

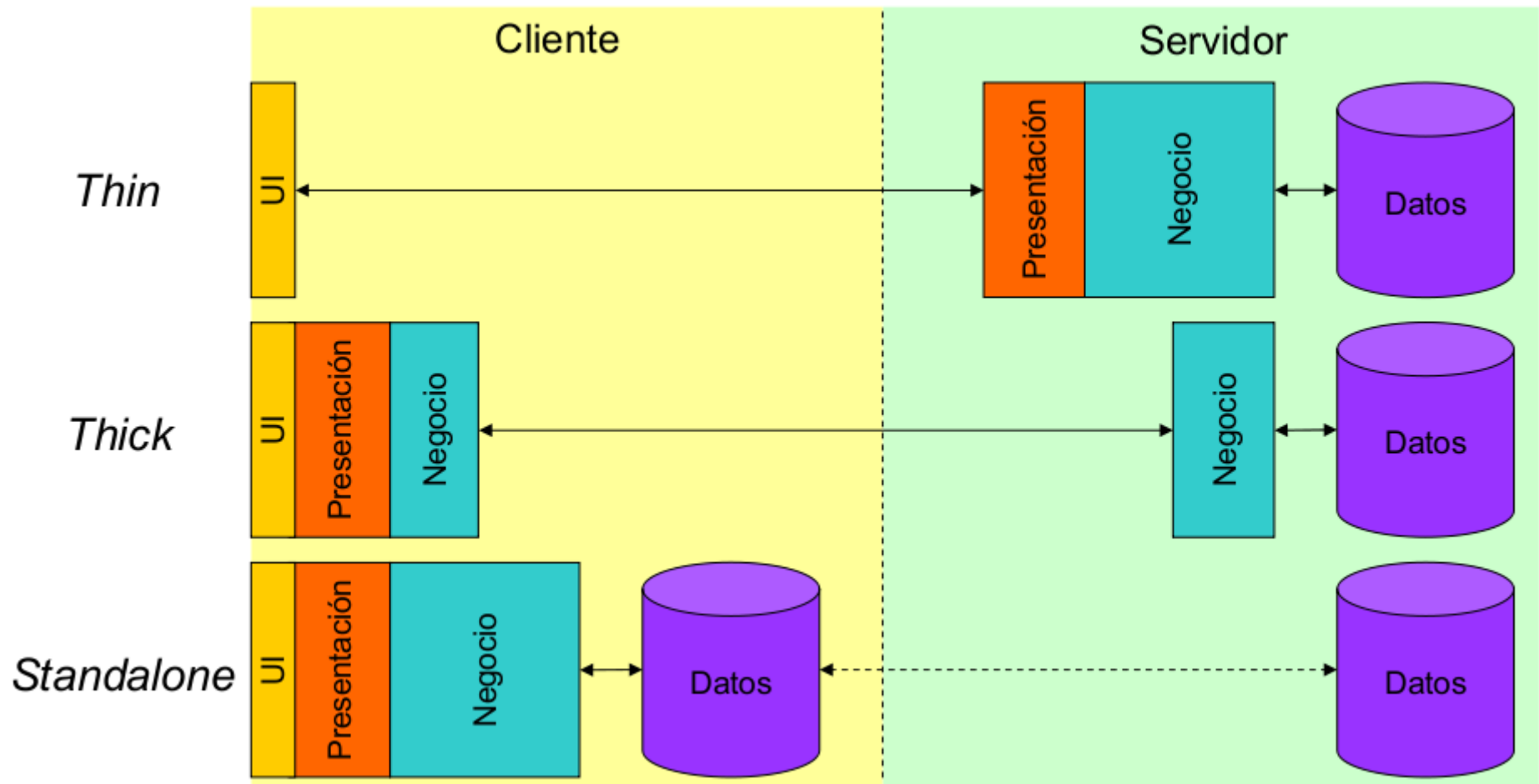


## Desenvolupament d'aplicacions mòbils



# Tipus de clients

- ♦ **Classificació segons la carrega entre client i servidor**





# Clients thin (clients lleugers)

- ♦ **Tot els processos es realitzen al servidor**
  - ♦ El client només s'encarrega de la interfície de l'usuari
  - ♦ Sovint no calen aplicacions client específiques
    - Navegador + Documents web (p. ex. XHTML)
    - Els documents HTML no són adequats per a mòbils
    - Nous formats
      - WML, XHTML-MP, cHTML, ...
    - **Microbrowser**: Navegadors específics per a aquests formats



## Clients thick (clients pesats)

- ♦ **Les aplicacions es descarreguen i s'executen al mòbil**
  - ♦ Treballa de forma coordinada amb el servidor
  - ♦ Es realitza al client tot el processament possible.
  - ♦ Aplicacions dedicades a tasques concretes
  - ♦ Només s'intercanvia informació entre client i servidor
    - HTML no és el millor format per intercanviar informació
    - Altres protocols de comunicació:
      - Serveis Web
      - Remote Procedure Control (RPC)
      - Intercanvi binari de dades (p. ex. amb sockets o connexió directa a un Servlet)



# Tecnologies per a clients thick

## ♦ **Sistemes operatius mòbils**

- ♦ Symbian OS, Palm OS, Windoze Pocket PC, Android...
- ♦ Poca portabilitat.
- ♦ Cal aprendre noves API específiques per a cada SO

## ♦ **Runtime Environments**

- ♦ S'executen sobre un sistema operatiu (firmware) mòbil
- ♦ BREW
- ♦ J2ME



# Thick o Thin?

## ♦ Thin

- ♦ Necessita connectar a la xarxa per a descarregar cada document
  - Velocitat de descàrrega lenta. Preu de la connexió
- ♦ Limitat a les possibilitats del navegador
- ♦ Fàcil de mantenir

## ♦ Thick

- ♦ Costos de manteniment
- ♦ Interfície d'usuari més flexible
- ♦ Minimitza el tràfic de xarxa (s'intercanvia la informació imprescindible)
- ♦ Pot funcionar sense connexió



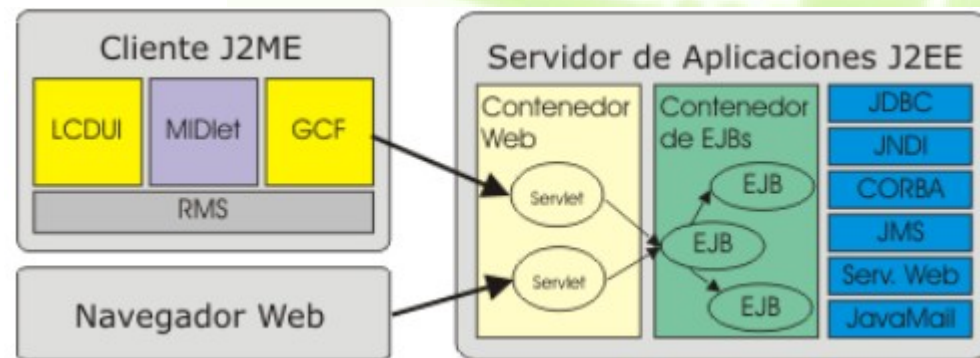
# Tecnologies per a clients thick

## ♦ El MIDlet client utilitza:

- ♦ GCF per a comunicar-se amb el servidor web
- ♦ LCDUI per a la interfície d'usuari
- ♦ RMS per emmagatzemar dades locals

## ♦ El servidor pot ser una aplicació J2EE

- ♦ Un *Servlet* que es comuniqui amb el client J2ME
- ♦ Es pot definir un altre *Servlet* per accedir per web
- ♦ Es pot tornar a utilitzar la lògica de negoci per a tots dos *Servlets*





# Exemple thick. Midlet+Servlet

- ♦ Implementar un MIDlet que es connecti a un Servlet per tal d'obtenir informació

- ♦ HelloServlet

```
/**
 * Exemple de servlet
 */
public class HelloServlet extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws IOException, ServletException {
        response.setContentType("text/plain");
        PrintWriter out = response.getWriter();
        out.println("Hola Mon!");
    }
}
```

- ♦ Exercici

- ♦ Seguiu les instruccions de la wiki per a crear un nou Servlet





# Exemple thick. Midlet+Servlet

## ◆ HelloServletMIDlet

```
public class FirstMidletServlet extends MIDlet {
    private Display display;
    String url =
    "http://somesite.com/servlet/HelloServlet";
    public FirstMidletServlet() {
        display = Display.getDisplay(this);
    }
    //Initialization. Invoked when MIDlet activates
    public void startApp() {
        try {
            invokeServlet(url);
        } catch (IOException e) {
            System.out.println("IOException " + e);
            e.printStackTrace();
        }
    }
    ...
}
```

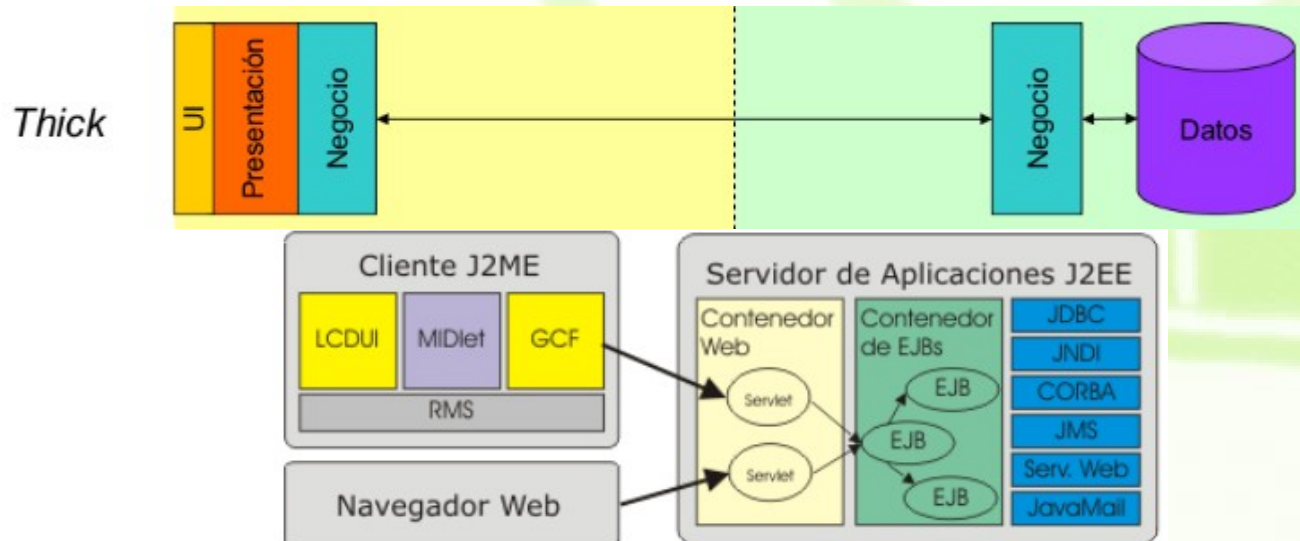
```
void invokeServlet(String url) throws IOException {
    HttpURLConnection c = null; InputStream is = null;
    StringBuffer b = new StringBuffer(); TextBox t = null;
    try {
        c = (HttpURLConnection)Connector.open(url);
        c.setRequestProperty("User-Agent",
            "Profile/MIDP-1.0 Configuration/CLDC-1.0");
        is = c.openDataInputStream();
        int ch;
        while ((ch = is.read()) != -1) {
            b.append((char) ch);
        }
        t = new TextBox("First Servlet", b.toString(), 1024, 0);
    } finally {
        if(is != null) { is.close(); }
        if(c != null) { c.close(); }
    }
    display.setCurrent(t);
}
}
```



# ProfesServlet. GestioAules

## ♦ ProfesServlet

- ♦ Adaptem Hola Mon! Servlet per tal de mostrar la llista de professors
- ♦ Podem codificar per codi font o podríem obtenir la llista de professors d'una base de dades
- ♦ En **Java EE** a més podríem separar la **lògica de negoci** del Servlet creant un **Enterprise Java Bean (EJB)**





## Reconeixement 3.0 Unported

### Sou lliure de:



copiar, distribuir i comunicar públicament l'obra



fer-ne obres derivades

### Amb les condicions següents:



**Reconeixement.** Heu de reconèixer els crèdits de l'obra de la manera especificada per l'autor o el llicenciador (però no d'una manera que suggereixi que us donen suport o rebeu suport per l'ús que feu l'obra).

- Quan reutilitzeu o distribuïu l'obra, heu de deixar ben clar els termes de la llicència de l'obra.
- Alguna d'aquestes condicions pot no aplicar-se si obteniu el permís del titular dels drets d'autor.
- No hi ha res en aquesta llicència que menyscabi o restringeixi els drets morals de l'autor.

Advertiment

Els drets derivats d'usos legítims o altres limitacions reconegudes per llei no queden afectats per l'anterior  
Això és un resum fàcilment llegible del text legal (la llicència completa).

<http://creativecommons.org/licenses/by/3.0/deed.ca>