



Πανεπιστήμιο Πειραιώς

Τμήμα Ψηφιακών Συστημάτων

Μεταπτυχιακό Πρόγραμμα Σπουδών
Διδακτική της Τεχνολογίας & Ψηφιακά Συστήματα
Κατεύθυνση Δικτυοκεντρικών Συστημάτων

Προσομοίωση λειτουργίας TaxisNet σε υπολογιστικό νέφος

Μαθιουδάκης Μανούσος ME10090

Πανεπιστήμιο Πειραιώς

Πίνακας περιεχομένων

Κεφάλαιο 1.....	4
1.1 Εισαγωγή.....	4
1.2 Σκοπός - αντικειμενικοί στόχοι.....	7
1.3 Δομή εργασίας.....	7
Κεφάλαιο 2.....	8
2.1 Χαρακτηριστικά υπολογιστικού νέφους.....	8
2.2 Μοντέλα ανάπτυξης υπολογιστικών νεφών.....	9
2.3 Συστατικά υπολογιστικού νέφους - μοντέλα υπηρεσιών.....	10
2.4 Πλεονεκτήματα υπολογιστικού νέφους.....	20
2.5 Μειονεκτήματα υπολογιστικού νέφους.....	22
2.6 Πότε το υπολογιστικό νέφος είναι κατάλληλο.....	24
2.7 Πότε να αποφεύγουμε το υπολογιστικό νέφος.....	24
Κεφάλαιο 3.....	26
3.1 Τι είναι το TAXISnet;.....	26
3.2 Στόχος του νέου TAXISnet.....	26
3.3 Προβλήματα λειτουργίας.....	27
3.4 Αίτια προβλημάτων.....	27
3.5 Η προσέγγιση του Υπολογιστικού Νέφους.....	28
3.6 Υποθέσεις.....	28
Κεφάλαιο 4.....	30
4.1 Τεκμηρίωση ερευνητικής μεθοδολογίας.....	30
4.1.1 Τρόπος εκτέλεσης.....	30
4.1.2 Πρώτο σενάριο δοκιμής.....	31
4.1.3 Δεύτερο σενάριο δοκιμής.....	31
4.1.4 Τρίτο σενάριο δοκιμής.....	31
5.1 Αρχιτεκτονική.....	32
5.1.1 Διάγραμμα παράταξης.....	33
5.1.2 Διαγράμματα πακέτων.....	36
5.2 Η βάση δεδομένων TaxisDB.....	38
5.2.1 Δομή πίνακα Contact.....	40
5.2.2 Δομή πίνακα E1.....	40
5.2.3 Δομή πίνακα E1DataFromTaxPayerFolder.....	41
5.2.4 Δομή πίνακα E1ExpensesRemovedFromTotalIncome.....	42
5.2.5 Δομή πίνακα E1IncomesReduceTaxes.....	44
5.2.6 Δομή πίνακα E1InfoData.....	46
5.2.7 Δομή πίνακα E1NAUTICALINCOMES.....	48
5.2.8 Δομή πίνακα E1ObjectiveSpending.....	49
5.2.9 Δομή πίνακα E1PersonDataBorneTaxpayer.....	53
5.2.10 Δομή πίνακα E1PrepaidTaxes.....	54
5.2.11 Δομή πίνακα E1ReduceTax.....	56
5.2.12 Δομή πίνακα E1RelatePersons.....	57
5.2.13 Δομή πίνακα E1TaxPayerBankAccount.....	58
5.2.14 Δομή πίνακα E1TaxableIncomes.....	59
5.2.15 Δομή πίνακα E2.....	62

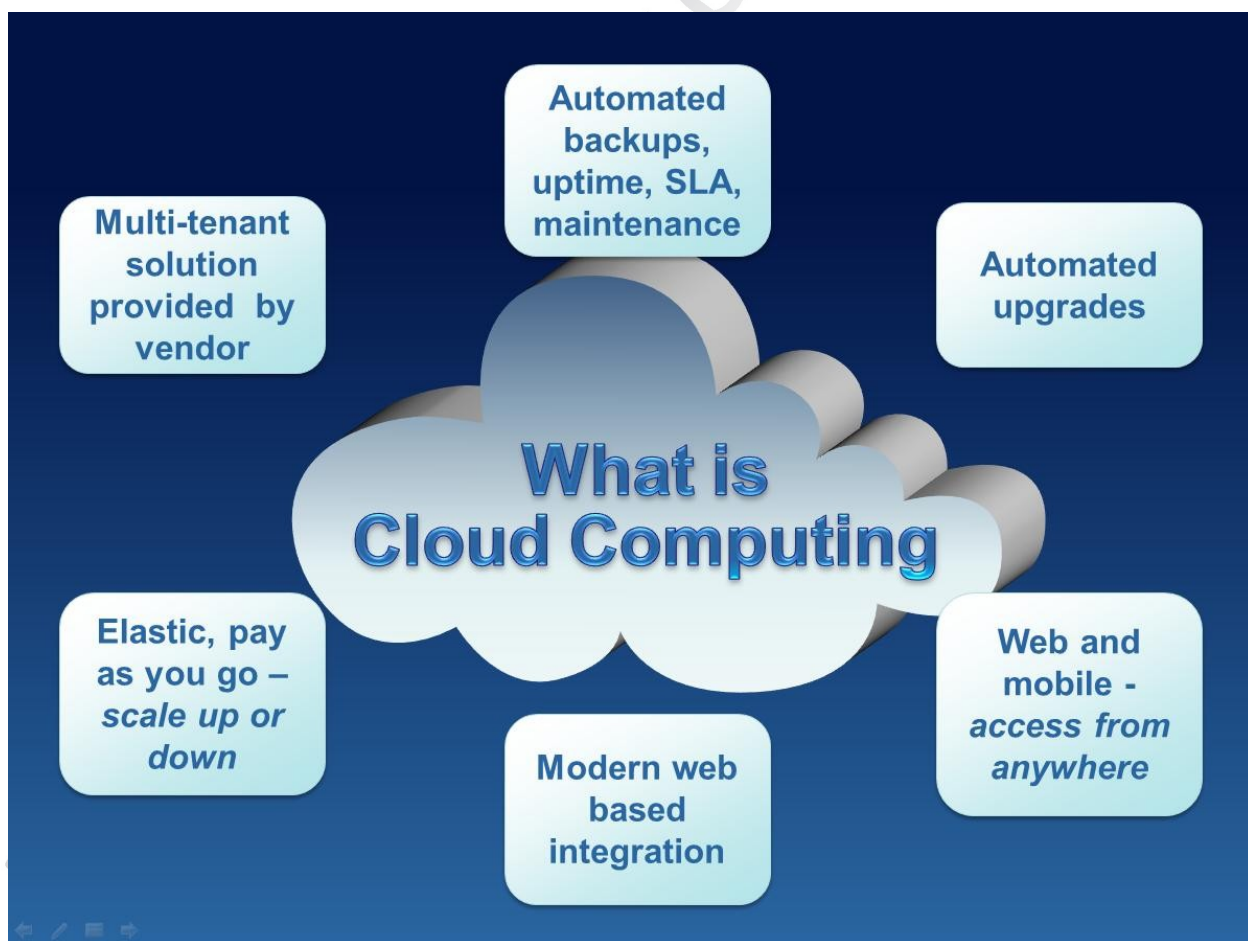
5.2.16 Δομή πίνακα E2Estate.....	63
5.2.17 Δομή πίνακα E2OtherEstate.....	64
5.2.18 Δομή πίνακα IncomeTax.....	65
5.2.19 Δομή πίνακα RelatePerson.....	66
5.2.20 Δομή πίνακα Taxpayer.....	67
5.2.21 Δομή πίνακα User.....	69
5.2.22 Δομή πίνακα E2coOwner.....	69
5.3 Τεκμηρίωση.....	71
5.3.1 Περιπτώσεις χρήσης.....	71
5.3.2 Διαγράμματα τάξεων.....	71
5.3.3 Διαγράμματα ακολουθίας.....	74
5.4 Οθόνες εφαρμογής.....	85
Κεφάλαιο 6.....	96
6.1 Αποτελέσματα Σεναρίου 1.....	96
6.2 Αποτελέσματα Σεναρίου 2.....	96
6.3 Αποτελέσματα Σεναρίου 3.....	97
6.4 Προβλήματα.....	98
6.5 Ανάλυση.....	99
6.6 Συμπεράσματα.....	100
Βιβλιογραφία.....	101
Πηγές.....	102

Πανεπιστήμιο Πειραιώς

Κεφάλαιο 1

1.1 Εισαγωγή

Το υπολογιστικό νέφος είναι το επόμενο στάδιο της επανάστασης που φέρνει το ιντερνέτ. Παρέχει τα μέσα με το οποίο τα πάντα από εφαρμογές, επιχειρηματικές διαδικασίες, υπολογιστική ισχύ, υπολογιστικές υποδομές κτλ μπορούν να μας διατεθούν ως υπηρεσία, όπου, όποτε και όσο το χρειαζόμαστε. Ως ορισμό του υπολογιστικού νέφους θα μπορούσαμε να πούμε ότι είναι ο συνδυασμός των δικτύων, του υλικού, των υποδομών, των υπηρεσιών, της αποθήκευσης και των υποδομών που προσφέρουν τις πτυχές του computing ως υπηρεσία. Οι υπηρεσίες του υπολογιστικού νέφους περιλαμβάνουν παροχή υπηρεσιών υποδομών, λογισμικού αποθήκευσης κτλ μέσω του ιντερνέτ, είτε ως ολοκληρωμένη πλατφόρμα είτε ως ξεχωριστό κομμάτι, ανάλογα τις απαιτήσεις του χρήστη.

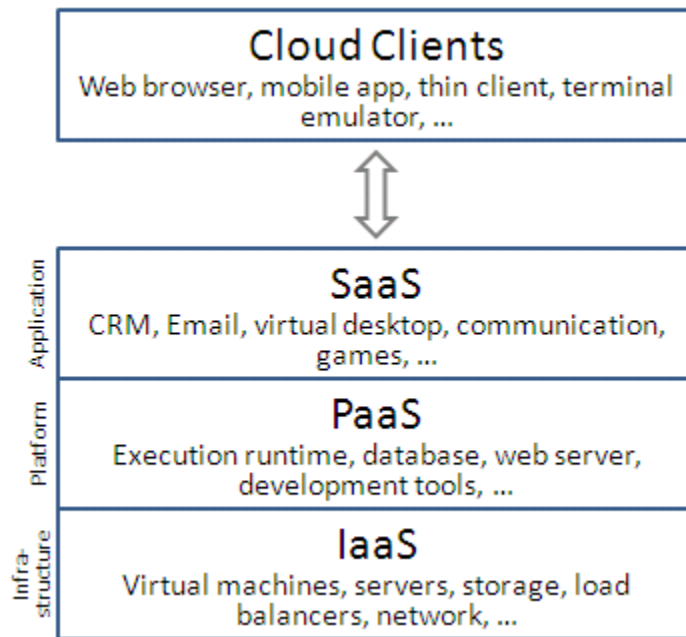


Εικόνα 1.1 Τι είναι υπολογιστικό νέφος
(<http://abouttmc.com/wp-content/uploads/2012/03/What-is-Cloud-Computing.jpg>)

Τα βασικότερα μοντέλα υπηρεσιών του υπολογιστικού νέφους είναι :

1. Υποδομές ως υπηρεσία (IaaS)

2. Πλατφόρμα ως υπηρεσία (PaaS)
3. Λογισμικό ως υπηρεσία (SaaS)



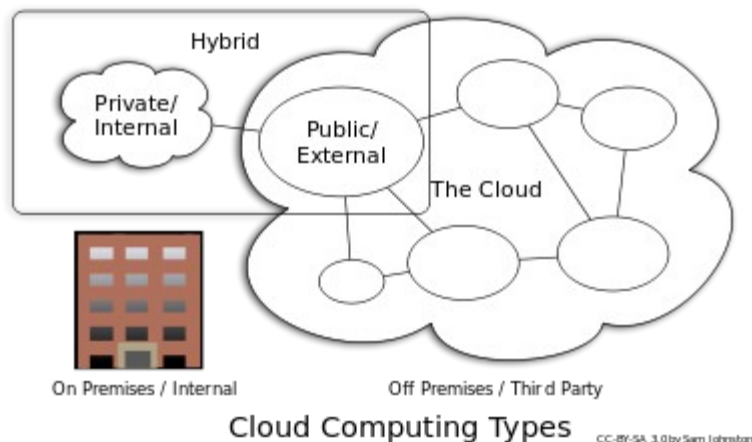
Εικόνα 1.2 Μοντέλα υπηρεσιών υπολογιστικού νέφους
 (http://upload.wikimedia.org/wikipedia/commons/3/3c/Cloud_computing_layers.png)

Τα σημαντικότερα χαρακτηριστικά του υπολογιστικού νέφους είναι :

1. Πληρωμή ανά χρήση
2. Elasticity και η δυνατότητα αναβάθμισης και υποβάθμισης των πόρων
3. Αυτοεξυπηρέτηση κατά απαίτηση

Τέλος το υπολογιστικό νέφος παρέχεται μέσω των παρακάτω τρόπων:

1. Δημόσιο υπολογιστικό νέφος
2. Ιδιωτικό υπολογιστικό νέφος
3. Υβριδικό υπολογιστικό νέφος (το οποίο αποτελείται από το δημόσιο και το ιδιωτικό)



Εικόνα 1.3 Τύποι υπολογιστικού νέφους
http://upload.wikimedia.org/wikipedia/commons/8/87/Cloud_computing_types.svg

Ως σχετικά νέα τεχνολογία, η υιοθέτηση του υπολογιστικού νέφους έχει κάποιες προκλήσεις που πρέπει να λυθούν. Οι πιο σημαντικές από αυτές είναι :

1. Ασφάλεια πληροφοριών
 - a. Ταυτοποίηση
 - b. Ιδιωτικότητα και ακεραιότητα
 - c. Έλεγχος πρόσβασης, πρόληψη επιθέσεων
 - d. Πιστοποίηση και επαλήθευση
 - e. Μυστικότητα δεδομένων (privacy)
2. Τιμολόγηση παρεχόμενων υπηρεσιών
 - a. Ποιότητα παρεχόμενων υπηρεσιών και παρακολούθηση SLA
 - b. Τύπος και τοποθεσία επεξεργασίας δεδομένων
 - c. Διαχείριση αδειών
3. Διαλειτουργικότητα και μεταφερσιμότητα
 - a. Μετανάστευση προς/από το υπολογιστικό νέφος
 - b. Δυνατότητα ολοκλήρωσης με τις υπάρχουσες (on premises) IT υποδομές
 - c. Μεταφερσιμότητα υπηρεσιών και δεδομένων
4. Συμμόρφωση με τις απαιτήσεις και τους κανόνες
5. Διασφάλιση δίκαιου ανταγωνισμού στην αγορά
6. Αποδοτικότητα παρεχόμενης υπηρεσίας
 - a. Χρήσης των συστατικών και εργαλείων ανάπτυξης
 - b. Δημιουργία επεκτάσιμων αρχιτεκτονικών
 - c. Διαχείριση πόρων
 - d. Διαθεσιμότητα υπηρεσιών
7. Αποτελεσματικότητα στη χρήση και έλεγχο των υπηρεσιών
 - a. Μηχανισμούς διακυβέρνηση
 - b. Έλεγχος υπηρεσιών από τους χρήστες
 - c. Συμβόλαια που περιλαμβάνουν θέματα ευθύνης

1.2 Σκοπός - αντικειμενικοί στόχοι

Σκοπός της παρούσας εργασίας είναι να αναδείξουμε ένα τρόπο για την αποφυγή των προβλημάτων (που έχουν εμφανιστεί ως σήμερα) και τη βελτίωση των παρεχόμενων υπηρεσιών του υπουργείου οικονομικών προς τα φυσικά και νομικά πρόσωπα, μέσω του πληροφοριακού συστήματος TaxisNet. Επιχειρώ να προσομοιώσω την λειτουργία ενός μικρού τμήματος αυτού του πληροφοριακού συστήματος που θα παρέχεται μέσω υπολογιστικού νέφους. Το τμήμα του συστήματος αφορά την φορολόγηση φυσικών προσώπων και συγκεκριμένα την καταχώριση και εκκαθάριση των φορολογικών εντύπων Ε1 και Ε2. Απώτερος στόχος είναι η παροχή ενός συστήματος όπου δε θα παρουσιάζει προβλήματα διαθεσιμότητας της υπηρεσίας.

1.3 Δομή εργασίας

Η παρούσα μεταπτυχιακή διπλωματική εργασία αποτελείται από έξι κεφάλαια. Στο κεφάλαιο 1 που αποτελεί την εισαγωγή, περιλαμβάνει συνοπτικά τις πρώτες βασικές έννοιες του υπολογιστικού νέφους και των βασικών συστατικών και χαρακτηριστικών του. Επιπλέον, γίνεται αναφορά στο σκοπό της συγκεκριμένης εργασίας.

Στο κεφάλαιο 2 αναλύονται οι θεωρητικές έννοιες σχετικά με το υπολογιστικό νέφος, τα χαρακτηριστικά, τα συστατικά, πλεονεκτήματα - μειονεκτήματα και πότε συνίσταται η υιοθέτηση αυτού.

Το κεφάλαιο 3 αφορά το πληροφοριακό σύστημα TaxisNet του υπουργείου οικονομικών. Συγκεκριμένα, γίνεται αναφορά στα προβλήματα που έχουν προκύψει στην έως τώρα λειτουργία του. Αναλύονται οι αιτίες των προβλημάτων και τέλος παρουσιάζεται η προσέγγιση του υπολογιστικού νέφους ως μοντέλο που μπορεί να δώσει λύση σε κάποια από αυτά τα προβλήματα. Τέλος, για τους σκοπούς αυτής της εργασίας γίνεται αναφορά σε κάποιους περιορισμούς και υποθέσεις που έθεσα κατά την ανάπτυξη του συστήματος.

Το κεφάλαιο 4 παρουσιάζει τον τρόπο με τον οποίο έγιναν οι έλεγχοι και τα διάφορα σενάρια δοκιμών στο σύστημα που αναπτύχθηκε στα πλαίσια της εργασίας.

Στο κεφάλαιο 5 παρουσιάζεται η αρχιτεκτονική του συστήματος που αναπτύχθηκε, η τεκμηρίωση, η σχεδίαση της βάσης δεδομένων και οι οθόνες της εφαρμογής.

Στο κεφάλαιο 6 γίνεται η ανάλυση των αποτελεσμάτων από τα σενάρια εκτέλεσης που περιγράφηκαν στο κεφάλαιο 4.

Κεφάλαιο 2

Σύμφωνα με τον NIST (National Institute of Standards and Technology), το υπολογιστικό νέφος είναι ένα μοντέλο πληρωμής κατά χρήση (pay per use) για την ενεργοποίηση διαθέσιμων, άνετων, κατά απαίτηση δικτυακή πρόσβαση σε διαμοιραζόμενους και διαμορφώσιμους υπολογιστικούς πόρους (όπως δίκτυα, εξυπηρετητές, χώρο αποθήκευσης, εφαρμογές, υπηρεσίες και άλλα) τα οποία μπορούν γρήγορα να παρέχονται και να διατίθενται με την ελάχιστη προσπάθεια διαχείρισης ή αλληλεπίδρασης με τον φορέα παροχής υπηρεσιών.

2.1 Χαρακτηριστικά υπολογιστικού νέφους

Το μοντέλο που ορίστηκε προηγουμένως, προάγει την διαθεσιμότητα και αποτελείται από πέντε βασικά χαρακτηριστικά:

1. Αυτοεξυπηρέτηση κατά απαίτηση (On-demand self-service). Επιτρέπουν στον χρήστη να αποκτούν, να ρυθμίζουν και να αναπτύσσουν υπηρεσίες μόνοι τους με την χρήση καταλόγων υπηρεσιών του υπολογιστικού νέφους χωρίς την βοήθεια κάποιου τμήματος IT. Η απαίτηση αυτοεξυπηρέτησης στο υπολογιστικό νέφος προτρέπει τους προμηθευτές υποδομών να δημιουργήσουν πρότυπα (templates) υπολογιστικών νεφών, τα οποία προέρχονται από καταλόγους υπηρεσιών υπολογιστικών νεφών. Μερικά τέτοια παραδείγματα προτύπων είναι το HP Cloud Maps¹ της Hewlett-Packard και το CloudForms² της RedHat.
2. Δικτυακή πρόσβαση από παντού (Ubiquitous network access). Είναι η δυνατότητα διαθεσιμότητας και πρόσβασης μέσω δικτύου, μέσω τυποποιημένων μηχανισμών που προωθούν την χρήση ετερογενών πλατφορμών πελάτη όπως κινητά τηλέφωνα, φορητούς υπολογιστές και ταμπλέτες.
3. Ανεξαρτησία από την φυσική τοποθεσία των πόρων (Location - independent resource pooling). Οι υπολογιστικοί πόροι του παρόχου ενώνονται για να εξυπηρετήσουν πολλούς καταναλωτές χρησιμοποιώντας ένα μοντέλο πολυενοικίασης. Με αυτό το μοντέλο διαφορετικοί φυσικοί και εικονικοί πόροι δυναμικά ανατίθενται και επανεκχωρούνται ανάλογα με την ζήτηση των πελατών. Δημιουργείτε η αίσθηση στον πελάτη της ανεξαρτησίας καθώς δεν γνωρίζει ακριβώς την φυσική τοποθεσία που βρίσκονται οι παρεχόμενοι πόροι που ενοικιάζει. Ενδέχεται να γνωρίζει μόνο την ήπειρο, τη χώρα ή μια περιφέρεια κάποιας χώρας.
4. Rapid elasticity. Είναι η δυνατότητα της δυναμικά και σε σχεδόν πραγματικό χρόνο γρήγορης και ελαστικής παροχής αναβάθμισης ή ομοίως υποβάθμισης των υπολογιστικών πόρων χωρίς την παρουσία κάποιου μηχανικού.

¹ http://www8.hp.com/us/en/business-solutions/solution.html?compURI=1216661#_UGnsXIUxo48

² <http://www.redhat.com/solutions/cloud-computing/red-hat-cloud/private-clouds-cloudforms.html>

5. Πληρωμή ανά χρήση (Pay per use). Είναι η δυνατότητα χρέωσης των υπηρεσιών του υπολογιστικού νέφους με έναν μετρητή. Τέτοια παραδείγματα μετρήσεων είναι η χωρητικότητα αποθήκευσης, το εύρος ζώνης, οι ώρες χρήσεις της κεντρικής μονάδας επεξεργασίας και άλλα. Ο πελάτης πληρώνει ότι χρησιμοποιεί και όσο το χρησιμοποιεί .

2.2 Μοντέλα ανάπτυξης υπολογιστικών νεφών

Το λογισμικό που εκτελείτε/ φιλοξενείται σε υπολογιστικό νέφος εκμεταλλεύεται πλήρως το μοντέλο του υπολογιστικού νέφους όταν είναι υπηρεσιοστρεφές. Παρόλο αυτά, όλες οι προσεγγίσεις υπολογιστικού νέφους δεν είναι ίδιες. Υπάρχουν διάφορα μοντέλα ανάπτυξης (deployment model) υπολογιστικού νέφους, όπως :

1. Δημόσιο υπολογιστικό νέφος (Public Clouds). Σε αυτό το μοντέλο οι πόροι και οι υπηρεσίες του υπολογιστικού νέφους παρέχονται στο ευρύ κοινό από κάποιον πάροχο. Αυτές οι υπηρεσίες μπορεί να είναι δωρεάν είτε να χρεώνονται με το μοντέλο πληρωμή ανά χρήση. Μπορεί να ανήκει, να το διαχειρίζεται και να λειτουργεί από μια εταιρεία, έναν οργανισμό, ένα ακαδημαϊκό ίδρυμα ή κάποιο κυβερνητικό οργανισμό. Οι υποδομές του υπάρχουν στις εγκαταστάσεις του παρόχου .
2. Ιδιωτικό υπολογιστικό νέφος (Private Clouds). Σε αυτό το μοντέλο οι υποδομές του υπολογιστικού νέφους είναι μισθωμένες ή ανήκουν σε κάποιο οργανισμό, εταιρεία κτλ και λειτουργούν αποκλειστικά για αυτόν. Την διαχείριση των υποδομών και των υπηρεσιών μπορεί να γίνεται είτε από τον ίδιο τον οργανισμό είτε από τρίτους. Η φιλοξενία μπορεί να είναι είτε εντός του οργανισμού είτε εκτός σε κάποιον πάροχο που προσφέρει ιδιωτικά νέφη. Οι υποδομές μπορεί να φιλοξενούνται εσωτερικά της εταιρείας/οργανισμού είτε εξωτερικά.

Τα ιδιωτικά νέφη είναι σαν τα υπολογιστικά νέφη υποδομών τα οποία αξιοποιούν την εικονικοποίηση (virtualization) και βρίσκονται μέσα σε ιδιωτικά υπολογιστικά κέντρα δεδομένων. Το υπολογιστικό νέφος είναι μια προσέγγιση για την βελτιστοποίηση της χρήσης του υλικού και του λογισμικού και μπορούμε να το επιτύχουμε στον ίδιο βαθμό, χρησιμοποιώντας στο υπολογιστικό κέντρο δεδομένων την εικονικοποίηση των πόρων. Στατιστικά, το 5% της μέγιστης διαθέσιμης χωρητικότητας υπολογιστικών πόρων (σε εξυπηρετητές βάσεων δεδομένων, εξυπηρετητές εφαρμογών κτλ), στα περισσότερα υπολογιστικά κέντρα δεδομένων χρησιμοποιείται συνέχεια.

Το λογισμικό εικονικοποίησης (virtualization) όπως το VMWare, μας δίνει την δυνατότητα να αντιμετωπίζουμε πολλούς φυσικούς εξυπηρετητές ως έναν εικονικό και έτσι αξιοποιούμε την επεξεργαστική ισχύ όλων των υπολογιστικών πόρων όπως αν ήταν ένας μοναδικός. Μπορούμε να βελτιστοποιήσουμε την χρήση των πόρων του συνολικού υλικού και του λογισμικού περισσότερο σε σχέση με το αν ήταν δεσμευμένο σε ένα συγκεκριμένο υλικό. Το λογισμικό εικονικοποίησης μπορεί να καταναίμει το φορτίο των διεργασιών σε όλους τους διαθέσιμους φυσικούς εξυπηρετητές . Με αυτόν τον τρόπο βελτιώνει την αξιοποίηση του κάθε υπολογιστικού πόρου. Το τελικό αποτέλεσμα είναι ότι μπορούμε να υποστηρίξουμε περισσότερα δεδομένα, υπηρεσίες και διεργασίες με λιγότερο αριθμό εξυπηρετητών . Ο μηχανισμός εικονικοποίησης μειώνει το κόστος.

Οι επιχειρήσεις ενδιαφέρονται περισσότερο για τα στα ιδιωτικά νέφη επειδή σε πολλές περιπτώσεις, δεν μπορούν να φιλοξενούνται τα δεδομένα τους εκτός του τοίχους

προστασίας τους (firewall) για λόγους ιδιωτικότητας αλλά και για νομικά θέματα. Όμως επιθυμούν να έχουν τα πλεονεκτήματα που προσφέρει αρχιτεκτονική του υπολογιστικού νέφους. Πολλές από αυτές τις εταιρείες/οργανισμούς έχουν ήδη επενδύσει σε υλικό και λογισμικό και η επένδυση αυτή πρέπει να αποσβεστεί, επιπλέον θέλουν να έχουν τον έλεγχο των συστημάτων και των πληροφοριών.

3. Κοινοτικό υπολογιστικό νέφος (Community cloud). Σε αυτό το μοντέλο οι υποδομές του υπολογιστικού νέφους μοιράζονται από κάποιες κοινότητες. Η λειτουργία, η διαχείριση και ο ιδιοκτήτης μπορεί να είναι μια από τις κοινότητες ή κάποιος τρίτος. Αυτός που έχει επιφορτιστεί με αυτά έχει ευθύνη για τις απαιτήσεις ασφάλειας, την πολιτική και άλλα.
4. Υβριδικό υπολογιστικό νέφος (Hybrid Clouds). Σε αυτή την αρχιτεκτονική οι υποδομές του υπολογιστικού νέφους αποτελούνται από συνδυασμό τουλάχιστον δύο ή περισσότερων εκ των αναφερόμενων αρχιτεκτονικών (ιδιωτικό, δημόσιο και κοινότητας) που παραμένουν μοναδικές οντότητες αλλά συνδέονται μεταξύ τους με τυποποιημένους ή ιδιόκτητη τεχνολογία που επιτρέπει την φορητότητα εφαρμογών και δεδομένων .

2.3 Συστατικά υπολογιστικού νέφους - μοντέλα υπηρεσιών

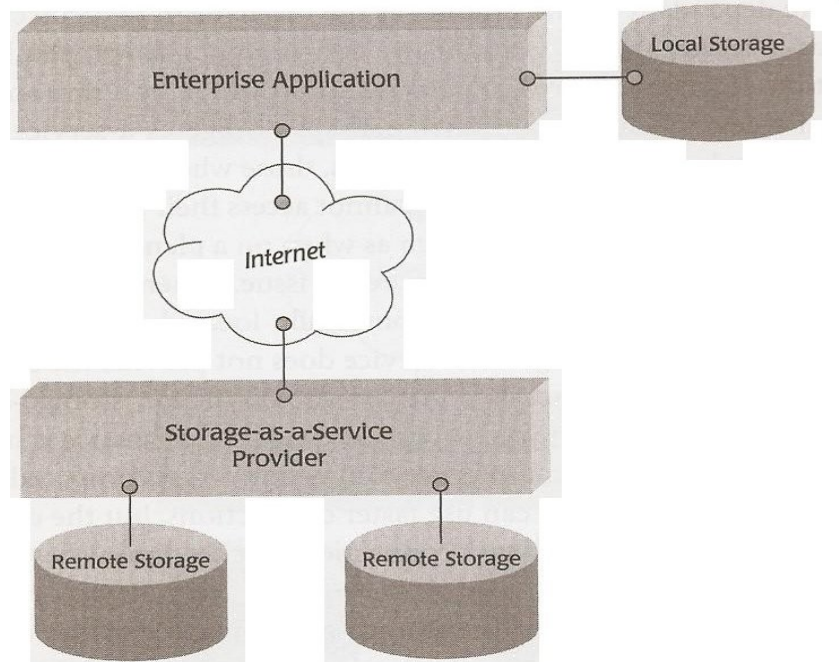
Υπάρχουν έντεκα κύρια μοντέλα υπηρεσιών που διατίθενται μέσω του υπολογιστικού νέφους.

Τα μοντέλα αυτά όπως φαίνονται και στο εικόνα 2.1 είναι:



Εικόνα 2.1 Συστατικά υπολογιστικού νέφους
(Cloud Computing and SOA Convergence in Your Enterprise . Pearson Education, 2009)

1. Αποθήκευση ως υπηρεσία (Storage as a service). Το μοντέλο αυτό μας επιτρέπει να αποθηκεύουμε δεδομένα σε απομακρυσμένο δίσκο σαν να ήταν τοπικός. Είναι το επιχειρηματικό μοντέλο όπου οι πάροχοι υπολογιστικών νεφών ενοικιάζει χώρο αποθήκευσης σε εταιρείες/οργανισμούς ή και ιδιώτες. Αυτή η υπηρεσία είναι το κυριότερο συστατικό του υπολογιστικού νέφους καθώς αξιοποιείτε από τα υπόλοιπα συστατικά. Παράδειγμα αυτής της υπηρεσίας είναι το Google Cloud Storage³. Η αρχιτεκτονική του μοντέλου αυτού απεικονίζεται στην εικόνα 2.2.



Εικόνα 2.2 Αρχιτεκτονική αποθήκευσης ως υπηρεσία (Storage as a service) (Cloud Computing and SOA Convergence in Your Enterprise. Pearson Education, 2009)

Μπορεί να μην ακούγεται λογικό να αποθηκεύουμε απομακρυσμένα τα δεδομένα μέσω ίντερντ την στιγμή που οι σκληροί δίσκοι είναι φτηνοί και συνεχώς πέφτουν οι τιμές τους. Όμως υπάρχουν κάποια πλεονεκτήματα τα οποία είναι :

1. Μας παρέχεται η δυνατότητα να επεκτείνουμε το διαθέσιμο χώρο αποθήκευσης και να πληρώνουμε μόνο αυτό που χρησιμοποιούμε . Ομοίως, μπορούμε να τον μειώσουμε. Με αυτόν τον τρόπο μειώνουμε το κόστος. Αυτή η λύση είναι αποδοτική (οικονομικά) για μεγάλο όγκο δεδομένων, τυπικά μεγαλύτερο από 500 GB. Επίσης μπορούμε να χρησιμοποιούμε αυτήν την υπηρεσία για λήψη μεγάλων αντιγράφων ασφαλείας για κρίσιμα αρχεία.
2. Δεν έχουμε να συντηρούμε το υλικό. Όταν αποθηκεύουμε δεδομένα τοπικά έχουμε επιφορτιστεί και με την συντήρηση και επίβλεψη των δίσκων. Σε αντίθεση, το υπολογιστικό νέφος μας απαλλάσσει από αυτήν την εργασία.

³ <https://cloud.google.com/products/cloud-storage>

3. Η υπηρεσία αποθήκευσης στο υπολογιστικό νέφος μας παρέχει σύστημα ανάκτησης δεδομένων από καταστροφή. Με αυτόν τον τρόπο μπορούμε να ανακτήσουμε τα διαγραμμένα αρχεία ή καταλόγους. Ο πάροχος δημιουργεί αντίγραφα ασφαλείας και επαναφέρει το σύστημα αρχείων όποτε απαιτείται. Δεν απαιτείται να πληρώσουμε για αυτή τη λειτουργία κάποιον. Δεν έχουμε την ευθύνη της συντήρησης του συστήματος αποθήκευσης. Τα έχει αναλάβει όλα ο πάροχος .

Παρόλα αυτά, υπάρχουν και μερικά μειονεκτήματα του μοντέλου αποθήκευσης ως υπηρεσία στο υπολογιστικό νέφος τα οποία είναι:

1. Είμαστε εξαρτημένοι από το ίντερνετ καθώς όπως φαίνεται και στο εικόνα 2.2 είναι ο μόνος τρόπος πρόσβασης στα δεδομένα μας. Αν πέσει η σύνδεση δεν θα έχουμε πρόσβαση σε αυτά.
 2. Προκύπτει θέμα απόδοσης. Αν συγκρίνουμε με την αποθήκευση στις εγκαταστάσεις του οργανισμού/εταιρείας ή ιδιώτη σε δικούς τους δίσκους, τότε η απόδοση δεν είναι ίδια με αυτήν που παρέχεται στην υπηρεσία αποθήκευσης. Η απόδοση είναι συνήθως στο μισό αν το συγκρίνουμε με την ταχύτητα του τοπικού δικτύου και την ταχύτητα του ίντερνετ.
 3. Το κόστος της υπηρεσίας αποθήκευσης είναι απαγορευτικό σε σχέση με την τοπική αποθήκευση. Όπως προείπαμε, το κόστος των δίσκων είναι πολύ μικρό και συνεχώς μειώνεται. Στην περίπτωση που όλοι οι υπάλληλοι βρίσκονται στο ίδιο κτήριο, το όφελος της υπηρεσίας αποθήκευσης σε υπολογιστικό νέφος σε σχέση με την τοπική αποθήκευση δεν υφίσταται.
2. Βάση δεδομένων ως υπηρεσίας (Database as a service). Μας επιτρέπει την πρόσβαση σε enterprise βάσεις δεδομένων μέσω ίντερνετ. Η υπηρεσία αυτή παρέχει τη δυνατότητα της απομακρυσμένης φιλοξενίας βάσης δεδομένων η οποία μοιράζεται με άλλους χρήστες και λειτουργεί σαν να ήταν τοπική βάση δεδομένων. Μπορούμε να δημιουργήσουμε πίνακες, να φορτώσουμε δεδομένα, να έχουμε πρόσβαση στα δεδομένα και όλα αυτά μέσω μιας διεπαφής του παροχέα.

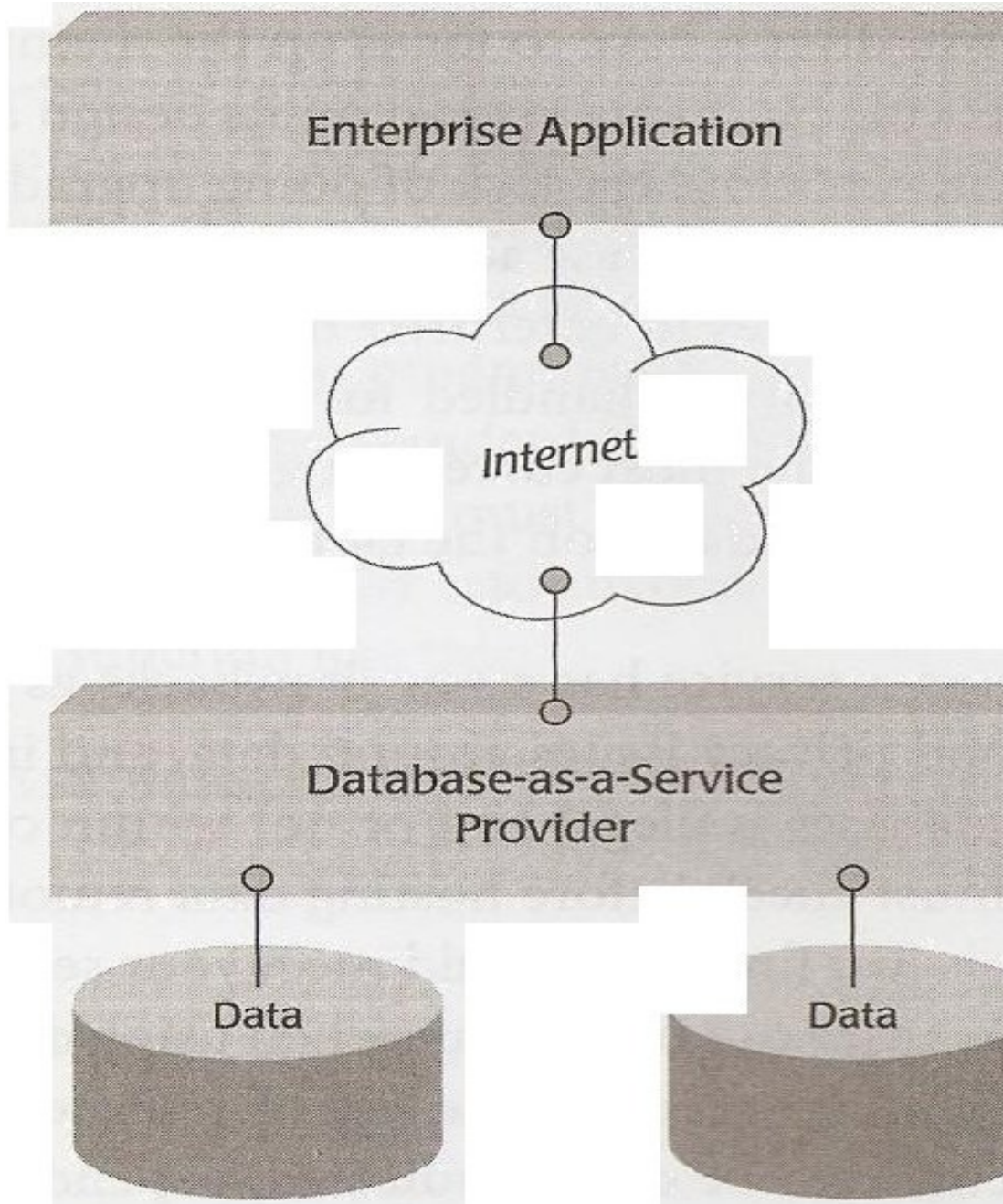
Όπως το μοντέλο αποθήκευσης ως υπηρεσία έτσι και στο μοντέλο βάσεων δεδομένων ως υπηρεσία μας παρέχεται απομακρυσμένη πρόσβαση στους πόρους και γλιτώνουμε το κόστος συντήρησης, τις άδειες λογισμικού και το κόστος του υλικού. Με την δυνατότητα αυτοεξυπηρέτησης καθίσταται ευκολότερη η παροχή μιας βάσης δεδομένων όποτε αυτή χρειάζεται. Για παράδειγμα, μπορούμε να έχουμε μια νέα βάση δεδομένων στις 10:00 και να μπει σε λειτουργία το μεσημέρι της ίδιας μέρας χωρίς να απαιτείται να αγοράσουμε νέο υλικό ή λογισμικό και χωρίς να φύγουμε από τον χώρο εργασίας μας.

Οι λειτουργίες των βάσεων δεδομένων που παρέχονται ως υπηρεσία είναι ακριβώς ίδιες με αυτές που έχουμε στις τοπικές. Ένας αποτελεσματικός παροχέας βάσεων δεδομένων ως υπηρεσία θα πρέπει να προσφέρει υπηρεσίες βάσεων δεδομένων που υπάρχουν και στις τοπικές, όσον αφορά τις επιδόσεις και τη λειτουργικότητα . Ωστόσο, όπως και στην υπηρεσία αποθήκευσης, υπάρχουν κάποια αντισταθμίσματα . Τα οφέλη από το μοντέλο βάσεις δεδομένων ως υπηρεσία είναι:

1. Η δυνατότητα να αποφύγουμε το κόστος του υλικού και του λογισμικού αξιοποιώντας την απομακρυσμένη βάση δεδομένων. Είναι γνωστό ότι οι άδειες βάσεων δεδομένων είναι το μεγαλύτερο μέρος από τον προϋπολογισμό ενός έργου.
2. Δεν χρειάζεται να ασχοληθούμε με την συντήρηση της βάσης δεδομένων όπως αντίγραφα ασφαλείας, ανάκτηση και διαχείριση χρηστών. Το μόνο που χρειάζεται να κάνουμε είναι ο σχεδιασμός της.
3. Με την βάση δεδομένων ως υπηρεσία γλιτώνουμε από εργασίες όπως η αναβάθμιση και επιδιόρθωση σφαλμάτων. Οι διαχειριστές βάσεων δεδομένων ξόδευσαν πολύ χρόνο ώστε να έχουν μια enterprise βάση δεδομένων χωρίς σφάλματα (bug free). Πλέον, με την χρήση βάσης δεδομένων ως υπηρεσία, ο πάροχος είναι επιφορτισμένος με αυτές τις εργασίες.

Αντίθετα, υπάρχουν και κάποια μειονεκτήματα τα οποία είναι:

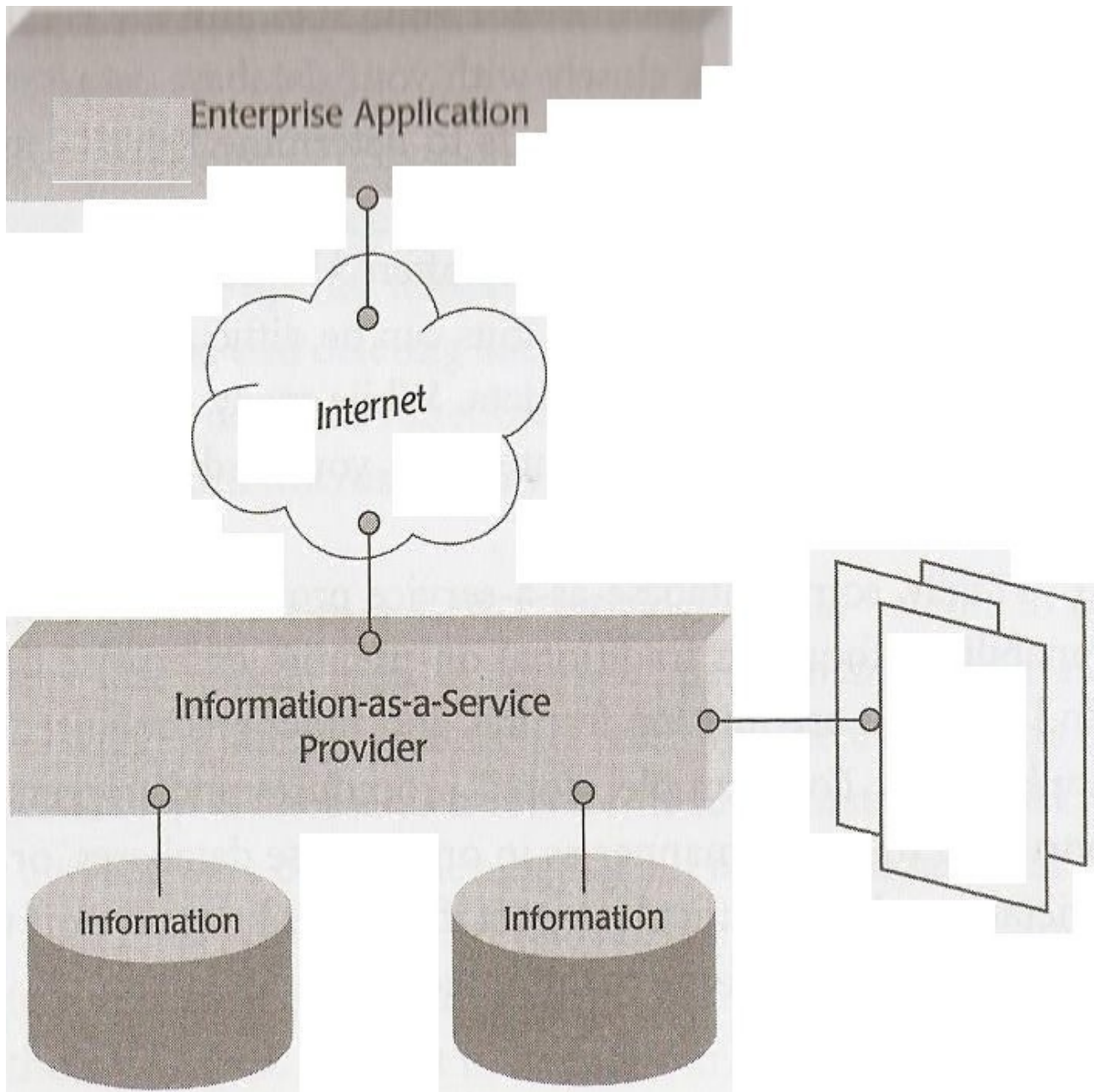
1. Υπάρχουν θέματα νομικά, συμμόρφωσης και ζητήματα απόρρητου για τα δεδομένα.
2. Η ασφάλεια των δεδομένων είναι ζήτημα όταν χρησιμοποιούμε βάσεις δεδομένων ως υπηρεσία. Όταν απαιτείτε απόλυτος έλεγχος, η χρήση απομακρυσμένης βάσης δεδομένων που δεν τελεί υπό τον απόλυτο έλεγχό μας, έτσι ανάλογα με τον τύπο των δεδομένων που θα αποθηκεύονται ενδέχεται να μην είναι κατάλληλη η χρήση αυτής της υπηρεσίας.
3. Μερικοί πάροχοι υπηρεσιών βάσεων δεδομένων προσφέρουν ένα μέρος από τις δυνατότητες που έχουν οι παραδοσιακές βάσεις δεδομένων. Ενδέχεται να μην έχουμε λειτουργίες και χαρακτηριστικά που απαιτούνται από το λογισμικό μας. Για παράδειγμα, διεργασίες αποθήκευσης (stored procedures) και triggers ενδέχεται να μην υποστηρίζονται .
4. Πολλές διεπαφές που παρέχονται από τους πάροχους υπηρεσιών βάσεων δεδομένων είναι αποκλειστικά μόνο για τη συγκεκριμένη εταιρεία και ενδέχεται να είναι δύσκολο να αξιοποιηθούν από τις υφιστάμενες εφαρμογές που πρέπει να έχουν πρόσβαση σε αυτές. Έως ότου οι πάροχοι δημιουργήσουν μια κοινή διεπαφή, ο κάθε μηχανικός λογισμικού πρέπει να καταλάβει και να μάθει το τρόπο λειτουργίας και χρήσης κάθε ένα μιας από αυτές.



Εικόνα 2.3 Αρχιτεκτονική βάση δεδομένων ως υπηρεσία (Database as a service) (Cloud Computing and SOA Convergence in Your Enterprise . Pearson Education, 2009)

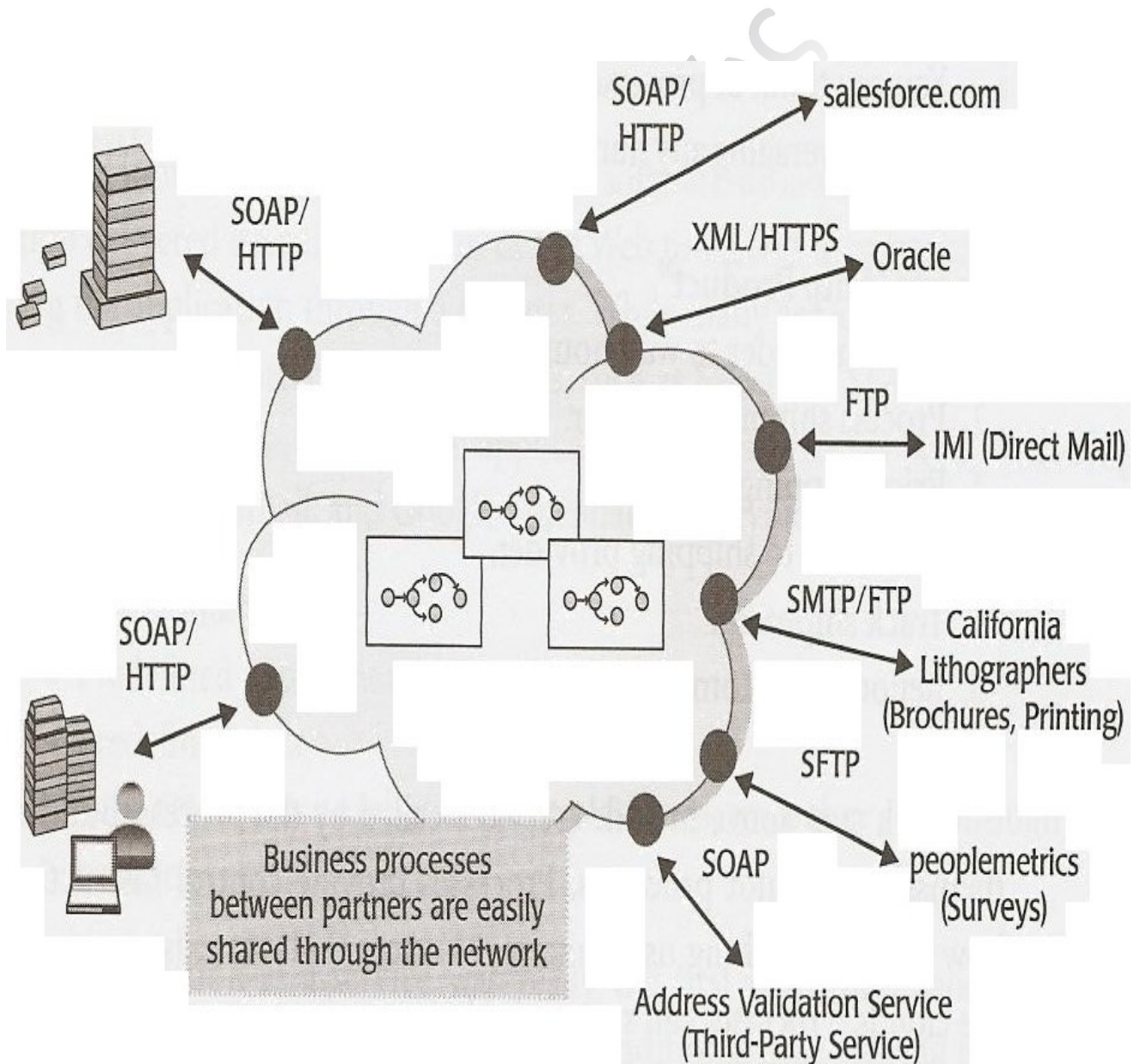
3. Πληροφορία ως υπηρεσία (Information as a service). Είναι η δυνατότητα να λαμβάνουμε απομακρυσμένα οποιαδήποτε πληροφορία μέσω μιας διασύνδεσης όπως είναι ένα API (application programming interface). Μερικά παραδείγματα από APIs είναι τα APIs των

μέσων κοινωνικής δικτύωσης όπως Twitter και Facebook.



Εικόνα 2.4 Αρχιτεκτονική πληροφορία ως υπηρεσία (Information as a service)
(Cloud Computing and SOA Convergence in Your Enterprise . Pearson Education, 2009)

4. Διεργασία ως υπηρεσία (Process as a service). Είναι ένας απομακρυσμένος πόρος ο οποίος μπορεί να δεσμεύσει πολλούς πόρους μαζί οι οποίοι είτε φιλοξενούνται στο ίδιο υπολογιστικό νέφος είτε στις εγκαταστάσεις της εταιρείας/οργανισμού ώστε να δημιουργήσουν μια επιχειρησιακή διεργασία (Εικόνα 2.4).



Εικόνα 2.5 Μοντέλο διεργασία ως υπηρεσία (Process as a service)
(Cloud Computing and SOA Convergence in Your Enterprise. Pearson Education, 2009)

5. Εφαρμογή ως υπηρεσία (Application (Software) as a service). Είναι το μοντέλο παροχής λογισμικού όπου το λογισμικό και τα σχετικά δεδομένα φιλοξενούνται σε κάποιο υπολογιστικό νέφος. Συνήθως ο τελικός χρήστης έχει πρόσβαση στην εφαρμογή χρησιμοποιώντας τον web browser. Μερικά παραδείγματα εφαρμογών ως υπηρεσία είναι το Gmail, Google Docs, Google Calendar και άλλα. Πλέον, πολλές εταιρείες διαθέτουν και enterprise λογισμικό ως υπηρεσία όπως CRM, MIS (management information system), ERP (enterprise resource planning), HRM (human resource management), CMS (content management system) και άλλα.

Το πλεονέκτημα του μοντέλου αυτού είναι η δυνατότητα που παρέχει να αξιοποιούμε λογισμικό enterprise κλίμακας χωρίς να χρειάζεται να το αγοράσουμε και να το εγκαταστήσουμε. Επιπλέον, εταιρείες όπως η Google παρέχει λογισμικό όπως email, επεξεργαστή κειμένου, λογιστικά φύλλα και άλλα δωρεάν. Άλλες εταιρείες παρέχουν εφαρμογές με μια μικρή συνδρομή όπως η Salesforce.com.

6. Πλατφόρμα ως υπηρεσία (Platform as a service). Το μοντέλο αυτό είναι μια ολοκληρωμένη πλατφόρμα που περιλαμβάνει ανάπτυξη εφαρμογών, ανάπτυξη διεπαφών, αποθήκευση, ανάπτυξη βάσεων δεδομένων και περιβάλλον για δοκιμές. Όλα αυτά παρέχονται μέσω μιας απομακρυσμένης πλατφόρμας φιλοξενίας στους συνδρομητές. Το μοντέλο πλατφόρμα ως υπηρεσία παρέχει αυτόνομες πλατφόρμες με οτιδήποτε χρειάζομαστε για την ανάπτυξη και τη φιλοξενία των εφαρμογών. Μερικές δημοφιλείς πλατφόρμες είναι το Force.com⁴ της Salesforce.com, το Google App Engine⁵ της Google, το Windows Azure Cloud Services⁶ της Microsoft, το OpenShift⁷ της Red Hat, το Oracle Cloud⁸ της Oracle και άλλα. Το μοντέλο αυτό αποτελείται από πέντε κύρια συστατικά, τα οποία είναι:

1. Σχεδιασμός. Είναι η δυνατότητα σχεδιασμού εφαρμογών και διεπαφών χρήστη.
2. Ανάπτυξη. Είναι η δυνατότητα να σχεδιάζουμε, να αναπτύσσουμε και να δοκιμάζουμε εφαρμογές κατά απαίτηση, χρησιμοποιώντας εργαλεία τα οποία παρέχονται από την πλατφόρμα κατά απαίτηση. Για παράδειγμα η Salesforce.com με την γλώσσα APEX παρέχει αυτές τις υπηρεσίες.
3. Εγκατάσταση (Deployment). Είναι η δυνατότητα να δοκιμάζουμε ή και να διαθέτουμε τις εφαρμογές που αναπτύξαμε. Οι εφαρμογές που φιλοξενούνται στο υπολογιστικό νέφος είναι προσβάσιμες μέσω περιηγητή ιστού ή υπηρεσιών ιστού.
4. Ολοκλήρωση. Είναι η δυνατότητα της ολοκλήρωσης των εφαρμογών που είναι αναπτυγμένες στον πάροχο της πλατφόρμας ως υπηρεσία με εφαρμογές που υπάρχουν εντός του οργανισμού/εταιρείας ή ακόμη και εφαρμογές που παρέχονται ως υπηρεσία από κάποιον πάροχο υπολογιστικών νεφών.
5. Λειτουργίες/Εργασίες. Είναι η ικανότητα να εκτελείτε μια εφαρμογή που ασχολείται για μεγάλο χρονικό διάστημα με εργασίες όπως λήψη αντιγράφων

⁴ <http://www.force.com/>

⁵ <https://appengine.google.com/>

⁶ <http://www.windowsazure.com/en-us/home/scenarios/cloud-services/>

⁷ <https://www.openshift.com/products>

⁸ <https://cloud.oracle.com/mycloud/f?p=service:home:0>

ασφαλείας, επαναφορά, χειρισμός εξαιρέσεων και άλλα πράγματα που προσθέτουν αξία στις λειτουργίες.

6. Αποθήκευση. Είναι η ικανότητα να παρέχουμε σε μια εφαρμογή η δυνατότητα αποθήκευσης δεδομένων στο υπολογιστικό νέφος είτε σε σύστημα αρχείων όπως το Google Cloud Storage⁹ της Google, το Amazon Simple Storage Service (Amazon S3)¹⁰, Amazon Elastic Block Store (EBS)¹¹, Big Data¹² της Amazon είτε σε βάσεις δεδομένων όπως το Amazon Relational Database Service¹³ της Amazon, το Google BigQuery¹⁴ της Google, το SQL Azure Database¹⁵ της Microsoft και άλλα.

Το πλεονέκτημα του μοντέλου αυτού είναι ότι μπορούμε να έχουμε πρόσβαση σε ένα ολοκληρωμένο περιβάλλον enterprise επιπέδου, με μικρό κόστος, έχοντας τη δυνατότητα να αναπτύξουμε enterprise εφαρμογές από το επίπεδο των δεδομένων έως αυτό της διεπαφής χρήστη. Αντίθετα, μειονέκτημα αποτελεί το γεγονός ότι κάποιιοι από τους παροχείς υπηρεσιών πλατφόρμας έχουν δικές τους γλώσσες και διεπαφές με αποτέλεσμα να είναι δύσκολη η μεταφερσιμότητα της εφαρμογής και εκτός υπολογιστικού νέφους αλλά και σε άλλη πλατφόρμα υπολογιστικού νέφους.

7. Ολοκλήρωση ως υπηρεσία (Integration as a service). Είναι η δυνατότητα της παροχής μιας πλήρους στοίβας ολοκλήρωσης από τον υπολογιστικό νέφος. Η στοίβα μπορεί να περιέχει διασυνδέσεις με εφαρμογές, σημασιολογική διαμεσολάβηση, έλεγχο ροής και σχεδιασμό ολοκλήρωσης. Στην ουσία το μοντέλο αυτό μας παρέχει τα περισσότερα από τα χαρακτηριστικά και λειτουργίες που βρίσκουμε στις παραδοσιακές EAI (Enterprise Application Intergration) τεχνολογίες με τη διαφορά ότι τώρα μέσω του υπολογιστικού νέφους μπορούν και παρέχονται ως υπηρεσία. Η ολοκλήρωση είναι δύσκολο ένα πρόβλημα για να επιλυθεί, η ολοκλήρωση κατά απαίτηση το κάνει ακόμη δυσκολότερο. Οι βασικές λειτουργίες ολοκλήρωσης είτε σε υπολογιστικό νέφος είτε όχι είναι:
 1. Μετατροπή. Σε αυτή τη λειτουργία μετατρέπουμε την σημασιολογική πληροφορία από το ένα σύστημα σε σημασιολογική πληροφορία του άλλου συστήματος ώστε το τελευταίο σύστημα να μπορεί να δεχτεί την πληροφορία στην μορφή που μπορεί να καταλάβει.
 2. Δρομολόγηση. Σε αυτή τη λειτουργία η πληροφορία δρομολογείτε στο σωστό σύστημα.
 3. Διασύνδεση. Σε αυτή τη λειτουργία μπορούμε να συνδεθούμε είτε στο αρχικό σύστημα (source) είτε στο σύστημα προορισμού (target) χρησιμοποιώντας οποιαδήποτε διασύνδεση μας παρέχουν.
 4. Καταγραφή (Logging). Με αυτή τη λειτουργία μπορούμε να καταγράψουμε όλες τις δραστηριότητες ολοκλήρωσης, όπως μηνύματα ροής εισόδου/εξόδου και άλλα συμβάντα.

⁹ <https://cloud.google.com/products/cloud-storage>

¹⁰ <http://aws.amazon.com/s3/>

¹¹ <http://aws.amazon.com/ebs/>

¹² <http://aws.amazon.com/big-data/>

¹³ <http://aws.amazon.com/rds/>

¹⁴ <https://cloud.google.com/products/big-query>

¹⁵ <http://www.windowsazure.com/en-us/services/data-management/>

Το πλεονέκτημα του μοντέλου ολοκλήρωσης ως υπηρεσία είναι ότι μπορούμε να έχουμε πρόσβαση σε αρκετά ακριβείς λειτουργικότητας λογισμικό ολοκλήρωσης για την τιμή που έχουμε μισθώσει τους πόρους του υπολογιστικού νέφους.

Αντίθετα, μειονέκτημα αποτελεί το γεγονός ότι μεσολαβούν πολλά τοίχοι προστασίας (firewalls) τα οποία χρήζουν αντιμετώπισης καθώς ενδέχεται να σταθούν ως εμπόδιο στην διαδικασία της ολοκλήρωσης. Για παράδειγμα πολλά συστήματα μπορεί να μην μας επιτρέπουν τη χρήση της πόρτας 80 η οποία χρησιμοποιείτε κατά κόρον από τις υπηρεσίες ιστού (web services). Σε αυτή την περίπτωση δεν είναι εφικτό να επικοινωνήσουμε στον απομακρυσμένο εξυπηρετητή ολοκλήρωσης που βρίσκεται εκτός του τοίχους προστασίας του οργανισμού/εταιρείας.

8. Ασφάλεια ως υπηρεσία (Security as a service). Είναι η δυνατότητα παροχής υπηρεσιών ασφάλειας απομακρυσμένα (remotely) μέσω του ίντερνετ. Οι υπηρεσίες ασφάλειας που παρέχονται σήμερα είναι στοιχειώδης. Πιο εξελιγμένες υπηρεσίες όπως η διαχείριση ταυτότητας γίνονται διαθέσιμες μέσω του μοντέλου αυτού. Το μοντέλο ασφάλειας ως υπηρεσία περιλαμβάνει εφαρμογές όπως αντιβιοτικά που παρέχονται μέσω ίντερνετ. Παράδειγμα τέτοιας υπηρεσίας είναι το *Panta Cloud anti-virus*¹⁶. Το μοντέλο αυτό μπορεί να παρέχει και υπηρεσίες διαχείρισης ασφάλειας εσωτερικά σε έναν οργανισμό/εταιρεία μέσω ίντερνετ. Τα πλεονεκτήματα του μοντέλου ασφάλειας ως υπηρεσία είναι :
 1. Συνεχής ενημέρωση του λογισμικού καταπολέμησης ιών χωρίς κάποια ενέργεια του χρήστη.
 2. Μεγαλύτερη εξειδίκευση στον τομέα ασφάλειας από ότι συνήθως συναντάμε σήμερα σε κάποιον οργανισμό/επιχείρηση, μιας και η υπηρεσία αυτή παρέχεται από εξειδικευμένες εταιρείες του είδους.
 3. Η εξωτερική ανάθεση των εργασιών ασφάλειας απαλλάσσει τον οργανισμό/εταιρεία από αυτές τις εργασίες, έτσι μπορεί να εξοικονομήσει χρόνο και χρήμα ώστε να αφοσιωθεί περισσότερο στο αντικείμενο που ειδικεύεται.
9. Διαχείριση/Διακυβέρνηση ως υπηρεσία (Managment/governance as a service). Είναι κάθε υπηρεσία κατά απαίτηση η οποία παρέχει την δυνατότητα να διαχειρίζεται μια ή περισσότερες υπηρεσίες υπολογιστικού νέφους, όπως τοπολογία, χρησιμοποίηση πόρων, virtualization κτλ. Τα συστήματα διακυβέρνησης έχουν τη δυνατότητα να επιβάλλουν καθορισμένες πολιτικές στα δεδομένα και στις υπηρεσίες. Αυτή η πτυχή του υπολογιστικού νέφους αργεί ακόμη να υιοθετηθεί όμως είναι απαραίτητη για να θεωρείτε επιτυχές το υπολογιστικό νέφος. Για παράδειγμα, αν έχουμε χιλιάδες υπηρεσίες και δεδομένα υπό τον έλεγχό μας, πρέπει να ελέγχουμε τον τρόπο πρόσβασης, ενημέρωσης, διαγραφής και εισαγωγής. Χρειαζόμαστε μια προσέγγιση, διεργασίες και τεχνολογία. Εν ολίγοις χρειαζόμαστε τη διακυβέρνηση .

¹⁶ https://shop.pandasecurity.com/cgi-bin/pp/ml=GR?id=A12CAESD1&track=51899&actp=1&nav=pp_homeuser_pcsecurity_A12CAESD1

10. Δοκιμή ως υπηρεσία (Testing as a service). Είναι η δυνατότητα δοκιμών σε τοπικά συστήματα ή συστήματα που παρέχονται από το υπολογιστικό νέφος με τη χρήση λογισμικού και υπηρεσιών που φιλοξενούνται στο υπολογιστικό νέφος.

Το πλεονέκτημα του μοντέλου δοκιμών ως υπηρεσία είναι ότι μας απαλλάσσει από έξοδα αγοράς εξυπηρετητή δοκιμών και άδειες λογισμικού.

11. Υποδομές ως υπηρεσία (Infrastructure as a service). Είναι η δυνατότητα απομακρυσμένης πρόσβασης σε πόρους. Στην ουσία μισθώνουμε έναν εξυπηρετητή. Ο εξυπηρετητής αυτό μπορεί να είναι είτε φυσικό είτε εικονικός. Η διαφορά σε σχέση με την απομακρυσμένη πρόσβαση σε ένα παραδοσιακό εξυπηρετητή είναι ότι ο πάροχος μας παρέχει μια διεπαφή χρήστη και μερικές υπηρεσίες που μπορούμε να αποκτήσουμε πρόσβαση στο μηχάνημα και στο λογισμικό αυτού. Παράδειγμα υποδομών ως υπηρεσία είναι το Amazon EC2¹⁷ από την Amazon, Google Compute Engine¹⁸ από την Google, το Windows Azure Virtual Machines¹⁹ από την Microsoft, το Rackspace Cloud Servers²⁰ από την Rackspace²¹ και άλλα. Η υπηρεσία αυτή είναι υπερσύνολο όλων των προηγούμενων που έχουμε αναφέρει. Μπορεί να μας παρέχει βάσεις δεδομένων, εφαρμογές, ασφάλεια, διακυβέρνηση κτλ. Οτιδήποτε μπορεί να υπάρξει σε ένα παραδοσιακό υπολογιστικό κέντρο δεδομένων μπορεί να μας δοθεί και από το μοντέλο υποδομές ως υπηρεσία.

Σε αυτό το μοντέλο η τιμολόγηση γίνεται με βάση την χρησιμοποίηση των υπολογιστικών πόρων σε σχέση με το μέγεθος που έχουμε επιλέξει. Για παράδειγμα ένας εξυπηρετητής με 4GB RAM και 2 GHz διπύρηνο επεξεργαστή για 12 ώρες λειτουργίας στοιχίζει λιγότερο για τις ίδιες ώρες χρήσης από ένα εξυπηρετητή με τη διπλάσια μνήμη και τετραπύρηνο επεξεργαστή.

Το πλεονέκτημα αυτού του μοντέλου είναι ότι μπορούμε να αποκτήσουμε πρόσβαση σε πολύ ακριβούς πόρους υπολογιστικών κέντρων δεδομένων μέσω μιας συμφωνίας ενοικίασης. Επιπλέον, κάποιος φροντίζει για εμάς ώστε να διαχειρίζεται τα φυσικά μηχανήματα, να επιλύει τα δικτυακά προβλήματα και ότι προκύπτει από βλάβες στο υλικό.

Το μειονέκτημα είναι ότι οι πόροι είναι λιγότερο επεκτάσιμοι κατά απαίτηση σε σχέση με το μοντέλο βάσεων δεδομένων ως υπηρεσία ή το αποθήκευση ως υπηρεσία που μπορούσαμε να αγοράσουμε πρόσθετη χωρητικότητα όποτε την χρειαζόμασταν. Τέλος, για την ανάπτυξη μιας εφαρμογής απαιτείτε από τον μηχανικό λογισμικού να εγκαταστήσει το λειτουργικό σύστημα στους εξυπηρετητές όπως και το απαραίτητο λογισμικό. Επιπλέον, είναι υπεύθυνος για την συντήρηση του λειτουργικού συστήματος, του λογισμικού και των μπαλωμάτων (patches) αυτών.

2.4 Πλεονεκτήματα υπολογιστικού νέφους

¹⁷ <http://aws.amazon.com/ec2/>

¹⁸ <https://cloud.google.com/products/compute-engine>

¹⁹ <http://www.windowsazure.com/en-us/services/virtual-machines/>

²⁰ <http://www.rackspace.com/cloud/servers/>

²¹

Οι περισσότεροι από αυτούς που ακούν για το υπολογιστικό νέφος και σκέφτονται να το υιοθετήσουν, το πρώτο και σημαντικότερο από τα πλεονεκτήματα που θέλουν να μάθουν είναι το μικρότερο κόστος σε σχέση με τις παραδοσιακές υλοποιήσεις. Τα πλεονεκτήματα από την υιοθέτηση του υπολογιστικού νέφους έχει να κάνει με:

1. Κόστος
2. Δίκτυο
3. Καινοτόμο
4. Επεκτάσιμο
5. Επιταχύνει την υλοποίηση
6. Φιλικό προς το περιβάλλον (green IT)

Το κόστος σημαίνει ότι το υπολογιστικό νέφος, ως αρχιτεκτονική υλοποίηση είναι τυπικά φτηνότερη από τις παραδοσιακές λύσεις ανάπτυξης/υλοποίησης σε υπολογιστικά κέντρα δεδομένων (data center). Στις παραδοσιακές λύσεις, στο κόστος λαμβάνουμε υπόψιν το κόστος του υλικού, του λογισμικού, των ανθρώπινων πόρων που απαιτούνται για την συντήρηση ενός συστήματος. Ενώ το υπολογιστικό νέφος δεν είναι πάντα ακριβότερο, θεωρητικά τουλάχιστον, είναι σίγουρα πιο αποδοτικό (οικονομικά) από τις παραδοσιακές λύσεις. Για παράδειγμα, όταν αναπτύσσουμε ένα νέο σύστημα, πρέπει να πληρώσουμε για την ανάπτυξη και πρέπει να το σχεδιάσουμε για το μέγιστο φορτίο το οποίο έχει ως αποτέλεσμα να δημιουργείται πλεονάζουσα χωρητικότητα η οποία δεν χρησιμοποιείται! Οι υπηρεσίες υπολογιστικού νέφους θα πρέπει να είναι λιγότερο ακριβές επειδή πληρώνουμε μόνο αυτό που χρησιμοποιούμε. Ένα άλλο παράδειγμα που μπορούμε να μειώσουμε το κόστος είναι όταν το λογισμικό βρίσκεται ακόμη στις πρώτες δοκιμαστικές ή αρχικές εκδόσεις. Σε αυτό το στάδιο απαιτούνται πολλές και συχνές ενημερώσεις του λογισμικού που διορθώνουν ή βελτιώνουν το λογισμικό. Όταν το λογισμικό αυτό παρέχεται ως υπηρεσία τότε χάρη στο μοντέλο του υπολογιστικού νέφους ο παροχέας του υπολογιστικού νέφους μπορεί να ενημερώσει, επιδιορθώσει ή να επανεγκαταστήσει το λογισμικό ανά πάσα στιγμή, όσες φορές κι αν χρειαστεί χωρίς να ενοχληθεί ή να αντιληφθεί κάτι ο τελικός χρήστης.

Με την έννοια δίκτυο εννοούμε ότι το υπολογιστικό νέφος είναι στο ίντερνετ και το ίντερνετ είναι συνδεδεμένο με πολλά άλλα πράγματα που προσθέτουν αξία όπως ιστοχώρους κοινωνικής δικτύωσης, APIs και άλλα υπολογιστικά νέφη. Έτσι μπορούμε να αναμείξουμε και να συνδυάσουμε υπηρεσίες υπολογιστικών νεφών για να δώσουμε λύση σε πολλά επιχειρηματικά προβλήματα. Ένα πραγματικό όφελος του υπολογιστικού νέφους είναι η δυνατότητα των υπηρεσιών νέφους να συνδυάζονται με άλλες υπηρεσίες νέφους φτιάχνοντας προσαρμοσμένες υπηρεσίες που είναι ακόμα πιο ισχυρές από το άθροισμα οποιαδήποτε τμήμα τους.

Με την έννοια καινοτόμο ευνοούμε ότι το υπολογιστικό νέφος και οι υπηρεσίες του που παρέχονται σήμερα είναι καινούριες, μοντέρνες και καινοτόμες. Αυτό θα συνεχίσει να έχει πολλά καινοτόμα χαρακτηριστικά που παρέχουν μεγάλη αξία στα χρήματα που ενδύονται σε αυτό. Οι εταιρίες που αξιοποιούν το υπολογιστικό νέφος, ιδιαίτερα αυτές που βρίσκονται στο ξεκίνημα θα διαπιστώσουν ότι η χρήση του υπολογιστικού νέφους προσθέτει αξία στην IT καινοτομία που δημιούργησαν.

Με την έννοια επεκτασιμότητα εννοούμε μπορούμε να προσθέσουμε περισσότερη χωρητικότητα αν χρειαστεί και όποτε χρειαστεί αυτό από τις αυξανόμενες ανάγκες. Επίσης, μπορούμε πολύ εύκολα να μειώσουμε την χωρητικότητα όταν δεν συντρέχουν αυτές οι αυξημένες ανάγκες. Για την επίτευξη της επεκτασιμότητας δεν χρειάζεται να εγκαταστήσουμε νέο υλικό και λογισμικό ώστε να αυξήσουμε τους διαθέσιμους πόρους. Το μόνο που χρειάζεται είναι μερικά κλικ.

Η επιτάχυνση της υλοποίησης είναι πλεονέκτημα διότι με το υπολογιστικό νέφος γλιτώνουμε αρκετές μέρες ίσως και ώρες που απαιτούνταν με το παραδοσιακό τρόπο υλοποίησης. Δε χρειάζεται πλέον να αγοράσουμε και να εγκαταστήσουμε το υλικό, να εγκαταστήσουμε το λειτουργικό σύστημα ή να ρυθμίσουμε διακαιώματα ώστε να πάρουμε πρόσβαση σε ένα τμήμα του κέντρου δεδομένων. Απλά εγγραφόμαστε σε μια υπηρεσία υπολογιστικού νέφους και έπειτα με μερικά κλικ αποκτούμε πρόσβαση στους πόρους του υπολογιστικού νέφους που χρειαζόμαστε .

Το υπολογιστικό νέφος θεωρείται φιλικό προς το περιβάλλον γιατί πλέον όλα τα “πεινασμένα” σε ηλεκτρική ενέργεια κέντρα δεδομένων που διαθέτει κάθε εταιρεία μπορούν να κλείσουν και στην θέση τους να μπουν τα υπολογιστικά νέφη όπου οι διαθέσιμοι πόροι διαμοιράζονται μέσω ίντερνετ προς όλους τους πελάτες. Σε αντίθεση με την υφιστάμενη κατάσταση που ο κάθε ένας τους διαθέτει από ένα τουλάχιστον αυτόνομο κέντρο δεδομένων. Επιπλέον, όλοι οι μεγάλοι πάροχοι υπολογιστικών νεφών, όπως Google, Amazon, Microsoft, Oracle, Hewlett-Packard έχουν μεριμνήσει ώστε οι εγκαταστάσεις που φιλοξενούν τα τεράστια κέντρα δεδομένων να είναι ιδικά σχεδιασμένες (βιοκλιματικός σχεδιασμός κτιρίων) ώστε να εξοικονομούν ενέργεια και να καταναλώνουν την ελάχιστη δυνατή. Επίσης έχει σχεδιαστεί και ειδικώς τρόπος ψύξης όλου του εξοπλισμού που απαρτίζει ένα υπολογιστικό κέντρο με γνώμονα την ελάχιστη κατανάλωση ηλεκτρικής ενέργειας.

2.5 Μειονεκτήματα υπολογιστικού νέφους

Όπως όλες οι νέες τεχνολογίες έτσι το υπολογιστικό νέφος δεν έχει μόνο πλεονεκτήματα αλλά και μειονεκτήματα Τα βασικότερα μειονεκτήματα είναι:

1. Ασφάλεια
2. Έλεγχος
3. Κόστος
4. Ανοικτό (Openness)
5. Συμμόρφωση
6. Συμφωνία επιπέδου παροχής υπηρεσιών (SLA)

Με τον όρο ασφάλεια εννοούμε ότι το υπολογιστικό νέφος μας παρέχει υποδομές και πόρους οι οποίοι δεν τελούν κάτω υπό τον άμεσο έλεγχο και επίβλεψη από εμάς. Αυτό έχει την δυνατότητα να αφήσει σε μερικές περιπτώσεις εκτεθειμένα τα δεδομένα μας! Ενώ σε επίπεδο ασφάλειας παρέχεται υποστήριξη κρυπτογράφησης ονόματος χρήστη, μυστικών κωδικών και βασικών στοιχείων διαχείρισης ταυτότητας ακόμη είναι πολλοί αυτοί στις μέρες μας που δεν εμπιστεύονται και δεν θέλουν να τοποθετούν “κρατικά μυστικά” στο υπολογιστικό νέφος.

Παρόλο αυτά το υπολογιστικό νέφος πηγαίνει καλύτερο στο τομέα της ασφάλειας. Όσο περνάει ο καιρός και ωριμάζει τεχνολογικά, δεν υπάρχει λόγος να ανησυχούμε για τα δεδομένα που βρίσκονται σε κάποιο υπολογιστικό νέφος σε σχέση με το αν βρίσκονταν στις εγκαταστάσεις του εκάστοτε οργανισμού ή εταιρείας.

Ο έλεγχος σημαίνει ότι ο πάροχος του υπολογιστικού νέφους που είμαστε πελάτες, μας δίνει τον έλεγχο από τις εικονικές IT υποδομές. Αυτό σημαίνει ότι είμαστε στο “έλεος” μιας άλλης εταιρείας η οποία θα μπορούσε να προκαλέσει αρκετά προβλήματα σε αρχεία, δεδομένα, διεργασίες και άλλα τα οποία λειτουργούν στις υποδομές της. Μπορεί ακούσια να παραβιάσει κάποια πολιτική και να βρούμε τον λογαριασμό μας απενεργοποιημένο . Μια άλλη περίπτωση είναι η εταιρεία που μας παρέχει μια υπηρεσία υπολογιστικού νέφους να εξαγοραστεί από μια άλλη και αυτή με τη σειρά της να αποφασίσει ότι η συγκεκριμένη υπηρεσία που εμείς χρησιμοποιούμε δεν είναι πλέον βιώσιμη ή κερδοφόρα και να την καταργήσει. Τότε εμείς θα πρέπει να μπούμε στη διαδικασία να βρούμε γρήγορα μια άλλη λύση. Υπάρχει πάντα ένα ρίσκο όταν εξαρτάσαι από άλλες εταιρείες και δεν έχεις τον συνολικό έλεγχο.

Ένα άλλο πρόβλημα που ενδέχεται να προκύψει με τον υπολογιστικό νέφος είναι η συμμόρφωση. Όσοι έχουν μάθει να ζουν με το audit θα δυσαρεστηθούν καθώς ενδέχεται να μη τους παρέχεται από τον παροχέα υπολογιστικού νέφους χαρακτηριστικά όπως το logging και το auditing. Αυτά τα χαρακτηριστικά ενδέχεται να είναι απαραίτητα για κάποιες εφαρμογές για λόγους νομικής συμμόρφωσης. Η τάση πάντως είναι οι πάροχοι υπολογιστικού νέφους να κάνουν το καλύτερο δυνατό για να καλύψουν αυτό το κενό. Επομένως απαιτείται προσοχή στην επιλογή του παροχέα σε περίπτωση που αυτά τα χαρακτηριστικά είναι κρίσιμα για το σύστημα που αναπτύσσουμε .

Openness σημαίνει ότι πολλές πλατφόρμες υπολογιστικών νεφών είναι τεχνολογικά “κλειστές” κατά κύριο λόγο. Όταν αναπτύσσουμε ένα σύστημα το οποίο είναι γραμμένο στην γλώσσα του παροχέα και βασίζεται στην αρχιτεκτονική του τότε δημιουργείτε ένα κόστος το οποίο θα είναι απαγορευτικό στην περίπτωση που θελήσουμε να μεταβούμε σε πλατφόρμα άλλου παροχέα ή να μεταφερθεί πίσω στις εγκαταστάσεις και υποδομές του οργανισμού ή της εταιρείας. Ενώ οι πάροχοι υπολογιστικών νεφών εργάζονται όσο γρήγορα μπορούν για να παρέχουν ένα σετ από πρότυπα τα οποία θα μειώσουν το ρίσκο του μεγάλου κόστους μεταφερσιμότητας, η εξοικονόμηση χρημάτων έχει καθοριστικό ρόλο στην απόφαση για το αν θα χρησιμοποιήσουμε τα πρότυπα του υπολογιστικού νέφους ή όχι.

Προηγουμένως αναφέραμε το κόστος ως πλεονέκτημα των υπολογιστικών νεφών. Υπάρχουν όμως πολλές περιπτώσεις όπου το υπολογιστικό νέφος δεν είναι καθόλου οικονομικά αποδοτικό. Μερικές φορές, για κάποιες εφαρμογές η φιλοξενία τους σε κάποιο υπολογιστικό νέφος είναι πιο ακριβή αν υπολογίσουμε και το κόστος φορητότητας. Κάποιες άλλες εφαρμογές μπορεί να έχουν κάποια ιδικά χαρακτηριστικά που όταν εκτελούνται από τις υποδομές του εκάστοτε οργανισμού ή εταιρείας (on-premises) να είναι λιγότερο ακριβά σε σχέση με το κόστος σε κάποιο υπολογιστικό νέφος. Σε κάθε περίπτωση πριν λάβουμε κάποια απόφαση για το κόστος θα πρέπει να σκεφτούμε και να υπολογίσουμε την απόδοση της επένδυσης (ROI) σε υπολογιστικό νέφος ή όχι.

Η συμφωνία επιπέδου παροχής υπηρεσιών αναφέρει τις υπηρεσίες, τις προτεραιότητες, τις ευθύνες και τις εγγυήσεις μεταξύ του παροχέα υπολογιστικού νέφους και του χρήστη του νέφους. Πολλοί πάροχοι υπολογιστικών νεφών δεν παρέχουν συμβόλαια επιπέδου παροχής υπηρεσιών. Η τάση όμως είναι οι πάροχοι να προσφέρουν SLAs.

2.6 Πότε το υπολογιστικό νέφος είναι κατάλληλο

1. Όταν οι διεργασίες, οι εφαρμογές και τα δεδομένα είναι ανεξάρτητα ή όταν δεν είναι στενά συνδεδεμένα (χαλαρή σύζευξη) με άλλες εφαρμογές ή δεδομένα. Όταν είναι στενά συνδεδεμένα είναι δύσκολο αν όχι αδύνατο, να αποσυνδεθούν και επομένως δεν θα λειτουργούν ανεξάρτητα σε μια απομακρυσμένη πλατφόρμα. Αν υπάρχει χαλαρή συνδεσιμότητα οι εφαρμογές είναι κατάλληλες για το υπολογιστικό νέφος.
2. Όταν τα σημεία της ολοκλήρωσης είναι καλά ορισμένα ή όταν έχουν οριστεί τα σημεία που η εφαρμογή θα ανταλλάσσει δεδομένα ή διεργασίες. Σε αυτήν την περίπτωση είναι εύκολο να ενσωματωθεί με εφαρμογές της επιχείρισης/οργανισμού.
3. Όταν μας αρκεί το κατώτατο επίπεδο ασφάλειας ή όταν οι πληροφορίες που θα περιέχονται στο περιβάλλον του υπολογιστικού νέφους απαιτούν χαμηλό επίπεδο ασφάλειας. Το υπολογιστικό νέφος τυπικά παρέχει καλό επίπεδο ασφάλειας αλλά όπως είπαμε και στα μειονεκτήματά του, δεν είναι έτοιμα να δεχτεί “κρατικά μυστικά”.
4. Όταν επιθυμούμε εφαρμογές ίντερνετ. Σήμερα με το έλεση των πλούσιων εφαρμογών ίντερνετ (RIAs) έχουμε εφαρμογές που εκτελούνται από τον browser και η εμφάνισή τους αλλά και η λειτουργικότητά τους είναι όμοια με τις παραδοσιακές τοπικές εφαρμογές.
5. Όταν ο πυρήνας της εσωτερικής αρχιτεκτονικής στην εταιρεία/οργανισμό είναι σωστά δομημένος τότε είναι ευκολότερο το υπολογιστικό νέφος να γίνει μέρος αυτής της αρχιτεκτονικής .
6. Όταν το κόστος είναι θέμα ή όταν υπάρχει ένα ξεκάθαρο όφελος κόστους με τη χρήση υπολογιστικού νέφους όπως αναφέραμε στα πλεονεκτήματά του. Στην περίπτωση που ζητούμενο είναι η ανάπτυξης μιας νέας εφαρμογής που θα κοστίσει λίγο (φτηνή) με χαμηλό κόστος λειτουργίας τότε το υπολογιστικό νέφος είναι μονόδρομος.
7. Τέλος όταν η εφαρμογή είναι νέα. Όταν είναι νέα μια εφαρμογή είναι ευκολότερη η εκτέλεσή της από κάποια πλατφόρμα υπολογιστικού νέφους σε σχέση με την μεταφορά μια υφιστάμενης εφαρμογής.

2.7 Πότε να αποφεύγουμε το υπολογιστικό νέφος

1. Όταν οι διεργασίες, οι εφαρμογές και τα δεδομένα είναι στενά συνδεδεμένα (μεγάλη σύζευξη). Αν οι εφαρμογές είναι αλληλοεξαρτώμενες δεν είναι καλή ιδέα να τις μεταφέρουμε σε υποδομές υπολογιστικού νέφους. Αργά ή γρήγορα θα καταρρεύσει. Γενικά όταν υπάρχει χαμηλή σύζευξη τότε συνίσταται η μετάβαση σε διαφορετική περίπτωση δεν συνίσταται .

2. Όταν το σημείο ολοκλήρωσης δεν είναι καλά ορισμένο ή όταν δεν υπάρχουν καλοί μηχανισμοί για να συγχρονίσουν τις πληροφορίες και τις διεργασίες που φιλοξενούνται στον πάροχο του υπολογιστικού νέφους με τα υπάρχον συστήματα της εταιρείας/οργανισμού. Η μετάβαση σε υπολογιστικό νέφος τέτοιων συστημάτων έχει πολύ ρίσκο.
3. Όταν έχετε ως ζητούμενο ένα υψηλό επίπεδο ασφάλειας ή όταν διακρίνεται ότι ο τομέας της ασφάλειας είναι επισφαλής όταν δεν μπορείτε να εμπιστευτείτε συστήματα στα οποία δεν οποία δεν έχετε πλήρη έλεγχο. Σε αυτήν την περίπτωση το υπολογιστικό νέφος δεν ταιριάζει στην εταιρεία/ οργανισμό σας.
4. Όταν ο έλεγχος είναι κρίσιμος για την επιχείρηση/οργανισμό. Αν δεν θέλετε να δώσετε σε κάποιον που δεν εμπιστεύεστε 100% με εξωτερική ανάθεση ένα κρίσιμο για σας συστατικό, τότε το υπολογιστικό νέφος δεν αποτελεί καλή λύση για το πρόβλημά σας.
5. Όταν ο πυρήνας της εσωτερικής αρχιτεκτονικής της εταιρείας/ οργανισμού δεν είναι σωστός, παρουσιάζει δυσλειτουργίες και απαιτείτε δουλειά για να διορθωθεί τότε δεν πρέπει να αποφύγουμε την λύση του υπολογιστικού νέφους έως ότου τακτοποιήσουμε την εσωτερική αρχιτεκτονική .
6. Όταν η εφαρμογή απαιτεί μια τοπική διεπαφή χρήστη όπως win32, win-forms, gtk+ κτλ.
7. Όταν οι εφαρμογές είναι παλαιού τύπου (legacy).

Κεφάλαιο 3

3.1 Τι είναι το TAXISnet;

Το TAXISnet είναι η υπηρεσία που παρέχει στους φορολογούμενους τη δυνατότητα να διευθετούν τις εκκρεμότητές τους με τις ΔΟΥ μέσω Internet, χωρίς να απαιτείται η επίσκεψη στις εκάστοτε ΔΟΥ. Είναι το πρώτο πληροφοριακό σύστημα στην χώρα μας για την εξυπηρέτηση των φορολογικών συναλλαγών φυσικών και νομικών προσώπων με ηλεκτρονικό τρόπο. Το σύστημα τέθηκε σε λειτουργία για πρώτη φορά το 2001 και έδινε την δυνατότητα στους φορολογούμενους να υποβάλλουν ηλεκτρονικά την φορολογική δήλωση εισοδήματος για τη χρήση του 2000 και την υποβολή δηλώσεων ΦΠΑ διμηνιαίες και ετήσιες.

3.2 Στόχος του νέου TAXISnet

Από το 2001 το TAXISnet έχει εμπλουτιστεί με νέες υπηρεσίες προς του πολίτες, τις επιχειρήσεις αλλά και την δημόσια διοίκηση. Το νέο TAXISnet έχει στόχο να επεκτείνει, να βελτιώσει και να κάνει φιλικότερες και περισσότερο χρήσιμες τις υπάρχουσες ηλεκτρονικές υπηρεσίες. Βασικός στόχος του νέου συστήματος, που εφαρμόζει στην πράξη τις απαιτήσεις της πρόσφατης νομοθεσίας, είναι η διευκόλυνση των φορολογουμένων πολιτών προκειμένου να αντεπεξέρχονται στις φορολογικές τους υποχρεώσεις κατά το δυνατόν ευκολότερα, γρηγορότερα και ορθότερα. Επιπλέον, θα συμβάλλει στην βελτίωση της απόδοσης των υπαλλήλων του υπουργείου οικονομικών αλλά και στην ενίσχυση της δημόσιας διοίκησης και τον έλεγχο των περιουσιακών και οικονομικών στοιχείων των φορολογουμένων. Βασικό του πλεονέκτημα, είναι η τακτοποίηση και η οριοθέτηση της σχέσης μεταξύ φορολογουμένων και λογιστών, στην περίπτωση των πολιτών που επιλέγουν να εμπιστευθούν σε επαγγελματία τη διεύθυνση των υποχρεώσεών τους.

Με το νέο σύστημα, ο λογιστής σας ΔΕΝ απαιτείται να γνωρίζει τους κωδικούς πρόσβασης που επιλέξατε ή τον κλειδί αριθμο που εκδίδεται αποκλειστικά για σας κατά την εγγραφή σας. Μετά την εξουσιοδότησή σας και με την προϋπόθεση ότι θα την αποδεχθεί θα μπορεί να ενεργεί για λογαριασμό σας αυστηρά στα αντικείμενα για τα οποία τον εξουσιοδοτήσατε .

Στην πλήρη του ανάπτυξη, το νέο σύστημα, ενδεικτικά:

1. Θα καλύπτει την εμπρόθεσμη ή εκπρόθεσμη υποβολή αρχικών ή συμπληρωματικών - τροποποιητικών δηλώσεων που σήμερα υποβάλλονται είτε ηλεκτρονικά είτε στη Δ.Ο.Υ. καθώς και τη χορήγηση των περισσότερων από τα υφιστάμενα πιστοποιητικά και βεβαιώσεις.
2. Θα παρέχει προσωποποιημένες πληροφορίες και θα υπενθυμίζει στους χρήστες τις φορολογικές τους υποχρεώσεις.
3. Θα δίνει στο χρήστη τη δυνατότητα να εξουσιοδοτεί έναν ή περισσότερους λογιστές που θα ενεργούν για λογαριασμό του σε μία ή περισσότερες από τις υπηρεσίες που σταδιακά θα προστίθενται στο νέο σύστημα.
4. Θα επιτρέπει τη μεταβολή προσωπικών στοιχείων μητρώου ή στοιχείων επιχείρησης, χωρίς να είναι υποχρεωτική η προσέλευση στη Δ.Ο.Υ.

Οι πρώτες εφαρμογές που εντάχθηκαν στο νέο σύστημα στις 6 Δεκεμβρίου, αφορούν στην εγγραφή - απόκτηση κωδικών χρήσης και στην υποβολή αρχικών εμπρόθεσμων περιοδικών δηλώσεων Φ.Π.Α. και πινάκων ενδοκοινοτικών συναλλαγών (V.I.E.S.).

3.3 Προβλήματα λειτουργίας

Από την αρχή λειτουργίας του έως και σήμερα έχουν παρατηρηθεί πολλές ώρες (downtime) που το σύστημα δεν ήταν διαθέσιμο στους χρήστες, κάτι που είναι αντίθετο με το αρχικό σχεδιασμό καθώς θα παρείχε υπηρεσίες στους φορολογούμενους εφτά ημέρες την εβδομάδα εικοσιτέσσερις ώρες την μέρα!

Ενδεικτικά, το 2010 παρουσιάστηκαν πολλά προβλήματα όπως:

1. Πρόβλημα στην έκδοση κλειδαριθμού
2. Πρόβλημα στην υποβολή φορολογικών δηλώσεων (E1, E2)
3. Πρόβλημα στην υποβολή φορολογικού εντύπου E9
4. Πρόβλημα στην υποβολή δηλώσεων ΦΠΑ
5. Πρόβλημα στην αποστολή του αποδεικτικού σημειώματος ορθής αποστολής της αίτησης εγγραφής
6. Καθυστέρηση στην έκδοση σημαντικών εγκυκλίων
7. Καθυστέρηση στην εκκαθάριση φορολογικών δηλώσεων του έτους 2009
8. Καθυστέρηση στην εκκαθάριση περίπου ενός εκατομμυρίου δηλώσεων ενιαίου τέλους ακινήτων του έτους 2009

Κατά το έτος 2012 το σύστημα δεν ήταν διαθέσιμο ή λειτουργούσε με μειωμένη ποιότητα παροχής υπηρεσιών. Το TAXISnet δεν άντεξε στο εξωφρενικά μεγάλο όγκο υποβολής αιτήσεων φόρου εισοδήματος (για χρήση του 2011), που προσπάθησαν να υποβάλουν οι φορολογούμενοι λίγες ώρες πριν εκπνεύσει η προθεσμία υποβολής τους.

3.4 Αίτια προβλημάτων

Οι βασικότεροι λόγοι για τα προβλήματα που αναφέρθηκαν ήταν οι αστοχίες στο υλικό. Παρατηρήθηκε μεγάλη συμφόρηση σε κάποιους μεταγωγής τους συστήματος. Αρκετές ήταν και πτώσεις του δικτύου. Όμως, δεν ευθύνεται μόνο το υλικό. Στο αρχικό σύστημα υπήρχαν βάσεις δεδομένων που είχαν πίνακες οι οποίοι δεν ήταν κανονικοποιημένοι. Άρα κακός σχεδιασμός του συστήματος. Άξιο αναφοράς είναι ότι την περίοδο που γίνονταν η υποβολή αιτήσεων από τους φορολογούμενους είχαν προγραμματιστεί και πραγματοποιηθεί εργασίες συντήρησης και αναβάθμισης του εξοπλισμού, κάτι που τελικά είχε αρνητικές συνέπειες για την αδιάλειπτη λειτουργία του συστήματος.

Ένας ακόμη σοβαρός λόγος για την μη διαθεσιμότητα του TAXISnet ήταν ο μεγάλος όγκος δεδομένων που είχαν και εξακολουθούν να έχουν τα έντυπα φορολογικών δηλώσεων αλλά και οι φόρμες της απογραφής των δημοσίων υπαλλήλων. Σε συνδυασμό με την αυξημένο αριθμό αιτήσεων που προσπαθούσαν να καταχωρήσουν οι πολίτες λίγες ώρες πριν εκπνεύσουν οι σχετικές προθεσμίες υποβολής. Αυτό δείχνει ότι δεν υπήρχαν οι απαιτούμενες υποδομές για να σηκώσουν το απότομο φόρτο δεδομένων και κίνησης στο TAXISnet !

Τέλος, από στελέχη της Γενικής Γραμματείας Πληροφοριακών Συστημάτων αναφέρθηκε ότι μέχρι και πυρκαϊά είχε εκδηλωθεί στον χώρο που φιλοξενούνταν οι υποδομές (εξοπλισμός) του TAXISnet.

3.5 Η προσέγγιση του Υπολογιστικού Νέφους

Τον κακό σχεδιασμό σε επίπεδο βάσης δεδομένων και λογισμικού το υπολογιστικό νέφος δεν μπορεί να δώσει λύση. Προβλήματα όμως όπως αυτά που παρουσιάστηκαν στο TaxisNet και αφορούν διάφορες αστοχίες στους διαθέσιμους πόρους (υλικό) ή καταστροφές αυτών ή μη επάρκεια πόρων από μη αναμενόμενη κίνηση, το υπολογιστικό νέφος μπορεί να δώσει λύση σε τέτοιου είδους προβλήματα.

Για την εξάλειψη των προβλημάτων αυτών προτείνω την υιοθέτηση του υπολογιστικού νέφους. Η Γενική Γραμματεία Πληροφοριακών Συστημάτων θα πρέπει να βασίζεται πλέον στο μοντέλο, υποδομές ως υπηρεσία (IaaS) και να παρέχει όλες της σημερινές υπηρεσίες προς φυσικά και νομικά πρόσωπα με το μοντέλο λογισμικό ως υπηρεσία (SaaS).

Αξίζει να αναφερθεί ότι εκτός από τα προβλήματα που θα λύσει η υιοθέτηση του υπολογιστικού νέφους το κράτος μπορεί να εξοικονομήσει και αρκετούς οικονομικούς πόρους από την χρήση υπολογιστικού νέφους κάτι που είναι πολύ σημαντικό για την δύσκολη δημοσιονομική κατάσταση που βρίσκεται τα τελευταία χρόνια η Ελλάδα. Τέλος, όφελος μπορεί να έχει και από τα υπόλοιπα πλεονεκτήματα της υιοθέτησης του υπολογιστικού νέφους όπως έχουν αναφερθεί στο κεφάλαιο 2.4 .

Στην παρούσα εργασία θα επιχειρήσω να προσημειώσω ένα μικρό κομμάτι από το πλήθος των υπηρεσιών που προσφέρει σήμερα η Γενική Γραμματεία Πληροφοριακών Συστημάτων και είναι αυτό της φορολόγησης εισοδήματος φυσικών προσώπων. Συμπεριλαμβάνω τις διεργασίες :

1. Εγγραφή στο σύστημα
2. Είσοδο στο σύστημα
3. Καταχώρηση φορολογικών εντύπων E1 και E2
4. Υπολογισμός φόρου

Η υπηρεσία αυτή παρέχεται με το μοντέλο λογισμικό ως υπηρεσία (SaaS) κάνοντας χρήση του μοντέλου, υποδομές ως υπηρεσία (IaaS) του υπολογιστικού νέφους που παρέχει η εταιρεία Microsoft. Η επιλογή αυτού του παρόχου έγινε μόνο και μόνο διότι ήταν ο μόνος πάροχος που προσέφερε περισσότερους υπολογιστικούς πόρους με ακαδημαϊκή άδεια χωρίς καμία οικονομική επιβάρυνση για το πανεπιστήμιο .

3.6 Υποθέσεις

Για τον υπολογισμό τους φόρου εισοδήματος, έπειτα από την εκκαθάριση του φορολογικού εντύπου E1, έχω κάνει κάποιες υποθέσεις καθώς δεν κατάφερα να βρω τον αλγόριθμο υπολογισμού του φόρου αλλά ούτε και να μάθω πως γίνεται ακριβώς ο υπολογισμός αυτός. Έτσι βασίστηκα σε όσες πληροφορίες βρήκα στο έντυπο του υπουργείου οικονομικών “Οδηγίες για τη συμπλήρωση της δήλωσης φορολογίας εισοδήματος φυσικών προσώπων οικονομικού έτους 2012”.

Συγκεκριμένα για τον υπολογισμό του φόρου εισοδήματος δεν υπολογίζω τα παρακάτω:

1. Εάν ο ένας σύζυγος δεν έχει εισόδημα ή αυτό που έχει είναι μικρότερο από το αφορολόγητο ποσό του πρώτου κλιμακίου της κλίμακας, το αφορολόγητο ποσό που αφορά τα τέκνα ή η διαφορά που προκύπτει και μέχρι το αφορολόγητο ποσό που αφορά τα τέκνα , προστίθεται στο αφορολόγητο ποσό του άλλου συζύγου.
2. Αν στο συνολικό εισόδημα περιλαμβάνει και εισόδημα από ακίνητα, εκτός από το απαλλασσόμενο εισόδημα από ιδιοκατοίκηση γενικά, το ακαθάριστο ποσό αυτού υποβάλλεται και σε συμπληρωματικό φόρο 1,5%.
3. Περαιτέρω επιβολή συμπληρωματικού φόρου 3% στο ακαθάριστο εισόδημα εφόσον, η επιφάνεια κατοικίας υπερβαίνει τα (300) τετραγωνικά μέτρα της κατοικίας ή πρόκειται για επαγγελματική στέγη ή εμπορική μίσθωση.
4. Θεωρώ ως αφορολόγητο όριο τα 11999€. Επομένως δεν υπολογίζω τυχόν εκπτώσεις του αφορολόγητου ορίου.
5. Μεταφορά τεκμηρίου (αντικειμενικών δαπανών) μεταξύ υπόχρεου και συζύγου. Όταν ο ένας εκ των δύο έχει μεγαλύτερο τεκμήριο από το ετήσιο εισόδημα, τότε το υπόλοιπο της διαφοράς αυτής δε το μεταφέρω στον άλλο.
6. Τα στοιχεία του πίνακα 8 από το φορολογικό έντυπο Ε1
7. Τα στοιχεία του πίνακα 9 από το φορολογικό έντυπο Ε1
8. Τα στοιχεία του πίνακα 10 από το φορολογικό έντυπο Ε1
9. Τα στοιχεία του πίνακα 12 από το φορολογικό έντυπο Ε1

Κεφάλαιο 4

Στο κεφάλαιο αυτό παρουσιάζεται η ερευνητική μεθοδολογία που ακολούθησα και ο τρόπος που πραγματοποίησα τις δοκιμές. Η ερευνητική μεθοδολογία laboratory experiment είναι μια συστηματική και επιστημονική προσέγγιση της έρευνας στην οποία ο ερευνητής χειρίζεται μία ή περισσότερες μεταβλητές, και ελέγχους και μετρά οποιαδήποτε αλλαγή σε άλλες μεταβλητές.

4.1 Τεκμηρίωση ερευνητικής μεθοδολογίας

Για τον έλεγχο των “αντοχών” του συστήματος στο υπολογιστικό νέφος εκτέλεσα δοκιμές αντοχής (stress test) και φορτίου (load test). Για τις ανάγκες των δοκιμών δημιούργησα δύο νέες εφαρμογές. Η πρώτη εφαρμογή (manousos-stresstest.exe) γραμμένη σε F# θα εκτελεί (σε παραλληλία) την δεύτερη εφαρμογή (TaxisTest.jar) η οποία είναι γραμμένη σε Java και εκτελεί τις βασικές λειτουργίες ενός χρήστη στο TaxisNet όπως:

1. Εγγραφή (register)
2. Σύνδεση (login)
3. Υποβολή φορολογικών εντύπων E1 και E2
4. Εμφάνιση φόρου εισοδήματος για το τρέχον οικονομικό έτος (εκκαθάριση βάση του εντύπου E1)

Για τις ανάγκες των δοκιμών η δεύτερη εφαρμογή δέχεται κάθε φορά ένα ΑΦΜ και για αυτό το ΑΦΜ δημιουργεί ένα νέο χρήστη του TaxisNet (register) με συγκεκριμένα προσωπικά στοιχεία. Έπειτα πραγματοποιεί είσοδο στο σύστημα για τον χρήστη που μόλις δημιούργησε και υποβάλλει τυχαία ένα από τα διαθέσιμα έντυπα E1 και E2 (αν απαιτείται) που έχω δημιουργήσει. Τέλος, έπειτα από την επιτυχή καταχώρηση του εντύπου πραγματοποιείται η εκκαθάριση με τον υπολογισμό του φόρου όπως περιγράφεται στο κεφάλαιο 5.

Βασίζομενος στα στατιστικά που εξέδωσε η Γενική Γραμματεία Πληροφοριακών Συστημάτων, για την περίοδο υποβολής φορολογίας εισοδήματος για το οικονομικό έτος 2012, ο μέγιστος αριθμός καταχώρησης αιτήσεων σε μια μέρα ήταν εκατόν εβδομήντα πέντε χιλιάδες αιτήσεις. Επομένως, εκτέλεσα τρία διαφορετικά σενάρια δοκιμών όπως περιγράφω στην συνέχεια.

Θεωρητικά, λόγω του ότι οι υποδομές παρέχονται ως υπηρεσία από το υπολογιστικό νέφος, θα ήταν αναμενόμενο όλα τα σενάρια δοκιμών να στεφθούν με απόλυτη επιτυχία. Και αυτό διότι θα έχουμε την δυνατότητα να αξιοποιήσουμε ένα από τα βασικά χαρακτηριστικά του υπολογιστικού νέφους, την ελαστικότητα (Scalability and elasticity). Δηλαδή σε πραγματικό χρόνο όταν εμφανιστεί burst ή γενικά αυξημένη και μη αναμενόμενη κίνηση να μπορούμε να προσθέτουμε επιπλέον πόρους ώστε να αντεπεξέλθουν στις πρόσκαιρα αυξημένες απαιτήσεις. Έτσι ως τελικό αποτέλεσμα ανανεώνουμε να επιτευχθεί η εκκαθάριση (υπολογισμός φόρου εισοδήματος) των εντύπων στο συνολικό πλήθος αιτήσεων του κάθε σεναρίου εκτέλεσης.

4.1.1 Τρόπος εκτέλεσης

Εκτελώ το manousos-stresstest.exe που είναι η πρώτη εφαρμογή (γραμμένη σε F#) από γραμμή εντολών, δίνοντας παραμέτρους:

- a. Το όνομα του εκτελέσιμου της δεύτερης εφαρμογής TaxisTest.jar, αλλά επειδή είναι εκτελέσιμο Java χρειάζεται και το "java -jar"
- b. Το αρχικό ΑΦΜ (ΑΦΜ εκκίνησης)
- c. Το τελικό ΑΦΜ (ΑΦΜ τερματισμού)

Ένα παράδειγμα είναι:

```
./manousos-stresstest.exe "java" "-jar TaxisTest.jar" 2000 3000
```

Αυτό σημαίνει ότι η εφαρμογή manousos-stresstest θα εκτελέσει χίλιες φορές την εφαρμογή TaxisTest. Επομένως δημιουργούνται χίλιες διεργασίες του TaxisTest, μια για κάθε ΑΦΜ. Στο παράδειγμα αυτό το τερματικό θα επιχειρήσει να υποβάλει και να γίνει εκκαθάριση του εντύπου E1 για τους φορολογούμενους με ΑΦΜ από το 000002000 έως το ΑΦΜ 000003000.

4.1.2 Πρώτο σενάριο δοκιμής

Σε αυτό το σενάριο χρησιμοποίησα εικοσιπέντε τερματικά που διέθεταν διπύρηνο επεξεργαστή . Επομένως σε κάθε κύκλο ρολογιού θα έχω ταυτόχρονα έως δύο χρήστες στο σύστημα TaxisNet. Άρα συνολικά επιτυγχάνω παραλληλία με έως πενήντα χρήστες στο TaxisNet σε κάθε κύκλο ρολογιού. Συνολικά, κάθε τερματικό έπρεπε να εκτελέσει την εφαρμογή TaxisTest για επτά χιλιάδες διαφορετικούς φορολογούμενους (επτά χιλιάδες διαφορετικά ΑΦΜ).

4.1.3 Δεύτερο σενάριο δοκιμής

Σε αυτό το σενάριο χρησιμοποίησα εικοσιτέσσερα τερματικά που διέθεταν διπύρηνο επεξεργαστή . Επομένως σε κάθε κύκλο ρολογιού θα έχω ταυτόχρονα έως δύο χρήστες στο σύστημα TaxisNet. Άρα συνολικά επιτυγχάνω παραλληλία με έως σαράντα οχτώ χρήστες στο TaxisNet σε κάθε κύκλο ρολογιού. Συνολικά, κάθε τερματικό έπρεπε να εκτελέσει την εφαρμογή TaxisTest για επτά χιλιάδες διαφορετικούς φορολογούμενους (επτά χιλιάδες διαφορετικά ΑΦΜ).

4.1.4 Τρίτο σενάριο δοκιμής

Σε αυτό το σενάριο χρησιμοποίησα δύο τερματικά όπου το ένα διέθετε διπύρηνο επεξεργαστή και το άλλο τετραπύρηνο επεξεργαστή. Επομένως σε κάθε κύκλο ρολογιού θα έχω ταυτόχρονα έως έξη χρήστες στο σύστημα TaxisNet. Άρα συνολικά επιτυγχάνω παραλληλία με έως έξη χρήστες στο TaxisNet σε κάθε κύκλο ρολογιού. Το τερματικό με τον διπύρηνο επεξεργαστή έπρεπε να εκτελέσει την εφαρμογή TaxisTest για εκατό χιλιάδες διαφορετικούς φορολογούμενους (εκατό χιλιάδες διαφορετικά ΑΦΜ). Το τερματικό με τον τετραπύρηνο επεξεργαστή, έπρεπε να εκτελέσει την εφαρμογή TaxisTest για εκατόν εβδομήντα πέντε χιλιάδες διαφορετικούς φορολογούμενους (εκατόν εβδομήντα πέντε χιλιάδες διαφορετικά ΑΦΜ).

Κεφάλαιο 5

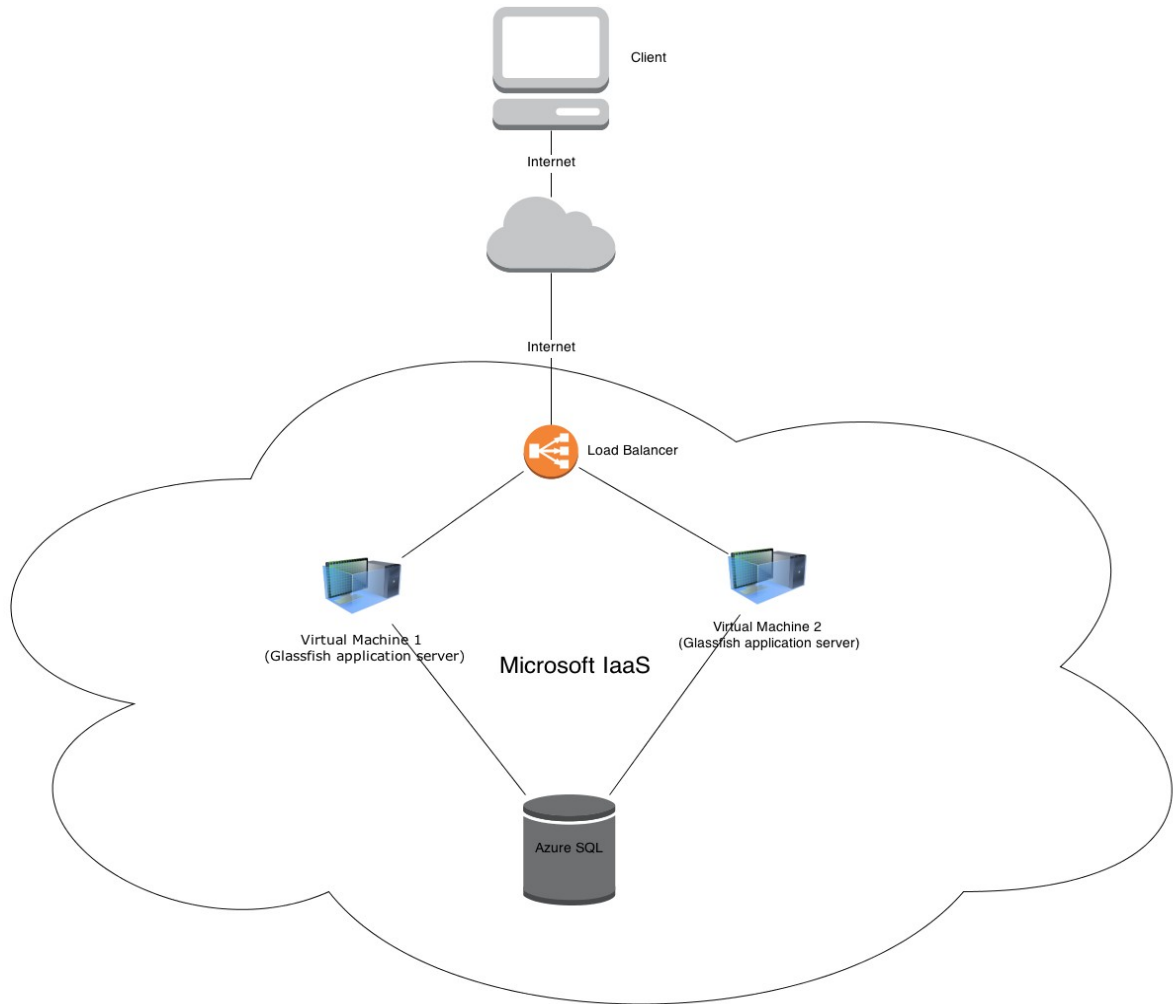
Η υλοποίηση των υπηρεσιών υλοποιήθηκε με Java 6. Ο εξυπηρετητής των υπηρεσιών είναι ο Oracle Glassfish 3.1.2. Για βάση δεδομένων στα πλαίσια της ανάπτυξης χρησιμοποιήσα MySQL 5 και όταν έγινε η μετάβαση στο υπολογιστικό νέφος για τις δοκιμές και την εκτέλεση των σεναρίων, έγινε migration σε Microsoft SQL Server.

Τα frameworks, API, πλατφόρμες και βιβλιοθήκες που χρησιμοποιήθηκαν είναι :

1. Hibernate 3.5 (Java persistence framework)
2. J2EE
3. Jersey (RESTful Web Services framework)
4. JavaServer Faces 2 (Web application framework)
5. Apache Commons Logging 1.1.1 API
6. Apache log4j 1.2.17 (logging library for Java)

5.1 Αρχιτεκτονική

Στην εικόνα 5.1 απεικονίζεται η αρχιτεκτονική ανάπτυξης του συστήματος στο υπολογιστικό νέφος της Microsoft (Azure). Οι πελάτες στέλνουν τις διάφορες αιτήσεις τους προς το σύστημα μέσω Internet (TCP/IP). Ο εξισορροπητής φορτίου (load balancer) αναλαμβάνει να δρομολογήσει την αίτηση στον εξυπηρετητή με το μικρότερο φόρτο. Στη συγκεκριμένη περίπτωση είχα διαθέσιμους μόνο δύο εξυπηρετητές (οι λόγοι αναφέρονται στο κεφάλαιο 6.4). Στους εξυπηρετητές ο εξυπηρετητής εφαρμογών που είναι εγκατεστημένος αναλαμβάνει να εκτελέσει το κάθε αίτημα. Όλοι οι εξυπηρετητές έχουν πρόσβαση στην βάση δεδομένων μέσω TCP.



Εικόνα 5.1 Αρχιτεκτονική υλοποίησης

5.1.1 Διάγραμμα παράταξης

Στην εικόνα 5.2 απεικονίζεται η φυσική αρχιτεκτονική τους συστήματος που υλοποιήσα μέσω UML διαγράμματος παράταξης. Σε κάθε εικονικό μηχάνημα έχει εγκατασταθεί το λειτουργικό σύστημα “UBUNTU 13.04 64bit Server edition”. Επιπλέον έχει εγκατασταθεί το εξής λογισμικό:

1. Glassfish 3.1.2 (Open-source application server)
2. openjdk-6-jre (περιβάλλον εκτέλεσης Java)
3. openjdk-6-jdk (περιβάλλον ανάπτυξης Java)

Ο εξυπηρετητής εφαρμογών (Glassfish) φιλοξενεί τις δύο εφαρμογές

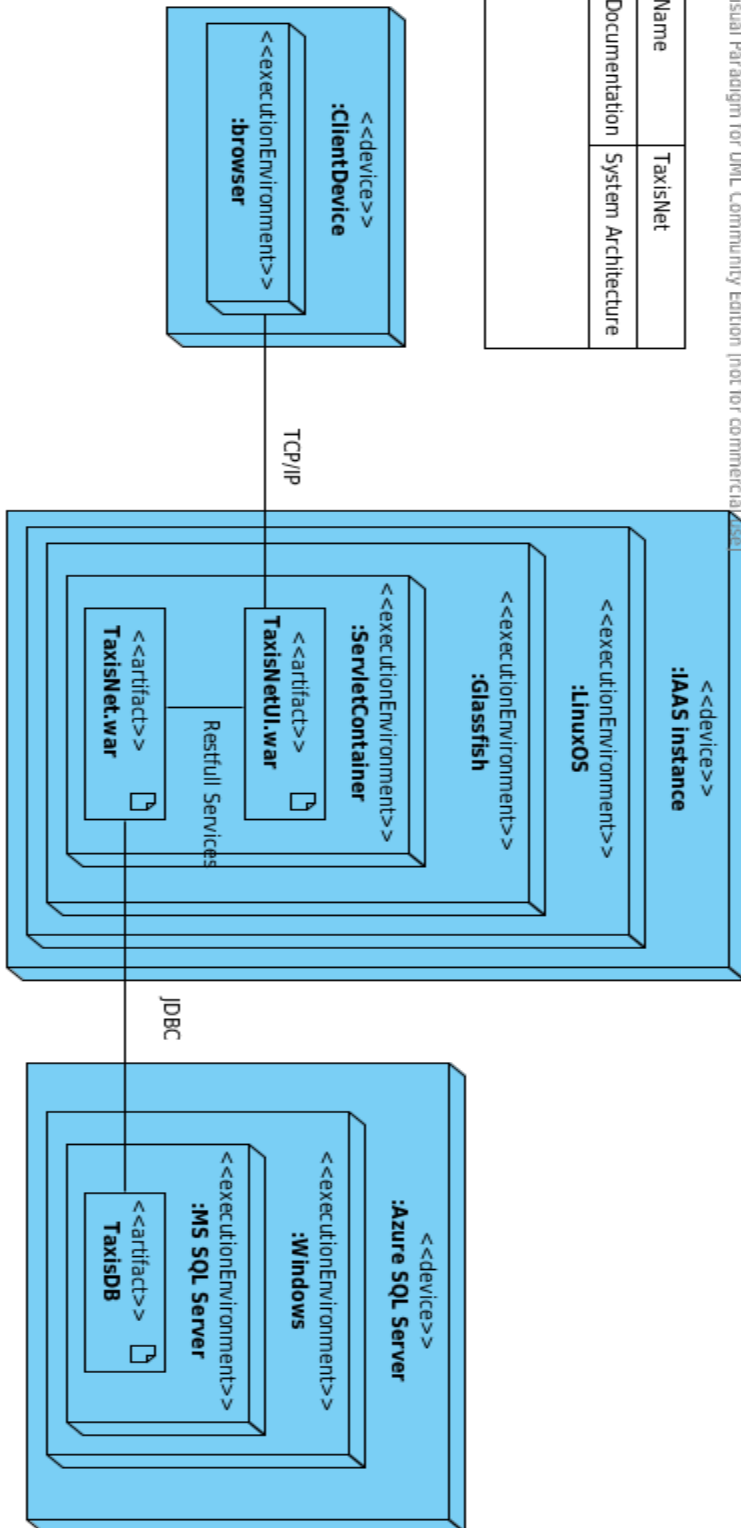
1. **TaxisNetUI**
2. **TaxisNet**

Και οι δύο εφαρμογές παρέχονται ως λογισμικό ως υπηρεσία (SaaS). Η πρώτη εφαρμογή περιέχει τη διεπαφή χρήστη (εφαρμογή ιστού). Η εφαρμογή αυτή για να εκτελέσει όλες τις λειτουργίες που παρέχει προς τον φορολογούμενο επικοινωνεί κάνοντας χρήση υπηρεσιών Rest με την εφαρμογή TaxisNet, η οποία είναι η βασική εφαρμογή που επεξεργάζεται όλες τις

αιτήσεις των χρηστών. Η μορφή των μηνυμάτων που ανταλλάσσονται μεταξύ των δύο εφαρμογών, μπορεί να είναι είτε σε JSON είτε σε XML. Η εφαρμογή TaxisNet επικοινωνεί με την βάση δεδομένων TaxisDB που φιλοξενείται στην πλατφόρμα υπηρεσιών Azure και παρέχεται ως βάση δεδομένων ως υπηρεσία (DaaS).

Πανεπιστήμιο Πειραιώς

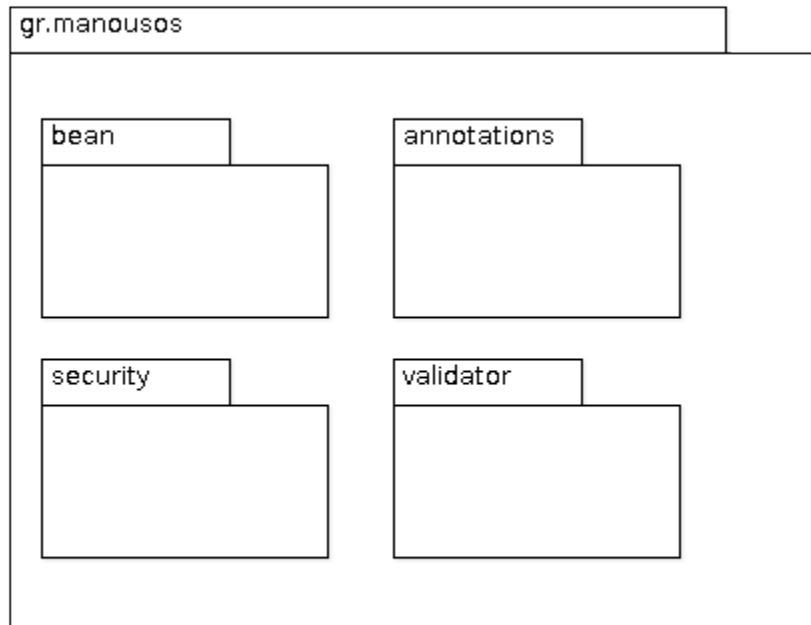
Name	TaxisNet
Documentation	System Architecture



Εικόνα 5.2 Διάγραμμα παράταξης

5.1.2 Διαγράμματα πακέτων

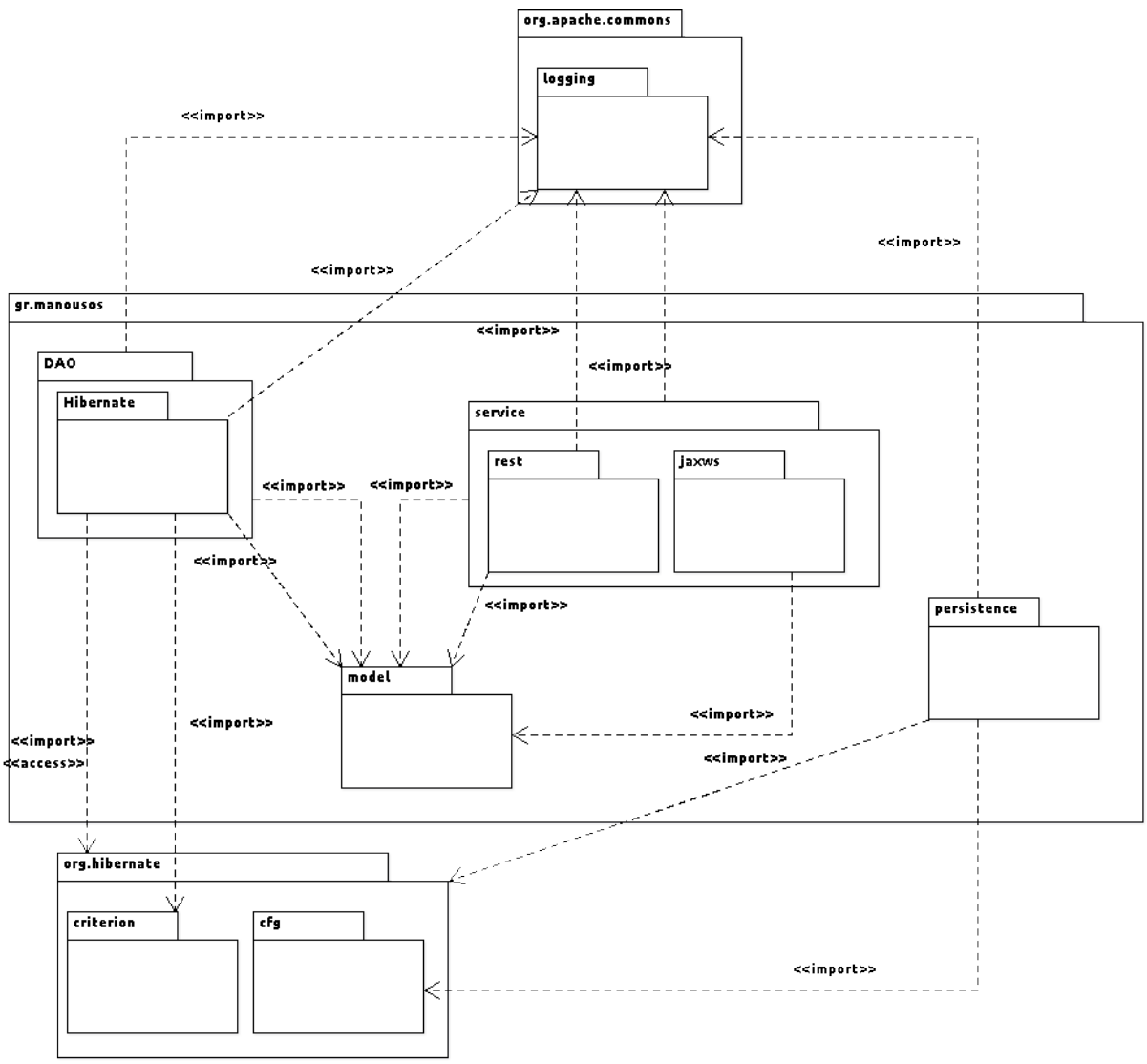
Οργάνωση πακέτων εφαρμογής TaxisNetUI (User interface)



Εικόνα 5.3 Διάγραμμα πακέτων TaxisNetUI

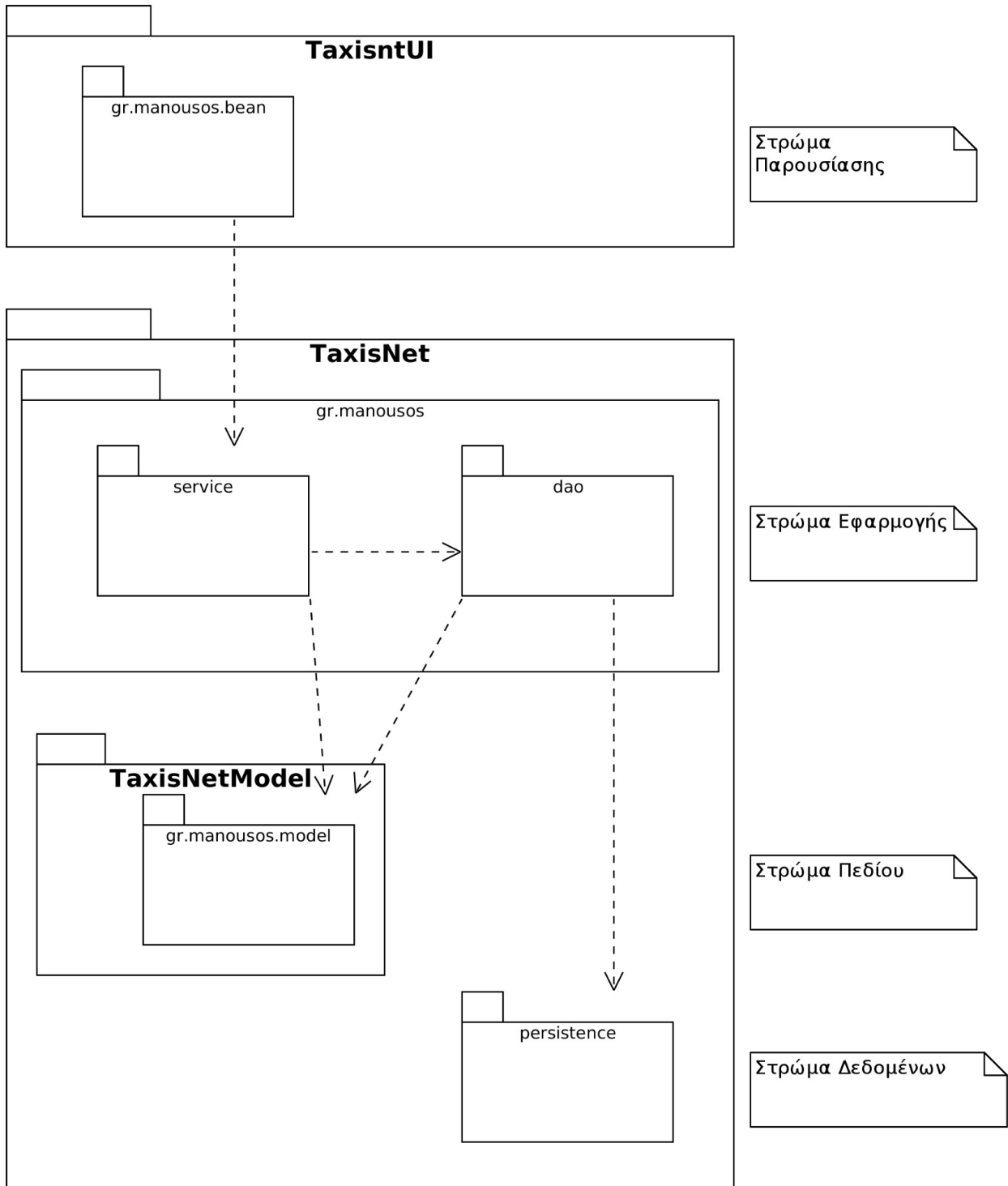
Οργάνωση πακέτων εφαρμογής TaxisNet

ΠΑΝΕΠΙΣΤΗΜΙΟΝ



Εικόνα 5.4 Διάγραμμα πακέτων TaxisNet

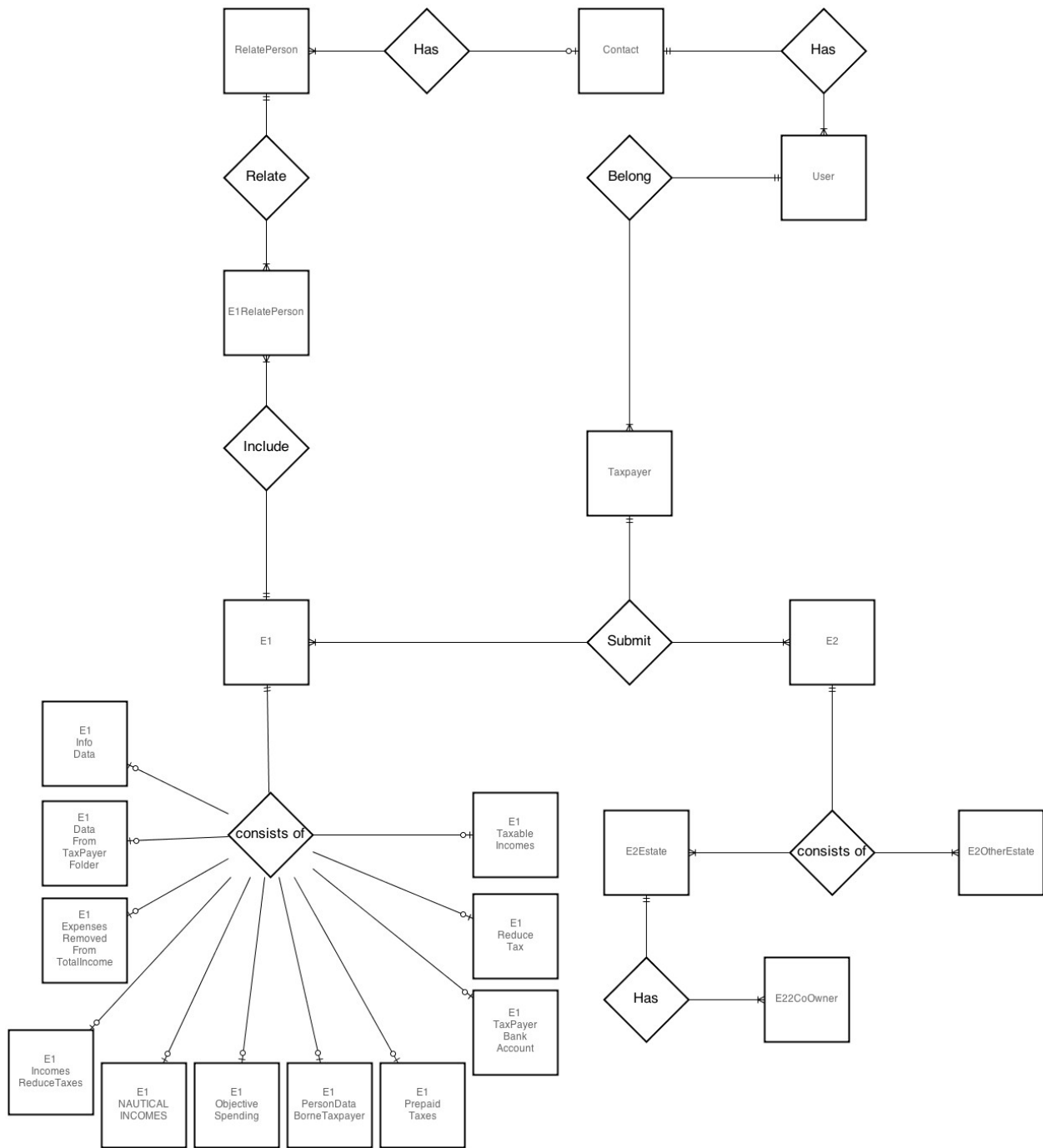
Στην εικόνα 5.5 απεικονίζεται η διαστρωμάτωση των συστατικών της εφαρμογής TaxisNet. Με έντονα γράμματα είναι το όνομα των εφαρμογών ή της βιβλιοθήκης. Στο πρώτο επίπεδο είναι το στρώμα παρουσίασης. Αποτελείται από την εφαρμογή **TaxisNetUI**, η οποία είναι web διεπαφή χρήστη. Μέσω της κάθε τάξης του πακέτου *bean*, πραγματοποιείται επικοινωνία μέσω υπηρεσιών ιστού με το στρώμα εφαρμογής και συγκεκριμένα το πακέτο *service*. Το στρώμα αυτό αποτελείται από την βιβλιοθήκη **TaxisNetModel** και την εφαρμογή **TaxisNet**. Το πακέτο *service* επικοινωνεί με το πακέτο *dao*. Και τα δύο αυτά πακέτα επικοινωνούν με το στρώμα πεδίου το οποίο αποτελείται από τη βιβλιοθήκη TaxisNetModel ή οποία περιλαμβάνει τις οντότητες (entities) της εφαρμογής TaxisNet. Τέλος το στρώμα εφαρμογής επικοινωνεί με το στρώμα δεδομένων μέσω του πακέτου *dao* προς το πακέτο *persistence*, το οποίο είναι επιφορτισμένο με τις λειτουργίες της ανάκτησης και αποθήκευσης των δεδομένων προς τη βάση δεδομένων TaxisDB (κεφάλαιο 5.2)



Εικόνα 5.5 Διαστρωμάτωση συστατικών

5.2 Η βάση δεδομένων TaxisDB

Στην σχεσιακή βάση δεδομένων αποθηκεύονται όλες οι πληροφορίες που αφορούν τους φορολογούμενους και τα φορολογικά έντυπα E1 και E2.
 Στη συνέχεια παρουσιάζω τη δομή των πινάκων και το πως συσχετίζονται μεταξύ τους.



Εικόνα 5.6 Διάγραμμα οντοτήτων συσχετίσεων TaxisDB

5.2.1 Δομή πίνακα Contact

Στον πίνακα Contact αποθηκεύονται οι πληροφορίες επικοινωνίας των φορολογούμενων

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
ContactID	integer	255	false	true
Phone	varchar(15)	15	false	false
Fax	varchar(15)	15	true	false
Cell	varchar(15)	15	false	false
Email	varchar(100)	100	false	false

Πρωτεύον κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	ContactID

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	ContactID

5.2.2 Δομή πίνακα E1

Στον πίνακα E1 αποθηκεύονται οι βασικές πληροφορίες του φορολογικού εντύπου E1

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
TaxpayerID	integer	255	false	false
Year	integer	255	false	false
DOY	integer	45	false	false
ATID	varchar(45)	10	false	false
TaxpayerAddress	varchar(10)	500	false	false
isComplete	longtext	255	true	false
Marriage	integer	255	true	false
DateInserted	integer	19	true	false

Πρωτεύον κλειδί	
-----------------	--

Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	TaxpayerID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
FK88C6C298EE8	Taxpayer	TaxpayerID
fk_E1_Taxpayer	Taxpayer	TaxpayerID

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	TaxpayerID, Year

5.2.3 Δομή πίνακα E1DataFromTaxPayerFolder

Στον πίνακα E1DataFromTaxPayerFolder αποθηκεύονται τα δεδομένα από το φάκελο του φορολογούμενου .

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
_901	float	255	true	false
_902	float	255	true	false
_903	float	255	true	false
_904	float	255	true	false
_341	float	255	true	false
_995	float	255	true	false
_996	float	255	true	false

Πρωτεύον κλειδί	
Όνομα κλειδιού	Όνομα πεδίου

PRIMARY	tID, Year
---------	-----------

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1DataFromTaxPayerFolder_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.4 Δομή πίνακα E1ExpensesRemovedFromTotalIncome

Στον πίνακα E1ExpensesRemovedFromTotalIncome αποθηκεύονται τα ποσά που αφαιρούνται από το συνολικό εισόδημα ή από το φόρο.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
_049	float	255	true	false
_051	float	255	true	false
_052	float	255	true	false
_053	float	255	true	false
_054	float	255	true	false
_057	float	255	true	false
_058	float	255	true	false
_031	float	255	true	false
_032	float	255	true	false
_059	float	255	true	false
_060	float	255	true	false
_075	float	255	true	false
_076	float	255	true	false
_061	float	255	true	false
_062	float	255	true	false
_071	float	255	true	false
_063	float	255	true	false
_064	float	255	true	false

069	float	255	true	false
070	float	255	true	false
RentalForFamilyOwnerName1	varchar(45)	45	true	false
801	varchar(10)	10	true	false
092	integer	255	true	false
091	float	255	true	false
097	integer	255	true	false
811	float	255	true	false
812	float	255	true	false
RentalForFamilyOwnerName2	varchar(45)	45	true	false
802	varchar(10)	10	true	false
094	integer	255	true	false
093	float	255	true	false
098	integer	255	true	false
813	float	255	true	false
814	float	255	true	false
RentalForFamilyOwnerName3	varchar(45)	45	true	false
803	varchar(10)	10	true	false
096	integer	255	true	false
095	float	255	true	false
099	float	255	true	false
815	float	255	true	false
816	float	255	true	false
RentalForStudyOwnerName1	varchar(45)	45	true	false
804	varchar(10)	10	true	false
817	float	255	true	false
RentalForStudyOwnerName2	varchar(45)	45	true	false
805	varchar(10)	10	true	false
819	float	255	true	false
RentalForStudyOwnerName3	varchar(45)	45	true	false
806	varchar(10)	10	true	false
821	float	255	true	false
RentalForStudyOwnerName4	varchar(45)	45	true	false
807	varchar(10)	10	true	false
823	float	255	true	false
073	float	255	true	false
074	float	255	true	false
089	float	255	true	false
090	float	255	true	false
087	float	255	true	false
088	float	255	true	false
079	float	255	true	false
080	float	255	true	false
081	float	255	true	false
082	float	255	true	false
083	float	255	true	false
084	float	255	true	false
085	float	255	true	false
077	float	255	true	false
078	float	255	true	false
663	float	255	true	false
664	float	255	true	false
033	integer	255	true	false

034	integer	255	true	false
035	integer	255	true	false
036	integer	255	true	false

Πρωτεύον κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1ExpensesRemovedFromTotalIncome_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.5 Δομή πίνακα E1IncomesReduceTaxes

Στον πίνακα E1IncomesReduceTaxes αποθηκεύονται τα ποσά που μειώνουν την ετήσια αντικειμενική δαπάνη .

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
_655	float	255	true	false
_656	float	255	true	false
_693	float	255	true	false
_694	float	255	true	false
_659	float	255	true	false
_660	float	255	true	false

_657	float	255	true	false
_658	float	255	true	false
_661	float	255	true	false
_662	float	255	true	false
_431	float	255	true	false
_432	float	255	true	false
_433	float	255	true	false
_434	float	255	true	false
_305	float	255	true	false
_306	float	255	true	false
_477	float	255	true	false
_478	float	255	true	false
LessorName1	varchar(45)	45	true	false
_790	varchar(10)	10	true	false
_795	float	255	true	false
LessorName2	varchar(45)	45	true	false
_791	varchar(10)	10	true	false
_796	float	255	true	false
_793	float	255	true	false
_794	float	255	true	false
_615	float	255	true	false
_616	float	255	true	false
LessorNameForStudent	varchar(45)	45	true	false
_417	varchar(10)	10	true	false
AreaStudentHouse	float	255	true	false
_419	float	255	true	false
_420	float	255	true	false
_735	float	255	true	false
_736	float	255	true	false
_781	float	255	true	false
_782	float	255	true	false
_783	float	255	true	false
_784	float	255	true	false
_787	float	255	true	false
_788	float	255	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1IncomesReduceTaxes_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.6 Δομή πίνακα E1InfoData

Στον πίνακα E1InfoData αποθηκεύονται τα πληροφοριακά δεδομένα που περιέχει το έντυπο E1

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
_327	integer	255	true	false
_328	integer	255	true	false
_319	integer	255	true	false
_320	integer	255	true	false
_023	integer	255	true	false
_024	integer	255	true	false
_329	integer	255	true	false
_330	integer	255	true	false
_331	integer	255	true	false
_011	integer	255	true	false

_012	integer	255	true	false
_013	integer	255	true	false
_014	integer	255	true	false
_015	integer	255	true	false
_016	integer	255	true	false
_017	integer	255	true	false
_018	integer	255	true	false
_021	integer	255	true	false
_022	integer	255	true	false
_025	integer	255	true	false
_026	integer	255	true	false
_007	integer	255	true	false
_008	integer	255	true	false
_617	integer	255	true	false
_385	integer	255	true	false
_386	integer	255	true	false
_029	integer	255	true	false
_030	integer	255	true	false
_905	integer	255	true	false
_906	integer	255	true	false
_911	integer	255	true	false
_912	integer	255	true	false
_010	integer	255	true	false

Κύριο κλειδί		
Όνομα κλειδιού		Όνομα πεδίου
PRIMARY		tID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1InfoData_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.7 Δομή πίνακα E1NAUTICALINCOMES

Στον πίνακα E1NAUTICALINCOMES αποθηκεύονται τα δεδομένα που αφορούν εισοδήματα αξιωματικών και ημεδαπού κατώτερου πληρώματος εμπορικού ναυτικού, που περιέχει το έντυπο Ε1 .

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
_255	float	255	true	false
_256	float	255	true	false
_257	float	255	true	false
_258	float	255	true	false
_263	float	255	true	false
_264	float	255	true	false
_265	float	255	true	false
_266	float	255	true	false
_253	float	255	true	false
_254	float	255	true	false
_201	float	255	true	false
_202	float	255	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

Ξένα κλειδιά		
--------------	--	--

Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1NAUTICALINCOMES_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.8 Δομή πίνακα E1ObjectiveSpending

Στον πίνακα E1ObjectiveSpending αποθηκεύονται τα δεδομένα που αφορούν τον προσδιορισμό της ετήσιας αντικειμενικής δαπάνης, που περιέχει το έντυπο E1 .

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
_205	varchar(45)	45	true	false
HouseAddr1	varchar(200)	200	true	false
Floor1	varchar(45)	45	true	false
_203	integer	255	true	false
_240	integer	255	true	false
_211	float	255	true	false
_212	float	255	true	false
_213	float	255	true	false
_214	float	255	true	false
_215	integer	255	true	false
_216	float	255	true	false
HouseAddr2	varchar(200)	200	true	false
Floor2	varchar(45)	45	true	false
_207	integer	255	true	false
_241	integer	255	true	false
_218	float	255	true	false

_219	float	255	true	false
_220	float	255	true	false
_221	float	255	true	false
_222	integer	255	true	false
_223	float	255	true	false
HouseAddr3	varchar(200)	200	true	false
Floor3	varchar(45)	45	true	false
_209	integer	255	true	false
_242	integer	255	true	false
_225	float	255	true	false
_226	float	255	true	false
_227	float	255	true	false
_228	float	255	true	false
_229	integer	255	true	false
_230	float	255	true	false
_707	float	255	true	false
_708	float	255	true	false
AFM1	varchar(10)	10	true	false
_750	varchar(7)	7	true	false
_703	integer	255	true	false
_761	integer	255	true	false
_771	float	255	true	false
_775	integer	255	true	false
AFM2	varchar(10)	10	true	false
_751	varchar(7)	7	true	false
_704	integer	255	true	false
_762	integer	255	true	false
_772	float	255	true	false
_776	integer	255	true	false
AFM3	varchar(10)	10	true	false
_752	varchar(7)	7	true	false
_705	integer	255	true	false
_763	integer	255	true	false
_773	float	255	true	false
_777	integer	255	true	false
AFM4	varchar(10)	10	true	false
_753	varchar(7)	7	true	false

_706	integer	255	true	false
_764	integer	255	true	false
_774	float	255	true	false
_778	integer	255	true	false
_851	float	255	true	false
_852	float	255	true	false
_853	float	255	true	false
_854	float	255	true	false
_855	float	255	true	false
_856	float	255	true	false
_857	float	255	true	false
_858	float	255	true	false
ShippingName1	varchar(45)	45	true	false
ShippingRegister1	varchar(100)	100	true	false
ShippingCountry1	varchar(45)	45	true	false
SailShip1	integer	255	true	false
AccommodationSpace1	integer	255	true	false
PercentPrincipalCoOwner1	float	255	true	false
PercentWifeCoOwner1	float	255	true	false
FirstRegister1	integer	255	true	false
OwnershipMonths1	integer	255	true	false
_747	float	255	true	false
ShippingName2	varchar(45)	45	true	false
ShippingRegister2	varchar(100)	100	true	false
ShippingCountry2	varchar(45)	45	true	false
SailShip2	integer	255	true	false
AccommodationSpace2	integer	255	true	false
PercentPrincipalCoOwner2	float	255	true	false
PercentWifeCoOwner2	float	255	true	false
FirstRegister2	integer	255	true	false
OwnershipMonths2	integer	255	true	false
_748	float	255	true	false
_711	float	255	true	false
_712	float	255	true	false
_713	float	255	true	false
_714	float	255	true	false
_731	float	255	true	false

_732	float	255	true	false
AircraftRegisterData	varchar(100)	100	true	false
AircraftType	varchar(45)	45	true	false
AircraftSN	varchar(45)	45	true	false
AirportBase	varchar(45)	45	true	false
AircraftOwnerShip	integer	255	true	false
AircraftPowerLibres	integer	255	true	false
AircraftFirstRegister	integer	255	true	false
_715	float	255	true	false
_716	float	255	true	false
_767	float	255	true	false
PoolPrincipalCoOwnerOutdoor	float	255	true	false
PoolWifeCoOwnerOutdoor	float	255	true	false
_768	float	255	true	false
PoolPrincipalCoOwnerIndoor	float	255	true	false
PoolWifeCoOwnerIndoor	float	255	true	false
_765	float	255	true	false
_766	float	255	true	false
_769	float	255	true	false
_770	float	255	true	false
_719	float	255	true	false
_720	float	255	true	false
_721	float	255	true	false
_722	float	255	true	false
_723	float	255	true	false
_724	float	255	true	false
_725	float	255	true	false
_726	float	255	true	false
_727	float	255	true	false
_728	float	255	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1ObjectiveSpending_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.9 Δομή πίνακα E1PersonDataBorneTaxpayer

Στον πίνακα E1PersonDataBorneTaxpayer αποθηκεύονται τα δεδομένα των προσώπων που συνοικούν με τους φορολογούμενους και τους βαρύνουν.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
Name1	varchar(45)	45	true	false
BirthYear1	integer	255	true	false
SchoolName1	varchar(100)	100	true	false
_831	varchar(10)	10	true	false
Name2	varchar(45)	45	true	false
BirthYear2	integer	255	true	false
SchoolName2	varchar(100)	100	true	false
_832	varchar(10)	10	true	false
Name3	varchar(45)	45	true	false
BirthYear3	integer	255	true	false
SchoolName3	varchar(100)	100	true	false
_833	varchar(10)	10	true	false
Name4	varchar(45)	45	true	false
BirthYear4	integer	255	true	false
SchoolName4	varchar(100)	100	true	false
_834	varchar(10)	10	true	false
FullName1	varchar(45)	45	true	false
_835	varchar(10)	10	true	false

RelationshipWithPrincipal1	varchar(45)	45	true	false
RelationShipWithWife1	varchar(45)	45	true	false
FullName2	varchar(45)	45	true	false
_836	varchar(10)	10	true	false
RelationshipWithPrincipal2	varchar(45)	45	true	false
RelationShipWithWife2	varchar(45)	45	true	false
FullName3	varchar(45)	45	true	false
_837	varchar(45)	45	true	false
RelationshipWithPrincipal3	varchar(45)	45	true	false
RelationShipWithWife3	varchar(45)	45	true	false
FullName4	varchar(45)	45	true	false
_838	varchar(10)	10	true	false
RelationshipWithPrincipal4	varchar(45)	45	true	false
RelationShipWithWife4	varchar(45)	45	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1PersonDataBorneTaxpayer_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.10 Δομή πίνακα E1PrepaidTaxes

Στον πίνακα E1PrepaidTaxes αποθηκεύονται τα δεδομένα από τους προκαταβληθέντες - παρακρατηθέντες φόρους.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
_601	float	255	true	false
_602	float	255	true	false
_603	float	255	true	false
_604	float	255	true	false
_605	float	255	true	false
_606	float	255	true	false
_607	float	255	true	false
_608	float	255	true	false
_609	float	255	true	false
_610	float	255	true	false
_651	float	255	true	false
_652	float	255	true	false
_293	float	255	true	false
_294	float	255	true	false
_313	float	255	true	false
_314	float	255	true	false
_315	float	255	true	false
_316	float	255	true	false
_297	float	255	true	false
_298	float	255	true	false
_127	float	255	true	false
_128	float	255	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1PrepaidTaxes_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.11 Δομή πίνακα E1ReduceTax

Στον πίνακα E1ReduceTax αποθηκεύονται τα ποσά που αφαιρούνται λόγω αναπηρίας κτλ και μείωση φόρου.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
_001	integer	255	true	false
_002	integer	255	true	false
_003	integer	255	true	false
_004	integer	255	true	false
_005	integer	255	true	false
_006	integer	255	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1ReduceTax_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.12 Δομή πίνακα E1RelatePersons

Στον πίνακα E1RelatePersons αποθηκεύονται τα δεδομένα των συσχετιζόμενων προσώπων με τον φορολογούμενο, όπως σύζυγος και εκπρόσωπος .

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
TaxpayerID	integer	255	false	false
Year	integer	255	false	false
idRelatePerson	integer	255	false	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	TaxpayerID, Year, idRelatePerson

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1ReduceTax_E1	E1	tID, Year

fk_E1RelatePersons_RelatePerson	RelatePerson	idRelatePerson

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	TaxpayerID, Year, idRelatePerson

Ευρετήρια	
Όνομα ευρετηρίου	Όνομα πεδίου
fk_E1RelatePersons_E1_idx	TaxpayerID , Year
fk_E1RelatePersons_1_idx	idRelatePerson

5.2.13 Δομή πίνακα E1TaxPayerBankAccount

Στον πίνακα E1TaxPayerBankAccount αποθηκεύονται τα δεδομένα του τραπεζικού λογαριασμού του φορολογούμενου, για κατάθεση του επιστρεφόμενου ποσού .

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
BIC	varchar(45)	45	true	false
IBAN	varchar(45)	45	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1PrepaidTaxes_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.14 Δομή πίνακα E1TaxableIncomes

Στον πίνακα E1TaxableIncomes αποθηκεύονται τα δεδομένα που αφορούν τα εισοδήματα του τρέχοντος οικονομικού έτους .

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
tID	integer	255	false	false
Year	integer	255	false	false
301	float	255	true	false
302	float	255	true	false
303	float	255	true	false
304	float	255	true	false
321	float	255	true	false
322	float	255	true	false
317	float	255	true	false
318	float	255	true	false
461	float	255	true	false
462	float	255	true	false
921	float	255	true	false
922	float	255	true	false
919	float	255	true	false
920	float	255	true	false
915	float	255	true	false
916	float	255	true	false
923	float	255	true	false
924	float	255	true	false
335	float	255	true	false
336	float	255	true	false

326	float	255	true	false
337	float	255	true	false
338	float	255	true	false
339	float	255	true	false
340	float	255	true	false
465	float	255	true	false
466	float	255	true	false
467	float	255	true	false
468	float	255	true	false
475	float	255	true	false
476	float	255	true	false
479	float	255	true	false
480	float	255	true	false
481	float	255	true	false
482	float	255	true	false
401	float	255	true	false
402	float	255	true	false
403	float	255	true	false
404	float	255	true	false
405	float	255	true	false
406	float	255	true	false
407	float	255	true	false
408	float	255	true	false
413	float	255	true	false
414	float	255	true	false
415	float	255	true	false
416	float	255	true	false
425	float	255	true	false
426	float	255	true	false
501	float	255	true	false
502	float	255	true	false
503	float	255	true	false
504	float	255	true	false
505	float	255	true	false
506	float	255	true	false
507	float	255	true	false
508	float	255	true	false
511	float	255	true	false
512	float	255	true	false
515	float	255	true	false
516	float	255	true	false
517	float	255	true	false
518	float	255	true	false
103	float	255	true	false
104	float	255	true	false
121	float	255	true	false
122	float	255	true	false
105	float	255	true	false
106	float	255	true	false
107	float	255	true	false
108	float	255	true	false
109	float	255	true	false
110	float	255	true	false

101	float	255	true	false
102	float	255	true	false
909	float	255	true	false
910	float	255	true	false
111	float	255	true	false
112	float	255	true	false
113	float	255	true	false
114	float	255	true	false
129	float	255	true	false
130	float	255	true	false
143	float	255	true	false
144	float	255	true	false
145	float	255	true	false
146	float	255	true	false
147	float	255	true	false
148	float	255	true	false
141	float	255	true	false
142	float	255	true	false
701	float	255	true	false
702	float	255	true	false
123	float	255	true	false
124	float	255	true	false
125	float	255	true	false
126	float	255	true	false
151	float	255	true	false
152	float	255	true	false
163	float	255	true	false
164	float	255	true	false
165	float	255	true	false
166	float	255	true	false
159	float	255	true	false
160	float	255	true	false
175	float	255	true	false
176	float	255	true	false
131	float	255	true	false
132	float	255	true	false
133	float	255	true	false
134	float	255	true	false
741	float	255	true	false
742	float	255	true	false
291	float	255	true	false
292	float	255	true	false
389	float	255	true	false
390	float	255	true	false
391	float	255	true	false
392	float	255	true	false
463	float	255	true	false

Κύριο κλειδί

Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_E1TaxableIncomes_E1	E1	tID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	tID, Year

5.2.15 Δομή πίνακα E2

Στον πίνακα E2 αποθηκεύονται οι βασικές πληροφορίες του φορολογικού εντύπου E2 για κάθε οικονομικό έτος.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
TaxierID	integer	255	false	false
Year	integer	255	false	false
isComplete	integer	255	true	false
DateInserted	datetime	19	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	TaxierID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
FK_Taxpayer_E2	Taxpayer	TaxierID
FK88DFA500F69	Taxpayer	TaxierID

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	TaxierID, Year

5.2.16 Δομή πίνακα E2Estate

Στον πίνακα E2Estate αποθηκεύονται τα δεδομένων του φορολογικού εντύπου E2 που αφορούν την αναλυτική κατάσταση για τα μισθώματα ακινήτων.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
EstateID	integer	255	false	true
TaxierID	integer	255	false	false
Year	integer	255	false	false
Location	varchar(200)	200	false	false
Position	varchar(100)	100	false	false
EstateUsage	varchar(50)	50	false	false
Area	float	255	false	false
TenantFullName	varchar(50)	50	false	false
TenantAFM	varchar(10)	10	false	false
FromMonth	integer	255	false	false
ToMonth	integer	255	false	false
MonthlyRental	float	255	false	false
PersentCoOwner	float	255	false	false
RevenueFreeHome	float	255	false	false
RevenueFreeOffice	float	255	false	false
RevenueFreeField	float	255	false	false
RevenuePrivateHotel	float	255	false	false
RevenuePrivateOffice	float	255	false	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	EstateID

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
FKD0F1D239A8B43C49	E2	TaxierID, Year
FK_E2_Estate	E2	TaxierID, Year

Ευρετήρια	
Όνομα ευρετηρίου	Όνομα πεδίου
FK_E2_Estate_idx	TaxierID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	EstateID

5.2.17 Δομή πίνακα E2OtherEstate

Στον πίνακα E2OtherEstate αποθηκεύονται τα δεδομένων του φορολογικού εντύπου E2 που αφορούν τα ακίνητα του προηγούμενου οικονομικού έτους τα οποία ήταν ημιτελή ή μεταβιβάστηκαν ή αποκτήθηκαν από αγορά, κληρονομιά, δωρεά, γονική παροχή κτλ.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
ID	integer	255	false	true
TaxpayerID	integer	255	false	false
Year	integer	255	false	false

Title	longtext	2048	false	false
Location	varchar(200)	200	false	false
Position	varchar(100)	100	false	false
EstateUsage	varchar(50)	50	false	false
Area	float	255	false	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	ID

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
FK_E2_OtherEstate	E2	TaxpayerID, Year
FKB050CFAF1A8DBBC8	E2	TaxpayerID, Year

Ευρετήρια	
Όνομα ευρετηρίου	Όνομα πεδίου
FK_E2_OtherEstate_idx	TaxpayerID, Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	ID

5.2.18 Δομή πίνακα IncomeTax

Στον πίνακα IncomeTax αποθηκεύονται τα δεδομένα της εκκαθάρισης φόρου από το φορολογικό έντυπο Ε1 για τον υπόχρεο και τη σύζυγο.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
TaxpayerID	integer	255	false	false
Year	integer	255	false	false
PrincipalTax	float	255	true	false
WifeTax	float	255	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	TaxpayerID, Year

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_IncomeTax_E1	E1	TaxpayerID Year

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	TaxpayerID, Year

5.2.19 Δομή πίνακα RelatePerson

Στον πίνακα RelatePerson αποθηκεύονται τα δεδομένα των συσχετιζόμενων προσώπων με τον υπόχρεο του φορολογικού εντύπου E1.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
idRelatePerson	integer	255	false	true
Type	integer	255	true	false
Address	longtext	500	true	false
AFM	varchar(45)	45	true	false
ContactID	N/D	255	true	false
FName	varchar(45)	45	true	false

LName	varchar(45)	45	true	false
FatherName	varchar(10)	10	true	false
ATID	varchar(10)	10	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	idRelatePerson

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_RelatePerson_Contact	Contact	ContactID
FKF131842E7BE29302	Contact	ContactID

Ευρετήρια	
Όνομα ευρετηρίου	Όνομα πεδίου
fk_RelatePerson_Address_idx	Address
fk_RelatePerson_1_idx	ContactID

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	idRelatePerson

5.2.20 Δομή πίνακα Taxpayer

Στον πίνακα Taxpayer αποθηκεύονται τα προσωπικά δεδομένα του υπόχρεου για το φορολογικό έντυπο Ε1 .

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
ID	integer	255	false	true
UserID	integer	255	false	false
AFM	varchar(10)	10	false	true
FName	varchar(50)	50	false	false
LName	varchar(50)	50	false	false
FatherName	varchar(50)	50	false	false
Contact	N/D	255	false	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	ID

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
fk_taxPayer_Contact	Contact	Contact
FKF43CEAAACD1B0DE7	Contact	Contact
fk_taxPayer_User	User	UserID
FKF43CEAAA509091AA	User	UserID

Ευρετήρια	
Όνομα ευρετηρίου	Όνομα πεδίου
fk_taxPayer_Contact_idx	Contact
fk_taxPayer_User_idx	UserID

Μοναδικό κλειδί	
-----------------	--

Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	ID
AFM_UNIQUE	AFM

5.2.21 Δομή πίνακα User

Στον πίνακα User αποθηκεύονται τα δεδομένα εισόδου στο σύστημα TaxisNet.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
UserID	integer	255	false	true
UserName	varchar(50)	50	false	true
Password	longtext	21844	false	false
Active	integer	255	false	false
OTP	varchar(15)	15	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	UserID

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	UserID
UserName_UNIQUE	UserName

5.2.22 Δομή πίνακα E2coOwner

Στον πίνακα E2coOwner αποθηκεύονται τα δεδομένα των συν-δικαιούχων σε ακίνητα για το φορολογικό έντυπο E2.

Όνομα πεδίου	Τύπος Sql	Μήκος	Nullable	Unique
ID	integer	255	false	true
EstateID	N/D	255	false	false
FullName	varchar(50)	50	false	false
AFM	varchar(10)	10	false	false
Address	varchar(100)	100	false	false
Percent	float	255	false	false
Rent	float	255	true	false

Κύριο κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	ID

Ξένα κλειδιά		
Όνομα κλειδιού	Αναφερόμενος Πίνακας	Όνομα πεδίου
FK7770887AA8611699	E2Estate	EstateID
FK_E2Estate_E2EstateCoOwner	E2Estate	EstateID

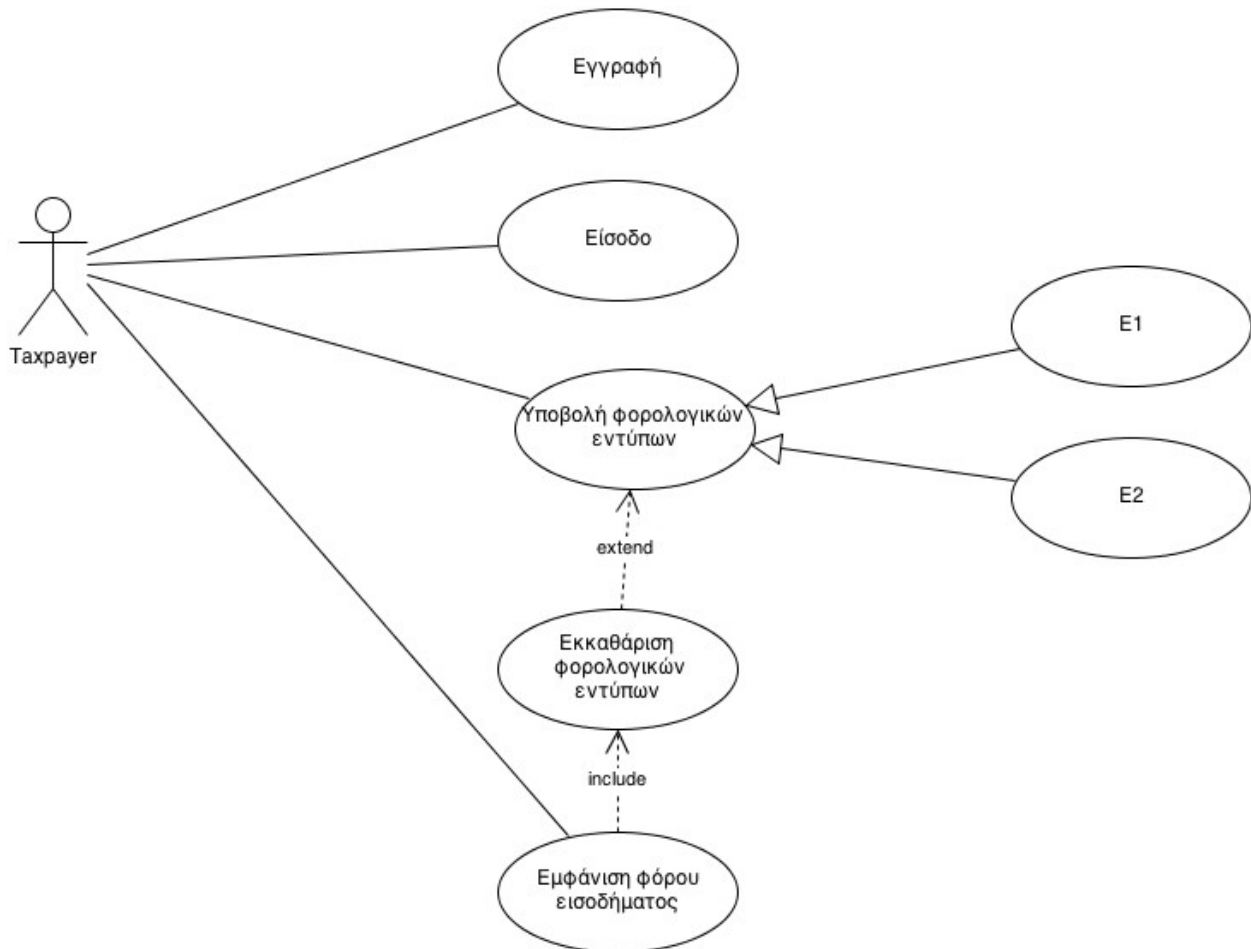
Ευρετήρια	
Όνομα ευρετηρίου	Όνομα πεδίου
FK_E2Estate_E2EstateCoOwner_idx	EstateID

Μοναδικό κλειδί	
Όνομα κλειδιού	Όνομα πεδίου
PRIMARY	ID

5.3 Τεκμηρίωση

5.3.1 Περιπτώσεις χρήσης

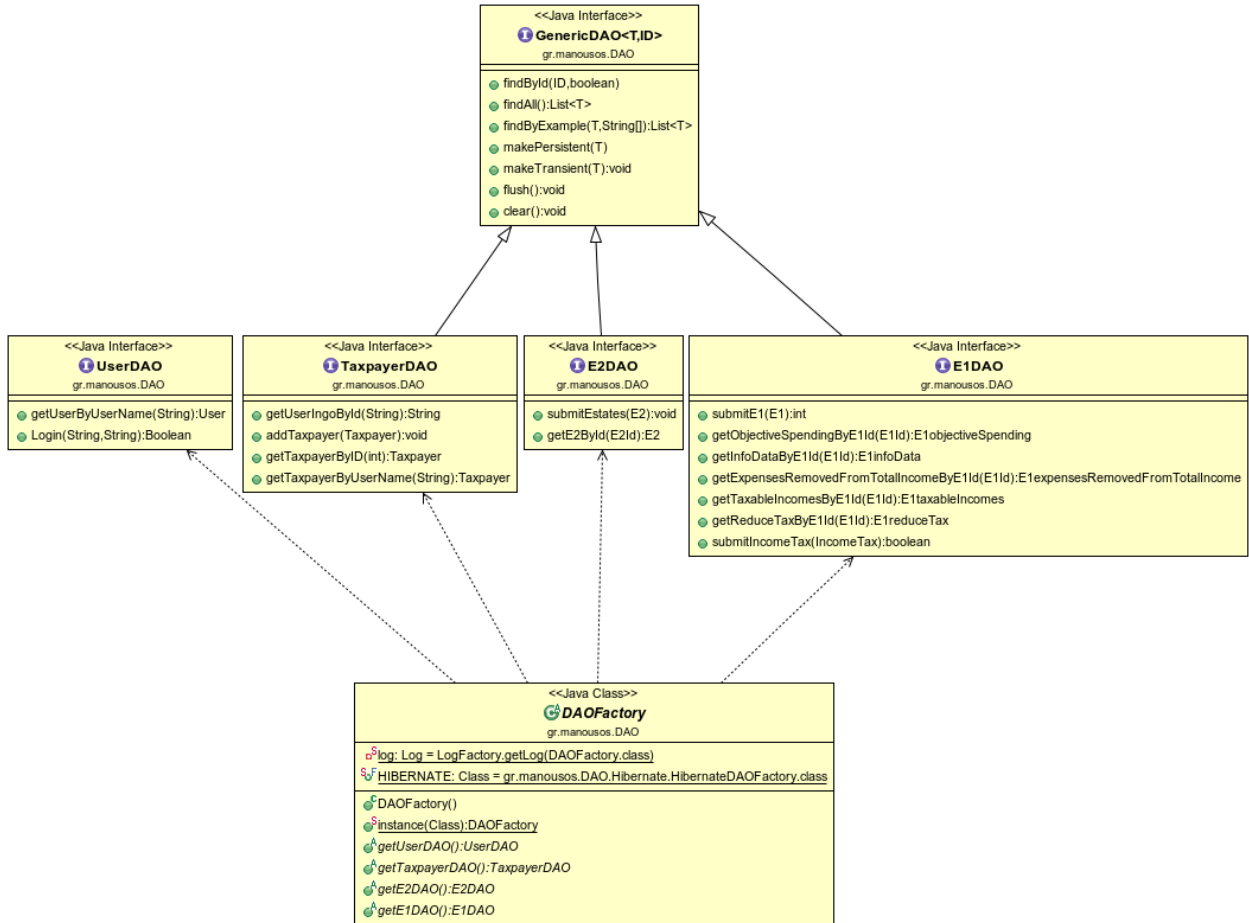
Στην εικόνα 5.7 απεικονίζονται οι λειτουργίες που μπορεί να κάνει ο χρήστης στο TaxisNet.



Εικόνα 5.7 Περιπτώσεις χρήσεις

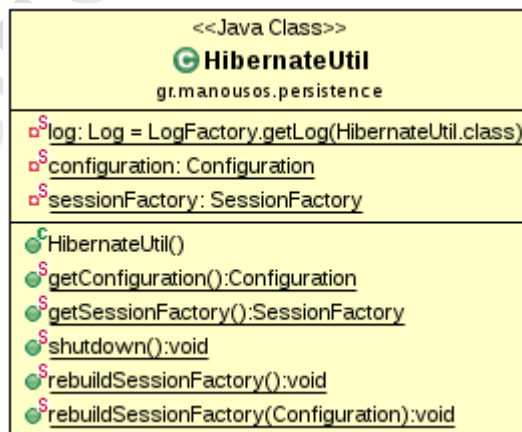
5.3.2 Διαγράμματα τάξεων

Διάγραμμα τάξεων πακέτου DAO



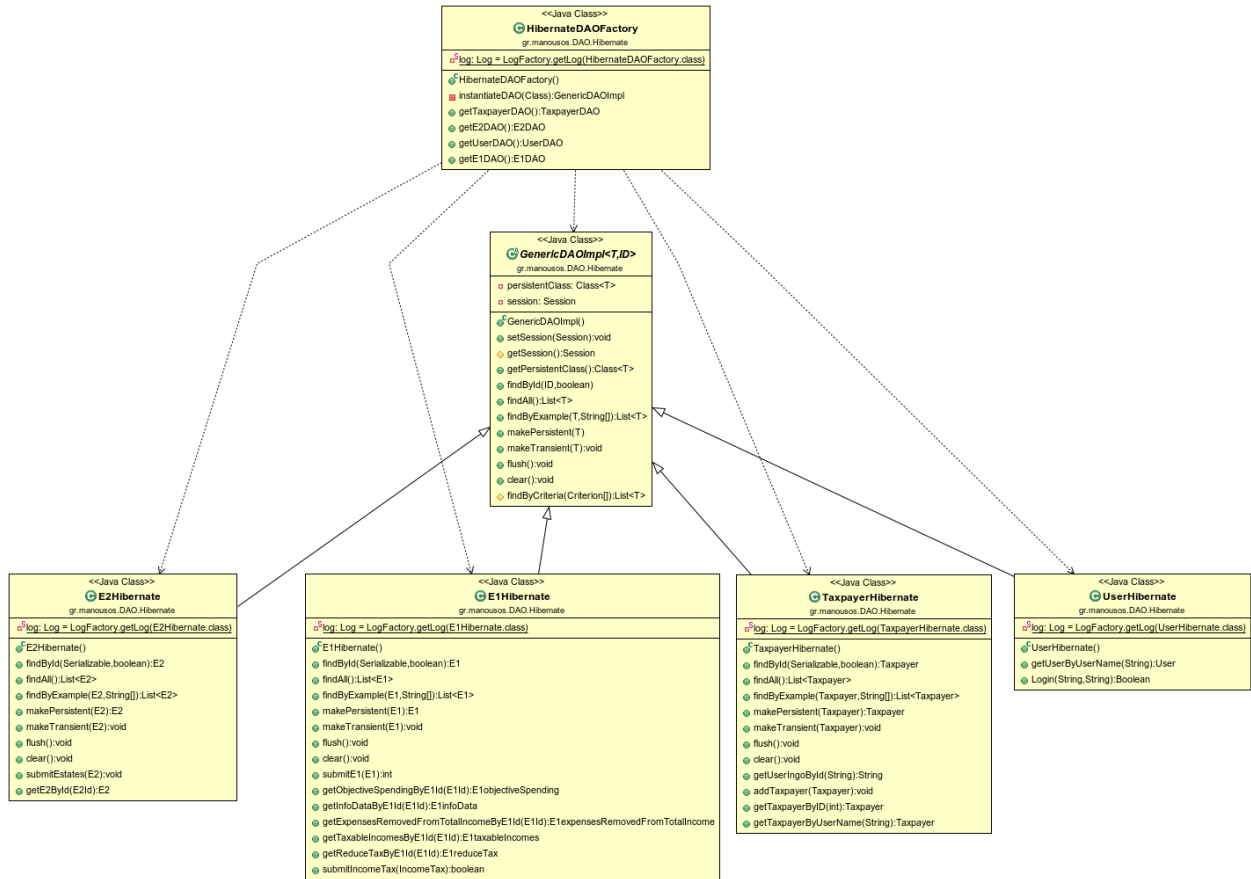
Εικόνα 5.8 Διάγραμμα τάξεων πακέτου DAO

Διάγραμμα τάξεων πακέτου Persistence



Εικόνα 5.9 Διάγραμμα τάξεων πακέτου Persistence

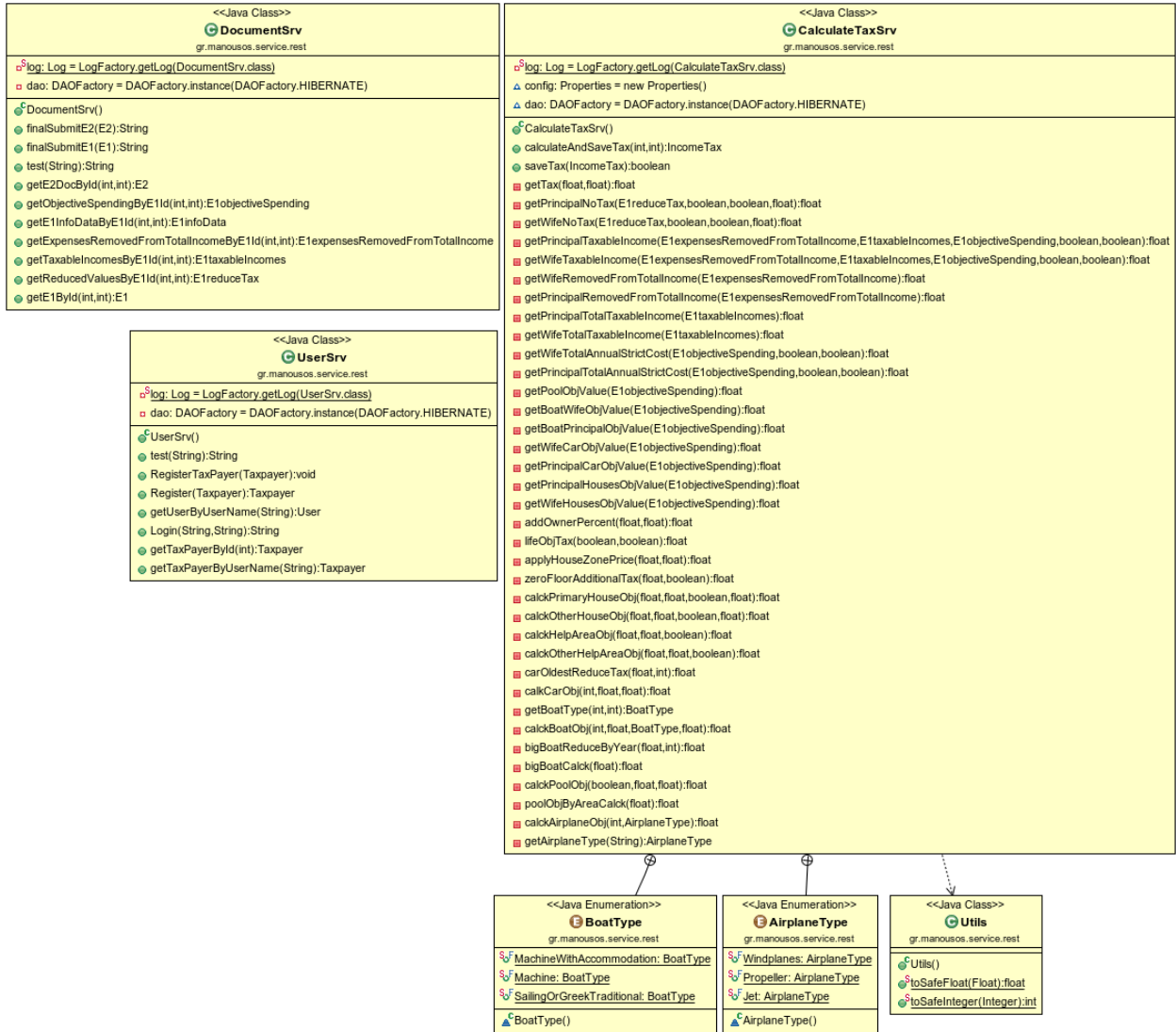
Διάγραμμα τάξεων πακέτου Hibernate



Εικόνα 5.10 Διάγραμμα τάξεων πακέτου Hibernate

Διάγραμμα τάξεων πακέτου rest

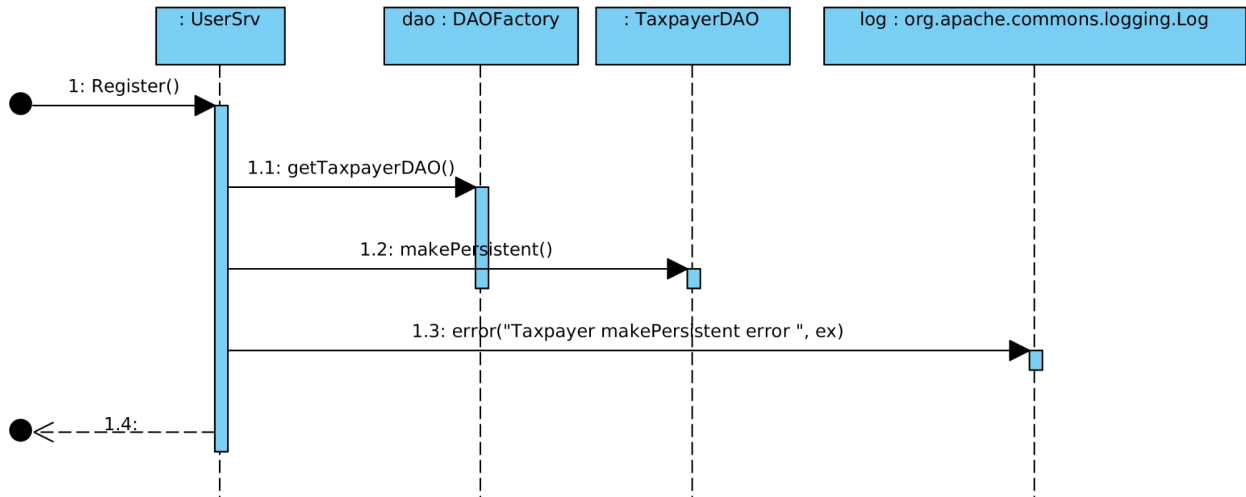
ΠΑΝΕΠΙΣΤΗΜΙΟ



Εικόνα 5.11 Διάγραμμα τάξεων πακέτου rest

5.3.3 Διαγράμματα ακολουθίας

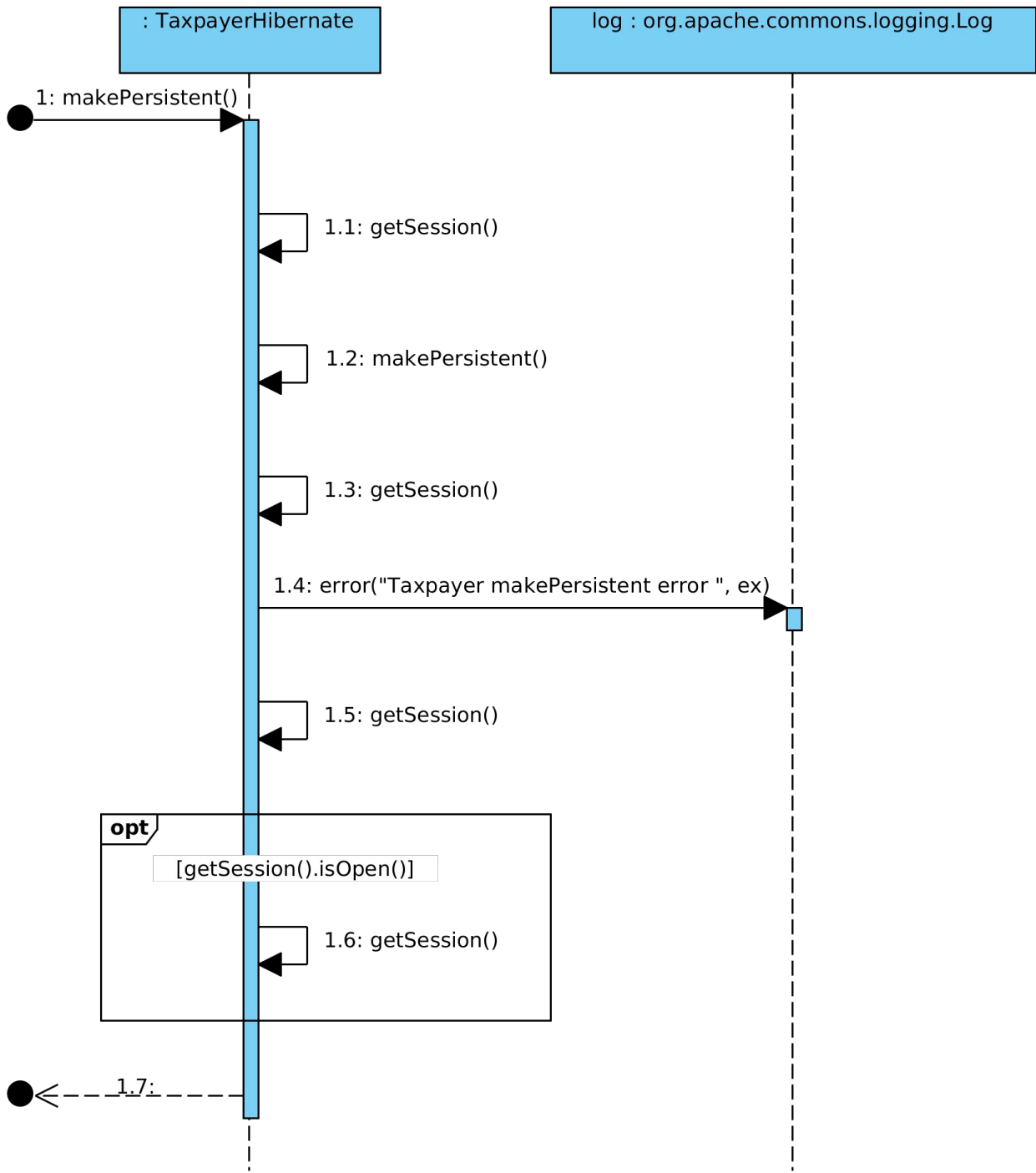
Διάγραμμα ακολουθίας για τη υπηρεσία εγγραφής στο TaxisNet



Εικόνα 5.12 Διάγραμμα ακολουθίας υπηρεσίας εισόδου στο TaxisNet

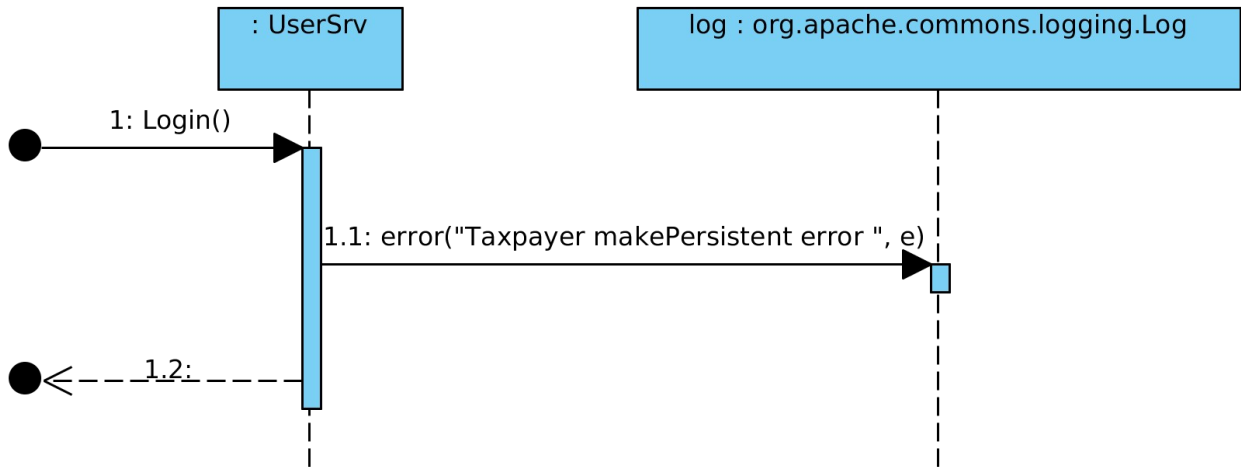
Διάγραμμα ακολουθίας εγγραφής στο TaxisNet (ανάλυση 1.2 makePersistent())

Πανεπιστήμιο Πειραιώς



Εικόνα 5.13 Ανάλυση 1.2 `makePersistent()` εικόνας 5.12

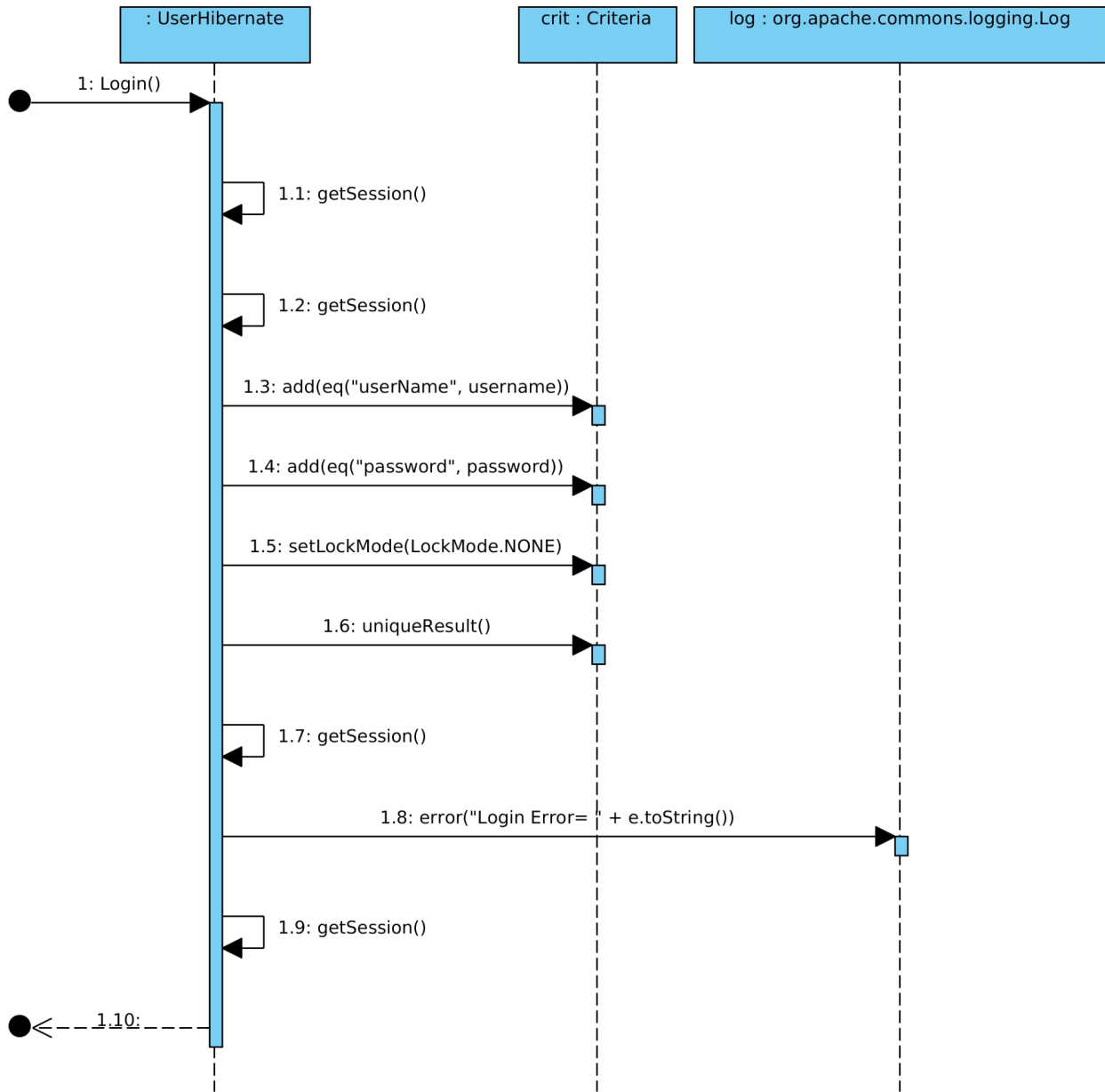
Διάγραμμα ακολουθίας για τη υπηρεσία είσοδο στο TaxisNet



Εικόνα 5.14 Διάγραμμα ακολουθίας υπηρεσίας εισόδου στο TaxisNet

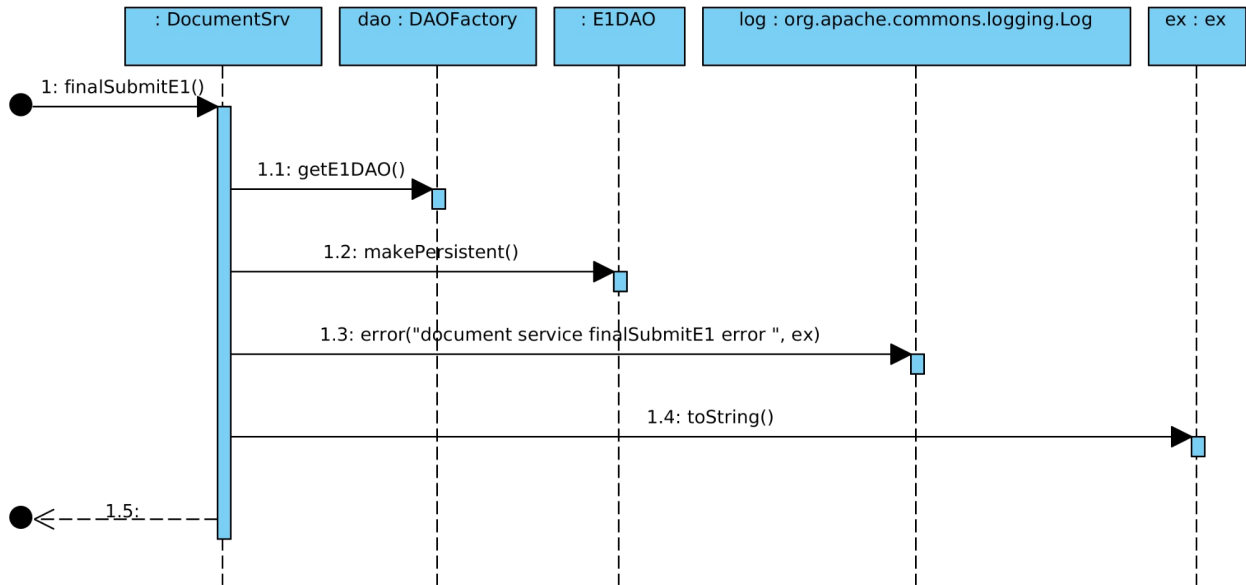
Διάγραμμα ακολουθίας για την λειτουργία Login()

Πανεπιστήμιο Πειραιώς



Εικόνα 5.15 Ανάλυση λειτουργίας Login()

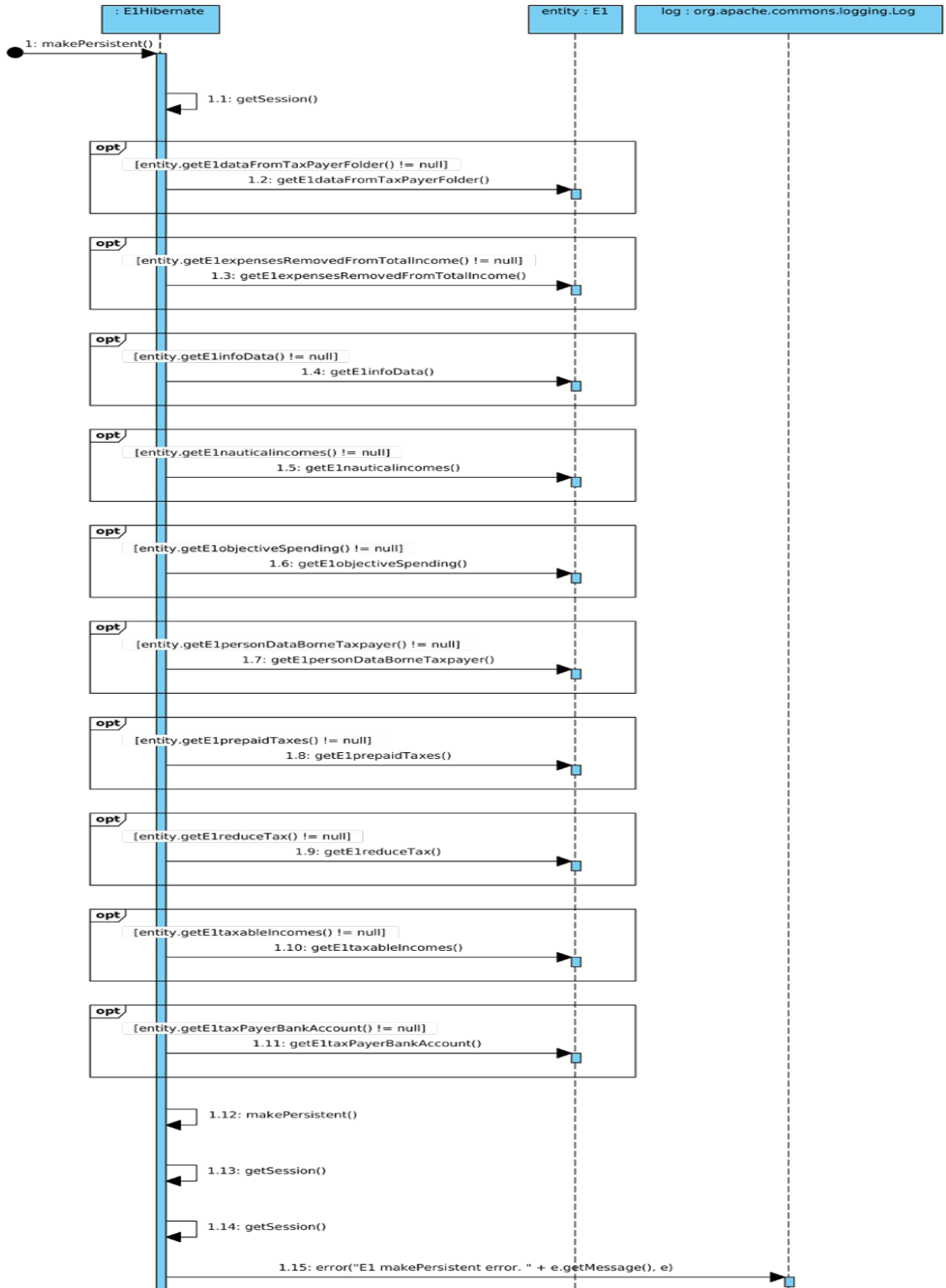
Διάγραμμα ακολουθίας για τη υπηρεσία καταχώρησης φορολογικού εντύπου Ε1



Εικόνα 5.16 Διάγραμμα ακολουθίας υπηρεσίας υποβολής φορολογικού εντύπου E1

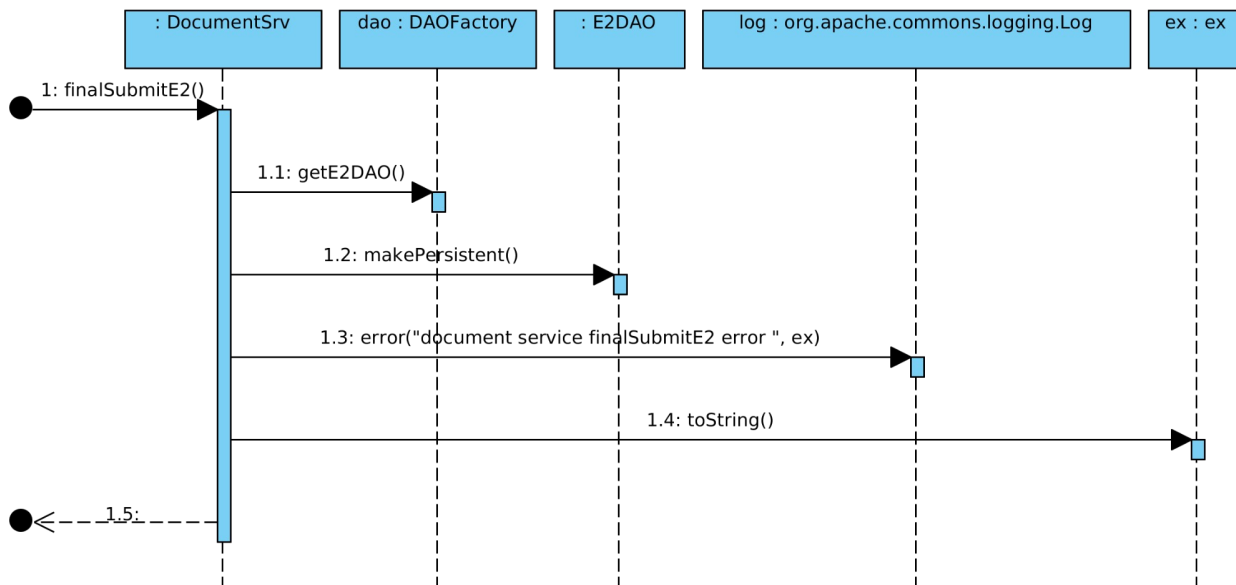
Διάγραμμα ακολουθίας υποβολής φορολογικού εντύπου E1 (ανάλυση 1.2: makePersistent())

Πανεπιστήμιο ΓΠΣ



Εικόνα 5.17 Ανάλυση 1.2 makePersistent() εικόνας 5.16

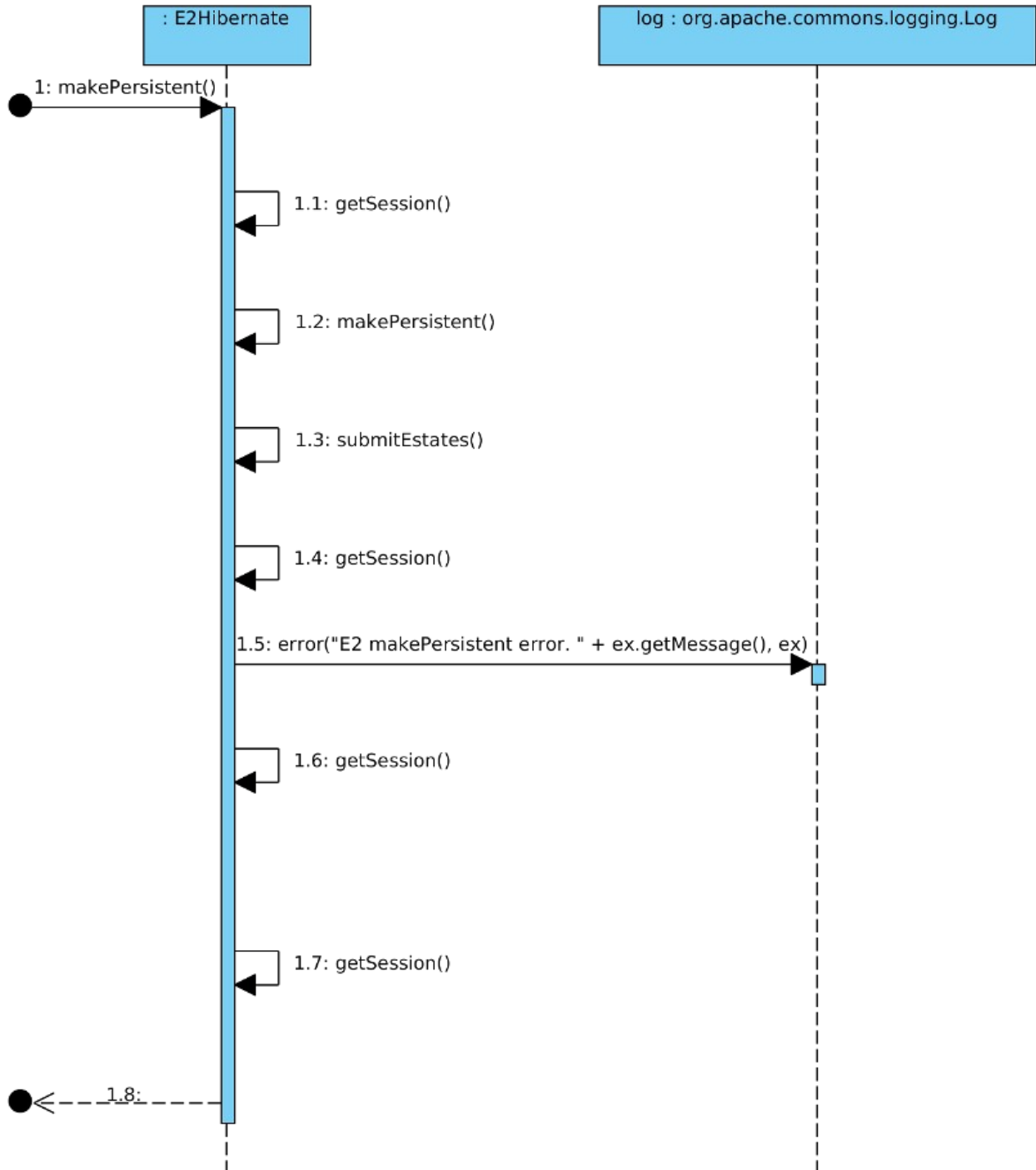
Διάγραμμα ακολουθίας για τη υπηρεσία καταχώρησης φορολογικού εντύπου E2



Εικόνα 5.18 Διάγραμμα ακολουθίας υπηρεσίας υποβολής φορολογικού εντύπου E2

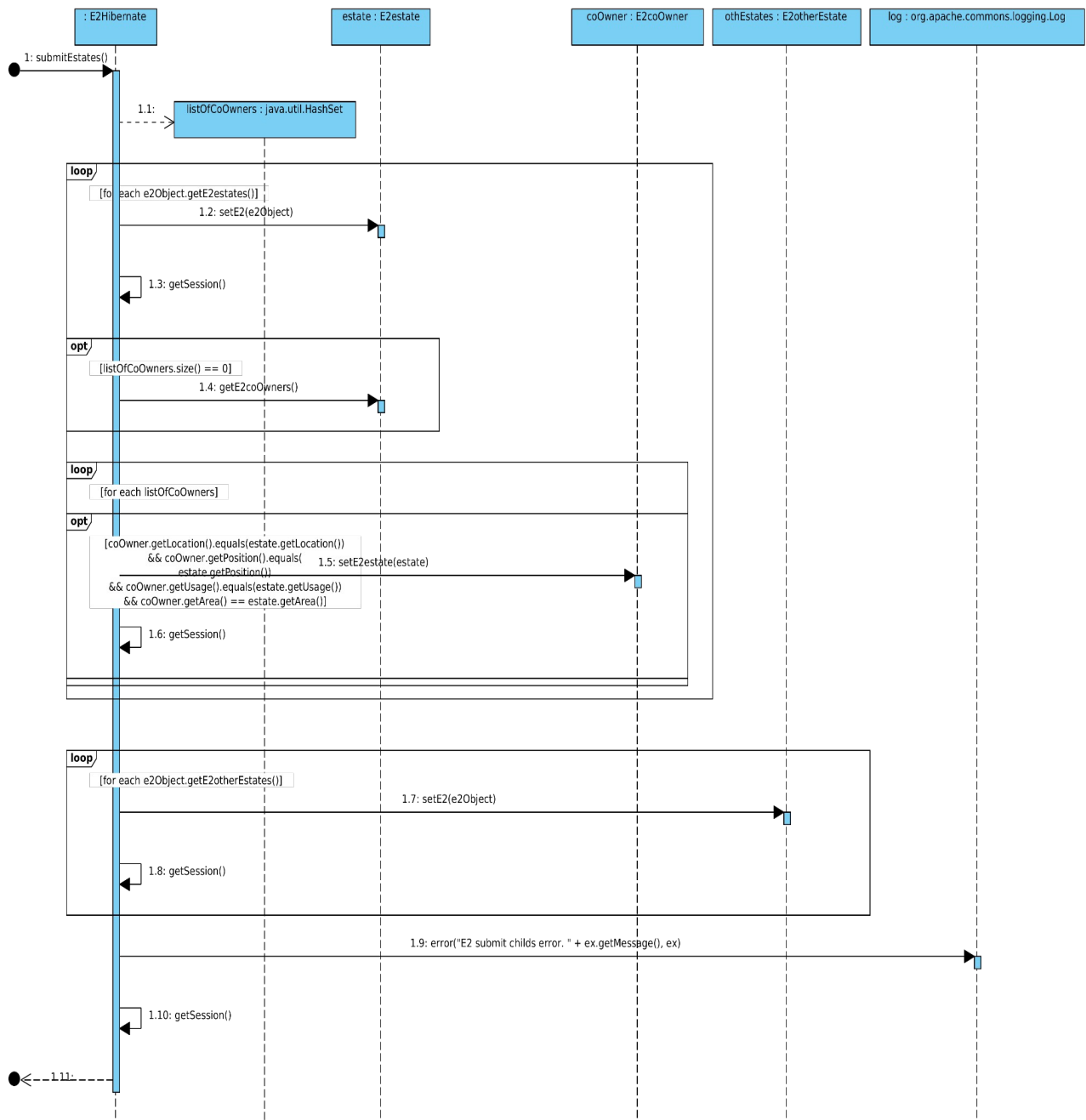
Διάγραμμα ακολουθίας υποβολής φορολογικού εντύπου E2 (ανάλυση 1.2: makePersistent())

Πανεπιστήμιο



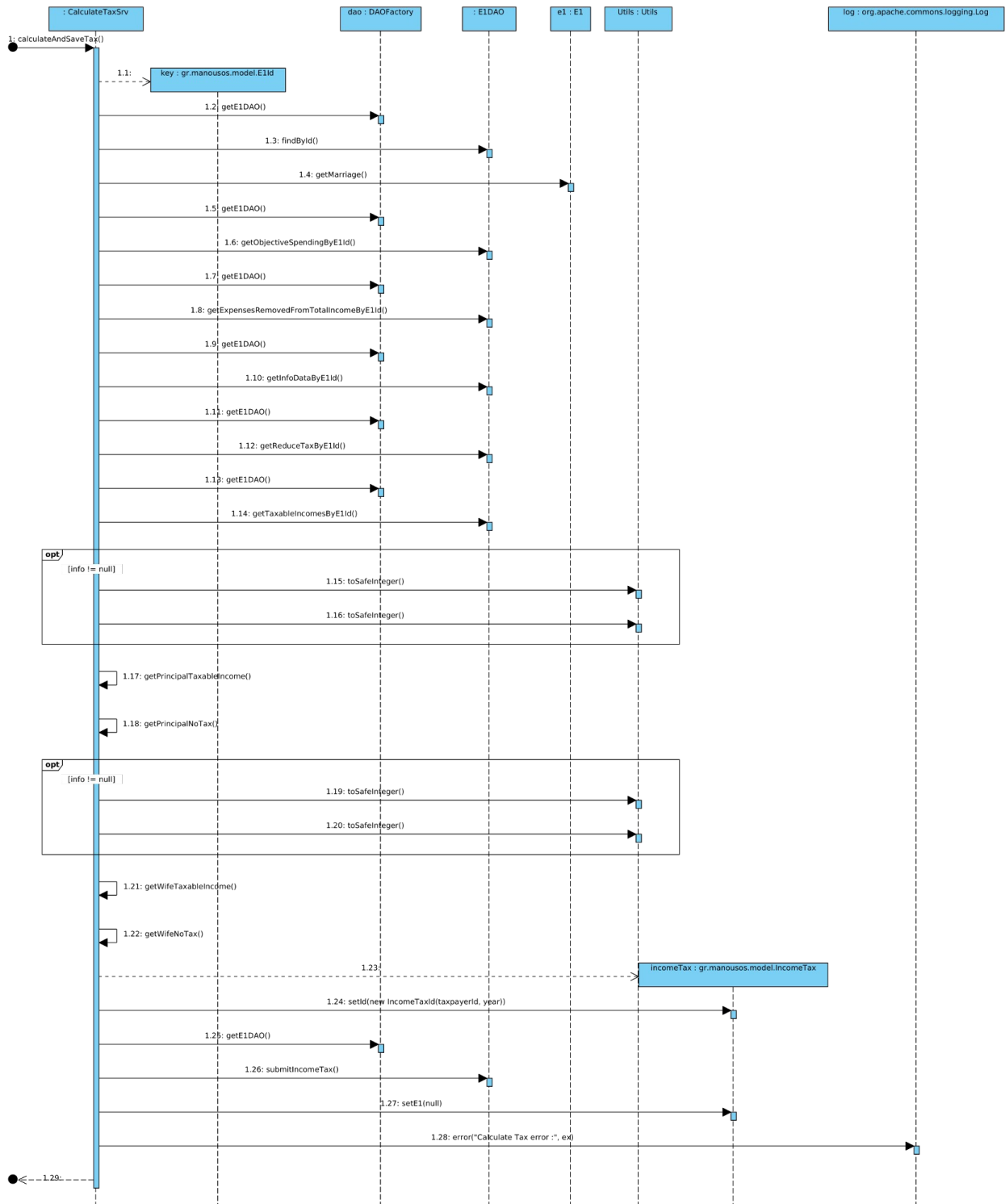
Εικόνα 5.19 Ανάλυση 1.2 makePersistent() εικόνας 5.18

Διάγραμμα ακολουθίας υποβολής φορολογικού εντύπου E2 (ανάλυση 1.3: submitEstates())



Εικόνα 5.20 Ανάλυση 1.3: submitEstates() εικόνας 5.19

Διάγραμμα ακολουθίας για τη υπηρεσία εκκαθάρισης φορολογικού εντύπου E1



Εικόνα 5.21 Διάγραμμα ακολουθίας υπηρεσίας εκκαθάρισης φορολογικού εντύπου E1

5.4 Οθόνες εφαρμογής

Οθόνη εγγραφής χρήστη

Επιλέξτε username και κωδικό

Username

Κωδικός • Πρέπει να εισάγετε Κωδικό

Επιβεβαίωση κωδικού Πρέπει να εισάγετε Επιβεβαίωση κωδικού

Συμπληρώστε τον ΑΦΜ και το ονοματεπώνυμό σας

ΑΦΜ • Πρέπει να εισάγετε ΑΦΜ

Επώνυμο • Πρέπει να εισάγετε Επώνυμο

Όνομα • Πρέπει να εισάγετε Όνομα

Όνομα Πατρός • Πρέπει να εισάγετε Όνομα Πατρός

Συμπληρώστε τα στοιχεία επικοινωνίας σας

E-mail Πρέπει να εισάγετε E-mail

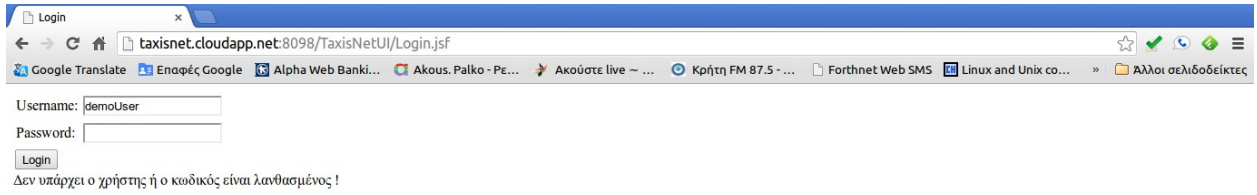
Αρ. τηλεφώνου Πρέπει να εισάγετε Αρ. τηλεφώνου

Αρ. κινητού τηλεφώνου Έχετε εισάγει μη έγκυρους χαρακτήρες, επιτρεπτή μορφή +30691123456789

Fax

Εικόνα 5.22 Οθόνη εγγραφής χρήστη

Οθόνη εισόδου χρήστη



Εικόνα 5.23 Οθόνη εισόδου χρήστη

Οθόνη συμπλήρωσης φορολογικού εντύπου Ε2

Welcome: demoUser

E2 ΑΝΑΛΥΤΙΚΗ ΚΑΤΑΣΤΑΣΗ ΓΙΑ ΤΑ ΜΙΣΘΩΜΑΤΑ ΑΚΙΝΗΤΩΝ ΟΙΚΟΝ. ΕΤΟΥΣ 2012

Εισοδήματα από 1/1 έως 31/12/2011

ΣΤΟΙΧΕΙΑ ΕΙΣΟΔΗΜΑΤΑ (ΦΥΣΙΚΟΥ Ή ΝΟΜΙΚΟΥ ΠΡΟΣΩΠΟΥ)

ΟΝΟΜΑΤΕΠΩΝΥΜΟ Η ΕΠΩΝΥΜΙΑ
 ΟΝΟΜΑ ΠΑΤΕΡΑ
 ΑΦΜ

ΠΕΡΙΓΡΑΦΗ ΑΚΙΝΗΤΟΥ

ΤΟΠΟΘΕΣΙΑ (Οδός-Αριθ.-Πόλη ή Χωριό-Θέση-ΤΚ)
 ΘΕΣΗ (ισόγειο, 1ος όρ., 2ος όρ. κτλ.)
 ΧΡΗΣΗ (οικία,κατ/μα,γραφ.,αγρός κτλ.)
 ΕΠΙΦΑΝΕΙΑ (σε τ.μ. ή στρέμ.)

ΣΤΟΙΧΕΙΑ ΕΝΟΙΚΙΑΣΤΗ

Όνοματεπωνυμο
 Α.Φ.Μ

ΔΙΑΡΚΕΙΑ ΕΝΟΙΚΙΑΣΗΣ

ΑΠΟ
 ΕΩΣ
 ΜΗΝΕΣ
 ΠΡΑΓΜΑΤΙΚΟ Η
 ΤΕΚΜΑΡΤΟ ΜΗΝΙΑΙΟ ΜΙΣΘΩΜΑ
 ΠΟΣΟΣΤΟ ΣΥΝΔΕΣΙΑΣ (%)

Κατοικιών
 Ξενοδοχείων
 Κλινών
 Θεάτρων κτλ.
 Καταστημάτων
 Γραφείων
 Βιομηχανοεγκαταστάσεων
 Αποθηκών κτλ.
 Γηραίων - Γεών
 Οικοδομών σε έδαφος του εκμισθωτή - χώρων επεργασιών κτλ.
 Ξενοδοχείων κτλ.
 Καταστημάτων
 Γραφείων κτλ.

ΣΥΜΠΛΗΡΩΜΑΤΙΚΑ ΣΤΟΙΧΕΙΑ ΑΚΙΝΗΤΩΝ

I. ΕΚΜΙΣΘΟΥΜΕΝΑ ΚΤΛ. ΑΚΙΝΗΤΑ

ΣΤΟΙΧΕΙΑ ΑΚΙΝΗΤΟΥ

ΤΟΠΟΘΕΣΙΑ (Οδός-Αριθ.-Πόλη ή Χωριό-Θέση-ΤΚ)
 ΘΕΣΗ (ισόγειο, 1ος όρ., 2ος όρ. κτλ.)
 ΧΡΗΣΗ (οικία,κατ/μα,γραφ.,αγρός κτλ.)
 ΕΠΙΦΑΝΕΙΑ (σε τ.μ. ή στρέμ.)

ΣΤΟΙΧΕΙΑ ΣΥΝΙΔΙΟΚΤΗΤΩΝ, ΣΥΝΕΠΙΚΑΡΠΩΤΩΝ ΚΤΛ. ΑΚΙΝΗΤΩΝ

Όνοματεπωνυμο
 Α.Φ.Μ
 Διεύθυνση συνιδιοκτήτη
 ΠΟΣ. ΣΥΝΔΕΣΙΑΣ (%)

ΥΠΕΚΜΙΣΘΩΣΕΙΣ

Μίσθωμα που καταβλήθηκε

II. ΑΚΙΝΗΤΑ ΠΟΥ ΤΟ 2011 ΕΙΝΑΙ ΗΜΙΤΕΛΗ Ή ΜΕΤΑΒΙΒΑΣΤΗΚΑΝ Ή ΑΠΟΚΤΗΘΗΚΑΝ ΑΠΟ ΑΓΟΡΑ - ΚΛΗΡΟΝΟΜΙΑ - ΔΩΡΕΑ - ΓΟΝΙΚΗ ΠΑΡΟΧΗ ΚΤΛ.

ΣΤΟΙΧΕΙΑ ΑΚΙΝΗΤΟΥ

ΤΟΠΟΘΕΣΙΑ (Οδός-Αριθ.-Πόλη ή Χωριό-Θέση-ΤΚ)
 ΘΕΣΗ (ισόγειο, 1ος όρ., 2ος όρ. κτλ.)
 ΧΡΗΣΗ (οικία,κατ/μα,γραφ.,αγρός κτλ.)
 ΕΠΙΦΑΝΕΙΑ (σε τ.μ. ή στρέμ.)
 ΤΙΤΛΟΣ ΚΤΗΣΗΣ Ή ΜΕΤΑΒΙΒΑΣΗΣ ΑΚΙΝΗΤΟΥ, ΑΡΙΘΜ. ΣΥΜΒΟΛΑΙΟΥ, ΟΝΟΜ/ΜΟ ΣΥΜΒΟΛΑΙΟΓΡΑΦΟΥ

Εικόνα 5.24 Οθόνη συμπλήρωσης φορολογικού εντύπου E2

Οθόνη συμπλήρωσης φορολογικού εντύπου E1

Welcome: user000636079

Προς τη Δ.Ο.Υ.

ΥΠΕΠΩΝΥΜΟ

Πατριδούχος

Ο ΔΙΕΥΘΥΝΣΗ

Χ

Σ ΕΠΙΘΥΝΩΜΟ

Υ

Ε ΠΙΘΟΥΝΥΜΟ

Κ

Π ΔΙΕΥΘΥΝΣΗ

Π

ΔΗΛΩΣΗ ΦΟΡΟΛΟΓΙΑΣ ΕΙΣΟΔΗΜΑΤΟΣ ΟΙΚΟΝΟΜΙΚΟΥ ΕΤΟΥΣ 2012

ΕΣΟΔΗΜΑΤΑ ΠΟΥ ΑΠΟΚΤΗΘΗΚΑΝ ΑΠΟ 1-1-2011 ΕΩΣ 31-12-2011

ΑΡΙΘΜΟΣ ΚΑΤΑΧΡΗΣΗΣ:

Α.Φ.Μ.

ΑΡΙΘ. ΤΑΥΤ.

ΚΙΝΗΤΟ ΤΗΛΕΦΩΝΟ

ΟΝ. ΠΑΤ.

ΤΗΛΕΦΩΝΟ

ΟΝ. ΠΑΤ.

ΤΗΛΕΦΩΝΟ

ΟΝ. ΠΑΤ.

ΤΗΛΕΦΩΝΟ

Α.Φ.Μ.

ΑΡΙΘ. ΤΑΥΤ.

ΚΙΝΗΤΟ ΤΗΛΕΦΩΝΟ

ΟΝ. ΠΑΤ.

ΤΗΛΕΦΩΝΟ

ΟΝ. ΠΑΤ.

ΤΗΛΕΦΩΝΟ

ΟΝ. ΠΑΤ.

ΤΗΛΕΦΩΝΟ

Α.Φ.Μ. ΕΚΠΡΟΪΟΥ

ΥΠΟΧΡΕΟΥ ΤΗΣ ΣΥΖΥΓΙΟΥ

1. Είστε νέος φορολογούμενος (υποβάλετε δήλωση για πρώτη φορά);
2. Είστε κύριος εξωτερικού και υποχρεούστε να υποβάλετε φορολ. δήλωση στην Ελλάδα;
3. Εξαρτάστε από την προσκόμιση αποδείξεων (έδωχο γ' παρ. 1 άρθρο 9 Κ.Φ.Ε.);
4. Υποβάλλεται η δήλωση από κλημένα σχολάζουσες κληρονόμους, μεσεγγυούχο, προσοφρωτό θυγατρική ή συνώνο πτώχευσης;
5. Υποβάλλεται η δήλωση από επίτροπο, κλημένο ανήλικο ή δικαστικό συμπαραστάτη;
6. Υποβάλλεται η δήλωση από κληρονόμο του φορολογούμενου που απεβίωσε;
7. Είστε μεθωτός και πήρατε στεγαστικό επίδομα μέσα στο 2011;
8. Είστε συνταξιούχος και γεννηθήκατε μέχρι 31-12-1946;
9. Είστε μεθωτός ή συνταξιούχος σε παρωμένες περιουσίες;
10. Γεννηθήκατε από 01-01-1981 και μετά;
11. Έχετε υποπέσει σε παρωμένες των παρ. α' & η' παρ. 8 και παρ. α' και β' παρ. 10 άρθρο 5 ν. 2523/97;
12. Εμπάτε στις διατάξεις της παρ. 5 του άρθρου 73 ν. 3842/2010;
13. Κατοικείτε μόνιμα σε νησί με πληθυσμό κάτω από 3.100 κατοίκους;
14. Μεταβλήθηκε η περιουσιακή σας κατάσταση ή άλλα στοιχεία του Ε9 το 2011 κτλ.;
15. Είστε κύριος γάμος της Ε.Ε. (εκτός Ελλάδας) και ασκητήσατε στην Ελλάδα πάνω από το 90% του συνολ. εισοδ. σας;
16. Έχετε περιουσία (ακίνητα, καταθέσεις κλπ.) στο εξωτερικό;
17. Ανήκατε στη κατηγορία των ούλωχοις, κληρικά αντήρων πάνω από το 80% κτλ.;
18. Είστε αξιωματικός ή ημεδαπό καλύτερο πλήρωμα εμπορ. πλοίου;
19. Δηλώσατε Χ αν δεν εσθιαίστε να ενταχθείτε στην τηλεφωνική και ηλεκτρονική πληροφορία για την εκκαθάριση της δήλωσής σας.

ΗΜΜΕΣ 3. ΑΦΑΙΡΕΣΗ ΠΟΣΟΥ ΛΟΓΩ ΑΝΑΠΗΡΙΑΣ ΚΤΛ ΚΑΙ ΜΕΙΩΣΗ ΦΟΡΟΥ

1. Έχετε αυξημένο φορολόγητο 2.000 ευρώ λόγω αναπηρίας 67% και πάνω κτλ.;
2. Αριθμός παιδιών του Πινάκα 9 παρ. 1 (αριθμητικά)

3. Αριθμός προσώπων του Πινάκα 9 με αναπηρία 67% και πάνω κτλ. (αριθμητικά)

ΗΜΜΕΣ 4. ΦΟΡΟΛΟΓΟΥΜΕΝΑ ΕΙΣΟΔΗΜΑΤΑ

Α. ΕΙΣΟΔΗΜΑ ΑΠΟ ΜΙΣΘΩΤΕΣ ΎΠΗΡΕΣΙΕΣ

ΥΠΟΧΡΕΟΥ	ΤΗΣ ΣΥΖΥΓΙΟΥ
327 ΝΑΙ	328 ΝΑΙ
319 ΝΑΙ	320 ΝΑΙ
023 ΝΑΙ	024 ΝΑΙ
329 ΝΑΙ	
330 ΝΑΙ	
331 ΝΑΙ	
011 ΝΑΙ	012 ΝΑΙ
013 ΝΑΙ	014 ΝΑΙ
015 ΝΑΙ	016 ΝΑΙ
017 ΝΑΙ	018 ΝΑΙ
021 ΝΑΙ	022 ΝΑΙ
025 ΝΑΙ	026 ΝΑΙ
007 ΝΑΙ	008 ΝΑΙ
617 ΝΑΙ	
385 ΝΑΙ	386 ΝΑΙ
029 ΝΑΙ	030 ΝΑΙ
905 ΝΑΙ	906 ΝΑΙ
911 ΝΑΙ	912 ΝΑΙ
010	

ΥΠΟΧΡΕΟΥ ΤΗΣ ΣΥΖΥΓΙΟΥ

001 ΝΑΙ	002 ΝΑΙ
003	004
005	006

ΥΠΟΧΡΕΟΥ ΤΗΣ ΣΥΖΥΓΙΟΥ

1. Αθροισμα καθαρών ποσών από μισθούς, πληροφόρητα κτλ. (εκτός περπτ. 2, 3, 4)
2. Αθροισμα καθαρών ποσών από κύριες συντάξεις (εκτός περπτ. 1, 3, 4)
3. Αθροισμα καθαρών ποσών από επαγγελματικές συντάξεις, μερίσματα, κτλ. (εκτός περπτ. 1, 2, 4)
4. Αποβές αθλήτων και επιδόματα ανέργων των παρ. 2 & 3 του αρθρ. 33 ν. 1892/90

B. ΕΠΙΣΟΛΗΜΑ ΑΠΟ ΤΕΡΠΙΚΕΣ ΕΠΙΧΕΙΡΗΣΕΙΣ

1. Καθαρό εισόδημα ατομικής επιχείρησης βάσει βιβλίων ή στοιχείων	401	402
2. Καθαρό εισόδημα προηγ. περπτ. 1 μόνο από δάλευστη δημοσιότητα	921	922
3. Καθαρό εισόδημα από συμμετοχή σε συνεκμετάλλευση μη υφιστάμενη στο αρθρ. 10 ΚΦΕ	919	920

4. Καθαρό εισόδημα ατομικής επιχείρησης με βάση το αντικειμενικό σύστημα

Περιγραφή (Δημοσίως ελεγχόμενα ή Κοινότητα εκμετάλλευσής)	Είδος παροχής: α) Κεφάλαιους της βρετανικής ζώνης γ) άλλης μονάδας	Αριθμός (1) α) σκευμάτων β) ζώων γ) άλλων μονάδων	Ορισμό Ημιστενό Πεδινό	Απόδομένο Μη απόδομένο	Καθαρό εισόδημα (2) α) ανά στέγασμα β) ανά κερφόλη ζώνη γ) ανά άλλη μονάδα	Συνολικό Καθαρό Εισόδημα (3) = (1) X (2)
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>

Συνολο καθαρών γεωργικού εισοδήματος με βάση το αντικειμενικό σύστημα

5. Καθαρό εισόδημα προηγ. περπτ. 4 μόνο από δάλευστη δημοσιότητα	915	916
Μισοί: α) Ένοικια που καταβλήθηκαν για μίσθωση γεωργικής γης	923	924
β) Δείγια κανονόγνου πύγνου εξοδάτιστοι 326 <input type="text"/> x 25% ή 50%	335	336
γ) Τα κατά κύριο επάγγελμα σπρόδες 1.500 ή 3.000 ευρώ αν πληρο εξοσιστικές αναζημιώσεις ή 2.250 (ή 1.875) ή 4.500 (ή 3.750) ευρώ, αντιστογχα, αν είναι νεοι σπρόδες	337	338
δ Ζημιτά του ίδιου οικονομικού έτους από ατομική άσκηση γεωργικής επιχείρησης	339	340
ε Ζημιτά που απορροφώμενων οικονομικών έτων από ατομική άσκηση γεωργικής επιχείρησης	465	466
ς Ακαθάριστα έσοδα από ατομική άσκηση γεωργικής επιχείρησης	467	468
ζ Ακαθάριστα κέρδη από ατομική άσκηση γεωργικής επιχείρησης	475	476
η Καθαρά κέρδη από συνεσώσεις, πηγές ενέργειας μέγρη 100κ€	479	480
θ Καθαρά κέρδη από γεωροτουριστική μονάδα μέγρη 10 δωμάτια	481	482

F. ΕΠΙΣΟΛΗΜΑ ΑΠΟ ΕΜΠΟΡΙΚΕΣ ΕΠΙΧΕΙΡΗΣΕΙΣ

1. Καθαρά κέρδη από ατομική επιχείρηση (εκτός περπτ. 4)	401	402
2. Επιχειρηματική αμοιβή από Ο.Ε. ή Ε.Ε. ή Κοινωμία κτλ. κληρονομικού δικαίου	403	404
3. Καθαρά κέρδη από Ο.Ε. ή Ε.Ε. ή Κοινωμία κτλ. μη υφιστάμενες στο άρθρο 10 Κ.Φ.Ε.	405	406
4. Υπερτίμημα από πτώλωση αυτοκινήτων επιχείρησης	407	408
5. Ζημιτά του ίδιου οικονομικού έτους από ατομική επιχείρηση (εκτός περπτ. 4)	413	414
6. Ζημιτά προηγούμενων οικονομικών έτων από ατομική επιχείρηση	415	416
7. Ακαθάριστα έσοδα από ατομική επιχείρηση	425	426

A. ΕΠΙΣΟΛΗΜΑ ΑΠΟ ΕΛΕΥΘΕΡΙΑ ΕΠΑΓΓΕΛΜΑΤΑ

1. Καθαρό εισόδημα από ατομικό επάγγελμα	501	502
2. Καθαρές αμοιβές ελεύκτη Α.Ε., διατηρητή, εκτελεστή δαδήςκας κτλ.	503	504

3.Ποσό διατροφής που έχει εισπραχθεί από στήζογο (εκτός των παιδιών)

4.Εισόδημα που δεν εντάσσεται σε άλλη περίπτωση του Πιν. 4

5.Ζημιά του ίδιου οικονομικού έτους από ατομικό επάγγελμα

6.Ζημιές προηγούμενου οικον. έτους από ελεuthέριο επάγγελμα

7.Ακαθάριστα έσοδα από ατομική άσκηση ελεuthέρου επαγγέλματος

Ε. ΕΙΣΟΔΗΜΑ ΑΠΟ ΑΚΙΝΗΤΑ 1. Ακαθάριστο εισόδημα από εκμίσθωση:

α) κατοικιών

β) ξενοδοχείων, κλινικών, σχολείων, αθλοστών κινήων ή θεάτρων κτλ.

γ) καταστημάτων, γραφείων, αποθηκών κτλ.

δ) γηπέδων, χώρων τοποθέτησης επηγραφών

ε) βιομηχανοστασίων, εμπορικών κέντρων που επείξεζαν υπηρωγή σε ΦΠΑ

στ) γαών

ζ) γαών με βάση το αντικειμενικό σύστημα

2.α) Ακαθάριστο εισόδημα από υπεκμίσθωση ακινήτων

β) Στην περίπτωση αυτή, ποίο μίσθωμα έχετε καταβάλει

3. Ακαθάριστο εισόδημα από θωρεάν παραχώρηση - ιδώγηση:

α) κατοικιών (μόνο θωρεάν παραχώρηση)

β) ξενοδοχείων, κλινικών, σχολείων, αθλοστών κινήων ή θεάτρων κτλ.

γ) καταστημάτων, γραφείων, αποθηκών κτλ.

δ) γηπέδων, χώρων τοποθέτησης επηγραφών

ε) γαών (μόνο θωρεάν παραχώρηση)

στ) γαών (μόνο θωρεάν παραχώρηση) με βάση το αντικειμενικό σύστημα

4.Ακαθάριστο εισόδημα οικοδομών: α) σε έδαφος του εκμίσθωτη

β) σε έδαφος του παρεχόντος θωρεάν τη χρήση

5.α) Δαπάνες για ασφαλιστικά, δικαστ. δαπάνη, αμοιβές δικηγόρων (περ. 1α,1β,1γ,1ε,3α,3β,3γ)

β) Αποζημίωση καταβληθείσα με νόμο στο μισθωτή κατά τη λήση επαγγελματικής μίσθωσης

γ) Ακαθάριστο εισόδημα από την εκμίσθωση του ακινήτου της περ.πρ. 5β

δ) Λοιπές περιπτώσεις δαπανών άρθρου 23 ΚΦΕ (γυαές κτλ.)

6.Ακαθάριστο εισόδημα κατοικιών επήφειας πάνω από 300 τ.μ. καθεμιά (περ. 1α, 2α και 3α)

7.Ακαθάριστο εισόδημα ανακατασκευασμένων διατηρητέων (περ 1α, 1β, 3α και 3β)

8.Ακαθάριστο εισόδημα ανακατασκευασμένων διατηρητέων (περ 1γ, 1ε και 3γ)

9.Ακαθάριστο εισόδημα για το οποίο δε βεβαιώνεται χροτόσημο, εκτός από την περ.πρ. 1ε

ΣΤ. ΕΙΣΟΔΗΜΑ ΑΠΟ ΚΙΝΗΤΕΣ ΑΕΙΕΣ

1.Καθαρό εισόδημα από τόκους δανείων κτλ. ημεδαπής προέλευσης

Ζ. ΕΙΣΟΔΗΜΑ ΑΝΟΔΑΠΗΣ ΠΡΟΕΛΕΥΣΗΣ

1.Καθαρό ποσό από μισθούς κτλ. αλλοδαπής προέλευσης

2.Καθαρό ποσό από συντάξεις κτλ. αλλοδαπής προέλευσης

505		506	
507		508	
511		512	
515		516	
517		518	

103		104	
121		122	
105		106	
107		108	
109		110	
101		102	
909		910	
111		112	
113		114	

129		130	
143		144	
145		146	
147		148	
141		142	
701		702	
123		124	
125		126	
151		152	
163		164	
165		166	
159		160	
175		176	
131		132	
133		134	
741		742	

291		292	
-----	--	-----	--

389		390	
391		392	

3. Καθαρά κέρδη από γεωργικές επιχειρήσεις (ατομική, εταιρικές) στην αλλοδαπή	403	464	
4. Ζημιές από γεωργικές επιχειρήσεις (ατομική, εταιρικές) στην αλλοδαπή	471	472	
5. Καθαρά κέρδη από εμπορικές επιγ/σεις (ατομική, εταιρικές) στην αλλοδαπή	411	412	
6. Ζημιές από εμπορικές επιχειρήσεις (ατομική, εταιρικές) στην αλλοδαπή	421	422	
7. Καθαρά κέρδη από ελεύθερο επάγγελμα (ατομικά, εταιρικά) στην αλλοδαπή	509	510	
8. Ζημιές από ελεύθερο επάγγελμα (ατομικά, εταιρικά) στην αλλοδαπή	513	514	
9. Εισοδήματα από κινητές αξίες αλλοδαπής προέλευσης που φορολογούνται με τις γενικές διατάξεις	295	296	
10. Ακαθάριστο εισόδημα από ακίνητα που βρίσκονται στην αλλοδαπή	171	172	
11. Καθαρό εισόδημα της περιτ. 10	173	174	
12. Ακαθάριστο εισόδημα της περιτ. 10 από κατοικίες, με εμφάνεια πάνω από 300 τ.μ. η καθεμιά	395	396	

ΥΠΟΧΡΕΟΥ

ΠΙΝΑΚΑΣ 5. ΠΡΟΣΛΟΠΙΣΜΟΣ ΕΤΗΣΙΑΣ ΑΝΤΙΚΕΙΜΕΝΙΚΗΣ ΑΙΔΑΝΗΣ

1.Ετήσια αντικειμενική δαπάνη διυθμίσεις με βάση: α) Την ετήσια αντικειμενική δαπάνη για ιδιοκατοικημένη ή μισθωμένη ή δωρεάν παρ/γωρούμενη κύρια κατοικία και έως δύο δευτερεύουσες κατοικίες.

ΣΤΟΙΧΕΙΑ ΑΚΙΝΗΤΟΥ ΔΙΕΥΘΥΝΣΗ - ΟΔΟΙ ΠΟΥ ΠΕΡΙΛΑΜΒΑΙΝΟΥΝ ΤΟ ΤΕΙΡΑΙΩΜΟ	ΑΡΙΘΜΟΣ ΠΑΡΟΧΗΣ ΡΕΥΜΑΤΟΣ	ΜΙΣΘΩΜΕΝΗ ΚΑΤΟΙΚΙΑ Η ΔΩΡΕΑΝ	ΜΟΝΟΚΑ- ΤΟΙΚΙΑ	ΘΕΣΗ ΟΡΘΟΟΣ	ΕΠΙΘΑΝΕΙΑ ΚΥΡΙΩΝ ΧΩΡΩΝ	ΕΠΙΘΑΝΕΙΑ ΒΟΗΘΗΤΙΚΩΝ ΧΩΡΩΝ	ΠΟΣΟΣΤΟ ΣΥΝΔΙΟΚΤΗΣΙΑΣ Ή ΧΡΗΣΗΣ	ΜΗΝΕΣ ΙΑΟΚΑ- ΤΟΙΚΗΣΗΣ	ΤΙΜΗ ΖΩΝΗΣ ή ΤΙΜΗ ΕΚΚΙΝΗΣΗΣ
α) ΚΥΡΙΑ ΚΑΤΟΙΚΙΑ	205	203	240	211	212	213	214	215	216
		NAI	NAI						
β) ΠΡΩΤΗ ΔΕΥΤΕΡΕΥΟΥΣΑ	207	207	241	218	219	220	221	222	223
		NAI	NAI						
γ) ΔΕΥΤΕΡΗ ΔΕΥΤΕΡΕΥΟΥΣΑ	209	209	242	225	226	227	228	229	230
		NAI	NAI						

β) Η ετήσια αντικειμενική δαπάνη για λοιπές δευτερεύουσες κατοικίες
 γ) Για επιβατικά οχήματα Ι.Χ., Μ.Χ. (οικονομικές, ατομικές-εταιρικές επιγ/σεις, κοινωνικών και κοινοπραξιών)

Α/Α Α.Φ.Μ. κυρίου ή κυράγου Ε.Ι.Χ. Μ.Χ.Κ.Α.	Κ.Α. κατοικίας	Κ.Α. κατοικίας	Κ.Α. κατοικίας	Κ.Α. κατοικίας	Κ.Α. κατοικίας	Κ.Α. κατοικίας	Κ.Α. κατοικίας	Κ.Α. κατοικίας	Κ.Α. κατοικίας	
										Στοιχεία Γράμματα Αριθμός
1.	750	703	761	771	775				851	852
2.	751	704	762	772	776				853	854
3.	752	705	763	773	777				855	856
4.	753	706	764	774	778				857	858

Όνομα σκύλου	Αριθμός και λήξιμη ημερομηνία	Χώρα	Ισοποφία	Χώροι	Ποσοστό συνόλ. ενόχλησης υτόχγου	Πρώτη ημερομηνία	Μήνες κυρ/τας	Κ.Α. κατοικίας	Μέτρα μίσθωσης	
			NAI	NAI					711	712
			NAI	NAI					713	714

δούλες αμοιβές πληρωμάτων οκτωβρών αντανύξης
 ε)τα αεροσκάφη, ελκυστρερα Ι.Χ. και ανεμόστρερα

Στοιχεία εθνικότητας και ηηολόγητης: Τύπος Αριθμ. σειράς/Αερολιμένας συνήθους
 κωδίστη Μήνας κυρ'τας μέσου στο 2011 Τίτρον ή Δίμηνας
 ηηολόγ.

στ)τις δεξιμενές κολύμβητης (τισίτες)

Κ.Α	Εξωστρεική (τερ. μέτρο)	Ποσοστό συνδιοκρησίης % υπόγρεου	Κ.Α	Εσωστρεική (τερ. μέτρο)	Ποσοστό συνδιοκρησίης % υπόγρεου
767	<input type="text"/>	<input type="text"/>	768	<input type="text"/>	<input type="text"/>

ζ)Γράψτε την αντικειμενική διάσταση για διδάκτρα σε σχολεία

η)Γράψτε την αντικειμενική διάσταση για οκτακούς βοηθούς, οδηγούς αυτ'των και δασκάλους

2.Απο τη διάστηη που κερτάβιάατε:

α)Τα την αγροπή ή χρημηατοδοστηκή μίεθλοση αυτοκνητήτων, δίτρογων κτλ. ορημήατων

β)Τα την αγροπή ή χρημηατοδοστηκή μίεθλοση πάλσιων αντανύξης κτλ. σκαφών αντανύξης και αεροσκαφών

γ)Τα την αγροπή ή χρημηατοδοστηκή μίεθλοση κνητήτων ηρηρήατων άξις πάνω από 10.000 ευρώ
 ό)για δωρεές ή γωνικές παρηγές ή χρηρηγίες χρημηατικών ποσών (εκτός προς το Δήμοστο, κτλ.)
 ε)για την τοκογρεολογηκή απόδωβση διδώνων ή πασίωσεων κτλ.

ηηικασε α ΠΡΟΣΘΕΤΑ ΠΗΗΡΟΦΟΡΗΑΚΑ ΣΤΟΙΧΕΙΑ - ΠΟΣΑ ΠΟΥ ΜΕΙΩΝΟΥΝ ΤΗΝ ΕΤΗΕΙΑ ΔΑΠΗΝΗ

1.Καθαρό εισόδημα που δεν υπήρηγε την 1-1-2012

2.Ποσό ετήσιως διάστηης που δεν υπήρηγε την 1-1-2012

3.Εισοδήματα που αποκρησσε το 2011, τα οποία απαλίωσσενα από το φόρο, φορολογώντα με ειδικό τρόπο, καθός και από μερήαματα ημεδιστών Α.Ε., ΕΠΕ, κτλ.

4.Εισοδήματα που εξαμρώντα της εισφορός (ααρ. 2 άρθρ. 29 ν. 3986/2011)

5.Επίδομα ανεργίης του Ο.Α.Ε.Δ.

6.Καθαρά κέρδη από εταρίες κτλ., που φορολογήρηκαν με το άρθρο 10 Κ.Φ.Ε.

7.Φόρος που παρηκαρηστήρηκε στα εισοδήματα της περ. 3 και στα καθαρά κέρδη της περ. 6

8.Καθαρό ποσό επιδότητος κωνωνικής ώλιη/ετήης (ΕΚΑΣ)

9.Αγροτικές ενισχύσεις (επιδότ., ασόζημ., οκον.ενισχ. κτλ.) εκτός αυτών επί της παρησώρης

0.Ενοίκιο εγκαίωστασης ατομηκής εταγείρης ή ελεθέρου εταγείματος

Ονομασπώνημο ή Εταωνυμία εκμωθιστή	Κ.Α.	Α.Φ.Μ. εκμωθιστή	Κ.Α.	Επιφάνεια σε τ.μ.
<input type="text"/>	790	<input type="text"/>	795	<input type="text"/>
<input type="text"/>	791	<input type="text"/>	796	<input type="text"/>
<input type="text"/>	417	<input type="text"/>	797	<input type="text"/>

11.Ενοίκιο ακνητηων εκτός από κήρια κητοική, κητοική παθών που σπουδίζουον και περήτ. 10

12.Δαστήη για την αγροπή ή ανεργεση ακνητηων ή χρημηατοδοστηκή μίεθλοση αυτών κτλ.

13.Χρημηατικά ποσά που παρέρηγοντα από διδωβση παρηουσιακών στοιχείων, εισαγωγή χρημηατικών κεφάλαιων ώλιη/εταγής, δάνεια, δωρεές κτλ.

14.Εταωνιασπείρημενα κεφάλια (άρθρ. 18ν.3842/10)

15.Ανώλωση κεφάλαιου που ήδη φοροδότηρηκε ή απαλίωσσενα από το φόρο

731	<input type="text"/>	732	<input type="text"/>
715	<input type="text"/>	716	<input type="text"/>
705	<input type="text"/>	706	<input type="text"/>
709	<input type="text"/>	<input type="text"/>	
770	<input type="text"/>	<input type="text"/>	
719	<input type="text"/>	720	<input type="text"/>
721	<input type="text"/>	722	<input type="text"/>
723	<input type="text"/>	724	<input type="text"/>
725	<input type="text"/>	726	<input type="text"/>
727	<input type="text"/>	728	<input type="text"/>
ΥΠΟΧΡΕΟΥ			
655	<input type="text"/>	656	<input type="text"/>
693	<input type="text"/>	694	<input type="text"/>
659	<input type="text"/>	660	<input type="text"/>
657	<input type="text"/>	658	<input type="text"/>
661	<input type="text"/>	662	<input type="text"/>
431	<input type="text"/>	432	<input type="text"/>
433	<input type="text"/>	434	<input type="text"/>
305	<input type="text"/>	306	<input type="text"/>
477	<input type="text"/>	478	<input type="text"/>
ΤΗΣ ΣΥΖΥΓΙΟΥ			
793	<input type="text"/>	794	<input type="text"/>
615	<input type="text"/>	616	<input type="text"/>
419	<input type="text"/>	420	<input type="text"/>
735	<input type="text"/>	736	<input type="text"/>
781	<input type="text"/>	782	<input type="text"/>
783	<input type="text"/>	784	<input type="text"/>
787	<input type="text"/>	788	<input type="text"/>
ΥΠΟΧΡΕΟΥ			
ΤΗΣ ΣΥΖΥΓΙΟΥ			

1. Δαπάνη αγοράς οχημάτων και παροχής υπηρεσιών (παρ. 1 άρθρο 9 Κ.Φ.Ε.)

2. Εξόδα υπηρετικής και νοσοκομειακής περιθάλψης

3. Υποχρεωτικές εισφορές σε τιμεία ασφαλισής (Ο.Α.Ε.Ε. κτλ.)

4. Δαπάνες υπηρετικών μηχανημάτων κτλ., σε νοσοκομεία κτλ.

5. Δαπάνες χηλημ. ποσών στο Δημόσιο κτλ.: Στο εξωτερικό 031 032 και συνολικά

6. Δαπάνες χηλημ. ποσών στο λογισμισμό αλληλοβοήθειας για την απόβλεση του Δημοσίου χρέους

7. Πλοιστατικές χορηγίες του ν.3525/2007:

8. Δεδουλειμμένοι τόκοι που καταβλήθηκαν για στεγαστικά δάνεια απόκτησης πρώτης κατοικίας και αναστήλωσης κτλ.. διατηρητέων κτηρίων:Επιφάνεια σε τ.μ. 071

9. Συνολικό ποσό δανείων της προηγούμενης περιόδου της

10. Ενοίκιο που πλήρωσατε για κτήρια κατοικίας της οικογένειας:

Ονοματεπώνυμο ή Επωνυμία εκμισθωτή Κ.Α.Α.Φ.Μ. εκμισθωτή:Κ.Α.Φιλολογικά.Α.Ε. Επιφάνεια Κ.Α. Μήνες σε τ.μ.

801 092 091 097

802 094 093 098

803 096 095 099

11. Ενοίκιο που πλήρωσατε για κατοικία παιδιών της οικογένειας που σπουδάζουν στο εξωτερικό:

804

805

806

807

12. Ποσό ασφαλίσεων για ασφάλιση ζωής, προσωπικών ατυχημάτων, ασθένειας

13. Ποσό δαπάνης που καταβλήσε ο ένας σύζυγος στον άλλον

14. Δαπάνη για άμεση εγκατάσταση κινωπίου ή για εγκατάσταση φυσικού αερίου κτλ.

15. Δαπάνη σε φροντιστήρια εκπαιδευτικών μαθημάτων ή ξένων γλωσσών:

α) Για τον υπόχρεο και τη σύζυγο 081

β) Για κάθε παιδί τους: χορηγία 082

083

16. Μισθώματα επιχειρήσεων τριτογενούς τομεία (παρ. Β2 άρθρ.43 και 44 ν.4030/2011)

17. Ποσό εισόδου για κινηματογραφικές ταινίες (παρ. 9 άρθρ. 73 ν. 3842/2010)

18. Εισιτήρια στις διατάξεις των περιπτώσεων α ή β της παρ. 9 του άρθρου 73 ν. 3842/2010:

μηναίοι & ΠΡΟΚΑΤΑΒΗΘΕΝΤΕΣ - ΠΑΡΑΚΑΤΗΘΕΝΤΕΣ ΦΟΡΟΙ

1. Φόροι 4%, 10% και 15% που προκαταβλήθηκαν (άρθρο 52 Κ.Φ.Ε.)

2. Φόροι 1%, 4%, 8%, 15% και 20% που παρακρατήθηκαν (άρθρο 55 Κ.Φ.Ε.)

3. Φόροι 20% και 10% (άρθρο 58 Κ.Φ.Ε.) και 15% (παρ. 7 παρ. 5 άρθρ. 52 Κ.Φ.Ε.) που παρακρατήθηκαν

4. Φόρος που καταορίστηκε για ωφέλεια από πώληση αυτοκινήτου επιχείρησης

5. Φόρος που παρακρατήθηκε στις αμοιβές της παρ. 4 του Π.ν. 4Α

6. Φόρος που καταβλήθηκε στο εξωτερικό

7. Φόρος που παρακρατήθηκε στα εισοδήματα από τόκους δανείων κτλ. ημεδαμής (Π.ν. 4 ΣΤ)

8. Φόρος που αναλογεί στους μισθούς, συντάξεις (παρ. 1, 2 και 3 του Π.ν. 4Α)

049

051

053

057

059

075

061

063

069

052

054

058

060

076

062

064

070

811

813

815

812

814

816

817

819

821

823

073

089

087

079

084

077

663

033 α 035 β

034 α 036 β

ΥΠΟΧΡΕΟΥ

601

603

605

607

609

651

293

313

602

604

606

608

610

652

294

314

1ο παιδί

2ο παιδί

3ο παιδί

074

090

088

080

085

078

664

THE ΣΥΖΥΓΙΟΥ

602

604

606

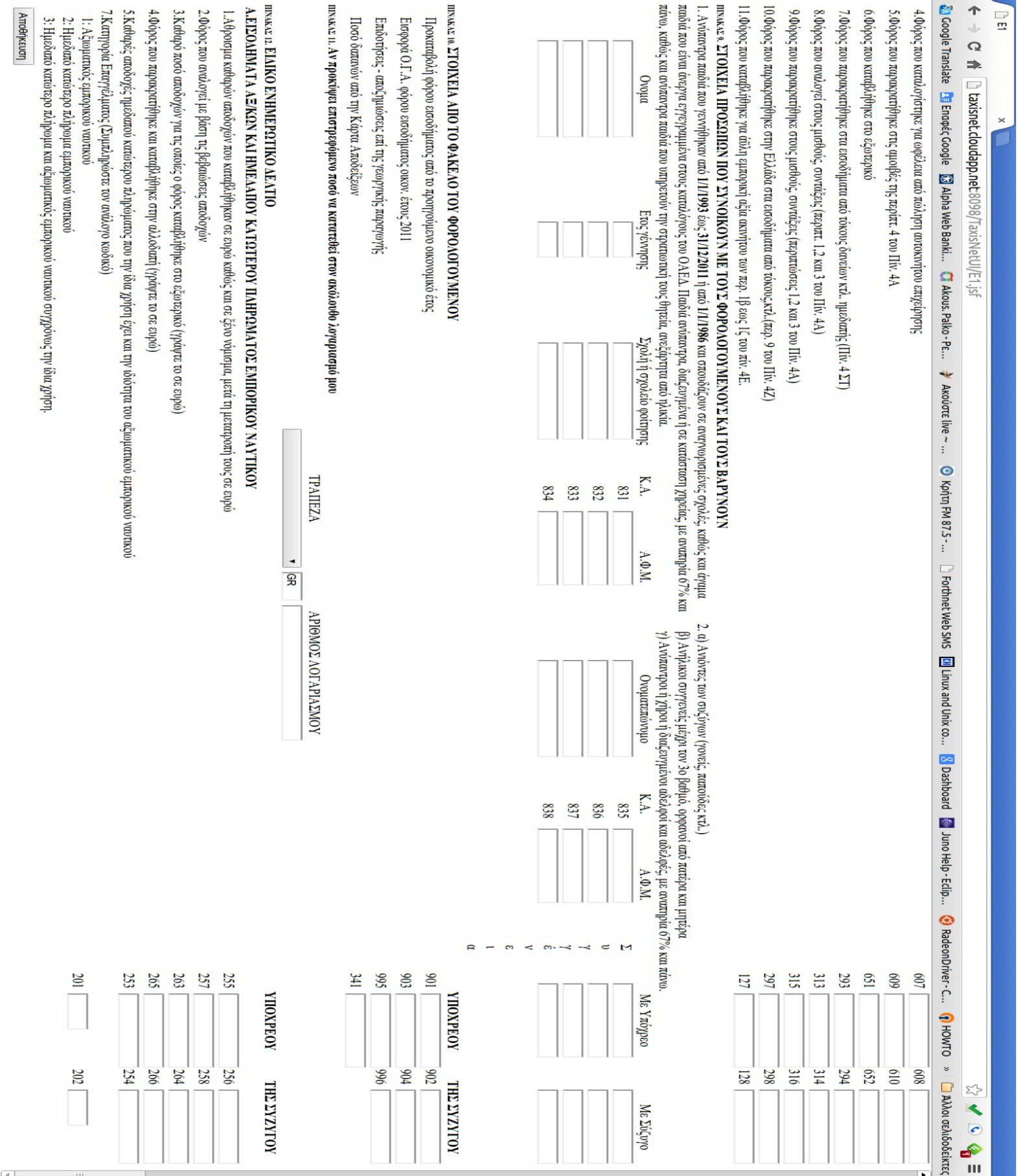
608

610

652

294

314



Εικόνα 5.25 Οθόνη συμπλήρωσης φορολογικού εντύπου Ε1

Οθόνη εμφάνισης φόρου εισοδήματος

Εκκαθάριση φόρου εισοδήματος	
Φόρος κύριου:	879.10€
Φόρος συζύγου:	0.0€
Σύνολο:	879.10€

Εικόνα 5.26 Οθόνη εμφάνισης φόρου εισοδήματος

Πανεπιστήμιο

Κεφάλαιο 6

Σε αυτό το κεφάλαιο παρουσιάζονται και αναλύονται τα αποτελέσματα από τα τρία σενάρια δοκιμών που περιέγραψα στο κεφάλαιο 4.

6.1 Αποτελέσματα Σεναρίου 1

Συνολικός αριθμός σεναρίων προς εκτέλεση ²²	175000
Πλήθος τερματικών	25
Πλήθος σεναρίων εκτέλεσης ανά τερματικό	7000
Στάλθηκαν ²³	Μη διαθέσιμο
Μέγιστος αριθμός ταυτόχρονων συνδέσεων στην βάση δεδομένων	100
Αποτυχημένες συνδέσεις στην βάση δεδομένων	Μη διαθέσιμο
Εγγεγραμμένοι χρήστες	72615
Υποβολή εντύπων E1 ή και E2	72077
Εκκαθάριση εντύπων E1	72018

Ποσοστό επιτυχών εγγεγραμμένων χρηστών στο σύστημα	41,50%
Ποσοστό επιτυχών καταχωρήσεων φορολογικών εντύπων (από τους εγγεγραμμένους χρήστες)	99,26%
Απόκλιση από το επιθυμητό πλήθος καταχωρήσεων φορολογικών εντύπων	58,81%
Ποσοστό επιτυχών εκκαθαρίσεων E1 (από τα καταχωρημένα έντυπα)	99,18%
Απόκλιση από το επιθυμητό πλήθος εκκαθαρίσεων	58,85%

6.2 Αποτελέσματα Σεναρίου 2

²² ο συνολικός αριθμός σεναρίων περιλαμβάνει τις λειτουργίες του TaxisNet όπως περιγράφονται στο κεφάλαιο 4

²³ ο συνολικός αριθμός αιτήσεων (τουλάχιστον εγγραφής στο σύστημα) πριν καταρρεύσει ο υπολογιστής ή ο δικτυακός εξοπλισμός

Συνολικός αριθμός σεναρίων προς εκτέλεση	168000
Πλήθος τερματικών	24
Πλήθος σεναρίων εκτέλεσης ανά τερματικό	7000
Στάλθηκαν	18777
Μέγιστος αριθμός ταυτόχρονων συνδέσεων στην βάση δεδομένων	124
Αποτυχημένες συνδέσεις στην βάση δεδομένων	76
Εγγεγραμμένοι χρήστες	18755
Υποβολή εντύπων E1 ή και E2	17689
Εκκαθάριση εντύπων E1	17564

Ποσοστό επιτυχών εγγεγραμμένων χρηστών στο σύστημα	11,16%
Ποσοστό επιτυχών καταχωρήσεων φορολογικών εντύπων (από τους εγγεγραμμένους χρήστες)	94,31%
Απόκλιση από το επιθυμητό πλήθος καταχωρήσεων φορολογικών εντύπων	89,47%
Ποσοστό επιτυχών εκκαθαρίσεων E1 (από τα καταχωρημένα έντυπα)	99,18%
Απόκλιση από το επιθυμητό πλήθος εκκαθαρίσεων	89,55%

Υπολογίζοντας μόνο τις αιτήσεις που τελικά στάλθηκαν από τα τερματικά (18777), αξίζει να δούμε και τα εξής ποσοστά:

Ποσοστό επιτυχών εγγεγραμμένων χρηστών στο σύστημα	99,88%
Απόκλιση από το επιθυμητό πλήθος καταχωρήσεων φορολογικών εντύπων	5,8%
Απόκλιση από το επιθυμητό πλήθος εκκαθαρίσεων	6,46%

6.3 Αποτελέσματα Σεναρίου 3

Συνολικός αριθμός σεναρίων προς εκτέλεση	275000
--	--------

Πλήθος τερματικών	2
Πλήθος σεναρίων εκτέλεσης ανά τερματικό	100000 (πρώτο τερματικό) 175000 (δεύτερο τερματικό)
Στάλθηκαν	Μη διαθέσιμο
Μέγιστος αριθμός ταυτόχρονων συνδέσεων στην βάση δεδομένων	2152
Αποτυχημένες συνδέσεις στην βάση δεδομένων	Μη διαθέσιμο
Εγγεγραμμένοι χρήστες	1665
Υποβολή εντύπων E1 ή και E2	879
Εκκαθάριση εντύπων E1	876

Ποσοστό επιτυχών εγγεγραμμένων χρηστών στο σύστημα	0,6%
Ποσοστό επιτυχών καταχωρήσεων φορολογικών εντύπων (από τους εγγεγραμμένους χρήστες)	52,79%
Απόκλιση από το επιθυμητό πλήθος καταχωρήσεων φορολογικών εντύπων	99,68%
Ποσοστό επιτυχών εκκαθαρίσεων E1 (από τα καταχωρημένα έντυπα)	99,65%
Απόκλιση από το επιθυμητό πλήθος εκκαθαρίσεων	99,68%

6.4 Προβλήματα

Κατά την διάρκεια εκτέλεσης των σεναρίων παρουσιάστηκαν τεχνικά προβλήματα που είχαν να κάνουν με τους διαθέσιμους πόρους που είχα στην διάθεση μου. Συγκεκριμένα και στα τρία σενάρια εκτέλεσης λόγω του μεγάλου αριθμού αιτήσεων και της καθυστέρησης επεξεργασίας των αιτήσεων από το υπολογιστικό νέφος είχε ως αποτέλεσμα έπειτα από κάποια χρονική διάρκεια όλα τα τερματικά (και ανάλογα με τα τεχνικά χαρακτηριστικά του) να παρουσίαζαν έλλειψη μνήμης (out of memory exception) . Επίσης στα πρώτα δυο σενάρια εκτέλεσης, (τα οποία εκτελέστηκαν από το εργαστήριο του πανεπιστημίου) παρατηρήθηκε ότι οι μικροί μεταγωγής (switch) που συνέδεαν του υπολογιστές είχαν κολλήσει λόγω του μεγάλου όγκου δεδομένων.

Ως αντίκτυπο ήταν να μη καταστεί δυνατό να εκτελεστούν όλα τα σενάρια από κάθε τερματικό σταθμό όπως αρχικά ήταν σχεδιασμένο. Αξίζει να αναφερθεί ότι κανένα τερματικό και από τα τρία σενάρια εκτέλεσης δε κατάφερε να ολοκληρώσει τον συνολικό αριθμό που έπρεπε να εκτελέσει. Επίσης στα δυο πρώτα σενάρια λόγω του προβλήματος του παρουσιάστηκε με τους μεταγωγής υπήρχαν υπολογιστές που δεν κατάφεραν να εκτελέσουν ούτε ένα σενάριο!

Εκτός από τους φυσικούς διαθέσιμους πόρους που είχα στην διάθεση μου είτε από το εργαστήριο του πανεπιστημίου είτε από τους προσωπικούς μου υπολογιστές, σημαντικός παράγοντας ήταν η διαθέσιμοι πόροι που μου διετεθήσαν στο υπολογιστικό νέφος!

Όπως ανέφερα στο κεφάλαιο 3.5, για την ανάπτυξη των υπηρεσιών χρησιμοποιήθηκε υπολογιστικό νέφος της εταιρείας Microsoft (Azure) με ακαδημαϊκή άδεια. Αυτό είχε ως αποτέλεσμα ένα σύνολο από περιορισμούς στους πόρους και τις υπηρεσίες που μπορούσα να χρησιμοποιήσω. Συγκεκριμένα, από το σύνολο των υπηρεσιών που παρείχε η Microsoft για σχεσιακή βάση δεδομένων είχα στην διάθεση μου δύο σχεσιακές βάσεις δεδομένων (από τις οποίες χρησιμοποίησα μόνο μια) με μέγιστη με μέγιστη χωρητικότητα 5 GB. Οι προδιαγραφές για τους διάσημους πόρους υλικού (μνήμη, κεντρική μονάδα επεξεργασία, κτλ.) που φιλοξενούνταν η βάση δεδομένων είναι άγνωστα. Από τα αποτελέσματα φάνηκε ότι δεν επαρκούσαν για την αυξημένη κίνηση (burst) που δημιουργήσα καθώς υπήρχαν πολλά "connection timeout" με την βάση δεδομένων. Από εικονικά μηχανήματα (virtual machines) που φιλοξενούσαν μόνο τον εξυπηρετητή εφαρμογών (web server) είχα δύο (χρησιμοποίησα και τα δύο), που διέθεταν 1.75 GB RAM, 35 GB αποθηκευτικό χώρο και μια κεντρική μονάδα επεξεργασίας (αγνώστων χαρακτηριστικών). Το πρόβλημα που εμφανίστηκε είχε να κάνει με το πλήθος των διαθέσιμων εικονικών μηχανών και τα περιορισμένα χαρακτηριστικά του υλικού τους (επίπεδο επιδόσεων, ενός μέτριου έξυπνου κινητού). Κατά την διάρκεια της εκτέλεσης των σεναρίων υπήρχε μέγιστη χρήση σε μνήμη και θα ήταν επιθυμητό να είχα στην διάθεσή μου τουλάχιστον άλλα δύο, ώστε να εκμεταλλευόμουν ένα από τα βασικά χαρακτηριστικά του υπολογιστικού νέφους την ελαστικότητα. Περιορισμός υπήρχε και στην κίνηση δεδομένων. Μπορούσα να μεταφέρω από και προς τις υποδομές έως 8 GB δεδομένων.

6.5 Ανάλυση

Τα παράπονα πρόβλημα συνιστούν σημαντικό παράγοντα στην μεγάλη απόκλιση (όπως εμφανίζεται στα αποτελέσματα των σεναρίων) από τον στόχο. Ο στόχος δεν ήταν άλλος από την συνολική καταχώρηση όλων των σεναρίων προς εκτέλεση. Χαρακτηριστικό είναι το παράδειγμα από το δεύτερο σενάριο όπου έχουμε διαθέσιμο το πλήθος των σεναρίων που δεν εκτελέστηκαν λόγω των προβλημάτων που αναφέρονται προηγουμένως.

Τα αποτελέσματα του σεναρίου 1, όπως φαίνονται και στον σχετικό πίνακα, είναι τα καλύτερα. Πέτυχα περισσότερες εγγραφές (72615) στο σύστημα και τελικά περισσότερες εκκαθαρίσεις φορολογικών εντύπων (72018) σε σχέση με τα άλλα δύο.

Στο σενάριο 2, λόγω του ότι είχαν προκύψει τα προβλήματα με τους διαθέσιμους πόρους των υπολογιστών του εργαστηρίου, υπάρχει διαθέσιμο το πλήθος των σεναρίων που στάλθηκε για εκτέλεση. Επιπλέον έγινε μια προσπάθεια για βελτίωση των αποτελεσμάτων με μια σειρά από ενέργειες σχετικά με της ρυθμίσεις σύνδεσης στην βάση δεδομένων όπως:

1. διπλασιασμός της δεξαμενής συνδέσεων (connection pool), από 1000 σε 2000
2. αύξηση του αριθμού επιπρόσθετων συνδέσεων στην βάση δεδομένων όταν η δεξαμενή συνδέσεων έχει φτάσει το μέγιστο όριο (2000) από 1 σε 10.
3. αφαίρεση χρόνου λήξης ανοίγματος σύνδεσης με την βάση δεδομένων. Στο σενάριο 1 υπήρχε μέγιστος χρόνος αναμονής 5 λεπτών.

Η τελευταία αλλαγή φαίνεται να επηρέασε πολύ αρνητικά στην εκτέλεση των σεναρίων διότι, όταν μια συναλλαγή απαιτούσε σύνδεση με την βάση δεδομένων, δεν αποκρυπτόταν εντός κάποιου χρονικού ορίου όταν δεν γινόταν εφικτό να αποκατασταθεί μια σύνδεση. Ως αποτέλεσμα να υπάρξει αυξημένος φόρτος στην βάση δεδομένων. Αυτό είχε ως αποτέλεσμα τα τερματικά να έχουν περισσότερα νήματα ενεργά διότι περίμεναν επ' αόριστο μέχρι να λάβουν απάντηση με τα αποτελέσματα της αίτησης που είχαν στείλει. Έτσι τα τερματικά έμεναν πιο γρήγορα από μνήμη και για αυτό εκτελέστηκαν πολύ λιγότερα σεναρία σε σύγκριση με τον πρώτο σεναριο εκτέλεσης.

Έτσι λοιπόν έχει ενδιαφέρον να δούμε τα αποτελέσματα από το πλήθος των σεναρίων που τελικά εκτελέστηκαν . Από τα 168000 σεναρία που ήταν ο αρχικώς στόχος, τελικά εκτελέστηκαν μόνο τα 18777 ! Τα υπόλοιπα δεν εκτελέστηκαν για τους λόγους που αναφέρονται στο κεφάλαιο 6.4 . Σε αυτό το σύνολο παρατηρείται υψηλό ποσοστό επιτυχίας της τάξεων του 99,18%. Αυτό είναι λογικό καθώς σε σύγκριση με το σεναριο 1 το σύστημα είχε να επεξεργαστεί λιγότερες αιτήσεις και συγκεκριμένα 53838 λιγότερες από το πρώτο σεναριο.

Στο σεναριο 3, έπειτα από την εμπειρία από το σεναριο 2 και σχετικά με τις ρυθμίσεις σύνδεσης με την βάση δεδομένων επανέφερα τον χρόνο ορίου λήξης (connection time out) στα 5 λεπτά. Στα αποτελέσματα του σεναρίου 3, λόγω του ότι το σεναριο εκτελέστηκε μόνο από δύο τερματικά, πρόλαβαν να σταλούν πολύ λίγες αιτήσεις σε σχέση με τα άλλα δύο σεναρία. Αυτό δικαιολογεί τον πολύ μικρό αριθμό εγγεγραμμένων χρηστών αλλά και καταχωρημένων και εκκαθαρισμένων φορολογικών εντύπων. Αξίζει να αναφερθεί ότι σε σύγκριση με τα άλλα δύο σεναρία εκτέλεσης, εμφανίζεται πολύ μικρό ποσοστό επιτυχών καταχωρήσεων φορολογικών εντύπων (από τους εγγεγραμμένους χρήστες). Στο συγκεκριμένο σεναριο είναι της τάξεως του 52,79% ενώ στα άλλα ξεπερνάει το 93,4% και ειδικά στο πρώτο έχουν καταχωρηθεί σχεδόν όλες (99,88%).

6.6 Συμπεράσματα

Τα πειραματικά αποτελέσματα έδειξαν ότι με τους περιορισμένους υπολογιστικούς πόρους που είχα στην διάθεση μου δεν κατέστη δυνατό να εξυπηρετηθεί το μέγιστο φόρτο για τα συγκεκριμένα σεναρία δοκιμών. Αυτό είναι λογικό καθώς συγκεκριμένοι υπολογιστικοί πόροι, είτε παρέχονται on – premises είτε μέσω υπολογιστικού νέφους έχουν τις ίδιες δυνατότητες. Η διαφορά που υπάρχει στο υπολογιστικό νέφος είναι δυνατότητα του «**elasticity**» (δυναμικά και κατά απαίτηση)

Τα προβλήματα που παρουσιάστηκαν στη ΓΠΣ όπως, μη διαθεσιμότητας πόρων λόγω συντήρησης , καταστροφή εξοπλισμού από φυσικά αίτια (πυρκαγιά) ή διακοπής ρεύματος μπορούν να αποφευχθούν με την υιοθέτηση του υπολογιστικού νέφους. Επιπλέον μέσω των SLA's προσφέρεται περισσότερη διασφάλιση και εγγύηση για την συνεχόμενη παροχή των υπηρεσιών.

Βιβλιογραφία

- Γιακουμάκης Μανόλης & Διαμαντίδης Νίκος. Τεχνολογία λογισμικού. Σταμούλη Α.Ε., 2009
- David S. Linthicum. Cloud Computing and SOA Convergence in Your Enterprise A Step-by-Step Guide. Pearson Education, 2009
- Judith Hurwitz, Marcia Kaufman, Fern Halper, Dan Kirsch. Hybrid Cloud For Dummies. John Wiley and Sons Ltd, 2012
- Peter Mell, Timothy Grance. The NIST Definition of Cloud Computing. Special Publication 800-145, 2011
- David Geary & Cay S. Horstmann. Core JavaServer Faces. Prentice Hall, 2010
- Christian Bauer & Gavin King. Java Persistence with Hibernate. Manning Publications, 2006
- Jeff Linwood. Beginning Hibernate 2nd Edition. APRESS, 2010
- Anghel Leonard. JSF 2.0 Cookbook. PACKT PUBLISHING, 2010
- Deepak Vohra. Java 7 JAX-WS Web Services. PACKT PUBLISHING, 2012
- Deepak Vohra. Java EE Development with Eclipse. PACKT PUBLISHING, 2012
- Masoud Kalali. Developing RESTful Services with JAX-RS 2.0, WebSockets, and JSON. PACKT PUBLISHING, 2013
- Javier Fernández González. Java 7 Concurrency Cookbook. Packt Publishing, 2012
- Kirk Knoernschild. Java Application Architecture: Modularity Patterns with Examples Using OSGi: A Roadmap for Enterprise Development (Agile Software Development). Prentice Hall, 2012
- Binildas Christudas, Malhar Barai & Vincenzo Caselli. Service Oriented Architecture with Java. PACKT PUBLISHING, 2008
- Mick Knutson. Java EE 6 Cookbook for Securing, Tuning, and Extending Enterprise Applications. PACKT PUBLISHING, 2012
- Sunil Gulabani. Developing RESTful Web Services with Jersey 2.0. PACKT PUBLISHING, 2013
- Naveen Balani & Rajeev Hathi. Apache CXF Web Service Development. PACKT PUBLISHING, 2009
- Hurwitz Judith, Bloor Robin, Kaufman Marcia, Halper Fern, Cloud Computing For Dummies Wiley and Sons Ltd, 2010
- Judith Hurwitz, Marcia Kaufman. Private Cloud For Dummies. John Wiley and Sons Ltd, 2011
- Roger Jennings. Cloud Computing with the Windows Azure Platform. Wrox, 2009
- Jothy Rosenberg & Arthur Mateos. The Cloud at Your Service. Manning Publications, 2010
- Lee Chao. Cloud Database Development and Management. Auerbach Publications, 2013
- Sriram Krishnan. Programming Windows Azure: Programming the Microsoft Cloud. O'Reilly Media, 2010
- Chris Hay & Brian Prince. Azure in Action. Manning Publications, 2010
- Tejaswi Redkar. Windows Azure Platform. APRESS, 2010
- Roberto Brunetti. Windows Azure Step By Step. MICROSOFT PRESS, 2011
- Scott Klein. Pro SQL Database for Windows Azure: SQL Server in the Cloud 2nd Edition. APRESS, 2012
- Dan Marinescu. Cloud Computing: Theory and Practice. Morgan Kaufmann, 2013
- Aidan Finn, Hans Vredevort, Patrick_Lownds & Debian Flynn. Microsoft Private Cloud Computing. John Wiley & Sons, 2012

Kees Blokland & Martin Pol. Testing Cloud Services: How to Test SaaS, PaaS & IaaS. Rocky Nook, 2013

Garima Gulati. Multi-tenant Architecture In A Private Cloud: Application-level Virtualization In SAAS. LAP LAMBERT Academic Publishing, 2012

John Rhoton. Cloud Computing Explained: Implementation Handbook for Enterprises. Recursive Press, 2010

Danny Garber, Jamal Malik & Adam Fazio. Windows Azure Hybrid Cloud. Wrox, 2013

Douglas K. Barry. Web Services, Service-Oriented Architectures, and Cloud Computing: The Savvy Manager's Guide. Morgan Kaufmann, 2012

Neil Mackenzie. Microsoft Windows Azure Development Cookbook. PACKT PUBLISHING, 2011

Bill Wilder. Cloud Architecture Patterns: Using Microsoft Azure. O'Reilly, 2012

Kalman Toth. Windows Azure SQL Database Programming & Design. CreateSpace Independent Publishing Platform, 2013

Scott Klein. Pro SQL Azure. APRESS, 2010

Πηγές

http://en.wikipedia.org/wiki/Cloud_computing

<http://searchstorage.techtarget.com/definition/Storage-as-a-Service-SaaS>

<http://searchsecurity.techtarget.com/definition/Security-as-a-Service>

http://www.gsis.gr/on_line_ypiresies/polites/e-entipa_files/entypa/e-docs.html

<http://www.forin.gr/articles/article/5954/ta-tekmhria-dapanwn-diabiwshs-gia-to-oikonomiko-etos-2012>

<http://www.taxheaven.gr/pages/tekmiria2012>

<http://www.youtube.com/watch?v=x00KjQZ9XvA>

<http://www.youtube.com/watch?v=4WYMI7EGlek>

<http://blogs.sap.com/innovation/cloud-computing/top-9-challenges-in-cloud-computing-that-are-slowing-its-adoption-011918>

<http://www.cloudtweaks.com/2012/08/top-five-challenges-of-cloud-computing/>

<http://www.cloudtweaks.com/2012/08/top-five-challenges-of-cloud-computing/>

Οδηγίες για τη συμπλήρωση της δήλωσης εισοδήματος φυσικών προσώπων οικονομικού έτους 2012. Υπουργείο οικονομικών

Παράρτημα

Για τις ανάγκες της εργασίας δημιούργησα συνολικά τέσσερις εφαρμογές μία βιβλιοθήκη και μια βάση δεδομένων.

Όλα τα εκτελέσιμα και ο κώδικας επισυνάπτονται στο συνοδευτικό CD.

Όνομα	Αρχείο που περιέχει τον κώδικα	Όνομα εκτελέσιμου αρχείου
TaxisNetUI	TaxisNetUI.zip ή https://github.com/manousos/TaxisNetUI.git	TaxisNetUI.war
TaxisNet	TaxisNet.zip ή https://github.com/manousos/TaxisNet.git	TaxisNet.war
manousos-stresstest	manousos-stresstest.zip	manousos-stresstest.exe
TaxisTest	TaxisStressTest.zip	TaxisTest.jar ²⁴
TaxisNetModel	TaxisNetModel.zip ή https://github.com/manousos/TaxisNetModel.git	TaxisModel.jar
TaxisDB	taxisDB-mssql-azure-schema-only.sql	

A. Οδηγίες

Για τον τρόπο εκτέλεσης των εφαρμογών manousos-stresstest και TaxisTest δείτε στο κεφάλαιο 4.1. Στο μέρος Γ του παραρτήματος παρατίθεται ο κώδικας της εφαρμογής TaxisTest. Εκεί μπορείτε να δείτε και τον τρόπο χρήσης - κλήσεις των υπηρεσιών Rest (της εφαρμογής TaxisNet) από οποιαδήποτε άλλη εφαρμογή.

Για την δημιουργία της βάσης δεδομένων TaxisDB, δημιουργούμε στον MS SQL Server μια βάση δεδομένων με όνομα TaxisDB και έπειτα εκτελούμε το αρχείο taxisDB-mssql-azure-schema-only.sql.

²⁴Εντός του συμπιεσμένου αρχείου manousos-stresstest.zip, φάκελος Debug

Για την εγκατάσταση των εφαρμογών TaxisNet και TaxisNetUI το μόνο που χρειάζεται είναι να κάνουμε deploy τα αντίστοιχα war αρχεία. Κάθε application server έχει τον δικό του τρόπο για την συγκεκριμένη διαδικασία. Παραδείγματος χάρη στον glassfish είτε μπορούμε να το πραγματοποιήσουμε μέσω web interface είτε μέσω γραμμής εντολών²⁵.

B. Κώδικας προγράμματος manousos-stresstest

```
module manousosstresstest.Main
// Created by George Dernikos
open System
open System.Net
open System.Diagnostics
open Microsoft.FSharp.Control.WebExtensions

let runAsyncCmd(cmd, fromAfm : int, toAfm : int) =
    let afms = [|fromAfm .. toAfm|]
    let runCmd afm =
        async {
            try
                let proc = Process.Start(cmd, afm)
                proc.WaitForExit()
            with
                | ex -> printfn "%s" (ex.Message);
        }
    afms
    |> Array.map string
    |> Array.map runCmd
    |> Async.Parallel
    |> Async.RunSynchronously
    |> ignore

[<EntryPoint>]
let main args =
    let (cmd, fromAfm, toAfm) =
        match args with
        | [| cmd; fromAfm; toAfm |] ->
            (cmd, int fromAfm, int toAfm)
        | _ -> failwith
            (sprintf "invalid args. expected: %A"
                ("cmd", "fromAFM", "toAFM"))
    runAsyncCmd(cmd, fromAfm, toAfm) |> ignore
    0
```

²⁵<http://docs.oracle.com/cd/E19798-01/821-1757/6nmni99ao/index.html>

Γ. Κώδικας προγράμματος TaxisTest

```
package TaxisTest;

import gr.manousos.model.Contact;
import gr.manousos.model.E1;
import gr.manousos.model.E1Id;
import gr.manousos.model.E1expensesRemovedFromTotalIncome;
import gr.manousos.model.E1infoData;
import gr.manousos.model.E1objectiveSpending;
import gr.manousos.model.E1personDataBorneTaxpayer;
import gr.manousos.model.E1prepaidTaxes;
import gr.manousos.model.E1taxPayerBankAccount;
import gr.manousos.model.E1taxableIncomes;
import gr.manousos.model.E2;
import gr.manousos.model.E2Id;
import gr.manousos.model.E2coOwner;
import gr.manousos.model.E2estate;
import gr.manousos.model.E2otherEstate;
import gr.manousos.model.IncomeTax;
import gr.manousos.model.RelatePerson;
import gr.manousos.model.Taxpayer;
import gr.manousos.model.User;

import java.util.Calendar;
import java.util.HashSet;
import java.util.Set;

import javax.ws.rs.core.MediaType;

import com.sun.jersey.api.client.Client;
import com.sun.jersey.api.client.ClientResponse;
```

```

import com.sun.jersey.api.client.WebResource;
import com.sun.jersey.api.client.config.ClientConfig;
import com.sun.jersey.api.client.config.DefaultClientConfig;
import com.sun.jersey.api.representation.Form;

public class Main {

    private static final int testCases = 5;
    private static final String url = "http://taxisnet.cloudapp.net:8098/TaxisNet/rest/";

    private int afm;
    private String userName = "";

    public Main(String afm) {
        this.afm = Integer.parseInt(afm);
    }

    public static void main(String[] args) {
        new Main(args[0]).run();
        System.out.println("Test End");
    }

    public void run() {
        System.out.println("Execute for afm: " + this.afm);

        ClientConfig conf = new DefaultClientConfig();
        Client client = Client.create(conf);

        User u = buildUser();
        Contact con = buildContact();
        Taxpayer taxPayer = buildTaxpayer(u, con);
    }
}

```

```

try {
    Taxpayer newUser = null;
    WebResource r = client.resource(url);

    newUser = r.path("UserService/Register").accept("application/json")
        .type("application/json").post(Taxpayer.class, taxPayer);
    if (newUser == null || newUser.getId() == null)
        // throw new Exception("Register for user with AFM: "
        // + String.valueOf(this.afm) + " failed");
        System.out.println("Register for user with AFM: "
            + String.valueOf(this.afm) + " failed");
    // org.junit.Assert.fail("Register for user with AFM: "
    // + String.valueOf(fromAFM) + " failed");
    else {
        this.userName = u.getUserName();
        System.out.println("Username " + this.userName);

        if (login(client)) {

            switch (new java.util.Random().nextInt(testCases)) {
            case 1:
                if (!submitE1(client, buildE1Simple2(newUser.getId())))
                    // throw new Exception(
                    // "Fail to submit E1[Simple2] for taxpayer with username "
                    // + this.userName);
                    System.err
                        .println("Fail to submit E1[Simple2] for taxpayer with username "
                            + this.userName);
                    // fail("Fail to submit E1[Simple2] for taxpayer with username "
                    // + this.userName);
                else {
                    if (!calculateTax(client, newUser.getId(), Calendar

```

```

        .getInstance().get(Calendar.YEAR)))
        // throw new Exception(
        // "Fail to calculate tax from taxpayer with ID "
        // + newUser.getId());
        System.err
            .println("Fail to calculate tax from taxpayer with ID "
                + newUser.getId());
    }
    break;
case 2:
    if (!submitE1(client,
        buildE1WithWifeAnd2Kids(newUser.getId())))
        // throw new Exception(
        // "Fail to submit E1[WithWifeAnd2Kids] for taxpayer with username "
        // + this.userName);
        System.err
            .println("Fail to submit E1[WithWifeAnd2Kids] for taxpayer with
username "
                + this.userName);
    else {
        if (!calculateTax(client, newUser.getId(), Calendar
            .getInstance().get(Calendar.YEAR)))
            // throw new Exception(
            // "Fail to calculate tax from taxpayer with ID "
            // + newUser.getId());
            System.err
                .println("Fail to calculate tax from taxpayer with ID "
                    + newUser.getId());
    }
    break;
case 3:
    if (!submitE1(client,

```

```

        buildE1WithWifeAnd3Kids(newUser.getId()))
// throw new Exception(
// "Fail to submit E1[WithWifeAnd3Kids] for taxpayer with username "
// + this.userName);
System.err
        .println("Fail to submit E1[WithWifeAnd3Kids] for taxpayer with
username "
                + this.userName);
else {
    if (!calculateTax(client, newUser.getId(), Calendar
        .getInstance().get(Calendar.YEAR)))
        // throw new Exception(
        // "Fail to calculate tax from taxpayer with ID "
        // + newUser.getId());
        System.err
            .println("Fail to calculate tax from taxpayer with ID "
                    + newUser.getId());
    }
    break;
case 4:
    if (!submitE1(client,
        buildE1WithWifeAndHouses(newUser.getId()))
        // throw new Exception(
        // "Fail to submit E1[WithWifeAndHouses] for taxpayer with username "
        // + this.userName);
        System.err
            .println("Fail to submit E1[WithWifeAndHouses] for taxpayer
with username "
                    + this.userName);
    else {
        if (!calculateTax(client, newUser.getId(), Calendar
            .getInstance().get(Calendar.YEAR)))
            // throw new Exception(

```

```

        // "Fail to calculate tax from taxpayer with ID "
        // + newUser.getId());
        System.err
            .println("Fail to calculate tax from taxpayer with ID "
                + newUser.getId());

        if (!submitE2(client,
            buildE2WithWifeAndHouses(newUser.getId())))
            System.err
                .println("Fail to submit E2 for taxpayer with username "
                    + this.userName);
    }

    break;
default:
    if (!submitE1(client, buildE1Simple(newUser.getId())))
        System.err
            .println("Fail to submit E1[Simple] for taxpayer with username "
                + this.userName);
    else {
        if (!calculateTax(client, newUser.getId(), Calendar
            .getInstance().get(Calendar.YEAR)))
            System.err
                .println("Fail to calculate tax from taxpayer with ID "
                    + newUser.getId());
    }

    break;
}
} else
    System.err
        .println("Fail to logind for taxpayer with username "

```

```

        + this.userName);
    }
} catch (Exception ex) {
    System.err.println("Exception on afm " + this.afm + " "
        + ex.toString());
}
}

private boolean login(Client client) {
    WebResource r = client.resource(url);

    Form f = new Form();
    f.add("username", this.userName);
    f.add("password", "123456");

    ClientResponse response = r.path("UserService/Login")
        .type(MediaType.APPLICATION_FORM_URLENCODED)
        .post(ClientResponse.class, f);

    if (response.getStatus() == 200)
        if (response.getEntity(String.class).equals("1")) {
            return true;
        }

    return false;
}

private boolean submitE1(Client client, E1 e1) {
    WebResource r = client.resource(url);

```

```

String result = "";
result = r.path("DocumentService/submitE1").accept("application/json")
        .type("application/json").post(String.class, e1);

// this.error = result;
if (result.equals("E1 Saved !!"))
    return true;
else {
    // org.junit.Assert.fail(result);
    return false;
}
}

private boolean submitE2(Client client, E2 e2) {
    WebResource r = client.resource(url);

    String result = "";
    result = r.path("DocumentService/submitE2").accept("application/json")
            .type("application/json").post(String.class, e2);

    // this.error = result;
    if (result.equals("E2 Saved !!"))
        return true;
    else {
        // org.junit.Assert.fail(result);
        return false;
    }
}

private boolean calculateTax(Client client, int taxpayerID, int year) {
    IncomeTax tax = null;

```



```

WebResource restSrv2 = client.resource(url);

tax = (IncomeTax) restSrv2.path("TaxCalcService/tax/")
    .queryParams("tId", String.valueOf(taxpayerID))
    .queryParams("year", String.valueOf(year))
    .accept(MediaType.APPLICATION_JSON).get(IncomeTax.class);

return (tax != null) ? true : false;
}

private User buildUser() {
    User u = new User();
    u.setUsername("user" + this.zeroPadLeft(this.afm));
    u.setPassword("123456");
    u.setActive(1);
    u.setOtp("987654");

    return u;
}

private Contact buildContact() {
    Contact con = new Contact();
    con.setPhone("+302101234567");
    con.setFax("");
    con.setCell("+306956234797");
    con.setEmail("manouso1@freemail.gr");
    return con;
}

private Taxpayer buildTaxpayer(User u, Contact con) {
    Taxpayer taxPayer = new Taxpayer();
    taxPayer.setAfm(this.zeroPadLeft(this.afm));
}

```

```

taxPayer.setContact(con);
taxPayer.setFatherName("Αλέκος");
taxPayer.setFname("Μάριος");
taxPayer.setLname("Παπαδόπουλος");
taxPayer.setUser(u);

return taxPayer;
}

private E1 buildE1WithWifeAnd2Kids(int taxPayerID) {
    Set<RelatePerson> relatePersons = new HashSet<RelatePerson>();

    Contact wifeContact = new Contact("2111231233", "6956565656", "");
    Contact delegateContact = new Contact("2101231236", "", "");

    RelatePerson wife = new RelatePerson();
    wife.setAddress("Καπου 32");
    wife.setAfm("908765461");
    wife.setAtid("Γ123456");
    wife.setContact(wifeContact);
    wife.setFatherName("Ευάγγελος");
    wife.setFname("Ασπασία");
    wife.setLname("Μαρκουλάκη");
    wife.setType(1);

    RelatePerson delegate = new RelatePerson();
    delegate.setAddress("Αλαμάνας 22");
    delegate.setAfm("976546548");
    delegate.setFatherName("ΑΝΕΣΤΗΣ");
    delegate.setFname("ΝΙΚΟΛΑΟΣ");
    delegate.setLname("ΝΙΚΟΛΑΟΥ");
    delegate.setContact(delegateContact);
}

```

```
delegate.setType(2);
```

```
relatePersons.add(wife);
```

```
relatePersons.add(delegate);
```

```
E1Id key = new E1Id(taxPayerID, Calendar.getInstance().get(
    Calendar.YEAR));
```

```
E1expensesRemovedFromTotalIncome expensesRemovedFromTotalIncomeObj = new
E1expensesRemovedFromTotalIncome();
```

```
expensesRemovedFromTotalIncomeObj.set_049(10000f);
```

```
expensesRemovedFromTotalIncomeObj
```

```
    .setRentalForFamilyOwnerName1("ΜΑΡΚΑΚΗΣ ΙΩΑΝΝΗΣ");
```

```
expensesRemovedFromTotalIncomeObj.set_801("095874598");
```

```
expensesRemovedFromTotalIncomeObj.set_092(1);
```

```
expensesRemovedFromTotalIncomeObj.set_091(125f);
```

```
expensesRemovedFromTotalIncomeObj.set_097(12);
```

```
expensesRemovedFromTotalIncomeObj.set_811(4.000f);
```

```
expensesRemovedFromTotalIncomeObj.set_073(1000f);
```

```
expensesRemovedFromTotalIncomeObj.set_084(500f);
```

```
expensesRemovedFromTotalIncomeObj.setId(key);
```

```
E1personDataBorneTaxpayer personDataBorneTaxpayerObj = new
E1personDataBorneTaxpayer();
```

```
personDataBorneTaxpayerObj.setBirthYear1(1999);
```

```
personDataBorneTaxpayerObj.setName1("ΜΑΡΚΑΚΗΣ ΝΙΚΟΛΑΟΣ");
```

```
personDataBorneTaxpayerObj.set_831("012356789");
```

```
personDataBorneTaxpayerObj.setSchoolName1("3ο ΓΥΜΝΑΣΙΟ ΑΘΗΝΩΝ");
```

```
personDataBorneTaxpayerObj.setBirthYear2(2001);
```

```
personDataBorneTaxpayerObj.setName2("ΜΑΡΚΑΚΗ ΝΕΦΕΛΗ");
```

```
personDataBorneTaxpayerObj.set_832("012356289");
```

```
personDataBorneTaxpayerObj.setSchoolName1("3ο ΓΥΜΝΑΣΙΟ ΑΘΗΝΩΝ");
```

```
personDataBorneTaxpayerObj.setId(key);
```

```
E1prepaidTaxes prepaidTaxesObj = new E1prepaidTaxes();
prepaidTaxesObj.set_313(214.27f);
prepaidTaxesObj.set_315(211.06f);
prepaidTaxesObj.setId(key);
```

```
E1taxableIncomes taxableIncomeObj = new E1taxableIncomes();
taxableIncomeObj.set_301(35000f);
taxableIncomeObj.setId(key);
```

```
E1taxPayerBankAccount taxPayerBankAccountObj = new E1taxPayerBankAccount();
taxPayerBankAccountObj.setBic("012");
taxPayerBankAccountObj.setIban("GR1001409080988002340020081");
taxPayerBankAccountObj.setId(key);
```

```
E1 e1 = new E1();
e1.setId(key);
e1.setIsComplete(1);
e1.setAtid("I150080");
e1.setDoy("Α' Πειραιά");
e1.setMarriage(1);
e1.setTaxpayerAddress("Καπου 32");
e1.setDateInserted(new java.util.Date());
// e1.setE1dataFromTaxPayerFolder(dataFromTaxPayerFolderObj);
e1.setE1expensesRemovedFromTotalIncome(expensesRemovedFromTotalIncomeObj);
// e1.setE1incomesReduceTaxes(incomesReduceTaxesObj);
// e1.setE1infoData(infoDataObj);
// e1.setE1objectiveSpending(objectiveSpendingObj);
e1.setE1personDataBorneTaxpayer(personDataBorneTaxpayerObj);
e1.setE1prepaidTaxes(prepaidTaxesObj);
e1.setE1personDataBorneTaxpayer(personDataBorneTaxpayerObj);
// e1.setE1reduceTax(reduceTaxObj);
```

```

    e1.setE1taxableIncomes(taxableIncomeObj);
    e1.setE1taxPayerBankAccount(taxPayerBankAccountObj);
    // e1.setE1nauticalincomes(nautical);
    e1.setRelatePersons(relatedPersons); // Many to Many
    // e1.setTaxpayer(taxpayer);

    return e1;
}

private E1 buildE1WithWifeAnd3Kids(int taxpayerID) {
    // *Εγγαμος με τρία παιδιά και ετήσιο εισόδημα 32.000 ευρώ.
    // Πληρώνει για τόκους στεγαστικού δανείου πρώτης κατοικίας 4.000 ευρώ
    // ετησίως, νοσήλια 3.000 ευρώ, δίδακτρα για το ένα παιδί 700 ευρώ και
    // έχει συγκεντρώσει αποδείξεις αξίας 15.000 ευρώ.*

    Set<RelatePerson> relatedPersons = new HashSet<RelatePerson>();

    Contact wifeContact = new Contact("210978123", "6975515606", "");
    Contact delegateContact = new Contact("2101231236", "", "");

    RelatePerson wife = new RelatePerson();
    wife.setAddress("ΚΥΜΗΣ 30");
    wife.setAfm("087157621");
    wife.setAtid("Γ875198");
    wife.setContact(wifeContact);
    wife.setFatherName("Ιωάννης");
    wife.setFname("ΕΛΕΝΗ");
    wife.setLname("ΚΑΡΑΦΩΤΗ");
    wife.setType(1);

    RelatePerson delegate = new RelatePerson();
    delegate.setAddress("Αλαμάνας 22");

```

```
delegate.setAfm("976546548");
delegate.setFatherName("ΑΝΕΣΤΗΣ");
delegate.setFname("ΝΙΚΟΛΑΟΣ");
delegate.setLname("ΝΙΚΟΛΑΟΥ");
delegate.setContact(delegateContact);
delegate.setType(2);
```

```
relatePersons.add(wife);
relatePersons.add(delegate);
```

```
E1Id key = new E1Id(taxPayerID, Calendar.getInstance().get(
    Calendar.YEAR));
```

```
E1objectiveSpending objectiveSpendingObj = new E1objectiveSpending();
objectiveSpendingObj
    .setHouseAddr1("ΚΥΜΗΣ 30 - Τυχης - Μήλου - Αμοργού");
objectiveSpendingObj.setFloor1("3");
objectiveSpendingObj.set_205("1234432");
objectiveSpendingObj.set_211(120f);
objectiveSpendingObj.set_212(10f);
objectiveSpendingObj.set_213(100f);
objectiveSpendingObj.set_215(12);
objectiveSpendingObj.set_216(1500f);
objectiveSpendingObj.setId(key);
```

```
E1personDataBorneTaxpayer personDataBorneTaxpayerObj = new
E1personDataBorneTaxpayer();
personDataBorneTaxpayerObj.setBirthYear1(1995);
personDataBorneTaxpayerObj.setName1("ΠΑΠΑΔΟΠΟΥΛΟΣ ΝΙΚΟΛΑΟΣ");
personDataBorneTaxpayerObj.set_831("012356781");
personDataBorneTaxpayerObj.setSchoolName1("3ο ΛΥΚΕΙΟ ΑΘΗΝΩΝ");
personDataBorneTaxpayerObj.setBirthYear2(2002);
```

```
personDataBorneTaxpayerObj.setName2("ΠΑΠΑΔΟΠΟΥΛΟΥ ΑΡΕΤΗ");
personDataBorneTaxpayerObj.setSchoolName1("3ο ΓΥΜΝΑΣΙΟ ΑΘΗΝΩΝ");
personDataBorneTaxpayerObj.setBirthYear2(2006);
personDataBorneTaxpayerObj.setName2("ΠΑΠΑΔΟΠΟΥΛΟΥ ΕΥΑΝΘΙΑ");
personDataBorneTaxpayerObj.setSchoolName1("3ο ΔΗΜΟΤΙΚΟ ΑΘΗΝΩΝ");
personDataBorneTaxpayerObj.setId(key);
```

```
E1taxableIncomes income = new E1taxableIncomes();
income.set_301(32000f);
income.setId(key);
```

```
E1expensesRemovedFromTotalIncome removed = new
E1expensesRemovedFromTotalIncome();
removed.set_049(15000f);
removed.set_051(3000f);
removed.set_071(130f);
removed.set_063(4000f);
removed.set_081(700f);
removed.setId(key);
```

```
E1taxPayerBankAccount taxPayerBankAccountObj = new E1taxPayerBankAccount();
taxPayerBankAccountObj.setBic("012");
taxPayerBankAccountObj.setIban("GR1731409080988002340597108");
taxPayerBankAccountObj.setId(key);
```

```
E1 e1 = new E1();
e1.setId(key);
e1.setIsComplete(1);
e1.setAtid("A998088");
e1.setDoy("B' Πειραιά");
e1.setMarriage(1);
e1.setTaxpayerAddress("ΚΥΜΗΣ 30");
```

```

e1.setDateInserted(new java.util.Date());
// e1.setE1dataFromTaxPayerFolder(dataFromTaxPayerFolderObj);
e1.setE1expensesRemovedFromTotalIncome(removed);
// e1.setE1incomesReduceTaxes(incomesReduceTaxesObj);
// e1.setE1infoData(infoDataObj);
e1.setE1objectiveSpending(objectiveSpendingObj);
e1.setE1personDataBorneTaxpayer(personDataBorneTaxpayerObj);
// e1.setE1prepaidTaxes(prepaidTaxesObj);
e1.setE1personDataBorneTaxpayer(personDataBorneTaxpayerObj);
// e1.setE1reduceTax(reduceTaxObj);
e1.setE1taxableIncomes(income);
e1.setE1taxPayerBankAccount(taxPayerBankAccountObj);
// e1.setE1nauticalincomes(nautical);
e1.setRelatePersons(relatedPersons); // Many to Many

return e1;
}

```

```

private E1 buildE1Simple(int taxPayerID) {
    Set<RelatePerson> relatedPersons = new HashSet<RelatePerson>();

    Contact delegateContact = new Contact("2101231236", "", "");

    RelatePerson delegate = new RelatePerson();
    delegate.setAddress("Αλαμάνας 22");
    delegate.setAfm("976546548");
    delegate.setFatherName("Ανέστης");
    delegate.setFname("Νικόλαος");
    delegate.setLname("Νικολάου");
    delegate.setContact(delegateContact);
    delegate.setType(2);
}

```



```

// relatePersons.add(wife);
relatePersons.add(delegate);

E1Id key = new E1Id(taxPayerID, Calendar.getInstance().get(
    Calendar.YEAR));

E1expensesRemovedFromTotalIncome expensesRemovedFromTotalIncomeObj = new
E1expensesRemovedFromTotalIncome();
expensesRemovedFromTotalIncomeObj.set_049(3290f);
expensesRemovedFromTotalIncomeObj.set_051(50f);
expensesRemovedFromTotalIncomeObj
    .setRentalForFamilyOwnerName1("ΜΑΘΙΟΥΔΑΚΗΣ ΙΩΑΝΝΗΣ");
expensesRemovedFromTotalIncomeObj.set_801("095874598");
expensesRemovedFromTotalIncomeObj.set_092(1);
expensesRemovedFromTotalIncomeObj.set_091(75f);
expensesRemovedFromTotalIncomeObj.set_097(12);
expensesRemovedFromTotalIncomeObj.setId(key);

E1infoData infoDataObj = new E1infoData();
infoDataObj.set_017(1);
infoDataObj.setId(key);

E1taxableIncomes taxableIncomeObj = new E1taxableIncomes();
taxableIncomeObj.set_301(13000f);
taxableIncomeObj.setId(key);

E1objectiveSpending objectiveSpendingObj = new E1objectiveSpending();
objectiveSpendingObj
    .setHouseAddr1("ΜΑΝΗΣ 61 - ΛΑΚΩΝΙΑΣ - ΓΡΑΒΙΑΣ - ΙΩΛΚΟΥ");
objectiveSpendingObj.setFloor1("2");
objectiveSpendingObj.set_205("195237");
objectiveSpendingObj.set_211(60f);

```

```
objectiveSpendingObj.set_212(5f);
objectiveSpendingObj.set_213(100f);
objectiveSpendingObj.set_215(12);
objectiveSpendingObj.set_216(1900f);
objectiveSpendingObj.setld(key);
```

```
E1prepaidTaxes prepaidTaxesObj = new E1prepaidTaxes();
prepaidTaxesObj.set_313(214.27f);
prepaidTaxesObj.set_315(211.06f);
prepaidTaxesObj.setld(key);
```

```
E1taxPayerBankAccount taxPayerBankAccountObj = new E1taxPayerBankAccount();
taxPayerBankAccountObj.setBic("011");
taxPayerBankAccountObj.setIban("GR7201409880988002340000622");
taxPayerBankAccountObj.setld(key);
```

```
E1 e1 = new E1();
e1.setld(key);
e1.setIsComplete(1);
e1.setAtid("X900157");
e1.setDoy("Μήλου");
e1.setMarriage(0);
e1.setTaxpayerAddress("ΣΥΓΓΡΟΥ 32");
e1.setDateInserted(new java.util.Date());
// e1.setE1dataFromTaxPayerFolder(dataFromTaxPayerFolderObj);
e1.setE1expensesRemovedFromTotalIncome(expensesRemovedFromTotalIncomeObj);
// e1.setE1incomesReduceTaxes(incomesReduceTaxesObj);
e1.setE1infoData(infoDataObj);
// e1.setE1objectiveSpending(objectiveSpendingObj);
// e1.setE1personDataBorneTaxpayer(personDataBorneTaxpayerObj);
e1.setE1prepaidTaxes(prepaidTaxesObj);
// e1.setE1personDataBorneTaxpayer(personDataBorneTaxpayerObj);
```

```

// e1.setE1reduceTax(reduceTaxObj);
e1.setE1taxableIncomes(taxableIncomeObj);
e1.setE1taxPayerBankAccount(taxPayerBankAccountObj);
// e1.setE1nauticalincomes(nautical);
e1.setRelatePersons(relatedPersons); // Many to Many

return e1;

}

private E1 buildE1Simple2(int taxpayerID) {
    Set<RelatePerson> relatedPersons = new HashSet<RelatePerson>();
    Contact delegateContact = new Contact("2101231236", "", "");

    RelatePerson delegate = new RelatePerson();
    delegate.setAddress("Αλαμάνας 22");
    delegate.setAfm("976546548");
    delegate.setFatherName("Ανέστης");
    delegate.setFname("Νικόλαος");
    delegate.setLname("Νικολάου");
    delegate.setContact(delegateContact);
    delegate.setType(2);

    // relatedPersons.add(wife);
    relatedPersons.add(delegate);

    E1Id key = new E1Id(taxpayerID, Calendar.getInstance().get(
        Calendar.YEAR));

    // E1dataFromTaxPayerFolder dataFromTaxPayerFolderObj = new
    // E1dataFromTaxPayerFolder();
    // dataFromTaxPayerFolderObj.set_341(341.40f);

```

```

E1expensesRemovedFromTotalIncome expensesRemovedFromTotalIncomeObj = new
E1expensesRemovedFromTotalIncome();
expensesRemovedFromTotalIncomeObj.set_049(4000f);
expensesRemovedFromTotalIncomeObj
    .setRentalForFamilyOwnerName1("ΜΑΛΑΜΑΣ ΙΩΑΝΝΗΣ");
expensesRemovedFromTotalIncomeObj.set_801("095834598");
expensesRemovedFromTotalIncomeObj.set_811(3500f);
expensesRemovedFromTotalIncomeObj.set_092(0);
expensesRemovedFromTotalIncomeObj.set_091(50f);
expensesRemovedFromTotalIncomeObj.set_097(12);
expensesRemovedFromTotalIncomeObj.setId(key);

E1infoData infoDataObj = new E1infoData();
infoDataObj.set_017(1);
infoDataObj.setId(key);

E1taxableIncomes taxableIncomeObj = new E1taxableIncomes();
taxableIncomeObj.set_301(13000f);
taxableIncomeObj.setId(key);

E1objectiveSpending objectiveSpendingObj = new E1objectiveSpending();
objectiveSpendingObj
    .setHouseAddr1("ΠΗΛΙΟΥ 8 - ΖΑΓΟΡΑΣ - ΠΕΡΕΒΟΥ - ΚΑΡΤΑΛΗ");
objectiveSpendingObj.setFloor1("3");
objectiveSpendingObj.set_203(1);
objectiveSpendingObj.set_205("938501");
objectiveSpendingObj.set_211(50f);
objectiveSpendingObj.set_212(5f);
objectiveSpendingObj.set_213(100f);
objectiveSpendingObj.set_215(12);
objectiveSpendingObj.set_216(1900f);
objectiveSpendingObj.setId(key);

```

```
E1prepaidTaxes prepaidTaxesObj = new E1prepaidTaxes();
prepaidTaxesObj.set_313(214.27f);
prepaidTaxesObj.set_315(211.06f);
prepaidTaxesObj.setId(key);
```

```
E1taxPayerBankAccount taxPayerBankAccountObj = new E1taxPayerBankAccount();
taxPayerBankAccountObj.setBic("032");
taxPayerBankAccountObj.setIban("GR0201609883988002040030020");
taxPayerBankAccountObj.setId(key);
```

```
E1 e1 = new E1();
e1.setId(key);
e1.setIsComplete(1);
e1.setAtid("Φ400157");
e1.setDoy("ΑΡΑΧΩΒΑΣ");
e1.setMarriage(0);
e1.setTaxpayerAddress("ΙΟΥ 70");
e1.setDateInserted(new java.util.Date());
// e1.setE1dataFromTaxPayerFolder(dataFromTaxPayerFolderObj);
e1.setE1expensesRemovedFromTotalIncome(expensesRemovedFromTotalIncomeObj);
// e1.setE1incomesReduceTaxes(incomesReduceTaxesObj);
e1.setE1infoData(infoDataObj);
e1.setE1objectiveSpending(objectiveSpendingObj);
// e1.setE1personDataBorneTaxpayer(personDataBorneTaxpayerObj);
e1.setE1prepaidTaxes(prepaidTaxesObj);
// e1.setE1personDataBorneTaxpayer(personDataBorneTaxpayerObj);
// e1.setE1reduceTax(reduceTaxObj);
e1.setE1taxableIncomes(taxableIncomeObj);
e1.setE1taxPayerBankAccount(taxPayerBankAccountObj);
// e1.setE1nauticalincomes(nautical);
e1.setRelatePersons(relatePersons); // Many to Many
```

```

return e1;

}

private E2 buildE2WithWifeAndHouses(int taxPayerID) {
    // E2 primary KEY
    E2Id key = new E2Id();
    key.setTaxierId(taxPayerID);
    key.setYear(Calendar.getInstance().get(Calendar.YEAR));

    // master table !
    E2 e2 = new E2();
    e2.setId(key);
    e2.setIsComplete(1);
    e2.setDateInserted(new java.util.Date());

    E2estate e2estateObj = new E2estate();
    e2estateObj.setArea(35);
    e2estateObj.setFrom(1);
    e2estateObj.setLocation("Ιστρού 125 Αθήνα 11142");
    e2estateObj.setMonthlyRental(58.96f);
    e2estateObj.setPosition("Ισόγειο");
    e2estateObj.setPercentCoOwner(50);
    e2estateObj.setTenantAfm("004556677");
    e2estateObj.setTenantFullName("Μάριος Κασσωτάκης");
    e2estateObj.setTo(12);
    e2estateObj.setUsage("Κατάστημα");
    e2estateObj.setRevenueFreeOffice(353.14f);

    E2coOwner e2coOwners = new E2coOwner();
    e2coOwners.setAddress("Ιστρού 125 Αθήνα 11142");

```

```
e2coOwners.setAfm("955821070");
e2coOwners.setFullName("Ιστρου 109 Αθήνα 11142");
e2coOwners.setPercent(50);
e2coOwners.setLocation(e2estateObj.getLocation().trim());
e2coOwners.setPosition(e2estateObj.getPosition().trim());
e2coOwners.setUsage(e2estateObj.getUsage().trim());
e2coOwners.setArea(e2estateObj.getArea());
```

```
E2estate e2estateCretaObj = new E2estate();
e2estateCretaObj.setArea(80);
e2estateCretaObj.setFrom(9);
e2estateCretaObj.setLocation("Γιαμπουδάκη 3 Ρέθυμνο 74100");
e2estateCretaObj.setMonthlyRental(300f);
e2estateCretaObj.setPosition("Ισόγειο");
e2estateCretaObj.setPercentCoOwner(100);
e2estateCretaObj.setTenantAfm("004864321");
e2estateCretaObj.setTenantFullName("Μανώλης Λεγάκης");
e2estateCretaObj.setTo(12);
e2estateCretaObj.setUsage("Οικία");
e2estateCretaObj.setRevenueFreeHome(1200);
```

```
E2estate e2estate300Obj = new E2estate();
e2estate300Obj.setArea(301);
e2estate300Obj.setFrom(1);
e2estate300Obj.setLocation("Θέμιδος 17 Κηφισιά 14564");
e2estate300Obj.setMonthlyRental(3000);
e2estate300Obj.setPosition("Ισόγειο");
e2estate300Obj.setPercentCoOwner(100);
e2estate300Obj.setTenantAfm("109852147");
e2estate300Obj.setTenantFullName("Γεωργόπουλος Ιωάννης");
e2estate300Obj.setTo(12);
e2estate300Obj.setUsage("Οικία");
```

```
e2estate300Obj.setRevenueFreeHome(36000);
```

```
E2estate e2estateVineryObj = new E2estate();  
e2estateVineryObj.setArea(20000);  
e2estateVineryObj.setFrom(1);  
e2estateVineryObj.setLocation("Δειλινάτα Κεφ/νιά 28100");  
e2estateVineryObj.setMonthlyRental(29.35f);  
e2estateVineryObj.setPosition("Επιφ");  
e2estateVineryObj.setPersentCoOwner(100);  
e2estateVineryObj.setTenantAfm("123456780");  
e2estateVineryObj.setTenantFullName("Θεόδωρος Χρήστου");  
e2estateVineryObj.setTo(12);  
e2estateVineryObj.setUsage("Αμπελώνας");  
e2estateVineryObj.setRevenueFreeField(352.2f);
```

```
E2otherEstate otherEst = new E2otherEstate();  
otherEst.setArea(115f);  
otherEst.setLocation("Ίωνος 130 Παλαιά Πεντέλη 15236");  
otherEst.setPosition("Ισογειο");  
otherEst.setTitle("12345/20-12-2011 Συμβο/φου Αθηνών Νότη Νοταρά");  
otherEst.setUsage("Οικία");
```

```
Set<E2estate> listOfE2estates = new HashSet<E2estate>();  
Set<E2coOwner> listOfE2coOwner = new HashSet<E2coOwner>();  
Set<E2otherEstate> listOfOtherEstates = new HashSet<E2otherEstate>();
```

```
listOfOtherEstates.add(otherEst);  
listOfE2coOwner.add(e2coOwners);  
listOfE2estates.add(e2estateObj);  
listOfE2estates.add(e2estate300Obj);  
listOfE2estates.add(e2estateVineryObj);  
listOfE2estates.add(e2estateCretaObj);
```



```

    e2.setE2otherEstates(listOfOtherEstates);
    e2estateObj.setE2coOwners(listOfE2coOwner);
    e2.setE2estates(listOfE2estates);

    return e2;
}

private E1 buildE1WithWifeAndHouses(int taxPayerID) {
    Set<RelatePerson> relatePersons = new HashSet<RelatePerson>();

    Contact wifeContact = new Contact("2103581480", "6971890646", "");

    RelatePerson wife = new RelatePerson();
    wife.setAddress("ΠΟΔΟΥ 58");
    wife.setAfm("015190021");
    wife.setAtid("Z982408");
    wife.setContact(wifeContact);
    wife.setFatherName("ΝΙΚΟΛΑΟΣ");
    wife.setFname("ΜΑΡΙΑ");
    wife.setLname("ΚΩΠΕ");
    wife.setType(1);

    relatePersons.add(wife);

    E1Id key = new E1Id(taxPayerID, Calendar.getInstance().get(
        Calendar.YEAR));

    E1infoData info = new E1infoData();
    info.set_015(1);

    E1objectiveSpending objectiveSpendingObj = new E1objectiveSpending();

```

```
objectiveSpendingObj.setHouseAddr1("ΡΟΔΟΥ 58 - ΛΕΡΟΥ - ΚΩ - ΠΕΡΕΒΟΥ");
objectiveSpendingObj.setFloor1("6");
objectiveSpendingObj.set_205("98207");
objectiveSpendingObj.set_211(280f);
objectiveSpendingObj.set_212(20f);
objectiveSpendingObj.set_213(100f);
objectiveSpendingObj.set_215(12);
objectiveSpendingObj.set_216(1900f);
objectiveSpendingObj.setHouseAddr1("ΡΟΔΟΥ 58 - ΛΕΡΟΥ - ΚΩ - ΠΕΡΕΒΟΥ");
objectiveSpendingObj.setFloor2("1");
objectiveSpendingObj.set_218(90f);
objectiveSpendingObj.set_219(10f);
objectiveSpendingObj.set_220(100f);
objectiveSpendingObj.set_222(12);
objectiveSpendingObj.set_216(1900f);
objectiveSpendingObj.setId(key);
```

```
E1taxableIncomes income = new E1taxableIncomes();
income.set_302(12600f);
income.set_303(12500f);
income.set_321(3815f);
income.set_103(7362.96f);
income.set_105(1526.04f);
income.set_101(352.2f);
income.set_909(604.08f);
income.set_129(3521.64f);
income.set_147(2400f);
income.set_123(4402f);
income.set_151(1864.82f);
income.set_175(15084.4f);
income.set_131(1200f);
income.setId(key);
```

```
E1prepaidTaxes prepaidTaxesObj = new E1prepaidTaxes();
prepaidTaxesObj.set_313(871.5f);
prepaidTaxesObj.set_314(508f);
prepaidTaxesObj.set_315(858.43f);
prepaidTaxesObj.set_316(500.38f);
prepaidTaxesObj.setId(key);
```

```
E1taxPayerBankAccount taxPayerBankAccountObj = new E1taxPayerBankAccount();
taxPayerBankAccountObj.setBic("032");
taxPayerBankAccountObj.setIban("GR0908001234008002340597900");
taxPayerBankAccountObj.setId(key);
```

```
E1 e1 = new E1();
e1.setId(key);
e1.setIsComplete(1);
e1.setAtid("I308501");
e1.setDoy("Αλίμου");
e1.setMarriage(1);
e1.setTaxpayerAddress("ΠΟΔΟΥ 58");
e1.setDateInserted(new java.util.Date());
// e1.setE1dataFromTaxPayerFolder(dataFromTaxPayerFolderObj);
// e1.setE1expensesRemovedFromTotalIncome(removed);
// e1.setE1incomesReduceTaxes(incomesReduceTaxesObj);
// e1.setE1infoData(infoDataObj);
e1.setE1objectiveSpending(objectiveSpendingObj);
// e1.setE1personDataBorneTaxpayer(personDataBorneTaxpayerObj);
e1.setE1prepaidTaxes(prepaidTaxesObj);
// e1.setE1personDataBorneTaxpayer(personDataBorneTaxpayerObj);
// e1.setE1reduceTax(reduceTaxObj);
e1.setE1taxableIncomes(income);
e1.setE1taxPayerBankAccount(taxPayerBankAccountObj);
```

```
// e1.setE1nauticalincomes(nautical);
e1.setRelatePersons(relatedPersons); // Many to Many

return e1;
}

public String zeroPadLeft(int s) {
    // int n
    // return String.format("%1$" + n + "s", s);
    return String.format("%09d", s);
}
}
```

Πανεπιστήμιο Πειραιώς