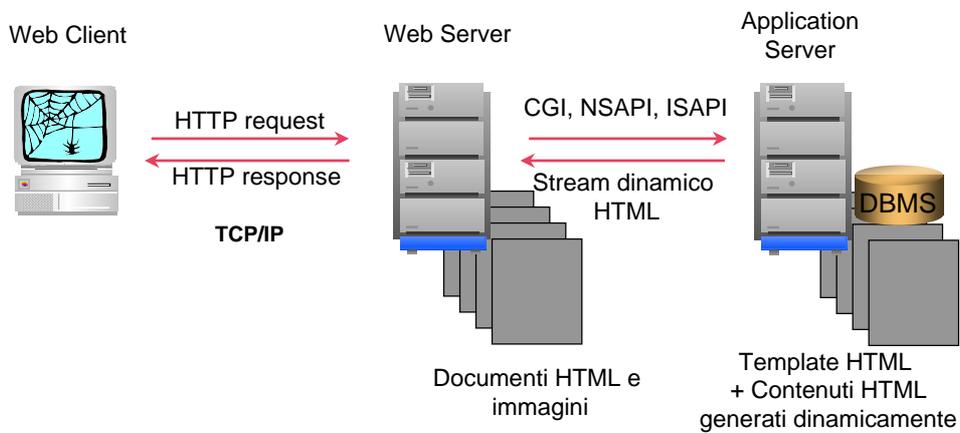
## Multi-Tier Client/Server



Ing. del Sw: Architettura - 9

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Architettura Web base

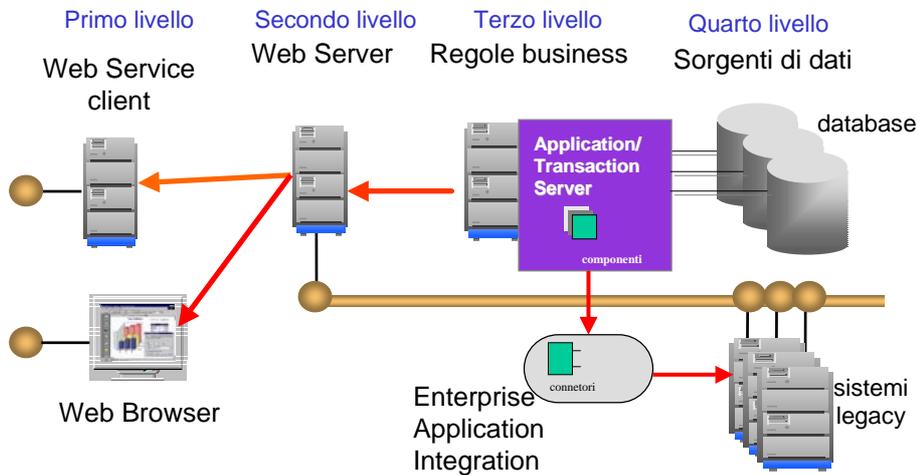


Ing. del Sw: Architettura - 10

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Architettura Web: estensioni

---



Ing. del Sw: Architettura - 11

Giulio Destri - © AreaSP for Univ. Parma, 2006

## L'interfaccia utente

---

- Riga comandi (es. DOS, UNIX)
- Riga comandi/menu (es. TSO-MVS)
- Maschera testo (es. applicativi gestionali prime generazioni)
- GUI a finestre (es. Windows, Motif)
- GUI/Maschera Web

Ing. del Sw: Architettura - 12

Giulio Destri - © AreaSP for Univ. Parma, 2006

## **L'interfaccia utente Web**

---

- Maschera/tabella Web base (es. Form HTML)
- Lightweight client Web (es. programmi JavaScript, Flash con retroconnessione)
- Applet Java (es. JDE for Web, versione Java)
- Controllo ActiveX

Ing. del Sw: Architettura - 13 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## **Le funzioni dell'interfaccia utente**

---

- Gestione degli eventi dell'utente
- Immissione dati
- Verifica coerenza fra dati e regole dell'applicazione
- Visualizzazione dati (risultati elaborazione)
- Notifica eventi interni all'applicazione

Ing. del Sw: Architettura - 14 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## Le funzioni dell'interfaccia utente - 2

---

- La verifica della coerenza dati viene fatta
  - Dalla interfaccia utente?
  - Trasmessa ad un altro modulo e qui fatta?
  - Quando (a fine compilazione o campo per campo)?
- La gestione eventi viene fatta
  - Localmente?
  - Da un altro modulo?

Ing. del Sw: Architettura - 15 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## La logica business

---

- Offre le funzionalità di azioni sui dati
- Le funzionalità possono anche essere asincrone rispetto all'interfaccia utente (avvio senza attesa bloccante)
- La successione degli eventi trasmessagli dalla interfaccia utente determina la successione di azioni sui dati
- Si connette alla base di dati

Ing. del Sw: Architettura - 16 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## La base di dati

---

- Nella stragrande maggioranza dei casi è un database relazionale
- Non è detto che sia visto come tale dalla logica business
  - Meccanismi trasparenti di persistenza di oggetti (es. EJB)
  - Accesso dati in formato XML (es. Oracle, ADO-NET)

Ing. del Sw: Architettura - 17 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## La base di dati - 2

---

- Quasi sempre occorre una mappatura relazionale-oggetto
- Nei casi XML e EJB esistono sistemi automatici di ausilio alla mappatura
- Il database deve essere progettato a partire dall'analisi a oggetti dell'applicazione

Ing. del Sw: Architettura - 18 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## Problematiche in applicazioni complesse

---

- Necessità di accedere alle stesse funzioni da tipi diversi di interfacce (es. approccio multi-canale)
- Necessità di porting di applicazioni su piattaforme diverse con il massimo riuso possibile del codice
- Problematiche di GUI su piattaforme diverse (tipiche di Java)

Ing. del Sw: Architettura - 19 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## Il pattern Model View Controller (MVC)

---

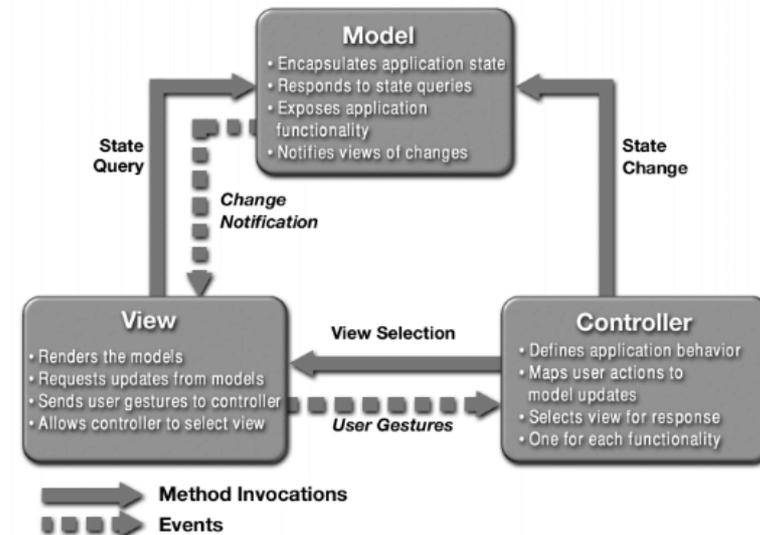
E' un modo per suddividere un'applicazione (o anche solo la sua interfaccia utente) in tre parti

- Controller = input
- Model = elaborazione
- View = output

Ing. del Sw: Architettura - 20 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## MVC: architettura teorica

---



Ing. del Sw: Architettura - 21 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## MVC: separazione dei ruoli

---

- L'input dell'utente,
- il programma che modella i processi del dominio business
- la risposta visuale (testo, immagini ecc...) all'utente
- sono separati fra loro e gestiti rispettivamente
  - dal modulo controller
  - dal model
  - dal view

Ing. del Sw: Architettura - 22 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## MVC: il modulo controller

---

- Il controller interpreta gli eventi di input via tastiera o mouse dell'utente
- Mappa questi eventi
- traducendoli in comandi/segnali
- che invia al modello e/o al view
- per realizzare il cambiamento/azione richiesto

Ing. del Sw: Architettura - 23 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## MVC: il modulo model

---

- Il model gestisce uno o più elementi dei dati
- Risponde alle interrogazioni relative al suo stato
- Reagisce alle istruzioni relative al cambio di stato

Ing. del Sw: Architettura - 24 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## MVC: il modulo view

---

- Il view (detto anche viewport) gestisce un'area del display
- E' responsabile della presentazione dei dati all'utente
- attraverso un'apposita combinazione di elementi grafici e testuali

Ing. del Sw: Architettura - 25 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## MVC: approfondimenti sul model

---

- Il model viene usato
- per trattare le informazioni
- per notificare gli osservatori (ossia i moduli view che mostrano i dati all'utente)
- quando le informazioni cambiano
- (es. il news ticker di borsa)

Ing. del Sw: Architettura - 26 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## MVC: approfondimenti sul model

---

- Il model contiene soltanto dati e funzionalità legate da uno scopo comune
- Per mettere assieme due gruppi di dati e funzionalità non in relazione tra loro sarebbe bene creare due model separati, uno per ciascuno di essi.

Ing. del Sw: Architettura - 27 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## MVC: approfondimenti sul model

---

Possiamo considerare il model come

- il risultato di una analisi
- modellazione
- ed adattamento al mondo informatico
- del mondo reale

Ing. del Sw: Architettura - 28 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## **MVC: approfondimenti sul model**

---

- Ovvero come una "approssimazione" computazionale
- di un sistema o processo del mondo reale (o meglio del dominio di business)

Ing. del Sw: Architettura - 29 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## **MVC: approfondimenti sul model**

---

Il model contiene quindi

- non solo lo stato del sistema reale
- ma anche i suoi principali processi di funzionamento

Ing. del Sw: Architettura - 30 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## **MVC: approfondimenti sul model**

---

In pratica quindi in un'applicazione ad oggetti il model

- è l'insieme dei componenti interni dell'applicazione,
- risultanti dalla proiezione sotto forma di classi ed oggetti entro il dominio software
- delle entità del dominio di business e delle loro relazioni

Ing. del Sw: Architettura - 31 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## **MVC: approfondimenti sul model**

---

In particolare si può pensare ad un model che crea un "ponte" fra un sistema informatico e la interfaccia utente

- In questo scenario il modello "avvolge" ed astrae le funzionalità del sistema (software o hardware)
- e agisce da collegamento con il mondo esterno
- ovvero l'interfaccia utente

Ing. del Sw: Architettura - 32 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## **MVC: approfondimenti sul controller**

---

- Il controller è il modulo attraverso cui l'utente interagisce con l'applicazione
- Un controller riceve l'input dall'utente ed istruisce il model e il view a compiere le azioni associate a tale input
- Il controller è responsabile di mappare le azioni dell'utente sulle risposte dell'applicazione

Ing. del Sw: Architettura - 33 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## **MVC: approfondimenti sul view**

---

- Il view ha lo scopo di mappare una visualizzazione (grafica o no) su un dispositivo di output
- Tipicamente il view ha una corrispondenza con le caratteristiche del dispositivo e "sa" come rappresentare i dati su di esso
- Il view si "aggancia" ad un model e visualizza i suoi contenuti sul dispositivo di output

Ing. del Sw: Architettura - 34 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## MVC: approfondimenti sul view

---

- In più, quando il model cambia, il view automaticamente aggiorna le parti corrispondenti della visualizzazione, in modo da rendere visibili tali cambiamenti

Ing. del Sw: Architettura - 35 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

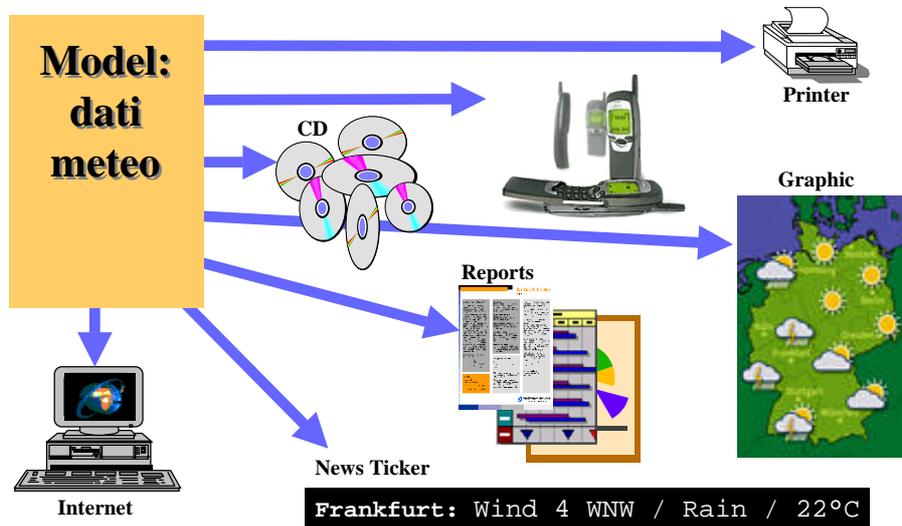
## MVC: approfondimenti sul view

---

- Per lo stesso model possono esistere diversi view e ciascuno di essi può visualizzare gli stessi dati su dispositivi di output diversi in modo diverso
- Un view inoltre può essere composito, ovvero formato da vari sub-view, eventualmente a loro volta compositi

Ing. del Sw: Architettura - 36 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## MVC: approfondimenti sul view



Ing. del Sw: Architettura - 37 Giulio Destri - © AreaSP for Univ. Parma, 2006

## Scomposizione in elementi tecnici

- Interfacce verso altri sistemi
- Interfacce utente (es. GUI, Web)
- Logica di business
- Base di dati

Ing. del Sw: Architettura - 38 Giulio Destri - © AreaSP for Univ. Parma, 2006

## Descrizioni delle interfacce

---

- Interfacce verso altri sistemi
  - Titolo
  - Descrizione (breve)
  - Elenco dei metodi
  - Elenco dei possibili errori
- Interfacce utente (es. GUI, Web)
  - Prototipo di analisi
  - O schematic

Ing. del Sw: Architettura - 39 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## Descrizioni della logica di business

---

- Titolo
- Descrizione (breve)
- Relazione con altri controller
- Relazione con le entità
- Elenco dei metodi
- Elenco degli errori
- Modello a oggetti
- Osservazioni

Ing. del Sw: Architettura - 40 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## Descrizioni della base di dati

---

- Titolo
- Attributi
- Metodi
- Possibili errori
- Relazioni
- Generalizzazione/specializzazione
- Consistenza

Ing. del Sw: Architettura - 41 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

## Sommario

---

- Le componenti del sistema
- Applicazioni monolitiche e client-server
- L'approccio Model-View-Controller (MVC)

Ing. del Sw: Architettura - 42 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)