

The background features a light gray gradient with abstract geometric elements. On the left, a network diagram consists of dark gray nodes connected by thin lines. Scattered across the right side are various thin-lined triangles and small dots, creating a sense of depth and complexity.

06

Κατανεμημένα Συστήματα

Επικοινωνία Αίτησης – Απάντησης
(Μέρος Β: Βλάβες) – RPC



02

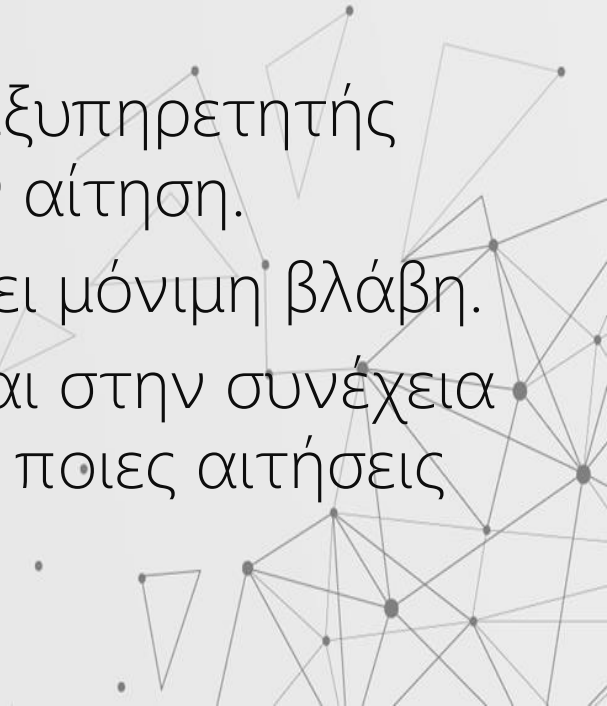
ΒΛΑΒΕΣ ΠΕΛΑΤΗ -
ΕΞΥΠΗΡΕΤΗΤΗ

Βλάβη εξυπηρετητή

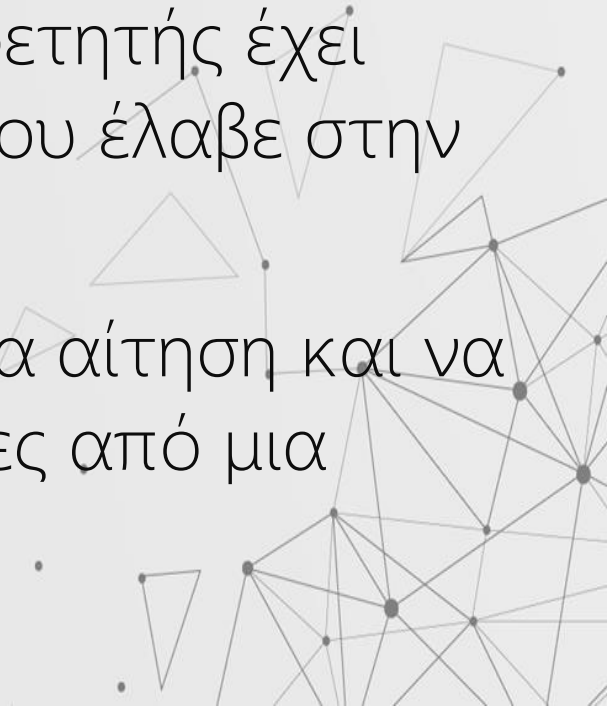
- Όσο ο εξυπηρετητής λειτουργεί χωρίς πρόβλημα, κάποια στιγμή θα λάβει την αίτηση του πελάτη, θα την επεξεργαστεί, και θα στείλει την απάντηση
- Τι γίνεται αν ο εξυπηρετητής παρουσιάσει βλάβη;
- Υπάρχει περίπτωση μια ή περισσότερες αιτήσεις που έχει στείλει ο πελάτης να μείνουν χωρίς απάντηση.
- Τι μπορεί να κάνει ο πελάτης;
- 3 σενάρια:
 - At most once,
 - At least once,
 - Exactly once.



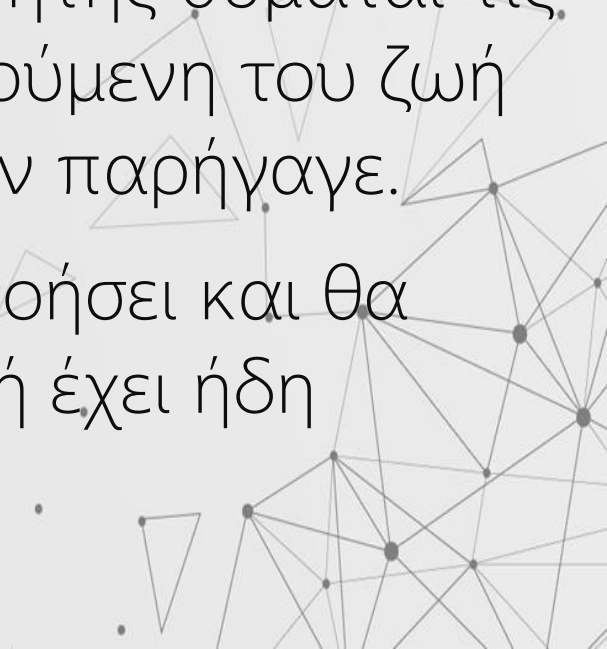
At most once

- Ο πελάτης προσπαθεί να στείλει την αίτηση στον εξυπηρετητή, μια ή περισσότερες φορές ώσπου:
 - να λάβει μία απάντηση (αποτέλεσμα), ή
 - να λήξει η αναμονή του (η αίτησή του μπορεί να έχει εκτελεστεί μία ή καμία φορά).
 - Αν μετά από έναν αριθμό προσπαθειών ο πελάτης δεν λάβει απάντηση, εγκαταλείπει.
 - Ο πελάτης δεν γνωρίζει κατά πόσο ο εξυπηρετητής έλαβε ή κατάφερε να επεξεργαστεί την αίτηση.
 - Ο εξυπηρετητής μπορεί να παρουσιάσει μόνιμη βλάβη.
 - Διαφορετικά, αν παρουσιάσει βλάβη και στην συνέχεια γίνει επανεκκίνηση, πρέπει να θυμάται ποιες αιτήσεις είχε λάβει στην προηγούμενη του ζωή.
- 

At least once

- Ο πελάτης προσπαθεί να στείλει την αίτηση στον εξυπηρετητή, μέχρι να λάβει απάντηση.
 - Υποθέτει πως ακόμα κι αν ο εξυπηρετητής παρουσιάσει βλάβη, τελικά θα γίνει επανεκκίνηση και θα ανακάμψει.
 - Αν μετά την επανεκκίνηση ο εξυπηρετητής έχει αμνησία, δεν θυμάται τις αιτήσεις που έλαβε στην προηγούμενη του ζωή.
 - Υπάρχει περίπτωση να λάβει την ίδια αίτηση και να την επεξεργαστεί ξανά (περισσότερες από μια φορές).
- 

Exactly once

- Ο πελάτης προσπαθεί να στείλει την αίτηση στον εξυπηρετητή, μέχρι να λάβει απάντηση.
 - Υποθέτει πως ακόμα κι αν ο εξυπηρετητής παρουσιάσει βλάβη, τελικά θα γίνει επανεκκίνηση και θα ανακάμψει.
 - Μετά την επανεκκίνηση, ο εξυπηρετητής θυμάται τις αιτήσεις που είχε λάβει στην προηγούμενη του ζωή καθώς και τις απαντήσεις που τυχόν παρήγαγε.
 - Αν λάβει την ίδια αίτηση θα την αγνοήσει και θα στείλει πίσω την απάντηση (αν αυτή έχει ήδη παραχθεί).
- 

Χειρισμός στον πελάτη/εξυπηρετητή

	Πελάτης		Εξυπηρετητής	
	προσπάθειες	μπλοκάρισμα*	χειρισμός διπλοτύπων**	αποθήκευση απαντήσεων
at most once	περιορισμένες	ΟΧΙ	ΝΑΙ	ΠΡΟΑΙΡΕΤΙΚΑ
at least once	απεριόριστες (όσες χρειαστεί)	ΝΑΙ	ΟΧΙ	ΟΧΙ
exactly once	απεριόριστες (όσες χρειαστεί)	ΝΑΙ	ΝΑΙ	ΝΑΙ

(*) μέχρι να παραληφθεί απάντηση

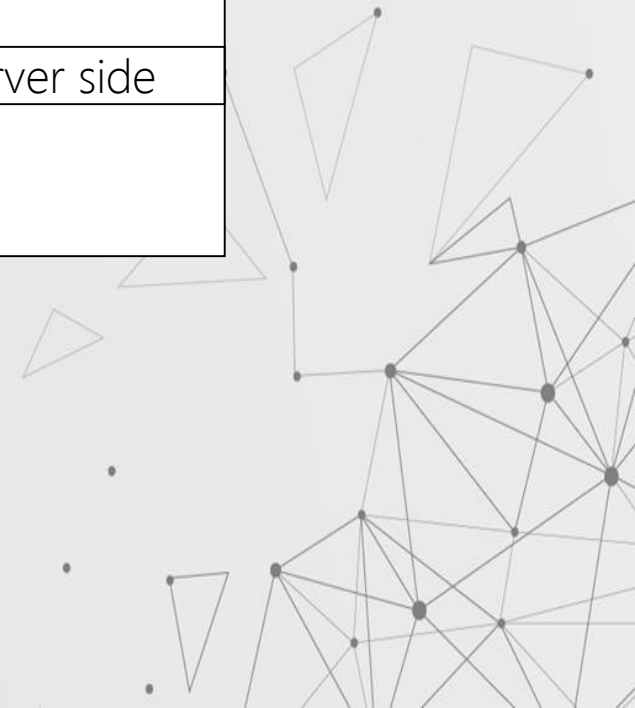
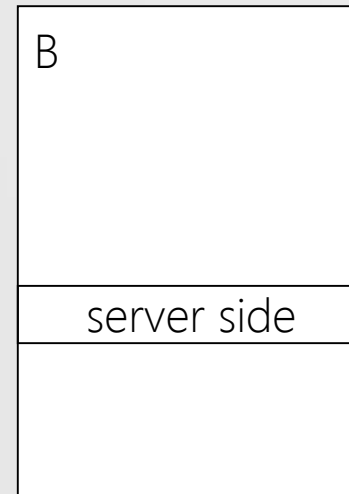
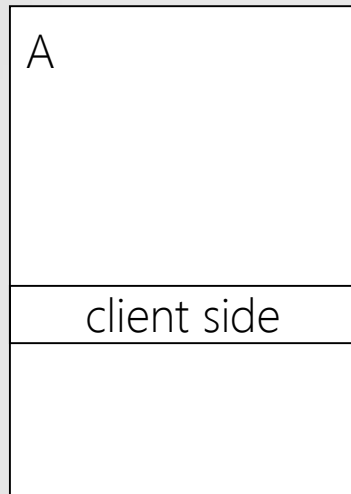
(**) λόγω επανειλημμένης αποστολής αιτήσεων από τον πελάτη.

Βλάβη πελάτη

- Οι αιτήσεις του πελάτη δεν έχουν πλέον νόημα.
 - ακόμα και αν το μηχάνημα του πελάτη ανακάμψει αργότερα
- Οι «ορφανές» αιτήσεις πρέπει να αγνοηθούν.
- Οι ενέργειες που πρέπει να κάνει ο εξυπηρετητής εξαρτώνται από το στάδιο που βρίσκεται η αίτηση.
- Αν ο εξυπηρετητής δεν έχει αρχίσει την επεξεργασία, τότε η αίτηση απλά μπορεί να αγνοηθεί.
- Αν ο εξυπηρετητής έχει αρχίσει αλλά δεν έχει ολοκληρώσει την επεξεργασία της αίτησης, η επεξεργασία πρέπει να ακυρωθεί / τερματιστεί.
- Αν ο εξυπηρετητής έχει ολοκληρώσει την επεξεργασία, τότε η (αποθηκευμένη) απάντηση πρέπει να αγνοηθεί.

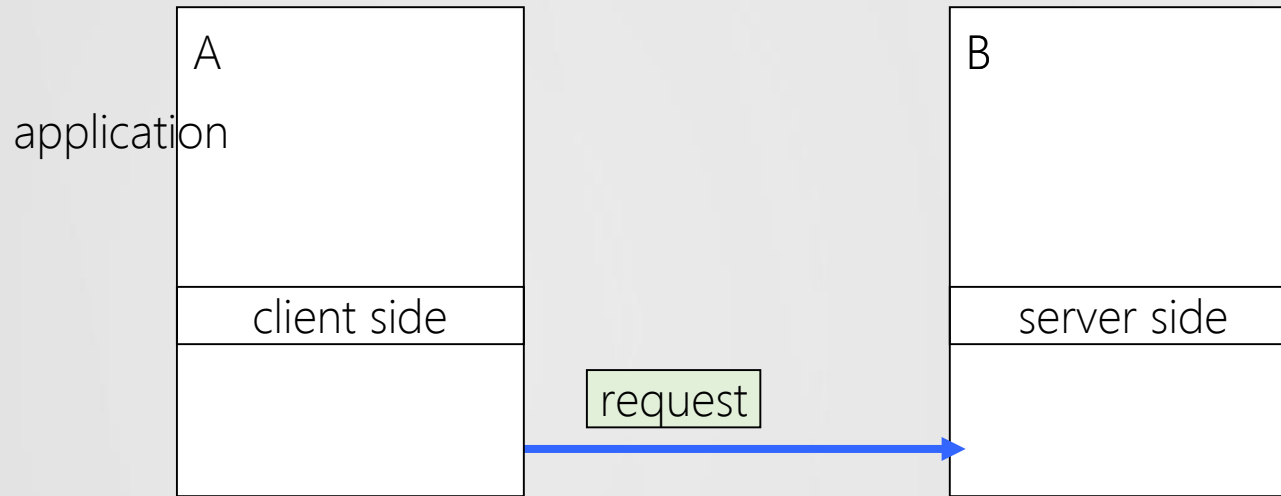
Βλάβη πελάτη – Παράδειγμα (1)

αρχική κατάσταση



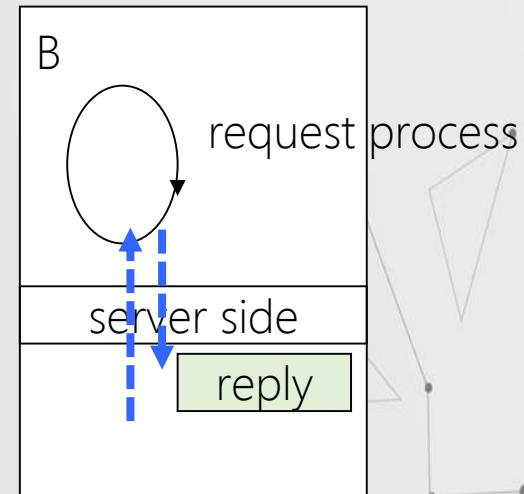
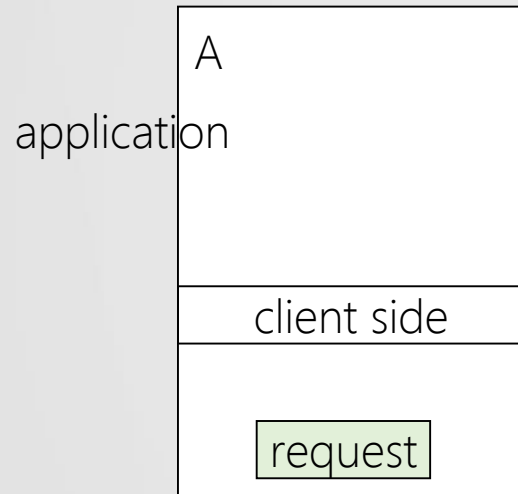
Βλάβη πελάτη – Παράδειγμα (2)

αποστολή αίτησης
εφαρμογής πελάτη



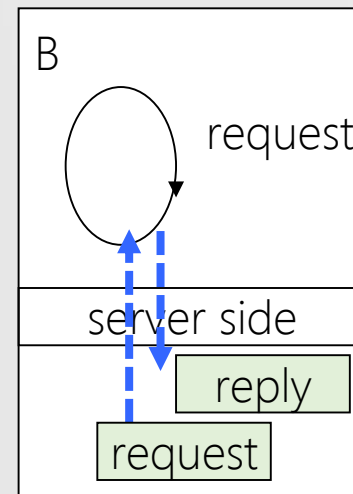
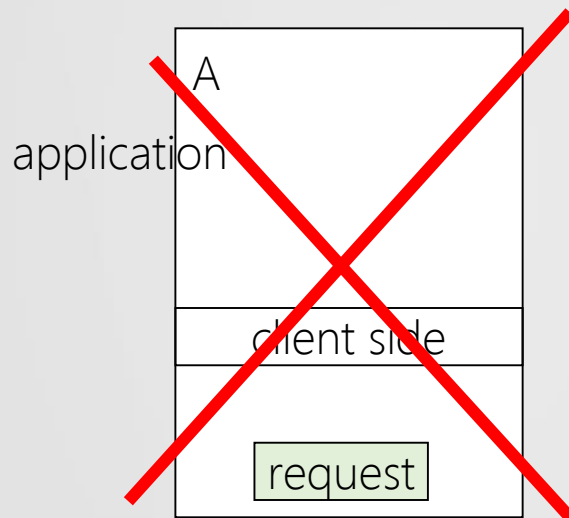
Βλάβη πελάτη – Παράδειγμα (3)

επεξεργασία αίτησης
στον εξυπηρετητή



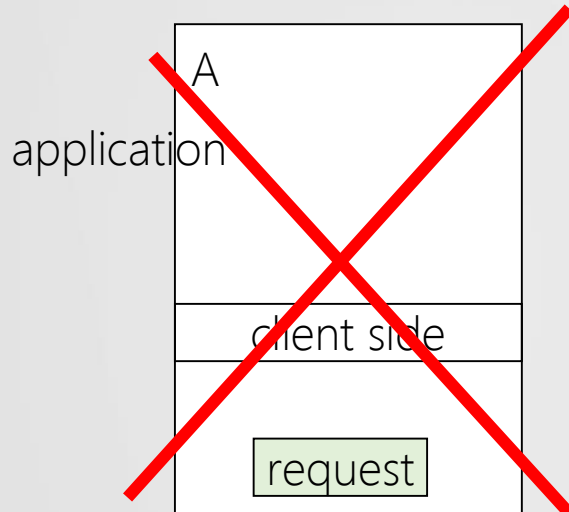
Βλάβη πελάτη – Παράδειγμα (4)

βλάβη πελάτη

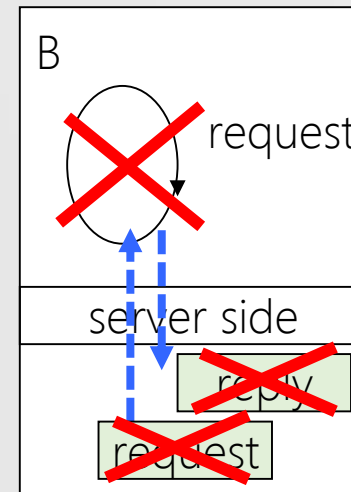


Βλάβη πελάτη – Παράδειγμα (5)

βλάβη πελάτη



ignore

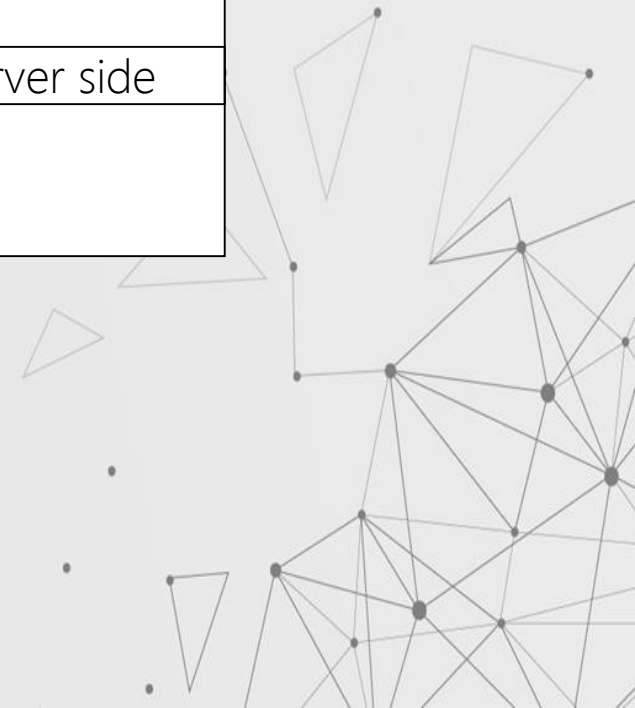
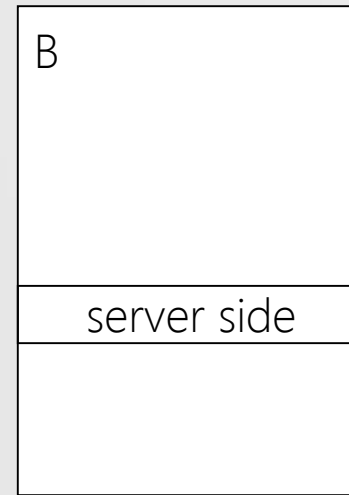
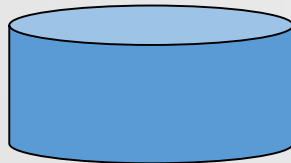
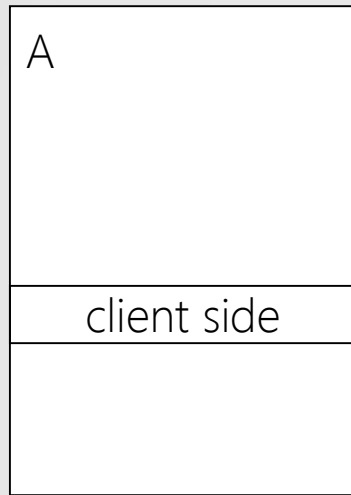


Εξολόθρευση

- Βλάβη πελάτη με επανεκκίνηση & μόνιμη μνήμη.
- Ο πελάτης καταγράφει σε μόνιμη μνήμη τα στοιχεία κάθε νέας αίτησης (+ την διεύθυνση του εξυπηρετητή).
- Τα στοιχεία διαγράφονται από την μόνιμη μνήμη όταν ο πελάτης παραλάβει το αποτέλεσμα της αίτησης.
- Όταν ο πελάτης ανακάμψει από την βλάβη, διαβάσει τα στοιχεία των (εκκρεμών) αιτήσεων από την αποθήκη.
- Για κάθε τέτοια αίτηση, στέλνει αίτηση ακύρωσης στον αντίστοιχο εξυπηρετητή.
- Όταν λάβει επιβεβαίωση από τον εξυπηρετητή, απομακρύνει την εγγραφή από την μόνιμη μνήμη.

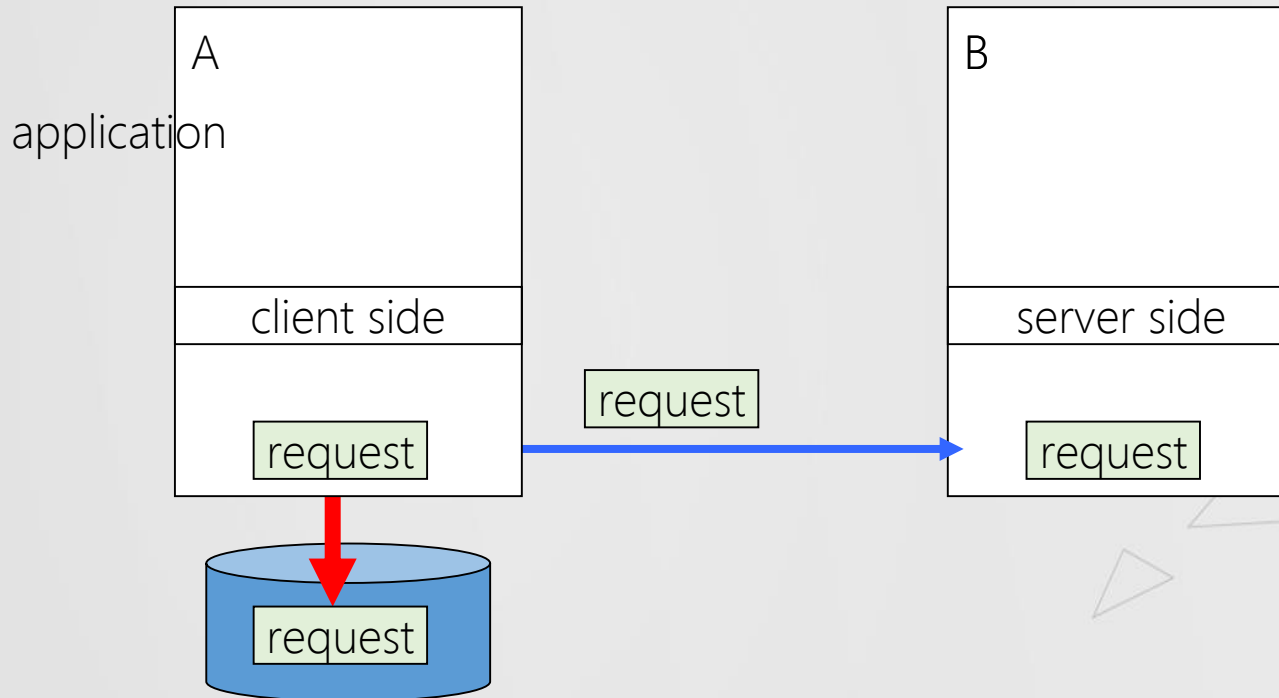
Εξολόθρευση – Παράδειγμα (1)

αρχική κατάσταση

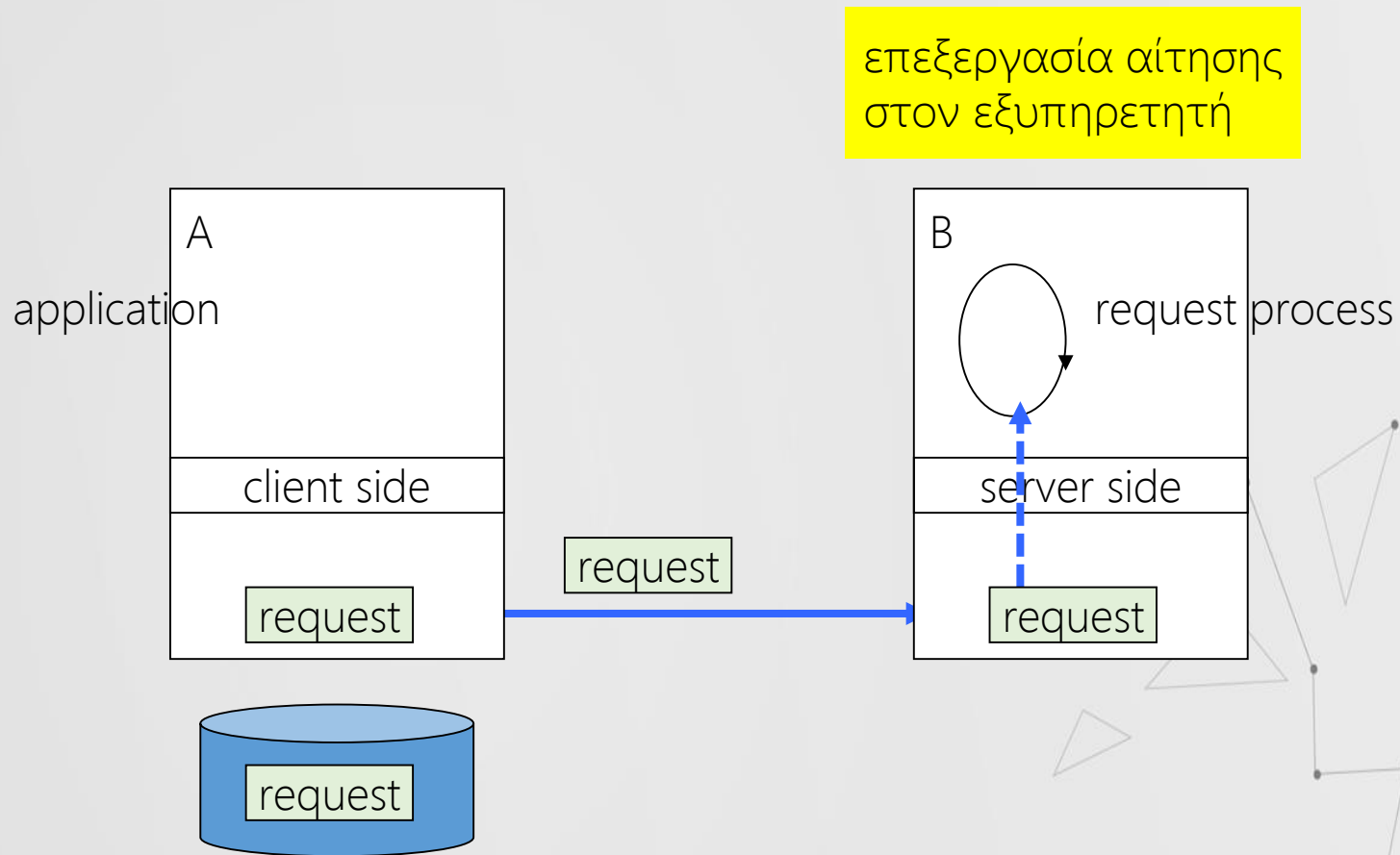


Εξολόθρευση – Παράδειγμα (2)

αποστολή αίτησης
εφαρμογής πελάτη

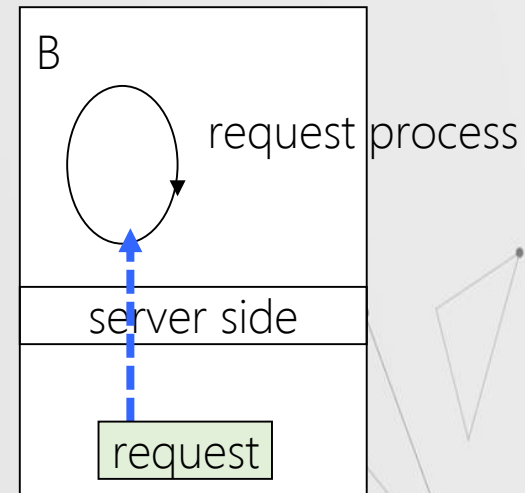
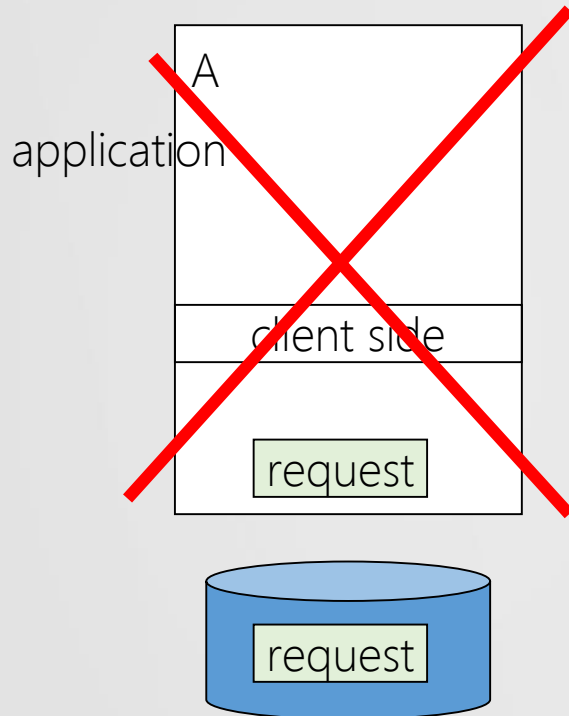


Εξολόθρευση – Παράδειγμα (3)



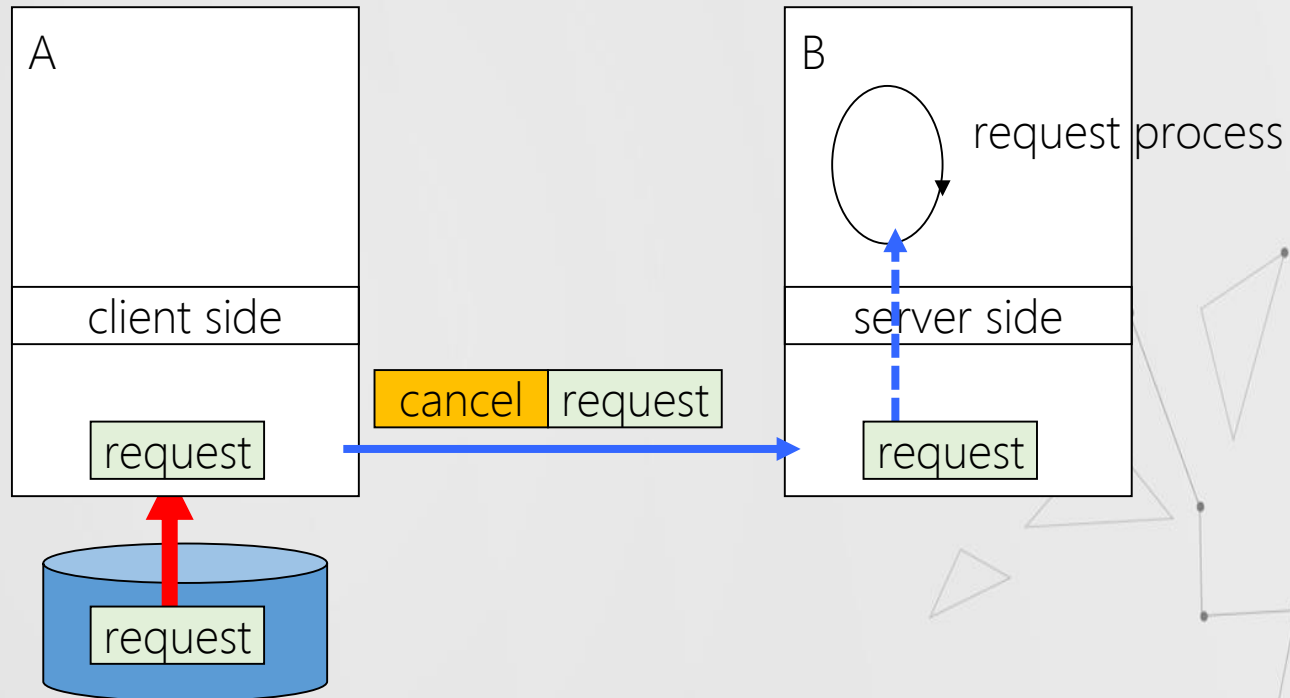
Εξολόθρευση – Παράδειγμα (4)

βλάβη πελάτη



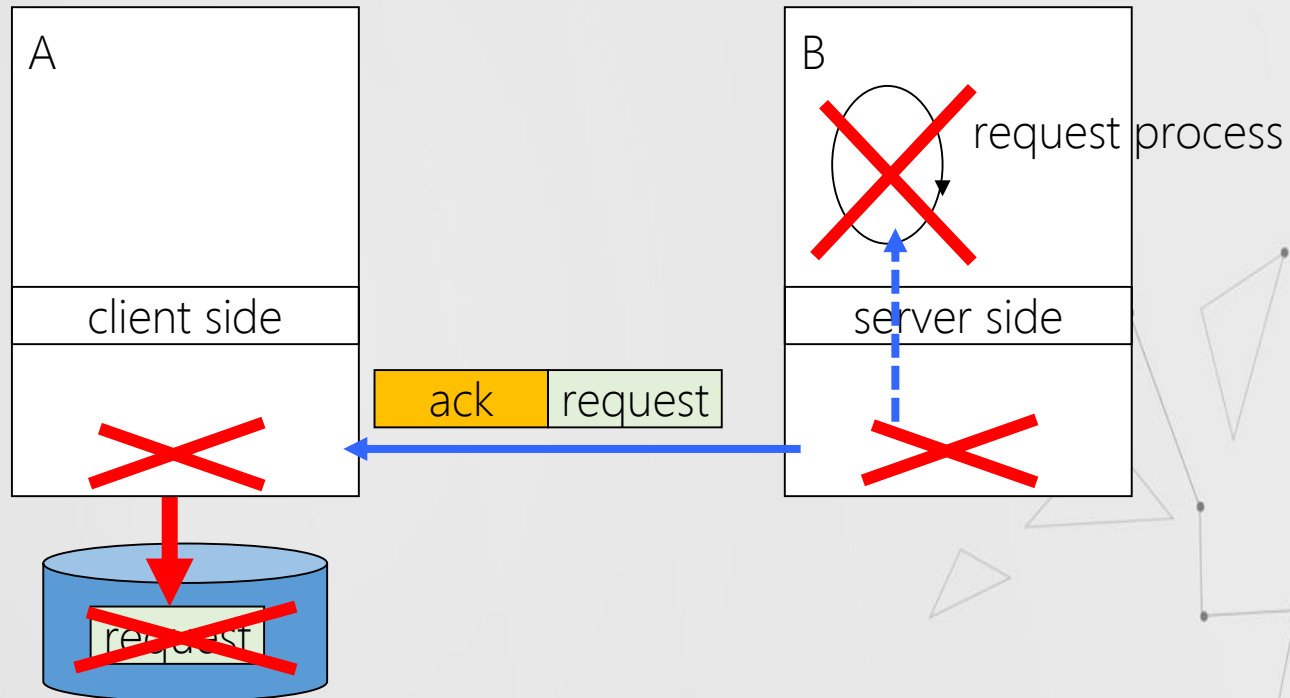
Εξολόθρευση – Παράδειγμα (5)

ανάκαμψη πελάτη &
ειδοποίηση εξυπηρετητή

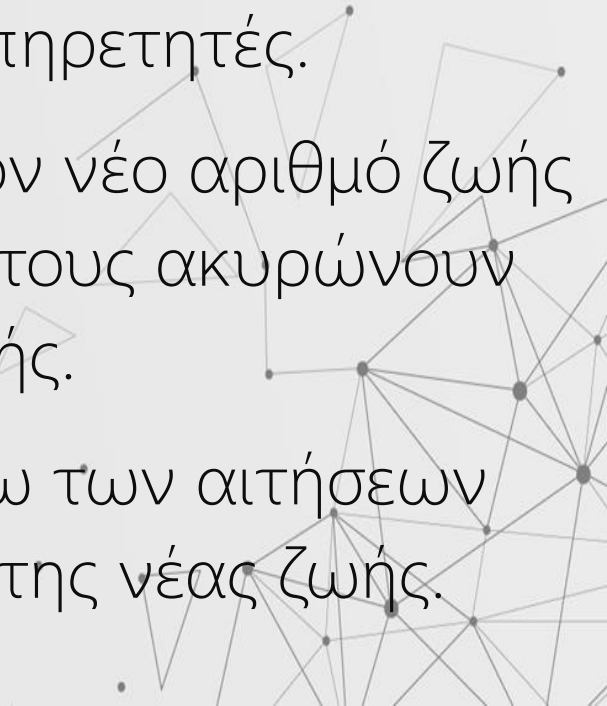


Εξολόθρευση – Παράδειγμα (6)

ακύρωση αίτησης
στον εξυπηρετητή

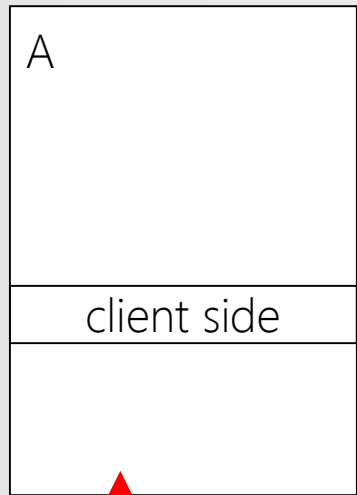


Αναβίωση

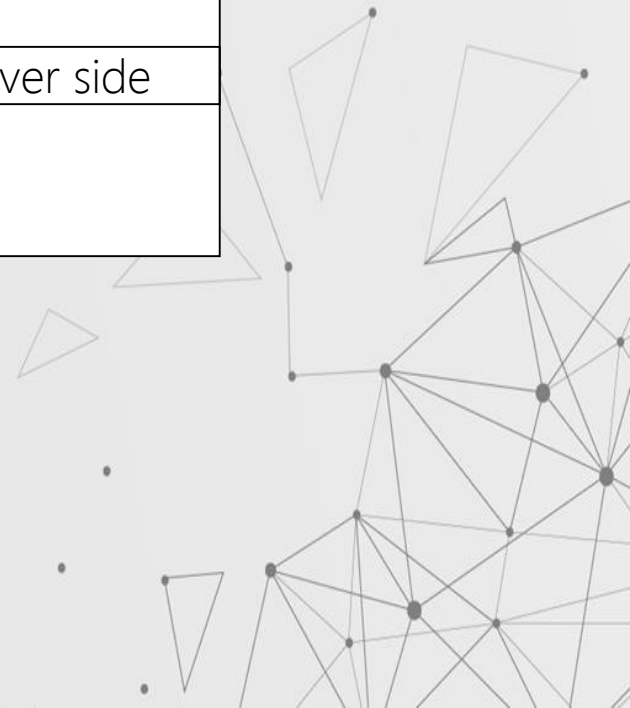
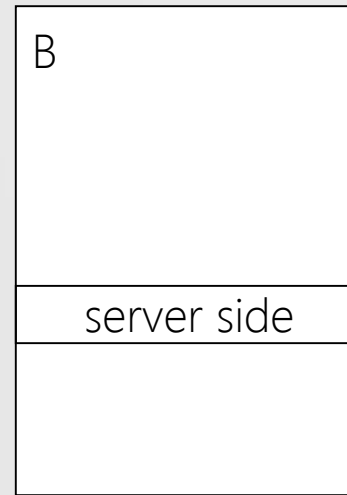
- Βλάβη πελάτη με επανεκκίνηση & (μερική) αμνησία.
 - Κάθε φορά που γίνεται επανεκκίνηση, αυξάνεται αυτόματα ο αριθμός ζωής (σε μόνιμη μνήμη).
 - εναλλακτικά, χρησιμοποιείται η μέρα/ώρα εκκίνησης.
 - Ο πελάτης επισυνάπτει τον τρέχοντα αριθμός ζωής σε κάθε αίτηση που στέλνει στους εξυπηρετητές.
 - Όταν ο πελάτης ανακάμψει, στέλνει τον νέο αριθμό ζωής στους εξυπηρετητές που με την σειρά τους ακυρώνουν τις αιτήσεις με διαφορετικό αριθμό ζωής.
 - Αυτό μπορεί να γίνει «τεμπέλικά», μέσω των αιτήσεων που θα στείλει ο πελάτης στο πλαίσιο της νέας ζωής.
- 

Αναβίωση – Παράδειγμα (1)

αρχική κατάσταση

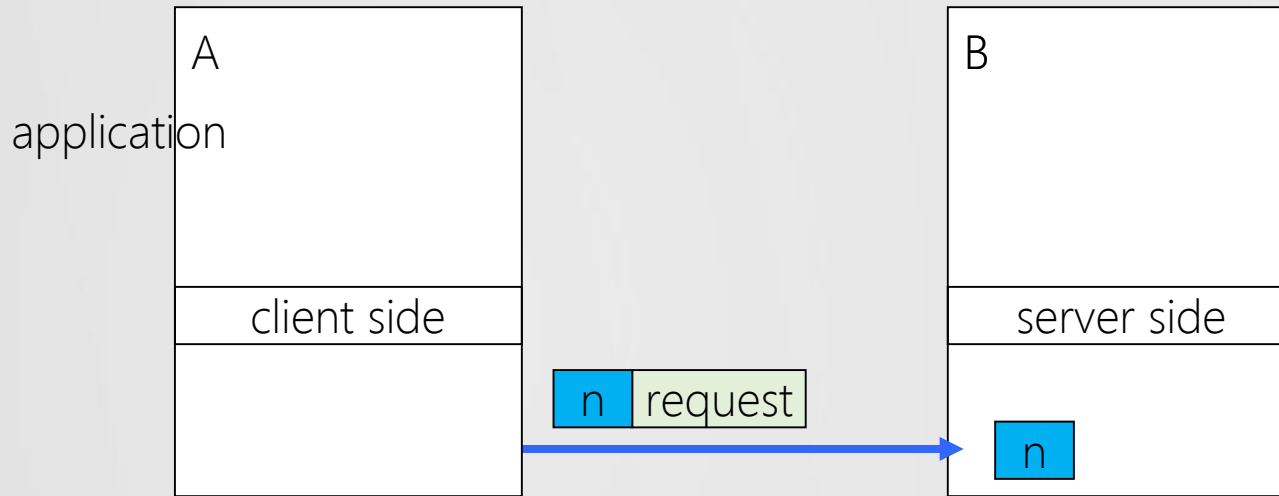


incarnation
number



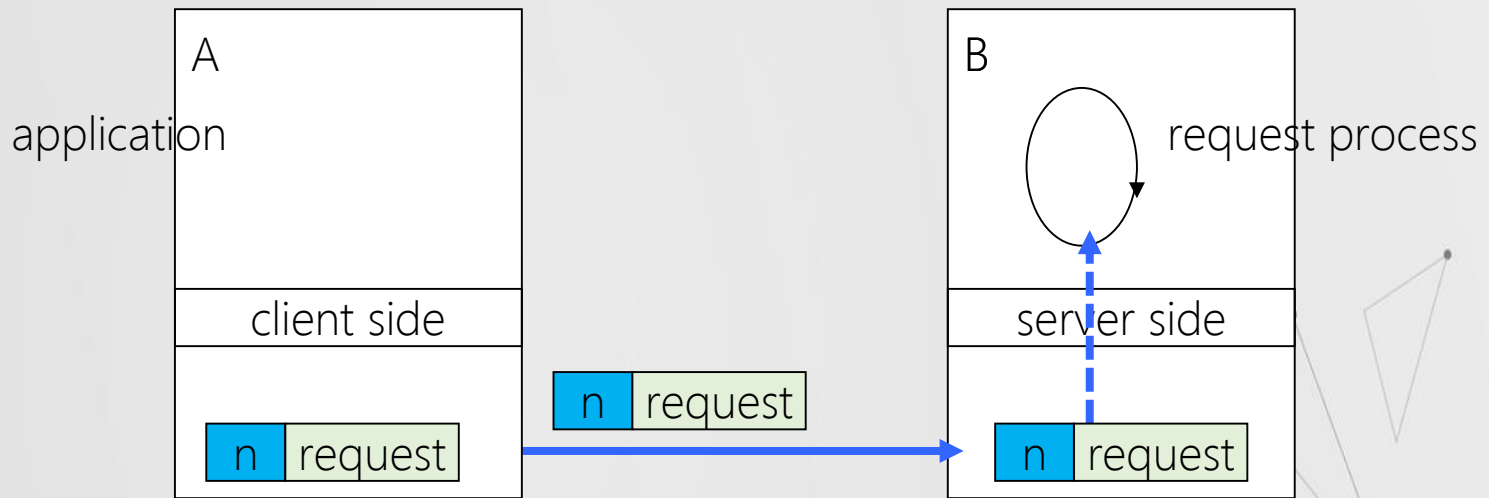
Αναβίωση – Παράδειγμα (2)

αποστολή αίτησης
εφαρμογής πελάτη



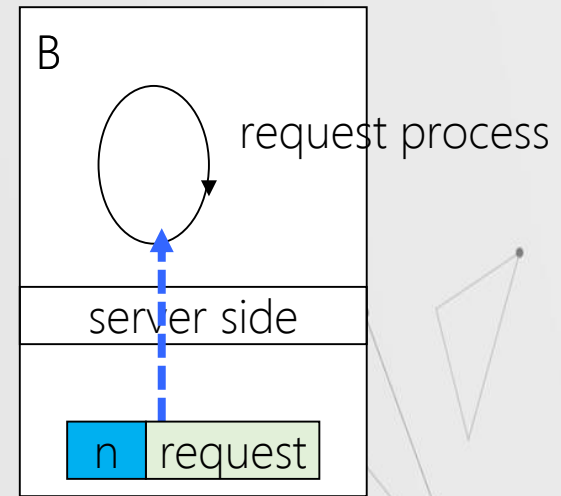
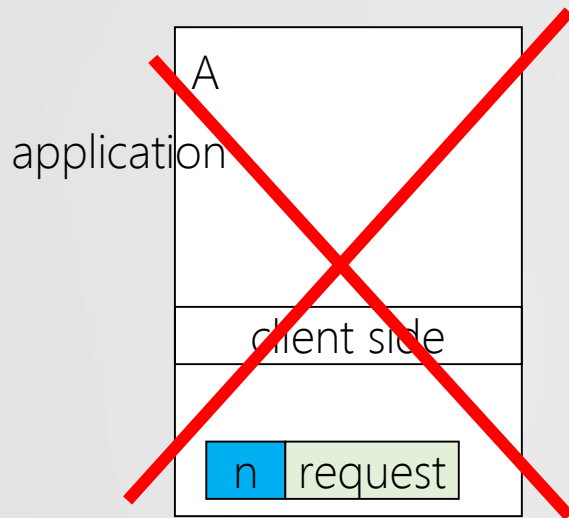
Αναβίωση – Παράδειγμα (3)

επεξεργασία αίτησης
στον εξυπηρετητή



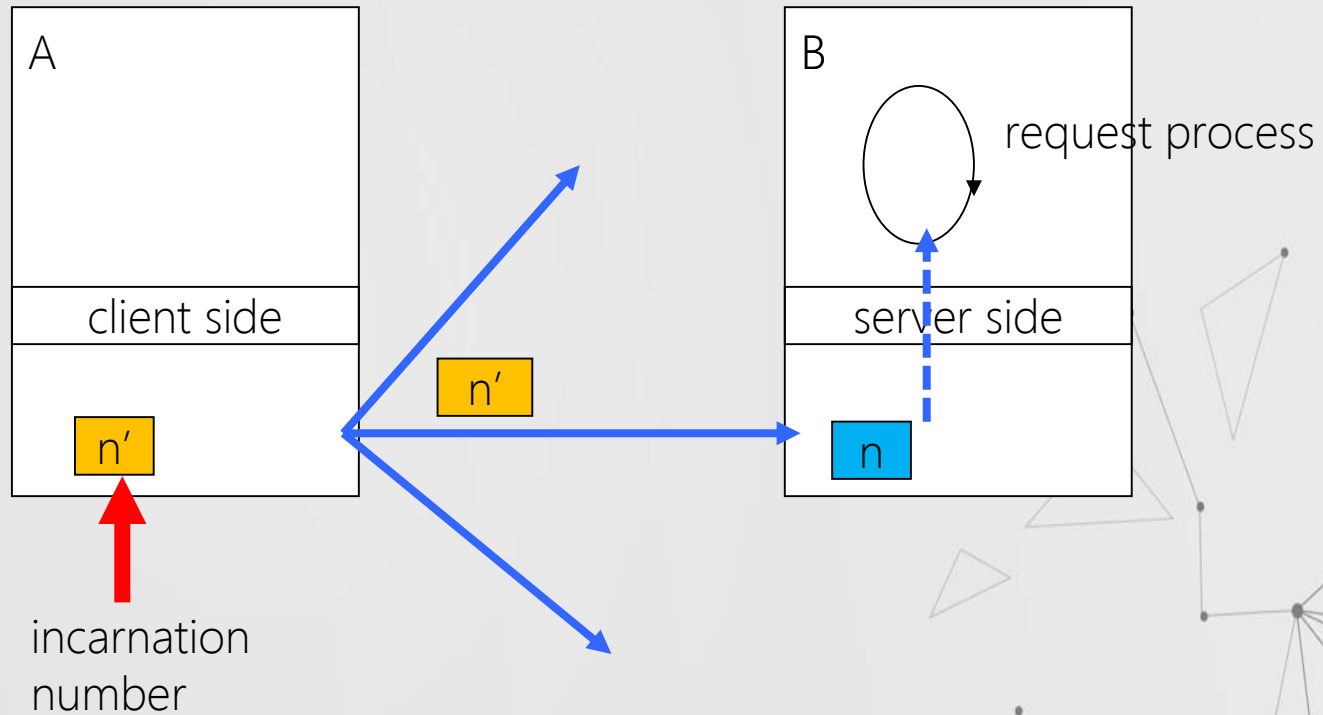
Αναβίωση – Παράδειγμα (4)

βλάβη πελάτη



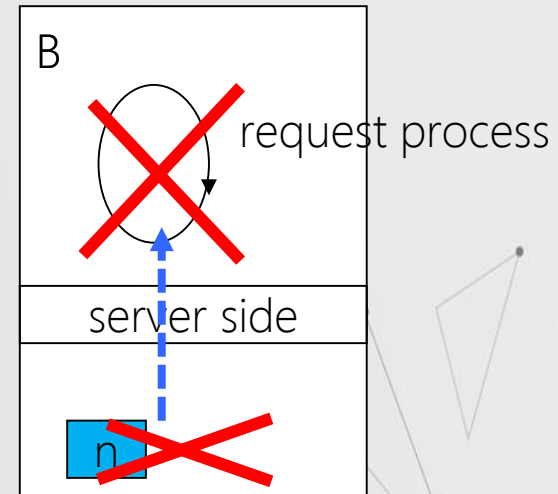
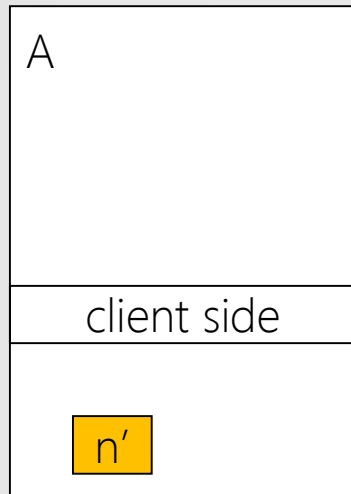
Αναβίωση – Παράδειγμα (5)

ανάκαμψη πελάτη &
ειδοποίηση εξυπηρετητών



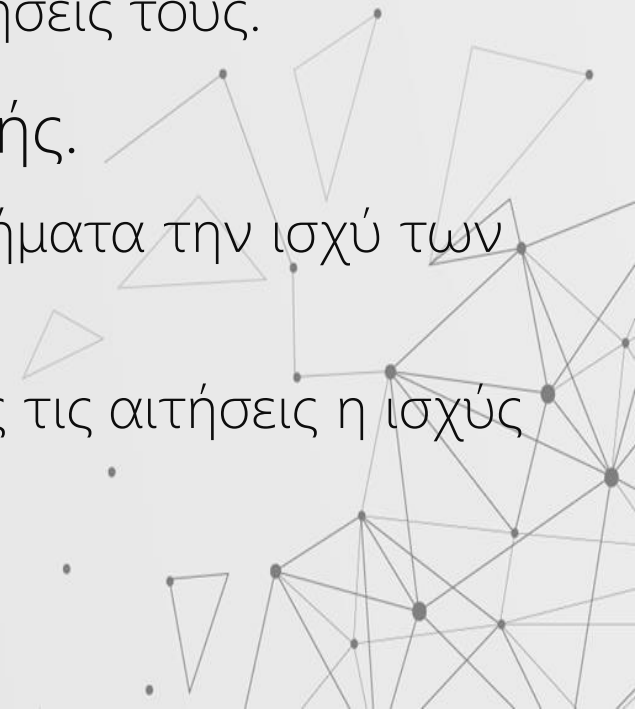
Αναβίωση – Παράδειγμα (6)

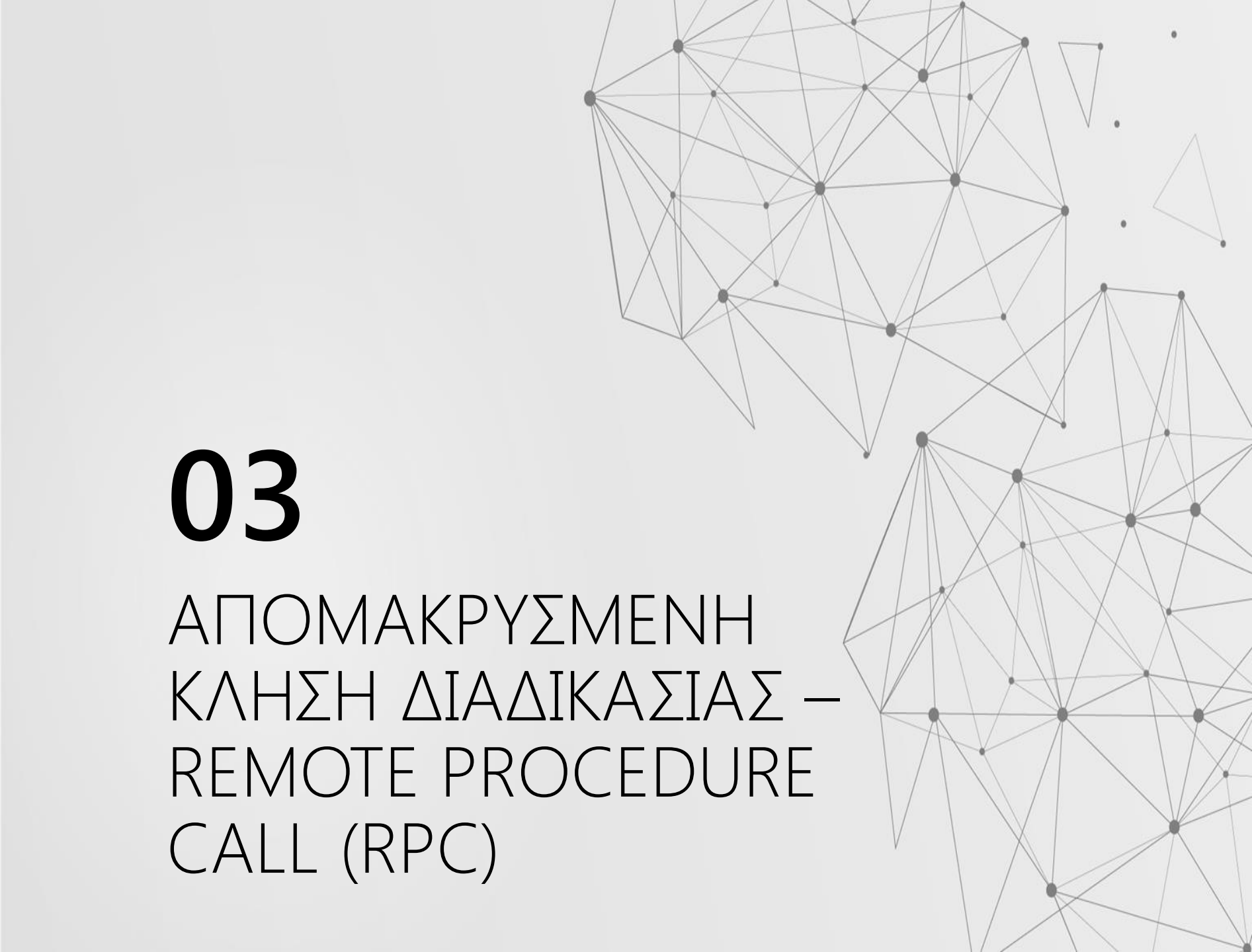
ακύρωση αίτησης
στον εξυπηρετητή



Βλάβη πελάτη χωρίς επανεκκίνηση

- Η αρμοδιότητα ακύρωσης ορφανών αιτήσεων ανήκει αποκλειστικά στον εξυπηρετητή.
- Ρητός έλεγχος λειτουργίας πελάτη.
 - Ο εξυπηρετητής ελέγχει περιοδικά την λειτουργία των πελατών που έχουν εκκρεμείς αιτήσεις,
 - Αν δεν λάβει απάντηση, ακυρώνει τις αιτήσεις τους.
- Αιτήσεις με περιορισμένη διάρκεια ζωής.
 - Ο πελάτης «ανανεώνει» ανά τακτά διαστήματα την ισχύ των αιτήσεων που έχει ήδη στείλει.
 - Ο εξυπηρετητής ακυρώνει αυτόματα όλες τις αιτήσεις η ισχύς των οποίων έχει «λήξει».





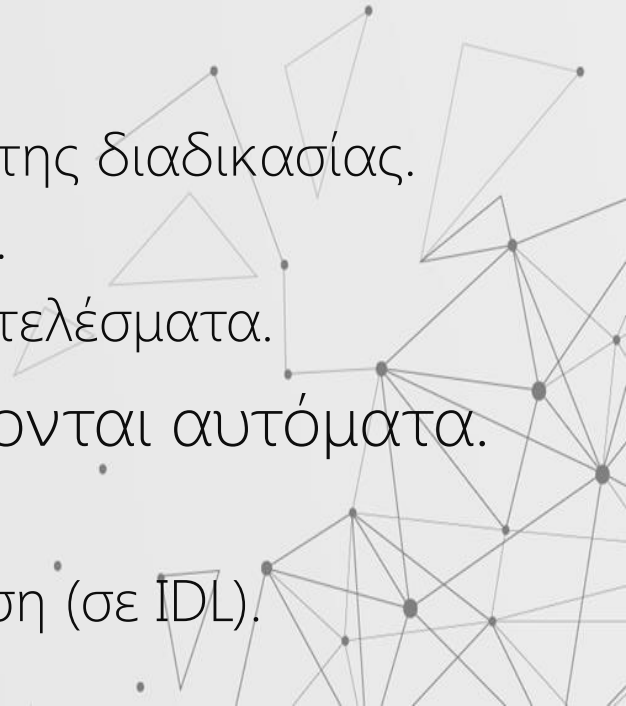
03

ΑΠΟΜΑΚΡΥΣΜΕΝΗ
ΚΛΗΣΗ ΔΙΑΔΙΚΑΣΙΑΣ –
REMOTE PROCEDURE
CALL (RPC)

Απομακρυσμένη κλήση διαδικασίας

- Αλληλεπίδραση client-server προσαρμοσμένη ή ενσωματωμένη σε γλώσσα/περιβάλλον προγραμματισμού.
- Η διασύνδεση προγραμματισμού εφαρμογής έχει την μορφή συμβατικών κλήσεων συνάρτησης.
 - που εκτελούνται απομακρυσμένα.
 - απόκρυψη των λεπτομερειών της επικοινωνίας και (απο)σειριοποίησης δεδομένων πάνω από το δίκτυο.
- Οικείο μοντέλο προγραμματισμού.
 - πιο γρήγορη ανάπτυξη λογισμικού με λιγότερα λάθη.
- Στο παρασκήνιο, μια σειρά από λειτουργίες υλοποιούν την επιθυμητή λειτουργικότητα με διαφανή τρόπο.

Δόμηση λογισμικού υπηρεσίας RPC

- RPC runtime:
 - υλοποιεί το πρωτόκολλο επικοινωνίας πάνω από το δίκτυο.
 - Client stub:
 - ανάμεσα στον πελάτη και το RPC runtime.
 - πακετάρει τις παραμέτρους της κλήσης.
 - ξεπακετάρει τα αποτελέσματα.
 - Server stub:
 - ανάμεσα στο RPC runtime και τον κώδικα της διαδικασίας.
 - ξεπακετάρει τις παραμέτρους της κλήσης.
 - καλεί τη διαδικασία και πακετάρει τα αποτελέσματα.
 - Τα client/server stubs μπορεί να παράγονται αυτόματα.
 - μέσω ειδικών εργαλείων (stub generators).
 - με βάση μια συντακτική περιγραφή/δήλωση (σε IDL).
- 

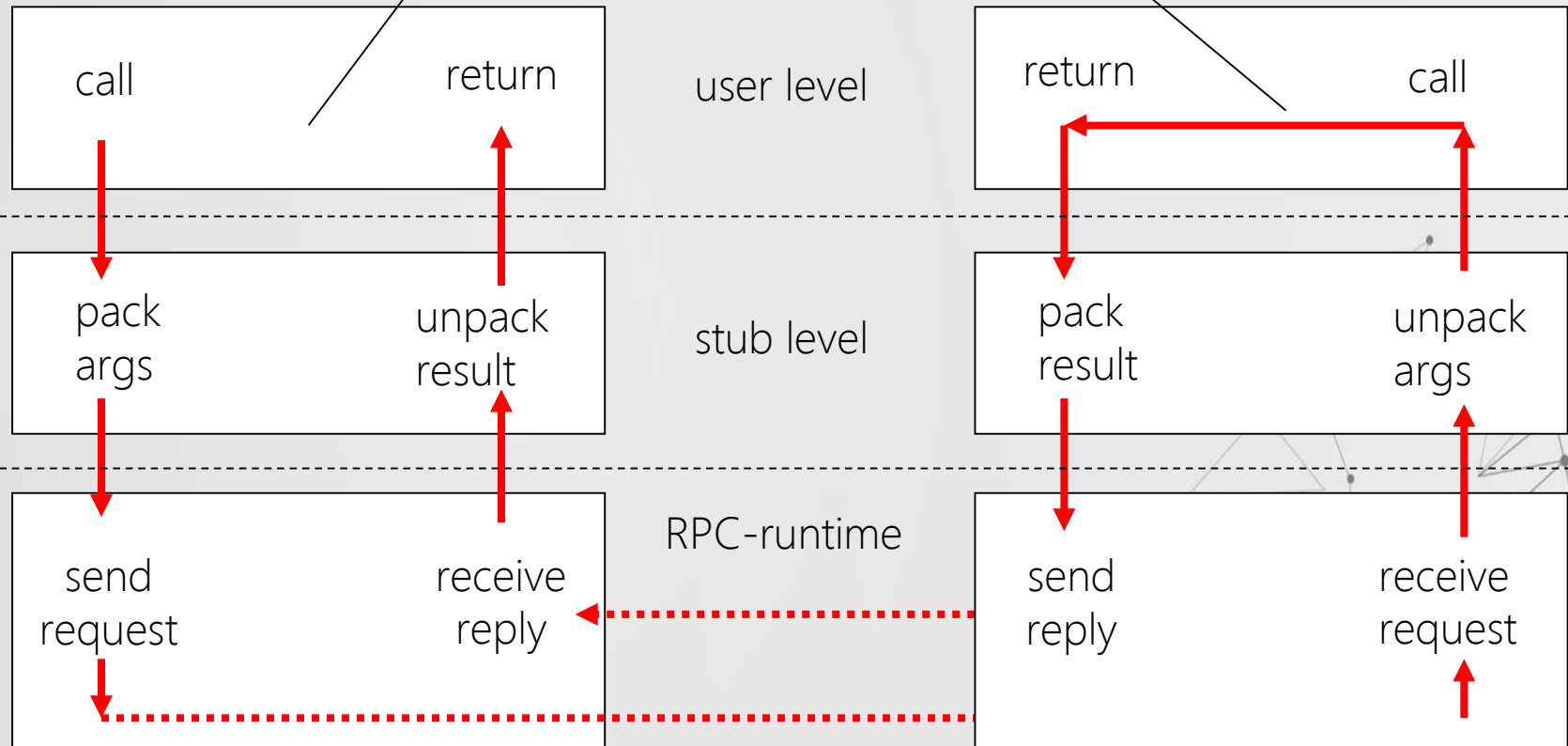
Δόμηση λογισμικού υπηρεσίας RPC (1)

η εφαρμογή έχει την
ψευδαίσθηση ότι κάλεσε
μια **τοπική** συνάρτηση

Client

ο κώδικας που καλείται
είναι μια **συμβατική**
συνάρτηση

Server



Δόμηση λογισμικού υπηρεσίας RPC (2)

μοναδικό αναγνωριστικό
διασύνδεσης, για ανακάλυψη
και έλεγχο συμβατότητας

Uuidgen

περιγραφή διασύνδεσης
ανεξάρτητη γλώσσας
προγραμματισμού

Interface
definition file

IDL compiler

πρόγραμμα
πελάτη

κατασκευή
client stub

κατασκευή
server stub

πρόγραμμα
εξυπηρετητή

Client code

Client stub

Header

Server stub

Server code

#include

#include

C compiler

C compiler

διασύνδεσης
στην γλώσσα
εφαρμογής

C compiler

C compiler

Client
object file

Client stub
object file

Server stub
object file

Server
object file

Linker

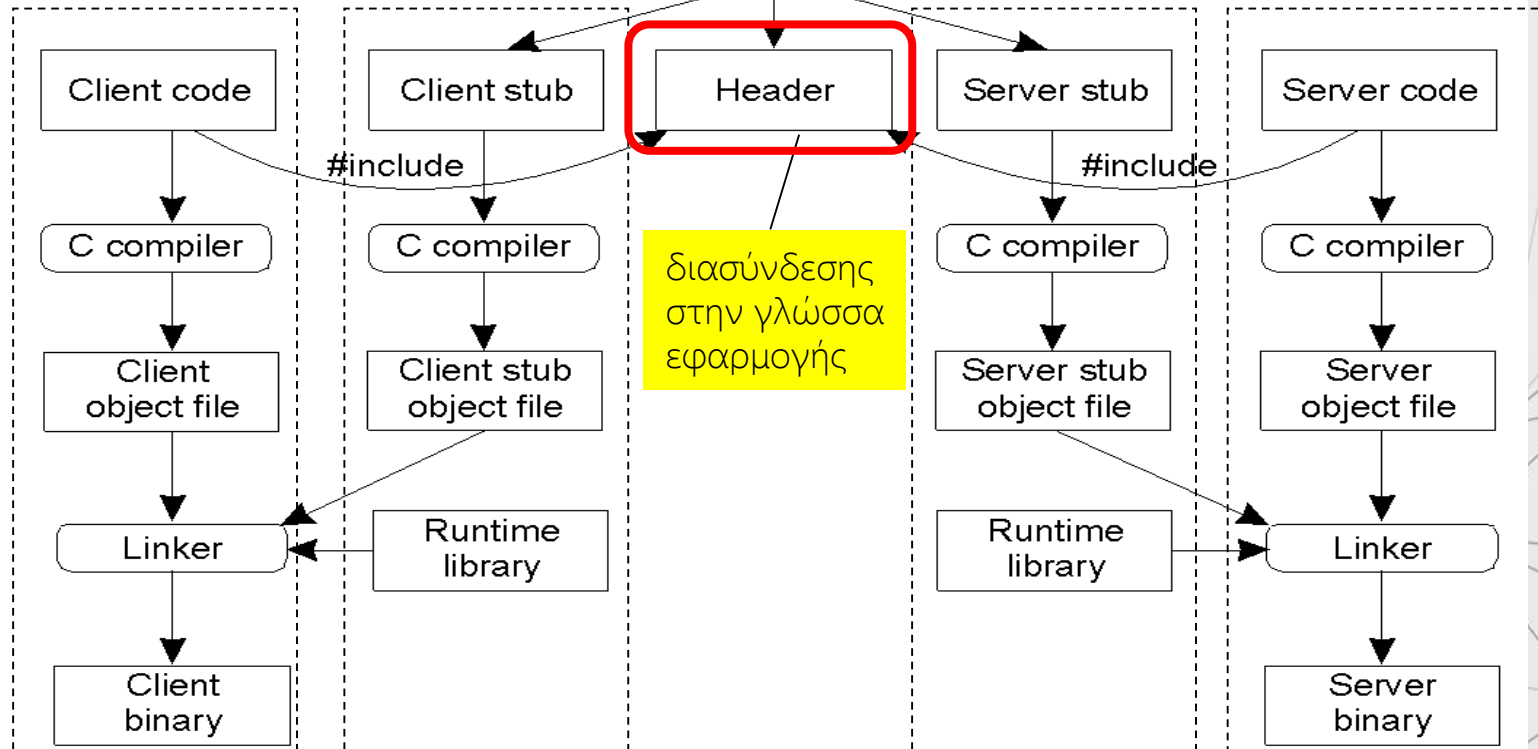
Runtime
library

Runtime
library

Linker

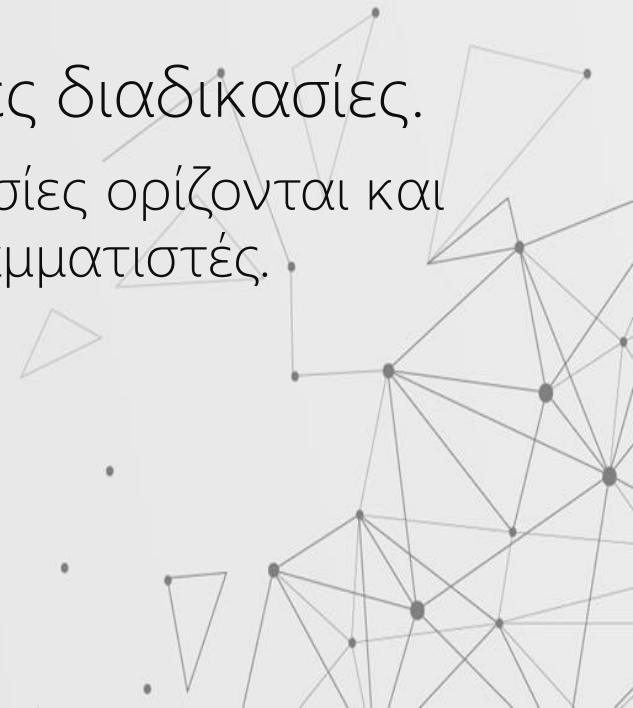
Client
binary

Server
binary



Ζητήματα διαφάνειας

- Σημασιολογία κλήσης.
 - At most once, at least once, exactly once.
- Πέρασμα παραμέτρων & επιστροφή αποτελεσμάτων.
 - κοινή αναπαράσταση / serialization δεδομένων.
- Ανεύρεση του κόμβου του εξυπηρετητή.
 - μηχανισμός/πρωτόκολλο ανακάλυψης.
- Διαχωρισμός ανάμεσα σε διαφορετικές διαδικασίες.
 - ανοιχτό σύστημα όπου διάφορες διαδικασίες ορίζονται και υλοποιούνται από διαφορετικούς προγραμματιστές.
 - αποφυγή «σύγκρουσης» ονομάτων.
- Απόδοση.
 - ελαχιστοποίηση κόστους/καθυστερήσης.
 - ταυτόχρονη/παράλληλη εξυπηρέτηση.



Υποστήριξη για διαφορετικές γλώσσες

- Η εφαρμογή που καλεί την απομακρυσμένη διαδικασία μπορεί να είναι γραμμένη σε διαφορετική γλώσσα από την υλοποίηση της διαδικασίας στον εξυπηρετητή.
 - επιθυμητή ανεξαρτησία ανάπτυξης πελάτη-εξυπηρετητή.
- Για να επιτευχθεί διαλειτουργικότητα, απαιτείται:
 - κοινή γλώσσα προσδιορισμού της διασύνδεσης,
 - κοινό πρωτόκολλο επικοινωνίας πελάτη-εξυπηρετητή,
 - πρότυπα αναπαράστασης / serialization δεδομένων.
- Για κάθε περιβάλλον / γλώσσα προγραμματισμού
 - χρειάζονται μετατροπείς (adapters) για την κατασκευή των client και server stubs που επικοινωνούν με το RPC runtime.
- Σύγχρονη λύση: web-based protocols, SOAP/W3C