## Web versus Traditional Application

<table>
<thead>
<tr>
<th>Application Characteristic</th>
<th>Web Environment</th>
<th>Traditional Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Clients per Application</td>
<td>Millions</td>
<td>&lt; 100</td>
</tr>
<tr>
<td>Number of Servers per Application</td>
<td>&gt; 1,000,000</td>
<td>1-10</td>
</tr>
<tr>
<td>Geography</td>
<td>Global</td>
<td>Campus</td>
</tr>
<tr>
<td>Server-to-server invocations</td>
<td>Yes</td>
<td>Rarely</td>
</tr>
<tr>
<td>Middleware</td>
<td>ORBs/Active Agents, Etc</td>
<td>SQL and stored procedures</td>
</tr>
<tr>
<td>Client/Server Architecture</td>
<td>3-tier or greater</td>
<td>2-tier</td>
</tr>
<tr>
<td>Transactional updates</td>
<td>Pervasive</td>
<td>Very Infrequent</td>
</tr>
<tr>
<td>Multimedia Content</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Mobile Agents</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Pg 55 Client/Server Programming with Java and CORBA with updates.
Definitions

- CORBA – Common Object Request Broker Architecture
- IIOP – Internet Inter-ORB Protocol
- ORB – Object Request Broker
- IDL – Interface Definition Language
“We expect that over the next few years IIOP will become as ubiquitous as HTTP and CGI, IIOP provides a comprehensive system through which objects can request services from one another across the wide variety of platforms or database systems they’re built on. Just as Web technology has helped companies simplify and centralize the distribution of information, distributed objects will help them simplify and centralize their enterprise applications… Now that we have standard ways to build networks and run services on them, we have an opportunity that never existed before - to build network applications. Let’s take advantage of it.”

Marc Andreessen, Netscape Co-founder
October 1996
Who Controls CORBA?

• Object Management Group (OMG)
• Over 700 member companies
  – including
    Sun Microsystems  CMSTAT
    Visigenics      Matrix One
    IBM            Iona
    Sherpa         Metaphase
Migration to CORBA

FTP
User was required to know where the information was located

Gopher
User was able to do string searches for information but there was no linking of information

Web Pages
Adds links between pages so information can be related, still no context tags to aid in information retrieval

XML Pages
Adds context tags to enable better information retrieval, still limited control over data methods

CORBA Objects
Full object definitions, including attributes and methods
Why use CORBA?

- Improved Performance over HTTP & CGIs
- Persistent Connections
- Inter-Operability between ORB Vendors
- Good for Inter-Language Communications
- Excellent Interface to Legacy Systems
- Can be Delivered using existing Web Solutions.
## CORBA IIOP versus HTTP-CGI

<table>
<thead>
<tr>
<th>Feature</th>
<th>CORBA IIOP</th>
<th>HTTP-CGI</th>
</tr>
</thead>
<tbody>
<tr>
<td>State preservation across invocations</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>IDL and interface repository</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Metadata support</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Dynamic Invocations</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Transactions</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Security</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Rich object services</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Callbacks</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Server/server infrastructure</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Server scalability</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>IDL-defined methods</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Pg 35 Instant CORBA
Interoperability

- By using CORBA version 2, it is possible to interconnect ORBs from different vendors.
- Demonstrated by Software Engineering Institute (SEI) of Carnegie Mellon University (CMU)
- SEI also demonstrated that ORBs can be connected to Microsoft DCOM
# Comparison of CORBA ORBs

<table>
<thead>
<tr>
<th>CORBA Feature</th>
<th>ObrixWeb</th>
<th>Joe</th>
<th>Visibroker for Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client-side Java</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Static method invocations</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dynamic method invocations</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Interface Repository</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Server Callbacks</td>
<td>Yes (Ver 2)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Native Java over IIOP</td>
<td>No</td>
<td>No</td>
<td>Yes (Caffeine)</td>
</tr>
<tr>
<td>Server-side Java</td>
<td>Yer (Ver 2)</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Wide market support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Circa Late 1996

Pg 50 Instant CORBA
At the Heart of CORBA

- CORBA deals with Objects.
- All Interactions take place via the ORB or “Object Request Broker”
- All Interfaces to an Object are defined using IDL or “Interface Definition Language”.
The ORB and the IDL
Legacy System Interfacing

WWW (HTML/XML) -> Web Server

Client Stubs <-> CORBA ORB

Server Skeleton <-> Legacy Data Source
Using the Web with CORBA

HTML Request via HTTP → HTTP Server → CORBA ORB → CORBA Business Object → Legacy Database
Using the Web with CORBA

The Internet

HTTP Server

CORBA ORB

CORBA Business Object

Legacy Database

HTML Request via HTTP

Applet Returned via HTTP
Using the Web with CORBA

The Internet

HTTP Server

Applet Returned via HTTP

HTML Request via HTTP

Locally Running Applet

CORBA ORB

CORBA Business Object

Legacy Database
Using the Web with CORBA

The Internet

Locally Running Applet

HTML Request via HTTP

Applet Returned via HTTP

HTTP Server

IIOP Session

CORBA ORB

CORBA Business Object

Legacy Database
The IDL

- Language Neutral Description of the Object to be exchanged
- Contains definitions for the Attributes to be exchanged and what Methods can be applied to those attributes
The ORB and the IDL
From IDL to Interface

Create IDL Definition

IDL to Language Compilers

Server Skeletons

Server Application Software

Server Skeletons

Server Application Software

Server Skeletons

Server Application Software

Client Stubs

Client Application Software

Client Stubs

Client Application Software

Client Stubs

Client Application Software
Module MyAnimals
{
    /* Class Definition of Dog */
    interface Dog:Pet, Animal
    {
        attribute integer age;
        exception NotInterested {string explanation};

        void Bark(in short how_long)
            raises (NotInterested);

        void Sit(in string where)
            raises (NotInterested);

        void Growl(in string at_whom)
            raises (NotInterested);
    }
} /* End of MyAnimals */
Module MyAnimals
{
   /* Class Definition of Dog */
   interface Dog:Pet, Animal
   {
      attribute integer age;
      exception NotInterested {string explanation};

      void Bark(in short how_long)
      raises (NotInterested);

      void Sit(in string where)
      raises (NotInterested);

      void Growl(in string at_whom)
      raises (NotInterested);
   }
} /* End of MyAnimals */
Module MyAnimals
{
   /* Class Definition of Dog */
   interface Dog: Pet, Animal
   {
      attribute integer age;
      exception NotInterested { string explanation; }

      void Bark(in short how_long)
      raises (NotInterested);

      void Sit(in string where)
      raises (NotInterested);

      void Growl(in string at_whom)
      raises (NotInterested);
   }
} /* End of MyAnimals */
Module MyAnimals
{
   /* Class Definition of Dog */
   interface Dog:Pet, Animal
   {
      attribute integer age;
      exception NotInterested {string explanation};

      void Bark(in short how_long)
         raises (NotInterested);

      void Sit(in string where)
         raises (NotInterested);

      void Growl(in string at_whom)
         raises (NotInterested);
   }
} /* End of MyAnimals */
Module MyAnimals
{
    /* Class Definition of Dog */
    interface Dog:Pet, Animal
    {
        attribute integer age;
        exception NotInterested {string explanation};

        void Bark(in short how_long)                    DEFINES A METHOD
            raises (NotInterested);

        void Sit(in string where)                      DEFINES A METHOD
            raises (NotInterested);

        void Growl(in string at_whom)                  DEFINES A METHOD
            raises (NotInterested);
    }
} /* End of MyAnimals */
CORBA Services

Object Request Broker

- Naming
- Persistence
- Externalization
- Collections
- Relationships
- Events
- Security
- Life Cycle
- Time
- Licensing
- Query
- Properties
- Concurrency
- Trader
- Transactions
Where to go for more information

- OMG Website www.omg.org
- Instant CORBA: Orfali, Harkey & Edwards
- Client/Server programming with JAVA and CORBA: Orfali & Harkey
- CORBA Design Patterns:
Are there any Questions or Comments

Thank you for your attention
Break Time