CPS616 Computational Science for Information Applications

aka

Technologies for an Information Age

Introductory Material

Instructor: Nancy McCracken

using material from Geoffrey Fox, Wojtek Furmanski

and many talented NPAC students

Syracuse University

NPAC

111 College Place

Syracuse NY 13244 4100

Abstract of CPS616 Spring 2000 Introduction

- **This Foilset contains introductory material on CPS616 course for spring 2000**
- **Some Aspects of Course Logistics -- all students must go to web site for complete discussion of this**
  - **http://www.npac.syr.edu/projects/cps616spring00/**
- **Overview of Field and Material covered and relation to other courses CPS606, CPS640, CPS714 and Syracuse University CIS PhD Qualifying exams**
- **Summary of Base Distributed Object Web and Relevant Technologies**

Overview of CPS Web/Information Technology Courses - I

- CPS606 Taught last semester is basic Java for distributed systems, including user interfaces, RMI, and servlets
- CPS616 is critical leading edge distributed object and web software system and application building technologies including JavaScript, Advanced Java Capabilities, Web-linked Databases, Security, Object Web.
- CPS616 contains core software technologies needed to build world wide distributed systems -- this is the key challenge today in computer science
- CPS714 is new and specialized topics in the same area as CPS616 and is set up as a mix of lectures and a project course
- CPS640 is MultiMedia and Network Systems including digital video -- it is the hardware and network technologies needed for world wide distributed systems
- CPS690 are introductory research projects.

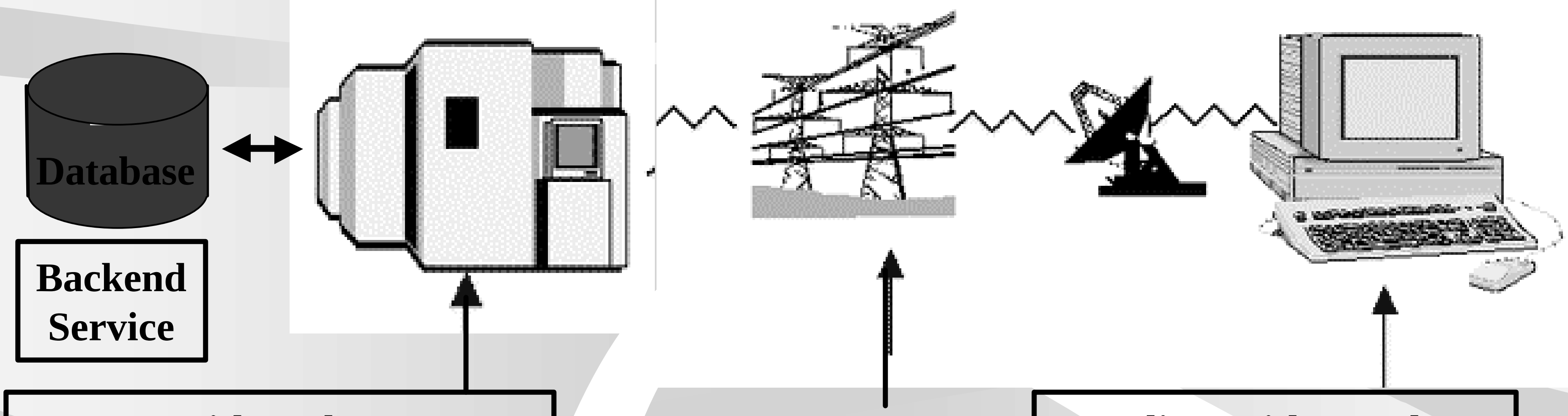
Overview of CPS Web/Information Technology Courses - II

- Courses CPS606 616 714
  <--- HTML Java Web Technologies Web Systems <--
- Material changes with time(<--) so that as new technologies added in CPS714, older and better understood ones are moved into CPS616 which itself hands technologies to CPS606!
  - Example: RMI (Java Remote Method Invocation) and servlets were taught in CPS606 last semester. Previously they were in CPS616. VRML has been de-emphasized as it appears to decline in interest
  - Security, object/component technologies (such as Javabeans) and XML were originally covered in CPS714 and moved to CPS616
  - Web Computing and Collaboration will stay in CPS714

Some Course Prerequisites

- **We will assume Basic Web Browsing and HTML expertise and Java at the level of CPS606**
  - **Permission of Instructor is needed if you have not taken CPS606**
- **You should be familiar with either PC or UNIX environment and program in at least one real language including Java**
  - **Perl is still widely used, but not taught here?**
- **We will not assume any database or CORBA knowledge and will review basic material such as SQL**
- **NPAC provides servers for you to access Oracle databases and other needed core resources**
- **You need a UNIX workstation or a PC running Windows (95,98 or NT)**

Components of a Basic Web System



Database

Backend Service

Host with Web Server and attached CGI Script in Perl or Java
Perhaps linking to a Backend service such as a database or digital video server

Host holds HTML files typically stored in a UNIX/Windows NT file system but could also be in a database such as Oracle or Microsoft Access

Network trying to provide Quality of Service and using compression to make better use of available bandwidth

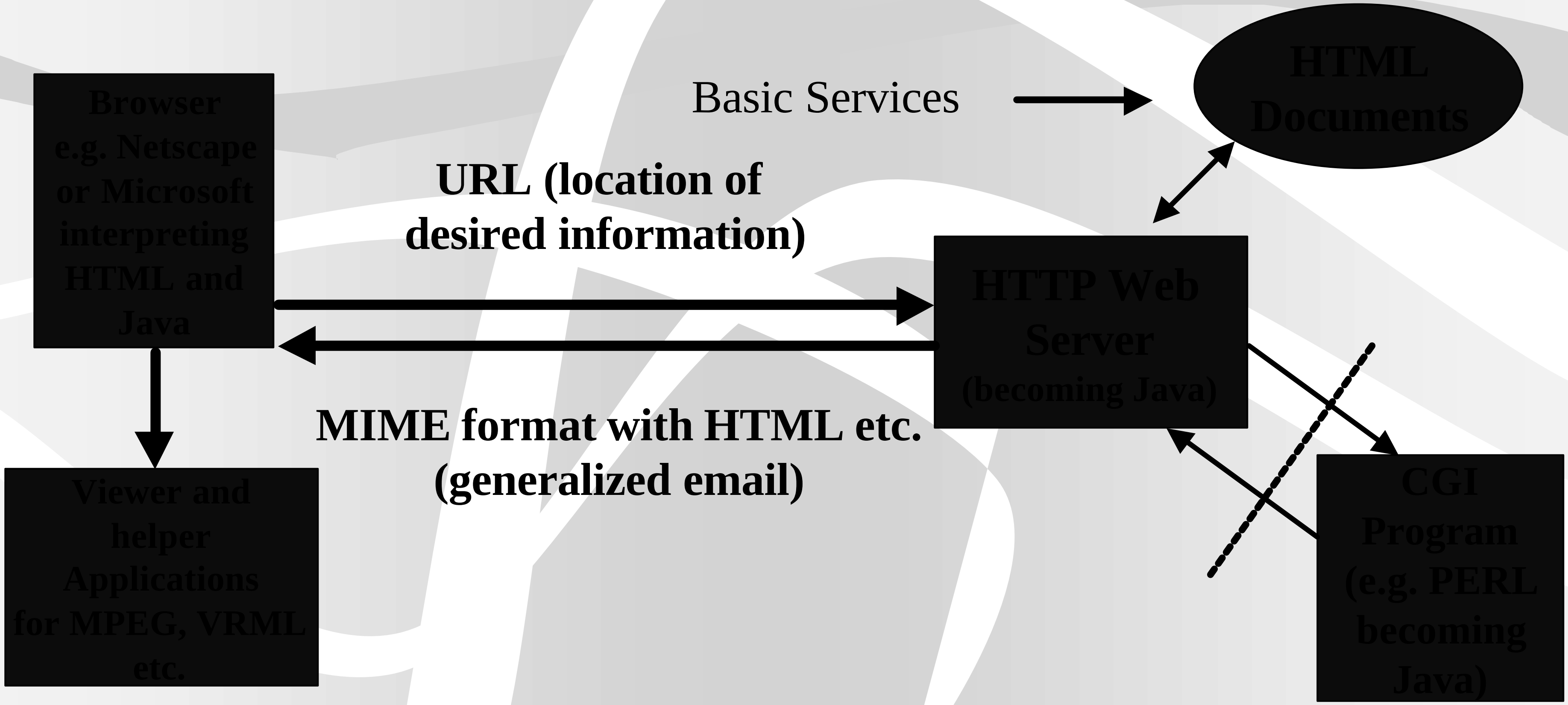
Client with a Web Browser displaying “simple” HTML Text and Images obtained from host.
Client runs JavaScript (from HTML documents) and Java Interpreters (acting on downloaded JavaVM bytecodes)
Java and JavaScript give general dynamic behavior

Where to learn What you Want!

- CPS606: HTML, Java and CGI Scripts
- CPS640: Network Services, Multimedia Systems including Server and Client Digital Video
- CPS616: Web-linked Databases (JDBC to Cold Fusion), JavaScript, Javabeans, dynamic HTML, XML, Java Web Servers, Servlets, Active Server Pages, RMI, Java IDL, CORBA, COM, ActiveX, JINI, Security, JDK1.2, and some mention of Lotus Notes, VRML 2.0, Java2D and Java3D
- CPS714: Collaborative and Computing Technologies and whatever is on leading edge

Basic (CPS606) Structure of World Wide Web

- **Browsers have SAME interface on ALL Computers**
- **CGI Programs were originally usually written in PERL but can be essentially any Process and so do simulation, database access (this is JDBC), advanced document processing etc. Java (servlets) is of growing importance in Server Code**



Basic Services

HTML Documents

Browser e.g. Netscape or Microsoft interpreting HTML and Java

URL (location of desired information)

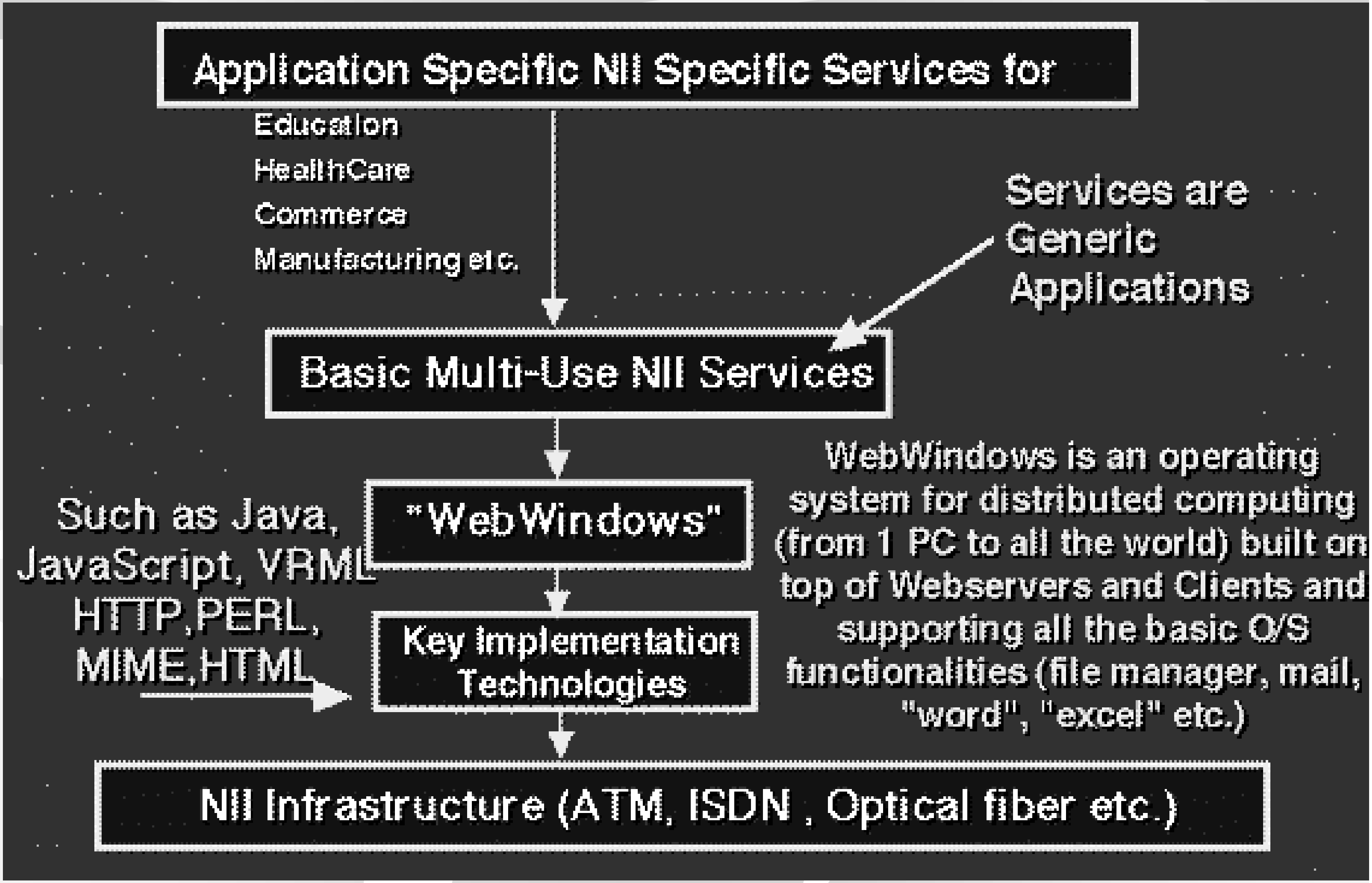
HTTP Web Server (becoming Java)

MIME format with HTML etc. (generalized email)

Viewer and helper Applications for MPEG, VRML etc.

CGI Program (e.g. PERL becoming Java)

Architecture of Web Software



Application Specific NII Specific Services for

Education

HealthCare

Commerce

Manufacturing etc.

Services are Generic Applications

Basic Multi-Use NII Services

"WebWindows"

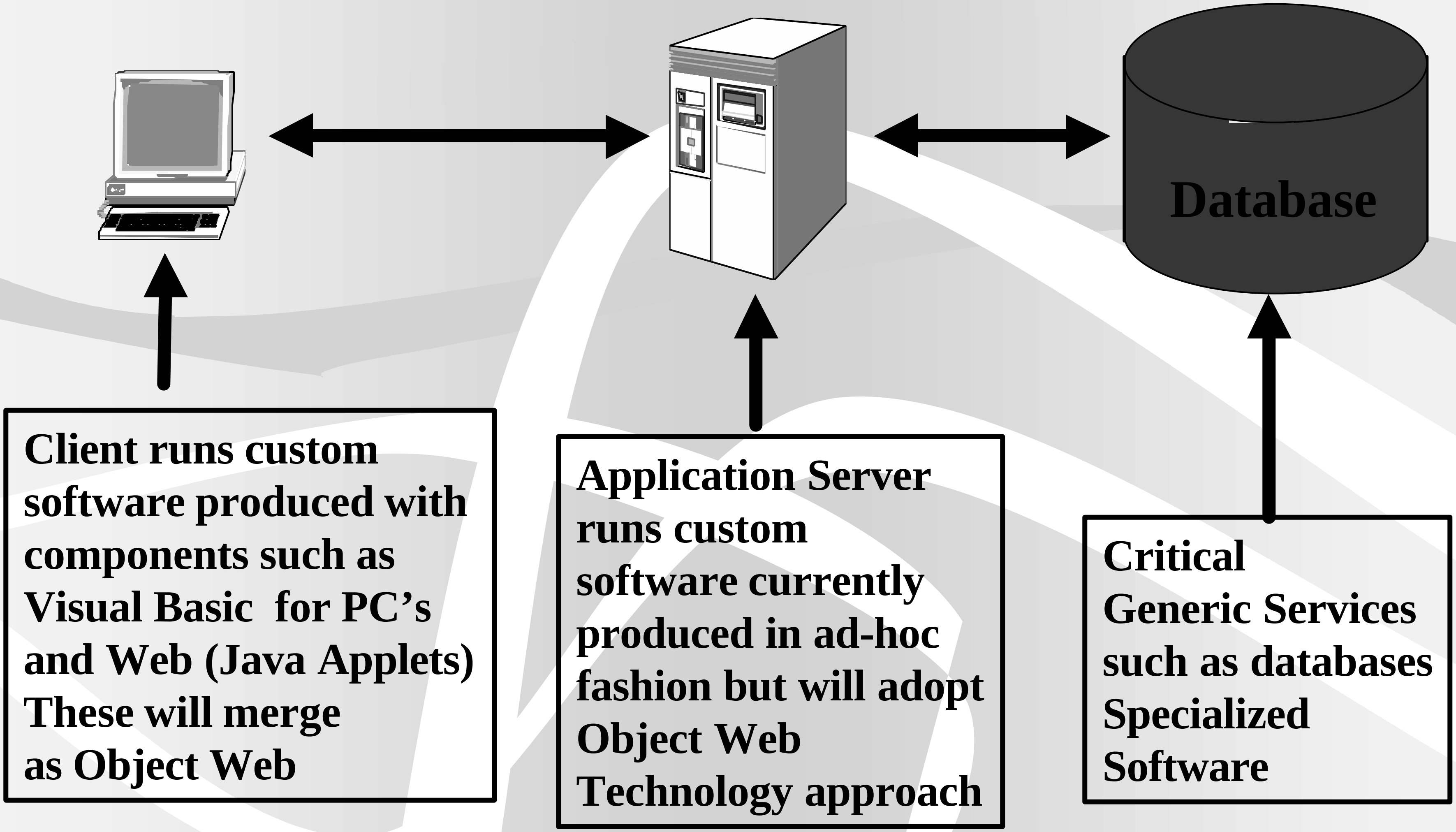
WebWindows is an operating system for distributed computing (from 1 PC to all the world) built on top of Webservers and Clients and supporting all the basic O/S functionalities (file manager, mail, "word", "excel" etc.)

Such as Java, JavaScript, VRML HTTP,PERL, MIME,HTML

Key Implementation Technologies

NII Infrastructure (ATM, ISDN , Optical fiber etc.)

The 1998 Information System Architecture



Client runs custom software produced with components such as Visual Basic for PC’s and Web (Java Applets) These will merge as Object Web

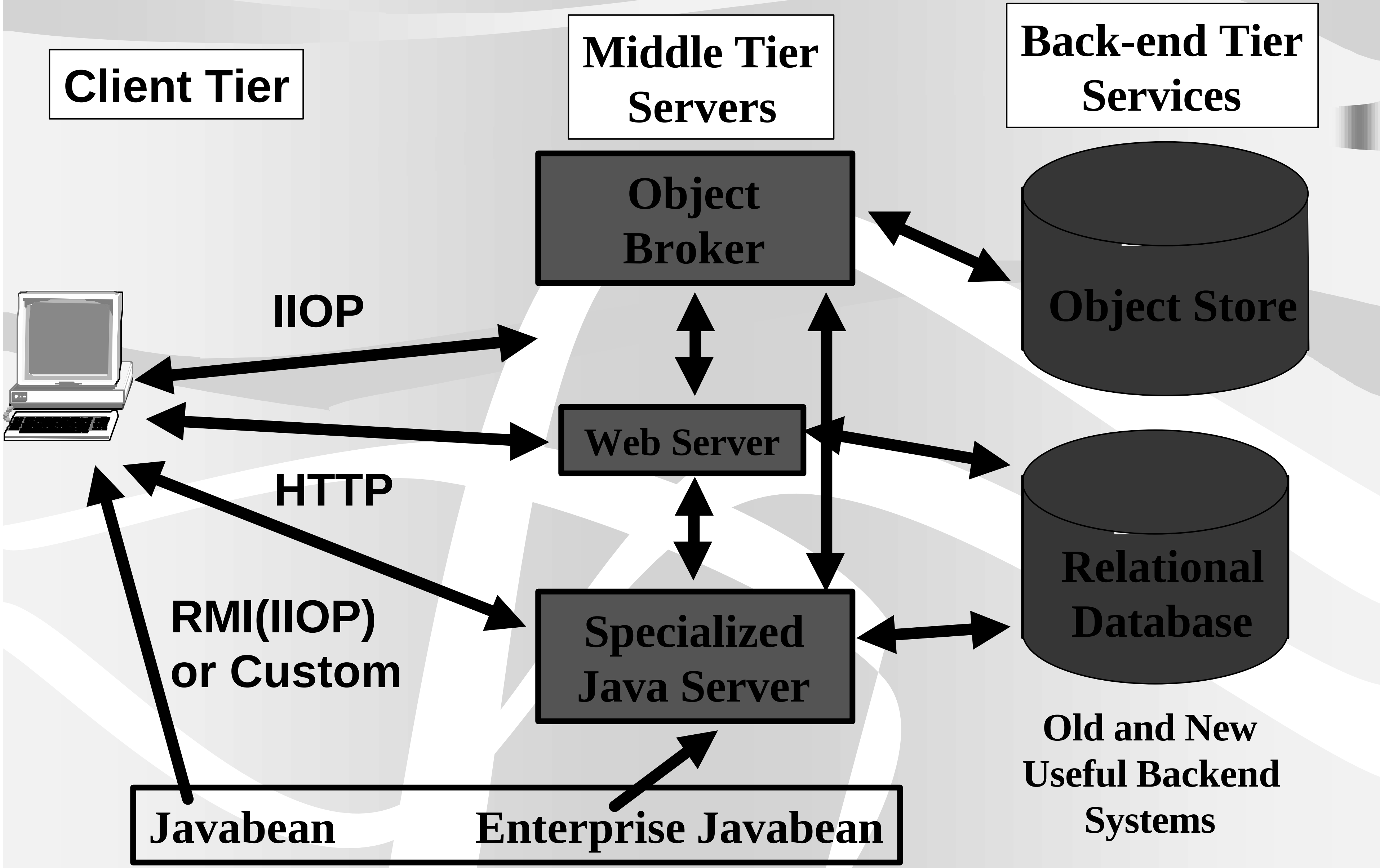
Application Server runs custom software currently produced in ad-hoc fashion but will adopt Object Web Technology approach

Critical
Generic Services such as databases
Specialized Software

Distributed Object Web Technology Model - I

- **Basic Vision: The current incoherent but highly creative Web will merge with distributed object technology in a multi-tier client-server-service architecture with Java based combined Web-ORB's**
- **Need to abstract entities (Web Pages, database entries, simulations) and services as objects with methods(interfaces)**
  - **CORBA .. XML is "just" CGI done right**
- **COM(Microsoft) and CORBA(world) are competing cross platform and language object technologies**
  - **Every Netscape4 browser has a Visigenic ORB built in**
- **Javabeans plus RMI and JINI is 100% pure Java distributed object technology**
- **W3C says you should use XML which defines a better IDL and perhaps an object model -- certainly does for documents**
- **How do we do this while technology is still changing rapidly!**

Multi-Tier Client Server Service



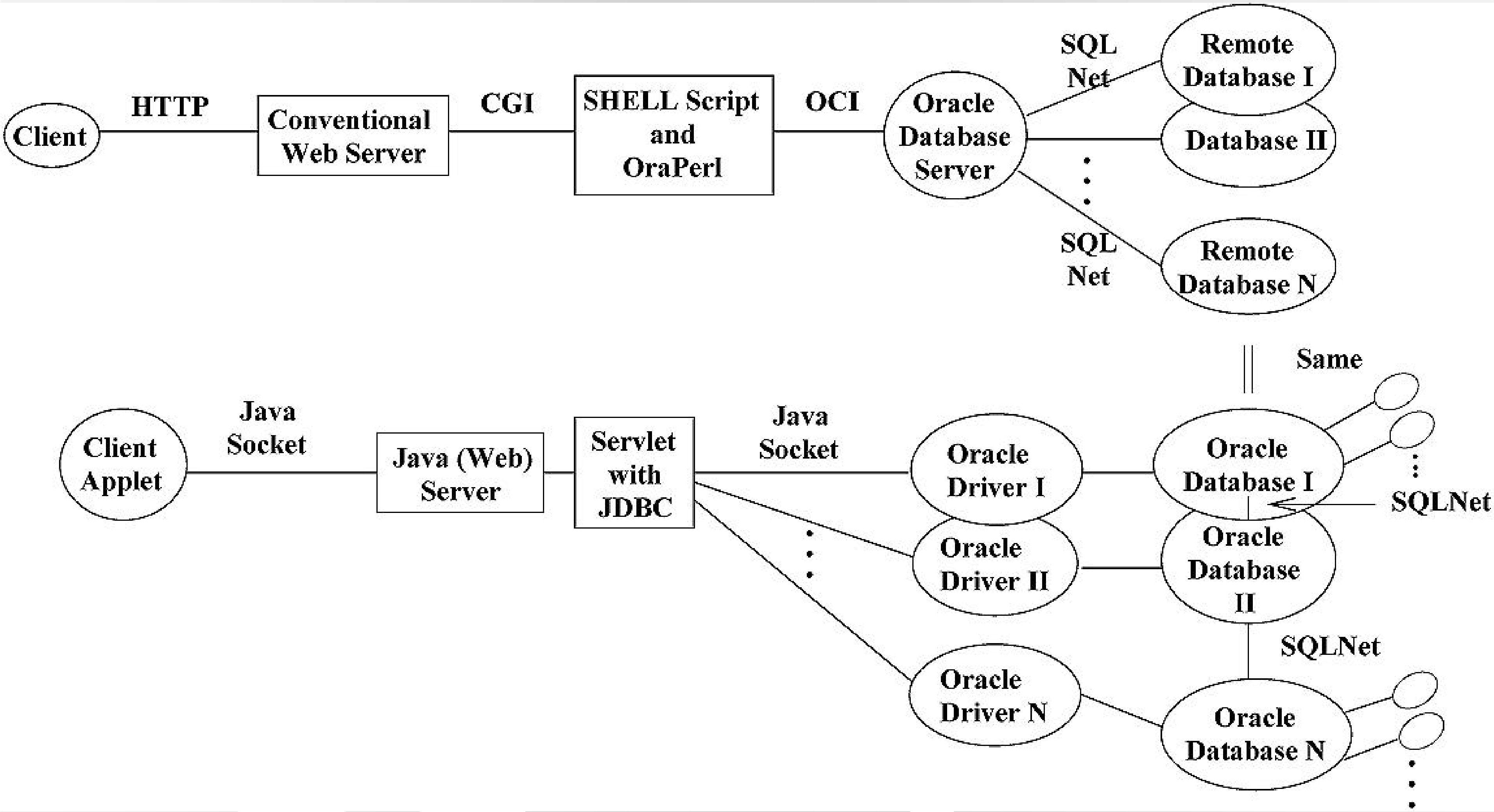
Distributed Object Web Technology Model - II

- Need to use mix of approaches -- choosing what is good and what will last
- For example develop Web-based databases with Java objects using standard JDBC (Java Database Connectivity) interfaces
  - Oracle DB2 Informix Sybase, Lotus Notes, Object database confusion becomes an issue of performance/robustness NOT functionality
- Even better use (Enterprise) Javabeans which are Java’s (middle tier) or client componentware offering visual interfaces, containers (here they are consistent with CORBA standard) and standard software engineering interfacing rules
  - e.g. Java Blend is built on top of JDBC to use enterprise Javabeans to store Java Objects in relational databases
- Use CORBA to wrap existing applications
- Note Middle tier insulates client from backend -- can use one object model for user level and different one for backend
  - specialized object databases getting “overwhelmed” by multi-tier approach with Oracle etc. traditional backends

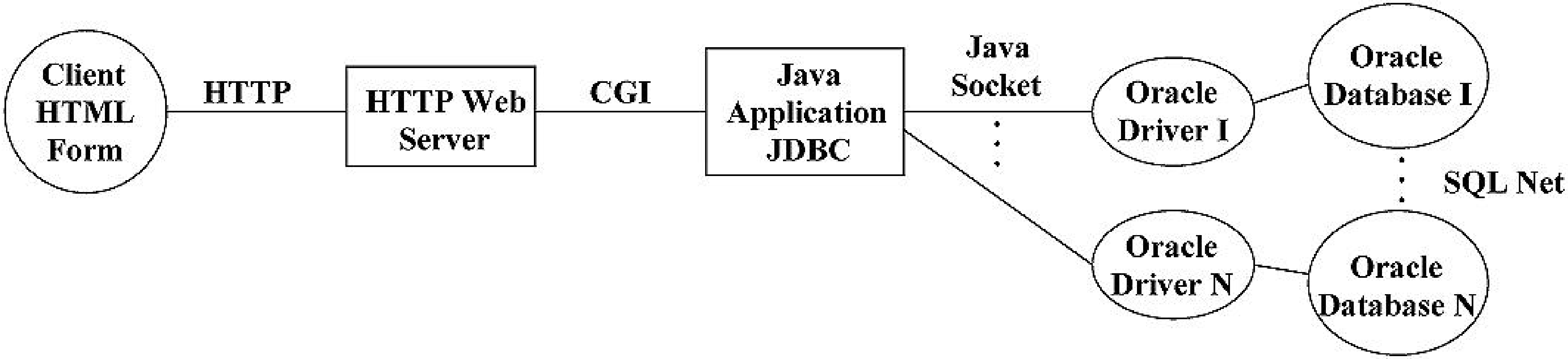
Distributed Objects

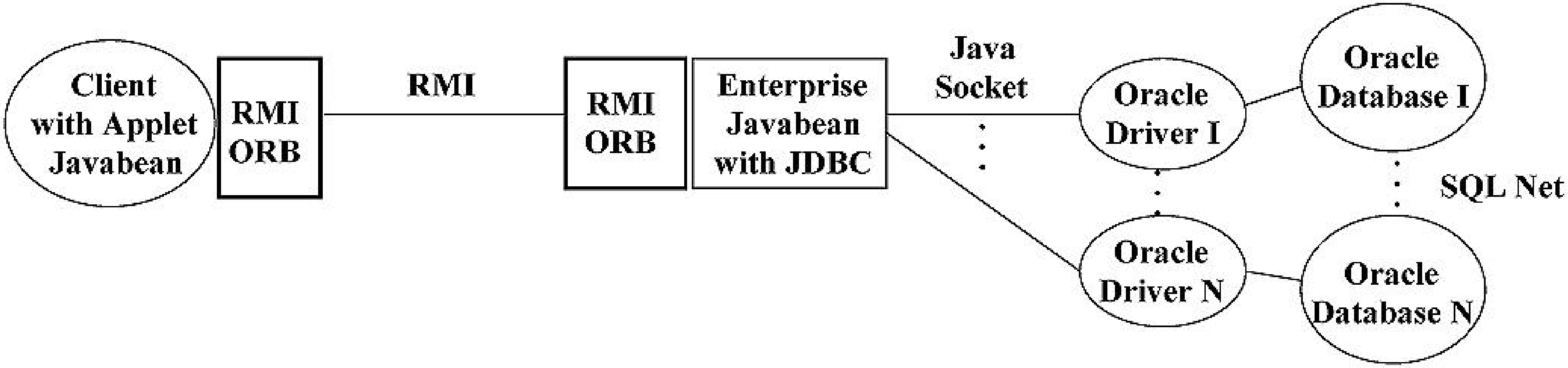
- **Examples of current object technologies**
  - **Documents -- URL**
  - **"General Programs including database invocations"**
    - **Old style Web -- CGI**
    - **New Style Web -- XML makes server side objects look like applets as far as invocation goes**
    - **CORBA and COM -- special "interface definition language" (IDL) defines invocation in C++ like syntax**
    - **RMI uses Java language as IDL language**
- **Benefits of distributed objects**
  - **allows objects written in different languages to communicate seamlessly via standardized messaging protocols embodied by middleware.**
  - **Higher levels of transparency of interoperability**
  - **Objects can be “self-managing” of resources**
  - **provides flexible grain of decomposition for building complex systems**

Two Database Web Linkages

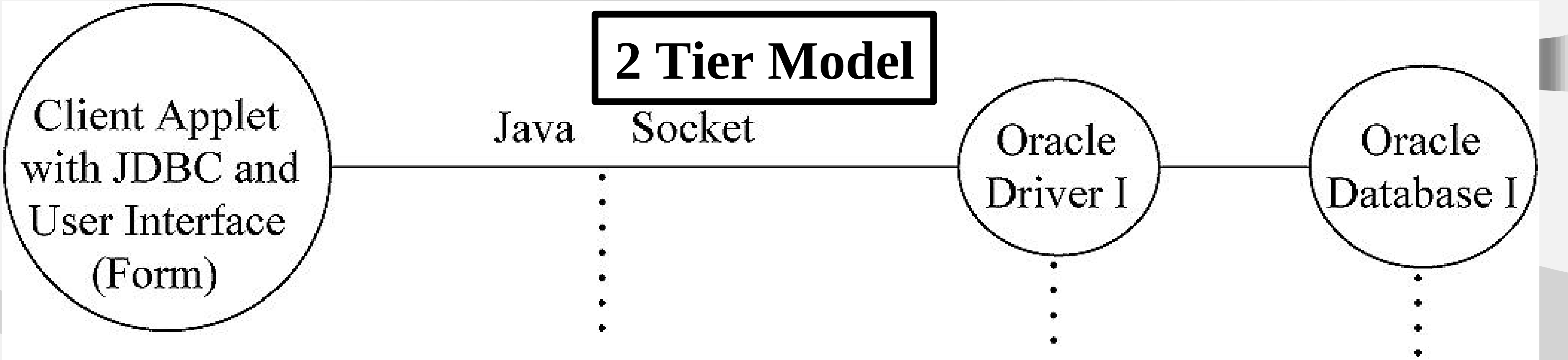


Two More 3 Tier Web Database Links

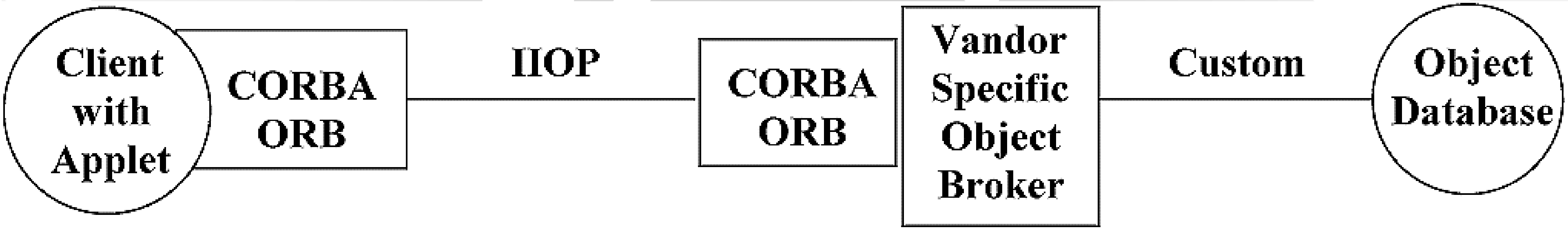




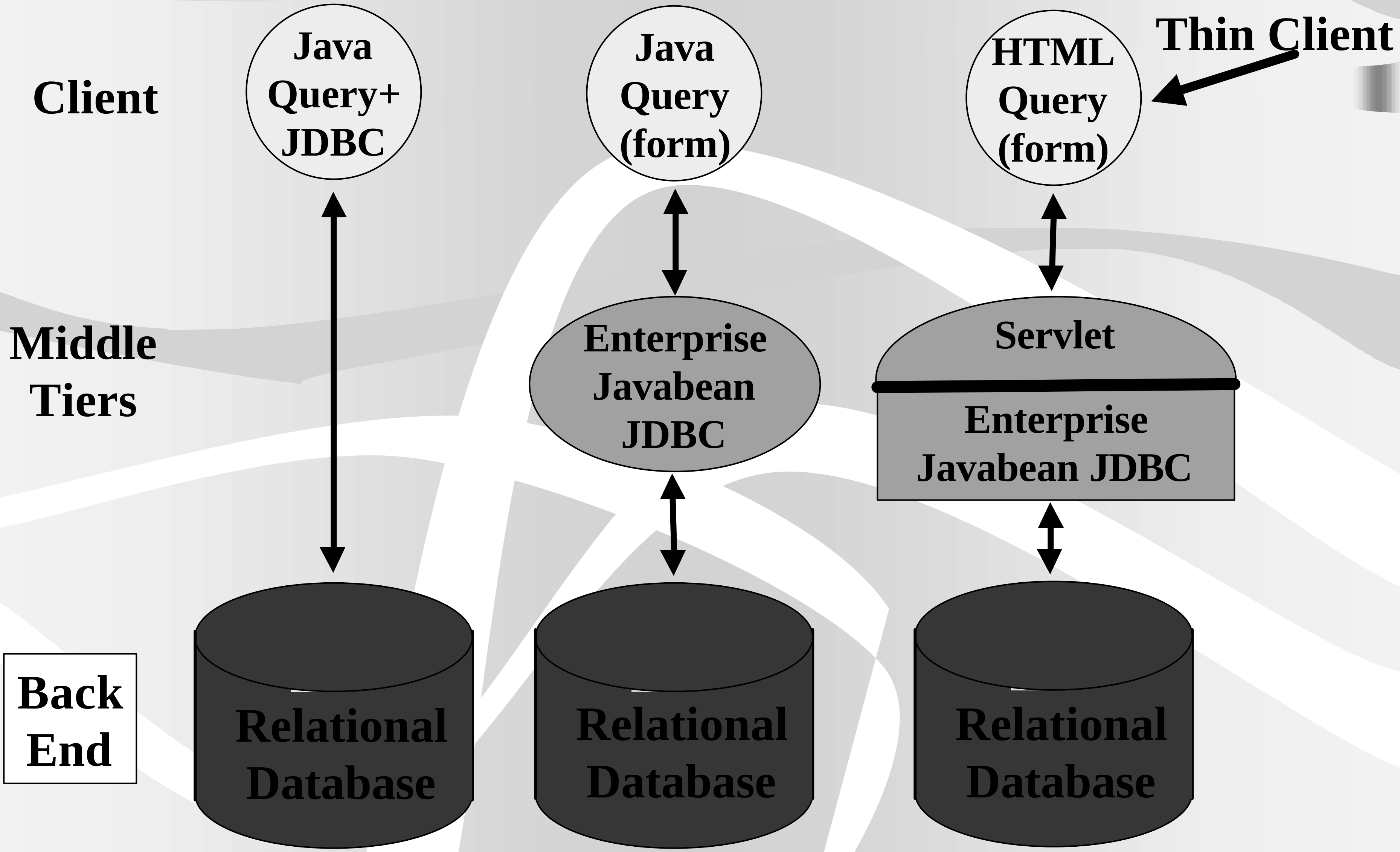
2 Tier and CORBA Models



In this model, linking to database servers can be established as many Java Socket connections. Each connection will link to a driver and a database server, but Oracle database servers can talk to each other via SQL*Net.

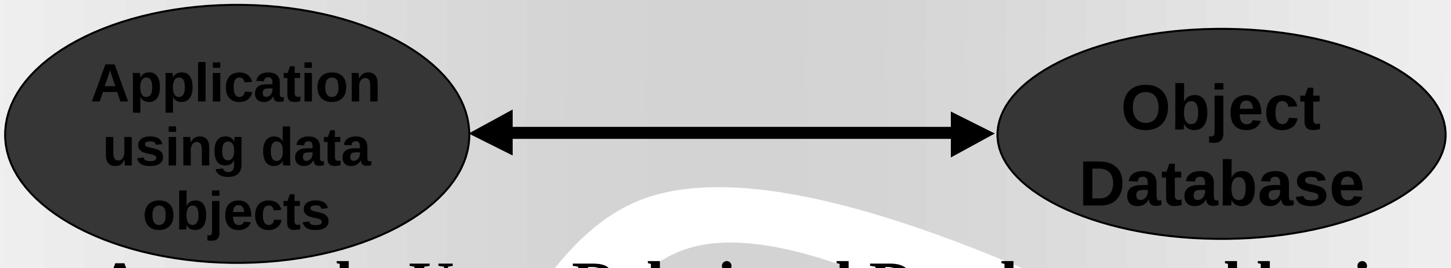


Comparison of 2 3 and 4 Tier Models



Two ways of Implementing Data Objects

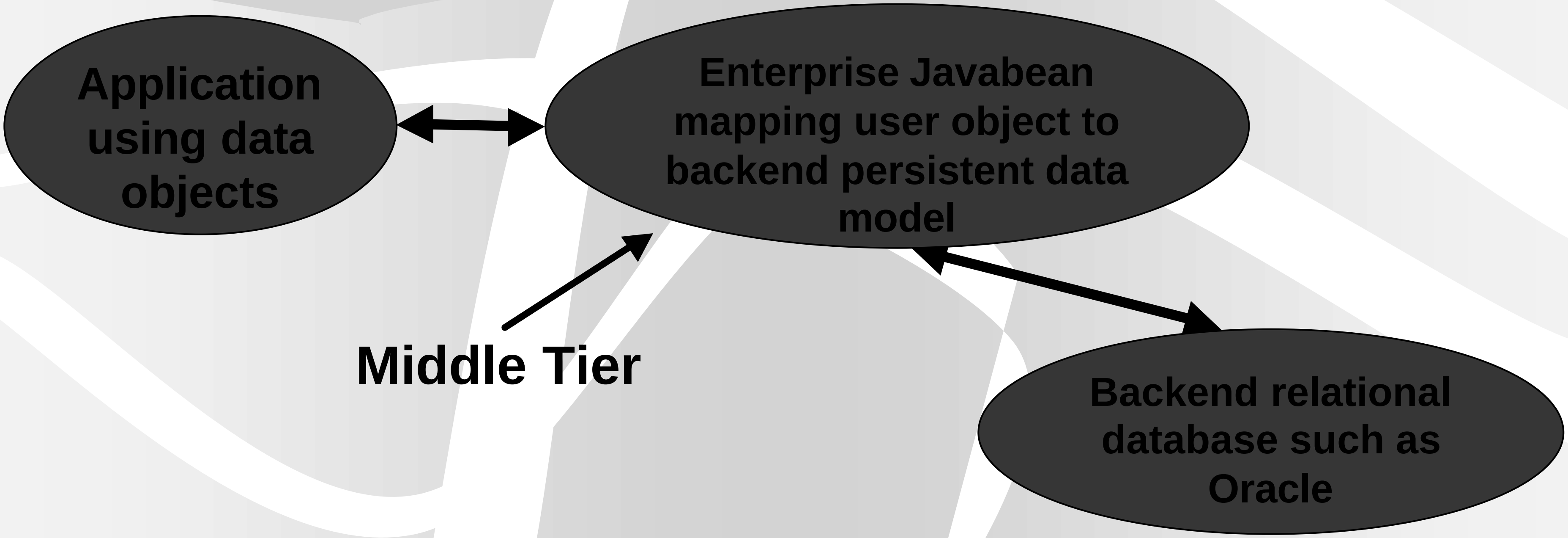
- Old way: Use an Object Database



Application using data objects

Object Database

- Current Approach: Use a Relational Database and business logic in EJB



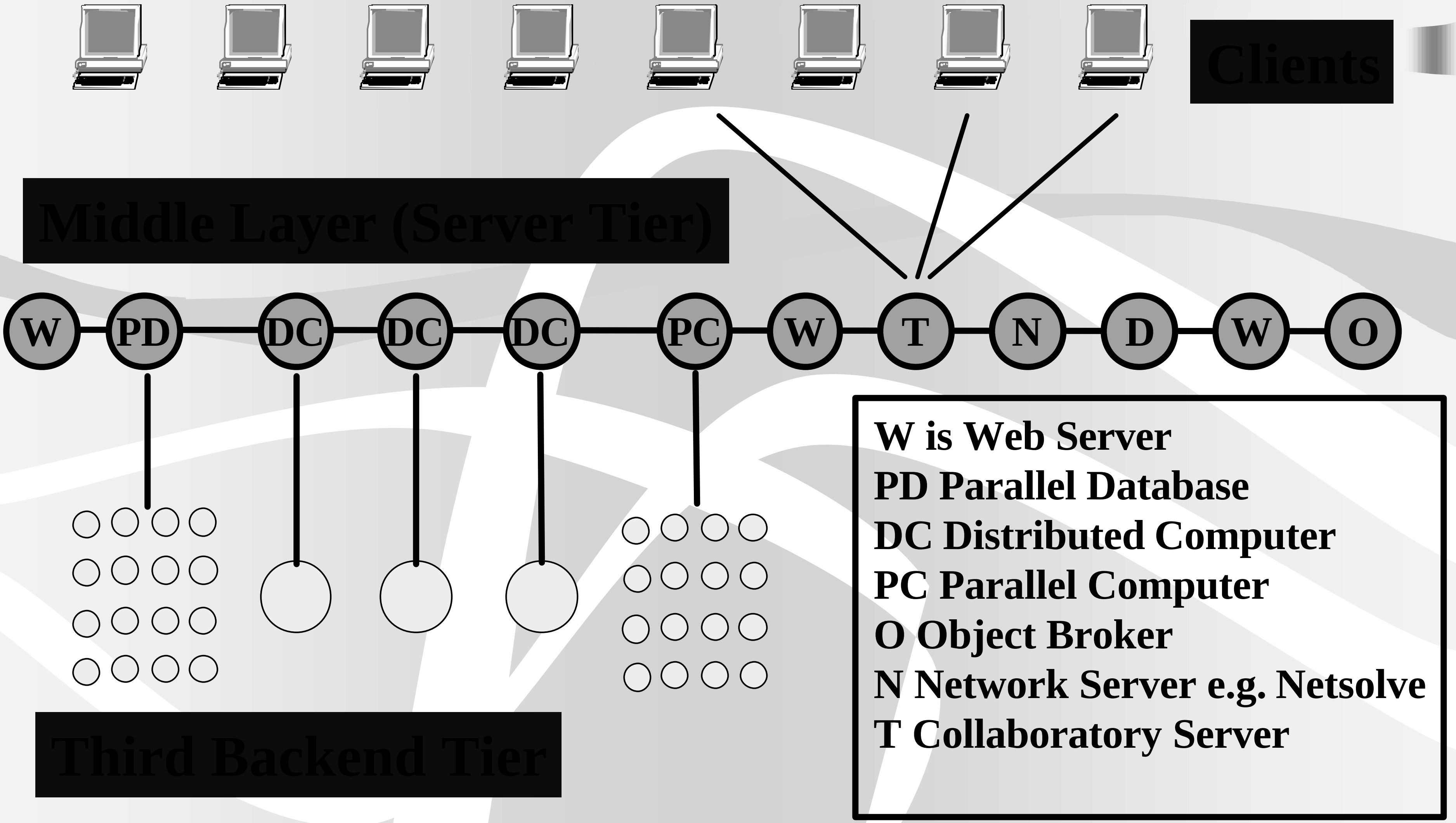
Application using data objects

Enterprise Javabean mapping user object to backend persistent data model

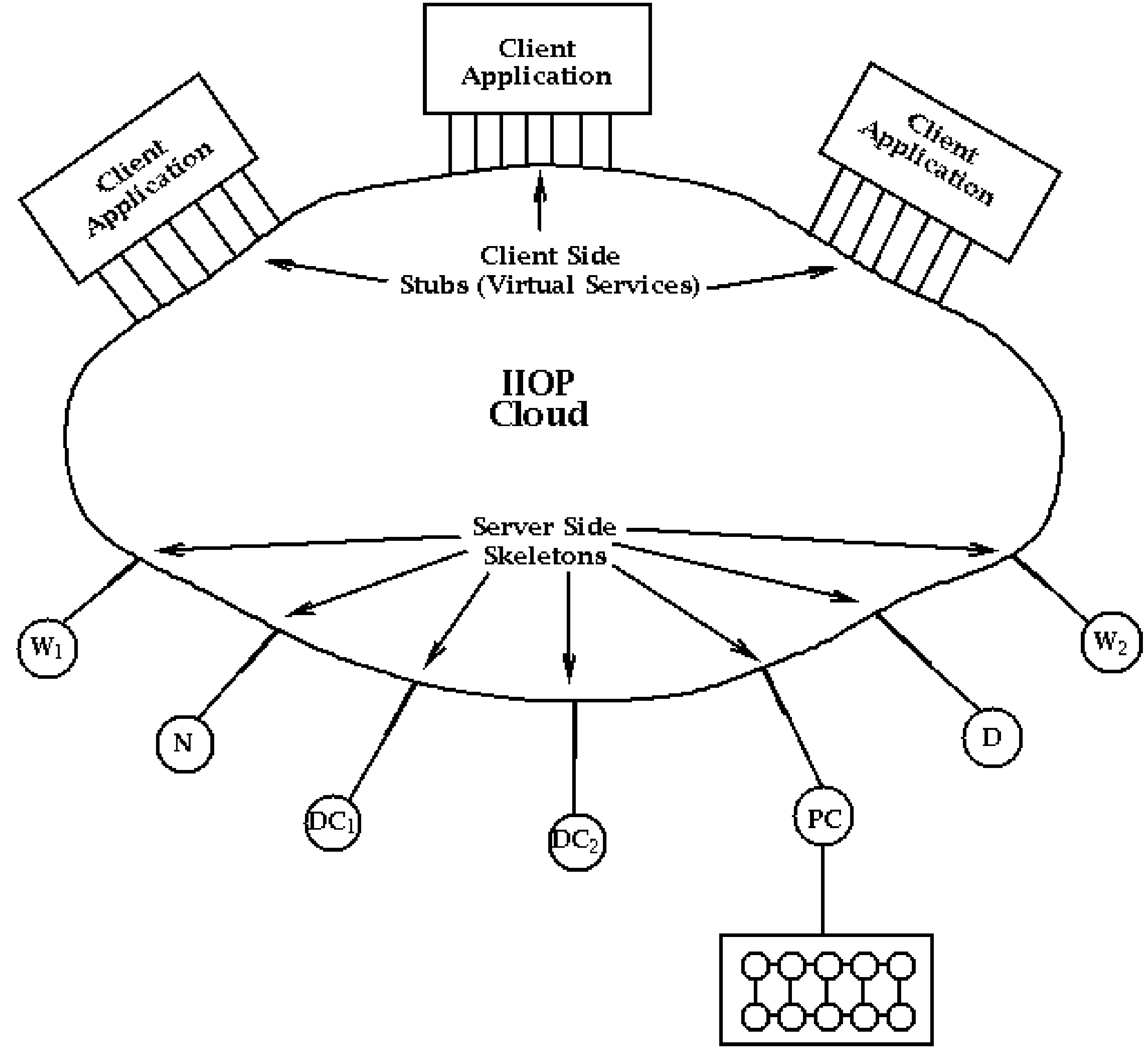
Middle Tier

Backend relational database such as Oracle

Today’s Distributed Object Web: The Confusing Multi-Technology Real World Middleware Server Layer



W is Web Server
PD Parallel Database
DC Distributed Computer
PC Parallel Computer
O Object Broker
N Network Server e.g. Netsolve
T Collaboratory Server

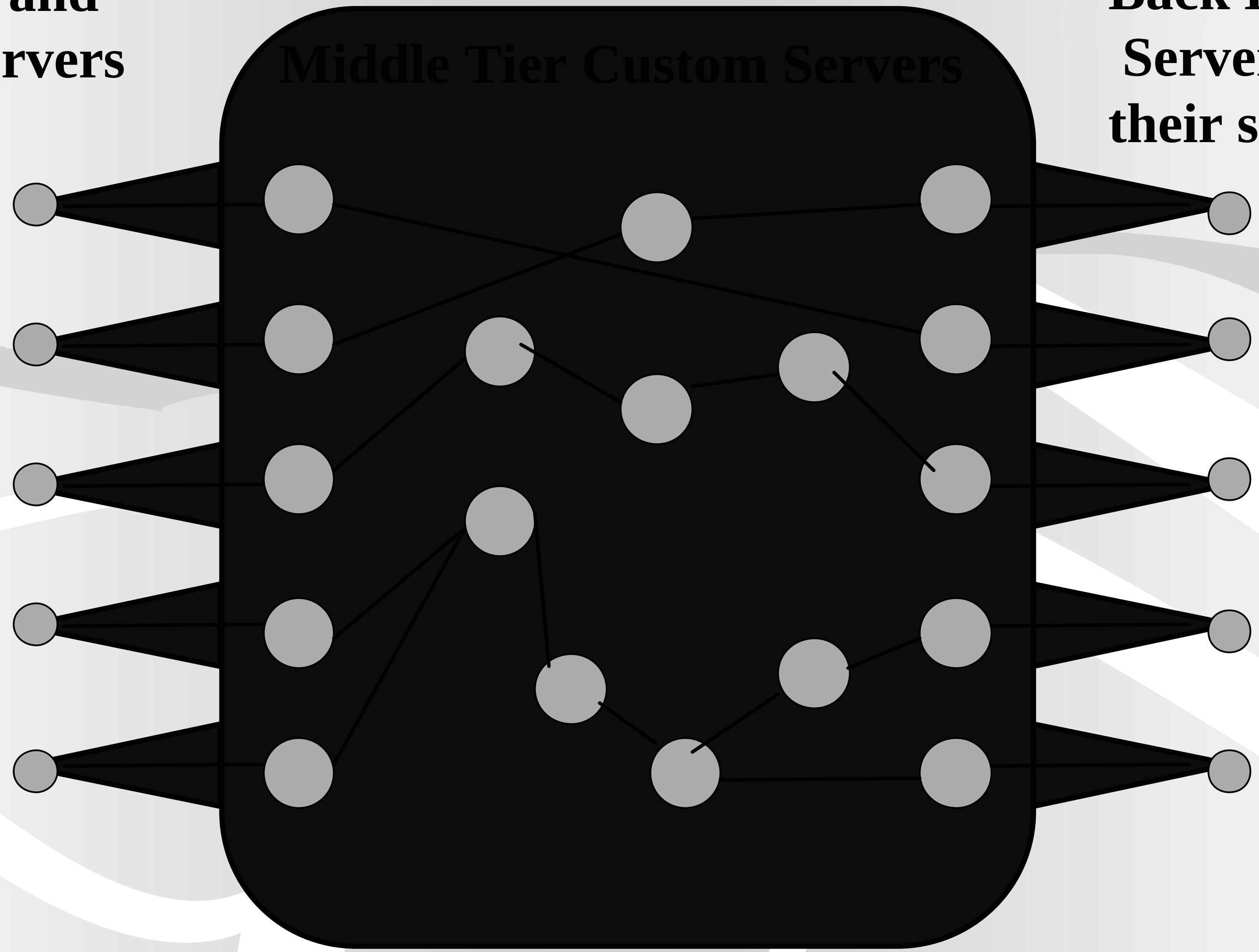


Todays Complex World may evolve to something like the pure CORBA Architecture for a distributed Information System (There are similar COM and Javabean /RMI Versions)

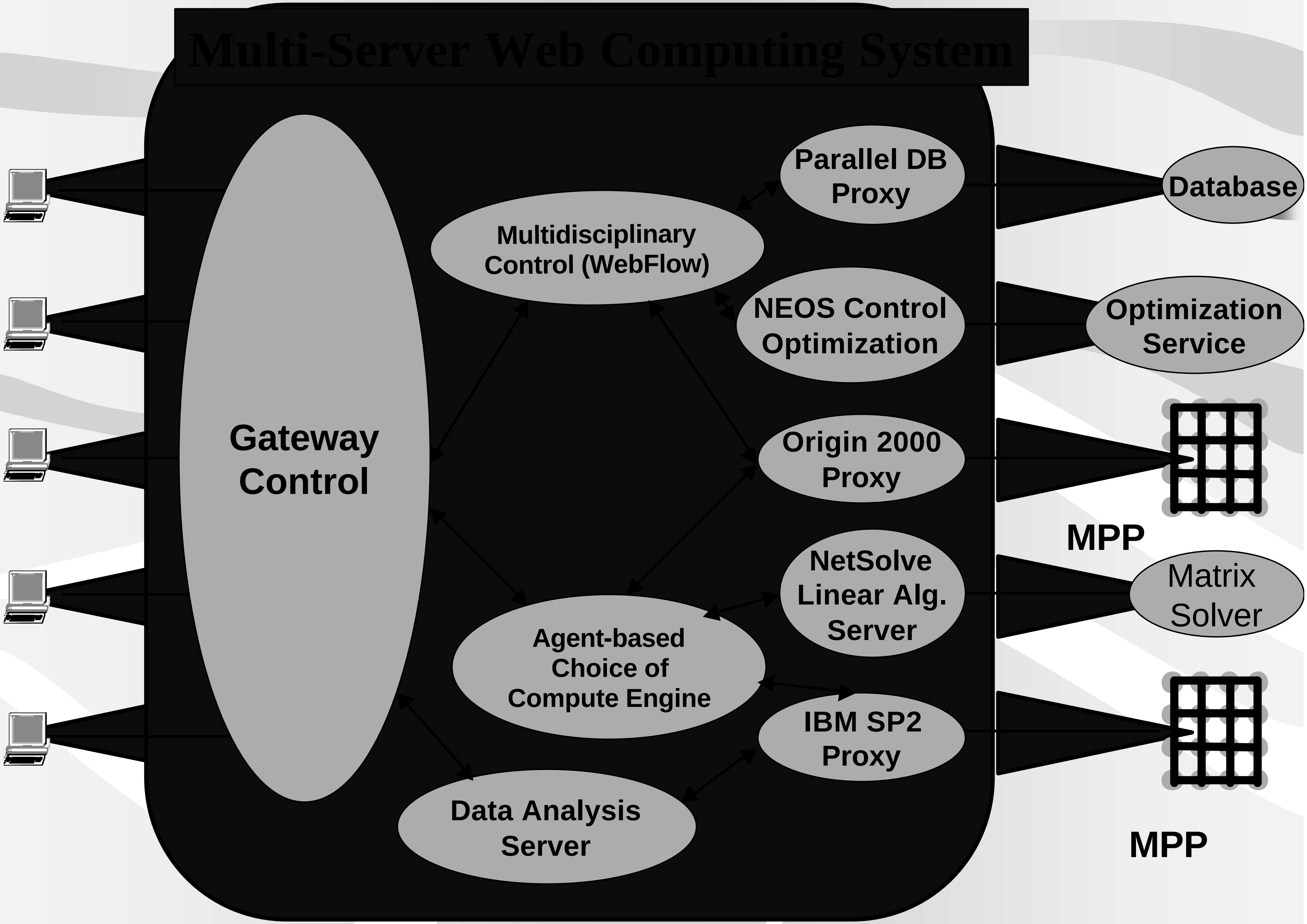
Emerging Object Web Multi-Server Model

Clients and
their servers

Back End
Servers and
their services



Multi-Server Web Computing System



NPAC Concept: Summary of Pragmatic Object Web

- 3-(or more)-tier architecture - Web browser front-ends, legacy (e.g. databases, HPC modules) backends; fat middleware
- Use as appropriate the alternative / competing Middleware models:
  - Java RMI+ EJB (Enterprise Javabean) - single language solution by Sun
  - CORBA - all languages solution by OMG
  - COM - multi-language solution by Microsoft
  - WOM/XML - emergent solution by the Web Consortium
- Each model has different tradeoffs (most elegant, powerful, fastest, simplest)
- POW attempts to integrate various models and services in terms of multi-protocol middleware servers (JWORB)
  - Note Java is often the best language to build middleware whether this is Java or some other distributed object model
  - Most commercial Java activity is on Server not Client

References for Distributed Object Computing

- Frameworks for Component-based Client/Server Computing, Scott M. Lewandowski, ACM Computing Surveys, March 1998.
- Why interaction is more powerful than algorithms, by Peter Wegner, Communications of the ACM, May 1997.
- Customization and Composition of Distributed Objects: Middleware Abstractions for Policy Management, Mark Astley and Gul Agha, Proceedings of the ACM SIGSOFT symposium on the Foundations of Software Engineering, November 1998.

Web Technologies in a Nutshell -- Java

- Java -- Objected Oriented version of C/C++ supporting Interactive Distributed Computing. Original Web architecture (e.g. CGI) was server-side. Java allows design and Implementation of balanced Client Server Applications
- Java likely to be a dominant software engineering and Scientific Computing language -- see http://www.javagrande.org
- This course will not discuss Java as a language but rather as a system building tool
- Java will probably be preferred language for development of next generation general or custom Web servers and clients
  - NPAC's TANGO collaboratory built around a custom Java Tier-2 server
- Java can build client side customized GUI's and graphics/image processing but JavaScript and DHTML competes here and MOST Industry use of Java is in middle tier
- New Java 1.1/1.2 have several enhancements including very many specialized API's
- Javabeans are (visual) component model for Java applications

Web Technologies in a Nutshell - JavaScript

- JavaScript -- only superficially related to Java and was called LiveScript -- is Netscape's (somewhat supported by Microsoft) fully interpreted Client side extension of HTML. This is a good Client Window integration /customization technology where flexibility more important than performance
- i.e. use JavaScript for Rapid Prototyping of Complex User Interfaces
  - First examples use JavaScript together with frames (HTML extension) for interactive multi-window technologies
  - JavaScript is roughly equivalent to "Abstract Windowing Toolkit/ Layout Manager" in Java but applied to Browser Frames and not Java windows
  - JavaScript cannot build complex filters or simulations as slow
  - But JavaScript with dynamic HTML is powerful client technology which is often easier and faster than Java -- it is faster as invokes optimized browser functions
  - both Internet Explorer 4 and Netscape have excellent JavaScript support
- Server side version of JavaScript called LiveWire runs on Netscape Servers -- unsuccessful
- Expect client side use of JavaScript to grow in importance

Web Technologies in a Nutshell - DHTML

- There is an emerging DOM or Document Object Model which will be uniform model used by W3C, Netscape, Microsoft
  - It allow you to address individual components of a page e.g. text box, image or collections thereof as separate entities
  - DOM is quite close to IE 4.0 conventions
- Cascading Style Sheets allow one more powerful ways of assigning properties (such as color fonts etc.) to these components using either name(id) or type (<h2> tag etc.)
- DHTML or dynamic HTML allows one to address the components of document and change on the fly (without reloading page) the properties of these components
  - This includes not only natural style properties but also position, size and "visibility"
  - DHTML currently handicapped by major differences between IE4 and Netscape 4 -- functionalities are similar but syntax very different
  - JavaScript combined with DHTML allows animations, graphs and replacement of just parts of text

Web Technologies in a Nutshell - XML

- HTML is powerful but does not separate display and form (structure of document component as an object)
- XML is a generalization of HTML which allows definition of arbitrary tags
- e.g. <student name="Jane Doe" class="CPS616" grade="…" >Working Hard</student> is more elegant way of capturing information in a reliable fashion than HTML
- <h2>Students</h2>
  <ul><li>Jane Doe: Working Hard</li><ul>
          <li>Class: CPS616</li>
          <li>Grade: …</li> …. </ul>
  </ul> with a PERL program to extract data
- XML allows powerful way of defining dynamic Ascii databases useful for "modest size data" such as people, document citations etc.
- XML parsers map XML tags into HTML for display
- XML can also be used to define extensions to HTML such as special tags for mathematics or chemistry or …..
- XML defines syntax for "serializing" Web objects and transmitting between clients and servers

Web Technologies in a Nutshell - PERL

- PERL is a relatively old technology which is being overtaken by Java tidal wave.
- Still PERL has significantly better Systems and Document handling capability than Java
  - Very good for UNIX as much easier than Shell for system scripts -- PC versions exist but not so well integrated into O/S
  - Wonderful regular expression handling
- PERL is traditional but probably not best choice for server CGI extensions and development of filters
- -- except for simpler cases involving text documents
- PERL5 is object oriented but much less elegant (in my opinion) than Java
  - PERL5 has very useful multidimensional associative and regular arrays

Web Technologies in a Nutshell - Databases

- The Web provides a convenient integration environment for "mature" technologies migrating from existing computer environments.
- Relational databases are a good example where it is now straightforward in Microsoft Access, Oracle, DB2, Informix, Sybase etc. to provide a Web Interface which can be used for data (mail, curricula material etc.) with Java/JavaScript/Forms based Interfaces
- Object databases such as Illustra also interfaced to Web
- Systems such as Cold Fusion provide convenient high level interfaces to Web-linked databases
- Several excellent Java to Database packages becoming available with the JDBC standard based on ODBC -- more powerful but lower level than systems like Cold Fusion
- CORBA will have good Web and Java Interfaces and we will discuss integration of Web CORBA and database technologies
  - CORBA views a database as a managed persistent object

Web Technologies in a Nutshell - VRML

- VRML plays same role to 3D worlds that HTML does to documents
- VRML 1.0 has been widely available and specifies static 3D scenes through which you can navigate. Already provides universal visualization environment and we have examples of use In Geographical Information Systems
  - Note can embed clickable URL's as with ImageMaps which can be used to annotate images to provide interactive resources
- VRML 2.0 is now the standard with critical enhancements so that individual elements of 3D world are dynamic and can be programmed
  - It is designed to support full interactivity (televirtuality) with texture mapped video, avatars etc.
  - VRML 2.0 could require huge computing resources whether used as the virtual car-dealership / interactivity gaming or more academic uses such as collaboration between teachers and students in 3D virtual classroom
- Bandwidth and computing needs of VRML are handicapping acceptance and appears that VRML will NOT “make it” -- replacement unclear
  - Microsoft ChromeEffects (XML based) and
  - Java3D address some but not all VRML applications