ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υλοποίηση Γραφικού Περιβάλλοντος για το Irmos Project

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αντρέας, Ι. Γκριμπαβιώτης

Επιβλέπων : Θεοδώρα Βαρβαρίγου,

Καθηγητρια Ε.Μ.Π.

Αθήνα, Μάρτιος 2010

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υλοποίηση Γραφικού Περιβάλλοντος για το Irmos Project

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αντρέας, Ι. Γκριμπαβιώτης

Επιβλέπων : Θεοδώρα Βαρβαρίγου

Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 9 Μαρτίου 2010

.....	.....	.....
Θεοδώρα Βαρβαρίγου	Συμεών Παπαβασιλείου	Βασίλειος Λούμος
Καθηγήτρια Ε.Μ.Π.	Αν. Καθηγητης Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2010

.....

Αντρέας ,Ι. Γκριμπαβιώτης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Ευχαριστίες

Με την ολοκήρωση της διπλωματικής αυτής εργασίας αισθάνομαι την ανάγκη να ευχαριστίσω τους ανθρώπους που μου στάθηκαν καθ'όλη την διάρκεια των σπουδών μου.Ένα ιδιαίτερο ευχαριστώ πηγαίνει προς την οικογένεια μου – τους γονείς και τον αδερφό μου - για την ηθική και όχι μόνο υποστήριξη και συμπραρασταση όλα αυτά τα χρόνια.

Επίσης θέλω να ευχαριστήσω την επιβλέπουσα καθηγήτρια κ. Θεοδώρα Βαρβαρίγου για την ευκαιρία που μου εδωσε να συνεργαστώ μαζί της. Επίσης τον υποψήφιο διδακτωρα Γρηγόρη Κατσαρό για τις χρήσιμες συμβουλές , τον χρόνο που αφιέρωσε και την συνεχή υποστήριξη και καθοδήγηση καθ'ολη την διάρκεια εκπονησης της διπλωματικής εργασίας καθώς και την υποψήφια διδακτωρα Κλεοπάτρα Κωνσταντέλη για την συνεργασία και την συμβολή της.

Περίληψη

Στόχος της παρούσας διπλωματικής εργασίας είναι η περιγραφή , η ανάλυση , η σχεδίαση και τελικά η υλοποίηση ενός γραφικού περιβάλλοντος διεπαφής χρήστη για την ευκολη διεκπεραίωση των λειτουργιών που περιγράφονται από το έργο Irmos. Για να γίνουν πλήρως κατανοητές οι έννοιες πίσω από το έργο Irmos πρώτα ορίζονται και περιγράφονται οι τεχνολογίες που το απαρτίζουν. Γίνεται αναφορά και ορίζεται το Grid(Υπολογιστικό Πλέγμα) καθώς και τα στοιχεία που το αποτελούν.Παρουσιάζονται συνοπτικά συστήματα Grid που λειτουργούν σήμερα. Επίσης αναλύονται οι Υπηρεσίες Διαδικτύου και οι λειτουργίες που προσφέρουν και γιατί είναι τόσο σημαντικές για το Grid. Όμως οι απλές Υπηρεσίες διαδικτύου κρίνονται ανεπαρκείς για την χρήση με εφαρμογές Grid και για τον λόγο αυτό δημιουργήθηκε το WSRF , που δίνει στις υπηρεσίες αυτές επιπλέον δυνατότητες και βελτιώνει αρκετές από τις παραμέτρους τους.Αναλύετε το έργο Irmos και οι καινοτομίες που προσφέρει στο χώρο των Grid εφαρμογών όπως η εκτελεση εφαρμογών πραγματικού χρόνου πανω σε μια Υποδομη προσανατολισμένη στην Υπηρεσία και η εφασφάλιση της Ποιότητας της Υπηρεσίας. Στο δευτερο μέρος της διπλωματικής αυτής παρουσιάζεται η εφαρμογή που αναπτύχθηκε για το έργο Irmos. Αναλύεται και σχεδιάζετε η εφαρμογή με χρήση διαγραμμάτων περιπτώσεων χρήσης και ακολουθιακών διαγραμμάτων. Στην συνέχεια παρουσιάζονται οι τεχνολογίες πάνω στις οποίες έγινε η υλοποίηση και δικαιολογείτε η χρήση τους. Τέλος και πιο σημαντικα αναλύεται η ίδια η υλοποίηση της εφαρμογής παρουσιάζονται και επεξηγώντας τα κυριότερα σημεία του κώδικα που την αποτελούν.

Λέξεις Κλειδιά: Πλεγμα , OGSA , Υπηρεσίες Διαδικτύου , WSRF ,GLOBUS , Έργο Irmos , Πραγματικός χρόνος , Υποδομή προσανατολισμένη στην Υπηρεσία, SOI, Java , γραφικό περιβάλλον , JSP , HTML , CSS , Apache Tomcat.

Abstract

The objective of this diplwma thesis is the description, analysis, design and implementation of a graphic user interface environment for the conduct of operations described by the Irmos project. In order for the concepts behind the project Irmos to be fully understood first the composed technologies are defined and described. Grid Computing is defined along with the components it consists of. Also elaborates on the Web Services and their offered functions and why they are so important for Grid Computing. But the simple Web services appear inadequate for use with grid applications, and that's the reason behind the creation of WSRF, which gives extra features and enhances many of their parameters. Consequently the project Irmos is analyzed and its innovations in the area of Grid applications such as real-time execution of applications in a Service-oriented Infrastructure and assurances of Quality of Service. In the second part of the diplwma thesis the actual application for the Irmos project is developed. The application is analyzed and designed using use cases and sequence diagrams. Then we discuss the technologies on which it was implemented and justify their use. Finally, and most important is the completion of the application and main portions of the source code are presented and explained.

Keywords: Grid , OGSA , Web Services , WSRF , Web Services Resource Framework , GLOBUS , Irmos Project , Real time , Service Oriented Infrastracture, SOI, Java , Graphic User Interface , JSP , HTML , CSS , Apache Tomcat.

Περιεχόμενα

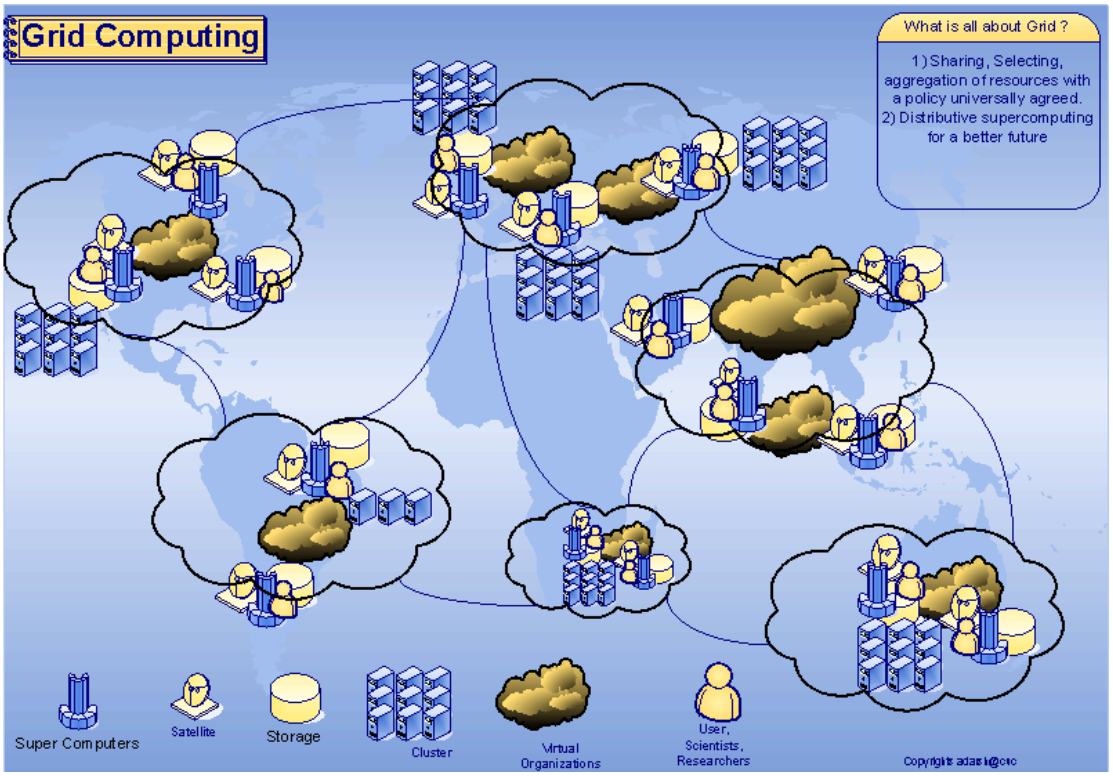
1 Το GRID

1.1 Γενικά για το GRID

1.1.1 Ορισμός

Το ***Grid (****Δίκτυο Κατανεμημένης Υπέρ-υπολογιστικής Ισχύος****)*** είναι μια αρχιτεκτονική διαμοιρασμού εφαρμογών/πόρων που δίνει την ικανότητα σε συνδεδεμένα ετερογενή συστήματα και εφαρμογές να μοιράζονται υπολογιστικούς και αποθηκευτικούς πόρους.

Στόχος αυτής της αρχιτεκτονικής είναι από μεγάλο αριθμό διαφορετικών συστημάτων που συνδέονται μεταξύ τους πάνω από προηγμένα δίκτυα , να δημιουργηθεί ένα απλό, εικονικό, ενοποιημένο σύστημα. Το εικονικό, ενιαίο αυτό σύστημα δίνει σε χρήστες και εφαρμογές πρόσβαση σε υπολογιστικούς πόρους, συσκευές και υπηρεσίες. Οι πόροι που διαμοιράζονται σε μια αρχιτεκτονική Grid μπορούν να είναι ετερογενείς(να έχουν υλοποιηθεί σε διαφορετικές πλατφόρμες, αρχιτεκτονικές υλικού/λογισμικού, γλώσσες προγραμματισμού). Μπορεί να βρίσκονται σε απομακρυσμένες γεωγραφικά τοποθεσίες και να ανήκουν σε διαφορετικές διαχειριστικές περιοχές (administrative domains). Σε σύγκριση με τον Παγκόσμιο Ιστό που είναι μια υπηρεσία για τη διανομή πληροφοριών μέσω του Διαδικτύου, θα μπορούσαμε να ισχυριστούμε ότι το Grid είναι μια υπηρεσία για τη διανομή υπολογιστικής δύναμης, αποθηκευτικών χώρων και άλλων πόρων μέσω του Διαδικτύου.



Εικόνα 1.1 Υπολογιστικό Πλέγμα

1.1.2 Ένας πιο τυπικός ορισμός

Δυστυχώς, οι τυπικοί ορισμοί του Grid computing είναι όπως οι πόροι σε ένα πλέγμα: πολυάριθμοι και ετερογενής. Ένας από τους καλύτερους ορισμούς είναι του Ian Foster στο βιβλίο του WHAT IS THE GRID? A THREE POINT CHECKLIST [1]. Μολονότι, όπως επισημαίνει και ο ίδιος, η λίστα ελέγχου εξακολουθεί να αφήνει περιθώρια για συζήτηση, παρέχει ωστόσο έναν απλό και περιεκτικό ορισμό. Παραθέτεται παρακάτω επακριβώς ο τυπικός ορισμός από τη λίστα ελέγχου:

«Το πλέγμα είναι ένα σύστημα που:

1. *συντονίζει πόρους που δεν υπόκεινται σε κεντρικό σύστημα ελέγχου...*
2. *.. .χρησιμοποιώντας τυποποιημένα, ανοικτά, γενικής χρήσης πρωτόκολλα και διασυνδέσεις...*
3. *.. .ώστε να παρέχει μη στοιχειώδης(προχωρημένη) Ποιότητα Υπηρεσίας (QoS)»*

Μια πιο προσεκτική ματιά στα πρώτα δύο σημεία:

1. ***Συντονίζει πόρους που δεν υπόκεινται σε κεντρικό έλεγχο***

   Αυτό αναφέρεται στην έννοια της «*Χρήσης πόρων από πολλούς οργανισμούς*». Σαφέστερα, αναφέρεται στην «ενσωμάτωση και τον συντονισμό πόρων και χρηστών που συνυπάρχουν σε διαφορετικούς τομείς ελέγχου(*control domain*)». Αυτοί οι «τομείς ελέγχου» θα μπορούσε να είναι διαφορετικές διοικητικές μονάδες μέσα σε μια εταιρεία.

2. ***Χρησιμοποιεί τυποποιημένα, ανοικτά, γενικής χρήσης πρωτόκολλα και διασυνδέσεις***

   Οι πόροι σε ένα πλέγμα είναι προφανώς ετερογενής. Σε ένα σύστημα πλέγματος, θα πρέπει με κάποιο τρόπο να πάρουμε ένα σύνολο από διαφορετικούς μεταξύ τους πόρους (ανεξάρτητους υπολογιστές, συμπλέγματα υπολογιστών , υπέρ-υπολογιστές, αποθηκευτικοί πόροι, κ.λπ.) που χρησιμοποιούν μια μεγάλη ποικιλία από λειτουργικά συστήματα και αρχιτεκτονικές, και με κάποιον τρόπο να συνεργαστούν για την επίλυση υπολογιστικών εργασιών. Ο μόνος τρόπος για να επιτευχθεί αυτό είναι με χρήση «τυποποιημένων, ανοικτών, γενικής χρήσης πρωτόκολλων και διασυνδέσεων» παρέχοντας μια «κοινή γλώσσα» που όσοι βρίσκονται στο πλέγμα μπορούν να κατανοήσουν. Η προσέγγιση με βάση το πρωτόκολλο επιτρέπει την ετερογένεια των πόρων και το γεγονός ότι βασίζεται σε πρότυπα προωθεί τη διαλειτουργικότητα.

Και πάλι, υπάρχουν μεγάλα περιθώρια για βελτίωση, αλλά ο τυπικός αυτός ορισμός είναι ένα σταθερό σημείο εκκίνησης.

1.1.3 Παραδείγματα συστημάτων Grid

- ***EGEE: Enabling Grids for E-Science in Europe*** (http://public.eu-egee.org/): Ένα φιλόδοξο έργο που δίνει στου επιστήμονες την δυνατότητα πρόσβασης σε υπολογιστικούς πόρους 27 χωρών. Το EGGE είναι επίσης υπευθυνο για την υπολογιστή ισχύ του επιταχυντή LHC.
- ***NEESit*** (http://it.nees.org/): Παρέχει μια εκτεταμένη υποδομή για τον NEES(Network for Earthquake Engineering Simulation – δίκτυο για την εξομοίωση σεισμών) και συνδέει μεταξύ τους ερευνητικά κέντρα σεισμών σε όλες της Ηνωμένες Πολιτείες Αμερικής.

- ***TeraGrid*** (http://www.teragrid.org/): Ένα σύστημα Grid που παρέχει μια δυνατή υποδομή για ανοικτή επιστημονική ερευνά. Από το 2004 το TeraGrid αποδίδει 20 teraflops υπολογιστής δύναμης και μπορεί να αποθηκεύσει 1 petabyte σε διανεμημένους αποθηκευτικούς πόρους.
- ***Access Grid*** (http://www.accessgrid.org/): Ένα σύστημα Grid που χρησιμοποιείται για την περαίωση «μεγάλης κλίμακας διανεμημένων συναντήσεων , συνεργασίες εργασίας , σεμινάρια , μαθήματα και εκπαίδευση».
- ***eDiaMoND*** (http://www.ediamond.ox.ac.uk/): Το έργο eDiaMoND είναι ένα καλό παράδειγμα χρήσης των τεχνολογιών Grid για την υγεία. Το έργο αυτό «συγκεντρώνει και διανέμει πληροφορίες για την θεραπεία του καρκίνου του μαστού, επιτρέπει την έγκαιρη εξέταση και διάγνωση και παρέχει στους επαγγελματίες υγείας εργαλεία και πληροφορίες για την θεραπεία της ασθένειας αυτής.

Για πληρέστερη λίστα των τρεχουσών εφαρμογών Grid απευθυνθείτε στην Βιβλιογραφία.

1.2 Κατηγοριοποίηση Συστημάτων Grid

Ο όρος Grid αναφέρεται σε ένα σύνολο από υπολογιστικούς πόρους οι οποίοι αναλαμβάνουν να εκτελέσουν διεργασίες. Έτσι στα μάτια του χρήστη φαίνεται ένας απλός υπολογιστικός πόρος, ενώ στην πραγματικότητα υπάρχει ένα πολυδύναμο κατανεμημένο πλέγμα από υπολογιστικούς πόρους. Ο διαχειριστής υπολογιστικών πόρων αναλαμβάνει εργασίες που υποβάλουν οι χρήστες και τις δρομολογεί ανάλογα για εκτέλεση στο Πλέγμα. Έτσι ο κάθε χρήστης μπορεί να υποβάλει ένα μεγάλο αριθμό εργασιών χωρίς να τον απασχολεί πού θα εκτελεστούν.

Η τεχνολογία Πλέγματος αναπτύχθηκε γιατί έγινε πλέον αποδεκτό στην επιχειρηματική κοινότητα η βαρύτητα των κατανεμημένων συστημάτων σε εφαρμογές όπως η οικονομική μοντελοποίηση και η ανάσυρση δεδομένων. Έτσι μέσα σε ένα Grid υπολογιστές, servers, αποθηκευτικά μέσα, βάσεις δεδομένων μπορούν να συνδυαστούν ώστε να δημιουργήσουν μαζική υπολογιστική ισχύ. Η αφομοίωση της τεχνολογίας Grid γίνεται με πολύ υψηλούς ρυθμούς.

Η τεχνολογία Grid μπορεί να κατανεμηθεί σε 3 μεγάλα μέρη: ***Cluster***, ***Enterprise*** και ***Global***:

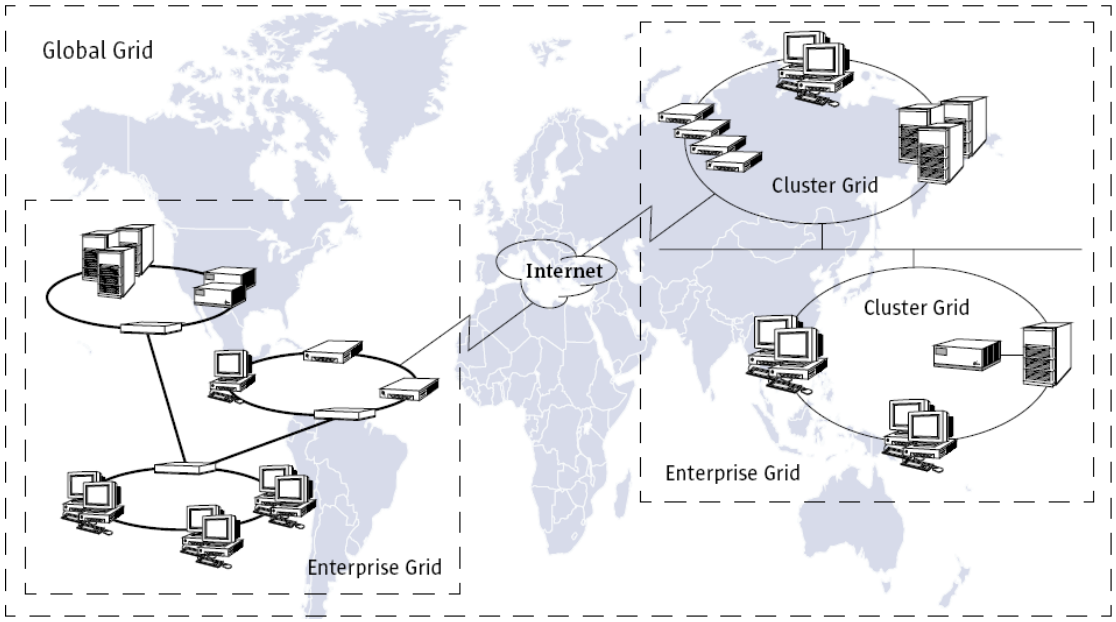
1. ***Cluster Grids*** *:* Σήμερα τα Cluster Grids αποτελούν την πιο διαδεδομένη και απλή μορφή Grid. Προσπαθώντας να ανταποκριθούν στις απαιτήσεις των περισσότερων οργανισμών, τα Cluster Grids αποτελούνται από ένα ή περισσότερα συστήματα τα οποία συνεργάζονται για να παράγουν προς το χρήστη ένα σημείο πρόσβασης. Τα Cluster Grids απασχολούν ένα μικρό αριθμό χρηστών οι οποίοι βρίσκονται σε ένα συγκεκριμένο project ή τμήμα μιας επιχείρησης προσφέροντας υποστήριξη σε εργασίες που απαιτούν υψηλές απαιτήσεις και υψηλή υπολογιστική ισχύ. Οι πόροι ενός Grid μπορεί να εστιαστούν είτε σε ένα σύνολο επαναλαμβανόμενων εργασιών, ή σε εργασίες που υποστηρίζουν παράλληλη εκτέλεση. Έτσι ένα Grid μπορεί να αναλάβει τη διεκπεραίωση εργασιών ανεξαρτήτων μεταξύ τους, αλλά και την ολοκλήρωση εργασιών που εκτελούνται σε διαφορετικές υπολογιστικές μονάδες αλλά επικοινωνούν ωστόσο μεταξύ τους για την επίλυση ενός συνόλου προβλημάτων.

2. ***Enterprise Grids :*** Καθώς οι ανάγκες για εξοικονόμηση χωρητικότητας συνεχώς αυξάνονται , οι οργανισμοί συνδυάζουν Cluster Grids σχηματίζοντας Enterprise Grids. Τα Enterprise Grids επιτρέπουν σε πολλαπλά project ή τμήματα μιας υπηρεσίας να μοιράζονται υπολογιστικές πηγές.

   Τα Enterprise Grids αποτελούνται από ένα εξαπλωμένο σύνολο από servers οι οποίοι μπορεί να είναι σε διάφορα μέρη μιας επιχείρησης ωστόσο αλληλεπιδρούν μεταξύ τους. Μια επιχείρηση χρησιμοποιεί τα Enterprise Grid για να διαχειριστεί μια μεγάλη ποικιλία εργασιών, συμπεριλαμβανομένου του data mining,της επεξεργασία frames για animation, απορρόφηση μεγάλου φόρτου εργασίας κατά περιόδους σε μια επιχείρηση και πολλά άλλα.

3. ***Global Grids*** : Όταν ένα Enterprise Grid δεν μπορεί να ανταποκριθεί στις ελάχιστες απαιτήσεις μιας εφαρμογής, μια επιχείρηση μπορεί να αναζητήσει διαθέσιμους υπολογιστικούς πόρους μέσω ενός Global Grid. Σχεδιασμένα για τις ανάγκες εφαρμογών που απαιτούν πολλούς υπολογιστικούς πόρους τα Global Grids παρέχουν την ισχύ των κατανεμημένων πόρων σε χρήστες παντού στον κόσμο. Τα Global Grids είναι ένα σύνολο από Enterprise Grids , τα οποία προορίζονται για παγκόσμια χρήση. Οι υπολογιστικοί πόροι ενδέχεται να είναι γεωγραφικά διασκορπισμένοι, ωστόσο είναι συνδεδεμένοι μεταξύ τους. Μπορούν να

χρησιμοποιηθούν από κάθε χρήστη ανεξάρτητα, από επιχειρήσεις και οργανισμούς, ανταποκρινόμενα σε βαρύ φόρτο εργασίας.



Εικόνα 1.2 Η δύναμη του Grid

1.3 Σύγκριση με άλλες τεχνολογίες

Στην πραγματικότητα το Grid αποτελεί την τελευταία και ποιο ολοκληρωμένη εξέλιξη πιο γνωστών τεχνολογιών, όπως το distributed computing(διανεμημένα υπολογιστικά συστήματα), το διαδίκτυο, και τα δίκτυα P2P(peer-to-peer).

Όπως το διαδίκτυο, το Grid κρύβει την πολυπλοκότητα που υπάρχει πίσω από την τεχνολογία. Πολλοί χρήστες απολαμβάνουν μία μοναδική, ενοποιημένη εμπειρία. ***Αντίθετα με το διαδίκτυο***, το οποίο κυρίως επιτρέπει την επικοινωνία, το Grid επιτρέπει την πλήρη συνεργασία για την υλοποίηση κοινών επαγγελματικών στόχων.

Όπως τα δίκτυα peer-to-peer, το Grid επιτρέπει στους χρήστες να μοιράζονται αρχεία. ***Αντίθετα με τα δίκτυα peer-to-peer***, το Grid επιτρέπει ανάμεσα σε πολλούς χρήστες όχι μόνο τον διαμοιρασμό αρχείων, αλλά και άλλων πόρων όπως υπολογιστικούς κύκλους , μνήμη κ.α.

Όπως τα clusters και το distributed computing, το Grid ενοποιεί υπολογιστικούς πόρους. ***Αντίθετα με τα clusters και distributed computing***, τα οποία χρειάζονται φυσική γειτνίαση και υπολογιστική ομοιογένεια, τα Grid μπορούν να είναι γεωγραφικά κατανεμημένα και ετερογενή.

Όπως οι τεχνολογίες virtualization, το Grid επιτρέπει τη εικονοποίηση IT (Information Technology) πόρων. ***Αντίθετα με τις τεχνολογίες virtualization***, οι οποίες εικονοποιούν ένα μοναδικό σύστημα, το Grid επιτρέπει την εικονοποίηση αχανών και διάσπαρτων IT πόρων.

1.4 Ιστορία του Grid

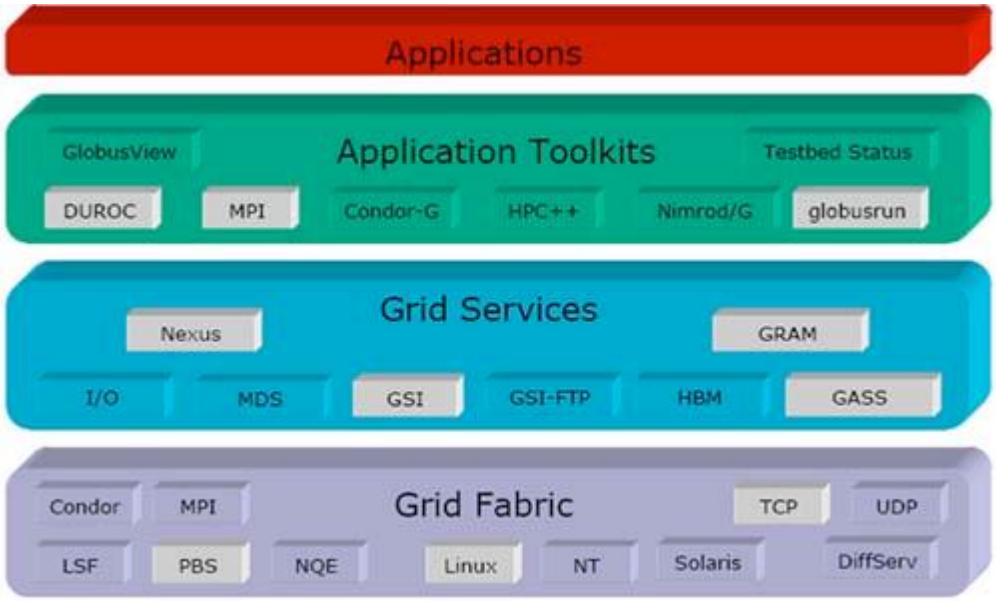
Το ιστορικό των τεχνολογιών διασύνδεσης των υπολογιστών αρχίζει στις αρχές της δεκαετίας του 70 στην Αμερική από την ερευνητική ομάδα ARPA (Advanced Research Projects Agency) που είχε δημιουργηθεί από την κυβέρνηση για στρατιωτικούς σκοπούς. Από την ομάδα αυτή αναπτύχθηκαν πολλά σημαντικά πρωτόκολλα και τεχνολογίες όπως το TCP/IP.Υπό την καθοδήγηση του Dr. J.C.R. Licklider και την συμβολή ερευνητών από διάφορα πανεπιστήμια της χώρας η έρευνα κατέληξε στη δημιουργία του πρώτου δικτύου υπολογιστών, προάγγελο του Internet, γνωστό ως ARPANET, στα 50 kbps. Η ερευνητική δραστηριότητα στον τομέα των δικτύων υπολογιστών συνεχίστηκε, με αποτέλεσμα την δημιουργία του NSFNET [1986], δικτύου στα 56Kbps που συνέδεε τα πέντε NSF κέντρα υπερ-υπολογιστών.

Ως συνέχεια και εξέλιξη αυτών των τεχνολογιών μπορούμε να θεωρήσουμε το πρόγραμμα Condor [1988] του πανεπιστημίου του Wisconsin. Το σύστημα αυτό είναι ένας «διαχειριστής φόρτου εργασίας» (workload manager), με δυνατότητες παρακολούθησης και διαχείρισης πόρων, δρομολόγησης εργασιών και αποτελεί το πρώτο πρόγραμμα με κατεύθυνση προς την αξιοποίηση των Grid υπηρεσιών.

Η ανάπτυξη δικτύων υψηλών ταχυτήτων και η ανάγκη για μεγάλη επεξεργαστική ισχύ οδήγησε σε έντονη ερευνητική δραστηριότητα στον τομέα των Grid τεχνολογιών. Η έρευνα αυτή κατέληξε σε ενδιαφέροντα αποτελέσματα με πιο σημαντικά τα προγράμματα LEGION [1993], SRB [1997], GLOBUS [1998]. Το πρώτο βασίζεται στην ιδέα του εικονικού υπολογιστή (virtual computer) : όλοι οι πόροι συνδεδεμένοι μεταξύ τους, εμφανίζονται στον χρήστη ως μία εικονική μηχανή, με αρκετά μειονεκτήματα όμως, όπως η πολύπλοκη

υλοποίηση και η μικρή αποδοτικότητα. Το SRB (Storage Resource Broker) ήταν μια πλατφόρμα διαχείρισης αποθηκευτικών πόρων που βοήθησε πολύ στην ανάπτυξη των Grid τεχνολογιών, αφού αντιμετώπισε τα προβλήματα μεταφοράς δεδομένων σε Grid περιβάλλον.

Τέλος, το πιο διαδεδομένο σύστημα διαχείρισης Grid υπηρεσιών είναι το GLOBUS, που αναπτύχθηκε στο Argonne National Lab στο πανεπιστήμιο του Berkeley. Το GLOBUS προσωποποίησε πρωτοκόλλα για τη ασφάλεια, μεταφορά δεδομένων, ανακάλυψη πόρων και εκτέλεση εργασιών. Λειτουργεί σε χαμηλό επίπεδο και είναι ανεπτυγμένο σε επίπεδα υπηρεσιών:



Εικόνα 1.3 Βασική Grid Αρχιτεκτονική

Η συνέχεια γίνεται με την ανάπτυξη των Web Services [2001] , υπηρεσιών που είναι προσβάσιμες μέσω του διαδικτύου και χρησιμοποιούν συγκεκριμένα πρωτόκολλα περιγραφής (XML, SOAP, WSDL) και τέλος με την εγκαθίδρυση του OGSA (Open Grid Service Architecture) [2002] ως κύριας αρχιτεκτονική της Grid τεχνολογίας. Το πρότυπο αυτό είναι το πιο σημαντικό για τις τεχνολογίες του Grid, αφού περιγράφει τις δυνατότητες του συστήματος αυτού να αναλύσει την λειτουργία του σε όλα τα επίπεδά του.

1.5 Αρχιτεκτονικές Συστημάτων Grid

Υπάρχουν δύο βασικές αρχιτεκτονικές πίσω από την οικοδόμηση κλιμακωτών (scalable) Grid συστημάτων :

- ***Service-Oriented Architecture (SOA)*** ή αλλιώς Αρχιτεκτονική προσανατολισμένη σε Υπηρεσίες. Η αρχιτεκτονική αυτή συνίσταται στην συνάθροιση(aggregation) επαναχρησιμοποιήσιμων προγραμμάτων, τα οποία ονομάζουμε Υπηρεσίες (services). Τα services καλούνται από απομακρυσμένους πελάτες , πάνω από κάποιο δίκτυο κατά τρόπο ανεξάρτητου πλατφόρμας ή γλώσσας προγραμματισμού.
- ***Peer to peer (P2P)*** Με τον όρο αυτό εννοούμε μια συνάθροιση ισοδύναμων προγραμμάτων , τα οποία παρέχουν κάποια συγκεκριμένη λειτουργικότητα και έχουν την δυνατότητα να δανείζουν και να δανείζονται διάφορους πόρους του δικτύου. Χαρακτηριστικά παραδείγματα είναι το εύρος ζώνης , υπολογιστική ισχύς, ικανότητα αποθήκευσης κτλ. Μεγάλα πλεονεκτήματα της συγκεκριμένης αρχιτεκτονικής είναι ο μεγάλος βαθμός κλιμάκωσης (scalability) και η μεγάλη αντοχή στα λάθη (fault tolerance).

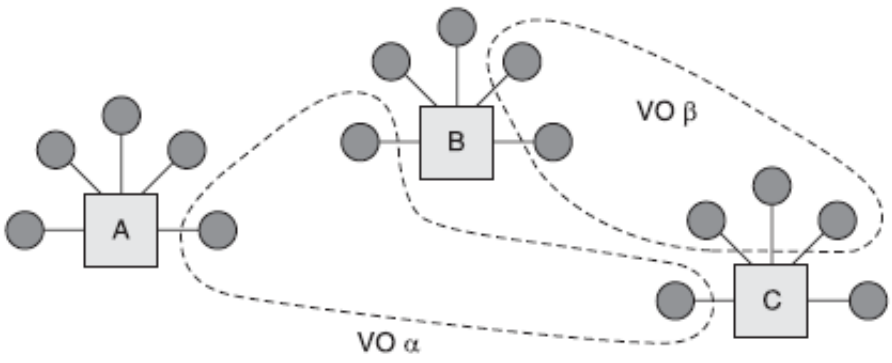
Παρόλα τα πλεονεκτήματα της p2p αρχιτεκτονικής , σε Grid συστήματα συνηθίζεται να ακολουθούμε αρχιτεκτονική προσανατολισμένη σε Υπηρεσίες . Οι βασικότεροι λόγοι γι' αυτό είναι:

- Οι πελάτες είναι απλούστεροι (light-weight) , το οποίο σημαίνει ότι μπορούν να εγκατασταθούν και να συντηρηθούν και από μη έμπειρους χρήστες, «κρύβοντας» ουσιαστικά την πιθανώς περίπλοκη υλοποίηση της υπηρεσίας. Οι πελάτες δεν είναι τόσο ισχυρά συνδεδεμένοι με το υπόλοιπο σύστημα, πράγμα που δίνει την δυνατότητα στους χρήστες να κινούνται, να μοιράζονται και να έχουν πρόσβαση στις υπηρεσίες από διαφορετικές περιοχές. Ενισχύεται η διαλειτουργικότητα (interoperability) , δημιουργώντας πρότυπα καλώς ορισμένα για την επικοινωνία με απομακρυσμένες υπηρεσίες.
- Διευκολύνεται η συντήρηση των συστημάτων.

2 OGSA (The Open Grid Services Architecture)

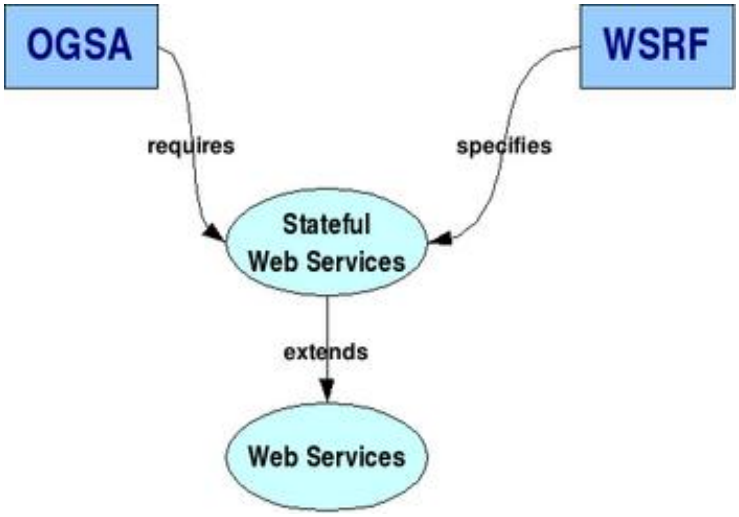
Ένα κατανεμημένο σύστημα είναι λογικό ότι χρειάζεται να έχει μια αρχιτεκτονική η οποία να επιτρέπει τον συντονισμό και τη διαχείριση του κατά το δυνατόν καλύτερο τρόπο. Αυτό σημαίνει ότι ένα σύστημα Grid χωρίζεται σε μικρότερες και διαφορετικές μονάδες οι οποίες διαθέτουν διαφορετικό ρόλο για την καλύτερη αξιοποίηση χρόνου και πόρων κατά τη διάρκεια μιας ή και πολλών εργασιών[2] . Ένα σύστημα Grid αποτελείται συνήθως από διαφορετικές υπολογιστικές μονάδες.

Εικόνα 2.1 Ένας οργανισμός και οι πόροι του



Εικόνα 2.2 Ένας Εικόνικός Οργανισμός

- ***Υπηρεσία διαχείρισης VO (Εικονικού Οργανισμού):*** Έχει ως ρόλο την διαχείριση των κόμβων και χρηστών που αποτελούν μέλη του Εικονικού Οργανισμού.
- ***Υπηρεσία Αναζήτησης και Διαχείρισης Πόρων:*** Με αυτό τον τρόπο οι εφαρμογές Grid θα ζητούν πόρους που ανταποκρίνονται στις απαιτήσεις τους και έπειτα θα τους διαχειρίζονται.
- ***Υπηρεσία διαχείρισης εργασιών:*** Ο κάθε χρήστης μπορεί να υποβάλει εργασίες προς το Grid.
- ***Άλλες υπηρεσίες*** *που έχουν να κάνουν με την ασφάλεια, με την διαχείριση δεδομένων κτλ.*



Εικόνα 2.3 Σχέση μεταξύ OGSA, WSRF και Web Services

Όλες αυτές οι υπηρεσίες αλληλεπιδρούν συνεχώς. Για παράδειγμα ***η υπηρεσία Διαχείρισης Εργασιών*** μπορεί να συμβουλευτεί την ***υπηρεσία Αναζήτησης Πόρων*** για να βρει υπολογιστικούς πόρους που ανταποκρίνονται στις απαιτήσεις τις τρέχουσας εργασίας. Ένα τόσο μεγάλο πλήθος υπηρεσιών δε θα ήταν δύσκολο να προκαλέσει χάος. Ο κάθε πάροχος τέτοιων υπηρεσιών θα μπορούσε να υλοποιήσει δικές του υπηρεσίες με τα ίδια καθήκοντα , αλλά με διαφορετικό τρόπο, παρέχοντας με αυτόν τον τρόπο διαφορετικές λειτουργίες. Κάτι τέτοιο θα αποτελούσε ανασταλτικό παράγοντα στην συνεργασία ενός μεγάλου αριθμού υπηρεσιών με διαφορετικές υλοποιήσεις.

Η λύση σε αυτό το πρόβλημα είναι στο να υπάρχει ένα κοινό πρότυπο για κάθε τύπο υπηρεσίας, μία βάση που θα εξυπηρετούσε στην υλοποίηση υπηρεσιών με κανόνες τους κανόνες της βάσης αυτής. Αναφέροντας ως παράδειγμα τον Παγκόσμιο Ιστό (World Wide Web) γίνεται ξεκάθαρη η πολυτιμότητα των προτύπων για εφαρμογές προς ευρεία χρήση. Ένας από τους λόγους που καθιέρωσε τον Παγκόσμιο Ιστό (World Wide Web) προς τον περισσότερο κόσμο είναι το γεγονός ότι βασίζεται σε πρότυπα (HTML , HTTP κτλ) τα οποία ακολουθούν όλοι οι κατασκευαστές των Internet Browsers (Microsoft, Mozilla , κτλ) . Έτσι έχοντας ένα Internet Browser ο οποίος ακολουθεί αυτά τα πρότυπα είμαστε σε θέση να έχουμε πρόσβαση στα περισσότερα websites ανεξάρτητα από το τι τεχνολογία χρησιμοποιούν.

Το OGSA, υλοποιήθηκε από το Global Grid Forum ώστε να θέσει ένα κοινό πρότυπο για τις εφαρμογές που βασίζονται στην τεχνολογία Grid. Ο στόχος του OGSA είναι να ορίσει πρακτικά ένα κοινό πρότυπο για όλες τις υπηρεσίες που είναι διαθέσιμες σε ένα Grid σύστημα (Υπηρεσία διαχείρισης εργασιών, Υπηρεσία Αναζήτησης και Διαχείρισης Πόρων, υπηρεσίες ασφαλείας, κτλ) υλοποιώντας ένα σύνολο από interfaces για τις υπηρεσίες αυτές. Το OGSA έχει θέσει τα πρότυπα για τις πιο σημαντικές υπηρεσίες που θα συναντήσει κάποιος σε ένα Grid σύστημα και που η μοντελοποίηση τους θα αποφέρει μεγάλα οφέλη.

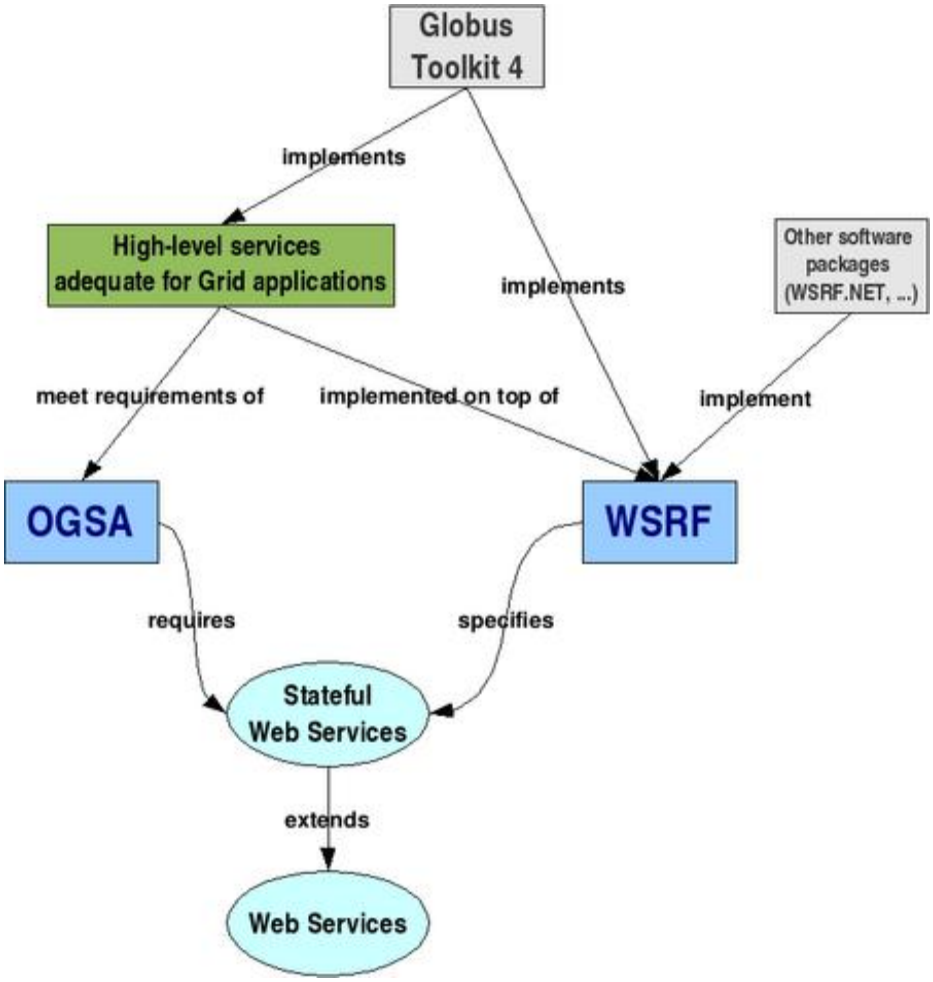
Από τη στιγμή που είχε σχεδιαστεί η αρχιτεκτονική αυτή, έγινε αισθητή η ανάγκη για ένα κατανεμημένο middleware , στο οποίο θα βασιζόταν η αρχιτεκτονική αυτή. Το middleware αυτό θα είχε ως στόχο τον κοινό τρόπο κλήσης λειτουργιών , από τη στιγμή που η αρχιτεκτονική αυτή έπρεπε να έχει ευρεία χρήση από επιχειρήσεις και όχι μόνο. Υπάρχουν πολλά κατανεμημένα middleware τα οποία είναι σε θέση να αποτελέσουν την βάση αυτή της αρχιτεκτονικής (CORBA, RMI, RPC), ωστόσο προτιμήθηκε η λύση των Web Services, για λόγους που διευκρινίζονται παρακάτω στην ανάλυση τους.

Αν και η λύση των Web Services ήταν η ιδανική λύση, δεν υποστήριζαν μια βασική απαίτηση του OGSA το οποίο ήταν ότι το middleware έπρεπε να υποστηρίζει *stateful* ιδιότητες, δηλαδή να είναι σε θέση να συγκρατεί πληροφορία σε κάποιο state. Ωστόσο τα Web Services ενώ στην θεωρία μπορούν να είναι είτε *stateless* είτε *stateful*, στην πράξη χρησιμοποιούνται ως *stateless*.

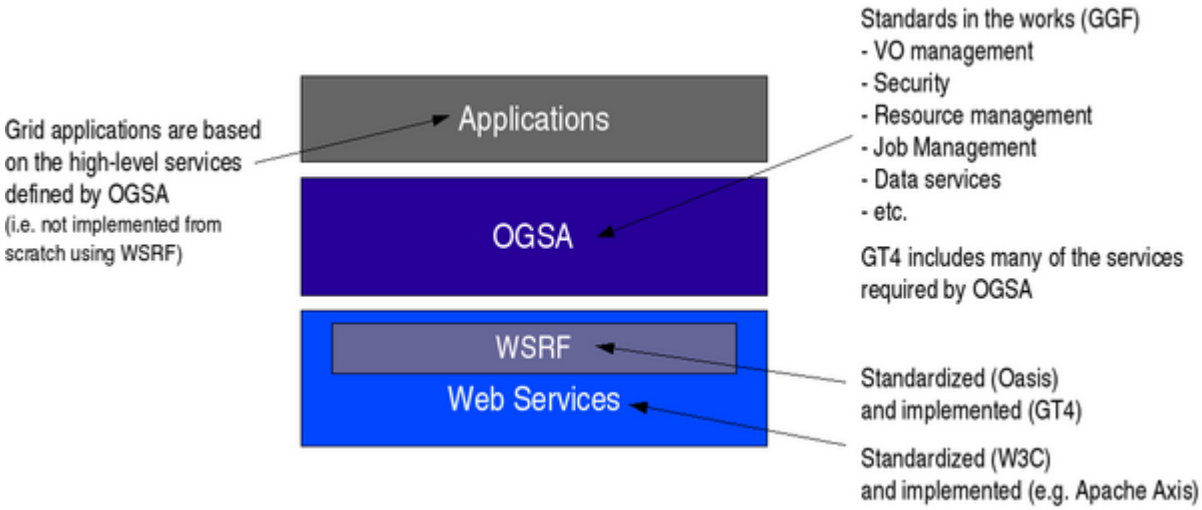
Σε αυτό το σημείο τη λύση έρχεται να δώσει το WSRF (The Web Services Resource Framework). Το WSRF καθορίζει πως τα Web Services μπορούν να συμπεριφέρονται ως

stateful και τους προσδίδουνε τέτοια συμπεριφορά, συμπεριλαμβάνοντας και άλλα επιπρόσθετα χαρακτηριστικά. Έτσι λοιπόν σχηματίζεται μια σχέση μεταξύ OGSA και WSRF, το OGSA είναι η *αρχιτεκτονική* και το WSRF είναι η *υποδομή* πάνω στην οποία έχει κτιστεί η αρχιτεκτονική αυτή.

Εκτός της αρχιτεκτονικής αυτής και της υποδομής της , απαραίτητη κρίνεται η ύπαρξη ενός λογισμικού το οποίο θα είναι αρμόδιο για τη δημιουργία Grid συστημάτων. Ένα τέτοιο λογισμικό είναι το Globus . Το Globus περιλαμβάνει υπηρεσίες υψηλού επιπέδου οι οποίες με κατάλληλη χρήση είναι ικανές για τη βοήθεια στη δημιουργία ενός Grid συστήματος. Αυτές οι υπηρεσίες, ανταποκρίνονται στις απαιτήσεις που θέτει το OGSA. Έτσι κάποιες λειτουργίες του Globus είναι , παρακολούθηση πόρων, υπηρεσία αναζήτησης, υποδομή υποβολής εργασιών, υποδομή ασφαλείας, και υπηρεσία διαχείρισης δεδομένων. Οι περισσότερες από αυτές τις υπηρεσίες υλοποιούνται πλήρως πάνω σε ήδη υπάρχουσες υπηρεσίες του WSRF . Αυτό γίνεται πιο κατανοητό μελετώντας το επόμενο σχήμα.



Εικόνα 2.4 Η σχέση μεταξύ των OGSA,GT4,WSRF και Web Services



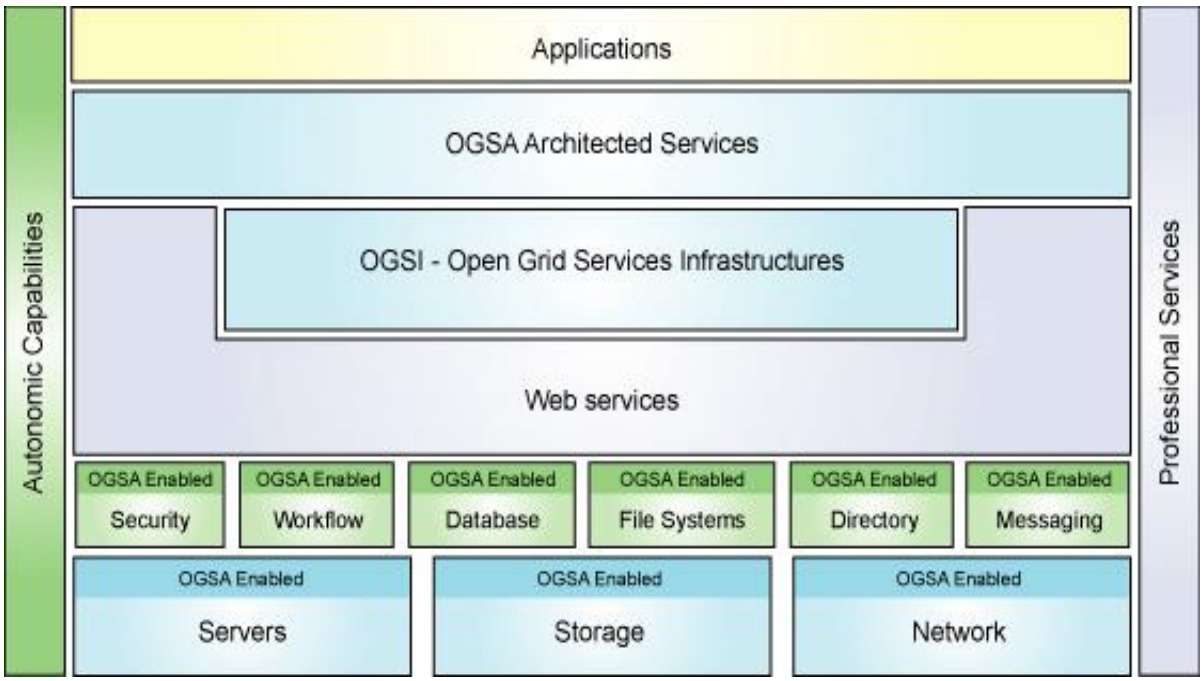
Εικόνα 2.5 Τμηματικό διάγραμμα των OGSA,GT4,WSRF και Web Services

2.1.1.1 Ποιοι είναι οι στόχοι του OGSA;

Οι κύριοι στόχοι του OGSA είναι :

- να διαχειρίζεται πόρους απομακρυσμένων ετερογενών συστημάτων
- να αποδίδει σημαντική ποιότητα στις υπηρεσίες που παρέχει (QoS)
- να παρέχει τις βάσεις για αυτόνομες λύσεις διαχείρισης. Η διαφορετικότητα των πόρων που συμμετέχουν σε ένα Grid και ο δυναμικός τρόπος συνεργασίας τους απαιτεί ένα σύστημα διαχείρισης που θα προσαρμόζεται σύμφωνα με τις εκάστοτε ανάγκες.
- Να καθορίζει γνωστά πρότυπα και πρωτόκολλα λειτουργίας. Η δυναμική εισαγωγή και συνεργασία ετερογενών συστημάτων στο δίκτυο του Grid, βασίζεται στην υιοθέτηση γνωστών, ευρείας χρήσης προτύπων.
- Να εκμεταλλεύεται υπάρχουσες τεχνολογίες και να τις προσαρμόζει στο Grid, αν είναι εφικτό. Το OGSA βασίζεται στη τεχνολογία των Web Services (υπηρεσίες διαδικτύου).

2.1.1.2 Η αρχιτεκτονική του OGSA



Εικόνα 2.6 : OGSA Αρχιτεκτονική

Το OGSA αποτελείται από τέσσερα βασικά επίπεδα :

- τους πόρους (φυσικούς και λογικούς)
- τις Web Services συμπεριλαμβανομένου και της OGSI υποδομής που έχει άμεση σχέση με τις υπηρεσίες
- τις υπηρεσίες σχεδιασμένες από το OGSA
- τις Grid εφαρμογές

2.1.1.3 Επίπεδο φυσικών και λογικών πόρων

Αυτό είναι το πιο χαμηλό επίπεδο από τα τέσσερα και έχει να κάνει με τους πόρους που συμμετέχουν στο Grid. Η έννοια του «πόρου» στο OGSA είναι πολύ σημαντική και δεν είναι συγκεκριμένη. Ως πόρος (resource) σε ένα Grid μπορεί να θεωρηθεί από τον επεξεργαστή ενός υπολογιστή, μέχρι το τμήμα υπολογιστών μιας μεγάλης εταιρίας. Επίσης εκτός από υπολογιστικές μονάδες, συμμετέχουν και μονάδες αποθήκευσης, βάσεις δεδομένων, δικτυακές υποδομές κλπ. Αυτά που αναφέρθηκαν παραπάνω είναι κυρίως οι φυσικοί πόροι.

Εκτός από αυτούς έχουμε και τους λογικούς, κάποια ενδιάμεσα συστήματα (middleware) που χρησιμοποιώντας τους φυσικούς πόρους, προσφέρουν στοιχειώδεις υπηρεσίες, όπως διαχείριση αρχείων ή βάσεων δεδομένων, στο παραπάνω επίπεδο.

2.1.1.4 Επίπεδο υπηρεσιών διαδικτύου (Web Services)

Μία πολύ βασική θεώρηση του OGSA είναι ότι όλοι οι πόροι του συστήματος αντιστοιχίζονται με υπηρεσίες. Έτσι η OGSI υποδομή (Open Grid Services Infrastructure) χρησιμοποιεί συγκεκριμένες, γνωστές υπηρεσίες δικτύου και πρωτόκολλα όπως XML και WSDL για να δημιουργήσει επαφές και διασυνδέσεις κάθε Grid υπηρεσίας με τους πόρους του συστήματος. Η OGSI επεκτείνει τις δυνατότητες των δικτυακών υπηρεσιών έτσι ώστε να παρέχει δυναμικές και αξιόπιστες υπηρεσίες όπως απαιτείται για τη μοντελοποίηση των πόρων.

2.1.1.5 Επίπεδο Grid υπηρεσιών

Οι υπηρεσίες διαδικτύου και οι δυνατότητες του OGSI παρέχουν τη βασική υποδομή για το επόμενο επίπεδο, των Grid υπηρεσιών. Αυτή την εποχή υπάρχει μεγάλη δραστηριότητα προς την κατεύθυνση προσδιορισμού και προτυποποίησης τέτοιων υπηρεσιών, όπως υπηρεσίες Υπολογισμού.

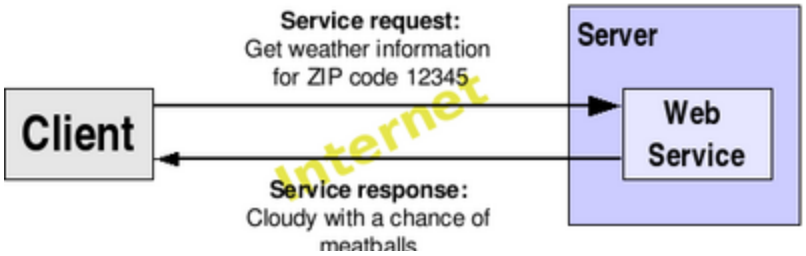
3 Web Services

3.1 Εισαγωγή στα Web Services

Οι σύγχρονες τάσεις στις τεχνολογίες πλέγματος δεικτοδοτούν την αξιοποίηση τεχνολογιών από το χώρο των υπηρεσιών διαδικτύου (Web Services) για την υλοποίηση συστημάτων Grid. Η τάση αυτή ενισχύεται ακόμα περισσότερο από τη δυνατότητα που παρέχουν οι τεχνολογίες Υπηρεσιών Διαδικτύου για λειτουργικότητα, ευχρηστία, απόδοση και κλιμάκωση.

Καθώς προχωρά ο χρόνος η τεχνολογία των Web Services γίνεται όλο και πιο απαραίτητη και σημαντική στους τομείς του web development και intergration. Τα Web Services επιτρέπουν σε web based εφαρμογές να επικοινωνούν με μηνύματα XML και να σχηματίζουν κατανεμημένα συστήματα ανεξάρτητα το ένα από το άλλο. Αναφερόμενος κάποιος σε Web Services εννοεί αυτόνομα προγράμματα τα οποία είναι διαθέσιμα στο Internet και μπορούν να αλληλεπιδρούν το ένα με το άλλο με μηνύματα μορφής XML . Είναι αυτοπεριγραφόμενα , κάτι το οποίο σημαίνει ότι το public interface ενός Web Service, αποτελούμενο από public μεθόδους και παραμέτρους πρέπει να συνοδεύει το service.

Τα Web Services λοιπόν αποτελούν ένα ιδιαίτερα χρήσιμο μέσο επικοινωνίας για εφαρμογές τύπου client / server *[2]* . Τα κύρια στοιχεία ενός Web Service είναι το service request το οποίο ζητά μια πληροφορία και το service response το οποίο αποστέλει την πληροφορία για το response που τη ζήτησε. Έστω λοιπόν ότι θέλει κάποιος να γνωρίσει τον καιρό μέσω μιας εφαρμογής που προσφέρει αυτές τις πληροφορίες και το μέσο μετάδοσης είναι τα web services , ένα απλό σχήμα που περιγράφει την ιδέα ενός web service φαίνεται παρακάτω.



Εικόνα 3.1: Web Services

Ωστόσο η χρήση Web Services είναι κάτι το οποίο πρέπει να γίνεται λαμβάνοντας υπόψη τα πλεονεκτήματα αλλά και τα μειονεκτήματα τους, τα οποία και παραθέτονται .

Τα βασικά χαρακτηριστικά των Web services συνίστανται στα ακόλουθα:

- Είναι ***αυτόνομα components***, τα οποία επεξεργάζονται καλώς ορισμένα XML μηνύματα. Παρέχουν ένα καλώς ορισμένο interface, το οποίο περιγράφεται από ένα XML έγγραφο, με τη χρήση της *WSDL (Web Services Description Language).* Το έγγραφο αυτό καλείται WSDL συμβόλαιο (contract) και περιγράφει τις μεθόδους τις οποίες υποστηρίζει η εκάστοτε υπηρεσία , του τύπους των δεδομένων που χρησιμοποιούνται και τον τρόπο με τον οποίο ένας πελάτης μπορεί να επικοινωνήσει με αυτές.
- ***Παρέχουν endpoints***, στα οποία πελάτες ή άλλες υπηρεσίες μπορούν να συνδεθούν. Η πληροφορία αυτή δίνεται μέσω της *διεύθυνσης* της υπηρεσίας , το οποίο είναι συνήθως ένα *URL*.

3.1.1 Πλεονεκτήματα

Τα Web Services είναι platform-independent και language-independent , λόγω της χρήσης της γλώσσας XML. Αυτό σημαίνει ότι ένα πρόγραμμα client ενδέχεται να έχει υλοποιηθεί σε C++ και να εκτελείται σε περιβάλλον Windows, ενώ το Web Service είναι προγραμματισμένο σε Java και εκτελείται σε περιβάλλον Linux.

Τα περισσότερα Web Services χρησιμοποιούν το πρωτόκολλο HTTP για την μετάδοση μηνυμάτων (όπως τα service request και service response). Αυτό αποτελεί μεγάλο πλεονέκτημα για όσους θέλουνε να αναπτύξουν μια Internet εφαρμογή και θέλει να αποφύγει επιπλοκές με Internet proxies και Firewalls αφού τα ίδια δεν είναι υπεύθυνα για τον έλεγχο μεταφοράς δεδομένων μέσω HTTP.

Τα Web services είναι ανάλογα με τα παραδοσιακά αντικειμενοστρεφή components, στο σημείο ότι έχουν ένα καλώς ορισμένο interface και παρέχουν πρόσβαση σε μία ή περισσότερες μεθόδους. Η κύρια διαφορά τους συνίσταται στο γεγονός ότι τα Web services είναι πιο χαλαρά συνδεδεμένοι με τους πελάτες τους, απ' ότι τα παραδοσιακά αντικειμενοστρεφή (Object Oriented - OO) components. Για παράδειγμα, μια αλλαγή στο interface ενός παραδοσιακού component, θα προκαλέσει το χτίσιμο του πελάτη της υπηρεσίας από την αρχή, καθώς είναι ισχυρά συζευγμένα. Αντίθετα , στο περιβάλλον των Web services, ο πελάτης μπορεί εύκολα και γρήγορα να «αναγεννήσει» τον πελάτη της αντίστοιχης Υπηρεσίας , μέσω του WSDL εγγράφου, μιας και αυτό το έγγραφο περιγράφει πλήρως την υπηρεσία. Η ιδιότητα αυτή καλείται *χαλαρή σύζευξη (loose coupling)* .

Ένα ακόμα μεγάλο πλεονέκτημα , όσον αφορά τα Web Services, είναι η εύκολη επεκτασιμότητα τους, πράγμα που οφείλεται στο WSDL έγγραφο. Ένα πρακτικό παράδειγμα φαίνεται στην περίπτωση της υιοθέτησης κάποιας πολιτικής ασφαλείας για κάποια υπηρεσία. Αν και το SOAP, από μόνο του δεν μπορεί να διατηρήσει κατάσταση (stateless), με τις κατάλληλες επεκτάσεις στο WSDL συμβόλαιο, μπορούμε να διαμορφώσουμε το SOAP έτσι ώστε να μεταφέρει επιπλέον πληροφορίες (enhancements), για την απ' άκρου σε άκρο ασφάλεια της επικοινωνίας. Χαρακτηριστικά παραδείγματα είναι το είδος κρυπτογράφησης που θα χρησιμοποιηθεί , η πολιτική ασφαλείας που έχει οριστεί και άλλα.

3.1.2 Μειονεκτήματα

- ***Δημιουργούν ιδιαίτερο βάρος στην μετάδοση της πληροφορίας***. Η μετάδοση όλων των δεδομένων με τη μορφή XML είναι λιγότερο αποδοτικό από την μετάδοση δεδομένων με δυαδική μορφή. Το κέρδος σε φορητότητα που προσφέρει το XML , χάνεται σε απόδοση. Ακόμα και έτσι, αυτό το overhead είναι αποδεκτό από τις περισσότερες εφαρμογές.
- ***Είναι ακόμα σε πρώιμο στάδιο***. Τα Web Services είναι σχετικά νέα τεχνολογία, ακόμα και αν χρησιμοποιούν γλώσσες όπως XML, WSDL και πρωτόκολλα όπως HTTP, SOAP τα οποία είναι ιδιαίτερα σταθερά. Ο ρυθμός εξέλιξής τους παρόλο το ότι βρίσκονται σε πρώιμο στάδιο είναι ιδιαίτερα γρήγορος.

3.2 Στοίβα των Web Services

Το κλειδί για την διαλειτουργικότητα , που είναι βασικό χαρακτηριστικό των Web services είναι μια 3-επιπέδων στοίβα προτύπων, όπως εξηγείται παρακάτω.

3.2.1 HTTP

Το ***HTTP*** (*Hyper Text Transfer Protocol*) είναι το πιο συνηθισμένο πρωτόκολλο επικοινωνίας στα Web services αλλά και γενικότερα στο Web, για την επικοινωνία πάνω από το TCP/IP . Μπορεί να προσφέρει αναβαθμισμένη ασφάλεια , αφού μπορεί να δουλέψει πάνω και από TLS/SSL.

3.2.2 SOAP

Το ***SOAP*** (*Simple Object Access Protocol*) είναι ένα XML πρότυπο, το οποίο ορίζει την δομή των μηνυμάτων, που ανταλλάσσονται μεταξύ απομακρυσμένων υπηρεσιών. Το SOAP έχει την ευελιξία να *υποστηριχθεί* και από πρωτόκολλα πέραν του HTTP, όπως είναι το FTP, το SMTP και άλλα. Παρόλα αυτά, αν και φαίνεται περίεργο , τα Web services, δεν απαιτούν την χρήση του SOAP για την μεταξύ τους επικοινωνία , αν και είναι ευκολότερη με αυτόν τον τρόπο.

3.2.3 WSDL

Το ***WSDL*** (*Web Service Description Language*) είναι ένα XML πρότυπο , μέσω του οποίου περιγράφονται με ακρίβεια το interface ενός service. Η ύπαρξη του , είναι πρωτεύουσας σημασίας για την επίτευξη της διαλειτουργικότητας στο περιβάλλον των Web services, γι’ αυτό το λόγο η δομή του αναλύεται περαιτέρω παρακάτω.

3.2.4 UDDI

Το ***UDDI*** (*Universal Description, Discovery and Integration*) είναι ένα specification, το οποίο ορίζει τον τρόπο οργάνωσης και αποθήκευσης πληροφοριών σχετικών με επαγγελματικές Web services, σε κατανεμημένους καταλόγους (*registries*). Οι πληροφορίες αυτές συνήθως αφορούν το WSDL έγγραφο και την URL διεύθυνση κάθε, υποψήφιας προς

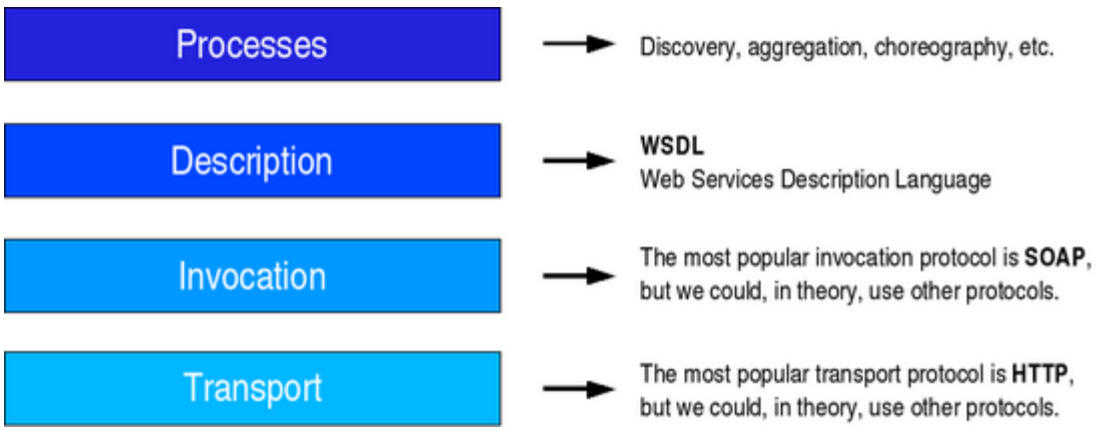
καταχώρηση, Υπηρεσίας . Μέσω του UDDI, οι Υπηρεσίες αυτές *δημοσιοποιούνται* ώστε να μπορούν να γίνουν προσβάσιμες .

3.2.5 WS-Security

Το ***WS-Security*** είναι ένα specification, το οποίο περιγράφει τον τρόπο με τον οποίο μπορούμε να «ενισχύσουμε» ένα SOAP μήνυμα , ώστε αυτό να προσφέρει μεγαλύτερη ασφάλεια . Μπορούμε δηλαδή να διασφαλίσουμε την *ακεραιότητα* των μηνυμάτων αυτών μέσω της ***ψηφιακής υπογραφής*** *(XML Digital Signature)* ή να προστατεύσουμε την *εμπιστευτικότητα* των μηνυμάτων μέσω κρυπτογράφησης. Οι μηχανισμοί αυτοί μπορούν να εφαρμοστούν σε οποιοδήποτε μοντέλο ασφάλειας , όπως το PKI (Public Key Interface). Οι πολιτικές ασφαλείας μπορούν να εφαρμοστούν ***τόσο σε επίπεδο δικτύου*** (*TLS/SSL*), όσο και ***σε επίπεδο μηνύματος*** *(WS-Security).*

3.3 Η αρχιτεκτονική των Web Services

Σε αυτό το κομμάτι θα αναλυθεί η αρχιτεκτονική ενός Web Service, δίνοντας περισσότερο φως στις έννοιες SOAP και WSDL. Τα βασικά στοιχεία ενός Web Service δίνονται στο παρακάτω σχήμα.



Εικόνα 3.2 Globus Toolkit 4

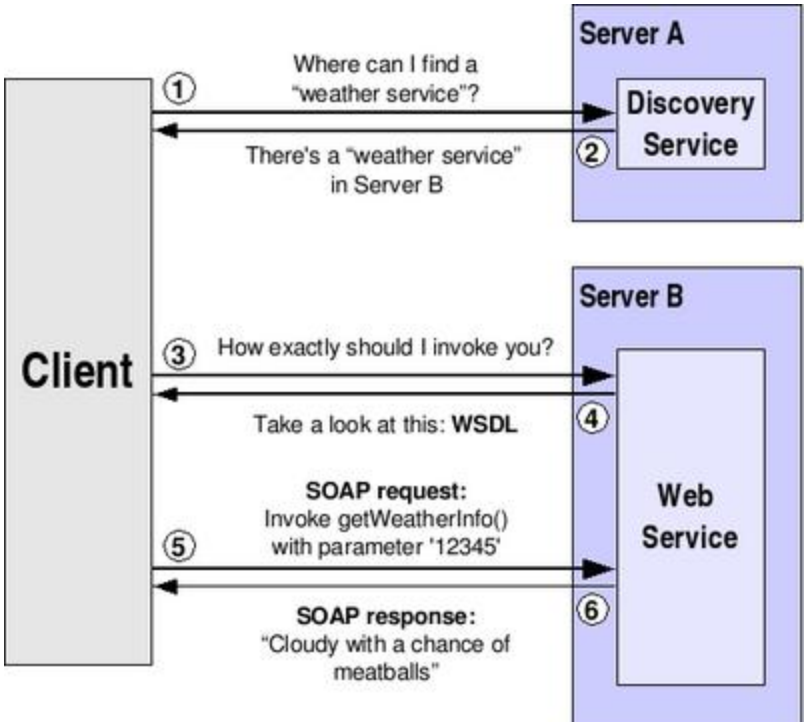
- ***Διεργασίες του Web Service (Service Processes):*** Το μέρος αυτό της αρχιτεκτονικής έχει να κάνει με περισσότερα από ένα Web Service. Παραδείγματος χάρη η αναζήτηση ενός

Web Service μας επιτρέπει την αναζήτηση ενος συγκεκριμένου Web Service μέσα σε ένα πλήθος από Web Services.

- ***Αυτοπεριγραφή του Web Service (Service Description):*** Ένα από τα πιο ενδιαφέροντα προσόντα των Web Services είναι ότι είναι αυτοπεριγραφόμενα. Αυτό σημαίνει ότι, όταν έχει εντοπίσει κάποιος ένα Web Service, είναι σε θέση να ζητήσει την αυτοπεριγραφή του μεταδίδοντας με αυτόν τον τρόπο προς τον ενδιαφερόμενο , τις λειτουργίες που υποστηρίζει και τον τρόπο κλήσης του. Αυτό το ελέγχει η περιγραφική γλώσσα του Web Service (Web Services Description Language -WSDL).
- ***Κλήση του Web Service (Service Invocation):*** Η κλήση ενός Web Service (και γενικά κάθε είδους κλήση σε ένα κατανεμημένο service όπως για παράδειγμα τα Enterprise Java Beans) περιλαμβάνει την μετάδοση μηνυμάτων μεταξύ client και server. Το SOAP (Simple Object Access Protocol) δηλώνει τον τρόπου που πρέπει να έχει η μορφή της αίτησης προς τον server καθώς και τον τρόπο που ο server θα πρέπει να μορφοποιεί της απαντήσεις του.
- ***Μετάδοση (Transport):*** Όλα αυτά τα μηνύματα πρέπει να μεταδοθούν με ένα τρόπο μεταξύ του client και του server. Το κομμάτι αυτό της αρχιτεκτονικής καλύπτεται με τη χρήση του πρωτόκολλου HTTP (Hyper Text Transfer Protocol), το ίδιο δηλαδή πρωτόκολλο που χρησιμοποιείται για την πρόσβαση των ιστοσελίδων στο Internet.

3.4 Ένα τυπικό παράδειγμα κλήσης Web Service

1. Ένας Client ενδέχεται να μην έχει γνώση για το ποιο Web Service θα καλέσει. Έτσι σαν πρώτο βήμα πρέπει να ερευνηθεί αν το Web Service είναι συμβατό με τις απαιτήσεις μας. Θα χρησιμοποιηθεί πάλι το παράδειγμα της αίτησης για πληροφορία για τις καιρικές συνθήκες που επικρατούν σε μια περιοχή. Αυτό επιχειρείται με κλήση ενός διερευνητικού Web Service.



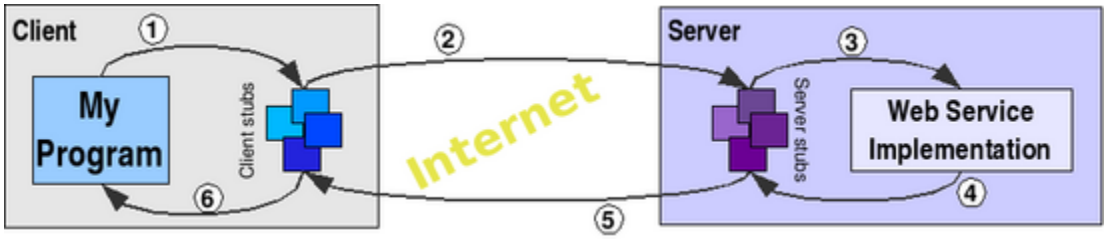
Εικόνα 3.3 Τυπικό παράδειγμα κλήσης Web Service

2. Αυτό το διερευνητικό Web Service θα στείλει μια απάντηση για το αν το Web Service ανταποκρίνεται στις απαιτήσεις του client. Είναι πλέον γνωστή η τοποθεσία του Web Service, αλλά δεν είναι γνωστό προς το παρόν ο τρόπος κλήσης του. Μπορεί είναι γνωστό ότι θα μας δώσει τα προγνωστικά καιρού για την τοποθεσία που θέλουμε, ωστόσο ο τρόπος κλήσης της υπηρεσίας αυτής παραμένει κάτι άγνωστο. Ίσως η μέθοδος να έχει την ονομασία `String getCityForecast(int CityPostalCode)` ή ακόμα και την ονομασία `getUSCityWeather(String cityName, bool isFarenheit)`. Για αυτόν τον λόγο πρέπει να γίνει μια αίτηση προς το Web Service για την αυτοπεριγραφή του.
3. Η απάντηση του Web Service είναι σε γλώσσα WSDL.
4. Τελικά γίνεται γνωστό προς τον Client η τοποθεσία του Web Service καθώς και ο τρόπος κλήσης του. Η κλήση ενός Web Service είναι διαμορφωμένη σύμφωνα με τη γλώσσα SOAP. Για αυτό το λόγω αποστέλλεται πρώτα μια ερώτηση SOAP η οποία ζητά πληροφορία για τα προγνωστικά καιρού μιας τοποθεσίας.

5. Το Web Service θα αποστείλει μια απάντηση SOAP η οποία θα περιλαμβάνει την πληροφορία που ζητήθηκε, ή ένα μήνυμα λάθους άν η ερώτηση SOAP ήταν λανθασμένη.

3.5 Ένα τυπικό παράδειγμα κλήσης Web Service (Ανάλυση)

Λαμβάνοντας ως δεδομένο ότι το Web Service έχει εντοπιστεί και ο Client είναι έτοιμος για κλήση του, παραθέτονται τα βήματα τα οποία ακολουθούνται μέχρι να έρθει σε πέρας η διαδικασία αυτή.



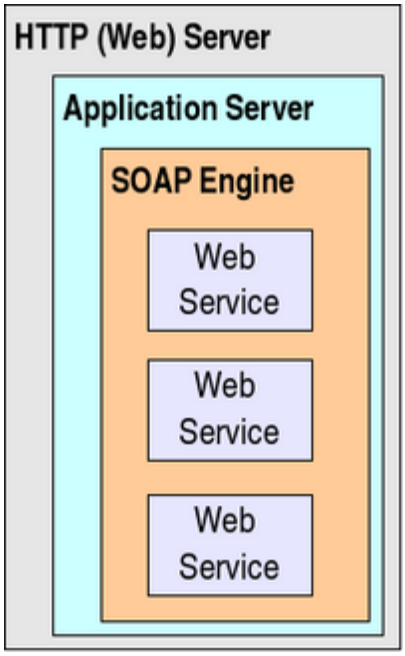
Εικόνα 3.4 Τυπικό παράδειγμα κλήσης Web Service Invocation (πιο λεπτομερές)

1. Όταν η εφαρμογή του Client επιθυμεί την κλήση του Web Service, θα μορφοποιήσει κατάλληλα την αίτηση του σύμφωνα με το πρότυπο που επιβάλει το SOAP. Αυτή η διαδικασία ονομάζεται ως διαδικασία marshalling ή serializing.
2. Η αίτηση SOAP αποστέλλεται μέσω του HTTP. Ο server λαμβάνει την αίτηση SOAP και την τροποποιεί σε κατάλληλη μορφή ώστε να μπορεί να αποδέσμευση την αίτηση της εφαρμογής του client.
3. Όταν επέλθει η αίτηση του client στην κατάλληλη μορφή τότε ο server εκτελεί την αντίστοιχη μέθοδο προς την αίτηση του client.
4. Το αποτέλεσμα της μεθόδου αυτής τροποποιείται κατάλληλα ώστε να είναι συμβατή με μια απάντηση SOAP.
5. Η απάντηση SOAP αποστέλλεται μέσω HTTP. Ο client λαμβάνει την απάντηση από τον server και την τροποποιεί κατάλληλά ώστε να αποδέσμευση την απαραίτητη πληροφορία για την εφαρμογή που κάλεσε το Web Service.

6. Τελικά ο client λαμβάνει την πληροφορία και την χρησιμοποιεί για τους δικούς τους σκοπούς.

3.5.1 Ο Εξυπηρετητής

Σε αυτό το μέρος γίνεται μια ανάλυση για τη δομή που έχει ένας εξυπηρετητής (server) για Web Service.



Εικόνα 3.5 Μορφή μιας εφαρμογής Web Service από το μέρος του Server

- ***Web Service:*** Το Web Service είναι και το πρωταρχικό στοιχείο. Είναι πλέον γνωστό ότι ένα Web Service είναι μια εφαρμογή λογισμικού που περιέχει ένα σύνολο *λειτουργιών.* Αν το Web Service ήταν υλοποιημένο με τη γλώσσα προγραμματισμού Java , τότε θα ήταν μια κλάση τύπου Java και οι λειτουργίες του θα ήταν Java μέθοδοι) . Προφανώς, χρειάζεται ένα σύνολο clients ώστε να είναι σε θέση να κληθούν όλες αυτές τις λειτουργίες. Ωστόσο δεν είναι αρμοδιότητα του Web Service η μετατροπή των SOAP αιτήσεων, ούτε τη δημιουργία SOAP απαντήσεων. Αυτή είναι αρμοδιότητα του SOAP engine.
- ***Soap Engine:*** Πρόκειται για εκείνο το κομμάτι λογισμικού που διαχειρίζεται τις αιτήσεις και της απαντήσεις SOAP. Ένα πολύ καλό παράδειγμα Soap Engine είναι το Apache Axis. Ωστόσο η λειτουργία ενός Soap Engine περιορίζεται στην διαχείρηση του SOAP. Για να μπορέσει να λειτουργήσει ως ένας server ο οποίος να λαμβάνει αιτήσεις από

διαφορετικούς clients, το Soap Engine πρέπει να λειτουργήσει εντός ενός Application Server.

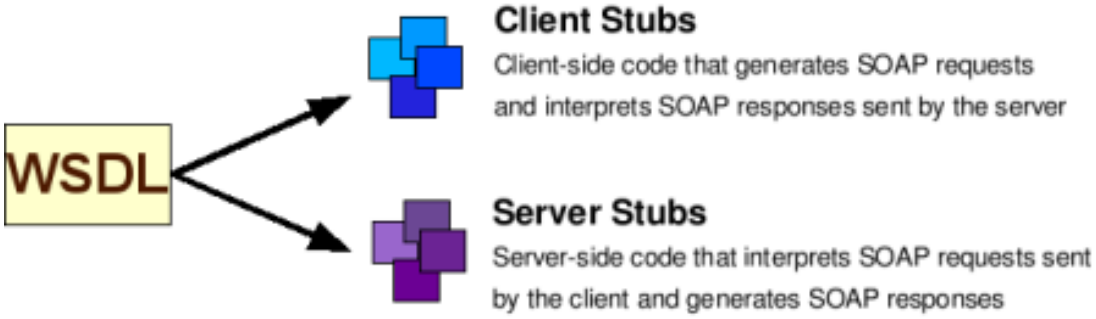
- ***Application Server:*** Πρόκειται για το μέρος του server το οποίο επιτρέπει στις εφαρμογές να απασχολούνται από διαφορετικούς χρήστες. Το Soap Engine εκτελείται ως μια εφαρμογή εντός του Application Server . Ένα καλό παράδειγμα είναι ο server Jakarta Tomcat, ένα container για Java Servlets και Java Server Pages του οποίου η χρήση συνδυάζεται μαζί με τον Apache Axis. Οι περισσότεροι application servers διαθέτουν λειτουργίες HTTP, οπότε μπορούμε να έχουμε στη διάθεσή μας εγκαθιστώντας ένα Soap Engine και ένα Application Server. Αν όμως ο Application Server δεν υποστηρίζει λειτουργικότητα HTTP τότε χρειαζόμαστε ένα HTTP Server.
- ***HTTP Server:*** Ο HTTP Server καλείται πολύ συχνά και ως Web Server. Πρόκειται για ένα server ο οποίος είναι αρμόδιος για τη διαχείριση των HTTP μηνυμάτων. Ένα καλό παράδειγμα είναι ο Apache HTTP Server, ένας από τους πιο διάσημους Web Servers στο Internet.

3.6 Ένα πρακτικό ζήτημα

Στον προγραμματισμό Web Services υπάρχουν πολλά πρωτόκολλα και γλώσσες προγραμματισμού που μπορούν να χρησιμοποιηθούν , αλλά οι προγραμματιστές συνήθως προγραμματίζουν στην αγαπημένη τους γλώσσα προγραμματισμού και ίσως να χρειαστεί να γράψουν κάποια WSDL σε μερικές περιπτώσεις. Ο κώδικας του SOAP , από την άλλη μεριά πάντα παράγεται και μεταφράζετε αυτόματα . Όταν φτάσουμε στο προγραμματιστικό σημείο όπου η εφαρμογή μας θέλει να καλέσει μια Web Service , παραχωρούμε την εκτέλεση αυτή σε ένα κομμάτι κώδικα που ονομάζεται ***stub*** και το οποίο είναι σε θέση να μεταφράζει τις κλήσεις μας από και προς τον SOAP. Υπάρχουν πάρα πολλά εργαλεία διαθέσιμα που θα παράγουν τα stubs για εμάς τα οποία συνήθως βασίζονται στην περιγραφή WSDL της Web Service.

Η χρήση των stubs απλοποιεί τις εφαρμογές-πελάτες σε τεράστιο βαθμό. Δεν χρειάζεται να γραφεί πολύπλοκος κώδικας για τα προγράμματα-πελάτες που παράγει δυναμικά τις αιτήσεις SOAP και μεταγλωττίζει τις απαντήσεις SOAP(και το ίδιο ακριβώς συμβαίνει και στην πλευρά του εξυπηρετητή). Όποτε απαιτείται μόνο η παραγωγή κώδικα για το πρόγραμμα-πελάτη με τις λειτουργίες που απαιτούνται και η λειτουργία της

επικοινωνίας αφήνεται στα stubs (τα όποια και δεν πρέπει να γραφούν αφού παράγονται μόνα τους από την WSDL περιγραφή της υπηρεσίας).



Εικόνα 3.6 Τα stubs για τον Client και τον Server παράγονται από το αρχείο WSDL

Τα *stubs* γενικά παράγονται μόνο μια φορά έτσι δεν χρειάζεται να περάσουμε από την διαδικασία ανακάλυψης των Web Services κάθε φορά που χρειάζεται να καλέσουμε μια Web Service και να παράγουμε τα Client stubs κάθε φορά που καλούμε μια υπηρεσία. Κατά βάση θα χρειαστεί να περάσουμε την διαδικασία αναζήτησης μία φορά , μετά να γίνει η παραγωγή των stubs μία φορά(βασισμένη στο WSDL της υπηρεσίας που ανακαλύψαμε) και μετά μπορούμε να επαναχρησιμοποιήσουμε τα stubs όσες φορές χρειαστεί(εκτός και αν ο συντονιστής της υπηρεσίας αλλάξει το interface της υπηρεσίας και κατά συνέπεια την WSDL περιγραφή της). Στην πραγματικότητα τα σενάρια κλήσης μιας Web Service ενδέχεται να γίνουν πολύ πιο πολύπλοκα από αυτό το απλό παράδειγμα.

4 WSRF - Web Services Resource Framework

4.1 Γενικά

Το 2002-2003, το ***OGSI (Open Grid Service Architecture)*** κάτω από την γενικότερη επίβλεψη του OGSA, προσπάθησε να συνδυάσει “παραδοσιακές” πλατφόρμες για την υλοποίηση του Grid Computing, όπως το Globus και το Legion, με την αναδυόμενη τεχνολογία των Web Services. Το OGSI προσπάθησε να εκμεταλλευτεί τα XML πρωτόκολλα, που συναντάμε στα Web Services.

Το OGSI εμφανίστηκε το 2003 αλλά γρήγορα αντικαταστάθηκε από το WSRF.Ο λόγος της αποτυχίας βρισκόταν στο γεγονός , ότι τα Grid services που μπορούσαν να δημιουργηθούν στο πλαίσιο του OGSI δεν ήταν «καθαρά» Web Services, ή για να είμαστε πιο ακριβείς δεν ήταν υποσύνολο των Web Services. Με άλλα λόγια, μια OGSI-συμβατή service δεν μπορούσε να επικοινωνήσει με ένα μη-συμβατό OGSI service. Το OGSI απαιτούσε τροποποίηση του WSDL, το οποίο μετονομαζόταν σε Grid-WSDL (GWSDL). Γι’ αυτό το λόγο, κάποια εμπορικά και μη εργαλεία, που χρησιμοποιούνταν για τα Web services, δεν ήταν πλέον συμβατά και έπρεπε να δημιουργηθούν άλλα από την αρχή.

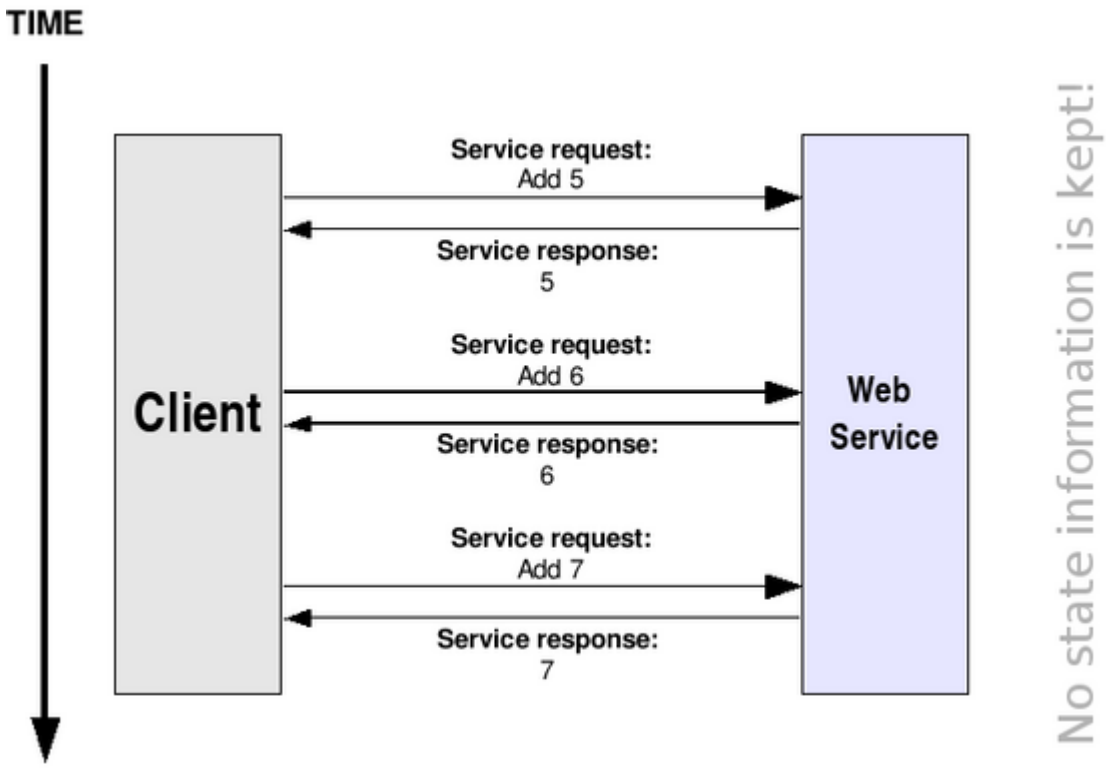
Τον Ιανουάριο του 2004, μια ομάδα από την ***IBM*** και την ***Globus Alliance*** εισήγαγε το ***Web Services Resource Framework (WSRF),*** σαν μια προσπάθεια να επαναπροσδιοριστούν κάποιες προδιαγραφές του OGSI, ώστε να είναι συμβατά με την πραγματικότητα , των αναδυόμενων τότε, Web Services. Ο βασικός στόχος πίσω από το WSRF, ήταν η διατήρηση της κατάστασης. Στο περιβάλλον των απλών Web Services, η κατάσταση δεν διατηρείται ή αν χρειάζεται να γίνει κάτι τέτοιο, τότε αυτό γίνεται σε ανώτερο επίπεδο όπως το επίπεδο Εφαρμογής. Η βασική διαφορά, δηλαδή, WSRF και OGSI , βρίσκεται στο ότι το WSRF δεν απαιτεί τροποποίηση των ήδη υπαρχόντων εργαλείων των Web Services σε αντίθεση με το OGSI, όπως αναφέρθηκε πριν. Τα Grid services, που

δημιουργούνται με το WSRF, είναι Web Services, με κάποια επιπλέον χαρακτηριστικά, ώστε να μπορεί να διατηρείται η κατάσταση.

4.2 Stateless – Statefull Web Services στο Globus Toolkit

Γνωρίσαμε προηγουμένως τα Web Services και τη σημαντικότητα που κατέχουν ως επιλογή για Internet εφαρμογές με loosely coupled client και servers. Το γεγονός αυτό καθιερώνει τα Web Services ως την φυσική επιλογή για την επόμενη γενιά των Grid εφαρμογών. Ωστόσο δεν πρέπει να παραληφθεί ότι τα Web Services έρχονται μαζί με κάποιους περιορισμούς. Ουσιαστικά τα απλά Web Services όπως έχουν καθοριστεί από το W3C χαρακτηρίζονται ανεπαρκείς για την κατασκευή μιας Grid εφαρμογής. Αυτός είναι και ο λόγος ύπαρξης του WSRF, η βελτίωση αρκετών παραμέτρων των Web Services καθιστώντας τα ικανά για την κατασκευή και χρήση εφαρμογών Grid.

Τα απλά Web Services είναι *stateless* (έστω και αν θεωρητικά, δεν υπάρχει αρχιτεκτονική για Web Services η οποία να διευκρινίζει να είναι stateful). Ο όρος *stateless* σημαίνει ότι το Web Service δεν είναι σε θέση να κρατά σε μνήμη πληροφορίες, από την μια κλήση στην άλλη. Για να γίνει πιο ξεκάθαρη η έννοια του όρου *stateless* ας σκεφτεί κάποιος ένα πολύ απλό Web Service, του οποίου η λειτουργία του είναι αθροιστής ακεραίων. Αυτός ο αθροιστής αρχικοποιείται στο μηδέν και ζητούμενο είναι να αθροίσουμε αριθμούς με αυτόν.

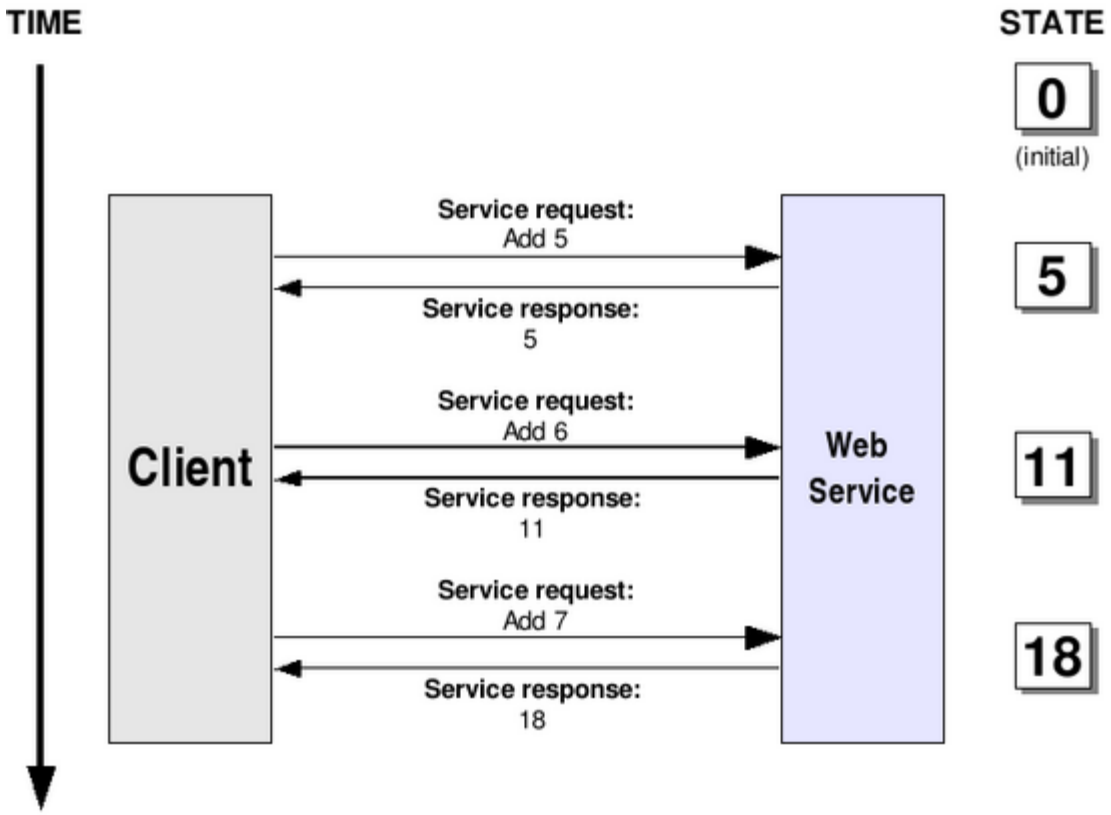


Εικόνα 4.1 Κλήση Stateless Web Service

Υποθέτοντας ότι έχουμε την πράξη της πρόσθεσης η οποία λαμβάνει την τιμή που θέλουμε να προσθέσουμε και κατόπιν της πράξης επιστρέφει την τιμή που προκύπτει από τον αθροιστή. Παρατηρώντας το παρακάτω σχήμα κάτι τέτοιο φαίνεται να λειτουργεί στην πρώτη κλήση του Web Service. Όμως λαμβάνοντας υπόψιν ότι το Web Service είναι *stateless* , στις επόμενες κλήσεις δεν είναι σε θέση να γνωρίζουν τα αποτελέσματα προηγουμένων κλήσεων. Για αυτόν τον λόγο στην δεύτερη κλήση που προστίθεται το 6, λαμβάνεται ως απάντηση του Web Service προς τον client το 6, αντί του 11 (το οποίο θα ήταν και το αναμενόμενο αν το Web Service ήταν *stateful*).

Το γεγονός ότι τα Web Services είναι *stateless* δεν είναι απαραίτητα ένα σημαντικό μειονέκτημα. Υπάρχουν πολλές εφαρμογές στις οποίες δεν υπάρχει λόγος για την ύπαρξη ενός *stateful* Web Service. Ένα Web Service για προγνωστικά καιρού, είναι ένα καλό παράδειγμα στο οποίο δεν υπάρχει λόγος ύπαρξης *stateful* Web Service.

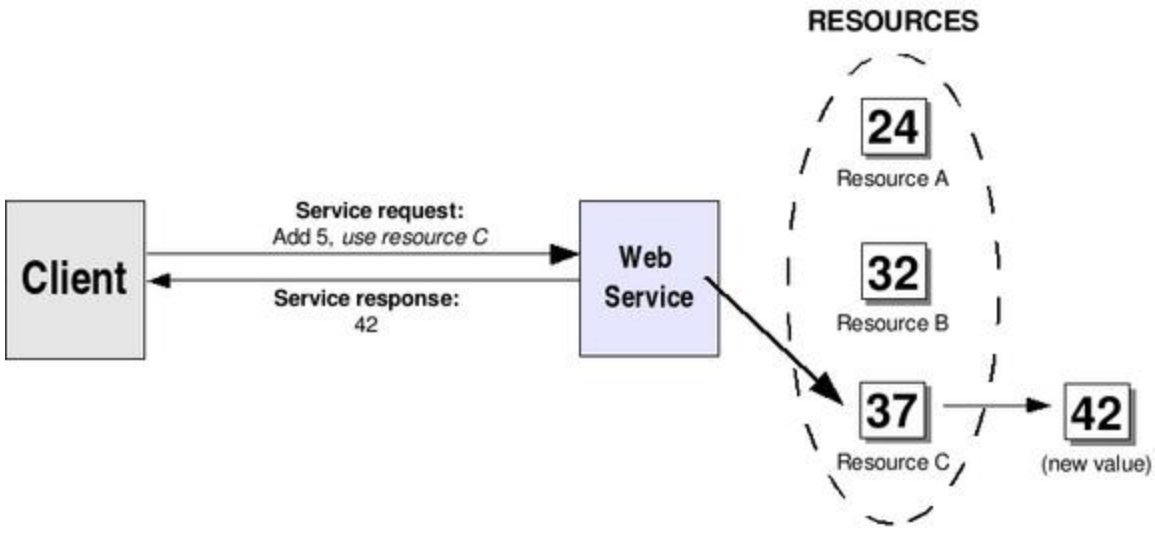
Όσον αφορά τις Grid εφαρμογές όμως η χρήση *stateful* Web Services κρίνεται απαραίτητη. Επιθυμητό κρίνεται για τα Web Services, με βάση τον παραπάνω συλλογισμό να συγκρατούν με κάποιο τρόπο πληροφορία. Αυτό έρχεται σε αντιπαράθεση με το γεγονός ότι τα Web Services είναι *stateless*. Προσδίδοντας την ιδιότητα στα Web Services να συγκρατούν πληροφορία σε κάποιο state ενώ παράλληλα πρέπει να είναι *stateless*, αποτελεί ένα περίπλοκο πρόβλημα. Η λύση σε αυτό το πρόβλημα δίνεται αν κρατήσουμε τα Web Services από την πληροφορία που φυλάσσεται σε κάποιο state σε δύο εντελώς ανεξάρτητα επίπεδα. Έτσι αντί να εμπλουτίσουμε ένα Web Service με το state, κάτι το οποίο μετατρέπει το Web Service σε *stateful*, θα κρατήσουμε την πληροφορία σε ένα αποθηκευτικό πόρο. Κάθε αποθηκευτικός πόρος θα έχει ένα *μοναδικό κλειδί* με το οποίο θα ξεχωρίζει από τους υπόλοιπους, έτσι δίνουμε την ιδιότητα *stateful* στα Web Services τα οποία θα ανατρέξουν για κάποια πληροφορία που πρέπει να αποθηκευτεί όχι σε κάποιο state εντός του Web Service, αλλά σε ένα συγκεκριμένο αποθηκευτικό πόρο εκτός του.



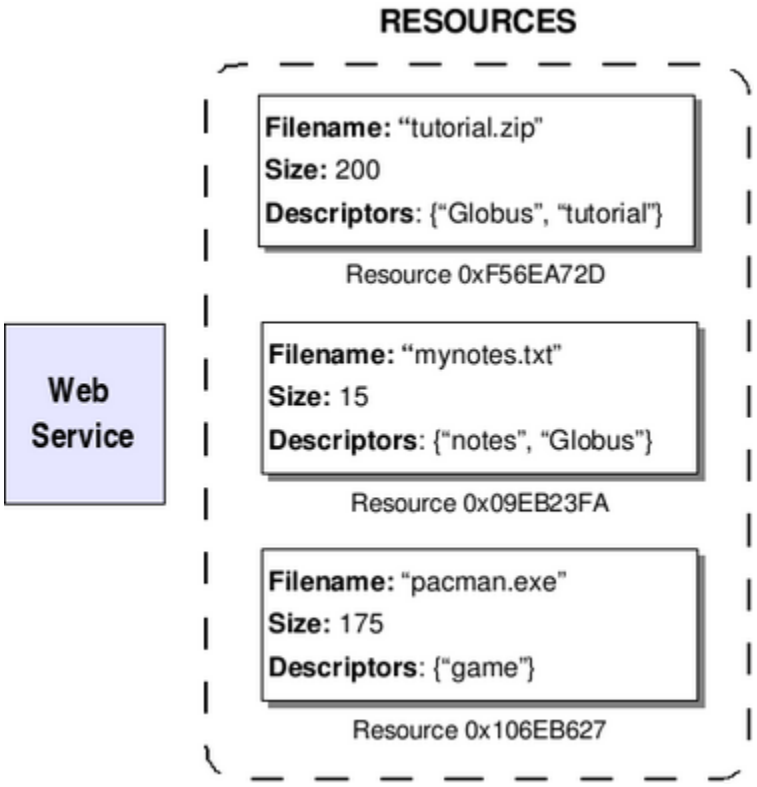
Εικόνα 4.2 Κλήση Web Service τύπου statefull

Στο προηγούμενο παράδειγμα του αθροιστή, το Web Service μπορεί να έχει στη διάθεση του τρεις διαφορετικούς αποθηκευτικούς πόρους. Έτσι ανάλογα με το ποιός καλεί το Web Service, θα καλείται ο ανάλογος αποθηκευτικός πόρος που θα ανταποκρίνεται στα ζητούμενα του client. Στο παρακάτω σχήμα φαίνεται πως ο client ζητά την κλήση της πρόσθεσης συγκρατώντας πληροφορίες σε ένα συγκεκριμένο αποθηκευτικό πόρο τον οποίο ονομάζουμε Α. Όταν το Web Service λάβει μια αίτηση για πρόσθεση, ανατρέχει στον πόρο Α, ώστε να πραγματοποιήσει την πρόσθεση λαμβάνοντας υπόψη την υπάρχουσα πληροφορία που έχει καταχωρηθεί στον Α. Οι αποθηκευτικοί πόροι μπορεί να είναι μνήμη, σκληρός δίσκος, ή ακόμα και βάση δεδομένων. Αξιοσημείωτο είναι ότι ένα Web Service μπορεί να έχει πρόσβαση σε παραπάνω από ένα πόρους.

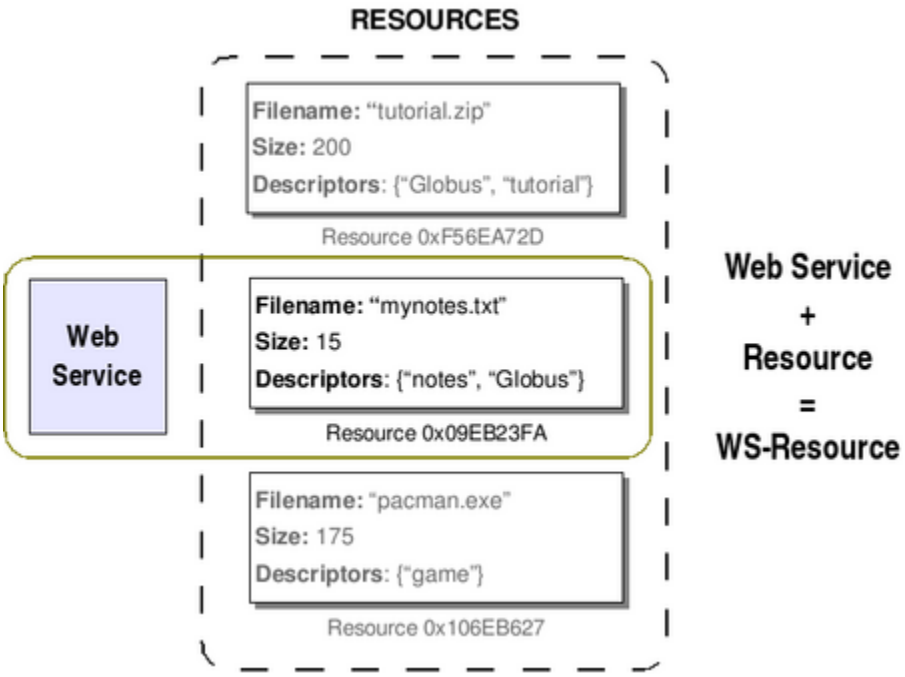
Ερώτημα αποτελεί ο τρόπος που ένας client καθορίζει τον αποθηκευτικό πόρο που πρέπει να χρησιμοποιηθεί. Το ταίριασμα ενός Web Service με ένα αποθηκευτικό πόρο ονομάζεται *WSResource* . Άρα αυτό το οποίο είναι το ζητούμενο είναι πως κατευθύνουμε τα Web Services προς τα *WS-Resources*. Ο πλέον ιδανικός τρόπος ονομάζεται *WS-Addressing* ο οποίος προσφέρει ένα πιο ευέλικτο τρόπο διευθυνσιοδότησης Web Services.



Εικόνα 4.3 Κατανομή πόρων σε statefull εφαρμογές



Εικόνα 4.4 A Web Service με διάφορες πηγές

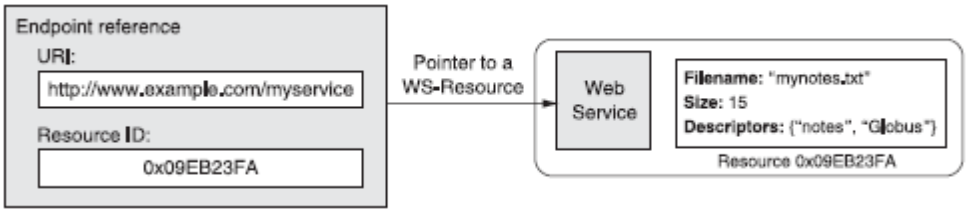


Εικόνα 4.5 WS-Resource

Εικόνα 4.6 Endpoint – reference

Το WS-Addressing καθορίζει μια δομή με την ονομασία *endpoint – reference* η οποία επιτρέπει να καθορίσουμε μια διεύθυνση προς το Web Service. Το παραπάνω σχήμα δίνει μια καλύτερη εικόνα για την έννοια του *endpoint – reference*.

Το ερώτημα για το πως ένα Web Service αναζητά το κατάλληλο αποθηκευτικό πόρο παραμένει αναπάντητο. Τη λύση δίνει το *endpoint – reference* το οποίο είναι σε θέση να συμπεριλάβει και άλλες πληροφορίες εκτός του URI, όπως και τον τρόπο αναγνώρισης πόρων. Ένα τέτοιο *endpoint – reference* ονομάζεται ως *WS-Resource-qualified endpoint reference* και φαίνεται καλύτερα στο παρακάτω σχήμα.



Εικόνα 4.7 WS-Resource σε σύνδεση με endpoint reference

Ουσιαστικά ένα endpoint reference (ERP, για συντομία) είναι ένας δείκτης σε ένα *WSResource*. Για τους κατασκευαστές ενός Web Service, είναι απαραίτητο να γνωρίζουνε τη δομή ενός ERP. Ωστόσο για τις εφαρμογές του client τα ERP πρέπει να χρησιμοποιούνται ως έχουν, χωρίς καμία επέμβαση στη δομή τους, ούτε χρειάζεται ο χρήστης να γνωρίζει οτιδήποτε για την υλοποίηση των ERP.

```
      <ns1:ASLAResourceReference xsi:type="ns2:EndpointReferenceType"
xmlns:ns1="http://www.globus.org/namespaces/irmos/asla/ASLAService"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns2="http://www.w3.org/2005/08/addressing">
      <ns2:Address
xsi:type="ns2:AttributedURI">http://192.108.38.54:8080/wsrf/services
/irmos/asla/factory/ASLAService</ns2:Address>
```

```
      <ns2:ReferenceParameters
xsi:type="ns2:ReferenceParametersType">
      <ns1:ASLAResourceKey
xmlns:ns1="http://www.globus.org/namespaces/irmos/asla/factory">1632
5775</ns1:ASLAResourceKey>
      </ns2:ReferenceParameters>
      </ns1:ASLAResourceReference>
```

Κώδικας 1 : Παράδειγμα αρχείου endpoint reference (EPR)

Όσον αφορά τους αποθηκευτικούς πόρους, τα δεδομένα μέσα σε αυτούς ονομάζονται resource properties και μας παρέχουν μια όψη της παρούσας κατάστασης του αποθηκευτικού πόρου. Για παράδειγμα τρεις παράμετροι των resource properties είναι το όνομα του αρχείου, το μέγεθος και η περιγραφή του. Τα resource properties χρησιμεύουν για την αποθήκευση των παρακάτω πληροφοριών:

- ***Δεδομένα για το Service:*** Παροχή πληροφοριών για την παρούσα κατάσταση του Web Service, όπως αποτελέσματα από λειτουργίες του service, ενδιάμεσα αποτελέσματα, πληροφορίες κατά την εκτέλεση κτλ. Όνομα του αρχείου, το μέγεθος και η περιγραφή του είναι κάποια από αυτά τα δεδομένα.
- ***Πληροφορίες για τα δεδομένα του Web Service (Metadata):*** Πληροφορίες σχετικά με τα δεδομένα του Web Service, όπως για παράδειγμα την ημερομηνία αλλαγής ενός αρχείου ή τον χρήστη που το άλλαξε.
- ***Πληροφορίες που απαιτούνται για την διαχείριση της κατάστασης του Service:*** Είναι ένας τύπος πληροφορίας παρόμοιος με τα metadata, αλλά αναφέρεται στον αποθηκευτικό πόρο ως σύνολο. Ένα τέτοιο παράδειγμα είναι ο χρόνος λήξης του resource , κάτι που σημαίνει μια παράμετρο που δείχνει πότε ένας πόρος έχει τερματίσει την λειτουργία του.

4.2.1 Προδιαγραφές του Web Services Resources Framework

Το Web Services Resources Framework είναι μια συλλογή τεσσάρων διαφορετικών προδιαγραφών , όλες εκ των οποίων σχετίζονται , με τον έναν ή τον άλλον τρόπο , με την διαχείριση των WS-Resources.

WS-ResourceProperties

Η προδιαγραφή αυτή παρέχει τα interfaces που επιτρέπουν στους χρήστες την πρόσβαση , την δυνατότητα αλλαγών και επερωτήσεων πάνω στα Resource Properties.

WS-ResourceLifetime

Οι πόροι δεν έχουν τετριμμένο κύκλο ζωής. Με άλλα λόγια δεν είναι στατικές οντότητες που δημιουργούνται κατά την εκκίνηση του Server και καταστρέφονται όταν αυτός τερματίσει. Οι πόροι μπορούν να δημιουργηθούν και να καταστραφούν οποιαδήποτε στιγμή. Το WS-ResourceLifetime παρέχει βασικούς μηχανισμούς για την διαχείριση του κύκλου ζωής των Resources.

WS-ServiceGroup

Πολλές φορές ενδιαφερόμαστε να διαχειριστούμε ομάδες από Web Services ή ομάδες από WS-Resources , και η εκτέλεση εργασιών όπως η «προσθήκη υπηρεσίας στην ομάδα» , η «αφαίρεση υπηρεσίας από την ομάδα» και (πιο σημαντική) η «εύρεση υπηρεσίας στην ομάδα που πληροί τις τάδε προϋποθέσεις». Το WS-ServiceGroup προδιαγράφει επακριβώς πώς να ομαδοποιηθούν οι υπηρεσίες ή οι WS-Resources. Παρόλο που οι λειτουργίες που περιγράφει αυτή η προδιαγραφή είναι εντελώς βασικές , αποτελούν την βάση για πιο προχωρημένες υπηρεσίες αναζήτησης (όπως η IndexService του Globus Toolkit 4) που μας επιτρέπουν να ομαδοποιήσουμε διαφορετικές υπηρεσίες και να τις αναζητήσουμε από ένα σημείο εισόδου (Την ομάδα υπηρεσιών-ServiceGroup).

WS-BaseFaults

Η προδιαγραφή αυτή παρέχει έναν καθορισμένο τρόπο για την αναπαράσταση των λαθών που συμβαίνουν κατά την κλήση των WS-Service.

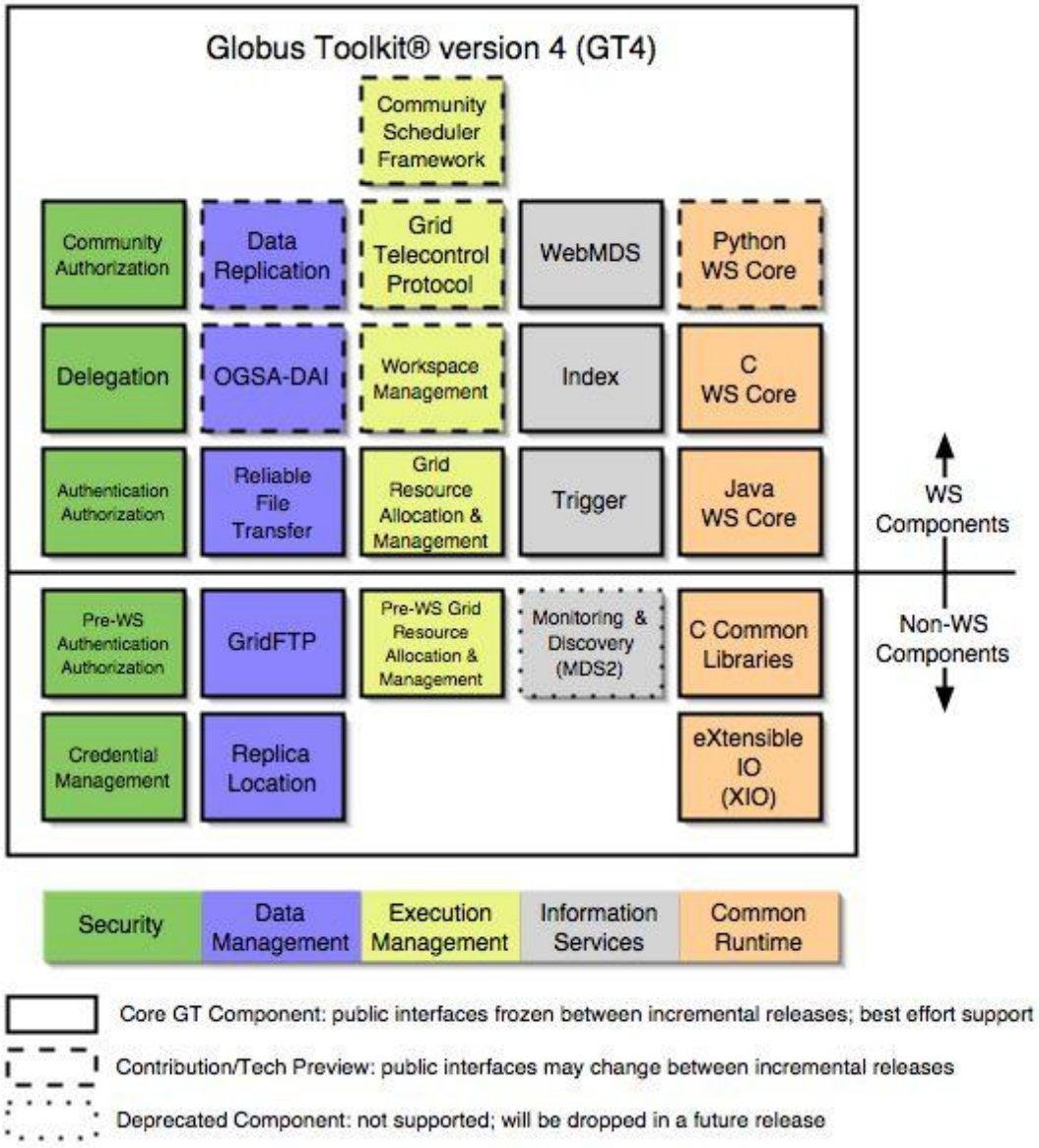
4.2.2 Επιπλέον Προδιαγραφές

WS-Notification

Το WS-Notification είναι μια άλλη συλλογή προδιαγραφών που, αν και δεν είναι μέρος του WSRF, συνδέεται στενά με αυτό. Αυτή η προδιαγραφή επιτρέπει σε μια υπηρεσία Web να είναι ρυθμισμένη ως ένας *notification producer*, και ορισμένα προγράμματα-πελάτες να είναι notification consumers (ή συνδρομητές). Αυτό σημαίνει ότι εάν προκύψει μια αλλαγή στην υπηρεσία Web (ή, πιο συγκεκριμένα, σε έναν από τους WSResources), αυτή η αλλαγή κοινοποιείται σε όλους τους συνδρομητές (δεν κοινοποιούνται όλες οι αλλαγές ,αλλά μόνο εκείνες που ο προγραμματιστής της Web Service θέλει).

WS-Addressing

Το WS-Addressing παρέχει μηχανισμούς για να διευθυνσιοδοτήσουμε μια Web Service με τρόπους πιο ευέλικτους από το απλό URI. Συγκεκριμένα μπορούμε να χρησιμοποιήσουμε το WS-Addressing για να δώσουμε μια διεύθυνση σε μια Web Service και έναν πόρο (WS-Resource).



Εικόνα 4.1.7 Globus Toolkit

5 Irmos Project (Interactive Realtime Multimedia Applications on Service Oriented Infrastructures)

5.1 Λίγα λόγια για το έργο

Εικόνα 5.1.1 Irmos Project

Το Irmos Project είναι ένα έργο της τεχνολογίας Grid που προσπαθεί να επιτύχει την αλληλεπίδραση σε πραγματικό χρόνο μεταξύ ανθρώπων και εφαρμογών πάνω σε μια Υποδομή προσανατολισμένη στην Υπηρεσία(Service Oriented Infrastracture).

Το Irmos Project ξεχωρίζει γιατί παρέχει μια ενοποιημένη Υποδομή σε πολλούς οργανισμούς με ιδιότητες πραγματικού χρόνου σε όλα τα επίπεδα της, εξασφαλίζει την Ποιότητα της Υπηρεσίας και επιτρέπει την αυτοματοποίηση διαδικασιών των επιχειρήσεων. Επίσης καινοτομεί γιατι παρέχει μια Υποδομή προσανατολισμένη στην Υπηρεσία με δυνατότητες πραγματικού χρόνου,η οποία χρησιμοποιεί τους πόρους του δικτύου έξυπνα και βάση της παρεχόμενης Ποιότητας Υπηρεσίας, και παρέχει ένα περιβάλλον ανάπτυξης εφαρμογών SOI πραγματικού χρόνου.

5.2 Έννοια και στόχοι του έργου

Ο απώτερος στόχος του Irmos είναι να επιτρέπει την αλληλεπίδραση σε «πραγματικό χρόνο» μεταξύ των ανθρώπων και των εφαρμογών σε μια υποδομή λογισμικού προσανατολισμένη στις Υπηρεσίες , όπου η αποθήκευση , ή επεξεργασία και οι δικτυακές ανάγκες μπορούν να συνδυαστούν και να παραδοθούν στους τελικούς χρήστες σε ένα εγγυημένο επίπεδο ποιότητας.

Παραδοσιακά , ο όρος «*πραγματικός χρόνος*» αναφέρεται στα «*σκληρά*» συστήματα πραγματικού χρόνου, όπου και μία απλή παραβίαση της ηθελημένης χρονικής συμπεριφοράς θεωρείται απαράδεκτη, παραδείγματος χάριν επειδή οδηγεί σε ολική αποτυχία , πιθανότατα κοστίζοντας και ανθρώπινες ζωές. Ωστόσο, υπάρχει και ένα μεγάλο φάσμα εφαρμογών που επίσης έχουν πιεστικές και αυστηρές ανάγκες χρονισμού και απόδοσης , αλλά κάποιες μικρές αποκλίσεις στην Ποιότητα της Υπηρεσίας(Quality of Service-QoS) είναι αποδεκτές , δεδομένου ότι αυτές είναι κατανοητές και διαχειρίζονται προσεκτικά. Αυτά είναι τα «μαλακά» συστήματα πραγματικού χρόνου και περιλαμβάνουν ευρεία γκάμα αλληλεπιδραστικών και συνεργατικών εργαλείων και περιβαλλόντων, όπως η ταυτόχρονη σχεδίαση και απεικόνιση στον τομέα των μηχανικών , η παραγωγή ταινιών στην δημιουργική βιομηχανία , και το περιβάλλον πολλών χρηστών στην εκπαίδευση και τα ηλεκτρονικά παιχνίδια. Συγκεκριμένα το Irmos εστιάζει στις αλληλεπιδραστικές μαλακές

εφαρμογές πραγματικού χρόνου όπου ένας ή περισσότεροι χρήστες αλληλεπιδρούν με την εφαρμογή και μεταξύ τους.

Οι *μαλακές* εφαρμογές πραγματικού χρόνου παραδοσιακά αναπτύσσονται χωρίς καμία μεθοδολογία πραγματικού χρόνου ή χωρίς καμία υποστήριξη χρόνου εκτέλεσης από την υποδομή στην οποία εκτελούνται. Το αποτέλεσμα είναι είτε ακριβό και εξειδικευμένο υλικό που πρέπει να αγοραστεί ώστε να εξασφαλιστεί ένα καλό επίπεδο αλληλεπίδρασης και απόδοσης , ή χρησιμοποιούνται πόροι γενικής χρήσης σαν συμβιβαστική λύση και δεν υπάρχει τρόπος να εγγυηθεί ή να ελεγχθεί η συμπεριφορά της εφαρμογής.

Το *Irmos* στοχεύει να σπάσει αυτή την πρακτική επιτρέποντας στης μαλακές εφαρμογές πραγματικού χρόνου να παρέχονται μέσα από αγορές που περνάνε τα σύνορα των οργανισμών με μια υποδομή βασισμένη στην Υπηρεσία που επιτρέπει την αλληλεπίδραση σε πραγματικό χρόνο ενός διαμερισμένου(γεωγραφικά) συνόλου ανθρώπων και πόρων.

Το *Irmos* ξεχωρίζει από τα σημερινά συστήματα βασισμένα στην Υπηρεσία λόγω των ακόλουθων χαρακτηριστικών κλειδιά:

i. Το *Irmos* θα επιτρέπει τον διαμοιρασμό των αλληλεπιδραστικών εφαρμογών πραγματικού χρόνου πέρα από τα σύνορα των οργανισμών , αντί της χρήσης ενός εξειδικευμένου , ακριβού υλικού που βρίσκεται σε έναν χώρο.
i. Οι επιχειρήσεις θα μπορούν να έρχονται σε επαφή αποδοτικά και γρήγορα χρησιμοποιώντας το Irmos για να ταυτοποιήσεων , να συμφωνήσουν και να παρέχουν εφαρμογές πραγματικού χρόνου αποφεύγοντας παρατεταμένες διαπραγματεύσεις ή ρήτρες υπηρεσιών.
ii. Οι πάροχοι θα είναι σε θέση να παρέχουν υπηρεσίες που είναι αποδοτικές ως προς το κόστος και έχουν εγγυημένη ποιότητα Υπηρεσίας χρησιμοποιώντας το Irmos που θα τους παρέχει πλήρη έλεγχο στους πόρους τους.
iii. Το *Irmos* δίνει σε όλους τους συμμετέχοντες στις δια-οργανικές αγορές που περιλαμβάνουν τους παρόχους υπηρεσιών και τους καταναλωτές την πεποίθηση ότι οι διαδραστικές εφαρμογές πραγματικού χρόνου θα παρέχονται σε ένα προβλεπόμενο, αξιόπιστο και αποδοτικό τρόπο.
iv. Το *Irmos* παρέχει μια εξειδικευμένη προσέγγιση που απευθύνεται στον πραγματικό χρόνο σε όλα τα επίπεδα(δίκτυο, επεξεργασία δεδομένων , αποθήκευση ,

εφαρμογή , ροή εργασιών και επιχειρήσεις) ώστε να επιτρέπει την κατασκευή ολοκληρωμένων και από άκρη σε άκρη λύσεων χρησιμοποιώντας μία και μοναδική υποδομή.

Το *Irmos* project θα εφαρμόσει μεθοδολογίες πραγματικού χρόνου και τεχνολογίες που είναι κατάλληλες για συστήματα πραγματικού χρόνου στο πλαίσιο των Υποδομών προσανατολισμένες στην Υπηρεσία(***Service Oriented Infrastacture-SOI***). Θα δημιουργηθούν οι θεμέλιες δομές για τις επόμενες γενιές διαδραστικών εφαρμογών πραγματικού χρόνου που θα επωφελούνται τόσο από την ανεξαρτησία του χώρου όσο και από την ελαστικότητα που τυπικά έχουν όλες οι SOI εφαρμογές καθώς και από την προβλεπόμενη χρονική συμπεριφορά(διαδραστικότητα και χρόνος απόκρισης, κέρδος) που τυπικά παρουσιάζουν τα μαλακά συστήματα πραγματικού χρόνου.

Οι ιδιότητες πραγματικού χρόνου στα SOI απαιτούν την εκτέλεση των υπηρεσιών σε περιορισμένα χρονικά πλαίσια, τόσο για να καλύψουν τις προθεσμίες εκτέλεσης των εργασιών που έχουν τεθεί από τους χρήστες όσο και για την διατήρηση της αναμενόμενης απόδοσης. Συγκεκριμένα , αυτές οι χρονικά κρίσιμες διαδικασίες περιλαμβάνουν σύγχρονες επικοινωνίες μεταξύ των υπηρεσιών , κάτι που έχει άμεσο αντίκτυπο στην μελέτη ,στην υλοποίηση και την ενσωμάτωση των Υποδομών προσανατολισμένες στις Υπηρεσίες.

Το πιο δύσκολο κομμάτι είναι η ενσωμάτωση των ιδιοτήτων πραγματικού χρόνου σε όλα τα επίπεδα των συστημάτων προσανατολισμένα στην Υπηρεσία. Η προσέγγιση του Irmos Project είναι να εστιάσει στην ανάγκη για μεθοδολογίες , εργαλεία και αρχιτεκτονικές για περίπλοκα διανεμημένα συστήματα που αντιμετωπίζουν τα πρακτικά ζητήματα για την εγγύηση της απόδοσης της ορισμένης χρονικά εκτέλεσης , της διαχείρισης των πόρων σε πραγματικό χρόνο , της σύγχρονης επικοινωνίας κάτω από διάφορες συνθήκες φόρτου , την ικανοποίηση των περιορισμών της Ποιότητας της Υπηρεσίας(QoS) καθώς και την αντιμετώπιση των αμοιβαίων συγκρούσεων που προκύπτουν από αυτά τα ζητήματα.

Η Υποδομή του Irmos θα ενσωματώσει διάφορα αποτελέσματα ερευνών και τεχνολογιών χρησιμοποιώντας στο έπακρο τα πλεονεκτήματα των υπηρεσιών βασισμένες σε Grid, και των υπαρχόντων συστημάτων ρήτρας όπως τα IP Multimedia Subsystem (IMS).

5.2.1 Γιατί να χρησιμοποιήσουμε την λύση που παρέχει το Irmos;

Ο απώτερος στόχος των Υποδομών προσανατολισμένες στην Υπηρεσία (SOI) μπορεί να γίνει πραγματικότητα μόνο εάν η αξιοπιστία και η ευελιξία των διαθέσιμος βοηθητικών προγραμμάτων είναι συγκρίσιμης ποιότητας με τις εσωτερικές ρυθμίσεις. Καμία εταιρεία δεν θα αναθέσει σε εξωτερικές πηγές τμήμα της ροής της εργασίας της αν δεν πληρείται ένα συγκεκριμένο επίπεδο ποιότητας της εκτέλεσης ή αν η υποδομή δεν είναι σε θέση να αντιδράσει δυναμικά στις εναλλασσόμενες συνθήκες και να προσαρμοστεί σε αυτές ταχύτατα. Η βασική διαφοροποίηση του *Irmos* είναι η υπόθεση ότι η υλοποίηση μιας τέτοιας υποδομής είναι εφικτή μόνο επεκτείνοντας το παράδειγμα των SOA με δυνατότητες πραγματικού χρόνου. Η υπόθεση αυτή βασίζεται στο γεγονός ότι κάθε SOI, παρά τις πολλές εικονικοποιήσεις (virtualizations) , στην βάση του αποτελείται από διάφορα δίκτυα και ένα μεγάλο αριθμό υπολογιστικών συστημάτων και υπηρεσιών εφαρμογής που πρέπει να συντονίζονται και να καθοδηγούνται από τους χρονικούς περιορισμούς ώστε να ενεργοποιηθεί η ανάπτυξη προχωρημένων και εμπορικά βιώσιμων εφαρμογών .

Η μεγάλη καινοτομία που προσφέρεται από το έργο *Irmos* είναι το πλαίσιο που επιτρέπει την δημιουργία εφαρμογών πραγματικού χρόνου που επεκτείνονται εκτός των ορίων των οργανισμών. Οι επιλεγμένες εφαρμογές του *Irmos* εκτείνονται από εφαρμογές επαγγελματικής συνεργασίας υψηλού προφίλ μέχρι εφαρμογές που αφορούν πολλούς τελικούς χρήστες. Όλοι τους έχουν σημαντικές απαιτήσεις σχετικά με την επεξεργασία, την αποθήκευση και την δικτυακές ικανότητες.

Εισάγοντας τις δυνατότητες πραγματικού χρόνου σε ένα SOI μας επιτρέπει την χρονικά περιορισμένη εκτέλεση των υπηρεσιών , στοχεύοντας στην ικανοποίηση των προθεσμιών των εκτελούντων εργασιών όπως αυτές ορίστηκαν από τους χρήστες. Αυτές οι χρονικά περιορισμένες εργασίες περιέχουν εκτός άλλων μια σύγχρονη λειτουργία επικοινωνίας μεταξύ των υπηρεσιών και έχουν άμεσο αντίκτυπο στο σχεδιασμό , την υλοποίηση και την ενσωμάτωση των υπηρεσιών στις υποδομές αυτές.

Το να καταστήσουμε το SOA μια επιχειρηματική πρόταση είναι μεγάλη πρόκληση και ένα βασικό συστατικό για την επέκταση της βάσης των χρηστών , επεκτείνοντας τα SOA συστήματα (απλά ασύγχρονα , δεν καθορίζονται από την απόδοση, υποδομή επικοινωνίας βασισμένη στο γεγονός-event based messaging infrastructure) όπως πραγματοποιούνται

από το SOAP(Simple Object Access Protocol) και τις Υπηρεσίες Δικτύου(Web Services) σε μια πιο ολοκληρωμένη υποδομή με ενσωματωμένες δυνατότητες πραγματικού χρόνου και σύγχρονης επικοινωνίας.

Σε μια τέτοια προσέγγιση εστιάζουμε κυρίως στην αρχιτεκτονική τέτοιων πολύπλοκών διανεμημένων συστημάτων τα οποία ταυτόχρονα αντιμετωπίζουν πρακτικά ζητήματα για την διασφάλιση της απόδοσης του συστήματος (χρονομετρημένη εκτέλεση διαχείρισης πόρων, συγχρονισμένη επικοινωνία υπό διάφορες συνθήκες φόρτισης, ικανοποίηση των περιορισμών QoS), καθώς και την αντιμετώπιση των αμοιβαίων συγκρούσεων που προκύπτουν από αυτά τα ζητήματα.

Η Υποδομή του Irmos θα ενσωματώσει διάφορα αποτελέσματα ερευνών και τεχνολογιών χρησιμοποιώντας στο έπακρο τα πλεονεκτήματα των υπηρεσιών βασισμένες σε Grid, και των υπαρχόντων συστημάτων Multimedia provisioning όπως το IMS και των υπαρχόντων και εξελισσόμενων στάνταρτ γύρω από το παράδειγμα των SOA.

5.2.2 Γιατί προσανατολισμός στην Υπηρεσία ;

Μια *Αρχιτεκτονική προσανατολισμένη στις Υπηρεσίες(SOA)* είναι μια συλλογή υπηρεσιών που στοχεύουν να επικοινωνούν μεταξύ τους. Η επικοινωνίας αυτή μπορεί να περιέχει είτε ανταλλαγή δεδομένων είτε να εμπεριέχει περισσότερες υπηρεσίες που διαχειρίζονται μια συγκεκριμένη δραστηριότητα. Στο πλαίσιο αυτό, μια υπηρεσία είναι μια λειτουργία που είναι καλώς ορισμένη, αυτό-περιεχόμενη , και δεν εξαρτάται από το πλαίσιο ή την κατάσταση στην οποία βρίσκονται άλλες υπηρεσίες.

Το *SOA* προσφέρει πολλά πλεονεκτήματα έναντι των παραδοσιακών διανεμημένων συστημάτων και για τον λόγο αυτό τείνουν να αντικαθιστούν τις πλατφόρμες πάνω στις οποίες οι επιχειρηματικές υπηρεσίες προσφέρονται στους πελάτες. Παρέχουν γεωγραφική ανεξαρτησία των υπηρεσιών , κάτι που σημαίνει ότι οι υπηρεσίες αυτές δεν χρειάζεται να βρίσκονται σε συγκεκριμένο σύστημα ή σε συγκεκριμένο δίκτυο. Η αναζήτηση και σύνδεση σε κάθε υπηρεσία μπορεί να είναι δυναμική και να ακολουθεί μια χαλαρά συζευγμένη προσέγγιση επιτρέποντας τον σχηματισμό υποδομών παροχής υπηρεσιών γενικής χρήσης που θα λειτουργούν για μεγάλο σύνολο τεχνολογιών και επιχειρηματικών διαδικασιών.

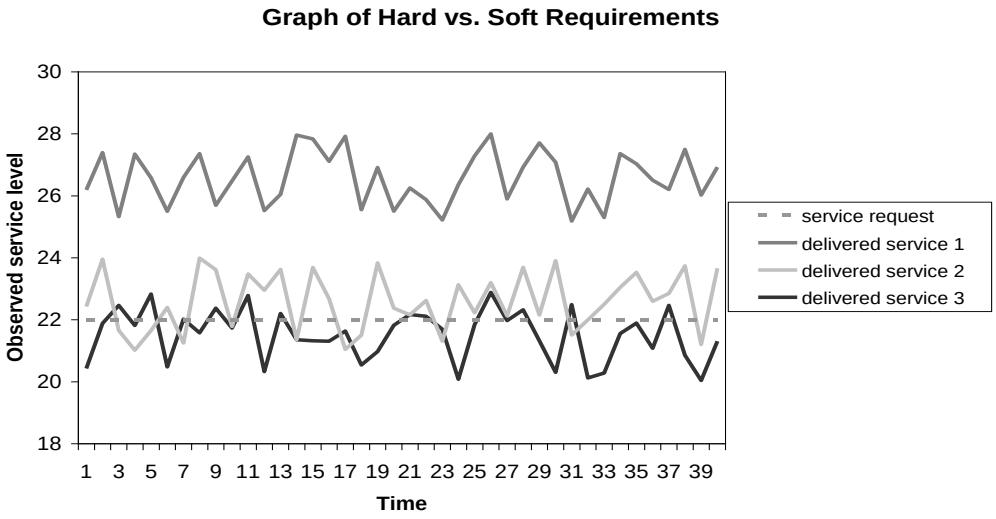
Άλλα πλεονεκτήματα της SOAs είναι η ικανότητά τους για να δημιουργούν σύνθετες εφαρμογές μειώνοντας ταυτόχρονα το κόστος διαχείρισης και το κόστος που συνδέεται με την απόκτηση και την συντήρηση της τεχνολογίας και επιτρέπουν την δημιουργία υποδομών που είναι επεκτάσιμες και των οποίων η απόδοση είναι σύμφωνη με τις επιχειρηματικές απαιτήσεις.

5.2.3 Γιατί πραγματικός χρόνος ;

Ένα σύστημα *πραγματικού χρόνου* είναι ένα υπολογιστικό σύστημα του οποίου η ορθή συμπεριφορά εξαρτάται τόσο από το αποτέλεσμα της εξόδου του όσο και από τον χρόνο που απαιτείται για να υπολογίσουν το αποτέλεσμα αυτό. Παραδοσιακά ο όρος αυτός αναφέρεται στα *«σκληρά»* συστήματα πραγματικού χρόνου , για τα οποία ακόμα και μία παραβίαση της προσχεδιασμένης χρονικής συμπεριφοράς δεν είναι αποδεκτή , επειδή μπορεί να οδηγήσει σε ολική κατάρρευση του συστήματος , κοστίζοντας πιθανώς και ανθρώπινες ζωές. Ωστόσο μεγάλο εύρος εφαρμογών με χρονικές απαιτήσεις αποκτούν ορμή τον τελευταίο καιρό , για τις οποίες μερικές παραβιάσεις των προσχεδιασμένων χρονικών απαιτήσεων μπορεί να είναι αποδεκτές , δεδομένου ότι η συχνότητα τους και η σπουδαιότητα τους διατηρούνται εντός ελέγχου. Αυτά είναι τα «μαλακά» συστήματα πραγματικού χρόνου , και παραδοσιακά αναπτύσσονται χωρίς την χρήση μεθοδολογίας πραγματικού χρόνου ή υποστήριξης χρόνου εκτέλεσης από το λειτουργικό σύστημα. Το αποτέλεσμα είναι οι εφαρμογές αυτές να εκτελούνται σε ακριβά και εξειδικευμένα συστήματα με καλά επίπεδα αποκρισιμότητας και απόδοσης ή να εκτελούνται σε υλικό και λειτουργικά συστήματα γενικής χρήσης χωρίς τα παραπάνω επίπεδα αποκρισιμότητας και απόδοσης , λόγω της αδυναμίας ελέγχου όλων των παρεμβολών με τις άλλες εφαρμογές και υπηρεσίες που εκτελούνται στους ίδιους κόμβους.

Το έργο Irmos θα εφαρμόσει τις αναδυόμενες μεθοδολογίες πραγματικού χρόνου και τεχνολογίες που είναι κατάλληλες για συστήματα πραγματικού χρόνου μέσα στο πλαίσιο των SOI συστημάτων , με σκοπό την κατασκευή των θεμελιωδών δομών για τις εφαρμογές επόμενης γενιάς που θα επωφελούνται τόσο από την γεωγραφική ανεξαρτησία και την ευρωστία που χαρακτηρίζει τυπικά τα SOI , όσο και από την προβλεψιμότητα στην

χρονική συμπεριφορά (αλληλεπίδραση και χρόνος απόκρισης , απόδοση) που χαρακτηρίζει τυπικά τα *«μαλακά»* συστήματα πραγματικού χρόνου.



Εικόνα 10: «*Σκληρές*» και «*Μαλακές*» *προθεσμίες*

Η εικόνα 10 δείχνει τις διαφορές μεταξύ των σκληρών και των μαλακών περιορισμών πραγματικού χρόνου. Στην παρουσία της ζητούμενης υπηρεσίας , η παρεχόμενη υπηρεσία 1 μπορεί να θεωρηθεί ότι συναντά ένα σκληρό χρονικό περιορισμό αφού είναι η μόνη που παρέχει περισσότερα από όσα της ζητήθηκαν. Η παρεχόμενη υπηρεσία 1 και η παρεχόμενη υπηρεσία 2 συναντάν έναν μαλακό χρονικό περιορισμό καθώς , κατά μέσο όρο , προσφέρουν καλύτερη υπηρεσία από ότι ζητήθηκε. Η παρεχόμενη υπηρεσία 3 αποτυγχάνει τόσο στους σκληρούς όσο και στους μαλακούς χρονικούς περιορισμούς , καθώς δεν προσφέρει υπηρεσία κατά μέσο όρο πάνω το ζητούμενο ακόμα και αν υπάρχουν χρονικές στιγμές στις οποίες προσφέρει επίπεδα υπηρεσίας πάνω από το ζητούμενο , ακόμα και μεγαλύτερα επίπεδα από την παρεχόμενη υπηρεσία 2 σε κάποιες άλλες στιγμές.

5.2.4 Πραγματικός Χρόνος: Μια διεπιστημονική προσέγγιση σχεδίασης

Τα συστήματα πραγματικού χρόνου είναι πολύπλοκα και η σχεδίαση και ανάλυση τους εμπεριέχει πολλές από τις αρχές της επιστήμης των υπολογιστών και του ηλεκτρολόγου μηχανικού (όπως: γλώσσες προγραμματισμού , μηχανική λογισμικού,

αρχιτεκτονική υπολογιστών , λειτουργικά συστήματα , θεωρία αναμονής και χρονοδιαγράμματος , αλγορίθμους κ ι πολυπλοκότητα κ.α.) .

Η σχεδίαση και υλοποίηση των συστημάτων πραγματικού χρόνου απαιτεί να δοθεί προσοχή σε ένα μεγάλο σύνολο διαφορετικών θεμάτων όπως:

- Η επιλογή του λογισμικού και του υλικού , η αξιολόγηση των αμοιβαίων συγκρουόμενων συμφερόντων που εμφανίζονται προσπαθώντας να ικανοποιήσουμε τόσο την απόδοση όσο και την ποιότητα των προσφερόμενων υπηρεσιών , και εμπεριέχει επίσης θέματα παραλληλισμού και συγχρονισμού.
- Οι επιπλοκές στην διαδικασία παραγωγής του λογισμικού που εμφανίζονται λόγω του πραγματικού χρόνου(από τις γλώσσες προγραμματισμού έως τον κώδικα μηχανής)
- Ο εξαντλητικός έλεγχος όλων των στοιχείων που αποτελούν το έργο με δοκιμές πριν την τελική τους εγκατάσταση σε κανονική λειτουργία (κυρίως μετρώντας και προβλέποντας τους χρόνους απόκρισης και μειώνοντας τους όσο είναι δυνατόν).
- Μεγιστοποίηση της ανοχής σφαλμάτων του συστήματος και της αξιοπιστίας μέσω προσεκτικού σχεδιασμού.

Είναι προφανές ότι οι μηχανικές τεχνικές που χρησιμοποιούνται στα συστήματα πραγματικού χρόνου μπορούν να χρησιμοποιηθούν στην μηχανική όλων των άλλων τύπων συστημάτων , όπως το SOI με αποτέλεσμα την καλύτερη απόδοση και ευστάθεια.

5.3 Κίνητρο πίσω από τις επιλεγμένες εφαρμογές

Η κοινοπραξία του *Irmos* ερεύνησε εκτεταμένα για την επιλογή των υποψήφιων εφαρμογών για την τελική ανάπτυξη της πλατφόρμας και την αξιολόγηση της καθώς και την αξιολόγηση του τελικού προϊόντος του έργου Irmos.Οι εφαρμογές αυτές παρουσιάζουν ένα σημαντικό μέρος των ιδιοτήτων και των απαιτήσεων που τέθηκαν υπόψη για την επίδειξη της προσέγγισης του *Irmos*.

Πρώτα μια σύνοψη των χρονικών απαιτήσεων και των προβλεπόμενων ιδιοτήτων που πρέπει να εμπεριέχονται στην σχεδίαση και υλοποίηση της τελικής υποδομής του Irmos παρουσιάζονται στους παρακάτω πίνακες.

	Real-Time capability that is required			
Application candidates	*A/V Streaming*	*Computing or transcoding*	*Desktop or Application Sharing*	*Interaction through Virtual Reality*
Advanced Virtual and Augmented Reality	X	X	X	X
Learning and Venue Visiting	X		X	X
Digital Film Postproduction	X	X	X	

Πίνακας 1: Απεικόνιση των εφαρμογών του Irmos και των χρονικά κρίσιμων δυνατοτήτων που αυτές απαιτούν.

Real-Time capability	**Min computation power (wrt P4 Class)**	**Min Throughput**	**Max Latency**	**Jitter Tolerance**
Streaming	Low-Average	Palm: 5Kb/s DVD: 3-10 Mb/s HDTV: 15Mb/s	1-way: 500ms-2s 2-way:	Audio: low Video: higher
Computing or transcoding	High (cluster computing)			
Desktop or	Low (text,	~Kbps to	7	Average –

Application Sharing	board) Higher (images)	~Mbps	5-500ms	High
Virtual & Augmented Reality	Medium – High (depending on the data set)	10kbps – 10gbps (depending on the data set and current action)	~ 100 – 200ms for interactions and frame latency	For a short time high, but overall low jitter is required

Πινάκας 2 Οι χρονικές απαιτήσεις για την καλύτερη εμπειρία του χρήστη στις αναγνωρισμένες χρονικά σημαντικές δυνατότητες.

Οι χρονικές απαιτήσεις των παραπάνω εφαρμογών είναι πιθανόν υπό κανονικές συνθήκες να μπορούσαν να εκτελεστούν σε συστήματα και δίκτυα γενικής χρήσης. Η εκτέλεση τους σε μια πλατφόρμα που μπορεί να εφαρμόσει έλεγχο QoS και κατά συνέπεια να εξασφαλίσει την εκτέλεση τους σε «*μαλακό*» πραγματικό χρόνο θα δείξει ότι η συνολική εμπειρία βελτιώνεται αισθητά. (Ο *μαλακός* πραγματικός χρόνος είναι αρκετός αφού μια μικρή προσωρινή μείωση της υπηρεσίας προκαλεί στην χρήστη μια μικρή ενόχληση που δεν έχει μακροπρόθεσμα αποτελέσματα).

6 Ανάλυση και Σχεδίαση του Συστήματος - Προσδιορισμός απαιτήσεων

Στην ενότητα αυτή θα περιγράψουμε την εφαρμογή Γραφικού περιβάλλοντος διεπαφής χρήστη για το Irmos Project. Θα περιγράψουμε τον τρόπο υλοποίησης της καθώς και τα εργαλεία που χρησιμοποιήθηκαν κατά την ανάπτυξη της . Επίσης θα γίνει περιγραφη των προβλημάτων που παρουσιάστηκαν κατά την φάση της υλοποίησης και οι τρόποι με τους οποίους λύθηκαν.Τέλος θα γίνει παρουσίαση της εφαρμογής με χρήση εικόνων, και του πηγαίου κώδικα.

6.1 Λειτουργικές και μη λειτουργικές Απαιτήσεις της εφαρμογής

Οι λειτουργικές απαίτήσεις της εφαρμογής είναι η εκτέλεση των βασικών ρυθμίσεων του Irmos Project , η τελική εκτέλεση και ο έλεγχος της διαδικασίας εκτέλεσης σε πραγματικό χρόνο όλα μέσα από ένα γραφικό περιβάλλον εργασίας.Η ακριβής διαδικασία που πρέπει να ακολουθεί το γραφικό περιβάλλον είναι αρκετά σαφής , καλώς ορισμένη και σχεδόν μονοδρομική κατί που θα γίνει πιο σαφές στην επόμενη παράγραφο.

Αναλυτικότερα οι απαιτήσεις που πρέπει να πληρούνται από την εφαρμογή χωρίζονται σε 4 διακριτές φάσεις. Την φαση της Διαπραγματευσής με το σύστημα (Negotiation Phase) , την φάση της Κράτησης (Reservation Phase) , την φάση της Εκτέλεσης (Execution Phase) και τέλος τη φάση του ελέγχου (Monitoring Phase). Αναλυτικότερα η ακριβής διαδικασία που πρέπει να ακολουθηθεί φαίνεται στο παρακάτω ακολουθιακό διάγραμμα. Στο διάγραμμα αυτό το γραφικό περιβάλλον διασύνδεσης παίζει τον ρόλο του «*WP4 CLIENT*». Όπως είναι εμφανές απο το διάγραμμα η μόνη ανταλλάγη μυνημάτων μεταξύ του γραφικού περιβάλλοντος και του υπόλοιπου συστήματος είναι μέσω της Web Service με όνομα IrmosPortal. Η υπηρεσία αυτή παίζει τον ρόλο του ενδιάμεσου μεταξυ του τελικού χρήστη (που χρησιμοποιεί το γραφικό περιβάλλον) και του υπόλοιπου

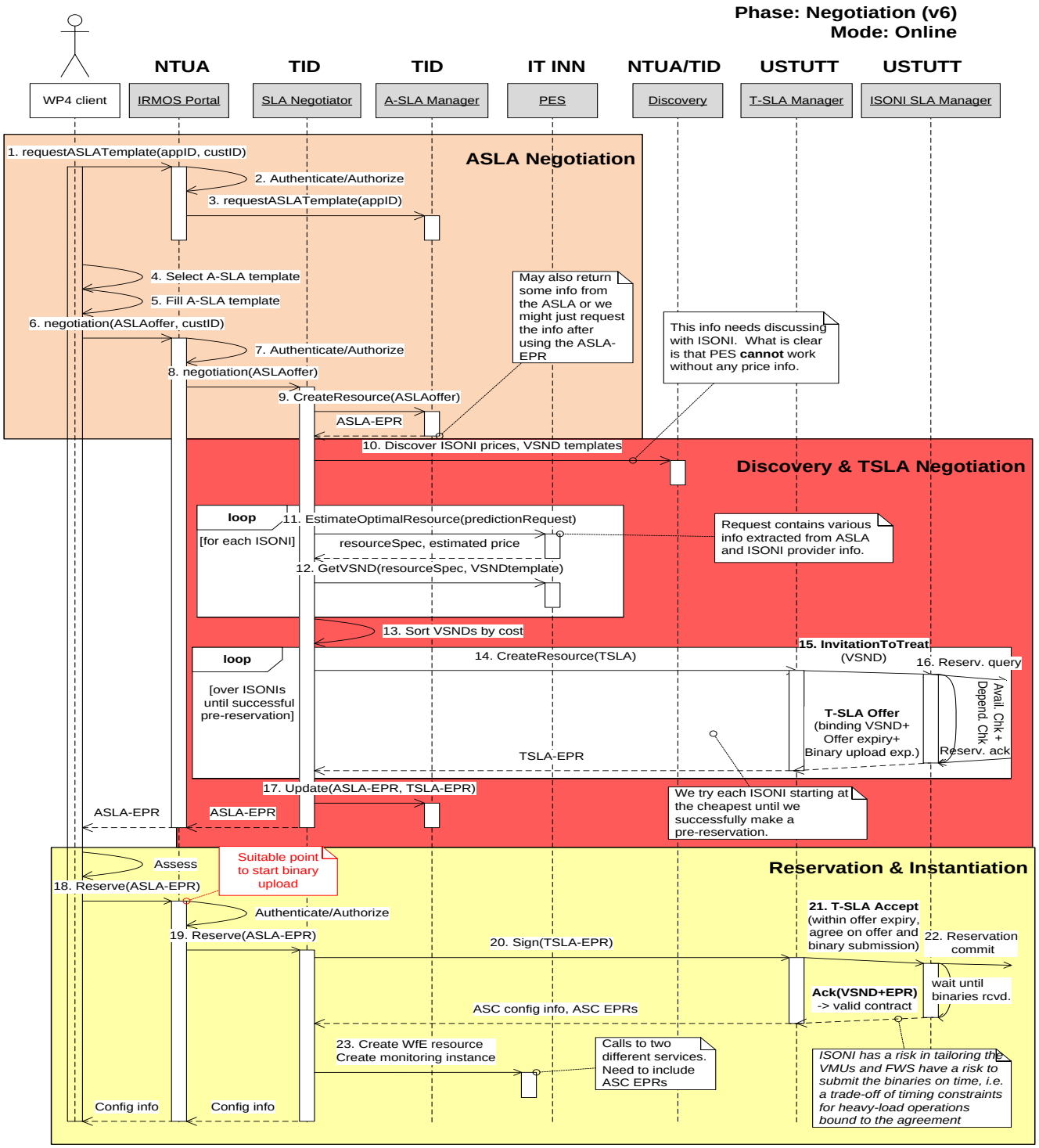
συστήματος. Επίσης είναι υπευθυνη για την μεταφόρα από και προς το γραφικό περιβάλλον όλων των δεδομένων που απαιτούνται. Έτσι επιτυγχάνεται η απομώνοση του τελικού χρήστη από το υπόλοιπο σύστημα και απόκρύπτονται οι λεπτομέρειες της υλοποίησης καθώς και όλων των διαδικάσιων που εκτελούνται από τις ενέργειες του χρήστη(κάτι που ούτως ή άλλως δεν πρέπει να τον ενδιαφέρει).Έτσι επιτυγχάνεται η απλότητα στην χρήση-μια από τις μη λειτουργικές απαίτησης του συστήματος.

Κατά την φάση της *Διαπραγμάτευσης* ,όπως την βλέπουμε και στο παρακάτω διάγραμμα (**Εικ. 6.1**) μπορεί να χωριστεί σε δύο υπο-φάσεις .Η υπο-φάση του Negotiation που διαρκεί από τα βήματα 1 εώς 17 από τα οποία τα βήματα 9 εως και 17 συμβαίνουν εσωτερικά του συστήματος χωρίς την παρέμβαση του χρήστη και δεν είναι ορατά από αυτόν. Κατά την φάση αυτή ο χρήστης-πελάτης θα πρέπει να συμπληρώνει τα στοιχεία του , συγκεκριμένα ένα Customer id και ένα Application id , που πρέπει να είναι γνωστά εκ των προτέρων. Με βάση τα στοιχεία αυτά που προωθούνται περαιτερω στο σύστημα πρέπει να είναι σε θέση να ολοκληρώσει την *Διαπραγματευση* με το σύστημα. Πιο αναλυτικά η διαδικασία θα αναλυθεί στην επομενη παράγραφο.

Η υπο-φάση του Reservation διαρκεί από τα βήματα 18 εώς 23 , στα οποία η παρέμβαση του χρήστη είναι απαραίτητη σε δύο μόνο σημεία , στα βήματα 18 και κατά το τελικό βήμα. Απαραίτητη προυπόθεση για την υπο-φάση του Reservation είναι η σωστή και πλήρης ολοκλήρωση της φάσης του Negotiation και για τον λόγο αυτό δεν μπορεί να διαχωριστεί από αυτήν και θεωρείται υποπερίπτωση της. Στην φάση αυτή όλες οι πληροφορίες για τον πελάτη είναι γνωστές στο σύστημα και αυτός το μόνο που πρέπει να κάνει είναι να δηλώσει την επιθυμία του για την ολοκλήρωση της φάσης (παραδείγματος χάριν με το πάτημα ενός κουμπιού). Επίσης θα πρέπει να είναι σε θέση να «κατεβάσει» στον υπολογιστή του τις πληροφορίες παραμετροποίησης της διαδικασίας που είναι γνωστές στο σύστημα από την ολοκλήρωση της φάση του Negotiation(και πάλι με το πάτημα ενός κουμπιού).

Η φάση του Ελέγχου (Monitoring) είναι πολύ πιο απλή από την προηγούμενη φάση. Ο χρήστης θα πρέπει να μπορεί με το πάτημα μιας επιλογής να βλέπει στην οθόνη του όλες τις διαθέσιμες πληροφορίες και δεδομένα που προκύπτουν από την εκτέλεση της εφαρμογής του στο σύστημα. Φυσικά οι πληροφορίες αυτές θα πρέπει να ανανεώνονται σε τακτά και σύντομα χρονικά διαστήματα για να εμφανίζονται στην οθόνη οι πιο πρόσφατες

κάθε φορά. Επίσης για την διευκόλυνση του χρήστη θα πρέπει οι τιμές των δεδομένων που εμφανίζονται στην οθόνη να συγκρίνονται κάθε στιγμή με τις ανεμενόμενες τιμές που πρέπει να έχουν και όσες δεν συμφωνούν να γίνονται ευδιάκριτες (παραδείγματος χάριν αλλάζοντας χρώμα σε κόκκινο) . *Αυτό επίσης κάνει εμφανή την εξασφαλιση της* ***ποιότητας της υπηρεσίας*** *(**QOS**) που προσφέρει το Irmos Project.*



Εικόνα 6.1.1 Ακολοθιακό διάγραμμα Negotiation – Reservation

Τέλος η φάση της Εκτέλεσης είναι η πιο απλή από όλες και προυποθέτει την ολοκλήρωση των φάσεων Negotiation και Reservation. Ο χρήστης θα πρέπει να μπορεί με το πάτημα ενός κουμπιού να δρομολογεί στο σύστημα την εκκίνηση της εφαρμογής του και να επιβεβαιώνει με απάντηση από αυτό ότι δρομολογήθηκε σωστά.

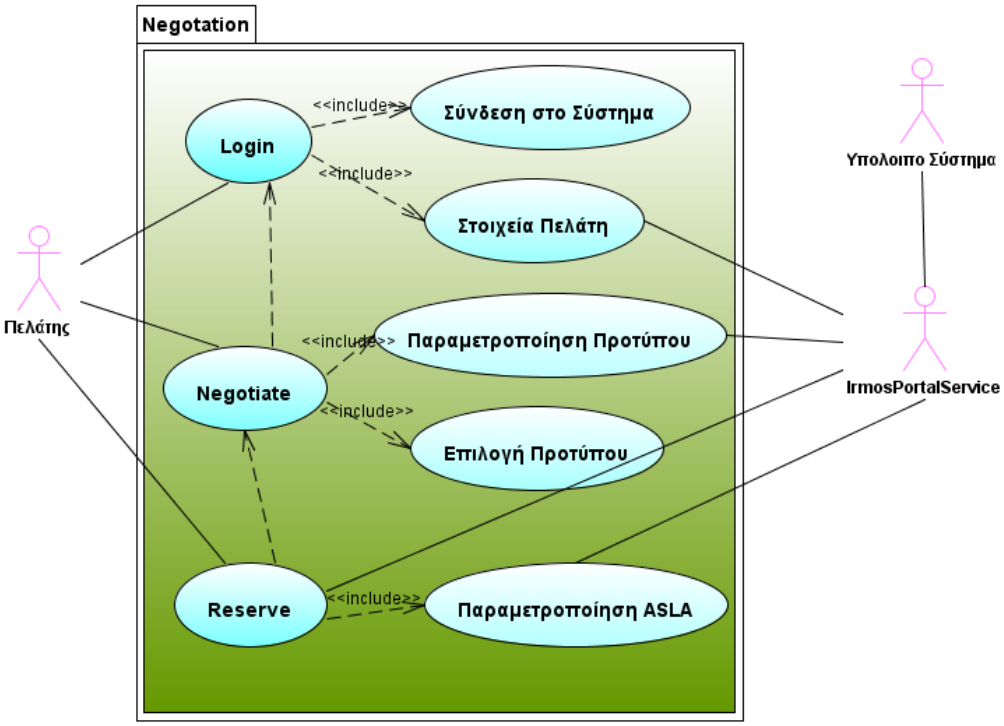
6.2 Περιορισμοί

- Όλοι οι χρήστες του συστήματος πρέπει να διαθέτουν Web browser
- Όλοι οι χρήστες του συστήματος θα πρέπει να συνδέονται με χρήση username και password
- Τα δεδομένα του χρήστη διατηρούνται στο σύστημα για όσο διαρκεί η συνεδρία(session).Αυτό ορίζεται από το server που βρίσκεται εγκατεστημένη η εφαρμογή.

6.3 Μοντέλο Περιπτώσεων Χρήσης

6.3.1 Περίπτωση Χρήσης 1 Negotiation – Reservation

Στα παρακάτω διαγράμματα φαίνονται αναλυτικά οι περιπτώσεις χρήσεις που προκύπτουν από την φάση του Negotiation και του Reservation.Επίσης παρουσιάζονται και τα ακολουθιακά διαγράμματα των περιπτώσεων χρήσης και αναλύεται η βασική και η εναλλακτική ροή αυτών. Όπως είναι εμφανές η φάση της Διαπραγματευσης αποτελείται από αρκετές περιπτώσεις χρήσης οι οποίες όμως αλληλοεξαρτόνται μεταξύ τους σε μεγάλο βαθμό και δεν μπορούν να υπάρξουν ανεξάρτητα. Για αυτό το λόγο η ροή όλων αυτών των περιπτώσεων χρήσης είναι μία.



Εικόνα 6.2 Περίπτωση Χρήσης 1 Negotiation-Reservation

Βασική Ροή Negotiation

1. Ο χρήστης συνδέεται στο σύστημα χρησιμοποιώντας Username και Password.
2. Στην οθόνη του χρήστη εμφανίζεται το κεντρικό μενού με όλες τις επιλογές της εφαρμογής
3. Ο χρήστης επιλέγει Negotiation
4. Εμφανίζεται η αρχική σελίδα του Negotiation που ζητά από το χρήστη να εισάγει το Customer id του και το κουμπί Proceed
5. Ο χρήστης εισάγει το Customer id , πατάει Proceed.Εμφανίζεται πεδίο που ζητά από το χρήστη να εισάγει το Application id και το πλήκτρο RequestSLA το όποιο και ο χρήστης πατάει.
6. Α) Αν υπάρχουν SLA πρότυπα με βάση τα στοιχεία του χρήστη παραλαμβάνονται από το σύστημα και εμφανίζονται στο χρήστη με μορφή dropdown Box
7. Α) Ο χρήστης επιλέγει το πρότυπο της προτίμησης του και πατάει το πλήκτρο EditSla

8. Α) Στην οθόνη του χρήστη εμφανίζονται όλες οι παράμετροι που ορίζει το πρότυπο και οι προεπιλέγμένες τίμες τους.Ο χρήστης ρυθμίζει όποιες παραμέτρους επιθυμεί και πατάει το πλήκτρο Submit για να συνεχίσει στην φάση του Reservation

Εναλλακτική ροή Negotiation

Αν τα στοιχεία του χρήστη δεν είναι σωστά ή συμβεί κάποιο λάθος κατά το χρόνο εκτέλεσης της εφαρμογής η ροή αλλάζει.

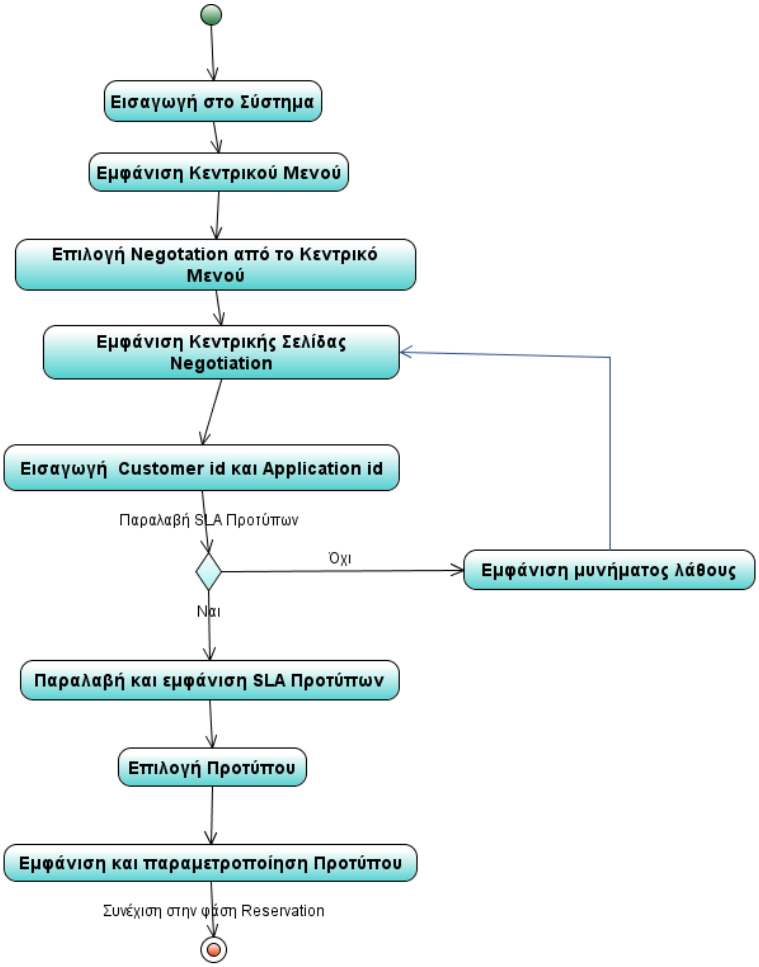
6. Β) Στην οθόνη του χρήστη εμφανίζεται η σελίδα που τον πληροφορεί αναλυτικά για το σφάλμα που συνέβη. Ο χρήστης πατάει το πλήκτρο Return To Negotiation
7. Β) Η ροή μεταφέρεται και πάλι στο βήμα 4.

Βασική Ροή Reservation

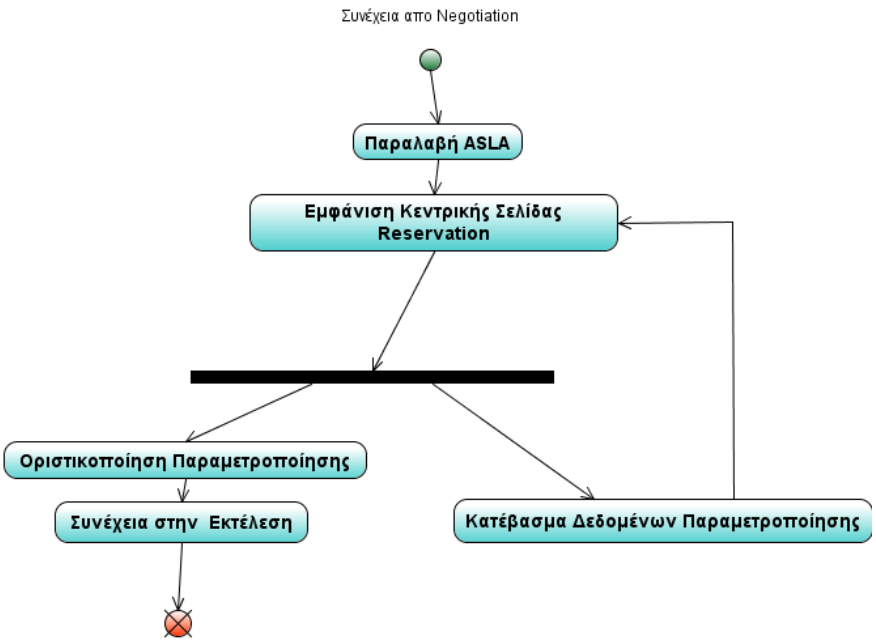
1. Τα στοιχεία του χρήστη υπάρχουν ήδη αποθηκευμένα στις εσωτερικες δομές του συστήματος. Βάση αυτών παραλαμβάνονται τα ASLA από το υπόλοιπο σύστημα.Εμφανίζονται στο χρήστη οι ρυθμίσεις και οι τιμές των παραμέτρων που έχουν οριστεί μέσω του ASLA(Config Info).Ο χρήστης έχει την δυνατότητα να επιλέξει να κατεβάσει τον υπολογιστή του τις πληροφόρίες αυτές ή να τις αποδεχτεί και να ολοκληρώσει την παραμετροποίηση πατώντας το πλήκτρο Configure.
2. Α) Ο χρήστης έχει επιλέξει την οριστικοποίηση της παραμετροποίησης. Οι πληροφορίες προωθούνται στο σύστημα και όταν είναι ετοίμο εμφανίζεται στον χρήστη η επιλογή για εκκίνηση της εκτέλεσης της εφαρμογής του.
3. Α) Ο χρήστης επιλέγει την εκκίνηση της εφαρμογής του πατώντας το πλήκτρο Execution.

Εναλλακτική ροή Reservation

2. Β) Ο χρήστης επιλέγει το κατέβασμα των ρυθμίσεων στον υπολογιστή του.
3. Β) Εμφανίζεται στο χρήστη ένα παράθυρο αποθηκευσης αρχείου , και επιστρέφει στην ίδια σελίδα.
4. Β) η ροή συνεχίζεται από το βήμα 1.



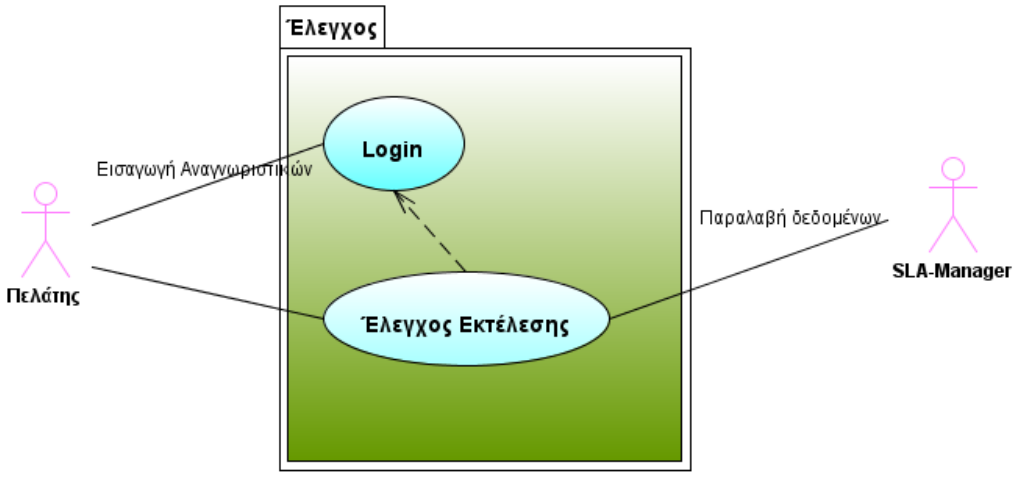
Εικόνα 6.3 Διάγραμμα Δραστηριότητας περιπτωσης χρήσης Negotiation



Εικόνα 6.4 Διάγραμμα Δραστηριότητας περιπτωσης χρήσης Reservation

6.3.2 Περιπτωση Χρήσης 2 Έλεγχος(Monitoring)

Η περιπτώσεις χρήσης για την φάση του Ελέγχου (Monitoring) είναι πιο απλές . Συγκεκριμένα ο χρήστης μπορεί να συνδεθεί στο σύστημα και βάση των στοιχείων του να ελέγξει τα δεδομένα που προκύπτουν από την εκτέλεση της εφαρμογής του στο σύστημα. Τα δεδομένα αυτά του παρουσιάζονται στην οθόνη μορφοποιημένα και με χρώματα βάση των τιμών τους όπως εξηγήθηκε και σε προηγούμενη παράγραφο(*βλ. Λειτουργικές και μη λειτουργικές απαιτήσεις της εφαρμογης)* . Τα δεδομένα αυτά προκύπτουν με κλήση στο υπόλοιπο σύστημα. Τα δεδομένα αυτά ανανεώνονται σε σύντομα χρονικά διαστήματα ώστε να παρουσιάζονται στο χρήστη κάθε στίγμη τα πιο πρόσφατα. Τέλος, παρακάτω παρουσιάζονται τα ακολουθιακά διαγράμματα της περίπτωσης χρήσης του Ελέγχου και επεξηγείται ή βασική ροή αυτής.



Εικόνα 6.5 Περίπτωση Χρήσης Έλεγχος (Monitoring)

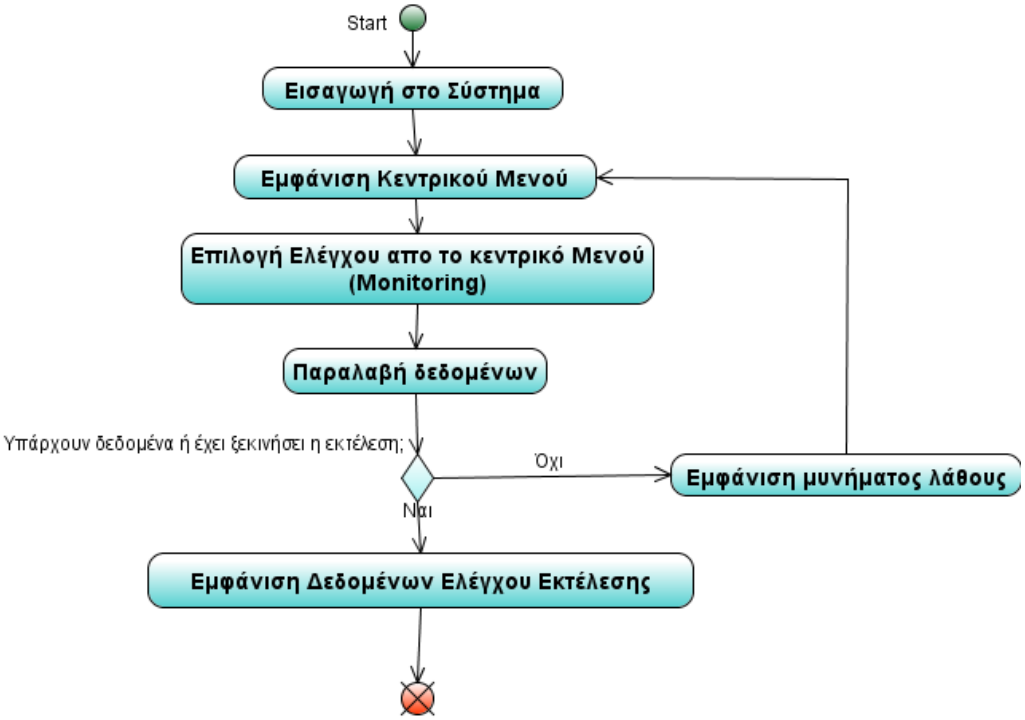
Βασική Ροή Monitoring

1. Ο χρήστης συνδέεται στο σύστημα χρησιμοποιώντας Username και Password.
2. Στην οθόνη του χρήστη εμφανίζεται το κεντρικό μενού με όλες τις επιλογές της εφαρμογής.
3. Ο χρήστης επιλέγει Monitoring.
4. Το σύστημα παραλαμβάνει τα πιο πρόσφατα δεδομένα
5. Α) Στην οθόνη του χρήστη εμφανίζονται μορφοποιημένα τα τρέχοντα δεδομένα και παράμετροι.
6. Α) Με το πέρας ορισμένου χρονικού διαστήματος η ροή επιστρέφει στο βήμα 4

***Εναλλακτική ροή* Monitoring**

Αν δεν υπάρχουν διαθέσιμα δεδομένα ή δεν έχει ξεκινήσει η εκτέλεση της εφαρμογής του χρήστη ή συμβεί κάποιο σφάλμα κατά το χρόνο εκτέλεσης της εφαρμογής η ροή αλλάζει.

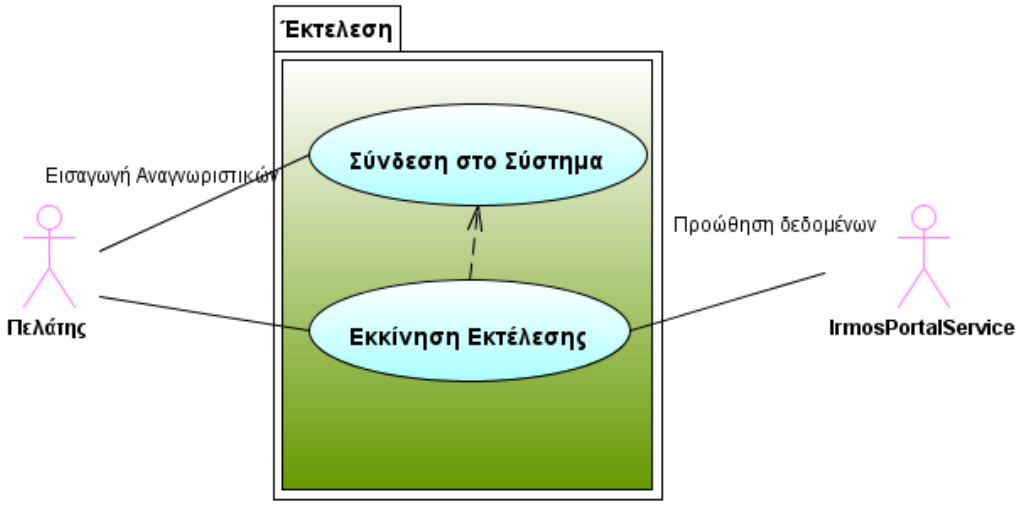
5. Β) Στην οθόνη του χρήστη εμφανίζεται η σελίδα που τον πληροφορεί αναλυτικά για το σφάλμα που συνέβη. Ο χρήστης πατάει το πλήκτρο Return To Index.
6. Β) Η ροή συνεχίζεται από το βήμα 2.



Εικόνα 6.6 Διάγραμμα Δραστηριότητας Περίπτωσης Χρήσης - Έλεγχος (Monitoring)

6.3.3 Περίπτωση Χρήσης 3 Εκτέλεση (Execution)

Κατά την περίπτωση χρήσης της εκτέλεσης ο χρήστης απλά επιλέγει την εκκίνηση της εφαρμογής του στο σύστημα. Το γραφικό περιβάλλον έχει ήδη όλες τις διαθέσιμες πληροφορίες για τον χρήστη-πελάτη και τις προωθεί στο υπόλοιπο σύστημα.Όπως και προηγουμένως , παρακάτω παρουσιάζονται το ακολουθιακό διάγραμμα της περιπτωσης χρήσης της εκτέλεσης και αναλύεται η βασική ροή αυτής.



Εικόνα 6.7 Περίπτωση Χρήσης Εκτέλεση (Execution)

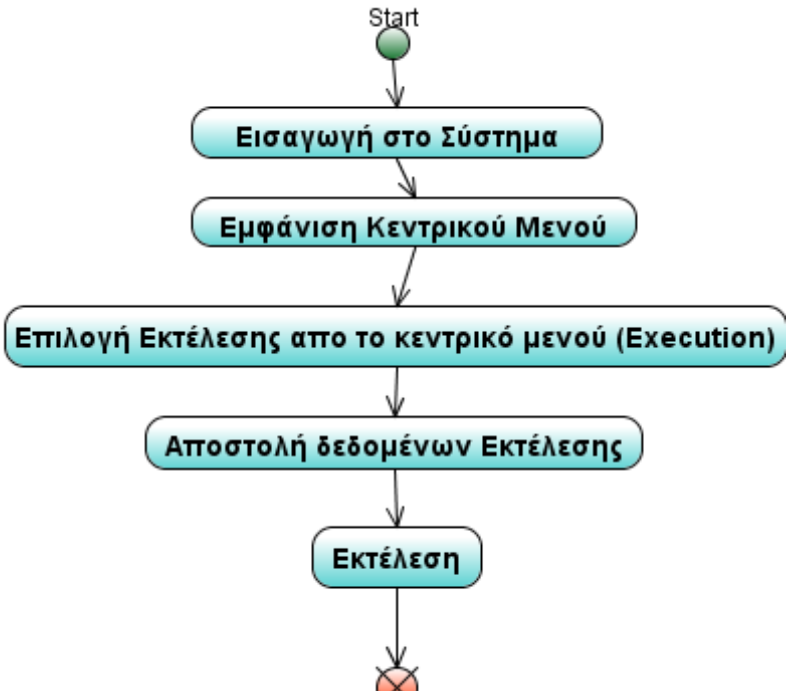
Βασική Ροή Execution

1. Ο χρήστης συνδέεται στο σύστημα χρησιμοποιώντας Username και Password.
2. Στην οθόνη του χρήστη εμφανίζεται το κεντρικό μενού με όλες τις επιλογές της εφαρμογής.
3. Ο χρήστης επιλέγει Execution.
4. Το σύστημα μεταφέρει τα δεδομένα και τις ρυθμίσεις του χρήστη στο υπόλοιπο σύστημα
5. Α)Η εκτέλεση της εφαρμογής ξεκινά
6. Β)Η ροή συνεχίζεται από το βήμα 2

Εναλλακτική ροή Execution

Αν συμβεί κάποιο σφάλμα κατά το χρόνο εκτέλεσης της εφαρμογής η ροή αλλάζει.

5. Β) Στην οθόνη του χρήστη εμφανίζεται η σελίδα που τον πληροφορεί αναλυτικά για το σφάλμα που συνέβη. Ο χρήστης πατάει το πλήκτρο Return To Index.
6. Β) Η ροή συνεχίζεται από το βήμα 2.



Εικόνα 6.8 Διάγραμμα Δραστηριότητας Περιπτωσης Χρήσης Εκτέλεσης (Execution)

6.4 Γιατί Java Server Pages

Στο σημείο αυτό θα πρέπει να δικαιολογήσουμε πιο αναλυτικά τους λόγους για τους οποίους επιλέχθηκε η τελική υλοποίηση του γραφικού περιβάλλοντος να γίνει με χρήση της τεχνολογίας Java Server Pages έναντι κάποιας άλλης εναλλακτικής υλοποίησης σε συμβατικές γλώσσες προγραμματισμού όπως C,C++ ή ακόμα και απλή εφαρμογή Java(με χρήση παραδείγματος χάριν της βιβλιοθήκης για γραφικά περιβάλλοντα Swing).

Οι σημαντικότεροι λόγοι για την επιλογή των Java Server Pages για την υλοποίηση του γραφικού περιβάλλοντος είναι:

- Η ασφάλεια
- Η φορητότητα
- Η ετερογένεια των συστημάτων που καλείται να εκτελεστεί
- Η γεωγραφική κατανομή των συστημάτων
- Είναι απλή και συνεργάζεται με την Java

Καταρχήν η ίδια η φύση του έργου Irmos και η χρήση των τεχνολογιών Grid είναι διανεμημένη και λειτουργεί σε συστήματα τα οποία είναι εντελώς ετερογενή μεταξύ τους με διαφορετικά λειτουργικά συστήματα και υπολογιστικούς πόρους και δυνατότητες. Οι χρήστες αυτοί θα πρέπει να είναι σε θέση να εκτελέσουν την ίδια εφαρμογή γραφικού περιβάλλοντος για να αλληλεπιδράσουν με το σύστημα. Το γεγονός αυτό λοιπόν μας οδηγεί άμεσα στο συμπέρασμα ότι μια συμβατική γλώσσα προγραμματισμού δεν ανταποκρίνεται στις προσδοκίες αυτές αφού απαιτείται μεταγλώττιση του προγράμματος για κάθε διαφορετικό σύστημα και λειτουργικό σύστημα ακόμα και για διαφορετικές εκδόσεις αυτού και υπόκειται σε όλα τα σφάλματα ασυμβατότητας που ενδέχεται να προκύψουν από τις διάφορες εκδόσεις βιβλιοθηκών , μεταγλωττιστών και λειτουργικών συστημάτων. Γίνεται αντιληπτό λοιπόν ότι οι «μεταβλητές» που πρέπει να ελεγχθουν για την ορθή λειτουργία της γραφικής εφαρμογής σε όλα τα ετερογενή συστήματα είναι πολλές. Από την άλλη με χρήση των Java Server Pages όλοκληρη η εφαρμογή μεταγλωτίζεται σε ένα σύστημα με Web Server για συγκεκριμένο λειτουργικό και εκδόσεις βιβλιοθηκών ενώ εκτελείται και στο ίδιο σύστημα και ο χρήστης βλέπει το τελικό παραγώμενο αποτέλεσμα αρκεί να έχει συμβατο browser.

Επιπλέον τα συστήματα στα οποία καλείτε να εκτελεστεί η γραφική εφαρμογή είναι γεωγραφικά κατανεμημένα σε μεγάλες αποστάσεις μεταξύ τους . Η χρήση των Jsp και ενός απλού Web Server μας επιτρέπει να υπάρχει ένα κεντρικό σύστημα στο οποίο όλοι οι χρήστες του συστήματος μπορούν να συδεθούν και να το διαχειριστούν. Το σύστημα αυτό μπορεί και να αποθηκευει ρυθμίσεις και προτιμήσεις των χρηστών απομακρυσμένα ώστε να μην χρειάζεται να βρίσκονται μπροστά στο σύστημα με το οποίο περάσαν τις αρχικές ρυθμίσεις.

Οι σελίδες και τα προγράμματα που βασίζονται στην τεχνολογία Java Server Pages σύνεργάζονται άψογα με την γλώσσα Java και μπορούν να εκτελούν κώδικα τόσο μέσα στην ίδια την σελίδα όσο και με χρήση εξωτερικών βιβλιοθηκών και πακέτων υποστήριξης. Το γεγονός αυτό είναι πολύ σημαντικό γιατι απαιτεί προυπόθεση για την λειτουργία του συστήματος αφού θα πρέπει να μπορεί να καλέσει τόσο βιβλιοθήκες Java για την συνεργασία του γραφικού περιβάλλοντος με το Globus Toolkit και με τα υπόλοιπα μέρη του Irmos Project (κυρίως με το IrmosPortalService) .

Τέλος υπάρχει και το ζήτημα της ασφάλειας. Το κεντρικό αυτό σημείο εκτέλεσης της εφαρμογής μπορεί να υπόκειται σε αυστηρούς έλεγχους για την εξασφάλιση της ασφάλειας τους συστήματος τόσο με απλούς ελεγχους εισόδου με ονόμα χρήστη και κωδικό (Login Form) όσο και με την ασφάλεια του ίδιου του Web Server από επιθέσεις.

7 Τεχνολογίες

7.1 Java Server Pages (JSP)

Η Java είναι μία αντικειμενοστραφής γλώσσα που αποτελεί ένα ισχυρό εργαλείο για δομημένο δικτυακό προγραμματισμό. Για το λόγο αυτό χρησιμοποιήσαμε την έκδοση Java 1.6.0_14, και την τεχνολογία JSPs που παρέχεται μέσω της πλατφόρμας Java 2 Enterprise Edition (J2EE 1.4.2), την οποία χρησιμοποιήσαμε για την ανάπτυξη της εφαρμογής μας στον server. Τα JSPs είναι μια τεχνολογία της Java που επιτρέπει στους προγραμματιστές λογισμικού να δημιουργούν δυναμικά-παραγόμενους ιστοχώρους, με χρήση HTML, XML, ή άλλων τύπων εγγράφων, σε απάντηση ενός αιτήματος πελάτη του παγκόσμιου ιστού. Η τεχνολογία επιτρέπει στον κώδικα της Java και ορισμένες προκαθορισμένες ενέργειες να ενσωματωθούν σε στατικό περιεχόμενο. Η σύνταξη JSP προσθέτει επιπλέον ετικέτες παρόμοιες με εκείνες της XML, τις λεγόμενες JSP actions, που χρησιμοποιούνται για να επικαλεσθούν την ενσωματωμένη λειτουργία. Επιπλέον, η τεχνολογία αυτή επιτρέπει τη δημιουργία JSP βιβλιοθηκών ετικετών (tag libraries) που λειτουργούν ως επεκτάσεις στις ήδη υπάρχουσες ετικέτες HTML ή XML. Οι βιβλιοθήκες ετικετών παρέχουν έναν ανεξάρτητο από την πλατφόρμα ανάπτυξης τρόπο για την επέκταση των δυνατοτήτων ενός server. Στην συγκεκριμένη περίπτωση χρησιμοποιήσαμε τις Jsp Tags που αφορούν την ανάγνωση και επεξεργασία δεδομένων XML για την προβολή τους στην σελίδα κατά την φάση του Monitoring*(http://java.sun.com/jstl/xml).*

7.2 Apache Tomcat

Ως εξυπηρετητής του διαδικτύου στον οποίο εγκαταστάθηκε η web εφαρμογή μας, χρησιμοποιήθηκε ο Apache Tomcat, στην έκδοση 5.5. Η τελευταία έκδοση του tomcat κατά την συγγραφή της διπλωματικής εργασίας είναι η 6.0.20 , αλλά δεν χρησιμοποιήθηκε γιατί παρουσιάζει ασυμβατότητες και μη σωστή λειτουργία με το Globus Toolkit 4.2.1. Ο Tomcat είναι ένα servlet container ή servlet engine. Υποστηρίζει servlets και JSPs καθώς και

στατικές σελίδες. Αναπτύχθηκε από την Apache Software Foundation (www.apache.org) και διατίθεται δωρεάν στο διαδίκτυο.

7.3 Netbeans και Globus Toolkit 4.2.1

Ένα προηγμένο Ενσωματωμένο Περιβάλλον Ανάπτυξης (integrated development environment –IDE) είναι μια εφαρμογή λογισμικού που παρέχει περιεκτικές και πλήρης δυνατότητες στους προγραματιστές υπολογιστών για την ανάπτυξη λογισμικού. Ένα IDE τυπικά αποτελείται από :

- Έναν επεξεργαστή πηγαίου κώδικα
- Έναν μεταγλωτιστή
- Εργαλεία αυτοματοποίησης του « χτισίματος» της εφαρμογής
- Έναν από-σφαλματωτή (debugger)

Συνήθως στα πιο σύγχρονα περιβάλλοντα περιλαμβάνονται και επιπλέον εργαλεία όπως εργαλεία ελέγχου έκδοσης , εργαλεία διευκόλυνσης της κατασκευής γραφικών περιβάλλοντων (Visual Editor) καθώς επίσης και επεξεργαστή κλάσεων , αντικειμένων , διγραμμάτων κληρονομικότητας κ.α για την διευκόλυνση προγραμματισμού εφαρμογών Object Oriented.

Το Netbeans IDE είναι ένα προηγμένο Ενσωματωμένο Περιβάλλον Ανάπτυξης που λειτουργεί σε όλες τις πλατφόρμες και λειτουργικά συστήματα. Αποτελείται ένα IDE ανοικτού κώδικα και μια πλατφόρμα εφαρμογών που επιτρέπει στους προγραμματιστές να δημιουργήσουν γρήγορα εφαρμογές διαδικτύου , επιχειρήσεων , desktop και κινήτες χρησιμοποιώντας την πλατφόρμα της Java , Java FX, PHP, JavaScript, Ajax ,Ruby ,C++ κ.α Το Netbeans Project υποστηρίζεται ενεργά και δραστήρια από την κοινότητα που παρέχουν είτε τεκμηρίωση και εκπαιδευτικές ενότητες αλλά και μια μεγάλη συλλογή από plugins που επεκτείνουν την λειτουργία του περιβάλλοντος.

Όλες οι παραπάνω δυνατότητες και λειτουργίες υποδεικνύουν τους λόγους για τους οποίους το NetBeans IDE είναι ιδανικό για την ανάπτυξη της εν λόγω εφαρμογής.Επίσης παρέχει ενσωματωμένη υποστήριξη της τεχνολογίας JSP σε συνδυασμό με εργαλεία για την αυτοματοποίηση διαδικασιών όπως η μεταγλώτιση των σελιδών σε servlet καθώς και το deployment σε server Apache Tomcat.

Όμως απαραίτητη προυπόθεση για την λειτουργία της εφαρμογής είναι η συνεργασία του κώδικα με το Globus Toolkit . Σύμφωνα με την επίσημη τεκμηρίωση του Globus Toolkit [4] δεν παρέχεται υποστήριξη για την εκτέλεση κώδικα πελάτη του Globus Toolkit μέσα από γραφικά περιβάλλοντα ανάπτυξης εφαρμογών όπως το NetBeans IDE. Η εναλλακτική λύση η μεταγλώτιση και εκτέλεση του κώδικα από γραμμή εντολών. Η διανομή του Globus Toolkit παρέχει scripts που ρυθμίζουν τις παραμέτρους του συστήματος για να είναι σε θέση να μεταγλωτίσει και να εκτελέσει κώδικα Java που χρησιμοποιεί τις βιβλιοθήκες του Globus, συγκεκριμένα ορίζουν την καθολική μεταβλητή του συστήματος CLASSPATH. Η διαδικασία προυποθέτει ότι έχουμε ορίσει σωστά μια καθολική μεταβλητή του συστήματος με όνομα GLOBUS_LOCATION και τιμή ίση με το φάκελο που βρίσκετε η εγκατάσταση του Globus Toolkit. Έπειτα η ρύθμιση είναι τόσο απλή όσο η εκτέλεση της παρακάτω εντολής σε γραμμή εντολών σε Windows ,

```
%GLOBUS_LOCATION%\etc\globus-devel-env.bat
```

ή τις παρακάτω εντολής σε συστηματα Linux.

```
$ . $GLOBUS_LOCATION/etc/globus-devel-env.sh
```

Έπειτα η εκτέλεση του κώδικα γίνεται με την χρήση της εντολής java όπως για παράδειγμα παρακάτω:

```
java -DGLOBUS_LOCATION=%GLOBUS_LOCATION% foo.MyClass
```

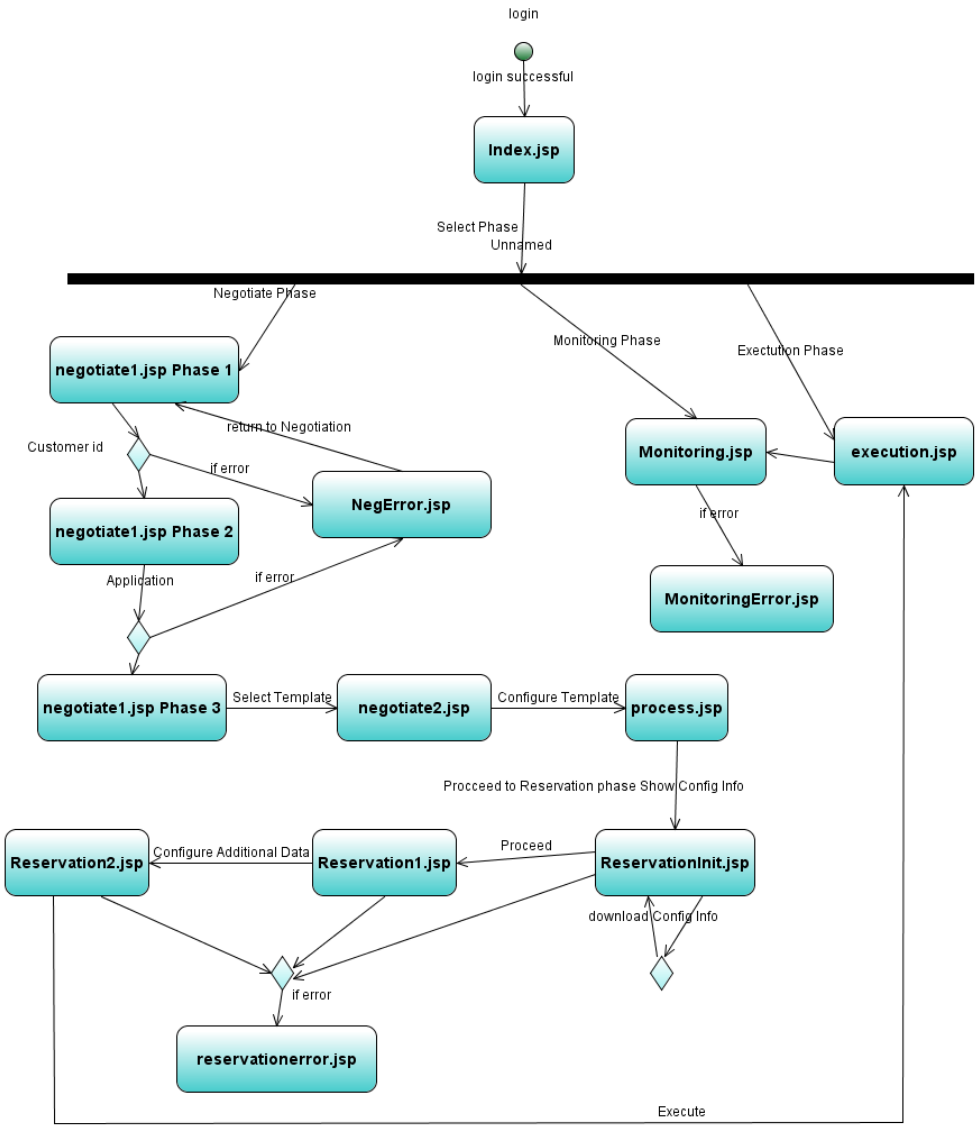
Το μεγάλο πρόβλημα όμως είναι πως η παραπάνω λύση δεν είναι διαθέσιμη στην προκειμένη περίπτωση για πολλούς λόγους. Κατά πρώτον τα JSP’s απαιτούν η μεταγλώτιση τους να γίνει από τον μεταγλωτιστή που παρέχει ο ίδιος ο server στον οποίο τελικά θα εγκατασταθούν(εν προκειμένω ο Jakarta). Μαλιστα η ίδια η διαδικασία της μεταγλώτισης τους εμπεριέχει τις ρυθμίσεις παρα πολλών παραμέτρων του συστήματος και την ύπαρξη επιπροσθετων βιβλιοθηκών σε συγκεκριμένες θέσεις στο σύστημα,προβλήματα τα οποία αναλαμβάνουν να λύσουν τα Ant scripts που παρέχονται από το ίδιο τον server(Apache Tomcat)[21] . Οι ρυθμίσεις αυτές για την μεταγλώτιση των JSP αρχικά αγνοούν εντελώς τις ρυθμίσεις που γίνονται από το script globus-devel-env για το Globus , και σε περιπτωση που (βεβιασμένα με παρέμβαση μας) τις λάβουν υπόψη έρχονται σε σύγκρουση με αποτέλεσμα να μην γίνεται η μεταγλώτιση.

Ο δευτερος λόγος που δεν μπορούμε να χρησιμοποιήσουμε τις επίσημες ρυθμίσεις του Globus Toolkit είναι ότι η εκτέλεση της εφαρμογής γίνεται μέσα στο περιβάλλον του Server ο οποίος επιβάλει τις δικές του ρυθμίσεις συστήματος , και αγνοεί εντελώς τις ρυθμίσεις των scripts του Globus Toolkit. Αυτό έχει ως αποτέλεσμα ακόμα και αν γίνει σωστά η μεταγλώτιση των JSP's εμφανίζονται σφάλματα κατά τον χρόνο εκτέλεσης (run-time errors), κυρίως σφάλματα έλλειψης βιβλιοθηκών αλλά και ασυμβατότητες με τις εκδόσεις των βιβλιοθηκών(αφού αρκετές από αυτές είναι κοινές μεταξύ του Server και του Apache Tomcat αλλά σε διαφορετικές εκδόσεις)[22].

Η λύση που τελικά επιλέχθηκε (ελλείψη και κάποιας επίσης τεκμηρίωσης) είναι απλή. Θα γίνει χρήση του Netbeans IDE για την ανάπτυξη του κώδικα , την ρύθμιση όλων των απαραίτητων παραμέτρων για την μεταγλώτιση των JSP's και για την εγκατάσταση τους στο περιβάλλον του Server(διαδικασία που είναι αυτοματοποιημένη στο Netbeans). Για την χρήση του Globus Toolkit αντιγράφουμε όλες τις βιβλιοθήκες που παρέχονται κατά την εγκατάσταση τους στο περιβάλλον των JSP's δηλαδή στο φάκελο `/WEB-INF/lib` ,διατηρώντας την σχετική τους θέση(όσες βιβλιοθήκες βρισκόταν αρχίκα στο φάκελο libs του Globus μεταφέρονται στον `/WEB-INF/lib` , όσες βρισκόταν στο `/libs/common` μεταφέρονται στον /`WEB-INF/lib/common`). Επίσης διαπιστώθηκε ότι κατά την εκτέλεση κώδικα του Globus απαιτείται να υπάρχει το αρχείο client-config.wsdd , που βρίσκεται μέσα στην εγκατάσταστη του Globus Toolkit , σε κάποιον από τους φακέλους που ορίζονται στο classes CLASSPATH δηλαδή θα πρέπει η εικονική μηχανή της Java να έχει ορατότητα στο αρχείο αυτό . Για το λόγο αυτό το αρχείο αυτό αντιγράφεται από την εγκατάσταση του Globus Toolkit στο φάκελο `/WEB-INF/classes` των JSP(είναι φάκελος που υπάρχουν τα μεταγλωτισμένα αρχεία κώδικα Java που περιέχονται στην εφαρμογή).

8 Περιγραφή της Υλοποίησης

Στην παράγραφο αυτή θα περιγράψουμε την ακριβή υλοποίηση της εφαρμογής για κάθε σελίδα που έχει δημιουργηθεί ξεχωριστά παρουσιάζοντας επίσης και τα κυριότερα κομμάτια πηγαίου κώδικα(για τον πλήρη κώδικα βλ. Παραρτημα Α). Στο παρακάτω διάγραμμα (**Εικ.10**) για λόγους ευκολίας και αναφοράς παρουσιάζεται σε μορφή διαγράμματος ακολουθίας η συνολική ροή των σελίδων ανάλογα με τις επιλογές του χρήστη. Με βάση το διάγραμμα αυτό μπορεί να γίνει κατανοητή η πορεία του χρήστη μέσα από τις σελίδες της εφαρμογής.

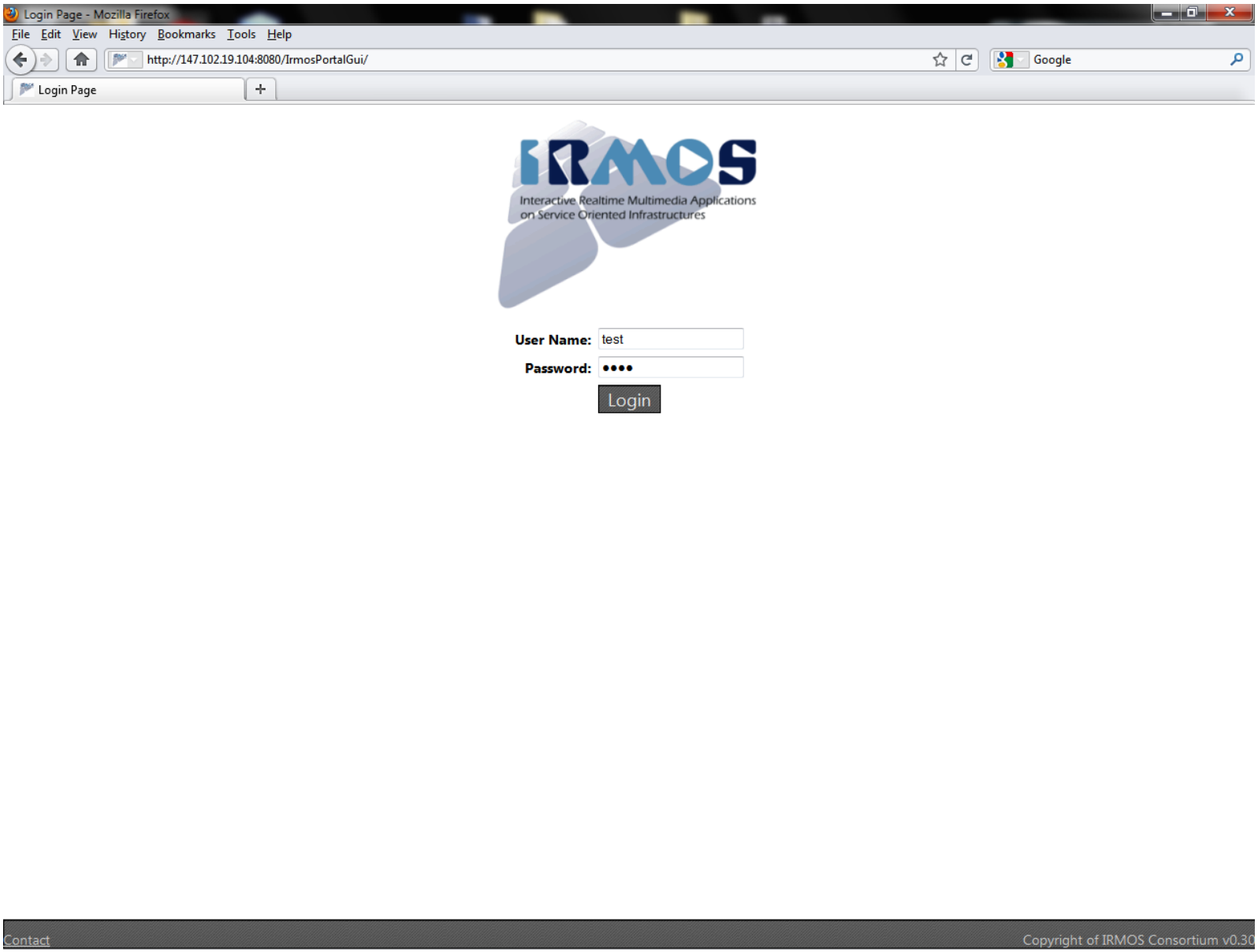


Εικόνα 8.1 Διάγραμμα ακολουθίας σελίδων

8.1 Login.jsp

Αυτή είναι η πρώτη σελίδα που βλέπει ο χρήστης κατά την είσοδο στην εφαρμογή. Επίσης η σελίδα αυτή εμφανίζεται αυτόματα σε περίπτωση που η σύνοδος του χρήστη λήξει. Περιλαμβάνει δύο πεδία τα οποία καλείται ο χρήστης να συμπληρώσει με τα στοιχεία του Username και Password. Η διασφάλιση της εφαρμογής γίνεται μέσω της ρύθμισης του Apache Tomcat Server κατά το deployment της εφαρμογής , ρυθμίζοντας κατάλληλα το αρχείο *web.xml* (Context Descriptor για την Web Application). Τα στοιχεία username,password θα πρέπει να υπάρχουν στην λίστα χρηστών του ApacheTomcat , συγκεκριμένα στο αρχείο users.xml(στο φάκελο του Tomcat) και επίσης έχει οριστεί ότι για ένα σωστό login ο χρήστης θα πρέπει να έχει ρόλο manager ή user. Ο κώδικας της σελίδας είναι απλή HMTL φόρμα αλλά για την ταυτοποίηση των χρηστών χρησιμοποιήθηκε η μέθοδος j_security_check που παρέχεται από τις βιβλιοθήκες του Apache Tomcat. Για την ορθή χρήση της μεθόδου αυτής τα πεδία username και password πρέπει αναγκαστηκά να ονομάζονται j_username και j_password αντίστοιχα.

```
      <form name="login" method="POST" action='<%=
response.encodeURL("j_security_check")%>'>
      ……
      <input type="text" name="j_username">
      ……
      <input type="password" name="j_ password ">
      ……
```



Εικόνα 8.2 Login.jsp

8.2 Index.jsp, Navbar.html, Footer.jsp και newcss.css

Η κεντρική σελίδα της εφαρμογής στην οποία βρίσκονται όλες οι επιλογές που μπορεί να κάνει ο χρήστης είναι η *index.jsp.* Η σελίδα αυτή δίνει και την αισθητική της όλης εφαρμογής για την οποία γίνεται προσπάθεια να διατηρηθεί στις υπόλοιπες σελίδες. Ουσιαστικά χωρίζει το χώρο σε πέντε διακριτά μέρη με χρήση του HTML tag * * . Το πάνω μέρος όπου αναγράφεται με κόκκινα γράμματα ο τίτλος της σελίδας κάτι που διατηρείται και στις υπόλοιπες σελίδες και υπενθυμίζει στο χρήστη σε ποια λειτουργία βρίκεται(Negotiation,Monitoring κτλ.) . Ακριβώς από κάτω βρίσκεται η εικόνα και λογότυπο του Irmos Project. Αριστερά βρίσκεται το κεντρικό μενού το οποίο δημιουργείται συμπεριλαμβάνοντας το αρχείο Navbar.html .

```
…<jsp:include page="navbar.html"/>  ….
```

Ακριβώς δεξία από αυτό υπάρχει ο κενός χώρος στον οποίο εμφανίζεται η φάση του Monitoring ή το εγχειρίδιο της εφαρμογής όταν ο χρήστης το επιλέξει. Η επιλογή αυτή γίνεται με χρήση παραμέτρων που δέχεται η σελίδα από την navbar.html και βάση της επιλογής του χρήστη.

```
<%if (request.getParameter("monitoring")!=null){%>
                <jsp:include page="monitoring.jsp" flush="true" />
                <%}%>
                <%if (request.getParameter("Faq")!=null){%>
                  <jsp:include page="Manual.txt" flush="true" />
                <%}%>
```

Τέλος στο κάτω μέρος υπάρχει το υποσέλιδο της σελίδας με πληροφορίες και επιλογές για την εφαρμογή όπως η έκδοση του προγράμματος και η αποστολή mail στους διαχειριστές. Επίσης το υποσέλιδο αυτό είναι κάτι που παραμένει σταθερό σε όλη την εφαρμογή. Το υποσέλιδο επίσης δημιουργείται από ξεχωριστό αρχείο , *footer.jsp*.

```
<jsp:include page="footer.jsp"
```

Η σελίδα Navbar.html είναι μια απλή σελίδα HTML(χωρίς δυναμικό κώδικα) η οποία δημιουργεί ένα πίνακα με τα κατάλληλα links στις σελίδες με τις υπόλοιπες λειτουργίες της εφαρμογής.

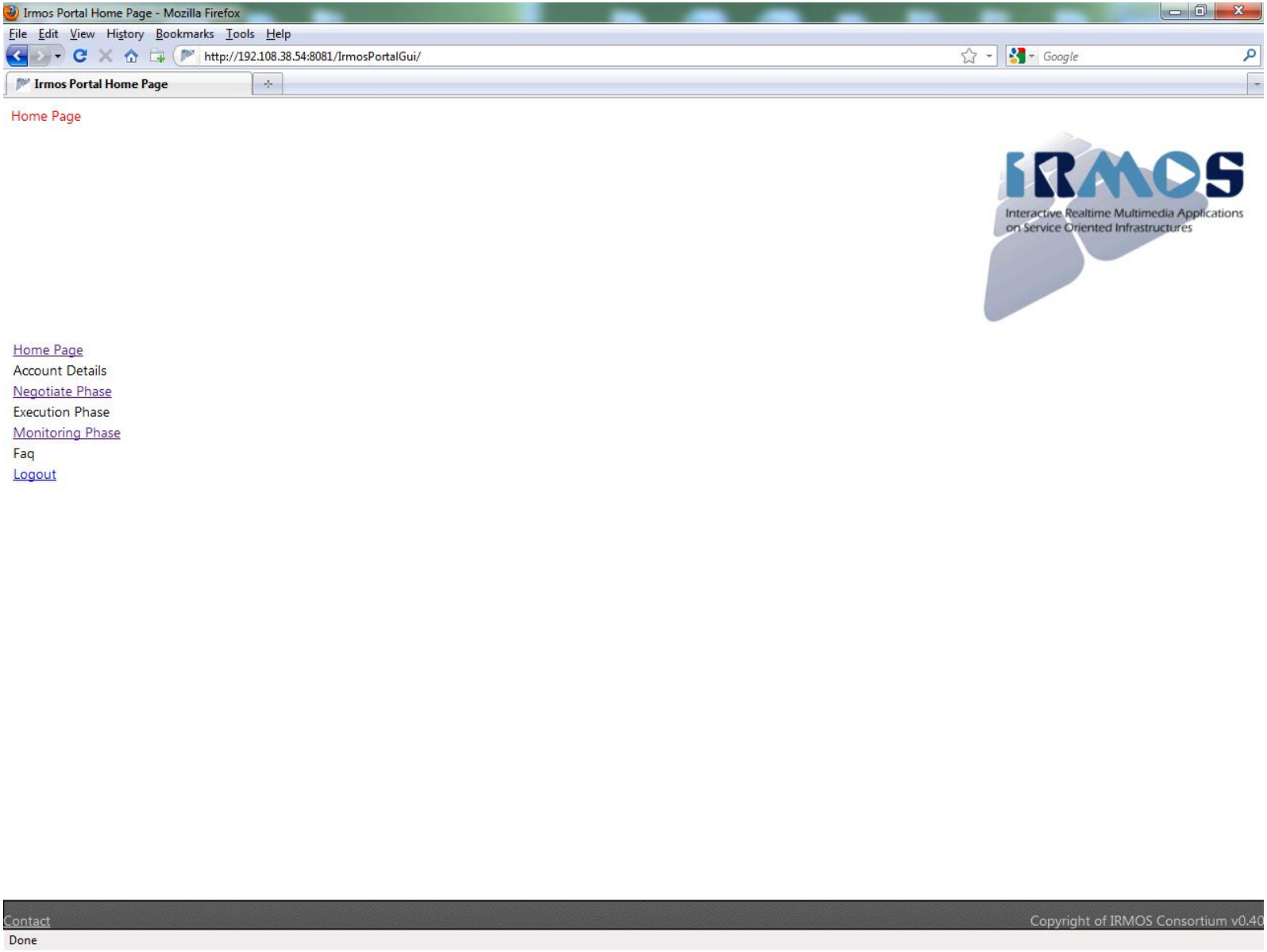
```
<tr>
                        <td><a
href="negotiate1.jsp?phase=1">Negotiate Phase</a></td>
```

```
                    </tr>
                    <tr>
                        <td>Execution Phase</td>
                    </tr>
                    <tr>
                        <td><a href="index.jsp?monitoring">Monitoring
Phase</a></td>
                    </tr>
```

Η σελίδα *footer.jsp* είναι επίσης πολύ απλή και αποτελείται από δύο HTML tags που με χρήση ρυθμίσεων css βρίσκονται πάντα κολημμένα στο κάτω μέρος της σελίδας. Επίσης υπάρχει ένα απλό mailto: link για την αποστολή email στους διαχειριστές.

```
    <div style="position:fixed;bottom:0;left:0;border:1px solid
black; height:2em;clear:both;font-size:14px;background:
url(templatemo_footer.jpg) repeat-x;width: 100%;">
```

Τέλος η σελίδα *newcss.css* καθορίζει την εμφάνιση ολόκληρης της εφαρμογής με χρήση κανόνων css. Ορίζει τις γραματοσειρές που χρησιμοποιούνται και το μέγεθος τους καθώς και την εμφάνιση των κουμπιών της εφαρμογής. Η σελίδα αυτή συμπεριλαμβάνεται σε κάθε σελίδα της εφαρμογής για να ορίζει την εμφάνιση τους.



Εικόνα 8.3 Index.jsp

8.3 Monitoring.jsp

Η σελίδα Monitoring.jsp είναι αυτή που υλοποιεί την φάση του ελέγχου όπως περιγραφηκε παραπάνω. Η σελίδα ενσωματώνεται στην κεντρική σελίδα με χρήση της εντολής *<jsp include>* , αλλάζοντας και τον κόκκινο τίτλο της σελίδας να αντικατοπτρίζει την φάση στην οποία βρίσκεται ο χρήστης. Η φάση του Monitoring διαβάζει ,μορφοποιεί και εμφανίζει δεδομένα από το Webmds που έχουν την παρακάτω μορφή(τα δεδομένα σε κόκκινη γραματοσειρά πρέπεί να παραβλέπονται):

```
<MonitoringData>
<VSN_ID>http://irmos.rus.uni-stuttgart.de/550e8400-e29b-11d4-a716-446655440000</VSN_ID>
<ASC_ID> dustbuster2</ASC_ID>
<timestamp>2010-02-02 15:06:13:0499</timestamp>

<MonitoringParameter>
<Name>state</Name>
<Value>started</Value>
<Type>string</Type>
<Unit>enum</Unit>
</MonitoringParameter>

<MonitoringParameter>
<Name>fps</Name>
<Value>25</Value>
<Type>float</Type>
<Unit>frames</Unit>
</MonitoringParameter>

<WorkfowParameters>
        <x>10</x>
</WorkflowParameters>

</MonitoringData>
```

Μέσα από το VSN_ID και συγκεκριμένα από το τελευταίο μέρος του βρίσκουμε το αρχείο ASLA από το οποίο προέκυψε η εφαρμογή που εκτελείται (από την οποία προκύπτουν τα δεδομένα που ελέγχουμε) και βάση αυτού θα πρέπει να συγκρίνουμε τις τιμές των παραμέτρων που προκύπτουν(από το monitoring) με τις τιμές που αναμένουμε(στο αρχείο ASLA).

Για την ανάγνωση των δεδομένων XML χρησιμοποιούνται τα jsp tags που μας επιτρέπουν να διατρέχουμε το XML δέντρο με χρήση της τεχνικής XPATH. Αρχικά στον κώδικα ελέγχεται αν υπάρχουν οποιαδήποτε δεδομένα και αν όχι εμφανίζεται το σχετικό μήνυμα «No Monitoring Data».

```
<x:when select="$DOC//MonitoringData"/>
```

```
                    <x:otherwise><div
style="clear:both;padding:5em;font-style:italic;color:RED;text-
align:center">No Monitoring Data Available</div></x:otherwise>
                </x:choose>
```

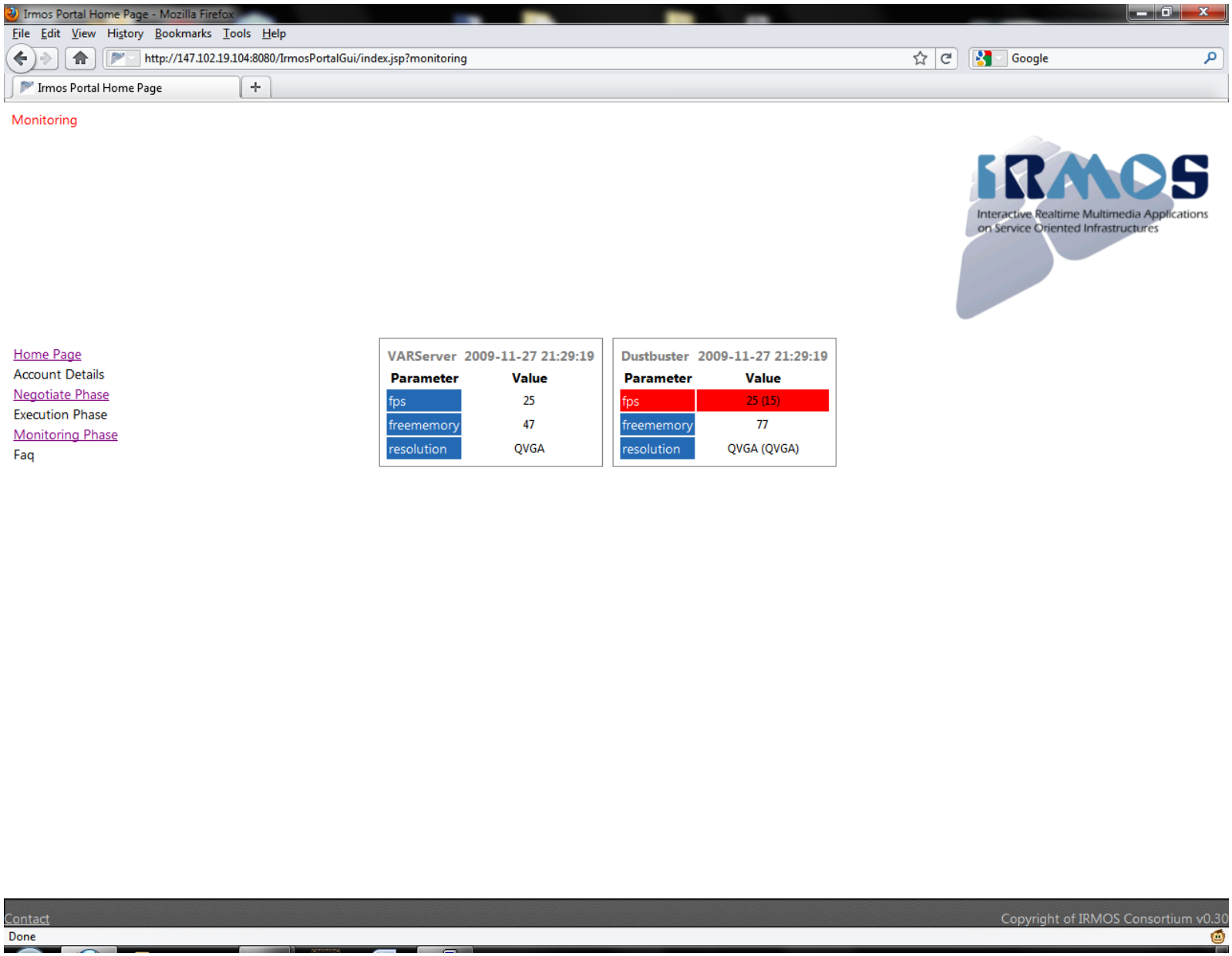
Έπειτα ξεκινάει ένας βρόγχος για κάθε MonitoringData που βρίσκουμε ,στον οποίο αρχικά διαβάζουμε τα VSN_ID , ASC_ID και timestamp. Βάση του VSN_ID βρίσκουμε το ASLA αρχείο και (αν υπάρχει) το ανοίγουμε και το διαβάζουμε.

```
                <x:forEach var="Data" select="$DOC//MonitoringData">
      …
                <x:set var="vsnid" select="string($Data/VSN_ID)"/>
                <x:set var="ascid" select="string($Data/ASC_ID)"/>
                <c:set var="split" value="${fn:split(vsnid, '/')}" />
                <c:set var="datafile" value="${split[fn:length(split)-
1]}"/>
      ……….
       <x:parse doc="${s}" var="sla1"/>
```

Αμέσως μετά δημιουργείται ένας HTML πίνακας με επικεφαλίδες τα VSN_ID, ASC_ID και timestamp που μόλις διαβάσαμε. Στο κυρίως μέρος του πίνακα ξεκινάει ένας δευτερος βρόγχος για κάθε MonitoringParameter. Στον βρόγχο αυτό διαβάζονται οι τιμές Name και Value για κάθε παράμετρο και εμφανίζονται στο κυρίως μέρος του πίνακα. Επίσης με χρήση ενός αρκέτα μεγάλου XPATH διαβάζεται η τιμή της συγκεκριμένης παραμέτρου από το ASLA αρχείο και αν υπάρχει ελέγχεται με την τιμή που μόλις διαβάστηκε από το δεδομένα ελέγχου.Να σημειωθεί ότι η συγκριση γίνεται αλφαριθμητικά και ελέγχεται αν είναι ίσα ή διάφορα μόνο(όχι μεγαλύτερο μικρότερο).

```
      table style="…">
                <thead>
      <th style="…"><c:out value="${vsnid}"/></th>
      <th style="…"><c:out value="${ascid}"/></th>
      <th style="…"><x:out select="$Data/timestamp"/></th>
      <x:forEach var="MonitoringParameter"
select="$Data/MonitoringParameter" >
      …
       <x:set var="Name" select="string($MonitoringParameter/Name)"/>
       <x:set var="Value" select="string($MonitoringParameter/Value)"/>
      …
      <x:set var="defaultvalue" select="string($sla1//*[local-
name()='components' and @id=$ascid]/parameters[@name=$Name]/*[local-
name()='BasicValue']/@value)"/>
                              <%   }%>
                        <tr>
       <c:if test="${Value!=defaultvalue && !empty defaultvalue}">
          <% Color = "RED"; %>
         </c:if>
```

```
      <td style="….."><c:out value="${Name           </td>
            <td style="…:<% if(Color.equals("RED")){out.print(Color);}%>">
                  <c:out value="${Value}"/>
                 <c:if test="${!empty defaultvalue}">
                (<c:out value="${defaultvalue}"/>)
                                  </c:if>
                   </td>
                 </tr>
                   </x:forEach>
```



Εικόνα 8.4 Monitoring.jsp Σελίδα φάσης Monitoring

8.4 Negotiate1.jsp

Η σελίδα Negotiate1.jsp χωρίζεται σε τρείς φάσεις. Οι φάσεις αυτές διαχωρίζονται με χρήση μιας παρεμέτρου που δέχεται η σελίδα με όνομα *"Phase"* και τιμή έναν ακέραιο που αντιπροσωπευει την φάση στην οποία βρίσκεται(π.χ. *Negotiate1.jsp?phase=2*). Δεν υπάρχουν περιθώρια παραποίησης αυτής της παραμέτρου αφού εσωτερικά γίνονται έλεγχοι με κώδικα για το ποια φάση βρίσκετε η σελίδα ώστε να γίνουν οι απαραίτητες ενέργειες. Παραποίηση της παραμέτρου χρησιμοποιώντας για παράδειγμα String αντί ακεραίου επιφέρουν μυνήματα λάθους και δεν συνεχίζει η διαδικασία , αλλά ούτε και άλματα φάσης με παρέμβαση του χρήστη αφού κάθε φάση απαιτεί δεδομένα από την προηγούμενη για να ολοκληρωθεί. Εμφανισιακά η σελίδα ακολουθεί την αισθητική της εφαρμογής όπως έχει οριστέι και παραπάνω, με κόκκινη γραμματοσειρά εμφανίζεται πάνω αριστερά η φάση στην οποία βρισκόμαστε(Negotiation) , δεξιά υπάρχει το λογότυπο του IrmosPortal ενώ στο κάτω μέρος υπάρχει και πάλι το υποσέλιδο.

Στην σελίδα αυτή γίνεται και για πρώτη φορά η αρχικοποίηση των Jsp Beans που θα χρησιμοποιηθούν καθόλη την διάρκεια της εφαρμογής . Τα Jsp Beans είναι ειδικά αντικείμενα της Java(με κώδικα είτε δικό μας είτε από τις βιβλιοθήκες) τα οποία δημιουργούνται και καταστρέφονται αυτόματα από τον Container που εκτελεί τις σελίδες Jsp(εδώ ο Apache Tomcat). Η δημιουργία τους γίνεται αυτόματα την πρώτη φορά που τα «ενεργοποιήσουμε» με χρήση της <Jsp:useBean> . Για κάθε Bean ορίζουμε το ευρος για το οποίο αυτά είναι διαθέσιμα(scope) το οποίο μπορεί να είναι η τρέχουσα σελίδα(page), η τρέχουσα συνεδρία(session) ή ολόκληρη η εφαρμογή(application). Όταν περάσει το ευρος που ορίσαμε αυτόματα τα Beans καταστρέφονται. Στην συγκεκριμένη περίπτωση χρησιμοποιούμε για ευρος την συνεδρία(session) γιατί στα Beans αποθηκευουμε πληροφορίες που θέλουμε να χρησιμοποιήσουμε και στις παρακάτω σελίδες ενώ θέλουμε ο χρήστης να μπορεί να τα αλλάξει αν θέλει (λήγωντας την συνεδρία) ή πολλοί χρήστες ταυτόχρονα να έχουν διαφορετικά μεταξύ τους δεδομένα.

```
        <jsp:useBean id="reqsla" scope="session"
class="ntua.irmos.wp5.stubs.singleton.IrmosPortal.Requestaslatemplat
e" />
        …
      <jsp:useBean id="service" scope="session"
class="irmosgui.serviceURI" />
```

Στην πρώτη φάση της σελίδας αυτής (*Phase=1*) εμφανίζεται μόνο ένα πλαίσιο κειμένου στο οποίο ο χρήστης συμπληρώνει το Customer id του. Πατώντας το πλήκτρο

Procced προχωρά στην δευτερη φάση της σελίδας.Επίσης κατά την φάση αυτή καταστρέφουμε τα Beans που ενδέχεται να έχουν υπάρξει από προηγούμενες απόπειρες του χρήστη ή κάποιο άλλο εσωτερικό σφάλμα του συστήματος για απόλυτη σιγουριά ότι δεν έχουμε άκυρα δεδομένα.

```
      int phase = 1;
           if (request.getParameter("phase")!=null)
            phase = Integer.valueOf(request.getParameter("phase"));

            if (phase==1){
            session.removeAttribute("reqsla");
            session.removeAttribute("service");
            session.removeAttribute("slatemplate");
            session.removeAttribute("prtl");
            }
```

Στην δευτερη φάση της σελίδας εμφανίζεται επιπλέον ένα πλαίσιο κειμένου που ζητά από τον χρήστη να εισάγει το Application id για το οποίο ενδιαφέρεται να διαπραγματευτεί. Με το πλήκτρο RequestASLA προχωρά στην τρίτη φάση της σελίδας. Η αλλαγή φάσεων γίνεται μεσα από την ίδια φόρμα HTML και για τις τρείς φάσεις χρησιμοποιώντας κώδικα Java ενσωματωμένο στην HTML.

```
      <form name="negid" action="<%if (phase==1){ %>
negotiate1.jsp?phase=2 <%}else if
(phase==2){%>negotiate1.jsp?phase=3<%}else{%>negotiate2.jsp<%}%>"
method="POST">
```

Στην τρίτη φάση που είναι και η πιο σημαντική γίνεται η λήψη των ASLA προτύπων μέσω του IrmosPortalService πριν ακόμα ο χρήστης δει οτιδήποτε στην οθόνη του. Η κλήση γίνεται μέσω της μεθόδου reqslatemplates με παράμετρο ένα αντικείμενο τύπου *ntua.irmos.wp5.stubs.singleton.IrmosPortal.Requestaslatemplate* το οποίο περιέχει τα Customer id και Application id που ο χρήστης μόλις συμπλήρωσε. Αν η κλήση αποτύχει τότε εμφανίζεται μήνυμα λάθους μέσω του μηχανισμού των Exceptions. Αν η κλήση είναι επιτυχής τότε δεχόμαστε ως απάντηση δύο πίνακες από String , ο ένας περιέχει τα EPR από το οποία μπορούμε να αντλήσουμε τα ASLA πρότυπα (όπως θα γίνει στην αμέσως επόμενη σελίδα) ενώ ο άλλος τα ονόματα τους.

```
      if (phase == 3){
                                    ……….
                                    …...
      slatemplate = service.getPrtl().requestaslatemplate(reqsla);
                                    ……
      service.setASLATemplateEPR(slatemplate.getAslaTemplatesEpr());
service.setASLATemplateType(slatemplate.getLabel());
```

```
…..
```

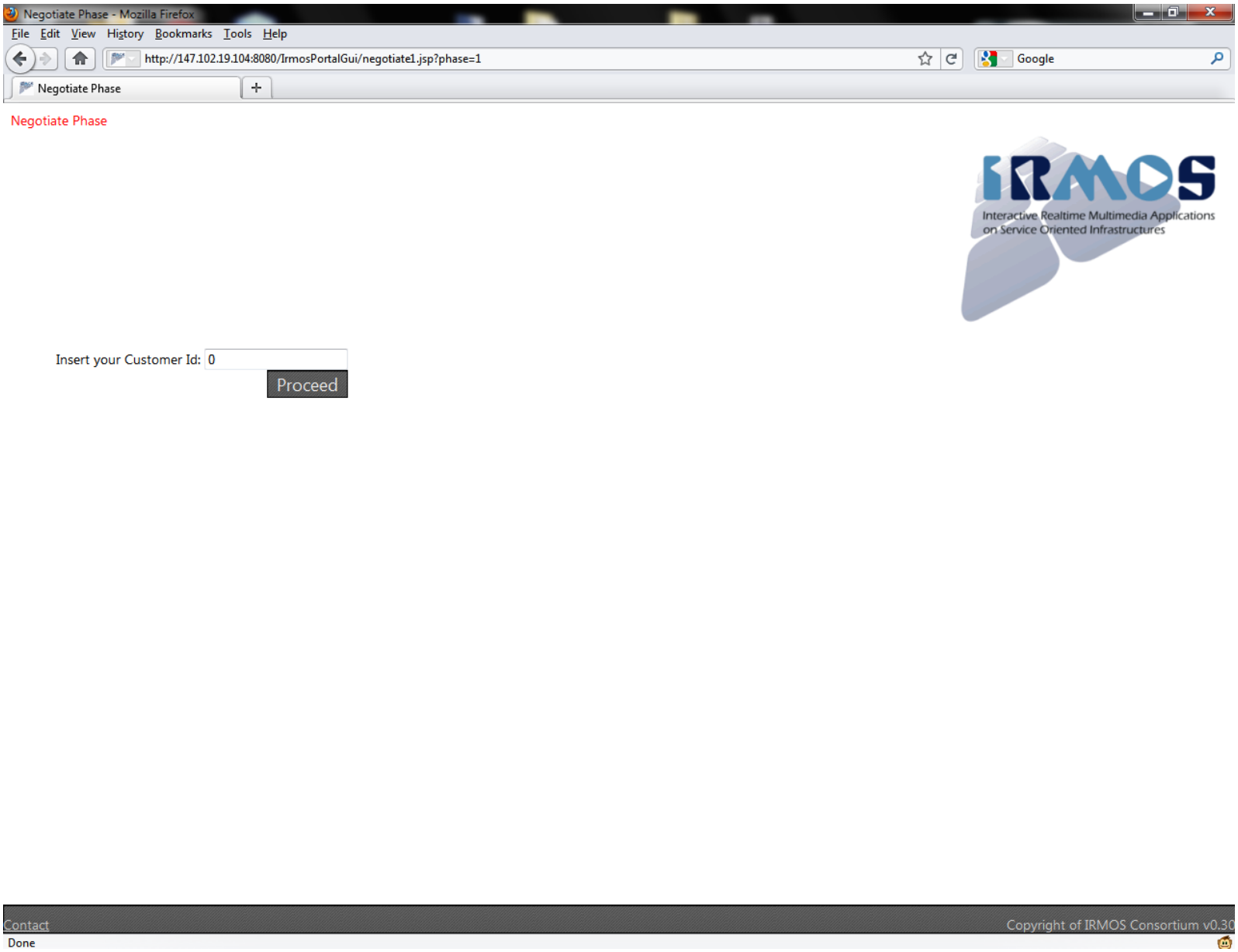
Τέλος με βάση τον πίνακα με τα ονόματα των EPR που επιστράφηκαν εμφανίζουμε στον χρήστη ένα dropdown Box που μπορεί να επιλέξει κάποιο αυτά. Πατώντας το πλήκτρο EditSLA ο χρήστης μεταφέρεται στην επόμενη σελίδα , στην οποία περνάμε ως παράμετρο τον αυξων αριθμό του προτύπου που επέλεξε ο χρήστης.

```
      <select STYLE="width: 10.5em" name="Epr">
         <%for (int i=0;i<service.getASLATemplateType().length;i++){
            String s = service.getASLATemplateType()[i];
            out.println("<option value=\""+Integer.toString(i)+"\">"+
s +"</option>"); }%>
      </select>
```

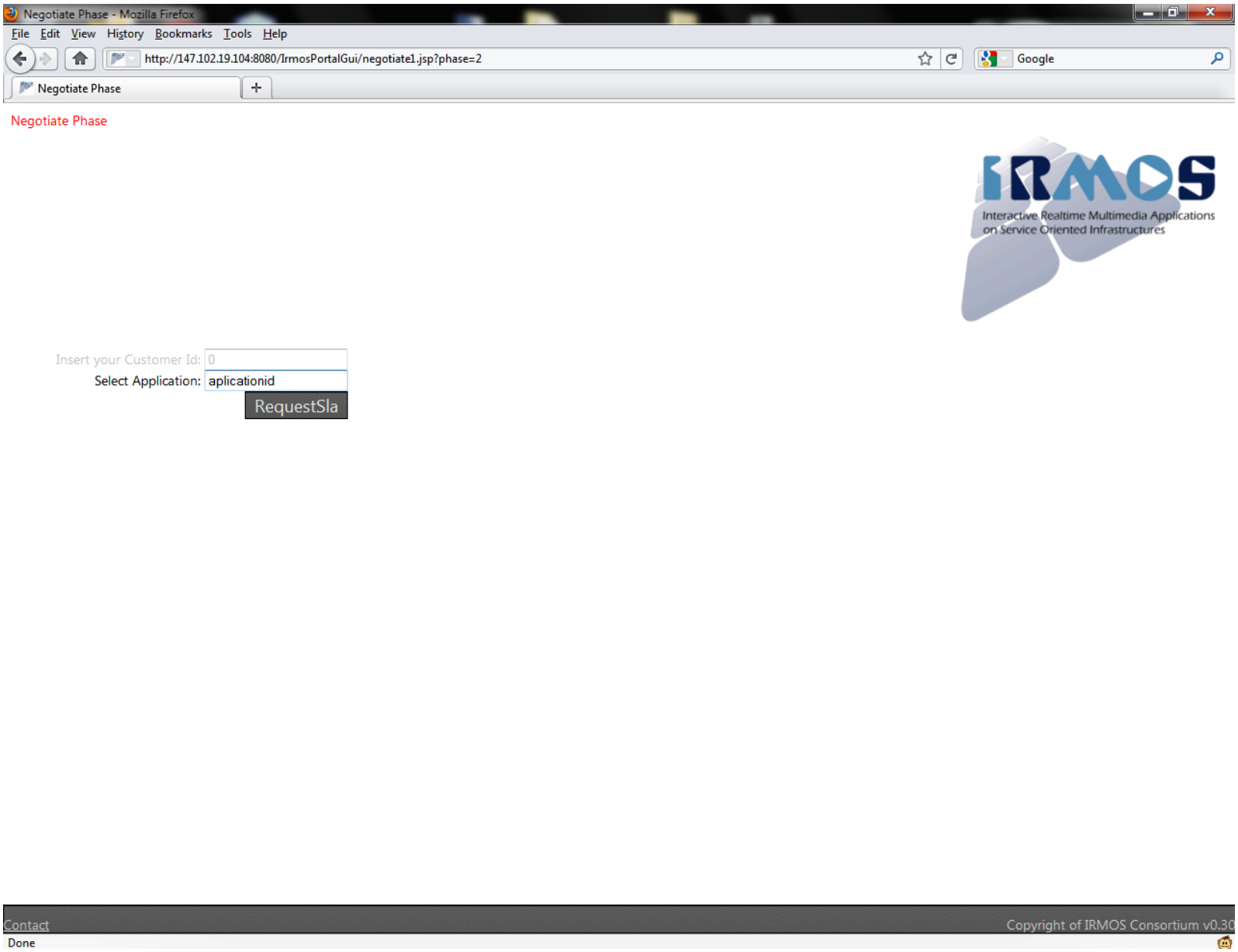
8.4.1 ServiceURI.java

Το αντικείμενο τύπου ServiceURI είναι πολύ σημαντικό για την όλη λειτουργία της εφαρμογής και στο σημείο αυτό θα πρέπει να επεξηγηθούν τα πιο σημαντικά κομμάτια του κώδικα του. Ο κύριος σκοπός του αντικειμένου αυτού είναι να κρατά πληροφορίες απαραίτητες για την ορθή λειτουργία της εφαρμογής και να είναι διαθέσιμες σε όλες τις σελίδες. Αναλυτικά τα δεδομένα που αποθηκευονται στο αντικείμενο είναι το URI που βρίσκουμε το IrmosPortalService, το URI που βρίσκουμε τα δεδομένα για την φάση του Monitoring, το φάκελο που βρίσκονται τα αρχεία ASLA και τέλος ένα αντικείμενο τύπου IrmosPortalPortType που κρατά την σύνδεση με το IrmosPortalService. Τα περισσότερα από αυτά διαβάζονται από ένα αρχείο που περιέχει τις αρχικές τιμές τους στον κατασκευαστή του αντικειμένου αυτού ενώ τα υπόλοιπα ορίζονται μέσα από τις σελίδες . Επομένως οι περισσότερες πληροφορίες είναι διαθέσιμες κατά την κατασκευή του αντικειμένου (συνήθως κατά την πρώτη φάση του Negotiation ή κατά την φάση του Monitoring).

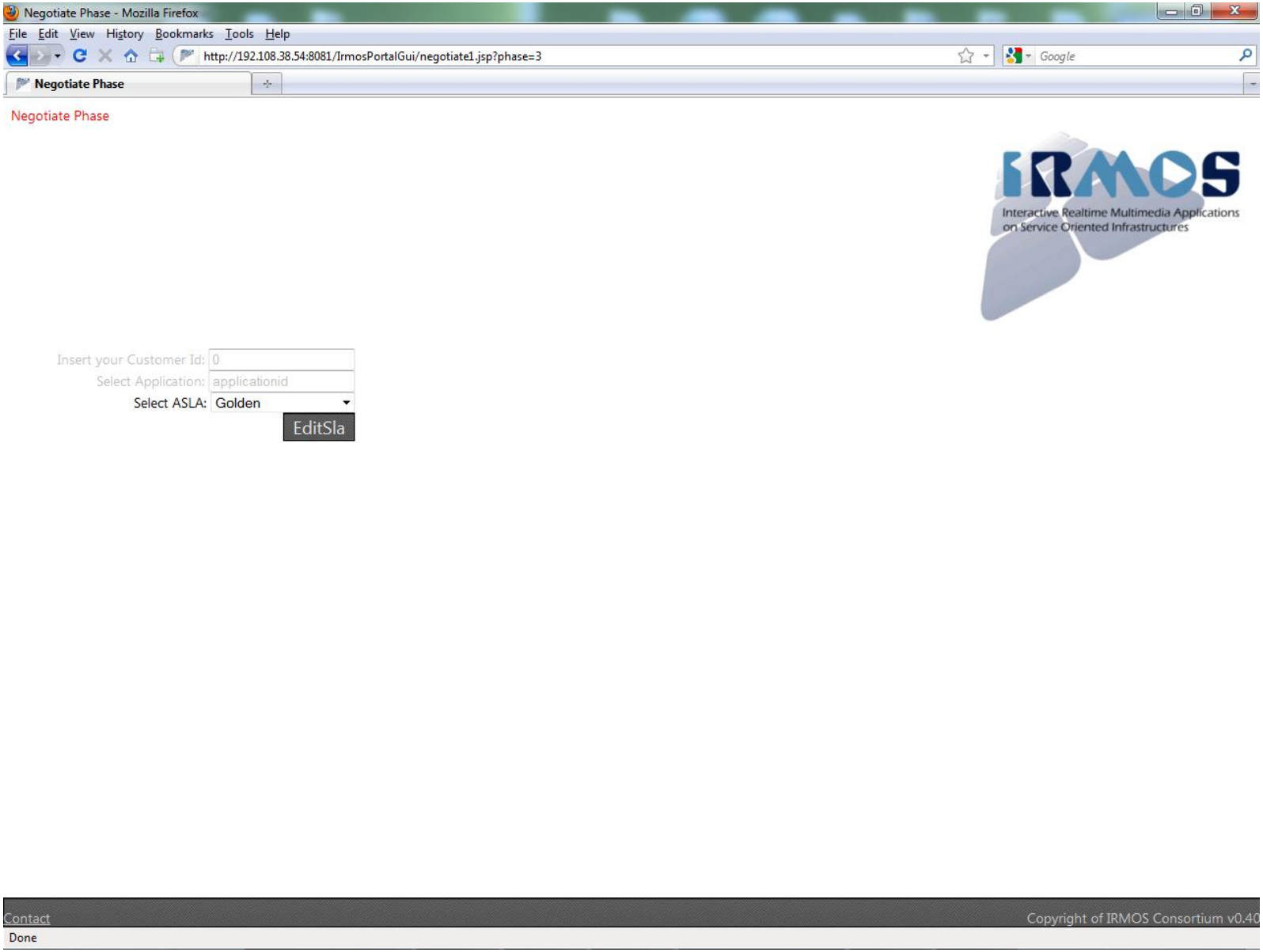
```
          public String serviceURI; // Selected ServiceURI
          public String mdsURI;
          public String datadir;
                                        ……
          IrmosPortalPortType prtl;
                                        ……
      public serviceURI() {…
       URL myURL = this.getClass().getResource("conf.properties");
                 in = myURL.openStream();
                 Properties p = new Properties();
                 p.load(in);
                 this.serviceURI = p.getProperty("serviceURI");
                 this.mdsURI = p.getProperty("mdsURI");
                 this.datadir = p.getProperty("datadir");
                 this.datadir1 = p.getProperty("datadir1");
                                        ……
```



Εικόνα 8.5 Negotiate1.jsp Phase 1



Εικόνα 8.6 Negotiation1 Phase 2



Εικόνα 8.7 Negotiate1 Phase 3

8.5 Negotiate2.jsp

Η σελίδα Negotiatiate2.jsp είναι αυτή στην οποία ο χρήστης ρυθμίζει τελικά το ASLA πρότυπο που επέλεξε στην προηγούμενη φάση. Αρχικά η σελίδα διαβάζει την παράμετρο με όνομα Epr και βάση αυτής βρίσκει το όνομα του αρχείου που περιέχει το ASLA πρότυπο που έχει διαλέξει ο χρήστης. Το αρχείο αυτό ανοίγεται και διαβάζεται ως XML .

```
….
int index = Integer.valueOf(request.getParameter("Epr"));
….
String datafile =
s.getParameters().get_any()[0].getValue().toString();
….
Document doc = builder.parse(new InputSource(new
StringReader(ASLATemplate)));
```

Στην συνέχεια δημιουργείται ένας βρόγχος για κάθε αντικείμενο με όνομα "*app:components*" που εμφανίζεται στο πρότυπο ASLA. Για κάθε ένα δημιουργείται ένα καινούργιο αντικείμενο τύπου *Component,* το οποίο αποθηκευεται σε μια λίστα για χρήση αργότερα ,και διαβάζονται οι παράμετροι που περιέχει(Ο κώδικας για την ανάγνωση των παραμέτρων υπάρχει στο *Component.java*).

```
NodeList app = root.getElementsByTagName("app:components");

            for (int i=0;i<app.getLength();i++ )
            {
                Node n = app.item(i);
                Component C = new Component(n);
                Comp.CompList.add(C);
            }%>
```

Αφού έχουν διαβαστεί και αποθηκευτει σε εσωτερικές δομές όλες οι παράμετροι που υπάρχουν στο πρότυπο ASLA , αυτές εμφανίζονται στην οθόνη σε μορφή λίστας (διπλός βρόγχος για κάθε component και για κάθε παράμετρο που αναγνώστηκε). Δίπλα από κάθε παράμετρο εμφανίζεται η αρχική τιμή της όπως επίσης διαβάστηκε από το αρχείο , την οποία ο χρήστης μπορεί να αλλάξει. Η εμφάνιση κάθε παραμέτρου στην οθόνη γίνεται μέσω της μεθόδου toHTML που έχουμε δημιουργήσει (*Parameter.java*). Η μέθοδος αυτή για παραμέτρους που μπορούν να πάρουν οποιαδήποτε τιμή εμφανίζει ένα απλό κουτί κειμένου-textbox-(με την αρχική τιμή) ενώ για παραμετρούς που μπορούν να πάρουν συγκεκριμένες τιμες από λίστα εμφανίζει ένα dropdown box με τις τιμές αυτές.

```
for (Component C : Comp.CompList)
….
for (Parameter P : C.Pl )
```

```
      ….
      <li style="clear:both;list-style:none;padding-bottom:10px;">
<div style="width:140px;float:left;"><%=P.name %></div>
<%=P.toHTML() %> </li>
      ….
```

8.5.1 Component.java

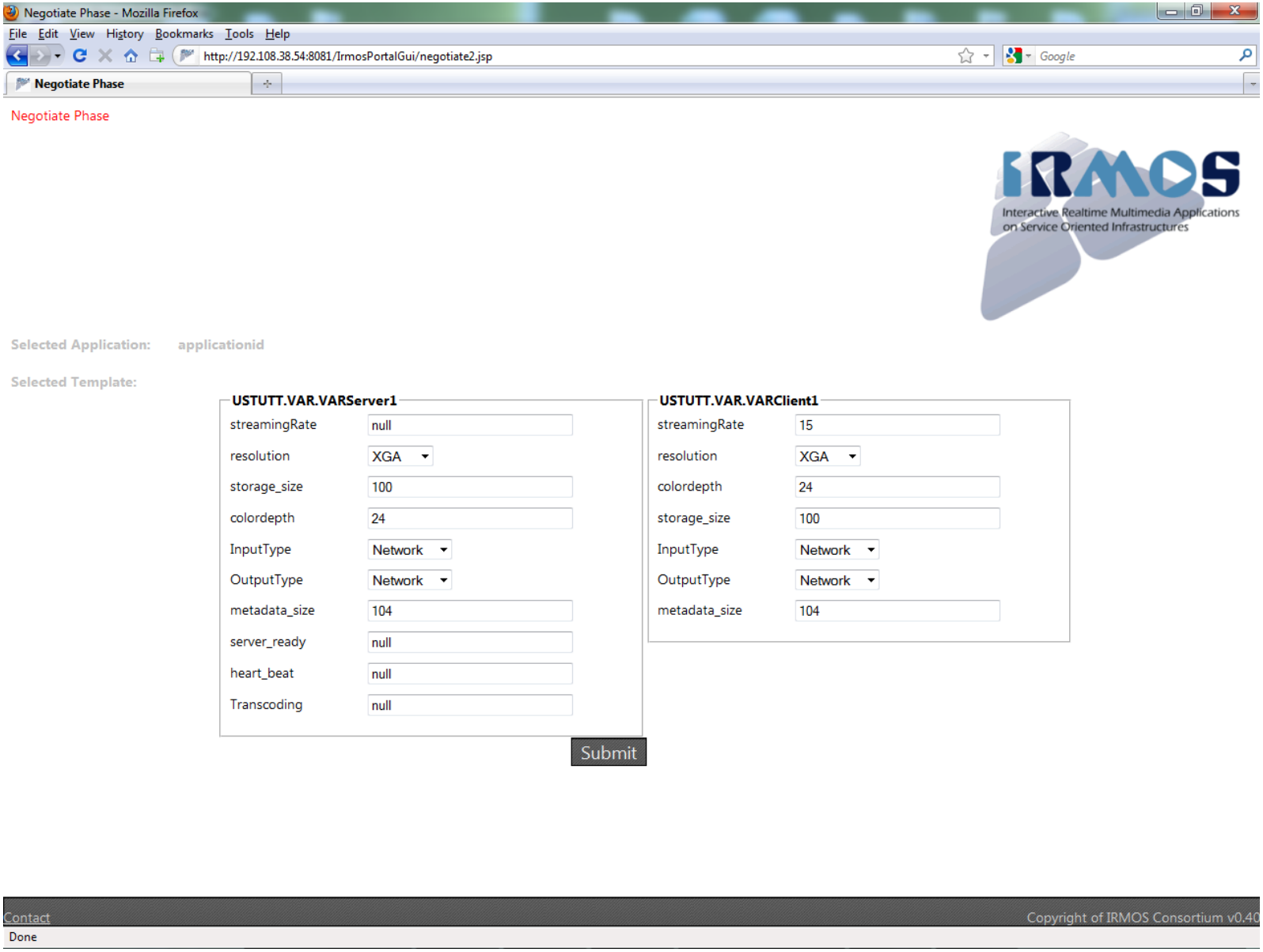
Η ανάγνωση των παραμέτρων από το ASLA πρότυπο γίνεται μέσα από τον κατασκευαστή του αντικειμένου Component. Ο κατασκευαστής δέχεται ως παράμετρο τον ίδιο τον κόμβο του XML αρχείου. Από αυτόν διαβάζει το attribute id και το αποθηκευει ως όνομα του Component. Έπειτα ένας βρόγχος για όλα τα παιδιά του Component στο XML δέντρο ψάχνει όσα έχουν όνομα "*parameters*". Για κάθε ένα από αυτά που βρίσκει δημιουργεί ένα αντικείμενο τύπου Parameter και το βάζει σε μια λίστα για περαιτέρω χρήση.

```
           public Component(Node n){
              name =
n.getAttributes().getNamedItem("id").getNodeValue();
              for(Node
childNode=n.getFirstChild();childNode!=null;childNode=childNode.getN
extSibling())
              {
               if (childNode.getNodeType()== Node.ELEMENT_NODE &&
childNode.getNodeName().equals("parameters")){
                        Parameter p = new Parameter(childNode,this);
                        Pl.add(p);
                }
              }

          }
```

8.5.2 Parameter.java

Ο κατασκευαστής των αντικειμένων Parameter που χρησιμοποιείται μέσα από τον κατασκευαστή του Component παίρνει ως παράμετρο τον κόμβο που το ορίζει στο XML δεντρο του ASLA προτύπου.Διαβάζει το όνομα κάθε παραμέτρου και ελέγχει αν ο κόμβος στο XML δένδρο έχει παιδια. Αν έχει παιδιά αυτό σημαίνει πως έχει οριστεί αρχική τιμή για την παράμετρο αυτή η οποία πρέπει να διαβαστεί. Επίσης ως παιδιά ορίζονται και οι πιθανές τιμές που μπορεί να πάρει μια παράμετρος(για παραμέτρους που παίρνουν τιμές απο λίστα) οι οποίες πρέπει να διαβαστούν επίσης.

```
      public class Parameter {
        public  Parameter(Node n,Component c){
            thisNode = n;
            Parent = c;
            if (n.hasChildNodes())
              setValuesAllowed();

            name  =
thisNode.getAttributes().getNamedItem("name").getNodeValue();
        };
```



Εικόνα 8.8 Negotiate2

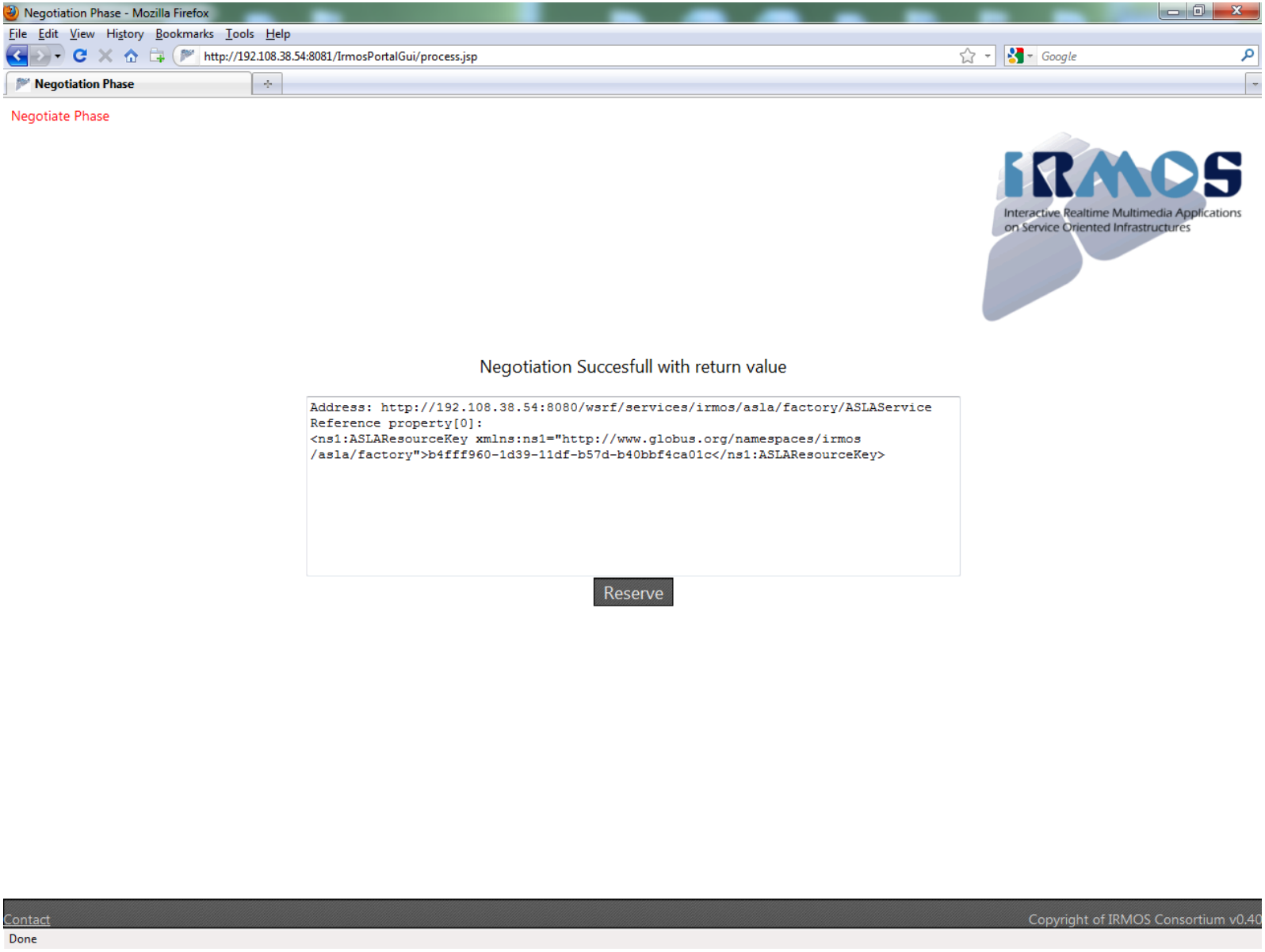
8.6 Process.jsp

Η σελίδα αυτή δρά ως ενδιάμεση της φάσης του Negotiation και του Reservation. Δέχεται ως παραμέτρους από την προηγούμενη σελίδα(*Negotiate2.jsp*) όλες τις παραμέτρους που ρύθμισε ο χρήστης βάση του προτύπου ASLA. Ο κώδικας ξεκινά με βρόγχο για όλες τις παραμέτρους που δέχθηκε η σελίδα. Για κάθε έναν από αυτούς γίνεται αναζήτηση με βάση το όνομα στις υπάρχοντες παραμέτρους που έχουν κρατηθεί εσωτερικά στο σύστημα (μέσα στο JSP bean *ComponentBean*), και όταν βρεθεί αποθηκευεται, επίσης εσωτερικά , η τιμή που έχει ορίσει ο χρήστης στην προηγούμενη σελίδα.

```
                    Enumeration keys = request.getParameterNames();
                    while (keys.hasMoreElements() )
      String value = request.getParameter(key);…
      ……
      ……
      Comp.getComponentByName(CompName).getParameterByName(ParameterN
ame).setValue(value);
           }%>
```

Στην συνέχεια δημιουργούνται δύο πίνακες από String , ο ένας περιέχει όλα τα ονόματα των παραμέτρων και ο άλλος όλες τις τιμές. Αυτοί οι πίνακες ,μαζί με επιπλέον στοιχεία όπως το Customerid και το Applicationid, δίνονται ως παράμετροι στην κλήση της μεθόδου Negotiation του IrmosPortalService.Έπειτα πραγματοποιείται η κλήση της μεθόδου Negotiation του IrmosPortalService και αν επιτύχει εμφανίζεται μια HTML φόρμα με ένα κουμπί που μπορεί να πατήσει ο χρήστης για να προχωρήσει στην φάση του Reservation. Τέλος αποθηκευεται στο Jsp Bean με όνομα Exec το τελικό ASLA που ρύθμισε ο χρήστης για χρήση στις παρακάτω σελίδες.

```
      ……
                    aslaOfferLabel.toArray(Label);
                    aslaOfferValue.toArray(Value);
               Negotiation neg = new
Negotiation(Integer.toString(reqsla.getCostumerID()),(service.getASL
ATemplateEPR()[service.getIndex()]),Label,Value);
               String ASLAEPR = service.getPrtl().negotiation(neg);
      ……..
      <form action="ReservationInit.jsp">
```



Εικόνα 8.9 Process

8.7 ReservationInit.jsp

Αρχικά στην σελίδα αυτή γίνεται κλήση της μεθόδου Reservation του IrmosPortalService με παράμετρο το ASLA που ρύθμισε ο χρήστης προηγούμενα. Η κλήση της μεθόδου αυτής επιστρέφει ένα String με πληροφορίες για την ρύθμιση του συστήματος(Config Info). Στην συνέχεια οι πληροφορίες αυτές εμφανίζονται στην οθόνη με ένα textarea και χρήση ενός html link δίνεται στον χρήστη η δυνατότητα να τις κατεβάσει στον υπολογιστή του σε μορφή αρχείου κειμένου. Επίσης εμφανίζεται ένα κουμπί με το οποίο ο χρήστης προχωρά στην επόμενη σελίδα.

```
      String ConfigInfo =
service.getPrtl().reservation(Exec.getASLA_EPR());
      …
      <textarea name="ASLA" rows="26" cols="80"
readonly="readonly"><%out.print(ConfigInfo);%>
                </textarea>
                <a href="ReservationInit.jsp?download">Download
file</a>
                </div>
                <form action="Reservation2.jsp" >
                   <input style="margin-left:45%" type="submit"
value="Configure" name="Configure" />
                </form>
      …
```

Όπως φαίνεται και παραπάνω το link για το κατέβασμα του αρχείου οδηγεί στην ίδια σελίδα αλλά αποστέλεται ως παράμετρος το String Download. Στην αρχή της σελίδας γίνεται ο έλεγχος για την ύπαρξη του String αυτού , και όταν βρεθεί αποστέλει μια HTTP request στον χρήστη που ξεκινά το κατέβασμα των πληροφοριών Config Info ως αρχείο κειμένου.

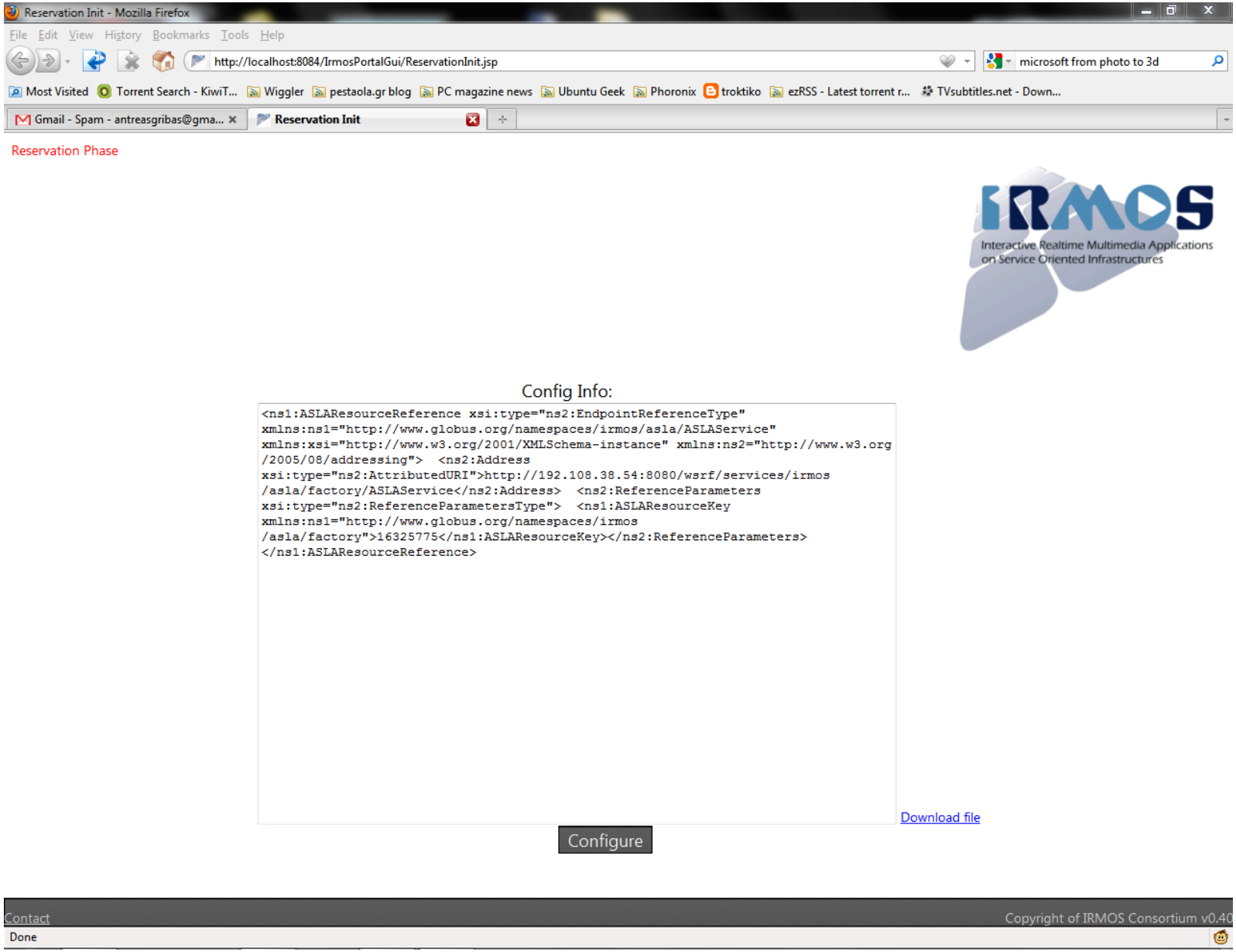
```
      if (request.getParameter("download")!=null)
              {
              ServletOutputStream myOut=null;
              try{
                  myOut = response.getOutputStream( );
                  response.setContentType("text/plain");
                  response.addHeader("Content-
Disposition","attachment; filename="+"ConfigInfo.txt");
                  response.setContentLength( (int)
Exec.getASLA_EPR().toString().length() );
                  myOut.print(Exec.getASLA_EPR().toString());
              } catch (IOException ioe){
                  throw new ServletException(ioe.getMessage( ));
                   } finally {
                       if (myOut != null)
                           myOut.close( );
                   }
```

```
                    }
```

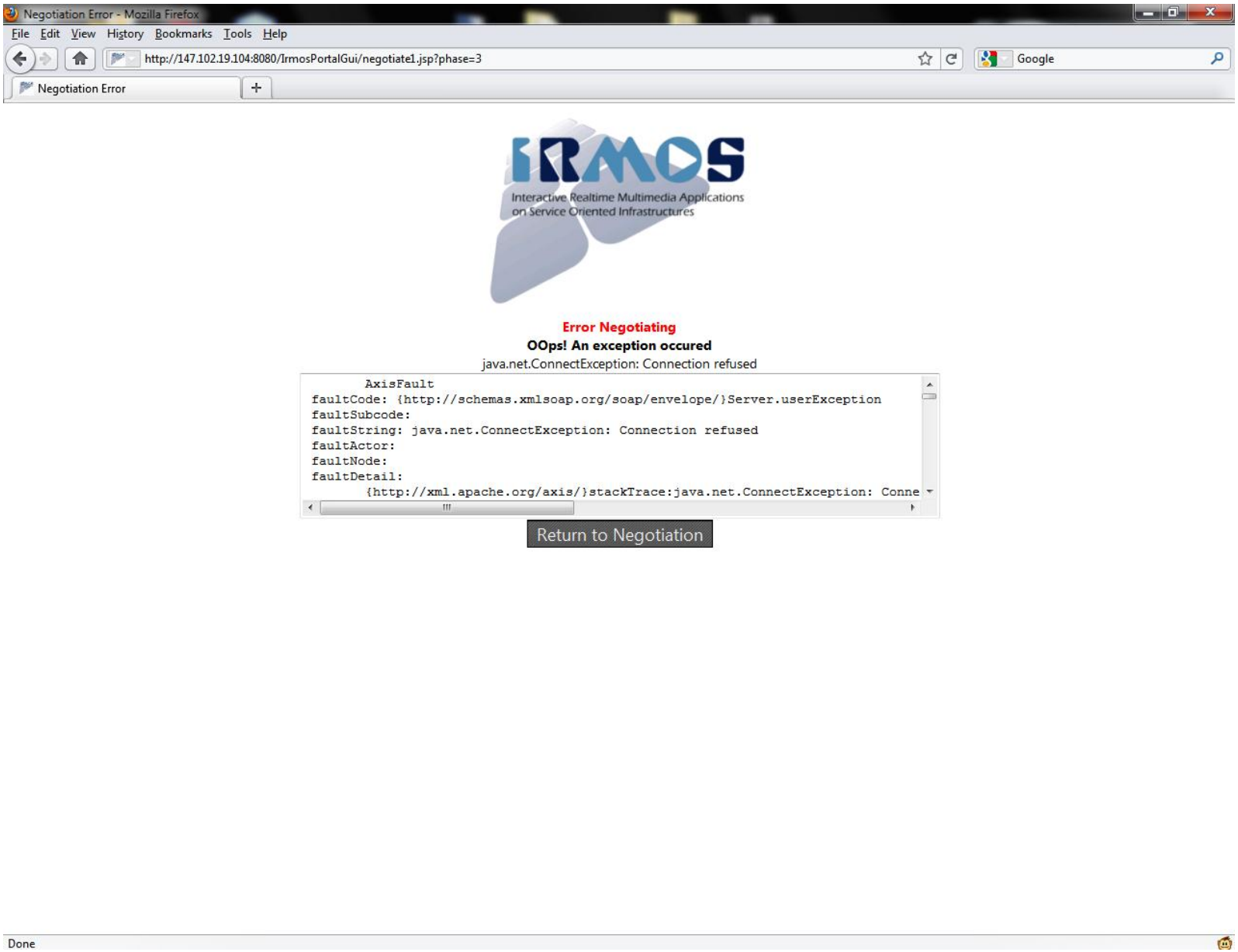
8.8 Reservation2.jsp

Η σελίδα Reservation2.jsp απλά καλεί την μέθοδο Configuration του IrmosPortalService με παράμετρο το ASLA του χρήστη. Αν η κλήση επιτύχει εμφανίζει μια απλή φόρμα με ένα κουμπί για να προχωρήσει ο χρήστης στην φάση του Execution.

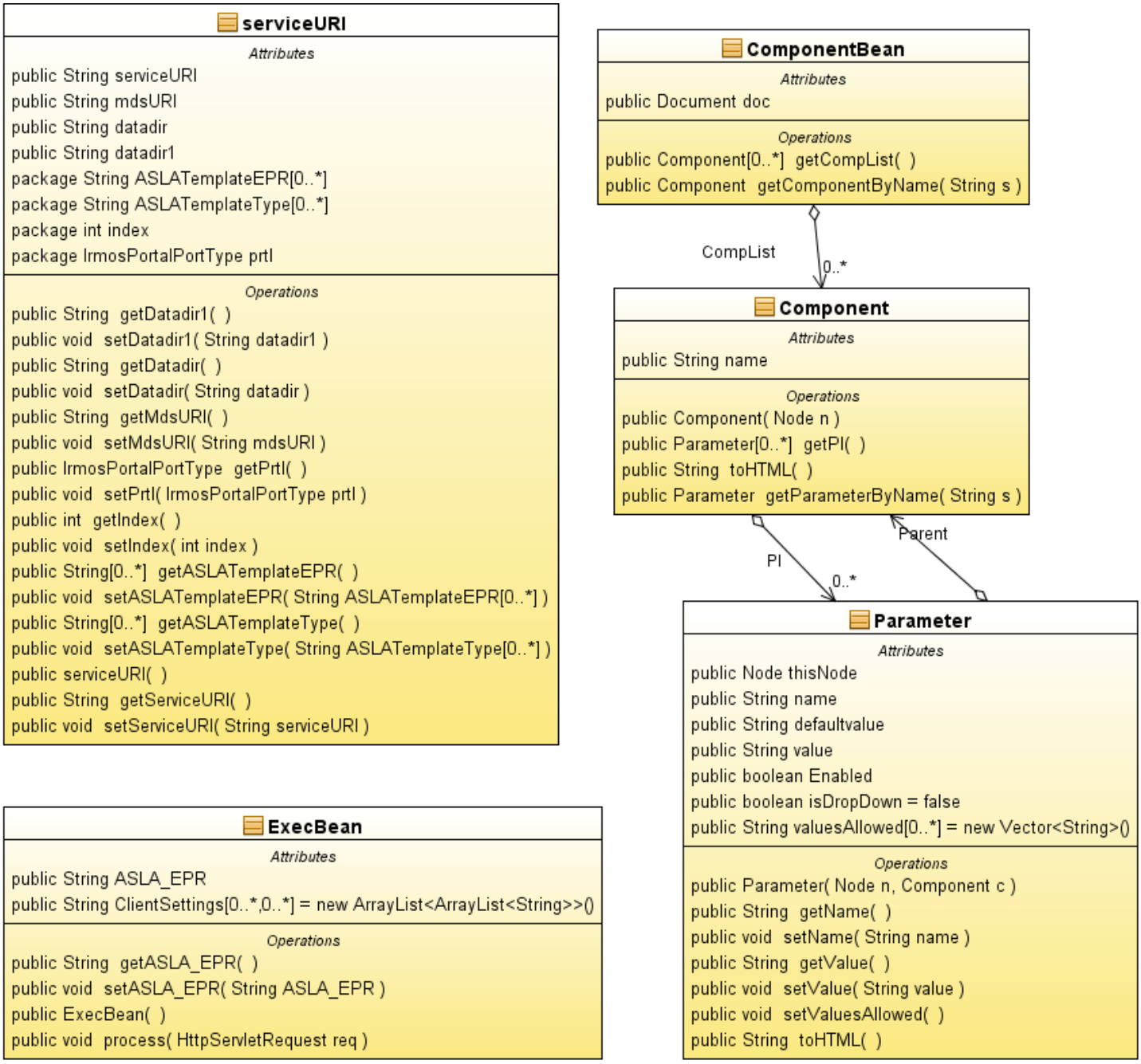
```
….
service.getPrtl().configuration(Exec.getASLA_EPR());
….
<form action="Execute.jsp" >
…..
```



Εικόνα 8.10 ReservationInit Σελίδα φάσης Reservation



Εικόνα 8.11 Σφάλμα Negotiation



Εικόνα 8.12 Διαγράμματα κλάσης για τα JSP Beans

9 *Βιβλιογραφία*

[1] http://gdp.globus.org/gt4-tutorial/singlehtml/progtutorial_0.2.1.html

[2] http://gdp.globus.org/gt4-tutorial/singlehtml/progtutorial_0.2.1.html#chap_core_first

[3] http://www.globus.org/toolkit/docs/4.0/common/javawscore/admin-index.html

[4] http://www.globus.org/toolkit/docs/4.0/common/javawscore/user-index.html

[5] Irmos Project Deliverables. WP3_3_1_2_v1_0 "IRMOS Overall Architecture" . http://www.irmosproject.eu/Deliverables/Download.aspx?ID=30

[6] Irmos Project Deliverables IRMOS_D4_2_1_v1_0 "Interface Definition of IRMOS SOI". http://www.irmosproject.eu/Deliverables/Download.aspx?ID=36

[7] Irmos Project Deliverables IRMOS_WP7_D7_3_1_v1_0 "Inital Version of Flow Control Architecture".http://www.irmosproject.eu/Deliverables/Download.aspx?ID=37

[8] Irmos Project Deliverables IRMOS WP7_D7_4_1_V1_0 "Initial Version of Path Supervision Architecture".http://www.irmosproject.eu/Deliverables/Download.aspx?ID=38

[9] Foster. "What is the Grid? A Three Point Checklist". GRIDToday, July 20,2002.

[10] Foster and C. Kesselman. The Grid 2: Blueprint for a New Computing Infrastructure.Morgan Kaufmann, 2nd edition, November 18, 2003.

[11] Fran Berman, Geoffrey Fox, and Anthony J.G. Hey. Grid Computing: Making The Global Infrastructure a Reality. John Wiley & Sons, April 8, 2003.

[12] Fran Berman, Geoffrey Fox, and Anthony J.G. Hey. Grid Computing: Making The Global Infrastructure a Reality. John Wiley & Sons, April 8, 2003.

[13] Grid Café. http://gridcafe.web.cern.ch/gridcafe/.

[14] EGEE (Enabling Grids for E-Science in Europe) Website. http://public.eu-egee.org/.

[15] Global Grid Forum (GGF). http://www.ggf.org/.

[16] OASIS. http://www.oasis-open.org/.

[17] OASIS WSRF Technical Committee.http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf.

[18] The Globus Alliance. http://www.globus.org/

[19] Apache Axis. http://ws.apache.org/axis/.

[20] Apache Jakarta Tomcat. http://jakarta.apache.org/tomcat/.

[21] http://tomcat.apache.org/tomcat-5.5-doc/deployer-howto.html

[22] http://tomcat.apache.org/tomcat-5.5-doc/class-loader-howto.html

[23] IBM WebSphere Software. http://www.ibm.com/websphere.

[24] Apache HTTPD server. http://httpd.apache.org/.

[25] W3 Schools – WSDL Tutorial. http://www.w3schools.com/wsdl/.

[26] W3 Schools–XML Schema Tutorial.http://www.w3schools.com/schema/.

[27] The Globus Toolkit. http://www.globus.org/toolkit/.

[28] Ian Foster. A Globus Toolkit Primer. (Draft). April 26, 2005. http://www.globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf.

Παράρτημα Α

Πηγαίος κώδικας προγράμματος

Κώδικας 10.1 Login.jsp

```
        <%--
            Document   : Login.jsp
            Created on : 14 Ιουλ 2009, 10:33:56 μμ
            Author     : Antreas
        --%>

        <%@page contentType="text/html" pageEncoding="UTF-8"%>
        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
           "http://www.w3.org/TR/html4/loose.dtd">
        <html>
            <head>
                <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
                <title>Login Page</title>
                <link rel="stylesheet" type="text/css"
href="newcss.css" />
                <link rel="shortcut icon" href="favicon.ico" >
            </head>
          <body >
              <div style="text-align:center;font-size: 14px;">
          <img src="irmoslogo.gif" width="286" height="214"
alt="irmoslogo"/>
          <form name="login" method="POST" action='<%=
response.encodeURL("j_security_check") %>' >
             <table border="0" cellspacing="5" style="margin-left:auto;
margin-right:auto;font-family: 'Segoe UI';">
            <tr>
              <th align="right">User Name:</th>
              <td align="left"><input type="text"
name="j_username"></td>
            </tr>
            <tr>
              <th align="right">Password:</th>
              <td align="left"><input type="password"
name="j_password"></td>
            </tr>
            <tr>
                <td></td>
              <td align="left"><input type="submit" value="Login"></td>
              <!--<td align="right"><input type="reset"></td>-->
            </tr>
            </table>
```

```
            </form>
              </div>
             <jsp:include page="footer.jsp" flush="true"/>
            </body>
        </html>
```

Κώδικας 10.2 Logout.jsp

```
        <%--
            Document   : Logout
            Created on : 19 Φεβ 2010, 6:02:36 μμ
            Author     : Antreas
        --%>

        <%@page contentType="text/html" pageEncoding="UTF-8"%>
        <%@ page session="true"%>

        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
           "http://www.w3.org/TR/html4/loose.dtd">

        <html>
            <head>
                <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
                <title>Logout</title>
                <link rel="shortcut icon" href="favicon.ico" >
                 <link rel="stylesheet" type="text/css"
href="newcss.css" />
            </head>
            <body>
               <div align="center"><img src="irmoslogo.gif" width="286"
height="214" alt="irmoslogo"/><br>
                <span style="color:red;font-weight:bold">
                 User '<%=request.getRemoteUser()%>' has been logged
out.
                 </span><br>
                 <% session.invalidate(); %>
                <br/><br/>
                 <a href="index.jsp">Click here to go to index</a>
                 </div>
            </body>
        </html>
```

Κώδικας 10.3 Index.jsp

```
        <%--
            Document   : index
            Created on : 15 Ιουλ 2009, 3:08:03 μμ
            Author     : Antreas
        --%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>Irmos Portal Home Page</title>
        <link rel="stylesheet" type="text/css"
href="newcss.css" />
        <link rel="shortcut icon" href="favicon.ico" >
    </head>
    <body>
     <div style="color:RED">
        <%if
(request.getParameter("monitoring")!=null){%>Monitoring
        <%}else{%>Home Page<%}%> </div>
         <div style="text-align:right"> <img
src="irmoslogo.gif" width="286" height="214" alt="irmoslogo"/>
      </div>

    <TABLE WIDTH=100% CELLSPACING=0 CELLPADDING=0>
          <TR><TD WIDTH="30%">
                  <jsp:include page="navbar.html"
flush="true"/>
        </TD>
        <td>
            <%if (request.getParameter("monitoring")!=null){%>
            <jsp:include page="monitoring.jsp" flush="true" />
            <%}%>
            <%if (request.getParameter("Faq")!=null){%>
              <jsp:include page="Manual.txt" flush="true" />
            <%}%>
        </td>
    </TABLE>
     <jsp:include page="footer.jsp" flush="true"/>
    </body>
</html>
```

Κώδικας 10.4 Navbar.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <link rel="stylesheet" type="text/css" href="newcss.css" />
    <link rel="shortcut icon" href="favicon.ico" >
  </head>
  <body>
      <div >
```

```
            <table border="0" cellpadding="0">
                <tbody>
                    <tr>
                        <td><a href="index.jsp">Home Page</a></td>
                    </tr>
                    <tr>
                        <td>Account Details</td>
                    </tr>
                    <tr>
                        <td><a
href="negotiate1.jsp?phase=1">Negotiate Phase</a></td>
                    </tr>
                    <tr>
                        <td>Execution Phase</td>
                    </tr>
                    <tr>
                        <td><a href="index.jsp?monitoring">Monitoring
Phase</a></td>
                    </tr>
                    <tr>
                        <!--<td><a href="index.jsp?Faq">Faq</a></td>-
->
                        <td>Faq</td>
                    </tr>
                </tbody>
            </table>
            </div>
        </body>
    </html>
```

Κώδικας 10.5 Error.jsp

```
    <%--
        Document   : Loginerror
        Created on : 16 Ιουλ 2009, 11:29:41 μμ
        Author     : Antreas
    --%>

    <%@page isErrorPage="true"contentType="text/html"
pageEncoding="UTF-8"%>
    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
       "http://www.w3.org/TR/html4/loose.dtd">

    <html>
        <head>
            <title>Login Test: Error logging in</title>
            <link rel="shortcut icon" href="favicon.ico" >
            <link rel="stylesheet" type="text/css"
href="newcss.css" />
        </head>
        <body bgcolor="white">
            <h1>Error Logging In</h1>
            <br/>
        </body>
```

```
</html>
```

Κώδικας 10.6 newcss.css

```
root {
    display: block;
}

        body{
        font-family:'Segoe UI',Calibri, fantasy;
        font-size: 14px;
        height:100%;
        }

        input[type="submit"],input[type="reset"]
        {
        font-size:18px;
        /*font-weight:lighter;*/
        font-family:'Segoe UI',Calibri, fantasy;
        /*background: #969696 ;*/
        color: #eeeeee;
        border: 1px solid #000000;
        background-color: #969696;
        background-repeat: repeat;
        background-image: url(templatemo_footer.jpg);
}

        select {
             font-size:14px;
        }

        label {
        /*display: block;*/
        /*float: left;*/
        margin-right: 0em;
        text-align: right;
        width: 12em;
        }
```

Κώδικας 10.7 footer.jsp

```
<%--
    Document   : footer
    Created on : 19 Σεπ 2009, 6:33:17 πμ
    Author     : Antreas
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
      <div style="position:fixed;bottom:0;left:0;border:1px solid
black; height:2em;clear:both;font-size:14px;background:
url(templatemo_footer.jpg) repeat-x;width: 100%;">
          <div style="color: #ccc;position:fixed;bottom:0;left:0;
width:20%"><a style="color: #ccc;"
href="mailto:irmos@telecom.ntua.gr">Contact</a></div>
        <div style="color:
#ccc;position:fixed;bottom:0;right:0;">Copyright of IRMOS Consortium
v0.40</div>
      </div>
```

Κώδικας 10.8 Negotiate1.jsp

```
      <%--
          Document   : negotiate1
          Author     : Antreas
      --%>

      <%@page contentType="text/html" pageEncoding="UTF-8"
errorPage="negerror.jsp"%>

      <%@ page import="org.globus.axis.message.addressing.Address,
org.globus.axis.message.addressing.EndpointReferenceType" %>
      <%@ page
import="ntua.irmos.wp5.stubs.singleton.IrmosPortal.service.IrmosPort
alServiceAddressingLocator,
ntua.irmos.wp5.stubs.singleton.IrmosPortal.IrmosPortalPortType,
ntua.irmos.wp5.stubs.singleton.IrmosPortal.Requestaslatemplate,ntua.
irmos.wp5.stubs.singleton.IrmosPortal.RequestaslatemplateResponse"
%>
      <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
         "http://www.w3.org/TR/html4/loose.dtd">

      <html>
        <head>
           <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
           <title>Negotiate Phase</title>
           <link rel="stylesheet" type="text/css" href="newcss.css"
/>
           <link rel="shortcut icon" href="favicon.ico" >
        </head>
        <%  int phase = 1;
           if (request.getParameter("phase")!=null)
            phase = Integer.valueOf(request.getParameter("phase"));

            if (phase==1){
            session.removeAttribute("reqsla");
            session.removeAttribute("service");
            session.removeAttribute("slatemplate");
            session.removeAttribute("prtl");
            }
           %>
```

```
        <jsp:useBean id="slatemplate" scope="request"
class="ntua.irmos.wp5.stubs.singleton.IrmosPortal.RequestaslatemplateResponse"/>
        <jsp:useBean id="reqsla" scope="session"
class="ntua.irmos.wp5.stubs.singleton.IrmosPortal.Requestaslatemplate" />
        <jsp:setProperty name="reqsla" property="*" />
        <jsp:useBean id="service" scope="session"
class="irmosgui.serviceURI" />
        <jsp:setProperty name="service" property="*" />
        <%
           String serviceURI =
"http://192.108.38.51:8080/wsrf/services/irmos/wp5/singleton/IrmosPortalService";
              if (phase == 3){
                  IrmosPortalServiceAddressingLocator locator = new
IrmosPortalServiceAddressingLocator();
                // try {
                  if (service.getServiceURI()!=null)
                        serviceURI=service.getServiceURI();
                             // Creat be endpoint reference to
service
                             EndpointReferenceType endpoint = new
EndpointReferenceType();
                             endpoint.setAddress(new
Address(serviceURI));


service.setPrtl(locator.getIrmosPortalPortTypePort(endpoint));

                             // Get PortType

                             slatemplate =
service.getPrtl().requestaslatemplate(reqsla);

//****************************************getPrtl************

service.setASLATemplateEPR(slatemplate.getAslaTemplatesEpr());

service.setASLATemplateType(slatemplate.getLabel());

//*****************************************************
                  }  %>
       <body bgcolor="white">
           <div style="clear:both;">
           <span style="color:RED;text-
align:left;float:left;width:10%"> Negotiate Phase </span>
      <%--     <div style="float:left;width:90%;text-align:center">
               <form name="service"
action="negotiate1.jsp?phase=<%=phase%>" method ="post">
               <input type="text" name="serviceURI"
value="<%=service.getServiceURI()%>" style="font-
size:12px;color:RED" size="100"/>
               <input style="font-size:12px;color:lime" type="submit"
value="Set" name="set" />
             </form>
               </div>--%>
```

```
            </div>
            <div style="clear:both;text-align:right"> <img
src="irmoslogo.gif" width="286" height="214" alt="irmoslogo"/></div>
             <div style="width:25em;text-align:right">
                 <form name="negid" action="<%if (phase==1){ %>
negotiate1.jsp?phase=2 <%}else if
(phase==2){%>negotiate1.jsp?phase=3<%}else{%>negotiate2.jsp<%}%>"
method="POST">
                      <ul style="border:0; padding:0; list-
style:none;margin-left:auto;margin-right:auto">
                          <li>
                              <label style="<%if (phase!=1)
{%>color:#C0C0C0<%}%>">Insert your Customer Id:</label>
                              <input style="<%if (phase!=1)
{%>color:#C0C0C0<%}%>" type="text" name="costumerID"
value="<%=reqsla.getCostumerID()%>"/>
                          </li>
                             <%if (phase == 1){%>
                             <li>
                              <label> </label>
                              <input type="submit" value="Proceed"/>
                             </li>
                          <%}%>
                             <%if (phase!=1) {%>
                          <li>
                              <label style="<%if (phase!=2)
{%>color:#C0C0C0<%}%>">Select Application:</label>
                              <input style="<%if (phase!=2)
{%>color:#C0C0C0<%}%>" type="text" name="applicationID" value="<% if
(reqsla.getApplicationID()!=null){
out.println(reqsla.getApplicationID());}%>" />
                             </li>
                            <%if (phase == 2){%>
                          <li>
                              <label> </label> <input type="submit"
value="RequestSla"/>
                         </li>
                          <%}%>
                             <%if (phase!=2) {%>
                          <li>
                              <label>Select ASLA:</label>
                              <select STYLE="width: 10.5em" name="Epr">
                               <%
                                for (int
i=0;i<service.getASLATemplateType().length;i++){
                                    String s =
service.getASLATemplateType()[i];
                                    out.println("<option
value=\""+Integer.toString(i)+"\">"+ s +"</option>");
                                }
                                %>
                              </select>
                              </li>
                              <%if (phase==3) {%>
                          <li>
                              <label> </label>
                              <input type="submit" value="EditSla"/>
```

```
                        </li>
                            <%}%>
                            <%}%>
                            <%}%>
                    </ul>
            </form>
            </div>
               <jsp:include page="footer.jsp" flush="true"/>
      </body>
    </html>
```

Κώδικας 10.9 Negotiate2.jsp

```
    <%--
        Document   : negotiate2
        Author     : Antreas
    --%>

    <%@page contentType="text/html" pageEncoding="UTF-8"
errorPage="negerror.jsp"%>
    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
       "http://www.w3.org/TR/html4/loose.dtd">
    <%@page import="org.w3c.dom.Node, org.w3c.dom.Element,
org.w3c.dom.Document, org.w3c.dom.NodeList,org.w3c.dom.NamedNodeMap,
javax.xml.parsers.DocumentBuilder,
javax.xml.parsers.DocumentBuilderFactory,org.globus.axis.message.add
ressing.EndpointReferenceType" %>
    <%@page import="java.util.* , irmosgui.Component
,irmosgui.Parameter, java.util.ArrayList"%>
    <%@page import="java.io.StringReader,org.xml.sax.InputSource,
java.io.*,org.globus.axis.message.addressing.EndpointReferenceType"%
>
    <%
        if (session.getAttribute("Comp")!=null)
            session.removeAttribute("Comp");
    %>
    <jsp:useBean id="Comp" scope="session"
class="irmosgui.ComponentBean" />
    <jsp:useBean id="reqsla" scope="session"
class="ntua.irmos.wp5.stubs.singleton.IrmosPortal.Requestaslatemplat
e" />
    <jsp:useBean id="service" scope="session"
class="irmosgui.serviceURI" />
    <html>
        <head>
            <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
            <title>Negotiate Phase</title>
            <link rel="stylesheet" type="text/css"
href="newcss.css" />
            <link rel="shortcut icon" href="favicon.ico" >
        </head>

           <body>
               <% int index =
Integer.valueOf(request.getParameter("Epr"));
```

```
                    EndpointReferenceType s =
service.getASLATemplateEPR()[index];
                    service.setIndex(index);
       String datafile =
s.getParameters().get_any()[0].getValue().toString();
                %>
            <div style="margin: 0 auto 4em;">
           <div style="color:RED"> Negotiate Phase </div>
                <div style="text-align:right"> <img
src="irmoslogo.gif" width="286" height="214" alt="irmoslogo"/>
            </div>
           <div>
           <span style="color:#C0C0C0;font-weight:bold">Selected
Application:</span><span style="color:#C0C0C0;font-
weight:bold;margin-
left:2em"><%=reqsla.getApplicationID()%></span><br>
           <br>
           <span style="color:#C0C0C0;font-weight:bold">Selected
Template:</span>
                <form style="margin-left:15em" name="form1"
action="process.jsp" method="POST">
                <%
                     FileInputStream fis = null;
                     /* We will store the RPs in these variables */
                     String ASLATemplate;
                             /* Open the file */

System.out.println((service.getDatadir())+datafile+".data");

                             fis = new
FileInputStream((service.getDatadir())+datafile+".data");
                             ObjectInputStream ois = new
ObjectInputStream(fis);

                             System.out.println("ASLAResource->Just
before");

                             /* Read the RPs */
                             ASLATemplate = ois.readUTF();
                             System.out.println(ASLATemplate);
                             /* Make sure we clean up, whether the
load succeeds or not */
                             if (fis != null) {
                                fis.close();
                             }

                    DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
                    DocumentBuilder        builder =
factory.newDocumentBuilder();
                    Document doc = builder.parse(new InputSource(new
StringReader(ASLATemplate)));
                    Element root = doc.getDocumentElement();
                     Comp.doc = doc;
         NodeList app = root.getElementsByTagName("app:components");

                      for (int i=0;i<app.getLength();i++ )
```

```
                        {
                            Node n = app.item(i);
                            Component C = new Component(n);
                            Comp.CompList.add(C);

                        }%>
                        <% for (Component C : Comp.CompList) {%>
                        <fieldset style="float:left;width:410px">
                            <legend style="font-
weight:bold"><%=C.name%></legend>
                        <ul style="border:0; margin:0; padding:0; list-
style:none;">
                           <% for (Parameter P : C.Pl ) {%>
                            <li style="clear:both;list-
style:none;padding-bottom:10px;"> <div
style="width:140px;float:left;"><%=P.name %></div> <%=P.toHTML() %>
</li>
                            <% } %>
                        </ul>
                        </fieldset>
                       <% } %>
                       <div style="clear:both;">
                        <input style="margin-left:20em" type="submit"
value="Submit" name="submit"/>
                       </div>
                    </form>
                </div>
                 </div>
                <jsp:include page="footer.jsp" flush="true"/>
               </body>
        </html>
```

Κώδικας 10.10 process.jsp

```
        <%--
            Document   : process.jsp
            Author     : Antreas
        --%>

        <%@page contentType="text/html" pageEncoding="UTF-8"
errorPage="negerror.jsp"%>
        <%@page import="org.w3c.dom.Node, org.w3c.dom.Element,
org.w3c.dom.Document, org.w3c.dom.NodeList,org.w3c.dom.NamedNodeMap,
javax.xml.parsers.DocumentBuilder,
javax.xml.parsers.DocumentBuilderFactory" %>
        <%@page import="java.util.* , java.util.ArrayList ,
irmosgui.Component ,irmosgui.Parameter, java.util.ArrayList"%>
        <%@page import =
"ntua.irmos.wp5.stubs.singleton.IrmosPortal.Negotiation"%>

        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
        "http://www.w3.org/TR/html4/loose.dtd">

      <jsp:useBean id="Comp" scope="session"
class="irmosgui.ComponentBean" />
      <jsp:useBean id="reqsla" scope="session"
class="ntua.irmos.wp5.stubs.singleton.IrmosPortal.Requestaslatemplat
e" />
      <jsp:useBean id="service" scope="session"
class="irmosgui.serviceURI" />
      <jsp:useBean id="Exec" scope="session"
class="irmosgui.ExecBean" />
      <%
                    Enumeration keys = request.getParameterNames();
                    while (keys.hasMoreElements() )
                     {
                        String key = (String)keys.nextElement();
                        //out.println(key);
                        int index = key.indexOf(":");
                        if (index == -1)
                            continue;
                        String CompName = key.substring(0, index) ;
                        String ParameterName =
key.substring(index).substring(1);

                         //To retrieve a single value
                        String value = request.getParameter(key);
      Comp.getComponentByName(CompName).getParameterByName(ParameterN
ame).setValue(value);
           }%>
      <html>
          <head>
              <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
              <title>Negotiation Phase</title>
              <link rel="stylesheet" type="text/css"
href="newcss.css" />
              <link rel="shortcut icon" href="favicon.ico" >
          </head>
          <body>
           <div style="margin: 0 auto 4em;">
           <div style="color:RED"> Negotiate Phase </div>
                <div style="text-align:right"> <img
src="irmoslogo.gif" width="286" height="214" alt="irmoslogo"/>
            </div>
           <div align="center">
              <%
              ArrayList <String> aslaOfferLabel = new
ArrayList<String>();
              ArrayList <String> aslaOfferValue = new
ArrayList<String>() ;
              int i=-1;
                   for (Component c: Comp.getCompList())
                       for (Parameter p : c.getPl()){
                           if (p.getValue() != null)
                               {i++;
                                aslaOfferLabel.add( p.name);
                                aslaOfferValue.add( p.getValue());
```

```
                                    }
                                }
                                 String [] Label = new
String[aslaOfferLabel.size()];
                                 String [] Value = new
String[aslaOfferValue.size()];
                                aslaOfferLabel.toArray(Label);
                                aslaOfferValue.toArray(Value);
                          Negotiation neg = new
Negotiation(Integer.toString(reqsla.getCostumerID()),(service.getASL
ATemplateEPR()[service.getIndex()]),Label,Value);
                          String ASLAEPR = service.getPrtl().negotiation(neg);
                          Exec.setASLA_EPR(ASLAEPR);
                         %>
                         <p style="font-size:18px;">Negotiation Succesfull with
return value</p>
                         <textarea rows="10" cols="80"
readonly="readonly"><%=Exec.getASLA_EPR()%></textarea>
                         <form action="ReservationInit.jsp">
                         <div style="clear:both;">
                           <input  type="submit" value="Reserve"
name="reserve"/>
                         </div>
                        </form>
                     </div>
                      </div>
                     <jsp:include page="footer.jsp" flush="true"/>
                    </body>
                </html>
```

Κώδικας 10.11 negerror.jsp

```
        <%--
            Document   : negerror
                Author     : Antreas
        --%>

        <%@page isErrorPage="true" import="java.util.*"
contentType="text/html" pageEncoding="UTF-8"%>
        <%@page import = "java.io.StringWriter,java.io.PrintWriter"%>
        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
           "http://www.w3.org/TR/html4/loose.dtd">
        <jsp:useBean id="reqsla" scope="session"
class="ntua.irmos.wp5.stubs.singleton.IrmosPortal.Requestaslatemplat
e" />
        <html>
            <head>
                <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
                <title>Negotiation Error</title>
                <link rel="stylesheet" type="text/css"
href="newcss.css" />
                <link rel="shortcut icon" href="favicon.ico" >
            </head>
            <body bgcolor="white">
```

```
                <div align="center"><img src="irmoslogo.gif"
width="286" height="214" alt="irmoslogo"/><br>
                <span style="color:red;font-weight:bold">Error
Negotiating</span><br>
                    <span style="font-weight:bold">OOps! An exception
occured</span><br>
                <%=exception.toString()%><br>
                <textarea wrap="off" name="StackTrace" rows="8"
cols="80" readonly="readonly">
                <%
                StringWriter sw = new StringWriter();
                PrintWriter pw = new PrintWriter(sw);
                exception.printStackTrace(pw);
                out.print(sw);
                sw.close();
                pw.close();
                %>
                </textarea>
              <form name="negid" action="negotiate1.jsp?phase=1"
method="POST">
                <input type="submit" value="Return to Negotiation"
name="ret" />
             </form>
            </div>
            </body>
        </html>
```

Κώδικας 10.12 ReservationInit.jsp

```
        <%--
            Document   : executioninit
            Author     : Antreas
        --%>

        <%@page contentType="text/html" pageEncoding="UTF-8"
errorPage="Reservationerror.jsp"%>
        <%@page
import="java.io.BufferedReader,java.io.FileReader,java.io.InputStrea
mReader"%>
        <%@page import="org.w3c.dom.Node, org.w3c.dom.Element,
org.w3c.dom.Document, org.w3c.dom.NodeList,org.w3c.dom.NamedNodeMap,
javax.xml.parsers.DocumentBuilder,
javax.xml.parsers.DocumentBuilderFactory" %>
        <%@page import="java.util.* , irmosgui.Component
,irmosgui.Parameter, java.util.ArrayList"%>
        <%@page import="java.io.StringReader,org.xml.sax.InputSource"%>
        <%@ page import="org.globus.axis.message.addressing.Address,
org.globus.axis.message.addressing.EndpointReferenceType" %>
        <%@ page
import="ntua.irmos.wp5.stubs.singleton.IrmosPortal.service.IrmosPort
alServiceAddressingLocator,
ntua.irmos.wp5.stubs.singleton.IrmosPortal.IrmosPortalPortType,
ntua.irmos.wp5.stubs.singleton.IrmosPortal.Requestaslatemplate,ntua.
```

```
irmos.wp5.stubs.singleton.IrmosPortal.RequestaslatemplateResponse"
%>
      <%@ page import="java.io.IOException, java.io.*" %>


      <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
      "http://www.w3.org/TR/html4/loose.dtd">

      <jsp:useBean id="Exec" scope="session"
class="irmosgui.ExecBean" />
      <jsp:useBean id="service" scope="session"
class="irmosgui.serviceURI" />
      <jsp:useBean id="slatemplate" scope="request"
class="ntua.irmos.wp5.stubs.singleton.IrmosPortal.RequestaslatemplateResponse"/>
      <jsp:useBean id="reqsla" scope="session"
class="ntua.irmos.wp5.stubs.singleton.IrmosPortal.Requestaslatemplate" />
      <jsp:setProperty name="reqsla" property="*" />
      <jsp:setProperty name="service" property="*" />
      <%
      if (request.getParameter("download")!=null)
              {
              ServletOutputStream myOut=null;
              try{
                  myOut = response.getOutputStream( );
                  response.setContentType("text/plain");
                  response.addHeader("Content-
Disposition","attachment; filename="+"ConfigInfo.txt");
                  response.setContentLength( (int)
Exec.getASLA_EPR().toString().length() );
                  myOut.print(Exec.getASLA_EPR().toString());
              } catch (IOException ioe){
                  throw new ServletException(ioe.getMessage( ));
                   } finally {
                       if (myOut != null)
                           myOut.close( );
                   }
                  }
          %>
      <html>
          <head>
              <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
              <title>Reservation Init</title>
               <link rel="stylesheet" type="text/css"
href="newcss.css" />
               <link rel="shortcut icon" href="favicon.ico">
          </head>
          <body>
            <div style="margin: 0 auto 3em;">
            <div style="color:RED">Reservation Phase</div>
            <div style="clear:both;text-align:right"> <img
src="irmoslogo.gif" width="286" height="214" alt="irmoslogo"/></div>
            <div style="text-align:left">
              <p>
              <%
```

```
                    String ConfigInfo =
service.getPrtl().reservation(Exec.getASLA_EPR());
                    if (ConfigInfo==null){
                        throw new Exception("Config Info is null");
                        }
                   %>
               <div style="margin-left:42%">
                <span style="font-size:18px;">Config Info:</span>
              </div>
                <div align="center">
                 <textarea name="ASLA" rows="26" cols="80"
readonly="readonly"><%out.print(ConfigInfo);%>
                 </textarea>
                 <a href="ReservationInit.jsp?download">Download
file</a>
                 </div>
                 <form action="Reservation2.jsp" >
                    <input style="margin-left:45%" type="submit"
value="Configure" name="Configure" />
                 </form>
                </div>
                </div>
              <jsp:include page="footer.jsp" flush="true"/>
           </body>
       </html>
```

Κώδικας 10.13 Reservation2.jsp

```
       <%--
           Document   : execution2
           Created on : Dec 18, 2009, 9:33:48 PM
           Author     : Administrator
       --%>

       <%@page contentType="text/html" pageEncoding="UTF-8"
errorPage="Reservationerror.jsp"%>
       <%@page import="java.util.ArrayList" %>
       <jsp:useBean id="Exec" scope="session"
class="irmosgui.ExecBean" />
       <jsp:useBean id="service" scope="page"
class="irmosgui.serviceURI" />
       <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
          "http://www.w3.org/TR/html4/loose.dtd">

       <html>
            <head>
               <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
               <title>Reservation Page</title>
                <link rel="stylesheet" type="text/css"
href="newcss.css" />
```

```
                <link rel="shortcut icon" href="favicon.ico">
            </head>
            <body>
              <div style="margin: 0 auto 3em;">
              <div style="color:RED">Reservation Phase</div>
              <div style="clear:both;text-align:right"> <img
src="irmoslogo.gif" width="286" height="214" alt="irmoslogo"/></div>
              <div style="text-align:left">
                <%
                service.getPrtl().configuration(Exec.getASLA_EPR());
                %>
                 <form action="Execute.jsp" >
                    <input style="margin-left:45%" type="submit"
value="Execute" name="Execute" />
                 </form>
              </div>
              </div>
                <jsp:include page="footer.jsp" flush="true"/>
            </body>
        </html>
```

Κώδικας 10.14 ReservationError.jsp

```
        <%--
            Document   : Reservationerror
            Author     : Antreas
        --%>

        <%@page isErrorPage="true" import="java.util.*"
contentType="text/html" pageEncoding="UTF-8"%>
        <%@page import = "java.io.StringWriter,java.io.PrintWriter"%>
        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
           "http://www.w3.org/TR/html4/loose.dtd">

        <html>
            <head>
                <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
                <title>Reservation Error</title>
                 <link rel="shortcut icon" href="favicon.ico" >
                 <link rel="stylesheet" type="text/css"
href="newcss.css" />
            </head>
           <body bgcolor="white">
                <div align="center"><img src="irmoslogo.gif"
width="286" height="214" alt="irmoslogo"/><br>
                <span style="color:red;font-weight:bold">Error
Monitoring</span><br>
                    <span style="font-weight:bold">OOps! An exception
occured</span><br>
                <%=exception.toString()%><br>
                <textarea wrap="off" name="StackTrace" rows="8"
cols="80" readonly="readonly">
                <%
```

```
                StringWriter sw = new StringWriter();
                PrintWriter pw = new PrintWriter(sw);
                exception.printStackTrace(pw);
                out.print(sw);
                sw.close();
                pw.close();
                %>
                </textarea>
              <form name="negid" action="index.jsp" method="POST">
                <input type="submit" value="Return to Index" name="ret"
/>
             </form>
            </div>
            </body>
        </html>
```

Κώδικας 10.15 Monitoring.jsp

```
        <%--
            Document   : monitoring
            Author     : Antreas
        --%>

        <%@page contentType="text/html" pageEncoding="UTF-8"
errorPage="monerror.jsp"%>
        <%@page import="java.io.StringReader,org.xml.sax.InputSource,
java.io.*,org.globus.axis.message.addressing.EndpointReferenceType"%
>

        <%@ taglib uri="http://java.sun.com/jsp/jstl/xml"  prefix="x"
%>
        <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"
%>
        <%@ taglib uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
        <jsp:useBean id="service" scope="session"
class="irmosgui.serviceURI" />
        <jsp:setProperty name="service" property="mdsURI" />
        <html>
          <head>
            <title>Monitoring</title>
            <meta http-equiv="refresh" content="15">
            <link rel="shortcut icon" href="favicon.ico" >
            <link rel="stylesheet" type="text/css" href="newcss.css" />
          </head>
          <body>
                  <c:import url="<%=service.getMdsURI()%>"
var="inputDoc" />
                  <x:parse doc="${inputDoc}" var="DOC" />
                  <%String Color= "#2469b4";
                    boolean  CheckisOn=false;
                  %>
                   <x:choose >
```

```
                        <x:when select="$DOC//MonitoringData"/>
                       <x:otherwise><div
style="clear:both;padding:5em;font-style:italic;color:RED;text-
align:center">No Monitoring Data Available</div></x:otherwise>
                    </x:choose>
                <div style="text-align: left;">
                    <x:forEach var="Data"
select="$DOC//MonitoringData">
                        <% CheckisOn = false; %>
                    <x:set var="vsnid" select="string($Data/VSN_ID)"/>
                    <x:set var="ascid" select="string($Data/ASC_ID)"/>
                    <c:set var="split" value="${fn:split(vsnid, '/')}"
/>
                    <c:set var="datafile"
value="${split[fn:length(split)-1]}"/>
                    <%   String uri =  service.getDatadir();

uri+=(String)pageContext.getAttribute("datafile");
                        uri+=".data";
                        FileInputStream fis = null;
                       /* We will store the RPs in these variables */
                       String ASLATemplate;
                               /* Open the file */
      try{
                               fis = new FileInputStream(uri);
                               ObjectInputStream ois = new
ObjectInputStream(fis);

                               /* Read the RPs */
                               ASLATemplate = ois.readUTF();
                               System.out.println(ASLATemplate);
                               /* Make sure we clean up, whether the
load succeeds or not */
                               if (fis != null) {
                                  fis.close();
                               }
                               StringReader s = new
StringReader(ASLATemplate);
                               CheckisOn = true;
      %>

               <x:parse doc="${s}" var="sla1"/>
              <%   }catch(Exception e){CheckisOn = false;}
                %>
                    <table
style="float:left;padding:5px;margin:5px;border: 1px solid gray;">
                    <thead>
                        <tr style="color:gray;">
                            <th style="text-align:center;"
colspan="2"><c:out value="${vsnid}"/></th>
                        </tr>
                        <tr style="color:gray;">
                            <th style="text-align:center;"><c:out
value="${ascid}"/></th>
                            <th style="text-align:center;"><x:out
select="$Data/timestamp"/></th>
                        </tr>
```

```
                        <tr>
                            <th style="text-
align:center;">Parameter</th>
                            <th style="text-align:center;">Value</th>
                        </tr>
                    </thead>
                    <tbody>
                        <x:forEach var="MonitoringParameter"
select="$Data/MonitoringParameter" >
                            <% Color = "#2469b4";%>
                            <x:set var="Name"
select="string($MonitoringParameter/Name)"/>
                            <x:set var="Value"
select="string($MonitoringParameter/Value)"/>
                            <% if (CheckisOn){%>
                            <x:set var="defaultvalue"
select="string($sla1//*[local-name()='components' and
@id=$ascid]/parameters[@name=$Name]/*[local-
name()='BasicValue']/@value)"/>
                            <%   }%>
                        <tr>
                            <c:if test="${Value!=defaultvalue &&
!empty defaultvalue}">
                                <% Color = "RED"; %>
                            </c:if>
                        <td style="font-size: 14px;padding:
2px;background-color: <%=Color%>;color: #FFFFFF;">
                            <c:out value="${Name}"/>
                            </td>
                            <td style="text-align:center;font-
size: 12px;color: #000000;background-color:<% if
(Color.equals("RED")){out.print(Color);}%>">
                            <c:out value="${Value}"/>
                            <c:if test="${!empty defaultvalue}">
                            (<c:out value="${defaultvalue}"/>)
                            </c:if>
                        </td>
                    </tr>
                    </x:forEach>
                    <% if (CheckisOn){ %>
                    <tr><td colspan="2"><div style="font-
style:italic;color:green;text-align:center">Data CrossCheck is
On</div></td> </tr>
                    <%}else{%>
                    <tr><td colspan="2"><div style="font-
style:italic;color:red;text-align:center">Data CrossCheck is
Off</div></td> </tr>
                    <%}%>
                </tbody>
            </table>
        </x:forEach>
        </div>
    </body>
</html>
```

Κώδικας 10.16 Monerror.jsp

```
<%--
    Document   : monerror
    Author     : Antreas
--%>

<%@page isErrorPage="true" import="java.util.*"
contentType="text/html" pageEncoding="UTF-8"%>
<%@page import = "java.io.StringWriter,java.io.PrintWriter"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
        <title>Monitoring Error</title>
        <link rel="shortcut icon" href="favicon.ico" >
        <link rel="stylesheet" type="text/css"
href="newcss.css" />
    </head>
     <body bgcolor="white">
        <div align="center"><img src="irmoslogo.gif"
width="286" height="214" alt="irmoslogo"/><br>
        <span style="color:red;font-weight:bold">Error
Monitoring</span><br>
            <span style="font-weight:bold">OOps! An exception
occured</span><br>
        <%=exception.toString()%><br>
        <textarea wrap="off" name="StackTrace" rows="8"
cols="80" readonly="readonly">
        <%
        StringWriter sw = new StringWriter();
        PrintWriter pw = new PrintWriter(sw);
        exception.printStackTrace(pw);
        out.print(sw);
        sw.close();
        pw.close();
        %>
        </textarea>
       <form name="negid" action="index.jsp" method="POST">
        <input type="submit" value="Return to Index" name="ret"
/>
       </form>
    </div>
    </body>
</html>
```

Κώδικας 10.17 irmosgui\Component.java

```
      package irmosgui;

      import java.util.ArrayList;
      import org.w3c.dom.Node;

      /**
       *
       * @author Antreas
       */
      public class Component {

           public Component(Node n){
              name =
n.getAttributes().getNamedItem("id").getNodeValue();
              for(Node
childNode=n.getFirstChild();childNode!=null;childNode=childNode.getN
extSibling())
              {
               if (childNode.getNodeType()== Node.ELEMENT_NODE &&
childNode.getNodeName().equals("parameters")){
                        Parameter p = new Parameter(childNode,this);
                        Pl.add(p);
                }
              }

          }

          public ArrayList<Parameter> Pl = new
ArrayList<Parameter>();

          public ArrayList<Parameter> getPl() {
              return Pl;
          }
           public String name;
           public String toHTML(){
               String  s= "";
               s = "<ul>";
               for (Parameter P : Pl)
               {
                   s += "<li>"+"<div
style=\"width:30px;\">"+P.thisNode.getAttributes().getNamedItem("nam
e").getNodeValue()+"</div>"+P.toHTML()+"</li>\n";
               }
               s += "</ul>";
               return s;
           }
           public Parameter getParameterByName(String s)
           {
               for (Parameter p: Pl)
                   if
(p.thisNode.getAttributes().getNamedItem("name").getNodeValue().equa
ls(s))
                        return p;
               return null;
           }

      }
```

Κώδικας 10.18 irmosgui\ComponentBean.java

```
package irmosgui;

import java.util.ArrayList;
import org.w3c.dom.Document;

/**
 *
 * @author Antreas
 */
public class ComponentBean {
  public Document doc;

  public ArrayList<Component> CompList = new
ArrayList<Component>();//Used for Negotiation

    public ArrayList<Component> getCompList() {
        return CompList;
    }

  public Component getComponentByName(String s)
  {
      for (Component c: CompList)
         if(c.name.equals(s))
             return c;
      return null;
  }
}
```

Κώδικας 10.19 irmosgui\ExecBean.java

```
package irmosgui;


import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;


/**
 *
 * @author Antreas
 */
public class ExecBean {
    public String ASLA_EPR;
    public ArrayList<ArrayList<String>> ClientSettings = new
ArrayList<ArrayList<String>>();//Used for execution

    public String getASLA_EPR() {
        return ASLA_EPR;
```

```
        }

        public void setASLA_EPR(String ASLA_EPR) {
            this.ASLA_EPR = ASLA_EPR;
        }

        public ExecBean() {

        }
        public void process(HttpServletRequest req ){
            int i=0;
            if (req.getParameter("Add")!=null){
               for (i=0;i<ClientSettings.size();i++)
               {
                   String name=this.ClientSettings.get(i).get(0);
                   if (name.equals(req.getParameter("Add"))){
                       this.ClientSettings.get(i).add("");
                        break;
                   }
               }
            }
            for (i=0;i<ClientSettings.size();i++){
                for (int j=1;j<ClientSettings.get(i).size();j++){
                 this.ClientSettings.get(i).remove(j);

this.ClientSettings.get(i).add(j,req.getParameter(this.ClientSetting
s.get(i).get(0) + Integer.toString(j) ));
               }
            }

        }

    }
```

Κώδικας 10.20 irmosgui\Parameter.java

```
    package irmosgui;


    import java.util.Iterator;
    import java.util.Vector;
    import org.w3c.dom.Node;


    /**
     *
     * @author Antreas
     */

    public class Parameter {
      public  Parameter(Node n,Component c){
          thisNode = n;
          Parent = c;
          if (n.hasChildNodes())
            setValuesAllowed();
```

```
            name  =
thisNode.getAttributes().getNamedItem("name").getNodeValue();
        };
        public  Component Parent;
        public  Node thisNode;
        public  String name,defaultvalue,value;

          public String getName() {
              return name;
          }

          public void setName(String name) {
              this.name = name;
          }

          public String getValue() {
              return value;
          }

          public void setValue(String value) {
              this.value = value;
          }

        public  boolean Enabled;
        public  boolean isDropDown=false;
        public  Vector<String> valuesAllowed=new Vector<String>();

        public  void setValuesAllowed(){
             for (Node n =
thisNode.getFirstChild();n!=null;n=n.getNextSibling()){
                 if (n.getNodeName().equals("valuesAllowed")){
                     isDropDown=true;
                     valuesAllowed.add(new
String(n.getAttributes().getNamedItem("value").getNodeValue()));
                 }
                 else if
(n.getNodeName().equals("speclang:BasicValue")){
                     defaultvalue =
n.getAttributes().getNamedItem("value").getNodeValue();
                 }
             }
          }
        public String toHTML(){
            String s;
             if (isDropDown){
                 s="<select name=\"" + Parent.name + ":" +
thisNode.getAttributes().getNamedItem("name").getNodeValue() +"\" >"
;
                for (Iterator<String>
it=valuesAllowed.iterator();it.hasNext(); )
                 {
                     String a=it.next();
                     s+="<option>" + a + "</option>";
                 }
                     s+="</select>";
             }
             else
```

```
            {
               s="<input type=\"text\" name=\""+  Parent.name + ":" +
thisNode.getAttributes().getNamedItem("name").getNodeValue() + "\"
value=\""+defaultvalue+"\"  size=30/>";
            }

            return s;
        }



    }
```

Κώδικας 10.21 irmosgui\ServiceURI.java

```
    package irmosgui;

    import java.io.IOException;
    import java.io.InputStream;
    import java.net.URL;
    import java.util.Properties;
    import java.util.logging.Level;
    import java.util.logging.Logger;

    import
ntua.irmos.wp5.stubs.singleton.IrmosPortal.IrmosPortalPortType;

    /**
     *
     * @author Antreas
     */
    public class serviceURI {
        public String serviceURI; // Selected ServiceURI
        public String mdsURI;
        public String datadir;
        public String datadir1;
        String [] ASLATemplateEPR ;// EPRS retreived
        String [] ASLATemplateType ;// EPRS retreived
        int index;  // Selected Index in the ASLATemplate arrays
        IrmosPortalPortType prtl;


        public String getDatadir1() {
            return datadir1;
        }

        public void setDatadir1(String datadir1) {
            this.datadir1 = datadir1;
        }

        public String getDatadir() {
            return datadir;
        }
```

```
            public void setDatadir(String datadir) {
                this.datadir = datadir;
            }


            public String getMdsURI() {
                return mdsURI;
            }

            public void setMdsURI(String mdsURI) {
                this.mdsURI = mdsURI;
            }


            public IrmosPortalPortType getPrtl() {
                return prtl;
            }

            public void setPrtl(IrmosPortalPortType prtl) {
                this.prtl = prtl;
            }

            public int getIndex() {
                return index;
            }

            public void setIndex(int index) {
                this.index = index;
            }

            public String[] getASLATemplateEPR() {
                return ASLATemplateEPR;
            }

            public void setASLATemplateEPR(String[] ASLATemplateEPR) {
                this.ASLATemplateEPR = ASLATemplateEPR;
            }

            public String[] getASLATemplateType() {
                return ASLATemplateType;
            }

            public void setASLATemplateType(String[] ASLATemplateType)
{
                this.ASLATemplateType = ASLATemplateType;
            }

            public serviceURI() {
                InputStream in = null;
                try {
                    URL myURL =
this.getClass().getResource("conf.properties");
                    in = myURL.openStream();
                    Properties p = new Properties();
                    p.load(in);
                    this.serviceURI = p.getProperty("serviceURI");
```

```
                this.mdsURI = p.getProperty("mdsURI");
                this.datadir = p.getProperty("datadir");
                this.datadir1 = p.getProperty("datadir1");
            } catch (IOException ex) {

Logger.getLogger(serviceURI.class.getName()).log(Level.SEVERE, null,
ex);
            } finally {
                try {
                    in.close();
                } catch (IOException ex) {

Logger.getLogger(serviceURI.class.getName()).log(Level.SEVERE, null,
ex);
                }
            }
           // this.serviceURI =
"http://147.102.19.104:45000/wsrf/services/irmos/wp5/singleton/Irmos
PortalService";
            //this.mdsURI = "webmds.xml";
        }

        public String getServiceURI() {
            return serviceURI;
        }

        public void setServiceURI(String serviceURI) {
            this.serviceURI = serviceURI;
        }
    }
```

Κώδικας 10.22 irmosgui\conf.properties

```
    serviceURI =
http://192.108.38.51:8080/wsrf/services/irmos/wp5/singleton/IrmosPor
talService
    mdsURI = webmds.xml
    datadir = /home/globus/.globus/persisted/192.108.38.54-
8080/ASLAResource/
    datadir1 = /home/globus/.globus/persisted/192.108.38.54-
8080/ASLAResource/
```

Κώδικας 10.23 Παράδειγμα αρχείου Monitoring webmds.xml

```
    <?xml version="1.0" encoding="UTF-8"?>
    <ns0:IndexRP
xmlns:ffs="http://www.ntua.gr/namespaces/irmos/wp5/MonitoringService
" xmlns:glue="http://mds.globus.org/glue/ce/1.1"
xmlns:metric="http://mds.globus.org/metadata/2005/02"
xmlns:ns0="http://mds.globus.org/index"
```

```
xmlns:ns1="http://docs.oasis-open.org/wsrf/sg-2"
xmlns:ns2="http://www.w3.org/2005/08/addressing"
xmlns:ns3="http://www.globus.org/namespaces/2008/03/gram/job"
xmlns:ns4="http://mds.globus.org/inmemoryservicegroup"
xmlns:ns44="http://mds.globus.org/aggregator"
xmlns:ns45="http://mds.globus.org/metrics/2004/09"
xmlns:ns5="http://mds.globus.org/aggregator/types"
xmlns:ns52="http://docs.oasis-open.org/wsn/b-2"
xmlns:ns56="http://docs.oasis-open.org/wsrf/rl-2"
xmlns:ns6="http://www.ntua.gr/namespaces/irmos/wp5/MonitoringService
_instance" xmlns:rft="http://www.globus.org/namespaces/2008/04/rft">
          <ns1:Entry>
              <ns1:ServiceGroupEntryEPR>

<ns2:Address>http://192.108.38.52:8080/wsrf/services/DefaultIndexSer
viceEntry</ns2:Address>
                    <ns3:ReferenceParameters
xmlns:ns3="http://www.w3.org/2005/08/addressing">
                        <ns4:ServiceGroupEntryKey>
                            <ns5:GroupKey>33549034</ns5:GroupKey>
                            <ns6:EntryKey
xmlns:ns6="http://mds.globus.org/aggregator/types">4117493</ns6:Entr
yKey>
                        </ns4:ServiceGroupEntryKey>
                    </ns3:ReferenceParameters>
                </ns1:ServiceGroupEntryEPR>
                <ns1:MemberServiceEPR>
                    <ns7:Address
xmlns:ns7="http://www.w3.org/2005/08/addressing">local:/wsrf/service
s/ReliableFileTransferFactoryService</ns7:Address>
                </ns1:MemberServiceEPR>
                <ns1:Content>
                    <ns8:AggregatorContent
xmlns:ns8="http://mds.globus.org/aggregator/types">
                        <ns8:AggregatorConfig>
                            <agg:GetMultipleResourcePropertiesPollType
xmlns:agg="http://mds.globus.org/aggregator/types">

<agg:PollIntervalMillis>60000</agg:PollIntervalMillis>

<agg:ResourcePropertyNames>rft:TotalNumberOfBytesTransferred</agg:Re
sourcePropertyNames>

<agg:ResourcePropertyNames>rft:TotalNumberOfActiveTransfers</agg:Res
ourcePropertyNames>

<agg:ResourcePropertyNames>rft:RFTFactoryStartTime</agg:ResourceProp
ertyNames>

<agg:ResourcePropertyNames>rft:ActiveResourceInstances</agg:Resource
PropertyNames>

<agg:ResourcePropertyNames>rft:TotalNumberOfTransfers</agg:ResourceP
ropertyNames>

</agg:GetMultipleResourcePropertiesPollType>
                        </ns8:AggregatorConfig>
```

```
                        <ns8:AggregatorData>
                            <ns1:TotalNumberOfBytesTransferred
xmlns:ns1="http://www.globus.org/namespaces/2008/04/rft">0</ns1:Tota
lNumberOfBytesTransferred>
                            <ns2:TotalNumberOfActiveTransfers
xmlns:ns2="http://www.globus.org/namespaces/2008/04/rft">0</ns2:Tota
lNumberOfActiveTransfers>
                            <ns3:RFTFactoryStartTime
xmlns:ns3="http://www.globus.org/namespaces/2008/04/rft">2009-10-
23T13:55:00.562Z</ns3:RFTFactoryStartTime>
                            <ns4:ActiveResourceInstances
xmlns:ns4="http://www.globus.org/namespaces/2008/04/rft">0</ns4:Acti
veResourceInstances>
                            <ns5:TotalNumberOfTransfers
xmlns:ns5="http://www.globus.org/namespaces/2008/04/rft">0</ns5:Tota
lNumberOfTransfers>
                        </ns8:AggregatorData>
                    </ns8:AggregatorContent>
                </ns1:Content>
            </ns1:Entry>
            <ns9:Entry xmlns:ns9="http://docs.oasis-open.org/wsrf/sg-
2">
                <ns9:ServiceGroupEntryEPR>
                    <ns10:Address
xmlns:ns10="http://www.w3.org/2005/08/addressing">http://192.108.38.
52:8080/wsrf/services/DefaultIndexServiceEntry</ns10:Address>
                    <ns11:ReferenceParameters
xmlns:ns11="http://www.w3.org/2005/08/addressing">
                        <ns12:ServiceGroupEntryKey
xmlns:ns12="http://mds.globus.org/inmemoryservicegroup">
                            <ns13:GroupKey
xmlns:ns13="http://mds.globus.org/aggregator/types">33549034</ns13:G
roupKey>
                            <ns14:EntryKey
xmlns:ns14="http://mds.globus.org/aggregator/types">17330288</ns14:E
ntryKey>
                        </ns12:ServiceGroupEntryKey>
                    </ns11:ReferenceParameters>
                </ns9:ServiceGroupEntryEPR>
                <ns9:MemberServiceEPR>
                    <ns15:Address
xmlns:ns15="http://www.w3.org/2005/08/addressing">local:/wsrf/servic
es/ManagedJobFactoryService</ns15:Address>
                    <ns16:ReferenceParameters
xmlns:ns16="http://www.w3.org/2005/08/addressing">
                        <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2008/03/gram/job">Multi<
/ns1:ResourceID>
                    </ns16:ReferenceParameters>
                </ns9:MemberServiceEPR>
                <ns9:Content>
                    <ns17:AggregatorContent
xmlns:ns17="http://mds.globus.org/aggregator/types">
                        <ns17:AggregatorConfig>
                            <agg:GetMultipleResourcePropertiesPollType
xmlns:agg="http://mds.globus.org/aggregator/types"
xmlns:factory="http://www.globus.org/namespaces/2008/03/gram/job">
```

```
<agg:PollIntervalMillis>60000</agg:PollIntervalMillis>

<agg:ResourcePropertyNames>glue:GLUECE</agg:ResourcePropertyNames>

<agg:ResourcePropertyNames>metric:ServiceMetaDataInfo</agg:ResourceP
ropertyNames>

<agg:ResourcePropertyNames>factory:localResourceManager</agg:Resourc
ePropertyNames>

</agg:GetMultipleResourcePropertiesPollType>
                        </ns17:AggregatorConfig>
                        <ns17:AggregatorData>
                            <ns1:ServiceMetaDataInfo
xmlns:ns1="http://mds.globus.org/metadata/2005/02">
                                <ns1:startTime>2009-10-
23T13:54:57.399Z</ns1:startTime>
                                <ns1:version>4.2.1</ns1:version>

<ns1:serviceTypeName>ManagedJobFactoryService</ns1:serviceTypeName>
                            </ns1:ServiceMetaDataInfo>
                            <ns2:localResourceManager
xmlns:ns2="http://www.globus.org/namespaces/2008/03/gram/job">Multi<
/ns2:localResourceManager>
                        </ns17:AggregatorData>
                    </ns17:AggregatorContent>
                </ns9:Content>
            </ns9:Entry>
            <ns18:Entry xmlns:ns18="http://docs.oasis-open.org/wsrf/sg-
2">
                <ns18:ServiceGroupEntryEPR>
                    <ns19:Address
xmlns:ns19="http://www.w3.org/2005/08/addressing">http://192.108.38.
52:8080/wsrf/services/DefaultIndexServiceEntry</ns19:Address>
                    <ns20:ReferenceParameters
xmlns:ns20="http://www.w3.org/2005/08/addressing">
                        <ns21:ServiceGroupEntryKey
xmlns:ns21="http://mds.globus.org/inmemoryservicegroup">
                            <ns22:GroupKey
xmlns:ns22="http://mds.globus.org/aggregator/types">33549034</ns22:G
roupKey>
                            <ns23:EntryKey
xmlns:ns23="http://mds.globus.org/aggregator/types">12203633</ns23:E
ntryKey>
                        </ns21:ServiceGroupEntryKey>
                    </ns20:ReferenceParameters>
                </ns18:ServiceGroupEntryEPR>
                <ns18:MemberServiceEPR>
                    <ns24:Address
xmlns:ns24="http://www.w3.org/2005/08/addressing">http://127.0.0.1:8
080/wsrf/services/monitoringexec</ns24:Address>
                </ns18:MemberServiceEPR>
                <ns18:Content>
                    <ns25:AggregatorContent
xmlns:ns25="http://mds.globus.org/aggregator/types">
                        <ns25:AggregatorConfig>
```

```
                            <agg:ExecutionPollType
xmlns:agg="http://mds.globus.org/aggregator/types">

<agg:PollIntervalMillis>30000</agg:PollIntervalMillis>

<agg:ProbeName>monitoringexec</agg:ProbeName>
                            </agg:ExecutionPollType>
                        </ns25:AggregatorConfig>
                        <ns25:AggregatorData>
                               <MonitoringData>
                                <ASC_ID>VARServer</ASC_ID>
                                <timestamp>2009-11-27
21:29:19</timestamp>
                                <MonitoringParameter>
                                    <Name>fps</Name>
                                    <Value>25</Value>
                                    <Unit>int</Unit>
                                </MonitoringParameter>
                                <MonitoringParameter>
                                    <Name>freememory</Name>
                                    <Value>47</Value>
                                    <Unit>byte</Unit>
                                </MonitoringParameter>
                                <MonitoringParameter>
                                    <Name>resolution</Name>
                                    <Value>QVGA</Value>
                                    <Unit>string</Unit>
                                </MonitoringParameter>
                            </MonitoringData>

                        </ns25:AggregatorData>
                    </ns25:AggregatorContent>
                </ns18:Content>
            </ns18:Entry>
            <ns26:Entry xmlns:ns26="http://docs.oasis-open.org/wsrf/sg-
2">
                <ns26:ServiceGroupEntryEPR>
                    <ns27:Address
xmlns:ns27="http://www.w3.org/2005/08/addressing">http://192.108.38.
52:8080/wsrf/services/DefaultIndexServiceEntry</ns27:Address>
                    <ns28:ReferenceParameters
xmlns:ns28="http://www.w3.org/2005/08/addressing">
                        <ns29:ServiceGroupEntryKey
xmlns:ns29="http://mds.globus.org/inmemoryservicegroup">
                            <ns30:GroupKey
xmlns:ns30="http://mds.globus.org/aggregator/types">33549034</ns30:G
roupKey>
                            <ns31:EntryKey
xmlns:ns31="http://mds.globus.org/aggregator/types">30356947</ns31:E
ntryKey>
                        </ns29:ServiceGroupEntryKey>
                    </ns28:ReferenceParameters>
                </ns26:ServiceGroupEntryEPR>
                <ns26:MemberServiceEPR>
                    <ns32:Address
xmlns:ns32="http://www.w3.org/2005/08/addressing">local:/wsrf/servic
es/irmos/wp5/factory/MonitoringService</ns32:Address>
```

```
                    <ns33:ReferenceParameters
xmlns:ns33="http://www.w3.org/2005/08/addressing">
                        <ns1:MonitoringResourceKey
xmlns:ns1="http://www.ntua.gr/namespaces/irmos/wp5/MonitoringService
_instance">30216727</ns1:MonitoringResourceKey>
                    </ns33:ReferenceParameters>
                </ns26:MemberServiceEPR>
                <ns26:Content>
                    <ns34:AggregatorContent
xmlns:ns34="http://mds.globus.org/aggregator/types">
                        <ns34:AggregatorConfig>
                            <agg:GetMultipleResourcePropertiesPollType
xmlns:agg="http://mds.globus.org/aggregator/types">

<agg:PollIntervalMillis>20000</agg:PollIntervalMillis>

<agg:ResourcePropertyNames>ffs:slaviolation</agg:ResourcePropertyNam
es>

</agg:GetMultipleResourcePropertiesPollType>
                        </ns34:AggregatorConfig>
                        <ns34:AggregatorData>

                            <MonitoringData>
                                <ASC_ID>Dustbuster</ASC_ID>
                                <timestamp>2009-11-27
21:29:19</timestamp>
                                <MonitoringParameter>
                                    <Name>fps</Name>
                                    <Value>25</Value>
                                    <Unit>int</Unit>
                                </MonitoringParameter>
                                <MonitoringParameter>
                                    <Name>freememory</Name>
                                    <Value>77</Value>
                                    <Unit>byte</Unit>
                                </MonitoringParameter>
                                <MonitoringParameter>
                                    <Name>resolution</Name>
                                    <Value>QVGA</Value>
                                    <Unit>string</Unit>
                                </MonitoringParameter>
                            </MonitoringData>

                        </ns34:AggregatorData>
                    </ns34:AggregatorContent>
                </ns26:Content>
            </ns26:Entry>
            <ns35:Entry xmlns:ns35="http://docs.oasis-open.org/wsrf/sg-
2">
                <ns35:ServiceGroupEntryEPR>
                    <ns36:Address
xmlns:ns36="http://www.w3.org/2005/08/addressing">http://192.108.38.
52:8080/wsrf/services/DefaultIndexServiceEntry</ns36:Address>
                    <ns37:ReferenceParameters
xmlns:ns37="http://www.w3.org/2005/08/addressing">
```

```
                        <ns38:ServiceGroupEntryKey
xmlns:ns38="http://mds.globus.org/inmemoryservicegroup">
                            <ns39:GroupKey
xmlns:ns39="http://mds.globus.org/aggregator/types">33549034</ns39:G
roupKey>
                            <ns40:EntryKey
xmlns:ns40="http://mds.globus.org/aggregator/types">19525584</ns40:E
ntryKey>
                        </ns38:ServiceGroupEntryKey>
                    </ns37:ReferenceParameters>
                </ns35:ServiceGroupEntryEPR>
                <ns35:MemberServiceEPR>
                    <ns41:Address
xmlns:ns41="http://www.w3.org/2005/08/addressing">local:/wsrf/servic
es/ManagedJobFactoryService</ns41:Address>
                    <ns42:ReferenceParameters
xmlns:ns42="http://www.w3.org/2005/08/addressing">
                        <ns1:ResourceID
xmlns:ns1="http://www.globus.org/namespaces/2008/03/gram/job">Fork</
ns1:ResourceID>
                    </ns42:ReferenceParameters>
                </ns35:MemberServiceEPR>
                <ns35:Content>
                    <ns43:AggregatorContent
xmlns:ns43="http://mds.globus.org/aggregator/types">
                        <ns43:AggregatorConfig>
                            <agg:GetMultipleResourcePropertiesPollType
xmlns:agg="http://mds.globus.org/aggregator/types"
xmlns:factory="http://www.globus.org/namespaces/2008/03/gram/job">

<agg:PollIntervalMillis>60000</agg:PollIntervalMillis>

<agg:ResourcePropertyNames>glue:GLUECE</agg:ResourcePropertyNames>

<agg:ResourcePropertyNames>metric:ServiceMetaDataInfo</agg:ResourceP
ropertyNames>

<agg:ResourcePropertyNames>factory:localResourceManager</agg:Resourc
ePropertyNames>

</agg:GetMultipleResourcePropertiesPollType>
                        </ns43:AggregatorConfig>
                        <ns43:AggregatorData>
                            <ns1:GLUECE
xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
                                <ns1:ComputingElement
ns1:Name="default" ns1:UniqueID="default">
                                    <ns1:Info ns1:GRAMVersion="4.2.1"
ns1:HostName="ksat116" ns1:LRMSType="Fork" ns1:LRMSVersion="1.0"
ns1:TotalCPUs="2"/>
                                    <ns1:State
ns1:EstimatedResponseTime="0" ns1:FreeCPUs="0" ns1:RunningJobs="0"
ns1:Status="enabled" ns1:TotalJobs="0" ns1:WaitingJobs="0"
ns1:WorstResponseTime="0"/>
                                    <ns1:Policy ns1:MaxCPUTime="-1"
ns1:MaxRunningJobs="-1" ns1:MaxTotalJobs="-1"
ns1:MaxWallClockTime="-1" ns1:Priority="0"/>
```

```
                                </ns1:ComputingElement>
                            </ns1:GLUECE>
                            <ns2:ServiceMetaDataInfo
xmlns:ns2="http://mds.globus.org/metadata/2005/02">
                                <ns2:startTime>2009-10-
23T13:54:57.630Z</ns2:startTime>
                                <ns2:version>4.2.1</ns2:version>

<ns2:serviceTypeName>ManagedJobFactoryService</ns2:serviceTypeName>
                            </ns2:ServiceMetaDataInfo>

<ns3:localResourceManager>Fork</ns3:localResourceManager>
                        </ns43:AggregatorData>
                    </ns43:AggregatorContent>
                </ns35:Content>
            </ns35:Entry>
            <ns44:RegistrationCount>
                <ns45:startTime>2009-10-
23T13:55:28.912Z</ns45:startTime>
                <ns46:lastChange
xmlns:ns46="http://mds.globus.org/metrics/2004/09">2009-10-
23T14:05:03.021Z</ns46:lastChange>
                <ns47:total
xmlns:ns47="http://mds.globus.org/metrics/2004/09">5</ns47:total>
                <ns48:expRate
xmlns:ns48="http://mds.globus.org/metrics/2004/09">
                    <ns48:rate>0.010934136033443877</ns48:rate>
                    <ns48:decay>60</ns48:decay>
                </ns48:expRate>
                <ns49:expRate
xmlns:ns49="http://mds.globus.org/metrics/2004/09">
                    <ns49:rate>0.001236173112029375</ns49:rate>
                    <ns49:decay>3600</ns49:decay>
                </ns49:expRate>
                <ns50:expRate
xmlns:ns50="http://mds.globus.org/metrics/2004/09">
                    <ns50:rate>5.758541815183995E-5</ns50:rate>
                    <ns50:decay>86400</ns50:decay>
                </ns50:expRate>
            </ns44:RegistrationCount>
            <ns51:ServiceMetaDataInfo
xmlns:ns51="http://mds.globus.org/metadata/2005/02">
                <ns51:startTime>2009-10-
23T13:55:28.915Z</ns51:startTime>
                <ns51:version>4.2.1</ns51:version>

<ns51:serviceTypeName>DefaultIndexService</ns51:serviceTypeName>
            </ns51:ServiceMetaDataInfo>
            <ns52:FixedTopicSet>false</ns52:FixedTopicSet>
            <ns54:TopicSet xmlns:ns53="http://docs.oasis-
open.org/wsrf/sg-2" xmlns:ns54="http://docs.oasis-open.org/wsn/b-2"
Dialect="http://docs.oasis-open.org/wsn/t-
1/TopicExpression/Simple">ns53:Entry</ns54:TopicSet>
            <ns55:TopicExpressionDialect xmlns:ns55="http://docs.oasis-
open.org/wsn/b-2">http://docs.oasis-open.org/wsn/t-
1/TopicExpression/Simple</ns55:TopicExpressionDialect>
```

```
          <ns56:TerminationTime
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true"/>
          <ns57:CurrentTime xmlns:ns57="http://docs.oasis-
open.org/wsrf/rl-2">2009-10-23T14:05:28.346Z</ns57:CurrentTime>
      </ns0:IndexRP>
```

Κώδικας 10.24 Παράδειγμα αρχείου A-SLA (A-SLA.xml)

```
      <?xml version="1.0" encoding="UTF-8"?>
      <p:Template xmlns:p="http://www.ggf.org/namespaces/ws-
agreement" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ggf.org/namespaces/ws-agreement
wsagreement.xsd ">
        <p:Context>
              <p:AgreementInitiator>
                      <p:Name>NCName</p:Name></p:AgreementInitiator>
              <p:AgreementProvider>

<p:Name>NCName</p:Name></p:AgreementProvider></p:Context>
        <p:Terms>
              <p:ServiceDescriptionTerm>
              <app:Application
binaryUri="http://www.somewhere.com/appURI"
id="VARLowResApplicaiton" version="0.5" xmlns:app="http://www.irmos-
project.eu/application" xmlns:speclang="http://www.irmos-
project.eu/speclang/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:wf="http://www.irmosproject.eu/workflow/1.0"
xsi:schemaLocation="http://www.irmos-project.eu/application
application.xsd http://www.irmosproject.eu/workflow/1.0 workflow.xsd
">
                <app:qoS mtbf="1D" mttr="1D"/>
                <app:appParam name="serverTranscoding"
paramRef="VarServer.transcoding" value=""/><app:appParam
name="consumerTranscoding" paramRef="VarClient.transcoding"
value=""/>
                 <app:components
binaryUri="http://www.somewhere.com/ascURI" id="Dustbuster"
version="0.1">
                      <parameters name="fps" type="int"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                            <speclang:BasicValue type="number"
value="15"/>
                      </parameters>
                      <parameters name="resolution" type="string"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                            <speclang:BasicValue type="string"
value="QVGA"/>
                            <valuesAllowed type="string"
value="XGA"/>
```

```
                                <valuesAllowed type="string"
value="QVGA"/>
                                <valuesAllowed type="string"
value="SVGA"/>
                                <valuesAllowed type="string"
value="VGA"/>
                                <valuesAllowed type="string"
value="SXGA"/>
                                <valuesAllowed type="string"
value="UXGA"/>
                                <valuesAllowed type="string"
value="DV"/>
                                <valuesAllowed type="string"
value="720p"/>
                                <valuesAllowed type="string"
value="1080p"/>
                        </parameters>
                        <parameters name="storage_size" type="int"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="string"
value="100"/>
                        </parameters>
                        <parameters name="colordepth" type="int"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="number"
value="24"/>
                        </parameters>
                        <parameters name="InputType" type="string"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="string"
value="Network"/>
                                <valuesAllowed type="string"
value="Network"/>
                                <valuesAllowed type="string"
value="Filesytem"/>
                        </parameters>
                        <parameters name="OutputType" type="string"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="string"
value="Network"/>
                                <valuesAllowed type="string"
value="Network"/>
                                <valuesAllowed type="string"
value="Filesytem"/>
                        </parameters>
                        <parameters name="metadata_size" type="int"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="number"
value="104"/>
                        </parameters>
```

```
                        <parameters name="server_ready" type="boolean"
isWorkload="false" isConfig="false" description=""
monitorable="true">
                        </parameters>
                        <parameters name="heart_beat" type="boolean"
isWorkload="false" isConfig="false" description=""
monitorable="true">
                        </parameters>
                        <parameters name="Transcoding" type="boolean"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                        </parameters>
                </app:components>
                <app:components
binaryUri="http://www.somewhere.com/ascURI" id="VARClient"
version="0.1">
                        <parameters name="fps" type="int"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="number"
value="15"/>
                        </parameters>
                        <parameters name="resolution" type="string"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="string"
value="QVGA"/>
                                <valuesAllowed type="string"
value="XGA"/>
                                <valuesAllowed type="string"
value="QVGA"/>
                                <valuesAllowed type="string"
value="SVGA"/>
                                <valuesAllowed type="string"
value="VGA"/>
                                <valuesAllowed type="string"
value="SXGA"/>
                                <valuesAllowed type="string"
value="UXGA"/>
                                <valuesAllowed type="string"
value="DV"/>
                                <valuesAllowed type="string"
value="720p"/>
                                <valuesAllowed type="string"
value="1080p"/>
                        </parameters>
                        <parameters name="colordepth" type="int"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="number"
value="24"/>
                        </parameters>
                        <parameters name="storage_size" type="int"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="number"
value="100"/>
```

```
                        </parameters>
                        <parameters name="InputType" type="string"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="string"
value="Network"/>
                                <valuesAllowed type="string"
value="Network"/>
                                <valuesAllowed type="string"
value="Filesytem"/>
                        </parameters>
                        <parameters name="OutputType" type="string"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="string"
value="Network"/>
                                <valuesAllowed type="string"
value="Network"/>
                                <valuesAllowed type="string"
value="Filesytem"/>
                        </parameters>
                        <parameters name="metadata_size" type="int"
isWorkload="false" isConfig="false" description=""
monitorable="false">
                                <speclang:BasicValue type="number"
value="104"/>
                        </parameters>
                </app:components>
                <app:components id="VARServiceProvider"/>
                <app:components id="VARApplicationConsumer"/>
                <app:links endOne="VARServiceProvider"
endTwo="VARServer" id="Provider_Server">
                  <app:QoSParameter bandwidth="0" jitter="0"
latency="0"/>
                </app:links>
                <app:links endOne="VARServer" endTwo="VARClient"
id="Server_Client">
                        <app:QoSParameter bandwidth="0" jitter="0"
latency="0"/>
                </app:links>
                <app:links endOne="VARApplicationConsumer"
endTwo="VarClient" id="Client_Consumer">
                        <app:QoSParameter bandwidth="0" jitter="0"
latency="0"/>
                </app:links>
                <app:workflow name="VARWorkflow">
                        <sequence>
                                <empty/>
                        </sequence></app:workflow>
              </app:Application>
              </p:ServiceDescriptionTerm></p:Terms>
      </p:Template>
```

Κώδικας 10.25 Ρυθμίσεις ασφαλείας για τον Tomcat Server (web.xml)

```
      <?xml version="1.0" encoding="UTF-8"?>
      <web-app version="2.5"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
          <display-name>IrmosPortal Gui</display-name>
          <session-config>
              <session-timeout>20</session-timeout>
          </session-config>
          <welcome-file-list>
              <welcome-file>index.jsp</welcome-file>
              </welcome-file-list>
          <security-constraint>
              <display-name>Userconstraint</display-name>
              <web-resource-collection>
                  <web-resource-name>secured</web-resource-name>
                  <description/>
                  <url-pattern>/index.jsp</url-pattern>
                  <url-pattern>/negotiate1.jsp</url-pattern>
                  <url-pattern>/negotiate2.jsp</url-pattern>
                  <url-pattern>/monitoring.jsp</url-pattern>
                  <http-method>GET</http-method>
                  <http-method>POST</http-method>
                  <http-method>HEAD</http-method>
                  <http-method>PUT</http-method>
                  <http-method>OPTIONS</http-method>
                  <http-method>TRACE</http-method>
                  <http-method>DELETE</http-method>
              </web-resource-collection>
              <auth-constraint>
                  <description/>
                  <role-name>User</role-name>
                  <role-name>manager</role-name>
                  </auth-constraint>
              </security-constraint>
          <login-config>
              <auth-method>FORM</auth-method>
              <realm-name>file</realm-name>
              <form-login-config>
                  <form-login-page>/login.jsp</form-login-page>
                  <form-error-page>/error.jsp</form-error-page>
                  </form-login-config>
              </login-config>
          <security-role>
              <description/>
              <role-name>User</role-name>
          </security-role>
          <security-role>
              <description/>
              <role-name>manager</role-name>
```

```
        </security-role>
        </web-app>
```