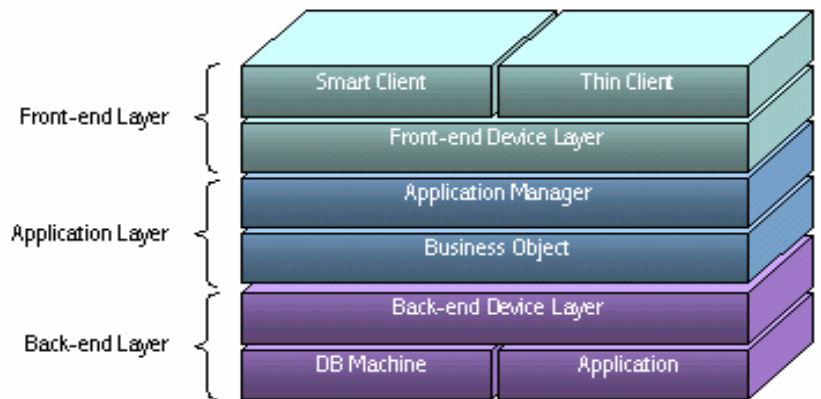


**x-Brick** è un framework, sviluppato in architettura three tier, che consente di minimizzare i tempi di realizzazione del software, concentrando l'attenzione del programmatore sulla complessità della logica applicativa e lasciando la gestione di tutte le problematiche di "presentation" e di accesso ai dati, preliminari allo sviluppo vero e proprio, ad un middleware che realizza lo standard "implicito" ed omogeneo per tutta l'applicazione.

Le applicazioni basate su questo framework presentano: un basso costo di manutenzione, sia in termini di interoperabilità delle risorse allocate sul progetto (le classi definiscono un pattern da seguire durante tutto il ciclo di vita del software), sia in termini di distribuzione del software (l'applicazione è priva di componenti da installare in periferia: viene infatti richiesto solo un browser; non viene fatto alcun uso di applet e di plug-in); una naturale adattabilità agli ambienti Server più diffusi, in termini di Sistema



Operativo, di Web Server e di DBMS; infine un'alta flessibilità per adeguarsi alle evoluzioni delle organizzazioni aziendali e alle logiche di presentazione dei siti che ospitano le componenti applicative.

Tra i servizi messi a disposizione dalle diverse componenti previste dalla architettura (Front-end, Application e Back-end) troviamo: la configurazione delle organizzazioni e delle applicazioni, la profilazione degli utenti interni ed esterni; la gestione della sicurezza, sia del software che delle comunicazioni tra componenti; la localizzazione dell'installazione; logiche di localizzazione; l'indipendenza dalla fonte dati; la remotizzazione; la transazionalità; il balancing; la distribuzione dei messaggi; tracing delle modifiche e log applicativi; la console applicativa; supporto nell'accesso a basi dati relazionali (SQL embedded e stored procedures); la gestione della configurazione; supporto nella gestione di documenti XML; su supporto della comunicazione con un Mainframe via APPC; supporto nell'utilizzo della tecnologia smart-card; supporto nella generazione della GUI, basata su HTML o XWindow.

**L'architettura prevede una netta separazione delle tre diverse componenti di una qualunque applicazione: presentazione dei dati, logica applicativa e di validazione, accesso alle fonti dati.**

#### **Front-end Layer**

Allo strato di Front-end attiene tutta la logica di visualizzazione e di presentazione dei dati. Deve quanto più possibile limitarsi alla gestione dell'interfaccia utente, delegando le logiche di validazione e di salvataggio alla componente Application.

La componente di Front-end si limita a richiedere servizi alla componente server e ad utilizzare eventuali logiche di interfaccia (ad esempio: ordinamenti, filtri, ...) e metafore di interazione standard (ad esempio: liste, alberi, zoom di campo e di finestra, ...), messe a disposizione dal framework realizzato.

#### **Application Layer**

L'Application Layer gestisce tutta la logica applicativa e di validazione prevista dall'applicativo da realizzare. Collega le componenti di Front-end e di Back-end, facendosi carico principalmente delle seguenti attività:

- ricevere le richieste di informazioni delle componenti di Front-end e girarle alla componente di Back-end;
- ricevere dalla componente di Back-end le informazioni richieste e inviarle alla componente di Front-end;
- ricevere le modifiche ai dati da memorizzare e gestire la validazione e la richiesta di salvataggio delle modifiche validate;
- gestire le chiamate al motore di scripting per la gestione dei processi di validazione e di salvataggio di tipo "custom";
- gestire gli errori di validazione derivanti dalla richiesta di salvataggio delle modifiche pervenute;
- coordinare l'interazione tra tutte le componenti server presenti sull'Application Layer per gestire correttamente tutti i servizi aggiuntivi previsti dall'architettura del framework.

#### **Back-end Layer**

Allo strato di Back-end compete tutta la gestione relativa al recupero dei dati dalle fonti definite ed al salvataggio delle modifiche apportate ai dati stessi.

Lo strato di Back-end non aggiunge nessuna logica applicativa, di validazione e di salvataggio: si limita a rendere permanenti le modifiche apportate così come gli vengono passate.

**Sono disponibili package per lo sviluppo in ambiente Java, e sorgenti per lo sviluppo in ambiente COBOLII CICS.**

## j-Brick : x-Brick for Java

- ☞ Sviluppo di applicazioni Web based.
- ☞ Rapida implementazione interfaccia HTML per manipolazione dati
- ☞ Rapida implementazione interfaccia HTML per presentazione dati.
- ☞ JDK 1.2.2 o superiore.
- ☞ Application Object, Java 99.9%
- ☞ Client basato su browser (Servlet, JSP)
- ☞ Sicurezza (JSA, Repository)
- ☞ Componenti orizzontali: XML, Accesso fonte dati, Generazione codice HTML.

### Package:

- ☞ xbrick.misc  
Trace delle operazioni, Configurazione della applicazione, Presentazione dei dati, Elaborazioni asincrone, Gestione di file di log.
- ☞ xbrick.security  
Riconoscimento dell'utente, Gestione delle abilitazioni, Supporto nella gestione della firma digitale
- ☞ xbrick.businessobject
  - Logiche di astrazione.  
Struttura dati indipendente dalla fonte, Accesso alla fonte dati (Database, transazioni Host via APPC, XML), Personalizzazione ed estensione.
  - Servizi

- Validazione, Decodifica e lista valori, QBE e gestioni standard
- ☞ xbrick.servlet
  - Logiche di astrazione  
Struttura della pagina HTML, Personalizzazione ed estensione
  - Servizi  
Presentazione, Generazione del codice HTML, Definizione della componente JS utilizzata
- ☞ xbrick.htmlhelper  
Logiche di astrazione (Tag, Attributi), Servizi (Generazione del codice HTML)
- ☞ xbrick.jsphelper  
Script a supporto dello sviluppatore, global.js, stato delle modifiche, presentazione banner, tree view, cookie, ...
- ☞ xbrick.rmi
  - Definizione interfaccia  
Esposta dal componente ospitato da un Application Layer, Utilizzata dal componente di Front-end
  - Servizi  
Regole di naming, Stato RMI Registry, Applicazioni RMIServer e RMIClient.
- ☞ xbrick.xml  
Wrapper XML parser (JAXP 1.1), Servizi di Navigazione, Validazione, Lettura e salvataggio.

