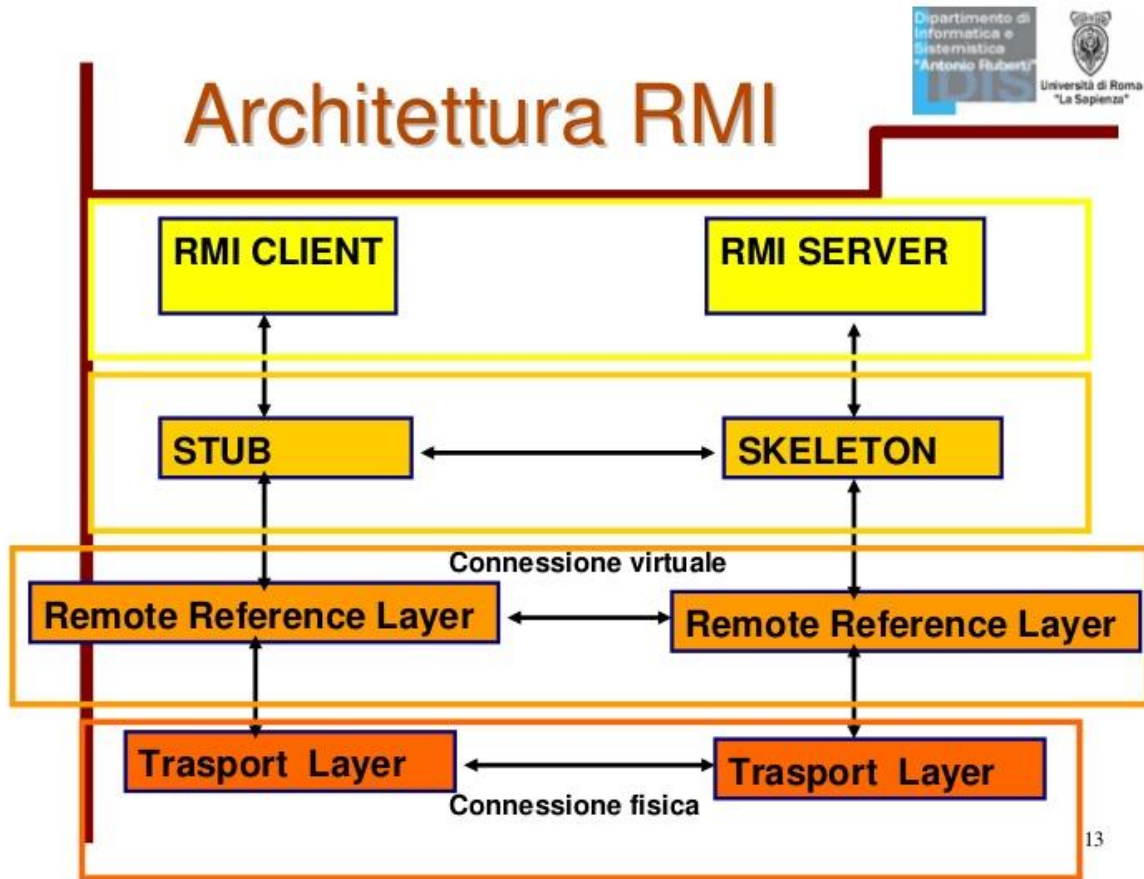


# RMI

1)



RMI funktioniert ähnlich wie ein Client-Server Programm. Der Client erstellt ein Objekt, das auf dem Server implementiert ist, und lässt es auf dem Server auf dem Server ausführen. Der Server hat alle bereitgestellten Remote-Objekte in der Registry liegen. Der Client sucht das richtige heraus und verwendet es.

**Stub:** Simuliertes Objekt auf dem Client der das Remote-Objekt darstellt. Wie ein Proxy für das reale Objekt.

**Skeleton:** Gegenstück zum Stub, liegt auf dem Server und behandelt die Anfragen des Clients.

**Codebase:** definiert wie die Hauptklasse heißt und wo sie eventuell liegt.

**Hostname:** gibt die IP-Adresse des Servers an

**useCodebaseOnly:** lädt wenn true nur den Codebase

**security.policy:** Zusätzliche Policy-Dateien einbinden lassen.

Für normale Programmierer nicht wichtig aber Professionelle können weitere Rechte, Sicherungen,... hiermit einbinden.

**SecurityManagers:** prüft ob Klassen eine Operation durchführen darf, Zugriffe Kontrolliert(Netz, Dateisystem, Betriebssystem), hat aber einige Sicherheitslücken,

Mirco Borri