



X36DSV – 2. cvičení

RMI

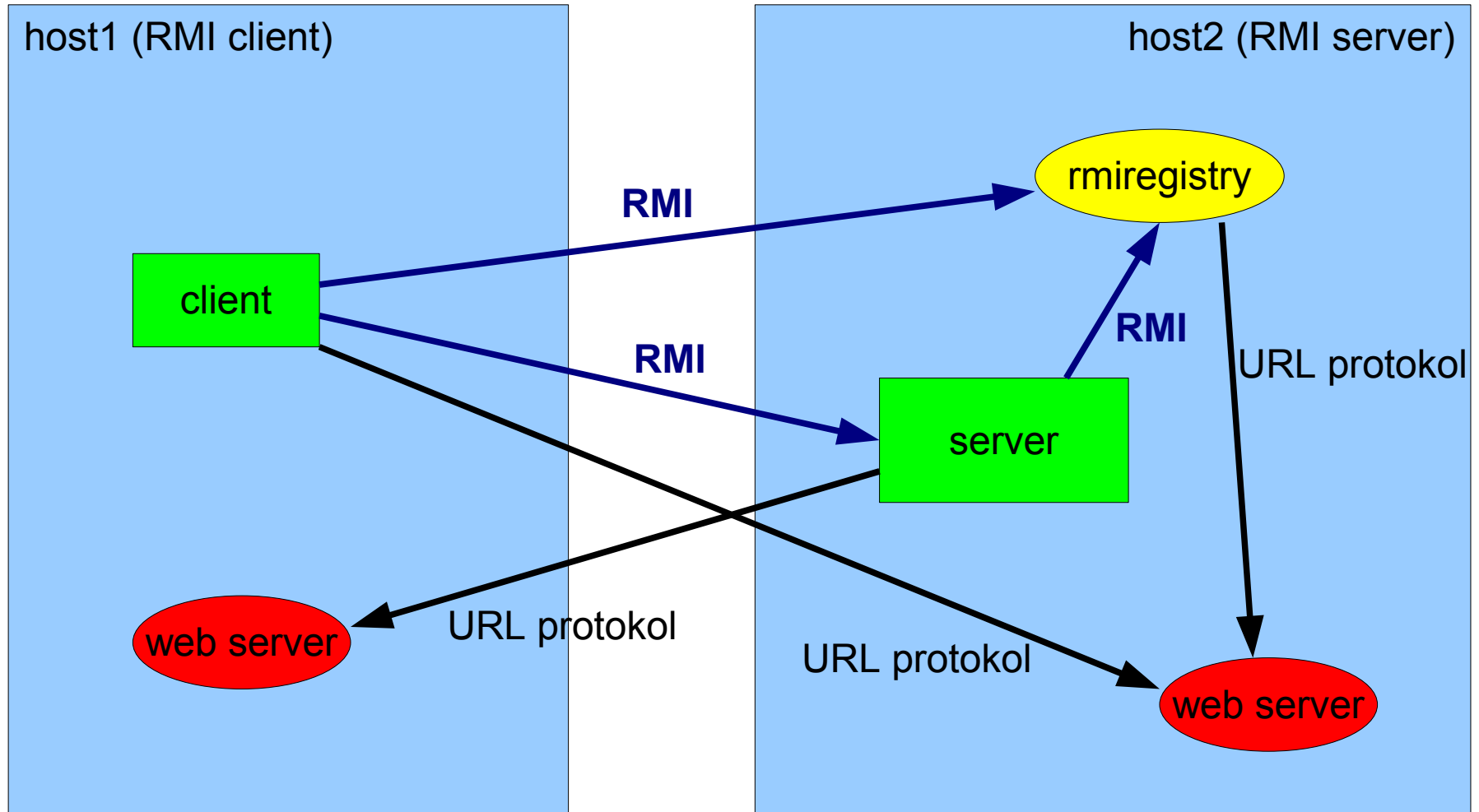
Remote Method Invocation



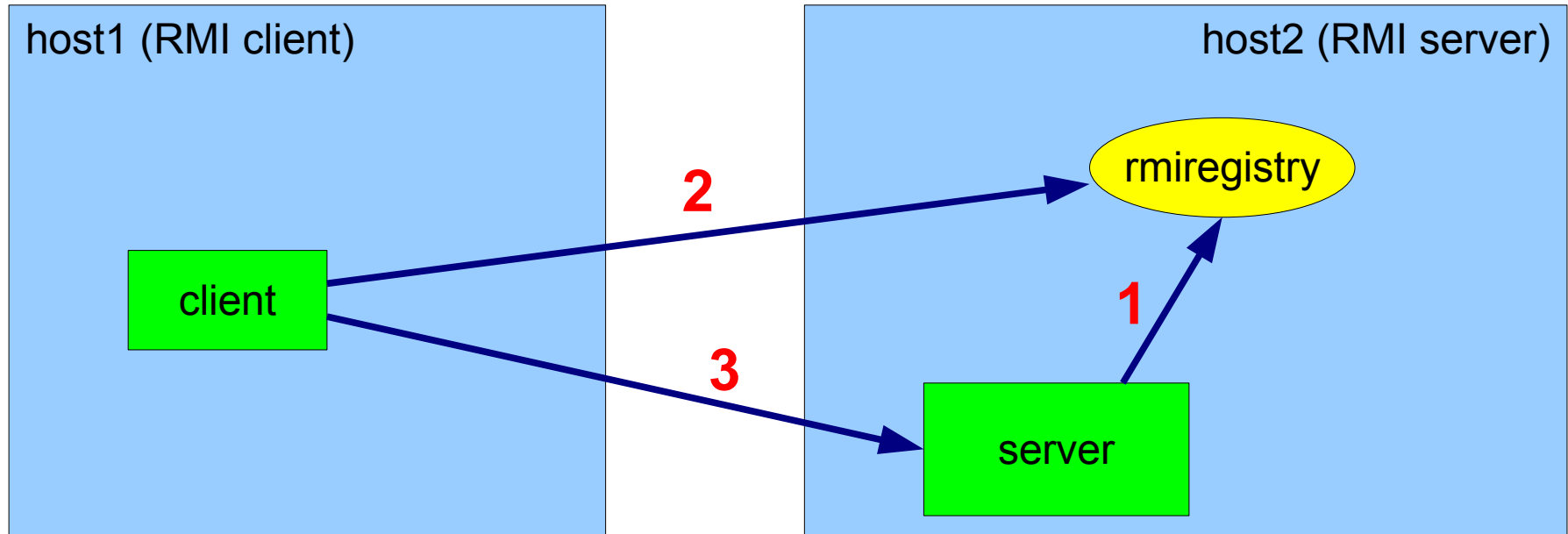
RMI – co to je?

- vyvolání metody z jiné JVM
 - lokalizace vzdáleného objektu
 - komunikace se vzdálenými objekty
 - přenos objektu v bytecode
- typicky klient – server
- <http://java.sun.com/docs/books/tutorial/rmi/index.html>

RMI aplikace



RMI aplikace



- 1 – registrace objektu (svázání objektu se jménem) - `Registry.rebind`
- 2 – dotaz na objekt příslušící danému jménu- `Registry.lookup`
- 3 – práce se vzdáleným objektem - `Remote_object.remote_method`



Části Java RMI aplikace

- vzdálený interface
 - deklarace metod vzdáleného objektu vyvolávaných klientem
- vzdálený objekt
 - implementace vzdáleného interface
- klient
- komunikace
 - stub, skeleton (Java 2 skeleton nemá)



Remote

- remote methods
 - metody volané mezi JVM
- remote objects
 - objekty s remote metodami
- remote interface
 - určuje remote objekty, rozšiřuje interface *java.rmi.Remote*
 - všechny remote metody musí deklarovat *java.rmi.RemoteException*
- remote stub
 - nahrazuje remote objekt lokální reprezentací (proxy)
 - stejná množina remote interface, marshaling, unmarshaling



Tvorba aplikace

- remote interface
- implementace remote objektů
- implementace klienta
- překlad zdrojových kódů
- vytvoření stubu
- distribuce aplikace
- start aplikace



Implementace

- deklarace remote interface
- definice konstrukturu remote objektu
- implementace všech remote metod
- vytvoření a instalace remote objektů
 - main metoda
 - vytvoření a instalace security manageru
 - vytvoření instancí remote objektů
 - registrace objektů ve jmenných službách
 - rmiregistry, JNDI



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 engine;

import java.rmi.*;
import java.rmi.server.*;

public class MathServerImpl extends UnicastRemoteObject implements
MathServer {

    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 engine;

import java.rmi.*;

public class Server
{
    public static void main(String[] args) {
        if (System.getSecurityManager() == null)
            System.setSecurityManager(new RMISecurityManager());
        String name = "//localhost:" + args[0] + "/mathserver";
        try {
            MathServerImpl obj = new MathServerImpl();
            Naming.rebind(name, obj);
        }
        catch (Exception e) {
            System.err.println("Data exception: " + e.getMessage());
        }
    }
}
```



RMI - Client

```
package RMI;

import java.rmi.*;

public class Client {
    public static void main(String args[]) {
        if (System.getSecurityManager() == null)
            System.setSecurityManager(new RMISecurityManager());

        try {
            MathServer mth;
            String name = "//" + args[0] + ":" + args[1] + "/mathserver";
            mth = (MathServer) Naming.lookup(name);
            int a = Integer.valueOf(args[2]).intValue();
            int b = Integer.valueOf(args[3]).intValue();
            System.out.println(a + " + " + b + " = " + mth.secti(a, b));
        }
        catch (Exception e) {
            System.err.println("Data exception: " + e.getMessage());
        }
    }
}
```



Bezpečnost

- defaultní metody
- policytool.exe || policytool
- java.security.policy

```
grant {  
    permission java.net.SocketPermission "*:1024-65535", "connect,accept";  
    permission java.io.FilePermission "\data\-", "read";  
    permission java.io.FilePermission "c:\\home\\ann\\classes\-", "read";  
};
```

```
grant {  
    permission java.security.AllPermission;  
};
```



Překlad a spuštění

```
javac RMI/MathServerImpl.java
```

```
rmi -d . RMI.MathServerImpl
```

```
rmiregistry
```

```
java -Djava.security.policy=policy RMI/Server  
-Djava.rmi.server.codebase=file:/home/class/
```