Programming Homework 3

Deposit beacon periodically

Agent (java)

RMI

Manager (java)

Send commands and get result back

JNI

C/C++

1. Re-implement hw1 using RMI and JNI
2. Optional: Support version upgrade
   - Upon detecting a new agent, the server checks the agent's version and upgrade its code on-the-fly when necessary
     1) Send the new code to the agent
     2) Install and run the new version
     3) Terminate the old version
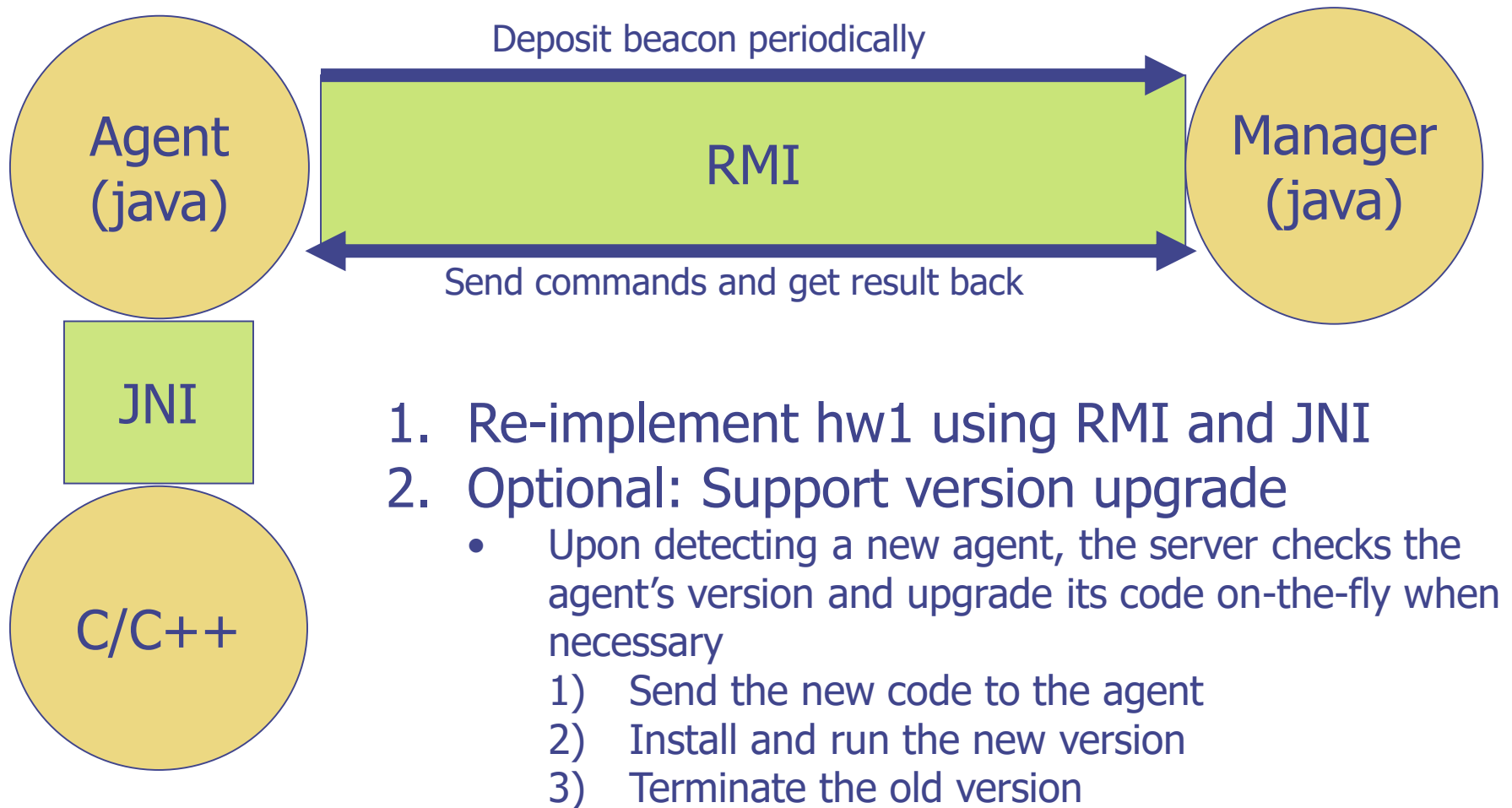
Manager Design: Two threads

◆ Manager: maintain a list of beacons, print out if an agent is gone

◆ BeaconListenerRegister: register one remote object providing this interface:

```
public interface BeaconListener extends java.rmi.remote
{
        public int deposit(Beacon b); // put b to the list
}
```

◆ If a beacon, say b, is from a new agent, the manager gets the client's local time and version by making two RMI calls which pass these objects

```
public class GetLocalTime
{
        int time;
        char valid;
}

public class GetVersion
{
        int version;
}
```

Agent Design: Two Threads

◈ BeaconSender: periodically deposits a beacon to manager using RMI

```
class BEACON
{
    int         ID;            /* randomly generated at startup */
    int         StartUpTime;   /* the time when this client starts */
    String      CmdAgentID;    /* the registry string of CmdAgent */
}
```

◈ CmdRegister: registers a remote object that provides this interface:

```
public interface CmdAgent extends java.rmi.remote
{
    public Object execute(String CmdID, Object CmdObj);
}
```

CmdAgent.execute()

- ◆ Pseudo code

```
public Object execute(String CmdID, Object CmdObj)
{
    if (CmdID.equal("GetLocalTime")
    {
        return C_GetLocalTime((GetLocalTime) CmdObj);
    }
    else
    ...
}
```

- ◆ C_GetLocalTime() must be implemented in C/C++

```
jobject C_GetLocalTime execute(jobject CmdObj)
{
    1. Set CmdObj.valid = 0
    2. Get current system time and store it to CmdObj.time.
    3. If step 2 is successful, set CmdObj.valid = 1
    4. return CmdObj
}
```

Code Submission and Testing

- Submit your code through your Canvas account by the deadline
- You may be asked to run your code on your computer to show it works properly
    - Run the server
    - Open several consoles
    - Run a client on each console
        - the server should be able to detect when a new client shows up by printing out the client's OS and startup time
    - Terminate one client
        - the server should be able to detect the client dies by printing out a message
    - Terminate one client and run the client again immediately
        - the serve should be able to detect the client dies and resurrects