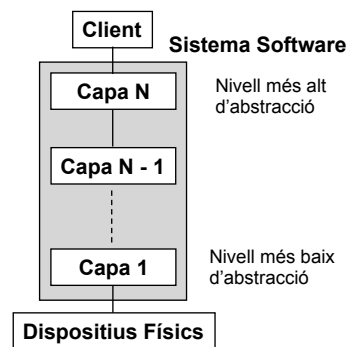


Arquitectura en capes

- Estructura i comportament
- Consideracions en la definició de l'arquitectura
- Beneficis i inconvenients
- Aplicació als Sistemes d'Informació
 - Arquitectura en tres capes
 - Vista de desenvolupament de l'arquitectura en tres capes
 - Vista física (de desplegament) de l'arquitectura en tres capes
 - Especificació dels serveis de cada capa
- Bibliografia

Estructura

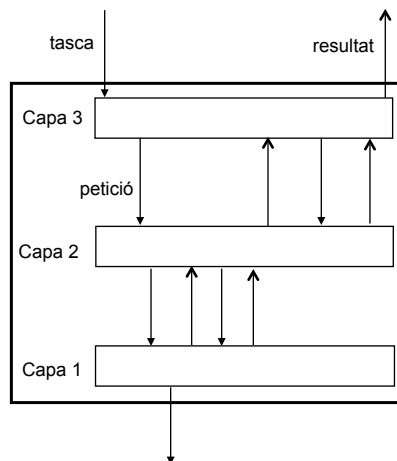
- Estructurar el sistema en un nombre apropiat de capes.
- Tots els components d'una mateixa capa han de treballar al mateix nivell d'abstracció.
- El nombre de capes i el nivell d'abstracció és una decisió a prendre.
- Els serveis que proporciona una capa utilitzen serveis proporcionats per la capa inferior i ofereixen serveis a les capes superiors.
- Els serveis d'una capa poden dependre d'altres serveis en la mateixa capa.



Comportament

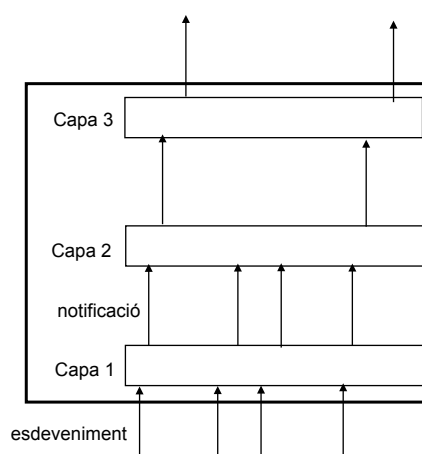
Comunicació de dalt cap a baix

Un usuari realitza una petició d'una tasca a la capa superior i rep un resultat



Comunicació de baix cap a dalt

Un dispositiu físic detecta l'ocurrència d'un esdeveniment a la capa inferior i ho notifica a les superiors



Consideracions a la definició de l'arquitectura

- Definir el criteri d'abstracció i el nombre capes
- Especificar els serveis de cada capa:
 - Cap component pot estar repartit en dues o més capes.
 - Capa superior conté serveis a nivell de sistema.
 - Pocs serveis en les capes inferiors.
- Especificar una interfície per cada capa:
 - Normalment, capa com a caixa negra per la capa superior
 - Alternatives: caixa blanca, caixa gris
- Estructurar les capes individualment
- Especificar la comunicació entre capes adjacents:
 - Model "empenta": la informació es comunica en la petició del servei
 - Model "estirada": el servei demanat estira la informació de la capa superior
- Dissenyar una estratègia de tractament d'errors.
 - Tractar errors en la capa on es detecten o a capes superiors

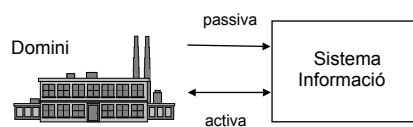
Beneficis i Inconvenients

- Beneficis:
 - Canviable, Reusable, Portable, Provable
- Inconvenients:
 - Eficiència
 - Feina innecessària o redundant
 - Dificultat en establir la granularitat i nombre de capes



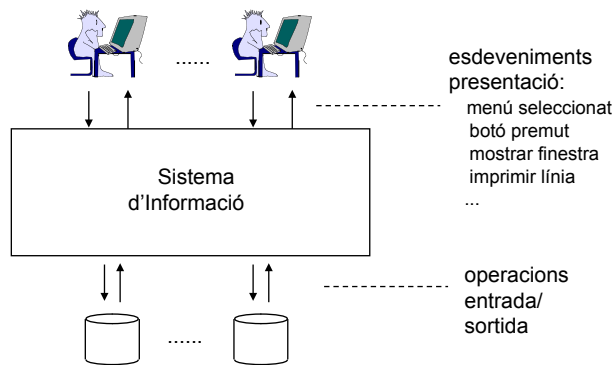
- **Arquitectura en capes relaxat**
 - Una capa pot usar els serveis de qualsevol capa inferior
 - Tots o només part dels serveis de la capa (opacitat parcial)
 - Conseqüències:
 - Possible guany en flexibilitat i eficiència
 - Possible pèrdua en la canvialilitat, reusabilitat

Aplicació als Sistemes d'Informació

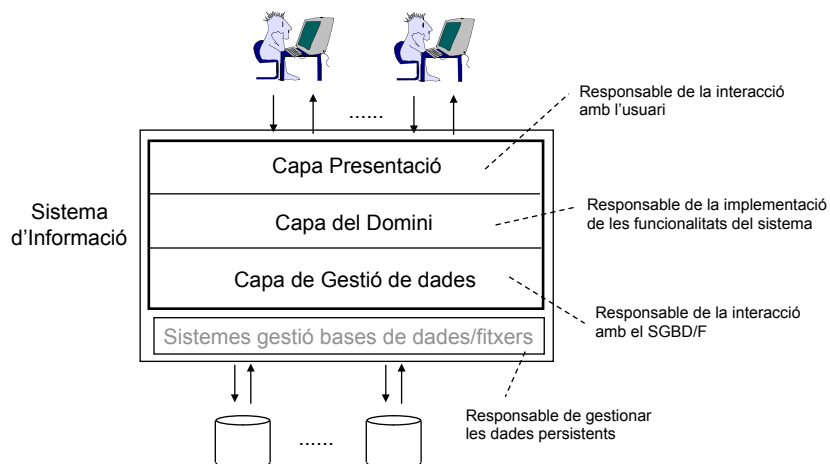


- Funcionalitat del SI:
 - Mantenir una representació consistent de l'estat del domini:
 - Capturar els esdeveniments que ocorren al domini
 - Actualitzar l'estat del sistema d'informació com a conseqüència d'aquests esdeveniments
 - Assegurar la consistència de la representació
 - Respondre a consultes sobre l'estat del domini.
 - Produir reaccions quan es donen certes condicions predefinides.

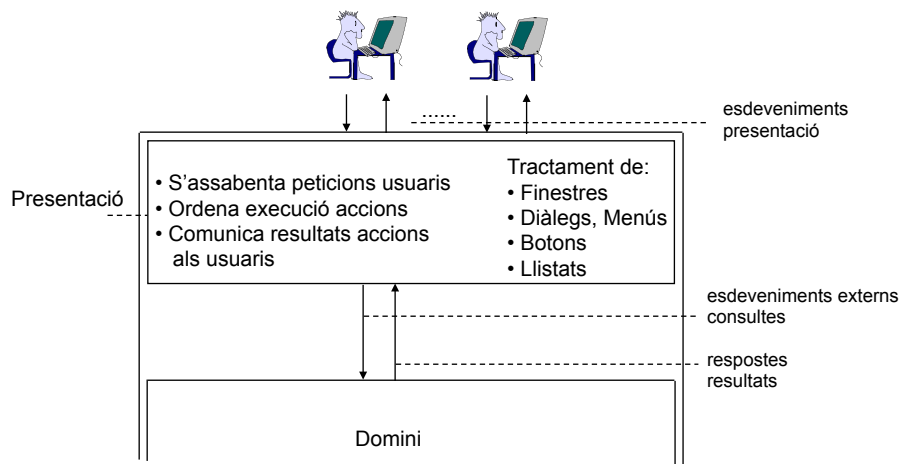
Context dels Sistemes d'Informació



Arquitectura en capes d'un Sistema d'Informació: Definició de criteris d'abstracció i nombre de capes

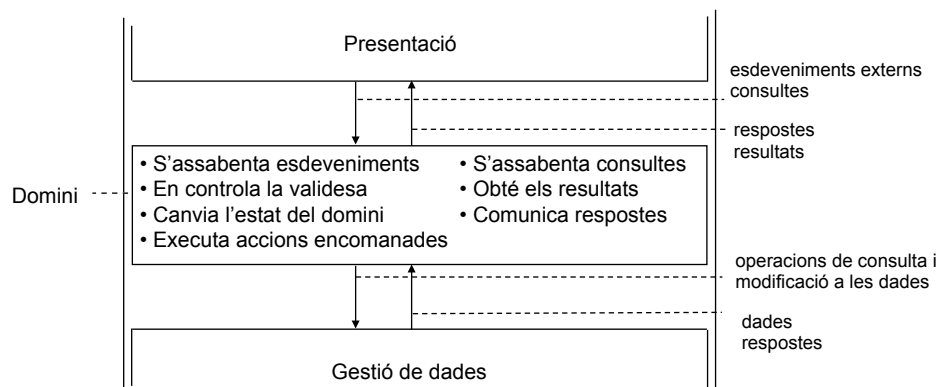


Arquitectura en tres capes d'un Sistema d'Informació: Capa de Presentació



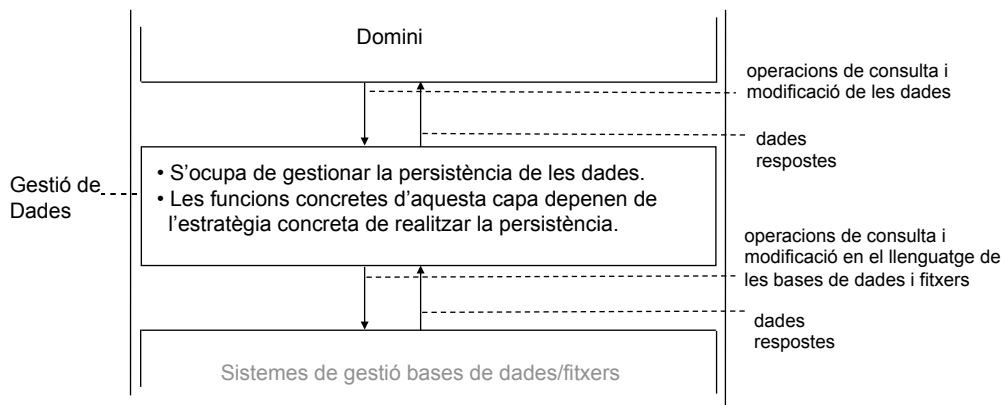
La Capa de Presentació coneix com presentar les dades a l'usuari, però ignora quines transformacions cal fer per donar resposta a les peticions de l'usuari

Arquitectura en tres capes d'un Sistema d'Informació: Capa del Domini



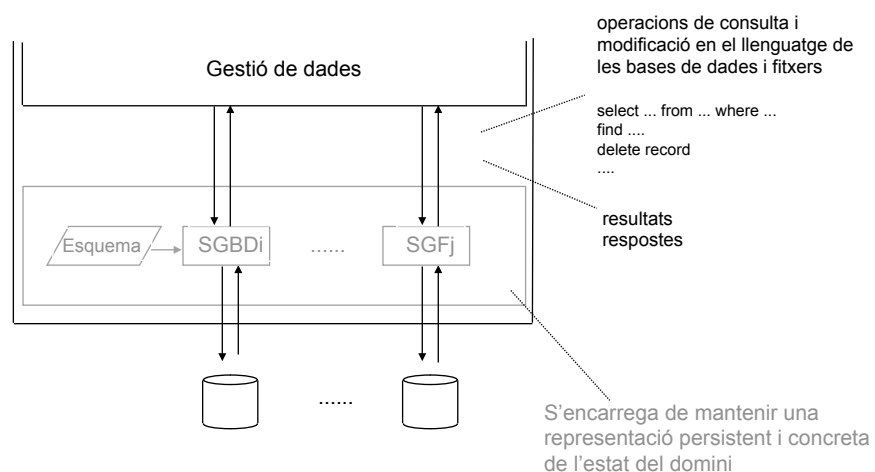
La Capa de Domini coneix com satisfer les peticions de l'usuari, però ignora on es guarden les dades i com es presenten a l'usuari

Arquitectura en tres capes d'un Sistema d'Informació: Capa Gestió de Dades

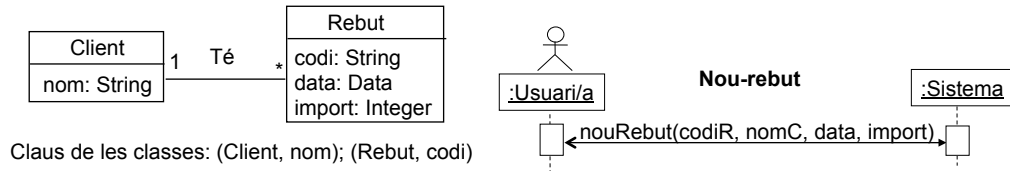


La capa de Gestió de Dades coneix on i com estan emmagatzemades les dades, però desconeix com tractar-les

Arquitectura en tres capes d'un Sistema d'Informació: Els sistemes de gestió de bases de dades/fitxers



Exemple: models de partida



context nouRebut (codiR: String, nomC: String, data: Data, import: Integer)

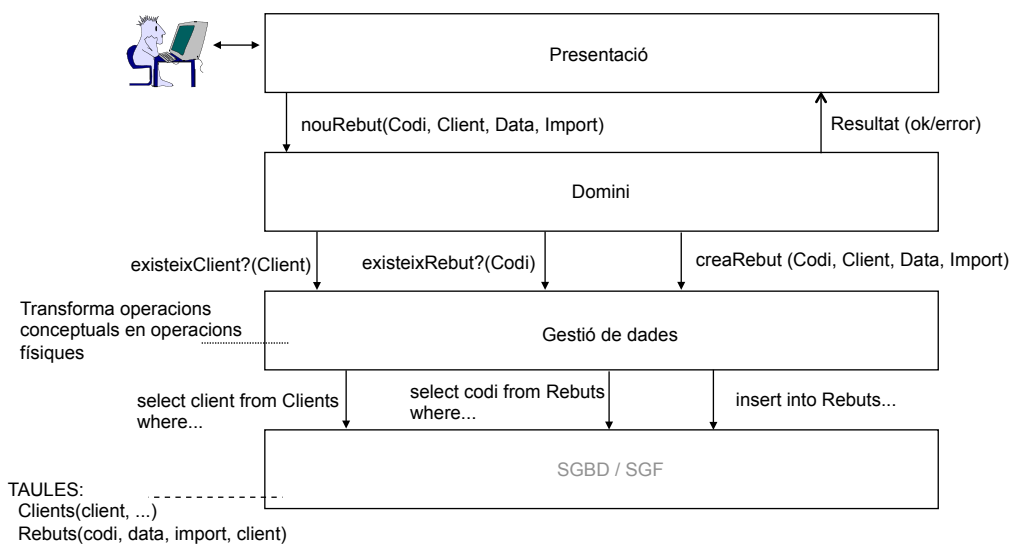
-- enregistrar un nou rebut per un client

pre: 1.1 existeix el client amb nomC

post: 2.1 alta d'una instància de rebut R amb R.codi=codiR, R.data=data i R.import=import

2.2 alta d'una instància de l'associació 'Té' que associa el rebut R i el client amb nom=nomC.

Exemple: comunicació entre capes



Vista física (desplegament) de l'arquitectura en capes

- Desplegament de l'arquitectura (assignació d'elements de disseny a elements hardware)
- De vegades poden fusionar-se capes adjacents i/o dividir capes per tal d'assolir els requisits no funcionals
- Opcions:
 - Totes les capes estan a la mateixa localització física (sistema centralitzat):
 - . 1 capa => Presentació-Domini-Gestió Dades
 - . 2 capes => Presentació-Domini + Gestió Dades (SI fet amb eines de desenv. ràpid)
 - . 2 capes => Presentació + Domini-Gestió Dades (SI gestió bàsic de dades)
 - . 3 capes => Presentació + Domini + Gestió Dades
 - Les capes estan en diferents localitzacions físiques (sistema distribuït)
 - . Cada capa en una localització diferent
 - . Dues capes en una mateixa localització i l'altre en una localització diferent
 - . Repartició d'una o més capes en localitzacions diferents

Tipus de sistemes distribuïts

Algunes fronteres habituals són:

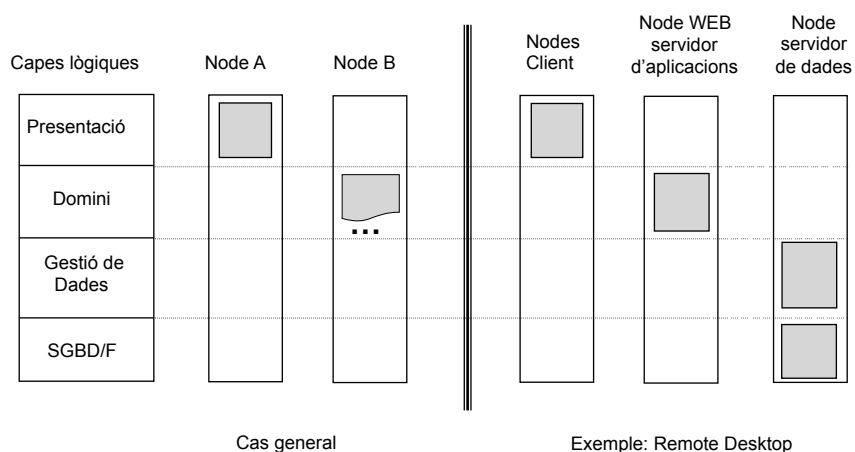
- Presentació remota → típic sistema RPC
- Presentació distribuïda → típic sistema Web
- Dades remotes → protocol SGBD
- Dades distribuïdes → distribució ad-hoc (e.g., actualització de 2 BDs); replicació
- Distribució de la capa de domini → típic sistema SOA

Freqüentment dins un mateix sistema distribuït es tracen diverses fronteres

També trobem algunes situacions que condicionen el traç de les fronteres. p.e.:

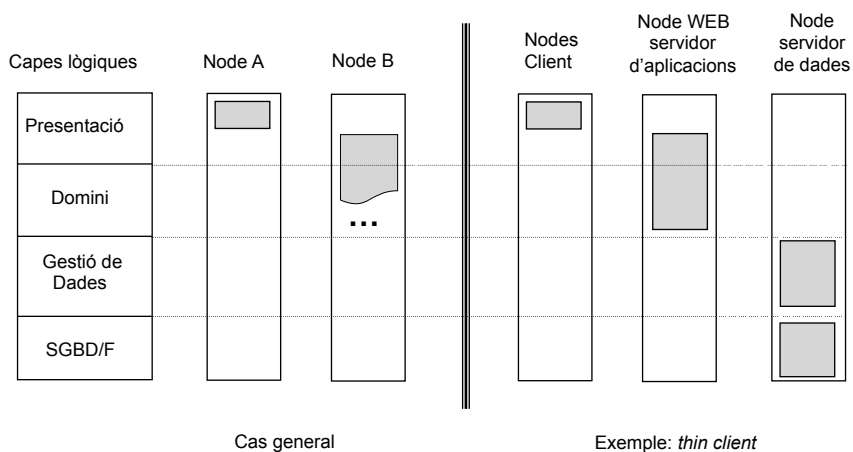
- Integració de sistemes heretats
- Integració de sistemes o serveis de tercers

Presentació remota



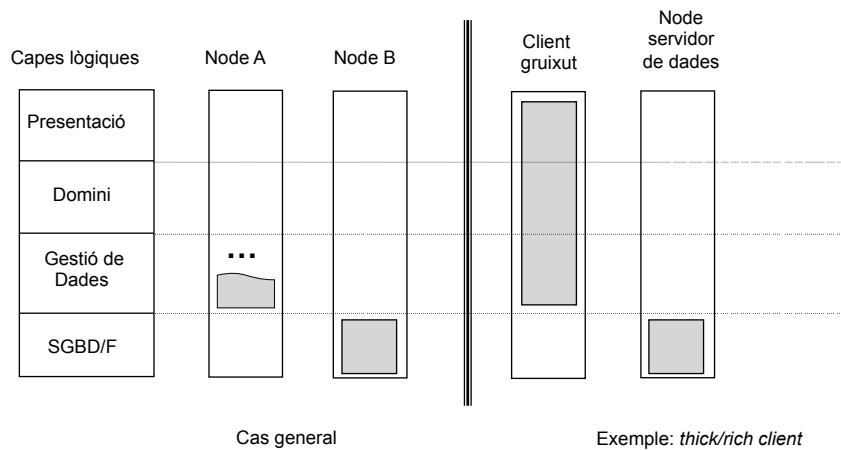
Remote Desktop applications are based on remote procedure call protocol where a client cause execution of procedures or actios in another computer.

Presentació distribuïda



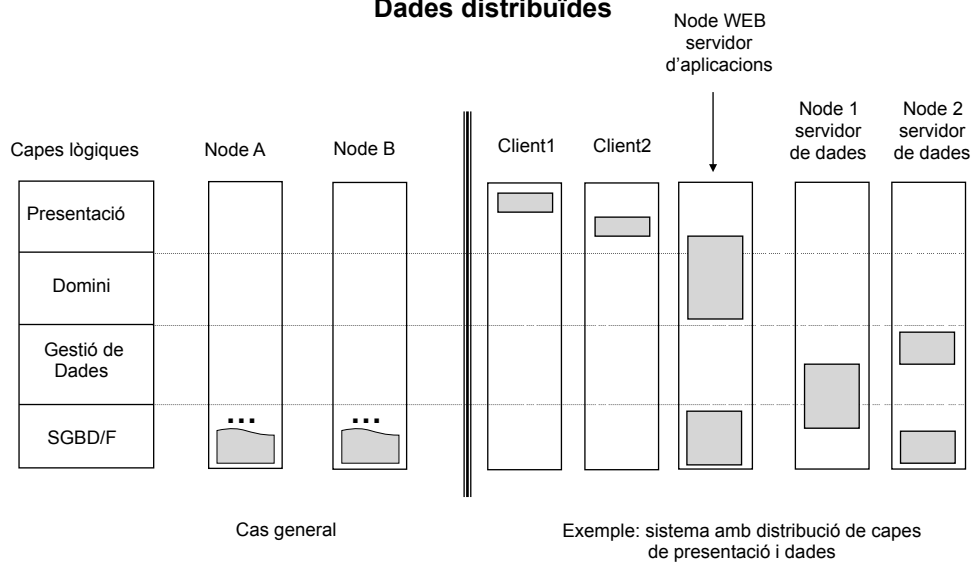
A thin client architecture describes a computer heavily dependent on a server's applications. It generally does as little processing as possible and relies on accessing the server each time input data needs to be processed or validated.

Dades remotes

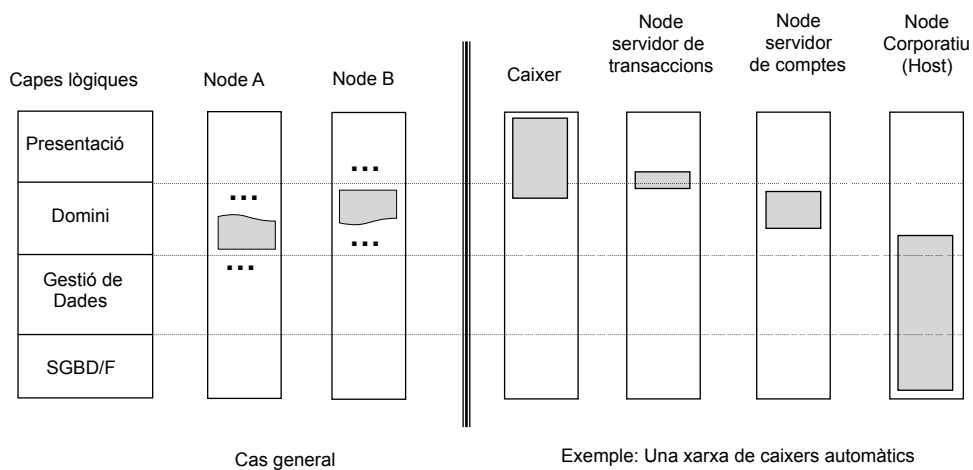


A rich client architecture provides rich functionality to client.
It is often characterised by the ability to perform some functions without connection to the server.

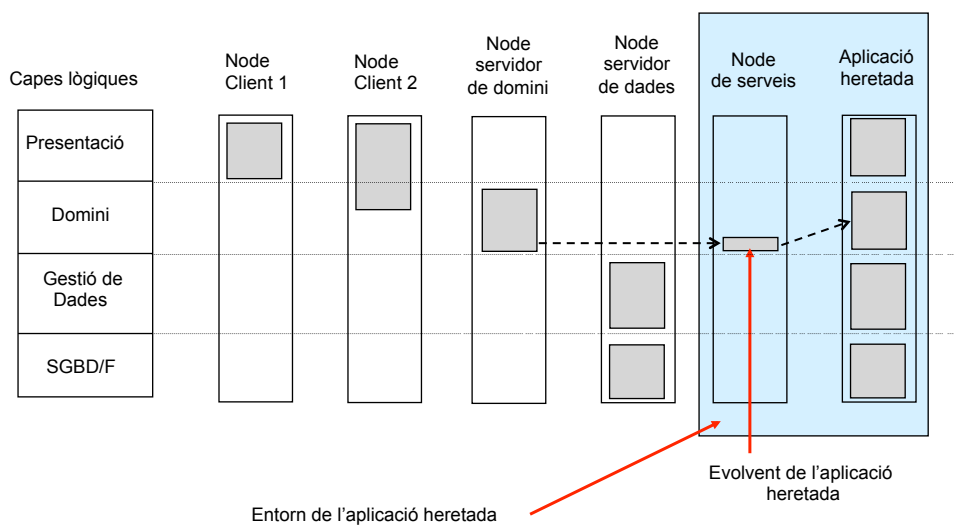
Dades distribuïdes



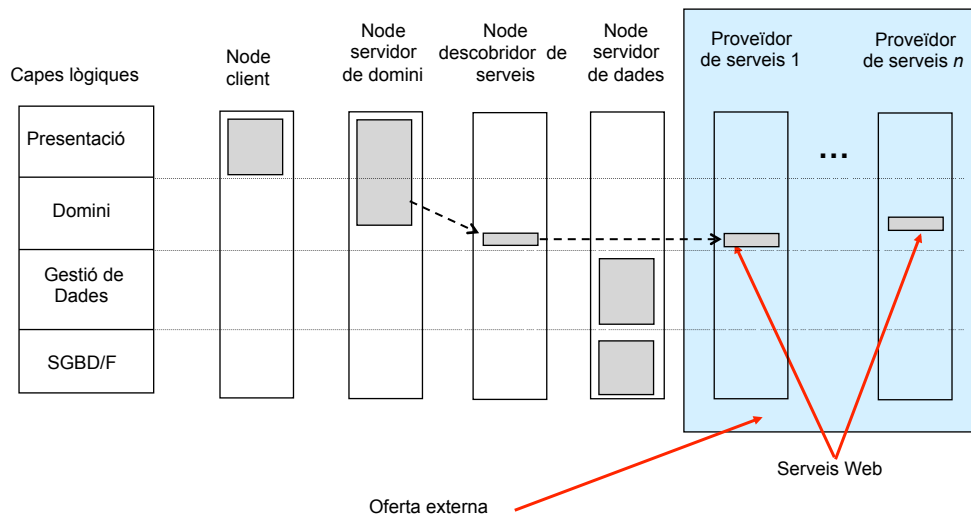
Domini distribuït



Exemple: integració de sistemes heretats Utilització de tècniques d'evolvent



Exemple: integració de sistemes de tercers Serveis Web



Bibliografia

- *Pattern-oriented Software Architecture. A System of patterns*
F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal
John Wiley & Sons, 1996. Pàgines 31-51
- *Analysis Patterns*
Martin Fowler
Addison-Wesley, 1997, cap. 12.
- *Enginyeria del Software 2 (Pla Estudis 2003).*
Transparències de l'assignatura de ES2
Facultat Informàtica de Barcelona