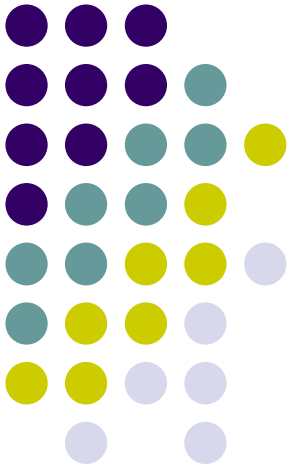
Κατανεμημένα Συστήματα

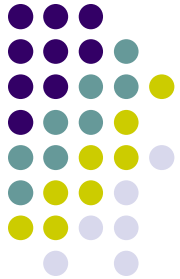
Μάθημα #4



Περιεχόμενα

- RPC στο Σύστημα SUN
- Απομακρυσμένα αντικείμενα
- Κλήση απομακρυσμένων μεθόδων (RMI)
- Κατανεμημένα αντικείμενα στο DCE
- Σύντομη αναφορά στο Java RMI

RPC στο Σύστημα SUN



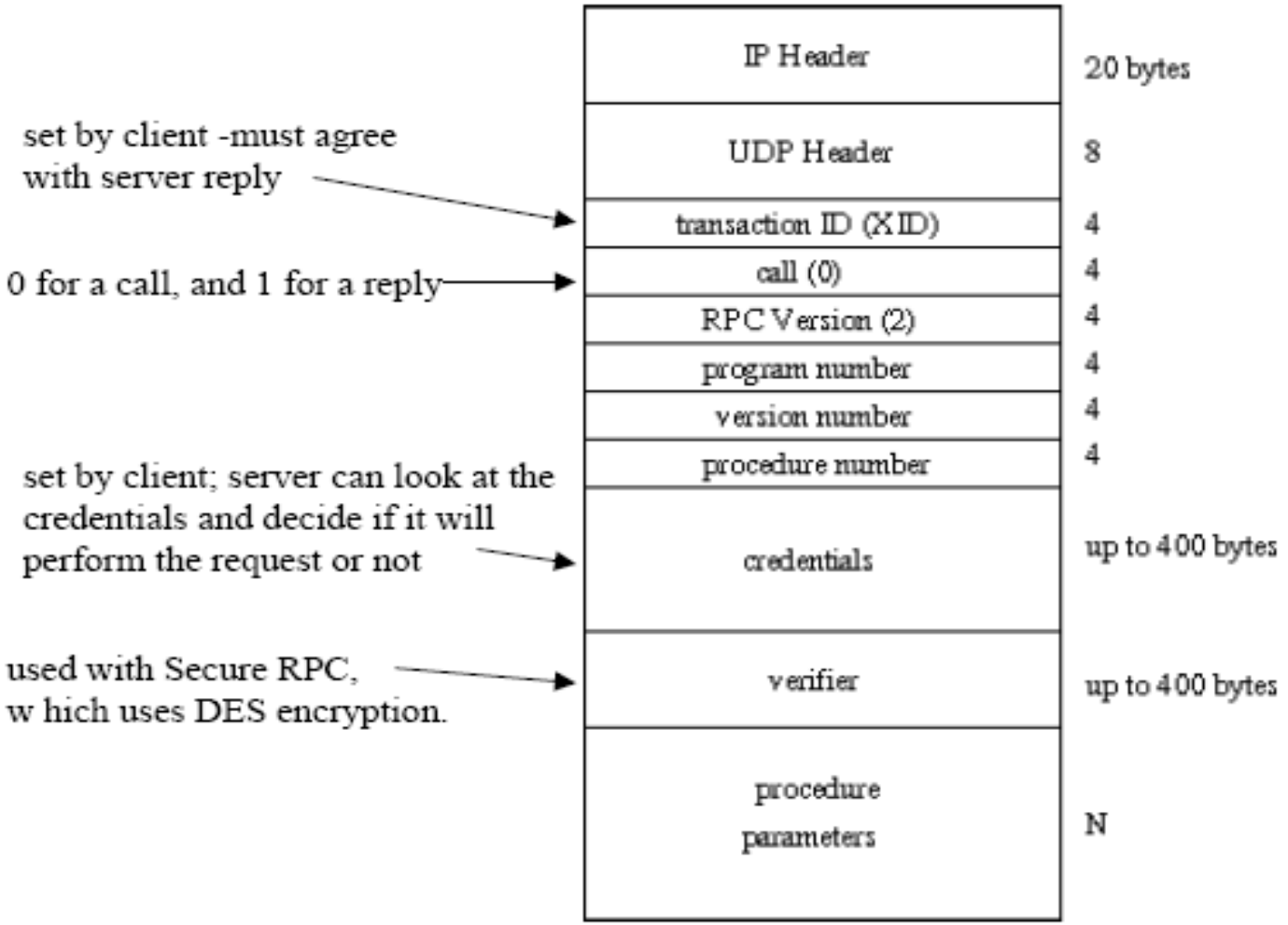
- **Μία συλλογή απομακρυσμένων διαδικασιών αποτελεί ένα *πρόγραμμα***
- **Κάθε πρόγραμμα έχει ένα *μοναδικό αριθμό* (στον server) και έναν *αριθμό έκδοσης* (version number)**
- **Κάθε διαδικασία του προγράμματος έχει και αυτή έναν αριθμό (μοναδικό εντός προγράμματος)**
- **Μία διαδικασία σε ένα server καθορίζεται μοναδικά από:**

{program-number, version-number, procedure-number}

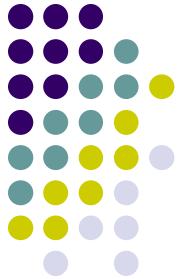
RPC στο Σύστημα SUN

- **Καθορίζει format για μηνύματα, παραμέτρους, αποτελέσματα**

Παράδειγμα μηνύματος RPC με UDP



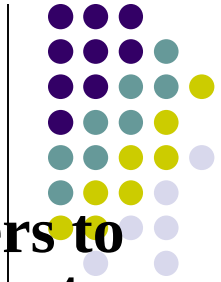
RPC στο Σύστημα SUN



- **Ορίζεται ένα timeout, και ένας συνολικός χρόνος αναμονής total_time**
  - **Μετά την πρώτη εκπομπή, ο client περιμένει απάντηση για χρονικό διάστημα ίσο με timeout**
  - **Αν δεν λάβει απάντηση, επανεκπέμπει όσες φορές χρειαστεί, έως ότου εκπνεύσει ο total_time**
  - **Αν εκπνεύσει, τότε το RPC Runtime επιστρέφει timeout error**

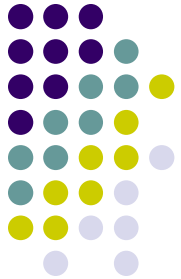
- **Χρησιμοποιεί XDR για ορίσματα, αποτελέσματα και header data**

XDR



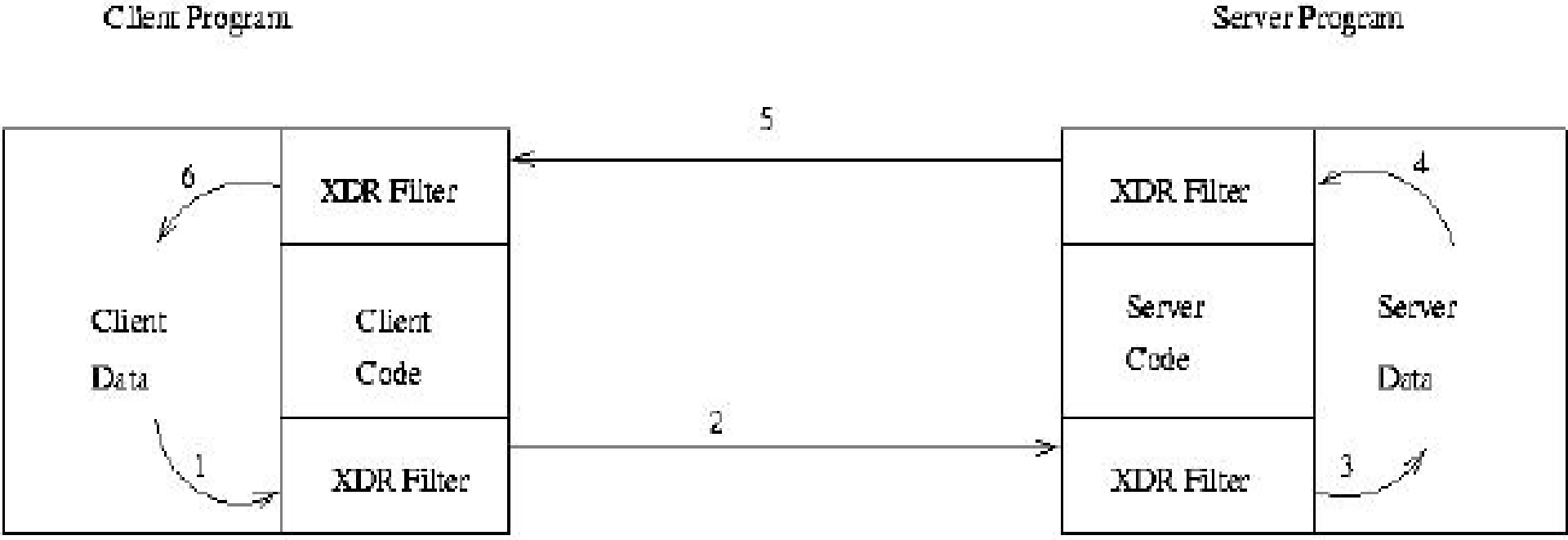
- **XDR is a set of library routines that enable C programmers to describe arbitrary data structures in a machine-independent way.**
- **XDR is the backbone of the Sun RPC package. Data for RPCs are transmitted using this standard.**
- **XDR works across different languages, operating systems, and machine architectures.**
- **One can use rpcgen to write XDR routines even in cases where no RPC calls are being made.**
- **C programs that use XDR routines must include the file <rpc/xdr.h>, which contains all the necessary interfaces to the XDR system.**

RPC στο Σύστημα SUN



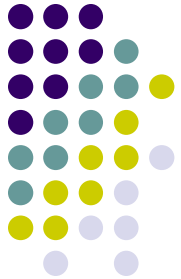
- Χρησιμοποιεί ένα binder (rpcbind deamon) σε κάθε server μηχάνημα
  - Με *rpcinfo* παίρνουμε κατάλογο όλων των registered προγραμμάτων από τον binder
- Ο binder είναι και αυτός ένα RPC πρόγραμμα (ID 100000, version=2). Ο binder ξεκινά πριν από όλους τους servers
- Κάθε RPC server στην εκκίνησή του συνδέεται με ένα socket και κάνει registration στον rpcbind (μέσω ενός RPC call) τα
  - program number, version number,
  - και socket (protocol, port number)
  - Ο RPC client στέλνει τα program number, version number, και protocol στον binder και μαθαίνει το port number του server
  - Κατόπιν ο client στέλνει RPC call message στο socket του server, ο οποίος εκτελεί την κατάλληλη διαδικασία

RPC Dataflow



1. Client encodes data through XDR filter.
2. Client passes XDR encoded data accross network to remote host
3. Server decodes data through XDR filter.
4. Server encodes function call result through XDR filter
5. Server pass XDR encoded data accross network back to client
6. Client decodes RPC result through XDR filter and continues processing

Υλοποίηση SUN RPC



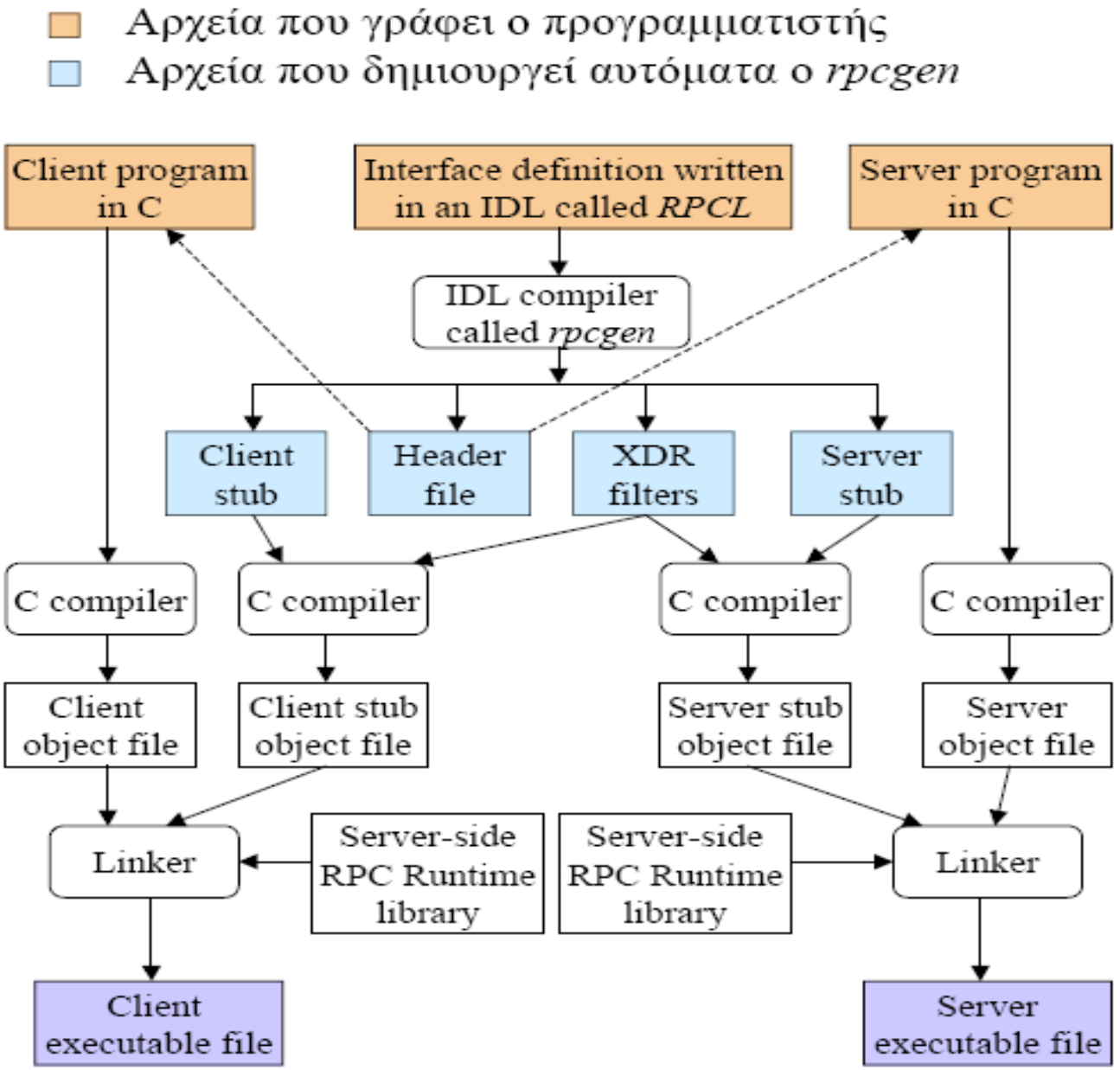
- **Απαιτείται η περιγραφή του RPC interface μιας εφαρμογής (ποιες απομακρυσμένες διαδικασίες προσφέρονται και με τι ορίσματα, όνομα προγράμματος και version number κλπ)**

- **Γράφεται από τον προγραμματιστή σε ένα IDL (interface definition language) που ονομάζεται RPCL και αποτελεί επέκταση του XDR**

- **Διατίθεται compiler (rpcgen) για αυτόματη δημιουργία stub routines, header file, xdr filter file**

- **Ο προγραμματιστής γράφει επίσης την απομακρυσμένη διαδικασία, και το πρόγραμμα client**

Υλοποίηση SUN RPC



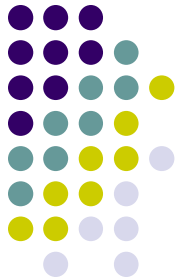
SUN RPC - Μετατροπή Τοπικής Διαδικασίας σε Απομακρυσμένη

Έστω το πρόγραμμα (printmsg.c), το οποίο τυπώνει ένα μήνυμα στην κονσόλα n φορές, και τρέχει σε έναν Η/Υ.

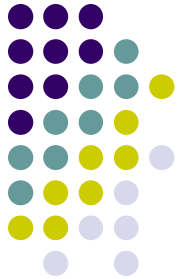
Η μετατροπή του σε πρόγραμμα που τυπώνει σε απομακρυσμένους Η/Υ, μέσω RPC απαιτεί την εξής διαδικασία:

- **γράφουμε ένα protocol specification σε γλώσσα RPC (msg.x) το οποίο περιγράφει την απομακρυσμένη έκδοση του `printmsg()'**
- **εκτελούμε *rpcgen msg.x* και δημιουργούνται αυτόματα τα αρχεία:**
  - **msg.h // include file**
  - **msg_xdr.c //XDR filters**
  - **msg_clnt.c // client stub**
  - **msg_svc.c //server stub**

SUN RPC - Μετατροπή Τοπικής Διαδικασίας σε Απομακρυσμένη

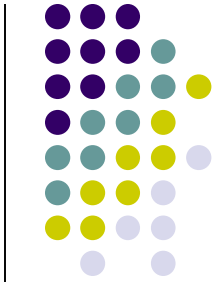


- **Στη συνέχεια γράφουμε το client πρόγραμμα rprintmsg.c και το απομακρυσμένο πρόγραμμα msg_proc.c**
- **Κατόπιν δημιουργούμε το εκτελέσιμο αρχείο του server:**
  - ***gcc msg_proc.c msg_svc.c msg_xdr.c –o msg_server***
- **Και το εκτελέσιμο του client:**
  - ***gcc rprintmsg.c msg_clnt.c msg_xdr.c –o rprintmsg***
- **Αν ο msg_server εκτελείται στη μηχανή hostx.yyy.gr, μπορούμε να τυπώσουμε 5 φορές το μήνυμα «!?» από απομακρυσμένο Η/Υ με την εντολή**
  - ***rprintmsg* hostx.yyy.gr «!?» *5***



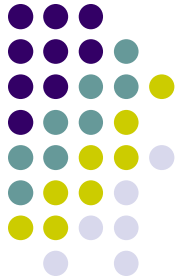
Κλήση απομακρυσμένων μεθόδων - Remote Method Invocation (RMI)

Αντικείμενα



- Οι **κλάσεις (classes)** ορίζουν τύπους αντικειμένων.
- Τα **αντικείμενα** (objects) είναι **στιγμιότυπα (instances)** των κλάσεων
- Ένα αντικείμενο ενθυλακώνει (encapsulates): δεδομένα (**κατάσταση**) και λειτουργίες (**μέθοδοι**)
- Οι μέθοδοι γίνονται διαθέσιμες μέσω μιας **διασύνδεσης (interface)**
- Ένα αντικείμενο μπορεί να υλοποιεί πολλές διασυνδέσεις.
- **Κατανεμημένο αντικείμενο:** ένα αντικείμενο του οποίου οι μέθοδοι μπορούν να κληθούν εξ' αποστάσεως. Το αντικείμενο και η διασύνδεσή του βρίσκεται σε ένα μηχάνημα αλλά η διασύνδεσή του μπορεί να γίνεται διαθέσιμη σε απομακρυσμένες διεργασίες

Κλήση απομακρυσμένων μεθόδων (RMI)



RMI: Αν ένας πελάτης **«συνδυαστεί με ένα αντικείμενο» τότε μπορεί** να καλέσει τις μεθόδους του αντικειμένου μέσω του **εξουσιοδότη** ή **διαμεσολαβητή**.

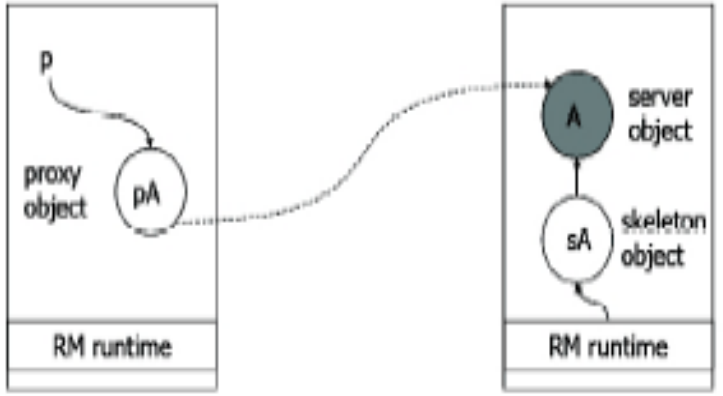
Αντικείμενο Εξουσιοδότης ή Διαμεσολαβητής (proxy): μια υλοποίηση της διασύνδεσης ενός αντικειμένου που φορτώνεται στο χώρο διευθύνσεων του πελάτη όταν ο πελάτης συνδυάζεται (bind) με το αντικείμενο.

Ο διαμεσολαβητής παρατάσσει αιτήσεις κλήσεων μεθόδων σε μηνύματα και αποπαρατάσσει τις απαντήσεις

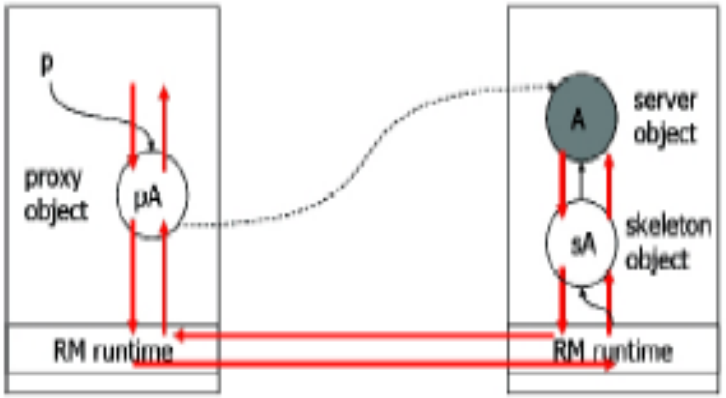
Αντικείμενο Σκελετός (skeleton): στέλεχος διακομιστή που παραλαμβάνει τις αιτήσεις, τις αποπαρατάσσει σε κλήσεις μεθόδων της διασύνδεσης που βρίσκεται στον διακομιστή και παρατάσσει και προωθεί τις απαντήσεις στον διαμεσολαβητή.

'Εξουσιοδοτημένο' αντικείμενο

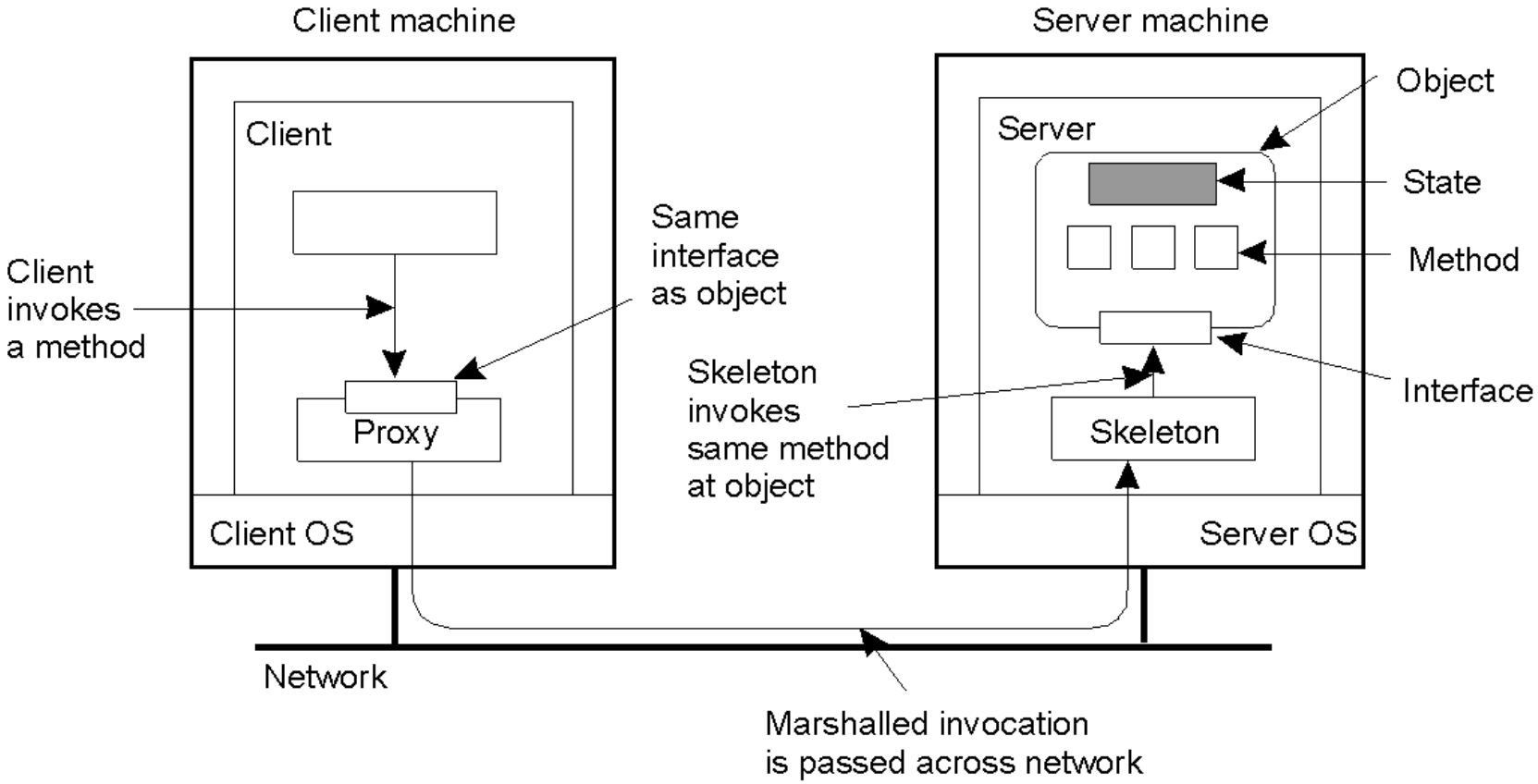
- Για κάθε αντικείμενο υπάρχει ένα αντικείμενο σκελετού (skeleton object)
- Για κάθε απομακρυσμένη αναφορά υπάρχει ένα αντικείμενο εξουσιοδότης (proxy object)
- Το αντικείμενο εξουσιοδότης αντιστοιχεί στο client stub
- Το αντικείμενο σκελετού αντιστοιχεί στο server stub
- Η χρήση των μεθόδων του κατανεμειμένου αντικείμενου γίνεται μέσω του RM runtime

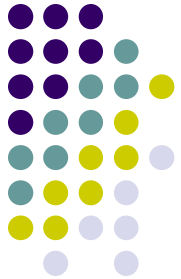


Κλήση διαδικασίας

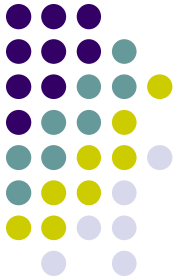


Κατανεμημένα Αντικείμενα

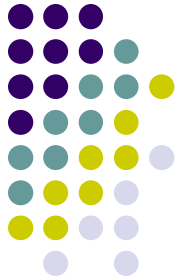




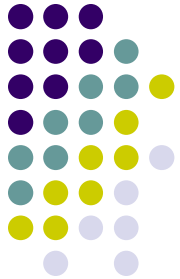
- Ένα κατανεμημένο αντικείμενο δημιουργείτε από την κλάση που το ορίζει με την πρώτη αναφορά – ίδιος τρόπος με τα `απλά', τοπικά, αντικείμενα
- Η μονάδα που δημιούργησε το αντικείμενο είναι υπεύθυνη για την διαχείριση του
- Κάθε απομακρυσμένη διεργασία που επιθυμεί να χρησιμοποιήσει το αντικείμενο, πρέπει να επικοινωνήσει με την μονάδα που το δημιούργησε (διαχειριστή)
  - Η μονάδα διαχειριστής κρατάει δείκτες προς τις διεργασίες που έχουν ζητήσει πρόσβαση στο κατανεμημένο αντικείμενο
  - Ορισμένοι μηχανισμοί περισυλλογής άχρηστων αντικείμενων (Garbage Collection) ακολουθούν την ίδια τακτική
- Όταν ένα αντικείμενο δεν έχει κανένα δείκτη προς κάποια διεργασία, θεωρείτε άχρηστο και περισυλλέγεται



- Μία διεργασία εξυπηρέτης δημιουργεί το πρώτο αντικείμενο σύμφωνα με την κλάση που το ορίζει
- Καταχωρεί τα στοιχεία του αντικειμένου (και της κλάσης που το ορίζει) σε μια υπηρεσία καταλόγου σύμφωνα με ένα όνομα
- Όταν μια διεργασία πελάτης θελήσει να χρησιμοποιήσει το κατανεμημένο αντικείμενο, το αναζητά μέσω της υπηρεσίας καταλόγου με βάση το όνομα
- Η αναζήτηση επιστρέφει πληροφορίες για την τοποθεσία της κλάσης του αντικειμένου, την ύπαρξη ή όχι αντικειμένου

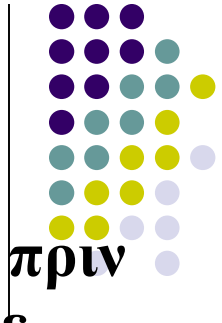


- Ένα αντικείμενο μπορεί να μην χρησιμοποιηθεί για μεγάλο χρονικό διάστημα
  - για λόγους εξοικονόμησης πόρων, το σύστημα *απομακρύνει* τα ανενεργά αντικείμενα σε δευτερεύουσα μονάδα αποθήκευσης
- Όταν μια διεργασία χρησιμοποιήσει τον τοπικό εξουσιοδότη (proxy) το σύστημα επαναφέρει το αντικείμενο στην πρωτεύουσα μονάδα αποθήκευσης (*επαναφορά*)
- Η απομάκρυνση και επαναφορά αντιστοιχεί στην λογική διαχείρισης σελίδων εικονικής μνήμης



- Το σύστημα (*εξυπηρέτης*) που δημιούργησε το κατανεμημένο αντικείμενο A , διατηρεί ένα διάνυσμα $A.v$
- Όταν μια διεργασία *p* (*πελάτης*) δημιουργήσει έναν τοπικό εξουσιοδότη (proxy) για το κατανεμημένο αντικείμενο
  - Το σύστημα εξυπηρέτης προσθέτει την *p* στο $A.v$
- Όταν μια διεργασία *p* (*πελάτης*) περισυλλέξει τον τοπικό εξουσιοδότη (proxy) -- δεν υπάρχει καμία (τοπική) αναφορά
  - Το σύστημα εξυπηρέτης αφαιρεί την *p* από το $A.v$
- Αν ένα αντικείμενο έχει κενό διάνυσμα για ορισμένο χρονικό διάστημα – το σύστημα θεωρεί ότι δεν χρησιμοποιείτε: το αντικείμενο περισυλλέγεται

Συνδυασμός πελάτη με αντικείμενο



Όταν μια διεργασία διατηρεί μια αναφορά σε αντικείμενο, πριν καλέσει τις μεθόδους του πρέπει να συνδυαστεί (bind) με αυτό.

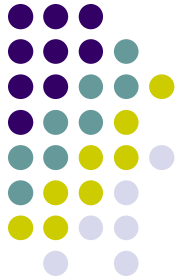
Ο συνδυασμός έχει ως αποτέλεσμα να τοποθετηθεί στο χώρο διευθύνσεων της διεργασίας ένα αντικείμενο διαμεσολαβητής ο οποίος υλοποιεί μία διασύνδεση με τις μεθόδους που μπορεί να καλεί μια διεργασία.

Πώς γίνεται ο συνδυασμός?

Έμμεσος συνδυασμός: απλός μηχανισμός μέσω του οποίου ο πελάτης μπορεί να καλεί μεθόδους χρησιμοποιώντας μια αναφορά σε ένα αντικείμενο.

Ρητός συνδυασμός: καλείται ειδική συνάρτηση που επιστρέφει δείκτη σε τοπικό διαμεσολαβητή, ο οποίος τότε γίνεται διαθέσιμος τοπικά.

Υλοποίηση αναφορών αντικειμένων



Κάθε αναφορά αντικειμένου θα πρέπει να περιλαμβάνει

- **ένδειξη για το ποιο είναι το αντικείμενο**
- **τη διεύθυνση δικτύου του μηχανήματος όπου βρίσκεται το αντικείμενο**
- **το ακραίο σημείο του διακομιστή που το διαχειρίζεται**

Πρόβλημα #1: αν το μηχάνημα του διακομιστή πάθει βλάβη μετά την ανάκαμψη μπορεί να δοθεί στο διακομιστή νέο ακραίο σημείο και όλες οι αναφορές αντικειμένων να είναι άκυρες

Λύση: χρήση τοπικού δαίμονα στο μηχάνημα του αντικειμένου, που θα κρατά τα ζεύγη (διακομιστής, ακραίο σημείο)

Πρόβλημα #2: Η κωδικοποίηση της διεύθυνσης δικτύου σε αναφορά αντικειμένου προϋποθέτει ότι ο διακομιστής δεν μπορεί να μετακινηθεί σε άλλο μηχάνημα.

Λύση: Η αναφορά αντικειμένου να περιλαμβάνει τη διεύθυνση δικτύου ενός διακομιστή θέσεων (location server) που θα παρακολουθεί σε ποιο μηχάνημα εκτελείται ο διακομιστής του αντικειμένου

Η λύση αυτή παρουσιάζει προβλήματα αν μας ενδιαφέρει η επεκτασιμότητα!

Υλοποίηση αναφορών αντικειμένων

Κάθε αναφορά αντικειμένου μπορεί επίσης να περιλαμβάνει:

- **Πληροφορία για τα πρωτόκολλα που χρησιμοποιούν πελάτης και διακομιστής (π.χ. πρωτόκολλο για το συνδυασμό με ένα αντικείμενο).**
- **Λαβή υλοποίησης (implementation handle): παραπέμπει σε μια πλήρη υλοποίηση ενός διαμεσολαβητή που μπορεί να φορτώνει δυναμικά ο πελάτης όταν συνδυάζεται με ένα αντικείμενο.** Π.χ. μια λαβή υλοποίησης μπορεί να έχει τη μορφή διεύθυνσης URL που «δείχνει σε ένα αρχείο. Το πρωτόκολλο συνδυασμού πρέπει να ορίζει ότι ένα τέτοιο αρχείο θα λαμβάνεται δυναμικά.

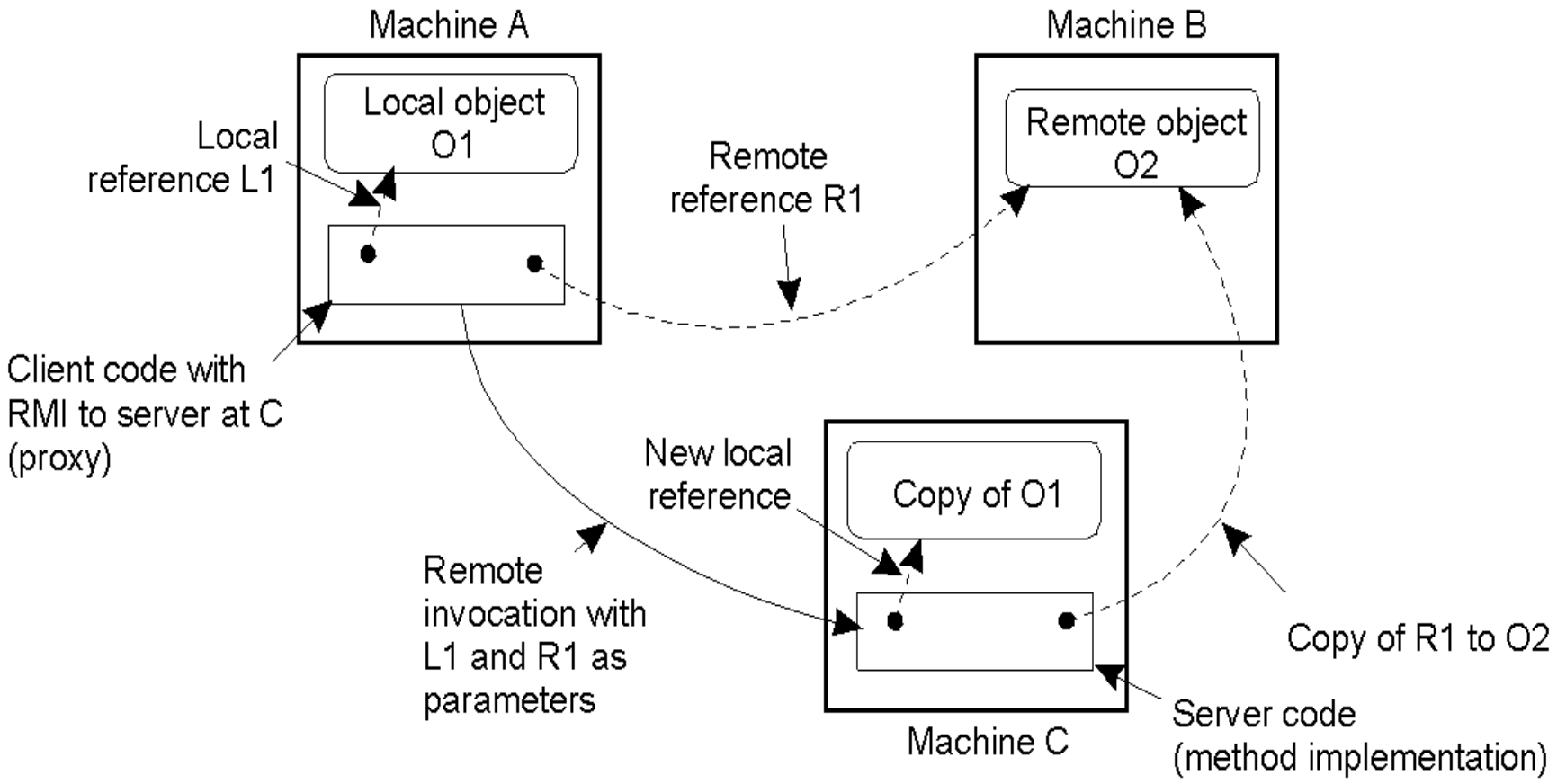
Μεταβίβαση παραμέτρων

- **Όταν καλείται μια μέθοδος με μια αναφορά αντικειμένου ως παράμετρο, η αναφορά αυτή αντιγράφεται και μεταβιβάζεται ως τιμή παραμέτρου μόνο όταν «δείχνει» σε απομακρυσμένο αντικείμενο *(το αντικείμενο μεταβιβάζεται κατ' αναφορά).***

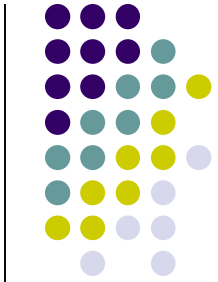
- **Όταν η αναφορά παραπέμπει σε τοπικό αντικείμενο, τότε αυτό αντιγράφεται ολόκληρο και μεταβιβάζεται με την κλήση *(το αντικείμενο μεταβιβάζεται κατ' αξία)***

Π.χ., στην επόμενη διαφάνεια, όταν στη μηχανή Α καλείται το πρόγραμμα διακομιστή που εκετελείται στη μηχανή C, του μεταβιβάζεται ένα αντίγραφο του Ο1 αλλά μόνον ένα αντίγραφο της αναφοράς στο Ο2

Μεταβίβαση παραμέτρων κατ' αναφορά και κατ' αξία



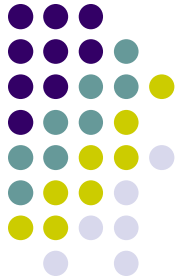
Κατανεμημένα αντικείμενα στο DCE



- **Ένας διακομιστής είναι υπεύθυνος για να δημιουργεί αντικείμενα τοπικά και να κάνει τις μεθόδους τους διαθέσιμες σε απομακρυσμένους πελάτες.**

- **Στο DCE τα κατανεμημένα αντικείμενα προστέθηκαν ως βελτίωση των RPCs. Αντί ο πελάτης να προσδιορίζει μια απομακρυσμένη διαδικασία σε ένα διακομιστή, τώρα προσδιορίζει μια απομακρυσμένη διαδικασία σε ένα αντικείμενο διακομιστή.**

Απομακρυσμένα αντικείμενα στο DCE

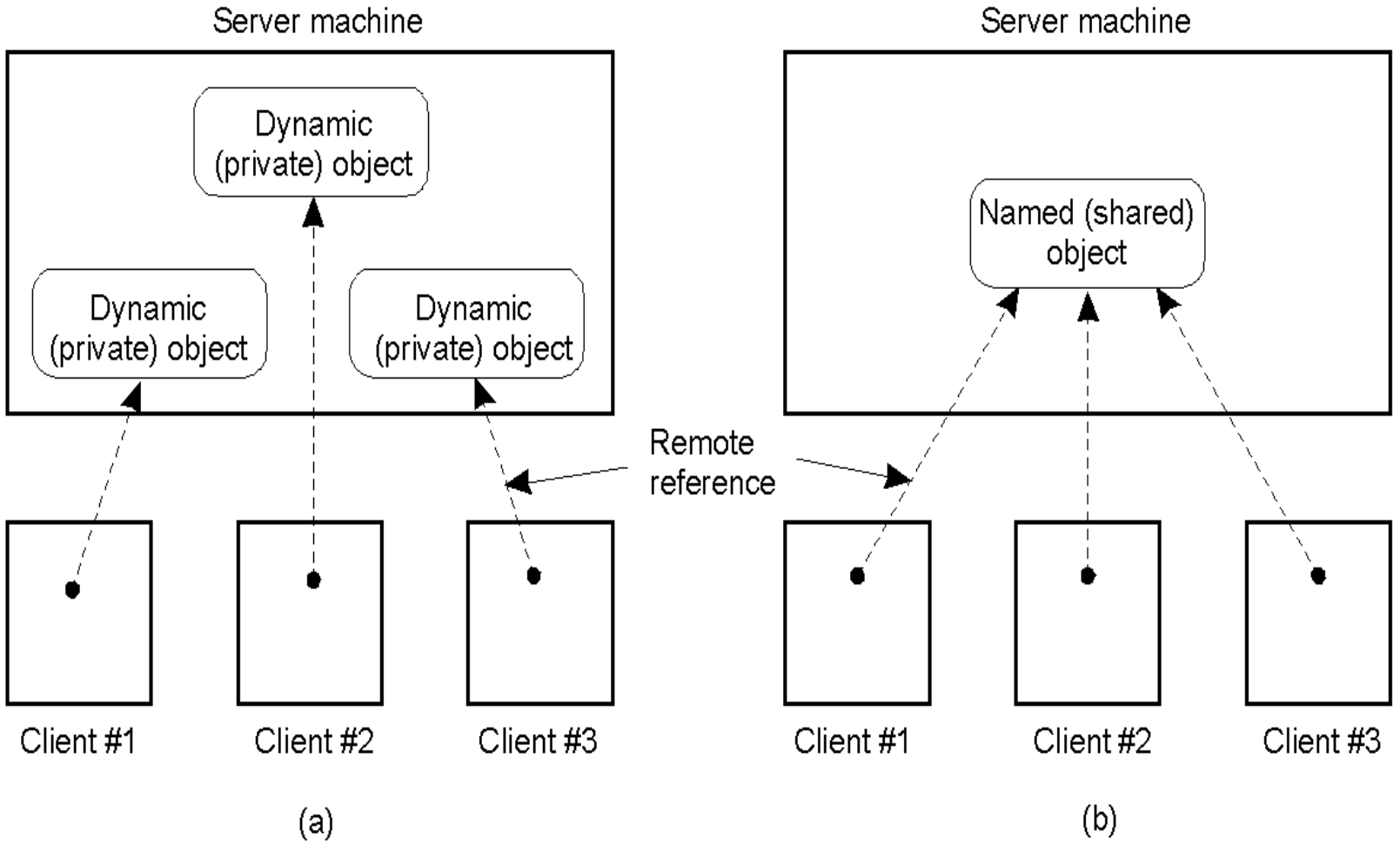


Δύο τύποι αντικειμένων:

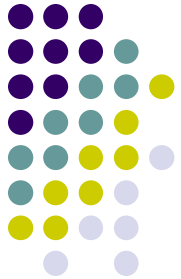
(α) **Δυναμικά (ιδιωτικά) αντικείμενα**: προσπελάσιμα μόνο από τον πελάτη για τον οποίο δημιουργήθηκαν. Για να δημιουργηθεί ένα τέτοιο αντικείμενο, ένας πελάτης πρέπει να υποβάλλει αίτηση στο διακομιστή. Κάθε κλάση δυναμικών αντικειμένων διαθέτει μια διαδικασία create, η οποία καλείται με μια συνηθισμένη RPC.

(β) **Επώνυμα (κοινόχρηστα) αντικείμενα**: δημιουργούνται από ένα διακομιστή με σκοπό να μοιράζονται μεταξύ πολλών πελατών. Καταχωρίζονται σε υπηρεσία καταλόγου ώστε ένας πελάτης να μπορεί να τα αναζητήσει και να συνδυαστεί με αυτά.

Απομακρυσμένα αντικείμενα στο DCE



Απομακρυσμένα αντικείμενα στο DCE



- **Κάθε κλήση απομακρυσμένου αντικειμένου γίνεται με κλήση απομακρυσμένης διαδικασίας. Όταν ένας πελάτης καλεί μία μέθοδο ενός αντικειμένου, μεταβιβάζει στο διακομιστή το ID του αντικειμένου, το ID της διασύνδεσης που περιέχει τη μέθοδο, το ID της μεθόδου και τις παραμέτρους.**
- **Ο διακομιστής τηρεί πίνακα αντικειμένων.**

JAVA RMI

Το RMI αποτελεί ένα εργαλείο ανάπτυξης κατανεμημένων εφαρμογών οι οποίες υλοποιούνται ως προγράμματα Java, τα οποία εκτελούνται σε διαφορετικές JVM.

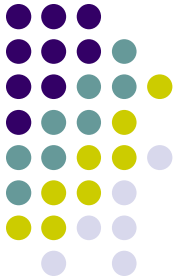
Το Java RMI

- **επιτρέπει σε εφαρμογές την κλήση απομακρυσμένων μεθόδων**
- **Απαιτείται η ύπαρξη ενός RMI client και ενός RMI server.**
- **Παρέχει το μηχανισμό επικοινωνίας ανάμεσα στον Client και στον Server**
- **επιτρέπει σε οποιοδήποτε αντικείμενο της Java να χρησιμοποιηθεί**

Υποστηρίζεται με τα πακέτα
java.rmi, java.rmi.server, java.rmi.registry

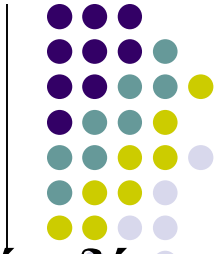
Κατά την εκτέλεση μιας απομακρυσμένης διαδικασίας, το java.rmi κωδικοποιεί τις παραμέτρους κλήσης και τις στέλνει στο server. Αυτός τις αποκωδικοποιεί και καλεί τη μέθοδο. Κωδικοποιεί τα αποτελέσματα και τα στέλνει πίσω στον client. Το java.rmi στον client αποκωδικοποιεί την απάντηση

Εφαρμογές Client & Server

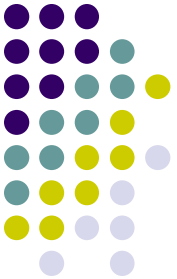


- **Μια τυπική εφαρμογή server δημιουργεί:**
  - **απομακρυσμένα αντικείμενα**
  - **αναφορές σε αυτά ώστε να είναι προσβάσιμα**
  - **περιμένει τους πελάτες να καλέσουν μεθόδους πάνω σε αυτά.**
- **Μια τυπική εφαρμογή client**
  - **καλεί μεθόδους πάνω σε απομακρυσμένα αντικείμενα μέσω μιας απομακρυσμένης αναφοράς.**

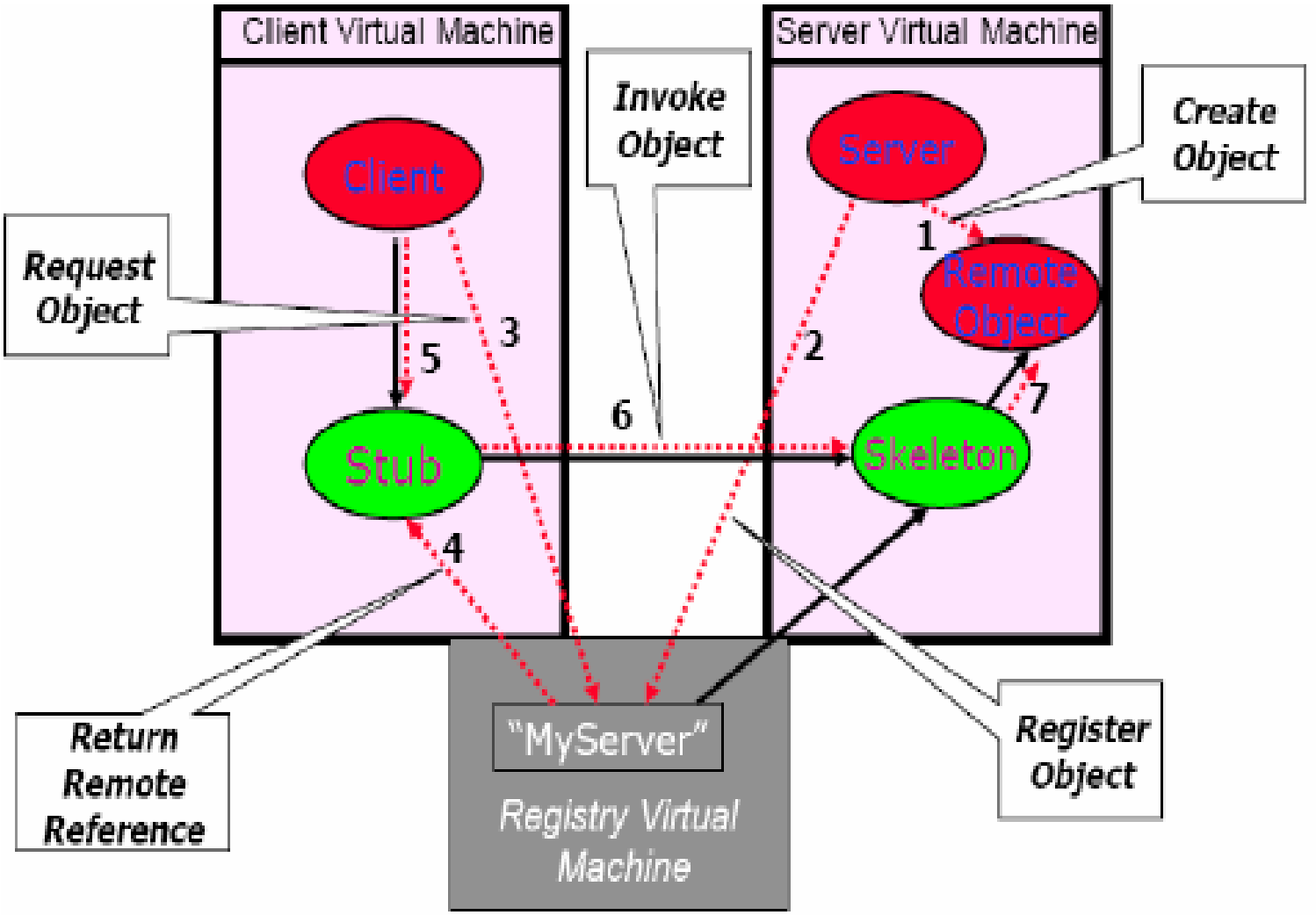
Εύρεση Απομακρυσμένων Αντικειμένων – Χρήση του Registry



- **Το RMI προσφέρει ένα απλό σχήμα ονομασίας, το οποίο δίνει ένα όνομα σε ένα απομακρυσμένο αντικείμενο όταν τρέχει για πρώτη φορά.**
- **Στη συνέχεια καταχωρείται στο RMI registry με μια διαδικασία εγγραφής.**
- **Το registry συνδέει το όνομα του αντικειμένου (όχι το όνομα της κλάσης) και το ίδιο το αντικείμενο.**
- **Στη Java, όταν ένα απομακρυσμένο αντικείμενο εγγράφεται στο registry μίας συγκεκριμένης μηχανής, συνδέεται με ένα αντικείμενο ονοματοδοσίας.**
- **Αν ένα πελάτης θέλει να χρησιμοποιήσει ένα αντικείμενο, το οποίο βρίσκεται σε έναν απομακρυσμένο κόμβο Α**
  - **κάνει μία αναζήτηση στο Registry του Α.**
  - **Χρησιμοποιεί το αποτέλεσμα της αναζήτησης για να συνδεθεί με το απομακρυσμένο αντικείμενο, και να παρεμβάλλει τις μεθόδους του.**

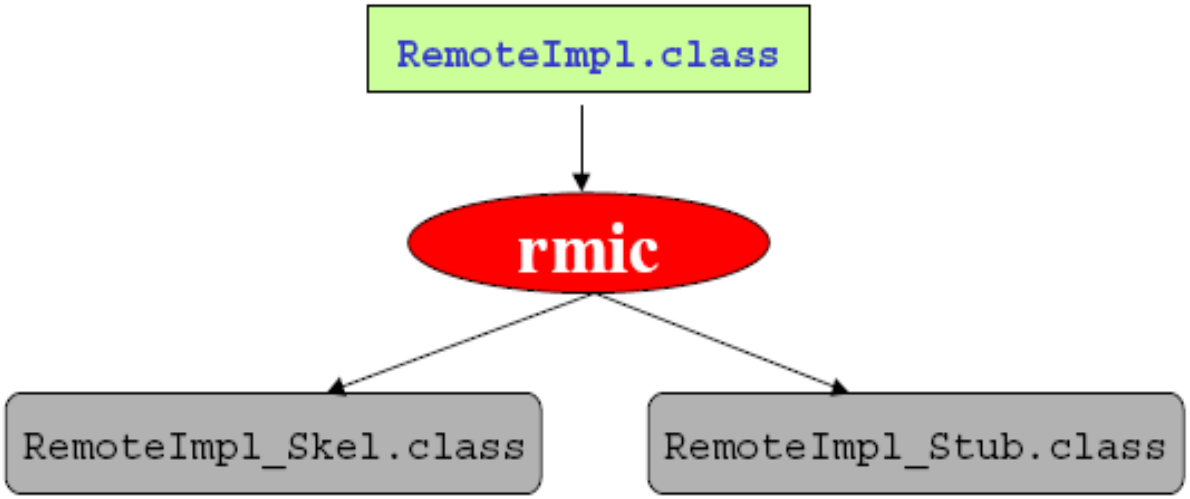


- **Ο server δημιουργεί ένα απομακρυσμένο αντικείμενο και το καταχωρεί στο registry**
- **Ο client το ζητά από το registry**
- **Το registry επιστρέφει μια απομακρυσμένη αναφορά (και δημιουργείται το stub)**
- **Ο client καλεί τη μέθοδο του stub**
- **Το stub μιλά με το skeleton**
- **Το skeleton καλεί την μέθοδο του απομακρυσμένου αντικειμένου**

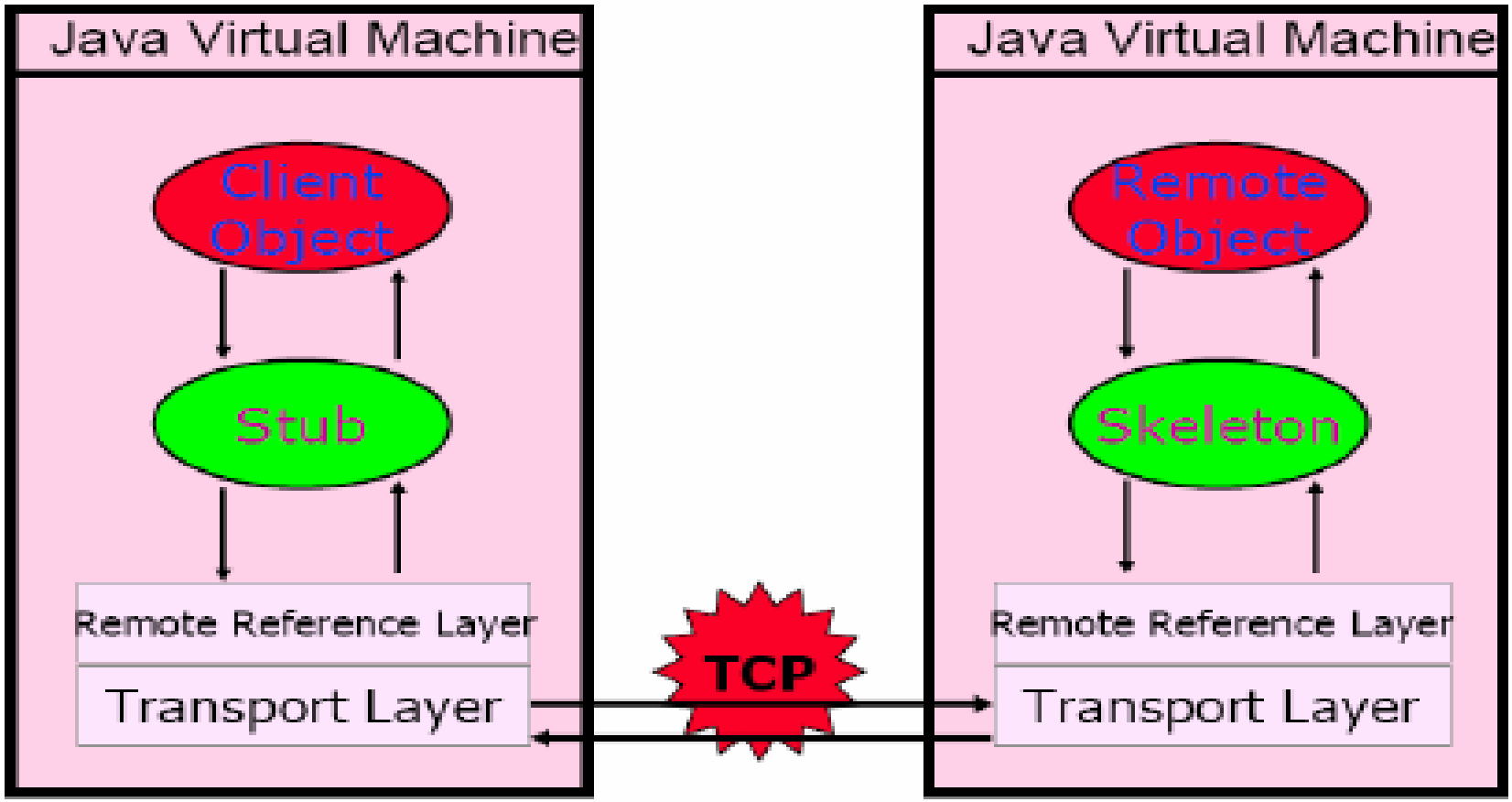

Client Virtual Machine
Server Virtual Machine
Invoke Object
Client
Server
Create Object
Request Object
1
Remote Object
3
5
2
6
7
Stub
Skeleton
4
"MyServer"
Registry Virtual Machine
Return Remote Reference
Register Object

Stub & Skeleton Generation

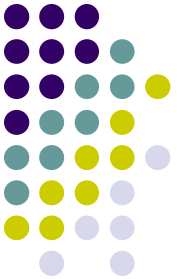
- Client Stubs & Server Skeletons are generated by the `rmic` compiler.
- The `rmic` compiler takes as input a class implementing remote interfaces and outputs a Stub and a Skeleton class



RMI


Java Virtual Machine
Client Object
Stub
Remote Reference Layer
Transport Layer
TCP
Java Virtual Machine
Remote Object
Skeleton
Remote Reference Layer
Transport Layer

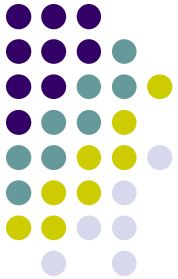
Λειτουργία του stub



Κατά την επίκληση μίας μεθόδου του stub, γίνονται τα παρακάτω:

- **Εκκίνηση μίας σύνδεσης με την απομακρυσμένη JVM που περιέχει το απομακρυσμένο αντικείμενο.**
- **Εγγραφή και μετάδοση των παραμέτρων στην απομακρυσμένη JVM.**
- **Αναμονή του αποτελέσματος της απομακρυσμένης επίκλησης της μεθόδου.**
- **Ανάγνωση της απάντησης ή της εξαίρεσης που επιστρέφεται από την απομακρυσμένη επίκληση.**
- **Επιστροφή της τιμής στον καλούντα.**

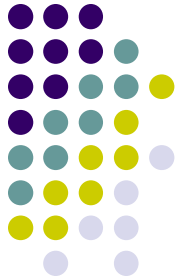
Λειτουργία του skeleton



Όταν το skeleton λαμβάνει ένα μήνυμα κάνει τα εξής:

- **Διαβάζει τις παραμέτρους για την απομακρυσμένη μέθοδο**
- **Κάνει επίκληση της μεθόδου στο πραγματικό απομακρυσμένο αντικείμενο**
- **Γράφει και μεταδίδει το αποτέλεσμα (τιμή ή εξαίρεση) στον πελάτη.**

Διαδικασία Μεταγλώτισης και Εκτέλεσης Client και Server



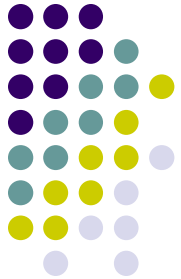
Για τον εξυπηρέτη:

- **Μεταγλώτισση του εξυπηρέτη:**
  - **javac server.java**
- **Δημιουργία των Skeleton και Stub:**
  - **rmic server**
- **Εκκίνηση του registry:**
  - **rmiregistry &**
- **Εκτέλεση του εξυπηρέτη:**
  - **java server *hostname***

Για τον πελάτη:

- **Μεταγλώτισση του πελάτη**
  - **javac client.java**
- **Εκτέλεση του πελάτη και καθορισμό της θέσης του stub**
  - **java -Djava.rmi.server.codebase = http://*hostname*/~*username*/*directory*/ *clientclass hostname method parameters***

Blocking αντικειμένων

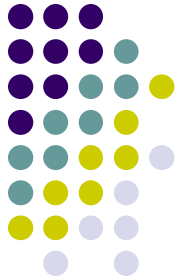


Η Java επιτρέπει την κατασκευή ενός αντικειμένου ως ελεγκτή (monitor) με τη δήλωση μιας μεθόδου ως συγχρονισμένης (synchronized).

Αν δύο εργασίες καλέσουν ταυτόχρονα μια συγχρονισμένη μέθοδο, μόνο η μία μπορεί να προχωρήσει ενώ η άλλη θα μπλοκαριστεί.

Πώς υλοποιείται το μπλοκάρισμα απομακρυσμένου αντικειμένου?

Blocking απομακρυσμένων αντικειμένων



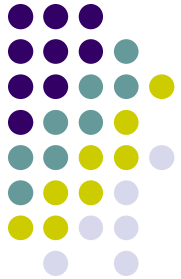
Προσέγγιση 1: Μπλοκάρονται οι πελάτες στο στέλεχος πελάτη που υλοποιεί τη διασύνδεση αντικειμένου.

- **Η λύση αυτή απαιτεί το συγχρονισμό διαφορετικών πελατών σε διαφορετικά μηχανήματα.**
- **Ο κατανεμημένος συγχρονισμός είναι αρκετά πολύπλοκος.**

Προσέγγιση 2: Το μπλοκάρισμα επιτρέπεται μόνο στο διακομιστή.

- **Προκύπτουν προβλήματα σε περίπτωση που ένας πελάτης καταρρεύσει ενώ ο διακομιστής χειρίζεται την κλήση του.**

Java RMI: Blocking απομακρυσμένων αντικειμένων



- **Το μπλοκάρισμα απομακρυσμένων αντικειμένων περιορίζεται μόνο στους διαμεσολαβητές.**
- **Τα απομακρυσμένα αντικείμενα δε προστατεύονται από την ταυτόχρονη προσπέλαση από διεργασίες που εκτελούνται σε διαφορετικούς διαμεσολαβητές με τη χρήση συγχρονισμένων μεθόδων.**
- **Χρησιμοποιούνται τεχνικές ρητών κατανεμημένων κλειδωμάτων**