Java Remote Method Invocation (RMI)

Object Request Brokers (ORB)

- Providing a framework where remote objects can be consumed over the network in the same way as consuming local objects
  - Providing functionalities for marshalling and unmarshalling when objects are transmitted between clients and servers
  - Hiding implementation details, which enables developers to implement distributed applications efficiently without facing the complex underlying operating systems and network communications

Overview of Java RMI

- Writing distributed objects using Java
- Simple and direct model for distributed computation with Java objects
- Centered around Java, thus bringing the power of Java safety and portability to distributed computing
- Behavior can be moved
- Connected to existing and legacy systems via JNI; connected to relational DB via JDBC

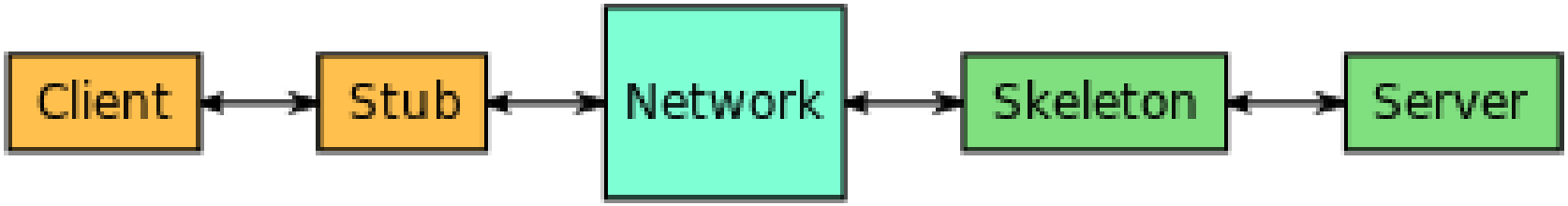
Advantages

- Object oriented
  - Full objects as arguments and return values
- Mobile behavior
  - Class implementations can be moved
- Design patterns
  - Passing objects ensures full power of OO tech
- Safe and secure
  - Built-in Java security mechanisms: security manager

Advantages (cont.)

- Easy to write/use/maintain
- Connects to existing/legacy systems
  - RMI/JNI, RMI/JDBC
- Write once, run anywhere
  - 100% portable to any JVM
- Distributed garbage collection
- Parallel computing
  - Multi-threaded, concurrent processing

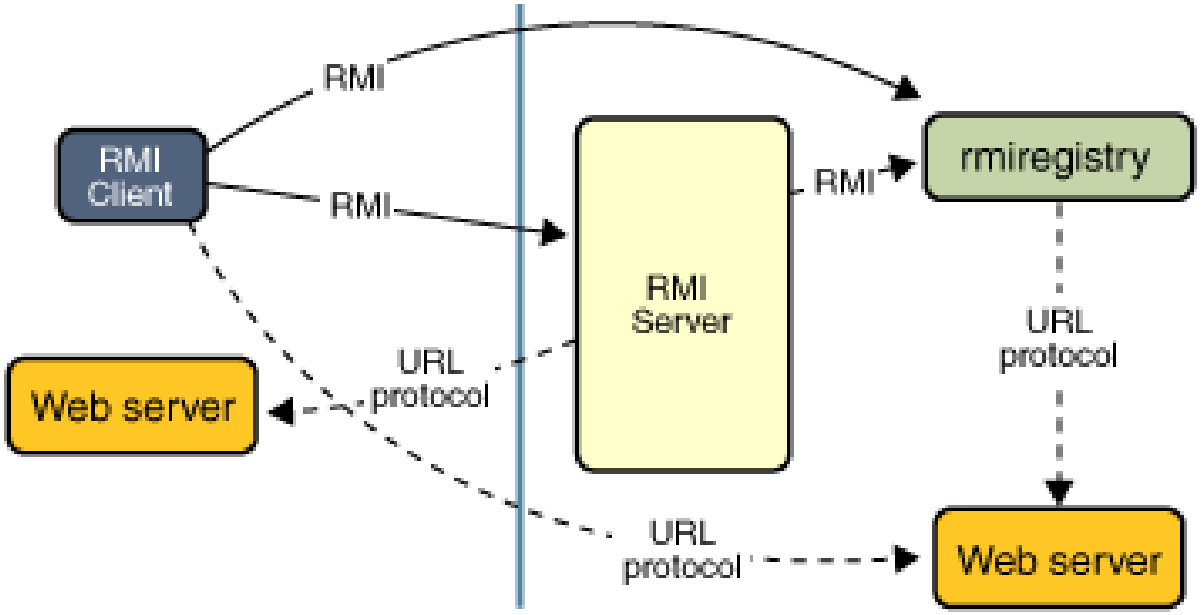
Architecture



- Stub
  - Marshaling arguments, sending invocation
- Skeleton
  - Unmarshaling arguments, invoking server's implementation

RMI Distributed Application

- Locate remote objects
- Communicate with remote objects
- Load class definitions for objects that are passed around



Remote Interfaces, Objects and Methods

- An object becomes remote by implementing a remote interface with following characteristics
  - A remote interface extends the interface *java.rmi.Remote*
  - Each method of the interface declares *java.rmi.RemoteException* in its throws clause

Major Steps

- Defining the remote interfaces
  - Specifying the methods that can be invoked remotely
- Implementing the remote objects
  - May include implementations of interfaces/methods that are local
- Implementing the clients
  - Implemented at any time after remote interfaces are defined

Lab

- Using Java RMI to implement distributed applications

Programming Practice

- Develop a light-weight geographic information library system
  - Server maintains a geographic information library including lists of countries and cities
  - Services provided by the server
    - Add/delete a country
    - Add/delete a city for a specified country
    - Retrieve the list of countries
    - Retrieve the list of cities for a specified country
  - The client application supports console-based UI