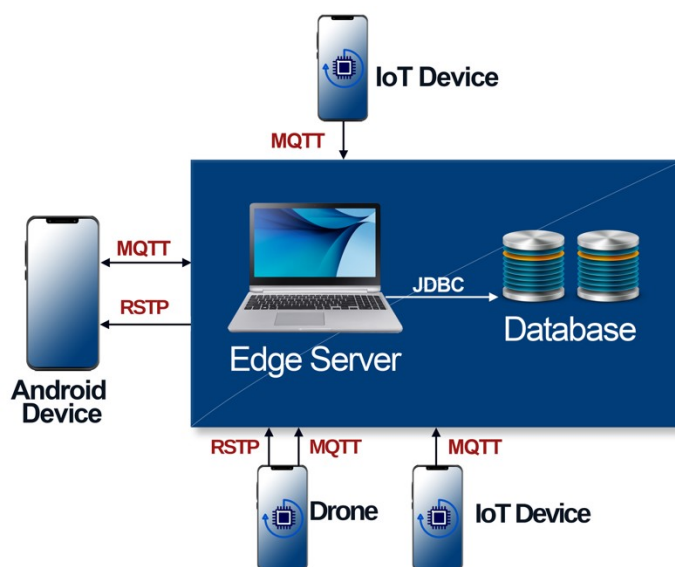




Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Ανάπτυξη Λογισμικού για Δίκτυα και Τηλεπικοινωνίες
Χειμερινό εξάμηνο
2025 – 2026

Σκοπός της παρούσας εργασίας είναι η ανάπτυξη ενός συστήματος Περιβαλλοντικής Διαχείρισης, το οποίο εστιάζει στην ασφάλεια των πολιτών σε καταστάσεις έκτακτης ανάγκης.

Το σχήμα που ακολουθεί αποτελεί μία γραφική απεικόνιση των δομικών στοιχείων της αρχιτεκτονικής, η οποία και θα αναπτυχθεί στα πλαίσια της εργασίας. Επιπρόσθετα, ακολουθεί αναλυτική περιγραφή των οντοτήτων που απαρτίζουν την αρχιτεκτονική αλλά και των interfaces που τα διασυνδέουν.



Android Device: Πρόκειται για την κινητή συσκευή η οποία θα βρίσκεται στα χέρια κάποιου χρήστη με σκοπό την ενημέρωσή του σε περίπτωση ανίχνευσης κινδύνου. Η android συσκευή συνδέεται μέσω πρωτοκόλλου MQTT με τον Edge Server (θα αναλυθεί παρακάτω). Επιπρόσθετα, οι android συσκευές αποστέλλουν στον Edge Server το στίγμα GPS τους για την εύρεση της θέσης τους μέσα στο χώρο με χειροκίνητο ή αυτόματο τρόπο. Η επικοινωνία των android συσκευών με τον Edge Server είναι αμφίδρομη καθώς πέραν από την αποστολή δεδομένων, θα λαμβάνουν ειδοποιήσεις σχετικά με τις περιβαλλοντικές συνθήκες που επικρατούν. Οι σχετικές ειδοποιήσεις που λαμβάνουν τα τερματικά περιγράφονται αναλυτικά παρακάτω. Επίσης, οι android

συσκευές πρέπει να αποστέλλουν περιοδικά στον Edge Server δεδομένα τηλεμετρίας (όπως το ποσοστό μπαταρίας τους) ώστε να παρακολουθείται η κατάστασή τους.

IoT Device: Πρόκειται στατικές συσκευές οι οποίες θα βρίσκονται τοποθετημένες στο περιβάλλον με σκοπό την αποστολή μετρήσεων που θα συλλέγουν μέσω από μία σειρά ενσωματωμένων αισθητήρων. Όλες οι IoT συσκευές θα πρέπει περιοδικά να αποστέλλουν το στίγμα GPS τους με χειροκίνητο ή αυτόματο τρόπο καθώς και το ποσοστό μπαταρίας που διαθέτουν. Θα επικοινωνούν αποκλειστικά και μόνο με τον Edge Server και πάλι μέσω πρωτοκόλλου MQTT.

Drone: Πρόκειται για μη επανδρωμένα εναέρια οχήματα (drones), τα οποία παραμένουν αρχικά σταθευμένα σε σταθμούς φόρτισης και διαθέτουν περιορισμένη διάρκεια πτήσης λόγω μπαταρίας. Σε περίπτωση εντοπισμού κινδύνου, ο Edge Server μπορεί να δώσει εντολή σε ένα διαθέσιμο drone να απογειωθεί και να κατευθυνθεί στο σημείο του περιστατικού. Κατά την άφιξη, το drone εκκινεί ζωντανή μετάδοση βίντεο (RTSP streaming) προς τον Edge Server, η οποία στη συνέχεια μπορεί να προωθηθεί σε κοντινές Android συσκευές. Καθ'όλη τη διάρκεια της πτήσης, το drone αποστέλλει δεδομένα θέσης (GPS) και στοιχεία τηλεμετρίας (πχ. Υπολειπόμενος χρόνος πτήσης/ποσοστό μπαταρίας) στον Edge Server. Το drone επιστρέφει στη βάση του είτε κατόπιν εντολής του Edge Server είτε αυτόματα όταν η μπαταρία του κοντεύει να εξαντληθεί.

Edge Server (ES): Ο Edge Server είναι υπεύθυνος για τη συλλογή των δεδομένων που αποστέλλονται από τα τερματικά (Android, IoT συσκευές και drones) καθώς και για την ενημέρωση των android συσκευών σε περίπτωση εντοπισμού κινδύνου. Ακόμα, στον Edge Server θα λάβει χώρα η ανάλυση των δεδομένων που λαμβάνει από τις IoT συσκευές προκειμένου να γίνει έγκαιρα ο εντοπισμός κινδύνου. Οι κίνδυνοι που εντοπίζονται θα πρέπει να καταχωρούνται σε μία βάση δεδομένων. Επιπλέον, μπορεί να αποστέλλει εντολές σε διαθέσιμα drones προκειμένου να παρέχουν ζωντανή εικόνα από την περιοχή του συμβάντος.

MySQL Database: Το τελευταίο δομικό στοιχείο της εργασίας είναι η MySQL βάση δεδομένων, στην οποία θα διατηρούνται όλες οι ανιχνεύσεις κινδύνου που εντοπίζονται από τον Edge Server, καθώς και όλα τα δεδομένα τηλεμετρίας που συλλέγονται από τις συσκευές (Android, IoT, Drones) και τον ίδιο τον Edge Server.

SUMO Data: Στα πλαίσια της εργασίας θα δοθούν 2 xml αρχεία (**android_1.xml**, **android_2.xml**), τα οποία έχουν παραχθεί με τη χρήση του SUMO και περιέχουν αντιπροσωπεύουν την κίνηση 2 οχημάτων μέσα στην περιοχή της Πανεπιστημιούπολης. Παρακάτω δίνεται ένα παράδειγμα xml αρχείου.

```
<?xml version="1.0" encoding="UTF-8"?>
<fcd-export xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/fcd_file.xsd">
  <timestep time="0.00">
    <vehicle id="27" x="23.774376" y="37.968361" angle="152.898572" type="car" speed="0.000000" pos="5.100000" lane="196650677_0" slope="0.000000"/>
  </timestep>
  <timestep time="1.00">
    <vehicle id="27" x="23.774381" y="37.968353" angle="152.898572" type="car" speed="1.000000" pos="6.100000" lane="196650677_0" slope="0.000000"/>
  </timestep>
  <timestep time="2.00">
    <vehicle id="27" x="23.774390" y="37.968337" angle="152.898572" type="car" speed="2.000000" pos="8.100000" lane="196650677_0" slope="0.000000"/>
  </timestep>
  <timestep time="3.00">
    <vehicle id="27" x="23.774405" y="37.968312" angle="152.898572" type="car" speed="3.000000" pos="11.100000" lane="196650677_0" slope="0.000000"/>
  </timestep>
</fcd-export>
```

- **timestep**: Χρόνος στο simulation σε seconds
- **id**: Το αναγνωριστικό του κάθε οχήματος
- **x**: longitude
- **y**: latitude
- **angle**: Η γωνία του οχήματος (0 °-360 °) σύμφωνα με το navigational standard
- **type**: Ο τύπος του vehicle
- **speed**: Η ταχύτητα του οχήματος (m/s)
- **pos**: Η θέση του οχήματος μετρώντας την από την αρχή της τρέχουσας λωρίδας που διανύει το όχημα
- **lane**: Το αναγνωριστικό της τρέχουσας λωρίδας
- **slope**: κλίση οχήματος

Στα πλαίσια της εργασίας θα χρειαστούν **μόνο** τα εξής στοιχεία: **x (longitude)**, **y (latitude)**

Αναλυτική Περιγραφή Εργασίας

Android Συσκευή (όχημα)

Θα πρέπει να υλοποιήσετε μια εφαρμογή σε android η οποία επικοινωνεί με τον Edge Server μέσω MQTT με την χρήση publish και subscribe. Θα κάνει **publish** καθώς και **subscribe** σε **ένα topic**. Το topic στο οποίο θα κάνει **publish** θα χρησιμοποιείται για την αποστολή των γεωγραφικών συντεταγμένων της (**latitude**, **longitude**) καθώς και ένα **αναγνωριστικό της συσκευής (device_id)** (είναι στην επιλογής σας το είδος του αναγνωριστικού) στον Edge Server. Πολύ σημαντικό σε αυτό το σημείο είναι να αναφερθεί ο τρόπος με το οποίον θα γίνεται αυτή η αποστολή. Όπως αναφέρθηκε στην εισαγωγή μία android συσκευή θα μπορεί να στέλνει την θέση της με χειροκίνητο ή αυτόματο τρόπο.

1. **Αυτόματός Τρόπος**

Θα πρέπει να λαμβάνει μετρήσεις κατευθείαν μέσω του GPS sensor που βρίσκεται προεγκατεστημένος στο android.

2. **Χειροκίνητος Τρόπος**

Θα πρέπει να χρησιμοποιεί ένα από τα αρχεία που έχουν δοθεί μαζί με την εκφώνηση. Θα επιλέγετε τυχαία ένα από τα 2 που υπάρχουν και θα χρησιμοποιείτε αυτό το στίγμα για την αναφορά θέσης του τερματικού. Θα πρέπει να μετατρέψετε το xml αρχείο σε csv μέσα στο οποίο θα υπάρχουν μόνο οι μετρήσεις που χρειάζονται (ας αναφερθούμε σε κάθε γραμμή ως **Διάνυσμα Μετρήσεων**):

<Latitude, Longitude>

Η συχνότητα αποστολής των διανυσμάτων μετρήσεων είναι **σταθερή** και δεν αλλάζει. Κάθε διάνυσμα μετρήσεων αντιστοιχεί σε **1 δευτερόλεπτο χρόνου**. Αυτό σημαίνει ότι η android συσκευή πρέπει να στέλνει ένα διάνυσμα μετρήσεων

από το csv του **ανά 1 δευτερόλεπτο** έτσι ώστε πάντα ο Edge Server να ενήμερος για την θέση του.

Ακόμη, ο χρήστης θα μπορεί να ορίσει, μέσα από το menu ρυθμίσεων της εφαρμογής, το **χρόνο** που θα διαρκέσει συνολικά η αποστολή των δεδομένων. Για παράδειγμα, εάν ο χρήστης επιλέξει 10 δευτερόλεπτα, αυτό σημαίνει ότι η αποστολή θα διαρκέσει 10 δευτερόλεπτα, οπότε θα σταλθούν 10 διανύσματα μετρήσεων στον Edge Server. Η **default** επιλογή θα είναι να σταλθούν όλα τα διανύσματα μετρήσεων στον Edge Server.

Επίσης, θα πρέπει να υπάρχει η **δυνατότητα για διακοπή της αποστολής** των μετρήσεων οποιαδήποτε στιγμή. Ακόμα δηλαδή και αν ο χρήστης έχει επιλέξει να σταλθούν όλα τα διανύσματα μετρήσεων, θα μπορεί ανά πάσα στιγμή με ένα κουμπί της εφαρμογής, που καλείστε να υλοποιήσετε, να τερματίζει την αποστολή.

Σημαντικό!

Ο χρήστης θα πρέπει να μπορεί να ρυθμίζει ανά πάσα στιγμή τον τρόπο με τον οποίο θα γίνεται η αποστολή στίγματος (θα μπορεί να σταματάει τον αυτόματο και να γυρίζει κατευθείαν σε χειροκίνητο καθώς και το ανάποδο).

Επιπρόσθετα, η android συσκευή θα διαθέτει **menu** ρυθμίσεων που θα περιέχει:

- Επιλογή για τον ορισμό της IP και του Port επικοινωνίας με τον **Edge Server**.
- Επιλογή του χρόνου αποστολής των **διανυσμάτων μετρήσεων** σε περίπτωση επιλογής χειροκίνητου τρόπου όπως αναλύθηκε παραπάνω.
- Επιλογή για έξοδο από την εφαρμογή. Με την επιλογή αυτή, θα πρέπει να προβληθεί στο χρήστη προειδοποιητικό μήνυμα αυτής της ενέργειας προκειμένου να επιλέξει εάν όντως επιθυμεί να τερματίσει την εφαρμογή.

Συνεχίζοντας, το **topic** στο οποίο πρέπει να είναι **subscribed** η android συσκευή θα έχει τον ρόλο λήψης ειδοποιήσεων σε περίπτωση εντοπισμού κινδύνου. Πιο συγκεκριμένα όταν η android συσκευή λάβει ειδοποίηση κινδύνου από τον Edge Server θα πρέπει να προβάλλει **οπτικοακουστική** ειδοποίηση, η οποία θα διαφοροποιείται ανάλογα με τον βαθμό επικινδυνότητας του Event κινδύνου. Θεωρούμε ότι υπάρχουν δύο βαθμοί επικινδυνότητας «Μέτριος Κίνδυνος» και «Υψηλός Κίνδυνος». Η υλοποίηση αυτής της ειδοποίησης είναι στην ευχέρειά σας. Επιπρόσθετα, μαζί με τον βαθμό επικινδυνότητας θα αποστέλλεται και η απόσταση της android συσκευής από την/τις IoT συσκευές (περιγράφεται αναλυτικά παρακάτω).

Επιπρόσθετα, η android συσκευή θα πρέπει να κάνει **περιοδικούς ελέγχους** για το αν υπάρχει σύνδεση στο **Internet**. Η επιλογής της περιόδου ελέγχου ορίζεται από εσάς. Σε περίπτωση που διαπιστωθεί ότι δεν υπάρχει σύνδεση στο δίκτυο, θα πρέπει η εφαρμογή να παραπέμπει τον χρήστη, με τρόπο της επιλογής σας, στην ενεργοποίησή της.

Τέλος, η Android συσκευή θα πρέπει να αποστέλλει περιοδικά στον Edge Server δεδομένα τηλεμετρίας της, όπως το ποσοστό μπαταρίας, τη χρήση της CPU, τη διαθέσιμη μνήμη και αποθηκευτικό χώρο, καθώς και παραμέτρους δικτύου (π.χ. ισχύ εκπομπής και διαθέσιμο εύρος ζώνης, μαζί με τη χρήση τους), συνοδευόμενα από χρονική σήμανση (timestamp). Τα δεδομένα αυτά θα προέρχονται από dataset που θα σας δοθεί. Επιπλέον, όταν μια Android συσκευή (όχημα) βρίσκεται εντός ορισμένης ακτίνας από ένα drone που βρίσκεται σε περιστατικό κινδύνου, θα πρέπει να λαμβάνει σε πραγματικό χρόνο το ζωντανό βίντεο που μεταδίδει το drone, μέσω του Edge Server.

IoT Συσκευές

Για την υλοποίηση αυτού του δομικού στοιχείου θα χρειαστούν **2 κινητά τηλέφωνα** τα οποία θα έχουν των ρόλο των IoT συσκευών. Αυτές οι συσκευές θα θεωρήσουμε ότι είναι απλές χαμηλής υπολογιστικής ισχύς και μοναδικό σκοπό έχουν την συλλογή μετρικών από του ενσωματωμένους αισθητήρες που διαθέτουν. Κάθε συσκευή θεωρούμε ότι διαθέτει **4** αισθητήρες οι οποίοι θα πρέπει να υλοποιηθούν από εσάς. Παρακάτω ακολουθεί μια αναλυτική παρουσίαση των αισθητήρων που καλείστε να υλοποιήσετε:

1. Αισθητήρας Καπνού – Smoke Sensor

Πρόκειται για αισθητήρες, οι οποίοι όπως μαρτυρά και το όνομά τους χρησιμοποιούνται για τον εντοπισμό καπνού. Για τις ανάγκες της εργασίας θα θεωρήσουμε ότι μετράνε το Optical Density (OD) της ατμόσφαιρας και όταν ένα κατώφλι ξεπεραστεί ενεργοποιούνται. Αυτό το κατώφλι το θεωρούμε σταθερό και είναι της τάξης των >0.14 OD. Το συνολικό εύρος τιμών του αισθητήρα είναι $[0, 0.25]$.

2. Αισθητήρας Αερίων - Gas Sensor

Πρόκειται για αισθητήρες οι οποίοι έχουν σκοπό να μετράνε την συγκέντρωση εύφλεκτων υλικών στην ατμόσφαιρα όπως Προπάνιο, Μεθάνιο Βουτάνιο καθώς και Αμμωνία. Για τις ανάγκες της εργασίας θα θεωρήσουμε ότι αυτοί οι αισθητήρες μετράνε την περιεκτικότητα στο αέρα ενός από τα παραπάνω υλικά. Όταν η περιεκτικότητα ξεπεράσει και πάλι κάποιο κατώφλι ενεργοποιούνται. Αυτό το κατώφλι το θεωρούμε σταθερό και είναι της τάξης του $>9.15\%$. Το συνολικό εύρος τιμών του αισθητήρα είναι $[0, 11]$.

3. Αισθητήρας Θερμοκρασίας

Πρόκειται για αισθητήρες οι οποίοι έχουν σκοπό να μετράνε την θερμοκρασία του περιβάλλοντος. Πάλι εάν η θερμοκρασία ξεπεράσει κάποιο όριο ενεργοποιούνται. Αυτά τα όρια είναι >50 °C. Το συνολικό εύρος τιμών του αισθητήρα είναι $[-5, 80]$.

4. Αισθητήρας Υπεριώδους ακτινοβολίας

Πρόκειται για αισθητήρες οι οποίοι έχουν σκοπό να μετράνε την ένταση της υπεριώδους ακτινοβολίας χρησιμοποιώντας τον δείκτη UV. Τιμή στον δείκτη UV > 6 θεωρείται μεγάλος κίνδυνος. Το συνολικό εύρος τιμών του αισθητήρα είναι $[0, 11]$.

Επιπρόσθετα, η IoT συσκευή θα διαθέτει **menu** ρυθμίσεων που θα περιέχει:

- Επιλογή για τον ορισμό της IP και του Port επικοινωνίας με τον **Edge Server**.
- Δυνατότητα επιλογής χειροκίνητου ή αυτόματου τρόπου αποστολής της θέσης του αισθητήρα. Στην περίπτωση του αυτόματου τρόπου, θα πρέπει να λαμβάνει μετρήσεις κατευθείαν μέσω του GPS sensor που βρίσκεται προεγκατεστημένος στο android, ενώ στον χειροκίνητο η θέση του κάθε αισθητήρα θα είναι προεπιλεγμένη και θα βρίσκεται στο τμήμα Πληροφορικής και Τηλεπικοινωνιών. Πιθανές προεπιλεγμένες θέσεις αισθητήρων:
 - ο 37.96809452684323, 23.76630586399502
 - ο 37.96799937191987, 23.766603589104385
 - ο 37.967779456380754, 23.767174897611685
 - ο 37.96790421900921, 23.76626294807113
- Επιλογή για τη δημιουργία νέου αισθητήρα (η λειτουργία αυτή αναλύεται παρακάτω).
- Επιλογή για έξοδο από την εφαρμογή. Με την επιλογή του, θα πρέπει να προβληθεί στο χρήστη προειδοποιητικό μήνυμα αυτής της ενέργειας προκειμένου να επιλέξει εάν όντως επιθυμεί να τερματίσει την εφαρμογή.

Με το άνοιγμα της android εφαρμογής που καλείστε να υλοποιήσετε, κάθε αισθητήρας θα πρέπει να έχει δικό του tab, το οποίο θα περιλαμβάνει:

- Τον τύπο του αισθητήρα
- Έναν slider, με εύρος τιμών ανάλογα με τον τύπο αισθητήρα. Μέσω του slider θα μπορεί να επιλέγεται η μέτρηση του αισθητήρα στο προαναφερθέν εύρος τιμών του.
- Κουμπί «Ενεργοποίησης/Απενεργοποίησης» αισθητήρα. Εδώ να σημειωθεί, πως η ενεργοποίηση/απενεργοποίηση αφορά την αποστολή των μετρήσεων στον Edge Server.

Από τους παραπάνω αισθητήρες θεωρούμε ότι οι πρώτοι 2 είναι προεγκατεστημένοι στην android εφαρμογή και οι υπόλοιποι 2 θα πρέπει να δημιουργηθούν από εσάς σε πραγματικό χρόνο. Πιο συγκεκριμένα:

1. **Προεγκατεστημένος αισθητήρας:**
Κάθε προεγκατεστημένος αισθητήρας θα έχει ήδη δικό του tab κατά το άνοιγμα της android εφαρμογής, με τις λειτουργίες που αναφέρθηκαν παραπάνω.
2. **Δημιουργία νέου αισθητήρα:**
Από το **menu** ρυθμίσεων θα δίνεται η δυνατότητα δημιουργίας νέου τύπου αισθητήρα, τα βήματα της οποίας περιλαμβάνουν:
 1. Ορισμός τύπου αισθητήρα
 2. Ορισμός εύρους τιμών αισθητήρα
 3. Δημιουργία νέου tab αισθητήρα με τις λειτουργίες που αναφέρθηκαν παραπάνω

Κάθε IoT συσκευή θα αποστέλλει κάθε 1 δευτερόλεπτο στον Edge Server:

- Τις επιλεγμένες από τον slider μετρήσεις των ενεργών αισθητήρων

- Τις γεωγραφικές συντεταγμένες της IoT συσκευής
- Το ποσοστό μπαταρίας της IoT συσκευής
- Την τρέχουσα κατανάλωση ενέργειας της IoT συσκευής.

Η αποστολή θα πραγματοποιείται μέσω της χρήσης του MQTT πρωτοκόλλου. Πιο συγκεκριμένα, κάθε IoT συσκευή θα κάνει **publish** σε ένα topic όλα τα δεδομένα προς αποστολή στον Edge Server (μετρήσεις από όλους τους ενεργούς αισθητήρες, την τοποθεσία και το ποσοστό της μπαταρίας της IoT συσκευής). **Προσοχή!** Κάθε IoT συσκευή θα έχει το δικό της topic και δεν θα το διαμοιράζεται με άλλη συσκευή.

Drone (Android συσκευή)

Το δομικό αυτό στοιχείο αφορά έναν στόλο από drones, τα οποία για τους σκοπούς της εφαρμογής αντιμετωπίζονται ως Android συσκευές. Κάθε drone παραμένει αρχικά σε κατάσταση αναμονής σε έναν σταθμό φόρτισης και διαθέτει συγκεκριμένη αυτονομία πτήσης (π.χ. έως 30 λεπτά συνεχούς λειτουργίας). Η συγκεκριμένη αυτονομία θα πρέπει να μπορεί να ορίζεται από εσάς. Σε περίπτωση που ο Edge Server εντοπίσει συμβάν κινδύνου και υπάρχει διαθέσιμο drone, θα δοθεί εντολή εκκίνησής του ώστε να απογειωθεί και να κατευθυνθεί προς την τοποθεσία του περιστατικού. Εάν δεν υπάρχει διαθέσιμο drone τη δεδομένη στιγμή, το σύστημα θα πρέπει να καταγράφει κατάλληλα αυτή την έλλειψη (π.χ. εμφανίζοντας σχετικό μήνυμα). Όταν το drone φτάσει στην περιοχή του συμβάντος, εκκινεί μια ζωντανή ροή βίντεο (RTSP stream) προς τον Edge Server. Το βίντεο που θα πρέπει να μεταδοθεί θα σας το παρέχουμε εμείς. Ο Edge Server προωθεί τη ροή αυτή σε πραγματικό χρόνο προς τις κοντινές Android συσκευές (οχήματα) που βρίσκονται σε ακτίνα γύρω από το drone (η ακτίνα αυτή είναι ρυθμιζόμενη από εσάς). Καθ' όλη τη διάρκεια της πτήσης του, το drone αποστέλλει στον Edge Server την τρέχουσα γεωγραφική θέση του (στίγμα GPS) καθώς και δεδομένα τηλεμετρίας, όπως το ποσοστό μπαταρίας, τη χρήση της CPU, τη διαθέσιμη μνήμη και αποθηκευτικό χώρο, καθώς και παραμέτρους δικτύου (π.χ. ισχύ εκπομπής και διαθέσιμο εύρος ζώνης, μαζί με τη χρήση τους), συνοδευόμενα από χρονική σήμανση (timestamp). Τα δεδομένα αυτά θα προέρχονται από dataset που θα σας δοθεί. Το drone επιστρέφει στον σταθμό βάσης του είτε μετά από εντολή του Edge Server είτε αυτόνομα, όταν η διαθέσιμη μπαταρία του μειωθεί σε επίπεδο τέτοιο ώστε να επαρκεί οριακά για την ασφαλή επιστροφή.

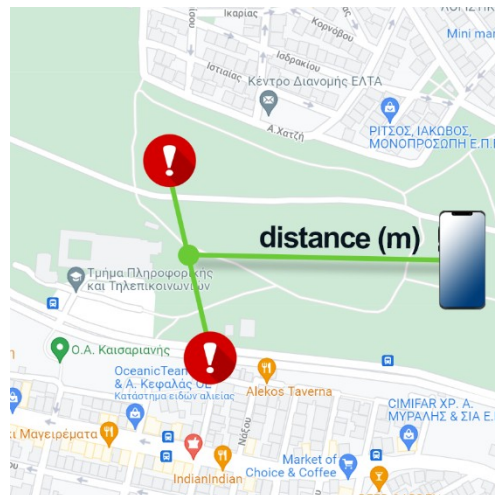
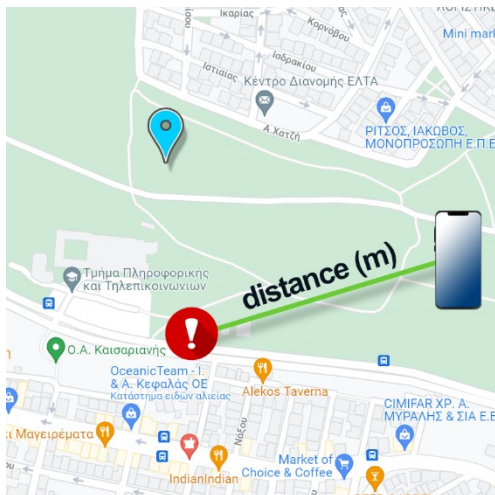
Edge Server

Ο Edge Server είναι ένα από τα πιο σημαντικά δομικά στοιχεία αυτής της εφαρμογής καθώς είναι υπεύθυνος για την επεξεργασία όλων των δεδομένων που έρχονται από τους αισθητήρες. Επιπρόσθετα, είναι υπεύθυνος για την ενημέρωση των android συσκευών σε περίπτωση εντοπισμού κινδύνου. Πιο συγκεκριμένα ο Edge Server χρησιμοποιεί μία threshold based λογική για τον εντοπισμό κινδύνου. Εσείς καλείστε να υλοποιήσετε αυτή την λογική σύμφωνα με τα βήματα που παρουσιάζονται παρακάτω.

1. Ο Edge Server λαμβάνει τα δεδομένα που έχουν γίνει publish από την IoT συσκευή στο αντίστοιχο topic.

2. Για κάθε αισθητήρα πραγματοποιεί έλεγχο σχετικά με το αν η μέτρηση που στάλθηκε βρίσκεται κάτω από το κατώφλι κινδύνου του (η τιμή του οποίου έχει δοθεί στην περιγραφή των αισθητήρων) ή όχι.
3. Ανάλογα με το ποιοι τύποι αισθητήρων έχουν ξεπεράσει τα κατώφλια κινδύνου, προσδιορίζεται και ο βαθμός επικινδυνότητας ενός ενδεχόμενου event. Πιο συγκεκριμένα ο βαθμός επικινδυνότητας υπολογίζεται σύμφωνα με:
 - Εάν ο αισθητήρας καπνού σε συνδυασμό με τον αισθητήρα αερίου έχουν ξεπεράσει τα προκαθορισμένα κατώφλια τότε □ «Υψηλός Κίνδυνος»
 - Εάν ο αισθητήρας καπνού σε συνδυασμό με τον αισθητήρα αερίου δεν έχουν ξεπεράσει τα προκαθορισμένα κατώφλια αλλά τα έχουν ξεπεράσει ο αισθητήρας θερμοκρασίας σε συνδυασμό με τον αισθητήρα υπεριώδους ακτινοβολίας τότε □ «Μέτριος Κίνδυνος»
 - Εάν μόνο ο αισθητήρας αερίου έχει ξεπεράσει το προκαθορισμένο κατώφλι τότε □ «Υψηλός Κίνδυνος»
 - Εάν όλοι οι αισθητήρες έχουν ξεπεράσει τα προκαθορισμένα κατώφλια τότε □ «Υψηλός Κίνδυνος»

Μετά την ολοκλήρωση της απόφασης για τον βαθμό επικινδυνότητας, ο Edge Server θα πρέπει να ενημερώσει την Android συσκευή για το συγκεκριμένο περιστατικό. Επιπρόσθετα, ο Edge Server θα πρέπει να υπολογίσει και να αποστείλει (στην Android συσκευή) την απόσταση σε μέτρα της Android συσκευής από την IoT συσκευή από την οποία προήλθε το event κινδύνου. Σε περίπτωση που και οι 2 IoT συσκευές σημάνουν ταυτόχρονα κίνδυνο, τότε θα πρέπει να υπολογιστεί το κέντρο της απόστασης μεταξύ τους και στη συνέχεια να υπολογιστεί η απόσταση της Android συσκευής από αυτό το κέντρο. (Για τον υπολογισμό των αποστάσεων μπορείτε να χρησιμοποιήσετε τις γεωγραφικές συντεταγμένες της Android συσκευής καθώς και των IoT συσκευών. Προς διευκόλυνσή σας προτείνεται το παρακάτω link για τον υπολογισμό της απόστασης δύο γεωγραφικών σημείων: <https://www.geodatasource.com/developers/java>.)



Στη συνέχεια, εφόσον υπάρχει διαθέσιμο drone, ο Edge Server δίνει εντολή εκκίνησης σε ένα drone για να μεταβεί στην τοποθεσία του περιστατικού. Αν δεν υπάρχει διαθέσιμο drone, θα πρέπει να καταγραφεί αυτή η έλλειψη (π.χ. με κατάλληλο μήνυμα στη διεπαφή ή στα logs). Όταν το drone φτάσει στην περιοχή του συμβάντος, αρχίζει να μεταδίδει ζωντανά βίντεο (RTSP streaming) προς τον Edge Server. Ο Edge Server με τη σειρά του θα μεταδίδει τη ροή αυτή στις Android συσκευές (οχήματα) που βρίσκονται σε ακτίνα κοντά στο drone (η ακτίνα αυτή είναι ρυθμιζόμενη από τον χρήστη). Τέλος, το συγκεκριμένο event κινδύνου θα πρέπει να καταχωρηθεί στη βάση δεδομένων, στη μορφή που περιγράφεται στο κεφάλαιο σχετικά με τη MySQL βάση.

Στον Edge Server θα πρέπει να υλοποιηθεί μία γραφική διεπαφή για την επισκόπηση των IoT και Android συσκευών που είναι συνδεδεμένες στο δίκτυο. Πιο συγκεκριμένα, με χρήση του Google Maps API θα πρέπει να εμφανίζονται επί του χάρτη, markers για κάθε συσκευή του δικτύου βάσει της γεωγραφικής τους θέσης. Το εικονίδιο του marker για τις IoT συσκευές θα πρέπει να είναι διαφορετικό από αυτό των συσκευών Android, για λόγους διάκρισης του είδους συσκευής. Για κάθε marker του χάρτη, κι επομένως για κάθε συσκευή του δικτύου, θα πρέπει να υπάρχει σχετικό παράθυρο πληροφόρησης ("Info window"), προκειμένου να δίνονται λεπτομέρειες που αφορούν την κάθε συσκευή.

Πιο συγκεκριμένα:

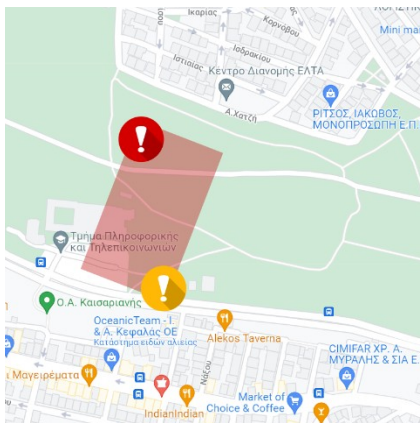
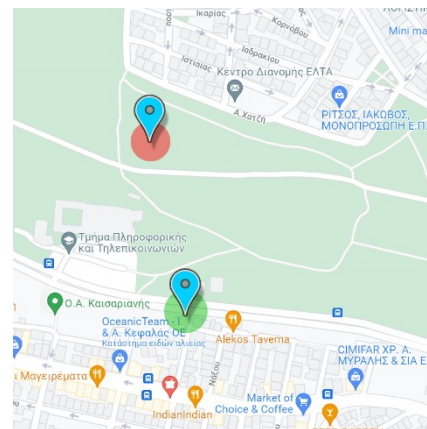
Το **InfoWindow των IoT συσκευών** θα πρέπει να περιλαμβάνει real-time (ανανέωση κάθε δευτερόλεπτο) πληροφορία σχετικά με:

1. Τύπος αισθητήρα
2. Μέτρηση αισθητήρα
3. Γεωγραφικές συντεταγμένες

Το **InfoWindow των Android συσκευών** θα πρέπει να περιλαμβάνει real-time (ανανέωση κάθε δευτερόλεπτο) πληροφορία σχετικά με:

1. Αναγνωριστικό συσκευής (**Device_id**)
2. Γεωγραφικές συντεταγμένες

Γύρω από κάθε IoT συσκευή θα πρέπει να σχηματίζεται ένας κύκλος με πράσινη ή κόκκινη χρωματική ένδειξη γνωστοποιώντας αν η συσκευή είναι ενεργή ή όχι. Μια IoT συσκευή θεωρείται ανενεργή, όταν **και οι 4 αισθητήρες** της είναι απενεργοποιημένοι. Για την διευκόλυνσή σας δίνεται μία οπτικοποίηση στο διπλανό σχήμα.



Στην περίπτωση που ο Edge Server εντοπίσει Event Κινδύνου **και στις 2** IoT συσκευές, θα πρέπει αυτό το event να οπτικοποιηθεί και στη γραφική διεπαφή.

Πιο συγκεκριμένα, θα πρέπει να σχηματιστεί ορθογώνιο παραλληλόγραμμο με κόκκινη χρωματική ένδειξη, με τις 2 IoT συσκευές τοποθετημένες στα άκρα μιας εκ των δύο διαγωνίων (ανάλογα με τις

θέσεις τους στο χάρτη). Το ορθογώνιο παραλληλόγραμμο θα υποδηλώνει ότι η γεωγραφική περιοχή, που βρίσκεται εντός αυτού, θεωρείται “μπλοκαρισμένη” ως επικίνδυνη. Επιπλέον, τα icons των markers των συσκευών θα αλλάζουν σε προειδοποιητικά κινδύνου και το χρώμα αυτών θα είναι είτε πορτοκαλί για “Μέτριο Κίνδυνο”, είτε κόκκινο για “Υψηλό Κίνδυνο”. Για την διευκόλυνσή σας δίνεται μία οπτικοποίηση στο διπλανό σχήμα.

Τέλος, ο Edge Server θα πρέπει να:

- Διατηρεί μία ενημερωμένη λίστα διαθέσιμων drones. Η λίστα αυτή θα αποθηκεύεται στη βάση δεδομένων και θα ενημερώνεται δυναμικά κατά τη λειτουργία του συστήματος, περιλαμβάνοντας την κατάσταση κάθε drone (charging, standby, flying) και τον υπολειπόμενο χρόνο πτήσης (μπαταρία) του.
- Την καταγραφή και αποθήκευση των δεδομένων τηλεμετρίας που συλλέγονται από όλες τις συσκευές (drones, Android οχήματα και IoT συσκευές).
- Την καταγραφή και αποθήκευση των δεδομένων τηλεμετρίας του ίδιου του Edge Server (π.χ. ποσοστά χρήσης CPU, μνήμης κ.λπ.).

Σημείωση: Η συλλογή δεδομένων τηλεμετρίας όλων των συσκευών μπορεί να υλοποιηθεί είτε από το dataset που θα σας παρέχουμε είτε με χρήση εξειδικευμένων εργαλείων λογισμικού για πραγματικές μετρήσεις (βλ. ενότητα Bonus).

MySQL Βάση:

Στα πλαίσια της εργασίας θα γίνει χρήση MySQL βάσης για αποθήκευση δεδομένων που θα προκύψουν από τη λειτουργία του συστήματος. Πιο συγκεκριμένα, στην βάση θα διατηρούνται όλες οι ανιχνεύσεις κινδύνου που εντοπίζονται από τον Edge Server. Καλείστε να δημιουργήσετε τη βάση δεδομένων και να συνδεθείτε με αυτήν μέσω **JDBC driver**. Στη βάση θα πρέπει υποχρεωτικά να λαμβάνονται υπόψιν οι εξής τύποι δεδομένων:

1. Timestamp Event κινδύνου με μορφοποίηση [dd/mm/yyyy hh:mm:ss].
2. Γεωγραφικές συντεταγμένες IoT συσκευής.
3. Τιμές ανά αισθητήρα τις IoT συσκευής, που ανίχνευσαν Event κινδύνου.
4. Βαθμός επικινδυνότητας Event κινδύνου.

Επιπλέον, στη βάση δεδομένων θα πρέπει να αποθηκεύονται και τα εξής δεδομένα κατά τη λειτουργία του συστήματος:

1. Οι μετρήσεις τηλεμετρίας όλων των συσκευών (Android συσκευών/οχημάτων, IoT συσκευών και drones), με τις αντίστοιχες χρονικές σημάνσεις τους.
2. Η λίστα με τα drones που είναι διαθέσιμα, συμπεριλαμβανομένης της τρέχουσας κατάστασής τους (Charging, Standby ή Flying) και του υπολειπόμενου χρόνου πτήσης (μπαταρία) για το καθένα.
3. Οι μετρήσεις τηλεμετρίας του ίδιου του Edge Server (π.χ. χρήση CPU, μνήμης κ.λπ.).

Σε αυτό το παραδοτέο θα πρέπει να παραδοθεί και ένα αρχείο **README** το οποίο θα περιέχει αναλυτικά τον τρόπο λειτουργίας του προγράμματος.

Τεχνολογίες συστήματος ανάπτυξης:

- Java Oracle SE 8
- Android API 4.1 ή νεότερο
- Android Studio (Εργαλείο ανάπτυξης της εφαρμογής Android)
- IntelliJ IDEA (Εργαλείο ανάπτυξης της Java εφαρμογής)
- MQTT mosquitto broker
- Eclipse Paho (Java & Android MQTT client)

Κατά τη διαδικασία της ανάπτυξης της εργασίας είναι υποχρεωτική η χρήση του εργαλείου Git (Version Control) και της πλατφόρμας gitlab (anapgit.scanlab.gr) που παραχωρείται για τις ανάγκες του μαθήματος.

Η υλοποίηση της εφαρμογής **θα πρέπει**:

- Να υπακούει στις αρχές του αντικειμενοστραφούς προγραμματισμού
- Να γίνεται σωστή και αποδοτική οργάνωση του κώδικα σε κλάσεις και **πακέτα**.
- Να είναι όσο το δυνατό παραμετροποιήσιμη και δυναμική γίνεται.
- Να γίνεται αποκλειστικά από εσάς **χωρίς την χρήση έτοιμων εργαλείων**. Επιτρέπεται η χρήση δημόσιων βιβλιοθηκών **ΜΟΝΟ με την ύπαρξη αναφοράς στον readme και πάντα σε λογικά πλαίσια**.

Bonus (Προαιρετικό):

1. (+0.5/10) Εγκατάσταση και χρήση εργαλείου της επιλογής σας στις Android συσκευές (όχημα, IoT, drone), με σκοπό τη συλλογή και δημιουργία dataset καταγραφών για τις συσκευές, διάρκειας 7 ημερών. Ένα παράδειγμα τέτοιας εφαρμογής είναι η «[Tucandeira](#)».
2. (+0.5/10) Εγκατάσταση και χρήση του εργαλείου της επιλογής σας στον Edge Server, με σκοπό τη συλλογή ενεργειακών μετρήσεων και δημιουργία dataset για τον server, διάρκειας 7 ημερών. Ένα παράδειγμα τέτοιου εργαλείου είναι το [PowerAPI](#) σε συνδυασμό με το **SmartWatts Formula**