Πανεπιστήμιο Πειραιώς - Τμήμα Πληροφορικής

ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΧΡΗΣΤΩΝ**
Ονοματεπώνυμο Φοιτητή	Κολυβάς Βασίλειος του Γεωργίου
Αριθμός Μητρώου	ΜΠΣΠ/07042
Κατεύθυνση	Δικτυοκεντρικά Πληροφοριακά Συστήματα
Υπεύθυνος/επιβλέπων Καθηγητής	Χρήστος Δουληγέρης, Καθηγητής

Ημερομηνία Παράδοσης ΜΑΙΟΣ 2010

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Όνομα Επώνυμο Βαθμίδα	Όνομα Επώνυμο Βαθμίδα	Όνομα Επώνυμο Βαθμίδα

1 Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον υπεύθυνο καθηγητή κ. Δουληγέρη Χρήστο που με τις πολύτιμες συμβουλές του και την καθοδήγησή του κατά την συγγραφή της διπλωματικής διατριβής με θέμα «υπηρεσία διαχείρισης χρηστών», με βοήθησε να ολοκληρώσω τις σπουδές μου στο Πανεπιστήμιο Πειραιά. Ανάγκη έχω ακόμα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για τη συμπαράστασή τους και την ηθική τους υποστήριξη, όλο αυτόν τον καιρό…

2 Περίληψη

Ο σκοπός της μεταπτυχιακής διατριβής μου είναι τη μελέτη, τεκμηρίωση και λειτουργία εξυπηρετητή υπηρεσιών καταλόγου (directory server) μέσω του πρωτοκόλλου Lightweight Directory Access Protocol (LDAP). Η υπηρεσία αυτή έχει σα βασικό ρόλο τη δημιουργία, διατήρηση, ενημέρωση και διαγραφή πληροφοριών (διευθύνσεις ηλεκτρονικού ταχυδρομείου, τηλέφωνα επικοινωνίας, κωδικοί πρόσβασης υπηρεσιών, κ.λπ.) για τους πολίτες που σχετίζονται άμεσα ή έμμεσα με τα Ακαδημαϊκά Ιδρύματα της χώρας. Αυτές θα μπορούν να χρησιμοποιηθούν για την ταυτοποίηση και την εξουσιοδότηση των χρηστών σε κεντρικά υπολογιστικά συστήματα, αλλά και στη διάθεση σε αυτούς συγκεκριμένων υπηρεσιών. Η υπηρεσία διαχείρισης των χρηστών προσφέρει τα εργαλεία για την δημιουργία προσωπικών λογαριασμών στους χρήστες και ορίζει το μηχανισμό πιστοποίησης της ταυτότητας του χρήστη και το σύνολο των προσωπικών υπηρεσιών που μπορεί να έχει στην διάθεση του μετά την πιστοποίηση αυτή.

Κατ' επέκταση, στόχος της εργασίας είναι η παρουσίαση του τρόπου δημιουργίας μιας δικτυακής εφαρμογής γραμμένης σε PHP η οποία μπορεί να επικοινωνεί με την βάση δεδομένων μέσω του πρωτοκόλλου SOAP.
Η υπηρεσία καταλόγου θα πρέπει να καλύψει τις απαιτήσεις αποθήκευσης χαρακτηριστικών χρηστών των Ακαδημαϊκών Ιδρυμάτων, του οποίου το πλήθος μπορεί να υπερβαίνει κατά πολύ τις δυνατότητες των αντίστοιχων σε λειτουργία υποδομών τους. Ειδικότερα μία τέτοια επέκταση της υποδομής σε νέες οντότητες πρακτικά οδηγεί σε επανασχεδίαση της υπηρεσίας

Παραδείγματα δικτυακών υπηρεσιών είναι προβολή των προσωπικών στοιχείων των φοιτητών και των καθηγητών (όνομα, διεύθυνση, Ίδρυμα, σχολή, τμήμα, Αριθμός Μητρώου κ.τ.λ.) από το σύστημα. Ο τρόπος με τον οποίο γίνεται η επικοινωνία Πανεπιστημίου – πελάτη είναι οι υπηρεσίες ιστού μέσω του πρωτοκόλλου SOAP.

Συγκεκριμένα, δημιουργήθηκε μία βάση δεδομένων και μία δικτυακή εφαρμογή που χρησιμοποιεί την τεχνολογία WSDL (Web Service Description Language) "μία γλώσσα σε XML μορφή η οποία περιγράφει απόλυτα μία web service" και το UDDI (Universal Description, Discovery, and Integration) το οποίο αποτελεί ένα πρωτόκολλο καταχώρησης για τις υπηρεσίες ιστού με σκοπό να δημιουργηθούν οι προσφερόμενες υπηρεσίες.

Εκτός των παραπάνω χρησιμοποιήθηκε πληθώρα προγραμματιστικών εργαλείων τα περισσότερα των οποίων προσφέρονται δωρεάν από διάφορους φορείς του Διαδικτύου και αποτελούν καθιερωμένα εργαλεία τα οποία χρησιμοποιούνται από πολλές επιχειρήσεις που δραστηριοποιούνται στον τομέα των δικτυακών εφαρμογών. Επιπλέον, η πλειοψηφία των προγραμμάτων αυτών είναι open source και δίνεται ως συνέπεια η ευκαιρία στο χρήστη να προσθέσει λειτουργίες σε αυτά και να τα ταιριάξει στην εφαρμογή του. Για την παρακάτω πτυχιακή χρησιμοποιήθηκε κώδικας από τον ιστό-χώρο sourceforge.net/projects/lums.

Λέξεις κλειδιά: web services, δικτυακή υπηρεσία, δικτυακή εφαρμογή, πρωτόκολλο SOAP, PHP, nuSOAP, LDAP, εξυπηρετητής εφαρμογών, δικτυακός εξυπηρετητής, εξυπηρετητής βάσεων δεδομένων, υπηρεσίες ιστού

3 Πίνακας Περιεχομένων

1 Ευχαριστίες....2
2 Περίληψη....3
3 Πίνακας Περιεχομένων....4
4 Εισαγωγή....6
4.1 Αντικείμενο της μεταπτυχιακής διατριβής....6
4.2 Η εξέλιξη των πρωτοκόλλων καταλόγων....6
4.2.1 Τι είναι το DAP (Directory Access Protocol);....6
4.2.2 Τι είναι το LDAP (Lightweight Directory Access Protocol);....7
5 Επισκόπηση της τεχνολογίας LDAP....7
5.1 Κατάλογοι....7
5.2 Επισκόπηση του Πρωτοκόλλου LDAP....9
5.3 Οι λειτουργίες του LDAP (αναλυτικά)....10
5.3.1 StartTLS....10
5.3.2 Bind (authenticate)....11
5.3.3 Search and Compare....11
5.3.4 Update Data....12
5.3.5 Extended operations....12
5.3.6 Abandon....12
5.3.7 Unbind....12
5.3.8 LDAP Αντιγραφή (replication)....13
5.4 OpenLDAP....13
5.5 Προβλήματα στην πραγματοποίηση αλλαγών μέσω LDAP....14
5.6 Διαφορές μεταξύ του πρωτοκόλλου LDAP και βάσεων Δεδομένων (RDBMS)....15
5.7 Τεχνικές διαφορές μεταξύ του LDAP και βάσεων Δεδομένων....16
5.8 Ορατότητα δεδομένων....16
5.9 Συγχρονισμός δεδομένων....17
6 Υπηρεσίες ιστού (Web Services)....17
6.1 Επισκόπηση της τεχνολογίας των υπηρεσιών ιστού....17
6.1.1 Remote procedure call....19
6.1.2 Message passing....19
6.2 Αρχιτεκτονικά μοντέλα των υπηρεσιών ιστού....19
6.3 Πλεονεκτήματα της αρχιτεκτονικής των υπηρεσιών ιστού....23
6.3.1 Πλεονεκτήματα της χρήσης υπηρεσιών ιστού....23
7 Ανάπτυξη συστήματος....24
7.1 Βασικές απαιτήσεις προτεινόμενου συστήματος....24
7.2 Διαθέσιμα λογισμικά....25
7.3 Τελική επιλογή συστήματος....27
7.4 Πλατφόρμα λογισμικού....27
7.5 Διεπαφή (interface)....28
7.5.1 Αρχιτεκτονική....28
7.5.2 Πλατφόρμα λογισμικού....29
7.6 Περιγραφή nuSoap API....29
7.6.1 Παράδειγμα πελάτη για υπηρεσίες ιστού....30
8 Πιστοποίηση και Πολιτική Ασφάλειας....32
8.1 Προτάσεις Βελτίωσης....33
8.1.1 Versioning ερωτημάτων/απαντήσεων....34
8.1.2 Uniform Resource Identifier (URI)....35

8.2 Ουρές Αναμονής 37
9 Ολοκληρωμένη Υπηρεσία 38
9.1 Αρχιτεκτονική υπηρεσίας 38
9.1.1 Βιβλιοθήκη λειτουργιών LUMS 39
9.1.2 Συναρτήσεις 39
9.1.3 Μορφή Αρχείων 41
9.1.4 Αρχείο Διαμόρφωσης 42
9.1.5 Constant 44
9.1.6 Callfunc 44
9.1.7 Autoincrement 44
9.1.8 Unique 45
9.1.9 Mapping 45
9.2 Παράδειγμα Αρχείου Διαμόρφωσης 45
9.2.1 Περιορισμοί Τιμών 45
9.2.2 Συναρτήσεις που χρησιμοποιούνται 45
10 Έλεγχοι λειτουργίας συστήματος 46
10.1 Κεντρική υπηρεσία καταλόγου 46
10.2 Υπηρεσίες ιστού 47
10.3 Επικοινωνία με εφαρμογή διαχείρισης χρηστών 48
11 Εγκατάσταση 49
11.1 Εγκαθιστούμε και εκκινούμε τον Apache (httpd) στο Linux 49
11.2 Εγκαθιστούμε την PHP (Hypertext Preprocesso) 49
11.3 Εγκαθιστούμε την PECL και την APC 50
11.4 Εγκαθιστούμε τον openldap server και ενεργοποιούμε την php για χρήση στον ldap: 50
11.4.1 Παραμετροποίηση του LDAP 50
11.5 Τώρα εκκινούμε τον LDAP 51
12 ΠΑΡΑΡΤΗΜΑ Ι 52
13 ΠΑΡΑΡΤΗΜΑ ΙΙ 54
14 ΠΑΡΑΡΤΗΜΑ ΙΙΙ 55
15 ΑΚΡΩΝΥΜΙΑ 56
16 ΑΝΑΦΟΡΕΣ 57

4 Εισαγωγή

4.1 Αντικείμενο της μεταπτυχιακής διατριβής

Οι πλατφόρμες προγραμματισμού για την υλοποίηση δικτυακών εφαρμογών αυξάνονται, και κάθε μία έχει τα πλεονεκτήματα και τα μειονεκτήματά της. Οι δύο κυριότερες πλατφόρμες είναι η πλατφόρμα .NET της Microsoft και η πλατφόρμα Java 2 Enterprise Edition (J2EE) της Sun Microsystems. Η J2EE προηγήθηκε της .NET ενώ η τελευταία αποτελεί ενοποίηση των ήδη υλοποιημένων πλατφόρμων της Microsoft σε ένα ενιαίο πακέτο.
Συγκρίνοντας τις δύο πλατφόρμες, οι υπέρμαχοι της Microsoft χρησιμοποιούν το πακέτο .NET λόγω της ταχύτητας των εκτελέσιμων εφαρμογών, της συμβατότητάς του με παλαιότερες τεχνολογίες της Microsoft και άλλων Διαδικτυακών τεχνολογιών και της πολύ καλής βιβλιογραφίας του υπογεγραμμένης από τη μεγαλύτερη σήμερα εταιρία λογισμικού στον κόσμο. Στην αντίπερα όχθη, οι φανατικοί της Java μιλούν για την πληθώρα βιβλιοθηκών και προγραμματιστικών εργαλείων που παρέχει η Sun και οι επίσημοι ή ανεπίσημοι συνεργάτες της με κυριότερο πλεονέκτημα ότι τα περισσότερα είναι δωρεάν προς χρήση και καλύπτουν σχεδόν όλες τις πτυχές του σύγχρονου κόσμου της πληροφορίας από προσωπικούς υπολογιστές και δικτυακούς εξυπηρετητές μέχρι κινητά τηλέφωνα και ρομποτική. Βέβαια δεν πρέπει να παραγκωνιστεί το γεγονός ότι η Java τρέχει πάνω από διαφορετικές πλατφόρμες (cross platform), δηλαδή δεν εξαρτάται από το λειτουργικό σύστημα στο οποίο εκτελείται (να σημειωθεί ότι και από πλευράς Microsoft γίνονται προσπάθειες για την μεταφορά του .NET σε άλλα λειτουργικά συστήματα, χωρίς όμως να είναι σίγουρο ότι θα υλοποιηθεί και εμπορικά).
Από τη μεριά του συγγραφέα, όπως προαναφέρθηκε δεν επιλέχτηκε καμία από τις παραπάνω πλατφόρμες, αλλά ο ορισμός του interface της πτυχιακής έγινε με τη χρήση της γλώσσας προδιαγραφής Web Services WSDL (Web Service Description Language) την XML γλώσσα δηλαδή που έχει προδιαγραφεί από το W3 Consortium για την υλοποίηση και περιγραφή των υπηρεσιών ιστού. Η υλοποίηση των υπηρεσιών ιστού πραγματοποιήθηκε με τη χρήση SOAP πάνω από το πρωτόκολλο HTTP, παρέχοντας έτσι συμβατότητα με την πλειοψηφία των αντίστοιχων πακέτων λογισμικού (.Net FrameWork, Java κτλ).
Στόχος της παρούσας εργασίας είναι ο σχεδιασμός ενός κατανεμημένου συστήματος παροχής υπηρεσιών και ανταλλαγής δεδομένων βασισμένου στο πρωτόκολλο SOAP. Η δικτυακές εφαρμογές σήμερα χρησιμοποιούνται κυρίως στον εμπορικό τομέα ενώ στην παρούσα εργασία θα παρουσιαστεί η εφαρμογή τους στις υπηρεσίες εκπαίδευσης.

4.2 Η εξέλιξη των πρωτοκόλλων καταλόγων

4.2.1 Τι είναι το DAP (Directory Access Protocol);

Το Directory Access Protocol (DAP) είναι ένα πρότυπο δικτύωσης υπολογιστών που εκδόθηκε από την ITU-T και το πρότυπο ISO το 1988 για την πρόσβαση σε μια υπηρεσία καταλόγου X.500 (directory service). Το πρωτόκολλο DAP επρόκειτο να χρησιμοποιηθεί από τα συστήματα υπολογιστή-πελάτη, αλλά δεν έγινε ποτέ δημοφιλές αφού υπήρχαν λίγες εφαρμογές που χρησιμοποιούσαν την πλήρη στοίβα πρωτοκόλλου του OSI για επιτραπέζιους υπολογιστές και συνάμα δεν διέθεταν το κατάλληλο υλικό και τα λειτουργικά συστήματα για να λειτουργήσουν με αυτό. Οι βασικές λειτουργίες του πρωτοκόλλου DAP: Read, List, Search, Compare, Modify, Add, Delete και ModifyRDN, έπειτα προσαρμόστηκαν στο Novell Directory Services (NDS) και στο Lightweight Directory Access Protocol (LDAP).
Το πρότυπο X.500 που προτάθηκε από τον οργανισμό ISO (International Standards Organisation) ήταν αυτό που προτάθηκε αρχικά και χρησιμοποιήθηκε για την καθιέρωση μιας παγκόσμιας ονοματολογίας καταλόγου. Καθιέρωσε ένα ανοιχτό πρωτόκολλο επικοινωνίας, το Directory Access Protocol (DAP) το οποίο μπορούσε να χρησιμοποιηθεί από οποιαδήποτε εφαρμογή για την πρόσβαση στον κατάλογο. Το X.500 είχε αρκετά πλεονεκτήματα, όπως για παράδειγμα υποστήριζε τη δημιουργία ενός επεκτάσιμου σχήματος στον κατάλογο ανάλογα με τις ανάγκες, και στο οποίο μπορούσε να αποθηκευτεί οποιαδήποτε μορφή πληροφορίας. Το κύριο μειονέκτημά του όμως, ήταν ότι προσέφερε ένα επικοινωνιακό πρωτόκολλο το

οποίο δεν βασιζόταν στο πρωτόκολλο TCP/IP, ενώ επιπλέον παρουσίαζε σημαντική πολυπλοκότητα στη διαχείριση της ονοματολογίας του καταλόγου.
Αναλυτικά, το X.500 είναι μια σειρά από πρότυπα δικτύωσης ηλεκτρονικών υπολογιστών που καλύπτουν υπηρεσίες καταλόγου. Η σειρά X.500 αναπτύχθηκε από την ITU-T, παλαιότερα γνωστή ως CCITT. Οι υπηρεσίες καταλόγου αναπτύχθηκαν με σκοπό να υποστηρίξουν τις απαιτήσεις της X.400 υπηρεσίας καταλόγου (την ηλεκτρονική ανταλλαγή αλληλογραφίας).
Τα πρωτόκολλα που ορίζονται από το X.500 περιλαμβάνουν τα ακόλουθα:

DAP (Directory Access Protocol)
DSP (Directory System Protocol)
DISP (Directory Information Shadowing Protocol)
DOP (Directory Operational Bindings Management Protocol)

Επειδή αυτά τα πρωτόκολλα χρησιμοποιούσαν το πρωτόκολλο δικτύωσης της OSI στοίβας, αναπτύχθηκαν μια σειρά εναλλακτικών λύσεων για το DAP με σκοπό να επιτρέψουν στους πελάτες να έχουν πρόσβαση στον κατάλογο X.500, χρησιμοποιώντας το πρωτόκολλο TCP / IP. Η πιο γνωστή εναλλακτική λύση για το DAP είναι το Lightweight Directory Access Protocol (LDAP). Το LDAP παραμένει ένα δημοφιλές πρωτόκολλο πρόσβασης καταλόγου για τον λόγο ότι χρησιμοποιεί τις λειτουργίες του DAP και των υπολοίπων πρωτοκόλλων X.500 μέσω του πρωτοκόλλου TCP / IP.

4.2.2 Τι είναι το LDAP (Lightweight Directory Access Protocol);

Το πρωτόκολλο LDAP (Lightweight Directory Access Protocol) που καθιερώθηκε στη συνέχεια, διατήρησε τα θετικά στοιχεία του X.500, ενώ ταυτόχρονα παρείχε ένα επικοινωνιακό πρωτόκολλο βασισμένο στο TCP/IP. Προτάθηκε αρχικά το 1996 από την Netscape Communications Corp., το Πανεπιστήμιο του Michigan και περισσότερες από 40 εταιρίες, ως ένα ανοιχτό πρότυπο για τις Υπηρεσίες Καταλόγου στο Internet. Ο όρος Lightweight στην ονομασία του, σημαίνει ότι συγκρινόμενο με το DAP περιέχει λιγότερο κώδικα και πιο απλές συναρτήσεις, και συνεπώς είναι για τους κατασκευαστές πιο εύκολα υλοποιήσιμο και μικρότερο σε κόστος.
Πιο συγκεκριμένα το Lightweight Directory Access Protocol (LDAP) είναι ένα πρωτόκολλο της υπηρεσίας καταλόγου που τρέχει πάνω ακριβώς από την στοίβα TCP / IP. Το μοντέλο των πληροφοριών (τόσο για τα δεδομένα όσο και για τα σύνολα των χαρακτηριστικών, namespaces) του LDAP είναι παρόμοια με εκείνη της υπηρεσίας X.500 του OSI καταλόγου, αλλά με λιγότερες δυνατότητες και χαμηλότερες απαιτήσεις πόρων από το X.500. Σε αντίθεση με τα περισσότερα άλλα πρωτόκολλα του Internet, το LDAP έχει μια συνδεδεμένη API που απλοποιεί τις αιτήσεις εγγραφής του καταλόγου. Το LDAP API χρησιμοποιείται για την διαχείριση του καταλόγου και για την περιήγηση σε εφαρμογές που δεν έχουν τον κατάλογο υποστήριξης υπηρεσιών ως κύρια λειτουργία τους. Το LDAP δεν μπορεί να δημιουργήσει καταλόγους ή να καθορίσει πώς μια υπηρεσία καταλόγου λειτουργεί.

5 Επισκόπηση της τεχνολογίας LDAP

5.1 Κατάλογοι

Οι κατάλογοι είναι παρόμοιοι με τις βάσεις δεδομένων, μόνο που περιέχουν περισσότερο περιγραφική πληροφορία. Τα περιεχόμενα τους στηρίζονται στην έννοια της κλάσης (class), της εγγραφής (entry) και των χαρακτηριστικών (attributes). Η οργάνωση των πληροφοριών ακολουθεί το ιεραρχικό (σε αντίθεση με το σχεσιακό) μοντέλο. Η πληροφορία είναι οργανωμένη σε δεντρική δομή με τα δεδομένα να περιέχονται σε εγγραφές στα φύλλα του δέντρου πληροφοριών. Κάθε εγγραφή είναι πλήρης και περιέχει το σύνολο των δεδομένων που αντιστοιχούν σε αυτή (σε αντίθεση με τις ιεραρχικές βάσεις δεδομένων όπου τα στοιχεία είναι διασκορπισμένα ανάμεσα σε πολλαπλούς πίνακες δεδομένων). Οι πληροφορίες που περιέχει ο κατάλογος τείνουν να γίνουν συνεχώς περισσότερες. Αρχικά περιείχαν τις

εγγραφές φυσικών προσώπων με τα χαρακτηριστικά (στοιχεία) τους, χαρακτήρας που προφανώς κληρονομήθηκε από προγενέστερες μορφές και διαδικασίες αποθήκευσης (π.χ. τηλεφωνικοί κατάλογοι). Η τάση όμως είναι να ενσωματώνουν όσο το δυνατό περισσότερες πληροφορίες, τόσο για φυσικά πρόσωπα (με ολοκληρωμένα προφίλ και πληροφορία προσαρμοσμένη σε ειδικές ανάγκες) όσο και για οτιδήποτε αφορά ένα υπολογιστικό σύστημα ή ένα δίκτυο (συσκευές, εφαρμογές), αφενός προκειμένου να είναι προσπελάσιμες οι πληροφορίες από όσους ενδιαφέρονται, αφετέρου προκειμένου να γίνεται διαχείριση των πληροφοριών αυτών και των δικαιωμάτων πρόσβασης.
Είναι προφανές ότι όσο μεγαλύτερο είναι το δίκτυο τόσο μεγαλύτερη γίνεται και η ανάγκη για τη χρήση καταλόγων. Και αυτό γιατί είναι περισσότεροι οι πόροι και τα άτομα που χρειάζεται κανείς να προσπελάσει και να εντοπίσει. Επίσης είναι μεγαλύτερη η ανάγκη για προτυποποιημένες (standardized) μεθόδους προσπέλασης και επικοινωνίας των καταλόγων αυτών.
Έχοντας υπόψη την ανάγκη για ένα παγκόσμιο κατάλογο όπου κάθε στοιχείο του θα έχει και ένα μοναδικό όνομα (όπως κάθε υπολογιστής έχει μια μοναδική διεύθυνση) έπρεπε να βρεθεί μια ονοματολογία που θα ικανοποιούσε το σκοπό αυτό. Με βάση παρόμοιες περιπτώσεις σχεδιάστηκε μια ιεραρχική ονοματολογία που θα αντανακλούσε την ιεραρχική οργάνωση των ευρετηρίων. Η ονοματολογία αυτή προτυποποιήθηκε μέσω του πρωτοκόλλου X.500-DAP (Directory Access Protocol) που σχεδιάστηκε από τον ISO (International Standards Organisation) και ενσωματώθηκε στο πιο πρόσφατο LDAP (Lightweight Directory Access Protocol) που σχεδιάστηκε από την Internet κοινότητα και όπως γίνεται σε ανάλογες περιπτώσεις προτάθηκε μέσω των RFCs (Request for Comments).

Το πρωτόκολλο LDAP ορίζεται από τα παρακάτω αρχεία Request For Comments [9]:

- RFC 4510 - Lighweight Directory Access Protocol (LDAP) Technical Specification Roadmap (replaced the previous LDAP Technical specification, RFC 3377, in its entirety)
- RFC 4511 - Lightweight Directory Access Protocol (LDAP): The Protocol
- RFC 4512 - Lightweight Directory Access Protocol (LDAP): Directory Information Models
- RFC 4513 - Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms
- RFC 4514 - Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names
- RFC 4515 - Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters
- RFC 4516 - Lightweight Directory Access Protocol (LDAP): Uniform Resource Locator RFC 4517 - Lightweight Directory Access Protocol (LDAP): Syntaxes and Matching Rules
- RFC 4518 - Lightweight Directory Access Protocol (LDAP): Internationalized String Preparation
- RFC 4519 - Lightweight Directory Access Protocol (LDAP): Schema for User Applications

Οι LDAPv3 extensions ορίζονται από τα παρακάτω αρχεία Request For Comments:

- RFC 2247 - Use of DNS domains in distinguished names
- RFC 2307 - Using LDAP as a Network Information Service
- RFC 2589 - LDAPv3: Dynamic Directory Services Extensions
- RFC 2649 - LDAPv3 Operational Signatures
- RFC 2696 - LDAP Simple Paged Result Control
- RFC 2798 - inetOrgPerson LDAP Object Class
- RFC 2849 - The LDAP Data Interchange Format (LDIF)
- RFC 2891 - Server Side Sorting of Search Results
- RFC 3045 - Storing Vendor Information in the LDAP root DSE
- RFC 3062 - LDAP Password Modify Extended Operation
- RFC 3296 - Named Subordinate References in LDAP Directories
- RFC 3671 - Collective Attributes in LDAP
- RFC 3672 - Subentries in LDAP
- RFC 3673 - LDAPv3: All Operational Attributes

- RFC 3687 - LDAP Component Matching Rules
- RFC 3698 - LDAP: Additional Matching Rules
- RFC 3829 - LDAP Authorization Identity Controls
- RFC 3866 - Language Tags and Ranges in LDAP
- RFC 3909 - LDAP Cancel Operation
- RFC 3928 - LDAP Client Update Protocol
- RFC 4370 - LDAP Proxied Authorization Control
- RFC 4373 – LBURP
- RFC 4403 - LDAP Schema for UDDI
- RFC 4522 - LDAP: Binary Encoding Option
- RFC 4523 - LDAP: X.509 Certificate Schema
- RFC 4524 - LDAP: COSINE Schema (replaces RFC 1274)
- RFC 4525 - LDAP: Modify-Increment Extension
- RFC 4526 - LDAP: Absolute True and False Filters
- RFC 4527 - LDAP: Read Entry Controls
- RFC 4528 - LDAP: Assertion Control
- RFC 4529 - LDAP: Requesting Attributes by Object Class
- RFC 4530 - LDAP: entryUUID
- RFC 4531 - LDAP Turn Operation
- RFC 4532 - LDAP Who am I? Operation
- RFC 4533 - LDAP Content Sync OperationLDAPv2 was specified in the following RFCs:
- RFC 1777 - Lightweight Directory Access Protocol (replaced RFC 1487)
- RFC 1778 - The String Representation of Standard Attribute Syntaxes (replaced RFC 1488)
- RFC 1779 - A String Representation of Distinguished Names (replaced RFC 1485)

Σχετικά RFCs:

- RFC 2254 - The String Representation of LDAP Search Filters
- RFC 4520 (also BCP 64) - Internet Assigned Numbers Authority (IANA) Considerations for the Lightweight Directory Access Protocol (LDAP) (replaced RFC 3383)
- RFC 4521 (also BCP 118) - Considerations for Lightweight Directory Access Protocol (LDAP) Extensions

Το πρωτόκολλο LDAP είναι ένα ασφαλές πρωτόκολλο, το οποίο χρησιμοποιεί τον έλεγχο ταυτότητας για να διασφαλίσει ότι η μεταφορά των δεδομένων είναι ασφαλής. Ο έλεγχος αυτός χρησιμοποιείται από τον διακομιστή, ο οποίος ελέγχει την ταυτότητα του πελάτη. Η τρέχουσα έκδοση 3 του LDAP χρησιμοποιεί το SASL (Simple Authentication and Security Layer -SASL), χαρακτηριστικό το οποίο παρέχει μια ευελιξία στην επιλογή του μηχανισμού ελέγχου ταυτότητας. Το πρωτόκολλο Secure Socket Layer (SSL), είναι το πιο δημοφιλές για το σκοπό αυτό, παρέχοντας ένα υψηλό επίπεδο ασφάλειας.

5.2 Επισκόπηση του Πρωτοκόλλου LDAP

Ένας πελάτης αρχίζει μια συνεδρία LDAP όταν συνδέεται στον εξυπηρετητή, η προεπιλεγμένη πόρτα TCP είναι η 389. Ο πελάτης τότε στέλνει ένα αίτημα λειτουργίας στον εξυπηρετητή, και με την σειρά του ο εξυπηρετητής απαντάει. Αυτή η διαδικασία δεν είναι απαραίτητο να είναι σειριακή καθώς πέραν μερικών εξαιρέσεων ο πελάτης δεν χρειάζεται να περιμένει για μια απάντηση συγκεκριμένη από τον εξυπηρετητή αλλά και οι απαντήσεις του εξυπηρετητή μπορεί να στέλνονται με οποιαδήποτε σειρά.

Ο πελάτης συγκεκριμένα περνάει από τις παρακάτω διεργασίες:

- Start TLS — χρησιμοποιεί την επέκταση του LDAPv3 Transport Layer Security (TLS) για να δημιουργήσει μια ασφαλή σύνδεση.
- Bind — αυθεντικοποιεί και καθορίζει την έκδοση του πρωτοκόλλου LDAP
- Search — ψάχνει και ανακτά τις εγγραφές του καταλόγου.

- Compare — τεστάρει εάν μια εγγραφή περιέχει την δοσμένη ιδιότητα τιμής.
- Προσθέτει μια νέα εγγραφή
- Διαγράφει μια εγγραφή
- Μετατρέπει μια εγγραφή
- Modify Distinguished Name (DN) — μεταφέρει και μετατρέπει τις εγγραφές
- Abandon — εγκαταλείπει μια προηγούμενη αίτηση
- Extended Operation — μια γενική λειτουργία που χρησιμοποιείται για να καθορίσει άλλες λειτουργίες του συστήματος
- Unbind — κλείνει την σύνδεση του πελάτη με τον διακομιστή / εξυπηρετητή

Επιπρόσθετες λειτουργίες μπορεί να χαρακτηριστούν οι "Unsolicited Notifications" που στέλνει ο εξυπηρετητής προς τους πελάτες που δεν καταγράφονται σαν απαντήσεις σε κανένα αίτημα, παραδείγματος χάρη πριν κλείσει μια σύνδεση με τον πελάτη λόγω λήξης του χρονικού ορίου.

Μια κοινή διαφορετική μέθοδος για να ασφαλίσουμε την υπηρεσία επικοινωνίας με τον LDAP είναι να χρησιμοποιήσουμε SSL tunnel από τον εξυπηρετητή μέχρι τον πελάτη. Το παραπάνω δηλώνεται στα LDAP URLs και η προεπιλεγμένη πόρτα που χρησιμοποιείται στο SSL είναι η 636. Η χρήση του LDAP πάνω από SSL είναι κοινή στην έκδοση 2 (LDAPv2) αλλά δεν ήταν ποτέ τυποποιημένη σε κάθε επίσημη προδιαγραφή. Αυτή η χρήση υποβιβάστηκε συνολικά μαζί με το LDAPv2, το οποίο επίσημα αποσύρθηκε το 2003.

Γενικά το LDAP καθορίζεται με όρους ASN.1 και τα μηνύματα του πρωτοκόλλου κωδικοποιούνται σε δυαδική μορφή Βασικών Κανόνων Κωδικοποίησης (Basic Encoding Rules – BER). Επίσης χρησιμοποιεί αναπαράσταση βασισμένη σε κείμενο για ένα αριθμό από ASN.1 πεδία και τύπους.

Στις τηλεπικοινωνίες και στα δίκτυα υπολογιστών το Abstract Syntax Notation One (ASN.1) είναι ένα πρότυπο και ένα ευέλικτο σύστημα χαρακτήρων που περιγράφει δομές δεδομένων για αναπαράσταση, κωδικοποίηση, αναμετάδοση και αποκωδικοποίηση των δεδομένων. Αυτό παρέχει ένα σύνολο από τυπικούς κανόνες που περιγράφουν την δομή των αντικειμένων. Παράλληλα οι Basic Encoding Rules (BER) είναι ένα από τα format κωδικοποίησης που καθορίζονται σαν ένα μέρος του ASN.1 πρότυπο όπως καθορίζεται από την ITU στο X.690.

5.3 Οι λειτουργίες του LDAP (αναλυτικά)

Ο πελάτης δίνει σε κάθε αίτηση ένα Message ID. Ο εξυπηρετητής χρησιμοποιεί το ίδιο Message ID. Η απάντηση συμπεριλαμβάνει ένα αριθμητικό καταληκτικό κώδικα που δηλώνει επιτυχία, κάποια κατάσταση λάθους ή κάποια άλλη ιδιαίτερη περίπτωση. Πριν την απάντηση ο εξυπηρετητής μπορεί να στέλνει άλλα μηνύματα με άλλα αποτελέσματα, για παράδειγμα εάν ψάχνει για μια κάποια εγγραφή και βρεθεί η εγγραφή αυτή από την διαδικασία της Εύρεσης (Search), επιστρέφεται από το παραπάνω μήνυμα.

5.3.1 StartTLS

Η παραπάνω διαδικασία εγκαθιδρύει την «Transport Layer Security» τον απόγονο του SSL στη νέα σύνδεση που δημιουργείται. Αυτή η ασφάλεια μπορεί να παρέχει εμπιστευτικότητα και ακεραιότητα στα δεδομένα που ανταλλάσσονται με τον εξυπηρετητή. Κατά την διάρκεια της διαπραγμάτευσης του TLS ο εξυπηρετητής στέλνει το πιστοποιητικό X.509 για να αποδείξει την ταυτότητά του. Ο πελάτης μπορεί επίσης να αποστείλει και αυτός το πιστοποιητικό της ταυτότητάς του. Μετά από αυτήν την διαδικασία ο πελάτης μπορεί να χρησιμοποιήσει το SASL/EXTERNAL (Simple Authentication and Security Layer) με σκοπό να έχει αυτήν την ταυτότητα για να την χρησιμοποιεί σε κάθε μία απόφαση που παίρνει για θέματα που αφορούν αυθεντικοποίηση στον LDAP.
Οι εξυπηρετητές επίσης συνήθως υποστηρίζουν το πρωτόκολλο «LDAPS» (Secure LDAP, που ως συνήθως είναι γνωστό και ως LDAP πάνω από SSL) σε μια ξεχωριστή πόρτα που όπως προαναφέραμε είναι η 636. Το LDAPS διαφέρει από το απλό LDAP σε δύο μέρη: 1) κατά την σύνδεση, ο πελάτης και ο εξυπηρετητής εγκαθιδρύουν TLS πριν οποιοδήποτε

μήνυμα LDAP μεταφερθεί από το μέσο (στο LDAPS δεν υπάρχει η ενέργεια StartTLS) , και 2) η σύνδεση του LDAPS πρέπει να κλείσει όταν τερματιστεί η TLS.
Το LDAPS χρησιμοποιήθηκε αρχικά με το LDAPv2 γιατί η λειτουργία StartTLS δεν είχε ακόμα καθοριστεί. Τώρα η χρήση του LDAPS έχει προ πολλού εγκαταλειφτεί και όλα τα νέα λογισμικά χρησιμοποιούν την StartTLS.

5.3.2 Bind (authenticate)

Η διαδικασία αυτή αυθεντικοποιεί τον πελάτη στον εξυπηρετητή.

"Simple Bind" Η απλή αυτή διαδικασία μπορεί να στέλνει απλά το DN (Distinguished Name) και τον κωδικό (password) ως απλό κείμενο, έτσι ώστε η σύνδεση μπορεί να προστατευτεί από TLS. Ο εξυπηρετητής απλά ελέγχει τον εισηγημένο κωδικό έναντι του χαρακτηριστικού userPassword στην συγκεκριμένη καταχώρηση.

"Anonymous Bind" Η ανώνυμη διαδικασία (με κενό DN και password) επανατοποθετεί την σύνδεση σε ανώνυμη κατάσταση.

"SASL Bind" (Simple Authentication and Security Layer) προσφέρει υπηρεσίες αυθεντικοποίησης με πολλούς διαφορετικούς τρόπους, όπως π.χ. το Kerberos ή το πιστοποιητικό του πελάτη να στέλνεται με TLS.

Κατά την διαδικασία bind επίσης στέλνεται η έκδοση του LDAP, κανονικά οι πελάτες χρησιμοποιούν την έκδοση του LDAPv3. Στο LDAPv3 δεν είναι απαραίτητη η διαδικασία του Bind ενώ στο LDAPv2 είναι προαπαιτούμενη.

5.3.3 Search and Compare

Η διαδικασία αυτή χρησιμοποιείται για να ψάξει και να διαβάσει τις εγγραφές σε μια βάση LDAP. Οι παράμετροι που χρησιμοποιούνται είναι οι παρακάτω:

baseObject
Σε αυτήν την παράμετρο καθορίζεται το DN (Distinguished Name) της εγγραφής από το οποίο δύναται να ξεκινήσει ο έλεγχος.

scope
Αυτό που βρίσκεται κάτω από το baseObject όταν ψάχνουμε. Π.χ. μπορεί να είναι το BaseObject δηλαδή μια ονοματισμένη εγγραφή, όπου αυτό χρησιμοποιείται για να διαβάσουμε μόνο μια εγγραφή, μπορεί να είναι το singleLevel όπου είναι οι εγγραφές ακριβώς κάτω από το DN, και τέλος wholeSubtre όπου αυτό έχει ολόκληρο το υποδένδρο αρχίζοντας από την βάση του DN.

filter

Το κριτήριο που χρησιμοποιείται όταν επιλέγεται κάποιο στοιχείο μέσα στο scope. Για παράδειγμα θεωρούμε το παρακάτω filter:

(&(objectClass=person)(|(givenName=Mike)(mail=mike*)))

Αυτό θα επιλέξει τα "persons" ως στοιχεία του objectClass person ,
που είτε έχουν το όνομα " Mike " ή ένα e-mail που αρχίζει με την συμβολοσειρά "mike ".

derefAliases
Χρησιμοποιείται όπου απαιτείται να οδηγηθεί σε μια άλλη εγγραφή. (alias) όπου μια εγγραφή δείχνει μια άλλη.

attributes
Ποιες ιδιότητες χρειάζονται να επιστραφούν στις εγγραφές των αποτελεσμάτων.

sizeLimit, timeLimit

Ο μέγιστος αριθμός των εγγραφών που χρειάζεται να επιστραφούν από μια αναζήτηση και ο μέγιστος χρόνος που επιτρέπεται να τρέξει η αναζήτηση στην βάση.

typesOnly

Αυτή η παράμετρος επιτρέπει να επιστρέφονται ιδιότητες που περιέχουν μόνο τύπους και όχι τιμές.

Ο εξυπηρετητής επιστρέφει τις εγγραφές που ταιριάζουν και τις ενδεχόμενες συνεχιζόμενες αναφορές. Αυτές μπορεί να παρουσιαστούν με κάθε σειρά και το τελικό αποτέλεσμα θα περιλαμβάνει τον κώδικα των αποτελεσμάτων.

Η διαδικασία Compare παίρνει ένα DN, ένα όνομα ιδιότητας και μια τιμή ιδιότητας και ελέγχει εάν η ονοματισμένη εγγραφή περιέχει αυτή την ιδιότητα με την συγκεκριμένη τιμή.

5.3.4 Update Data

Add, Delete, και Modify DN – όλα αυτά αλλάζουν το DN μιας εγγραφής.

Το Modify παίρνει μια λίστα από χαρακτηριστικά και τα διαγράφει, προσθέτει νέες τιμές ή ακόμα και αντικαθιστά τις τωρινές τιμές με νέες. Παραδείγματος χάριν η λειτουργία Modify DN (move/rename entry) παίρνει το νέο RDN (Relative Distinguished Name) κοιτάει ένα flag που λέει εάν θα διαγράψει ή όχι τις τιμές του παλιού RDN και αναλόγως το αντικαθιστά. Ο εξυπηρετητής μπορεί να υποστηρίζει μετονομασία ολόκληρου του υποδένδρου του καταλόγου.

Η Add λειτουργία μπορεί να προσθέσει πρόσθετα χαρακτηριστικά αλλά και τιμές για αυτά τα χαρακτηριστικά.

Η λειτουργία update είναι ατομική: άλλες λειτουργίες θα δουν είτε την νέα εγγραφή είτε την παλιά. Από την άλλη μεριά στον LDAP δεν μπορούν να καθοριστούν εκτελέσεις πολλών λειτουργιών. Π.χ. εάν ανακαλέσουμε μια εγγραφή και μετά την αλλάξουμε με την εντολή modify, ένας άλλος πελάτης μπορεί να έχει αλλάξει (update) την ίδια εγγραφή κατά την διάρκεια αυτή που εκτελέστηκε το modify. Έτσι οι εξυπηρετητές έχουν αναπτύξει κάποιες επεκτάσεις στα λογισμικά τους για να ικανοποιήσουν και αυτή την περίπτωση.

5.3.5 Extended operations

Η διαδικασία Extended Operations είναι μια γενική λειτουργία του LDAP που μπορεί να χρησιμοποιηθεί για να καθορίσει νέες διαδικασίες όπως το Cancel, το Password Modify και το Start TLS.

5.3.6 Abandon

Η διαδικασία Abandon χρησιμοποιείται όταν ο εξυπηρετητής εγκαταλείπει μια διαδικασία που του δίνεται από ένα message ID. Ο εξυπηρετητής δεν χρειάζεται να απαντήσει σε αυτό το μήνυμα και τελικά ούτε η διαδικασία Abandon ούτε η εγκαταλελειμμένη διαδικασία στέλνει μια απάντηση. Αυτό δημιουργεί πρόβλημα και για να καλυφθεί και αυτή η αδυναμία του συστήματος υπάρχει η παρόμοια Extended Operation Cancel η οποία στέλνει απάντηση. Βέβαια δεν υποστηρίζουν όλες οι εκδόσεις της LDAP αυτό.

5.3.7 Unbind

Η λειτουργία Unbind εγκαταλείπει όλες τις διαδικασίες που εκτελούνται και έπειτα κλείνει την σύνδεση με τον εξυπηρετητή. Αυτή δεν στέλνει καμία απάντηση και το όνομα που χρησιμοποιήθηκε εδώ (Unbind) δεν σημαίνει πως είναι η αντίστροφη διαδικασία από το Bind.

Απλά οι πελάτες θα μπορούσαν να κλείσουν την σύνδεση τους με τον εξυπηρετητή, αλλά υπό κανονικές συνθήκες χρησιμοποιούν συνολικά την λειτουργία Unbind, η οποία επιτρέπει στον εξυπηρετητή να κλείσει ομαλά την σύνδεση με τον πελάτη και να απελευθερώσει τους πόρους που χρησιμοποιούσε. Αν δεν γίνει αυτό τότε ο εξυπηρετητής κρατάει τους πόρους αυτούς για ένα εύλογο χρονικό διάστημα που καθορίζεται από τον χρόνο που κάνει ο πελάτης να φανεί πως έχει εγκαταλείψει την σύνδεση. Επίσης είναι σημαντική λειτουργία καθώς με το Unbind στέλνονται οι λειτουργίες cancel σε όποιες διαδικασίες μπορεί να εγκαταλειφθούν και σταματά τον εξυπηρετητή από το να στέλνει απαντήσεις για τις διαδικασίες που δεν μπορεί να διακοπούν.

5.3.8 LDAP Αντιγραφή (replication)

Ένας άλλος παράγοντας που εξασφαλίζει την καλή απόδοση και την αξιοπιστία ενός συστήματος υπηρεσίας καταλόγου είναι η δυνατότητα χρήσης μηχανισμών που βελτιώνουν συνολικά την επίδοση του συστήματος και εξασφαλίζουν την άμεση και αποτελεσματική αντιμετώπιση τυχόν προβλημάτων. Στους διακομιστές LDAP η δυνατότητα αυτή παρέχεται μέσω της δημιουργίας και χρήσης μηχανισμών αντιγραφής (replication). Το κύριο πλεονέκτημα της λειτουργίας αυτής είναι ότι μπορούμε να δημιουργούμε αντίγραφα ενός διακομιστή υπηρεσίας καταλόγου για να εξασφαλίσουμε την ανεκτικότητα σε σφάλματα, το διαμοιρασμό του φόρτου μέσω της κατανομής των λειτουργιών σε διαφορετικά μηχανήματα και τη μεγαλύτερη ασφάλεια των δεδομένων.

Η πιο απλή μορφή αντιγραφής είναι η single-master αντιγραφή όπου ένας κύριος διακομιστής αποτελεί τον προμηθευτή (supplier) που προωθεί τις αλλαγές σε διακομιστές-αντίγραφά του, τους καταναλωτές (consumers). Κάποια από τα προσφερόμενα προϊόντα, υποστηρίζουν multi-master αντιγραφή όπου ένας διακομιστής μπορεί να είναι ταυτόχρονα προμηθευτής και καταναλωτής και κατά συνέπεια το ίδιο υποδένδρο (subtree) υπάρχει και ενημερώνεται από περισσότερους από έναν διακομιστές. Το υποδένδρο αποτελεί ξεχωριστό αντίγραφο ανάγνωσης/γραφής σε καθέναν από τους κύριους διακομιστές. Ο κάθε καταναλωτής διατηρεί το δικό του read-only replica. Οι καταναλωτές δέχονται αλλαγές από τους προμηθευτές και μπορούν να διαθέτουν ορισμένα referrals σχετικά με αυτούς ώστε να ανακατευθύνουν τα αιτήματα που προέρχονται από αυτούς.

Τα σημαντικότερα πλεονεκτήματα που παρουσιάζονται με την υλοποίηση της multi-master αντιγραφής είναι η υποστήριξη ανοχής σε σφάλματα (fault tolerance) που μπορεί να παρέχει ο καθένας από τους κύριους διακομιστές στην περίπτωση που υπάρχει κάποιο πρόβλημα με τον άλλο, καθώς επίσης και το γεγονός ότι οι αλλαγές θα γίνονται στον τοπικό διακομιστή του κατανεμημένου συστήματος.

Η υλοποίηση του μηχανισμού αντιγραφής πραγματοποιείται στα περισσότερα συστήματα με τη χρήση χρονοσφραγίδων. Υποστηρίζονται δύο πρωτόκολλα, αντιγραφής το Change Log πρωτόκολλο και το LCUP (LDAP Client Update Protocol). Με τη χρήση του LCUP οι κύριοι διακομιστές ενημερώνουν τους καταναλωτές, με τη χρήση cookies, χωρίς να μεταφέρουν ολόκληρη τη πληροφορία. Με τη χρήση του Change Log πρωτόκολλου κάθε κύριος διακομιστής διατηρεί το δικό του αρχείο καταγραφής αλλαγών για κάθε αντίγραφο. Το αρχείο αυτό περιγράφει τις αλλαγές που πρέπει να προωθηθούν στους καταναλωτές αλλά και στους υπόλοιπους κύριους διακομιστές στην περίπτωση της multi-master αντιγραφής.

5.4 OpenLDAP

Ο OpenLDAP [1] συνιστά λογισμικό ανοικτού κώδικα Υπηρεσίας Καταλόγου. Η ανάπτυξή του υποστηρίζεται από μία παγκόσμια κοινότητα εθελοντών που επικοινωνούν μεταξύ τους με σκοπό την υλοποίηση του λογισμικού και του συνοδευτικού υλικού. Συνεπώς αποτελεί προϊόν μιας συλλογικής προσπάθειας με στόχο τη δημιουργία ενός πλήρους λειτουργικού προϊόντος λογισμικού. Η χρήση του λογισμικού OpenLDAP είναι ιδιαίτερα διαδεδομένη στα πανεπιστημιακά ιδρύματα και τους ερευνητές, όμως για τις περισσότερες επιχειρήσεις δεν παρουσιάζει μια βιώσιμη off-the-shelf λύση υπηρεσίας καταλόγου.

Το λογισμικό του OpenLDAP περιλαμβάνει τον μοναδικό (stand-alone) διακομιστή LDAP (slapd), τον μοναδικό διακομιστή αντίγραφου (replication server) LDAP (slurpd), LDAP C++ σετ εργαλείων ανάπτυξης λογισμικού (Software Development Kit), τις βιβλιοθήκες (lldap – LDAP βιβλιοθήκη πελάτη, llber - BER/DER βιβλιοθήκη κωδικοποίησης/αποκωδικοποίησης), τα συμπληρωματικά εργαλεία (LDIF εργαλεία για μετατροπή δεδομένων, τα LDAP εργαλεία γραμμής εντολών, SNACC - ASN.1 εργαλεία ανάπτυξης, clients κλπ) και τη σχετική τεκμηρίωση.

Η έκδοση, του OpenLDAP 2.4.21 (19/2/2010) υποστηρίζει τις περισσότερες πλατφόρμες Unix ή UNIX-like (Linux, FreeBSD, NetBSD, OpenBSD, AIX, HP-UX, Solaris κλπ). Προηγούμενες εκδόσεις του λογισμικού διατίθενται για άλλες πλατφόρμες όπως Apple MacOS X, IBM zOS, και Microsoft Windows 2000 (OpenLDAP 2.2.29).

5.5 Προβλήματα στην πραγματοποίηση αλλαγών μέσω LDAP

Στην τρέχουσα κατάσταση που επικρατεί στην ακαδημαϊκή κοινότητα, όλες οι λειτουργίες αλληλεπίδρασης των εφαρμογών/υπηρεσιών με την υπηρεσία καταλόγου πραγματοποιούνται απευθείας με τη χρήση του πρωτοκόλλου LDAP (και τη χρήση κατάλληλης πιστοποίησης και Access List ανά υπηρεσία). Στην περίπτωση των λειτουργιών ανάγνωσης δε δημιουργείται πρόβλημα αλλά στην περίπτωση λειτουργιών εγγραφής (προσθήκη, διαγραφή και ανανέωση εγγραφών) παρουσιάζονται οι παρακάτω δυσκολίες:

- Αλλαγές συμβαίνουν από μία πλειάδα πελατών και όχι από ένα σημείο εισόδου-εξόδου, γεγονός το οποίο μειώνει τις δυνατότητες αυστηρού ελέγχου της διαδικασίας, αναλυτικής καταγραφής των αιτήσεων (logging) και εφαρμογής μέτρων ελέγχου της πρόσβασης (access lists).
- Κάθε υπηρεσία διαθέτει γνώση μόνο του σχήματος δεδομένων που αναφέρεται στην ίδια την υπηρεσία και όχι του συνολικού σχήματος που απαιτείται να διατηρείται σε κάθε εγγραφή. Αυτό γίνεται ιδιαίτερα σαφές στην περίπτωση της λειτουργίας δημιουργίας εγγραφής. Στην πλειοψηφία των περιπτώσεων μία εγγραφή αναφέρεται σε φυσικό πρόσωπο στο οποίο παρέχεται ένας αριθμός από δικτυακές υπηρεσίες. Οι υπηρεσίες αυτές απαιτούν την ύπαρξη συγκεκριμένων attributes στην εγγραφή του χρήστη, attributes τα οποία είναι γνωστά (και χρησιμοποιούνται) μόνο από την εκάστοτε υπηρεσία. Κατά συνέπεια εάν η δημιουργία της εγγραφής χρήστη πραγματοποιείται από μία υπηρεσία η εγγραφή θα λαμβάνει μόνο τα attributes που αντιστοιχούν στην υπηρεσία αυτή και δε θα απολαμβάνει πρόσβαση σε άλλες υπηρεσίες.
- Ένας αριθμός από attributes (πχ username, email address κτλ) απαιτείται να είναι μοναδικός για το σύνολο των εγγραφών της υπηρεσίας καταλόγου. Κάτι τέτοιο δεν είναι ιδιαιτέρως εύκολο να το εξασφαλίσει μία υπηρεσία κατά τη δημιουργία μίας εγγραφής αλλά μόνο η υπηρεσία καταλόγου η ίδια.
- Η δημιουργία (ή αλλαγή) κάποιων attributes απαιτεί την ταυτόχρονη αλλαγή κάποιων άλλων παραγόμενων attributes (πχ η αλλαγή του ονόματος ενός χρήστη απαιτεί την ταυτόχρονη αλλαγή του παραγόμενου ονοματεπώνυμου).
- Αλλαγές σε εγγραφές (κάθε είδους) σε πολλές περιπτώσεις απαιτούν την εκτέλεση εξωτερικών εργασιών ανεξάρτητων από την ίδια την υπηρεσία καταλόγου.

Οι παραπάνω δυσκολίες δεν μπορούν να απλοποιηθούν από την τρέχουσα υλοποίηση της υπηρεσίας καταλόγου. Η υπηρεσία καταλόγου παρέχει μόνο απευθείας LDAP πρόσβαση η οποία δεν δίνει τη δυνατότητα λειτουργιών post-operation ή αυτόματης αλλαγής σε attributes πέραν αυτών για τα οποία αιτήθηκε αλλαγή. Τέτοιες λειτουργίες είναι ευθύνη ενός ενδιάμεσου στρώματος λειτουργιών (middleware) το οποίο αναλαμβάνει να αυξήσει την ‘εξυπνάδα’ στις λειτουργίες εγγραφής και να εκτελέσει τυχόν επιπλέον λειτουργίες που απαιτούνται μετά την πραγματοποίηση κάποιας αλλαγής.

Το middleware αυτό απαιτεί επιπλέον φόρτο τόσο αρχικής ανάπτυξης του όσο και δημιουργίας συγκεκριμένων interfaces για κάθε προβλεπόμενη λειτουργία. Η ανάπτυξη του όμως αυξάνει την ευελιξία της υπηρεσίας, προσφέρει ένα έτοιμο και πλήρες interface για κάθε νέα υπηρεσία που επιθυμεί να συνδεθεί στην υπηρεσία καταλόγου και επιτρέπει την εύκολη

επέκταση των λειτουργιών εγγραφής όποτε απαιτηθεί. Για αυτό το λόγω κατόπιν της ανάπτυξης της υπηρεσίας καταλόγου θα ήταν απαραίτητη η ανάπτυξη των υπηρεσιών δικτύου (web services), που αναλύονται παρακάτω.

5.6 Διαφορές μεταξύ του πρωτοκόλλου LDAP και βάσεων Δεδομένων (RDBMS)

Η τεχνολογία LDAP έρχεται να καλύψει το κενό που αφήνουν οι βάσεις δεδομένων στα επαναλαμβανόμενα μοτίβα διαχείρισης ατόμων, που συναντώνται στις επιχειρήσεις και τους οργανισμούς. Προτυποποιεί δομές τις οποίες οι διαχειριστές συστημάτων ούτως ή άλλως ήταν αναγκασμένοι να εφευρίσκουν και ταυτόχρονα προσφέρει εξειδικευμένες λειτουργίες για την αποδοτική διαχείρισή τους. Ακριβώς αυτά τα χαρακτηριστικά συμβάλλουν στην ταχεία και ευρεία αποδοχή της τεχνολογίας LDAP.

Στο ίδιο πνεύμα, είναι φανερό πως η τεχνολογία LDAP δεν έρχεται να αντικαταστήσει τις βάσεις δεδομένων, αλλά μάλλον να δουλέψει παράλληλα με αυτές σε ένα επιχειρησιακό περιβάλλον: Η αναγκαιότητα ύπαρξης ενός συστήματος βάσης δεδομένων ήταν - και παραμένει- αδιαμφισβήτητη, η έλευση του LDAP απλώς αμφισβητεί το κατά πόσο μία βάση δεδομένων είναι το ιδανικό σύστημα για τη διαχείριση εγγραφών χρηστών.

Το πρωτόκολλο LDAP συνήθως συγκρίνεται με τις βάσεις δεδομένων, γιατί είναι αυτή η υποδομή που καλείται να αντικαταστήσει. Παρακάτω παρουσιάζεται ένας πίνακας με ορισμένα από τα πλεονεκτήματα και μειονεκτήματα χρήσης της τεχνολογίας LDAP:
Υπέρ:

- Παρέχει βελτιστοποιημένες δομές για τη διενέργεια μεγάλου αριθμού αναγνώσεων.
- Παρέχει έναν πολύ ευέλικτο και συμπαγή μηχανισμό εκχώρησης δικαιωμάτων.
- Επιτρέπει την άμεση ομαδοποίηση εγγραφών και την ενιαία αντιμετώπισή τους.
- Βασίζεται σε ένα προτυποποιημένο πρωτόκολλο και για το λόγο αυτό ανεξάρτητο από την πλατφόρμα που χρησιμοποιείται για την προσπέλασή της.
- Το ιεραρχικό μοντέλο ονοματολογίας επιτρέπει τη μονοσήμαντη προσπέλαση των εγγραφών.

Κατά:

- Η πλειοψηφία των οργανισμών διαθέτει ήδη μία καλά οργανωμένη υπηρεσία διαχείρισης χρηστών, χρησιμοποιώντας παραδοσιακές βάσεις δεδομένων και η μετάβαση από το ένα μοντέλο στο άλλο δεν είναι απλή.
- Διαχείριση ενός εξυπηρετητή LDAP προσθέτει επιπλέον βάρος στην συνολική διαχείριση των υπολογιστικών συστημάτων ενός οργανισμού.
- Οι εφαρμογές LDAP δεν είναι κατάλληλες για την αποθήκευση αντικειμένων μεγάλου μεγέθους , ή αντικειμένων που ανανεώνονται συχνά.
- Η χρήση του LDAP δεν έχει νόημα αν δεν αναπτυχθούν επιπλέον υπηρεσίες που θα κάνουν χρήση των δυνατοτήτων του.

Πέρα από τις βασικές δυνατότητες που παρέχει το πρωτόκολλο LDAP, ένα επιχειρησιακό περιβάλλον επωφελείται επίσης από υπηρεσίες που βασίζονται σε αυτό, κυρίως σε επίπεδο ταυτοποίησης και εκχώρησης δικαιωμάτων. Τα τελευταία χρόνια παρατηρείται μία αλματώδης αύξηση στην ανάπτυξη τέτοιων εφαρμογών ενώ ταυτόχρονα όλο και περισσότερες επιχειρήσεις, ιδρύματα και οργανισμοί υιοθετούν την τεχνολογία αυτή.

Το πρωτόκολλο LDAP είναι ένα πρωτόκολλο ερωτήσεων και τροποποιήσεων των περιεχομένων ενός καταλόγου πάνω από το δικτυακό πρωτόκολλο TCP/IP. Ένας κατάλογος είναι ένα σύνολο πληροφοριών με τις παρόμοιες ιδιότητες που οργανώνονται κατά τρόπο λογικό και ιεραρχικό. Το πιο κοινό παράδειγμα καταλόγου είναι ο τηλεφωνικός κατάλογος, που αποτελείται από μια σειρά ονομάτων που οργανώνονται αλφαβητικά, με συνημμένο τον αντίστοιχο τηλεφωνικό αριθμό. Ένας κατάλογος LDAP απεικονίζει συχνά τα διάφορα πολιτικά, γεωγραφικά ή/και οργανωτικά δομικά στοιχεία του οργανισμού που μοντελοποιεί, ανάλογα με το πρότυπο που επιλέγεται.

Η πληθώρα των LDAP υλοποιήσεων τείνουν να χρησιμοποιούν αυτή την στιγμή τα στοιχεία της υπηρεσίας ονοματολογίας διαδικτύου (DNS) για την δόμηση τουλάχιστον των κορυφαίων επιπέδων της ιεραρχίας. Βαθύτερα μέσα στον κατάλογο είναι δυνατόν να εμφανιστούν

καταχωρήσεις που αντιπροσωπεύουν ανθρώπους, μηχανήματα, οργανωτικές μονάδες, τα έγγραφα, ομάδες ανθρώπων ή γενικώς οτιδήποτε είναι δυνατόν να αντιπροσωπεύει μια δεδομένη ιεραρχική μορφή.

Η υπηρεσία LDAP αποτελεί υποστηρικτική υπηρεσία υπό την έννοια ότι σπάνια εμφανίζεται η χρήση της στους τελικούς χρήστες, αλλά συχνά αποτελεί την βάση δεδομένων ενός πλήθους υπηρεσιών στις οποίες έχουν πρόσβαση οι χρήστες όπως υπηρεσίες ηλεκτρονικού ταχυδρομείου, φιλοξενίας ιστοτόπων, συστήματος ονοματολογίας. Ένας λόγος να επιλεχτεί μία ιεραρχική δομή LDAP ως υποστηρικτική υπηρεσία είναι η αρκετά ευρεία υποστήριξη του πρωτοκόλλου ενώ ταυτόχρονα είναι αρκετά γενικό και περιλαμβάνει βασικά χαρακτηριστικά γνωρίσματα όπως η ασφάλεια, και η υποστήριξη πολλών τύπων εφαρμογών.

Κοινές εφαρμογές με χρήση LDAP είναι η αποθήκευση στοιχείων χρηστών/ομάδας υπολογιστών και πληροφοριών επικοινωνίας. Πολλοί πελάτες ηλεκτρονικού ταχυδρομείου υποστηρίζουν αναζητήσεις επαφών μέσω LDAP.

5.7 Τεχνικές διαφορές μεταξύ του LDAP και βάσεων Δεδομένων.

Το πρωτόκολλο LDAP χαρακτηρίζεται ως «write-once-read-many-times» υπηρεσία. Δηλαδή, το είδος των δεδομένων που θα έπρεπε κανονικά να αποθηκεύονται σε μια υπηρεσία LDAP δεν αναμένεται να αλλάξει σε κάθε πρόσβαση. Για επεξήγηση: το LDAP δεν είναι κατάλληλο για την τήρηση αρχείων συναλλαγών στον τραπεζικό τομέα, επειδή, από τη φύση τους, αλλάζουν σε κάθε πρόσβαση (συναλλαγή). Το LDAP θα ήταν, ωστόσο, πολύ κατάλληλο για να τηρεί τις λεπτομέρειες των τραπεζικών υποκαταστημάτων, των ωρών λειτουργίας, των εργαζομένων, κ.λπ.

Read:write ratios

Δεν είναι σαφές κατά τη φράση «write-once-read-many-times» πόσες πολλές φορές είναι δυνατόν να διαβάζουμε από μια βάση LDAP. Επιπρόσθετα δεν γνωρίζουμε που είναι η γραμμή μεταξύ λογικής χρήσης του LDAP έναντι μιας κλασικής συναλλαγής προσανατολισμένης σε σχεσιακή βάση δεδομένων, για παράδειγμα, MySQL, PostgreSQL.
Δεν υπάρχει απλή απάντηση, για τα παραπάνω εύγλωττα ερωτήματα αλλά οι ακόλουθες απαντήσεις μπορούν να φανούν χρήσιμες:
Κατά την εγγραφή σε μια βάση οποιασδήποτε μορφής, για να μετρήσουμε τις επιδόσεις της είναι να λάβουμε υπόψη το πόσο γρήγορα επικαιροποιούνται οι δείκτες (indexes). Όσο περισσότερους δείκτες έχουμε (για ταχύτερη ανάγνωση) τόσο λιγότερο συχνά θα πρέπει να ενημερώνουμε τον κατάλογο. Μια αναλογία διάβασε / γράψε (Read:write ratios) μικρότερη από 1,000:1 είναι απολύτως λογική για ένα απλό κατάλογο LDAP, εάν είναι υψηλότερη προτείνεται για βελτιστοποιημένο κατάλογο LDAP.
Εάν ο όγκος των δεδομένων είναι μεγάλος (ας πούμε> 10.000) ο χρόνος που απαιτείται για την ενημέρωση ακόμη και ενός μικρού αριθμού των δεικτών μπορεί να είναι τόσο σοβαρό και επιβαρυντικό για το σύστημα καταλόγου που καθιστά απαραίτητο να κρατήσουμε τις ενημερώσεις όσο το δυνατόν χαμηλότερα του (10,000:1).
Και αν ο όγκος των δεδομένων είναι σχετικά μικρός (δηλαδή <1.000 εγγραφές), απαιτούνται μικρά ευρετήρια και δεν χρησιμοποιείται αναπαραγωγή της βάσης (replication), δεν θεωρούμε ότι υπάρχει εγγενής λόγος για να μην μπορούμε να χρησιμοποιήσουμε LDAP .

5.8 Ορατότητα δεδομένων

Για την δημιουργία ενός καταλόγου LDAP χρησιμοποιείται ένα μοντέλο δεδομένων που αντλείται από τη φυσική οργάνωση του οργανισμού. Για να καθοριστεί το μοντέλο δεδομένων στον LDAP δεν είναι απαραίτητο να γνωρίζουν την πραγματική δομή των δεδομένων. Αυτό έρχεται σε έντονη αντίθεση με την SQL κατά την οποία τα ερωτήματα έχουν πλήρη και λεπτομερή γνώση των δομών δεδομένων και την οργάνωση σε πίνακες, συνδέσεις, κ.λπ.. Η βασική διαφορά τους είναι στην πολυπλοκότητα σχεδιασμού και ερωτημάτων.

5.9 Συγχρονισμός δεδομένων

Οι σχεσιακές βάσεις δεδομένων μπορούν να έχουν τεράστιο μέγεθος και λειτουργούν έτσι ώστε να εξασφαλιστεί ότι τα δεδομένα είναι συνεπή κατά τη διάρκεια της εγγραφής / ενημέρωσης σε οποιεσδήποτε συναλλαγές, κλειδώματα και άλλες διαδικασίες. Αυτό είναι μια ζωτική και αναγκαία προϋπόθεση. Τα δεδομένα από ένα κύριο LDAP διακομιστή για να μεταφερθούν στους άλλους δευτερεύον (slaves) χρησιμοποιούν μια απλή ασύγχρονη διαδικασία αναπαραγωγής. Αυτό έχει ως αποτέλεσμα την έξοδο των MASTER και SLAVE συστημάτων έξω από τον συγχρονισμό των δεδομένων κατά τη διάρκεια του κύκλου αναπαραγωγής. Ένα ερώτημα προς τον MASTER και προς τον SLAVE κατά τη διάρκεια αυτής (συνήθως μικρής), μπορεί να δώσει μια διαφορετική απάντηση.

6 Υπηρεσίες ιστού (Web Services)

6.1 Επισκόπηση της τεχνολογίας των υπηρεσιών ιστού

Οι υπηρεσίες ιστού (web services) είναι μια ανερχόμενη και πολλά υποσχόμενη τεχνολογία που συνεχώς εξαπλώνεται σε εφαρμοσμένα πληροφοριακά συστήματα. Οι υπηρεσίες ιστού είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Μια υπηρεσία ιστού είναι μια διεπαφή λογισμικού (software interface) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων μηνυμάτων XML. Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα XML για να περιγράψει μία λειτουργία (operation) προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή. Μια ομάδα από υπηρεσίες ιστού οι οποίες αλληλεπιδρούν μεταξύ τους αποτελεί μια εφαρμογή υπηρεσίας ιστού (web services).

Σύμφωνα με τον οργανισμό W3C (World Wide Web Consortium) “η υπηρεσία ιστού αποτελεί ένα σύστημα λογισμικού σχεδιασμένο να υποστηρίζει τη διαλειτουργικότητα μεταξύ των μηχανημάτων πάνω από δίκτυο”. Μία υπηρεσία ιστού συνεπώς αποτελεί σύστημα λογισμικού το οποίο αναγνωρίζεται από ένα URI [IETF RFC 2396], ενώ η διεπαφή καθώς και οι δράσεις της ορίζονται πλήρως και περιγράφονται σε eXtensble Markup Language (XML) μορφή.

Παρά το γεγονός ότι η χρήση των υπηρεσιών ιστού δεν περιορίζεται με συγκεκριμένα πρωτόκολλα, μετά από μία κοινή αποδοχή των μεγαλύτερων εταιρειών λογισμικού στον κόσμο της αρχιτεκτονικής αυτής, έχει πλέον καθοριστεί ένα πιο συγκεκριμένο μοντέλο σύμφωνα με το οποίο θα πρέπει οι εταιρείες να παράγουν και να χρησιμοποιούν τις συγκεκριμένες υπηρεσίες..

«Το SOAP στην έκδοση 1.2 είναι ένα ελαφρύ πρωτόκολλο προορισμένο για την ανταλλαγή δομημένων πληροφοριών σε ένα αποκεντρωμένο, διανεμημένο περιβάλλον. Χρησιμοποιεί τεχνολογίες XML για να καθορίσει ένα επεκτάσιμο πλαίσιο παρέχοντας μια δομή μηνυμάτων η οποία μπορεί να ανταλλαχθεί πάνω από ποικίλα δικτυακά πρωτόκολλα. Το πλαίσιο έχει σχεδιαστεί να είναι ανεξάρτητο από οποιοδήποτε προγραμματιστικό μοντέλο και σημασιολογία υλοποίησης» ***UDDI.org***

Για την επικοινωνία συνεπώς με βάση το μοντέλο αυτό στο επίπεδο εφαρμογής, χρησιμοποιείται συνήθως το πρωτόκολλο HTTP. Πάνω συνήθως από το πρωτόκολλο HTTP χρησιμοποιείται το πρωτόκολλο της W3C Simple Object Access Protocol (SOAP), πρωτόκολλο το οποίο επιτρέπει την ανταλλαγή μηνυμάτων XML (αντικειμένων) πάνω από δίκτυο. Το πρωτόκολλο SOAP μπορεί να λειτουργήσει και πάνω από άλλα πρωτόκολλα (FTP, SMTP κλπ). Με τη χρήση του πρωτοκόλλου αυτού υλοποιείται ο διακομιστής SOAP (αρχιτεκτονική client-server). Για κάθε λειτουργία της υπηρεσίας ιστού, ορίζεται μία λειτουργία στον διακομιστή SOAP. Έπειτα, το σύνολο των λειτουργιών του διακομιστή SOAP καθώς και τα υπόλοιπα χαρακτηριστικά του, περιγράφονται σε ένα αρχείο που ονομάζεται WSDL (Web Service Discription Language) και το οποίο δημοσιεύεται στο Internet έτσι ώστε να δίνεται η

δυνατότητα σε κάθε ενδιαφερόμενο χρήστη να μπορεί άμεσα να χρησιμοποιήσει την υπηρεσία. Οι περισσότερες γλώσσες πλέον προγραμματισμού από την Delphi v7 μέχρι το Visual Studio .net, την PHP, την JSP και άλλες πολλές, υποστηρίζουν τη δημιουργία SOAP διακομιστών με πάρα πολύ απλό τρόπο.

«Η WSDL είναι ένα σχήμα XML για την περιγραφή δικτυακών υπηρεσιών σαν ένα σύνολο από τελικά σημεία που λειτουργούν σε μηνύματα τα οποία περιέχουν πληροφορία είτε προσανατολισμένη στα έγγραφα είτε προσανατολισμένη στις διαδικασίες.» ***W3C***

Η WSDL (Web Service Description Language) είναι μία γλώσσα σε XML μορφή η οποία περιγράφει απόλυτα μία web service. Με πιο απλά λόγια η WSDL μας βοηθάει να περιγράψουμε ένα σύνολο από μηνύματα και το πώς αυτά τα μηνύματα ανταλλάσσονται. Έτσι για κάθε υπηρεσία ιστού που δημιουργείται, αντίστοιχα πρέπει να δημιουργείται ένα αρχείο WSDL στο οποίο θα καταγράφονται όλες οι πληροφορίες για την ίδια την υπηρεσία. Πιο συγκεκριμένα εκεί καταγράφεται η διεύθυνση IP του διακομιστή, ποιες λειτουργίες υποστηρίζει καθώς και πώς δέχεται και πώς επιστρέφει τα δεδομένα για κάθε λειτουργία. Υπάρχουν πάρα πολλά διαθέσιμα εργαλεία που δημιουργούν αυτόματα ένα αρχείο WSDL παράλληλα με τη δημιουργία του SOAP server.

«*Οι υπηρεσίες ιστού έχουν νόημα μόνο όταν δυνητικοί χρήστες μπορούν να βρουν πληροφορίες ικανές ώστε να επιτρέψουν την εκτέλεσή τους.»* ***OASIS***

Τέλος το UDDI (Universal Description, Discovery, and Integration) αποτελεί ένα πρωτόκολλο καταχώρησης για υπηρεσίες ιστού. Χρησιμοποιείται για να μπορούμε να παρέχουμε πληροφορίες για τις υπηρεσίες ιστού. Κάθε καταχώρηση περιέχει το αρχείο WSDL και τη διεύθυνση που λειτουργεί η υπηρεσία στο Internet. Επιπρόσθετα σε κάθε καταχώρηση υπάρχουν και διάφορες άλλες πληροφορίες για την υπηρεσία που σχετίζονται με τον ιδιοκτήτη της, την πολιτική του κλπ. Υπάρχουν διαφορετικοί τύποι καταχωρήσεων μίας υπηρεσίας. Πιο συγκεκριμένα υπάρχουν καταχωρήσεις που μπορούν να γίνουν για υπηρεσίες από όλο τον κόσμο και που απευθύνονται σε όλο τον κόσμο, αλλά και καταχωρίσεις που απευθύνονται μόνο σε εξειδικευμένες επιχειρήσεις προωθώντας έτσι και το B2B μοντέλο συνεργασίας.
Τέλος υπάρχουν και καταχωρήσεις υπηρεσιών για πιο εξειδικευμένες περιπτώσεις.

Πληροφορία	**Λειτουργίες**
White pages: Πληροφορίες όπως το όνομα, η διεύθυνση, το τηλέφωνο και άλλες πληροφορίες επικοινωνίας για μία επιχείρηση.	**Publish**: Πώς ο προμηθευτής ενός web service καταχωρεί των εαυτό του.
Yellow Pages: Πληροφορίες που κατηγοριοποιούν επιχειρήσεις. Βασίζονται σε υπάρχοντα πρότυπα κατηγοριοποίησης (μη ηλεκτρονικά).	**Find**: Πώς μία εφαρμογή βρίσκει ένα συγκεκριμένο web service.
Green Pages: Τεχνικές πληροφορίες για τα web service που παρέχονται από μία επιχείρηση.	**Bind**: Πώς μία εφαρμογή συνδέεται, και αλληλεπιδρά με ένα web service αφού αυτό βρεθεί.

Πίνακας 1 Υπηρεσίες του UDDI

Επίπεδο	**Τεχνολογία**
Ομοιόμορφη μορφή και ανταλλαγή δεδομένων	XML
Πρότυπο κανάλι επικοινωνίας	SOAP
Πρότυπη περιγραφική γλώσσα για την περιγραφή των παρεχόμενων υπηρεσιών	WSDL
Καταχώρηση και εντοπισμός των παρεχόμενων υπηρεσιών	UDDI

Πίνακας 2 Βασικές Τεχνολογίες των υπηρεσιών ιστού

6.1.1 Remote procedure call

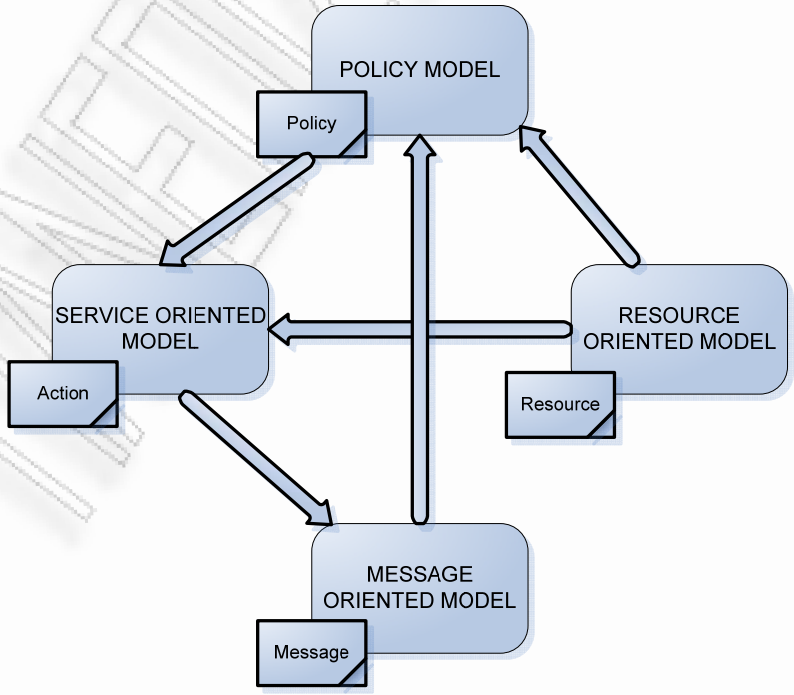
Κλήση απομακρυσμένης διαδικασίας (RPC) είναι μια διαδικασία επικοινωνίας που επιτρέπει σε ένα πρόγραμμα να προκαλέσει μια υπορουτίνα ή διαδικασία για την εκτέλεση σε άλλο χώρο διευθύνσεων (συνήθως σε κάποιον άλλο υπολογιστή σε έναν κοινόχρηστο δίκτυο) χωρίς ο προγραμματιστής να καθορίζει τις λεπτομέρειες για αυτή την απομακρυσμένη αλληλεπίδραση. Δηλαδή, ο προγραμματιστής θα γράψει τον ίδιο κώδικα είτε η υπορουτίνα είναι τοπική, ή εξ αποστάσεως. Όταν το εν λόγω λογισμικό είναι γραμμένο με αντικειμενοστραφείς αρχές (object-oriented), το RPC ενδέχεται να αναφέρεται ως απομακρυσμένης επίκλησης (remote invocation).

6.1.2 Message passing

Το RPC είναι ένα προφανές και δημοφιλές μοντέλο για την εφαρμογή του μοντέλου πελάτη-διακομιστή. Ένα RPC ξεκινά από τον πελάτη στέλνοντας ένα αίτημα για ένα γνωστό απομακρυσμένο server για να εκτελέσει μια συγκεκριμένη διαδικασία παρέχοντάς του συγκεκριμένες παραμέτρους. Έπειτα η απάντηση επιστρέφεται στον πελάτη, όπου συνεχίζει να εκτελείται μαζί με τη διαδικασία. Ενώ ο διακομιστής επεξεργάζεται την κλήση, ο πελάτης σταματάει την εκτέλεση προγραμμάτων (περιμένει μέχρι ο διακομιστής να ολοκληρώσει την επεξεργασία πριν αρχίσει και πάλι την εκτέλεση).

Μια σημαντική διαφορά μεταξύ των κλήσεων απομακρυσμένης διαδικασίας και των τοπικών κλήσεων είναι ότι οι απομακρυσμένες κλήσεις μπορεί να αποτύχουν εξαιτίας των απρόβλεπτων προβλημάτων δικτύου. Επίσης, οι καλούντες σε γενικές γραμμές πρέπει να ανταπεξέρχονται σε τέτοιου είδους βλάβες, ότι δηλαδή η απομακρυσμένη διαδικασία έγινε στην πραγματικότητα. Όταν οι διαδικασίες αυτές είναι γραμμένες σε κώδικα που δίνει την δυνατότητα σε μια ρουτίνα να τις καλεί περισσότερες από μια φορές και να μην αλλάζουν το αποτέλεσμα, μπορεί εύκολα να διαχειριστούν, αλλά δημιουργούνται αρκετές δυσκολίες όταν είναι γραμμένες οι διαδικασίες για συστήματα χαμηλού επιπέδου, όπου τότε οποιαδήποτε αλλαγή επιφέρει προβλήματα.

6.2 Αρχιτεκτονικά μοντέλα των υπηρεσιών ιστού

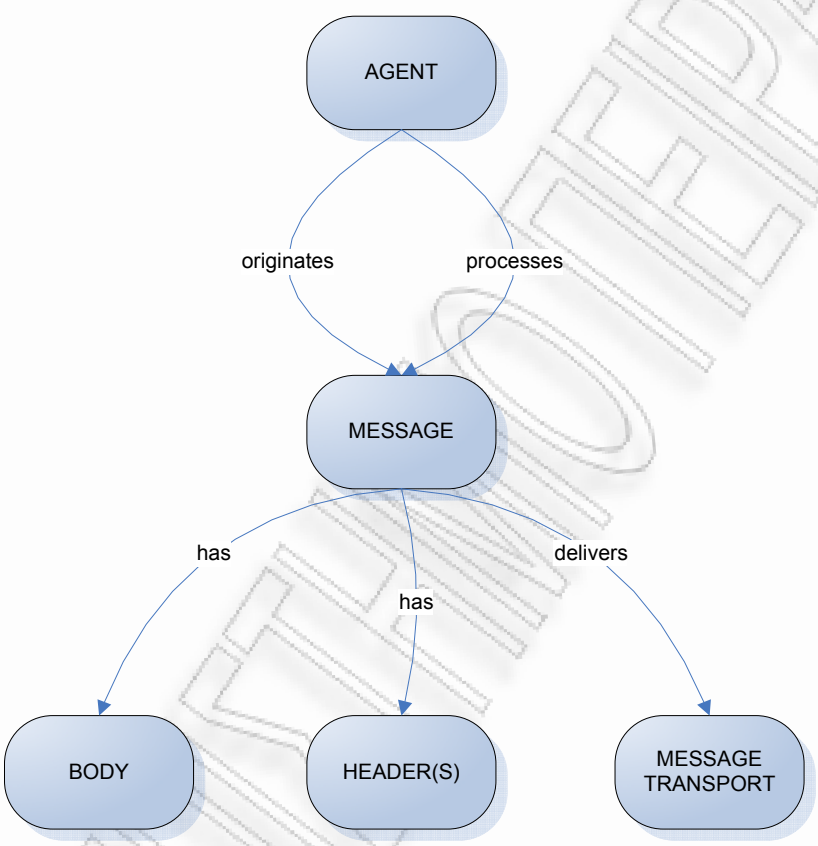


Σχήμα 1. Σύνδεση των μοντέλων των υπηρεσιών ιστού

Στο παραπάνω διάγραμμα φαίνεται η ενδεχόμενη σύνδεση των αρχιτεκτονικών μοντέλων των υπηρεσιών ιστού. Θα εξηγήσουμε παρακάτω την επικοινωνία των τεσσάρων μοντέλων μεταξύ τους καθώς αναλύουμε το κάθε ένα μοντέλο ξεχωριστά.

Τα τέσσερα μοντέλα είναι τα εξής:

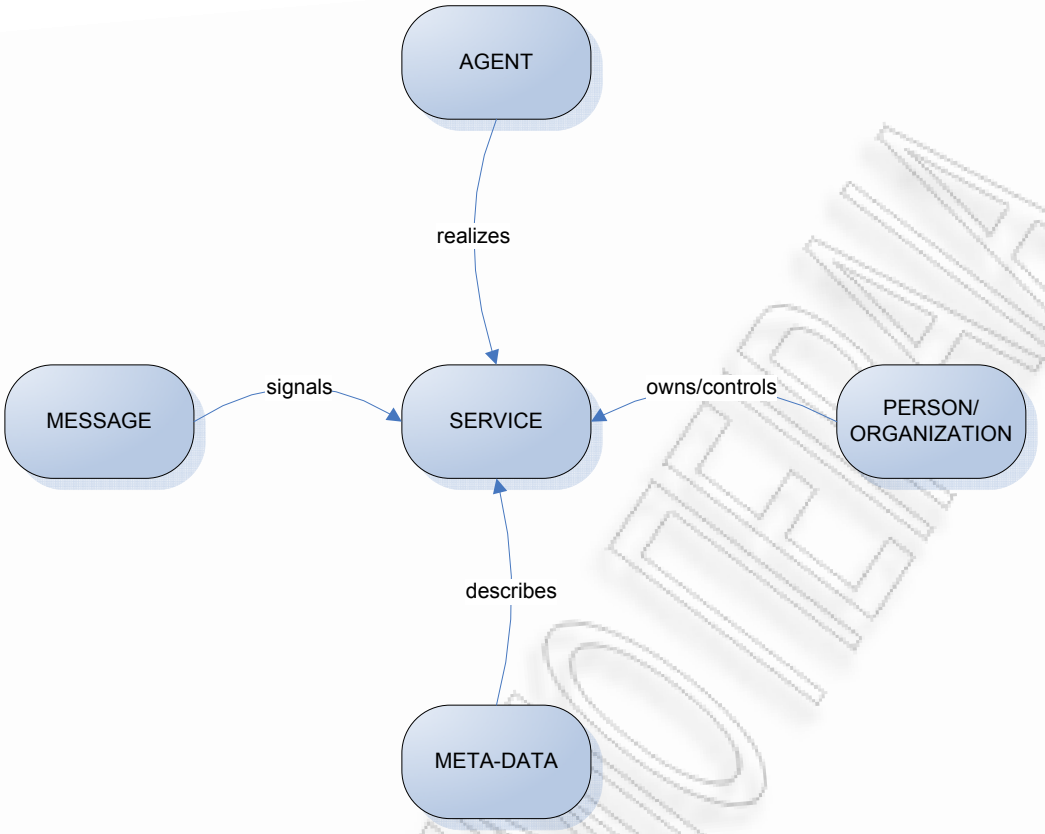
1. Το Message Oriented μοντέλο επικεντρώνεται στα μηνύματα, στην δομή του μηνύματος, στην μεταφορά του μηνύματος και δίχως να δίνει ιδιαίτερη αναφορά ως προς τους λόγους δημιουργίας των μηνυμάτων, ούτε για την σημασία τους.



Σχήμα 2. Απλοποιημένη μορφή μοντέλου Μηνύματος

Η ουσία του παραπάνω μοντέλου βρίσκεται γύρω από μερικές βασικές έννοιες: η πράκτορας (agent) που στέλνει και λαμβάνει τα μηνύματα, η δομή του μηνύματος όσο αναφορά τις κεφαλίδες των μηνυμάτων (message headers) και το κυρίως σώμα (body) καθώς και τους μηχανισμούς που χρησιμοποιούνται για την παράδοση των μηνυμάτων. Φυσικά, υπάρχουν πρόσθετα στοιχεία για να ληφθούν υπόψη, όπως ο ρόλος των πολιτικών και το πώς αυτές διέπουν το πρότυπο μήνυμα.

2. Το Service Oriented μοντέλο επικεντρώνεται στις πτυχές της υπηρεσίας και της δράσης. Είναι σαφές ότι, σε κάθε κατανεμημένο σύστημα, οι υπηρεσίες δεν μπορούν να υλοποιηθούν επαρκώς χωρίς την ύπαρξη των μηνυμάτων, ενώ το αντίστροφο δεν συμβαίνει: Τα μηνύματα δεν χρειάζονται να αφορούν τις υπηρεσίες.



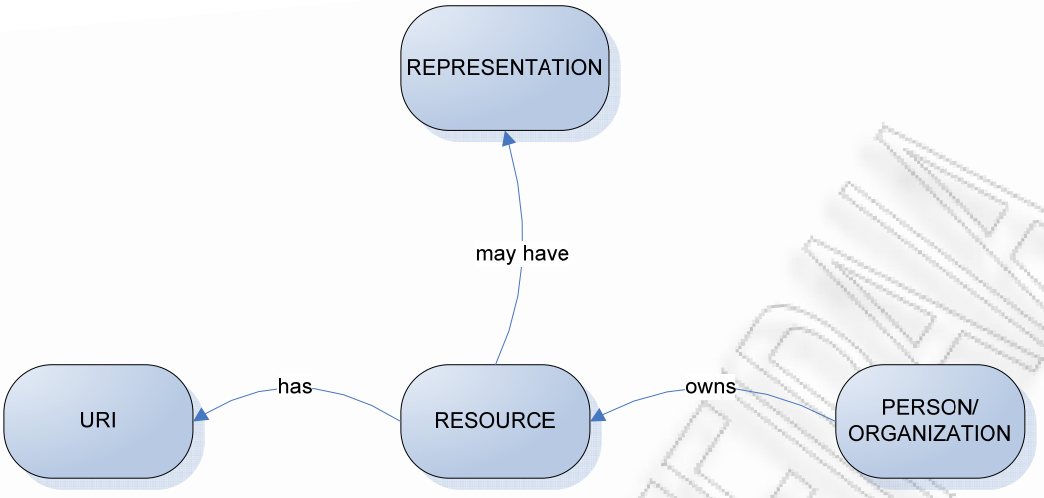
Σχήμα 3. Απλοποιημένο Service Oriented Μοντέλο

Το Service Oriented μοντέλο είναι το πιο περίπλοκο από όλα τα μοντέλα της αρχιτεκτονικής. Ωστόσο, είναι στηριγμένο σε πολύ λίγες βασικές ιδέες. Μια υπηρεσία πραγματοποιείται από έναν πράκτορα (agent) και χρησιμοποιείται από άλλον πράκτορα. Οι Υπηρεσίες αυτές επάγονται μέσω των μηνυμάτων που ανταλλάσσονται μεταξύ των πρακτόρων που τις παρέχουν και των πρακτόρων που τις αιτούνται.

Μια πολύ σημαντική πτυχή των υπηρεσιών είναι η σχέση τους με τον πραγματικό κόσμο: οι υπηρεσίες έχουν αναπτυχθεί κυρίως για να προσφέρουν λειτουργικότητα στον πραγματικό κόσμο. Αυτό συνεπάγεται πως κάποιος όταν δημιουργεί μια υπηρεσία έχει ένα σκοπό να εξυπηρετήσει. Ο ιδιοκτήτης της υπηρεσίας αυτής μπορεί να είναι ένα πρόσωπο ή μια οργάνωση, η οποία έχει μια πραγματική παγκόσμια ευθύνη για την υπηρεσία.

Το Service Oriented μοντέλο κάνει χρήση των μετα-δεδομένων και αποτελεί βασική ιδιοκτησία της Service Oriented Αρχιτεκτονικής. Αυτά τα μετα-δεδομένα χρησιμοποιούνται για να καταγράψουν πολλές πτυχές των υπηρεσιών: από τα στοιχεία της διεπαφής και τον τρόπο μετάδοσής τους, μέχρι την σημασιολογία της υπηρεσίας και ποια πολιτική ασφαλείας έχει δοθεί στην υπηρεσία από τον ιδιοκτήτη της. Για αυτόν τον λόγο η παροχή μιας πλούσιας περιγραφής είναι το κλειδί για την επιτυχή ανάπτυξη και χρήση των υπηρεσιών μέσω του Internet.

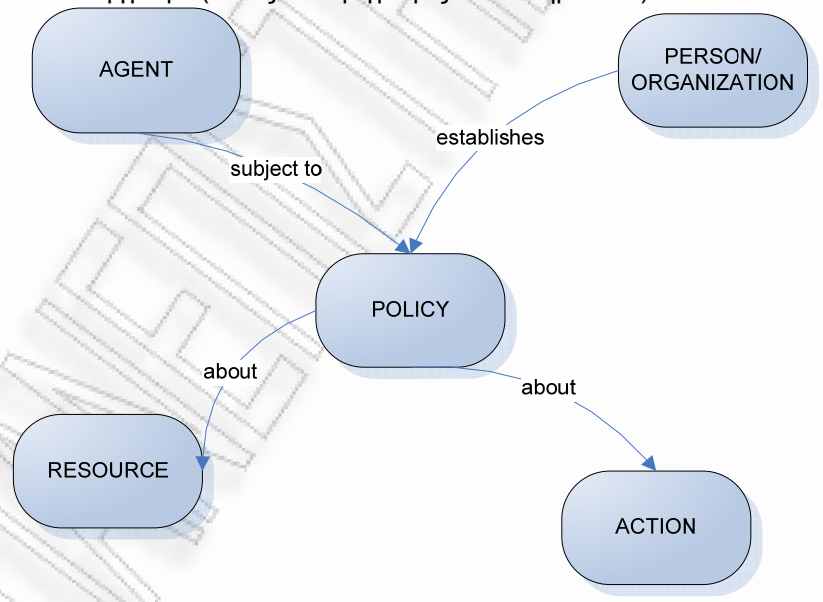
3. Το Resource Oriented μοντέλο επικεντρώνεται στους πόρους που υπάρχουν και έχουν ιδιοκτήτες.



Σχήμα 4. Απλοποιημένη μορφή Resource Oriented Μοντέλου

Το μοντέλο των πόρων πηγάζει από την ίδια ιδέα της Web Αρχιτεκτονικής των πόρων. Επεκτείνοντας αυτό μπορούμε να συμπεριλάβουμε τις σχέσεις μεταξύ των πόρων και των ιδιοκτητών. Εξηγώντας θα έπρεπε να προστεθεί πως ο Uniform Resource Identifier στην πληροφορική (URI) αποτελείται από μια σειρά χαρακτήρων που χρησιμοποιούνται για τον εντοπισμό ενός πόρου στο Διαδίκτυο. Η ταυτοποίηση αυτή επιτρέπει την αλληλεπίδραση με τυχόν αναπαραστάσεις (representation) που μπορεί να έχουν οι πόροι μέσω δικτύου (συνήθως στο World Wide Web) χρησιμοποιώντας ειδικά πρωτόκολλα.

4. Τέλος το μοντέλο πολιτικής (Policy Model) επικεντρώνεται σε προβλήματα σχετικά με τη συμπεριφορά των πρακτόρων και των υπηρεσιών. Μπορούμε να γενικεύσουμε αυτό στους πόρους (όπως πριν) εφόσον οι πολιτικές μπορούν να εφαρμόζονται εξίσου στα έγγραφα (όπως οι περιγραφές των υπηρεσιών).



Σχήμα 5. Απλοποιημένο Μοντέλο Policy

Οι πολιτικές δημιουργούνται για τους πόρους. Οι πολιτικές αυτές εφαρμόζονται στους πράκτορες που μπορεί να επιχειρήσουν την πρόσβαση στους πόρους αυτούς, και έχουν τεθεί σε εφαρμογή, ή είναι εγκατεστημένοι, από ανθρώπους που έχουν την ευθύνη των πόρων. Πολιτικές μπορούν να ληφθούν για να απαλείψουν τυχόν ανησυχίες για την ασφάλεια, για την ποιότητα των υπηρεσιών, για την διαχείριση τους και για ανησυχίες κατά την εφαρμογή τους.

6.3 Πλεονεκτήματα της αρχιτεκτονικής των υπηρεσιών ιστού

Η αρχιτεκτονική των υπηρεσιών ιστού παρέχει αρκετά πλεονεκτήματα όπως

- Διαλειτουργικότητα: Μία υπηρεσία ιστού παρέχει ανεξαρτησία τόσο από λειτουργικό σύστημα όσο και από το υλικό που χρησιμοποιείται. Οποιοδήποτε πρόγραμμα που υποστηρίζει αυτή τη τεχνολογία μπορεί πολύ εύκολα να προσπελάσει μία τέτοια υπηρεσία.
- Ενσωμάτωση: Η δημιουργία μίας υπηρεσία ιστού δεν απαιτεί αλλαγές στον μηχανισμό του λογισμικού συστήματος από το οποίο χρησιμοποιείται.
- Διαθεσιμότητα και δημοσίευση: Οι πληροφορίες για τις υπηρεσίες ιστού δημοσιεύονται, οπότε η εύρεση και η χρήση τους μπορεί να είναι ταχύτατες.
- Επεκτασιμότητα: Η λειτουργία μίας υπηρεσίας ιστού είναι δυνατό να ανανεωθεί με εύκολο τρόπο παρέχοντας έτσι επιπρόσθετες υπηρεσίες στους χρήστες της.
- Χαμηλό κόστος δημιουργίας και χρήσης: Το κόστος ανάπτυξης – επέκτασης σε ένα υπάρχον λογισμικό σύστημα εφαρμογής σε περιβάλλον web για την δημιουργία υπηρεσίας ιστού της εφαρμογής κοστίζει ελάχιστα. Το κόστος ενσωμάτωσης μίας υπηρεσίας ιστού σε κάποιο δικτυακό τόπο ή σε εφαρμογή είναι πάρα πολύ μικρό.
- Χρήση λογισμικών συστημάτων: Όλα τα λογισμικά συστήματα και ειδικότερα οι ιστοσελίδες που χρησιμοποιούν έτοιμες υπηρεσίες γίνονται πιο λειτουργικά και πιο φιλικά αφού παρέχουν περισσότερες υπηρεσίες και σε εφαρμογές εκτός από χρήστες.

6.3.1 Πλεονεκτήματα της χρήσης υπηρεσιών ιστού

Το middleware αυτό μπορεί να υλοποιηθεί με τη χρήση της τεχνολογίας των υπηρεσιών ιστού. Πιο συγκεκριμένα, για κάθε εξωτερική υπηρεσία που απαιτείται να πραγματοποιήσει αλλαγές στην υπηρεσία καταλόγου δημιουργείται κατάλληλο προγραμματιστικό interface μέσω υπηρεσιών ιστού. Το interface αυτό λαμβάνει περιγραφικό όνομα και ορίζονται συγκεκριμένες λειτουργίες τις οποίες χρησιμοποιεί η εξωτερική υπηρεσία για την πραγματοποίηση των αλλαγών. Οι λειτουργίες αυτές αναφέρονται σε υψηλού επιπέδου εργασίες (πχ Δημιουργία Γονέα, Ενεργοποίηση Υπηρεσίας) και όχι στις τελικές αλλαγές επί των δεδομένων της υπηρεσίας καταλόγου. Κάθε λειτουργία λαμβάνει συγκεκριμένες εισόδους (που εξαρτώνται από τις ανάγκες της λειτουργίας) και επιστρέφει το αποτέλεσμα της εργασίας (επιτυχία ή αποτυχία), τυχόν μήνυμα λάθους καθώς και άλλα στοιχεία που απαιτούνται ανά περίπτωση. Κατ' αυτόν τον τρόπο:

- Διατηρείται η ήδη υπάρχουσα λογική ανάπτυξης εφαρμογών που προβλέπει τη δημιουργία συγκεκριμένων και σε αυστηρά ορισμένο πλαίσιο συναρτήσεων – λειτουργιών οι οποίες καλούνται για την υλοποίηση εργασιών (με συγκεκριμένες παραμέτρους) και οι οποίες αναλαμβάνουν την εκτέλεσή τους και την επιστροφή των αποτελεσμάτων.
- Χρησιμοποιείται μία τεχνολογία (web service) που είναι ανεξάρτητη της προγραμματιστικής πλατφόρμας με συνέπεια να παρέχει τη δυνατότητα συνεργασίας μεταξύ ετερογενών συστημάτων (πχ .Net με Java), ενώ τα πρωτόκολλα επικοινωνίας (HTTP, XML, WSDL, SOAP) είναι ιδιαίτερα διαδεδομένα και πολύ καλά ορισμένα με συνέπεια να αποφεύγονται τυχόν περιπτώσεις ασυμβατότητας.
- Όλες οι αλλαγές στα δεδομένα της υπηρεσίας καταλόγου πραγματοποιούνται κεντρικά διαμέσου των υπηρεσιών ιστού και όχι με απευθείας πρόσβαση από τις εξωτερικές υπηρεσίες. Αυτό έχει ως συνέπεια να μπορεί να οριστεί πολύ καλύτερα η πολιτική ασφάλειας επί των δεδομένων και να υπάρχει πλήρης έλεγχος επί των λειτουργιών που πραγματοποιούνται. Αντί οι εξωτερικές υπηρεσίες να έχουν αυξημένα δικαιώματα αλλαγών στα δεδομένα, απλά καλούν συγκεκριμένες συναρτήσεις. Έτσι, οι αλλαγές που μπορεί να πραγματοποιηθούν είναι πολύ συγκεκριμένες, γίνονται όλες από το ίδιο σημείο και μπορεί εύκολα να παρακολουθηθούν και να αποσφαλματοποιηθούν.
- Η κεντρική πραγματοποίηση των λειτουργιών επιτρέπει τη διατήρηση ενός κεντρικού ιστορικού αλλαγών. Παράλληλα, ο συγγραφέας/διαχειριστής των υπηρεσιών ιστού έχει τη δυνατότητα να επιλέξει το επίπεδο λεπτομέρειας εργασιών που θα διατηρούνται στο ιστορικό. κάτι που δεν είναι δυνατό με την υπηρεσία καταλόγου

όπου απλά διατηρούνται γενικές πληροφορίες για τις ενέργειες που εκτελούνται (αναλυτικό ιστορικό δεν είναι δυνατό να διατηρείται λόγω του πολύ μεγάλου πλήθους λειτουργιών, ιδιαίτερα αναζητήσεων, που πραγματοποιούνται σε μία υπηρεσία καταλόγου).

- Είναι δυνατόν να οριστούν εξωτερικές λειτουργίες που εκτελούνται πριν ή μετά την κλήση/ολοκλήρωση των υπηρεσιών ιστού. Έτσι για παράδειγμα η μετακίνηση ενός χρήστη στο δέντρο πληροφοριών, μπορεί να οδηγεί στην εκτέλεση εξωτερικής λειτουργίας που μετακινεί το mailbox του σε άλλο εξυπηρετητή email.
- Δεν απαιτείται η παροχή όλων των τυχόν attributes που μπορεί να περιέχει μία εγγραφή αλλά μόνο των απολύτως απαραίτητων για την πραγματοποίηση της λειτουργίας. Τα υπόλοιπα attributes μπορεί να είναι παραγόμενα με βάση κατάλληλους κανόνες και συναρτήσεις. Έτσι για παράδειγμα μπορεί από το username του χρήστη να προκύπτει το email του (με βάση τον τύπο email=<username>@<domain>) ενώ τα όρια χρήσης της dialup σύνδεσης του (ημερήσιο και εβδομαδιαίο όριο) να είναι στατικά ορισμένα και να λαμβάνουν συγκεκριμένες και προκαθορισμένες τιμές τις οποίες δεν μπορεί να επηρεάσει ούτε ο χρήστης, ούτε η εξωτερική υπηρεσία που καλεί το web service.
- Ο διαχειριστής των υπηρεσιών ιστού μπορεί να ορίσει συγκεκριμένες και αρκετά περίπλοκες πολιτικές επί των τιμών των παρερχομένων attributes, κάτι που δεν είναι δυνατό από την υπηρεσία καταλόγου. Έτσι για παράδειγμα μπορεί να ορίσει συγκεκριμένη πολιτική επί του επιλεγόμενου username/password ή να ορίσει απαγορευμένες τιμές για άλλα attributes.
- Οι εξωτερικές υπηρεσίες δεν απαιτείται να γνωρίζουν απολύτως τίποτα για το σχήμα δεδομένων που χρησιμοποιείται από την υπηρεσία καταλόγου. Απλά παρέχουν τιμές σε συγκεκριμένα και αυστηρά ορισμένα ορίσματα των λειτουργιών που έχουν οριστεί χωρίς να έχουν γνώση πώς αυτά αντιστοιχίζονται σε attributes στις εγγραφές της υπηρεσίας καταλόγου. Ο διαχειριστής των δεδομένων, μπορεί να κάνει οποιαδήποτε μετατροπή, προσθήκη και αφαίρεση στο σχήμα δεδομένων χωρίς να απαιτείται να ενημερώσει τις εξωτερικές υπηρεσίες παρά μόνο να κάνει τις κατάλληλες αλλαγές στον κώδικα των υπηρεσιών ιστού (αν απαιτείται).
- Η λειτουργία διαγραφής χρήστη μπορεί να υλοποιηθεί με την μορφή λήξης εγγραφής και όχι διαγραφής της. Έτσι για παράδειγμα κάποιος χρήστης του οποίου η εγγραφή έχει λήξει μπορεί να ενημερωθεί μέσω email (χρησιμοποιώντας εξωτερικό port-operation όπως περιγράφηκε προηγουμένως) ώστε να μπορέσει να αλλάξει email διεύθυνση σε εύλογο χρονικό διάστημα. Κατ' αυτόν τον τρόπο δίνεται η δυνατότητα για τον ορισμό συγκεκριμένης πολιτικής λήξης και διαγραφής εγγραφών αντί για την απλή διαγραφή τους όταν αυτό ζητείται από μια εξωτερική υπηρεσία. Παράλληλα, η εγγραφή ενός χρήστη στην υπηρεσία καταλόγου μπορεί να αντιστοιχεί σε πολλαπλές εγγραφές σε εξωτερικές υπηρεσίες στην οποία περίπτωση η λειτουργία διαγραφής θα πρέπει απλά να διαγράφει τα attributes που αντιστοιχούν στην κάθε εξωτερική υπηρεσία κάτι που μπορεί να υλοποιηθεί πολύ εύκολα μέσω των υπηρεσιών ιστού.

7 Ανάπτυξη συστήματος

7.1 Βασικές απαιτήσεις προτεινόμενου συστήματος

Η υπό ανάπτυξη υπηρεσία καταλόγου θα πρέπει να χαρακτηρίζεται από τα ακόλουθα στοιχεία:

- Πλήρη υποστήριξη του πρωτοκόλλου της υπηρεσίας (LDAPv3)
- Επεκτασιμότητα σε επίπεδο σχήματος δεδομένων και λειτουργικότητας
- Ασφάλεια και εμπιστευτικότητα δεδομένων
- Ανοικτό κώδικα για την ικανοποίηση βασικών απαιτήσεων του έργου αλλά και για την εύκολη επέκταση της υπηρεσίας και επίλυση τυχόν προβλημάτων στο λογισμικό
- Δομή υψηλής διαθεσιμότητας

- Δυνατότητα συγχρονισμού με εξωτερικά συστήματα
- Αυξημένες επιδόσεις οι οποίες να έχουν όριο μόνο το υλικό της υπηρεσίας και όχι εγγενείς περιορισμούς. Η απόδοση της υπηρεσίας θα πρέπει να μπορεί να αυξηθεί γραμμικά με βάση το διαθέσιμο υλικό καθώς και να υποστηρίζει το διαχωρισμό της υπηρεσίας σε πολλαπλούς χαμηλού κόστους εξυπηρετητές αντί για μία υψηλού κόστους μηχανή.
- Χρήση πολλαπλών εξυπηρετητών με συγχρονισμό δεδομένων μεταξύ τους. Τυχόν απώλεια ενός εξυπηρετητή δε θα πρέπει να επηρεάζει τη συνολική υπηρεσία πέραν της απώλειας του αντίστοιχου ποσοστού συνολικής ισχύος (no single point of failure)
- Δυνατότητα απομακρυσμένης διαχείρισης της υπηρεσίας

Σε επίπεδο λογισμικού υπηρεσίας καταλόγου οι παραπάνω απαιτήσεις μεταφράζονται ως εξής:

- Ανοικτός κώδικας λογισμικού
- Παροχή plugin interface για την επέκταση της λειτουργικότητας της υπηρεσίας καταλόγου
- Παροχή λειτουργίας συγχρονισμού δεδομένων (replication)
- Πολλαπλούς write master δεδομένων (multi master replication)
- Παροχή κονσόλας διαχείρισης της υπηρεσίας.
- Ενσωμάτωση των στοιχείων διαμόρφωσης της υπηρεσίας εντός του δέντρου δεδομένων ώστε να είναι δυνατή η απομακρυσμένη διαχείριση της υπηρεσίας με απευθείας χρήση του πρωτοκόλλου ldap
- Υποστήριξη για πρωτόκολλο ασφαλούς μετάδοσης δεδομένων SSL/TLS.
- Δυνατότητα κρυπτογράφησης συγκεκριμένων attributes
- Δυνατότητα λήψης αντιγράφων ασφαλείας της βάσης δεδομένων της υπηρεσίας και επαναφοράς της υπηρεσίας από αντίγραφο ασφαλείας (backup, restore)
- Όλες οι λειτουργίες διαχείρισης θα πρέπει να μπορούν να γίνουν και να ολοκληρωθούν εν λειτουργία (online) χωρίς να απαιτείται η απενεργοποίηση της υπηρεσίας (import/export δεδομένων, backup/restore δεδομένων)
- Ορισμός των στοιχείων πρόσβασης (Access Lists) εντός του δέντρου δεδομένων ώστε η αλλαγή τους να μπορεί να γίνει απομακρυσμένα και να μην απαιτεί επανεκκίνηση της υπηρεσίας
- Υποστήριξη περιορισμών χρήσης πόρων (resource limits) για την αποφυγή κακής χρήσης της υπηρεσίας (Denial of Service attacks). Είναι επιθυμητό οι περιορισμοί αυτοί να μπορούν να τεθούν ανά συνδεόμενο χρήστη.
- Υποστήριξη της γλώσσας (**Directory Service Markup Language)** DSML για την πρόσβαση στην υπηρεσία καταλόγου από εξωτερικά υποσυστήματα που δεν υποστηρίζουν LDAP αλλά μόνο πρόσβαση μέσω υπηρεσιών ιστού.
- Δυνατότητα ορισμού πολιτικής επί των passwords των χρηστών (password policy) καθώς και απενεργοποίησης της πρόσβασης σε μία εγγραφή χωρίς να απαιτείται η πλήρη διαγραφή της (account deactivation).
- Πλήρες logging της υπηρεσίας.

7.2 Διαθέσιμα λογισμικά

Παρακάτω παρουσιάζονται εμπορικά και ελεύθερα πακέτα λογισμικού που μπορεί να χρησιμοποιηθούν για την υλοποίηση της υπηρεσίας καταλόγου. Συνοπτικά αυτά είναι τα:

- OpenLDAP Directory Server

- Fedora Directory Server
- Red Hat Directory Server
- Sun Java System Directory Server Enterprise Edition
- Windows Active Directory
- IBM Tivoli Directory Server
- Novell eDirectory

Τα χαρακτηριστικά των εν λόγω λογισμικών παρουσιάζονται συνοπτικά παρακάτω.

Λειτουργία σε περιβάλλον UNIX: Ολα τα παρουσιαζόμενα λογισμικά με την εξαίρεση του Windows Active Directory, ήτοι τα OpenLDAP, Fedora Directory Server, Red Hat Directory Server, Sun Java System Directory Server Enterprise Edition, IBM Tivoli Directory Server, Novell eDirectory, είναι δυνατόν να λειτουργήσουν σε περιβάλλον UNIX.

Λειτουργία σε περιβάλλον Windows: Από τα παρουσιαζόμενα λογισμικά δεν είναι δυνατόν να λειτουργήσουν σε περιβάλλον Windows τα Fedora Directory Server και Red Hat Directory Server.

Ενσωματωμένος μηχανισμός ελέγχου πρόσβασης (ACLs)*:* Ο ενσωματωμένος μηχανισμός ελέγχου πρόσβασης επιτρέπει τον έλεγχο πρόσβασης βάσει χαρακτηριστικών που ανήκουν σε αντικείμενα ενσωματωμένα μέσα στην βάση καταλόγου. Από όλα τα παρουσιαζόμενα λογισμικά όλα υποστηρίζουν τον μηχανισμό αυτό.

Δυνατότητα επέκτασης του σχήματος: Όλα τα υπό παρουσίαση λογισμικά επιτρέπουν εύκολη επέκταση του σχήματος με προσθήκη χαρακτηριστικών γνωρισμάτων και κλάσεων στα υπάρχοντα αντικείμενα απαραίτητη προϋπόθεση για την ομαλή κλιμάκωση και εξέλιξη της υπηρεσίας.

Δυνατότητα απομακρυσμένης διαχείρισης μέσω κονσόλας και/ή LDAP: Η δυνατότητα απομακρυσμένης διαχείρισης είτε μέσω εξειδικευμένου λογισμικού ή/και λειτουργικών LDAP θεωρείται απαραίτητη λόγω της κατακόρυφης μείωσης του διαχειριστικού κόστους. Και τα επτά (7) λογισμικά παρουσιάζουν την εν λόγω δυνατότητα παρέχοντας δυνατότητα αποθήκευσης των ρυθμίσεων στην ίδια την υπηρεσία καταλόγου.

Υποστήριξη multi-master replication: Για την μελλοντική επέκταση της υπηρεσίας καταλόγου θεωρείται σημαντική η δυνατότητα για multi-master replication που θα επιτρέπει σε περισσότερους του ενός κεντρικούς εξυπηρετητές να λειτουργούν σε κατάσταση master χωρίς απώλεια δεδομένων ή/και συγχρονισμού. Το λογισμικό OpenLDAP ακολουθεί διαφορετική θεώρηση στον τομέα του replication με την υποστήριξη ενός νέου πρωτοκόλλου (SyncRepl), το mirrormode host standby master (στη διαμόρφωση αυτή ένας εξυπηρετητής είναι πάντα master και ένας άλλος λειτουργεί ως hot standby σε mirror mode έτοιμος να αναλάβει όλες τις λειτουργίες σε περίπτωση δυσλειτουργίας του master). Η υπό ανάπτυξη έκδοση 2.4 πρόκειται να παρέχει υποστήριξη για multimaster replication.

Υποστήριξη μεγάλου αριθμού εγγραφών: Η ανάγκη για υψηλής απόδοσης υποστήριξη μεγάλου αριθμού εγγραφών καθορίζουν την επιλογή του κατάλληλου λογισμικού. Τα υπό μελέτη λογισμικά παρουσιάζουν δυνατότητες υποστήριξης αριθμού εγγραφών μεγαλύτερου του ενός εκατομμυρίου (1.000.000). Με βάση ανεξάρτητες δοκιμές επιδόσεων τα λογισμικά OpenLDAP και Sun ONE φαίνεται να προσφέρουν τις μεγαλύτερες επιδόσεις από πλευράς αριθμού ταυτόχρονων λειτουργιών και υποστήριξης μεγάλου αριθμού εγγραφών.

Συγχρονισμός με Active Directory: Δεδομένης της πιθανής ανάγκης στο μέλλον για διαλειτουργικότητα με την υπηρεσία Active Directory της εταιρείας Microsoft θεωρείται επιθυμητή η δυνατότητα συγχρονισμού με την εν λόγω υπηρεσία. Πέραν του λογισμικού Active Directory που προσφέρει την δυνατότητα εγγενώς και του λογισμικού OpenLDAP που δεν προσφέρει εγγενώς την εν λόγω δυνατότητα, όλα τα υπό μελέτη λογισμικά έχουν την δυνατότητα να συγχρονιστούν με Active Directory.

Γραφικό περιβάλλον διαχείρισης: Τέλος, πλεονέκτημα θεωρείται η ύπαρξη γραφικού περιβάλλοντος διαχείρισης. Από τα υπό μελέτη συστήματα ο OpenLDAP δεν προσφέρει εγγενώς τη συγκεκριμένη δυνατότητα. Παρόλα αυτά καθώς το σύνολο της διαμόρφωσης είναι αποθηκευμένο σε LDAP η διαχείριση του συστήματος είναι δυνατόν να γίνει απευθείας με τη χρήση κάποιου από τα διαθέσιμα λογισμικά τύπου LDAP browser.

Παρακάτω παρουσιάζονται τα εργαλεία ελεύθερου λογισμικού / λογισμικού ανοικτού κώδικα ανάπτυξης και προσπέλασης υπηρεσιών καταλόγου:

Α) LDAPExplorerTool **http://ldaptool.sourceforge.net/ - 7/5/2010**
Β) Apache Directory Studio **http://directory.apache.org/studio/ - 7/5/2010**
Γ) Corendal Directory **http://www.corendal.com/products/corendal-directory**
Δ) JXplorer **http://jxplorer.org/ - 7/5/2010**
Ε) Luma **http://luma.sourceforge.net/ - 7/5/2010**

Για την παρουσίαση και επεξεργασία του καταλόγου LDAP της πτυχιακής επιλέξαμε το **Apache Directory Studio.**

7.3 Τελική επιλογή συστήματος

Η ανάλυση των διαφόρων προϊόντων λογισμικού υπηρεσίας καταλόγου οδηγεί στη δημιουργία ενός συγκριτικού πίνακα ανάμεσα στα βασικά χαρακτηριστικά τους. Η σύγκριση αφορά σε κάθε περίπτωση τις τελευταίες εκδόσεις των προϊόντων. Τα προϊόντα έχουν παρεμφερείς ιδιότητες με εξαίρεση τον OpenLDAP ο οποίος υπολείπεται σε σχέση με τα εμπορικά προϊόντα σε λειτουργίες όπως ο συγχρονισμός χρηστών και ομάδων μεταξύ OpenLDAP και Active Directory, και ύπαρξη γραφικού περιβάλλοντος διαχείρισης. Το Red Hat DS (ανοιχτού κώδικα) το οποίο εξετάστηκε και παρουσιάζεται συγκριτικά πλήρες σε σχέση με τα υπόλοιπα εμπορικά προϊόντα, καθώς προέρχεται από τον Netscape Directory Server και διαθέτει το σχεδιασμό, τις βασικές αλλά και τις προηγμένες λειτουργίες του. Επιπλέον η ανάπτυξή του υποστηρίζεται από την κοινότητα ανοιχτού κώδικα και επιπλέον διαθέτει την υποστήριξη της Red Hat Systems. Από τα εμπορικά προϊόντα ξεχωρίζουν ο Sun Java System Directory Server, ο IBM Tivoli Directory Server και Novell eDirectory καθώς υποστηρίζουν όλες τις βασικές αλλά και αρκετές προηγμένες λειτουργίες.

Εκτός από τα παραπάνω στοιχεία σημαντικό στοιχείο που λήφθηκε υπόψη στην τελική επιλογή ήταν το τρέχον ποσοστό χρήσης κάθε συστήματος στις εγκαταστάσεις υπηρεσίας καταλόγου στην ακαδημαϊκή κοινότητα. Επιπλέον δόθηκε μεγάλη βαρύτητα στη συνολική απόδοση κάθε συστήματος, στο βαθμό υποστήριξης του από την κοινότητα, στην ωριμότητα του λογισμικού και στις προοπτικές περαιτέρω ανάπτυξης κάθε συστήματος λογισμικού.

Λαμβάνοντας υπόψη όλα τα παραπάνω, για την υλοποίηση της υπηρεσίας καταλόγου επιλέγουμε το λογισμικό υπηρεσίας καταλόγου OpenLDAP, το οποίο αποτελεί το πλέον διαδεδομένο λογισμικό υπηρεσιών καταλόγου παγκοσμίως, έχει συνεχή υποστήριξη από την κοινότητα, αδιάκοπη ανάπτυξη ενώ με βάση πλειάδα μελετών παρουσιάζει τις καλύτερες επιδόσεις ανάμεσα στα διαθέσιμα συστήματα ανοικτού λογισμικού.

7.4 Πλατφόρμα λογισμικού

Δοκιμάζοντας τα διαθέσιμα πακέτα λογισμικού ανοικτού κώδικα καταλήξαμε στην χρησιμοποίηση του πακέτου nuSoap, με βάση την ευκολία εγκατάστασης και της ανάπτυξης εφαρμογών. Το nuSoap χρησιμοποιεί την γλώσσα web PHP με την οποία μπορούμε να αναπτύσσουμε εύκολα εφαρμογές ιστού.

Για τις λειτουργίες των εφαρμογών ιστού θα χρησιμοποιηθεί ο συνδυασμός Apache και PHP. Αυτό σημαίνει πως στον Apache server υπάρχει ένα module ο (PHP interpreter) ο οποίος μειώνει κατά πολύ το κόστος εκτέλεσης, καθώς και αξιοποιεί τα apache server threads (παρέχει πολλαπλά instances για εκτέλεση δυναμικών σελίδων.

Επίσης χρησιμοποιούμε PHP Accelerators (όπως ο APC Accelerator με την μορφή port – pecl package) για να αυξήσουμε αρκετά την απόδοση του interpreter.

Το πακέτο nuSoap δίνεται με την μορφή βιβλιοθήκης, όπου τα αρχεία του μπορούν να συμπεριληφθούν απευθείας από τις σελίδες PHP (π.χ. include nuSoap.php). Εν συνεχεία η διεπαφή προγραμματισμού (API) του nuSoap είναι αυτή που μας παρέχει την δυνατότητα δημιουργίας τόσο παραγωγών υπηρεσιών ιστού (web service providers) όσο και πελατών (web service clients). Με βάση την παραπάνω λειτουργία του Application Programming Interface μπορούμε να δημιουργήσουμε δοκιμαστικούς πελάτες (clients) για τις προς υλοποίηση εφαρμογές μας. Δοκιμαστικές σελίδες έχουν δημιουργηθεί στα πλαίσια της πτυχιακής για την δημιουργία ψεύτικων πελατών και το testing τις σωστής λειτουργίας του nuSoap API.

Τέλος για να επικοινωνήσει το PHP LDAP module (και ο LDAP) με το nuSoap API χρειάζεται η ανάπτυξη του LDAP User Management System.

7.5 Διεπαφή (interface)

Η εφαρμογή που θα αναπτυχθεί υλοποιεί ένα προγραμματιστικό interface κλήσης κατάλληλων λειτουργιών εγγραφών υπό την μορφή συναρτήσεων. Ο ορισμός του interface αυτού γίνεται με τη χρήση της γλώσσας προδιαγραφής Web Services WSDL (Web Service Description Language) την XML γραμματική δηλαδή που έχει προδιαγραφεί από το W3 Consortium για την υλοποίηση και περιγραφή υπηρεσιών ιστού. Η υλοποίηση των υπηρεσιών ιστού πραγματοποιείται με τη χρήση SOAP πάνω από το πρωτόκολλο HTTP, παρέχοντας έτσι συμβατότητα με την πλειοψηφία των αντίστοιχων πακέτων λογισμικού (.Net FrameWork, Java κτλ).
Για κάθε εξωτερική υπηρεσία που πρόκειται να προσπελάσει την εφαρμογή web service θα δημιουργηθεί κατάλληλη σελίδα σε PHP που υλοποιεί το σύνολο των απαιτούμενων συναρτήσεων-λειτουργιών καθώς και η αντίστοιχη προδιαγραφή σε WS + XML. Η πρόσβαση θα πραγματοποιείται με τη χρήση περιορισμού πρόσβασης με βάση την IP του αιτούντα (ως αιτών ορίζεται ο εξυπηρετητής που φιλοξενεί την αντίστοιχη εξωτερική υπηρεσία) καθώς και κατάλληλου ζευγαριού username/password (HTTP Authentication) ανά υπηρεσία ενώ θα προστατεύεται με χρήση του πρωτοκόλλου HTTPS. Η WSDL προδιαγραφή του κάθε web service θα παρέχεται με κατάλληλη κλήση της σελίδας ως εξής:
http://<web server>/<web service>?wsdl

7.5.1 Αρχιτεκτονική

Η εφαρμογή που θα αναπτυχθεί θα εγκατασταθεί σε κατάλληλο εξυπηρετητή της υπηρεσίας. Για κάθε υπηρεσία που αιτείται web service θα δημιουργούνται δύο αρχεία. Ένα δημοσίως προσβάσιμο στο οποίο θα πραγματοποιείται ο ορισμός όλων των συναρτήσεων-λειτουργιών του web service και ο σκελετός κλήσης τους και ένα δεύτερο ιδιωτικό στο οποίο θα πραγματοποιείται η υλοποίηση της κάθε λειτουργίας. Κάθε web service θα πρέπει να πραγματοποιεί πλήρη καταγραφή των καλούμενων λειτουργιών με διατήρηση των ακόλουθων στοιχείων:

- Ημερομηνία και ώρα κλήσης λειτουργίας
- Όνομα λειτουργίας
- Αιτών
- Εγγραφή στην οποία ζητείται η πραγματοποίηση της λειτουργίας (employeenumber ή άλλος μοναδικός δείκτης ανά περίπτωση)
- MD5 δεδομένων
- Αποτέλεσμα εκτέλεσης λειτουργίας
- Τυχόν μήνυμα λάθους

7.5.2 Πλατφόρμα λογισμικού

Καθώς οι υπηρεσίες ιστού θα είναι προσβάσιμες μέσω web (HTTP) η επιλογή της πλατφόρμας ανάπτυξης θα πρέπει να γίνει μεταξύ των διαθέσιμων γλωσσών προγραμματισμού web εφαρμογών. Από την επισκόπηση των διαθέσιμων τεχνολογιών προκύπτουν οι εξής διαθέσιμες γλώσσες προγραμματισμού:

- PHP. Μία ισχυρή, ανοικτού κώδικα και επεκτάσιμη υψηλού επιπέδου γλώσσα προγραμματισμού σε χρήση κυρίως σε περιβάλλοντα Unix/Apache. Μέσω επεκτάσεων η γλώσσα παρέχει τη δυνατότητα εκτέλεσης πληθώρας λειτουργιών και χρήσης βάσεων δεδομένων, πλήθους πρωτοκόλλων κτλ.
- Java. Είναι μία αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems.
- ASP.NET Είναι ένα πλαίσιο υπηρεσιών καταλόγου που έχει αναπτυχθεί και διατίθενται στο εμπόριο από την Microsoft για να επιτρέψει στους προγραμματιστές να κατασκευάσουν δυναμικές ιστοσελίδες, δικτυακές εφαρμογές και υπηρεσίες δικτύου.

Στην περίπτωση της παρούσης μεταπτυχιακής διατριβής λόγω χρήσης πλατφόρμας Unix και με σκοπό την εύκολη και γρήγορη των εφαρμογών επιλέχθηκε η χρήση της γλώσσας PHP.

Μετά από δοκιμή των διαθέσιμων πακέτων ανοικτού λογισμικού επιλέχθηκε η χρήση του πακέτου **nuSoap** με γνώμονα ανάμεσα στα άλλα την ευκολία εγκατάστασης και ανάπτυξης εφαρμογών. Το πακέτο nuSoap λειτουργεί σε περιβάλλον γλώσσας προγραμματισμού εφαρμογών web PHP και παρέχει τη δυνατότητα εύκολης και γρήγορης ανάπτυξης εφαρμογών υπηρεσιών ιστού με το ελάχιστο κόστος προγραμματισμού. Παράλληλα, η χρήση της γλώσσας PHP δίνει τη δυνατότητα χρήσης ήδη διαθέσιμων πακέτων διασύνδεσης με μία ποικιλία πρωτοκόλλων και υπηρεσιών που επιτρέπουν τη γρήγορη και εύκολη ανάπτυξη πολύπλοκων εφαρμογών. Το πακέτο NuSOAP είναι βασισμένο πάνω στο SOAPx4 (το SOAPX4 είναι η υλοποίηση του SOAP για PHP)και αντιγράφει πολλές από τις εφαρμογές του. Παρέχεται από την NuSphere και τον Dietrich Ayala. Είναι μια σειρά από PHP classes (δίχως να χρειάζονται PHP extensions) που επιτρέπει στους προγραμματιστές να δημιουργήσουν και να χρησιμοποιήσουν υπηρεσίες ιστού που είναι βασισμένες σε SOAP 1.1, WSDL 1.1 και HTTP 1.0/1.1.

7.6 Περιγραφή nuSoap API

Η διεπαφή προγραμματισμού εφαρμογών του nuSoap παρέχει συναρτήσεις για τη συγγραφή τόσο πελατών όσο και παρόχων υπηρεσιών ιστού.

Το παρακάτω παράδειγμα έχει κάποιες απλές συναρτήσεις που βρίσκονται μέσα στο αρχείο 'lib/nusoap.php'.

Πριν από οτιδήποτε άλλο απαιτείται η συμπερίληψη (include) του αρχείου *nusoap.php* το οποίο και περιλαμβάνει τους ορισμούς (και τον κώδικα) όλων των συναρτήσεων του πακέτου. Η δημιουργία πελάτη (client) υπηρεσιών ιστού είναι αρκετά απλή και απαιτεί μόνο τα ακόλουθα βήματα:

- Κλήση της συνάρτησης δημιουργίας *new soapclient* με τη web διεύθυνση του παρόχου web service.
- Κλήση της συνάρτησης *call()* με τα κατάλληλα ορίσματα για την κλήση κάθε συναρτησιακής διεπαφής (function) που παρέχει ο πάροχος και χρήση των αποτελεσμάτων που επιστρέφονται.

Οι ορισμοί των παραπάνω συναρτήσεων είναι οι εξής:

```
soapclient($url or $wsdl, boolean is wsdl)

call($operation, $params, $namespace='http://tempuri.org', $soapAction='',
```

```
$headers=false, $rpcParam=null (not used), $style='rpc|document' (default is rpc),
$use='encoded|literal' (what to use when serializing parameters))
```

7.6.1 Παράδειγμα πελάτη για υπηρεσίες ιστού

(ο παρακάτω κώδικας βρίσκεται στο αρχείο «arithmetic.php»)

```
<?php
require_once('lib/nusoap.php');
$client = new soapclient('http://localhost/ws/arithmetic.php');

echo "<br><b>" . date('r') . "</b><br>\n";
$action = $_REQUEST['action'];
if ($action == 'add'){
echo "<br><b>AddNumbers</b><br>\n";
$result = $client->call('AddNumbers',
      array('First' => '10', 'Second' => '4'));
print_r($result);
}
if ($action == 'sub'){
echo "<br><b>SubstractNumbers</b><br>\n";
$result = $client->call('SubstractNumbers',
      array('First' => '10', 'Second' => '4'));
print_r($result);
}
?>
```

Η δημιουργία παροχέα υπηρεσιών ιστού είναι και αυτή απλή αλλά με μεγαλύτερες δυνατότητες:

- Κατ' αρχήν απαιτείται η αρχικοποίηση του παροχέα με κλήση της *new soap_server();*
- Εν συνεχεία πρέπει να οριστεί το WSDL και το πεδίο ονοματολογίας της υπηρεσίας ιστού. Για το WSDL ορίζεται μόνο το URL που θα το περιέχει καθώς αυτό δημιουργείται αυτόματα με βάση τις συναρτήσεις που θα οριστούν. Για τους ορισμούς χρησιμοποιούνται οι κλήσεις configureWSDL και schemaTargetNamespace.
- Το βασικό κομμάτι είναι ο ορισμός των συναρτήσεων που θα παρέχει η υπηρεσία ιστού με χρήση της κλήσης *register* στην οποία παρέχεται το όνομα της κάθε συνάρτησης, η είσοδος και έξοδος, το σχήμα καθώς και τυχόν σχόλια για τη λειτουργία της συνάρτησης.
- Τέλος, ο παροχέας ενεργοποιείται με την κλήση της *service().*

Οι ορισμοί των συναρτήσεων είναι ως εξής:

```
     * @param    string $data usually is the value of $HTTP_RAW_POST_DATA
     * @access   public
     */
     function service($data)

     * register a service function with the server
     *
     * @param    string $name the name of the PHP function, class.method or class..method
     * @param    array $in assoc array of input values: key = param name, value = param
type
     * @param    array $out assoc array of output values: key = param name, value = param
type
     * @param        mixed $namespace the element namespace for the method or false
     * @param        mixed $soapaction the soapaction for the method or false
```

```
      * @param       mixed $style optional (rpc|document) or false Note: when 'document' is
specified, parameter and return wrapp
ers are created for you automatically
      * @param       mixed $use optional (encoded|literal) or false
      * @param       string $documentation optional Description to include in WSDL
      * @param       string $encodingStyle optional (usually
'http://schemas.xmlsoap.org/soap/encoding/' for encoded)
      * @access   public
      */
      function
register($name,$in=array(),$out=array(),$namespace=false,$soapaction=false,$style=false,$
use=false,$documentation='
',$encodingStyle=)


  /**
  * Sets up wsdl object.
  * Acts as a flag to enable internal WSDL generation
  *
  * @param string $serviceName, name of the service
  * @param mixed $namespace optional 'tns' service namespace or false
  * @param mixed $endpoint optional URL of service endpoint or false
  * @param string $style optional (rpc|document) WSDL style (also specified by operation)
  * @param string $transport optional SOAP transport
  * @param mixed $schemaTargetNamespace optional 'types' targetNamespace for service
schema or false
  */
  function configureWSDL($serviceName,$namespace = false,$endpoint = false,$style='rpc',
$transport = 'http://schemas.xmlsoap.org/
soap/http', $schemaTargetNamespace = false)
```

Παράδειγμα παροχέα υπηρεσιών ιστού (απλές συναρτήσεις αριθμητικής):

```
<?php
require_once('lib/nusoap.php');

$server = new soap_server();
$server->configureWSDL('Arithmetic',"http://localhost/ws/arithmetic.php");
$server->wsdl->schemaTargetNamespace="http://localhost/ws/arithmetic.php?wsdl";
$server->register('AddNumbers',
      array('First' => 'xsd:string','Second' => 'xsd:integer'),
      array('Status' => 'xsd:boolean','Result' => 'xsd:string'),
      "http://nic.att.sch.gr/ws/arithmetic.php");
$server->register('SubstractNumbers',
      array('First' => 'xsd:string','Second' => 'xsd:integer'),
      array('Status' => 'xsd:boolean', 'Result' => 'xsd:string'),
      "http://nic.att.sch.gr/ws/arithmetic.php");
function AddNumbers($First,$Second)
{
      $Result = $First + $Second;
      return array('Status' => '1', 'Result' => $Result);
}
function SubstractNumbers($First,$Second)
{
      $Result = $First - $Second;
      return array('Status' => '1', 'Result' => $Result);
}
$server->service($HTTP_RAW_POST_DATA);
?>
```

Η σελίδα ιστού που δημιουργείται στον παροχέα εκτός από διεπαφή τύπου SOAP για εκτέλεση υπηρεσιών ιστού είναι απευθείας προσβάσιμη και μέσω διαδικτύου. Στην περίπτωση αυτή παρέχει γενική και ειδική εποπτεία των παρεχόμενων συναρτήσεων καθώς και δυνατότητα μεταφόρτωσης του WSDL που περιγράφει τη λειτουργία τους. Το WSDL είναι προσβάσιμο στη σελίδα που ορίστηκε κατά την κλήση της συνάρτησης *configureWDSL()* (συνήθως http://<web service url>?wsdl

8 Πιστοποίηση και Πολιτική Ασφάλειας

Καθώς με χρήση των υπηρεσιών ιστού μεταφέρεται ευαίσθητη πληροφορίας (όπως προσωπικά στοιχεία χρηστών, password κτλ) είναι απαραίτητο η χρήση τους να προστατεύεται με τη χρήση ενός ασφαλούς πρωτοκόλλου επικοινωνίας, όπως το HTTPS. Παράλληλα, καθώς οι πελάτες της υπηρεσίας είναι συγκεκριμένοι και περιορισμένοι (συγκεκριμένες web εφαρμογές) προτείνεται η προσθήκη προστασίας με περιορισμό πρόσβασης στις σελίδες των υπηρεσιών ιστού μόνο από τις διευθύνσεις IP των εξυπηρετητών web στις οποίες εκτελούνται οι web εφαρμογές των πελατών.
Η πιστοποίηση βασίζεται στη χρήση του μηχανισμού HTTP Authentication κατά την κλήση των υπηρεσιών ιστού. Ο καλών παρέχει κατάλληλο ζεύγος username/password για την πρόσβαση του στην υπηρεσία. Το ζεύγος αυτό πιστοποιείται με κατάλληλο μηχανισμό. Ο μηχανισμός αυτός μπορεί να είναι ένα απλό password file αλλά προτείνεται να πραγματοποιείται απευθείας πιστοποίηση από την υπηρεσία καταλόγου με χρήση των αντίστοιχων μεθόδων που παρέχονται από τον εξυπηρετητή web. Παρακάτω φαίνεται παράδειγμα ενεργοποίησης πιστοποίησης μέσω LDAP στον εξυπηρετητή apache με χρήση του apache module mod_auth_ldap:

```
<Directory "/usr/local/www/ws">
  AllowOverride None
  Options MultiViews Indexes FollowSymlinks
  AddType application/x-httpd-php .php .php3

  AuthType Basic
  AuthName "web-services"
  AuthLDAPURL ldap://localhost/ou=people,o=papei,c=gr?uid
  Require valid-user
  Satisfy all
  Order deny,allow
  Deny from all
  Allow from 147.102.220.0/24
</Directory>
```

Μετά την επιτυχή πιστοποίηση του χρήστη τα στοιχεία του (username,password) είναι διαθέσιμα με την μορφή server μεταβλητών στις υπηρεσίες ιστού. Το username του χρήστη μπορεί να αντιστοιχιστεί με κατάλληλο τρόπο στο DN του. Αυτό μπορεί να γίνει με δύο τρόπους:

A. Στην περίπτωση στην οποία όλες οι εγγραφές χρηστών είναι κάτω από το ίδιο υποδέντρο (πχ ou=people,dc=<institution>,dc=gr), τότε η αντίστοιχιση μπορεί να γίνει άμεσα με τη φόρμουλα dn: uid=<username>,<dn υποδέντρου>.
B. Στην περίπτωση στην οποία η μορφή του DN του χρήστη δεν μπορεί να αντιστοιχιστεί απευθείας, είναι δυνατόν να πραγματοποιηθεί πρώτα αναζήτηση στην υπηρεσία καταλόγου με βάση το username του χρήστη και από την επιστρεφόμενη εγγραφή να προκύψει το DN. Η αναζήτηση μπορεί να γίνει με τα ακόλουθα στοιχεία:
    - Base: DN κάτω από το οποίο είναι αποθηκευμένες οι εγγραφές των χρηστών
    - Scope: Subtree
    - Filter: (uid=<username>)
    - Επιστρεφόμενα attributes: uid. Καθώς μας ενδιαφέρει μόνο το DN της εγγραφής δεν χρειάζεται να αιτηθούμε να επιστραφεί κάποιο attribute της εγγραφής.

Εάν η αναζήτηση επιστρέψει περισσότερες από μία εγγραφές τότε θεωρούμε ότι απέτυχε και το web service μπορεί να επιστρέψει κατάλληλο μήνυμα λάθους στον καλούντα.

Από τη στιγμή που είναι διαθέσιμο το DN/password του χρήστη, τότε μπορεί να χρησιμοποιηθεί αυτό κατά την πραγματοποίηση αλλαγών στην υπηρεσία καταλόγου όπως για παράδειγμα στην αλλαγή του password του χρήστη. Αυτό έχει ως συνέπεια ότι:

- Δεν απαιτείται η δημιουργία ειδικού χρήστη με πλήρη δικαιώματα αλλαγών σε όλο το δέντρο των χρηστών
- Η αλλαγή στην εγγραφή ενός χρήστη ουσιαστικά πραγματοποιείται από τον ίδιο το χρήστη, κάτι που επιτρέπει τη σωστή, μινιμαλιστική και καθαρή διαμόρφωση της πολιτικής ασφάλειας της υπηρεσίας καταλόγου καθώς απαιτείται απλά ο ορισμός των attributes τα οποία έχει το δικαίωμα να αλλάξει ο χρήστης στην δική του εγγραφή

Ο παραπάνω μηχανισμός είναι χρήσιμος και βέλτιστος στην περίπτωση αλλαγών στην ίδια την εγγραφή του χρήστη (όπως αλλαγή password, στοιχείων υπηρεσιών όπως mail alias κτλ) αλλά υπάρχουν περιπτώσεις στις οποίες δεν κρίνεται σκόπιμο να χρησιμοποιηθεί όπως:

- Αλλαγή στοιχείων χρήστη τα οποία δε θα πρέπει να έχει το δικαίωμα αλλαγής τους ο ίδιος όπως το επίσημο ονοματεπώνυμο, δικαιώματα χρήσης και πρόσβασης σε υπηρεσίες, όρια χρήσης κτλ
- Προσθήκη και διαγραφή εγγραφών

Για τις περιπτώσεις αυτές είναι σκόπιμο να δημιουργηθεί χρήστης με κατάλληλα δικαιώματα ο οποίος θα χρησιμοποιείται από τον καλών. Ο χρήστης αυτός προτείνεται να έχει όνομα που θα αποκαλύπτει απευθείας το ρόλο του (πχ web-services-user) καθώς και να χρησιμοποιείται αποκλειστικά και μόνο για το σκοπό αυτό.

Το username/password του χρήστη μπορεί να χρησιμοποιηθεί στις περιπτώσεις όπου χρήστης της καλούσας εφαρμογής είναι ο ίδιος ο χρήστης, ο οποίος και αιτείται την αλλαγή για την οποία γίνεται η κλήση του web service. Η περίπτωση του μοναδικού χρήστη με πλήρη δικαιώματα συνήθως χρησιμοποιείται στις περιπτώσεις αυτόματου συγχρονισμού μεταξύ ετερογενών βάσεων δεδομένων χρηστών. Η πρωτογενής πληροφορία για τους χρήστες μίας εφαρμογής μπορεί να συντηρείται σε μία βάση δεδομένων και ο συγχρονισμός με την υπηρεσία καταλόγου να πραγματοποιείται με τη χρήση υπηρεσιών ιστού.

Ο παραπάνω μηχανισμός παρουσιάζει το εγγενές μειονέκτημα ότι απαιτεί την μεταφορά και χρήση των passwords των χρηστών. Η τρέχουσα πρακτική στην πιστοποίηση χρηστών σε σελίδες web προβλέπει τη χρήση τεχνολογίας Single Sign On, η οποία επιτρέπει την πιστοποίηση του χρήστη σε ένα κεντρικό σημείο και την αποφυγή μεταφοράς των διαπιστευτηρίων του ανάμεσα σε εφαρμογές. Οι δορυφορικές εφαρμογές εμπιστεύονται την κεντρική υπηρεσία πιστοποίησης και λαμβάνουν ψηφιακά υπογεγραμμένη την ταυτότητα του χρήστη.
Στην περίπτωση των υπηρεσιών ιστού η χρήση Single Sign On δεν είναι δυνατή για τους ακόλουθους λόγους:

- Οι λειτουργίες σε πολλές περιπτώσεις πραγματοποιούνται offline και χωρίς την άμεση αλληλεπίδραση του χρήστη (πχ συγχρονισμός μεταξύ ετερογενών βάσεων δεδομένων μία φορά την μέρα).
- Η πιστοποίηση στην υπηρεσία καταλόγου (LDAP Bind) απαιτεί τη χρήση κατάλληλου ζεύγους Bind DN/password με συνέπεια να είναι απαραίτητα κατάλληλα διαπιστευτήρια.

Για την υπέρβαση του μειονεκτήματος αυτού μπορεί απλά να μη χρησιμοποιούνται τα διαπιστευτήρια των χρηστών αλλά ένας κεντρικός λογαριασμός με αυξημένα δικαιώματα στο δέντρο των χρηστών. Η web εφαρμογή – πελάτης των υπηρεσιών ιστού μπορεί να πιστοποιήσει μέσω Single Sign On τον χρήστη και στη συνέχεια να αποστείλει την αίτηση εκ μέρους του παρέχοντας τα διαπιστευτήρια του κεντρικού λογαριασμού για πιστοποίηση στο web service.

8.1 Προτάσεις Βελτίωσης

Παρακάτω παρατίθενται ορισμένες προτάσεις βελτίωσης και εξέλιξης των λειτουργιών μέσω υπηρεσιών ιστού ώστε να μπορούν να λειτουργήσουν καλύτερα σε πραγματικά περιβάλλοντα λειτουργίας.

8.1.1 Versioning ερωτημάτων/απαντήσεων

Μία πρόταση είναι η προσθήκη αριθμού έκδοσης σχήματος στις κλήσεις των λειτουργιών όσο και στις αντίστοιχες απαντήσεις ανά interface. Ο αριθμός έκδοσης ορίζει την έκδοση του σχήματος του interface το οποίο χρησιμοποιείται ανά πάσα στιγμή. Με την προσθήκη του αριθμού αυτού είναι δυνατή η εύκολη και ομαλή μετάβαση σε νεότερες εκδόσεις ενός interface χωρίς να υπάρχει ανάγκη δημιουργίας νέου ιστοτόπου για το νέο interface. Κατά την κλήση των συναρτήσεων προστίθεται και ο αριθμός έκδοσης με συνέπεια η καλούμενη συνάρτηση να έχει άμεση γνώση των αναμενόμενων ορισμάτων και να μπορεί ανάλογα με την έκδοση να καλέσει διαφορετική συνάρτηση υλοποίησης της λειτουργίας. Παράλληλα, στην επιστρεφόμενη απάντηση θα προστίθεται ο αριθμός έκδοσης ώστε και ο καλώντας να γνωρίζει με βάση τον αριθμό το σχήμα της απάντησης που θα πρέπει να αναμένει.

Από πλευράς υλοποίησης η προσθήκη του αριθμού έκδοσης απλά απαιτεί ένα νέο όρισμα στο σχήμα των συναρτήσεων που υλοποιεί το interface τόσο στην είσοδο όσο και στην έξοδο. Ο αριθμός έκδοσης μπορεί να είναι αλφαριθμητικό (xsd:string) της μορφής *major.minor* (πχ 1.1).

Σε ένα πραγματικό σύστημα θα ήταν απαραίτητο να έχουμε λειτουργίες ασφαλείας που θα καθορίζουν την πολιτική της εφαρμογής όσον αναφορά την επικοινωνία με πολλαπλούς χρήστες σε απομακρυσμένα σημεία. Για αυτό τον λόγο έχουν αναπτυχθεί ορισμένες προδιαγραφές ή βρίσκονται ακόμα σε ανάπτυξη για την επέκταση των δυνατοτήτων των υπηρεσιών ιστού. Αυτές οι προδιαγραφές αναφέρονται γενικά ως WS-*. Μερικές από τις προδιαγραφές είναι οι εξής:

- WS-Security

Προσδιορίζει τον τρόπο χρήσης κρυπτογράφησης XML και XML Υπογραφή (XML Encryption, XML Signature) στο SOAP για την ασφαλή ανταλλαγή μηνυμάτων, ως εναλλακτική λύση ή παράλληλα με τη χρήση HTTPS για να παρέχουμε ασφάλεια στο κανάλι.

Περιγράφει πώς επισυνάπτονται υπογραφές και κρυπτογραφούνται οι κεφαλίδες στα SOAP μηνύματα. In addition, it describes how to attach security tokens, including binary security tokens such as X.509 certificates and Kerberos tickets, to messages. Επιπλέον, περιγράφει πώς προστίθενται στα μηνύματα κώδικες ασφάλειας, όπως πιστοποιητικά X.509.

Η WS-Security ενσωματώνει χαρακτηριστικά ασφαλείας στην κεφαλίδα ενός μηνύματος SOAP και λειτουργεί στο επίπεδο εφαρμογών (application layer). Έτσι εξασφαλίζεται ασφάλεια end-to-end.

- WS-Transaction

Μια προδιαγραφή Web Υπηρεσιών που περιγράφει τους τύπους συντονισμού. Αυτό ορίζει δύο τύπους συντονισμού: Atomic Transaction (AT) για τις ατομικές επιχειρήσεις, και επαγγελματικής δραστηριότητας (BA) για τις μεγάλες συναλλαγές. Οι Προγραμματιστές μπορούν να χρησιμοποιήσουν ένα ή δύο από αυτούς τους τύπους συντονισμού όταν δημιουργούν εφαρμογές.

- WS-Addressing

Μια προδιαγραφή που καθορίζει τον τρόπο εισαγωγής διευθύνσεων στην κεφαλίδα του SOAP. Με αυτήν την προδιαγραφή περιγράφεται η επικοινωνία μεταξύ των υπηρεσιών ιστού όταν χρειάζεται να ανταλλάσσονται διευθύνσεις.

Με αυτήν τυποποιείται ο τρόπος να συμπεριλαμβάνονται δεδομένα δρομολόγησης εντός των SOAP κεφαλίδων (headers). Αντί να στηρίζονται οι υπηρεσίες ιστού στο network-level επίπεδο δικτύου για να μεταφέρουν πληροφορίες δρομολόγησης, χρησιμοποιώντας το WS-Addressing μπορεί να περιλαμβάνουν τα δικά τους δεδομένα δρομολόγησης σε τυποποιημένη κεφαλίδα SOAP.
Το επίπεδο δικτύου είναι υπεύθυνο μόνο για την παράδοση του μηνύματος σε έναν αποστολέα που μπορεί να υπολογίσει τα WS-Addressing μεταδεδομένα. Μόλις το μήνυμα φτάνει στο αποστολέα που ορίζεται από την URI, η εργασία του επίπεδου δικτύου έχει τελειώσει.

Το WS-Addressing υποστηρίζει τη χρήση της ασύγχρονης αλληλεπίδρασης με τον καθορισμό μιας κοινής κεφαλίδας SOAP (WSA: ReplyTo) που περιέχει το τελικό σημείο αναφοράς (EPR) στο οποίο η απάντηση είναι να σταλεί. Ο φορέας παροχής υπηρεσιών μεταδίδει το απαντητικό μήνυμα σε μια ξεχωριστή σύνδεση με το WSA: ReplyTo καταληκτικό σημείο.

Το τελικό σημείο αναφοράς, Endpoint Reference (EPR), είναι ένα XML κείμενο που παρέχει πληροφορίες χρήσιμες για να δρομολογηθεί ένα μήνυμα σε ένα Web service. Αυτό περιλαμβάνει την διεύθυνση προορισμού του μηνύματος, τυχόν πρόσθετες παραμέτρους (που ονομάζονται παράμετροι αναφοράς) που είναι αναγκαίες για να δρομολογηθεί το μήνυμα προς τον προορισμό, και προαιρετικά μεταδεδομένα (όπως WSDL ή WS-Policy) για την υπηρεσία.

Οι ιδιότητες του μηνύματος δρομολόγησης είναι οι παρακάτω:
α) ο προορισμός του μηνύματος URI (Message destination), β) το καταληκτικό σημείο της πηγής που απέστειλε αυτό το μήνυμα EPR (Source endpoint), γ) το τελικό σημείο για το οποίο τα μηνύματα θα πρέπει να αποσταλούν (EPR) (Reply endpoint), δ) το τελικό σημείο για το οποίο τα μηνύματα σφάλματος θα πρέπει να αποσταλούν (EPR) (Fault endpoint), ε) το μοναδικό αναγνωριστικό μηνύματος (URI Unique message ID), ζ) η σχέση με τα προηγούμενα μηνύματα (Ένα ζευγάρι URIs).

8.1.2 Uniform Resource Identifier (URI)

Για να κατανοήσουμε καλύτερα πως λειτουργεί ένα μήνυμα δρομολόγησης αναγκαία είναι η αναφορά στο Uniform Resource Identifier ή αλλιώς URI.

Στην πληροφορική, ένας Uniform Resource Identifier (URI) αποτελείται από μια σειρά χαρακτήρων που χρησιμοποιούνται για τον εντοπισμό ή το όνομα ενός πόρου για το Διαδίκτυο. Η ταυτοποίηση αυτή επιτρέπει την αλληλεπίδραση μεταξύ των πόρων ενός δικτύου (όπως το World Wide Web) χρησιμοποιώντας ειδικά πρωτόκολλα.

Το URL (locator - εντοπισμού) και το URN (name - όνομα) αποτελούν υποσύνολα του URI. Από τεχνικής άποψης μπορεί να καθοριστεί μια διεύθυνση URL ως URI. Εκτός από τον προσδιορισμό ενός πόρου, παρέχει ένα μέσο το για να περιγραφεί η τοποθεσία του δικτύου. Για παράδειγμα, η διεύθυνση URL http://www.papei.gr/ εντοπίζει έναν πόρο (papei home page) και σημαίνει ότι ο χρήστης μπορεί να πάρει μια αναπαράσταση του εν λόγω πόρου (όπως ο κώδικας HTML της αρχικής σελίδας του) μέσω HTTP από ένα όνομα www.papei.gr. Το Uniform Resource Name (URN) περιλαμβάνει ένα URI που προσδιορίζει έναν πόρο με βάση το όνομα σε ένα συγκεκριμένο πεδίο ονομάτων. Μπορεί να χρησιμοποιηθεί ένα URN για να αναφερθεί για έναν πόρο δίχως να δηλώνεται η θέση του ή πώς να αποκτηθεί πρόσβαση. Για παράδειγμα, το URN: ISBN :7-258-45789-2 είναι ένα URI που αναφέρει το αρχείο (πόρο) που θέλουμε, δηλαδή International Standard Book Number (ISBN), καθώς και ότι μιλάμε για ένα βιβλίο, αλλά δεν δείχνει πού και πώς μπορούμε να βρούμε ένα πραγματικό αντίγραφό του.

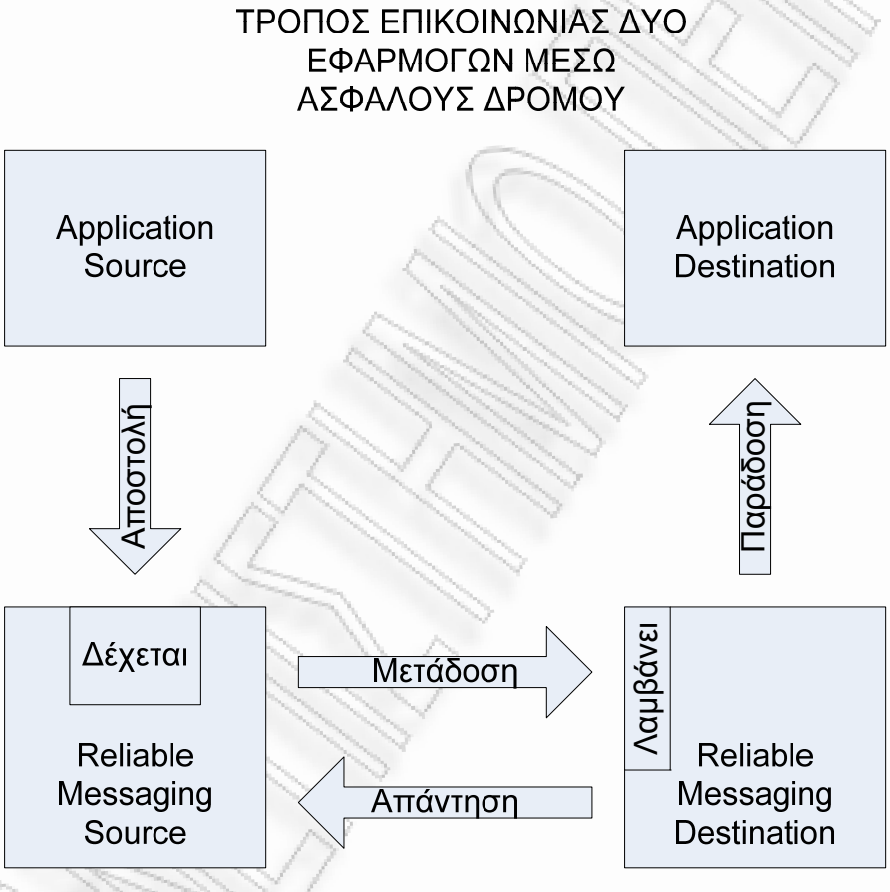
- WS-Reliability

Το WS-Reliability είναι ένα πρωτόκολλο προτύπου OASIS για αξιόπιστη ανταλλαγή μηνυμάτων μεταξύ δύο υπηρεσιών Web. Το SOAP πάνω από HTTP δεν επαρκεί, όταν

ένα πρωτόκολλο ανταλλαγής μηνυμάτων πρέπει να εγγυάται κάποιο επίπεδο αξιοπιστίας και ασφάλειας. Η προδιαγραφή αυτή έχει σχεδιαστεί για χρήση σε συνδυασμό με άλλα συμπληρωματικά πρωτόκολλα και βασίζεται σε προηγούμενες προδιαγραφές, όπως π.χ., ebXML(e business) Message Service και WS-Reliable Messaging που αναλύεται παρακάτω.

- WS-ReliableMessaging

Το WS- ReliableMessaging περιγράφει ένα πρωτόκολλο που επιτρέπει την αξιόπιστη παράδοση μηνυμάτων SOAP μεταξύ των κατανεμημένων εφαρμογών ακόμα και όταν υπάρχει πρόβλημα στο δίκτυο.



Σχήμα 6

Μια πηγή εφαρμογής (Application Source - AS) επιθυμεί να στείλει μηνύματα με αξιοπιστία σε ένα προορισμό εφαρμογής (Application Destination - AD) πάνω από μια αξιόπιστη υποδομή. Για να επιτευχθεί αυτό κάνουν χρήση μιας αξιόπιστης πηγής μηνυμάτων (RMS) και ενός αξιόπιστου προορισμού μηνύματος (RMD). Ο AS στέλνει ένα μήνυμα προς την RMS. Η RMS χρησιμοποιεί το WS-Reliable Messaging (WS-RM) πρωτόκολλο για τη μετάδοση του μηνύματος προς τον RMD. Ο RMD παραδίδει το μήνυμα στο AD. Εάν ο RMS δεν μπορεί να μεταδώσει το μήνυμα προς τον RMD για κάποιο λόγο, θα πρέπει να υποδείξει στον AS ότι το μήνυμα δεν διαβιβάστηκε. Ο AS και ο RMS μπορούν να υλοποιηθούν εντός της ίδιας διαδικασίας ή μπορεί να υλοποιούνται σε διαφορετικά στοιχεία. Ομοίως, ο AD και ο RMD μπορεί να υλοποιηθούν εντός της ίδιας διαδικασίας ή μπορεί να υλοποιούνται σε διαφορετικά στοιχεία.

Το σημαντικό που πρέπει να αναφέρουμε είναι ότι η προδιαγραφή WS-RM ασχολείται μόνο με το περιεχόμενο και τη συμπεριφορά των μηνυμάτων που εμφανίζονται "on the wire". Πώς στέλνονται τα μηνύματα από το AS στην RMS και πώς παραδίδονται από το RMD στο AD, αν τα μηνύματα βρίσκονται σταθερά στο δίσκο, ή περνάνε από τη μνήμη; Κανένα από αυτά δεν αποτελούν μέρος της προδιαγραφής WS-RM.

Το πρωτόκολλο WS-RM ορίζει και υποστηρίζει μια σειρά Εγγυητικών παράδοσης (Delivery Assurances). Αυτές είναι:

- AtLeastOnce - Κάθε μήνυμα θα παραδοθεί στον AD τουλάχιστον μία φορά. Αν ένα μήνυμα δεν μπορεί να παραδοθεί, ένα λάθος πρέπει να παρουσιαστεί από το RMS και / ή το RMD. Τα μηνύματα μπορούν να παραδοθούν στο AD πάνω από μία φορά (δηλαδή ο AD μπορεί να πάρει διπλά μηνύματα).
- AtMostOnce - Κάθε μήνυμα θα παραδοθεί στον AD το πολύ μία φορά. Μηνύματα μπορεί να μην παραδοθούν στο AD, αλλά ο AD ποτέ δεν θα πάρει διπλά μηνύματα.
- ExactlyOnce - Κάθε μήνυμα θα παραδοθεί στον AD ακριβώς μια φορά. Αν ένα μήνυμα δεν μπορεί να παραδοθεί, πρέπει να παρουσιαστεί ένα λάθος από το RMS και / ή το RMD. Ο AD ποτέ δεν θα πάρει διπλά μηνύματα.
- InOrder – τα μηνύματα θα πρέπει να παραδοθούν από το RMD στον AD. κατά τη σειρά που αποστέλλονται από τον AS στο RMS. Η διαβεβαίωση αυτή μπορεί να συνδυαστεί με οποιαδήποτε από τις παραπάνω εγγυήσεις.

Στην συγκεκριμένη εφαρμογή που παρουσιάζουμε δεν έχουμε εισαγάγει κανένα από τα παραπάνω πρωτόκολλα ασφαλείας και δεν χρησιμοποιήσαμε κάποια κωδικοποίηση στο κανάλι επικοινωνίας με τον κεντρικό υπολογιστή. Αφήνοντας τις προκαθορισμένες παραμέτρους του διακομιστή οριοθετούμε την ασφάλεια σε ένα σύστημα όταν ολοκληρωθεί. Συνήθως σε ένα ολοκληρωμένο σύστημα η ασφάλεια παρέχεται από κάποια εξωτερική εφαρμογή και όχι από το ίδιο το web service interface.

8.2 Ουρές Αναμονής

Η μέχρι τώρα περιγραφή της υλοποίησης του μηχανισμού των υπηρεσιών ιστού εμπεριείχε την υπόθεση ότι τα αντίστοιχα interface είναι συνεχώς διαθέσιμα ώστε κάθε κλήση των συναρτήσεων να λαμβάνει άμεσα απάντηση με μόνη καθυστέρηση τον χρόνο επεξεργασίας και εκτέλεσης των λειτουργιών. Η πραγματικότητα όμως είναι ότι τα interface είναι δυνατόν να μην είναι διαθέσιμα για περιορισμένο ή αυξημένο χρονικό διάστημα για διάφορους λόγους όπως:

- Απώλεια παροχής ρεύματος στον εξυπηρετητή που τα φιλοξενεί
- Πρόβλημα δικτύου στη διαδρομή μεταξύ πελάτη και παρόχου των interface
- Πρόβλημα λογισμικού στο λειτουργικό του εξυπηρετητή
- Πρόβλημα στη λειτουργία του λογισμικού web πάνω στο οποίο εκτελούνται τα interface

Η διαθεσιμότητα των υπηρεσιών ιστού μπορεί να αυξηθεί με τη χρήση διατάξεων υψηλής διαθεσιμότητας όπως UPS (για αδιάλειπτη παροχή ρεύματος), διπλά στοιχεία εξυπηρετητών (τροφοδοτικά, κάρτες δικτύου, RAID-X συστοιχίες δίσκων) και πολλαπλούς εξυπηρετητές και στοιχεία δικτύου (switches, routers).
Όλες αυτές οι διατάξεις μειώνουν την πιθανότητα μη διαθεσιμότητας της υπηρεσίας αλλά δεν την εξαφανίζουν. Παράλληλα, είναι δυνατόν παροδικά προβλήματα (πχ δικτύου) να αυξήσουν το χρόνο εξυπηρέτησης μίας λειτουργίας ή ακόμα και να οδηγήσουν στην προσωρινή αποτυχία της με συνέπεια να απαιτείται η επανεκτέλεση της.

Με βάση τα παραπάνω, είναι σημαντικό να δημιουργηθεί κατάλληλη ουρά αναμονής από την πλευρά του πελάτη κατά την κλήση των υπηρεσιών ιστού. Κάτι τέτοιο είναι σημαντικό ιδιαίτερα στις περιπτώσεις που δεν υπάρχει ανθρώπινη αλληλεπίδραση όπως στις περιπτώσεις αυτόματου συγχρονισμού μεταξύ βάσεων δεδομένων. Ο μηχανισμός που θα υλοποιηθεί θα πρέπει να προβλέπει κατάλληλη δομή ουράς αναμονής στην οποία θα αποθηκεύονται οι κλήσεις που θα πρέπει να πραγματοποιηθούν από τις κατάλληλες

διεργασίες συγχρονισμού. Εξωτερική διεργασία θα αναλαμβάνει να εκτελέσει τις λειτουργίες αυτές (αναλαμβάνοντας να πραγματοποιήσει αυτόματα τυχόν failover μεταξύ των παρόχων υπηρεσιών ιστού εάν απαιτείται) μέχρι να λάβει απάντηση. Οι λειτουργίες θα πρέπει να εκτελούνται σειριακά με τη σειρά αποθήκευσής τους, ώστε να λαμβάνεται πρόνοια για λειτουργίες οι οποίες για να εκτελεστούν βασίζονται στην επιτυχή εκτέλεση προηγούμενης λειτουργίας (πχ προσθήκη χρήστη κάτω από μονάδα μόνο μετά την επιτυχή προσθήκη της μονάδας). Η ουρά αναμονής θα πρέπει να επιτρέπει τον ορισμό λειτουργιών βασιζόμενων σε προηγούμενες λειτουργίες (πχ με την απόδοση transaction id ανά λειτουργία και τον ορισμό transaction id στο οποίο βασίζεται η λειτουργία) ώστε μόνο επιτυχής ολοκλήρωση μίας λειτουργίας να οδηγεί στην εκτέλεση επόμενης λειτουργίας. Σε περίπτωση που μία λειτουργία είναι ανεπιτυχής η διεργασία αποστολής θα πρέπει να καταγράφει το μήνυμα λάθους, να μην εκτελεί τυχόν λειτουργίες που βασίζονται σε αυτή και να ενημερώνει το διαχειριστή για την επίλυση του προβλήματος.

9 Ολοκληρωμένη Υπηρεσία

9.1 Αρχιτεκτονική υπηρεσίας

Η εφαρμογή εγκαθίσταται σε εξυπηρετητή ο οποίος διαθέτει ήδη εγκατεστημένο web server Apache με υποστήριξη για PHP. Για παροχή υψηλής διαθεσιμότητας, προτείνεται να εγκατασταθεί σε δύο κατάλληλους εξυπηρετητές, σε αρχιτεκτονική υψηλής διαθεσιμότητας. Μία λειτουργεί ως κύρια εφαρμογή και η δεύτερη ως αντίγραφο ασφαλείας για την παροχή υψηλής διαθεσιμότητας.
Ο κατάλογος στον οποίο είναι εγκατεστημένες οι υπηρεσίες ιστού προτείνεται να είναι ο ./usr/local/www/data/ws (εμείς τον εγκαταστήσαμε στο /var/www/html/ws όπου βρίσκονται οι δημοσίως προσβάσιμες σελίδες web του apache).
Για κάθε υπηρεσία που αιτείται υπηρεσία ιστού θα δημιουργούνται δύο αρχεία. Ένα δημοσίως προσβάσιμο στο οποίο θα πραγματοποιείται ο ορισμός όλων των συναρτήσεων-λειτουργιών της υπηρεσίας ιστού και ο σκελετός κλήσης τους και ένα δεύτερο ιδιωτικό στο οποίο θα πραγματοποιείται η υλοποίηση της κάθε λειτουργίας. Κάθε υπηρεσία ιστού πραγματοποιεί πλήρη καταγραφή των καλούμενων λειτουργιών με διατήρηση των ακόλουθων στοιχείων (μία γραμμή ανά κλήση συνάρτησης).

- Ημερομηνία και ώρα κλήσης λειτουργίας
- Όνομα λειτουργίας
- Αιτών
- Εγγραφή στην οποία ζητείται η πραγματοποίηση της λειτουργίας (όνομα χρήστη-username- ή άλλος μοναδικός δείκτης ανά περίπτωση)
- Αποτέλεσμα εκτέλεσης λειτουργίας
- Τυχόν μήνυμα λάθους

Η καταγραφή αυτή γίνεται σε ένα αρχείο ανά υπηρεσία κάτω από τον κατάλογο /var/log/web_services.

Fri, 06 Mar 2009 16:28:05 +0200 Reader WS: GetUserRolesDesc(Client=192.168.240.133): Bind Username= 'cn=manager, dc=vkolyvas,dc=gr',UserName='admin1'

Fri, 06 Mar 2009 16:28:05 +0200 Reader WS: GetUserRolesDesc: User was not found

Fri, 06 Mar 2009 16:33:46 +0200 Reader WS: GetUserRolesDesc(Client=192.168.240.133): Bind Username= 'cn=manager, dc=vkolyvas,dc=gr',UserName='admin1'

Fri, 06 Mar 2009 16:34:25 +0200 Reader WS: GetUserRolesDesc(Client=192.168.240.133): Bind Username= 'cn=manager, dc=vkolyvas,dc=gr',UserName='admin1'

Fri, 06 Mar 2009 16:39:43 +0200 Reader WS: GetUserRolesDesc(Client=192.168.240.133): Bind Username= 'cn=manager, dc=vkolyvas,dc=gr',UserName='admin1'

Fri, 06 Mar 2009 16:54:29 +0200 Reader WS: GetUserRolesDesc(Client=192.168.240.133): Bind Username= 'cn=manager, dc=vkolyvas,dc=gr',UserName='admin1'

```
Fri, 06 Mar 2009 16:59:47 +0200 Reader WS: GetUserRolesDesc(Client=192.168.240.133): Bind Username= 'cn=manager, dc=vkolyvas,dc=gr',UserName='admin1'

Fri, 06 Mar 2009 17:00:55 +0200 Reader WS: GetUserRolesDesc(Client=192.168.240.133): Bind Username= 'cn=manager, dc=vkolyvas,dc=gr',UserName='admin1'

Fri, 06 Mar 2009 17:06:11 +0200 Reader WS: GetUserRolesDesc(Client=192.168.240.133): Bind Username= 'cn=manager, dc=vkolyvas,dc=gr',UserName='admin1'

Fri, 06 Mar 2009 17:06:11 +0200 Reader WS: GetUserRolesDesc: User was not found

root@vkolyvas
```

9.1.1 Βιβλιοθήκη λειτουργιών LUMS

Η υλοποίηση του κώδικα πραγματοποίησης αλλαγών στην υπηρεσία καταλόγου (προσθήκη εγγραφής, αλλαγή, διαγραφή) παρουσιάζει ιδιαίτερες απαιτήσεις όπως:

- Δυνατότητα για παραγόμενες τιμές σε πεδία (attributes) με βάση τις τιμές άλλων (εισαγόμενων από το χρήστη) και με δυνατότητα εκτέλεσης οποιασδήποτε δυνατής έκφρασης ή συνάρτησης PHP για την παραγωγή των τιμών.
- Ορισμό περιορισμών στον τύπο των τιμών των πεδίων (τύπος email, τηλέφωνο κτλ) όσο και επιπλέον περιορισμών (πχ επιλογή δυνατών τιμών από συγκεκριμένη λίστα τιμών)
- Ειδικούς τύπους πεδίων όπως αυτόματης παραγωγής αυξανόμενων τιμών (auto-increment)
- Βοηθητικές λειτουργίες για τη διατήρηση της ακεραιότητας των δεδομένων στο δέντρο πληροφοριών όπως:
    A. Αναφορική Ακεραιότητα (Referential Integrity): Σε περίπτωση διαγραφής εγγραφής, αυτή διαγράφεται από το σύνολο των αντικειμένων στα οποία έχει προστεθεί ως αναφορά (συνήθως, ομάδες χρηστών)
    B. Μοναδικότητα Πεδίου (Attribute Uniqueness): Σε περιπτώσεις απαιτείται η τιμή ενός πεδίου να είναι μοναδική σε ολόκληρο το δέντρο πληροφοριών. Παραδείγματα είναι το όνομα χρήστη (username), διεύθυνση ηλεκτρονικού ταχυδρομείου (email) και κωδικός εργαζόμενου (employeenumber) ενός χρήστη.
- Δυνατότητα πραγματοποίησης εξωτερικών διεργασιών πριν και μετά την πραγματοποίηση των αλλαγών σε εγγραφή στην υπηρεσία καταλόγου.

Για την καλύτερη ικανοποίηση των παραπάνω απαιτήσεων, την μείωση του απαιτούμενου κώδικα και τη δυνατότητα επαναχρησιμοποίησης του σε οποιαδήποτε νέα εφαρμογή υπηρεσίας ιστού, δημιουργήθηκε κατάλληλη διεπαφή προγραμματισμού εφαρμογών σε PHP με όνομα LDAP User Management Service (LUMS). Η διεπαφή αυτή δίνεται με την μορφή βιβλιοθήκης η οποία μπορεί να συμπεριληφθεί απευθείας από οποιαδήποτε σελίδα PHP απαιτείται. Η βιβλιοθήκη παρέχει βασικές συναρτήσεις αλληλεπίδρασης με την υπηρεσία καταλόγου και ένα ιδιαίτερα εξελιγμένο αρχείο διαμόρφωσης της βιβλιοθήκης που παρέχει πληθώρα δυνατοτήτων..

9.1.2 Συναρτήσεις

Οι συναρτήσεις που είναι διαθέσιμες είναι οι ακόλουθες:

- function LUMS_ldap_search($L_binddn, $L_bindpassword, $L_basedn, $L_scope, $L_filter, $L_attrs_array)

- Περιγραφή: Πραγματοποιεί αναζήτηση LDAP
- Είσοδος:
  - L_binddn: Το διακεκριμένο όνομα χρήστη κατά τη διαδικασία δέσμευσης (Bind Distinguish Name)
  - L_bindpassword: Το συνθηματικό που θα χρησιμοποιηθεί κατά τη διαδικασία δέσμευσης (bind)
  - L_basedn: Η βάση (base) της αναζήτησης
  - L_scope: Το πεδίο (scope) της αναζήτησης
  - L_filter: Το φίλτρο της αναζήτησης
  - L_attrs_array: Προαιρετική δομή τύπου array με τα πεδία που θα επιστραφούν. Αν δε δοθεί επιστρέφονται όλα τα πεδία
- Έξοδος: Οι εγγραφές που βρέθηκαν (όπως επιστρέφονται από τη συνάρτηση ldap_search() της PHP), είτε κατάλληλο αλφαριθμητικό με το μήνυμα λάθους σε περίπτωση αποτυχίας

- function LUMS_ldap_add_entry($L_binddn, $L_bindpassword, $L_object_type, $L_entrydn, $L_entry_info).
  - Περιγραφή: Προσθήκη εγγραφής
  - Είσοδος:
    - L_binddn: Το διακεκριμένο όνομα χρήστη κατά τη διαδικασία δέσμευσης (Bind Distinguish Name)
    - L_bindpassword: Το συνθηματικό που θα χρησιμοποιηθεί κατά τη διαδικασία δέσμευσης (bind)
    - L_object_type: Ο τύπος του αντικειμένου
    - L_entrydn: Το διακεκριμένο όνομα της νέας εγγραφής
    - L_entry_info: Τα στοιχεία της νέας εγγραφής υπό την μορφή πίνακα
  - Έξοδος: 0 για επιτυχία ή κατάλληλο αλφαριθμητικό με το μήνυμα λάθους σε περίπτωση αποτυχίας
- function LUMS_ldap_change_password($L_binddn, $L_bindpassword, $L_entrydn, $L_newpassword)
  - Περιγραφή: Αλλαγή του συνθηματικού εγγραφής
  - Είσοδος:
    - L_binddn: Το διακεκριμένο όνομα χρήστη κατά τη διαδικασία δέσμευσης (Bind Distinguish Name)
    - L_bindpassword: Το συνθηματικό που θα χρησιμοποιηθεί κατά τη διαδικασία δέσμευσης (bind)
    - L_entrydn: Το διακεκριμένο όνομα της εγγραφής
    - L_newpassword: Το νέο συνθηματικό της εγγραφής
  - Έξοδος: 0 για επιτυχία ή κατάλληλο αλφαριθμητικό με το μήνυμα λάθους σε περίπτωση αποτυχίας
- function LUMS_ldap_modify_entry($L_binddn, $L_bindpassword, $L_object_type, $L_entrydn, $L_change_info)
  - Περιγραφή: Αλλαγή στοιχείων εγγραφής (με τον ίδιο τρόπο όπως η συνάρτηση ldap_modify() της PHP)
  - Είσοδος

- L_binddn: Το διακεκριμένο όνομα χρήστη κατά τη διαδικασία δέσμευσης (Bind Distinguish Name)
- L_bindpassword: Το συνθηματικό που θα χρησιμοποιηθεί κατά τη διαδικασία δέσμευσης (bind)
- L_object_type: Ο τύπος του αντικειμένου
- L_entrydn: Το διακεκριμένο όνομα της εγγραφής
- L_change_info: Τα στοιχεία της εγγραφής που απαιτούν αλλαγή

o Έξοδος: 0 για επιτυχία ή κατάλληλο αλφαριθμητικό με το μήνυμα λάθους σε περίπτωση αποτυχίας

- function LUMS_ldap_delete_entry($L_binddn, $L_bindpassword, $L_object_type, $L_entrydn)
  - o Περιγραφή: Διαγραφή εγγραφής
  - o Είσοδος
    - L_binddn: Το διακεκριμένο όνομα χρήστη κατά τη διαδικασία δέσμευσης (Bind Distinguish Name)
    - L_bindpassword: Το συνθηματικό που θα χρησιμοποιηθεί κατά τη διαδικασία δέσμευσης (bind)
    - L_object_type: Ο τύπος του αντικειμένου
    - L_entrydn: Το διακεκριμένο όνομα της εγγραφής
  - o Έξοδος: 0 για επιτυχία ή κατάλληλο αλφαριθμητικό με το μήνυμα λάθους σε περίπτωση αποτυχίας
- function LUMS_ldap_rename_entry($L_binddn, $L_bindpassword, $L_object_type, $L_entrydn, $L_newrdn, $L_newparent, $L_deleteoldrdn)
  - o Περιγραφή: Μετονομασία της εγγραφής (πραγματοποίηση ModRDN)
  - o Είσοδος
    - L_binddn: Το διακεκριμένο όνομα χρήστη κατά τη διαδικασία δέσμευσης (Bind Distinguish Name)
    - L_bindpassword: Το συνθηματικό που θα χρησιμοποιηθεί κατά τη διαδικασία δέσμευσης (bind)
    - L_object_type: Ο τύπος του αντικειμένου
    - L_entrydn: Το διακεκριμένο όνομα της εγγραφής
    - L_newrdn: Το νέο σχετικό διακεκριμένο όνομα (Relative Distinguished Name) της εγγραφής
    - L_newparent: Ο νέος πατέρας της εγγραφής
    - L_deleteoldrdn: Δυαδική μεταβλητή που ορίζει κατά πόσον θα διαγραφεί το παλαιό σχετικό διακεκριμένο όνομα.
  - o Έξοδος: 0 για επιτυχία ή κατάλληλο αλφαριθμητικό με το μήνυμα λάθους σε περίπτωση αποτυχίας

Σε όλες τις περιπτώσεις εφόσον το L_binddn παραμείνει κενό θα χρησιμοποιηθούν τα στοιχεία που έχουν οριστεί στο αρχείο διαμόρφωσης της βιβλιοθήκης.

9.1.3 Μορφή Αρχείων

Η βιβλιοθήκη προτείνεται να εγκατασταθεί στον κατάλογο /usr/local/lums. Κάτω από τον κατάλογο αυτό περιλαμβάνονται δύο κατάλογοι:

- Ο κατάλογος config ο οποίος περιέχει το αρχείο config.php το οποίο και είναι το αρχείο διαμόρφωσης της βιβλιοθήκης
- Ο κατάλογος lib ο οποίος περιέχει τον κώδικα της βιβλιοθήκης:
    - Το αρχείο main.php είναι το κύριο αρχείο το οποίο πρέπει να γίνεται include για τη χρήση της βιβλιοθήκης
    - Το αρχείο functions.php το οποίο είναι βοηθητικό αρχείο της βιβλιοθήκης
    - Το αρχείο extras.php στο οποίο μπορούν να προστεθούν οι βοηθητικές συναρτήσεις που ορίζονται από το χρήστη της βιβλιοθήκης. Περισσότερες πληροφορίες για τις συναρτήσεις αυτές περιέχονται παρακάτω στο παρόν κείμενο.

9.1.4 Αρχείο Διαμόρφωσης

Το αρχείο διαμόρφωσης επιτρέπει τον ορισμό όλων λειτουργιών που περιγράφηκαν παραπάνω όσο και άλλων επιπλέον. Το αρχείο είναι σε μορφή μεταβλητών PHP ώστε να είναι εύκολη και άμεση η ανάγνωση του και χωρίζεται σε τρία μέρη.

Στο πρώτο ορίζονται ορισμένα βασικά στοιχεία για τη βιβλιοθήκη. Το charset που θα χρησιμοποιηθεί για τιμές που δεν είναι σε αγγλικό αλφάβητο, ο τύπος της κρυπτογράφησης που χρησιμοποιείται για τα passwords και το attribute το οποίο ορίζει τον τύπο των αντικειμένων (είσοδος $L_object_type στις συναρτήσεις):

```
$LUMS_Config[Main][objecttype_attribute] = 'edupersonorgunitdn';
$LUMS_Config[Main][encryption_scheme] = 'crypt';
$LUMS_Config[Main][interfaceid] = '1';
$LUMS_Config[Main][config_format_version] = '1.1';
$LUMS_Config[Main][non_english_charset] = 'iso-8859-7';
$LUMS_Config[Main][debug] = 0;
```

Στο δεύτερο μέρος ορίζονται τα στοιχεία για τον εξυπηρετητή καταλόγου που θα χρησιμοποιηθεί όπως το hostname, το Bind DN και Password, το base και το DN κάτω από το οποίο θα διατηρούνται τυχόν αντικείμενα counters για την υποστήριξη autoincrement μεταβλητών:

```
$LUMS_Config[LDAP][server] = 'ldap://localhost';
$LUMS_Config[LDAP][binddn] = '';
$LUMS_Config[LDAP][bindpassword] = '';
$LUMS_Config[LDAP][base] = 'ou=people,o=papei,c=gr';
$LUMS_Config[LDAP][countersdn] = 'ou=sequences,ou=config,dc=company,dc=com';
```

Στο τρίτο μέρος γίνεται ο ορισμός των αντικειμένων των εγγραφών. Οι εγγραφές χωρίζονται ανά τύπο αντικειμένου και στη συνέχεια γίνεται ο ορισμός των ακόλουθων στοιχείων:

- Λειτουργίες (Operations) οι οποίες μπορούν να εκτελεστούν πριν και μετά την πραγματοποίηση των αλλαγών στις εγγραφές. Οι λειτουργίες αυτές ορίζονται με την

μορφή συναρτήσεων και αναφέρονται στις περιπτώσεις προσθήκης (add), ενημέρωσης (modify), μετονομασίας (rename) και διαγραφής (delete) εγγραφών.

- Mappings που χρησιμοποιούνται στην περίπτωση των mapped attributes (περιγράφονται παρακάτω στο κείμενο).
- Attributes εγγραφής και στοιχείων που αφορούν αυτά. Τα στοιχεία αυτά αναλύονται στη συνέχεια.

Παράδειγμα ορισμού λειτουργιών:

```
#$LUMS_Config[Object][parent][operations][preadd] = 'LUMS_operations_parent_preadd';
#$LUMS_Config[Object][parent][operations][postadd] = 'LUMS_operations_parent_postadd';
#$LUMS_Config[Object][parent][operations][premodify] =
'LUMS_operations_parent_premodify';
#$LUMS_Config[Object][parent][operations][postmodify] =
'LUMS_operations_parent_postmodify';
#$LUMS_Config[Object][parent][operations][prerename] =
'LUMS_operations_parent_prerename';
#$LUMS_Config[Object][parent][operations][postrename] =
'LUMS_operations_parent_postrename';
#$LUMS_Config[Object][parent][operations][predelete] =
'LUMS_operations_parent_predelete';
#$LUMS_Config[Object][parent][operations][postdelete] =
'LUMS_operations_parent_postdelete';
```

Τα mappings ορίζονται ως εξής:

[mappings][<index attribute name>][<mapped attribute name>][<index attribute value>] = '<mapped attribute value>'

Παράδειγμα ορισμού mappings:

```
$LUMS_Config[Object][parent][mappings][indexattr][businesscategory][a] = 'Undergraduate Student';
$LUMS_Config[Object][parent][mappings][indexattr][businesscategory][b] = 'Administrative Personnel';
$LUMS_Config[Object][parent][mappings][indexattr][edupersonaffiliation][a] = 'student';
$LUMS_Config[Object][parent][mappings][indexattr][edupersonaffiliation][b] = 'employee';
```

Για τον ορισμό των attributes ακολουθείται η ακόλουθη λογική:

$LUMS_Config[Object][<object type name>][<attribute name>][<directive>]

Δυνατές επιλογές για την τιμή directive:

- Required = 1 or 0: Ορίζει αν το attribute είναι απαιτούμενο (και πρέπει να υπάρχει στην είσοδο κατά τη δημιουργία της εγγραφής)
- Multivalue = 1 or 0
- : Ορίζει αν το attribute είναι πλειότιμο ή όχι
- Type = 'string|dn|binary|mail|telephonenumber|password': Ορίζει τον τύπο του attribute
- Checktypefunction = <function name>: Ορίζει επιπλέον συνάρτηση ελέγχου της τιμής του attribute. Η συνάρτηση πρέπει να έχει οριστεί στο αρχείο lib/extras.php και επιστρέφει 0 σε περίπτωση αποτυχίας ή θετική τιμή εάν η τιμή είναι επιτρεπόμενη.

- Valuetype = 'constant|uservalue|callfunc|autoincrement|virtual|mapping|copyattr': Ορίζει τον τύπο της τιμής για το attribute:
    - Constant: Η τιμή είναι σταθερή και ορίζεται στο ίδιο το αρχείο διαμόρφωσης.
    - Uservalue: Η τιμή εισάγεται ως είσοδος κατά την κλήση των συναρτήσεων
    - Callfunc: Η τιμή λαμβάνεται από την έξοδο κατάλληλης συνάρτησης
    - Autoincrement: Η τιμή προκύπτει από αυτόματη αυξανόμενη αρίθμηση την οποία πραγματοποιεί η ίδια η βιβλιοθήκη.
    - Virtual: Η τιμή είναι εικονική. Χρησιμοποιείται μόνο ως δείκτης για τον προσδιορισμό τιμών στα mappings.
    - Mapping: Η τιμή δημιουργείται μέσω κατάλληλου πίνακα. Η τιμή ενός virtual attribute χρησιμοποιείται ως δείκτης για την επιλογή της τιμής από τον πίνακα
    - Copyattr: Η τιμή προκύπτει απλά με την αντιγραφή της τιμής άλλου attribute.

Η σειρά του evaluation για τις παραπάνω κατηγορίες είναι:

1. constant
2. uservalue
3. callfunc (η συνάρτηση μπορεί να πραγματοποιήσει τα πάντα οπότε θεωρούμε ότι μπορεί να περιλάβει και αντιγραφή της λειτουργίας του mapping και copyattr)
4. mapping
5. copyattr (ώστε να είναι διαθέσιμα όλα τα attributes τα οποία μπορεί να αντιγραφούν).

9.1.5 Constant

Στην περίπτωση των σταθερών τιμών μπορούν να δωθούν με την μορφή:

[constant][values][<index>] = <value>

Όπου index είναι νούμερο στοιχείου πίνακα (αρίθμηση ξεκινάει από το 0). Κατ' αυτόν τον τρόπο είναι δυνατόν να δωθούν πολλαπλές τιμές για την περίπτωση πλειότιμων attributes.

9.1.6 Callfunc

Στην περίπτωση αυτή ορίζεται η συνάρτηση προς κλήση με την μορφή:

[callfunction] = <function name>

9.1.7 Autoincrement

Στην περίπτωση των counters αυτοί υλοποιούνται από την ίδια τη βιβλιοθήκη με τη βοήθεια κατάλληλων εγγραφών στην υπηρεσία καταλόγου. Για το λόγο αυτό ορίστηκε αρχικά το countersdn στο αρχείο διαμόρφωσης. Για κάθε counter δημιουργείται διαφορετική εγγραφή κάτω από το countersdn ενώ στη διαμόρφωση πρέπει να οριστούν τα εξής:

9.1.8 Unique

Για κάθε attribute είναι δυνατόν να οριστεί ότι πρέπει να είναι μοναδικό για ένα συγκεκριμένο υποδέντρο. Ο ορισμός γίνεται ως εξής:

[constraint_unique] = '1'

[unique][base] = "<base>" όπου <base> είναι το υποδέντρο κάτω από το οποίο θα πρέπει να λαμβάνει μοναδικές τιμές το attribute.

9.1.9 Mapping

Για τις περιπτώσεις των mapped attributes απαιτείται να οριστεί το attribute το οποίο θα χρησιμοποιηθεί ως δείκτης σε πίνακα για την εξαγωγή τιμών. Οι τιμές αυτές είναι οι τελικές που θα λάβει το mapped attribute:

[mapping][indexattribute] = '<index attribute name>'

9.2 Παράδειγμα Αρχείου Διαμόρφωσης

Στο Παράρτημα ΙΙ περιέχεται αναλυτικό παράδειγμα πλήρους αρχείου διαμόρφωσης για την υπηρεσία.

9.2.1 Περιορισμοί Τιμών

Είναι δυνατό να οριστούν περιορισμοί στις τιμές που μπορεί να λάβει ένα attribute. Σε αυτή τη φάση οι περιορισμοί είναι ένας πίνακας από δυνατές τιμές. Η διαμόρφωση γίνεται ως εξής:

[constraint][type] = 'arrayOfValues'

[constraint][values][#] = '<val>'

Όπου # είναι array index αριθμός (ξεκινά από το 0) και <val> είναι η δυνατή τιμή του attribute.

9.2.2 Συναρτήσεις που χρησιμοποιούνται

Οι παρακάτω βοηθητικές συναρτήσεις μπορούν να οριστούν από το χρήστη και προστίθονται στο αρχείο lib/extras.php όπως περιγράφηκε στη δομή της βιβλιοθήκης. Οι τύποι των διαθέσιμων συναρτήσεων είναι οι ακόλουθοι:

- **Οι συναρτήσεις ελέγχου τύπου:**

function LUMS_checktypefun_<attribute name>($attribute_values)

Το $attribute_values είναι μορφής πίνακα αν το attribute είναι πλειότιμο. Η συνάρτηση επιστρέφει 1 για επιτυχία και 0 για αποτυχία ελέγχου

- **Οι συναρτήσεις callfunc:**

function LUMS_callfun_<name>_<attribute>($info)

Το $info είναι $entry_info για τις συναρτήσεις προσθήκης εγγραφών και $change_info για τις συναρτήσεις αλλαγής εγγραφών. Η συνάρτηση επιστρέφει πίνακα. Το πρώτο στοιχείο είναι

ένα από τα 'notset', 'empty', 'ok', 'error' notset επιστρέφεται αν τα attributes που είναι απαραίτητα δεν είναι διαθέσιμα, empty αν είναι διαθέσιμα αλλά οι τιμές τους είναι κενές, error αν υπήρξε σφάλμα (στην περίπτωση αυτή το δεύτερο στοιχείο είναι περιγραφή του σφάλματος) και ok αν όλα ήταν επιτυχή. Το δεύτερο στοιχείο είναι η επιστρεφόμενη τιμή.

- **Οι συναρτήσεις auto increment:**

function LUMS_incrementfun_<name>($LDAP_Conn, $CountersDN, $CounterName, $InterfaceID)

Η ήδη διαθέσιμη συνάρτηση LUMS_incrementfun_incrementbyone είναι ήδη διαθέσιμη και επιστρέφει την αμέσως επόμενη τιμή. Το $InterfaceID είναι διαθέσιμο στη διαμόρφωση του LUMS και πρέπει να είναι μοναδικό ανά interface. Χρησιμοποιείται για να παράγει μοναδικές τιμές ακόμα και αν υπάρχουν πολλαπλές εγκαταστάσεις του LUMS. Η εγγραφή του counter έχει την μορφή:

dn: cn=$CounterName, $CountersDN

objectclass: top

objectclass: person

cn: $CounterName

sn: <value increment by one>.$InterfaceID

10 Έλεγχοι λειτουργίας συστήματος

10.1 Κεντρική υπηρεσία καταλόγου

Για την διασφάλιση της ομαλής λειτουργίας και αποδοχής της κεντρικής υπηρεσίας LDAP προτείνονται οι παρακάτω διαδικασίες.

- Έλεγχος συνδεσιμότητας. Αποστολή ενός ICMP echo request πακέτου με σκοπό τον έλεγχο της δικτυακής ύπαρξης του υλικού εξοπλισμού. Εάν αυτός ο έλεγχος αποτύχει σημαίνει πως ο εξοπλισμός και κατά συνέπεια και το λογισμικό βρίσκονται εκτός δικτύου.
- Έλεγχος διεργασίας. Μέσω των εργαλείων του λειτουργικού συστήματος που φιλοξενεί το λογισμικό LDAP θα πρέπει να ελεγχθεί η ύπαρξη της διεργασίας του εξυπηρετητή LDAP.
- Επιτυχής σύνδεση. Το λογισμικό εξυπηρετητή LDAP επιτρέπει την σύνδεση από εξωτερικούς φορείς στις TCP πόρτες 389 και 636 . Μέσω αυτού του μηχανισμού διεξάγεται κάθε επικοινωνία του λογισμικού με τις εφαρμογές που πρέπει να έχουν πρόσβαση στην υπηρεσία. Σύνδεση με αυτόν τον τρόπο στην υπηρεσία επιβεβαιώνει ότι η υπηρεσία είναι προσβάσιμη τουλάχιστον στο επίπεδο μεταφοράς δεδομένων.
- Εκτέλεση λειτουργιών. Η συμπεριφορά του λογισμικού σε επίπεδο εφαρμογής, δηλαδή η ανταπόκριση βασικές λειτουργίες που πρέπει να επιτυγχάνει, ήτοι αναζήτηση, προσθήκη, μεταβολή, διαγραφή εγγραφών, θα πρέπει να ελεγχθεί. Για αυτόν τον έλεγχο αρκεί ένας μηχανισμός που θα αναζητά, προσθέτει, μεταβάλλει και διαγράφει συγκεκριμένες εγγραφές.
- Εκτέλεση αλλαγής συνθηματικού. Ειδική υποπερίπτωση των παραπάνω ελέγχων είναι η αλλαγή συνθηματικού σε τουλάχιστον μία εγγραφή. Αυτός ο έλεγχος είναι απαραίτητος και χρήζει ειδικής μνείας αν και είναι δυνατόν να ενσωματωθεί στον παραπάνω μηχανισμό.
- Έλεγχος των αρχείων καταγραφής. Η παρακολούθηση των αρχείων καταγραφής των εξυπηρετητών μέσω της γραμμής εντολών επιτρέπει να πιστοποιηθεί με τον πλέον σίγουρο τρόπο η εύρυθμη λειτουργία της υπηρεσίας..

- Επιτυχής επανεκκίνηση της υπηρεσίας. Η εφαρμογή θα πρέπει να επανεκκινηθεί τουλάχιστον μία φορά μέσω της γραμμής εντολών και των εργαλείων που προσφέρει για αυτή την περίπτωση το πακέτο λογισμικού.
- Επιτυχής απόκτηση αντιγράφου ασφαλείας. Θα πρέπει να πραγματοποιηθεί η επιτυχημένη απόκτηση αντιγράφου ασφαλείας και η ορθότητα των περιεχομένων του. Αυτή η διαδικασία δεν πρέπει να συγχέεται με υπάρχουσα υποδομή αντιγράφων ασφαλείας και θα πρέπει να εκτελεστεί χειροκίνητα από τους διαχειριστές μέσω της γραμμής εντολών.
- Επιτυχής ανάκτηση δεδομένων από αντίγραφο ασφαλείας. Θα πρέπει να πραγματοποιηθεί επιτυχής ανάκτηση των δεδομένων από προηγούμενο αντίγραφο ασφαλείας. Η υπηρεσία μετά την ανάκτηση δεδομένων θα πρέπει να ικανοποιήσει τους ελέγχους που προδιαγράφηκαν προηγουμένως και να περιέχει πλήρη και ορθά δεδομένα, όμοια με αυτά που λήφθηκαν κατά τη διαδικασία απόκτησης αντίγραφου ασφαλείας.

10.2 Υπηρεσίες ιστού

Στην περίπτωση των υπηρεσιών ιστού που θα διατίθενται στις εξωτερικές υπηρεσίες απαιτείται ο έλεγχος των εξής στοιχείων:

- **Έλεγχος πρόσβασης μέσω Web (HTTP).** Ο έλεγχος αυτός μπορεί να πραγματοποιηθεί πολύ απλά με τη δοκιμαστική πρόσβαση στη σελίδα web που αντιστοιχεί στο κάθε web service

- **Έλεγχος αρχείου WSDL.** Για την ανάπτυξη και ολοκλήρωση της διαλειτουργικότητας μέσω υπηρεσιών ιστού είναι απαραίτητο να είναι διαθέσιμος ο ορισμός των λειτουργιών σε μορφή WSDL. Επιπλέον ο ορισμός αυτός θα πρέπει να είναι συντακτικά σωστός και πλήρης. Η πρόσβαση στο WSDL αρχείο κάθε web service μπορεί να γίνει απευθείας μέσω web με την κλήση της σελίδας web με την προσθήκη του argument *?wsdl* στο URL. Εν συνεχεία το αρχείο μπορεί να ελεγχθεί αν είναι συντακτικά και λειτουργικά σωστό και πλήρες.

- **Έλεγχος λειτουργικότητας web service.** Το πλέον βασικό στάδιο ελέγχου είναι η επιβεβαίωση της καλής λειτουργίας του ίδιου του web service. Για το σκοπό αυτό θα δημιουργηθεί δοκιμαστική σελίδα για κάθε web service που θα αναλάβει να εκτελέσει κάθε προβλεπόμενη λειτουργία με κατάλληλο δοκιμαστικό input. Η σελίδα αυτή θα πρέπει να πραγματοποιεί HTTP Authentication με τα προβλεπόμενα στοιχεία χρήστη που θα χρησιμοποιηθούν στην εν λειτουργία χρήση των υπηρεσιών ιστού. Ο έλεγχος θα είναι επιτυχής εφόσον:
  - Η σελίδα επιστρέφει επιτυχώς για κάθε προβλεπόμενη λειτουργία.
  - Το web service έχει καταγράψει επιτυχώς και πλήρως όλες τις εκτελεσμένες λειτουργίες στο προβλεπόμενο log file.
  - Ο έλεγχος των δεδομένων της υπηρεσίας καταλόγου επιβεβαιώνει την επιτυχή εκτέλεση των αντίστοιχων λειτουργιών.
  - Η λειτουργία ολοκληρώνεται σε εύλογο χρονικό διάστημα μερικών δευτερολέπτων.

  Καθώς οι λειτουργίες που προβλέπεται να υποστηρίξουν οι υπηρεσίες ιστού δεν προβλέπεται να είναι μαζικές αλλά συγκεκριμένες και χρονικά περιορισμένες (μικρός αριθμός αλλαγών εγγραφών χρηστών/μέρα) δεν κρίνεται σκόπιμη η δοκιμή υψηλού φόρτου των υπηρεσιών ιστού. Εφόσον η ολοκλήρωση κάθε λειτουργίας γίνεται σε επαρκώς μικρό χρονικό διάστημα (επιθυμητά λίγα δευτερόλεπτα) δε θεωρείται ότι σε

συνθήκες πραγματικής λειτουργίας των υπηρεσιών ιστού θα παρουσιάσουν πρόβλημα υποστήριξης του προβλεπόμενου φόρτου εργασίας.

10.3 Επικοινωνία με εφαρμογή διαχείρισης χρηστών

Ο έλεγχος της επικοινωνίας περιέχει 3 στάδια:

- Έλεγχο της βασικής σύνδεσης με την υπηρεσία καταλόγου από τον εξυπηρετητή στον οποίο είναι εγκατεστημένη η εφαρμογή διαχείρισης χρηστών και πραγματοποίηση κατάλληλων αναζητήσεων παρόμοιων με αυτές που θα πραγματοποιηθούν σε λειτουργία.
- Εκτέλεση των ελέγχων υπηρεσιών web για τη λειτουργία των υπηρεσιών ιστού.
- Δοκιμαστική εκτέλεση των σελίδων διαχείρισης με κατάλληλα δοκιμαστικά στοιχεία. Η δοκιμή θεωρείται επιτυχής εφόσον:
    - Η σελίδα επιστρέφει επιτυχώς για κάθε προβλεπόμενη λειτουργία.
    - Το web service (όπου υπάρχει) έχει καταγράψει επιτυχώς και πλήρως όλες τις εκτελεσμένες λειτουργίες στο προβλεπόμενο log file.
    - Ο έλεγχος των δεδομένων της υπηρεσίας καταλόγου επιβεβαιώνει την επιτυχή εκτέλεση των αντίστοιχων λειτουργιών (στις περιπτώσεις αλλαγών).
    - Η σελίδα επιστρέφει τα αναμενόμενα στοιχεία στις περιπτώσεις ανάγνωσης.
    - Η λειτουργία ολοκληρώνεται σε εύλογο χρονικό διάστημα μερικών δευτερολέπτων.

Στο ΠΑΡΑΡΤΗΜΑ ΙΙΙ παρουσιάζεται το αρχείο που δημιουργήθηκε για την επικοινωνία και τον έλεγχο των δεδομένων της υπηρεσίας καταλόγου.

11 Εγκατάσταση

Σε ένα εγκατεστημένο instance του fedora core 11 – i386 δίνουμε προσοχή πως αυτά που χρειαζόμαστε δεν υπάρχουν και θα πρέπει να την εμπλουτίσουμε με εγκατάσταση των απαραίτητων αρχείων για να τρέξει η εφαρμογή μας.

Αρχικά ανοίγουμε το Terminal και δίνουμε τις παρακάτω εντολές:

“su" και τον κωδικό του Super User που είναι το “allaxeme”

11.1 Εγκαθιστούμε και εκκινούμε τον Apache (httpd) στο Linux

Ο Apache HTTP Server είναι ένα δωρεάν πρόγραμμα/ανοικτού κώδικα web server για το Fedora και για τα Unix-like συστήματα.

Δίνουμε την παρακάτω εντολή:

```
# yum install httpd
```

Εκκινούμε τον Apache:

```
# chkconfig httpd on

# /etc/init.d/httpd start
```

11.2 Εγκαθιστούμε την PHP (Hypertext Preprocesso)

Η PHP είναι μια server-side γλώσσα προγραμματισμού Ιστού που μπορεί να ενσωματωθεί σε σελίδες HTML. Όταν ένας χρήστης αποκτήσει πρόσβαση σε μια σελίδα βασισμένη σε PHP,η PHP δημιουργεί δυναμικά μια ιστοσελίδα που μετά περνάει στο πρόγραμμα περιήγησης.

Η PHP δουλεύει με Apache, Lighttpd και άλλους webservers. Η PHP προσφέρει ενσωματωμένη βάση δεδομένων για την ολοκλήρωση πολλών εμπορικών και μη συστημάτων διαχείρισης βάσεων δεδομένων. Έχει επίσης τη δυνατότητα να εκτελεί πολλά χρήσιμα Web-tasks χρησιμοποιώντας ένα μεγάλο σύνολο ενσωματωμένων λειτουργιών.

Η PHP τρέχει σε γενικές γραμμές σε ένα web server, λαμβάνοντας PHP κώδικα ως είσοδο της και δημιουργεί ιστοσελίδες, αλλά και εντολές δέσμης ενεργειών γραμμής (command-line scripting) και client-side GUI εφαρμογές αποτελούν μέρος από τις τρεις κύριες χρήσης της PHP. Η PHP μπορεί να αναπτυχθεί σε οποιοδήποτε web server και σε σχεδόν κάθε πλατφόρμα OS δωρεάν. Η εγκατάσταση της PHP είναι εύκολη και γίνεται με τις παρακάτω εντολές:

```
# yum install php
```

Και για να αναβαθμίσουμε την php όταν χρειάζεται:

```
# yum update php
```

11.3 Εγκαθιστούμε την PECL και την APC

Η PECL είναι μια αποθήκη για PHP Extensions, παρέχοντας έναν κατάλογο όλων των γνωστών επεκτάσεων για τη λήψη και την ανάπτυξη PHP επεκτάσεων.

Η Alternative PHP Cache (APC) είναι μια ελεύθερη και ανοιχτή opcode cache για την PHP. Στόχος της είναι να παρέχει ένα ελεύθερο, ανοικτό, και ισχυρό πλαίσιο για την προσωρινή αποθήκευση και τη βελτιστοποίηση του ενδιάμεσου κώδικα που παράγεται από την PHP.

```
# yum install php-pecl-apc
```

11.4 Εγκαθιστούμε τον openldap server και ενεργοποιούμε την php για χρήση στον ldap:

```
# yum install openldap-servers

# yum install php-ldap
```

Μπορούσαμε να είχαμε εγκαταστήσει και τον «openldap-client», αλλά το πακέτο του πελάτη δεν χρειάζεται στον server. Παρεμπιπτόντως το πακέτο «nss_ldap», εγκαθίσταται από μόνο του, το οποίο περιέχει το «libnss_ldap» και το «pam_ldap», τα οποία χρειάζονται για έναν πελάτη. Το «pam_ldap» θα βοηθήσει στην ολοκλήρωση του LDAP με το email, SSH, FTP, Samba, όπου στην συγκεκριμένη περίπτωση δεν είναι απαραίτητο για την ανάπτυξη της εφαρμογής, αλλά για μελλοντική επέκτασή της.

11.4.1 Παραμετροποίηση του LDAP

Η αρχή κάθε παραμετροποίησης μιας βάσης δεδομένων είναι η δημιουργία ενός κωδικού ασφαλείας για τον root χρήστη του LDAP server. Συγκεκριμένα με το «slappasswd» μπορούμε να αντικαταστήσουμε τον κωδικό με έναν κωδικοποιημένο με SHA. Δημιουργούμε πρώτα τον χρήστη και μετά προσθέτουμε κωδικό ασφαλείας και στον LDAP χρήστη.

Οι εντολές είναι οι παρακάτω:

```
Create a root Password:
  slappasswd
New password:
Re-enter new password:
 {SHA}b2hqheVCg/H6xXArKpwnlR3eelg=
```

Για πιο σωστή αντιμετώπιση της βάσης δεδομένων που φτιάξαμε καλό θα ήταν να περάσουμε τον κωδικό για τον root χρήστη στο αρχείο slapd.config.

Έτσι αντιγράφουμε το αποτέλεσμα του slappasswd μέσα στο /etc/openldap/slapd.conf με τις παρακάτω εντολές:

```
# rootdn directive for specifying a superuser on the database. This is needed
# for syncrepl.
```

```
rootdn          "cn=manager,dc=vkolyvas,dc=gr"
rootpw          {SHA}b2hqheVCg/H6xXArKpwnlR3eelg=
```

Πριν εκκινήσουμε τον LDAP πρέπει να δηλώσουμε τον τύπο της βάσης #1, το suffix του domain μας, να δηλώσουμε το rootdn,το rootdn password και το που έχουμε εγκαταστήσει τον ldap (που βρίσκονται τα αρχεία μας).

Κάνουμε τις απαραίτητες αλλαγές στο slapd.conf εκτελώντας τα παρακάτω:

```
vi /etc/openldap/slapd.conf
database        bdb
suffix          " dc=vkolyvas,dc=gr"
rootdn          "cn=manager, dc=vkolyvas,dc=gr"
rootpw          {SHA}b2hqheVCg/H6xXArKpwnlR3eelg=
directory       "/var/lib/ldap"
```

πριν προσθέσουμε το «init.ltif» χρειάζεται να διαγράψουμε τους παλιούς καταλόγους (εάν είχαμε παλιές εγγραφές), τώρα είναι όμως νέα, καθαρή εγκατάσταση και δεν χρειάζεται η παρακάτω εντολή:

```
# rm -rf /var/lib/ldap/*
```

Επειδή πρόκειται να τρέξουμε μια υπηρεσία με πάρα πολλές εγγραφές χρηστών καλό θα ήταν να βελτιώσουμε τις παραμέτρους της βάσης , προσθέτοντας το "cachesize" και μερικά ακόμα "indexes". Επίσης εάν τα δεδομένα μας είναι μεγάλης σημασίας καλό θα είναι να τρέχουμε πριν την εκκίνηση του Ldap την εντολή "slapindex".

Δημιουργούμε ένα αρχείο «/var/lib/ldap/DB_CONFIG» με τις παρακάτω παραμέτρους:

```
set_cachesize 0 15000000 1
set_lg_regionmax 262144
set_lg_bsize 2097152
set_flags DB_LOG_AUTOREMOVE
```

11.5 Τώρα εκκινούμε τον LDAP

```
service ldap start
```

12 ΠΑΡΑΡΤΗΜΑ Ι

Εντολές στον vi

Παρακάτω παραθέτω μερικές εντολές που χρησιμοποίησα στον vi.

Σημειώνοντας θα ήθελα να παρατηρήσω πως όπως και στο linux οι εντολές που δίνουμε στον editor είναι Case Sensitive.

Για να κατευθύνουμε τον κέρσορα χρησιμοποιούμε τα βελάκια (σταυρό)

ctrl-F Μετακινούμε τον κέρσορα μπροστά 1 οθόνη.
ctrl-B Μετακινούμε τον κέρσορα πίσω 1 οθόνη.
$ Μετακινούμε τον κέρσορα στο τέλος της γραμμής.
^ Μετακινούμε τον κέρσορα στην αρχή της γραμμής.
:1 Μετακινούμε τον κέρσορα στην πρώτη γραμμή του αρχείου.
:$ Μετακινούμε τον κέρσορα στην τελευταία γραμμή του αρχείου.
/ Ψάχνουμε προς το τέλος μια σειρά από χαρακτήρες (character string)
? Ψάχνουμε προς τα πάνω μια σειρά από χαρακτήρες (character string)
x Αποκόβουμε τον χαρακτήρα στην τοποθεσία που έχουμε τον κέρσορα
dd Αποκόβουμε όλη την γραμμή που βρίσκεται ο κέρσορας.
p Επικολούμε τα δεδομένα που αποκόψαμε από τις 2 παραπάνω εντολές
u Undo.

Για να γράψουμε μέσα στο αρχείο (input mode)

a Προσθέτουμε χαρακτήρες (κείμενο) μετά τον κέρσορα.
i Προσθέτουμε χαρακτήρες (κείμενο) πριν τον κέρσορα.
R Αντικαθιστούμε χαρακτήρες (κείμενο) στον κέρσορα.
O Εισαγάγουμε μια γραμμή κάτω από τον κέρσορα.

Για να δώσουμε εντολές (Command mode)

esc Αλλάζουμε από το Input mode στο Command mode.

Για να σώσουμε και να βγούμε από το αρχείο

:w Για να γράψουμε στο αρχείο δίχως να βγούμε από τον editor
ZZ Σώζουμε το αρχείο και βγαίνουμε.
:q! Βγαίνουμε από τον editor δίχως να σώσουμε.

Terminal

Για να εργαστούμε στη γραμμή εντολών δεν είναι τόσο δύσκολο έργο και παράλληλα δεν υπάρχει ειδική γνώση που απαιτείται. Τα περισσότερα πράγματα στο Linux μπορεί να γίνουν χρησιμοποιώντας την γραμμή εντολών. Αν και υπάρχουν γραφικά εργαλεία για τα περισσότερα προγράμματα, μερικές φορές απλώς δεν αρκούν. Σε αυτό το σημείο η γραμμή εντολών χρησιμοποιείται.

Το terminal ονομάζεται συχνά η γραμμή εντολών (command prompt) ή το κέλυφος (Cell). Παλαιότερα, αυτός ήταν ο τρόπος που επικοινωνούσε ο χρήστης με τον υπολογιστή. Ωστόσο, οι χρήστες του Linux έχουν διαπιστώσει ότι η χρήση του terminal μπορεί να είναι πιο γρήγορη από μια γραφική μέθοδο και εξακολουθεί να κατέχει μέχρι και σήμερα σημαντική παρουσία στις server εφαρμογές.

Η κονσόλα εκκινείτε από το K-menu → System → Konsole (Terminal Program).

Βασικές Εντολές στην Κονσόλα.

- Βλέπουμε τα αρχεία και τους φακέλους : - ls (LiSt)
- Δημιουργεί καταλόγους : - mkdir (directory name)
- Αλλάζει κατάλογο : - cd (/directory/location)
- Αντιγράφει αρχεία και καταλόγους : - cp (file or directory name) (to directory or filename)
- Διαγράφει αρχεία και καταλόγους : - rm (file or directory name)
- Μεταφέρει ή μετονομάζει αρχεία και καταλόγους : - mv (file or directory name)
- Βρίσκει αρχεία/ καταλόγους : - locate (file or directory name)

Μπορούμε να χρησιμοποιούμε σε όλες τις εντολές τα δύο παρακάτω για να βρούμε ένα ή και περισσότερα αρχεία. Το "*" για όλα τα αρχεία ή το "?" για να καθορίσουμε έναν χαρακτήρα.

13 ΠΑΡΑΡΤΗΜΑ ΙΙ

Το αρχείο διαμόρφωσης “config.php” είναι αυτό που χρησιμοποιείται για να οριστούν περιορισμοί στις τιμές που μπορεί να λάβει ένα attribute.

(config.php στο συνοδευτικό υλικό)

14 ΠΑΡΑΡΤΗΜΑ ΙΙΙ

Το αρχείο “testing.webservice.php” είναι αυτό που χρησιμοποιείται για τον έλεγχο της υπηρεσίας.

(testing.webservice.php στο συνοδευτικό υλικό)

15 ΑΚΡΩΝΥΜΙΑ

AAP	Attribute Acceptance Policies
ACL	Access List
ADSL	Asymmetric Digital Subscriber Line
AES	Advanced Encryption Standard
API	Application Programming Interface
ARP	Attribute Release Policies
B2B	Business to Business
BER	Basic Encoding Rules
CA	Certification Authority
CARP	Common Address Redundancy Protocol
CPU	Central Processing Unit
CRL	Certificate Revocation List
DAP	Directory Access Protocol
DEN	Directory Enabled Networking
DER	Distinguished Encoding Rules
DIT	Directory Information Tree
DN	Distinguished Name
DNS	Domain Name System
DoS	Denial of Service
DSML	Directory Services Markup Language
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
ISDN	Integrated Services Digital Network
ISO	International Standards Organisation
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
OS	Operating System
PKI	Public Key Infrastructure
RADIUS	Remote Authentication Dial In User Service
RAM	Random Access Memory
RFC	Request for Comments
SAML	Security Assertion Markup Language
SASL	Simple Authentication and Security Layer
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
SSO	Single Sign On
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDDI	Universal Description, Discovery & Integration
VRRP	Virtual Routing Redundancy Protocol
WS	Web Services
WSDL	Web Service Discription Language
XML	eXtensble Markup Language
AAI	Authentication and Authorization Infrastructure

16 ΑΝΑΦΟΡΕΣ

[1] http://www.openldap.org/

[2] http://sourceforge.net/projects/nusoap/

[3] http://directory.fedora.redhat.com/

[4] http://www.redhat.com/software/rha/directory/

[5] http://www.sun.com/software/products/directory_srvr_ee/

[6] http://www.microsoft.com/

[7] http://www-306.ibm.com/software/tivoli/products/directory-server/

[8] http://www.novell.com/products/edirectory/

[9] http://en.allexperts.com/e/l/li/lightweight_directory_access_protocol.htm

[10] http://www.mozilla.org/mailnews/arch/LDAP_replication2.html

[11] The LDAP Content Synchronization Operation <draft-zeilenga-ldup-sync-05.txt>.

[12] http:///www.go-online.gr

[13] http://www.switch.ch/aai/

[14] http://www.faqs.org/rfcs/rfc4520.html

[15] http://www.faqs.org/rfcs/rfc4521.html

[16] http://www.educause.edu/eduperson/

[17] http://www.freeradius.org/

[18] http://www.utoronto.ca/ns/stats/ldap.html

[19] http://www.bris.ac.uk/is/computing/applications/email/directory/stats.html

[20] Heartbeat: http://www.linux-ha.org/Heartbeat

[21] http://www.w3.org/TR/ws-arch/

[22] http://www.zytrax.com/books/ldap/apa/

[23] http://deliver.pigeonair.net/doc/recipes/postfix/postfix/x98.html