

# **Esercitazioni Ing.Sw**

Alessandro Margara / Gian Enrico Conti  
II sem 2023

Webex:

<https://politecnicomilano.webex.com/meet/gianenrico.conti>

# RMI vs TCP

ing. Gian Enrico Conti

Prova Finale - Ingegneria del Software - AA 2022/23

- Sono simili?
- Implementiamo entrambi.. da cosa parto?

...

- Sono simili?
- Implementiamo entrambi.. da cosa parto?

...

Assolutamente NO!

- TCP (socket..)
- RMI ha GIA astrazione di rete

...

TCP: dove fare voi i thread

RMI: ha gia un thread x invocazione remota sul server

..

TCP: e' async e quindi attenzione!

RMI: e' sincrónico!

(Provate esempio con breakpoint nel metodo remoto..**public boolean login(String nick)** .. il client e' bloccato!)

Accesso mutuamente esclusivo?

Nel model sul server, SI (forse overkill...)

Sia x TCP che RMI.

(Se avete SOLO TCP... fine....)

Logica:

RMI piu recente -> piu astratto -> si parte da RMI

"Ho scritto del codice RMI che fa:"

```
bool sendMesg(Msg msg, int opt1, int opt2, int PlayerID) {..
```

```
....
```

```
    bool sent = sendMesg(Msg(MOVE_TILE, 2,3, 1));
```

NO!

...



```
bool moveTile(tile, toPosition) {..
```

```
....
```

```
Tile tile = new Tile(....);
```

```
bool weSuccessfullyMoved = moveTile();
```

```
..
```

RMI offre invocazione di metodi remoti

Una implementazione è considerata corretta se sfrutta questo modello.

In concreto: usare RMI per invocare un metodo “sendMessage” NON è una implementazione paradigmatica, mentre una invocazione remota dei metodi del controller lo è.

**nota:**

Le specifiche non richiedono cambiamento “on the fly” del protocollo di comunicazione per un client, ma richiedono che diversi client possano usare protocolli diversi.

Suggeriamo di partire da RMI e poi passare a socket:  
Questo dovrebbe far capire che con i socket sarà necessario aggiungere un **layer** che di fatto realizza l'invocazione dei metodi a partire dai messaggi ricevuti.

Mandare comandi?

- Pensa astratto.. moveTo.. shuffle.. (in RMI posso aver un result...)

```
bool weSuccessfullyMoved = moveTile();  
If (weSuccessfullyMoved) {  
    Board updatedBoard = askUpdatedBoard();  
}
```

E TCP?

Mandiamo messaggi.. avremo strato di astrazione ..

TCP:

```
bool sent = moveTile();
```

(Bool in questo caso non arriva dal server..)

Mandiamo messaggi, avremo strato di astrazione (altra classe..)

```
bool moveTile(){  
..... Msg msg = new Msg(....
```

TCP:

```
bool sent = moveTile();
```

```
...
```

E SULLA RISPOSTA TCP nel thread avro':  
(al netto del parser... dei cmd..)

```
If (receivedMesg instanceof Board){  
    Board updatedBoard = (Board)receivedMesg;
```

(E lo passo al modello del client.. "osservato") -> UI

**instanceOf ???**