Interazione Uomo-Macchina e tecnologie web

Prof.ssa Liliana Ardissono
Dipartimento di Informatica
Università di Torino

Applicazioni web - architetture a 3 livelli

Applicazioni Web-based e non

Elementi di base di un'applicazione

- I dati da trattare (dati sugli utenti, sul servizio offerto, etc.)
- La logica applicativa (***business logic***), ovvero le funzionalità che l'applicazione deve offrire, il tipo di trattamento dei dati previsto (quali operazioni si possono fare), etc.
- L'interfaccia utente
  - interfaccia web (e.g., per browser, o testuale)
  - stand-alone (applicazione locale alla macchina su cui gira)

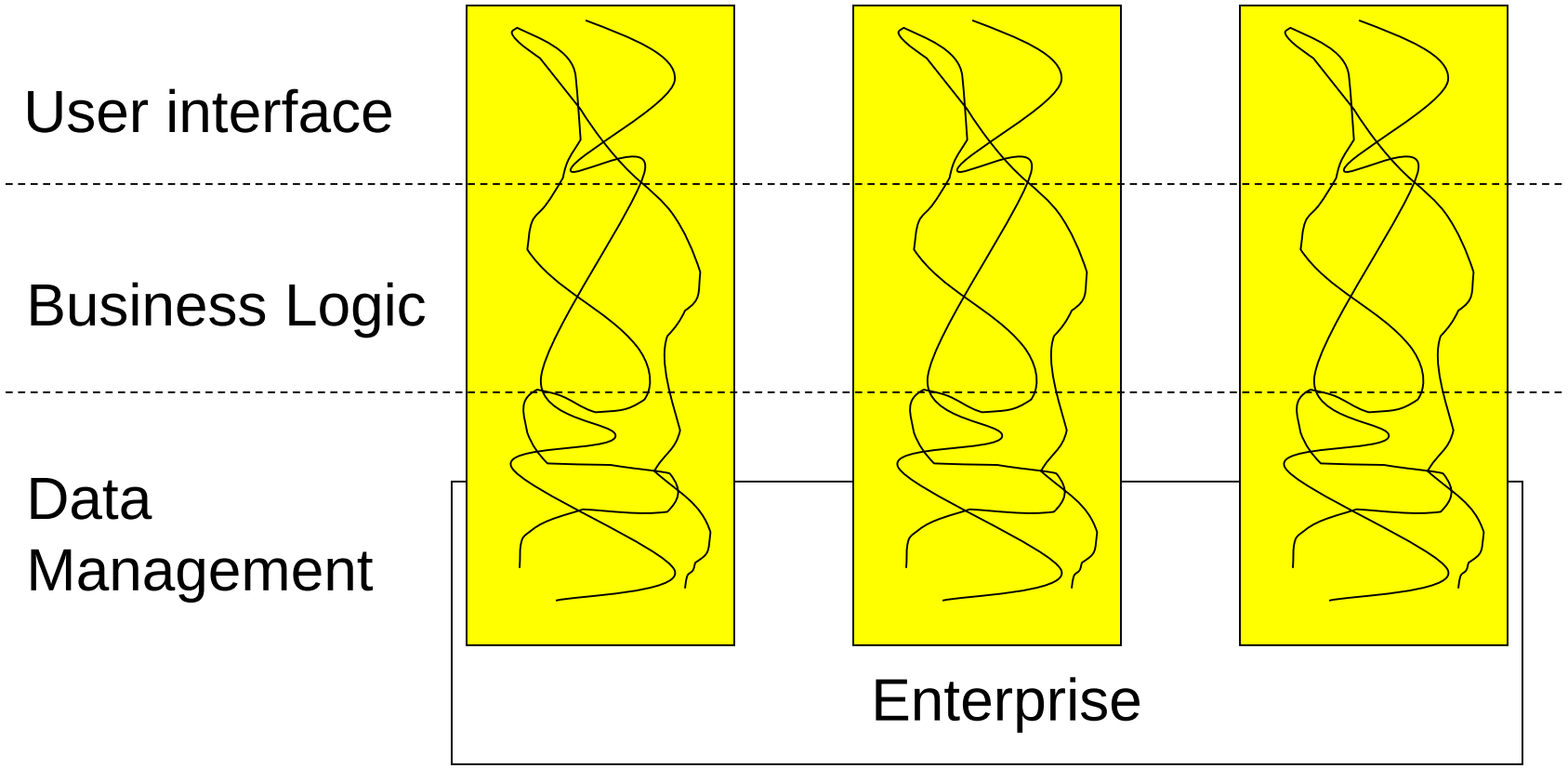
Architetture delle applicazioni

- Monoliti
- Client/server (two-tier)
- Three-tier
- N-tier

Monoliti (IT stovepipe) - I

- I monoliti erano popolari ai tempi dei mainframe
- **I Monoliti**
  - erano pezzi di codice indivisibili
  - controllavano l'intera logica dell'applicazione, dalla gestione (e memorizzazione) dei dati, fino all'interfaccia utente
  - gestivano applicazioni stand-alone
- La popolarità dei monoliti era dovuta a:
  - mainframe adatti a eseguire pochi processi stand-alone, anziché diversi processi comunicanti
  - i database non esistevano ancora

Monoliti (IT stovepipe) - II



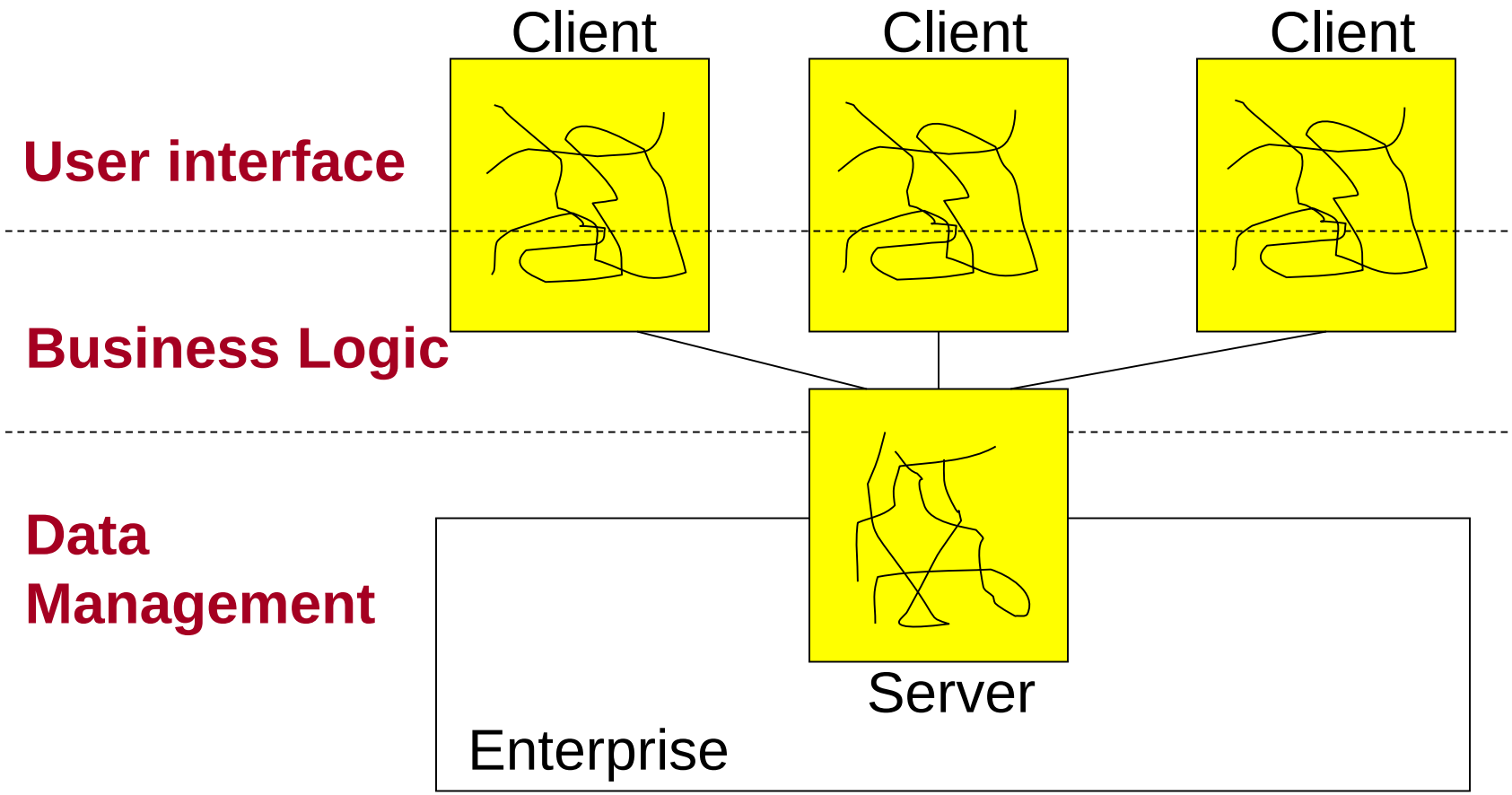
Architetture client/server - I

- Dalla fine degli anni ‘70 alla metà degli anni ’80 si sono diffuse le architetture client-server
  - Diffusione di server più piccoli ed economici dei mainframe, e di workstations e PC
  - Diffusione di RDBMS
- Suddivisione delle applicazioni in due parti:
  - **Il backend (server)** gestisce i database e i compiti di manipolazione dei dati
  - **Il frontend (client)** gestisce l’interfaccia utente

  $\Rightarrow$ maggiore scalabilità rispetto ai monoliti (carico computazionale distribuito sui client)

  $\Rightarrow$ sviluppo più veloce di applicazioni che accedono agli (stessi) dati

Architetture client/server - III



Architetture client/server - II

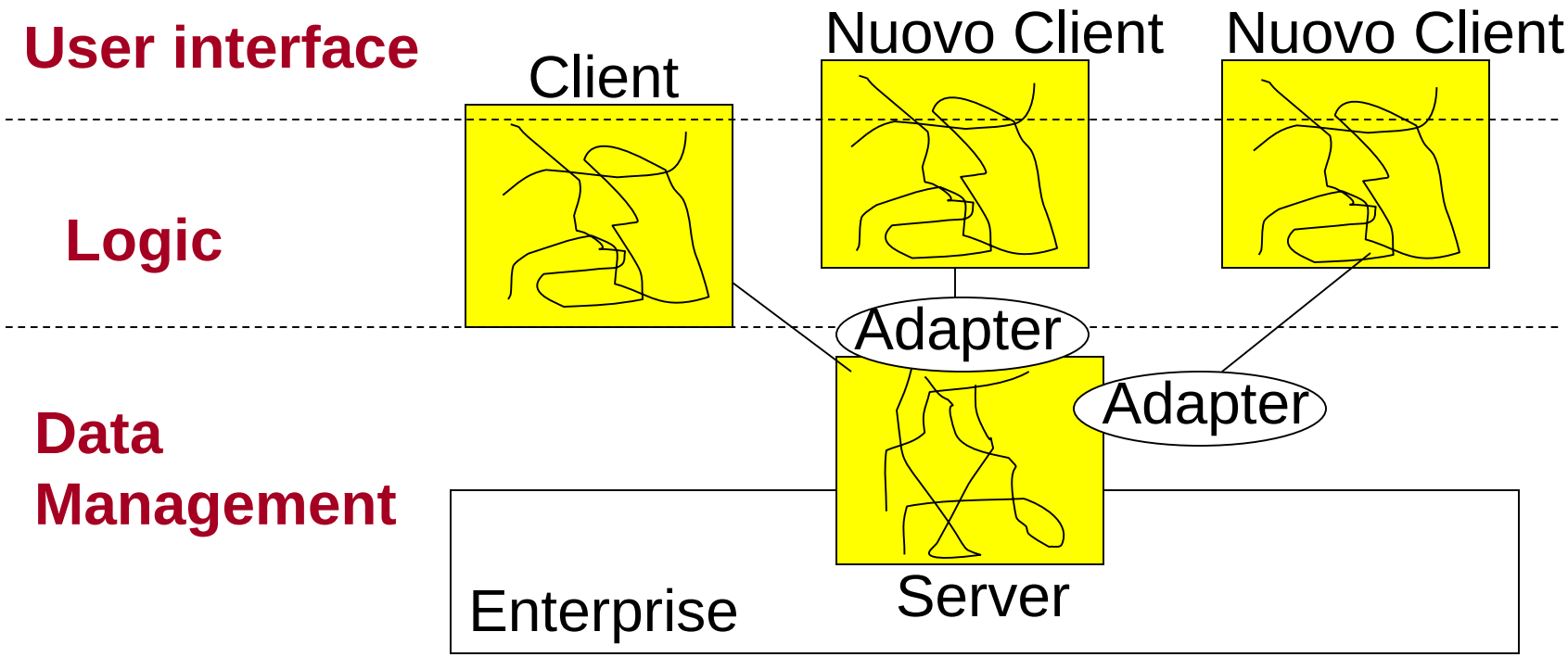
Svantaggi:

- La logica di business è suddivisa tra frontend e backend perchè non viene gestita in una componente separata dell'architettura
- $\Rightarrow$ il client e il server dell'applicazione dipendono l'uno dall'altro →
  - È difficile riutilizzare l'interfaccia utente in un servizio che accede a dati diversi
  - È difficile utilizzare DB da frontend diversi
  - La business logic viene cablata nell'interfaccia utente $\Rightarrow$ un cambiamento della logica implica il cambiamento dell'interfaccia utente

Architetture client/server - IV

Il problema è il mancato riconoscimento dell'importanza della business logic

- Es: servizio accessibile da più device (telefonino, desktop) ⇒ stessa logica, interfaccia diversa



Architetture three-tier - I

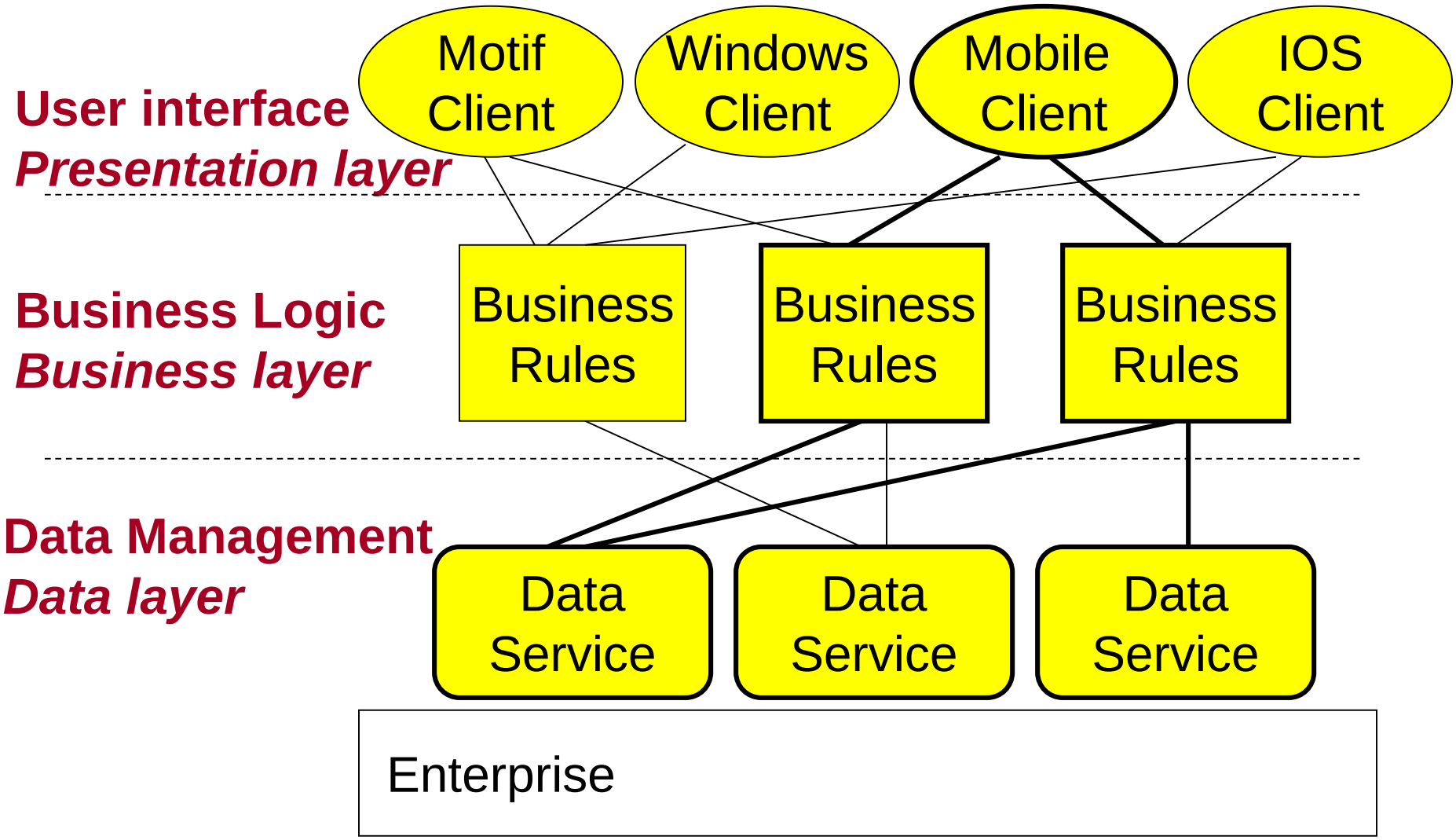
Le architetture 3-tier sono state introdotte all'inizio degli anni '90

- **Esse trattano la business logic in modo esplicito**:
  - ***livello 1*: gestione dei dati (DBMS, documenti XML, …)**
  - ***livello 2*: business logic (processamento dei dati, …)**
  - ***livello 3*: interfaccia utente (presentazione dei dati, servizi)**
- Nessun livello fa assunzioni sulla struttura o sull'implementazione degli altri livelli:
  - Il livello 2 non fa assunzioni sulla rappresentazione dei dati, né sull'implementazione, dell'interfaccia utente
  - Il livello 3 non fa assunzioni su come opera la business logic

Architetture three-tier - II

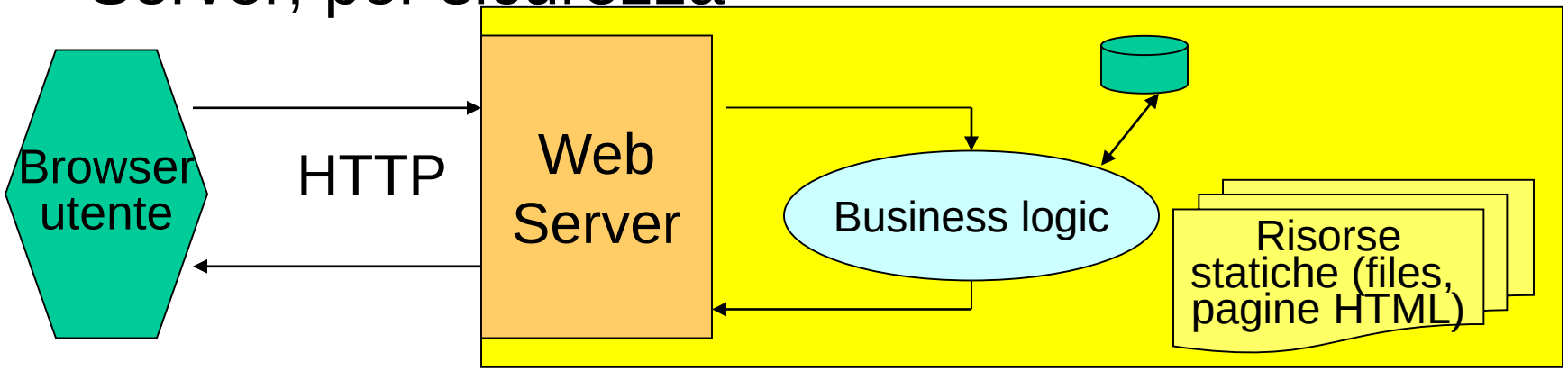
- Non c’è comunicazione diretta tra livello 1 e livello 3
  - L’interfaccia utente non riceve, né inserisce direttamente i dati nel livello di data management
  - La business logic filtra tutti i passaggi di informazione nei due sensi
- I livelli operano senza assumere di essere parte di una specifica applicazione
  - $\Rightarrow$ le applicazioni sono viste come **collezioni di componenti cooperanti**
  - $\Rightarrow$ **ogni componente può essere contemporaneamente parte di applicazioni diverse** (e.g., database, o componente logica di configurazione di oggetti complessi)

Architetture three-tier - III

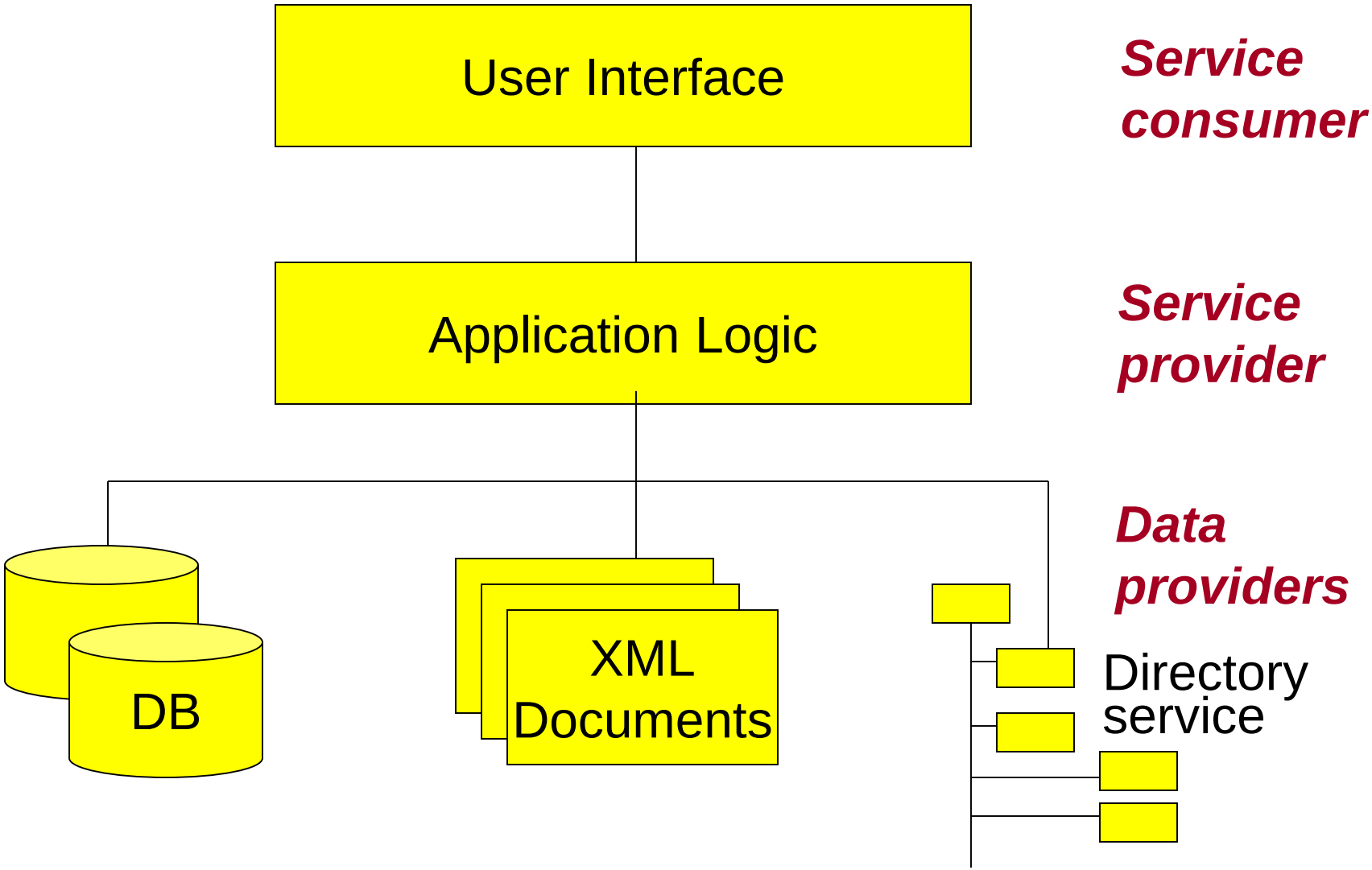


Applicazione Web-based tipica (3-tier)

- **L'interfaccia utente** è gestita sul browser utente (pagine Web statiche e dinamiche)
- **La logica applicativa** è gestita sul server (e.g., tramite Servlet java). Il server si interfaccia al web attraverso il Web Server
- **Il livello dei dati** è su database, eventualmente localizzato su una macchina ≠ da quella del Web Server, per sicurezza



Architetture three-tier - IV



Vantaggi di architetture three-tier - I

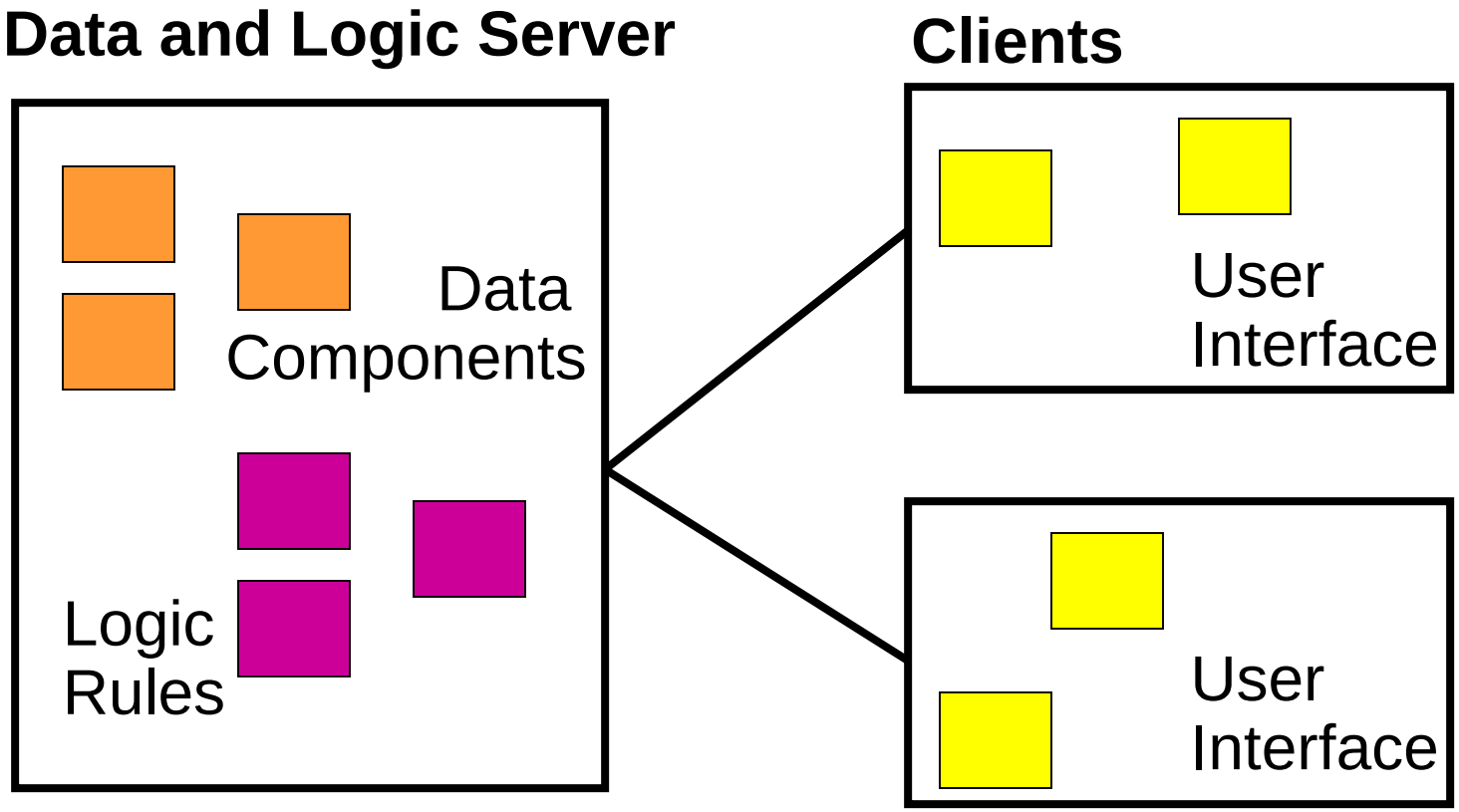
- **Flessibilità** e **modificabilità** dei sistemi formati da componenti separate
  - Le componenti sono utilizzabili in sistemi diversi
  - Modificare una componente non impatta sul resto del sistema (a meno di cambiamenti negli API)
  - La ricerca dei bug è focalizzata (separazione ed isolamento delle funzionalità del sistema)
  - L'aggiunta di funzionalità all'applicazione implica l'estensione delle sole componenti coinvolte, o l'aggiunta di nuove componenti

Vantaggi di architetture three-tier - II

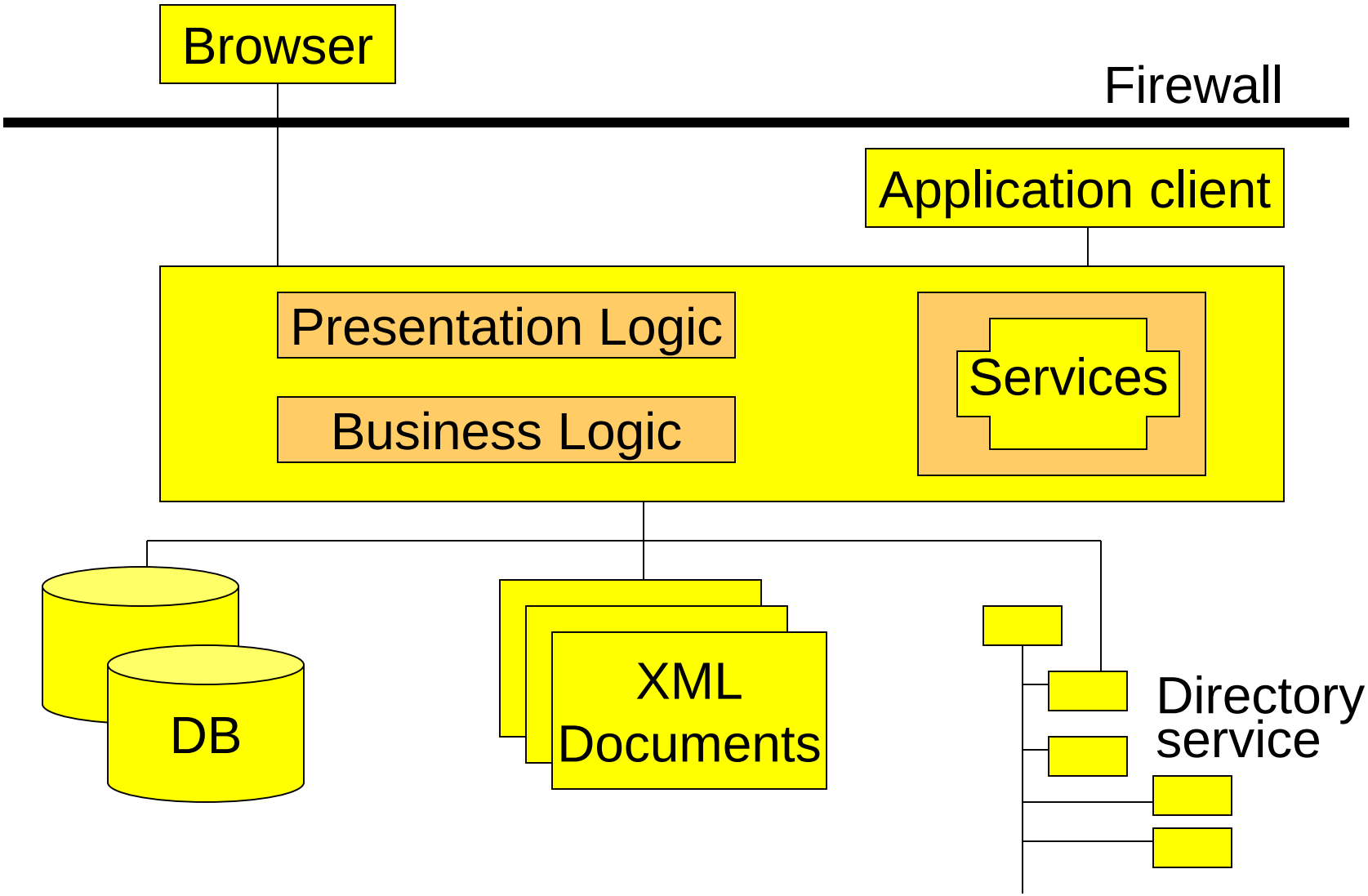
- **Interconnettività**
  - Le API delle componenti superano il problema degli adattatori del modello client server $\Rightarrow$ N interfacce diverse possono essere connesse allo stesso servizio
  - Applicazioni diverse possono accedere ai dati comuni facilmente, usando lo stesso gestore dei dati
- **Gestione di sistemi distribuiti**
  - La business logic di applicazioni distribuite può essere aggiornata senza richiedere l'aggiornamento dei client (e.g., sistemi informativi con alcuni server replicati e client remoti)

Three-tier è concettuale, non fisico

*Si possono implementare architetture **three-tier** su **due** livelli di macchine, o anche su **uno** solo...*



Architetture n-Tier - I



Architetture n-Tier - II

Permettono configurazioni diverse. Elementi fondamentali di queste architetture:

- Interfaccia utente (UI)
  - gestisce l'interazione con l'utente
  - può essere un web browser, minibrowser, interfaccia grafica, …
- Presentation logic
  - definisce cosa la UI presenta e come gestire le richieste utente
- Business logic
  - gestisce le regole di business dell'applicazione

Architetture n-Tier - III

- Infrastructure services
  - forniscono funzionalità supplementari alle componenti dell’applicazione (messaging, supporto alle transazioni, …)
- Data layer:
  - livello dei dati dell’applicazione