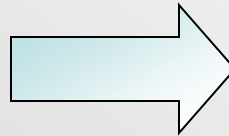


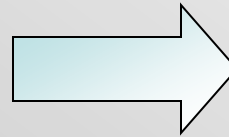
Αντικειμενοστραφής προγραμματισμός

Κλασικές γλώσσες
προγραμματισμού
(FORTRAN, C, Pascal)



Προσανατολισμός
στα δεδομένα

Αντικειμενοστραφείς γλώσσες
προγραμματισμού
(Java, C++)



Προσανατολισμός
στα αντικείμενα

Τί είναι αντικείμενο;

Αντικείμενο: Μία οντότητα λογισμικού η οποία χαρακτηρίζεται από την **κατάσταση** και την **συμπεριφορά** της

Κατάσταση: Μέλη του αντικειμένου (ομοιότητα με τις μεταβλητές των κλασικών γλωσσών)

Συμπεριφορά: Μέθοδοι του αντικειμένου (ομοιότητα με τις συναρτήσεις των κλασικών γλωσσών)

Σχόλιο: Τα αντικείμενα λογισμικού έχουν μεγάλη ομοιότητα με τα αντικείμενα του πραγματικού κόσμου

Επικοινωνία αντικειμένων

Ένα πρόγραμμα αντικειμενοστραφούς προγραμματισμού μπορεί να θεωρηθεί ως ένα σύνολο αντικειμένων τα οποία **συνεργάζονται** μεταξύ τους προκειμένου να λύσουν το συγκεκριμένο πρόβλημα.

Συνεργασία:

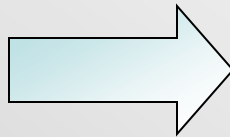
Το κάθε αντικείμενο έχει τη δυνατότητα να μεταβάλλει τη δική του κατάσταση, την κατάσταση των άλλων αντικειμένων, καθώς και να επικοινωνεί με αυτά με μηνύματα έχοντας πρόσβαση στις μεθόδους τους

Σχεδίαση εφαρμογής:

Αποσύνθεση σε αντικείμενα

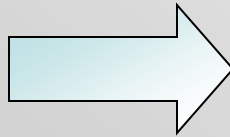
Πλεονεκτήματα αντικειμενοστραφούς προγραμματισμού

Απόκρυψη πληροφορίας
(το κάθε αντικείμενο βλέπει
το άλλο σαν μαύρο κουτί)



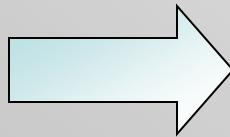
Ευκολότερη συντήρηση, κατανόηση
και αναβάθμιση του κώδικα (modularity)
Επαναχρησιμοποίηση του κώδικα (το
ίδιο αντικείμενο μπορεί να χρησιμοποιηθεί
σε διάφορα προγράμματα χωρίς ο
προγραμματιστής να γνωρίζει την υλοποίη-
σή του)

Αφαιρετικότητα και
εκφραστική δύναμη



Μέσω της χρήσης των αντικειμένων,
η λύση του προβλήματος περιγράφεται
μέσω όρων που συνιστούν το πρόβλημα

Ευελιξία



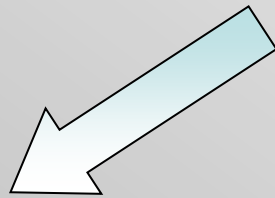
Μέσω της χρήσης της κληρονομικότητας
και του πολυμορφισμού

Επικοινωνία αντικειμένων σε ένα κατανεμημένο περιβάλλον¹

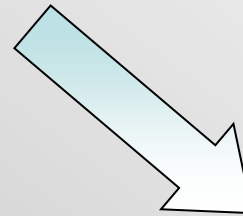
Τα αντικείμενα τα οποία συνθέτουν την εφαρμογή εκτελούνται σε διαφορετικά περιβάλλοντα εκτέλεσης

Το κάθε αντικείμενο έχει ως στόχο του σε μία δεδομένη στιγμή του προγράμματος να εκτελέσει μία συγκεκριμένη εργασία.

Το αντικείμενο αυτό χρειάζεται:



Να έχει τη γνώση (κώδικα)
για το πώς θα εκτελεστεί η εργασία



Να έχει πρόσβαση στους απαραίτητους πόρους
(π.χ αρχεία, βάσεις δεδομένων κλπ)
για την εκτέλεση της εργασίας

Κατανεμημένα αντικείμενα σαν RPC

- Συνήθη συστατικά
 - Γλώσσα περιγραφής διεπαφών (Interface Description Language)
 - Ορίζει την εξωτερική όψη των αντικειμένων
 - Μεταγλωτιστές – Προ-μεταγλωτιστές
 - Επεκτάσεις των γλωσσών
 - Σύστημα εκτέλεσης
 - Υπηρεσίες καταλόγου
 - Μετατροπή δεδομένων
 - Ασφάλεια / Επικύρωση
- Κατηγοριοποίηση
 - Ανεξαρτησία από γλώσσα
 - Ανεξαρτησία από πλατφόρμα
 - Επιπλέον λειτουργίες
 - Ανεκτικότητα σε σφάλματα
 - Δυνατότητες κατά την εκτέλεση
 - ...

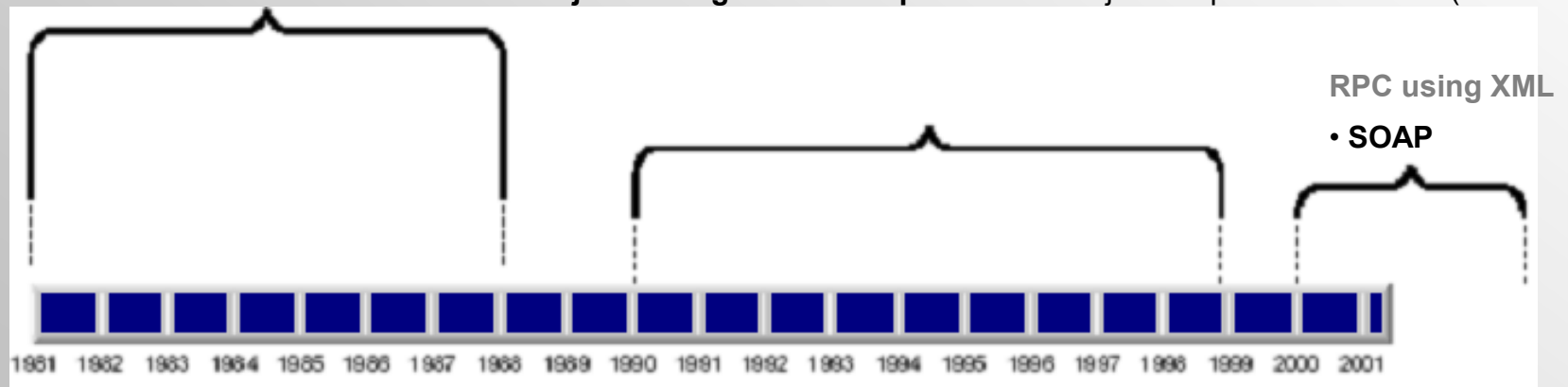
Εξέλιξη των πρωτοκόλλων RPC

API based RPC methods

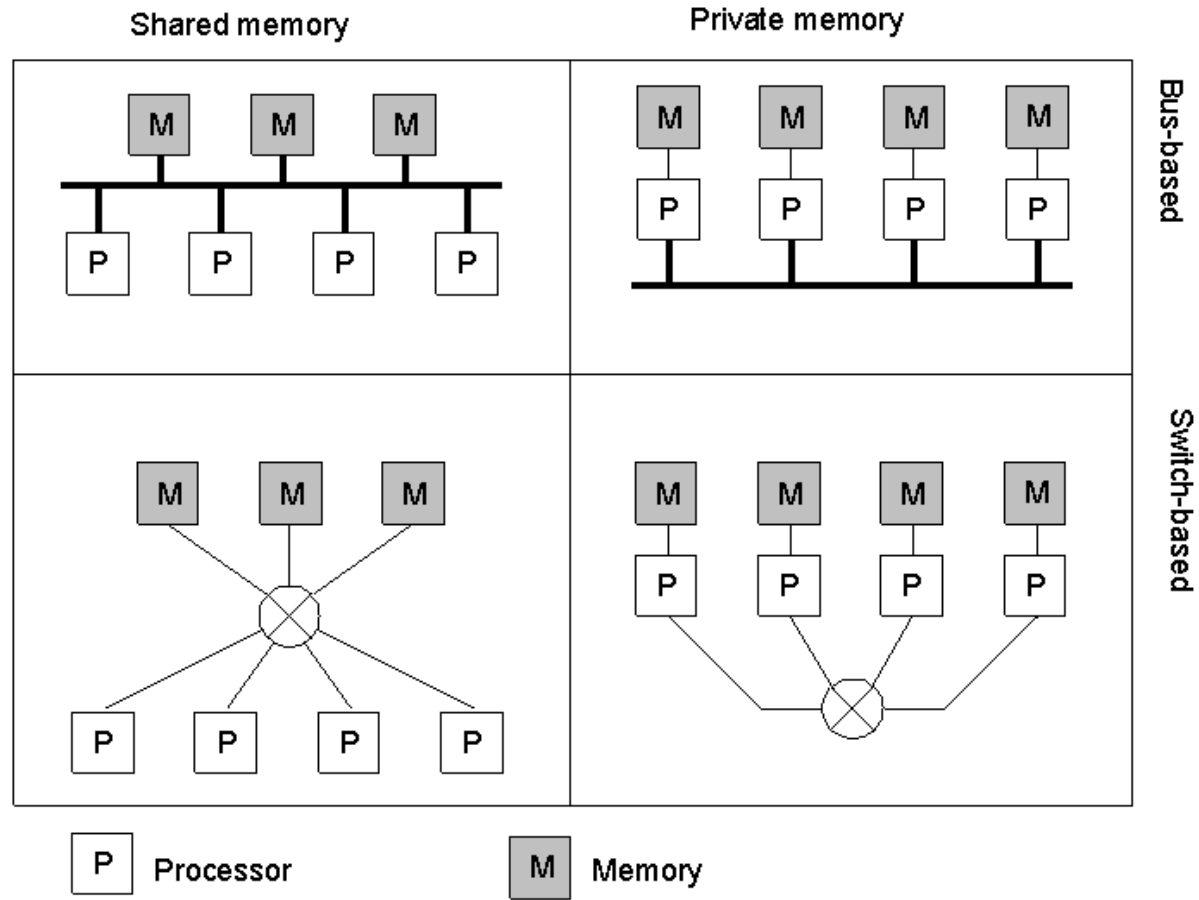
- **IBM**: Application-to-application communications (**APPC**)
- **SUN**: SUN RPC/Network File System (**NFS**) for workstations
- **Open Systems Foundation**: Distributed Computing Environment (**DCE**) RPC

Object-Oriented RPC methods

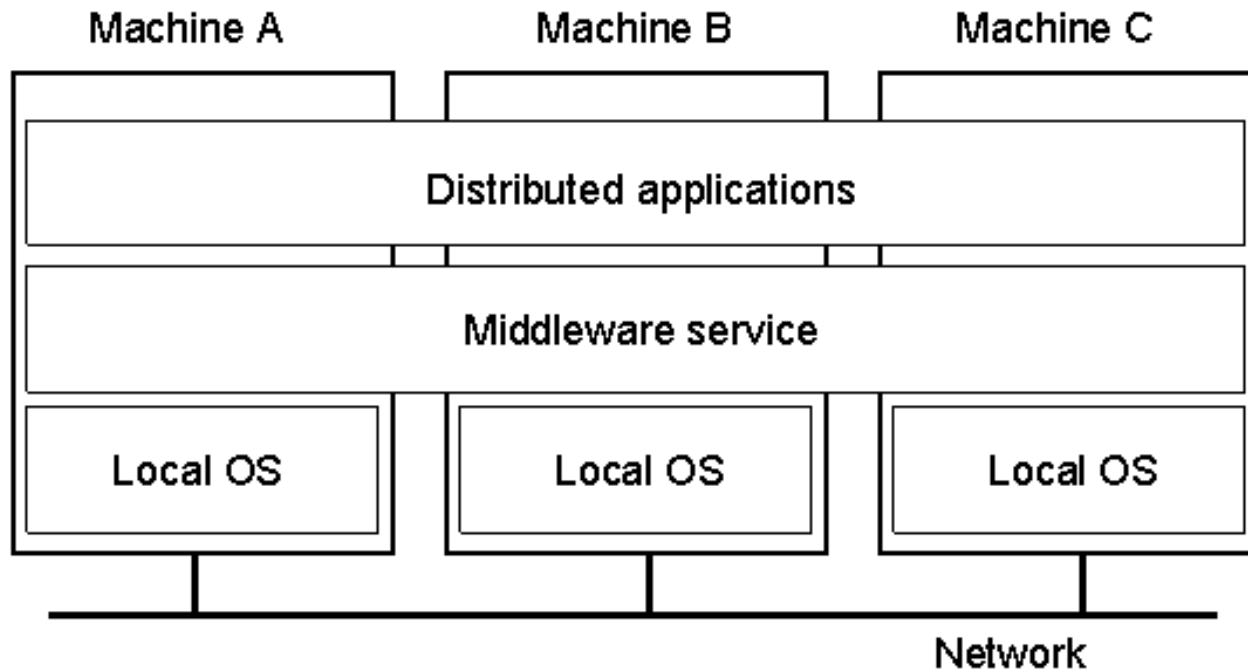
- **Object Management Group**: Common Object Request Architecture (**CORBA**)



Κατανομημένα συστήματα¹



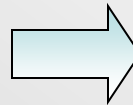
Κατανεμημένα συστήματα²



Επικοινωνία αντικειμένων σε ένα κατανεμημένο περιβάλλον²

Απομακρυσμένη εκτέλεση (Remote Evaluation):

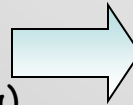
Το αντικείμενο A έχει τη γνώση για την εκτέλεση της εργασίας, αλλά δεν έχει πρόσβαση στους απαιτούμενους πόρους



Το αντικείμενο A στέλνει εντολές σε ένα απομακρυσμένο αντικείμενο B, το οποίο έχει πρόσβαση στους πόρους (αλλά όχι τη γνώση για την εκτέλεση της εργασίας), το B εκτελεί την εργασία και επιστρέφει το αποτέλεσμα στο A

Κώδικας μετά από ζήτηση (Code on Demand):

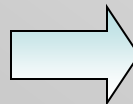
Το αντικείμενο A έχει πρόσβαση στους απαιτούμενους πόρους, αλλά όχι τη γνώση (κώδικα) για την εκτέλεση της εργασίας.



Ο κώδικας μεταφέρεται στη φυσική θέση του A, αποκτά πρόσβαση στους πόρους, η εργασία εκτελείται και το αποτέλεσμα επιστρέφεται στο A

Κινητοί Πράκτορες (Mobile Agents):

Το αντικείμενο A έχει τη γνώση για την εκτέλεση της εργασίας, αλλά δεν έχει πρόσβαση στους απαιτούμενους πόρους



Το αντικείμενο A μεταναστεύει στη φυσική θέση των πόρων ως μία ολόκληρη υπολογιστική οντότητα (κώδικας και κατάσταση), αποκτά πρόσβαση στους πόρους και εκτελείται η εργασία

Αρχιτεκτονικές κατανεμημένων αντικειμένων

Αρχιτεκτονική CORBA (Common Object Request Broker Architecture)

- Επικοινωνία αντικειμένων με βάση τις διεπαφές που ορίζονται από τη γλώσσα IDL (Interface Definition Language)
- Υποστήριξη για ανάπτυξη αντικειμένων σε διάφορες γλώσσες προγραμματισμού
- Βασικό στοιχείο: Το αντικείμενο ORB (Object Request Broker), το οποίο αναλαμβάνει την εκτέλεση μεθόδων απομακρυσμένων αντικειμένων

Remote Method Invocation (RMI)

Αρχιτεκτονική DCOM (Distributed Component Object Model)

- Επικοινωνία κατανεμημένων αντικειμένων σε συστήματα με λειτουργικά συστήματα της Microsoft
- Πλεονέκτημα: Η εκμετάλλευση της ήδη υπάρχουσας υποδομής για την ανάπτυξη κατανεμημένων εφαρμογών

Υπηρεσίες Ιστού (Web Services)

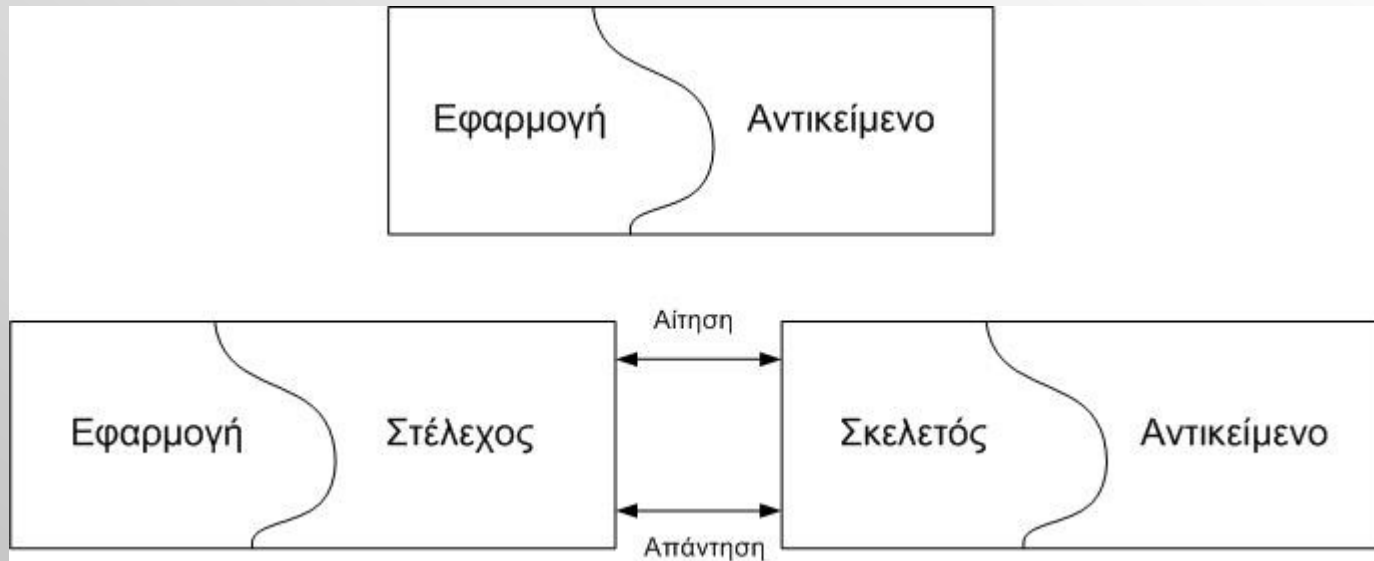
Κινητοί Πράκτορες (mobile agents)

Πλέγμα (Grid)

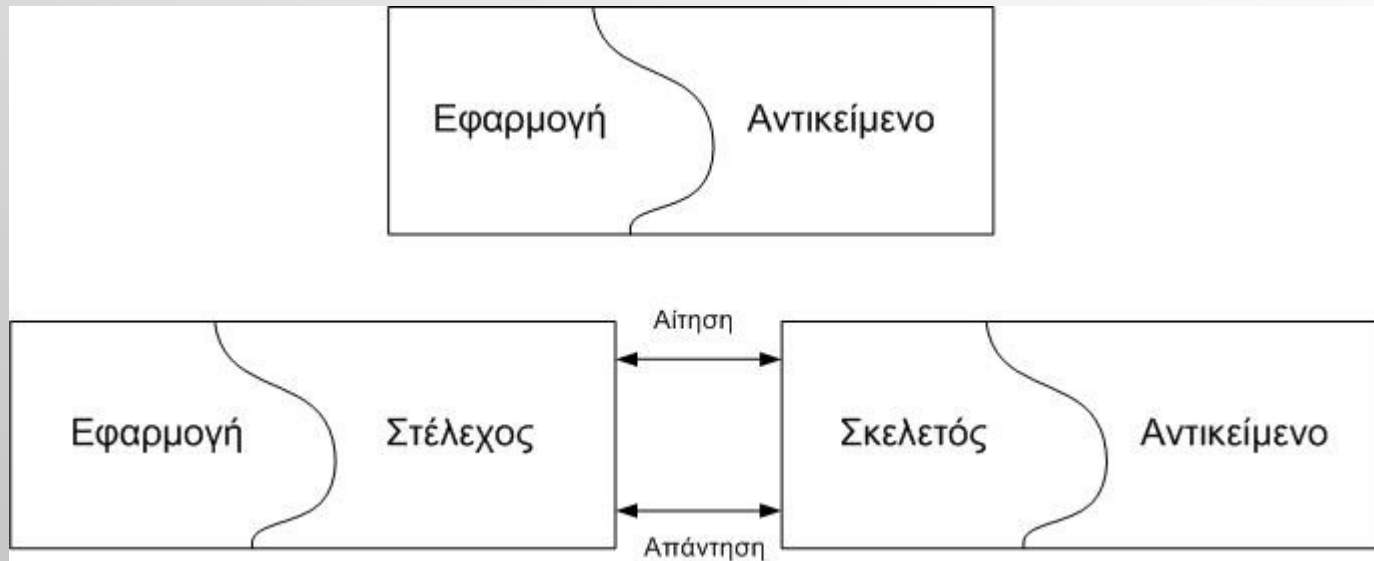
Τεχνολογίες κατανεμημένων αντικειμένων - CORBA¹

- Object Management Group
 - Πρότυπο που χρηματοδοτήθηκε από εταιρίες
 - Τα μέλη του group ανέπτυξαν και ψήφισαν για το πρότυπο
 - Τρέχουσες προδιαγραφές συμπεριλαμβάνουν την UML, CORBA
- Ξεκίνησε τον Απρίλιο του 1989
- Βασικά συστατικά
 - Interface Definition Language (IDL)
 - Object Request Broker (ORB)
 - Internet Inter-ORB Protocol (IIOP)

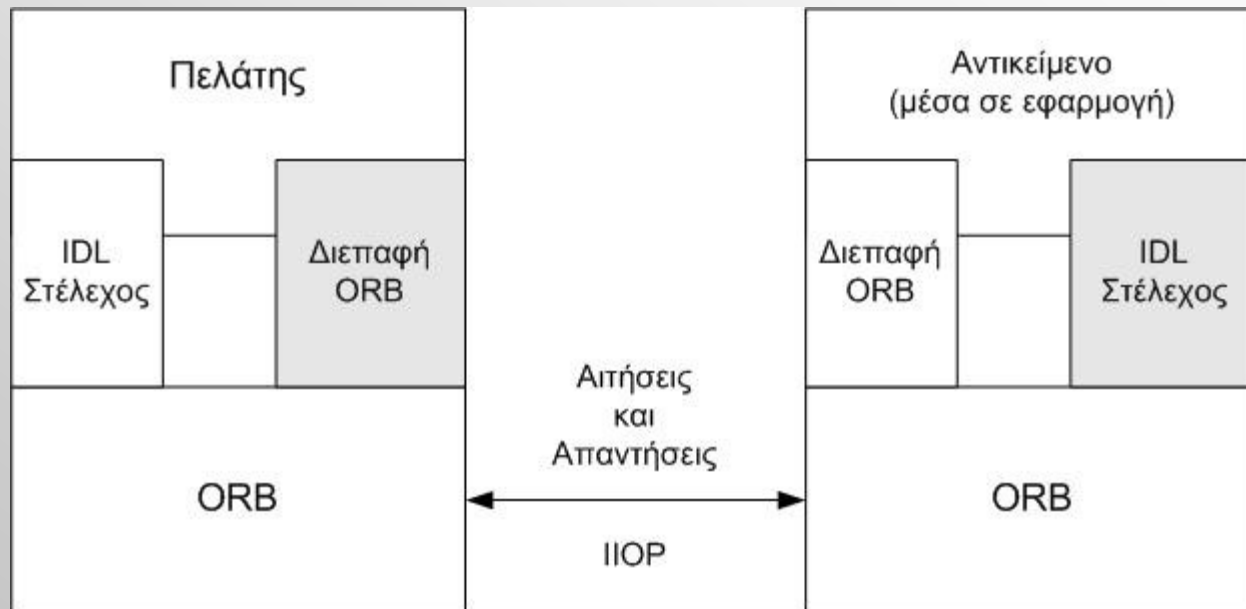
Τεχνολογίες καταναμημένων αντικειμένων - CORBA¹



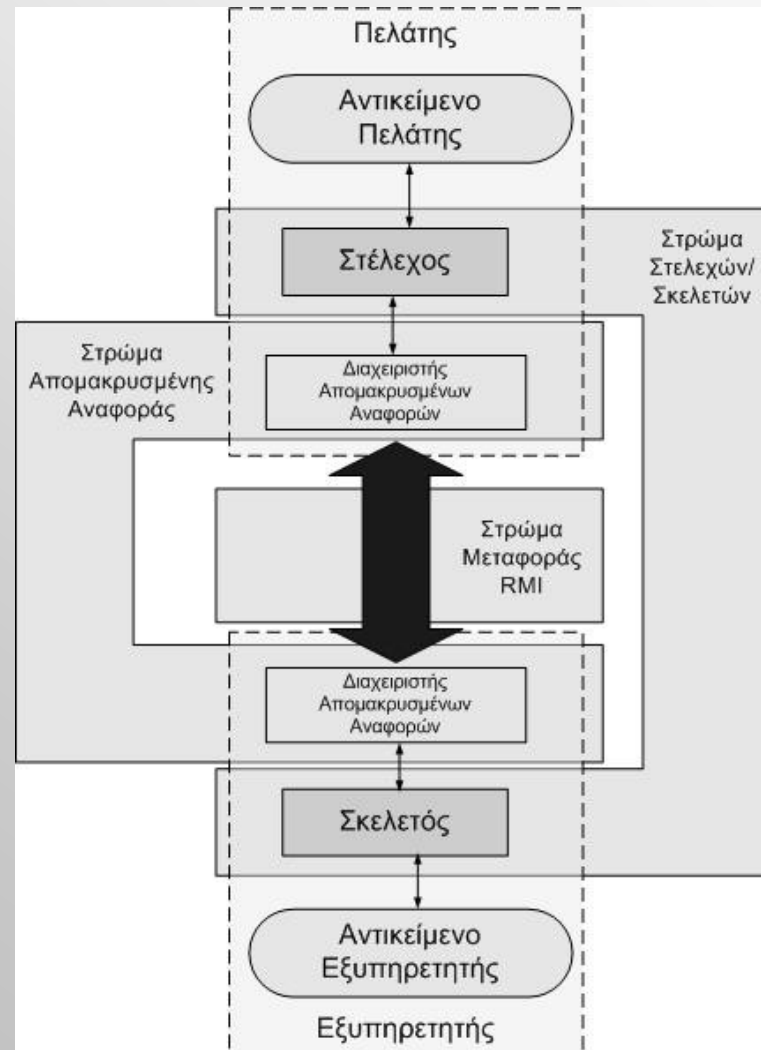
Τεχνολογίες καταναμημένων αντικειμένων - CORBA¹



Τεχνολογίες κατανεμημένων αντικειμένων - CORBA²



Remote Method Invocation - RMI



Web Services - Simple Object Access Protocol¹

- «Ελαφρύ» πρωτόκολλο για την ανταλλαγή πληροφοριών με έναν αποκεντρωμένο τρόπο σε ένα κατανεμημένο περιβάλλον
- Σχεδιασμένο σαν ένας απλός μηχανισμός για την πρόσβαση σε αντικείμενα ή συστατικά που βρίσκονται σε απομακρυσμένους κόμβους δικτύου (Επικοινωνία εφαρμογής με εφαρμογή)
- Το SOAP είναι ένα σύστημα μηνυμάτων για αίτηση υπηρεσιών
- Η εφαρμογή-πελάτης στέλνει μια αίτηση σε μορφή XML και παίρνει απάντηση επίσης σε μορφή XML
- Τα αντικείμενα σεριοποιούνται και κωδικοποιούνται σύμφωνα με το σχήμα κωδικοποίησης του SOAP
- Ανεξάρτητο του ΛΣ, γλώσσας προγραμματισμού και συστατικών
- Μόνη απαίτηση ο πελάτης και ο εξυπηρετητής να μπορούν να επεξεργαστούν κείμενα XML (marshalling / unmarshalling)
- Ανεξάρτητο από το δίκτυο και το πρωτόκολλο επικοινωνίας – το HTTP έχει γίνει το πιο συνηθισμένο εργαλείο

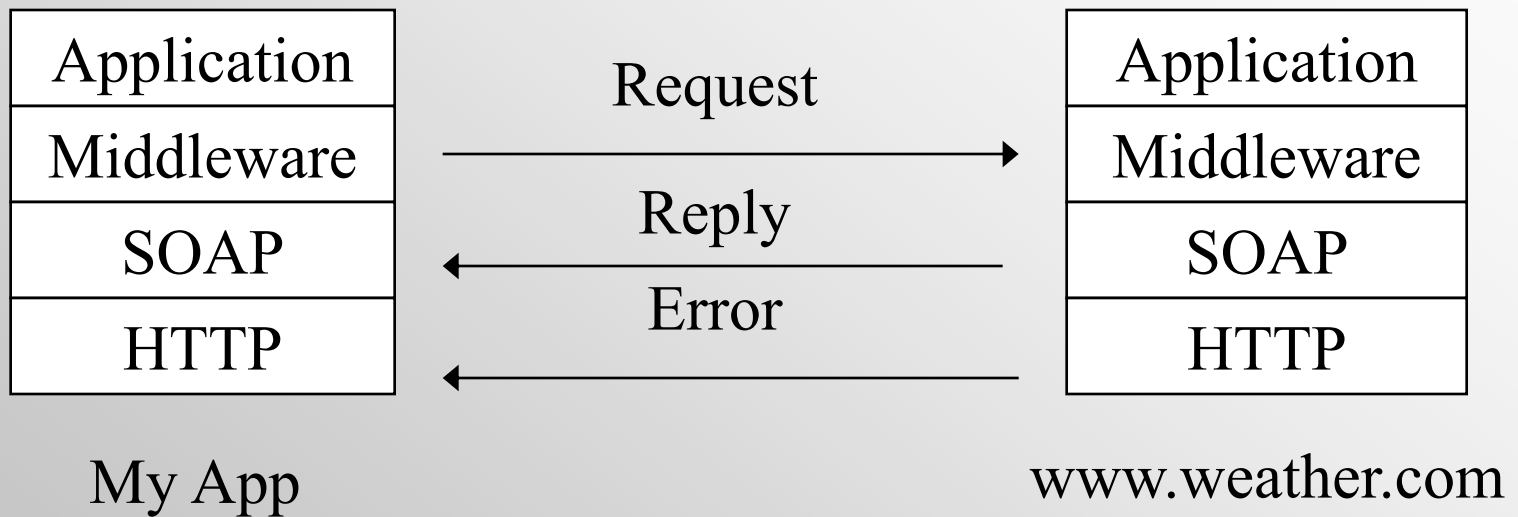
Web Services - Simple Object Access Protocol²

- Firewalls προφυλάσσουν τα Intranet
- Με το SOAP, όλη η μεταφέρσιμη πληροφορία μετασχηματίζεται σε XML και αποστέλλεται πάνω από το HTTP

Στυλ επικοινωνίας του SOAP

- Η Κλήση Απομακρυσμένης Διαδικασίας (Remote Procedure Call) καλεί μια μέθοδο ενός αντικείμενο που βρίσκεται σε ένα απομακρυσμένο κόμβο δικτύου, και επιστρέφει το αποτέλεσμα
- Επικοινωνία είτε μέσω εγγράφων είτε μέσω μηνυμάτων, όπου οι παράμετροι και η απάντηση μπορούν να είναι οποιοδήποτε κείμενο XML

Web Services - Simple Object Access Protocol⁴



Web Services - Simple Object Access Protocol⁴

REQUEST Example

POST /Temperature HTTP/1.1

Host: www.weather.com

Content-Type: text/xml

Content-Length: <whatever>

SOAPMethodName: <some-URI>#CurrentTemp

Http Header

Soap Extensions

<SOAP:Envelope xmlns:SOAP="urn:schemas-xmlsoap-org:soap.v1">

<SOAP:Body>

<m:CurrentTemp xmlns:m="some-URI">

<zip_code>37919</zip_code>

<m:CurrentTemp>

</SOAP:BODY>

<SOAP:Envelope>

Xml Payload

URI- Uniform Resource Identifier

some-URI -> www.netsolve.com or www.globus.com

Web Services - Simple Object Access Protocol⁵

RESPONSE Example

HTTP/1.1 200 OK

Content-Type: text/xml

Content-Length: <whatever>

Http Header

```
<SOAP:Envelope xmlns:SOAP="urn:schemas-xmlsoap-org:soap.v1">
  <SOAP:Header>
    <t:Transaction xmlns:t="some-URI">
      5
    </t:Transaction>
  </SOAP:Header>
  <SOAP:Body>
    <m:CurrentTempResponse xmlns:m="some-URI">
      <return>42</return>
    </m:CurrentTempResponse>
  </SOAP:Body>
</SOAP:Envelope>
```

Xml Payload

Κινητοί πράκτορες (mobile agents)¹

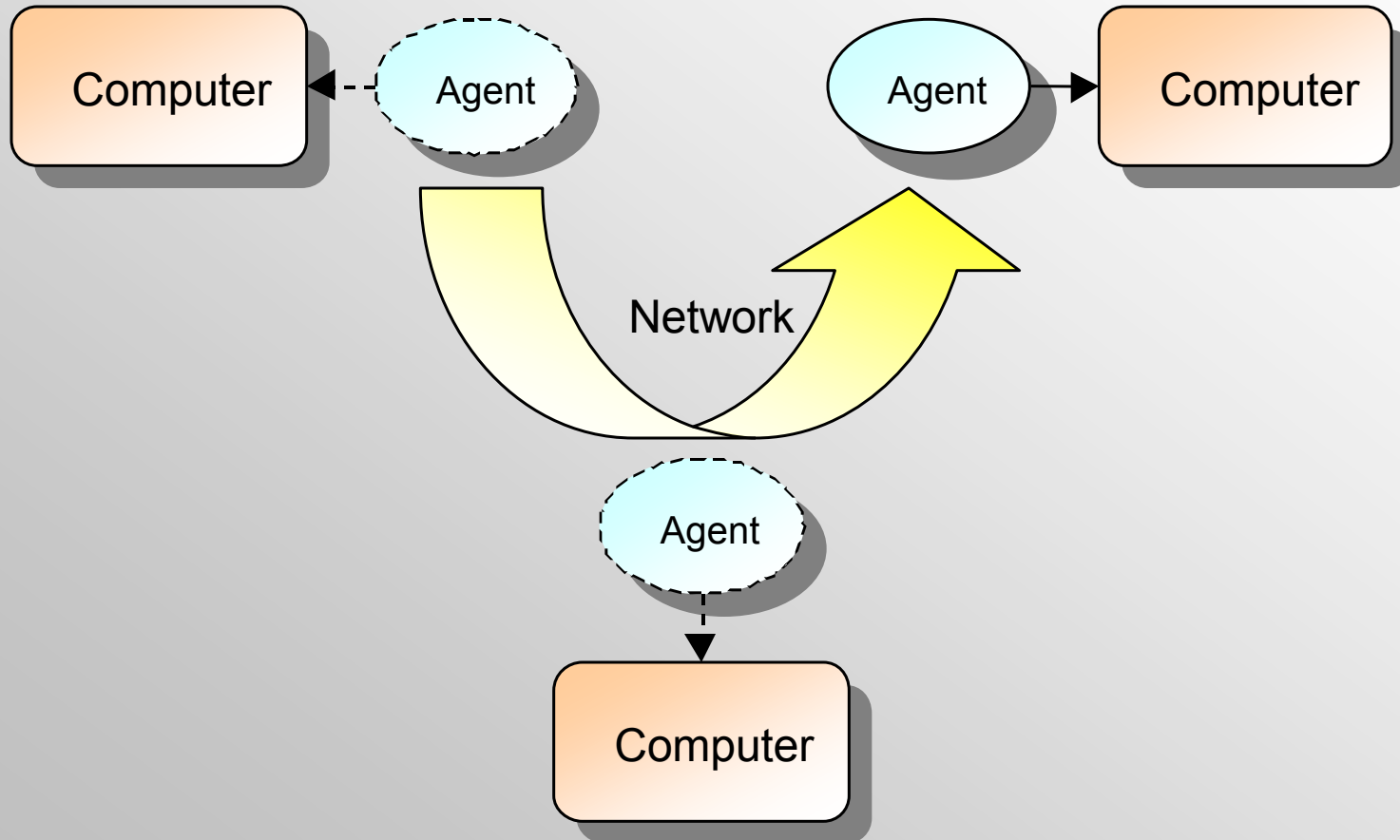
Ένας πράκτορας είναι μία οντότητα λογισμικού η οποία βοηθά το χρήστη, δρώντας εκ μέρους του. Η λειτουργία του είναι να επιτρέπει στους χρήστες να αναθέτουν τις εργασίες, τις οποίες θα έκαναν οι ίδιοι, στο ίδιο το λογισμικό.

Πιο αυστηρά, ένας πράκτορας είναι ένα αντικείμενο με προσωπικό νήμα εκτέλεσης (thread), με άλλα λόγια ένα ενεργό αντικείμενο το οποίο έχει τη μοναδική ικανότητα να μεταφέρεται αυτόνομα από μία δικτυακή τοποθεσία σε μία άλλη.

Χαρακτηριστικά

- ✓ Αυτόνομος (autonomous)
- ✓ Οδηγούμενος απ' το στόχο (goal-driven)
- ✓ Αντιδραστικός (reactive)
- ✓ Κοινωνικός (social)
- ✓ Προσαρμοστικός (adaptive)
- ✓ Αξιόπιστος (reliable)
- ✓ Κινητός (mobile)

Κινητοί πράκτορες (mobile agents)¹



Χαρακτηριστικά κινητών πρακτόρων

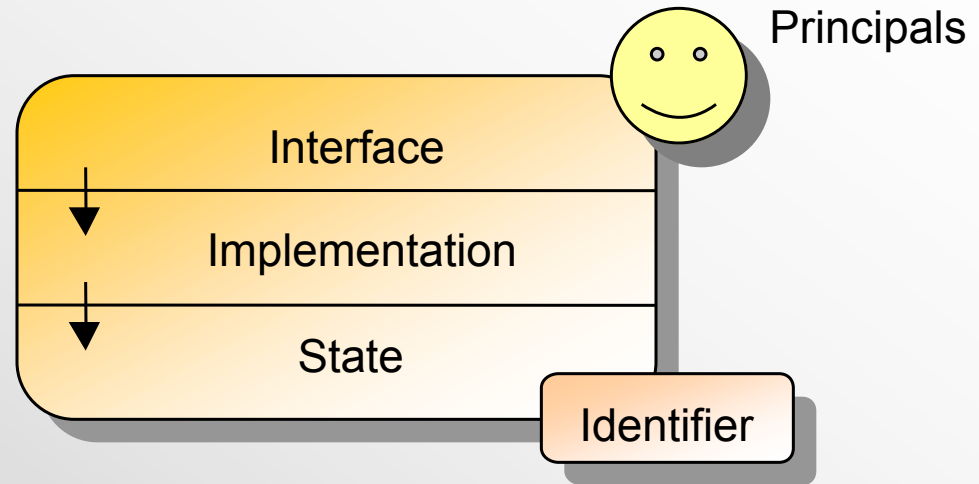
Κατάσταση (state): απαραίτητη για την συνέχιση των υπολογισμών του αντιπρόσωπου έπειτα από τη μετανάστευση

Υλοποίηση (implementation): απαραίτητη για την εκτέλεση του πράκτορα ανεξάρτητα από την τοποθεσία (π.χ if state=1 then)

Διεπαφή (interface): απαραίτητη για την επικοινωνία μεταξύ των πρακτόρων (π.χ `Class myAgent implements IAgentServer`)

Ταυτότητα (Identifier): απαραίτητη για την αναγνώριση, διαχωρισμό και εντοπισμό των μετακινούμενων πρακτόρων

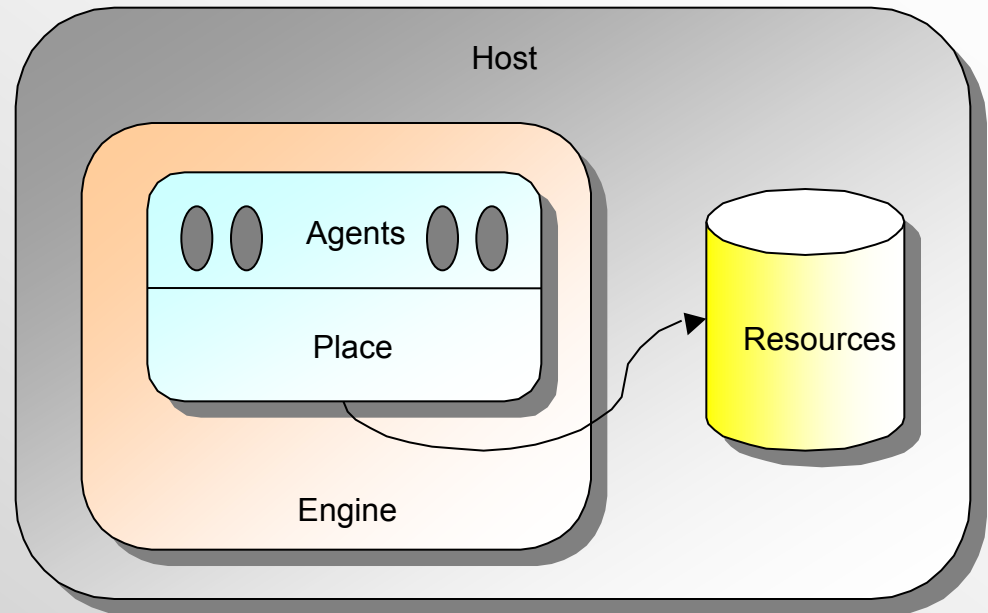
Αρχές (Principals): απαραίτητη για τον προσδιορισμό ηθικών υπευθυνοτήτων από την πλευρά των πρακτόρων



Το περιβάλλον εκτέλεσης των κινητών πρακτόρων

Engine: Η εικονική μηχανή που υποστηρίζει μία ή περισσότερες τοποθεσίες (places)

Resources: Βάσεις δεδομένων, επεξεργαστές, καθώς και άλλες υπηρεσίες που μπορεί να παρέχονται από το σύστημα

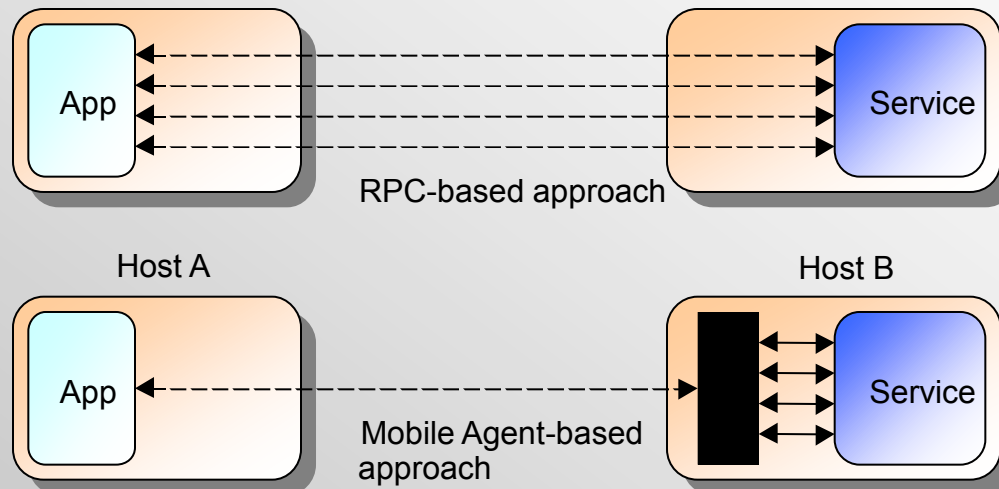


Place: Η δικτυακή διεύθυνση μίας δεδομένης τοποθεσίας

Principals: Είναι υπεύθυνες για τη λειτουργία μίας τοποθεσίας

Τα πλεονεκτήματα από τη χρήση των κινητών πρακτόρων¹

1) Μειώνουν το φορτίο του δικτύου



2) Αντιμετωπίζουν τη δικτυακή καθυστέρηση

- Συστήματα πραγματικού χρόνου όπως είναι π.χ τα κατασκευαστικά ρομπότ χρειάζεται να μπορούν να αντιδράσουν σε εντολές εισόδου σε πραγματικό χρόνο.
- Οι κινητοί πράκτορες προσφέρουν λύση, αφού αποστέλλονται από ένα κεντρικό ελεγκτή και δρουν τοπικά εκτελώντας τις διαταγές του ελεγκτή.

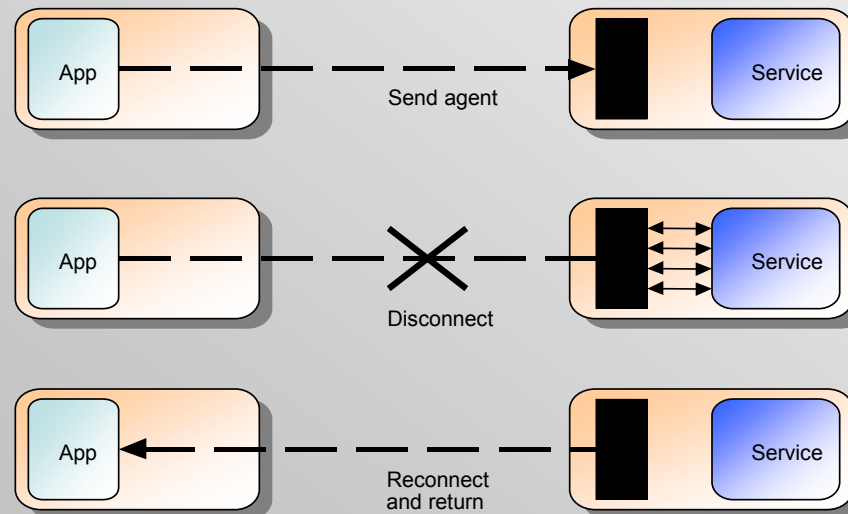
Τα πλεονεκτήματα από τη χρήση των κινητών πρακτόρων²

3) Ενσωματώνουν πρωτόκολλα

Καθώς τα πρωτόκολλα εξελίσσονται για να ενσωματώσουν καινούργιες απαιτήσεις όπως π.χ ασφάλεια, καθίσταται πολύ δύσκολη η αναβάθμιση του πρωτοκόλλου.

Οι κινητοί πράκτορες είναι ικανοί να μεταναστεύσουν σε απομακρυσμένους υπολογιστές προκειμένου να τους καταστήσουν ικανούς να επεξεργαστούν τα δεδομένα σύμφωνα με το εκάστοτε πρωτόκολλο.

4) Εκτελούνται ασύγχρονα και αυτόνομα



Τα πλεονεκτήματα από τη χρήση των κινητών πρακτόρων²

5) Προσαρμόζονται δυναμικά

Οι κινητοί πράκτορες έχουν την ικανότητα να αντιλαμβάνονται το περιβάλλον εκτέλεσής τους και να αντιδρούν αυτόνομα στις αλλαγές.

Πολλαπλοί πράκτορες έχουν τη μοναδική ικανότητα να κατανέμουν τους εαυτούς τους μεταξύ των διαθέσιμων υπολογιστών του δικτύου προκειμένου να πετύχουν το βέλτιστο τρόπο για την επίλυση ενός προβλήματος.

6) Είναι ετερογενείς από τη φύση τους

Η δικτυακή επεξεργασία είναι από τη φύση της ετερογενής, συχνά τόσο από την πλευρά του υλικού όσο και λογισμικού.

Καθώς οι κινητοί πράκτορες είναι ανεξάρτητοι από το επίπεδο συστήματος και το επίπεδο μεταφοράς (transport layer) και εξαρτώνται μόνο από το περιβάλλον εκτέλεσής τους, παρέχουν τις κατάλληλες συνθήκες για την ολοκλήρωση των ετερογενών μερών του συστήματος.

7) Παρουσιάζουν ευρωστία και αντοχή σε σφάλματα

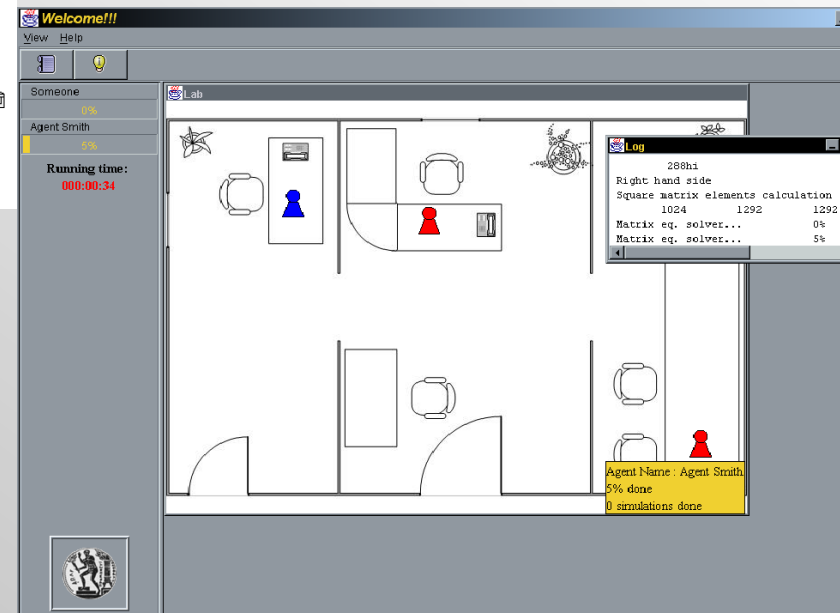
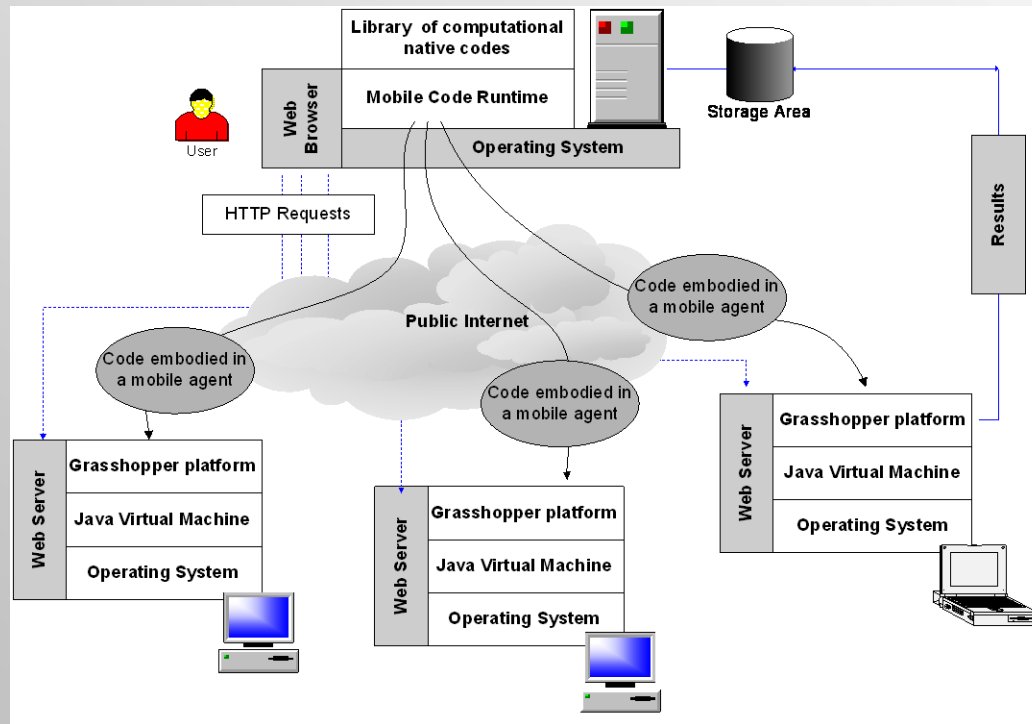
Η ικανότητα των κινητών πρακτόρων να αντιδρούν σε ανεπιθύμητες καταστάσεις και γεγονότα είναι πιο εύκολη η δημιουργία εύρωστων συστημάτων.

Εάν ένας υπολογιστής σταματήσει να λειτουργεί, οι πράκτορες που τρέχουν σε αυτόν είναι δυνατό να ενημερωθούν έγκαιρα, να μεταναστεύσουν και να συνεχίσουν την εργασία τους από το σημείο που είχε σταματήσει.

Εφαρμογές που επωφελούνται από τη χρήση των κινητών πρακτόρων

- Εφαρμογές ελέγχου ροής (Workflow applications)
- Ηλεκτρονικό εμπόριο (E-Commerce)
- Κατανεμημένη ανάκτηση πληροφοριών
- Διάδοση πληροφοριών
- Απομακρυσμένη εκτέλεση και παρακολούθηση διεργασιών
- Παράλληλη επεξεργασία

Παραμετρική εκτέλεση υπολογιστικών κωδίκων



Κατανεμημένη παράλληλη επεξεργασία

- Αποσύνθεση υπολογιστικού χώρου (Domain decomposition)
 - 1-D και 2-D επίλυση των εξισώσεων Laplace και Poisson
 - Με τη μέθοδο Jacobi
 - Με τη μέθοδο SOR
 - 2-D επίλυση προβλήματος σκέδασης με τη μέθοδο FDTD

Η απήχηση της Java¹

- **Ανεξαρτησία από πλατφόρμα**
 - Η Java είναι σχεδιασμένη να λειτουργεί σε ετερογενή περιβάλλοντα.
 - Επιτρέπει τη δημιουργία ενός αντιπρόσωπου που μπορεί να εκτελεστεί σε οποιαδήποτε πλατφόρμα που έχει εγκατεστημένη την εικονική μηχανή της Java.
- **Ασφαλής εκτέλεση**
 - Η Java έχει ένα μοντέλο δεικτών (pointers) που εξαφανίζει την πιθανότητα της εγγραφής θέσεων της μνήμης ή την καταστροφή δεδομένων
 - Η αρχιτεκτονική ασφάλειας της Java επιτρέπει την ασφαλή εκτέλεση ενός μη-έμπιστου αντιπρόσωπου διότι ο πράκτορας δε μπορεί να έχει πρόσβαση σε απόρρητες πληροφορίες.
- **Δυναμικό φόρτωμα κλάσεων**
 - Η εικονική μηχανή μπορεί να φορτώνει και να ορίζει κλάσεις κατά την διάρκεια εκτέλεσης ενός προγράμματος.
- **Πολυνηματικός προγραμματισμός (Multithreaded programming)**
 - Οι αντιπρόσωποι είναι εξ' ορισμού αυτόνομοι
 - Η Java όχι μόνο υποστηρίζει πολυνηματικό προγραμματισμό, αλλά υποστηρίζει επίσης ένα σύνολο από δυνατότητες συγχρονισμού (synchronization primitives) δομημένες στην ίδια τη γλώσσα.

Η απήχηση της Java²

- **Σειριοποίηση αντικειμένων**

- Ένα βασικό χαρακτηριστικό των κινητών πρακτόρων είναι η δυνατότητά τους για σειριοποίηση και αποσειριοποίηση.
- Η Java παρέχει ένα ενδογενή μηχανισμό σειριοποίησης ο οποίος αναπαριστά την κατάσταση ενός αντικειμένου σε σειριοποιημένη μορφή με αρκετά λεπτομερή τρόπο ώστε το αντικείμενο να μπορεί να ανακατασκευαστεί αργότερα.

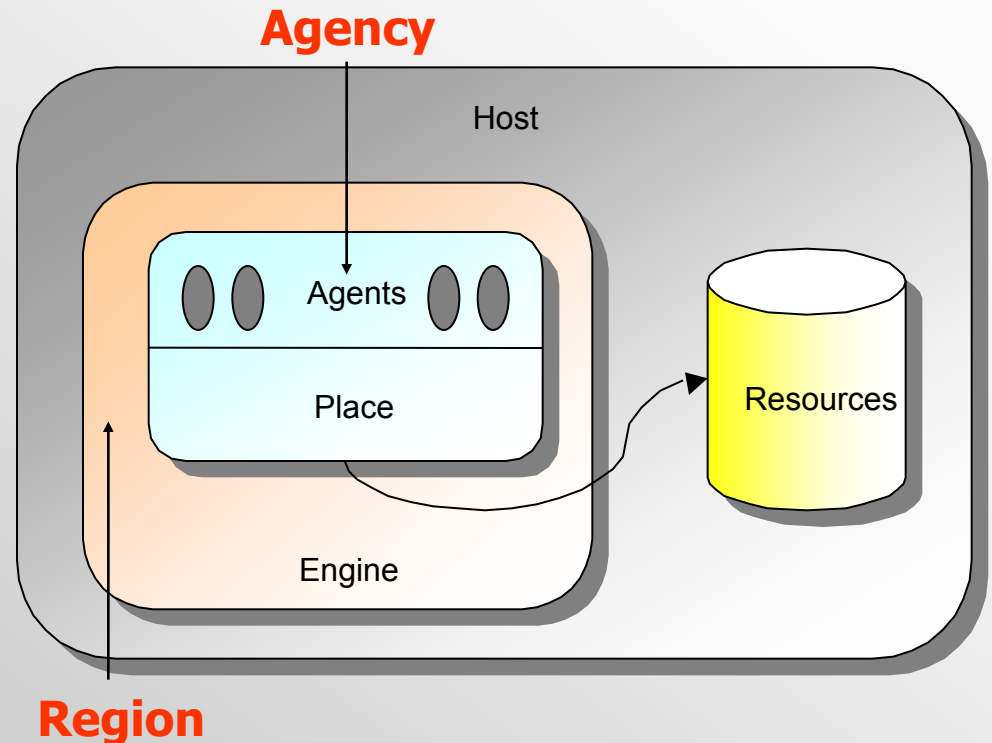
Μειονεκτήματα της Java

- **Ανεπαρκής υποστήριξη ελέγχου πόρων**
 - Δυστυχώς, η Java δεν παρέχει τρόπους περιορισμού των πόρων του επεξεργαστή και της μνήμης που διατίθεται σε κάποιο αντικείμενο ή νήμα.
 - Παρόμοιο θέμα είναι η ικανότητα ενός πράκτορα να δεσμεύει πόρους εξωτερικούς του προγράμματος, για παράδειγμα με το να ανοίγει αρχεία και διόδους επικοινωνίας (sockets), και παράγοντας παράθυρα.
- **Δεν υπάρχει προστασία των αναφορών ενός αντικειμένου**
 - Οι public μέθοδοι ενός Java αντικειμένου είναι διαθέσιμες σε κάθε άλλο αντικείμενο που έχει αναφορά σε αυτό.
 - Δεν υπάρχει τρόπος σε κάποιον πράκτορα να επιβλέπει και να ελέγχει τα αντικείμενα (πράκτορες) που έχουν πρόσβαση στις μεθόδους του.
- **Δεν υπάρχει υποστήριξη για διατήρηση και επανάκτηση της κατάστασης εκτέλεσης**
 - Είναι προς το παρόν αδύνατη η επανάκτηση της πλήρους κατάστασης εκτέλεσης ενός αντικειμένου με τη Java.
 - Πληροφορίες όπως η κατάσταση του μετρητή προγράμματος (program counter) και της στοίβας ετολών είναι απαγορευτικές για προγράμματα της Java (γλώσσα υψηλού επιπέδου).

Κινητοί πράκτορες - το πρότυπο MASIF¹

Περιοχή (Region): Διευκολύνει τη διαχείριση των πρακτορείων (agencies), τοποθεσιών (places) και πρακτόρων (agents). Μία περιοχή περιέχει μία μόνον περιοχή καταχώρησης (region registry) και πολλά πρακτορεία.

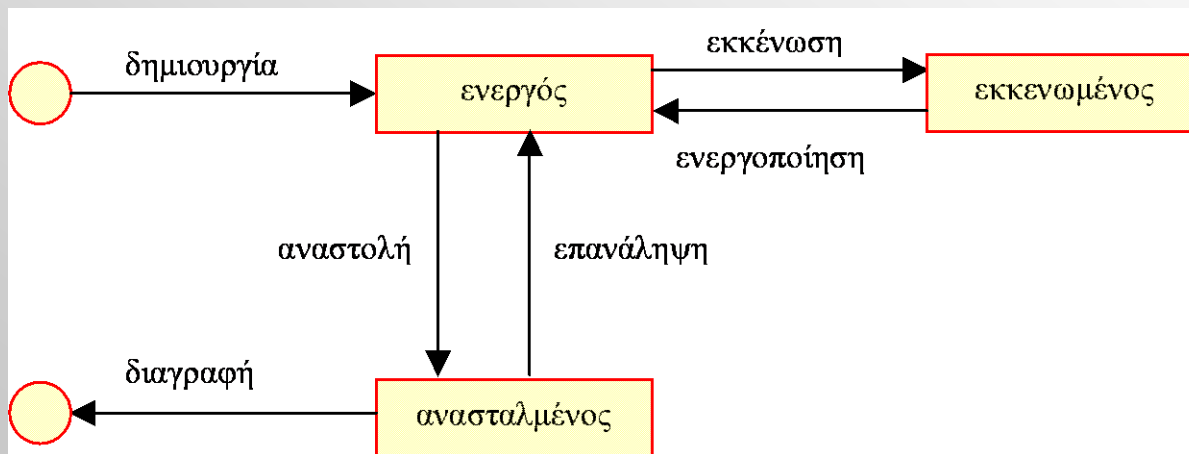
Περιοχή Καταχώρησης (Region Registry): Παρέχει τη βάση δεδομένων με πληροφορία σχετική για τις πρακτορεία, τοποθεσίες και τους πράκτορες μιας περιοχής.



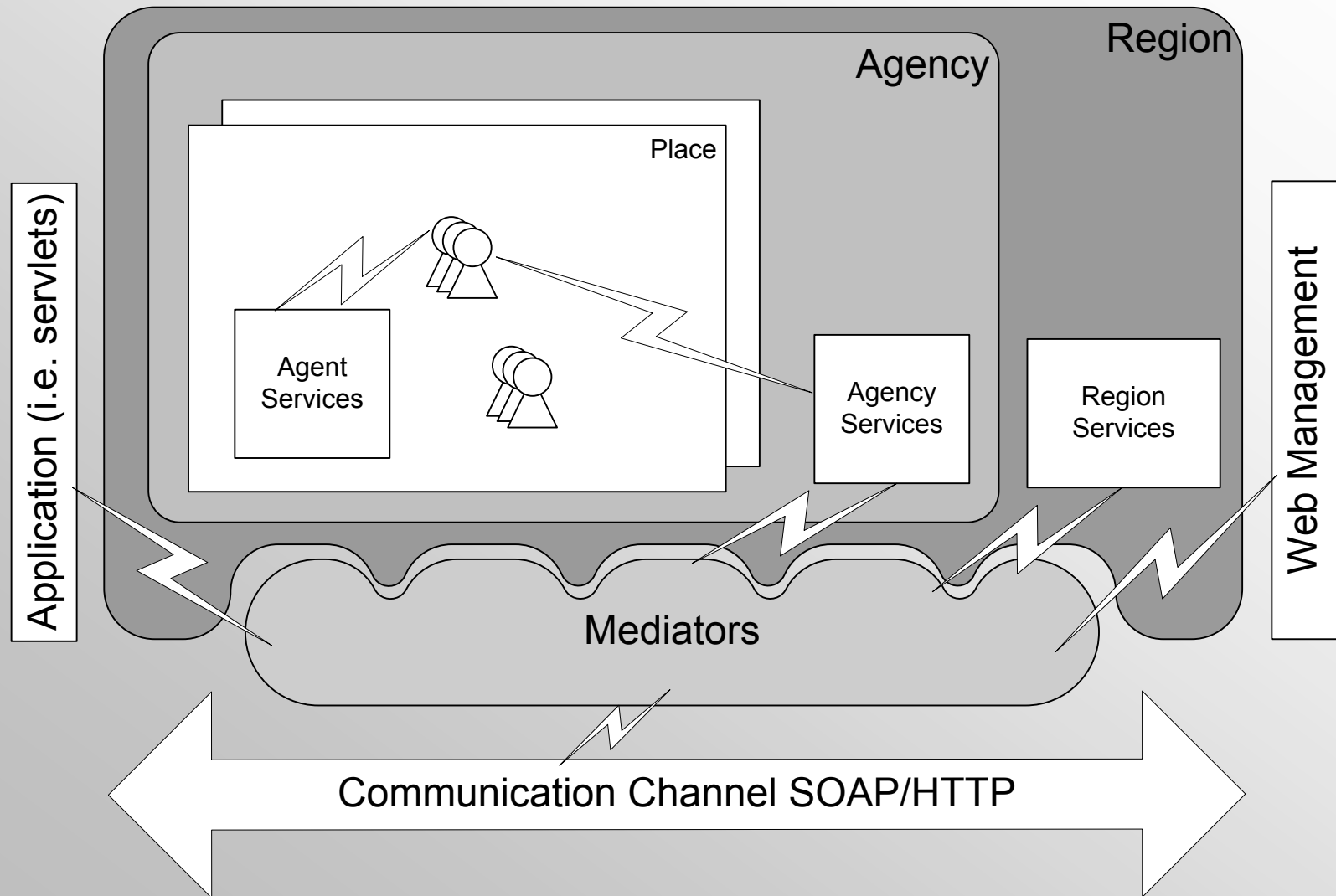
Κινητοί πράκτορες - το πρότυπο MASIF¹

- **Πρακτορείο (Agency):** Παρέχει το περιβάλλον εκτέλεσης των κινητών πρακτόρων. Κάθε πρακτορείο εκτελείται σε ξεχωριστή εικονική μηχανή της Java και αποτελείται μία ή περισσότερες τοποθεσίες. Το πρακτορείο παρέχει τις βασικές λειτουργίες, όπως διαχείριση του κύκλου ζωής του πράκτορα, μεταφορά, επικοινωνία και ασφάλεια.
- **Τοποθεσία (Place):** Επιτρέπει την οργάνωση των πρακτόρων, που παρέχουν ίδιες υπηρεσίες. Οι τοποθεσίες ανήκουν στα πρακτορεία. Κάθε τοποθεσία μπορεί να συσχετιστεί με μια πολιτική ασφάλειας (security policy), για τη διευκόλυνση της διαχείρισης της ασφάλειας.
- **Πράκτορας (agent):** Οντότητα λογισμικού, που έχει τη δυνατότητα να δρα αυτόνομα, αποτελούμενη από μία ή περισσότερες Java κλάσεις. Υπάρχουν δύο είδη πρακτόρων: οι κινητοί (mobile) και οι στάσιμοι (stationary).

Ο κύκλος ζωής του κινητού πράκτορα



Πλατφόρμα WebMAges



Πλέγμα (Grid)¹

- Κεντρική ιδέα η δημιουργία ενός δικτύου το οποίο θα επιτρέπει το διαμερισμό των πόρων στους χρήστες του δικτύου
- Τρεις τύποι
 - Computational Grids
 - Data Grids
 - Service Grids
- Τρεις γενιές

Πλέγμα (Grid)²

Τρεις γενιές συστημάτων

- Πρώτη γενιά

- Προγράμματα για να συνδέσουν περιοχές με δυνατότητες supercomputing
- FAFNER, I-WAY

- Δεύτερη γενιά

- Τρία κύρια ζητήματα
 - Ετερογένεια
 - Εξελιξιμότητα
 - Προσαρμοστικότητα
- Μεσισμικό
- Παραλλαγές – π.χ. φύλαξη δημοφιλούς πληροφορίας σε πολλούς κόμβους

Πλέγμα (Grid)³

- Τρίτη γενιά
 - Στόχος: e-Science
 - Αρχιτεκτονικές βασισμένες στις υπηρεσίες
 - Διαδοχική μετάβαση
 - Web services
 - Open Grid Services Architecture (OGSA)
 - WS-Resource Framework (WS-RF)

OGSA/OGSI

