Distributed Heterogeneous Applications

CAT3053/N Distributed Computing

Heterogeneity

- Started with the assumption we have a heterogeneous distributed system.
- But the dimensions of heterogeneity are many.

Heterogeneity

- Hardware-Is not just what type of computer available but also are there any embedded devices involved.
- Different operating system
- Programming language
- Standards/protocol/API
- Access mechanism
- Devices
- Different version.

Heterogeneity in Distributed Systems

- Distributed applications are typically heterogeneous:
  - different hardware: mainframes, workstations, PCs, servers, etc.;
  - different software: UNIX, MS Windows, IBM OS/2, Real-time OSs, etc.;
  - unconventional devices: teller machines, telephone switches, robots, manufacturing systems, etc.;
  - diverse networks and protocols: Ethernet, FDDI, ATM, TCP/IP, Novell Netware, etc.

Why?

- Different hardware/software solutions are considered to be optimal for different parts of the system.
- Different users which have to interact are deciding for different hardware/software solutions/vendors.
- Legacy systems

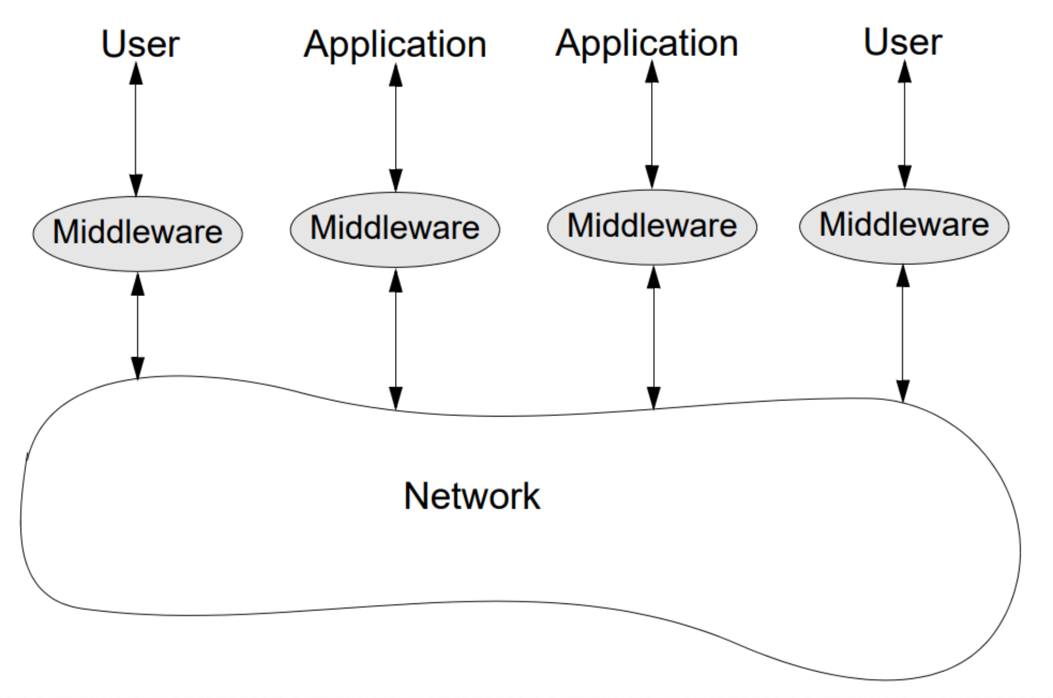
Implication of Heterogeneity

- How are you going to deal with a question of heterogeneity? Some approaches
- Standard APIs – Backward and forward compatibility
- Normative architectures-Model Driven Architecture (MDA)
- Vendor comprehensive solution

Middleware

- A key component of a heterogeneous distributed client-server environment is middleware.
- Middleware is a set of services that enable applications and end users to interact with each other across a heterogeneous distributed system. Middleware software resides above the network and below the application software.

Middleware


User
Application
Application
User
Middleware
Middleware
Middleware
Middleware
Network

Middleware

- Middleware is a general term for software that serves to "glue together" separate, often complex and already existing, programs.
- Middleware makes it easier for software developers to perform communication and input/output, so they can focus on the specific purpose of their application.
- Middleware is the software that connects software components or enterprise applications and it lies between the operating system and the applications on each side of a distributed computer network

Middleware

- Middleware should make the network transparent to the applications and end users
  - users and applications should be able to perform the same operations across the network that they can perform locally.
- Middleware should hide the details of computing hardware, OS, software components across networks.
- Different kind of software qualifies, to certain extent, as middleware:
  - File-transfer packages (FTP) and email;
  - Web browsers;
  - CORBA

Heterogeneous Computing

- Heterogeneous computing in heterogeneous system architecture.
- Computer use CPU to process application step by step.
- Computer use GPU-great for graphic processing and handles large block of data done in parallel

Heterogeneous Computing

- Refers to system that use more than one kind of processor.
- Often is the combination of the CPU and GPU.
- Heterogeneous System architecture. (HSA)

Importance of Heterogeneous Computing

- Is important and has many benefits over traditional homogenous computing in which the GPU and CPU share the same chip often result in
- Better battery life
- Lower cost
- Accelerated Processing Unit (APU)

CPU

- CPU design has become more complex over time to meet the high demands from users.
- There are 3 distinct PC eras
- Single Core era – all CPU were originally single core and was constrained by power and complexity.
- Multi core Era – multiple CPU is on the chip any by the end of 2010, all new desktop use multi core processor.
- Heterogeneous system era

Real world Application

- Intel CPU with HD graphics
- AMD APUs
- Smartphone's
- Laptops
- PlayStation Series
- Xbox Series

GPU

- Originally designed for games and graphic applications.
- Specialize in many small tasks in parallel.
- Millions of calculation per seconds
- GPU processor can be used for typical CPU Computations with HSA.
- Improve performance in cryptography, gesture recognition and so much more

Heterogeneous System Architecture (HSA)

- Heterogeneous System Architecture (HSA) is an open standards technology
- HSA intends to make it easier for the GPU and CPU to interact
- HSA foundation helps standardize the heterogeneous computing industry.
- HSA foundation created HSA Intermediate language.

Programming for HSA

- New Software design is needed for full advantages of HAS
- Traditional software design will run on it however it will not take full advantages of the GPU for certain CPU task.
- OpenCL
- Cuda
- hUMA

OpenCL

- Originally Apple but now open standard and royalty free.
- Open standard for heterogeneous computing
- Allow existing source code to target CPU or GPU
- Programmer adapts algorithm to take advantage of CPU or GPU.
- Cross platform.
- C99 based

CUDA

- Nvidia propriety and heterogeneous computing software platform.
- Parallel computing platform and programming model.
- Platform Contains: Libraries, compiler directives, C, C++ and Fortran extensions.
- Free
- Limited to only NVIDIA hardware

hUMA

- AMD propriety Heterogeneous Uniform Memory Access
- Shared Memory Architecture used in APUs in traditional system architecture separate GPU and CPU each has their own pool of memory.
- Same Memory and cache
- Result in support for mainstream languages such as Phyton, C++, and Java

hUMA

- To do computation on GPU the data has to be copied from the system memory to gpu memory and when work is done copied back to the system this add the total time and negate the value of doing computation on the GPU
- In most HSA architecture the implementation of GPU and CPU have separate slices of the system memory to work with gpu can access the same memory as a cpu however GPU and CPU cant work together on the same data still need to be copied over. hUMA fix this CPU and GPU share one piece of memory together.

Pros

- Small and efficient processor
- Better battery life
- Lower cost
- Increase performance of parallel application

Cons

- A discrete CPU and GPU can provide more power.
- New Software design models

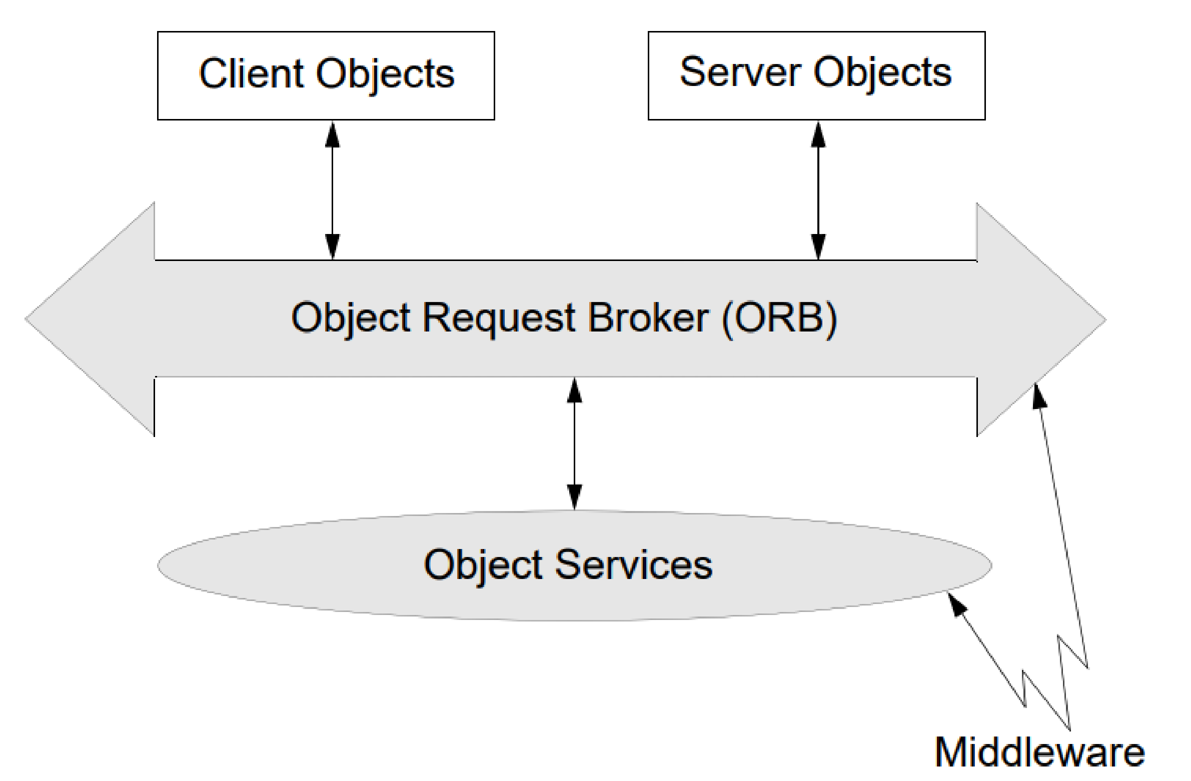
Future

- “GPUs has evolved to the point where many real world application are easily implemented on them to run significantly faster than on multicore systems. Future Computing architectures will be hybrid systems with parallel core GPUs working in tandem with multi core CPUs.” – Jack Dongarra, Professor at University of Tennessee

Entering Heterogeneous System Era

- HSA – the design and software implementation for different types of processor on a single chip.
- CPU is great for sequential tasks.
- GPU is great for small tasks in parallel.
- OpenCL, CUDA and hUMA help programming for HSA easier
- Can improve performance, lower cost, and improve battery life.
- Heterogeneous Computing is the future.

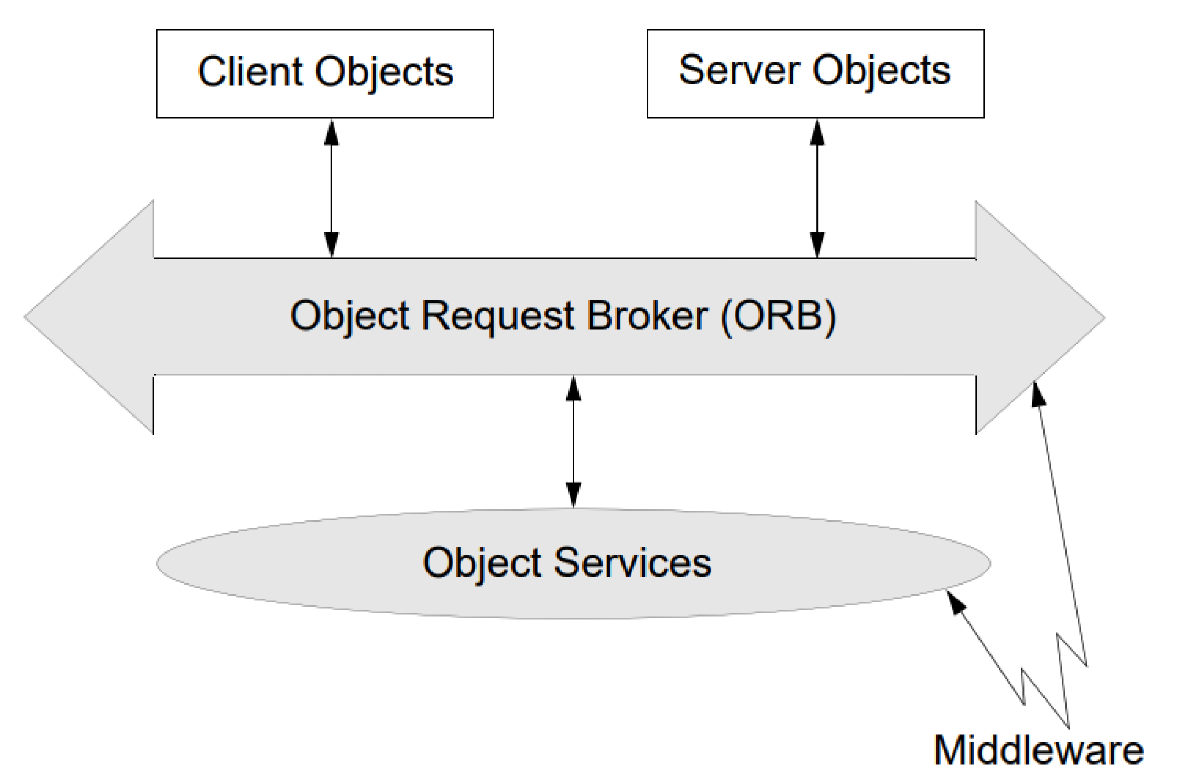
Objects in Distributed Systems



A distributed application can be viewed as a collection of objects

- user interfaces,
- databases,
- application modules,
- customers.

Objects in Distributed Systems

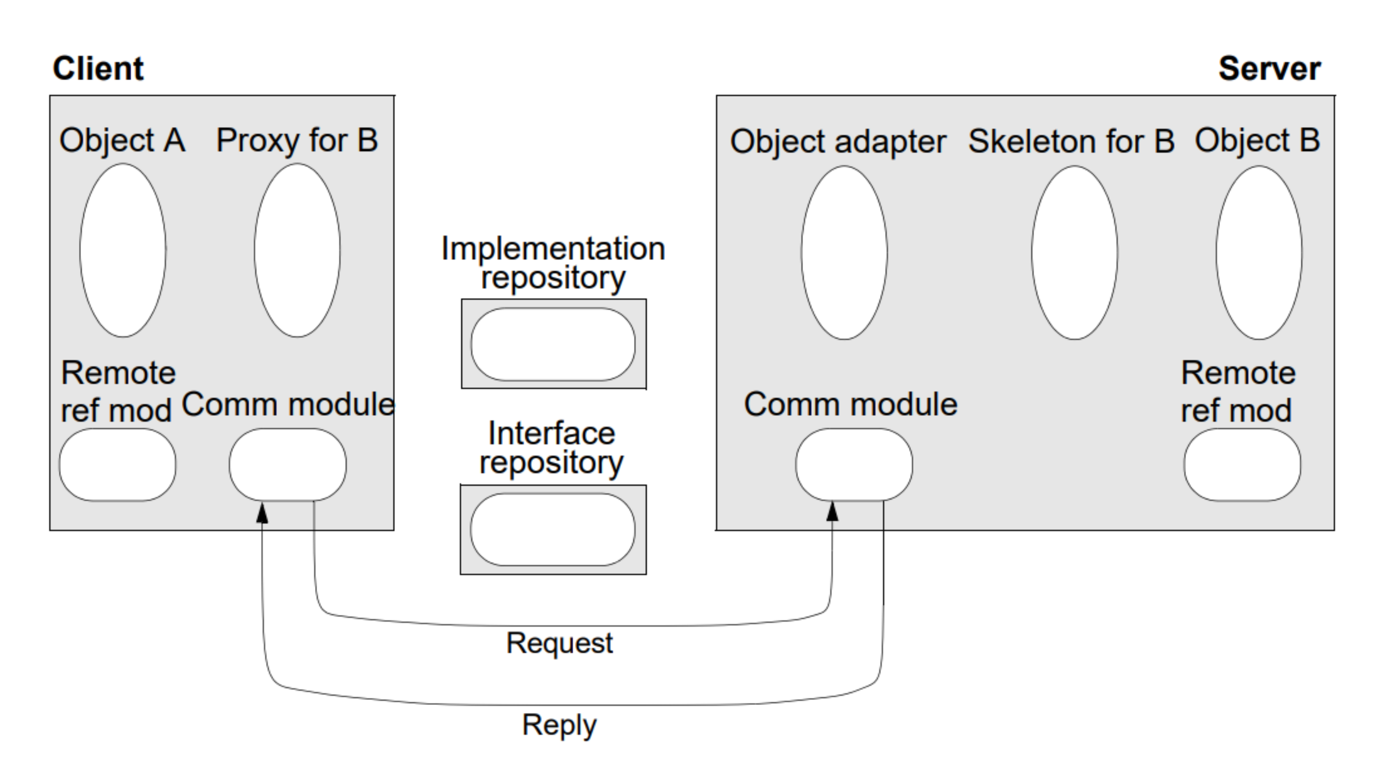


Object: data surrounded by code; has its own attributes and methods which define the behavior of the object; objects can be clients, servers, or both.

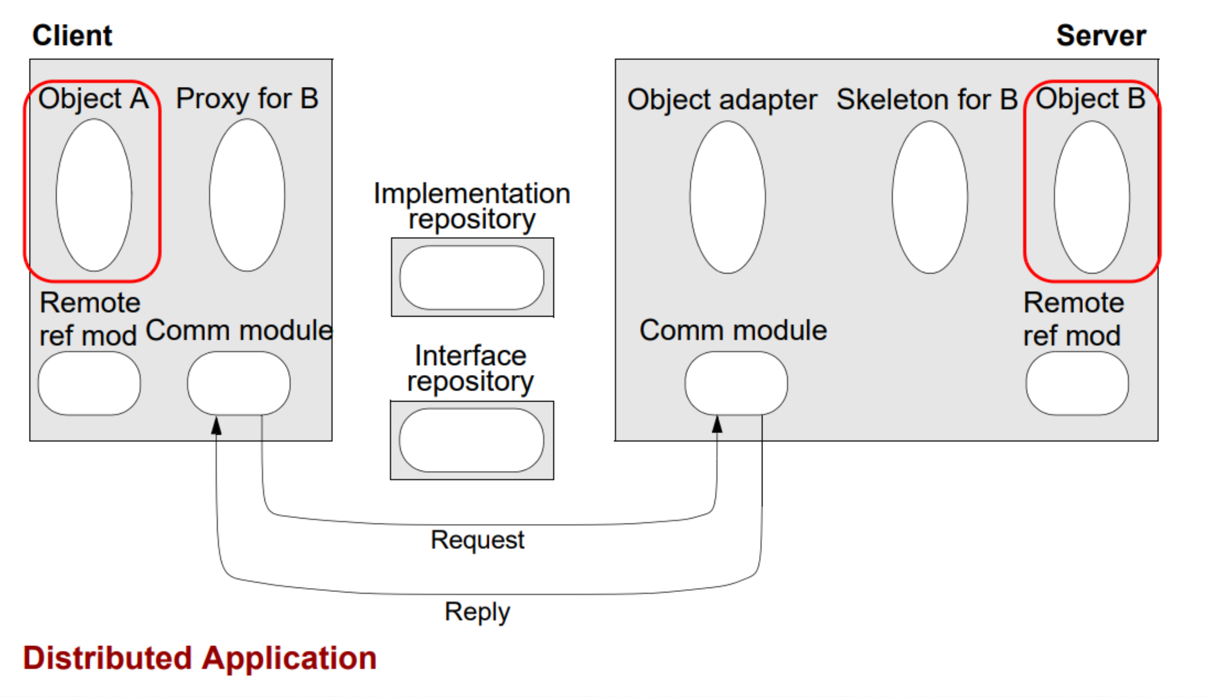
Object broker: allows objects to find each other and interact over a network; they are the backbone of the distributed system.

Object services: allow to create, name, move, copy, store, delete, restore, and manage objects.

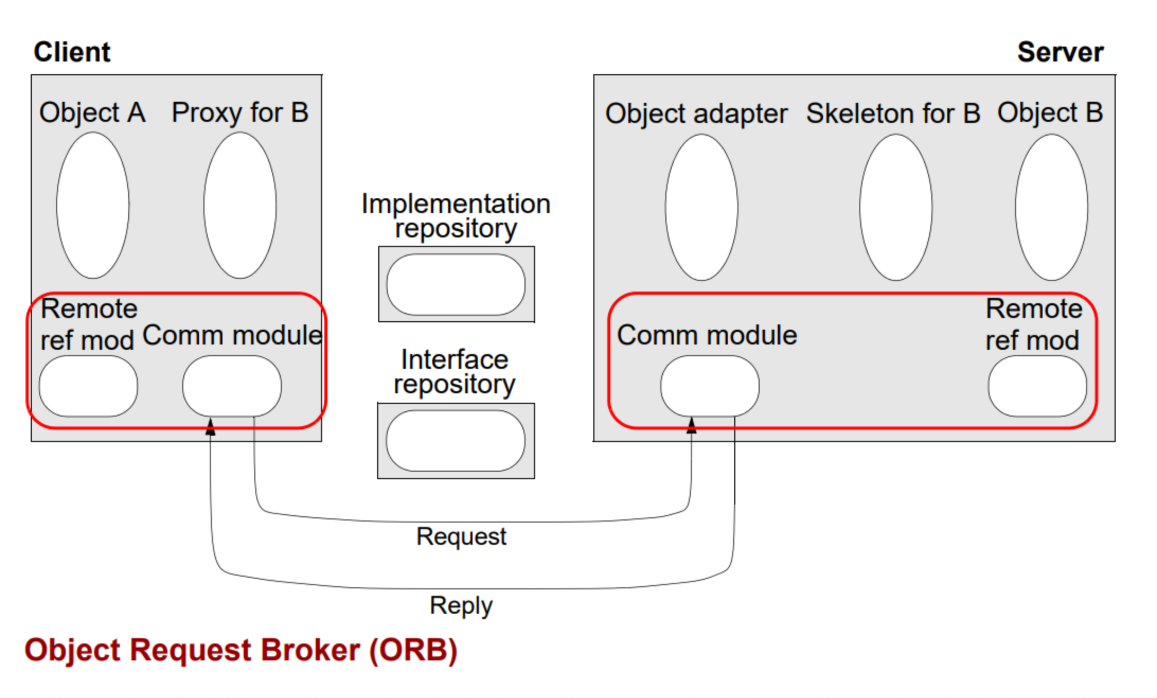
RMI: Objects in Distributed Systems



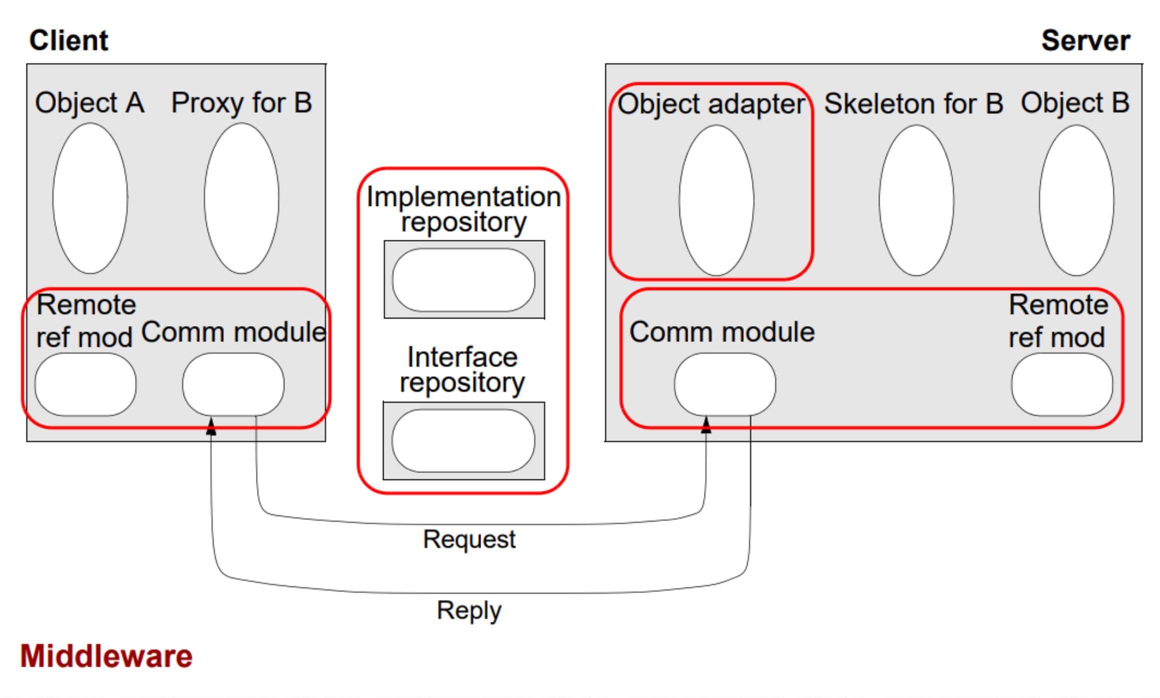
RMI: Objects in Distributed Systems



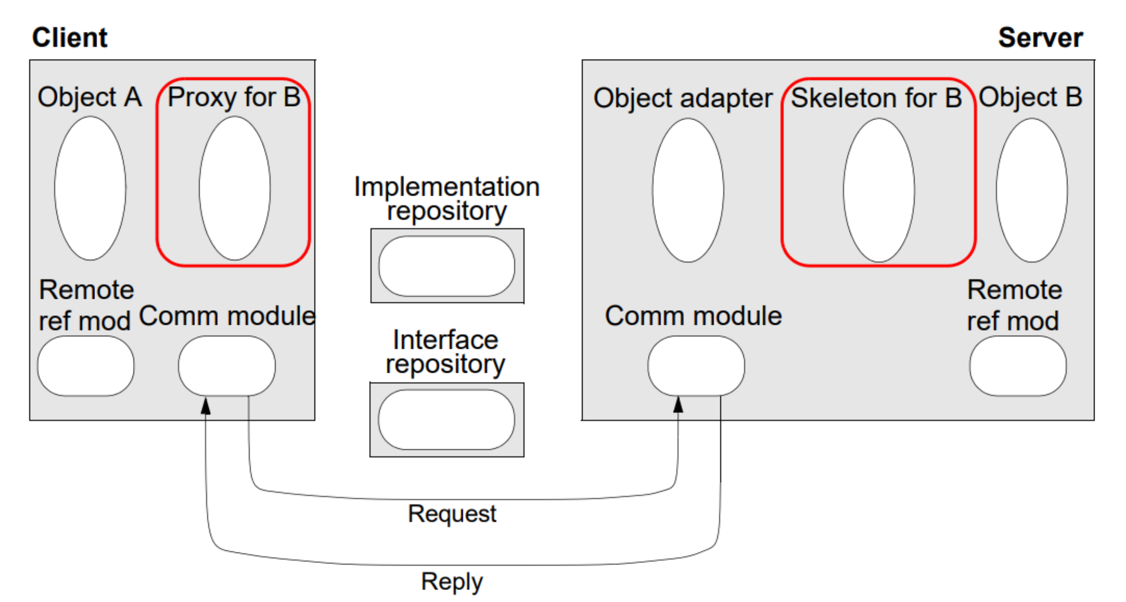
RMI: Objects in Distributed Systems



RMI: Objects in Distributed Systems



RMI: Objects in Distributed Systems



Border between Application and Middleware