

ΠΑΝΟΣ ΦΙΤΣΙΛΗΣ

# ΣΥΓΧΡΟΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΕΠΙΧΕΙΡΗΣΕΩΝ

ERP-CRM-BPR

2η έκδοση

# Η Αρχιτεκτονική των Πληροφοριακών Συστημάτων Επιχειρήσεων

Καθηγητής Δρ. Πάνος Φιτσιλής

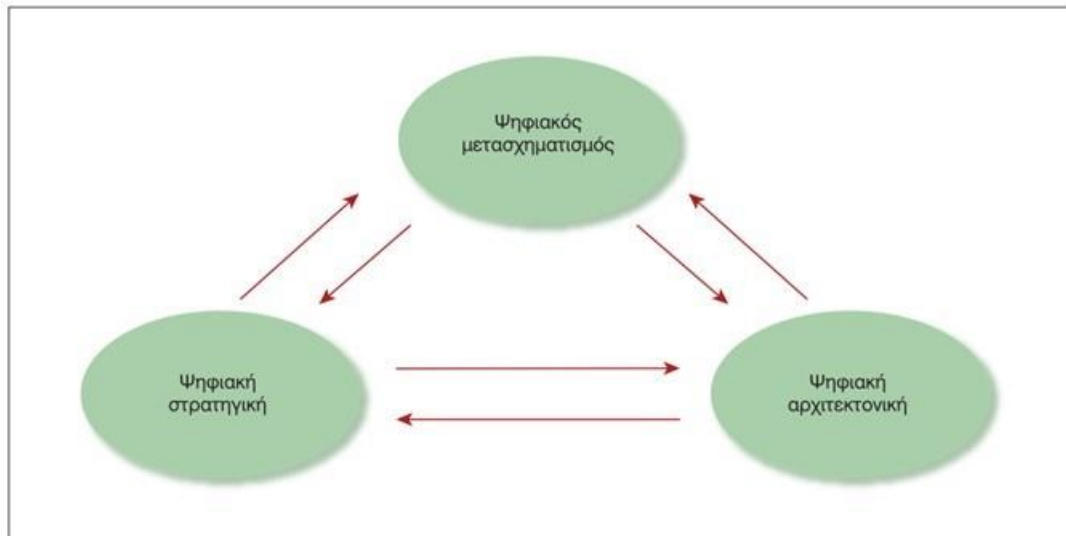


# Στόχοι του κεφαλαίου

- Να κατανοήσετε την έννοια της αρχιτεκτονικής ενός πληροφοριακού συστήματος.
- Να κατανοήσετε τα διαφορετικά επίπεδα και τις διαφορετικές αρχιτεκτονικές όψεις ενός πληροφοριακού συστήματος.
- Να γνωρίζετε ποια είναι τα χαρακτηριστικά μιας επιχειρησιακής αρχιτεκτονικής.
- Να γνωρίζετε ποια είναι τα βασικά αρχιτεκτονικά πρότυπα που χρησιμοποιούνται στην ανάπτυξη πληροφοριακών συστημάτων.
- Να γνωρίζετε τις βασικές παραμέτρους της πληροφοριακής υποδομής μιας επιχείρησης.
- Να γνωρίζετε την έννοια των μη λειτουργικών απαιτήσεων και ποιες είναι αυτές.



# Ψηφιακή στρατηγική και πληροφοριακά συστήματα



**EIKONA 2.1**

Σχέση ψηφιακής στρατηγικής με την αρχιτεκτονική των πληροφοριακών συστημάτων.

Η **ψηφιακή στρατηγική** της επιχείρησης συνδέεται πλέον στενά με την αρχιτεκτονική των πληροφοριακών συστημάτων και επηρεάζει άμεσα το επιχειρηματικό και λειτουργικό μοντέλο της επιχείρησης, τον τρόπο αλληλεπίδρασης με τους πελάτες και τους συνεργάτες, ενώ επιτρέπει στην επιχείρηση να καινοτομεί. Επομένως ο ρόλος της αρχιτεκτονικής των πληροφοριακών συστημάτων είναι σημαντικότερος από ποτέ και είναι αλληλένδετος με την ψηφιακή στρατηγική, αλλά και τον ψηφιακό μετασχηματισμό της επιχείρησης.



# Η αρχιτεκτονική ενός πληροφοριακού συστήματος περιλαμβάνει

- Πιο συγκεκριμένα, η αρχιτεκτονική είναι:
  - ένα σύνολο αποφάσεων με σκοπό την οργάνωση ενός πληροφοριακού συστήματος,
  - η επιλογή των δομικών στοιχείων ενός πληροφοριακού συστήματος καθώς και των διεπαφών τους,
  - η συμπεριφορά των δομικών στοιχείων όπως αυτή καθορίζεται από την αλληλεπίδρασή τους,
  - η σύνθεση των δομικών στοιχείων σε προοδευτικά μεγαλύτερα υποσυστήματα,
  - το αρχιτεκτονικό στυλ που καθοδηγεί την οργάνωση των δομικών στοιχείων (δηλαδή αυτά τα ίδια τα στοιχεία, οι διεπαφές τους, οι συνεργασίες τους και η σύνθεσή τους).

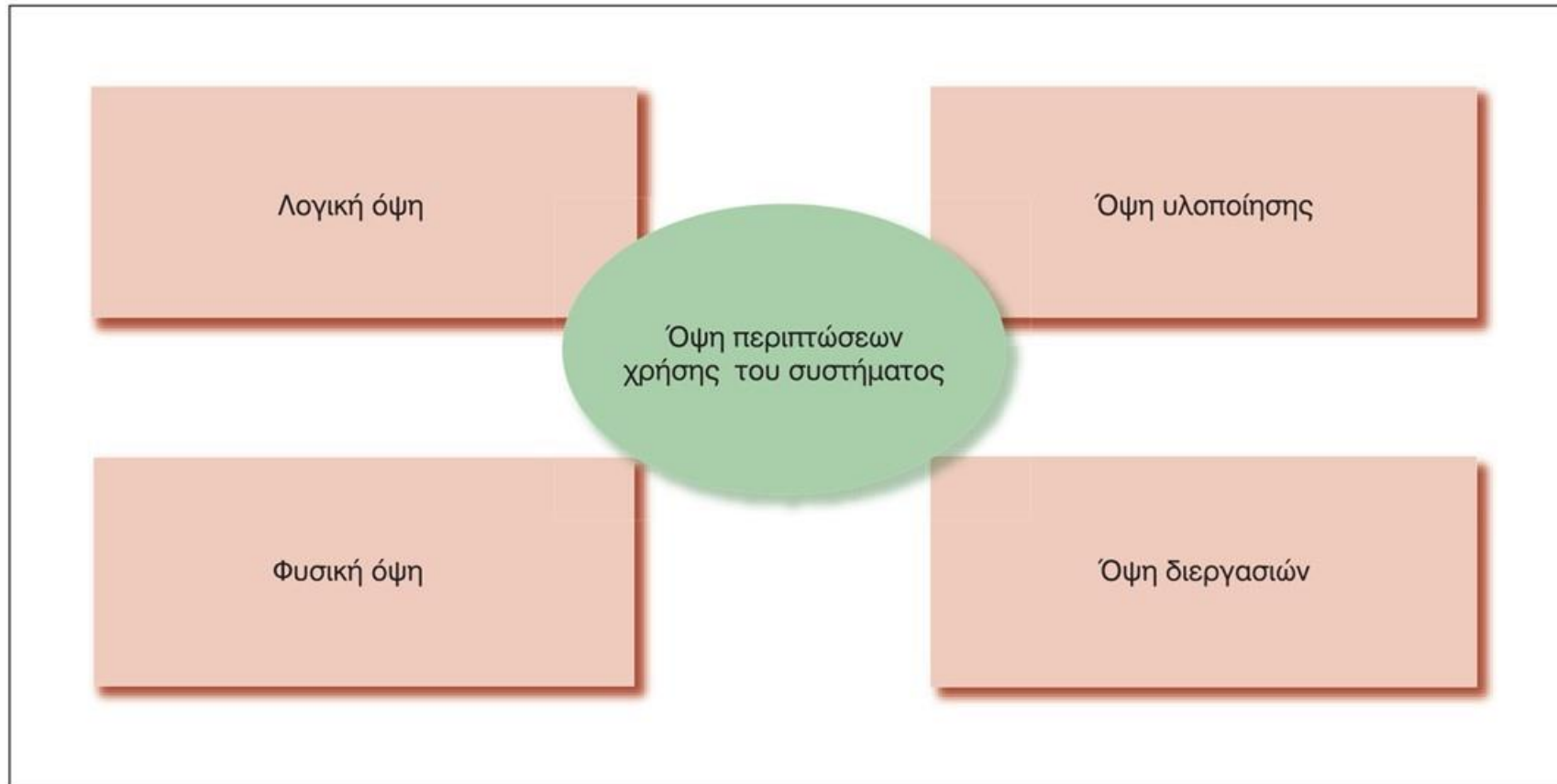


# Οι όψεις ενός πληροφοριακού συστήματος

- Οι πέντε αυτές όψεις είναι:
  - **Λογική όψη.** Περιγράφει με αφηρημένο τρόπο τα συστατικά ενός πληροφοριακού συστήματος. Χρησιμοποιείται για να καταδείξει από τι αποτελείται ένα σύστημα και πώς τα συστατικά αυτά αλληλεπιδρούν μεταξύ τους.
  - **Όψη διεργασιών.** Περιγράφει τις διεργασίες στο εσωτερικό του συστήματος. Είναι ιδιαίτερα χρήσιμη στο να κατανοήσουμε την εσωτερική λειτουργία του συστήματος.
  - **Όψη υλοποίησης.** Περιγράφει τον τρόπο με τον οποίο τα συστατικά του συστήματος είναι οργανωμένα σε ομάδες. Είναι χρήσιμη για τη διαχείριση των επιπέδων της αρχιτεκτονικής του συστήματος.
  - **Φυσική όψη.** Περιγράφει τον τρόπο με τον οποίο ο σχεδιασμός του συστήματος έχει υλοποιηθεί σε υλικό (hardware) και λογισμικό.
  - **Όψη περιπτώσεων χρήσης του συστήματος.** Περιγράφει τη λειτουργικότητα του συστήματος όπως αυτή είναι κατανοητή από τους χρήστες του συστήματος. Η όψη αυτή είναι πάντα αναγκαία καθώς περιγράφει τι θα κάνει το πληροφοριακό σύστημα και αποτελεί την βάση για τις άλλες όψεις. Άλλωστε αυτός είναι ο λόγος που το μοντέλο αυτό ονομάζεται 4 + 1.



# Το μοντέλο του Kruchten



**EIKONA 2.2**

Το μοντέλο του Kruchten των 4+1 όψεων.



# Η αρχιτεκτονική σε τρία διαφορετικά επίπεδα

- **Επίπεδο της επιχειρησιακής αρχιτεκτονικής** (Enterprise Architecture – EA), δηλαδή το πώς το πληροφοριακό σύστημα εντάσσεται γενικότερα στην επιχείρηση.
- **Επίπεδο της αρχιτεκτονικής συστημάτων** (system architecture), δηλαδή το πώς το υλικό (hardware) σε συνεργασία με το λογισμικό (software) συντίθενται ώστε να αποτελέσουν ένα ενιαίο πληροφοριακό σύστημα.
- **Επίπεδο της αρχιτεκτονικής λογισμικού** (software architecture), που παρουσιάζει το πώς τα υποσυστήματα λογισμικού οργανώνονται σε ένα σύστημα λογισμικού.



# Τα επίπεδα της αρχιτεκτονικής

Επίπεδο	Εύρος (scope)	Λεπτομέρεια αρχιτεκτονικής	Αντίκτυπος	Συμμετέχοντες
Επιχειρησιακή αρχιτεκτονική	Όλη η επιχείρηση	Μικρή	Επιχειρηματική στρατηγική	Ανώτατη διοίκηση
Αρχιτεκτονική πληροφοριακού συστήματος	Επιχειρηματικός τομέας	Μεσαία	Στρατηγική επιχειρηματικού τομέα	Διοίκηση τομέα
Αρχιτεκτονική λογισμικού	Επιχειρηματική διεργασία	Μεγάλη	Λειτουργική στρατηγική	Χρήστες

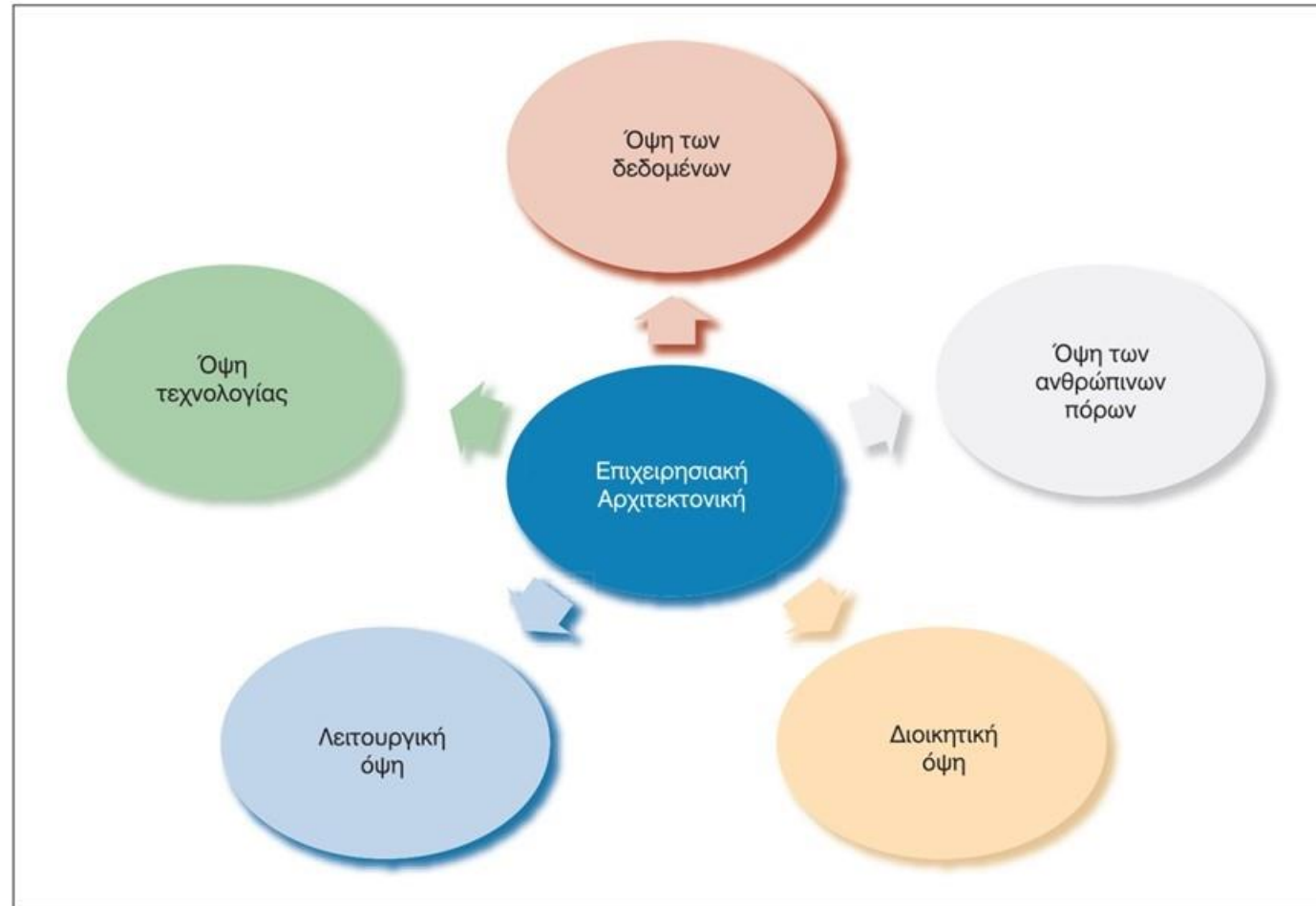
**ΕΙΚΟΝΑ 2.3**

Τα επίπεδα της αρχιτεκτονικής.





# Πληροφοριακό σύστημα και δομικά στοιχεία



**ΕΙΚΟΝΑ 2.4**

Οι όψεις της επιχειρησιακής αρχιτεκτονικής.



# Το πλαίσιο επιχειρησιακής αρχιτεκτονικής Zachman

		Τι	Πώς	Πού	Ποιος	Πότε	Γιατί
1	Όψη ανώτατης διοίκησης – Planner						
2	Επιχειρηματική Όψη - Owner						
3	Αρχιτεκτονική συστήματος – Designer						
4	Όψη κατασκευαστή – Builder						
5	Όψη υπεργολάβου - Subcontractor						
6	Λειτουργία						

ΕΙΚΟΝΑ 2.5

Το πλαίσιο της επιχειρησιακής αρχιτεκτονικής Zachman.



# Οι όψεις του μοντέλου Zachman

- Η **όψη της ανώτατης διοίκησης** (planner). Είναι μια υψηλού επιπέδου αρχιτεκτονική που έχει ως σκοπό να παρουσιάσει το εύρος του συστήματος, πόσο θα κοστίσει το σύστημα και πώς θα συ-σχετίζεται με το ευρύτερο περιβάλλον.
- Η **όψη της διοίκησης της επιχείρησης** (owner). Εστιάζεται στο επιχειρηματικό μοντέλο με το οποίο θα λειτουργεί η επιχείρηση και παρουσιάζει τις επιχειρηματικές οντότητες και διεργασίες και τον τρόπο με τον οποίο συνδέονται (business concepts). Είναι γνωστή και ως άποψη του ιδιοκτήτη της επιχείρησης.
- Η **αρχιτεκτονική συστήματος** (designer). Αντιστοιχεί στο μοντέλο συστήματος όπως σχεδιάζεται από έναν αναλυτή συστημάτων, ο οποίος πρέπει να καθορίσει τη δομή των δεδομένων, τις επιχειρηματικές διεργασίες, καθώς και τις λογικές ροές των διεργασιών.
- Η **όψη του κατασκευαστή** (builder perspective). Η αρχιτεκτονική δοσμένη στο επίπεδο λεπτομέρειας των γλωσσών προγραμματισμού, των συσκευών εισόδου/εξόδου ή άλλων απαιτούμενων τεχνολογιών υποστήριξης, και αφορά τη σχεδίαση του συστήματος (design).
- Η **όψη του υπεργολάβου** (subcontractor perspective). Περιγράφει τις λεπτομερείς απαιτήσεις για τα υποσυστήματα και τα συστατικά των συστημάτων. Αναφέρεται ως η όψη του υπεργολάβου, διότι αφενός κάποια τμήματα της αρχιτεκτονικής κατασκευάζονται από εξωτερικούς συνεργάτες της επιχείρησης, αφετέρου διότι στο επίπεδο αυτό δεν απαιτείται η γενικότερη γνώση της επιχείρησης. Είναι ένα καθαρά τεχνικό επίπεδο.
- Η **όψη της λειτουργίας της επιχείρησης** (functioning enterprise perspective). Είναι το σύστημα όπως αυτό λειτουργεί στην επιχείρηση.



# Η χρήση του μοντέλου Zachman

	Τι (Δεδομένα)	Πώς (Διεργασίες)	Πού (Δίκτυο)	Ποιος (Ανθρώπινο Δυναμικό)	Πότε (time)	Γιατί (motive)
Planner	Απαρίθμηση σημαντικών εννοιών	Λίστα επιχειρηματικών διεργασιών	Λίστα τοποθεσιών λειτουργίας	Λίστα οργανισμών	Επιχειρηματικές απαιτήσεις χρόνου	Επιχειρηματικοί στόχοι
Owner	Σημασιολογικό μοντέλο δεδομένων	Επιχειρηματικό μοντέλο	Επιχειρηματικό μοντέλο εφοδιαστικής	Μοντέλο ροής εργασιών	Κύριο χρονοπρόγραμμα	Επιχειρηματικό πλάνο
Designer	Λογικό μοντέλο δεδομένων	Αρχιτεκτονική εφαρμογών	Αρχιτεκτονική καταναμημένου συστήματος	Αρχιτεκτονική διεπαφής χρηστών	Κύκλοι και επιχειρηματικά γεγονότα	Επιχειρηματικοί κανόνες
Builder	Φυσικό μοντέλο δεδομένων	Σχεδιασμός συστήματος	Τεχνολογική αρχιτεκτονική	Αρχιτεκτονική παρουσίασης	Δομές ελέγχου	Σχεδιασμός κανόνων
Subcontractor	Ορισμός δεδομένων	Προδιαγραφές εφαρμογών	Αρχιτεκτονική δικτύου	Αρχιτεκτονική ασφάλειας	Γεγονότα	Προσδιορισμός κανόνων
Σύστημα σε λειτουργία	Δεδομένα	Διεργασίες	Δίκτυο	Οργάνωση	Χρονοπρόγραμμα	Στρατηγική

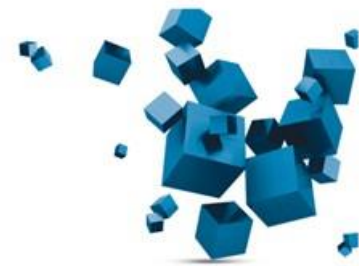
**EIKONA 2.6**

Τα μοντέλα της επιχειρησιακής αρχιτεκτονικής Zachman.



# Η αρχιτεκτονική πρέπει να λάβει υπόψη τις ακόλουθες απαιτήσεις

- Τις επιχειρησιακές διεργασίες που θα υποστηρίξει το πληροφοριακό σύστημα.
- Τις υπάρχουσες υποδομές Τεχνολογιών Πληροφορικής και Τηλεπικοινωνιών (ΤΠΕ) στην επιχείρηση.
- Τις ανάγκες που θα δημιουργήσει το πληροφοριακό σύστημα. Για παράδειγμα, ανάγκες σε ποιότητα και διαφύλαξη δεδομένων, ανάγκες σε ασφάλεια, σε διαχειρησιμότητα και σε επεκτασιμότητα.
- Τα εμπλεκόμενα μέρη στη διαμόρφωση της αρχιτεκτονικής και τους χρήστες του συστήματος.
- Την εμβέλεια της επιχείρησης. Η εμβέλεια αφορά τη γεωγραφική έκταση στην οποία δραστηριοποιείται η επιχείρηση και επηρεάζει, εκτός από το μέγεθος, διάφορες παραμέτρους του συστήματος (π.χ. χρησιμοποιούμενες γλώσσες, χρονικές ζώνες).
- Τα δεδομένα που παράγονται και διατηρούνται στο σύστημα.



# Απλές αρχιτεκτονικές

## Αρχιτεκτονική πελάτη/ εξυπηρετητή (client/server)

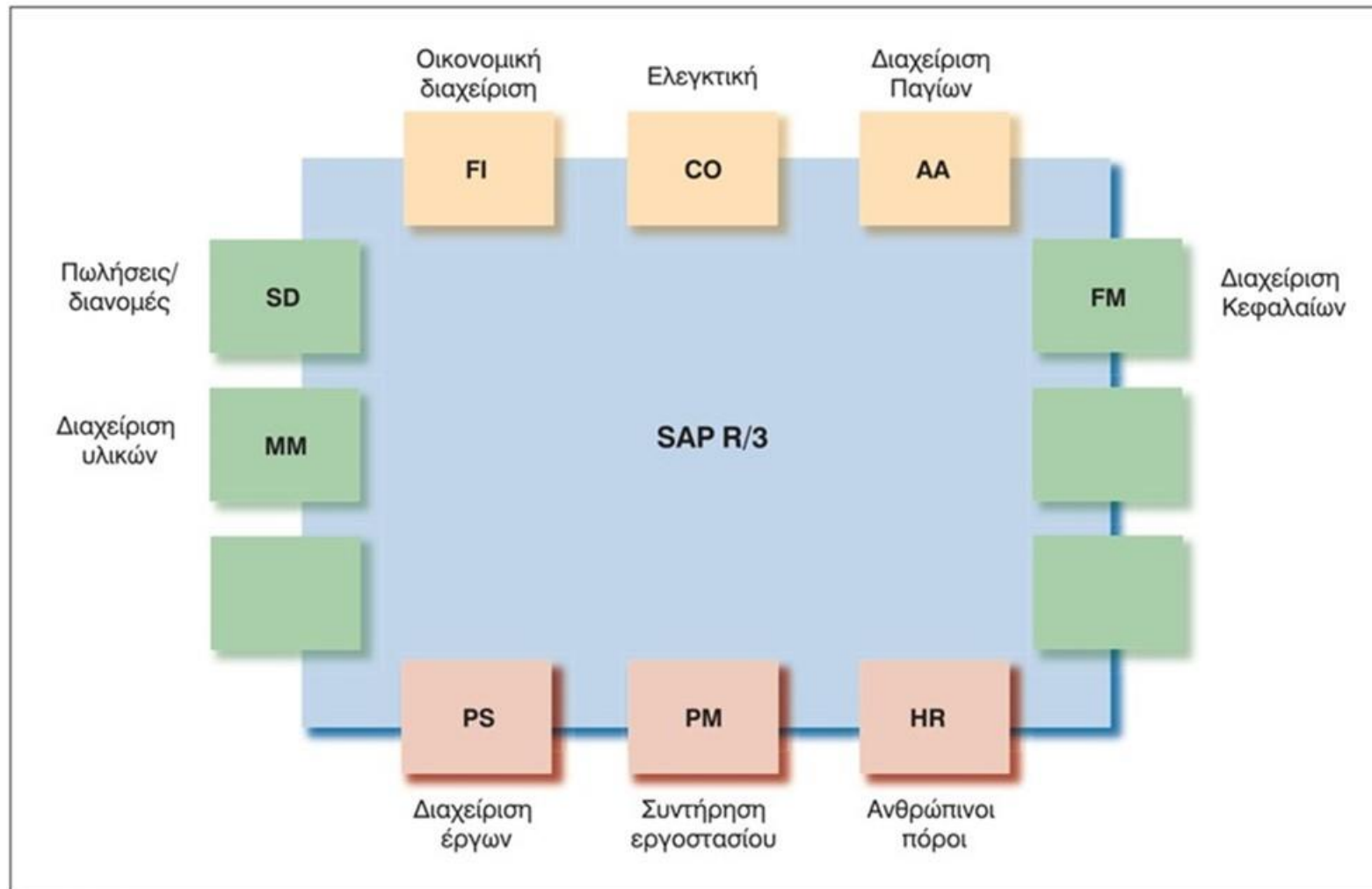
- Σε αυτή την αρχιτεκτονική υπάρχουν δύο βασικές οντότητες:
  - Του πελάτη (client), δηλαδή ένας σταθμός εργασίας που διαθέτει μια εφαρμογή που εκτελείται σε αυτόν και στην οποία έχει πρόσβαση ένας χρήστης.
  - Του εξυπηρετητή (server), δηλαδή μια συσκευή εξυπηρετητή (server) όπου εκτελείται μια άλλη εφαρμογή που εξυπηρετεί την εφαρμογή πελάτη. Συχνά, οι δύο εφαρμογές (πελάτη και εξυπηρετητή) μπορεί να βρίσκονται στην ίδια ή σε διαφορετικές «μηχανές».

## Αρχιτεκτονική thin-client

- Επί της ουσίας, οι εφαρμογές thin (thin clients) χρησιμοποιούν την εφαρμογή φυλλομετρητή (browser) για να εκτελεστούν, αντικαθιστώντας την προσέγγιση των τυφλών τερματικών που παρουσιάστηκε ως λύση για την αντικατάσταση των εφαρμογών πελάτη/εξυπηρετητή



# Αρθρωτή αρχιτεκτονική

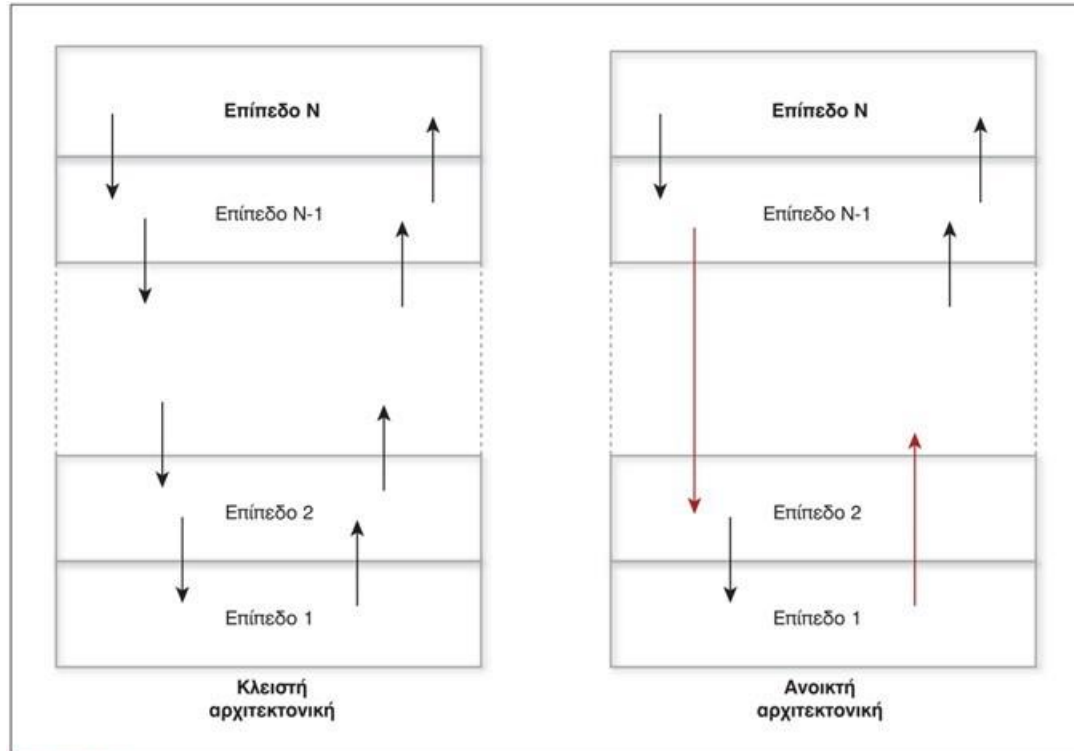


**ΕΙΚΟΝΑ 2.7**

Το σύστημα SAP R/3 ως παράδειγμα ενός αρθρωτού συστήματος.



# Πολυεπίπεδη αρχιτεκτονική



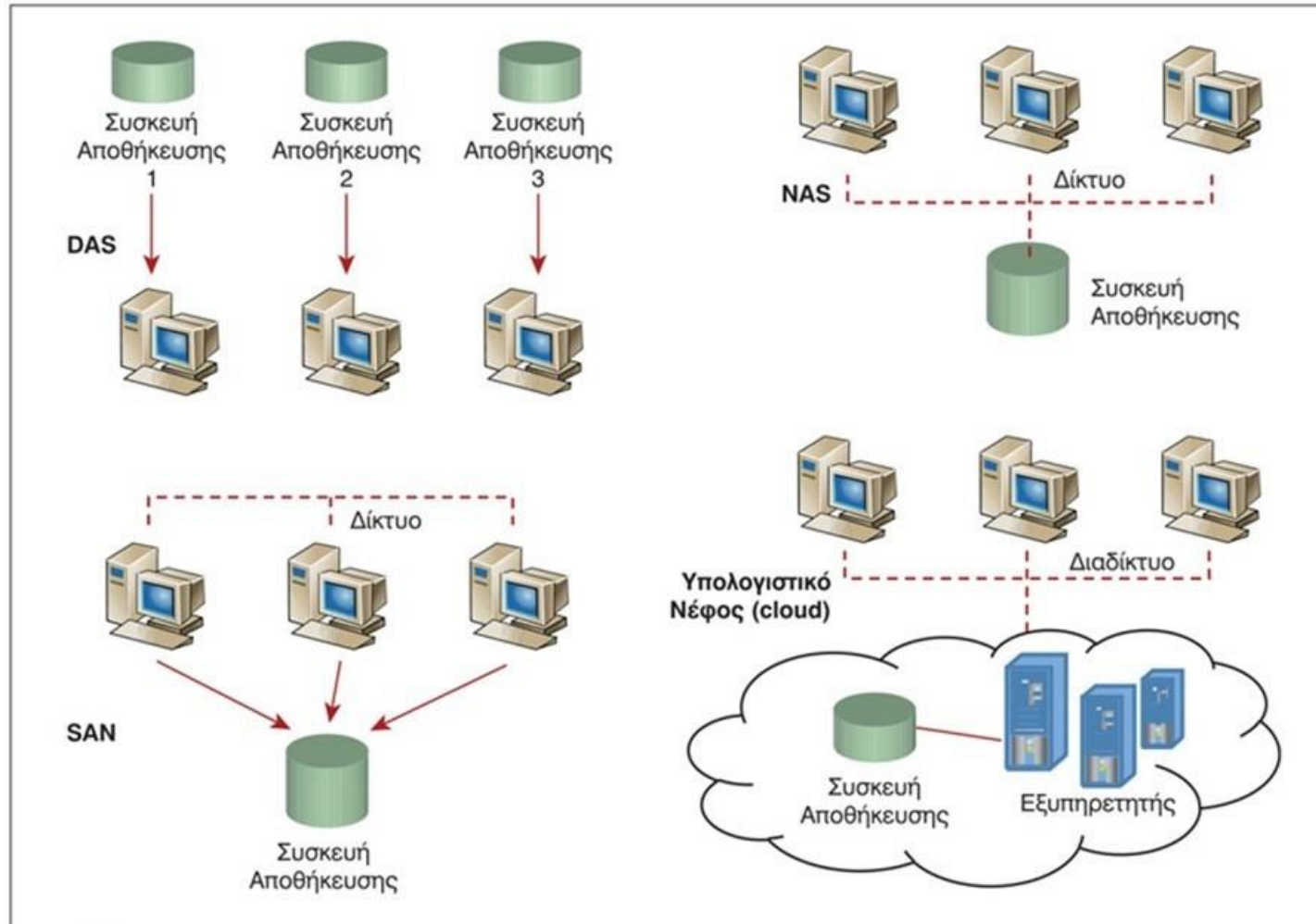
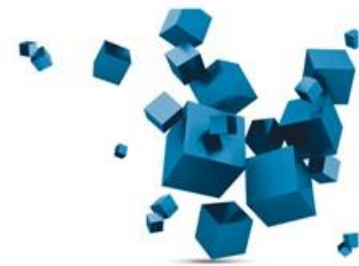
ΕΙΚΟΝΑ 2.8

Παράδειγμα ανοικτής και κλειστής αρχιτεκτονικής.

- Επίπεδο παρουσίασης/χρηστών (presentation tier)
- Επιχειρησιακό Επίπεδο (business tier)
- Επίπεδο διασύνδεσης (interconnection tier)
- Επίπεδο δεδομένων (data tier)



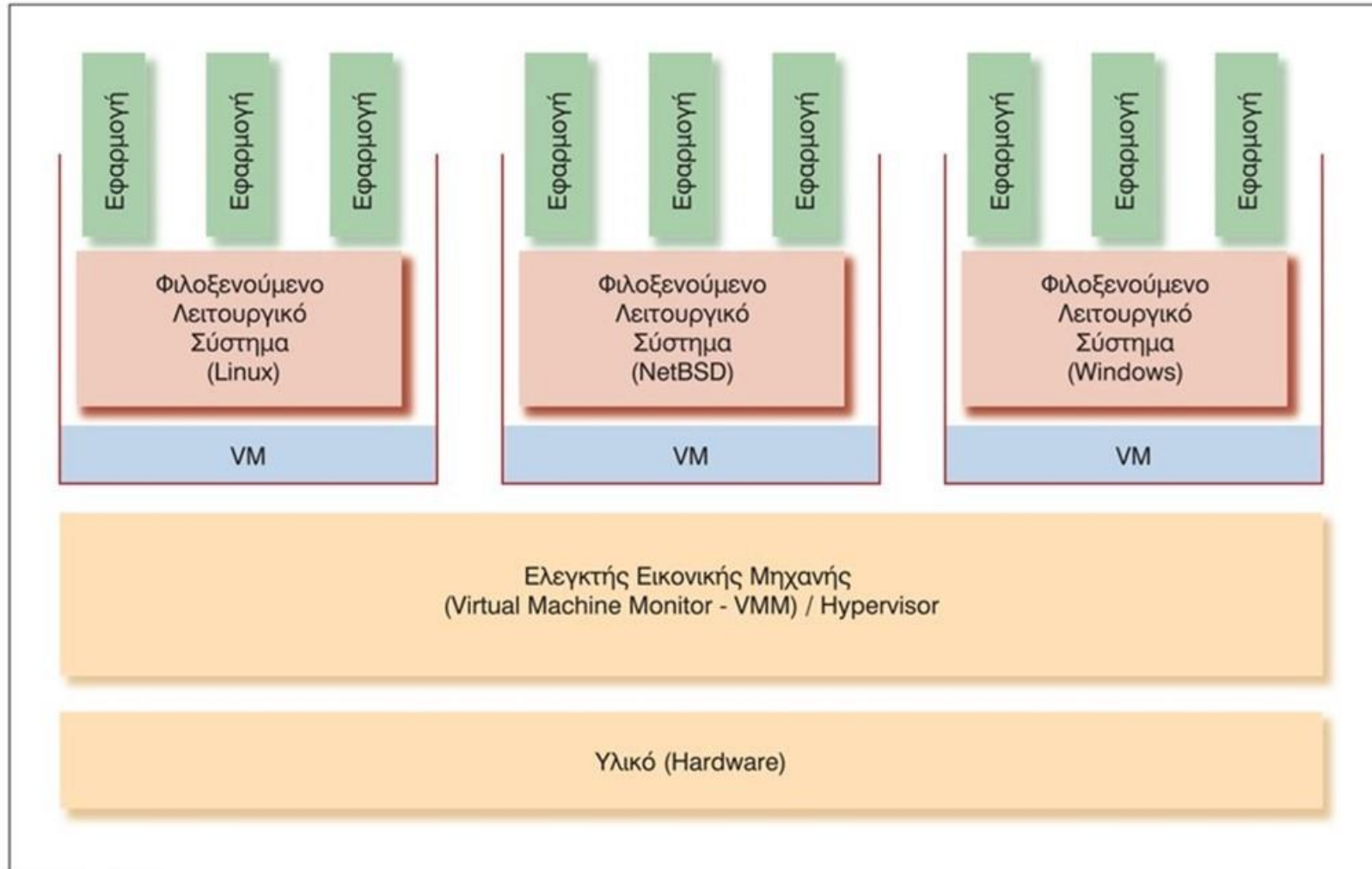
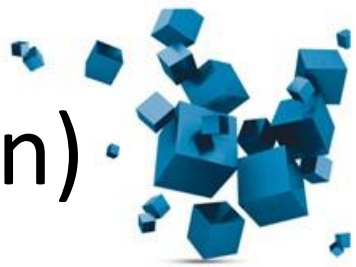
# Αρχιτεκτονικές αποθήκευσης δεδομένων



**EIKONA 2.9**

Αρχιτεκτονικές αποθήκευσης δεδομένων.

# Αρχιτεκτονική με εικονικές μηχανές (virtualization)

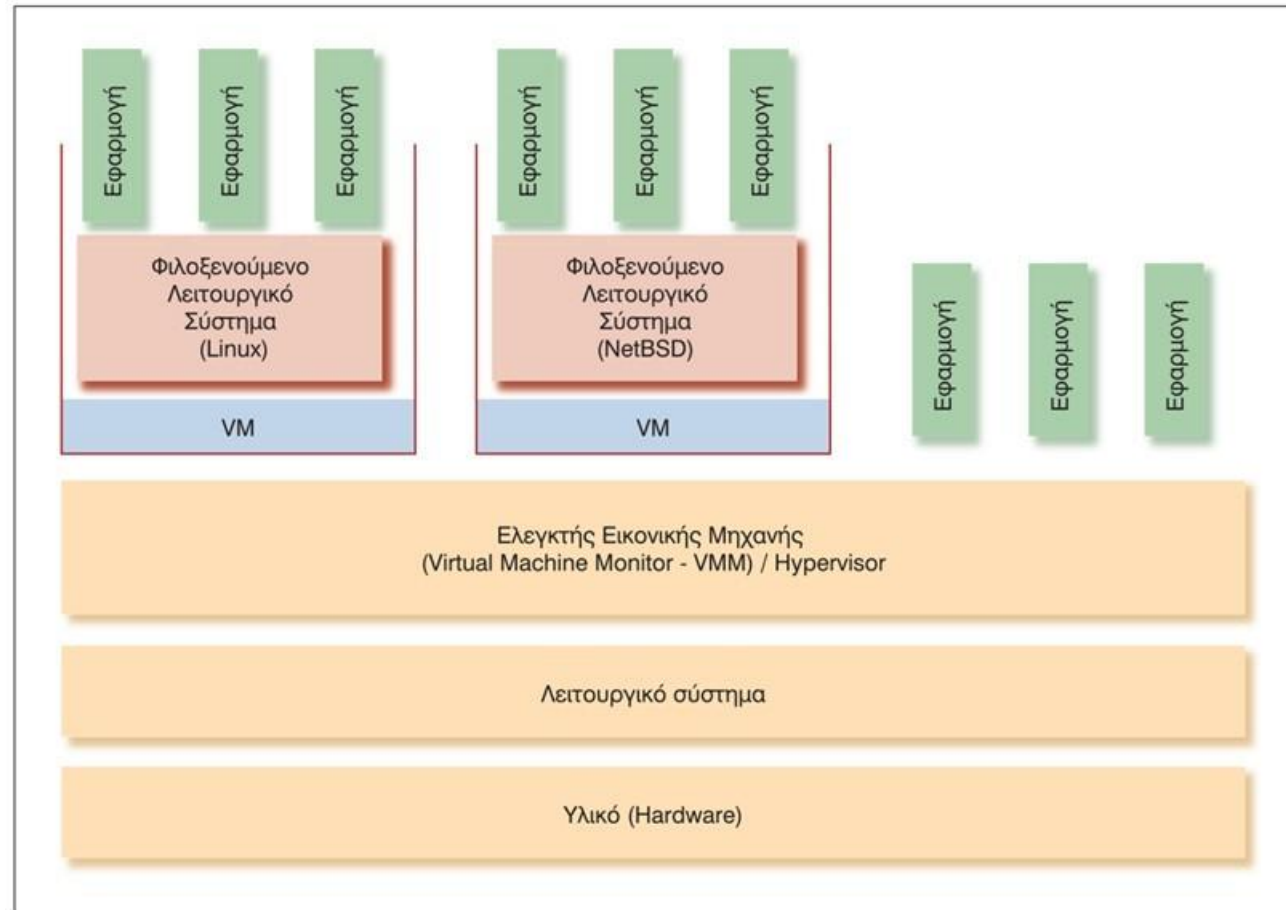


**ΕΙΚΟΝΑ 2.10**

Αρχιτεκτονική εικονικών μηχανών με hypervisor τύπου 1.

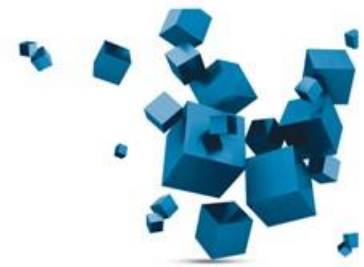


# Αρχιτεκτονική εικονικών μηχανών με hypervisor τύπου 2



**ΕΙΚΟΝΑ 2.11**

Αρχιτεκτονική εικονικών μηχανών με hypervisor τύπου 2.

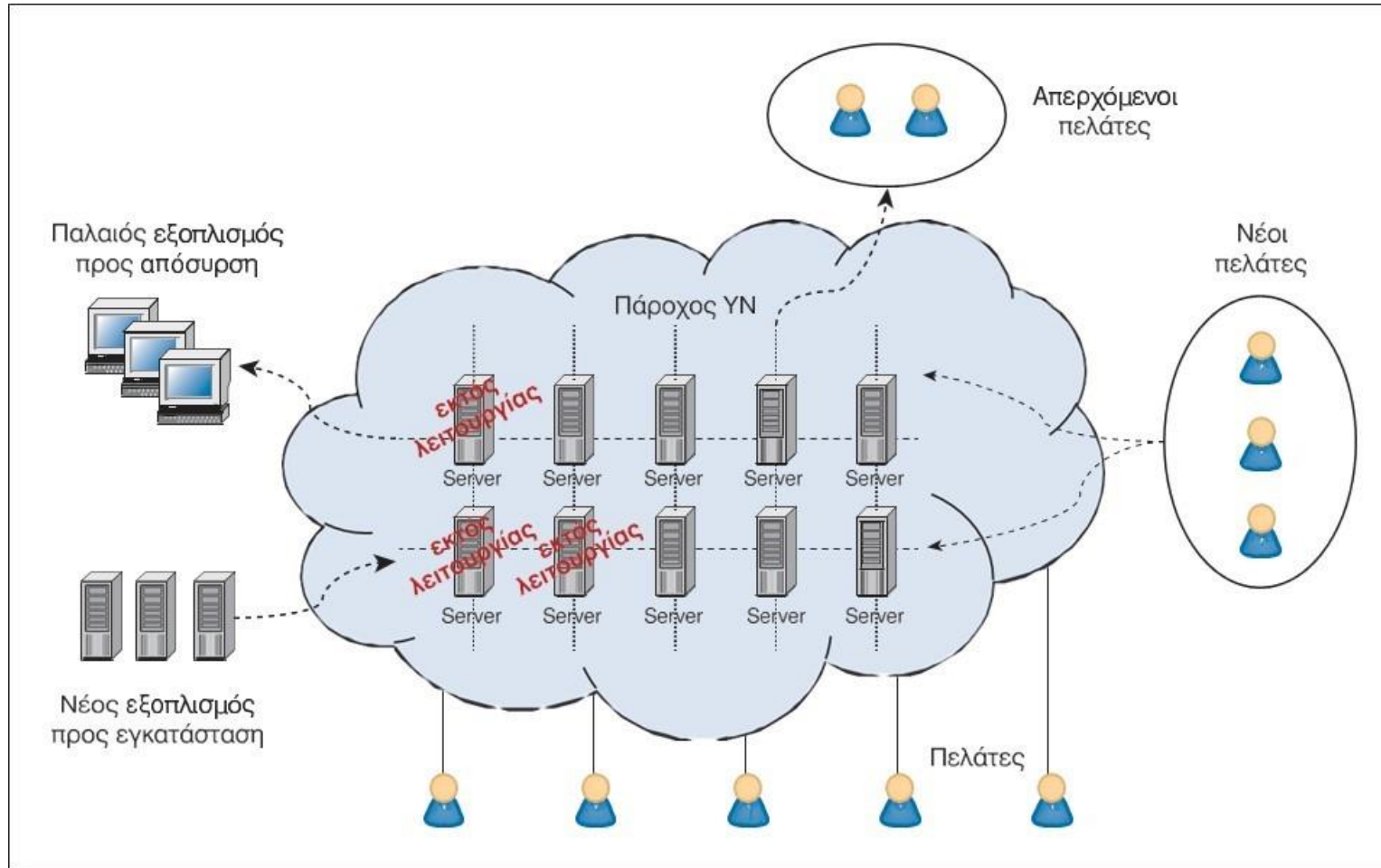


# Πλεονεκτήματα εικονοποίησης





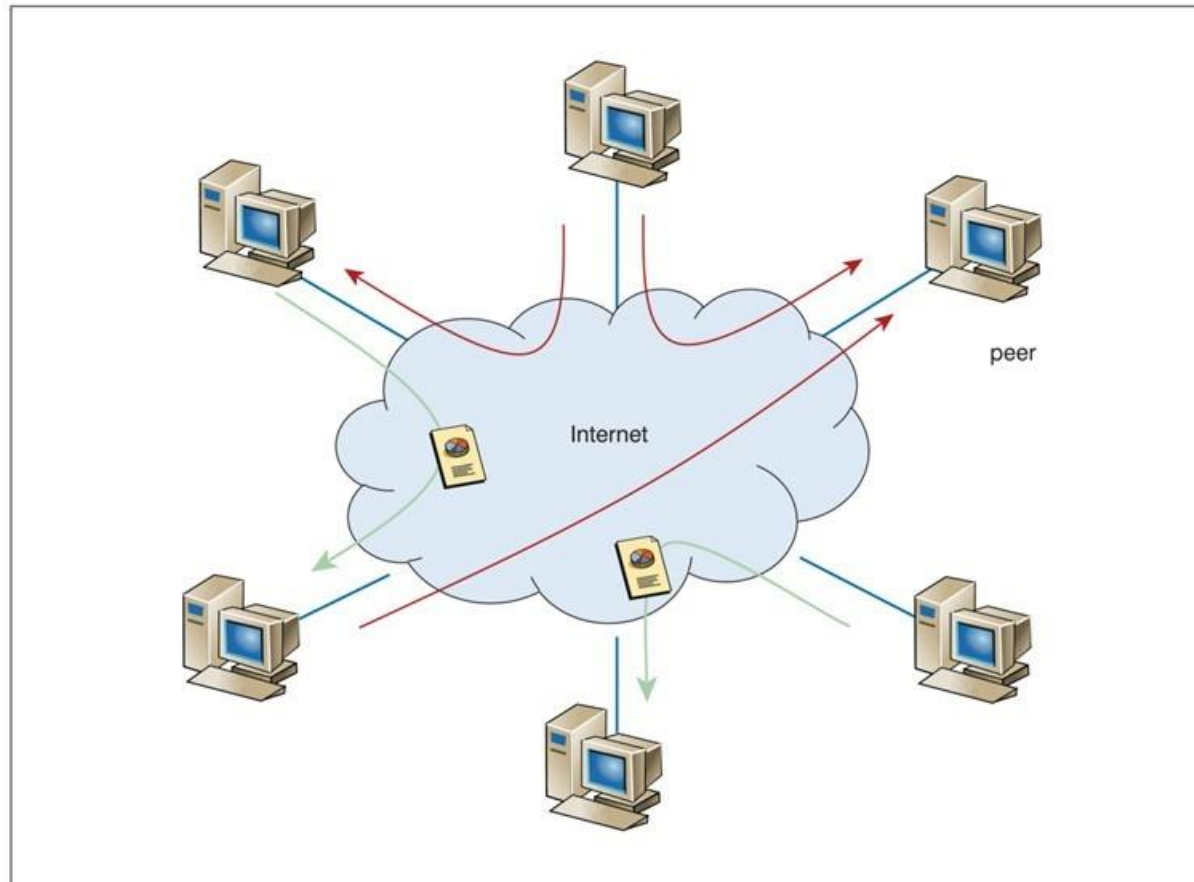
# Αρχιτεκτονική υπολογιστικού νέφους



Θα καλυφθεί στο κεφάλαιο 8

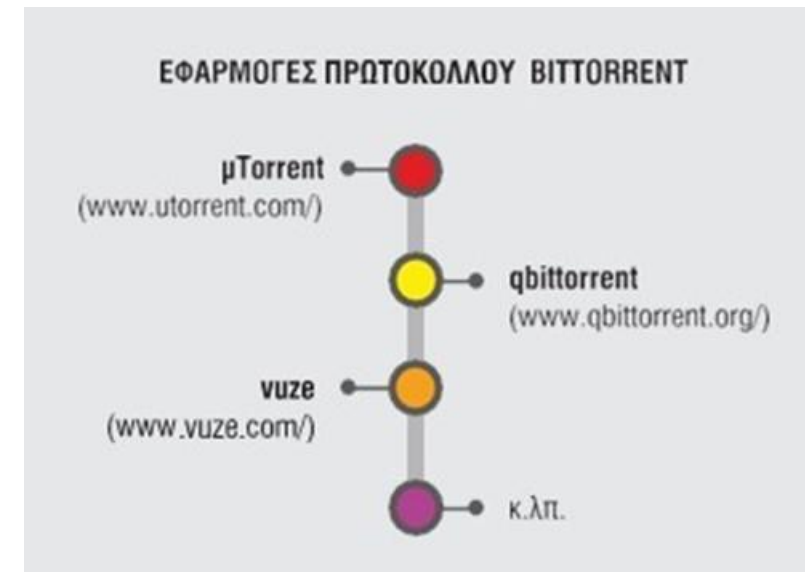


# Αρχιτεκτονική peer-to-peer



**ΕΙΚΟΝΑ 2.12**  
Αρχιτεκτονική peer-to-peer.

## Εφαρμογές bittorrent





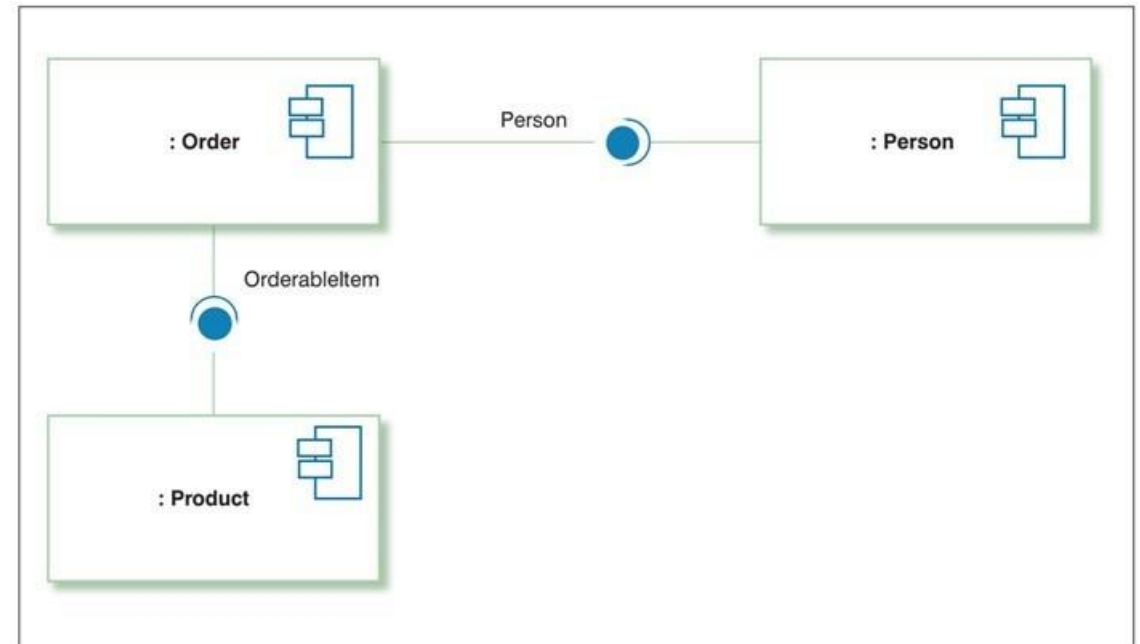
# Αρχιτεκτονικές εφαρμογών

- Η αρχιτεκτονική εφαρμογών είναι
  - Η δομή ή οι δομές μιας εφαρμογής που περιλαμβάνει τις προγραμματιστικές οντότητες (software components) που τη συνθέτουν,
  - Οι εξωτερικές ιδιότητες των οντοτήτων αυτών, καθώς και τη μεταξύ τους διασύνδεση.
- Η αρχιτεκτονική εφαρμογών περιλαμβάνει τις αποφάσεις για τη δομή ή τις δομές μιας εφαρμογής, καθώς και τις αλληλεπιδράσεις μεταξύ των δομών αυτών.
- Η αρχιτεκτονική αυτή καθορίζει την ανάπτυξη, την υποστήριξη και τη συντήρηση της εφαρμογής



# Αρχιτεκτονική συστατικών

- Ένα συστατικό :
  - είναι μια αυτόνομη μονάδα λογισμικού,
  - έχει διεπαφές (interfaces), δηλαδή ένα σύνολο από λειτουργίες και υποχρεώσεις,
  - έχει εξαρτήσεις, δηλαδή σχέσεις με άλλα συστατικά, και
  - έχει ports, δηλαδή σημεία διεπαφής με άλλα συστατικά και το περιβάλλον.



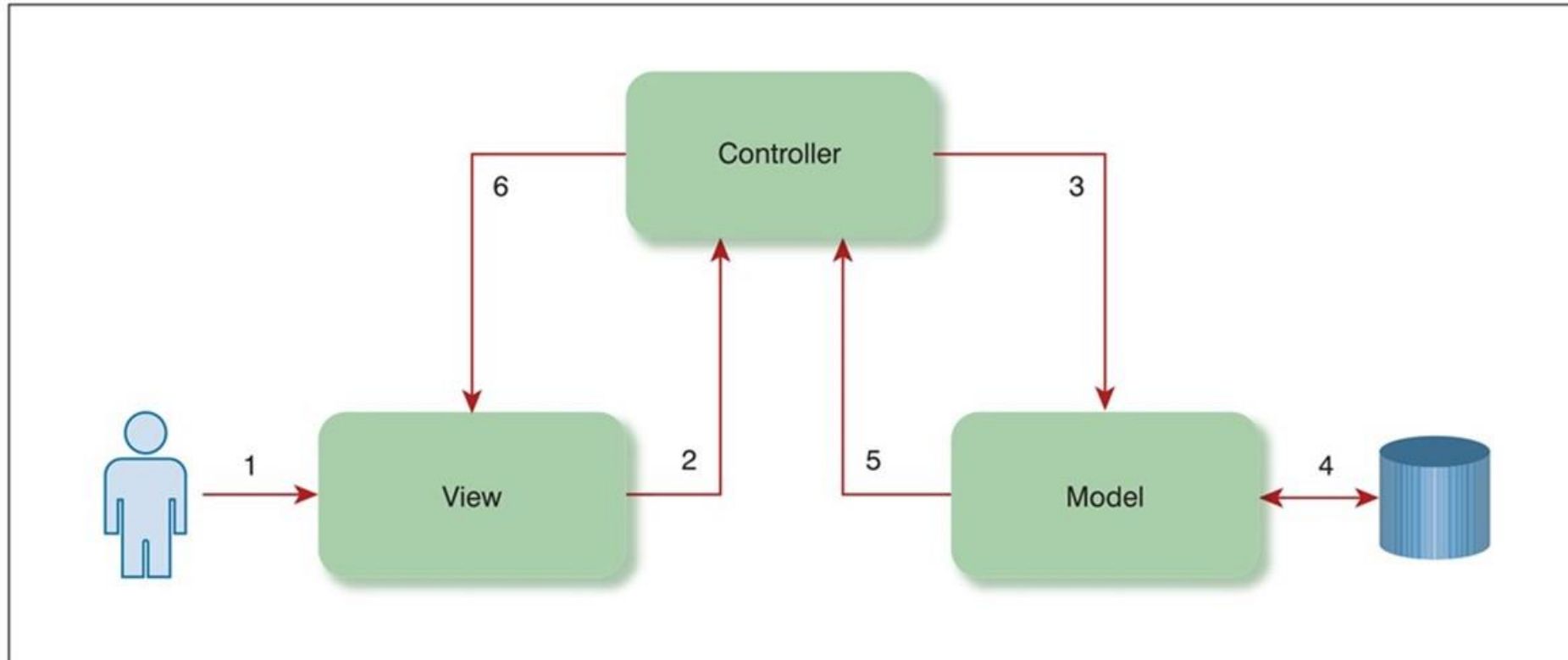
**EIKONA 2.13**

Διάγραμμα συστατικών της γλώσσας UML.





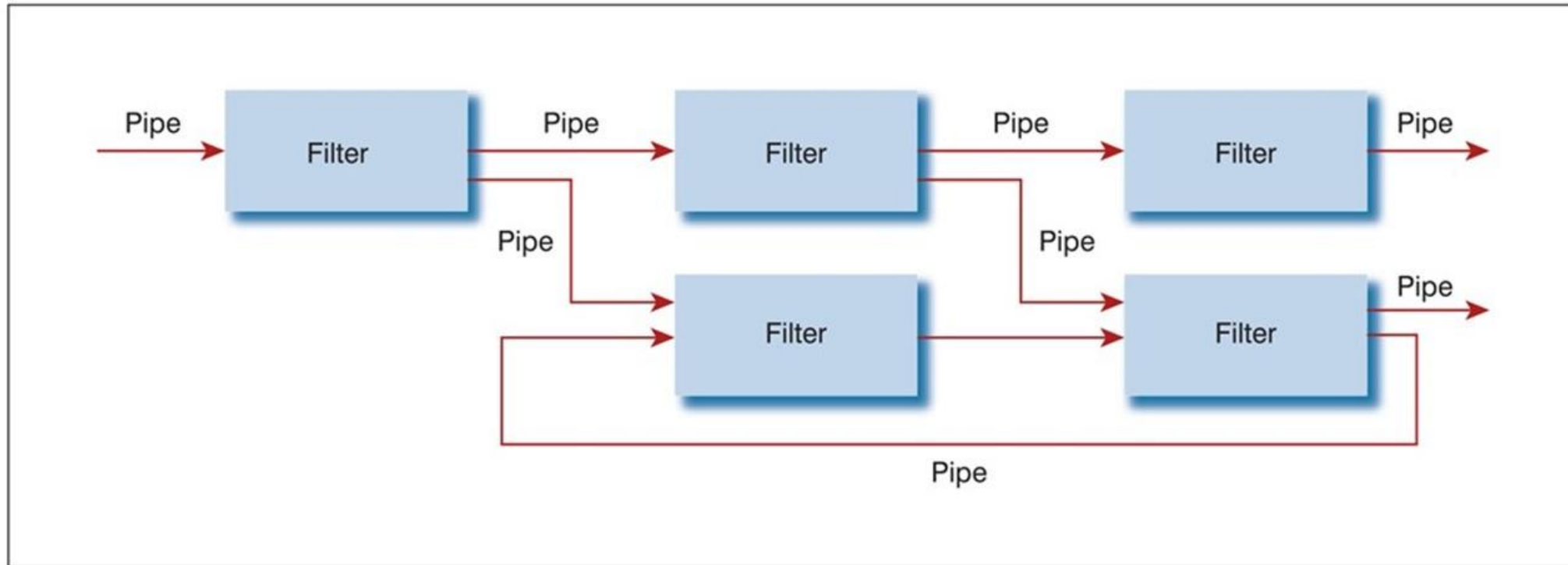
# Το αρχιτεκτονικό πρότυπο Model-View-Controller



**EIKONA 2.14**

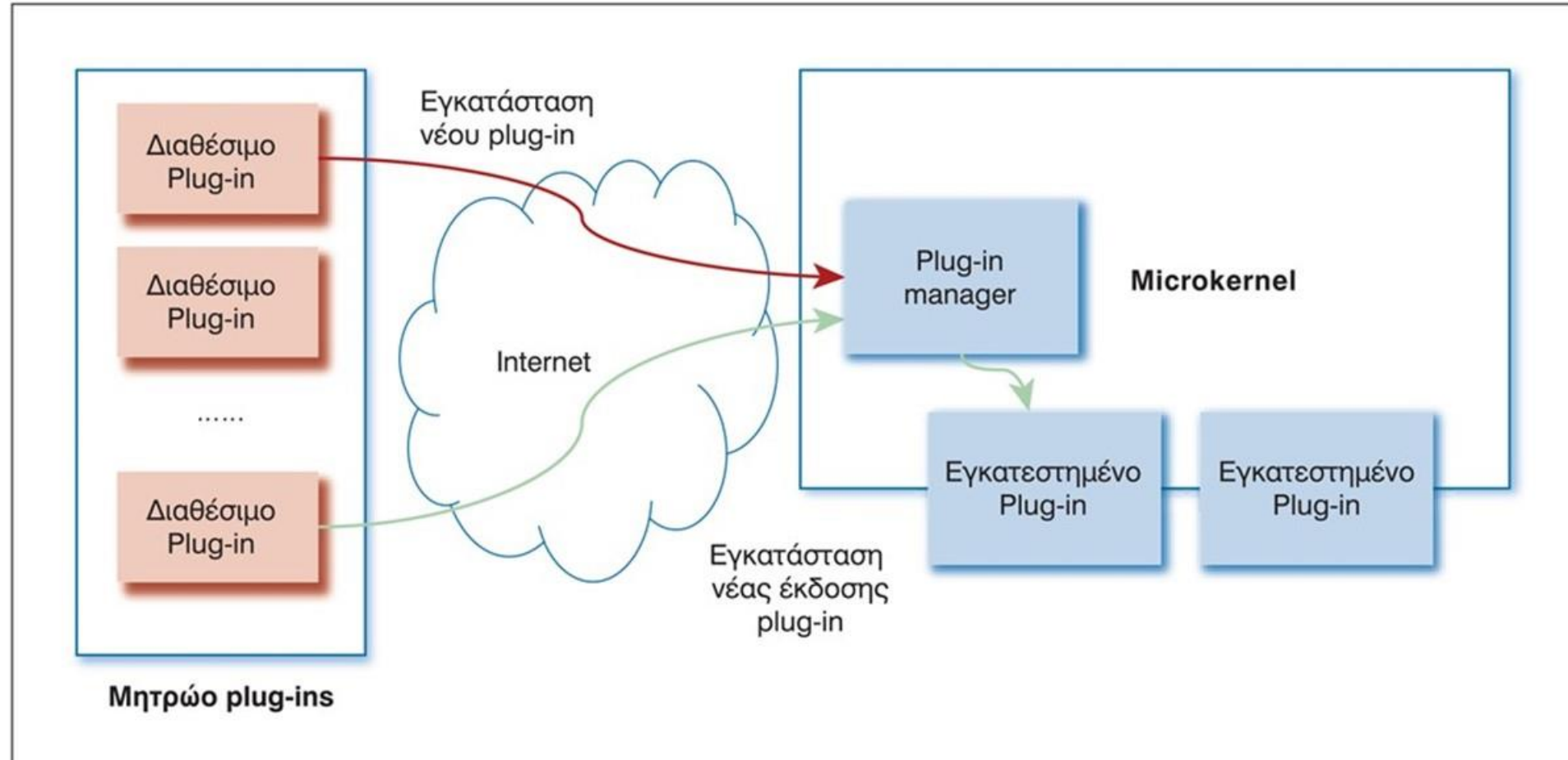
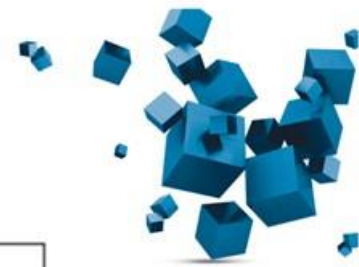
Αρχιτεκτονικό πρότυπο MVC.

# Το αρχιτεκτονικό πρότυπο αγωγοί και φίλτρα

**ΕΙΚΟΝΑ 2.15**

Αρχιτεκτονικό πρότυπο pipe-filter.

# Αρχιτεκτονικό πρότυπο microkernel ή plug-ins

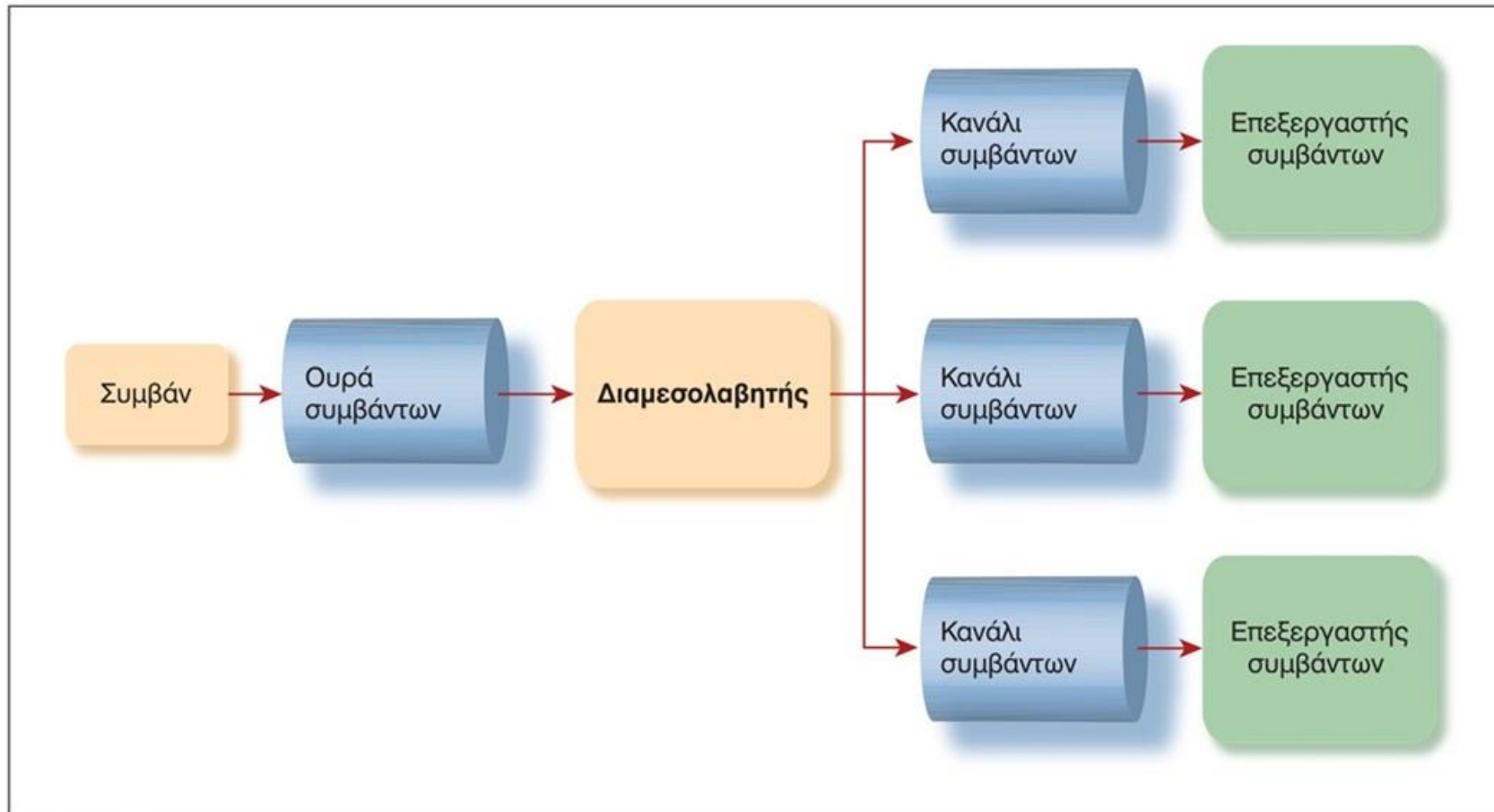


**ΕΙΚΟΝΑ 2.16**

Αρχιτεκτονικό πρότυπο microkernel ή plug-ins.



# Αρχιτεκτονική βασισμένη σε συμβάντα (event-driven) – πρότυπο διαμεσολαβητή

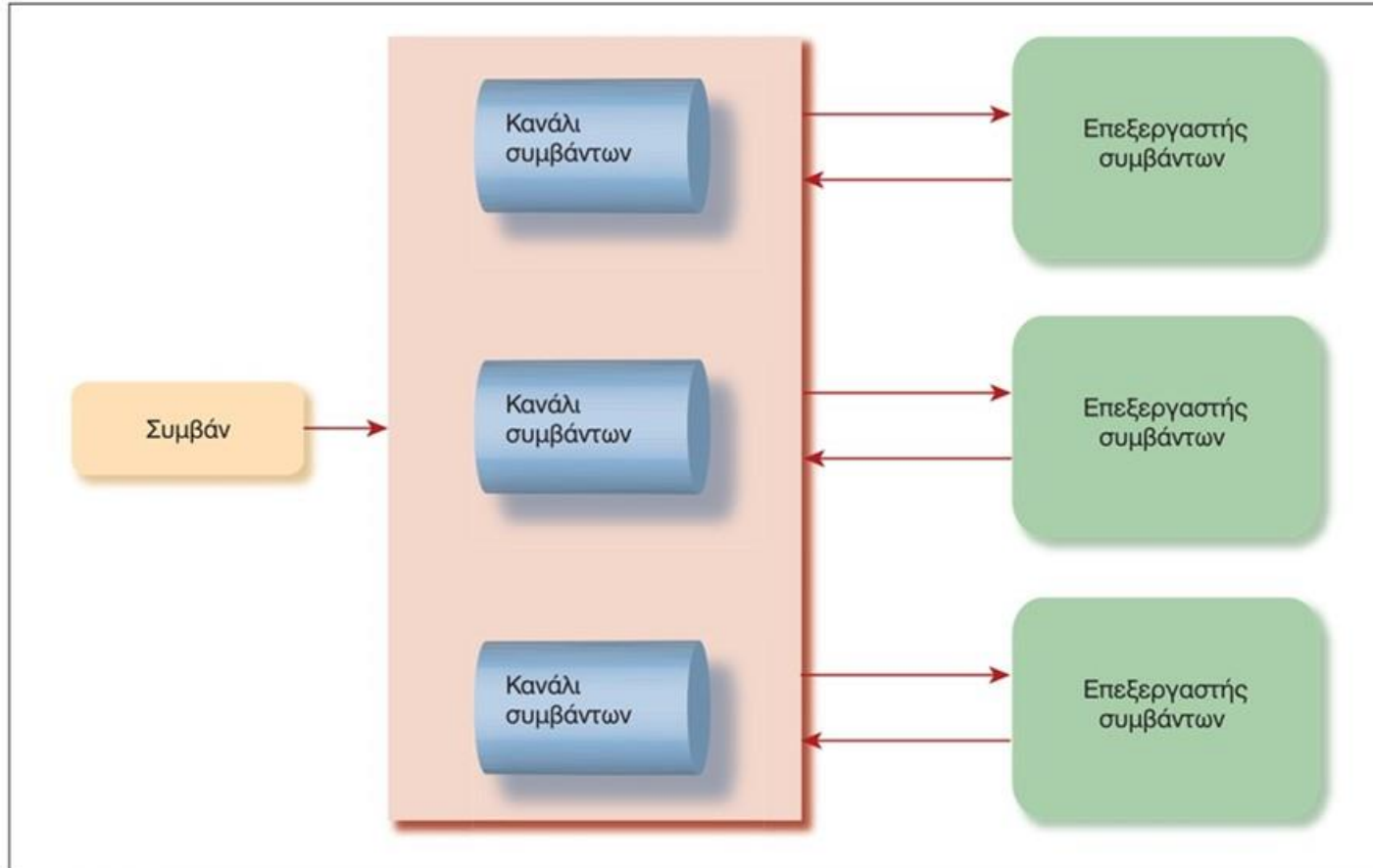


**EIKONA 2.17**

Το αρχιτεκτονικό πρότυπο του διαμεσολαβητή (mediator).



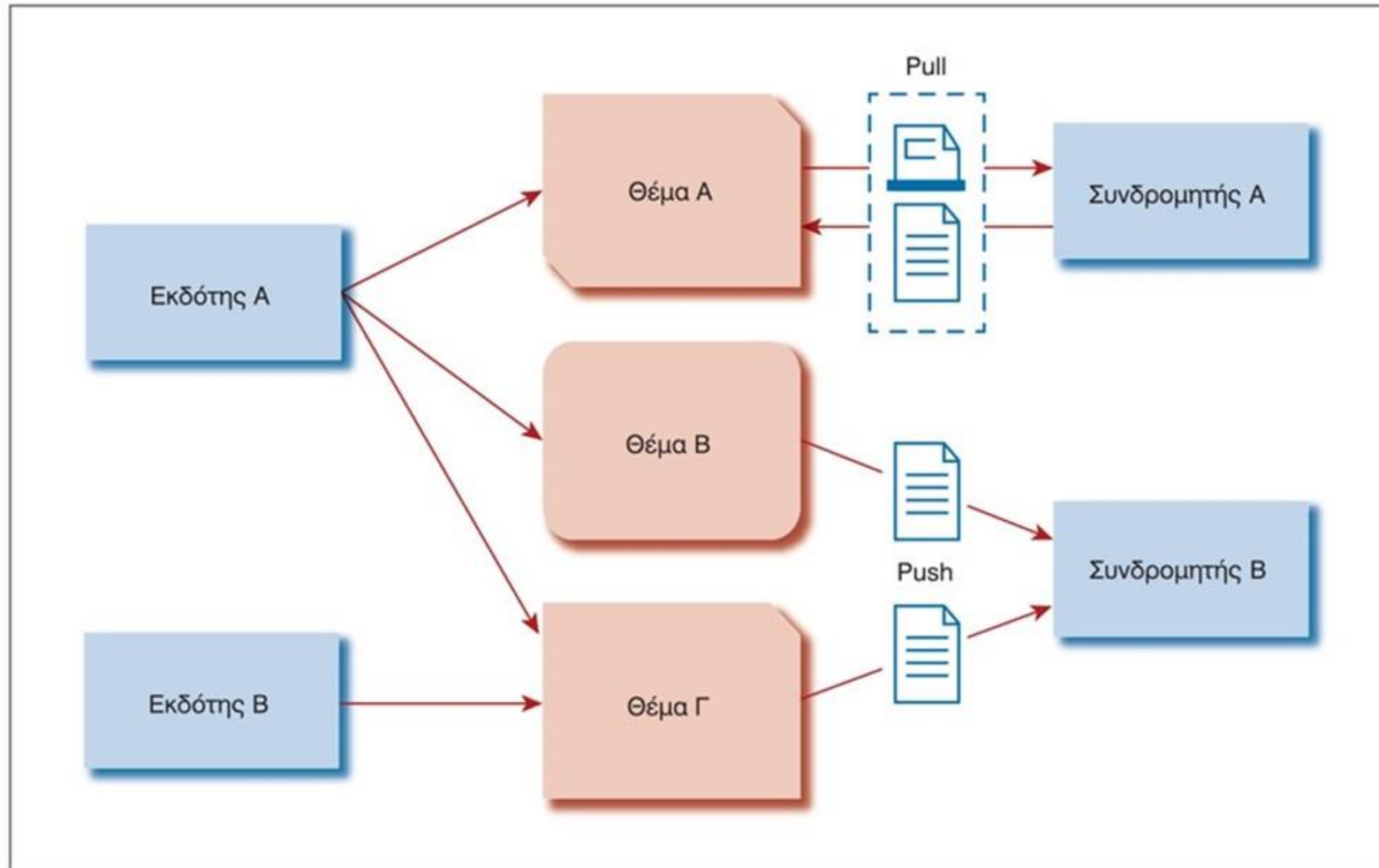
# Αρχιτεκτονική βασισμένη σε συμβάντα (event-driven) – πρότυπο μεσίτη



**ΕΙΚΟΝΑ 2.18**

Το αρχιτεκτονικό πρότυπο του μεσίτη (broker).

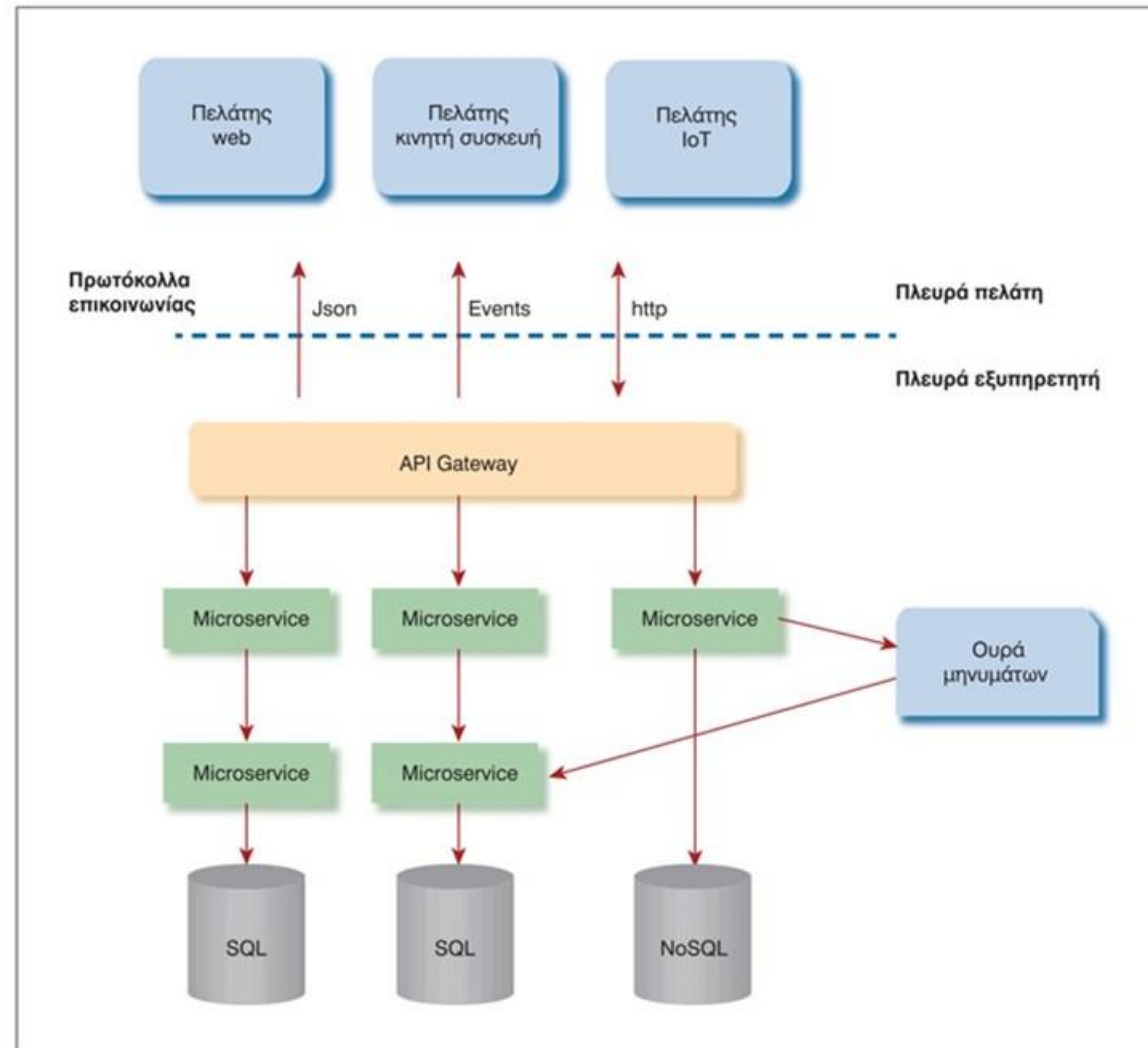
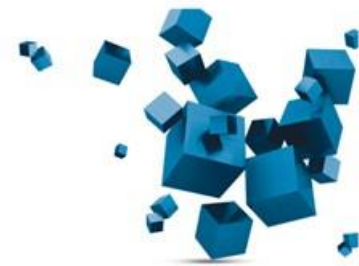
# Το αρχιτεκτονικό πρότυπο publisher-subscriber



**ΕΙΚΟΝΑ 2.19**

Το αρχιτεκτονικό πρότυπο publisher-subscriber.

# Αρχιτεκτονικό πρότυπο μικρο-υπηρεσιών

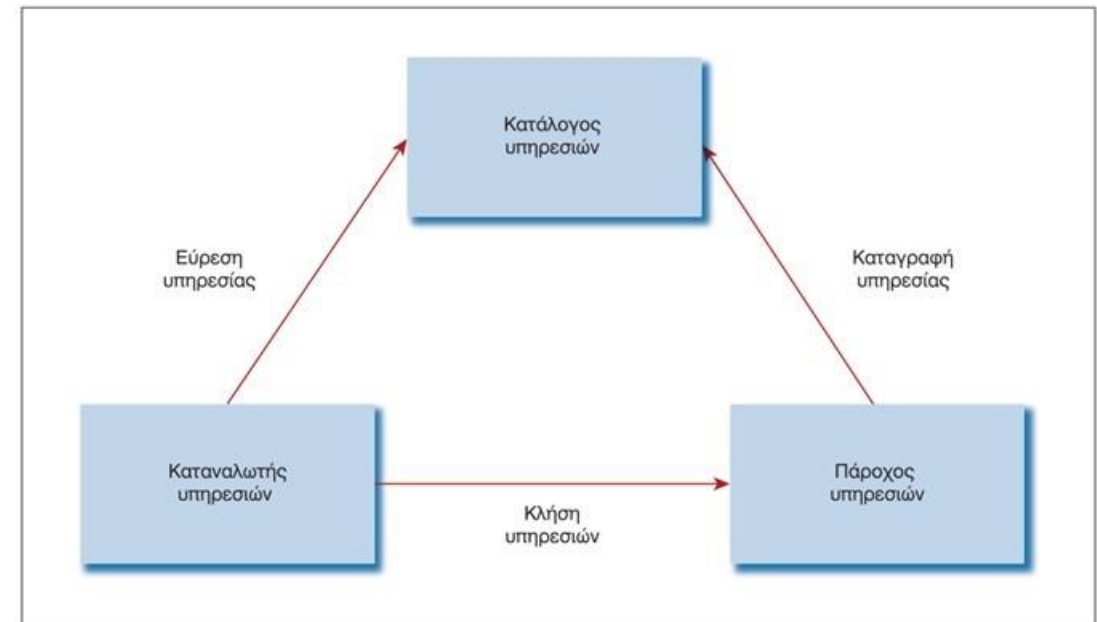


**ΕΙΚΟΝΑ 2.20**  
Αρχιτεκτονική μικροϋπηρεσιών.



# Αρχιτεκτονική προσανατολισμένη σε υπηρεσίες

- Οι βασικές αρχές της αρχιτεκτονικής SOA είναι:
  - Οι υπηρεσίες είναι αυτόνομες. Κάθε υπηρεσία αναπτύσσεται, εγκαθίσταται και συντηρείται ανεξάρτητα.
  - Οι υπηρεσίες είναι κατακεκομμένες. Μια υπηρεσία μπορεί να βρίσκεται οπουδήποτε στο διαδίκτυο.
  - Οι υπηρεσίες είναι χαλαρά συνδεδεμένες (loosely coupled). Κάθε υπηρεσία είναι ανεξάρτητη από τις άλλες και μπορεί να αντικατασταθεί χωρίς να διαταραχθούν οι εφαρμογές που την χρησιμοποιούν, με την προϋπόθεση ότι η διεπαφή (service interface) που δίνει είναι συμβατή.
  - Οι υπηρεσίες παρέχουν ένα σαφώς ορισμένο συμβόλαιο διεπαφών (interface contract).



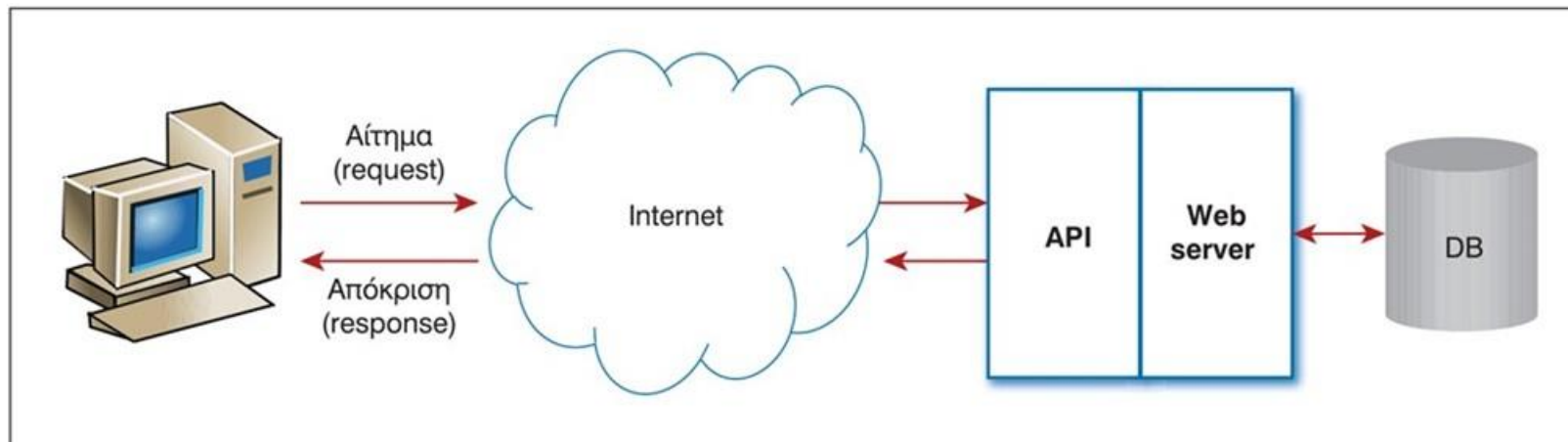
**ΕΙΚΟΝΑ 2.21**  
Η φιλοσοφία της αρχιτεκτονικής SOA.





# Η αρχιτεκτονική REST

- Ομοιόμορφη επαφή (uniform interface).
- Διαχωρισμός πελάτη και εξυπηρετητή (client - server separation).
- Χωρίς κατάσταση (stateless).
- Σύστημα δομημένο σε επίπεδα (layered system).
- Τα δεδομένα να μπορούν να αποθηκευτούν στη μνήμη Cache (cacheable).
- Κώδικας μετά από απαίτηση (code-on-demand).



**ΕΙΚΟΝΑ 2.22**  
Η αρχιτεκτονική με API.



# To Uniform Resource Identifier (URI)

Ένα URI είναι της μορφής

*scheme://host:port/path?queryString#fragment*

ΠΙΝΑΚΑΣ 2.1 Η δομή του URI	
Τμήμα URI	Επεξήγηση
Scheme:	Πρωτόκολλο που χρησιμοποιείται για την επικοινωνία. Μπορεί να είναι http, https, ftp, telnet, mailto.
Host:	Όνομα DNS ή διεύθυνση IP.
Port:	Θύρα μέσω της οποίας γίνεται η επικοινωνία. Στα πρωτόκολλα TCP και UDP, τα οποία αποτελούν βασικά δικτυακά πρωτόκολλα του διαδικτύου, η θύρα είναι το σημείο που αποστέλλονται τα δεδομένα και είναι διαφορετική για κάθε δικτυακό πρωτόκολλο. Για παράδειγμα, στη θύρα 80 στέλνουμε δεδομένα στην περίπτωση του πρωτοκόλλου HTTP ή στη θύρα 20 για το πρωτόκολλο FTP. Για την πλήρη λίστα των θυρών μπορείτε να ανατρέξετε στο <a href="https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers">https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers</a> .
Path:	Η διαδρομή από τον κεντρικό κατάλογο στη θέση που βρίσκεται ο πόρος.
queryString:	Οι παράμετροι του ερωτήματος.
Fragment:	Ακολουθεί τον χαρακτήρα # και χρησιμοποιείται για να καθορίσει το ακριβές σημείο του πόρου αναφοράς του ερωτήματος.



# Παραδείγματα εντολών REST API

ΠΙΝΑΚΑΣ 2.2 Παραδείγματα εντολών REST API

Εντολή	URL	Περιγραφή
POST	<a href="http://example.com/EmployeeService/employee/1">http://example.com/EmployeeService/employee/1</a>	Ο εξυπηρετητής θα δημιουργήσει έναν εργαζόμενο.
GET	<a href="http://example.com/EmployeeService/employee">http://example.com/EmployeeService/employee</a>	Ο εξυπηρετητής θα μας επιστρέψει τη λίστα όλων των εργαζομένων.
GET	<a href="http://example.com/EmployeeService/employee/1">http://example.com/EmployeeService/employee/1</a>	Ο εξυπηρετητής θα μας επιστρέψει τον εργαζόμενο με αριθμό 1 της λίστας.
PUT	<a href="http://example.com/EmployeeService/employee/2">http://example.com/EmployeeService/employee/2</a>	Ο εξυπηρετητής θα ενημερώσει κάποιο χαρακτηριστικό του εργαζομένου 2.
DELETE	<a href="http://example.com/EmployeeService/employee/">http://example.com/EmployeeService/employee/</a>	Ο εξυπηρετητής θα διαγράψει όλους τους εργαζόμενους.



# Ανταλλαγή δεδομένων μεταξύ πελάτη εξυπηρετητή

- Η ανταλλαγή δεδομένων γίνεται με χρήση γλωσσών δεδομένων
  - Π.χ. XML, JSON
- Παράδειγμα json

```
{  
  "employee_first_name": "Panos",  
  "employee_family_name": "Fitsilis",  
  "address": "Koumoundourou 6",  
  "zip": "12300",  
  "city": "Larissa",  
  "email": pfitsilis@example.com,  
}
```



# Η πληροφοριακή υποδομή της επιχείρησης

- Οι υποστηρικτικές υπηρεσίες που παρέχονται μπορούν να ταξινομηθούν ενδεικτικά στις παρακάτω γενικές κατηγορίες :
  - υποσύστημα διαχείρισης ταυτότητας χρηστών,
  - υποσύστημα διαχείρισης δικτυακής πύλης,
  - υποσύστημα διαχείρισης ροής εργασιών,
  - υποσύστημα διαχείρισης εγγράφων, και
  - υποσυστήματα ανάπτυξης/προσαρμογής λογισμικού.

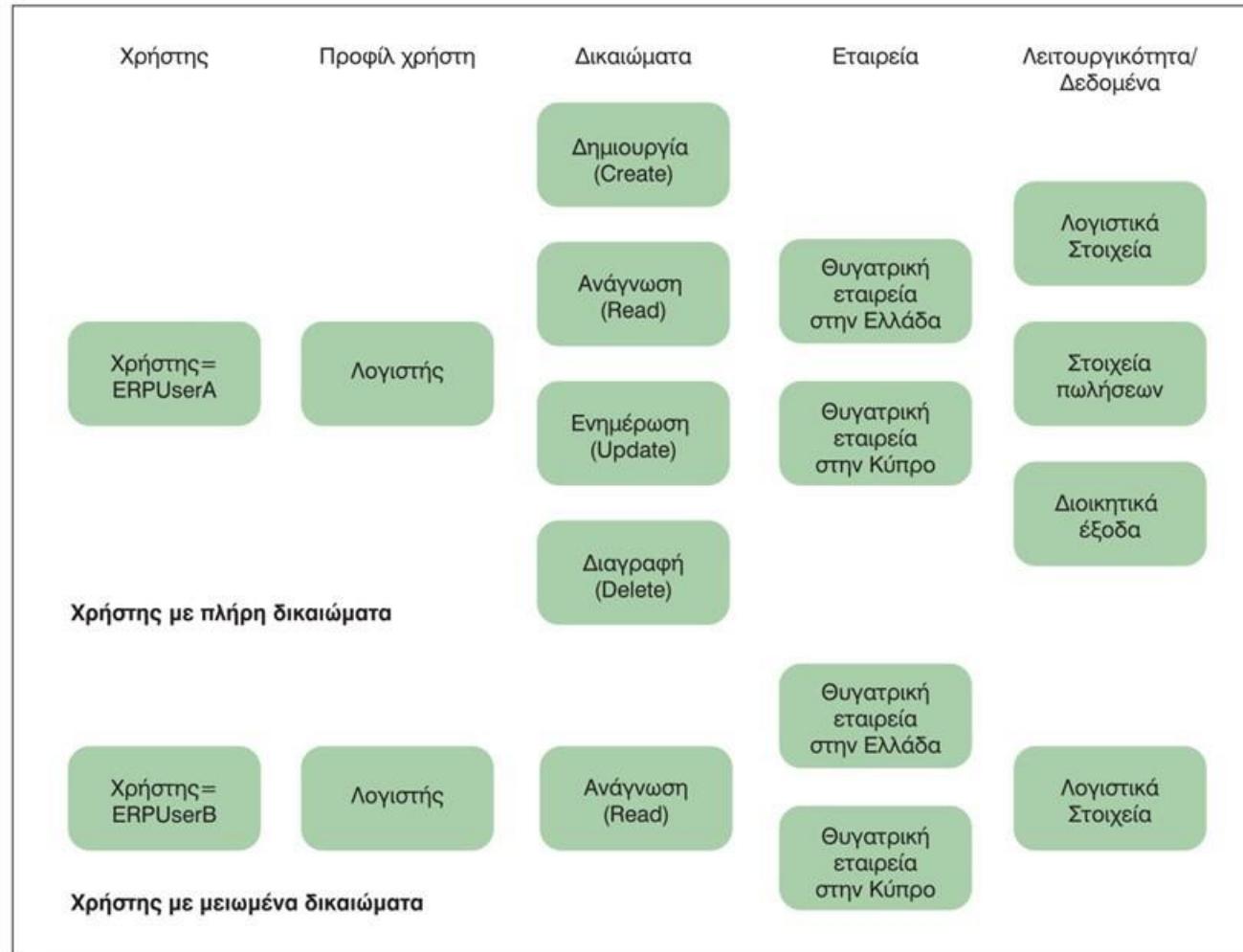


# Η οργάνωση των δικαιωμάτων των χρηστών

- Πρόσβαση σε λειτουργίες του συστήματος (function access control):
  - Αφορά τον ορισμό των λειτουργιών που θα είναι διαθέσιμες σε κάθε χρήστη. Για παράδειγμα, ποια μενού θα είναι διαθέσιμα σε έναν πωλητή και ποια θα είναι διαθέσιμα σε έναν διευθυντή τμήματος πωλήσεων;
- Πρόσβαση σε δεδομένα του συστήματος (data access control):
  - Αφορά τον ορισμό των οντοτήτων που θα είναι προσπελάσιμες από κάθε χρήστη. Για παράδειγμα, ποιοι πελάτες είναι ορατοί σε έναν πωλητή και ποιες σε ένα λογιστή της ίδιας εταιρείας ή ποιοι πελάτες θα συμπεριλαμβάνονται στην αναφορά που παράγεται για έναν διευθυντή τμήματος πωλήσεων;
- Πρόσβαση με βάση το προφίλ του χρήστη (Role Based Access Control - RBAC):
  - Συνήθως, σε μια επιχείρηση ο ρόλος/προφίλ του χρήστη συνδέεται με τις εξουσιοδοτήσεις, τις ευθύνες, την πρόσβαση σε δεδομένα, αλλά και τις λειτουργίες του συστήματος.
  - Η διαχείριση των δικαιωμάτων χρηστών με βάση το προφίλ είναι ιδιαίτερα χρήσιμη, διότι αυτά ορίζονται για όλους τους χρήστες που έχουν το συγκεκριμένο προφίλ, γεγονός που απλοποιεί τη διαδικασία διαχείρισης. Επιπλέον, μας δίνει τη δυνατότητα να οργανώσουμε τους χρήστες του συστήματος σε ιεραρχίες κληρονομικότητας. Για παράδειγμα, ένας διευθυντής πωλήσεων έχει τα δικαιώματα του πωλητή και κάποια επιπλέον.



# Δικαιώματα χρηστών



**EIKONA 2.23**  
Δικαιώματα χρηστών.



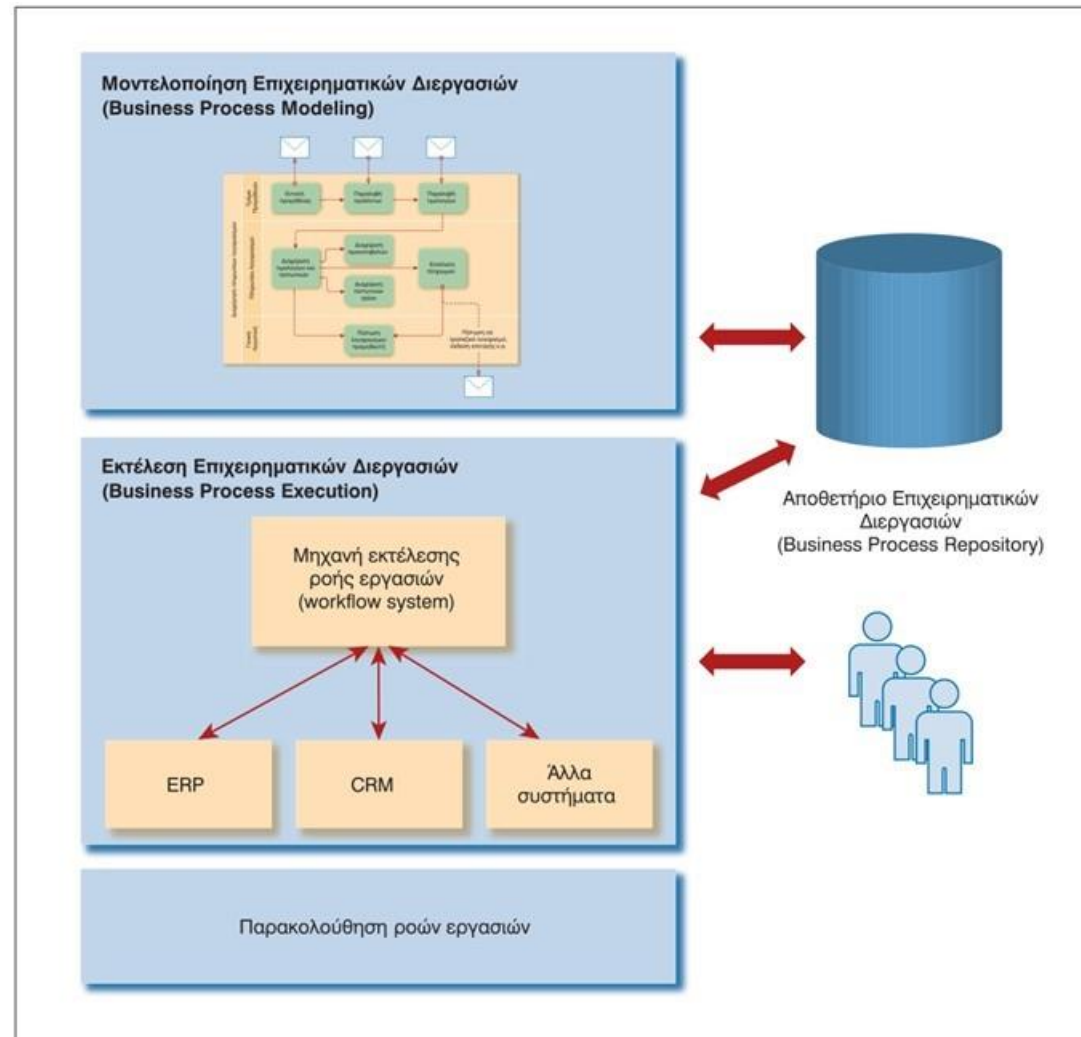
# Υποσύστημα ροής εργασιών

- Δυνατότητα σχεδιασμού σεναρίων ροής εργασιών και παρουσίασης/διεκπεραίωσης της τρέχουσας διαδικασίας με γραφικό (διαγραμματικό) τρόπο.
- Ολοκλήρωση με τα άλλα υποσυστήματα πληροφοριακής υποδομής.
- Δυνατότητα ανάθεσης ενέργειας με ή χωρίς την απαίτηση αντίδρασης από τον χρήστη.
- Δρομολόγηση ενεργειών και εγγράφων σε μεμονωμένους χρήστες, αλλά και σε ομάδες, ορίζοντας προθεσμία διεκπεραίωσης.
- Εμφάνιση των εκκρεμούντων ανατεθεισών ενεργειών στον χρήστη με γραφικό τρόπο ως εγγραφές στο γραφικό περιβάλλον υποδοχής εργασιών του.
- Αυτόματη αποστολή email στον χρήστη που του έχει ανατεθεί μια εργασία.
- Δυνατότητα συσχέτισης μιας νέας εργασίας με ένα ή περισσότερα έγγραφα.
- Σχεδίαση και προγραμματισμό των διαδικασιών με έναν πλήρως γραφικό τρόπο (visual design).
- Δυνατότητα επανασχεδίασης μιας διαδικασίας.
- Ανάθεση εργασίας σε χρήστες με διάφορους τρόπους όπως π.χ. με χρήση του φόρτου εργασίας χρηστών
- Δυνατότητα αυτόματων εγκρίσεων ή αυτόματων απορρίψεων
- Δυνατότητα χρήσης χρονολογημένων γεγονότων (timed events)





# Υποσύστημα ροής εργασιών



**EIKONA 2.24**  
Υποσύστημα ροής εργασιών.

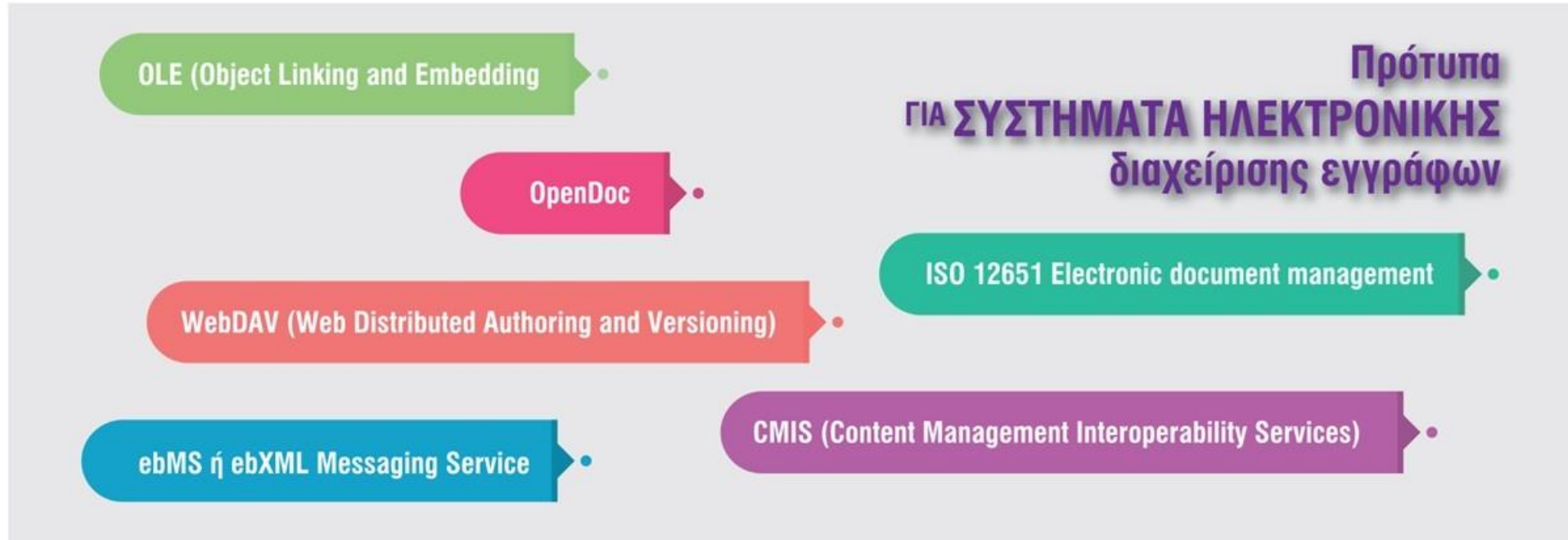


# Υποσύστημα διαχείρισης εγγράφων

- Ένα έγγραφο έχει έναν κύκλο ζωής, που περιλαμβάνει στάδια όπως:
  - Δημιουργία του εγγράφου ή σάρωση αυτού σε περίπτωση που δεν είναι σε ηλεκτρονική μορφή (capture).
  - Επεξεργασία και έγκριση (edit, review/approve).
  - Χαρακτηρισμός του εγγράφου (classification) ανάλογα με τη διαβάθμιση ασφαλείας (αν υπάρχει).
  - Αποθήκευση και ανάκτηση (store/retrieve).
  - Χρησιμοποίηση του εγγράφου (utilize).
  - Διάθεση – διατήρηση, εξαγωγή, καταστροφή του (disposition – retain, export, destroy).
  - Μεταδεδομένα (metadata) του εγγράφου.
  - Οργάνωση των εγγράφων με την εφαρμογή συγκεκριμένων προτύπων.
  - Παρακολούθηση και έλεγχος.

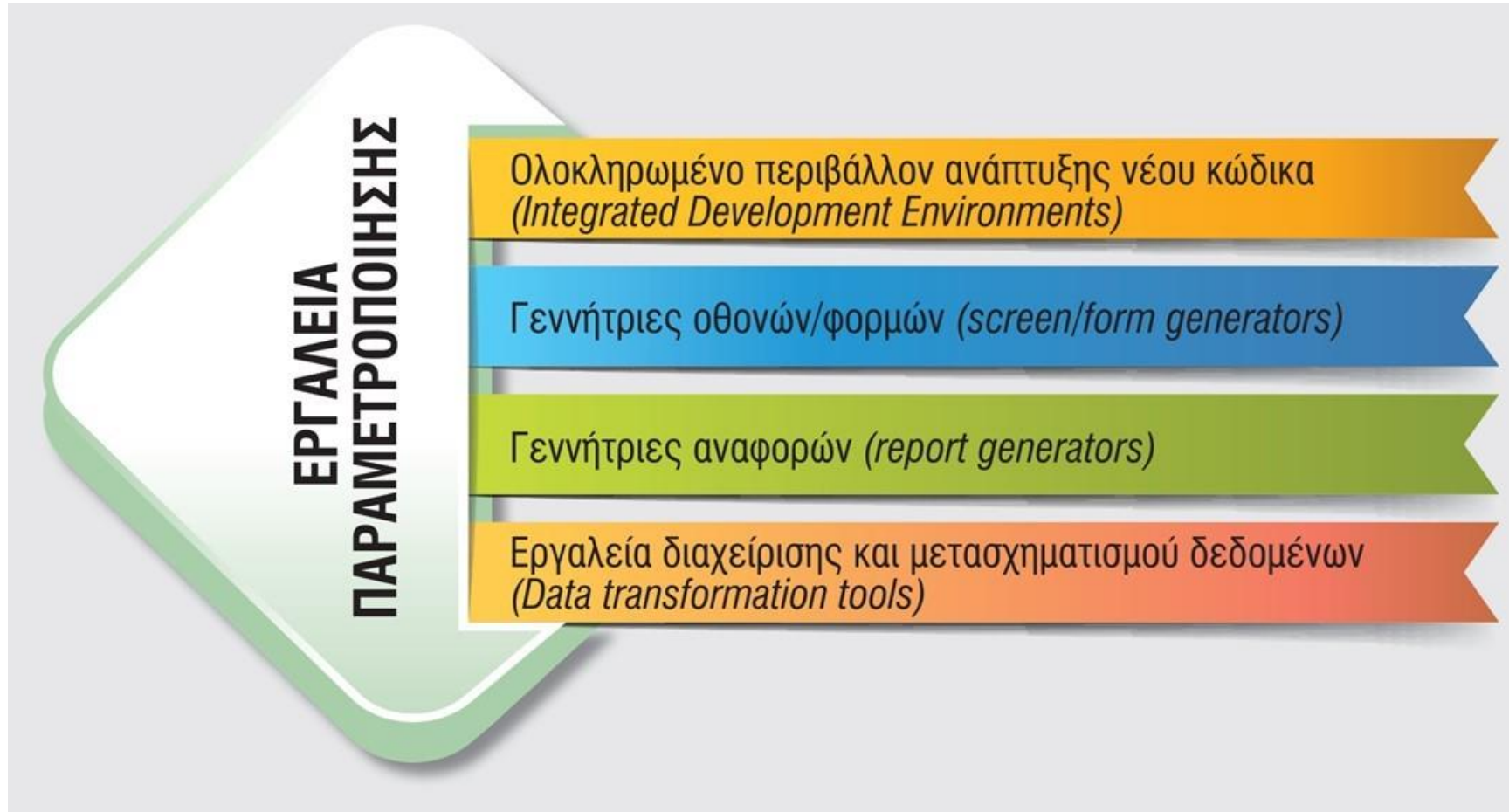


# Πρότυπα διαχείρισης εγγράφων





# Εργαλεία παραμετροποίησης





# Ένα ολοκληρωμένο περιβάλλον ανάπτυξης

## ΕΝΑ ΣΥΓΧΡΟΝΟ IDE ΥΠΟΣΤΗΡΙΖΕΙ:



την ανάπτυξη λογισμικού με συστατικά (components)



την υποστήριξη διαφορετικών γλωσσών προγραμματισμού μέσα στο ίδιο περιβάλλον



τη διαχείριση βιβλιοθηκών λογισμικού



την ανάπτυξη λογισμικού με συστατικά (components)



τη σύνδεση με διαφορετικές βάσεις δεδομένων



την αυτοματοποίηση της αποσφαλματοποίησης (debugging) και του ελέγχου



την ανάπτυξη του περιβάλλοντος διεπαφής με τους χρήστες με γραφικό τρόπο (visual programming)



τη σύνδεση με εργαλεία διαχείρισης σχηματισμών για τον καλύτερο έλεγχο πηγαίου κώδικα (configuration management)

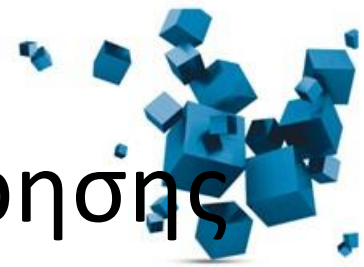


κ.α.



# Παράμετροι επιλογής αρχιτεκτονικής πληροφοριακού συστήματος

- Στις βέλτιστες συνιστώμενες πρακτικές οι θεωρήσεις που λαμβάνονται υπόψη για τον προσδιορισμό της αρχιτεκτονικής είναι:
  - το μέγεθος και η γεωγραφική κατανομή της επιχείρησης,
  - οι επιχειρηματικές λειτουργίες που θα υποστηριχθούν από το πληροφοριακό σύστημα,
  - ο αριθμός και το είδος των χρηστών καθώς και
  - ο όγκος των δεδομένων.



# Μέγεθος και γεωγραφική κατανομή της επιχείρησης

## Το μέγεθος της επιχείρησης μετράται ως εξής:

- κτηριακές εγκαταστάσεις,
- πλήθος εσωτερικών χρηστών,
- υποδομές ΤΠΕ εντός της επιχείρησης (σταθμοί εργασίας, τοπικά ή/και υπερτοπικά δίκτυα, υφιστάμενες εφαρμογές),
- ομάδες αποδεκτών και δυνητικός αριθμός εξυπηρετούμενων χρηστών,
- πλήθος προσφερόμενων προϊόντων ή/και υπηρεσιών, και
- γεωγραφική κάλυψη.

## Από την ανάλυση του μεγέθους της επιχείρησης προκύπτει:

- ανάγκες σε υποσυστήματα που θα εκτελούν τις επιχειρησιακές λειτουργίες της επιχείρησης,
- ανάγκες σε διαθεσιμότητα και αποκρισιμότητα εξυπηρετητών και εφαρμογών,
- ανάγκες σε άδειες χρήσης λογισμικού που θα καλύπτουν το πλήθος των χρηστών,
- ανάγκες για τη διαχείριση και αποθήκευση δεδομένων που παράγονται από το σύστημα,
- απαιτήσεις για κατανομή του πληροφοριακού συστήματος σε γεωγραφικές περιοχές,
- χαρακτηριστικά γεωγραφικών περιοχών και πιθανοί κίνδυνοι.



# Χρήστες του συστήματος







# Όγκος και είδος των δεδομένων

## ΔΕΔΟΜΕΝΑ ΕΠΙΧΕΙΡΗΣΗΣ

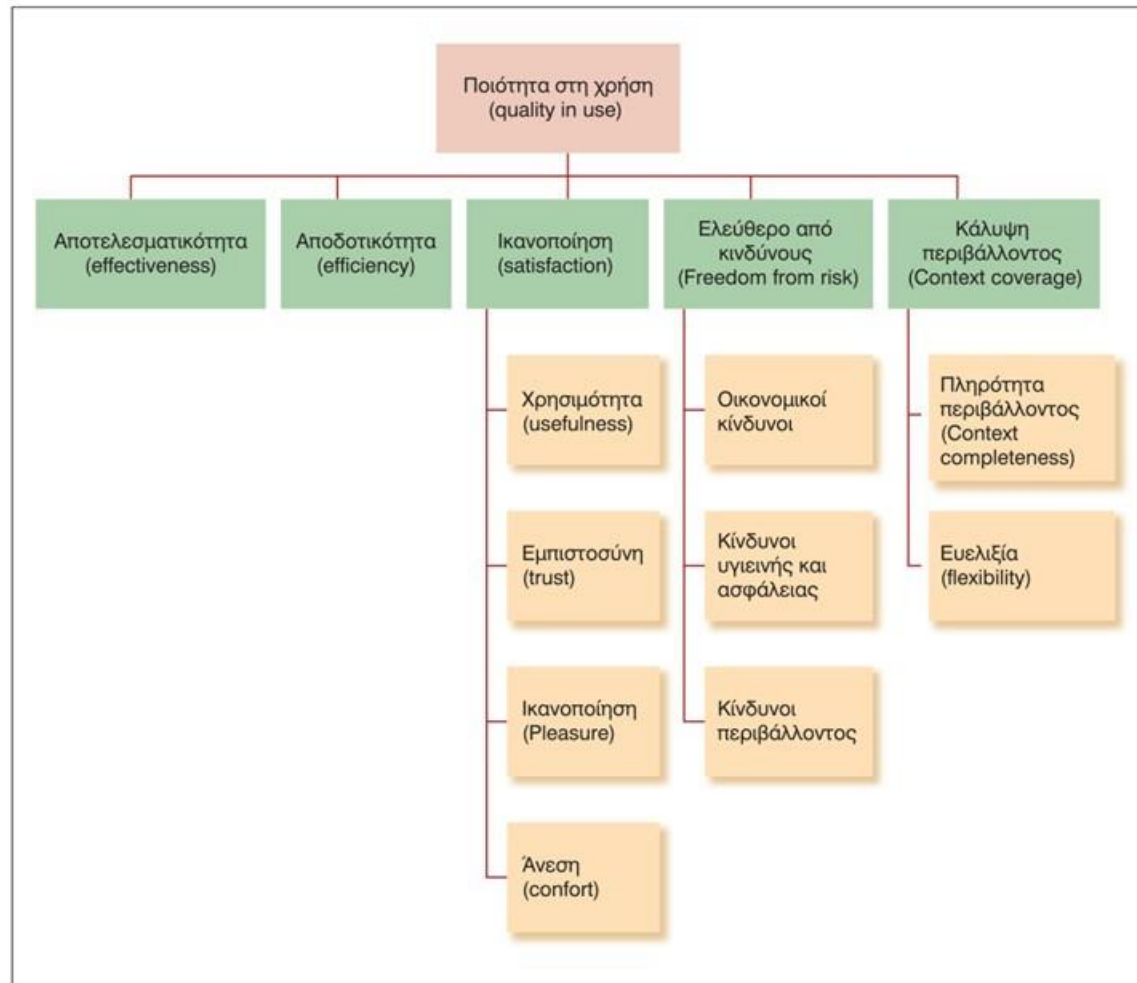
έγγραφα και αρχεία που παράγονται από την επιχείρηση

• εγγραφές που παράγονται κατά την εκτέλεση των επιχειρησιακών λειτουργιών και πρέπει να τηρούνται στο σύστημα.

• δεδομένα που παράγονται από αισθητήρες που ελέγχουν τις λειτουργίες του πληροφοριακού συστήματος

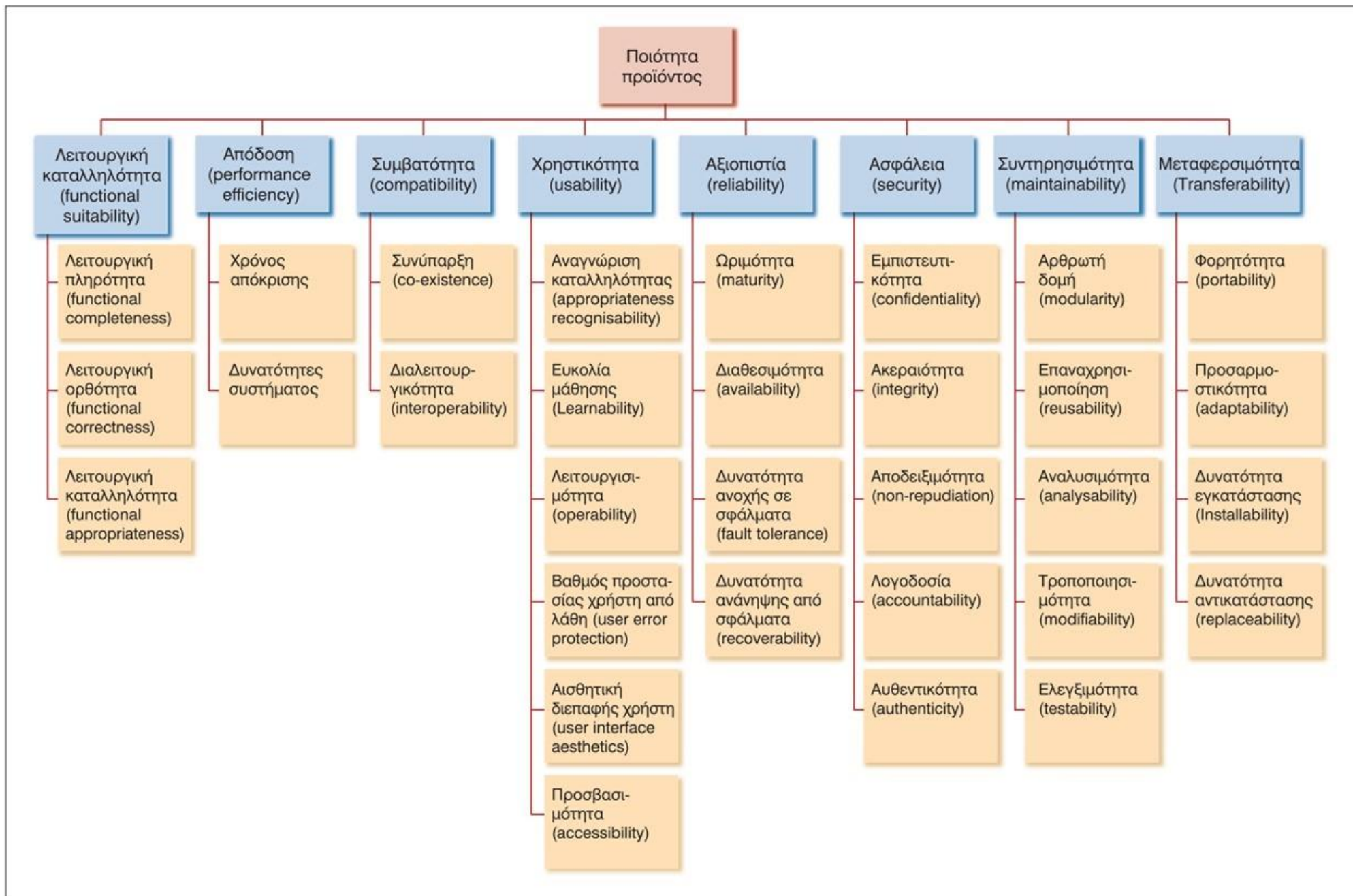
• δεδομένα που προκύπτουν από τα ημερολόγια καταγραφής των συμβάντων του πληροφοριακού συστήματος.

# Ποιότητα πληροφοριακών συστημάτων – ISO / IEC 25010:2011



ΕΙΚΟΝΑ 2.25

Η ποιότητα στη χρήση πληροφοριακού συστήματος σύμφωνα με το ISO / IEC 25010: 2011.



**ΕΙΚΟΝΑ 2.26**

Η ποιότητα πληροφοριακού συστήματος (προϊόντος) σύμφωνα με το ISO / IEC 25010: 2011.

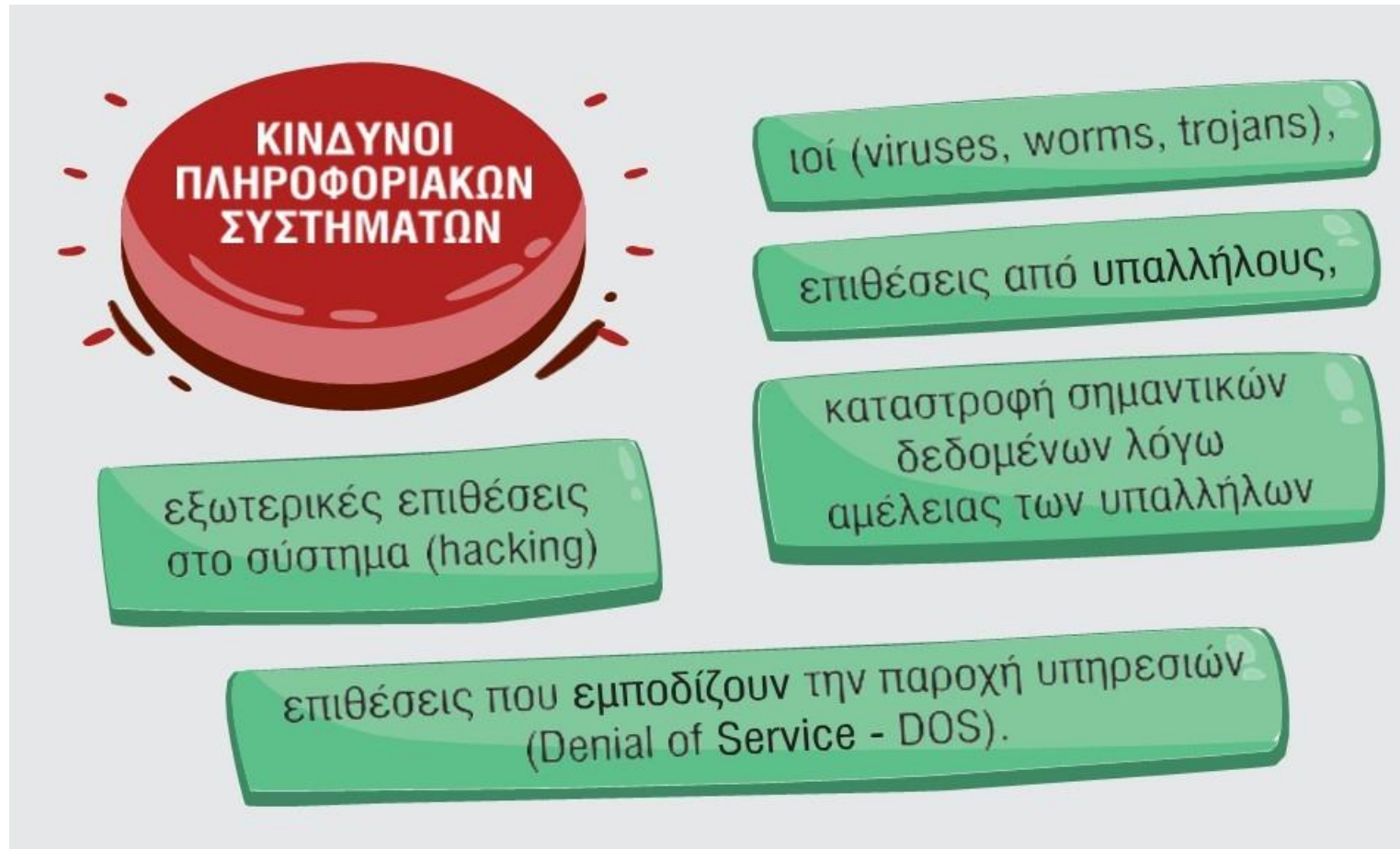


# Διαλειτουργικότητα





# Απαιτήσεις ασφαλείας





# Απαιτήσεις απόδοσης



ΠΑΝΟΣ ΦΙΤΣΙΛΗΣ

# ΣΥΓΧΡΟΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΕΠΙΧΕΙΡΗΣΕΩΝ

ERP-CRM-BPR

2η έκδοση

# Η Αρχιτεκτονική των Πληροφοριακών Συστημάτων Επιχειρήσεων

Καθηγητής Δρ. Πάνος Φιτσιλής