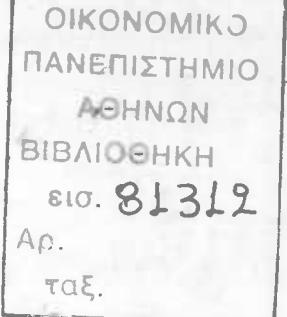




ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



Διπλωματική Εργασία  
Μεταπτυχιακού Διπλώματος Ειδίκευσης

«A PROTOCOL FOR ENTERPRISE NETWORK INTERCONNECTION»

Στεφανίδης Χαράλαμπος

Επιβλέπων: Γεώργιος Πολύζος

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ  
ΚΑΤΑΛΟΓΟΣ



ΑΘΗΝΑ, ΙΟΥΝΙΟΣ 2007



ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ

ΑΓΩΝΗΣ

ΒΙΒΛΙΟΘΗΚΗ

αι. 81319

Αρ.

ταξ.



ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

## ABSTRACT

### A PROTOCOL FOR ENTERPRISE NETWORK INTERCONNECTION

Work that was submitted from  
Charalampo Stefanidi  
as partial achievement of requirements for the acquisition

MASTER OF SCIENCE

Supervisor: George C. Polyzos

*"In this master thesis, a protocol to support communication and collaboration between Enterprise Networks, as defined by TrustCoM project, has been proposed. The main goal was the support of bigger scale Virtual Organizations which uses resources from different ENs, achieving efficient use of the available resources. Moreover, the current protocol has been implemented with use of Web Services, in order to simulate its use under real conditions".*

<http://pages.cs.aueb.gr/~xstefan05/>  
xstefan05@aueb.gr



**ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΠΕΡΙΛΗΨΗ**

**A PROTOCOL FOR ENTERPRISE  
NETWORK INTERCONNECTION**

Εργασία που υποβλήθηκε από τον  
Στεφανίδη Χαράλαμπο  
ως μερική εκπλήρωση των απαιτήσεων  
για την απόκτηση

**ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ**

Επιβλέπων Καθηγητής: Γιώργος Πολάζος

“Στην παρούσα εργασία προτείνεται ένα πρωτόκολλο για την υποστήριξη επικοινωνίας και συνεργασίας μεταξύ Enterprise Networks όπως αυτά έχουν οριστεί στα πλαίσια του TrustCoM project. Βασικός στόχος είναι η δημιουργία εικονικών οργανισμών μεγαλύτερης κλίμακας, οι οποίοι θα χρησιμοποιούν πόρους από διαφορετικά EN επιτυγχάνοντας αποδοτικότερη χρήση των διαθέσιμων πόρων. Επιπλέον υλοποιήθηκε το παραπάνω πρωτόκολλο με χρήση Web Services έχοντας ως στόχο την προσομοίωση χρήσης του σε πραγματικές συνθήκες”.

<http://pages.cs.aueb.gr/~xstefan05/>  
[xstefan05@aueb.gr](mailto:xstefan05@aueb.gr)



## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΙΣΑΓΩΓΗ.....	1
ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ ΔΙΑΔΙΚΤΥΑΚΩΝ ΥΠΗΡΕΣΙΩΝ .....	3
ΟΡΙΣΜΟΣ ΔΙΑΔΙΚΤΥΑΚΩΝ ΥΠΗΡΕΣΙΩΝ .....	4
ΠΡΟΤΥΠΑ ΔΙΑΔΙΚΤΥΑΚΩΝ ΥΠΗΡΕΣΙΩΝ .....	9
XML.....	9
SOAP .....	13
WSDL .....	16
UDDI.....	17
ENTERPRISE NETWORK INTERCONNECTION.....	26
EN INTERCONNECTION.....	26
ΟΡΙΣΜΟΣ.....	26
ΣΤΟΧΟΙ.....	26
ΔΙΑΦΟΡΑ ΠΡΩΤΟΚΟΛΛΟΥ ΑΠΟ ΆΛΛΑ .....	28
ΘΕΤΙΚΑ ΚΑΙ ΑΡΝΗΤΙΚΑ.....	30
ΑΡΧΙΤΕΚΤΟΝΙΚΗ.....	33
VO LIFECYCLE.....	33
IDENTIFICATION .....	34
FORMATION .....	34
OPERATION .....	34
EVOLUTION .....	35
DISSOLUTION & TERMINATION .....	35
SUBSYSTEMS .....	36
SERVICE CONSUMER (SC).....	36
SERVICE PROVIDER (SP).....	36
VO MANAGER .....	37
SLA NEGOTIATOR .....	38
POLICY SUBSYSTEM .....	38
EMS (EXECUTION MANAGEMENT SERVICE) .....	38
BROKER.....	39
SLA MANAGER.....	39
SLA SIGNER.....	39
NOTARY .....	39
EN SLA REPOSITORY .....	39
ΠΡΩΤΟΚΟΛΛΟ .....	40
ΔΙΑΔΙΚΑΣΙΑ .....	40
ΥΛΟΠΟΙΗΣΗ .....	49



ΜΔΕ: “*Επιστήμη Υπολογιστών*”

<b>ΕΙΣΑΓΩΓΗ .....</b>	<b>49</b>
<b>ΑΡΧΙΤΕΚΤΟΝΙΚΗ .....</b>	<b>50</b>
<b>INITIATOR WINDOWS APPLICATION .....</b>	<b>50</b>
<b>NOTARY WEB SERVICE.....</b>	<b>52</b>
<b>EN WEB SERVICE.....</b>	<b>53</b>
<b>XML DOCUMENTS .....</b>	<b>54</b>
<b>DATABASE INTERFACE.....</b>	<b>56</b>
<b>SLA INTERFACE .....</b>	<b>58</b>
<b>RESULT INTERFACE.....</b>	<b>62</b>



## ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΑΡΙΘΜΟΣ ΕΙΚΟΝΑΣ	ΣΕΛΙΔΑ
Σχήμα 1: Web Services .....	4
Σχήμα 2: Βασική λειτουργία WS.....	6
Σχήμα 3: Συνεργασία διαδικτυακών υπηρεσιών.....	8
Σχήμα 4: Τμήμα Xml εγγράφου.....	10
Σχήμα 5: Τυπική Λειτουργία UDDI.....	18
Σχήμα 6: Αφηρημένη μορφή Super EN.....	28
Σχήμα 7: Αφηρημένη μορφή Κοινού UDDI.....	29
Σχήμα 8: Κύκλος ζωής VO.....	33
Σχήμα 9: Αρχιτεκτονική (α) .....	41
Σχήμα 10: Αρχιτεκτονική (β) .....	45
Σχήμα 11: Αρχιτεκτονική (γ) .....	46
Σχήμα 12: Αρχιτεκτονική (δ) .....	47
Σχήμα 13: Αρχιτεκτονική (ε).....	48
Σχήμα 14: Αρχιτεκτονική προγράμματος.....	51
Σχήμα 15: Xml Schema (a).....	54
Σχήμα 16: Xml Schema (b).....	55
Σχήμα 17: Εισαγωγή Προϊόντων στον DB Wizard .....	57
Σχήμα 18: Εισαγωγή κατηγοριών σε προϊόν .....	58
Σχήμα 19: Δημιουργία SLA-Template.....	59
Σχήμα 20: Δημιουργία πολλών SLA-Template με αυτόματο τρόπο .....	61
Σχήμα 21: Διεπαφή αλλαγής ώρας .....	62



ΜΔΕ: “*Επιστήμη Υπολογιστών*”

## ΕΥΧΑΡΙΣΤΙΕΣ

Στα πλαίσια αυτής της εργασίας θα ήθελα να ευχαριστήσω τον επόπτη καθηγητή μου κ. Γεώργιο Πολύζο που μου έδωσε τη δυνατότητα να ασχοληθώ με το θέμα της εργασίας αυτής και με καθοδήγησε σε όλη τη διάρκειά της. Ιδιαίτερα τον ευχαριστώ για τις ώρες συζητήσεων σχετικά με την εργασία και γιατί πάντοτε προσέφερε οποιαδήποτε βοήθεια χρειαζόμουν.

Ο κ. Γιάννης Μαριάς παρείχε σημαντική βοήθεια μέσω συζητήσεων, σχολίων και εύστοχων παρατηρήσεων.

Ένα μεγάλο ευχαριστώ επίσης αξίζει στον φίλο και συνάδερφο Γιώργο Πάπαρη που με καθοδήγησε στην φάση της υλοποίησης, προτείνοντας μου εργαλεία και τεχνολογίες που πίστευε ότι θα με βοηθούσαν.

Πάνω από όλους όμως, θέλω να ευχαριστήσω τους γονείς μου, Παναγιώτη και Μαλαματή, για τη συμπαράσταση και την αμέριστη υποστήριξη που μου παρείχαν καθόλη τη διάρκεια των σπουδών μου.

Αφιερωμένο στην αδερφή μου Κατερίνα.

---

Στεφανίδης Χαράλαμπος



ΜΔΕ: “*Επιστήμη Υπολογιστών*”



ΜΔΕ: “Επιστήμη Υπολογιστών”

## *Kεφάλαιο 1*

### ΕΙΣΑΓΩΓΗ

Λαμβάνοντας υπόψη την οικονομική ανταγωνιστικότητα μιας παγκόσμιας οικονομίας και τη συγκρουόμενη επιθυμία για υψηλή ποιότητα ζωής στην Ευρώπη, συμφωνήθηκε από τους επικεφαλής των κυβερνήσεων στη Λισσαβόνα [1] το 2000, ότι η Ευρώπη θα εισήγαγε μια οικονομία γνώσης παρά μια οικονομία βασισμένη στην κατασκευή ή στην γεωργία. Σε μια οικονομία γνώσης, το ανταγωνιστικό πλεονέκτημα προέρχεται από την ευελιξία των οργανισμών να αποκρίνονται στις ευκαιρίες της αγοράς. Ένας μηχανισμός που διαχειρίζεται αποτελεσματικά μια τέτοια ευελιξία είναι η αυτοματοποίηση της διαχείριση του supply chain για μεγάλους οργανισμούς, ή η παροχή περιβαλλόντων για την υποστήριξη του σχηματισμού εικονικών οργανισμών (VO) μεσαίων και μεγάλων οργανισμών [2][3] ώστε να συγκεντρωθούν επαρκής πόροι για να εκμεταλλευτούν ευκαιρίες όπου κανένας οργανισμός δεν θα μπορούσε από μόνος του.

Ένα τέτοιο περιβάλλον για να υποστηρίξει το σχηματισμό και τη λειτουργία VO, πρέπει να είναι έμπιστο και να παρέχει μια βάση εμπιστοσύνης για τους οργανισμούς που συμμετέχουν. Η εμπιστοσύνη μεταξύ των μελών ενός VO μπορεί να υποστηριχθεί από κάθε έναν που γνωρίζει επακριβώς τις υποχρεώσεις και τις επιδώσεις των υπολοίπων, έτσι ώστε οι επιχειρησιακοί κίνδυνοι να μετριάζονται, και να ελέγχονται. Το TrustCoM project [4] παρέχει την δυνατότητα δημιουργίας δυναμικών εικονικών οργανισμών σε έναν ασφαλές και βασισμένο σε συμβόλαια περιβάλλον.



## ΜΔΕ: “Επιστήμη Υπολογιστών”

Η ζήτηση για πόρους συνεχώς αυξάνεται και ίσως κάποιοι Enterprise Networks να μην είναι σε θέση να καλύψουν τις ανάγκες των πελατών τους λόγο περιορισμένων πόρων. Σύμφωνα με το TrustCoM project, αυτοί οι πελάτες είτε θα έπρεπε να περιμένουν μέχρι να απελευθερωθούν οι απαραίτητοι πόροι, είτε θα έπρεπε να μειώσουν τις απαιτήσεις τους, είτε τέλος θα έπρεπε να απευθυνθούν σε κάποιον άλλο EN. Ακόμη όμως και να γίνονταν όλα αυτά, κάποια εργασία ίσως να μην μπορούσε να εκτελεστεί, αν κανένας EN δεν είχε τόσους πόρους στην διάθεσή του.

Το παραπάνω κενό έρχεται να καλύψει η παρούσα εργασία που έχει ως στόχο την επικοινωνία μεταξύ EN, ώστε να χρησιμοποιηθούν πόροι από διαφορετικούς EN με σκοπό την δημιουργία εικονικών οργανισμών μεγαλύτερης κλίμακας, καθώς και η αποδοτικότερη χρήση των πόρων τους.

Στο δεύτερο κεφάλαιο περιγράφονται βασικές τεχνολογίες πάνω στις οποίες στηρίζεται τόσο η αρχιτεκτονική του TrustCoM project, όσο και η δική μας. Στο τρίτο κεφάλαιο περιγράφεται με περισσότερες λεπτομέρειες το κενό που ήρθε να συμπληρώσει η παρούσα εργασία. Στο τέταρτο κεφάλαιο περιγράφεται η αρχιτεκτονική, τα υπάρχοντα υποσυστήματα του TrustCoM project πάνω στα οποία στηρίχθηκε η εργασία αυτή, καθώς και νέα που προστέθηκαν. Στο πέμπτο κεφάλαιο περιγράφεται ο τρόπος λειτουργίας του πρωτοκόλλου που προτείνουμε. Τέλος στο έκτο κεφάλαιο παρουσιάζουμε ένα πρόγραμμα που υλοποιήθηκε και έχει ως σκοπό την προσομοίωση όλης της διαδικασίας του πρωτοκόλλου. Στο Παράρτημα μπορείτε να βρείτε περισσότερες πληροφορίες για τεχνολογίες, ή έννοιες που αναφέρονται στο παρόν έγγραφο.

## *Kεφάλαιο 2*

### **ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΩΝ ΔΙΑΔΙΚΤΥΑΚΩΝ ΥΠΗΡΕΣΙΩΝ**

Οι διαδικτυακές υπηρεσίες είναι μια σχετικά καινούργια τεχνολογία, η οποία έχει λάβει ευρεία αναγνώριση σαν μια σημαντική υλοποίηση της αρχιτεκτονικής προσανατολισμένη σε υπηρεσίες (Service Oriented Architecture). Αυτό γίνεται γιατί οι διαδικτυακές υπηρεσίες παρέχουν μια κατανεμημένη υπολογιστική προσέγγιση για την ενοποίηση εξαιρετικά ετερογενών εφαρμογών μέσω του Διαδικτύου. Οι προδιαγραφές ηλεκτρονικών υπηρεσιών είναι εντελώς ανεξάρτητες από την γλώσσα προγραμματισμού, το λειτουργικό σύστημα και το υλικό. Η τεχνολογία βασίζεται σε ανοιχτές τεχνολογίες όπως είναι:

- eXtensible Markup Language (XML)
- Simple Object Access Protocol (SOAP)
- Universal Description, Discovery and Integration (UDDI)
- Web Services Description Language (WSDL)

Χρησιμοποιώντας ανοιχτά πρότυπα παρέχει ευρέως διαλειτουργικότητα μεταξύ διαφορετικών λύσεων. Τέτοιες αρχές σκοπεύουν ώστε οι εταιρείες να μπορούν να υλοποιήσουν διαδικτυακές υπηρεσίες χωρίς να έχουν καμία γνώση των καταναλωτών υπηρεσιών και αντίστροφα. Αυτό διευκολύνει την ενοποίηση και επιτρέπει στις επιχειρήσεις να εγκαταστήσουν νέες συνεργασίες εύκολα και δυναμικά.

## ΟΡΙΣΜΟΣ ΔΙΑΔΙΚΤΥΑΚΩΝ ΥΠΗΡΕΣΙΩΝ

Η ομάδα Αρχιτεκτονικής Διαδικτυακών Υπηρεσιών του W3C οργανισμού συμφώνησε στον εξής ορισμό για την Διαδικτυακή υπηρεσία:

«Η Διαδικτυακή Υπηρεσία είναι μια εφαρμογή λογισμικού που προσδιορίζεται από το URI, της οποίας οι διεπαφές και οι προσδέσεις είναι ικανές να περιγραφούν μέσω των XML τεχνολογιών. Μια Διαδικτυακή Υπηρεσία υποστηρίζει απευθείας αλληλεπιδράσεις με άλλους πράκτορες λογισμικού χρησιμοποιώντας μηνύματα που στηρίζονται στην XML και ανταλλάσσονται μέσω των πρωτοκόλλων του Διαδικτύου».

Οι διαδικτυακές υπηρεσίες συνδυάζουν την δύναμη των δύο ευρέως διαδεδομένων τεχνολογιών: της XML (μια καθολική γλώσσας περιγραφής δεδομένων) και του HTTP (ένα πρωτόκολλο μεταφοράς δεδομένων που υποστηρίζεται από όλους τους browsers και τους εξυπηρετητές του Διαδικτύου).



**Σχήμα 1: Web Services**

## ΚΥΡΙΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΔΙΑΔΙΚΤΥΑΚΩΝ ΥΠΗΡΕΣΙΩΝ

Οι διαδικτυακές υπηρεσίες είναι:

- **Αυτό-συγκροτημένες**

Από την πλευρά του πελάτη, δεν χρειάζεται επιπλέον λογισμικό. Μια γλώσσα προγραμματισμού με υποστήριξη σε XML και HTTP, είναι αρκετό για να ξεκινήσετε. Από την πλευρά του εξυπηρετητή, απαιτούνται μόνο ένας εξυπηρετητής Διαδικτύου και μια μηχανή servlet.

- **Αυτό-περιγραφόμενες**

Ούτε ο πελάτης ούτε και ο εξυπηρετητής χρειάζεται να γνωρίζουν οτιδήποτε σχετικά με την δομή και το περιεχόμενο των μηνυμάτων των αιτήσεων και των απαντήσεων. Ο ορισμός ενός μηνύματος ταξιδεύει με το μήνυμα. Δεν απαιτούνται καμιά εξωτερική αποθήκη μεταδεδομένων και εργαλεία παραγωγής κώδικα.

- **Αρθρωτές**

Οι διαδικτυακές πύλες είναι μια τεχνολογία για ανάπτυξη και παροχή πρόσβασης σε επιχειρησιακές λειτουργίες μέσω του Διαδικτύου. Το .NET, το J2EE, το CORBA και άλλα πρότυπα είναι τεχνολογίες για την υλοποίηση τέτοιων διαδικτυακών υπηρεσιών.

- **Δημοσιεύονται, εντοπίζονται και καλούνται μέσω του Διαδικτύου**

Τα πρότυπα που χρειάζεται για να γίνει αυτό είναι τα εξής:

ΜΔΕ: “Επιστήμη Υπολογιστών”

- **Simple Object Access Protocol (SOAP)**

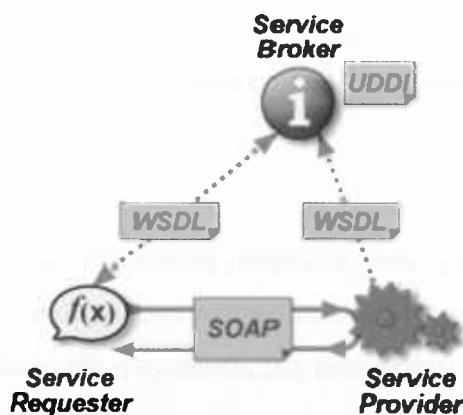
Πρόκειται για ένα πρωτόκολλο μηνυμάτων που είναι γνωστό και σαν αρχιτεκτονικό πρωτόκολλο προσανατολισμένο σε υπηρεσίες.

- **Web Service Description Language (WSDL)**

Πρόκειται για μια περιγραφική διεπαφή και μια γλώσσα σύνδεσης πρωτοκόλλων

- **Universal Description, Discovery, and Integration (UDDI)**

Πρόκειται για έναν μηχανισμό μητρώου το οποίο μπορεί να χρησιμοποιηθεί για την ανεύρεση των περιγραφών των διαδικτυακών υπηρεσιών.



Σχήμα 2: Βασική λειτουργία WS

- **Ανεξάρτητες από γλώσσες προγραμματισμού και διαλειτουργικές**

Η αλληλεπίδραση μεταξύ ενός παρόχου υπηρεσίας και ενός καταναλωτή υπηρεσίας είναι σχεδιασμένη ώστε να είναι εντελώς ανεξάρτητη από πλατφόρμα και γλώσσα προγραμματισμού. Αυτή η αλληλεπίδραση απαιτεί ένα WSDL έγγραφο που ορίζει την διεπαφή και περιγράφει την υπηρεσία, μέσω ενός πρωτοκόλλου δικτύου (που είναι συνήθως το HTTP). Η διαλειτουργικότητα επιτυγχάνεται από το γεγονός ότι ούτε ο πάροχος αλλά ούτε και ο καταναλωτής υπηρεσίας έχουν ιδέα για την πλατφόρμα και την γλώσσα προγραμματισμού που χρησιμοποιεί ο άλλος.

- **Στηρίζονται σε ανοιχτά πρότυπα**

Η XML και το HTTP αποτελούν θεμέλιους λίθους για τις διαδικτυακές υπηρεσίες. Ένα μεγάλο μέρος της τεχνολογίας των διαδικτυακών υπηρεσιών έχει χτιστεί πάνω σε έργα ανοιχτού κώδικα. Για αυτό το λόγο, η διαλειτουργικότητα αποτελεί ρεαλιστικό στόχο.

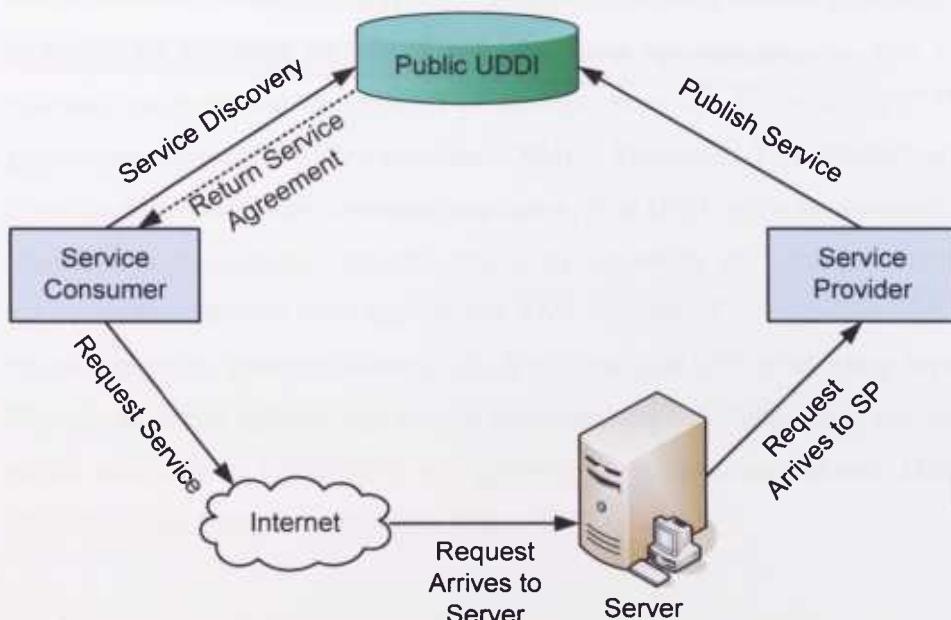
- **Δυναμικές**

Η δυναμική ηλεκτρονική επιχείρηση μπορεί να γίνει πραγματικότητα χρησιμοποιώντας τις διαδικτυακές υπηρεσίες επειδή με το UDDI και την WSDL η περιγραφή και η ανεύρεση των διαδικτυακών υπηρεσιών μπορεί να αυτοματοποιηθεί.

- **Ικανές για περαιτέρω σύνθεση**

Απλές διαδικτυακές υπηρεσίες μπορούν να συναθροιστούν σε περισσότερο πολύπλοκες, είτε χρησιμοποιώντας τεχνικές ροών εργασίας είτε καλώντας χαμηλότερου επιπέδου διαδικτυακές πύλες.

Το *Σχήμα 3* δείχνει μια τυπική συνεργασία Διαδικτυακής υπηρεσίας η οποία βασίζεται στο μοντέλο της αρχιτεκτονικής προσανατολισμένη σε υπηρεσίες όπως φάνηκε και στο προηγούμενο *Σχήμα 2*.



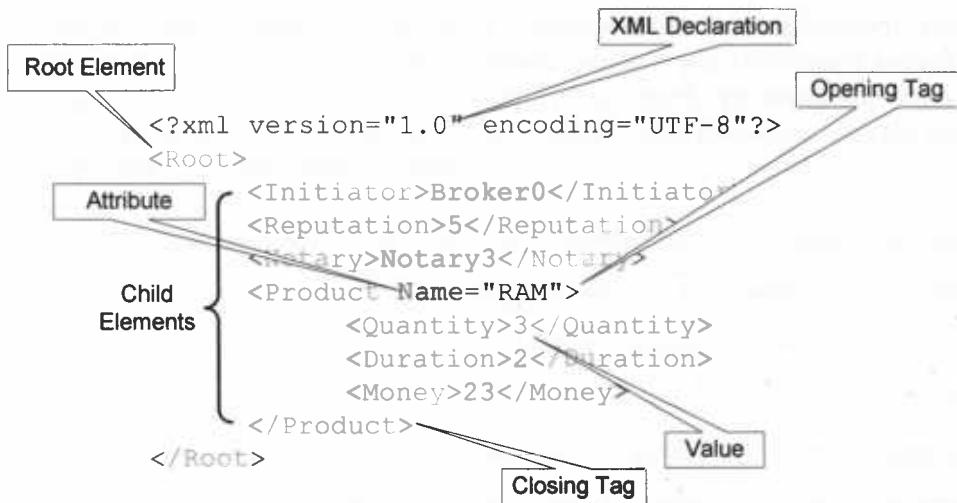
Σχήμα 3: Συνεργασία διαδικτυακών υπηρεσιών

## ΠΡΟΤΥΠΑ ΔΙΑΔΙΚΤΥΑΚΩΝ ΥΠΗΡΕΣΙΩΝ

### XML

Η εμφάνιση της XML τα τελευταία χρόνια την έχει καταστήσει την κυριότερη επιλογή για μία γλώσσα περιγραφής της ανταλλαγής πληροφορίας μεταξύ ανόμοιων συστημάτων. Το πρότυπο της XML (EXtensible Markup Language) προτάθηκε από τον οργανισμό W3C και αποτελεί μία γλώσσα υπομνηματισμού, όπως η HTML, για την περιγραφή δεδομένων. Ο υπομνηματισμός γίνεται με τη χρήση ετικετών της μορφής <όνομα ετικέτας>, οι οποίες σε αντίθεση με την HTML που είναι προκαθορισμένες από τη γλώσσα, ορίζονται από το χρήστη. Για την περιγραφή των δεδομένων η XML χρησιμοποιεί ένα DTD (Τύπο Εγγράφου XML – Document Type Definition), όντας με αυτό τον τρόπο αυτοπεριγραφόμενη. Ένα DTD ορίζει τα επιτρεπτά, σύμφωνα με το πρότυπο, στοιχεία ενός XML εγγράφου και χρησιμοποιείται για να καθοριστεί πότε είναι έγκυρο ένα XML έγγραφο. Για την παρουσίαση της πληροφορίας χρησιμοποιούνται ένα ή περισσότερα CSS (Cascading Style Sheets), τα οποία ορίζουν την οπτική μορφοποίηση των δεδομένων και τον τρόπο που αυτά θα προβληθούν στο χρήστη. Τα δομικά στοιχεία ενός DTD εγγράφου είναι όπως φαίνονται στο Σχήμα 4.

## ΜΔΕ: “Επιστήμη Υπολογιστών”



Σχήμα 4: Τμήμα Xml εγγράφου

- **Στοιχεία (elements)**

Είναι τα βασικά δομικά στοιχεία τόσο της XML όσο και της HTML και για τον υπομνηματισμό τους χρησιμοποιούνται ετικέτες.

- **Γνωρίσματα (attributes)**

Τα γνωρίσματα παρέχουν επιπλέον πληροφορία για τα στοιχεία που χρησιμοποιούνται, πέραν του περιεχομένου και του τύπου τους. Τα γνωρίσματα τοποθετούνται μέσα στην ετικέτα αρχής ενός στοιχείου και συνοδεύονται από την τιμή τους. Παραδείγματος χάριν, στην έκφραση <Product Name="RAM">...</Product>, το γνώρισμα **Name** χρησιμοποιείται για να δηλώσει ότι το όνομα του αντικειμένου **Product** είναι “RAM”.

## ΜΔΕ: “*Επιστήμη Υπολογιστών*”

Η έννοια του εγγράφου στην XML διαφέρει από την κλασσική του ερμηνεία, δεδομένου ότι ένα XML έγγραφο διαθέτει μία εσωτερική λογική, ανεξάρτητη από τη γλώσσα και τη μορφή. Η δομή αυτή έχει σημασία, καθώς η σειρά εμφάνισης των στοιχείων και η σύνδεση τους χρησιμοποιούνται για να εκφράσουν μια εσωτερική λογική.

Η χρησιμότητα της XML έγκειται στο γεγονός της λειτουργίας της ως συμβιβασμού ανάμεσα στις βάσεις δεδομένων με την αυστηρότητα στη διατύπωση των δομών τους και στο ελεύθερο κείμενο, γεγονός που την καθιστά αναγνώσιμη και στους μη ειδικούς. Η απλότητά της όμως δεν μειώνει την εκφραστικότητά της. Η ευκολία στη χρήση της έγκειται αφενός στην ευκολία ανταλλαγής των δεδομένων μεταξύ εφαρμογών, αλλά και στη δυνατότητα δόμησης πρώτα των δεδομένων και κατόπιν του σχήματος, το οποίο μπορεί να μην έχει τυπική μορφή, αλλά δύναται να επεκταθεί. Η XML ακολουθεί την προσέγγιση των ημιδομημάτων δεδομένων, τα οποία έχουν ενσωματωμένο σχήμα: κάθε στιγμιότυπο μεταφέρει στη διατύπωσή του το σχήμα που χρησιμοποιεί. Για παράδειγμα, το XML έγγραφο του Σχήματος 4 εσωκλείει την ερμηνεία του και είναι αναγνώσιμο χωρίς να απαιτείται γνώση του σχήματος. Η ιδιότητα αυτή διευκολύνει τη διακίνηση δεδομένων και διακρίνει την σημασιολογία των δεδομένων από την παρουσίασή τους, γεγονός που επιτρέπει την επαναχρησιμοποίηση της πληροφορίας. Επιπλέον, η ίδια σημασιολογία μπορεί να κωδικοποιηθεί με διαφορετικές μορφές, καθότι ο σχεδιασμός ενός DTD εξαρτάται από τη χρήση την οποία θα έχει.

## ΜΔΕ: “Επιστήμη Υπολογιστών”

### Πρότυπα παροχής δικτυακών υπηρεσιών

Ο Παγκόσμιος Ιστός, και στο μέλλον ο Σημασιολογικός Ιστός, χρησιμοποιείται όλο και περισσότερο ως πλατφόρμα επικοινωνίας εφαρμογών. Οι προγραμματιστικές διασυνδέσεις που διατίθενται πάνω από τον Παγκόσμιο Ιστό αναφέρονται ως «Διαδικτυακές Υπηρεσίες» (Web Services). Η εμφάνιση των Διαδικτυακών Υπηρεσιών έχει ορίσει μία μέθοδο για την έκθεση της λειτουργικότητας μίας εφαρμογής και την ανάλωσή της από άλλη εφαρμογή / σύστημα. Αν και οι τεχνολογίες και τα πρότυπα για τις διαδικτυακές υπηρεσίες που αφορούν τον εντοπισμό, περιγραφή, ασφάλεια και μεταφορά δεδομένων είναι ακόμα υπό ερευνητική σκοπιά, η XML έχει επικρατήσει ως πρότυπο ανταλλαγής πληροφορίας. Για το λόγο αυτό, τα πρότυπα παροχής διαδικτυακών υπηρεσιών που θα παρουσιάσουμε βασίζουν τη λειτουργικότητα τους στη χρήση της XML ως τεχνολογικό υπόβαθρο.

## SOAP

Πρόκειται για ένα πρωτόκολλο μηνυμάτων που προορίζεται για την ανταλλαγή δομημένης πληροφορίας σε ένα αποκεντρωμένο, κατανεμημένο περιβάλλον. Το SOAP χρησιμοποιεί το πρωτόκολλο HTTP, που ούτως ή άλλως χρησιμοποιείται ευρέως για τις μεταφορές μηνυμάτων, αλλά παρέχει μια πλατφόρμα για πιο σύνθετη ανταλλαγή πληροφορίας από εκείνη που επιτυγχάνεται με το HTTP, εισάγοντας ένα σύνολο HTTP επικεφαλίδων (headers) και XML δεδομένα. Πιο συγκεκριμένα καθορίζει:

- Ένα πρότυπο επικοινωνίας με βάση το οποίο η πληροφορία καταγράφεται σε ένα XML έγγραφο.
- Ένα σύνολο από συμβάσεις για τη χρήση του SOAP οι οποίες καθορίζουν πως γίνεται η αλληλεπίδραση με το RPC. Καθορίζει δηλαδή, πως ένας πελάτης μπορεί να στείλει ένα μήνυμα SOAP το οποίο θα ενεργοποιήσει μια απομακρυσμένη κλήση καθώς και πώς η Υπηρεσία η οποία θα δεχθεί την κλήση θα απαντήσει στέλνοντας ένα αντίστοιχο μήνυμα.
- Ένα σύνολο από κανόνες τους οποίους πρέπει να ακολουθεί κάθε κόμβος ο οποίος λαμβάνει ένα μήνυμα SOAP.
- Μια περιγραφή σχετικά με το πως θα πρέπει να μεταδίδεται ένα μήνυμα SOAP πάνω από το HTTP ή το SMTP.

Το SOAP είναι πρωτόκολλο μιας κατεύθυνσης (One-way). Δεν λαμβάνει υπόψη του την σημασιολογία των μηνυμάτων τα οποία αποστέλλονται. Οποιοδήποτε πρότυπο επικοινωνίας συμπεριλαμβανομένου και της αίτησης ή απάντησης, υλοποιείται από το σύστημα που βρίσκεται από κάτω. Αυτό

## ΜΔΕ: “*Επιστήμη Υπολογιστών*”

συμβαίνει διότι το SOAP σχεδιάστηκε για να προσφέρει “χαλαρή” συνδεσιμότητα μεταξύ των εφαρμογών οι οποίες αλληλεπιδρούν ανταλλάσσοντας ασύγχρονα μηνύματα μιας κατεύθυνσης. Οποιαδήποτε άλλη δυνατότητα επικοινωνίας, όπως για παράδειγμα ανταλλαγή μηνυμάτων διπλής κατεύθυνσης ή RPC κλήσεις θα πρέπει να γίνουν σε συνεργασία του SOAP με το πρωτόκολλο το οποίο βρίσκεται από κάτω. Για να γίνει, ενδεικτικά μια κλήση RPC, μέσω SOAP στην οποία διοχετεύονται κάποιες παράμετροι εισόδου και επιστρέφεται ένα αποτέλεσμα εξόδου, θα πρέπει να κωδικοποιηθούν οι παράμετροι εισόδου και η κλήση RPC σε ένα μήνυμα SOAP. Ομοίως και η απάντηση θα πρέπει να κωδικοποιηθεί σε SOAP. Ένα σύγχρονο πρωτόκολλο μεταφοράς θα πρέπει να χρησιμοποιηθεί για να μεταφερθούν τα δύο μηνύματα. Συνήθως το πρωτόκολλο αυτό είναι το HTTP. Το πρώτο μήνυμα το οποίο καλεί την υπηρεσία αποστέλλεται με το HTTP request ενώ η απάντηση με το HTTP reply. Για τον καθορισμό του τύπου δεδομένων των εγγράφων το SOAP χρησιμοποιεί το W3C XML Schema.

### Δομή και περιεγόμενο ενός SOAP μηνύματος

Το SOAP ανταλλάσσει πληροφορίες χρησιμοποιώντας μηνύματα. Τα μηνύματα αυτά χρησιμοποιούνται ως φάκελοι μέσα στους οποίους μπορεί κανείς να περιλάβει οποιουδήποτε είδους πληροφορία θέλει να στείλει. Ο φάκελος αποτελείται από δύο μέρη: την επικεφαλίδα (Header) και το κυρίως μέρος (Body). Η ύπαρξη της επικεφαλίδας είναι προαιρετική, ενώ το κυρίως μέρος είναι υποχρεωτικό. Και τα δύο μέρη μπορούν να έχουν μικρότερες επικεφαλίδες και κυρίως μέρη μέσα τους αντίστοιχα.

Το πρωτόκολλο SOAP υποθέτει ότι το μήνυμα αποστέλλεται από ένα κόμβο προς ένα άλλο και ότι ενδέχεται να μεσολαβούν στην διαδρομή αυτή κάποιοι κόμβοι οι οποίοι μπορεί να το επεξεργαστούν. Η πληροφορία την οποία θέλει

## ΜΔΕ: “*Επιστήμη Υπολογιστών*”

να στείλει στον παραλήπτη του μηνύματος, τοποθετείται στο κυρίως μέρος του μηνύματος, ενώ οποιαδήποτε επιπλέον πληροφορία τοποθετείται στην επικεφαλίδα. Η επικεφαλίδα γενικά περιλαμβάνει πληροφορία η οποία επεξεργάζεται από τους ενδιάμεσους κόμβους. Αυτός είναι ο λόγος για τον οποίο η επικεφαλίδα είναι προαιρετική.

Για την οργάνωση της επικεφαλίδας ή του κυρίως σώματος δεν υπάρχει κάποιος άλλος κανόνας. Η δομή τους καθορίζεται από δύο παράγοντες:

### 1. Τον χαρακτήρα αλληλεπίδρασης (interaction style)

Ο χαρακτήρας αλληλεπίδρασης μπορεί να είναι με βάση τα έγγραφα (Document Style) ή με το βάση το RPC (RPC style). Στην πρώτη περίπτωση οι κόμβοι οι οποίοι αλληλεπιδρούν έχουν συμφωνήσει στην δομή των εγγράφων τα οποία αποστέλλουν. Τα έγγραφα αυτά βρίσκονται μέσα στο φάκελο. Στην δεύτερη περίπτωση ο φάκελος περιλαμβάνει το αίτημα ή την απάντηση στο αίτημα. Η διαφορά στις δύο αυτές περιπτώσεις έγκειται στον τρόπο με τον οποίο τα μηνύματα αυτά κατασκευάζονται. Το κυρίως μέρος του μηνύματος αίτησης περιλαμβάνει την κλήση. Περιλαμβάνει το όνομα της μεθόδου την οποία θα καλέσει και τις παραμέτρους εισόδου. Το κυρίως μέρος του μηνύματος στην απάντηση περιλαμβάνει το αποτέλεσμα και τις παραμέτρους εξόδου. Και οι δύο εφαρμογές που αλληλεπιδρούν θα πρέπει να συμφωνήσουν για την υπογραφή της RPC μεθόδου. Η διαδικασία της μετάφρασης της υπογραφής της μεθόδου αυτής σε μηνύματα SOAP είναι συνήθως κρυμμένη από το ενδιάμεσο λογισμικό.

## 2. Από τους κανόνες κωδικοποίησης (encoding rules)

Οι κανόνες αυτοί καθορίζουν πως αναπαρίσταται τα δεδομένα σε XML. Και σε αυτήν την περίπτωση οι εφαρμογές οι οποίες αλληλεπιδρούν θα πρέπει να συμφωνήσουν σε μια κοινή κωδικοποίηση. Παρακάτω βλέπουμε ένα παράδειγμα μηνύματος σε SOAP:

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
    soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    <soap:Body>
        <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
            <m:Item>Apples</m:Item>
        </m:GetPrice>
    </soap:Body>
</soap:Envelope>
```

## WSDL

Καθώς τα πρωτόκολλα επικοινωνίας και οι τύποι μηνυμάτων προτυποποιούνται στην κοινότητα του Διαδικτύου, η ανάγκη για την περιγραφή της επικοινωνίας μεταξύ συστημάτων με δομημένο τρόπο γίνεται πιο επιτακτική. Θεωρούμενη μαζί με το SOAP και το UDDI ως ένα από τα τρία βασικά πρότυπα για τις διαδικτυακές υπηρεσίες, η WSDL αντιμετωπίζει αυτή την ανάγκη, αποτελώντας ένα XML πρότυπο για την περιγραφή δικτυακών υπηρεσιών ως ένα σύνολο σημείων που επικοινωνούν βάσει μηνυμάτων, τα οποία περιέχουν πληροφορία είτε σχετικά με έγγραφα είτε με διαδικασίες.

Ένας WSDL ορισμός, αποτελούμενος από ένα XML έγγραφο, περιγράφει τον τρόπο με τον οποίο μπορεί να γίνει προσβάσιμη μια Διαδικτυακή υπηρεσία και το σύνολο των λειτουργιών που θα εκτελέσει. Αποτελείται από δύο μέρη: το πρώτο μέρος είναι αφηρημένο (Abstract), ενώ το δεύτερο είναι συγκεκριμένο (Concrete). Στο πρώτο μέρος καθορίζονται οι τύποι δεδομένων των παραμέτρων, τα μηνύματα τα οποία στέλνονται και λαμβάνονται, οι συναρτήσεις και οι τύποι πύλης (Port types). Στο δεύτερο μέρος καθορίζεται η σύνδεση με κάποιο πρωτόκολλο μεταφοράς, οι υπηρεσίες και οι πύλες.

- **Types**

Καθορίζονται οι τύποι δεδομένων χρησιμοποιώντας ένα σύστημα τύπων, για παράδειγμα το XSD.

- **Messages**

Ορίζονται με αφηρημένο τρόπο ο τύπος μηνυμάτων τα οποία ανταλλάσσονται.

- **Operations**

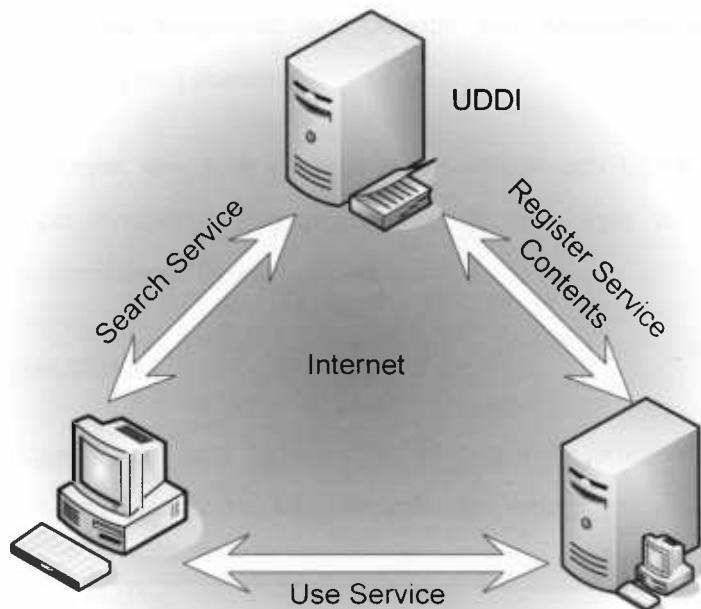
Περιγράφονται με αφηρημένο τρόπο οι ενέργειες που υποστηρίζονται από μια υπηρεσία

## **UDDI**

Το UDDI είναι ένας μηχανισμός μητρώου το οποίο μπορεί να χρησιμοποιηθεί για την ανεύρεση των περιγραφών των διαδικτυακών υπηρεσιών.

### Ορισμός

Το UDDI προέρχεται από τα αρχικά των λέξεων Universal Description and Discovery Integration. Πρόκειται για ένα σύνολο από μητρώα τα οποίο είναι προσβάσιμα μέσω του Διαδικτύου. Παρουσιάζουν πληροφορίες για έναν οργανισμό και για την τεχνική διεπαφή (API) της. Τα μητρώα αυτά χρησιμοποιούνται από πολλούς ιστότοπους στους οποίους υπάρχουν χειριστές. Οποιοσδήποτε θέλει να δημοσιεύσει κάποιες πληροφορίες για κάποιον οργανισμό μπορεί να το κάνει, όπως επίσης μπορεί να αναζητήσει σχετικές πληροφορίες. Οι λειτουργίες αυτές δεν χρεώνονται.



Σχήμα 5: Τυπική Λειτουργία UDDI

Πληροφορίες τις οποίες μπορεί να καταγράψει μια εταιρεία στο ευρετήριο αφορούν τα δεδομένα τα οποία βοηθούν τους υπόλοιπους να απαντήσουν στις ερωτήσεις «ποιος, τι, πού και πώς». Πληροφορίες που αφορούν το όνομα, προσδιοριστικά του οργανισμού, ή τρόποι επικοινωνίας μαζί του απαντούν στην ερώτηση «ποιος». Η ερώτηση «τι» αφορά ταξινομήσεις με βάση τα διεθνή πρότυπα καθώς και περιγραφές για την υπηρεσία που ο οργανισμός προσφέρει. Πληροφορίες που αφορούν την διεύθυνσή του στο Διαδίκτυο ή τη διεύθυνση του ηλεκτρονικού του ταχυδρομείου απαντούν στην ερώτηση «πού». Η ερώτηση «πώς» απαντάται κάνοντας αναφορές σε πληροφορίες σχετικές με διεπαφές και άλλες ιδιότητες μιας συγκεκριμένης υπηρεσίας. Οι

## ΜΔΕ: “*Επιστήμη Υπολογιστών*”

ιδιότητες αυτής της υπηρεσίας περιγράφουν πώς λειτουργεί ένα πακέτο λογισμικού ή μια τεχνική διεπαφή.

Η κατηγοριοποίηση των παρεχόμενων πληροφοριών του UDDI μπορεί να γίνει καλύτερα αντιληπτή χρησιμοποιώντας ως αναλογία τη δομή ενός τηλεφωνικού καταλόγου. Έχουμε συνεπώς:

- Λευκές σελίδες

Παρέχουν μια λίστα οργανισμών, πληροφορίες επικοινωνίας καθώς και τις παρεχόμενες από κάθε εταιρεία υπηρεσίες.

- Κίτρινες σελίδες

Αποτελούν ταξινομήσεις τόσο των εταιρειών όσο και των υπηρεσιών με βάση επίσημα πρότυπα ή πρότυπα καθορισμένα από το χρήστη.

- Πράσινες σελίδες

Οι πληροφορίες αυτού του είδους περιγράφουν πώς γίνεται κλήση των υπηρεσιών.

### Ορισμός δομών δεδομένων

Στο UDDI αποθηκεύονται τέσσερα είδη πληροφοριών: πληροφορίες που αφορούν οργανισμούς, υπηρεσίες, συνδέσεις και πληροφορίες για τον καθορισμό προτύπων.

- Business Entity

Περιγράφει έναν οργανισμό ο οποίος παρέχει μια υπηρεσία. Καταγράφει το όνομα της εταιρείας, την διεύθυνσή της και άλλα σχετικά στοιχεία επικοινωνίας.

- Business Service

Περιγράφει ένα σύνολο από διαδικτυακές υπηρεσίες, οι οποίες παρέχονται από έναν οργανισμό που εκφράζεται από μια Business Entity

- Business Template

Η δομή αυτή αποθηκεύει πληροφορίες για την χρήση-κλήση μιας υπηρεσίας.

- Tmodels

Για να μπορούν δύο κομμάτια λογισμικού να είναι συμβατά μεταξύ τους, να μπορούν δηλαδή να συνεργάζονται, θα πρέπει να μοιράζονται μερικά κοινά πρότυπα και σχεδιαστικούς στόχους.

- tModels

Για να υποστηριχθεί η διακριτή αναγνώριση δημοσίων και ιδιωτικών προδιαγραφών χρησιμοποιούνται πληροφορίες για τις περιγραφές. Οι πληροφορίες αυτές, οι οποίες στην ουσία είναι μεταδεδομένα, λέγονται tModels.

ΜΔΕ: “*Επιστήμη Υπολογιστών*”

## Ταξινόμηση και Προσδιοριστικές προδιαγραφές

Ένα βασικό πλεονέκτημα καταγραφής των πληροφοριών σε έναν οργανισμό στο UDDI είναι η δυνατότητα ταξινόμησής ή ανάθεσης σε αυτές κωδικών κατηγοριών. Ενδεικτικοί κώδικες είναι ο NAICS, UN/SPC και ο SIC Codes.

Οι οργανισμοί μπορούν να χαρακτηρίσουν τα δεδομένα που καταγράφουν με οποιοδήποτε αριθμό κωδικών κατηγοριών. Με βάση την ταξινόμηση ή και με βάση το χαρακτηρισμό μπορούν να γίνουν μια σειρά από ερωτήματα. Στην δεύτερη έκδοση του UDDI οι προδιαγραφές μπορούν να περιλαμβάνουν και ελεγμένες κατηγοριοποιήσεις καθώς και ταξινομήσεις προσδιοριστικών πληροφοριών.

## Σχεδίαση και Αρχιτεκτονική

Η διεπαφή του UDDI είναι σχεδιασμένη για να παρέχει ένα απλό μηχανισμό αίτησης-απόκρισης για να ανακαλυφθούν στοιχεία για τους οργανισμούς και τεχνικές πληροφορίες σύνδεσης (binding). Η βασική σχεδιαστική αρχή του UDDI είναι η απλότητα. Δόθηκε ιδιαίτερη προσοχή στο να αποφευχθεί η πολυπλοκότητα και να δοθεί η δυνατότητα για απευθείας πρόσβαση στην καταγεγραμμένη πληροφορία.

## Ασφάλεια

Η πρόσβαση στο UDDI προγραμματιστικά γίνεται μέσω κλήσεων του API. Υπάρχουν δύο είδη API. Το API του εκδότη ή αλλιώς δημοσίευσης (Publisher API) και το API για επερωτήσεις (Inquiry API). Το πρώτο χρησιμοποιείται για την αλληλεπίδραση των παροχέα της υπηρεσίας με το UDDI με σκοπό την

## ΜΔΕ: “*Επιστήμη Υπολογιστών*”

δημοσίευση πληροφοριών σχετικά με την οργανισμό. Το API των επερωτήσεων χρησιμοποιείται για την πρόσβαση στα στοιχεία του UDDI.

Για να μπορέσει κανείς να χρησιμοποιήσει το API του εκδότη, θα πρέπει πρώτα να πιστοποιηθεί ότι είναι αυτός που ισχυρίζεται. Αυτό γίνεται με την βοήθεια του πρωτοκόλλου που έχει επιλέξει ο κάθε χειριστής (operator) και αφού έχει εγγραφεί πρώτα στην Διαδικτυακή τοποθεσία η οποία είναι υπεύθυνη για το UDDI.

Τόσο το API δημοσίευσης όσο και το API επερωτήσεων εκτίθενται ως SOAP μηνύματα πάνω από το HTTP. Το HTTPS χρησιμοποιείται όταν πρόκειται να δημοσιευτούν κάποια στοιχεία. Αντιθέτως, δεν χρησιμοποιείται επικύρωση όταν πρόκειται απλά να γίνουν επερωτήσεις στο UDDI.

### Εκδόσεις

Όπως κάθε API έτσι και τα API του UDDI έχουν κάποιες εκδόσεις. Για να αποφευχθεί, μάλιστα, το γεγονός κάποιος να χρησιμοποιεί άλλο API από αυτό που πρέπει, χρησιμοποιούνται σφραγίδες σε κάθε μήνυμα SOAP οι οποίες προσδιορίζουν την έκδοση του API στην οποία αναφέρεται το μήνυμα. Η σφραγίδα αυτή έχει τη μορφή μιας XML ιδιότητας (Attribute). Η ιδιότητα αυτή ονομάζεται generic.

### Μηνύματα SOAP

Το UDDI χρησιμοποιεί SOAP και HTTP για να έχει έναν απλό μηχανισμό για να στέλνει XML μηνύματα σε Διαδικτυακούς τόπους χειριστών χρησιμοποιώντας το πρωτόκολλο HTTP-POST. Όλες οι απαντήσεις επιστρέφονται με τη μορφή ενός HTTP εγγράφου.

### **Συμβάσεις XML**

Η προγραμματιστική διεπαφή του UDDI βασίζεται στην XML.

#### **Χειρισμός λαθών**

Στην περίπτωση που γίνει κάποια λανθασμένη κλήση ή κάποια κλήση στην οποία το καλούμενο αντικείμενο δεν μπορεί να προσδιοριστεί τότε θα επιστραφεί αναφορά για λάθος SOAP μαζί με τους αντίστοιχους κωδικούς. Αν κατά την επεξεργασία ενός μηνύματος προκύψει σφάλμα στο επίπεδο της εφαρμογής τότε θα επιστραφεί η αντίστοιχη αναφορά (Disposition Report) η οποία θα βρίσκεται μέσα στην αναφορά του SOAP για λάθος.

#### **Κενά**

Επιτρέπεται η αποθήκευση δεδομένων όταν αυτά έχουν λευκά κενά. Η μοναδική περίπτωση που απαγορεύεται είναι όταν το σχήμα του UDDI δεν το επιτρέπει.

### **XML Κωδικοποίηση**

Η προεπιλεγμένη διάταξη για τα δεδομένα τα οποία αποθηκεύονται στο UDDI είναι η δυαδική. Η επιλογή αυτή είναι άνευ σημασίας για μερικές γλώσσες ενώ ευνοεί τις γλώσσες της αλφαριθμητικής. Ομοίως, η XML επιτρέπει έναν μεγάλο αριθμό από επιλογές κωδικοποίησης. Οι χειριστές του UDDI χρειάζονται να υποστηρίζουν μόνο μια από αυτές, την UTF-8.

## Τύποι επερωτήσεων

Επερωτήσεις στο UDDI μπορούν να γίνουν με τους εξής τρεις τρόπους:

- Επερώτηση μέσα από το φυλλομετρητή (Browse pattern)

Το UDDI παρέχει την δυνατότητα επερωτήσεων μέσα από φυλλομετρητή με την μορφή των `find_xx` κλήσεων του API. Αυτές οι κλήσεις σχηματίζουν τα κριτήρια αναζήτησης. Το αποτέλεσμα αυτών των ερωτήσεων είναι μηνύματα τα οποία πληροφορούν για τα δεδομένα τα οποία είναι καταγεγραμμένα στο UDDI. Μια τυπική αλληλουχία επερωτήσεων μπορεί να είναι να βρεθούν οι πληροφορίες οι οποίες έχουν καταγραφεί για λογαριασμό κάποιας εταιρείας (οργανισμού). Η αλληλουχία αυτή θα ξεκινούσε με κλήση της συνάρτηση `find_business`, με ορίσματα ίσως κάποια γράμματα από το όνομα που ίσως ψάχνουμε. Η επερώτηση αυτή επιστρέφει μια Business List. Το αποτέλεσμα αυτό περιέχει πληροφορίες όπως κλειδιά, ονόματα και περιγραφές τα οποία προκύπτουν από την εγγεγραμμένη Business Entity της οποίας το όνομα ταιριάζει με τα αρχικά που δόθηκαν ως παράμετροι αναζήτησης. Στην περίπτωση που εντοπιστεί η ζητούμενη εταιρεία (Business) μπορεί κανείς να εξάγει την πληροφορία για το Business Service, αναζητώντας συγκεκριμένους τύπους υπηρεσιών χρησιμοποιώντας την `find_service` του API.

- Drill-down pattern

Από τη στιγμή που θα προκύψει ένα κλειδί για ένα από τους τέσσερις τύπους δεδομένων του UDDI, μπορεί να χρησιμοποιηθεί το κλειδί αυτό για να προσπελάσει κανείς τις λεπτομέρειες που έχουν

## ΜΔΕ: “*Επιστήμη Υπολογιστών*”

αποθηκευτεί για το συγκεκριμένο στιγμιότυπο. Οι τύποι δεδομένων του UDDI είναι Business Entity, Business Service, Binding Template και tModel. Για να προσπελάσουμε πλήρως τις πληροφορίες κάθε μια από αυτές τις δομές χρησιμοποιούμε την `get_xx` του API με παράμετρο το αντίστοιχο κλειδί.

- Invocation pattern

Ο τρόπος αυτός χρησιμοποιείται κυρίως από προγράμματα για κλήση με αυτόματο τρόπο. Το πρόγραμμα το οποίο αναλαμβάνει να καλέσει την υπηρεσία η οποία έχει εντοπιστεί, συλλέγει την κατάλληλη πληροφορία από την δομή Binding Template, θα πρέπει να εντοπίσει στο UDDI. Συνήθως το ποια υπηρεσία θα κληθεί, αποφασίζεται τη στιγμή της κλήσης, οπότε γίνεται και η αντίστοιχη αναζήτηση στο UDDI.

ΜΔΕ: “*Επιστήμη Υπολογιστών*”

## *Kεφάλαιο 3*

# ENTERPRISE NETWORK INTERCONNECTION

Σε αυτό το κεφάλαιο θα εξετάσουμε τους βασικούς στόχους που θέσαμε για την δημιουργία του πρωτοκόλλου. Στην συνέχεια θα προσπαθήσουμε να το συγκρίνουμε με κάποια “φανταστικά” πρωτόκολλα που σκεφτήκαμε ότι θα ήταν απλά στην υλοποίησή τους, αλλά στην συνέχεια τα απορρίψαμε για διάφορους λόγους. Τέλος θα αναφέρουμε τα πλεονεκτήματα και μειονεκτήματα που φέρνει το πρωτόκολλο σε όλους τους συμμετέχοντες.

## EN INTERCONNECTION

### ΟΡΙΣΜΟΣ

Πρόκειται για ένα τρόπο επικοινωνίας μεταξύ Enterprise Network (EN), ώστε να δημιουργηθούν Εικονικοί Οργανισμοί (VO) που χρησιμοποιούν Web Services (WS) από διαφορετικούς EN.

### ΣΤΟΧΟΙ

Οι βασικοί σχεδιαστικοί στόχοι ήταν η δημιουργία ενός πρωτοκόλλου, που θα πληρούσε τις εξής προδιαγραφές:

- **Εύκολο στην εγκατάσταση**

Να λειτουργεί σε αρμονία με το ήδη εγκατεστημένο σύστημα του TrustCom Project. Αυτό σημαίνει ότι πρέπει να παρέχει όλες τις λειτουργίες για τις οποίες είναι υπεύθυνο, προσθέτοντας όμως όσο το δυνατόν λιγότερα components.

- **Ανεξάρτητο**

Να μην εξαρτάται από αλλαγές που θα γίνουν στα components πάνω στα οποία στηρίζεται, εφόσον βέβαια αυτά, πληρούν τις ίδιες διεπαφές.

- **Επεκτάσιμο**

Οποιοδήποτε component του πρωτοκόλλου μπορεί να αντικατασταθεί, ή να προστεθούν νέα, μιας και καθένα θεωρήθηκε στην διάρκεια της σχεδίασης σαν ανεξάρτητη οντότητα.

Για όλους τους παραπάνω λόγους χρησιμοποιήθηκαν Web Services που χρησιμοποιούν για την επικοινωνία τους μηνύματα XML.

Οι λόγοι όμως που οδήγησαν στην δημιουργία αυτού του πρωτοκόλλου, ήταν:

- **Η πληρότητα**

Επειδή όλοι οι EN έχουν πεπερασμένο αριθμό πόρων, είναι σχεδόν αδύνατο να επιτύχουν πλήρη κάλυψη τους. Σχεδόν πάντα θα περισσεύουν κάποιοι πόροι που δεν θα επαρκούν για την κάλυψη κάποιων αναγκών. Με την εφαρμογή του πρωτοκόλλου όμως, μπορούμε να διαθέσουμε αυτούς τους πόρους σε κάποιον άλλον EN.

- **Η κλιμακωσιμότητα**

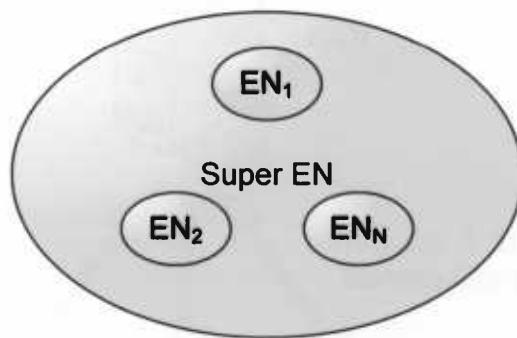
Επειδή μπορεί να συνεργαστούν πάρα πολλοί EN και ο καθένας τους να προσφέρει ακόμη και όλους τους πόρους που διαθέτει, μπορούν να δημιουργηθούν πολύ μεγαλύτερα VO, που θα μπορούν να καλύψουν ανάγκες που σε άλλη περίπτωση θα ήταν αδύνατο να εξυπηρετηθούν.

## ΔΙΑΦΟΡΑ ΠΡΩΤΟΚΟΛΛΟΥ ΑΠΟ ΑΛΛΑ

Για την υλοποίηση του πρωτοκόλλου θα μπορούσαν να χρησιμοποιηθούν διαφορετικοί, απλούστεροι στην υλοποίηση τρόποι όπως είναι αυτοί που ακολουθούν στα επόμενα δύο υποκεφάλαια:

### Δημιουργία Super Enterprise Network

Όλη η αρχιτεκτονική που χρησιμοποιεί το TrustCom Consortium θα μπορούσε να χρησιμοποιηθεί αυτούσια ώστε να δημιουργηθεί ένας Super EN. Στην ουσία αυτό θα καταργούσε την ύπαρξη των υπολοίπων EN. Επίσης θα παρουσίαζε σημαντικά προβλήματα κλιμάκωσης, μιας και ένας EN θα επωμίζονταν το βάρος όλων των VO που υπάρχουν.



Σχήμα 6: Αφηρημένη μορφή Super EN

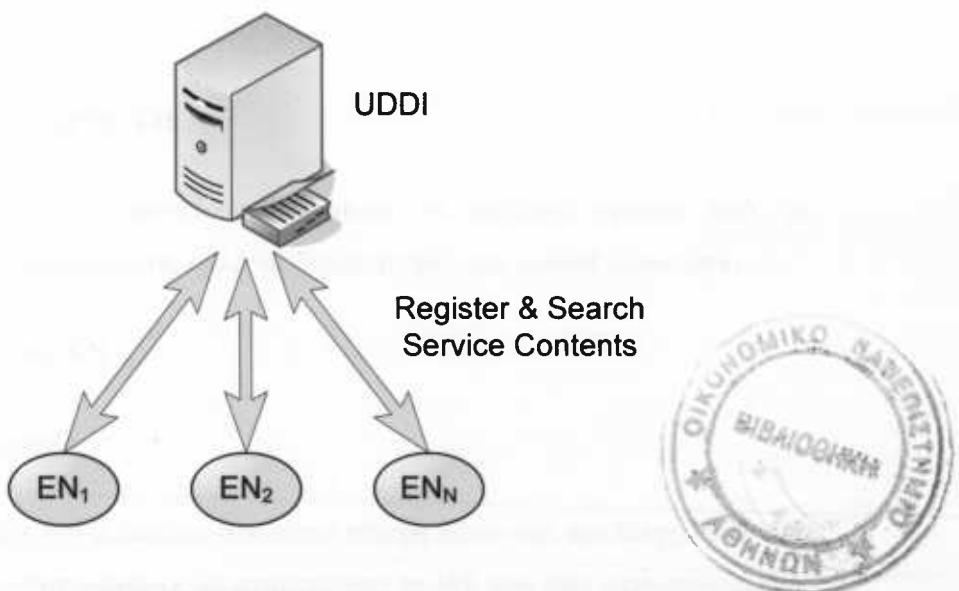
Ένα άλλο σημαντικό πρόβλημα, ήταν το κατά πόσο θα επιθυμούσαν όλοι οι EN να συνεργαστούν, ώστε να δημιουργηθεί αυτός ο EN. Προβλήματα θα ανέκυπταν με τις πολιτικές που θα έπρεπε να εφαρμοστούν, τους τρόπους πληρωμής, την ανάληψη ευθυνών κτλ. Τέλος μην ξεχνάμε ότι ένας τέτοιος

## ΜΔΕ: “Επιστήμη Υπολογιστών”

οργανισμός θα οδηγούσε την αγορά σε μια μονοπωλιακή κατάσταση με όλα τα θετικά και αρνητικά στοιχεία που επιφέρει αυτό.

### Κοινό UDDI

Αντί της δημιουργίας ενός Super EN, θα μπορούσε να προταθεί η δημιουργία ενός κοινού UDDI ή μιας βάσης δεδομένων, στην οποία όποιος EN επιθυμούσε θα μπορούσε να ανακοινώσει τα WS που διαθέτει. Στην συνέχεια, όποιος EN ήθελε θα ήταν σε θέση με απλές ερωτήσεις σε αυτό να βρει με ποιους EN θα έπρεπε να συνεργαστεί.



Σχήμα 7: Αφηρημένη μορφή Κοινού UDDI

Αυτός ο τρόπος, αν και λύνει πολλά από τα προβλήματα του προηγούμενου, συνεχίζει να έχει κάποια αρνητικά στοιχεία που πρέπει να λυθούν όπως:

## ΜΔΕ: “Επιστήμη Υπολογιστών”

- Ποιος ελέγχει το κεντρικό UDDI
- Πώς γίνεται η αξιολόγηση τόσο των EN που συμμετέχουν όσο και των WS που προσφέρουν
- Με ποιόν τρόπο επιστρέφονται τα αποτελέσματα στον χρήστη

Τέλος θα πρέπει να σκεφτούμε ότι οι EN δεν έχουν κανένα λόγο να αποκαλύψουν στους ανταγωνιστές τους τον αριθμό των διαθέσιμων πόρων που έχουν ανά πάσα στιγμή, μιας και κάτι τέτοιο θα μπορούσε να οδηγήσει σε αθέμιτο ανταγωνισμό (π.χ. αναπροσαρμογή των τιμών ανάλογα με τον φόρτο εργασίας των υπολοίπων).

## ΘΕΤΙΚΑ ΚΑΙ ΑΡΝΗΤΙΚΑ

Η υλοποίηση που προτείνουμε έρχεται να καλύψει αρκετά από τα προηγούμενα προβλήματα, αλλά να δημιουργήσει και μερικά καινούργια.

### Θετικά για τους EN

- Αυτονομία

Ο κάθε EN βρίσκεται ανά πάσα στιγμή κάτω από τον έλεγχο του. Δεν είναι υποχρεωμένος να ανακοινώσει τα WS που έχει καταγεγραμμένα, τις τιμές που τα κοστολογεί, αλλά ούτε και την διαθεσιμότητά τους. Έτσι μπορεί να εφαρμόσει όποια πολιτική εξυπηρετεί καλύτερα τα συμφέροντά του. Επιπλέον μπορεί να χρησιμοποιήσει οποιαδήποτε υποσυστήματα επιθυμεί, χωρίς να είναι υποχρεωμένος να αποκαλύψει την εσωτερική του δομή.

- Αξιοπιστία

Εξαιτίας της χρήσης μηχανισμών φήμης [5] για όλους τους συμμετέχοντες στο πρωτόκολλο, αλλά και της μεταβολής της από τους ίδιους τους EN, η αξιοπιστία του συστήματος είναι σε υψηλά επίπεδα.

- Ασφάλεια

Η χρησιμοποίηση WS και άρα η επικοινωνία πάνω από δίκτυα υπολογιστών, έχει ως αποτέλεσμα τα μηνύματα (Xml αρχεία) που αποστέλλονται να είναι ευάλωτα σε επιθέσεις και αλλαγές. Για τον λόγο αυτό προτείνουμε η ανταλλαγή μηνυμάτων να γίνεται πάνω από ασφαλή κανάλια επικοινωνίας και με την χρήση μεθόδων κρυπτογράφησης [6][7]. Να μην ξεχνάμε βέβαια ότι η όλη διαδικασία της εύρεσης πόρων κρατάει κάποιο μικρό χρονικό διάστημα, πράγμα που δυσκολεύει ακόμη περισσότερο την αλλαγή κάποιου μηνύματος.

Επίσης με την προσθήκη ενός TTP (Notary) ανάμεσα στον EN που ζητάει κάποια προσφορά και σε αυτούς που επιθυμεί να του απαντήσουν, η ασφάλεια μεγαλώνει μιας και το Notary είναι υπεύθυνο εκτός των άλλων για την πιστοποίηση των συμμετεχόντων, καθώς και για τον έλεγχο των μηνυμάτων για σφάλματα.

- Εύρεση καλύτερης προσφοράς

Ο EN ζητάει να πάρει προσφορές μόνο από αυτούς που επιθυμεί με βάση ένα SLA-template. Αφού συγκεντρώσει τις προσφορές, είναι σε θέση να διαλέξει αυτή που τον συμφέρει καλύτερα, ή να τις απορρίψει όλες.

ΜΔΕ: “*Επιστήμη Υπολογιστών*”

## Αρνητικά για τους EN

Το μόνο αρνητικό στοιχείο που υπάρχει, είναι η αύξηση της πολυπλοκότητας που επιφέρει η χρήση του πρωτοκόλλου. Αυτό όμως είναι αναπόφευκτο, αν επιθυμούμε να υποστηρίζουμε και αυτή την λειτουργία.

## Θετικά για τους Service Providers

- Ανωνυμία

Οι Service Providers είναι σίγουροι πως από την στιγμή που θα δηλώσουν τις υπηρεσίες που διαθέτουν σε κάποιον EN της εμπιστοσύνης τους, θα διατηρήσουν την ανωνυμία τους, μιας και προς τους υπόλοιπους EN, φαίνεται ότι η υπηρεσία που διαθέτεται είναι κομμάτι του αρχικού EN.

- Αύξηση Χρήσης

Υπάρχει αύξηση της πελατειακής βάσης όλων των EN, πράγμα το οποίο οδηγεί στην αύξηση της ζήτησης για πόρους, με αποτέλεσμα την αύξηση των εσόδων των Service Provider.

## Αρνητικά για τους Service Providers

Το βασικότερο πρόβλημα που θα αντιμετωπίσουν οι Service Providers είναι η αύξηση του ανταγωνισμού, μιας και τώρα θα έχουν να ανταγωνιστούν περισσότερους.

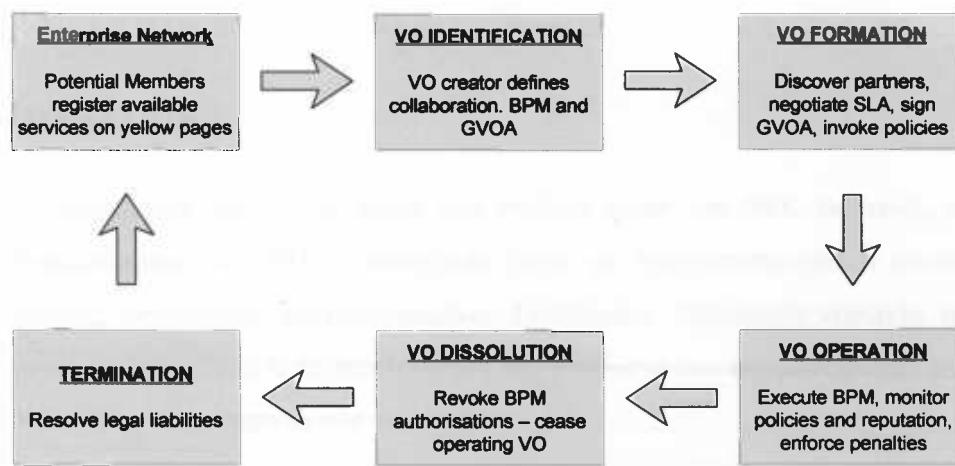
## Κεφάλαιο 4

### ΑΡΧΙΤΕΚΤΟΝΙΚΗ

Σε αυτό το κεφάλαιο θα δούμε την αρχιτεκτονική που χρησιμοποιήθηκε για να υποστηριχθεί η επικοινωνία μεταξύ EN πάνω από την ήδη υπάρχουσα αρχιτεκτονική του TrustCom Project[8].

### VO LIFECYCLE

To TrustCom project ακολουθεί το μοντέλο κύκλου ζωής που περιλαμβάνει μια σειρά από φάσεις. Σε κάθε φάση είναι απαραίτητο να γίνουν κάποιες ενέργειες, ώστε να υποστηριχθεί η σωστή και ομαλή λειτουργία των VO. Αυτές οι φάσεις είναι αυτές που φαίνονται στο Σχήμα 8.



Σχήμα 8: Κύκλος ζωής VO

## IDENTIFICATION

Σε αυτό το στάδιο καθορίζεται το είδος υπηρεσίας και ο ρόλος που θα έχει ο κάθε Application Service Provider, καθώς επίσης καθορίζεται και ο κοινός επιχειρησιακός στόχος που έχουν θέσει. Τέλος ορίζονται επίπεδα Trust, Security και Contract Management (TSC) σε κάθε διαφορετικό ρόλο.

## FORMATION

Σε αυτό το στάδιο, συγκεντρώνονται όλα εκείνα τα WS που μπορούν να επιλύσουν τον κοινό στόχο και πληρούν τις προδιαγραφές TSC. Οι προδιαγραφές TSC μετατρέπονται σε policies και υπογράφεται το General VO Agreement (GVOA) όπου περιγράφονται όλες οι πολιτικές του VO καθώς και τα SLA που σχετίζονται με τον κάθε ASP. Το SLA θα τεθεί σε διαπραγμάτευση με κάθε μέλος του VO, για κάθε WS που παρέχει.

## OPERATION

Πρόκειται για την κύρια φάση του κύκλου ζωής του VO. Σε αυτή, οι συμμετέχοντες του VO συνεισφέρουν ώστε να πραγματοποιηθεί ο κοινός στόχος, εκτελώντας προκαθορισμένες διαδικασίες. Σημαντικά στοιχεία σε αυτή την φάση είναι η παρακολούθηση της απόδοσης των συμμετεχόντων του VO καθώς και η τήρηση των πολιτικών.

ΜΔΕ: “*Επιστήμη Υπολογιστών*”

## EVOLUTION

Πρόκειται για ένα τμήμα του Operation σταδίου. Όταν ένα μέλος του VO αποτύχει να εκπληρώσει τις υποχρεώσεις του ή παραβεί κάποιουν κανόνες, τότε ο VO Manager θα πρέπει πιθανότατα να τον αντικαταστήσει με κάποιον άλλο. Σε αυτή την φάση μπορεί να περιέχονται και τμήματα από τις φάσεις του Identification και Formation, όπως η εύρεση καινούργιων μελών και η επαναδιαπραγμάτευση των SLA. Προφανώς υπάρχει σημαντικό πρόβλημα στη δομή και τον τρόπο λειτουργίας του VO, που μπορεί να δημιουργήσει ακόμη και στην επανέναρξη λειτουργίας του από μηδενική βάση.

## DISSOLUTION & TERMINATION

Πρόκειται για το τελικό στάδιο στον κύκλο ζωής του VO και επέρχεται όταν έχει επιτευχθεί ο στόχος του. Η δομή του VO διαλύεται και εκτελούνται όλες εκείνες οι διαδικασίες που απαιτούνται για να καλυφθούν οι οικονομικές απαιτήσεις των μελών. Από την μεριά του Trust και του Security, περιλαμβάνονται διαδικασίες όπως:

- Διάλυση των συνεργασιών
- Ακύρωση των πιστοποιητικών
- Ενημέρωση του Reputation όλων των μελών

Η φάση του Termination μπορεί να γίνει αρκετό καιρό μετά το Dissolution καθώς κάποια στοιχεία για τον συγκεκριμένο VO θα πρέπει να φυλαχτούν.

## SUBSYSTEMS

Για την υποστήριξη της λειτουργίας υπογραφής συμβολαίων μεταξύ του EN και των SP που περιέχει, έχουν αναπτυχθεί μια σειρά από υποσυστήματα, όπου κάθε ένα είναι επιφορτισμένο με την εκτέλεση μιας συγκεκριμένης λειτουργίας. Στα ήδη υπάρχοντα υποσυστήματα, προσθέσαμε και κάποια ακόμη που συνεργάζονται με τα προηγούμενα, ώστε να μπορέσουμε να υποστηρίξουμε την διασύνδεση διαφορετικών EN. Παρακάτω ακολουθεί μια σύντομη περιγραφή για τις βασικές λειτουργίες που επιτελεί κάθε υποσύστημα.

### SERVICE CONSUMER (SC)

Πρόκειται για τον καταναλωτή-χρήστη της υπηρεσίας. Στο πρωτόκολλο που προτείνουμε, τον ρόλο αυτό τον παίζει ένας EN που ζητάει κάποια υπηρεσία από άλλους, ώστε να δημιουργηθεί ένας VO.

### SERVICE PROVIDER (SP)

Πρόκειται για τον πάροχο της υπηρεσίας. Στο πρωτόκολλο που προτείνουμε, τον ρόλο αυτό τον παίζει ένας ή περισσότεροι EN από τους οποίους ζητείται να προσφέρουν κάποια υπηρεσία με συγκεκριμένες προδιαγραφές, ώστε να δημιουργηθεί ένας VO.

## VO MANAGER

Πρόκειται για τον κύριο μηχανισμό μάνατζμεντ που λειτουργεί για λογαριασμό του SC. Είναι υπεύθυνος για τον συντονισμό και την ομαλή λειτουργία του VO, σε όλες της φάσεις του κύκλου ζωής του μιας και εκτελεί:

- Την εύρεση πιθανών συνεργατών για την δημιουργία του VO με την βοήθεια του μηχανισμού Reputation που υπάρχει στην φάση του Identification.
- Την τήρηση των συμβολαίων για ασφάλεια και εχεμύθεια
- Την εφαρμογή των πολιτικών που προκύπτουν από τα συμβόλαια για παροχή συγκεκριμένων υπηρεσιών, τόσο σε ποιότητα όσο και σε χρόνο.
- Την επιβολή ποινών αν κάτι τέτοιο κριθεί απαραίτητο.
- Την μετεξέλιξη του VO στην φάση του evolution.

Είναι χωρισμένο σε τρία τμήματα

### 1. VO Lifecycle Management

Είναι υπεύθυνο για τις αλλαγές που προκύπτουν κατά την διάρκεια του κύκλου ζωής.

### 2. VO Membership Management

Είναι υπεύθυνο για τον συντονισμό και έλεγχο των συμμετεχόντων κάθε VO.

### 3. GVOA Management

Είναι υπεύθυνο για τον έλεγχο του General VO Agreement

## ΜΔΕ: “*Επιστήμη Υπολογιστών*”

Αυτά τα τμήματα επικοινωνούν κυρίως με το υποσύστημα SLA Management που δημιουργεί και διαχειρίζεται τα SLAs και το υποσύστημα Business Process Enactment and Orchestration που καθορίζει την λειτουργία του VO και την θέτει σε εφαρμογή στην φάση του Operation.

### SLA NEGOTIATOR

Πρόκειται για ένα υποσύστημα που παρέχει υποστήριξη κατά την διαπραγμάτευση του SLA. Στην πραγματικότητα προσφέρει υλοποιήσεις πρωτοκόλλων και λειτουργίες που είναι υπεύθυνες για την διαπραγμάτευση. Η λογική που θα αποφασίσει για τι θεωρείται επιτυχημένη διαπραγμάτευση έγκειται έξω από αυτό.

### POLICY SUBSYSTEM

Πρόκειται για ένα υποσύστημα που έχει σαν κύριο σκοπό να μετατρέψει της γενικές οδηγίες που του παρέχει ο SC σε συγκεκριμένα QoS, που θα μπορέσουν να χρησιμοποιηθούν ευκολότερα από τα υπόλοιπα υποσυστήματα.

### EMS (Execution Management Service)

Πρόκειται για εκείνο το υποσύστημα που είναι υπεύθυνο να εντοπίζει μέσα στον EN, ποια WS που πληρούν κάποιες συγκεκριμένες απαιτήσεις είναι ελεύθερα. Επίσης είναι υπεύθυνο για την προσωρινή δέσμευση των πόρων για συγκεκριμένο χρονικό διάστημα.

ΜΔΕ: “*Επιστήμη Υπολογιστών*”

## BROKER

Είναι το υποσύστημα που θα αναλάβει την εύρεση, διαπραγμάτευση και τέλος την υπογραφή του κατάλληλου συμβολαίου. Στην πραγματικότητα λειτουργεί σαν διεπαφή ανάμεσα στον EN που παίζει τον ρόλο του καταναλωτή και των υπολούπων EN που παίζουν το ρόλο των παρόχων.

## SLA MANAGER

Αυτό το υποσύστημα είναι υπεύθυνο για την διαχείριση των SLA και περιλαμβάνει την διαπραγμάτευση των παραμέτρων, καθώς και την παρακολούθηση και αξιολόγηση τους.

## SLA SIGNER

Αυτό το υποσύστημα είναι υπεύθυνο για την υπογραφή του SLA. Διαφορετικά πρωτόκολλα signing απαιτούν την ύπαρξη διαφορετικών SLA Signer.

## NOTARY

Πρόκειται για ένα TPP, που μεσολαβεί ανάμεσα στον SC και τον SP ώστε να γίνει εφικτή η υπογραφή του SLA. Επίσης Είναι υπεύθυνος για την πιστοποίηση τόσο των ίδιων, όσο και των υπογραφών τους.

## EN SLA REPOSITORY

Πρόκειται για μια Βάση Δεδομένων στην οποία αποθηκεύονται τα SLA που έχουν υπογραφεί από στον εκάστοτε EN ώστε να είναι διαθέσιμα τόσο κατά την φάση του Operation, όσο και μετά τον τερματισμό του VO.

ΜΔΕ: “*Επιστήμη Υπολογιστών*”

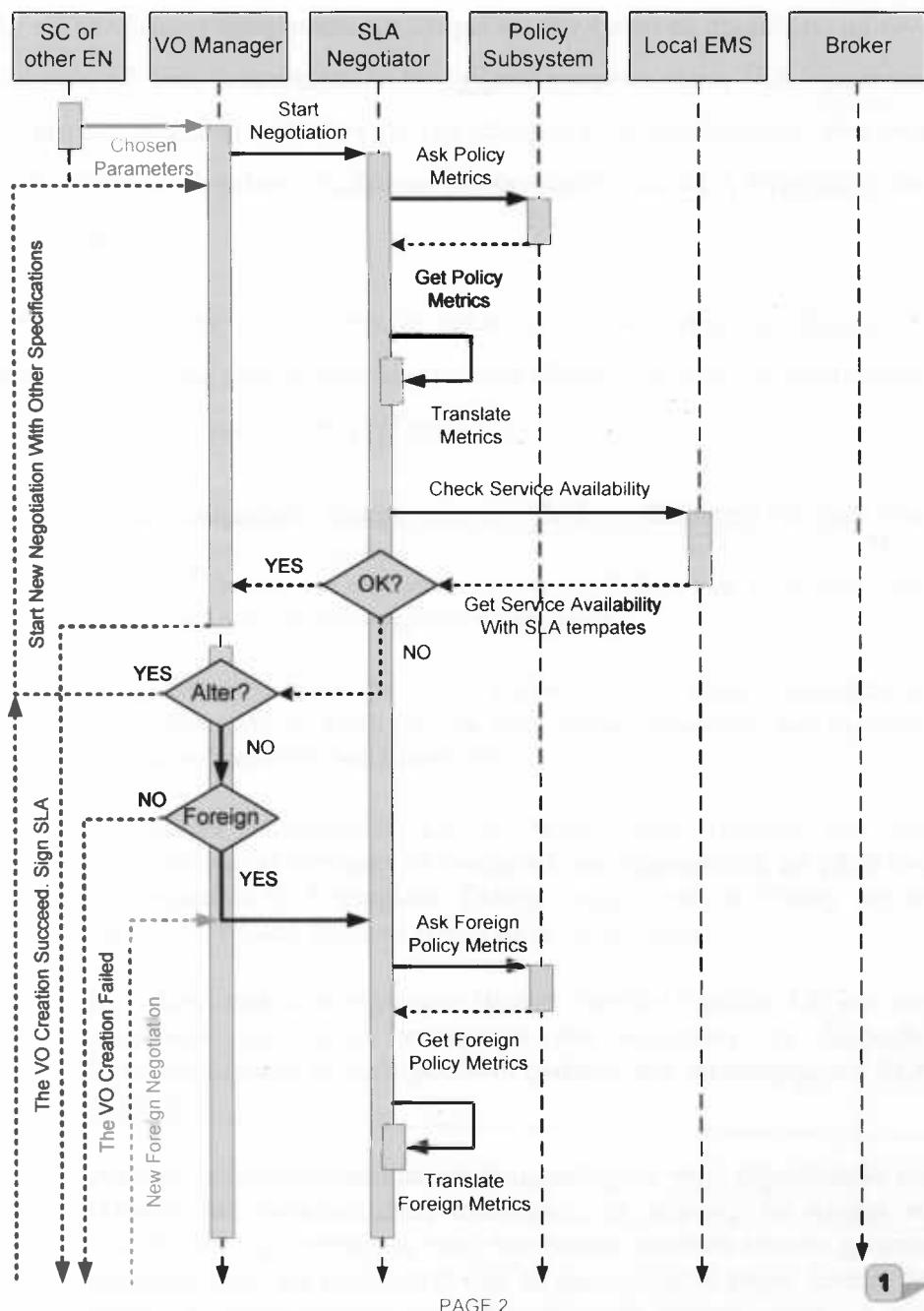
## *Kεφάλαιο 5*

### **ΠΡΩΤΟΚΟΛΛΟ**

Σε αυτό το κεφάλαιο θα περιγράψουμε τον τρόπο με τον οποίο επικοινωνούν όλα τα υποσυστήματα, ώστε να βρεθούν τα κατάλληλα WS και να υπογραφούν τα SLAs.

### **ΔΙΑΔΙΚΑΣΙΑ**

Στην αρχή ο SC ζητάει από κάποιον EN κάποιους πόρους με συγκεκριμένες απαιτήσεις. Ο VO Manager του συγκεκριμένου EN, αναλαμβάνει να ελέγξει αν οι πόροι αυτοί είναι διαθέσιμοι με την βοήθεια του SLA Negotiator. Ο SLA Negotiator, ζητάει από τον Local EMS να ελέγξει την Βάση Δεδομένων του, για το αν υπάρχουν διαθέσιμοι αυτοί οι πόροι και από ποιους SP. Στο τέλος στέλνει μια απάντηση στον VO Manager, για το αν είναι εφικτό να δημιουργηθεί το συμβόλαιο. Ο VO Manager είναι υπεύθυνος να αποφασίσει αν θα γίνει δεκτό το συμβόλαιο, αν θα πρέπει να επαναληφθεί η διαδικασία με διαφορετικές παραμέτρους, ή αν θα πρέπει να αναζητήσει τους πόρους που του ζητήθηκαν από άλλους EN. (*Σχήμα 9*).



PAGE 2

1

Σχήμα 9: Αρχιτεκτονική (α)

## ΜΔΕ: “Επιστήμη Υπολογιστών”

Αν ο VO Manager αποφασίσει ότι μπορεί να υπογράψει το συμβόλαιο με τους τοπικούς SP, τότε το πρωτόκολλο συνεχίζεται ανάμεσα στους SLA Signer των SC και SP, που είναι υπεύθυνη για την υπογραφή του συμβολαίου, μέσο ενός TTP (Notary). Το τελικό συμβόλαιο αποθηκεύεται στο SLA Repository του EN (*Σχήμα 11*).

Αν ο VO Manager αποφασίσει ότι πρέπει να επικοινωνήσει με άλλους EN ώστε να βρει τους πόρους που απαιτούνται (*Σχήμα 10*), τότε το πρωτόκολλο συνεχίζεται όπως φαίνεται στην (*Σχήμα 13*).

1. Ο SLA Negotiator ζητάει από τον Broker<sub>0</sub> (Initiator) να βρει τους απαραίτητους πόρους που θα ικανοποιούν το συγκεκριμένο SLA-template. Επίσης του ανακοινώνει και τον βαθμό του Trust που θέλει να υπάρχει για την συγκεκριμένη εφαρμογή.
2. Ο Initiator ελέγχει με την βοήθεια του Trust Management συστήματος για το ποιοι EN και ποιο Notary από αυτά που γνωρίζει είναι τόσο έμπιστοι όσο απαιτείται.
3. Ο Initiator επικοινωνεί με το Notary που επέλεξε και του ανακοινώνει ότι επιθυμεί να διεξαχθεί μια δημοπρασία, με βάση ένα συγκεκριμένο SLA-template. Επίσης ενημερώνει το Notary για το ποιοι EN - Broker θέλει να συμμετάσχουν σε αυτή.
4. Το Notary, επικοινωνεί με τους Broker (Service Provider SP) που του ζητήθηκαν και τους ενημερώνει ότι πρόκειται να διεξαχθεί δημοπρασία ώστε να καλυφθούν οι ανάγκες του συγκεκριμένου SLA template.
5. Όσοι SP που ενδιαφέρονται να συμμετάσχουν στην δημοπρασία και πληρούν τις συγκεκριμένες απαιτήσεις σε πόρους, θα πρέπει να καταθέσουν τις προσφορές τους στο Notary μέσα σε κάποιο χρονικό διάστημα που έχει καθοριστεί από το προηγούμενο βήμα. Επίσης θα πρέπει να «μαρκάρουν» τους απαραίτητους πόρους, ώστε να μην χρησιμοποιηθούν σε κάποιο άλλο SLA-template.
6. Το Notary αφού συγκεντρώσει όλες τις προσφορές, τις αποστέλλει στον Initiator.

7. Ο Initiator υπολογίζει την προσφορά που θα του προσφέρει τα περισσότερα πλεονεκτήματα, χρησιμοποιώντας τα ακόλουθα μεγέθη, ανάλογα με την στρατηγική που έχει επιλέξει<sup>1</sup>:

- **Trust**

Είναι ο βαθμός εμπιστοσύνης που έχουμε για τον κάθε SP.

- **Offer**

Είναι η προσφορά που έχει καταθέσει ο SP για κάποια συγκεκριμένη παράμετρο του SLA-template.

- **Number of parameters won**

Είναι ο αριθμός των παραμέτρων που έχει κερδίσει ο κάθε SP στο συγκεκριμένο SLA-template. Αυτό θα μπορούσε να χρησιμοποιηθεί από τον Initiator, γιατί πιθανότατα θα θέλει να επικοινωνήσει με λιγότερους SP, ώστε να μειώσουμε τον αριθμό των SLA καθώς και τον χρόνο που απαιτείται για την δημιουργία τους.

8. Επιστρέφεται η κατάλληλη απάντηση στον SLA Negotiator για να αποφασίσει αν θα δεχθεί την προσφορά ή όχι.

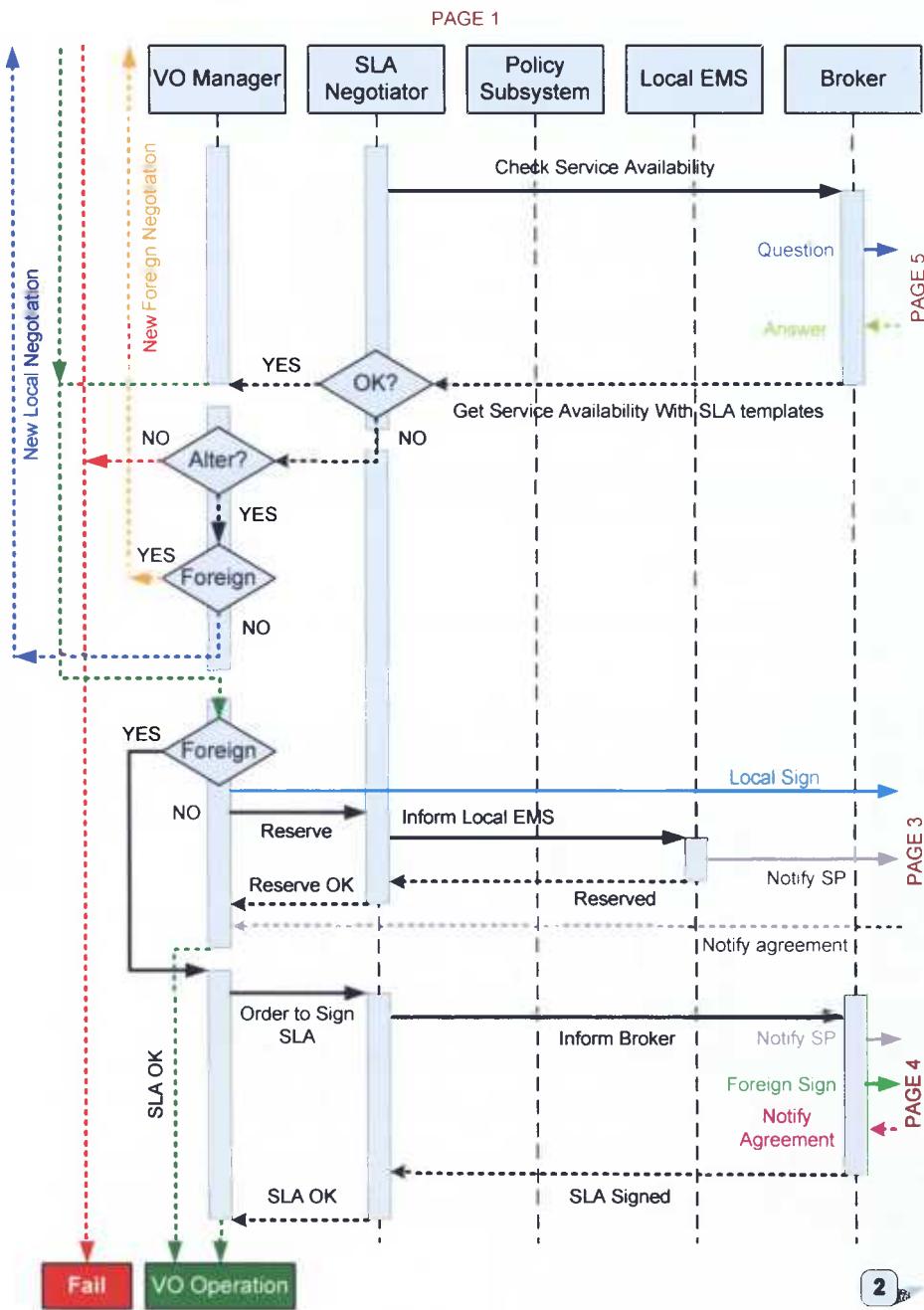
9. Ο SLA Negotiator αποφασίζει αν θα πρέπει να γίνει δεκτή η προσφορά και ενημερώνει τον Broker, που με την σειρά του ενημερώνει το Notary για το αποτέλεσμα.

10. To Notary πληροφορεί όλους τους SPs που συμμετείχαν για το αν έχουν κερδίσει (και άρα πρέπει να υπογράψουν συμβόλαιο) ή αν έχουν χάσει και άρα πρέπει να αποδεσμεύσουν τους πόρους που είχαν κάνει pre-reserved.

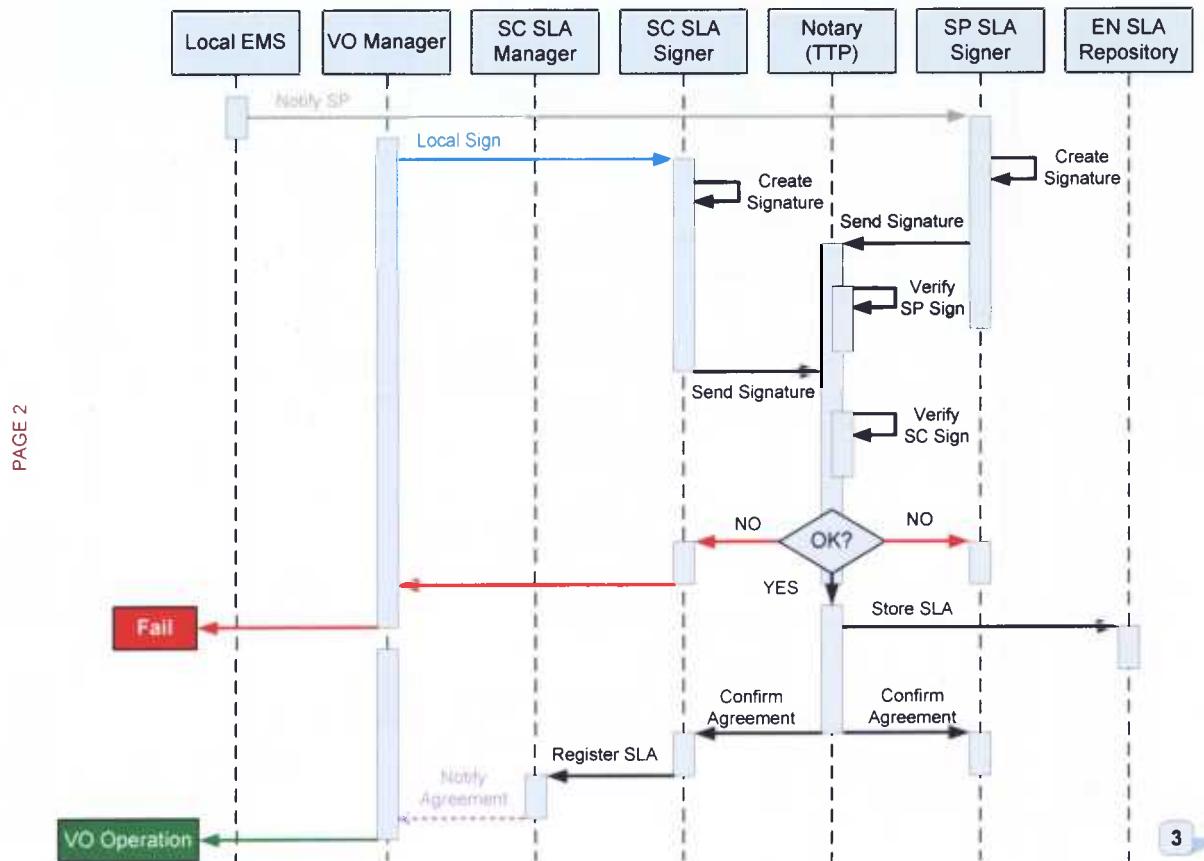
<sup>1</sup> Αν δούμε αυτό το σημείο με ένα πιο αφηρημένο τρόπο, διακρίνουμε εύκολα ότι πρόκειται για Δημοπρασία 1<sup>st</sup> τιμής με σφραγισμένες προσφορές (1<sup>st</sup> price sealed bid) που αναφέρεται όμως σε πολλαπλά αντικείμενα διαφορετικού τύπου [9].

### ΜΔΕ: “Επιστήμη Υπολογιστών”

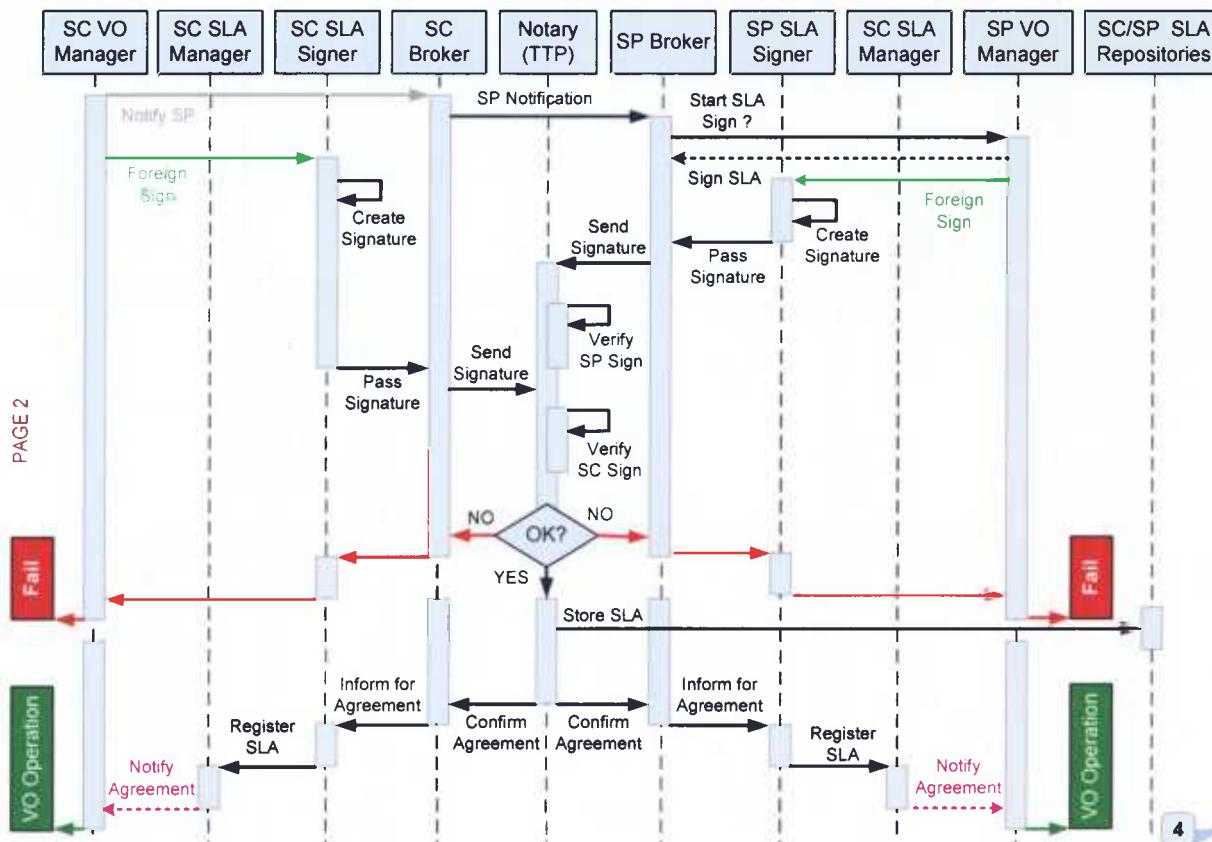
Αν η προσφορά δεν γίνει δεκτή, τότε ενημερώνεται ο SC ότι δεν είναι διαθέσιμοι οι πόροι που ζητάει, διαφορετικά ακολουθείται το πρωτόκολλο που φαίνεται στην (*Σχήμα 12*). Η βασική αρχή που χρησιμοποιείται είναι η ίδια με αυτή της (*Σχήμα 11*), μόνο που τώρα υπεύθυνοι για την υπογραφή του συμβολαίου είναι οι Brokers του SC και των SP. Τα τελικά συμβόλαια, αποθηκεύονται στο Repository του αρχικού EN, καθώς και στα Repositories των υπολοίπων EN που πρόσφεραν πόρους.



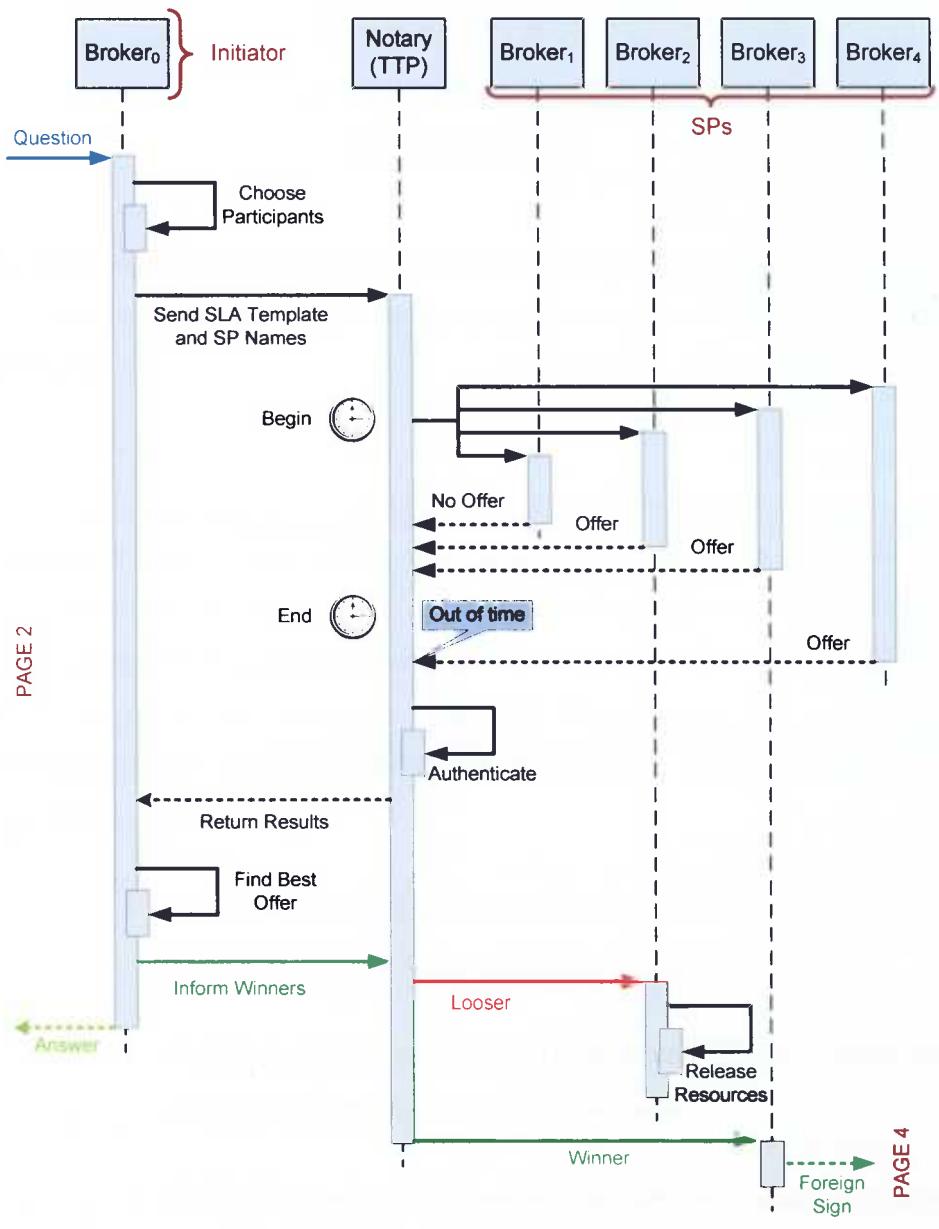
Σχήμα 10: Αρχιτεκτονική (β)



Σχήμα 11: Αρχιτεκτονική (γ)



Σχήμα 12: Αρχιτεκτονική (δ)



Σχήμα 13: Αρχιτεκτονική (ε)

ΜΔΕ: “*Επιστήμη Υπολογιστών*”

## *Kεφάλαιο 6*

### ΥΛΟΠΟΙΗΣΗ

Στα πλαίσια αυτής της Διπλωματικής Εργασίας, αναπτύχθηκε ένα πρόγραμμα το οποίο προσομοιώνει την συμπεριφορά του πρωτοκόλλου που έχουμε προτείνει.

### ΕΙΣΑΓΩΓΗ

Το πρόγραμμα που αναπτύχθηκε, είναι γραμμένο σε C# και έγινε προσπάθεια ώστε να είναι όσο πιο κοντά στην πραγματικότητα γίνεται. Επίσης έγινε εκτενής χρήση Xml, XPath, XQuery καθώς και χρήση Microsoft SQL Server 2005 για την αποθήκευση και ανάκτηση πληροφοριών. Το περιβάλλον ανάπτυξης ήταν το Microsoft Visual Studio 2005. Τα κύρια μέρη του προγράμματος είναι:

- Ένα windows application που προσομοιώνει τον EN (Initiator) που ζητάει πόρους για να καλύψει τις ανάγκες μιας αίτηση
- Από έναν αριθμό Notaries που καθορίζεται από τον χρήστη και είναι στην υλοποιημένα ως Web Services
- Από έναν αριθμό EN (Service Providers) που είναι αυτοί που θα προσφέρουν στον Initiator τις υπηρεσίες που αυτός επιθυμεί.

ΜΔΕ: “*Επιστήμη Υπολογιστών*”

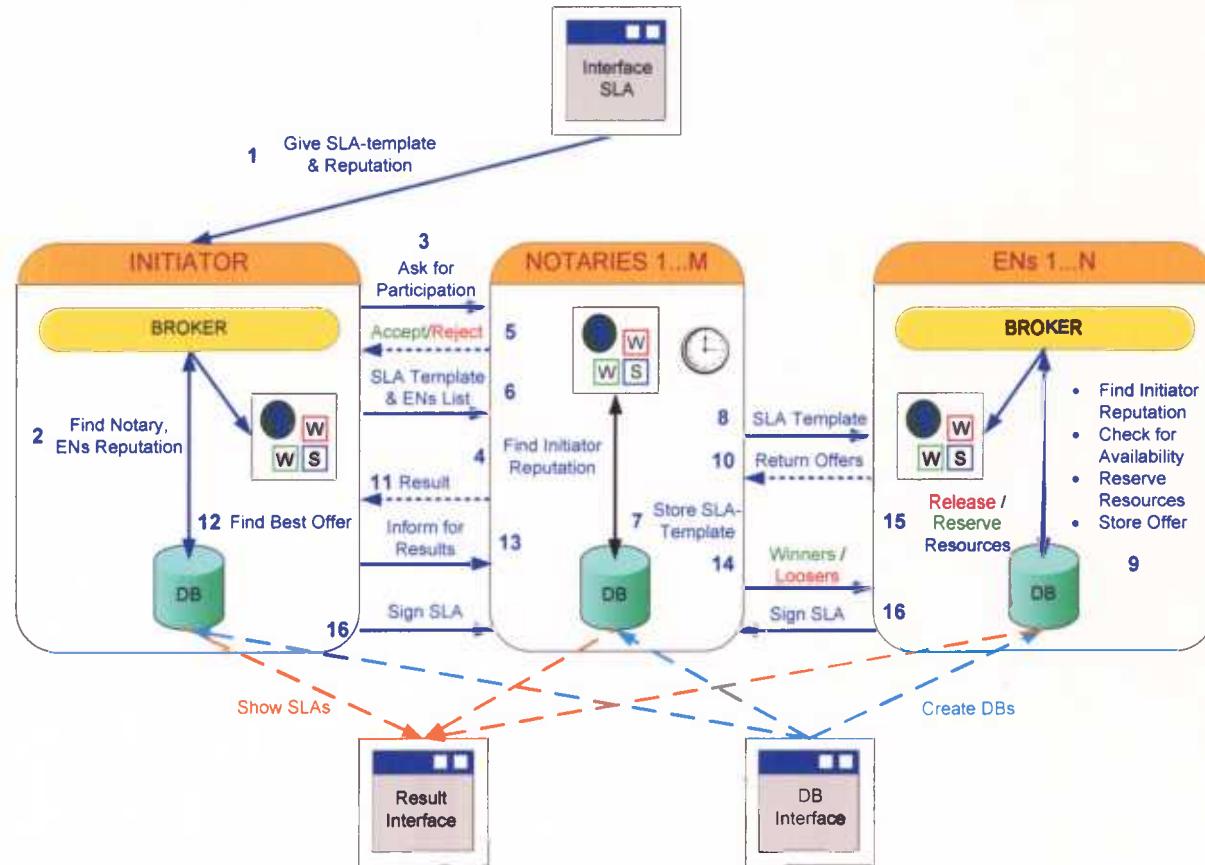
- Υπάρχουν επίσης 3 Interfaces, κάθε ένα από τα οποία είναι υπεύθυνο για την εισαγωγή ή προβολή απαραίτητων πληροφοριών

## ΑΡΧΙΤΕΚΤΟΝΙΚΗ

Η αρχιτεκτονική που ακολουθήσαμε είναι αυτή που φαίνεται στο (*Σχήμα 14*). Κάθε EN καθώς και κάθε Notary αποτελείται από την δικιά του Βάση Δεδομένων στην οποία αποθηκεύει τις απαραίτητες πληροφορίες και από ένα Web Service που είναι άμεσα προσβάσιμο μέσω δικτύου.

## INITIATOR WINDOWS APPLICATION

Πρόκειται για ένα windows application, σκοπός του οποίου είναι να διαβάσει τις πληροφορίες που του δίνει ο χρήστης και αφού δεσμεύσει τους πόρους που έχει διαθέσιμους να δημιουργήσει ένα SLA-template με τους υπόλοιπους. Επίσης πρέπει να ελέγξει στην Βάση Δεδομένων του για να βρει με ποιους EN αλλά και ποιο Notary θα πρέπει να επικοινωνήσει με βάση το Reputation που έχει γι' αυτά. Ακόμη είναι υπεύθυνο για τον έλεγχο των αποτελεσμάτων που θα του στείλει το Notary και για την εύρεση της καλύτερης δυνατής προσφοράς. Στο τελικό στάδιο θα πρέπει να ζητήσει από το Notary να ενημερώσει τους EN, για το ποιος κέρδισε και άρα πρέπει να ξεκινήσει την διαδικασία υπογραφής συμβολαίου, αλλά και για το ποιος έχασε και πρέπει να απελευθερώσει τους πόρους που έκανε pre-reserved.



Σχήμα 14: Αρχιτεκτονική προγράμματος

## NOTARY WEB SERVICE

Αυτό το Web Service είναι ο ενδιάμεσος ανάμεσα στον Initiator και τους άλλους EN. Για να επιτύχει αυτές τις υποχρεώσεις έχει τρία Web Methods που καλούνται μόνο από τον Initiator:

- Accept

To Notary ελέγχει το reputation που έχει για τον Initiator. Αν αυτό είναι κάτω από κάποιο όριο, τότε του αρνείται την συμμετοχή αλλιώς τον δέχεται.

- Ask

Μέσω αυτού του Web Method ο Initiator ζητάει από το Notary να επικοινωνήσει με την λίστα των EN που του αποστέλλει και να τους ζητήσει προσφορές για το συγκεκριμένο SLA-template.

- releaseResources

Μέσω αυτού του Web Method ο Initiator ζητάει από το Notary να επικοινωνήσει με την λίστα των EN που του αποστέλλει και να τους ζητήσει να απελευθερώσουν τους πόρους που είχαν κάνει pre-reserved. Αυτό μπορεί να συμβεί είτε γιατί δεν έγινε δεκτή η προσφορά, είτε γιατί δεν μπόρεσε να δημιουργηθεί ο VO.

## EN WEB SERVICE

Αυτό το Web Service είναι υπεύθυνο για τον έλεγχο και την κατάθεση προσφορών. Για να το επιτύχει αυτό έχει δύο Web Methods που καλούνται μόνο από τα Notary:

- Ask

Σκοπός του είναι να κοιτάξει στην Βάση Δεδομένων του συγκεκριμένου EN και να βρει αν υπάρχουν διαθέσιμοι πόροι και αν μπορεί και επιθυμεί να κάνει προσφορά. Αν δεν μπορεί τότε επιστρέφει ένα κενό xml αλλιώς αποστέλλει την προσφορά μέσω xml αρχείου. Οι πόροι που προτείνονται στην προσφορά γίνονται pre-reserved, ώστε να μην μπορούν να χρησιμοποιηθούν σε κάποια άλλη αίτηση.

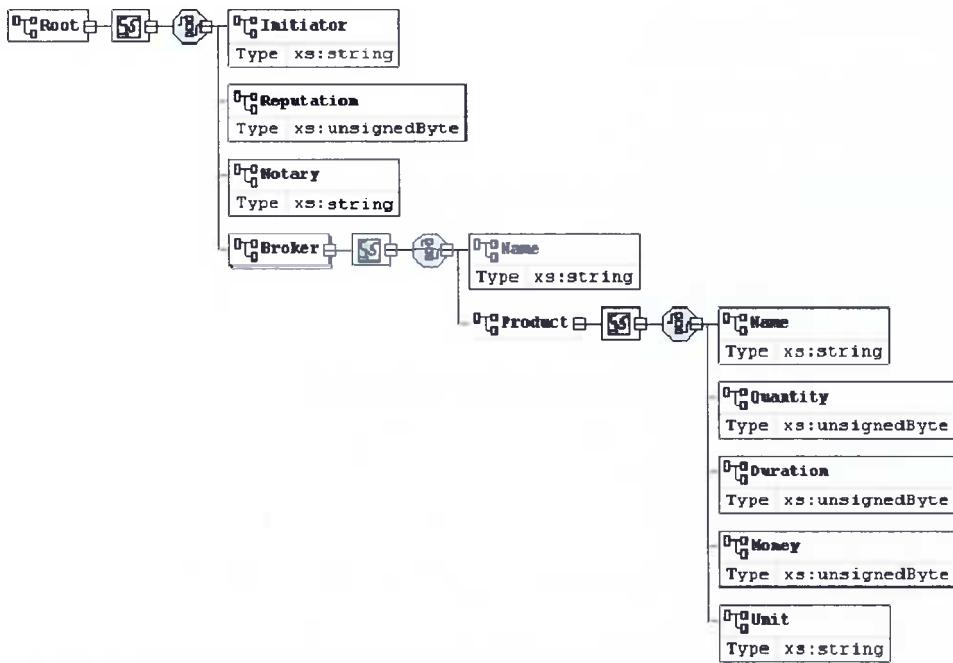
- releaseResources

Σκοπός του είναι η απελευθέρωση συγκεκριμένων πόρων που είχαν γίνει στο προηγούμενο βήμα pre-reserved. Αυτό μπορεί να συμβεί είτε γιατί δεν έγινε δεκτή η προσφορά, είτε γιατί δεν μπόρεσε να δημιουργηθεί ο VO.



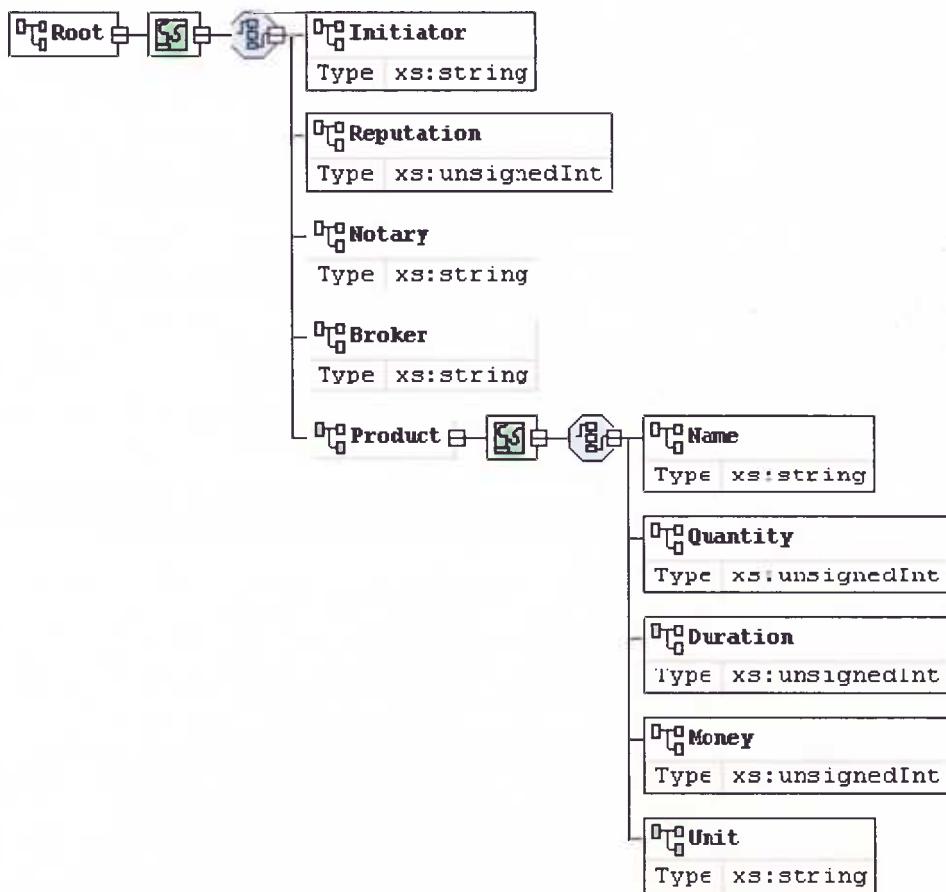
## XML DOCUMENTS

Για την επικοινωνία μεταξύ των Web Services και του Initiator χρησιμοποιήθηκαν Xml Document. Τα αρχεία αυτά είναι σύμφωνα με δύο Xml Schemas πράγμα το οποίο μας δίνει αρκετά πλεονεκτήματα όπως την ευκολία εντοπισμού λάθους ή αλλαγής στα μηνύματα που μεταφέρονται, την βελτιστοποίηση που παρουσιάζει ο Microsoft SQL Server 2005, την ευκολότερη συγγραφή κώδικα κτλ. Παρακάτω παρουσιάζουμε γραφικά τα Xml σχήματα που χρησιμοποιήσαμε:



Σχήμα 15: Xml Schema (a)

ΜΔΕ: “Επιστήμη Υπολογιστών”



Σχήμα 16: Xml Schema (b)

## DATABASE INTERFACE

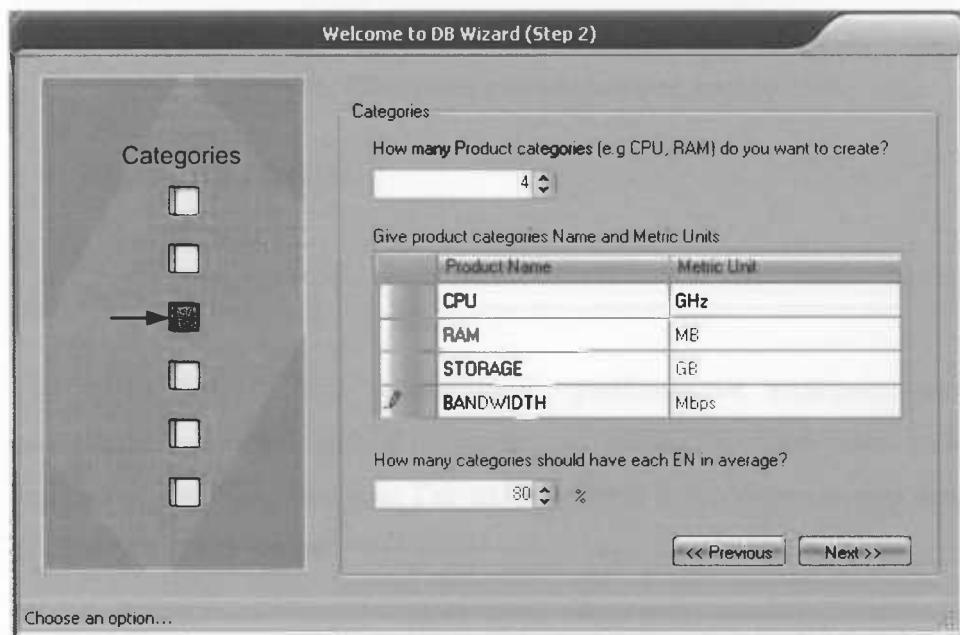
Για την προσομοίωση τις λειτουργίας χρειαζόμασταν ένα τρόπο για την δημιουργία πολλαπλών Βάσεων Δεδομένων. Μία για κάθε EN και Notary που θα μας ζητούσε ο χρήστης. Επίσης θα έπρεπε να δημιουργηθούν πίνακες διαφορετικοί σε κάθε EN (ανάλογα με τα προϊόντα που θα είχε στην διάθεση του ο καθένας). Τέλος οι πίνακες αυτοί θα έπρεπε να γεμίσουν με τυχαία στοιχεία. Για να ξεπεραστεί αυτό το εμπόδιο δημιουργήθηκε ένας Database Wizard σκοπός του οποίου είναι η συγκέντρωση πληροφοριών από τον χρήστη για το πώς επιθυμεί να είναι οι ΒΔ που θα δημιουργηθούν. Οι ρυθμίσεις που μπορεί να κάνει ο χρήστης είναι οι ακόλουθες:

- Ο αριθμός των EN που επιθυμεί να δημιουργηθούν
- Ο αριθμός των Notaries που επιθυμεί να δημιουργηθούν
- Το όνομα που θα έχουν οι EN αλλά και το όνομα που θα έχει η ΒΔ
- Τον αριθμό, το είδος και την μονάδα μέτρησης των προϊόντων.
- Τον αριθμό των προϊόντων που θα έχει κάθε EN στην διάθεσή του
- Την ελάχιστη και μέγιστη τιμή των διαστημάτων που θα έχει κάθε προϊόν και η τιμολόγηση του κάθε διαστήματος χωριστά
- Τον αριθμό των διαστημάτων που θα έχει ο κάθε EN στην διάθεσή του
- Το ποσοστό που ρυθμίζει κατά πόσο θα διαφέρουν οι τιμές μεταξύ διαφορετικών EN.
- Την μεγαλύτερη και μικρότερη τιμή της διαθεσιμότητας

ΜΔΕ: “Επιστήμη Υπολογιστών”

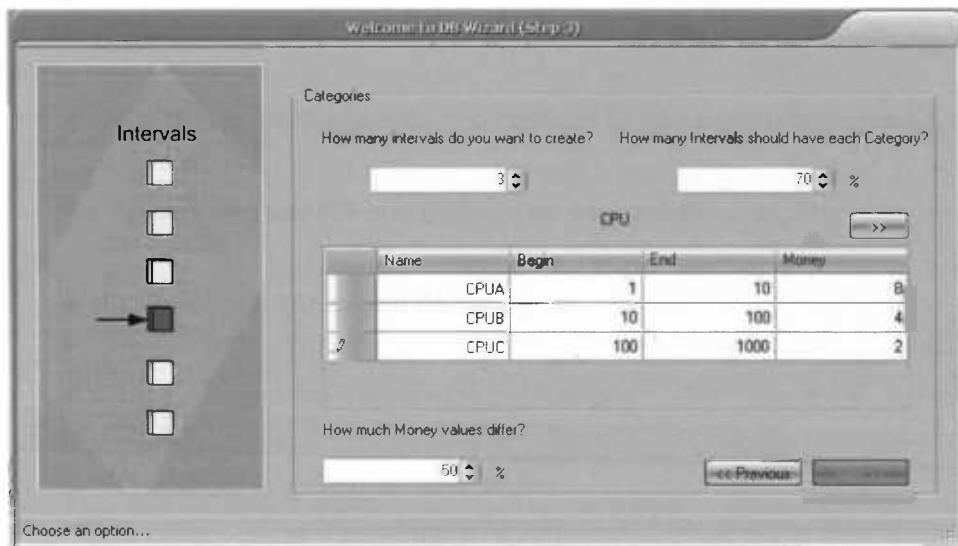
- Την μεγαλύτερη και μικρότερη τιμή Reputation
- Τα ονόματα του Data File και του Log File, τα μεγέθη που θα έχουν, τους ρυθμούς αύξησης τους καθώς και τα μέρη που πρέπει να αποθηκευτούν

Τέλος προσφέρεται και η δυνατότητα μέσω φόρμας του ελέγχου των ΒΔ που υπάρχουν στο σύστημα και η διαγραφή οποιασδήποτε από αυτές επιθυμούμε. Τα ακόλουθα *Σχήματα* μας δείχνουν ενδεικτικά κάποιες φόρμες εισαγωγής στοιχείων.



**Σχήμα 17: Εισαγωγή Προϊόντων στον DB Wizard**

## ΜΔΕ: “Επιστήμη Υπολογιστών”



**Σχήμα 18: Εισαγωγή κατηγοριών σε προϊόν**

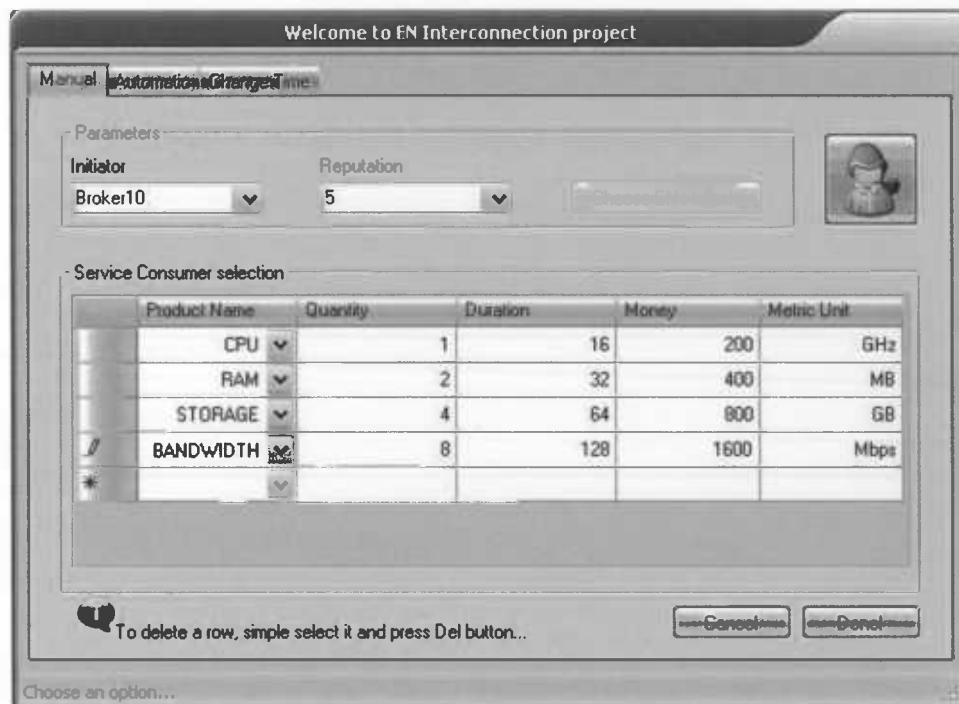
## SLA INTERFACE

Για την λειτουργία του πρωτοκόλλου χρειάζεται να παρέχουμε στο πρόγραμμα μια σειρά από πληροφορίες, οι οποίες θα βοηθήσουν στην δημιουργία του SLA Template. Για τον λόγο αυτό έχουν δημιουργηθεί ένας αυτόματος και ένα manual.

### Manual

Με αυτό τον τρόπο ο χρήστης έχει την δυνατότητα να εισάγει οποιεσδήποτε παραμέτρους επιθυμεί για την δημιουργία του SLA-Template (Σχήμα 19). Γενικά μπορεί να διαλέξει:

- Ποιος EN θα είναι ο Initiator
- Τον βαθμό του Reputation που επιθυμεί να έχει το συγκεκριμένο SLA
- Τα ονόματα των EN που επιθυμεί να συμμετάσχουν ως SPs
- Το είδος των πόρων που θα υπάρχουν
- Την ποσότητα κάθε προϊόντος ξεχωριστά
- Την διάρκεια χρήση του κάθε πόρου
- Την μέγιστη τιμή χρημάτων που επιθυμεί να δώσει για την χρήση του συγκεκριμένου πόρου για την συγκεκριμένη διάρκεια

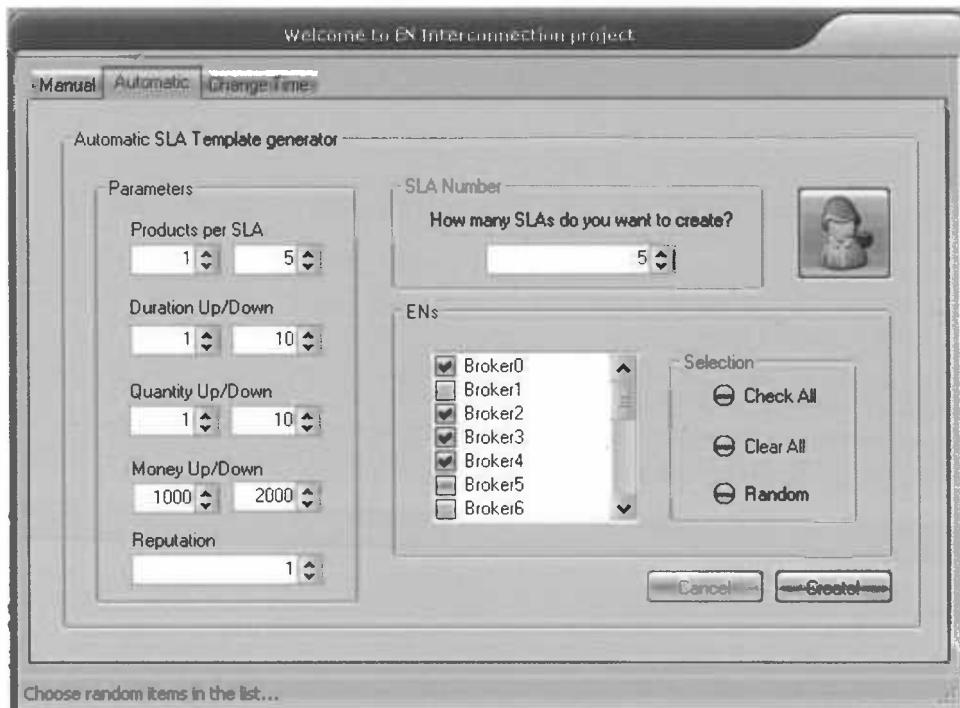


Σχήμα 19: Δημιουργία SLA-Template

## Αυτόματος

Με αυτό τον τρόπο ο χρήστης έχει την δυνατότητα να εισάγει κάποιες γενικές πληροφορίες και στην συνέχεια να αφήσει το πρόγραμμα να δημιουργήσει όσα SLA-template του ζητηθούν (*Σχήμα 20*). Γενικά μπορεί να διαλέξει:

- Το ανώτερο και κατώτερο αριθμό προϊόντων που πρέπει να έχει κάθε SLA-template
- Την μέγιστη και ελάχιστη τιμή της διάρκειας χρήσης ενός πόρου
- Την μέγιστη και ελάχιστη τιμή της ποσότητας χρήσης ενός πόρου
- Την ελάχιστη τιμή του Reputation
- Τον αριθμό των SLA-templates που θα δημιουργηθούν
- Τα ονόματα των EN που επιθυμεί να συμμετάσχουν ως SPs



**Σχήμα 20: Δημιουργία πολλών SLA-Template με αυτόματο τρόπο**

### Time

Αν το πρόγραμμα υποστήριζε μόνο τα παραπάνω στοιχεία, τότε μετά από την δημιουργία κάποιου αριθμού SLA-templates οι πόροι που υπάρχουν στο σύστημα θα τελείωναν. Αυτό θα συνέβαινε γιατί κανείς δεν τους έχει απελευθερώσει. Για να απελευθερωθούν όμως θα έπρεπε να περάσει η χρονική περίοδος δέσμευσης τους. Άρα θα έπρεπε με κάποιο τρόπο να προσομοιωθεί ο χρόνος. Έτσι κάθε φορά που δημιουργείται ένα manual συμβόλαιο, ή κάθε φορά που δημιουργούνται μια ομάδα συμβολαίων με αυτόματο τρόπο, θεωρούμε ότι έχει περάσει μία ώρα. Επίσης έχει δημιουργηθεί ένα interface στο οποίο μπορούμε να δηλώσουμε τον αριθμό των ορών που επιθυμούμε να “περάσουν” (Σχήμα 21)

ΜΔΕ: “Επιστήμη Υπολογιστών”



**Σχήμα 21: Διεπαφή αλλαγής ώρας**

Και στις τρεις παραπάνω περιπτώσεις ενημερώνονται όλοι οι EN ότι έχει περάσει κάποιο χρονικό διάστημα. Στην συνέχεια, κάθε ένας από αυτούς ελέγχει την Βάση Δεδομένων του για να δει ποιοι πόροι πρέπει να ελευθερωθούν και ποια συμβόλαια πρέπει να θεωρήσει ότι έχουν λήξει.

## RESULT INTERFACE

Πρόκειται για ένα interface που έχει ως κύριο σκοπό να δείξει των αριθμό των συμβολαίων που έχουν γραφεί σε κάθε EN καθώς και τις παραμέτρους που αυτά φέρουν. Ο χρήστης το μόνο που έχει να κάνει είναι να επιλέξει τον EN που τον ενδιαφέρει και στην συνέχεια από την λίστα που θα εμφανιστεί να διαλέξει το κατάλληλο συμβόλαιο.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Lisbon European Council 23 and 24 March 2000 presidency conclusions - A Strategic Goal for the Next Decade. (Last accessed at 22 June 2007) [http://www.europarl.europa.eu/summits/lis1\\_en.htm](http://www.europarl.europa.eu/summits/lis1_en.htm)
- [2] William E. Johnston, Sumit Talwar, Keith R. Jackson: ‘Overview of Security Considerations for Computational and Data Grids’ In: Proc 10th IEEE International Symposium on High Performance Distributed Computing. San Francisco, CA, USA:IEEE Computer Society, 2001.439-440
- [3] I. Foster, C. Kesselman, S. Tuecke “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”, International journal of Supercomputer Applications and High Performance Computing, 2001, 200-222
- [4] Michael Wilson et al. “The TrustCoM approach to enforcing agreements between interoperating enterprises”
- [5] Theo Dimitrakos (2003) A Service-Oriented Trust Management Framework, Proc. International Workshop on Trust in Agent Societies, Melbourne, Australia, 14-15 Jul 2003, Trust, Reputation and Security: Theories and Practice.
- [6] Lavanya Ramakrishnan. “Securing Next-Generation Grids”. IEEE IT Pro, March — April 2004. IEEE Computer Society, 2004
- [7] C. Geuer-Pollmann, J. Claessens. Web Services and Web Service security standards. Information Security Technical report, Elsevier, 10(1)15:24, 2005.
- [8] TrustCoM project website. (Last accessed at 22 June 2007) <http://www.eu-trustcom.com/>
- [9] Harry J. Paarsch, Jacques Robert. Testing Equilibrium Behaviour At First-Price, Sealed-Bid Auctions With Discrete Bid Increments. 2003

## ΕΥΡΕΤΗΡΙΟ

### API

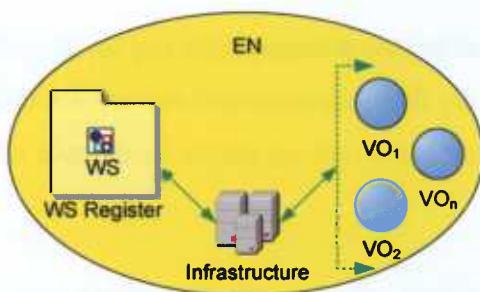
To Application Programming Interface είναι μια διεπαφή πηγαίου κώδικα, που παρέχεται από ένα υπολογιστικό σύστημα ή μια βιβλιοθήκη προγραμμάτων με σκοπό να εξυπηρετήσει αιτήσεις υπηρεσιών άλλων υπολογιστικών προγραμμάτων.

### CERTIFIED AUTHORITY

Η Certificate Authority ή Certification Authority (CA) είναι μια οντότητα που εκδίδει ψηφιακά πιστοποιητικά που χρησιμοποιούνται από άλλους. Είναι ένα παράδειγμα trusted third party.

### ENTERPRISE NETWORK

Πρόκειται για έναν οργανισμό που έχει εγκαταστήσει την κατάλληλη υποδομή, ώστε να μπορεί να δεχθεί την εγγραφή νέων WS από διαφορετικούς ASP, να μπορεί να δημιουργήσει VO για την επύλυση κάποιου συγκεκριμένου προβλήματος που θα του τεθεί από κάποιο καταναλωτή (SC), καθώς και να μπορέσει να διευθετήσει όλες εκείνες τις λεπτομέρειες που απαιτούνται στον κύκλο ζωής του VO (*Σχήμα 22*).



Σχήμα 22: Αφηρημένη δομή EN

## **SLA**

To Service Level Agreement είναι μια ηλεκτρονική μορφή συμβολαίου, που έχει περιορισμένη νομική επίδραση. Περιγράφει τα Quality of Service (QoS) που πρέπει να εφαρμοστούν.

## **SOAP**

To Originally Simple Object Access Protocol είναι ένα πρωτόκολλο για ανταλλαγή μηνυμάτων βασισμένα σε XML πάνω από δίκτυα υπολογιστών που τυπικά χρησιμοποιούν HTTP. Το SOAP υλοποιεί την βάση της στοίβας των Web Services και παρέχει ένα framework, πάνω στο οποίο μπορούν να στηριχθούν πιο αφηρημένα επίπεδα.

## **TRUST**

Ο Dimitrakos [5] έχει ορίσει το Trust σαν μια διαδικασία όπου ο Α ζητάει μια υπηρεσία Χ από τον πάροχο Β ως εξής:

“Το Trust του μέλους Α για κάποιο μέλος Β για μια συγκεκριμένη υπηρεσία Χ, είναι ο βαθμός της πεποίθησης που έχει ο Α για την συμπεριφορά του Β για ένα συγκεκριμένο χρονικό διάστημα και πλαίσιο σε σχέση με την υπηρεσία Χ.”

- Ένα μέλος μπορεί να είναι μια ατομική οντότητα, μια ομάδα ανθρώπων ή διαδικασιών ή ένα σύστημα (προφανώς ο trustor πρέπει να είναι μια οντότητα που μπορεί να αποκτήσει κάποιο βαθμό αξιοπιστίας).

ΜΔΕ: “*Επιστήμη Υπολογιστών*”

- Ο όρος υπηρεσία χρησιμοποιείται σκόπιμα με μια πιο γενική έννοια για να συμπεριλάβει δοσοληψίες, υποδείξεις, εκδόσεις πιστοποιητικών, ασφάλιση κ.α.
- Το συγκεκριμένο χρονικό διάστημα μπορεί να αναφέρεται στο παρελθόν, κατά την διάρκεια που παρέχεται η υπηρεσία, στο μέλλον (σε ένα προγραμματισμένο ή προβλέψιμο χρονικό διάστημα), ή ακόμη και πάντα.
- Η αξιοπιστία χρησιμοποιείται με γενικό τρόπο ώστε να συμπεριλάβει θέματα ασφάλειας, αξιοπιστίας, επικαιρότητας και συντηρησιμότητας.
- Ο όρος πλαίσιο αναφέρεται στα σχετικά συμβόλαιο παροχής υπηρεσιών, το ιστορικό της υπηρεσίας, στην τεχνολογική υποδομή, σε νομοθετικά και ρυθμιστικά πλαίσια που μπορεί να υπάρχουν.
- Το Trust μπορεί να περιλαμβάνει αντικειμενικές πληροφορίες με υποκειμενική γνώμη, που δημιουργείται με βάση πραγματικά γεγονότα και προτάσεις από μια μεσιτική αρχή.
- Το Trust επιτρέπει σε έναν παράγοντα να βασίζεται για κάποιο χρονικό διάστημα στην συμπεριφορά, ή στις πληροφορίες που έχει συγκεντρώσει από άλλους παράγοντες. Η τιμή της εξαρτάται από στην υποκειμενική πιθανότητα ότι ένας παράγοντας θα εκτελέσει μια συγκεκριμένη ενέργεια (που ο trustor δεν θα μπορεί να ανιχνεύσει) μέσα σε ένα πλαίσιο που επηρεάζει τις ενέργειες του trustor.

## TRUSTED THIRD PARTY

Με τον όρο Trusted Third Party (TTP) ονομάζουμε εκείνη την οντότητα που βοηθάει στις αλληλεπιδράσεις μεταξύ δύο άλλων μερών που της έχουν εμπιστοσύνη. Για παράδειγμα στην κρυπτογραφία το ρόλο του TTP τον παίζει μία Certified Authority.

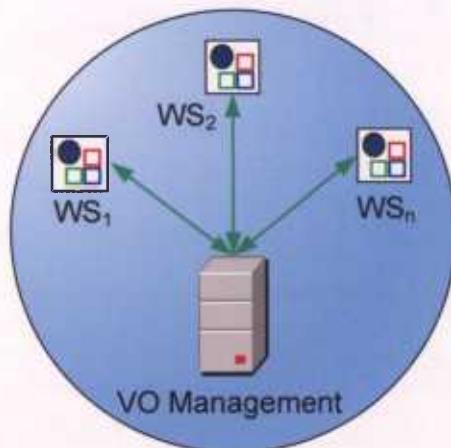
Υπάρχουν πέντε στάδια στον κύκλο ζωής ενός Εικονικού Οργανισμού, όπως έχουν οριστεί στα πλαίσια του TrustCom project, που διαφοροποιούν με λογικό τρόπο τις ενέργειες που πρέπει να γίνουν σε καθένα.

## VIRTUAL ORGANIZATION

Σαν Εικονικό Οργανισμό, ορίζουμε ένα σύνολο (νόμιμων) ανεξάρτητων επιχειρησιακών οντοτήτων (Application Service Providers), που μοιράζονται τους πόρους και τις δεξιότητές τους με συντονισμένο τρόπο, ώστε να επιτύχουν ένα κοινό στόχο. Η επικοινωνία και ο συντονισμός μεταξύ των μελών του VO, γίνεται κυρίως μέσω δικτύων υπολογιστών. Ένα VO είναι μια μορφή Collaborative Network. Ο συντονισμός και η συνεργασία, είναι γνωστή στο χώρο των WS σαν η χορογραφία (choreography) του VO.

Σε ένα υψηλότερο επίπεδο, μπορούμε να πούμε ότι ένας VO είναι μία συντονισμένη συνεργασία μεταξύ ανεξάρτητων παρόχων WS. (*Σχήμα 23*)

Στην εργασία αυτή, υποθέτουμε ότι οτιδήποτε μπορεί, ή είναι συνδεδεμένο με έναν υπολογιστή μπορεί να συμμετάσχει στην δημιουργία ενός VO, επικοινωνώντας μέσω WS.



Σχήμα 23: Αφηρημένη δομή VO

## WSDL

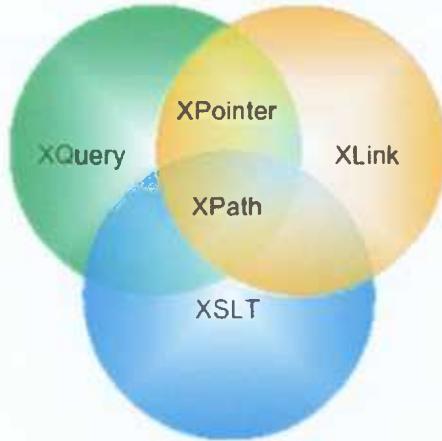
Η Web Services Description Language είναι μια γλώσσα βασισμένη σε XML που παρέχει ένα μοντέλο περιγραφής των Web Services.

## XPath (XML Path Language)

Πρόκειται για μια περιγραφική γλώσσα που χρησιμοποιείτε για την διεύθυνσιοδότηση τμημάτων Xml αρχείων, τον υπολογισμό τιμών (αλφαριθμητικών, αριθμών ή Boolean τιμών) βασισμένων στα περιεχόμενα το Xml εγγράφου.

Η XPath γλώσσα, είναι βασισμένη πάνω στην δενδρική αναπαράσταση του Xml εγγράφου και παρέχει την δυνατότητα της πλοήγησης σε αυτό, επιλέγοντας κόμβους με βάση μια σειρά κριτηρίων.

Το XPath είναι ένα βασικό στοιχείο στο W3C XSLT standard και το XQuery και XPointer στηρίζονται πάνω σε αυτό.



## XQuery (XML Query Language)

Πρόκειται για μια γλώσσα ερωτήσεων (με κάποια προγραμματιστικά στοιχεία), Που έχει σχεδιαστεί να ρωτάει συλλογές από έγγραφα Xml. Σημασιολογικά είναι ίδια με την SQL.

