

JAVA RMI

103

103

Java Remote Method Invocation (RMI)

- Distributed garbage collection
- Local vs remote objects
 - Local objects passed *by value*
- Dynamic stub downloading
- Remote exceptions

104

104

Defining Remote Interfaces

```
import java.rmi.*;
public interface IServer extends Remote {
    public void cd (String filename)
        throws RemoteException;
    public String[] dir () throws RemoteException;
}
```

105

105

Defining a Remote Implementation

```
import java.rmi.*;
public class Server extends UnicastRemoteObject
    implements IServer {
    private Stack<String> cwd;
    private String path() { ... }
    public cd (String dir) throws RemoteException {
        cwd.push(dir);
    }
    public String[] dir () throws RemoteException {
        return new File(path()).list();
    }
}
```

106

106

RMI Registry

Flat non-persistent namespace

```
interface Registry extends Remote {  
    Remote lookup (String name);  
    void bind (String name, Remote obj);  
}  
class LocateRegistry extends Object {  
    static Registry getRegistry (String host, int port);  
    static Registry createRegistry (int port);  
}
```

107

107

Factory Objects

```
import java.rmi.*;  
interface IServerFactory extends Remote {  
    public IServer createServer () throws RemoteException;  
}  
  
class ServerFactory extends UnicastRemoteObject  
    implements IServerFactory {  
    public IServer createServer () throws RemoteException {  
        return new Server (...);  
    }  
}
```

108

108

Example Server

```
import java.rmi.*;
class Server {
    public static void main (String[] args) {
        System.setSecurityManager
            (new RMISecurityManager());
        ServerFactory stub = new ServerFactory (...);
        Registry registry =
            LocateRegistry.createRegistry(serverPort);
        registry.rebind(serverName, stub);
    }
}
```

109

109

Example Client

```
import java.rmi.*;
class Client {
    public static void main (String[] args) {
        System.setSecurityManager
            (new RMISecurityManager ());
        IServerFactory factory;
        // Look up factory in registry
        IServer server = factory.createServer();
    }
}
```

110

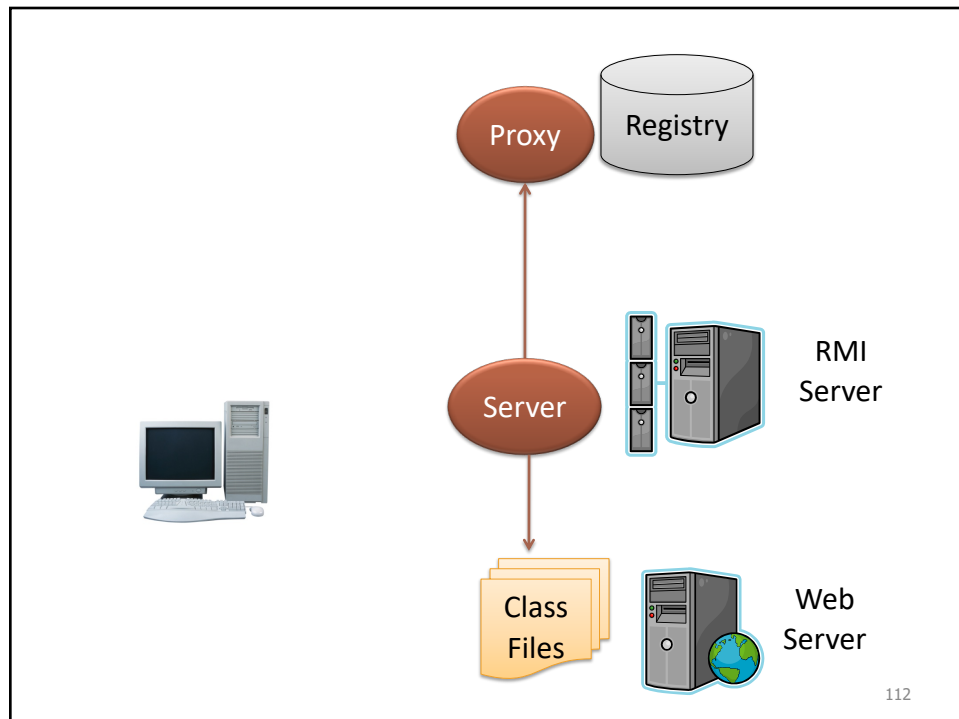
110

Putting it all together

- Compile the client and server
`javac Client Server`
- Run the registry (on guinness)
`rmiregistry 5000 &`
- Run server on server host, client anywhere
`java Server &`
`java Client`

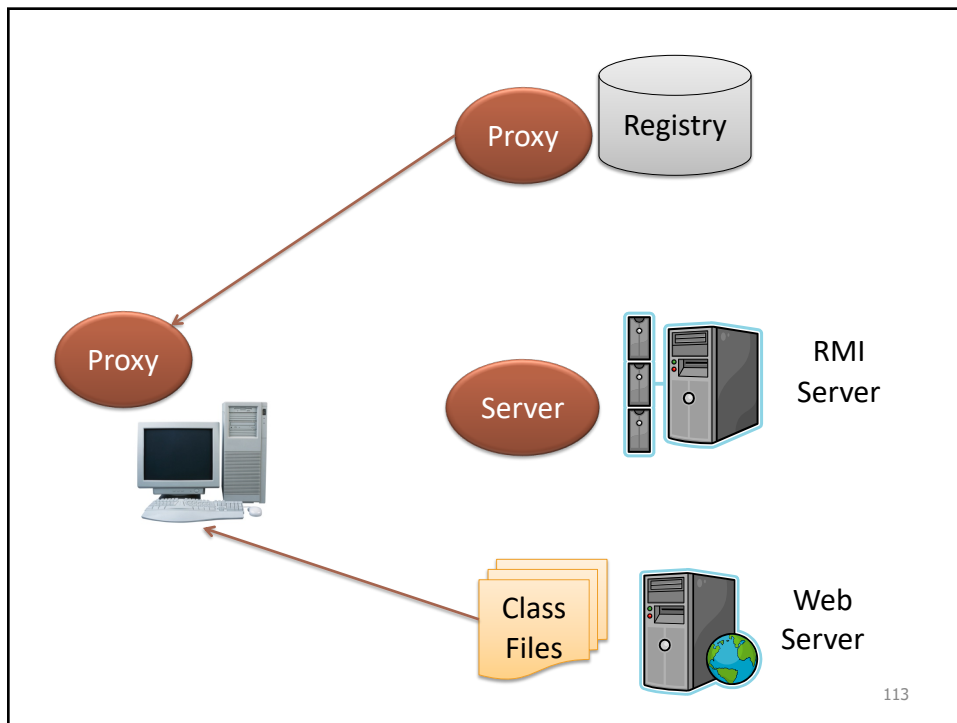
111

111



112

112



113

Dynamic Stub Loading

- URL for missing class codebase specified:
 - Dynamically in object reference (URL)
 - Statically in `java.rmi.server:codebase` property
- Disable downloading by setting
 - `java.rmi.server:useCodebaseOnly` to true
- `RMIClassLoader` loads stubs across network
- Security manager must be installed

114

114