



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη contextual AI chatbot σε περιβάλλοντα
υποβοηθούμενης διαβίωσης

ΓΕΩΡΓΙΟΣ ΚΑΡΑ ΖΑΦΕΙΡΗΣ

ΕΠΙΒΛΕΠΩΝ: Δρ. Αντωνόπουλος Χρήστος
ΣΥΝΕΠΙΒΛΕΠΟΥΣΑ: Δρ. Φαλιάγκα Ευανθία

ΠΑΤΡΑ 2022

Περιεχόμενα	
Κεφάλαιο 1 - Εισαγωγή	2
Κεφάλαιο 2 - Διασύνδεση & Συλλογή Μετρήσεων	8
Κεφάλαιο 3 - Ενοποίηση Chatbot	14
Κεφάλαιο 4 - Ανάλυση Υλοποίησης Κώδικα	18
Κεφάλαιο 5 - Συμπεράσματα και επόμενα βήματα	28
Βιβλιογραφία	35

Περιεχόμενα Εικόνων

→ Εικόνα 1. Υπηρεσίες AAL	6
→ Εικόνα 2. Οι τρεις γενιές του AAL	9
→ Εικόνα 3. Οι Υπηρεσίες ενός Chatbot	21
→ Εικόνα 4. Η σύνδεση και επικοινωνία αντικειμένων στην αρχή της αντικειμενοστρέφειας	22
→ Εικόνα 5. Η εξουσιοδότηση του client για την χρήση του REST API	27
→ Εικόνα 6. Η δομή της λάμδα ανώνυμης συνάρτησης στην Python	28

Abstract

This thesis aims to study and analyse measurements extracted from a group of smart home sensors within ESDALAB. More specifically, the collected values will be checked against values from external APIs referring to weather parameters and will return values according to the location specified. Measurements will also be extracted from a pressure sensor in order to recognize the user's sitting patterns during his stay in the smart home. The data will then be collected and the results will be processed and structured for use in the Kalliope structure. Data sets collected from all remote interfaces will be normalised (where necessary) and used in machine learning algorithms for either prediction estimations or result classification.

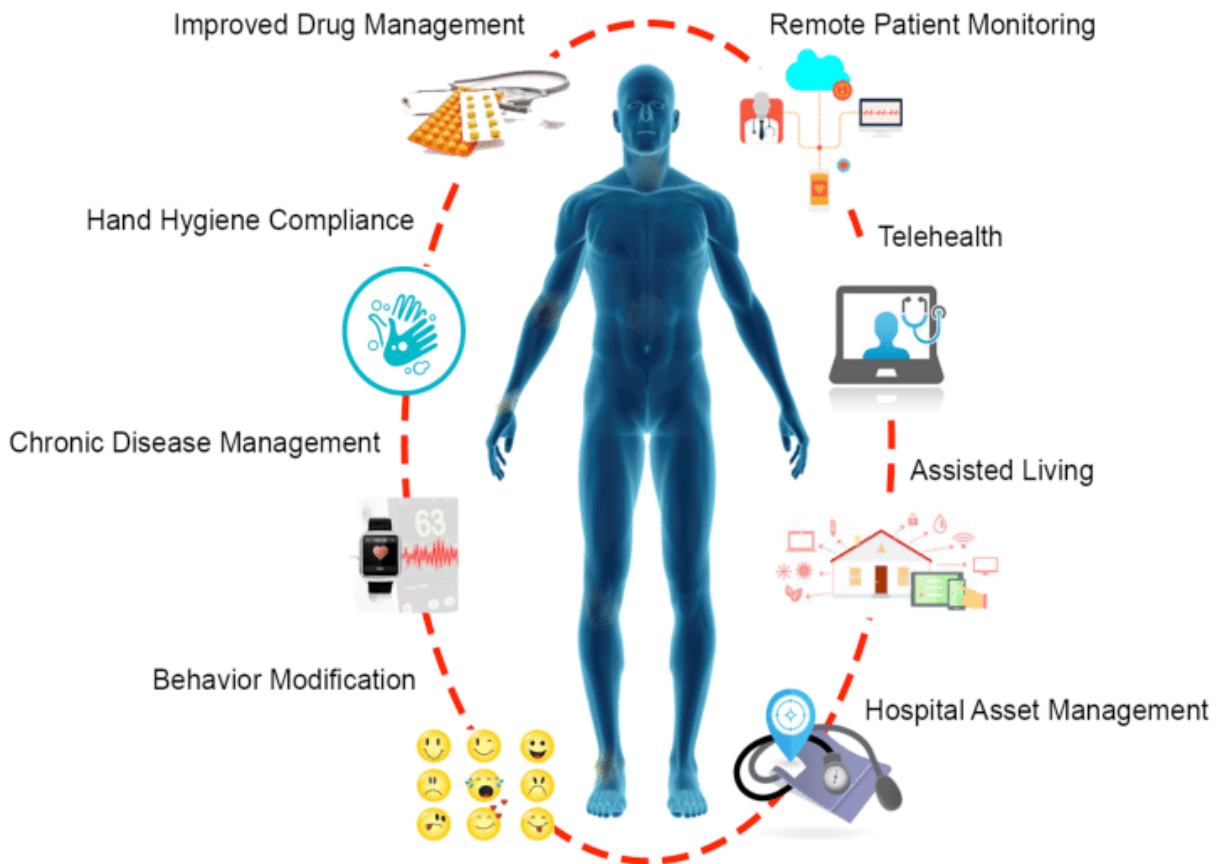
Κεφάλαιο 1 - Εισαγωγή

Η παρούσα πτυχιακή, έχει στόχο να μελετήσει και να αναλύσει μετρήσεις που εξάγονται από μια ομάδα αισθητήρων του έξυπνου σπιτιού εντός του ESDALAB. Πιο αναλυτικά, οι τιμές που θα συλλέγονται θα εξετάζονται συγκριτικά με τιμές από εξωτερικά API που αναφέρονται σε παραμέτρους καιρικών συνθηκών και θα επιστρέφουν τιμές σύμφωνα με την τοποθεσία που ορίζεται. Μετρήσεις θα εξαχθούν επίσης και από έναν αισθητήρα πίεσης με σκοπό την αναγνώριση καθιστικών μοτίβων του χρήστη κατά την διαμονή του στο έξυπνο σπίτι. Τα στοιχεία, τότε θα συλλέγονται και θα γίνεται η επεξεργασία και η δόμηση των αποτελεσμάτων για αξιοποίησή τους στην δομή της Καλλιόπης. Τα σύνολα δεδομένων που θα έχουν συλλεχθεί από όλες τις απομακρυσμένες διεπαφές θα κανονικοποιούνται, (όπου αυτό είναι απαραίτητο) και θα χρησιμοποιούνται σε αλγορίθμους μηχανικής μάθησης είτε για εκτιμήσεις προβλέψεων είτε για ταξινόμηση αποτελεσμάτων.

Ο ψηφιακός βοηθός ή αλλιώς το παράγωγο bot από τα δομικά στοιχεία της Καλλιόπης, θα δέχεται κάποια σήματα - ερεθίσματα, τα οποία στη προκειμένη θα είναι στην μορφή ερωτήσεων. Τότε ο ψηφιακός βοηθός θα είναι σε θέση να συνδέσει την συγκεκριμένη ερώτηση με μια απάντηση νευρώνα η οποία θα επιστρέφει στην περίπτωση μας, τις διαφορές στις τιμές των καιρικών συνθηκών και πίεσης. Σε αυτό το σημείο είναι σημαντικό να θέσουμε ένα ερώτημα. Γιατί χρειάζεται να υπάρχει ένας ψηφιακός βοηθός και ποιός ο ρόλος του στο πλαίσιο του έξυπνου σπιτιού; Καθημερινά, ο αριθμός των ανθρώπων που χρειάζονται υποστήριξη αυξάνεται σημαντικά την τελευταία δεκαετία και θα συνεχίσει να αυξάνεται σταθερά στο μέλλον. Αυτό ισχύει για τον υποπληθυσμό των ηλικιωμένων ενηλίκων, αλλά και για άτομα με γνωστικά προβλήματα με συγγενή ή επίκτητα ελαττώματα του εγκεφάλου, των οποίων το προσδόκιμο όριο ζωής είναι στις μέρες μας όπως όλων των άλλων.

Τα άτομα και από τις δύο ομάδες μπορεί να παρουσιάσουν γνωστικούς περιορισμούς, είτε λόγω ψυχικής φθοράς που σχετίζεται με την ηλικία, άνοιας ή αναπηρίας. Τέτοιοι περιορισμοί μπορεί να κυμαίνονται από σοβαρή λήθη έως πλήρη έλλειψη αίσθησης του χρόνου και μπορεί να τους δυσκολέψουν να οργανώσουν και να διατηρήσουν αυτόνομα την καθημερινή τους ζωή με τακτικές δραστηριότητες όπως γεύματα, φάρμακα ή κοινωνικές εκδηλώσεις, καθώς και έκτακτα γεγονότα όπως ραντεβού με γιατρό. Ωστόσο, το να επιτραπεί σε αυτούς τους ανθρώπους να παραμείνουν αυτόνομα στο περιβάλλον του σπιτιού τους και να έχουν έναν αυτοκαθορισμένο τρόπο ζωής για όσο το δυνατόν περισσότερο, έχει χαρακτηριστεί ως ένας από τους σημαντικότερους κοινωνικούς στόχους του μέλλοντος. Τα συστήματα περιβαλλοντικού ελέγχου αναπτύχθηκαν αρχικά τη δεκαετία του 1960 για ασθενείς που είχαν υποστεί τραυματισμό του νωτιαίου μυελού. Ο στόχος αυτών των συστημάτων και συστημάτων περιβαλλοντικού ελέγχου σήμερα είναι να επιτρέψουν

στα άτομα με αναπηρία να χειρίζονται εξοπλισμό στο άμεσο περιβάλλον τους, δίνοντάς τους τη δυνατότητα να κάνουν πράγματα όπως να απαντούν σε ενδοεπικοινωνία, να ανοίγουν την πόρτα, να καλούν βοήθεια ή να χειρίζονται τηλεόραση κ.λπ.



Εικόνα 1. Υπηρεσίες AAL

Τα πρώτα συστήματα ήταν μεγάλες επιτοίχιες συσκευές συνδεδεμένες με καλώδια στις συσκευές που λειτουργούσαν (ενσύρματα). Σημαντικές εξελίξεις στον περιβαλλοντικό έλεγχο έχουν σημειωθεί από τη δεκαετία του '60 κυρίως λόγω της προόδου της τεχνολογίας όπως πομποδέκτες ραδιοφώνου και υπέρυθρων, τρανζίστορ και μικροελεγκτές. Αυτές οι εξελίξεις επέτρεψαν στις συσκευές να γίνουν μικρότερες και φορητές, να εκτελούν περισσότερες λειτουργίες και να μην είναι πλέον συνδεδεμένες με τις περιφερειακές συσκευές.

Παρά αυτές τις προόδους, οι κύριες μέθοδοι λειτουργίας των συστημάτων περιβαλλοντικού ελέγχου έχουν αλλάξει πολύ λίγο. Τα περισσότερα συστήματα λειτουργούν είτε μέσω ενός μόνο διακόπτη, ενός αριθμού διακοπών ή μέσω ενός πληκτρολογίου στη συσκευή. Ενώ η είσοδος του πληκτρολογίου μπορεί να είναι μια αποτελεσματική μέθοδος ελέγχου για εκείνους των οποίων οι φυσικές ικανότητες είναι σχετικά καλές, αυτό δεν ισχύει για πολλούς ανθρώπους που έχουν ανάγκη για συστήματα περιβαλλοντικού ελέγχου. Η πλειονότητα των συστημάτων

περιβαλλοντικού ελέγχου ελέγχεται από την πρόσβαση διακόπτη, η οποία μπορεί να είναι αργή, επίπονη και απογοητευτική. Η ομιλία είναι μια πιθανή μέθοδος ελέγχου για πολλούς τρέχοντες και πιθανούς χρήστες περιβαλλοντικών ελέγχων ή άλλης υποστηρικτικής τεχνολογίας και μπορεί να κάνει τον περιβαλλοντικό έλεγχο ταχύτερο, λιγότερο επίπονο και λιγότερο απογοητευτικό. Ωστόσο, οι υπάρχοντες περιβαλλοντικοί έλεγχοι που βασίζονται στην ομιλία φαίνεται να έχουν πολύ χαμηλό ποσοστό συνταγογράφησης. Ο έλεγχος ομιλίας των προσωπικών υπολογιστών είναι ευρύτερα γνωστός και χρησιμοποιείται εντός του πεδίου, γεγονός που υποδηλώνει ότι ο έλεγχος ομιλίας είναι αποδεκτός από τους χρήστες. Το ερώτημα γιατί η ομιλία δεν χρησιμοποιείται ευρέως ως μέθοδος ελέγχου για συστήματα περιβαλλοντικού ελέγχου παραμένει ανοιχτό.

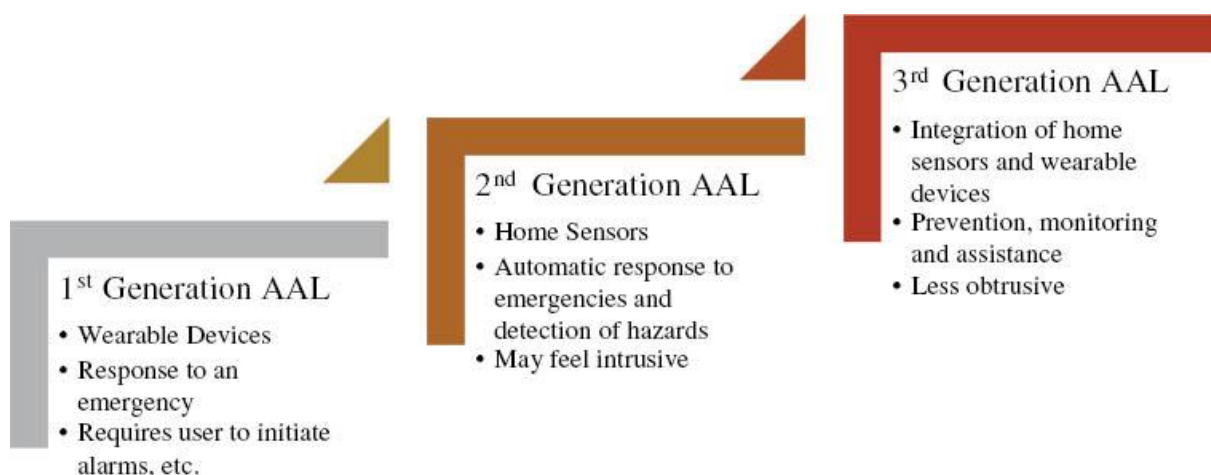
Οι τεχνολογίες υποβοηθούμενης διαβίωσης (AAL) έχουν εμφανιστεί τις τελευταίες δεκαετίες ως παρέχοντας καινοτόμες προσεγγίσεις στις προκλήσεις ενός γηράσκοντος πληθυσμού. Η τεχνολογία AAL έχει τη δυνατότητα να επιτρέψει στους ανθρώπους να ζουν και να γερνούν καλά με τους ακόλουθους τρόπους: υποστήριξη ατόμων να διατηρήσουν και να συνεχίσουν τις τρέχουσες δραστηριότητές τους. διευκόλυνση και διευκόλυνση της συνεχούς συμμετοχής και εμπλοκής σε δραστηριότητες στο σπίτι και στην κοινότητα· και βελτίωση της σχέσης κόστους-αποτελεσματικότητας και ποιότητας των υπηρεσιών υγείας και κοινωνικών υπηρεσιών. Ο όρος «υποβοηθούμενη στο περιβάλλον διαβίωση» χρησιμοποιείται ευρέως στη γεροτεχνολογική έρευνα και ανάπτυξη παρά το γεγονός ότι δεν έχει υιοθετηθεί καθολικός ορισμός της AAL. Ωστόσο, το AAL αναφέρεται γενικά στη χρήση τεχνολογιών πληροφοριών και επικοινωνιών (ΤΠΕ), αυτόνομων βοηθητικών συσκευών και τεχνολογιών έξυπνου σπιτιού στο καθημερινό περιβάλλον διαβίωσης και εργασίας ενός ατόμου για να επιτρέψουν στα άτομα να παραμείνουν ενεργά περισσότερο, να παραμείνουν κοινωνικά συνδεδεμένα και να ζουν ανεξάρτητα. σε μεγάλη ηλικία. Το AAL παρέχει υποστηρικτικά οικιακά περιβάλλοντα ενσωματώνοντας αισθητήρες, ενεργοποιητές, έξυπνες διεπαφές και τεχνητή νοημοσύνη. Το AAL έχει τις ρίζες του στα εξής: παραδοσιακές υποστηρικτικές τεχνολογίες για άτομα με αναπηρία. καθολικές προσεγγίσεις σχεδιασμού για την προσβασιμότητα, τη χρηστικότητα και την αποδοχή των διαδραστικών τεχνολογιών· και το υπολογιστικό παράδειγμα αναδυόμενης νοημοσύνης περιβάλλοντος (AI), το οποίο παρέχει έξυπνη, διακριτική και πανταχού παρούσα βοήθεια [50].

Η τεχνολογία AAL έχει την ικανότητα να προβλέπει και να ανταποκρίνεται στις μεταβαλλόμενες ανάγκες των ηλικιωμένων. Ωστόσο, η δυνατότητα της τεχνολογίας AAL είναι να είναι προσαρμοστική στην ετερογενή φύση της τρίτης ηλικίας. Μέχρι σήμερα, η έρευνα της AAL έχει επικεντρωθεί στενά στις δυνατότητες τέτοιων τεχνολογιών για ενήλικες μεγαλύτερης ηλικίας που είναι υγιείς, αδύναμοι ή πάσχουν από άνοια. Η προηγούμενη έρευνα έχει παραμελήσει σε μεγάλο βαθμό τις δυνατότητες της τεχνολογίας AAL να βελτιώσει τη ζωή όσων έχουν ήπια γνωστική εξασθένηση (MCI). Η MCI ορίζεται ως μια αισθητή μείωση των γνωστικών ικανοτήτων (όπως η μνήμη, η λήψη αποφάσεων, η επίλυση προβλημάτων και η

κατανόηση) που δεν εμποδίζει τις καθημερινές δραστηριότητες. Εμφανίζεται σε σημαντικό μέρος του ηλικιωμένου πληθυσμού, επηρεάζοντας έως και το ένα πέμπτο των ατόμων ηλικίας άνω των 65 ετών. Η αποτελεσματική υποστήριξη για ανεξάρτητη διαβίωση και γήρανση είναι ιδιαίτερα κρίσιμη για τα άτομα με MCI καθώς, εάν είναι επιτυχής, είναι δυνατόν να διατηρήσουν υψηλή ποιότητα ζωής. Αντίθετα, η αποτυχία μπορεί να οδηγήσει σε ταλαιπωρία ή/και υψηλό κόστος που σχετίζεται με την αποκλειστική φροντίδα. Η AAL είναι η πιο ελκυστική επιλογή για την παροχή τέτοιας υποστήριξης, καθώς δεν υπάρχουν επί του παρόντος αποτελεσματικές φαρμακολογικές επιλογές. Ωστόσο, το MCI αναφέρεται σε πολλαπλές συνθήκες που διαφέρουν ως προς την παρουσίαση και είναι ελάχιστα κατανοητές, περιπλέκοντας τις προσπάθειες παροχής λύσεων. Αυτή η ανασκόπηση του πεδίου εφαρμογής σχεδιάστηκε με σκοπό τον εντοπισμό μόνο τεχνολογίας που σχετίζεται ειδικά με το MCI, αλλά οι τεχνολογίες που προσδιορίζονται είναι ευρύτερα χρήσιμες για την αντιμετώπιση προκλήσεων που σχετίζονται με τη γήρανση και για άτομα με άλλες γνωστικές αναπηρίες, όπως η φυσιολογική γνωστική γήρανση και για προηγούμενα στάδια άνοια. Έτσι, παρουσιάζουμε τα ευρήματά μας ως εξής: τεχνολογίες AAL που αναπτύχθηκαν ως υποστήριξη για προκλήσεις στην καθημερινή ζωή που αντιμετωπίζουν ηλικιωμένοι που ζουν με γνωστικές αναπηρίες (με επίκεντρο το MCI), καθώς και τεχνολογία AAL που μπορεί να είναι γενικότερα χρήσιμη σε αυτήν την ομάδα χρηστών.

Οι γνωστικές διαταραχές, γενικά, είναι γνωστό ότι επηρεάζουν την ικανότητα ενός ατόμου να παρατηρεί, να κωδικοποιεί, να συσσωρεύει, να ανακτά και να χρησιμοποιεί πληροφορίες. Διάφορα μονοπάτια μπορεί να οδηγήσουν σε γνωστική εξασθένηση. Για παράδειγμα, υπάρχουν νευροεκφυλιστικές καταστάσεις που μπορούν να επηρεάσουν τη γνωστική έκπτωση, οι οποίες περιλαμβάνουν τη νόσο του Αλτσχάιμερ, τη νόσο του Pick, τη νόσο του Πάρκινσον, τη νόσο του σώματος Lewy, τη νόσο του Huntington, την προοδευτική υπερπυρηνική παράλυση και τον εκφυλισμό της παρεγκεφαλίδας. Επιπλέον, αγγειακές διαταραχές όπως τα εγκεφαλικά επεισόδια και η εγκεφαλική εμβολική νόσος μπορούν επίσης να οδηγήσουν σε ελλείμματα στις γνωστικές διεργασίες. Τα άτομα που ζουν με κάποια μορφή γνωστικής εξασθένησης μπορεί να αντιμετωπίσουν μια σειρά από συμπτωματικές προκλήσεις που σχετίζονται με απώλεια μνήμης, ελλείψεις στη λεκτική επικοινωνία, λειτουργική έκπτωση, καθώς και άλλα ποικίλα ελλείμματα προσοχής, κρίσης και ενόρασης. Υπάρχουν επίσης ζητήματα συμπεριφοράς όπως η κατάθλιψη, η διέγερση ή η ψύχωση που συνδέονται με τη γνωστική εξασθένηση, η οποία μπορεί να επηρεάσει περαιτέρω την ευημερία και τη συνολική ποιότητα ζωής. Τα συμπτώματα της γνωστικής έκπτωσης μπορεί να επηρεάσουν την προσωπική ασφάλεια και το αίσθημα ασφάλειας, ανεξαρτησίας και αυτοπεποίθησης ενός ατόμου, τα οποία επηρεάζουν άλλους τομείς της ζωής, όπως η ικανότητα συμμετοχής και συμμετοχής σε κοινωνικές δραστηριότητες καθώς και η

ικανότητα να ζει ελεύθερα. Για να καταστεί δυνατή η ανεξάρτητη διαβίωση και να βοηθηθεί η βελτίωση της ποιότητας ζωής για άτομα που ζουν με γνωστική εξασθένηση, διάφορες τεχνολογικές παρεμβάσεις έχουν εμφανιστεί τις τελευταίες δεκαετίες για την υποστήριξη ατόμων με γνωστική εξασθένηση, συμπεριλαμβανομένων των περιβαλλοντικών προσαρμογών «χαμηλής τεχνολογίας», τεχνολογιών για την υγεία διαχείρισης, καθώς και για βοήθεια σε θέματα κινητικότητας και διαχείριση περιπλανώμενων συμπεριφορών. Πιο συγκεκριμένα, υπάρχουν δυνατότητες που προέρχονται από την τεχνολογία AAL που μπορούν να βελτιώσουν τη ζωή όσων ζουν με γνωστική έκπτωση. Υπάρχουν τρεις γενιές τεχνολογίας που έχουν σχεδιαστεί για την υποστήριξη της ανεξάρτητης διαβίωσης των ηλικιωμένων, τις οποίες περιγράφουμε εδώ. Μια περίληψη αυτών των τριών γενεών παρουσιάζεται στην εικόνα 2 παρακάτω. Αυτές οι τεχνολογίες αναφέρονται ως «τηλεκαθαρίσεις» με βάση ένα μοντέλο του HB. Ωστόσο, αυτό το μοντέλο παρέχει επίσης πληροφορίες σχετικά με την εξέλιξη των τεχνολογιών AAL εκτός του Ηνωμένου Βασιλείου.



Εικόνα 2. Οι τρεις γενιές του AAL

Πρώτη Γενιά AAL

Ο μεγαλύτερος ενήλικας πιέζει το κουμπί ή το μενταγιόν για να ενεργοποιήσει το συναγερμό σε περίπτωση έκτακτης ανάγκης, όπως πτώση. Μόλις ειδοποιηθεί, ένα άτομο του προσωπικού σε ένα τηλεφωνικό κέντρο 24 ωρών επικοινωνεί με το ηλικιωμένο άτομο και καθορίζει εάν χρειάζονται ή όχι άτυποι φροντιστές ή υπηρεσίες έκτακτης ανάγκης. Ένα πολύ γνωστό παράδειγμα αυτής της τεχνολογίας πρώτης γενιάς είναι το LifeCall (πρόσφατα μετονομάστηκε LifeAlert), του οποίου το αξέχαστο "Help, I've fallen and I can't up to up!" διαφημιστικό από το 1992 ευαισθητοποίησε το κοινό σε αυτές τις συσκευές.

Προηγούμενες έρευνες δείχνουν ότι αυτοί οι συναγερμοί πρώτης γενιάς έχουν πολλά οφέλη που σχετίζονται με την ασφάλεια και την ασφάλεια των ηλικιωμένων που κατοικούν στην κοινότητα. Αυτά τα οφέλη περιλαμβάνουν μειωμένα επίπεδα άγχους μεταξύ των ηλικιωμένων, των οικογενειών και των φροντιστών. μειωμένες εισαγωγές στο νοσοκομείο· πρόωρη εξιτήριο από το νοσοκομείο· και καθυστερημένη είσοδος σε εγκαταστάσεις μακροχρόνιας φροντίδας . Αν και είναι ευεργετική σε ορισμένες περιπτώσεις, η έρευνα καταδεικνύει συγκεκριμένες αδυναμίες που σχετίζονται με αυτήν την τεχνολογία.

Δεύτερη Γενιά AAL

Αυτές οι γενιές τεχνολογιών χαρακτηρίζονται από την ενσωμάτωση ηλεκτρονικών εξαρτημάτων, τα οποία βοήθησαν στην αντιμετώπιση των περιορισμών της πρώτης γενιάς. Αυτές οι τεχνολογίες δεν ανταποκρίνονται απλώς. ανιχνεύουν επίσης πιθανές καταστάσεις έκτακτης ανάγκης, όπως πτώση ή περιβαλλοντικούς κινδύνους, χρησιμοποιώντας αισθητήρες. Μέσα στο σπίτι, οι ηλικιωμένοι παρακολουθούνται χρησιμοποιώντας αισθητήρες, οι οποίοι ζητούν βοήθεια χωρίς να βασίζονται στον χρήστη για την ενεργοποίηση του συναγερμού. Για παράδειγμα, εάν υπάρχει διαρροή αερίου μέσα στο σπίτι αφήνοντας τον ηλικιωμένο ενήλικα ανίκανο, άγνοιο ή/και ανίκανο να αναζητήσει βοήθεια, το σύστημα παρακολούθησης του αισθητήρα ενεργοποιεί αυτόματα τον συναγερμό και επικοινωνεί με τις αρμόδιες αρχές. Αυτά τα συστήματα αισθητήρων δεύτερης γενιάς μπορεί να αποδειχθούν ωφέλιμα για ενήλικες μεγαλύτερης ηλικίας με MCI, καθώς αυτά τα άτομα παρουσιάζουν δυσκολίες στη χρήση οικιακών συσκευών. Συχνά, αυτοί οι ηλικιωμένοι ξεχνούν να σβήσουν συσκευές, όπως τη σόμπα ή το σίδερο, αφού ολοκληρώσουν μια εργασία. Οι τεχνολογίες δεύτερης γενιάς μπορεί να αντιληφθούν ότι η σόμπα παραμένει αναμμένη για πολύ καιρό και να ειδοποιήσουν το άτομο πριν γίνει επικίνδυνη. Παρά τα πιθανά οφέλη, μια αδυναμία που σχετίζεται με αυτήν τη γενιά είναι το γεγονός ότι ορισμένοι χρήστες θεωρούν ότι είναι παρεμβατική. Αυτές οι τεχνολογίες εμφανίζονται στην αγορά και χρησιμοποιούνται από ενήλικες μεγαλύτερης ηλικίας μέσα στο σπίτι.

Τρίτη Γενιά AAL

Το πιο πρόσφατο κύμα τεχνολογίας AAL εμφανίστηκε μαζί με την πρόοδο των ΤΠΕ. Σε αυτό το στάδιο, η τεχνολογία όχι μόνο εντοπίζει και αναφέρει προβλήματα. αποτρέπει προβλήματα χρησιμοποιώντας την έννοια του AAL. Τα συστήματα AAL που «παρέχουν τις υπηρεσίες τους με ευαίσθητο και αποκριτικό τρόπο και ενσωματώνονται διακριτικά στο καθημερινό μας περιβάλλον αναφέρονται ως ευφυή περιβάλλοντος». Ένα παράδειγμα περιλαμβάνει συστήματα οικιακής παρακολούθησης που χρησιμοποιούν μη παρεμβατικές μεθόδους, οι οποίες αναιρούν την ανάγκη χειροκίνητης ενεργοποίησης για συναγερμούς, ενώ μειώνουν την εξάρτηση από την ενεργή επίβλεψη. Αυτά τα συστήματα ενσωματώνουν υπολογιστικά

συστήματα και βοηθητικές συσκευές σε καθημερινά περιβάλλοντα ζωής, προκειμένου όχι μόνο να παρακολουθούν το οικιακό περιβάλλον αλλά και να παρακολουθούν το ηλικιωμένο άτομο. Περιβαλλοντικοί και φορητοί αισθητήρες παρακολουθούν τα ζωτικά σημάδια καθώς και τις αλλαγές στην κινητικότητα και τα μοτίβα δραστηριότητας. Αυτά μπορεί να είναι ενδεικτικά αλλαγών στην κατάσταση της υγείας.

Οι ενεργοποιητές, οι οποίοι είναι μηχανικές συσκευές που χρησιμοποιούνται για τον έλεγχο ενός μηχανισμού ή συστήματος, παρέχουν βοήθεια στο ηλικιωμένο άτομο, ενώ οι έξυπνες διεπαφές παρέχουν πληροφορίες, υποστήριξη και ενθάρρυνση. Επί του παρόντος, αυτή η τρίτη γενιά βρίσκεται σε ανάπτυξη. Ωστόσο, έξυπνα συστήματα και απομακρυσμένες υπηρεσίες υιοθετούνται όλο και περισσότερο σε αυτόν τον τομέα για τη βελτίωση της ποιότητας ζωής, την υποστήριξη της αυτοανεξαρτησίας και τη μείωση του κόστους. Ένα πιθανό όφελος αυτής της τεχνολογίας είναι η μείωση του στίγματος που σχετίζεται με τις συσκευές παρακολούθησης και βοήθειας με την ενσωμάτωση της τεχνολογίας αόρατα σε καθημερινά αντικείμενα. Τα άτομα με MCI μπορεί ήδη να αισθάνονται αυτοσυνείδητα σχετικά με τη μεταβαλλόμενη γνωστική τους ικανότητα και να μην θέλουν την πρόσθετη αμηχανία των προφανών βοηθητικών συσκευών. Αυτή η πρόσθετη αμηχανία μπορεί να έχει ως αποτέλεσμα οι ηλικιωμένοι να αρνούνται να χρησιμοποιήσουν τις τεχνολογίες. Ως εκ τούτου, η διακριτική φύση των τεχνολογιών AAL μπορεί να αφαιρέσει αυτό το στίγμα ή την αμηχανία και έτσι να αυξήσει τη χρήση.

Κεφάλαιο 2 - Διασύνδεση & Συλλογή Μετρήσεων

Η υλοποίηση της διασύνδεσης των αισθητήρων γίνεται με την βοήθεια της προγραμματιστικής γλώσσας Python, η οποία αποτελεί μια υψηλού επιπέδου γλώσσα που σκοπός της είναι να απλοποιήσει την συγγραφή προγραμμάτων που λύνουν κάποιο πρόβλημα. Ιδιαιτερότητά της αποτελεί η έλλειψη συμβόλων και η υποχρεωτική εσοχή (indentation) που στόχο έχει να αυτοματοποιήσει και να επιβάλλει στον χρήστη μια συνετή και οργανωτική δομή του κώδικα. Ο Guido Van Rossum δημοσίευσε την πρώτη έκδοση του κώδικα της Python (έκδοση 0.9.0) στο alt.sources τον Φεβρουάριο του 1991. Αυτή η έκδοση περιλάμβανε ήδη το χειρισμό εξαιρέσεων, συναρτήσεις και τους βασικούς τύπους δεδομένων list, dict, str και άλλους. Ήταν επίσης αντικειμενοστραφής και διέθετε ένα σύστημα ενοτήτων. Η έκδοση 1.0 της Python κυκλοφόρησε τον Ιανουάριο του 1994. Τα σημαντικότερα νέα χαρακτηριστικά που περιλάμβανε αυτή η έκδοση ήταν τα εργαλεία λειτουργικού προγραμματισμού lambda, map, filter και reduce, τα οποία δεν άρεσαν ποτέ στον Guido Van Rossum.

Εξήμισι χρόνια αργότερα, τον Οκτώβριο του 2000, παρουσιάστηκε η Python 2.0. Αυτή η έκδοση περιλάμβανε την κατανόηση λίστας, έναν πλήρη συλλέκτη σκουπιδιών και υποστήριζε unicode. Η Python άκμασε για άλλα 8 χρόνια στις εκδόσεις 2.x πριν κυκλοφορήσει η επόμενη μεγάλη έκδοση ως Python 3.0 (επίσης γνωστή ως "Python 3000" και "Py3K"). Η Python 3 δεν είναι συμβατή προς τα πίσω με την Python 2.x. Η έμφαση στην Python 3 είχε δοθεί στην αφαίρεση των διπλών προγραμματιστικών δομών και ενοτήτων, εκπληρώνοντας έτσι ή πλησιάζοντας στην εκπλήρωση του 13ου νόμου του Zen της Python: "Θα πρέπει να υπάρχει ένας - και κατά προτίμηση μόνο ένας - προφανής τρόπος για να το κάνεις". [1]

Για να επιτευχθεί ο στόχος αυτής της πτυχιακής, εκτός από την ορθή χρήση που πρέπει να γίνει στη τεχνολογία που προαναφέρθηκε, πρέπει να περιγράψουμε αναλυτικά τις επιμέρους διεργασίες που εμπεριέχονται στη διαδικασία, από την διασύνδεση με τους αισθητήρες και την αποστολή εντολών στην Alexandra έως και την ανάλυση των δεδομένων από αλγορίθμους που εφαρμόζονται στην μηχανική μάθηση. Οι κύριες διεργασίες αναλύονται παρακάτω:

- Κύρια δομή προγράμματος και δημιουργία αρχείου csv για την εγγραφή των δεδομένων που συλλέγονται
- Εξουσιοδότηση χρήστη για εξωτερικό και εσωτερικό API και κατανάλωση
- Επεξεργασία και αποθήκευση δεδομένων
- Εφαρμογή πρωτοκόλλου επικοινωνίας για την Alexandra
- Φόρτωση και αποστολή εντολών προς κατανάλωση

Το πρώτο βήμα αφορά την δημιουργία της κύριας δομής του προγράμματος. Στην συγκεκριμένη υλοποίηση θα γίνει η χρήση του αντικειμενοστραφούς μοντέλου για την καλύτερη επέκταση, συντήρηση και κατανόηση της λογικής. Το αντικειμενοστραφές μοντέλο αποτελείται από την δημιουργία μιας κύριας ή πολλών διαφορετικών κλάσεων οι οποίες παρέχουν τις οδηγίες για την δημιουργία αντικειμένων. Ανάλογα με τον σκοπό του κάθε αντικειμένου που δημιουργείται, η κάθε κλάση μπορεί να φέρει στοιχεία από άλλες κλάσεις, αν το απαιτεί η λογική, δημιουργώντας μια σχέση γονέα - παιδιού ή αδελφών. Στη συγκεκριμένη υλοποίηση δεν υπάρχει τέτοια σχέση καθώς το αντικείμενο που δημιουργείται, έχει ευθύνη να εκτελέσει όλες τις επιμέρους εργασίες που χρειάζονται για την πραγματοποίηση του στόχου.

Πιο αναλυτικά, στο επίπεδο του κατασκευαστή της κλάσης, έχουμε την εκτέλεση όλων εκείνων των υποστηρικτικών μεθόδων, που στόχο έχουν να συμβάλλουν στην συλλογική λογική της υλοποίησης, είτε με την εκτέλεση της μεθόδου εξουσιοδότησης είτε με την δημιουργία αρχείων και την καταγραφή των δεδομένων. Ο κατασκευαστής σκοπό έχει να κατευθύνει την ροή του προγράμματος με την κλήση μεθόδων και των ορισμό ιδιοτήτων. Η πρώτη μέθοδος που καλείται, στόχο έχει τον έλεγχο ύπαρξης και δημιουργίας του αρχείου καταγραφής σε τύπο csv (comma separated values). Η ιδιαιτερότητα αυτού του τύπου αρχείων είναι ότι

συνήθως είναι ο τύπος που παράγεται από διάφορες μηχανές όταν κάνουμε εξαγωγές μετρήσεων από αυτές και η μοναδική δομή του το καθιστά εύκολο και βολικό να μπορούμε να το διαβάσουμε και να το επεξεργαστούμε ακόμα και χωρίς την ύπαρξη κάποιας βιβλιοθήκης που στόχο έχει να παραμετροποιεί csv.

Η δεύτερη μέθοδος αφορά την λειτουργία της εξουσιοδότησης των δύο τύπων API (Application Programming Interfaces) που χρησιμοποιούνται στην υλοποίηση, το εσωτερικό που είναι άμεσα συνδεδεμένο με το τμήμα και τα δύο εξωτερικά που αφορούν απομακρυσμένους πόρους εκτός του τμήματος. Πριν αναφερθούμε αναλυτικά στις ενέργειες των δύο τύπων API που καταναλώνουμε, είναι συνετό να δώσουμε έναν ορισμό για το API. Ένα API, αποτελεί μια υπηρεσία η οποία στόχο έχει να σερβίρει ορισμένους πόρους που διαθέτει μέσα από ένα μέσο το οποίο είναι το πρωτόκολλο HTTP. Πιο συγκεκριμένα, το API είναι ένα σύνολο καθορισμένων κανόνων που εξηγούν πώς οι υπολογιστές ή οι εφαρμογές επικοινωνούν μεταξύ τους. Τα API βρίσκονται μεταξύ μιας εφαρμογής και του διακομιστή web, λειτουργώντας ως ενδιάμεσο επίπεδο που επεξεργάζεται τη μεταφορά δεδομένων μεταξύ συστημάτων. Η επικοινωνία μεταξύ πελάτη και API περιγράφεται παρακάτω:

1. Μια εφαρμογή πελάτη ξεκινά μια κλήση API για την ανάκτηση πληροφοριών γνωστή και ως αίτημα ή request. Αυτό το αίτημα υποβάλλεται σε επεξεργασία από μια εφαρμογή στον διακομιστή ιστού μέσω του Uniform Resource Identifier (URI) του API και περιλαμβάνει ένα ρήμα αιτήματος, κεφαλίδες και μερικές φορές ένα σώμα αιτήματος.
2. Μετά τη λήψη ενός έγκυρου αιτήματος, το API πραγματοποιεί μια κλήση στο εξωτερικό πρόγραμμα ή διακομιστή ιστού.
3. Ο διακομιστής στέλνει μια απάντηση στο API με τις ζητούμενες πληροφορίες.
4. Το API μεταφέρει τα δεδομένα στην αρχική αίτηση αίτησης.

Ενώ η μεταφορά δεδομένων θα διαφέρει ανάλογα με την υπηρεσία web που χρησιμοποιείται, αυτή η διαδικασία αιτημάτων και απόκρισης γίνεται μέσω ενός API. Ενώ μια διεπαφή χρήστη έχει σχεδιαστεί για χρήση από ανθρώπους, τα API έχουν σχεδιαστεί για χρήση από υπολογιστή ή εφαρμογή. Τα API προσφέρουν ασφάλεια από το σχεδιασμό, επειδή η θέση τους ως μεσάζων διευκολύνει την αφαίρεση της λειτουργικότητας μεταξύ δύο συστημάτων - το τελικό σημείο API αποσυνδέει την καταναλωτική εφαρμογή από την υποδομή που παρέχει την υπηρεσία. Οι κλήσεις API περιλαμβάνουν συνήθως διαπιστευτήρια εξουσιοδότησης για τη μείωση του κινδύνου επιθέσεων στον διακομιστή και μια πύλη API μπορεί να περιορίσει την πρόσβαση για να ελαχιστοποιήσει τις απειλές ασφαλείας. Επίσης, κατά τη διάρκεια της ανταλλαγής, οι κεφαλίδες HTTP, τα cookie ή οι παράμετροι συμβολοσειράς ερωτήματος παρέχουν πρόσθετα επίπεδα ασφαλείας στα δεδομένα. Γιατί όμως χρειαζόμαστε τα API; Οι λόγοι αναγράφονται παρακάτω:

1. Βελτιωμένη συνεργασία: Τα API επιτρέπουν την ενοποίηση, έτσι ώστε αυτές οι πλατφόρμες και οι εφαρμογές να μπορούν να επικοινωνούν απρόσκοπτα

μεταξύ τους. Μέσω αυτής της ενοποίησης, οι εταιρείες μπορούν να αυτοματοποιήσουν τις ροές εργασίας και να βελτιώσουν τη συνεργασία στο χώρο εργασίας. Χωρίς API, πολλές επιχειρήσεις θα στερούνταν συνδεσιμότητας και θα υπέφεραν από ενημερωτικά σιλό που θέτουν σε κίνδυνο την παραγωγικότητα και την απόδοση.

2. Ευκολότερη καινοτομία: Τα API προσφέρουν ευελιξία, επιτρέποντας στις εταιρείες να κάνουν συνδέσεις με νέους επιχειρηματικούς εταίρους, να προσφέρουν νέες υπηρεσίες στην υπάρχουσα αγορά τους και, τελικά, να έχουν πρόσβαση σε νέες αγορές που μπορούν να αποφέρουν τεράστιες αποδόσεις και να οδηγήσουν στον ψηφιακό μετασχηματισμό. Για παράδειγμα, η εταιρεία Stripe ξεκίνησε ως API με μόλις επτά γραμμές κώδικα.
3. Δημιουργία εσόδων δεδομένων: Πολλές εταιρείες επιλέγουν να προσφέρουν δωρεάν API, τουλάχιστον αρχικά, ώστε να μπορούν να δημιουργήσουν ένα κοινό προγραμματιστών γύρω από την επωνυμία τους και να σφυρηλατήσουν σχέσεις με πιθανούς επιχειρηματικούς συνεργάτες. Ωστόσο, εάν το API παραχωρήσει πρόσβαση σε πολύτιμα ψηφιακά στοιχεία, μπορείτε να δημιουργήσετε έσοδα από αυτό πουλώντας πρόσβαση (αυτό αναφέρεται ως οικονομία API).
4. Προστιθέμενη ασφάλεια: Όπως σημειώθηκε παραπάνω, τα API δημιουργούν ένα πρόσθετο επίπεδο προστασίας μεταξύ των δεδομένων σας και ενός διακομιστή. Οι προγραμματιστές μπορούν να ενισχύσουν περαιτέρω την ασφάλεια API χρησιμοποιώντας διακριτικά, υπογραφές και κρυπτογράφηση Transport Layer Security (TLS). με την εφαρμογή πυλών API για τη διαχείριση και τον έλεγχο ταυτότητας της κυκλοφορίας· και με την πρακτική αποτελεσματική διαχείριση API.

Η ιδέα του API δεν είναι νέα. Πιθανότατα εμφανίστηκε για πρώτη φορά το 1968 και ορίστηκε ως «μια συλλογή από ρουτίνες κώδικα που παρέχει σε εξωτερικούς χρήστες λειτουργίες δεδομένων και δεδομένων» (Cotton and Greatorex, 1968). Επειδή τα API είναι γενικές τεχνολογικές λύσεις, μπορούν να χρησιμοποιηθούν για πολλούς σκοπούς. Έτσι, μπορούμε να υιοθετήσουμε μια πιο πρόσφατη και εκτεταμένη εξήγηση των API που τα ορίζει ως «τις κλήσεις, οι υπορουτίνες ή οι διακοπές λογισμικού που περιλαμβάνουν ένα τεκμηριωμένη διεπαφή έτσι ώστε ένα πρόγραμμα εφαρμογής να μπορεί να χρησιμοποιεί τις υπηρεσίες και τις λειτουργίες ενός άλλου εφαρμογή, λειτουργικό σύστημα, λειτουργικό σύστημα δικτύου, πρόγραμμα οδήγησης ή άλλο πρόγραμμα λογισμικού χαμηλότερου επιπέδου» (Shnier, 1996).

Από την άποψη της μηχανικής λογισμικού, τα API αποτελούν τις διεπαφές των διαφόρων δομικών στοιχείων που ο προγραμματιστής μπορεί να συναρμολογήσει για να δημιουργήσει μια εφαρμογή. Ένας προγραμματιστής εφαρμογών χρησιμοποιεί API για τη δημιουργία μιας εφαρμογής συνδυάζοντας διάφορες διαθέσιμες βιβλιοθήκες λογισμικού για την επίτευξη ενός συγκεκριμένου στόχου. Ενώ η έννοια του

προγραμματικού διεπαφές ως μια συλλογή μεθόδων που εξάγονται από μια συγκεκριμένη βιβλιοθήκη κώδικα δεν είναι νέα, με την εμφάνιση του web, και ειδικότερα στο Web 2.0, εισήχθη η έννοια των web API για να υποδείξει αυτά τα API που λειτουργούν σε ο ιστός. Τα Web API χρησιμοποιούνται για να παρέχουν στους προγραμματιστές τα δομικά στοιχεία που απαιτούνται για τη δημιουργία διαδικτυακού 6 εφαρμογές λογισμικού. Καθώς μας ενδιαφέρουν ιδιαίτερα τα web API, στο υπόλοιπο αυτού του εγγράφου, εκτός αν ειδάλως καθορίζεται, ο όρος «API» θα χρησιμοποιείται για να αναφέρεται σε web API. [2]

Το πρώτο εξωτερικό API που χρησιμοποιείται στην εφαρμογή είναι σχετικό με την εξαγωγή στοιχείων τοποθεσίας από την IP του χρήστη και πιο συγκεκριμένα τις συντεταγμένες του και ονομάζεται ip-api.com. Ο λόγος που εξάγεται η τοποθεσία, είναι για να χρησιμοποιηθεί ως παράμετρος στην κατανάλωση του δεύτερου εξωτερικού API. Επειδή ο σένσορας που αντλούνται οι μετρήσεις αφορά μετρήσεις θερμοκρασίας, είναι κατανοητό ότι το δεύτερο API αντλεί καιρικές πληροφορίες βάσει τοποθεσίας. Η χρήση των συντεταγμένων απλώς επιτυγχάνουν πιο ακριβή εξαγωγή πληροφορίας.

Ενώ το πρώτο εξωτερικό API κατά την λειτουργία του αιτήματος δεν συμπεριλαμβάνει καθόλου παραμέτρους, το δεύτερο προαπαιτεί τουλάχιστον δύο για να λειτουργήσει όπως προβλέπεται. Είναι σημαντικό να σημειωθεί ότι η υπηρεσία ip-api.com όπως και η api.openweathermap.org, υπεύθυνη για τις καιρικές πληροφορίες δεν απαιτούν καθόλου εξουσιοδότηση από την εφαρμογή του πελάτη (client app) οπότε πρακτικά οποιοσδήποτε μπορεί να χρησιμοποιήσει την ip του για να καταγράψει τις συντεταγμένες της τοποθεσίας του, να τις συμπεριλάβει στο URI της δεύτερης και να εξάγει τις καιρικές μετρήσεις που αφορούν το τόπο που βρίσκεται. Η εξουσιοδότηση για το εσωτερικό API, δηλαδή της υπηρεσίας του τμήματος που βρίσκεται στην πλατφόρμα ATLAS που αναφέρεται στο δίκτυο αισθητήρων του Εργαστηρίου Σχεδιασμού Ενσωματωμένων Συστημάτων και Εφαρμογών, απαιτεί πιστοποίηση με κάθε αποστολή αιτήματος. Πιο συγκεκριμένα, για να είναι εφικτή η εξαγωγή κάποιας απάντησης από το API, θα πρέπει στις κεφαλίδες του αιτήματος να συμπεριλάβουμε ορισμένες παραμέτρους σε μορφή JSON ή (JavaScript Object Notation). Το JSON είναι μια μορφή αντικειμένου η οποία ερμηνεύεται αρχικά ως ένας γράφος. Παρακάτω παρουσιάζεται ένα αντικείμενο:

```
obj = {
  parent: {
    child_one: { value: any },
    child_two: { value: any },
    child_three: {
      value: any,
      timestamp: "2020-02-22"
    }
  }
}
```

Όπως προαναφέρθηκε, εξετάζοντας το παραπάνω αντικείμενο με την βοήθεια ενός γράφουμε, μπορούμε να εντοπίσουμε την ρύζα ή την αφετηρία του στο πρώτο υπο-αντικείμενο ονόματι **parent**. Το αντικείμενο αυτό, εμπεριέχει τρία υπο-αντικείμενα ή παιδιά του, τα **child_one**, **child_two**, **child_three**. Με τη σειρά τους τα παιδιά έχουν παραμέτρους -στη προκειμένη any δηλαδή οποιουδήποτε τύπου- ή κάποια χρονο ετικέτα. Κάθε πληροφορία που χρειάζεται να εξάγουμε από ένα αντικείμενο θα πρέπει να γίνεται με την λογική των κλειδιών. Πιο αναλυτικά, στην περίπτωση που θα θέλαμε να πάρουμε την χρονο ετικέτα από το παιδί τρία, θα πρέπει να αναφερθούμε κλιμακωτά και προς τα εντός από την ρύζα. Παράδειγμα:

```
obj.parent.child_three.timestamp
```

- θα μας δώσει το "2020-02-22"

Στην Python, η μοντελοποίηση και ο χειρισμός των αντικειμένων JSON, γίνεται με την χρήση λεξικών (dictionaries) που επιτρέπουν στην χρήση αναφοράς κλειδιών για την εξαγωγή των αντίστοιχων τιμών (λεγόμενα key - value pairs).

Με την εξαγωγή των μετρήσεων θερμοκρασίας και πίεσης από τους αισθητήρες του έξυπνου σπιτιού, η διαδικασία συνεχίζεται με την μετατροπή των τιμών στην ιδανική μορφή και έπειτα εγγραφή αυτών σε ένα αρχείο csv. Τα αρχεία csv αποτελούνται από σειρές εγγραφών, στις οποίες οι στήλες είναι διαχωρισμένες με κόμματα. Η συγκεκριμένη μορφή είναι ευέλικτη στην ανάγνωση και εγγραφή δεδομένων από χρήστες και μηχανές. Η πρώτη γραμμή του csv αρχείου, συχνά περιγράφει τις στήλες, οι οποίες μορφοποιούνται αργότερα ως τα κλειδιά των πινάκων ή λιστών. Πιο αναλυτικά, δημιουργείται ένα λεξικό (dictionary) το οποίο εμπεριέχει κλειδιά τα οποία έχουν ως τιμή τους μονοδιάστατους πίνακες με δεδομένα ίδιου τύπου (τα οποία είναι η εκάστοτε γραμμές για την στήλη που εξετάζεται). Με αυτόν το τρόπο, δημιουργείται μια παράλληλη σχέση των πινάκων, οι οποία είναι χρήσιμη και αποδοτική τόσο στην προσπέλαση όσο και στην αντιστοίχιση με γειτονικές στήλες. Για παράδειγμα: Έστω ότι έχει πραγματοποιηθεί εξαγωγή τριών πινάκων εκατό εγγραφών.

Οι πίνακες ονομάζονται A, B, Γ.

- Στην περίπτωση που θέλουμε να ψάξουμε για μια τιμή στον πίνακα A, μπορούμε να χρησιμοποιήσουμε συνθήκες ελέγχου στους γειτονικούς πίνακες B και Γ ώστε να απορρίψουμε ή να στοχεύσουμε πιο έξυπνα το σύνολο αναζήτησης του A.
- Αν έχουμε τιμή από τον πίνακα A και θέλουμε να βρούμε τις αντίστοιχες τιμές στους γειτονικούς πίνακες, μένει μόνο να χρησιμοποιήσουμε τον δείκτη της θέσης της αρχικής τιμής που βρίσκεται στον A εξοικονομώντας πόρους και χρόνο. Αν στην θέση 0 του A υπάρχει τιμή ενδιαφέροντος, τότε οι αντίστοιχες τιμές ενδιαφέροντος θα είναι στη θέση 0 του B και Γ.

Όταν η καταγραφή των δεδομένων από το εξωτερικό και το εσωτερικό API έχει ολοκληρωθεί επιτυχώς, τα δεδομένα στέλνονται με δημιουργία πρόθεσης (intent) στον εγκέφαλο της αλεξάνδρας όπου θα γίνει η δημιουργία της φράσης η οποία θα δώσει έναυσμα για την εκτέλεση και αποστολή μιας προκαθορισμένης απάντησης, συμπεριλαμβάνοντας την μέση τιμή από τα σύνολα δεδομένων των δύο τύπων APIs. Η επικοινωνία με το chatbot και του προγράμματος (clay) γίνεται μέσω MQTT broker. Το Message Queuing Telemetry Transport (MQTT) είναι ένα πρωτόκολλο ανταλλαγής μηνυμάτων που βασίζεται στη δημοσίευση/συνδρομή, που έχει σχεδιαστεί για περιορισμένες συσκευές και δίκτυα χαμηλού εύρους ζώνης, υψηλής καθυστέρησης και αναξιόπιστα. Οι αρχές σχεδιασμού του πρωτοκόλλου είναι η ελαχιστοποίηση του εύρους ζώνης δικτύου και των πόρων της συσκευής απαιτήσεις, ενώ προσπαθεί επίσης να διασφαλίσει την αξιοπιστία και κάποιο βαθμό διασφάλισης της παράδοσης. Το MQTT λειτουργεί μέσω του πρωτοκόλλου TCP/IP. Χρησιμοποιεί τη θύρα 1883 που έχει εκχωρηθεί από το Διαδίκτυο Αρχή Εκτεθέντων Αριθμών (IANA). Για τη χρήση του MQTT μέσω SSL, χρησιμοποιείται η θύρα 8883. Μερικοί από τους τοπικούς μεσίτες για το MQTT είναι οι Mosquitto, Mosca, HiveMQ, κ.λπ. [3]

Η υλοποίηση του πρωτοκόλλου για την ολοκληρωμένη λειτουργία της αλληλεπίδρασης, γίνεται με την βοήθεια της βιβλιοθήκης raho που παρέχεται δωρεάν στους από την Eclipse Foundation. Η γενική ροή χρήσης έχει ως εξής:

- Δημιουργείται μια παρουσία πελάτη
- Γίνεται σύνδεση σε έναν μεσίτη χρησιμοποιώντας μία από τις συναρτήσεις `connect*()`.
- Καλείται μία από τις συναρτήσεις βρόχου*() για να διατηρηθεί η ροή κυκλοφορίας δικτύου με τον μεσίτη
- Χρησιμοποιείται το `subscribe()` για να γίνει εγγραφή σε ένα θέμα και να πραγματοποιηθεί η λήψη μηνυμάτων
- Χρησιμοποιείται το `public()` για να γίνει δημοσίευση μηνυμάτων στον μεσίτη
- Γίνεται χρήση του `disconnect()` για την αποσύνδεση από τον μεσίτη

Να σημειωθεί ότι θα πρέπει να κληθούν επανακλήσεις για να επιτραπεί στην εφαρμογή να επεξεργάζεται συμβάντα όπως απαιτείται. [4]

Κεφάλαιο 3 - Ενοποίηση Chatbot

Με την ολοκλήρωση συλλογής μετρήσεων και την δημιουργία της επικοινωνίας μεταξύ του αισθητήρα πίεσης και του chat bot, το επόμενο βήμα είναι να αναλύσουμε την τεχνολογία που θα χρησιμοποιηθεί για την διεξαγωγή των ερωτοαπαντήσεων της διεπαφής. Το chat bot “Αλεξάνδρα” έχει αρχικά υλοποιηθεί με την βοήθεια του σκελετού της Καλλιόπης. Η συγκεκριμένη τεχνολογία αποτελεί το μοντέλο διαδικασιών το οποίο παράγει εικονικούς βοηθούς. Ένας εικονικός βοηθός ορίζεται ως

μια εφαρμογή υψηλού επιπέδου που στόχο έχει την απλοποίηση κάποιων καθημερινών και όχι εργασιών του χρήστη. Η παραπάνω εφαρμογή, ανάλογα με την πολυπλοκότητά της, μπορεί να χρησιμοποιήσει μεθόδους εκμάθησης των συνηθειών του χρήστη κατά την γραφή και ομιλία, έχοντας έτσι πιο βελτιωμένη προσέγγιση κάθε φορά που χρήστης εκτελεί ορισμένες εργασίες. Η εκμάθηση μπορεί να γίνεται είτε μέσω άλλων εφαρμογών και ρουτινών μηχανικής μάθησης είτε με πρωτοβουλία του χρήστη, δηλαδή ο χρήστης να διδάσκει την εφαρμογή πως να αντιδρά και με τι μοτίβα να προσέχει. Είναι σημαντικό ωστόσο να θέσουμε το ερώτημα, ποιοί είναι εκείνοι οι παράγοντες που θα κινητοποιήσουν τον χρήστη στο να χρησιμοποιήσει μια τέτοια εφαρμογή; Ο πιο συχνά αναφερόμενος κινητήριος παράγοντας είναι η παραγωγικότητα. Τα chatbots βοηθούν τους χρήστες να αποκτούν έγκαιρα και αποτελεσματικά βοήθεια ή πληροφορίες. Οι χρήστες του Chatbot ανέφεραν επίσης κίνητρα σχετικά ψυχαγωγία, κοινωνικούς και σχεσιακούς παράγοντες και περιέργεια για το τι βλέπουν ως νέο φαινόμενο. [5]

Τα chatbots είναι πράκτορες μηχανών που χρησιμεύουν ως διεπαφές χρήστη φυσικής γλώσσας για δεδομένα και παρόχους υπηρεσιών. Επί του παρόντος, τα chatbot σχεδιάζονται και αναπτύσσονται συνήθως για εφαρμογές ανταλλαγής μηνυμάτων για κινητές συσκευές και πιο συγκεκριμένα σε εφαρμογές κοινωνικής δικτύωσης τόσο με σκοπό την ψυχαγωγία όσο και για την εργασία. Τα chatbots θεωρούνται ως μέσα για άμεση αφοσίωση χρήστη ή πελάτη μέσω μηνυμάτων κειμένου για τον πελάτη. [6]

Στην περίπτωση της τεχνολογίας της Καλλιόπης δίνεται η δυνατότητα να δημιουργηθεί ένα bot στα μέτρα του χρήστη, απλά επιλέγοντας και συνθέτοντας τους υπάρχοντες νευρώνες χωρίς την συγγραφή κώδικα. Τα βασικά στοιχεία του bot είναι τα παρακάτω:

- Ο εγκέφαλος (brain.yml)
- Τα σήματα (signals)
- Οι νευρώνες (neurons)
- Η μνήμη του bot (kalliope_memory)

Ο εγκέφαλος είναι το βασικό συστατικό της Καλλιόπης. Είναι μια ενότητα που δίνει μια διαμόρφωση στις ρουτίνες του προσωπικού βοηθού και, επομένως, καθορίζει τη συμπεριφορά και τις λειτουργίες του. Η Καλλιόπη θα αναζητήσει τα αρχεία του εγκεφάλου με την παρακάτω σειρά:

- Από τον τρέχοντα φάκελο που βρίσκεται το αρχείο yml, π.χ. /home/pi/my_kalliope/brain.yml
- Από το /etc/kalliope/brain.yml
- Από το προεπιλεγμένο brain.yml που βρίσκεται στη ρίζα του δέντρου του έργου.

Ο εγκέφαλος αποτελείται σε επίπεδο προγράμματος όπως αντίστοιχα του ανθρώπινου, αποτελείται από πολυάριθμες συνάψεις. Μια σύναψη είναι ο σύνδεσμος μεταξύ των ενεργειών εισόδου (σημάτων) και εξόδου (νευρώνες). Ένα σήμα είναι ένα συμβάν εισόδου που ενεργοποιείται από μια σύναψη. Όταν γίνει λήψη ενός σήματος από τη λίστα, η Καλλιόπη εκτελεί τους συνδεδεμένους νευρώνες της σύναψης. Το σήμα στη προκειμένη, αποτελεί ένα κλειδί το οποίο στόχο έχει να προσπελάσει μια συγκεκριμένη αντίστοιχη τιμή. Η λογική αυτή εφαρμόζεται σε δομές όπως τα λεξικά, τα αντικείμενα και τις λίστες σε προγραμματιστικό επίπεδο. Για να αρχικοποιήσουμε ένα σήμα, μένει να χρησιμοποιήσουμε τη παρακάτω σύνταξη:

signals:

- signal_name: parameter

ή

signals:

- signal_name:

- parameter_key1: parameter_value1

- parameter_key2: parameter_value2

Ορισμένα σήματα θα στείλουν μια λίστα παραμέτρων σε όλους τους νευρώνες όταν ενεργοποιηθούν. Οι νευρώνες είναι τότε ελεύθεροι να τους χρησιμοποιούν ή όχι. Για παράδειγμα, το σήμα `mqtt_subscriber(signals/mqtt_subscriber.md)` στέλνει μια μεταβλητή που ονομάζεται `"mqtt_subscriber_message"` όταν ενεργοποιείται. Στο παρακάτω παράδειγμα, ο νευρώνας `"μίλησε"` χρησιμοποιεί αυτή τη μεταβλητή για να κάνει την Καλλιόπη να μιλήσει δυνατά μια κατάσταση που έλαβε από τον μεσίτη MQTT.

Ένας νευρώνας είναι μία μεμονωμένη μονάδα ελέγχου που εκτελεί μια συγκεκριμένη ενέργεια. Τον χρησιμοποιούμε για να δημιουργούμε συνάψεις. Αποτελεί αντίστοιχη δομή με τα σήματα με τη μόνη διαφορά ότι αναλαμβάνει να επιστρέψει μια έξοδο σύμφωνα με κάποιο ερέθισμα (σήμα εισόδου). Δεν υπάρχει περιορισμός στον αριθμό νευρώνων που μπορεί να οριστεί, ωστόσο, είναι σημαντικό να σημειωθεί όλοι οι νευρώνες που περιέχονται σε μια λίστα θα, με την λήψη του σήματος, θα εκτελεστούν σειριακά ένας προς έναν. Οι νευρώνες δηλώνονται στο τμήμα νευρώνων μιας σύναψης στο αρχείο του εγκέφαλου σας. Η ενότητα νευρώνες είναι μια λίστα (επειδή ξεκινά με ένα "-") που περιέχει ονόματα μονάδων νευρώνων. Η δήλωση γίνεται παρακάτω:

neurons:

- neuron_1_name

- neuron_2_name

- another_neuron

Ορισμένοι νευρώνες χρειάζονται παραμέτρους που μπορούν να μεταβιβαστούν ως ορίσματα ακολουθώντας την παρακάτω σύνταξη:

neurons:

- neuron_name:
 - parameter1: "value1"
 - parameter2: "value2"

Οι νευρώνες απαιτούν ορισμένες παραμέτρους από τη δήλωση συνάψεων για να λειτουργήσουν. Αυτές οι παράμετροι, που ονομάζονται επίσης ορίσματα, μπορούν να περάσουν στον νευρώνα από:

- τη δήλωση του νευρώνα
- καθολικές μεταβλητές
- η συλληφθείσα παραγγελία
- παράμετρος εξόδου από ένα σήμα
- τη μνήμη της καλλιόπης

Ορισμένοι νευρώνες θα επιστρέψουν μεταβλητές σε ένα λεξικό αξίας. Αυτές οι τιμές μπορούν να χρησιμοποιηθούν για να δημιουργηθεί μια προσαρμοσμένη απάντηση στη Καλλιόπη μέσω ενός προτύπου. Ο στόχος της χρήσης ενός προτύπου είναι να δοθεί ελευθερία στον χρήστη να επιλέξει την ενέργεια που θέλει να κάνει την Καλλιόπη να λέει στη δική του γλώσσα και φράσεις. Ένα πρότυπο είναι ένα αρχείο κειμένου που περιέχει μεταβλητές, οι οποίες αντικαθίστανται με τιμές όταν το πρότυπο αξιολογείται από τη μηχανή προτύπου, και από ετικέτες που ελέγχουν τη λογική του προτύπου. Στο επίπεδο της μνήμης, η Καλλιόπη μπορεί να αποθηκεύσει σε μια βραχυπρόθεσμη μνήμη μια μεταβλητή από μια παραγγελία ή που δημιουργείται από έναν νευρώνα: - παραμέτρους εξόδου από έναν νευρώνα - μεταβλητές παραμέτρους που λαμβάνονται από μια παραγγελία. Παρακάτω η σύνταξη με παραμέτρους εξόδου από νευρώνα:

- name: "synapse-name"

signals:

- order: "my order"

neurons:

- neuron_name:

kalliope_memory:

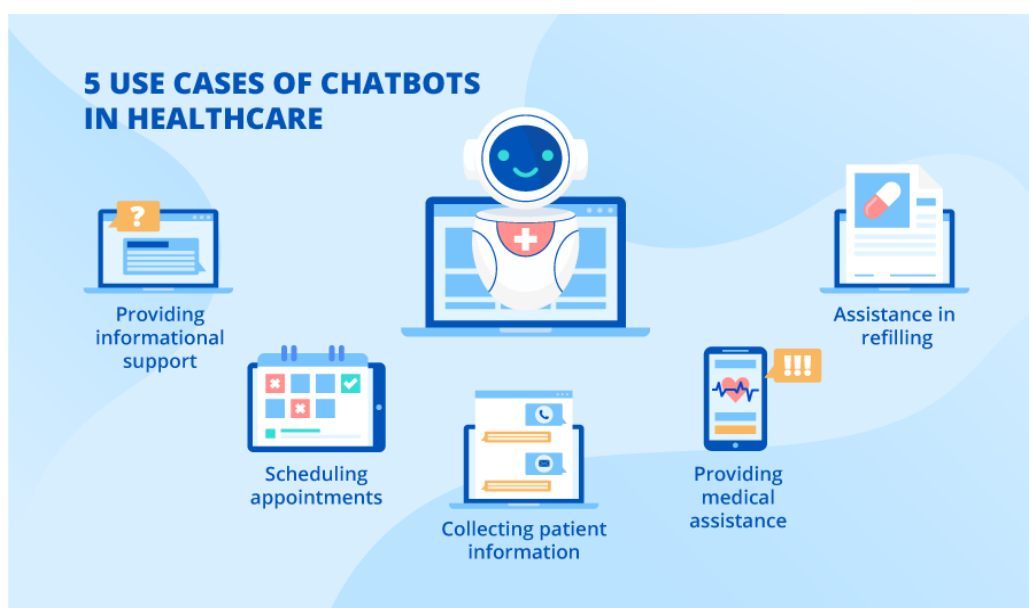
key_name_in_memory: "{{ output_variable_from_neuron }}"

other_key_name_in_memory: "{{ other_output_variable_from_neuron }}"

Σύνταξη για επαναχρησιμοποίηση απομνημονευμένων παραμέτρων σε άλλη σύναψη:

```
- name: "synapse-name"  
signals:  
  - order: "an order"  
neurons:  
  - neuron_name:  
    parameter1: "{{ kalliope_memory['key_name_in_memory'] }}"  
    parameter2: "{{ kalliope_memory['other_key_name_in_memory'] }}"
```

Σήμερα, οι ασθενείς χρειάζονται έναν βολικό τρόπο για να παραμείνουν σε επαφή με τους παρόχους φροντίδας ανά πάσα στιγμή και οι πάροχοι φροντίδας χρειάζονται τρόπους για να εξοικονομήσουν χρόνο για τους γιατρούς και τις νοσοκόμες τους, κάτι που γίνεται ιδιαίτερα σημαντικό εν όψει της πανδημίας COVID-19. Τα chatbots μπορούν να βοηθήσουν και τις δύο πλευρές με τις ανάγκες τους και να αυξήσουν τη συνολική ποιότητα και διαθεσιμότητα της περίθαλψης.



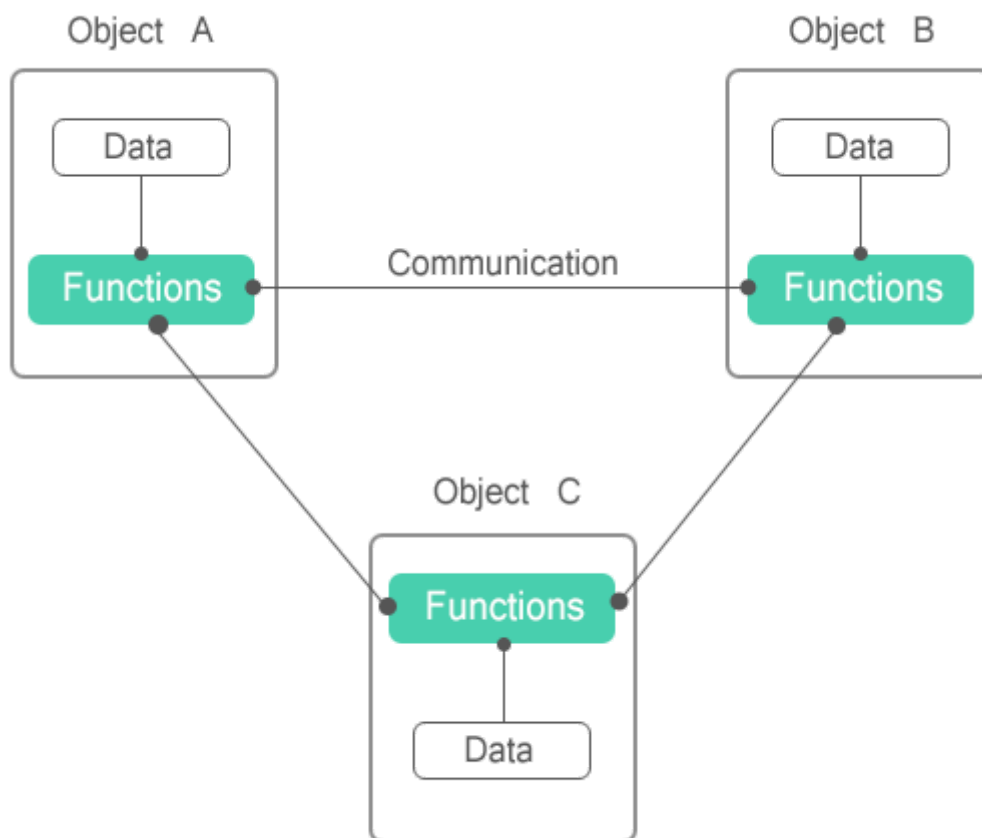
Εικόνα 3. Οι Υπηρεσίες ενός Chatbot

Κεφάλαιο 4 - Ανάλυση Υλοποίησης Κώδικα

Για να επιτευχθεί η κατάλληλη χρήση των παραμέτρων που εξάγονται από τους μετρητές και τα εξωτερικά API που στόχο έχουν την ενοποίηση αυτών στο chatbot της Καλλιόπης, είναι σημαντικό να σημειωθεί ότι πρέπει να γίνει ανάλυση του πηγαίου κώδικα των διαδικασιών. Η κάθε διαδικασία που δημιουργείται για να λύσει ένα συγκεκριμένο πρόβλημα, υλοποιείται ως μεμονωμένη μονάδα (module) η οποία

μπορεί να χρησιμοποιηθεί και από άλλη λογική. Πιο συγκεκριμένα, ο κώδικας που συνοδεύει τις παραπάνω λειτουργίες έχει δημιουργηθεί με την προγραμματιστική αρχιτεκτονική που αναφέρουμε ως Object Oriented Paradigm ή αντικειμενοστρέφια.

Η αντικειμενοστραφής προσέγγιση εστιάζει στα αντικείμενα που απαρτίζουν το σύστημα. Ένα αντικείμενο είναι οτιδήποτε μπορούμε να ονομάσουμε και να περιγράψουμε. Τα αντικείμενα είναι «πράγματα» που θέλουμε να μοντελοποιήσουμε μέσα στα προγράμματά μας και, επομένως, ο στόχος ενός αντικειμενοστρεφούς προγράμματος είναι να αναπαραστήσει τον πραγματικό κόσμο σε κώδικα. Εάν λάβετε υπόψη το παράδειγμα ενός κολεγίου, τα αντικείμενα θα ήταν πράγματα όπως μαθητές, δάσκαλοι, αίθουσες διδασκαλίας και υπολογιστές. Τα αντικείμενα μπορεί επίσης να είναι λιγότερο συγκεκριμένα πράγματα, όπως μαθήματα, δανεισμοί βιβλιοθήκης και καταχωρίσεις εξετάσεων.



Εικόνα 4. Η σύνδεση και επικοινωνία αντικειμένων στην αρχή της αντικειμενοστρέφειας

Ο όρος «αντικείμενο» στον αντικειμενοστραφή προγραμματισμό αντιπροσωπεύει έναν συγκεκριμένο τρόπο οργάνωσης κώδικα και μπορεί να οριστεί, τεχνικά, ως «μια οντότητα που ενσωματώνει τόσο δεδομένα όσο και συμπεριφορά». Αυτό σημαίνει ότι όλα τα δεδομένα και όλες οι διαδικασίες/λειτουργίες ενός συστήματος ανήκουν σε ένα ή περισσότερα αντικείμενα. Τα αντικείμενα στον αντικειμενοστραφή προγραμματισμό

συνεργάζονται μεταξύ τους, στέλνοντας και λαμβάνοντας μηνύματα. Είναι σημαντικό να θυμόμαστε ότι τόσο τα διαδικαστικά όσο και τα αντικειμενοστρεφή παραδείγματα προγραμματισμού υποστηρίζουν διαφορετικούς τρόπους επίλυσης του ίδιου προβλήματος. Στο διαδικαστικό παράδειγμα, το πρόβλημα χωρίζεται σε μικρότερα μέρη που ονομάζονται διαδικασίες (ή συναρτήσεις) και τα συστήματα χτίζονται με συνδυασμό διαδικασιών. Αυτές οι διαδικασίες μοιράζονται δεδομένα περνώντας δεδομένα μεταξύ τους ή μέσω καθολικών μεταβλητών.

Στο αντικειμενοστρεφή παράδειγμα, το πρόβλημα χωρίζεται σε μικρότερα μέρη που ονομάζονται αντικείμενα και τα συστήματα χτίζονται γύρω από αντικείμενα. Τα αντικείμενα είναι αναπαραστάσεις πραγμάτων που υπάρχουν στον πραγματικό κόσμο και τα οποία θέλουμε να μοντελοποιήσουμε σε ένα σύστημα υπολογιστή. Σύμφωνα με τον πηγαίο κώδικα, η λογική αποτελείται από μια κλάση App η οποία υλοποιεί ορισμένες μεθόδους. Κάθε μέθοδος από αυτές είναι κρίσιμη στη συλλογική λειτουργία και δόμηση των τελικών αποτελεσμάτων. Οι μέθοδοι που υλοποιούνται αναγράφονται παρακάτω:

- `get_location`
- `get_openapi_weather`
- `get_esdalab_params`
- `send` (MQTT Client)
- `start` (entry point)

Η `get_location` είναι μια μέθοδος η οποία στόχο έχει να χρησιμοποιήσει την IP του εκάστοτε μηχανήματος με σκοπό να επιστρέψει την τοποθεσία του χρήστη σε μορφή JSON ώστε τελικά να χρησιμοποιηθούν τα δεδομένα που επιστρέφει από άλλη συνάρτηση.

```
def get_location(self):  
    loc_api = 'http://ip-api.com/json/'  
    loc_req = requests.get(url=loc_api)  
    result = loc_req.json()  
    if (result):  
  
        return {'lat':result['lat'], 'lon':result['lon']}  
    return []
```

Αρχικά ορίζεται ο σύνδεσμος - endpoint του API που χρησιμοποιείται και έπειτα με την βοήθεια της requests βιβλιοθήκης -η οποία έρχεται μαζί με την Python- διεξάγεται ένα GET αίτημα στον παραπάνω σύνδεσμο. Αν όλα πάνε καλά και η υπηρεσία παραμένει ενεργή χωρίς προβλήματα, επιστρέφεται μια απάντηση response η οποία

μετατρέπεται σε JSON και δομείται κατάλληλα σε ένα λεξικό με τις παραμέτρους latitude και longitude το οποίο επιστρέφεται στο σημείο που έχει κληθεί η μέθοδος. Η επόμενη συνάρτηση που βρίσκεται στη λογική είναι η `get_openapi_weather` η οποία χρησιμοποιεί τα δεδομένα από την `get_location` ώστε να φέρει τις παραμέτρους των καιρικών συνθηκών της ζητούμενης τοποθεσίας. Πιο αναλυτικά:

```
def get_openapi_weather(self):
    location = self.get_location()
    ow_url =
'http://api.openweathermap.org/data/2.5/weather?lat={}&lon={}&appid={}'.format(
    location['lat'], location['lon'], ow_key())
    ow_req = requests.get(url=ow_url)
    result = ow_req.json()
    date_time = datetime.now().strftime("%m/%d/%Y, %H:%M:%S")
    if (result):
        return {
            'temp': round(result['main']['temp']-273.0, 3),
            'humidity':result['main']['humidity'],
            'recorded':date_time
        }
    return []
```

Στο πρώτο βήμα γίνεται κλήση της συνάρτησης `get_location` η οποία επιστρέφει ένα λεξικό με δύο κλειδιά και τις δύο αντίστοιχες τιμές τους latitude και longitude. Έπειτα αρχικοποιείται ο σύνδεσμος - endpoint του νέου API που καλείται και δομείται κατάλληλα το περιεχόμενό του με την χρήση της `format` η οποία τοποθετεί στοχευμένα τις παραμέτρους πάνω στο σημείο του ερωτήματος χρησιμοποιώντας τα σύμβολα `{ }` στη σειρά που παρατίθενται. Έπειτα αφού δομηθεί σωστά το endpoint - url, γίνεται η κλήση του API περνώντας παράλληλα το κλειδί για τη ταυτοποίηση. Η ταυτοποίηση γίνεται με χρήση κλειδιού το οποίο δημιουργήθηκε με την χρήση ενός άλλου εξωτερικού αρχείου, το οποίο μας το σερβίρει με την χρήση μιας `lambda` συνάρτησης. Πριν αναλύσουμε περαιτέρω, θα πρέπει να δοθεί εξήγηση για το τι είναι ένα endpoint, τι είναι η ταυτοποίηση σε επίπεδο API, ποια είδη ταυτοποίησης υπάρχουν και τέλος τι είναι και πως συντάσσεται μια `lambda` συνάρτηση.

Ένα τελικό σημείο (endpoint) του API είναι ένα σημείο στο οποίο ένα API δηλαδή ο κώδικας που επιτρέπει σε δύο προγράμματα λογισμικού να επικοινωνούν μεταξύ τους, συνδέεται με το πρόγραμμα λογισμικού. Τα API λειτουργούν με την αποστολή αιτημάτων για πληροφορίες από μια εφαρμογή Ιστού ή διακομιστή Ιστού και τη λήψη απάντησης. Με άλλα λόγια, τα τελικά σημεία API είναι η συγκεκριμένη

ψηφιακή τοποθεσία όπου αποστέλλονται αιτήματα για πληροφορίες από ένα πρόγραμμα για την ανάκτηση του ψηφιακού πόρου που υπάρχει εκεί. Τα τελικά σημεία καθορίζουν πού μπορούν τα API να έχουν πρόσβαση σε πόρους και συμβάλλουν στην εγγύηση της σωστής λειτουργίας του ενσωματωμένου λογισμικού. Η απόδοση ενός API εξαρτάται από την ικανότητά του να επικοινωνεί επιτυχώς με τα τελικά σημεία του API.

Τα προγράμματα λογισμικού έχουν συνήθως πολλαπλά τελικά σημεία API. Για παράδειγμα, τα τελικά σημεία του Instagram περιλαμβάνουν ένα που επιτρέπει στις επιχειρήσεις και τους δημιουργούς να μετρούν τις αλληλεπιδράσεις μέσω και προφίλ. Ένα που τους επιτρέπει να εποπτεύουν τα σχόλια και τις απαντήσεις τους και ένα τρίτο που τους επιτρέπει να ανακαλύψουν και να εξερευνήσουν απόψεις και τάσεις μέσα από τις ετικέτες. Τα συστήματα που επικοινωνούν μέσω API είναι ολοκληρωμένα συστήματα. Η μία πλευρά στέλνει τις πληροφορίες στο API και ονομάζεται διακομιστής. Η άλλη πλευρά, ο πελάτης, κάνει τα αιτήματα και χειρίζεται το API. Η πλευρά του διακομιστή που παρέχει τις ζητούμενες πληροφορίες ή πόρους είναι το τελικό σημείο του API. Για να υποβληθεί σε επεξεργασία ένα αποτελεσματικό αίτημα από το τελικό σημείο, ο πελάτης πρέπει να παρέχει έναν ενιαίο εντοπισμό πόρων (URL), μια μέθοδο, μια λίστα κεφαλίδων και ένα σώμα. Οι κεφαλίδες παρέχουν μεταδεδομένα σχετικά με ένα αίτημα και το σώμα διατηρεί τα δεδομένα που αποστέλλονται από τον πελάτη στον διακομιστή.

Τα τελικά σημεία λειτουργούν παράλληλα με τις μεθόδους API. Οι μέθοδοι είναι επιτρεπόμενα αιτήματα που μπορούν να υποβληθούν, όπως GET, DELETE, PATCH ή POST. Οι μέθοδοι -- που συχνά ονομάζονται ρήματα στη σύνταξη επικοινωνιών -- τοποθετούνται συχνά ακριβώς πριν από το καθορισμένο τελικό σημείο σε μια πλήρη διεύθυνση URL.

Παραδείγματα τελικών σημείων API:

Ο κώδικας που χρησιμοποιείται για την υποβολή αιτήματος για μια συγκεκριμένη σελίδα στατιστικών στον ιστότοπο του NBA μπορεί να είναι:

```
GET @ https://stats.nba.com/stats/allstarballotpredictor
```

Σε αυτό το παράδειγμα, το GET είναι η μέθοδος, ενώ το τελικό σημείο είναι το συγκεκριμένο τμήμα της διεύθυνσης web που σημειώνεται ως /stats/allstarballotpredictor.

Εάν, από την άλλη πλευρά, μια εφαρμογή ζητά πληροφορίες από την υπηρεσία DynamoDB της Amazon, το αίτημά της μπορεί να είναι:

```
https://dynamodb.us-west-2.amazonaws.com
```

Εξετάστε περαιτέρω το παράδειγμα του MySpace: Εάν ένας προγραμματιστής του Facebook ήθελε να ζητήσει μετρήσεις σε έναν επιχειρηματικό λογαριασμό ή λογαριασμό από έναν απλό χρήστη του, θα μπορούσε να χρησιμοποιήσει το Instagram Graph API για να υποβάλει ερώτημα στο τελικό σημείο μετρήσεων του Instagram. Το αίτημα θα μοιάζει κάπως έτσι:

```
GET @  
graph.instagram.com/17841405822304914/insights?metric=impressions,reach,profile_views&period=day
```

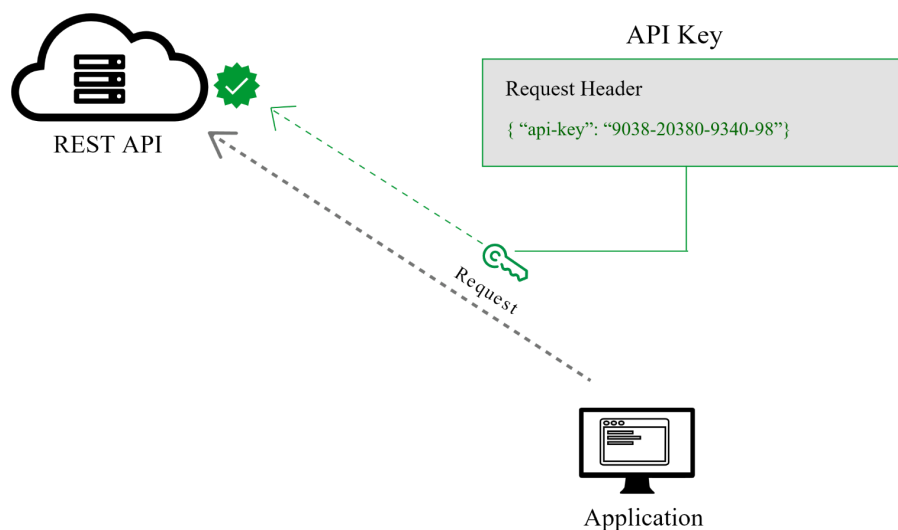
Συνήθως επειδή η πληροφορία που χρησιμοποιείται από εταιρίες δημιουργίας ολοκληρωμένων συστημάτων έχουν σχέση με προσπέλαση ευαίσθητων πληροφοριών του πελάτη, όπως μοτίβα συναλλαγών ή καρτέλες λογαριασμών των χρηστών, η χρήση οποιουδήποτε endpoint, με σκοπό να ενημερώσουμε, να αφαιρέσουμε ή να εισάγουμε δεδομένα, υλοποιείται ένα ενδιάμεσο στρώμα ασφαλείας το οποίο προστατεύει το κάθε endpoint, από όλες ή ορισμένες μεθόδους του HTTP. Αυτό σημαίνει ότι πλέον, για να καταφέρουμε να κατεβάσουμε τα στοιχεία του πελάτη ή να δούμε μια συναλλαγή του, θα πρέπει να σερβίρουμε μαζί με την όποια μέθοδο που υλοποιούμε και ένα TOKEN ή κλειδί πρόσβασης. Η παραπάνω διαδικασία ονομάζεται εξουσιοδότηση.

Η εξουσιοδότηση είναι η διαδικασία καθορισμού δικαιωμάτων/προνομίων πρόσβασης σε πόρους, η οποία συνδέεται με την ασφάλεια πληροφοριών γενικά και την ασφάλεια των υπολογιστών ειδικότερα, καθώς και με τον έλεγχο πρόσβασης. Μεμονωμένα αρχεία ή δεδομένα ενός αντικειμένου, προγράμματα υπολογιστή, συσκευές υπολογιστών και δυνατότητες εφαρμογών υπολογιστή είναι όλα παραδείγματα πόρων. Οι χρήστες υπολογιστών, το λογισμικό υπολογιστών και άλλο υλικό υπολογιστή είναι παραδείγματα πελατών. Η εξουσιοδότηση είναι μια συνάρτηση της φάσης καθορισμού πολιτικής, η οποία προηγείται της φάσης επιβολής πολιτικής, κατά την οποία τα αιτήματα πρόσβασης γίνονται δεκτά ή απορρίπτονται ανάλογα με τις εξουσιοδοτήσεις που έχουν καθοριστεί προηγουμένως. Υπάρχουν 4 είδη εξουσιοδότησης που χρησιμοποιούνται για τα APIs:

1. Με API κλειδιά (API keys)
2. Basic Auth (Βασική εξουσιοδότηση)
3. HMAC
4. OpenAuth 1.0 & 2.0

Για να χρησιμοποιήσετε τα περισσότερα API, πρέπει πρώτα να εγγραφείτε για ένα κλειδί API. Το κλειδί API είναι μια μεγάλη συμβολοσειρά που συνήθως περιλαμβάνεται στη διεύθυνση URL αιτήματος ή στην κεφαλίδα. Το κλειδί API χρησιμοποιείται ως επί το πλείστον για την αναγνώριση του ατόμου που εκτελεί την

κλήση API (για έλεγχο ταυτότητας για να χρησιμοποιήσετε το API). Το κλειδί API θα μπορούσε ενδεχομένως να συνδεθεί με μια συγκεκριμένη εφαρμογή που έχετε γράψει. Μπορεί να λάβετε τόσο δημόσια όσο και ιδιωτικά κλειδιά από API. Το δημόσιο κλειδί περιλαμβάνεται συνήθως στο αίτημα, ενώ το ιδιωτικό κλειδί χρησιμοποιείται κυρίως για επικοινωνία διακομιστή με διακομιστή και αντιμετωπίζεται περισσότερο σαν κωδικός πρόσβασης. Όταν συνδέεστε σε ορισμένους ιστότοπους τεκμηρίωσης API, το κλειδί API παρέχεται αυτόματα στο δείγμα κώδικα.



Εικόνα 5. Η εξουσιοδότηση του client για την χρήση του REST API

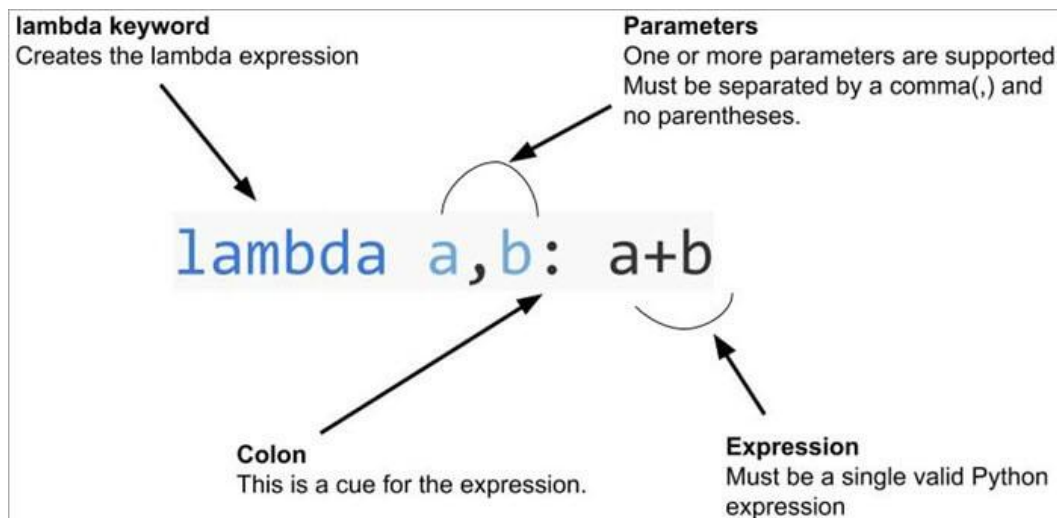
Το Basic Auth είναι ένας άλλος τύπος εξουσιοδότησης. Ο αποστολέας εισάγει ένα όνομα χρήστη: κωδικό πρόσβασης στην κεφαλίδα του αιτήματος χρησιμοποιώντας αυτόν τον τρόπο. Το Base64 είναι μια τεχνική κωδικοποίησης που μετατρέπει τη σύνδεση και τον κωδικό πρόσβασης σε ένα σύνολο 64 χαρακτήρων για να διασφαλίσει την ασφαλή μετάδοση. Τα API που υποστηρίζουν τη Βασική Εξουσιοδότηση θα υποστηρίζουν επίσης το HTTPS, το οποίο κρυπτογραφεί το περιεχόμενο του μηνύματος εντός του πρωτοκόλλου μεταφοράς HTTP. (Χωρίς HTTPS, οι χάκερ θα μπορούσαν εύκολα να αποκρυπτογραφήσουν το όνομα χρήστη και τον κωδικό πρόσβασης.) Ο διακομιστής API αποκρυπτογραφεί το μήνυμα και ελέγχει την κεφαλίδα όταν το λαμβάνει. Επιλέγει αν θα αποδεχτεί ή θα απορρίψει το αίτημα αφού αποκωδικοποιήσει τη συμβολοσειρά και αξιολογήσει το όνομα χρήστη και τον κωδικό πρόσβασης. Η εφαρμογή βασικού ελέγχου ταυτότητας HTTP (BA) είναι η απλούστερη τεχνική για την επιβολή ελέγχων πρόσβασης σε πόρους Ιστού, επειδή δεν απαιτεί cookie, αναγνωριστικά περιόδου σύνδεσης ή σελίδες σύνδεσης. Αντίθετα, ο βασικός έλεγχος ταυτότητας HTTP χρησιμοποιεί τυπικά πεδία στην κεφαλίδα HTTP.

Το HMAC σημαίνει κωδικός ελέγχου ταυτότητας μηνυμάτων που βασίζεται σε τεχνικές κατακερματισμού. Είναι ένας αλγόριθμος ψηφιακής υπογραφής που έχει σχεδιαστεί για να επαναχρησιμοποιεί τον αλγόριθμο αφομοίωσης μηνυμάτων όπως ο MD5 και ο SHA-1 και να παρέχει έναν αποτελεσματικό μηχανισμό πρωτοκόλλου ακεραιότητας δεδομένων. Καθώς το HMAC χρησιμοποιείται για την κρυπτογράφηση του απλού κειμένου με ασφαλή τρόπο, χρησιμοποιείται στο πρωτόκολλο Secure Socket Layer, πιστοποιητικό SSL και έχει επιλεγεί ως υποχρεωτική εφαρμογή ασφαλείας για το πρωτόκολλο διαδικτύου. Υπάρχουν 7 βήματα που εμπλέκονται στον Κώδικα ελέγχου ταυτότητας μηνυμάτων που βασίζεται σε κατακερματισμό. Το σημαντικό σημείο είναι ότι μόνο ο αποστολέας και ο παραλήπτης έχουν πρόσβαση στο μυστικό κλειδί (το οποίο απαιτείται για την ανακατασκευή του κατακερματισμού). Το αίτημα δεν περιλαμβάνει το μυστικό κλειδί. Όταν θέλουμε να βεβαιωθούμε ότι ένα αίτημα είναι και αυθεντικό και ότι δεν έχει παραβιαστεί, χρησιμοποιούμε την ασφάλεια HMAC.

Ένας άλλος τύπος εξουσιοδότησης είναι το OAuth, το πρότυπο εκχώρησης ανοιχτής πρόσβασης που επιτρέπει στους χρήστες του Διαδικτύου να παραχωρούν σε ιστότοπους ή εφαρμογές πρόσβαση στις πληροφορίες τους σε άλλους ιστότοπους χωρίς να χρειάζεται να τους δώσουν τους κωδικούς πρόσβασης τους. Εταιρείες όπως η Amazon, η Google, το Facebook, η Microsoft και το Twitter χρησιμοποιούν αυτήν την τεχνολογία για να επιτρέπουν στους χρήστες να ανταλλάσσουν πληροφορίες σχετικά με τους λογαριασμούς τους με εφαρμογές ή ιστότοπους τρίτων. Εκ μέρους ενός κατόχου πόρων, το OAuth παραχωρεί στους πελάτες «ασφαλή ανάθεση πρόσβασης» σε πόρους διακομιστή.

Περιγράφει πώς οι κάτοχοι πόρων μπορούν να παραχωρήσουν πρόσβαση τρίτων στους πόρους του διακομιστή τους χωρίς να χρειάζεται να παρέχουν διαπιστευτήρια. Το OAuth είναι ένα πρωτόκολλο που επιτρέπει σε έναν διακομιστή εξουσιοδότησης να παρέχει διακριτικά πρόσβασης σε τρίτους πελάτες με την άδεια του κατόχου του πόρου. Δημιουργήθηκε ρητά για χρήση με το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HTTP). Το τρίτο μέρος χρησιμοποιεί στη συνέχεια το διακριτικό πρόσβασης για να αποκτήσει πρόσβαση στους προστατευμένους πόρους του διακομιστή πόρων.

Για να διεξαχθούν λοιπόν οι κλήσεις που γίνονται σε ασφαλισμένα endpoints είτε στο weather api είτε στο atlas, είναι σημαντικό να έχουμε τα κατάλληλα κλειδιά ώστε να καταφέρουμε να προσπελάσουμε τα δεδομένα που επιστρέφουν. Στο παραπάνω παράδειγμα, για να κάνουμε κλήση σε ένα από αυτά, χρησιμοποιούμε το κλειδί που παρέχεται από μια lambda συνάρτηση, τις λέμε και ανώνυμες συναρτήσεις.



Εικόνα 6. Η δομή της λάμδα ανώνυμης συνάρτησης στην Python

Οι ανώνυμες συναρτήσεις μπορούν να χρησιμοποιηθούν για να υλοποιήσουν και να περιγράψουν λειτουργίες που δεν χρειάζεται να ονομαστούν και πιθανώς για βραχυπρόθεσμη χρήση. Η χρήση ανώνυμων συναρτήσεων είναι θέμα στυλ. Η χρήση τους δεν είναι ποτέ ο μόνος τρόπος επίλυσης ενός προβλήματος. Κάθε ανώνυμη συνάρτηση θα μπορούσε αντ' αυτού να οριστεί ως ονομαστική συνάρτηση και να καλείται ονομαστικά. Μερικοί προγραμματιστές χρησιμοποιούν ανώνυμες συναρτήσεις για να ενσωματώσουν συγκεκριμένο, μη επαναχρησιμοποιήσιμο κώδικα χωρίς να γεμίζουν τον κώδικα με πολλές μικρές κανονικές συναρτήσεις μιας γραμμής. Σε ορισμένες γλώσσες προγραμματισμού, οι ανώνυμες συναρτήσεις συνήθως υλοποιούνται για πολύ συγκεκριμένους σκοπούς, όπως η δέσμευση συμβάντων σε επανακλήσεις ή η δημιουργία της συνάρτησης για συγκεκριμένες τιμές, οι οποίες μπορεί να είναι πιο αποτελεσματικές, πιο ευανάγνωστες και λιγότερο επιρρεπείς σε σφάλματα από την κλήση μιας πιο γενικής συνάρτησης με όνομα.

Στην συνέχεια της συνάρτησης έχουμε την κλήση στο endpoint του API που θα μας επιστρέψει τις καιρικές παραμέτρους, οι οποίες παρέχονται δωρεάν από την δημόσια υπηρεσία που έχει συνταχθεί και στεγάζει το API. Όσο υπάρχουν διαθέσιμα παραμετρικά στο URL σε συνδυασμό με το κλειδί ταυτοποίησης, μπορούν να γίνουν κλήσεις χωρίς πρόβλημα.

Η επόμενη συνάρτηση που καλούμαστε να επεξεύσουμε, πρόκειται για την `get_esdalab_params` η οποία στόχο έχει να ανατρέξει στο API του έξυπνου σπιτιού και να εξάγει τις μετρήσεις της θερμοκρασίας και της υγρασίας για το εσωτερικό περιβάλλον του. Η συνάρτηση περιγράφεται παρακάτω:

```
def get_esdalab_params(self, param):
    url = "https://api.atlas.esdalab.ece.uop.gr/v1/ambient/"
    hds = {"X-Application-Key":
"a553b11b0f3841bc82c9ca01e5e125cf4b874cc1472671a25ba8d143d0adadbb","X-Request-For": "atlas"}
```

```

date_time = datetime.now().strftime("%m/%d/%Y, %H:%M:%S")
if param == 'temperature':
    temp_req = requests.get(url = url+param, headers=hds)
    temp_res = temp_req.json()
    if (temp_res):
        return {
            'temp':temp_res['measurement'],
            'recorded': date_time
        }
if param == 'humidity':
    temp_req = requests.get(url = url+param, headers=hds)
    temp_res = temp_req.json()
    if (temp_res):
        return {
            'hum':temp_res['measurement'],
            'recorded': date_time
        }
return []

```

Στις πρώτες γραμμές της συνάρτησης, γίνεται η αρχικοποίηση του URL συνδέσμου που αφορά το endpoint του API όπως και ένα λεξικό το οποίο εμπεριέχει όλα εκείνα τα παραμετρικά που θα σερβιριστούν μαζί με το request για να δοθεί εξουσιοδότηση και έπειτα πρόσβαση στις πληροφορίες του ATLAS.

Μετά την καταγραφή της ώρας που θα γίνει η κλήση, ορίζουμε ποιά παράμετρο θέλουμε να κάνουμε λήψη, καλούμε το API με τα δεδομένα από τις αρχικοποιήσεις και έπειτα μετατρέπουμε την απάντηση σε JSON και τα αποθηκεύουμε σε ένα λεξικό με δύο κλειδιά, το ένα το όνομα της παραμέτρου με αντίστοιχη τιμή το νούμερο που θα λάβουμε και έπειτα το timestamp με αντίστοιχη τιμή την ώρα που αρχικοποιούμε. Το ίδιο ακριβώς το κάνουμε και για την παράμετρο της υγρασίας. Η επόμενη συνάρτηση, καθιστά δυνατή την επικοινωνία της εφαρμογής με το chatbot Αλεξάνδρα, χρησιμοποιώντας το MQTT πρωτόκολλο.

```

def send(self, param):
    MQTT_HOST = "10.10.20.214"
    MQTT_PORT = 1883
    MQTT_CLIENT_ID = 'ale@detection'
    MQTT_USER = 'alexandra'
    MQTT_PASSWORD = 'esda'
    TOPIC = 'aalhouse/alexandra/clay/greet'

```

```

def ON_PUBLISH(client,userdata,result):
    pass

client = paho.Client(MQTT_CLIENT_ID)
client.username_pw_set(MQTT_USER, MQTT_PASSWORD)
client.connect(MQTT_HOST, MQTT_PORT)
payload = json.dumps(param,ensure_ascii=False)
client.publish(TOPIC, payload)

```

Η λογική της συνάρτησης είναι πολύ απλή. Στην πρώτες γραμμές γίνεται η αρχικοποίηση των μεταβλητών επικοινωνίας ώστε να μπορεί να στείλουμε στην Καλλιόπη το payload της επιλογής μας ή αλλιώς τα δεδομένα σε μορφή JSON. Τα δεδομένα μπορεί να είναι οποιαδήποτε μορφή και ιεραρχία επιλέξουμε.

Στη προκειμένη, το payload θα φορτωθεί με τις μετρήσεις που εξάγουμε από τα APIs, δίνοντάς τους κατάλληλα κλειδιά, ώστε μετέπειτα να μπορούμε να ρυθμίσουμε την Αλεξάνδρα να τα χρησιμοποιεί ώστε να μας επιστρέφει τις κατάλληλες απαντήσεις. Στην συνέχεια, αρχικοποιούμε μια εσωτερική συνάρτηση που πρέπει να υπάρχει ώστε να μπορεί να δουλέψει σωστά το module paho της Python. Η ON_PUBLISH συνάρτηση, μπορούμε να την περιγράψουμε και ως έναν event listener, δηλαδή μια διαδικασία που ενεργοποιείται όταν μια ορισμένη ενέργεια πραγματοποιηθεί. Αφού αυτή η ενέργεια ολοκληρωθεί, συνήθως καλείται μια callback συνάρτηση που σκοπό έχει να πραγματοποιήσει κάποιον υπολογισμό ή να μεταβολή στο περιβάλλον της εφαρμογής ή του εκάστοτε eco-system.

Τέλος, δημιουργούμε το αντικείμενο του πελάτη με την βοήθεια του κατασκευαστή paho, αφού ορίσουμε και εισάγουμε σωστά όλα τα κλειδιά (credentials). Η επόμενη ενέργεια είναι να συνδεθούμε στο IoT περιβάλλον ώστε να μπορεί να γίνεται η επικοινωνία με τον αισθητήρα της επιλογής μας. Για να στείλουμε τα δεδομένα που χρειαζόμαστε, μένει μόνο να αλλάξουμε την param μεταβλητή με το μήνυμα της επιλογής μας ή στη προκειμένη με τις μετρήσεις της επιλογής μας. Η json.dumps() θα μετασχηματίσει τα δεδομένα σε κατάλληλο αντικείμενο JSON για την χρήση του από τον αισθητήρα ή στη περίπτωσή μας, το chatbot Αλεξάνδρα. Το τελικό μέρος του κώδικα, αφορά την δημιουργία του αρχείου που θα αποθηκεύει τα δεδομένα και την κύρια λογική που όλες οι παραπάνω συναρτήσεις θα συντονιστούν προς ένα αποτέλεσμα. Συνήθως, το κώδικα αυτόν τον ονομάζουμε root code ή core code, επειδή χωρίς αυτόν, δεν γίνεται να εναρμονιστεί κάποια ενέργεια ή στην περίπτωσή μας, συλλογή δεδομένων. Ας εξετάσουμε αναλυτικά τον παρακάτω κώδικα της συνάρτησης start():

```

def start(self):
    # create data file if not exists
    file_name = 'data.csv'
    if (os.path.exists('data.csv')):
        # get measurements to plot
        esda = []
        open_api = []
        pass
    else:
        with open(file_name, 'w') as fn:
            fn.write('parameter_type, value, date,\n')
            fn.close()

    # get parameters from esda lab and open api
    esda = self.get_esdalab_params('temperature')
    open_api = self.get_openapi_weather()

    # write measurements in data.csv
    with open(file_name, 'a') as fn:
        fn.write('temp,'+str(esda['temp'])+', '+str(esda['recorded'])+',\n')
        fn.write('temp,'+str(open_api['temp'])+', '+str(open_api['recorded'])+',\n')
        fn.close()

# entry point
App().start()

```

Το αρχικό τμήμα της συνάρτησης, στόχο έχει να ελέγξει αν στο λειτουργικό σύστημα υπάρχει αντίστοιχο όνομα του αρχείου που προσπαθούμε να δημιουργήσουμε. Αυτό το κάνουμε για να μην έχουμε επανεγγραφή των δεδομένων, χάνοντας τα περιεχόμενα και είναι καλή πρακτική στο επίπεδο της μνήμης.

Στη περίπτωση που ήδη υπάρχει το αρχείο csv (comma separated values), απλά αρχικοποιούμε δύο κενές λίστες ώστε να μπορούμε να τραβήξουμε τα δεδομένα που ήδη έχουν γραφεί. Στην αντίθετη περίπτωση, δημιουργούμε έναν δείκτη στο αρχείο data.csv και γράφουμε τις πρώτες στήλες της πρώτης γραμμής ώστε να μπορούμε να μάθουμε τι είδους δεδομένα αντιστοιχούν σε τι στήλες. Στο συγκεκριμένο αρχείο, μας ενδιαφέρουν τρεις παράμετροι, το είδος της παραμέτρου, η τιμή της και η ημερομηνία που την τραβήξαμε.

Στο επόμενο βήματα χρησιμοποιούμε τις δύο συναρτήσεις που αφορούν τις μετρήσεις των αισθητήρων και τις μετρήσεις από την υπηρεσία open weather api και τις αποθηκεύουμε στις δύο λίστες που ορίσαμε και αναλόγως της ύπαρξης του αρχείου csv, είναι διαθέσιμες σε εμάς για εγγραφή ή όχι. Έπειτα, ξανά ανοίγουμε το αρχείο (αν αυτό υπάρχει -αν δεν υπάρχει δημιουργείται εκ νέου αυτομάτα-) και με τη σήμανση

append = 'a' ή προσάρτηση, ορίζουμε ότι το αρχείο πρόκειται να εγγραφεί στην συνέχεια των ήδη υπαρχόντων δεδομένων. Υπάρχουν έξι διαφορετικές σημάνσεις που μπορούν να χρησιμοποιηθούν:

1. **"r"** - Ανάγνωση - Προεπιλεγμένη τιμή. Ανοίγει ένα αρχείο για ανάγνωση, σφάλμα εάν το αρχείο δεν υπάρχει
2. **"a"** - Προσάρτηση - Ανοίγει ένα αρχείο για προσάρτηση, δημιουργεί το αρχείο εάν δεν υπάρχει
3. **"w"** - Write - Ανοίγει ένα αρχείο για εγγραφή, δημιουργεί το αρχείο εάν δεν υπάρχει
4. **"x"** - Δημιουργία - Δημιουργεί το καθορισμένο αρχείο, επιστρέφει ένα σφάλμα εάν το αρχείο υπάρχει
5. Επιπλέον, μπορείτε να καθορίσετε εάν το αρχείο θα πρέπει να χειρίζεται ως δυαδικό ή σε λειτουργία κειμένου
6. **"t"** - Κείμενο - Προεπιλεγμένη τιμή. Λειτουργία κειμένου
7. **"b"** - Δυαδική - Δυαδική λειτουργία (π.χ. εικόνες)

Έπειτα, μετά την εγγραφή των παραμέτρων της επιλογής μας στο αρχείο csv, ορίζουμε το entry point ή σημείο εισόδου της εφαρμογής με την βοήθεια του κατασκευαστή της κλάσης App, (App()), δημιουργώντας ένα αντικείμενο το οποίο με την σειρά του θα καλέσει την μέθοδο start() η οποία πραγματοποιεί την εναρμόνιση της εφαρμογής.

Κεφάλαιο 5 - Συμπεράσματα και επόμενα βήματα

Είναι κατανοητό, ότι με το κατάλληλο περιβάλλον και την σωστά ρυθμισμένη επικοινωνία των αισθητήρων που εμπεριέχει, ανάλογα με το ρόλο του κάθε υποσυστήματος, είμαστε σε θέση να κάνουμε την χρήση του και κατ επέκταση την διαβίωση σε αυτό μια ενέργεια πολύ αποδοτική, ξεκούραστη και ίσως και παραγωγική. Το έξυπνο σπίτι κατασκευάστηκε με στόχο να βελτιώσει την διαβίωση ηλικιωμένων ανθρώπων, με το να τους παρέχει εύκολες και έξυπνες λύσεις σε καθημερινά ζητήματα που ίσως σε κάποιο άλλο περιβάλλον να ήταν σχεδόν αδύνατες αλλά και χρονοβόρες, αφού οι ρυθμοί των ηλικιωμένων είναι σημαντικά χαμηλομένοι, πράγμα που σημαίνει ότι αν υφίσταται κάποια αδυναμία ή αναπηρία, ίσως τελικά η ενέργεια που πρόκειται να διεξαχθεί να μην καταφέρει να ολοκληρωθεί. Ορισμένες ενέργειες όταν αναφερόμαστε σε διαφορετικές ηλικιακές ομάδες, μπορεί να σημαίνει την διαφορά στη ζωή από τον θάνατο. Ένα παράδειγμα, είναι να μην εκτελεστεί σωστά ή εγκαίρως μια ειδοποίηση για την χορήγηση φαρμάκων, ή να μην λειτουργήσουν κάποιοι αυτοματισμοί που καταγράφουν τις ζωτικές λειτουργίες του ανθρώπου με αποτέλεσμα να μην ειδοποιηθούν οι κατάλληλες υπηρεσίες παροχής

υποστήριξης και υγείας. Στη προκειμένη, η ενοποίηση του chatbot και η εκμάθηση των πιθανών ερωτήσεων που μπορεί να λάβει, αποτελεί ακόμα ένα εύκολο μέσο διευκόλυνσης καθημερινών αναγκών για τους χρήστες. Αυτό σημαίνει ότι αν πρόκειται ο βγει ο χρήστης εκτός του χώρου, θα μπορεί να μάθει όλες εκείνες τις λεπτομέρειες σχετικά με καιρικές μετρήσεις, δίνοντας έτσι μια επιπλέον ευκολία στην προετοιμασία του. Ακριβώς επειδή η πληροφορία θα λαμβάνεται με την βοήθεια μιας απλής ερώτησης προς έναν εικονικό βοηθό, θα είναι και αρκετά αποδοτική όπως και ευχάριστη καθώς η πράξη θα διεξάγεται με μια πινελιά ανθρωπιάς.

Τα επόμενα βήματα καλούνται να ενορχηστρώσουν την ιδέα ενός εικονικού βοηθού ο οποίος θα μπορεί να κάνει συζήτηση με τον εκάστοτε χρήστη, παρέχοντάς του συντροφιά και δοκιμάζοντας το μυαλό του με διάφορα θέματα προς συζήτηση. Ο συγκεκριμένος εικονικός βοηθός, θα είναι σε θέση να ακούσει φράσεις μέσω μιας συσκευής bluetooth που θα φοράει ο χρήστης, κάνοντας έτσι την επικοινωνία πιο ποιοτική και πιο άμεση. Αυτό σημαίνει, ότι ο βοηθός θα μπορεί να αναλύει όλα εκείνα τα γραμματικά μοτίβα της φράσης και να απαντάει κατάλληλα δίνοντας έμφαση στην ψυχική και πνευματική κατάσταση του χρήστη. Η παραπάνω πρόταση, θα αποτελέσει μια ενδιαφέρουσα δοκιμασία σε επίπεδο υλοποίησης, καθώς θα πρέπει να υπάρχει τρόπος να μάθει ο βοηθός μοτίβα ψυχολογίας στα οποία θα επενεργεί κατάλληλα. Μια επιπλέον λειτουργικότητα που θα πρέπει να υλοποιηθεί είναι ο έλεγχος των αισθητήρων ή ορισμένων διαδικασιών που αφορούν το έξυπνο σπίτι όπως και την συσκευή που στεγάζει τον εικονικό βοηθό. Θα πρέπει να δίνεται η δυνατότητα στον χρήστη να ενεργοποιεί λειτουργίες με φωνητικές εντολές, κατά την διάρκεια της συζήτησης. Για να επιτευχθεί η υλοποίηση και δημιουργία του εικονικού βοηθού, θα πρέπει να αναλύσουμε την διαδικασία σε τέσσερα βήματα:

- 1. Δημιουργία λειτουργίας αναγνώρισης φωνητικών εντολών και ομιλίας**
(κατά προτίμηση μεθόδων που παρέχουν την δυνατότητα μετατροπής φωνητικών εντολών σε μορφή κειμένου. Το πρώτο βήμα σε αυτό το σημείο είναι να χρησιμοποιεί χωρίς πρόβλημα το μικρόφωνο και να υλοποιηθούν οι αντίστοιχες ρουτίνες που θα χειρίζονται τα δεδομένα.
- 2. Γραμματική ανάλυση και δημιουργία ποιοτικού ελέγχου:** Στην συγκεκριμένη διαδικασία, θα πρέπει να χρησιμοποιηθούν μέθοδοι ανάλυσης γραμματικής, ώστε ο εικονικός βοηθός να μπορεί να απαντήσει στα ερωτήματα και σκέψεις του χρήστη, με την χρήση σωστής γραμματικής σύνταξης, έστω και αν οι απαντήσεις του βασίζονται σε μια πίσινα απαντήσεων (responses pool). Αυτό το βήμα περιέχει και την υλοποίηση της ρουτίνας που αρχικά θα αναγνωρίζει αν ο χρήστης χρησιμοποιεί κατάφαση, άρνηση, ερώτηση, δήλωση ή συνδυασμό αυτών παράλληλα με κάποιο απλό συναίσθημα, όπως ο ενθουσιασμός.

3. **Συλλογή δεδομένων και εκπαίδευση:** Στο αρχικό αυτό σημείο, ο βοηθός θα πρέπει να έρθει σε επαφή με τον χρήστη, ώστε ο χρήστης να μπορεί να του διδάξει κάποιες ενδεικτικές απαντήσεις σε ερωτήσεις και δηλώσεις του δίνοντάς του τα αρχικά ερεθίσματα. Η βέλτιστη λύση θα ήταν να εξάγουμε ένα μεγάλο όγκο ερωτοαπαντήσεων από τον διαδίκτυο και να τα φορτώσουμε στον βοηθό. Ωστόσο, για να πραγματοποιηθεί κάτι τέτοιο, θα πρέπει να γίνει βελτιστοποίηση εκείνων των ερωτημάτων αναζήτησης αυτών των φράσεων με τρόπο τέτοιο ώστε να μπορούμε να χτίσουμε εξαρχής την κάθε φράση, στην σωστή γραμματική σύνταξη.

4. **Ενοποίηση νευρωνικού δικτύου για την επιλογή απαντήσεων και ανάλυση συναισθημάτων:** Αυτό το στάδιο πρόκειται για το πιο πειραματικό μέρος της ανάπτυξης, καθώς στόχο θα έχει να διδάξει τον εικονικό βοηθό όλα εκείνα τα σημεία του λόγου που ο χρήστης εκφράζει και μεταδίδει έντονα συναισθήματα. Αυτό σημαίνει ότι θα προσπαθεί να δημιουργήσει ένα μοτίβο ώστε να βρεί την ψυχική κατάσταση του χρήστη και έπειτα να επιλέγει πώς να ενεργεί, δηλαδή να απαντάει στον χρήστη, με κατάλληλο τρόπο ώστε ο χρήστης να μην περνάει σε κατάσταση άμυνας. Μια αντίστοιχη μέθοδος θα μπορούσε να δημιουργηθεί για την αντιμετώπιση κατάθλιψης σε ομάδες ανθρώπων μεγαλύτερης ηλικίας, με το να ενεργεί με τρόπο ανατρεπτικό για την ανθρώπινη συναισθηματική κατάσταση.

Ορισμένα έργα έχουν διερευνήσει εάν οι ενσωματωμένοι εικονικοί πράκτορες και η είσοδος ή η έξοδος ομιλίας μπορούν να είναι επωφελείς. Το έργο GUIDE, χρησιμοποιώντας επίσης ομάδες εστίασης, ανακάλυψε ότι η προφορική γλώσσα ήταν μακράν η πιο προτιμώμενη μέθοδος αλληλεπίδρασης για ηλικιωμένους χρήστες που δεν είναι εξοικειωμένοι με την τεχνολογία και ότι τα άβαταρ θα έπρεπε να εξαφανίζονται σε αλληλεπίδραση χωρίς προβλήματα, αλλά θα πρέπει να επανεμφανίζονται αυτόματα όταν υπάρχει πρόβλημα εργασίας. εντοπίστηκε. Beskow et al. εφάρμοσε μια προσέγγιση σχεδιασμού με επίκεντρο τον χρήστη για να αναπτύξει ένα σύστημα υπενθύμισης για δύο χρήστες με γνωστικά προβλήματα μετά από χειρουργική επέμβαση. Παρουσίασαν ένα πρωτότυπο για πολυτροπική διαχείριση χρονοδιαγράμματος με χρήση αναγνώρισης χειρογράφου και προφορικού διαλόγου με εικονικό χαρακτήρα.

Όσον αφορά την εισαγωγή ομιλίας, οι ήπιες έως μέτριας σοβαρότητας διαταραχές της άρθρωσης, όπως η ελαφρώς ασυνήθιστη ή μπερδεμένη ομιλία, εστιάζονται ιδιαίτερα στην έρευνα τα τελευταία χρόνια, με λύσεις για μειωμένα ποσοστά σφαλμάτων αναγνώρισης που κυμαίνονται από την εφαρμογή ακουστικών μοντέλων ηλικιωμένων ηχείων, έως πολυτροπική συγχώνευση εισόδου ομιλίας και δακτυλογραφημένη εισαγωγή αρχικών γραμμμάτων για αποσαφήνιση. Ακόμη και πριν από ασθενείς με βαριά αναπηρία, υπάρχουν λύσεις που μπορούν να παρέχουν αρκετά αξιόπιστο έλεγχο του περιβάλλοντος μέσω της αναγνώρισης μιας λέξης.

Αρκετοί βοηθοί για συγκεκριμένους τομείς με τη μορφή chatbots έχουν κατακτήσει πολλές εμπορικές και ιδιωτικές περιοχές. Ωστόσο, εξακολουθεί να υπάρχει ένα περιορισμένο επίπεδο συστηματικής γνώσης των διακριτικών χαρακτηριστικών των στοιχείων σχεδιασμού για chatbots για τη διευκόλυνση της ανάπτυξης, υιοθέτησης, υλοποίησης και περαιτέρω έρευνας. Διάφορα επιστημονικά άρθρα εξετάζουν το σχεδιασμό και τη μηχανική των chatbots από διαφορετικές τεχνικές προοπτικές, π.χ. τη συναισθηματική νοημοσύνη (π.χ. Feine et al. 2019) ή τα ανθρωπόμορφα χαρακτηριστικά (Kim et al. 2018) και άλλα επικεντρώνονται στη μελέτη των chatbots σε συγκεκριμένες εφαρμογές τομείς (π.χ. Bittner et al. 2019). Οι Gnewuch et al. (2017) παρέχουν μια βασική ταξινόμηση των chatbot με βάση δύο διαστάσεις που ονομάζονται «πλαίσιο» και «κύριος τρόπος επικοινωνίας». Η πρώτη διάσταση κατηγοριοποιεί τα chatbot σε γενικής χρήσης και σε συγκεκριμένο τομέα, ενώ η δεύτερη διάσταση τα ταξινομεί σε κείμενο και ομιλία.

Τα chatbot βασισμένα στην ομιλία γενικής χρήσης, όπως το Google Assistant, η Cortana, η Alexa και το Siri είναι τα πιο διαδεδομένα και συχνά χρησιμοποιούμενα chatbots (Di Prospero et al. 2017; Keruska and Bohouta 2018). Αυτές οι βοηθητικές εφαρμογές που ενεργοποιούνται με φωνή εγκαθίστανται συνήθως απευθείας από τον κατασκευαστή του smartphone ή της έξυπνης συσκευής και προσφέρουν μια μεγάλη ποικιλία λειτουργιών (Keruska και Bohouta 2018). Από την άλλη πλευρά, τα chatbot που βασίζονται σε ομιλία και για συγκεκριμένο τομέα, μπορούν να βρεθούν, για παράδειγμα, ως βοηθοί εντός οχήματος σε αυτοκίνητα (Diederich et al. 2019) ή σύμφωνα με την περίπτωση μας, εντός του χώρου ενός σπιτιού. Τα chatbot ειδικά για τον τομέα και τα βασισμένα σε κείμενο αλληλεπιδρούν με ανθρώπους κυρίως μέσω μηνυμάτων κειμένου για ένα συγκεκριμένο θέμα (Gnewuch et al. 2017; Diederich et al. 2019). Αυτά τα chatbots αναλαμβάνουν διαφορετικές εργασίες σε αμέτρητους τομείς εφαρμογών, όπως η υποστήριξη πελατών, η εκπαίδευση, τα ταξίδια, τα οικονομικά και η κινητικότητα (Følstad et al. 2019), γεγονός που καθιστά τα chatbot για συγκεκριμένο τομέα προκλητικά για τους ερευνητές. Στην επιστημονική βιβλιογραφία, τα chatbot για συγκεκριμένο τομέα έχουν αναλυθεί και ταξινομηθεί σύμφωνα με ορισμένα κριτήρια. Για παράδειγμα, οι Maedche et al. (2016) αναφέρουν ότι οι διαφορές μεταξύ των chatbots μπορούν κυρίως να αφαιρεθούν ως προς τα επίπεδα αλληλεπίδρασης και νοημοσύνης.

Οι Chaves και Gerosa (2019) περιγράφουν τη νοημοσύνη ως την ικανότητα ενός chatbot να συμμετέχει σε έναν διάλογο έχοντας επίγνωση του συζητούμενου θέματος, ενώ ο Jain et al. (2018) πιστεύουν ότι η ευφυΐα ενός chatbot μπορεί επίσης να συναχθεί από την ικανότητά του να θέτει προληπτικά κατάλληλες ερωτήσεις και να εμπλέκει τον συμμετέχοντα σε έναν ουσιαστικό και ανθρώπινο διάλογο. Σε ολιστικό επίπεδο, σύμφωνα με την προτεινόμενη τελική ταξινόμηση, τα στοιχεία σχεδιασμού για chatbots που σχετίζονται με συγκεκριμένα χαρακτηριστικά νοημοσύνης μπορούν να περιγραφούν χρησιμοποιώντας 15 χαρακτηριστικά, τα οποία με τη σειρά τους μπορούν να κατηγοριοποιηθούν στις ακόλουθες 5 διαστάσεις: Το πλαίσιο νοημοσύνης D1 απεικονίζει τον υποκείμενο γνωστικό σχεδιασμό συστήματος που οριοθετεί τις

τεχνικές αρχές βάσει των οποίων ένα chatbot επικοινωνεί, επεξεργάζεται πληροφορίες ή/και επιλέγει μια ενέργεια ή απάντηση (Saravanan et al. 2017; Knote et al. 2018; Diederich et al. 2019). Το πηλίκο νοημοσύνης D2 υποδεικνύει εάν ένα chatbot βασίζεται κυρίως σε απλούς κανόνες αντιστοίχισης μοτίβων “αν-τότε”, αν κατανοεί την εισαγωγή κειμένου ή αν έχει τη δυνατότητα να βελτιώσει τις απαντήσεις του μέσω μαθηματικών υπολογισμών, συμπερασμάτων ή αναγνώρισης φωτογραφιών κ.λπ. (Wei et al. 2018· Knote et al. 2018).

Ενώ το πλαίσιο νοημοσύνης ταξινομεί ολόκληρη τη διαδικασία συνομιλίας, η διάσταση πηλίκου νοημοσύνης περιγράφει τη νοημοσύνη στην αξιολόγηση μιας μεμονωμένης απάντησης. Η επεξεργασία προσωπικότητας D3 χαρακτηρίζει την ικανότητα του chatbot να προσαρμόζει με έμφαση τις απαντήσεις σημειογραφίας του στη συγκεκριμένη προσωπικότητα και διάθεση του χρήστη, προσδιορίζοντας το χαρακτηριστικό της προσωπικότητας του αντίστοιχου (Di Prospero et al. 2017; Yorita et al. 2019). Το chatbot προσαρμόζεται στο χαρακτηριστικό της προσωπικότητας του χρήστη που αναγνωρίζεται σε πραγματικό χρόνο (Yorita et al. 2019). Η κοινωνικο-συναισθηματική συμπεριφορά D4 χαρακτηρίζει την ικανότητα συντονισμού του chatbot να δείχνει στοργή ή ενσυναίσθηση προς τις ατομικές ανάγκες και την άμεση ανατροφοδότηση του χρήστη, την οποία ο χρήστης αποκαλύπτει μέσω συναισθημάτων που αντηχούν μέσα σε έναν διάλογο (Bittner et al. 2019; Yalçın 2019).

Ο Κιούσης (2002, σελ. 372) ορίζει την αλληλεπίδραση ως “ο βαθμός στον οποίο μια τεχνολογία επικοινωνίας μπορεί να δημιουργήσει ένα διαμεσολαβημένο περιβάλλον στο οποίο οι συμμετέχοντες μπορούν να επικοινωνούν [. . .], τόσο συγχρονισμένα όσο και ασύγχρονα, και να συμμετέχουν σε αμοιβαίες ανταλλαγές μηνυμάτων”. Σύμφωνα με αυτό, τα chatbot επιτρέπουν στους ανθρώπους να αλληλεπιδρούν με συστήματα υπολογιστών μέσω γραπτής ή/και προφορικής φυσικής γλώσσας με στόχο να οδηγήσουν την αλληλεπίδραση όσο το δυνατόν πιο φυσικά ώστε να μοιάζει με διάλογο πρόσωπο με πρόσωπο (Diederich et al. 2019). Ωστόσο, η βασική πρόκληση σχεδιασμού σε αυτό το επίπεδο αφαίρεσης είναι η δημιουργία φυσικών αλληλεπιδράσεων με ανθρώπινα στοιχεία για την υποστήριξη της εμπειρίας αλληλεπίδρασης (Bittner et al. 2019; Gnewuch et al. 2018a, b). Ενσωματώνοντας εννοιολογικές και εμπειρικές γνώσεις, εντοπίσαμε 17 χαρακτηριστικά διαδραστικών λειτουργιών που επιτρέπουν στα chatbot να αλληλεπιδρούν με τους χρήστες τους. Αυτά τα χαρακτηριστικά μπορούν να αναπαρασταθούν μέσω των επόμενων 7 διαστάσεων: Η πολυτροπικότητα D6 δείχνει προς την ικανότητα του chatbot να λαμβάνει εισόδους και να ανταποκρίνεται μέσω ενός ή διαφόρων τρόπων αλληλεπίδρασης ή καναλιών επικοινωνίας, π.χ. κειμένου, φωνής, έκφρασης προσώπου κ.λπ. (Knote et al. 2018). Η ταξινόμηση αλληλεπίδρασης D7 χαρακτηρίζει τη μέθοδο αλληλεπίδρασης ανθρώπου-υπολογιστή (HCI) που χρησιμοποιείται από το chatbot (den Boer 2017).

Το πλαίσιο γενικά είναι το σύνολο όλων των σιωπηρών και ρητών πληροφοριών κατάστασης σχετικά με ανθρώπους, αντικείμενα, χρόνο και τοποθεσία μέσα σε μια αλληλεπίδραση που μπορεί να χρησιμοποιηθεί για να περιγράψει μια κατάσταση (Abowd et al. 1999; Kim et al. 2018). Το πλαίσιο δείχνει εάν και σε ποιον τομέα λειτουργεί το chatbot (Gnewuch et al. 2017; Diederich et al. 2019). Τα χαρακτηριστικά του περιβάλλοντος στο οποίο λαμβάνουν χώρα οι αλληλεπιδράσεις μπορούν να ταξινομηθούν και να κατηγοριοποιηθούν σε 17 χαρακτηριστικά ομαδοποιημένα στις 5 διαστάσεις που περιγράφονται παρακάτω.

Ο ρόλος του chatbot D13 υποδηλώνει τον ρόλο που παίζει το chatbot κατά τη διάρκεια της αλληλεπίδρασης (Bittner et al. 2019). Η διάρκεια της σχέσης D14 περιγράφει την ικανότητα του chatbot να θυμάται πληροφορίες από προηγούμενες συνομιλίες για να επηρεάζει μελλοντικές αλληλεπιδράσεις (Wei et al. 2018; Følstad et al. 2019). Ο τομέας εφαρμογής D15 καθορίζει τον κύριο σκοπό εφαρμογής για τον οποίο έχει σχεδιαστεί το chatbot (Zumstein and Hundertmark 2017; Knote et al. 2018). Ο στόχος συνεργασίας D16 καθορίζει εάν το chatbot βοηθά ή όχι τον χρήστη να επιτύχει έναν κοινό στόχο ή εργασία (Bittner et al. 2019). Το κίνητρο για τη χρήση chatbot D17 προσδιορίζει το κύριο εξωτερικό κίνητρο για τη χρήση του chatbot από την οπτική γωνία του χρήστη (Deci and Ryan 2000, Brandtzaeg and Følstad 2017).

Σε μια ανάλυση των συζητήσεων γύρω από την τεχνολογία και την υγειονομική περίθαλψη, οι Greenhalgh et al. διαπίστωσε ότι «οι διαφορετικοί ενδιαφερόμενοι έχουν διαφορετικές υποθέσεις, αξίες και κοσμοθεωρίες, συζητούν ο ένας τον άλλον και ανταγωνίζονται για αναγνώριση και πόρους». Ενώ η ανάλυση των Greenhalgh et al. εντόπισε τέσσερις αλληλεπικαλυπτόμενους λόγους (μοντερνιστικό, ουμανιστικό, πολιτική οικονομία και διαχείριση της αλλαγής), αυτό θα εξετάσει τη διασταύρωση των κυρίαρχων βιοϊατρικών και νεοφιλελεύθερων λόγου και του εναλλακτικού λόγου της ενεργού γήρανσης. [14] Ο βιοϊατρικός λόγος κυριαρχεί στην ακαδημαϊκή βιβλιογραφία που αναφέρεται στην τεχνολογία και τη γήρανση. Αυτή η ομιλία δίνει έμφαση στις βλάβες των ηλικιωμένων και ιατροποιεί τη γήρανση ως ασθένεια. Στην ανασκόπηση των συστημάτων AAL, ορισμένες συσκευές επικεντρώθηκαν αποκλειστικά στην αντιμετώπιση φυσιολογικών αναγκών. Η υπερβολική έμφαση στη βιοϊατρική προοπτική θέτει σε κίνδυνο τη νομιμοποίηση των υφιστάμενων σχέσεων εξουσίας, όπως η σχέση γιατρού-ασθενούς, ο ρόλος των φαρμακευτικών προϊόντων στη θεραπεία ασθενειών και η έμφαση στην οξεία και όχι στην προληπτική υγειονομική περίθαλψη. Ο νεοφιλελεύθερος λόγος για την υγεία και τη γήρανση εκτιμά την ιδέα των οικονομικών συστημάτων ως ορθολογικών και αυτορυθμιζόμενων συστημάτων που βασίζονται στη σχέση μεταξύ προσφοράς και ζήτησης. [14]

Οι συζητήσεις σχετικά με τη γήρανση του πληθυσμού δείχνουν πώς ο νεοφιλελεύθερος και ο βιοϊατρικός λόγος όχι μόνο διασταυρώνονται αλλά υποστηρίζουν ο ένας τον άλλον. Οι ηλικιωμένοι συχνά παρουσιάζονται ως μη παραγωγικά μέλη της κοινωνίας που χρησιμοποιούν υπερβολικούς πόρους, δημιουργώντας έτσι βάρος στα παραγωγικά μέλη της κοινωνίας. Οι τεχνολογίες AAL

θεωρούνται λύσεις στα προβλήματα που σχετίζονται με τη γήρανση του πληθυσμού. Ειδικότερα, οι τεχνολογίες υποτίθεται ότι μειώνουν τη χρήση σπάνιων πόρων υγειονομικής περίθαλψης και κοινωνικών πόρων και τις εισαγωγές σε νοσοκομεία και οίκους ευγηρίας. Αν και αυτό είναι ίσως μια υπεραπλούστευση των πολύπλοκων διαδικασιών που διέπουν την ανάπτυξη των λύσεων AAL, δυστυχώς αντηχεί με τις εμπειρίες πολλών γεροντολόγων που εργάζονται σε αυτόν τον τομέα (συμπεριλαμβανομένων ορισμένων από τους σημερινούς συγγραφείς). Συνήθως, οι «ανάγκες» των ηλικιωμένων εξομοιώνονται με ελλείμματα, απώλειες, ασθένειες και εξαρτήσεις και οι λύσεις θεωρούνται συχνά ως άμεσες διορθώσεις σε αυτά. Οι λύσεις συχνά αποτυγχάνουν να κατανοήσουν το πλαίσιο μέσα στο οποίο βιώνονται αυτές οι «ανάγκες» (π.χ. διαθεσιμότητα υποστήριξης έναντι απομόνωσης), τις προτιμήσεις του ατόμου (π.χ. συναισθήματα στιγματισμού έναντι αισθήσεων κυριαρχίας) και τον αντίκτυπο στην καθημερινότητα ζωής (π.χ. αλλαγές στις σχέσεις εξουσίας μεταξύ ατόμου και φροντιστή). Επιπλέον, η προβληματική της εμπειρίας της τρίτης ηλικίας παρέχει έναν ιδιαίτερο φακό που αποκλείει τις πολλές θετικές πλευρές της γήρανσης του ατόμου, οι οποίες, αν και διακυβεύονται, δεν πρέπει να αγνοούνται. Από ηθική άποψη, όταν οι τεχνολογίες AAL αναπτύσσονται χωρίς να αμφισβητείται η βιοϊατρική ατζέντα, μπορεί ακούσια να υποστηρίζουν υποθέσεις και προσδοκίες που εξισώνουν τους ηλικιωμένους με τις ιατρικές τους καταστάσεις. [14]

Στο πλαίσιο της γεροτεχνολογίας, το AAL είναι μια σχετικά νέα προσέγγιση για το πώς οι τεχνολογικές λύσεις μπορούν να ανταποκριθούν στις απαιτήσεις και τις ανάγκες των ηλικιωμένων ενηλίκων. Αυξάνοντας την αυτονομία, την αυτοπεποίθηση και την κινητικότητα: αυξάνοντας ή διατηρώντας την υγεία και τη λειτουργική ικανότητα και προωθώντας τον ενεργό τρόπο ζωής, οι τεχνολογίες AAL αναμένεται να μειώσουν τον κίνδυνο αναπηρίας και ιδρυματοποίησης, να ενισχύσουν την ασφάλεια, να αποτρέψουν την κοινωνική απομόνωση και να διατηρήσουν τα δίκτυα υποστήριξης. δίνοντας τη δυνατότητα στους ηλικιωμένους να γερνούν στη θέση τους. Προκειμένου να επιτευχθεί ο στόχος της γήρανσης επί τόπου, η έρευνα και ανάπτυξη AAL πρέπει να περιλαμβάνει μια συνεργασία μεταξύ ειδικών πεδίου στις επιστήμες υγείας, αποκατάστασης, γεροντολογίας και κοινωνικών επιστημών, καθώς και τεχνικών ειδικών στη μηχανική, την επιστήμη υπολογιστών και τη ρομποτική [14]. Με γνώμονα τα παραπάνω, η επέκταση της συγκεκριμένης εργασίας, θα επιχειρήσει να αναλύσει και να δημιουργήσει μια εξειδικευμένη μορφή ψηφιακού βοηθού, ο οποίος θα είναι ικανός να παρουσιάζει απόψεις λαμβάνοντας υπόψη τα πλαίσια που θέτει ο χρήστης, δίνοντας βάση στον συναισθηματικό και λογικό του κόσμο, χωρίς να αναιρεί ψυχρά την τοποθέτησή του. Η κύρια δοκιμασία στην εργασία θα είναι ο βοηθός να έχει την ικανότητα -ή να του δοθεί η ικανότητα- να χαρτογραφεί συναισθήματα του χρήστη όταν αλληλεπιδρά μαζί του με την βοήθεια μοτίβων συμπεριφοράς και σώματος, αν πρόκειται να υπάρξει αναβάθμιση με κάμερα αναγνώρισης κινήσεων.

Βιβλιογραφία

1. History and Philosophy of Python -
<https://python-course.eu/python-tutorial/history-and-philosophy-of-python.php>
2. Web Application Programming Interfaces (APIs): general-purpose standards, terms and European Commission initiatives APIs4DGov study — digital government APIs: the road to value-added open API-driven services Santoro, M., Vaccari, L., Mavridis, D., Smith, R. S., Posada, M., Gattwinkel, D.
3. A comprehensive study on MQTT as a low power protocol for internet of things application, Diploma of Telecommunication Engineering, Universitas Telkom, Jl. Telekomunikasi no.1, Bojongsoang, Bandung, Indonesia
4. Eclipse Foundation - Python Paho Client Documentation
<https://www.eclipse.org/paho/index.php?page=clients/python/docs/index.php>
5. Why people use chatbots, Petter Bae Brandtzaeg and Asbjørn Følstad, SINTEF, Forskningsveien 1, 0314 Oslo, Norway
6. Xu, A., Liu, Z., Guo, Y., Sinha, V., Akkiraju, R.: A New Chatbot for Customer Service on Social Media. In: Proceedings of the ACM Conference on Human Factors in Computing Systems (2017)
7. Hawley, M.S., Enderby, P., Green, P., Cunningham, S., Brownsell, S., Carmichael, J., Parker, M., Hatzis, A., O'Neill, P., Palmer, R.: A speech-controlled environmental control system for people with severe dysarthria. *Medical Engineering & Physics*, 29(5):586–593. (2007)
8. Virtual agents as daily assistants for elderly or cognitively impaired people Studies on acceptance and interaction feasibility Ramin Yaghoubzadeh, Marcel Kramer, Karola Pitsch, and Stefan Kopp CITEC, Bielefeld University
9. Speech-driven environmental control systems – a qualitative analysis of users' perceptions - *Disability & Rehabilitation: Assistive Technology*, 2009 Zoe Robertson, Simon Judge, Mark Hawley, Pamela Enderby
10. GUIDE Consortium: User Interaction & Application Requirements - Deliverable D2.1. (2011)
11. Beskow, J., Edlund, J., Granström, B., Gustafson, J., Skantze, G., Tobiasson, H.: The MonAMI Reminder : a spoken dialogue system for face-to-face interaction. In: Proceedings of the 10th Annual Conference of the International Speech Communication Association, INTERSPEECH 2009. pp. 300–303. (2009)
12. Young, V., Mihailidis, A.: Difficulties in Automatic Speech Recognition of Dysarthric Speakers and Implications for Speech-Based Applications Used by the Elderly: A Literature Review. In: *Assistive Technology*, 22(2):99–112. (2010)
13. Fager, S.K., Beukelman, D.R., Jakobs, T., Hosom, J.-P.: Evaluation of a Speech Recognition Prototype for Speakers with Moderate and Severe Dysarthria: A Preliminary Report. *Augmentative and Alternative Comm.*, 26(4):267–277. (2010)

14. Ambient Assisted Living Technologies for Aging Well: A Scoping Review
Stephanie Blackman, Claudine Matlo, Charisse Bobrovitskiy, Ashley Waldoch,
Mei Lan Fang, Piper Jackson, Alex Mihailidis, Louise Nygård, Arlene Astell
and Andrew Sixsmith