

Αρχιτεκτονική Πληροφοριακών Συστημάτων

Η ανάπτυξη μεγάλων εφαρμογών επικεντρωμένων στην ανταλλαγή δεδομένων μέσω δικτύου οδήγησε στο περιβάλλον Πελάτη / Εξυπηρετητή (Client/Server) στη δεκαετία του 1980. Σε αυτού του περιβάλλοντος τις αρχιτεκτονικές υπάρχει μια κεντρική εφαρμογή που λέγεται Εξυπηρετητής (Server) και με την οποία συνδέονται και ανταλλάσσουν πληροφορίες άλλες εφαρμογές που βρίσκονται στους υπολογιστές των χρηστών και χαρακτηρίζονται ως Πελάτες (Clients). Η ανταλλαγή της πληροφορίας γίνεται μέσω συναλλαγών (transactions), δηλαδή μέσω καλά ορισμένων αιτημάτων (requests) και αποκρίσεων (responses).



Διάγραμμα Κλάσεων: Client – Server

Το περιβάλλον Πελάτη / Εξυπηρετητή αφορά στη φυσική αρχιτεκτονική και εμφανίζει επιμέρους αρχιτεκτονικές.

1. Thin Client – Fat Server

Όταν το μεγαλύτερο μέρος της επεξεργασίας (λογική της εφαρμογής) γίνεται στον Εξυπηρετητή και ο Πελάτης αναλαμβάνει κυρίως την παρουσίαση αποτελεσμάτων της διαδικασίας, τότε έχουμε την περίπτωση Αδύνατου Πελάτη (Thin Client) – Παχύ Εξυπηρετητή (Fat Server). Στην ειδική περίπτωση του Διαδικτύου, όπου αποκλειστικά και μόνο το επίπεδο της παρουσίασης είναι στο φυλλομετρητή (browser), ενώ το σύνολο της επεξεργασίας είναι στον Εξυπηρετητή, μιλάμε για Υπέρ-Παχύ Εξυπηρετητή (Super-Fat Server).

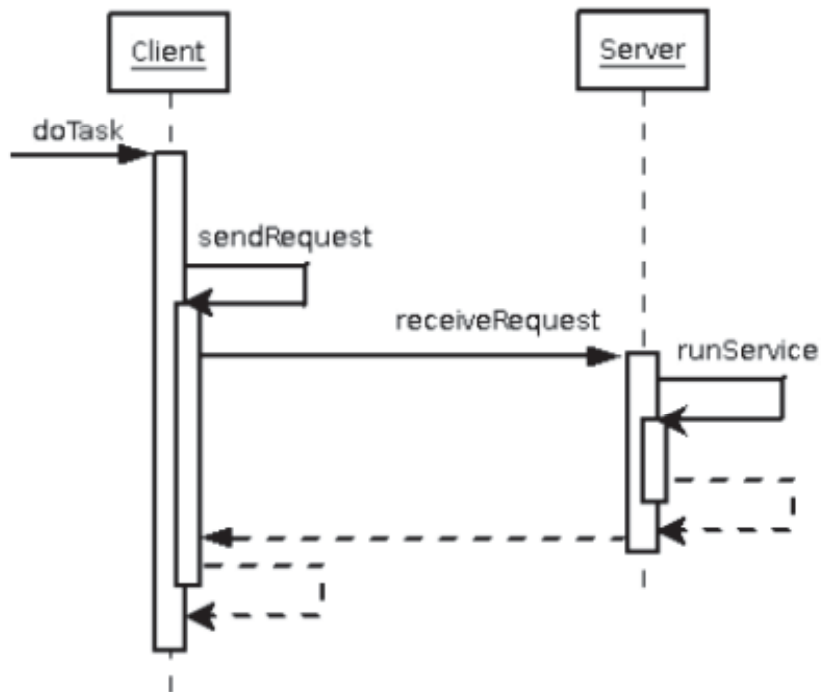
Η αρχιτεκτονική thin-client έχει ορισμένα μειονεκτήματα:

1. Προκαλεί σημαντικό φόρτο στο δίκτυο. Επίσης, προκαλεί σημαντικές καθυστερήσεις όταν οι διεργασίες πραγματοποιούνται μέσω του Διαδικτύου, λόγω των ροών εργασίας μεταξύ του φυλλομετρητή (client) και του εξυπηρετητή.
2. Η εφαρμογή πρέπει να έχει υλοποιηθεί ώστε να αποκρίνεται το ίδιο καλά σε κάθε γνωστό φυλλομετρητή. Το πρόβλημα αυτό -συνήθως- δεν υπάρχει στο εσωτερικό μιας επιχείρησης, όπου η διοίκηση των συστημάτων μπορεί να επιβάλλει, τη χρήση ενός συγκεκριμένου φυλλομετρητή.
3. Οι εφαρμογές φυλλομετρητή δύσκολα επιτυγχάνουν υψηλής ποιότητας περιβάλλοντα διεπαφών χρήστη. Αν και οι σύγχρονες σουίτες ανάπτυξης διαδικτυακών εφαρμογών (π.χ. Java SDK, Visual Studio κ.λπ.) προσφέρουν σημαντικές δυνατότητες ανάπτυξης πλούσιων διεπαφών χρήστη (user interfaces - UI), ακόμη και σήμερα δεν μπορούν να επιτύχουν λειτουργίες αντάξιες των εφαρμογών πελάτη που αναπτύσσονται μέσα από προγραμματιστικά περιβάλλοντα. Παρόλα αυτά, σήμερα, με την πρωτότυπη χρήση γνωστών τεχνολογιών όπως η Javascript, η XML και τα CSS (Cascading Style Sheets) Διαδοχικά Φύλλα Στυλ στο πλαίσιο της τεχνολογίας AJAX (Asynchronous Java and XML), υπάρχει η δυνατότητα κατασκευής διεπαφών φυλλομετρητή που προσομοιάζουν αυτών των εφαρμογών σταθμών εργασίας (desktop applications).

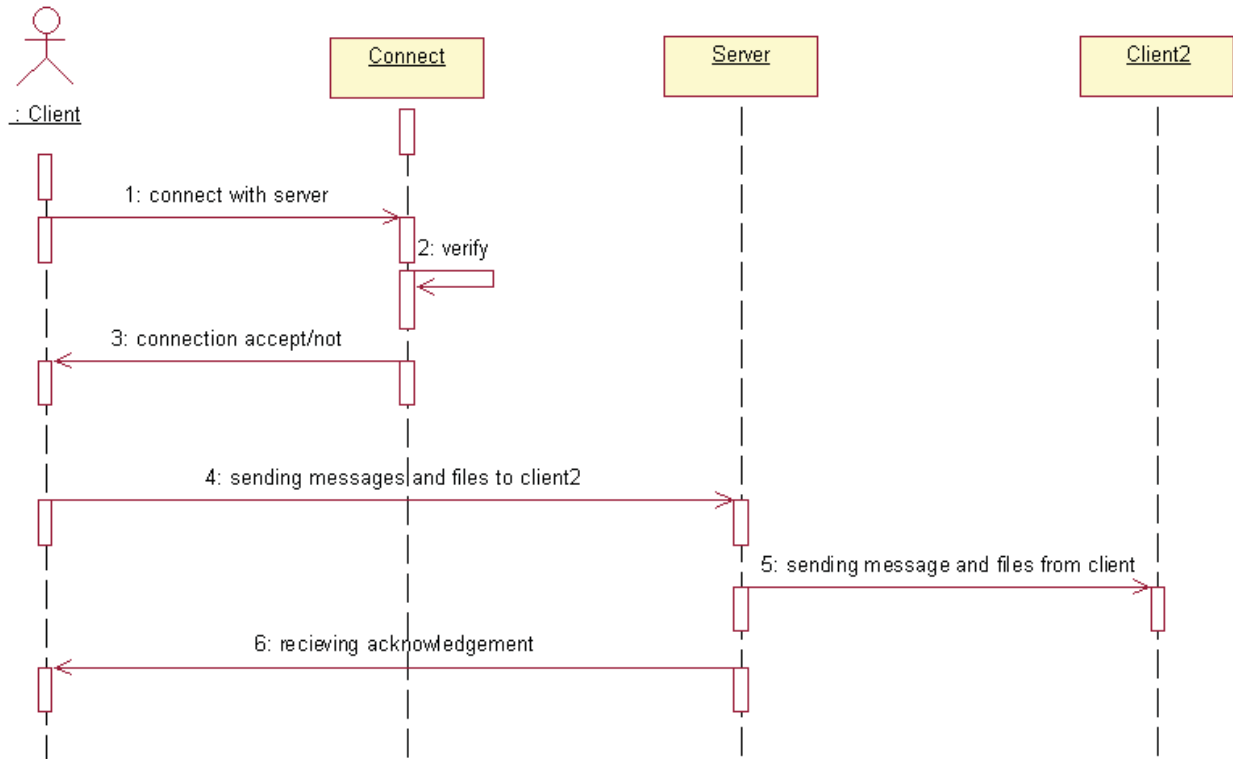
2. Fat Client - Thin Server

Όταν ο Πελάτης αναλαμβάνει το μεγαλύτερο μέρος της επεξεργασίας μιλάμε για Παχύ Πελάτη (Fat Client)–Αδύνατο Εξυπηρετητή (Thin Server).

- Στον thin client model, το presentation layer υλοποιείται στον client και όλες οι υπόλοιπες βαθμίδες (layers) υλοποιούνται στον server.
- Στον fat client, ένα μέρος ή και το σύνολο του application processing πραγματοποιείται στον client. Το Data management και οι λειτουργίες της database υλοποιούνται από τον server.
- Ο thin client συνεπάγεται υψηλό processing βάρος τόσο για τον server όσο και για το network. Αυτό συμβαίνει διότι ο server είναι υπεύθυνος για το σύνολο των υπό εκτέλεση υπολογισμών (computation) πράγμα που μπορεί να οδηγήσει σε σημαντικό network traffic μεταξύ client και server.



Διάγραμμα Ακολουθίας: Client - Server



Παράδειγμα Διαγράμματος Ακολουθίας: Client - Server

Λογικός διαχωρισμός μιας εφαρμογής

Ο λογικός διαχωρισμός μιας εφαρμογής γίνεται σε βαθμίδες ή αλλιώς layers και αποσκοπεί στην μεγαλύτερη δυνατή ανεξαρτησία ανάμεσα στις συναρτήσεις της εφαρμογής, ώστε να υπάρχει η δυνατότητα επέκτασης της χωρίς να είναι απαραίτητες μεγάλες αλλαγές στον κώδικα της.

Οι βαθμίδες στις οποίες διαχωρίζεται μια διαδικτυακή εφαρμογή είναι 3:

(Α.) Παρουσίασης

Η βαθμίδα αυτή απαρτίζεται από το User interface (UI) του προγράμματος. Το επίπεδο παρουσίασης είναι υπεύθυνο για τις ακόλουθες διαδικασίες:

1. Συλλογή δεδομένων και πληροφοριών από τον χρήστη
2. Αποστολή δεδομένων και πληροφοριών στη βαθμίδα των διεργασιών
3. Λήψη δεδομένων και πληροφοριών από την βαθμίδα των διεργασιών
4. Παρουσίαση των δεδομένων και των πληροφοριών στον χρήστη.

(Β.) Λειτουργιών -Διεργασιών

Η βαθμίδα αυτή είναι η βαθμίδα στην οποία πραγματοποιούνται όλες οι εργασίες του προγράμματος και αποτελεί το πιο σημαντικό κομμάτι σε μια εφαρμογή. Σε αυτή την βαθμίδα φιλοξενείται η επιχειρησιακή λογική της εφαρμογής.

Η βαθμίδα αυτή είναι υπεύθυνη για τις ακόλουθες διαδικασίες:

1. Λήψη δεδομένων και πληροφοριών από την βαθμίδα της παρουσίασης
2. Επεξεργασία και αξιολόγηση των ληφθέντων πληροφοριών
3. Επικοινωνία με την βαθμίδα των δεδομένων σε περίπτωση που χρειάζεται επεξεργασία δεδομένων
4. Αποστολή των πληροφοριών στη βαθμίδα παρουσίασης ώστε να έχει πρόσβαση σε αυτά ο χρήστης.

(Γ.) Δεδομένων

Στη βαθμίδα δεδομένων αποθηκεύονται όλα τα δεδομένα της εφαρμογής. Είναι επίσης το μέρος από όπου αντλούνται τα δεδομένα προκειμένου να τους γίνει επεξεργασία και να παρουσιάσουν στον χρήστη. Η βαθμίδα δεδομένων μπορεί να αποτελείται από σχεσιακές ή και μη σχεσιακές βάσεις δεδομένων.

Η βαθμίδα αυτή είναι υπεύθυνη για τις ακόλουθες διαδικασίες:

1. Αποθήκευση, ενημέρωση και διαγραφή δεδομένων από το σύστημα
2. Δυνατότητα αναζήτησης και άντλησης δεδομένων του συστήματος
3. Διατήρηση των πληροφοριών
4. Εξασφάλιση ακεραιότητας των δεδομένων
5. Γρήγορη πρόσβαση στα δεδομένα

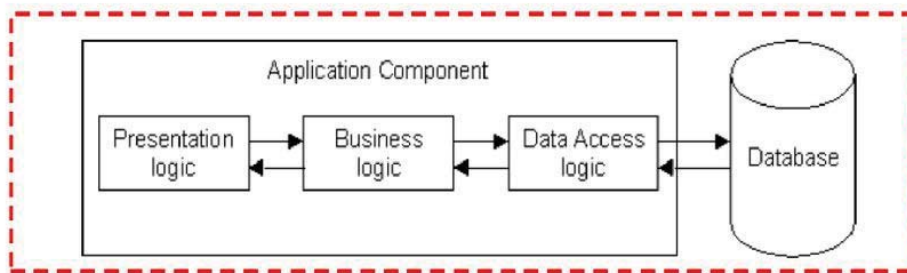
Φυσικός διαχωρισμός μιας εφαρμογής

Ο φυσικός διαχωρισμός μιας εφαρμογής γίνεται σε επίπεδα ή αλλιώς tiers.

A. Αρχιτεκτονική ενός επιπέδου

Η απλούστερη αρχιτεκτονική είναι η αρχιτεκτονική μιας βαθμίδας (1-tier) στην οποία ο υπολογιστής που φιλοξενεί την εφαρμογή είναι ο ίδιος με τον οποίο ο χρήστης επικοινωνεί με την εφαρμογή. Στην 1-tier αρχιτεκτονική τα δεδομένα αποθηκεύονται στον τοπικό δίσκο ενός προσωπικού υπολογιστή και ταυτόχρονα παρέχει τη δυνατότητα προσπέλασης των δεδομένων μέσω διεπαφής που βρίσκεται στον ίδιο υπολογιστή.

1-Tier Architecture



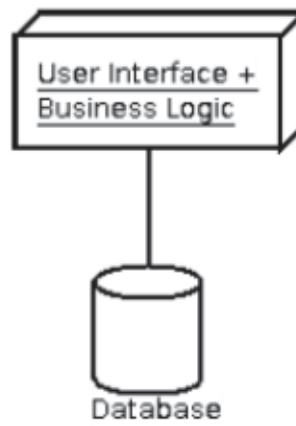
- Και τα 3 layers βρίσκονται στον ίδιο υπολογιστή
- Όλη η επεξεργασία πραγματοποιείται στον υπολογιστή
- Το presentation, logic και data layer είναι στενά συνδεδεμένα μεταξύ τους
- Αλλαγές σε ένα layer, απαιτούν αλλαγές και στα υπόλοιπα layers

Business logic is the part of the program that encodes the real-world business rules that determine how data can be created, stored, and changed. Business logic also handles the exchange of information between a database and user interface. Business logic is real-world business rules put into computer code and shown in a computer program via a user interface. Business logic is most evident in its role in creating workflows that pass data between users and software systems.

Data access logic consists of the definitions of database tables and columns and the computer logic that is needed to navigate the database. The data access layer enforces rules regarding the storage and access of information. For example: dates must be valid dates and numeric fields must never contain alphanumeric characters. **Data access logic** used to keep the code you use to pull data from your database separate from business logic and presentation code. This way, if you have to change database, you don't end up rewriting the whole thing.

B. Αρχιτεκτονική δύο βαθμίδων (2-tier)

Σε αυτή την αρχιτεκτονική, ο πελάτης στέλνει ένα αίτημα (request) για δεδομένα στον εξυπηρετητή και αυτός επιστρέφει την απάντηση (response), την οποία επεξεργάζεται ο πελάτης και εμφανίζει στο χρήστη τα αποτελέσματα. Πρόκειται για την περίπτωση που ένα κεντρικό σύστημα, πολλές φορές με αποκλειστική αρμοδιότητα (server), παρέχει τη λειτουργικότητα για πολλά συστήματα χρηστών (clients). Το σύστημα βάσεως δεδομένων είναι στον εξυπηρετητή (server), με τον οποίο επικοινωνούν οι εφαρμογές – πελάτες. Στην αρχιτεκτονική δύο βαθμίδων (2-tier) ο client χειρίζεται τόσο το Presentation layer όσο και το Application layer (logical operations), ενώ ο server χειρίζεται το database layer.



Αρχιτεκτονική δύο βαθμίδων (2-tier)

2-Tiered Architecture



Advantages:

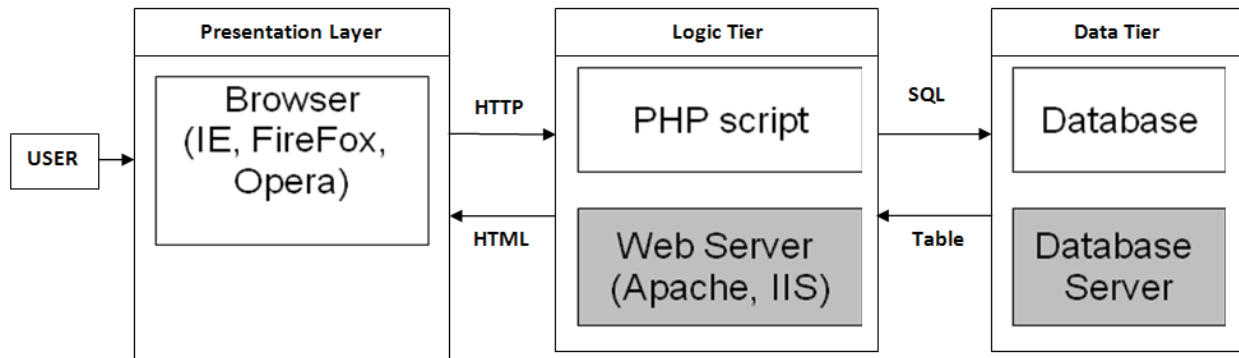
1. Easy to maintain and modification is easy
2. Communication is faster

Disadvantages:

1. In two tier architecture application performance will be degrade upon increasing the users.
2. Επίσης η εμφάνιση των δεδομένων και η επεξεργασία τους γίνεται από το ίδιο πρόγραμμα, τον πελάτη (client). Αν υπάρχει συχνή αλλαγή στη μορφή παρουσίασης, τότε θα πρέπει να αλλάζει κάθε φορά η client εφαρμογή.

Γ. Αρχιτεκτονική τριών βαθμίδων (3-tier)

Στην αρχιτεκτονική τριών βαθμίδων (3-tier), υπάρχει σαφής φυσικός διαχωρισμός της επεξεργασίας από τη βάση δεδομένων και την παρουσίαση. Η 3-tier αρχιτεκτονική αποτελεί μία από τις μεγαλύτερες καινοτομίες στα πληροφοριακά συστήματα.



Database Server

Ο Database Server παρέχει όλες τις απαραίτητες λειτουργίες για την αποθήκευση, ανάκτηση, ενημέρωση και συντήρηση των δεδομένων του συστήματος καθώς επίσης και όλους τους απαραίτητους μηχανισμούς για την ακεραιότητα των δεδομένων (Data Integrity).

Application Server

Αποτελεί το κύριο τμήμα του λογισμικού, στο οποίο εκτελούνται οι περισσότερες λειτουργίες, εκτός εκείνων που σχετίζονται με τη διαμόρφωση των οθονών εργασίας. Υπάρχει δυνατότητα εγκατάστασης περισσότερων του ενός Application Servers σε διαφορετικά μηχανήματα, αξιοποιώντας, με τον τρόπο αυτό, οποιαδήποτε διαθέσιμη υπολογιστική ισχύ και εξασφαλίζοντας εξαιρετικά αποτελέσματα ανταπόκρισης, αξιοπιστίας και επεκτασιμότητας. Με την κατανομή των Application Servers σε ανεξάρτητα μηχανήματα, επιτυγχάνεται αποσυμφόρηση του συνολικού φόρτου του συστήματος, αφού κάθε Application Server είναι σε θέση να υποστηρίξει ένα υποσύνολο του συνολικού αριθμού των Remote Clients (π.χ. Ο Application Server A θα εξυπηρετεί τους Clients του υποκαταστήματος A, ενώ ο Application Server B θα εξυπηρετεί τους Clients του υποκαταστήματος B).

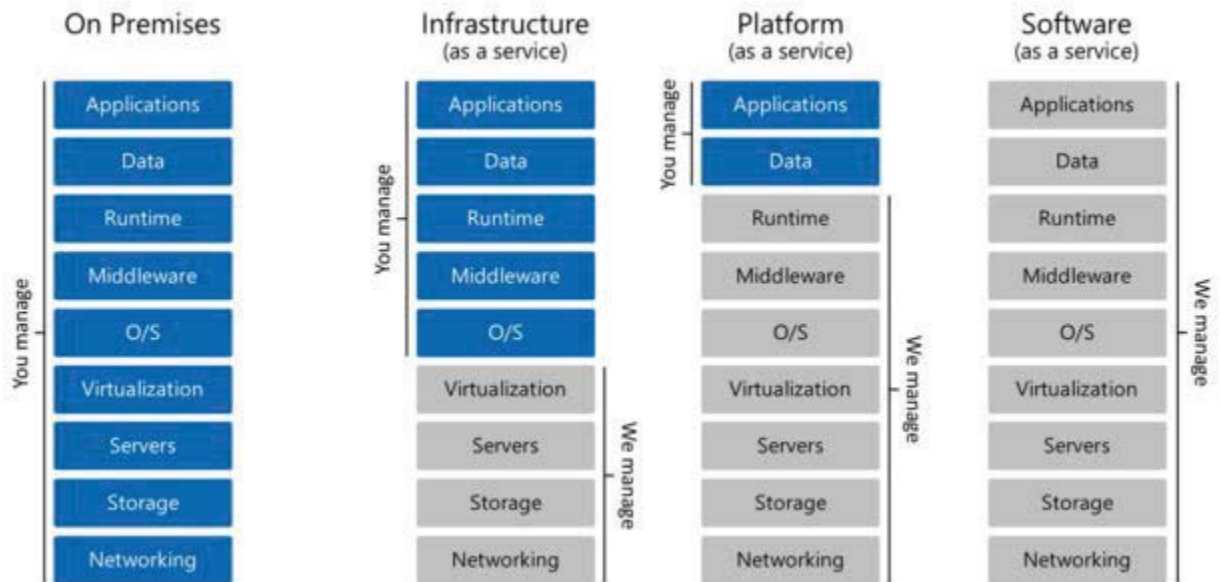
Client - Presentation

Το τρίτο επίπεδο του λογισμικού αποτελεί τη επαφή του χρήστη με το σύστημα (User Interface). Στο επίπεδο αυτό, πραγματοποιείται η διαχείριση των Οθονών Εργασίας (User Screens) καθώς επίσης και η μορφοποίηση των δεδομένων που εμφανίζονται.

Η συγκρότηση του συστήματος σε τρία επίπεδα (3-tier) εξασφαλίζει:

1. Την ελαχιστοποίηση της επιβάρυνσης του δικτύου λόγω μεταφοράς μεγάλου όγκου δεδομένων π.χ. η εκτέλεση ενός Query για την ανάκτηση μερικών εγγραφών από έναν πίνακα με δεκάδες χιλιάδες εγγραφές γίνεται στο διακομιστή εφαρμογής (Application Server), από τον οποίο μεταφέρεται στο χρήστη μόνο το αποτέλεσμα.
2. Τη δυνατότητα διαχωρισμού του διακομιστή δεδομένων (Database Server) από το διακομιστή ή τούς διακομιστές εφαρμογής (Application Servers), ώστε να εκτελούνται σε διαφορετικά μηχανήματα. Κατά συνέπεια, ο καθορισμός των κρίσιμων μεγεθών απόδοσης των αντίστοιχων μηχανών (sizing) μπορεί να γίνεται ανεξάρτητα, ενώ παράλληλα εξασφαλίζεται απεριόριστη επεκτασιμότητα, χωρίς ανακατασκευή, του λογισμικού.
3. Τη μέγιστη ευελιξία στην επιλογή του διακομιστή δεδομένων (Database Server), καθώς επιτρέπεται η χρήση οποιουδήποτε μηχανήματος με οποιοδήποτε λειτουργικό σύστημα (π.χ. Windows NT ή UNIX etc), με μοναδική απαίτηση τη δυνατότητα επικοινωνίας δια μέσου TCP/IP πρωτοκόλλου. Έτσι, είναι δυνατή η μεταγενέστερη αναβάθμιση ως προς τη βάση δεδομένων με την αλλαγή / αναβάθμιση του μηχανήματος, χωρίς να επηρεάζεται το υπόλοιπο σύστημα.

ERP-as-a-Service



Software-as-a-Service (SaaS)

SaaS is often the top choice for any organization that wants a low-overhead, scalable and always up-to-date replacement for traditional software. Under a SaaS licensing agreement, we take care of the entire support stack for you, from the network to the application itself, including all of the virtualization, middleware, etc. in between. This is the inverse of the on-prem approach, where you manage everything.

Platform-as-a-Service (PaaS)

PaaS requires slightly more management from you, the customer, than SaaS. You will typically be responsible for managing the application and its data, while we oversee the rest of the stack. With PaaS, you get to build and run applications on the cloud infrastructure instead of your own, which can save you the time and overhead of having to deal with everything in-house.

Infrastructure-as-a-Service (IaaS)

IaaS is the minimal hosting service level, putting most of the control in your hands while we handle the servers, storage, virtualization and networking. Many businesses rely on IaaS to extend their on-premise infrastructure and achieve a level of on-demand resourcing that they simply cannot replicate on their own at a reasonable cost with any effectiveness.