

Architecture Implementation: Concepts and Example

- Implementation concepts
- Simple client/side applications
- Web-tier applications using CGI, Servlets, JSPs
- OCSI application implementation concepts
- OCSI applications with Web-Java-CORBA
- EJBs - using the J2EE framework
- XML Web and SOAP - Using Dot Net
- Transaction processing

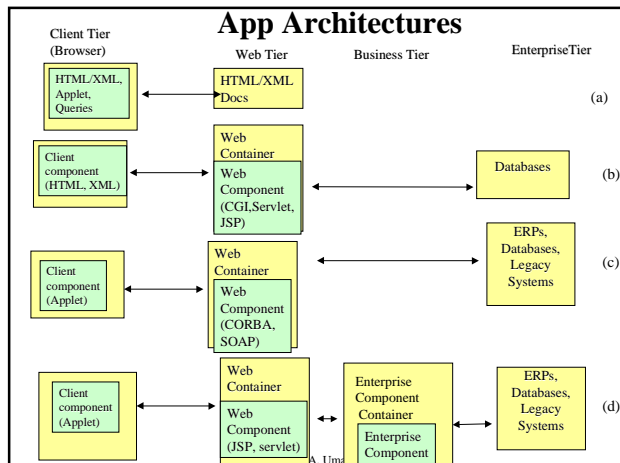
Amjad Umar

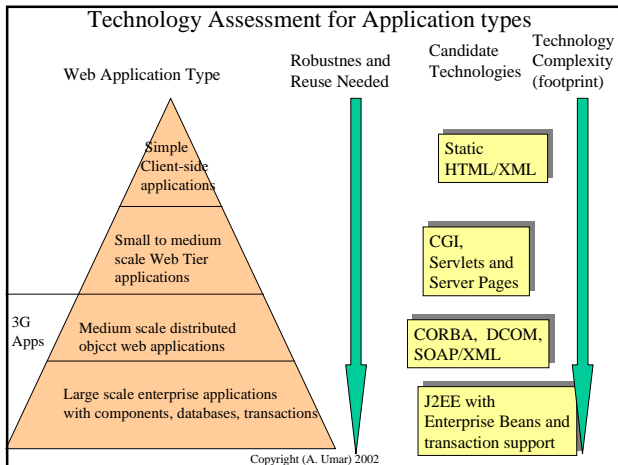
Implementation concepts

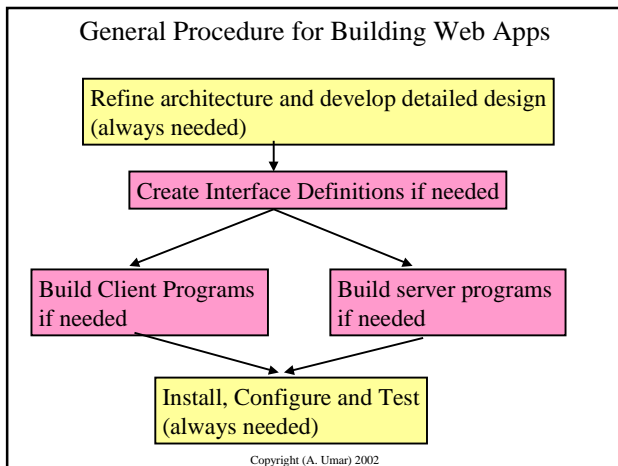
Implementation depends on type of apps

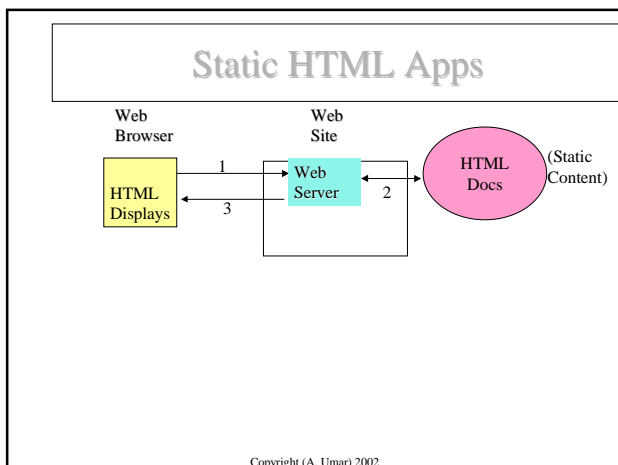
- Simple web client-side applications** that use static HTML/XML for web advertising or use Java applets that access a remote database through a JDBC interface.
- Small to medium web-tier apps** (e.g., retrieve a customer record based on a web user query). These may use CGI, JSP, ASP, or a servlet.
- Distributed object apps** that model the application as a set of reusable objects that can be invoked from Web or non-web users. This type of application can be developed by using technologies such as CORBA, DCOM, SOAP/XML.
- Component-based enterprise wide applications** that imbed business logic as components. These applications require strong transactional support. These apps should be developed by using extensive platforms such as Sun J2EE.

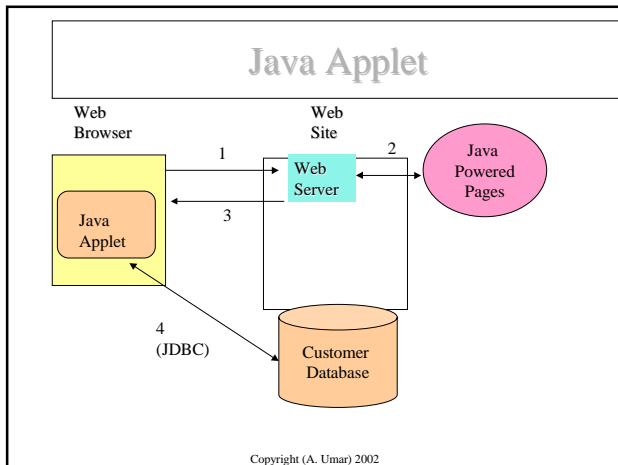
Copyright (A. Umar) 2002

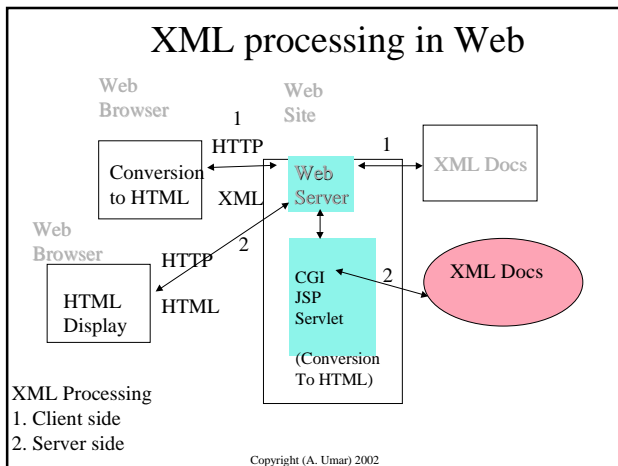












XML Processing at Client through XSL

File: test.xml

```

<?xml-stylesheet type="text/xsl" href="transform.xml"
<!-- test -->
<document>
  <body>
    Hello, world
    <warning> it is strange out there </warning>
  </body>
</document>

```

File transform.xml

```

<xsl: template match="warning"
<xsl: margin, color, etc > </xsl>
</xsl>

<xsl>
<rule>
<target-element type=customer/>
<HTML>
  <BODY>
    <children/>
  </BODY>
</HTML>
</rule>
</xsl>
<xsl>

<xsl: template match="/"
<xsl: for-each body="hello" >
<xsl: value-of ="howde">
</xsl: for-each>
</xsl>

```

How to run example:

```

c:\>xt test.xml transform.xml test.html
xt = xsl engine (www.jclark.com/xml/xt.html)
test.xml = source
transform.xml = xsl
test.html = destination (default = screen displays)

```

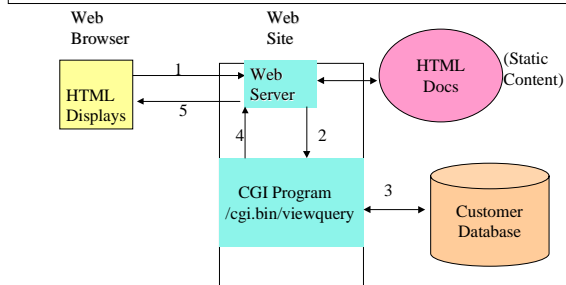
Copyright (A. Umar) 2002

Web Tier Apps

- These applications use the web tier (web server site) heavily for business processing
- Typically known as server side applications
- Technologies used for simple to medium sized apps:
 - CGI
 - servlets
 - server pages (JSP, ASP, XSP)
- Technologies used for complex apps:
 - CORBA
 - EJBs

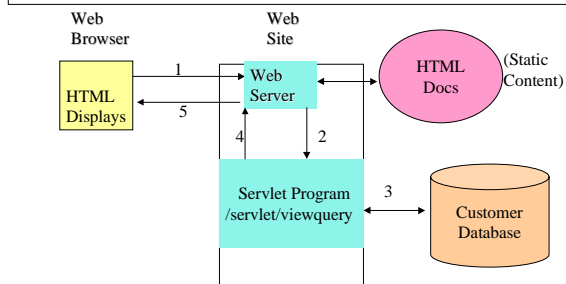
Copyright (A. Umar) 2002

CGI

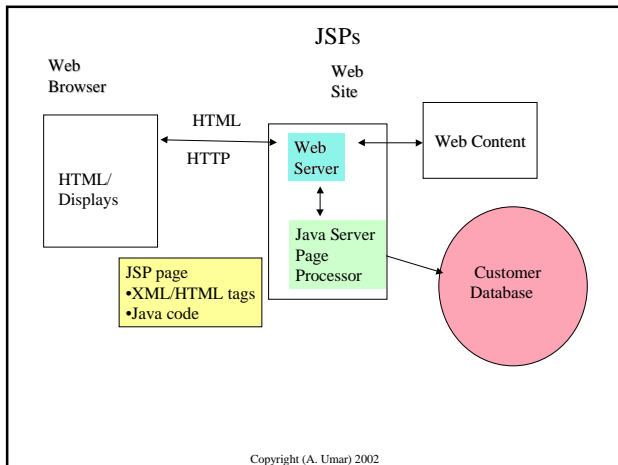


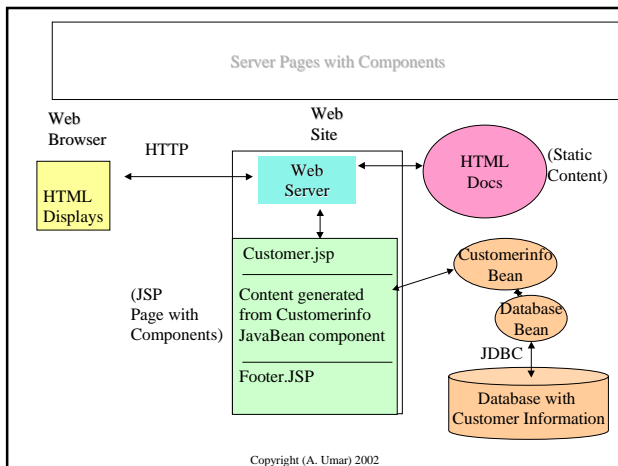
Copyright (A. Umar) 2002

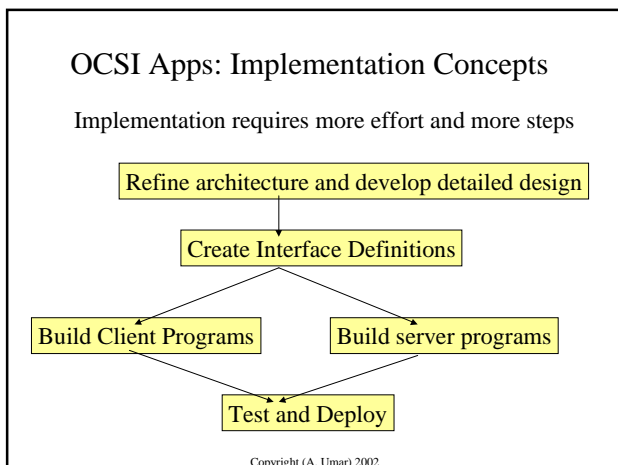
Servlets

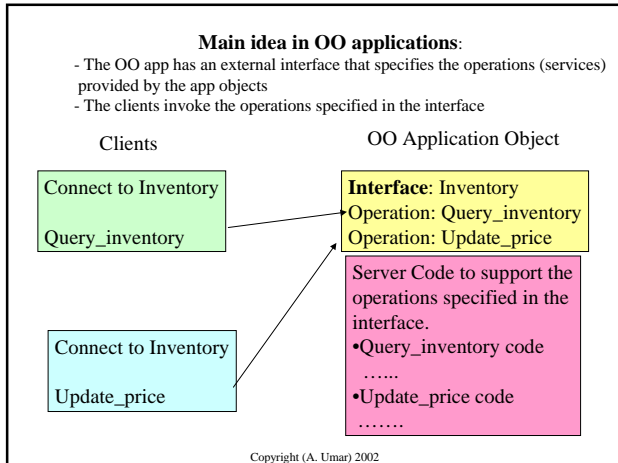


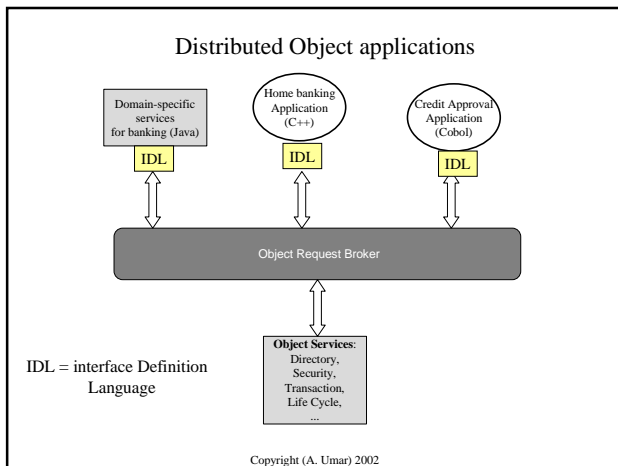
Copyright (A. Umar) 2002

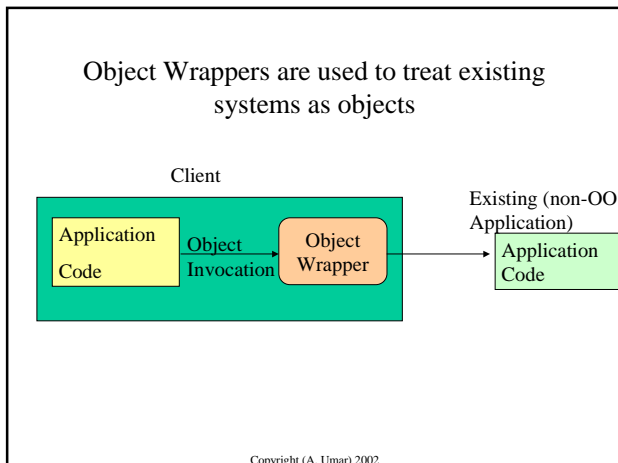


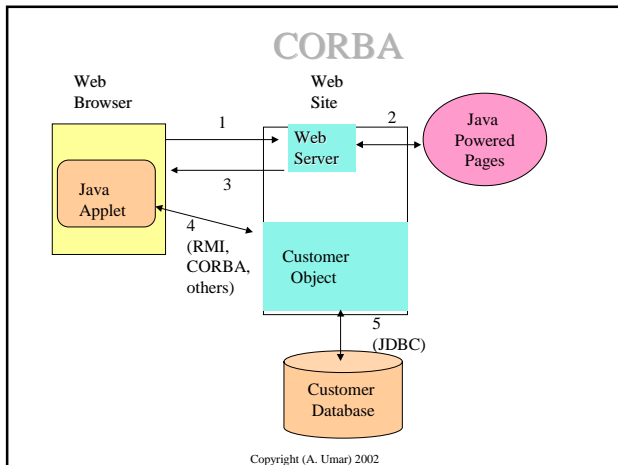


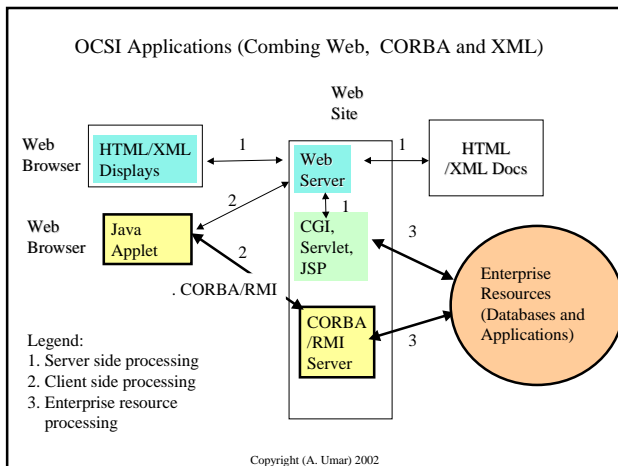


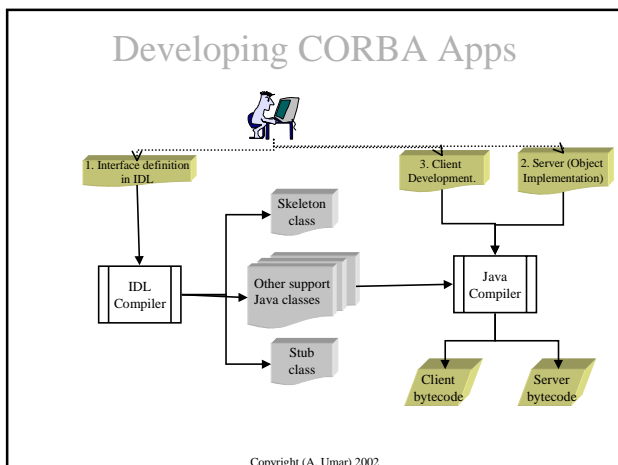










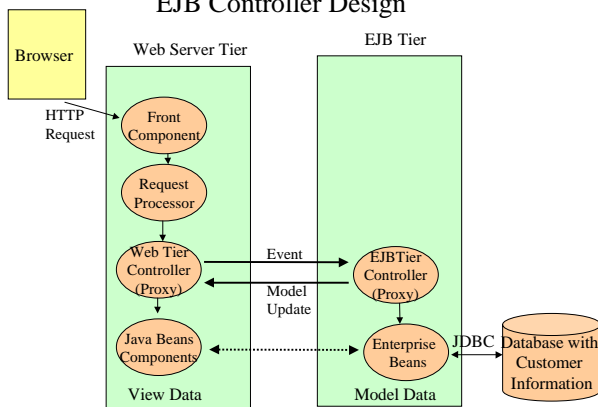


Enterprise Java Beans (EJBs)

- **Enterprise Java Beans (EJBs)** are perhaps the best known examples of enterprise components.
- EJBs come in the following flavors:
 - **Session Beans.** A session bean represents a transient conversation with a client. When the client finishes executing, the session bean and its data disappear.
 - **Entity Bean.** An entity bean represents persistent data stored in a database. If the client terminates or if the server shuts down, the bean "plumbing" services ensure the entity bean data is saved.
 - **Message Driven Bean.** A message-driven bean allows a business component to receive JMS (Java Message Services) messages asynchronously. This is a new bean to support message queuing and publish/subscribe.

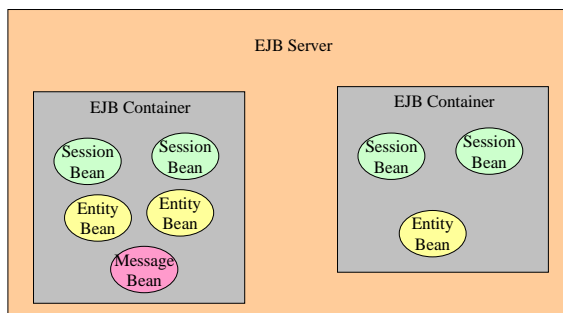
Copyright (A. Umar) 2002

EJB Controller Design



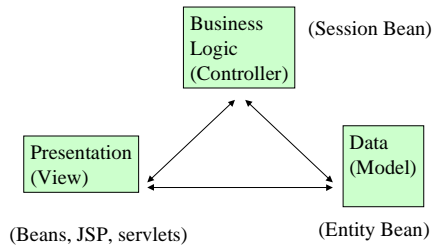
Copyright (A. Umar) 2002

EJB Environment



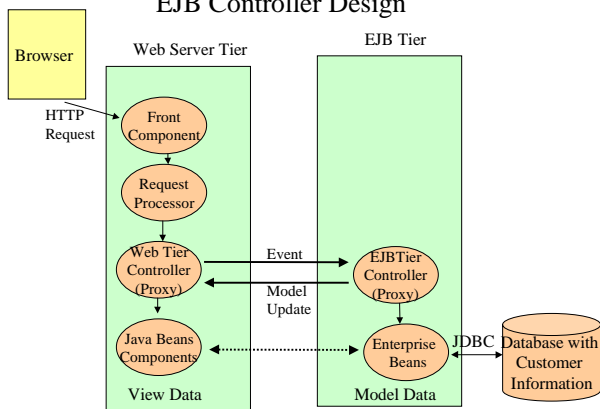
Copyright (A. Umar) 2002

Model-View-Controller Design for EJBs



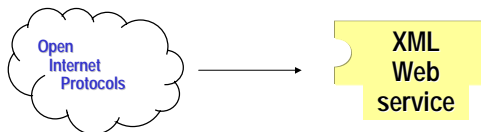
Copyright (A. Umar) 2002

EJB Controller Design



Copyright (A. Umar) 2002

XML Web Services and SOAP



A programmable application component accessible via standard Web protocols

- Provide a directory of services on the Internet
- XML Web services are defined in terms of the formats and ordering of messages
- XML Web service consumers can send and receive messages using XML
- Built using open Internet protocols

Copyright (A. Umar) 2002

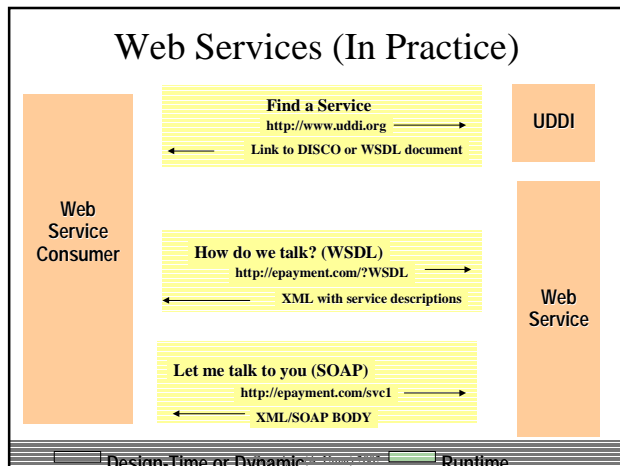


Web XML and SOAP Implementation

- SOAP (Simple Object Access Protocol)
 - Explicit serialization (HTTP + XML description) protocol used in service exchanges
- WSDL (Web Service Description Language)
 - XML document describing the location and interfaces a particular service supports – the client's contract
- DISCO (Discovery)
 - XML document describing (URI) of service
- UDDI (Universal Description Discovery and Integration)
 - Yellow pages directory for services

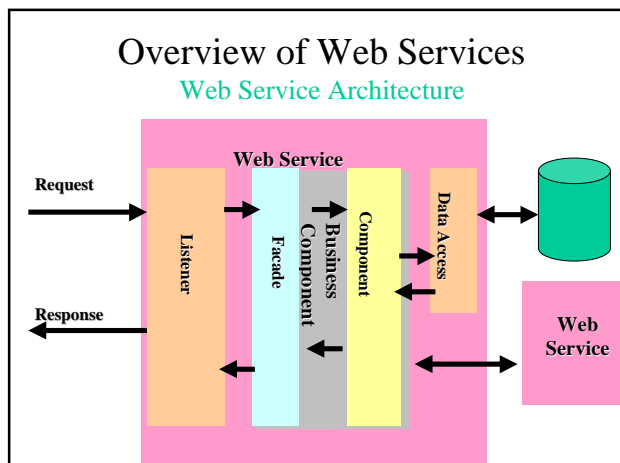
Copyright (A. Umar) 2002

Web Services (In Practice)



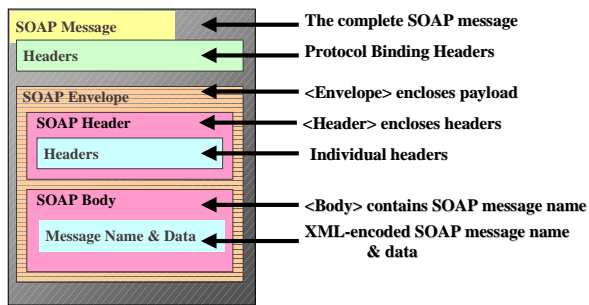
Overview of Web Services

Web Service Architecture



SOAP

Message Structure



Copyright (A. Umar) 2002

Key Points

Implementation depends on type of apps:

- Simple client/side applications
- Web-tier applications using CGI, Servlets, JSPs
- OCSE application implementation concepts
- OCSE applications with Web-Java-CORBA
- EJBs - using the J2EE framework
- XML Web and SOAP - Using Dot Net
- Transaction processing

Copyright (A. Umar) 2002
