ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ WEB SERVICE ΩΣ ΥΠΗΡΕΣΙΑ ΒΙΟΜΕΤΡΙΚΗΣ ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗΣ

Του φοιτητή

Καρυώτη Ιωάννη

Αρ. Μητρώου: 154461

Επιβλέπων

Ηλιούδης Χρήστος

Καθηγητής

Θεσσαλονίκη, Σεπτέμβριος 2021

Τίτλος Δ.Ε. Ανάπτυξη web service ως υπηρεσία βιομετρικής αυθεντικοποίησης
Κωδικός Δ.Ε. 20223
Ονοματεπώνυμο φοιτητή Καρυώτης Ιωάννης
Ονοματεπώνυμο εισηγητή Ηλιούδης Χρήστος
Ημερομηνία ανάληψης Δ.Ε. 01-11-2020
Ημερομηνία περάτωσης Δ.Ε. 10-09-2021

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Δύο τομείς, που μου προκάλεσαν μεγάλο ενδιαφέρον κατά τη διάρκεια φοίτησής μου στο τμήμα πληροφορικής, αποτελούν τη βάση έμπνευσης αυτής της εργασίας. Ο πρώτος αφορά την μηχανική μάθηση και πιο συγκεκριμένα το Deep Learning, που αποτέλεσε κομμάτι της πρώτης μου πρακτικής άσκησης. Ενώ ο δεύτερος αποτελεί το Web Development, το οποίο πλέον είναι μέρος της καθημερινής μου εργασίας και απασχόλησης. Η αγάπη μου γι' αυτούς τους δυο τομείς και η θέληση να τους συνδυάσω, βοήθησε στην αρχική επινόηση του θέματος της παρούσας πτυχιακής και τελικά στην υλοποίησή της.

Περίληψη

Ο όρος Βιομετρία χρησιμοποιείται ιδιαίτερα συχνά στα σύγχρονα υπολογιστικά συστήματα και η χρήση του έχει αυξηθεί με το πέρασμα των χρόνων. Τα βιομετρικά χαρακτηριστικά των ανθρώπων ίσως είναι το πιο ακριβές μέσο εξακρίβωσης της ταυτότητας ενός ατόμου. Η μοναδική αυτή τους ιδιότητα έχει οδηγήσει στην ραγδαία αύξηση των βιομετρικών συστημάτων στις μέρες μας. Τα βιομετρικά αυτά συστήματα κάνουν χρήση διάφορων μεθόδων αναγνώρισης και ταυτοποίησης, μέσω των χαρακτηριστικών προσώπου, το δαχτυλικό αποτύπωμα κ.λπ. Ωστόσο, το πλήθος των βιομετρικών συστημάτων, που προσφέρουν βιομετρική αυθεντικοποίηση κατά απαίτηση, είναι μικρό, πόσο μάλλον τα συστήματα που προφέρουν αυθεντικοποίηση με τη χρήση των χαρακτηριστικών του προσώπου. Η παρούσα πτυχιακή εργασία παρουσιάζει την πρώτη έκδοση μιας υπηρεσίας ιστού, που λειτουργεί ως ένα βιομετρικό σύστημα και, ακολουθώντας το «As A Service» επιχειρησιακό μοντέλο, προσφέρει διαλειτουργική, βιομετρική αυθεντικοποίηση, χρησιμοποιώντας τα χαρακτηριστικά του προσώπου.

Συνεπώς, ο κύριος στόχος της υπηρεσίας αυτής, είναι να συνδέσει και να προσφέρει υποστήριξη βιομετρικής αυθεντικοποίησης σε οργανισμούς ή εφαρμογές τρίτων, χωρίς αυτοί να αλλάξουν τις υποδομές τους ή να υλοποιήσουν ειδικές λύσεις. Η παρούσα έκδοση προσφέρει υπηρεσίες σύνδεσης εφαρμογών και αυθεντικοποίησης χρηστών, χρησιμοποιώντας την προτυποποιημένη ροή του πρωτοκόλλου εξουσιοδότησης OAuth 2.0. Με αυτόν τον τρόπο, οι χρήστες αποκτούν τη δυνατότητα να συνδεθούν σε εφαρμογές τρίτων, επαληθεύοντας τη ταυτότητά τους με τα χαρακτηριστικά των προσώπων τους, μέσω της υπηρεσίας αυτής.

Development of a web service that serves as a biometric authentication server

Ioannis Karyotis

Abstract

The term Biometry, is used rather commonly in modern computer systems and its use has been increasing throughout the years. People's biometric characteristics, may be the most accurate means of defining one's identity. This unique quality has led to a rapid increase in biometric systems nowadays. These biometric systems offer services of user recognition and identification that work with facial characteristics, fingerprints etc. However, the number of biometric systems that offer biometric authentication on demand, is rather small, let alone systems that offer authentication using facial characteristics. This thesis presents the first version of a web service that works as a biometric system and by following a "As a Service" business model, it offers interoperable biometric authentication using facial characteristics.

As a conclusion, the main goal of this web service is to connect and offer biometric authentication support to third party organizations or applications, without changing their main infrastructure or them implementing their own custom solutions. The present version offers services of connecting third party apps and authenticating users, using the standard flow of the OAuth 2.0 authorization protocol. In this way, users are able to register and sign in to third party applications using their facial characteristics.

Ευχαριστίες

Η παρούσα πτυχιακή εργασία σηματοδοτεί το κλείσιμο της φοιτητικής μου πορείας ως Μηχανικός Πληροφορικής στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου Ελλάδος. Μιας πορείας που θα ήταν πολύ διαφορετική χωρίς τη στήριξη, τη συνεργασία, τη βοήθεια και την καθοδήγηση μερικών ανθρώπων τους οποίους θα ήθελα να ευχαριστήσω.

Αρχικά θα ήθελα να ευχαριστήσω, τον καθηγητή μου κ. Ηλιούδη για τη καθοδήγηση αλλά και για τη ομαλή και καλή συνεργασία που είχαμε. Ταυτόχρονα, θέλω να ευχαριστήσω το φίλο και συνάδελφο Βασίλη Παπουτσάκη για τη καθοδήγηση και τις γνώσεις που μου πρόσφερε το τελευταίο χρόνο, γεγονός που συνέβαλε αρκετά στην υλοποίηση της πρακτικής εφαρμογής. Με έκανε να αγαπήσω την επιστήμη μου ακόμα περισσότερο και γι' αυτό τον ευχαριστώ θερμά. Ευχαριστώ επίσης, το φίλο και συμφοιτητή Γιώργο Μπάρκο που με βοήθησε στη τελική μορφοποίηση του κειμένου της πτυχιακής εργασίας.

Τέλος, το μεγαλύτερο ευχαριστώ χρωστάω στις δύο μου οικογένειες, τη μια που μας ενώνει το αίμα, και τους φίλους μου, που είναι πάντα στο πλευρό μου και αποτελούν πρότυπά μου για να γίνομαι συνεχώς καλύτερος άνθρωπος.

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

2.F.A.	Two Factor Authentication
O.T.P.	One Time Password
A.A.S.	As A Service
B.A.A.S.	Biometric As A Service
P.K.I.	Public Key infrastructure
T.S.A.	Time Stamping Authority
S.A.M.L.	Security assertion markup language
O.AUTH.	Open Authorization
O.ID.C.	Open ID Connect
J.W.T	Json Web Token
F.A.R	False Acceptance Rate
F.R.R	False Rejection Rate
S.S.D.	Single Shot Multibox Detector
C.N.N.	Convolution Neutral Networks
M.T.C.N.N.	Multi-task Cascaded Convolutional Neural Network
G.U.I.D.	Globally Unique Identifier
M.V.C	Model View Controller

Κεφάλαιο 1ο: Περιοχή μελέτης και στόχοι πτυχιακής εργασίας

1.1 Εισαγωγή

Ζούμε σε μια εποχή, όπου η εξέλιξη της τεχνολογίας των υπολογιστών και της πληροφορικής έχει φτάσει σε τέτοιο επίπεδο, όπου απαιτείται η ανάπτυξη όλο και πιο ασφαλών, αλλά και ταυτόχρονα αξιόπιστων μεθόδων πρόσβασης και παροχής υπηρεσιών στους κοινούς, καθημερινούς χρήστες.

Από τις αρχές του διαδικτύου είχαν εφευρεθεί διάφοροι τρόποι ταυτοποίησης και αυθεντικοποίησης των χρηστών σε Web υπηρεσίες. Το πιο κλασικό παράδειγμα, αποτελεί η χρήση ψευδωνύμου (username) και ενός αναγνωριστικού ή αλλιώς, κωδικού πρόσβασης. Με τα χρόνια όμως, έχουν βρεθεί πολλές ευπάθειες αυτών των μεθόδων, με αποτέλεσμα πολλοί κακόβουλοι χρήστες, αξιοποιώντας έξυπνες τεχνικές hacking, να μπορούν να προσπερνούν την ασφάλεια των OnLine υπηρεσιών και να μπορούν να μιμούνται εικονικά την ταυτότητα άλλων χρηστών, χρησιμοποιώντας την έτσι, για το δικό τους συμφέρον.

Ωστόσο, οι κλασικοί τρόποι αυθεντικοποίησης συνέχισαν να υπάρχουν, αλλά και να ενισχύονται με πολλούς τρόπους, έτσι ώστε να εξαλείφονται οι ευπάθειές τους. Παράδειγμα θα μπορούσε να αποτελέσει η πλέον απαίτηση χρήσης ισχυρότερων κωδικών πρόσβασης με περισσότερους χαρακτήρες, σύμβολα και γράμματα, ώστε να αποτρέπεται εύκολα ή εξακρίβωση του συνθηματικού μέσω της μεθόδου Brute Force.

Παράλληλα όμως, ενισχύοντας τις κλασσικές μεθόδους αυθεντικοποίησης κατά των κακόβουλων επιθέσεων, μπορεί να χειροτερεύει η εμπειρία του χρήστη ή της εφαρμογής. Επιπλέον, όπως αναφέρθηκε πρωτύτερα, με τη ραγδαία ανάπτυξη της τεχνολογίας, δεν θα ήταν καθόλου απίθανο να εξακριβωθούν και νέοι τρόποι παράκαμψης των τωρινών, θεωρητικά ασφαλών τρόπων επαλήθευσης των χρηστών.

Όπως προειπώθηκε, οι κακόβουλες επιθέσεις γίνονται με σκοπό τη μίμηση της ταυτότητας ενός χρήστη. Θεωρητικά, η ισχυρότερη αυθεντικοποίηση θα γινόταν μόνο με ένα χαρακτηριστικό ή στοιχείο το οποίο δε θα μπορούσε να μιμηθεί από κανέναν άλλον άνθρωπο. Εδώ έρχεται και ενσωματώνεται η επιστήμη της βιομετρίας, καθώς κάτι τέτοιο θα αποτελούσε μόνο ένα βιομετρικό χαρακτηριστικό.

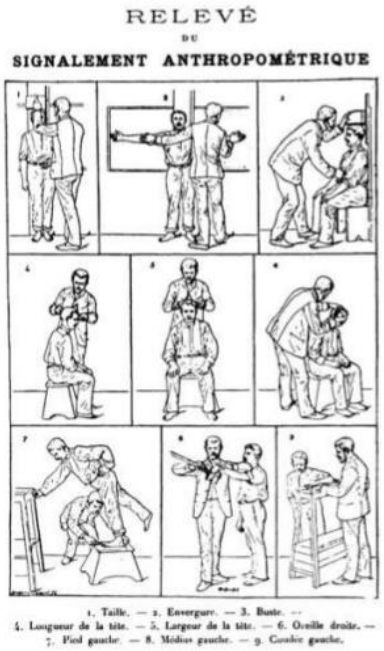
Η Βιομετρία, (biometry ή biometrics), είναι μια εξειδικευμένη επιστήμη, που το αντικείμενο της έρευνάς της είναι η ανάλυση των βιολογικών στοιχείων μέσω δικών της στατιστικών και μαθηματικών μεθόδων.[1] Κάθε άτομο κατέχει μοναδικά χαρακτηριστικά, οπότε η βιομετρία με σύνθετες μεθόδους και διαδικασίες μπορεί να εξακριβώσει και να πιστοποιήσει την ταυτότητα του. Ο λόγος δημιουργίας αυτών των μεθόδων οφείλεται στο γεγονός ότι οι παραδοσιακές μέθοδοι ταυτοποίησης δεν ήταν πια επαρκείς.[2]

1.2 Ιστορική αναδρομή

Η χρήση της Βιομετρίας δεν είναι ένα φαινόμενο της μοντέρνας εποχής. Είχε ξεκινήσει από τους αρχαίους χρόνους. Ωστόσο, η πρώτη καταγραφή έγινε τον 19ο αιώνα όταν χρησιμοποιήθηκε ως μέσο ταυτοποίησης των ατόμων για τη μη εξαπάτηση του συστήματος πληρωμής. Πιο αναλυτικά, παρουσιάζεται παρακάτω μία ιστορική αναδρομή των σημαντικότερων εξελίξεων των βιομετρικών συστημάτων μέχρι σήμερα. [3] [4]

Ένα από τα παλαιότερα καταγεγραμμένα συμβάντα αξιοποίησης των βιομετρικών χαρακτηριστικών των ανθρώπων ήταν στην αρχαία Αίγυπτο [13], κατά τη διάρκεια κατασκευής της πυραμίδας Khufu. Συλλέγονταν βιομετρικές πληροφορίες των ανθρώπων, ώστε να εξασφαλίζεται η μη εξαπάτηση του συστήματος πληρωμής. Η σύγκριση γινόταν με βάση το ύψος, το βάρος, σημάδια και άλλα συμπεριφορικά χαρακτηριστικά. Το ίδιο έγινε και το 1985 όταν ο William Herschel, απεικόνιζε τα αποτυπώματα χεριών κάθε εργάτη στο πίσω μέρος των συμβολαίων τους, έτσι ώστε να αποτρέπει την τυχόν απόπειρα τρίτων, να παραλαμβάνουν τον μισθό των δικαιούχων αντί αυτών, την ημέρα της πληρωμής.

Το 1870 ο Alphonse Bertillon ανέπτυξε το σύστημα "Bertillonage" ή αλλιώς την ανθρωπομετρία, η οποία είναι μία μέθοδος αναγνώρισης τρίτων βασισμένη σε λεπτομερείς μετρήσεις του σώματος, περιγραφών και φωτογραφιών. Αργότερα, ο ίδιος ως προϊστάμενος του Αστυνομικού Τμήματος στο Παρίσι, χρησιμοποίησε έναν αριθμό βιομετρικών μετρήσεων του συστήματος που επινόησε, για την εξακρίβωση των εγκληματιών ανάμεσα σε άλλους.



Bertillonage metrics

1. *Height*
2. *Stretch: Length of body from left shoulder to right middle finger when arm is raised*
3. *Bust: Length of torso from head to seat, taken when seated*
4. *Length of head: Crown to forehead*
5. *Width of head: Temple to temple*
6. *Length of right ear*
7. *Length of left foot*
8. *Length of left middle finger*
9. *Length of left cubit: Elbow to tip of middle finger*
10. *Width of cheeks*

Εικόνα 1.1: Βιομετρικά χαρακτηριστικά του συστήματος Bertillonage

Όμως λόγω της σημαντικής επινόησης του Francis Galton, η μέθοδος του επισκιάστηκε. Το σύστημα "Bertillonage" κατέρρευσε οριστικά όταν δύο άντρες, που αργότερα αποδείχθηκε ότι ήταν αδέρφια, καταδικάστηκαν στο αναμορφωτήριο γιατί είχαν σχεδόν τις ίδιες μετρήσεις.

Το 1893 ο Francis Galton εισήγαγε για πρώτη φορά τη βιομετρική ταυτοποίηση με τη χρήση δακτυλικών αποτυπωμάτων, η οποία χρησιμοποιείται μέχρι και σήμερα. Με τη μέθοδο του Galton αποδείχθηκε επίσης πως τα αποτυπώματα των δακτύλων διαφέρουν ακόμα και μεταξύ ομοζυγωτικών διδύμων.

Τη δεκαετία του 1960 η αναγνώριση προσώπου γίνεται ημιαυτόματη, χάρη στον Woodrow W. Bledsoe.[3] Το σύστημα που εφηύρε μπορούσε να αναγνωρίσει τα μάτια, τη μύτη, το στόμα και τα

αυτιά σε φωτογραφίες ανθρώπων. Αργότερα χρησιμοποιήθηκαν και άλλα χαρακτηριστικά, όπως το πάχος των χειλιών και το χρώμα των μαλλιών για να γίνει η διαδικασία πιο αυτοματοποιημένη. Μέχρι τα μέσα της δεκαετίας του 1990, η τεχνολογία της αναγνώρισης προσώπου εξελίχθηκε τόσο πολύ, που ήταν εφικτή πλέον η αναγνώριση σε πραγματικό χρόνο.

Σήμερα, αν και η βιομετρία έχει κάνει τεράστια βήματα, έχει ακόμα πολλά περιθώρια εξέλιξης. Τη μεγαλύτερη ανάπτυξη μέχρι τώρα, έχουν τα συστήματα που αφορούν το πρόσωπο, το δακτυλικό αποτύπωμα και την ίριδα του ματιού. Αυτά αποτελούν πλέον και τα πιο αποδεκτά βιομετρικά χαρακτηριστικά. Καθώς οι βιομετρικές τεχνολογίες εξελίσσονται, έχει επιτευχθεί και η συγχώνευση τους με την επιστήμη της τεχνητής νοημοσύνης. Ο στόχος της συγχώνευσης αυτών των δύο πεδίων, είναι η κατασκευή βιομετρικών συσκευών και συστημάτων που θα μπορούν, μέσω των βιομετρικών στοιχείων, να μάθουν και να προσαρμοστούν στους χρήστες τους, δημιουργώντας έτσι ένα καλύτερο και πιο αξιόπιστο αποτέλεσμα χρήσης των βιομετρικών μεθόδων αναγνώρισης.[4]

1.3 Βιομετρικά στη σύγχρονη αγορά

Η αναγνώριση των ανθρώπων με τη μέτρηση αυτών των μοναδικών βιολογικών, ανατομικών ή άλλων χαρακτηριστικών συμπεριφοράς, οδήγησαν σε μια συγκεκριμένη ερευνητική περιοχή που ονομάζεται βιομετρική αναγνώριση.[7] Οι βιομετρικές τεχνολογίες παρέχουν έναν ισχυρό μηχανισμό ελέγχου ταυτότητας ή αυθεντικοποίησης και βρίσκονται υπό συνεχή ανάπτυξη. Η διάδοσή τους υποστηρίζεται κυρίως από κυβερνήσεις, υπηρεσίες, τόσο εγκληματολογικές όσο και επιβολής του νόμου, με στόχο τη βελτίωση της δημόσιας ασφάλειας. Γενικά η βιομετρική αυθεντικοποίηση και ταυτοποίηση δεν βελτιώνει άμεσα την ασφάλεια αλλά δρα αποτρεπτικά για παράνομες δραστηριότητες.

Πιο συγκεκριμένα, η χρήση των βιομετρικών μπορεί να ταξινομηθεί σε αυτές τις κατηγορίες: [11][15]

- **Επιβολή του νόμου και ιατροδικαστικές εφαρμογές**: Η επιβολή του νόμου μέσω της βιομετρίας αναφέρεται σε εφαρμογές βιομετρικών συστημάτων που αφορούν λύσεις βασισμένες σε ειδικά αναγνωριστικά εγκληματιών, όπως είναι το σύστημα αναγνώρισης AFIS, όπου μπορεί να αποθηκεύει, να αναζητεί και να ανακτά, εικόνες δακτυλικών αποτυπωμάτων και αρχεία άλλων μοτίβων. Σήμερα, τα αυτοματοποιημένα συστήματα βιομετρικής αναγνώρισης (ABIS) μπορούν να δημιουργήσουν και να αποθηκεύσουν βιομετρικές πληροφορίες που ταιριάζουν με τα βιομετρικά πρότυπα προσώπου, το δάχτυλο και την ίριδα. Τέλος αξιοποιώντας τα παραπάνω, η διαδικασία αναγνώρισης θυμάτων θα γινόταν εξαιρετικά πιο εύκολη σε συγκεκριμένες περιπτώσεις.

- **Στρατός:** Ο στρατός των Ηνωμένων Πολιτειών συλλέγει πρόσωπα, ίριδες, δακτυλικά αποτυπώματα και δεδομένα DNA σε ένα βιομετρικό σύστημα αναγνώρισης από τον Ιανουάριο του 2009. Το βιομετρικό πρόγραμμα ξεκίνησε το 2004, αρχικά συλλέγοντας δακτυλικά αποτυπώματα. Υπεύθυνος για το εγχείρημα ήταν ο Οργανισμός Άμυνας Ιατροδικαστικής και Βιομετρίας (DFBA).

- **Σύνορα, ταξίδια, έλεγχος μετανάστευσης**: Το ηλεκτρονικό διαβατήριο (e-passport) είναι ένα πλέον οικείο βιομετρικό ταξιδιωτικό έγγραφο. Η δεύτερη γενιά αυτών των εγγράφων, επίσης γνωστά ως βιομετρικά διαβατήρια, περιλαμβάνουν δύο δακτυλικά αποτυπώματα και μια φωτογραφία διαβατηρίου. Το έγγραφο αυτό διευκολύνει τη διέλευση των συνόρων, χάρη στην

εύκολη διαδικασία σάρωσης των δακτυλικών αποτυπωμάτων ή της φωτογραφίας μέσω ειδικών σαρωτών.

- **Αναγνώριση πολιτών:** Οι βάσεις δεδομένων AFIS (Αυτόματο Σύστημα Αναγνώρισης Δακτυλικών Αποτυπωμάτων), διασφαλίζουν την ακεραιότητα της ταυτότητας των πολιτών, ανάμεσα στον υπόλοιπο πληθυσμό, με αξιόπιστο, γρήγορο και αυτοματοποιημένο τρόπο. Η ίδια αναγνώριση μπορεί να γίνεται και με συνδυασμό των χαρακτηριστικών προσώπου ή της ίριδας. Επιπλέον, η αναγνώριση πολιτών μπορεί να φανεί χρήσιμη και στις εκλογές για την ταυτοποίηση του ατόμου που ψηφίζει.

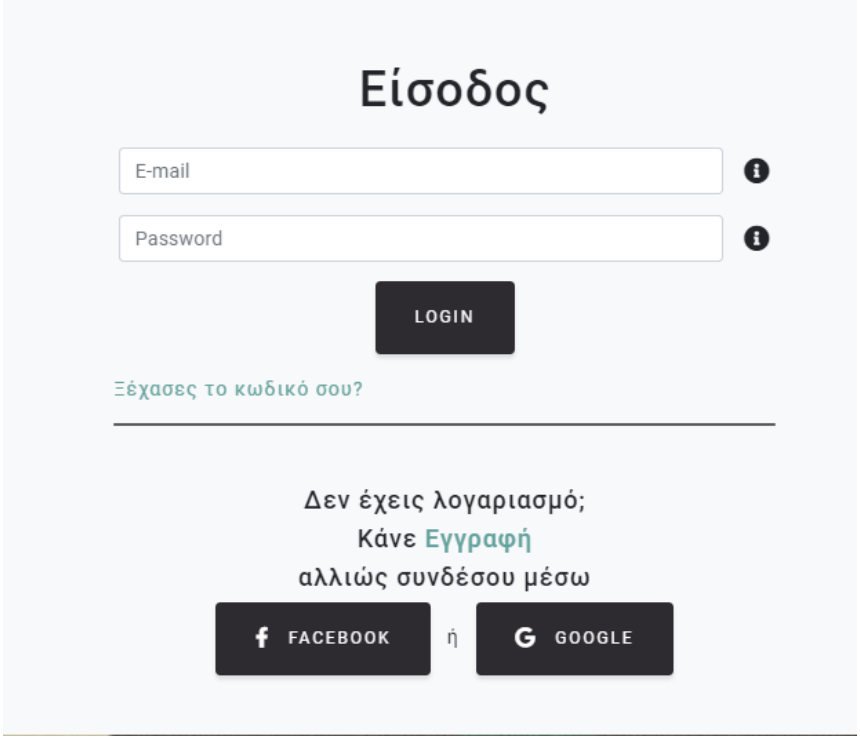
- **Υγεία:** Υπάρχουν εφαρμογές ευρέως διαδεδομένες σε χώρες της Ευρώπης, της Μέσης Ανατολής και στην Αφρική για προγράμματα ασφάλισης υγείας. Με τη χρήση βιομετρικών ταυτοτήτων, τα δακτυλικά αποτυπώματα, που αποτυπώνονται σε αυτές, χρησιμοποιούνται για να επιβεβαιώσουν την ταυτότητα του πολίτη, πριν από την πρόσβαση σε υπηρεσίες υγειονομικής περίθαλψης.

- **Εμπορικές εφαρμογές**: Οι εφαρμογές e-banking αποτελούν ένα χαρακτηριστικό παράδειγμα αυτής της κατηγορίας. Συνήθως γίνεται χρήση των δακτυλικών αποτυπωμάτων για πρόσβαση στις εφαρμογές διαχείρισης των τραπεζικών λογαριασμών και καρτών.

- **Βιομετρικά συστήματα**: Τα βιομετρικά συστήματα ελέγχου πρόσβασης και αυθεντικοποίησης, συμβάλλουν στον αποκλεισμό πρόσβασης μη εξουσιοδοτημένων ατόμων σε μη δημόσια και ευαίσθητα δεδομένα ή υπηρεσίες. Η δράση τους εφαρμόζεται είτε σε φυσικές εγκαταστάσεις ή σε υπολογιστικά συστήματα και δίκτυα, που αφορούν και το θέμα αυτής της εργασίας. Στην πληροφορική, ο βιομετρικός έλεγχος πρόσβασης μπορεί να αποτελέσει ένα συμπληρωματικό παράγοντα ελέγχου ταυτότητας χρήστη ονόματι **2FA** (Two-Factor-Authentication). Σε αντίθεση με τους κωδικούς πρόσβασης ή τις κάρτες πρόσβασης, που βασίζονται σε δεδομένα που μπορούν να ξεχαστούν ή να χαθούν, ο βιομετρικός έλεγχος ταυτότητας βασίζεται στο ποιοι είναι οι άνθρωποι (και όχι τι έχουν). Χαρακτηριστικό παράδειγμα των παραπάνω αποτελούν τα κινητά smartphones, τα οποία αποτελούν και αυτά ένα είδος υπολογιστικού συστήματος. Σε πάρα πολλές συσκευές έχει πλέον ενσωματωθεί η βιομετρική αυθεντικοποίηση μέσω προσώπου, δακτυλικών αποτυπωμάτων, ακόμα και μέσω ίριδας.

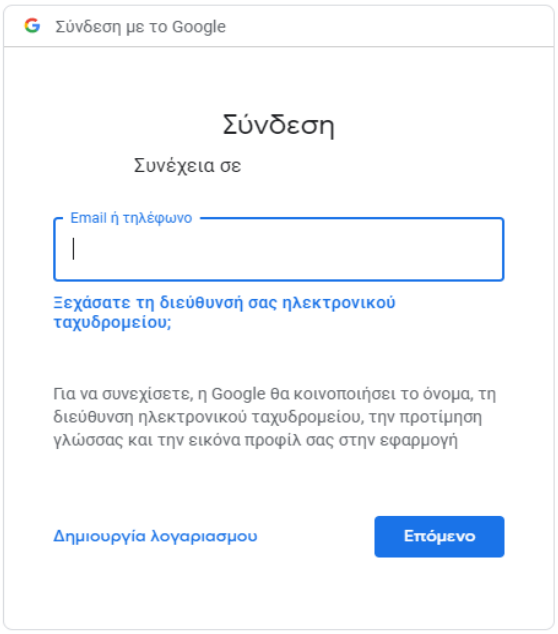
1.4 Στόχοι της εργασίας

Πολλές εταιρείες κολοσσοί, όπως η Google, το Facebook και η Apple, έχουν πλέον δημιουργήσει τους δικούς τους διακομιστές (servers) αυθεντικοποίησης, των οποίων οι υπηρεσίες χρησιμοποιούνται από άλλες εφαρμογές ή οργανισμούς. Μία από αυτές τις υπηρεσίες είναι η παροχή της λειτουργικότητας αυθεντικοποίησης. Δηλαδή ένας χρήστης μπορεί να αυθεντικοποιηθεί, πραγματοποιώντας τη διαδικασία εισόδου που παρέχει η λειτουργικότητα αυτή και να συνδεθεί στην εφαρμογή που χρησιμοποιεί. Χρησιμοποιώντας συγκεκριμένα πρωτόκολλα αυθεντικοποίησης ή εξουσιοδότησης, τα οποία θα αναφερθούν παρακάτω, οι υπηρεσίες που παρέχουν αυτή τη λειτουργικότητα ταυτοποιούν χρήστες που είναι ήδη εγγεγραμμένοι στη βάση δεδομένων τους. Με αυτό το τρόπο, τους παρέχεται πρόσβαση στο περιεχόμενο της εξωτερικής εφαρμογής. Κοινώς, χρησιμοποιώντας αυτή τη μέθοδο, η εξωτερική εφαρμογή δεν είναι απαραίτητο να έχει υλοποιημένο το δικό της μηχανισμό αυθεντικοποίησης, αλλά ο κάθε χρήστης μπορεί να επαληθεύει τη ταυτότητα του μέσω μιας έμπιστης υπηρεσίας, όπως είναι αυτές που αναφέρθηκαν πρωτύτερα.



Εικόνα 1.2: Είσοδος σε εφαρμογή με email και κωδικό πρόσβασης

Ένα τυπικό παράδειγμα αυθεντικοποίησης ενός χρήστη μέσω της google θα ήταν το παραπάνω. Ο χρήστης έχοντας ήδη ένα λογαριασμό στην υπηρεσία της google, επιθυμεί να πραγματοποιήσει σύνδεση σε μια εξωτερική εφαρμογή μέσω της λειτουργικότητας αυθεντικοποίησης που του παρέχει αυτή. Πατώντας το κουμπί για σύνδεση μέσω google, ο χρήστης αυτόματα ανακατευθύνεται στη σελίδα αυθεντικοποίησης της, η οποία μοιάζει κάπως έτσι:



Εικόνα 1.3: Επαλήθευση ταυτότητας μέσω της υπηρεσίας Google

Έπειτα ο χρήστης πρέπει να συμπληρώσει το email και τον κωδικό πρόσβασης του λογαριασμού του στη Google. Εφόσον αυτά τα δύο πεδία είναι σωστά, ο χρήστης αυθεντικοποιείται από τη google και ανακατευθύνεται πίσω στη εφαρμογή, δίνοντας του πρόσβαση σε πηγές οι οποίες δεν θα ήταν προσβάσιμες σε άλλους χρήστες πέρα από τον συγκεκριμένο.

Παρατηρείται όμως, πως, εκτός από κωδικό πρόσβασης, δεν χρησιμοποιείται κάποιο άλλο αναγνωριστικό για την επαλήθευση. Δηλαδή, σε περίπτωση που κάποιος κακόβουλος χρήστης, γνωρίζοντας το email και ανακαλύπτοντας το κωδικό πρόσβασής του, θα μπορούσε να προσπεράσει το 'φράγμα' προστασίας και τα αποκτήσει πρόσβαση στα δεδομένα του αληθινού χρήστη, μιμούμενος τη ταυτότητα του. Υπάρχουν κάποιες περιπτώσεις όπου γίνεται χρήση **2FA (Two Factor Authentication)** μέσω **OTP** (One-time password), αλλά τις περισσότερες φορές, μόνο με τη συναίνεση του ίδιου του χρήστη. Σε αυτό το σημείο τίθεται το εξής ερώτημα: λαμβάνοντας υπόψη τη συνεχή και αυξανόμενη ανάγκη για ασφάλεια και γνωρίζοντας τα πλεονεκτήματα που παρέχει η αναγνώριση με βιομετρικά χαρακτηριστικά, πόσο πιο χρήσιμο και αξιόπιστο θα ήταν, αντί μόνο για κωδικό πρόσβασης ή OTP, να γίνεται μία δεύτερη αυθεντικοποίηση του χρήστη μέσω των βιομετρικών του χαρακτηριστικών; Ή πιο συγκεκριμένα, μέσω των χαρακτηριστικών του προσώπου του;

Στόχος τη παρούσας εργασίας λοιπόν, είναι η δημιουργία μιας Web υπηρεσίας που θα παρέχει τη δυνατότητα αυθεντικοποίησης χρηστών της από εφαρμογές τρίτων, χρησιμοποιώντας όμως και βιομετρική αναγνώριση προσώπου ως ένα ακόμα βήμα επιτυχούς επαλήθευσης.

Το αποτέλεσμα της πτυχιακής αυτής είναι η δημιουργία της κύριας δομής μιας εξωτερικής υπηρεσίας αυθεντικοποίησης με βιομετρικό έλεγχο προσώπου. Θέτει τη βάση για μελλοντικές επεκτάσεις και ενσωματώσεις της βιομετρικής αυθεντικοποίησης σε περισσότερες υπηρεσίες ή εφαρμογές, αλλά και της γενικότερης χρήσης των, αμφιλεγόμενων ακόμα και σήμερα, βιομετρικών χαρακτηριστικών του ανθρώπου.

1.5 Διάρθρωση της πτυχιακής εργασίας

Στο 2ο κεφάλαιο της εργασίας, γίνεται μια εισαγωγή στο γενικότερο θέμα της αυθεντικοποίησης. Αναλύονται τα επίπεδα αυθεντικοποίησης που θα πρέπει να τηρηθούν από μια υπηρεσία συγκριτικά με το επίπεδο εμπιστοσύνης που θα παρέχει αυτή. Επιπλέον, γίνεται μια διάκριση ανάμεσα στη αυθεντικοποίηση και στη ταυτοποίηση ενός χρήστη, καθώς αναλύονται οι διαφορές αυτών των δύο εννοιών.

Το 3ο κεφάλαιο αναλύει τη διαλειτουργικότητα. Εξηγείται με ορισμούς και έννοιες για το πως αυτή προκύπτει μεταξύ διαφορετικών οργανισμών και υπηρεσιών. Στη συνέχεια, γίνεται κατανοητό για το πως σύγχρονα πρωτόκολλα χρησιμοποιούν την γενική έννοια της διαλειτουργικότητας και πως περιγράφεται ο τρόπος λειτουργίας του πρωτοκόλλου εξουσιοδότησης OAuth 2.0 που θα χρησιμοποιηθεί και στη πιλοτική εφαρμογή.

Το 4ο κεφάλαιο αναφέρεται στη βιομετρική αυθεντικοποίηση. Αναλύονται όλα τα βιομετρικά χαρακτηριστικά και αναφέρονται τα πλεονεκτήματα χρήσης των βιομετρικών του προσώπου. Ύστερα, περιγράφεται η λειτουργία και η διαδικασία ανίχνευσης και σύγκρισης των βιομετρικών στοιχείων του προσώπου.

Στο 5ο κεφάλαιο ακολουθεί η περιγραφή ενός βασικού βιομετρικού συστήματος και στη συνέχεια διευκρινίζονται όλες οι αποδεκτές αρχιτεκτονικές βιομετρικών συστημάτων μέχρι σήμερα. Τέλος

ακολουθεί το συμπέρασμα για την αρχιτεκτονική βιομετρικής αυθεντικοποίησης που θα χρησιμοποιηθεί στο συγκεκριμένο Project.

Το 6ο κεφάλαιο κάνει αναφορά στις τεχνολογίες As A Service (aaS) και γίνεται επισήμανση της χρησιμότητάς τους στο παγκόσμιο δίκτυο. Αναλύεται μια επέκταση του μοντέλου aaS, ονόματι Biometric as a Service (BaaS), πως η πιλοτική εφαρμογή θα αποτελέσει μια τέτοια υπηρεσία και ποιο βιομετρικό αρχιτεκτονικό μοντέλο του κεφαλαίου 5 επιλέχθηκε για να επιτευχθεί η υλοποίηση αυτή.

Στο 7ο κεφάλαιο γίνεται επεξήγηση της λειτουργίας της εφαρμογής βήμα προς βήμα με παραδείγματα και εικόνες. Ακολουθείται λεπτομερής περιγραφή κάθε βήματος, ώστε να αποτελεί κατανοητή όλη η ροή και η χρησιμότητα της εφαρμογής.

Στο τελευταίο και ουσιαστικό 8ο κεφάλαιο αναλύονται τα συμπεράσματα της πτυχιακής εργασίας και παρουσιάζονται προτάσεις βελτίωσής της.

Τέλος, ακολουθεί η παράθεση των βιβλιογραφικών πηγών που αξιοποιήθηκαν.

1.6 Επίλογος

Σε αυτό το κεφάλαιο, έγινε εισαγωγή στη χρησιμότητα της επιστήμης της βιομετρίας, κάνοντας μια μικρή ιστορική αναδρομή και αναφέροντας τη χρησιμότητα των βιομετρικών στη σύγχρονη αγορά. Επίσης διατυπώνεται ο στόχος της παρούσας εργασίας με ένα παράδειγμα και παρατίθεται το περιεχόμενο των επόμενων κεφαλαίων. Το κεφάλαιο αυτό, ουσιαστικά αποτελεί το πρώτο σκαλοπάτι για τη κατανόηση των ενοτήτων και των εννοιών που παρουσιάζονται στη συνέχεια.

Κεφάλαιο 2ο: Αυθεντικοποίηση

2.1 Εισαγωγή

Η αυθεντικοποίηση αποτελεί τη διαδικασία πιστοποίησης και επιβεβαίωσης της ταυτότητας των χρηστών μιας πλατφόρμας ή μιας υπηρεσίας, η οποία βασίζεται στα διαπιστευτήρια που κατέχει ο καθένας ή αυτά που έχει ορίσει η ίδια η υπηρεσία.

Συγκεκριμένα κατά τη διαδικασία αυθεντικοποίησης, αναγνωρίζεται και επιβεβαιώνεται η ορθότητα της ταυτότητας ενός χρήστη ή κάποιων χαρακτηριστικών του. Σε καμία περίπτωση δε θα πρέπει η αυθεντικοποίηση ενός χρήστη να συγχέεται με την παροχή εξουσιοδότησης (authorization). Θα γίνει επεξήγηση των διαφορών αυθεντικοποίησης και εξουσιοδότησης (authentication vs authorization) σε παρακάτω κεφάλαιο. [6]

Στο κεφάλαιο αυτό παρουσιάζονται οι εναλλακτικοί μηχανισμοί και μέθοδοι αυθεντικοποίησης που είναι ρεαλιστικά εφικτό να αξιοποιηθούν. Θα πρέπει να σημειωθεί, ότι η επιλογή κάποιας συγκεκριμένης μεθόδου αυθεντικοποίησης δεν αποτελεί αντικείμενο του κεφαλαίου, καθώς εξαρτάται από το επίπεδο εμπιστοσύνης στο οποίο έχει ενταχθεί η υπηρεσία. Επισημαίνεται όμως, ότι όσο μεγαλύτερο είναι το επίπεδο εμπιστοσύνης στο οποίο εντάσσεται μία υπηρεσία, τόσο ισχυρότερος μηχανισμός αυθεντικοποίησης απαιτείται.

2.2 Μηχανισμοί Αυθεντικοποίησης

Τα συστήματα που χρησιμοποιούν μηχανισμούς αυθεντικοποίησης κατηγοριοποιούνται συνήθως, με βάση τη μέθοδο που χρησιμοποιείται για την εξακρίβωση της ταυτότητας του κάθε χρήστη. Οι μηχανισμοί αυτοί διαχωρίζονται με βάση τα εξής χαρακτηριστικά: [5]

- Κάτι που γνωρίζει ο χρήστης, για παράδειγμα ένας κωδικός πρόσβασης.
- Κάτι που κατέχει ο χρήστης, για παράδειγμα ενα USB stick.
- Κάποιο χαρακτηριστικό γνώρισμα, για παράδειγμα βιομετρικά χαρακτηριστικά.
- Συνδυασμός των παραπάνω.

Οι μηχανισμοί αυθεντικοποίησης, που χρησιμοποιούν κρυπτογραφία, ανεξάρτητα από τα χαρακτηριστικά που υιοθετούν, αξιοποιούν δύο τύπους κλειδιών:

- Μυστικά κλειδιά: Σε αυτά συμπεριλαμβάνονται τα συνθηματικά, οι κωδικοί και τα συμμετρικά κλειδιά.
- Ασύμμετρα κλειδιά: Σε αυτά συμπεριλαμβάνονται ζεύγη κλειδιών, από τα οποία το ένα είναι δημόσια γνωστό (δημόσιο κλειδί), ενώ το άλλο παραμένει μυστικό (ιδιωτικό κλειδί).

Τα συστήματα αυθεντικοποίησης μπορούν να χαρακτηριστούν ως μονοδιάστατα ή πολυδιάστατα. Αυτοί οι χαρακτηρισμοί προκύπτουν ανάλογα με το ποσό των χαρακτηριστικών που χρησιμοποιούν. Ο αριθμός των χαρακτηριστικών αυτών προκύπτει από το επίπεδο βεβαιότητας που θέλουν να επιτύχουν. Με τη λέξη βεβαιότητα εννοείται το πόσο σίγουρο είναι πως η ηλεκτρονική ταυτότητα ενός χρήστη ανήκει όντως σε αυτόν.

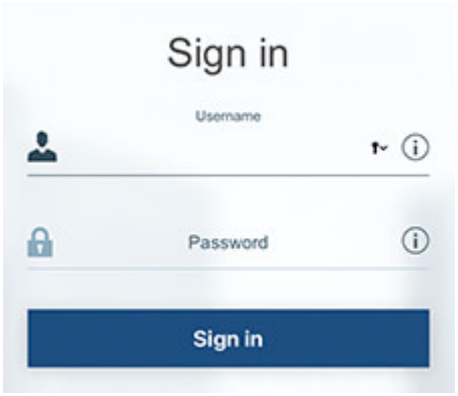
2.3 Διακριτικά Αυθεντικοποίησης

Προκειμένου να εξακριβωθεί η ορθότητα της ηλεκτρονικής ταυτότητας των χρηστών ενός συστήματος, γίνεται χρήση των διακριτικών αυθεντικοποίησης, τα οποία είναι πληροφορίες που μόνο ο ίδιος ο χρήστης γνωρίζει. Με αυτό το τρόπο γίνεται η επιβεβαίωση της ταυτότητάς του. Ανάλογα με το επίπεδο ασφάλειας που θέλει να εφαρμόσει μια υπηρεσία, υιοθετείται και ο ανάλογος συνδυασμός χαρακτηριστικών και κλειδιών αυθεντικοποίησης. [5]

2.3.1 Συνθηματικά

Τα συνθηματικά ή αλλιώς κωδικοί πρόσβασης, αποτελούν πλέον τον πιο κοινό τρόπο ταυτοποίησης της ηλεκτρονικής ταυτότητας χρήστη. Συνήθως είναι ένας συνδυασμός χαρακτήρων, συμβόλων και γραμμάτων, τον οποίο μόνο ο χρήστης που τον σκέφτηκε πρέπει να ξέρει. Δεν πρέπει ο κωδικός αυτός να γνωστοποιείται σε άλλους χρήστες.

Παρακάτω βλέπουμε ένα τυπικό παράδειγμα:



Εικόνα 2.1: Παράδειγμα αυθεντικοποίησης με συνθηματικό

2.3.2 Διακριτικά συνθηματικών μιας χρήσης (one time password tokens)

Τα συνθηματικά μιας χρήσης, σε αντίθεση με τους κωδικούς πρόσβασης, δεν απαιτούν την απομνημόνευσή τους από τον χρήστη, καθώς χρησιμοποιούνται μόνο μία φορά. Αυτό σημαίνει πως αν ο χρήστης επιχειρούσε την αυθεντικοποίηση του με ένα ήδη χρησιμοποιημένο συνθηματικό, αυτή θα αποτύγχανε. Η παραγωγή αυτών των συνθηματικών προέρχεται από συγκεκριμένους αλγορίθμους κρυπτογράφησης και όχι από τον χρήστη αυτόν καθ' αυτόν.

2.3.3 Διακριτικά Χαλαρής Αποθήκευσης (soft tokens)

Τα διακριτικά χαλαρής αποθήκευσης αναφέρονται σε μυστικά κλειδιά, τα οποία αποθηκεύονται σε κάποιο μέσο αποθήκευσης. Τέτοια μέσα θεωρούνται φυσικά αντικείμενα όπως είναι ένα USB stick, ένα CD κ.α. Η αποθηκευμένη πληροφορία των κλειδιών μέσα σε αυτά τα μέσα αποθήκευσης είναι κρυπτογραφημένη. Η αποκρυπτογράφηση της γίνεται μόνο με τη χρήση ενός κωδικού.

2.3.4 Διακριτικά Υλικού – Σκληρής Αποθήκευσης (hard tokens)

Τα διακριτικά υλικού σκληρής αποθήκευσης αναφέρονται σε συσκευές υλικού, οι οποίες αποθηκεύουν τα απαιτούμενα μυστικά κλειδιά και προσφέρουν tamper proof προστασία. Αυτό γίνεται καθώς οι διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης γίνονται μέσα στην ίδια τη συσκευή. Έτσι δε

γίνεται να διαρρεύσουν τα μυστικά κλειδιά για αποκρυπτογράφηση. Τα κλειδιά αυτά ενεργοποιούνται με τη χρήση κάποιου κωδικού.

2.4 Επίπεδα Εμπιστοσύνης

Τα δεδομένα που αξιοποιούνται από υπηρεσίες μπορούν να κατηγοριοποιηθούν με βάση το βαθμό κρισιμότητάς τους σε: [5]

- Προσωπικά δεδομένα, δηλαδή πληροφορίες που αφορούν το ίδιο το πρόσωπο.
- Ευαίσθητα δεδομένα, όπως είναι τα στοιχεία καρτών τράπεζας ή οτιδήποτε άλλο μπορεί να θεωρηθεί σημαντικό που η τυχόν διαρροή του θα προκαλούσε καταστροφικές συνέπειες.
- Οικονομικά δεδομένα: Αυτά θεωρούνται ευαίσθητα δεδομένα εάν σχετίζονται με παροχές που εμπίπτουν στην κοινωνική πρόνοια. Απεναντίας, εάν αφορούν φυσικά πρόσωπα, θεωρούνται προσωπικά δεδομένα.

Τα παραπάνω δεδομένα μπορεί να θεωρούνται κρίσιμα, ανάλογα με τις επιπτώσεις που θα υπάρξουν σε περίπτωση που αυτά διαρρεύσουν. Μια τέτοια διαρροή θα μπορούσε να έχει επίπτωση: [5]

- για το χρήστη ή / και το φορέα που προσφέρει την υπηρεσία, λόγω της αποκάλυψης ή «παράνομης και αθέμιτης» χρήσης των δεδομένων,
- στην ιδιωτικότητα του ατόμου.

Για την ανάλυση των επιπέδων εμπιστοσύνης λαμβάνονται υπόψη τα παραπάνω κριτήρια. Όσο πιο κρίσιμα θεωρούνται τα δεδομένα μίας υπηρεσίας, τόσο μεγαλύτερο επίπεδο εμπιστοσύνης απαιτείται.

Το επίπεδο εμπιστοσύνης για κάθε ηλεκτρονική υπηρεσία διαμορφώνεται ανάλογα με την αξία των συναλλαγών, την κρισιμότητα των δεδομένων που χρησιμοποιούνται, των άμεσων ή έμμεσων επιπτώσεων που μπορεί να προκύψουν από την εκδήλωση επιθέσεων, καθώς επίσης και από την αντίστοιχη επιρροή του θεσμικού πλαισίου. Λαμβάνοντας υπόψη τα παραπάνω, τα επίπεδα εμπιστοσύνης που διαμορφώνονται μπορεί να είναι τα εξής: [5]

2.4.1 Επίπεδο Εμπιστοσύνης 0 (EEM0)

Σε αυτό το επίπεδο ανήκουν οι υπηρεσίες που χρησιμοποιούν πληροφορίες οι οποίες είναι δημόσια διαθέσιμες και δεν έχουν στόχο τη χρήση ή ανταλλαγή προσωπικών ή οικονομικών δεδομένων. Η μοναδική απαίτηση των υπηρεσιών αυτών είναι ή συνεχής διαθεσιμότητα των δεδομένων καθώς κάθε άλλη περίπτωση θεωρείται ασήμαντη.

2.4.2 Επίπεδο Εμπιστοσύνης 1 (EEM1)

Στο επίπεδο εμπιστοσύνης 1 απαιτείται μικρός βαθμός βεβαιότητας και ορθότητας καθώς οι υπηρεσίες που το υλοποιούν κάνουν χρήση και ανταλλαγή δεδομένων μικρής κρισιμότητας, όπως είναι το ονοματεπώνυμο, το e-email κλπ. Σε αυτό το επίπεδο υπάρχουν επιπτώσεις διαρροής δεδομένων, τα οποία αν και ελάχιστης σημασίας, είναι καλό να παρθούν μέτρα ασφαλείας για την ελαχιστοποίηση εμφάνισης κάποιας απειλής.

2.4.3 **Επίπεδο Εμπιστοσύνης 2 (EEM2)**

Στο επίπεδο εμπιστοσύνης 2 εντάσσονται υπηρεσίες που απαιτούν ανταλλαγή προσωπικών δεδομένων, τα οποία δεν είναι χαρακτηρισμένα ως ευαίσθητα, όπως για παράδειγμα στοιχεία που αφορούν την οικογενειακή κατάσταση του χρήστη, ημερομηνία γέννησης, φύλο κ.λπ. Οι επιπτώσεις που προκύπτουν από κάποια διαρροή δεδομένων σε κάποια επίθεση, μπορούν να θεωρηθούν σημαντικές κατά κάποιο βαθμό καθώς ενδέχεται δεδομένα χρηστών να χρησιμοποιηθούν για κάποιο ειδικό σκοπό ή από το ευρύ κοινό, χωρίς όμως με τη συγκατάθεση τους.

2.4.4 **Επίπεδο εμπιστοσύνης 3 (EEM3)**

Σε αυτό το επίπεδο ανήκουν υπηρεσίες οι οποίες διαχειρίζονται ευαίσθητα προσωπικά δεδομένα τα οποία είναι σημαντικό να μένουν και να αποθηκεύονται σε ένα ασφαλές από απειλές περιβάλλον. Οι επιπτώσεις που μπορεί να προκληθούν από κάποιο περιστατικό ασφαλείας, είναι μεγάλες και απαιτείται να λαμβάνονται τα απαραίτητα μέτρα προστασίας που θα διασφαλίζουν τον υψηλό βαθμό εμπιστοσύνης για τη ηλεκτρονική ταυτότητα ενός χρήστη.

2.5 Απαιτήσεις Αυθεντικοποίησης

Η στρατηγική αυθεντικοποίησης μιας υπηρεσίας θα πρέπει να σχεδιάζεται με βάση τη βεβαιότητα που απαιτείται για την ορθότητα της ηλεκτρονικής ταυτότητας μίας οντότητας. Για το λόγο αυτό, πριν αποφασιστεί ο συγκεκριμένος μηχανισμός αυθεντικοποίησης για κάποια υπηρεσία, θα πρέπει να καθοριστεί και να αποφασιστεί το επίπεδο εμπιστοσύνης και αυθεντικοποίησης για αυτή.

Με βάση όσα αναφέρθηκαν, τα επίπεδα αυθεντικοποίησης που προκύπτουν είναι τρία και είναι τα παρακάτω [5]:

2.5.1 **Επίπεδο Αυθεντικοποίησης 0 (EA0)**

Σε αυτό το επίπεδο δεν απαιτείται αυθεντικοποίηση του χρήστη καθώς οποιαδήποτε οντότητα αυτού, μπορεί να έχει πρόσβαση στις πληροφορίες, καθώς αυτές θεωρούνται ανοιχτές σε όλους. Συνήθως, τέτοιου τύπου υπηρεσίες είναι όσες παρέχουν δημόσιο υλικό.

2.5.1.1 Απαιτήσεις ασφάλειας

Έτσι ώστε να τηρούνται οι απαιτήσεις ασφάλειας σε αυτό το επίπεδο θα πρέπει να εξασφαλίζεται η ακεραιότητα του παρεχόμενου πληροφοριακού υλικού, όπως και η αυθεντικότητα της υπηρεσίας.

2.5.1.2 Συσχετισμός με επίπεδο εμπιστοσύνης

Το επίπεδο αυθεντικοποίησης 0 σχετίζεται με το επίπεδο εμπιστοσύνης 0, καθώς δεν απαιτείται η επιβεβαίωση της ορθότητας της ηλεκτρονικής ταυτότητας του χρήστη.

2.5.1.3 Προτεινόμενος μηχανισμός αυθεντικοποίησης

δεν απαιτείται μηχανισμός αυθεντικοποίησης.

2.5.2 **Επίπεδο Αυθεντικοποίησης 1 (ΕΑ1)**

Σε αυτό το επίπεδο αυθεντικοποίησης είναι απαραίτητο να εφαρμόζεται και να επιβεβαιώνεται σε μικρό ή σε μεσαίο βαθμό η ορθότητα ενός χρήστη ή μιας οντότητας. Για να έχει πρόσβαση στους πόρους μιας υπηρεσίας που εφαρμόζει αυτό το επίπεδο αυθεντικοποίησης, θα πρέπει κάποιος να είναι εξουσιοδοτημένος χρήστης, ώστε να του δοθεί το δικαίωμα πρόσβασης.

2.5.2.1 Απαιτήσεις ασφάλειας

Αυτό το επίπεδο ασφαλείας απαιτεί την εμπιστευτικότητα των διαπιστευτηρίων και δεδομένων ταυτοποίησης (αναγνωριστικά) του χρήστη, δηλαδή τη τήρηση των κανόνων προστασίας προσωπικών δεδομένων του. Απαιτεί επίσης την ακεραιότητα των αναγνωριστικών, των διαπιστευτηρίων του χρήστη αλλά και των δεδομένων που λαμβάνονται από την υπηρεσία. Τέλος όπως και στο προηγούμενο επίπεδο, η αυθεντικότητα της ίδιας της υπηρεσίας είναι βασική προϋπόθεση

2.5.2.2 Συσχετισμός με επίπεδο εμπιστοσύνης

Το επίπεδο αυθεντικοποίησης 1 σχετίζεται με τα επίπεδα εμπιστοσύνης 1 και 2, καθώς απαιτείται έως και μέτρια βεβαιότητα για την ορθότητα της ηλεκτρονικής ταυτότητας του χρήστη.

2.5.2.3 Προτεινόμενος μηχανισμός αυθεντικοποίησης

Οι μηχανισμοί αυθεντικοποίησης που προτείνονται για το συγκεκριμένο επίπεδο συμπεριλαμβάνουν: συνθηματικά, συνθηματικά μιας χρήσης ή συνδυασμό αυτών.

2.5.3 **Επίπεδο Αυθεντικοποίησης 2 (ΕΑ2)**

Σε αυτό το επίπεδο αυθεντικοποίησης είναι απαραίτητο να εφαρμόζεται και να επιβεβαιώνεται σε υψηλό βαθμό η ορθότητα ενός χρήστη ή μιας οντότητας. Είναι κρίσιμο μόνο οι εξουσιοδοτημένοι χρήστες να έχουν πρόσβαση στους πόρους μιας υπηρεσίας. Συνήθως αυτό το επίπεδο αναφέρεται σε υπηρεσίες που διαχειρίζονται ευαίσθητα δεδομένα ή συναλλαγές τρίτων.

2.5.3.1 Απαιτήσεις ασφάλειας

Στο επίπεδο αυθεντικοποίησης 2 θα πρέπει να διασφαλίζονται κατ' ελάχιστον η εμπιστευτικότητα και η ακεραιότητα των αναγνωριστικών, των διαπιστευτηρίων του χρήστη μαζί με τα δεδομένα που αποστέλλονται και λαμβάνει ο χρήστης από την ηλεκτρονική υπηρεσία. Όπως και στα προηγούμενα επίπεδα, απαιτείται η αυθεντικότητα της υπηρεσίας. Επιπλέον όμως, θα πρέπει να μην γίνεται αποποίηση της αποστολής και της λήψης δεδομένων, να υπάρχουν υπηρεσίες εποπτείας και να γίνεται χρονοσήμανση των ενεργειών των χρηστών.

2.5.3.2 Προτεινόμενος μηχανισμός αυθεντικοποίησης

Η αυθεντικοποίηση στο παρόν επίπεδο είναι απαραίτητο να κάνει χρήση και να αξιοποιεί ψηφιακά πιστοποιητικά (digital certificates) τα οποία θα εκδίδονται από την κατάλληλη Υποδομή Δημοσίου Κλειδιού (PKI) και την Αρχή Χρονοσήμανσης (Time Stamping Authority - TSA). Σε κάθε περίπτωση, αποκλειστικά υπεύθυνος για την τελική επιλογή του τύπου διακριτικών αποθήκευσης είναι πάντοτε ο φορέας παροχής της ηλεκτρονικής υπηρεσίας. Όποια επιλογή και αν τελικά υιοθετηθεί, τα διακριτικά αποθήκευσης θα πρέπει να προστατεύονται από τους αντίστοιχους προσωπικούς κωδικούς του χρήστη.

2.5.3.3 Συσχετισμός με επίπεδο εμπιστοσύνης

Το επίπεδο αυθεντικοποίησης 2 σχετίζεται με το επίπεδο εμπιστοσύνης 3 καθώς απαιτείται υψηλή βεβαιότητα για την ορθότητα της ηλεκτρονικής ταυτότητας του χρήστη.

2.6 Σύνοψη Συσχετισμού Επιπέδων Εμπιστοσύνης & Αυθεντικοποίησης

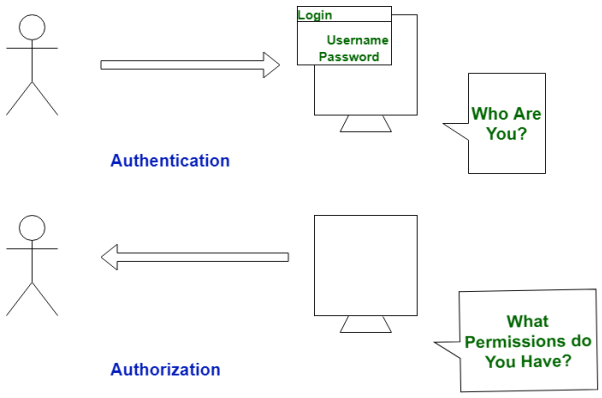
Στον παρακάτω πίνακα συνοψίζεται η συσχέτιση μεταξύ επιπέδων εμπιστοσύνης και επιπέδων αυθεντικοποίησης. Όπως είναι φανερό, η συσχέτιση αυτή δεν αποτελεί ένα προς ένα αντιστοιχία.

Επίπεδο Εμπιστοσύνης	Επίπεδο Αυθεντικοποίησης
EEM0	**EA0**
EEM1, EEM2	**EA1**
EEM3	**EA2**

Πίνακας 2.1: Συσχέτιση Επιπέδου Εμπιστοσύνης & Επιπέδου Αυθεντικοποίησης

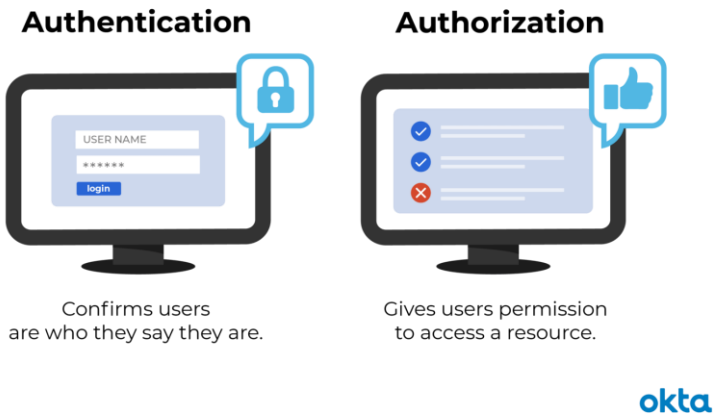
2.7 Αυθεντικοποίηση εναντίον Εξουσιοδότησης

Όπως αναφέρθηκε παραπάνω, η αυθεντικοποίηση (authentication) διαφέρει από την εξουσιοδότηση (authorization) του χρήστη σε μια ηλεκτρονική υπηρεσία. Η αυθεντικοποίηση επιβεβαιώνει ότι οι χρήστες είναι αυτοί που λένε ότι είναι ενώ η εξουσιοδότηση δίνει σε αυτούς τους χρήστες άδεια πρόσβασης σε έναν πόρο. Σε ασφαλή περιβάλλοντα, η εξουσιοδότηση πρέπει πάντα να ακολουθεί της αυθεντικοποίησης .



Εικόνα 2.2: Διαφορά αυθεντικοποίησης και εξουσιοδότησης χρήστη

Οι χρήστες πρέπει πρώτα να αποδείξουν ότι οι ταυτότητές τους είναι γνήσιες προτού οι διαχειριστές του οργανισμού τους παραχωρήσουν πρόσβαση στους πόρους που ζητήθηκαν.



Εικόνα 2.3: Δεύτερο παράδειγμα διαφοράς αυθεντικοποίησης και εξουσιοδότησης χρήστη

Παρακάτω φαίνεται η διαφορά αυθεντικοποίησης και εξουσιοδότησης σε πίνακα:

	Authentication	**Authorization**
1.	Η ταυτότητα των χρηστών ελέγχεται για την παροχή πρόσβασης στο σύστημα.	Η ταυτότητα των χρηστών ελέγχεται για πρόσβαση στους πόρους.
2.	Οι χρήστες επαληθεύονται.	Οι χρήστες επικυρώνονται.
3.	Γίνεται πριν διαδικασία εξουσιοδότησης	Γίνεται μετά τη διαδικασία αυθεντικοποίησης
4.	Συνήθως χρειάζεται τα στοιχεία εισόδου χρήστη	Χρειάζεται προνόμια του χρήστη ή επίπεδα ασφάλειας.
5.	Καθορίζει εάν το άτομο είναι χρήστης ή όχι.	Καθορίζει τις άδειες που έχει ο χρήστης.

Πίνακας 2.2: Παράδειγμα διαφοράς αυθεντικοποίησης και εξουσιοδότησης χρήστη

2.8 Επίλογος

Στο παρόν κεφάλαιο, έγινε επεξήγηση της έννοιας της αυθεντικοποίησης, σε ποιες περιπτώσεις χρησιμοποιείται και με ποιο τρόπο. Αναλύεται η συσχέτιση των επιπέδων αυθεντικοποίησης και επιπέδων εμπιστοσύνης των υπηρεσιών. Τέλος παρουσιάζεται η διαφορά της αυθεντικοποίησης με την έννοια της εξουσιοδότησης. Η διαφορά αυτή είναι σημαντική για τη κατανόηση μέρους του κεφαλαίου 3 που θα παρουσιαστεί παρακάτω.

Κεφάλαιο 3ο: Διαλειτουργική αυθεντικοποίηση

3.1 Εισαγωγή

Οι διαφορετικές και ποικίλες ανάγκες των δημοσίων φορέων, των οργανισμών και των επιχειρήσεων για υποστήριξη από συστήματα πληροφορικής, οι πολλές διαθέσιμες τεχνολογικές λύσεις και σχεδιαστικές επιλογές και το διαφορετικό επίπεδο ψηφιακής οργάνωσης κάθε οργανισμού, οδηγούν στη δημιουργία ειδικών πληροφοριακών συστημάτων, που είναι σχεδιασμένα να καλύπτουν τις ανάγκες ενός οργανισμού, αλλά δεν είναι σίγουρο πως θα εξυπηρετούν και άλλους ταυτόχρονα.

Η λύση στο σοβαρό αυτό πρόβλημα δίνεται με τη θέσπιση διαδικασιών και προτύπων, τα οποία θα εισάγουν έναν κοινό τρόπο επικοινωνίας και εξασφάλισης της σωστής κατανόησης της επικοινωνίας μεταξύ δύο οργανισμών που επιδιώκουν έναν κοινό σκοπό ή λειτουργία. Το σύνολο αυτών των διαδικασιών και προτύπων λοιπόν, διαμορφώνουν την έννοια της διαλειτουργικότητας.

3.2 Διαλειτουργικότητα

Η διαλειτουργικότητα ορίζεται ως η ικανότητα ενός πληροφοριακού συστήματος να επικοινωνεί με το εξωτερικό περιβάλλον του. Αυτό σημαίνει πως προορίζεται για την ανταλλαγή δεδομένων με άλλα συστήματα, την επεξεργασία πληροφοριών που μεταφέρονται και την κατανόηση αυτών αλλά και ενός κώδικα επικοινωνίας μεταξύ τους. Οι πληροφορίες δηλαδή αυτές, θα πρέπει να γίνονται κατανοητές και από τα δύο συστήματα που επικοινωνούν. [25]

Για την επίτευξη της διαλειτουργικότητας ενός συστήματος, πρέπει οι διεπαφές του να είναι πλήρως τεκμηριωμένες και να έχουν δημόσιο χαρακτήρα.

Προϋποθέσεις για την εξασφάλιση διαλειτουργικότητας αποτελούν: [25]

- Η υιοθέτηση ανοιχτών αρχιτεκτονικών, στις οποίες καθορίζεται ο ρόλος κάθε υποσυστήματος. Μια ανοιχτή αρχιτεκτονική ορίζεται από την ελευθερία επιλογών που παρέχει στα συστατικά του. Σε περίπτωση ύπαρξης μιας καλύτερης και αποδοτικότερης λύσης τα συστατικά αυτά εναλλάσσονται. Η αρχιτεκτονική αυτή δε δεσμεύεται από ιδιόκτητες τεχνολογίες ενός κατασκευαστή.

- Η ύπαρξη κοινών και ευρύτερα αποδεκτών προτύπων (Standards). Αυτά περιγράφουν τον τρόπο επικοινωνίας μεταξύ των υποσυστημάτων και τη μορφή των πληροφοριών που ανταλλάσσονται ώστε να υπάρχει μια κοινή συνεννόηση.

- Ο έλεγχος και η έρευνα ποικίλων προϊόντων διαφόρων οργανισμών, με σκοπό τη συμμόρφωσή τους με τα κοινά πρότυπα.

Ένα παράδειγμα τέτοιων προτύπων, που προωθούν τη διαλειτουργικότητα, αποτελούν τα authorization protocols ή αλλιώς τα πρότυπα εξουσιοδότησης και τα authentication protocols ή πρωτόκολλα αυθεντικοποίησης. Τα πρωτόκολλα

αυθεντικοποίησης, επιτρέπουν τη σύνδεση και αυθεντικοποίηση χρηστών μέσω τρίτων εφαρμογών στο πληροφοριακό σύστημα που επιθυμούν. Τα δε πρωτόκολλα εξουσιοδότησης, επιτρέπουν στον χρήστη να εξουσιοδοτεί το πληροφοριακό σύστημα, που χρησιμοποιεί, ώστε να έχει πρόσβαση στις πληροφορίες του χρήστη, από μια τρίτη υπηρεσία. Και τα δύο είδη προτύπων είναι ευρέως διαδεδομένα και χρησιμοποιούνται πλέον από ένα πολύ μεγάλο ποσοστό εφαρμογών και υπηρεσιών. Αυτά κάνουν τη εμπειρία του χρήστη σε μια εφαρμογή πιο ευχάριστη και πιο γρήγορη, καθώς παρέχουν αυτόματες και εύκολες λύσεις, πλέον, κοινώς αποδεκτές από όλους.

Δύο από τα πιο δημοφιλή πρότυπα είναι το **Security assertion markup language (SAML)** και το **Open Authorization (OAuth)**. Το πρώτο αποτελεί πρότυπο αυθεντικοποίησης και το δεύτερο εξουσιοδότησης. Για την υλοποίηση της πιλοτικής εφαρμογής, χρειάζεται η επιλογή ενός τέτοιου προτύπου ώστε να μπορούν οι χρήστες τρίτων, αυτόνομων εφαρμογών να συνδέονται και να αυθεντικοποιούνται, χρησιμοποιώντας τη βιομετρική αυθεντικοποίηση της εφαρμογής που θα τους παρέχεται [26].

Τόσο το OAuth όσο και το SAML είναι πρωτόκολλα που ενθαρρύνουν και τυποποιούν τη διαλειτουργικότητα. Οι άνθρωποι τα χρησιμοποιούν για να αποφύγουν μια συνεχώς διευρυνόμενη λίστα με ονόματα χρήστη και κωδικούς πρόσβασης. Για τους κατόχους εφαρμογών, το OAuth και το SAML επιτρέπουν την εύκολη ενσωμάτωση και τη δυνατότητα διαχείρισης χρηστών. Για τους διαχειριστές, αυτά τα εργαλεία σημαίνουν γρήγορη ενσωμάτωση και κεντρικό έλεγχο ταυτότητας χρηστών.

Το SAML επαληθεύει την ταυτότητα και προσφέρει έλεγχο αυτής. Σε ένα τυπικό περιβάλλον γραφείου, ένας υπάλληλος πρέπει να συνδεθεί για να αποκτήσει πρόσβαση σε οποιοδήποτε μέρος των εσωτερικών πόρων της εταιρείας. Με τον έλεγχο ταυτότητας SAML, ο χρήστης μπορεί να έχει πρόσβαση σε μια ολόκληρη σειρά εργαλείων, συμπεριλαμβανομένου ενός εταιρικού intranet, του Microsoft Office και ενός προγράμματος περιήγησης. Το SAML επιτρέπει στον χρήστη να αξιοποιήσει όλους αυτούς τους πόρους κάτω από μία ψηφιακή υπογραφή. Ακόμα και σε εταιρείες με αυστηρότερη ασφάλεια, το SAML επιτρέπει στον χρήστη να ανοίξει μια πόρτα ή να ξεκλειδώσει μια οθόνη υπολογιστή. Γενικώς, μπορεί να απαιτείται εξουσιοδότηση πριν ο χρήστης επιτραπεί να κάνει οποιαδήποτε άλλη ενέργεια, συμπεριλαμβανομένης, ακόμα και της πρόσβασης σε αρχεία.

Αντιθέτως το OAuth χρησιμοποιείται για την παροχή εξουσιοδότησης από τη μία υπηρεσία στην άλλη, προστατεύοντας ταυτόχρονα το όνομα χρήστη και τον κωδικό πρόσβασης κάποιου. Το OAuth μπορεί να χαρακτηριστεί ως καίρια εξοικονόμηση χρόνου, σε ένα περιβάλλον όπου ο μέσος εργαζόμενος συνδέεται αρκετές φορές την ημέρα σε άλλες εφαρμογές κρίσιμες για την εργασία του. Μερικές φορές, οι υπάλληλοι θέλουν έναν τρόπο μετάβασης από τη μία εφαρμογή στην άλλη χωρίς να συνδέονται ξανά και ξανά. Το OAuth το κάνει δυνατό, καθώς ένας υπάλληλος με ενεργό λογαριασμό Google θα μπορούσε να χρησιμοποιήσει τα ίδια διαπιστευτήρια για να αξιοποιήσει τα δεδομένα που βρίσκονται σε υπηρεσίες όπως το Microsoft 365, το SalesForce και άλλα. Ο εργαζόμενος χρειάζεται όλα αυτά τα διαδικτυακά προγράμματα για να ενεργεί γρήγορα και παραγωγικά. Όμως, όταν απαιτείται να δημιουργήσει και να θυμάται πλήθος διαφορετικών συνόλων ονομάτων χρήστη και κωδικών πρόσβασης, αποτελεί εμπόδιο στη παραγωγικότητα αυτή. Για να αποφευχθεί η δημιουργία πολλών διαφορετικών ζευγών ονομάτων χρήστη και κωδικών πρόσβασης, η είσοδος με χρήση μιας υπηρεσίας θα αποτελούσε μεγάλη εξοικονόμηση χρόνου.

Επίσης, η αντιγραφή των ονομάτων χρήστη και των κωδικών πρόσβασης είναι ένα στοίχημα ασφάλειας. Εάν ένας ιστότοπος αποτύχει, τα κρίσιμα δεδομένα του χρήστη εκτίθενται και είναι ευάλωτα σε όλες τις πλατφόρμες. Αλλά η σύνδεση σε έναν άλλο ιστότοπο με επικύρωση που παρέχεται από τον πρώτο είναι πολύ διαφορετική. Αυτό είναι ένα ακόμα πλεονέκτημα χρήσης του OAuth.

Η φιλοσοφία της πιλοτικής εφαρμογής, που αναφέρεται στο πρώτο κεφάλαιο, για τη σύνδεση χρηστών από μια υπηρεσία που θα παρέχει ισχυρή αυθεντικοποίηση μέσω βιομετρικών χαρακτηριστικών, ταιριάζει με το OAuth πρότυπο. Αυτό λοιπόν ακολουθείται για την υλοποίηση της επαήθευσης ταυτότητας από εφαρμογές τρίτων. Πιο συγκεκριμένα, χρησιμοποιείται το **OAuth 2.0** το οποίο αποτελεί την νεότερη έκδοση του πρώτου.

Πως, όμως, μπορεί ένα πρότυπο εξουσιοδότησης να εξυπηρετήσει μια διαδικασία αυθεντικοποίησης χρήστη που θα παρέχει η υπηρεσία; Η απάντηση είναι πως αυτό γίνεται με δύο τρόπους. Είτε με την παροχή, μόνο πληροφοριών του χρήστη από κάποια πηγή, είτε τη χρήση **OpenID Connect (OIDC)**. Το OIDC αποτελεί ένα πρότυπο αυθεντικοποίησης χτισμένο πάνω στην υλοποίηση του OAuth 2.0. Η ροή χρήσης του OAuth 2.0 με openID χρησιμοποιείται από διάφορες υπηρεσίες γι' αυτό το σκοπό, αλλά δεν είναι απαραίτητη για την επίτευξη της αυθεντικοποίησης. Και στις δύο περιπτώσεις, χρήσης ή μη του OpenID, η τρίτη εφαρμογή λαμβάνει κάποιες πληροφορίες για τον χρήστη. Από εκείνο το σημείο και μετά, είναι ευθύνη της εφαρμογής να αυθεντικοποιήσει τον χρήστη μέσω δικιάς της υλοποίησης, έτσι ώστε να διατηρεί κάποιο state και να θεωρείται ταυτοποιημένος κατά τη διάρκεια περιήγησής του σε αυτή. Απλά, στην περίπτωση του OpenID ο τρόπος με τον οποίο λαμβάνεται η πληροφορία του χρήστη, αποτελεί ένα τυποποιημένο τρόπο. Δηλαδή, ένα πρότυπο, χάρη στο οποίο, η τρίτη εφαρμογή θα ξέρει ακριβώς τι είδους πληροφορίες για τον χρήστη να περιμένει. Πιο συγκεκριμένα, αυτή η πληροφορία βρίσκεται μέσα σε ένα **Json Web Token (JWT)**, που μπορεί να περιέχει πληροφορίες σχετικά με το όνομα, επώνυμο, ηλικία κλπ. Αυτό εξαρτάται με τι θα ζητήσει, ή αλλιώς, θα κάνει claim η εφαρμογή.

Στην περίπτωση της πιλοτικής εφαρμογής έχει ακολουθηθεί η υλοποίηση OAuth 2.0, δίχως το OpenID. Παρέχεται ένας μη τυποποιημένος τρόπος παροχής πληροφοριών του χρήστη μετά την εξουσιοδότηση μεσω OAuth 2.0, που θα περιγραφεί στο 7ο κεφάλαιο. Εδώ να αναφερθεί πως αυτή η υλοποίηση δεν είναι καθοριστική, καθώς η ενσωμάτωση του προτύπου OpenID δεν θα επηρεάσει την ύπαρξη άλλων μεθόδων που έχουν ήδη υλοποιηθεί. Από εκείνο το σημείο και μετά, θα είναι στο χέρι του προγραμματιστή να επιλέξει με ποιον τρόπο θα εξουσιοδοτήσει τον χρήστη μέσω της υπηρεσίας που του παρέχεται.

3.3 OAuth 2.0 Authorization Framework

Για να κατανοηθεί πλήρως το OAuth 2.0, παρατίθεται ο παρακάτω ορισμός:

Το πρότυπο εξουσιοδότησης OAuth 2.0 επιτρέπει την απόκτηση περιορισμένης πρόσβασης σε μια HTTP υπηρεσία σε τρίτες εφαρμογές, είτε εκ μέρους ενός κατόχου κάποιου πόρου, δίνοντας τη δυνατότητα μιας αλληλεπίδρασης έγκρισης μεταξύ του κατόχου του πόρου και της υπηρεσίας, είτε επιτρέποντας την τρίτη εφαρμογή να αποκτήσει πρόσβαση για λογαριασμό του. Η υλοποίηση αυτή αντικαθιστά απόλυτα το πρωτόκολλο OAuth 1.0. [27]

3.3.1 **Ρόλοι**

Για την υλοποίηση του OAuth πρέπει να κατανοηθούν οι ρόλοι που ορίζει αυτό:

Resource Owner: Αποτελεί μια οντότητα (χρήστη), δυνατή να παρέχει πρόσβαση σε μια προστατευμένη πηγή. Όταν ο resource owner είναι ένας άνθρωπος, αποκαλείται end-user.

Resource Server: Αποτελεί το server που φιλοξενεί την προστατευμένη πηγή. Είναι ικανός να δέχεται και να απαντάει σε αιτήσεις απόκτησης αυτών των προστατευμένων πηγών με τη χρήση access tokens.

Client: Αποτελεί μια εφαρμογή που κάνει αιτήσεις προστατευόμενων πηγών εκ μέρους κάποιου resource owner με την εξουσιοδότησή του.

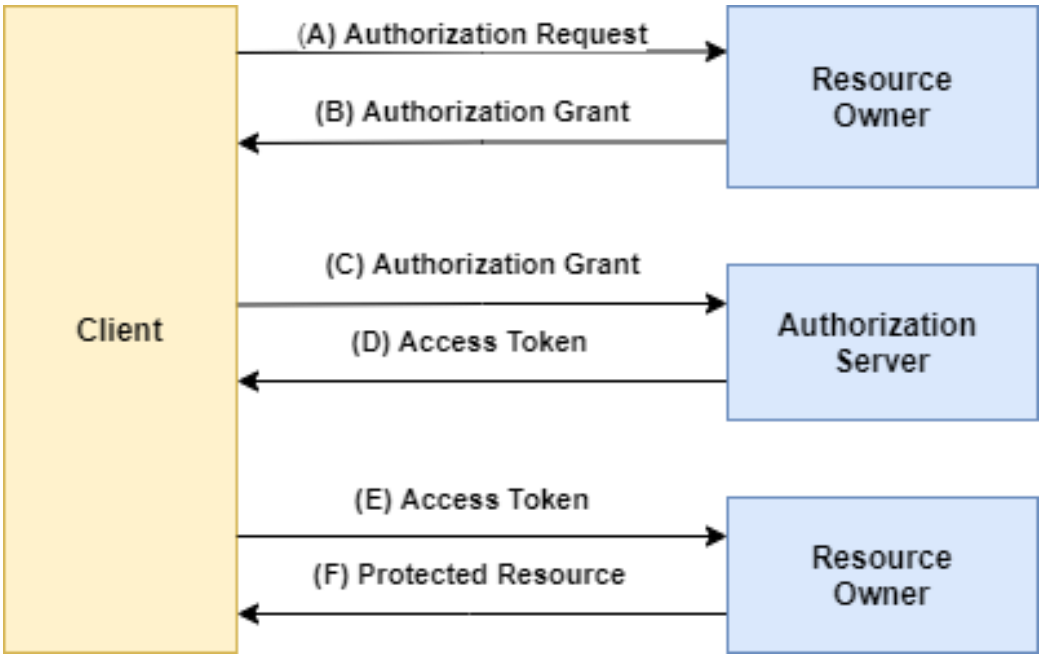
Authorization Server: Αποτελεί τον server που παρέχει access tokens στον client μετά την επιτυχή αυθεντικοποίηση του resource owner και απόκτηση της εξουσιοδότησης μέσω αυτού.

3.3.2 **Διαμόρφωση**

Η αλληλεπίδραση μεταξύ του **Resource Server** και του **Authorization Server** εξαρτάται από την εκάστοτε υλοποίηση μιας εφαρμογής και δεν ορίζεται από αυτή τη προδιαγραφή. Η υλοποίηση των διεργασιών του **Resource Server** και του **Authorization Server** μπορεί να είναι ένας server μόνος του ή δύο ξεχωριστοί που συνδέονται μεταξύ τους. Ένας μόνο **Authorization Server** μπορεί να εκδώσει διακριτικά πρόσβασης αποδεκτά από πολλαπλούς **Resource Servers**.

3.3.3 **Ροή πρωτοκόλλου**

Η βασική ροή το πρωτοκόλλου φαίνεται στο παρακάτω διάγραμμα:



Σχήμα 3.1: Βασική ροή πρωτοκόλλου OAuth 2.0

Η ροή που αναπαρίσταται στο παραπάνω σχήμα αναφέρεται στα εξής βήματα:

A. Ο **client** ζήτα ένα **access token,** πραγματοποιώντας αυθεντικοποίηση στον **authorization server** και έτσι παρουσιάζει ένα **authorization grant.**

B. Ο **authorization server** αυθεντικοποιεί τον **client** και επικυρώνει το **authorization grant.** Αν αυτό είναι έγκυρο, τότε δημιουργεί ένα **access token** και ένα **refresh token**, στέλνοντας το στον **client.**

C. Ο **client** κάνει ένα αίτημα απόκτησης μια προστατευμένης πηγής από τον **resource server** με το **access token.**

D. Αν το **access token** είναι έγκυρο, τότε στέλνεται στον **client** η προστατευμένη πληροφορία.

E. Τα τρία προηγούμενα βήματα συνεχίζονται μέχρι να λήξει το το **access token.** Αν γίνει αυτό, αμέσως μετά, έρχεται το βήμα **G.**

F. Όταν το **access token** σταματήσει να είναι έγκυρο, επιστρέφεται ένα **invalid token error.**

G. Ο **client** ζητά ένα καινούργιο **access token**, παρουσιάζοντας το **refresh token** που ήδη έχει.

H. Αν το **refresh token** είναι σωστό, τότε δημιουργείται ένα καινούργιο **access token** και στέλνεται στον **client.**

3.3.4 **Authorization Grant**

Το πρωτόκολλο ορίζει πολλούς τρόπους με τους οποίους ένας client μπορεί να παραλάβει το access token και να αποκτήσει πρόσβαση σε μια προστατευμένη πηγή. Κάθε μέθοδος αποτελεί και μια διαφορετική υλοποίηση και επέκταση της γενικής ροής που παρουσιάστηκε παραπάνω. Οι μέθοδοι αυτές είναι **1) Authorization Code flow, 2) Implicit Flow, 3) Resource Owner Password Credentials** και **4) Client Credentials.** Οι πιο διαδεδομένες μέθοδοι είναι οι δύο πρώτες. Χρησιμοποιούνται από μεγάλους οργανισμούς όπως είναι το Facebook και η Google.

Η ροή Implicit flow είναι μια πιο απλοποιημένη μορφή της Authorization Code flow. Προορίζεται για περιπτώσεις όπου ο client αποτελεί μια web εφαρμογή που χρησιμοποιεί scripting γλώσσα όπως είναι αυτή της Javascript.

Η Πιλοτική εφαρμογή που έχει υλοποιηθεί, ακολουθεί τη ροή Authorization Code, καθώς η client test εφαρμογή που θα βοηθήσει στην υλοποίηση είναι server based.

3.3.5 **Authorization Code Grant Flow**

Ο τύπος grant authorization code χρησιμοποιείται για να παραλαμβάνει access tokens και refresh tokens και προορίζεται για εμπιστευτικούς clients. Εφόσον αυτός ο τύπος ροής σχετίζεται με ανακατευθύνσεις (redirections), ο client πρέπει να είναι ικανός να αλληλοεπιδρά με τον user-agent του resource owner. Δηλαδή να μπορεί να έχει πρόσβαση με τον web browser και να μπορεί να δέχεται εισερχόμενες αιτήσεις (μέσω ανακατεύθυνσης) απο τον authorization server.

Η ροή authorization code που παρουσιάζεται και στο σχήμα 3.2, ακολουθεί τα επόμενα βήματα:

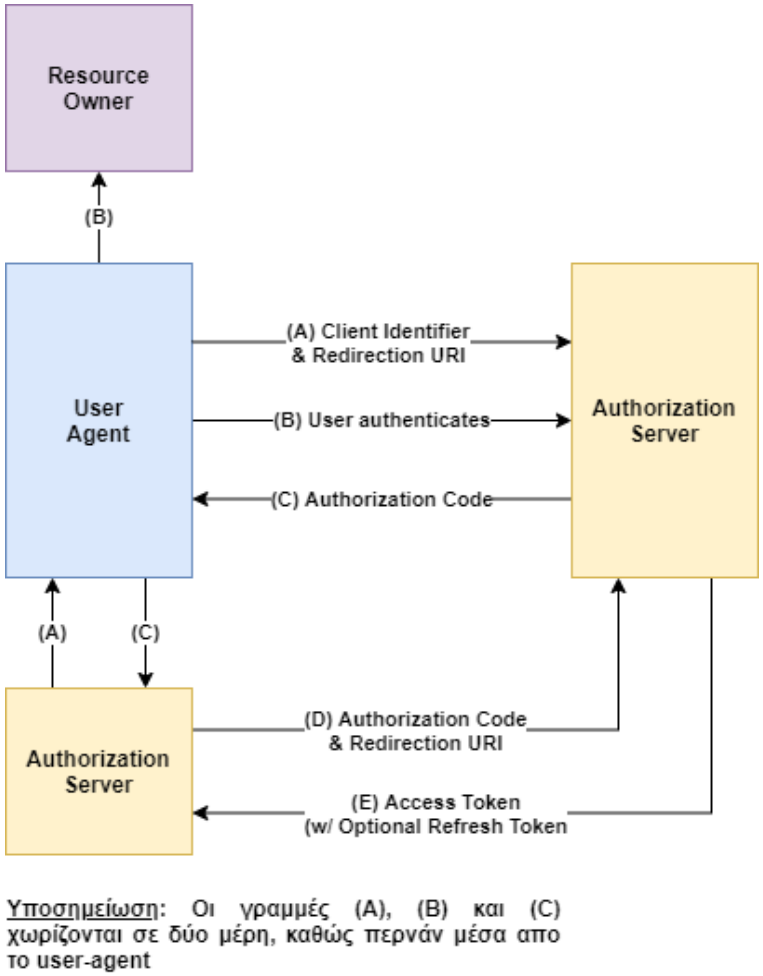
A. Η ροή ξεκινάει με τον **client**, ο οποίος ανακατευθύνει τον agent του **resource owner** στον **authorization server**. Ο **client** κατέχει ένα αναγνωριστικό ως προς τον **server**. Μαζί με αυτό κατέχει scope, δηλαδή το περιεχόμενο που θέλει, ένα state της κατάστασής του και ένα uri ανακατεύθυνσης στο οποίο ο user-agent θα πάει όταν του δοθεί ή αρνηθεί η πρόσβαση

B. Ο **authorization server** αυθεντικοποιεί τον **resource owner** μέσω του user-agent και επικυρώνει αν αυτός εξουσιοδοτεί ή όχι τον **client** για πρόσβαση.

C. Θεωρώντας πως ο **resource owner** εξουσιοδοτεί τη πρόσβαση, ο **authorization server** ανακατευθύνει τον **client** στο uri που είχε δώσει πριν. Αυτό το uri ανακατεύθυνσης περιέχει, επίσης, έναν κωδικό εξουσιοδότησης ή αλλιώς authorization code. Αυτός είναι και ο κωδικός που ορίζει τη φύση αυτής της ροής. Τέλος περιέχει το state που έστειλε πριν ο **client**.

D. Ο **client** πρέπει να επικυρώσει ότι το state είναι ίδιο με αυτό που έστειλε για να συνεχίσει. Εφόσον είναι σωστό, κάνει αίτηση στον **authorization server** μαζί με το authorization code που έλαβε στο προηγούμενο βήμα. Το αίτημα αυτό προορίζεται για το token endpoint του server. Ουσιαστικά, ζητά να του στείλει το τελικό access token ώστε να έχει πρόσβαση στα resources του χρήστη. Στο αίτημα αυτό, συμπεριλαμβάνει και το uri ανακατεύθυνσης του **client** για παραπάνω ασφάλεια.

E. Ο **authorization server** αυθεντικοποιεί τον **client**, επικυρώνει το authorization code και βεβαιώνει ότι το uri ανακατεύθυνσης είναι ίδιο με αυτό που ανακατεύθυνε τον user πριν στο βήμα C. Αν είναι όλα έγκυρα, απαντά πίσω στον client με ένα access token, και κατά επιλογή ένα refresh token.



Σχήμα 3.2: Ροή Authorization code grant πρωτοκόλλου OAuth 2.0

Η παραπάνω ροή υλοποιείται μεταξύ της πιλοτικής εφαρμογής που περιγράφεται στο κεφάλαιο 7, που θα έχει τον ρόλο του authorization και resource server. Μαζί με αυτή, έχει υλοποιηθεί μία άλλη test εφαρμογή που προσομοιώνει το ρόλο του client.

3.4 Επίλογος

Ο ρόλος της διαλειτουργικότητας, στη διασύνδεση σύγχρονων υπηρεσιών και οργανισμών, με ένα κοινό κώδικα επικοινωνίας, έχει συμβάλλει σημαντικά στη δημιουργία προτύπων εξουσιοδότησης, όπως είναι το OAuth 2.0. Η ροή του προτύπου αυτού, όπως αναφέρθηκε, θα χρησιμοποιηθεί για τη πραγματοποίηση διαλειτουργικής αυθεντικοποίησης, με βιομετρικά χαρακτηριστικά, μέσω της πιλοτικής εφαρμογής, στα πλαίσια αυτής της πτυχιακής εργασίας.

Η υλοποίηση της ροής αυτής για τη πιλοτική εφαρμογή όμως, δεν είναι εφικτή χωρίς την αναγνώριση βιομετρικών χαρακτηριστικών. Το κεφάλαιο 4 λοιπόν, αναφέρεται στην βιομετρική αυθεντικοποίηση και στις έννοιες σχετικά με αυτή.

Κεφάλαιο 4ο: Βιομετρική αυθεντικοποίηση

4.1 Εισαγωγή

Η βιομετρική αυθεντικοποίηση αποτελεί επέκταση της τεχνολογίας της βιομετρίας. Όπως αναφέρθηκε στο 1ο και στο 2ο κεφάλαιο, ένας τρόπος αυθεντικοποίησης αποτελεί η χρήση ενός βιομετρικού χαρακτηριστικού. Αυτό μπορεί να αποτελεί και ένα από τα πιο ισχυρά και ασφαλή χαρακτηριστικά που επιβεβαιώνουν την ταυτότητα ενός ανθρώπου, γιατί είναι σχεδόν αδύνατο να αντιγραφούν και να μιμηθούν από άλλους. Σύμφωνα με τα δεδομένα αυτά, προκύπτει ο ορισμός της βιομετρικής αυθεντικοποίησης.

4.2 Ορισμός

Η βιομετρική αυθεντικοποίηση είναι μια διαδικασία ασφάλειας που βασίζεται στα μοναδικά βιολογικά χαρακτηριστικά των ατόμων, για να επαληθεύσει ότι είναι αυτά που ισχυρίζονται ότι είναι. Τα βιομετρικά συστήματα ελέγχου ταυτότητας συγκρίνουν φυσικά ή συμπεριφορικά χαρακτηριστικά με αποθηκευμένα δεδομένα σε μια βάση δεδομένων. Εάν τα δύο δείγματα των βιομετρικών δεδομένων ταιριάζουν, επιβεβαιώνεται ο έλεγχος ταυτότητας. Συνήθως, ο βιομετρικός έλεγχος ταυτότητας χρησιμοποιείται για τη διαχείριση πρόσβασης σε φυσικούς και ψηφιακούς πόρους, όπως κτίρια, δωμάτια και υπολογιστικές συσκευές. [10]

4.3 Αναγνώριση με βιομετρικά χαρακτηριστικά

Όπως αναλύθηκε παραπάνω, οι παραδοσιακοί τρόποι αναγνώρισης και αυθεντικοποίησης βασίζονται είτε σε κάτι που ο χρήστης κατέχει, όπως είναι ένα δελτίο ταυτότητας ή μια έξυπνη κάρτα, είτε σε κάτι που ξέρει, για παράδειγμα ένα συνθηματικό ή ένας κωδικός πρόσβασης. Τα προβλήματα που δημιουργούνται σε αυτές τις περιπτώσεις είναι η τυχόν πλαστογράφηση, υποκλοπή και απώλεια των παραπάνω μέσων αναγνώρισης.

Τα βιομετρικά συστήματα, χρησιμοποιώντας τα μοναδικά βιομετρικά χαρακτηριστικά κάθε ανθρώπου, καταφέρνουν να εξαλείψουν τα μειονεκτήματα των παραδοσιακών μεθόδων, εφόσον ένα βιομετρικό χαρακτηριστικό είναι δύσκολο να ξεχαστεί, να χαθεί ή να μιμηθεί.

4.4 Βιομετρική επαλήθευση και ταυτοποίηση

Η βιομετρική αναγνώριση προτύπων μπορεί να χρησιμοποιηθεί τόσο για αυθεντικοποίηση όσο και για ταυτοποίηση. Είναι δύο διαφορετικές έννοιες που μπορεί κάποιος να μπερδέψει. Όπως αναλύθηκε και στο προηγούμενο κεφάλαιο, η αυθεντικοποίηση μιας υπηρεσίας έχει ως προϋπόθεση το άτομο να έχει αρχικά ισχυριστεί ότι έχει μια συγκεκριμένη ταυτότητα την οποία καλείται να επαληθεύσει, είτε με κάποιο συνθηματικό είτε με βιομετρικά χαρακτηριστικά. Συγκεκριμένα, σε περίπτωση βιομετρικής αυθεντικοποίησης, μια υπηρεσία ή ένα σύστημα καλείται να συγκρίνει δύο δείγματα βιομετρικών χαρακτηριστικών και να αυθεντικοποιήσει τον χρήστη, εφόσον τα δύο αυτά δείγματα είναι όμοια σε μεγάλο βαθμό. Ουσιαστικά, πρόκειται για μια ταύτιση ένας προς ένα (one-to-one), η οποία δεν απαιτεί τεράστια υπολογιστική ισχύ.

Σε αντίθεση, η ταυτοποίηση, και πιο συγκεκριμένα, η βιομετρική ταυτοποίηση, καλείται να απαντήσει στο ερώτημα “ποιος είμαι ;“. Δηλαδή, ένα σύστημα βιομετρικής ταυτοποίησης προσπαθεί να συγκρίνει και να ταιριάξει το βιομετρικό δείγμα που του παρέχεται ανάμεσα με όλα όσα είναι αποθηκευμένα στη

βάση δεδομένων του. Αυτό απαιτεί τεράστια υπολογιστική ισχύ και η απόδοσή τους σε χρόνο και ακρίβεια εξαρτάται από το πλήθος χρηστών των οποίων τα δείγματα αποθηκεύει.

Εικόνα 4.1 : One to one verification / authentication

Εικόνα 4.2: One to many identification

Συνοψίζοντας, ο βασικός διαχωρισμός των δύο παραπάνω εννοιών είναι η ερώτηση:

“Είμαι αυτός που ισχυρίζομαι ότι είμαι;” και “Ποιος είμαι;” για τα συστήματα αυθεντικοποίησης και ταυτοποίησης αντίστοιχα.

4.5 Είδη Βιομετρικών Χαρακτηριστικών

Τα βιομετρικά χαρακτηριστικά ενός ατόμου μπορούν να χωριστούν σε τρεις βασικές κατηγορίες. Σε φυσιολογικά, συμπεριφορικά και βιοχημικά.

Τα φυσιολογικά βιομετρικά αφορούν την μέτρηση χαρακτηριστικών του ανθρωπίνου σώματος [11]:

- **Πρόσωπο:** Η αναγνώριση προσώπου βασίζεται στην αυτοματοποιημένη, πλέον, διαδικασία της ανάλυσης της γεωμετρίας, βάση ορισμένων σημείων, και τη δημιουργία ενός face print. Η εξαγωγή των χαρακτηριστικών αυτών βασίζεται στον φωτισμό και την οπτική γωνία της φωτογραφίας, που συχνά δυσκολεύει τη λειτουργία αυτή.
- **Θερμογραφική Προσώπου**: Εντοπισμός θερμότητας φλεβών κάτω από το δέρμα προσώπου.
- **Εγκεφαλικά Κύματα**: Θεωρείται πως τα εγκεφαλικά κύματα και σήματα κάθε εγκεφάλου παράγουν ένα μοναδικό σύνολο χαρακτηριστικών που διαφέρει από κάθε άλλο σύνολο. Βάσει αυτής της θεωρίας, εξελίσσεται μια καινούργια τεχνολογία βιομετρίας.
- **Δακτυλικό Αποτύπωμα**: Το δακτυλικό αποτύπωμα αποτελεί μοναδικό χαρακτηριστικό κάθε ατόμου, καθώς κάθε άνθρωπος έχει διαφορετικό μοτίβο.
- **Υπέρυθρη εικόνα δακτύλου**: Αναλύει το, μοναδικό για τον καθένα, θερμικό αποτύπωμα.

- **Ίριδα ματιού**: Η υφή της ίριδας περιέχει διακριτά μοτίβα που χρησιμοποιούνται για ταυτοποίηση. Η υφή αυτή είναι διαφορετική για κάθε άτομο, ακόμα και ανάμεσα σε ομοζυγωτικά δίδυμα, και δε μπορεί να αλλάξει ούτε χειρουργικά.

- **Αμφιβληστροειδής Αδένας**: Αποτελείται από τα μοτίβα των αιμοφόρων αγγείων του ματιού και έχει τα ίδια χαρακτηριστικά με την ίριδα του ματιού.

- **Εξωτερικό αυτί**: Τα χαρακτηριστικά του αποτελούν μοναδικά για κάθε άτομο. Ωστόσο, δεν αποτελεί μια ευρέως διαδεδομένη τακτική αυθεντικοποίησης τη σύγχρονη εποχή.

- **Σώμα**: Άσχετα που κάθε ανθρώπινο σώμα μπορεί να διαφέρει από κάθε άλλο, η ανθρωπομετρία δεν αποτελεί 100% αξιόπιστο τρόπο ταυτοποίησης.

- **Φλέβες**: Όπως στα χέρια, έτσι και σε όλο το σώμα, παράγουν μοναδικά μοτίβα.

- **Γεωμετρία και κινήσεις χειρός**: Διακρίνει και συγκρίνει το μήκος, το πλάτος και το άνοιγμα των δακτύλων στο χώρο. Πρόσφατα, οι κινήσεις των χεριών στο χώρο αποτελούν και αυτές συντελεστή.

- **DNA**: Το DNA παραμένει αναλλοίωτο καθ' όλη τη διάρκεια της ζωής του ατόμου, ακόμα και μετά το θάνατο. Αποτελεί μια από τις ακριβότερες μεθόδους εξαγωγής του. Ένα μειονέκτημα είναι, πως δε μπορεί να ξεχωρίσει ανάμεσα σε μονοζυγωτικά δίδυμα.

Τα συμπεριφορικά βιομετρικά αφορούν την έμμεση μέτρηση ανθρώπινων χαρακτηριστικών. Βασίζονται σε μετρήσεις αποτελεσμάτων συμπεριφοράς του ατόμου. Τέτοια χαρακτηριστικά αποτελούν:

- **Φωνή**: Εξαρτάται από τη φυσιολογία των φωνητικών χορδών, τη ρινική και στοματική κοιλότητα κ.ά. τα οποία παράγουν μοναδικό ήχο κατά την ομιλία του ατόμου και μένουν αναλλοίωτα, αλλά και από συμπεριφορικούς παράγοντες όπως η ηλικία, η υγεία, η ψυχική κατάσταση κλπ. Θεωρείται ηθικά αποδεκτή μέθοδος αλλά είναι ευάλωτη σε απάτες.

- **Βάδισμα**: Ο τρόπος που περπατά ένα άτομο. Αποτελεί ένα πολύπλοκο βιολογικό χαρακτηριστικό. Δεν είναι μοναδικό χαρακτηριστικό, και αλλάζει με την πάροδο του χρόνου, αλλά αρκεί για ταυτοποίηση.

- **Πληκτρολόγηση (keystroke)**: Συνδέεται με το τρόπο τον οποίο πληκτρολογεί ένα άτομο. Θεωρείται ανεπαρκής μέθοδος αυθεντικοποίησης.

- **Υπογραφή**: Αποτελεί τη, θεωρητικά, μοναδική υπογραφή που αποφασίζει να έχει κάθε άνθρωπος. Μπορεί εύκολα να υποκλαπεί και είναι επιρρεπής σε απάτες.

Τέλος, τα βιοχημικά χαρακτηριστικά για την αναγνώριση ενός ατόμου αφορούν τη μελέτη της βιοχημείας του ανθρώπινου σώματος, όπως:

- **Οσμή σώματος**: Αποσκοπεί στην αναγνώριση του ατόμου μέσω της οσμής που εκπέμπουν οι ίδιοι, από διάφορα μέρη του σώματος. Ωστόσο, αποτελεί ακόμα εξελισσόμενη τεχνολογία και δεν είναι αξιόπιστη.

- **Αλατότητα σώματος**: Επίσης αναπτυσσόμενη. Μέσω μικρών ηλεκτρικών παλμών μέσα στο σώμα, προσδιορίζονται τα επίπεδα άλατος και δημιουργούνται αντίστοιχα προφίλ.

Στον παρακάτω πίνακα αναφέρονται κάποια από τα παραπάνω είδη βιομετρικών χαρακτηριστικών και συγκρίνονται μεταξύ τους, με βάση τη καθολικότητα (universality), τη διακριτικότητα (distinctiveness), τη μονιμότητα (permanence), τη συλλεκτικότητα (collectability), την απόδοση

(performance), την αποδοχή (acceptability) και την καταστρατήγηση (circumvention). Το γράμμα '**H**' υποδηλώνει υψηλό βαθμό, ενώ το '**M**' και το '**L**' μεσαίο και μικρό αντίστοιχα.

Βιομετρικά Χαρακτηριστικά	**Καθολικότητα**	**Διακριτικότητα**	**Μονιμότητα**	**Συλλεκτικότητα**	**Απόδοση**	**Αποδοχή**	**Καταστρατήγηση**
Θερμογραφική Προσώπου	**H**	**H**	**L**	**H**	**M**	**H**	**L**
Φλέβα Χεριού	**M**	**M**	**M**	**M**	**M**	**M**	**L**
Βάδισμα	**M**	**L**	**L**	**H**	**L**	**H**	**M**
Πληκτρολόγηση	**L**	**L**	**L**	**M**	**L**	**M**	**M**
Οσμή	**H**	**H**	**H**	**L**	**L**	**M**	**L**
Εξωτερικό Αυτιού	**M**	**M**	**H**	**M**	**M**	**H**	**M**
Γεωμετρία Χεριού	**M**	**M**	**M**	**H**	**M**	**M**	**M**

Δακτυλικό Αποτύπωμα	**M**	**H**	**H**	**M**	**H**	**M**	**M**
Πρόσωπο	**H**	**L**	**M**	**H**	**L**	**H**	**H**
Αμφιβληστροειδής χιτώνας	**H**	**H**	**M**	**L**	**H**	**L**	**L**
Ίρις	**H**	**H**	**H**	**M**	**H**	**L**	**L**
Αποτύπωμα παλάμης	**M**	**H**	**H**	**M**	**H**	**M**	**M**
Φωνή	**M**	**L**	**L**	**M**	**L**	**H**	**H**
Υπογραφή	**L**	**L**	**L**	**H**	**L**	**H**	**H**
DNA	**H**	**H**	**H**	**L**	**H**	**L**	**L**

(**H**: High, **M**: Medium, **L**: Low)

Πίνακας 4.1: Σύγκριση ιδιοτήτων διάφορων βιομετρικών χαρακτηριστικών

4.6 Αναδυόμενες περιπτώσεις χρήσης βιομετρικής αυθεντικοποίησης

Οι επιχειρήσεις αρχίζουν να χρησιμοποιούν όλο και περισσότερες μεθόδους βιομετρικής αυθεντικοποίησης, έτσι ώστε να μπορούν να ανταπεξέλθουν στις διαρκώς αναδυόμενες περιπτώσεις χρήσης που προκύπτουν, όταν ανιχνεύονται ύποπτες συνδέσεις χρηστών. Αυτές μπορεί να είναι οι εξής: [15]

- **Secondary authentication** (Δευτερεύουσα αυθεντικοποίηση): Αντί να χρησιμοποιούν μόνο το username και τον κωδικό πρόσβασης, οι οργανισμοί μπορούν να χρησιμοποιούν κάποιο βιομετρικό χαρακτηριστικό, ως δεύτερο παράγοντα ελέγχου ταυτότητας.

- **Εξουσιοδότηση ενεργειών υψηλού κινδύνου**: Επαλήθευση των χρηστών πριν από συναλλαγές υψηλής αξίας, όπως τραπεζικές μεταφορές, ηλεκτρονικές αγορές ή πληρωμή λογαριασμών. Απαιτώντας κάποιο έγκυρο βιομετρικό χαρακτηριστικό, τα χρηματοπιστωτικά

ιδρύματα και οι διαδικτυακοί πελάτες τους μπορούν να είναι σίγουροι ότι το αίτημα είναι νόμιμο και έχει εγκριθεί.

- **Ξεκλείδωμα πορτών**: Για παράδειγμα, μπορεί ένας χρήστης να έχει κάνει κράτηση ενοικίασης αυτοκινήτου και ο πελάτης να ζητήσει αυθεντικοποίηση, για να ξεκλειδώσει το αυτοκίνητο.

- **Self check-in**: Κατά τη διάρκεια του check-in, τα ξενοδοχεία θα μπορούσαν να υλοποιήσουν μια λειτουργία αυτόματης αναγνώρισης και ταυτοποίησης του πελάτη, που έκανε την κράτηση, μέσω των βιομετρικών χαρακτηριστικών του.

- **Ανανέωση στοιχείων χρήστη**: Για κάθε περίπτωση, στην οποία απαιτείται έλεγχος ταυτότητας. Για παράδειγμα, στις συνδέσεις ή σε περιπτώσεις επαναφοράς κωδικών πρόσβασης.

- **Συνεχής ασφάλεια**: Να ζητείται η ταυτοποίηση του χρήστη, μέσω βιομετρικών χαρακτηριστικών, ώστε να αποτρέπεται η περίπτωση υποκλοπής του λογαριασμού.

- **Μάθηση εξ αποστάσεως**: Τα πανεπιστήμια, οι πάροχοι ηλεκτρονικής μάθησης και οι υπηρεσίες διδασκαλίας, χρειάζονται συχνά μια αξιόπιστη λύση, για να είναι βέβαιοι, ότι ένας μαθητής, που θα χρειάζεται να δώσει εξετάσεις, είναι πραγματικά το άτομο που πρέπει να πάρει μέρος. Θα ζητείται, δηλαδή, από τον χρήστη, να πραγματοποιήσει έλεγχο ταυτότητας, πριν και κατά τη διάρκεια της εξέτασης.

4.7 Πλεονεκτήματα βιομετρικής επαλήθευσης προσώπου

Η χρήση της βιομετρικής αυθεντικοποίησης επιτρέπει στις διαδικτυακές επιχειρήσεις να πιστοποιούν αξιόπιστα τους χρήστες, για τακτικές συνδέσεις, συναλλαγές υψηλού κινδύνου και για μια ποικιλία αναδυόμενων περιπτώσεων χρήσης. Και το πιο σημαντικό, βοηθά στην εξουδετέρωση του κινδύνου υποκλοπής άλλων, μη βιομετρικών χαρακτηριστικών αυθεντικοποίησης, όπως ένα username ή ένας κωδικός πρόσβασης, που θα μπορούσαν, αν όχι εύκολα, αλλά με διάφορες μεθόδους, να κλαπούν.

Για την επίτευξη της βιομετρικής αυθεντικοποίησης στα πλαίσια της πιλοτικής εφαρμογής, αποφασίστηκε πως θα χρησιμοποιηθεί αναγνώριση με τα βιομετρικά χαρακτηριστικά του προσώπου. Η αυθεντικοποίηση με βάση το πρόσωπο έχει ορισμένα πλεονεκτήματα έναντι των παραδοσιακών μεθόδων ελέγχου ταυτότητας: [8]

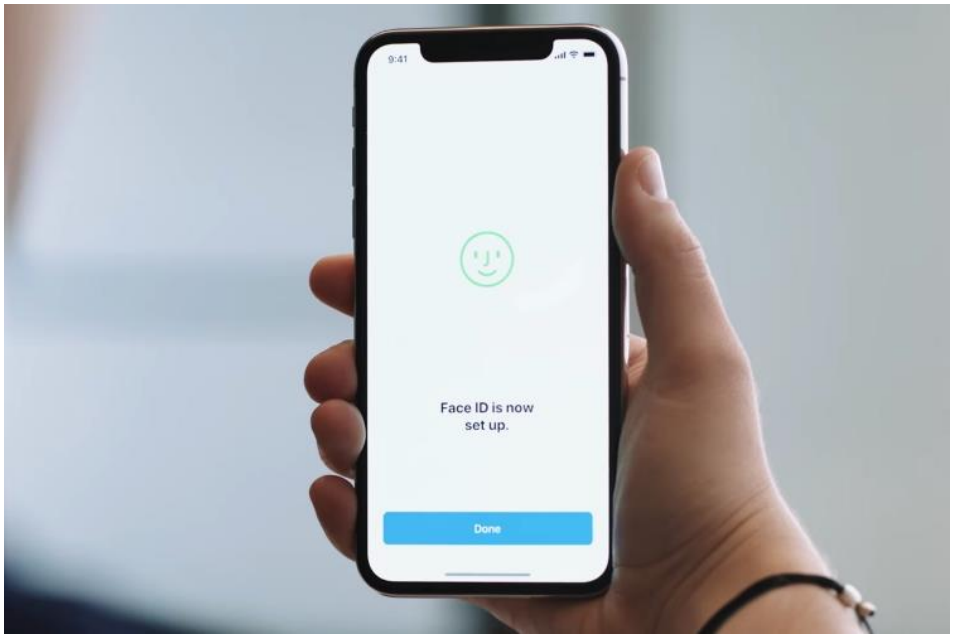
4.7.1 Διασφάλιση ταυτότητας

Μια υπηρεσία δεν μπορεί να είναι ποτέ σίγουρη για το άτομο που βρίσκεται πίσω από μια σύνδεση με κωδικό πρόσβασης ή αν είναι αυτό που ισχυρίζεται πως είναι. Πολλές από τις παραδοσιακές μεθόδους ελέγχου ταυτότητας (π.χ. KBA, 2FA), που βασίζονται σε SMS, δεν παρέχουν πραγματικά μεγάλη διασφάλιση ταυτότητας. Χάρη σε παραβιάσεις δεδομένων μεγάλης κλίμακας και κλοπές ταυτοτήτων, οι επιχειρήσεις δεν μπορούν να εμπιστευτούν απόλυτα ένα άτομο, ακόμα κι αν έχουν τη διεύθυνση αλληλογραφίας του ή διαθέτουν τον σωστό αριθμό Κοινωνικής Ασφάλισης.

Ο βιομετρικός έλεγχος ταυτότητας, με βάση το πρόσωπο, δεν είναι μόνο πολύ πιο βολικός από τις παραδοσιακές μεθόδους διαδικτυακής επαλήθευσης, αλλά είναι επίσης πολύ πιο ασφαλής. Τα βιομετρικά δεδομένα δεν μπορούν να παραβιαστούν ή να αναπαραχθούν. Επίσης, τα δεδομένα μπορούν να διατηρηθούν στη συσκευή και όχι σε κάποιον διακομιστή ή στο cloud. Τέλος, ακόμα και αν κλαπεί η συσκευή, αυτά μπορούν να παραμείνουν ασφαλή. Εξίσου σημαντικό είναι πως προσφέρει μια απλή λύση, πραγματοποιώντας την αυθεντικοποίηση σε ένα βήμα, σε αντίθεση με τις περιπτώσεις που χρησιμοποιούν PIN και κωδικούς πρόσβασης.

4.7.2 **Ευκολία στη χρήση**

Τα βιομετρικά στοιχεία, με βάση το πρόσωπο, έχουν γίνει πλέον μία από τις πιο δημοφιλής μεθόδους ελέγχου ταυτότητας, σε μεγάλο βαθμό, χάρη στη λειτουργία αναγνώρισης προσώπου, που εφάρμοσε η Apple στο iPhone. Το Face ID είναι πλέον το μοναδικό μέσο βιομετρικού ελέγχου ταυτότητας στα iPhone της Apple και φαίνεται ότι η εταιρεία θα διατηρήσει αυτό το σύστημα για το άμεσο μέλλον. Όλες οι νέες, φορητές συσκευές της Apple έχουν εγκαταλείψει τον έλεγχο ταυτότητας δακτυλικών αποτυπωμάτων Touch ID υπέρ του Face ID, ενός συστήματος υπέρυθρης αναγνώρισης προσώπου 3D.



Εικόνα 4.3: Παράδειγμα χρήσης Face ID

4.7.3 **Ανίχνευση απάτης**

Ο βιομετρικός έλεγχος ταυτότητας προσφέρει, επίσης, ισχυρότερη ανίχνευση απάτης, καθώς βασίζεται σε βιομετρικά δεδομένα, που είναι μοναδικά για κάθε άτομο. Η βιομετρική προσώπου πλεονεκτεί κατά βάση, εξαιτίας της απαίτησης από τον χρήστη να τραβήξει μια φωτογραφία του εαυτού του. Αυτή η μέθοδος αφήνει λίγα περιθώρια στους απατεώνες, οι οποίοι, χρησιμοποιώντας τις παραδοσιακές μεθόδους, είχαν ως στόχο την απομίμηση ενός ατόμου. Στη περίπτωση της βιομετρικής, η απομίμηση των βιομετρικών χαρακτηριστικών ενός προσώπου είναι εξαιρετικά δύσκολο να μιμηθεί.

Εταιρείες που υιοθετούν βιομετρικό έλεγχο ταυτότητας, παρέχουν ισχυρότερο έλεγχο και βοηθούν να κάνουν την ασφάλεια “αόρατη” στους πελάτες τους, με αποτέλεσμα υψηλότερα ποσοστά μετατροπών, υψηλότερα ποσοστά εντοπισμού απάτης και υψηλότερη ικανοποίηση των πελατών.

4.8 **Επίλογος**

Στο παρόν κεφάλαιο, παρατίθεται ο ορισμός της βιομετρικής αυθεντικοποίησης, για το πως αυτή χρησιμοποιείται στα σύγχρονα συστήματα και η διαφορά της βιομετρικής αναγνώρισης και ταυτοποίησης. Η βιομετρική αυθεντικοποίηση μπορεί να γίνει με τη χρήση διαφόρων βιομετρικών χαρακτηριστικών. Αυτά, λοιπόν, αναφέρονται εκτενέστερα και ύστερα γίνεται αναφορά στα πλεονεκτήματα για τα οποία η βιομετρική ταυτοποίηση, με χαρακτηριστικά προσώπου, επιλέχθηκε για την υλοποίηση της πιλοτικής εφαρμογής της πτυχιακής αυτής.

Κεφάλαιο 5ο: Βιομετρικά συστήματα

5.1 Εισαγωγή

Η πιλοτική εφαρμογή αποτελεί ένα βιομετρικό σύστημα. Επιβάλλεται, λοιπόν, ακολούθηση ενός προτύπου αρχιτεκτονικής βιομετρικού συστήματος. Μια λίστα αρχιτεκτονικών βιομετρικών συστημάτων αναφέρεται παρακάτω και γίνεται επεξήγηση του καθενός. Πρέπει πρώτα, όμως, να γίνει κατανοητή η έννοια του βιομετρικού συστήματος.

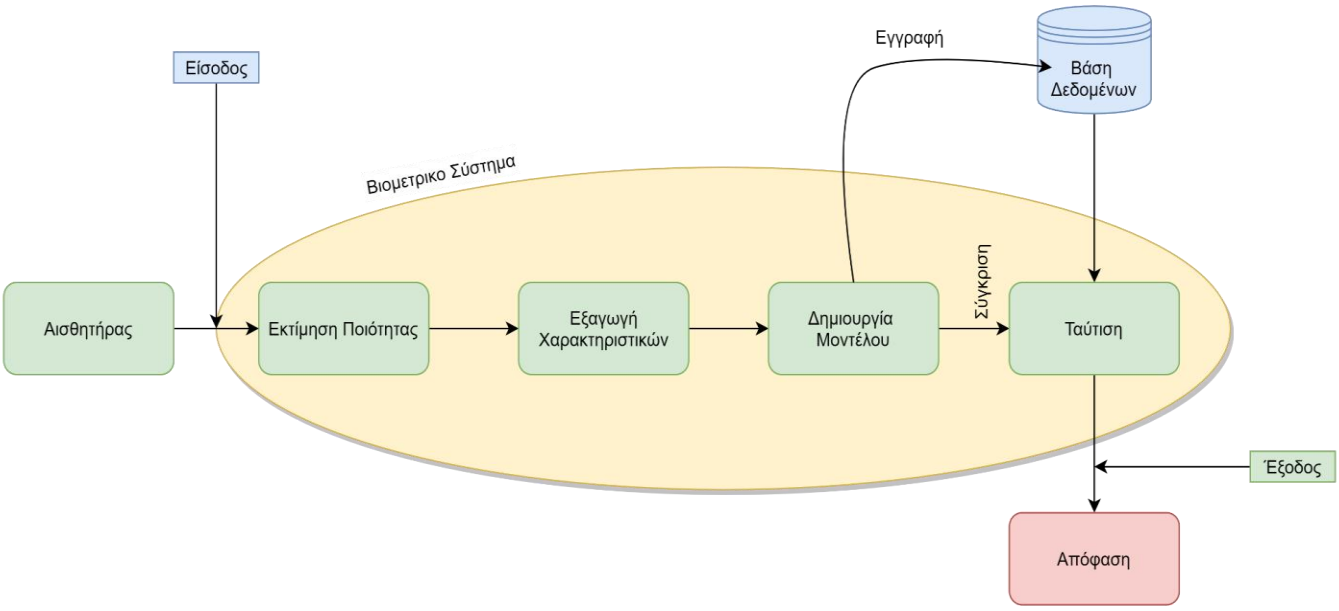
5.2 Χαρακτηριστικά βιομετρικών συστημάτων

Ένα βιομετρικό σύστημα είναι ένα τεχνολογικό σύστημα που χρησιμοποιεί πληροφορίες σχετικά με ένα άτομο ή άλλο βιολογικό οργανισμό, για την αναγνώριση αυτού του ατόμου. Τα βιομετρικά συστήματα βασίζονται σε συγκεκριμένα δεδομένα, σχετικά με μοναδικά βιολογικά χαρακτηριστικά, για να λειτουργήσουν αποτελεσματικά. Ένα βιομετρικό σύστημα περιλαμβάνει τη σύγκριση δεδομένων, μέσω αλγορίθμων, για ένα συγκεκριμένο αποτέλεσμα, που συνήθως σχετίζεται με μια θετική ταυτοποίηση ή επαλήθευση ενός χρήστη. [12]

Οι κυβερνήσεις, οι επιχειρήσεις και οι οργανισμοί μπορούν να χρησιμοποιήσουν βιομετρικά συστήματα για να λάβουν περισσότερες πληροφορίες για τα άτομα ή για τον πληθυσμό στο σύνολό του. Πολλά βιομετρικά συστήματα έχουν αναπτυχθεί για εφαρμογές ασφάλειας. Για παράδειγμα, συσκευές σάρωσης σε αεροδρόμια, συστήματα "βίο-κωδικού πρόσβασης" ή κάποιο πρωτόκολλο εσωτερικής συλλογής δεδομένων.

Κατά κόρον, ένα βιομετρικό σύστημα αποτελείται συνήθως από τα εξής χαρακτηριστικά : [17]

1. Έναν αισθητήρα για τη συλλογή και ψηφιοποίηση του βιομετρικού χαρακτηριστικού
2. Ένα τμήμα αξιολόγησης ποιότητας του δείγματος και εξαγωγής χαρακτηριστικών
3. Ένα τμήμα σύγκρισης, που συγκρίνει και, όπου γίνεται, ταυτίζει το δείγμα με άλλα υπάρχοντα στη βάση δεδομένων
4. Η Βάση Δεδομένων, όπου αποθηκεύονται τα δείγματα
5. Το τμήμα απόφασης, που με βάση τα αποτελέσματα της σύγκρισης επιστρέφει ένα αποτέλεσμα



Σχήμα 5.1: Τυπικό Βιομετρικό Σύστημα

Το κυρίως σύστημα περιλαμβάνει και εσωτερικά υποσυστήματα, και τις εξής βασικές λειτουργίες. Αρχικά, ο επιλεγμένος τύπος αισθητήρα εισάγει στο σύστημα τα βιομετρικά δεδομένα του ατόμου (Raw data). Ύστερα, ακολουθεί μια ακολουθία λειτουργιών. Αρχικά, γίνεται εκτίμηση ποιότητας των δεδομένων που έχουν εισαχθεί, καθώς πολλές φορές αυτά μπορεί να μην είναι κατάλληλα για σύγκριση ή εγγραφή, εξαιτίας πολλών εξωτερικών παραγόντων, όπως για παράδειγμα μία φωτογραφία προσώπου μπορεί να είναι πολύ σκοτεινή και να μην είναι εφικτό να γίνει η εξαγωγή χαρακτηριστικών που γίνεται στο επόμενο βήμα.

Το δεύτερο βήμα αποτελεί η εξαγωγή χαρακτηριστικών. Η διαδικασία αυτή διαφέρει σε κάθε σύστημα, καθώς υπάρχουν αρκετοί και διάφοροι τρόποι και μέθοδοι μηχανικής μάθησης, ώστε να γίνει εξαγωγή δεδομένων από μια πηγή. Ωστόσο, θα αναφερθούν παραδείγματα σε επόμενο κεφάλαιο.

Το επόμενο βήμα στη ροή του βιομετρικού συστήματος είναι, είτε η εγγραφή της καινούργιας ταυτότητας ενός χρήστη είτε η σύγκριση των μόλις εξαγόμενων χαρακτηριστικών με τα χαρακτηριστικά μιας ήδη αποθηκευμένης ταυτότητας χρήστη, που βρίσκεται στη βάση δεδομένων. Η εγγραφή περιλαμβάνει την αποθήκευση, ως template, στη βάση δεδομένων, τα προσωπικά στοιχεία του χρήστη (ονοματεπώνυμο, ID, PIN, διεύθυνση κ.λπ.) μαζί με τα βιομετρικά χαρακτηριστικά που εξήχθησαν στο προηγούμενο βήμα.

Το τελευταίο βήμα αποτελεί η απόφαση της επιτυχημένης ή μη αυθεντικοποίησης της ταυτότητας του χρήστη, μετά τη σύγκριση των αποθηκευμένων χαρακτηριστικών και αυτών που μόλις εξήχθησαν. Συνήθως, ανάλογα το τύπο του βιομετρικού συστήματος, αποφασίζεται ένα κατώφλι (threshold), το οποίο ορίζει το ελάχιστο επιτρεπτο ποσοστό ταύτισης των δύο συνόλων χαρακτηριστικών. Με βάση αυτήν την τιμή, θα αποφασίζεται, αν η αυθεντικοποίηση θα είναι επιτυχής ή όχι, μέσα από τον αλγόριθμο λήψης απόφασης. Αφού ληφθεί η απόφαση, το σύστημα, είτε παραχωρεί πρόσβαση στον χρήστη ή την αρνείται.

Η απόδοση ενός βιομετρικού συστήματος μετράται με δύο παράγοντες : [17]

- **False Acceptance Rate (FAR)**: Μετρά την πιθανότητα ένα βιομετρικό σύστημα να ταυτοποιήσει εσφαλμένα έναν μη εξουσιοδοτημένο χρήστη ως έγκυρο. Προφανώς, χαμηλό FAR σημαίνει υψηλό επίπεδο ασφάλειας

$$FAR = \frac{Πλήθος\ Εσφαλμένων\ Αποδοχών}{Πλήθος\ Προσπαθειών\ Ταυτοποίησης}$$

- **False Rejection Rate (FRR)** : Μετρά το ποσοστό ή τη συχνότητα απόρριψης και άρνησης πρόσβασης έγκυρων χρηστών από το σύστημα

$$FRR = \frac{Πλήθος\ Εσφαλμένων\ Αρνήσεων}{Πλήθος\ Προσπαθειών\ Ταυτοποίησης}$$

Η εσφαλμένη εγκατάσταση ή η κακή λειτουργία των αισθητήρων, προκαλεί υψηλό FAR. Η σωστή υλοποίηση του βιομετρικού αισθητήρα απαιτεί τον ορισμό μιας υλικής διεπαφής, που συνήθως βασίζεται σε βιομηχανικά πρότυπα. Αυτά ορίζουν, τόσο τη διεπαφή όσο και τα πρότυπα επικοινωνίας μεταξύ συσκευών. Τα raw data που εξάγει ο αισθητήρας, συνήθως αποθηκεύονται ως μεγάλα αρχεία, οπότε απαιτείται μια διεπαφή με ικανοποιητική ταχύτητα συνδεσιμότητας. Επίσης, αυτά τα δεδομένα αναλύονται και αξιολογούνται στο τμήμα αξιολόγησης, που αναφέρθηκε πρωτύτερα. Η διαδικασία αυτή βελτιώνει την ποιότητα των αρχείων και αξιολογεί την καταλληλόλητά τους. Συνολικά, έτσι ώστε η ροή αυτών των βημάτων να είναι γρήγορη, αποτελεσματική και ασφαλής, απαιτείται η σωστή λειτουργία και αξιοποίηση των αισθητήρων .

Η μέτρηση της απόδοσης ενός βιομετρικού συστήματος είναι αρκετά σημαντική, καθώς καθορίζει την επιτυχία του αλγορίθμου να ταυτοποιήσει τον χρήστη ορθά. Αυτό συμβάλλει σημαντικά στην συνολική ασφάλεια του συστήματος ώστε να παρέχεται μια ισχυρή ταυτοποίηση και επαλήθευση.

5.3 Μοντέλα βιομετρικών συστημάτων

Το Standards document ISO/IEC 24745 που δημοσιεύτηκε το 2011 [9] αναφέρει όλα τα πλέον αποδεκτά μοντέλα βιομετρικών συστημάτων που μπορούν να χρησιμοποιηθούν σε πραγματικές συνθήκες. Το έγγραφο είναι ακόμα σε ισχύ και πολλά συστήματα εφαρμόζουν τις αρχιτεκτονικές που προτείνει. Αυτές διαχωρίζονται με βάση δύο κύριους παράγοντες. Ο πρώτος αποτελεί τη περιοχή σύγκρισης των βιομετρικών στοιχείων και ο δεύτερος τη περιοχή αποθήκευσης αυτών. Ως περιοχή εννοείται το μέρος του συστήματος, δηλαδή, είτε στον server ή στον client είτε σε ένα token.

Server ή αλλιώς εξυπηρετητής, είναι ένα μηχάνημα με ειδικό λογισμικό που αναλαμβάνει την παροχή διαφόρων υπηρεσιών. Ο ρόλος του είναι να «εξυπηρετεί» αιτήσεις άλλων προγραμμάτων, γνωστά ως πελάτες (clients) [16].

Client ή αλλιώς πελάτης, μπορεί να θεωρηθεί είτε κάποιο λογισμικό που επικοινωνεί και υποβάλλει αιτήματα στον εξυπηρετητή είτε ένας άλλος υπολογιστής που επικοινωνεί με τον server. Η δεύτερη περίπτωση προϋποθέτει και οι 2 υπολογιστές να είναι συνδεδεμένοι σε ένα δίκτυο. [16]

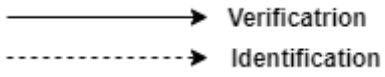
Token είναι μια μικρή φορητή συσκευή που επιτρέπει πρόσβαση σε μια υπηρεσία δικτύου. Η συσκευή αυτή μπορεί να αποτελέσει οτιδήποτε μπορεί να αποθηκεύσει, στη περίπτωση αυτή, βιομετρικά χαρακτηριστικά ή στοιχεία που θα χρησιμοποιηθούν για τη σύγκριση των βιομετρικών δεδομένων στο

εκάστοτε βιομετρικό σύστημα. Παραδείγματα αυτών μπορεί να είναι ένα USB stick, μια έξυπνη κάρτα ή ένα e-passport. [17]

Αναφορά Ταυτότητας είναι η εικονική οντότητα του χρήστη σε ένα σύστημα ή υπηρεσία και σχετίζεται με δεδομένα που είναι αποθηκευμένα σε μία βάση δεδομένων ή ένα αποθηκευτικό χώρο.

Βιομετρική Αναφορά αποτελεί το σύνολο των δεδομένων που έχουν εισαχθεί από ένα βιομετρικό χαρακτηριστικό και έχουν δεχτεί επεξεργασία ώστε να είναι συγκρίσιμα. Τα σύνολα αυτών των δεδομένων (βιομετρικές αναφορές) συγκρίνονται στα βιομετρικά συστήματα για τη λήψη απόφασης μιας ταυτοποίησης ή επαλήθευσης χρήστη.

Σε αυτό το σημείο θα πρέπει να αναφερθεί πως, στα παρακάτω διαγράμματα, τα διακεκομμένα βελάκια υποδηλώνουν την ταυτοποίηση και τα μη διακεκομμένα, την επαλήθευση χρήστη.

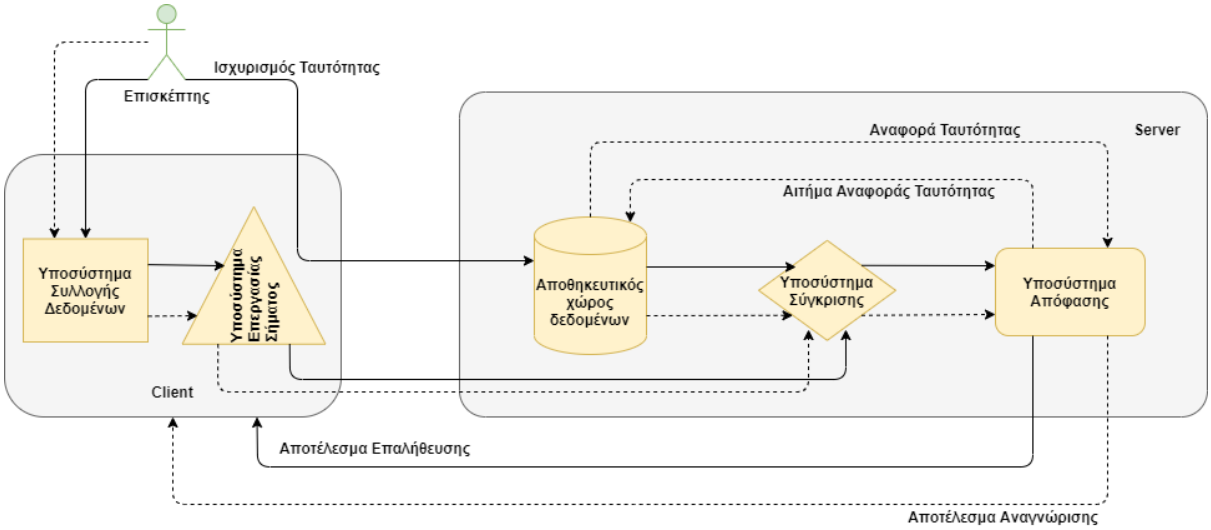


5.3.1 **Αποθήκευση στον server και σύγκριση στον server**

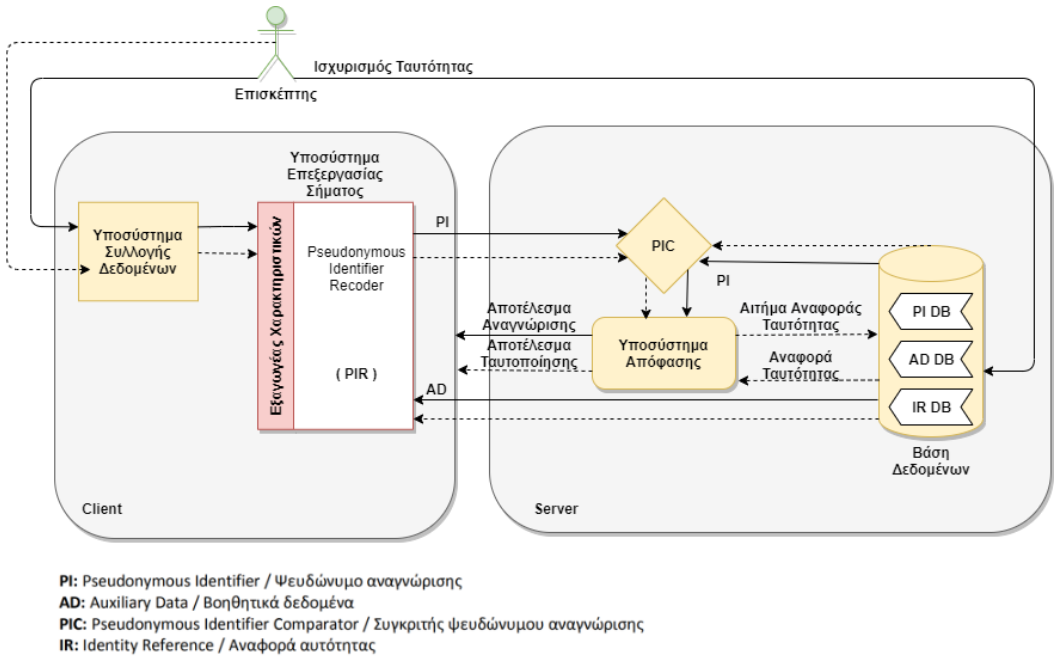
Σύμφωνα με το μοντέλο αυτό, τα βιομετρικά δεδομένα συγκρίνονται και αποθηκεύονται μόνο στον server. Η διαδικασία περιλαμβάνει την εξαγωγή ή τη λήψη του βιομετρικού χαρακτηριστικού στον client, όταν ο χρήστης επιθυμεί να ταυτοποιήσει ή να επαληθεύσει τη ταυτότητά του. Από το χαρακτηριστικό αυτό εξάγονται δεδομένα όπου και στέλνονται στον server για σύγκριση. Η σύγκρισή μπορεί να γίνει είτε υπολογίζοντας την ομοιότητα ενός ήδη αποθηκευμένου συνόλου βιομετρικών δεδομένων που αποθηκεύτηκαν κατά την εγγραφή του χρήστη είτε δημιουργώντας και αποθηκεύοντας ένα καινούργιο σύνολο δεδομένων, κάθε φορά που ο χρήστης υποβάλει κάποιο χαρακτηριστικό του για αναγνώριση. Σε περίπτωση που ο χρήστης δεν προϋπήρχε στο σύστημα, τότε θα γίνεται αποθήκευση αυτών των δεδομένων σε μια βάση δεδομένων που είναι εγκατεστημένη στον server.

Το μοντέλο αυτό χρησιμοποιείται για ταυτοποίηση και επαλήθευση δειγμάτων. Ένα σύστημα, για να υλοποιήσει αυτό το μοντέλο, προϋποθέτει την εξασφάλιση ασφάλειας των δεδομένων που θα αποθηκεύει. Επίσης, η ασφάλεια δικτύου αποτελεί έναν καθοριστικό παράγοντα για την επιτυχία αυτής της αρχιτεκτονικής.

Παρακάτω παρουσιάζονται διαγράμματα και για τις δύο περιπτώσεις σύγκρισης:



Σχήμα 5.2: Αποθήκευση στον server και σύγκρισή στον server με προϋπάρχουσα βιομετρική αναφορά

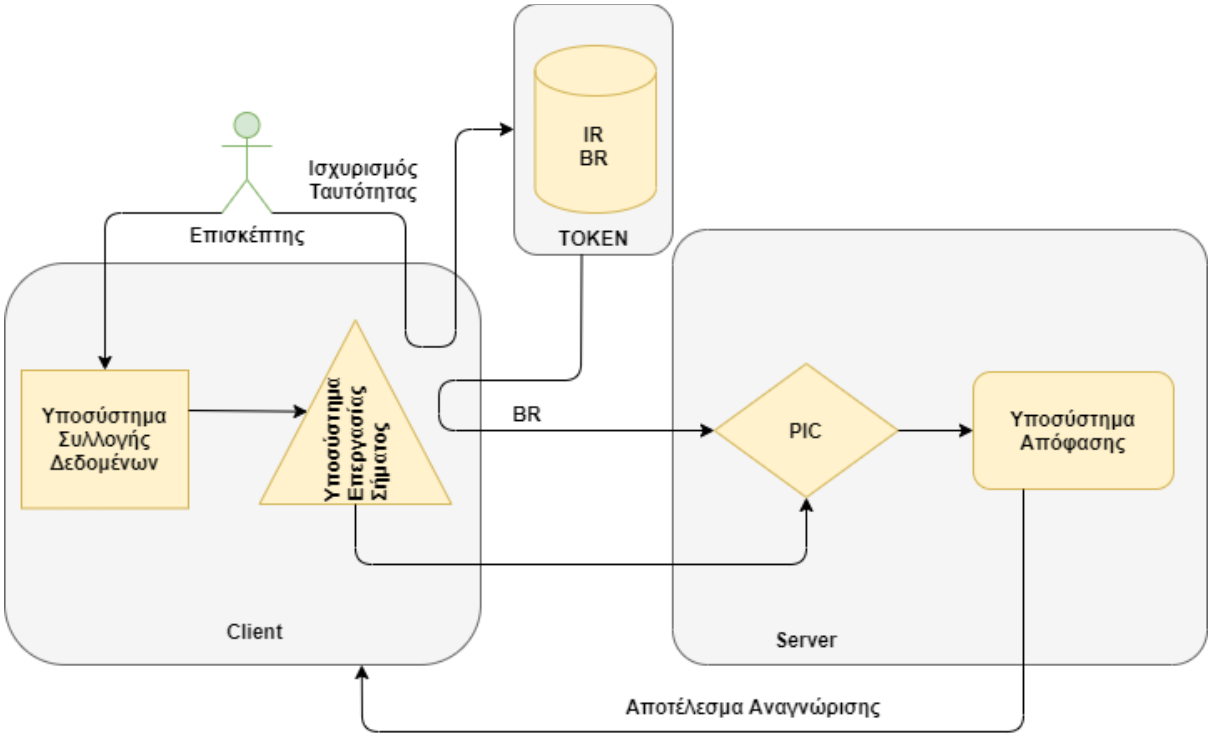


Σχήμα 5.3 : Αποθήκευση στον server και σύγκρισή στον server με χρήση νέας βιομετρικής αναφοράς

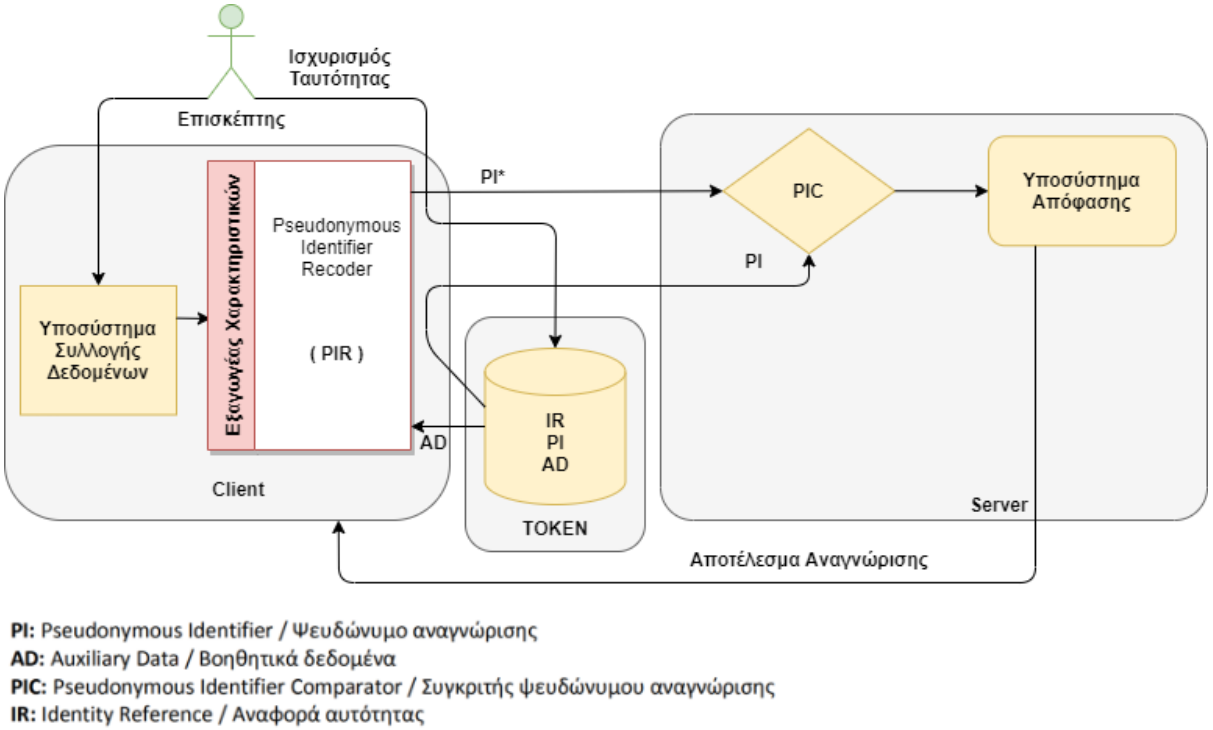
5.3.2 **Αποθήκευση σε Token και σύγκριση στον server**

Σε αυτή τη περίπτωση χρησιμοποιείται ένα Token που θα περιέχει ένα βιομετρικό χαρακτηριστικό. Το token αυτό θα το έχει στη κατοχή του μόνο ο χρήστης που το δικαιούται. Υποβάλλοντας αυτό το token στον client, καταγράφονται τα βιομετρικά χαρακτηριστικά και γίνεται ή κατάλληλη επεξεργασία. Ύστερα, αυτά αποστέλλονται στον server όπου και θα πραγματοποιηθεί η σύγκριση. Το μοντέλο αυτό είναι κατάλληλο για επαλήθευση (verification) και όχι για ταυτοποίηση, καθώς η σύγκριση γίνεται μόνο μέσω μιας βιομετρικής αναφοράς. Ο χρήστης, δηλαδή, τίθεται να αποδείξει ότι είναι αυτός που

ισχυρίζεται ότι είναι. Η τελική διαδικασία γίνεται ως εξής: Ο client στέλνει τα επεξεργασμένα δεδομένα στον server και το Token στέλνει σήμα στον server ώστε να ξέρει με τι θα συγκρίνει αυτά τα δεδομένα. Η δεύτερη μέθοδος σύγκρισης που μπορεί να πραγματοποιηθεί, είναι με τη χρήση μιας νέας, κάθε φορά, βιομετρικής αναφοράς. Κατα τη διάρκεια της εγγραφής, δημιουργείται ένα ψευδώνυμο αναγνώρισης (PI), το οποίο αποθηκεύεται στο token. Οπότε, στον server στέλνεται για σύγκριση το καινούργιο ψευδώνυμο αναγνώρισης μαζί με ένα καινούργιο ψευδώνυμο που κατασκευάστηκε από το βιομετρικό χαρακτηριστικό. Παρακάτω φαίνονται τα διαγράμματα σχετικά με τη χρήση αυτών των δύο μεθόδων.



Σχήμα 5.4: Αποθήκευση σε token και σύγκρισή στον server με χρήση υπάρχουσας βιομετρικής αναφοράς

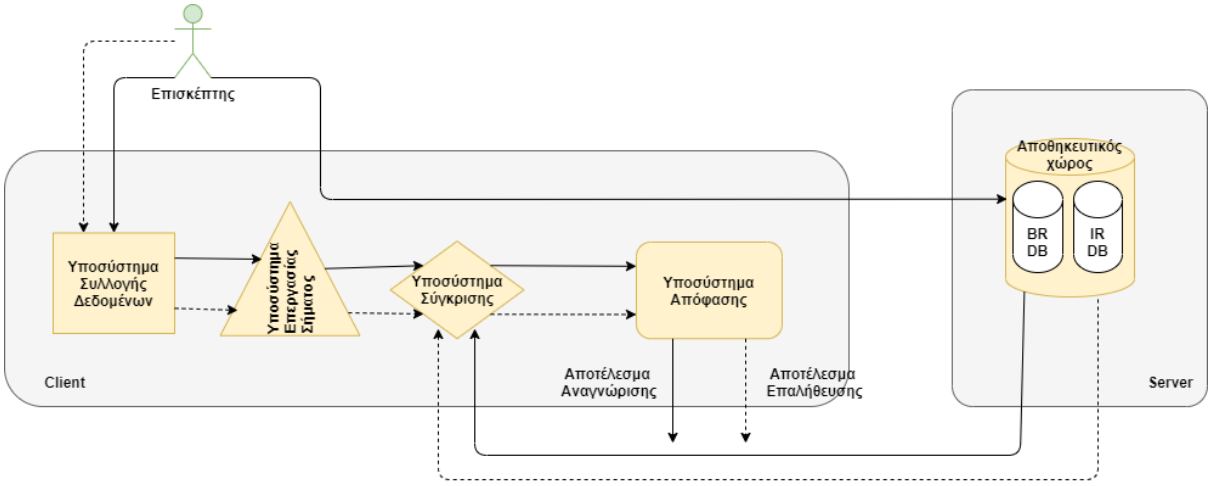


Σχήμα 5.5:Αποθήκευση σε token και σύγκρισή στον server με χρήση νέας βιομετρικής αναφοράς

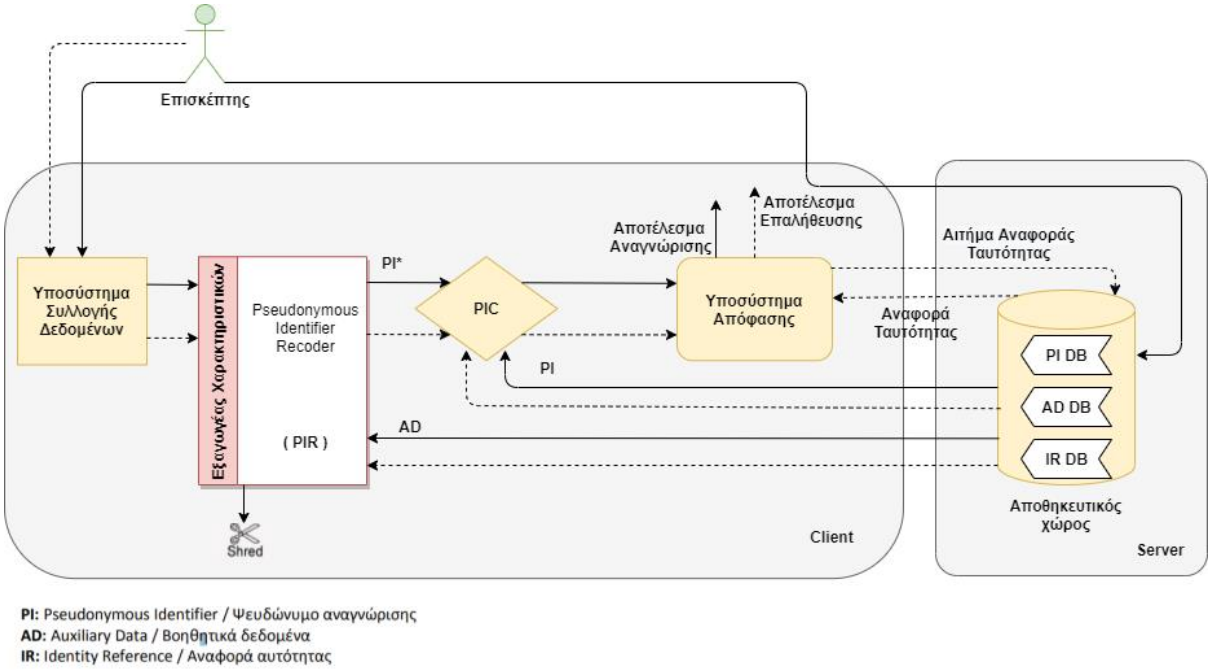
5.3.3 **Αποθήκευση στον server και σύγκριση στον client**

Το μοντέλο αυτό χρησιμοποιείται και για ταυτοποίηση και για επαλήθευση χρήστη. Όπως και στα προηγούμενα μοντέλα, δημιουργείται και αποθηκεύεται μια αναφορά ταυτότητας του χρήστη κατά τη διαδικασία της εγγραφής του στο σύστημα. Όταν κάποιος χρήστης θέλει να πιστοποιήσει την ταυτότητά του, υποβάλει το βιομετρικό του χαρακτηριστικό στον client. Ύστερα από τη κατάλληλη επεξεργασία του χαρακτηριστικού, ο client σύμφωνα με τα δεδομένα που προέκυψαν, θα ζητήσει την αντίστοιχη αναφορά ταυτότητας από τον Server. Τέλος μετά το αίτημα αυτό, ο server αποστέλλει την αποθηκευμένη αναφορά που έχει, ώστε να γίνει σύγκριση αυτών στον client. Το μοντέλο αυτό προϋποθέτει ότι ο client θα έχει ένα βιομετρικό αισθητήρα για καταγραφή και πως ο server από τον οποίο γίνεται η επικοινωνία, να αποτελεί έμπιστη και ασφαλής πηγή.

Η σύγκριση των αναφορών μπορεί να πραγματοποιηθεί με τη χρήση προϋπάρχουσας ή νέας βιομετρικής αναφοράς.



Σχήμα 5.6: Αποθήκευση στον server και σύγκρισή στον client με χρήση προϋπάρχουσας βιομετρικής αναφοράς



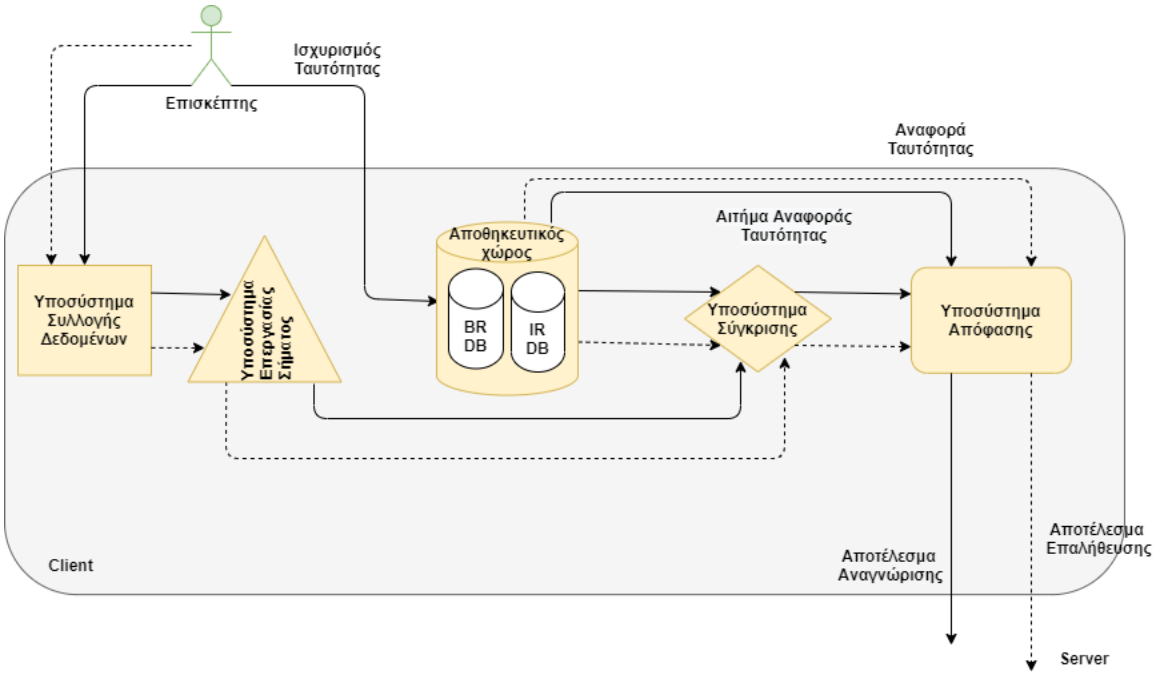
Σχήμα 5.7 : Αποθήκευση στον server και σύγκρισή στον client με χρήση νέας βιομετρικής αναφοράς

5.3.4 **Αποθήκευση στον client και σύγκριση στον client**

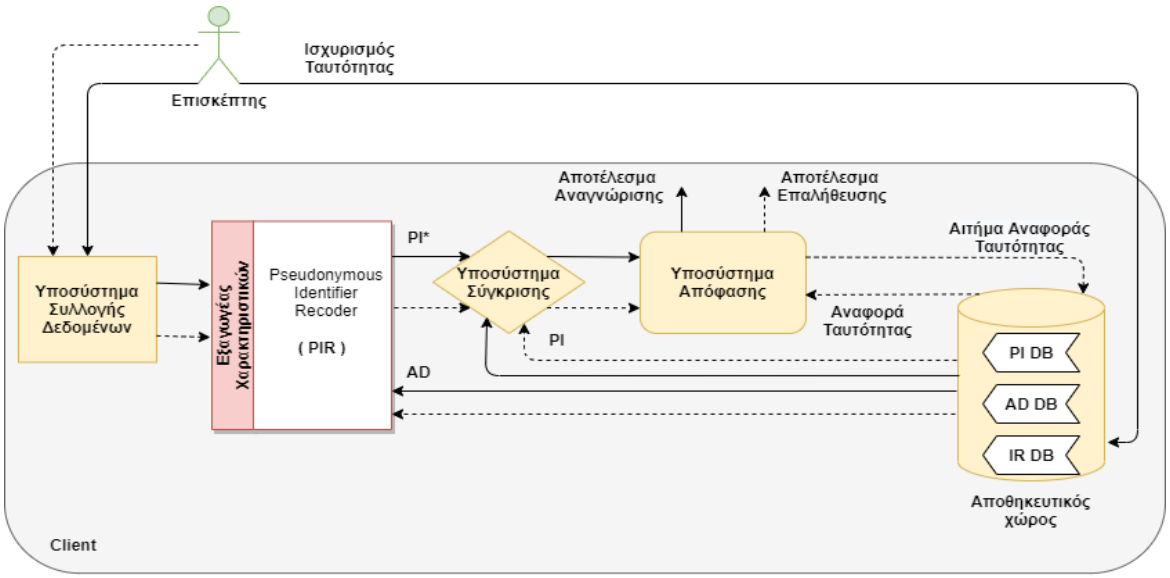
Το μοντέλο αυτό ενδείκνυται και για επαλήθευση και για ταυτοποίηση. Είναι απαραίτητο ο client να διαθέτει έναν αισθητήρα για την εξαγωγή των βιομετρικών χαρακτηριστικών, αλλά και ένα αλγόριθμο σύγκρισης βιομετρικών αναφορών και απόφασης κατά τη σύγκριση αυτών. Όταν ένας χρήστης εγγράφεται στη βάση δεδομένων στο server, τότε δημιουργείται μια αναφορά ταυτότητας και μια βιομετρική αναφορά στον client. Οπότε, όταν ο χρήστης επιθυμεί να ταυτοποιηθεί ή να αυθεντικοποιηθεί, γίνεται λήψη, επεξεργασία και εξαγωγή βιομετρικών δεδομένων από τον αισθητήρα και συγκρίνονται με τη βιομετρική αναφορά που είναι ήδη αποθηκευμένη στον client. Ύστερα, ο client παίρνει την απόφαση και αυτή αποστέλλεται στο server. Το σύστημα αυτό δεν προαπαιτεί τη ύπαρξη

server για να χρησιμοποιηθεί σε ορισμένες περιπτώσεις. Σε άλλες όμως περιπτώσεις, μπορεί και ο ίδιος ο server να επιβεβαιώνει το αποτέλεσμα της απόφασης του client για περισσότερη ασφάλεια.

Το πρότυπο αυτό δεν έχει μεγάλες απαιτήσεις ασφάλειας δικτύου, καθώς δεν μεταφέρονται δεδομένα χρήστη στον Server. Ωστόσο, συνίσταται η χρήση μιας αξιόπιστης βάσης δεδομένων και η δημιουργία και χρήση νέας βιομετρικής αναφοράς κάθε φορά. Συνήθως, χρησιμοποιείται σε προσωπικές συσκευές, όπως είναι ένας υπολογιστής ή ένα κινητό.



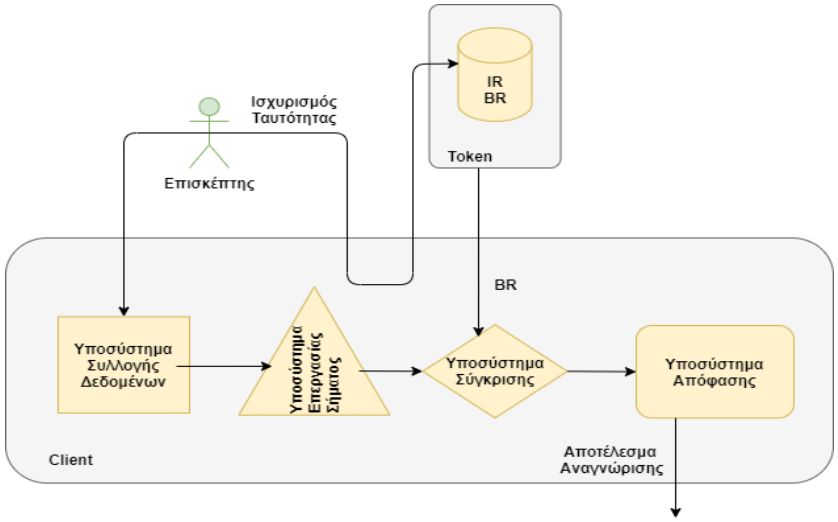
Σχήμα 5.8: Αποθήκευση στον client και σύγκρισή στον client με χρήση προϋπάρχουσας βιομετρικής αναφοράς



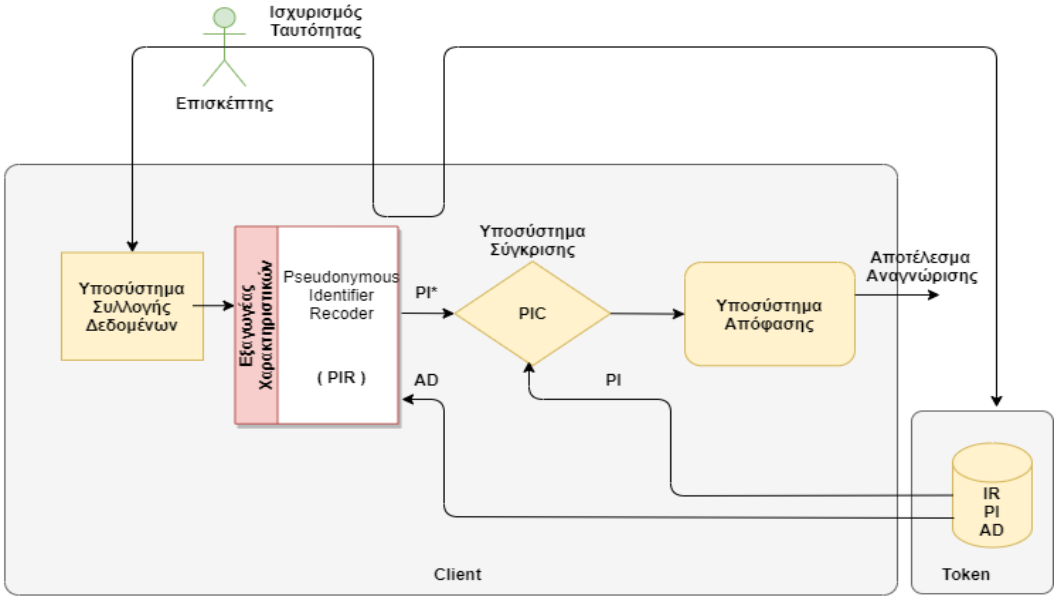
Σχήμα 5.9 : Αποθήκευση στον client και σύγκρισή στον client με χρήση νέας βιομετρικής αναφοράς

5.3.5 **Αποθήκευση σε token και σύγκριση στον client**

Το μοντέλο αυτό ενδείκνυται για επαλήθευση ενός χρήστη. Προϋποθέτει πως ο client θα διαθέτει έναν αλγόριθμο σύγκρισης βιομετρικών αναφορών και λήψης αποφάσεων κατά τη σύγκρισή τους. Επίσης, πρέπει να διαθέτει κάποιο βιομετρικό αισθητήρα και αλγόριθμο εξαγωγής βιομετρικών χαρακτηριστικών και δεδομένων. Κατά τη διάρκεια της εγγραφής του χρήστη, γίνεται συσχέτιση της βιομετρικής αναφοράς με την αναφορά ταυτότητας του. Το token, που θα πρέπει να έχει στη κατοχή του μόνο ο εξουσιοδοτημένος χρήστης, θα περιέχει το βιομετρικό χαρακτηριστικό, το οποίο θα παρέχεται στον client κατά τη διαδικασία ταυτοποίησης. Έτσι, όταν υποβληθεί το token στον client, γίνεται επεξεργασία του χαρακτηριστικού, λήψη της βιομετρικής αναφοράς, σύγκριση με την ήδη αποθηκευμένη στον client και λήψη απόφασης πιστοποίησης ή μη.



Σχήμα 5.10: Αποθήκευση σε token και σύγκρισή στον client με χρήση προϋπάρχουσας βιομετρικής αναφοράς

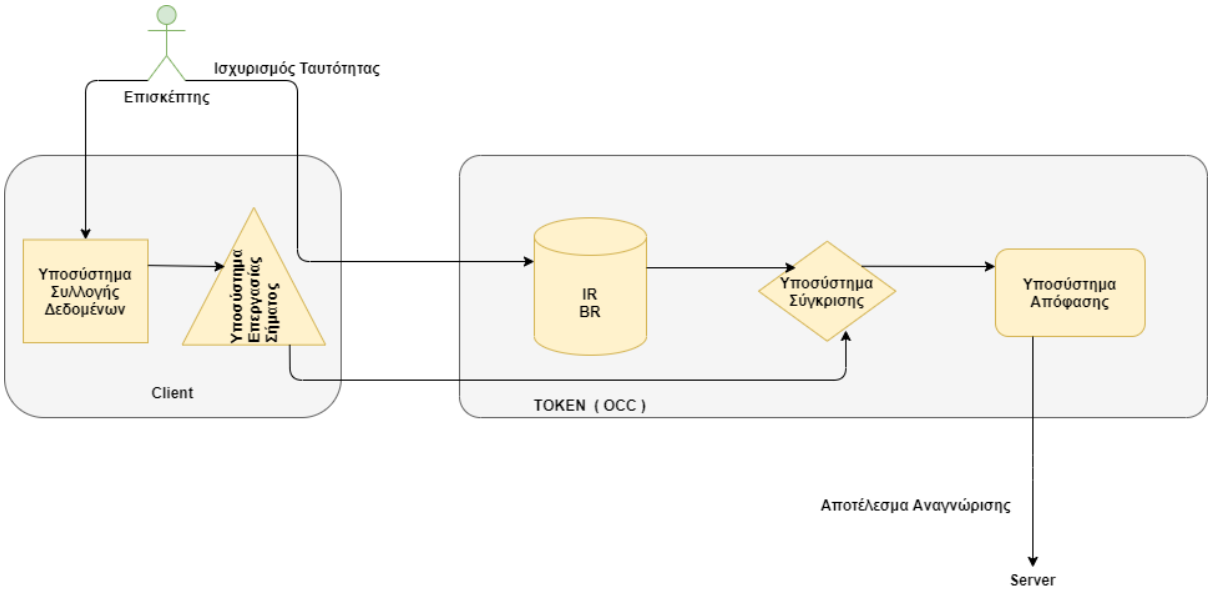


PI: Pseudonymous Identifier / Ψευδώνυμο αναγνώρισης
AD: Auxiliary Data / Βοηθητικά δεδομένα
IR: Identity Reference / Αναφορά αυτότητας

Σχήμα 5.11: Αποθήκευση σε token και σύγκρισή στον client με χρήση νέας βιομετρικής αναφοράς

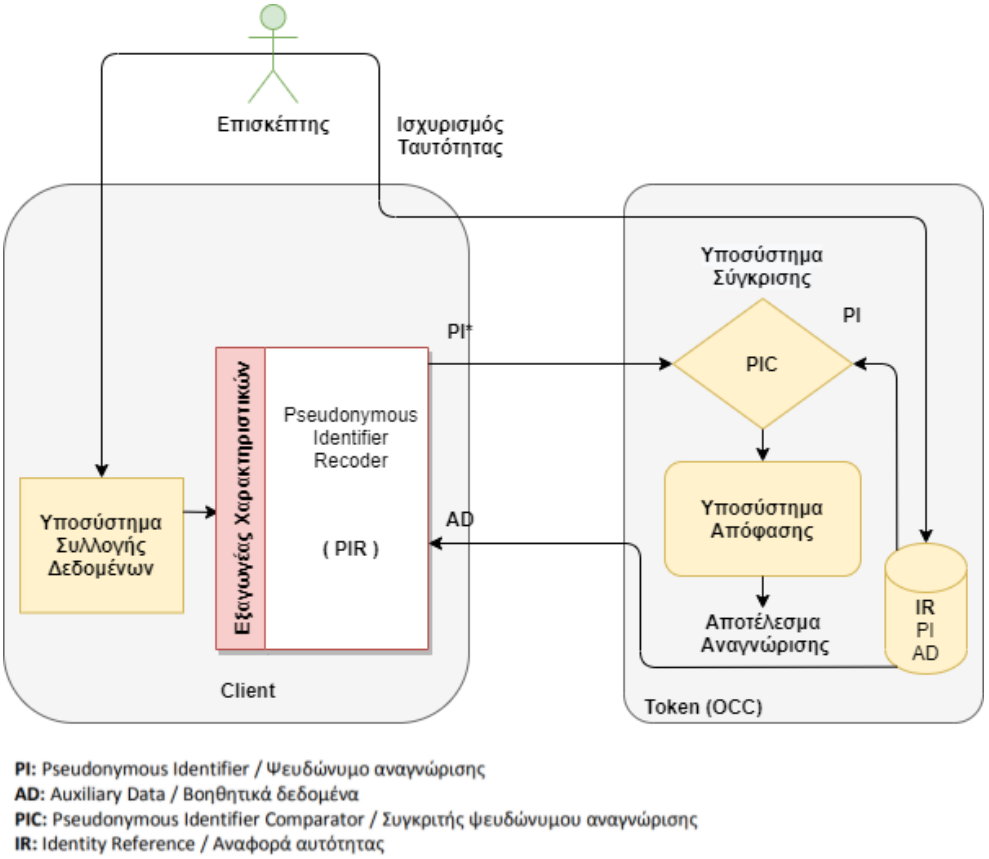
5.3.6 **Αποθήκευση σε token και σύγκριση σε token**

Το μοντέλο αυτό ενδείκνυται για επαλήθευση ενός χρήστη. Προϋποθέτει πως το token θα διαθέτει έναν αλγόριθμο σύγκρισης βιομετρικών αναφορών και λήψης αποφάσεων κατά τη σύγκρισή τους. Ο client πρέπει να διαθέτει κάποιο βιομετρικό αισθητήρα και αλγόριθμο εξαγωγής βιομετρικών χαρακτηριστικών και δεδομένων. Κατά τη διάρκεια της εγγραφής του χρήστη γίνεται συσχέτιση της βιομετρικής αναφοράς με την αναφορά ταυτότητας του. Το token που θα κάνει τη σύγκριση θα έχει αποθηκευμένα και τα βιομετρικά χαρακτηριστικά του χρήστη. Έτσι, όταν υποβληθεί ένα βιομετρικό χαρακτηριστικό στον client, γίνεται επεξεργασία αυτού και λήψη της βιομετρικής αναφοράς. Ύστερα, η βιομετρική αναφορά μεταφέρεται στο token για σύγκριση και λήψη απόφασης πιστοποίησης ή μη. Το τελικό αποτέλεσμα καταλήγει στον server. Το σύστημα αυτό αποτελεί το πιο ασφαλές, όσον αφορά την προστασία δεδομένων, διότι δεν μεταφέρονται δεδομένα στον server, ούτε γίνεται χρήση κάποιας βάσης δεδομένων.



Σχήμα 5.12: Αποθήκευση σε token και σύγκρισή στο token με χρήση υπάρχουσας βιομετρικής αναφοράς

Όπως και στα προηγούμενα μοντέλα, μπορεί και σε αυτό να γίνει σύγκριση με χρήση καινούργιας βιομετρικής αναφοράς. Σε αυτήν τη περίπτωση, στέλνονται από το token στον client, βοηθητικά δεδομένα για τη επεξεργασία των βιομετρικών χαρακτηριστικών. Το ψευδώνυμο αναγνώρισης βρίσκεται στο token, όπου θα βοηθήσει στη σύγκριση μετά την αποστολή των βοηθητικών δεδομένων.



Σχήμα 5.13: Αποθήκευση σε token και σύγκρισή στο token με χρήση νέας βιομετρικής αναφοράς

5.3.7 Άλλες περιπτώσεις

Σε αυτό το σημείο αξίζει να αναφερθούν άλλες δύο ακόμα περιπτώσεις, κατά τις οποίες, η αποθήκευση της βιομετρικής αναφοράς είναι κατανεμημένη. Η πρώτη περίπτωση αποτελεί την κατανεμημένη αποθήκευση σε token και σε server και η σύγκριση γίνεται στον server. Η δεύτερη αποτελεί την κατανεμημένη αποθήκευση σε token και σε server, ενώ η σύγκριση γίνεται στον client. Το πλεονέκτημα και των δύο περιπτώσεων είναι ότι η βιομετρική αναφορά αποθηκεύεται σε δύο αποθηκευτικούς χώρους, πράγμα που κάνει την επαλήθευση πιο έγκυρη και πιο ασφαλή. Το πρώτο μοντέλο, συνήθως, χρησιμοποιείται σε συστήματα που σχετίζονται με συναλλαγές, όπως για παράδειγμα μια e-banking εφαρμογή. Το δεύτερο μοντέλο χρησιμοποιείται σε δημόσιους χώρους, όπως αεροδρόμια ή υπηρεσίες που απαιτείται επαλήθευση ταυτότητας, όπως για παράδειγμα ή επαλήθευση ταυτότητας στα σύνορα.

5.4 Επίλογος

Σε αυτό το κεφάλαιο, γίνεται αναφορά για το ποια είναι αυτά τα χαρακτηριστικά που διέπουν ένα βιομετρικό σύστημα. Ύστερα, γίνεται επεξήγηση για όλα τα είδη κοινώς αποδεκτών αρχιτεκτονικών βιομετρικών συστημάτων με κείμενο και απεικόνιση αντίστοιχων διαγραμμάτων. Το κεφάλαιο αυτό είναι ιδιαίτερα σημαντικό, καθώς βοηθάει στην επιλογή αρχιτεκτονικής υλοποίησης της πιλοτικής εφαρμογής. Περισσότερα πάνω σε αυτό, αναφέρονται στο κεφάλαιο 6.

Κεφάλαιο 6ο: Τεχνολογίες As A Service (AAS)

6.1 Εισαγωγή

Ανέκαθεν, μια ερώτηση πολλών ιδιοκτητών μικρών επιχειρήσεων, σχετικά με τον κλάδο της πληροφορικής, περιλάμβανε, σχεδόν πάντα, την τιμή για την εξασφάλιση καλύτερου, νεότερου και γρηγορότερου hardware ή λογισμικού έτσι ώστε να παραμένουν ενημερωμένοι, σχετικά με τις απαιτήσεις επεξεργασίας δεδομένων των πελατών τους.

Τα τελευταία χρόνια όμως, η μόδα στον κλάδο της πληροφορικής έχει μεταφερθεί προς τη κατεύθυνση του As-a-Service model. Με τις τεχνολογίες as a service, το πρόβλημα της υλοποίησης κοστοβόρων λύσεων κατά παραγγελία, λύνεται με τη χρήση συνδρομητικών υπηρεσιών, που υλοποιούνται με τέτοιο τρόπο, ώστε να ταιριάζουν σε ένα μεγάλο ποσοστό πελατών και να παρέχουν τελευταίας τεχνολογίας λύσεις, ενώ ταυτόχρονα ο πάροχος της υπηρεσίας as a service να κρατάει την ίδια δομή για όλους τους πελάτες του.

6.2 Τι είναι το as a service model

Πιο συγκεκριμένα, το μοντέλο as a service η αλλιώς **SaaS (Software as a service),** είναι ένα εγκεκριμένο λογισμικό που προσφέρει συγκεκριμένες υπηρεσίες τεχνολογίας με κάποιο συνδρομητικό πλάνο ή όχι. Μια υπηρεσία SaaS θεωρείται πως είναι hosted σε κάποιο server και διαθέσιμη σε όλους, μέσω του διαδικτύου. Αναφέρεται ως λογισμικό on-demand, δηλαδή κατ' απαίτηση ή αλλιώς είναι γνωστό ως web-based/web-hosted λογισμικό [28].

Το μοντέλο SaaS και οι υποκατηγορίες του, θεωρούνται μέρος του cloud computing ή αλλιώς το υπολογιστικό Νέφος. Μερικές υποκατηγορίες του είναι οι υλοποιήσεις infrastructure as a service (IaaS), platform as a service (PaaS), desktop as a service (DaaS), managed software as a service (MSaaS), mobile backend as a service (MBaaS), datacenter as a service (DCaaS), information technology management as a service (ITMaaS). Η υποκατηγορία που αφορά όμως την υλοποίηση της εφαρμογής της παρούσας πτυχιακής είναι η **Biometric as a service (BaaS).**

6.3 Biometric as a service

Μία biometric as a service υπηρεσία μπορεί να παρέχει τεχνολογίες όπως σάρωση δακτυλικών αποτυπωμάτων, αναγνώριση προσώπου και άλλες βιομετρικές τεχνολογίες. Με το BaaS, οι πελάτες μπορούν να έχουν πρόσβαση σε τέτοιου είδους υπηρεσίες, με συνδρομή ή όχι μέσω του διαδικτύου.

Με αυτόν τον τρόπο, οργανισμοί μπορούν να κάνουν τη διαδικασία αναγνώρισης και επιβεβαίωσης της ταυτότητας των πελατών τους πιο εύκολη, πιο γρήγορη, και πιο ασφαλή, χωρίς να χρειάζεται να υλοποιήσουν το δικό τους σύστημα βιομετρικής αναγνώρισης στη δική τους υποδομή λογισμικού και βάσης δεδομένων. Επίσης, τους παρέχεται μεγαλύτερη αποτροπή απάτης από ψεύτικους λογαριασμούς ή κακόβουλους χρήστες.

6.4 Επιλογή αρχιτεκτονικής βιομετρικού συστήματος πιλοτικής εφαρμογής

Όπως λοιπόν είναι αντιληπτό, η πιλοτική εφαρμογή της παρούσας πτυχιακής αποτελεί ένα Biometric as a service, όπου παρέχει βιομετρική αυθεντικοποίηση προσώπου για άλλους οργανισμούς ή υπηρεσίες μέσω του προτύπου διαλειτουργικότητας OAuth2.0. Η παρούσα υλοποίηση δεν έχει

προδιαγράψει την πίστωση οργανισμών με κάποιου είδους συνδρομή. Ωστόσο, αυτό θα μπορούσε να αποτελεί ένα επιπλέον χαρακτηριστικό σε μια μελλοντική έκδοσή της.

Αρχικά, για τη σωστή υλοποίηση μιας τέτοιας υπηρεσίας, χρειάζεται η κατάλληλη επιλογή αρχιτεκτονικής, ώστε να διασφαλίζονται η χρησιμότητα και η ασφάλεια που υπόσχεται το μοντέλο **SaaS** και εκτενέστερα το **BaaS.** Τυπικά, οποιαδήποτε υπηρεσία που κάνει χρήση της βιομετρίας για αυθεντικοποίηση ή ταυτοποίηση, αποτελεί ένα βιομετρικό σύστημα. Συνεπώς, μια υπηρεσία **BaaS** αποτελεί και αυτή ένα Βιομετρικό σύστημα.

Λαμβάνοντας υπόψην τη θεωρία του κεφαλαίου 4, έγινε εκτενής μελέτη για το είδος και την αρχιτεκτονική του βιομετρικού συστήματος που θα ακολουθηθεί γι' αυτήν την περίπτωση. Σύμφωνα με τη λίστα των αρχιτεκτονικών που αναφέρθηκαν, ακολουθήθηκε η αφαιρετική μέθοδος, αρχικά με το ποιες αρχιτεκτονικές δεν ταιριάζουν καθόλου σε μια τέτοια υλοποίηση μοντέλου **BaaS.** Συνεπώς, όλες οι αρχιτεκτονικές που αφορούν την ύπαρξη κάποιου token έχουν απορριφθεί. Όπως αναφέρθηκε στο κεφάλαιο 4, ένα token αποτελεί μια μικρή φορητή συσκευή που επιτρέπει πρόσβαση σε μια υπηρεσία δικτύου και έχει τη δυνατότητα συγκράτησης και αποθήκευσης δεδομένων. Η παρούσα υλοποίηση δεν απαιτεί την ύπαρξη κάποιου token, οπότε απορρίπτονται οι αρχιτεκτονικές :

- Αποθήκευση σε token και σύγκριση στον server
- Αποθήκευση σε token και σύγκριση στον client
- Αποθήκευση σε token και σύγκριση σε token
- Κατανεμημένη αποθήκευση σε token και σε server και η σύγκριση στον server
- Κατανεμημένη αποθήκευση σε token και σε server και η σύγκριση στον client

Σε δεύτερη φάση, απορρίφθηκε η αρχιτεκτονική βιομετρικού συστήματος σύγκριση στο client και αποθήκευση στο client. Καθώς η παρούσα υλοποίηση αφορά τη χρήση βιομετρικών δεδομένων στο cloud, συνεπώς στο διαδίκτυο, απαιτείται η διασφάλιση της ακεραιότητας των βιομετρικών δεδομένων που διακινούνται από υπηρεσία σε υπηρεσία. Στην περίπτωση αυτή, ο client θα αποτελούσε έναν web browser. Δε θα ήταν λοιπόν συνετό να αποθηκεύεται μια τέτοιου είδους πληροφορία εκεί, καθώς θα μπορούσε να είναι προσβάσιμη από οπουδήποτε και από οποιονδήποτε. Ωστόσο, αυτή η αρχιτεκτονική θα ήταν χρήσιμη για μια smartphone εφαρμογή, όπου ο client θα αποτελούσε την ίδια τη συσκευή. Έτσι, η βιομετρική εφαρμογή θα ήταν προσβάσιμη μόνο από τον κάτοχο της.

Τελικά, η λογική καταλήγει στα δύο τελευταία αρχιτεκτονικά πρότυπα της λίστας που έμειναν. Τα πρότυπα-ανταγωνιστές που ταιριάζουν ταυτόχρονα στη παρούσα εργασία είναι τα εξής:

1. Σύγκριση στον server και αποθήκευση στο server
2. Σύγκριση στο client και αποθήκευση στο server

Από τα δύο πρότυπα ανταγωνιστές επιλέχθηκε το πρώτο, δηλαδή η σύγκριση στον server και αποθήκευση στο server. Η δεύτερη λύση, αν και αποδοτική, απορρίφθηκε καθώς προϋπέθετε την αποστολή των προσωπικών βιομετρικών χαρακτηριστικών του κάθε χρήστη στον client του browser, κάθε φορά που θα γίνεται κάποια είσοδος στο σύστημα. Αν και θεωρητικά δεν υπάρχει κάποιος κίνδυνος υποκλοπής της πληροφορίας αυτής, αφότου γίνει η μεταφορά από το server στο client, διατρέχει, ωστόσο, μεγάλο κίνδυνο να δεχτεί κάποιος χρήστης επίθεση τύπου man-in-the-middle. Η επίθεση **Man-In-The-Middle** (MITM) μπορεί να υποκλέψει μια επικοινωνία μεταξύ δύο συστημάτων.

Για παράδειγμα, σε μια **HTTP** επικοινωνία, όπου ο στόχος είναι η σύνδεση TCP μεταξύ πελάτη και διακομιστή. Χρησιμοποιώντας διάφορες τεχνικές, ο εισβολέας χωρίζει την αρχική σύνδεση TCP σε 2 νέες συνδέσεις, η μία μεταξύ του πελάτη και του εισβολέα και η άλλη μεταξύ του εισβολέα και του διακομιστή. Μόλις υποκλαπεί η σύνδεση TCP, ο εισβολέας ενεργεί ως ένας διακομιστής μεσολάβησης ή αλλιώς proxy server, που μπορεί να διαβάσει, να εισάγει και να τροποποιήσει τα δεδομένα στην επικοινωνία που έχει παραβιαστεί [18].

Το πρότυπο αρχιτεκτονικής, σύγκριση στον server και αποθήκευση στον server, δεν διατρέχει κίνδυνο από μια τέτοιου είδους επίθεση. Δεν θα ήταν αποτελεσματική, διότι και να υποκλέψει ένας εισβολέας μια πληροφορία, δε θα μπορεί να τη χρησιμοποιήσει είτε για μια πιθανή απόπειρα εισόδου του στο σύστημα ούτε για να μιμηθεί τη ταυτότητα του πραγματικού χρήστη.

6.5 Επίλογος

Σε αυτό το κεφάλαιο παρατίθεται ο ορισμός του του μοντέλου **SaaS**, η χρησιμότητα αυτού και, πιο συγκεκριμένα, οι ιδιότητες ενός **BaaS.** Λαμβάνοντας υπόψη τα παραπάνω, η μόνη υποψήφια και κατάλληλη αρχιτεκτονική βιομετρικού συστήματος που ταιριάζει στο μοντέλο **BaaS,** είναι η σύγκριση και η αποθήκευση στο server. Εφόσον η backend υλοποίηση στον hosted server τηρεί όλες τις προδιαγραφές ασφάλειας και τα απαραίτητα μέτρα προστασίας και ακεραιότητας δεδομένων των χρηστών, η εμπειρία του εκάστοτε χρήστη θα είναι απόλυτα ασφαλής και θα διαδραματίζεται σε ένα λειτουργικό και καλά καλωδιωμένο περιβάλλον.

Κεφάλαιο 7ο: Πρακτική εφαρμογή

7.1 Εισαγωγή

Σε αυτή την ενότητα, και με βάση τη θεωρία και τις πληροφορίες που εξετάσαμε στα προηγούμενα κεφάλαια, θα γίνει σχεδιασμός και ανάπτυξη της as a service εφαρμογής που υλοποιήθηκε στα πλαίσια της παρούσας πτυχιακής εργασίας. Θα εξεταστούν οι τεχνολογίες και οι μεθοδολογίες που χρησιμοποιήθηκαν, θα γίνει επεξήγηση της ροής της εφαρμογής αλλά και των δυνατοτήτων που θα παρέχει στους χρήστες. Τέλος, θα παρουσιαστούν ιδέες για το πως θα μπορούσε μια τέτοια υπηρεσία να εξελιχθεί για να εξυπηρετεί πελάτες σε πραγματικά δεδομένα.

Η εφαρμογή, όπως αναφέρεται και εξηγείται στο κεφάλαιο 6, αποτελεί Biometric as a Service, το οποίο λειτουργεί ως ένας authorization/authentication server. Αυτό σημαίνει πως τρίτες εφαρμογές έχουν τη δυνατότητα αυθεντικοποίησης των χρηστών τους μέσω της υπηρεσίας που παρέχει η εφαρμογή αυτής της πτυχιακής. Η περαιτέρω λειτουργικότητα που παρέχει η υπηρεσία, είναι η αυθεντικοποίηση του χρήστη μέσω των βιομετρικών χαρακτηριστικών προσώπου του, το οποίο είναι μια λειτουργικότητα που θα την καθιστά μοναδική, ανάμεσα σε άλλους authentication ή authorization servers. Τέλος, παρέχεται η δυνατότητα σε κάθε εγγεγραμμένο χρήστη να συνδέει και εφαρμογές μέσω ενός γραφικού περιβάλλοντος, παράγοντας μυστικά κλειδιά που θα μπορούν να χρησιμοποιηθούν κατά τη διαδικασία υλοποίησης του external authentication από τον developer της τρίτης εφαρμογής που συνδέεται.

Το όνομα που έχει επιλεχθεί γι' αυτή την web εφαρμογή είναι το " MyAuth ", το οποίο σημαίνει " η αυθεντικοποίησή μου". Είναι φτιαγμένη με τέτοιο τρόπο ώστε να προσελκύσει άλλους developers να τη χρησιμοποιήσουν, με απώτερο σκοπό την ενσωμάτωσή των υπηρεσιών της σε άλλες εφαρμογές που θα ήθελαν να αξιοποιήσουν τη βιομετρική αυθεντικοποίηση, ώς ένα παραπάνω μέτρο ασφάλειας και εμπιστοσύνης του πελάτη-χρήστη που θα του παραχθεί πρόσβαση σε αυτές.

7.2 Ανάλυση τεχνολογιών

Για την ανάπτυξη όλων των back-end υποσυστημάτων, χρησιμοποιείται ως γλώσσα προγραμματισμού η C#, και πιο συγκεκριμένα, χρησιμοποιείται το framework .Net Core στην έκδοση 3.1. Το συγκεκριμένο framework επιλέχθηκε λόγω της υψηλής απόδοσης, αλλά και την δυνατότητα εκτέλεσης των παραγόμενων εφαρμογών σε διαφορετικά λειτουργικά συστήματα.

Για την αυτοματοποίηση της διαδικασίας του deployment, σε όλα τα περιβάλλοντα που υλοποιηθούν μέχρι την τελική έκδοση του έργου, θα χρησιμοποιηθεί Github CI/CD, όπου κατά την διαδικασία της ανάπτυξης του λογισμικού, όλα τα επιμέρους μέρη του κώδικα που θα ολοκληρώνονται, θα εκτελούν μεμονωμένα τεστ και στην συνέχεια θα προωθούνται στο αντίστοιχο περιβάλλον.

Για τη βάση δεδομένων χρησιμοποιείται η PostgreSQL, η οποία είναι μια ιδανική λύση για οποιουδήποτε όγκου δεδομένα με ιδιαίτερη αξιοπιστία και με πλήρεις λύσεις αυτοματοποιημένων υπηρεσιών συγχρονισμού.

Παρακάτω, ακολουθούν γενικές περιγραφές για κάθε τεχνολογία που χρησιμοποιείται για την υλοποίηση του έργου.

7.2.1 Γλώσσα προγραμματισμού C#

Εικόνα 7.1: Λογότυπο C#

Η γλώσσα προγραμματισμού η C# είναι μία γλώσσα προγραμματισμού που δημιουργήθηκε από την Microsoft το 2000 για την πλατφόρμα .NET.

Είναι μία από τις γλώσσες προγραμματισμού που δημιουργήθηκαν για την Common Language Infrastructure, ένα ανοιχτών δεδομένων τεχνικό πρότυπο που δημοσιεύθηκε από τον Διεθνή Οργανισμό Προτύπων (ISO).

Παρόλο που ονομαστικά αναφέρεται στην "οικογένεια" C των γλωσσών προγραμματισμού, έχει περισσότερα κοινά με την Java παρά με άλλες γλώσσες C. Η C# χρησιμοποιείται ευρέως για παντός τύπου επιχειρηματικές λύσεις. Ωστόσο, κάποια από τα χαρακτηριστικά που την διαφοροποιούν από τις περισσότερες γλώσσες προγραμματισμού, είναι ότι έχει δημιουργηθεί με γνώμονα την παραγωγικότητα των προγραμματιστών και ταυτόχρονα την μέγιστη βελτιστοποίηση στις δομές της για θέματα ασφάλειας. Πρόκειται για μία γλώσσα ανοιχτού κώδικα με την υπογραφή – πλήρη υποστήριξη της Microsoft με ένα από τα μεγαλύτερα community προγραμματιστών.

7.2.2 .Net Core v3.1 framework

Εικόνα 7.2: Λογότυπο .Net Core

Το συγκεκριμένο framework εκδόθηκε το έτος 2016 και χρησιμοποιήθηκε άμεσα καθώς μέσα από το ιδιαίτερα μεγάλο community, δόθηκε η δυνατότητα να δοκιμαστεί σε διαφορετικές εφαρμογές και να δοθεί μια ξεκάθαρη εικόνα για την απόδοση του framework, όπου άμεσα κρίθηκε – χαρακτηρίστηκε ως ένα framework με υψηλές επιδόσεις.

Ίσως το πιο βασικό χαρακτηριστικό του .Net Core framework είναι η υποστήριξη της δυνατότητας να δομηθεί κώδικας που θα είναι εκτελέσιμος σε πολλαπλά λειτουργικά συστήματα (Windows, Linux, Rhel), διαφορετικών εκδόσεων και αρχιτεκτονικών συμπεριλαμβανομένων x64, x86, ARM, έχοντας την ίδια συμπεριφορά ανεξαρτήτως του περιβάλλοντος εκτέλεσης. Είναι ένα λογισμικό ανοικτού κώδικα όπου παρόλα αυτά παραμένει κάτω από την μέριμνα του .NET Foundation της Microsoft.

Ένα ακόμα χαρακτηριστικό ιδιαίτερης σημασίας είναι ότι υποστηρίζει dependency injection, όπου με αυτό τον τρόπο και με συνδυασμό χρήσης διαφόρων βιβλιοθηκών, βελτιστοποιεί την υλοποίηση και εκτέλεση αυτοματοποιημένων τεστ. Με αυτόν τον τρόπο, δίνεται η δυνατότητα στους προγραμματιστές, κατά την δημιουργία διαφορετικών δομών της εφαρμογής, να δημιουργούνται και μεμονωμένα τεστ, ταυτόχρονα, καθώς και να μπορούν να προχωρήσουν σε συνδυαστικά τεστ με άλλες δομές που έχουν υλοποιηθεί.

Επίσης το .Net Core framework παρέχει την δυνατότητα να μπορεί να χρησιμοποιηθεί με μία πλατφόρμα λογισμικού ανοιχτού κώδικα (Docker) που παρέχει την δυνατότητα της εικονικοποίησης (virtualisation) σε επίπεδο λειτουργικών συστημάτων που επιτρέπει την ανάπτυξη του λογισμικού και προσομοίωση σε όλα τα περιβάλλοντα που θα χρησιμοποιηθούν καθώς και βοηθάει στην διαδικασία της αυτοματοποίησης του deployment της εφαρμογής.

7.2.3 **PostgreSQL**

Εικόνα 7.3: Λογότυπο PostgreSQL

Η PostgreSQL είναι μία σχεσιακή βάση ανοικτού κώδικα με ιδιαίτερα πολλές δυνατότητες, μέσα από την συνεχόμενη ανάπτυξη του λογισμικού της με διάρκεια πάνω από δύο δεκαετίες έχει αποδειχθεί ως μια από τις πιο αξιόπιστες λύσεις σε ακεραιότητα δεδομένων, γρήγορη λειτουργία καθώς και υψηλά στάνταρ σε επίπεδα ασφαλείας.

Κάποια από τα χαρακτηριστικά της είναι Online αντίγραφα ασφαλείας, υψηλότατοι έλεγχοι ασφάλειας και διαθεσιμότητα των δεδομένων, πλήρης υποστήριξη συναρτήσεων συγκεντρωτικών αποτελεσμάτων, πλήρης λύση υπηρεσιών συγχρονισμού (Replication).

7.2.4 **Github CI/CD**

Εικόνα 7.4: Λογότυπο GitHub

Μέσω του Github CI/CD μπορούμε να αυτοματοποιήσουμε τη διαδικασία τοποθέτησης στους εξυπηρετητές, του προετοιμασμένου (build), προς εκτέλεση κώδικα, ανάλογα με το πιο περιβάλλον αφορά τη στιγμή που ο εκάστοτε προγραμματιστής αποθηκεύει τον πηγαίο κώδικα.

Ως εργαλείο συντήρησης των διαφορετικών εκδόσεων του πηγαίου κώδικα του συγκεκριμένου έργου, συμπεριλαμβανομένων όλων των μελών της ομάδας, θα χρησιμοποιηθεί το Git, και ως αποθηκευτικός χώρος, θα χρησιμοποιηθεί η διαδικτυακή εφαρμογή Github.

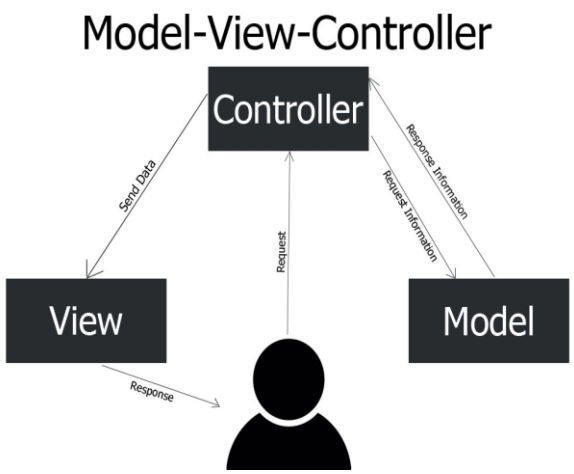
Για την υλοποίηση της διεπαφής της υπηρεσίας, έχει δημιουργηθεί μία φιλική προς τον χρήστη Εφαρμογή Ιστού (Web Application), εύχρηστη και λειτουργική ως προς όλες τις ενέργειες που θα πρέπει να διεκπεραιωθούν, με ιδιαίτερα εύκολη και φιλική διεπιφάνεια χρήσης (user interface). Για την ανάπτυξη της client εφαρμογής frontend, έχει γίνει χρήση του angular framework. Μαζί έχει γίνει χρήση των τεχνολογιών HTML5, CSS και JAVASCRIPT που κρίνονται απαραίτητα για την ανάπτυξη μιας client web εφαρμογής.

7.2.5 **Angular**

Εικόνα 7.5: Λογότυπο Angular

Η Angular είναι μία ανοιχτού λογισμικού πλατφόρμα ανάπτυξης για τη δημιουργία αποτελεσματικών και εξελιγμένων εφαρμογών ιστού. Δημιουργήθηκε και συντηρείται από την Google, καθώς και από μια μεγάλη παγκόσμια κοινότητα προγραμματιστών και εταιρειών. Είναι μία από τις πιο πολυχρησιμοποιημένες πλατφόρμες της εποχής μας. Είναι βασισμένη στην TypeScript της Microsoft, ένα υπερ-σύνολο της JavaScript, που εξασφαλίζει μεγαλύτερη ασφάλεια και την ελαχιστοποίηση προβλημάτων. Οι εφαρμογές, που αναπτύσσονται με την Angular, ακολουθούν την αρχιτεκτονική MVC (Model-View-Controller), η οποία εξυπηρετεί την επαναχρησιμοποίηση και την εύκολη συντήρηση του κώδικα. Συνολικά, προσφέρει ευκολία και ταχύτητα κατά την ανάπτυξη, διαθέτει κλιμακούμενη, παραμετροποίηση υποδομή και εγγυάται την ταχύτατη εκτέλεση των λειτουργιών της εφαρμογής.

7.3 Αρχιτεκτονική



Εικόνα 7.6: Αρχιτεκτονική MVC

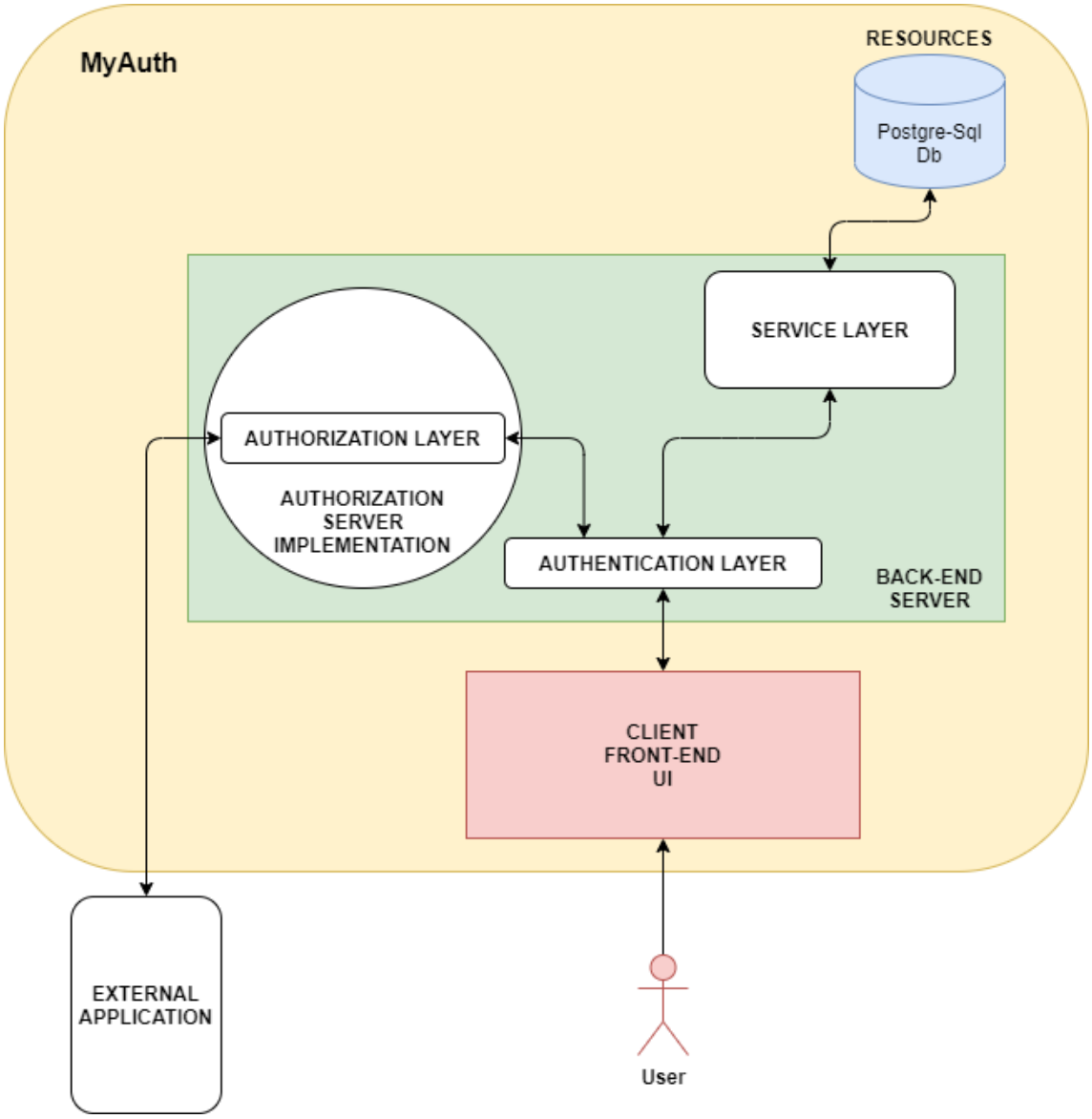
Στο σύνολο του έργου "MyAuth", όσον αφορά την ανάπτυξη λογισμικού για το Backend – Frontend ως αρχιτεκτονικό πρότυπο, θα χρησιμοποιηθεί το πρότυπο MVC (Model View Controller). Το συγκεκριμένο πρότυπο επιλέχθηκε, γιατί επιτρέπει την παράλληλη ανάπτυξη του λογισμικού από μεγάλες ομάδες προγραμματιστών, οι οποίες λειτουργούν ανεξάρτητα και προσφέρει πολύ καλό διαχωρισμό και ξεκάθαρο αρχιτεκτονικό διαχωρισμό του κώδικα των εφαρμογών (code base). Επίσης, λόγω της δομής του συγκεκριμένου προτύπου διευκολύνεται η επαναχρησιμοποίηση ήδη υπάρχοντος κώδικα, καθώς λόγω της αρχιτεκτονικής του δομής, χωρίζεται σε μικρότερες ξεχωριστές υπηρεσίες.

Το πρότυπο MVC είναι ένα πρότυπο που χρησιμοποιείται ευρέως σε διαδικτυακές εφαρμογές και αποτελείται από τρία βασικά επίπεδα:

- **Επίπεδο Model**, που περιλαμβάνει τα κεντρικά στοιχεία και την δομή των δεδομένων - οντοτήτων της εφαρμογής
- **Επίπεδο View**, που περιλαμβάνει τα βασικά στοιχεία και τα χαρακτηριστικά της απεικόνισης της πληροφορίας στους χρήστες
- **Επίπεδο Controller**, που περιλαμβάνει όλη τη λογική και τις διαδικασίες επεξεργασίας των δεδομένων και των λειτουργιών. Πολύ συχνά σε αυτό το επίπεδο θα υπάρχουν και Services τα οποία εξυπηρετούν, μεμονωμένα, δομές υπηρεσιών.

Το επιλεγμένο αρχιτεκτονικό πρότυπο σχεδίασης (MVC) υποστηρίζεται εγγενώς από το framework .Net Core που θα χρησιμοποιηθεί για την υλοποίηση των back-end υποσυστημάτων του έργου, όπως περιγράφεται αναλυτικά παραπάνω.

Παρακάτω παρουσιάζεται ένα ολοκληρωτικό διάγραμμα της αρχιτεκτονικής της εφαρμογής:



Σχήμα 7.1: Βασική Αρχιτεκτονική πιλοτικής εφαρμογής MyAuth

Στη συγκεκριμένη αρχιτεκτονική, η εξουσιοδότηση χρηστών μέσω τρίτων εφαρμογών, η αυθεντικοποίηση και η αποθήκευση των δεδομένων γίνεται σε μία υλοποίηση back-end server και μόνο. Σε άλλες υπηρεσίες, όπως αυτή της **Google,** υπάρχουν ξεχωριστά ένας authorization server και ένας resource server, οι οποίοι συνδέονται μεταξύ τους. Ωστόσο, όπως περιγράφεται και στην **ενότητα 3.2.2,** δεν είναι απαραίτητη μια τέτοια υλοποίηση σε αυτή την περίπτωση, καθώς η αποθήκευση των resources, μαζί με την λειτουργικότητα ενός authorization server, μπορεί να εξυπηρετηθεί από μία μεμονωμένη back-end υλοποίηση όπως είναι αυτή που παρουσιάζεται παραπάνω. Το **Authentication Layer** στηρίζει την επαλήθευση ενός χρήστη που είναι εγγεγραμμένος στην υπηρεσία. Αν, και εφόσον έχει κάποιος αυθεντικοποιηθεί, έχει πρόσβαση μέσω του **Service Layer**, στα resources της υπηρεσίας που αφορούν αυτόν. Κατά την αυθεντικοποίηση μέσω τρίτης εφαρμογής, γίνεται επικοινωνία μέσω του **Authorization Layer** το οποίο κατά την επιτυχή εξουσιοδότηση του User Agent που κάνει χρήση της τρίτης εφαρμογής, πραγματοποιεί αυθεντικοποίηση της οντότητας του χρήστη στο **Authentication Layer** και δίνεται πρόσβαση στα resources της υπηρεσίας, όπως περιγράφηκε και πριν. Τα resources αποθηκεύονται στη βάση δεδομένων του server. Τέλος, ο server εξυπηρετεί την front-end client

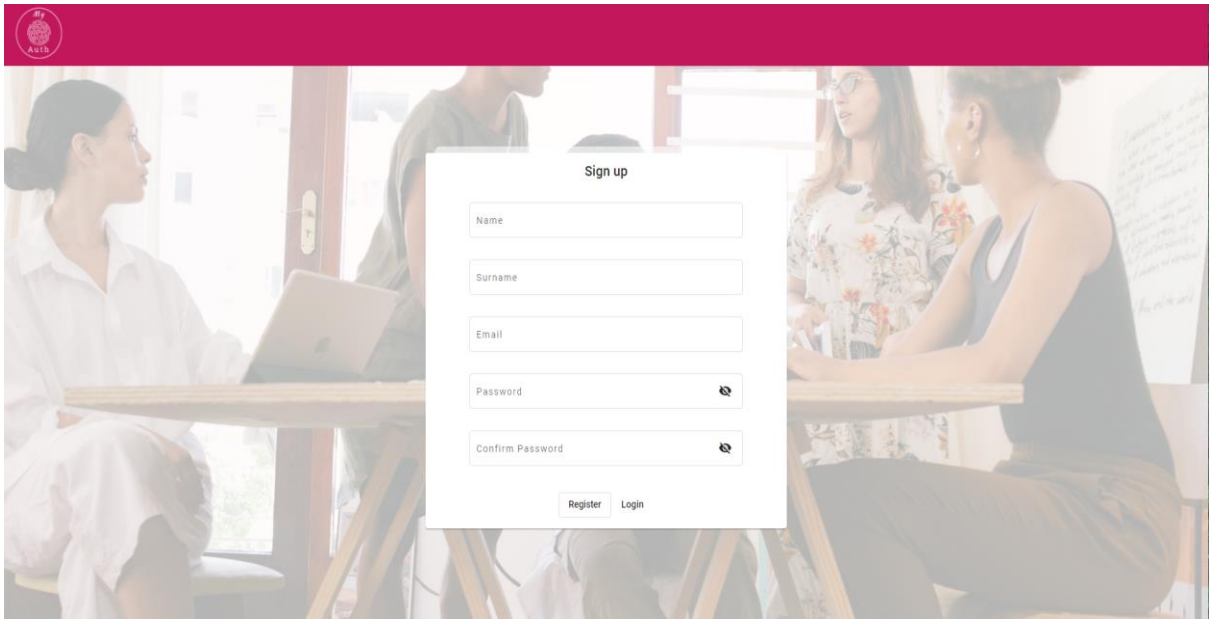
εφαρμογή που αποτελεί το γραφικό περιβάλλον με το οποίο αλληλεπιδρά ο κάθε χρήστης. Στη client εφαρμογή γίνεται και η ανάληψη των βιομετρικών χαρακτηριστικών προσώπου του χρήστη. Περισσότερες λεπτομέρειες θα αναφερθούν παρακάτω.

7.4 Application Mapping

Στα πλαίσια αυτού του σταδίου της μελέτης, θα παρουσιαστούν δύο πρωτογενή και απλά διαγράμματα ροής, με σκοπό την απεικόνιση των εκτελούμενων διαδικασιών, όπως διακρίνονται από τη χρήση της εφαρμογής.

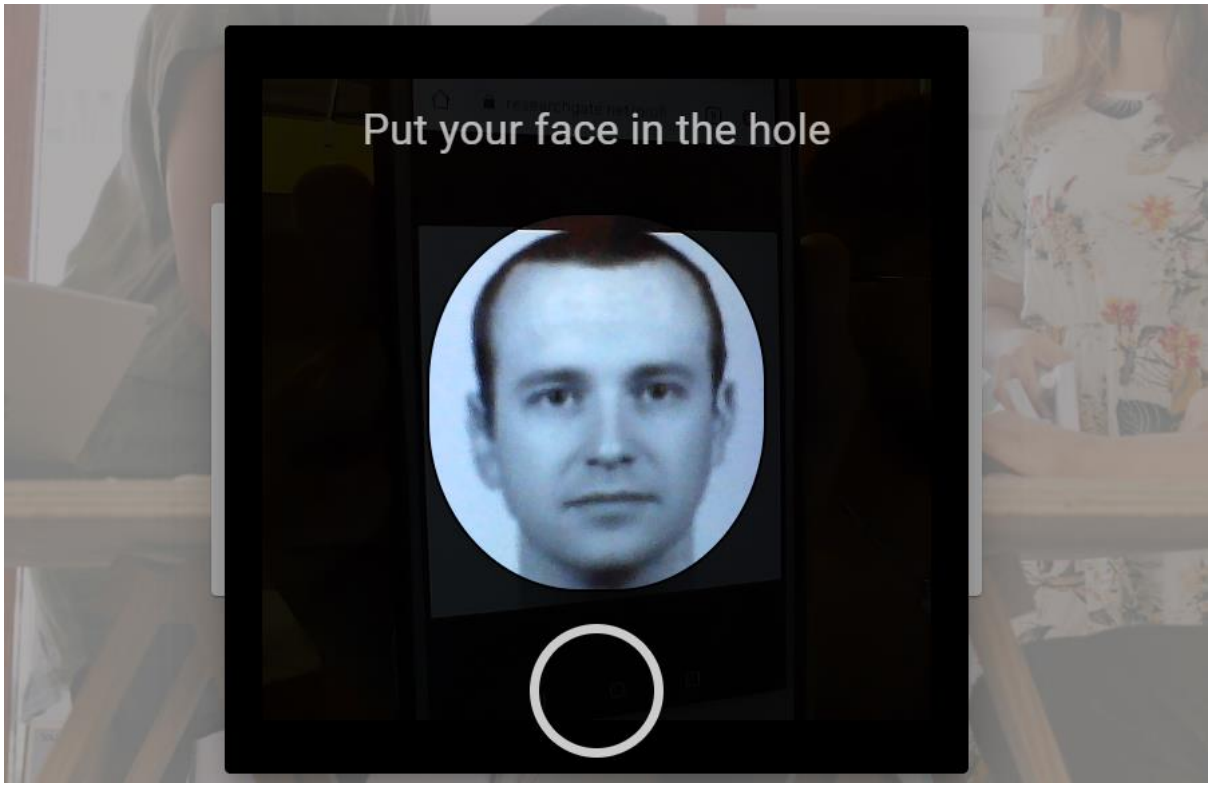
7.4.1 Sign Up

Μια διαδικασία που πρέπει να αναλυθεί, αποτελεί την εγγραφή του χρήστη στο σύστημα. Κατά την εγγραφή του χρήστη στην εφαρμογή, του ζητούνται διαπιστευτήρια σχετικά με το ονοματεπώνυμό του και το email του. Επίσης θα πρέπει να εισάγει έναν ισχυρό κωδικό πρόσβασης. Αρχικά, γίνεται ένας έλεγχος για το αν το email χρησιμοποιείται ήδη και αν ο κωδικός πρόσβασης τηρεί τις προϋποθέσεις. Οι προϋποθέσεις αυτές αφορούν τη χρήση τουλάχιστον 8 χαρακτήρων με ένα κεφαλαίο, ένα αριθμό και έναν ειδικό χαρακτήρα.



Εικόνα 7.7: Σελίδα εγγραφής χρήστη της υπηρεσίας MyAuth

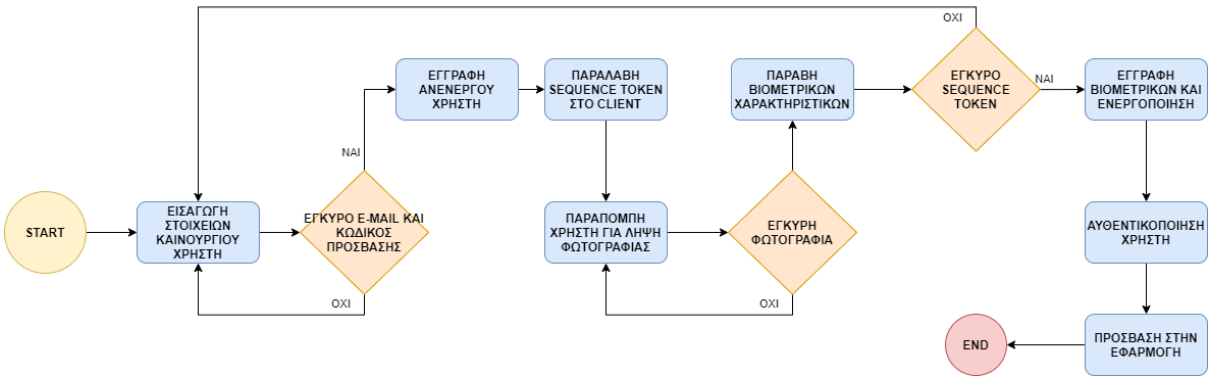
Εφόσον όλα τα στοιχεία που εισάγει ο χρήστης είναι εντάξει, γίνεται η εγγραφή του στη βάση δεδομένων. Παρόλα αυτά, δεν αποτελεί ακόμα ενεργό χρήστη, καθώς απομένει ακόμα ένα βήμα. Θα πρέπει να γίνει καταγραφή και αποθήκευση των βιομετρικών στοιχείων του προσώπου του. Μετά την προσωρινή εγγραφή του, ο client λαμβάνει ένα **Sequence Token.** Αυτό γίνεται καθώς, κατά την αποστολή των βιομετρικών δεδομένων στο σερβερ για αποθήκευση, πρέπει να επιβεβαιώνεται πως η κλήση έγινε από τον ίδιο το χρήστη που έκανε την εγγραφή. Ο server θα αναγνωρίσει αυτό το token και θα επιτρέψει τη αποθήκευση των βιομετρικών δεδομένων και εν τέλη την επιτυχή ενεργοποίηση του χρήστη. Πριν την αποστολή των βιομετρικών δεδομένων και κατά τη λήψη του Token από τον client, η εφαρμογή παραπέμπει το χρήστη να τραβήξει μια φωτογραφία του προσώπου του. Αυτόματα, γίνεται έλεγχος βασισμένος σε τρεις παραμέτρους.



Εικόνα 7.8: Σελίδα λήψης φωτογραφίας προσώπου του συνδεόμενου χρήστη

Οι τρεις αυτοί παράμετροι αποτελούν, πρώτον, την ανίχνευση προσώπου στη φωτογραφία, δεύτερον, εφόσον γίνει ανίχνευση, να διατηρεί την έκφρασή του ουδέτερη και τρίτον, να βρίσκεται σε θέση όπου η φωτογραφία να μην βγαίνει πολύ σκοτεινή, ώστε να είναι ευδιάκριτα τα χαρακτηριστικά του. Μέχρι να τηρηθούν αυτοί οι τρεις παράμετροι, δεν γίνεται εξαγωγή των βιομετρικών χαρακτηριστικών από την εφαρμογή και ο χρήστης παραπέμπεται συνεχώς στη λήψη φωτογραφίας. Αν και δίνεται η δυνατότητα στο χρήστη για τη λήψη σωστής φωτογραφίας, σε περίπτωση που η αναγνώριση είναι επιτυχής, αλλά έχει παρέλθει ο χρόνος ισχύος του token, δεν θα γίνει αποθήκευση. Θα πρέπει ο χρήστης να επαναλάβει τη διαδικασία από την αρχή. Αντιθέτως, αν τηρηθούν οι προϋποθέσεις που αναφέρθηκαν και γίνει αποστολή των χαρακτηριστικών μέσα στο χρόνο ισχύος του sequence token, θα γίνει επιτυχής αποθήκευση και ενεργοποίηση του χρήστη. Ταυτόχρονα, αυθεντικοποιείται και του δίνεται πρόσβαση στην εφαρμογή.

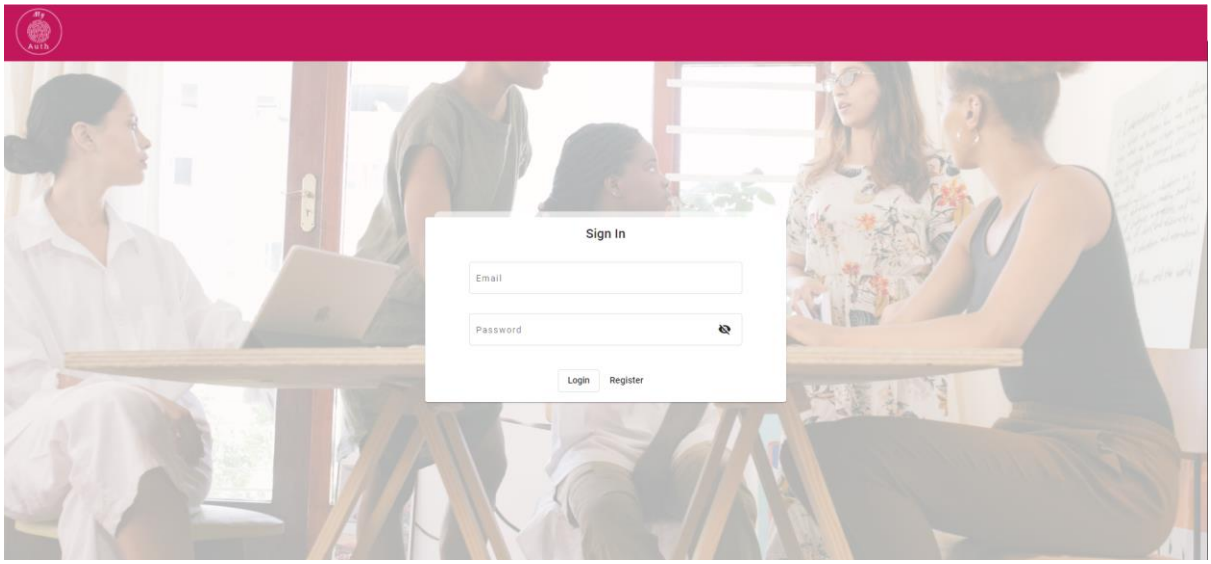
Παρακάτω ακολουθεί ένα διάγραμμα ροής της διαδικασίας εγγραφής του χρήστη που περιεγράφηκε.



Σχήμα 7.2: Σχεδιάγραμμα ροής εγγραφής χρήστη στην υπηρεσία MyAuth

7.4.2 **Sign In**

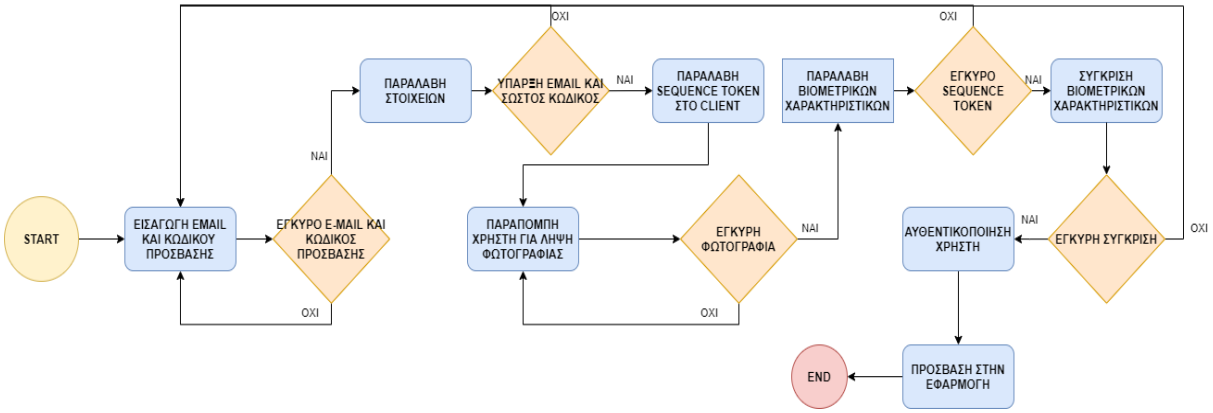
Εκτός από τη διαδικασία εγγραφής καινούργιου χρήστη, προφανώς, υπάρχει και η είσοδος ενός ήδη υπάρχων. Η διαδικασία αυτή ξεκινά από την εισαγωγή email και κωδικού πρόσβασης από τον χρήστη.



Εικόνα 7.9: Σελίδα εισόδου χρήστη στην υπηρεσία MyAuth

Γίνεται έλεγχος στον server για την ορθότητα αυτών των δύο χαρακτηριστικών. Στην περίπτωση επιτυχίας, στέλνεται στο client ένα sequence token για τον ίδιο λόγο που περιγράφηκε στη διαδικασία της εγγραφής χρήστη. Ύστερα ο χρήστης καλείται να βγάλει μια φωτογραφία και ακολουθείται η ίδια διαδικασία με την εγγραφή χρήστη. Η μόνη διαφορά είναι πως γίνεται σύγκριση των βιομετρικών χαρακτηριστικών της φωτογραφίας που μόλις λήφθηκε με αυτών που έχουν αποθηκευτεί κατά την εγγραφή. Εάν και εφόσον η σύγκριση είναι επιτυχής και τα χαρακτηριστικά μοιάζουν, τότε ο χρήστης αυθεντικοποιείται και του δίνεται πρόσβαση στην εφαρμογή.

Παρακάτω παρουσιάζεται το διάγραμμα ροής της εισόδου χρήστη.



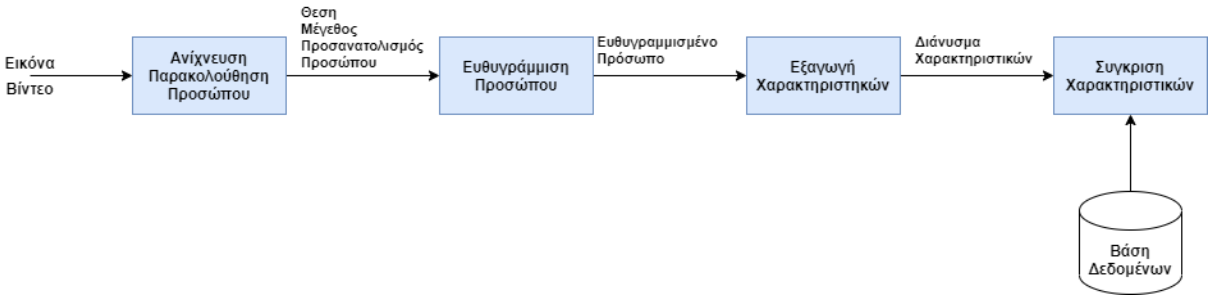
Σχήμα 7.3: Σχεδιάγραμμα ροής εισόδου χρήστη στην υπηρεσία MyAuth

7.5 Εξαγωγή βιομετρικών χαρακτηριστικών

7.5.1 Τεχνολογία που χρησιμοποιήθηκε

Η διεκπεραίωση της διαδικασίας αναγνώρισης, καταγραφής και εξαγωγής βιομετρικών χαρακτηριστικών προσώπου, πραγματοποιήθηκε με τη χρήση της βιβλιοθήκης **face-api.js.** Ένα τυπικό σύστημα αναγνώρισης προσώπου αποτελείται γενικά από τέσσερα υποσυστήματα. Αυτά αποτελούν την ανίχνευση, την ευθυγράμμιση, την εξαγωγή χαρακτηριστικών και την σύγκριση. Το face-api.js, όπως προειπώθηκε, εξυπηρετεί τα τρία πρώτα βήματα.

Παρακάτω ακολουθεί μια αναπαράσταση ενός τέτοιου συστήματος.



Σχήμα 7.4: Σχεδιάγραμμα ροής λειτουργίας βασικού βιομετρικού συστήματος

Όπως παρατηρείται, η παραπάνω αρχιτεκτονική συμπίπτει με αυτή του βιομετρικού συστήματος που επιλέχθηκε για την υλοποίηση της εφαρμογής. Όπως φαίνεται και στο σχήμα 5.2, αυτή αποτελείται από μια είσοδο, την ανίχνευση, την επεξεργασία και την εξαγωγή χαρακτηριστικών προσώπου. Τα τρία πρώτα βήματα γίνονται στον client όπως προστάζει η υλοποίηση αυτή. Η επεξεργασία, συγκεκριμένα σε αυτή την περίπτωση, αποτελεί την εύρεση οροσήμων προσώπου και την ευθυγράμμιση που παρέχει το face-api.js. Το τέταρτο βήμα, που αποτελεί τη σύγκριση, γίνεται στον server.

Πριν γίνει επεξήγηση λειτουργίας αυτής της διαδικασίας, θα γίνει παράθεση μερικών χρήσιμων ορισμών ώστε να είναι κατανοητή η εξήγηση στην πορεία.

7.5.1.1 Χρήσιμοι ορισμοί

Face-Api.js: Είναι ένα javascript module, βασισμένο στο tensorflow-core.js framework, που δίνει τη δυνατότητα χρήσης διαφόρων CNN (Convolutional Neural Networks) για την επίλυση του

προβλήματος της ανίχνευσης και της αναγνώρισης προσώπου, αλλά και αυτού της ανίχνευσης οροσήμων (landmarks) του προσώπου. Το face-api.js παρέχει μια βελτιστοποιημένη εμπειρία χρήσης των λειτουργιών αυτών για εφαρμογές τύπου web ή κινητών συσκευών. [19]

TensorFlow.js: Είναι μια βιβλιοθήκη της Javascript για την εκπαίδευση και την εξαγωγή μοντέλων μηχανικής μάθησης στους Browsers και στο Framework της javascript, Node.js.[20]

Convolutional Neural Network: Ένα Convolutional Neural Network (ConvNet/CNN) είναι ένας Deep Learning αλγόριθμος που μπορεί να λάβει ως είσοδο μια εικόνα, να αποδώσει βαθμό "σημαντικότητας" (δηλαδή weights και biases) σε διάφορες μικρότερες όψεις/πλαίσια της εικόνας και να μπορεί να διαφοροποιήσει το ένα από το άλλο. Η αρχιτεκτονική ενός Convolutional Network είναι ανάλογη με εκείνη του μοτίβου συνδεσιμότητας των νευρώνων του ανθρώπινου εγκεφάλου και εμπνεύστηκε από την οργάνωση του Visual Cortex. Οι μεμονωμένοι νευρώνες ανταποκρίνονται στα ερεθίσματα μόνο σε μια περιορισμένη περιοχή του οπτικού πεδίου, γνωστή ως Δεκτικό Πεδίο. Μια συλλογή τέτοιων πεδίων επικαλύπτουν ολόκληρη την οπτική περιοχή. [22]

Face Detection / Ανίχνευση Προσώπου: Η ανίχνευση προσώπου (face detection) αναφέρεται στον έλεγχο, αν υπάρχουν πρόσωπα σε μία εικόνα, και στον προσδιορισμό της θέσης και των διαστάσεών τους. [21]

Face Recognition / Αναγνώριση Προσώπου: Η αναγνώριση προσώπου είναι ένα πρόβλημα αναγνώρισης προτύπων (pattern recognition). Ένα δοκιμαστικό πρόσωπο (face probe), που βρίσκεται σε μια εικόνα και υπόκειται σε μεταβολές, ανάλογα με τον φωτισμό, τον προσανατολισμό, τους μορφασμούς κλπ., πρέπει να αναγνωρισθεί συγκρινόμενο με δισδιάστατες ή τρισδιάστατες εικόνες μιας συλλογής προσώπων, που είναι διαθέσιμες σε μία βάση δεδομένων ή σε κάποιο άλλο χώρο αποθήκευσης. [21]

Face Landmarks / Ορόσημα Προσώπου: Κατά την ευθυγράμμιση του προσώπου, επιλέγονται από τον αλγόριθμο κάποια χαρακτηριστικά. Αυτά αποτελούν σημεία αναφοράς ή αλλιώς ορόσημα (landmarks) του προσώπου, από συγκεκριμένα σημεία ενδιαφέροντος. Αυτά, συνήθως, μπορεί να είναι χαρακτηριστικά που διαφέρουν από άνθρωπο σε άνθρωπο, όπως τα μάτια, τα φρύδια, η μύτη και το στόμα. [21] Η ευθυγράμμιση του προσώπου είναι απαραίτητη για την καλύτερη εξαγωγή χαρακτηριστικών.

WebGL: Το WebGL είναι ένα JavaScript API που χρησιμοποιείται για αναπαράσταση και απόδοση διαδραστικών 3D και 2D γραφικών σε οποιοδήποτε συμβατό web browser, χωρίς τη χρήση plug-ins. [23]

Έχει ενσωματωθεί πλήρως σε όλα τα web πρότυπα του προγράμματος περιήγησης. Επιτρέπει την επεξεργασία του καμβά της ιστοσελίδας και ότι αφορά τη φυσική και την επεξεργασία εικόνας. Τα στοιχεία που παρέχει το WebGL αναμειγνύονται με τα στοιχεία HTML ή το φόντο της σελίδας. Τα WebGL προγράμματα αποτελούνται από κώδικα γραμμένο σε JavaScript και από κώδικα (shader) που εκτελείται στην μονάδα επεξεργασίας γραφικών του υπολογιστή (GPU).

7.5.1.2 Πως λειτουργεί το Face-api.js

Όπως ειπώθηκε πρωτύτερα, το face-api.js είναι βασισμένο στο tensorflow-core.js και παρέχει τις ίδιες δυνατότητες αναγνώρισης, σύγκρισης και ανίχνευσης προσώπου, αλλά στους browsers. Επιπλέον, το face-api.js παρέχει μοντέλα, τα οποία είναι βελτιστοποιημένα για τον ιστό και για εκτέλεση σε φορητές

συσκευές. Αυτά αποτελούν ήδη εκπαιδευμένα μοντέλα (pre-trained models), γνωστά στη κοινότητα και που τα οποία εξυπηρετούν κάθε λειτουργία που περιλαμβάνει την αναγνώριση και ανίχνευση προσώπου σε μια εικόνα. Τέλος, οι λειτουργίες αυτές επιτυγχάνονται μέσω GPU, χρησιμοποιώντας το WebGL backend.

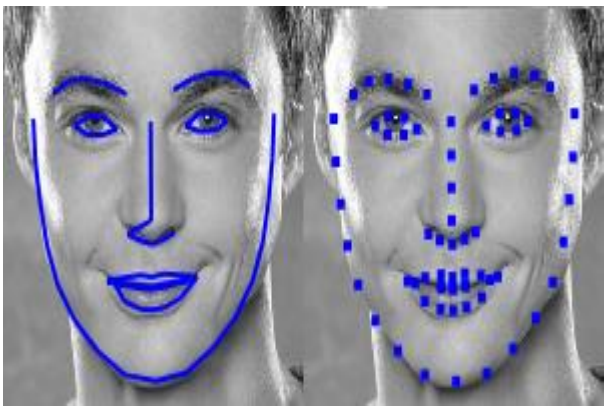
Παίρνοντας λοιπόν ένα-ένα τα βήματα ενός συστήματος αναγνώρισης προσώπου, το face-api δέχεται ως είσοδο μια εικόνα που του παρέχεται. Το πρώτο βήμα είναι να γίνει ανίχνευση του προσώπου στην εικόνα εισόδου. Το Face-api.js μπορεί να χρησιμοποιήσει πολλαπλά pre-trained μοντέλα για ανίχνευση προσώπου για διαφορετικές περιπτώσεις χρήσης. Στην περίπτωση της εργασίας αυτής, χρησιμοποιείται και το πιο ακριβές μοντέλο που παρέχει το api. Αυτό είναι το SSD (Single Shot Multibox Detector), το οποίο είναι βασικά ένα CNN που βασίζεται στο MobileNet V1, με μερικά επιπλέον επίπεδα box prediction στο πάνω μέρος του νευρωνικού δικτύου. Άλλα μοντέλα ανίχνευσης που χρησιμοποιεί το face-api.js είναι το Tiny Face Detector, το οποίο είναι χρήσιμο για αναγνώριση σε πραγματικό χρόνο, αλλά είναι λιγότερο ακριβές από SSD που χρησιμοποιείται στο σύστημά μας. Τέλος παρέχεται και το MTCNN (Multi-task Cascaded Convolutional Neural Network) που χρησιμοποιείται κυρίως για πειραματικούς σκοπούς.

Η ανίχνευση επιστρέφει τα bounding boxes (πλαίσια γύρω από το πρόσωπο) κάθε προσώπου, με τις αντίστοιχες βαθμολογίες τους, π.χ. η πιθανότητα κάθε πλαισίου να δείχνει ένα πρόσωπο. Οι βαθμολογίες χρησιμοποιούνται για το φιλτράρισμα των πλαισίων οριοθέτησης, καθώς μια εικόνα μπορεί να μην περιέχει κάποιο πρόσωπο. Ένα παράδειγμα ανίχνευσης με bounding boxes είναι το παρακάτω:

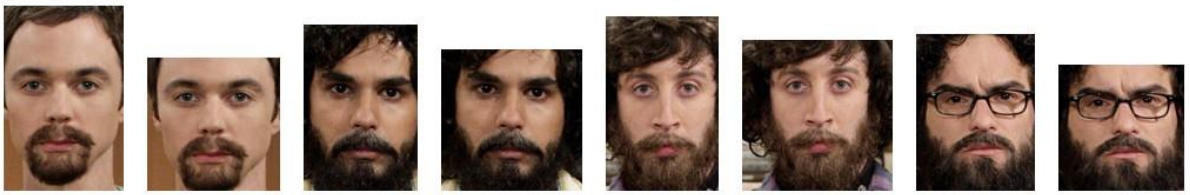
Εικόνα 7.10: Εύρεση bounding boxes μέσω Face-api.js

Στο δεύτερο βήμα, όπου πρέπει να γίνει η επεξεργασία της ανίχνευσης, πραγματοποιείται ή ευθυγράμμιση των bounding boxes, έτσι ώστε να εξαχθούν οι εικόνες κεντραρισμένες στο πρόσωπο που βρέθηκε μέσα σε κάθε πλαίσιο. Η συγκεκριμένη διαδικασία χρειάζεται διότι, χωρίς αυτή, η αναγνώριση και σύγκριση προσώπου γίνεται πιο κοστοβόρα. Επίσης, χρησιμοποιώντας την ευθυγράμμιση, η αναγνώριση προσώπου γίνεται πιο ακριβής.

Η ευθυγράμμιση αυτή επιτυγχάνεται, καθώς το face-api.js υλοποιεί ένα απλό CNN, το οποίο επιστρέφει ορόσημα για 68 σημεία μιας εικόνας προσώπου. Με βάση αυτά τα ορόσημα, ο αλγόριθμος πραγματοποιεί ευθυγράμμιση των bounding boxes. Έτσι, κάθε πρόσωπο υπόκειται σε επεξεργασία και είναι έτοιμο για το επόμενο βήμα. Παρακάτω ακολουθούν κάποια παραδείγματα ευθυγράμμισης. Στα αριστερά κάθε παραδείγματος βρίσκεται η πραγματική φωτογραφία προσώπου, ενώ στα δεξιά η ευθυγραμμισμένη:



Εικόνα 7.11: Εύρεση των face landmarks μέσω Face-pi.js



Εικόνα 7.12: Αποτέλεσμα ευθυγράμμισης προσώπων μέσω του Face-api.js

Το τρίτο και τελευταίο βήμα για το οποίο χρησιμοποιείται το face-api.js, είναι η εξαγωγή χαρακτηριστικών, μέσω των οποίων γίνεται η τελική σύγκριση. Οι ευθυγραμμισμένες εικόνες προορίζονται για το δίκτυο αναγνώρισης προσώπου που υλοποιεί το face-api. Το δίκτυο αυτό είναι βασισμένο σε μια αρχιτεκτονική τύπου **ResNet-34**. Έχει εκπαιδευτεί έτσι ώστε να χαρτογραφεί τα χαρακτηριστικά ενός ανθρώπινου προσώπου σε ένα **Face Descriptor** (ένα διάνυσμα χαρακτηριστικών με 128 τιμές), το οποίο επίσης συχνά αναφέρεται ως **Face Embedding**. Αυτό το διάνυσμα θα χρειαστεί αργότερα για την σύγκριση και αποθήκευση στο back-end server.

7.5.2 Υλοποίηση κώδικα μέσω face-api.js

Η διαδικασία ανίχνευσης, επεξεργασίας και εξαγωγής χαρακτηριστικών γίνεται αρκετά απλή χρησιμοποιώντας το **face-api.js.** Κατά τη διάρκεια προτροπής του χρήστη να τραβήξει τη φωτογραφία του προσώπου του, φορτώνονται κάποια στατικώς αποθηκευμένα στον client, προ-εκπαιδευμένα μοντέλα. Παρακάτω ακολουθεί ένα παράδειγμα κώδικα φόρτωσης αυτών:

```
48    async ngAfterViewInit(){
49      if(this.isLogged == false){
50        await faceapi.nets.ssdMobilenetv1.loadFromUri('/assets/faceModels');
51        await faceapi.nets.faceLandmark68Net.loadFromUri('/assets/faceModels');
52        await faceapi.nets.faceRecognitionNet.loadFromUri('/assets/faceModels');
53        await faceapi.nets.faceExpressionNet.loadFromUri('/assets/faceModels');
54        faceapi.loadTinyFaceDetectorModel('/assets/faceModels').then(() =>{
55          this.startVideo();
56        });
57      }else{
58        this.startVideo();
59      }
60    }
```

Εικόνα 7.13: Παράδειγμα κώδικα για τη φόρτωση μοντέλων αναγνώρισης προσώπου

Τα μοντέλα που φορτώνονται στη προκειμένη περίπτωση είναι τα εξής:

- **SSDMobileNetV1 :** Το οποίο χρησιμοποιείται για την ανίχνευση όπως προαναφέρθηκε
- **FaceLandmark68Net :** Το οποίο βρίσκει τα ορόσημα του προσώπου σε 68 σημεία
- **FaceExpressionNet :** Το οποίο αναγνωρίζει την έκφραση του προσώπου σε ποσοστό.
- **FaceRecognitionNet :** Το οποίο παρέχει την αναγνώριση του προσώπου.

Το face-api.js συνδυάζει αυτά τα μοντέλα, δίνοντας πρόσβαση στον developer να χρησιμοποιήσει ένα higher api, με το οποίο εξάγονται όλα τα απαραίτητα αποτελέσματα. Γίνεται λοιπόν χρήση αυτού του higher api και ο κώδικας μοιάζει κάπως έτσι:

```
captureFace() {

  this.image = document.getElementById('screenshot');

  faceapi.detectSingleFace(this.image).withFaceLandmarks().withFaceExpressions().withFaceDescriptor().run().then(res => {
    this.result = res;
    if((this.result != null && this.result != undefined && this.result.expressions.neutral >= 0.6) ){
      this.serverCall(this.result.descriptor);
    }
    else if((this.result != null && this.result != undefined &&  this.result.expressions.neutral < 0.6)){
      this._snackBar.open("Please keep your expression neutral","OK",{
        duration : 3000,
        panelClass: ['failure-snackbar']
      });
      this.result = null;
      this.capture = null;
      this.startVideo();

    }
    else if(this.result == null || this.result == undefined){
      this._snackBar.open("Couldn't track a face. Please try again","OK",{
        duration : 3000,
        panelClass: ['failure-snackbar']
      });
      this.result = null;
      this.capture = null;
      this.startVideo();
    }
  });
}
```

Εικόνα 7.14: Παράδειγμα κώδικα λειτουργίας του higher api του face-api.js

Όταν ο χρήστης πατάει το κουμπί για λήψη φωτογραφίας καλείται η μέθοδος **captureFace(),** η οποία αρχικά παίρνει τη φωτογραφία του χρήστη και την εισάγει σαν παράμετρο στο face-api αντικείμενο που έχει εισαχθεί στα dependencies. Από την εικόνα αυτή γίνεται με τη σειρά ανίχνευση του προσώπου, εξαγωγή των οροσήμων, εξακρίβωση της έκφρασης και τέλος η εξαγωγή του διανύσματος ή αλλιώς του **face embedding.** Χρησιμοποιείται δηλαδή αυτό το κομμάτι κώδικα:

```
faceapi.detectSingleFace(this.image).withFaceLandmarks().withFaceExpres
sions().withFaceDescriptor().run();
```

Εικόνα 7.15: Παράδειγμα κώδικα χρήσης του αντικειμένου faceapi

Αφού γίνει εξαγωγή των στοιχείων αυτών ο αλγόριθμος ελέγχει αν έχει βρεθεί κάποιο πρόσωπο στη φωτογραφία. Αν όχι, προτρέπει το χρήστη για δεύτερη λήψη. Εφόσον βρεθεί όμως κάποιο πρόσωπο, ελέγχεται αν ή έκφρασή του είναι ουδέτερη. Αυτό επιτυγχάνεται, με το αν το ποσοστό ουδετερότητας

που προκύπτει από το μοντέλο faceExpressionNet, ξεπερνά το 60% . Το threshold αυτό προέκυψε καθώς μετά από δοκιμές το 70% αποτελούσε ο μέσος όρος ποσοστού ουδετερότητας που προκύπτει από λήψη σχετικά καλών, δηλαδή μη σκοτεινών ή κουνημένων, φωτογραφιών. Επιλέχθηκε το 60% ώστε να παρέχεται στο χρήστη ένα ποσοστό χαλαρότητας, καθώς ενδέχεται το face-api να μπερδεύεται και να μην κρίνει το βαθμό ουδετερότητας πάντα αντικειμενικά. Αυτό προκύπτει εξαιτίας κάποιων παραμέτρων, όπως είναι ο κακός φωτισμός της φωτογραφίας ή η απόσταση του προσώπου από τη κάμερα.

Εφόσον λοιπόν, η έκφραση του προσώπου της φωτογραφίας είναι ουδέτερη, ο αλγόριθμος εξάγει το διάνυσμα του προσώπου από το face-api αντικείμενο που έχει εισαχθεί. Το διάνυσμα αυτό, μαζί με το sequence token που αναφέρθηκε πρωτύτερα, μετατρέπονται σε ένα json object που θα αποτελεί το body ενός http αιτήματος, που θα πραγματοποιηθεί στο backend server για να γίνει η τελική σύγκριση.

```
133   serverCall(descriptor : Float32Array){
134
135     let existingUserInfo = this.sessionService.UserInfo;
136
137     let facialRecoUser = new LoginFacialReqModel()
138     facialRecoUser.x_seq = existingUserInfo.x_seq;
139     facialRecoUser.faceDescriptor = JSON.stringify(Array.from(descriptor));
140
141     this.authService.FacialAuthentication(facialRecoUser)
142     .pipe(first())
143     .subscribe({
144         next: () => {
145           this.alertService.successAlert('Login successful');
146           this.result = null;
147           this.capture = null;
148           this.router.navigate(['/account']);
149         },
150         error: error => {
151             console.log(error);
152             this.result = null;
153             this.capture = null;
154             this.router.navigate(['/']);
155           }
156     });
157   }
```

Εικόνα 7.16: Παράδειγμα κώδικα αποστολής του face embedding στο server

Περισσότερα για τη σύγκριση και την υλοποίηση του backend server κώδικα, θα αναφερθεί στην ενότητα 7.5.3.3 παρακάτω.

7.5.3 Υλοποίηση κώδικα στον backend server

Ο back-end server αποτελεί το τελευταίο κομμάτι της υλοποίησης της βιομετρικής αυθεντικοποίησης. Καθώς το face-api πραγματοποιεί τα πρώτα βήματα του βιομετρικού συστήματος που επιλέχθηκε, ο αλγόριθμος που θα περιγραφεί παρακάτω, πραγματοποιεί τη σύγκριση των διανυσμάτων ή **αλλιώς face-embendings** και φροντίζει για την αποθήκευση στο server, όπως ορίζει η υλοποίηση βιομετρικού συστήματος σύγκριση στο server και αποθήκευση στο server.

7.5.3.1 Επαλήθευση Sequence Token

Το πρώτο βήμα αποτελεί την εισαγωγή email και κωδικού από το χρήστη. Μετά τη επαλήθευση αυτή, ο server περιμένει ένα αίτημα από τον client το οποίο θα εμπεριέχει το sequence token και ένα διάνυσμα το οποίο θα αποτελεί το face embedding του προσώπου του χρήστη, που μόλις έγινε λήψη φωτογραφίας. Αρχικά, ελέγχεται αν το sequence token του αιτήματος είναι έγκυρο. Κατά τη είσοδο του χρήστη με κωδικό πρόσβασης, αποθηκεύεται στη cache μνήμη του Server μία hash τιμή, η οποία έχει αναγνωριστικό το ID του χρήστη. Αυτό αποτελεί ένα **Guid** ή αλλιώς Globally Unique Identifier. Αυτό το hash, δηλαδή το sequence token που αποθηκεύεται, ουσιαστικά αποτελεί έναν συνδυασμό τιμών οι οποίες μετατρέπονται σε ένα Base64 string. Οι τιμές αυτές αποτελούν το email, το Id και την ώρα την οποία γίνεται η αποθήκευση του token. Αυτές οι τρεις τιμές διαχωρίζονται με **“ :: “** η κάθε μία. Το τελικό token είναι έγκυρο για ένα λεπτό πριν διαγραφεί από τη προσωρινή μνήμη και δεν είναι πλέον διαθέσιμο. Ο σχετικός κώδικας για τη παραπάνω διαδικασία αποθήκευσης του token είναι ο εξής:

```
21      public string CombineAndSaveHash(string value, Guid? guid = null)
22      {
23          Guid TheGuid;
24          if (!guid.HasValue || (guid.HasValue && guid.Value == Guid.Empty))
25          {
26              TheGuid = Guid.NewGuid();
27          }
28          else
29          {
30              TheGuid = guid.Value;
31          }
32          var finalStrHash = $"{TheGuid.ToString()}::{value}::{DateTime.Now}";
33
34          var valRes = ToBase64Str(finalStrHash);
35
36          return _cache.StoreInMemoryAbsoluteCustomKey<string>(TheGuid.ToString(), valRes, 60 * 60);
37      }
```

Εικόνα 7.17: Παράδειγμα κώδικα αποθήκευσης sequence_token

Κατά τη παραλαβή του sequence token λοιπόν από τον client, ο server side αλγόριθμος ακολουθεί μια συγκεκριμένη διαδικασία για να σιγουρέψει ότι το αίτημα για σύγκριση βιομετρικών χαρακτηριστικών προέρχεται από τον ίδιο client που πραγματοποίησε είσοδο πριν με το κωδικό πρόσβασης.

Αρχικά ο αλγόριθμος σιγουρεύει πως το hash που έρχεται από το client είναι έγκυρο. Αυτό γίνεται ως εξής. Κάνει αποκωδικοποίηση του sequence token από Base64 string σε ένα string. Θεωρητικά το sequence token, θα πρέπει να αποτελεί ένα Base64 string το οποίο αν το αποκωδικοποιήσεις θα περιέχει τρεις τιμές. Δηλαδή αυτές που περιγράφηκαν και νωρίτερα, το email, το Id και μια ημερομηνία. Έτσι αυτό το hash, αν δεν έχει τροποποιηθεί ή κατασκευαστεί από κάποιο κακόβουλο χρήστη, θα πρέπει, κατά το decode, να επιστρέψει ένα άλλο string που θα είναι της μορφής:

“{{ UserId }}::{{ UserEmail }}::{{ CreationDate }}”

Ειδάλλως η αποκωδικοποίηση δε θα παράγει ένα τέτοιο string και ο αλγόριθμος θα οδηγηθεί σε σφάλμα, με αποτέλεσμα να μην εγκυροποιηθεί το sequence token και ο κώδικας να μη προχωρήσει ποτέ στη σύγκριση των face embendings. Άπαξ και το token είναι σωστό, εξάγονται αυτές οι τρεις τιμές, και ελέγχεται αν υπάρχει στη μνήμη cache κάποια τιμή αποθηκευμένη με το Id του χρήστη. Αν δεν έρθει κάποιο αποτέλεσμα καταλήγουμε σε δύο συμπεράσματα:

- Ότι ο χρήστης έχει ταυτοποιηθεί με κωδικό πρόσβασης, αλλά ο χρόνος εγκυρότητας του token έχει παρέλθει
- Ότι το Id του χρήστη δεν υπάρχει και πως κάποιος κακόβουλος χρήστης έχει προσπεράσει το πρώτο βήμα επαλήθευσης με συνθηματικό.

Σε περίπτωση όμως που βρεθεί η τιμή, εξάγονται οι τιμές του αποθηκευμένου στη cache μνήμη hash και ελέγχεται στη βάση αν υπάρχει κάποιος χρήστης με το ίδιο ID και email. Εφόσον και το τελευταίο αυτό βήμα είναι έγκυρο, ο αλγόριθμος προχωράει στη σύγκριση των διανυσμάτων. Ο σχετικός κώδικας είναι ο εξής:

```
62      public bool RetrieveValidateDiscardHash(string hashStr, bool discard = false)
63      {
64          if (hashStr == null)
65          {
66              goto FailureCase;
67          }
68          try
69          {
70              var main = Base64StringDecode(hashStr)?.Split("::").ToList() ?? null;
71
72              var res = _cache.GetFromMemoryCustomKey<string>(main.First());
73
74              if (res == null)
75              {
76                  return false;
77              }
78
79              if (res != null && discard)
80              {
81                  _cache.RemoveFromMemoryCustomKey(main.First());
82              }
83              return res == hashStr;
84          }
85          catch (Exception)
86          {
87              goto FailureCase;
88          }
89
90
91      FailureCase: return false;
92      }
```

Εικόνα 7.18: Παράδειγμα κώδικα επαλήθευσης sequence_token

7.5.3.2 Αποθήκευση Face Embendings

Πριν προχωρήσουμε στη σύγκριση των face embendings κατά την είσοδο του χρήστη στην υπηρεσία, θα πρέπει να ξεκαθαριστεί ο τρόπος με τον οποίο γίνεται η αποθήκευση αυτών κατά την εγγραφή του. Η μέχρι στιγμής διαδικασία αποθήκευσης και επαλήθευσης sequence token αποτελεί την ίδια και για την είσοδο και για την εγγραφή του χρήστη. Η συνέχεια όμως, διαφέρει σε κάθε περίπτωση. Κατά την εγγραφή λοιπόν, εφόσον πραγματοποιηθούν τα παραπάνω βήματα, γίνεται αποθήκευση του διανύσματος που ήρθε απο το client. Η τιμή αυτή όμως, που αποτελεί ένα float Array, δεν αποθηκεύεται κατευθείαν ως ένα πεδίο της εγγραφής του χρήστη στη βάση δεδομένων. Αντιθέτως στο πεδίο αυτό αποθηκεύεται η τοποθεσία ενός αρχείου που περιέχει κρυπτογραφημένο το διάνυσμα χαρακτηριστικών του χρήστη. Η μέθοδος αυτή ακολουθείται για τη περίπτωση διαρροής των δεδομένων της βάσης σε κάποιο κακόβουλο χρήστη. Με αυτό το τρόπο δεν θα μπορεί να έχει πρόσβαση στη τιμή του διανύσματος βιομετρικών χαρακτηριστικών του χρήστη, παρά μόνο στη τοποθεσία του αρχείου στο

σύστημα αρχείων του server. Επίσης υπάρχει και η σπάνια περίπτωση που ένας κακόβουλος χρήστης καταφέρει και αποκτήσει πρόσβαση στο server. Αυτό μπορεί να συμβεί όταν κάποιος επιχειρεί και καταφέρει τη σύνδεση μέσω SSH στην διεύθυνση IP του. Αυτό όμως προϋποθέτει να υπάρχει στη κατοχή του hacker ένα private key για τη σύνδεση, το οποίο συνήθως δίνεται από το διαχειριστή του κάθε συστήματος. Επιπλέον, χρειάζεται επαλήθευση κατά την SSH σύνδεση με κάποιο συνθηματικό που έχει οριστεί από το ίδιο το διαχειριστή. Στη περίπτωση όμως, που κατορθώσει και αποκτήσει πρόσβαση στο server, θα έχει πρόσβαση στο αρχείο που συγκρατεί την ευαίσθητη πληροφορία αλλά και πάλι θα έχει ως εμπόδιο την αποκρυπτογράφηση της τιμής που είναι αποθηκευμένη σε αυτό.

Η κρυπτογράφηση του διανύσματος ακολουθεί τα εξής βήματα. Αρχικά χρησιμοποιείται κάθε φορά ένα encryption key, το οποίο λαμβάνεται από τα configurations του server. Κάθε κλειδί, λαμβάνεται από μια συλλογή κλειδιών ώστε να μην επιλέγεται το ίδιο για κάθε κρυπτογράφηση. Αυτό αποτελείται από ένα σύνολο αλφαριθμητικών χαρακτήρων μεγάλου μήκους, ογδόντα-τεσσάρων στη προκειμένη περίπτωση, για την αύξηση της δυσκολίας εύρεσής του σε τυχόν απόπειρα αποκρυπτογράφησης κάποιας τιμής.

Για τη κρυπτογράφηση χρησιμοποιείται η βιβλιοθήκη **NIMBUS JOSE-JWT.** Πιο συγκεκριμένα γίνεται χρήση του αλγορίθμου κρυπτογράφησης **PBES2** με **HMAC SHA-256** και **AES** κρυπτογράφηση κλειδιού με 128 bit. Ο αλγόριθμος κρυπτογράφησης **PBES2,** είναι μια επέκταση του **PKCS,** ή αλλιώς **Password-Based Cryptography Specification**. Ο **PBES2** συνδυάζει μια συνάρτηση εξαγωγής κλειδιού βασισμένη σε κάποιο **encryption key**, με ένα υποκείμενο **encryption scheme**. Το κλειδί, το μήκος του και άλλες παράμετροι για το υποκείμενο σχήμα της κρυπτογράφησης, εξαρτώνται ανάλογα τη διαφορετική περίπτωση. Σύμφωνα με το έγγραφο RFC 8018 v.2.1 απο **Internet Engineering Task Force (IETF),** με τίτλο **PKCS #5: Password-Based Cryptography Specification** [24] **,** ο αλγόριθμος PBES2 είναι ο κατάλληλος για κρυπτογράφηση σε νέες εφαρμογές [24].

Λαμβάνοντας υπόψη τα παραπάνω, τα face-embeddings κάθε χρήστη παρέχονται μια ισχυρή κρυπτογράφηση, η οποία κρατάει τα προσωπικά τους δεδομένα ασφαλή. Στο file system του server υπάρχει ένας φάκελος, ο οποίος περιέχει όλα τα αρχεία με κρυπτογραφημένες τιμές. Κάθε ένα από αυτά, ονομάζεται σύμφωνα με το UUID ή αλλιώς μοναδικό ID του εγγεγραμμένου χρήστη.

Εν κατακλείδι, κατά την εγγραφή ενός χρήστη, επιλέγεται το float array, μετατρέπεται σε ενα string αντικείμενο και αυτό δέχεται κρυπτογράφηση με το encryption key που έχει επιλεχθεί στη συγκεκριμένη διεργασία. Αμέσως, ακολουθεί μια διαδικασία δημιουργίας ενός αρχείου μορφής **TXT**, με όνομα αρχείου ίδιο με το ID του χρήστη. Στο αρχείο αυτό, εισάγεται στη πρώτη γραμμή το encrypted string του αποτελέσματος της κρυπτογράφησης. Ύστερα το absolute path του αρχείου, το Url, το όνομα και το file extension αποθηκεύονται ως ένα jsonb αντικείμενο, ως ένα πεδίο στην καινούργια εγγραφή του χρήστη στη βάση δεδομένων.

Το jsonb πεδίο αυτό είναι της μορφής που φαίνεται παρακάτω:


```
{
  "Ext": ".txt",
  "Url": "/FaceDescriptors/d2442012-be17-4035-9f8c-fc3220e55019.txt",
  "Path": "C:\\Users\\source\\repos\\MyAuth\\MyAuth\\wwwroot\\FaceDescriptors",
  "FileName": "d2442012-be17-4035-9f8c-fc3220e55019.txt"
```

}

Επίσης, το απόσπασμα του κώδικα που πραγματοποιεί τη διαδικασία της αποθήκευσης είναι το εξής:

```
MyAuthUser UserExists = await _context.MyAuthUsers.Where(d => d.Email == newUser.Email).FirstOrDefaultAsync();
if (UserExists == null)
{
    string encryptedFaceDescriptor = _encrypterDecrypter.EncryptString(newUser.FaceDescriptor);
    Guid newUserId = Guid.NewGuid();

    var txtFile = await _txtService.WriteToNewTxt(encryptedFaceDescriptor, newUserId, "FaceDescriptors");

    if (txtFile.Status == false)
    {
        return new HttpResponseData<SuccessfulLoginRespModel, ClientsApiErrorCodes>(ClientsApiErrorCodes.InternalError);
    }

    var newUserRecord = new MyAuthUser()
    {
        Id = newUserId,
        Name = newUser.Name,
        Surname = newUser.Surname,
        Email = newUser.Email,
        Created = DateTime.Now,
        HasFaceRegistered = true,
        FaceDescriptor = txtFile.Data,
        Password = _encrypterDecrypter.MD5Hash(newUser.Password)
    };

    _context.MyAuthUsers.Add(newUserRecord);
    await _context.SaveChangesAsync();
    _logger.LogInformation("User created a new account with password.");

    return await DoActualUserlogin(newUserRecord);
}
return new HttpResponseData<SuccessfulLoginRespModel, ClientsApiErrorCodes>(ClientsApiErrorCodes.AlreadyExistingUser);
```

Εικόνα 7.19: Παράδειγμα κώδικα αποθήκευσης face embedding

7.5.3.3 Σύγκριση Face Embendings

Όπως αναφέρθηκε και πριν, η σύγκριση των face embendings αποτελεί το τελευταίο βήμα επιτυχούς επαλήθευσης του χρήστη. Η σύγκριση πραγματοποιείται κατά την είσοδο και ακολουθεί αρχικά τα βήματα γνησιότητας του sequence token. Εφόσον όλα πάνε σωστά, γίνεται ανάκληση του αρχείου που είναι εγγεγραμμένο το αποθηκευμένο face embedding του χρήστη. Ύστερα, γίνεται αποκρυπτογράφηση της τιμής του με τον ίδιο αλγόριθμο και το ίδιο encryption key που έγινε η κρυπτογράφηση κατά την εγγραφή. Το αποτέλεσμα, προφανώς, αποτελεί ένα float array. Μένει μόνο η σύγκριση αυτών των δύο float arrays ή αλλιώς vectors. Με τη λέξη σύγκριση εννοείται η εξακρίβωση της απόστασης αυτών των δυο διανυσμάτων.

Η απόσταση ή αλλιώς distance παίζει σημαντικό ρόλο στη μηχανική μάθηση. Αποτελεί το θεμέλιο για πολλούς δημοφιλείς και αποτελεσματικούς αλγόριθμους μηχανικής μάθησης. Την απόσταση αυτή την ορίζει συνήθως ένας classifier στα μοντέλα μηχανικής μάθησης. Ο classifier ή αλλιώς, ο ταξινομητής, ορίζει την ομοιότητα των τιμών του μοντέλου μηχανικής μάθησης και πραγματοποιεί τον υπολογισμό στο τελευταίο βήμα ταξινόμησης και απόφασης. Για παράδειγμα, ένα μοντέλο τίθεται να εξακριβώσει αν μια εικόνα που απεικονίζει μία γάτα, είναι μια εικόνα που απεικονίζει έναν σκύλο. Ο ταξινομητής θα έπαιρνε την απόφαση, υπολογίζοντας την απόσταση των δύο διανυσμάτων των εικόνων. Αν αυτή η απόσταση ξεπερνούσε ένα threshold που έχει οριστεί, τότε η απόφαση αυτή θα ήταν θετική. Το threshold αυτό ορίζεται πάντα από τον δημιουργό του μοντέλου.

Η απόσταση δύο διανυσμάτων μπορεί να εξακριβωθεί πλέον με πολλές μεθόδους. Για παράδειγμα, με ευκλείδεια, cosine ή μπαεσιανή απόσταση. Η Ευκλείδεια απόσταση αποτελεί ένα ικανοποιητικό μέτρο σύγκρισης και εξακρίβωσης της ομοιότητας των δύο διανυσμάτων. Προφανώς, υπάρχουν καλύτεροι και πιο ειδικοί τρόποι σύγκρισης. Ωστόσο, η μέθοδος που ακολουθήθηκε με την ευκλείδεια απόσταση,

επιφέρει σίγουρα αποτελέσματα με εύκολο και αποτελεσματικό τρόπο, καθώς έχει ακολουθηθεί σε πολλά μοντέλα μηχανικής μάθησης και deep learning, σχετιζόμενα με ταξινόμηση φωτογραφιών.

Η ευκλείδεια απόσταση των δύο διανυσμάτων είναι μικρότερη, όσο περισσότερο αυτά μοιάζουν μεταξύ τους. Δηλαδή, όσο πιο όμοιες είναι οι φωτογραφίες, δηλαδή τα εικονιζόμενα πρόσωπα, τόσο πιο κοντά στο μηδέν θα βρίσκεται η τιμή της απόστασης. Αυτό μπορεί να πάρει τιμή από 0.0 έως 1. Το default κατά κανόνα threshold μιας απόφασης είναι το 0.5 ή 50%. Στη περίπτωση όμως αυτής της σύγκρισης, επειδή εμπλέκονται ευαίσθητα δεδομένα, χρειάζεται ένας πιο λεπτός χειρισμός της τελικής απόφασης της σύγκρισης.

Τελικά, επιλέχθηκε το threshold του αριθμού 0.4. Μετά από μια σειρά προσπαθειών, φαίνεται πως οι τιμές υπολογισμού της ευκλείδειας απόστασης με ένα σωστό πρόσωπο, κυμαίνονταν από το 3.2 έως το 3.9. Απεναντίας, με ένα λανθασμένο πρόσωπο οι τιμές κυμαίνονται από 5.5 και άνω.

Εδώ να σημειωθεί πως αυτό το threshold χαλαρότητας θα μπορεί να είναι παραμετροποιήσιμο και να ρυθμίζεται από τα configuration files της εφαρμογής. Οπότε λοιπόν, κάποιος οργανισμός ή κάποιος developer, που θέλει να το αυξομειώσει, δεδομένου του συστήματος που κατέχει και της κρισιμότητας των δεδομένων που φιλοξενεί, θα μπορεί να επιλέξει τον επιθυμητό, γι' αυτόν, βαθμό χαλαρότητας επιτυχούς επαλήθευσης προσώπου.

Συμφωνά με τα παραπάνω, αν η ευκλείδεια απόσταση μεταξύ των δύο συγκρινόμενων face embedding είναι κάτω από 0.4, τότε ο χρήστης επαληθεύεται και του παρέχεται πρόσβαση στην εφαρμογή.

Ακολουθεί σχετικό τμήμα κώδικα αυτής της σύγκρισης:

```
132             string encryptedFaceDescriptor;
133             using (StreamReader streamReader = new StreamReader(Path.Combine(existingUser.FaceDescriptor.Path, existingUser.FaceDescriptor.FileName)))
134             {
135                 encryptedFaceDescriptor = streamReader.ReadLine();
136             }
137
138             string decryptedFaceDescriptor = _encrypterDecrypter.DecryptString(encryptedFaceDescriptor);
139
140             float[] actualFaceDescriptor = JsonConvert.DeserializeObject<float[]>(decryptedFaceDescriptor);
141             float[] comparingFaceDescriptor = JsonConvert.DeserializeObject<float[]>(input.FaceDescriptor);
142
143
144             var P1 = np.array(actualFaceDescriptor);
145             var P2 = np.array(comparingFaceDescriptor);
146             float ex = (float) np.linalg.norm(P2 - P1);
147
148             if (ex >= 0.4)
149             {
150                 return new HttpResponseData<SuccessfulLoginRespModel, ClientsApiErrorCodes>(ClientsApiErrorCodes.BiometricAuthenticationFailure);
151             }
152
153             return await DoActualUserlogin(existingUser);
```

Εικόνα 7.20: Παράδειγμα κώδικα σύγκρισης face embeddings

7.6 Διαλειτουργική αυθεντικοποίηση με OAuth 2.0

Η παρούσα εφαρμογή, ώστε να υποστηρίζει τις λειτουργίες και να θεωρείται ένα BaaS, προσφέρει διαλειτουργική αυθεντικοποίηση με το πρωτόκολλο εξουσιοδότησης OAuth 2.0. Σύμφωνα με την απεικόνιση της ροής OAuth με authorization code flow, που περιεγράφηκε στο κεφάλαιο 3, η πιλοτική εφαρμογή αποτελεί τον συνδυασμό authorization server και resource server σε μια μονή υλοποίηση backend service. Για ολόκληρη την υλοποίηση της ροής OAuth, χρειάζεται ένας client.

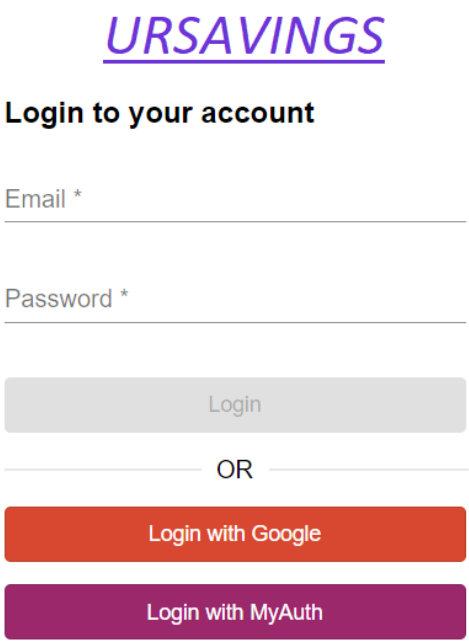
Ο client λοιπόν, αναπαριστάται με μια δεύτερη εφαρμογή, η οποία, σε πραγματικές συνθήκες, θα χρησιμοποιούσε την BaaS υπηρεσία MyAuth για να παρέχει στους πελάτες έναν τρόπο αυθεντικοποίησης με τα βιομετρικά χαρακτηριστικά των προσώπων τους. Η εφαρμογή-παράδειγμα που χρησιμοποιείται, αποτελεί ένα web application εισαγωγής και διαχείρισης εσόδων και εξόδων,

ονόματι URSAVINGS. Οι πελάτες της, που είναι εγγεγραμμένοι και τη χρησιμοποιούν, θα μπορούν να κάνουν είσοδο σε αυτή μέσω της υπηρεσίας MyAuth και να έχουν πρόσβαση στις λειτουργίες που παρέχει.

Στις επόμενες ενότητες, θα γίνει επεξήγηση και αναπαράσταση εικόνων σχετικά με τη διαδικασία διαλειτουργικής αυθεντικοποίησης μεταξύ του budget application και της υπηρεσίας MyAuth.

7.6.1 Είσοδος στην εφαρμογή client

Αρχικά ένας χρήστης της εφαρμογής URSAVINGS, θέλει να κάνει είσοδο σε αυτή, έτσι ώστε να διαχειριστεί και να εισάγει τη τελευταία του συναλλαγή αυτό το μήνα. Η πρώτη οθόνη που βλέπει είναι η εξής:



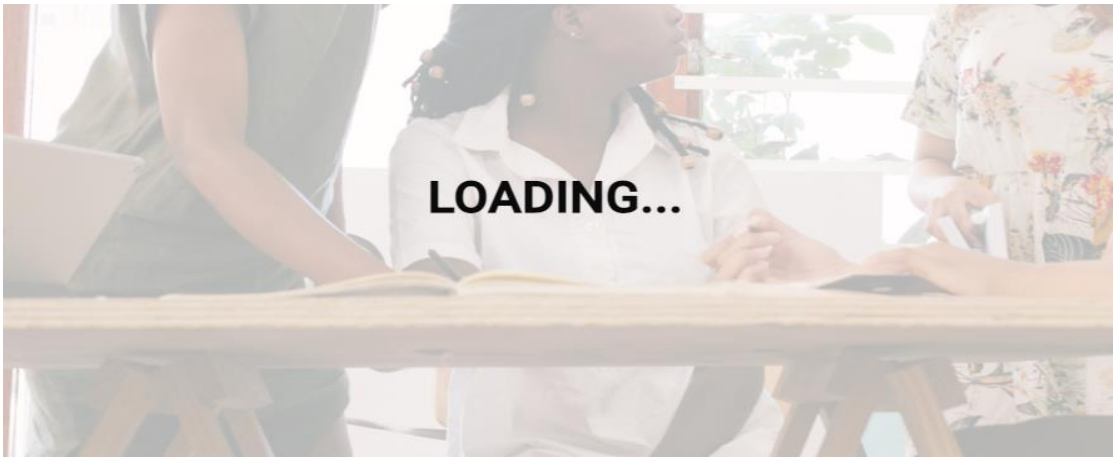
Εικόνα 7.21: Εικόνα σελίδας εισόδου εξωτερικής εφαρμογής

Όπως παρατηρείται, ο χρήστης έχει τη δυνατότητα σύνδεσης χρησιμοποιώντας το email του και κάποιο συνθηματικό, να συνδεθεί μέσω Google ή μέσω της υπηρεσίας MyAuth. Σε αυτήν την περίπτωση, οι δύο τελευταίοι τρόποι σύνδεσης, ακολουθούν ροή OAuth με authorization code.

Η εφαρμογή URSAVINGS δίνει τη δυνατότητα χρήσης ενός λογαριασμού για πολλούς χρήστες. Για παράδειγμα, ένας λογαριασμός μπορεί να αντιστοιχεί σε μια οικογένεια, όπου θα συνδέονται όλα τα μέλη και θα καταγράφουν τα οικογενειακά έξοδα, ανάλογα με τις αγορές του κάθε μέλους. Μια τέτοια υλοποίηση προϋποθέτει την ύπαρξη ρόλων για τη διαχείριση των λογαριασμών. Πιο συγκεκριμένα, ένα ή περισσότερα μέλη θα πρέπει να είναι οι διαχειριστές του λογαριασμού, ώστε να μπορούν να διαχειρίζονται τον λογαριασμό, την ενεργοποίησή του, την έγκριση ή απόρριψη εγγραφών εξόδων, πρόσκληση άλλων μελών κλπ. Έτσι, πέρα από τον ρόλο του διαχειριστή, υπάρχει και ο ρόλος του Editor, που δεν έχει πρόσβαση στη διαχείριση του λογαριασμού.

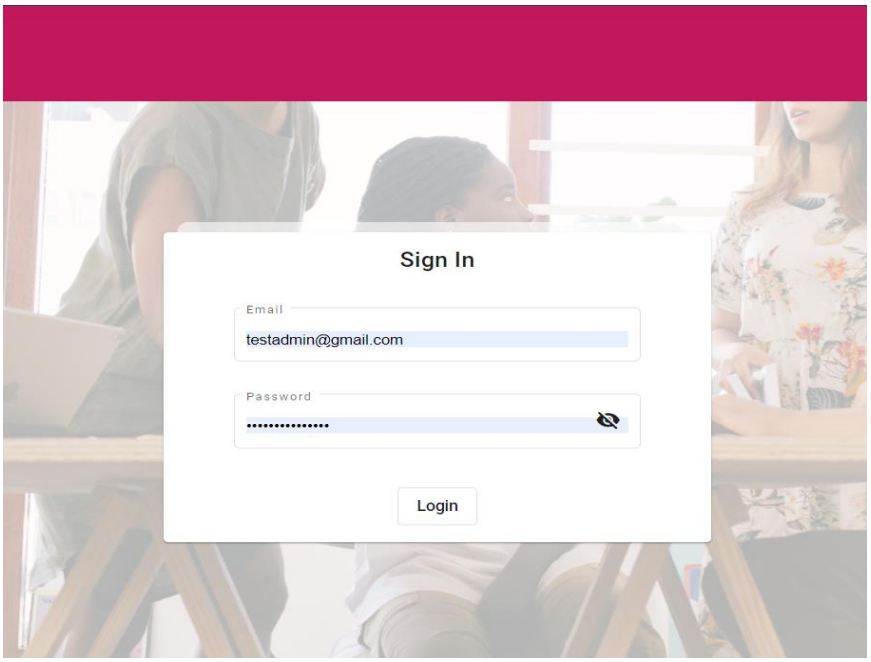
Μπορεί να υπάρχει η περίπτωση να χειρίζεται την εφαρμογή μια εταιρία. Αυτό θα προϋπέθετε πως μόνο έμπιστα άτομα που ανήκουν στην εταιρία θα συνδεόταν στην εφαρμογή, καθώς κανένας άλλος, μη εγκεκριμένος χρήστης, δεν ήταν ασφαλές να εξουσιοδοτηθεί, ώστε να δει τα έξοδα και τα έσοδα της εταιρείας. Μια πολύ καλή λύση θα ήταν, η σύνδεση των μελών να γίνεται μέσω της υπηρεσίας MyAuth που θα επέτρεπε μόνο τα άτομα που τα βιομετρικά του προσώπου τους θα περνούσαν τον έλεγχο.

Έστω πως η εταιρεία αυτή έχει υλοποιήσει μόνο αυτή τη μέθοδο σύνδεσης. Ένας χρήστης του λογαριασμού της εταιρείας πατάει το κουμπί **“Login with MyAuth”** και ακολουθώντας τη ροή OAuth 2.0 ανακατευθύνεται στην εξής σελίδα:



Εικόνα 7.22: Ανακατεύθυνση στην υπηρεσία MyAuth για είσοδο

Αυτή αποτελεί τη σελίδα σύνδεσης στην υπηρεσία MyAuth. Αρχικά φορτώνεται η σελίδα και τα μοντέλα μηχανικής μάθησης. Όταν ολοκληρωθεί η φόρτωση, ο χρήστης που αποτελεί το user-agent μπορεί να κάνει σύνδεση καθώς του εμφανίζεται η εξής σελίδα:



Εικόνα 7.23: Εισαγωγή στοιχείων για επαλήθευση μέσω MyAuth

Σε αυτό το σημείο πρέπει να αναφερθεί πως ο χρήστης έχει ανακατευθυνθεί σε διαφορετικό url από αυτό που περιγράφικε στην ενότητα 6.4.2 . Το url αυτό είναι το εξής:

http://localhost:4200/accounts/external-auth/oauth/verify-client?response_type=code&client_id=Client_ID&redirect_uri=http:%2F%2Flocalhost:8080%2Fapi%2Fauth%2Fexternal%2FmyAuth%2Fcallback%2Flogin&scope=profile&state=my99WOQauNknCws5gjOoB1330qJtRsv3

Αυτό το url ανήκει στη ροή αυθεντικοποίησης OAuth. Παρατηρείται πως το url περιέχει κάποια query values. Αυτά είναι τα:

- **response_type** : Με το οποίο δηλώνεται στον server ότι επιδιώκεται σύνδεση OAuth με τύπο authorization code grant
- **client_id:** Όπου αποτελεί το μοναδικό αναγνωριστικό της εφαρμογής URSAVINGS στη βάση δεδομένων της MyAuth
- **redirect_url:** Όπου είναι το url στο οποίο θα ανακατευθυνθεί ο χρήστης, όταν λάβει το authorization code
- **scope :** Όπου αποτελεί το περιεχόμενο της πληροφορίας που επιθυμεί ο client να λάβει από τη MyAuth, όταν εξουσιοδοτηθεί από τον χρήστη. Σε αυτήν την περίπτωση, υπάρχει μόνο το profile, δηλαδή οι πληροφορίες του προφίλ του χρήστη καθώς αυτές μόνο θα χρειαστούν για την αυθεντικοποίηση.
- **state :** Αποτελεί ένα μοναδικό αναγνωριστικό της κατάστασης της εφαρμογής του client κατά τη διάρκεια χρήσης της από κάποιον χρήστη. Αυτό το **state** θα αποθηκευτεί και θα σταλεί ύστερα στο redirect_uri για επιβεβαίωση από τον client.

Εφόσον ο χρήστης έχει ανακατευθυνθεί στο παραπάνω url, σημαίνει πως έχει γίνει ήδη επιβεβαίωση στον server για τα πεδία αυτά. Δηλαδή, θα πρέπει κάθε query value να περάσει κάποιον έλεγχο. Το response_type θα πρέπει να έχει την τιμή “code” και το scope να είναι “profile”, καθώς μόνο αυτές οι τιμές υποστηρίζονται προς το παρόν. Δεύτερον, το client_id και το redirect_id, πρέπει να είναι ίδια με αυτά που έχει δηλώσει ο χρήστης ή κάτοχος του λογαριασμού στο προφίλ του. Το state δεν περνάει κάποιον έλεγχο.

Ο χρήστης, πλέον, μπορεί να εισάγει το email και τον κωδικό του. Αν είναι σωστά, μπορεί να κάνει λήψη της φωτογραφίας του προσώπου του για αναγνώριση. Ουσιαστικά, ακολουθούνται τα ίδια βήματα με τις ενότητες 7.4.1 και 7.4.2.

Η διαφορά στην εξής περίπτωση είναι πως, όταν ο χρήστης ολοκληρώσει επιτυχώς όλα τα βήματα της εισόδου στην υπηρεσία MyAuth, ανακατευθύνεται στο redirect_uri, στο οποίο έστειλε ο client το αρχικό αίτημα

7.6.2 Ανακατεύθυνση στον client και αίτηση access_token

Έτσι με την επιτυχή αυθεντικοποίηση του χρήστη γίνεται ανακατεύθυνση στον client. Το callback url αυτό περιέχει δύο query parameters. Αυτά είναι το authorization code και το state που έστειλε ο client στο προηγούμενο αίτημα. Ο client ελέγχει αν η τιμή του state που έστειλε πριν είναι η ίδια με αυτό που έλαβε στο callback url. Αν είναι ίδια σημαίνει πως η σύνδεση μεταξύ των δύο υπηρεσιών είναι έμπιστη.

Εφόσον αυτό ισχύει, μπορεί πλέον ο client να ζητήσει ένα access_token με το authorization code που μόλις έλαβε. Έτσι πραγματοποιεί ένα Post αίτημα στο MyAuth server με body που περιέχει τις παρακάτω τιμές:

- **grant_type :** Όπου υποδηλώνει το τύπο grant που ακολουθείται για αυτή τη ροή OAuth
- **code:** Που αποτελεί το authorization code που έλαβε ο client
- **redirect_uri:** Το url ανακατεύθυνσης που έχει δηλωθεί. Αυτό στέλνεται ξανά για έλεγχο ώστε να επιβεβαιώνεται για άλλη μια φορά η εμπιστευτικότητα της πηγής.
- **client_id :** Όπου αποτελεί το μοναδικό αναγνωριστικό της εφαρμογής URSAVINGS στη βάση δεδομένων της MyAuth. Και αυτό με τη σειρά του να ελέγχεται για ένα παραπάνω μέτρο ασφαλείας.

Ένα απόσπασμα κώδικα που έχει υλοποιηθεί στην εφαρμογή URSAVINGS για τη παραπάνω διαδικασία είναι το εξής:

```
router.get('/:provider/callback/login', function (req, res) {
  const provider = req.params.provider;
  const authCode = req.query.code;
  stateService.assertStateIsValid(req.session, req.query.state).then(() =>
    externalAuthService.login(provider, authCode, req.session).then(() => {
      res.redirect('/');
    })
  ).catch((err) => {
    res.redirect(`/login?msg=${err ? err : 'Login failed'}`);
  });
});
```

Εικόνα 7.24: Επαλήθευση κατάστασης state κατά τη λήψη authorization code

Εδώ γίνεται έλεγχος του state με τη μέθοδο assertStateIsValid(). Εφόσον ο έλεγχος του state είναι επιτυχής, καλείται η μέθοδος login. Μέσα εκεί, καλείται η μέθοδος getAccessToken(), από την οποία γίνεται ένα HTTP post request στο MyAuth server με το body που αναφέρθηκε παραπάνω:

```
login(provider: string, authCode: string, session: any): Promise<void> {
  const authProvider = getExternalAuthProvider(provider);
  return authProvider.getAccessToken(authCode, 'login').then((token: string) =>
    authProvider.getUserInfo(token).then(userInfo =>
      userRepository.getUserByExternalId(provider, userInfo.id).then(user => {
        session.user = user;
        log.info(`auth.${provider}.session_login_successful`, { user });
      }).catch(() => {
        log.error(`auth.${provider}.session_login_failed`, { userInfo });
        return Promise.reject('User not found');
      })
    )
  );
}
```

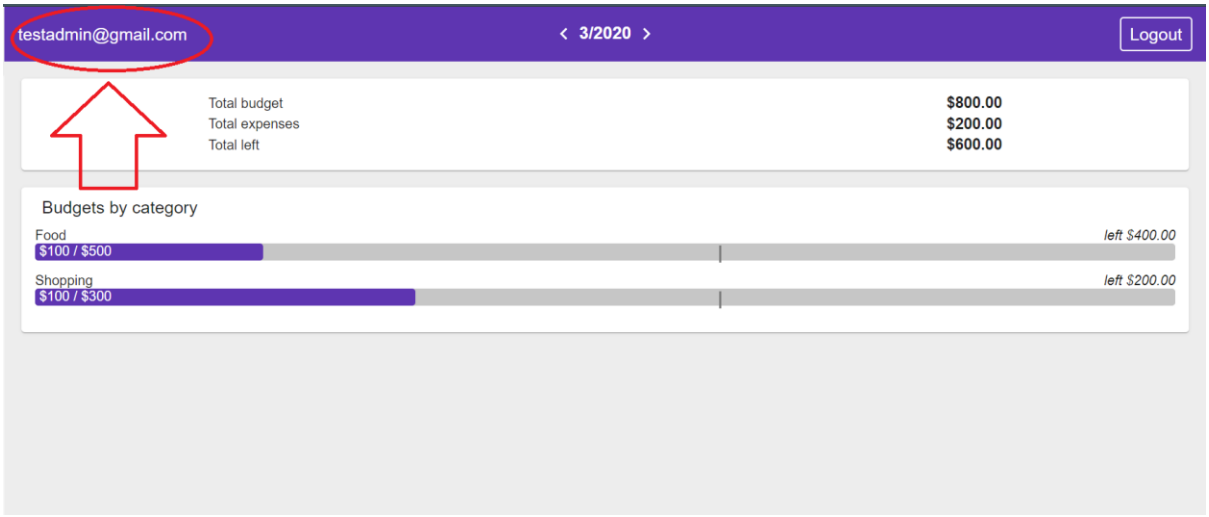
Εικόνα 7.25 : Ζήτηση access token με authorization code

7.6.3 Παραλαβή access_token και ζήτηση πληροφοριών χρήστη

Όταν ο server MyAuth παραλάβει το body του προηγούμενου αιτήματος, γίνεται έλεγχος για αν το grant type έχει τη τιμή "**authorization_code**". Όπως και πριν γίνεται έλεγχος ταιριάσματος cliend_id και redirect_uri και τέλος ελέγχεται αν το code που στάλθηκε στον client είναι έγκυρο και ταιριάζει με αυτό που υπογράφηκε γι' αυτόν συγκεκριμένα. Αν όλοι οι παραπάνω έλεγχοι είναι σωστοί, στέλνεται πίσω στον client το access_token και μπορεί πλέον με τη τιμή του να ζητήσει από τον MyAuth server ότι πληροφορία σχετίζεται με το προφίλ του χρήστη .

Έτσι, γίνεται ένα τελευταίο Http request στο endpoint του MyAuth server που σερβίρει τη πληροφορία του χρήστη. Αυτό προφανώς έχει στο body του το access token. Εφόσον το αίτημα φτάσει στο server, γίνεται έλεγχος για το αν το access_token είναι valid ή αν έχει λήξει. Αν η ημερομηνία λήξης δεν έχει παρέλθει, τότε ο server στέλνει πίσω στο client πληροφορία του χρήστη. Τότε ο client αυθεντικοποιηθεί το χρήστη και αποθηκεύονται στο session του ένα δικό του token, μαζί με τις πληροφορία που μόλις παρέλαβε.

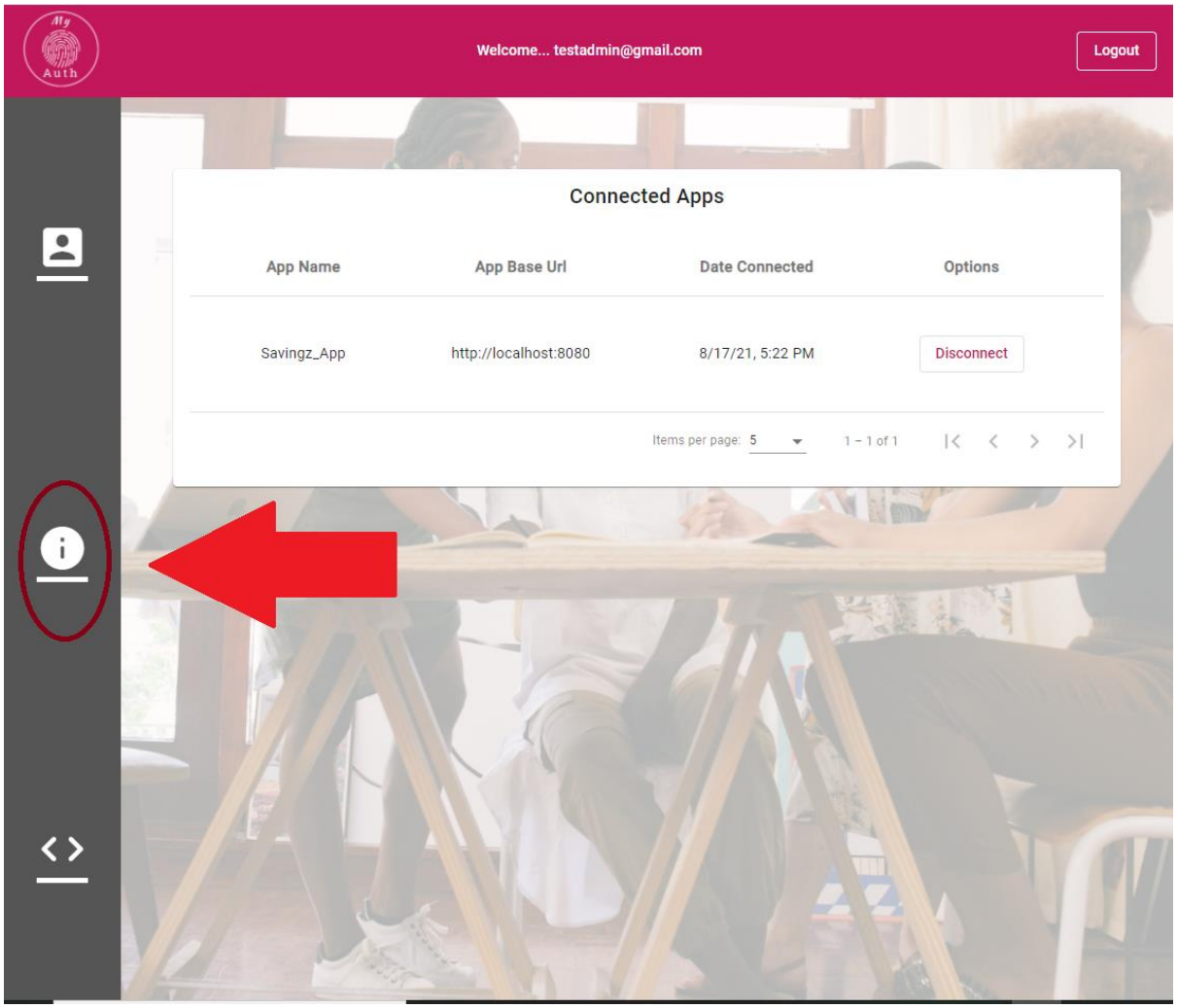
Τέλος ο χρήστης ανακατευθύνεται στη μέσα στην εφαρμογή και έτσι έχει πρόσβαση στις λειτουργίες που του προσφέρει αυτή. Στη παρακάτω εικόνα, φαίνεται πως η πληροφορία του χρήστη, όπως είναι το email, έχει επιτυχώς ληφθεί από τη client εφαρμογή.



Εικόνα 7.26 : Επιτυχής αυθεντικοποίηση και είσοδος στην εξωτερική εφαρμογή

7.7 Μέσα στην εφαρμογή

Όταν ένας χρήστης πραγματοποιεί είσοδο στην υπηρεσία MyAuth, ανακατευθύνεται σε ένα κεντρικό μενού. Σε αυτό του δίνονται κάποιες επιλογές.



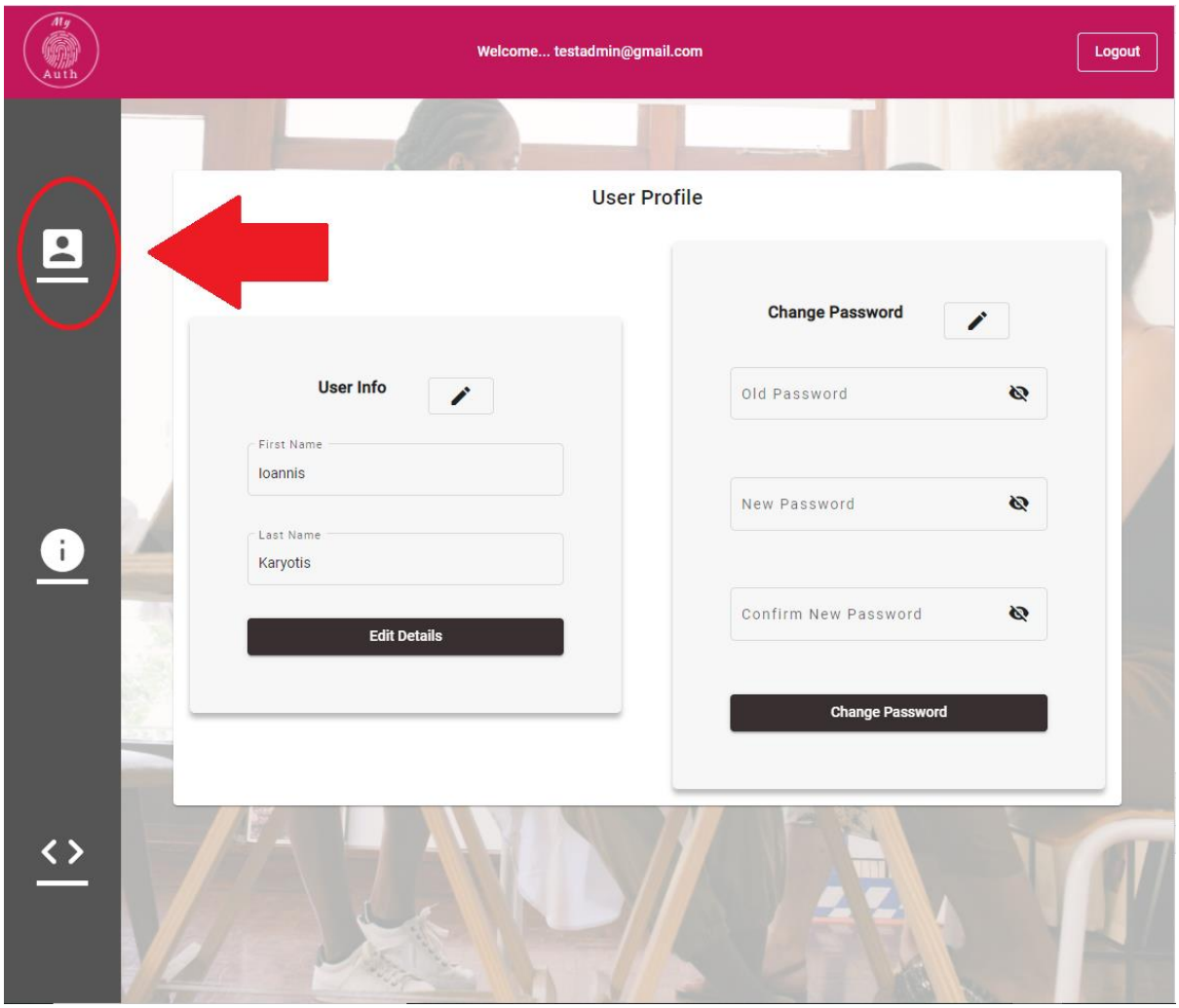
Εικόνα 7.27 : Πίνακας συνδεδεμένων εφαρμογών χρήστη μέσω MyAuth

Το μενού αριστερά, περιέχει 3 κουμπιά. Το δεύτερο κουμπί ανακατευθύνει το χρήστη στη σελίδα της παραπάνω εικόνας. Ουσιαστικά, δείχνει στο χρήστη από ποιες εξωτερικές εφαρμογές έχει πραγματοποιήσει σύνδεση, μέσω της υπηρεσίας MyAuth. Στη προκειμένη περίπτωση ο test admin που χρησιμοποιείται γι' αυτό το παράδειγμα έχει συνδεθεί στη εφαρμογή URSAVINGS που αναφέρθηκε στη προηγούμενη ενότητα. Λοιπές πληροφορίες για τις συνδεδεμένες εφαρμογές, είναι το url της εφαρμογής και η ημερομηνία σύνδεσης. Επίσης δίνεται η δυνατότητα στο χρήστη να αποσυνδεθεί η εγγραφή του από αυτή την εφαρμογή. Από τη πλευρά της άλλης εφαρμογής θα πρέπει να υλοποιείται ένας μηχανισμός διαγραφής των στοιχείων του χρήστη αυτού. Έτσι πατώντας το κουμπί disconnect, η υπηρεσία MyAuth θα επικαλείται την ενεργοποίηση αυτού του μηχανισμού από την άλλη εφαρμογή. Στη παρούσα περίπτωση διαγράφεται η εγγραφή της σύνδεσης του χρήστη από τη βάση δεδομένων της MyAuth.

Πατώντας το πρώτο κουμπί, ο χρήστης ανακατευθύνεται στη οθόνη τροποποίησης στοιχείων. Εκεί του δίνονται οι επιλογές να αλλάξει στοιχεία όπως το όνομα και το επίθετό του. Σε μελλοντικές εκδόσεις, θα μπορούσαν να εισαχθούν και άλλες πληροφορίες οι οποίες θα μπορούσαν να ληφθούν από πολλαπλές εφαρμογές. Τέτοια παραδείγματα πληροφοριών, θα μπορούσαν να είναι προσωπικά στοιχεία όπως η ηλικία, το ύψος, το βάρος αλλά και στοιχεία που τον αφορούν σαν πολίτη. Τέτοια θα μπορούσαν να είναι ο αριθμός φορολογικού μητρώου (ΑΦΜ) ή ο αριθμός μητρώου κοινωνικής

ασφάλισης (ΑΜΚΑ). Τέλος, δίνεται η δυνατότητα αλλαγής κωδικού πρόσβασης που εισάγει ο χρήστης στο πρώτο βήμα αυθεντικοποίησης κατά την είσοδό του.

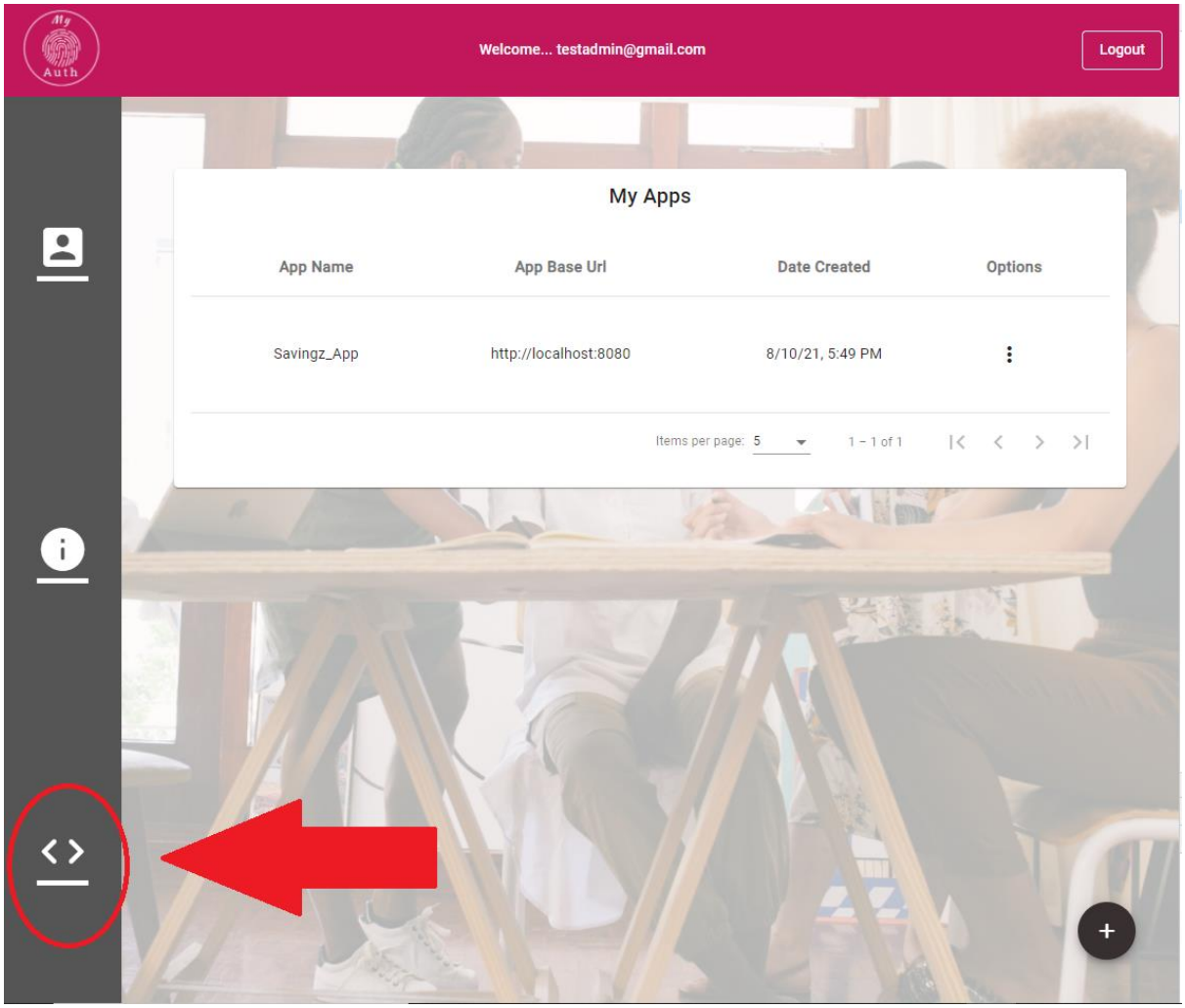
Παρακάτω παρουσιάζεται η οθόνη τροποποίησης στοιχείων :



Εικόνα 7.28 : Τροποποίηση στοιχείων χρήστη

Το τρίτο και τελευταίο κουμπί ανακατευθύνει το χρήστη σε μια οθόνη που απευθύνεται σε developers. Ουσιαστικά εκεί, μπορεί να προσθέσει και να ρυθμίσει μια εξωτερική εφαρμογή, από την οποία προορίζεται η υλοποίηση σύνδεσης χρηστών μέσω της υπηρεσίας MyAuth. Για παράδειγμα, η εξωτερική εφαρμογή URSAVINGS έχει ρυθμιστεί από κάποιο developer μέσω αυτής της οθόνης.

Πατώντας λοιπόν το τρίτο κουμπί, ο χρήστης ανακατευθύνεται και βλέπει μια λίστα εφαρμογών που έχουν ρυθμιστεί από τον ίδιο, ώστε να μπορούν αυτές να προσφέρουν αυτή τη διασύνδεση. Η οθόνη αυτή μοιάζει κάπως έτσι



Εικόνα 7.29 : Πίνακας εφαρμογών του χρήστη που προορίζονται για διασύνδεση με MyAuth

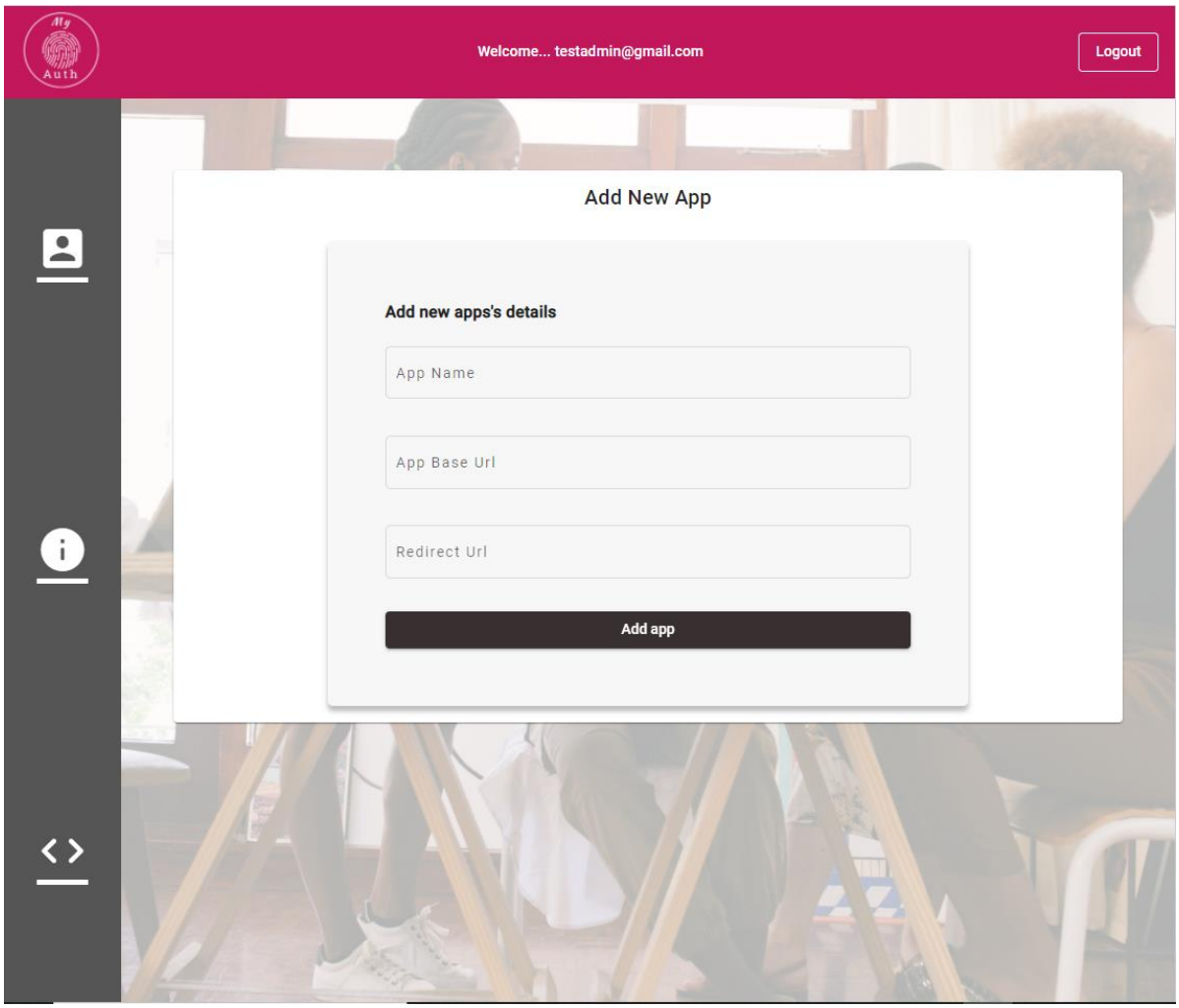
Όπως παρατηρείται, ο χρήστης testadmin@gmail.com έχει προσθέσει μία εφαρμογή. Αυτή είναι η URSAVINGS από τη οποία κιόλας έχει πραγματοποιήσει σύνδεση όπως είδαμε παραπάνω. Ένας developer για να μπορέσει να υλοποιήσει μια ροή σύνδεσης τύπου OAuth, θα πρέπει να εισάγει κάποια χαρακτηριστικά στην υπηρεσία που έχει τη προστατευμένη πληροφορία και στην εφαρμογή που υλοποιεί, συνεπώς η MyAuth και η URSAVINGS αντίστοιχα. Αυτό, γίνεται ώστε να υπάρχουν κοινά αναγνωριστικά μεταξύ τους, που θα συμβάλουν στην εμπιστευτικότητα και για να επιτευχθεί η τελική επικοινωνία.

Αυτά τα χαρακτηριστικά αποτελούν το Base URL που είναι hosted η εφαρμογή, το CallBack Url όπου θα ανακατευθύνεται ο user again όταν λαμβάνει το authorization code και δύο μοναδικά αναγνωριστικά για την εξωτερική εφαρμογή. Αυτά είναι το client id και το client secret, τα οποία είναι μοναδικά για κάθε εφαρμογή και μπορούν να ανανεώνονται.

Αυτά τα στοιχεία λοιπόν θα πρέπει να αποθηκεύονται και από την MyAuth στη βάση δεδομένων και από την εξωτερική εφαρμογή που θα επιτρέπει την επικοινωνία μέσω αυτής.

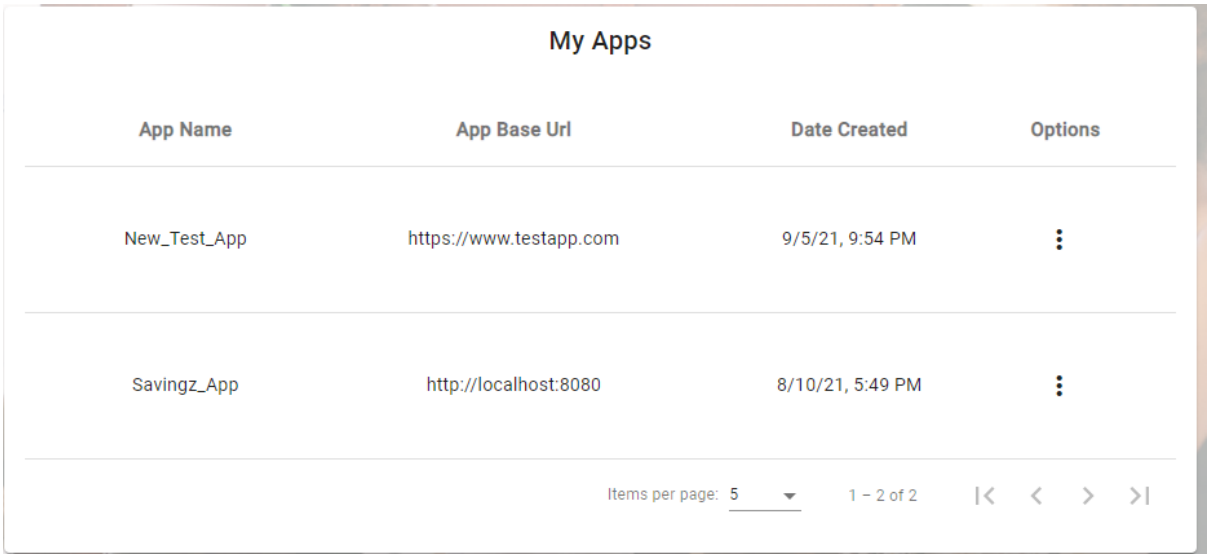
Πατώντας λοιπόν ο χρήστης το κουμπί πρόσθεσης κάτω δεξιά, πηγαίνει στην οθόνη προσθήκης μιας καινούργιας εφαρμογής που προορίζεται για διασύνδεση.

Η επόμενη οθόνη προσθήκης νέας εφαρμογής είναι η εξής:

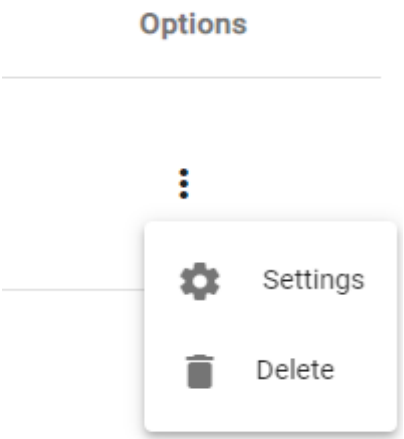


Εικόνα 7.30 : Προσθήκη νέας εφαρμογής για διασύνδεση με MyAuth

Ο developer καλείται να εισάγει το όνομα της εφαρμογής, το URL της και το redirect URL. Κάνοντας λοιπόν προσθήκη μια νέας εφαρμογής, αυτή εγγράφεται στη βάση δεδομένων και προστίθεται στη προηγούμενη λίστα.



Εικόνα 7.31: Επιτυχής δημιουργία νέας εφαρμογής για διασύνδεση με MyAuth

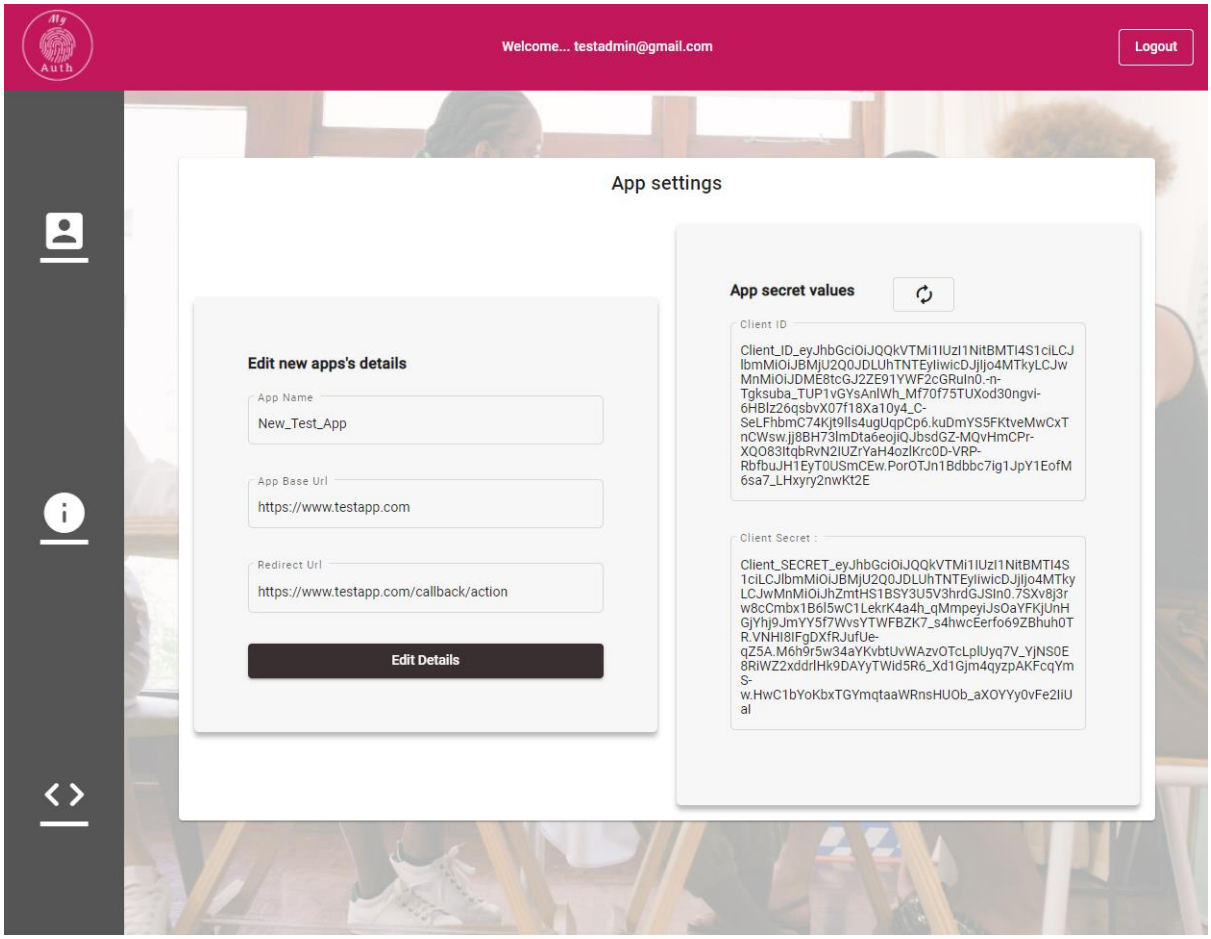


Μετά από αυτό το βήμα, ο developer έχει δύο επιλογές. Πατώντας το κουμπί “Options” για κάθε μία εφαρμογή από τη λίστα μπορεί να επιλέξει διαγραφή ή τροποποίηση των στοιχείων που εισήγαγε πριν.

Αν επιλέξει Διαγραφή, τότε η εφαρμογή διαγράφεται και τα στοιχεία της παύουν πια να υπάρχουν στη βάση δεδομένων. Έτσι ταυτόχρονα η διασύνδεση παύει να υπάρχει και οποιαδήποτε προσπάθεια γίνει από τη client εφαρμογή, θα είναι άκυρη.

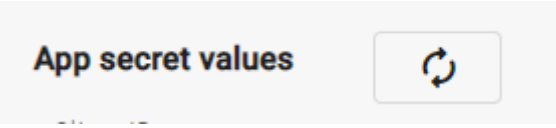
Αν επιλεχθεί το κουμπί “Settings” τότε ο developer μπορεί να δει τα στοιχεία της και τα τροποποιήσει. Στην παρακάτω οθόνη φαίνονται επίσης και το client id και το cliend secret που δημιουργούνται αυτόματα κατά της δημιουργία της:

Εικόνα 7.32: Επιλογές διαχείρισης εφαρμογής χρήστη



Εικόνα 7.33: Τροποποίηση στοιχείων εφαρμογής χρήστη

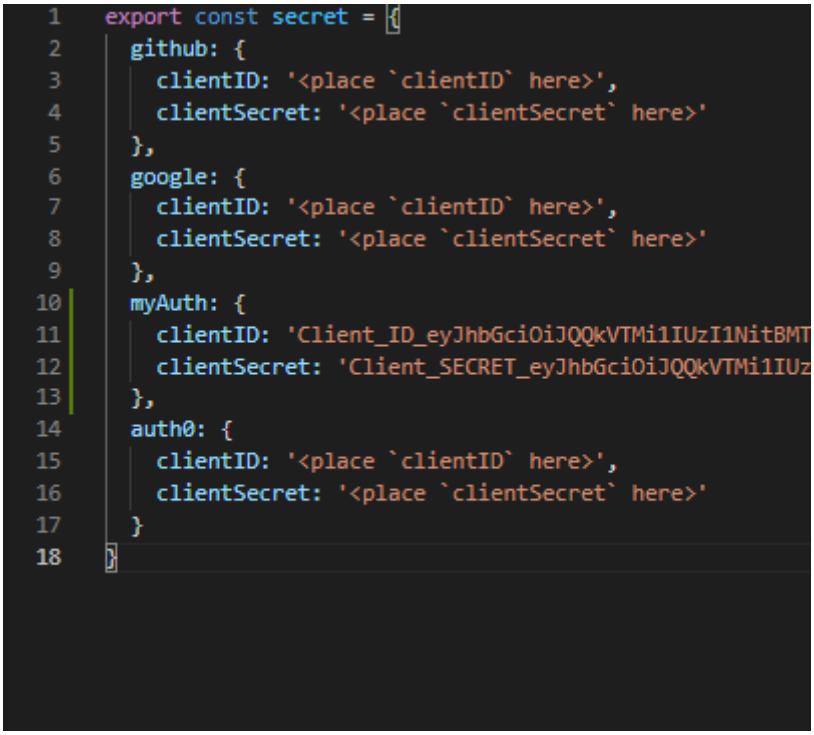
Υπάρχει δυνατότητα για ανανέωση του client ID και του client Secret, πατώντας του κουμπί ανανέωσης μέσα στο πλαίσιο “App secret values”



Εικόνα 7.34: Ανανέωση Client Id και Client Secret για την εφαρμογή

Εν κατακλείδι, το τελευταίο βήμα που έχει να κάνει ο developer για την αρχή της υλοποίησης σύνδεσης με OAuth, είναι να αποθηκεύσω το redirect url, το client id και το client secret στη δική του εφαρμογή. Αυτά τα πεδία θα σταλούν στο MyAuth server σε κάθε απόπειρα σύνδεσης από τη εξωτερική εφαρμογή.

Για παράδειγμα η εφαρμογή URSAVINGS αποθηκεύει τις δικές της τιμές σε ένα configuration json file που έχει αποθηκευμένο στο σύστημα αρχείων της και τις παίρνει από κει ώστε να σταλούν για επαλήθευση.



Εικόνα 7.35: Αποθήκευση μυστικών κλειδιών στην εξωτερική εφαρμογή

Όπως φαίνεται το πεδίο MyAuth έχει το client ID και το client Secret. Μαζί υπάρχουν πεδία και από άλλες υπηρεσίες που μπορεί να υλοποιούν σύνδεση με OAuth όπως είναι η Google ή το Github.

Σε αυτό το σημείο τελειώνει και η ροή σύνδεσης μιας εφαρμογής μέσω της υπηρεσίας MyAuth.

7.8 Επίλογος

Τα παραπάνω βήματα λοιπόν, αποτελούν οι οθόνες της υλοποιημένης πιλοτικής εφαρμογής. Γίνεται διατύπωση με λόγια, εικόνες και κώδικα για τα σημαντικότερα σημεία της κύριας ροής σύνδεσης, εγγραφής αλλά και του μενού διαχείρισης του χρήστη. Επιπλέον γίνεται επεξήγηση για το πως παρέχεται στο συνδεδεμένο χρήστη μια πολύ βασική δυνατότητα υλοποίησης διασύνδεσης μέσω της ροής OAuth 2.0 με άλλες εφαρμογές. Στο επόμενο κεφάλαιο θα προταθούν κάποιοι τρόποι και ιδέες για το πώς θα μπορούσε η παρούσα υλοποίηση να αναπτυχθεί και να προσφέρει ακόμα περισσότερες υπηρεσίες σε χρήστες. Τέλος, γίνεται πρόταση για το πως αυτή θα μπορούσε να εξελιχθεί για να συνυπάρχει σε ένα διαδικτυακό περιβάλλον και να προσφέρει υπηρεσίες σε εφαρμογές του πραγματικού κόσμου.

Κεφάλαιο 8ο: Συμπεράσματα και προτάσεις βελτίωσης

Η παρούσα υλοποίηση της πιλοτικής εφαρμογής, παρέχει βασική λειτουργικότητα της ροής προτύπου OAuth 2.0 με authorization code grant. Ο κάθε χρήστης μπορεί να πραγματοποιήσει εγγραφή, είσοδο με τα βιομετρικά του χαρακτηριστικά και του δίνονται δυνατότητες τροποποίησης των στοιχείων του καθώς και η διαχείριση εξωτερικών εφαρμογών που προορίζονται για διασύνδεση μέσω της υπηρεσίας MyAuth. Η υπηρεσία αυτή, αν και παραπέμπει στην φύση ενός μοντέλου Biometric as a service δεν κάνει χρήση κάποιου συνδρομητικού πλάνου με πίστωση, όπως εξηγεί ο ορισμός του μοντέλου as a Service. Θεωρητικά, αν η υπηρεσία φιλοξενείται σε κάποιο server, θα μπορούσε να χρησιμοποιηθεί δωρεάν από άλλες υπηρεσίες ή εφαρμογές που θα ήθελαν να ενσωματώσουν τη βιομετρική αυθεντικοποίηση στο σύστημά τους.

Μια σημαντική βελτίωση της παρούσας υλοποίησης, θα ήταν η υλοποίηση και άλλων μεθόδων παραλαβής access_token, που ορίζει το πρωτόκολλο OAuth 2.0. Για παράδειγμα, ροής παραλαβής grant με Implicit flow και άλλα. Αυτή η βελτίωση θα επέτρεπε και σε άλλων ειδών εφαρμογές να κάνουν διασύνδεση μέσω της υπηρεσία MyAuth. Για παράδειγμα, η ροή Implicit που αποσκοπεί μόνο σε εφαρμογές βασισμένες στον client.

Μια δεύτερη βελτίωση που θα μπορούσε να υπάρξει, είναι η ύπαρξη διαφορετικών ειδών scope που θα μπορούσε να ζητήσει η τρίτη εφαρμογή. Αυτή τη στιγμή μπορεί να ζητήσει μόνο χαρακτηριστικά του προφίλ του χρήστη. Θα ήταν χρήσιμη λοιπόν παροχή διαφορετικών πληροφοριών με χρήση άλλων scopes. Τέλος, μέσα σε αυτά τα scopes θα μπορούσε να είναι και το open id connect scope, όπου σύμφωνα με τη κατάλληλη υλοποίηση και ακολουθώντας το πρότυπο OICD, η υπηρεσία θα επέστρεφε ένα JWT όπως περιεγράφηκε στο κεφάλαιο 3.

Επιπλέον, με τη διαλειτουργική αυθεντικοποίηση, θα ήταν σωστό να γνωστοποιηθεί το API που αφορά τη υλοποίηση σύνδεσης με OAuth 2.0 και να υπάρχουν οδηγοί σχετικά με όλους πιθανούς τρόπους και σενάρια διασυνδέσεων μέσω της υπηρεσίας MyAuth.

Ένα άλλο μεγάλο κομμάτι που θα μπορούσε να αποτελέσει βελτίωση της παρούσας υπηρεσίας, είναι η υλοποίηση συνδρομητικών πλάνων που θα αφορούσαν καθαρά μεγάλους οργανισμούς. Ένας οργανισμός που θα επέλεγε ένα συνδρομητικό πλάνο, θα του δινόταν περισσότερες λειτουργίες διαχείρισης. Ο λογαριασμός του οργανισμού θα αφορούσε όλους τους χρήστες του και η διασύνδεση μεταξύ εφαρμογής και υπηρεσίας θα γινόταν με διαφορετικό τρόπο αποκλειστικά για τον οργανισμό αυτό. Ο διαχειριστής του οργανισμού θα είχε πρόσβαση στους διασυνδεδεμένους χρήστες του, θα μπορούσε να τους διαχειριστεί ή να τους διαγράψει. Τέλος, θα μπορούσε ο ίδιος ακόμα να ορίσει το βαθμό χαλαρότητας (threshold) επαλήθευσης κατά την αυθεντικοποίηση των χρηστών του με βιομετρικά προσώπου.

Τέλος, για να γίνουν όλα τα παραπάνω εφικτά, θα πρέπει η υπηρεσία να φιλοξενηθεί εν τέλη σε κάποιο server όπου θα είναι ορατός από όλους στο διαδίκτυο. Όλα τα μέτρα ασφαλείας θα πρέπει να τηρούνται για να παραμένουν τα δεδομένα των χρηστών ασφαλή και να γίνει, εν τέλη, η επικοινωνία μεταξύ άλλων οργανισμών, υπηρεσιών, εφαρμογών ή μεμονωμένων ανθρώπων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Βιομετρία, Wikipedia. [Online]. Available: .

[2] Ευαγγελία Γ. Αλεξοπούλου, "Βιομετρικό σύστημα πρόσβασης (Μεταπτυχιακή εργασία) ", pp. 9, 2017.

[3] Stephen Mayhew, "History of Biometrics", 2012. [Online]. Available: History of Biometrics.

[4] Amanda Douglas, "A Brief History of Biometrics", 2020. [Online]. Available: A Brief History of Biometrics.

[5] "Πλαίσιο Ηλεκτρονικής Διακυβέρνησης", Vol 4.0, pp. 39-45, 2012. [Online]. Available: Πλαίσιο Ψηφιακής Αυθεντικοποίησης.

[6] Stamoulis Petros (MSC ICSD Aegean University), "Αυθεντικοποίηση οντοτήτων στις υπηρεσίες ηλεκτρονικής διακυβέρνησης", 2015. [Online]. Available: Κατηγορία:Αυθεντικοποίηση οντοτήτων στις υπηρεσίες ηλεκτρονικής διακυβέρνησης - wiki Πλαισίου Ηλεκτρονικής Διακυβέρνησης

[7] Anil K. Jain, Arun Ross, and Salil Prabhakar, "An introduction to biometric recognition". in IEEE Trans. on Circuits and Systems for Video Technology, pp. 4–20, 2004.

[8] Dean Nicolls, "What is Biometric Authentication?", Jumio, 2019. [Online]. Available: Biometric Authentication: Complete Overview for 2020

[9] ISO/IEC 24745:2011 , "Information technology — Security techniques — Biometric information protection" , 2011. [Online]. Available: https://www.iso.org/standard/52946.html

[10] Biometric Authentication, 2021. [Online]. Available: What is Biometric Authentication?.

[11] Αγγελική Ζαπαλίδη, "Βιομετρική Αυθεντικοποίηση σε Android (Μεταπτυχιακή εργασία)" , pp. 12- 16, 2020. [Online]. Available: Βιομετρική Αυθεντικοποίηση σε Android.

[12] Technopedia, "Biometric System" , 2012. [Online]. Available: What is a Biometric System? - Definition from Techopedia

[13]Γεροντίδης Ευγένιος, "Βιομετρικά Συστήματα Ασφαλείας. Τεχνικές Υλοποίησης και Εφαρμογές τους (Πτυχιακή Εργασία)", pp.7-29, 2012.

[14]Julius Davies, Daniel German, Mike Godfrey, and Abram Hindle, "Software Bertillonage: Finding the Provenance of an Entity", Presented in IEEE Intl Conf on Mining Software Repositories, 2004. [Online]. Available: software bertillonage finding the provenance of an entity.

[15] Thales Group, "Biometrics: definition, use cases and latest news", 2021. [Online]. Available: Biometric news in 2021.

[16] ΠΛΕΜΜΕΝΟΠΟΥΛΟΣ ΠΑΝΑΓΙΩΤΗΣ, "ΗΛΕΚΤΡΟΝΙΚΕΣ ΥΠΗΡΕΣΙΕΣ ΑΞΙΟΛΟΓΗΣΗΣ ΤΜΗΜΑΤΟΣ (Πτυχιακή εργασία)", pp. 13, 2013. [Online]. Available: ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ.

[17] Καρατσιώλη Ευαγγελία, "Αυθεντικοποίηση με πολύτροπα βιομετρικά συστήματα", Πτυχιακή εργασία, pp. 42- 58, 2012. [Online]. Available: Αυθεντικοποίηση με πολύτροπα βιομετρικά χαρακτηριστικά.

[18] OWasp, "Manipulator-in-the-middle attack". [Online] Available: Manipulator-in-the-middle attack | OWASP

[19] Vincent Mühler, "face-api.js — JavaScript API for Face Recognition in the Browser with tensorflow.js", 2018. [Online]. Available: face-api.js — JavaScript API for Face Recognition in the Browser with tensorflow.js.

[20] Tensorflow.js, "Tensorflow Documentation". [Online]. Available: Get Started.

[21] Σκουρλής Ιωάννης, "ΜΕΘΟΔΟΙ ΑΝΙΧΝΕΥΣΗΣ ΚΑΙ ΑΝΑΓΝΩΡΙΣΗΣ ΠΡΟΣΩΠΟΥ (Διπλωματική εργασία)", pp. 8-9, 2016. [Online] Available: ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ.

[22] Sumit Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way", Towards Data Science, 2018. [Online]. Available: A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.

[23] Wikipedia, "WebGL", [Online]. Available: WebGL - Βικιπαίδεια.

[24] Internet Engineering Task Force (IETF), "PKCS #5: Password-Based Cryptography Specification", vol. 2.1, pp. 13-16, 2017. [Online]. Available: rfc8018.

[25] Ανοικτή Διακυβέρνηση Υπουργείου Παιδείας, Έρευνας και Θρησκευμάτων, "Διαλειτουργικότητα". [Online]. Available: Διαλειτουργικότητα – Ανοικτή Διακυβέρνηση.

[26] Okta, "SAML vs. OAuth: Comparison and Differences". [Online]. Available: SAML vs. OAuth: Comparison and Differences

[27] Internet Engineering Task Force (IETF), "The OAuth 2.0 Authorization Framework", pp. 4-42, 2012. [Online]. Available: rfc6749

[28] Wikipedia, "Software as a Service". [Online]. Available: Software as a service