



X36DSV – 2. practice

RMI

Remote Method Invocation

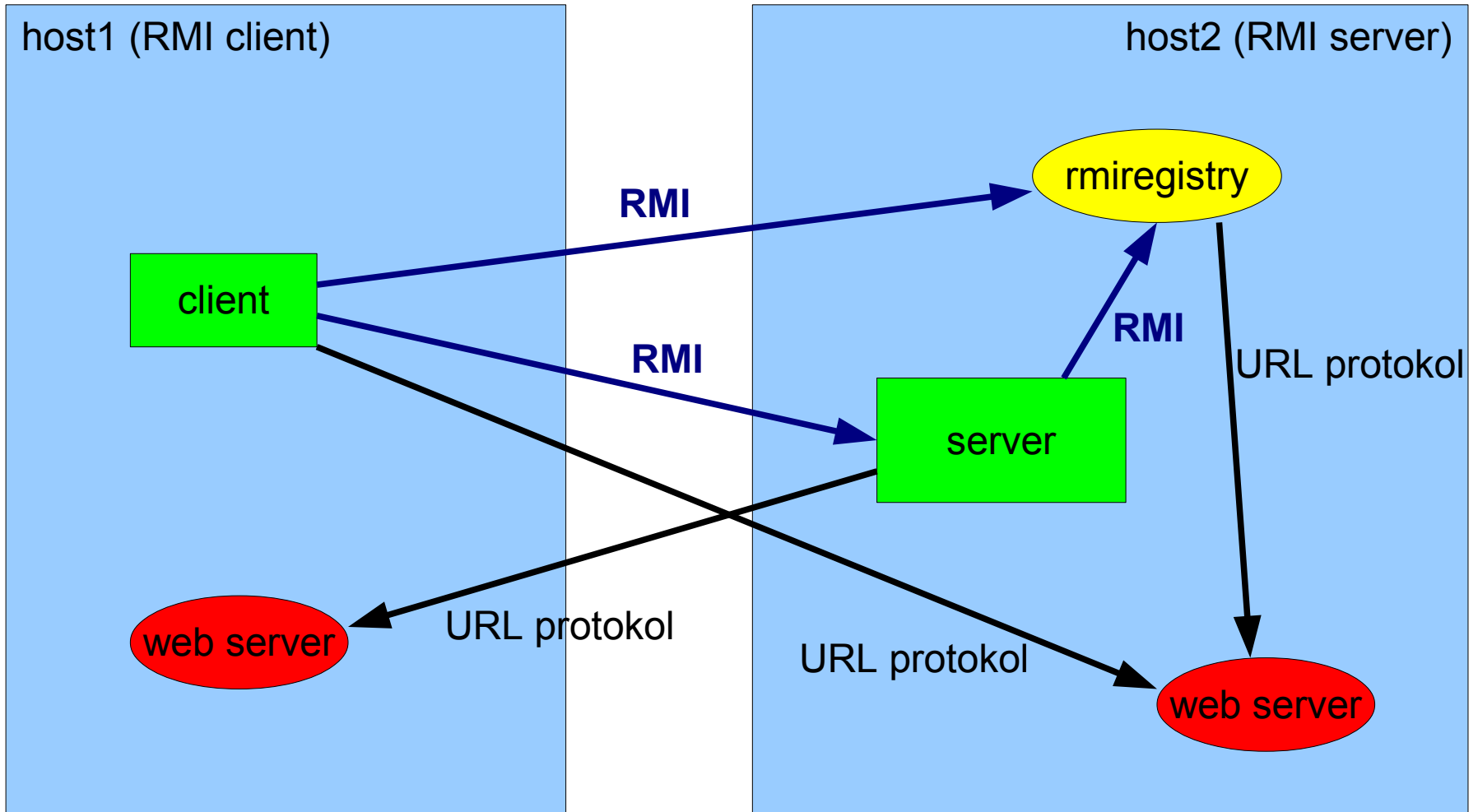


RMI – what it is?

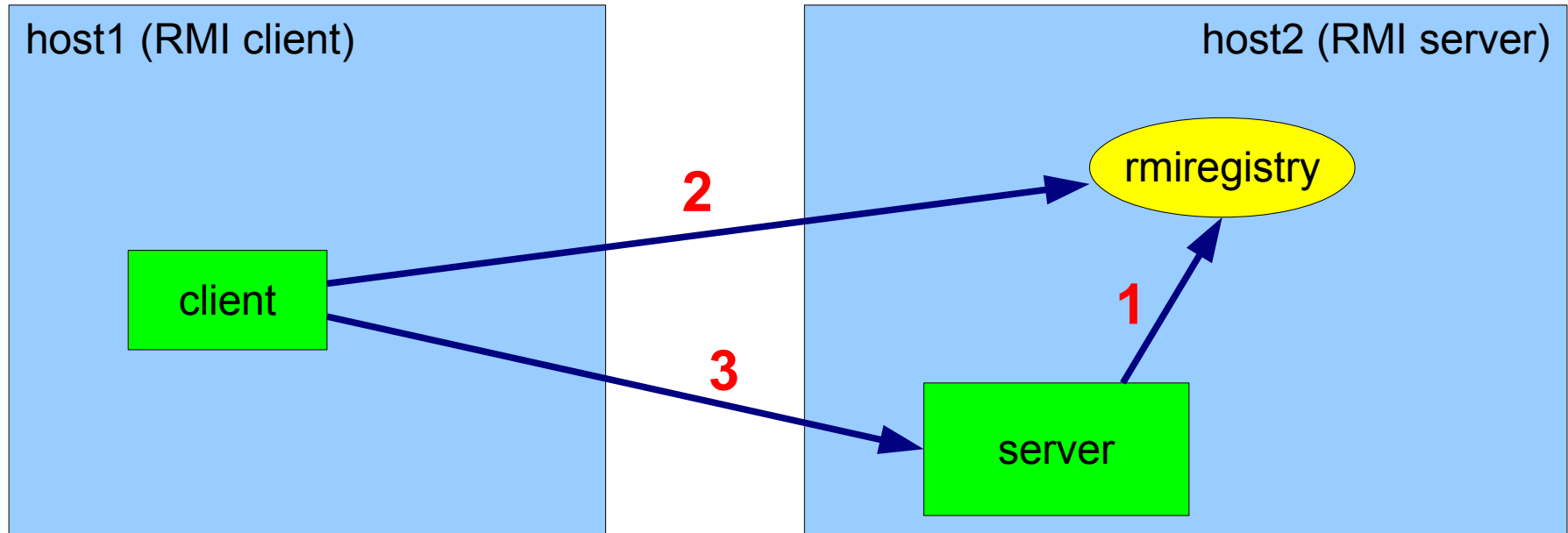
- calling methods from another JVM
 - localization of remote object
 - communication between objects
 - transfer bytecode of unknown classes
- typically client – server
- <http://java.sun.com/docs/books/tutorial/rmi/index.html>



RMI application



RMI aplikace



- 1 – object registration (binding object and name) - `Registry.(re)bind`
- 2 – get object by given name - `Registry.lookup`
- 3 – using remote object - `Remote_object.remote_method`



Remote

- remote methods
 - methods called between JVM
- remote interface
 - describes remote methods
 - extends interface *java.rmi.Remote*
 - all remote methods must throw *java.rmi.RemoteException*



Remote

- remote objects
 - objects with remote methods
- remote stub
 - acts as local object (proxy)
 - have all methods from remote interface
 - handles marshaling, unmarshaling



Design sequence

- design remote interface
- remote object implementation
- Client implementation
- compile
- distribution of jars
- start



RMI - Interface

```
package compute;  
  
import java.rmi.Remote;  
import java.rmi.RemoteException;  
  
public interface MathServer extends Remote {  
    public int secti(int a, int b) throws RemoteException;  
}
```




RMI – Remote object

```
package server;

import java.rmi.*;
import compute.*;

public class MathServerImpl implements MathServer {

    // pro potrebu serializace (marshalingu)
    private static final long serialVersionUID = -386L;

    public MathServerImpl() throws RemoteException {
        super();
    }

    public int secti(int a, int b) throws RemoteException {
        int result;
        result=a+b;
        System.out.println(a + " + " + b + " = " + result);
        return result;
    }
}
```



RMI - Server

```
package server;

import java.rmi.*;
import compute.*;

public class Server {
    public static void main(String[] args) {
        if (System.getSecurityManager() == null)
            System.setSecurityManager(new RMISecurityManager());
        String name = "MathD";
        try {
            MathServer msi = new MathServerImpl();
            MathServer stub =
                (MathServer) UnicastRemoteObject.exportObject(msi);
            Registry registry = LocateRegistry.createRegistry(2010);
            registry.rebind(name, stub);
        }
        catch (Exception e) {
            System.err.println("Data exception: " + e.getMessage());
        }
    }
}
```



RMI - Client

```
package client;

import java.rmi.*;
import compute.*;

public class Client {
    public static void main(String args[]) {
        if (System.getSecurityManager() == null)
            System.setSecurityManager(new RMISecurityManager());
        try {
            MathServer mth;
            String name = "MathD";
            Registry registry = LocateRegistry.getRegistry("localhost",
2010);

            mth = (MathServer) registry.lookup(name);
            int a = Integer.valueOf(args[0]).intValue();
            int b = Integer.valueOf(args[1]).intValue();
            System.out.println(a + " + " + b + " = " + mth.secti(a, b));
        }
        catch (Exception e) {
            System.err.println("Data exception: " + e.getMessage());
        }
    }
}
```



Compile and run

```
# compilation (Win i Lin)
javac compute/MathServer.java
javac server/MathServerImpl.java server/Server.java
javac client/Client.java

# standalone RMI registry
start rmiregistry
rmiregistry &

# server start (pod Lin ';' -> ':')
java -cp compute.jar;server.jar
    -Djava.security.policy=java.policy server.Server

# client start (pod Lin ';' -> ':')
java -cp compute.jar;client.jar
    -Djava.security.policy=java.policy client.Client 64 46
```



Odkazy

- Oracle RMI tutorial
 - <http://download.oracle.com/javase/tutorial/rmi/index.html>
- more about codebase
 - <http://www.kedwards.com/jini/codebase.html>