Computer and Automation Research Institute
Hungarian Academy of Sciences

Distributed Client/Server technology: From Sockets to Object Web

P. Kacsuk

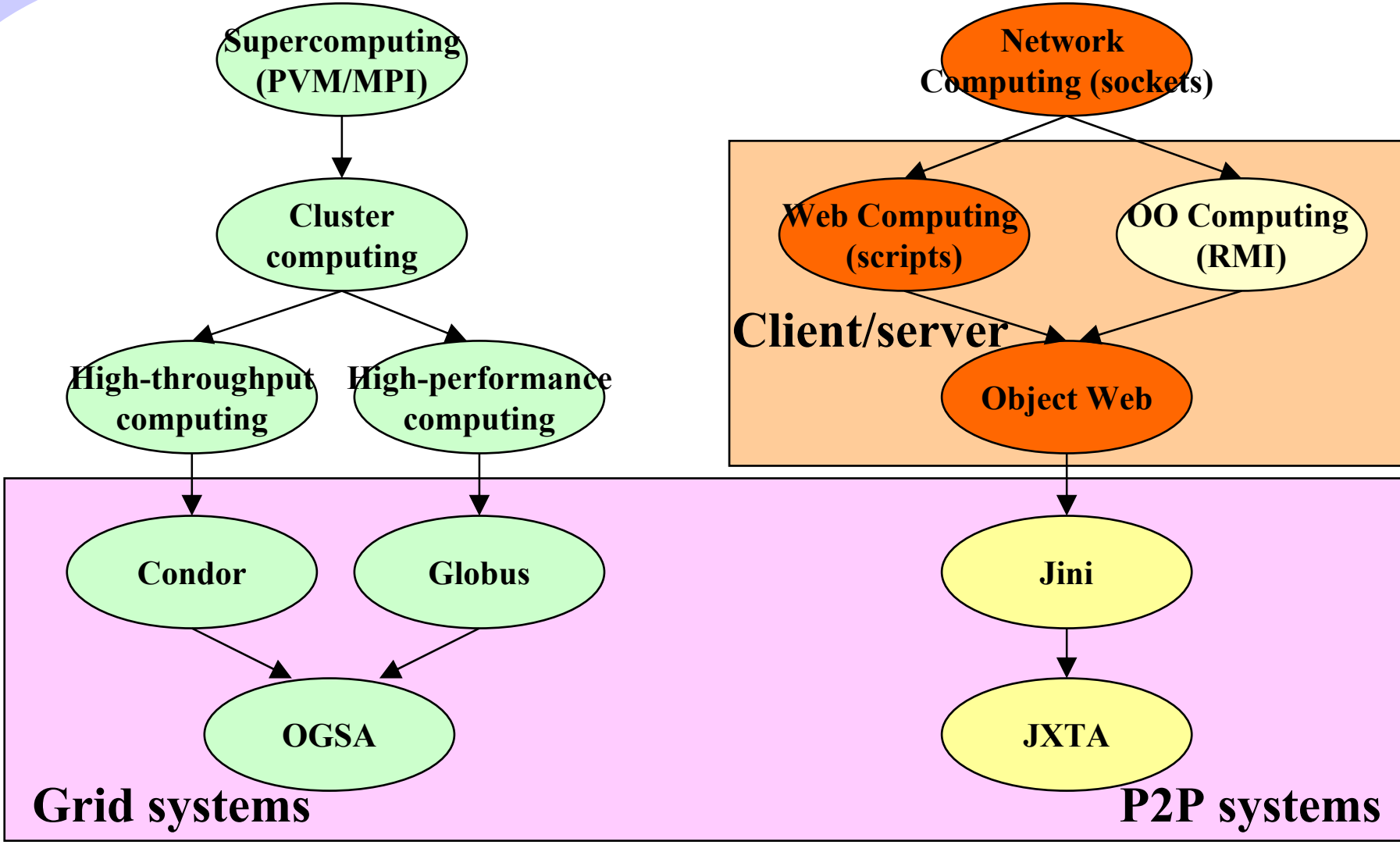
Laboratory of Parallel and Distributed Systems
MTA SZTAKI Research Institute

kacsuk@sztaki.hu
www.lpds.sztaki.hu

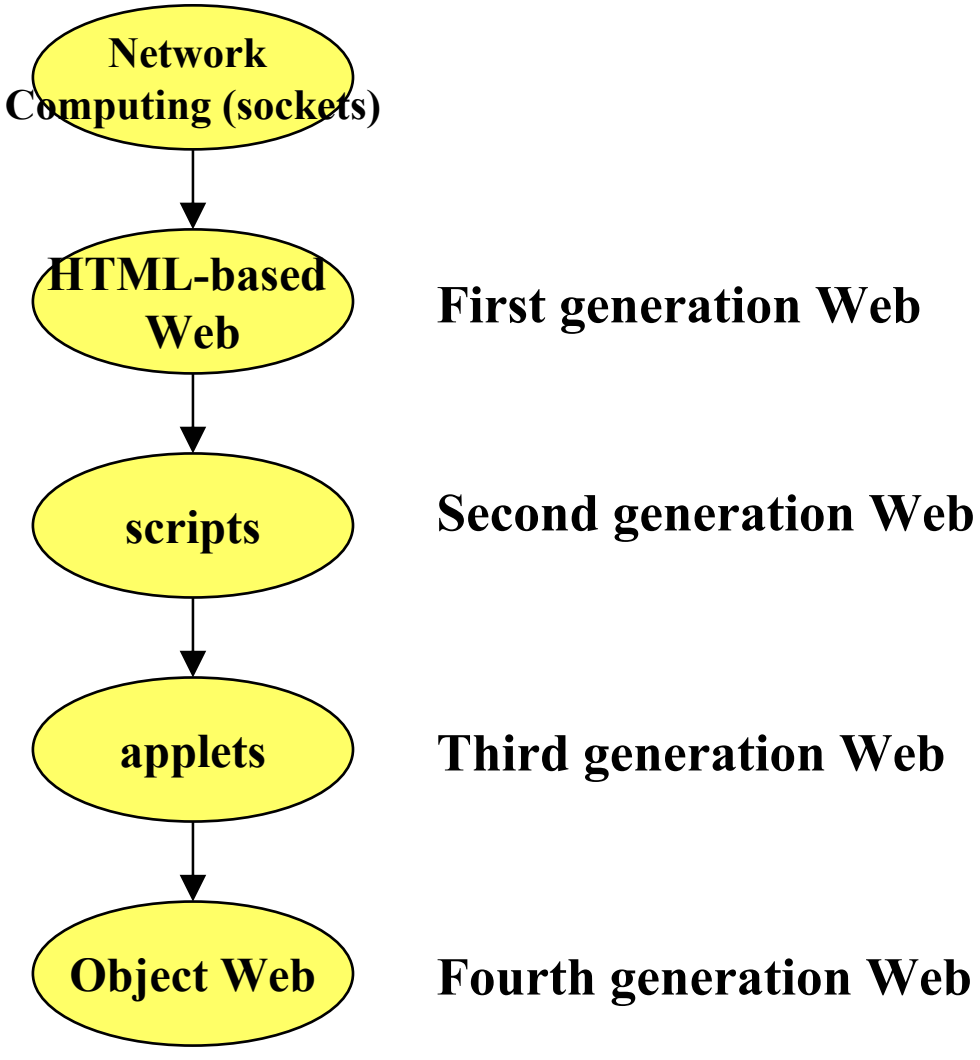
Contents

- Progress in Web technology
  - HTTP
  - Scripts
  - Applets
- Progress in OO client/server technology
  - Java RMI
  - CORBA

Progress to Grid and P2P systems



Progress in Web technology



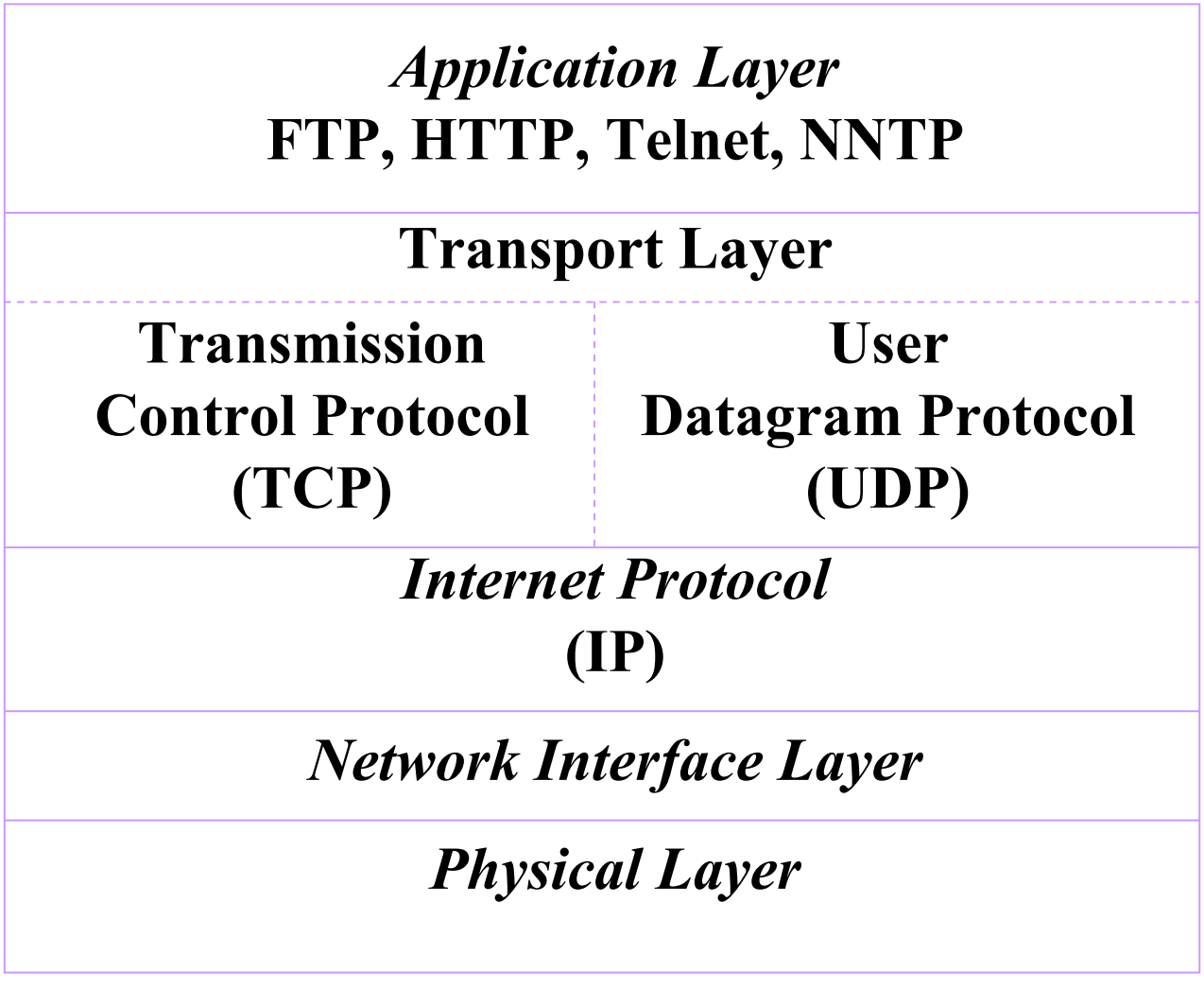
Internet Protocols

Protocols - A set of rules that determine how two computers communicate with one another over a network

- The protocols embody a series of design principles

  - **Interoperable**— the system supports computers and software from different vendors.
  - **Layered**— the collection of Internet protocols work in layers with each layer building on the layers at lower levels.
  - **Simple**— each of the layers in the architecture provides only a few functions or operations.
  - **End-to-End**— the Internet is based on "end-to-end" protocols, i.e., the interpretation of the data happens at the application layer and not at the network layers.

Layered protocol architecture



TCP/IP

- Solves the global internetworking problem
- **Internet Protocol (IP)**
  - Formats the **packets** and assigns **addresses**
    - packets are labeled with the addresses of the sending and receiving computers
- **Transmission Control Protocol (TCP)**
  - Ensures that 2 computers can communicate with one another in a **reliable** fashion

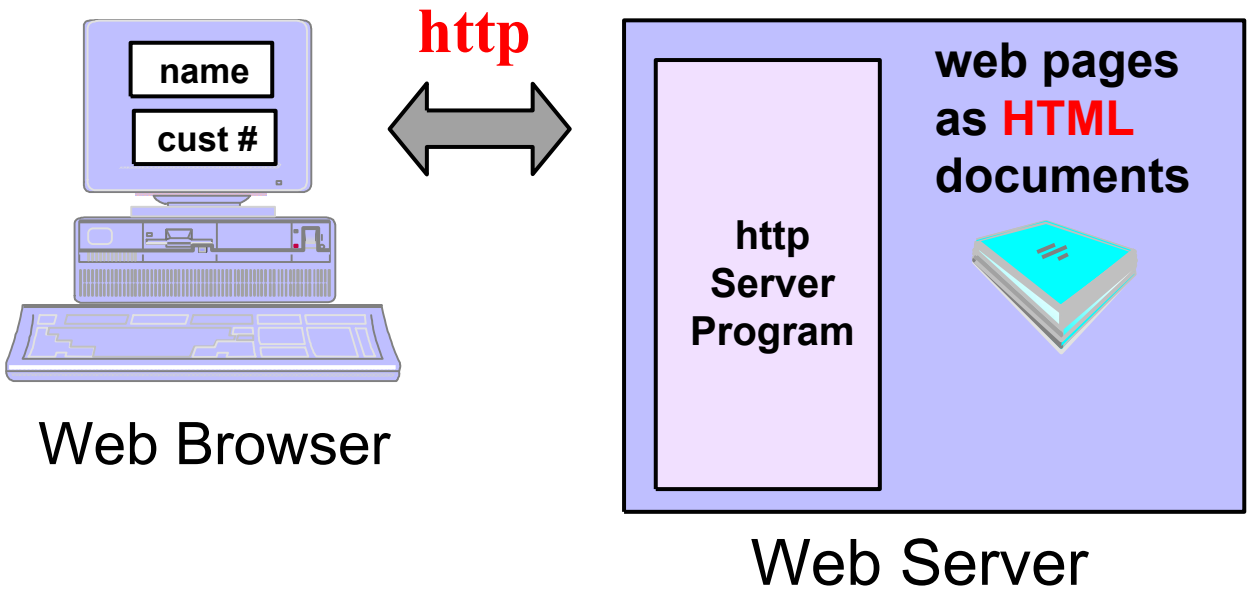
Domain Names

- Reference particular computers on the Internet
- Divided into segments separated by periods
  - For example, in the case of "**lutra.sztaki.hu**"
    - "**hu**" is the top level domain
    - "**sztaki**" is the subdomain
    - "**lutra**" is the specific computer
  - Internet Assigned Numbers Authority (IANA)
    - controls the domain name system
  - Network Solutions, Inc. (NSI)
    - issues and administers domain names for most of the top level domains

Internet Client/Server Applications

Application	Protocol	Purpose
E-mail	Simple Mail Transport Protocol (SMTP) Post Office Protocol version 3 (POP3) Multipurpose Internet Mail Extensions (MIME)	Allows the transmission of tex messages and binary attachments across the Intern
File Transfer	File Transfer Protocol (TP)	Enables files to be uploaded and downloaded across the Internet.
Chat	Internet Relay Chat Protocol (IRC)	Provides a way for users to talk to one another in real-time over the Internet.
UseNet Newsgroups	Network News Transfer Protocol (NNTP)	Discussion forums where user can asynchronously post messages and read messages posted by others.
World Wide Web (Web)	Hypertext Transport Protocol (HTTP)	Offers access to hypertext documents, executable programs, and other Internet resources.

HTML-based Web



- Access to HTML documents
- Only information (data) retrieval

Web-enabling technologies

- **Web browsers and servers**:
  - Locate each other so they can send requests and responses back and forth
  - Communicate with one another through the Internet
- **Uniform Resource Locators (URLs)**
  - A new addressing scheme
  - Allowing almost any kind of information to be retrieved from almost anywhere on the Internet
- **The HyperText Transfer Protocol (HTTP)**
  - The lingua franca of Web browsers and servers
  - Allowing many different programs to work together
- **The HyperText Markup Language (HTML)**
  - Allowing authors to create multimedia hypertext that can be used by any Web browsers

Web-based Client/Server (cont.)

- **Hypertext Transport Protocol (HTTP)**
  - Lightweight, stateless protocol that browsers and servers use to communicate with one another
  - Statelessness - **every request that a browser makes opens a new connection that is immediately closed after the document is returned**
  - MIME (Multipurpose Internet Mail Extension)
    - describes the contents of the document
    - in the case of an HTML page the header is "Content-type: text/html"

Hypertext Transfer Protocol (HTTP)

Client/server protocol designed to access URL-named resources

Stateless protocol:

Every request that a browser makes opens a new connection that is immediately closed after the document is returned

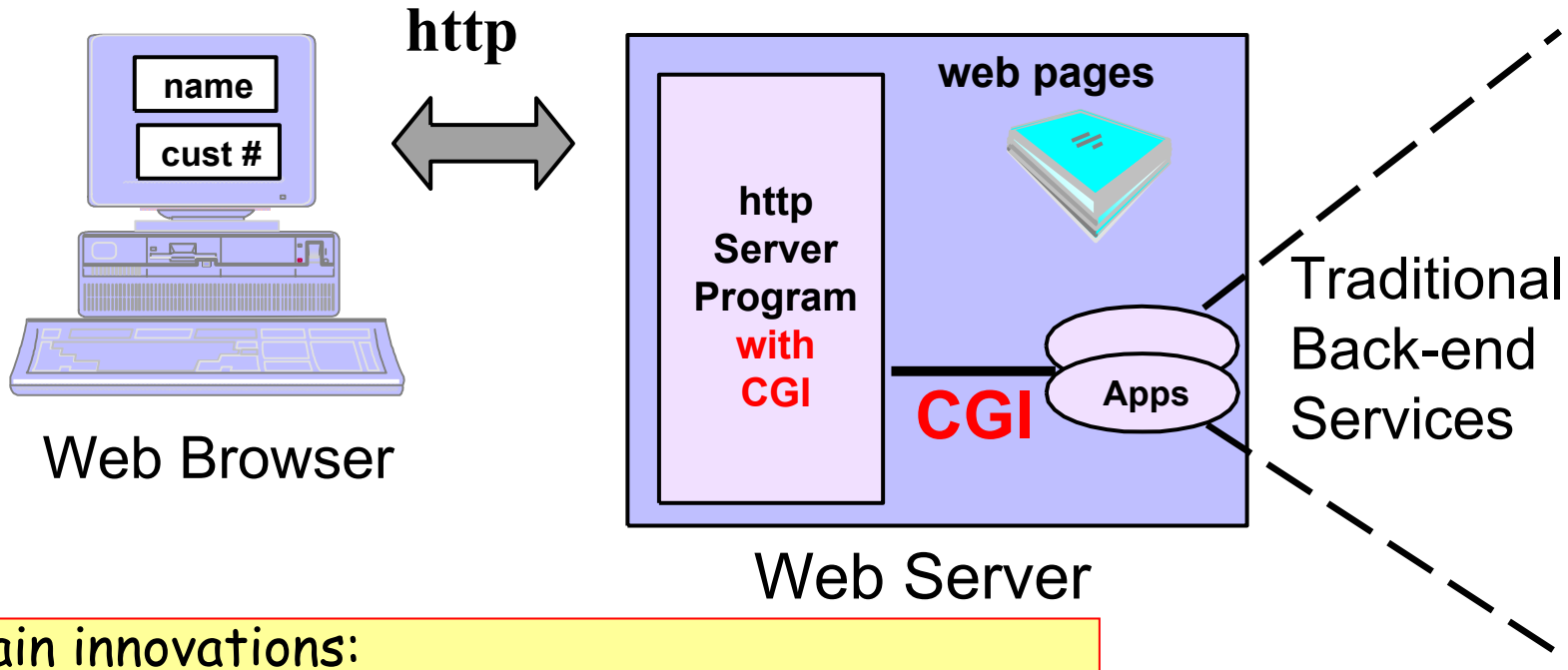
Http can:

- Transfer requests from client to server
- Deliver requested resources (text, multimedia) from server to client
- Invoke other programs to handle processing requests from clients

Http cannot:

Process data

Script-based Web

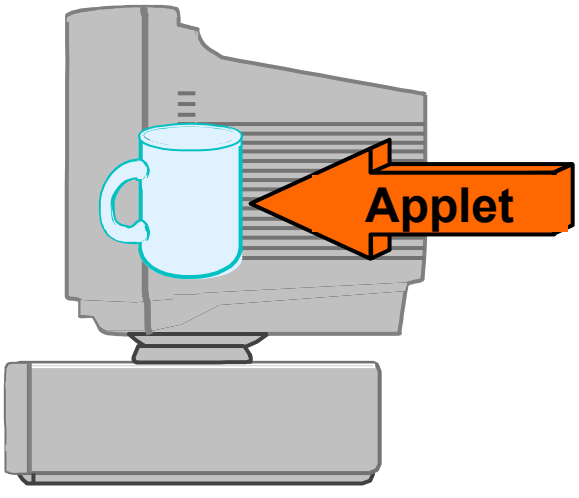


Main innovations:

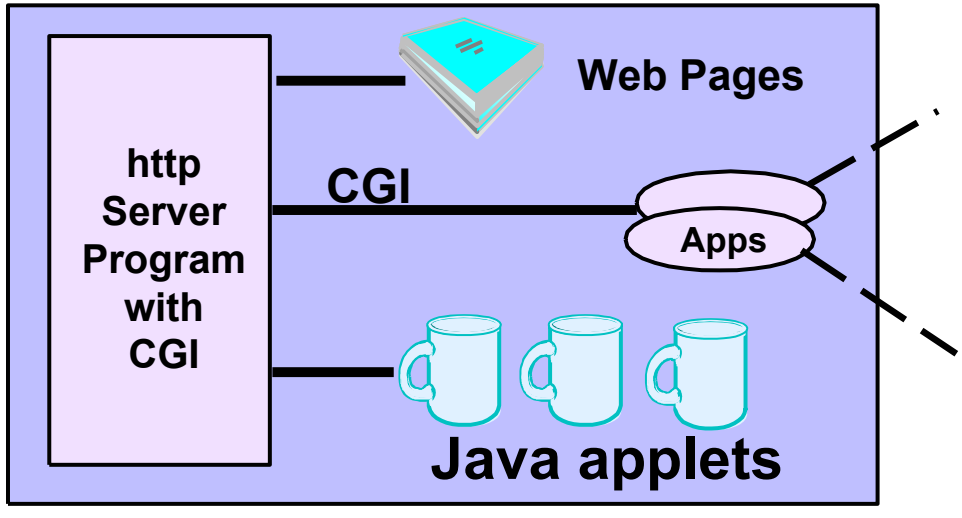
1. Fields, forms - dynamic page content
2. Server-side processing - CGI
3. Access to legacy data

1. A program on a server, triggered by input from a browser
2. The CGI program links the HTTP server to another program, such as a database or transaction system

Applet-based Web



Web Browser



Web Server

Main innovations:

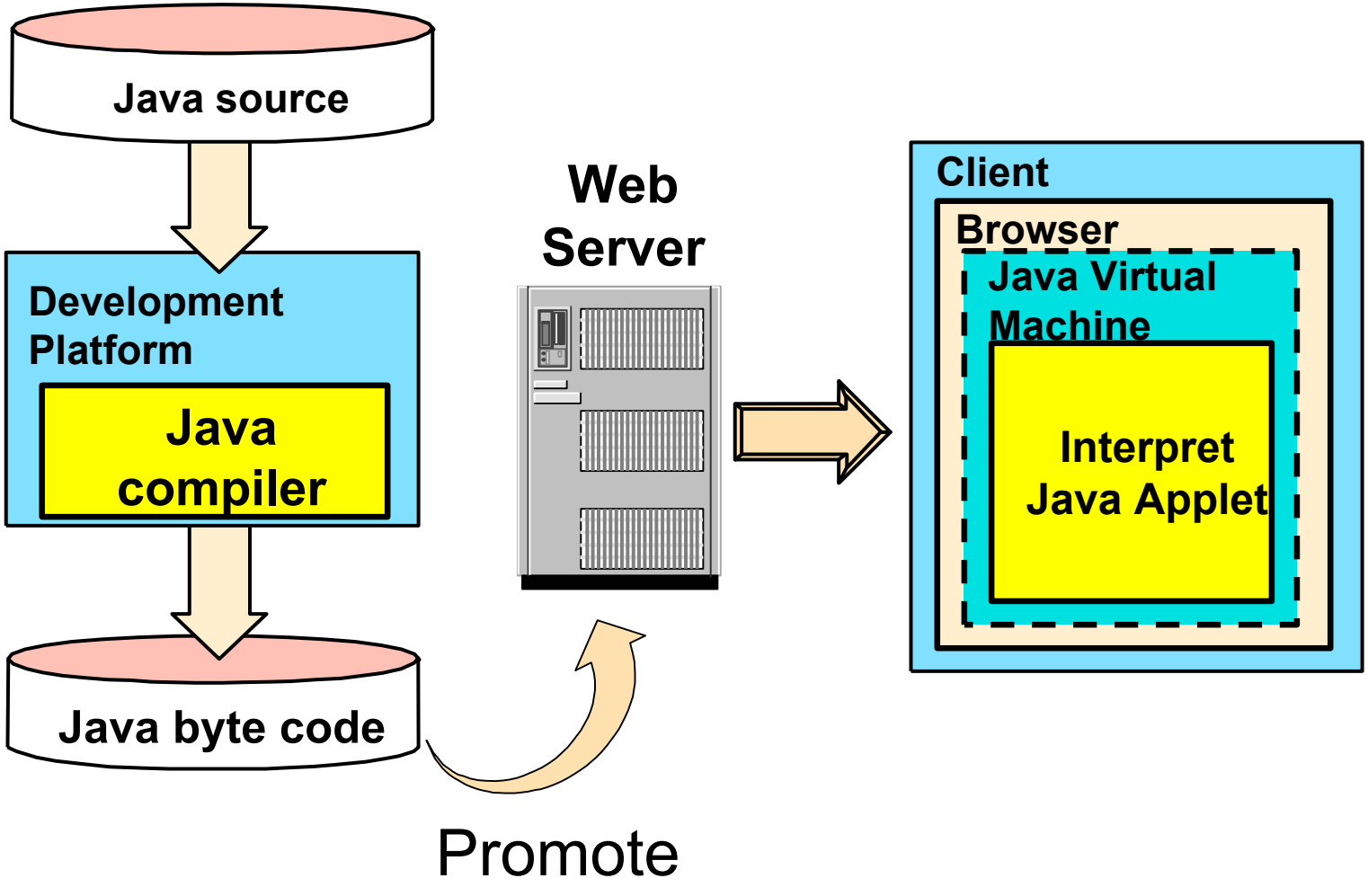
1. Application processing on the **client**
2. Offload server/Faster user response
3. Provide GUI applications thru browser

What is Java?

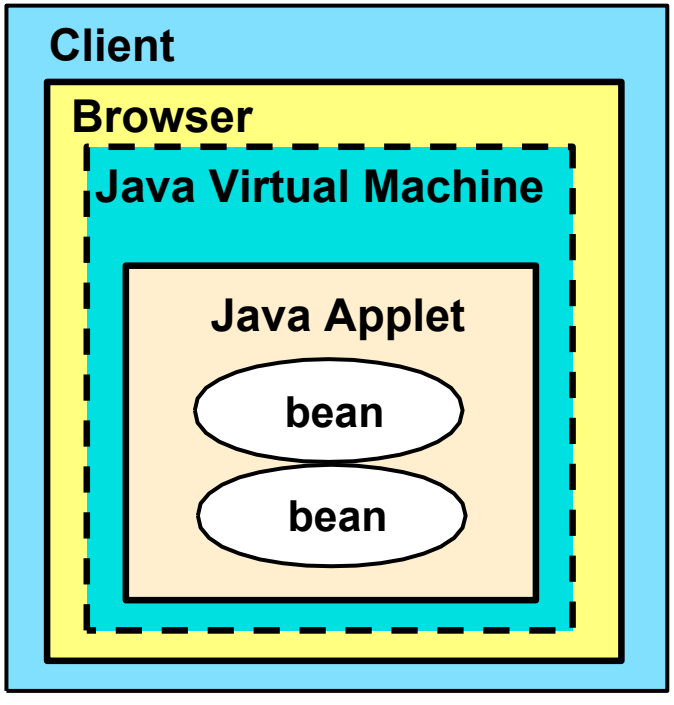
1. An object-oriented programming language developed by Sun

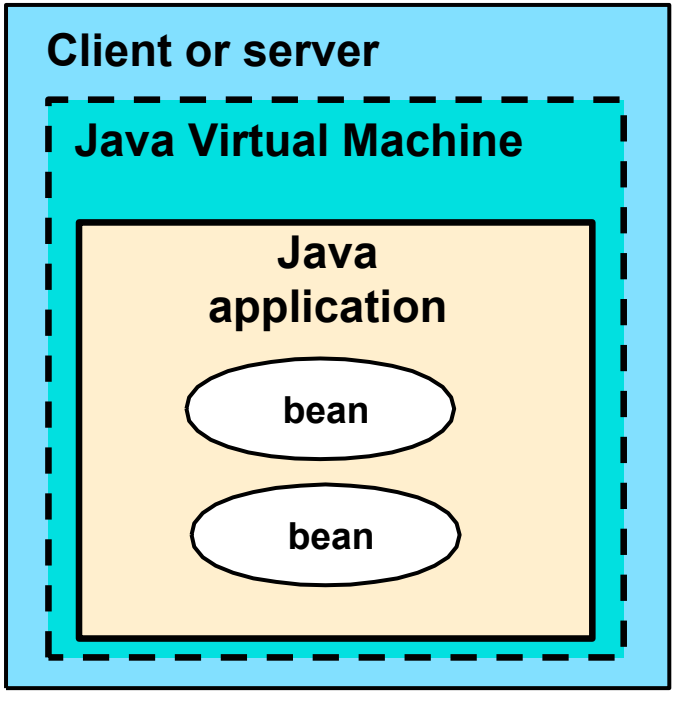
2. Java characteristics:
   - Architecture neutral
   - Small
   - Secure
   - Compiled and Interpreted

How does Java Work?



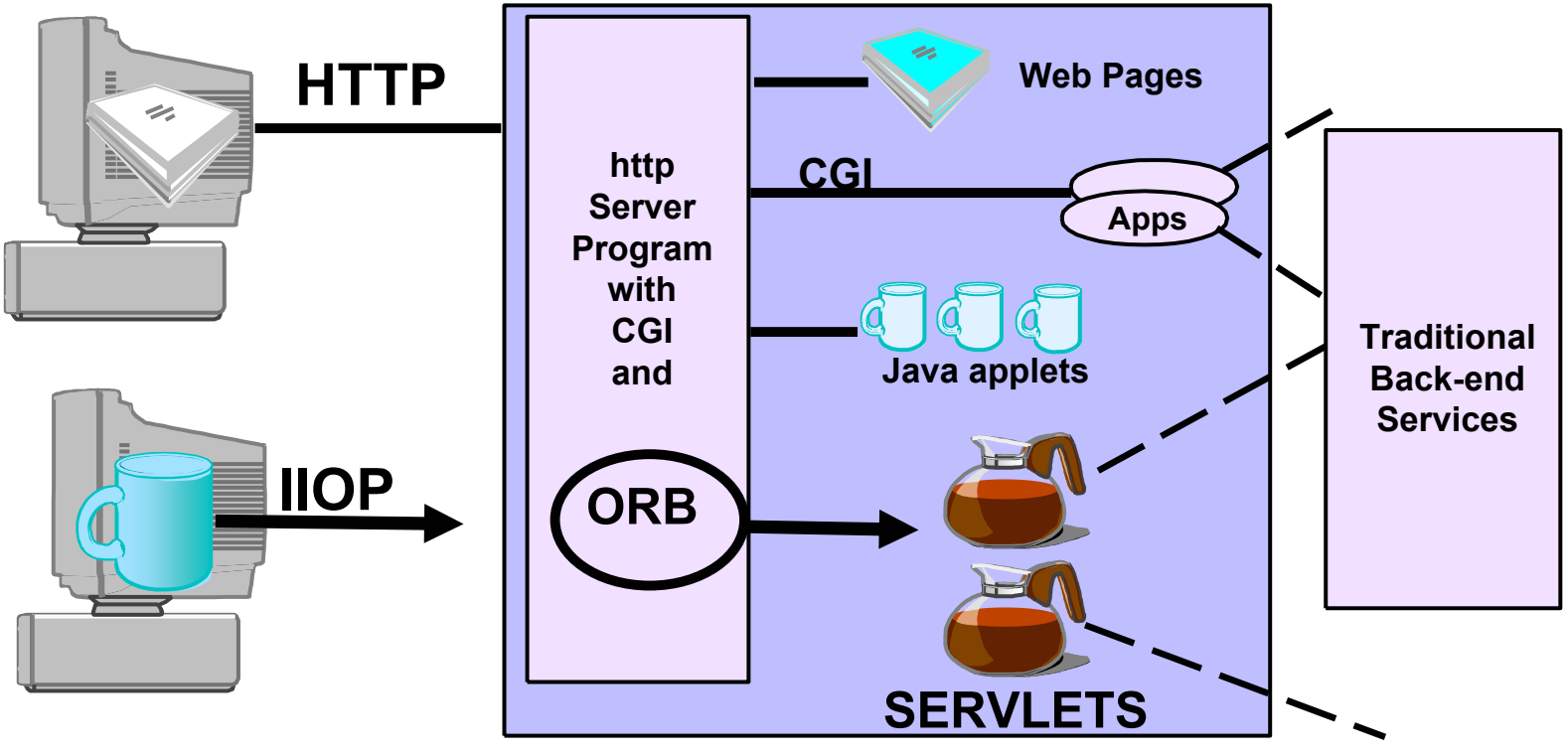
Java Programs



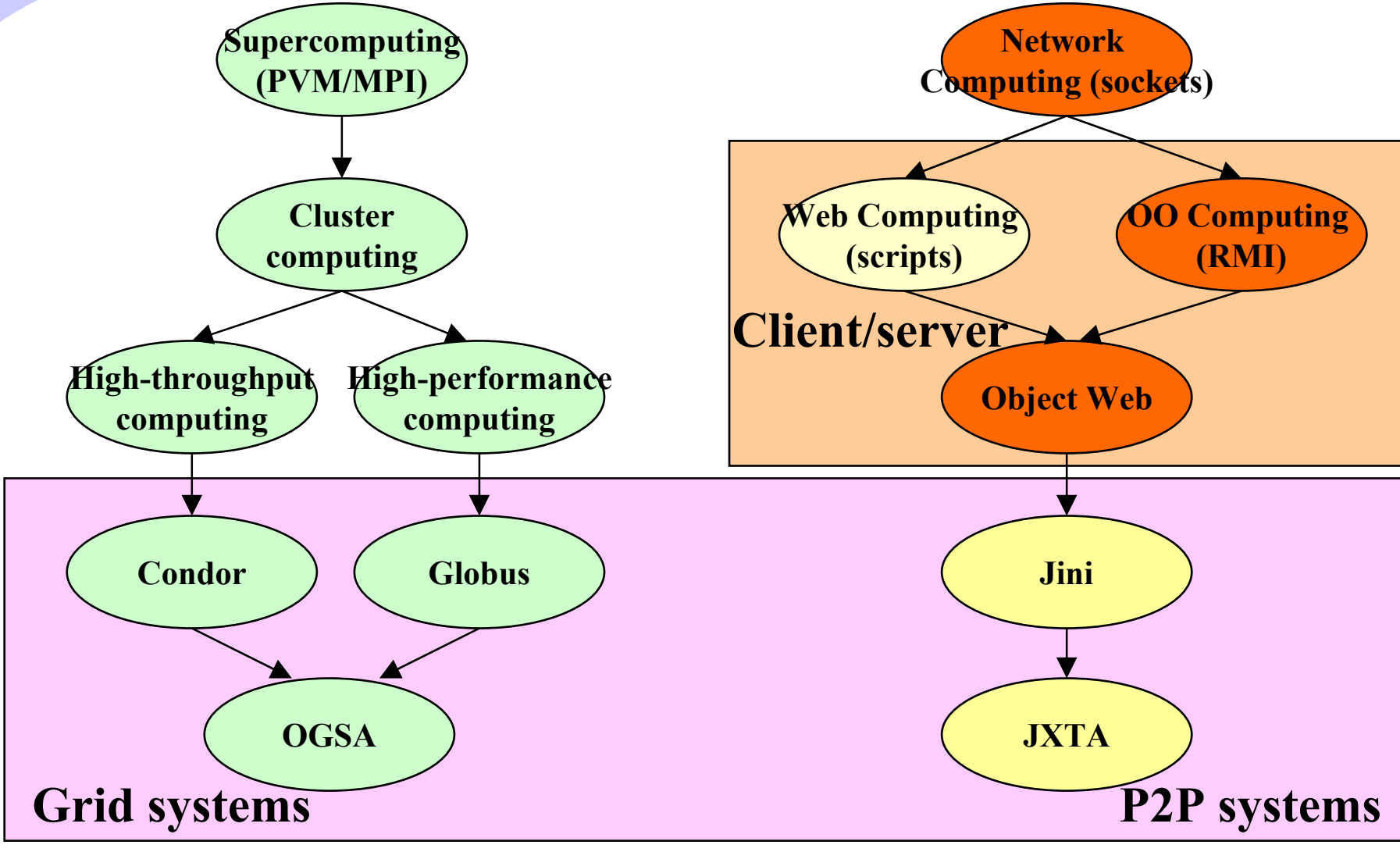


- Applets - Java program for browser use
- Application - stand-alone program or server applet (servlet)
- Java bean - component of applet or application

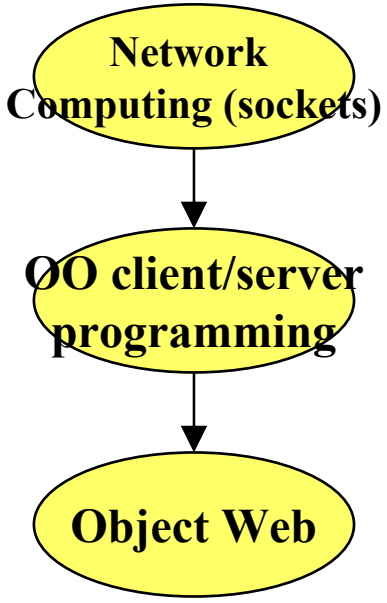
Object Web


HTTP
IIOP
http Server Program with CGI and
ORB
Web Pages
CGI
Apps
Java applets
SERVLETS
Traditional Back-end Services
Clients
Web Server

Progress to Grid and P2P systems



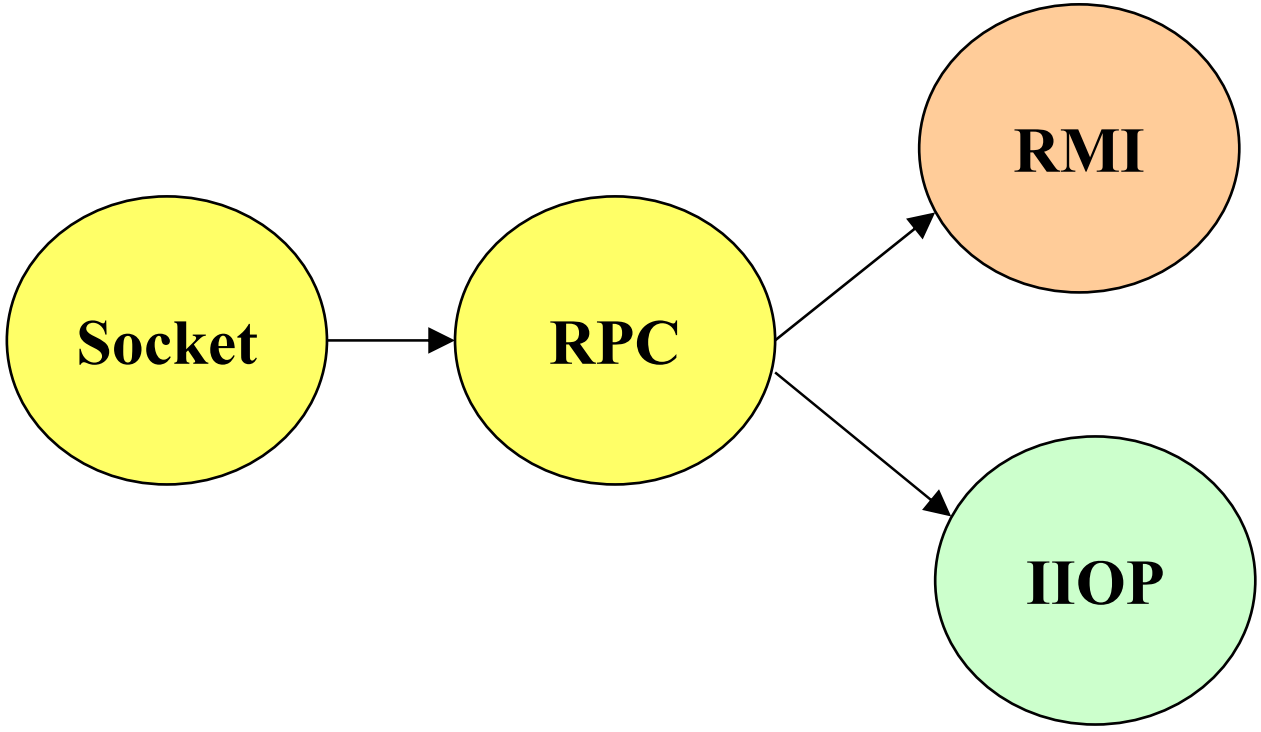
Progress through OO programming



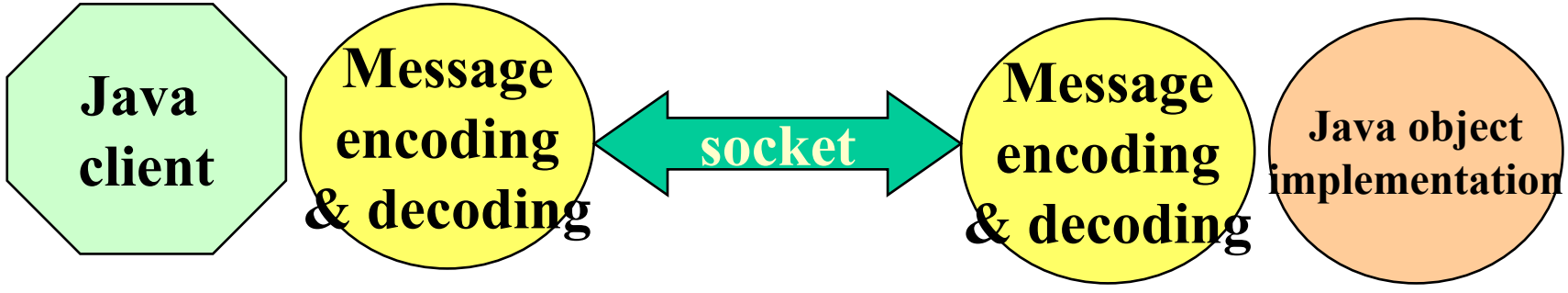
RMI, CORBA, DCOM

Fourth generation Web

Progress from sockets to OO client/server

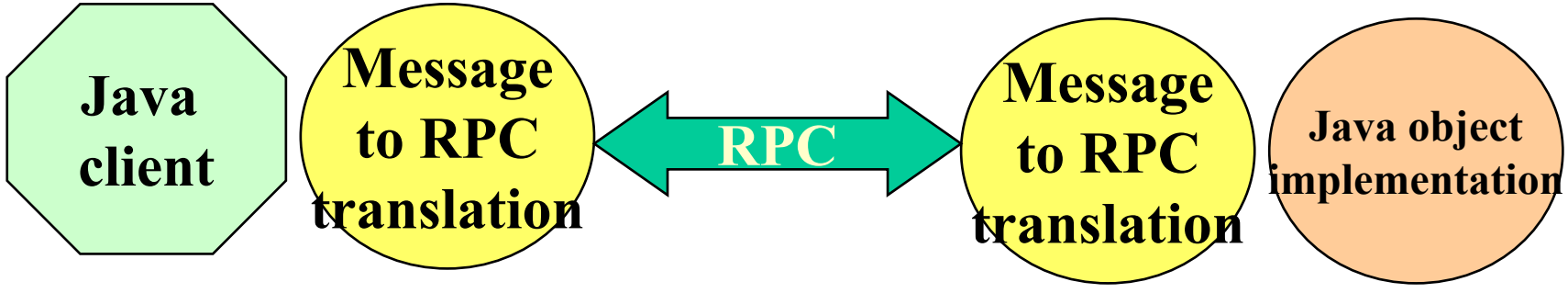


Socket-based Java



- Advantage: best performing approach
- Disadvantages:
  - requires significant programming effort
  - requires encoding and decoding of messages and parameters
  - requires detailed specification of interfaces between objects
  - difficult to reuse
  - errors are discovered only at run-time

RPC-based Java



- Advantage:
  - still well performing approach
  - provides improved mechanisms for communications including the handling of parameters
- Disadvantages:
  - not built for object-oriented application communication
  - requires the application code to map Java communications to a procedure call structure

RMI-based Java

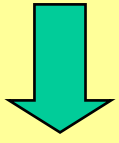
- Advantage:
  - It has a Java-only perspective - RMI was optimised for Java applications
  - Expands invocation support from a local VM to distributed VMs, including callbacks from the server
  - makes local and distributed object models transparent
  - Easy-to-use approach
- Disadvantages:
  - It has a Java-only perspective - many applications consists of a variety of non-Java code

OO Client/server systems

- The three competing technology:
  - **RMI** (Remote Method Invocation)
  - **CORBA** (Common Object Request Broker Architecture)
  - **DCOM**
- **Common solution scheme**
  - The server publishes its object in an available naming directory (predecessor of Grid info system)
    - **RMI registry** in RMI
    - **naming service** in CORBA/DCOM
  - The details of the service are published through an interface definition language:
    - **Java interface** in RMI
    - **IDL** in CORBA/DCOM

OO Client/server systems (2)

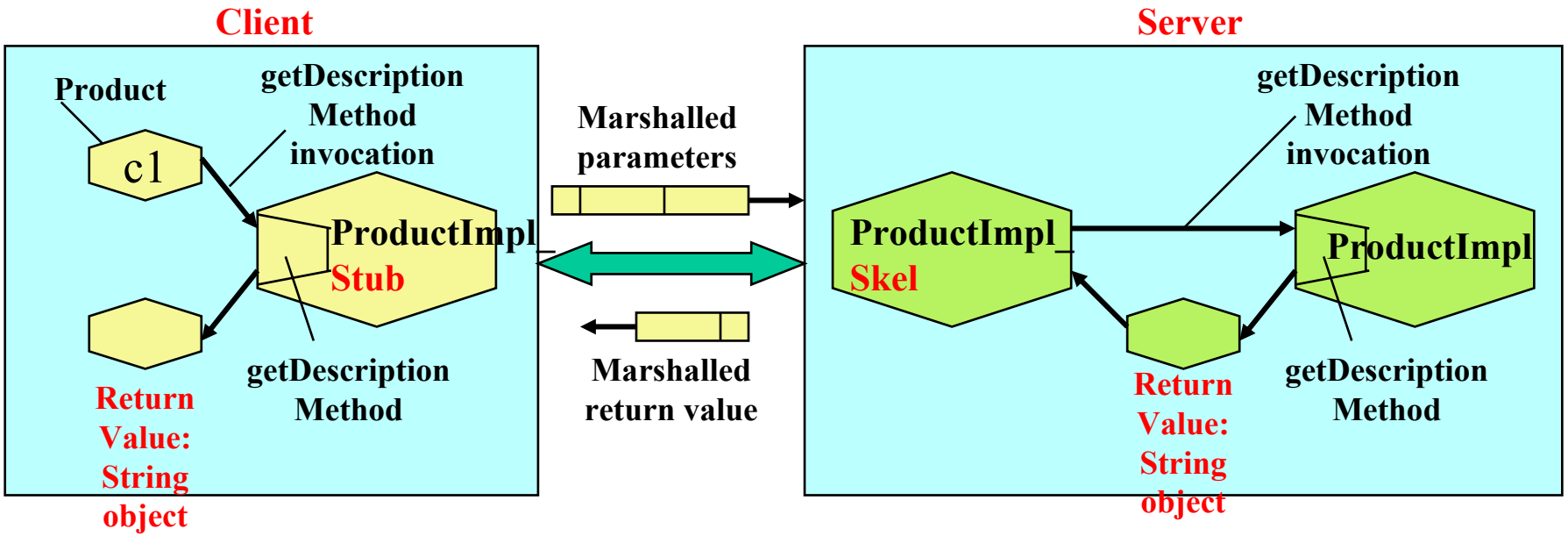
- **Common solution scheme (cont.)**
  - The client should know of the various services.



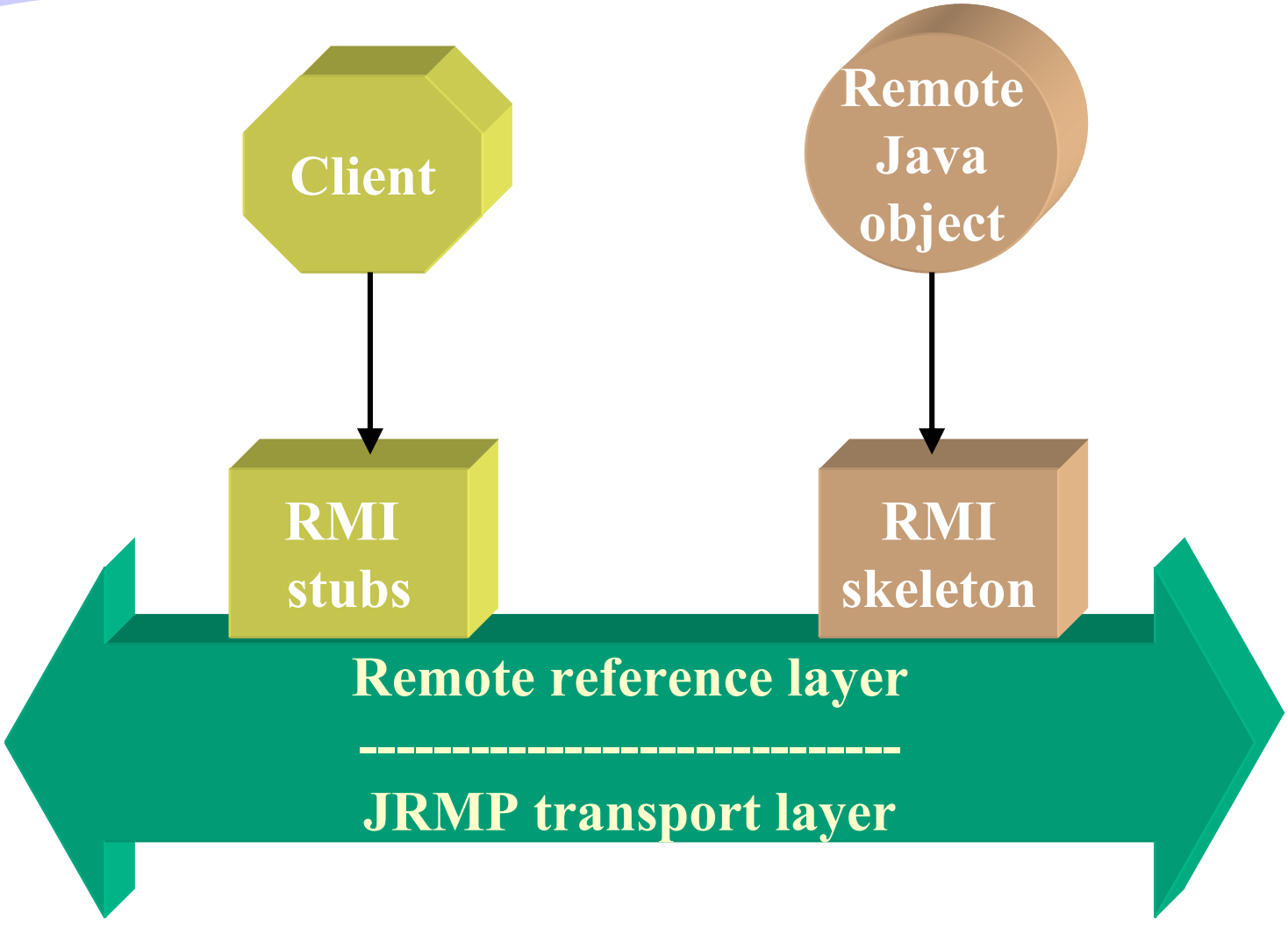
    Prewiring the client with client-side libraries or stubs
  - Any method call invoked by the client passes locally through the client proxy called **stub** that
    - serializes and
    - passes

    the method parameters to a server side object called **skeleton**.
  - The **skeleton**
    - reconstructs the method parameters
    - invokes the called method on the server object
    - returns the method result

Solution scheme in RMI



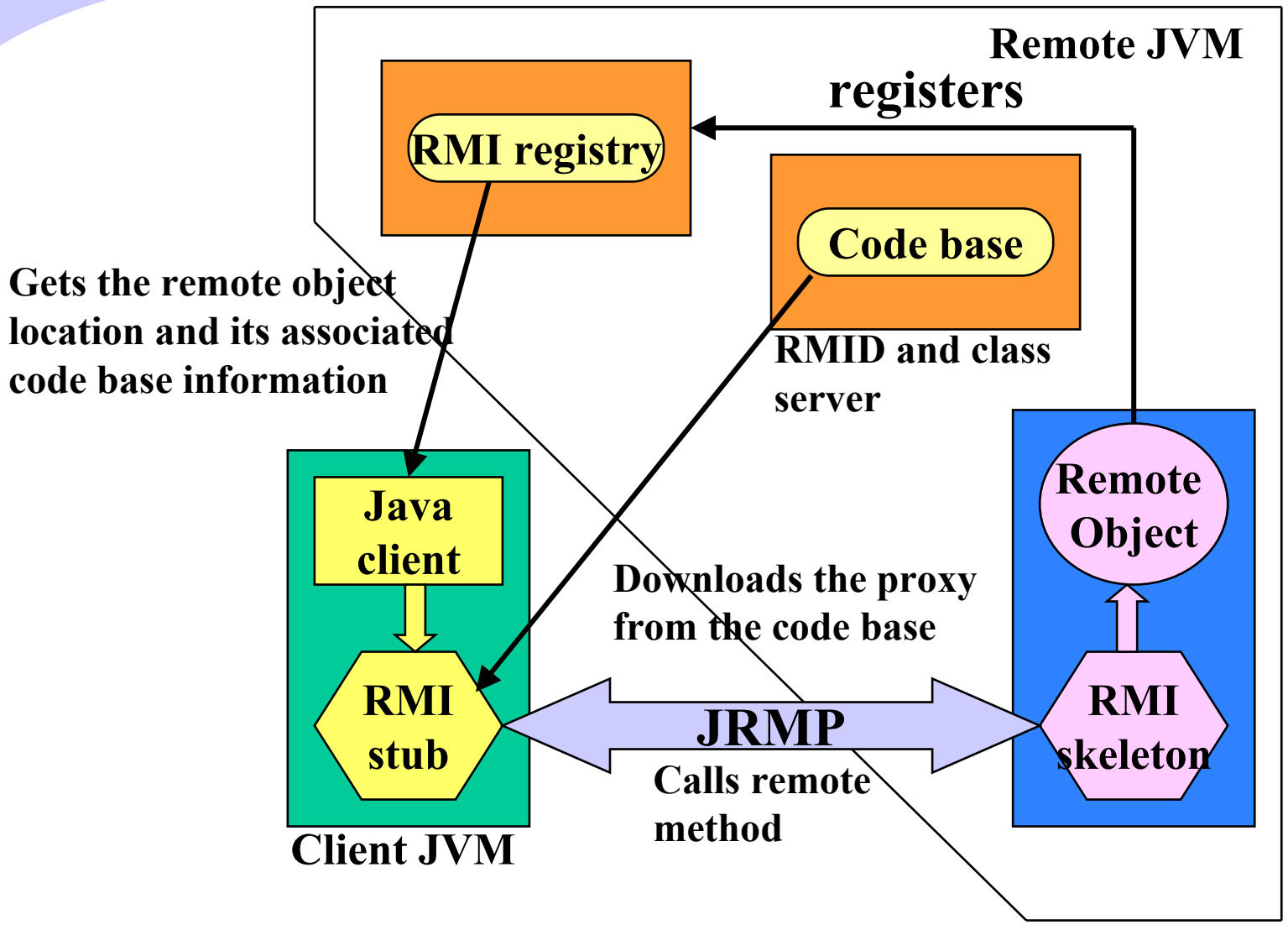
RMI Architecture



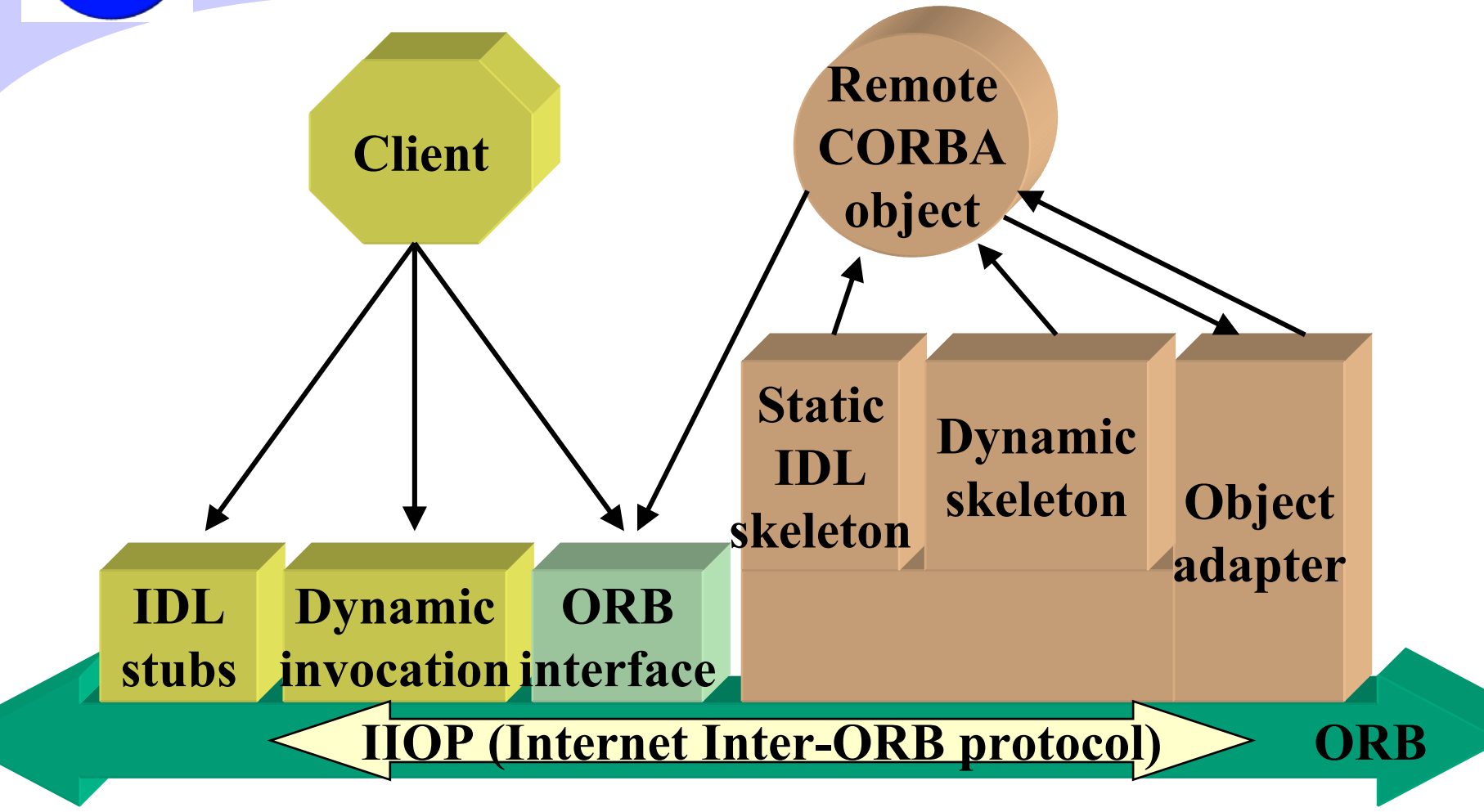
Three layers of the RMI architecture

- Stub/skeleton layer:
  - See its role in previous slides.
- Remote reference layer: Allows different invocation strategies to be implemented as a protocol both on the client and server side. Examples:
  - point-to-point
  - replication
  - persistent reference
- Java Remote Method Protocol (JRMP) Transport layer
  Manages a connection including:
  - listening
  - setting up
  - monitoring
  - transmitting an invocation

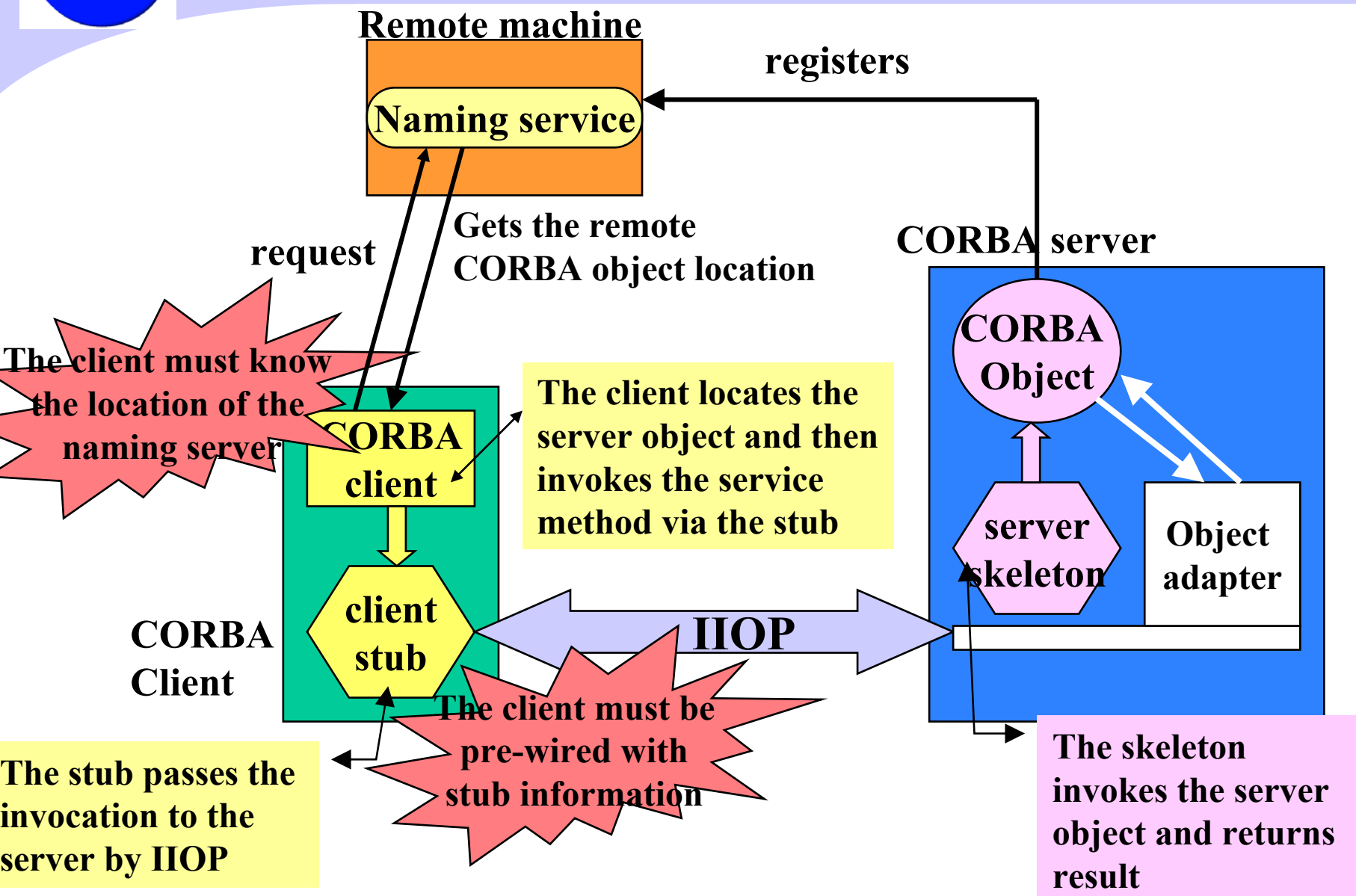
How Java RMI works



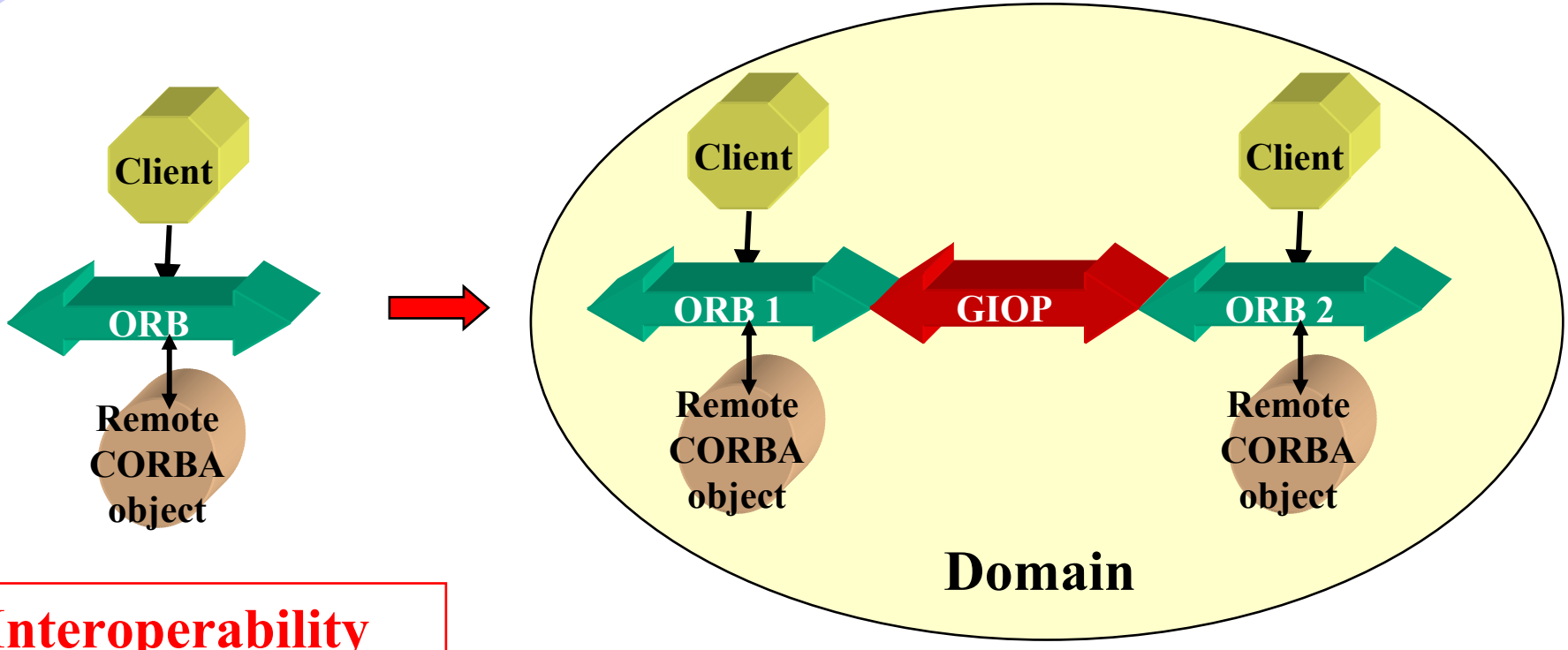
CORBA architecture



How CORBA works



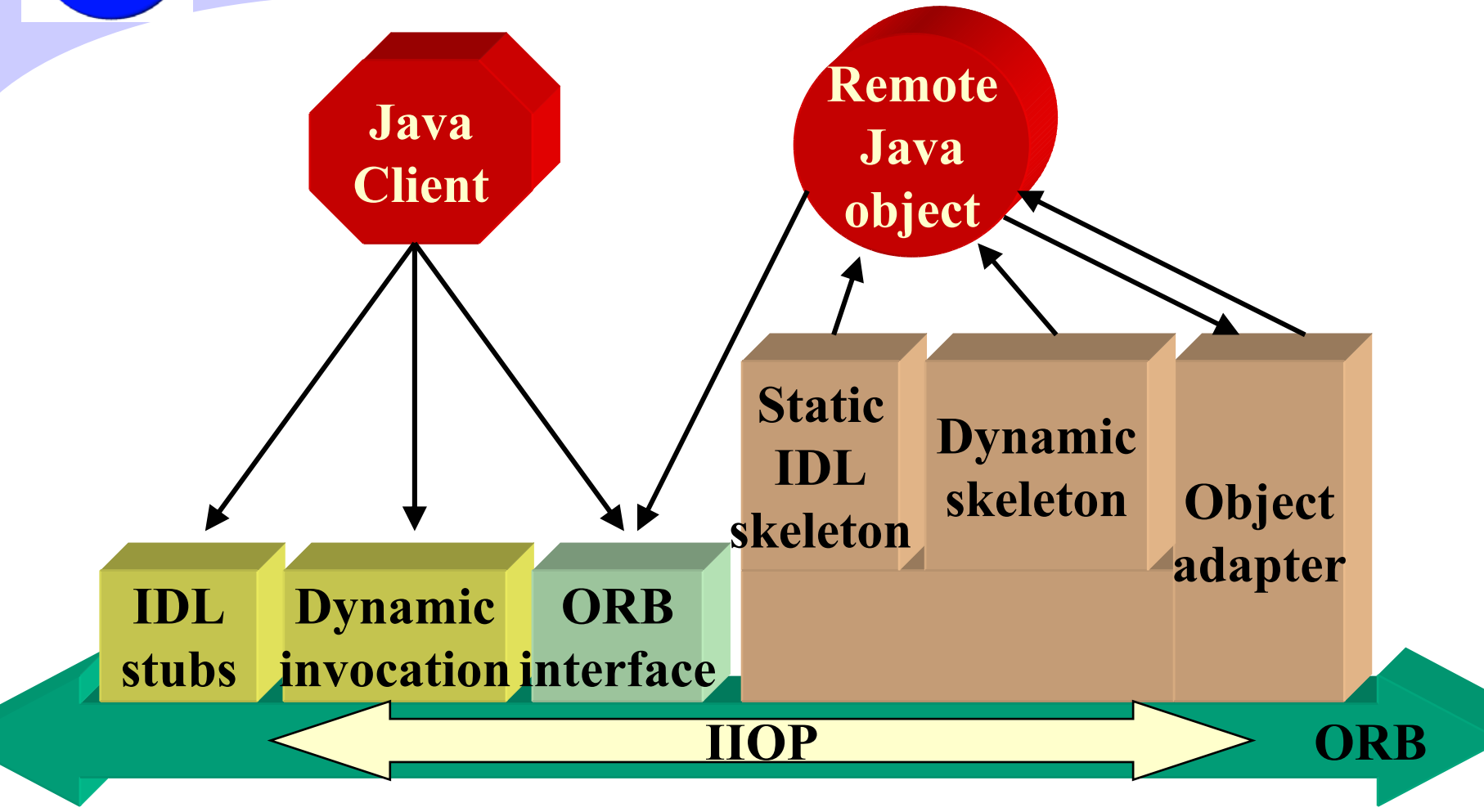
Scalability in CORBA



Interoperability among clients and service objects inside ORB

Interoperability among ORBs based on GIOP (General Inter-ORB Protocol)

Java/CORBA Architecture



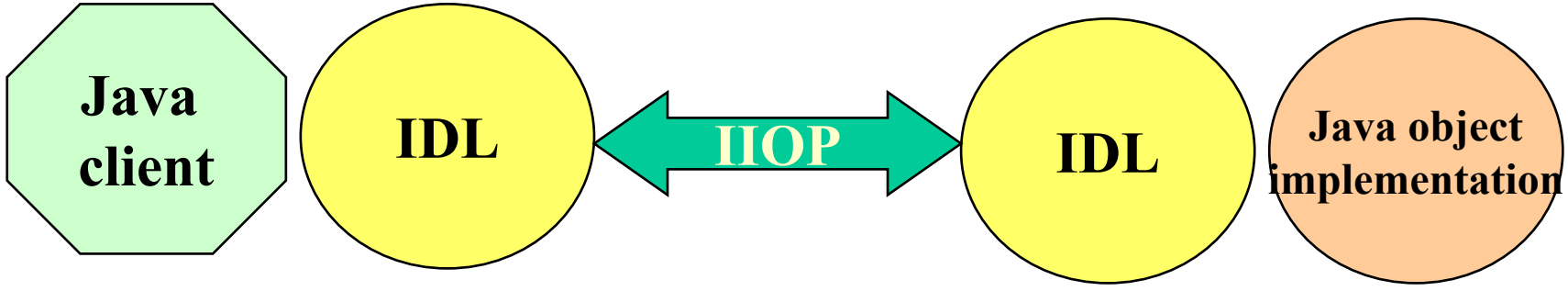
Java/CORBA Architecture - Approach 1

- Step 1. The developer writes an IDL for interface specification.
- Step 2. A Java IDL compiler is used to generate the Java code.
- Step 3. The IDL generates CORBA stubs and skeletons.
- Advantages:
  - All of the CORBA services are available to the developer
  - This approach is the most heterogeneous
- Disadvantage:
  - It requires the developer to work with the CORBA IDL.

Java/CORBA Architecture - Approach 2

- Step 1. The developer writes the Java code and interfaces.
- Step 2. Applying Java IDL, the Java interfaces are used to generate CORBA IDL interfaces.
- Advantages:
  - Does not require the Java developer to have any CORBA IDL knowledge
- Disadvantage:
  - It generates less efficient interfaces than approach 1.

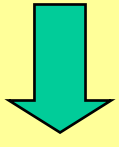
CORBA-based Java



- Advantage:
  - IIOP complements the native RMI approach by addressing all of its shortcomings:
    - Improves support for interoperability with other languages
    - Platform independent and vendor neutrality
    - CORBA support
- Disadvantage:
  - More difficult to use than the native RMI approach

Differences between RMI and CORBA

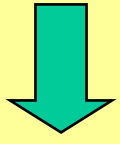
- The transport protocol in RMI is the JRMP, in CORBA it is IIOP.
- In RMI the client need not be prewired with the server object information
  - The client proxy/stub is downloaded at run-time from a specified server's code base
  - The stub knows where to find the remote object



RMI is closer to the GRID solution in this respect

Differences between RMI and CORBA

- RMI is tightly connected with Java.
- CORBA can connect any legacy code written in any language



CORBA is closer to the GRID solution in this respect

Thank you