



## Ενότητα 5 (κεφάλαιο 18) – Τεχνολογία Λογισμικού για Κατανεμημένα Συστήματα

Οι διαφάνειες αυτές έχουν συμπληρωματικό και επεξηγηματικό χαρακτήρα και σε καμία περίπτωση δεν υποκαθιστούν το βιβλίο

---

---

---

---

---

---

---

---

### Περιεχόμενα



- ✦ Θέματα κατανεμημένων συστημάτων.
- ✦ Υπολογισμός τύπου πελάτη-διακομιστή (client-server).
- ✦ Αρχιτεκτονικά υποδείγματα για κατανεμημένα συστήματα.
- ✦ Λογισμικό ως υπηρεσία (Software as a Service – SaaS).

---

---

---

---

---

---

---

---

### Κατανεμημένα συστήματα



- ✦ Σήμερα, σχεδόν όλα τα μεγάλα συστήματα που βασίζονται σε υπολογιστές είναι πλέον κατανεμημένα.  
«... μία συλλογή από ανεξάρτητους υπολογιστές που εμφανίζονται στο χρήστη ως ένα ενιαίο συνεκτικό σύστημα.»
- ✦ Η επεξεργασία των πληροφοριών κατανέμεται σε πολλούς υπολογιστές και δεν περιορίζεται σε μία μόνο μηχανή.
- ✦ Επομένως η τεχνολογία λογισμικού για κατανεμημένα συστήματα έχει μεγάλη σημασία για τα υπολογιστικά συστήματα των εταιριών.

---

---

---

---

---

---

---

---

## Χαρακτηριστικά κατανεμημένων συστημάτων



- ❖ Μερισμός πόρων.
  - Κοινή χρήση πόρων υλικού και λογισμικού.
- ❖ Ανοικτή λειτουργία.
  - Χρήση εξοπλισμού και λογισμικού από διάφορους κατασκευαστές.
- ❖ Ταυτοχρονισμός.
  - Ταυτόχρονη επεξεργασία για τη βελτίωση της απόδοσης.
- ❖ Επεκτασιμότητα.
  - Αυξημένη διεκπεραιωτική ικανότητα με την προσθήκη νέων πόρων.
- ❖ Ανοχή σφαλμάτων.
  - Η ικανότητα συνέχειας της λειτουργίας αφού ανακūψει κάποιο σφάλμα.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Κατανεμημένα Συστήματα

4

---

---

---

---

---

---

---

---

---

---

## Ζητήματα σχετικά με κατανεμημένα συστήματα



- ❖ Τα κατανεμημένα συστήματα είναι πιο πολύπλοκα από συστήματα που εκτελούνται σε ένα επεξεργαστή.
- ❖ Η πολυπλοκότητα οφείλεται στο ότι ένα τέτοιο σύστημα αποτελείται από διαφορετικά μέρη των οποίων η διαχείριση γίνεται ανεξάρτητα του ενός από τα άλλα, όπως συμβαίνει και με το δίκτυο.
- ❖ Δεν υπάρχει μία μοναδική αρχή που να είναι υπεύθυνη του συστήματος και επομένως ο έλεγχος από πάνω προς τα κάτω δεν είναι δυνατός.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Κατανεμημένα Συστήματα

5

---

---

---

---

---

---

---

---

---

---

## Ζητήματα σχεδιασμού



- ❖ **Διαφάνεια.**
  - Μέχρι ποιού βαθμού θα πρέπει το κατανεμημένο σύστημα να παρουσιάζεται στον χρήστη ως ένα ενιαίο σύστημα;
- ❖ **Ανοικτή λειτουργία.**
  - Θα πρέπει ένα σύστημα να σχεδιάζεται με χρήση καθιερωμένων πρωτοκόλλων που υποστηρίζουν διαλειτουργικότητα;
- ❖ **Επεκτασιμότητα.**
  - Πως μπορεί να κατασκευαστεί το σύστημα έτσι ώστε να είναι εύκολα επεκτάσιμο;
- ❖ **Ασφάλεια.**
  - Πως μπορούν να ορισθούν και υλοποιηθούν εύκολες στη χρήση πολιτικές ασφάλειας;
- ❖ **Ποιότητα υπηρεσιών.**
  - Πως πρέπει να καθορισθούν οι κανόνες ποιότητας των παρεχόμενων υπηρεσιών;
- ❖ **Διαχείριση σφαλμάτων.**
  - Πως μπορούν τα σφάλματα του συστήματος να ανιχνευθούν, περιορισθούν και διορθωθούν;

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Κατανεμημένα Συστήματα

6

---

---

---

---

---

---

---

---

---

---

## Διαφάνεια



- ❖ Ίδανικά, οι χρήστες δεν θα πρέπει να γνωρίζουν ότι ένα σύστημα είναι κατανεμημένο και οι παρεχόμενες υπηρεσίες πρέπει να είναι ανεξάρτητες από χαρακτηριστικά κατανομής.
- ❖ Στην πράξη όμως αυτό είναι αδύνατον γιατί κάποια μέρη του συστήματος διαχειρίζονται ανεξάρτητα από άλλα και επίσης λόγω καθυστερήσεων του δικτύου.
  - Συχνά είναι καλύτερα να γνωρίζουν οι χρήστες την κατανομή του συστήματος για να μπορούν να αντιμετωπίσουν τυχόν προβλήματα.
- ❖ Για να επιτευχθεί διαφάνεια, αναφορά στους πόρους του συστήματος πρέπει να γίνεται μέσω αφαίρεσης με λογικό παρά με φυσικό τρόπο. Το ενδιάμεσο λογισμικό (middleware) αντιστοιχεί λογικούς σε φυσικούς πόρους.

---

---

---

---

---

---

---

---

---

---

## Ανοικτή λειτουργία



- ❖ Τα ανοικτά κατανεμημένα συστήματα είναι συστήματα που δημιουργούνται με βάση γενικά αποδεκτών προτύπων.
- ❖ Συστατικά στοιχεία από οποιονδήποτε προμηθευτή μπορούν να ενσωματωθούν στο σύστημα και να συνεργασθούν με τα υπόλοιπα συστατικά στοιχεία του συστήματος.
- ❖ Η ανοικτή λειτουργία υπονοεί ότι τα συστατικά στοιχεία ενός συστήματος μπορούν να δημιουργηθούν με οποιαδήποτε γλώσσα προγραμματισμού και αν συμμορφώνονται με τα καθιερωμένα πρότυπα θα μπορούν να λειτουργήσουν με άλλα συστατικά στοιχεία.
- ❖ Έχουν δημιουργηθεί ανοικτής λειτουργίας πρότυπα υπηρεσιών διαδικτύου για υπηρεσιοκεντρικές αρχιτεκτονικές.

---

---

---

---

---

---

---

---

---

---

## Επέκτασιμότητα



- ❖ Η επέκτασιμότητα ενός συστήματος αντανακλάει τη δυνατότητά του να συνεχίζει να παρέχει υψηλής ποιότητας υπηρεσίες καθώς αυξάνεται η ανάγκη χρήσης του συστήματος.
  - *Μέγεθος.* Θα πρέπει να είναι δυνατή η προσθήκη περισσότερων πόρων σε ένα σύστημα για την αντιμετώπιση αυξανόμενων αριθμών χρηστών.
  - *Κατανομή.* Θα πρέπει να είναι δυνατή η γεωγραφική κατανομή των συστατικών στοιχείων ενός συστήματος χωρίς αυτό να οδηγεί σε μείωση της απόδοσής του.
  - *Διαχειρισσιμότητα.* Θα πρέπει να είναι δυνατή η διαχείριση ενός συστήματος το οποίο αυξάνεται σε μέγεθος, ακόμα και αν κάποια μέρη του συστήματος βρίσκονται σε ανεξάρτητους οργανισμούς.
- ❖ Υπάρχει διαφορά μεταξύ αναβάθμισης (scaling up) και επέκτασης (scaling out). Με αναβάθμιση το σύστημα καθίσταται πιο δυνατό ενώ με επέκταση προστίθενται επιπλέον στιγμιότυπα του συστήματος.

---

---

---

---

---

---

---

---

---

---

## Ασφάλεια



- ✧ Όταν ένα σύστημα είναι καταμετρημένο, οι τρόποι που μπορεί να δεχθεί επιθέσεις αυξάνονται σε σχέση με τα συγκεντρωτικά συστήματα.
- ✧ Αν η επίθεση σε κάποιο μέρος του συστήματος είναι επιτυχής, τότε ο επιτιθέμενος μπορεί να το χρησιμοποιήσει αυτό ως μία «πίσω πόρτα» για πρόσβαση σε άλλα μέρη του συστήματος.
- ✧ Εδώ δημιουργούνται σχετικές δυσκολίες γιατί διαφορετικά μέρη του συστήματος μπορεί να ανήκουν σε διαφορετικούς οργανισμούς. Αυτοί οι οργανισμοί μπορούν να έχουν αμοιβαία μη συμβατές πολιτικές και μηχανισμούς ασφάλειας.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Καταμετρημένα Συστήματα

10

---

---

---

---

---

---

---

---

---

---

## Είδη επιθέσεων



- ✧ Τα είδη των απειλών από τα οποία ένα καταμετρημένο σύστημα πρέπει να προφυλαχθεί είναι τα ακόλουθα:
  - **Υποκλοπή**, όπου η επικοινωνία μεταξύ κάποιων μερών του συστήματος υποκλέπεται από έναν επιτιθέμενο και δημιουργείται απώλεια εχεμυθείας.
  - **Διακοπή**, όπου κάποιες από τις υπηρεσίες του συστήματος δέχονται επίθεση και δεν μπορούν να λειτουργήσουν όπως αναμένεται από αυτές.
    - Οι επιθέσεις που έχουν σχέση με αρνήσεις παροχής υπηρεσιών γίνονται με τον βομβαρδισμό ενός κόμβου με αθέμιτες αιτήσεις έτσι ώστε ο κόμβος να μην μπορεί να ανταποκριθεί στις θεμιτές αιτήσεις.
  - **Τροποποίηση**, όπου δεδομένα ή υπηρεσίες ενός συστήματος τροποποιούνται από τον επιτιθέμενο.
  - **Κατασκευάσματα**, όπου ένας επιτιθέμενος παράγει πληροφορίες που δεν θα έπρεπε να υπάρχουν και μετά τις χρησιμοποιεί για να αποκτήσει δικαιώματα στο σύστημα.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Καταμετρημένα Συστήματα

11

---

---

---

---

---

---

---

---

---

---

## Ποιότητα υπηρεσιών



- ✧ Η ποιότητα υπηρεσιών (QoS) που προσφέρεται από ένα καταμετρημένο σύστημα αντανακλάει τη δυνατότητα του συστήματος να παρέχει τις υπηρεσίες του αξιόπιστα και με χρόνο απόκρισης και διεκπεραιωτική ικανότητα που να είναι αποδεκτές από τους χρήστες του.
- ✧ Η ποιότητα υπηρεσιών είναι ιδιαίτερα σημαντικός παράγων όταν το σύστημα διαχειρίζεται χρονικά κρίσιμα δεδομένα, όπως ροές ήχου ή εικόνας.
  - Σε τέτοιες περιπτώσεις, αν η ποιότητα της υπηρεσίας πέσει κάτω από ένα ελάχιστο επίπεδο τότε η ποιότητα του ήχου ή της εικόνας υποβαθμίζεται τόσο πολύ που δεν είναι δυνατή η κατανόησή τους.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Καταμετρημένα Συστήματα

12

---

---

---

---

---

---

---

---

---

---

## Διαχείριση σφαλμάτων



- ❖ Σε ένα καταμετρημένο σύστημα είναι αναπόφευκτο να υπάρξουν σφάλματα και το σύστημα πρέπει να είναι σχεδιασμένο με τρόπο που να είναι ελαστικό στην αντιμετώπισή τους.

*«Αντιλαμβάνεσαι ότι έχεις ένα καταμετρημένο σύστημα όταν η κατάρρευση ενός συστήματος για το οποίο δεν έχεις ακούσει ποτέ τίποτα σε εμποδίζει από το να εργασθείς.»*

- ❖ Τα καταμετρημένα συστήματα πρέπει να εμπεριέχουν μηχανισμούς για την ανίχνευση συστατικών στοιχείων τα οποία έχουν υποστεί σφάλματα, να μπορούν να συνεχίσουν να παρέχουν όσες περισσότερες πληροφορίες γίνεται παρά την ύπαρξη αυτών των σφαλμάτων και στο βαθμό που είναι δυνατόν να ανακάμπτουν από τα σφάλματα αυτάματα.

---

---

---

---

---

---

---

---

---

---

## Μοντέλα αλληλεπίδρασης



- ❖ Δύο τύποι αλληλεπιδράσεων υπάρχουν μεταξύ των συστατικών στοιχείων ενός καταμετρημένου συστήματος.

- Διαδικασιακή αλληλεπίδραση, όπου ένας υπολογιστής καλεί μία υπηρεσία που γνωρίζει ότι υπάρχει σε κάποιον άλλο υπολογιστή και περιμένει απάντηση από αυτήν.
- Αλληλεπίδραση με ανταλλαγή μηνυμάτων, όπου ένας υπολογιστής στέλνει σε κάποιον άλλο υπολογιστή ένα μήνυμα με πληροφορίες για το τι ακριβώς ζητάει από αυτόν. Σε αυτήν την περίπτωση ο πρώτος υπολογιστής δεν περιμένει κατ' ανάγκη να πάρει απάντηση από τον δεύτερο πριν συνεχίσει την εκτέλεση των υπόλοιπων λειτουργιών του.

---

---

---

---

---

---

---

---

---

---

## Διαδικασιακή αλληλεπίδραση μεταξύ ενός πελάτη και ενός γκαρσονιού



---

---

---

---

---

---

---

---

---

---

## Αλληλεπίδραση με ανταλλαγή μηνυμάτων μεταξύ ενός γκαρσονιού και της κουζίνας



```
<starter>
<dish name = "σούπα" type = "ντοματόσουπα" />
<dish name = "σούπα" type = "ψαρόσουπα" />
<dish name = "τονοσαλάτα" />
</starter>

<main course>
<dish name = "μπριζόλα" type = "κόντρα φιλέτο" cooking = "μέτριο" />
<dish name = "μπριζόλα" type = "φιλέτο" cooking = "σενιάν" />
<dish name = "λαβράκι" />
</main>

<accompaniment>
<dish name = "τηγανητές πατάτες" portions = "2" />
<dish name = "σαλάτα" portions = "1" />
</accompaniment>
```

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Καταναεμμένα Συστήματα

16

---

---

---

---

---

---

---

---

---

---

## Κλήσεις απομακρυσμένων διαδικασιών



- ❖ Η διαδικασιακή επικοινωνία σε ένα καταμεμημένο περιβάλλον υλοποιείται μέσω των κλήσεων απομακρυσμένων διαδικασιών (remote procedure calls – RPC).
- ❖ Σε μία κλήση απομακρυσμένης διαδικασίας, ένα συστατικό στοιχείο καλεί κάποιο άλλο ως το δεύτερο να είναι μία τοπική διαδικασία ή μέθοδος. Το ενδιαμέσο λογισμικό του συστήματος παρεμβάλλεται σε αυτή την κλήση και τη μεταφέρει στο απομακρυσμένο συστατικό στοιχείο.
- ❖ Το τελευταίο αυτό εκτελεί τους αιτούμενους υπολογισμούς και πάλι μέσω του ενδιαμέσου λογισμικού επιστρέφει το αποτέλεσμα στο συστατικό στοιχείο που έκανε την κλήση.
- ❖ Ένα πρόβλημα με την τεχνική RPC είναι ότι και τα δύο μέρη πρέπει να είναι διαθέσιμα τη στιγμή της επικοινωνίας και θα πρέπει να ξέρουν πώς να καλέσουν το ένα το άλλο.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Καταναεμμένα Συστήματα

17

---

---

---

---

---

---

---

---

---

---

## Ανταλλαγή μηνυμάτων



- ❖ Στην αλληλεπίδραση με βάση ανταλλαγή μηνυμάτων, ένα συστατικό στοιχείο δημιουργεί ένα μήνυμα που εμπεριέχει πληροφορίες για τις υπηρεσίες που χρειάζεται από κάποιο άλλο στοιχείο.
- ❖ Το μήνυμα αυτό αποστέλλεται μέσω του ενδιαμέσου λογισμικού στο συστατικό στοιχείο που είναι παραλήπτης του μηνύματος.
- ❖ Το στοιχείο-παραλήπτης αναλύει συντακτικά το μήνυμα, επιτελεί τους υπολογισμούς που τα περιεχόμενα του μηνύματος καθορίζουν και δημιουργεί ένα νέο μήνυμα το οποίο εμπεριέχει τα αποτελέσματα που ζήτησε το στοιχείο-αποστολέας και το οποίο μέσω πάλι του ενδιαμέσου λογισμικού αποστέλλεται σε αυτό το στοιχείο.
- ❖ Στην αλληλεπίδραση με βάση ανταλλαγή μηνυμάτων δεν είναι αναγκαίο για τον αποστολέα και τον παραλήπτη να γνωρίζουν ο ένας τον άλλο. Η επικοινωνία τους γίνεται απλά μέσω του ενδιαμέσου λογισμικού.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Καταναεμμένα Συστήματα

18

---

---

---

---

---

---

---

---

---

---

## Ενδιάμεσο λογισμικό



- ✦ Τα συστατικά στοιχεία ενός καταμεμημένου συστήματος μπορούν να υλοποιηθούν σε διαφορετικές γλώσσες προγραμματισμού και να εκτελούνται σε εντελώς διαφορετικούς τύπους επεξεργαστών. Τα μοντέλα δεδομένων, η αναπαράσταση των πληροφοριών και τα πρωτόκολλα επικοινωνίας μπορεί όλα να είναι διαφορετικά.
- ✦ Το ενδιάμεσο λογισμικό είναι λογισμικό το οποίο μπορεί να διαχειριστεί όλα αυτά τα ετερογενή μέρη ενός καταμεμημένου συστήματος και εγγυάται ότι αυτά μπορούν να επικοινωνούν μεταξύ τους και να ανταλλάσσουν δεδομένα.

---

---

---

---

---

---

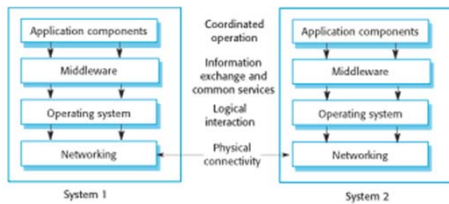
---

---

---

---

## Ενδιάμεσο λογισμικό σε ένα καταμεμημένο σύστημα



---

---

---

---

---

---

---

---

---

---

## Τι υποστηρίζει το ενδιάμεσο λογισμικό



- ✦ Υποστήριξη για αλληλεπίδραση, όπου το ενδιάμεσο λογισμικό συντονίζει αλληλεπιδράσεις μεταξύ διαφορετικών συστατικών στοιχείων του συστήματος.
  - Με τη χρήση ενδιάμεσου λογισμικού δεν είναι απαραίτητο τα συστατικά στοιχεία του συστήματος να γνωρίζουν το φυσικό χώρο στον οποίο βρίσκονται άλλα στοιχεία.
- ✦ Παροχή κοινών υπηρεσιών, όπου το ενδιάμεσο λογισμικό παρέχει επαναχρησιμοποιήσιμες υλοποιήσεις υπηρεσιών τις οποίες μπορεί να χρειάζονται κάποια συστατικά στοιχεία σε ένα καταμεμημένο σύστημα.
  - Με τη χρήση αυτών των κοινών υπηρεσιών, τα συστατικά στοιχεία μπορούν εύκολα να αλληλεπιδράσουν μεταξύ τους και να παρέχουν υπηρεσίες χρήστη με συνεπή τρόπο.

---

---

---

---

---

---

---

---

---

---

### Υπολογισμός πελάτη-διακομιστή



- ✦ Τα κατανεμημένα συστήματα των οποίων η πρόσβαση γίνεται μέσω του διαδικτύου είναι συνήθως οργανωμένα ως συστήματα πελάτη-διακομιστή.
- ✦ Σε ένα τέτοιο σύστημα, ο χρήστης αλληλεπιδρά με ένα πρόγραμμα το οποίο εκτελείται στον τοπικό υπολογιστή (π.χ. ένα πρόγραμμα περιήγησης ιστού ή μία τηλεφωνική εφαρμογή). Αυτό το πρόγραμμα αλληλεπιδρά με κάποιο άλλο πρόγραμμα το οποίο εκτελείται σε έναν απομακρυσμένο υπολογιστή (π.χ. έναν διακομιστή ιστού).
- ✦ Ο απομακρυσμένος υπολογιστής παρέχει υπηρεσίες, όπως πρόσβαση σε ιστοσελίδες, που έτσι καθίστανται διαθέσιμες σε εξωτερικούς πελάτες.

---

---

---

---

---

---

---

---

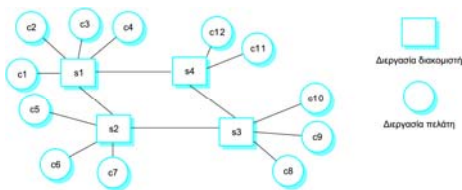
---

---

---

---

### Αλληλεπίδραση πελάτη-διακομιστή




---

---

---

---

---

---

---

---

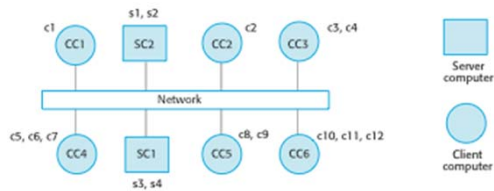
---

---

---

---

### Απεικόνιση των πελατών και των διακομιστών σε ένα σύνολο από δικτυωμένους υπολογιστές




---

---

---

---

---

---

---

---

---

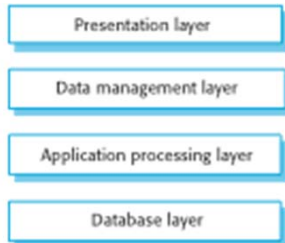
---

---

---



## Διαστρωματωμένο αρχιτεκτονικό μοντέλο για εφαρμογές πελάτη-διακομιστή



Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Κατανεμημένα Συστήματα

25

---

---

---

---

---

---

---

---

## Αρχιτεκτονικά υποδείγματα



- ✦ Οι ευρέως χρησιμοποιούμενοι τρόποι οργάνωσης της αρχιτεκτονικής ενός κατανεμημένου συστήματος είναι οι ακόλουθοι:
  - Αρχιτεκτονική κυρίων-δευτερευουσών μονάδων (master-slave), που χρησιμοποιείται σε συστήματα πραγματικού χρόνου στα οποία πρέπει να υπάρχει εγγυημένος χρόνος απόκρισης κατά την αλληλεπίδραση.
  - Διστρωματική (two-tier) αρχιτεκτονική πελάτη-διακομιστή, που χρησιμοποιείται σε απλά συστήματα πελάτη-διακομιστή και όπου το σύστημα είναι συγκεντρωτικό για λόγους ασφάλειας.
  - Πολυστρωματική (multi-tier) αρχιτεκτονική πελάτη-διακομιστή, που χρησιμοποιείται όταν υπάρχει ένας μεγάλος όγκος από συναλλαγές που πρέπει να υποστούν επεξεργασία από το διακομιστή.
  - Αρχιτεκτονική κατανεμημένων συστατικών στοιχείων, που χρησιμοποιούνται όταν πόροι από διαφορετικά συστήματα και βάσεις δεδομένων πρέπει να συνδυασθούν ή ως μοντέλο υλοποίησης για πολυστρωματικά συστήματα πελάτη-διακομιστή.
  - Ομάτη (peer-to-peer – P2P) αρχιτεκτονική, που χρησιμοποιείται όταν οι πελάτες ανταλλάσσουν πληροφορίες που είναι αποθηκευμένες τοπικά και ο ρόλος του διακομιστή είναι μόνο να συστήνει έναν πελάτη στους υπόλοιπους.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Κατανεμημένα Συστήματα

26

---

---

---

---

---

---

---

---

## Αρχιτεκτονική κυρίων-δευτερευουσών μονάδων



- ✦ Χρησιμοποιούνται συχνά σε συστήματα πραγματικού χρόνου όπου δύναται να υπάρχουν διαφορετικοί επεξεργαστές σχετιζόμενοι με συλλογή δεδομένων από το περιβάλλον του συστήματος, επεξεργασία αυτών των δεδομένων και διαχείριση των μηχανισμών δράσης.
- ✦ Η κύρια διεργασία είναι συνήθως υπεύθυνη για τους υπολογισμούς, το συντονισμό και την επικοινωνία και ελέγχει τις δευτερεύουσες μονάδες.
- ✦ Οι δευτερεύουσες μονάδες ασχολούνται με συγκεκριμένες ενέργειες, όπως τη συλλογή δεδομένων από μία ομάδα αισθητήρων.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Κατανεμημένα Συστήματα

27

---

---

---

---

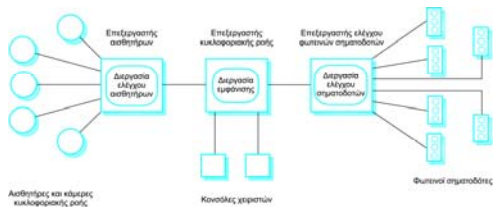
---

---

---

---

### Ένα σύστημα ελέγχου οδικής κυκλοφορίας σχεδιασμένο με αρχιτεκτονική κυρίων- δευτερευουσών μονάδων




---

---

---

---

---

---

---

---

---

---

---

---

### Διστρωματική αρχιτεκτονική πελάτη-διακομιστή



✦ Στη διστρωματική αρχιτεκτονική πελάτη-διακομιστή, το σύστημα υλοποιείται ως ένας ενιαίος λογικός διακομιστής μαζί με ένα ακαθόριστο αριθμό από πελάτες που χρησιμοποιούν το διακομιστή.

✦ Η αρχιτεκτονική αυτή μπορεί να πάρει δύο μορφές:

- Μοντέλο ελαφρύ πελάτη (thin-client), όπου το λογισμικό παρουσίασης εκτελείται στον πελάτη και όλες οι άλλες λειτουργίες (διαχείριση δεδομένων, επεξεργασία εφαρμογής και βάση δεδομένων) εκτελούνται στο διακομιστή.
- Μοντέλο βαρύ πελάτη (fat-client), όπου μέρος ή όλη η επεξεργασία της εφαρμογής εκτελούνται από τον πελάτη και ο διακομιστής είναι υπεύθυνος μόνο για τη διαχείριση των δεδομένων και τη βάση δεδομένων.

---

---

---

---

---

---

---

---

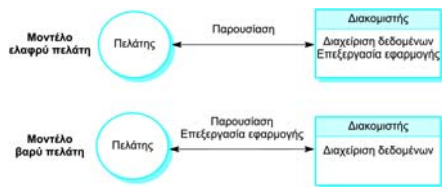
---

---

---

---

### Αρχιτεκτονικό μοντέλο ελαφρών και βαρέων πελατών




---

---

---

---

---

---

---

---

---

---

---

---

### Μοντέλο ελαφρύ πελάτη



- ✦ Χρησιμοποιείται όταν τα κληρονομημένα συστήματα μεταφέρονται σε αρχιτεκτονική πελάτη-διακομιστή.
  - Το κληρονομημένο σύστημα αποτελεί και το ίδιο ένα διακομιστή, ενώ η διασύνδεση με γραφικό περιβάλλον υλοποιείται σε έναν πελάτη.
- ✦ Μεγάλο μειονέκτημα αποτελεί ο μεγάλος επεξεργαστικός φόρτος τόσο για το διακομιστή όσο και για το δίκτυο.

---

---

---

---

---

---

---

---

### Μοντέλο βαρύ πελάτη



- ✦ Μεταβιβάζεται μεγαλύτερος φόρτος επεξεργασίας στον πελάτη αφού οι λειτουργίες επεξεργασίας της εφαρμογής εκτελούνται τοπικά.
- ✦ Ταιριάζει περισσότερο σε νέα συστήματα πελάτη-διακομιστή, στα οποία οι δυνατότητες του συστήματος πελάτη είναι εκ των προτέρων γνωστές.
- ✦ Είναι πιο περίπλοκο από το μοντέλο ελαφρύ πελάτη, ειδικά στον τομέα της διαχείρισης. Οι νέες εκδόσεις της εφαρμογής πρέπει να εγκαθίστανται σε κάθε πελάτη.

---

---

---

---

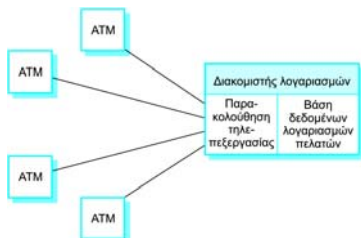
---

---

---

---

### Αρχιτεκτονική βαρύ πελάτη για ένα σύστημα ATM



---

---

---

---

---

---

---

---

## Πολυστρωματική αρχιτεκτονική πελάτη-διακομιστή



- ✦ Σε μία πολυστρωματική αρχιτεκτονική πελάτη-διακομιστή, τα διαφορετικά επίπεδα ενός συστήματος, δηλαδή διαχείριση δεδομένων, επεξεργασία της εφαρμογής και βάση δεδομένων, είναι διαφορετικές διεργασίες που εκτελούνται σε διαφορετικούς επεξεργαστές.
- ✦ Αυτή η προσέγγιση αποφεύγει προβλήματα σχετιζόμενα με επεκτασιμότητα ή απόδοση αν χρησιμοποιείται το μοντέλο του διστρωματικού ελαφρού πελάτη, όπως και προβλήματα διαχείρισης του συστήματος αν γίνεται χρήση του μοντέλου του βαρύτερου πελάτη.

---

---

---

---

---

---

---

---

## Τριστρωματική αρχιτεκτονική ενός διαδικτυακού τραπεζικού συστήματος



---

---

---

---

---

---

---

---

## Χρήση αρχιτεκτονικών υποδειγμάτων πελάτη-διακομιστή (α)



Αρχιτεκτονική	Εφαρμογές
Διστρωματική αρχιτεκτονική Π/Δ με ελαφρείς πελάτες.	Κληρονομημένα συστήματα εφαρμογών, στα οποία ο διαχωρισμός της επεξεργασίας εφαρμογής και της διαχείρισης δεδομένων είναι ανέφικτος. Οι πελάτες μπορούν να έχουν πρόσβαση σε αυτά ως υπηρεσίες. Εφαρμογές με έμφαση στους υπολογισμούς, όπως οι μεταγλωττιστές, οι οποίες έχουν περιορισμένες (ή και μηδαμινές) απαιτήσεις διαχείρισης δεδομένων. Εφαρμογές που χειρίζονται μεγάλο όγκο δεδομένων (για περιήγηση και υποβολή ερωτημάτων), οι οποίες έχουν περιορισμένες ή μηδαμινές απαιτήσεις επεξεργασίας. Τυπικό παράδειγμα εδώ είναι η περιήγηση στο διαδίκτυο.

---

---

---

---

---

---

---

---

## Χρήση αρχιτεκτονικών υποδειγμάτων πελάτη-διακομιστή (β)



Αρχιτεκτονική	Εφαρμογές
Διατρωματική αρχιτεκτονική Π/Δ με βαρείς πελάτες.	Εφαρμογές στις οποίες η επεξεργασία παρέχεται στον πελάτη μέσω έτοιμου λογισμικού (π.χ. Microsoft Excel). Εφαρμογές που απαιτούν επεξεργασία δεδομένων με μεγάλες απαιτήσεις υπολογιστικής ισχύος (π.χ. οπτικοποίηση δεδομένων). Κινητές εφαρμογές, όπου η συνεχής και απρόσκοπτη σύνδεση με το διαδίκτυο δεν είναι εγγυημένη και είναι δυνατή η επεξεργασία τοπικά στο περιβάλλον του πελάτη κάποιων πληροφοριών από τη βάση δεδομένων.
Τριτρωματική ή πολυστρωματική αρχιτεκτονική Π/Δ.	Εφαρμογές μεγάλης κλίμακας με εκατοντάδες ή χιλιάδες πελάτες. Εφαρμογές στις οποίες τόσο τα δεδομένα όσο και η εφαρμογή είναι εμετάβλητα. Εφαρμογές όπου ενosiouνται δεδομένα από πολλές πηγές.

---

---

---

---

---

---

---

---

---

---

## Αρχιτεκτονική κατανεμημένων συστατικών στοιχείων



- ✦ Στην αρχιτεκτονική κατανεμημένων συστατικών στοιχείων δεν γίνεται διάκριση μεταξύ διακομιστών και πελατών.
- ✦ Κάθε οντότητα που μπορεί να κατανεμηθεί είναι ένα συστατικό στοιχείο που παρέχει υπηρεσίες σε άλλα στοιχεία και λαμβάνει υπηρεσίες από άλλα στοιχεία.
- ✦ Η επικοινωνία των συστατικών στοιχείων γίνεται μέσω ενός συστήματος ενδιάμεσου λογισμικού.
- ✦ Ωστόσο ο σχεδιασμός αρχιτεκτονικών κατανεμημένων συστατικών στοιχείων είναι πιο σύνθετος από αυτόν των συστημάτων πελάτη-διακομιστή.

---

---

---

---

---

---

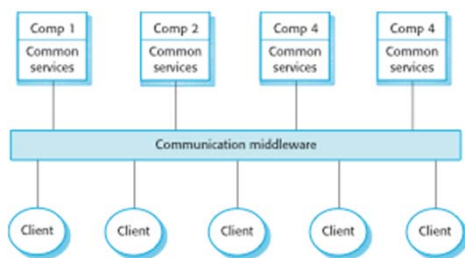
---

---

---

---

## Μία αρχιτεκτονική κατανεμημένων συστατικών στοιχείων



---

---

---

---

---

---

---

---

---

---

### Πλεονεκτήματα της αρχιτεκτονικής κατανεμημένων συστατικών στοιχείων



- ✦ Επιτρέπει στο σχεδιαστή του συστήματος να καθυστερήσει τη λήψη αποφάσεων για το πού και το πώς θα πρέπει να παρέχονται οι υπηρεσίες.
- ✦ Είναι μία πολύ ανοιχτή αρχιτεκτονική συστήματος η οποία επιτρέπει την προσθήκη νέων πόρων όποτε χρειαστεί.
- ✦ Το σύστημα είναι ευέλικτο και επεκτάσιμο.
- ✦ Είναι εφικτή η δυναμική επαναδιευθέτηση του συστήματος μέσω της μετακίνησης συστατικών στοιχείων σε άλλη θέση του δικτύου όταν χρειάζεται.

---

---

---

---

---

---

---

---

---

---

### Χρήσεις αρχιτεκτονικής κατανεμημένων συστατικών στοιχείων



- ✦ Ως λογικού μοντέλου που επιτρέπει τη διάρθρωση και την οργάνωση του συστήματος. Σε αυτή την περίπτωση, εξετάζουμε τον τρόπο με τον οποίο θα διατίθενται οι λειτουργίες της εφαρμογής αποκλειστικά με βάση υπηρεσίες και συνδυασμούς υπηρεσιών.
- ✦ Ως ευέλικτης προσέγγισης για την υλοποίηση συστημάτων πελάτη-διακομιστή. Το λογικό μοντέλο του συστήματος είναι μοντέλο πελάτη-διακομιστή, αλλά τόσο οι πελάτες όσο και οι διακομιστές είναι κατανεμημένα συστατικά στοιχεία που επικοινωνούν μέσω ενός πλαισίου επικοινωνίας.

---

---

---

---

---

---

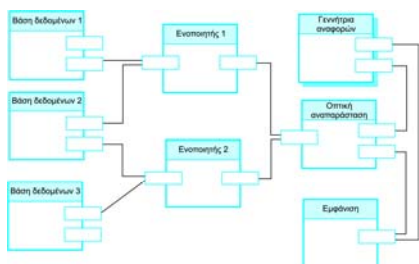
---

---

---

---

### Μία αρχιτεκτονική κατανεμημένων συστατικών στοιχείων για ένα σύστημα εξόρυξης δεδομένων



---

---

---

---

---

---

---

---

---

---

## Σύστημα εξόρυξης δεδομένων



- ✦ Το λογικό μοντέλο του συστήματος δεν είναι αυτό της παροχής υπηρεσιών, στο οποίο απαιτούνται διαφορετικές υπηρεσίες διαχείρισης δεδομένων.
- ✦ Επιτρέπει την αύξηση του πλήθους των βάσεων δεδομένων που είναι διαθέσιμες για προσπέλαση χωρίς διακοπή του συστήματος.
- ✦ Επιτρέπει την εξόρυξη νέων τύπων σχέσεων με την προσθήκη νέων αντικειμένων ενοποίησης.

---

---

---

---

---

---

---

---

## CORBA



- ✦ Το CORBA είναι ένα διεθνές πρότυπο Διαχειριστών Αιτήσεων Διάθεσης Αντικειμένων – ενδιάμεσου λογισμικού για τη διαχείριση επικοινωνιών μεταξύ κατανεμημένων αντικειμένων.
- ✦ Ενδιάμεσο λογισμικό για την υποστήριξη κατανεμημένης επεξεργασίας απαιτείται σε δύο επίπεδα:
  - Στο επίπεδο της λογικής επικοινωνίας, το ενδιάμεσο λογισμικό επιτρέπει σε αντικείμενα που βρίσκονται σε διαφορετικούς υπολογιστές να ανταλλάσσουν δεδομένα και πληροφορίες ελέγχου.
  - Σε επίπεδο συστατικών στοιχείων, το ενδιάμεσο λογισμικό παρέχει μία βάση για την ανάπτυξη συμβατών συστατικών στοιχείων. Έχουν οριστεί πρότυπα συστατικών στοιχείων CORBA.

---

---

---

---

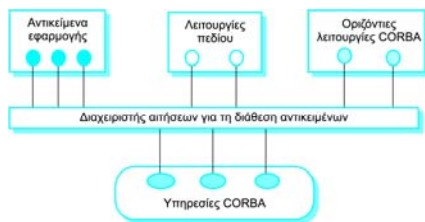
---

---

---

---

## Δομή εφαρμογής βασισμένης στο CORBA



---

---

---

---

---

---

---

---

## Δομή εφαρμογής



- ✧ Αντικείμενα εφαρμογής.
- ✧ Τυποποιημένα αντικείμενα, τα οποία ορίζονται από την OMG για ένα συγκεκριμένο πεδίο εφαρμογής, για παράδειγμα τον κλάδο ασφαλειών.
- ✧ Θεμελιώδεις υπηρεσίες CORBA, όπως η διαχείριση καταλόγων και η προστασία από εξωτερικούς κινδύνους.
- ✧ Οριζόντιες (δηλαδή κοινές σε πολλές εφαρμογές) λειτουργίες, όπως λειτουργίες διασύνδεσης χρήστη.

---

---

---

---

---

---

---

---

---

---

## Πρότυπα CORBA



- ✧ Ένα μοντέλο για αντικείμενα εφαρμογών:
  - Ένα αντικείμενο CORBA αποτελεί την ενθυλάκωση μίας κατάστασης με μία σαφώς καθορισμένη διασύνδεση ανεξάρτητη από γλώσσες προγραμματισμού, η οποία περιγράφεται σε μία γλώσσα ορισμού διασυνδέσεων (Interface Definition Language – IDL).
- ✧ Ένα διαχειριστή αιτήσεων για τη διάθεση αντικειμένων, που διαχειρίζεται τις αιτήσεις για υπηρεσίες των αντικειμένων.
- ✧ Ένα σύνολο υπηρεσιών αντικειμένων, οι οποίες χρησιμεύουν σε πολλές κατανεμημένες εφαρμογές.
- ✧ Ένα σύνολο κοινών συστατικών στοιχείων δομημένων επάνω σε αυτές τις υπηρεσίες.

---

---

---

---

---

---

---

---

---

---

## Αντικείμενα CORBA



- ✧ Τα αντικείμενα CORBA μοιάζουν, στη φιλοσοφία τους, με τα αντικείμενα των C++ και Java.
- ✧ ΠΡΕΠΕΙ να έχουν έναν ξεχωριστό ορισμό διασύνδεσης που εκφράζεται μέσω μίας κοινής γλώσσας (IDL) η οποία μοιάζει με τη C++.
- ✧ Υπάρχει αντιστοιχία από αυτή τη γλώσσα IDL σε άλλες γλώσσες προγραμματισμού (C++, Java, κλπ.).
- ✧ Επομένως, αντικείμενα που είναι γραμμένα σε διαφορετικές γλώσσες μπορούν να επικοινωνούν μεταξύ τους.

---

---

---

---

---

---

---

---

---

---



### Διαχειριστής αιτήσεων διάθεσης αντικειμένων



- ✦ Ο διαχειριστής αιτήσεων διάθεσης αντικειμένων (object request broker – ORB) χειρίζεται την επικοινωνία μεταξύ αντικειμένων. Έχει γνώση όλων των αντικειμένων του συστήματος και των διασυνδέσεών τους.
- ✦ Το αντικείμενο που πραγματοποιεί μία κλήση χρησιμοποιεί ένα διαχειριστή αιτήσεων διάθεσης αντικειμένων για να δεσμεύσει ένα στέλεχος IDL το οποίο ορίζει τη διασύνδεση του καλούμενου αντικειμένου.
- ✦ Η κλήση του στελέχους αυτού έχει αποτέλεσμα την κλήση του διαχειριστή αιτήσεων διάθεσης αντικειμένων, ο οποίος κατόπιν καλεί το απαιτούμενο αντικείμενο μέσω ενός σκελετού IDL που συνδέει τη διασύνδεση με την υλοποίηση της υπηρεσίας.

---

---

---

---

---

---

---

---

---

---

### Επικοινωνίες αντικειμένων μέσω ενός διαχειριστή αιτήσεων διάθεσης αντικειμένων



---

---

---

---

---

---

---

---

---

---

### Επικοινωνία μεταξύ διαχειριστών αιτήσεων διάθεσης αντικειμένων (α)



- ✦ Συνήθως οι διαχειριστές αιτήσεων διάθεσης αντικειμένων δεν είναι ξεχωριστά προγράμματα, αλλά ένα σύνολο αντικειμένων μίας βιβλιοθήκης που είναι συνδεδεμένα με μία εφαρμογή κατά την ανάπτυξή της.
- ✦ Οι διαχειριστές αιτήσεων διάθεσης αντικειμένων χειρίζονται την επικοινωνία μεταξύ αντικειμένων που εκτελούνται στο ίδιο μηχάνημα.
- ✦ Μπορεί να υπάρχουν πολλοί διαθέσιμοι διαχειριστές αιτήσεων διάθεσης αντικειμένων, ενώ επίσης κάθε υπολογιστής που ανήκει σε ένα κατανεμημένο σύστημα θα διαθέτει το δικό του διαχειριστή αιτήσεων διάθεσης αντικειμένων.
- ✦ Η επικοινωνία μεταξύ διαχειριστών αιτήσεων διάθεσης αντικειμένων χρησιμοποιείται στις κλήσεις κατανεμημένων αντικειμένων.

---

---

---

---

---

---

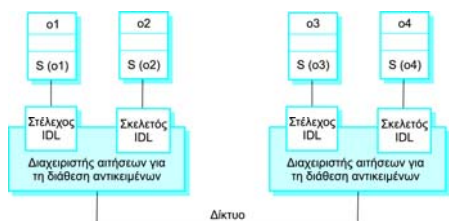
---

---

---

---

## Επικοινωνία μεταξύ διαχειριστών αιτήσεων διάθεσης αντικειμένων (β)



---

---

---

---

---

---

---

---

---

---

## Υπηρεσίες CORBA



- ✦ Υπηρεσίες ονομασίας και διακίνησης.
  - Επιτρέπουν στα αντικείμενα να εντοπίζονται και να αναφέρονται σε άλλα αντικείμενα του δικτύου.
- ✦ Υπηρεσίες ειδοποίησης.
  - Επιτρέπουν στα αντικείμενα να ειδοποιούν άλλα αντικείμενα ότι έλαβε χώρα κάποιο συμβάν.
- ✦ Υπηρεσίες συναλλαγών.
  - Υποστηρίζουν ατομικές συναλλαγές και επαναφορά σε περίπτωση αστοχίας.

---

---

---

---

---

---

---

---

---

---

## Μειονεκτήματα της αρχιτεκτονικής κατανεμημένων συστατικών στοιχείων



- ✦ Οι αρχιτεκτονικές κατανεμημένων συστατικών στοιχείων υποφέρουν από δυο βασικά μειονεκτήματα:
  - Είναι πιο πολύπλοκες στο σχεδιασμό από τα συστήματα πελάτη-διακομιστή και κατ' επέκταση είναι πιο δύσκολο να γίνουν κατανοητές και αντιληπτές από τον κόσμο.
  - Δεν έχει γίνει αποδεκτή η ύπαρξη πρότυπου και τυποποιημένου ενδιάμεσου λογισμικού (όπως το CORBA) από τις εταιρίες ανάπτυξης λογισμικού και διάφοροι παροχείς, όπως η Microsoft και η Sun, έχουν αναπτύξει διαφορετικά και ασύμβατα ενδιάμεσα λογισμικά.
- ✦ Ως αποτέλεσμα αυτών των προβλημάτων, οι υπηρεσιοκεντρικές αρχιτεκτονικές έχουν σε πολλές περιπτώσεις αντικαταστήσει τις αρχιτεκτονικές κατανεμημένων συστατικών στοιχείων .

---

---

---

---

---

---

---

---

---

---

### Ομότιμες αρχιτεκτονικές



- ✦ Τα ομότιμα συστήματα είναι αποκεντρωμένα συστήματα στα οποία κάθε δικτυακός κόμβος μπορεί να πραγματοποιεί υπολογιστικές λειτουργίες.
- ✦ Το συνολικό σύστημα είναι σχεδιασμένο για να εκμεταλλεύεται την υπολογιστική ισχύ και τα αποθηκευτικά μέσα ενός μεγάλου αριθμού υπολογιστών συνδεδεμένων μέσω δικτύου.
- ✦ Τα περισσότερα ομότιμα συστήματα είναι προσωπικά, αλλά παρατηρείται αυξανόμενη χρήση αυτής της τεχνολογίας σε επιχειρησιακό επίπεδο.

---

---

---

---

---

---

---

---

### Αρχιτεκτονικά μοντέλα ομότιμων εφαρμογών



- ✦ Η λογική αρχιτεκτονική του δικτύου.
  - Αποκεντρωτικές αρχιτεκτονικές.
  - Ημισυγκεντρωτικές αρχιτεκτονικές.
- ✦ Αρχιτεκτονική εφαρμογής.
  - Η γενική οργάνωση των συστατικών στοιχείων που απαρτίζουν μία ομότιμη εφαρμογή.
- ✦ Η επικέντρωση εδώ είναι σε αρχιτεκτονικές δικτύων.

---

---

---

---

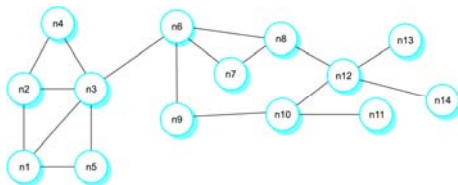
---

---

---

---

### Μία αποκεντρωμένη ομότιμη αρχιτεκτονική



---

---

---

---

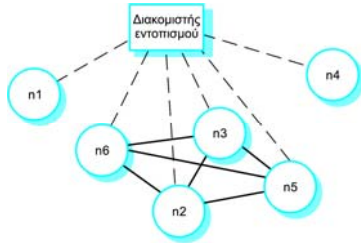
---

---

---

---

## Μία ημισυγκεντρωτική ομότιμη αρχιτεκτονική



Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Κατανεμημένα Συστήματα

58

---

---

---

---

---

---

---

---

## Λογισμικό ως υπηρεσία



✦ Εδώ το λογισμικό βρίσκεται σε κάποιο απομακρυσμένο σημείο και διατίθεται πρόσβαση σε αυτό μέσω του διαδικτύου.

- Το λογισμικό διανέμεται μέσω ενός διακομιστή (ή πιο συχνά μέσω ενός αριθμού διακομιστών) και πρόσβαση σε αυτό γίνεται μέσω ενός περιηγητή ιστού. Δεν εκτελείται σε τοπικό Η/Υ.
- Το λογισμικό ανήκει και το διαχειρίζεται κάποιος παροχέας λογισμικού και όχι ο οργανισμός ο οποίος χρησιμοποιεί το λογισμικό.
- Οι χρήστες πληρώνουν για το λογισμικό με βάση το πόσο συχνά ή πολύ το χρησιμοποιούν ή με βάση κάποια μηνιαία ή ετήσια συνδρομή. Μερικές φορές η χρήση του λογισμικού είναι δωρεάν αλλά οι χρήστες πρέπει να συμφωνήσουν να δέχονται διαφημίσεις.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Κατανεμημένα Συστήματα

59

---

---

---

---

---

---

---

---

## Λογισμικό ως υπηρεσία και υπηρεσιοκεντρικές αρχιτεκτονικές



✦ Το λογισμικό ως υπηρεσία είναι ένας τρόπος για να παρέχεται λειτουργικότητα ενός απομακρυσμένου διακομιστή μέσω της πρόσβασης σε αυτόν από ένα πελάτη με χρήση ενός περιηγητή ιστού. Ο διακομιστής διατηρεί τα δεδομένα και την κατάσταση του χρήστη στη διάρκεια μιας αλληλεπίδρασης αυτού με το σύστημα. Οι συναλλαγές αυτές συνήθως παίρνουν αρκετό χρόνο, π.χ. επεξεργασία ή διάρθρωση ενός κειμένου.

✦ Οι υπηρεσιοκεντρικές αρχιτεκτονικές (Service Oriented Architectures – SOA) είναι μία προσέγγιση για τη δόμηση ενός λογισμικού συστήματος ως ένα σύνολο από ξεχωριστές και μη καταστασιακές (stateless) υπηρεσίες. Αυτές μπορεί να παρέχονται από διαφορετικούς παροχείς και να είναι κατανεμημένες. Συνήθως, οι συναλλαγές είναι σύντομες, όπου κάποια υπηρεσία καλείται, επιτελεί κάποιο έργο και επιστρέφει κάποιο αποτέλεσμα.

Ενότητα 5 (Κεφάλαιο 18) — Τεχνολογία Λογισμικού για Κατανεμημένα Συστήματα

60

---

---

---

---

---

---

---

---

### Παράγοντες που επηρεάζουν την υλοποίηση υπηρεσιοκεντρικών αρχιτεκτονικών



- ❖ **Διευθέτηση**, δηλαδή πώς διευθετείται το λογισμικό για να ανταποκριθεί στις ειδικές ανάγκες του κάθε οργανισμού.
- ❖ **Πολυμίσθωση**, δηλαδή πώς να δίνεται σε κάθε χρήστη η εντύπωση ότι αυτός εργάζεται με το δικό του αντίγραφο του συστήματος, αλλά την ίδια ώρα να γίνεται επίσης αποδοτική χρήση των πόρων του συστήματος.
- ❖ **Επεκτασιμότητα**, δηλαδή ο σχεδιασμός του συστήματος με τρόπο που να είναι δυνατή η επέκτασή του για να ανταποκριθεί σε ένα απρόβλεπτα μεγάλο αριθμό χρηστών.

---

---

---

---

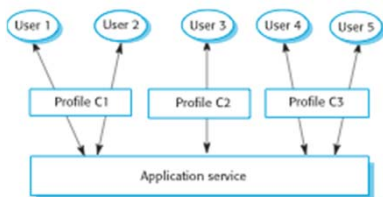
---

---

---

---

### Διευθέτηση ενός λογισμικού συστήματος το οποίο προσφέρεται ως υπηρεσία



---

---

---

---

---

---

---

---

### Είδη παρεχόμενων διευθετήσεων σε μία υπηρεσία



- ❖ **Επωνυμία**, όπου στους χρήστες κάθε οργανισμού παρέχεται ένα περιβάλλον αλληλεπίδρασης που αντανακλάει τον οργανισμό τους.
- ❖ **Επιχειρηματικοί κανόνες και ροές εργασίας**, όπου κάθε οργανισμός καθορίζει τους δικούς του κανόνες για τη χρήση κάποιας υπηρεσίας και των σχετικών δεδομένων της.
- ❖ **Επέκταση βάσης δεδομένων**, όπου ο κάθε οργανισμός ορίζει ο ίδιος πώς το γενικό μοντέλο δεδομένων υπηρεσιών επεκτείνεται για να ανταποκριθεί στις ειδικές ανάγκες του οργανισμού.
- ❖ **Έλεγχος πρόσβασης**, όπου οι πελάτες μίας υπηρεσίας δημιουργούν ατομικούς λογαριασμούς για τους υπαλλήλους τους και ορίζουν τους πόρους και λειτουργίες που θα παρέχονται στον κάθε χρήστη του συστήματος.

---

---

---

---

---

---

---

---

## Πολυμίσθωση



- ✦ Η πολυμίσθωση είναι μία κατάσταση στην οποία πολλοί διαφορετικοί χρήστες έχουν πρόσβαση στο ίδιο σύστημα και η αρχιτεκτονική του συστήματος ορίζεται με τρόπο που επιτρέπει τον αποδοτικό διαμοιρασμό των πόρων του συστήματος.
- ✦ Θα πρέπει να δίνεται στον κάθε χρήστη η εντύπωση ότι έχει αποκλειστική χρήση του συστήματος.
- ✦ Το σύστημα πρέπει να σχεδιασθεί με τέτοιο τρόπο ώστε να υπάρχει ένας απόλυτος διαχωρισμός μεταξύ της λειτουργικότητας του συστήματος και των δεδομένων του.

---

---

---

---

---

---

---

---

---

---

## Μία πολυμίσθωτική βάση δεδομένων



Tenant	Key	Name	Address
234	C100	XYZ Corp	43, Anystreet, Sometown
234	C110	BigCorp	2, Main St, Motown
435	X234	J. Bowie	56, Mill St, Starville
592	PP37	R. Burns	Alloway, Ayrshire

---

---

---

---

---

---

---

---

---

---

## Επεκτασιμότητα



- ✦ Ανέπτυξε τις εφαρμογές με τρόπο που το κάθε συστατικό στοιχείο υλοποιείται ως μία απλή και μη καταστασιακή υπηρεσία η οποία μπορεί να εκτελεστεί σε οποιονδήποτε διακομιστή.
- ✦ Σχεδίασε το σύστημα με χρήση ασύγχρονης αλληλεπίδρασης, έτσι ώστε η εφαρμογή να μη χρειάζεται να περιμένει για το αποτέλεσμα μιας αλληλεπίδρασης (π.χ. μία αίτηση για διάβασμα).
- ✦ Διαχειρίσου τους πόρους, όπως το δίκτυο και τις βάσεις δεδομένων, ως μία δεξαμενή έτσι ώστε κανένας διακομιστής να μη μείνει από πόρους.
- ✦ Σχεδίασε τη βάση δεδομένων σου με τρόπο που να επιτρέπει λεπτομερές (fine-grain) κλειδωμα, έτσι ώστε να μην είναι αναγκαίο το κλειδωμα μίας ολόκληρης εγγραφής όταν μόνο ένα μέρος αυτής της εγγραφής είναι σε χρήση.

---

---

---

---

---

---

---

---

---

---

### Κύρια σημεία (α)



- ✦ Τα πλεονεκτήματα των κατανεμημένων συστημάτων είναι ότι μπορούν να επεκταθούν για να αντιμετωπίσουν αυξημένη ζήτηση, μπορούν να συνεχίσουν την παροχή υπηρεσιών ακόμα και αν μέρος τους υποστεί βλάβη και υποστηρίζουν διαμοιρασμό πόρων.
- ✦ Τα θέματα τα οποία πρέπει να μας απασχολήσουν στη σχεδίαση ενός κατανεμημένου συστήματος είναι η διαφάνεια, ανοικτή λειτουργία, επεκτασιμότητα, ασφάλεια, ποιότητα υπηρεσιών και διαχείριση ασφαλιμάτων.
- ✦ Τα συστήματα πελάτη-διακομιστή είναι δομημένα σε επίπεδα, με το επίπεδο παρουσίασης να υλοποιείται στον υπολογιστή του πελάτη ενώ οι υπηρεσίες της διαχείρισης δεδομένων, εφαρμογής και βάσης δεδομένων υλοποιούνται στο διακομιστή.

---

---

---

---

---

---

---

---

---

---

### Κύρια σημεία (β)



- ✦ Τα συστήματα πελάτη-διακομιστή μπορεί να αποτελούνται από πολλά επίπεδα και το κάθε επίπεδο μπορεί να είναι κατανεμημένο σε διαφορετικούς υπολογιστές.
- ✦ Τα αρχιτεκτονικά υποδείγματα για κατανεμημένα συστήματα περιλαμβάνουν τις αρχιτεκτονικές κυρίων-δευτερευουσών μονάδων, δι-, τρι- και πολυστρωματικές αρχιτεκτονικές πελάτη-διακομιστή, αρχιτεκτονικές κατανεμημένων συστατικών στοιχείων και ομότιμες αρχιτεκτονικές.
- ✦ Τα συστήματα κατανεμημένων συστατικών στοιχείων απαιτούν ενδιάμεσο λογισμικό το οποίο χειρίζεται την επικοινωνία μεταξύ των στοιχείων και επιτρέπει την πρόσθεση και αφαίρεση στοιχείων από το σύστημα.

---

---

---

---

---

---

---

---

---

---

### Κύρια σημεία (γ)



- ✦ Τα πρότυπα CORBA είναι ένα σύνολο προτύπων ενδιάμεσου λογισμικού το οποίο υποστηρίζει αρχιτεκτονικές κατανεμημένων αντικειμένων.
- ✦ Οι ομότιμες αρχιτεκτονικές είναι αποκεντρωμένες και δεν υπάρχει διαχωρισμός μεταξύ πελατών και διακομιστών. Οι υπολογισμοί μπορεί να κατανεμηθούν σε πολλά συστήματα και σε διαφορετικούς οργανισμούς.
- ✦ Το λογισμικό ως υπηρεσία είναι μία προσέγγιση για την παροχή εφαρμογών ως ένα σύστημα ελαφρύ πελάτη-διακομιστή, όπου ο πελάτης είναι ένας περιηγητής ιστού.

---

---

---

---

---

---

---

---

---

---