Today

Distributed Objects (Lesson 6)

✓ Spring Kernel

⇒ Java RMI + EJB

Wednesday

Distributed Subsystems (Lesson 7)

Java History

Invented by James Gosling at Sun

Originally called Oak later Java

Originally intended for use with PDAs

then to Set-top boxes and then onto the Internet for powering e-Commerce!

Java Distributed object model

Java Distributed object model

Remote object
- accessible from different address spaces

Remote Interface
- Declarations for methods in a remote object

Failure semantics
- clients deal with RMI exceptions

Similarities/Differences to local objects
- object references can be params
- Param only as value/result

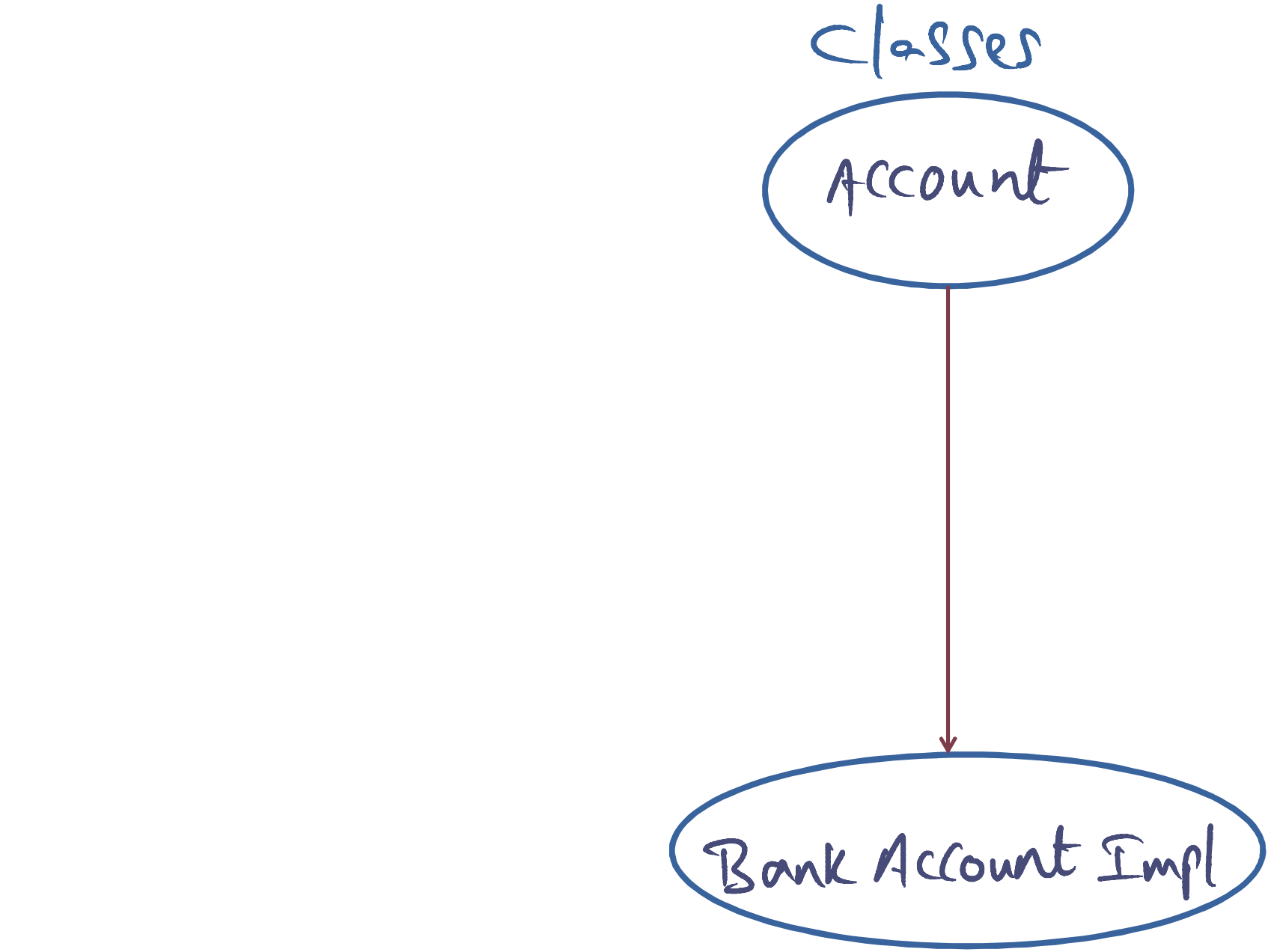
Bank account Example

Server API

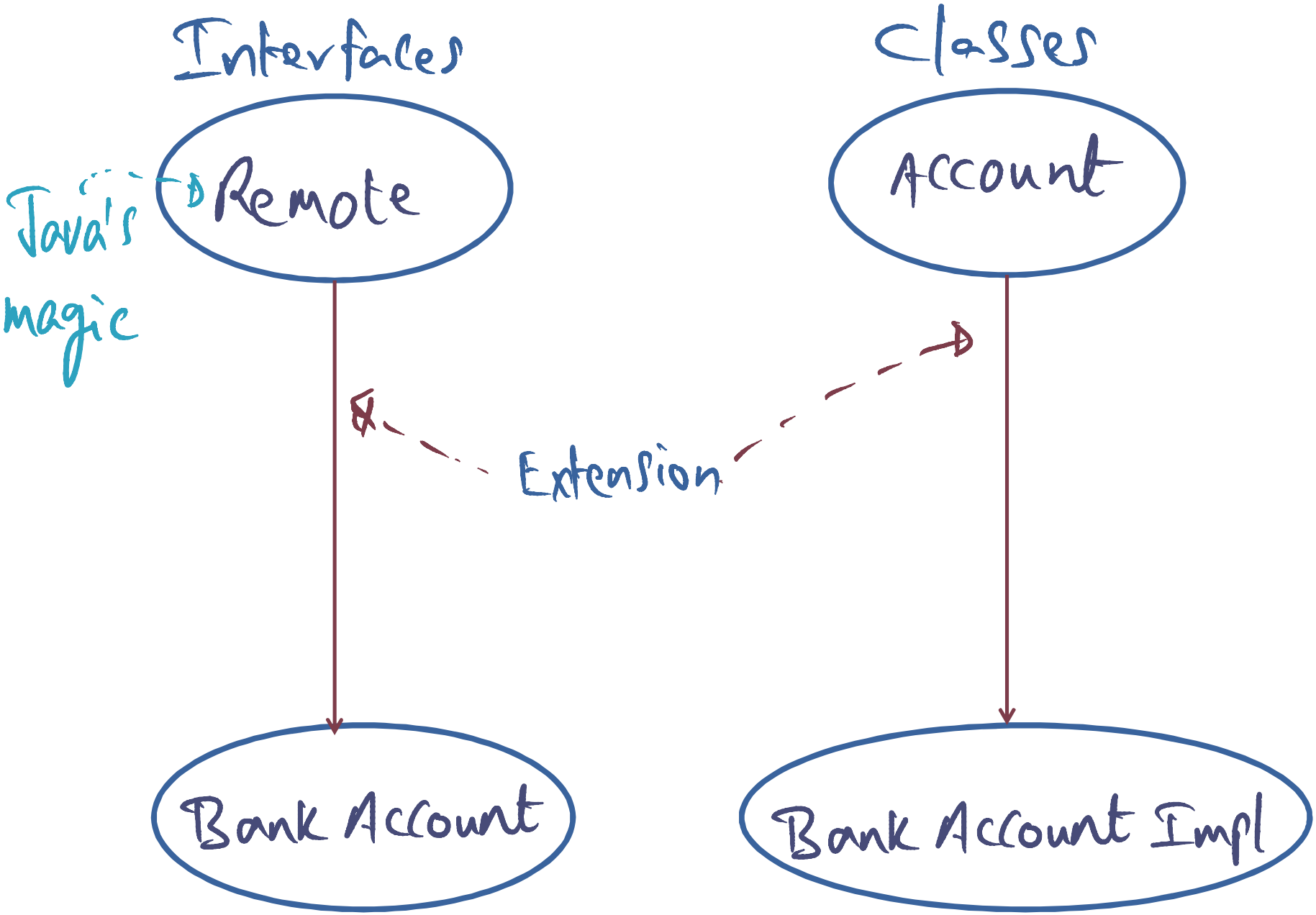
- Deposit
- Withdraw
- Balance

How Best to implement using Java?

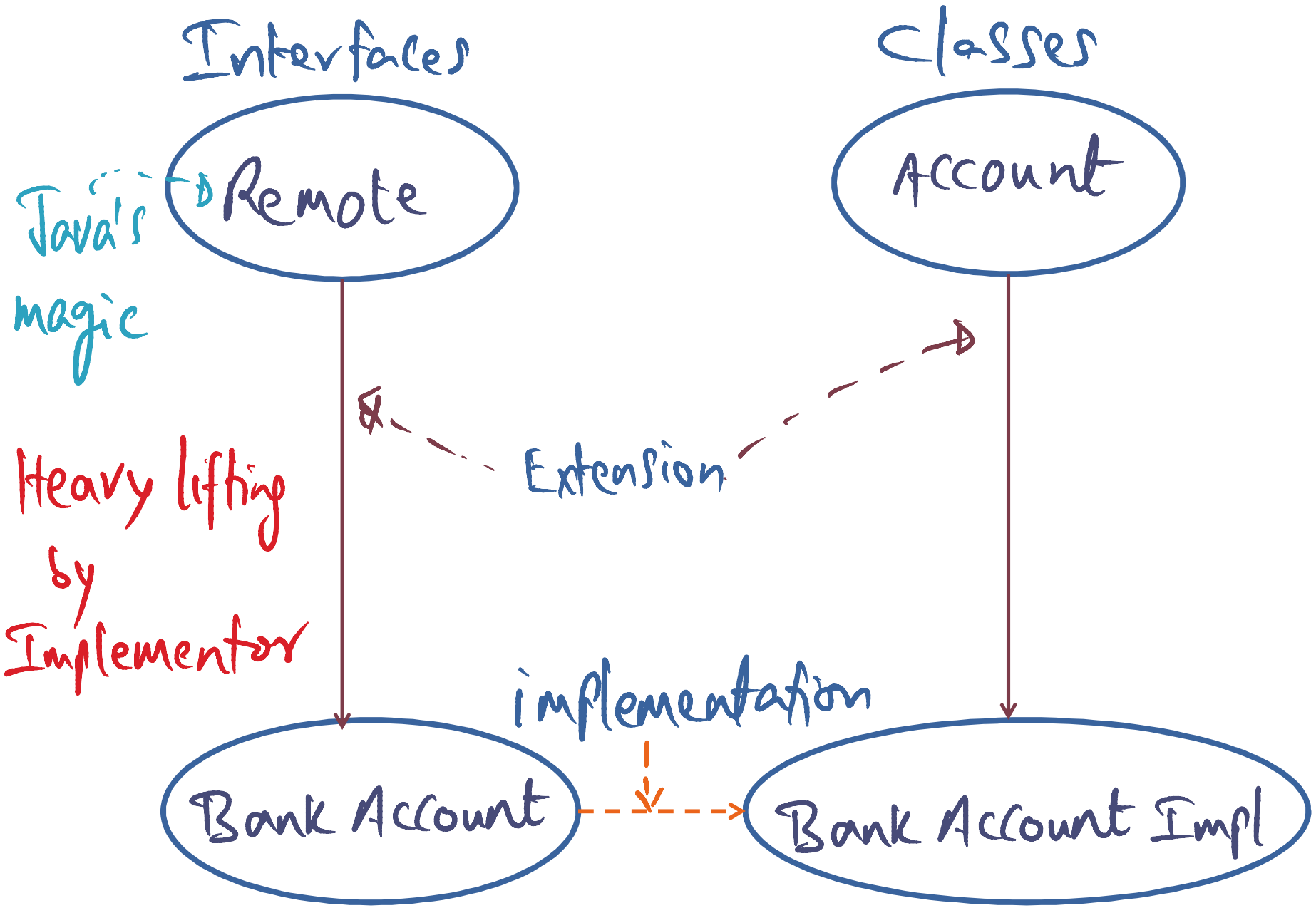
First choice: Reuse of local implementation



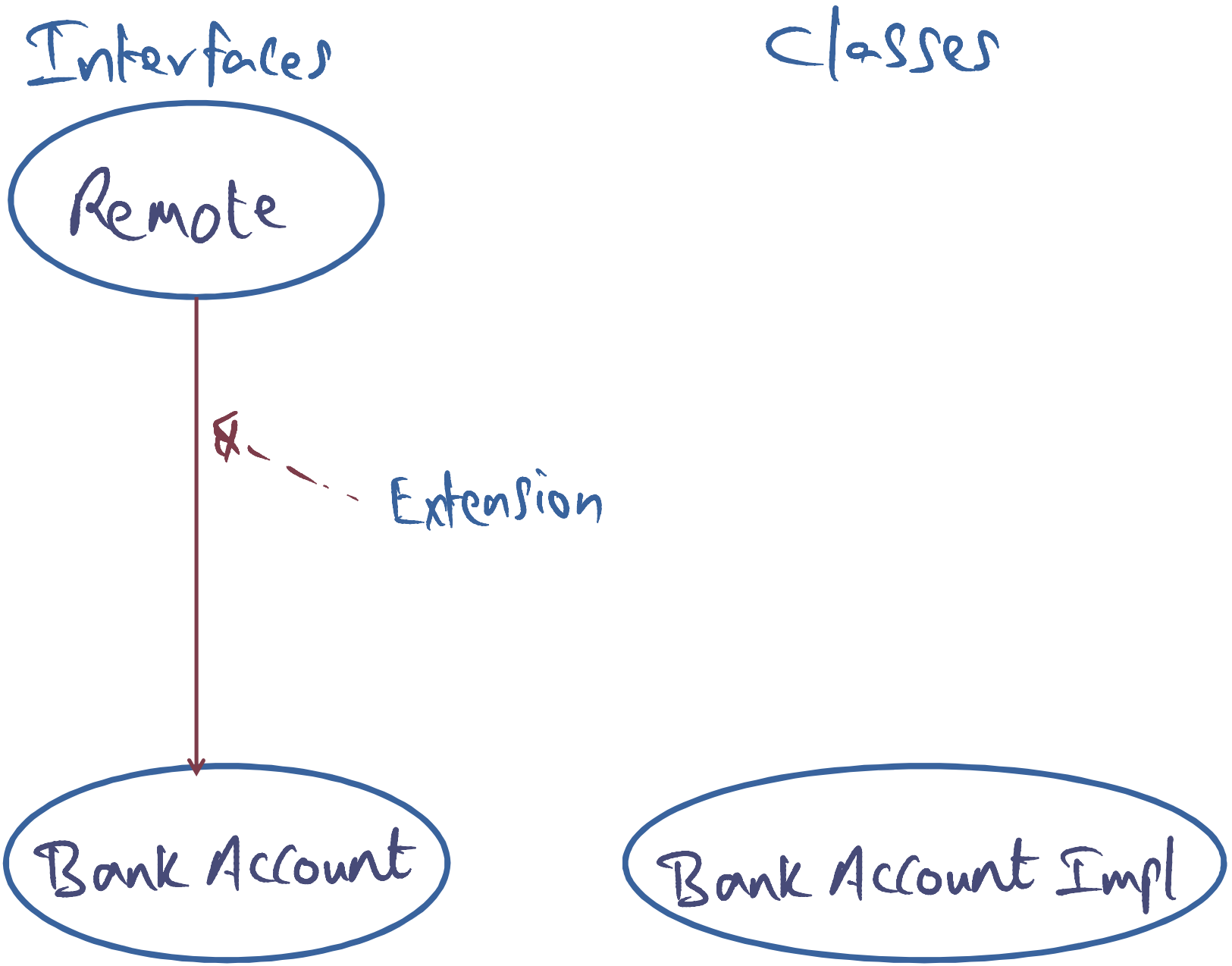
First choice: Reuse of local implementation



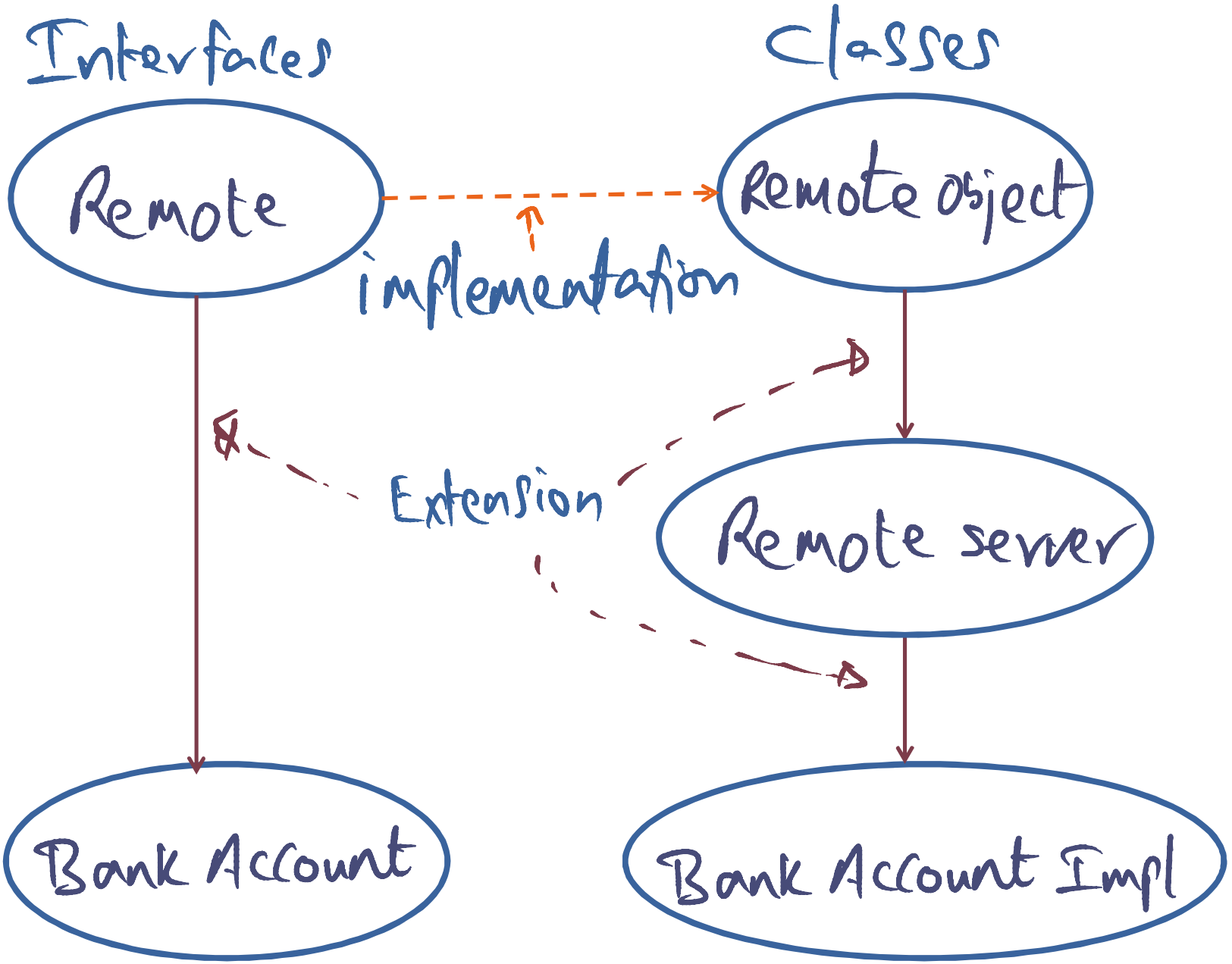
First choice: Reuse of local implementation



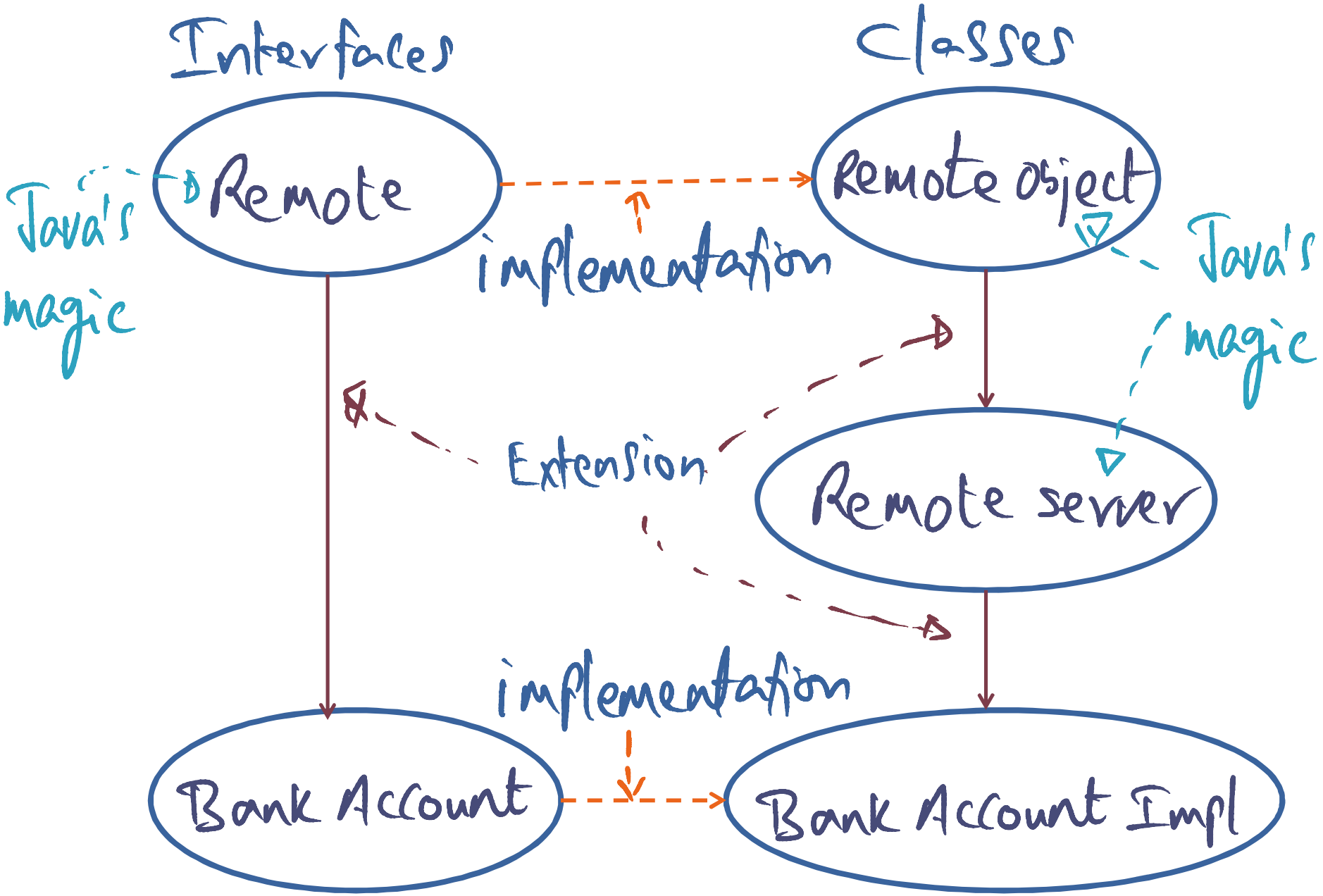
Second choice: Reuse of "Remote"



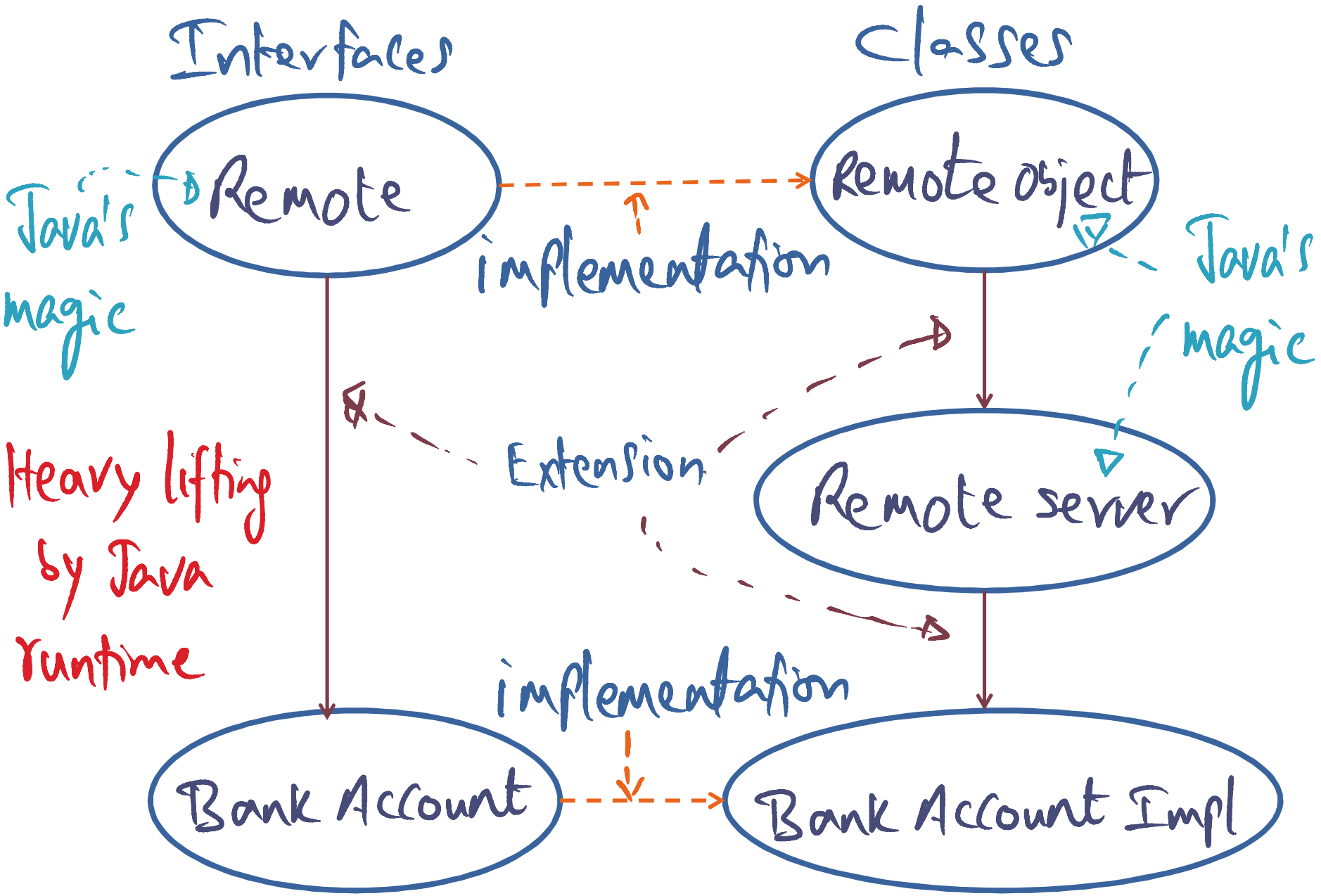
Second choice: Reuse of "Remote"



Second choice: Reuse of "Remote"



Second choice: Reuse of "Remote"



"Local" implementation

Implementor makes instances of objects remotely accessible ⇒ not preferable

"Remote" implementation

Java RMI does the heavy lifting to make server object visible to network clients

⇒ Preferable

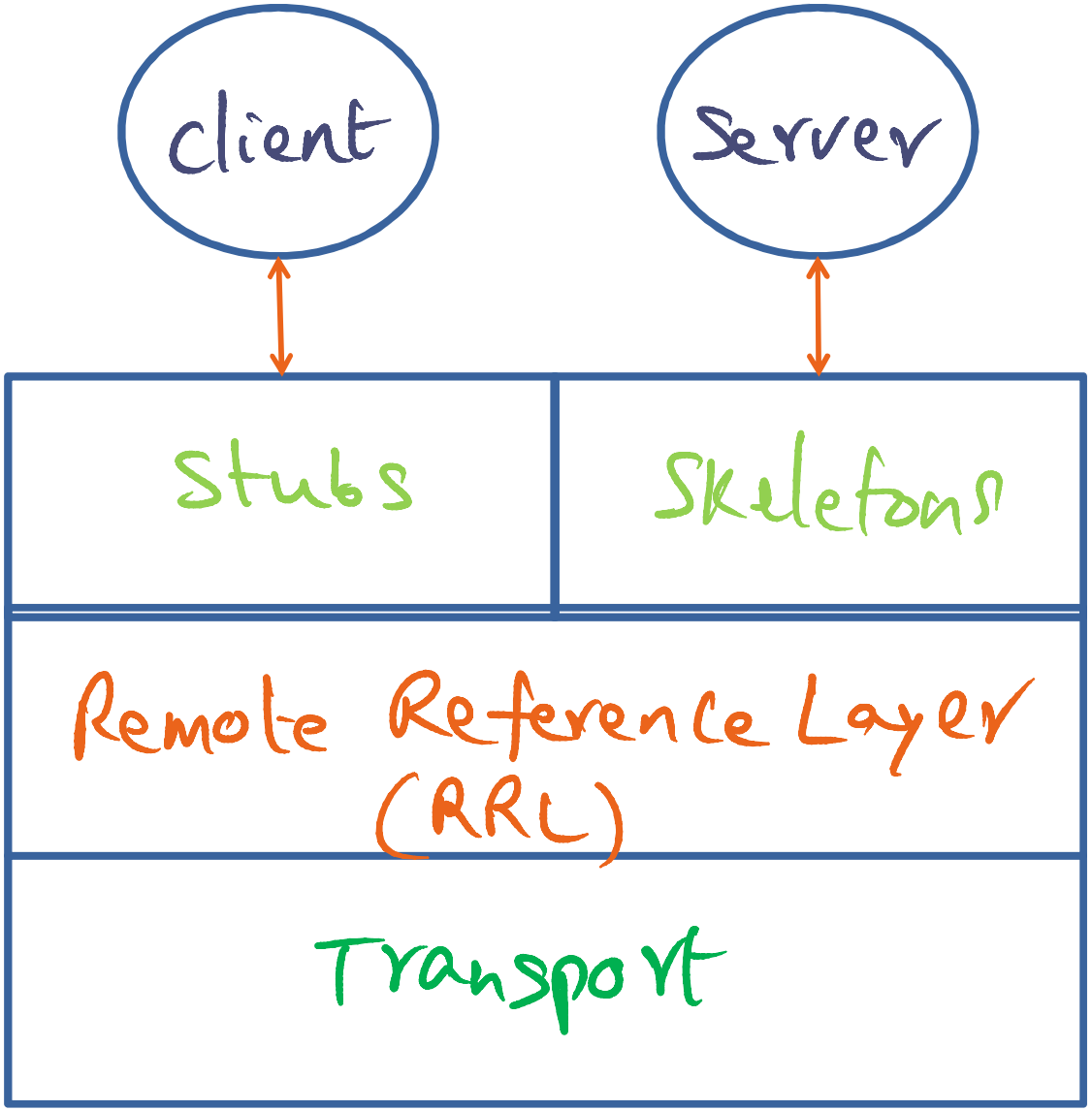
Java RMI at work – server

```
BankAccount acct = new BankAcctImpl();
URL url = new URL("my web address")
Java.rmi.Naming.bind(url, acct);
```

Java RMI at work – client

```
BankAccount acct =
                Java.rmi.Naming.lookup(url);
float balance;
acct.deposit (##);
acct.withdraw(##);
balance = acct.balance();
```

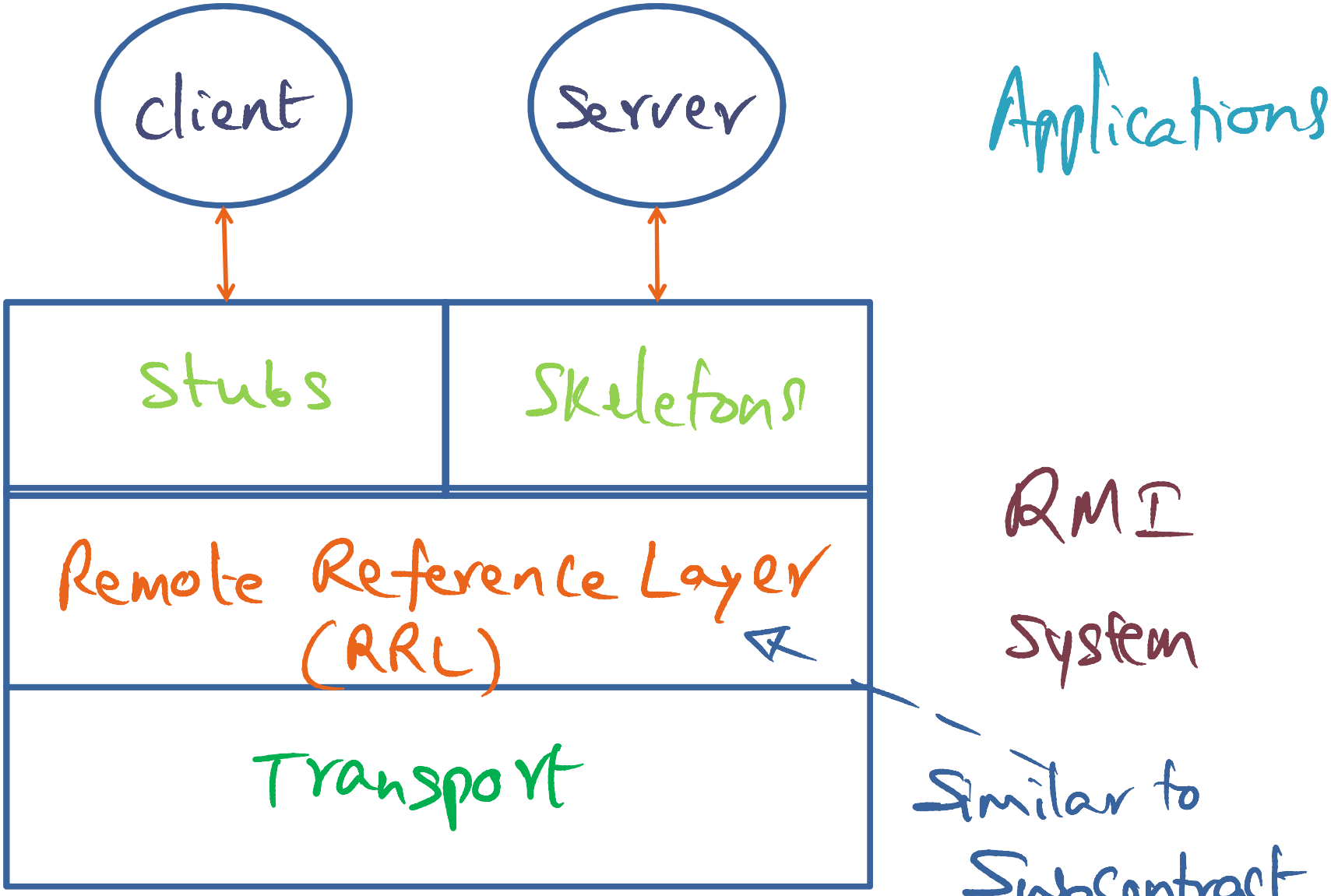
RMI Implementation – RRL



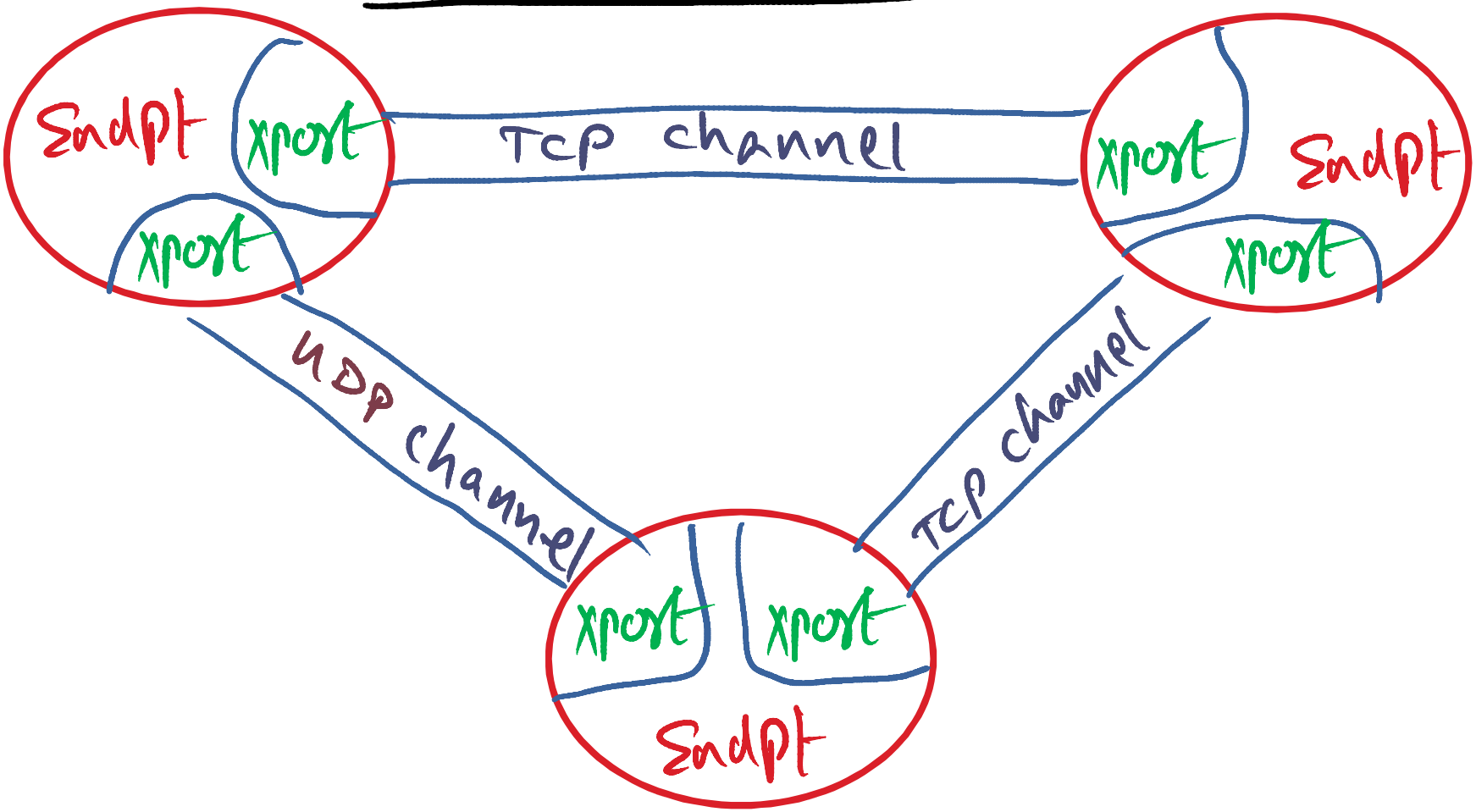
Applications

RMI System

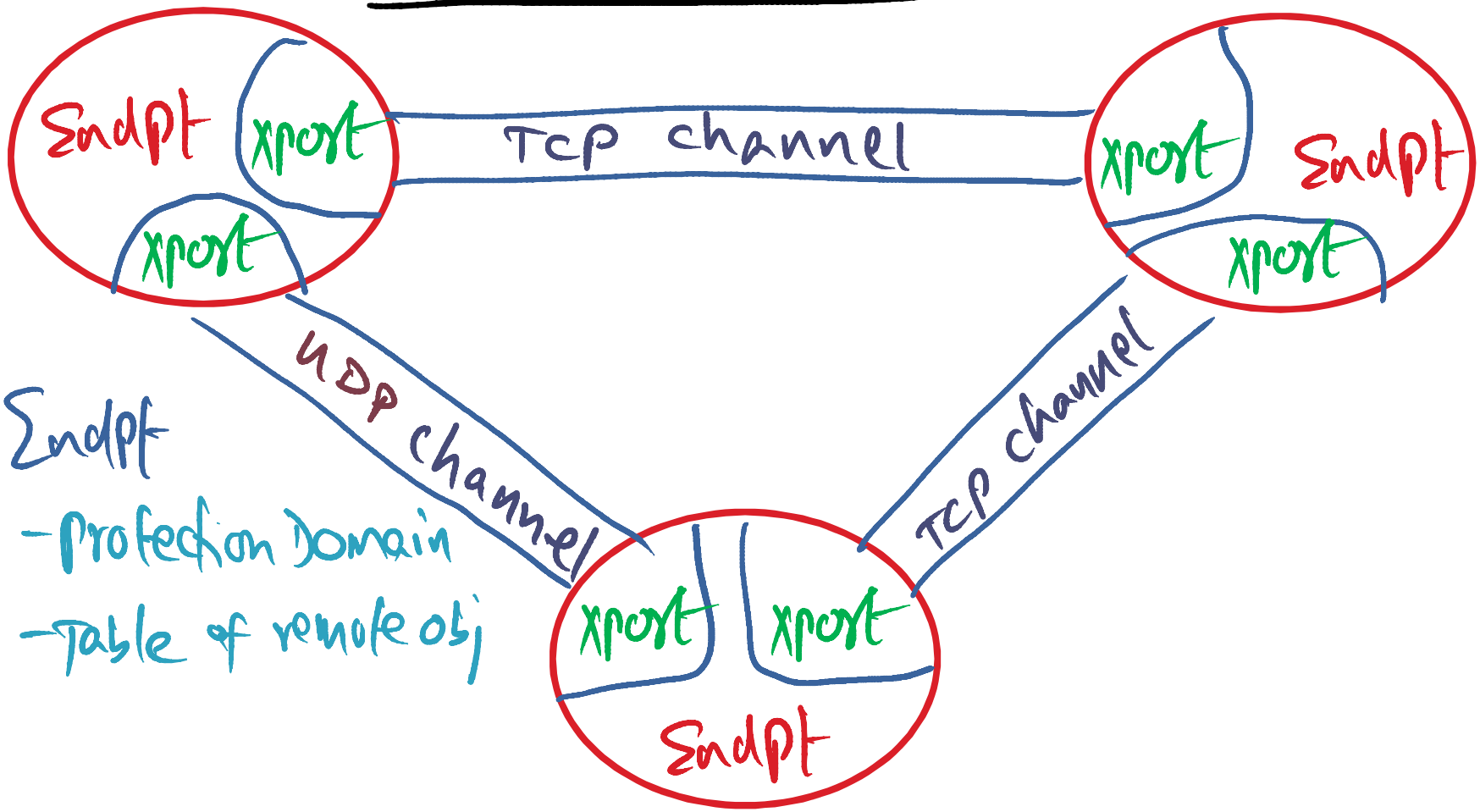
RMI Implementation – RRL



RMI Implementation – Transport

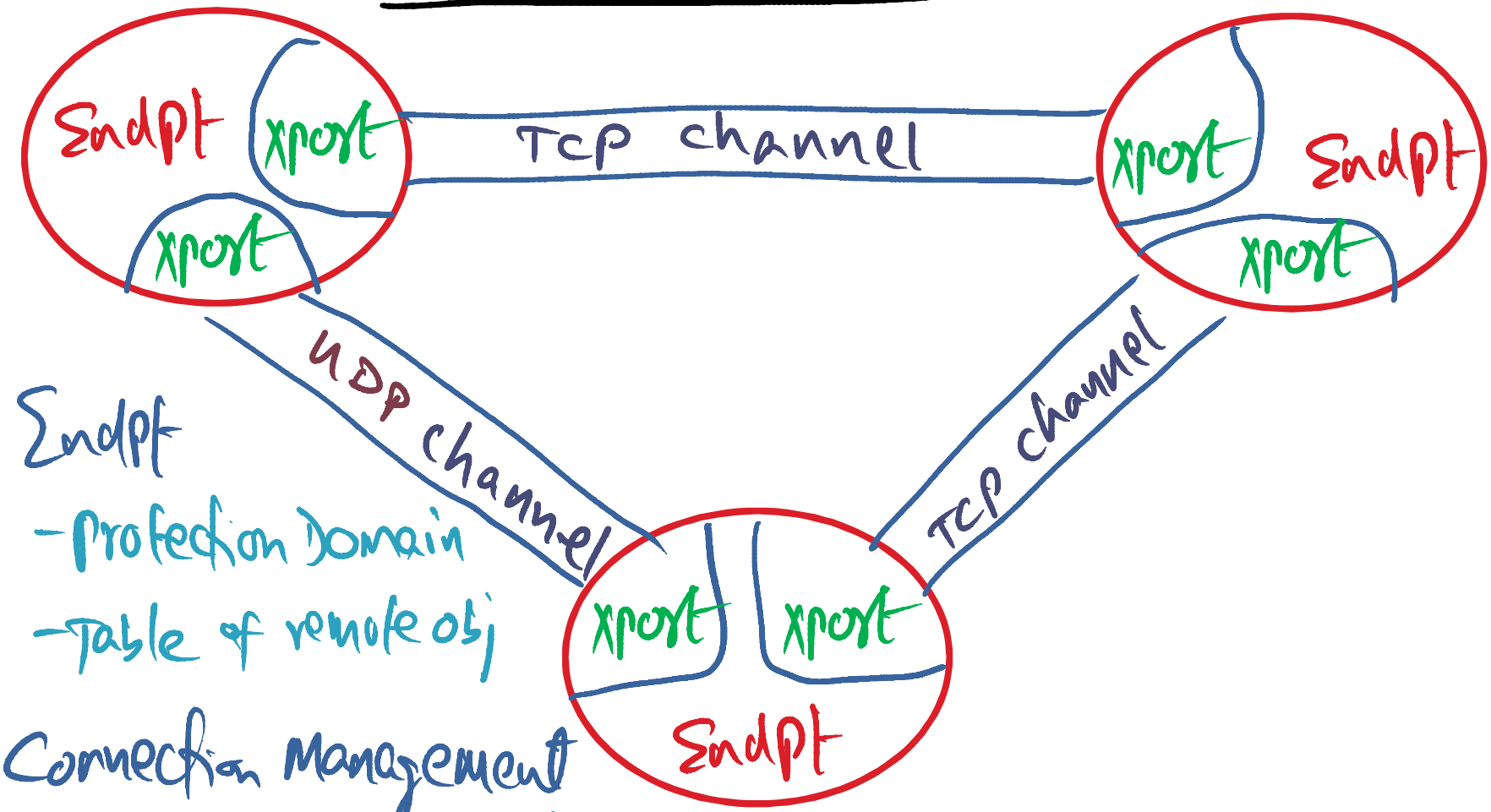


RMI Implementation – Transport



Endpt
- Protection Domain
- Table of remote obj

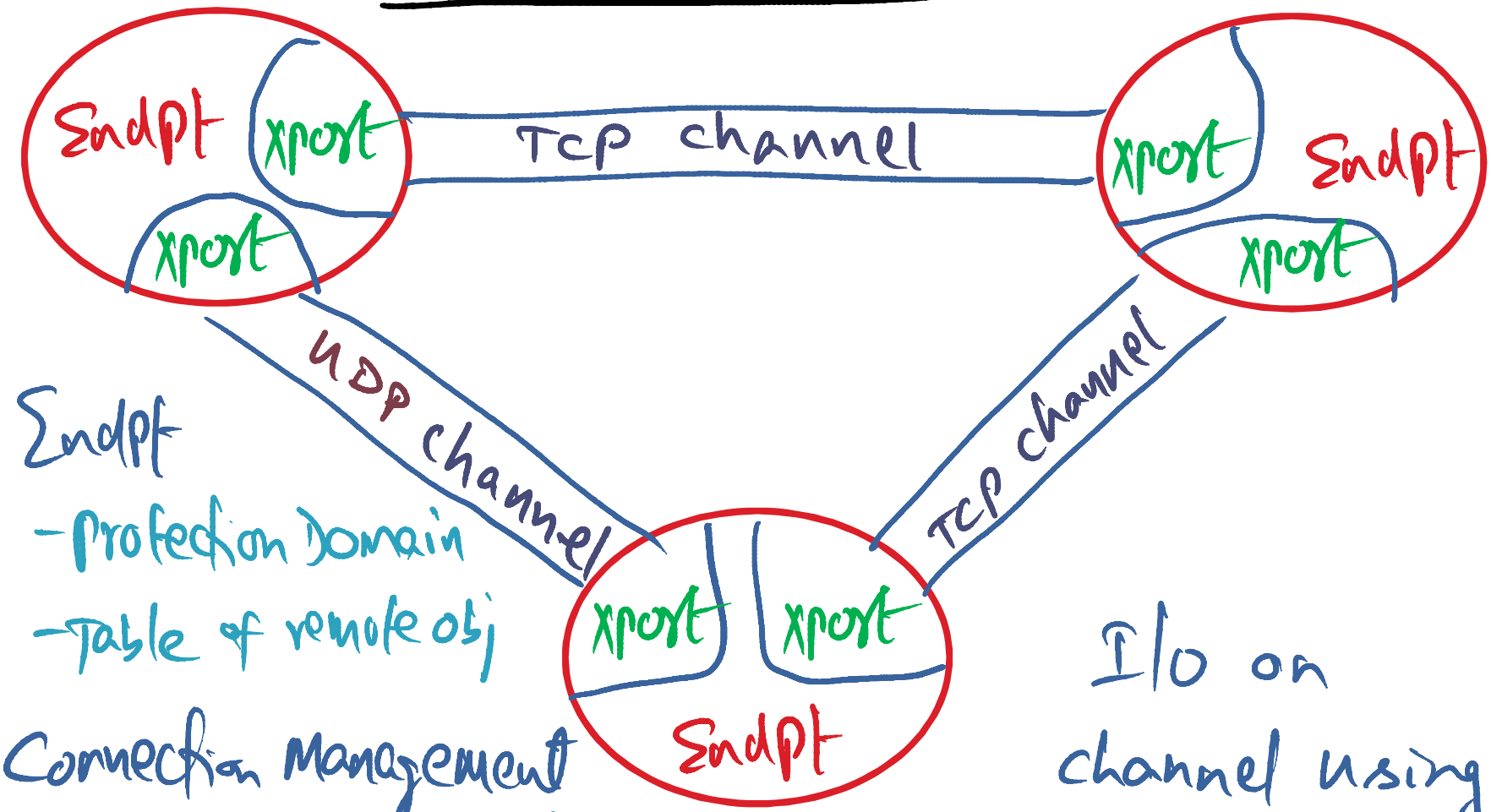
RMI Implementation – Transport



Endpt
- Protection Domain
- Table of remote objs

Connection Management
- Setup, teardown, listen
- liveness monitoring
- choice of xport

RMI Implementation – Transport

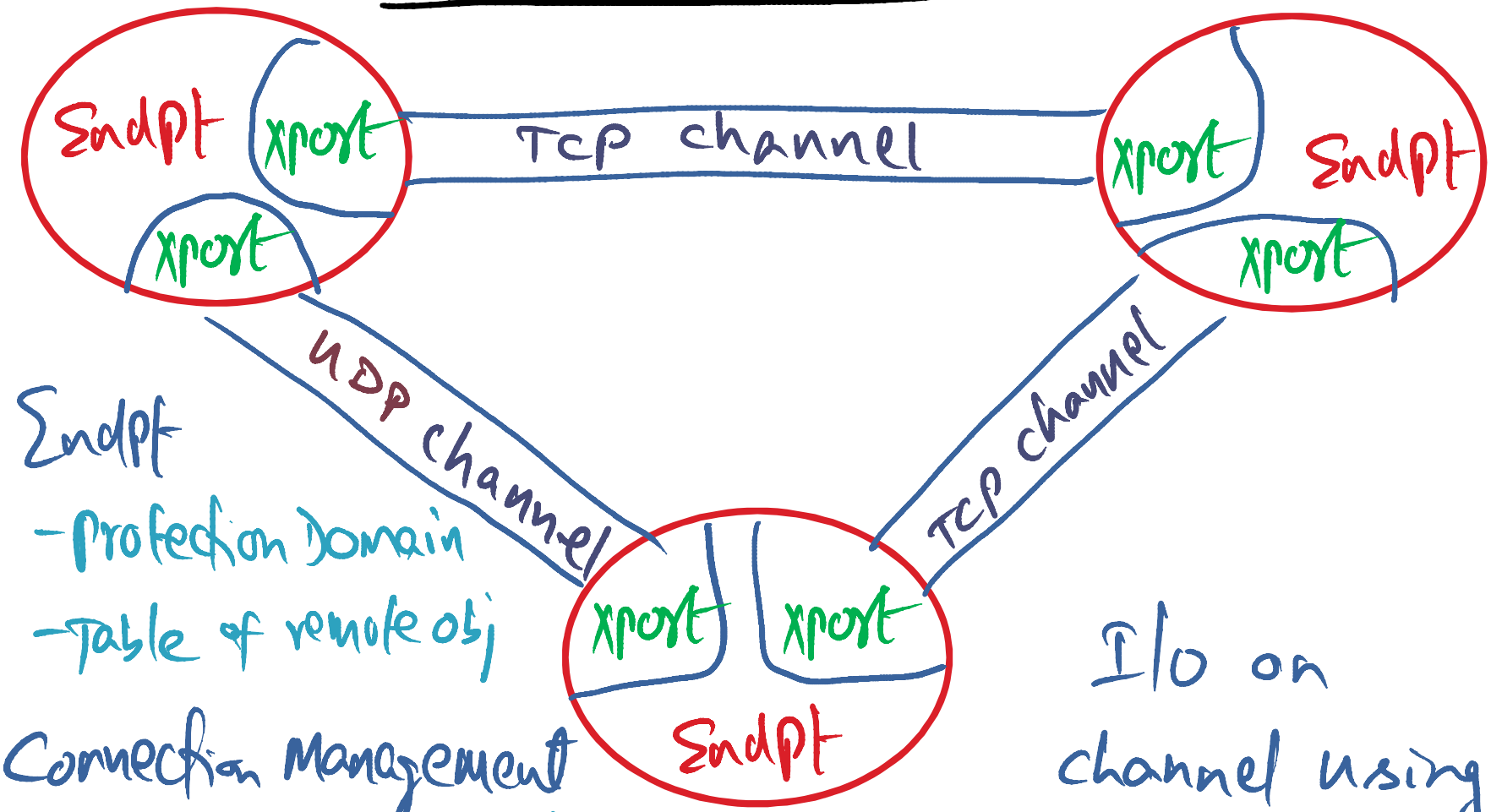


Endpt
- Protection Domain
- Table of remote objs

Connection Management
- Setup, teardown, listen
- liveness monitoring
- choice of xport

I/O on channel using Connections

RMI Implementation – Transport



Endpt
- Protection Domain
- Table of remote objs

Connection Management
- Setup, teardown, listen
- liveness monitoring
- choice of xport

I/O on channel using Connections

RRL decides right xport to use

Key Takeaways

- "Pie in the sky" research ideas => usable technology
- Connection between subcontract mechanism in Spring and Java RMI

Subtle issues involved in the implementation of the RMI system

- distributed garbage collection, dynamic loading of stubs on the client side, sophisticated "sandboxing" on the client and server sides to ward off security threats.

Learn more about these issues by reading the assigned paper and surfing the net.