



CTMaaS

**ΤΙΤΛΟΣ**

**«Διαχείριση της Κυκλοφορίας ως Υπηρεσία με την Εφαρμογή  
Συνεργατικών Ευφυών Συστημάτων Μεταφορών»**

**ΑΚΡΩΝΥΜΙΟ: CTMaaS**

Παραδοτέο

**Υλοποίηση επιμέρους υποσυστημάτων**

<b>Αρ. Παραδοτέου</b>	Π2.1
<b>Ενότητα Εργασίας</b>	Σχεδιασμός Συστήματος
<b>Υπεύθυνος Φορέας</b>	ΕΚΕΤΑ-ΙΜΕΤ
<b>Είδος Παραδοτέου</b>	Τελικό
<b>Παράδοση</b>	M16
<b>Σύντομη Περιγραφή</b>	Περιγραφή των επιμέρους υποσυστημάτων της πλατφόρμας CTMaaS.
<b>Έναρξη Προγράμματος, Διάρκεια</b>	8 Οκτωβρίου 2021, 18 Μήνες
<b>Φορείς</b>	ΕΚΕΤΑ-ΙΜΕΤ, Κυκλοφοριακή Τεχνική Α.Ε.
<p><i>Η εργασία υλοποιήθηκε στο πλαίσιο της Δράσης «Επενδυτικά Σχέδια Καινοτομίας» της Περιφέρειας Κεντρικής Μακεδονίας, και συγχρηματοδοτήθηκε από το Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης (ΕΤΠΑ) της Ευρωπαϊκής Ένωσης (ΕΕ) και την Ελλάδα στο πλαίσιο του Επιχειρησιακού Προγράμματος «Κεντρική Μακεδονία 2014-2020».</i></p> <p><i>(Κωδικός πράξης: KMP6-0204048   Κωδικό MIS: 5136509)</i></p>	

31 Ιουλίου 2023

**Λίστα συγγραφέων**

#	Όνοματεπώνυμο	Αρχικά	Φορέας	e-mail
1	Μπουτοβίνας Αντώνης	Μ.Α	ΕΚΕΤΑ-ΙΜΕΤ	<a href="mailto:anmpout@certh.gr">anmpout@certh.gr</a>
2	Σταύρου Κώστας	Σ.Κ.	ΕΚΕΤΑ-ΙΜΕΤ	<a href="mailto:kstavrou@certh.gr">kstavrou@certh.gr</a>
3	Στρωτού Μαρία	Σ.Μ.	ΕΚΕΤΑ-ΙΜΕΤ	<a href="mailto:mstrotou@certh.gr">mstrotou@certh.gr</a>
4	Κώτση Αρετή (contributor)	Κ.Α.	ΕΚΕΤΑ-ΙΜΕΤ	<a href="mailto:akotsi@certh.gr">akotsi@certh.gr</a>
5	Μητσάκης Ευάγγελος (contributor)	Μ.Ε.	ΕΚΕΤΑ-ΙΜΕΤ	<a href="mailto:emit@certh.gr">emit@certh.gr</a>
6	Κλήμη Βασιλεία (contributor)	Κ.Β.	ΕΚΕΤΑ-ΙΜΕΤ	<a href="mailto:vklimi@certh.gr">vklimi@certh.gr</a>
7	Παναγιωτίδης Ιωάννης	Π.Ι.	Κυκλοφοριακή Τεχνική Α.Ε.	<a href="mailto:johnpanagiotidis@gmail.com">johnpanagiotidis@gmail.com</a>
8	Λαζαρίδης Μιλτιάδης	Λ.Μ.	Κυκλοφοριακή Τεχνική Α.Ε.	<a href="mailto:lazaridismiltiadis@yahoo.gr">lazaridismiltiadis@yahoo.gr</a>
9	Καλιμπάνης Δημήτριος	Κ.Δ.	Κυκλοφοριακή Τεχνική Α.Ε.	<a href="mailto:dkaltibanis@traffictech.gr">dkaltibanis@traffictech.gr</a>
10	Αδαμόπουλος Παναγιώτης	Α.Π.	Κυκλοφοριακή Τεχνική Α.Ε.	<a href="mailto:pan.adamo94@gmail.com">pan.adamo94@gmail.com</a>
11	Καλιμπάνη Κλαίρη	Κ.Κ.	Κυκλοφοριακή Τεχνική Α.Ε.	<a href="mailto:klerikaltibani@traffictech.gr">klerikaltibani@traffictech.gr</a>
12	Νέστωρας Εμμανουήλ	Ν.Ε.	Κυκλοφοριακή Τεχνική Α.Ε.	<a href="mailto:nestoras@traffictech.gr">nestoras@traffictech.gr</a>
13	Τζιώγας Χαράλαμπος	Τ.Χ.	Κυκλοφοριακή Τεχνική Α.Ε.	<a href="mailto:tziogas.babis@gmail.com">tziogas.babis@gmail.com</a>

**Πίνακας αλλαγών**

#	Έκδοση	Ημερομηνία	Αλλαγές	Status
1	0.1	19/07/2023	Ενδιάμεση έκδοση	Draft
2	1.0	25/07/2023	Τελική έκδοση	Final



## Κατάλογος συντομογραφιών

Συντομογραφία	Ορισμός
API	Application Programming Interface
C-ITS	Cooperative Intelligent Transport Systems
KPI	Key Performance Indicator
UI	User Interface
IX	Ιδιωτικής Χρήσης



## Περίληψη

Το παρόν παραδοτέο δημιουργήθηκε στο πλαίσιο του έργου C-TMaaS το οποίο αποσκοπεί στην ανάπτυξη ενός ολοκληρωμένου συστήματος δυναμικής διαχείρισης και ελέγχου της κυκλοφορίας. Το παραδοτέο «Π2.1 Υλοποίηση επιμέρους υποσυστημάτων» περιλαμβάνει το σύνολο των υποσυστημάτων που δημιουργήθηκαν στο πλαίσιο ανάπτυξης της πλατφόρμας CTMaaS και της εφαρμογής Mobile App για τους χρήστες - οδηγούς, ενώ περιγράφονται και οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη αυτών. Επίσης αναλύονται και παρουσιάζονται οι τρόποι λειτουργίας των υποσυστημάτων κι οι δυνατότητες που παρέχονται από το σύστημα στους χρήστες, διαχειριστές στόλων οχημάτων και οδηγοί. Η πλατφόρμα CTMaaS αποτελεί επομένως προϊόν της εναρμόνισης και της ενσωμάτωσης όλων των επιμέρους υποσυστημάτων.

## Πίνακας Περιεχομένων

1. Εισαγωγή .....	8
1.2. Δομή Παραδοτέου.....	8
2. Υποσυστήματα .....	9
2.1. Περιβάλλον χρήστη - User Interface .....	9
2.1.1 Στοιχεία οδηγών .....	9
2.1.2 Στοιχεία στόλου οχημάτων .....	10
2.1.3 Απεικόνιση οχημάτων σε χάρτη και πληροφοριών C-ITS .....	11
2.1.4 Στοιχεία προγραμματισμού διαδρομών .....	11
2.1.5 Dashboard πληροφοριών και KPIs .....	12
2.2 Authorization.....	13
2.2.1 IdentityServer4 .....	14
2.3. Διεπαφή προγραμματισμού εφαρμογών (Web API - Application Programming Interface) - Βάση Δεδομένων .....	15
2.3.1 Υλοποίηση βάσης δεδομένων (back end) .....	20
2.4 Routing Engine.....	21
2.5 Vehicle Routing.....	22
2.6 Σύστημα Messaging Broker .....	23
2.6.1 EMQX.....	24
2.7 Mobile App .....	25

## Πίνακας εικόνων

Εικόνα 1: Βάση δεδομένων της πλατφόρμας CTMaaS .....	9
Εικόνα 2: Υποσύστημα στοιχείων οδηγών .....	10
Εικόνα 3: Υποσύστημα για τη δημιουργία προφίλ του στόλου οχημάτων .....	10
Εικόνα 4: Γενικό μενού της πλατφόρμας CTMaaS .....	11
Εικόνα 5: Οπτικοποίηση οχημάτων και C-ITS μηνύματα .....	11
Εικόνα 6: Οπτικοποίηση των διαδρομών των οχημάτων του στόλου .....	12
Εικόνα 7: Αναλυτικά στοιχεία δρομολογίου .....	12
Εικόνα 8: Dashboard για την πλατφόρμα CTMaaS .....	13
Εικόνα 9: Η προσέγγιση “Code First” .....	21
Εικόνα 10: Διάγραμμα ροής MVVM .....	26
Εικόνα 11: MVVM στην πλατφόρμα CTMaaS .....	26

## Πίνακας πινάκων

Πίνακας 1: Request για προσθήκη οχήματος.....	15
Πίνακας 2: Request για ενημέρωση οχήματος.....	16
Πίνακας 3: Request για διαγραφή οχήματος.....	16
Πίνακας 4: Request για ανάκτηση οχημάτων.....	16
Πίνακας 5: Request για ανάκτηση κατασκευαστών.....	16
Πίνακας 6: Request για ανάκτηση μοντέλων των οχημάτων.....	17
Πίνακας 7: Request για ανάκτηση μοντέλων των οχημάτων.....	17
Πίνακας 8: Request για ανάκτηση θέσεων στάθμευσης.....	17
Πίνακας 9: Request για Προθήκη οδηγού.....	17
Πίνακας 10: Request για ενημέρωση οδηγού.....	18
Πίνακας 11: Request για διαγραφή οδηγού.....	18
Πίνακας 12: Request για ανάκτηση οχημάτων.....	18
Πίνακας 13: Request για προσθήκη ανάθεσης οδηγού σε όχημα.....	18
Πίνακας 14: Request για διαγραφή ανάθεσης οδηγού σε όχημα.....	19
Πίνακας 15: Request για προσθήκη διαδρομής.....	19
Πίνακας 16: Request για ενημέρωση οχήματος.....	19
Πίνακας 17: Request για διαγραφή οχήματος.....	19
Πίνακας 18: Request για ανάκτηση διαδρομών.....	20

## 1. Εισαγωγή

Το παρόν παραδοτέο ανήκει στην Ενότητα Εργασίας 2 «Ανάπτυξη Συστήματος» και πιο συγκεκριμένα στη Δράση 2.1. «Ανάπτυξη Υποσυστημάτων» του έργου C-TMaaS. Για την ανάπτυξη των υποσυστημάτων και την ενοποίησή τους χρησιμοποιήθηκαν οι ακόλουθες τεχνολογίες, λαμβάνοντας υπόψη βέλτιστες διεθνείς πρακτικές:

- Angular
- Elastic
- Kibana
- Apache Kafka
- PostgreSQL
- OAuth
- OpenID
- Python/ Anaconda
- SUMO (Simulation of Urban Mobility)

Όλα τα υποσυστήματα που αναπτύχθηκαν είναι σε μορφή Docker, μεγιστοποιώντας έτσι την επεκτασιμότητα, την ταχύτητα και την ασφάλεια, παρέχοντας ταυτόχρονα δυνατότητες συνεχούς εξέλιξης.

Τα βασικά microservices που αναπτύχθηκαν για την πλατφόρμα CTMaaS είναι:

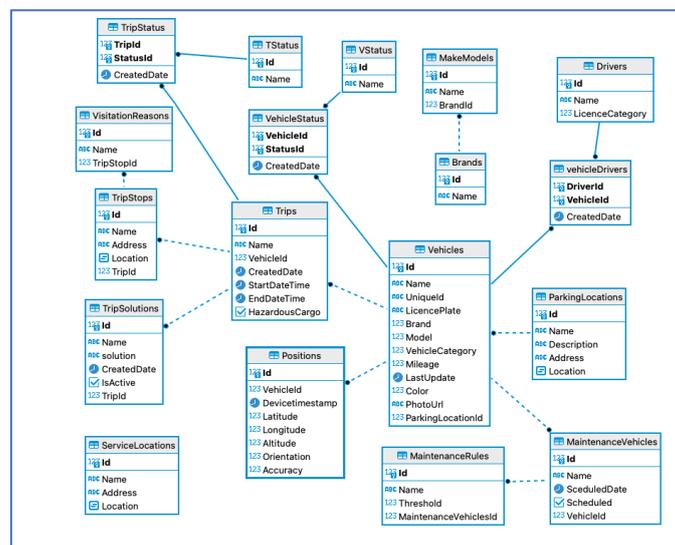
- User Interface
- Authorization
- Web API (Fleet management)
- Routing Engine
- Vehicle Routing
- Broker
- Mobile App

### 1.2. Δομή Παραδοτέου

Το πρώτο κεφάλαιο του παραδοτέου περιλαμβάνει τους στόχους που αναμένεται να επιτευχθούν μετά την ανάπτυξη των υποσυστημάτων. Το δεύτερο κεφάλαιο περιλαμβάνει τα υποσυστήματα που αναπτύχθηκαν και τους τρόπους με τους οποίους αλληλεπιδρούν στο πλαίσιο λειτουργίας της πλατφόρμας CTMaaS.

## 2. Υποσυστήματα

Η πλατφόρμα CTMaaS περιλαμβάνει συστήματα front end (User Interface - UI) για το διαχειριστή του στόλου και για τον οδηγό (Mobile App). Το περιβάλλον του χρήστη είναι συμβατό με το κινητό, τον υπολογιστή ή το tablet. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την υλοποίηση του web interface είναι η Angular. Για το Mobile App έχει χρησιμοποιηθεί το framework του Android και οι γλώσσες προγραμματισμού Java και Kotlin συνδυαστικά. Σχετικά με τη βάση δεδομένων που αναπτύχθηκε, είναι μια σχεσιακή βάση δεδομένων ανοιχτού λογισμικού PostgreSQL. Τέλος για τα υπόλοιπα υποσυστήματα χρησιμοποιήθηκε το framework του .NET core σε συνδυασμό με τη γλώσσα προγραμματισμού C#. Η αρχιτεκτονική της βάσης δεδομένων παρουσιάζεται στην επόμενη εικόνα.



Εικόνα 1: Βάση δεδομένων της πλατφόρμας CTMaaS

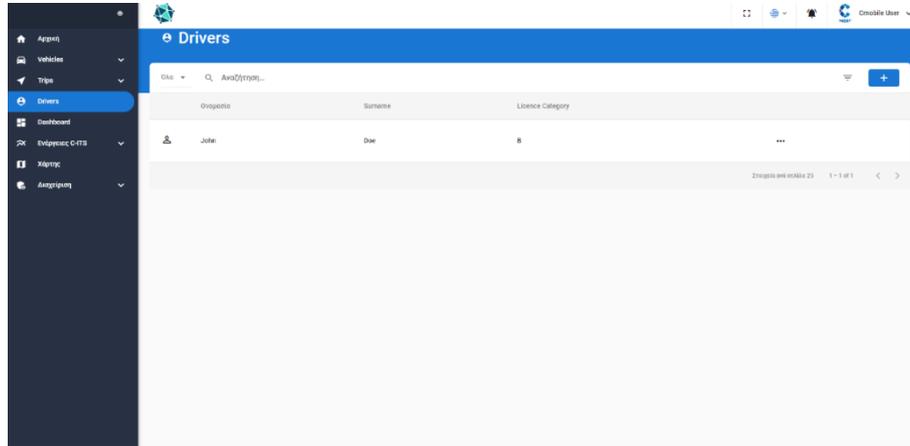
### 2.1. Περιβάλλον χρήστη - User Interface

Το περιβάλλον χρήστη (UI) αφορά σε όλες τις ενέργειες που κάνει ο χρήστης για να αλληλοεπιδρά με την πλατφόρμα ή την εφαρμογή. Ο όρος UI αναφέρεται σε όλα εκείνα τα χαρακτηριστικά μιας εφαρμογής που έχουν να κάνουν με το σχεδιασμό της ώστε να είναι ελκυστική στο χρήστη. Τα υποσυστήματα που αφορούν στο UI είναι τα ακόλουθα:

- Στοιχεία οδηγών.
- Στοιχεία στόλου οχημάτων.
- Χάρτης οχημάτων και πληροφοριών C-ITS.
- Στοιχεία προγραμματισμού διαδρομών του στόλου.
- Dashboard πληροφοριών και KPIs σχετικά με τον στόλο και τις διαδρομές αυτού.

#### 2.1.1 Στοιχεία οδηγών

Σε αυτό το υποσύστημα υπάρχουν όλες οι λειτουργίες εισαγωγής, τροποποίησης και διαγραφής οδηγών στο σύστημα. Μια επιπλέον λειτουργικότητα που υπάρχει στο παρόν υποσύστημα είναι αυτή της ανάθεσης οδηγού σε όχημα.

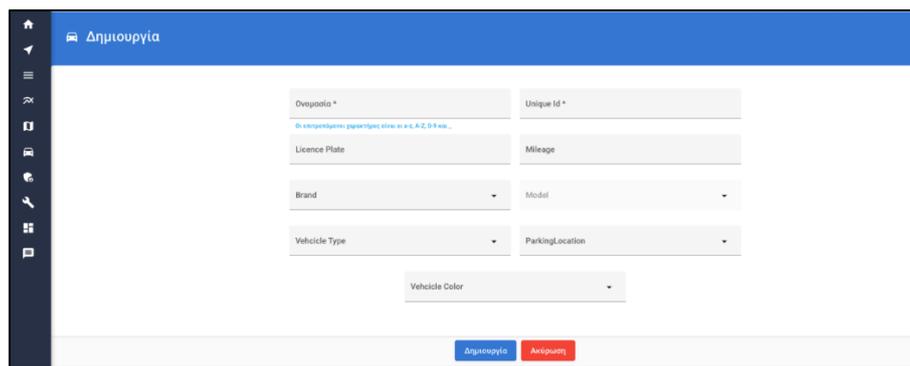


Εικόνα 2: Υποσύστημα στοιχείων οδηγών

### 2.1.2 Στοιχεία στόλου οχημάτων

Το δεύτερο υποσύστημα που αναπτύχθηκε περιλαμβάνει τη λειτουργία εισαγωγής οχημάτων του στόλου στο σύστημα. Αναπτύχθηκαν πεδία συμπλήρωσης των στοιχείων των οχημάτων όπως παρουσιάζεται στην παρακάτω εικόνα. Πιο αναλυτικά τα πεδία που καλείται να συμπληρώσει ο χρήστης είναι:

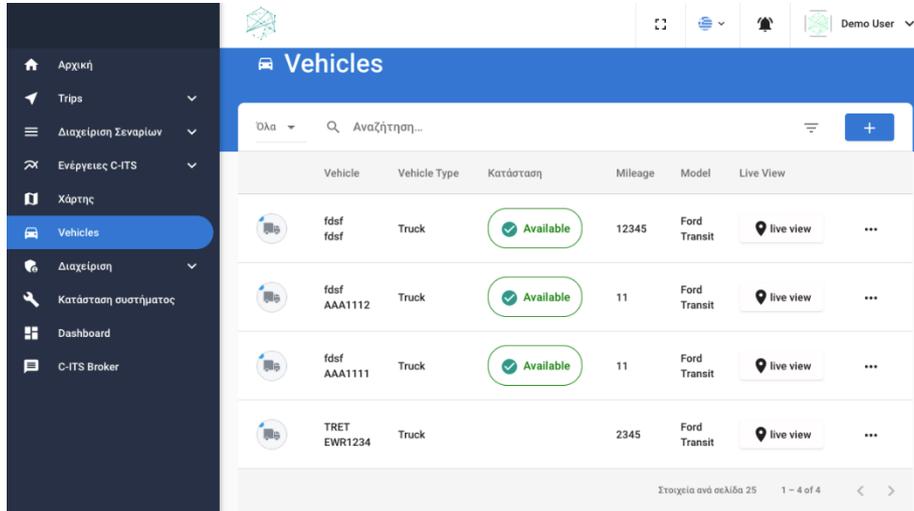
- Ονομασία και κωδικός του οχήματος.
- Πινακίδα, τύπος και χρώμα οχήματος.
- Σημείο στάθμευσης.



Εικόνα 3: Υποσύστημα για τη δημιουργία προφίλ του στόλου οχημάτων

Επιπλέον στο παρόν υποσύστημα υπάρχει μια λίστα με το σύνολο των οχημάτων του στόλου καθώς και συνοπτικές πληροφορίες για το κάθε ένα. Οι πληροφορίες αυτές είναι ο κωδικός του οχήματος, το είδος του οχήματος (π.χ. Ι.Χ, φορτηγό), η κατάσταση του (διαθέσιμο ή όχι), τα χιλιόμετρα του οχήματος. Σε αυτό το πεδίο υπάρχει επίσης σύνδεσμος που εμφανίζει σε χάρτη την απεικόνιση κάθε οχήματος σε πραγματικό χρόνο προκειμένου ο διαχειριστής του στόλου να μπορεί να εποπτεύει την ακριβή θέση του κάθε οχήματος.

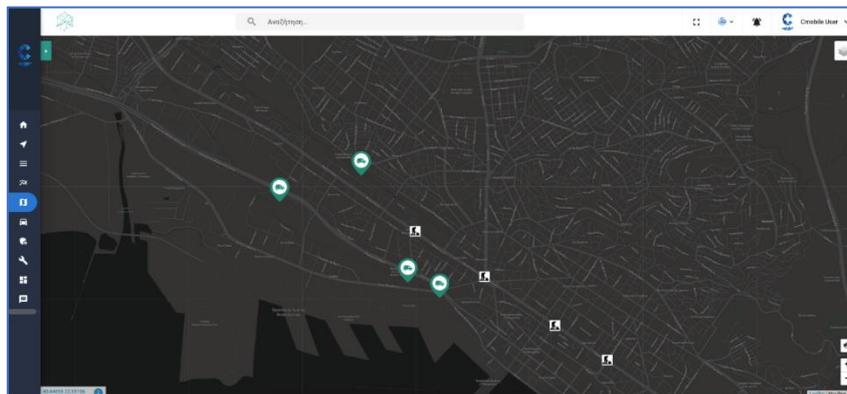
Η παρακάτω εικόνα παρουσιάζει το γενικό μενού της πλατφόρμας CTMaas όπως αυτό εμφανίζεται στο χρήστη.



Εικόνα 4: Γενικό μενού της πλατφόρμας CTMaas

### 2.1.3 Απεικόνιση οχημάτων σε χάρτη και πληροφοριών C-ITS

Τα στίγματα των οχημάτων εμφανίζονται με μορφή pin σε υπόβαθρο χάρτη (basemap) και οπτικοποιείται η διαδρομή τους σε πραγματικό χρόνο. Επιπλέον ο χρήστης του συστήματος μπορεί να επιλέξει και να δει στο χάρτη και C-ITS μηνύματα στην περιοχή κίνησης του στόλου του.

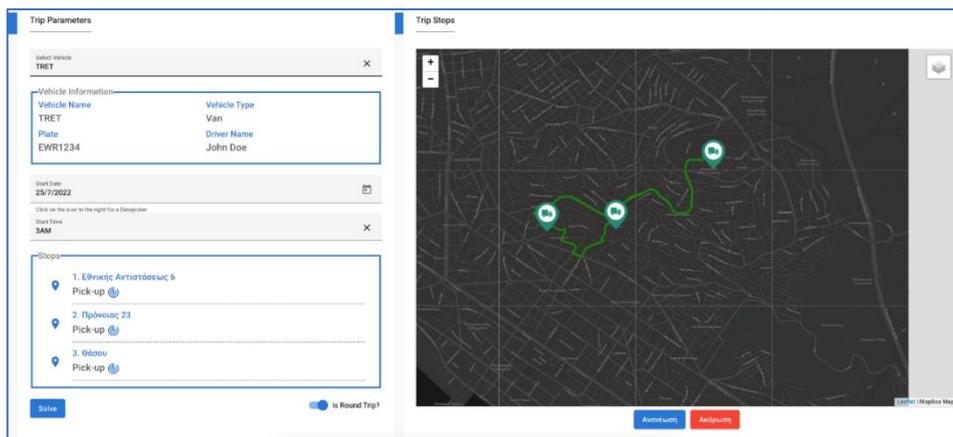


Εικόνα 5: Οπτικοποίηση οχημάτων και C-ITS μηνύματα

### 2.1.4 Στοιχεία προγραμματισμού διαδρομών

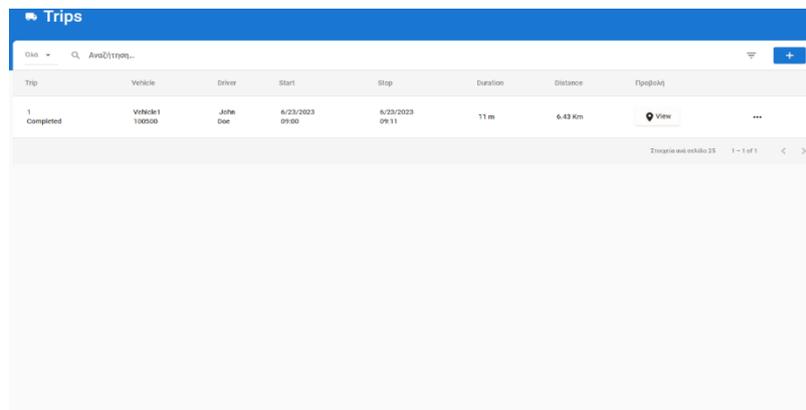
Μια από τις βασικές δυνατότητες που προσφέρει η πλατφόρμα CTMaas είναι ο προγραμματισμός δρομολόγησης (routing και re-routing) των οχημάτων και η ενημέρωση των οδηγών για το είδος της εργασίας που πρέπει να γίνει στο σημείο - προορισμό (π.χ.

παραλαβή, παράδοση). Στο πεδίο “Trip Parameters” εισάγονται τα στοιχεία των προορισμών και δημιουργείται - οπτικοποιείται στο χάρτη το δρομολόγιο των οχημάτων. Εμφανίζονται επίσης τα στοιχεία των οχημάτων, η ημερομηνία και η ώρα αναχώρησης του οχήματος από το σημείο εκκίνησης, καθώς και όλοι οι προορισμοί τους (διεύθυνση και είδος εργασίας στον προορισμό, π.χ. pick up - drop off). Στη συνέχεια καλείται το υποσύστημα Vehicle routing engine και υπολογίζει τη βέλτιστη διαδρομή που πρέπει να ακολουθήσει το όχημα. Αν ο χρήστης αποφασίσει ότι η λύση είναι ικανοποιητική τότε μπορεί να αποθηκεύσει το συγκεκριμένο δρομολόγιο. Το υποσύστημα δίνει τη δυνατότητα στο χρήστη να τροποποιήσει τις διαδρομές που δεν είναι προγραμματισμένες σε μελλοντικό χρόνο.



Εικόνα 6: Οπτικοποίηση των διαδρομών των οχημάτων του στόλου

Σε αυτό το υποσύστημα ο χρήστης μπορεί να δει όλα τα δρομολόγια που υπάρχουν στο σύστημα. Επιπλέον μπορεί να χρησιμοποιήσει τα φίλτρα ώστε να εντοπίσει ευκολότερα το δρομολόγιο ή δρομολόγια που τον ενδιαφέρουν.

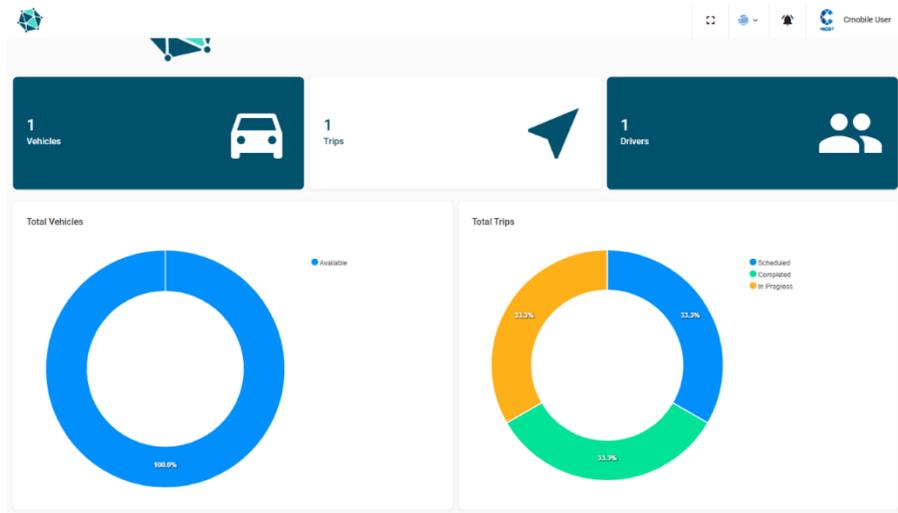


Εικόνα 7: Αναλυτικά στοιχεία δρομολογίου

### 2.1.5 Dashboard πληροφοριών και KPIs

Το Dashboard αποτελεί μια οπτική αναπαράσταση βασικών πληροφοριών και δεδομένων και παράλληλα παρέχει μια επισκόπηση σημαντικών μετρήσεων και πληροφοριών σε ενοποιημένη και εύκολα κατανοητή μορφή. Παρουσιάζει δεδομένα από διάφορες πηγές και

συστήματα με οπτικά ελκυστικό και διαδραστικό τρόπο, επιτρέποντας στο χρήστη του συστήματος να αντιλαμβάνεται γρήγορα τις τάσεις, τα μοτίβα και τις κρίσιμες πληροφορίες σχετικά με το στόλο. Στο Dashboard υπάρχουν οπτικά στοιχεία όπως διαγράμματα, γραφήματα και πίνακες για να παρουσιάσουν τα δεδομένα με οπτικά ελκυστικό τρόπο. Οι οπτικοποιήσεις διευκολύνουν την κατανόηση πολύπλοκων δεδομένων και το γρήγορο εντοπισμό τάσεων ή ανωμαλιών στη λειτουργία του στόλου. Στο Dashboard υπάρχουν επιπλέον βασικοί δείκτες απόδοσης KPIs οι οποίοι αφορούν σε συγκεκριμένες μετρήσεις ή μέτρα.



Εικόνα 8: Dashboard για την πλατφόρμα CTMaas

## 2.2 Authorization

Το υποσύστημα Authorization (εξουσιοδότηση) είναι μέρος του ευρύτερου συστήματος. Βασικός του ρόλος είναι ο έλεγχος της ταυτότητας και η εξουσιοδότηση των χρηστών. Αρχικά επαληθεύει την ταυτότητα των χρηστών και προσδιορίζει αν έχουν τα απαραίτητα δικαιώματα πρόσβασης σε συγκεκριμένους πόρους ή για την εκτέλεση συγκεκριμένων ενεργειών εντός του συστήματος. Το υποσύστημα Authorization συνήθως λειτουργεί σε συνδυασμό με άλλα υποσύστημα ή στοιχεία του συστήματος για την επιβολή πολιτικών ασφαλείας και ελέγχων πρόσβασης. Παρέχει μια κεντρική και επεκτάσιμη λύση για τη διαχείριση των δικαιωμάτων πρόσβασης των χρηστών σε διάφορα μέρη της πλατφόρμας. Παρακάτω περιγράφονται ορισμένες βασικές λειτουργίες και δυνατότητες που μπορεί να παρέχει το υποσύστημα Authorization.

- Έλεγχος ταυτότητας χρήστη: Το υποσύστημα χειρίζεται τη διαδικασία ελέγχου ταυτότητας επαληθεύοντας την ταυτότητα των χρηστών με βάση τα διαπιστευτήριά τους όπως το όνομα χρήστη και το κωδικό πρόσβασης. Μπορεί να υποστηρίξει διάφορους μηχανισμούς ελέγχου ταυτότητας όπως όνομα χρήστη/ κωδικό πρόσβασης, διακριτικά ή πρωτόκολλα Single Sign-On (SSO).
- Πολιτικές εξουσιοδότησης: Επιτρέπει τον καθορισμό και την επιβολή πολιτικών εξουσιοδότησης. Οι πολιτικές καθορίζουν τους όρους υπό τους οποίους χορηγείται ή απαγορεύεται σε ένα χρήστη η πρόσβαση σε συγκεκριμένους πόρους. Οι πολιτικές

μπορούν να βασίζονται σε διάφορους παράγοντες, όπως χαρακτηριστικά χρήστη, ρόλους και δικαιώματα.

- Access Token Management: Μπορεί να εκδίδει και να διαχειρίζεται διακριτικά πρόσβασης (tokens) που χρησιμοποιούνται για τον έλεγχο ταυτότητας και την εξουσιοδότηση αιτημάτων API από άλλα υποσυστήματα. Αυτά τα διακριτικά είναι συνήθως βραχύβια και μπορούν να ανανεωθούν ή να ανακληθούν όταν είναι απαραίτητο.
- Έλεγχος και καταγραφή: Μπορεί να καταγράφει συμβάντα που σχετίζονται με την εξουσιοδότηση και να παρέχει δυνατότητες ελέγχου για την παρακολούθηση της πρόσβασης των χρηστών και των αλλαγών αδειών. Αυτό βοηθά στην παρακολούθηση και στη διερεύνηση περιστατικών που σχετίζονται με την ασφάλεια του ευρύτερου συστήματος.

Το υποσύστημα Authorization διαδραματίζει κρίσιμο ρόλο στην ασφάλεια και στην ακεραιότητα ενός συστήματος λογισμικού χάρη στη διαχείριση του ελέγχου ταυτότητας χρήστη και στον έλεγχο πρόσβασης. Συγκεντρώνοντας αυτές τις λειτουργίες σε μια αποκλειστική υπηρεσία προωθεί την επεκτασιμότητα εντός της ευρύτερης αρχιτεκτονικής του συστήματος.

### 2.2.1 IdentityServer4

Για την υλοποίηση του υποσυστήματος Authorization επιλέχθηκε το έργο IdentityServer4. Το έργο IdentityServer4 είναι ένα έργο ανοικτού κώδικα το οποίο παρέχει μια ευέλικτη και βασισμένη σε πρότυπα λύση για την υλοποίηση ελέγχου ταυτότητας, εξουσιοδότησης και διαχείρισης ταυτότητας σε εφαρμογές .NET. Έχει σχεδιαστεί να ακολουθεί πρότυπα και πρωτόκολλα τα οποία είναι ευρέως διαδεδομένα, όπως το OAuth 2.0 και το OpenID Connect, καθιστώντας το συμβατό με ένα ευρύ φάσμα εφαρμογών.

Το έργο IdentityServer4 παρέχει μια ασφαλή και συγκεντρωτική λύση για τη διαχείριση του ελέγχου ταυτότητας και της εξουσιοδότησης χρηστών σε κατακεντρωμένα συστήματα. Ενεργεί ως πάροχος ταυτότητας επιτρέποντας στους χρήστες να πιστοποιούνται με διάφορα διαπιστευτήρια και να λαμβάνουν διακριτικά ταυτότητας και πρόσβασης. Οι clients, οι οποίοι μπορεί να είναι εφαρμογές ιστού, εφαρμογές για κινητά ή API, εγγράφονται στον IdentityServer4 και λαμβάνουν αναγνωριστικά client και secret keys για να αλληλοεπιδρούν μεταξύ τους.

Το έργο IdentityServer4 υποστηρίζει διάφορους τρόπους αποθήκευσης χρηστών, συμπεριλαμβανομένων βάσεων δεδομένων, LDAP και εξωτερικών παρόχων ταυτότητας. Στο παρόν έργο έχει επιλεγεί η αποθήκευση των χρηστών σε βάση δεδομένων. Εκδίδει διαφορετικούς τύπους tokens, συμπεριλαμβανομένων των tokens ταυτότητας, των access tokens και των refresh tokens, για να επιτρέπει την ασφαλή και εξουσιοδοτημένη πρόσβαση σε προστατευόμενους πόρους.

Η ενσωμάτωση με το ASP.NET Core απλοποιεί την ενσωμάτωση του IdentityServer4 στην πλατφόρμα CTMaaS, παρέχοντας ενδιάμεσο λογισμικό και υπηρεσίες για το χειρισμό του ελέγχου ταυτότητας και της εξουσιοδότησης. Το IdentityServer4 είναι εξαιρετικά επεκτάσιμο,

επιτρέποντας την προσαρμογή και την ενσωμάτωση πρόσθετων χαρακτηριστικών με βάση τις συγκεκριμένες απαιτήσεις της εφαρμογής. Δίνει έμφαση στις βέλτιστες πρακτικές ασφάλειας και παρέχει προστασία από κοινές ευπάθειες ασφαλείας.

Όπως προαναφέρθηκε, το έργο είναι ανοικτού κώδικα, φιλοξενείται στο GitHub και διαθέτει μια ενεργή κοινότητα προγραμματιστών που συμβάλλει στην ανάπτυξη και υποστήριξή του.

### 2.3. Διεπαφή προγραμματισμού εφαρμογών (Web API - Application Programming Interface) - Βάση Δεδομένων

Η διεπαφή προγραμματισμού εφαρμογών (Web API) αποτελεί τη μέθοδο εισαγωγής των στοιχείων - δεδομένων στη βάση χρησιμοποιώντας πρωτόκολλο HTTP. Η βασική λειτουργία του είναι η δυνατότητα διασύνδεσης της βάση δεδομένων με τρίτα συστήματα (π.χ. web applications, mobile apps) με στόχο να μπορούν να διαβαστούν, τροποποιηθούν, ή και να διαγραφούν ήδη αποθηκευμένα στοιχεία είτε να εισαχθούν νέα στη βάση. Στην πλατφόρμα CTMaaS εξυπηρετείται με αυτόν τον τρόπο η διασύνδεση του front end με τη βάση δεδομένων (back end).

Για να επιτυγχάνεται η ορθή λειτουργία του Web API και να διευκολύνεται η προβολή, αναζήτηση και αλλαγή πληροφοριών, το Web API πρέπει να καλύπτει 4 βασικές λειτουργίες:

1. Να δημιουργεί
2. Να διαβάζει
3. Να ενημερώνει
4. Να διαγράφει στοιχεία από και προς την βάση δεδομένων (CRUD).

Το Web API που αναπτύχθηκε ακολουθεί τις αρχές και τους κανόνες της τεχνικής REST η οποία δεν αποτελεί δεσμευτικό πρωτόκολλο/ πρότυπο και μπορεί να υλοποιηθεί με διαφορετικούς τρόπους.

Συγκεκριμένα στο σύστημα CTMaaS αναπτύχθηκαν τα ακόλουθα requests τα οποία καλύπτουν τις ανάγκες και τις προδιαγραφές του συστήματος όπως αυτές περιγράφονται στο παραδοτέο Π1.1.

Πίνακας 1: Request για προσθήκη οχήματος.

<b>Προσθήκη οχήματος</b>	
Url:/addVehicle	Method: Post
Παράμετροι: JSON Παράδειγμα { "name":"fdsf", "uniqueId":"111", "licencePlate":"AAA1111", "mileage":11, "brand":482, "model":2, "vehicleType":3, "parkingLocationId":2, "color":8, "photoUrl":"" }	Response: Text
<b>Περιγραφή</b>	
Η μέθοδος αποθηκεύει ένα νέο όχημα στη βάση δεδομένων του συστήματος.	

Πίνακας 2: Request για ενημέρωση οχήματος

<b>Ενημέρωση οχήματος</b>	
Url:/updateVehicle	Method: Put
Παράμετροι: JSON Παράδειγμα { "name":"fdsf","uniqueId":"111","licencePlate":"AAA1111", "mileage":11,"brand":482,"model":2,"vehicleType":3, "parkingLocationId":2,"color":8,"photoUrl":"","id":8 }	Response: Text
<b>Περιγραφή</b>	
Η μέθοδος ενημερώνει ένα υπάρχον όχημα στη βάση δεδομένων του συστήματος.	

Πίνακας 3:Request για διαγραφή οχήματος

<b>Διαγραφή οχήματος</b>	
Url:/deleteVehicle/{Id:int}	Method: Delete
Παράμετροι: url parameter Παράδειγμα /deleteVehicle/1	Response: Text
<b>Περιγραφή</b>	
Η μέθοδος διαγράφει ένα υπάρχον όχημα στην βάση δεδομένων του συστήματος.	

Πίνακας 4:Request για ανάκτηση οχημάτων

<b>Ανάκτηση οχημάτων</b>	
Url:/getAllVehicles?IncludeAssignment={boolean}	Method: Get
Παράμετροι: url parameter Παράδειγμα getAllVehicles?IncludeAssignment=true	Response: JSON
<b>Περιγραφή</b>	
Η μέθοδος ανακτά από την βάση δεδομένων όλα τα οχήματα που υπάρχουν σε αυτήν. Επιπλέον η παράμετρος <i>IncludeAssignment</i> χρησιμοποιείται για να ενσωματώσει ή όχι τις πληροφορίες που αφορούν τις αναθέσεις οχημάτων σε οδηγούς.	

Πίνακας 5:Request για ανάκτηση κατασκευαστών

<b>Ανάκτηση κατασκευαστών</b>	
Url:/getAllBrands	Method: Get
Παράμετροι:	Response: JSON

<b>Περιγραφή</b>	
Η μέθοδος ανακτά από την βάση δεδομένων όλους τους κατασκευαστές οχημάτων.	

Πίνακας 6: Request για ανάκτηση μοντέλων των οχημάτων

<b>Ανάκτηση μοντέλων των οχημάτων</b>	
Url: /getAllBrandsModels?brandId={int}	Method: Get
Παράμετροι: url parameter Παράδειγμα getAllBrandsModels?brandId=1	Response: JSON
<b>Περιγραφή</b>	
Η μέθοδος ανακτά από την βάση δεδομένων όλα τα μοντέλα οχημάτων βάση του Id του κατασκευαστή.	

Πίνακας 7: Request για ανάκτηση μοντέλων των οχημάτων

<b>Ανάκτηση μοντέλων των οχημάτων</b>	
Url: /getAllModelsBrandName?brandName={string}	Method: Get
Παράμετροι: url parameter Παράδειγμα getAllModelsBrandName? brandName=Fiat	Response: JSON
<b>Περιγραφή</b>	
Η μέθοδος ανακτά από την βάση δεδομένων όλα τα μοντέλα οχημάτων βάση του ονόματος του κατασκευαστή.	

Πίνακας 8: Request για ανάκτηση θέσεων στάθμευσης

<b>Ανάκτηση θέσεων στάθμευσης</b>	
Url: /getAllParkingLocations	Method: Get
Παράμετροι:	Response: JSON
<b>Περιγραφή</b>	
Η μέθοδος ανακτά από την βάση δεδομένων όλα τις προκαθορισμένες θέσεις στάθμευσης των οχημάτων.	

Πίνακας 9: Request για Προσθήκη οδηγού

<b>Προσθήκη οδηγού</b>	
Url: /addDriver	Method: Post
Παράμετροι: JSON	Response: Text

<b>Παράδειγμα</b> <pre>{ "name": "DDD", "surName": "DDD",   "licenceCategory": "1" }</pre>	
<b>Περιγραφή</b> Η μέθοδος αποθηκεύει ένα νέο οδηγό στην βάση δεδομένων του συστήματος.	

Πίνακας 10: Request για ενημέρωση οδηγού

<b>Ενημέρωση οδηγού</b>	
Url: /updateDriver	Method: Put
Παράμετροι: JSON Παράδειγμα <pre>{ "name": "DDD", "surName": "DDD", "licenceCategory": "1",   "id": -1 }</pre>	Response: Text
<b>Περιγραφή</b> Η μέθοδος ενημερώνει έναν υπάρχον οδηγό στην βάση δεδομένων του συστήματος.	

Πίνακας 11: Request για διαγραφή οδηγού

<b>Διαγραφή οδηγού</b>	
Url: /deleteDriver/{Id:int}	Method: Delete
Παράμετροι: url parameter Παράδειγμα /deleteDriver/1	Response: Text
<b>Περιγραφή</b> Η μέθοδος διαγράφει ένα υπάρχον οδηγό στην βάση δεδομένων του συστήματος.	

Πίνακας 12: Request για ανάκτηση οχημάτων

<b>Ανάκτηση οχημάτων</b>	
Url: /getAllDrivers	Method: Get
Παράμετροι: url parameter Παράδειγμα getAllDrivers	Response: JSON
<b>Περιγραφή</b> Η μέθοδος ανακτά από την βάση δεδομένων όλους τους οδηγούς που υπάρχουν σε αυτήν.	

Πίνακας 13: Request για προσθήκη ανάθεσης οδηγού σε όχημα

<b>Προσθήκη ανάθεσης οδηγού σε όχημα</b>	
Url: addVehicleDriverRelationship	Method: Post
Παράμετροι: JSON Παράδειγμα <pre>{ "driverId":10, "vehicleId":6, "startDate":"2022-06-23T09:28:20.242Z", "endDate":"2022-12-23T09:28:20.242Z" }</pre>	Response: Text
<b>Περιγραφή</b>	
Η μέθοδος προσθέτει μία ανάθεση ενός οδηγού σε ένα όχημα.	

Πίνακας 14: Request για διαγραφή ανάθεσης οδηγού σε όχημα

<b>Διαγραφή ανάθεσης οδηγού σε όχημα</b>	
Url: /deleteVehicleDriverRelationship/{VehicleId:int}/{DriverId:int}	Method: Delete
Παράμετροι: url parameter Παράδειγμα /deleteVehicleDriverRelationship/1/2	Response: Text
<b>Περιγραφή</b>	
Η μέθοδος διαγράφει προσθέτει μία ανάθεση ενός οδηγού σε ένα όχημα.	

Πίνακας 15: Request για προσθήκη διαδρομής

<b>Προσθήκη Διαδρομής</b>	
Url: /addTrip	Method: Post
	Response: Text
<b>Περιγραφή</b>	
Η μέθοδος αποθηκεύει μια νέα διαδρομή στην βάση δεδομένων του συστήματος.	

Πίνακας 16: Request για ενημέρωση οχήματος

<b>Ενημέρωση οχήματος</b>	
Url: /updateTrip	Method: Put
Παράμετροι: JSON Παράδειγμα	Response: Text
<b>Περιγραφή</b>	
Η μέθοδος ενημερώνει μια υπάρχουσα διαδρομή στην βάση δεδομένων του συστήματος.	

Πίνακας 17: Request για διαγραφή οχήματος

<b>Διαγραφή οχήματος</b>	
Url: /deleteTrip/{Id:int}	Method: Delete
Παράμετροι: url parameter Παράδειγμα /deleteTrip/1	Response: Text
<b>Περιγραφή</b>	
Η μέθοδος διαγράφει μια υπάρχουσα διαδρομή στην βάση δεδομένων του συστήματος.	

Πίνακας 18: Request για ανάκτηση διαδρομών

<b>Ανάκτηση διαδρομών</b>	
Url: /getAllTrips	Method: Get
Παράμετροι: url parameter	Response: JSON
<b>Περιγραφή</b>	
Η μέθοδος ανακτά από την βάση δεδομένων όλες τις διαδρομές που υπάρχουν σε αυτήν.	

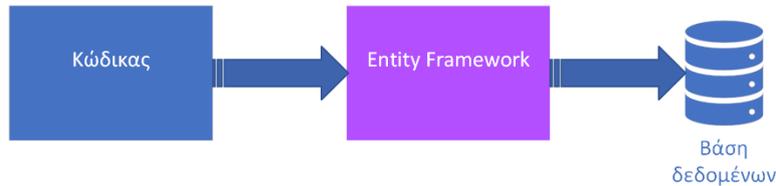
### 2.3.1 Υλοποίηση βάσης δεδομένων (back end)

Για την υλοποίηση της βάσης δεδομένων της πλατφόρμας CTMaaS έχει χρησιμοποιηθεί η τεχνική PostgreSQL. Η PostgreSQL είναι μια σχεσιακή βάση δεδομένων ανοικτού κώδικα με πολλές δυνατότητες που μπορεί να καλύψει τις ανάγκες του υπό ανάπτυξη συστήματος. Επιπρόσθετα με την PostgreSQL χρησιμοποιείται το PostGIS που είναι ένα εργαλείο επίσης ανοικτού κώδικα το οποίο επιτρέπει τη διαχείριση γεωγραφικών πληροφοριών στη βάση δεδομένων. Το εργαλείο PostGIS ακολουθεί το πρότυπο Simple Features το οποίο καθορίζει ένα κοινό μοντέλο αποθήκευσης και πρόσβασης γεωγραφικών χαρακτηριστικών που αποτελούνται κυρίως από δισδιάστατες γεωμετρίες (π.χ. σημείο, γραμμή, πολύγωνο, πολλαπλών σημείων, πολλαπλών γραμμών).

Η βάση δεδομένων της πλατφόρμας CTMaaS (Εικόνα 1) έχει υλοποιηθεί με το εργαλείο ανοικτού κώδικα Entity Framework Core. Το Entity Framework Core είναι ένα εργαλείο object - relational mapping (ORM) το οποίο υποστηρίζει την ανάπτυξη εφαρμογών προσανατολισμένων στα δεδομένα. Οι προγραμματιστές τέτοιου είδους εφαρμογών πρέπει να μοντελοποιήσουν τις οντότητες, τις σχέσεις καθώς και τη λογική των προβλημάτων που επιλύουν και πρέπει επίσης να διασυνδεθούν με το σύστημα RDBMS που χρησιμοποιείται για την αποθήκευση και την ανάκτηση των δεδομένων. Τα δεδομένα μπορούν να εκτείνονται σε πολλαπλά συστήματα RDBMS, το καθένα με τα δικά του πρωτόκολλα.

Τα εργαλεία object - relational mapping έχουν αναπτυχθεί για να επιτρέπουν στους προγραμματιστές να εργάζονται με δεδομένα με τη μορφή αντικειμένων, χωρίς να χρειάζεται να ανησυχούν για τους πίνακες της βάσης δεδομένων όπου αποθηκεύονται αυτά τα δεδομένα. Συγκεκριμένα στην παρούσα υλοποίησης βάσης δεδομένων ακολουθήθηκε η προσέγγιση “Code First” στην οποία αναπτύσσονται όλες οι κλάσεις των οντοτήτων καθώς

και των σχέσεων τους και στη συνέχεια με την object model mapper (Entity Framework Core) αντιστοιχίζονται σε πίνακες στη βάση δεδομένων. Όλοι οι πίνακες και οι σχέσεις στη βάση δεδομένων δημιουργούνται αυτόματα.



Εικόνα 9: Η προσέγγιση “Code First”

## 2.4 Routing Engine

Ένα ακόμη υποσύστημα που αναπτύχθηκε για την επιτυχή λειτουργία της πλατφόρμας και της εφαρμογής, είναι το microservice “Routing Engine” που είναι υπεύθυνο για τη δρομολόγηση των οχημάτων με το βέλτιστο τρόπο. Ένα routing engine είναι ένα σύστημα λογισμικού που υπολογίζει και καθορίζει τη βέλτιστη διαδρομή. Αναλύει διάφορους παράγοντες, όπως την τοπολογία του δικτύου, τους περιορισμούς και τις προτιμήσεις, για να καθορίσει την καλύτερη δυνατή διαδρομή από μια αφετηρία σε έναν προορισμό.

Ένα τέτοιο λογισμικό χρησιμοποιεί αλγορίθμους και δομές δεδομένων για τον υπολογισμό της πιο αποδοτικής διαδρομής ή μιας διαδρομής μεταξύ των σημείων αφετηρίας και προορισμού. Λαμβάνει υπόψη παράγοντες όπως η απόσταση, το κόστος, ο χρόνος ταξιδιού, οι συνθήκες κυκλοφορίας, η διαθεσιμότητα του δικτύου και άλλοι πιθανοί περιορισμοί. Αναλύει την τοπολογία του δικτύου η οποία περιλαμβάνει τους κόμβους (nodes) και τους συνδέσμους (links) που είναι οι συνδέσεις μεταξύ των κόμβων. Χρησιμοποιεί αυτές τις πληροφορίες για τον προσδιορισμό πιθανών διαδρομών και τη λήψη αποφάσεων δρομολόγησης.

Το Routing engine υλοποιεί συχνά πρωτόκολλα δρομολόγησης, όπως το OSPF (Open Shortest Path First), το BGP (Border Gateway Protocol) ή το RIP (Routing Information Protocol), για την ανταλλαγή πληροφοριών δρομολόγησης και τον προσδιορισμό των καλύτερων διαδρομών σε ένα δίκτυο. Παρακολουθεί συνεχώς το δίκτυο και λαμβάνει ενημερώσεις. Προσαρμόζει τους πίνακες δρομολόγησης και υπολογίζει εκ νέου τις διαδρομές δυναμικά με βάση τις αλλαγές στις συνθήκες του δικτύου, τις αποτυχίες συνδέσεων ή τις νέες συσκευές δικτύου που προστίθενται ή αφαιρούνται.

Σε σενάρια μεταφορών ή logistics, μια μηχανή δρομολόγησης μπορεί να βελτιστοποιήσει τις διαδρομές με βάση δεδομένα κυκλοφορίας σε πραγματικό χρόνο, οδικές συνθήκες, επίπεδα συμφόρησης ή άλλους παράγοντες. Στοχεύει στην ελαχιστοποίηση του χρόνου ταξιδιού, της κατανάλωσης καυσίμων ή του κόστους παράδοσης, επιλέγοντας την καταλληλότερη διαδρομή για τα οχήματα ή τα αγαθά.

Για το συγκεκριμένο microservice χρησιμοποιήθηκε η Μηχανή Δρομολόγησης Ανοικτού Κώδικα, OSRM η οποία σε υψηλή απόδοση βρίσκει τις συντομότερες διαδρομές στα οδικά δίκτυα. Πρόκειται για ένα εργαλείο που μπορεί να χρησιμοποιηθεί για τον υπολογισμό της

ταχύτερης ή συντομότερης διαδρομής μεταξύ δύο σημείων στο χάρτη και για την ανάλυση ολόκληρων δικτύων μεταφορών. Ο σχεδιασμός της διαδρομής, η πλοήγηση και η ανάλυση δικτύου παρέχονται διαδραστικά στο χρήστη. Το OSRM βασίζεται στο σύνολο των δεδομένων του Open Street Map (OSM) και χρησιμοποιεί μια αποτελεσματική δομή γραφήματος για να προσδιορίσει σε κλάσματα του δευτερολέπτου την καλύτερη διαδρομή για ένα δεδομένο σημείο «Αφετηρίας της διαδρομής» και «Προορισμού». Ενημερώνει τον χρήστη για τη χιλιομετρική απόσταση που πρέπει να διανύσει καθώς και για την ώρα άφιξης, υπολογίζοντας πάντα τις κυκλοφοριακές συνθήκες στο οδικό δίκτυο. Στο πεδίο των logistics και του σχεδιασμού μεταφορών και κινητικότητας, το OSRM είναι ένα πολύ χρήσιμο εργαλείο.

Ο τρόπος χρήσης του OSRM για την πλατφόρμα CTMaaS διαφέρει από τη συνήθη χρήση του. Χρησιμοποιείται ως calculation engine του Vehicle routing που αναφέρεται παρακάτω.

## 2.5 Vehicle Routing

Για την πιο αποτελεσματική δρομολόγηση των οχημάτων, η πλατφόρμα CTMaaS επιπρόσθετα του OSRM, χρησιμοποιεί το Vehicle Routing Open-source Optimization Machine - VROOM. Το VROOM είναι μια μηχανή βελτιστοποίησης ανοιχτού κώδικα γραμμένη σε C++17 που βασίζεται και χρησιμοποιεί τα δεδομένα του OpenStreetMap και στοχεύει στην παροχή λύσεων σε διάφορα προβλήματα δρομολόγησης οχημάτων (VRP) σε μικρό χρόνο.

Το VROOM μπορεί να δίνει λύσεις σε πολλούς γνωστούς τύπους προβλημάτων δρομολόγησης οχημάτων (VRP) ή και συνδυασμό αυτών όπως:

- TSP (travelling salesman problem)
- CVRP (capacitated VRP)
- VRPTW (VRP with time windows)
- MDHVRPTW (multi-depot heterogeneous vehicle VRPTW)
- PDPTW (pickup-and-delivery problem with TW)

Κάθε πρόβλημα δρομολόγησης μοντελοποιείται από το VROOM περιγράφοντας τα οχήματα, τις εργασίες παραλαβής και/ ή παράδοσης σε μια τοποθεσία (τύποι εργασίας) και τις εργασίες παραλαβής και παράδοσης που θα πρέπει να πραγματοποιούνται εντός της ίδιας διαδρομής (αποστολές). Για τις εργασίες και τις αποστολές εμφανίζει στο χρήστη ή στο διαχειριστή του στόλου τους χρόνους εξυπηρέτησης, τη διάρκεια της υπηρεσίας, τις δεξιότητες της κάθε αποστολής και την προτεραιότητα των αποστολών. Όσον αναφορά στα οχήματα, το VROOM διαχειρίζεται πληροφορίες όπως η αρχή και το τέλος της διαδρομής ή επισημαίνεται αν το ταξίδι είναι ανοιχτού τύπου, δηλαδή ορίζεται μόνο η αρχή ή το τέλος του και οι ώρες εργασίας του οδηγού.

Το VROOM χρησιμοποιεί αλγορίθμους και τεχνικές βελτιστοποίησης για τον υπολογισμό των πιο αποδοτικών διαδρομών για έναν στόλο οχημάτων. Λαμβάνει υπόψη παράγοντες όπως η απόσταση, η χωρητικότητα του οχήματος, τα χρονικά παράθυρα και διάφοροι περιορισμοί για να καθορίσει τη βέλτιστη ακολουθία στάσεων για κάθε όχημα. Ακόμη υποστηρίζει μια σειρά από περιορισμούς και επιλογές προσαρμογής για να χειρίζεται ποικίλα σενάρια δρομολόγησης. Αυτές περιλαμβάνουν περιορισμούς χωρητικότητας οχημάτων, χρονικά

παράθυρα για παραδόσεις ή παραλαβές, συγκεκριμένες απαιτήσεις πελατών και άλλους περιορισμούς που αφορούν στο πρόβλημα VRP που επιλύεται.

Το VROOM επιτρέπει τον ορισμό διαφορετικών τύπων οχημάτων με διακριτές χωρητικότητες, ταχύτητες ή άλλα χαρακτηριστικά. Αυτό επιτρέπει τη μοντελοποίηση σεναρίων όπου τα οχήματα έχουν διαφορετικές δυνατότητες ή περιορισμούς.

Η παροχή αυτών των πληροφοριών είναι απόλυτα σημαντική για μια πλατφόρμα που διαχειρίζεται στόλους οχημάτων, καθώς δίνει λύσεις σε μικρούς υπολογιστικούς χρόνους και μπορεί να τροποποιηθεί εύκολα για να χειριστεί μεγαλύτερα προβλήματα δρομολόγησης.

## 2.6 Σύστημα Messaging Broker

Ένα από τα κύρια υποσυστήματα της πλατφόρμας CTMaaS, το οποίο παίζει καθοριστικό ρόλο στην επικοινωνία των υποσυστημάτων είναι το σύστημα Messaging Broker. Ένα σύστημα Messaging Broker, παρέχει μια πλατφόρμα για αξιόπιστη και ασύγχρονη επικοινωνία μεταξύ κατανεμημένων εφαρμογών ή υποσυστημάτων. Διευκολύνει την ανταλλαγή μηνυμάτων ενεργώντας ως ενδιάμεσος που δρομολογεί, αποθηκεύει και παραδίδει μηνύματα μεταξύ αποστολέων και παραληπτών.

Τα πιο διαδεδομένα πρότυπα ανταλλαγής μηνυμάτων είναι τα παρακάτω:

- Publish-Subscribe (Pub/ Sub): Σε αυτό το πρότυπο τα μηνύματα δημοσιεύονται σε θέματα και πολλαπλοί συνδρομητές μπορούν να λαμβάνουν αντίγραφα αυτών των μηνυμάτων. Οι συνδρομητές ενδιαφέρονται συνήθως για συγκεκριμένα θέματα και λαμβάνουν μηνύματα με βάση τις συνδρομές τους.
- Queue-based: Σε αυτό το πρότυπο τα μηνύματα αποστέλλονται σε ουρές μηνυμάτων και κάθε μήνυμα καταναλώνεται από ένα μόνο παραλήπτη. Ο παραλήπτης επιβεβαιώνει την κατανάλωση του μηνύματος και μόλις επιβεβαιωθεί το μήνυμα αφαιρείται από την ουρά.

Βασικά τμήματα ενός συστήματος messaging broker είναι η ουρά μηνυμάτων και τα topics. Η ουρά μηνυμάτων είναι ένας μηχανισμός αποθήκευσης όπου τα μηνύματα αποθηκεύονται μέχρι να καταναλωθούν από έναν παραλήπτη. Τα μηνύματα σε μια ουρά ακολουθούν το πρότυπο Queue-based και τυπικά επεξεργάζονται με τη σειρά που παραλαμβάνονται. Το topic είναι ένα λογικό κανάλι στο οποίο δημοσιεύονται μηνύματα. Οι συνδρομητές που ενδιαφέρονται για ένα συγκεκριμένο topic λαμβάνουν τα μηνύματα που δημοσιεύονται σε αυτό το θέμα. Τα μηνύματα σε topics ακολουθούν το πρότυπο Publish - Subscribe και μπορούν να καταναλωθούν από πολλούς συνδρομητές.

Ένα από τα αξιοσημείωτα χαρακτηριστικά του Messaging Broker είναι οι μηχανισμοί που διαθέτει για την εγγυημένη παράδοση του μηνύματος. Ένας Messaging Broker παρέχει επιβεβαιώσεις (Acknowledgments) για να εξασφαλίσουν αξιόπιστη παράδοση μηνυμάτων. Αφού ένας παραλήπτης επεξεργαστεί επιτυχώς ένα μήνυμα, στέλνει μια επιβεβαίωση στον Broker επιβεβαιώνοντας την επιτυχή επεξεργασία. Εάν η παράδοση ενός μηνύματος αποτύχει ο Broker συχνά παρέχει μηχανισμούς επανάληψης για την εκ νέου παράδοση του μηνύματος ή την προσπάθεια παράδοσης σε εναλλακτικούς παραλήπτες. Τα μηνύματα

μπορούν να αποθηκευτούν μόνιμα στον Messaging Broker εξασφαλίζοντας τη διαχρονικότητα ακόμη και σε περίπτωση αποτυχιών ή επανεκκίνησης του συστήματος.

Ο Messaging Broker διαθέτει δυνατότητες επέκτασης καθώς και μηχανισμούς ώστε να καλύψει τις δυναμικές ανάγκες που μπορεί να προκύψουν στο σύστημα:

- **Message Batching:** Ο Messaging Broker μπορεί να ομαδοποιεί πολλαπλά μηνύματα σε δέσμες (batches) για να βελτιώσουν την αποδοτικότητα και να μειώσουν την επιβάρυνση του δικτύου.
- **Load Balancing:** Για τη διαχείριση μεγάλου όγκου μηνυμάτων ο Messaging Broker συχνά κατανέμει το φορτίο επεξεργασίας μηνυμάτων σε πολλαπλούς κόμβους για να επιτύχει την επεκτασιμότητα και να βελτιώσει την απόδοση.
- **Horizontal Scaling:** Ο Messaging Broker μπορεί να επεκταθεί οριζόντια με την προσθήκη περισσότερων κόμβων για την αντιμετώπιση της αυξημένης κίνησης μηνυμάτων.

Όσον αναφορά στην ασφάλεια του Messaging Broker καθώς και των μηνυμάτων, ο Broker διαθέτει μηχανισμούς αυθεντικοποίησης (Authentication) και κρυπτογράφησης. Ο μηχανισμός Authentication χρησιμοποιείται για να επιβεβαιώσει την ταυτότητα των χρηστών που συνδέονται στον Broker είτε για να καταναλώσουν είτε για να δημοσιεύσουν μηνύματα. Για την προστασία της εμπιστευτικότητας και της ακεραιότητας των μηνυμάτων χρησιμοποιείται κρυπτογράφηση μηνυμάτων κατά τη μετάδοση και την αποθήκευση.

### 2.6.1 EMQX

Το υποσύστημα Messaging Broker που έχει επιλεγεί για την πλατφόρμα CTMaaS είναι το EMQX. Το EMQX είναι ένα έργο ανοιχτού κώδικα. Είναι ένας εξαιρετικά επεκτάσιμος και πλούσιος σε χαρακτηριστικά Messaging Broker MQTT που παρέχει δυνατότητες ανταλλαγής μηνυμάτων σε πραγματικό χρόνο για IoT και άλλες εφαρμογές. Υποστηρίζει το πρωτόκολλο MQTT το οποίο χρησιμοποιείται ευρέως για αποτελεσματική και αξιόπιστη επικοινωνία μεταξύ συσκευών και εφαρμογών. Το EMQX προσφέρει μια σειρά χαρακτηριστικών και δυνατοτήτων για την υποστήριξη κατανεμημένων συστημάτων ανταλλαγής μηνυμάτων μεγάλης κλίμακας. Ακολουθούν ορισμένες βασικές πτυχές του EMQX.

- **Υποστήριξη πρωτοκόλλου MQTT:** Το EMQX υποστηρίζει πλήρως το πρωτόκολλο MQTT, συμπεριλαμβανομένων των MQTT 3.1, 3.1.1 και MQTT 5.0. Παρέχει αποτελεσματική και αξιόπιστη ανταλλαγή μηνυμάτων μεταξύ publishers και subscribers.
- **Επεκτασιμότητα και υψηλές επιδόσεις:** Το EMQX έχει σχεδιαστεί για υψηλή κλιμάκωση επιτρέποντάς του να διαχειρίζεται εκατομμύρια ταυτόχρονες συνδέσεις και εκατομμύρια μηνύματα ανά δευτερόλεπτο. Υποστηρίζει επίσης οριζόντια επέκταση, επιτρέποντας την προσθήκη πολλαπλών κόμβων για την κατανομή του φορτίου μηνυμάτων.
- **Χαρακτηριστικά broker MQTT:** Το EMQX περιλαμβάνει διάφορα χαρακτηριστικά και λειτουργίες που συνήθως συνδέονται με τους broker MQTT, όπως επίπεδα ποιότητας υπηρεσίας (QoS), διατηρούμενα μηνύματα και υποστήριξη τελευταίας βούλησης και διαθήκης (LWT).

Το EMQX χρησιμοποιείται ευρέως σε εφαρμογές IoT, έξυπνων πόλεων, συνδεδεμένα οχήματα, βιομηχανικό αυτοματισμό και άλλες εφαρμογές ανταλλαγής μηνυμάτων σε πραγματικό χρόνο. Η επεκτασιμότητα, οι επιδόσεις και το εκτεταμένο σύνολο χαρακτηριστικών του το καθιστούν δημοφιλή επιλογή για τη δημιουργία αξιόπιστων και επεκτάσιμων υποδομών ανταλλαγής μηνυμάτων MQTT.

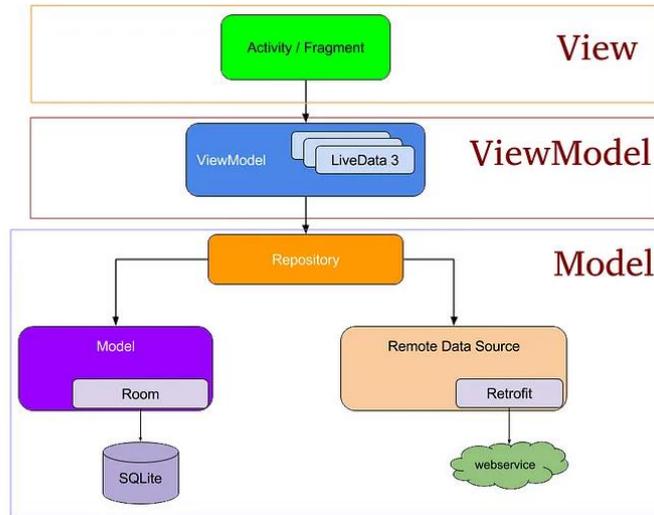
## 2.7 Mobile App

Κατά την υλοποίηση του Mobile Application χρησιμοποιήθηκε το MVVM (Model-View-ViewModel) το οποίο είναι ένα πρότυπο αρχιτεκτονικής λογισμικού που διαχωρίζει το UI (View) από τα δεδομένα (Model) και εισάγει ένα συστατικό διαμεσολαβητή που ονομάζεται ViewModel. Στόχος του είναι να παρέχει σαφή διαχωρισμό των δεδομένων από UI, να βελτιώσει τη δυνατότητα ελέγχου και να ενισχύσει τη συντηρησιμότητα της εφαρμογής. Αναλυτικότερα:

- **Model:** Αντιπροσωπεύει τα δεδομένα και την επιχειρησιακή λογική της εφαρμογής. Περιλαμβάνει τα μοντέλα των δεδομένων, λειτουργίες ανάκτησης δεδομένων μέσω δικτύου, αλληλεπιδράσεις με βάσεις δεδομένων και άλλες λειτουργίες που σχετίζονται με δεδομένα. Διαχειρίζεται τα δεδομένα της εφαρμογής και παρέχει μεθόδους για τον χειρισμό και την ανάκτηση δεδομένων.
- **View:** Αντιπροσωπεύει τα στοιχεία της διεπαφής χρήστη (UI) της εφαρμογής. Περιλαμβάνει activities, fragments και αρχεία διάταξης XML που καθορίζουν την οπτική αναπαράσταση και τη δομή του UI. Είναι υπεύθυνο για την εμφάνιση δεδομένων από το ViewModel και το χειρισμό των αλληλεπιδράσεων του χρήστη.
- **ViewModel:** Λειτουργεί ως διαμεσολαβητής μεταξύ του View και του Model. Κάνει διαθέσιμα τα δεδομένα και τις εντολές με τις οποίες το view μπορεί να συνδεθεί και να αλληλεπιδράσει. Κατέχει την κατάσταση που σχετίζεται με το UI και την επιχειρησιακή λογική που είναι απαραίτητη για την υποστήριξη του View. Περιλαμβάνει και προετοιμάζει τα δεδομένα από το Model για να παρουσιαστούν σε μορφή κατάλληλη στο View.

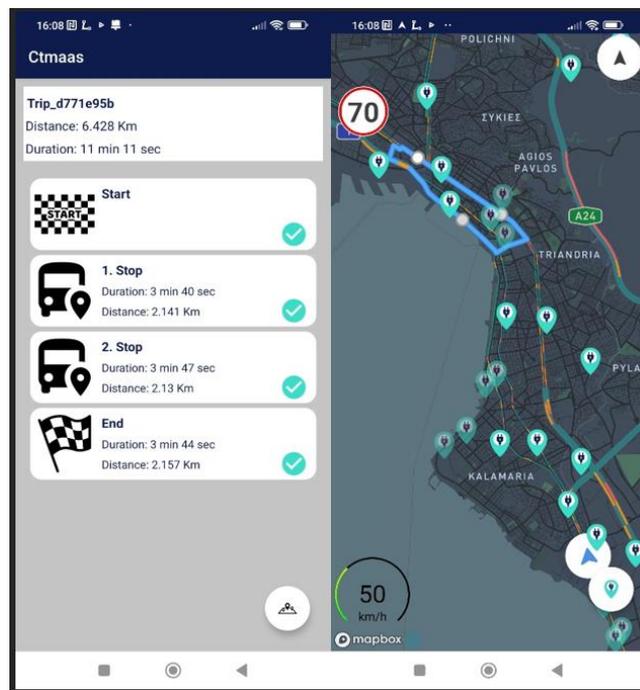
Η αλληλεπίδραση μεταξύ των στοιχείων στο πρότυπο MVVM ακολουθεί μια μονόδρομη ροή:

1. Το στοιχείο του View παρατηρεί τις αλλαγές στο ViewModel και ενημερώνει ανάλογα το UI μέσω τεχνικών data binding ή reactive programming.
2. Το View ειδοποιεί το ViewModel για τις αλληλεπιδράσεις του χρήστη, όπως τα κλικ σε κουμπιά ή οι υποβολές φόρμας.
3. Το ViewModel εκτελεί τις απαραίτητες λειτουργίες, όπως η επεξεργασία δεδομένων, η επικύρωση ή η υποβολή αιτημάτων προς το Model.
4. Το ViewModel ενημερώνει το View εκθέτοντας τα απαραίτητα δεδομένα ή ενεργοποιώντας συμβάντα που σχετίζονται με το UI.



Εικόνα 10: Διάγραμμα ροής MVVM

Κατά την υλοποίηση του MVVM για την εφαρμογή του CTMaaS χρησιμοποιήθηκαν δημοφιλή βιβλιοθήκες όπως οι Android Architecture Components (συμπεριλαμβανομένων των LiveData και ViewModel), η Data Binding Library και η RxJava για να διευκολύνουν την υλοποίηση και να απλοποιήσουν τη ροή δεδομένων μεταξύ των στοιχείων του MVVM.



Εικόνα 11: MVVM στην πλατφόρμα CTMaaS