Architecture Design For Web-Based Application Systems

  Graphic Web Browsers + Hyperlinks + HTML files.


Application: Only for documentation sharing and accesses.
  Static Web Site construction.

Pros: Simple and easy to set up.

Cons: Very server processing functions
  No dynamic HTML generation.
  No support for user interaction
  Not easy to maintain and manage
**Architecture Design For Web-Based Application Systems**

- **(1995-1996) Interactive HTML-based Web Systems:**
  Graphic Web Browsers + Hyperlinks + HTML files
  Plus: Forms, Tables, CGI,
  and Secured Transactions: SSL, S-HTTP, Firewalls

**Features:** Supporting user interactions and requests
  HTML-based client, CGI-based Server program
  Secured communications and transactions

**Applications:** DHTML web sites, Banking, On-line systems

**Pros:** Secured communications.
  Dynamic HTML generation.
  User friendly due to user interactions.

**Cons:** Slow speed, limited client functions and GUI support.
  Unstructured server programs with low concurrence
Architecture Design For Web-based Application Systems


 Client: Web Browser + HTML + JavaScript,
 Server: Web Server + CGI + Gateway Programs + Database Server
 Plus secured transaction based on secured protocol and firewalls

Features: Supporting better user interactions and requests
          HTML/JavaScript client, CGI-based Server program
          Secured communications and transactions

Applications: Web-base application systems, tools, group-ware, On-line systems

Pros: Global accesses, platform independent.
      Secured communication and transactions.
      Have a better GUI interface due to JavaScript features.

Cons: Limited support for graphics and window applications
      Unstructured server programs with low concurrence

Client:  Web Browser + HTML + Java Applets
Server:  Web Server + CGI (Gateway Programs) + Database Server
Plus  secured transaction based on secured protocol & firewalls
Features:  Supporting complex user interactions and requests
           HTML/JAVA client, CGI-based Server program
           Secured communications and transactions

Applications:  Web-base application systems, tools, group-ware, On-line systems

Pros:  Global accesses, platform independent.
       Secured communication and transactions.
       Supporting Complex GUI interface.
Cons:  Slow speed on Java applet download.
       Unstructured server programs with low concurrency
Architecture Design For Web-Based Application Systems

- (1996-1997 --> Future) Object Web:
  (a) Java Web: Java Applets + Mobile Components (RMI)

Features: Supporting complex user interactions and requests
RMI communications between clients and server
No standard communication protocols. Java Client and Server.

Applications: Intranet Web-base application systems, tools, group-ware.

Pros:
Intranet accesses, platform independent.
Support multithreading server.
Supporting Complex GUI interface.

Cons:
No secured communications.
Client and server must be written in Java.
Limitation on providing scalable servers.
Architecture Design For Web-based Application Systems

- (1996-1997 --> Future) Object Web:
  (b) JDBC-based: Java Applets + JDBC gateway programs

Features: Supporting complex data-centered user interactions and requests
RMI communications between clients and server
Java Client and Server.

Applications: Intranet Web-base application systems, tools, group-ware.

Pros: Intranet accesses, platform independent.
Support multithreading server.
Supporting Complex GUI interface.

Cons: No secured communications.
Client and server must be written in Java.
Limitation on providing scalable servers.
Architecture Design For Web-based Application Systems

(1996-1997 --> Future) Object Web:
(c) Java and CORBA (such as JOE, Iona):

- CORBA avoids the CGI bottleneck
- CORBA provides a scalable server-to-server infrastructure
- CORBA extends Java with a distributed object infrastructure
Architecture Design For Web-based Application Systems

- (1996-1997 --> Future) Object Web:
  (d) The Microsoft Object Web
Architecture Design For Web-based Application Systems

- (1996-1997 --> Future) Object Web:
  (e) The CORBA/Cyberdog Object Web
Bridging Legacy Systems to the Web:

An encapsulation programs puts a more friendly face on a legacy system

An encapsulation can present an interface different from the legacy system
Architecture Design For Web-based Application Systems

Bridging Legacy Systems to the Web:

A Java e-mail System

Translate a legacy interface into something newer clients can deal with
Architecture Design For Web-based Application Systems

Bridging Legacy Systems to the Web:

Native methods to access a legacy system

CORBA expands the accessibility of an encapsulation system
Architecture Design For Web-based Application Systems

Bridging Legacy Systems to the Web:

Legacy terminals and new workstations access the new system

An encapsulation can combine several systems together
Bridging Legacy Systems to the Web:

Web access to legacy systems involves a servlet

Web access to legacy system involves a servlet and an encapsulation (servlet and encapsulation reside in the same machine)
Client Architecture in Web-based Application Systems

  - Hypertext client --> based on HTML pages + hypertext links
  - Simple interactions between users and systems
  - Stateless client-server Communications
  - No cash on the client side
  - No data verification and validation on the client side

Structure:  a) Linear Structure,  b) Tree Structure,  c) Index Structure
  d) Network Structure,  e) Simple Window Menu Structure

(a) (b) (c) (d) (e)
Client Architecture in Web-based Application Systems

1995-1996: Thin Client Layer based on HTML + JavaScript
- HTML forms + JavaScript + hypertext links
- Simple interactions between users and systems
- Stateless client-server Communications
- Small data cash on the client side (cookies)
- Simple data verification and validation on the client side
- Dynamic generation of HTML pages, tables, links

Structure: a) Linear Structure, b) Tree Structure, c) Index Structure
d) Network Structure, e) Simple Window Menu Structure
Client Architecture in Web-based Application Systems

1996-1997: Java Client (Thin - Thick)
- HTML forms embedded Java Applets
- Cash data on the client side
- Stateless/State client-server Communications
- Completed data verification and validation on the client side
- Support complex GUI structure and graphic display

Structure:  
a) Single Applet
b) Multiple Applets in a HTML page (with multiple frames)
c) Multiple Applets in different HTML pages
Client Architecture in Web-based Application Systems

Future: The Future Web Client (Three Client Models):

- Browser as Desktop:
  assumes that people live within their browsers. This is the current Netscape model of the world.

- Web-Enabled Desktop Components:
  everything on the desktop is Web-enabled; the idea is that you will be able to access the Web from within any application or component without starting a browser. This is the Cyberdog model of the world; Microsoft will also support this model in Windows 97.

- Shippable Places:
  this lets you access the Web from within your places. A place can have multiple concurrent sessions with Web object servers. In addition, multiple places can be concurrently active on the same desktop.

The Object Web may end up supporting all three models.
Server Architecture in Web-based Application Systems

- Trade-off in building a web-based application server
  - Considering factors:
    - System performance
    - Fault tolerance
    - System scale-up
    - Complexity
    - System configuration and flexibility
  - Technology selection:
    - CORBA-based server or DCOM-based server
    - Java server
    - Server made by C++/C
  - Interaction style between client and server
    - Distributed objects
    - Communication protocol, like HTTP/HTTPS, or TCP/IP socket
  - Single thread vs. multiple thread
  - Multiple servers vs. single server
  - Server configuration structure: single box vs. multiple boxes
  - Layered architecture

- How to link a web-based application server to legacy software
- How to link a web-based application server to applications
- How to link a web-based application server to third-party software
Server Architecture in Web-based Application Systems

- Trade-off in building a web-based application server
  - Considering factors:
    - System performance
    - Fault tolerance
    - System scale-up
    - Complexity
    - System configuration
  - Technology selection:
    - CORBA-based server or DCOM-based server
    - Java server
    - Server made by C++/C
  - Interaction style between client and server
    - Distributed objects
    - Communication protocol, like HTTP/HTTPS, or TCP/IP socket
  - Single thread vs. multiple thread
  - Multiple servers vs. single server
  - Server configuration structure: single box vs. multiple boxes
  - Layered architecture

- How to link a web-based application server to legacy software
- How to link a web-based application server to applications
- How to link a web-based application server to third-party software