

Κατανεμημένα Συστήματα

Εισαγωγή

2015-2016

<http://www.cslab.ece.ntua.gr/courses/distrib>

Διαδικαστικά

- Διαλέξεις
 - Τετάρτη **15:15-18:00**, Νέο Κτίριο Ηλεκτρολόγων, Αίθουσα 007
- Διδάσκοντες
 - Καθ. Νεκτάριος Κοζύρης (Παλιό Κτίριο Ηλ. 2.1.7)
 - Δρ. Κατερίνα Δόκα (Παλιό Κτίριο Ηλ. 2.1.45)
- Βαθμολογία
 - Εξαμηνιαία εργασία (πριν τα Χριστούγεννα) – 4 μονάδες
 - Εξέταση – 7 μονάδες

Βιβλία και Υλικό μαθήματος

- Παρουσιάσεις των διαλέξεων
 - www.cslab.ece.ntua.gr/courses/distrib
- Βιβλίο
 - Κατανεμημένα Συστήματα - Αρχές και Υποδείγματα, Andrew S. Tanenbaum, Maarten Van Steen, Εκδόσεις Κλειδάριθμος, κωδικός Ευδόξου: 13777
- Βοηθητικά βιβλία
 - G. Coulouris, J Dollimore, T. Kindberg: Distributed Systems, Concepts and Design
 - N. Lynch: Distributed Algorithms

Απορίες

- Για οποιαδήποτε απορία ή διευκρίνιση

Δρ. Κατερίνα Δόκα

Παλιό Κτίριο Ηλ. 2.1.45

katerina@cslab.ece.ntua.gr

- Γραφτείτε στη λίστα του μαθήματος!!

<http://lists.cslab.ece.ntua.gr/mailman/listinfo/distrib>

Τι είναι ένα κατανεμημένο σύστημα;



Ορισμός

- «Το κατανεμημένο σύστημα είναι μια συλλογή από αυτόνομους υπολογιστές που συνδέονται μεταξύ τους μέσω ενός δικτύου και χρησιμοποιούν ειδικά σχεδιασμένο λογισμικό για την παροχή ενοποιημένων υπολογιστικών υπηρεσιών .» (G. Coulouris)
- «Σε ένα τέτοιο σύστημα οι διεργασίες που εκτελούνται από τους δικτυωμένους υπολογιστές επικοινωνούν μεταξύ τους και συντονίζουν τις κινήσεις τους μόνο μέσω ανταλλαγής μηνυμάτων.» (G. Coulouris)

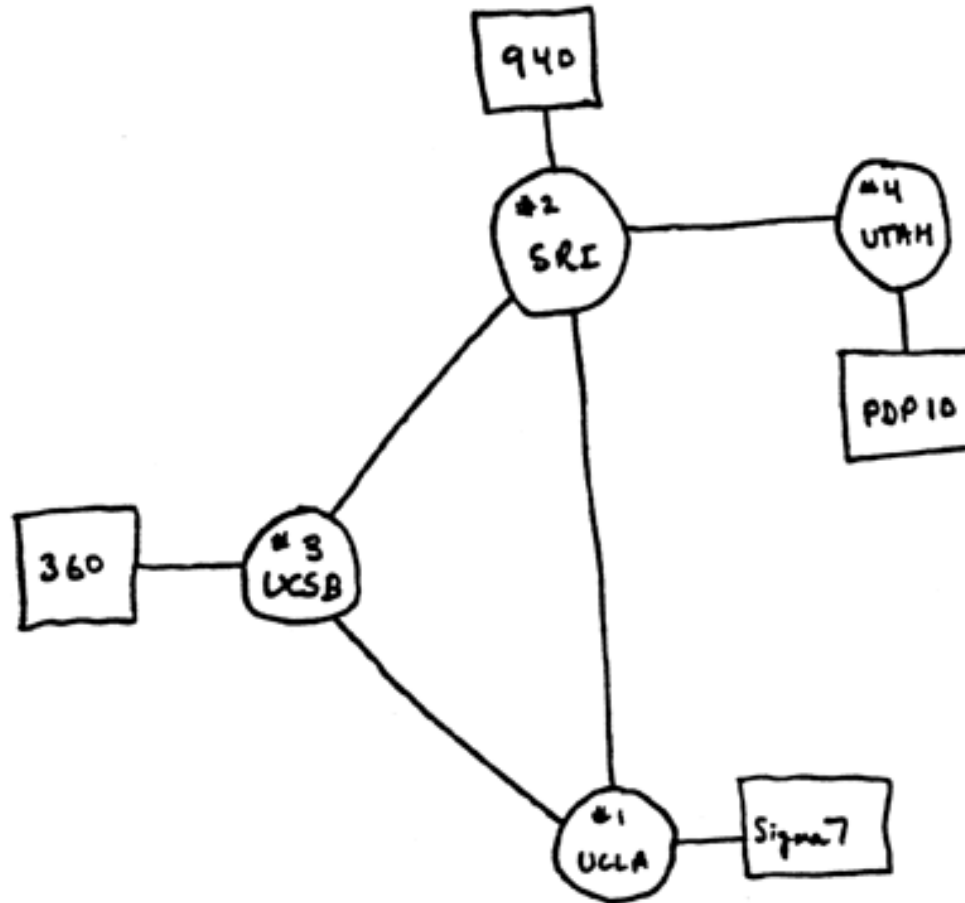
Βασικά Χαρακτηριστικά

- Τα βασικά χαρακτηριστικά των κατανεμημένων συστημάτων είναι
 - Ταυτοχρονισμός των στοιχείων που συμμετέχουν
 - Έλλειψη καθολικής εικόνας του χρόνου
 - Απουσία κοινόχρηστης μνήμης
 - Ενδεχόμενο σφάλματος σε κάθε στοιχείο

Γιατί είναι hot τώρα;

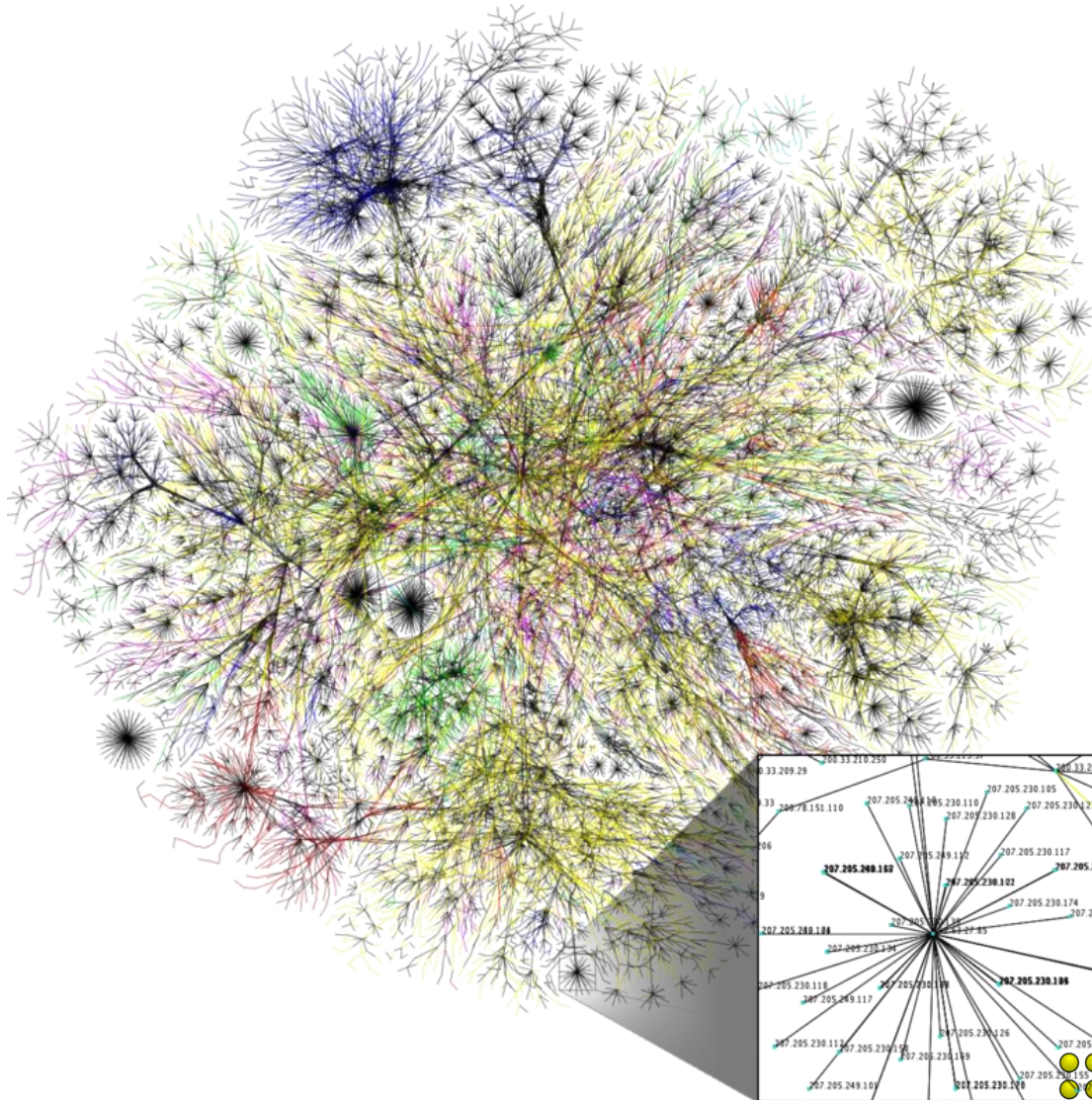
- Δίκτυα
- Επεξεργαστές
- Μνήμη
- Αποθηκευτικά μέσα

To Internet - 1969

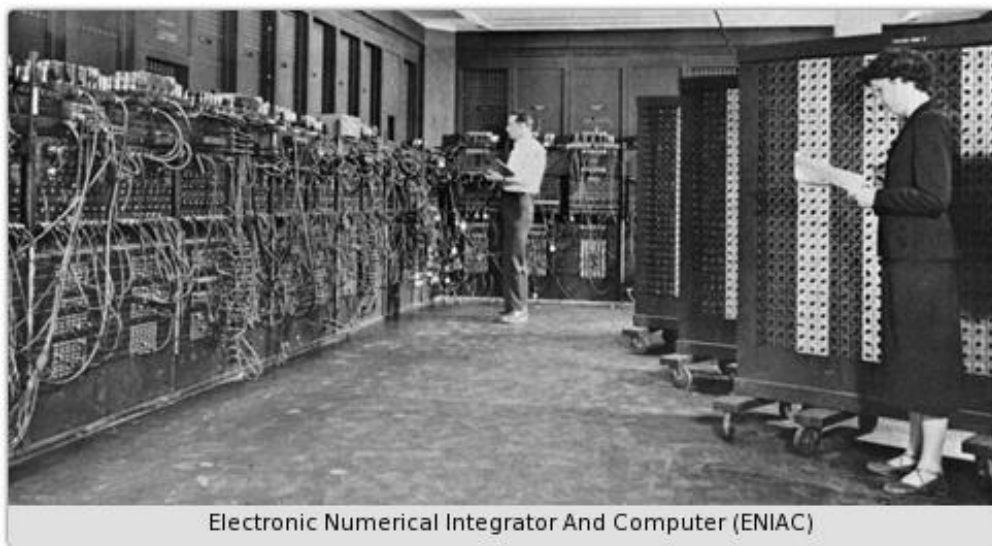


To Internet - σήμερα

>10⁹
nodes



Επεξεργασία



Electronic Numerical Integrator And Computer (ENIAC)

ENIAC: 1 processor

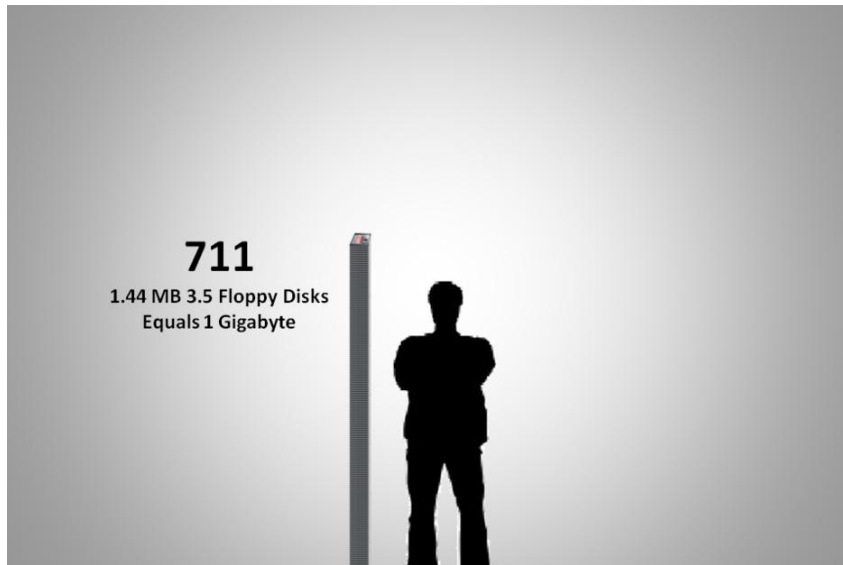


Blue Gene: 250K processors

Μνήμη

- 1977: 256KB, \$32000/MB μνήμης
- 2012: 8GB, 0.5 cent/MB μνήμης

Αποθηκευτικός χώρος



Γιατί τα θέλουμε;

- Αναλογία τιμής/επίδοσης
 - Δεν είναι εύκολη η κλιμάκωση multi-processor
- Κάποιες εφαρμογές είναι κατανεμημένες εκ φύσεως
- Διαδραστική επικοινωνία
 - Messaging, file/photo/video sharing, gaming, telephony
- Απομακρυσμένο περιεχόμενο
 - Web browsing, BitTorrent
- Κινητοί χρήστες
 - Laptops, smartphones, tablets
- Αξιοπιστία
- Σταδιακή αύξηση αναγκών

Trivia

- **Movie rendering:**
 - Η ταινία Disney Cars 2 χρειάζόταν 11.5 ώρες rendering για κάθε frame
 - Το Monsters University απαιτούσε 29 ώρες για κάθε frame
 - Συνολικός χρόνος: πάνω από 100 εκατομύρια CPU hours, 5K AMD processors
- **Google**
 - Εξυπηρετεί πάνω από 40,000 αναζητήσεις το δευτερόλεπτο
 - Δεικτοδοτεί >50 δις σελίδες
 - Χρησιμοποιεί χιλιάδες servers για να επιστρέψει απάντηση σε λιγότερο από δευτερόλεπτο
- **Για τη δεικτοδότηση 50 εκατομ. Σελίδων**
 - 1999: ένας μήνας
 - 2012: ένα λεπτό

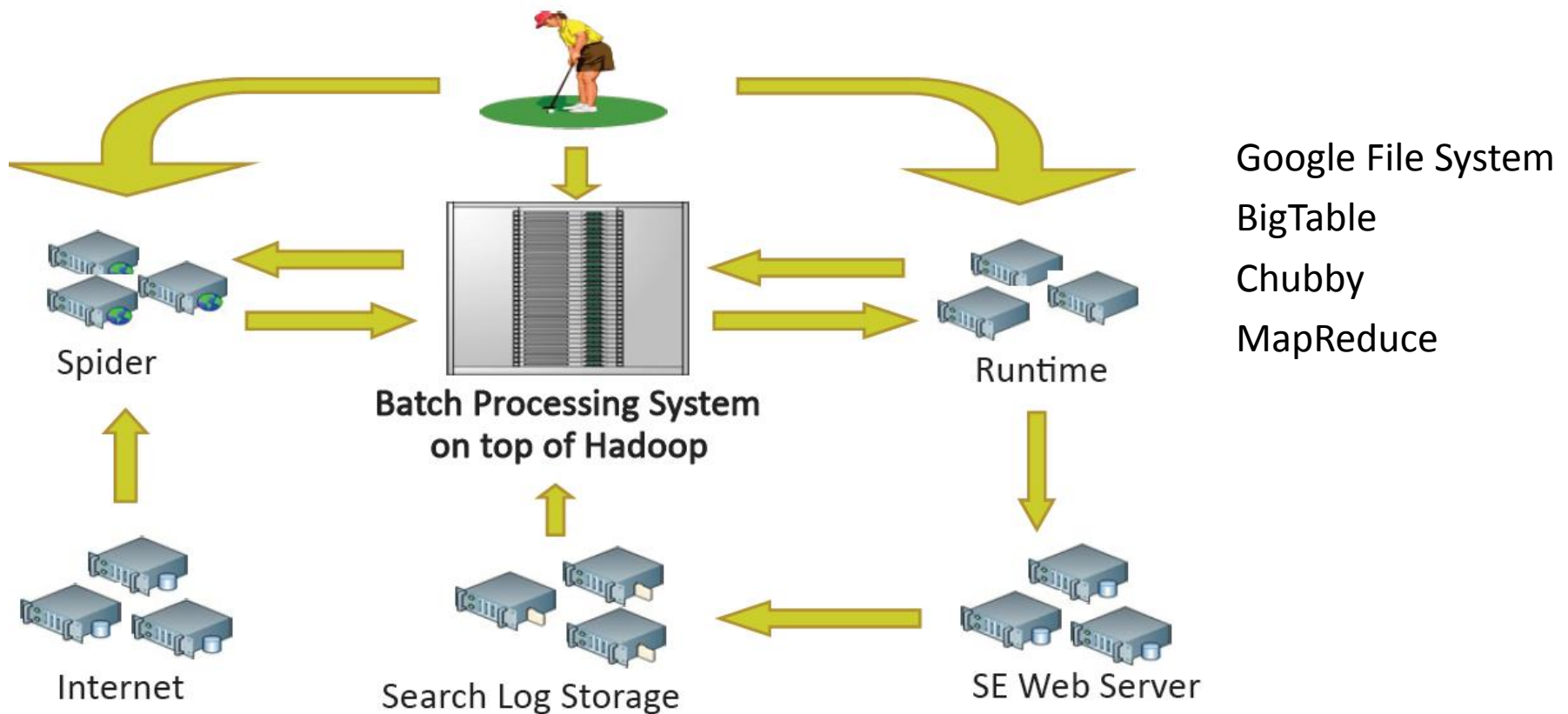
Προκλήσεις

- Είναι δύσκολο να σχεδιάσεις και να υλοποιήσεις ένα κατανεμημένο σύστημα γιατί:
 - Εμπλέκεται μεγάλος αριθμός υπολογιστών
 - Google: 4K
 - Yahoo!: 4K
 - Akamai: 70K
 - Facebook: 60K
 - Οι υπολογιστές αποτυγχάνουν
 - Yahoo!: 50 μηχανήματα αποτυγχάνουν κάθε μέρα (στα 20K)
 - Google: 1 δίσκος αποτυγχάνει ανά 6 ώρες (στους 16K)
 - Lamport: “You know you have a distributed system when the crash of a computer you’ve never heard of stops you from getting the work done”.

Και εμένα τι με νοιάζει;

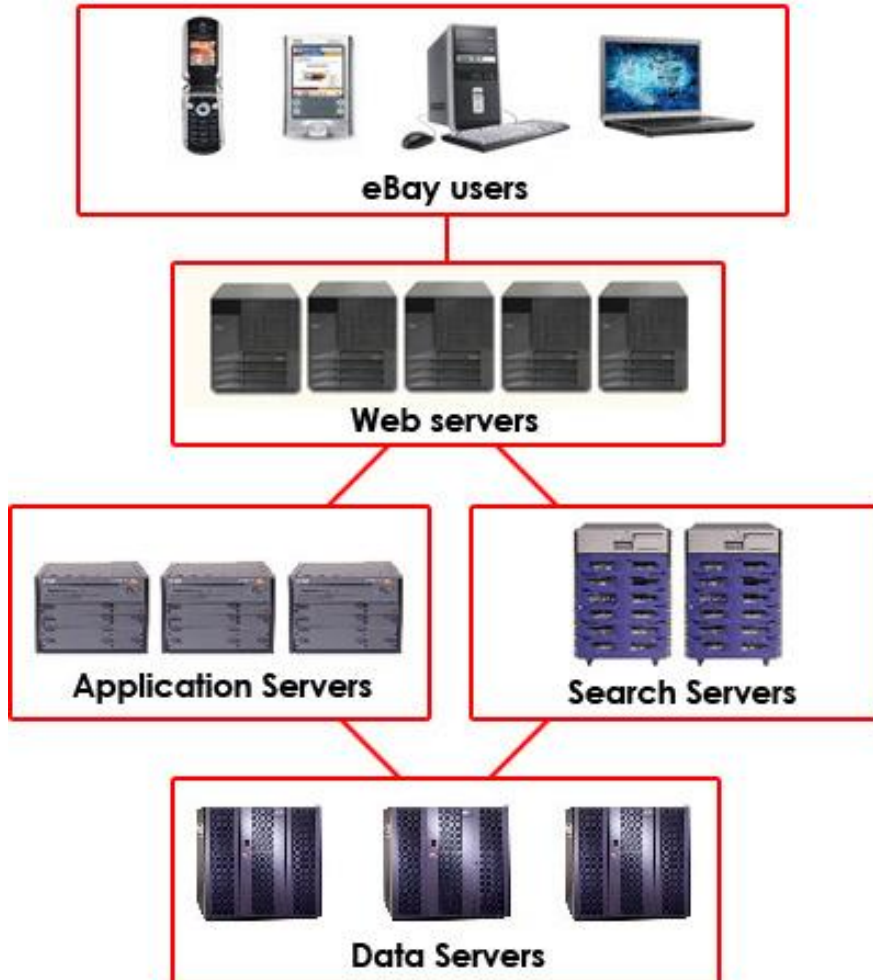
- Μπορεί να είσαι ο επόμενος Turing Award winner
 - Leslie Lamport, Turing Award 2013
- Μπορείς να πιάσεις δουλειά στην Amazon
 - Ο W. Vogels σε blog post για θέσεις στο group του:
“What kind of things am I looking for in you? You know your distributed systems theory.”
- Τα χρησιμοποιείς καθημερινά και δεν το ξέρεις!

Google Web Search



ebay

eBay Infrastructure



©2005 HowStuffWorks

Τι πρέπει να ξέρω;

- Δίκτυα υπολογιστών
 - Βασικές έννοιες (IP, DNS, TCP, κλπ)
- Λειτουργικά συστήματα
 - Συγχρονισμός διεργασιών
- Προγραμματισμός
 - Εμπειρία με threads, sockets, synchronization primitives, κλπ.

Τι θα μάθω;

- Βασικά θέματα καταναεμημένων συστημάτων
 - Πώς επικοινωνούν οι κόμβοι ενός καταναεμημένου συστήματος;
 - Πώς ελέγχεται η πρόσβαση σε πόρους;
 - Πώς έρχονται σε συμφωνία πολλοί υπολογιστές για έναν κοινό σκοπό;
 - Πώς εντοπίζει ένας κόμβος πού βρίσκονται τα δεδομένα και πώς έχει πρόσβαση σε αυτά;
 - Πώς ανιχνεύονται τα σφάλματα;
 - Πώς εξακολουθεί να δουλεύει το σύστημα μετά από σφάλμα;
 - Με ποιον τρόπο τρέχω μια εργασία καταναεμημένα;

Επικοινωνία



Ταυτοχρονισμός



Ομοφωνία



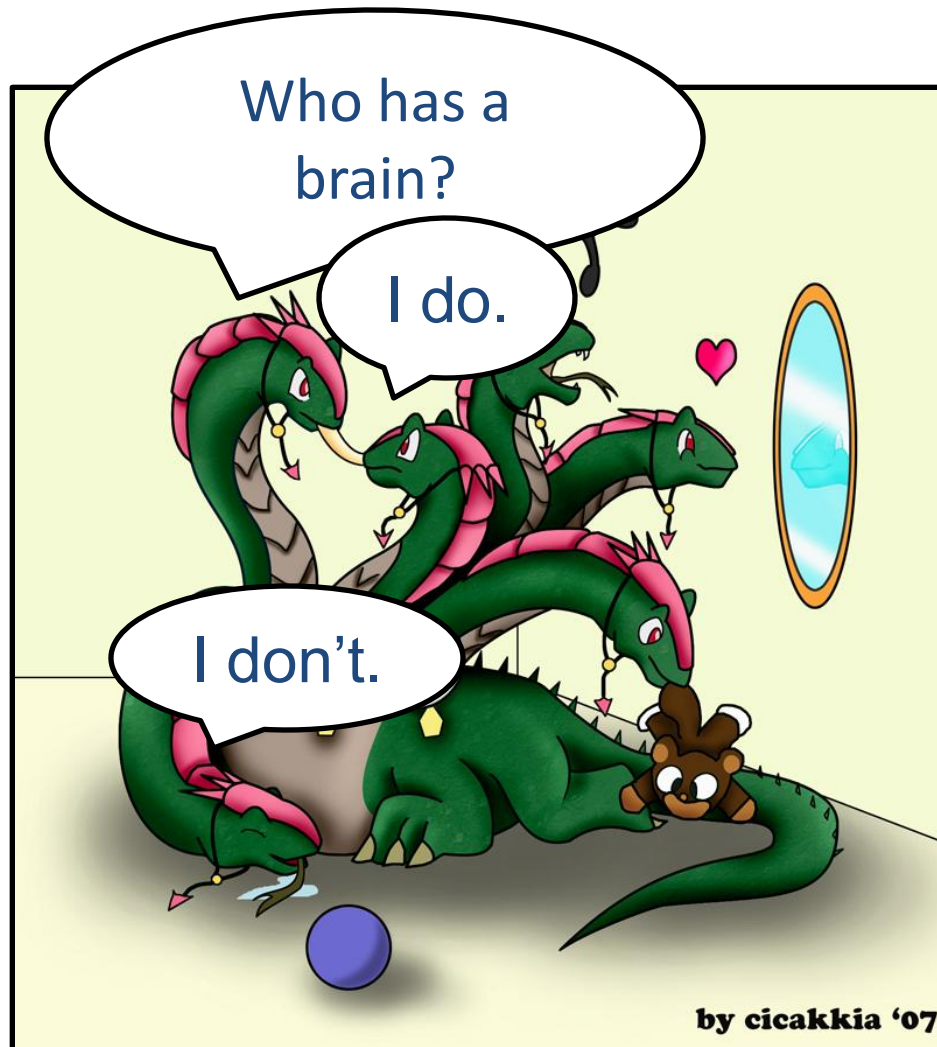
by cicakkia '07 Technical University of Athens

Ανίχνευση Σφαλμάτων



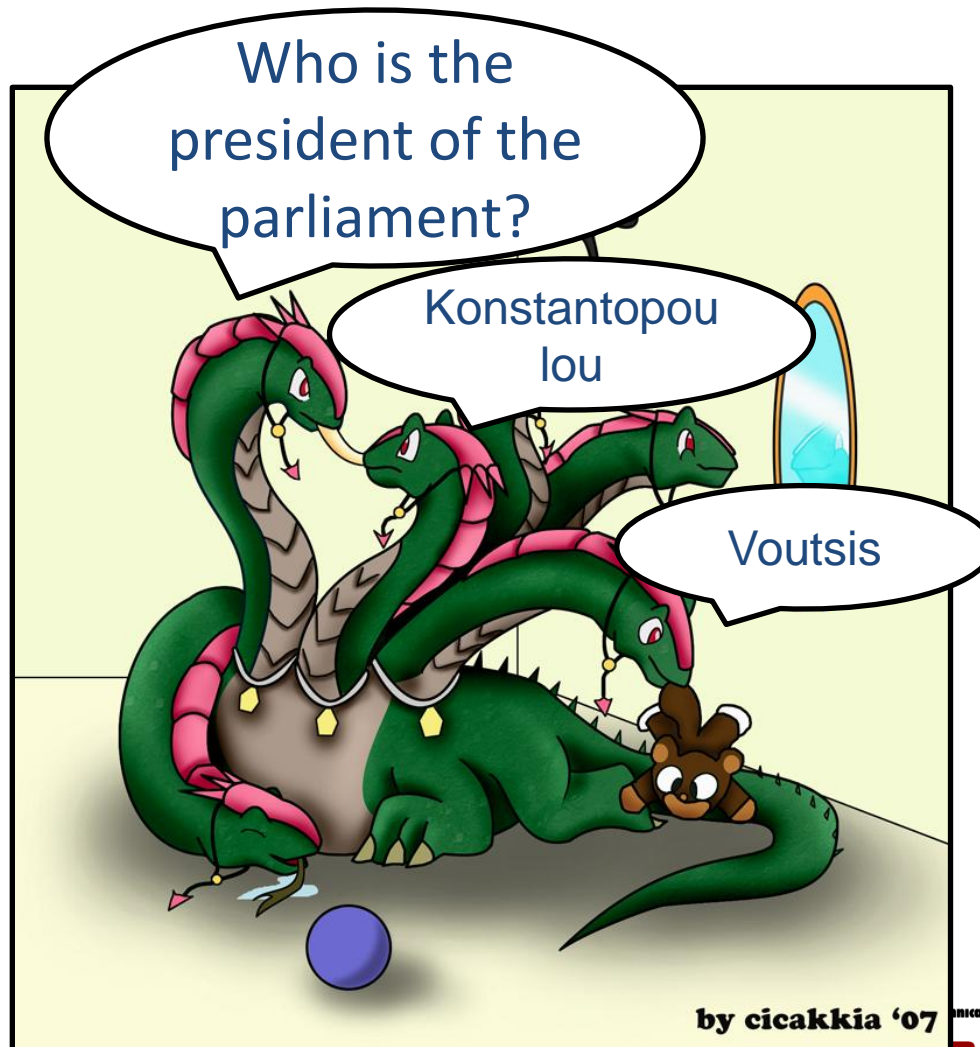
Technical University of Athens

Αποθήκευση δεδομένων



by cicakkia '07 National University of Athens

Διαχείριση αντιγράφων



Technical University of Athens

Ύλη Μαθήματος (1)

- Εισαγωγή
 - Αρχιτεκτονικά μοντέλα
- Συγχρονισμός
 - Φυσικά Ρολόγια
 - Λογικά Ρολόγια
 - Συγχρονισμός φυσικών και λογικών ρολογιών
 - Καθολικές Καταστάσεις
- Κατανεμημένος Συντονισμός
 - Ομαδική επικοινωνία
 - Αλγόριθμοι Εκλογής Αρχηγού
 - Αλγόριθμοι Αμοιβαίου Αποκλεισμού
 - Κατανεμημένος αλγόριθμος ομοφωνίας Paxos

Ύλη Μαθήματος (2)

- Δοσοληψίες
 - Ιδιότητες ACID
 - Έλεγχος ταυτοχρονισμού
 - Κλείδωμα 2 φάσεων
 - Διάταξη χρονοσφραγίδων
 - Αισιόδοξος έλεγχος ταυτοχρονισμού
- Κατανεμημένες Δοσοληψίες
 - Ατομικές δοσοληψίες
 - 2 phase commit
 - 3 phase commit
 - Έλεγχος ταυτοχρονισμού κατανεμημένων δοσοληψιών
 - Αλγόριθμοι εντοπισμού αδιεξόδων
 - Ανάνηψη από σφάλματα

Ύλη Μαθήματος (3)

- Αντίγραφα δεδομένων και διαχείρισή τους
 - Το θεώρημα CAP
 - Μοντέλα συνέπειας
 - Πρωτόκολλο gossip
- Δίκτυα Ομότιμων Κόμβων (P2P)
 - Κατηγορίες Δικτύων P2P
 - Κατανεμημένοι Πίνακες Κατακερματισμού (DHT)
 - Βασικές λειτουργίες DHT
 - Παράδειγμα: το σύστημα Chord

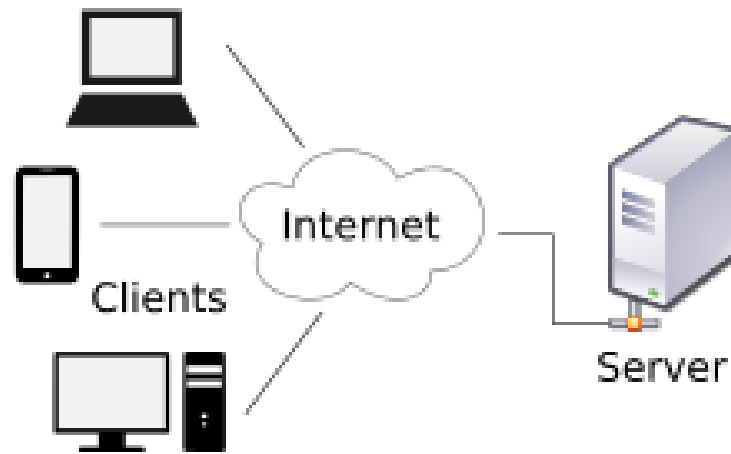
Ύλη Μαθήματος (4)

- Κατανεμημένα File Systems
 - Dropbox, Chubby, Google FS, HDFS
- Μοντέλα Κατανεμημένης Επεξεργασίας
 - Προγραμματιστικό Μοντέλο Map-Reduce
 - Προγραμματιστικό μοντέλο Bulk Synchronous Parallel (BSP)

Κεντρικό μοντέλο

- Παραδοσιακό σύστημα time-sharing
- Η επικοινωνία δεν γίνεται μέσω δικτύου
- Δεν κλιμακώνεται εύκολα
 - Όριο στον αριθμό CPUs ανά system bus
 - Μεγάλο contention για κοινόχρηστους πόρους

Μοντέλο πελάτη-εξυπηρετητή

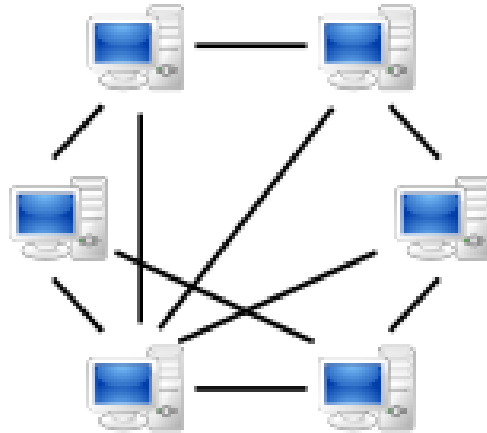


- Ο πελάτης στέλνει αιτήματα στον εξυπηρετητή
- Ο εξυπηρετητής παρέχει πόρους ή υπηρεσίες στους πελάτες
- Οι πελάτες δεν έχουν καμία μεταξύ τους επικοινωνία
- + **Εύκολη υλοποίηση και διαχείριση**
- **Single point of failure, δεν κλιμακώνεται εύκολα**
- E-mail, www, ftp, DNS, κλπ.

Thick και thin clients

- Δύο σχολές για τον διαχωρισμό του λογισμικού μεταξύ πελάτη και εξυπηρετητή
- Thin client: Ο πελάτης εκτελεί το λιγότερο δυνατό processing. Το βαρύ processing εκτελείται στον εξυπηρετητή
 - + Λιγότερες απαιτήσεις σε hardware και τεχνολογία
 - + Καθόλου διαχειριστικό κόστος
 - Latency δικτύου
- Thick client: Το μεγαλύτερο μέρος του processing στον πελάτη
 - +Υψηλές δυνατότητες-χαμηλό κόστος υπολογιστών

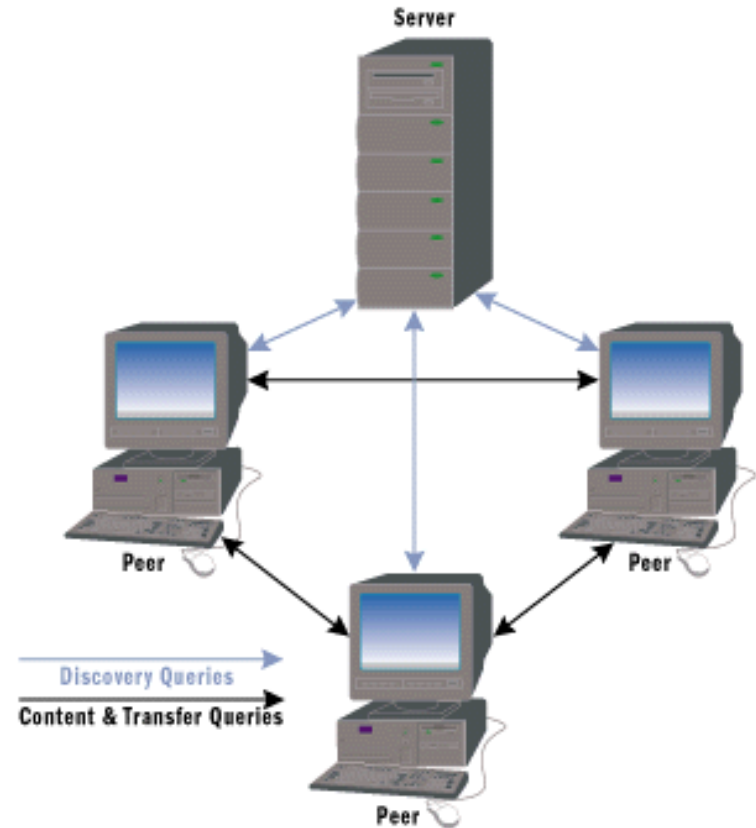
Το μοντέλο ομότιμων κόμβων (P2P)



- Όλοι οι κόμβοι είναι ισότιμοι (και πελάτες και εξυπηρετητές)
- Επικοινωνούν μεταξύ τους
- + **Robustness, scalability, αυτό-οργάνωση**
- **Δύσκολη διαχείριση, ασφάλεια**
- BitTorrent, skype, κλπ.

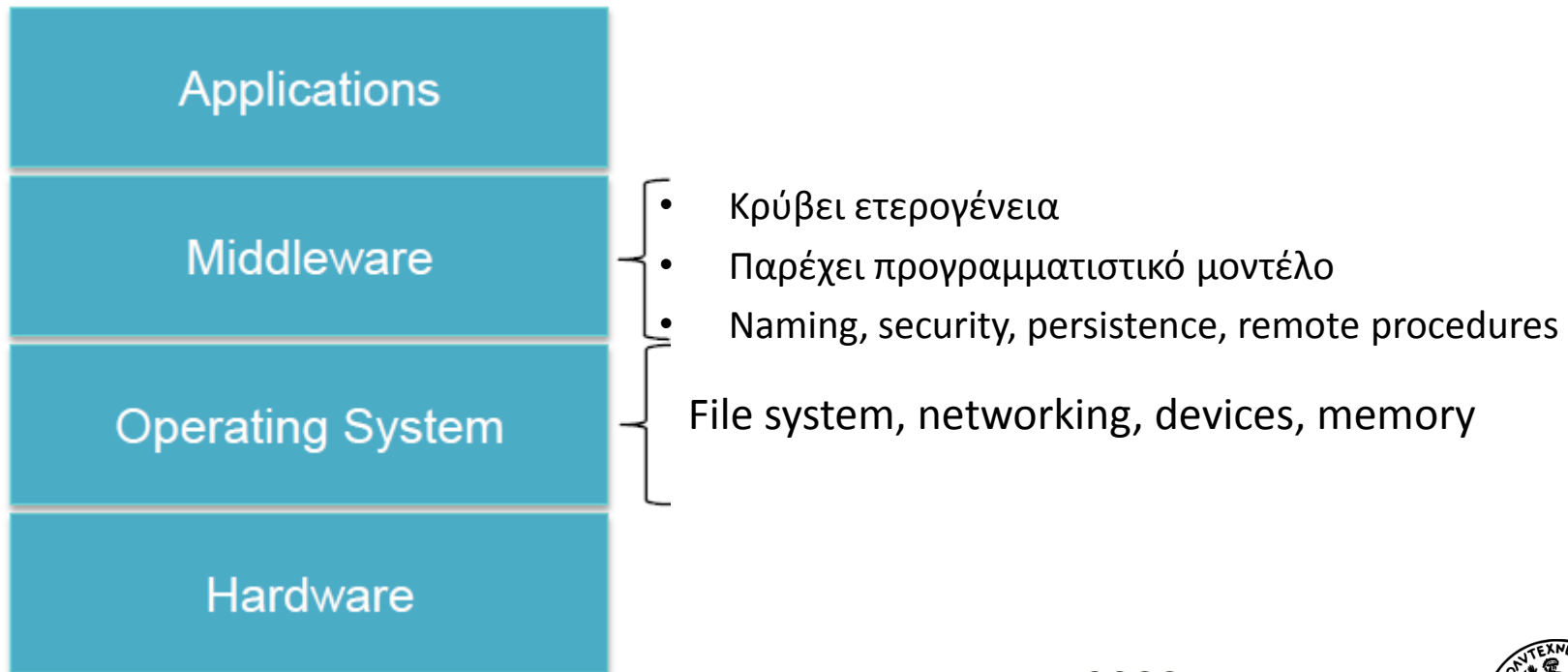
Το υβριδικό μοντέλο

- Συνδυασμός client-server και P2P
- Κεντρικός εξυπηρετητής για
 - Εντοπισμό κόμβων
 - Εντοπισμό περιεχομένων
 - Συντονισμό προσπέλασης



Layered αρχιτεκτονικές

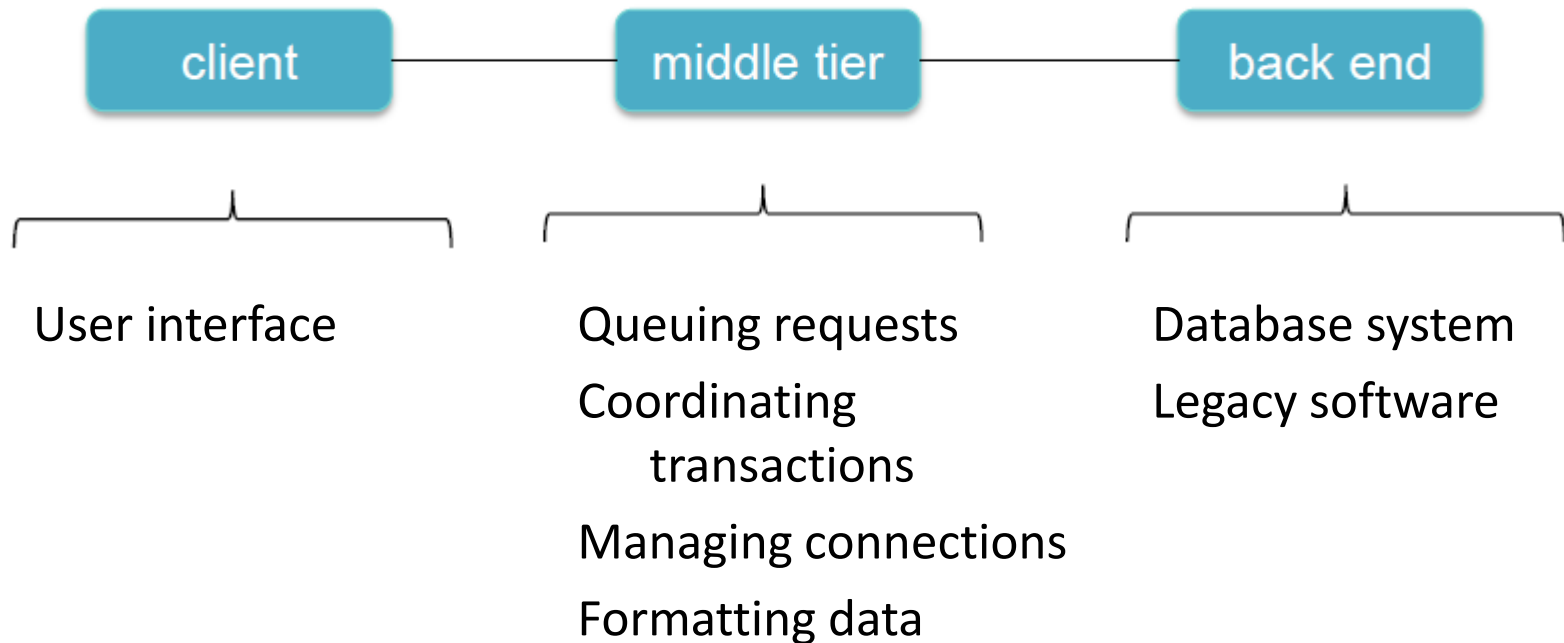
- Πολλά επίπεδα αφαίρεσης
- Κάθε επίπεδο χρησιμοποιεί υπηρεσίες του κατώτερου επιπέδου



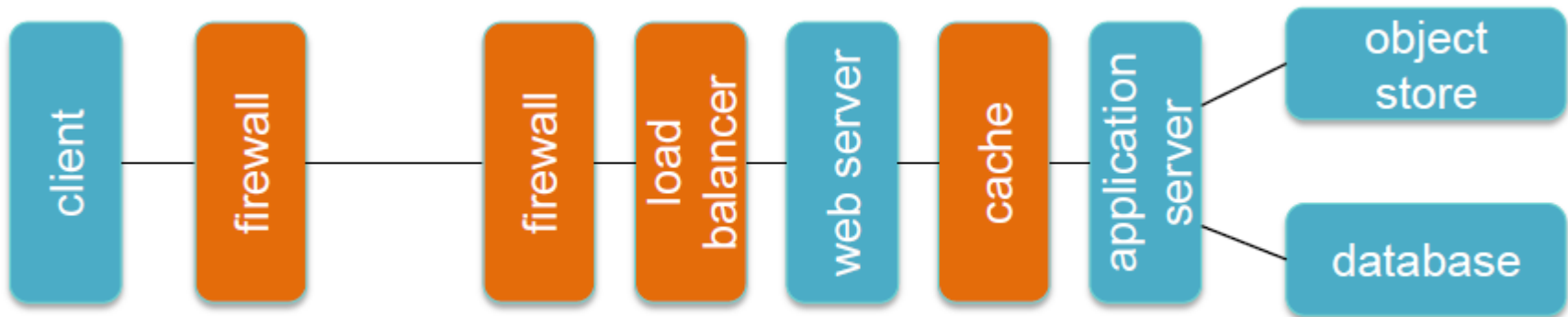
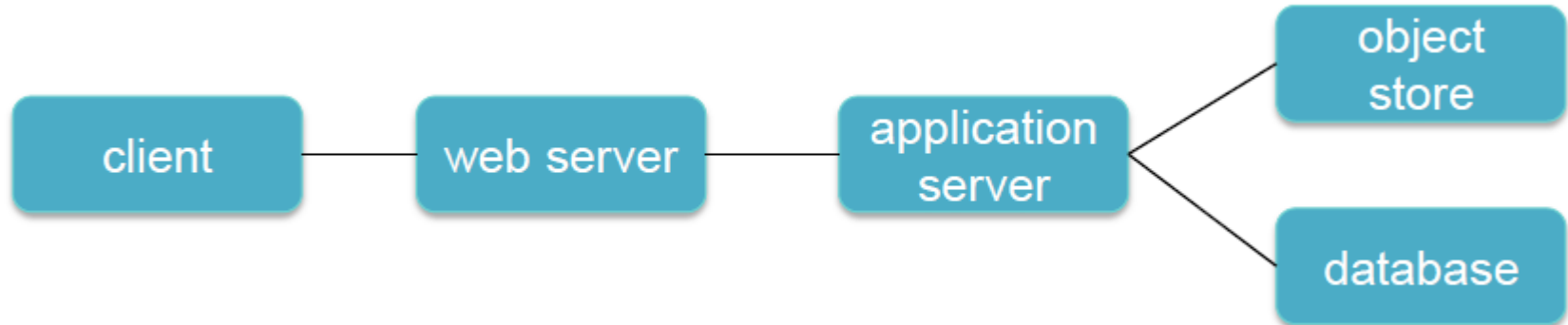
Tiered αρχιτεκτονικές

- Ο διαχωρισμός των λειτουργιών ενός επιπέδου και η κατανομή τους σε servers/κόμβους
- Κάθε tier
 - Ξεχωριστή δικτυακή υπηρεσία
 - Προσπελάζεται από τα γειτονικά tiers
- Το μοντέλο πελάτη εξυπηρετητή είναι μοντέλο 2-tier
 - Server: Υπεύθυνος για backend υπηρεσίες
 - Client: Υπεύθυνος για διάδραση με χρήστη

Multi-tier παράδειγμα



Multi-tier παράδειγμα



Cloud

Ο πόροι διατίθενται ως υπηρεσίες

- Software as a Service (SaaS)

Remotely hosted software

- Salesforce.com, Google Apps, Microsoft Office 365

- Infrastructure as a Service (IaaS)

Compute + storage + networking

- Microsoft Azure, Google Compute Engine, Amazon Web Services

- Platform as a Service (PaaS) Deploy & run web applications without setting up the infrastructure

- Google App Engine, AWS Elastic Beanstalk

- Storage Remote file storage

- Dropbox, Box, Google Drive, OneDrive, ...