
2. Η Αρχιτεκτονική των Πληροφοριακών Συστημάτων Επιχειρήσεων

Σύνοψη

Η αρχιτεκτονική των συστημάτων ERP είναι ιδιαίτερα σύνθετη και πολυεπίπεδη. Αυτό σε συνδυασμό με την πανσπερμία των αρχιτεκτονικών που προτείνονται από διάφορους κατασκευαστές λογισμικού, τη μεγάλη ποικιλία των τεχνολογιών που χρησιμοποιούνται, τις υπάρχουσες επικαλύψεις καθώς και την εκτενή ορολογία που χρησιμοποιείται κάνει την περιγραφή της αρχιτεκτονικής ενός τέτοιου συστήματος ιδιαίτερα πολύπλοκη. Μια αρχιτεκτονική θα πρέπει να λαμβάνει υπόψη συστήματα, επιχειρησιακές ενότητες όπως SCM, CRM, ηλεκτρονικό εμπόριο, ηλεκτρονικές προμήθειες, επιχειρηματική νοημοσύνη κ.λπ. Στο κεφάλαιο αυτό θα παρουσιαστούν βασικά στοιχεία της αρχιτεκτονικής, τα επίπεδα της αρχιτεκτονικής, η λειτουργικότητα του κάθε επιπέδου, οι διεπαφές κ.λπ. Επιπλέον, θα παρουσιαστεί η απαραίτητη πληροφοριακή υποδομή που απαιτείται σε μια σύγχρονη επιχείρηση και αναφέρεται στην ολοκληρωμένη διαχείριση χρηστών και δικαιωμάτων, τον σχεδιασμό επιχειρηματικών διεργασιών, την ολοκλήρωση της πληροφορίας κ.λπ.

Προαπαιτούμενη γνώση

Το κεφάλαιο απαιτεί κατανόηση των βασικών αρχών λειτουργίας πληροφοριακών επιχειρήσεων και της τεχνολογίας λογισμικού.

2.1 Αρχιτεκτονικές Συστημάτων

Ένα πληροφοριακό σύστημα είναι ένα σύνολο από διασυνδεδεμένες συσκευές, εφαρμογές και πόρους. Η αρχιτεκτονική είναι ένα μοντέλο που ορίζει τη δομή, τη συμπεριφορά και τις όψεις ενός πληροφοριακού συστήματος (Jaakkola, 2010). Σύμφωνα με τους Kruchten et al. (2004):

«Η αρχιτεκτονική ενός πληροφοριακού συστήματος εμπεριέχει το σύνολο των σημαντικών αποφάσεων σχετικά με την οργάνωση του συστήματος που περιλαμβάνουν την επιλογή των δομικών στοιχείων του και των διασυνδέσεών του· τη συμπεριφορά του, όπως ορίζεται στη συνεργασία μεταξύ των συστατικών του· τη σύνθεση αυτών των διαρθρωτικών συστατικών και των στοιχείων συμπεριφοράς σε μεγαλύτερα υποσυστήματα· και ένα αρχιτεκτονικό στυλ που καθοδηγεί αυτή την οργάνωση. Επίσης, η αρχιτεκτονική ενός πληροφοριακού συστήματος εμπεριέχει αποφάσεις σχετικά με τη λειτουργικότητα (functionality), τη χρηστικότητα (usability), την ανθεκτικότητα (resilience), τις επιδόσεις (performance), την επαναχρησιμοποίηση (reuse), τον εύληπτο χαρακτήρα (comprehensibility), τους οικονομικούς και τους τεχνολογικούς περιορισμούς (economic and technology constraints), αλλά και την αισθητική του συστήματος (aesthetics)».

Η αρχιτεκτονική ενός πληροφοριακού συστήματος αποτελεί τη γέφυρα μεταξύ των επιχειρηματικών και των τεχνικών απαιτήσεων, η οποία επιτυγχάνεται με την καταγραφή και αντιστοίχιση των περιπτώσεων χρήσης του συστήματος με τις τεχνικές λύσεις που τις υλοποιούν. Συνεπώς, στόχος της αρχιτεκτονικής θα πρέπει να είναι να προσδιοριστούν οι απαιτήσεις που επηρεάζουν τη δομή της τεχνικής λύσης και γενικότερα του πληροφοριακού συστήματος. Επομένως, μια καλή αρχιτεκτονική μειώνει τους επιχειρηματικούς κινδύνους που συνδέονται με τη δημιουργία του πληροφοριακού συστήματος. Επιπλέον, μια ευέλικτη σχεδίαση επιτρέπει να χειριστούμε με αποτελεσματικό τρόπο τις αλλαγές που θα προκύψουν κατά τη διάρκεια ζωής του πληροφοριακού συστήματος, οι οποίες προέρχονται είτε από αλλαγές στο λογισμικό συστημάτων, είτε από αλλαγές στο hardware, είτε από αλλαγές των αναγκών της επιχείρησης. Τέλος, η αρχιτεκτονική ομογενοποιεί το σύνολο, δίνοντας δομή στο πληροφοριακό σύστημα (Shaw, 1996).

Η αρχιτεκτονική ενός πληροφοριακού συστήματος λαμβάνει υπόψη τις ακόλουθες απαιτήσεις (Meier et al., 2009):

- 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 διαχειρισσιμότητα και σε επεκτασιμότητα.
- Τα εμπλεκόμενα μέρη στη διαμόρφωση της αρχιτεκτονικής και τους χρήστες του συστήματος.

- Την εμβέλεια της επιχείρησης. Η εμβέλεια αφορά στη γεωγραφική έκταση στην οποία δραστηριοποιείται η επιχείρηση και επηρεάζει εκτός από το μέγεθος, διάφορες παραμέτρους του συστήματος (π.χ. χρησιμοποιούμενες γλώσσες, time zones).
- Τα δεδομένα που παράγονται και τηρούνται στο σύστημα.

Στη συνέχεια παρουσιάζονται πιο αναλυτικά σε σχέση με την εισαγωγή οι πλέον γνωστές αρχιτεκτονικές συστημάτων, με τα πλεονεκτήματα και τις αδυναμίες τους, καθώς και ο βέλτιστος τρόπος χρήσης τους.

2.1.1 Παραδοσιακές Εφαρμογές (Legacy Applications)

Οι παραδοσιακές εφαρμογές που έχουν τα ακόλουθα χαρακτηριστικά είναι ιδιαίτερα προβληματικές:

- «Μονολιθικά» σχεδιασμένες εφαρμογές, που αποτελούνται από μια σειρά διεργασιών, χωρίς συγκεκριμένη λογική διασύνδεσης και οι οποίες δεν διαλειτουργούν με άλλες εφαρμογές.
- Εφαρμογές που διαθέτουν σταθερές (fixed) διεπαφές με τον χρήστη (user interfaces - UI). Οι διεπαφές αυτές είναι δύσκολο να αντικατασταθούν από διεπαφές φυλλομετρητή (web based UI).
- Εφαρμογές που διαθέτουν εσωτερικές δηλώσεις σταθερών και μεταβλητών (hard coded data definitions). Οι δηλώσεις αυτές είναι -συνήθως- συγκεκριμένες για την εφαρμογή και δεν συμμορφώνονται με προσεγγίσεις μοντέλων δεδομένων της επιχείρησης (enterprise data model approach).
- Εφαρμογές που διαθέτουν εσωτερικές δηλώσεις επιχειρησιακών κανόνων (hard coded business rules). Οι εφαρμογές αυτές είναι δύσκολο να επικαιροποιηθούν όταν οι επιχειρησιακοί κανόνες μεταβάλλονται λόγω μεταβολών στις επιχειρησιακές διεργασίες.
- Εφαρμογές που αποθηκεύουν οι ίδιες τα στοιχεία αυθεντικοποίησης των χρηστών (user credentials). Οι εφαρμογές αυτές δυσκολεύουν τη μετάβαση μιας επιχείρησης σε πληροφοριακά συστήματα που επιτρέπουν την πρόσβαση των χρηστών σε πόρους, με τη χρήση μοναδικής αυθεντικοποίησης (single sign-on) και με μεθόδους διαχείρισης ταυτότητας (identity management).

Η ύπαρξη παραδοσιακών εφαρμογών σε μια επιχείρηση πρέπει να ληφθεί υπόψη κατά τη σχεδίαση ενός νέου πληροφοριακού συστήματος ERP. Το νέο πληροφοριακό σύστημα θα πρέπει να διαθέτει μηχανισμούς διαλειτουργικότητας με παραδοσιακές εφαρμογές ή να διαθέτει νέες εφαρμογές που θα αποτελούν την εξέλιξη των παραδοσιακών. Στη δεύτερη περίπτωση, πρέπει να προηγηθεί η μεταφορά των δεδομένων από τις παραδοσιακές εφαρμογές στις νέες. Σε περίπτωση ανάπτυξης νέων πληροφοριακών συστημάτων όταν το επιβάλλουν οι επιχειρηματικές συνθήκες, απαραίτητη προϋπόθεση είναι να εξασφαλιστεί η διαλειτουργικότητα του συστήματος.

2.1.2 Αρχιτεκτονική πελάτη/εξυπηρετητή (client/server)

Σε αυτή την αρχιτεκτονική υπάρχουν δύο βασικές οντότητες:

1. Του πελάτη (client), δηλαδή ένας σταθμός εργασίας που διαθέτει μια εφαρμογή που εκτελείται σε αυτόν και στην οποία έχει πρόσβαση ένας χρήστης και
2. 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». Στην αρχιτεκτονική αυτή, οι απαιτούμενοι πόροι για την εκτέλεση μιας επιχειρησιακής διαδικασίας μοιράζονται μεταξύ πελάτη και εξυπηρετητή.

Η εν λόγω αρχιτεκτονική χρησιμοποιήθηκε ευρύτατα στο παρελθόν για πολλούς λόγους. Ένας λόγος ήταν ότι η αρχιτεκτονική πελάτη/εξυπηρετητή επέτρεπε σε διαφορετικούς χρήστες να έχουν πρόσβαση σε διαφορετικά δεδομένα, σε αντίθεση με προγενέστερες εφαρμογές (mainframe εφαρμογές). Ένας ακόμη λόγος ήταν η εξοικονόμηση χρημάτων, καθώς οι εφαρμογές πελάτη μπορούσαν να εκτελούνται σε σταθμούς εργασίας (H/Y) με μικρές υπολογιστικές δυνατότητες. Η εξέλιξη των δικτυακών τεχνολογιών (τοπικά και υπερτοπικά δίκτυα) αποτέλεσε έναν πρόσθετο παράγοντα ενίσχυσης της χρήσης της εν λόγω αρχιτεκτονικής.

Η εν λόγω αρχιτεκτονική οδήγησε στην ανάπτυξη ισχυρών εφαρμογών για σταθμούς εργασίας (desktop applications), καθώς και στη διάδοση της χρήσης εφαρμογών από χρήστες που δεν ήταν προγραμματιστές. Ωστόσο, η αρχιτεκτονική αυτή δεν χρησιμοποιείται πλέον, καθώς:

1. Τα κόστη συντήρησης και αναβάθμισης είναι σημαντικά. Η συντήρηση πολλαπλών σταθμών εργασίας έχει σημαντικό κόστος. Επίσης, κάθε αναβάθμιση της εφαρμογής πελάτη, απαιτεί την επικαιροποίηση της εφαρμογής αυτής σε όλους τους σταθμούς εργασίας, γεγονός που επιφέρει επίσης σημαντικά κόστη σε χρήμα και ανθρωπο-προσπάθεια.
2. Η εφαρμογή πελάτη μπορεί να συντηρεί ανομοιογενή δεδομένα στους σταθμούς εργασίας.
3. Η κεντρική διαχείριση πολλαπλών εφαρμογών πελάτη/εξυπηρετητή είναι μια δύσκολη διαδικασία λόγω της κατανομής της σε πολλούς σταθμούς εργασίας.

Η αντιμετώπιση των μειονεκτημάτων αυτού του τύπου εφαρμογών μπορεί να επιτευχθεί με τις ακόλουθες μεθόδους:

- Εγκατάσταση της εφαρμογής πελάτη σε έναν εξυπηρετητή και μετατροπή των σταθμών εργασίας σε «τυφλά τερματικά» (dumb terminals). Η διαδικασία αυτή είναι εφικτή σε κάθε πληροφοριακό σύστημα που βασίζεται σε λειτουργικά συστήματα με λειτουργίες πολλαπλών χρηστών (multi-user) και δίνει τη δυνατότητα κεντρικής διαχείρισης των εφαρμογών πελάτη/εξυπηρετητή.
- Εξαγωγή των λειτουργιών της εφαρμογής σε μια σειρά διεπαφών. Η εξαγωγή μπορεί να επιτευχθεί με τη μετατροπή της εφαρμογής πελάτη/εξυπηρετητή σε μια σειρά λειτουργικών οντοτήτων (components). Μερικές από τις οντότητες αυτές μπορούν να ενσωματωθούν σε τρίτες εφαρμογές. Με τη διαδικασία αυτή μπορεί να επιτευχθεί η σταδιακή αντικατάσταση της εφαρμογής πελάτη/εξυπηρετητή. Η αρχιτεκτονική συστημάτων που διαχειρίζεται τέτοιου είδους μεθόδους είναι η «αρχιτεκτονική προσανατολισμένη σε υπηρεσίες» (Service Oriented Architecture – SOA).
- Μετάπτωση των εφαρμογών αυτών σε εφαρμογές φυλλομετρητή (thin-client architecture). Η διαδικασία μετάπτωσης σημαίνει την αντικατάσταση της εφαρμογής πελάτη από αντίστοιχη εφαρμογή που εκτελείται από φυλλομετρητή. Η μετάπτωση είναι αναγκαία όταν οι εφαρμογές πελάτη/εξυπηρετητή έχουν πρόσβαση σε σημαντικά δεδομένα της επιχείρησης.

2.1.3 Αρχιτεκτονική thin-client

Η αρχιτεκτονική thin-client είναι μια δημοφιλής προσέγγιση στη διάκριση μεταξύ παρουσίασης, επιχειρησιακής λογικής και δεδομένων. 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 στην προηγούμενη ενότητα.

Όπως αναφέρθηκε στην προηγούμενη ενότητα, οι εφαρμογές πελάτη/εξυπηρετητή έχουν σημαντικά κόστη ως προς τη διαχείριση και τη συντήρηση ή αναβάθμισή τους. Η αρχιτεκτονική thin-client έρχεται να μειώσει τα ανωτέρω κόστη. Με την εν λόγω αρχιτεκτονική, όλες οι εργασίες που επιτελούνται από την εφαρμογή πραγματοποιούνται στον εξυπηρετητή.

Η εν λόγω αρχιτεκτονική έχει ως σημαντικό πλεονέκτημα ότι εκτελείται ανεξαρτήτως συσκευής (device independent) αλλά και λειτουργικού συστήματος συσκευής (operating system independent), καθώς η εκτέλεση της εφαρμογής εξαρτάται αποκλειστικά από τον φυλλομετρητή. Το γεγονός αυτό δίνει τη δυνατότητα να εκτελείται η εφαρμογή ακόμη και από χαμηλής ισχύος σταθμούς εργασίας ή και από φορητές συσκευές (π.χ. κινητά τηλέφωνα και ταμπλέτες).

Ωστόσο, 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).

4. Τα δεδομένα που δίνονται από τον χρήστη πρέπει να ελέγχονται (validated) ώστε να αποφεύγονται πολλαπλές ανταλλαγές αιτημάτων, ιδιαίτερα μέσω του Διαδικτύου, αλλά και να επιτυγχάνεται τόσο η ικανοποίηση του χρήστη όσο και η ασφάλεια των ανταλλαγών πληροφοριών.
5. Οι υπολογιστικοί πόροι των εξυπηρετητών ιστού που προσφέρουν τις εφαρμογές thin-client μπορούν να εξαντληθούν γρήγορα, όταν τα αιτήματα των χρηστών είναι πολλά.

Ο καλός σχεδιασμός μιας thin-client εφαρμογής είναι η λύση στα παραπάνω προβλήματα. Για παράδειγμα, η αντιμετώπιση ζητημάτων όπως το μέγεθος των δεδομένων που τηρούνται ανά σύνοδο (session), οι ενεργές συνδέσεις σε δεδομένα, ο χρόνος εκτέλεσης ενός αιτήματος σε μια βάση δεδομένων, μπορούν να επιφέρουν σημαντικές βελτιώσεις στην απόδοση μιας εφαρμογής thin-client.

2.1.4 Αρχιτεκτονική «προσανατολισμένη σε υπηρεσίες» (Service Oriented Architecture – SOA)

Μια προσανατολισμένη σε υπηρεσίες αρχιτεκτονική (SOA) είναι ένα αρχιτεκτονικό πρότυπο για την οργάνωση και τη χρήση δυνατοτήτων που μπορεί να είναι υπό τον έλεγχο των διαφόρων τομέων της ιδιοκτησίας. Παρέχει ένα ενιαίο μέσο για να προσφέρουν, να ανακαλύψουν, να αλληλεπιδρούν με τις δυνατότητες και τη χρήση για την παραγωγή επιθυμητών αποτελεσμάτων σύμφωνα με μετρήσιμες προϋποθέσεις και προσδοκίες.

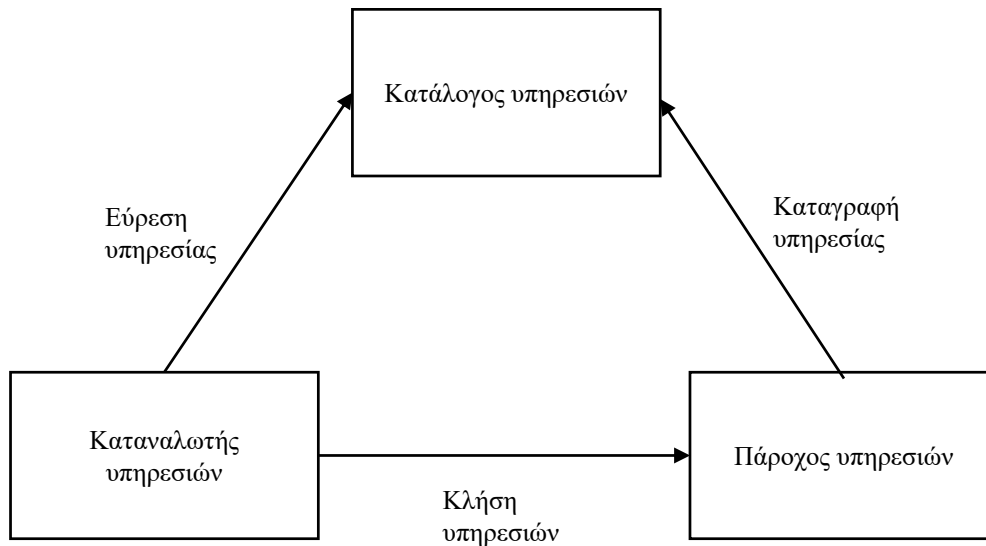
Η προσανατολισμένη σε υπηρεσίες αρχιτεκτονική (SOA) διαχωρίζει τις υπηρεσίες (λειτουργίες που προσφέρει ένα σύστημα) από τους αποδέκτες της (συστήματα που χρειάζονται αυτές τις λειτουργίες). Ο διαχωρισμός αυτός επιτυγχάνεται με μηχανισμούς που ονομάζονται συμβόλαια υπηρεσιών (*service contracts*), σε συνδυασμό με μηχανισμούς προς τους παρόχους (που δημοσιεύουν συμβόλαια) και τους πελάτες (που είναι οι αποδέκτες των συμβολαίων). Για την κατανόησή της θα πρέπει να θεωρήσουμε τις υπηρεσίες ως τις εφαρμογές του συστήματος, ενώ τα συμβόλαια είναι οι διεπαφές (interfaces) μεταξύ των εφαρμογών του συστήματος.

Σε αντίθεση με τις παραδοσιακές αρχιτεκτονικές που συνδυάζουν υπηρεσίες με πελάτες, η SOA διαχωρίζει το συμβόλαιο από την υλοποίησή του. Ο διαχωρισμός αυτός προσφέρει «χαλαρές» σχέσεις ανάμεσα σε υπηρεσίες και πελάτες. Μια υλοποίηση της SOA είναι οι web services.

Η SOA επιτυγχάνει την αυξημένη διαθεσιμότητα, διαλειτουργικότητα, διατηρησιμότητα και αξιοπιστία των εφαρμογών ενός συστήματος. Τα πλεονεκτήματα αυτά οφείλονται από τη διάσπαση των εφαρμογών ενός συστήματος σε οντότητες (modules) με σαφώς ορισμένο συμβόλαιο διεπαφών (interface contract), που συμβάλλει σε χαλαρές σχέσεις εφαρμογών και υπηρεσιών. Η χαλαρή αυτή σχέση (loose coupling) ανάμεσα στον πελάτη και τον εξυπηρετητή ευνοεί τον πελάτη, καθώς οι εφαρμογές πελάτη προστατεύονται από αλλαγές στις υλοποιήσεις εξυπηρετητή, ενώ ο πελάτης μπορεί να επιλέξει ανάμεσα σε διάφορους παρόχους υπηρεσιών. Επίσης, ευνοείται ο πάροχος, καθώς με την αρχιτεκτονική αυτή προκύπτουν εφαρμογές που αντιστοιχούν σε επιχειρησιακές διαδικασίες. Η αρχιτεκτονική αυτή ευνοεί την επίτευξη αλλαγών πάνω σε εφαρμογές (modifiability), χωρίς να επηρεάζεται το συνολικό πληροφοριακό σύστημα. Στην Εικόνα 2.1 παρουσιάζεται η βασική φιλοσοφία της αρχιτεκτονικής SOA.

Επομένως, οι βασικές αρχές της αρχιτεκτονικής SOA είναι:

- Οι υπηρεσίες είναι αυτόνομες. Κάθε υπηρεσία αναπτύσσεται, εγκαθίσταται και συντηρείται ανεξάρτητα.
- Οι υπηρεσίες είναι κατανεμημένες. Μια υπηρεσία μπορεί να βρίσκεται οπουδήποτε στο διαδίκτυο.
- Οι υπηρεσίες 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).



Εικόνα 2.1 Η φιλοσοφία της αρχιτεκτονικής SOA.

Συνηθισμένα παραδείγματα υπηρεσιοκεντρικών εφαρμογών (service-oriented) περιλαμβάνουν εφαρμογές ανταλλαγής πληροφοριών, εφαρμογές με χειρισμό πολλαπλών βημάτων, όπως τα συστήματα κρατήσεων και τα ηλεκτρονικά καταστήματα κ.ά.

2.1.5 Αρθρωτή Αρχιτεκτονική (modular)

Αφορά στην ανάλυση ενός πληροφοριακού συστήματος σε επιμέρους υποσυστήματα (modules), όπου το καθένα εκτελεί μια συγκεκριμένη, σαφώς προσδιορισμένη λειτουργία. Η αρχιτεκτονική αυτή επιτρέπει την προσθήκη ή την αφαίρεση ενός υποσυστήματος, χωρίς να καταργείται το συνολικό σύστημα. Επίσης, η αρχιτεκτονική αυτή επιτρέπει την επέκταση κάθε υποσυστήματος (π.χ. με την προσθήκη νέων συσκευών), χωρίς να διαταράσσεται η λειτουργία του συνολικού συστήματος. Η αρθρωτή αρχιτεκτονική χρησιμοποιείται πολύ συχνά στην ανάπτυξη συστημάτων ERP, όπου κάθε υποσύστημα καλύπτει ένα σύνολο λειτουργικότητας.

2.1.6 Πολυεπίπεδη αρχιτεκτονική (n-tier architecture)

Η αρχιτεκτονική αυτή αναλύει το πληροφοριακό σύστημα σε επίπεδα, όπου κάθε επίπεδο αποτελεί μια λογική οντότητα και περιλαμβάνει συσκευές και εφαρμογές λογισμικού που εκτελούν μια συγκεκριμένη λειτουργία του συστήματος. Κάθε επίπεδο περιλαμβάνει διασυνδέσεις με τα υπόλοιπα επίπεδα. Τα επίπεδα μπορεί να βρίσκονται στην ίδια (λειτουργώντας σε εικονικές μηχανές ή με λογικές διακρίσεις) ή σε διαφορετικές μηχανές. Όσο περισσότερα επίπεδα χρησιμοποιούνται σε μια αρχιτεκτονική, τόσο πιο συγκεκριμένος είναι ο ρόλος που επιτελείται από κάθε επίπεδο.

Οι βέλτιστες πρακτικές ανάπτυξης πληροφοριακών συστημάτων αναλύουν μια πολυεπίπεδη αρχιτεκτονική στα ακόλουθα επίπεδα λειτουργιών:

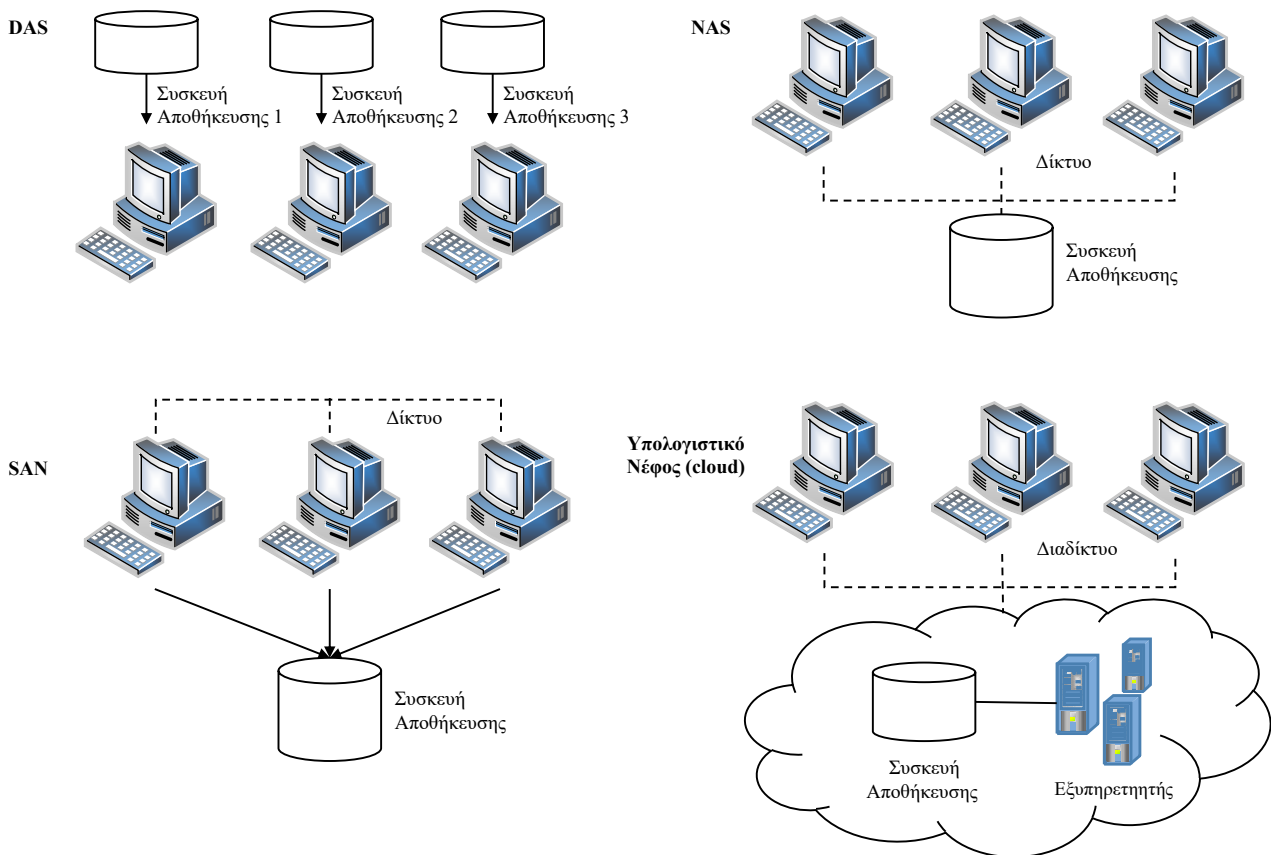
1. Επίπεδο παρουσίασης/χρηστών (presentation tier): Αφορά στις εφαρμογές στις οποίες έχουν πρόσβαση οι τελικοί χρήστες. Το επίπεδο περιλαμβάνει τα γραφικά περιβάλλοντα διεπαφών (UI) με τους τελικούς χρήστες καθώς και τις συσκευές που τα προσφέρουν, ενώ αναλύει τον τρόπο με τον οποίο δεδομένα και υπηρεσίες θα φτάνουν σε αυτούς.
2. Επιχειρησιακό Επίπεδο (business tier): Περιλαμβάνει τα υποσυστήματα που εκτελούν τις διάφορες επιχειρησιακές λειτουργίες που ζητούνται από το πληροφοριακό σύστημα (π.χ. διαχείριση δεδομένων, διαχείριση πελατειακών σχέσεων, επιχειρησιακή ευφυΐα κ.λπ.).
3. Επίπεδο διασύνδεσης (interconnection tier): Αφορά στις υποδομές και τις διεπαφές που διασυνδέουν τα υπόλοιπα επίπεδα του πληροφοριακού συστήματος μεταξύ τους.
4. Επίπεδο δεδομένων (data tier): Αφορά στις αποθήκες δεδομένων του πληροφοριακού συστήματος (συσκευές και μονάδες αποθήκευσης), στις βάσεις δεδομένων και στα δεδομένα του συστήματος (εγγραφές και αρχεία).

2.1.7 Αρχιτεκτονικές αποθήκευσης δεδομένων (storage architecture)

Ο όγκος δεδομένων μιας επιχείρησης αυξάνεται εκθετικά με τη χρήση πληροφοριακών συστημάτων αλλά και την ανάπτυξη του διαδικτύου. Τα κόστη που συνδέονται με τις αποθήκες των δεδομένων αφορούν στις ίδιες τις συσκευές που χρησιμοποιούνται για την αποθήκευση, στις μονάδες αποθήκευσης (π.χ. σκληροί δίσκοι, ταινίες κ.λπ.), αλλά και στις ανάγκες σε διαχείριση και επέκτασή τους.

Το κόστος σε αποθηκευτικές συσκευές συνήθως είναι «κρυφό», καθώς οι περισσότερες επιχειρήσεις αναβαθμίζουν συγχρόνως τις μηχανές και τις μονάδες αποθήκευσης. Το γεγονός αυτό επιφέρει πολλές μικρές αγορές που δεν αντανakλούν το συνολικό κόστος σε ανάγκες αποθήκευσης. Υπάρχουν τέσσερις αρχιτεκτονικές αποθήκευσης δεδομένων (βλέπε Εικόνα 2.2):

1. Απευθείας συσκευές αποθήκευσης (Direct Attached Storage: DAS): Πρόκειται για τη χρήση μονάδας αποθήκευσης (π.χ. σκληρός δίσκος) σε κάθε σταθμό εργασίας. Η λύση είναι ιδανική για περιπτώσεις όπου απαιτείται υψηλότατο επίπεδο ασφάλειας στη διαχείριση δεδομένων. Στις υπόλοιπες περιπτώσεις η λύση αυτή αντενδείκνυται, καθώς έχει σημαντικό κόστος διαχείρισης.
2. Δίκτυο αποθηκών (Storage Area Networks: SAN): Η λύση αυτή προσφέρει - θεωρητικά- απεριόριστο αποθηκευτικό χώρο, που εγκαθίσταται κεντρικά και είναι κεντρικά διαχειρίσιμος. Η λύση αφορά σε μια συσκευή (ή σε πολλές αντίστοιχες συσκευές διασυνδεδεμένες μεταξύ τους) με πολλαπλές μονάδες αποθήκευσης (πολλαπλοί δίσκοι), όπου η ανταλλαγή δεδομένων βασίζεται σε τεχνολογίες υψηλών ταχυτήτων με τη χρήση οπτικών διαύλων (FDDI). Επιπλέον, η λύση SAN επιτρέπει την τοποθέτηση ή αφαίρεση μιας συσκευής ή μιας μονάδας αποθήκευσης, χωρίς να επηρεαστεί η απόδοση ή η λειτουργία του πληροφοριακού συστήματος της επιχείρησης. Η λύση SAN είναι υψηλού κόστους, εφαρμόζεται όπου απαιτείται υψηλής ταχύτητας ανταλλαγής δεδομένων και όπου είναι δυνατή η διασύνδεση σημείων με υψηλές ταχύτητες, απαιτεί ιδιαίτερη ανάλυση στη σύνθεσή της, ενώ όταν πρόκειται για επιχείρηση με παραρτήματα σε διαφορετικές γεωγραφικές περιοχές απαιτούνται διαφορετικές αρχιτεκτονικές δεδομένων για κάθε σημείο ενδιαφέροντος.
3. Δικτυακή αποθήκη δεδομένων (Network Attached Storage: NAS): Πρόκειται για λύσεις ανάλογες των SAN, αλλά είναι χαμηλού κόστους και σημαντικά χαμηλότερης ταχύτητας. Οι NAS προσφέρουν αποκλειστικά αποθηκευτικό χώρο και δεν διαθέτουν κάποιον άλλο μηχανισμό διαχείρισης. Τα κόστη διαχείρισης των NAS είναι χαμηλά, ενώ η λύση ενδείκνυται σε περιπτώσεις που απαιτείται καταμερισμός των αποθηκευτικών χώρων σε διαφορετικές γεωγραφικές τοποθεσίες.
4. Αποθήκη δεδομένων στο cloud: Πρόκειται για λύση αποθήκευσης των δεδομένων στο υπολογιστικό νέφος. Η αρχιτεκτονική είναι ιδιαίτερα σύγχρονη και θα αναλυθεί παραπάνω στο κεφάλαιο 8.



Εικόνα 2.2 Αρχιτεκτονικές αποθήκευσης δεδομένων.

2.1.8 Αρχιτεκτονική Εφαρμογών (software architecture)

Όπως παρουσιάστηκε στην εισαγωγή της παρούσας ενότητας, μέρος του πληροφοριακού συστήματος είναι οι εφαρμογές. Κάθε εφαρμογή για να είναι κατανοητή στους εμπλεκόμενους χρήστες (σχεδιαστές, προγραμματιστές, συντηρητές, διαχειριστές, διευθυντές, τελικοί χρήστες) απαιτείται να ακολουθεί μια συγκεκριμένη αρχιτεκτονική. Η Αρχιτεκτονική Εφαρμογών (Software Architecture) είναι διακριτή από την αρχιτεκτονική του συστήματος, αλλά την επηρεάζει. Δεν υπάρχει ένας συγκεκριμένος ορισμός της αρχιτεκτονικής εφαρμογών. Ωστόσο, χρησιμοποιώντας ορισμούς από τη βιβλιογραφία θα λέγαμε ότι «η αρχιτεκτονική εφαρμογών είναι η δομή ή οι δομές μιας εφαρμογής που περιλαμβάνει τις προγραμματιστικές οντότητες (software components) που τη συνθέτουν, τις εξωτερικές ιδιότητες των οντοτήτων αυτών, καθώς και τη μεταξύ τους διασύνδεση. Η αρχιτεκτονική εφαρμογών περιλαμβάνει τις αποφάσεις για τη δομή ή τις δομές της εφαρμογής, καθώς και τις αλληλεπιδράσεις μεταξύ των δομών αυτών. Η αρχιτεκτονική αυτή καθορίζει την ανάπτυξη, την υποστήριξη και τη συντήρηση της εφαρμογής» (Bass, 2013).

Υπάρχουν ορισμένα πρότυπα (patterns) για την αρχιτεκτονική εφαρμογών:

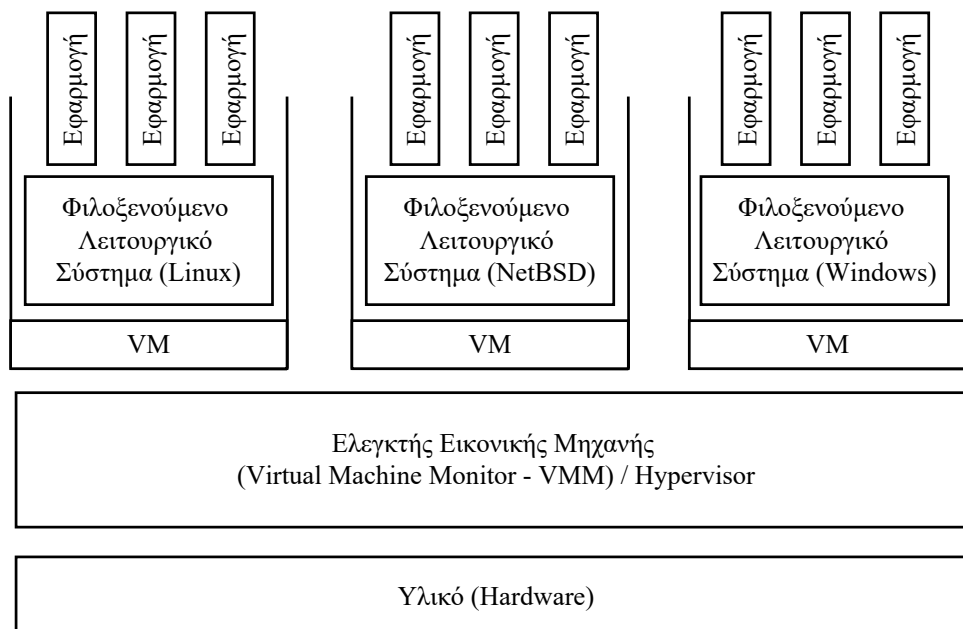
1. Μοντέλο-Απεικόνιση-Ελεγκτής (Model-View-Controller - MVC): Το Model-View-Controller (σε συντομογραφία αναφέρεται ως MVC) είναι ένα μοντέλο αρχιτεκτονικής λογισμικού το οποίο χρησιμοποιείται για τη δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη. Στο μοντέλο αυτό, η εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη, ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στον χρήστη από τη μορφή που έχει αποθηκευτεί στο σύστημα. Το κύριο μέρος του μοντέλου είναι το Μοντέλο (Model) το οποίο διαχειρίζεται την ανάκτηση/αποθήκευση των δεδομένων στο σύστημα. Το αντικείμενο Απεικόνιση (View) χρησιμοποιείται μόνο για να παρουσιάζεται η πληροφορία στον χρήστη (π.χ. με γραφικό τρόπο). Το τρίτο μέρος είναι ο Ελεγκτής (Controller), ο οποίος δέχεται την είσοδο και στέλνει εντολές στο αντικείμενο Model και στο View (Deacon, 2009).
2. Εκδίδω-εγγράφομαι (Publish-subscribe): Πρόκειται για ένα πρότυπο, στο οποίο ο εκδότης (publisher) δημοσιεύει δεδομένα σε έναν δίαυλο. Οι εγγεγραμμένες οντότητες (subscribers) εγγράφονται (subscribe) εκδηλώνοντας το ενδιαφέρον τους για τα δεδομένα αυτά και ειδοποιούνται για την παραλαβή των σχετικών μηνυμάτων όταν υπάρχουν διαθέσιμα μηνύματα που τους ενδιαφέρουν (Eugster, 2003).

3. Σωλήνες και φίλτρα (Pipes and filters): Όταν απαιτείται εκτέλεση εργασιών με σημαντική πολυπλοκότητα, μια λύση είναι η ανάλυση του προβλήματος σε μια σειρά βημάτων επεξεργασίας τα οποία ονομάζονται φίλτρα (filters). Κάθε φίλτρο έχει μια είσοδο και μια έξοδο. Τα φίλτρα συνθέτουν μια αλυσίδα, όπου το καθένα λαμβάνει δεδομένα από το προηγούμενο και δίνει δεδομένα στο επόμενο (Homer et al., 2014).
4. Επίπεδα (layers): Στο πρότυπο αυτό η εφαρμογή αναλύεται σε επίπεδα (layers), όπου κάθε επίπεδο αναλαμβάνει μια συγκεκριμένη λειτουργία της εφαρμογής. Κάθε επίπεδο διαθέτει εξαρτήσεις (dependencies) και διεπαφές (interfaces) προς τα άλλα επίπεδα. Διακρίνονται σε χαλαρές αρχιτεκτονικές επιπέδων (relaxed layered architectures) και σε αυστηρές αρχιτεκτονικές επιπέδων (strict layered architectures). Στις τελευταίες κάθε επίπεδο εξαρτάται αποκλειστικά και μόνο από το αμέσως χαμηλότερο επίπεδο, ενώ στις πρώτες τα ανώτερα επίπεδα μπορούν να χρησιμοποιούν απευθείας τις υπηρεσίες όλων των επιπέδων που βρίσκονται χαμηλότερα από αυτά. Στα πληροφοριακά συστήματα είναι πολύ συνηθισμένη μία χαλαρή αρχιτεκτονική επιπέδων.

2.1.9 Αρχιτεκτονική με εικονικές μηχανές (virtualization)

Πρόκειται για μια μοντέρνα προσέγγιση στην υλοποίηση πληροφοριακών συστημάτων. Η παραδοσιακή υλοποίηση πληροφοριακών συστημάτων προβλέπει -ανεξάρτητα από την επιλεγείσα αρχιτεκτονική- τη χρήση φυσικών μηχανών για τους ρόλους των εξυπηρετητών (servers) που απαιτούνται στο σύστημα. Χρησιμοποιείται στις περιπτώσεις πολύπλοκων πληροφοριακών συστημάτων που αναλύονται σε πολλαπλά επίπεδα (tiers) και περιλαμβάνουν συστοιχίες (clusters) εξυπηρετητών για να καλύψουν συνθήκες υψηλής διαθεσιμότητας.

Η αρχιτεκτονική με εικονικές μηχανές χρησιμοποιεί τους πόρους μιας ή περισσοτέρων φυσικών μηχανών εξυπηρετητή, καθώς και κατάλληλο λογισμικό, ώστε στην ίδια φυσική μηχανή να λειτουργούν εικονικές μηχανές εξυπηρετητών του ίδιου ή διαφορετικών επιπέδων ή/και υποσυστημάτων του πληροφοριακού συστήματος. Στην περίπτωση αυτή, ο εξυπηρετητής-φυσική μηχανή ονομάζεται «Hypervisor», ενώ στην αρχιτεκτονική του πληροφοριακού συστήματος ενσωματώνεται ένα πρόσθετο επίπεδο που αφορά στο λογισμικό διαχείρισης των εικονικών μηχανών του συστήματος. Μια συνηθισμένη τέτοιου είδους αρχιτεκτονική παρουσιάζεται στην Εικόνα 2.3.



Εικόνα 2.3 Αρχιτεκτονική εικονικών μηχανών.

Η αρχιτεκτονική με εικονικές μηχανές προσφέρει σημαντικά πλεονεκτήματα στην επιχείρηση και συγκεκριμένα:

1. Περιορισμό του κόστους του πληροφοριακού συστήματος. Σε ένα πολύπλοκο πληροφοριακό σύστημα που απαιτεί συστοιχίες εξυπηρετητών ανά επίπεδο, το κόστος υλοποίησής του με εικονικές μηχανές είναι σημαντικά χαμηλότερο σε σχέση με την υλοποίησή του με φυσικές μηχανές. Δεν θα πρέπει όμως

να υποτιμούμε το κόστος των κεντρικών μηχανών (hypervisors), καθώς θα πρέπει να διαθέτουν αρκετούς πόρους (μνήμη, υπολογιστική ισχύς) για να φιλοξενήσουν μεγάλο αριθμό εικονικών εξυπηρετητών, καθώς επίσης και το κόστος του λογισμικού διαχείρισης των εικονικών μηχανών (π.χ. VMWare, Microsoft Virtual Machine).

2. Περιορισμό του φυσικού όγκου του συστήματος.
3. Ευκολότερη διαχείριση του συνολικού συστήματος, που επιτυγχάνεται μέσω της κονσόλας του λογισμικού διαχείρισης των εικονικών μηχανών.
4. Ευκολότερη επέκταση του πληροφοριακού συστήματος (π.χ. η προσθήκη ή αναβάθμιση ενός εικονικού εξυπηρετητή πραγματοποιείται μέσα σε λίγα λεπτά χωρίς την αναβάθμιση της εικονικής μηχανής).
5. Ευκολότερη αναβάθμιση του συνολικού πληροφοριακού συστήματος. Η μετάπτωση των εικονικών μηχανών από έναν hypervisor σε έναν άλλο γίνεται άμεσα, ενώ οι εικονικές μηχανές λειτουργούν ανεξάρτητα του εξοπλισμού (hardware) του hypervisor.
6. Ευκολότερη διαχείριση καταστροφών (κάθε εικονική μηχανή είναι ένα αρχείο στον hypervisor, που λαμβάνεται εύκολα ως αντίγραφο ασφαλείας και μπορεί να ανακτηθεί σε οποιαδήποτε νέα μηχανή).

2.1.10 Αρχιτεκτονική του υπολογιστικού νέφους (cloud computing)

Υπολογιστικό Νέφος (ΥΝ) ονομάζεται η μετά από αίτηση διαδικτυακή κεντρική διάθεση υπολογιστικών πόρων (όπως δίκτυο, εξυπηρετητές, εφαρμογές και υπηρεσίες) με υψηλή ευελιξία, ελάχιστη προσπάθεια από τον χρήστη και υψηλή αυτοματοποίηση (NIST, 2011).

Λόγω του ενδιαφέροντος που παρουσιάζει αυτή η προσέγγιση θα παρουσιαστεί αναλυτικά στο Κεφάλαιο 8.

2.2 Η πληροφοριακή υποδομή της επιχείρησης

Τα υποσυστήματα που ανήκουν σε αυτή την κατηγορία λειτουργούν υποστηρικτικά και οριζοντίως στα λειτουργικά υποσυστήματα που περιγράφονται παραπάνω, παρέχοντας την πληροφοριακή υποδομή που απαιτείται για την εκτέλεση των υπηρεσιών που προσφέρουν στους χρήστες. Οι υπηρεσίες που παρέχονται μπορούν να κατηγοριοποιηθούν ενδεικτικά σε γενικές κατηγορίες (Heilig et al., 2007):

- Υποσύστημα διαχείρισης ταυτότητας χρηστών,
- Υποσύστημα διαχείρισης δικτυακής πύλης,
- Υποσύστημα διαχείρισης ροής εργασιών,
- Υποσύστημα διαχείρισης εγγράφων,
- Υποσυστήματα ανάπτυξης/προσαρμογής λογισμικού.

Στις επόμενες παραγράφους θα παρουσιάσουμε εν συντομία τη λειτουργικότητα αυτών των υποσυστημάτων.

2.2.1 Υποσυστήματα διαχείρισης ταυτότητας χρηστών (identity management)

Το υποσύστημα της διαχείρισης ταυτότητας χρηστών αποτελεί ένα διακριτό υποσύστημα που υπάρχει στο σιλό των εφαρμογών όλων των κατασκευαστών πληροφοριακών συστημάτων επιχειρήσεων. Αφορά τη διαχείριση του τρόπου πρόσβασης, των ρόλων, της διαθέσιμης πληροφορίας αλλά και της λειτουργικότητας που θα είναι διαθέσιμα για κάθε χρήστη του συστήματος/υποσυστήματος. Τα σύγχρονα περιβάλλοντα ανάπτυξης λογισμικού προσφέρουν έναν ενιαίο τρόπο πρόσβασης στα διαθέσιμα πληροφοριακά συστήματα των επιχειρήσεων ανεξαρτήτως του είδους του χρήστη (εσωτερικός ή εξωτερικός χρήστης) ή αν βασίζεται σε διαδικτυακές τεχνολογίες. Επιπλέον, η ταυτοποίηση των χρηστών γίνεται με μοναδικό και ενιαίο τρόπο και ανεξάρτητα από το πληροφοριακό σύστημα που θέλει να χρησιμοποιήσει.

Ο συνηθέστερος τρόπος για τη διαχείριση των ταυτοτήτων των χρηστών των πληροφοριακών συστημάτων είναι με τη χρήση του πρωτοκόλλου LDAP (Lightweight Directory Access Protocol). Το LDAP είναι ένα πρωτόκολλο ανοικτού προτύπου για την πρόσβαση σε υπηρεσίες καταλόγου τύπου X.500. Οι υπηρεσίες καταλόγου είναι μία βάση δεδομένων η οποία οργανώνει εγγραφές και είναι βελτιστοποιημένη για διαδικασίες ανάγνωσης και αναζήτησης για τις ηλεκτρονικές υπηρεσίες μιας επιχείρησης. Ο εξυπηρετητής που φιλοξενεί την υπηρεσία LDAP είναι υπεύθυνος για την ταυτοποίηση και είσοδο των χρηστών σε διάφορες ηλεκτρονι-

κές υπηρεσίες που θα παρέχονται από την επιχείρηση.

Τα δικαιώματα μπορούν να αφορούν τις δυνατότητες διαμόρφωσης του περιεχομένου του δικτυακού τόπου της επιχείρησης (Portal), αλλά επιπλέον δίνεται και η δυνατότητα απόδοσης συγκεκριμένων δικαιωμάτων εξειδικευμένα σε σελίδες ακόμη και αντικείμενα του δικτυακού τόπου όπως έγγραφα, εικόνες, κ.λπ. Τα είδη δικαιωμάτων για αντικείμενα (π.χ. έγγραφα) μπορεί να είναι:

- Own item: Πλήρης έλεγχος αντικειμένου και δικαιωμάτων σε αυτό,
- Manage item: Αλλαγή ή διαγραφή αντικειμένου,
- View item: Ανάγνωση εγγράφου.

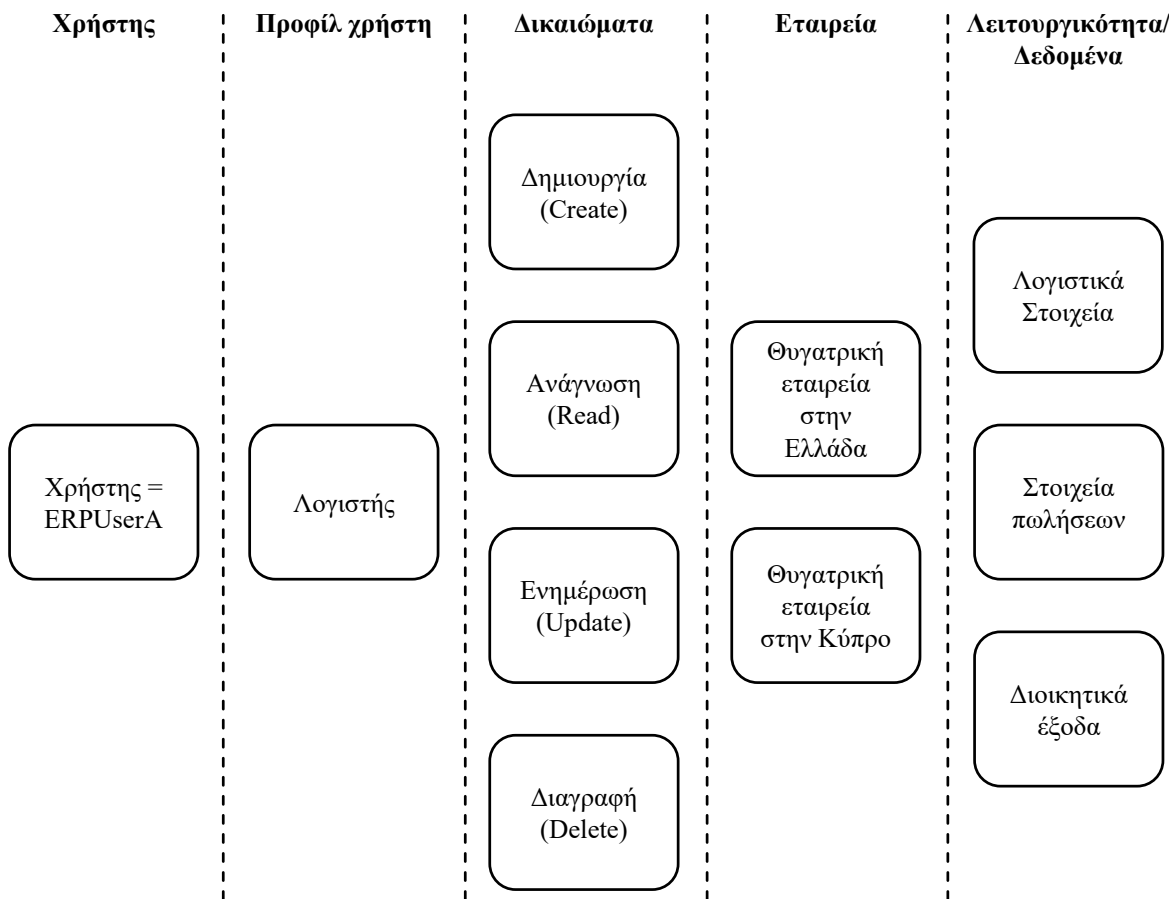
Στην ίδια υπηρεσία αποθηκεύονται και τα δικαιώματα των χρηστών όσον αφορά τις υπόλοιπες εφαρμογές της υποδομής του πληροφοριακού συστήματος, ώστε με κεντρική εισαγωγή των στοιχείων τους, οι χρήστες να αποκτούν πρόσβαση όπου έχουν δικαιώματα (δικτυακές εφαρμογές και παραδοσιακές εφαρμογές).

Για την περίπτωση των συστημάτων ERP η οργάνωση των δικαιωμάτων των χρηστών είναι αρκετά πιο σύνθετη αφού μπορεί να αφορά:

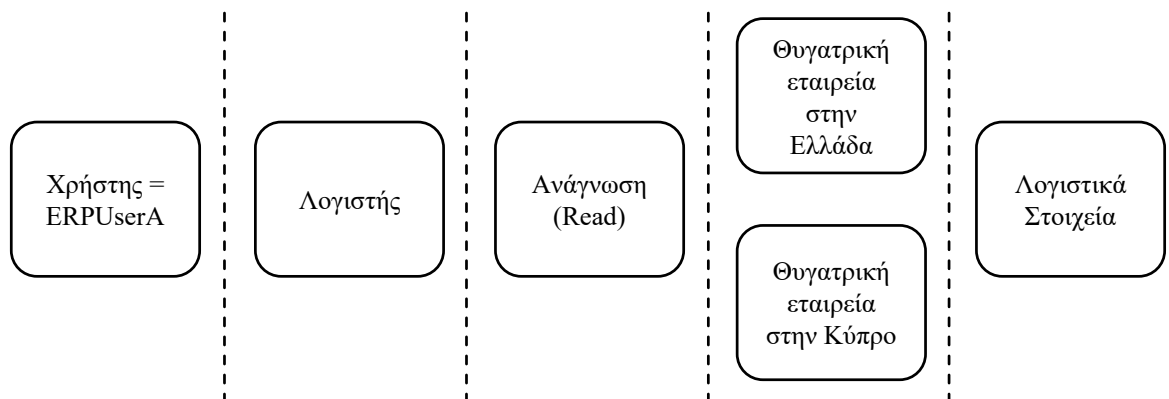
- Πρόσβαση σε λειτουργίες του συστήματος ERP (function access control): Αφορά τον ορισμό των λειτουργιών που θα είναι διαθέσιμες σε κάθε χρήστη. Για παράδειγμα, ποια μενού θα είναι διαθέσιμα σε έναν πωλητή και ποια θα είναι διαθέσιμα για έναν διευθυντή τμήματος πωλήσεων;
- Πρόσβαση σε δεδομένα του συστήματος ERP (data access control): Αφορά τον ορισμό των οντοτήτων που θα είναι διαθέσιμες σε κάθε χρήστη. Για παράδειγμα, ποιες οντότητες θα είναι διαθέσιμες για ανάγνωση, για τροποποίηση και για διαγραφή, σε έναν πωλητή και ποιες θα είναι διαθέσιμες για έναν διευθυντή τμήματος πωλήσεων;
- Πρόσβαση σε δεδομένα του συστήματος ERP (data access control): Αφορά τον ορισμό των οντοτήτων που θα είναι διαθέσιμες σε κάθε χρήστη. Για παράδειγμα, ποιες οντότητες θα είναι διαθέσιμες για ανάγνωση, για τροποποίηση και για διαγραφή, σε έναν πωλητή και ποιες θα είναι διαθέσιμες 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 στις λειτουργίες του συστήματος ERP. 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.

Οι παραπάνω πολιτικές πρόσβασης αφορούν τους βασικούς χρήστες των συστημάτων ERP. Ανάλογα, μπορούμε να ορίσουμε πολιτικές πρόσβασης για τους διαχειριστές των συστημάτων (administrators).

Ένα ενδεικτικό παράδειγμα πολιτικής πρόσβασης χρηστών δίνεται στην Εικόνα 2.4.



Χρήστης με πλήρη δικαιώματα



Χρήστης με μειωμένα δικαιώματα

Εικόνα 2.4 Δικαιώματα χρηστών.

Επιπλέον, το υποσύστημα της διαχείρισης ταυτότητας χρηστών επιτρέπει την κεντροκοιμημένη διαχείριση των χρηστών όλης της υποδομής. Η κεντροκοιμημένη διαχείριση των χρηστών όλης της υποδομής υλοποιείται με την πολιτική Single Sign-On που είναι ο μηχανισμός κεντρικής αυθεντικοποίησης και εξουσιοδότησης των χρηστών. Αναλαμβάνει τον έλεγχο της ταυτότητας των χρηστών και τους εξουσιοδοτεί με πρόσβαση στον δικτυακό τόπο αλλά και σε όλες τις εφαρμογές και δεδομένα στα οποία έχουν αντίστοιχα δικαιώματα. Οι τρόποι αυθεντικοποίησης είναι πολλοί και μπορεί να είναι:

- Χρήση usernames/passwords (βασική αυθεντικοποίηση),
- Χρήση X.509 ψηφιακών πιστοποιητικών (digital certificates) σε SSL σύνδεση, μέσω κρυπτογραφίας δημόσιου κλειδιού (Public Key Infrastructure -PKI),
- Έλεγχος της IP διεύθυνσης και/ή του domain name του καλούντος σε σχέση με λίστες πρόσβασης σε αρχεία, καταλόγους και γενικότερα URLs μέσω της δυνατότητας που παρέχει ο HTTP Server (με χρήση παραμέτρων allow και deny directives),

- Αυτόματη αυθεντικοποίηση χρήστη με Single Sign-On: Με το ενσωματωμένο χαρακτηριστικό Single Sign-On (SSO) προσφέρεται η δυνατότητα αυτόματης αυθεντικοποίησης των χρηστών, εφόσον έχουν εισάγει τα στοιχεία τους μια φορά στο σύστημα. Ο μηχανισμός SSO βασίζεται στην τεχνολογία των cookies, προκειμένου να ελεγχθεί αν ο χρήστης που ζητά πρόσβαση σε εφαρμογές του συστήματος, έχει ήδη αυθεντικοποιηθεί. Στην περίπτωση που όντως έχει προηγουμένως αυθεντικοποιηθεί και το συγκεκριμένο cookie δεν έχει λήξει τότε επιτρέπει την πρόσβαση σε εφαρμογές του συστήματος για τις οποίες έχει εξουσιοδότηση, χωρίς να ζητήσει πάλι την ταυτότητά του (username/password).

Οι ανάγκες για τους παραπάνω μηχανισμούς είναι ιδιαίτερα σημαντικές σήμερα, διότι οι χρήστες έχουν πρόσβαση στα ίδια συστήματα και στην ίδια πληροφορία μέσω διαφορετικών μηχανισμών-καναλιών. Για παράδειγμα, μπορούμε να αναφέρουμε τις εφαρμογές που χρησιμοποιούν τα στοιχεία και τους μηχανισμούς αυθεντικοποίησης του google ή του facebook ώστε να δώσουν άμεση πρόσβαση στο πληροφοριακό σύστημα χωρίς να χρειαστεί ο πελάτης να επανεισάγει τα στοιχεία του.

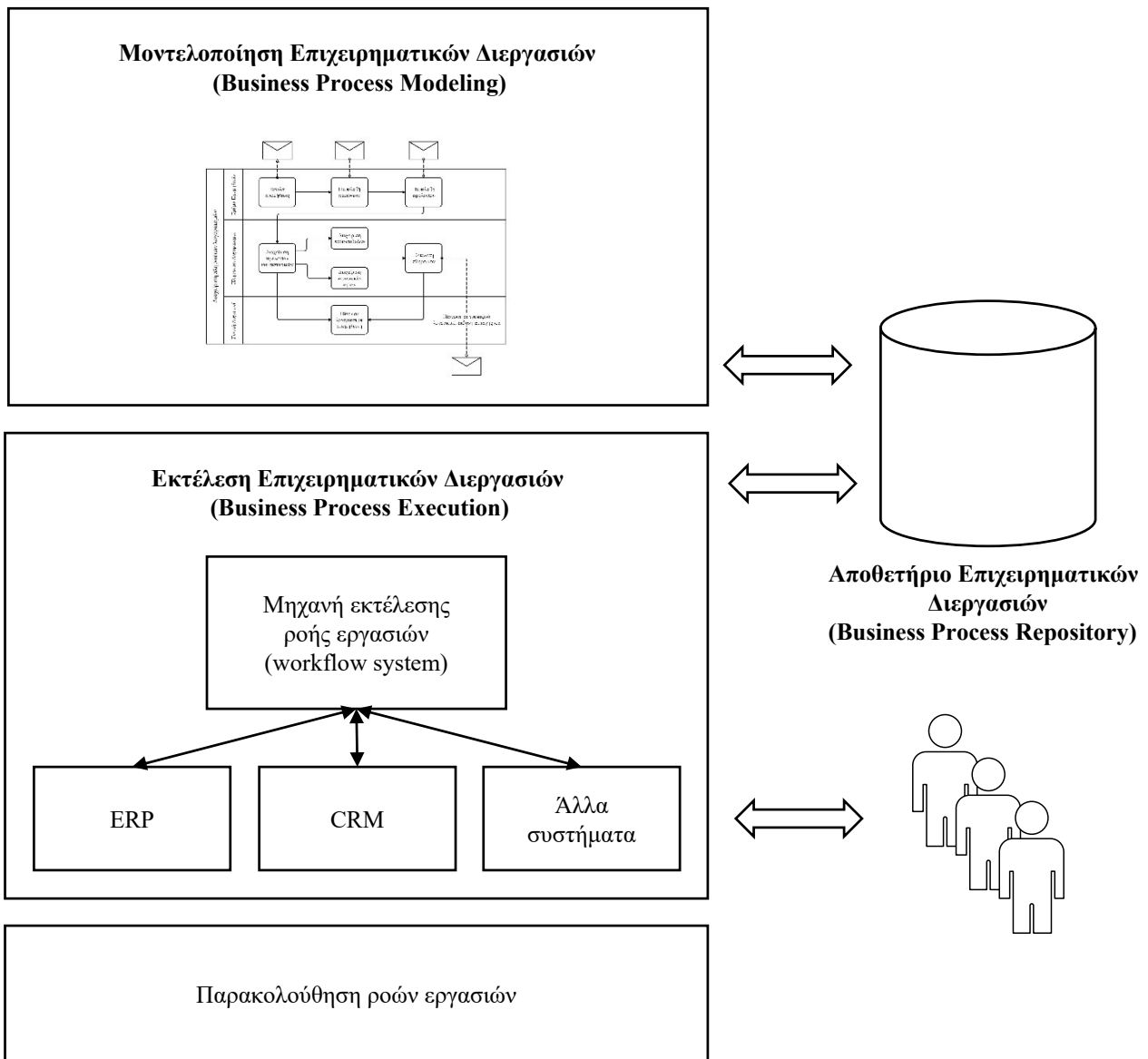
2.2.2 Υποσύστημα διαχείρισης ροής εργασιών (workflow management)

Η ολοκλήρωση των επιχειρηματικών διεργασιών (business process) αποτελεί ακρογωνιαίο λίθο επιτυχίας σε όλα τα σύγχρονα περιβάλλοντα μεγάλων επιχειρήσεων. Η ολοκληρωμένη διαχείριση εφαρμογών χρησιμοποιείται για την ολοκλήρωση των επιμέρους εφαρμογών σε ένα ενιαίο πληροφοριακό σύστημα και θα πρέπει να βασίζεται σε έναν γενικευμένο μηχανισμό ανταλλαγής δεδομένων. Επιπλέον, η ολοκλήρωση και ο ορισμός των διαδικασιών θα πρέπει να γίνεται με ένα εργαλείο παραμετρικού ορισμού διεργασιών. Το θέμα του ανασχεδιασμού των επιχειρηματικών διεργασιών αποτελεί το θέμα που αναλύουμε στο κεφάλαιο 5.

Αντίστοιχα, η διαχείριση και η εκτέλεση των εργασιών γίνεται μέσω του υποσυστήματος ροής εργασιών (workflow management system). Η συνολική αρχιτεκτονική ενός τέτοιου υποσυστήματος παρουσιάζεται στην Εικόνα 2.5. Τρία είναι τα βασικά υποσυστήματα: α) το υποσύστημα ορισμού των εργασιών, β) το υποσύστημα εκτέλεσης των εργασιών και γ) το αποθετήριο επιχειρηματικών διεργασιών.

Στις παραγράφους που ακολουθούν παρουσιάζεται εν συντομία η απαιτούμενη λειτουργικότητα από ένα τέτοιο υποσύστημα ροής εργασιών (workflow) (Snabe et al., 2008). Ένα τέτοιο υποσύστημα πρέπει να υποστηρίζει τον συντονισμό, τη δρομολόγηση και την εποπτεία όλων των επιχειρησιακών διαδικασιών που εκτελούνται στα πλαίσια των λειτουργικών υποσυστημάτων, παρέχοντας:

- Τη δυνατότητα σχεδιασμού σεναρίων ροής εργασιών και παρουσίασης/διεκπεραίωσης της τρέχουσας διαδικασίας με γραφικό (διαγραμματικό) τρόπο.
- Ολοκλήρωση με τα άλλα υποσυστήματα πληροφοριακής υποδομής.
- Τη δυνατότητα ανάθεσης ενέργειας με ή χωρίς την απαίτηση αντίδρασης από τον χρήστη.
- Τη δρομολόγηση ενεργειών και εγγράφων σε μεμονωμένους χρήστες αλλά και σε ομάδες ορίζοντας προθεσμία διεκπεραίωσης.
- Την εμφάνιση των εκκρεμούντων ανατεθεισών ενεργειών στον χρήστη με γραφικό τρόπο ως εγγραφές στο γραφικό περιβάλλον υποδοχής εργασιών του. Ενδεικτικά, για κάθε εργασία θα εμφανίζεται ο χρήστης που ανέθεσε την ενέργεια, η ημερομηνία ανάθεσης, η προθεσμία διεκπεραίωσης, το όνομα της ενέργειας και άλλες χρήσιμες πληροφορίες για την ενέργεια αυτή.
- Την αυτόματη αποστολή email στον χρήστη που του έχει ανατεθεί μια εργασία και θα τον ενημερώνει σχετικά.



Εικόνα 2.5 Υποσύστημα ροής εργασιών.

- Τη δυνατότητα συνοδείας μιας νέας εργασίας στο γραφικό περιβάλλον υποδοχής εργασιών του χρήστη από ένα ή περισσότερα έγγραφα, οδηγίες ή παρατηρήσεις καθώς και μεταβλητές και δεδομένα που θα προέρχονται είτε από τη βάση δεδομένων είτε από την ίδια την εκτέλεση της ροής εργασίας εφόσον υπάρχουν, κατά τη σχεδιάσή της, βήματα στα οποία ο χρήστης καλείται να εισάγει τιμές σε μεταβλητές.
- Τη σχεδίαση και τον προγραμματισμό των διαδικασιών με έναν πλήρως γραφικό τρόπο (visual design) χωρίς την ανάγκη εκμάθησης ιδιαίτερης γλώσσας προγραμματισμού από τον χρήστη.
- Τη δυνατότητα επανασχεδίασης μιας διαδικασίας χωρίς να απαιτείται διακοπή των διαδικασιών που βρίσκονται σε εξέλιξη, ώστε να μην επηρεάζονται κρίσιμες λειτουργίες.
- Ανάθεση εργασίας σε χρήστες με τρόπους όπως φόρτος εργασίας χρηστών, συγκεκριμένος χρήστης και τυχαία επιλογή διαθέσιμου χρήστη.
- Τη δυνατότητα αυτόματων εγκρίσεων ή αυτόματων απορρίψεων μετά από ένα καθορισμένο χρονικό διάστημα.
- Τη δυνατότητα χρήσης χρονολογημένων γεγονότων (timed events), τα οποία προκύπτουν όταν εκπληρώνονται κάποια κριτήρια σε καθορισμένους χρόνους κατά τη διάρκεια του ενεργού βήματος μιας ροής εργασίας. Ένα τέτοιο γεγονός μπορεί να σηματοδοτεί την αρχή μιας νέας ροής εργασίας ή την εισαγωγή επιπλέον βημάτων ή την αυτόματη ενεργοποίηση κάποιου προγράμματος.
- Αυτόματη ενεργοποίηση συστημάτων λογισμικού (π.χ. υποσυστημάτων διαχείρισης εγγράφων, επεξεργαστές κειμένων, spreadsheets, ActiveX objects, COM, static DLL, κ.ο.κ.) ή/και ανταλλαγή δεδομένων

με αυτά, ως συγκεκριμένη απαιτητή λειτουργία μετά την ικανοποίηση συνθήκης ή/και την εμφάνιση συγκεκριμένων γεγονότων (events) στο ενεργό βήμα των σεναρίων ροής εργασιών.

- Εργαλείο σχεδίασης ηλεκτρονικών φορμών που θα πρέπει να έχει τουλάχιστον τις ακόλουθες δυνατότητες:
 - ο Πλήρως παραμετρικές ροές βάσει χαρακτηριστικών των χρηστών, των δεδομένων καθώς επίσης και των εκάστοτε συνθηκών.
 - ο Τη δυνατότητα χειροκίνητης ή αυτόματης εκκίνησης μιας διαδικασίας.
 - ο Τη δυνατότητα ηλεκτρονικής υπογραφής συγκεκριμένων βημάτων μιας ροής εργασίας ή ενός εγγράφου που συνοδεύει κάποιο βήμα ροής εργασίας.
 - ο Τη δυνατότητα χρήσης προτύπων όπως η XML για την περιγραφή ανταλλάσσιμων δεδομένων ή/και εγγράφων διευκολύνοντας έτσι την ανταλλαγή δεδομένων αυτού του τύπου με άλλα παρόμοια συστήματα ή συναφείς τρίτες εφαρμογές.
 - ο Τη δυνατότητα διαχείρισης εκδόσεων των ροών εργασίας.
 - ο Τη δυνατότητα ελέγχου της ροής εκτέλεσης των εργασιών (π.χ. σε ποιο βήμα βρίσκεται η διαδικασία) αλλά και απολογιστικής στατιστικής ανάλυσης του χρόνου διεκπεραίωσης των διαδικασιών (π.χ. απαιτούμενος χρόνος ανά βήμα της διαδικασίας ή χρόνος/βήματα που απομένουν κ.λπ.).

2.2.3 Υποσύστημα διαχείρισης εγγράφων (document management)

Η αποτελεσματική διαχείριση εγγράφων αποτελεί πρόκληση για τους σύγχρονους φορείς και επιχειρήσεις, διότι σύμφωνα με μελέτες έγκυρων οργανισμών αποδεικνύουν ότι περίπου το 90% της επιχειρησιακής πληροφορίας βρίσκεται σε αδόμητη μορφή (κυρίως αποθηκευμένη σε χαρτί ή ηλεκτρονική μορφή). Επομένως, ο τρόπος συλλογής, ταξινόμησης και επεξεργασίας των εγγράφων επηρεάζει σε μεγάλο βαθμό τη λειτουργία μιας επιχείρησης. Όταν οι εργασίες αρχειοθέτησης πραγματοποιούνται με τον παραδοσιακό «χειροκίνητο» και χειρόγραφο τρόπο, τότε παρουσιάζονται προβλήματα όπως: δυσκολία στην αναζήτηση, καθυστερημένη διανομή των εγγράφων, δυσκολία της παρακολούθησης του εγγράφου, ακόμη και απώλειες εγγράφων. Αντίστοιχα, η εγκατάσταση και εφαρμογή της ηλεκτρονικής διαχείρισης εγγράφων σε επιχειρήσεις ή οργανισμούς παρέχει σημαντικά πλεονεκτήματα όπως:

- Σημαντική αύξηση της παραγωγικότητας,
- Ταχύτερη και περισσότερο ποιοτική εξυπηρέτηση των πελατών και συνεργατών λόγω της εύκολης πρόσβασης στις πηγές πληροφόρησης,
- Οργανωμένο και ασφαλή έλεγχο εγγράφων και πληροφοριών,
- Εξοικονόμηση χώρου αποθήκευσης των εγγράφων λόγω καλής συμπίεσης των ηλεκτρονικών δεδομένων.

Επομένως, το υποσύστημα διαχείρισης εγγράφων αποτελεί βασικό υποστηρικτικό σύστημα το οποίο σε συνεργασία με το υποσύστημα ροής εργασιών υποστηρίζει την κεντρική διαχείριση όλων των εγγράφων που δημιουργούνται κατά την εκτέλεση των διαφόρων διαδικασιών των λειτουργικών υποσυστημάτων, παρέχοντας (Stajda, 2013):

- Υποστήριξη διαφόρων μορφών (formats) εγγράφων, όπως για παράδειγμα OASIS, Open Document Format, HTML, Word, Excel, PowerPoint, Adobe Acrobat PDF, κ.ά., καθώς και τη δυνατότητα εμφάνισης των εγγράφων μέσα από ενσωματωμένους viewers ώστε να μη βασίζεται μόνο σε τρίτες εφαρμογές.
- Δυνατότητα εισαγωγής εγγράφου από διάφορες πηγές. Για παράδειγμα, θα πρέπει να υποστηρίζεται η εισαγωγή εγγράφων από σαρωτές κειμένου και να υποστηρίζεται η οπτική αναγνώριση εγγράφων όπου απαιτείται (Optical Character Recognition - OCR).
- Ολοκλήρωση με συνθήεις εφαρμογές αυτοματισμού γραφείου (π.χ. MS Office, Open Office). Η ολοκλήρωση θα πρέπει να γίνεται με χρήση ανοικτών προτύπων όπως:
 - ο WebDAV (Web Distributed Authoring and Versioning) που είναι μια επέκταση του πρωτοκόλλου HTTP, που επιτρέπει τη συνεργατική επεξεργασία εγγράφων μέσω του διαδικτύου.
 - ο CMIS (Content Management Interoperability Services) που επιτρέπει τη συνεργασία συστημάτων διαχείρισης περιεχομένου μέσω του διαδικτύου.
 - ο SOAP (Simple Object Access Protocol) που είναι ένα πρωτόκολλο ανταλλαγής πληροφορίας μεταξύ δικτυακών υπηρεσιών (web services) με τη χρήση της γλώσσας XML.

- Συνεργατική προσπέλαση των εγγράφων παρέχοντας το κλείδωμα ενός εγγράφου από διαγραφή/διαμόρφωσή του από άλλο χρήστη.
 - Διαχείριση εκδόσεων εγγράφων (version control), παρέχοντας μηχανισμούς ελέγχου και δημιουργίας εκδόσεων που θα μπορούν:
 - ο Να διαχειρίζονται πολλαπλές εκδόσεις ενός εγγράφου,
 - ο Να παρέχουν δυνατότητα αυτόματης δημιουργίας μιας νέας έκδοσης ενός εγγράφου μόλις αυτό αλλάξει,
 - ο Να επιτρέπουν την εισαγωγή της περιγραφής της φύσης των αλλαγών (track changes) που γίνονται καθώς και τα σημεία των αλλαγών αυτών σε κάθε νέα έκδοση ενός εγγράφου, καθώς και
 - ο Τη δυνατότητα της παρουσίασης της λίστας ιστορικότητας πρόσβασης και εκδόσεων για το κάθε έγγραφο, όπου θα παρέχονται πληροφορίες όπως συγγραφέας, ημερομηνία κ.λπ.
 - Υποστήριξη μεταδεδομένων (metadata) για κάθε έγγραφο. Τα μεταδεδομένα είναι δεδομένα περιγραφής βασικών χαρακτηριστικών των εγγράφων που διευκολύνουν την ταξινόμηση και την αναζήτηση των εγγράφων. Ορίζονται με βάση τις ανάγκες του κάθε συστήματος αλλά και γενικότερα της επιχείρησης. Για παράδειγμα, σε ένα σύστημα διαχείρισης τεχνικών εγγράφων τα μεταδεδομένα μπορεί να αναφέρονται στο προϊόν το οποίο σχετίζεται με το έγγραφο, σε ένα σύστημα διαχείρισης έργων, τον κωδικό της δραστηριότητας που σχετίζεται με το έγγραφο κ.λπ. Ως γενική κατεύθυνση τα μεταδεδομένα περιγράφουν: τον τρόπο δημιουργίας του εγγράφου, τον σκοπό του εγγράφου, τον χρόνο δημιουργίας του, τον συγγραφέα, τη θέση αποθήκευσης του εγγράφου κ.ά. Επομένως, ένα υποσύστημα που υποστηρίζει μεταδεδομένα στα έγγραφα θα πρέπει να παρέχει:
 - ο Τη δυνατότητα ανάθεσης σε κάθε νέο έγγραφο που αποθηκεύεται ένα σύνολο από πληροφορίες-μεταδεδομένα όπως όνομα, ώρα δημιουργίας, όνομα δημιουργού κ.λπ.
 - ο Τη δημιουργία προκαθορισμένων (predefined) μεταδεδομένων που θα παίρνουν τιμή από το σύστημα όπως π.χ. ημερομηνία και ώρα δημιουργίας ενός εγγράφου.
 - ο Τη δυνατότητα ανάθεσης τιμών σε μεταδεδομένα τα οποία είναι ορισμένα από τους χρήστες/διαχειριστές του υποσυστήματος και έχουν να κάνουν με τον τρόπο που οργανώνονται και διαχειρίζονται τα έγγραφα στην επιχείρηση π.χ. ο τύπος του εγγράφου (αίτηση, εσωτερικό σημείωμα, e-mail, φαξ κ.λπ.).
 - Υποστήριξη πρότυπων εγγράφων (template documents), παρέχοντας:
 - Ιεραρχίες καταλόγων, παρέχοντας την οργάνωση τοπικών ομαδοποιημένων εγγράφων σε ιεραρχίες καταλόγων (folder hierarchies). Οι κατάλογοι θα πρέπει να ορίζονται με βάση τη φύση και το αντικείμενο των εγγράφων και θα πρέπει να έχουν και αυτοί μεταδεδομένα που θα χρησιμοποιούνται κατά την αναζήτησή τους.
 - Ευέλικτους μηχανισμούς αναζήτησης εγγράφων, παρέχοντας εσωτερική μηχανή δεικτοδότησης (indexing) και αναζήτησης ώστε να είναι δυνατή η αναζήτηση ενός εγγράφου με βάση τα περιεχόμενά του (free text search), την αναζήτηση εγγράφων ή καταλόγων με βάση τη χρήση λέξεων-κλειδιών (keywords), την αναζήτηση εγγράφων ή καταλόγων με βάση τη συγκεκριμένη τιμή ενός μεταδεδομένου (ή περισσότερων μεταδεδομένων) του εγγράφου ή με βάση μια περιοχή της τιμής του, κ.ά.
- Επομένως, το υποσύστημα διαχείρισης εγγράφων αποτελεί βασικό υποσύστημα, διότι είναι αναγκαίο για τη λειτουργία όλων των υποσυστημάτων ενός συστήματος ERP, για τον λόγο ότι όλες οι επιχειρηματικές διεργασίες παράγουν έγγραφα. Επιπλέον αποτελεί τη βάση για την υλοποίηση λειτουργικότητας όπως:
- Της ψηφιακής υπογραφής εγγράφων, όπου το κάθε έγγραφο υπογράφεται χρησιμοποιώντας το ιδιωτικό κλειδί του χρήστη που υπογράφει. Στη συνέχεια το έγγραφο κρυπτογραφείται με βάση την υπογραφή του χρήστη, ώστε στο τέλος να δημιουργηθεί το ψηφιακά υπογεγραμμένο ηλεκτρονικό αποτύπωμα του εγγράφου.
 - Της δημιουργίας και λειτουργίας πρωτοκόλλου της επιχείρησης.
 - Της δημιουργίας και διαχείρισης συστήματος διαχείρισης κύκλου ζωής προϊόντων (product life cycle).

2.2.4 Υποσύστημα διαχείρισης δικτυακής πύλης (portal)

Μια δικτυακή πύλη είναι ένας δικτυακός τόπος που χρησιμεύει ως πύλη σε άλλους προορισμούς ή δραστηριότητες στο διαδίκτυο. Αντίστοιχα, μια εταιρική δικτυακή πύλη είναι ο δικτυακός τόπος της επιχείρησης όπου η επιχείρηση προσφέρει τις υπηρεσίες της, κεντρικά, τόσο στους πελάτες της, όσο και στους συνεργάτες της, στους εργαζομένους της, αλλά και στο ευρύ κοινό.

Μια διαδικτυακή πύλη αποτελεί το πρώτο σημείο επαφής των χρηστών των πληροφοριακών συστημάτων της επιχείρησης. Εκεί οι εσωτερικοί χρήστες, οι εργαζόμενοι της επιχείρησης διεκπεραιώνουν τις καθημερινές εργασίες τους, ενώ οι πελάτες είτε απλά ενημερώνονται είτε χρησιμοποιούν τις εφαρμογές άμεσης προσπέλασης (self-service) της επιχείρησης. Επομένως, μια δικτυακή πύλη θα πρέπει να προσφέρει τη δυνατότητα μιας ενιαίας εικαστικής προσέγγισης/εικόνας για όλα τα συστήματα της επιχείρησης. Η εικαστική προσέγγιση της δικτυακής πύλης ορίζεται από τα επιλεγμένα γραφικά, τις γραμματοσειρές, εικονίδια, λωρίδες ενημέρωσης (banners) κ.ά. Ταυτόχρονα, ο σχεδιασμός των εφαρμογών θα πρέπει να στοχεύει στην επίτευξη υψηλής χρηστικότητας και εργονομίας. Η χρηστικότητα των εφαρμογών θα πρέπει να βασίζεται εκτός από εργονομικούς κανόνες σε διεθνή πρότυπα όπως τα: WAI (Web Accessibility Initiative), CSS, κ.λπ.

Ο απλός χρήστης της δικτυακής πύλης θα πρέπει να έχει τη δυνατότητα χωρίς να χρειάζεται να έχει ιδιαίτερα σημαντικές τεχνικές γνώσεις να μπορεί να προσωποποιεί και να παραμετροποιεί το περιβάλλον του, να έχει πρόσβαση στη βοήθεια του συστήματος, ενώ η δικτυακή πύλη θα πρέπει να παρουσιάζει ομοιομορφία ως προς το περιβάλλον, όπως στα μενού, στις εργαλειοθήκες κ.ά. Αντίστοιχα, ο διαχειριστής του συστήματος θα πρέπει να μπορεί να δημιουργεί ένα αρχικό εξατομικευμένο περιβάλλον βάση των ενδιαφερόντων των διαφόρων ομάδων χρηστών. Επίσης, το σύστημα θα πρέπει να προτείνει στον χρήστη επιλογές/σελίδες ανάλογα με τις προγενέστερες επιλογές του.

Το περιεχόμενο μιας δικτυακής πύλης θα πρέπει να απεικονίζει το περιεχόμενο με δομημένο τρόπο. Αυτό σημαίνει ότι θα πρέπει να υπάρχει δυνατότητα κατηγοριοποίησης του περιεχομένου, δυνατότητα δημιουργίας ιεραρχικών ταξινομήσεων του περιεχομένου και συσχέτισης ενός αντικειμένου με πολλαπλές κατηγορίες. Η διάθεση του περιεχομένου της δικτυακής πύλης (κείμενο, εικόνες, video κ.λπ.) θα πρέπει να προσφέρεται με όσο το δυνατόν λιγότερους περιορισμούς μορφοποίησης και με χρήση των κατάλληλων viewers.

Η δικτυακή πύλη θα πρέπει να είναι δικτυωμένη με τα διαθέσιμα εργαλεία συνεργασίας της επιχείρησης - e-mail, χώροι συζητήσεων, ημερολόγια, νέα, συχνές ερωτήσεις κ.ά. Με τη χρήση αυτών των εργαλείων θα πρέπει να παρέχονται υπηρεσίες ειδοποίησης (alerts), υπενθύμισης με βάση ειδικά γεγονότα (π.χ. κλείσιμο ραντεβού, αλλαγή τιμών μετοχών, κ.λπ.). Επίσης θα πρέπει να δίνεται, αν απαιτείται, η δυνατότητα διενέργειας οικονομικών συναλλαγών με τη χρήση πιστωτικής κάρτας.

Μια ιδιαίτερα σημαντική λειτουργία της δικτυακής πύλης είναι η δυνατότητα αναζήτησης. Η πύλη θα πρέπει να παρέχει αναζήτηση ως προς λέξεις του περιεχομένου (full text search). Επιπλέον, η αναζήτηση θα πρέπει να είναι σύνθετη ως προς επιλεγμένα πεδία, ως προς λέξεις-κλειδιά, και ως προς μεταδεδομένα (metadata) περιεχομένου.

Η δικτυακή πύλη θα πρέπει να δίνει δυνατότητες για την υλοποίηση εφαρμογών συνεργατικού μάρκετινγκ και συνεπώς θα πρέπει να υπάρχει η δυνατότητα εμφάνισης περιεχομένου, διαφημίσεων και άλλου υλικού από τρίτες τοποθεσίες (syndication).

Όλο το περιεχόμενο της δικτυακής πύλης θα πρέπει να είναι αποθηκευμένο σε ένα σύστημα διαχείρισης περιεχομένου ώστε η διαχείρισή του να είναι εύχρηστη, με δυνατότητες αποδοτικής ανάπτυξης (αποκοπή, αντιγραφή, επικόλληση τμημάτων περιεχομένου στη δομή περιεχομένου του δικτυακού τόπου).

Όπου κριθεί απαραίτητο, οι ηλεκτρονικές υπηρεσίες θα πρέπει να είναι διαθέσιμες μέσω πολλαπλών καναλιών ώστε να αυξήσουμε τα σημεία επαφής με τους πελάτες. Για τον σκοπό αυτό οι ηλεκτρονικές υπηρεσίες θα χρησιμοποιούν κατά κύριο λόγο τη γλώσσα XML ως μέσο ανεξάρτητης απεικόνισης της πληροφορίας, ενώ η πληροφορία θα μετατρέπεται σε μορφή κατάλληλη για το εκάστοτε κανάλι παρουσίασης. Επίσης, αυτές οι ηλεκτρονικές υπηρεσίες θα βασίζονται στην αρχιτεκτονική σχεδίαση Model – View – Controller, που επιτρέπει την κατασκευή ηλεκτρονικών υπηρεσιών με πολλαπλά επίπεδα παρουσίασης τα οποία χρησιμοποιούν τον ίδιο κώδικα επιχειρηματικής λογικής.

Βασικό στοιχείο της δικτυακής πύλης είναι η δυνατότητα πρόσβασης των σελίδων μέσω όλων των χρησιμοποιούμενων φυλλομετρητών (browser), ακόμη και στις υποστηριζόμενες εκδόσεις από κινητές συσκευές (smart phones).

Προφανώς, οι λειτουργίες μιας δικτυακής πύλης τόσο όσον αφορά την οργάνωσή της, αλλά και τις παρεχόμενες λειτουργίες είναι ιδιαίτερα σύνθετες και η πλήρης ανάλυση αυτών των λειτουργιών είναι εκτός των στόχων του συγγράμματος.

2.2.5 Εργαλεία παραμετροποίησης πληροφοριακών συστημάτων (customization tools)

Τα πληροφοριακά συστήματα σήμερα καλούνται να καλύψουν τις πληροφοριακές ανάγκες επιχειρήσεων και οργανισμών πολλών διαφορετικών κλάδων. Αυτό σημαίνει ότι υπάρχει ανάγκη για σημαντικές διαφοροποιήσεις στη λειτουργικότητα, στη δομή, στο μέγεθος κ.λπ. (Passi & Ajnavaz, 2009). Η ανάγκη αυτή καλύπτεται με δύο βασικούς τρόπους:

- Με την ύπαρξη εξειδικευμένων λύσεων οι οποίες αυτοματοποιούν επιχειρηματικές διεργασίες συγκεκριμένων κλάδων. Για παράδειγμα, εξειδικευμένες λύσεις για τον τραπεζικό τομέα, για τις χημικές βιομηχανίες, για την ανώτατη εκπαίδευση κ.λπ.
- Με την παραμετροποίηση των πληροφοριακών συστημάτων.

Η παραμετροποίηση των πληροφοριακών συστημάτων επιτυγχάνεται με τη χρήση διαφορετικών εργαλείων τα οποία μπορούν να ενταχθούν στις παρακάτω κατηγορίες:

- Ολοκληρωμένο περιβάλλον ανάπτυξης νέου κώδικα (Integrated Development Environments),
- Γεννήτριες οθονών/φορμών (screen/form generators),
- Γεννήτριες αναφορών (report generators),
- Εργαλεία διαχείρισης και μετασχηματισμού δεδομένων (data transformation tools).

Ένα **ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού** (Integrated Development Environment) είναι ένα περιβάλλον που επιτρέπει την ανάπτυξη λογισμικού παρέχοντας στους προγραμματιστές τις αναγκαίες λειτουργίες και αυτοματισμούς. Εκτός από τη δημιουργία του νέου κώδικα, ένα σύγχρονο IDE υποστηρίζει:

- Την ανάπτυξη λογισμικού με συστατικά (components),
- Την ανάπτυξη του περιβάλλοντος διεπαφής με τους χρήστες με γραφικό τρόπο (visual programming),
- Την υποστήριξη διαφορετικών γλωσσών προγραμματισμού μέσα στο ίδιο περιβάλλον,
- Τη σύνδεση με διαφορετικές βάσεις δεδομένων,
- Τη σύνδεση με εργαλεία διαχείρισης σχηματισμών για τον καλύτερο έλεγχο πηγαίου κώδικα (configuration management),
- Τη διαχείριση βιβλιοθηκών λογισμικού,
- Την αυτοματοποίηση της αποσφαλματοποίησης (debugging) και ελέγχου,
- κ.ά.

Παραδείγματα δημοφιλών IDE αποτελούν τα συστήματα Eclipse (<https://eclipse.org>), Netbeans (<https://netbeans.org>), Microsoft Visual Studio (<https://www.visualstudio.com>) κ.ά.

Μια **γεννήτρια οθονών/φορμών** είναι ένα υποσύστημα που αυτοματοποιεί τη δημιουργία ή τροποποίηση των φορμών που εμφανίζονται κατά τη λειτουργία του συστήματος ERP. Η εφαρμογή αυτή μπορεί να βρίσκεται είτε μέσα στο IDE είτε να εργάζεται ανεξάρτητα. Η λειτουργικότητα ενός τέτοιου συστήματος θα πρέπει να περιλαμβάνει:

- Την ανάπτυξη του περιβάλλοντος διεπαφής με τους χρήστες με γραφικό τρόπο (visual programming),
- Τη σύνδεση της οθόνης με διαφορετικές βάσεις δεδομένων για ανάκτηση δεδομένων,
- Τη σύνδεση των συστατικών της φόρμας με γεγονότα είτε προερχόμενα από τους χρήστες είτε από το σύστημα (event based programming),
- Τη σύνδεση της οθόνης με το σύστημα ροής εργασιών, ώστε να εμφανίζονται τα σχετικά γεγονότα,
- Τη δυνατότητα δημοσίευσης της οθόνης στα διαφορετικά κανάλια επικοινωνίας με τους χρήστες του συστήματος (π.χ. σε παραδοσιακές παραθυρικές εφαρμογές, σε φυλλομετρητή, σε κινητές συσκευές).
- κ.ά.

Με την εγκατάσταση του συστήματος ERP, η επιχείρηση έχει στη διάθεσή της έναν σημαντικό αριθμό έτοιμων αναφορών οι οποίες καλύπτουν τις βασικές ανάγκες. Ποτέ όμως αυτές οι αναφορές δεν καλύπτουν πλήρως τις ανάγκες των χρηστών και σχεδόν πάντα προκύπτει η ανάγκη για δημιουργία νέων αναφορών, ανάγκη η οποία καλύπτεται με την ύπαρξη **υποσυστήματος δημιουργίας αναφορών**. Η ανάγκη για δημιουργία νέων αναφορών είναι μεγάλη κατά την αρχική φάση εγκατάστασης του συστήματος, αλλά εξακολουθεί να υπάρχει και σε όλη τη διάρκεια του κύκλου ζωής του συστήματος ERP.

Επομένως, ένα υποσύστημα δημιουργίας αναφορών είναι ιδιαίτερα σημαντικό και χρήσιμο, αφού η δυνατότητα αυτή επιτρέπει στους χρήστες να έχουν διαθέσιμη την κατάλληλη πληροφορία δοσμένη με τον σωστό τρόπο. Οι παραγόμενες αναφορές μπορεί να έχουν διαφορετικές μορφές και να εξυπηρετούν διαφορετικούς σκοπούς όπως:

- Λειτουργικές αναφορές, όπως για παράδειγμα ένα τιμολόγιο, μια εντολή πληρωμής, μια εντολή προμήθειας κ.λπ.
- Συγκεντρωτικές αναφορές που παράγονται σε περιοδική βάση (ημερήσια, εβδομαδιαία, μηνιαία). Για παράδειγμα, οι προγραμματισμένες παραγγελίες της ημέρας, οι μηνιαίες πωλήσεις ανά προϊόν κ.λπ.
- Περιστασιακές αναφορές (Ad hoc) που παράγονται συνήθως κατά περίπτωση ώστε να εξυπηρετήσουν τις ανάγκες της διοίκησης. Για παράδειγμα, ποιο είναι το κατάστημα με τις μεγαλύτερες πωλήσεις ανά προϊόν.
- Αναφορές απόδοσης που παρουσιάζουν βασικούς δείκτες απόδοσης της επιχείρησης (Key Performance Indicators – KPIs).

Ένα προϊόν κατάλληλο για τη δημιουργία αναφορών θα πρέπει να υποστηρίζει:

- Δημιουργία αναφορών σε διαφορετικές μορφές,
- Σύνδεση με διαφορετικές βάσεις δεδομένων αλλά και πηγές δεδομένων στην ίδια ή διαφορετικές αναφορές (π.χ. βάσεις δεδομένων, λογιστικά φύλλα, απλά αρχεία),
- Δημιουργία αναφορών με χρήση προτύπων (templates) ώστε οι χρήστες να έχουν δυνατότητα προσωποποίησης (personalization) της αναφοράς χωρίς την ανάγκη τεχνικών γνώσεων,
- Ολοκλήρωση με λογιστικά φύλλα (π.χ. Excel),
- Δημιουργία αναφορών για διαφορετικά κανάλια διανομής της αναφοράς. Για παράδειγμα, οι αναφορές θα πρέπει να εμφανίζονται σωστά στο διαδίκτυο, σε όλους τους διαφορετικούς φυλλομετρητές, σε κινητές συσκευές ή να αποθηκεύονται σε διαφορετικές μορφές αρχείων κ.λπ.,
- Δημιουργία δυναμικών – διαδραστικών αναφορών όπου εμφανίζονται βασικοί δείκτες απόδοσης (KPIs),
- Δυνατότητα υποστήριξης πολυδιάστατης ανάλυσης δεδομένων σε συνδυασμό με την τεχνολογία OLAP (On-Line Analytical Processing) (Βλέπε κεφάλαιο 4.4).

Σχετικά με τα εργαλεία διαχείρισης και μετασχηματισμού δεδομένων (data transformation tools) οι βασικές λειτουργίες θα επεξηγηθούν περισσότερο στην παράγραφο 4.4.

2.3 Παράμετροι επιλογής αρχιτεκτονικής πληροφοριακού συστήματος

Για τον σχεδιασμό της αρχιτεκτονικής ενός πληροφοριακού συστήματος για μια επιχείρηση λαμβάνεται υπόψη μια σειρά παραμέτρων. Κάθε παράμετρος αφορά σε μια διαφορετική οπτική γωνία (θεώρηση) υπό την οποία εκτιμάται το πληροφοριακό σύστημα. Στις βέλτιστες συνιστώμενες πρακτικές οι θεωρήσεις που λαμβάνονται υπόψη για τον προσδιορισμό της αρχιτεκτονικής είναι οι ακόλουθες:

2.3.1 Μέγεθος και γεωγραφική κάλυψη της επιχείρησης

Το μέγεθος και η γεωγραφική κάλυψη μιας επιχείρησης προσδιορίζουν το μέγεθος και την κατανομή των πόρων του απαιτούμενου πληροφοριακού συστήματος. 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,
- Γεωγραφική κάλυψη.

Από την ανάλυση του μεγέθους της επιχείρησης προκύπτουν ορισμένα σημαντικά στοιχεία για την αρχιτεκτονική του πληροφοριακού συστήματος που αφορούν:

- Ανάγκες σε υποσυστήματα που θα επιτελούν 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 σε δεδομένα που παράγονται από το σύστημα,
- Απαιτήσεις για κατανομή του πληροφοριακού συστήματος σε γεωγραφικές περιοχές,
- Χαρακτηριστικά γεωγραφικών περιοχών και κίνδυνοι φυσικών καταστροφών.
- Με την καταγραφή των ανωτέρω απαιτήσεων καταγράφονται πληροφορίες που αφορούν:
 - ο Στο μέγεθος των υποδομών του συστήματος (πλήθος εξυπηρετητών, μέγεθος αποθηκών δεδομένων).
 - ο Στην κατανομή του συστήματος σε αντίγραφα (mirrors) που λειτουργούν σε διαφορετικές γεωγραφικές περιοχές και ανταλλάσσουν δεδομένα με κάποιο κεντρικό σημείο.
 - ο Στη διασύνδεση του εξοπλισμού (απαιτήσεις για εξαιρετικά υψηλές ταχύτητες που επιτυγχάνονται μέσω FDDI κ.λπ.).
 - ο Στην αδειοδότηση των εφαρμογών έτοιμου λογισμικού που λειτουργούν στο πληροφοριακό σύστημα.
 - ο Στην κατάρτιση συμβολαίων εγγυημένου επιπέδου παροχής υπηρεσιών Service Level Agreement (SLA) με τρίτους.
 - ο Στην κατάρτιση σχεδίου επαναφοράς από φυσικές καταστροφές (disaster recovery plan).

2.3.2 Επιχειρησιακές λειτουργίες

Οι επιχειρησιακές λειτουργίες αφορούν στις διεργασίες που επιτελούνται εντός της επιχείρησης και οι οποίες θα επιτυγχάνονται μέσω του πληροφοριακού συστήματος. Από αυτές προκύπτει η διάκριση μεταξύ:

- Λειτουργιών που εκτελούνται αποκλειστικά εντός της επιχείρησης,
- Λειτουργιών που προσφέρουν υπηρεσίες σε ομάδες αποδεκτών της επιχείρησης.

Οι επιχειρησιακές λειτουργίες αναλύονται σε βήματα και φάσεις που διεκπεραιώνονται από οργανωτικά τμήματα και στελέχη της επιχείρησης και που θα υλοποιηθούν με εφαρμογές λογισμικού του πληροφοριακού συστήματος. Πρόκειται για τη σημαντικότερη θεώρηση για το πληροφοριακό σύστημα, καθώς απαιτείται η καταγραφή όλων των επιχειρησιακών διεργασιών και η αντιστοίχισή τους σε εφαρμογές λογισμικού και υποδομές που πρόκειται να τις υποστηρίξουν.

Στο πλαίσιο της καταγραφής προσδιορίζονται οι εφαρμογές λογισμικού που υφίστανται ήδη στην επιχείρηση. Καταγράφεται η αρχιτεκτονική τους, η δυνατότητα διαλειτουργικότητας με σύγχρονες εφαρμογές, το κόστος της μετάπτωσής τους στο νέο σύστημα και η σπουδαιότητα ή μη της διατήρησής τους.

2.3.3 Χρήστες του συστήματος

Οι χρήστες του πληροφοριακού συστήματος αφορούν σε:

- Στελέχη της επιχείρησης που θα χρησιμοποιούν τις εφαρμογές λογισμικού του συστήματος προκειμένου να διεκπεραιώνουν τις επιχειρησιακές διεργασίες,
- Πελάτες-αποδέκτες των προϊόντων, των πληροφοριών και των υπηρεσιών της επιχείρησης,
- Οργανωτικές οντότητες της επιχείρησης (π.χ. τμήματα, παραρτήματα, υποκαταστήματα, δομές και ομάδες χρηστών που θα συντονίζονται ή θα ενημερώνονται από το σύστημα),
- Συνεργάτες ή/και προμηθευτές των πρώτων υλών της επιχείρησης,
- Φορείς (π.χ. ασφαλιστικοί και οικονομικοί φορείς) που λαμβάνουν δεδομένα από την επιχείρηση,
- Διαχειριστές, αναλυτές (των εφαρμογών και των δεδομένων) και προγραμματιστές του πληροφορια-

κού συστήματος.

Από την ανωτέρω ανάλυση προκύπτουν σημαντικά στοιχεία αναφορικά με το πληροφοριακό σύστημα, που αφορούν:

- Τους ανθρώπους της επιχείρησης που πρόκειται να επηρεαστούν από την εγκατάσταση και λειτουργία του πληροφοριακού συστήματος,
- Την αρχιτεκτονική των εφαρμογών του συστήματος,
- Την κατανομή των περιφερειακών πόρων του συστήματος (σταθμοί εργασίας, τρίτες συσκευές),
- Τα περιβάλλοντα διεπαφής των χρηστών (ui) του συστήματος,
- Τις διεπαφές (interfaces) του συστήματος προς τρίτα συστήματα,
- Την αδειοδότηση των εφαρμογών έτοιμου λογισμικού του συστήματος,
- Τις ανάγκες διαθεσιμότητας και αποκρισιμότητας του συστήματος,
- Την παρακολούθηση και διαχείριση του συστήματος,
- Την ασφάλεια του συστήματος (ανάγκες διαβαθμισμένης πρόσβασης σε πόρους, κρίσιμα δεδομένα, διάκριση των λογικών περιοχών της αρχιτεκτονικής σε διακριτές ζώνες (zones), ανάγκες σε υποδομές ασφαλείας κ.λπ.).

2.3.4 Όγκος και είδος δεδομένων

Τα δεδομένα της επιχείρησης αφορούν σε:

- Έγγραφα και αρχεία που παράγονται από την επιχείρηση,
- Εγγραφές που παράγονται κατά την εκτέλεση των επιχειρησιακών λειτουργιών και πρέπει να τηρούνται στο σύστημα.

Από την ανάλυση των δεδομένων, του είδους και του όγκου τους, προκύπτουν σημαντικές πληροφορίες αναφορικά με:

- Την απαιτούμενη αρχιτεκτονική για αποθήκες δεδομένων,
- Την απαιτούμενη χωρητικότητα σε αποθήκες δεδομένων, με πλάνο επέκτασης των μονάδων αποθήκευσης,
- Χρονική ισχύς των αρχείων και των εγγραφών,
- Λύση και πλάνο τήρησης αντιγράφων ασφαλείας,
- Κατανομή και διαβάθμιση πληροφοριών,
- Ανάγκες σε βάσεις δεδομένων,
- Λύση για τη δημιουργία αποθηκών δεδομένων (data warehousing), πολυδιάστατη ανάλυση δεδομένων (OLAP) και εξόρυξη δεδομένων (data mining).

2.4. Μη λειτουργικές απαιτήσεις από την αρχιτεκτονική του πληροφοριακού συστήματος

Εκτός από τις βασικές και τις υποστηρικτικές λειτουργίες, ένα πληροφοριακό σύστημα ERP θα πρέπει να είναι σε θέση να ικανοποιεί μια σειρά μη λειτουργικών απαιτήσεων (non-functional requirements).

Οι μη-λειτουργικές απαιτήσεις περιγράφουν ιδιότητες του συστήματος ERP που έχουν άμεση σχέση με χαρακτηριστικά όπως διαχειρισιμότητα, επεκτασιμότητα, διαλειτουργικότητα, απόδοση, ασφάλεια κ.ά. Τα μη λειτουργικά χαρακτηριστικά επηρεάζουν σε μεγάλο βαθμό την αρχιτεκτονική ενός πληροφοριακού συστήματος ERP. Στις παρακάτω παραγράφους θα αναλύσουμε περισσότερο μερικά από αυτά.

2.4.1 Διαλειτουργικότητα

Ο όρος διαλειτουργικότητα (interoperability) αναφέρεται στη δυνατότητα επικοινωνίας συστημάτων και εφαρμογών που είναι ασύμβατα, δηλαδή χρησιμοποιούν διαφορετικά πρότυπα μοντελοποίησης πληροφοριών και διαφορετικά πρότυπα επικοινωνίας.

Η έννοια της διαλειτουργικότητας (interoperability) συγγέεται με τις «συγγενείς» έννοιες της ολοκλήρωσης (integration), της συνεργασίας (collaboration/synergy), της μεταφερσιμότητας (portability) και της ανταλλαξιμότητας (interchangeability). Ωστόσο, υπάρχει ένα σύνολο ουσιαστικών διαφορών μεταξύ των εννοιών αυτών και της έννοιας της διαλειτουργικότητας (Αλεξόπουλος, 2011). Οι διαφορές αυτές αναλύονται ακολούθως:

- **Διαλειτουργικότητα και ολοκλήρωση (Integration):** Η διαλειτουργικότητα εμπεριέχει την έννοια της συνύπαρξης και της αυτονομίας, σε αντίθεση με την ολοκλήρωση που στοχεύει στην ομογενοποίηση. Η ολοκλήρωση δηλαδή έχει την έννοια της στενής διασύνδεσης (tightly coupled) των διάφορων εμπλεκόμενων οντοτήτων σε τέτοιο βαθμό στον οποίο δεν είναι εφικτός ο διαχωρισμός αυτών, από το σύνολο το οποίο συνθέτουν. Αντιθέτως, η διαλειτουργικότητα σχετίζεται με την έννοια της χαλαρής διασύνδεσης (loose coupling), κατάσταση στην οποία οι εμπλεκόμενες επιχειρήσεις και οντότητες μπορούν να ανταλλάσσουν υπηρεσίες και να συνεργάζονται για την παροχή υπηρεσιών, διατηρούν ωστόσο την αυτονομία τους. Κατά συνέπεια και υπό αυτήν την έννοια κάθε ολοκληρωμένο σύστημα είναι οπωσδήποτε και διαλειτουργικό. Αντιθέτως, δυο συστήματα τα οποία διαλειτουργούν δεν είναι υποχρεωτικό να είναι και ολοκληρωμένα.
- **Διαλειτουργικότητα και συνεργασία (Collaboration):** Η διαλειτουργικότητα συσχετίζεται με τη συμβατότητα και την ικανότητα συνεργασίας, διότι δυο διαλειτουργούσες επιχειρήσεις είναι απαραίτητο να συνεργάζονται. Δύο συνεργαζόμενοι οργανισμοί όμως, μπορεί να μην διαλειτουργούν.
- **Διαλειτουργικότητα και μεταφερσιμότητα (Portability):** Η τελευταία αφορά την ικανότητα μεταφοράς συστημάτων ή λογισμικού από το περιβάλλον ενός οργανισμού στο περιβάλλον ενός άλλου χωρίς να χρειαστούν τροποποιήσεις.
- **Διαλειτουργικότητα και αντιμεταθετικότητα (Interchangeability):** Η τελευταία αναφέρεται στην ικανότητα συνέχισης παροχής του ίδιου επιπέδου σταθερής συμπεριφοράς υπηρεσιών παρά την αλλαγή ενός συστήματος ή τμήματος αυτού.

Υπάρχουν τρεις έννοιες διαλειτουργικότητας: (Cafezeiro, 2007)

- Η τεχνική διαλειτουργικότητα, η οποία ασχολείται με την τεχνολογική πλευρά της διασύνδεσης συστημάτων, με τον προσδιορισμό των σημείων διεπαφής μεταξύ των συστημάτων, με τη μορφοποίηση των δεδομένων και με τα πρωτόκολλα επικοινωνίας των συστημάτων.
- Η σημασιολογική διαλειτουργικότητα, η οποία αναφέρεται στην κατανόηση του περιεχομένου της ανταλλασσόμενης πληροφορίας μεταξύ των συστημάτων.
- Η οργανωσιακή διαλειτουργικότητα, η οποία αναφέρεται στη μοντελοποίηση των επιχειρηματικών μεθόδων και την ευθυγράμμιση των συστημάτων με τους στόχους των οργανισμών, οι οποίοι τα εγκαθιστούν και τα χρησιμοποιούν.

Η έννοια της διαλειτουργικότητας έχει μεγάλη σημασία για μια επιχείρηση, καθώς πολλά από τα συστήματα που λειτουργούν σε αυτόν είναι παραδοσιακά (legacy), έχουν εγκατασταθεί σε διαφορετικούς χρόνους και για διαφορετικό σκοπό. Τα συστήματα αυτά, εφόσον αξιολογηθεί η σημασία τους και επιβεβαιωθεί ότι πρέπει να διατηρηθούν και να αξιοποιηθούν, τότε πρέπει να επιτευχθεί η διασύνδεσή τους με το νέο σύστημα. Η διαλειτουργικότητα στοχεύει στην επίτευξη επικοινωνίας μεταξύ διαφορετικών συστημάτων και βασίζεται κυρίως στην ανάπτυξη υποστηρικτικών «ενδιάμεσων εφαρμογών» (middleware), αλλά και στη χρήση συγκεκριμένων σχημάτων σήμανσης δεδομένων (schemas), ώστε το σύγχρονο με τα παραδοσιακά συστήματα να επικοινωνούν και να ανταλλάσσουν δεδομένα.

Επομένως η υλοποίηση του χαρακτηριστικού της διαλειτουργικότητας περιλαμβάνει:

- Μια σαφώς προσδιορισμένη και καθορισμένη μορφή για τις πληροφορίες (πρότυπα δόμησης της πληροφορίας/δεδομένων και της μετα-πληροφορίας/δεδομένων),
- Έναν σαφώς προσδιορισμένο και καθορισμένο τρόπο για την ανταλλαγή των πληροφοριών (τεχνολογίες επικοινωνιών και πρωτόκολλα με τα οποία μεταφέρεται η πληροφορία με τη μορφή που καθορίζεται στο προηγούμενο σημείο),
- Έναν σαφώς προσδιορισμένο και καθορισμένο τρόπο για την πρόσβαση στις πληροφορίες και στα δεδομένα (ασφάλεια/έλεγχος πρόσβασης, δηλαδή τεχνολογίες που χρησιμοποιούνται για την προστασία των υπηρεσιών διαλειτουργικότητας),
- Έναν σαφώς προσδιορισμένο και καθορισμένο τρόπο για την αναζήτηση των πληροφοριών και των δεδομένων (τεχνολογίες μετα-δεδομένων, καταλόγου ή άλλες που χρησιμοποιούνται για την αναζήτηση πληροφοριών στα πλαίσια των διαλειτουργικών υπηρεσιών).

2.4.2 Απαιτήσεις ασφάλειας

Η διασφάλιση ενός πληροφοριακού συστήματος είναι μια ευρεία διεργασία που αγγίζει κάθε δραστηριότητα της επιχείρησης. Ένα από τα ζητήματα που εξετάζονται κατά τον σχεδιασμό της ασφάλειας του συστήματος είναι η επίτευξη λύσης που δεν θα επηρεάσει την ανταγωνιστικότητα της επιχείρησης. Η λύση που προκύπτει ονομάζεται «αρχιτεκτονική ασφαλείας» του συστήματος. Η αρχιτεκτονική ασφαλείας περιλαμβάνει (ORACLE, 2010):

- Πολιτική ασφαλείας, σωστά καταρτισμένη και γνωστοποιημένη σε όλα τα στελέχη του Οργανισμού,
- Υλοποίηση και εφαρμογή της πολιτικής ασφαλείας εντός της επιχείρησης,
- Ενσωμάτωση μεθόδων ασφαλείας σε κάθε επίπεδο και στο σύνολο του συστήματος.

Η διαμόρφωση της αρχιτεκτονικής ασφαλείας προκύπτει από τον προσδιορισμό:

- ο Ποιος πόρος του συστήματος κινδυνεύει και πρέπει να προστατευτεί;
- ο Το απαιτούμενο επίπεδο προστασίας στον πόρο που κινδυνεύει.

Οι σημαντικότεροι κίνδυνοι από τους οποίους πρέπει να διασφαλίζεται ένα πληροφοριακό σύστημα είναι οι ακόλουθοι (Rainer, 2012):

- Ιοί (viruses, worms, trojans),
- Επιθέσεις από υπαλλήλους,
- Καταστροφή σημαντικών δεδομένων λόγω αμέλειας των υπαλλήλων,
- Εξωτερικές επιθέσεις στο σύστημα (hacking),
- Επιθέσεις που εμποδίζουν την παροχή υπηρεσιών (Denial of Service - DOS).

Η χρήση εφαρμογών αντι-ικής προστασίας και η τακτική τήρηση αντιγράφων ασφαλείας είναι οι μηχανισμοί προστασίας από τους αντίστοιχους κινδύνους. Η προστασία του συστήματος από εσωτερικές επιθέσεις επιτυγχάνεται ως εξής:

- Με την οργάνωση της αρχιτεκτονικής του πληροφοριακού συστήματος σε ζώνες πρόσβασης (αποστρατιωτικοποιημένη ζώνη – DMZ, ζώνες αόρατες προς τους χρήστες),
- Με τη διασφάλιση του συστήματος από φυσική πρόσβαση,
- Με τη διασφάλιση των ευαίσθητων δεδομένων από υπεξαίρεση ή καταστροφή,
- Με τον προσδιορισμό ομάδων χρηστών και αντίστοιχων, κατάλληλων δικαιωμάτων πρόσβασης,
- Με τον ορισμό αποτελεσματικών πολιτικών σύνταξης κωδικών (password policies) για την προστασία των πόρων του συστήματος. Σε ορισμένες περιπτώσεις, η χρήση έξυπνων καρτών (smart cards) μπορεί να κριθεί αναγκαία,
- Με τον προσδιορισμό αποτελεσματικών πολιτικών για τους σταθμούς εργασίας (π.χ. απαγόρευση χρήσης περιφερειακών μονάδων αποθήκευσης, χρήση δικτυακών προφίλ, μετατροπή των σταθμών σε τυφλά τερματικά κ.λπ.),
- Με την αποτελεσματική διαχείριση και επικαιροποίηση της ασφάλειας του συστήματος,
- Με την τήρηση και παρακολούθηση των αρχείων καταγραφής συμβάντων (logging) του συστήματος.

Η προστασία του συστήματος από εξωτερικές επιθέσεις (hacking) επιτυγχάνεται με την κατάτμηση του πληροφοριακού συστήματος σε ζώνες, καθώς και με την ενσωμάτωση στο σύστημα υποδομών ασφαλείας (συσκευών ή εφαρμογών λογισμικού), που θα ελέγχονται και θα επικαιροποιούνται διαρκώς. Ο κίνδυνος από εξωτερικές επιθέσεις είναι διαρκής, αλλά σε γενικές γραμμές δεν μπορεί να θεωρείται κρίσιμος. Περισσότερο εκτεθειμένο είναι ένα πληροφοριακό σύστημα στους εσωτερικούς του χρήστες και για τον λόγο αυτό οι hackers επιτίθενται με τρόπους που τους δίνουν έλεγχο σε εσωτερικό χρήστη (μέσω Trojan).

Ο μεγαλύτερος κίνδυνος από εξωτερική επίθεση είναι οι επιθέσεις Denial Of Service (DOS). Για να επιτευχθούν θα πρέπει η επιχείρηση να είναι μεγέθους που την καθιστούν στόχο και απαιτείται η δέσμευση από πλευράς του επιτιθέμενου ενός μεγάλου αριθμού Η/Υ που θα στέλνουν μεγάλο αριθμό αιτημάτων προς το πληροφοριακό σύστημα, ώστε το τελευταίο να μην μπορεί να αποκριθεί. Μια λύση για τον κίνδυνο αυτό αφορά στην επίτευξη ενός επιπέδου παρουσίασης στο πληροφοριακό σύστημα με ισοκατανομή των πόρων (ισοκατανομή (load-balancing) φορτίου και χρηστών). Επιπλέον, ένας καλός τρόπος είναι η διάκριση των εξωτερικών χρηστών ανάλογα με τη γεωγραφική τους προέλευση και η δρομολόγησή τους σε διαφορετικά υποσυστήματα του συστήματος ή η εξυπηρέτησή τους από γεωγραφικά κατανεμημένα υποσυστήματα.

Γενικότερα, η ενσωμάτωση ασφαλείας στο πληροφοριακό σύστημα θα πρέπει να βασίζεται σε ευέλικτη αρχιτεκτονική ασφαλείας, που θα επιτρέπει τον επαναπροσδιορισμό και την επέκτασή της, καθώς οι μηχανισμοί ασφαλείας μπορεί να αυξήσουν σημαντικά το κόστος του πληροφοριακού συστήματος.

2.4.3 Απαιτήσεις απόδοσης

Λαμβάνοντας ως δεδομένο ότι ένα σύστημα ERP δεν είναι ένα μονοδιάστατο σύστημα αλλά αποτελείται από υπολογιστικό, αποθηκευτικό και δικτυακό εξοπλισμό, λογισμικό βασικών συστημάτων και εφαρμογές, η ομαλή και ικανοποιητική συνολική απόδοση του συστήματος εξαρτάται από την απόδοση όλων των επιμέρους συστατικών του.

Είναι λοιπόν σημαντικό η σχεδίαση ενός τέτοιου συστήματος να διασφαλίζει τη συνολικά ικανοποιητική απόδοση του πληροφοριακού συστήματος. Η αξιολόγηση της απόδοσης ενός πληροφοριακού συστήματος μπορεί να γίνει με την εκτέλεση συγκεκριμένων σεναρίων αξιολόγησης, σεναρίων που εξομοιώνουν βασικές λειτουργίες και συναλλαγές που εκτελούνται στο πληροφοριακό σύστημα. Για παράδειγμα, ένα συγκριτικό τεστ ελέγχου της απόδοσης (benchmark) του συστήματος SAP προσομοιώνει τις πιο χαρακτηριστικές συναλλαγές και ροές εργασιών συγκεκριμένων χρηστών. Το σενάριο εκτελείται σε μια προκαθορισμένη βάση δεδομένων πελατών, με δεδομένα που λειτουργούν ως σημείο αναφοράς. Οι συναλλαγές αναφοράς που εκτελούνται για το κάθε υποσύστημα είναι και αυτές συγκεκριμένες και συνήθως περιλαμβάνουν παραγγελίες, μετακινήσεις προϊόντων κ.λπ.) (Fujitsu, 2015).

Ο εξυπηρετητής που χρησιμοποιείται έχει το συνηθισμένο μέγεθος που θα επέλεγε μια τυπική επιχείρηση. Επιπλέον, το ίδιο σενάριο εκτελείται για 100 διαφορετικούς χρήστες ταυτόχρονα.

Για παράδειγμα, ένα τυπικό σενάριο αξιολόγησης περιλαμβάνει τα ακόλουθα:

- Δημιουργία παραγγελίας που περιλαμβάνει πέντε προϊόντα,
- Δημιουργία εντολής παράδοσης για τη συγκεκριμένη παραγγελία,
- Προβολή της παραγγελίας πελάτη,
- Αλλαγή της εντολής παράδοσης,
- Παρουσίαση των 40 τελευταίων παραγγελιών του συγκεκριμένου πελάτη,
- Δημιουργία τιμολογίου για την παραγγελία.

Σε αυτό το είδος της αξιολόγησης συνήθως μετράμε τον μέσο χρόνο απόδοσης, το ποσοστό χρήσης του επεξεργαστή, το ποσοστό χρήσης της μνήμης κ.λπ. Η αξιολόγηση της απόδοσης των πληροφοριακών συστημάτων ERP είναι ιδιαίτερα σημαντική διότι το αποτέλεσμά της επιτρέπει να προβλέψουμε την απόδοση των συστημάτων και να προβούμε σε διορθωτικές ενέργειες προληπτικά. Για τον λόγο αυτό η όλη διαδικασία έχει τυποποιηθεί μέσω του Transaction Processing Council (TPC) (www.tpc.org), τόσο για τα συστήματα λήψης αποφάσεων, όσο και για τα συστήματα συναλλαγών (On-Line Transaction Processing). Το τεστ συγκριτικής αξιολόγησης που χρησιμοποιείται για τα συστήματα ERP είναι το TCP-C καθώς και το νεότερο TCP-E.

Αντίστοιχα, για την αξιολόγηση της λειτουργίας των δικτυακών εφαρμογών χρησιμοποιείται το TPC-W. Οι χρόνοι απόκρισης στο συγκριτικό τεστ αξιολόγησης μετρούνται από τη στιγμή αποστολής αιτήματος από έναν εξομοιωτή φυλλομετρητή μέχρι τη στιγμή ολοκλήρωσης της λήψης πληροφοριών από τον browser. Για παράδειγμα, ενδεικτικά αναφέρονται οι μέσοι χρόνοι απόκρισης (Web Interaction Response Times – WIRT) μίας τυπικής δικτυακής εφαρμογής:

- Πρόσβαση σε σελίδες στατικού περιεχομένου: 3 sec
- Πρόσβαση σε σελίδες αποστολής τυποποιημένων παραμετρικών αιτημάτων: 3 sec
- Επιβεβαίωση επεξεργασίας αιτήματος: 5 sec
- Πρόσβαση σε σελίδες αναζήτησης πληροφοριών: 3 sec
- Αποστολή αποτελεσμάτων αναζήτησης: 10 sec
- Πρόσβαση σε σελίδες διαχείρισης εφαρμογής: 5 sec
- Επιβεβαίωση ενεργειών διαχείρισης εφαρμογής: 20 sec

2.4.4 Απαιτήσεις ευχρηστίας και προσβασιμότητας

Η ευκολία χρήσης είναι ένα, κατά βάση, υποκειμενικό χαρακτηριστικό που όμως είναι πολύ σημαντικό για την επιτυχία και την ευρεία αποδοχή ενός ηλεκτρονικού συστήματος. Ένα πληροφοριακό σύστημα ERP απευθύνεται σε μια ευρεία κλίμακα χρηστών, όχι απαραίτητα εξοικειωμένων σε μεγάλο βαθμό με την τεχνολογία, οπότε χρειάζεται να παρέχει απλές και διαισθητικές γραφικές διεπαφές για την υλοποίηση λειτουργιών που χρειάζονται είσοδο από κάποιο χρήστη. Παράλληλα, όπως κάθε σύστημα, πρέπει να παρέχει μια υπηρεσία βοήθειας και αναλυτικό εγχειρίδιο για την καθοδήγηση του χρήστη στην αλληλεπίδρασή του με αυτό. Επίσης, για να μπορέσει να ανταποκριθεί σε όσο το δυνατόν ευρύτερο κοινό, θα πρέπει να λαμβάνει υπόψη του κατάλληλες πρακτικές για να μπορεί να απευθύνεται και σε άτομα με ειδικές ανάγκες μέσω της υλοποίησης του WCAG 2.0. Οι Οδηγίες για την προσβασιμότητα του περιεχομένου του ιστού WCAG 2.0 καλύπτουν ένα ευρύ φάσμα θεμάτων και συστάσεων για να καταστεί το περιεχόμενο των δικτυακών τόπων πιο προσβάσιμο. Με τη βοήθεια αυτών των οδηγιών το περιεχόμενο μπορεί να γίνει προσβάσιμο από ένα ευρύτερο φάσμα ατόμων με αναπηρίες. Επιπρόσθετα, η τήρηση αυτών των οδηγιών θα καταστήσει το περιεχόμενο των δικτυακών τόπων πιο εύχρηστο γενικά για όλους τους χρήστες. Το WCAG 2.0 περιλαμβάνει οδηγίες για:

- Το αντιληπτό περιεχόμενο (π.χ. παροχή εναλλακτικών κειμένων για κάθε περιεχόμενο που δεν διατίθεται σε μορφή κειμένου, ώστε να μπορεί να αποδοθεί σε άλλες μορφές που χρειάζονται οι χρήστες ή παροχή εναλλακτικών λύσεων για πολυμέσα που εξαρτώνται από τον χρόνο),
- Το λειτουργικό περιεχόμενο (π.χ. δυνατή χρήση όλων των λειτουργιών του συστήματος μέσω πληκτρολογίου),
- Την κατανόηση του περιεχομένου (π.χ. δημιουργία ιστοσελίδων με προβλέψιμη διάταξη και λειτουργία),
- Εύρωστο περιεχόμενο (περιεχόμενο χωρίς λάθη).

Βιβλιογραφία/Αναφορές

- Bass, L., Clements, P. & Kazman, R. (2013). *Software Architecture in Practice*. 3rd edition. Addison-Wesley Professional.
- Cafezeiro, I. & Haeusler, E. H. (2007). Semantic interoperability via category theory. In *Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual modeling-Volume 83* (pp. 197-202). Australian Computer Society Inc.
- Deacon, J. (2009). *Model-view-controller (MVC) architecture*. Ανακτήθηκε 22/09/2015 από <http://www.jdl.co.uk/briefings/MVC.pdf>.
- Eugster, P. T., Felber, P. A., Guerraoui, R. & Kermarrec, A. M. (2003). The many faces of publish/subscribe. *ACM Computing Surveys (CSUR)*, 35(2), 114-131.
- Fujitsu (2015). *Benchmark Overview SAP SD Standard Application Benchmark, Version 2.1*. Ανακτήθηκε 22/09/2015 από <http://docs.ts.fujitsu.com/dl.aspx?id=0a1e69a6-e366-4fd1-a1a6-0dd93148ea10>.
- Heilig, L., Karch, S., Bttcher, O., Hofmann, C. & Pfennig, R. (2007). *SAP NetWeaver master data management*. SAP PRESS.
- Homer, A., Sharp, J., Brader, L., Narumoto, M. & Swanson, T. (2014). *Cloud Design Patterns: Prescriptive Architecture Guidance for Cloud Applications*. Microsoft patterns & practices.
- Jaakkola, H. & Thalheim, B. (2010). *Architecture-Driven Modelling Methodologies*. In *EJC* (pp. 97-116).
- Kruchten, P., Booch, G., Bittner, K. & Reitman, R. (2004). *Who are the Software Architects and What Do They Do?* Ανακτήθηκε 22/09/2015 από www.di.univaq.it/alfonso/slides.pdf.
- Meier, J. D., Hill, D., Homer, A., Jason, T., Bansode, P., Wall, L. & Bogawat, A. (2009). *Microsoft application architecture guide Version 2*. Microsoft Press.
- NIST (2011). *Final Version of NIST Cloud Computing Definition Published*. Ανακτήθηκε 22/09/2015 από <http://www.nist.gov/itl/csd/cloud-102511.cfm>.
- ORACLE (2010). *Oracle® E-Business Suite System Administrator's Guide – Security, Release 12.1*. Ανακτήθηκε 22/09/2015 από http://docs.oracle.com/cd/E18727_01/doc.121/e12843.pdf.
- Passi, A. & Ajvaz, V. (2009). *Oracle E-Business Suite Development & Extensibility Handbook*. McGraw-Hill Inc.

- Rainer, R. K. & Cegielski, C. G. (2012). *Introduction to information systems: Enabling and transforming business*. 5th edition. John Wiley & Sons.
- Shaw, M. & Garlan, D. (1996). *Software architecture: perspectives on an emerging discipline* (Vol. 1, p. 12). Englewood Cliffs: Prentice Hall.
- Snabe, J. H., Rosenberg, A., Miller, C. & Scavillo, M. (2008). *Business process management: The Sap roadmap*. SAP PRESS.
- Stajda, E. (2013). *Document Management with SAP DMS*. Galileo Press.
- WCAG 2.0 (2008). *Web Content Accessibility Guidelines 2.0*. Ανακτήθηκε 22/09/2015 από <http://www.w3c.gr/wai/translations/wcag20.html>.
- Αλεξόπουλος, Χ. (2011). *Διαλειτουργικότητα Πληροφοριακών Συστημάτων Ανάλυση Πεδίου και Θεωρητική Τεκμηρίωση*. Πανεπιστήμιο Αιγαίου. Ανακτήθηκε 22/09/2015 από <https://pithos.oceanos.grnet.gr/public/7xEDw8vU20IWmbWMTUak02>.

Κριτήρια Αξιολόγησης

Κριτήριο αξιολόγησης 1

Ποια είναι η βασική διαφορά της αρχιτεκτονικής πελάτη-εξυπηρετητή με την αρχιτεκτονική thin-client;

Κριτήριο αξιολόγησης 2

Βρείτε στο διαδίκτυο την αρχιτεκτονική του συστήματος της Γενικής Γραμματείας Πληροφοριακών Συστημάτων με την επωνυμία taxis. Σχολιάστε την αρχιτεκτονική. Τι τύπου είναι;

Κριτήριο αξιολόγησης 3

Σε μια αρχιτεκτονική με εικονικές μηχανές, τι είναι ο hypervisor; Με τη χρήση του διαδικτύου βρείτε και δώστε παραδείγματα υλοποιήσεων hypervisor.

Κριτήριο αξιολόγησης 4

Κάθε έγγραφο που δημιουργείται στο MS-Word περιέχει μεταδεδομένα. Δώστε παραδείγματα μεταδεδομένων.

Κριτήριο αξιολόγησης 5

Μια από τις βασικές κατηγορίες κινδύνων που αντιμετωπίζει μια επιχείρηση είναι οι εσωτερικοί κίνδυνοι που προέρχονται από εργαζόμενους. Με τη χρήση του διαδικτύου βρείτε και δώστε παραδείγματα τέτοιων κινδύνων.

Κριτήριο αξιολόγησης 6

Οι παρακάτω κανόνες είναι οι κανόνες της οδηγίας WCAG 2.0. Η οδηγία WCAG 2.0 παρέχει κανόνες τεσσάρων κατηγοριών: α) το αντιληπτό περιεχόμενο, β) το λειτουργικό περιεχόμενο (π.χ. δυνατή χρήση όλων των λειτουργιών του συστήματος μέσω πληκτρολογίου), γ) την κατανόηση του περιεχόμενου (π.χ. δημιουργία ιστοσελίδων με προβλέψιμη διάταξη και λειτουργία) και δ) το εύρωστο περιεχόμενο (περιεχόμενο χωρίς λάθη).

Κατατάξτε τους παρακάτω αναλυτικούς κανόνες στις παραπάνω τέσσερις κατηγορίες:

- Βοηθάτε τους χρήστες να αποφεύγουν και να διορθώνουν τυχόν λάθη τους.
- Δημιουργήστε ιστοσελίδες με προβλέψιμη διάταξη και λειτουργία.
- Δημιουργήστε περιεχόμενο που μπορεί να αποδοθεί με διαφορετικούς τρόπους (για παράδειγμα με απλούστερη διάταξη), χωρίς απώλεια πληροφοριών ή δομής.
- Διευκολύνετε την οπτική και ηχητική αντίληψη του περιεχομένου από τους χρήστες, συμπεριλαμβανομένης της διάκρισης των πληροφοριών προσκηνίου από το παρασκήνιο.
- Ενισχύστε τη συμβατότητα με τρέχοντες και μελλοντικούς πράκτορες χρήστη, συμπεριλαμβανομένων των υποστηρικτικών τεχνολογιών.
- Καταστήστε δυνατή τη χρήση όλων των λειτουργιών μέσω πληκτρολογίου.
- Μην σχεδιάζετε περιεχόμενο με τρόπο που είναι γνωστό ότι προκαλεί επιληπτικές κρίσεις λόγω φωτοευαισθησίας.
- Παρέχετε εναλλακτικά κείμενα για κάθε περιεχόμενο που δεν διατίθεται σε μορφή κειμένου, ώστε να μπορεί να αποδοθεί σε άλλες μορφές που χρειάζονται οι χρήστες, όπως για παράδειγμα με μεγάλη γραμματοσειρά, σε Μπράιγ, σε ομιλία, με χρήση συμβόλων ή σε πιο απλή γλώσσα.
- Παρέχετε εναλλακτικές λύσεις για πολυμέσα που εξαρτώνται από τον χρόνο.

- Παρέχετε μηχανισμούς που βοηθούν τους χρήστες να εντοπίζουν περιεχόμενο, να προσανατολίζονται και να περιηγούνται σε αυτό.
- Παρέχετε στους χρήστες επαρκή χρόνο για την ανάγνωση και χρήση του περιεχομένου.
- Το σε μορφή κειμένου περιεχόμενο πρέπει να είναι αναγνώσιμο και κατανοητό.