



Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Πληροφορικά Συστήματα

7ο Εξάμηνο 2021-22

Αρχιτεκτονική Πληροφοριακών Συστημάτων I

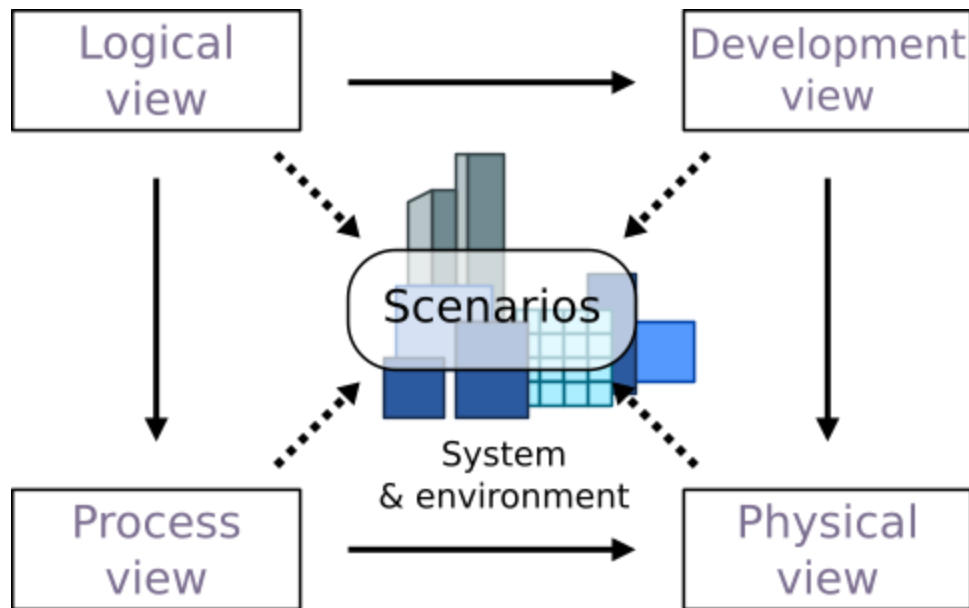
Δρ. Κώστας Σαΐδης (saiko@di.uoa.gr)

Αρχιτεκτονική λογισμικού (γενικά)

Περί τίνος πρόκειται

Η λήψη των θεμελιωδών δομικών και σχεδιαστικών αποφάσεων για το λογισμικό που είναι ακριβό (σε κόπο, χρόνο, χρήμα) να αλλάξουν άπαξ και υλοποιηθούν.

Πολλές αρχιτεκτονικές οπτικές (4+1)



By mpan - Based on File:4+1 Architectural View Model.jpg by User:Mdd, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=50144028>

Ειδικότερα

- Logical view: έμφαση στη λειτουργικότητα του συστήματος σε υψηλό επίπεδο
- Physical view: έμφαση στην τοπολογία και διασύνδεση των συστατικών του σε φυσικό επίπεδο (deployment)
- Development view: έμφαση στην οπτική του προγραμματιστή
- Process view: έμφαση στη δυναμική συμπεριφορά του συστήματος κατά την εκτέλεσή του (απόδοση, κλιμάκωση, κτλ.)
- Scenarios - Use case view: έμφαση στη χρηστική πλευρά του συστήματος και στους σχετικούς ελέγχους αποδοχής

Επιχειρησιακή Αρχιτεκτονική (Enterprise Architecture)

- Ολιστική οπτική των αρχιτεκτονικών ζητημάτων που εμπλέκονται στον Οργανισμό, στα Δεδομένα του, στις Διαδικασίες του και στην ΤΠΕ
- Διατρέχει το σύνολο των ΠΣ του Οργανισμού

Αρχιτεκτονικά πρότυπα (architectural patterns)

Γενικές κι επαναχρησιμοποιήσιμες λύσεις σε κοινά προβλήματα αρχιτεκτονικής.

Θα κουβεντιάσουμε (αλφαβητικά)

- Client-Server
- Component-based
- Layered / N-tier
- Master-slave/Master-replica
- Message-driven/Publish-subscribe
- Microservices

- Peer-to-peer (P2P)
- Pipeline / Pipe-filter
- Representation State Transfer (REST)
- Service-oriented
- Share-nothing architecture
- World Wide Web

Πριν ξεκινήσουμε

Βασικές έννοιες κατανεμημένων συστημάτων

Κατανεμημένα συστήματα

Ένα κατανεμημένο σύστημα αποτελείται από ξεχωριστά συστατικά που:

- Λειτουργούν σε ένα δίκτυο υπολογιστών.
- Επικοινωνούν μεταξύ τους μέσω ανταλλαγής μηνυμάτων.
- Αλληλοεπιδρούν για την επίτευξη ενός κοινού στόχου.

Βασικά χαρακτηριστικά ενός κατανεμημένου συστήματος

- Συνέπεια δεδομένων (Consistency)
- Διαθεσιμότητα συστήματος (Availability)
- Αστοχία δικτύου (Network partition)
- Καθυστέρηση αίτησης/απόκρισης (Latency)
- Αιτήσεις ανά μονάδα χρόνου (Throughput)
- Κλιμάκωση (Scalability)

Consistency (C)

- Η συνέπεια των δεδομένων.
- Ελάχιστο ζητούμενο: κάθε ανάγνωση (read) λαμβάνει την πιο πρόσφατη ενημέρωση (write) ή το σχετικό σφάλμα.

Availability (A)

- Η διαθεσιμότητα της εφαρμογής.
- Ζητούμενο: κάθε αίτηση (request) να λαμβάνει μια απάντηση (μη λάθους).
- Χωρίς να προσφέρονται πάντα εγγυήσεις ότι η απάντηση περιέχει την πιο πρόσφατη ενημέρωση (write).
- Η υψηλή διαθεσιμότητα απαιτεί αντιγραφές (replication).

Network Partition (P)

- Αστοχία δικτύου.
- Επιμερισμός του συστήματος σε "αποσυδεδεμένες" νησίδες.
- Παράδειγμα: απώλεια σύνδεσης με τη βάση δεδομένων.

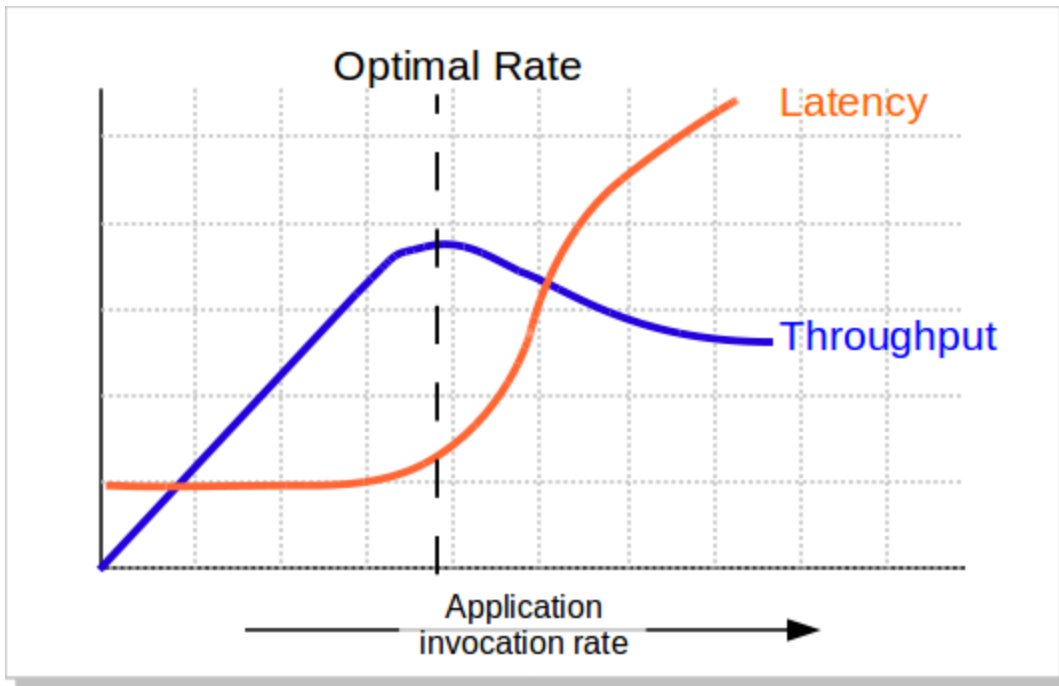
Latency

- Η καθυστέρηση στην απόκριση του συστήματος.
- Ζητούμενο: να ελαχιστοποιηθεί ο χρόνος που απαιτείται για την ικανοποίηση μιας αίτησης.

Throughput

- Το πλήθος των αιτήσεων που ικανοποιούνται από το σύστημα ανά χρονική στιγμή.
- Ζητούμενο: να μεγιστοποιηθεί το πλήθος των αιτήσεων που μπορούν να ικανοποιηθούν ανά χρονική στιγμή.

Latency vs Throughput



docs.voltdb.com

Scalability

Η δυνατότητα ενός συστήματος να επαυξηθεί για να διαχειριστεί αυξημένο φόρτο.

Ειδικότερα

Αν η αύξηση της απόδοσης του συστήματος είναι αναλογική της αύξησης σε υπολογιστικούς πόρους (προσθήκη υλικού), τότε το σύστημα κλιμακώνεται (scales).

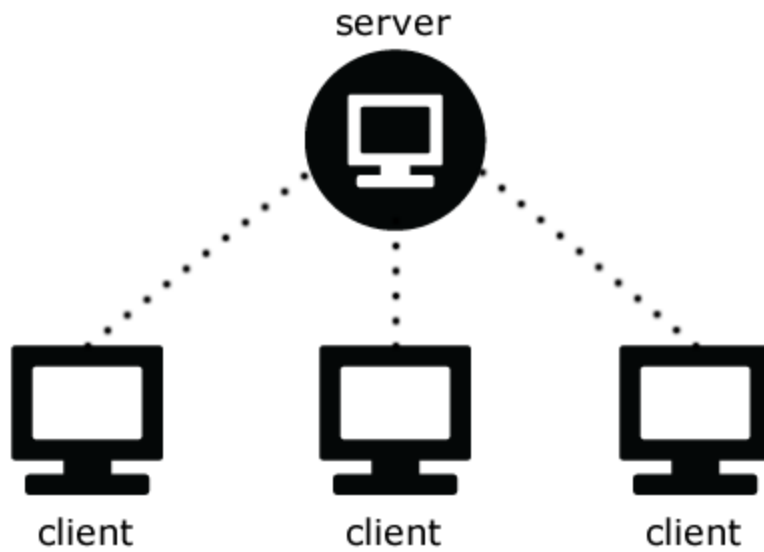
Οριζόντια και κάθετη κλιμάκωση

- Οριζόντια (scale out/in): αύξηση/μείωση των κόμβων.
- Κάθετη (scale up/down): αύξηση/μείωση των πόρων ενός κόμβου.

Επιστροφή

Στα αρχιτεκτονικά πρότυπα

Client-Server



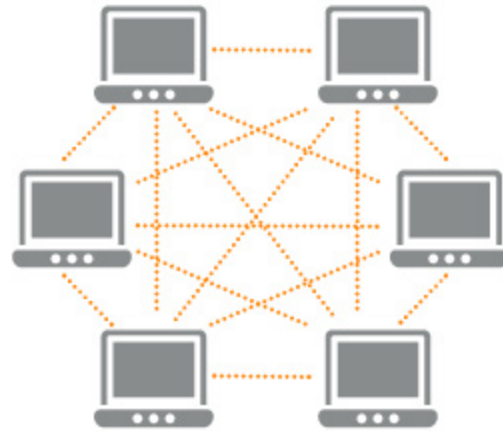
Χαρακτηριστικά

- Server-based
- N clients, 1 server
- Που επικοινωνούν μεταξύ τους με ένα συγκεκριμένο πρωτόκολλο για να υλοποιήσουν μια συγκεκριμένη "εφαρμογή"
- Παραδείγματα: WWW, IMAP, POP3, FTP, SSH, κ.ά

Peer-to-peer (P2P)



Server-Based

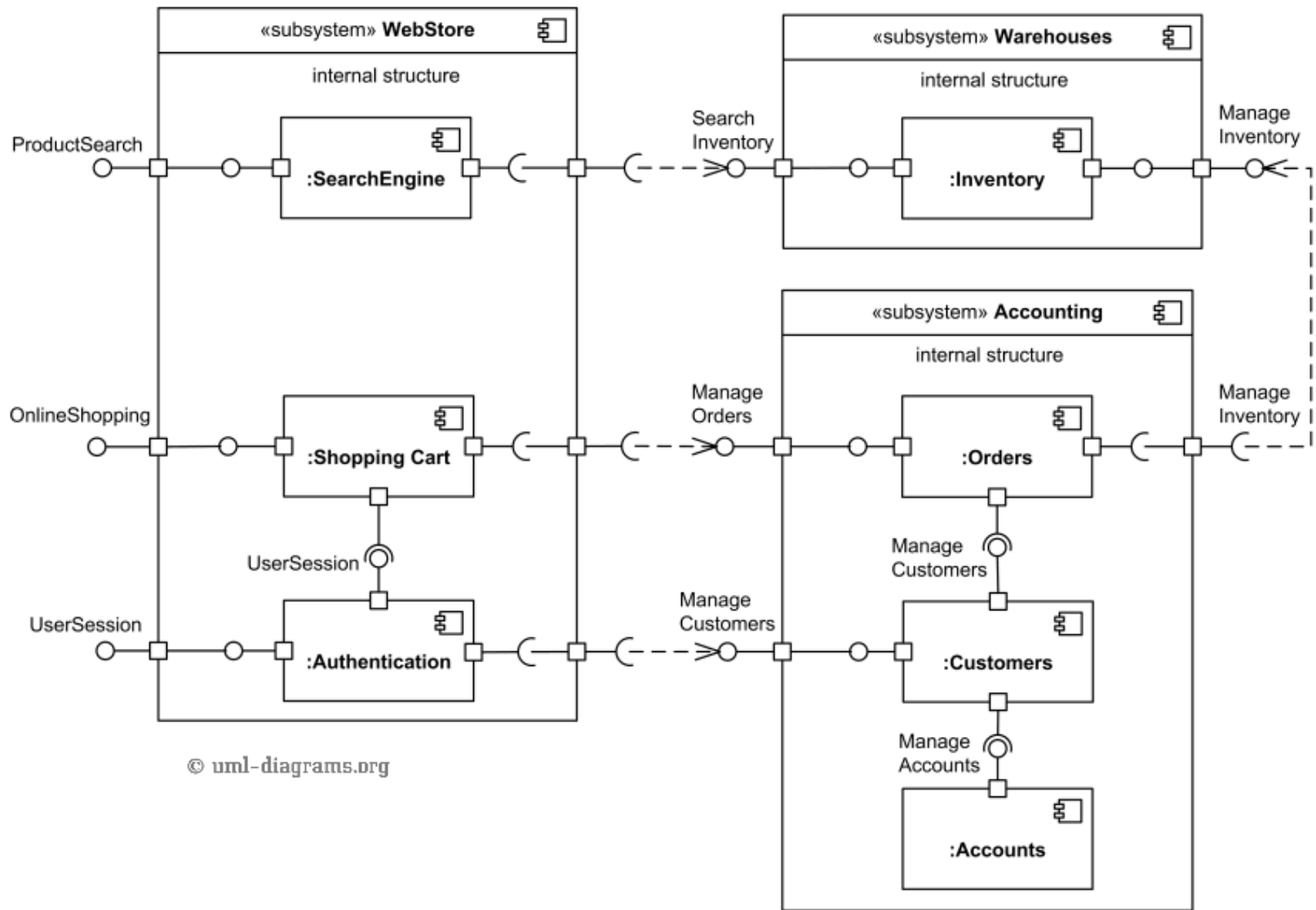


P2P

Χαρακτηριστικά

- Δίκτυο ομότιμων κόμβων
- Κάθε κόμβος είναι και client και server
- Οι κόμβοι επικοινωνούν μεταξύ τους με ένα συγκεκριμένο πρωτόκολλο για να υλοποιήσουν μια συγκεκριμένη "εφαρμογή"
- Παραδείγματα: File-sharing networks, Blockchain, Cryptocurrencies, κ.ά

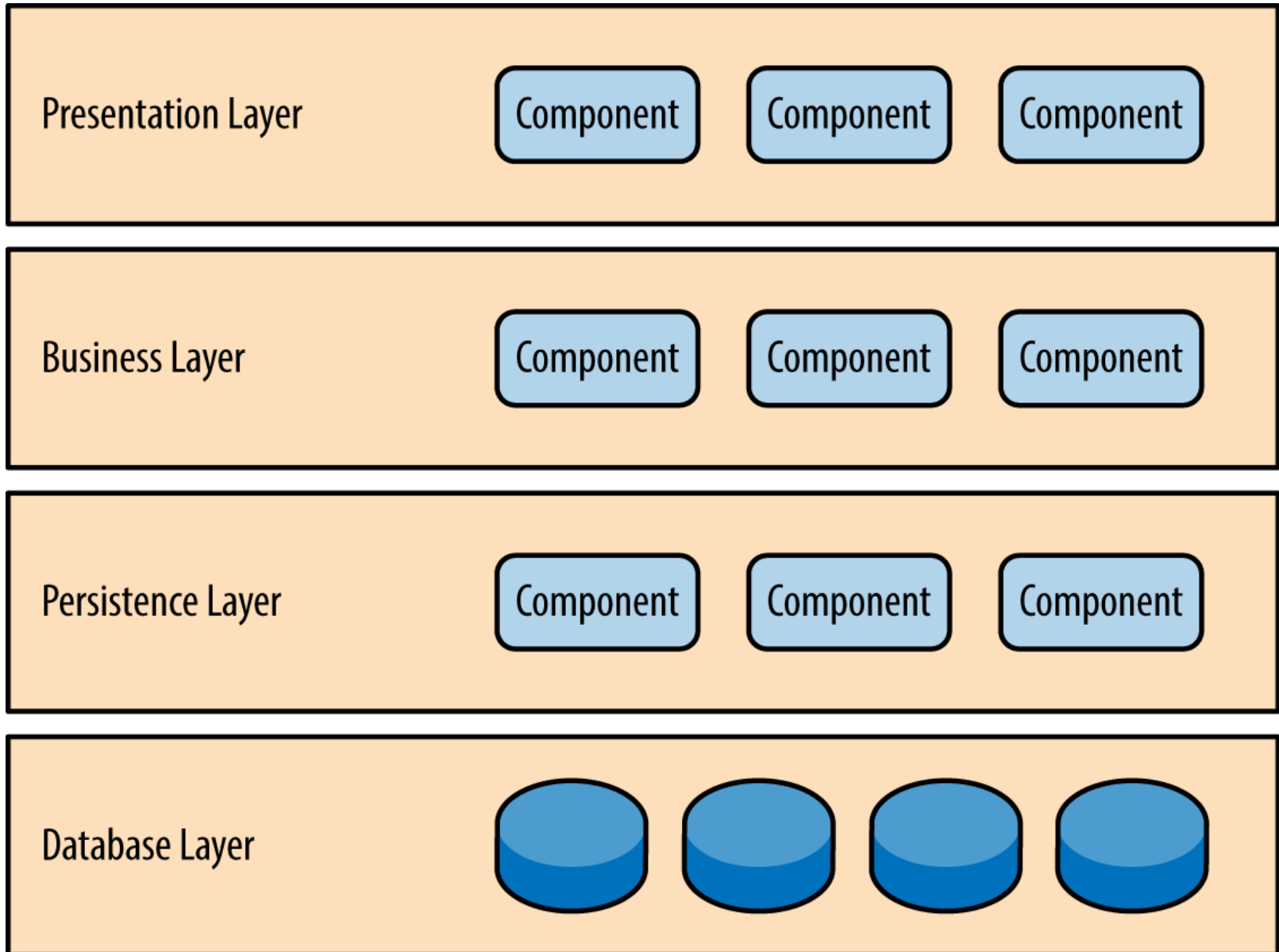
Component-based



Χαρακτηριστικά

- Σχεδιασμός και αλληλεπίδραση των συστατικών του λογισμικού μέσω Interfaces
- Ένα component παρέχει/υλοποιεί ένα interface και απαιτεί την ύπαρξη/κάνει χρήση ενός άλλου
- Χαλαρή σύνδεση (loose coupling) και διαχωρισμός ενδιαφερόντων (separation of concerns)
- Application server: το λογισμικό που φιλοξενεί τα components

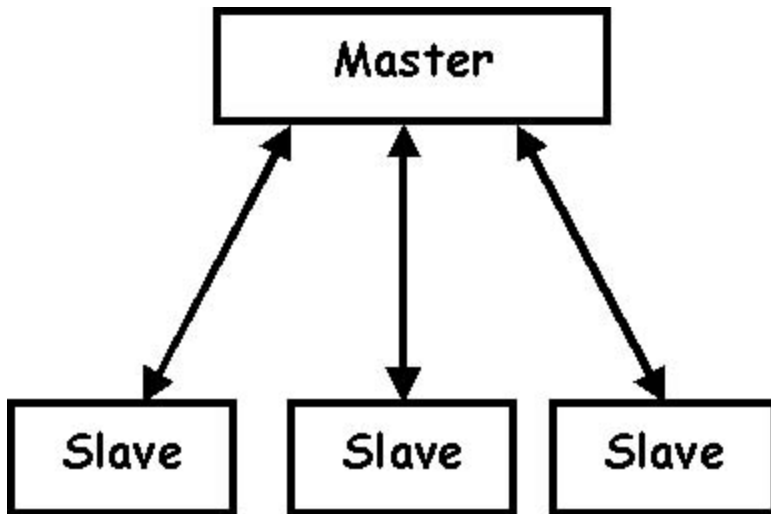
Layered/N-tier



Χαρακτηριστικά

- Server-based
- Λογική ή/και φυσική αρχιτεκτονική
- Ευρεία χρήση στις εφαρμογές διαδικτύου
- Frameworks: παρέχουν έτοιμα προς χρήση και παραμετροποίηση interfaces, components & layers

Master-Slave / Master-Replica



Χαρακτηριστικά

- N slaves, 1 master
- Master (authority), slaves (redundancy)
- Εφαρμογές: υψηλή διαθεσιμότητα, βελτίωση απόδοσης, επιμερισμός φόρτου, κ.ά
- Replication
 - Master copy of data, multiple replicas (slaves)
- Load balancing
 - Master dispatch logic, multiple "worker" nodes