Κοινή Διαχείριση Πόρων σε Επίπεδο Δήμων και Πόλεων με Νέες Τεχνολογίες - EfficienCity RMS



*Ψηφιακή Καινοτομία και Συστήματα Μεταφοράς Τεχνολογίας, Κωνσταντίνος Φούσκας*

*Άγγελος Φίκιας - iis23006, Αλέξανδρος Λαζαρίδης - iis23177, Βάιος Παλιούρας - iis23188, Ιωάννης Τσιρκινίδης - iis23172*

[**Περίληψη 2**](#_9n5emwpfr7ac)

[**Δομή ομάδας και ρόλοι 3**](#_jrk6kbh8tsns)

[**Το πρόβλημα και η προτεινόμενη λύση 3**](#_c6veg8lt5d4h)

[**Ορισμός χρηστών στόχου 4**](#_9ogd4v2dveb5)

[Χαρτογράφηση ενδιαφερομένων και των στόχων τους 4](#_dddxjp87nxfp)

[Συλλογή απαιτήσεων 5](#_hncsn1l1p873)

[Use Case Diagram - Visual Paradigm 7](#_aa961dgywemx)

[Καταγραφή περιορισμών 7](#_2tw6qgwon1gj)

[Οικονομική Βιωσιμότητα 8](#_11t07u6potwo)

[**Ανάλυση Υφιστάμενης Κατάστασης & Διαδικασιών 9**](#_91wo2mphjtm)

[Καταγραφή τρέχουσας ροής - BPMN 9](#_p12d9hco402w)

[Root Cause Analysis 10](#_2cazn7jbpx7)

[SWOT Analysis 11](#_rvv3cf9dw2o1)

[**Σχεδίαση Στόχου (“to‑be”) & Μετρικές Επιτυχίας 11**](#_wjqnudu7hp34)

[Ορισμός στόχων SMART 13](#_xhhde54ryksh)

[Μετρικές - KPIs 14](#_pb9junhfgpg)

[Κριτήρια Αποδοχής 15](#_tzndjvqp0d0m)

[**Αρχιτεκτονική & Τεχνικές Προδιαγραφές 15**](#_3oawp8zg08np)

[Components Εφαρμογής 15](#_m08mrevcirly)

[Frontend Επίπεδο 15](#_89s5izy1iz9j)

[Backend Υπηρεσίες 16](#_o1gkts1ceh99)

[Pipeline Δεδομένων και Υποδομή Βάσεων 16](#_a2cwp6r1mgk4)

[Επίπεδο Ανάλυσης - Big Data και AI 17](#_lyyuzq5rc0dg)

[Blockchain - Επίπεδο Διαφάνειας Συναλλαγών 18](#_ez5uzoh13lx2)

[Υποδομή Cloud 18](#_4ovgy8ajb3mr)

[Τεχνολογική Αρχιτεκτονική 20](#_oagtll4v1lg)

[Πίνακας Προδιαγραφών 20](#_smmgyeqq6sst)

[Data Schema 21](#_nfigipx4hdwx)

[PostgreSQL Schema 21](#_cnwszaq5tvpv)

[ClickHouse Schema 21](#_sy8598ubaww0)

[Kafka Topics 22](#_ur5fnraug6x4)

[Οντότητες Blockchain 22](#_ecfysstmi33x)

[**UX Σχεδίαση & Mockups 22**](#_zgj8pt36l63o)

[User Flow Diagrams + 22](#_qa9panjz5o56)

[Mockup Screens + 25](#_p2cai7nqo9ur)

[**Έλεγχοι 30**](#_oj4ducolohdn)

[Κατηγορίες Δοκιμών 30](#_utxa9cs2ytcg)

[Πίνακας Ιχνηλασιμότητας 31](#_ck64lq8rktri)

[Δοκιμές & Αποδοχή (User Acceptance Testing - UAT) 32](#_nlbc9bms5ih0)

[Σχεδιασμός Πιλοτικού 33](#_e3ijlmtb0raf)

[**Ενδεικτική Υλοποίηση - Proof of Concept 33**](#_s23j5v4kry6t)

[**Συμπεράσματα 35**](#_q34ger69u52w)

# Περίληψη

Η εργασία εξετάζει τη διαχείριση πόρων μεταξύ δήμων και προτείνει ένα σύστημα που ενισχύει τη συνεργασία, τη διαφάνεια και την αποδοτικότητα, το EfficienCity RMS (Resource Management System). Η μεθοδολογική προσέγγιση ξεκινά με την ανάλυση αναγκών και χρηστών και συνεχίζει με τη χαρτογράφηση της υπάρχουσας κατάστασης. Στη συνέχεια καθορίζονται οι στόχοι και οι δείκτες επιτυχίας για το μέλλον. Ακολουθεί ο σχεδιασμός της τεχνικής υποδομής με σύγχρονες τεχνολογίες και η δημιουργία διεπαφών χρήστη και mockups. Το επόμενο στάδιο περιλαμβάνει δοκιμές, πιλοτική εφαρμογή και διαχείριση κινδύνων. Τέλος, πραγματοποιείται η υλοποίηση του συστήματος, εξασφαλίζοντας λειτουργικότητα, ασφάλεια και βιωσιμότητα μέσω ενός mvp.

# Δομή ομάδας και ρόλοι

*Η ομάδα υιοθέτησε την Agile μεθοδολογία, αξιοποιώντας το Jira για την οργάνωση των εργασιών και το Notion, την παρακολούθηση της προόδου και τη συνεχή αναθεώρηση των στόχων βάσει feedback και συναντήσεων.*

*Μέλη:*

* *Άγγελος Φίκιας - iis23006 [Software Architect]*
* *Αλέξανδρος Λαζαρίδης - iis23177 [Design Lead]*
* *Βάιος Παλιούρας - iis23188 [Project Manager]*
* *Ιωάννης Τσιρκινίδης - iis23172 [Tech Research Lead]*

# Το πρόβλημα και η προτεινόμενη λύση

Στη σημερινή πραγματικότητα, οι δήμοι λειτουργούν σε μεγάλο βαθμό αυτόνομα, χωρίς κοινό σύστημα συνεργασίας και συντονισμού. Η έλλειψη διασύνδεσης οδηγεί σε χαμηλή αποδοτικότητα, καθώς σημαντικοί δημοτικοί πόροι, όπως οχήματα, μηχανήματα, εργαλεία και IT assets (εκτυπωτές, servers, Η/Υ), παραμένουν συχνά αδρανείς. Αυτό δημιουργεί περιττές δαπάνες, μειώνει την ευελιξία των υπηρεσιών και δυσχεραίνει την ικανότητα των δήμων να ανταποκριθούν αποτελεσματικά στις ανάγκες των πολιτών.

Για την αντιμετώπιση αυτών των προβλημάτων προτείνεται η ανάπτυξη ενός Ενοποιημένου Συστήματος Διαχείρισης Δημοτικών Πόρων (Resource Management System - RMS). Το RMS θα λειτουργεί ως κεντρική ψηφιακή πλατφόρμα, παρέχοντας πρόσβαση σε μια κοινή βάση δεδομένων με πλήρη και σε πραγματικό χρόνο καταγραφή των διαθέσιμων πόρων κάθε δήμου. Μέσω του συστήματος θα είναι δυνατή η αίτηση, διάθεση και βέλτιστη αξιοποίηση εξοπλισμού, υποστηριζόμενη από προτάσεις και αναλύσεις της πλατφόρμας. Παράλληλα, θα παράγονται και θα εξάγονται στατιστικές αναφορές, ενώ οι πολίτες θα έχουν δυνατότητα πλήρους διαφάνειας με δικαιώματα ανάγνωσης στη βάση και στο blockchain συναλλαγών.

Για παράδειγμα, αν ο Δήμος Θεσσαλονίκης διαθέτει δέκα απορριμματοφόρα και κάποια δεν χρησιμοποιούνται πλήρως, ενώ ο Δήμος Κοζάνης έχει αυξημένες ανάγκες αποκομιδής, το RMS θα προτείνει τη μεταφορά ή διάθεση οχημάτων μεταξύ των δήμων. Οι δήμοι αποφασίζουν για την ολοκλήρωση της συναλλαγής, και το σύστημα καταγράφει κάθε κίνηση, ενημερώνοντας τη βάση δεδομένων και το blockchain, εξασφαλίζοντας διαφάνεια, αποτελεσματική κατανομή πόρων, μείωση κόστους και ενίσχυση της συνεργασίας.

Με αυτόν τον τρόπο, το RMS συμβάλλει στη βιώσιμη ανάπτυξη, στην αύξηση της αποδοτικότητας και στη βελτίωση της ποιότητας των υπηρεσιών προς τους πολίτες.

# Ορισμός χρηστών στόχου

* Δημοτικές Αρχές
  + Πρόσβαση στην κοινή βάση δεδομένων πόρων (οχήματα, μηχανήματα, εξοπλισμός) και στο blockchain συναλλαγών.
  + Υποβολή και διαχείριση των δικών τους πόρων.
  + Υποβολή αιτήσεων για παροχή πόρων από άλλους δήμους.
  + Διάθεση αδρανών πόρων σε δήμους με ανάγκη.
  + Προβολή και εξαγωγή στατιστικών αναφορών.
* Πολίτες
  + Πρόσβαση στη δημόσια εκδοχή της βάσης δεδομένων πόρων και του blockchain.
  + Προβολή και εξαγωγή στατιστικών αναφορών.
* Διαχειριστές Συστήματος
  + Διαχείριση δικαιωμάτων πρόσβασης χρηστών (δήμοι, πολίτες).
  + Παρακολούθηση και διασφάλιση σωστής λειτουργίας και διαθεσιμότητας του συστήματος.
  + Δημιουργία και επαναφορά αντιγράφων ασφαλείας σε περίπτωση βλάβης.

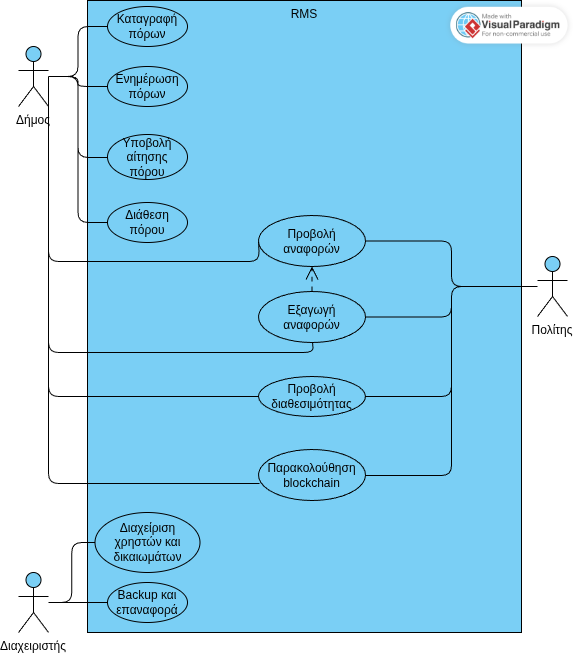
## Χαρτογράφηση ενδιαφερομένων και των στόχων τους

| **Stakeholder** | **Ρόλος** | **Στόχος / Όφελος** | **Παράδειγμα Εφαρμογής** |
| --- | --- | --- | --- |
| **Δημοτικές Αρχές** | Διαχείριση βάσης πόρων, υποβολή αιτήσεων, διάθεση εξοπλισμού | Βελτιστοποίηση χρήσης πόρων, μείωση κόστους, ενίσχυση συνεργασίας μεταξύ δήμων | Ο Δήμος Κοζάνης αιτείται απορριμματοφόρο από τη Θεσσαλονίκη. Ο Δήμος Λάρισας διαθέτει μηχάνημα έργου στον Δήμο Τρικάλων. |
| **Πολίτες** | Πρόσβαση στη δημόσια βάση πόρων για ενημέρωση | Διαφάνεια και εμπιστοσύνη στη διαχείριση των δημοτικών πόρων | Οι πολίτες ενημερώνονται για τον αριθμό και τη χρήση οχημάτων καθαριότητας του δήμου τους. |
| **Διαχειριστές Συστήματος** | Διαχείριση δικαιωμάτων, ασφάλειας και υποστήριξης συστήματος | Διασφάλιση ασφάλειας και σωστής λειτουργίας του συστήματος | Αναθέτουν δικαιώματα χρήσης, διασφαλίζουν συνεχιζόμενη λειτουργία και διαχειρίζονται αντίγραφα ασφαλείας και βλάβες. |

## Συλλογή απαιτήσεων

| **Requirement** | **Κατηγορία** | **MoSCoW** | **Ranking** | **Περιγραφή** |
| --- | --- | --- | --- | --- |
| Καταγραφή και ενημέρωση διαθέσιμων πόρων (οχήματα, εξοπλισμός, μηχανήματα) | Λειτουργική | Must | 1 | Οι δημοτικές αρχές να καταγράφουν και να ενημερώνουν τους πόρους τους. |
| Υποβολή αιτήσεων παροχής πόρων προς άλλους δήμους | Λειτουργική | Must | 2 | Οι δήμοι να μπορούν να ζητούν πόρους από άλλους δήμους. |
| Διάθεση πόρων σε άλλους δήμους | Λειτουργική | Must | 3 | Οι δήμοι να μπορούν να διαθέτουν πόρους σε άλλους δήμους. |
| Ορατότητα στη διαθεσιμότητα πόρων όλων των δήμων και blockchain (ΒΔ) | Λειτουργική | Must | 4 | Το σύστημα να εμφανίζει σε πραγματικό χρόνο τη διαθεσιμότητα πόρων. |
| Πρόσβαση πολιτών στη δημόσια βάση και blockchain για ενημέρωση (ΒΔ) | Λειτουργική | Should | 5 | Οι πολίτες να μπορούν να ενημερώνονται για τη χρήση πόρων. |
| Υποστήριξη δικαιωμάτων πρόσβασης ανά τύπο χρήστη | Λειτουργική | Must | 6 | Διαφορετικά δικαιώματα για δήμο, πολίτη και διαχειριστή. |
| Δημιουργία αντιγράφων ασφαλείας | Λειτουργική | Must | 7 | Δημιουργία αντιγράφων με σκοπό τη διαθεσιμότητα και ακεραιότητα των δεδομένων. |
| Παραγωγή στατιστικών αναφορών | Λειτουργική | Should | 7 | Το σύστημα να δημιουργεί αναφορές χρήσης πόρων. |
| Εξαγωγή αναφορών | Λειτουργική | Could | 8 | Οι αναφορές να μπορούν να εξαχθούν για περαιτέρω ανάλυση. |
| Φιλικό γραφικό περιβάλλον (UI) | Μη Λειτουργική | Should | 1 | Εύκολη πλοήγηση και αναζήτηση πόρων. |
| Ασφάλεια και προστασία δεδομένων | Μη Λειτουργική | Must | 2 | Role-based access, encryption, διασφάλιση εμπιστευτικότητας. |
| Επεκτασιμότητα | Μη Λειτουργική | Must | 3 | Προσθήκη νέων δήμων και δεδομένων χωρίς τεχνικούς περιορισμούς. |
| Υψηλή διαθεσιμότητα και αξιοπιστία | Μη Λειτουργική | Must | 4 | Uptime ≥ 99%. |
| Απόκριση λειτουργιών | Μη Λειτουργική | Should | 5 | Βασικές ενέργειες σε ≤ 3 δευτερόλεπτα. |
| Δημιουργία αντιγράφων ασφαλείας (backup & recovery) | Μη Λειτουργική | Must | 6 | Επαναφορά δεδομένων σε περίπτωση βλάβης. |

## Use Case Diagram - Visual Paradigm



## Καταγραφή περιορισμών

| **Κατηγορία** | **Περιορισμός** | **Σημειώσεις / Σχόλια** |
| --- | --- | --- |
| Κανονιστικοί / Νομικοί | Συμμόρφωση με GDPR | Τα προσωπικά δεδομένα πολιτών και υπαλλήλων πρέπει να είναι ανώνυμα και ασφαλή.NA |
| Κανονιστικοί / Νομικοί | Αποθήκευση δεδομένων εντός ΕΕ | Απαιτείται συμβατότητα με ευρωπαϊκές πολιτικές ασφάλειας πληροφοριών και χρήση cloud servers εντός ΕΕ. |
| Κανονιστικοί / Νομικοί | Νομική επαλήθευση συναλλαγών | Οι συναλλαγές μεταξύ δήμων πρέπει να καταγράφονται στο blockchain και να είναι ελέγξιμες από κρατικούς φορείς. |
| Λειτουργικοί | Ενημέρωση βάσης σε προκαθορισμένα διαστήματα | Διασφαλίζει συνέπεια και αποφυγή διπλών εγγραφών. |
| Λειτουργικοί | Δικαιώματα πρόσβασης | Οι πολίτες έχουν δικαίωμα ανάγνωσης, οι δήμοι δικαίωμα εισαγωγής/τροποποίησης, περιορισμός για λόγους ασφάλειας και διαφάνειας. |
| Τεχνικοί | Λειτουργία σε web περιβάλλον | Ευκολία πρόσβασης χωρίς εγκατάσταση λογισμικού. |
| Τεχνικοί | Υποστήριξη ταυτόχρονων συνδέσεων | Απαιτείται κλιμάκωση (scalability) και διαχείριση concurrency για πολλούς δήμους ταυτόχρονα. |
| Τεχνικοί | Ασφάλεια επικοινωνίας | Υποχρεωτική χρήση HTTPS, SSL/TLS και κρυπτογράφηση για ασφαλή μεταφορά δεδομένων. |

## Οικονομική Βιωσιμότητα

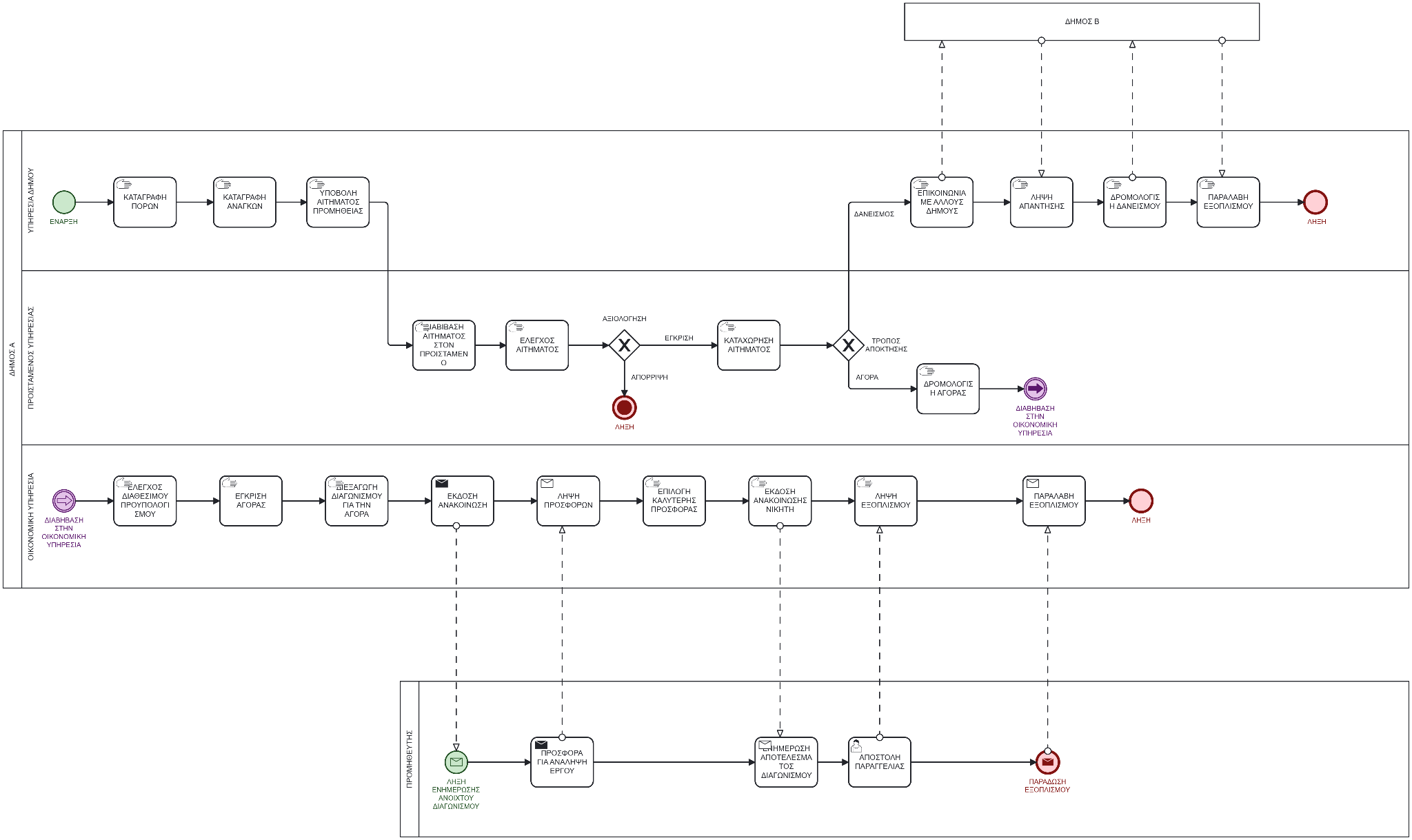
* Μείωση κόστους: Αξιοποίηση υπαρχόντων πόρων μειώνει την ανάγκη για νέες αγορές και συντήρηση αδρανών εξοπλισμών.
* Βελτιστοποίηση προϋπολογισμού: Κεντρική διαχείριση αιτήσεων και διάθεσης πόρων περιορίζει καθυστερήσεις και υπερβάσεις κόστους.
* Διαφάνεια και έλεγχος: Αναφορές κόστους και διαφάνεια συναλλαγών περιορίζουν σπατάλες και πιθανότητες καταχρήσεων.
* Στρατηγικές αποφάσεις: KPI και predictive analytics υποστηρίζουν τεκμηριωμένη κατανομή πόρων.
* ROI άμεσο και μετρήσιμο: Κάθε ευρώ που επενδύεται μετατρέπεται σε απτά οικονομικά και λειτουργικά οφέλη για τον δήμο.
* Αξιοπιστία και βιωσιμότητα: Η πλατφόρμα δημιουργεί ένα μακροπρόθεσμα αποδοτικό εργαλείο διαχείρισης δημοτικών πόρων.

# Ανάλυση Υφιστάμενης Κατάστασης & Διαδικασιών

## Καταγραφή τρέχουσας ροής - BPMN

Η απόκτηση πόρων από τους δήμους γίνεται με τον εξής τρόπο. Αρχικά ένας υπάλληλος της υπηρεσίας του δήμου καταγράφει τους πόρους που έχει στην κατοχή του ο δήμος αλλά και αυτούς που χρειάζεται. Κάνει την σχετική αίτηση και την καταθέτει στον προϊστάμενο για να την ελέγξει. Στην περίπτωση που η αίτηση εγκριθεί, το αίτημα διαβιβάζεται στην οικονομική υπηρεσία του δήμου όπου αποφασίζεται με βάση τον προϋπολογισμό αν θα προχωρήσουν σε αγορά η δανεισμό των πόρων από άλλους δήμους. Στην περίπτωση αγοράς διεξάγεται διαγωνισμός για την εύρεση της χαμηλότερης προσφοράς, ενώ στην περίπτωση δανεισμού, ο υπεύθυνος υπάλληλος επικοινωνεί με δήμους και αναζητά τα αγαθά προς δανεισμό.

Με βάση τη μοντελοποίηση της υφιστάμενης κατάστασης, όπως προέκυψε από εκτενείς συζητήσεις με τους υπαλλήλους του Δήμου, διαπιστώθηκε ότι οι τρέχουσες διαδικασίες είναι πρόχειρα δομημένες και στηρίζονται σε ξεπερασμένες πρακτικές. Οι πόροι καταγράφονται σε ασυντόνιστα φύλλα Excel, τα οποία σπάνια ενημερώνονται. Επιπλέον, στη διαδικασία δανεισμού οι διαπραγματεύσεις και οι συμφωνίες πραγματοποιούνται προφορικά, χωρίς επίσημη τεκμηρίωση. Ως συνέπεια, απουσιάζουν η ιχνηλασιμότητα και η διαφάνεια, γεγονός που αυξάνει τον κίνδυνο λαθών και καθιστά δύσκολο τον έλεγχο των ενεργειών.



## Root Cause Analysis

| **Αιτία** | **Αποτέλεσμα** |
| --- | --- |
| Έλλειψη κοινής βάσης δεδομένων πόρων | Κάθε δήμος διαχειρίζεται τα στοιχεία ξεχωριστά, χωρίς συγκεντρωμένη ορατότητα. |
| Χειροκίνητες διαδικασίες αγοράς | Οι διαδικασίες καταγραφής, έγκρισης και αγοράς γίνονται χειροκίνητα, προκαλώντας καθυστερήσεις και αυξημένα έξοδα. |
| Περιορισμένη διαφάνεια κόστους και χρήσης πόρων | Δεν υπάρχει κοινή παρακολούθηση ή αναφορά κόστους, δυσχεραίνοντας τον έλεγχο προϋπολογισμού. |
| Κατανεμημένα δεδομένα σε πολλαπλά συστήματα | Απαιτείται επαναληπτική καταγραφή και συντονισμός, αυξάνοντας χρόνο και κόστος εργασίας. |

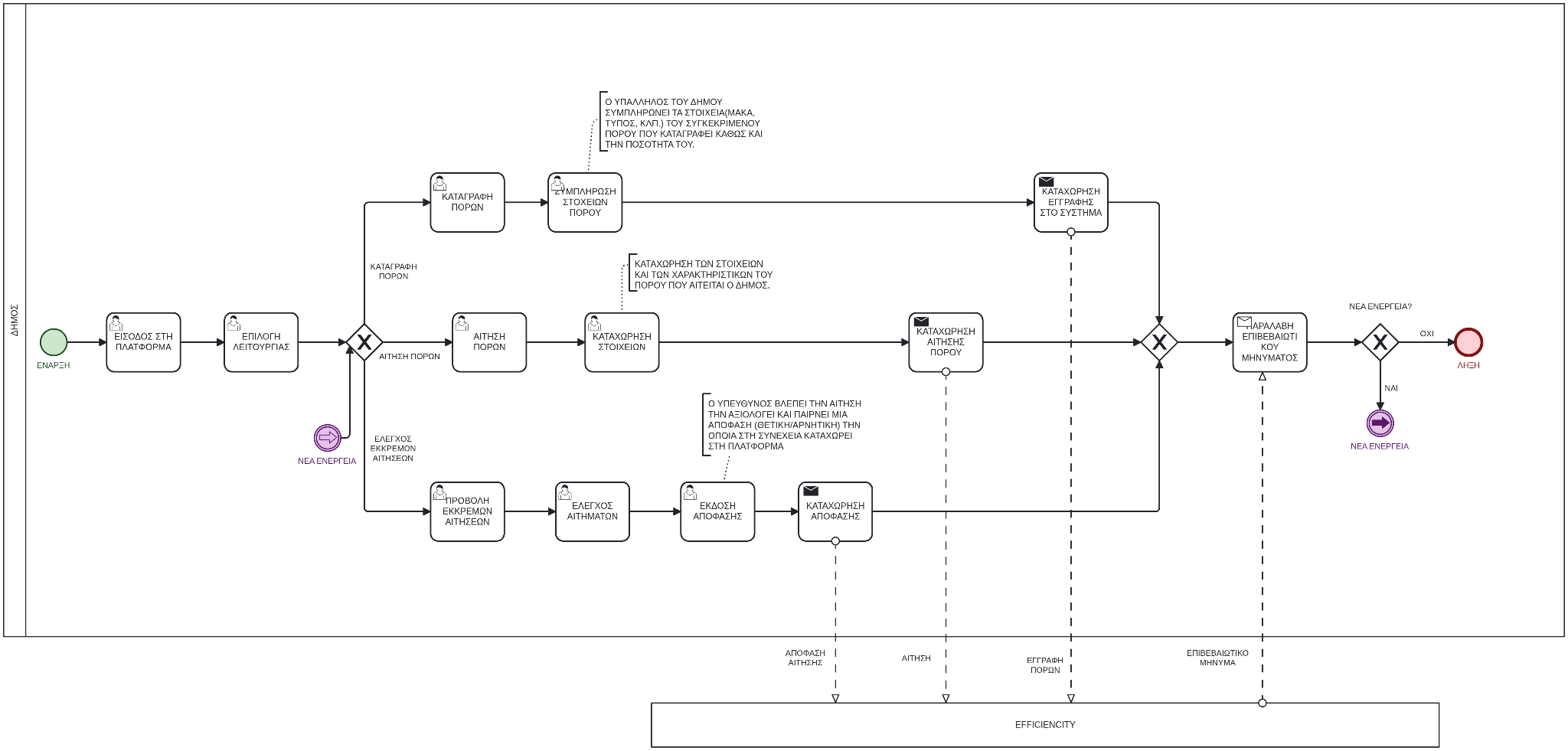
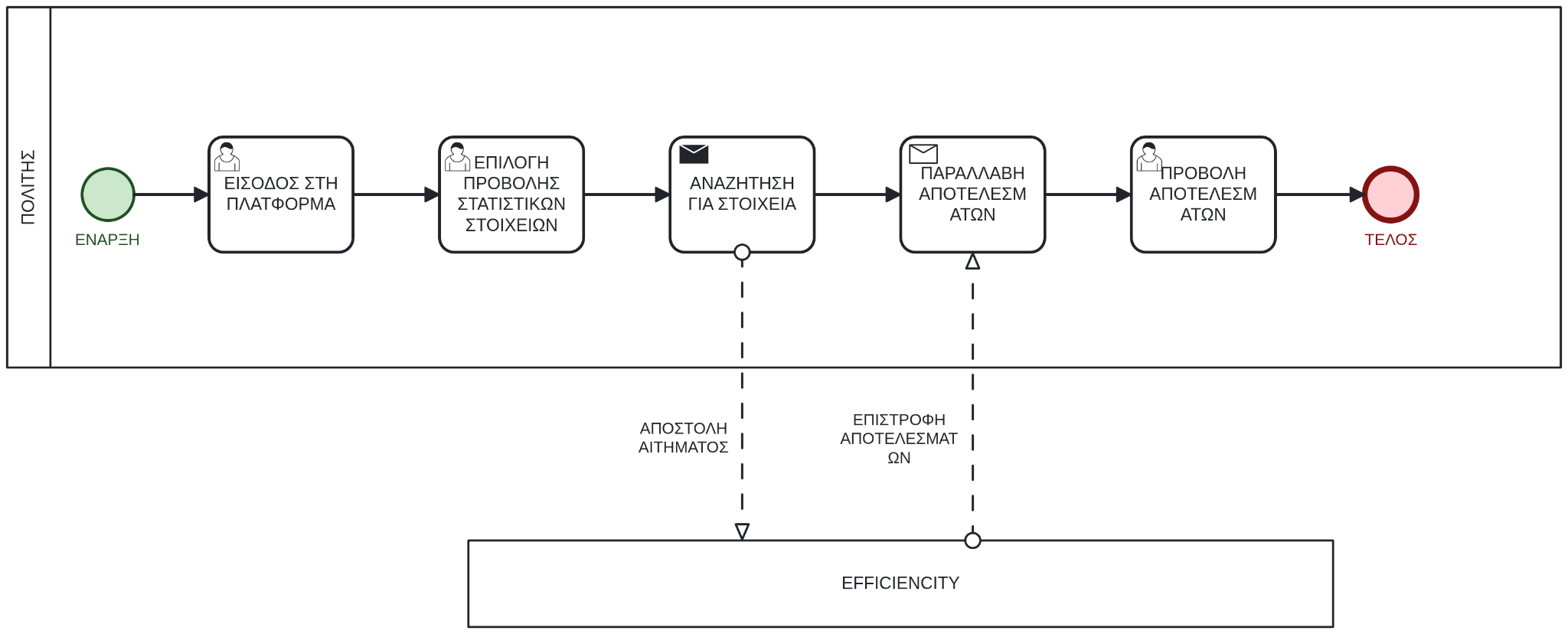
## SWOT Analysis

| **Κατηγορία** | **Στοιχεία** |
| --- | --- |
| **Strengths (Δυνατά Σημεία)** | - Κάθε δήμος γνωρίζει τις ανάγκες και τους διαθέσιμους πόρους του.  - Υπάρχουν ήδη υποδομές και προσωπικό για αγορές και καταγραφή πόρων. |
| **Weaknesses (Αδυναμίες)** | - Απουσία κεντρικής βάσης δεδομένων.  - Έλλειψη ακεραιότητας δεδομένων.  - Χειροκίνητες διαδικασίες.  - Αδυναμία τεκμηρίωσης και ιχνηλασιμότητας ενεργειών. |
| **Opportunities (Ευκαιρίες)** | - Δημιουργία ενιαίου συστήματος διαχείρισης πόρων (RMS).  - Μείωση κόστους μέσω συνεργασίας.  - Ενίσχυση διαφάνειας και εμπιστοσύνης των πολιτών προς τους δήμους. |
| **Threats (Απειλές)** | - Αντίσταση στην αλλαγή από δήμους.  - Κόστος υλοποίησης και συντήρησης νέου συστήματος.  - Τεχνικές προκλήσεις. |

# Σχεδίαση Στόχου (“to‑be”) & Μετρικές Επιτυχίας

Σκοπός της προτεινόμενης λύσης είναι ο σχεδιασμός και η υλοποίηση μιας ολοκληρωμένης ψηφιακής πλατφόρμας που θα ενισχύσει σημαντικά την κοινή διαχείριση πόρων μεταξύ των Δήμων. Η πλατφόρμα στοχεύει στον εκσυγχρονισμό και την απλοποίηση των σχετικών διαδικασιών, επιτρέποντας τη μετάβαση από αποσπασματικές, χειροκίνητες και μη τεκμηριωμένες πρακτικές σε μια πλήρως ψηφιακή, οργανωμένη και διαφανή λειτουργία.

Μέσω της πλατφόρμας, οι υπάλληλοι θα καταγράφουν τα αποθέματα του Δήμου σε ένα ενιαίο ψηφιακό περιβάλλον. Όταν προκύπτουν ελλείψεις, θα έχουν τη δυνατότητα να δημιουργούν ηλεκτρονικά αιτήματα πόρων, τα οποία το σύστημα θα επεξεργάζεται αυτόματα για να εντοπίζει πιθανές διαθέσιμες λύσεις. Παράλληλα, θα παρέχεται στους χρήστες άμεση πρόσβαση σε πληροφορίες αποθεμάτων άλλων Δήμων, επιτρέποντας την ταχεία υποβολή χειροκίνητων αιτημάτων δανεισμού όπου απαιτείται.



1. BPMN Πολιτη
2. BPMN Δημου +

Η λειτουργία της πλατφόρμας θα βασίζεται σε δύο συμπληρωματικούς τεχνολογικούς άξονες. Η προτεινόμενη μελλοντική κατάσταση προσφέρει συνολικά:

* *απλοποιημένες και ταχύτερες διαδικασίες,*
* *ενοποιημένη και ενημερωμένη καταγραφή πόρων,*
* *διαφάνεια στις συναλλαγές,*
* *ακριβή ιχνηλασιμότητα όλων των ενεργειών,*
* *άμεση πρόσβαση σε πόρους άλλων Δήμων*
* *μειωμένο κόστος μέσω αξιοποίησης διαθέσιμων πόρων πριν από νέες αγορές.*

## Ορισμός στόχων SMART

* Ενιαία βάση δεδομένων πόρων
  + Κεντρική καταγραφή οχημάτων, μηχανημάτων και εξοπλισμού για όλους τους δήμους
  + Οφέλη: Συγκεντρωμένη ορατότητα, βελτιωμένη συνεργασία, μείωση αχρησιμοποίητων πόρων
* Διαχείριση αιτήσεων διάθεσης/απόκτησης πόρων
  + Οι δήμοι βλέπουν διαθέσιμους πόρους και υποβάλλουν αιτήματα
  + Οφέλη: Μείωση καθυστερήσεων, αύξηση αποδοτικότητας, καλύτερη κατανομή πόρων
* Καταγραφή και παρακολούθηση χρήσης πόρων (Traceability)
  + Ιχνηλασιμότητα από αίτηση έως χρήση και επιστροφή πόρων
  + Οφέλη: Αποφυγή διπλών εγγραφών, πλήρης τεκμηρίωση, καλύτερη διαχείριση πόρων
* Διαφάνεια προϋπολογισμού
  + Προβολή διαθέσιμου προϋπολογισμού ανά δήμο
  + Οφέλη: Στρατηγικός προγραμματισμός, έλεγχος κόστους, μειωμένα έξοδα
* Φιλικό user interface (UI)
  + Εύχρηστο UI για δημοτικές αρχές & πολίτες (ανάγνωση μόνο)
  + Οφέλη: Αύξηση αποδοχής, συμμετοχή χρηστών, εύκολη πλοήγηση
* Αναφορές & Στατιστικά
  + Δημιουργία αναφορών για αξιολόγηση αποδοτικότητας
  + Οφέλη: Στρατηγικός σχεδιασμός, λήψη τεκμηριωμένων αποφάσεων
* Δικαιώματα πρόσβασης & ασφάλεια
  + Role-based access για δήμους, πολίτες, διαχειριστές
  + Οφέλη: Προστασία δεδομένων, εμπιστοσύνη χρηστών, ασφαλής διαχείριση
* Διαθεσιμότητα και αξιοπιστία συστήματος
  + Uptime ≥ 99%, web-based πρόσβαση
  + Οφέλη: Σταθερή λειτουργία 24/7, πρόσβαση σε πραγματικό χρόνο
* Διαφάνεια & Blockchain
  + Καταγραφή συναλλαγών μεταξύ δήμων στο blockchain
  + Οφέλη: Νομική επαλήθευση, διαφάνεια, έλεγχος από φορείς
* Κλιμάκωση / Scalability
  + Προσθήκη νέων δήμων και πόρων χωρίς περιορισμούς
  + Οφέλη: Υποστήριξη μελλοντικής ανάπτυξης, προσαρμογή σε αύξηση χρήσης
* Backup & Recovery
  + Δημιουργία αντιγράφων ασφαλείας & δυνατότητα επαναφοράς
  + Οφέλη: Ασφάλεια δεδομένων, επιχειρησιακή συνέχεια, μείωση κινδύνου απώλειας δεδομένων

## Μετρικές - KPIs

| **Μετρική** | **Τι μετράει** | **Στόχος / Τιμή Αναφοράς** |
| --- | --- | --- |
| Χρόνος ολοκλήρωσης διάθεσης πόρου | Από την υποβολή αίτησης μέχρι τη διαθεσιμότητα του πόρου | < 3 ημέρες |
| Αξιοποίηση πόρων | Ποσοστό χρόνου που οι πόροι είναι σε χρήση | ≥ 90% |
| Μείωση αδράνειας πόρων | Ποσοστό χρόνου που οι πόροι είναι αδρανείς | < 10% |
| Χρόνος κράτησης → διάθεσης | Μέσος χρόνος από την καταχώρηση ανάγκης μέχρι τη χρήση του πόρου | < 2 ημέρες |
| Διαθεσιμότητα πόρων | Ποσοστό των πόρων που είναι διαθέσιμοι ανά πάσα στιγμή | ≥ 95% |
| Διαφάνεια συναλλαγών | Ποσοστό συναλλαγών που καταγράφονται πλήρως στο σύστημα | 100% |
| Ικανοποίηση χρηστών | Ερωτηματολόγιο προς δημοτικές αρχές και πολίτες για ευκολία χρήσης | Βαθμός ≥ 4/5 |

## Κριτήρια Αποδοχής

* Προσβασιμότητα: Το σύστημα πρέπει να είναι λειτουργικό για όλους τους χρήστες, σε desktop και mobile, με σωστή αντίθεση και υποστήριξη πληκτρολογίου. Άρα να είναι web application.
* Χρηστικότητα: Οι βασικές ενέργειες (π.χ. αίτηση πόρου, διάθεση πόρου) να ολοκληρώνονται σε ≤ 4 βήματα. Οι χρήστες να λαμβάνουν σαφές feedback για κάθε ενέργεια.
* Απόδοση: Latency <1 δευτερόλεπτο για real-time updates, uptime >99.9%.
* Ασφάλεια: Κρυπτογράφηση δεδομένων, role-based access control, audit logs, GDPR compliant.
* Βιωσιμότητα / Ethics: Ελάχιστη κατανάλωση ενέργειας και συμμόρφωση με πράσινες πολιτικές.

# Αρχιτεκτονική & Τεχνικές Προδιαγραφές

Η αρχιτεκτονική του EfficienCity RMS ακολουθεί την προσέγγιση microservices με event-driven communication, εξασφαλίζοντας κλιμακωσιμότητα, ανθεκτικότητα και ευελιξία στην ανάπτυξη.

1. Microservices Architecture: Κάθε business capability υλοποιείται ως ανεξάρτητη υπηρεσία με δικό της database schema (Database per Service pattern).
2. Event-Driven Architecture: Τα microservices επικοινωνούν ασύγχρονα μέσω events, μειώνοντας το coupling και αυξάνοντας την ανθεκτικότητα.

## Components Εφαρμογής

### Frontend Επίπεδο

* Dashboard Δήμων:
  + Διαχείριση πόρων
  + Διαχείριση αιτήσεων
  + Real-time notifications
  + Analytics & KPIs
  + Export reports: Εξαγωγή αναφορών σε PDF και Excel.
* Dashboard Πολιτών:
  + Public view
  + Transparency portal
  + Στατιστικά
* Admin Console:
  + User management
  + Role-based access control (RBAC)
  + System monitoring
  + Audit logs
  + Configuration
* Τεχνολογία: **Angular**
* Accessibility & Mobile-First:
  + WCAG 2.1 AA συμμόρφωση: Σωστή αντίθεση χρωμάτων, ARIA labels, keyboard navigation.
  + Mobile-first design: Σχεδιασμένο πρώτα για κινητά, responsive για tablet και desktop.

### Backend Υπηρεσίες

* Microservices Architecture: Κάθε υπηρεσία είναι ανεξάρτητη, με δικό της database schema και lifecycle.
  + Resource Service
  + Request Service
  + User Service
  + Analytics Service
  + Notification Service
  + Blockchain Service
* Τεχνολογία: **Spring Boot**

### Pipeline Δεδομένων και Υποδομή Βάσεων

* RDBMS - **PostgreSQL**
* Data Warehouse - **ClickHouse**
* Message Queue - **Kafka**
* Data processing - **Apache Spark**
* Caching **- Redis**
* Backup & Recovery:
  + Καθημερινά full backups: Αυτόματα backups κάθε νύχτα (3:00 AM).
  + Incremental backups: Continuous archiving για minimal data loss.
  + Retention policy: 30 ημέρες για full backups, 90 ημέρες για transaction logs.
  + Recovery time objective (RTO): < 2 ώρες.
  + Recovery point objective (RPO): < 15 λεπτά (μέσω WAL).
  + Disaster recovery: Geo-redundant backups σε secondary Azure region.

### Επίπεδο Ανάλυσης - Big Data και AI

* Descriptive Analytics:
  + KPI Monitoring: Real-time υπολογισμός και visualization KPIs.
  + Αξιοποίηση πόρων: Ποσοστό χρόνου που οι πόροι είναι σε χρήση.
  + Διαθεσιμότητα: Ποσοστό πόρων διαθέσιμων για χρήση.
  + Χρόνος ολοκλήρωσης αιτήσεων: Από submission έως completion.
  + Μείωση αδράνειας: Tracking πόρων που δεν χρησιμοποιούνται.
* Reports & Dashboards: Intuitive visualizations για γρήγορη κατανόηση.
  + Time-series trends για historical patterns.
  + Geographical heatmaps για resource distribution.
  + Comparative analysis μεταξύ δήμων.
* Predictive Analytics:
  + Demand Forecasting: AI models για πρόβλεψη μελλοντικών αναγκών.
  + Features: εποχικότητα, ιστορική χρήση, population, upcoming events.
  + Optimal matching δήμων με ανάγκες και διαθέσιμους πόρους.
  + Transportation cost minimization.
  + Fraud detection σε συναλλαγές.
  + Usage pattern anomalies για preventive maintenance.
  + Automated Insights:
    - AI-generated recommendations στο dashboard.
    - Natural language insights: "Ο Δήμος X έχει 3 απορριμματοφόρα αδρανή 10+ ημέρες. Προτείνεται διάθεση σε Δήμο Y που έχει αυξημένη ζήτηση."
    - Proactive alerts.
* Τεχνολογία: **Python**
  + Libraries: Pandas για data manipulation, **Scikit-learn** για ML models, **TensorFlow/PyTorch** για deep learning (αν χρειαστεί).
  + Visualization: **Matplotlib, Seaborn** για exploratory analysis.
  + Deployment: ML models served μέσω REST API ή batch predictions.
  + MLOps: Model versioning, A/B testing, performance monitoring.

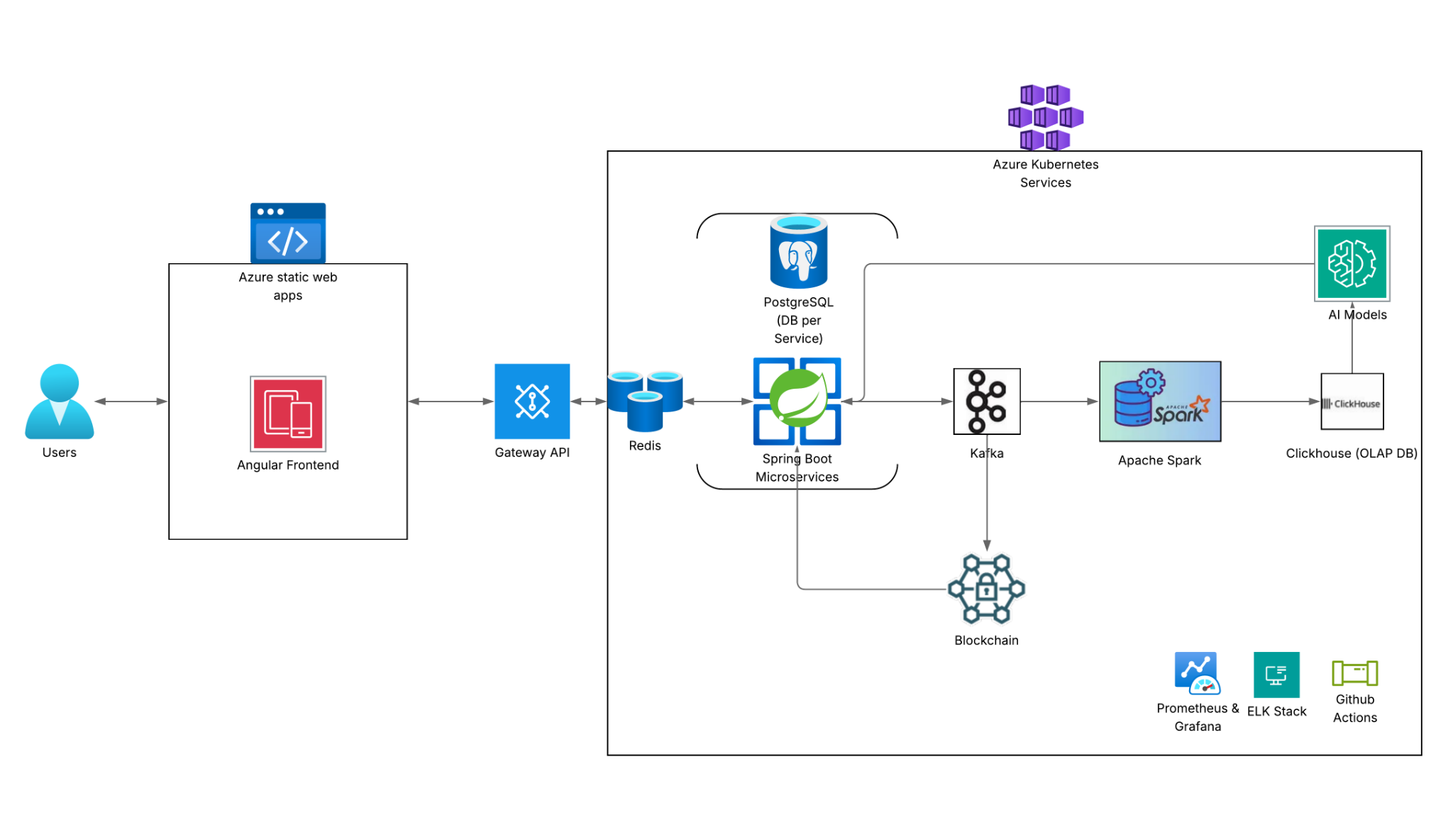
### Blockchain - Επίπεδο Διαφάνειας Συναλλαγών

* Immutable Ledger
* Καταγραφόμενες Ενέργειες
* Διαφάνεια & Έλεγχος
* Τεχνολογία: **Python + web3.py**

### Υποδομή Cloud

* Provider: **Microsoft Azure**
  + Hosting: Azure App Service για web apps, Azure Kubernetes Service (AKS) για microservices.
  + Scalability: Horizontal scaling με auto-scaling rules βασισμένα σε CPU/memory/requests.
  + Load Balancing: Azure Load Balancer για traffic distribution.
* Containerization:
  + **Docker**: Κάθε microservice σε ξεχωριστό container.
  + **Kubernetes (AKS)**: Orchestration, auto-healing, rolling updates.
* CI/CD Pipelines:
  + Source Control: **Git** με GitHub.
  + CI/CD Tool: **GitHub Actions**.
  + Pipeline stages:
    - Code commit → Automated tests (unit, integration).
    - Build Docker images.
    - Push to Container Registry.
    - Deploy to Staging environment.
    - Run E2E tests.
    - Manual approval για Production.
    - Deploy to Production με blue-green deployment.
    - Rollback capability: Instant rollback σε προηγούμενη έκδοση αν χρειαστεί.
* Monitoring & Alerting:
  + **Prometheus**: Metrics collection από όλα τα services.
  + **Grafana**: Dashboards για visualization (uptime, latency, error rates, throughput).
  + Azure Monitor: Application Insights για performance monitoring.
  + Alerting: Automated alerts μέσω email/Slack για:
    - Service downtime (uptime < 99.9%).
    - High latency (> 2 seconds).
    - Error rate spike (> 5%).
    - Resource exhaustion (CPU > 80%, memory > 90%).
  + Logging: Centralized logging με **ELK Stack (Elasticsearch, Logstash, Kibana)**.
* Security:
  + HTTPS: Υποχρεωτική χρήση SSL/TLS για όλες τις επικοινωνίες.
  + Encryption at rest: Database encryption με Azure Disk Encryption.
  + Encryption in transit: TLS 1.3 για API calls.
  + Secrets management: Azure Key Vault για passwords, API keys, certificates.
  + Network security: Virtual Network με Network Security Groups, WAF (Web Application Firewall).
  + DDoS protection: Azure DDoS Protection για αντιμετώπιση επιθέσεων.
* Cost Optimization:
  + Reserved instances για predictable workloads.
  + Spot instances για non-critical batch jobs.
  + Auto-shutdown για development/staging environments εκτός ωρών εργασίας.
  + Resource tagging για cost tracking ανά department/project.

## Τεχνολογική Αρχιτεκτονική



## Πίνακας Προδιαγραφών

| **Στοιχείο** | **Προδιαγραφή** |
| --- | --- |
| **Frontend** | Angular, Responsive design, WCAG 2.1 AA compliance |
| **Backend** | Spring Boot, Microservices & Event-driven architecture |
| **Database** | PostgreSQL (RDBMS), ClickHouse (Data Warehouse) |
| **Messaging Queue** | Kafka |
| **AI & Data analysis** | Python, Apache Spark, Scikit-learn,, TensorFlow, Matplotlib, Seaborn |
| **Blockchain** | Python, web3.py |
| **CI/CD** | GitHub, GitHub Actions |
| **Containerization** | Docker, Kubernetes (AKS) |
| **Cloud** | Azure App Services, Azure Kubernetes Service (AKS) |
| **Monitoring & Logging** | Prometheus, Grafana, ELK Stack |

## Data Schema

### PostgreSQL Schema

| **Οντότητα** | **Πεδία/Κύρια Στοιχεία** | **Περιγραφή** | **Σχέσεις** |
| --- | --- | --- | --- |
| **Χρήστες** | user\_id, name, role, email, created\_at | Αποθήκευση πληροφοριών χρηστών (δημότες, δήμοι, διαχειριστές) | One-to-Many με **Αιτήσεις** |
| **Πόροι** | resource\_id, name, type, status, availability | Αποθήκευση πόρων (π.χ. απορριμματοφόρα, εξοπλισμός) | Many-to-Many με **Αιτήσεις** |
| **Αιτήσεις** | request\_id, user\_id, resource\_id, status, created\_at | Καταγραφή αιτήσεων από χρήστες για συγκεκριμένους πόρους | Many-to-One με **Χρήστες**, Many-to-Many με **Πόρους** |
| **Συναλλαγές** | transaction\_id, user\_id, request\_id, amount | Καταγραφή συναλλαγών που σχετίζονται με τις αιτήσεις | Many-to-One με **Χρήστες**, Many-to-One με **Αιτήσεις** |
| **Αναφορές** | report\_id, user\_id, type, generated\_at | Δημιουργία αναφορών από χρήστες σχετικά με τη χρήση πόρων | One-to-Many με **Χρήστες** |

### ClickHouse Schema

| **Πίνακας** | **Πεδία** | **Περιγραφή** |
| --- | --- | --- |
| **AnalyticsData** | timestamp, resource\_id, usage\_percentage, downtime, active\_time | Δεδομένα ανάλυσης για τη χρήση των πόρων, π.χ. πόσο χρόνο ήταν ενεργοί οι πόροι |
| **TimeSeriesData** | timestamp, resource\_id, location, usage\_time, maintenance\_time, completion\_time | Χρονικά δεδομένα για τη χρήση πόρων και τον χρόνο συντήρησης, για να παρακολουθούνται τα patterns χρήσης |

### Kafka Topics

| **Topic** | **Περιγραφή** | **Σχετιζόμενα δεδομένα** |
| --- | --- | --- |
| **Resource Updates** | Αλλαγές στην κατάσταση των πόρων (π.χ. προσθήκη νέου πόρου, ενημέρωση διαθεσιμότητας) | resource\_id, status, location, availability |
| **Request Status Changes** | Ενημέρωση κατάστασης αιτήσεων (π.χ. από "εκκρεμής" σε "ολοκληρωμένη") | request\_id, status, completion\_time |
| **User Activity Logs** | Καταγραφή δραστηριοτήτων χρηστών (π.χ. δημιουργία αίτησης, αλλαγή κατάστασης) | user\_id, action, timestamp |

### Οντότητες Blockchain

| **Οντότητα** | **Πεδία/Κύρια Στοιχεία** | **Περιγραφή** |
| --- | --- | --- |
| **Transactions** | transaction\_id, timestamp, user\_id, resource\_id, status, transaction\_data | Καταγραφή των συναλλαγών (π.χ. χρήση πόρων, αιτήσεις, πληρωμές) |
| **Audit Logs** | log\_id, transaction\_id, timestamp, action, details | Καταγραφή ενεργειών για παρακολούθηση και έλεγχο της συστήματος |
| **Blockchain Ledger** | block\_id, previous\_block\_hash, timestamp, block\_data | Καταγραφή κάθε block με την πληροφορία του προηγούμενου και των δεδομένων του block |
| **Smart Contracts** | contract\_id, contract\_type, contract\_data, status | Καταγραφή συμφωνιών που εκτελούνται μέσω smart contracts |

# UX Σχεδίαση & Mockups

## User Flow Diagrams +

Δημοτικές Αρχές: καταγραφή πόρων → αίτηση → έγκριση → διάθεση → αναφορές.

Ροή καταγραφής πόρων:

1. Ο υπάλληλος του δήμου συνδέεται στο σύστημα.
2. Επιλέγει “Καταγραφή Πόρων” από το μενού.
3. Συμπληρώνει στοιχεία πόρου (όνομα, ποσότητα, κατηγορία).
4. Πατάει “Αποθήκευση”.
5. Το σύστημα δημιουργεί εγγραφή.
6. Εμφανίζεται μήνυμα επιτυχίας.

Ροή αίτησης πόρων:

1. Ο υπάλληλος του δήμου επιλέγει “Αίτηση Πόρων”.
2. Ο υπάλληλος συμπληρώνει τα στοιχεια τον πόρων που αναζητά.
3. Επιλέγει ποσότητα και τύπο πόρου.
4. Προσθέτει αιτιολόγηση.
5. Παταει “αναζήτηση”.
6. Το σύστημα ξεκινάει την αναζήτηση για διαθέσιμους πόρους στη βάση δεδομένων του.
7. Το σύστημα επιστρέφει τα αποτελέσματα της αναζήτησης.
8. Ο υπάλληλος επιλέγει τον πόρο που αντιστοιχεί στις ανάγκες του.
9. Ο υπάλληλος πατάει “Υποβολή αίτησης”.
10. Το σύστημα καταχωρεί την αίτηση με κατάσταση **“Σε αναμονή έγκρισης”**.

Ροή έγκρισης πόρων:

1. Υπάλληλος με δικαίωμα έγκρισης ανοίγει “Εκκρεμείς Αιτήσεις”.
2. Επιλέγει μία αίτηση.
3. Ελέγχει τα στοιχεία & τη διαθεσιμότητα.
4. Απόφαση:
   1. Έγκριση: Το σύστημα κάνει ενημέρωση αποθεμάτων ενω παράλληλα στέλνει μήνυμα έγκρισης στον αιτούντα.
   2. Απόρριψη: Αποστολή ειδοποίησης απόρριψης στον αιτούντα
5. Το σύστημα καταγράφει την ενέργεια.

Ροή για Αναφορές:

1. Ο υπάλληλος επιλέγει **“Αναφορές”**.
2. Ορίζει φίλτρα (ημερομηνίες, τύπος πόρου, κατάσταση).
3. Επιλέγει το κουμπί “Προβολή αναφοράς”.
4. Το σύστημα προβάλει την αναφορά.
5. Επιλέγει το κουμπί “Εξαγωγή PDF/Excel”.
6. Το σύστημα αποθηκεύει το αρχείο τοπικά στον υπολογιστή του χρήστη.

Πολίτες: δημόσια προβολή πόρων → αναφορές → ειδοποιήσεις.

Ροή για προβολή πόρων και στατιστικών:

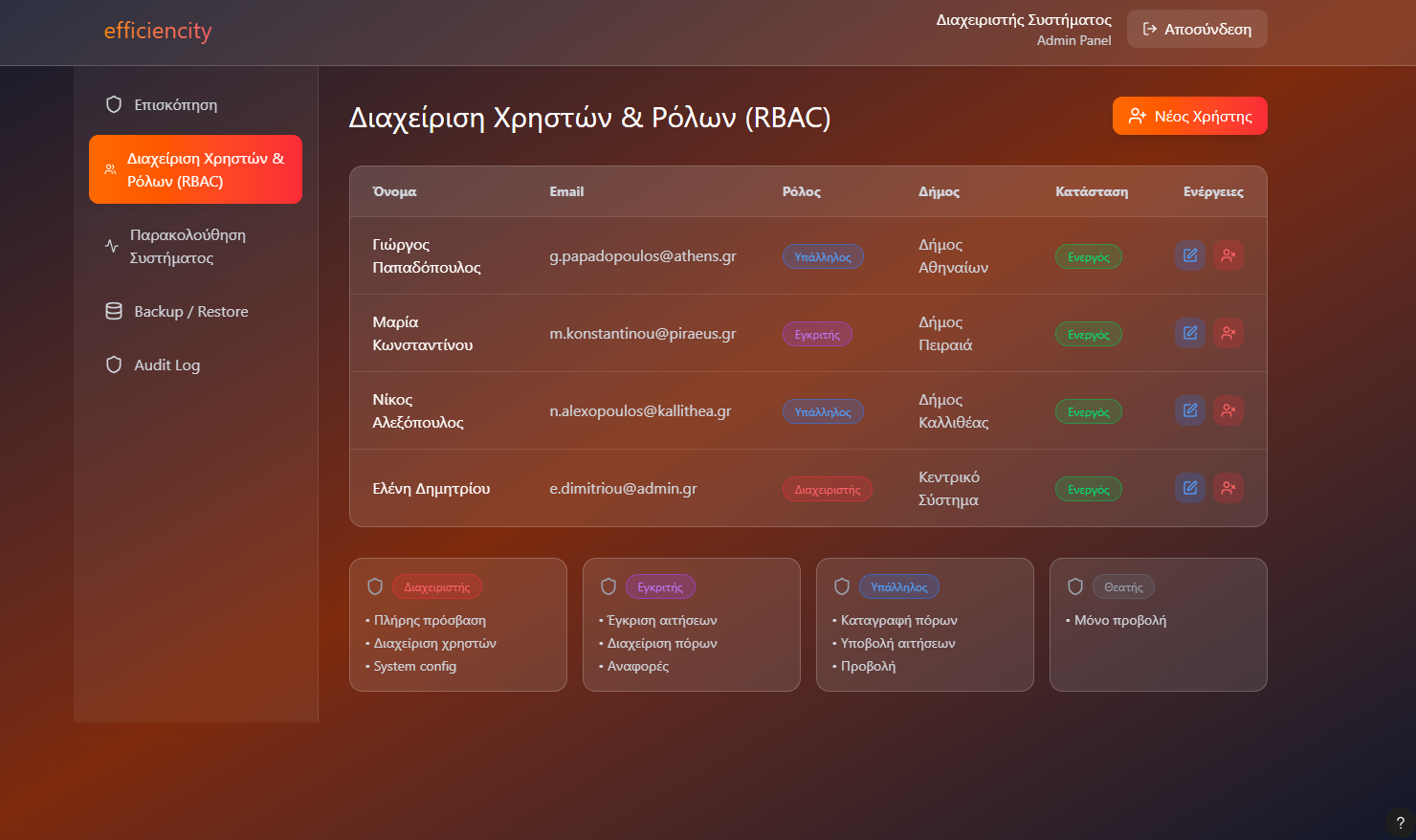
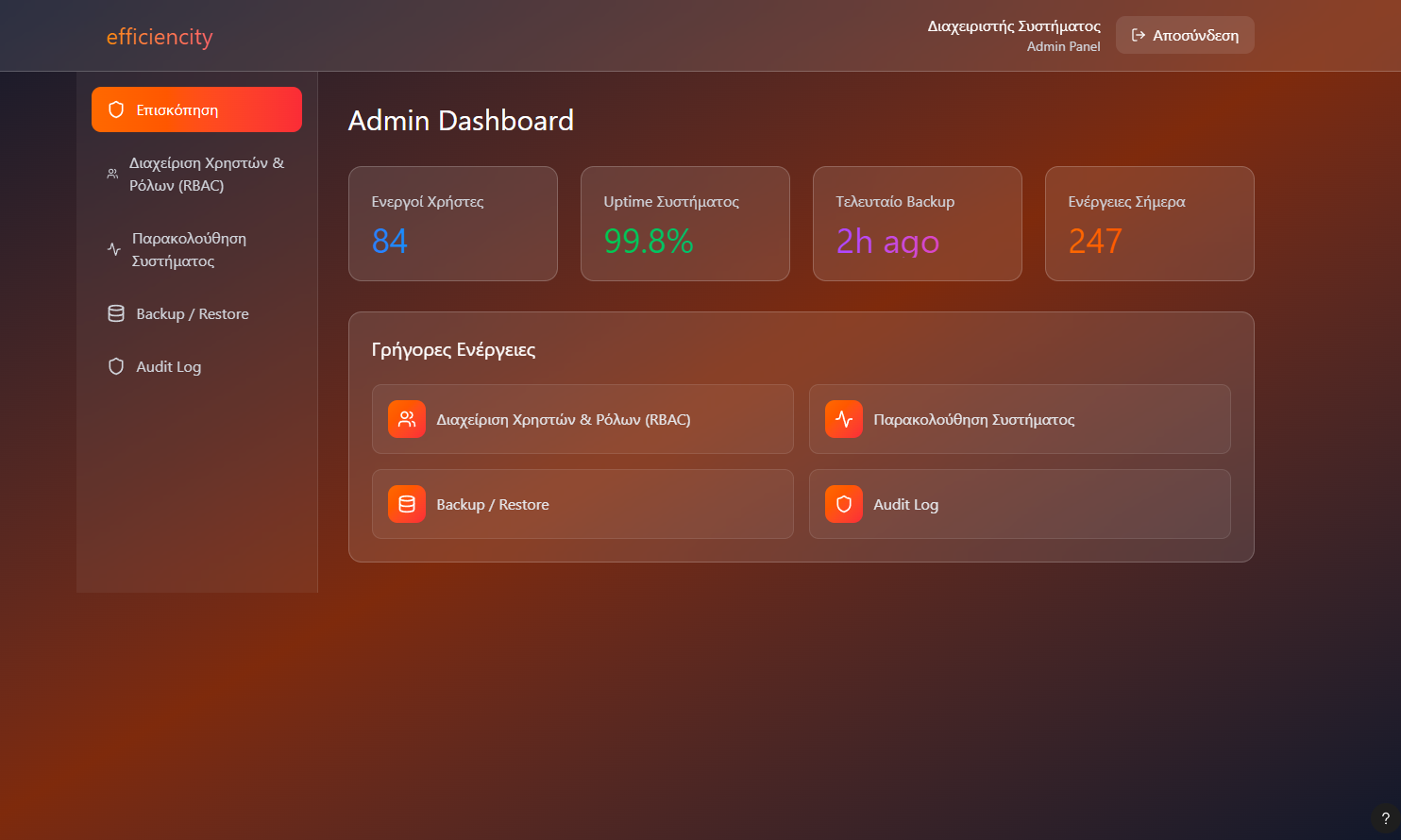
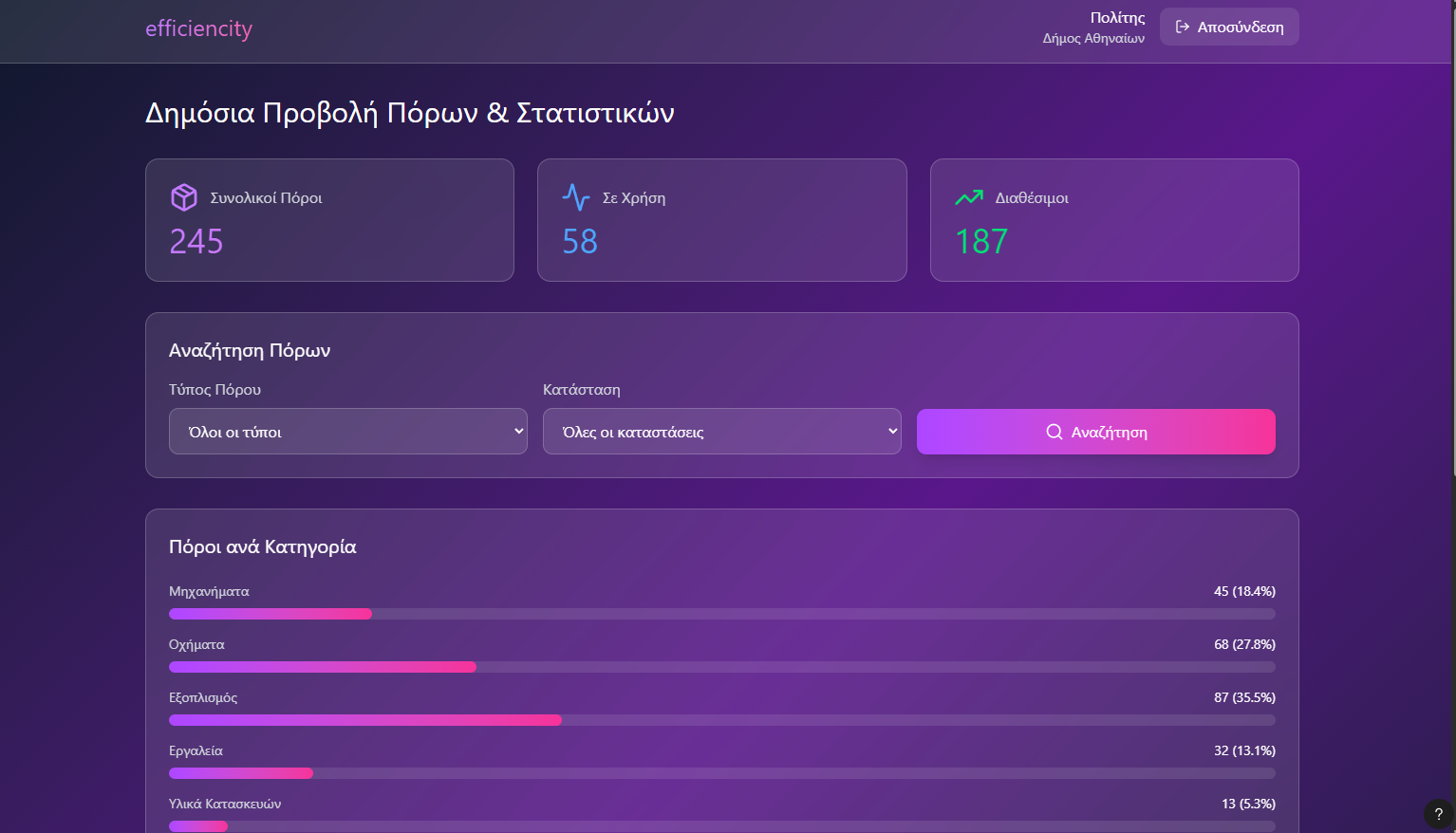
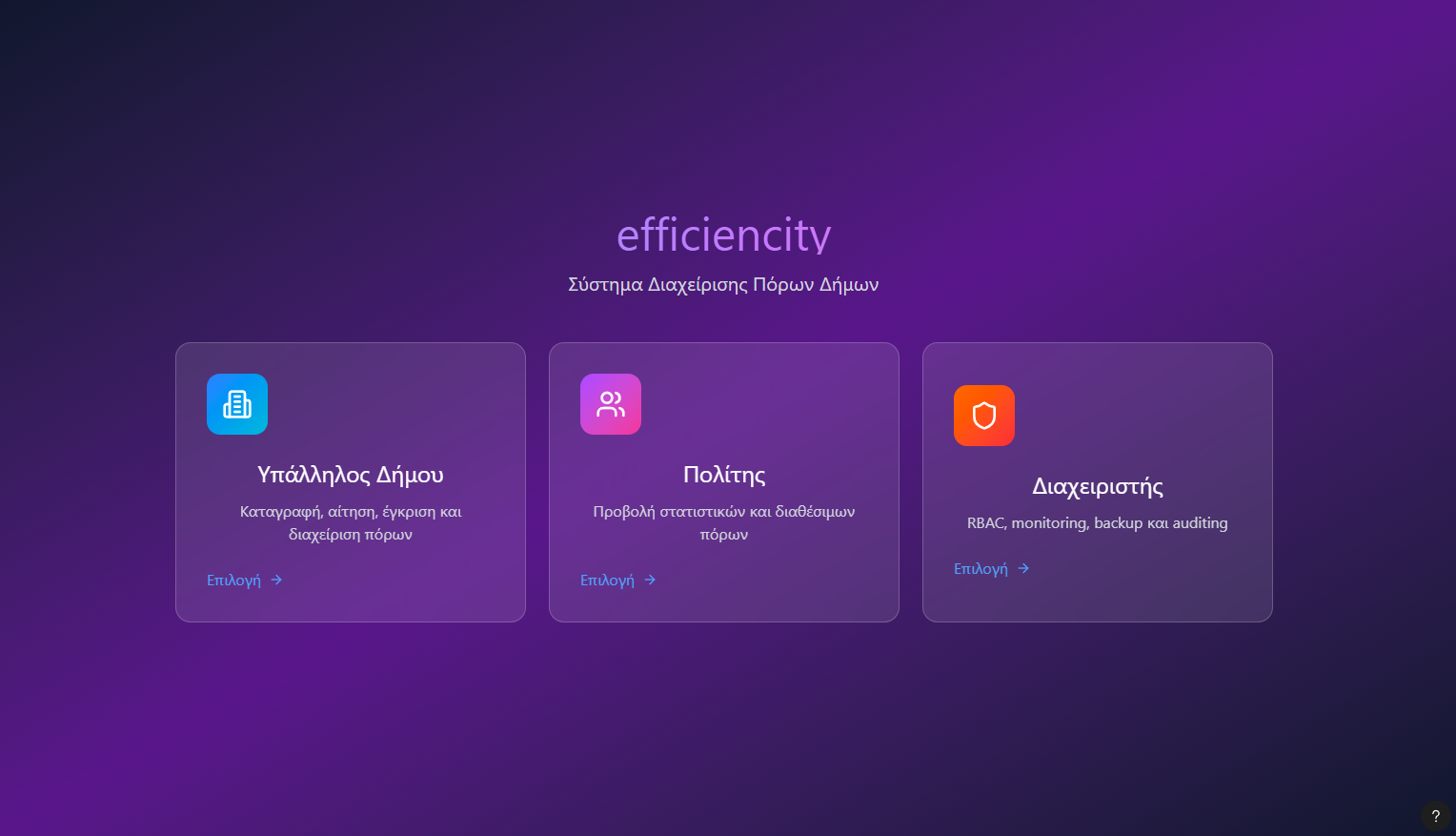
1. Ο πολίτης μπαίνει στη δημόσια πλατφόρμα με κωδικούς taxisnet.
2. Το σύστημα προβάλει την κεντρική σελίδα.
3. Επιλεγει “Προβολή Στατιστικών”
   1. Το σύστημα προβάλει γενικά στατιστικά με τον συνολικό αριθμό όλων των καταγεγραμμένων πόρων, τους πόρους που χρησιμοποιούνται και τους πόρους που μένουν αναξιοποίητοι.
   2. Ο χρηστης ορίζει φιλτρα (τυπος πορου, κατασταση του…)
4. Επιλεγει “Αναζήτηση”.
   1. Το σύστημα κάνει αναζήτηση για τα δεδομένα που ζητάει ο χρήστης.
   2. Το σύστημα προβάλει τα δεδομένα που προκύπτουν από την αναζήτηση του.
5. Επιλέγει “Προβολή συναλλαγών”
   1. Το σύστημα προβάλει το Blockchain

Διαχειριστές Συστήματος: RBAC → monitoring → backup/restore → auditing.

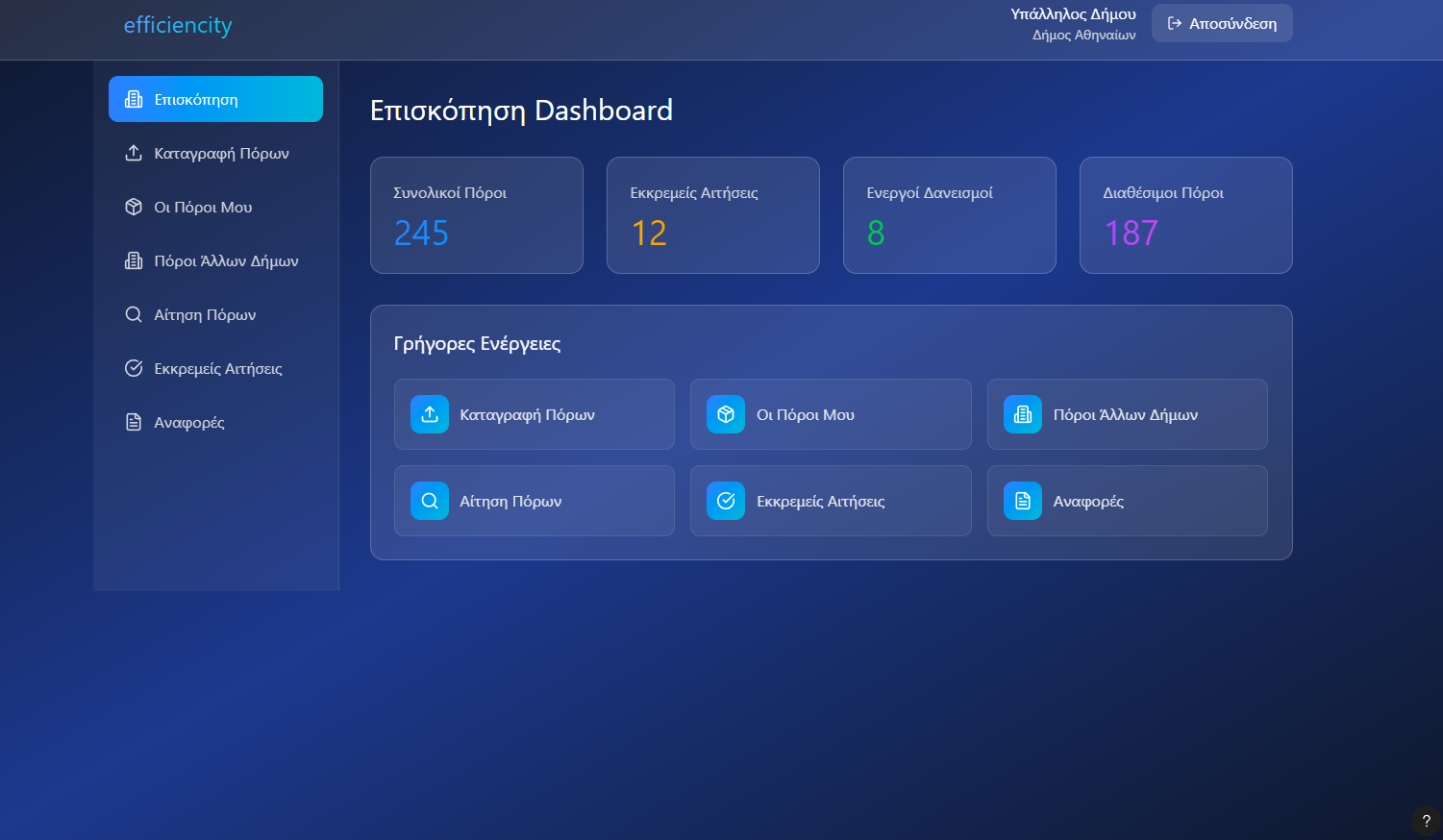
Ροή RBAC (Role-Based Access Control):

1. Ο διαχειριστής συνδέεται στο Admin Panel.
2. Επιλέγει “Διαχείριση Χρηστών & Ρόλων (RBAC)”.
3. Το σύστημα εμφανίζει χρήστες και τρέχοντες ρόλους.
4. Ο διαχειριστής επιλέγει ενέργεια:
   1. Επεξεργασία χρήστη
   2. Ανάθεση/Αφαίρεση ρόλων
   3. Απενεργοποίηση χρήστη
5. Αν δημιουργεί νέο χρήστη, τότε συμπληρώνει στοιχεία και ρόλο.
6. Αν επεξεργάζεται ρόλο, τότε επιλέγει δικαιώματα (read/write/approve/admin).
7. Πατά “Αποθήκευση”.
8. Το σύστημα ενημερώνει τη βάση RBAC και ενεργοποιεί τους νέους κανόνες πρόσβασης.
9. Καταγράφεται καταχώρηση στο Audit Log.

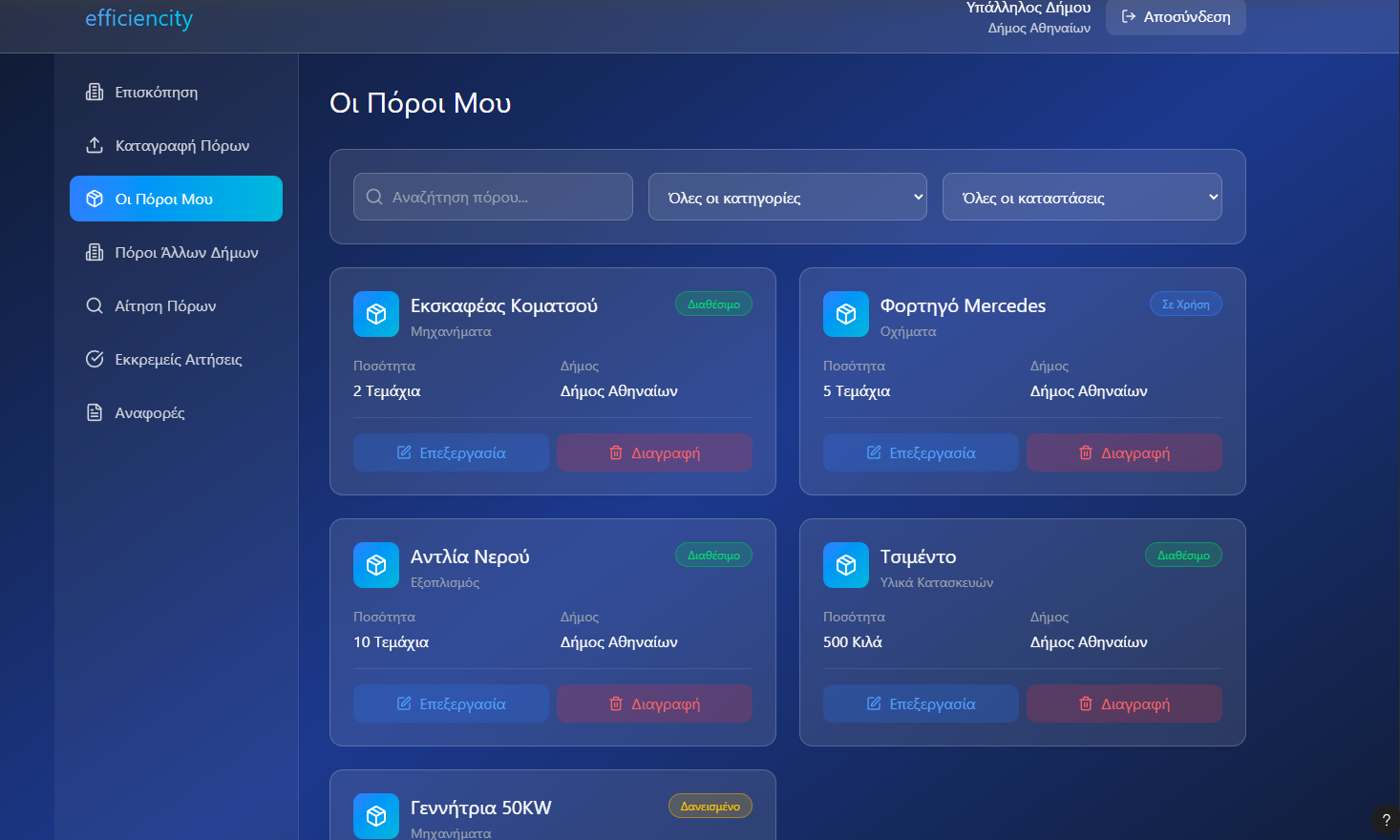
## Mockup Screens +

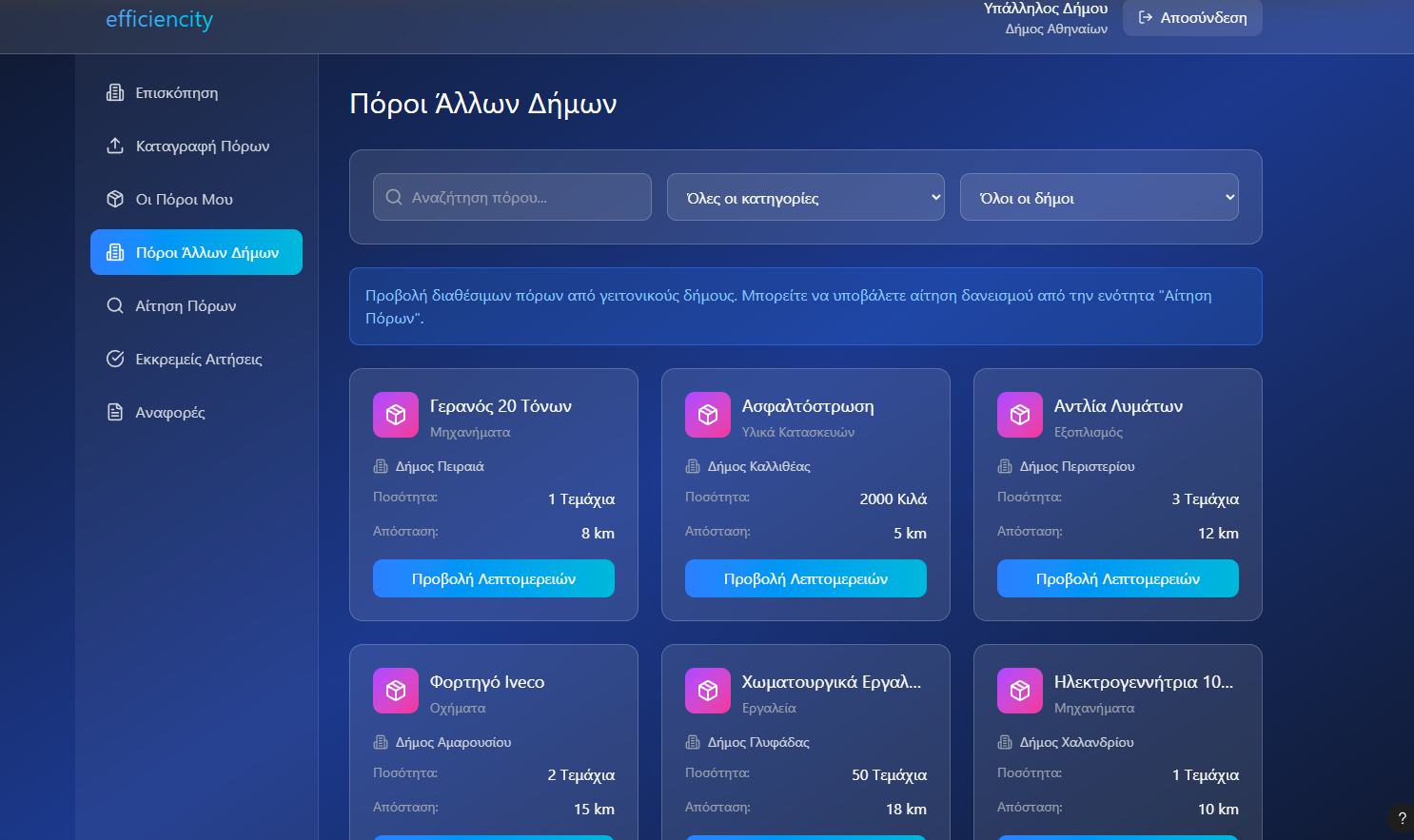


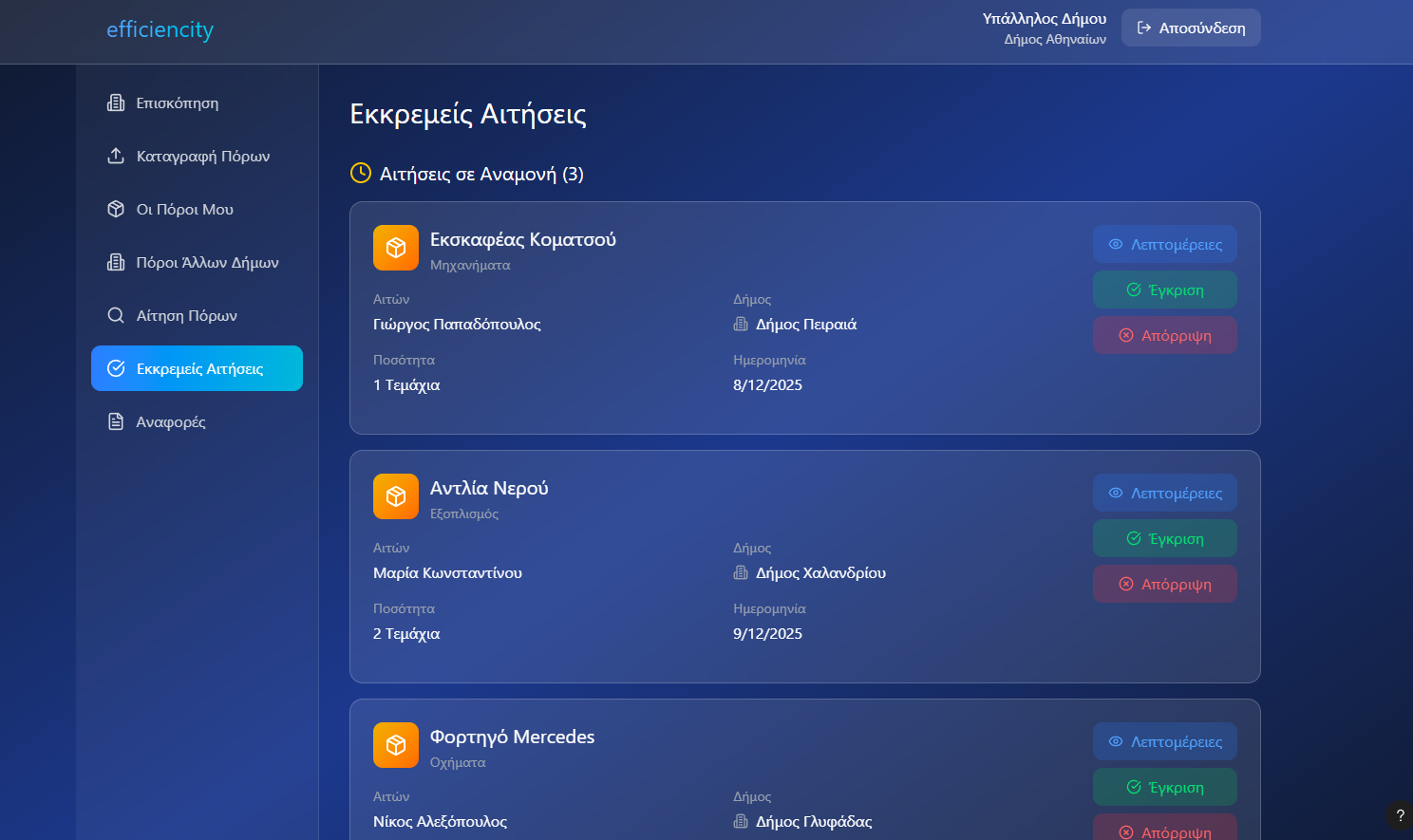


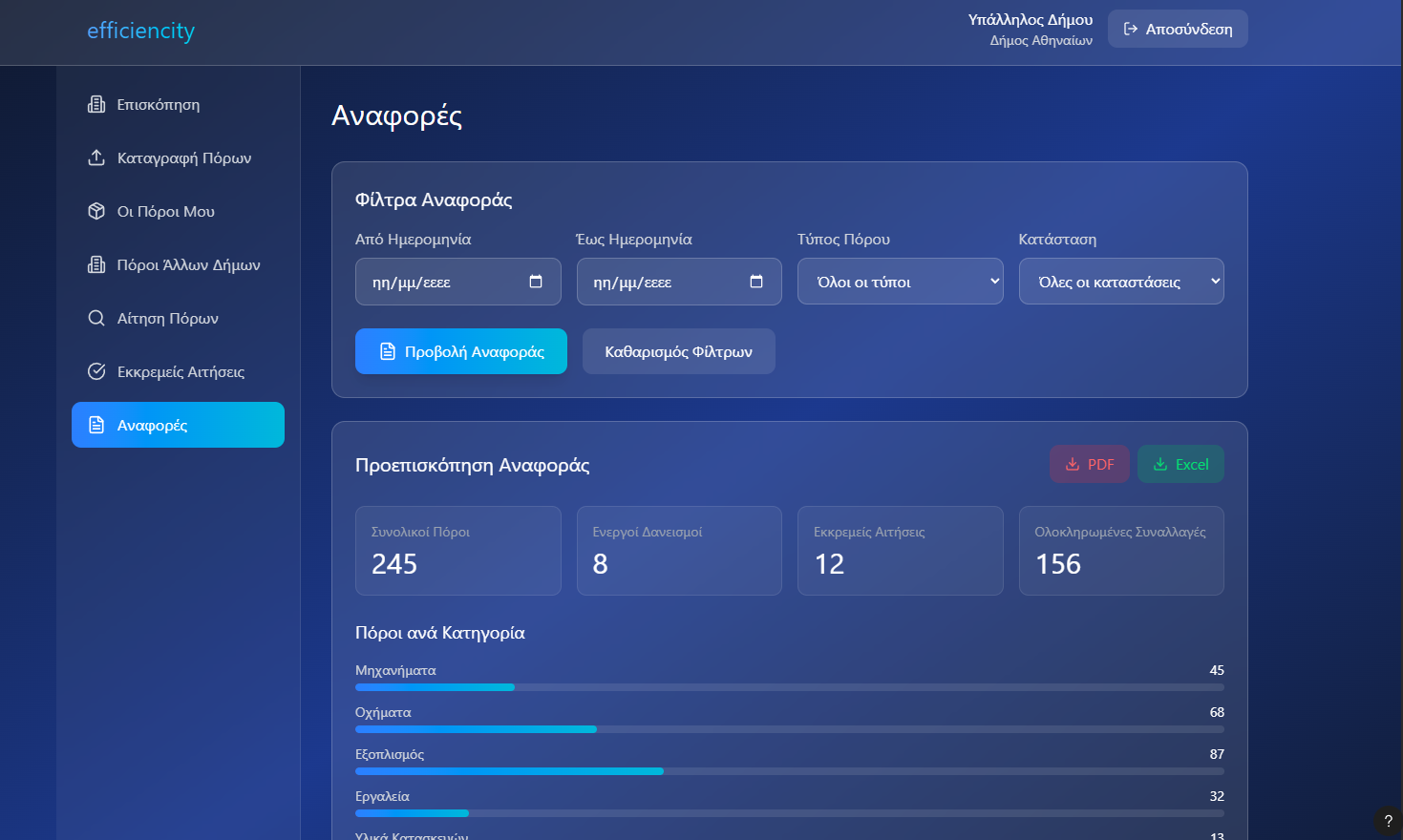












# Έλεγχοι

## Κατηγορίες Δοκιμών

* Unit Tests (Δοκιμές Μονάδας) [Junit, Unittest framework]
  + Παραδείγματα Ελέγχων:
  + Δημιουργία νέου πόρου (resource) με έγκυρα δεδομένα
  + Υπολογισμός διαθεσιμότητας πόρων
  + Validation φόρμας αιτήματος (required fields, format checking)
  + Έλεγχος δικαιωμάτων χρήστη (RBAC logic)
* Integration Tests (Δοκιμές Ενσωμάτωσης) [Postman]
  + Παραδείγματα Ελέγχων:
  + Request Service → Resource Service (αναζήτηση διαθέσιμων πόρων)
  + API Gateway → User Service (authentication & authorization flow)
  + Notification Service → Email/SMS providers (αποστολή ειδοποιήσεων)
  + Analytics Service → ClickHouse (query δεδομένων για reports)
  + Blockchain Service → Ethereum Node (καταγραφή συναλλαγών)
* End-to-End Tests (Δοκιμές Από Άκρη σε Άκρη) [Playwright]
  + Παραδείγματα Σεναρίων:
  + Login → Dashboard → Δημιουργία νέου αιτήματος → Έγκριση → Notification → Completion
  + Πολίτης πλοηγείται στο transparency portal και βλέπει ιστορικό συναλλαγών
  + Admin διαχειρίζεται χρήστες, αλλάζει ρόλους και ελέγχει audit logs
  + Export αναφοράς σε PDF με σωστή μορφοποίηση και δεδομένα
* Performance Tests (Δοκιμές Απόδοσης) [K6]
  + Τύποι Ελέγχων:
  + Load Test: 1000 ταυτόχρονοι χρήστες για 30 λεπτά
  + Stress Test: Σταδιακή αύξηση φορτίου μέχρι το σύστημα να φτάσει στα όριά του
  + Spike Test: Ξαφνική αύξηση traffic (π.χ. από 100 σε 5000 χρήστες)
  + Soak Test: Μακροχρόνια εκτέλεση για εντοπισμό memory leaks
* Security Tests (Δοκιμές Ασφάλειας) [OWASP ZAP]
  + Έλεγχοι:
  + SQL Injection: Προσπάθειες injection σε forms και APIs
  + Authentication bypass: Προσπάθειες unauthorized access
  + GDPR Compliance: Έλεγχος προστασίας προσωπικών δεδομένων
  + Encryption: Επαλήθευση SSL/TLS και data encryption at rest

## Πίνακας Ιχνηλασιμότητας

| **Τι Ελέγχουμε** | **Πως το Ελέγχουμε** | **Τι Θέλουμε να Επιταχύνουμε** |
| --- | --- | --- |
| Καταγραφή και διαχείριση πόρων | Unit + Integration Tests | Ορατότητα σε πραγματικό χρόνο |
| Υποβολή και έγκριση αιτήσεων | Integration + UAT | Ολοκλήρωση σε < 3 ημέρες |
| Διαμοιρασμός πόρων μεταξύ δήμων | Unit Test | Αξιοποίηση ≥ 90% |
| Διαφάνεια προς πολίτες | UAT | Διαθεσιμότητα ≥ 95% |
| Αναφορές και στατιστικά | System + UAT | Πλήρη και ακριβή δεδομένα |
| Ασφάλεια και δικαιώματα | Security Tests | GDPR συμμόρφωση |
| Αντίγραφα ασφαλείας | System Tests | Επαναφορά σε < 2 ώρες |
| Blockchain καταγραφές | Integration + UAT | 100% ιχνηλασιμότητα |

## Δοκιμές & Αποδοχή (User Acceptance Testing - UAT)

* Συμμετέχοντες:
  + Δημοτικοί υπάλληλοι (power users)
  + Διαχειριστές συστήματος
  + Πολίτες (sample group)
* Μέθοδος: Επιλεγμένοι χρήστες χρησιμοποιούν το σύστημα σε ρεαλιστικά σενάρια (π.χ. αίτηση πόρου, διάθεση, έγκριση) για:
  + Συλλογή σχολίων για UI/UX, ευχρηστία, ταχύτητα και κατανόηση διαδικασιών.
  + Προτάσεις για βελτιώσεις στη ροή εργασιών και στα reports.
* Εργαλεία Συλλογής Feedback:
  + Google Forms: Δομημένα ερωτηματολόγια
  + Hotjar: Heatmaps και session recordings για UX analysis
  + Direct interviews: Προσωπικές συνεντεύξεις με key users
* Κριτήρια επιτυχίας:
  + Ολοκλήρωση βασικών εργασιών χωρίς σφάλματα.
  + Αξιολόγηση χρηστών ≥ 4/5 για ευκολία χρήσης και ικανοποίηση.
  + Συμμόρφωση με απαιτήσεις ασφαλείας και προστασία δεδομένων.
* Αποτέλεσμα: Η επιτυχής ολοκλήρωση του UAT επιβεβαιώνει ότι το σύστημα είναι λειτουργικό, ασφαλές και έτοιμο για πιλοτική εφαρμογή και σταδιακή ανάπτυξη στους υπόλοιπους δήμους.

## Σχεδιασμός Πιλοτικού

* Επιλογή δήμου για αρχική εφαρμογή του RMS.
* Παρακολούθηση βασικών KPIs (χρόνος διάθεσης, αξιοποίηση, διαθεσιμότητα).
* Συλλογή feedback από χρήστες μέσω Forms, Hotjar, session recordings.
* Σταδιακή βελτίωση UI/UX και διαδικασιών με βάση τα δεδομένα του πιλοτικού.
* Σταδιακή επέκταση σε όλους τους δήμους μετά την επιτυχή ολοκλήρωση του πιλοτικού. Συνεχής παρακολούθηση (monitoring)
* Προσαρμογή υποδομής και workflow ανάλογα με τα αποτελέσματα, υποστήριξη πλήρους κλιμάκωσης σε Azure Cloud.

# Ενδεικτική Υλοποίηση - Proof of Concept

**Github Link:** <https://github.com/AngelosFikias0/Resource_Management_System>

**Youtube Link:** ….

| **Φάση** | **#** | **Ενέργεια / Στοιχείο** | **Τεχνολογία** | **Στόχος & Εφαρμογή** | **Backend & DB** |
| --- | --- | --- | --- | --- | --- |
| **[ΔΗΜΟΣ]**  **Backend** | 1.1 | **Database Schema** | PostgreSQL | Δημιουργία 4 βασικών πινάκων (User, Resource, Request, Audit\_Log). | Δημιουργία schema με βασικά πεδία |
|  | 1.2 | **Security & Auth** | Spring Security | Υλοποίηση RBAC για ρόλους (ADMIN, MUNICIPALITY, CITIZEN) και έκδοση Tokens. | Setup security με roles και tokens |
|  | 1.3 | **API: Resources** | Spring Boot | Δημιουργία endpoints για CRUD πόρων και εμφάνιση όλων μέσω /resources. | API controller για πόρους |
|  | 1.4 | **API: Requests** | Spring Boot | Δημιουργία endpoints για αιτήσεις (Υποβολή, Έγκριση, Απόρριψη). | API controller για αιτήσεις |
|  | 1.5 | **Caching** | Redis | Ενσωμάτωση Redis για caching στο /resources για βελτίωση απόδοσης. | Redis για caching |
| **Blockchain** | 2 | **Blockchain MVP** | Python, Django | Καταγραφή κρίσιμων ενεργειών στο Audit\_Log με SHA-256 hash. | Blockchain logging service |
| **Frontend** | 3.1 | **Frontend Development** | Angular | Δημιουργία frontend για επικοινωνία με το backend. | Δημιουργία Angular app |
| **Deployment** | 4.1 | **Containerization** | Docker | Δημιουργία Dockerfile για backend και frontend. | Docker για backend και frontend |
|  | 4.2 | **Azure Hosting** | Azure, AKS | Deploy στο Azure (Backend στο App Service, Frontend στο Static Web Apps). | Hosting μέσω Azure |
| **Demo & Docs** | 5.1 | **Testing** | Unit Tests | Συγγραφή unit tests για βασικές λειτουργίες (RBAC, Request Approval). | Unit tests για critical functionality |
|  | 6.1 | **Documentation** | README.md | Δημιουργία README με οδηγίες setup και Demo Flow. | Clear documentation |
|  | 6.2 | **YouTube Tutorial** | YouTube | Δημιουργία tutorial video για παρουσίαση του έργου. | Demo video tutorial |
| **Technologies Used** | - | **Summary** | PostgreSQL, Spring Boot, Redis, Python, Django, Angular, Docker, Azure | Χρήση για DB, Backend, Frontend, Caching, Blockchain, Containerization, Hosting |  |

# Συμπεράσματα

Το EfficienCity RMS προσφέρει μια ολοκληρωμένη λύση που μετατρέπει τον κάθε δήμο σε έναν τεχνολογικό κόμβο, ενοποιώντας πόρους, δεδομένα και διαδικασίες σε μια ενιαία πλατφόρμα. Η πλήρης καταγραφή και παρακολούθηση των πόρων, σε συνδυασμό με analytics και KPI, επιτρέπει τεκμηριωμένες αποφάσεις, βελτιώνει την απόδοση και μειώνει το κόστος. Το σύστημα ενισχύει την διαφάνεια και την σωστή αξιοποίηση με αποτέλεσμα να βελτιώνει την ποιότητα ζωής των πολιτών.

Notion Link: <https://www.notion.so/2bd9c27e8f1180909629d6343b3cd4b7?v=2bd9c27e8f1180bd94c8000cddcf1586&source=copy_link>