ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

# Ενισχυμένη διασφάλιση δεδομένων στο υπολογιστικό νέφος: Μια κατανεμημένη προσέγγιση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ντούφα Βασίλη
321/2011112

**Επιβλέπων :** Κοκολάκης Σπυρίδων, Αναπληρωτής Καθηγητής
Τμήμα Μηχανικών Πληροφοριακών και
Επικοινωνιακών Συστημάτων

**Μέλη εξεταστικής επιτροπής:**

Μαρία Καρύδα, Επίκουρη Καθηγήτρια,
Παν. Αιγαίου

Παναγιώτης Ριζομυλιώτης, Επίκουρος Καθηγητής,
Χαροκόπειο Πανεπιστήμιο

Σάμος 5/9/2018

# Πρόλογος

Η παρούσα πτυχιακή εργασία με τίτλο **«Ενισχυμένη διασφάλιση δεδομένων στο υπολογιστικό νέφος: Μια κατανεμημένη προσέγγιση»** εκπονήθηκε στο Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων, του Πανεπιστημίου Αιγαίου, στα πλαίσια του προπτυχιακού προγράμματος σπουδών του Τμήματος.

Η βασική ιδέα για τη διπλωματική αυτή εργασία γεννήθηκε μετά από υλοποιήσεις που έκανα για τα εργαστηριακά μέρη των μαθημάτων «Ασφάλεια Πληροφοριακών Συστημάτων», «Κατανεμημένα Συστήματα» και «Τεχνολογίες Δικτύων και Νέφους». Κατά την ενασχόλησή μου με τον προγραμματισμό για τα μαθήματα αυτά, μου δόθηκε η ευκαιρία αρχικά να συλλάβω μια πρωτότυπη δομή, έπειτα να σχεδιάσω την αρχιτεκτονική της και τέλος την υλοποιήσω.

Ο προγραμματισμός, που ιδιαίτερα αγαπώ και που λειτούργησε ως ο κύριος πόλος έλξης για το συγκεκριμένο τμήμα του Πανεπιστημίου Αιγαίου για τις σπουδές μου, η δομή και τα ζητούμενα των παραπάνω μαθημάτων, με οδήγησαν στη σύλληψη των όσων θα μελετήσετε στη συνέχεια.

Ενώ οι εφαρμογές της πληροφορικής έχουν εισέλθει σε κάθε πτυχή της καθημερινότητας του σύγχρονου πολιτισμού και οι χρήστες εξοικειώνονται στη χρήση της τεχνολογίας με πολύ διαφορετικούς ρυθμούς από ότι πριν 30 χρόνια, δεν είναι σαφές η έκταση της ικανότητας τους να κρίνουν από μόνοι τους και αποτελεσματικά το βαθμό που τα δεδομένα τους είναι ασφαλή. Τόσο σε τοπικό επίπεδο όσο και σε επίπεδο νέφους τα προϊόντα που χρησιμοποιούμε σε καθημερινή βάση (έξυπνα τηλέφωνα, τηλεοράσεις, υπολογιστές, υπηρεσίες ηλεκτρονικής αλληλογραφίας, υπηρεσίες νέφους κ.α.), υπόσχονται πως μας εξασφαλίζουν το απόρρητο των δεδομένων μας. Το πως όμως εκλαμβάνει ο χρήστης την υπόσχεση αυτή και το αν γνωρίζει τις τεχνικές λεπτομέρειες πίσω κάθε τέτοια υπόσχεση, δεν μπορούμε να το ξέρουμε. Υποπτευόμαστε ωστόσο (αν και θα πρέπει να το δεχόμαστε ως δεδομένο) πως ούτε το γνωστικό αντικείμενο των χρηστών είναι σχετικό με την πληροφορική, αλλά ούτε πως μπορεί να εξετάσει πως τα λεγόμενα των επιχειρήσεων ανταποκρίνονται στην πραγματικότητα. Το πρόβλημα διογκώνεται αν στην εξίσωση προσθέσουμε πως πλέον τα δεδομένα των χρηστών μεταφορτώνονται στη νέφος συνεχώς και ο μόνος τρόπος που έχει ένας χρήστης στη διάθεση του για να εξετάσει την ασφάλεια των δεδομένων του, είναι οι διαπιστεύσεις που λαμβάνει κάθε εταιρία από φορείς που διαπιστεύουν τις εταιρίες για την ορθή λειτουργία τους (π.χ. κατά ISO[1]).

Με λίγα λόγια δεν ξέρουμε αν ο χρήστης γνωρίζει πως να προφυλάξει τα δεδομένα του μόνος του, αλλά ούτε αν πρέπει να εμπιστευτούμε τις υπηρεσίες που χρησιμοποιούμε. Τετριμμένη διαπίστωση μεν, πάντα ισχύουσα δε.

Στην παρούσα εργασία που συνοδεύεται από εφαρμογή προτείνουμε μια τεχνική διαφύλαξης των δεδομένων των χρηστών, με μεθόδους διαχωρισμού, κρυπτογράφησης, μετονομασίας και διαμοιρασμού μεταξύ πολλών υπηρεσιών νέφους που τη στιγμή που γράφεται η εργασία θεωρούνται ασφαλείς.

# ABSTRACT

This thesis called “Enhanced data security in cloud computing; A distributed approach"
was prepared for Department of Information & Communication Systems Engineering of the University of the Aegean during the BsC curriculum.

The Cloud is nowadays a widespread way of sharing and saving data. The end users take advantage of it’s benefits due to lack of local storage and it’s ease of sharing documents and multimedia. The security of the files was supposed to be a solved issue as encryption used by the cloud services for data in storage was supposed to suffice. In the early and min 2010’s major corporate data breaches showed that there were a lot to be done in order to consider that data stored to the cloud are on the safe side. While the end user is responsible for the data uploaded to the cloud, we consider a technique to help end users take advantage of cloud services’ benefits with less skepticism.

This paper is accompanied by an implementation which takes care of all the aspects considered in the document and is used as a proof of concept (P.o.C.) of the solution we provide.

The main concern of the paper (and the application) is to provide an enhanced way for users to secure the files they distribute to cloud services, prior to uploading them, in such a way that only them have access to their files. User files are slit, encrypted, renamed and shared across multiple cloud services to ensure that there will be no single point of failure, capable of compromising the files.

Until cloud services come up with a model that balances the security of the files and the computational power needed for fast delivery of the files, this methodology could be used be end user applications to prevent critical data safe.

# Ευχαριστίες

Ευχαριστώ τον καθηγητή μου κ. Κοκολάκη Σπύρο για την ευκαιρία που μου έδωσε να δουλέψω πάνω στην ιδέα μου, να της δώσω υπόσταση και να προσφέρω από τη μεριά μου με τον τρόπο αυτό στην ακαδημαϊκή κοινότητα, και για το χρόνο που αφιέρωσε στη δουλειά μου. Ευχαριστώ την οικογένεια μου για την ηθική υποστήριξη μέσα στα χρόνια.

# Πίνακας περιεχομένων

# Περίληψη

Κεντρικός πυλώνας της εργασίας είναι διασφάλιση της εμπιστευτικότητας των αρχείων που αποθηκεύονται στο νέφος με μια απλή και πρωτότυπη λογική, όπως αναλυτικά περιγράφεται στη συνέχεια. Για να γίνει κατανοητή και να παρουσιαστεί η λογική αυτή αναπτύχθηκε εφαρμογή με δυνατότητα ο χρήστης να χρησιμοποιήσει μια σειρά παρόχων υπηρεσιών αποθήκευσης αρχείων στο νέφος(cloud storage services) και να ανεβάσει εκεί τα αρχεία του, εφόσον έχει προηγηθεί η επεξεργασία τους από πρόγραμμα . Η εφαρμογή, προσανατολισμένη στην ασφάλεια του περιεχομένου του χρήστη, επιτρέπει να μεγιστοποιηθεί η ασφάλεια των αρχείων που ανταλλάσσονται με το νέφος(cloud).

Η ανάγκη προκύπτει από την ασφάλεια που νοιώθει κάθε χρήστης (η εταιρία), όταν τα δεδομένα του είναι αποθηκευμένα «κοντά του» και αντίστοιχα την ανασφάλεια και τον σκεπτικισμό με τον οποίο αντιμετωπίζει το ενδεχόμενο να εμπιστευτεί τα δεδομένα του στο νέφος, επειδή τα δεδομένα του δεν είναι «κοντά του» σε φυσικό επίπεδο. Φυσικά το αν μπορεί να προστατέψει τα δεδομένα του όταν αυτά είναι στην κατοχή του (και «κοντά του») είναι ένα μεγάλο κεφάλαιο που απασχολεί την επιστήμη της πληροφορικής από τη γέννηση της και δεν απασχολεί την παρούσα εργασία.

Δεν είναι παράλογο να σκέφτεται κανείς αρνητικά την ιδέα του νέφους. Η υποκλοπή των δεδομένων μας μπορεί να γίνει όμως και από το τοπικό σκληρό μας δίσκο, όσο και από τον απομακρυσμένο σκληρό δίσκο (ή δίσκους) του νέφους. Θα προσπαθήσουμε να αποδώσουμε μια εικόνα ασφάλειας στο νέφος, ενισχύοντας τους δικούς του μηχανισμούς ασφάλειας για να φέρουμε πιο κοντά στη χρήση του τον τελικό χρήστη.

Στο παρόν περιγράφουμε την τεχνική που έχει ως σκοπό να διασφαλίσει το περιεχόμενο των αρχείων που αποστέλλουμε στο νέφος και τους λόγους που αυτή η λογική θωρακίζει της ασφάλεια μας, τόσο τοπικά όσο και στο νέφος. Για να καταλήξουμε σε αυτό, κάνουμε πρώτα μια αναδρομή στην χρήση του Νέφους, εξηγούμε τις υπηρεσίες που προσφέρει, παρουσιάζουμε τους λόγους που προτιμώνται οι υπηρεσίες νέφους, απαριθμώντας τους κινδύνους που έχουν οι συμβατικές μέθοδοι αποθήκευσης , αντιπαραβάλουμε τα πλεονεκτήματα που έχει στους τομείς της ανθρώπινης δραστηριότητας, αναδεικνύουμε προβλήματα διασφάλισης δεδομένων, αναπτύσσουμε αιτίες που μας οδήγησαν στην ανάπτυξη της νέας τεχνικής και τελικά αναλύουμε την τεχνική που ακολουθούμε ενώ περιγράφουμε σε σημεία τον προγραμματιστικό κώδικα(source code) που εκτελεί τις σημαντικές λειτουργίες στην εφαρμογή που συνοδεύει την εργασία και είναι υπεύθυνος για την διασφάλιση των δεδομένων.

Το θέμα της ασφάλειας των δεδομένων είναι πολύ μεγάλο και μέσα στο πλαίσιο αυτό προσπαθήσαμε να περιορίσουμε την έκταση της εργασίας στα στεγανά της ιδέας-πυλώνα της. Αυτό δεν θα πρέπει να περιορίσει τη φαντασία του μελετητή-αναγνώστη. Μια εργασία που θα εξασφάλιζε την πλήρη προστασία των δεδομένων από όλες τις απειλές ξεφεύγει από το πεδίο εφαρμογής της εργασίας αυτής. Η υπεύθυνη όμως οπτική στο ακαδημαϊκό καθήκον μας μας υπαγορεύει όχι μόνο να μην κρύψουμε ζητήματα που ανακύπτουν άλλα αντίθετα μάλιστα να τα αναδείξουμε. Στη λογική αυτή αφιερώνουμε ένα κεφάλαιο στην εξέταση μερικών πτυχών που

θεωρούμε πως μπορούνε να αποτελέσουν επέκταση της παρούσας εργασίας, από ένα μεταπτυχιακό φοιτητή ή στο σύνολο τους μια διδακτορική διατριβή, ίσως και ένα πλήρες εμπορικό προϊόν.

# 1 *Εισαγωγή*

## *1.1 Γενικά*

Η ασφάλεια των δεδομένων ήτανε, είναι και θα παραμείνει κεντρικός άξονας όλων των δραστηριοτήτων που εμπεριέχουν χρήση ψηφιακών τεχνολογιών. Το ιατρικό μας ιστορικό, ευαίσθητες γραφειοκρατικές διαδικασίες, on-line αγορές και λοιπές οικονομικές συναλλαγές όπως εφοριακές υποχρεώσεις, εταιρικά δεδομένα και ανταγωνιστικά μυστικά, οι κωδικοί της ηλεκτρονικής μας αλληλογραφίας, είναι μερικά παραδείγματα ευαίσθητων δεδομένων που χρήζουν προστασίας.

Σε αρκετά από τα παραπάνω παραδείγματα εντοπίζουμε την άμεση σύνδεση προσωπικών και ευαίσθητων στοιχείων που προστατεύονται μάλιστα και από νομικά πλαίσια (ιατρικό απόρρητο) ή ρήτρες εμπιστευτικότητας μεταξύ των εμπλεκόμενων μερών (εταιρικά δεδομένα). Το γεγονός την αποθήκευσης την πλειονότητας των ανωτέρω δεδομένων φτάνει να απασχολεί τόσο τους χρήστες σε προσωπικό επίπεδο αλλά και τα νομοπαρασκευαστικά σώματα και δηλώνει με σαφήνεια τη σημασία της απόκρυψης τους από αδιάκριτα βλέμματα, περισσότερο δε από κακοπροαίρετες λογικές.

Τα πλαίσια της εργασίας αυτής δεν επιτρέπουν αλλά ούτε και δικαιολογούν την εκτενή αναφορά μας σε πολιτικές ασφάλειας και εξειδικευμένες προσεγγίσεις τεχνικού τύπου. Σκοπός μας είναι η αναλυτική περιγραφή του προβλήματος που αναγνωρίζουμε και σκοπεύουμε να προσεγγίσουμε βελτιωτικά.

Προτείνουμε μια υλοποίηση που θεωρούμε βέλτιστη για χρήση από όσο το δυνατόν μεγαλύτερο κοινό ενώ την ίδια στιγμή εξασφαλίζουμε την ευκολία χρήσης, της υλοποίησης που παρουσιάζουμε, χωρίς να χάνουμε κανένα μέρος της λειτουργικότητας. Γι' αυτό το λόγο δεν θα περιορίσουμε τις αναφορές μας στα πέριξ του θέματος ζητήματα, φροντίζοντας να αποδώσουμε και να προσδιορίσουμε με τον καλύτερο δυνατό τρόπο το υπάρχον πλαίσιο στο οποίο προτείνουμε βελτιστοποίηση.

Για να προστατέψουμε αποτελεσματικά τα δεδομένα μας πρέπει να αποσοβήσουμε τους διαφορετικούς τύπους απειλών, που προέρχονται από ετερόκλητα περιβάλλοντα και λειτουργούν με ακαθόριστους τρόπους, για τα θύματα, ώστε να πετύχουν το στόχο τους, που

είναι η προσβολή και απόκτηση πρόσβασης στα δεδομένα που υπό κανονικές συνθήκες δε θα είχανε πρόσβαση. Η σημαντικότητα των δεδομένων, η φύση τους (ευαίσθητα ή όχι) και ο τρόπος που θα προσεγγίσουμε το πρόβλημα αυτό, διαμορφώνεται ανάλογα με το σενάριο που εξετάζουμε ενώ τα αντίμετρα που πρέπει να εφαρμόσουμε χρήζουν αδιάκοπης επανεξέτασης, δεδομένης της συνεχούς εξέλιξης των τεχνολογιών που χρησιμοποιούνται από τους οργανισμούς, τους τελικούς χρήστες αλλά προπαντός από τους επιτιθέμενους.

## *1.2 Αναγνωρίζοντας κινδύνους και απειλές*

Στην προσπάθεια μας να τεκμηριώσουμε την μεθοδολογία που προτείνουμε στην παρούσα εργασία πόνημα για την επιπλέον ασφάλεια που παρέχει η πρόταση μας, θα πρέπει αρχικά να αναγνωρίσουμε τους βασικούς κινδύνους και τις απειλές που σχετίζονται με τα δεδομένα μας, να προσφέρουμε την εικόνα αυτή μέσα από το πρίσμα της δικής μας οπτικής, χωρίς αυτό να σημαίνει πως παραγκωνίζουμε ή αποσοβούμε άλλες πτυχές της ασφάλειας των δεδομένων, προς όφελος της δικής μας τοποθέτησης. Άλλωστε η ασφάλεια στα πληροφοριακά συστήματα και δει η ασφάλεια της πληροφορίας, πρέπει να αναγνωρίζεται ως σύνθεση πολλών μερών που κανένα δε μπορεί να θεωρηθεί λιγότερο σημαντικό από κάποιο άλλο.

**Ευπάθεια**: ορίζεται ως το σχεδιαστικό(by design) ελάττωμα ενός συστήματος, εφαρμογής ή του συνδυασμού των δύο που μπορεί υπό προϋποθέσεις να οδηγήσει στην παραβίαση της ασφάλειας και της ακεραιότητας του συστήματος.

**Απειλή:** ορίζεται ένα μη επιθυμητό γεγονός που θα προκαλέσει μη διαθεσιμότητα (availability) ενός πληροφοριακού συστήματος, που με πρόθεση ή ακούσια αλλοιώνει ή καταστρέφει ή μπορεί χωρίς την κατάλληλη εξουσιοδότηση να αποκαλύψει ευαίσθητες πληροφορίες.

**Κίνδυνος**: ορίζεται η κατάσταση κατά την οποία μια συγκεκριμένη απειλή καταλήγει να μπορεί να εκμεταλλευτεί μια συγκεκριμένη ευπάθεια. Ο κίνδυνος εκφράζει το ενδεχόμενο για απώλεια.

Οι Απειλές χωρίζονται σε 3 βασικές κατηγορίες, τις φυσικές απειλές , τις αστοχίες υλικού και τις Ανθρώπινες Απειλές.

### *1.2.1 Φυσικές Απειλές*

Σε αυτή την κατηγορία κατατάσσουμε καταστροφές που σχετίζονται με φυσικά αίτια όπως πυρκαγιές, σεισμοί ή πλημμύρες. Οι απειλές αυτές αντιμετωπίζονται με τα αντίστοιχα φυσικά αντίμετρα σε επίπεδο κτηριακών υποδομών (αντισεισμικά κτήρια), εξοπλισμού κατάσβεσης και πολεοδομικών κανόνων που προβλέπουν τέτοιου τύπου προβληματικές εγκαταστάσεις.

### *1.2.2 Αστοχίες Υλικού*

Οι αστοχίες υλικού είναι μια κατηγορία απειλών περισσότερο συνηθισμένη από τις φυσικές απειλές. Για αυτό το λόγο οι σωστά ενημερωμένοι χρήστες, οι εταιρίες αλλά και οι μεγάλοι οργανισμοί χρησιμοποιούν διπλό εξοπλισμό αποθήκευσης . Εξασφαλίζεται έτσι η συνέχεια της δραστηριότητας όλων των παραγόντων που προαναφέρθηκαν. [] Τα δεδομένα μπορεί να αλλοιωθούν, από ένα χαλασμένο σκληρό δίσκο για παράδειγμα, και να μην είναι δυνατή η ανάκτηση τους για ανάγνωση από αυτό το μέσο.

### *1.2.3 Ανθρώπινες Απειλές*

Η κατηγορία αυτή είναι η λιγότερο προβλέψιμη, πιο δύσκολα εντοπίζεται από τα θύματα και τα αντίμετρα που καλούνται τα εμπλεκόμενα μέρη να πάρουν, είναι πολύ περισσότερα από τα αντίμετρα που προβλέπονται, συνολικά, για τις δύο προηγούμενες κατηγορίες. Λέμε "περισσότερα" επειδή οι τρόποι που μπορεί ο επιτιθέμενος να χρησιμοποιήσει ώστε να εκμαιεύσει πληροφορία δύναται να μην απαριθμούνται το ίδιο εύκολα με τις προηγούμενες δύο κατηγορίες. Ο λόγος είναι πως αυτή η κατηγορία χωρίζεται στις απειλές που συμβαίνουν α) με **σκοπιμότητα** β) **τυχαία**.

Σχεδιάζοντας την ασφάλεια των δεδομένων μας πρέπει να σκεφτούμε σε δύο(2) επίπεδα. Το πρώτο είναι το φυσικό επίπεδο ασφάλειας (physical security) και το δεύτερο είναι το επίπεδο δεδομένων.

## *1.3 Διασφαλίζοντας τα δεδομένα από απειλές σε φυσικό επίπεδο*

Σχεδιάζοντας την ασφάλεια των δεδομένων μας πρέπει να σκεφτούμε σε δύο(2) επίπεδα. Το πρώτο είναι το φυσικό επίπεδο ασφάλειας (physical security) και το δεύτερο είναι το επίπεδο δεδομένων.

Εξετάζοντας στην προηγούμενη παράγραφο τους κινδύνους που εκτίθενται τα δεδομένα μας, θα πρέπει στη συνέχεια να αναφέρουμε συνοπτικά και τρόπους για να τα διασφαλίσουμε από αυτούς. Στην παράγραφο 1.2.2 αναφερθήκαμε στις διπλές εγκαταστάσεις. Στην πραγματικότητα ο πλεονασμός (redundancy) στον εξοπλισμό μας επιτρέπει την αδιάλειπτη λειτουργία των συστημάτων μας εφόσον εμφανιστεί αστοχία ή δημιουργηθεί σκόπιμα πρόβλημα σε κάποιο αποθηκευτικό μέσο, υπολογιστή, διακομιστή, δικτυακή συσκευή κτλ. Σε επίπεδο δεδομένων αυτό σημαίνει πως η αστοχία π.χ. ενός μέσου αποθήκευσης , που εκμεταλλεύεται λειτουργία συστοιχίας (RAID mirroring ή όπως αλλιώς συμβολίζεται RAID 1), δε θα μας οδηγήσει σε απώλεια δεδομένων, αφού η τελευταία γράφει τα δεδομένα σε διαφορετικά αποθηκευτικά μέσα, εξασφαλίζοντας redundancy ένα-προς-ένα. Ταυτόχρονα ο τελικός χρήστης δε σταματά να έχει πρόσβαση στα δεδομένα του αλλά χρειάζεται μόνο η αντικατάσταση του μέσου αποθήκευσης που έχει υποστεί ζημιά. Φυσικά είτε με τη χρήση συστοιχίας, είτε χωρίς αυτή, κρίνεται σκόπιμο να διατηρούνται πάντα αντίγραφα ασφαλείας (backup), έτσι ώστε να μπορούν να ανακτηθούν τα αρχεία μας.

### *1.3.1 Φυσικό Επίπεδο*

Με τον όρο «Φυσικό Επίπεδο», στο παρόν, αναφερόμαστε στο μέσο και στο χώρο που χρησιμοποιούμε για να αποθηκεύσουμε τα δεδομένα μας, αναφερόμαστε στην ασφάλεια των μηχανημάτων και των συσκευών που αποθηκεύουν τα δεδομένα μας αλλά και στο χώρο που αυτά είναι εγκατεστημένα.

Τα δεδομένα πρέπει να προστατεύονται με τον καλύτερο δυνατό τρόπο από αστοχίες υλικού, καταστροφές υλικού και από αστάθμητους παράγοντες που αφορούν το περιβάλλον του υλικού, στο οποίο αυτά αποθηκεύονται. Τα μηχανικά μέσα αποθήκευσης έχουν μεγαλύτερο ποσοστό αστοχίας από τα ηλεκτρονικά μέσα και αυτά με τη σειρά τους κινδυνεύουν από ηλεκτροστατικά φορτία. Επειδή μας είναι δύσκολο να προβλέψουμε όλα τα σενάρια

καταστροφής των μέσων αποθήκευσης (και κατ' επέκταση και των δεδομένων μας), φροντίζουμε για την διατήρηση ενημερωμένων εφεδρικών αντιγράφων (backup).

Είτε για προσωπική χρήση είτε για επαγγελματικό επίπεδο διατίθενται στην αγορά λύσεις όπως φορητά αποθηκευτικά μέσα USB (flash drives, sticks), «εξωτερικοί» σκληροί δίσκοι και δικτυακές συσκευές αποθήκευσης (NAS) . Οι οικιακοί χρήστες προτιμούν να εξασφαλίσουν πως τα δεδομένα τους είναι ασφαλή με τη χρήση συσκευών αποθήκευσης και το ίδιο φαίνεται να συμβαίνει και στις μικρές επιχειρήσεις. Μεσαίες και μεγάλες επιχειρήσεις φροντίζουν με τους δικούς τους διακομιστές αποθήκευσης αρχείων (Storage Server) να διατηρούν τακτικά backup και να δίνουν τοπική πρόσβαση στα αρχεία των χρηστών τους . Επόμενο είναι να πρέπει να διασφαλιστεί και η φυσική ασφάλεια των φυσικών μέσων αποθήκευσης , αλλά και των μέσων στα οποία απόθηκεύουμε τα διπλότυπα αρχεία μας.

### *1.3.2 Η σύγχρονη προσέγγιση - Νέφος*

Η πιο σύγχρονη προσέγγιση στο χώρο της αποθήκευσης αρχείων, αφορά στο νέφος. Την στιγμή της συγγραφής του παρόντος υπάρχουν στην αγορά πληθώρα παρόχων της εν λόγω υπηρεσίας που προσφέρουν υπηρεσίες για οικιακούς χρήστες (home users) μέχρι εταιρικής κλίμακας στοχευμένες υπηρεσίες (enterprise level storage solutions). Οι υπηρεσίες αυτές εξασφαλίζουν με τη σειρά τους τα δεδομένα που φιλοξενούν, με τη χρήση πολύπλοκων συστοιχιών Raid Arrays και εξελιγμένων κέντρων δεδομένων(Data Centre), ελαχιστοποιώντας την περίπτωση να χαθούν/καταστραφούν τα αποθηκευμένα αρχεία από παράγοντες που θα επηρεάσουν το φυσικό κομμάτι. Επιστεγάζοντας την ασφάλεια των δεδομένων που αποθηκεύουν οι χρήστες σε αυτές, οι Cloud Storage Service Providers, εφαρμόζουν και τεχνικές κρυπτογράφησης των δεδομένων, κάτι θα μας απασχολήσει στη συνέχεια.

Μάλιστα η διευκόλυνση, ο χρήστης να έχει στη διάθεσή του τα αρχεία του μέσω φυλλομετρητή (browser) ενός έξυπνου τηλεφώνου(smartphone) ή άλλης φορητής συσκευής, οπουδήποτε και αν βρίσκεται, δίνει σοβαρό προβάδισμα στις Cloud Storage Services έναντι των παραδοσιακών μεθόδων. Μάλιστα οι μικρές και μεσαίες επιχειρήσεις τείνουν να χρησιμοποιούν όλο και περισσότερο αυτή τη μορφή αποθήκευσης . Οι εταιρίες που δραστηριοποιούνται εμπορικά στο χώρο της υπηρεσίας cloud storage παρέχουν δελεαστικά πακέτα μεγάλης χωρητικότητας και πολλαπλών λογαριασμών και άλλων διακεκριμένων λειτουργιών.

Η διαδικασία της αποθήκευσης των αρχείων στο νέφος απλοποιείται περισσότερο με τη χρήση εφαρμογών(applications) για υπολογιστές και φορητές συσκευές.

## *1.4 Ακεραιότητα Δεδομένων*

Στο τοπικό επίπεδο ο χρήστης πολύ δύσκολα θα καταλάβει άν ένα αρχείο του έχει αλλοιωθεί. Συνήθως ενημερώνεται από κάποια εφαρμογή πως το αρχείο του είναι κατακερματισμένο(corrupted) και τους λόγους περιγράψαμε νωρίτερα στο κεφάλαιο 1.2.2.

Η καθιερωμένη τεχνική που ακολουθούμε στις περιπτώσεις που θέλουμε να προστατέψουμε δεδομένα είναι η κρυπτογράφηση τους (encryption). Με τον όρο κρυπτογράφηση αναφερόμαστε

στην μετατροπή ενός μηνύματος σε μορφή ακατανόητη για τον άνθρωπο, μέσω χρήσης ειδικών αλγορίθμων, ούτως ώστε να μη μπορεί να διαβαστεί από μη εξουσιοδοτημένος παραλήπτες. Η κρυπτογράφηση των δεδομένων αρκεί ώστε να διατηρούμε τα αρχεία μας ασφαλή από τα αδιάκριτα μάτια και τις πληροφορίες που τα δεδομένα μας περιέχουν, κρυφά. Για να διαβάσει ο εξουσιοδοτημένος παραλήπτης το αρχικό μήνυμά χρειάζεται τον αλγόριθμο με το οποίο το μήνυμα μετετράπη στην ακατανόητη μορφή του, και επιπλέον ένα κλειδί που χρησιμοποιήθηκε ως είσοδος στον αλγόριθμο κρυπτογράφησης.

Παράλληλα με την αυξανόμενη ανάγκη για ιδιωτικότητα και προστασία των δεδομένων αναπτύχθηκαν και αλγόριθμοι κρυπτογράφησης. Στην πάροδο του χρόνου ο τρόπος λειτουργίας κάποιων αλγορίθμων παραβιάστηκε και αντικαταστάθηκαν από νεότερους, περισσότερο ασφαλείς, όπως για παράδειγμα ο αλγόριθμος DES (Data Encryption Standard) ο οποίος αντικαταστάθηκε από την εισαγωγή του AES.

Όπως αναφέραμε και σε προηγούμενη παράγραφο η ασφάλεια ενός συστήματος αναθεωρείται ανάλογα με την εφαρμογή και το σενάριο χρήσης, για αυτό το σκοπό η κρυπτογράφηση στην πιο διαδεδομένη μεταφορά δεδομένων (internet) έχει τη δική της συγκεκριμένη υλοποίηση. Για αυτό το σκοπό οι διαδικτυακές εφαρμογές και ιστοσελίδες χρησιμοποιούν TLS (Transport Layer Security) και παλαιότερα το SSL (Secure Socket Layer).

### *1.4.1 Cloud και εμπιστευτικότητα δεδομένων*

Για να χρησιμοποιήσει ο χρήστης μια Cloud Storage υπηρεσία πρέπει αρχικά να δημιουργήσει λογαριασμό στην υπηρεσία και έπειτα μπορεί να ξεκινήσει την μεταφόρτωση των αρχείων που θέλει, στο νέφος της υπηρεσίας είτε με τη χρήση του browser είτε με τη χρήση της εφαρμογής της υπηρεσίας. Συνηθίζεται όλες οι υπηρεσίες να παρέχουν εφαρμογές για όλες της πλατφόρμες (Microsoft Windows, Mac OS, Linux, iOS, Android, Windows Phone).

Η πρόσβαση στην υπηρεσία γίνεται με χρήση των διαπιστευτηρίων του (credentials), που έχει επιλέξει κατά την εγγραφή του στην υπηρεσία. Έτσι κάθε φορά μπορεί να απολαμβάνει πρόσβαση στις δυνατότητες του νέφους του παρόχου. Εκεί μπορεί να αποθηκεύσει τα αρχεία του, να δημιουργήσει τα backup του και να μοιραστεί τα αρχεία του με άλλους χρήστες.

Σε όλες τις υπηρεσίες νέφους τα δεδομένα των χρηστών προστατεύονται με κρυπτογράφηση, πράγμα που σημαίνει πως αυτόματα τίθενται άχρηστα στα χέρια τρίτων αν αυτά διαρρεύσουν αυτούσια από τους servers της εταιρίας που προσφέρει την υπηρεσία. Αναγκαία συνθήκη για να ισχύει το προηγούμενο είναι φυσικά η χρήση αδιάβλητης μεθόδου κρυπτογράφησης. Δεν συμβαίνει όμως το ίδιο και όταν ο επιτιθέμενος έχει στη διάθεσή του τα credentials του χρήστη. Είναι γνωστή η περίπτωση της διαρροής credentials για τέτοιου τύπου υπηρεσίες, με «δημοφιλέστερη» την περίπτωση διαρροής των στοιχείων των χρηστών του DropBox το 2012[2], όπου 68.000.000 κωδικοί (σε κρυπτογραφημένη μορφή) αναρτήθηκαν στο internet. Ακόμα μια σχετική επίθεση είχε δημοσιοποιηθεί τον Σεπτέμβριο του 2014, όπου φωτογραφίες επώνυμων, χρηστών συσκευών της Apple[2], διέρρευσαν στο internet. Σε ίδιου τύπου επιθέσεις έχει πέσει και η Sony λίγο νωρίτερα από τα 2 προηγούμενα περιστατικά, με ακόμα μεγαλύτερο αντίκτυπο εξαιτίας της επαναλαμβανόμενης φύσης των επιθέσεων στον ηλεκτρονικό κολοσσό, με την επίθεση στο Sony PlayStation Network το 2010 και στην Sony Online Entertainment το 2011.

Ο μεγάλος χρόνος αντίδρασης των χρηστών σε τέτοιους είδους περιστατικά, σε συνδυασμό με τη κακή ποιότητα κωδικών (εύκολοι και προβλέψιμοι, μικροί, ίδιοι κωδικοί σε πολλές υπηρεσίες κτλ.), μπορούν σε βάθος χρόνου να αποκαλύψουν τους κωδικούς των χρηστών.

Για να μειωθούν οι επιπτώσεις επιθέσεων, με γνωστά credentials, οι πάροχοι των υπηρεσιών δίνουν τη δυνατότητα στους χρήστες να ενεργοποιήσουν δυνατότητες ασφάλειας «δυο παραγόντων»(two-factor) ή «δύο βημάτων»(two-step) αυθεντικοποίησης. Η όλη διαδικασία αν και θεωρείται αρκετά αξιόπιστη που περιπλέκει την εμπειρία χρήσης και μερικές φορές αποτρέπει τον χρήστη από το να αυξήσει τα επίπεδα ασφάλειας των λογαριασμών του, λόγω των περισσότερων και επαναλαμβανόμενων ενδιάμεσων βημάτων που χρειάζονται για την αυθεντικοποίηση των χρηστών.

Επιπλέον οι μηχανισμοί κρυπτογράφησης που χρησιμοποιούνται στις υπηρεσίες νέφους για τα αποθηκευμένα δεδομένα (data at rest), δεν χρησιμοποιούν δυνατές κρυπτογραφήσεις για λόγους απόδοσης(efficiency), πράγμα πολύ λογικό για ένα επιχειρησιακό μοντέλο αλλά όχι κατανοητό από τους χρήστες οι οποίοι επιδιώκουν τη μέγιστη ασφάλεια των αρχείων τους από μάτια τρίτων. Περισσότερα θα δούμε στο κεφάλαιο 5.5.3.

Άρα είναι επιθυμητός κάθε επιπλέον μηχανισμός που μπορεί να εξασφαλίσει ακόμα μεγαλύτερο βαθμό εμπιστευτικότητας στα αρχεία μας, είτε backup είτε άλλων ευαίσθητων αρχείων.

### *1.4.2 Το πρόβλημα των διαπιστευτηρίων*

Η χρήση cloud storage services όπως την γνωρίζουμε έως σήμερα λειτουργεί στη λογική πως ο χρήστης ανεβάζει σε μια υπηρεσία το αρχείο του και ανά πάσα στιγμή μπορεί να κατεβάσει ένα αντίγραφο του στον υπολογιστή του. Η αυθεντικοποιησή του βασίζεται στην γνωστή και τετριμμένη λογική εισαγωγής ονόματος χρήστη και κωδικού στην πλατφόρμα του παρόχου της υπηρεσίας (αφορά την είσοδο στην υπηρεσία μέσω browser) των αρχείων εξαρτάται από ένα πλήθος παραμέτρων όπως:

- Η δυσκολία του κωδικού πρόσβασης στην υπηρεσία cloud storage
- Η ενεργοποίηση two-step ή two-factor authentication δυνατότητας (εφόσον αυτή παρέχεται από την υπηρεσία)
- Την ασφάλεια της βάσης δεδομένων των credentials των χρηστών της υπηρεσίας cloud
- Αν ο χρήστης έχει πέσει θύμα υποκλοπής των στοιχείων εισόδου του, στην εν λόγω υπηρεσία (phishing)
- Αν ο χρήστης έχει, εν αγνοία του, εγκατεστημένο λογισμικό υποκλοπής της εισόδου του πληκτρολογίου (key logger).
- Αν ο χρήστης δουλεύει σε μη μολυσμένο και καθαρό από ιούς σύστημα.
- Αν τα credentials του χρήστη έχουνε πέσει στην αντίληψη του επιτιθέμενου με άλλους τρόπους hacking (social engineering, social hacking, οπτική επαφή κατά την πληκτρολόγηση των κωδικών κτλ.) .

- Άλλες μεθόδους υποκλοπής των στοιχείων εισόδου του χρήστη(φυσική πρόσβαση σε φυσικό αρχείο/σημειωματάριο κωδικών).
- Υποκλοπή στοιχείων από φορητές συσκευές που έχουν υποστεί jailbreak.

Άρα αν ο επιτιθέμενος εξασφαλίσει τα διαπιστευτήρια ενός χρήστη αποκτά πρόσβαση και στα αρχεία που έχει αποθηκευμένα στο νέφος. Από τις παραπάνω αναφορές καταλαβαίνουμε πως είναι σχετικά εύκολο να συμβεί κάτι τέτοιο, αφού οι τρόποι είναι αρκετοί. Κατ΄επέκταση υπάρχει δυνατότητα βελτίωσης της ασφάλειας των αρχείων που αποθηκεύουμε στο νέφος.

## *1.5 Αντικείμενο διπλωματικής*

Αντικείμενο αυτής της διπλωματικής εργασίας είναι η παρουσίαση μεθόδου διασφάλισης των αρχείων που αποθηκεύονται στο νέφος. Η προσέγγιση που γίνεται πάνω σε αυτό το θέμα αποτελεί περισσότερο μια ερευνητική προσέγγιση στο κομμάτι της ασφάλειας των αρχείων από μη εξουσιοδοτημένο χρήση, παρά μια ολοκληρωμένη πρόταση ασφάλειας. Για αυτό το σκοπό έχουμε συντάξει επιπλέον κεφάλαιο που προτείνει βελτιώσεις πάνω στο ίδιο αντικείμενο και το αναφέρουμε ξανά αργότερα μέσα στην εργασία. Γίνεται σαφές ωστόσο πως η βασική αρχή της μπορεί να λειτουργήσει με επιτυχία σε οικιακό, εταιρικό/επαγγελματικό περιβάλλον σαν ένας ακόμα παράγοντας διασφάλισης των δεδομένων.

### *1.5.1 Πρόβλημα προς διαχείριση*

Το πρόβλημα που θέλουμε να αντιμετωπίσουμε μπορούμε να το εκφράσουμε απλά με την παρακάτω διατύπωση:

"Θέλουμε οι χρήστες που ανεβάζουν αρχεία στο νέφος, να νοιώθουν σίγουροι πως:

1. Τα δεδομένα τους δεν μπορούν να διαβαστούν από άλλους ακόμα και αν πέσουν στα χέρια τρίτων.
2. Είναι οι μοναδικοί που έχουν τη δυνατότητα να διαβάσουν τα αρχεία τους.
3. Τα δεδομένα τους να είναι πάντα διαθέσιμα σε εκείνους.
4. Αν τα δεδομένα τους αλλοιωθούν θα το γνωρίζουν.
5. Τα δεδομένα τους είναι ασφαλή από αστοχίες υλικού και φυσικές καταστροφές.
6. Αποδοτικότερη χρήση των πόρων που διαθέτει στους χρήστες της η υπηρεσία νέφους (αποθηκευτικός χώρος).
7. Όλα τα παραπάνω να συμβαίνουν ταυτόχρονα.

Με άλλα λόγια αποδίδουμε λύση στην ενίσχυση της εμπιστευτικότητας των υπηρεσιών νέφους, ενισχύουμε έμμεσα την εμπιστοσύνη των χρηστών στις υπηρεσίες νέφους και χρησιμοποιούμε πιο αποδοτικά τους πόρους που αυτές παρέχουν.

Η χρήση των credentials για την είσοδο σε υπηρεσίες είναι, όπως είδαμε σε προηγούμενη ενότητα, θα μπορούσε να παρακαμφθεί από κάποιον αφοσιωμένο επιτιθέμενο. Προς το παρόν ο μοναδικός τρόπος με τον οποίο η αποτρέπεται η επιτυχημένη χρήση των σωστών credentials από τρίτους είναι η διαδικασία two-factor/two-step verification, για όσους χρήστες κάνουν χρήση αυτού του τρόπου εισόδου. Βέβαια μπορούμε εύκολα να φανταστούμε ένα σενάριο επίθεσης

που η πρόσβαση στα αρχεία του θύματος γίνεται και με χρήση αυθεντικοποίησης two-factor/two-step verification. Το σενάριο αυτό περιλαμβάνει την υποκλοπή των SMS με το επιπλέον συνθηματικό εισόδου που χρειάζεται η υπηρεσία νέφους ή στο πιο εύκολο σενάριο, την φυσική κλοπή της συσκευής τηλεφώνου του θύματος. Όποιο και αν είναι το μονοπάτι που θα επιλέξει ο επιτιθέμενος για να αποκτήσει είσοδο στα αρχεία του θύματος, η όλη διαδικασία μπορεί να διαρκέσει μερικά δευτερόλεπτα, αν γνωρίζουμε τα αρχικά credentials του θύματος, όσο χρόνο δηλαδή χρειαζόμαστε για να πάμε σε ένα γειτονικό δωμάτιο του σπιτιού μας ή ένα γειτονικό γραφείο συναδέλφου.

Αυξάνοντας την πολυπλοκότητα(πολλά νέφη και άρα πολλά credentials), για να φτάσει κάποιος επιτιθέμενος στο σκοπό του (να καταφέρει δηλαδή να φτάσει μέχρι τα αρχεία μας), αποθαρρύνουμε επιτιθέμενους και πολλαπλασιάζουμε χρόνο και κόπο, μειώνοντας τις πιθανότητες για μια επιτυχή επίθεση.

Εκτός από την πολυπλοκότητα αυτή προτείνουμε τρόπο για να θωρακίσουμε τα δεδομένα μας από τρίτους. Το καταφέρνουμε με την υλοποίηση της κρυπτογράφησης στα αρχεία που θα αποδώσουμε στις υπηρεσίες νέφους.

### *1.5.2 Μια κατανεμημένη προσέγγιση*

Στα σενάρια που αναφέραμε μέχρι τώρα θέτουμε ως βασική παράμετρο την ύπαρξη ενός λογαριασμού νέφους στον οποίο οι χρήστες ανεβάζουν τα αρχεία τους και οι επιτιθέμενοι με τη σειρά τους προσπαθούν να εκμαιεύσουν αυτούς τους κωδικούς ή/και να τους παρακάμψουν με άλλο τρόπο, όπως επίσης η άμεση πρόσβαση στα αρχεία από μη εξουσιοδοτημένους χρήστες που έχουνε πρόσβαση στα αρχεία του νέφους μέσω των φυσικών εγκαταστάσεων των υπηρεσιών (π.χ. εργαζόμενοι στις εταιρίες που παρέχουν τις υπηρεσίες νέφους, οι λεγόμενοι abuse operators), εξαιτίας αδύναμης κρυπτογράφησης.

Στην παρούσα εργασία προτείνουμε μια προσέγγιση που μπορεί να χρησιμοποιηθεί σαν επιπλέον επίπεδο διασφάλισης των δεδομένων που ανεβαίνουν στο νέφος υπηρεσιών αποθήκευσης αρχείων, κάνοντας χρήση πολλαπλών τέτοιων υπηρεσιών. Σκοπεύουμε στην προσαύξηση της ασφάλειας των αρχείων αυτών κατά 2 παράγοντες, πέραν από αυτούς που ήδη μπορεί να χρησιμοποιούνται από τις υπηρεσίες νέφους ή/και τους χρήστες.

Χωρίζοντας το αρχείο σε πολλά μικρότερα κομμάτια και αποθηκεύοντάς τα κομμάτια αυτά σε διαφορετικές υπηρεσίες (ή σε διαφορετικούς λογαριασμούς της ίδιας υπηρεσίας), αυξάνουμε την πολυπλοκότητα της επίθεσης πολλές φορές και με πολλούς τρόπους ταυτόχρονα. Αν μάλιστα κρυπτογραφήσουμε τα κομμάτια αυτά ή το αρχικό αρχείο μας, πριν αυτό/αυτά φύγουν από τον υπολογιστή μας προς το νέφος, τότε έχουμε μια δομή που δύσκολα μπορεί να φανερώσει τα αρχεία μας σε τρίτους, χωρίς τη θέλησή μας.

- Ο επιτιθέμενος θα πρέπει να γνωρίζει όλους τους κωδικούς πρόσβασης για κάθε λογαριασμό κάθε υπηρεσίας.
- Ο επιτιθέμενος θα πρέπει να έχει εξασφαλίσει το κλειδί με το οποίο έχουμε κρυπτογραφήσει το αρχείο μας πριν αυτό φύγει από το σύστημα μας.
- Ο τρόπος με τον οποίο πρέπει να ενωθούν τα κομμάτια του αρχείου είναι άγνωστος και ο επιτιθέμενος πρέπει να τον ανακαλύψει.

- Όσο πιο πολλά είναι τα κομμάτια στα οποία χωρίσαμε το αρχικό μας αρχείο, τόσο πιο πολλούς συνδυασμούς πρέπει να κάνει ο επιτιθέμενος, ώστε τελικά να πάρει στα χέρια του ένα κρυπτογραφημένο αρχείο.

Κατά πάσα πιθανότητα σε αυτό το σημείο ο επιτιθέμενος έχει καταλάβει πως ο χρόνος που θα χρειαστεί ώστε να ενώσει σωστά τα τεμάχια και να παραβιάσει τελικά την κρυπτογράφηση, ίσως να μην μεταφράζεται σε κέρδος πληροφορίας που να δικαιολογεί την όλη επίθεση.

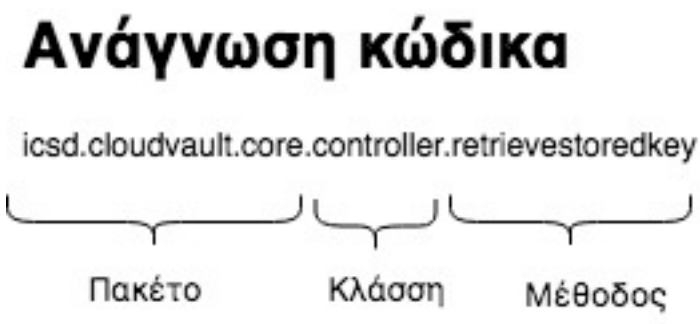
Σε αυτό το σημείο βάζουμε μια ακόμη παράμετρο, ώστε να κάνουμε την περίπτωση της παραβίασης της εμπιστευτικότητας, αδύνατη. Εφόσον τα τεμάχια που αποτελούν τα αρχικό αρχείο κρυπτογραφηθούν πριν αποδοθούν στα νέφη, τότε οποιαδήποτε προσπάθεια να ανακτηθεί το αρχείο καταλήγει σε μια πολύ χρονοβόρα διαδικασία και αποτρέπει τον επιτιθέμενο πλήρως.

Φτάνουμε όμως στο σημείο που πρέπει να εξετάσουμε τη περίπτωση χρήσης από τους καθημερινούς χρήστες, όχι μελετητές, ούτε έμπειρους χρήστες υπολογιστών. Περιγράφουμε στα κεφάλαια της υλοποίησης και μοντελοποίησης (κεφάλαιο 5) τον τρόπο με τον οποίο στην υλοποίηση μας αποκρύπτουμε από τον χρήστη όλες τις δομές που δεν τον αφορούν και παρεμβάλουμε μεταξύ αυτού και των νεφών, ως μοναδικό overhead, το γραφικό περιβάλλον της εφαρμογής μας που αναλαμβάνει να διευθετήσει το σύνολο των ενεργειών και να εκτελέσει εκ μέρους του όλη την πολύπλοκη δομή ενεργειών.

## *1.6 Πρακτικές οδηγίες μελέτης της εργασίας, του κώδικα και εκτέλεσης.*

### *1.6.1 Γενικές οδηγίες*

Εφόσον η παρούσα εργασία συνοδεύεται από πηγαίο κώδικα[9], είναι στη διακριτική ευχέρεια του αναγνώστη να ακολουθήσει τα πιο τεχνικά κομμάτια της από τον κώδικα Java. Για την ευκολία της συγγραφής και για λόγους περιήγησης στη δομή του κώδικα επιλέγουμε να αναφερόμαστε στα σημεία του κώδικα όπου αυτό είναι απαραίτητο με την εξής μορφή: "ΟνομαΠακέτου.ΟνομαΚλάσης", π.χ. icsd.cloudvault.core.generateCheckSums. Αν υπάρχει επιπλέον ανάγκη να υποδείξουμε συγκεκριμένη γραμμή κώδικα αναφερόμαστε στη γραμμή αυτή με το "#L" και δίπλα αναφέρουμε τον αριθμό της γραμμής, π.χ. #L53, για τη γραμμή 53.



### *1.6.2 Οδηγίες εκτέλεσης της εφαρμογής*

Για την εκτέλεση της εφαρμογής θα χρειαστεί να κατεβάσουμε τον κώδικα και να μελετήσουμε τα προ απαιτούμενα που περιγράφονται στο αρχείο /SecuringFilesInTheCloud/info/notes.txt . Σε παρακάτω κεφάλαιο (Κεφάλαιο 6), δίνουμε τον οδηγό χρήσης της εφαρμογής.

## *1.7 Δομή της διπλωματικής*

Στο πρώτο κεφάλαιο τοποθετούμε τη βάση για να στηρίξουμε την είσοδο του νέφους στην ζωή μας μέσα από μια πορεία σε απειλές που το νέφος άμβλυνε με τη ύπαρξη του ως μέσο αποθήκευσης . Εξηγούμε ποια χαρακτηριστικά της φύσης του αποδίδουν ασφάλεια στα δεδομένα μας ως έχει, αποσαφηνίζουμε ποια προβλήματα δημιουργούνται από τη χρήση σου για να καταλήξουμε στην περιγραφή της πρότασης μας. Η εργασία συνοδεύεται από υλοποίηση κώδικα και έχουμε συμπεριλάβει και οδηγίες για τον μελετητή που θα θελήσει να εκτελέσει την εφαρμογή ή να ακολουθήσει τις αναφορές σε κώδικα στα παρακάτω κεφάλαια.

Το δεύτερο κεφάλαιο παρουσιάζει εν τάχει άλλες μελέτες που σχετίζονται με το αντικείμενο της διπλωματικής εργασίας.

Το τρίτο κεφάλαιο κάνει μια περιγραφή του υπολογιστικού νέφους όπως αυτό αποδίδεται σήμερα στην εμπορική του μορφή. Περιέχει μια αναφορά στο Cloud Computing, τι είναι, ποιες ανάγκες μας οδήγησαν στην υλοποίηση και του, σε τι μας βοηθάει, ποιά τα πλεονεκτήματα του, αναφορά στο Cloud Stack και αναφορές στα επιμέρους κομμάτια του Cloud Stack (IaaS, PaaS, SaaS) για να καλύψουμε και να οδηγηθούμε λογικά στην υπηρεσία Cloud Storage την χρήση της οποίας θέλουμε να θωρακίσουμε με την πρόταση της εργασίας.

Το τέταρτο κεφάλαιο κάνει μια γενική επισκόπηση της υλοποίησης μας.

Το πέμπτο κεφάλαιο περιλαμβάνει την ανάλυση της υλοποίησης. Αναλύουμε τι κάνουμε στον κώδικα της εφαρμογής μας.

Στο έκτο κεφάλαιο προτείνουμε βελτιωτικές προσθήκες στην παρούσα υλοποίηση και μελέτες για την υλοποίηση νέων χαρακτηριστικών που θα αναβαθμίσουν την προσέγγιση μας.

Το έβδομο κεφάλαιο περιέχει τις βιβλιογραφικές αναφορές μου χρησιμοποιήθηκαν κατά τη συγγραφή της εργασίας.

# 2 *Σχετικές έρευνες*

## *2.1 Βιβλιογραφική επισκόπηση*

Η ραγδαία ενασχόληση του κοινού (οικιακού, εμπορικού) με το νέφος δεν μπορεί να αφήσει τους μελετητές της πληροφορικής αλλά και τους σκεπτικιστές του κλάδου αυτού της επιστήμης χωρίς άποψη επ᾽ αυτού. Φυσικά η επιστήμη και στην περίπτωση του νέφους έχει να προτείνει βελτιώσεις και να εφαρμόσει λογικές. Στη βιβλιογραφία όμως, της σχετική με την αποθήκευσης στο νέφος, έχουμε μοντέλα που σταματάνε όμως σε απλό επίπεδο αποτυγχάνοντας να συνδυάσουν περισσότερες από μια μεθόδους διασφάλισης των δεδομένων. Ενώ όλες οι εργασίες λοιπόν προτείνουν λύση δεν εντοπίσαμε κάποια που προσπαθεί με συνδυαστικό τρόπο να αυξήσει της ασφάλεια των δεδομένων πριν την παράδοσή τους στο νέφος.

Στο [3] οι συγγραφείς έχουν στηρίξει την προσπάθεια τους στην χαμηλή κρυπτογράφηση των δεδομένων από το ίδιο το νέφος ενώ ταυτόχρονα προτείνουν τον τεμαχισμό των αρχείων με σκοπό να αποδοθούν σε πολλαπλούς παρόχους και έτσι να αυξήσουν την ασφάλεια των χρηστών. Βασιζόμενοι μόνο στον τεμαχισμό των αρχείων διατηρείται έτσι η αδύναμη κρυπτογράφηση των παρόχων, ενώ αφήνει εκτεθειμένα τα δεδομένα στην δυνατότητα του επιτιθέμενου να αποκτήσει πρόσβαση στο αρχείο αν έχει στη διάθεση του πρόσβαση σε όλους τους λογαριασμούς του χρήστη.

Στο [4] γίνεται μια προσπάθεια να αυξηθεί η ασφάλεια των τεμαχίων που ανεβαίνουν στο νέφος, με την επιπλέον κρυπτογράφηση των τεμαχίων του αρχείου που αποστέλνονται στο νέφος. Αλλά δεν γίνεται πρόβλεψη για την διασφάλιση του κλειδιού σε τοπικό επίπεδο, κάτι που επιλύει η προσέγγιση που μας απασχολεί στην παρούσα εργασία.

# 3 Το Υπολογιστικό Νέφος

## 3.1 Το υπολογιστικό νέφος – Cloud computing

Το υπολογιστικό νέφος καλύπτει τις ανάγκες της απομακρυσμένης χρήση υπολογιστικών πόρων. Οι πόροι αυτοί μπορεί να είναι πόροι υποδομής (επεξεργαστική ισχύς, πόροι δεδομένων, πόροι αποθηκευτικού χώρου ή ένας συνδυασμός όλων των παραπάνω), πόροι πλατφόρμας (πλατφόρμες στις οποίες μπορούν να γίνουν deploy εφαρμογές) και πόροι λογισμικού(πρόσβαση σε λογισμικό).

Εδώ να υπενθυμίσουμε σε αυτό το σημείο πως το νέφος δεν είναι απλά ένα αφηρημένο σημείο αποθήκευσης πληροφορίας, ιστοσελίδων, βάσεων δεδομένων κτλ., αλλά ένα σύνολο υπολογιστικών πόρων που περιλαμβάνουν όλο το εύρος των πόρων του . Ι.Τ. , από αποθήκευσης , δίκτυα, συστήματα, βάσεις, διάφορων τύπων διακομιστές κ.α. Η φύση της εργασίας και το αντικείμενο της έχει σαφή ροπή προς την αποθηκευτική του ιδιότητα, αλλά ούτε ο συγγραφέας ούτε ο αναγνώστης πρέπει να παραβλέπει την συνολική υπόσταση του νέφους. Με άλλα λόγια μπορεί στην παρούσα εργασία να «κοιτάμε το δένδρο», σχεδόν αποκλειστικά, αλλά δεν θα πρέπει να αφήσουμε εκτός το «δάσος».

### 3.1.1 Η εμπορική εφαρμογή του

Η εμπορική διάθεση των πόρων αυτών είναι βασικό επιχειρηματικό μοντέλο μεγάλων οργανισμών και προκύπτει με αυτόν τον τρόπο και η αντίστοιχη ονοματολογία του Cloud Stack που αποτελείται από την παροχή της υποδομής ως υπηρεσία (Iaas), της πλατφόρμας ως υπηρεσία(PaaS), του λογισμικού ως υπηρεσία (SaaS).

Για να περιγράψουμε αυτή την διαφοροποίηση των πόρων στο νέφος χρησιμοποιούμε τον όρο Cloud Stack.

Οι διαφορετικές κατηγορίες πόρων χρησιμοποιούνται από διακριτές ομάδες χρηστών με διαφορετικό τρόπο η καθεμία και τελικά μπορούμε να διαχωρίσουμε το νέφος σε τρείς (3) κύριες κατηγορίες βάσει του πόρου τον οποίο κάθε μια χρησιμοποιεί: Το λογισμικό, την πλατφόρμα και την υποδομή.

## *3.2 Cloud Stack*

### *3.2.1 Infrastracture as a service – Η υποδομή ως υπηρεσία (IaaS)*

Στην περίπτωση του IaaS ο χρήστης/πελάτης έχει στη διάθεσή του πόρους που παραδοσιακά συναντά σε εγκαταστάσεις data centers, όπως διακομιστές, δικτυακό εξοπλισμό, εξοπλισμό storage κ.α. και μάλιστα πολλές από αυτές σε εικονικό επίπεδο (Virual Machine, VM). Συνοδευτικά με αυτές τις υπηρεσίες τοποθετούνται και διαχειριστικά εργαλεία και εργαλεία monitoring των συστημάτων που προαναφέρθηκαν. Από τη φύση του αυτό το κομμάτι υπηρεσιών αφορά κυρίως οργανισμούς που δεν έχουν δική τους υποδομή ή δεν θέλουν να επενδύσουν σε δική τους υποδομή. Οι λόγοι είναι συνήθως:

- Η συνεχείς εξέλιξη του hardware καθιστά πολλές φορές ασύμφορη την αγορά του, ειδικά σε μικρές και μεσαίες επιχειρήσεις. Η λύση του IaaS επιβάλει μόνο την αγορά τεχνογνωσίας.
- Μεγάλοι οργανισμοί χρειάζονται πολλές φορές data centers σε πολλές και απομακρυσμένες μεταξύ τους περιοχές. Τα data centers παρόχων IaaS μπορούν να καλύψουν τέτοιες ανάγκες, χωρίς να χρειάζεται ο εκάστοτε οργανισμός να επενδύσει σε δικά του πολυδάπανα data centers.
- Η επεκτασιμότητα της υποδομής που προσφέρουν οι πάροχοι IaaS είναι δυνητικά απεριόριστη. Αυτό σημαίνει πως ένας χρήστης/πελάτης μπορεί να αυξήσει την υποδομή του στον βαθμό που την χρειάζεται αγοράζοντας/νοικιάζοντας όσους πόρους επιθυμεί. Δίνει δηλαδή τη δυνατότητα να αυξομειώνει την υποδομή του κατά βούληση, χωρίς να δεσμεύεται από την αγορά εξοπλισμού και συμβολαίων μακροχρόνιας μίσθωσης (γνωστή και ως χρονομίσθωση ή Leasing).

### *3.2.2 Platform as a service – Η πλατφόρμα ως υπηρεσία (PaaS)*

Η αναφορά στην πλατφόρμα αφορά την βάση πάνω στην οποία προγραμματιστές θα στηριχτούν ώστε να αναπτύξουν το λογισμικό τους, χωρίς να χρειάζεται να ασχοληθούν και να εξατομικεύσουν το υποκείμενο υλισμικό και το κατάλληλο λειτουργικό σύστημα που θα υποδεχθεί τις εφαρμογές τους. Βελτιώνει:

- Τους χρόνους ανάπτυξης και παράδοσης του software
- Διευκολύνει την παράδοση του λογισμικού στον πελάτη
- Αποσοβεί τον κίνδυνο της λάθους παραμετροποίησης του περιβάλλοντος στο οποίο θα εγκατασταθεί η εφαρμογή
- Απομακρύνει σφάλματα που μπορεί να προκύψουν από την κακή παραμετροποίηση του περιβάλλοντος εγκατάστασης

### *3.2.3 Software as a Service – Το Λογισμικό ως υπηρεσία (SaaS)*

Αποτελεί την κορυφή της πυραμίδας του Cloud Stack. Σε αντίθεση με τα προηγούμενα 2 επίπεδα (layers) της στοίβας (stack) απευθύνεται σε μεγάλο κοινό γιατί είναι αυτό που «κουβαλά» την υπηρεσία υψηλότερου επιπέδου. Τα θετικά του έχουν αναλυθεί και περιγραφεί εκτεταμένα στη βιβλιογραφία [5]. Από αυτό το επίπεδο οι χρήστες απολαμβάνουν τις διαδεδομένες και δημοφιλείς υπηρεσίες νέφους, όπως είναι το Cloud Storage που απασχολεί ετούτη την εργασία.

## *3.3 Υπηρεσίες Αποθήκευσης δεδομένων στο νέφος - Cloud Storage Services*

### *3.3.1 Η διαδρομή προς την ανάγκη για αποθήκευσης δεδομένων στο νέφος.*

Με τη διάδοση των πολυμεσικών(multimedia) εφαρμογών και των αρχείων που τις συνοδεύουν, από τη δεκαετία του 1990 και έπειτα, γιγαντώθηκαν και αποθηκευτικές ανάγκες των χρηστών. Μουσική, εικόνες, βίντεο με τα πολυάριθμα φορμά (format) τους, με κάποια από αυτά να είναι «μη απωλεστικά» (lossless, όπως .wav, .flac) αλλά και τα απωλεστικά (π.χ. .mp3, .mp4, .avi), γέμισαν τα αποθηκευτικά μέσα των χρηστών οι οποίοι κατέφευγαν σε μόνιμες λύσεις (σκληροί δίσκοι) και ημιμόνιμες λύσεις (ψηφιακοί δίσκοι CD). Ταυτόχρονα είχαμε την έκρηξη νέων καναλιών διάθεσης των πολυμεσικών αρχείων με τα peer-to-peer δίκτυα. Σε αυτό το σημείο να σημειώσουμε πως η αναφορά στα πολυμεσικά αρχεία συμβαίνει να είναι και η οδηγός αλλαγή που οδήγησε στην αλματώδη αύξηση των αναγκών αποθήκευσης , τουλάχιστον για τον οικιακό χρήστη, ενώ γύρο από τα πολυμεσικά αρχεία χτίστηκαν οι σημερινοί τεχνολογικοί κολοσσοί του τομέα (βλ. YouTube) στα μέσα της δεκαετίας 2000. Και φτάνουμε έτσι στο σημείο που οι χρήστες εκμεταλλεύονται το όλο και μειούμενο κόστος των μαγνητικών αποθηκευτικών μέσων, όπως οι σκληροί δίσκοι.
Εκτός από τους οικιακούς χρήστες όμως οι κύρια εμπορική δύναμη στον κλάδο είναι οι εταιρίες οι οποίες έχουνε και αυτές πολύ μεγάλες αποθηκευτικές ανάγκες, όχι τόσο σε πολυμεσικό επίπεδο, αλλά εφόσον δεν θέλουν να χάσουν το «τραίνο» της νέας εποχής την πληροφορικής, μεταφέρουν τις δραστηριότητες τους στον ψηφιακό κόσμο. Επίσης η διαδικασία της ψηφιοποίησες της δραστηριότητας φορέων δημοσίου συμφέροντος, εκπαιδευτικού ενδιαφέροντος κτλ. δημιουργεί επιπλέον αποθηκευτικές ανάγκες.

### *3.3.2 Η αποκεντροποίηση της πληροφορίας*

Στη διαδικασία της συγκέντρωσης όμως της πληροφορίας σε τοπικά μέσα αποθήκευσης , οικιακοί χρήστες αλλά και οργανισμοί διακινδυνεύσανε (στις περιπτώσεις που δεν υπήρχανε back ups) τα πολύτιμα δεδομένα τους. Επίσης η εξέλιξη των φορητών συσκευών και η ταυτόχρονη μείωση του μεγέθους τους, δημιούργησαν την ανάγκη τα δεδομένα να είναι προσβάσιμα κάθε στιγμή, γεγονός που συνέδραμε και η ταυτόχρονη εξέλιξη των τηλεπικοινωνιών με τα νέα δίκτυα τρίτης και τέταρτης γενιάς (3G και 4G αντίστοιχα).

Έπρεπε λοιπόν τα δεδομένα να προωθούνται στους χρήστες τους ακόμα και εν κινήσει, αλλά η αξιοπιστία και η δυνατότητες των καναλιών διάθεσης τους από εταιρικούς και οικιακούς

εξοπλισμούς είχε συγκεκριμένη δυναμική, η οποία ειδικά στα εταιρικά περιβάλλοντα είναι σοφότερο να διατίθεται για τους λειτουργικούς σκοπούς των εταιριών και των οργανισμών. Επιπλέον η διάχυση του διαδικτύου στις ανθρώπινες κοινωνίες σε όλη την υδρόγειο προκαλεί και την άνθιση του εμπορίου αγαθών και υπηρεσιών μέσω του διαδικτύου. Αποτέλεσμα της είναι και η παράταξη/ανάπτυξη κέντρων δεδομένων σε όλη την έκταση του πολιτισμένου κόσμου, που επιφορτίζονται με την αποθήκευσης δεδομένων για διαφορετικές εκδοχές (instances) σελίδων και επιχειρηματικών αναγκών που βασίζονται στο διαδίκτυο, όπως χρηματοοικονομικές συναλλαγές, χρηματιστηριακές ανάγκες, στοιχηματικές επενδύσεις, εμπορικές ανάγκες κ.α.

Αναγνωρίζεται λοιπόν και η ανάγκη να υπάρχει ο ίδιος όγκος πληροφορίας σε πολλαπλά σημεία στον πλανήτη, ώστε να εξυπηρετείται η επιχειρηματική ανάγκη αλλά και να υπάρχει και δυνατότητα αλληλοϋποστήριξης των συστημάτων από αστοχίες δικτύου η υλικού.

### *3.3.3 Αναγνώριση του νέφους ως λύση*

Η λογική συνέχεια της δημιουργίας ιδιωτικών data centers για έναν οργανισμό, είναι το φυσικό επόμενο της επιχειρησιακής πλεύσης. Είναι όμως απαγορευτικό για μικρούς οργανισμούς να δημιουργήσουν και να συντηρήσουν κέντρα δεδομένων σε κάθε γωνιά της γης λόγω του κόστους που κάτι τέτοιο υπονοεί. Ενώ οι μεγάλοι οργανισμοί μπορούσαν να διαθέσουν αυτούς τους πόρους για μια τέτοια επένδυση, οι μικρότεροι οργανισμοί έπρεπε να βρουν άλλους τρόπους να επιτύχουν τους αναπτυξιακούς τους στόχους.

### *3.3.4 Εμπορική διάθεση*

Έτσι εισήλθαμε στο 2000, με την Amazon[6] να παρουσιάζει πρώτη μια πλατφόρμα ενοικίασης υπολογιστικών πόρων (Elastic Compute Cloud), με την Google[7] να ακολουθεί με σαφέστατη χρονική καθυστέρηση (2006) ανακοινώνοντας της δική της πλατφόρμα που υποστήριζε deployment διαδικτυακών εφαρμογών (web applications) με όνομα Google App Engine, και με ουραγό τη Microsoft το 2010 να παρουσιάζει τη δική της πλατφόρμα (Azure)[8].

# 4 *Η προσέγγιση σε λεπτομέρεια*

Σε αυτή την ενότητα παρουσιάζουμε την λύση που προτείνουμε και αναλύουμε την δομή της. Μέσα από διάγραμμα περιπτώσεων χρήσης (data flow diagrams) αποδίδουμε τη διαδρομή που ακολουθούν τα δεδομένα από το τοπικό σύστημα προς το νέφος και τις ενδιάμεσες τροποποιήσεις στις οποίες υπόκεινται σε όλη αυτή τη διαδρομή.

## *4.1 Λειτουργία εφαρμογής*

Η βασική λειτουργία που επιτελούμε είναι η μεταφόρτωση αρχείων από και προς το νέφος. Η διαφορά μας εναπόκειται στο γεγονός πως θέλουμε να το κάνουμε με τρόπο τέτοιο που δεν θα είναι εύκολο από τρίτους να αποκτήσουν πρόσβαση στα δεδομένα μας. Η προσέγγιση μας καθιστά τα δεδομένα που μεταφορτώνονται στο νέφος μη αξιοποιήσιμα και ο μόνος τρόπος να αποκτήσει ένας επιτιθέμενος την αποθηκευμένη πληροφορία στην ολότητα της, είναι να έχει στην κατοχή του:

- Το master κωδικό της εφαρμογής
- Τη δομή αποθήκευσης της εφαρμογής(περιέχει τα metadata της εφαρμογής)
- Τους κωδικούς από όλα τις υπηρεσίες νέφους.

Σε παρακάτω κεφάλαιο περιλαμβάνουμε και τον οδηγό χρήσης της εφαρμογής.

## *4.2 Πλεονεκτήματα της προσέγγισής μας*

Απαριθμώντας τα χαρακτηριστικά της προσέγγισης περιγράφουμε στα παρακάτω και τα πλεονεκτήματα που απορρέουν από εφαρμογή της. Πιο συγκεκριμένα:

- Η διαίρεση των αρχείων σε πολλαπλά τεμάχια κάνει δύσκολη την φυσική πρόσβαση του επιτιθέμενου στο σύνολο των κομματιών.
- Ο τρόπος με τον οποίο αυτά ονοματοδοτούνται στο νέφος κάνει τον συσχετισμό των τεμαχίων του αρχείου αδύνατο εκ πρώτης όψης.

- Η κρυπτογράφηση των τεμαχίων πριν την αποστολή στο νέφος καθιστά την επιτυχή επανασύνδεση από τον επιτιθέμενο, μια επίπονη διαδικασία. Το κόστος σε χρόνο είναι πολύ μεγάλο.
- Οι υπηρεσίες νεφών χρησιμοποιούνται πολύ πιο αποδοτικά, σε ότι αφορά τον διαθέσιμο τους χώρο.

## *4.3 Δυνατότητες εφαρμογής*

Όταν ο χρήστης αποφασίσει να μεταφορτώσει ένα αρχείο στο νέφος, το αρχείο:

1. Διαιρείται σε τρία (3) κομμάτια που μεταξύ τους έχουν ίδιο μέγεθος προσεγγιστικά.
2. Κάθε τεμάχιο από αυτά κρυπτογραφείται ξεχωριστά
3. Για κάθε κρυπτογραφημένο τεμάχιο παράγουμε το μοναδικό του hash.
4. Τα τεμάχια μεταφορτώνονται στο νέφος, έκαστο σε διαφορετικό πάροχο.
5. Τα τεμάχια έχουν τυχαία ονόματα και ο συσχετισμός τους δεν προκύπτει λογικά.
6. Τοπικά αποθηκεύουμε μια δομή(state), που περιγράφει το αρχικό αρχείο που τεμαχίστηκε, ποια είναι τα ονόματα των τεμαχίων, ποιο είναι το νέφος στο οποίο μεταφορτώθηκε έκαστο και ποια είναι τα hashes που αντιστοιχούν σε έκαστο, όπως και τα ονόματα των τεμαχίων.

Όταν ο χρήστης αποφασίσει να μεταφορτώσει ένα αρχείο από το νέφος, τότε:

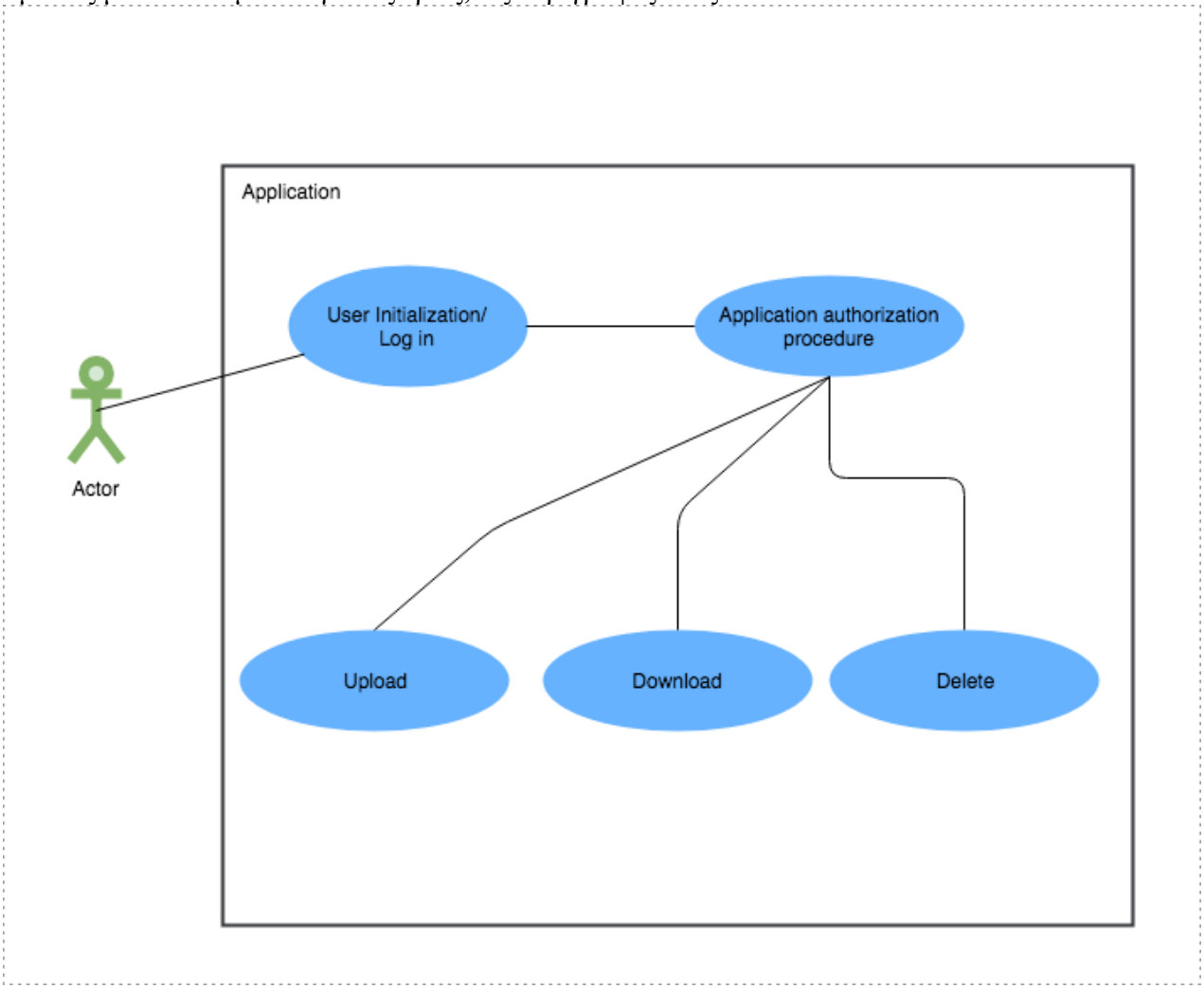
1. Επιλέγει το αρχείο που τον ενδιαφέρει
2. Ο αλγόριθμος επιλέγει από τη δομή τα ονόματα των τεμαχίων που δομούν το αρχικό (original) αρχείο, τις υπηρεσίες από τις οποίες θα πρέπει να τα ζητήσει και τα hashes των τεμαχίων έτσι όπως είχανε κρατηθεί στο state πριν αυτά μεταφορτωθούν στο νέφος.
3. Ξεκινάει η μεταφόρτωση του πρώτου αρχείου και μετά την ολοκλήρωση της, ελέγχεται το hash του τεμάχιού. Αν το τεμάχιο αποδώσει διαφορετικό hash από αυτό που έχει αποθηκευτεί στο state της εφαρμογής, τότε ο χρήστης λαμβάνει μήνυμα που τον ενημερώνει ποια υπηρεσία έχει αποδώσει αλλοιωμένο αρχείο. Έτσι εξετάζουμε το integrity των αρχείων που έχουμε μεταφορτώσει.
4. Εφόσον τα τεμάχια δεν έχουνε αλλοιωθεί, αποκρυπτογραφούνται με τη βοήθεια του state, περιλαμβάνει τις κατάλληλες πληροφορίες για τη διαδικασία αυτή.
5. Τα τεμάχια ενοποιούνται σε ένα αρχείο και πάλι με τη βοήθεια του state, που μας πληροφορεί για την σειρά με την οποία τα τεμάχια πρέπει να τοποθετηθούν.

Όταν ο χρήστης ζητάει την διαγραφή ενός αρχείου από το νέφος:

1. Επιλέγει το αρχείο που τον ενδιαφέρει.
2. Ο αλγόριθμος επιλέγει από τη δομή αποθήκευσης (state) τα απαραίτητα μεταδεδομένα που υποδεικνύουν από πιο νέφος και με πιο όνομα θα αναζητηθούν τα τεμάχια.
3. Ο αλγόριθμος αναζητά τα αρχεία στα νέφη.
4. Διαγράφει τα αρχεία από τα νέφη διαδοχικά
5. Διαγράφει τα μεταδεδομένα από την δομή αποθήκευσης .

## *4.4 Περιπτώσεις χρήσεις*

Στο παρακάτω διάγραμμα αποδίδουμε τις τρείς βασικές περιπτώσεις χρήσης της εφαρμογής, ενώ αμέσως μετά δίνουμε και για τις τρείς, τις περιγραφές τους.




Use case: Upload.
Actor: Χρήστης εφαρμογής.
Brief: Ο χρήστης κάνει το upload.
Precondition: Ο χρήστης έχει κάνει log in επιτυχώς στις υπηρεσίες μέσω της εφαρμογής.
Postcondition: Η εφαρμογή έχει εξουσιοδοτηθεί από τις υπηρεσίες επιτυχώς.
Trigger: Ανάγκη μεταφόρτωσης αρχείου προς τα νέφη.

Use case: Download.
Actor: Χρήστης εφαρμογής.
Brief: Ο χρήστης κάνει το upload.

Precondition: Ο χρήστης έχει κάνει log in επιτυχώς στις υπηρεσίες μέσω της εφαρμογής.
Postcondition: Η εφαρμογή έχει εξουσιοδοτηθεί από τις υπηρεσίες επιτυχώς.
Trigger: Ανάγκη ανάκτησης αρχείου από τα νέφη.

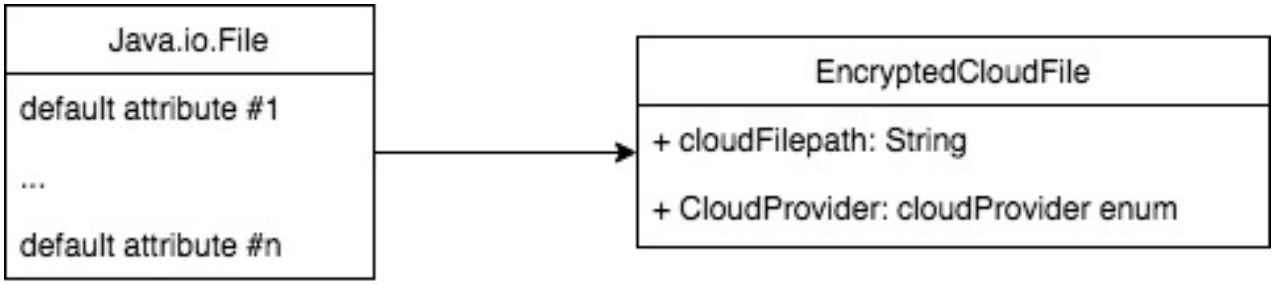
Use case: Delete.
Actor: Χρήστης εφαρμογής.
Brief: Ο χρήστης διαγράφει αρχείο.
Precondition: Ο χρήστης έχει κάνει log in επιτυχώς στις υπηρεσίες μέσω της εφαρμογής.
Postcondition: Η εφαρμογή έχει εξουσιοδοτηθεί από τις υπηρεσίες επιτυχώς.
Trigger: Ανάγκη διαγραφής αρχείου από τα νέφη.

# 5 *Θέματα υλοποίησης και μοντελοποίησης*

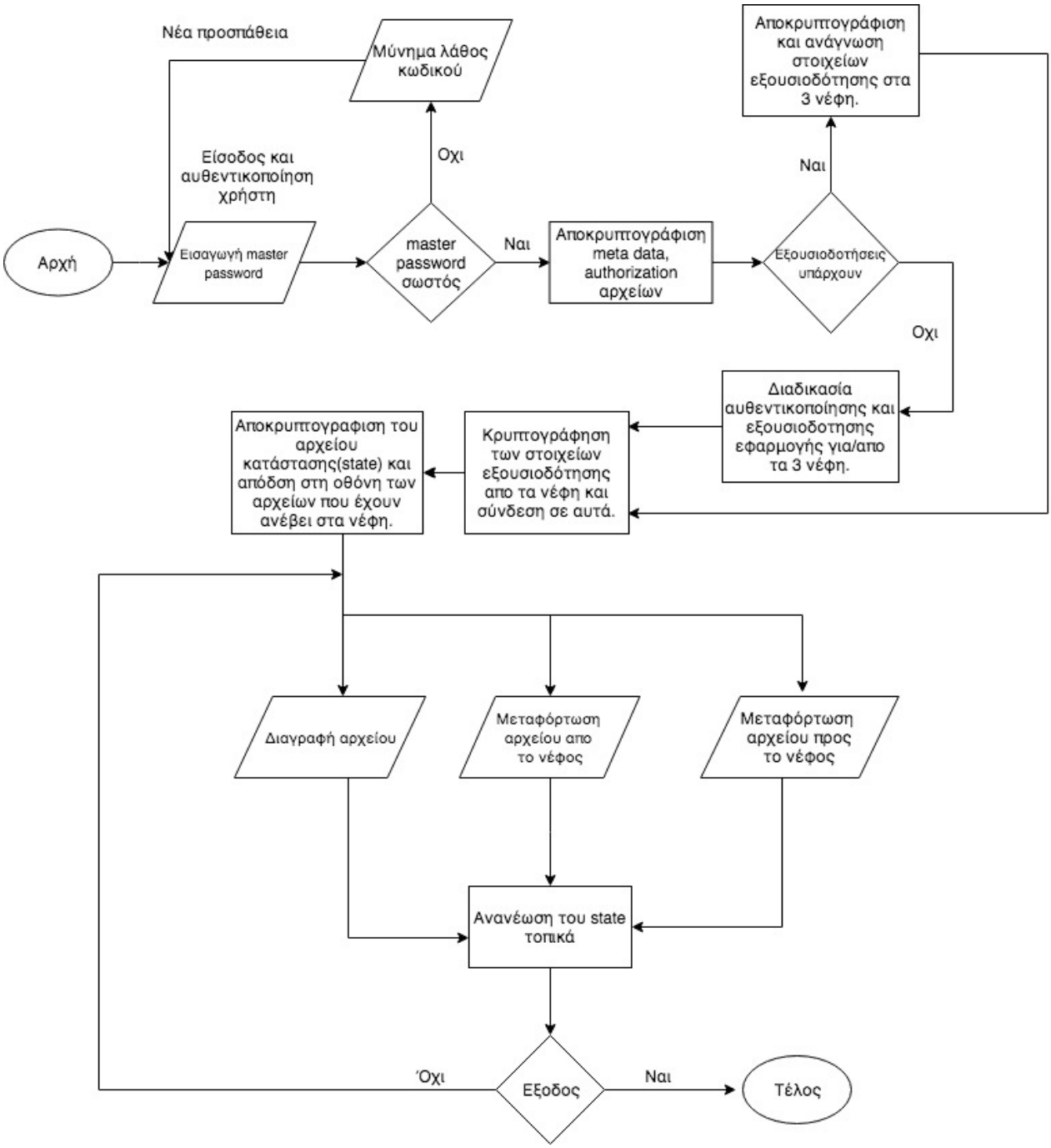
Έχοντας εξετάσει τη λειτουργία της δομής σε θεωρητικό επίπεδο προχωράμε στην παρουσίαση των συστατικών μερών της, των δομών που την ακολουθούν και την υποστηρίζουν, ενώ περιγράφουμε και τους λόγους που μας οδήγησαν στην επιλογή τους.

## *5.1 Αρχεία*

Ένα από τα σημεία που δόθηκε ιδιαίτερη προσοχή στην επιλογή της υλοποίησης είναι η δομή που αντιπροσωπεύει τα αρχεία. Ο λόγος είναι πως αποτελεί μια από τις σημαντικότερες υλοποιήσεις σε όλη την έκταση της εργασίας. Είναι η βασική οντότητα που αντιπροσωπεύει τα αρχεία. Στα `icsd.cloudvault.entities.EncryptedCloudFile` και `icsd.cloudvault.entities.SplittedFile` περιγράφονται οι συσχετίσεις των υλοποιήσεων που έχουμε στο κώδικα για τα επιπλέον χαρακτηριστικά που απαιτήθηκαν ώστε να περιλαμβάνεται σε αντικείμενα των κλάσεων αυτών τα απαραίτητα μεταδεδομένα.



## *5.2 Διάγραμμα ροής – Application flow*



## *5.3 Μέθοδος αποθήκευσης δεδομένων*

Σε αυτή ενότητα εξετάζουμε το ποιά αρχεία απασχολούν την δομή αποθήκευσης (state) της υλοποίησης.

### *5.3.1 Τοπική αποθήκευση*

Το ζήτημα της δομής αποθήκευσης των δεδομένων και μεταδεδομένων της εφαρμογής δημιούργησε πολλές φορές δεύτερες σκέψεις. Η επιλογή όμως τελικά έγινε με βάση την έκταση που θα λάμβανε η εργασία στο σύνολο της και με βάση την δυνατότητα να έχουμε στα χέρια μας ένα παραδοτέο που κάνει αυτό ακριβώς που περιγράφει η εργασία, χωρίς να είναι δύσκολο για τον μελετητή να εκτελέσει την εφαρμογή. Ενώ λοιπόν η συμβατική επιλογή θα ήτανε η χρήση μιας βάσης δεδομένων με το αντίστοιχο σχήμα για την αποθήκευσης των δεδομένων και των μεταδεδομένων, εδω έγινε χρήση τοπικών αρχείων για την διατήρηση αυτών των αρχείων. Τα βασικά αρχεία που χρησιμοποιούνται από την υλοποίηση είναι τα:

- `icsd.cloudvault.constants.constants` που αποθηκεύει παραμέτρους σχετικούς με τη λειτουργία της εφαρμογής, όπως οι διαδρομές και τα ονόματα άλλων σημαντικών για την εφαρμογή αρχείων, αλλά και αρχεία για το γραφικό της περιβάλλον. Στο ίδιο αρχείο θα εντοπίσουμε την πόρτα με την οποία αλληλοεπιδρά η εφαρμογή με κάθε υλοποίηση εφαρμογής πελάτη(client) που έχουμε αναπτύξει για κάθε μια από τις υπηρεσίες νέφους.
- `icsd.cloudvault.constants.config`
  Στο αρχείο αυτό θα εντοπίσουμε τα κλειδιά και τα μυστικά αναγνωριστικά που παρέχουν οι υπηρεσίες νέφους για την εξουσιοδότηση της εφαρμογής μας ώστε να έχει δυνατότητα να χρησιμοποιήσει τις υπηρεσίες τους. Επιπλέον από εδώ ενεργοποιούμε και απενεργοποιούμε τον μηχανισμό καταγραφής (logging) στην κονσόλα, πράγμα χρήσιμο για τον ερευνητή, αν θέλει να ακολουθήσει την ροή της εφαρμογής κατά την εκτέλεση της αλλά και για τον προγραμματιστή όταν χρειάζεται να κάνει εξάλειψη σφαλμάτων (debug), χωρίς τη χρήση των δυνατοτήτων που δίνει ένα σύγχρονο προγραμματιστικό περιβάλλον(IDE). Ο μηχανισμός αυτός ελέγχει την κατάσταση της μεταβλητής `LOGGING_SWITCH` και αποφασίζει αν θα εκτυπώσει η όχι μηνύματα στον χρήστη. Αυτή η λειτουργία για να ενεργοποιηθεί απαιτεί το compile της εφαρμογής.

## *5.4 Οργάνωση κλήσεων προς τις υπηρεσίες νέφους*

Για να μπορέσει να δομηθεί μια τέτοια πολύπλοκη δομή με πολλαπλές υπηρεσίες, χρειάστηκε να δημιουργηθεί μια αρχιτεκτονική δομή που θα επιτρέπει την κλήση των υπηρεσιών με τον ίδιο τρόπο. Με τη μελέτη των api εκάστης υπηρεσίας διαπιστώθηκε πως ενώ όλες έχουν το ίδιο αντικείμενο (μεταφόρτωση των δεδομένων από και προς το νέφος), ο κάθε κατασκευαστής λογισμικού έχει σαφείς διαφορές στον τρόπο που το επιτυγχάνει. Για να μπορέσουμε να δημιουργήσουμε μια διαχειρίσιμη δομή που λειτουργεί κατά το μέγιστο δυνατό τρόπο με όμοιο τρόπο για όλες τις υπηρεσίες, κρύψαμε τις κοινές λειτουργίες τους πίσω από μια διεπαφή (intreface). Στο `icsd.cloudvault.core.cloud.CloudInterface` βλέπουμε τις

μεθόδους που αποφασίσαμε να χρησιμοποιήσουμε στην υλοποίηση μας. Κατ΄επέκταση και όλες οι εφαρμογές πελάτη που υλοποιήσαμε:
`icsd.cloudvault.core.cloud.DropBoxClientService,`
`icsd.cloudvault.core.cloud.BoxClientService,`
`icsd.cloudvault.core.cloud.GoogleDriveClientService`
υλοποιούν με τη σειρά τους το `CloudInterface`.

## *5.5 Γραφικό περιβάλλον εφαρμογής*

Το γραφικό περιβάλλον(gui) έχει δομή, λειτουργία και λογική σε πολύ περιορισμένη έκταση. Προτιμούμε να φέρουμε την λογική της υλοποίησης στο πυρηνικό της κομμάτι(core), για να μπορέσει ο επόμενος ερευνητής να το αφαιρέσει με εύκολο τρόπο και στην θέση του να βάλει μια web υλοποίηση. Σε επίπεδο προτυποποίησης η παρούσα υλοποίηση του γραφικού κομματιού της εφαρμογής καλύπτει τις ανάγκες μας, δίνοντας μια περιγραφική και εύκολα κατανοητή εικόνα των κινήσεων που απαιτούνται από τον χρήστη για την ολοκλήρωση μια μεταφόρτωσης.

Με την παρούσα δομή ένας προγραμματιστής θα μπορούσε να μετατρέψει την υπάρχουσα υλοποίηση σε αρχιτεκτονική microservices, γράφοντας μια διαφορετική υλοποίηση για το front-end της εφαρμογής και δρομολογώντας τις κλήσεις στις μεθόδους από έναν ελάχιστα τροποποιημένο controller, που θα διαθέτει όλη την λογική και θα δρομολογεί τις κλήσεις στις εφαρμογές-πελάτες, όπως αυτές περιγράφονται από τις κλάσεις της ενότητας 5.2 .

## *5.6 Δομή αποθήκευσης μοναδικών αναγνωριστικών υπηρεσιών*

Για τη αποθήκευσης των μοναδικών αναγνωριστικών (tokens) που αποδίδονται στην εφαρμογή από τα νέφη, υλοποιούμε το αρχείο /data/auth. Εδώ τα δεδομένα μας κρυπτογραφούνται και μένουν ασφαλή σε τοπικό επίπεδο. Τα αρχεία αυτά ανοίγουν και κλείνουν άμεσα πριν και μετά τη χρήστη τους με παράλληλη αποκρυπτογράφηση και κρυπτογράφηση τους. Έτσι δεν εκτίθενται περισσότερο από το ελάχιστο χρονικό παράθυρο της διαδικασίας αυτής. Αυτό το διαπιστώνουμε από στο αρχείο icsd.cloudvault.utils.FunctionsToolkit.java στις #L45, #L93 (για την εγγραφή) με τις μεθόδους `saveCloudServicesTokensMap` και `saveCloudServiceToken,` στις #L139, L#161 για την ανάγνωση με τις μεθόδους `getServicesTokens` και `getServiceToken`.

## *5.7 Κρυπτογράφηση και αποθήκευσης δεδομένων*

### *5.7.1 Αποθήκευση κλειδιού εφαρμογής*

Το κλειδί της εφαρμογής (master password), αποτελεί ίσως την σημαντικότερη πτυχή της όλης διαδικασίας. Η εφαρμογή βασίζει όλη την κρυπτογράφηση σε αυτό το κλειδί. Όπως έχει αναφερθεί νωρίτερα, το κλειδί αυτό ποτέ δεν αποθηκεύεται αυτούσιο, αλλά το hash του κλειδιού είναι αυτό που εξετάζεται. Εξυπηρετούμε λοιπόν και την ασφαλή αποθήκευσης των στοιχείων της εφαρμογής.

### *5.7.2 Αποθήκευσης κλειδιών εξουσιοδότησης υπηρεσιών*

Σε συνέχεια της περιγραφής της παραγράφου 5.2 μελετώντας της μεθόδους που αναφέραμε σε αυτή θα εντοπίσουμε την κλήση των encrypt και decrypt μεθόδων που βρίσκονται στο αρχείο `icsd.cloudvault.utils.AdvnacedFileEncryption` και είναι υπεύθυνες για την ασφαλή κρυπτογράφηση και αποκρυπτογράφηση των αρχείων.

### *5.7.3 Κρυπτογράφηση από την υπηρεσία νέφους*

Αυτή το μέρος δεν απασχολεί τη δική μας υλοποίηση. Η κάθε υπηρεσία διαθέτει τις δικές τις δικλείδες ασφάλειας, τόσο κατά τη διαδικασία μεταφόρτωσης, όσο και για την αποθήκευσης [10],[11],[12]. Άλλωστε η εργασία αυτή υπονοεί κατά κάποιο τρόπο πως δεν υπάρχει εμπιστοσύνη προς τις υπηρεσίες νέφους και υιοθετεί επιπλέον κρυπτογράφηση στα τεμάχια των αρχείων που αποστέλλει σε αυτές.

### *5.7.4 Κρυπτογράφηση των αρχείων τοπικά, πριν την αποστολή στο νέφος*

Έκαστο τεμάχιο του αρχικού αρχείου κρυπτογραφείτε, υπολογίζεται το hash του, το οποίο απόθηκεύεται μαζί με το όνομα της υπηρεσίας νέφους, και το όνομα που του έχει αποδοθεί.

## *5.8 Η διαδικασία κρυπτογράφηση και αποκρυπτογράφησης*

Η διαδικασία κρυπτογράφησης βασίζεται στην βιβλιοθήκη Scrypt, και υλοποιείται στην `icsd.cloudvault.utils.AdvancedFileEncryption` κλάση, με τις μεθόδους encrypt (#L19) και decrypt(L#84). Οι διαδικασίες στηρίζονται στον Αλγόριθμο AES.

## *5.9 Τεμαχισμός αρχείων*

Τα αρχεία τεμαχίζονται κατά απλό τρόπο, σε τρία(3) ίσα μέρη. Το αρχείο αντιμετωπίζεται σαν ένα byte array[], χ μεγέθους όπου το πρώτο τεμάχιο αποτελείται από τα χ/3 πρώτα bytes, όπως και το δεύτερο, ενώ το τρίτο από τα υπολειπόμενα bytes του αρχικού αρχείου. Τη διαδικασία τεμαχισμού μπορεί κάποιος να την μελετήσει στο αρχείο `icsd.cloudvault.utils.FileSplitter.java`

## *5.10Εύρεση των αρχείων για ένωση*

Για να πάρουμε τα κρυπτογραφημένα τεμάχια από τα τις υπηρεσίες νέφους που τα φιλοξενούν θα χρειαστεί να μας γίνουν γνωστά από την δομή αποθήκευσης (state) τα παρακάτω:

- Το όνομα με το οποία θα αναζητήσουμε έκαστο τεμάχιο στην κάθε υπηρεσία
- Και η μοναδική αλληλουχία με την οποία θα πρέπει να τα ενώσουμε για να αποκτήσουμε το αρχικό αρχείο.

Με την οντότητα `icsd.cloudvault.entities.SplittedFile` μπορούμε να ανατρέξουμε στην υπηρεσία που περιέχει το αρχείο που αναζητούμε.

## *5.11Έλεγχος της ακεραιότητας των δεδομένων που επέστρεψαν*

Για να συμβεί ο έλεγχος αυτός, είναι απαραίτητο να έχουμε διατηρήσει στην δομή αποθήκευσης την τα hashes που αντιστοιχούν στα τεμάχια πριν αυτά ανέβουν και έπειτα να τα συγκρίνουμε με τα hashes που επιστρέφονται από τα νέφη.

Η πρώτη λειτουργία υλοποιείται από την `icsd.cloudvault.core.secureAndUpload` κλάση η οποία περιέχει την μέθοδο `secureAndUpload`, αυτή τη σειρά της καλεί την `generateCheckSums` στο αντικείμενο τύπου `icsd.cloudvault.entities.SplittedFile` (#L220).

Η λειτουργία της σύγκρισης απαιτεί την ανάκτηση από την δομή αποθήκευσης , των hashes των τεμαχίων, και τον έλεγχο τους με τα hashes των τεμαχίων που μεταφορτώθηκαν από το νέφος. Η σύγκριση συμβαίνει στην μέθοδο `downloadAndVerify` της `icsd.cloudvault.core.Controller` κλάσης στην #L315 .

## *5.12Ένωση τεμαχίων*

Η ένωση των τεμαχιών συμβαίνει ακολουθώντας την αντίστροφη διαδικασία που περιγράφεται από την ενότητα 5.7. Στο αρχείο `icsd.cloudvault.utils.FileSplitter.java` περιλαμβάνεται και η μέθοδος `mergePartsToFile` που κάνει την ένωση των τεμαχίων.

# 6 Οδηγός χρήσης εφαρμογής

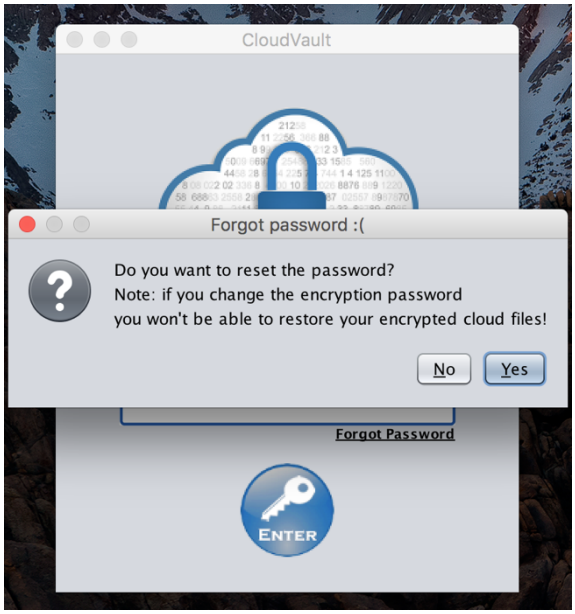
Στο κεφάλαιο αυτό περιλαμβάνουμε μια πλήρη επεξήγηση της λειτουργίας της εφαγμογής με εικόνες. Ενώ πιστεύουμε πως η εφαργμογή είναι απο μόνη της αρκετά απλή για τον τελικό χρήστη(end user), δίνουμε ένα σαφή προσανατολισμό μέσα απο εικόνες και βίντεο[13].

## 6.1 Εκκίνηση εφαρμογής

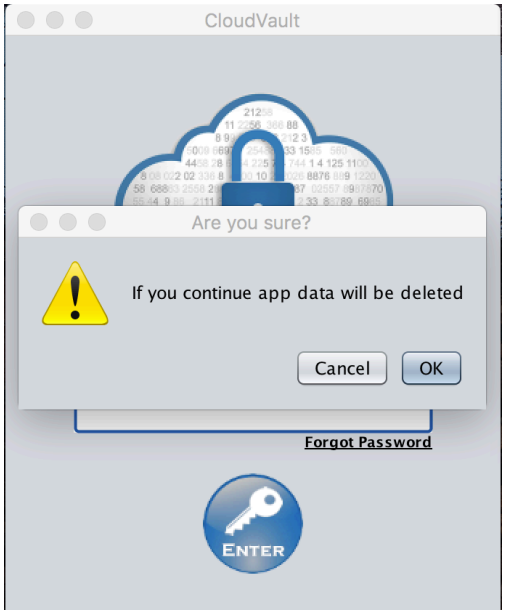
Για να εκτελέσουμε την εφαρμογή χρειαζόμαστε έκδοση Java 1.8. ή νεότερη, στο μηχάνημα μας.

Η πρώτη εικόνα που βλέπει ο χρήστης ζητάει τον master password της εφαρμογής.

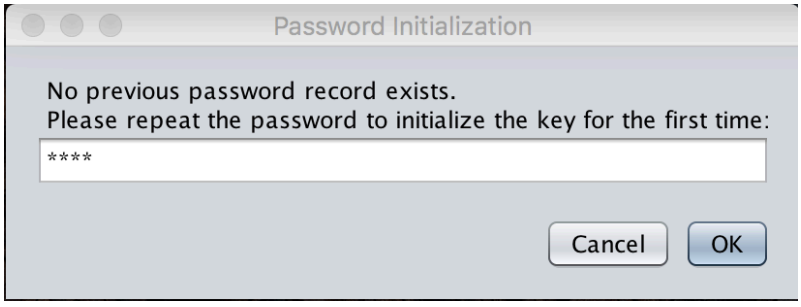

Ο χρήστης καλείται να δώσει το συνθηματικό που θα χρησιμοποιεί για να έχει πρόσβαση στα δεδομένα του. Ο κωδικός του χρήστη καλύπτεται κατά την πληκτρολόγηση για λόγους ασφάλειας, όπως άλλωστε συνηθίζεται.



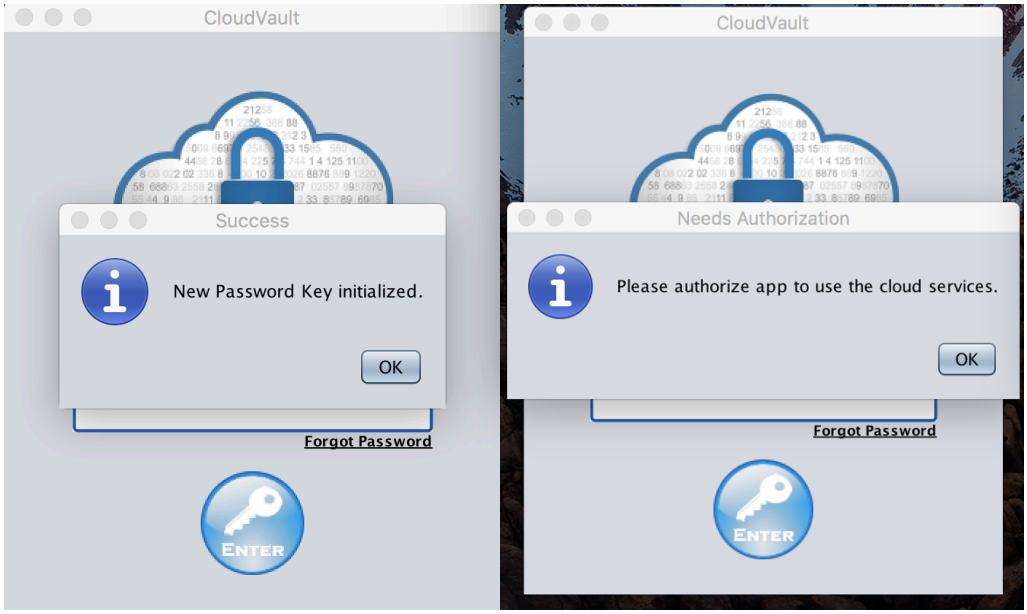

Σε αυτό το σημείο είναι σημαντικό να τονίσουμε πως τυχών απώλεια του κωδικού οδηγεί και σε απώλεια ανάκτησης των δεδομένων του χρήστη από τα νέφη. Η επιλογή "Forgot Password" διαγράφει τα μεταδεδομένα που έχουν αποθηκευτεί στον υπολογιστή από προηγούμενη χρήση της εφαρμογής.




Η επιλογή συνοδεύεται από παράθυρο επιβεβαίωσης.

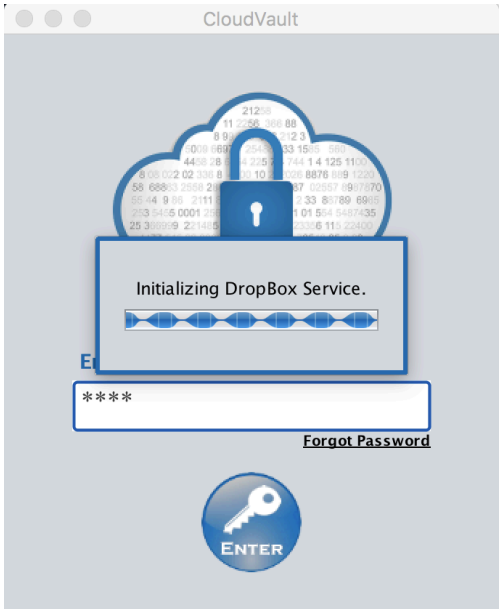


Εφόσον δεν υπάρχει αρχείο με καταχωρημένους κωδικούς, η εφαρμογή ζητάει επιβεβαίωση του κωδικού από το πρώτο βήμα. Και σε αυτή την περίπτωση ο κωδικός είναι κρυμμένος(masked).

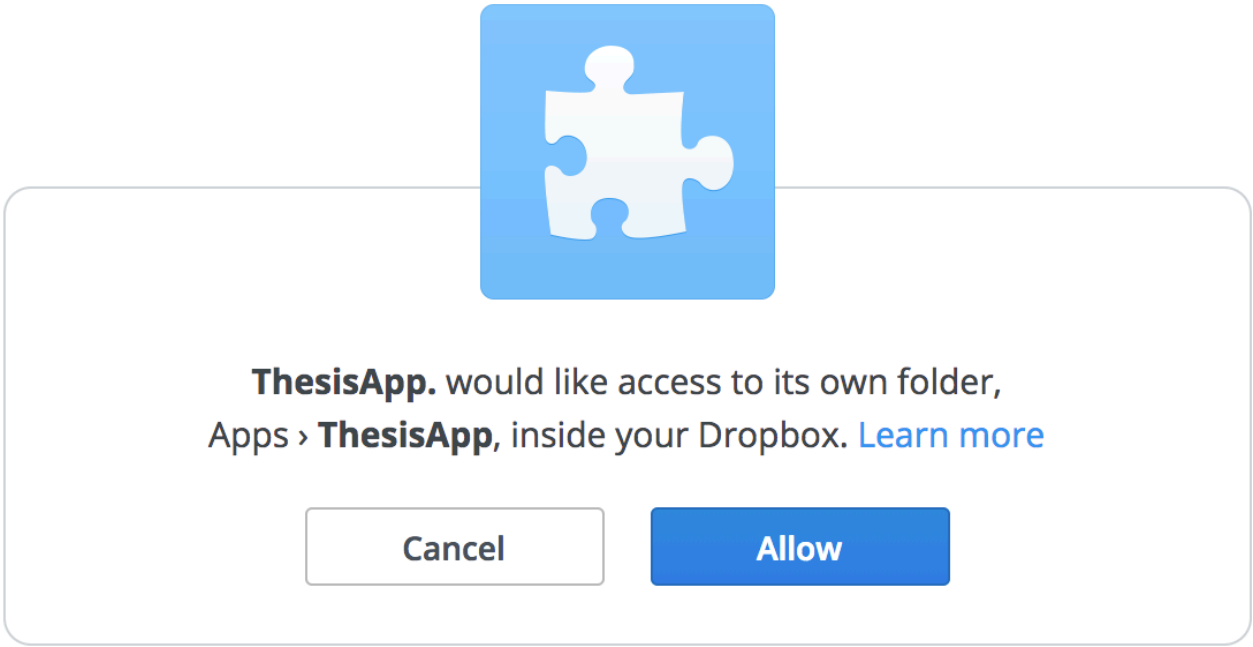


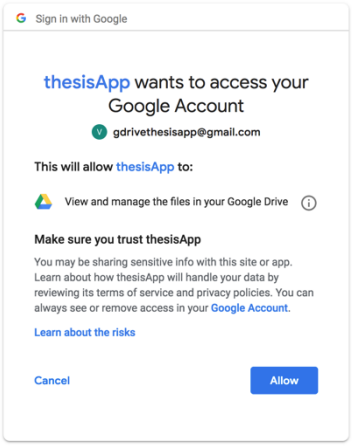

Η εφαρμογή στη συνέχεια ενημερώνει για την επιτυχή εξέλιξη της αρχικοποίησης του κωδικού με κατάλληλο μήνυμα και σε επόμενο παράθυρο ενημερώνει πως η εφαρμογή ζητάει authorization για τις υπηρεσίας cloud.

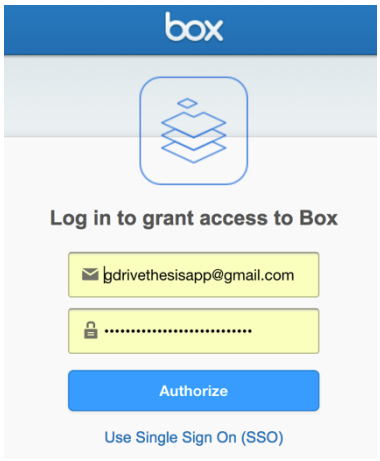
## *6.2 Αρχικοποίηση υπηρεσιών*




Το επόμενο βήμα περιλαμβάνει η αρχικοποίηση (initialization) όλων των υπηρεσιών. Το παράθυρο της εφαρμογής μένει ανοιχτό και δείχνει κάθε φορά την υπηρεσία με την οποία επικοινωνεί.




Για κάθε υπηρεσία ανοίγει tab του browser και ο χρήστης κάνει είτε log in, είτε αν είναι ήδη logged in στην υπηρεσία δίνει απλά τη συγκατάθεση του για να γίνει authorize η εφαρμογή. Στις πιο πάνω εικόνες βλέπουμε και τις 3 περιπτώσεις.

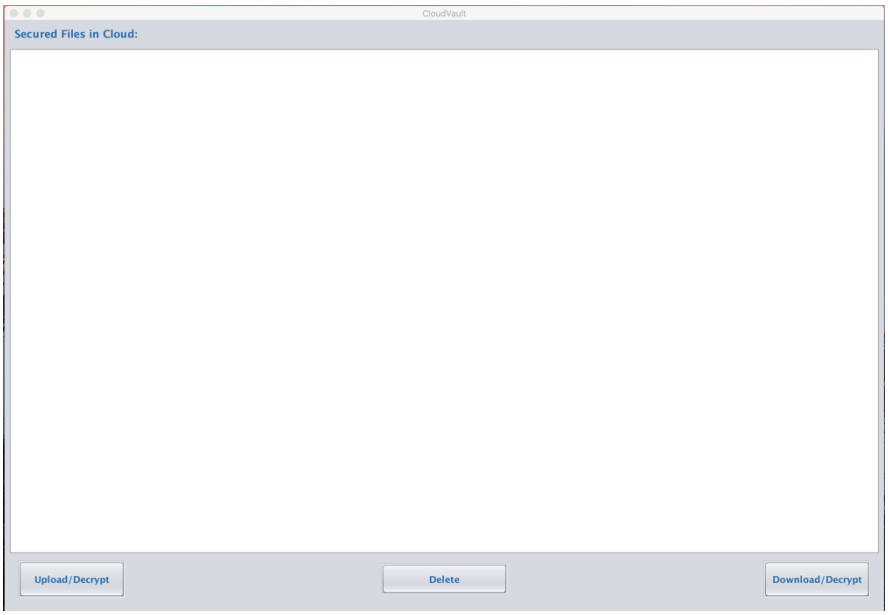
Received verification code for GoogleDrive cloud service.
You may now close this window.

Received verification code for DropBox cloud service.
You may now close this window.

Received verification code for Box cloud service.
You may now close this window.

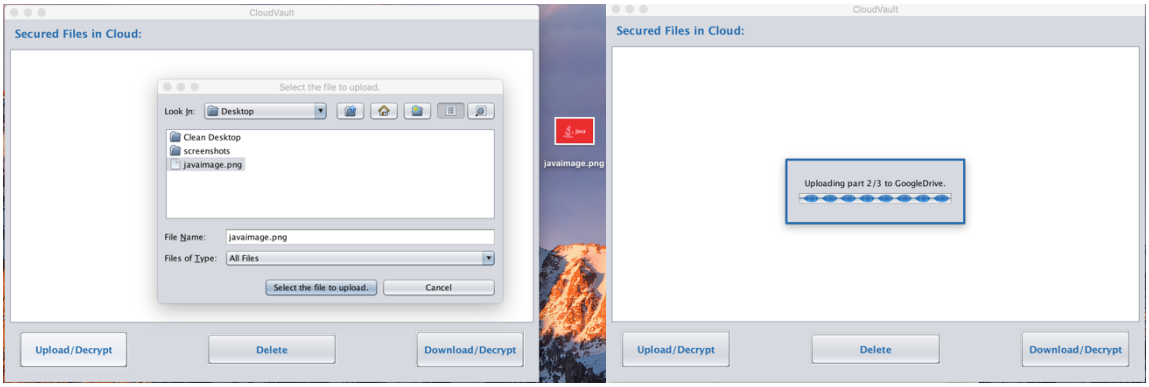
Κάθε φορά που εξουσιοδοτούμε την εφαρμογή μας, αυτή μας ενημερώνει πως η διαδικασία ολοκληρώθηκε και μπορούμε να κλείσουμε το αντίστοιχο tab του browser μας.

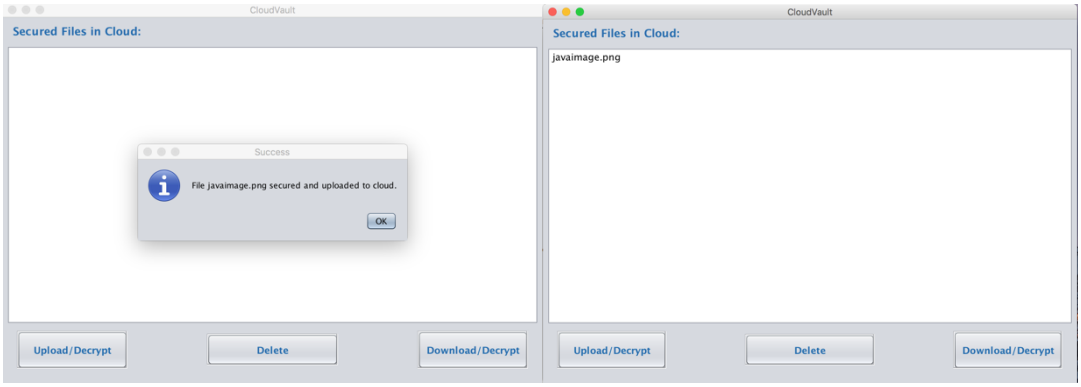
## *6.3 Κεντρική οθόνη ενεργειών*



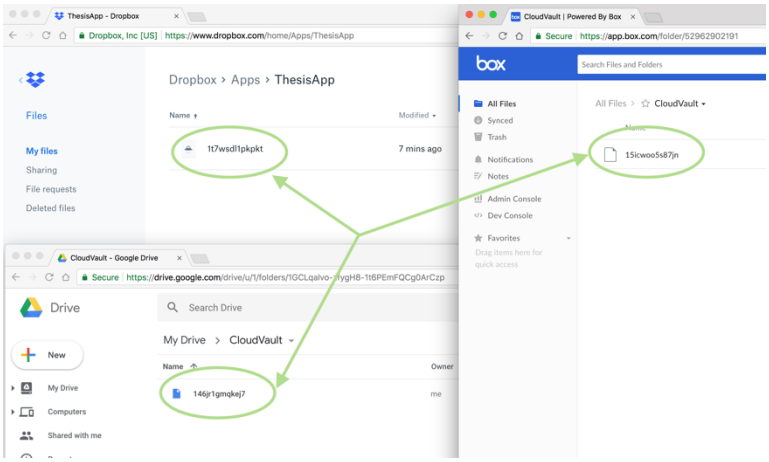

Εφόσον οι διαδικασίες της αυθεντικοποίησης τελειώσουν, η εφαρμογή αποδίδει στον χρήστη το κεντρικό παράθυρο. Εδώ ο χρήστης μπορεί να εκτελέσει τις βασικές λειτουργίες του upload, download και delete αρχείων στο νέφος.

## 6.4 Διαδικασία upload αρχείων



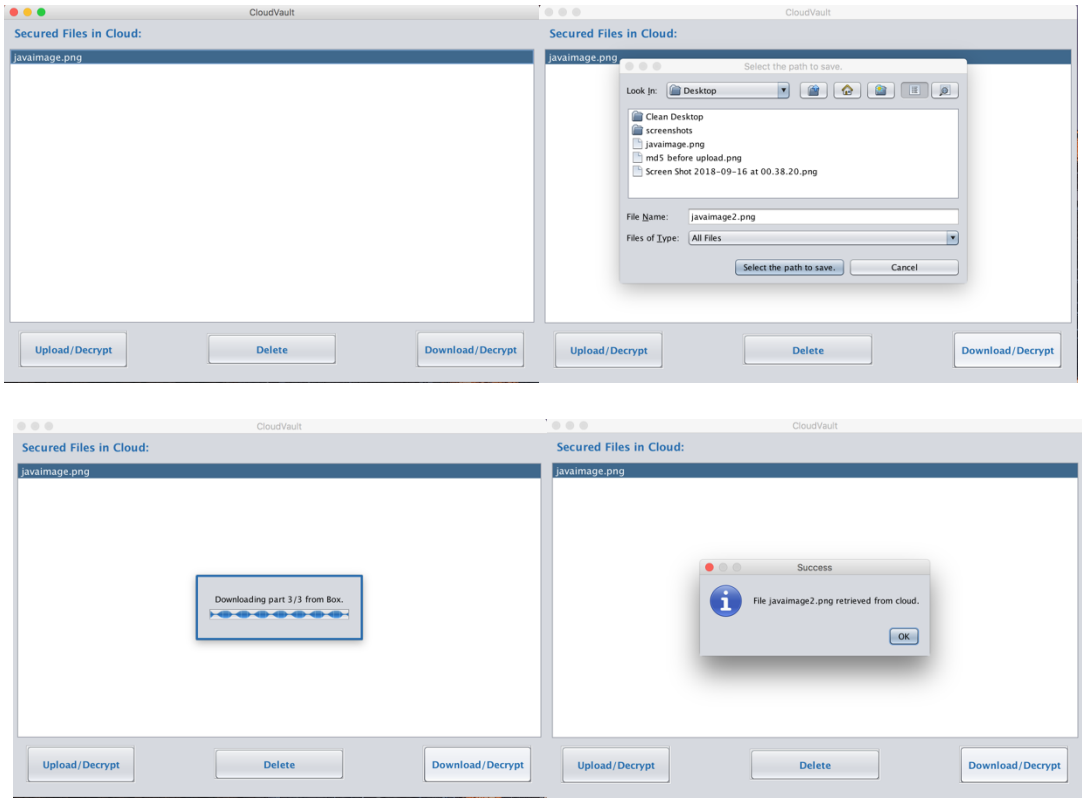

Για να ανεβάσουμε ένα αρχείο, αρκεί να πατήσουμε το πλήκτρο με Upload και η εφαρμογή θα κάνει τα υπόλοιπα, ενώ μας δίνει τη δυνατότητα να παρακολουθούμε της διαδικασία μεταφόρτωσης του αρχείου προς όλες τις υπηρεσίες, με pop ups και σχετικό μήνυμα που ειδοποιεί το χρήστη στο τέλος της επιτυχούς έκβασης της μεταφόρτωσης. Τελικά ο χρήστης μπορεί να δεί στην λίστα αρχείων(κεντρικό παράθυρο).




Αν θέλουμε μπορούμε να εξετάσουμε και τις υπηρεσίες, και εκεί θα βρούμε τα τεμάχια του αρχικού αρχείου. Η εφαρμογή όπως έχουμε περιγράψει έχει δώσει σε κάθε υπηρεσία ένα αρχείο

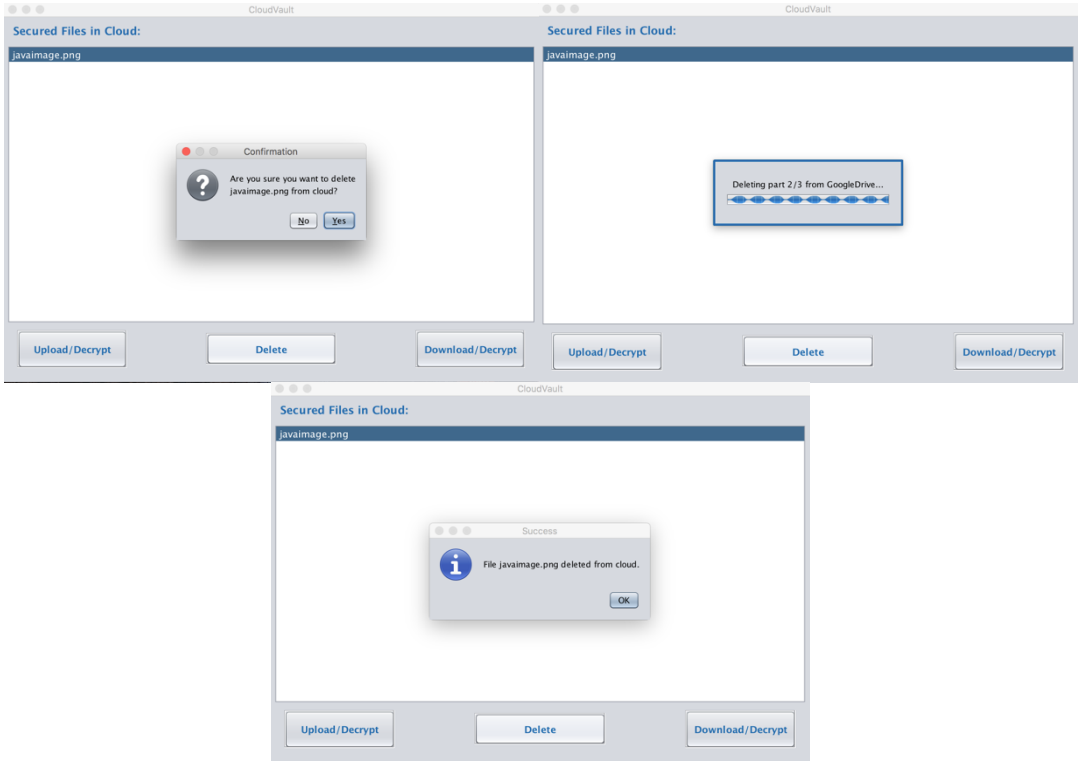
με όνομα που δεν μπορεί να συσχετιστεί με το αρχικό αρχικό αρχείο, ενώ τα κομμάτια μεταξύ τους δεν μπορούν επίσης να συσχετιστούν απο το όνομα τους, μεταξύ τους.

## *6.5 Διαδικασία download αρχείων*




Απο την κεντρική οθόνη ο χρήστης επιλέγει το αρχείο που τον ενδιαφέρει να κατεβάσει, και η πατώντας το πλήκτρο download, μπορεί να διαλέξει και το path που θέλει να σωθεί το αρχείο του. Έπειτα η εφαρμογή ξεκινάει τη μεταφόρτωση των τεμαχίων, ενώ τον ενημερώνει σχετικά με την εξέλιξη της και όταν τελειώσει η ένωση των τεμαχίων και εφόσον όλα έχουν πάει σωστά ( τα αρχεία δεν έχουν αλλοιωθεί) ο χρήστης ενημερώνεται πως το αρχείο του είναι πλέον στη διάθεση του.

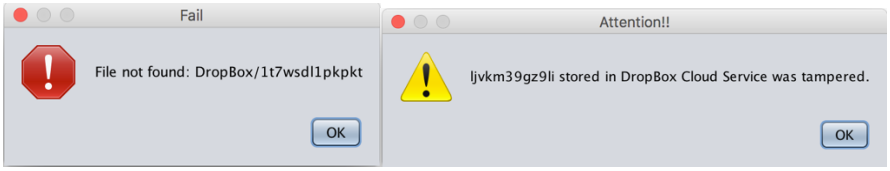
## 6.6 Διαδικασία διαγραφής αρχείου



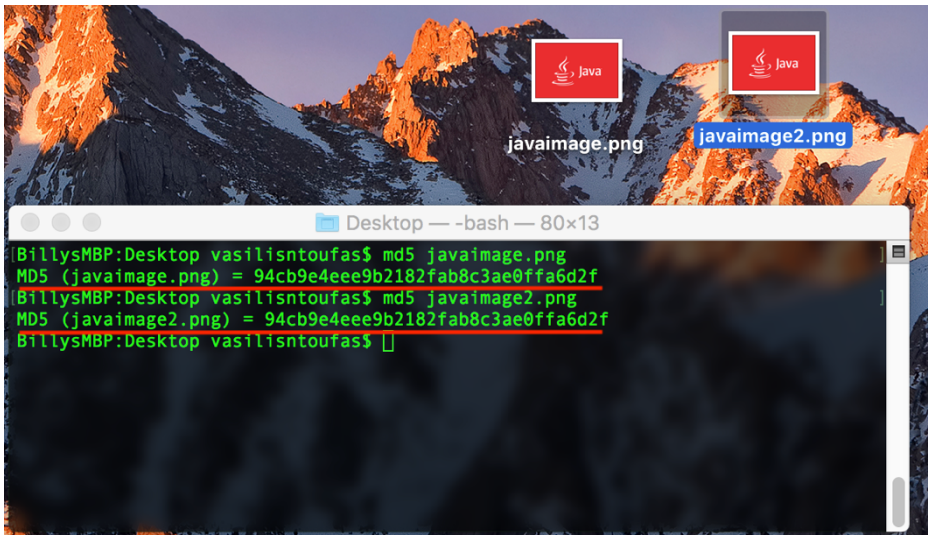

Στην επιλογή διαγραφής αρχείου, η εφαρμογή ακολοθεί το ίδιο μοτίβο συμπεριφοράς με τις προηγούμενες δύο λειτουργείες, ενημερώνοντας τον χρήστη κατάλληλα για την κατάσταση και την έκβαση της διαδικασίας. Στο τέλος το αρχείο που διεγράφει απο τις υπηρεσίες νέφους έχει αφαιρεθεί απο τη λίστα των διαθέσιμων αρχείων.

## 6.7 Ενημέρωση παραποίησης αρχείου

Η εφαρμογή δίνει τη δυνατότητα στον χρήστη να ενημερωθεί για τυχών παραποίηση αρχείου σε κάποια υπηρεσία νέφους. Εδώ προσομοιώνουμε το σενάριο με την εσκεμένη μεταφόρτωση αλλοιωμένου τεμαχίου και την εσκεμένη διαγραφή τεμαχίου απο υπηρεσία νέφους.



## 6.8 *Proof of concept*




Μπορούμε πολύ εύκολα να δείξουμε την ορθή λειτουργία της εφαρμογής μας μέσα από τη σύγκριση του αρχικού hash του αρχείου μας, με το hash του τελικού αρχείου. Στην εικόνα φαίνεται πως τα 2 αρχεία αποδίδουν το ίδιο md5 hash.

## 6.9 *Παρουσίαση μέσω βίντεο*

Στο [13] μπορεί ο ενδιαφερόμενος να παρακολουθήσει τη λειτουργία της εφαρμογής απο βίντεο.

# 7 *Αξιολόγηση εφαρμογής*

Σε αυτό το κεφάλαιο παρουσιάζουμε μια αξιολόγηση της εφαρμογής και τη συγκρίνουμε με παρόμοιες υλοποιήσεις που υπάρχουν σήμερα στην αγορά ως προϊόντα. Αυτό το κεφάλαιο έχει πολύ περιορισμένη έκταση εξαιτίας της περιορισμένης γκάμας επιλογών.

## *7.1 Διαθέσιμα προϊόντα*

Μια λύση που διαθέτει κάποια καλά χαρακτηριστικά είναι το odrive[14]. Το προϊόν αυτό «σκεπάζει» κάτω απο τον client του την μεγαλύτερη γκάμα υπηρεσιών νέφους και ταυτόχρονα κάνει και κρυπτογράφιση των αρχείων που αποστέλουμε στις υπηρεσίες αυτές. Μια διαφορά που εντοπίζουμε είναι πως παρέχει πρόσβαση και σε αρχεία κοινωνικών δικτιών.

Το crosscloud[15] κάνει μια λίγο διαφορετική προσέγγιση, κρυπτογραφώντας ισχυρά (AES-256 και RSA-4096), δίνει και τη δυνατότητα στον χρήστη να γνωρίζει ποια αρχεία του είναι κρυπτογραφημένα και ποια όχι ενώ τέλος δίνει τη δυνατότητα να μοιράζεται κρυπτογραφημένα αρχεία.

## *7.2 Σύγκριση*

Ενώ δεν μπορούμε να συγκρίνουμε εμπορικά προϊόντα με το αποτέλεσμα μιας ακαδημαϊκής εργασίας, μπορούμε να διαπιστώσουμε εύκολα πως το μικρό εύρος λειτουργιών της εφαρμογής μας λειτουργεί πιο αποτελεσματικά για την διασφάλιση των δεδεμένων απο τα εμπορικά προϊόντα. Όσο μεγαλώνει το εύρος των δυνατοτήτων των εφαρμογών πολλές φορές πρέπει να γίνονται υποχωρήσεις. Στη δική μας υλοποίηση δεν υπάρχει αυτό το πρόβλημα, ένας όμως μεγάλος οργανισμός ή υπηρεσία που παρέχει παρόμοια προϊόντα πρέπει να κάνει είτε σοβαρό αρχιτεκτονικό σχεδιασμό απο την αρχή, να καταφύγει σε εκπτώσεις στην πορεία για να μπορέσει να εξυπηρετήσει με μια λύση «χρυσής τομής» που θα συνδιάζει και τους επιχειρηματικούς στόχους του αλλά και την κάλυψη των τεχνολογικών απαιτήσεων.

# 8 *Συμπεράσματα και περαιτέρω έρευνα*

## *8.1 Προτάσεις για μελλοντική μελέτη*

Ελέγχοντας την προσέγγιση μας διαπιστώθηκαν τομείς που αποτελούν ενδιαφέροντα σημεία για περαιτέρω μελέτη, συγγραφή και υλοποίηση. Η κάθε μια από τις παρακάτω περιπτώσεις μπορεί να σταθεί και σαν δική της έρευνα με τα δικά της μετρήσιμα αποτελέσματα σε τομείς ταχύτητας, απόδοσης, βελτιστοποίησης ασφάλειας. Όλες οι περιπτώσεις μελέτης που περιγράφονται παρακάτω, θεωρούν δεδομένη την εφαρμογή της προσέγγισης που έχουμε ήδη περιγράψει στην εργασία αυτή και ως επέκταση της υιοθετούν χαρακτηριστικά που βελτιώνουν την περίπτωση μελέτης μας, αποδίδοντας χαρακτηριστικά ταχύτητας, μεγαλύτερης ασφάλειας και ευελιξίας βελτιστοποιώντας το τελικό μοντέλο υλοποίησης.

## *8.2 Απόδοση διαφορετικών κλειδιών σε κάθε υπηρεσία νέφους*

Στη προσέγγιση αυτή προτείνουμε τη χρήση διαφορετικού κλειδιού κρυπτογράφησης για κάθε μια από τις υπηρεσίες νέφους. Ο τρόπος αυτός μπορεί να υλοποιηθεί με διαμόρφωση των δομών που προτείνει η παρούσα εργασία, έτσι ώστε να παράγονται περισσότερα του ενός κλειδιά και να παράγονται νέα salts κάθε φορά κατά την κρυπτογράφηση. Στη συνέχεια να αποθηκεύονται ξεχωριστά τα hashes τους στη δομή που αναπαριστά τα αρχεία που έχουνε αποδοθεί στο νέφος, ώστε να μπορούν έπειτα να αποκρυπτογραφηθούν μετά από την σύγκριση των hashes των κλειδιών.

## *8.3 Τεμαχισμός αρχείων με βάση τα quotas*

Ο διαχωρισμός των αρχείων που χωρίζονται, με κριτήριο των ελεύθερο χώρο (quotas) που υπολείπεται σε κάθε υπηρεσία αποθήκευσης . Στην παρούσα εργασία ο διαχωρισμός γίνεται αμερόληπτα, χωρίς βάρος, χωρίς ποσόστωση, σε ίσα μέρη, μη λαμβάνοντας υπόψιν κάποιον παράγοντα που σχετίζεται με τη μεριά του νέφους . Η προσέγγιση που προτείνουμε για επιπλέον μελέτη θα βοηθούσε στην καλύτερη διαχείριση και χρήση υπηρεσιών που έχουνε διαφορετικούς διαθέσιμους χώρους, πράγμα αρκετά διαδεδομένο μεταξύ των δωρεάν υπηρεσιών. Η υλοποίηση

αυτή μπορεί να συμβεί με προγραμματιστική ερώτηση προς το νέφος, του quota του λογαριασμού του χρήστη.

## *8.4 Τεμαχισμός αρχείων με γεωγραφικά κριτήρια (απόσταση client - server)*

Σε αυτή την περίπτωση προτείνουμε τον τεμαχισμό των αρχείων σε μεγέθη που είναι ανάλογα με την απόσταση του client προγράμματος από τον διακομιστή της υπηρεσίας. Ποσόστωση των μεγεθών δηλαδή προς όφελος της ταχύτητας μεταφοράς των αρχείων. Μια προσέγγιση που θα βελτίωνε τους χρόνους μεταφοράς των αρχείων. Το πρόβλημα που έχει μια τέτοια υλοποίηση είναι ο τρόπος που θα μετράται η απόσταση μεταξύ εφαρμογής πελάτη και διακομιστή της υπηρεσίας, καθώς τα νέφη κρύβονται πίσω από δικτυακές υλοποιήσεις για να προσφέρεται redundancy στις υπηρεσίες. Αν υποθέσουμε πάντως πως μια μέτρηση των ενδιάμεσων σταθμών (hops από την εντολή tracert(windows)/traceroute(unix)) είναι πλήρως αντιπροσωπευτική, τότε η προγραμματιστική μέτρηση της απόστασης των διαδοχικών κόμβων συνδέσεων με γεωγραφική τοποθέτηση τους με βάση την ip, μπορεί να παράξει μια ενδιαφέρουσα περίπτωση μελέτης.

## *8.5 Έλεγχος απόπειρας παραβίασης αρχείου*

Αυτή η περίπτωση είναι πιο περίπλοκη από τις προηγούμενες. Αφορά ανάπτυξη μεθοδολογίας που θα επιτρέπει να εξάγουμε σαφή συμπέρασμα για το αν για το τεμάχιο του αρχείου που έχει παραδοθεί στο νέφος έχει γίνει προσπάθεια να παραβιαστεί η ασφάλεια του. Σαφώς εδώ το αντικείμενο μελέτης περνάει είτε σε επίπεδο αρχείου είτε σε επίπεδο μηχανισμού αποθήκευσης στο hardware. Το σενάριο περιπλέκεται ακόμα πιο πολύ αν υποθέσουμε πως ένας abuse operator αντιγράψει εκτός νέφους το εν λόγω αρχείο.

## *8.6 Υλοποίηση μηχανισμού για τον τεμαχισμό των αρχείων.*

Εδώ μπορούμε να θωρακίσουμε επιπλέον την διαδικασία μας με μια απλή μέθοδο. Αντί ο τεμαχισμός των αρχείων να γίνεται με απλό διαχωρισμό (μέγεθος αρχείου διαιρεμένο δια του τρία) όπως συμβαίνει στην παρούσα εργασία σε επίπεδο αρχείου, προτείνουμε τον διαχωρισμό σε επίπεδο byte αλλά με την παρεμβολή πιο σύνθετου αλγορίθμου που επιλέγει τις θέσεις τον bytes που θα πάρει από το αρχικό αρχείο για να δημιουργήσει το κάθε τεμάχιο. Μία περίπτωση θα μπορούσε είναι η δομή που περιγράφεται από το παρακάτω σχήμα.

Πίνακας bytes αρχικού αρχείου

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T[0] | T[1] | T[2] | T[3] | T[4] | T[5] | T[6] | T[7] | T[8] | T[9] | ... | ... | ... | T[N] |

Πίνακες bytes τεμαχίων A, B, C

| | | | | |
|---|---|---|---|---|
| Ta[T[0]] | Ta[T[3]] | Ta[T[6]] | ... | Ta[T[N] |

| | | | | |
|---|---|---|---|---|
| Tb[T[1]] | Tb[T[4]] | Tb[T[7]] | ... | Tb[T[N-2] |

| | | | | |
|---|---|---|---|---|
| Tc[T[2]] | Tc[T[5]] | Tc[T[8]] | ... | Tc[T[N-1] |

Για αριθμό bytes N, για το οποίο ισχύει N/3 = 0.

Εδώ περιγράφουμε την δημιουργία του πρώτου τεμάχιού από τις θέσεις bytes του αρχικού αρχείου που αντιστοιχούν στις θέσεις i = i + ν, για i από 0 έως μέγεθος αρχείου/3, όπου ν ο αριθμός των υπηρεσιών νέφους που θα μεταφορτωθεί το αρχείο. Έτσι για παράδειγμα στην πρώτη υπηρεσία, θα ανέβει αρχείο που θα έχει δημιουργηθεί από τις θέσεις bytes του αρχικού αρχείου που αντιστοιχούν στα bytes 0, 3, 6, 9 κ.ο.κ. ενώ το δεύτερο τεμάχιο θα αποτελείται από τα bytes που βρίσκονται στις θέσεις 1, 4, 7, 10 κ.ο.κ. του αρχικού αρχείου.

## *8.7 Μελέτη υλοποίησης μηχανισμού δομής αποθήκευσης*

Στην παρούσα εργασία, υλοποιήθηκε μια λύση με μια τοπική δομή αποθήκευσης , όπου τα μεταδιδόμενα που χρειάζεται η εφαρμογή αποθηκεύονται σε ένα αρχείο. Ακολουθήσαμε αυτή την μέθοδο και όχι την χρήση μια βάσης δεδομένων για καθαρά τεχνικούς λόγους ώστε να μην δώσουμε μεγαλύτερη έκταση σε μια υλοποίηση που ήδη ήτανε μεγάλη και την λειτουργία της παραμονεύουν οι αλλαγές των api των υπηρεσιών νεφών. Η δομή αποθήκευσης εδώ μπορεί να βελτιωθεί και να αποτελέσει αντικείμενο μελέτης στο κατά πόσο μπορούμε να μειώσουμε την έκταση των μεταδιδόμενων που αποθηκεύουμε τοπικά, στέλνοντας τα και αυτά στο νέφος, αλλά διατηρώντας το ίδιο επίπεδο ασφάλειας αν όχι αυξάνοντας το.

Σε επίπεδο byte Θα μπορούσαμε να προσαρτήσουμε στο τέλος κάθε τεμαχίου μια σειρά από μεταδιδόμενα που αφορούν το αρχικό αρχείο και του ελέγχους ακεραιότητας του αρχικού αρχείου ή/και του τεμαχίου. Αποθηκεύοντας έτσι λιγότερες πληροφορίες τοπικά, αποφεύγουμε δυσλειτουργίες του συστήματος από αστοχίες υλικού (π.χ. απώλεια αρχείου δομής αποθήκευσης λόγω αστοχίας σκληρού δίσκου).

# 9 *Αναφορές*

[1] https://www.iso.org/home.html
[2]DigitalGuardian, The history of data breaches, https://digitalguardian.com/blog/history-data-breaches
[3] Shushant Srivastava, Vikas Gupta, Rajesh Yadav, Krishna Kant (2010), Enhanced Distributed Storage on the Cloud. 2012 Third International Conference on Computer and Communication Technology (p.321-325).
[4] Fox, Armando, et al. "Above the clouds: A berkeley view of cloud computing." Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS 28.13 (2009): 2009.
[5] Keke Gai, Meikang Qiu, Hui Zhao (2016), Security-Aware Efficient Mass Distributed Storage approach for Cloud Systems in Big Data. 2016 IEEE 2nd International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, IEEE International Conference on Intelligent Data and Security (p.140 - 145).
[6] Amazon, "AmazonWeb Services,"http://aws.amazon.com/.
[7] Google, "Google app Engine" http://code.google.com/appengine/.
[8] Microsoft, "Windows Azure," http://www.microsoft.com/.
[9]Ντούφας Βασίλης (2018), "Securing File In the Cloud (Java Code) ", https://bitbucket.org/Vntoufas/thesis/src/master/SecuringFilesInTheCloud/
[10] Dropbox Business Security ,A Dropbox whitepaper https://cfl.dropboxstatic.com/static/business/resources/dfb_security_whitepaper-vflIocB9q.pdf
[11] Google Cloud Security https://support.google.com/googlecloud/answer/6056693?hl=en
[12] Box, is my data encrypted https://community.box.com/t5/How-to-Guides-for-Account/Is-My-Data-Encrypted/ta-p/32
[13] Video – Presentation, Ντούφας Βασίλης, Παρουσίαση εφαρμογής διπλωματικής εργασίας https://www.dropbox.com/s/kg2evvki984nyjy/Thesis%20video%20presentation.mp4?dl=0
[14] Odrive, https://www.odrive.com/
[15] Crosscloud , http://crosscloud.io/