



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τεχνολογία Λογισμικού, 7ο/9ο εξάμηνο 2018-2019

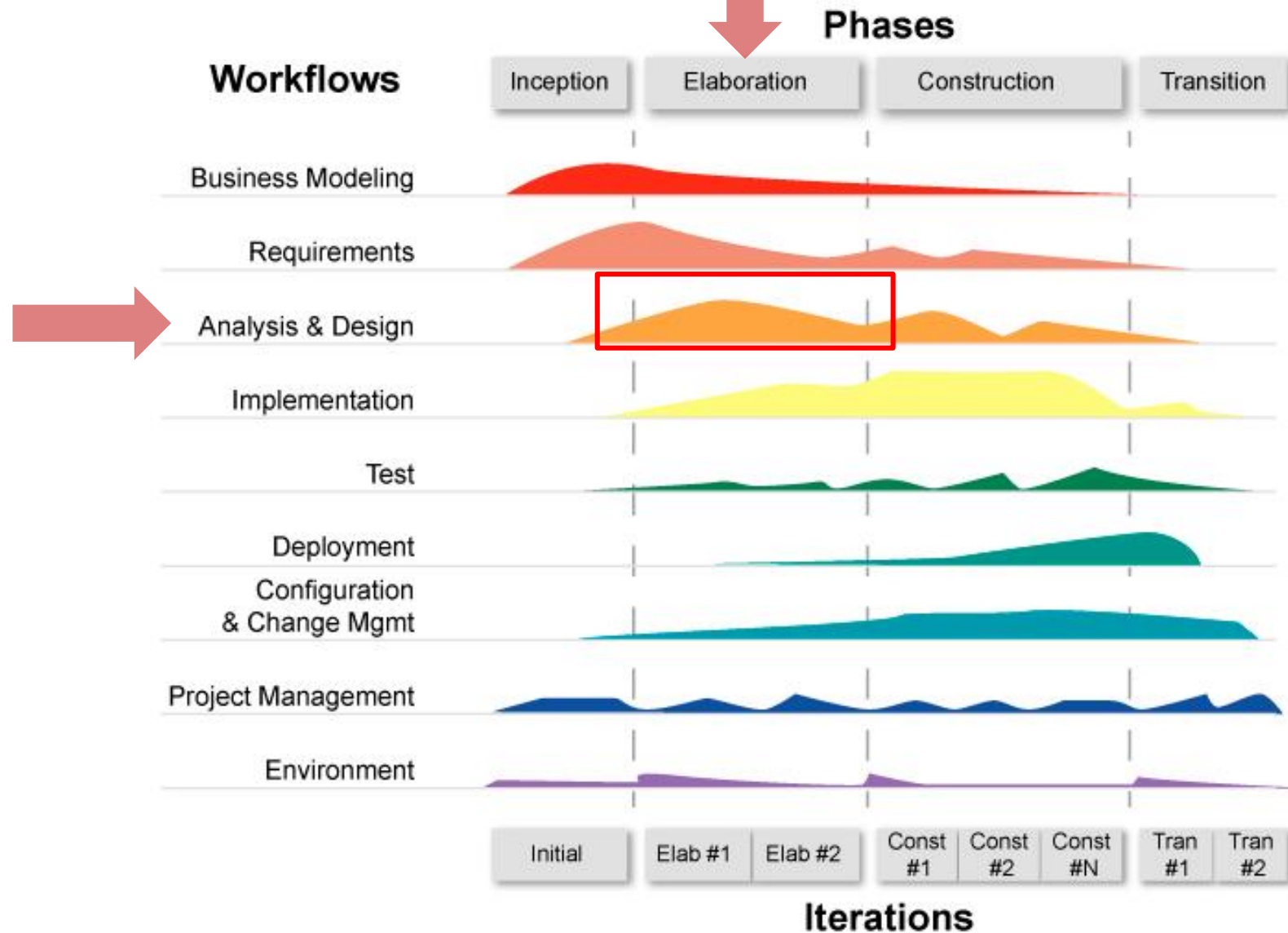
Τεχνολογία Λογισμικού

Ν. Παπασπύρου, Αν. Καθ. ΣΗΜΜΥ, nickie@softlab.ntua.gr

Β. Βεσκούκης, Αν. Καθ. ΣΑΤΜ, v.vescoukis@cs.ntua.gr

Κ. Σαΐδης, ΠΔ 407, saiko@softlab.ntua.gr

Φάσεις στη RUP: resource histogram



Δομημένη ανάλυση και σχεδιασμός λογισμικού

Δομημένη ανάλυση και σχεδίαση

Δομημένη σε αντιδιαστολή με τι;

- Εγκυκλοπαιδική γνώση: spaghetti programming 1960's, 1970's, 1980's, ...

Κεντρική ιδέα

- Η παράσταση των στοιχείων μιας εφαρμογής λογισμικού με όρους «δεδομένων» και «μετασχηματισμών»
- Η ταυτοποίηση των δεδομένων (ορολογία, λεξικό)
- Εύκολη μετάβαση σε στοιχεία υλοποίησης σε περιβάλλον δομημένου προγραμματισμού (**χωρίς go-to**), πχ συναρτήσεις (functions)

Διάγραμμα ροής δεδομένων

- Παραμένει και σήμερα εργαλείο business modeling

Δομημένη ανάλυση και σχεδίαση – βασικές αρχές

Τα δεδομένα είναι ανεξάρτητα από τις μονάδες λογισμικού που τα διαχειρίζονται / μετατρέπουν / δημιουργούν / αποθηκεύουν κλπ

Η οργάνωση των δεδομένων περιγράφεται σε επίπεδο εννοιολογικό (ανάλυση: ERD) και κατασκευαστικό (σχεδίαση: RDB)

Η συμπεριφορά των μονάδων λογισμικού προκύπτει (κυρίως) από την περιγραφή των λειτουργικών απαιτήσεων

Οι μονάδες λογισμικού ανταλλάσσουν δεδομένα προκειμένου να επιτελέσουν το σκοπό τους

Μπορεί να χρησιμοποιούνται επιπλέον περιγραφές (πχ διαγράμματα μετάβασης καταστάσεων) όπου είναι απαραίτητο, ώστε να γίνεται σαφής η απαιτούμενη συμπεριφορά του λογισμικού

Μοντέλα παράστασης λογισμικού

Διαγράμματα (συνήθως) που παριστούν διαφορετικές όψεις ενός συστήματος λογισμικού (δομή, συμπεριφορά, διάταξη κ.ά.)

Προφανής η ανάγκη να είναι συνεπή

Προφανής, όχι εύκολα ικανοποιήσιμη...

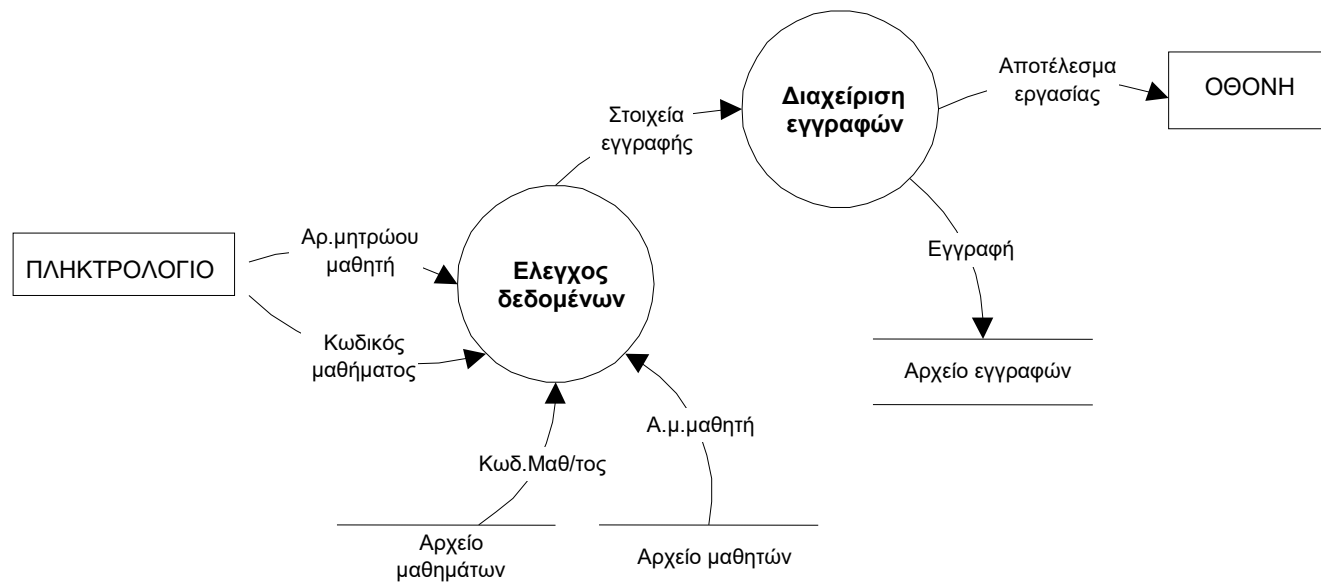
Παράδειγμα:



Διαγράμματα ροής δεδομένων

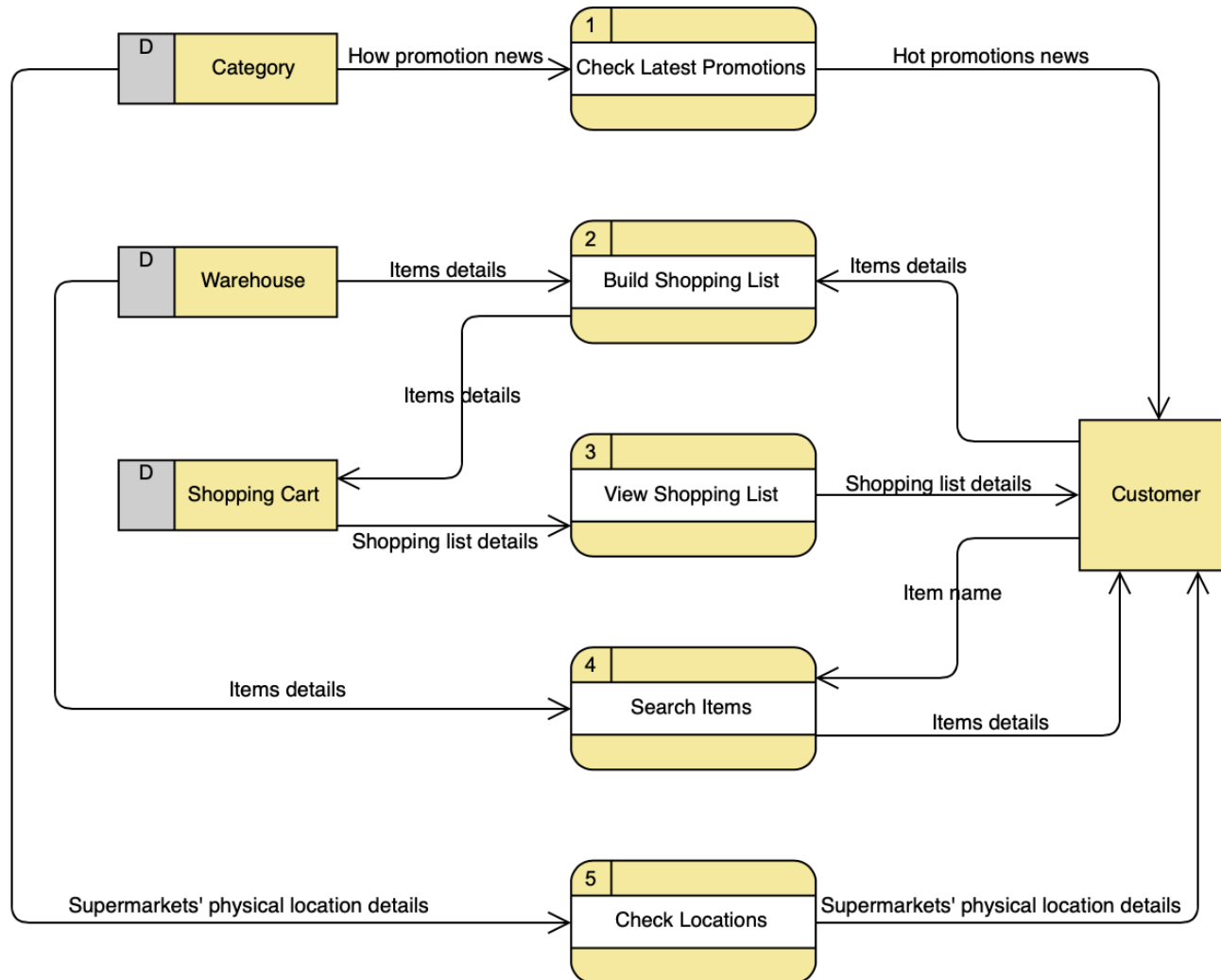
Τι είναι;

- Ένα δίκτυο όπου "ρέουν" δεδομένα τα οποία μετασχηματίζονται σε νέα δεδομένα από μετασχηματισμούς.
- Τα δεδομένα εισόδου μετασχηματίζονται μέχρις ότου προκύψουν τα επιθυμητά δεδομένα εξόδου.



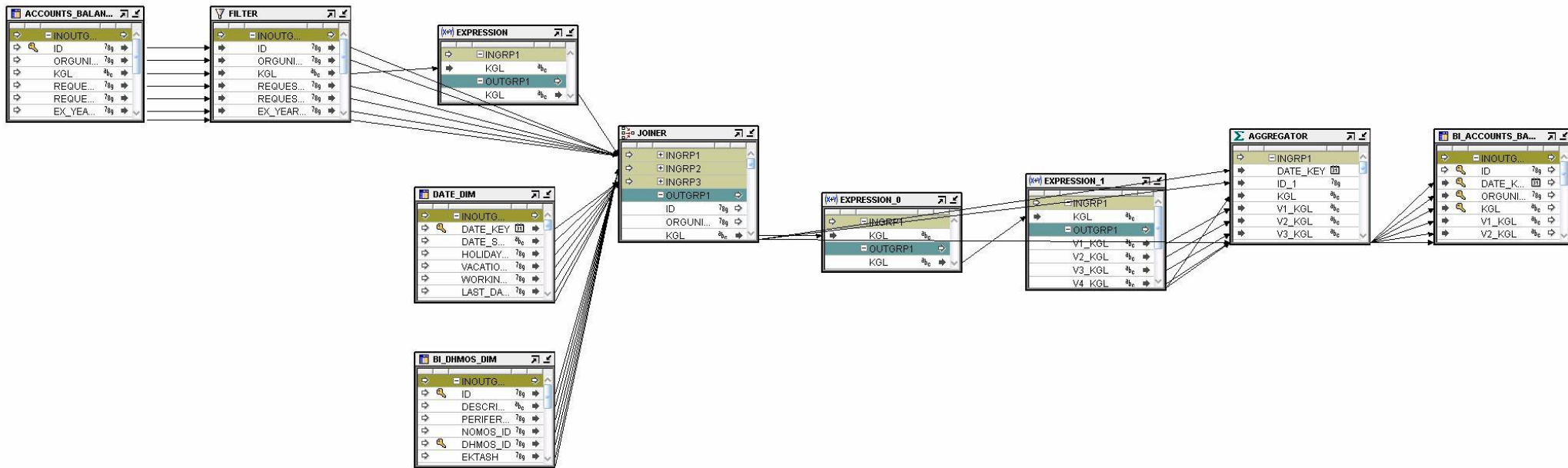
Διάγραμμα ροής δεδομένων

Παράδειγμα (online.visual-paradigm.com)



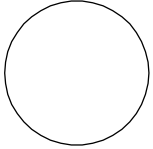


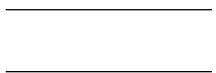
Διάγραμμα ροής δεδομένων

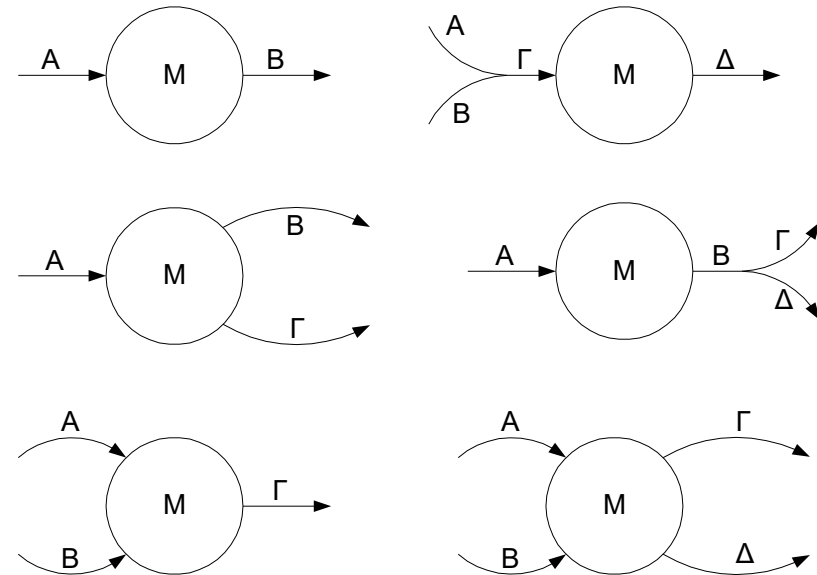
Μετασχηματισμοί ETL (= Extraction, Transformation, Loading)



Διαγράμματα ροής δεδομένων

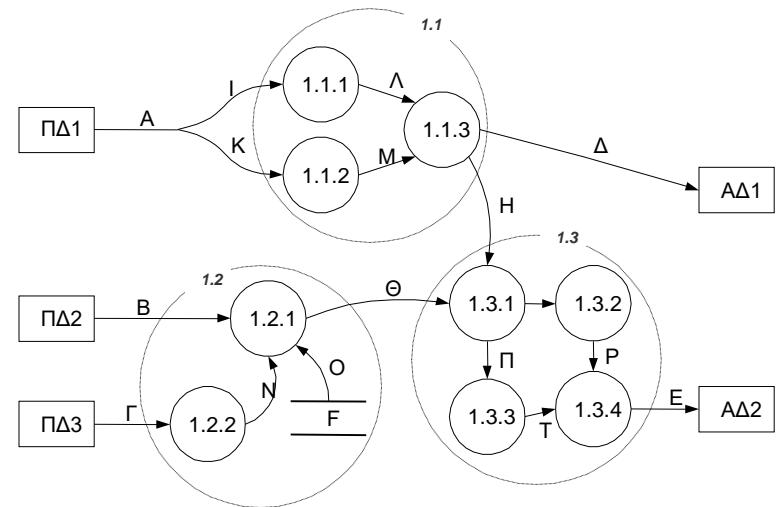
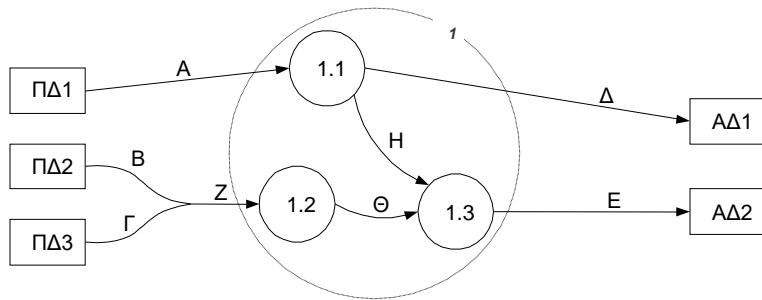
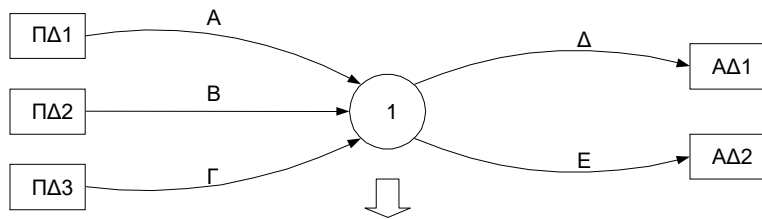
Συμβολισμοί και συμβάσεις

Συμβολισμοί διαγραμμάτων ροής δεδομένων	
	Διαδικασία / μετασχηματισμός δεδομένων
	Εξωτερική πηγή ή αποδέκτης δεδομένων
	Ροή δεδομένων
	Αποθήκη δεδομένων

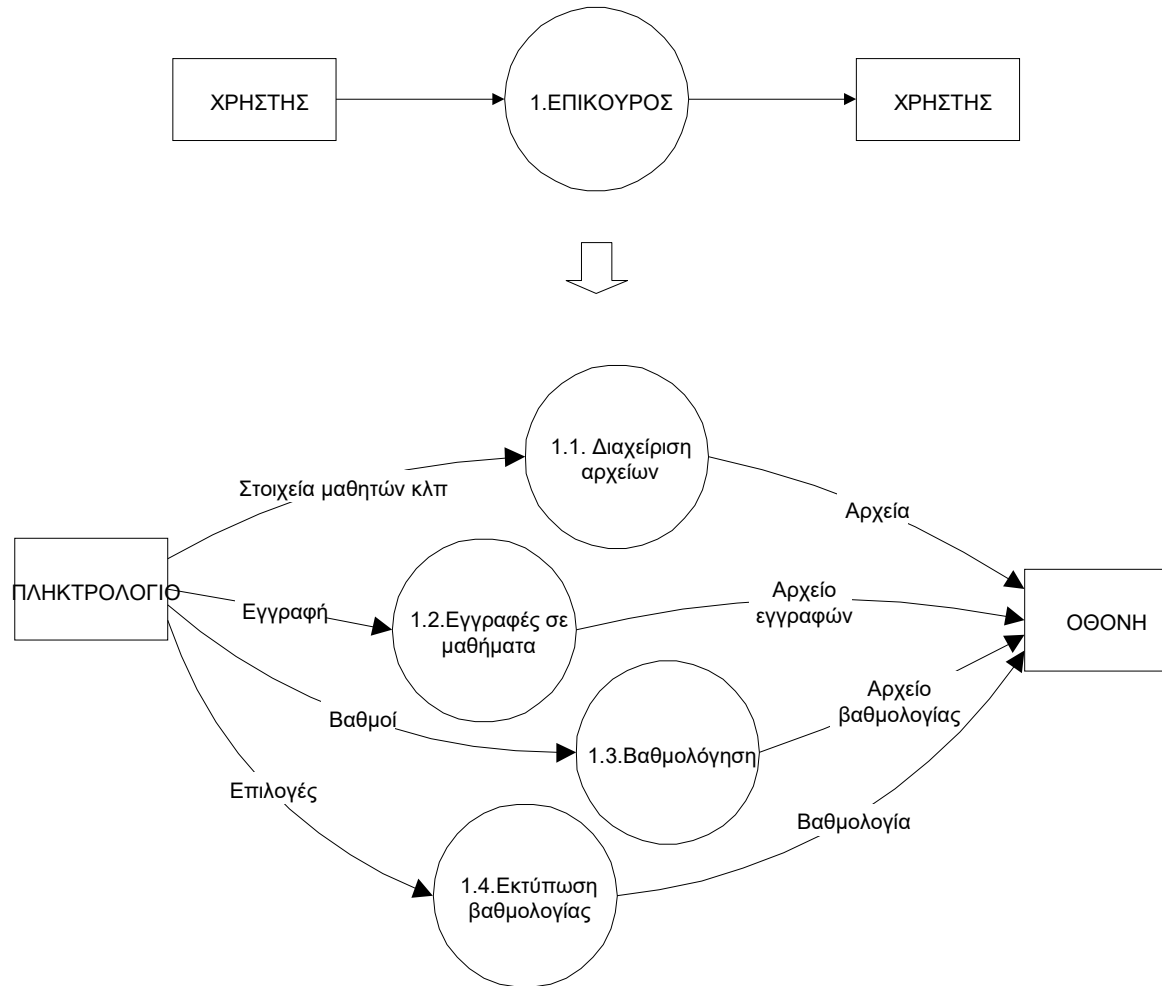


Διαγράμματα ροής δεδομένων

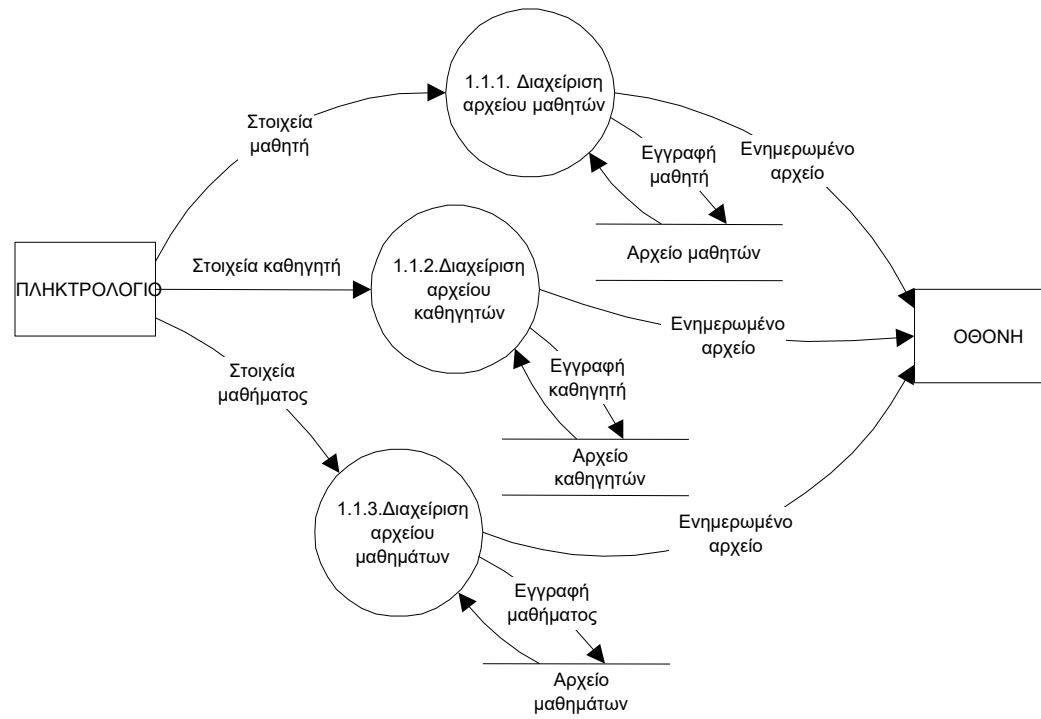
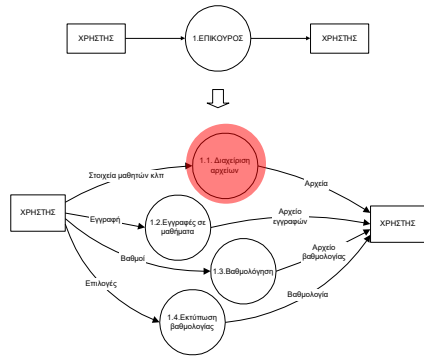
Διαδοχικά επίπεδα λεπτομέρειας



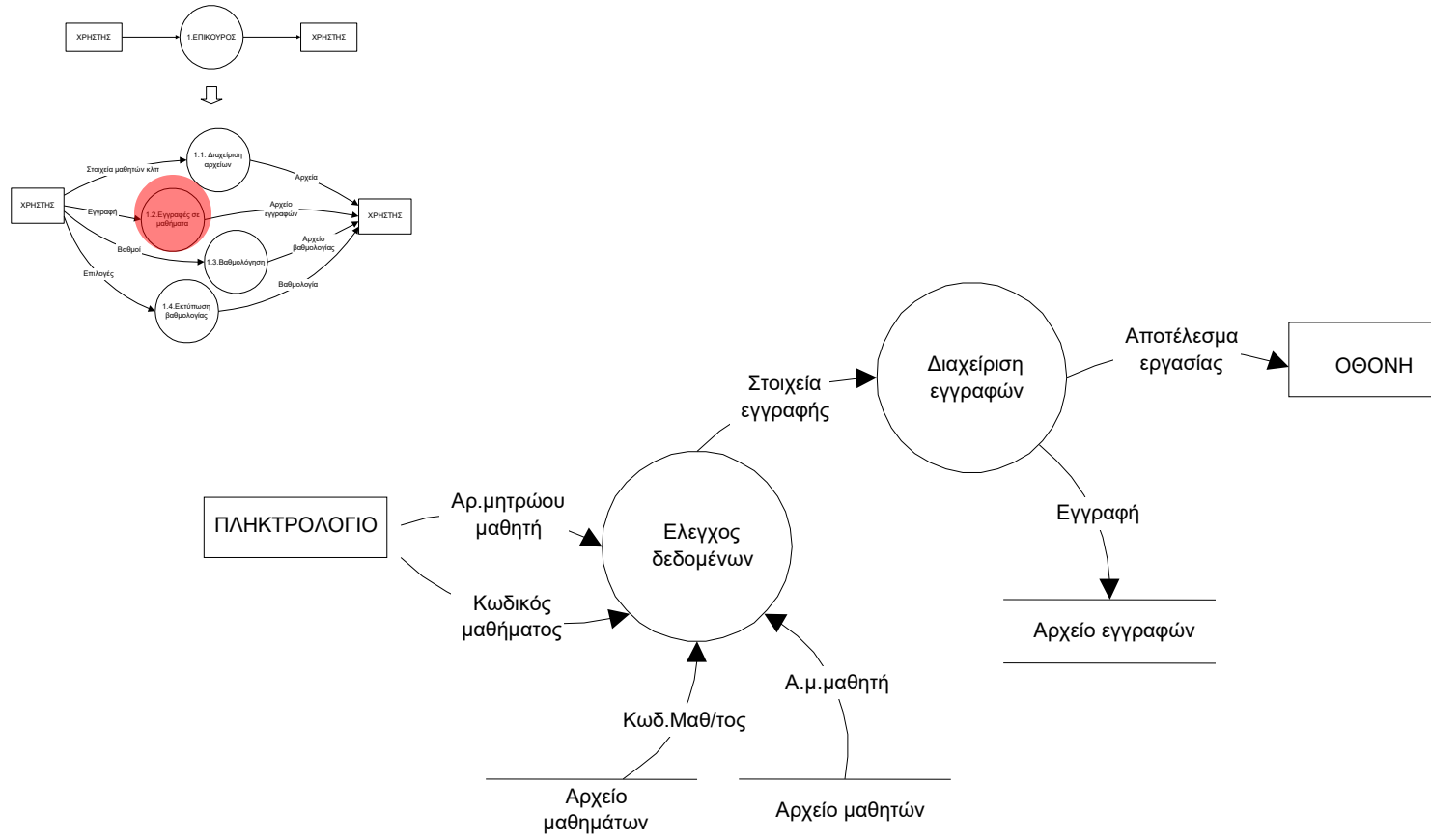
Διαγράμματα ροής δεδομένων - παράδειγμα



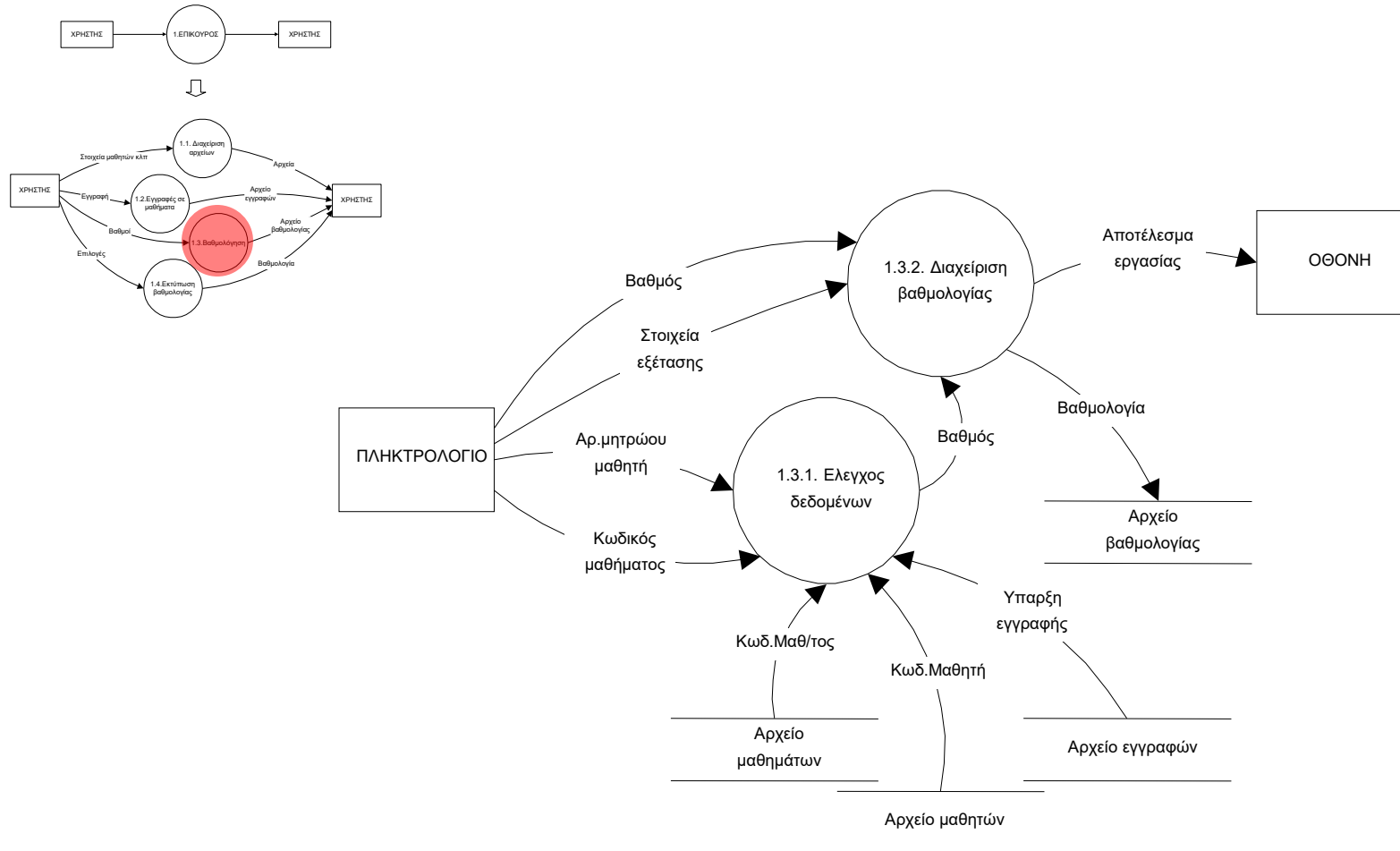
Διαγράμματα ροής δεδομένων – παράδειγμα



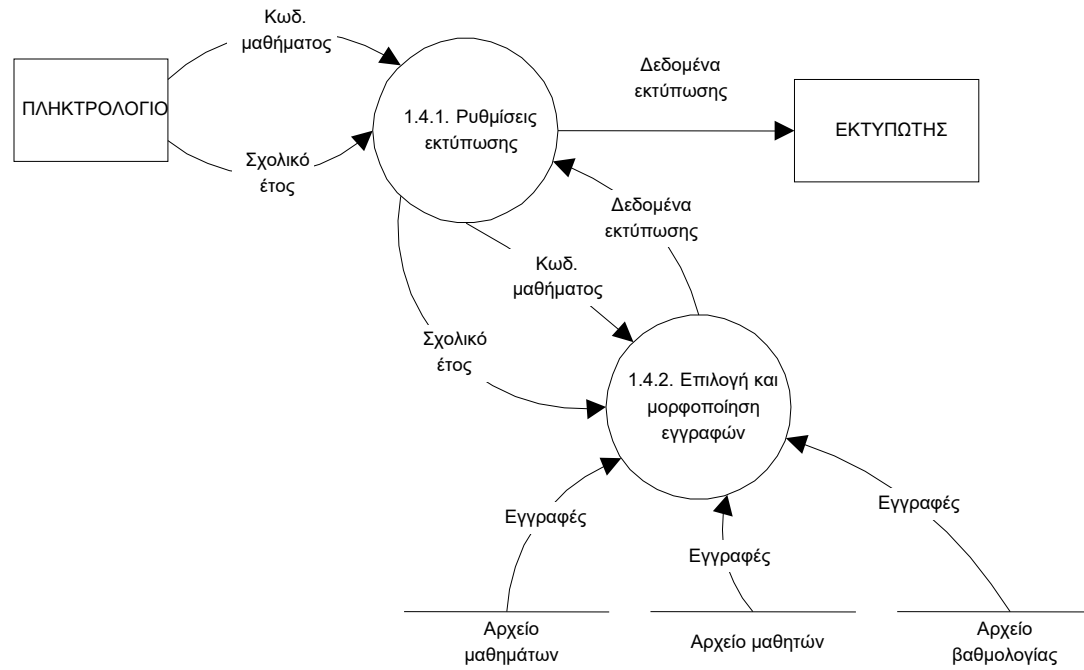
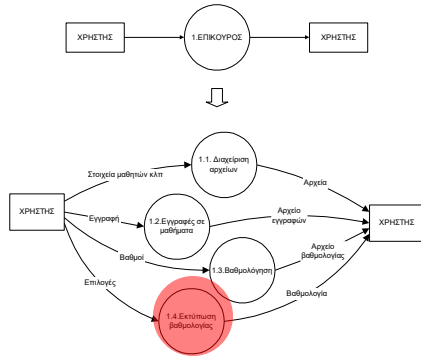
Διαγράμματα ροής δεδομένων – παράδειγμα



Διαγράμματα ροής δεδομένων – παράδειγμα



Διαγράμματα ροής δεδομένων – παράδειγμα



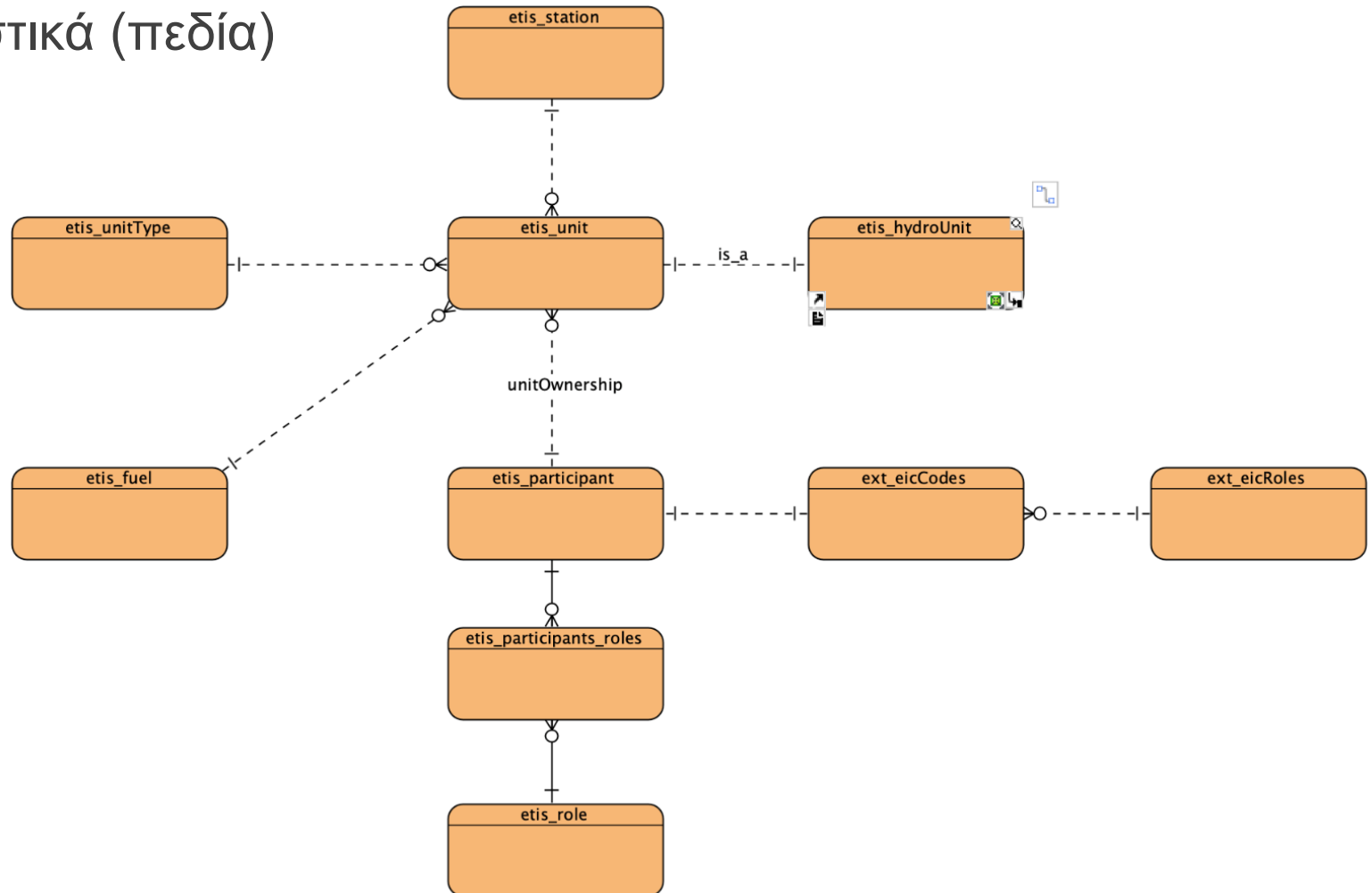
Διαγράμματα ροής δεδομένων

Χρήσιμες συμβουλές

- Σημασία στα κρίσιμα δεδομένα της εφαρμογής
- Κατά τη μετάβαση σε νέο επίπεδο λεπτομέρειας, αναλύονται και οι μετασχηματισμοί και τα δεδομένα
- Να μη συγχέεται με παράσταση αλγορίθμου
- Δεν παριστάνεται πληροφορία χρονισμού
- Ισορροπία μεταξύ ουσιώδους και κατασκευαστικής λεπτομέρειας
- Χρήση κατάλληλων εργαλείων

Εννοιολογικά μοντέλα δεδομένων (ERD)

Οντότητες και συσχετίσεις
Χαρακτηριστικά (πεδία)



Διαγράμματα μετάβασης καταστάσεων

Γεγονός

- Ένα γεγονός (event) είναι μια στιγμιαία μεταβολή στο περιβάλλον λειτουργίας του λογισμικού, η οποία προκαλείται από εξωτερικούς παράγοντες (χρήστης, λειτουργικό σύστημα, άλλες εφαρμογές λογισμικού).

Απόκριση

- Μια λειτουργία που εκτελεί το λογισμικό όταν προκαλείται ένα γεγονός, ονομάζεται απόκριση (response).

Κατάσταση

- Όταν το λογισμικό αναμένει γεγονότα, τότε λέμε ότι βρίσκεται σε μία κατάσταση. Όταν συμβεί ένα γεγονός, το λογισμικό μπορεί να εκτελεί μια λειτουργία και να μεταβαίνει σε μια άλλη κατάσταση.

Διαγράμματα μετάβασης καταστάσεων

Ένα διάγραμμα μετάβασης καταστάσεων...



- Έχει μια κατάσταση έναρξης και μια κατάσταση τέλους.
- Περιέχει τις δυνατές καταστάσεις, τα γεγονότα και τις μεταβάσεις
 - Για ολόκληρη την εφαρμογή ή
 - Για οποιοδήποτε τμήμα αυτής (μονάδα, υποσύστημα λογισμικού)

Κατασκευάζουμε ένα διάγραμμα μετάβασης καταστάσεων όταν...

- Κάνει σαφέστερη την περιγραφή της συμπεριφοράς του λογισμικού
- Κάνει ευκολότερη την περιγραφή της υλοποίησής του, δηλαδή της παραγωγής πηγαίου κώδικα
- Περιγράφουμε την κατάσταση δεδομένων

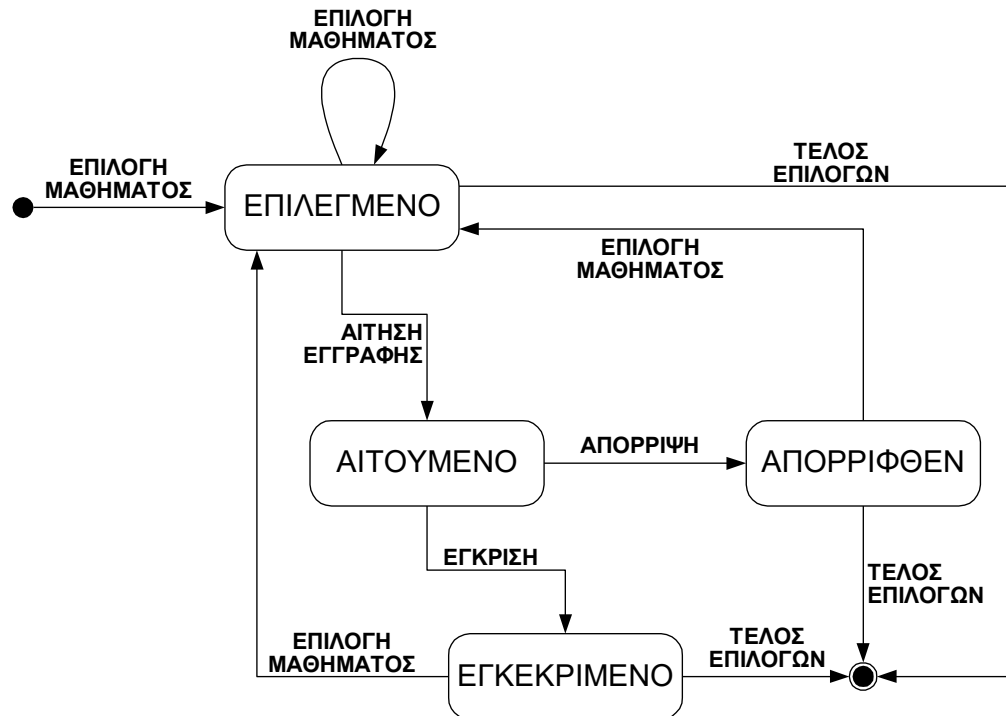
Διαγράμματα μετάβασης καταστάσεων

Συμβολισμοί

Συμβολισμοί διαγραμμάτων μετάβασης καταστάσεων	
	Κατάσταση
	Κατάσταση έναρξης
	Κατάσταση τέλους
	Μετάβαση σε άλλη κατάσταση / λειτουργία που εκτελείται
	Μετάβαση στην ίδια κατάσταση / λειτουργία που εκτελείται

Διαγράμματα μετάβασης καταστάσεων

Ένα παράδειγμα (εγγραφή σε μάθημα)



Το λεξικό δεδομένων

Μια οργανωμένη ταξινόμηση όλων περιγραφών των δεδομένων που αναφέρονται στα μοντέλα παράστασης λογισμικού (διαγράμματα κλπ)

- Με σαφήνεια και πληρότητα
- Ωστε αυτά να γίνονται κατανοητά τόσο από τους συμμετέχοντες στην ανάπτυξη και χρήση (;) του συστήματος

Το λεξικό δεδομένων

- Είναι βασικό στοιχείο της δομημένης ανάλυσης
- Κατασκευάζεται με πολλή δουλειά
- Και χρειάζεται περισσότερη για να διατηρηθεί ενημερωμένο

Το λεξικό δεδομένων

Ενας πίνακας (ή ΒΔ) που καταγράφει...

- **Ονομασία.** Το κύριο αναγνωριστικό της οντότητας, πεδίου ή ροής δεδομένων.
- **Βοηθητικές ονομασίες.** Ονομασίες που χρησιμοποιούνται ισοδύναμα (aliases).
- **Πού χρησιμοποιείται.** Αναφορά στους μετασχηματισμούς, οντότητες κλπ οι οποίοι χρησιμοποιούν το εν λόγω στοιχείο.
- **Πώς χρησιμοποιείται.** Αναφορά στον τρόπο με τον οποίο χρησιμοποιείται το εν λόγω στοιχείο (ως στοιχείο εισόδου, ως αποτέλεσμα, πεδίο, κ.ά.)
- **Τι περιέχει.** Περιγραφή του είδους και της μορφής της πληροφορίας που αποθηκεύεται σε αυτό.
- **Ορια τιμών.** Καθορισμός των επιτρεπτών τιμών που μπορεί να πάρει (αν απαιτείται).
- **Αρχική τιμή.** Καθορισμός της αρχικής τιμής του στοιχείου (αν απαιτείται).
- **Λοιπά στοιχεία.** Υπόλοιπες χρήσιμες πληροφορίες.

Το λεξικό δεδομένων

Παράδειγμα από μια εφαρμογή γραμματείας σχολείου
















ΟΝΟΜΑΣΙΑ	ΑΛΛΕΣ ΟΝΟΜΑΣΙΕΣ	ΠΟΥ	ΠΩΣ	ΠΕΡΙΕΧΟΜΕΝΑ	ΟΡΙΑ
ΚΑΘΗΓΗΤΗΣ	ΚΑΘ	Βάση Δεδομένων	Πίνακας της ΒΔ	Εγγραφές καθηγητών	-
ΜΑΘΗΜΑ	ΜΑΘ	Βάση Δεδομένων	Πίνακας της ΒΔ	Εγγραφές μαθημάτων	-
ΒΑΘΜΟΛΟΓΙΑ	ΒΑΘΜ	Βάση Δεδομένων	Πίνακας της ΒΔ	Εγγραφές βαθμολογίας	-
ΦΟΙΤΗΤΗΣ	ΦΟΙΤ	Βάση Δεδομένων	Πίνακας της ΒΔ	Εγγραφές φοιτητών	-
ΑΡ. ΤΑΥΤΟΤΗΤΑΣ	Α.Τ.	Πίνακας "ΚΑΘΗΓΗΤΗΣ"	Πεδίο	Πεδίο 7 χαρακτήρων	-
ΟΝΟΜΑ	ΟΝ	Πίνακας "ΚΑΘΗΓΗΤΗΣ"	Πεδίο	Πεδίο 25 χαρακτήρων	-
ΕΠΩΝΥΜΟ	ΕΠ	Πίνακας "ΚΑΘΗΓΗΤΗΣ"	Πεδίο	Πεδίο 25 χαρακτήρων	-
ΔΙΕΥΘΥΝΣΗ	ΔΙΕΥΘ	Πίνακας "ΚΑΘΗΓΗΤΗΣ"	Πεδίο	Πεδίο 50 χαρακτήρων	-
ΤΗΛΕΦΩΝΟ	ΤΗΛ	Πίνακας "ΚΑΘΗΓΗΤΗΣ"	Πεδίο	Πεδίο 20 χαρακτήρων	-
ΑΡ. ΤΑΥΤΟΤΗΤΑΣ	Α.Τ.	Πίνακας "ΦΟΙΤΗΤΗΣ"	Πεδίο	Πεδίο 7 χαρακτήρων	-
ΟΝΟΜΑ	ΟΝ.Φ.	Πίνακας "ΦΟΙΤΗΤΗΣ "	Πεδίο	Πεδίο 25 χαρακτήρων	-

Το λεξικό δεδομένων

Παράδειγμα (συνέχεια)

ΟΝΟΜΑΣΙΑ	ΑΛΛΕΣ ΟΝΟΜΑΣΙΕΣ	ΠΟΥ	ΠΩΣ	ΠΕΡΙΕΧΟΜΕΝΑ	ΟΡΙΑ
ΕΠΩΝΥΜΟ	ΕΠΦ.	Πίνακας "ΦΟΙΤΗΤΗΣ "	Πεδίο	Πεδίο 25 χαρακτήρων	-
ΔΙΕΥΘΥΝΣΗ	ΔΙΕΥΘ.Φ.	Πίνακας "ΦΟΙΤΗΤΗΣ "	Πεδίο	Πεδίο 50 χαρακτήρων	-
ΤΗΛΕΦΩΝΟ	ΤΗΛ.Φ.	Πίνακας "ΦΟΙΤΗΤΗΣ "	Πεδίο	Πεδίο 20 χαρακτήρων	-
ΤΜΗΜΑ	ΤΜ.Φ.	Πίνακας "ΦΟΙΤΗΤΗΣ "	Πεδίο	Πεδίο 20 χαρακτήρων	-
ΚΩΔΙΚΟΣ ΜΑΘ	Κ.Μ.	Πίνακας "ΜΑΘΗΜΑ"	Πεδίο	Πεδίο 10 αριθμητικών ψηφίων	-
ΘΕΜΑΤΙΚΗ ΕΝΟΤΗΤΑ	Θ.ΕΝ.	Πίνακας "ΜΑΘΗΜΑ"	Πεδίο	Πεδίο 10 χαρακτήρων	-
ΤΙΤΛΟΣ	-	Πίνακας "ΜΑΘΗΜΑ"	Πεδίο	Πεδίο 50 χαρακτήρων	-
ΔΙΔΑΣΚΩΝ	ΔΙΔ	Πίνακας "ΜΑΘΗΜΑ"	Πεδίο	Πεδίο 7 χαρακτήρων	-
ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ	ΗΜ.ΕΞ.	Πίνακας "ΒΑΘΜΟΛΟΓΙΑ"	Πεδίο	Πεδίο ημερομηνίας (ΗΗ/ΜΜ/ΕΕΕΕ)	< τρέχουσα ημερομηνία
ΕΙΔΟΣ ΕΞΕΤΑΣΗΣ	ΕΙΔ.ΕΞ.	Πίνακας "ΒΑΘΜΟΛΟΓΙΑ"	Πεδίο	Πεδίο 10 χαρακτήρων	Ενδιάμεση, Τελική
ΒΑΘΜΟΣ	ΒΑΘΜ.	Πίνακας "ΒΑΘΜΟΛΟΓΙΑ"	Πεδίο	Αριθμητικό πεδίο	0-10
ΠΑΡΑΤΗΡΗΣΕΙΣ	-	Πίνακας "ΒΑΘΜΟΛΟΓΙΑ"	Πεδίο	Πεδίο 100 χαρακτήρων	-

Ένα (σημερινό) λεξικό δεδομένων

Entity Name	Entity Description	Column Name	Column Description	Data Type	Length	Primary Key	Nullable	Unique
 ACCOUNT_C HART_TYPE		 CODE		varchar2	60	false	true	false
		 DESCRIPTION		varchar2	300	false	true	false
		 ID		number	19	true	false	false
 ACCOUNT_R ECORD		 APPROVED		number	15.2	false	false	false
		 BOUNDS_RECORD		number	1	false	false	false
		 DATE_INSERTED		date	0	false	true	false
		 ID	id	number	19	true	false	false
		 IS_DELETED		char	1	false	true	false
		 KAE		varchar2	32	false	true	false
		 LOG_ID		number	255	false	true	false
		 LOGGED_MONTH		number	5	false	true	false
		 LOGGED_YEAR		number	5	false	true	false
		 ORGANIZATION_CODE		number	1.2	false	true	false

Σχεδίαση λογισμικού

Κατά την ανάλυση ασχολούμαστε με το τι θα κάνει το λογισμικό και τι χαρακτηριστικά θα έχει

Στη σχεδίαση, ασχολούμαστε με το πώς θα επιτύχουμε την επιθυμητή συμπεριφορά και την απόδοση των απαιτούμενων χαρακτηριστικών

Θα πρέπει να εντοπίσουμε

- Ποια συστατικά αποτελούν την εφαρμογή μας
- Πώς αυτά συσχετίζονται μεταξύ τους
- Πώς διατάσσονται (deploy)
- Ποια είναι η εσωτερική τους δομή

Οι απαντήσεις στα ερωτήματα αυτά αποτελούν το σχέδιο του λογισμικού

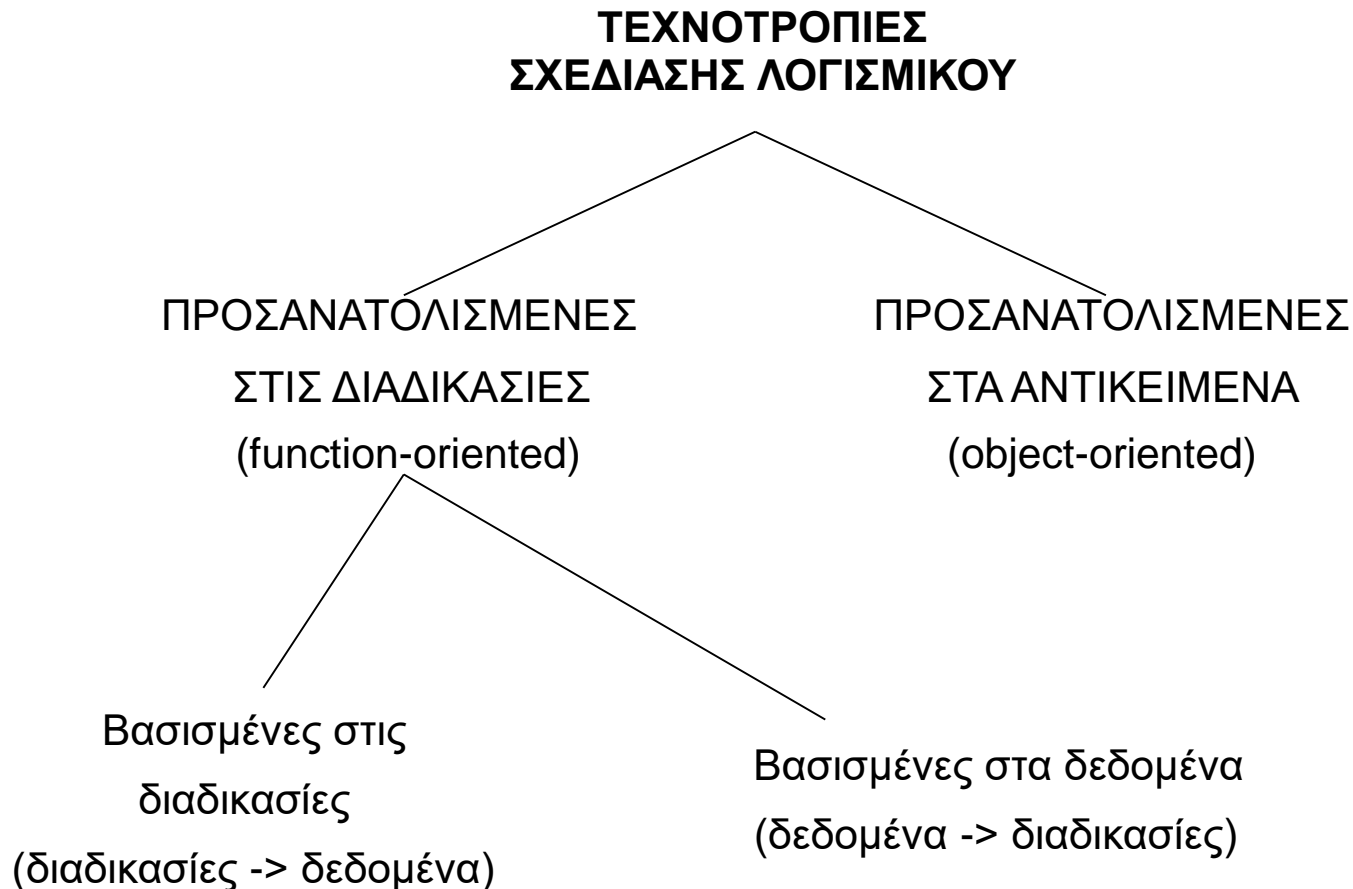
Σχεδίαση λογισμικού

Ένα σύνθετο πρόβλημα:

- Ποιος είναι ο καλύτερος τρόπος μετάβασης από τις προδιαγραφές στο σχέδιο του λογισμικού;
- Με πόσα σχέδια μπορεί να υλοποιηθεί κάθε προδιαγραφή και ποιο είναι το «καλύτερο»;
- Η σχεδίαση περιορίζεται ή υποστηρίζεται από το εκάστοτε περιβάλλον ανάπτυξης στο οποίο θα γίνει η υλοποίηση;
- Πόσο λεπτομερής είναι μια «καλή» αποτύπωση του σχεδίου και τι περιλαμβάνει αυτή;
- Πώς διασφαλίζεται η ποιότητα του λογισμικού;

Προσεγγίσεις σχεδίασης

Ανάλογα με τη φιλοσοφία ανάπτυξης, διακρίνουμε διαφορετικές προτεραιότητες στο πρόβλημα της σχεδίασης λογισμικού



Δομημένη σχεδίαση - εργασίες

Αρχιτεκτονική σχεδίαση

- Διάγραμμα δομής προγράμματος
- Διάγραμμα διάταξης

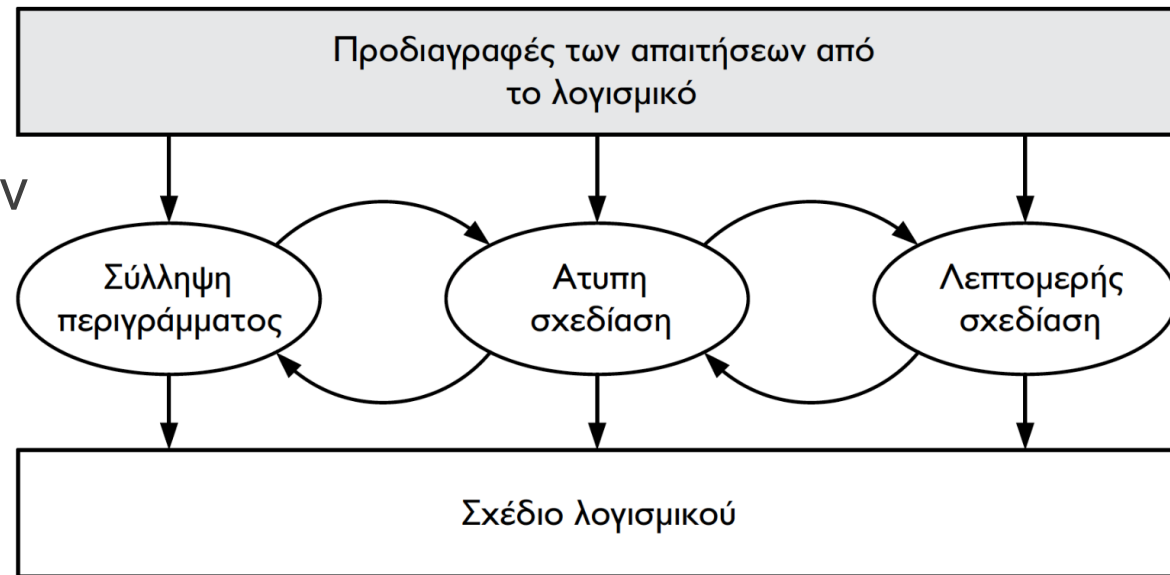
Σχεδίαση διεπαφών

Λεπτομερής σχεδίαση μονάδων

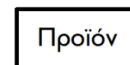
- Ψευδοκώδικας (!)

Σχεδίαση δεδομένων

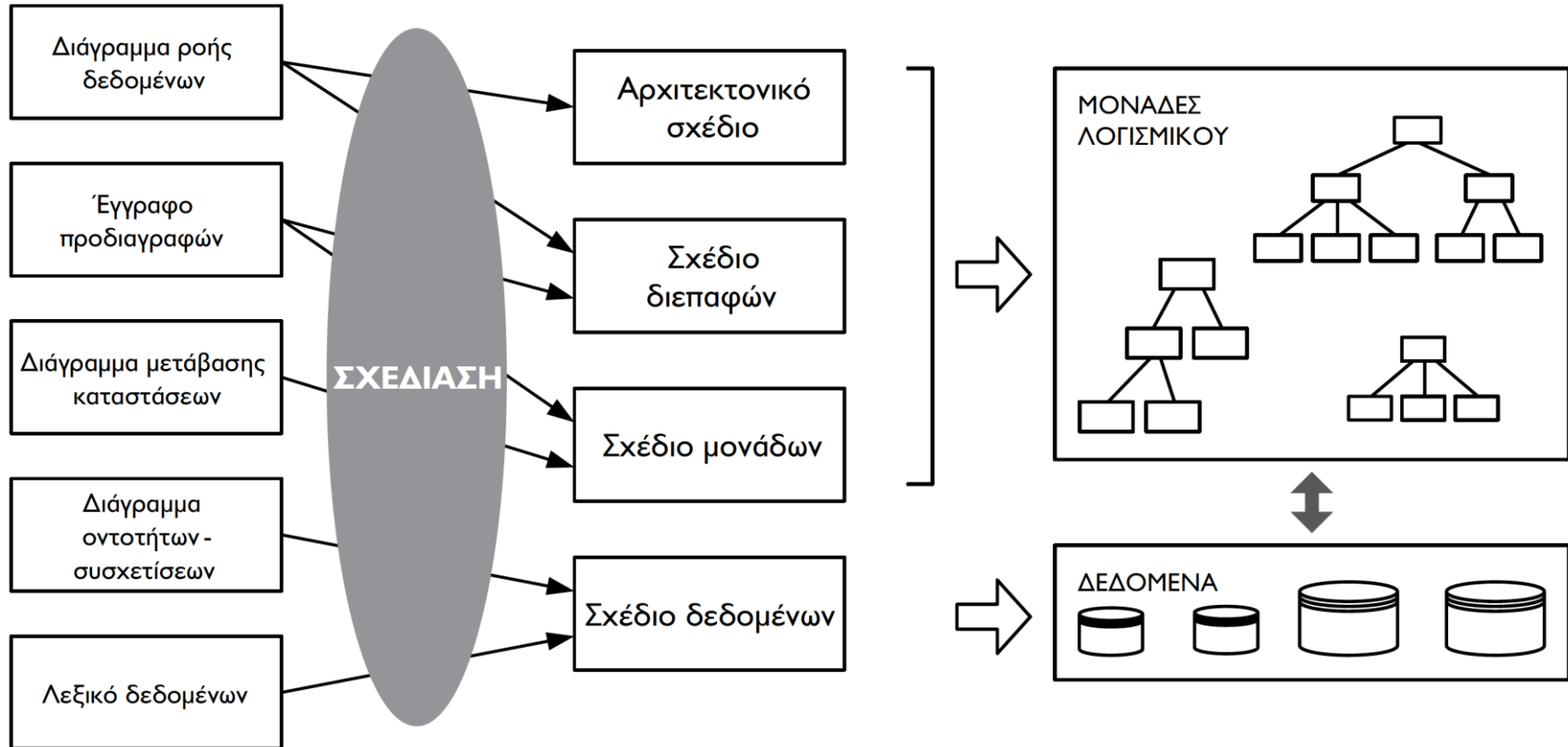
- Σχεσιακό μοντέλο



Συμβολισμοί :



Δομημένη σχεδίαση

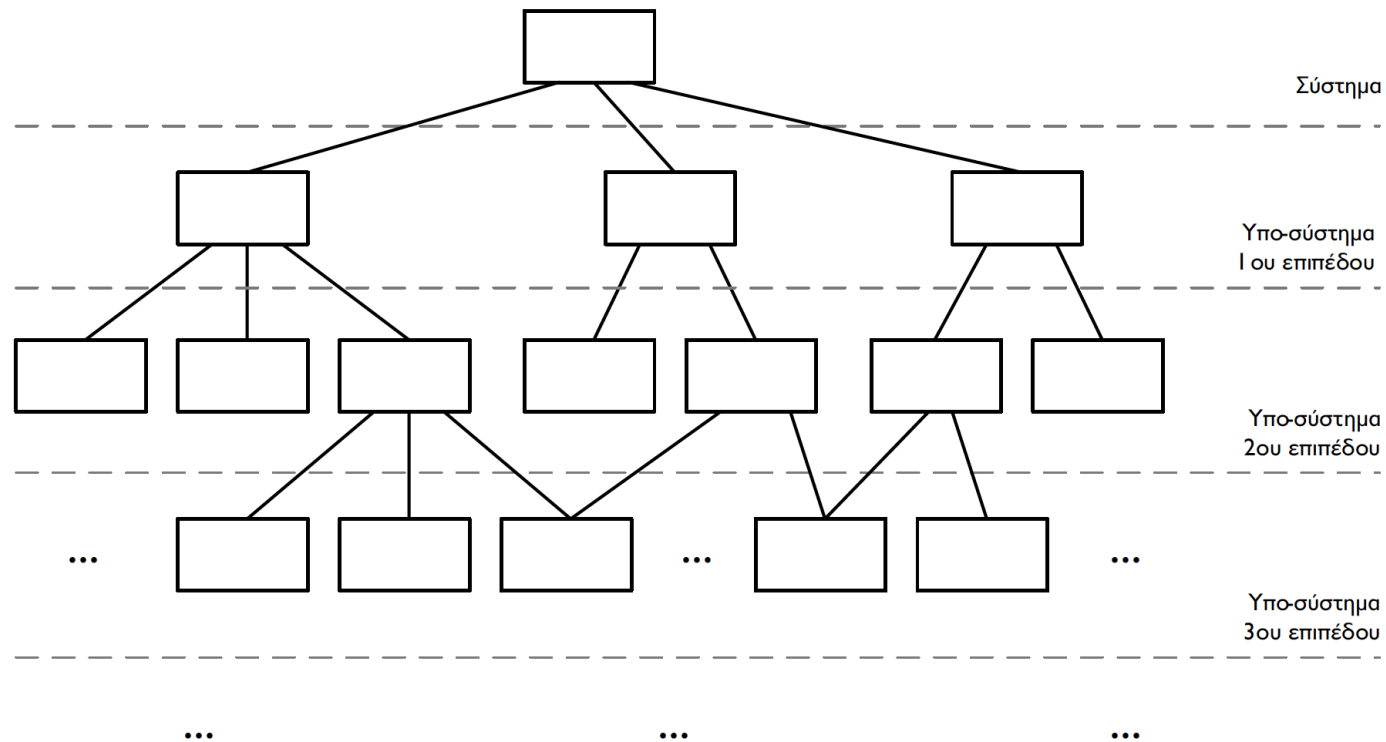


Αρχιτεκτονική σχεδίαση

Ορισμός υποσυστημάτων

Διαδοχικά επίπεδα λεπτομέρειας

Γενικό περίγραμμα της εφαρμογής



Σχεδίαση διεπαφών

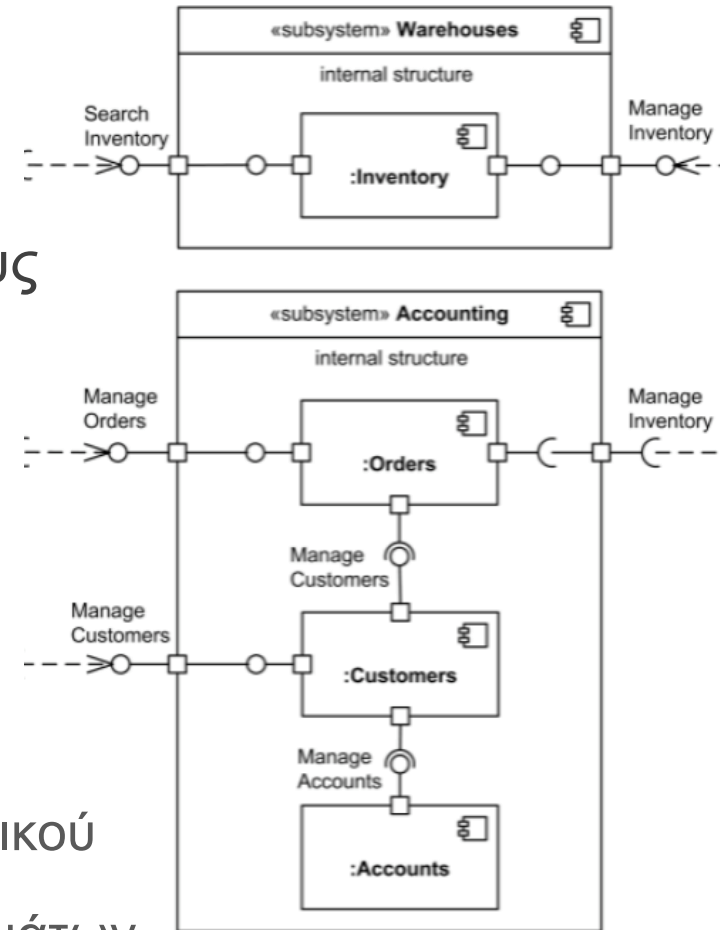
Οι μονάδες λογισμικού επικοινωνούν μεταξύ τους

Επικοινωνία σημαίνει:

- Μεταφορά του ελέγχου ροής του προγράμματος
- Μεταφορά δεδομένων με παραμέτρους

Στην εργασία αυτή καθορίζεται:

- Ο τύπος των παραμέτρων κάθε μονάδας λογισμικού
- Η φύση της επικοινωνίας μεταξύ των υποσυστημάτων
- Οι λεπτομέρειες της επικοινωνίας με εξωτερικές συσκευές κλπ
- Η επικοινωνία με τον χρήστη



Λεπτομερής σχεδίαση μονάδων

Εστίαση στο εσωτερικό κάθε μονάδας λογισμικού

Περιγραφή με τρόπο ώστε ο προγραμματιστής να μπορεί να κατασκευάσει «κατευθείαν» κάθε μονάδα

Λαμβάνονται υπόψη:

- Όλα τα μέχρι το σημείο αυτό προϊόντα της σχεδίασης
- Όλα τα σημεία των προδιαγραφών, τα οποία περιγράφουν τη συμπεριφορά του λογισμικού

Σχεδίαση δεδομένων

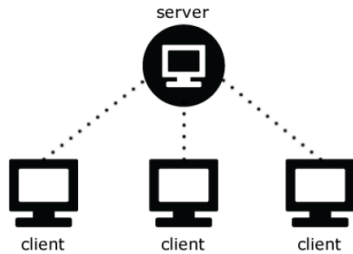
Αντικείμενο της γνωστικής περιοχής «Βάσεις Δεδομένων»

- Επαλήθευση του μοντέλου οντοτήτων – συσχετίσεων
- Βελτιστοποιήσεις και κανονικοποίηση του σχήματος δεδομένων
- Καθορισμός των τύπων των πεδίων κάθε πίνακα (φυσικό επίπεδο)
- Καθορισμός δεικτών και όψεων (views) (λογικό επίπεδο)

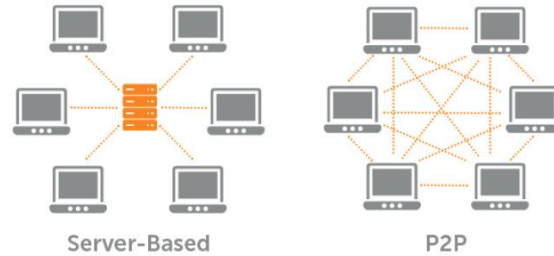
- Η σχεδίαση δεδομένων στην πράξη δεν είναι ανεξάρτητη από το περιβάλλον υλοποίησης, δηλαδή το DBMS που θα χρησιμοποιηθεί

Σχεδίαση και αρχιτεκτονικά πρότυπα λογισμικού

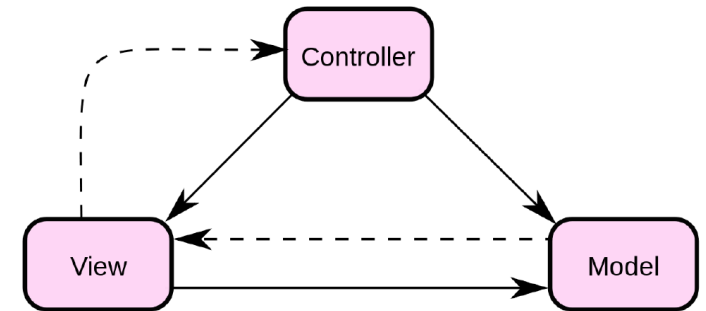
Client-Server



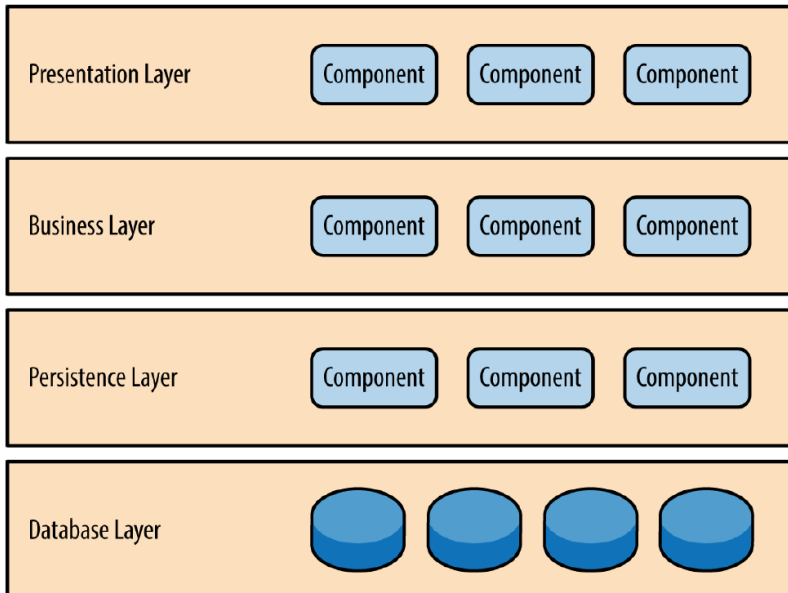
Peer-to-peer (P2P)



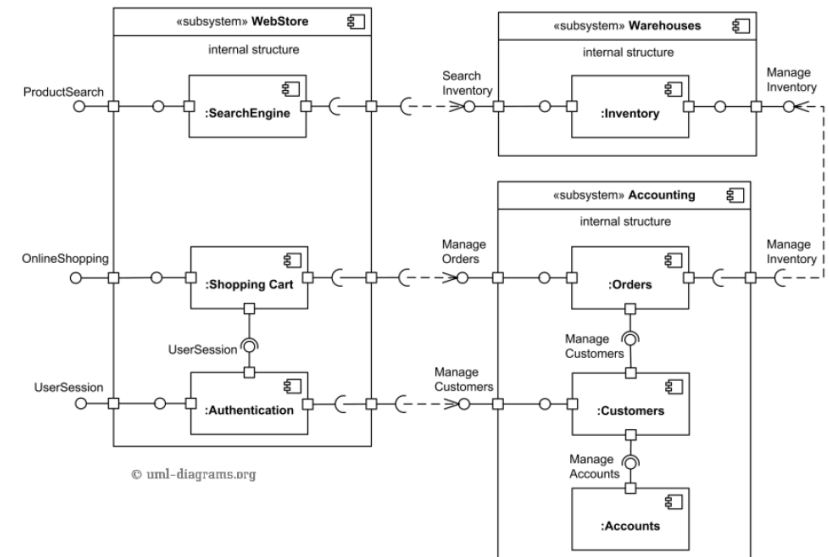
Model-View-Controller (MVC)



Layered/N-tier

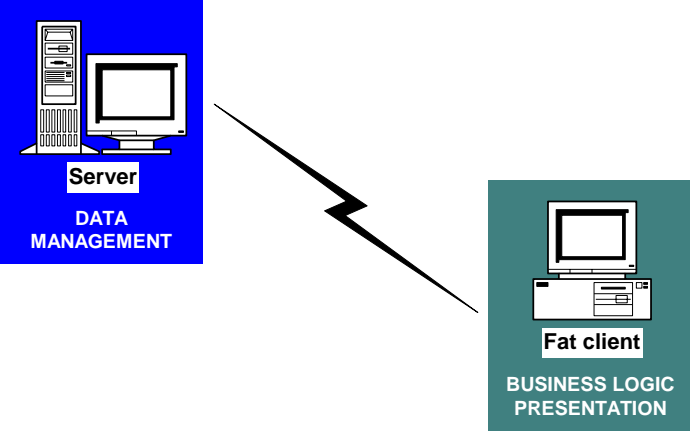


Component-based

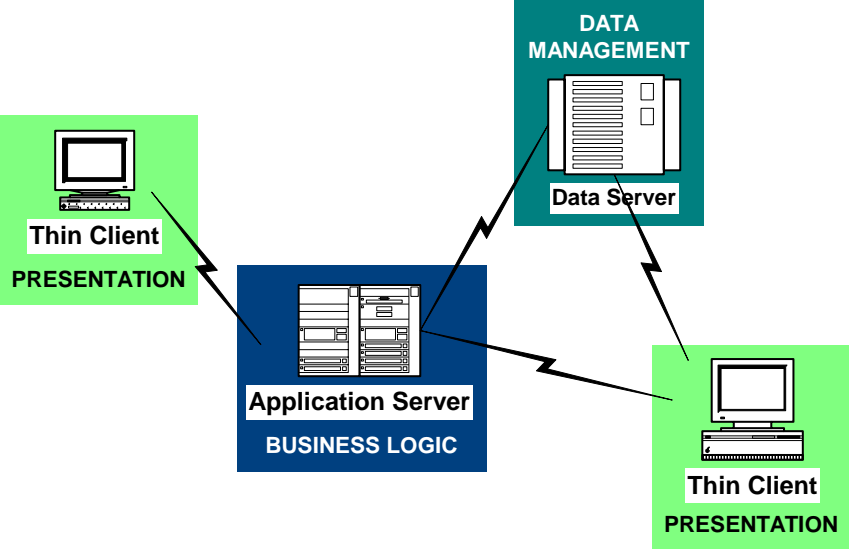


Client-server variants

"Fat" clients

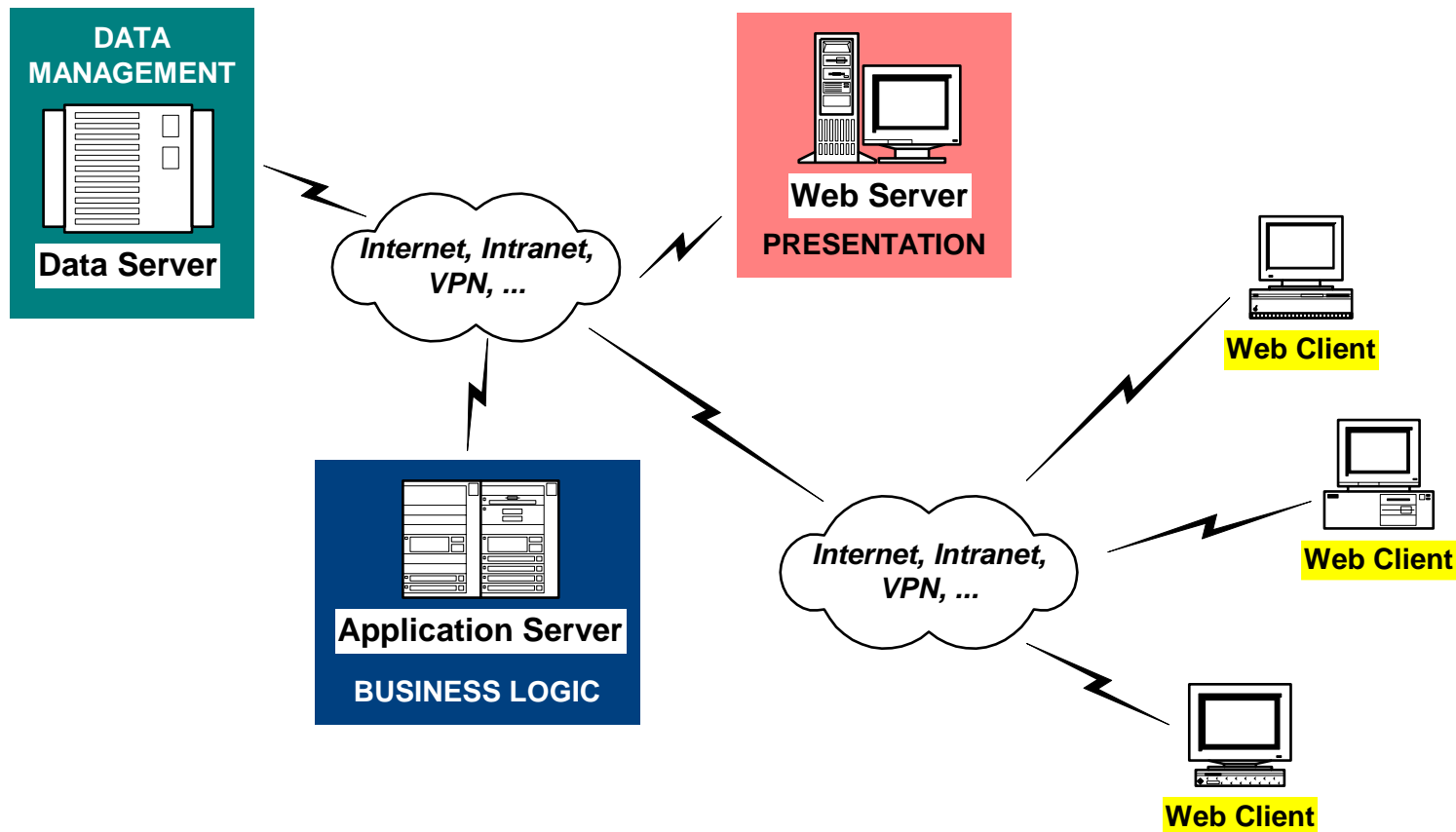


"Thin" clients

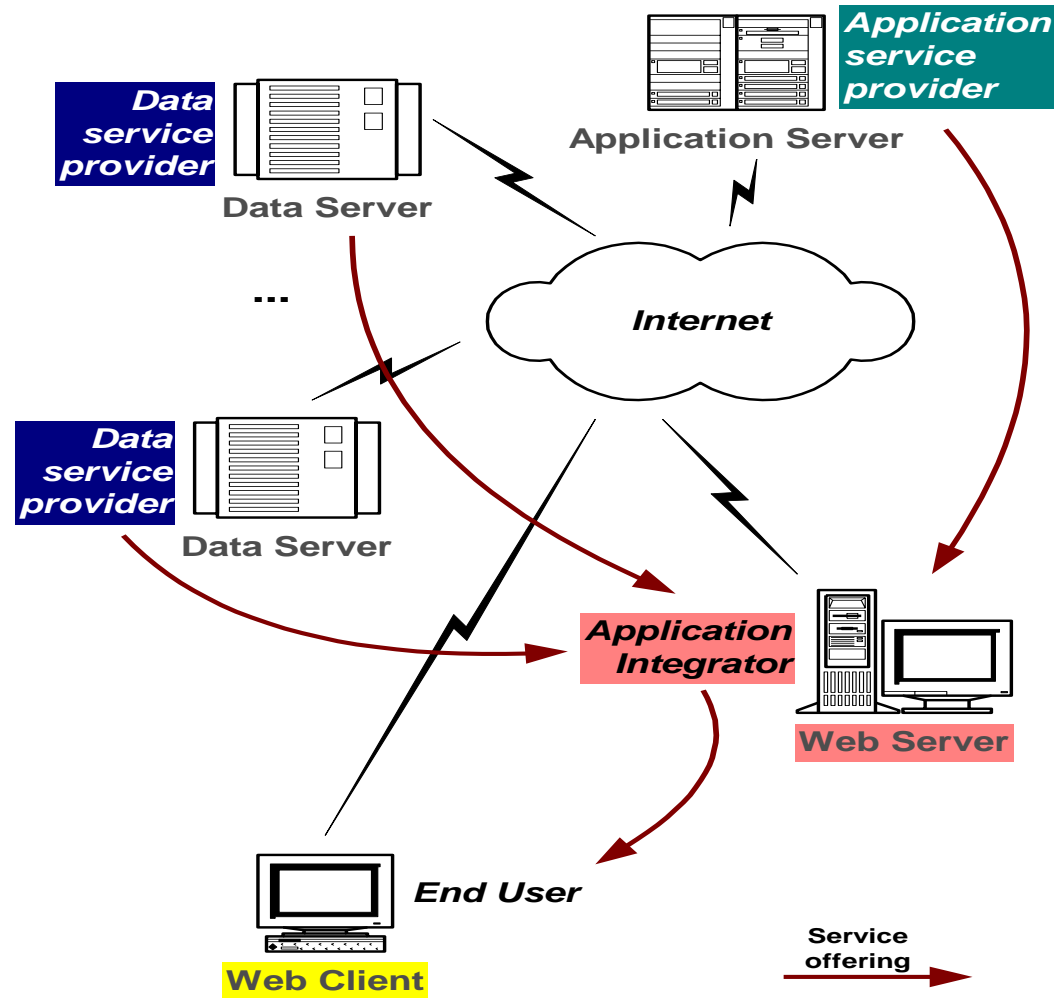


Multi-tier, web-based

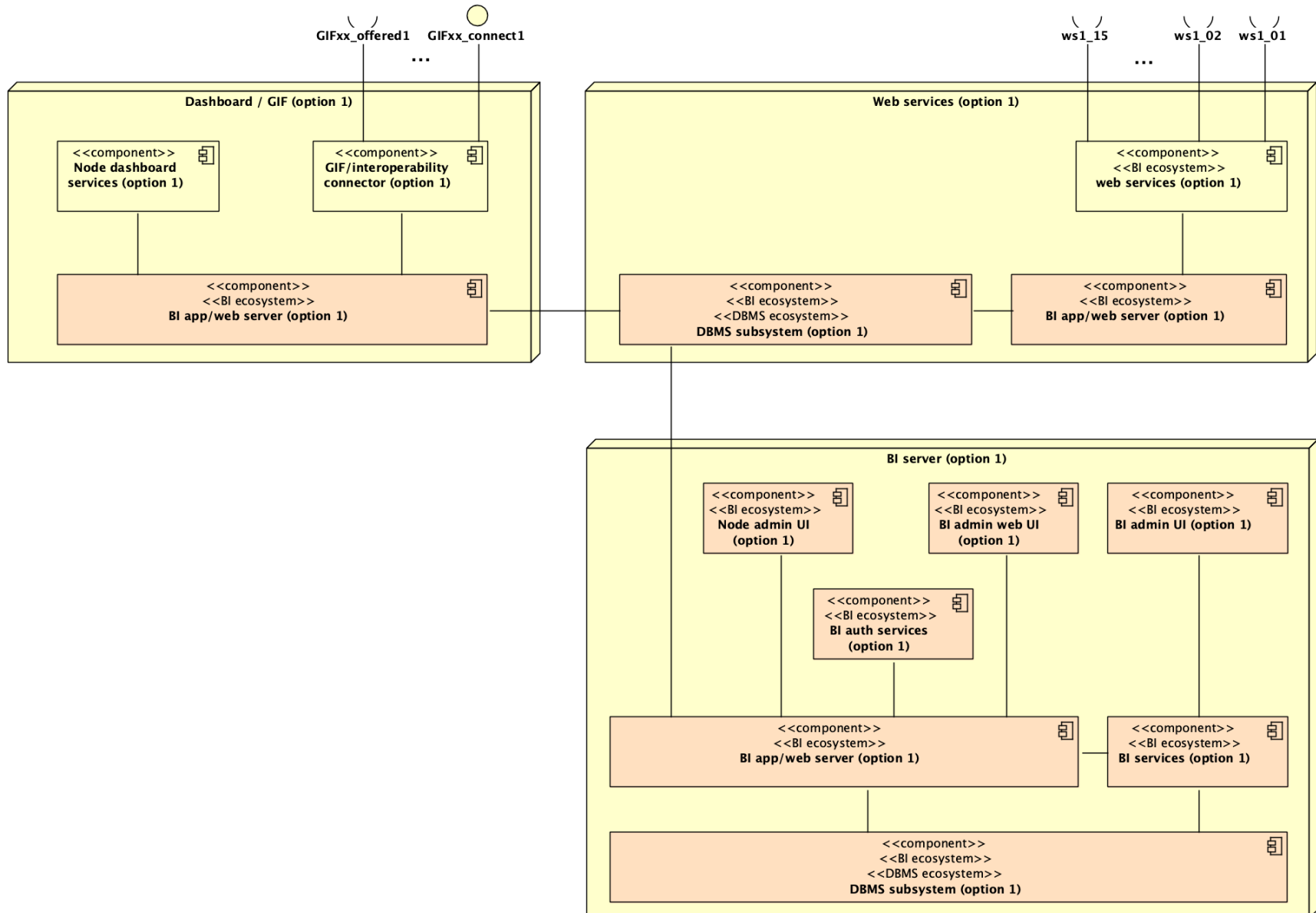
Πολυμερής κατανομημένη διάταξη με web clients



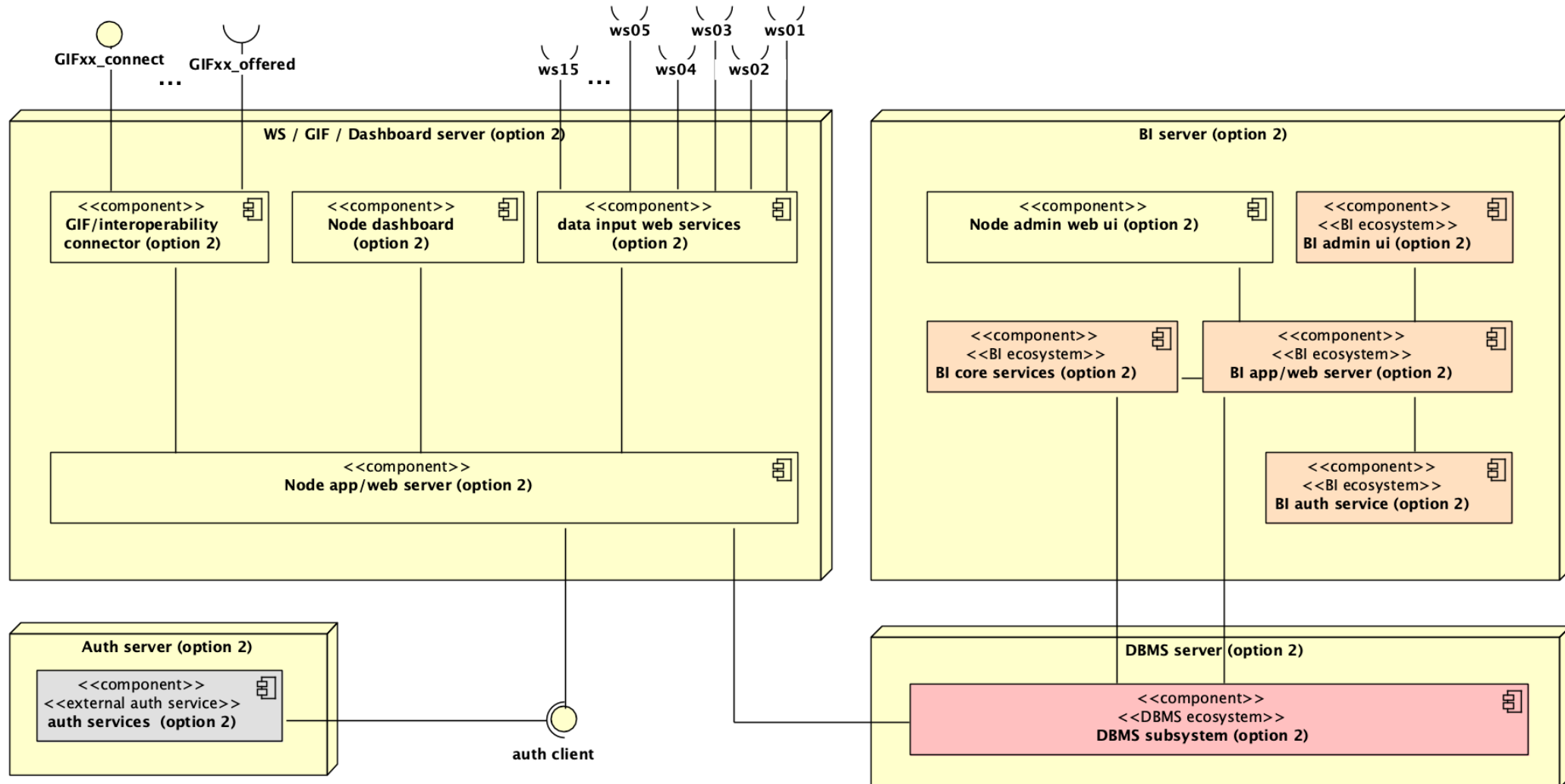
Multi-tier, web-based, service-oriented



Παράδειγμα από πραγματικό σύστημα



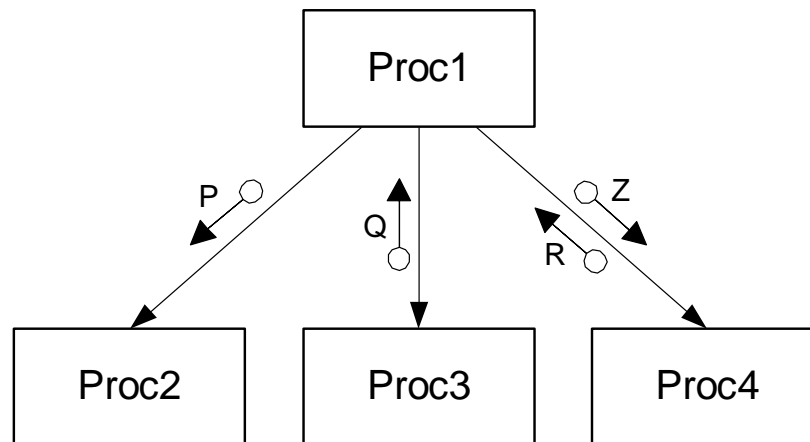
Παράδειγμα από πραγματικό σύστημα



Αρχιτεκτονική – δομημένη σχεδίαση

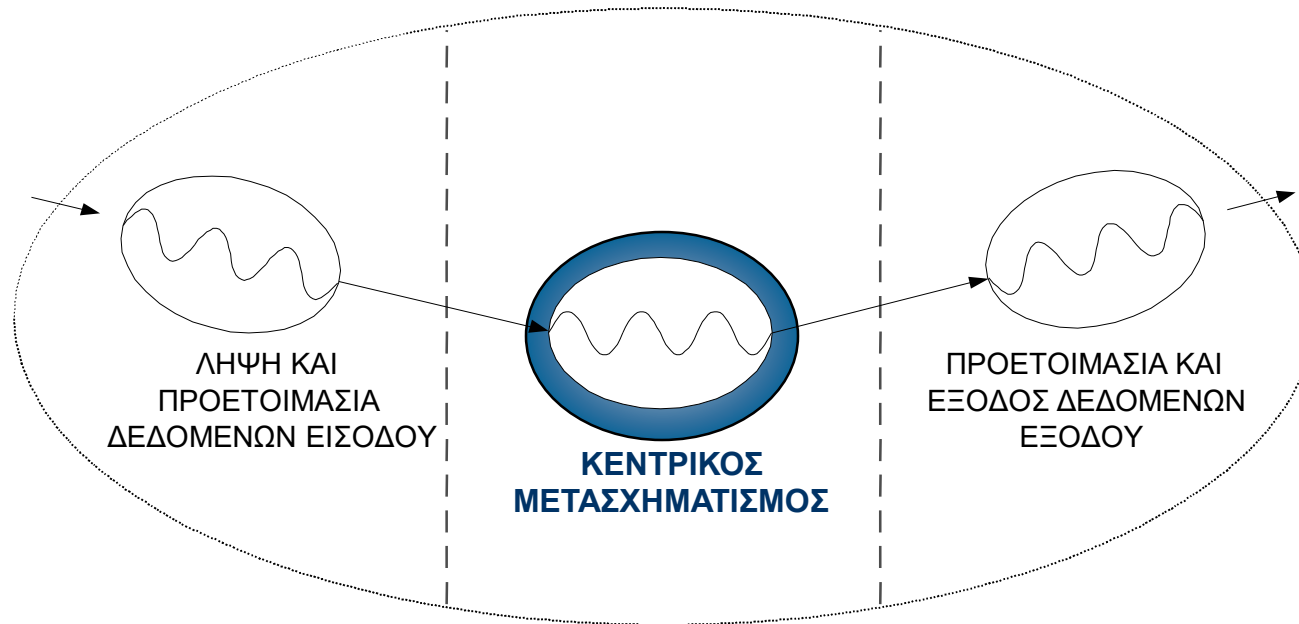
Από το Διάγραμμα Ροής Δεδομένων,
στο **Διάγραμμα Δομής Προγράμματος**

Συμβολισμοί



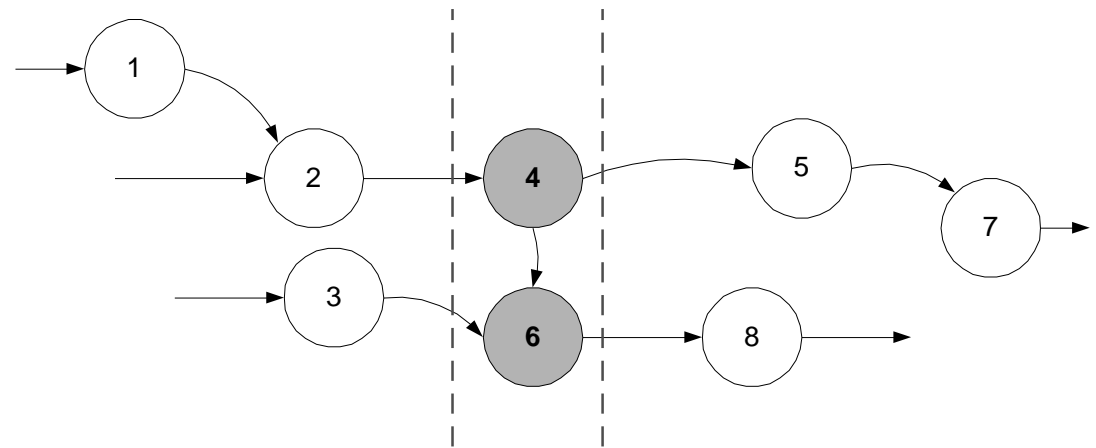
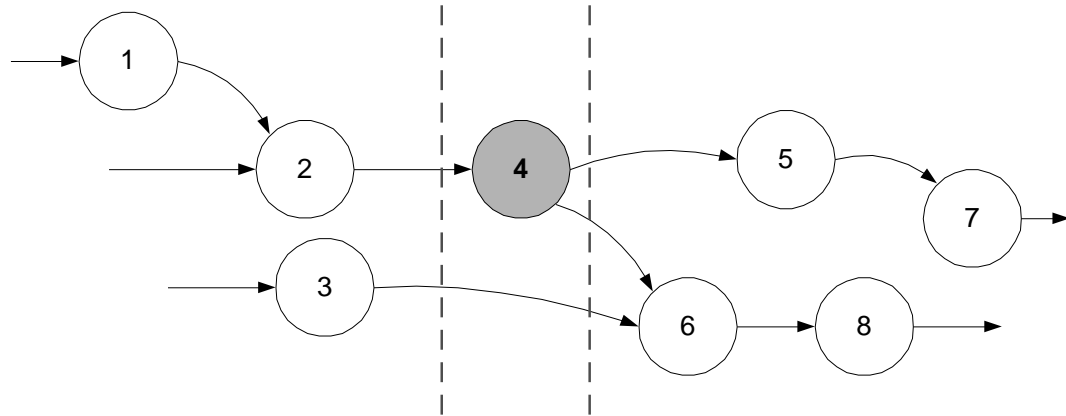
Αρχιτεκτονική – δομημένη σχεδίαση

Η έννοια του **κεντρικού μετασχηματισμού**
σε Διάγραμμα Ροής Δεδομένων



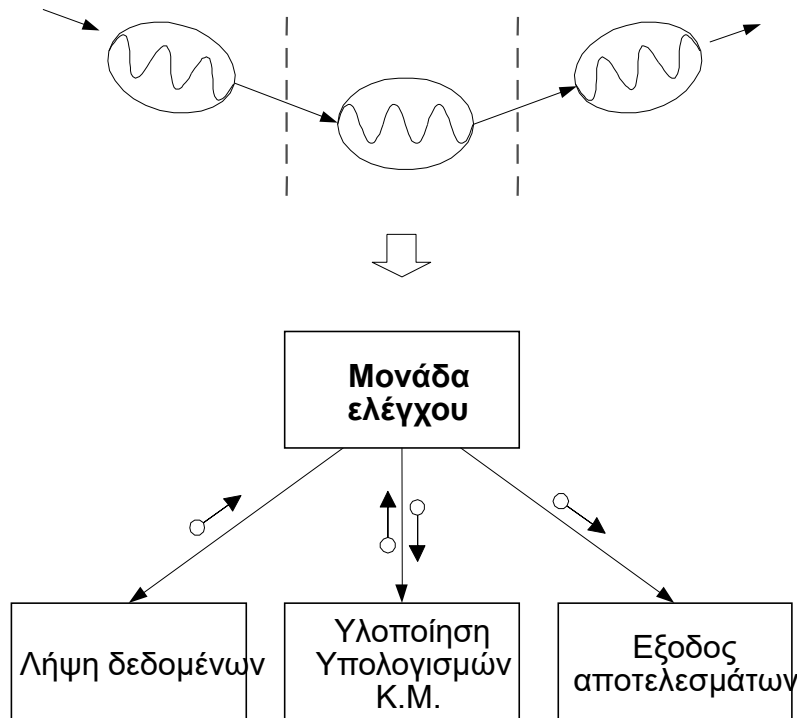
Αρχιτεκτονική – δομημένη σχεδίαση

Παραδείγματα κεντρικού μετασχηματισμού



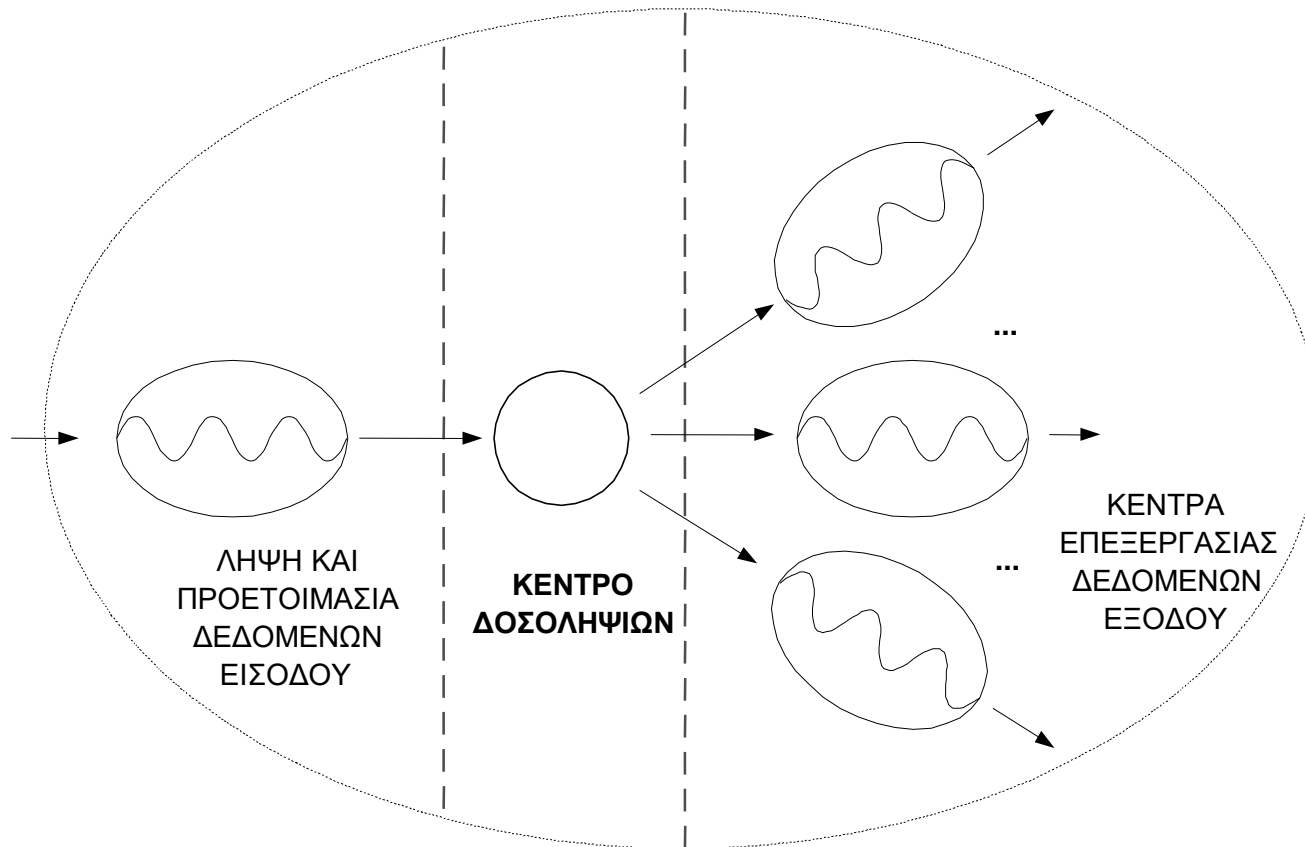
Αρχιτεκτονική – δομημένη σχεδίαση

Απεικόνιση κεντρικού μετασχηματισμού
σε **διάγραμμα δομής προγράμματος** (call graph)



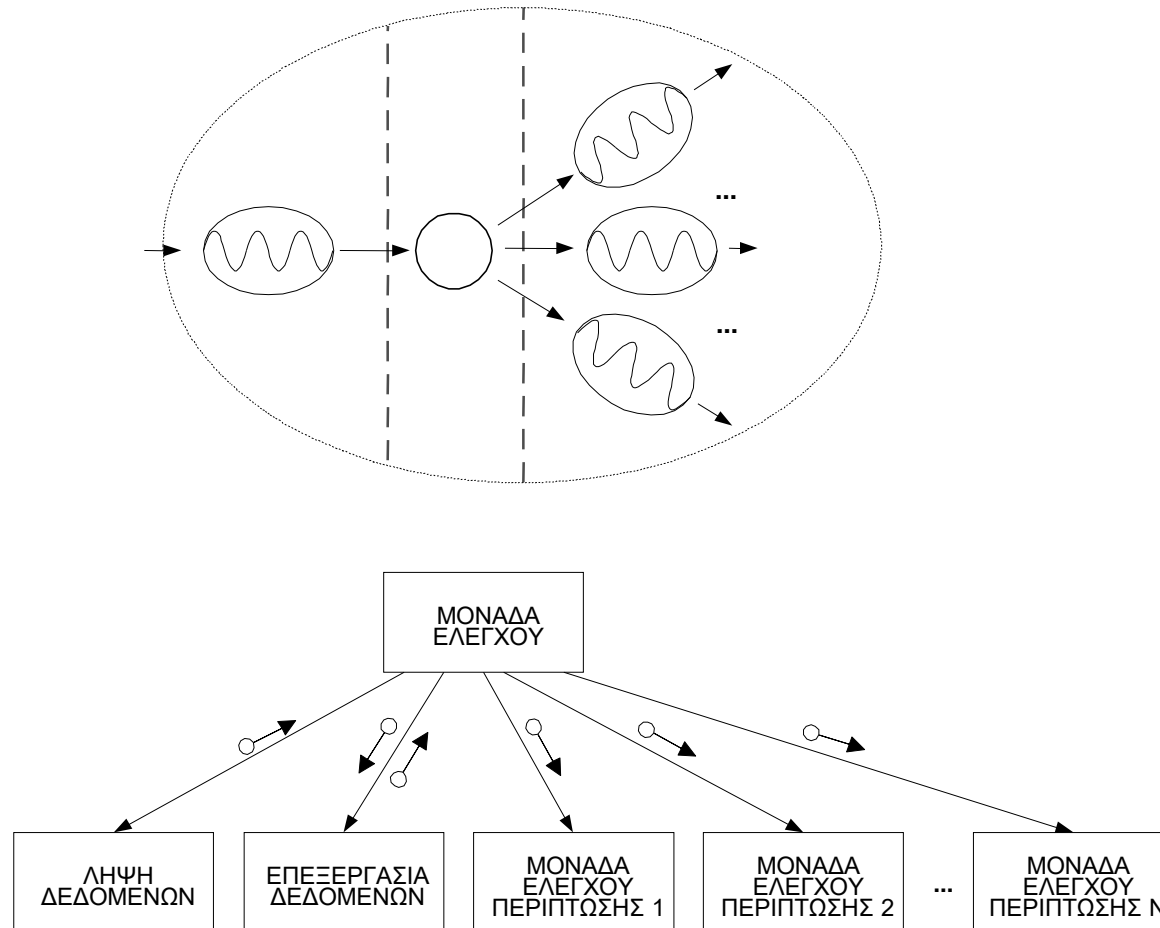
Αρχιτεκτονική – δομημένη σχεδίαση

Η έννοια του κέντρου δοσοληψιών (transactions center) σε Διάγραμμα Ροής Δεδομένων



Αρχιτεκτονική – δομημένη σχεδίαση

Απεικόνιση κέντρου δοσοληψιών σε διάγραμμα δομής προγράμματος



Αρχιτεκτονική – δομημένη σχεδίαση

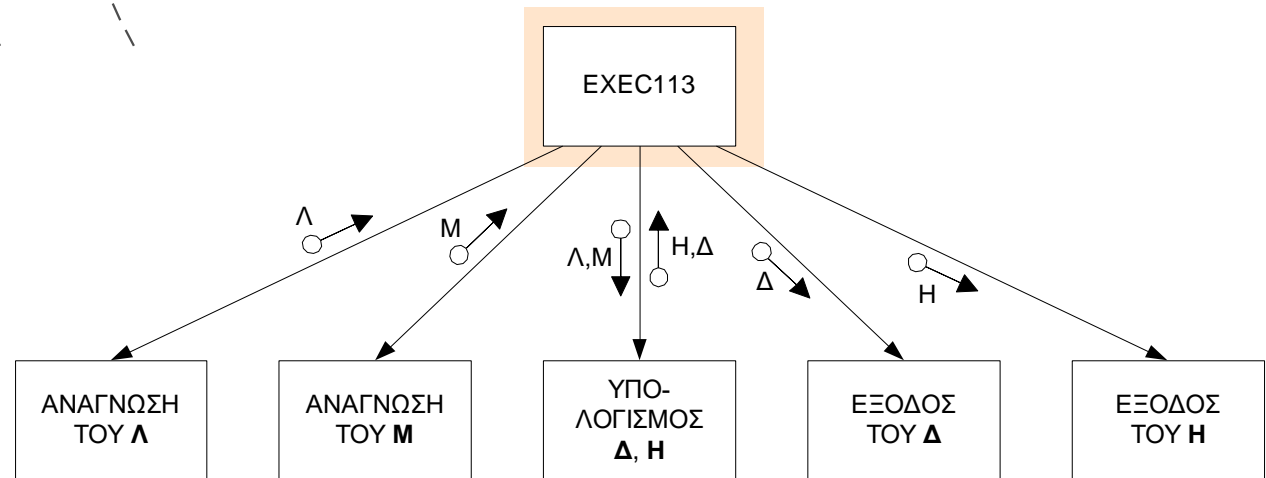
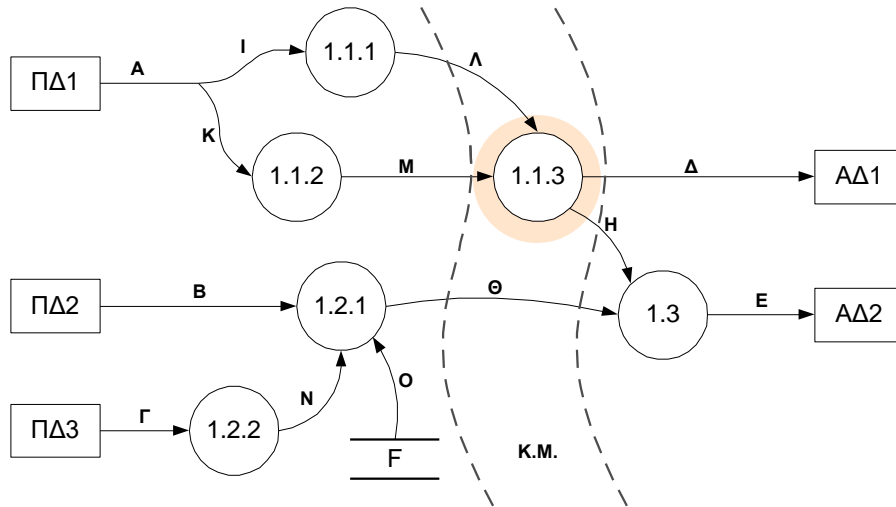
Βήματα κατασκευής διαγραμμάτων δομής

- Εντοπισμός ενός κεντρικού μετασχηματισμού ή κέντρου δοσοληψιών
- Απεικόνιση σε διάγραμμα δομής προγράμματος
- Παραγοντοποίηση:
επανάληψη για το δεξί και το αριστερό τμήμα του ΚΜ / ΚΔ
- Συνένωση τμημάτων διαγραμμάτων δομής που προκύπτουν

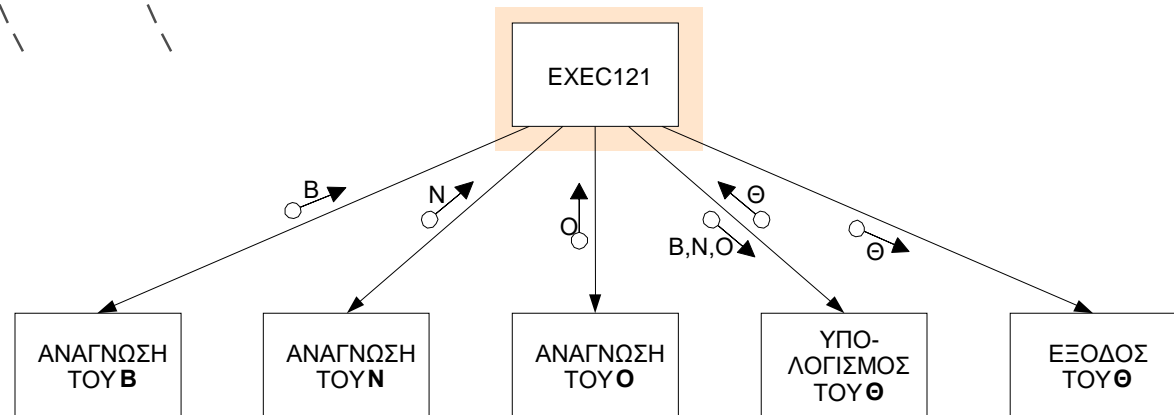
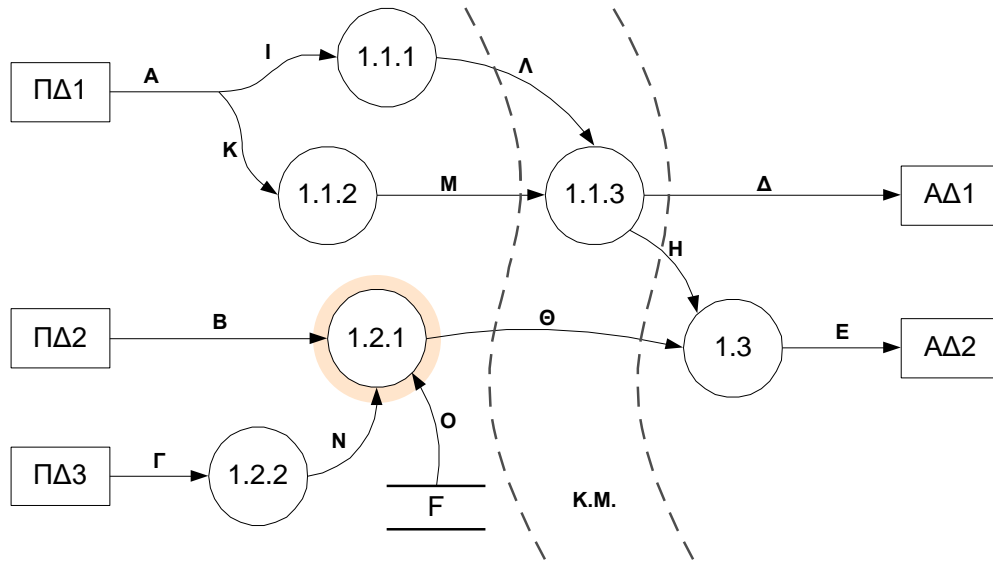
Αποτέλεσμα:

Ένα διάγραμμα δομής του λογισμικού με λεπτομέρεια ανάλογη της λεπτομέρειας του διαγράμματος ροής δεδομένων από το οποίο κατασκευάστηκε

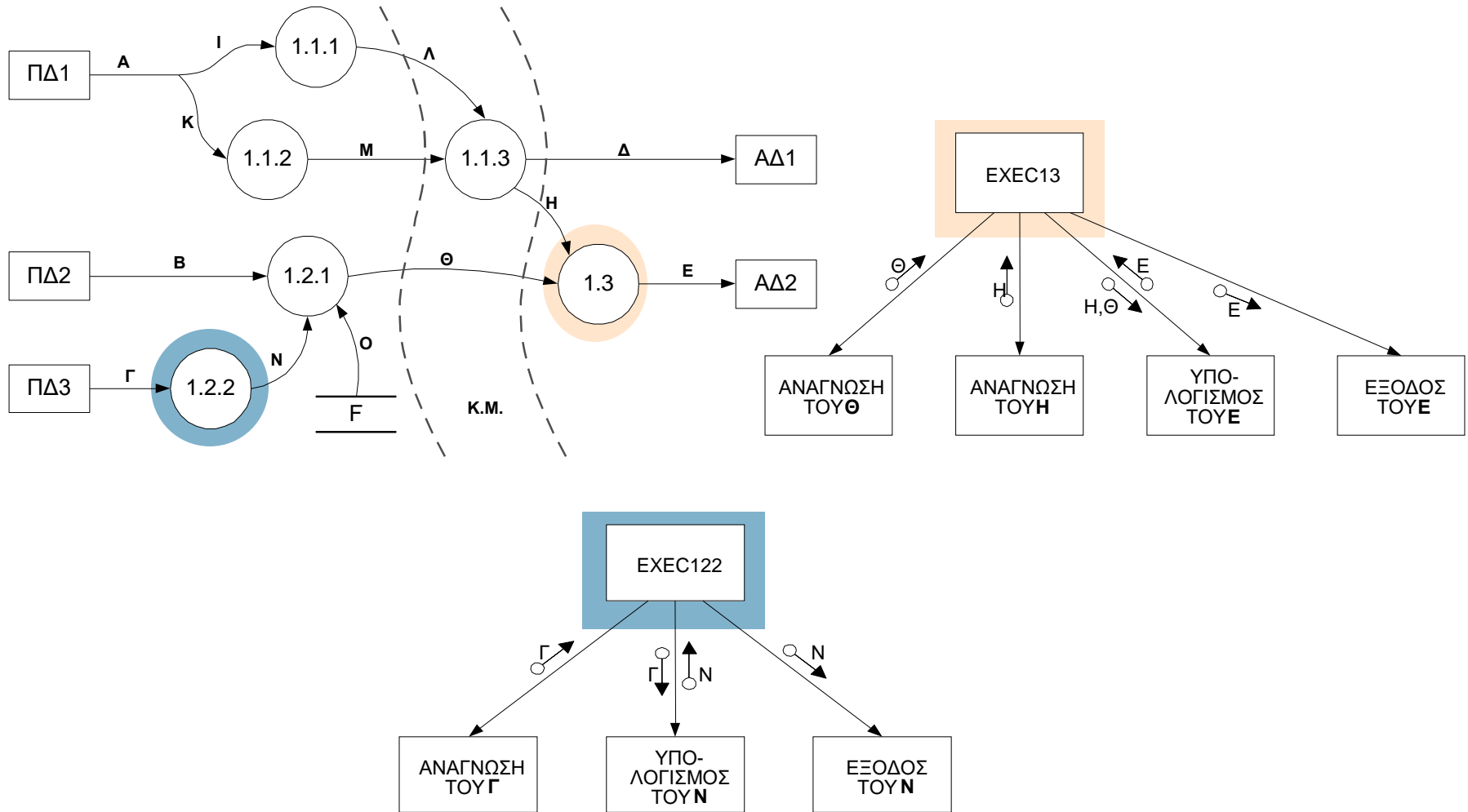
Παράδειγμα αρχιτεκτονικής στη δομημένη σχεδίαση



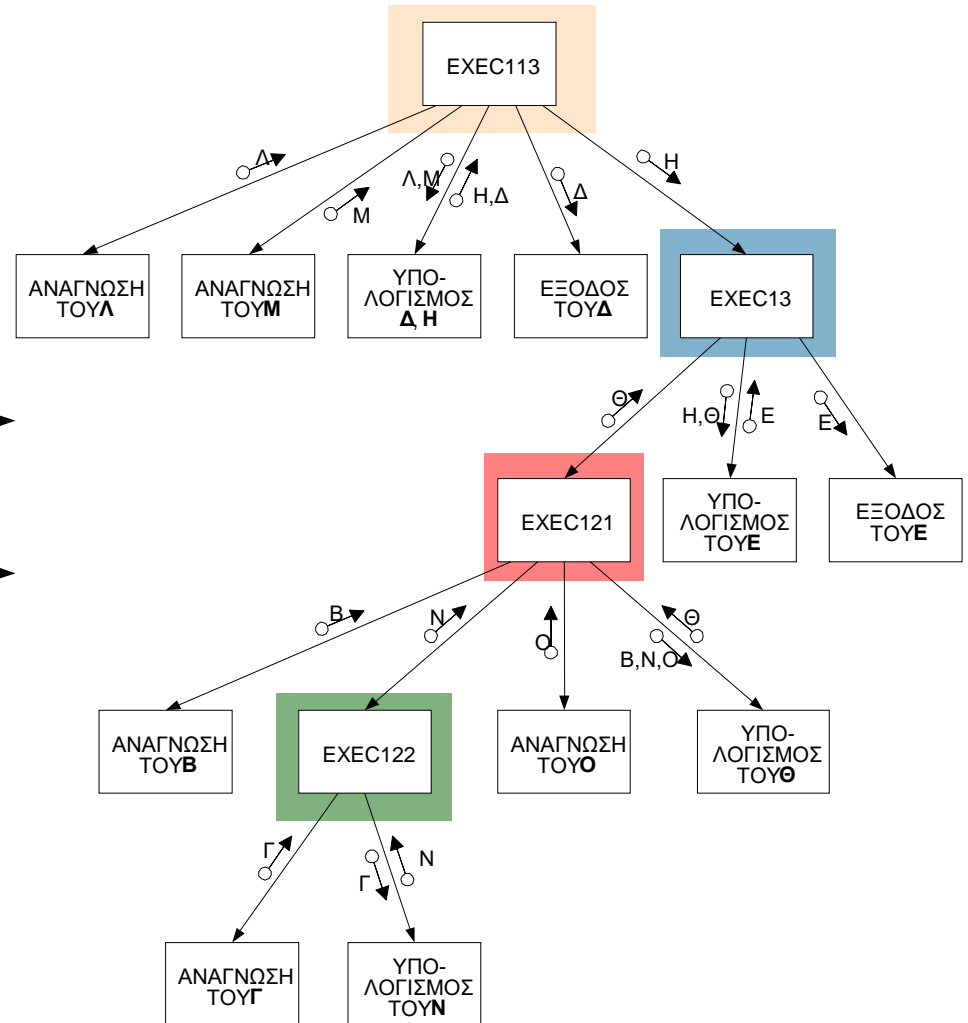
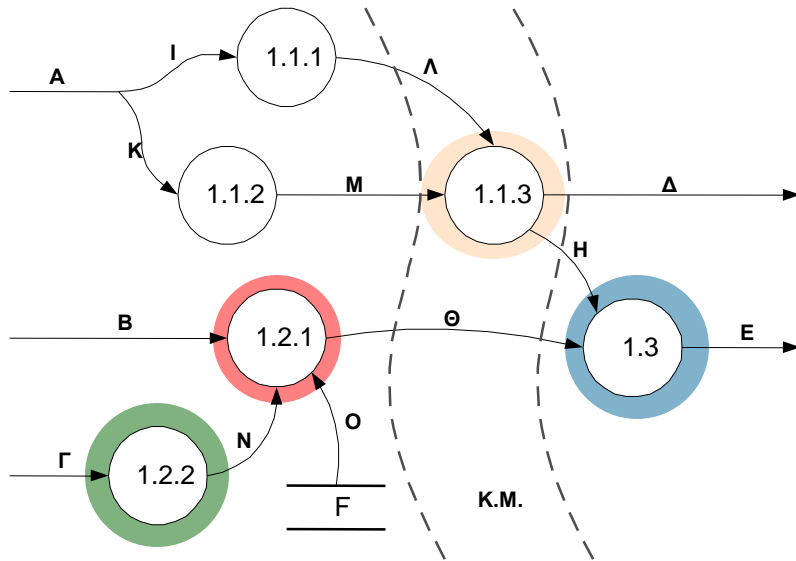
Παράδειγμα αρχιτεκτονικής στη δομημένη σχεδίαση



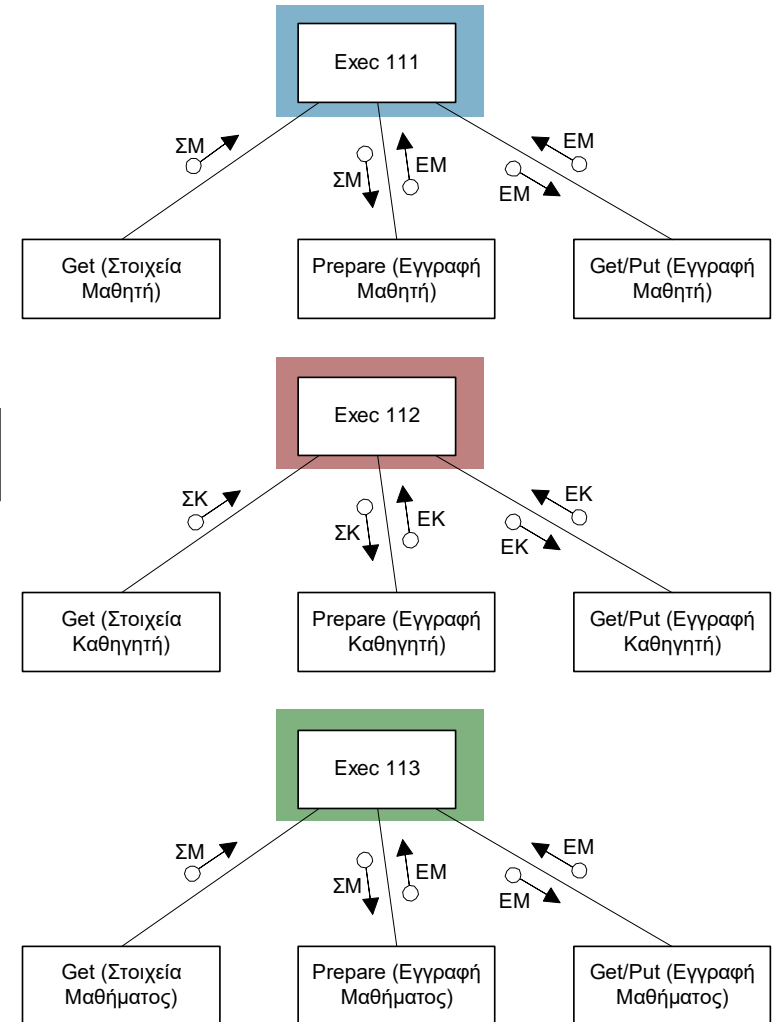
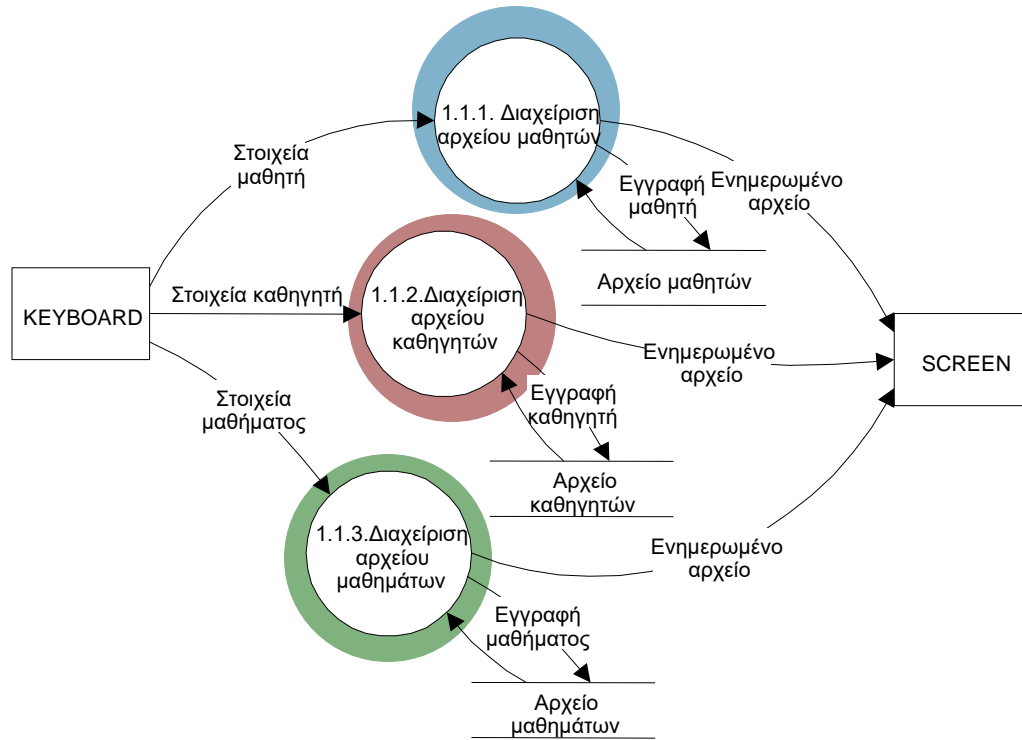
Παράδειγμα αρχιτεκτονικής στη δομημένη σχεδίαση



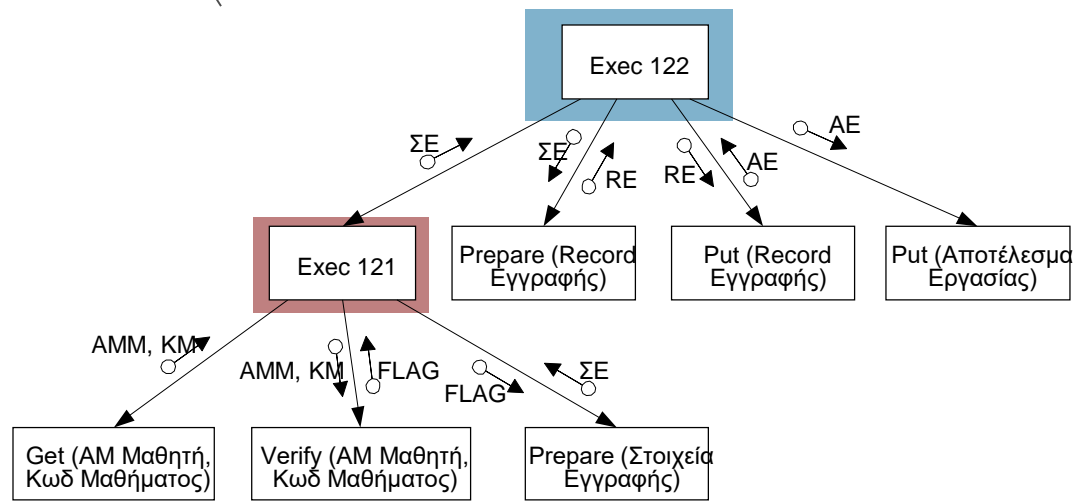
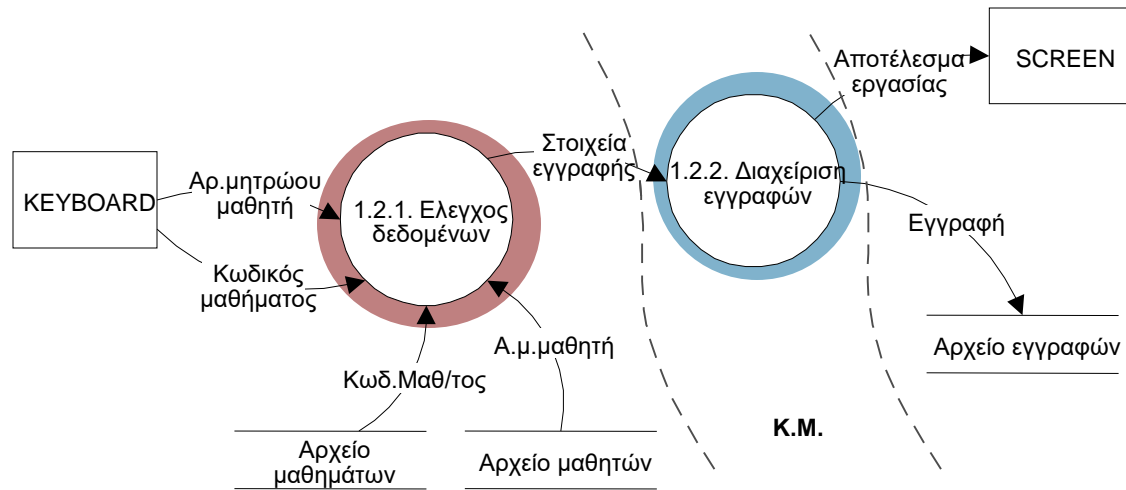
Παράδειγμα αρχιτεκτονικής στη δομημένη σχεδίαση



Μελέτη περίπτωσης



Μελέτη περίπτωσης



Λεπτομερής σχεδίαση μονάδων

Ψευδοκώδικας (old school, sort of)

Απλές εκφράσεις	Επαναληπτική εκτέλεση
<p><i>/*σχόλιο */</i> μεταβλητή := τιμή <i>/* ανάθεση */</i> φραστική περιγραφή ενέργειας + - * / ^ <i>/* μαθηματικές εκφράσεις */</i></p>	<p>FOR μτβλ FROM τιμή1 TO τιμή2 STEP τιμή3 DO (ενέργειες) END_FOR</p>
Εκτέλεση με επιλογή περίπτωσης	Εκτέλεση υπό συνθήκη
<p>CASE έκφραση OF (τιμή 1) : (ενέργειες) (τιμή 2) : (ενέργειες) ... (τιμή N) : (ενέργειες) OTHERWISE (εντολές αν η έκφραση έχει άλλη τιμή) END_CASE</p>	<p>IF συνθήκη THEN (ενέργειες αν η συνθήκη είναι αληθής) ELSE (εντολές αν η συνθήκη είναι ψευδής) END_IF</p>

Λεπτομερής σχεδίαση μονάδων

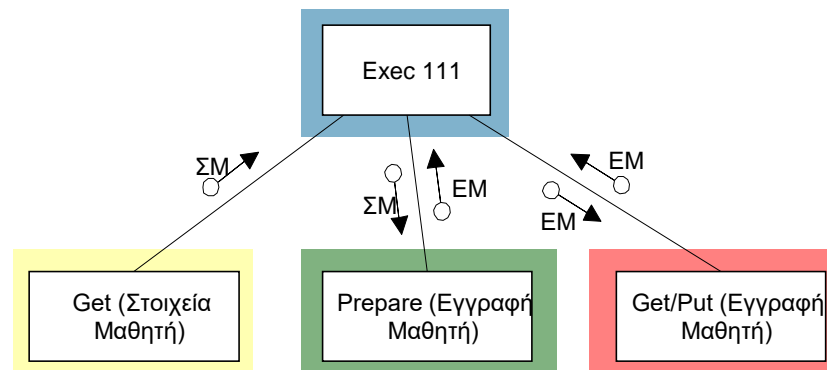
Ψευδοκώδικας

Επαναληπτική εκτέλεση με συνθήκη (1)	Επαναληπτική εκτέλεση με συνθήκη (2)
REPEAT (ενέργειες) UNTIL συνθήκη	WHILE συνθήκη DO (ενέργειες) END_WHILE
Ορισμός διαδικασιών	Ορισμός συναρτήσεων
PROCEDURE όνομα (παράμετρος :IN/OUT, ...) GLOBAL VAR όνομα1, όνομα2, ... LOCAL VAR όνομα1, όνομα2, (ενέργειες) ... CALL όνομα_διαδικασίας (παράμ1, παράμ2, ...) ... (ενέργειες) ... END_PROCEDURE	FUNCTION όνομα_συνάρτησης (παράμετρος, ...) GLOBAL VAR όνομα1, όνομα2, ... LOCAL VAR όνομα1, όνομα2, (ενέργειες) ... όνομα_συνάρτησης := τιμή ... (ενέργειες) ... END_FUNCTION

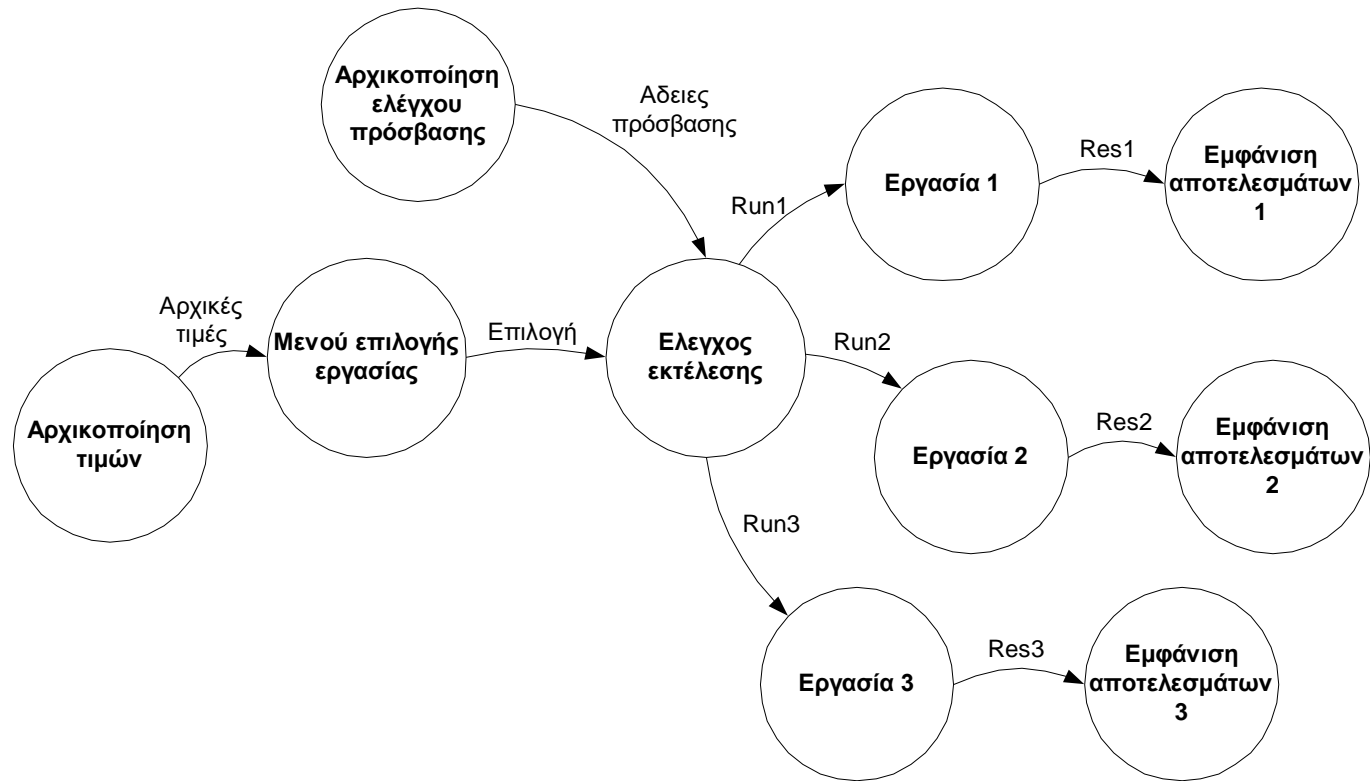
Λεπτομερής σχεδίαση μονάδων

Παράδειγμα ψευδοκώδικα

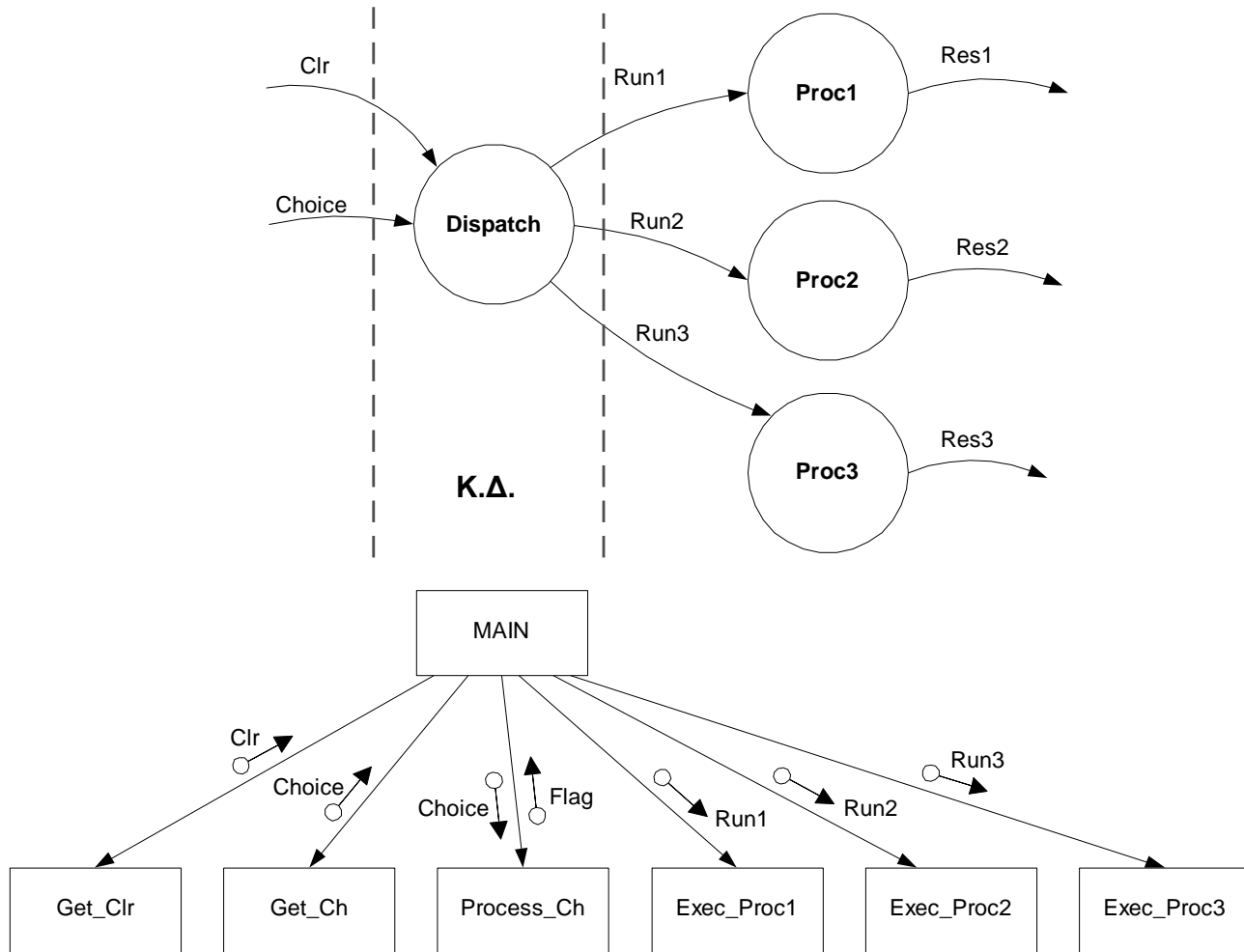
```
/*-----*/  
PROCEDURE Exec111  
/*-----*/  
LOCAL VAR στοιχεία_μαθητή, εγγραφή_μαθητή  
Αρχικοποίησε στοιχεία_μαθητή, εγγραφή_μαθητή  
  
WHILE στοιχεία_μαθητή <> κενό DO  
  
    CALL Get_ΣΜ(στοιχεία_μαθητή)  
    IF στοιχεία_μαθητή <> κενό THEN  
        CALL Prepare_ΣΜ(στοιχεία_μαθητή, εγγραφή_μαθητή)  
        CALL Put_EM(εγγραφή_μαθητή)  
    END_IF  
  
END_WHILE  
  
END_PROCEDURE
```



Παράδειγμα αρχιτεκτονικής στη δομημένη σχεδίαση



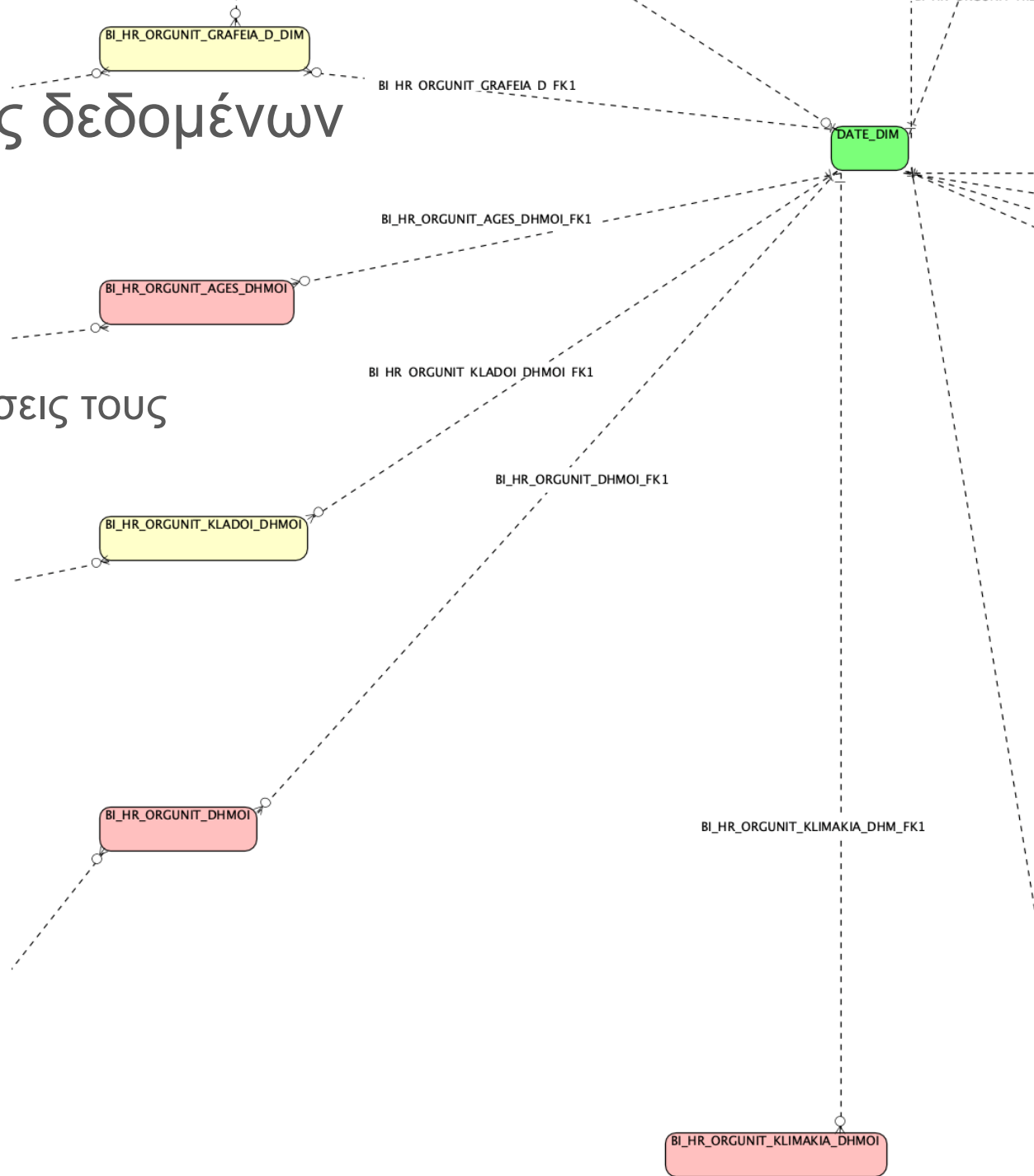
Παράδειγμα αρχιτεκτονικής στη δομημένη σχεδίαση



Παράδειγμα σχεδίασης δεδομένων

Στη φάση της ανάλυσης

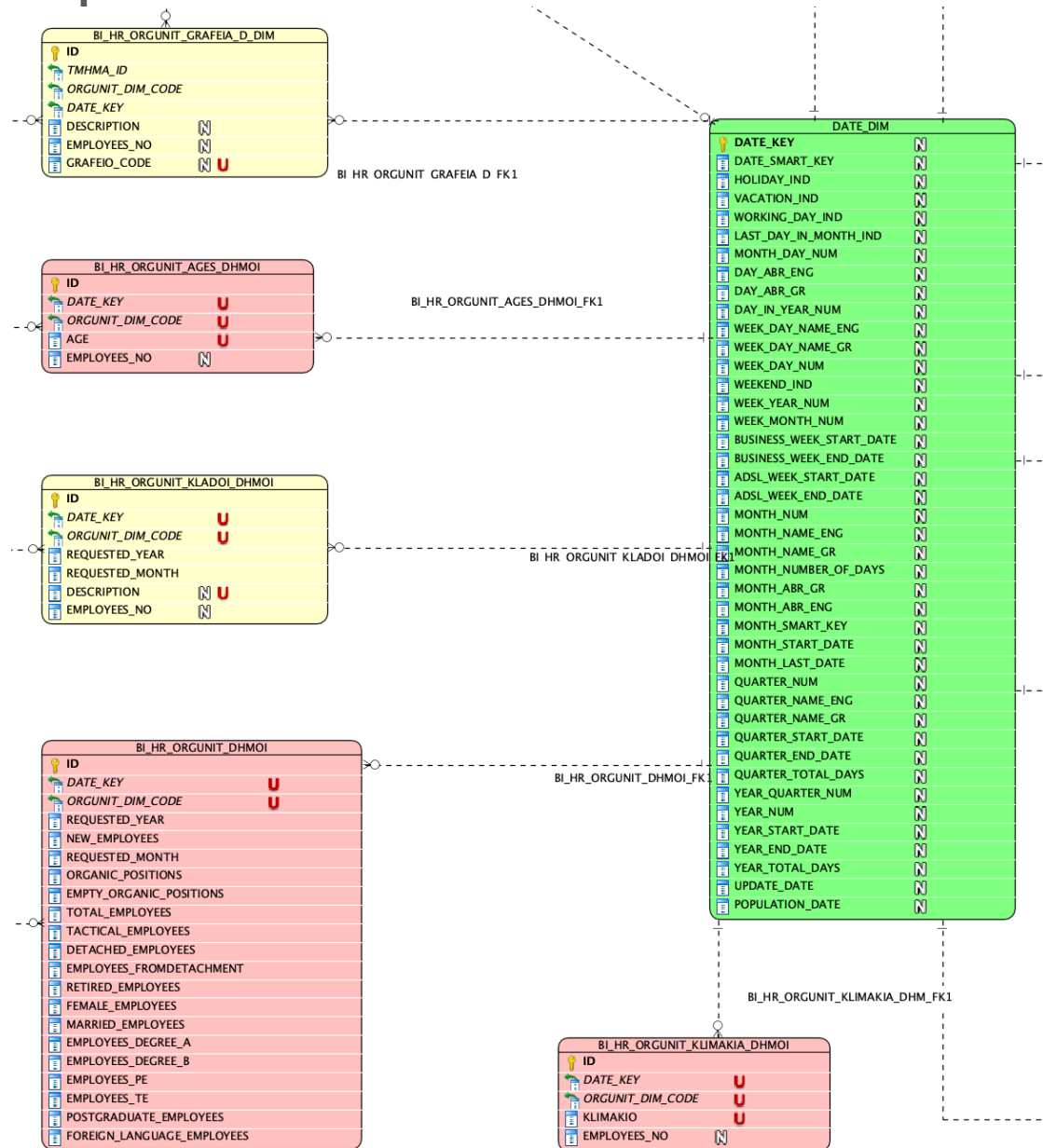
- Εντοπίζονται μόνο οι οντότητες και οι συσχετίσεις τους
- Ενδέχεται να μεταβληθούν στην πορεία



Παράδειγμα σχεδίασης δεδομένων

Στα αρχικά στάδια της σχεδίασης

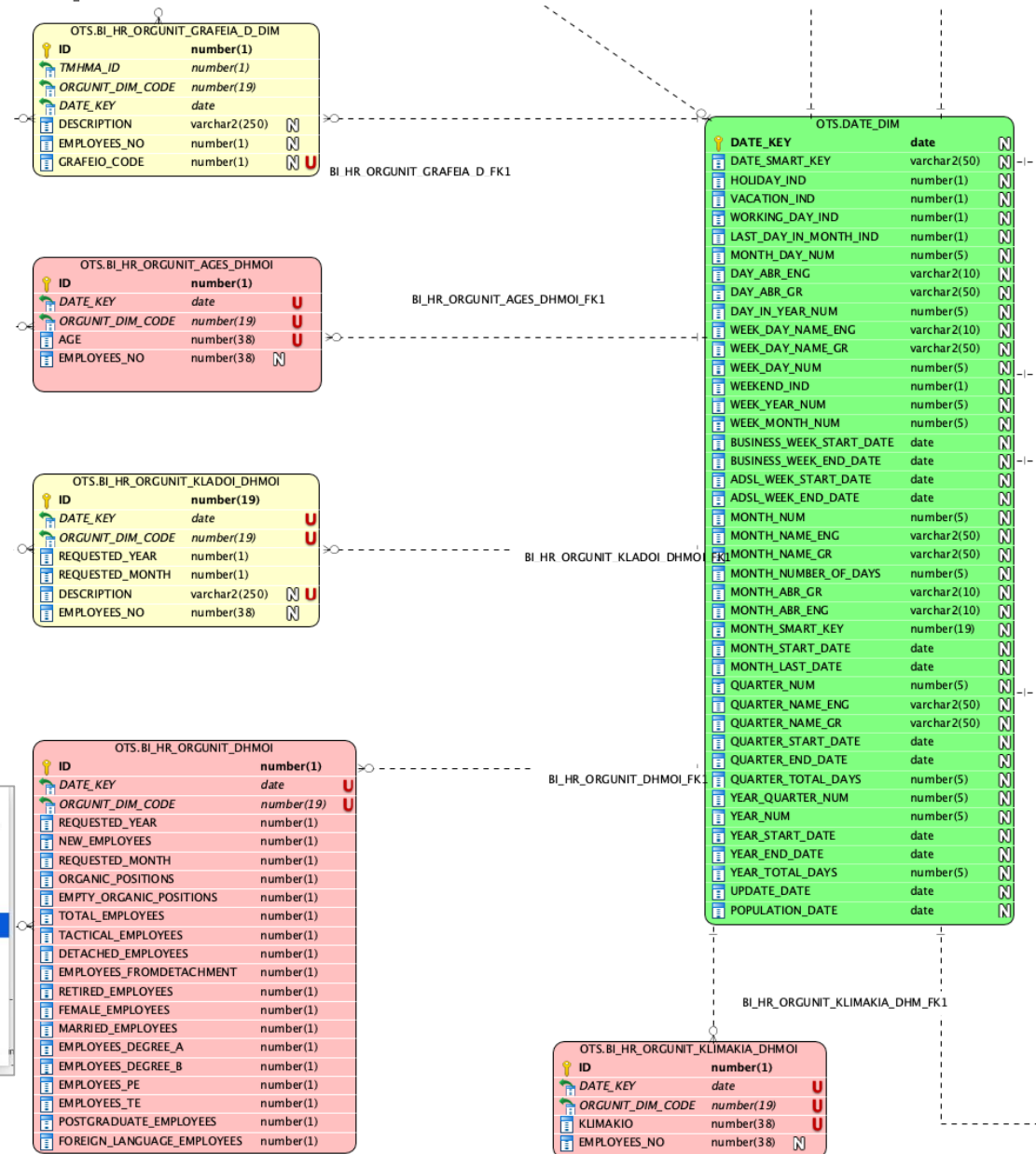
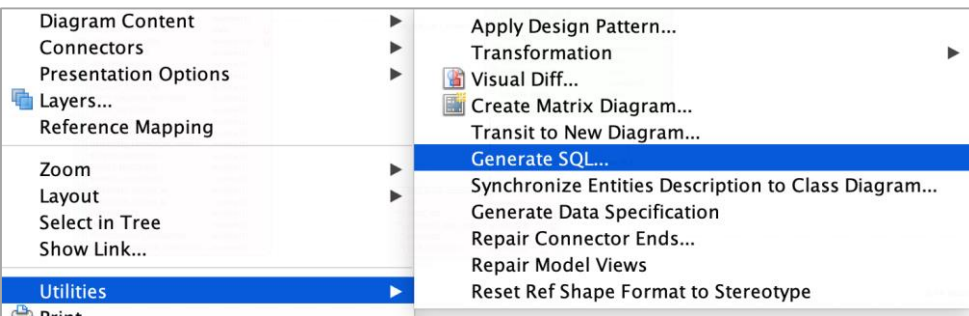
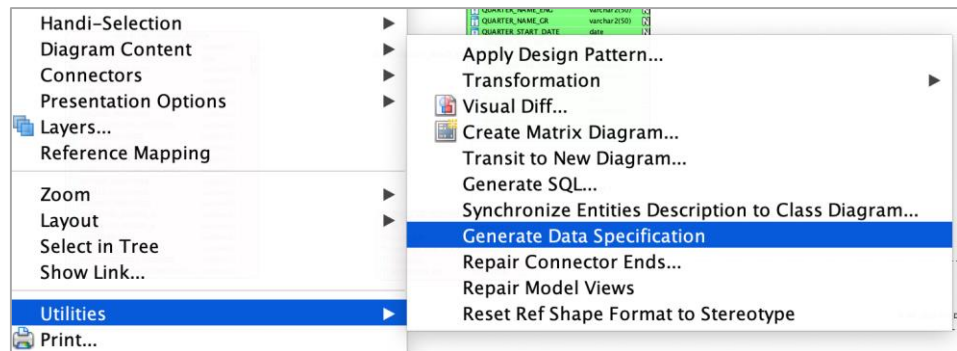
- Αποδίδονται ιδιώματα (πεδία) στις οντότητες
- Δεν απασχολούν οι κατασκευαστικές λεπτομέρειες



Παράδειγμα σχεδίασης δεδομένων

Ολοκληρωμένη σχεδίαση

- Περιέχεται πλήρης κατασκευαστική πληροφορία (τύποι, κλπ)



Παράδειγμα σχεδίασης δεδομένων

Ολοκληρωμένη σχεδίαση

■ Αυτόματη τεκμηρίωση

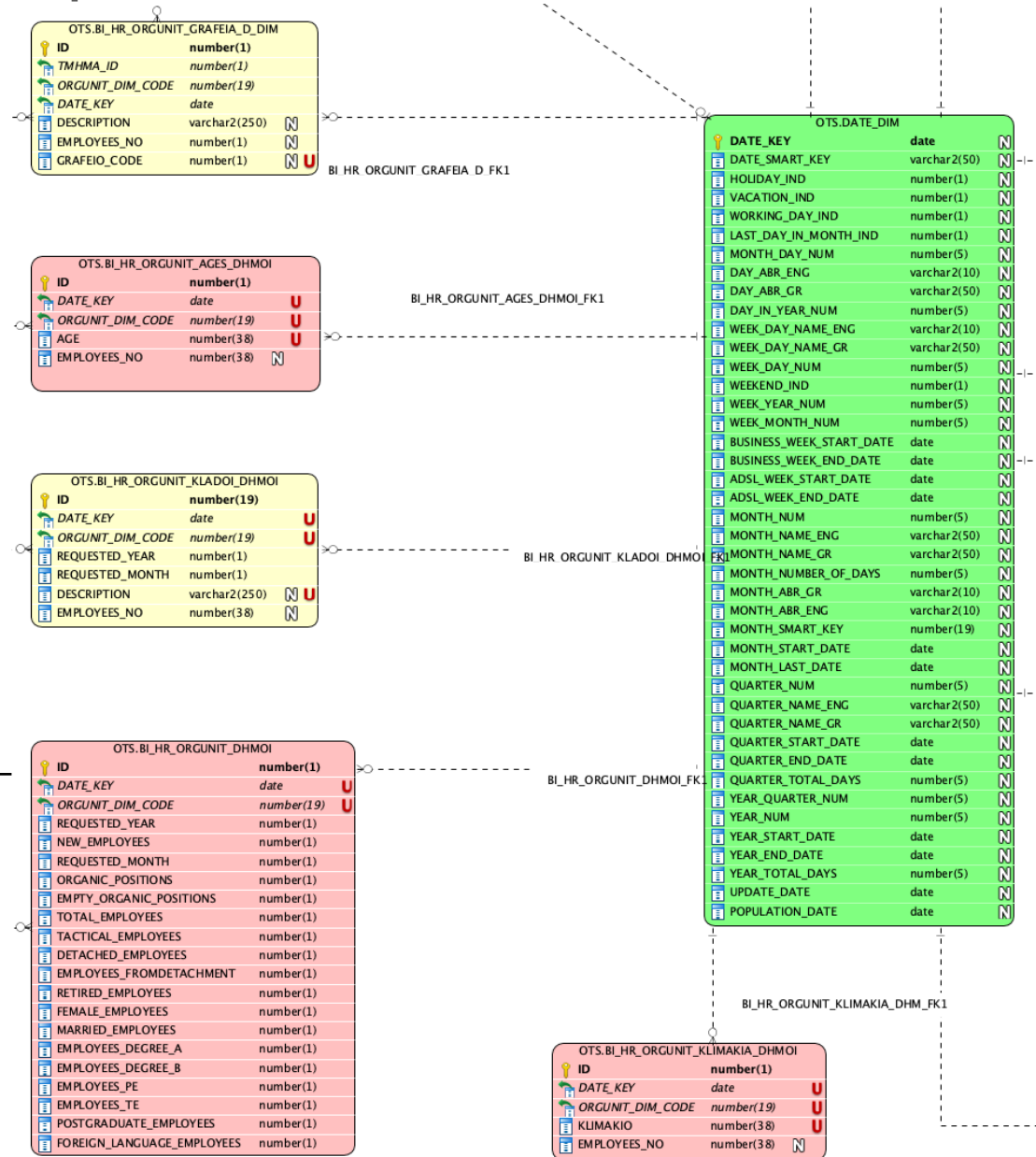
BI_HR_ORGUNIT_DHMOI

- DATE_KEY:dat
- DETACHED_EMPLOYEES:numbe
- EMPLOYEES_DEGREE_A:numbe
- EMPLOYEES_DEGREE_B:numbe
- EMPLOYEES_FROMDETACHMENT:numbe
- EMPLOYEES_PE:numbe
- EMPLOYEES_TE:numbe
- EMPTY_ORGANIC_POSITIONS:numbe
- FEMALE_EMPLOYEES:numbe
- FOREIGN_LANGUAGE_EMPLOYEES:numbe
- ID:numbe
- MARRIED_EMPLOYEES:numbe
- NEW_EMPLOYEES:numbe
- ORGANIC_POSITIONS:numbe
- ORGUNIT_DIM_CODE:number(1)
- POSTGRADUATE_EMPLOYEES:numbe
- REQUESTED_MONTH:numbe
- REQUESTED_YEAR:numbe
- RETIRED_EMPLOYEES:numbe
- TACTICAL_EMPLOYEES:numbe
- TOTAL_EMPLOYEES:numbe

Primary Keys: ID

Foreign Keys: DATE_KEY, ORGUNIT_DIM_CODE

Master Entities: DATE_DIM, BI_DHMOS_DIM



Παράδειγμα σχεδίασης δεδομένων

Ολοκληρωμένη σχεδίαση

- Αυτόματη παραγωγή κώδικα

```
CREATE TABLE BI_HR_ORGUNIT_DHMOI (  
  ID bit(1) NOT NULL,  
  DATE_KEY datetime NOT NULL,  
  ORGUNIT_DIM_CODE bigint(19) NOT NULL,  
  REQUESTED_YEAR bit(1) NOT NULL,  
  NEW_EMPLOYEES bit(1) NOT NULL,  
  REQUESTED_MONTH bit(1) NOT NULL,  
  ORGANIC_POSITIONS bit(1) NOT NULL,  
  EMPTY_ORGANIC_POSITIONS bit(1) NOT NULL,  
  TOTAL_EMPLOYEES bit(1) NOT NULL,  
  TACTICAL_EMPLOYEES bit(1) NOT NULL,  
  DETACHED_EMPLOYEES bit(1) NOT NULL,  
  EMPLOYEES_FROMDETACHMENT bit(1) NOT NULL,  
  RETIRED_EMPLOYEES bit(1) NOT NULL,  
  FEMALE_EMPLOYEES bit(1) NOT NULL,  
  MARRIED_EMPLOYEES bit(1) NOT NULL,  
  EMPLOYEES_DEGREE_A bit(1) NOT NULL,  
  EMPLOYEES_DEGREE_B bit(1) NOT NULL,  
  EMPLOYEES_PE bit(1) NOT NULL,  
  EMPLOYEES_TE bit(1) NOT NULL,  
  POSTGRADUATE_EMPLOYEES bit(1) NOT NULL,  
  FOREIGN_LANGUAGE_EMPLOYEES bit(1) NOT NULL,  
  CONSTRAINT BI_HR_ORGUNIT_DHMOI_PK  
  PRIMARY KEY (ID),  
  CONSTRAINT BI_HR_ORGUNIT_DHMOI_UK  
  UNIQUE (DATE_KEY, ORGUNIT_DIM_CODE));
```

