



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ**  
**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΠΛΗΡΟΦΟΡΙΑΚΑ & ΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

**Σημασιολογική πλατφόρμα για τη διαχείριση και  
αναζήτηση DPPs**

**(A semantic platform for the management and search of  
DPPs)**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

της

**Ζωής Ν. Λεβεσανού**

**Επιβλέπων:** Κυριάκος Κρητικός

**Μέλη εξεταστικής επιτροπής:** Σπυρίδων Κοκολάκης, Γιάννης Χαραλαμπίδης

Σάμος, [Μήνας Έτος]

Η σελίδα αυτή είναι σκόπιμα λευκή.

## **Πρόλογος και ευχαριστίες**

Πρώτη παράγραφος

Επόμενη παράγραφος....

© [ΕΤΟΣ]

της

**ΖΩΗ Ν. ΛΕΒΕΣΑΝΟΥ**

**Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων**

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ**

Η σελίδα αυτή είναι σκόπιμα λευκή.

## Πίνακας περιεχομένων

|                                                                                |           |
|--------------------------------------------------------------------------------|-----------|
| <b>1 Εισαγωγή</b>                                                              | <b>1</b>  |
| 1.1 Ψηφιακά διαβατήρια προϊόντων και σημασιολογική διαχείριση πληροφορίας      | 1         |
| 1.2 Αντικείμενο διπλωματικής                                                   | 2         |
| 1.3 Συνεισφορά και πλεονεκτήματα                                               | 3         |
| 1.4 Δομή της διπλωματικής                                                      | 3         |
| <b>2 Υπόβαθρο και τεχνολογίες</b>                                              | <b>4</b>  |
| 2.1 Τι είναι τα ψηφιακά διαβατήρια προϊόντων (DPPs)                            | 4         |
| 2.2 Νομοθετικό πλαίσιο και ευρωπαϊκές πρωτοβουλίες για τα DPPs                 | 5         |
| 2.3 Πρότυπα και τεχνολογίες αναπαράστασης DPPs                                 | 7         |
| 2.4 Υλοποιήσεις και πρωτοβουλίες σχετικές με τα DPPs                           | 9         |
| 2.5 Πληροφοριακά πεδία των DPPs & βασικές κατηγορίες πληροφορίας που καλύπτουν | 10        |
| 2.6 Συσχέτιση με το μοντέλο asset administration shell (AAS)                   | 13        |
| <b>3 Σχετικές εργασίες</b>                                                     | <b>16</b> |
| 3.1 CIRPASS project                                                            | 16        |
| 3.2 Battery pass project                                                       | 17        |
| 3.3 iPoint DPP framework                                                       | 18        |
| 3.4 Συγκριτική αξιολόγηση σχετικών εργασιών                                    | 19        |
| <b>4 Ανάπτυξη Συστήματος</b>                                                   | <b>23</b> |
| 4.1 Μοντέλο ανάπτυξης: καταρράκτη (waterfall model)                            | 23        |
| 4.2 Ανάλυση απαιτήσεων                                                         | 24        |
| 4.2.1 Λειτουργικές απαιτήσεις                                                  | 24        |
| 4.2.2 Μη λειτουργικές απαιτήσεις                                               | 31        |
| 4.2.3 Παρατηρήσεις και τεκμηρίωση                                              | 31        |
| 4.3 Σχεδίαση συστήματος                                                        | 32        |
| 4.3.1 Διάγραμμα περιβάλλοντος                                                  | 32        |
| 4.3.2 Διάγραμμα συστατικών                                                     | 35        |
| 4.3.3 Διαγράμματα περιπτώσεων χρήσης                                           | 38        |
| 4.3.4 Διάγραμμα κλάσεων                                                        | 42        |
| 4.3.5 Διαγράμματα ακολουθίας                                                   | 46        |
| 4.3.6 Διάγραμμα δραστηριοτήτων                                                 | 53        |
| 4.4 Υλοποίηση συστήματος                                                       | 59        |
| 4.4.1 Τεχνολογίες και περιβάλλον ανάπτυξης                                     | 59        |
| 4.4.2 Διαχείριση αρχείων και εξαγωγή δεδομένων                                 | 61        |
| 4.4.3 Υποδομή και οργάνωση πηγαίου κώδικα                                      | 62        |
| 4.4.4 Επικοινωνία frontend - backend                                           | 64        |
| 4.5 Επαλήθευση και αξιολόγηση                                                  | 64        |
| 4.5.1 Μέθοδος ελέγχου ποιότητας                                                | 64        |
| 4.5.2 Εργαλεία και περιβάλλον δοκιμών Backend                                  | 68        |
| 4.5.3 Εργαλεία και περιβάλλον δοκιμών Frontend                                 | 74        |
| 4.5.4 Ενδεικτικά σενάρια δοκιμών                                               | 76        |

|                                                                                                   |            |
|---------------------------------------------------------------------------------------------------|------------|
| 4.5.5 Συμπεράσματα δοκιμών                                                                        | 84         |
| 4.5.6 Ικανοποίηση απαιτήσεων                                                                      | 84         |
| <b>5 Εγκατάσταση και επίδειξη συστήματος</b>                                                      | <b>88</b>  |
| 5.1 Εγκατάσταση Συστήματος                                                                        | 88         |
| 5.1.1 Αυτόματη εγκατάσταση εφαρμογής                                                              | 88         |
| 5.1.2 Χειροκίνητη εγκατάσταση εφαρμογής                                                           | 90         |
| 5.1.2.1 Χειροκίνητη εγκατάσταση backend                                                           | 90         |
| 5.1.2.2 Χειροκίνητη εγκατάσταση frontend                                                          | 91         |
| 5.2 Επίδειξη συστήματος                                                                           | 92         |
| 5.2.1 Σενάριο 1: Εγγραφή και Σύνδεση Χρήστη (Ρόλος Viewer)                                        | 92         |
| 5.2.2 Σενάριο 2: Επισκόπηση και Πλοήγηση στο Σύστημα (Ρόλος User)                                 | 96         |
| 5.2.3 Σενάριο 3: Δημιουργία και Διαχείριση Ψηφιακού Διαβατηρίου (Ρόλος User)                      | 98         |
| 5.2.4 Σενάριο 4: Οπτικοποίηση Σημασιολογικού Γράφου (Semantic Graph Visualization)                | 104        |
| 5.2.5 Σενάριο 5: Σημασιολογικά ερωτήματα (SPARQL Query Editor)                                    | 106        |
| 5.2.6 Σενάριο 6: Διαχείριση Χρηστών και Δικαιωμάτων (Ρόλος Admin)                                 | 109        |
| 5.2.7 Λοιπές Λειτουργίες και Σενάρια                                                              | 113        |
| <b>6 Συμπεράσματα και μελλοντική επέκταση</b>                                                     | <b>114</b> |
| 6.1 Συμπεράσματα                                                                                  | 114        |
| 6.2 Μελλοντική επέκταση                                                                           | 115        |
| <b>Βιβλιογραφία</b>                                                                               | <b>117</b> |
| <b>Παράρτημα I: Τεκμηρίωση REST API</b>                                                           | <b>124</b> |
| I.1 Authentication (Αυθεντικοποίηση)                                                              | 124        |
| I.2 Users (Διαχείριση χρηστών)                                                                    | 125        |
| I.3 DPP JSON (Διαχείριση διαβατηρίων)                                                             | 128        |
| I.4 DPP SPARQL (Σημασιολογικά ερωτήματα)                                                          | 131        |
| I.5 DPP export & files                                                                            | 132        |
| I.6 System (Σύστημα)                                                                              | 134        |
| I.7 Metrics & Monitoring (Παρακολούθηση)                                                          | 136        |
| <b>Παράρτημα II: Παρακολούθηση Επιδόσεων και Υγείας Συστήματος (Grafana Monitoring Dashboard)</b> | <b>137</b> |
| II.1 System Health: Κατάσταση και Πόροι Συστήματος                                                | 137        |
| II.2 Παρακολούθηση Υλικών Πόρων (CPU & RAM)                                                       | 137        |
| II.3 API Performance: Ανάλυση Κίνησης και Απόκρισης                                               | 138        |

## Λίστα Σχημάτων

|                                                                                                                                                                                                                                                                                                                                                                                                |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Εικόνα 1: Ο κύκλος ζωής του προϊόντος και ο ρόλος του DPP στην κυκλική οικονομία. Πηγή: Finboot (2022) [22].....                                                                                                                                                                                                                                                                               | 6  |
| Εικόνα 2: Ο κύκλος ζωής ενός Ψηφιακού Διαβατηρίου Προϊόντος. Πηγή: CIRPASS Project (2023) [5].....                                                                                                                                                                                                                                                                                             | 11 |
| Εικόνα 3: Η αρχιτεκτονική δομή του Asset Administration Shell (AAS) και η οργάνωση των υπο-μοντέλων (submodels).....                                                                                                                                                                                                                                                                           | 14 |
| Εικόνα 4: Διάγραμμα περιβάλλοντος της πλατφόρμας διαχείρισης DPPs.....                                                                                                                                                                                                                                                                                                                         | 33 |
| Εικόνα 5: Διάγραμμα συστατικών της πλατφόρμας διαχείρισης DPPs.....                                                                                                                                                                                                                                                                                                                            | 37 |
| Εικόνα 6: Διάγραμμα περιπτώσεων χρήσης: Διαχείριση Χρηστών.....                                                                                                                                                                                                                                                                                                                                | 40 |
| Εικόνα 7: Διάγραμμα περιπτώσεων χρήσης: Διαχείριση Προϊόντων.....                                                                                                                                                                                                                                                                                                                              | 42 |
| Εικόνα 8: Διάγραμμα κλάσεων της πλατφόρμας διαχείρισης DPPs.....                                                                                                                                                                                                                                                                                                                               | 44 |
| Εικόνα 9: Διάγραμμα κλάσεων του σημασιολογικού μοντέλου της πλατφόρμας DPP.....                                                                                                                                                                                                                                                                                                                | 46 |
| Εικόνα 10: Διάγραμμα ακολουθίας εγγραφής.....                                                                                                                                                                                                                                                                                                                                                  | 47 |
| Εικόνα 11: Διάγραμμα ακολουθίας σύνδεσης.....                                                                                                                                                                                                                                                                                                                                                  | 48 |
| Εικόνα 12: Διάγραμμα ακολουθίας δημιουργίας νέου DPP.....                                                                                                                                                                                                                                                                                                                                      | 49 |
| Εικόνα 13: Διάγραμμα ακολουθίας αναζήτησης DPP.....                                                                                                                                                                                                                                                                                                                                            | 51 |
| Εικόνα 14: Διάγραμμα ακολουθίας εκτέλεσης SPARQL ερωτήματος.....                                                                                                                                                                                                                                                                                                                               | 52 |
| Εικόνα 15: Διάγραμμα δραστηριοτήτων διαχείρισης κατάστασης δημοσίευσης DPP.....                                                                                                                                                                                                                                                                                                                | 54 |
| Εικόνα 16: Διάγραμμα δραστηριοτήτων εισαγωγής αρχείου AASX.....                                                                                                                                                                                                                                                                                                                                | 55 |
| Εικόνα 17: Διάγραμμα δραστηριοτήτων ενημέρωσης προφίλ χρήστη.....                                                                                                                                                                                                                                                                                                                              | 57 |
| Εικόνα 18: Διάγραμμα δραστηριοτήτων διαγραφής DPP.....                                                                                                                                                                                                                                                                                                                                         | 58 |
| Εικόνα 19: Συνοπτική παρουσίαση των API Endpoints και των θεματικών ενοτήτων μέσω του εργαλείου Swagger UI.....                                                                                                                                                                                                                                                                                | 61 |
| Εικόνα 20: Επιτυχής διαδικασία αυθεντικοποίησης (Login) και έκδοση του ψηφιακού κλειδιού πρόσβασης (access_token).....                                                                                                                                                                                                                                                                         | 66 |
| Εικόνα 21: Ενσωμάτωση μηχανισμού αυθεντικοποίησης Bearer Token (JWT) στο σχήμα ασφαλείας του API, διασφαλίζοντας την εξουσιοδοτημένη πρόσβαση σε προστατευμένους πόρους (Protected Resources).....                                                                                                                                                                                             | 66 |
| Εικόνα 22: Παραμετροποίηση αιτήματος προηγμένης αναζήτησης (advanced search) με χρήση φίλτρων μερικού ταιριάσματος (partial match) στο Swagger UI.....                                                                                                                                                                                                                                         | 67 |
| Εικόνα 23: Αποτέλεσμα επιτυχούς ανάκτησης δεδομένων σε μορφή JSON, μετά την εφαρμογή του μηχανισμού flattening για την ανάγνωση των υπομοντέλων (submodels).....                                                                                                                                                                                                                               | 67 |
| Εικόνα 24: Παράδειγμα δοκιμής μονάδας (unit test) - Ο κώδικας ελέγχει την ορθή λειτουργία των συναρτήσεων κρυπτογράφησης (hash_password) και επαλήθευσης (verify_password) κωδικών πρόσβασης, διασφαλίζοντας ότι οι κωδικοί δεν αποθηκεύονται ως απλό κείμενο και ότι η διαδικασία ταυτοποίησης είναι αξιόπιστη.....                                                                           | 70 |
| Εικόνα 25: Παράδειγμα δοκιμής ενοποίησης (integration test) - Ο κώδικας προσομοιώνει μια πλήρη ροή χρήστη (End-to-End scenario), ξεκινώντας από την αυθεντικοποίηση (login) για την απόκτηση token και προχωρώντας στη δημιουργία ενός νέου DPP. Στο τέλος, επαληθεύεται ότι η εγγραφή δημιουργήθηκε σωστά στη βάση δεδομένων και ότι η απάντηση του API περιέχει τα αναμενόμενα δεδομένα..... | 71 |

|                                                                                                                                                                                                                                                                                                                                               |     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Εικόνα 26: Περιβάλλον ανάπτυξης με τον κώδικα των integration tests και το αποτέλεσμα της επιτυχούς εκτέλεσης στο terminal.....                                                                                                                                                                                                               | 71  |
| Εικόνα 27: Επιτυχής εκτέλεση λειτουργικής δοκιμής (Functional Test) στο endpoint αναζήτησης /dpp/json/search μέσω του Postman, με επαλήθευση της ορθής δομής των δεδομένων του ψηφιακού διαβατηρίου (JSON response).....                                                                                                                      | 73  |
| Εικόνα 28: Διαδικασία εξαγωγής δεδομένων DPP και επαλήθευση της απόκρισης μέσω του Swagger UI.....                                                                                                                                                                                                                                            | 74  |
| Εικόνα 29: : Δοκιμές Μονάδας Frontend (React Testing Library) - Έλεγχος της φόρμας εισόδου (Login Component).....                                                                                                                                                                                                                             | 75  |
| Εικόνα 30: Στιγμιότυπο επιτυχούς εκτέλεσης των μονάδων ελέγχου (Unit Tests) του Frontend, όπου επιβεβαιώνεται η ορθή λειτουργία των βασικών συστατικών και η σωστή διαχείριση ασφαλιμάτων.....                                                                                                                                                | 75  |
| Εικόνα 31: Επιτυχής αρχική εγγραφή χρήστη μέσω του Postman, η οποία χρησιμοποιείται ως βάση για τον έλεγχο των περιορισμών μοναδικότητας.....                                                                                                                                                                                                 | 77  |
| Εικόνα 32: Απόρριψη αίτησης εγγραφής από το API λόγω προϋπάρχουσας ηλεκτρονικής διεύθυνσης στη βάση δεδομένων (HTTP 400 Bad Request).....                                                                                                                                                                                                     | 78  |
| Εικόνα 33: Αποτυχία αυθεντικοποίησης στο endpoint /auth/login λόγω παροχής λανθασμένων στοιχείων, με την επιστροφή κατάλληλου μηνύματος σφάλματος από το API.....                                                                                                                                                                             | 79  |
| Εικόνα 34: Αυτόματη επικύρωση σχήματος δεδομένων από το backend και απόρριψη αιτήματος λόγω παράλειψης υποχρεωτικών πεδίων (HTTP 422 Validation Error).....                                                                                                                                                                                   | 80  |
| Εικόνα 35: Εκτέλεση προηγμένης αναζήτησης με φιλτράρισμα βάσει κατασκευαστή και επιβεβαίωση της ορθής ανάκτησης των σχετικών εγγραφών.....                                                                                                                                                                                                    | 81  |
| Εικόνα 36: Διαδικασία ανάκτησης για το αναγνωριστικό dpp_id από τον ιδιοκτήτη (John), το οποίο παραμένει σε μη δημοσιευμένη κατάσταση.....                                                                                                                                                                                                    | 82  |
| Εικόνα 37: Απόρριψη μη εξουσιοδοτημένου SPARQL ερωτήματος από το API (HTTP 403 Forbidden), διασφαλίζοντας την ιδιωτικότητα των δεδομένων του John.....                                                                                                                                                                                        | 83  |
| Εικόνα 38: Απόσπασμα αρχείου ρυθμίσεων περιβάλλοντος (.env) Το αρχείο περιλαμβάνει τις απαραίτητες μεταβλητές για τη σύνδεση με τη βάση δεδομένων PostgreSQL (εντός Docker container), τις παραμέτρους ασφαλείας για την κρυπτογράφηση των JWT tokens, καθώς και τις ρυθμίσεις διασύνδεσης με τον σημασιολογικό γράφο Apache Jena Fuseki..... | 89  |
| Εικόνα 39: Οθόνη εισόδου (Αρχική Σελίδα Εφαρμογής).....                                                                                                                                                                                                                                                                                       | 92  |
| Εικόνα 40: Η διεπαφή εγγραφής νέου χρήστη με ρόλο Επισκέπτη (Viewer).....                                                                                                                                                                                                                                                                     | 93  |
| Εικόνα 41: Επιτυχής ολοκλήρωση εγγραφής και αυτόματη ανακατεύθυνση.....                                                                                                                                                                                                                                                                       | 94  |
| Εικόνα 42: Εισαγωγή διαπιστευτηρίων για σύνδεση στο σύστημα.....                                                                                                                                                                                                                                                                              | 95  |
| Εικόνα 43: Ο Πίνακας Ελέγχου για τον ρόλο Viewer, με περιορισμένες επιλογές πλοήγησης.....                                                                                                                                                                                                                                                    | 96  |
| Εικόνα 44: Κεντρική διεπαφή χρήστη (ρόλος User) του συστήματος (Dashboard) με συγκεντρωτική απεικόνιση των DPPs και εργαλεία διαχείρισης.....                                                                                                                                                                                                 | 97  |
| Εικόνα 45: Διαδικασία επιλογής της ενότητας "Ελληνικά" από το πλευρικό μενού πλοήγησης.....                                                                                                                                                                                                                                                   | 98  |
| Εικόνα 46: Δυνατότητα εναλλαγής γλώσσας (Localization) της πλατφόρμας μεταξύ Ελληνικών και Αγγλικών.....                                                                                                                                                                                                                                      | 98  |
| Εικόνα 47: Εκκίνηση της διαδικασίας δημιουργίας νέου διαβατηρίου από τον Πίνακα Ελέγχου..                                                                                                                                                                                                                                                     | 99  |
| Εικόνα 48: Η φόρμα δημιουργίας νέου DPP, με πεδία για τα βασικά χαρακτηριστικά του προϊόντος.....                                                                                                                                                                                                                                             | 100 |
| Εικόνα 49: Προσθήκη υπο-μοντέλων AAS (Submodels) για την καταγραφή εξειδικευμένων δεδομένων.....                                                                                                                                                                                                                                              | 101 |

|                                                                                                                                                                                                                                 |     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Εικόνα 50: Χειροκίνητη εισαγωγή δεδομένων για τη δημιουργία νέου DPP.....                                                                                                                                                       | 102 |
| Εικόνα 51: Επιβεβαίωσης επιτυχούς καταχώρησης του νέου DPP, όπου προβάλλεται το αντίστοιχο μήνυμα ολοκλήρωσης και η ενημέρωση για την αυτόματη ανακατεύθυνση του χρήστη στο Dashboard.....                                      | 102 |
| Εικόνα 52: Εμφάνιση του νέου DPP στον Πίνακα Ελέγχου. Διακρίνονται τα εργαλεία διαχείρισης (Οπτικοποίηση, Δημοσίευση, Εξαγωγή, Διαγραφή) που είναι διαθέσιμα στον ιδιοκτήτη.....                                                | 103 |
| Εικόνα 53: Ενημέρωση κατάστασης σε "Published".....                                                                                                                                                                             | 104 |
| Εικόνα 54: Επιλογή εικονιδίου δικτύου για προβολή του οπτικοποιημένου γράφου.....                                                                                                                                               | 105 |
| Εικόνα 55: Διαδραστική οπτικοποίηση του σημασιολογικού γράφου. Οι κόμβοι αναπαριστούν τις οντότητες (Προϊόν, Υπο-μοντέλα) και οι ακμές τις μεταξύ τους σχέσεις.....                                                             | 105 |
| Εικόνα 56: Προβολή του κεντρικού πίνακα ελέγχου (Dashboard) του χρήστη και διαδικασία επιλογής της ενότητας "SPARQL Query" από το πλευρικό μενού πλοήγησης.....                                                                 | 107 |
| Εικόνα 57: Ο επεξεργαστής ερωτημάτων SPARQL. Διακρίνεται το πλαίσιο οδηγιών για τη χρήση του GRAPH clause.....                                                                                                                  | 107 |
| Εικόνα 58: Χρήση προκαθορισμένων προτύπων (templates) στον SPARQL Query Editor για την ταχεία ανάκτηση συγκεκριμένων πληροφοριών, όπως περιβαλλοντικά δεδομένα, χωρίς την ανάγκη χειροκίνητης σύνταξης κώδικα.....              | 108 |
| Εικόνα 59: Σύνταξη ερωτήματος SPARQL με χρήση προτύπου. Ο χρήστης ορίζει ρητά το επιθυμητό γράφημα για λόγους ασφαλείας.....                                                                                                    | 108 |
| Εικόνα 60: Αποτελέσματα του σημασιολογικού ερωτήματος. Εμφανίζεται ται ο πίνακας των περιβαλλοντικών δεικτών (CO2).....                                                                                                         | 109 |
| Εικόνα 61: Προβολή του κεντρικού πίνακα ελέγχου (Dashboard) του διαχειριστή και διαδικασία επιλογής της ενότητας "User Management" από το πλευρικό μενού πλοήγησης.....                                                         | 110 |
| Εικόνα 62: Η οθόνη διαχείρισης χρηστών, προσβάσιμη μόνο από διαχειριστές. Εμφανίζονται οι ρόλοι και η κατάσταση (Active/Inactive) κάθε χρήστη.....                                                                              | 110 |
| Εικόνα 63: Απενεργοποίηση λογαριασμού χρήστη. Η κατάσταση έχει αλλάξει σε "Inactive", απαγορεύοντας την πρόσβαση στο σύστημα.....                                                                                               | 111 |
| Εικόνα 64: Επιλογή του πλήκτρου "Create User" για την οριστικοποίηση της εγγραφής και την εισαγωγή των στοιχείων του νέου μέλους.....                                                                                           | 112 |
| Εικόνα 65: Επιβεβαίωσης επιτυχούς δημιουργίας χρήστη.....                                                                                                                                                                       | 112 |
| Εικόνα 66: Τα endpoints της ενότητας Authentication στο Swagger UI.....                                                                                                                                                         | 124 |
| Εικόνα 67: Τεκμηρίωση της μεθόδου POST /auth/login στο Swagger UI για τη διαδικασία αυθεντικοποίησης και εισόδου του χρήστη στην πλατφόρμα.....                                                                                 | 125 |
| Εικόνα 68: Τεκμηρίωση της μεθόδου GET /auth/me στο Swagger UI για την ανάκτηση των βασικών πληροφοριών του τρέχοντος συνδεδεμένου χρήστη.....                                                                                   | 125 |
| Εικόνα 69: Λίστα endpoints για τη διαχείριση χρηστών (Users Controller).....                                                                                                                                                    | 127 |
| Εικόνα 70: Τεκμηρίωση της μεθόδου POST /users/create στο Swagger UI για τη δημιουργία νέου χρήστη και τον ορισμό των αντίστοιχων ρόλων και υπο-ρόλων (roles & sub-roles).....                                                   | 128 |
| Εικόνα 71: Endpoints για CRUD λειτουργίες και αναζήτηση DPPs.....                                                                                                                                                               | 129 |
| Εικόνα 72: Τεκμηρίωση της μεθόδου POST /dpp/json/ για τη χειροκίνητη καταχώρηση και δημιουργία ενός νέου DPP μέσω της υποβολής δομημένων δεδομένων σε μορφή JSON.....                                                           | 130 |
| Εικόνα 73: Τεκμηρίωση της μεθόδου PUT /dpp/json/{dpp_id}. Εμφανίζεται η διαδικασία ενημέρωσης (Update) των στοιχείων ενός ήδη καταχωρημένου DPP, χρησιμοποιώντας το μοναδικό αναγνωριστικό (dpp_id) ως παράμετρο διαδρομής..... | 131 |
| Εικόνα 74: Endpoints εκτέλεσης SPARQL ερωτημάτων.....                                                                                                                                                                           | 132 |

|                                                                                                                                                                                                              |     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Εικόνα 75: Τεκμηρίωση της μεθόδου POST /dpp/sparql/query για την εκτέλεση σημασιολογικών ερωτημάτων SELECT στη βάση Fuseki, με ενσωματωμένους περιορισμούς ασφαλείας βάσει του ρόλου του χρήστη.....         | 132 |
| Εικόνα 76: Endpoints για εξαγωγή (Export) και μεταφόρτωση (Files).....                                                                                                                                       | 133 |
| Εικόνα 77: Τεκμηρίωση της μεθόδου POST /dpp/files/upload στο Swagger UI για τη μεταφόρτωση και αυτόματη επεξεργασία αρχείων τύπου .aasx ή .zip που περιέχουν δεδομένα διαβατηρίου.....                       | 134 |
| Εικόνα 78: Endpoint ελέγχου κατάστασης και υγείας συστήματος.....                                                                                                                                            | 135 |
| Εικόνα 79: Τεκμηρίωση της μεθόδου GET /system/health στο Swagger UI για τον έλεγχο της κατάστασης λειτουργίας (Health Check) του API και της σύνδεσης με τη βάση δεδομένων.....                              | 135 |
| Εικόνα 80: Δείγμα των μετρήσεων που παράγει το endpoint /metrics.....                                                                                                                                        | 136 |
| Εικόνα 81: Συνολική διάταξη της ενότητας System Health στο Grafana, όπου απεικονίζονται οι δείκτες κατάστασης, τα Gauges πόρων και το ιστορικό χρήσης μνήμης.....                                            | 138 |
| Εικόνα 82: Λεπτομερής απεικόνιση των panels API Performance και State Timeline, όπου καταγράφεται η σταθερή απόκριση του συστήματος κάτω από 5ms και η απουσία σφαλμάτων κατά τη διάρκεια των μετρήσεων..... | 140 |
| Εικόνα 83: Προβολή των ακατέργαστων μετρήσεων (raw metrics) στη διεπαφή του Prometheus. Διακρίνεται ο μετρητής http_requests_total και ο διαχωρισμός των δεδομένων ανά endpoint και status code.....         | 141 |

## **Λίστα Πινάκων**

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| Πίνακας 1: Αντιστοίχιση κατηγοριών πληροφορίας DPP με τα στάδια του κύκλου ζωής. | 13 |
| Πίνακας 2: Συγκριτική αξιολόγηση συναφών εργασιών                                | 21 |
| Πίνακας 3: Σενάρια ελέγχου λειτουργικότητας και ασφάλειας                        | 77 |
| Πίνακας 4: Βαθμός ικανοποίησης απαιτήσεων                                        | 86 |

## Ακρωνύμια

|       |                                               |
|-------|-----------------------------------------------|
| AAS   | Asset Administration Shell                    |
| AASX  | Asset Administration Shell Package            |
| API   | Application Programming Interface             |
| CRUD  | Create, Read, Update, Delete                  |
| CSS   | Cascading Style Sheets                        |
| DPP   | Digital Product Passport                      |
| ESPR  | Ecodesign for Sustainable Products Regulation |
| HTML  | Hypertext Markup Language                     |
| HTTP  | Hypertext Transfer Protocol                   |
| JSON  | JavaScript Object Notation                    |
| JSONB | JSON Binary                                   |
| JWT   | JSON Web Token                                |
| LCA   | Life Cycle Assessment                         |
| PDF   | Portable Document Format                      |
| RBAC  | Role-Based Access Control                     |
| RDF   | Resource Description Framework                |
| REST  | Representational State Transfer               |

|        |                                        |
|--------|----------------------------------------|
| SPARQL | SPARQL Protocol and RDF Query Language |
| UI     | User Interface                         |
| URI    | Uniform Resource Identifier            |
| XML    | eXtensible Markup Language             |

## Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει τον κρίσιμο ρόλο των Ψηφιακών Διαβατηρίων Προϊόντων (Digital Product Passports - DPPs) στη μετάβαση προς το μοντέλο της κυκλικής οικονομίας, όπως αυτό προβλέπεται από τον νέο Ευρωπαϊκό Κανονισμό ESPR. Το κύριο πρόβλημα που εντοπίζεται είναι η έλλειψη ολοκληρωμένων πληροφοριακών συστημάτων που να διασφαλίζουν τη διαλειτουργικότητα και τη διαφάνεια των δεδομένων σε όλο τον κύκλο ζωής ενός προϊόντος. Οι παραδοσιακές βάσεις δεδομένων αδυνατούν να προσφέρουν την απαιτούμενη ευελιξία για τη διαχείριση ετερογενών πληροφοριών, καθιστώντας αναγκαία την υιοθέτηση τεχνολογιών Σημασιολογικού Ιστού (Semantic Web) για την αυτόματη εξαγωγή συμπερασμάτων και την καθολική ανταλλαγή δεδομένων.

Για την αντιμετώπιση αυτών των προκλήσεων, σχεδιάστηκε και υλοποιήθηκε μια πρότυπη σημασιολογική πλατφόρμα ιστού για την ολοκληρωμένη διαχείριση και αναζήτηση DPPs. Η λύση βασίζεται στο πρότυπο Asset Administration Shell (AAS) και αξιοποιεί μια υβριδική αρχιτεκτονική αποθήκευσης, συνδυάζοντας τη σχεσιακή βάση PostgreSQL (JSONB) με το σημασιολογικό αποθετήριο Apache Jena Fuseki. Το backend αναπτύχθηκε με το πλαίσιο FastAPI (Python) και το frontend με τη βιβλιοθήκη React, προσφέροντας λειτουργίες διαχείρισης κύκλου ζωής DPPs, διαδραστική οπτικοποίηση γράφου γνώσης και σύνθετη αναζήτηση DPPs μέσω σημασιολογικών ερωτημάτων SPARQL.

Η αξιολόγηση της πλατφόρμας πραγματοποιήθηκε μέσω δοκιμών μονάδας και ενοποίησης, καθώς και μέσω της χρήσης εργαλείων παρακολούθησης (Prometheus, Grafana) για την ανάλυση της απόδοσης. Τα αποτελέσματα έδειξαν ότι το σύστημα ανταποκρίνεται με υψηλή ταχύτητα καθώς και εξασφαλίζει την ακεραιότητα των δεδομένων μέσω μηχανισμών ασφαλείας RBAC. Η εργασία τεκμηριώνει ότι η χρήση σημασιολογικών προτύπων (RDF, OWL, JSON-LD) αποτελεί τη βέλτιστη προσέγγιση για τη δημιουργία ενός βιώσιμου και επεκτάσιμου οικοσυστήματος DPPs, προσφέροντας ένα έτοιμο Proof-of-Concept για βιομηχανική εφαρμογή.

**Λέξεις Κλειδιά:** Ψηφιακά Διαβατήρια Προϊόντων (DPP), Σημασιολογικός Ιστός, Πλατφόρμα Διαχείρισης, Κυκλική Οικονομία, SPARQL, Asset Administration Shell (AAS), FastAPI, Γράφος Γνώσης, Διαλειτουργικότητα.

## Abstract

This master's thesis explores the critical role of Digital Product Passports (DPPs) in the transition towards a circular economy model, as mandated by the new European Ecodesign for Sustainable Products Regulation (ESPR). The primary challenge identified is the lack of integrated information systems capable of ensuring data interoperability and transparency throughout a product's lifecycle. Traditional databases often fail to provide the necessary flexibility for managing heterogeneous information, necessitating the adoption of Semantic Web technologies to enable automated reasoning and universal data exchange.

To address these challenges, a prototype semantic web platform for the integrated management and search of DPPs was designed and implemented. The solution is based on the Asset Administration Shell (AAS) standard and utilizes a hybrid storage architecture, combining a PostgreSQL relational database (JSONB) with the Apache Jena Fuseki semantic repository. The backend was developed using the FastAPI framework (Python) and the frontend with the React library, providing comprehensive DPP lifecycle management features, interactive knowledge graph visualization, and advanced DPP search capabilities via SPARQL semantic queries.

The platform's evaluation was conducted through unit and integration testing, as well as the implementation of monitoring tools (Prometheus, Grafana) for performance analysis. The results demonstrated that the system achieves high responsiveness and ensures data integrity through Role-Based Access Control (RBAC) security mechanisms. The thesis concludes that the use of semantic standards (RDF, OWL, JSON-LD) represents the optimal approach for creating a sustainable and scalable DPP ecosystem, offering a robust Proof-of-Concept for industrial application.

**Keywords:** *Digital Product Passports (DPP), Semantic Web, Management Platform, Circular Economy, SPARQL, Asset Administration Shell (AAS), FastAPI, Knowledge Graph, Interoperability.*

# 1

## *Εισαγωγή*

### *1.1 Ψηφιακά διαβατήρια προϊόντων και σημασιολογική διαχείριση πληροφορίας*

Η σύγχρονη παγκόσμια οικονομία βρίσκεται σε μια κρίσιμη φάση μετάβασης από το παραδοσιακό γραμμικό μοντέλο παραγωγής και κατανάλωσης προς ένα βιώσιμο πρότυπο **κυκλικής οικονομίας** [9]. Στο πλαίσιο αυτό, η ανάγκη για διαφάνεια, βιωσιμότητα και ανιχνευσιμότητα καθ' όλη τη διάρκεια του κύκλου ζωής ενός προϊόντος αναδεικνύεται σε κυρίαρχη επιδίωξη της Ευρωπαϊκής Ένωσης [13] [20], καθώς και ενός ευρέος φάσματος βιομηχανικών και καταναλωτικών φορέων [31]. Η συγκέντρωση και η παροχή κρίσιμων δεδομένων που αφορούν τη χημική σύσταση, την προέλευση των υλικών, τις περιβαλλοντικές επιπτώσεις, καθώς και τις δυνατότητες επισκευής ή ανακύκλωσης, δεν αποτελεί πλέον μόνο ζήτημα κανονιστικής συμμόρφωσης, αλλά ένα θεμελιώδες εργαλείο για την επίτευξη της κλιματικής ουδετερότητας και της αποδοτικής χρήσης των πόρων.

Ως απάντηση σε αυτές τις προκλήσεις, τα **Ψηφιακά Διαβατήρια Προϊόντων (Digital Product Passports – DPPs)** εισάγονται ως ένας καινοτόμος μηχανισμός μεταφοράς πληροφορίας, ο οποίος συνοδεύει κάθε φυσικό προϊόν στην αγορά, εμπλουτισμένος με δομημένα δεδομένα (structured data). Παρά τη θεσμική προώθηση των DPPs μέσω της νέας Ευρωπαϊκής Πράσινης Συμφωνίας και της στρατηγικής για τη Βιώσιμη Ψηφιακή Οικονομία [18], παρατηρείται ένα σημαντικό τεχνολογικό κενό: η έλλειψη ευέλικτων και σύγχρονων πληροφοριακών λύσεων που να υποστηρίζουν την ολοκληρωμένη διαχείριση αυτών των δεδομένων με τη χρήση τεχνολογιών **Σημασιολογικού Ιστού (Semantic Web)**.

Η αναγκαιότητα των σημασιολογικών τεχνολογιών έναντι των παραδοσιακών σχεσιακών βάσεων δεδομένων (RDBMS) είναι στρατηγικής σημασίας. Ενώ οι σχεσιακές βάσεις απαιτούν στατικά και δύσκαμπτα σχήματα (schemas) που δυσκολεύουν την ενοποίηση πληροφοριών από διαφορετικούς κλάδους, ο σημασιολογικός ιστός προάγει την καθολική διαλειτουργικότητα μέσω

κοινών οντολογιών. Επιπλέον, η χρήση προτύπων όπως το RDF και η OWL επιτρέπει την αυτοματοποιημένη εξαγωγή συμπερασμάτων (automated reasoning), δίνοντας στο σύστημα τη δυνατότητα να επαληθεύει αυτόματα τη συμμόρφωση ενός προϊόντος με περιβαλλοντικούς κανονισμούς, μια λειτουργία που δεν υποστηρίζεται εγγενώς από τις σχεσιακές προσεγγίσεις [31] [35].

Υπό αυτό το πρίσμα, οι υπάρχουσες προσεγγίσεις χαρακτηρίζονται ως σποραδικές, καθώς παραμένουν εγκλωβισμένες σε αποσπασματικές υλοποιήσεις και στενά κλαδικά πλαίσια (data silos), αποτυγχάνοντας να προσφέρουν μια οριζόντια και ενοποιημένη διαχείριση της πληροφορίας σε όλο το εύρος της οικονομίας [5]. Επιπλέον, οι λύσεις αυτές είτε δεν αξιοποιούν στο έπακρο σύγχρονες αρχιτεκτονικές λογισμικού, είτε δεν δίνουν τη δέουσα έμφαση στην αναζήτηση και επερώτηση σημασιολογικής πληροφορίας με ευχρηστία και ασφάλεια.

## 1.2 Αντικείμενο διπλωματικής

Αντικείμενο της παρούσας διπλωματικής είναι η ανάπτυξη μιας σύγχρονης και αποδοτικής εφαρμογής ιστού, η οποία θα επιτρέπει τη **διαχείριση DPPs** από τους δημιουργούς/ιδιοκτήτες τους και θα προσφέρει προηγμένες δυνατότητες **αναζήτησης και ανάκτησης πληροφορίας** για εγγεγραμμένους χρήστες. Στους χρήστες αυτούς περιλαμβάνονται και οι **δυνητικοί πελάτες** των προϊόντων, οι οποίοι μέσω των DPPs μπορούν να αποκτήσουν πρόσβαση σε κρίσιμα δεδομένα για τη λήψη αποφάσεων αγοράς. Ειδικότερα, η εργασία εστιάζει στην υλοποίηση ενός συστήματος που υποστηρίζει:

- Την ασφαλή, βασισμένη σε διακριτικά (token-based), αυθεντικοποίηση και εξουσιοδότηση χρηστών μέσω του προτύπου JSON Web Token (JWT), εξασφαλίζοντας την προστασία των δεδομένων ανάλογα με τους ρόλους (RBAC).
- Τη δημιουργία, δημοσίευση, αποδημοσίευση, τροποποίηση και διαγραφή DPPs
- Τη δομημένη περιγραφή των προϊόντων με χρήση διεθνών σημασιολογικών προτύπων, όπως τα RDF, OWL και JSON-LD.
- Την πολυεπίπεδη αναζήτηση DPPs μέσω τριών διακριτών μεθόδων: (α) απλή αναζήτηση με λέξεις-κλειδιά, (β) εξεζητημένη αναζήτηση βάσει κριτηρίων κλειδιού-τιμής (key-value) και (γ) σύνθετη σημασιολογική επερώτηση μέσω **SPARQL** πάνω σε βάση δεδομένων / γνώσης RDF.

Η τεχνολογική αρχιτεκτονική του συστήματος βασίζεται στο πλαίσιο (framework) **FastAPI**<sup>1</sup> (Python) για το backend, στην σχεσιακή βάση δεδομένων **PostgreSQL**<sup>2</sup> για την αποθήκευση δομημένων δεδομένων και στη βιβλιοθήκη **React**<sup>3</sup> για τη διεπαφή χρήστη, ενώ το σημασιολογικό επίπεδο (διαχείρισης και αναζήτησης δομημένων δεδομένων) υποστηρίζεται από τον διακομιστή (server) **Apache Jena Fuseki**<sup>4</sup>.

---

<sup>1</sup> FastAPI: <https://fastapi.tiangolo.com/>

<sup>2</sup> PostgreSQL: <https://www.postgresql.org/>

<sup>3</sup> React: <https://react.dev/>

<sup>4</sup> Apache Jena Fuseki: <https://jena.apache.org/documentation/fuseki2/>

Η εργασία λειτουργεί ως proof-of-concept, ωστόσο υλοποιεί πλήρως όλες τις λειτουργικές απαιτήσεις ενός συστήματος διαχείρισης και αναζήτησης DPP, ενσωματώνοντας τις αρχές της RESTful αρχιτεκτονικής, της διακριτής οργάνωσης frontend/backend και της ασφάλειας.

### **1.3 Συνεισφορά και πλεονεκτήματα**

Η παρούσα διπλωματική εργασία συνιστά μια ολοκληρωμένη προσπάθεια συνδυασμού τεχνολογιών Σημασιολογικού Ιστού με μοντέρνα εργαλεία ανάπτυξης εφαρμογών ιστού, αποσκοπώντας στη δημιουργία μιας λειτουργικής πλατφόρμας διαχείρισης Ψηφιακών Διαβατηρίων Προϊόντων (DPPs). Η βασική τεχνική συνεισφορά της εργασίας εστιάζει στην υλοποίηση ενός RESTful συστήματος με διακριτή αρχιτεκτονική frontend και backend, υποστήριξη πολλαπλών ρόλων χρηστών, ασφαλή αυθεντικοποίηση με JWT, και σημασιολογική αναπαράσταση των DPPs μέσω RDF/OWL/JSON-LD, δηλαδή με πλήρη υιοθέτηση σημασιολογικών προτύπων για τη διαχείριση DPPs, κάτι που σπανίζει στην υφιστάμενη τεχνολογική βιβλιογραφία και στις τρέχουσες βιομηχανικές υλοποιήσεις Ψηφιακών Διαβατηρίων Προϊόντων, οι οποίες συχνά περιορίζονται σε απλά σχήματα δεδομένων χωρίς δυνατότητα σημασιολογικής διασύνδεσης [31].

Παράλληλα, ιδιαίτερη έμφαση δίνεται στην αναζήτηση της πληροφορίας, η οποία υποστηρίζεται με πολλαπλές εναλλακτικές προσεγγίσεις — από απλές λέξεις-κλειδιά, μέχρι εξεζητημένα key-value κριτήρια και SPARQL ερωτήματα σε σημασιολογική βάση γνώσης. Πέρα όμως από τις συγκεκριμένες λειτουργικές συνιστώσες, η εργασία προσφέρει και γενικότερα πλεονεκτήματα: ενισχύει την κατανόηση της σημασίας των ανοιχτών προτύπων στην αναπαράσταση γνώσης· αναδεικνύει τον ρόλο των σημασιολογικών τεχνολογιών στην υποστήριξη της διαλειτουργικότητας μεταξύ συστημάτων· προάγει την αρχιτεκτονική σχεδίαση με βάση τον διαχωρισμό ανησυχιών (separation of concerns) και τη δυνατότητα επεκτασιμότητας· και τέλος, μπορεί να λειτουργήσει ως διδακτικό ή ερευνητικό πρότυπο για την ανάπτυξη παρόμοιων εφαρμογών με στόχο τη διαφάνεια και την αξιοποίηση δεδομένων σε ψηφιακά οικοσυστήματα προϊόντων. Επιπλέον, η ενασχόληση με τεχνολογίες αιχμής επέτρεψε την απόκτηση σημαντικής τεχνικής εμπειρίας και εμβάθυνση στον πλήρη κύκλο ανάπτυξης λογισμικού (full-stack), καθιστώντας την εργασία αυτή όχι μόνο τεχνολογικά επίκαιρη αλλά και παιδαγωγικά ουσιαστική.

### **1.4 Δομή της διπλωματικής**

Η παρούσα εργασία διαρθρώνεται σε έξι κεφάλαια. Στο Κεφάλαιο 2 παρουσιάζεται το απαραίτητο υπόβαθρο αναφορικά με την έννοια των DPPs, το νομοθετικό πλαίσιο και τις χρησιμοποιούμενες τεχνολογίες. Το Κεφάλαιο 3 εξετάζει συναφείς ερευνητικές ή τεχνολογικές εργασίες και προχωρά σε σύγκριση με τη δική μας προσέγγιση βάσει επιλεγμένων κριτηρίων. Στο Κεφάλαιο 4 περιγράφεται η πλήρης διαδικασία ανάπτυξης του συστήματος σύμφωνα με το μοντέλο του καταρράκτη· η ανάλυση αυτή περιλαμβάνει τον καθορισμό των απαιτήσεων, τον αρχιτεκτονικό σχεδιασμό, την τεχνική υλοποίηση με τη δομή του κώδικα, καθώς και τη διαδικασία επαλήθευσης μέσω σεναρίων δοκιμών. Το Κεφάλαιο 5 παρέχει οδηγίες εγκατάστασης και επίδειξης της εφαρμογής. Τέλος, το Κεφάλαιο 6 συνοψίζει τα συμπεράσματα και προτείνει κατευθύνσεις για μελλοντική επέκταση του συστήματος.

# 2

## Υπόβαθρο και τεχνολογίες

### 2.1 Τι είναι τα ψηφιακά διαβατήρια προϊόντων (DPPs)

Τα **Ψηφιακά Διαβατήρια Προϊόντων** (Digital Product Passports – DPPs) ορίζονται ως ολοκληρωμένες ψηφιακές εγγραφές, οι οποίες συνοδεύουν ένα φυσικό προϊόν καθ' όλη τη διάρκεια του κύκλου ζωής του, ενσωματώνοντας κρίσιμα δεδομένα που αφορούν την παραγωγή, τη χρήση, τη συντήρηση, καθώς και την τελική του διάθεση ή ανακύκλωση. Η εννοιολογική θεμελίωση των DPPs εδράζεται στο ευρύτερο στρατηγικό πλαίσιο για την μετάβαση στην κυκλική οικονομία και την προώθηση βιώσιμων προτύπων παραγωγής και κατανάλωσης. Βασικός στόχος αυτής της τεχνολογικής προσέγγισης είναι η διασφάλιση της πρόσβασης σε πλήρως δομημένη, τυποποιημένη και επαληθεύσιμη πληροφόρηση για το σύνολο των εμπλεκόμενων μερών, συμπεριλαμβανομένων των κατασκευαστών, των καταναλωτών, των φορέων ανακύκλωσης και των ρυθμιστικών αρχών [10] [19].

Η πληροφορία που εμπεριέχεται σε ένα DPP είναι πολυδιάστατη και μπορεί να καλύπτει ένα ευρύ φάσμα δεδομένων, όπως η ακριβής υλική και χημική σύνθεση του προϊόντος, οι δείκτες ενεργειακής απόδοσης, η γεωγραφική προέλευση των πρώτων υλών, καθώς και αναλυτικές οδηγίες για τη συντήρηση και την επισκευή του. Επιπλέον, περιλαμβάνονται δεδομένα σχετικά με την περιβαλλοντική επίδραση, πιστοποιήσεις ποιότητας και πληροφορίες κανονιστικής συμμόρφωσης που είναι απαραίτητες για την κυκλοφορία του προϊόντος στην αγορά [35]. Η διασύνδεση του φυσικού αντικειμένου με την ψηφιακή του πληροφορία επιτυγχάνεται συνήθως μέσω μοναδικών αναγνωριστικών ή τεχνολογιών σήμανσης, όπως οι κωδικοί QR, οι οποίοι επιτρέπουν την άμεση και απρόσκοπτη πρόσβαση στα δεδομένα μέσω εξειδικευμένων διαδικτυακών πλατφορμών [6].

Η ανάγκη για DPPs προκύπτει από τη σταδιακή μετάβαση της οικονομίας προς πιο υπεύθυνα, διαφανή και διαλειτουργικά μοντέλα παραγωγής, τα οποία απαιτούν: (α) τυποποιημένα μεταδεδομένα, (β) ανοιχτά πρότυπα αναπαράστασης γνώσης, (γ) μηχανισμούς ασφαλούς επεξεργασίας και διάχυσης πληροφορίας και (δ) σημασιολογική ευφυΐα για να υποστηρίζεται η λήψη αποφάσεων και η σύνδεση μεταξύ δεδομένων. Εντός αυτού του πλαισίου, το DPP λειτουργεί ως ένας ψηφιακός ενισχυτής πληροφορίας, προάγοντας την ανιχνευσιμότητα, την ανακυκλωσιμότητα, την πρόσβαση στην τεχνική τεκμηρίωση, και τελικά, την ενίσχυση της βιωσιμότητας στον κύκλο ζωής του προϊόντος [8].

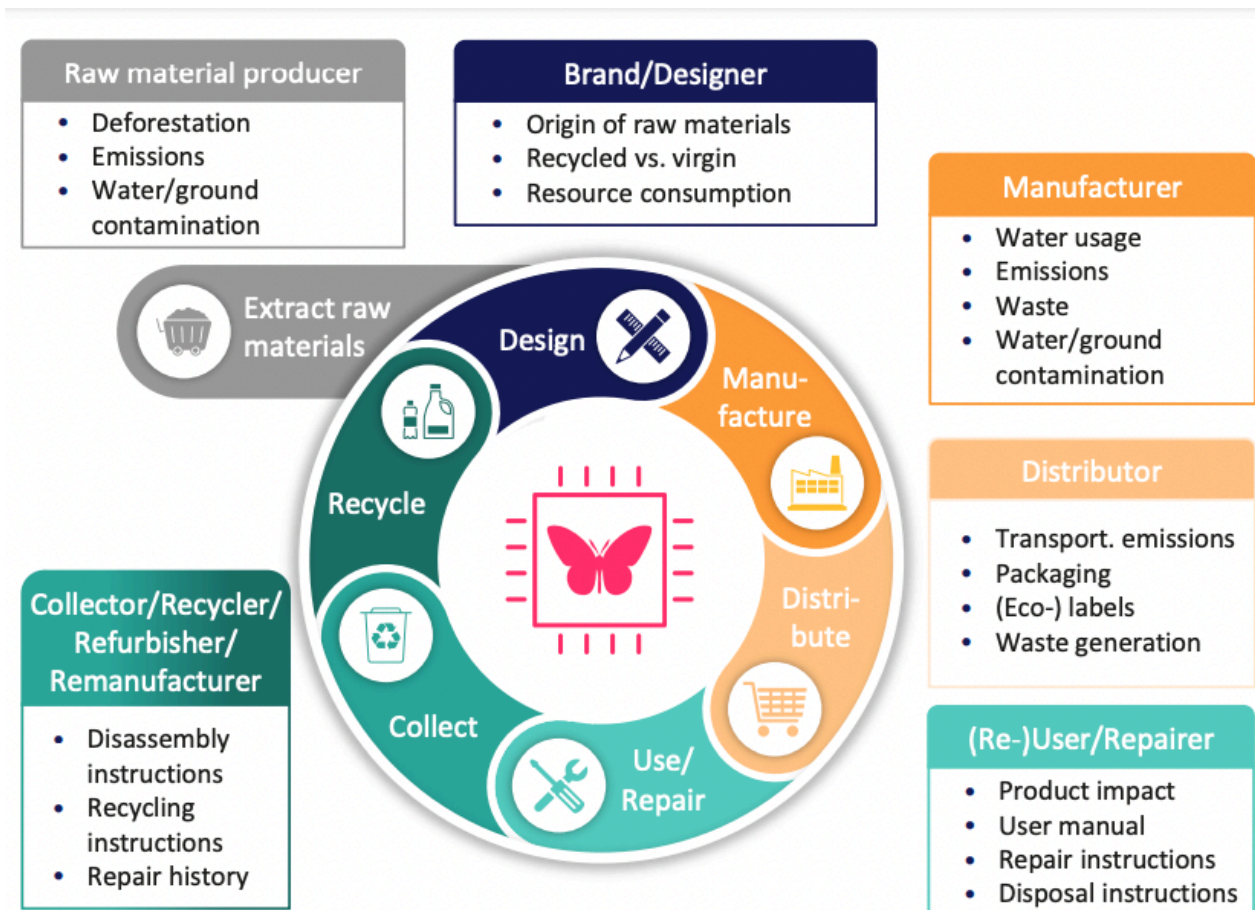
## 2.2 Νομοθετικό πλαίσιο και ευρωπαϊκές πρωτοβουλίες για τα DPPs

Η έννοια των Ψηφιακών Διαβατηρίων Προϊόντων (DPPs) αποτελεί πλέον το βασικό θεμέλιο της στρατηγικής της Ευρωπαϊκής Ένωσης για τη βιώσιμη ανάπτυξη, την ενίσχυση της κυκλικής οικονομίας και την επίτευξη των στόχων της κλιματικής ουδετερότητας. Συγκεκριμένα, τα DPPs εισήχθησαν επίσημα ως πρόταση πολιτικής στο πλαίσιο του **Νέου Σχεδίου Δράσης για την Κυκλική Οικονομία (Circular Economy Action Plan – CEAP)**, το οποίο παρουσιάστηκε το 2020 από την Ευρωπαϊκή Επιτροπή. Το στρατηγικό αυτό έγγραφο προβλέπει τη δημιουργία μηχανισμών πληροφόρησης για τα προϊόντα που διακινούνται στην ευρωπαϊκή αγορά, με κύριο στόχο τη βελτίωση της διαφάνειας και την εξασφάλιση της πρόσβασης στην πληροφορία για τους τελικούς χρήστες, τους καταναλωτές και όλους τους εμπλεκόμενους φορείς της εφοδιαστικής αλυσίδας [9] [15].

Η κεντρική θέση του διαβατηρίου στον κύκλο ζωής ενός προϊόντος αναδεικνύεται στην Εικόνα 1, όπου το DPP λειτουργεί ως ο συνδετικός κρίκος μεταξύ όλων των σταδίων παραγωγής και κατανάλωσης.

Όπως παρατηρείται στο διάγραμμα, το διαβατήριο επιτρέπει τη συγκέντρωση και διαμοίραση κρίσιμων δεδομένων ανά στάδιο:

- **Πρώτες ύλες και Σχεδιασμός:** Καταγράφονται στοιχεία για την αποδάσωση, τις εκπομπές ρύπων και τη χρήση ανακυκλωμένων υλικών έναντι πρωτογενών.
- **Παραγωγή και Διανομή:** Παρέχεται διαφάνεια για την κατανάλωση νερού, τη διαχείριση αποβλήτων και το περιβαλλοντικό αποτύπωμα της μεταφοράς και της συσκευασίας.
- **Χρήση και Τέλος Κύκλου Ζωής:** Ο χρήστης αποκτά πρόσβαση σε εγχειρίδια και οδηγίες επισκευής, ενώ οι φορείς ανακύκλωσης λαμβάνουν πολύτιμες οδηγίες αποσυναρμολόγησης, διευκολύνοντας την επιστροφή των υλικών στην παραγωγή ως δευτερογενείς πρώτες ύλες.



Εικόνα 1: Ο κύκλος ζωής του προϊόντος και ο ρόλος του DPP στην κυκλική οικονομία. Πηγή: Finboot (2022) [22].

Στην ίδια κατεύθυνση κινείται και ο **Κανονισμός για τον Οικολογικό Σχεδιασμό Βιώσιμων Προϊόντων (Ecodesign for Sustainable Products Regulation – ESPR)**, ο οποίος μετά την αρχική πρόταση του 2022, θεσπίστηκε επίσημα το 2024 (Κανονισμός ΕΕ 2024/1781). Ο ESPR αποτελεί το κύριο νομοθετικό εργαλείο για την υποχρεωτική ενσωμάτωση των DPPs σε ένα ευρύ φάσμα προϊόντων εντός της ενιαίας αγοράς, ξεκινώντας από κρίσιμους τομείς, όπως οι μπαταρίες, τα κλωστοϋφαντουργικά και τα ηλεκτρονικά. Ο εν λόγω κανονισμός αποσκοπεί στην ενίσχυση της ανιχνευσιμότητας, τη διευκόλυνση των διαδικασιών ανακύκλωσης και την παροχή λεπτομερών πληροφοριών για τη συντήρηση και την επισκευή, υποστηρίζοντας επιχειρηματικά μοντέλα που βασίζονται στην επαναχρησιμοποίηση και τη δραστική μείωση των αποβλήτων [13] [16].

Η ανάπτυξη των DPPs βρίσκεται επίσης σε πλήρη ευθυγράμμιση με ευρύτερες στρατηγικές πολιτικές, όπως η **Ευρωπαϊκή Στρατηγική για τα Δεδομένα (European Data Strategy)**<sup>5</sup>, η **Ψηφιακή Πράσινη Συμφωνία (Green Deal)**<sup>6</sup> και η δημιουργία του **Κοινού Ευρωπαϊκού**

<sup>5</sup> Η Ευρωπαϊκή Στρατηγική για τα Δεδομένα αποσκοπεί στη δημιουργία μιας ενιαίας αγοράς δεδομένων που θα διασφαλίζει την παγκόσμια ανταγωνιστικότητα της Ευρώπης, επιτρέποντας την ασφαλή ροή δεδομένων εντός της ΕΕ προς όφελος των επιχειρήσεων και των πολιτών.

<sup>6</sup> Η Ευρωπαϊκή Πράσινη Συμφωνία αποτελεί τη στρατηγική της ΕΕ για τη μετάβαση σε μια κλιματικά ουδέτερη και κυκλική οικονομία έως το 2050, προωθώντας τη βιώσιμη χρήση των πόρων και τη μείωση της ρύπανσης.

**Χώρου Δεδομένων Προϊόντων (Product Data Space)**<sup>7</sup>. Αυτές οι πρωτοβουλίες στοχεύουν στη δημιουργία ενιαίων υποδομών για την ασφαλή ανταλλαγή και επεξεργασία δεδομένων μεταξύ επιχειρήσεων και οργανισμών. Παράλληλα, πειραματικές δράσεις, όπως το **Digital Product Passport Pathfinder (CIRPASS)**, το κίνημα **Bauhaus** και το **Sustainable Product Initiative (SPI)** λειτουργούν ως πεδία αξιολόγησης για την εφαρμογή των DPPs σε πραγματικά βιομηχανικά περιβάλλοντα, προετοιμάζοντας το έδαφος για τις επερχόμενες κατ' εξουσιοδότηση πράξεις (delegated acts) που θα εξειδικεύουν τις απαιτήσεις ανά κατηγορία προϊόντος [4] [6].

Κομβικής σημασίας για την εξέλιξη αυτή αποτελεί ο **Κανονισμός για τις Μπαταρίες (Battery Regulation – Regulation (EU) 2023/1542)**, ο οποίος συνιστά το πρώτο νομοθετικό κείμενο που παρέχει μια πλήρη και ώριμη περιγραφή των απαιτούμενων πληροφοριών ενός DPP για έναν συγκεκριμένο τομέα. Ο εν λόγω κανονισμός ορίζει λεπτομερώς τα δεδομένα που πρέπει να περιλαμβάνονται στο «Διαβατήριο Μπαταρίας» (Battery Passport), όπως η χημική σύσταση, η χωρητικότητα, η διάρκεια ζωής και η κατάσταση υγείας (state of health) της μπαταρίας, λειτουργώντας ως πρότυπο για την επέκταση των DPPs και σε άλλες κατηγορίες προϊόντων στο μέλλον [36].

## 2.3 Πρότυπα και τεχνολογίες αναπαράστασης DPPs

Για την επιτυχή εφαρμογή αυτού του νομοθετικού πλαισίου, η Ευρωπαϊκή Επιτροπή έχει δώσει ιδιαίτερη έμφαση στην τυποποίηση, αναθέτοντας σε οργανισμούς όπως η **CEN** και η **CENELEC** την ανάπτυξη εναρμονισμένων προτύπων που θα διασφαλίζουν τη διαλειτουργικότητα των συστημάτων DPP [14]. Η αξιόπιστη, διαλειτουργική και επεκτάσιμη αναπαράσταση ενός Ψηφιακού Διαβατηρίου απαιτεί τη χρήση σημασιολογικών τεχνολογιών και διεθνών προτύπων που επιτρέπουν την ακριβή μοντελοποίηση της πληροφορίας.

Στο πλαίσιο αυτό, καθίσταται απαραίτητη η υιοθέτηση προτύπων όπως:

- **GS1 Standards (GTIN & Digital Link):** Το πρότυπο GTIN (Global Trade Item Number) αποτελεί το κυρίαρχο παγκόσμιο πρότυπο για την ταυτοποίηση εμπορικών ειδών, ενώ το GS1 Digital Link επιτρέπει τη διασύνδεση των φυσικών προϊόντων με τις ψηφιακές τους οντότητες (όπως το DPP) μέσω τυποποιημένων δομών URL, καθιστώντας την πληροφορία προσβάσιμη μέσω κοινών σαρωτών (scanners) και smartphone [23].
- **ISO 8000:** Διασφαλίζει την ποιότητα των δεδομένων (Data Quality), καθορίζοντας αυστηρές αρχές για την ορθότητα, την πληρότητα και τη συνέπεια της πληροφορίας που εγγράφεται στο διαβατήριο [37].
- **ISO/IEC 27001:** Παρέχει το πλαίσιο για την ασφάλεια πληροφοριών (Information Security), διασφαλίζοντας την εμπιστευτικότητα και την ακεραιότητα των ευαίσθητων δεδομένων της εφοδιαστικής αλυσίδας [38].
- **ISO/IEC 20248:** Αφορά τις **ψηφιακές υπογραφές** για οπτικούς φορείς δεδομένων (όπως οι κωδικοί QR), εξασφαλίζοντας την αυθεντικότητα και την επαληθευσσιμότητα της πηγής της πληροφορίας [39].
- **ISO/IEC 15459:** Για τη μοναδική ταυτοποίηση των αντικειμένων, εξασφαλίζοντας την παγκόσμια αναγνώριση των προϊόντων [26].

<sup>7</sup> Ο Κοινός Ευρωπαϊκός Χώρος Δεδομένων Προϊόντων στοχεύει στη δημιουργία ενός ασφαλούς ψηφιακού οικοσυστήματος για την ανταλλαγή πληροφοριών σχετικά με τα χαρακτηριστικά και τον κύκλο ζωής των προϊόντων, ενισχύοντας τη διαφάνεια στην εφοδιαστική αλυσίδα.

- **ISO/IEC 19987:2021 (EPCIS):** Για το διαμοιρασμό δεδομένων σχετικά με την κίνηση και την κατάσταση των προϊόντων στην εφοδιαστική αλυσίδα [27].

Η χρήση του **Σημασιολογικού Ιστού (Semantic Web)** προσφέρει ένα ισχυρό σύνολο τεχνολογιών που ενδείκνυνται για την περιγραφή DPPs με ακρίβεια και δυνατότητα ερμηνείας από μηχανές:

1. **RDF (Resource Description Framework):** Αποτελεί το βασικό πρότυπο αναπαράστασης πληροφορίας σε μορφή τριάδων, καθιστώντας δυνατή τη μοντελοποίηση πολύπλοκων οντοτήτων όπως προϊόντα, ιδιότητες και κατασκευαστές [35].
2. **RDFS & OWL:** Αποτελούν τα βασικά εργαλεία για τη μοντελοποίηση του σχήματος (schema modeling) μιας οντολογίας, επιτρέποντας την κάλυψη ενός συγκεκριμένου τομέα ή ενός προτύπου με τυποποιημένο τρόπο. Μέσω του ορισμού ιεραρχιών κλάσεων και προηγμένων οντολογικών περιγραφών, διασφαλίζουν τη σημασιολογική συνέπεια των δεδομένων και την ορθή αναπαράσταση των σύνθετων σχέσεων που διέπουν τα DPPs [35].
3. **JSON-LD (JSON for Linking Data):** Παρέχει έναν μηχανισμό αναπαράστασης RDF σε μορφή JSON, διασφαλίζοντας τη συμμόρφωση με τις αρχές των Συνδεδεμένων Δεδομένων (Linked Data)<sup>8</sup> και την προσβασιμότητα από εφαρμογές ιστού (web applications) [35].
4. **SWRL (Semantic Web Rule Language):** Η χρήση κανόνων SWRL ενισχύει τη λειτουργικότητα του DPP, επιτρέποντας την αυτοματοποιημένη επαλήθευση συμμόρφωσης (compliance checking). Για παράδειγμα, μπορούν να οριστούν κανόνες που ελέγχουν αυτόματα αν μια μπαταρία πληροί τις προϋποθέσεις ανακύκλωσης βάσει της χημικής της σύστασης ή αν οι εκπομπές ρύπων κατά την παραγωγή υπερβαίνουν τα επιτρεπτά όρια που θέτει ο κανονισμός ESPR [41].

Ιδιαίτερη βαρύτητα για τη βιομηχανική εφαρμογή των παραπάνω τεχνολογιών έχει το πρότυπο ECLASS (πρώην eCl@ss), το οποίο αποτελεί το κυρίαρχο σύστημα ταξινόμησης προϊόντων παγκοσμίως. Το ECLASS αξιοποιείται κυρίως από κατασκευαστές και προμηθευτές για τη μοντελοποίηση του σχήματος των δεδομένων τους, καθώς προσφέρει μια εκτενή βάση τυποποιημένων ιδιοτήτων. Με αυτόν τον τρόπο, οι επιχειρήσεις περιγράφουν τα τεχνικά χαρακτηριστικά των προϊόντων τους με κοινή γλώσσα, διευκολύνοντας τις ψηφιακές συναλλαγές εντός της εφοδιαστικής αλυσίδας [42].

Παράλληλα, το IEC CDD (Common Data Dictionary) λειτουργεί ως ένα διεθνές λεξικό κοινών δεδομένων (IEC 61360). Το λεξικό αυτό αποτελεί βασικό εργαλείο για μηχανικούς και σχεδιαστές συστημάτων, παρέχοντας σαφείς ορισμούς για τα τεχνικά χαρακτηριστικά ηλεκτροτεχνικών προϊόντων. Η χρήση του IEC CDD επιτρέπει σε ρυθμιστικές αρχές και φορείς ελέγχου να αξιολογούν με ακρίβεια τη συμμόρφωση των προϊόντων με τα διεθνή πρότυπα ασφάλειας και απόδοσης [43].

Σε επίπεδο οντολογιών, αξιοποιούνται υφιστάμενα λεξιλόγια όπως το [schema.org/Product](http://schema.org/Product), το GS1 Digital Link, το ECLASS (aka. eCl@ss) και το IEC CDD [5] [23]. Τα λεξιλόγια αυτά είναι ουσιαστικής σημασίας, καθώς επιτρέπουν τη μοντελοποίηση του σχήματος μιας οντολογίας με τυποποιημένο τρόπο, καλύπτοντας έτσι πλήρως τις ανάγκες ενός συγκεκριμένου βιομηχανικού

---

<sup>8</sup> Η αρχή των Συνδεδεμένων Δεδομένων (Linked Data) αναφέρεται σε ένα σύνολο βέλτιστων πρακτικών για τη δημοσίευση και διασύνδεση δομημένων δεδομένων στον Παγκόσμιο Ιστό. Βασίζεται στη χρήση μοναδικών αναγνωριστικών (URIs) για τον προσδιορισμό αντικειμένων, τη χρήση του πρωτοκόλλου HTTP για την ανάκτηση πληροφοριών και τη χρήση προτύπων όπως το RDF και η SPARQL, ώστε τα δεδομένα να είναι αναγνώσιμα από μηχανές και να συνδέονται με άλλες εξωτερικές πηγές γνώσης.

τομέα ή ενός διεθνούς προτύπου [31]. Τα εν λόγω πρότυπα χρησιμοποιούνται από πλατφόρμες ηλεκτρονικού εμπορίου (schema.org) για τη βελτίωση της αναζήτησης, καθώς και από διανομείς και καταναλωτές (GS1 Digital Link) για τη διασύνδεση του φυσικού προϊόντος με την ψηφιακή του πληροφορία μέσω QR codes.

Πιο εξειδικευμένες πρωτοβουλίες, όπως η Onto-DPP [31] [45] και η ProductOntology [46], προτείνουν συγκεκριμένα μοντέλα για τη σύνδεση των DPPs με τεχνικά τεχνουργήματα (technical artifacts). Αυτή η συντονισμένη τεχνολογική ώθηση βασίζεται στις αρχές των Συνδεδεμένων Δεδομένων (Linked Data), οι οποίες επιτρέπουν την ανάπτυξη καινοτόμων λύσεων που ικανοποιούν τόσο τις επιχειρησιακές απαιτήσεις για επεκτασιμότητα όσο και τους αυστηρούς κανονιστικούς περιορισμούς των ευρωπαϊκών πολιτικών.

## 2.4 Υλοποιήσεις και πρωτοβουλίες σχετικές με τα DPPs

Αν και το πλαίσιο εφαρμογής των Ψηφιακών Διαβατηρίων Προϊόντων (DPPs) βρίσκεται ακόμη υπό διαμόρφωση σε ευρωπαϊκό και διεθνές επίπεδο, ήδη έχουν αναπτυχθεί ή βρίσκονται σε εξέλιξη αρκετές πιλοτικές υλοποιήσεις και πρωτοβουλίες που αποσκοπούν στη διερεύνηση της σκοπιμότητας, της τεχνολογικής υλοποίησης και της πρακτικής αξίας των DPPs σε επιλεγμένους βιομηχανικούς τομείς.

Ένα από τα πιο χαρακτηριστικά παραδείγματα είναι το έργο **CIRPASS (Collaborative Initiative for a Standards-based Digital Product Passport)**, το οποίο χρηματοδοτείται από την Ευρωπαϊκή Ένωση και αποσκοπεί στην ανάπτυξη των βασικών απαιτήσεων και προδιαγραφών για την υλοποίηση DPPs σε τρεις βασικούς τομείς: ηλεκτρονικά, μπαταρίες και κλωστοϋφαντουργικά προϊόντα. Το CIRPASS λειτουργεί ως **προπαρασκευαστική πρωτοβουλία** για τη θέσπιση κοινών τεχνολογικών υποδομών, δεδομένων και κανόνων διαλειτουργικότητας, και αποτελεί σημαντικό παράδειγμα συνεργασίας μεταξύ δημόσιου και ιδιωτικού τομέα [6].

Άλλη αξιολογημένη υλοποίηση εντοπίζεται στο έργο **Battery Pass**, μια γερμανική εθνική πρωτοβουλία που στοχεύει στη δημιουργία DPPs για ηλεκτρικές μπαταρίες. Το έργο εξετάζει πρακτικά σενάρια χρήσης, τεχνολογικές λύσεις για την αναπαράσταση των δεδομένων και τρόπους ελέγχου της ποιότητας και της αξιοπιστίας των πληροφοριών [1]. Στο ίδιο πλαίσιο, το οικοσύστημα **Catena-X** αναδεικνύεται σε μια από τις πιο σημαντικές πρωτοβουλίες, αναπτύσσοντας ένα αποκεντρωμένο δίκτυο για την ασφαλή ανταλλαγή δεδομένων DPP στην αυτοκινητοβιομηχανία και τη διασφάλιση της διαφάνειας σε όλη την εφοδιαστική αλυσίδα [3]. Παρόμοιες πρωτοβουλίες έχουν ξεκινήσει και στον τομέα της μόδας και της κλωστοϋφαντουργίας, με χαρακτηριστικά παραδείγματα όπως το **Textile Genesis**, που βασίζεται σε blockchain και DPPs για την ιχνηλασιμότητα της εφοδιαστικής αλυσίδας [44].

Επιπρόσθετα, σε επίπεδο τεχνολογικών παρόχων, έχουν γίνει προσεγγίσεις με χρήση δεδομένων συνδεδεμένων RDF (Linked Data) και χρήση σημασιολογικών λεξιλογίων. Πλατφόρμες όπως η GS1 Digital Link [23], το iPoint DPP Framework [28], καθώς και οντολογίες όπως η Circular Twin [45] και η Open Product Data Ontology [46] προσφέρουν τεχνολογική βάση για την υλοποίηση λειτουργικών και σημασιολογικά εμπλουτισμένων DPPs.

Παρότι οι παραπάνω δράσεις συνεισφέρουν ουσιαστικά στην προτυποποίηση και τη διάδοση της έννοιας του DPP, παρατηρείται συχνά είτε έλλειψη ανοικτών λογισμικών υλοποιήσεων, είτε έλλειψη ενιαίας προσέγγισης ως προς την αρχιτεκτονική των συστημάτων, τη μοντελοποίηση των

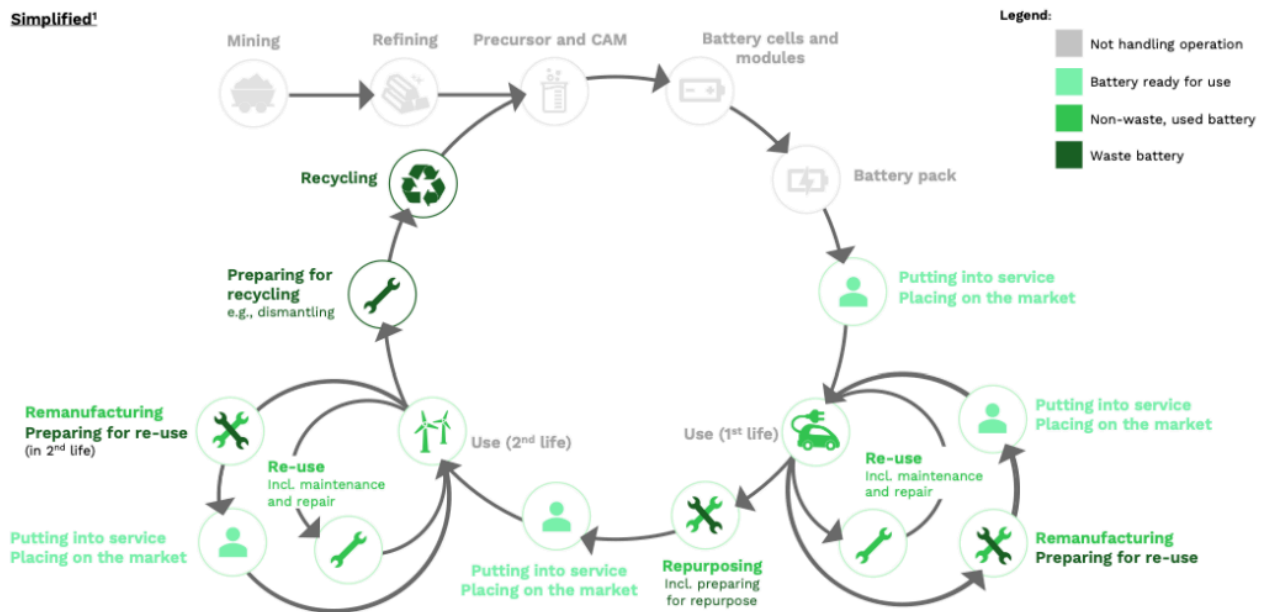
δεδομένων ή τους τρόπους επερώτησης και αναζήτησης. Αυτό καθιστά αναγκαία την ύπαρξη πρότυπων, επεκτάσιμων και ανοιχτών εφαρμογών που θα υποστηρίζουν την πλήρη διαχείριση DPPs με χρήση σημασιολογικών τεχνολογιών, διασφαλίζοντας την επαναχρησιμοποίηση, τη διαφάνεια και τη διαλειτουργικότητα.

## **2.5 Πληροφοριακά πεδία των DPPs & βασικές κατηγορίες πληροφορίας που καλύπτουν**

Η χρησιμότητα των Ψηφιακών Διαβατηρίων Προϊόντων (DPPs) καθορίζεται σε μεγάλο βαθμό από την πληρότητα και την ποιότητα των πληροφοριών που περιλαμβάνουν. Για να πετύχουν τον στόχο της διαλειτουργικής, διαφανούς και σημασιολογικά εμπλουτισμένης πληροφόρησης, τα DPPs θα πρέπει να περιλαμβάνουν ένα σύνολο δομημένων και τυποποιημένων πληροφοριακών πεδίων, τα οποία να καλύπτουν το σύνολο του κύκλου ζωής του προϊόντος, από την παραγωγή μέχρι και την τελική του διάθεση ή επαναχρησιμοποίηση [10].

Η Ευρωπαϊκή Επιτροπή, μέσω της πρότασης κανονισμού για τον οικολογικό σχεδιασμό βιώσιμων προϊόντων (ESPR), ορίζει συγκεκριμένες κατηγορίες πληροφοριών που θεωρούνται απαραίτητες για την καταγραφή σε ένα DPP. Οι κατηγορίες αυτές δεν είναι τυχαίες, αλλά βασίζονται σε ανάλυση των πληροφοριακών αναγκών των εμπλεκόμενων φορέων: κατασκευαστών, ρυθμιστικών αρχών, φορέων ανακύκλωσης, τεχνικών υποστήριξης, καταναλωτών και μεταπωλητών [19]. Η συγκέντρωση αυτής της πληροφορίας δεν αποσκοπεί μόνο στη ρύθμιση της αγοράς, αλλά λειτουργεί ως καταλύτης για τη μετάβαση προς πιο υπεύθυνα και κυκλικά παραγωγικά μοντέλα [8].

Η επιτυχής υλοποίηση ενός DPP προϋποθέτει μια τυποποιημένη πληροφοριακή δομή που να ανταποκρίνεται στις απαιτήσεις της κυκλικής οικονομίας. Η Εικόνα 2 απεικονίζει το εννοιολογικό πλαίσιο αυτής της δομής, αναδεικνύοντας πώς η πληροφορία διασυνδέεται με τα διαφορετικά στάδια του κύκλου ζωής του προϊόντος.



Εικόνα 2: Ο κύκλος ζωής ενός Ψηφιακού Διαβατηρίου Προϊόντος. Πηγή: CIRPASS Project (2023) [5].

Όπως παρατηρείται στο παραπάνω σχήμα σε συνδυασμό με το περιεχόμενο του Πίνακα 1, η πληροφορία οργανώνεται σε διακριτά επίπεδα που καλύπτουν όλο τον κύκλο ζωής του προϊόντος. Σύμφωνα με το έργο CIRPASS [6], το οποίο αποτελεί το τεχνικό προσχέδιο (blueprint) για την επιχειρησιακή εφαρμογή του ESPR και ευθυγραμμίζεται πλήρως με τις κατευθυντήριες γραμμές του, χωρίς να εισάγει επιπρόσθετες απαιτήσεις, τα κύρια πεδία πληροφόρησης που συνθέτουν ένα DPP περιλαμβάνουν:

- *Ταυτοποίηση προϊόντος*, με πληροφορίες όπως η εμπορική ονομασία, το μοντέλο, ο μοναδικός κωδικός (GTIN, SKU), ο τύπος προϊόντος και το αναγνωριστικό του URI ή QR code. Η καταγραφή αυτών των πεδίων είναι κρίσιμη για την ενοποιημένη αναγνώριση του προϊόντος σε ψηφιακά οικοσυστήματα.
- *Προέλευση και κατασκευαστής*, με αναφορά στον φορέα παραγωγής, τη χώρα κατασκευής, τις ημερομηνίες παραγωγής και τις βασικές πιστοποιήσεις συμμόρφωσης. Τα πεδία αυτά ενισχύουν την ιχνηλασιμότητα και τη νομιμότητα, και είναι ήδη απαραίτητα σε αρκετούς τομείς της εφοδιαστικής αλυσίδας [23].
- *Σύνθεση και υλικά*, περιλαμβάνοντας δεδομένα για τη χημική ή υλική σύσταση του προϊόντος, την παρουσία επικινδυνών ουσιών (σύμφωνα με REACH ή RoHS) και την ανακυκλωσιμότητα των επιμέρους υλικών. Η πληροφορία αυτή είναι καίρια για την περιβαλλοντική αξιολόγηση και την ασφαλή διαχείριση του προϊόντος στο τέλος του κύκλου ζωής του [7].
- *Ενεργειακή και οικολογική απόδοση*, όπως κατανάλωση ενέργειας, εκπομπές άνθρακα, βαθμολογίες ενεργειακής κλάσης ή οικολογικού αποτυπώματος (π.χ. με βάση το PEF – Product Environmental Footprint). Η παρουσία τέτοιων δεδομένων συνδέεται με την ανάγκη συμμόρφωσης προς τη Ψηφιακή Πράσινη Συμφωνία (Ευρωπαϊκό Green Deal) [18].
- *Επισκευασιμότητα και υποστήριξη*, μέσω στοιχείων όπως ο δείκτης επισκευασιμότητας, κατάλογοι ανταλλακτικών, χρόνοι υποστήριξης ή οδηγίες επιδιόρθωσης. Η κατηγορία αυτή ενισχύει τη βιωσιμότητα, μειώνοντας τα απόβλητα και παρατείνοντας τον κύκλο ζωής των προϊόντων [32].

- *Δυνατότητα ανακύκλωσης και απόρριψης*, που αφορά τη δυνατότητα αποσυναρμολόγησης, τα υλικά προς ιδιαίτερη διαχείριση και τις οδηγίες για ορθό περιβαλλοντικό χειρισμό στο τέλος ζωής. Αυτά τα πεδία στοχεύουν στην ενίσχυση των πρακτικών κυκλικής οικονομίας και στην υποστήριξη των φορέων ανακύκλωσης [17].
- *Το ιστορικό συντήρησης και μεταβίβασης* δύναται να περιλαμβάνει εγγραφές σχετικές με τεχνικό service, αλλαγές κατόχων ή πιστοποιημένες επισκευές. Αν και συχνά καταγράφονται σε προαιρετική βάση, η ύπαρξη αυτών των δεδομένων ενισχύει σημαντικά τη διαφάνεια, αυξάνει την αξιοπιστία των πληροφοριών που συνοδεύουν το προϊόν και διευκολύνει την ασφαλή αγορά μεταχειρισμένων προϊόντων.
- *Κανονιστική συμμόρφωση*, μέσω αναφοράς σε νομικά έγγραφα, πρότυπα, ή επίσημα μητρώα. Αυτό υποστηρίζει τόσο την εποπτεία των αρχών όσο και τη διαφάνεια προς τον καταναλωτή.

**Πίνακας 1: Αντιστοίχιση κατηγοριών πληροφορίας DPP με τα στάδια του κύκλου ζωής.**

| Κατηγορία Πληροφορίας           | Σχεδιασμός & Παραγωγή | Διανομή & Πώληση | Χρήση & Συντήρηση | Τέλος Κύκλου Ζωής (Recycling) |
|---------------------------------|-----------------------|------------------|-------------------|-------------------------------|
| Ταυτοποίηση Προϊόντος           | ✓                     | ✓                | ✓                 | ✓                             |
| Προέλευση & Κατασκευαστής       | ✓                     | ✓                |                   |                               |
| Σύνθεση & Υλικά                 | ✓                     |                  |                   | ✓                             |
| Ενεργειακή & Οικολογική Απόδοση |                       | ✓                | ✓                 |                               |
| Επισκευασιμότητα & Υποστήριξη   |                       |                  | ✓                 |                               |

|                                   |   |   |   |   |
|-----------------------------------|---|---|---|---|
| Ανακύκλωση & Απόρριψη             |   |   |   | ✓ |
| Ιστορικό Συντήρησης & Μεταβίβασης |   |   | ✓ |   |
| Κανονιστική Συμμόρφωση            | ✓ | ✓ | ✓ | ✓ |

Η σημασία της οργάνωσης αυτών των δεδομένων με σημασιολογικό τρόπο, δηλαδή με χρήση γλωσσών και προτύπων του Σημασιολογικού Ιστού (όπως RDF, OWL, JSON-LD), είναι κρίσιμη για την αυτοματοποίηση της επεξεργασίας, τη σύνδεση με εξωτερικά δεδομένα και την πραγματική διαλειτουργικότητα μεταξύ διαφορετικών εφαρμογών. Η αξιοποίηση υπαρχόντων λεξιλογίων, όπως το [schema.org/Product](http://schema.org/Product), το GS1 Digital Link, το ProductOntology ή το ECLASS, καθιστά εφικτή τη σύγκλιση μεταξύ επιχειρησιακής χρήσης και τεχνικής υλοποίησης, ενισχύοντας τη δυνατότητα επαναχρησιμοποίησης της πληροφορίας σε πολλαπλά πληροφοριακά περιβάλλοντα [31] [35].

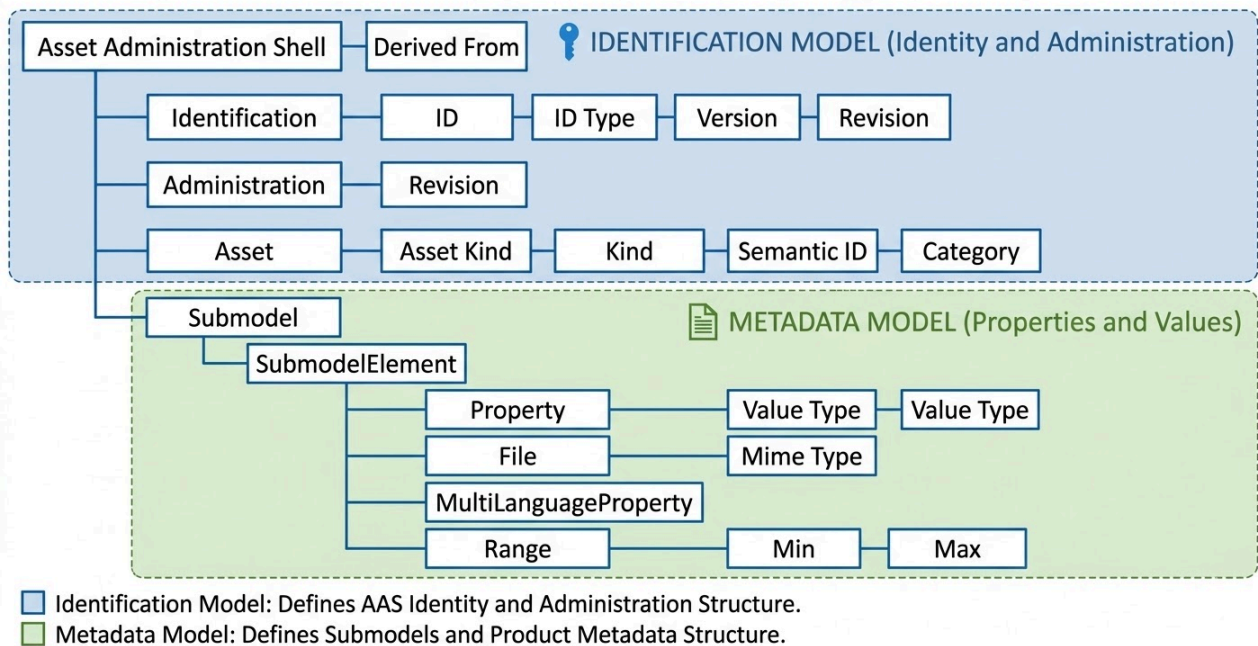
Συνολικά, η δομή των πληροφοριακών πεδίων στα DPPs αντανακλά όχι μόνο τεχνικές απαιτήσεις συστημάτων, αλλά και οικονομικές, ρυθμιστικές και κοινωνικές ανάγκες, τοποθετώντας το DPP στο επίκεντρο της ψηφιακής και πράσινης μετάβασης της βιομηχανίας.

## 2.6 Συσχέτιση με το μοντέλο *asset administration shell* (AAS)

Η προοδευτική ψηφιοποίηση της βιομηχανίας και η μετάβαση προς τα πρότυπα του Industry 4.0 έχουν αναδείξει την ανάγκη για τυποποιημένα ψηφιακά μοντέλα περιουσιακών στοιχείων (assets), τα οποία να υποστηρίζουν την ανταλλαγή πληροφορίας μεταξύ συστημάτων, οργανισμών και φάσεων ζωής του προϊόντος. Σε αυτό το πλαίσιο, το μοντέλο Asset Administration Shell (AAS) που έχει προταθεί από τη Γερμανική Πλατφόρμα Industrie 4.0 αποτελεί μια βασική ψηφιακή δομή για την περιγραφή ενός φυσικού ή λογικού asset, με τρόπο διαλειτουργικό και κατανεμημένο [30].

Το AAS λειτουργεί ως ψηφιακό περίβλημα (digital envelope) που περιλαμβάνει όλες τις απαραίτητες πληροφορίες για ένα asset — όπως τεχνικές προδιαγραφές, πληροφορίες κύκλου ζωής, κατασκευαστικά δεδομένα, οδηγίες λειτουργίας, αλλά και δυναμικά δεδομένα λειτουργίας. Κάθε AAS αποτελείται από ένα Identification layer (Επίπεδο Αναγνώρισης) και ένα Meta-model layer (Επίπεδο Μετα-μοντέλου), το οποίο επιτρέπει τον ορισμό ενός ή περισσότερων submodels,

τα οποία οργανώνουν την πληροφορία σε επιμέρους ενότητες (π.χ. Documentation (Τεκμηρίωση), Technical Data (Τεχνικά Δεδομένα), Maintenance (Διατήρηση)) [25].



**Εικόνα 3: Η αρχιτεκτονική δομή του Asset Administration Shell (AAS) και η οργάνωση των υπο-μοντέλων (submodels).**

Η εννοιολογική συγγένεια ανάμεσα στο AAS και το Digital Product Passport (DPP) είναι εμφανής, καθώς και τα δύο αποσκοπούν στην ψηφιακή αναπαράσταση ενός φυσικού προϊόντος ή συστήματος, με στόχο τη διαμοιραζόμενη, επαναχρησιμοποιήσιμη και αξιόπιστη πληροφόρηση. Ωστόσο, υπάρχει μια θεμελιώδης διαφοροποίηση ως προς τον σκοπό και το επίπεδο εφαρμογής.

Ειδικότερα, το DPP είναι κατά βάση ένα κανονιστικά καθορισμένο σχήμα πληροφορίας που αποσκοπεί στην εκπλήρωση νομικών, περιβαλλοντικών και καταναλωτικών απαιτήσεων, με έμφαση στη διαφάνεια και τη συμμόρφωση σε επίπεδο αγοράς [20]. Από την άλλη μεριά, το AAS εστιάζει σε βιομηχανικές εφαρμογές, αυτοματοποίηση διαδικασιών, και μηχανική διαλειτουργικότητα εντός της εφοδιαστικής ή παραγωγικής αλυσίδας, με έμφαση στη συνδεσιμότητα με PLCs, συστήματα MES/ERP και έξυπνες μονάδες παραγωγής [29].

Παρά τις διαφορές αυτές, έχει παρατηρηθεί ότι τα δύο μοντέλα μπορούν να λειτουργήσουν συμπληρωματικά. Πιο συγκεκριμένα, το AAS μπορεί να αξιοποιηθεί ως τεχνική βάση υλοποίησης ή φορέας πληροφορίας ενός DPP, ιδίως όταν απαιτείται η ενσωμάτωση του DPP σε σενάρια αυτοματοποίησης ή έξυπνης παραγωγής. Σε μια τέτοια προσέγγιση, τα submodels του AAS θα μπορούσαν να περιέχουν σημασιολογικά περιγραφόμενα δεδομένα σύμφωνα με τα πρότυπα DPP, ενώ ταυτόχρονα θα υποστηρίζεται η σύνδεση με φυσικά συστήματα και αισθητήρες, ενισχύοντας τη δυναμική παρακολούθηση κατάστασης προϊόντος (condition monitoring) [3].

Επιπλέον, η διαλειτουργικότητα μεταξύ DPP και AAS είναι εφικτή μέσω της χρήσης κοινών τεχνολογιών μοντελοποίησης, όπως RDF, JSON-LD, σε συνδυασμό με την υιοθέτηση τυποποιημένων διεπαφών επικοινωνίας, όπως τα RESTful APIs. Η χρήση κοινών οντολογιών ή μοντέλων αντιστοίχισης (mapping models) συμπληρώνει αυτή τη δομή, επιτρέποντας την απρόσκοπτη ανταλλαγή πληροφοριών. Πρωτοβουλίες όπως η Open Industry 4.0 Alliance, η IDSA (International Data Spaces Association) και το Catena-X διερευνούν ήδη αυτή τη διασύνδεση,

προσπαθώντας να ενοποιήσουν ρυθμιστικά πλαίσια (όπως το DPP) με ψηφιακά δίδυμα βασισμένα στο AAS [24].

Συμπερασματικά, ενώ τα DPPs και τα AASs διαφέρουν σε στοχευόμενο κοινό και βασικά χαρακτηριστικά, μπορούν να λειτουργήσουν συνεργατικά στο πλαίσιο μιας ενοποιημένης ψηφιακής υποδομής προϊόντων, προσφέροντας υψηλά επίπεδα πληροφορικής ευφυΐας, παρακολούθησης και τεκμηρίωσης σε ολόκληρη την αλυσίδα αξίας.

# 3

## Σχετικές εργασίες

Στο παρόν κεφάλαιο παρουσιάζονται και αναλύονται οι σημαντικότερες υπάρχουσες πρωτοβουλίες, τα ερευνητικά έργα και τα τεχνολογικά πλαίσια που αφορούν την ανάπτυξη και την υλοποίηση των Ψηφιακών Διαβατηρίων Προϊόντων (DPPs). Η ανασκόπηση ξεκινά με το έργο CIRPASS, το οποίο αποτελεί το βασικό τεχνικό προσχέδιο (blueprint) για την εφαρμογή των DPPs σε ευρωπαϊκό επίπεδο, και συνεχίζει με το Battery Pass project, μια εξειδικευμένη πρωτοβουλία για τον κρίσιμο τομέα των μπαταριών. Επιπλέον, εξετάζεται το iPoint DPP framework ως μια ολοκληρωμένη επιχειρησιακή λύση για τη διαχείριση δεδομένων κύκλου ζωής. Το κεφάλαιο ολοκληρώνεται με μια συγκριτική αξιολόγηση των ανωτέρω εργασιών σε άμεση συσχέτιση με την προτεινόμενη πλατφόρμα της παρούσας διπλωματικής, με στόχο την ανάδειξη των τεχνολογικών πλεονεκτημάτων, των καινοτομιών και του βαθμού συμμόρφωσής της με τις τρέχουσες κανονιστικές απαιτήσεις.

### 3.1 CIRPASS project

Μία από τις πιο σημαντικές πρωτοβουλίες στον ευρωπαϊκό χώρο που σχετίζονται με την υλοποίηση Ψηφιακών Διαβατηρίων Προϊόντων είναι το έργο CIRPASS (Collaborative Initiative for a Standards-based Digital Product Passport), το οποίο χρηματοδοτείται από την Ευρωπαϊκή Επιτροπή στο πλαίσιο της Πράσινης Συμφωνίας και της στρατηγικής για την κυκλική οικονομία. Το CIRPASS αποτελεί ένα προπαρασκευαστικό έργο που στοχεύει στη διαμόρφωση ενός ενιαίου πλαισίου αναφοράς για την ανάπτυξη DPPs στους τομείς των ηλεκτρονικών, των κλωστοϋφαντουργικών προϊόντων και των μπαταριών. Η βασική συνεισφορά του έργου είναι η τεκμηρίωση απαιτήσεων και προδιαγραφών, τόσο τεχνικών όσο και λειτουργικών, για την υλοποίηση διαλειτουργικών DPPs, καθώς και η διαμόρφωση ενός πρώτου κοινού μοντέλου δεδομένων [5].

Ανάμεσα στα θετικά σημεία του έργου συγκαταλέγεται η έμφαση στη σημασιολογική μοντελοποίηση, η εμπλοκή ενός μεγάλου αριθμού ενδιαφερόμενων φορέων (stakeholders) και η εστίαση στη συμβατότητα με το κανονιστικό πλαίσιο (ESPR, REACH, RoHS). Το έργο παρέχει σημαντική βιβλιογραφική και τεχνική βάση για όσες προσπάθειες στοχεύουν σε υλοποίηση DPPs. Παρόλα αυτά, το CIRPASS δεν αποτελεί πλήρη τεχνική υλοποίηση ή ανοιχτό λογισμικό σύστημα. Αντίθετα, λειτουργεί κυρίως σε επίπεδο σχεδιασμού και καθορισμού κατευθύνσεων, ενώ η υλοποίηση επιμέρους εργαλείων ή πλατφορμών παραμένει στη διακριτική ευχέρεια των εμπλεκόμενων οργανισμών [21] [5] [11].

Σε σύγκριση με την παρούσα εργασία, το CIRPASS προσφέρει ευρύτερη θεσμική κάλυψη και στρατηγική στόχευση, αλλά υστερεί ως προς την πρακτική υλοποίηση ενός λειτουργικού συστήματος. Η παρούσα διπλωματική, μολονότι κινείται σε μικρότερη κλίμακα, συνεισφέρει μια συγκεκριμένη τεχνική προσέγγιση υλοποίησης DPP πλατφόρμας, ενσωματώνοντας μια RESTful αρχιτεκτονική, τεχνολογίες Σημασιολογικού Ιστού και λειτουργικό περιβάλλον με διαχωρισμό μεταξύ frontend και backend. Επιπλέον, δίνεται έμφαση στη δυναμική αναζήτηση μέσω SPARQL ερωτημάτων, επί των σημασιολογικών δεδομένων, στοιχείο που δεν περιγράφεται ως λειτουργική απαίτηση στο προσχέδιο (blueprint) του CIRPASS, το οποίο εστιάζει κυρίως στην ανάκτηση δεδομένων μέσω μοναδικών αναγνωριστικών (IDs).

### 3.2 *Battery pass project*

Το έργο Battery Pass αποτελεί μια ακόμη σημαντική ευρωπαϊκή πρωτοβουλία που συνδέεται στενά με την ανάπτυξη και εφαρμογή των Digital Product Passports (DPPs), εστιάζοντας ειδικά στις μπαταρίες ως προϊόντα υψηλής στρατηγικής σημασίας για την Πράσινη Συμφωνία και την ενεργειακή μετάβαση. Το έργο ξεκίνησε το 2022 με την υποστήριξη της Γερμανικής κυβέρνησης και ειδικότερα του Υπουργείου Οικονομικών και Κλιματικής Δράσης (BMWK), με στόχο να αναπτύξει ένα τυποποιημένο πλαίσιο DPP για μπαταρίες ηλεκτρικών οχημάτων και σταθερών συστημάτων αποθήκευσης ενέργειας, σύμφωνα με τις επιταγές του Κανονισμού EU Battery Regulation (Κανονισμός (ΕΕ) 2023/1542) [1] [12].

Η βασική συνεισφορά του Battery Pass έγκειται στην εξειδίκευση του DPP σε έναν κρίσιμο τομέα, προσφέροντας ένα αναλυτικό μοντέλο δεδομένων, συμβατό με το θεσμικό πλαίσιο, καθώς και προτάσεις για τη σημασιολογική αναπαράσταση της πληροφορίας και τις τεχνολογικές προσεγγίσεις υλοποίησης. Το μοντέλο βασίζεται σε πρότυπα του Σημασιολογικού Ιστού (RDF, JSON-LD) και υποστηρίζει την αποθήκευση δεδομένων σε αποκεντρωμένες υποδομές (decentralized infrastructures), όπως χώροι δεδομένων (data spaces) και αλυσίδες μπλοκ (blockchains). Επιπλέον, δίνει έμφαση στην προσβασιμότητα και επαναχρησιμοποίηση των δεδομένων με βάση τις αρχές FAIR (Findable, Accessible, Interoperable, Reusable).

Μεταξύ των θετικών χαρακτηριστικών του έργου Battery Pass συγκαταλέγονται η εκτενής τεχνική τεκμηρίωση, η πολυθεματική συνεργασία δημόσιων και ιδιωτικών φορέων, όπως οι SAP, Fraunhofer, Audi και BASF, καθώς και η στρατηγική εστίαση σε θέματα βιωσιμότητας, ανακύκλωσης και διαχείρισης κινδύνων. Η πρωτοβουλία περιλαμβάνει επίσης την ανάπτυξη ενός λογισμικού επίδειξης (software demonstrator) που αποσκοπεί στην πειραματική αξιολόγηση της τεχνικής υλοποίησης ενός ψηφιακού διαβατηρίου για μπαταρίες, καλύπτοντας ολόκληρο τον κύκλο ζωής τους. Ωστόσο, το έργο παρουσιάζει ορισμένους περιορισμούς. Πιο συγκεκριμένα, παραμένει προσανατολισμένο αποκλειστικά στον τομέα των μπαταριών, χωρίς να προσφέρει έναν οριζόντιο σχεδιασμό που να επεκτείνεται και σε άλλες κατηγορίες προϊόντων. Επιπρόσθετα, μέχρι τη στιγμή συγγραφής της παρούσας εργασίας, δεν έχει διατεθεί πλήρως λειτουργική εφαρμογή frontend/backend με ανοιχτό κώδικα για την αναζήτηση και διαχείριση DPPs από τελικούς χρήστες. Παρότι έχει δημοσιευτεί το σχήμα δεδομένων σε δημόσιο αποθετήριο GitHub, το έργο δεν προσφέρει ολοκληρωμένη ψηφιακή εμπειρία σε επίπεδο διεπαφής ή ολοκληρωμένου συστήματος διαλειτουργικότητας [2].

Η παρούσα διπλωματική εργασία διαφοροποιείται σημαντικά, καθώς επιχειρεί να κατασκευάσει ένα πληροφοριακό σύστημα διαχείρισης DPPs, το οποίο, αν και γενικού σκοπού, υλοποιεί λειτουργικό περιβάλλον CRUD λειτουργιών για DPPs, επιλογή μεταξύ πολλαπλών

τρόπων αναζήτησης (λέξεις-κλειδιά, key-value, SPARQL), καθώς και μηχανισμό αυθεντικοποίησης και ελέγχου πρόσβασης με RBAC. Επιπλέον, σχεδιάστηκε ως μια εφαρμογή ιστού βασισμένη σε τεχνολογίες REST API με Python backend, σε συνδυασμό με σημασιολογική αναπαράσταση μέσω RDF και JSON-LD, προσφέροντας ένα πρακτικό και πλήρως υλοποιημένο τεχνικό πλαίσιο που υποστηρίζει την σημασιολογική αναπαράσταση, διαχείριση και αναζήτηση DPPs με ασφαλή τρόπο υποστηρίζοντας τη διαλειτουργικότητα.

### 3.3 iPoint DPP framework

Η εταιρεία iPoint-systems GmbH, με έδρα στη Γερμανία, προσφέρει μία από τις πρώτες εμπορικές πλατφόρμες για την υλοποίηση και διαχείριση Ψηφιακών Διαβατηρίων Προϊόντων (DPPs) με στόχευση κυρίως στη βιομηχανία, και ειδικότερα στους τομείς της αυτοκινητοβιομηχανίας, των ηλεκτρονικών και των πρώτων υλών. Το iPoint DPP Framework αποτελεί μια από άκρο σε άκρο (end-to-end) λύση, η οποία επιτρέπει τη δημιουργία, ενημέρωση, πιστοποίηση και ανταλλαγή DPPs μεταξύ εταιρών της αλυσίδας εφοδιασμού, βασισμένο σε τεχνολογίες Web APIs, Blockchain, και Σημασιολογικού Ιστού [28].

Η βασική συνεισφορά του συστήματος είναι η συνδυαστική ενσωμάτωση νομικών απαιτήσεων, όπως αυτές καθορίζονται από τους Κανονισμούς της Ευρωπαϊκής Ένωσης (π.χ. REACH, RoHS, ESPR, Battery Regulation), με τεχνολογικά πρότυπα και μηχανισμούς ιχνηλασιμότητας. Το πλαίσιο χρησιμοποιεί σημασιολογική μοντελοποίηση των προϊόντων μέσω RDF και JSON-LD, ενώ υποστηρίζει την αποθήκευση δεδομένων σε αποκεντρωμένες υποδομές, αξιοποιώντας πλατφόρμες όπως το IDS (International Data Spaces) και το οικοσύστημα (ecosystem) Catena-X [21] [3].

Μεταξύ των πλεονεκτημάτων της προσέγγισης αυτής είναι:

- Η πλήρης συμμόρφωση με ευρωπαϊκή και διεθνή νομοθεσία
- Η υποστήριξη προηγμένων μοντέλων πρόσβασης και ασφάλειας δεδομένων
- Η δυνατότητα σύνδεσης με συστήματα ERP/MES μεγάλων επιχειρήσεων

Η υιοθέτηση αρχιτεκτονικής μικρο-υπηρεσιών (microservices) και RESTful APIs, που καθιστούν το σύστημα επεκτάσιμο και διαλειτουργικό

Ωστόσο, το iPoint DPP Framework εμφανίζει και ορισμένους περιορισμούς: πρόκειται για ιδιόκτητη (proprietary) λύση, που απαιτεί εμπορική συμφωνία για την πρόσβαση, με περιορισμένη τεκμηρίωση διαθέσιμη στο κοινό. Επίσης, το σύστημα απευθύνεται κυρίως σε μεγάλες επιχειρήσεις, γεγονός που περιορίζει τη δυνατότητα αξιοποίησης του από μικρές και μεσαίες επιχειρήσεις ή από ερευνητικούς φορείς.

Σε σύγκριση με την παρούσα διπλωματική εργασία, το iPoint DPP Framework υλοποιεί ένα πλήρες και επαγγελματικό περιβάλλον παραγωγικής λειτουργίας, αλλά δεν προσφέρει ανοιχτή πρόσβαση ούτε τεκμηριωμένη RESTful διεπαφή για ερευνητικούς ή ακαδημαϊκούς σκοπούς. Η προτεινόμενη υλοποίηση της διπλωματικής στοχεύει στην εκπαιδευτική και ερευνητική τεκμηρίωση ενός πλήρους συστήματος διαχείρισης DPPs, με διακριτό frontend/backend, τεκμηριωμένο API (Swagger<sup>9</sup>), ενσωματωμένες λειτουργίες CRUD και αναζήτησης DPPs, και σημασιολογική αναπαράσταση DPPs σε RDF μέσω SPARQL endpoint. Έτσι, η παρούσα εργασία

<sup>9</sup> Swagger: <https://swagger.io/>

λειτουργεί ως ένα πρωτότυπο (proof-of-concept) με στόχο την επαναχρησιμοποίηση, επέκταση και προσαρμογή σε διαφορετικά πεδία εφαρμογής, ενώ προσφέρει έναν βαθμό τεχνικής διαφάνειας και προσβασιμότητας που δεν διατίθεται σε κλειστές εμπορικές πλατφόρμες [28].

### 3.4 Συγκριτική αξιολόγηση σχετικών εργασιών

Οι τρεις σχετικές εργασίες που παρουσιάστηκαν στις προηγούμενες ενότητες διακρίνονται για τη θεσμική και εμπορική βαρύτητα τους και παρέχουν χρήσιμες βάσεις για την κατανόηση της προοπτικής και της υλοποίησης των Digital Product Passports. Ωστόσο, η παρούσα διπλωματική εργασία επιχειρεί να καλύψει ένα σημαντικό κενό στον χώρο, προτείνοντας μια λειτουργική και επεκτάσιμη αρχιτεκτονική ενός συστήματος DPP, το οποίο μπορεί να προσαρμόζεται τόσο σε μικρές όσο και σε μεγάλες επιχειρήσεις, και επιπλέον να εφαρμόζεται πέραν του βιομηχανικού τομέα, σε περιβάλλοντα όπως ανακύκλωση, επισκευή, δημόσιες αρχές, ή καταναλωτικές εφαρμογές.

Σε αντίθεση με πρωτοβουλίες όπως το CIRPASS και το Battery Pass, που παρέχουν κυρίως θεωρητικές προδιαγραφές, ή το iPoint DPP Framework, που αποτελεί ιδιόκτητη λύση για μεγάλους επιχειρηματικούς πελάτες, το σύστημα που υλοποιείται στο πλαίσιο αυτής της διπλωματικής είναι πλήρως τεκμηριωμένο, ευέλικτο, και προσβάσιμο, με στόχο την ενίσχυση της πρακτικής αξιοποίησης των DPPs από διαφορετικούς ρόλους χρηστών. Ειδικότερα, υποστηρίζονται λειτουργίες για δημιουργία, ανάγνωση, ενημέρωση, δημοσίευση και αναζήτηση DPPs, με πολλαπλές εναλλακτικές μεθόδους επερώτησης (λέξεις-κλειδιά, key-value, SPARQL) ενώ πραγματοποιείται χρήση τεχνολογιών RESTful API και Σημασιολογικού Ιστού (RDF, JSON-LD) καθώς και εφαρμόζεται το RBAC μοντέλου για περισσότερη ασφάλεια.

Η χρήση τεχνολογιών όπως το πλαίσιο FastAPI στην Python για το backend, το Swagger για την αυτοματοποιημένη τεκμηρίωση του API, καθώς και η προγραμματιστική υποστήριξη της frontend διεπαφής με δυνατότητα απόκρισης (responsiveness) σε διαφορετικούς τύπους συσκευών, καθιστούν την παρούσα εργασία ένα ολοκληρωμένο proof-of-concept. Η επιλογή συγκεκριμένων βιβλιοθηκών και πλαισίων εξασφαλίζει ότι το σύστημα δεν είναι απλώς ένα στατικό μοντέλο, αλλά μια επεκτάσιμη πλατφόρμα έτοιμη για πιλοτικές εφαρμογές σε παραγωγικά περιβάλλοντα.

Συνεπώς, η προτεινόμενη λύση διαφοροποιείται όχι μόνο ως προς την υλοποίηση ενός πλήρους λειτουργικού πληροφοριακού συστήματος, αλλά και ως προς τη φιλοσοφία σχεδιασμού, καθώς επιδιώκει να εξυπηρετήσει έναν ευρύτερο και πιο ευέλικτο φάσμα χρηστών, προάγοντας την πρακτική υιοθέτηση των DPPs σε διαφορετικές αγορές, κλίμακες και σενάρια χρήσης.

Η ανάλυση των πρωτοβουλιών και των τεχνολογικών πλαισίων που προηγήθηκε συνοψίζεται στον Πίνακα 1, ο οποίος προσφέρει μια συγκριτική απεικόνιση της απόδοσης των συστημάτων βάσει συγκεκριμένων κριτηρίων αξιολόγησης. Τα κριτήρια αυτά επιλέχθηκαν ώστε να καλύπτουν τόσο τη θεσμική όσο και την τεχνική διάσταση των DPPs, αντικατοπτρίζοντας τις ανάγκες για διαλειτουργικότητα και επιχειρησιακή ετοιμότητα.

#### Ανάλυση των Κριτηρίων Αξιολόγησης

Η επιλογή των κριτηρίων έγινε με στόχο τη σε βάθος κατανόηση του τρόπου με τον οποίο κάθε προσέγγιση αντιμετωπίζει τις προκλήσεις του ψηφιακού διαβατηρίου:

- **Προσβασιμότητα και Τύπος Υλοποίησης:** Προσδιορίζουν το βαθμό ανοικτότητας (π.χ. ακαδημαϊκή, εμπορική) και το επίπεδο ωριμότητας του συστήματος. Εδώ διακρίνουμε αν μια πρωτοβουλία παραμένει σε επίπεδο θεωρητικού πλαισίου ή αν προχωρά σε πλήρη τεχνική υλοποίηση (Proof-of-Concept).
- **Σημασιολογική Μοντελοποίηση:** Αξιολογεί την υιοθέτηση προτύπων, όπως το RDF και το OWL. Η χρήση αυτών των τεχνολογιών εξασφαλίζει ότι τα δεδομένα δεν είναι απλώς αναγνώσιμα, αλλά και ερμηνεύσιμα από μηχανές, επιτρέποντας τη σύνδεση με εξωτερικές πηγές γνώσης.
- **Λειτουργικότητα (CRUD & Αναζήτηση):** Εξετάζει τη δυνατότητα δυναμικής διαχείρισης (δημιουργία, ανάγνωση, επεξεργασία, διαγραφή) και ανάκτησης της πληροφορίας. Η παρούσα εργασία υπερτερεί εδώ, προσφέροντας προηγμένη αναζήτηση μέσω SPARQL και λέξεων-κλειδιών.
- **Διεπαφές και Ασφάλεια:** Εστιάζει στην ύπαρξη τεκμηριωμένων RESTful APIs (π.χ. μέσω FastAPI και Swagger) και μηχανισμών ελέγχου πρόσβασης, όπως το μοντέλο RBAC και τα JWT tokens, τα οποία είναι απαραίτητα για την ασφάλεια των δεδομένων σε βιομηχανικό περιβάλλον.
- **Διαθεσιμότητα UI:** Αξιολογεί την ύπαρξη διεπαφής χρήστη, με έμφαση στο αν αυτή είναι αποκριτική (responsive), δηλαδή αν προσαρμόζεται σωστά σε διαφορετικούς τύπους συσκευών (mobile, desktop).

Πίνακας 2: Συγκριτική αξιολόγηση συναφών εργασιών

| Κριτήριο         | CIRPASS                | Battery Pass               | iPoint DPP Framework    | Προτεινόμενο Σύστημα                |
|------------------|------------------------|----------------------------|-------------------------|-------------------------------------|
| Προσβασιμότητα   | Ακαδημαϊκή/<br>Θεσμική | Ερευνητική/<br>Κανονιστική | Ιδιόκτητη/<br>Εμπορική  | Ανοικτή/<br>Ακαδημαϊκή              |
| Τύπος Υλοποίησης | Θεωρητικό πλαίσιο      | Τεχνική καθοδήγηση         | Πλήρης εμπορική λύση    | Proof-of-Concept με πλήρη υλοποίηση |
| Εστίαση          | Κανονιστική εναρμόνιση | Εξειδίκευση σε μπαταρίες   | Συμμόρφωση επιχειρήσεων | Ευελιξία και προσαρμογή σε          |

|                                        |                                          |                                  |                              |                                                                        |
|----------------------------------------|------------------------------------------|----------------------------------|------------------------------|------------------------------------------------------------------------|
|                                        |                                          |                                  |                              | πολλαπλά<br>περιβάλλοντα                                               |
| <b>Σημασιολογική<br/>μοντελοποίηση</b> | Ναι<br>(προδιαγραφές<br>RDF/JSON-L<br>D) | Ναι (FAIR<br>principles,<br>RDF) | Ναι (JSON-LD,<br>Ontologies) | Ναι (RDF, JSON-LD,<br>SPARQL endpoint)                                 |
| <b>CRUD<br/>λειτουργίες DPPs</b>       | Όχι                                      | Όχι                              | Ναι (μόνο<br>εμπορικά)       | Ναι (δημιουργία,<br>ανάγνωση,<br>επεξεργασία, διαγραφή,<br>δημοσίευση) |
| <b>Αναζήτηση DPPs</b>                  | Όχι                                      | Μερικώς                          | Περιορισμένη                 | Ναι (λέξεις-κλειδιά,<br>key-value, SPARQL)                             |
| <b>RBAC &amp;<br/>Authentication</b>   | Όχι                                      | Όχι                              | Ναι                          | Ναι (RBAC + JWT)                                                       |
| <b>REST API</b>                        | Όχι                                      | Ναι<br>(ατεκμηρίωτο)             | Ναι (ιδιόκτητο)              | Ναι (τεκμηριωμένο με<br>Swagger/OpenAPI)                               |
| <b>Δυνατότητα<br/>χρήσης από MME</b>   | Περιορισμένη                             | Περιορισμένη                     | Όχι (εμπορική<br>άδεια)      | Ναι                                                                    |

|                                |     |     |               |                                                               |
|--------------------------------|-----|-----|---------------|---------------------------------------------------------------|
| ή δημόσιους<br>φορείς          |     |     |               |                                                               |
| Διαθεσιμότητα<br>frontend / UI | Όχι | Όχι | Ναι (κλειστό) | Ναι (αποκριτικό<br>(responsive) UI με<br>παραδείγματα χρήσης) |

# 4

## Ανάπτυξη Συστήματος

Στο παρόν κεφάλαιο αναλύεται η μεθοδολογία ανάπτυξης και η τεχνική υλοποίηση της σημασιολογικής πλατφόρμας για τη διαχείριση και αναζήτηση Ψηφιακών Διαβατηρίων Προϊόντων (DPPs). Η παρουσίαση ακολουθεί τη ροή του μοντέλου καταρράκτη (waterfall model), ξεκινώντας από τον προσδιορισμό των λειτουργικών, μη λειτουργικών και σημασιολογικών απαιτήσεων του συστήματος.

Στη συνέχεια, περιγράφεται ο αρχιτεκτονικός σχεδιασμός της πλατφόρμας μέσω διαγραμμάτων UML και αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του backend, του frontend και του μηχανισμού σημασιολογικής ενοποίησης. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση της διαδικασίας επαλήθευσης και αξιολόγησης του συστήματος, διασφαλίζοντας ότι η πλατφόρμα ανταποκρίνεται πλήρως στις ανάγκες διαλειτουργικότητας και ασφάλειας που θέτουν τα σύγχρονα βιομηχανικά πρότυπα.

### 4.1 Μοντέλο ανάπτυξης: καταρράκτη (waterfall model)

Για την ανάπτυξη της πλατφόρμας διαχείρισης Ψηφιακών Διαβατηρίων Προϊόντων (DPPs), επιλέχθηκε το μοντέλο καταρράκτη (Waterfall Model) ως μεθοδολογία παραγωγής λογισμικού. Το Waterfall αποτελεί ένα δομημένο και γραμμικό μοντέλο ανάπτυξης λογισμικού, στο οποίο κάθε φάση ολοκληρώνεται πριν ξεκινήσει η επόμενη και τα παραδοτέα κάθε σταδίου χρησιμοποιούνται ως είσοδος στο επόμενο [34].

Η επιλογή του συγκεκριμένου μοντέλου στηρίχθηκε στους εξής λόγους:

- ❖ Παρέχει σαφή διαχωρισμό φάσεων με ξεκάθαρα παραδοτέα, κάτι που ευνοεί τη συστηματική τεκμηρίωση μιας διπλωματικής εργασίας.
- ❖ Προσφέρεται για έργα όπου οι λειτουργικές απαιτήσεις είναι σαφώς προσδιορισμένες από την αρχή, όπως συμβαίνει στην παρούσα εργασία.
- ❖ Διευκολύνει τον έλεγχο συνέπειας και επαλήθευσης των απαιτήσεων, ιδιαίτερα σε έργα proof-of-concept, όπου κάθε λειτουργία πρέπει να υλοποιείται με ακρίβεια.
- ❖ Ευθυγραμμίζεται με την δομή της παρούσας αναφοράς, που βασίζεται σε σειριακή περιγραφή των φάσεων ανάπτυξης.

Η προσέγγιση αυτή είναι ιδιαίτερα κατάλληλη για μονοπρόσωπα έργα ή μικρές ομάδες, στις οποίες απαιτείται πλήρης έλεγχος της ροής υλοποίησης και συστηματική τεκμηρίωση κάθε βήματος. Επιπλέον, η χρήση του Waterfall μοντέλου ενισχύει την αναπαραγωγιμότητα της εργασίας, αφού κάθε στάδιο μπορεί να αναλυθεί, αξιολογηθεί και επαναληφθεί εφόσον απαιτηθεί.

Οι κύριες φάσεις του μοντέλου καταρράκτη που υιοθετούνται στην παρούσα εργασία είναι οι εξής:

1. Ανάλυση Απαιτήσεων (Requirements Analysis)
2. Σχεδίαση Συστήματος (System Design)
3. Υλοποίηση (Implementation)
4. Επικύρωση και Επαλήθευση (Verification & Validation)

Κάθε φάση συνοδεύεται από τα αντίστοιχα παραδοτέα και θα αναλυθεί στις επόμενες ενότητες του κεφαλαίου, με στόχο την αναλυτική τεκμηρίωση του κύκλου ζωής του λογισμικού και την παρουσίαση ενός συνεκτικού, λειτουργικού και επεκτάσιμου πληροφοριακού συστήματος [33].

## 4.2 Ανάλυση απαιτήσεων

Η φάση της ανάλυσης απαιτήσεων αποτελεί τη θεμέλια βάση για τον σχεδιασμό και την υλοποίηση ενός πληροφοριακού συστήματος, καθώς καθορίζει με σαφήνεια τις λειτουργίες που πρέπει να υποστηρίζονται, τους χρήστες που θα αλληλεπιδρούν με την εφαρμογή, καθώς και τα ποιοτικά χαρακτηριστικά που αυτή πρέπει να επιδεικνύει. Στο πλαίσιο αυτό, οι απαιτήσεις του συστήματος διαχωρίζονται σε δύο βασικές κατηγορίες: τις λειτουργικές και τις μη λειτουργικές απαιτήσεις.

### 4.2.1 Λειτουργικές απαιτήσεις

#### 1) Ρόλοι χρηστών

Το σύστημα υποστηρίζει τρεις τύπους χρηστών, ο καθένας με διαφορετικό επίπεδο πρόσβασης:

- **Επισκέπτης (viewer):** Μπορεί να βλέπει βασικές πληροφορίες των προϊόντων (όπως την ονομασία, τον κατασκευαστή, το ανθρακικό αποτύπωμα και την ανακυκλωσιμότητα) και να αιτείται εγγραφή.
- **Κανονικός χρήστης (user):** Έχει πλήρη πρόσβαση σε λειτουργίες διαχείρισης DPPs και διαχείρισης του δικού του λογαριασμού.
- **Διαχειριστής (admin):** Έχει πλήρη πρόσβαση σε όλες τις λειτουργίες, καθώς και επιπλέον δυνατότητες διαχείρισης χρηστών (π.χ. δημιουργία/διαγραφή χρηστών, διαγραφή προϊόντων, πρόσβαση σε στατιστικά συστήματος μέσω Grafana).

#### 2) Λειτουργίες διαχείρισης χρηστών

Στην ενότητα αυτή αναλύονται οι λειτουργίες που αφορούν τη διαχείριση των λογαριασμών και την ασφάλεια της πρόσβασης στην πλατφόρμα, ακολουθώντας αυστηρά πρότυπα ελέγχου ώστε να διασφαλίζεται η εγκυρότητα των δεδομένων των DPPs.

### Εγγραφή και Ελεγχόμενη Απόδοση Πρόσβασης

Η διαδικασία δημιουργίας νέων μελών στο σύστημα μπορεί να ξεκινήσει από δύο διαφορετικές αφετηρίες, ανάλογα με τον ρόλο και τον τρόπο εισαγωγής του χρήστη:

- **Εγγραφή Επισκέπτη (Viewer):** Το σύστημα οφείλει να παρέχει τη δυνατότητα στους επισκέπτες να αιτούνται την εγγραφή τους στον ρόλο "Viewer" μέσω της διεπαφής χρήστη. Για την ολοκλήρωση του αιτήματος, θα πρέπει να προβλεφθεί η υποβολή συγκεκριμένων στοιχείων, τα οποία περιλαμβάνουν το όνομα χρήστη (username), το ονοματεπώνυμο, το email, τον κωδικό πρόσβασης και την επαγγελματική ιδιότητα του αιτούντος.
- **Δημιουργία λογαριασμού χρήστη (User) μέσω Διαχειριστή (Admin):** Ο διαχειριστής έχει τη δυνατότητα να δημιουργεί χρήστες (π.χ. με ρόλο USER για μεταφόρτωση DPPs) χρησιμοποιώντας απευθείας τα endpoints του Swagger UI. Θα μπορεί να εισάγει τα στοιχεία του νέου μέλους, επιτρέποντας έναν πιο εποπτευόμενο τρόπο εισόδου στην πλατφόρμα.

Η επιλογή να περιοριστεί η δημιουργία χρηστών (πέραν των Viewers) αποκλειστικά στον Διαχειριστή δεν είναι τυχαία: τα DPPs αποτελούν επίσημα έγγραφα με στοιχεία βιωσιμότητας και κυκλικότητας. Αν η δημιουργία λογαριασμών με δικαιώματα επεξεργασίας ήταν ελεύθερη, θα διακυβευόταν η ακεραιότητα και η αξιοπιστία της βάσης δεδομένων, καθώς οποιοσδήποτε θα μπορούσε να αναρτήσει μη έγκυρες πληροφορίες.

### Ταυτοποίηση και Σύνδεση (Authentication)

Η διαδικασία ταυτοποίησης επιτρέπει σε κάθε εγγεγραμμένο χρήστη να αποκτά πρόσβαση στο σύστημα εισάγοντας τα προσωπικά του διαπιστευτήρια (όνομα χρήστη και κωδικό πρόσβασης) στην αντίστοιχη διεπαφή σύνδεσης. Με την επιτυχή επιβεβαίωση της ταυτότητας του χρήστη, το σύστημα εκκινεί μια ασφαλή συνεδρία με συγκεκριμένη χρονική διάρκεια, η οποία επιτρέπει την αυτόματη αναγνώριση των δικαιωμάτων πρόσβασής του σε όλες τις επιμέρους λειτουργίες της πλατφόρμας.

Ο μηχανισμός αυτός διασφαλίζει ότι ο χρήστης παραμένει συνδεδεμένος και εξουσιοδοτημένος για την εκτέλεση των ενεργειών του, χωρίς την ανάγκη επανεισαγωγής των στοιχείων του σε κάθε μεμονωμένη δραστηριότητα, μέχρι τη λήξη της περιόδου εγκυρότητας της συνεδρίας του.

### Ενημέρωση Στοιχείων και Διαχείριση Κωδικού Πρόσβασης

Η λειτουργία ενημέρωσης στοιχείων επιτρέπει στους ταυτοποιημένους χρήστες, κανονικούς και διαχειριστές, να επικαιροποιούν τα προσωπικά τους δεδομένα μέσω της διεπαφής του προφίλ τους. Κατά τη διαδικασία αυτή, ο χρήστης επιλέγει τα πεδία που επιθυμεί να τροποποιήσει και το σύστημα αναλαμβάνει την ενημέρωση της πληροφορίας, διασφαλίζοντας τη συνέπεια των δεδομένων σε ολόκληρη την πλατφόρμα. Με την ολοκλήρωση της ενέργειας, το σύστημα προβάλλει το επικαιροποιημένο προφίλ, επιβεβαιώνοντας την επιτυχή εκτέλεση της απαίτησης.

Όσον αφορά τη διαχείριση της ασφάλειας, προβλέπεται η δυνατότητα αλλαγής του κωδικού πρόσβασης από τον ίδιο τον χρήστη αποκλειστικά εντός μιας ενεργής συνεδρίας. Για την επικύρωση της ενέργειας, το σύστημα απαιτεί την επαλήθευση του τρέχοντος κωδικού πριν επιτρέψει την εισαγωγή του νέου, διασφαλίζοντας έτσι ότι η αλλαγή πραγματοποιείται από τον νόμιμο κάτοχο του λογαριασμού. Σε περιπτώσεις απώλειας κωδικού, η πλατφόρμα δεν υποστηρίζει αυτοματοποιημένη επαναφορά για λόγους ασφαλείας, καθιστώντας απαραίτητη την επικοινωνία με τον Διαχειριστή. Ο Διαχειριστής, ως υπεύθυνος για την ακεραιότητα της πρόσβασης, αναλαμβάνει τη χειροκίνητη επαναφορά του κωδικού, διατηρώντας τον πλήρη έλεγχο των δικαιωμάτων εισόδου στο σύστημα. Η διαδικασία αυτή ολοκληρώνεται με την ασφαλή κοινοποίηση ενός προσωρινού κωδικού στον χρήστη (μέσω κρυπτογραφημένου καναλιού επικοινωνίας), ο οποίος καλείται να τον αντικαταστήσει άμεσα κατά την επόμενη σύνδεσή του, ώστε να διασφαλιστεί η αποκλειστική γνώση του νέου κωδικού.

### Εμφάνιση και Αναζήτηση Χρηστών Για Διαχειριστές (User Management)

Οι χρήστες με δικαιώματα διαχειριστή έχουν πρόσβαση σε ένα κεντρικό περιβάλλον επισκόπησης του συνόλου των εγγεγραμμένων μελών της πλατφόρμας. Η λειτουργικότητα αυτή περιλαμβάνει:

- **Συγκεντρωτική Προβολή:** Παράθεση των χρηστών σε μορφή λίστας, με εμφάνιση των κρίσιμων αναγνωριστικών στοιχείων (όνομα χρήστη, ονοματεπώνυμο, διεύθυνση ηλεκτρονικού ταχυδρομείου) και των δικαιωμάτων πρόσβασης (ρόλος και υπο-ρόλος).
- **Μηχανισμοί Πλοήγησης και Αναζήτησης:** Για τη βέλτιστη διαχείριση μεγάλου όγκου εγγραφών, το σύστημα υποστηρίζει σελιδοποίηση (pagination) των αποτελεσμάτων. Επιπλέον, παρέχονται εργαλεία φιλτραρίσματος βάσει ρόλου (π.χ. διαχωρισμός Τεχνικών από Ελεγκτές) και δυναμικής αναζήτησης με βάση το όνομα χρήστη ή το email, επιτρέποντας τον ταχύ εντοπισμό συγκεκριμένων λογαριασμών (χρηστών).

### Διαχείριση Προσωπικού Προφίλ (My Profile)

Κάθε αυθεντικοποιημένος χρήστης, ανεξαρτήτως ρόλου, διαθέτει πρόσβαση στην ενότητα του προσωπικού του προφίλ. Μέσω αυτής της διεπαφής, ο χρήστης δύναται να:

- Επισκοπεί τα καταχωρημένα προσωπικά του στοιχεία και τα δικαιώματα που του έχουν αποδοθεί στο σύστημα.
- Επικαιροποιεί πληροφορίες, όπως το ονοματεπώνυμο και τη διεύθυνση ηλεκτρονικού ταχυδρομείου.
- Τροποποιεί τα διαπιστευτήρια εισόδου (κωδικός πρόσβασης) για λόγους ασφαλείας, διασφαλίζοντας τον έλεγχο του λογαριασμού του.

### Διαγραφή Χρήστη

Η διαγραφή ακολουθεί συγκεκριμένη ιεραρχία δικαιωμάτων:

- **Κανονικός Χρήστης / Επισκέπτης:** Έχει περιορισμένα δικαιώματα διαγραφής, τα οποία αφορούν αποκλειστικά τον δικό του λογαριασμό. Δεν του επιτρέπεται η διαγραφή άλλων χρηστών, ανεξαρτήτως του ρόλου τους.
- **Διαχειριστής:** Διαθέτει το ανώτατο επίπεδο δικαιωμάτων, έχοντας τη δυνατότητα να διαγράψει οποιονδήποτε λογαριασμό χρήστη (Κανονικού Χρήστη ή Επισκέπτη) από το σύστημα. Ωστόσο, για την αποφυγή ακούσιου αποκλεισμού από την πλατφόρμα, το

σύστημα απαγορεύει στον διαχειριστή τη διαγραφή του ίδιου του του λογαριασμού (self-deletion).

### Αποσύνδεση Χρήστη

Η αποσύνδεση μπορεί να πραγματοποιηθεί είτε χειρωνακτικά μέσω της διεπαφής, είτε αυτόματα μετά από ένα διάστημα αδράνειας.

## 3) Λειτουργίες διαχείρισης DPPs

### Δημιουργία DPP και Επικύρωση Δεδομένων

Το σύστημα παρέχει μηχανισμό για την εισαγωγή νέων DPPs, διασφαλίζοντας την ακεραιότητα και την πληρότητα των δεδομένων μέσω αυστηρών ελέγχων εγκυρότητας (validation). Κατά τη διαδικασία δημιουργίας ενός νέου DPP, επιτελούνται οι κάτωθι ενέργειες και έλεγχοι προκειμένου να διασφαλιστεί η ποιότητα της εισαγόμενης πληροφορίας:

- **Υποχρεωτικά Πεδία:** Απαιτείται η συμπλήρωση κρίσιμων αναγνωριστικών στοιχείων, όπως ο μοναδικός κωδικός προϊόντος (Product ID/UUID) και ο τίτλος, τα οποία αποτελούν τη βάση για την ιχνηλασιμότητα του προϊόντος.
- **Έλεγχος Μοναδικότητας:** Το σύστημα διενεργεί αυτόματο έλεγχο για την αποφυγή διπλότυπων εγγραφών, απαγορεύοντας τη δημιουργία νέου DPP εάν το Product ID υφίσταται ήδη στη βάση δεδομένων.
- **Δομημένα Δεδομένα (AAS):** Υποστηρίζεται η εισαγωγή σύνθετων δεδομένων σύμφωνα με το πρότυπο Asset Administration Shell (AAS), επιτρέποντας την οργάνωση των πληροφοριών σε υπο-μοντέλα (Submodels) για τεχνικά χαρακτηριστικά, περιβαλλοντικό αποτύπωμα και άλλα μεταδεδομένα.
- **Αυτόματος Συγχρονισμός:** Με την επιτυχή καταχώρηση στη σχεσιακή βάση δεδομένων, το σύστημα ενεργοποιεί αυτόματα διαδικασία συγχρονισμού για τη μεταφορά των δεδομένων στο σημασιολογικό αποθετήριο (Semantic Store), καθιστώντας τα διαθέσιμα για σημασιολογική αναζήτηση.

### Διαχείριση Κατάστασης Δημοσίευσης

Το σύστημα ενσωματώνει μηχανισμό ελέγχου της ορατότητας των DPPs, επιτρέποντας στους χρήστες να διαχειρίζονται τον κύκλο ζωής των προϊόντων τους μεταξύ κατάστασης "Πρόχειρου" (Draft) και "Δημοσιευμένου" (Published).

- **Αρχική Κατάσταση (Default State):** Κάθε νέο DPP που δημιουργείται στο σύστημα τίθεται εξ ορισμού σε κατάσταση "Πρόχειρου" (Draft). Αυτή η σχεδιαστική επιλογή διασφαλίζει ότι τα δεδομένα δεν καθίστανται δημόσια διαθέσιμα προτού ο δημιουργός ολοκληρώσει την εισαγωγή και τον έλεγχο της ορθότητάς τους. Η δημοσίευση απαιτεί ρητή ενέργεια (Explicit Action) από τον χρήστη σε δεύτερο χρόνο.
- **Δημοσίευση (Publish):** Ο ιδιοκτήτης ενός DPP (ή ο Διαχειριστής) δύναται να μεταβάλει την κατάσταση ενός προϊόντος σε "Δημοσιευμένο". Η ενέργεια αυτή καθιστά το DPP ορατό και προσβάσιμο σε ευρύτερες αναζητήσεις εντός της πλατφόρμας, καθώς και σε εξουσιοδοτημένα σημασιολογικά ερωτήματα (SPARQL) από τρίτους χρήστες.
- **Αποδημοσίευση (Unpublish):** Αντίστροφα, παρέχεται η δυνατότητα απόσυρσης ενός DPP από τη δημόσια προβολή. Κατά την αποδημοσίευση, το προϊόν επανέρχεται σε

κατάσταση "Πρόχειρου", παραμένοντας ορατό αποκλειστικά στον ιδιοκτήτη του και στους διαχειριστές του συστήματος.

- **Έλεγχος Δικαιωμάτων:** Η αλλαγή κατάστασης δημοσίευσης αποτελεί προνομιακή ενέργεια, επιτρεπόμενη αποκλειστικά στον δημιουργό/ιδιοκτήτη του DPP και στους διαχειριστές, διασφαλίζοντας τον έλεγχο επί της διάθεσης των πληροφοριών.

### Επεξεργασία και Επικαιροποίηση Δεδομένων DPP

Το σύστημα παρέχει τη δυνατότητα τροποποίησης των στοιχείων ενός υφιστάμενου DPP, επιτρέποντας την επικαιροποίηση τεχνικών προδιαγραφών ή άλλων μεταδεδομένων. Η διαδικασία αυτή διέπεται από συγκεκριμένους κανόνες ακεραιότητας:

- **Δυνατότητα Τροποποίησης:** Ο ιδιοκτήτης (ή ο Διαχειριστής) μπορεί να επεξεργαστεί τον τίτλο, τα περιγραφικά στοιχεία και τα δεδομένα των υπο-μοντέλων (Submodels) του προϊόντος.
- **Περιορισμοί Ακεραιότητας:** Το σύστημα προστατεύει κρίσιμα πεδία που παράγονται αυτόματα από το σύστημα, όπως η ημερομηνία δημιουργίας (timestamp) και το εσωτερικό αναγνωριστικό εγγραφής (Database ID), καθιστώντας τα αμετάβλητα (read-only) για τη διασφάλιση της ιστορικότητας και της ιχνηλασιμότητας.
- **Διαχείριση Product ID:** Σε περίπτωση τροποποίησης του μοναδικού κωδικού προϊόντος (Product ID), διενεργείται εκ νέου έλεγχος μοναδικότητας για την αποφυγή συγκρούσεων με άλλα προϊόντα.
- **Συγχρονισμός:** Κάθε αλλαγή στα δεδομένα ενημερώνει ταυτόχρονα ενημερώνει ταυτόχρονα τα αντίστοιχα επίπεδα αποθήκευσης και σηματολογικής αναπαράστασης, διασφαλίζοντας τη συνέπεια της πληροφορίας σε όλα τα επίπεδα.

### Προβολή DPP

Η προβολή των δεδομένων ενός DPP ακολουθεί αυστηρή πολιτική ελέγχου πρόσβασης, προσαρμοσμένη στον ρόλο του χρήστη και την κατάσταση του προϊόντος:

- **Διαχειριστής (Admin):** Διαθέτει καθολική εποπτεία του συστήματος, έχοντας τη δυνατότητα να προβάλλει όλα τα DPPs που υπάρχουν στην πλατφόρμα, ανεξαρτήτως του δημιουργού τους και της κατάστασης δημοσίευσής τους (Draft ή Published).
- **Χρήστης (User):** Η πρόσβαση διαφοροποιείται ως εξής:
  - Για τα δικά του προϊόντα, έχει πλήρη πρόσβαση σε όλα τα DPPs που έχει δημιουργήσει ο ίδιος, είτε αυτά είναι δημοσιευμένα είτε βρίσκονται ακόμη σε στάδιο προετοιμασίας (Draft).
  - Για προϊόντα άλλων χρηστών, μπορεί να δει μόνο τα DPPs που έχουν δημοσιευτεί (Published). Τα μη δημοσιευμένα (Draft) προϊόντα άλλων χρηστών παραμένουν αόρατα και μη προσβάσιμα, διασφαλίζοντας την εμπιστευτικότητα των δεδομένων.

### Διαγραφή DPP

Η αφαίρεση ενός DPP από το σύστημα αποτελεί κρίσιμη ενέργεια και διέπεται από αυστηρούς κανόνες δικαιωμάτων για την αποφυγή απώλειας δεδομένων:

- Η δυνατότητα διαγραφής ενός DPP παρέχεται αποκλειστικά στον χρήστη που το δημιούργησε (Ιδιοκτήτης) και στους Διαχειριστές (Admins) του συστήματος. Τρίτοι χρήστες δεν έχουν καμία δικαιοδοσία διαγραφής προϊόντων που δεν τους ανήκουν.

- Η διαδικασία διαγραφής οφείλει να είναι οριστική και να συνεπάγεται την πλήρη αφαίρεση της πληροφορίας από όλα τα επίπεδα αποθήκευσης και σημασιολογικής αναπαράστασης του συστήματος. Με τον τρόπο αυτό, το σύστημα εγγυάται την ακεραιότητα των δεδομένων και την αποφυγή παραμονής πλεονάζουσας ή μη έγκυρης πληροφορίας στην πληροφοριακή του υποδομή.
- Πριν την εκτέλεση της διαγραφής, το σύστημα πραγματοποιεί έλεγχο ταυτοποίησης και εξουσιοδότησης για να επιβεβαιώσει ότι ο αιτών χρήστης έχει το νόμιμο δικαίωμα να προβεί στην ενέργεια.

## Εξαγωγή DPP

Για την ενίσχυση της διαλειτουργικότητας και τη δυνατότητα ανταλλαγής δεδομένων με εξωτερικά συστήματα, η πλατφόρμα παρέχει μηχανισμό εξαγωγής των Ψηφιακών Διαβατηρίων Προϊόντων σε τυποποιημένες μορφές του Σημασιολογικού Ιστού.

- Οι χρήστες ανάλογα με τον ρόλο τους (Διαχειριστές, Κανονικοί Χρήστες, Επισκέπτες) και την κατάσταση του DPP (Δημοσιευμένο/Πρόχειρο) δύνανται να εξάγουν το ψηφιακό διαβατήριο σε μορφές του Σημασιολογικού Ιστού, όπως Turtle (.ttl), RDF/XML ή JSON-LD, επιτρέποντας την άμεση αξιοποίηση των δεδομένων από εξωτερικές εφαρμογές και υπηρεσίες. Συγκεκριμένα, οι Διαχειριστές έχουν πλήρη πρόσβαση σε όλα τα DPPs, ενώ οι Κανονικοί Χρήστες και οι Επισκέπτες περιορίζονται στα Δημοσιευμένα (Published) DPPs, με εξαίρεση τους Κανονικούς Χρήστες που έχουν πρόσβαση και στα δικά τους Πρόχειρα (Drafts). Επιπλέον, για την άμεση οπτική επιθεώρηση και αρχειοθέτηση, παρέχεται η δυνατότητα εξαγωγής σε μορφή PDF (Human-Readable), καθώς και σε απλή μορφή JSON για ενσωμάτωση σε συμβατικές εφαρμογές.
- Το εξαγόμενο αρχείο ενσωματώνει το σύνολο των πληροφοριών του DPP (π.χ. περιβαλλοντικό αποτύπωμα, τεχνικές προδιαγραφές, οδηγίες ανακύκλωσης) σε μορφή Διασυνδεδεμένων Δεδομένων (Linked Data). Η δομή αυτή διασφαλίζει ότι η σημασία των δεδομένων παραμένει αναλλοίωτη κατά τη μεταφορά.
- Η εξαγωγή ακολουθεί διεθνή βιομηχανικά πρότυπα (όπως το πλαίσιο του Asset Administration Shell), καθιστώντας το DPP συμβατό με το ευρύτερο οικοσύστημα της Βιομηχανίας 4.0 και των κυκλικών αλυσίδων αξίας.

## Μηχανισμοί Αναζήτησης και Ανάκτησης Πληροφορίας

Η πλατφόρμα ενσωματώνει ένα πολυεπίπεδο σύστημα αναζήτησης, σχεδιασμένο να καλύπτει διαφορετικές ανάγκες χρηστών, από απλή εύρεση προϊόντων έως σύνθετα σημασιολογικά ερωτήματα. Οι μηχανισμοί αυτοί διακρίνονται σε τρεις κατηγορίες:

- **Απλή Αναζήτηση (Simple Search)** Στοχεύει στον γρήγορο εντοπισμό προϊόντων μέσω λέξεων-κλειδιών (keywords). Ο χρήστης εισάγει όρους αναζήτησης σε ένα ενιαίο πεδίο κειμένου. Το σύστημα αναζητά τους όρους σε όλα τα βασικά πεδία του DPP (Τίτλος, Product ID, Κατασκευαστής). Υποστηρίζεται η χρήση λογικών τελεστών (AND/OR) για τον συνδυασμό όρων (π.χ. "Battery AND Lithium"). Απαιτείται ελάχιστη είσοδος από τον χρήστη (μόνο κείμενο).
- **Προηγμένη Αναζήτηση (Advanced Search)** Επιτρέπει τον ακριβή εντοπισμό προϊόντων βάσει συγκεκριμένων κριτηρίων και δομημένων πεδίων. Ο χρήστης ορίζει ζεύγη "Πεδίο - Τιμή" (Key-Value pairs) μέσω μιας δυναμικής φόρμας. Το είδος της αναζήτησης καθορίζεται αυτόματα από το σύστημα βάσει του τύπου δεδομένων του πεδίου αναζήτησης. Συγκεκριμένα:

- **Ακριβής Ταύτιση (Exact Match):** Εφαρμόζεται αυτόματα για μοναδικά αναγνωριστικά όπως το Product ID ή το Serial Number.
- **Μερική Ταύτιση (Partial Match):** Εφαρμόζεται για πεδία κειμένου όπως ο Κατασκευαστής, επιτρέποντας την εύρεση αποτελεσμάτων με βάση τμήμα της λέξης.
- **Αριθμητικοί Τελεστές:** Υποστήριξη συγκρίσεων ( $>$ ,  $<$ ,  $=$ ,  $>=$ ,  $<=$ ) για αριθμητικά πεδία (π.χ. "Χωρητικότητα  $>$  50 kWh", "CO2  $<$  100 kg"), εφόσον οριστούν από τον χρήστη.

Απαιτείται γνώση της δομής των δεδομένων (ποια πεδία υπάρχουν) και επιλογή των κατάλληλων τελεστών σύγκρισης.

- **Σημασιολογική Αναζήτηση (SPARQL Search)** Παρέχει τη μέγιστη ευελιξία για την εκτέλεση πολύπλοκων ερωτημάτων πάνω στον γράφο γνώσης. Ο χρήστης (συνήθως προχωρημένος ή μηχανικός δεδομένων) συντάσσει ερωτήματα στη γλώσσα SPARQL. Επιτρέπει την πλοήγηση στις σχέσεις μεταξύ των οντοτήτων, την ανάκτηση δεδομένων από συγκεκριμένα υπο-μοντέλα (Submodels) και την εκτέλεση συνδυαστικών ερωτημάτων που δεν καλύπτονται από την προηγμένη αναζήτηση. Η εκτέλεση διέπεται από αυστηρούς κανόνες πρόσβασης, επιτρέποντας ερωτήματα μόνο σε δημοσιευμένα (Published) δεδομένα για τους απλούς χρήστες. Απαιτείται εξειδικευμένη γνώση της γλώσσας SPARQL.

## Ταξινόμηση Αποτελεσμάτων

Για τη βέλτιστη παρουσίαση των αποτελεσμάτων αναζήτησης, το σύστημα υποστηρίζει μηχανισμούς ταξινόμησης που επιτρέπουν στον χρήστη να ιεραρχεί τα ευρήματα βάσει της συνάφειάς τους.

- **Δυναμική Απόδοση Βαρών (User-Defined Weighting)** Κατά την εκτέλεση μιας αναζήτησης, ο χρήστης (μέσω της διεπαφής) έχει τη δυνατότητα να ορίσει συντελεστές βαρύτητας (weights) για συγκεκριμένα πεδία του DPP.  
Παράδειγμα: Ένας χρήστης μπορεί να ορίσει ότι το πεδίο "Αποδοτικότητα" έχει βαρύτητα 0.7 και το "Κόστος" 0.3.
- **Υπολογισμός Βαθμολογίας (Scoring)** Το σύστημα υπολογίζει μια συνολική βαθμολογία (score) για κάθε DPP που πληροί τα κριτήρια αναζήτησης, βασιζόμενο στο σταθμισμένο άθροισμα των τιμών των επιλεγμένων πεδίων. Ο υπολογισμός πραγματοποιείται δυναμικά στο επίπεδο της βάσης δεδομένων, πολλαπλασιάζοντας την κανονικοποιημένη τιμή κάθε πεδίου με το αντίστοιχο βάρος που όρισε ο χρήστης. Η κανονικοποίηση των τιμών σε κοινή κλίμακα (π.χ. 0-1) διασφαλίζει ότι πεδία με μεγάλες απόλυτες τιμές δεν κυριαρχούν στο τελικό αποτέλεσμα, επιτρέποντας την ισότιμη σύγκριση διαφορετικών χαρακτηριστικών.

Τα αποτελέσματα παρουσιάζονται σε φθίνουσα σειρά βάσει αυτής της βαθμολογίας, αναδεικνύοντας τα προϊόντα που ταιριάζουν περισσότερο στις προτεραιότητες του χρήστη.

#### 4.2.2 Μη λειτουργικές απαιτήσεις

Οι ποιοτικές απαιτήσεις του συστήματος εστιάζουν στη χρηστικότητα, ασφάλεια, αποδοτικότητα, αξιοπιστία και επεκτασιμότητα της πλατφόρμας.

| Κατηγορία                  | Προδιαγραφή                                                                                                                                                                                                               |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Χρηστικότητα</b>        | Ενιαίο look-and-feel διεπαφής χρήστη (UI) με αποκριτική (responsive) σχεδίαση (Web, Mobile, Tablet), πλοηγησιμότητα, δυναμική HTML/Ajax                                                                                   |
| <b>Ασφάλεια</b>            | Εφαρμογή RBAC, JWT, κρυπτογράφησης/κατακερματισμού κωδικών, CSRF/TLS προστασίας, προστασία έναντι επιθέσεων Injection (έγχυσης) (SQL, XSS, Command) μέσω της χρήσης ORM και αυτόματης εξουδετέρωσης (escaping) δεδομένων. |
| <b>REST Maturity Level</b> | Επίπεδο 3 REST με τεκμηρίωση Swagger                                                                                                                                                                                      |
| <b>Απόδοση</b>             | Χρόνος απόκρισης < 3 δευτερόλεπτα ανά λειτουργία                                                                                                                                                                          |
| <b>Αξιοπιστία</b>          | Ελάχιστη εμφάνιση σφαλμάτων: < 20 σφάλματα/μήνα                                                                                                                                                                           |
| <b>Φορητότητα</b>          | Dockerization με δυνατότητα εγκατάστασης μέσω script ή Docker Compose                                                                                                                                                     |
| <b>Κλιμάκωση</b>           | Υποστήριξη ταυτόχρονης χρήσης από τουλάχιστον 10 χρήστες                                                                                                                                                                  |

#### 4.2.3 Παρατηρήσεις και τεκμηρίωση

Η καταγραφή των παραπάνω απαιτήσεων προέκυψε από τη συστηματική μελέτη προτύπων, τεχνολογιών και σχετικών υλοποιήσεων, τα αποτελέσματα της οποίας καταγράφονται και αναλύονται διεξοδικά στα Κεφάλαια 2 και 3 της παρούσας αναφοράς. Η διαδικασία αυτή επέτρεψε τον εντοπισμό των κενών στις υπάρχουσες λύσεις και τον καθορισμό των προδιαγραφών που πρέπει να πληροί η προτεινόμενη πλατφόρμα.

Η πλήρης κάλυψη των λειτουργικών απαιτήσεων αποτελεί τον κεντρικό άξονα της υλοποίησης, διασφαλίζοντας ότι το σύστημα ανταποκρίνεται στις ανάγκες των χρηστών για διαχείριση και αναζήτηση DPPs. Παράλληλα, οι μη λειτουργικές απαιτήσεις λειτουργούν ως ο τεχνικός οδηγός για την ποιοτική αξιολόγηση της πλατφόρμας, με ιδιαίτερη έμφαση στη χρηστικότητα, την ασφάλεια και την τεχνική αρτιότητα της αρχιτεκτονικής.

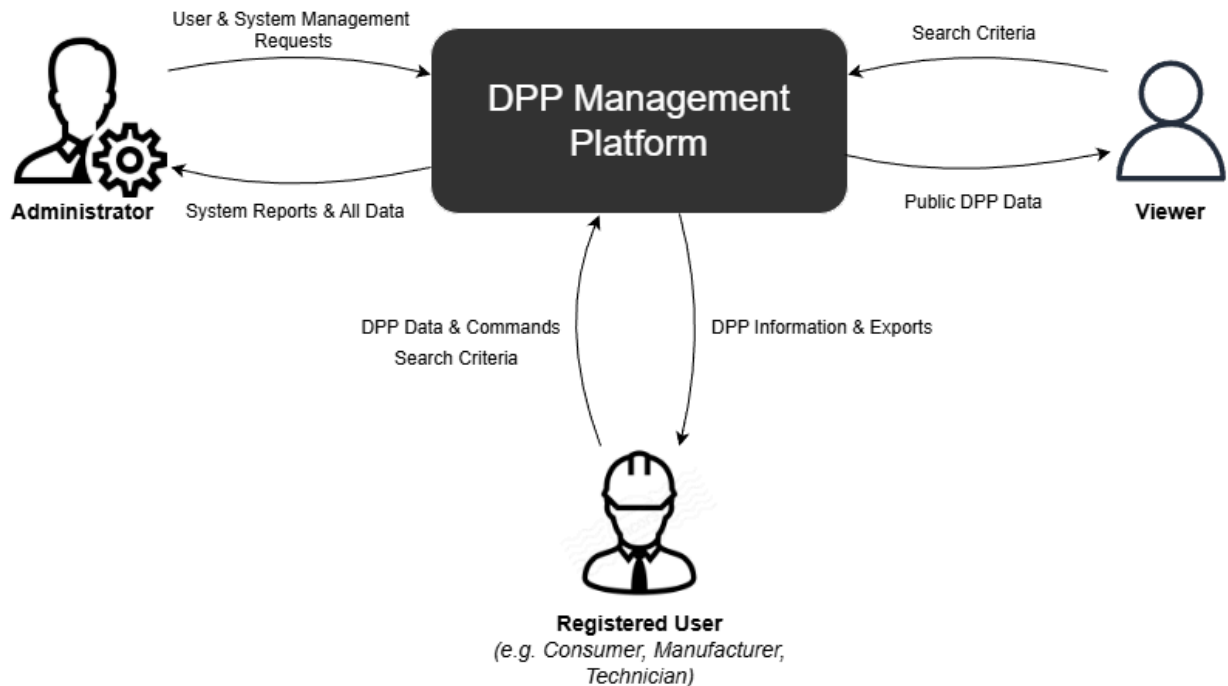
### 4.3 Σχεδίαση συστήματος

Η φάση της σχεδίασης αποτελεί το μεταβατικό στάδιο από την ανάλυση των απαιτήσεων στην τεχνική υλοποίηση, προσφέροντας μια λεπτομερή αρχιτεκτονική και λειτουργική απεικόνιση της προτεινόμενης πλατφόρμας. Στο πλαίσιο αυτό, η δομή και η δυναμική συμπεριφορά του συστήματος μοντελοποιούνται συστηματικά μέσω μιας σειράς σχεδιαστικών διαγραμμάτων της γλώσσας UML, τα οποία αποτυπώνουν τα σχεδιαστικά αποτελέσματα της μελέτης. Ειδικότερα, στις υποενότητες που ακολουθούν παρουσιάζονται τα διαγράμματα περιβάλλοντος, συστατικών, περιπτώσεων χρήσης, κλάσεων, ακολουθίας και δραστηριοτήτων, παρέχοντας μια πολυεπίπεδη τεχνική τεκμηρίωση που διασφαλίζει τη συνοχή, την ορθή ροή των δεδομένων και τη μελλοντική επεκτασιμότητα της υλοποίησης.

#### 4.3.1 Διάγραμμα περιβάλλοντος

Το διάγραμμα περιβάλλοντος (Context Diagram) αποτυπώνει τη συνολική εικόνα του συστήματος μας, το οποίο ονομάζεται DPP Management Platform, παρουσιάζοντάς το ως μια ενιαία οντότητα (black box) που αλληλεπιδρά με το εξωτερικό της περιβάλλον. Στόχος του διαγράμματος είναι να προσφέρει μια καθαρή, συνοπτική θεώρηση των εξωτερικών οντοτήτων που επικοινωνούν με την πλατφόρμα και των ροών δεδομένων που ανταλλάσσονται μεταξύ τους. Με αυτόν τον τρόπο, διευκολύνεται η κατανόηση του τρόπου λειτουργίας του συστήματος, προσδιορίζοντας τα όριά του και τον τρόπο που συμμετέχει σε καθολικές/ευρύτερες (επιχειρησιακές) διαδικασίες, χωρίς να απαιτείται ανάλυση της εσωτερικής του δομής.

## Context Diagram



Εικόνα 4: Διάγραμμα περιβάλλοντος της πλατφόρμας διαχείρισης DPPs

Στην περίπτωση της πλατφόρμας διαχείρισης Ψηφιακών Διαβατηρίων Προϊόντων (Digital Product Passports – DPPs), οι βασικές εξωτερικές οντότητες είναι οι ακόλουθες:

### *Διαχειριστής (Administrator)*

- Input: User & DPPs Management Requests (Είσοδος: Αιτήματα Διαχείρισης Χρηστών & DPPs)

Αυτή η ροή συγκεντρώνει όλες τις εντολές διαχείρισης που στέλνει ο Admin. Περιλαμβάνει εντολές για δημιουργία, τροποποίηση και διαγραφή λογαριασμών χρηστών (Users/Viewers) και εκτέλεση εξειδικευμένων σημασιολογικών ερωτημάτων (Raw SPARQL) για συντήρηση του γράφου.

- Output: System Reports & Full Data Access (Εξοδος: Αναφορές Συστήματος & Πλήρης Πρόσβαση Δεδομένων)

Αυτή η ροή περιγράφει την πληροφορία που επιστρέφει το σύστημα στον Admin. Επίσης, προβολή όλων των DPPs (Draft και Published) και των στοιχείων όλων των χρηστών, χωρίς τους περιορισμούς που ισχύουν για τους απλούς χρήστες.

### *Εγγεγραμμένος Χρήστης (Registered User)*

- Input: DPP Data & Commands (Είσοδος: Δεδομένα & Εντολές DPP)

Περιλαμβάνει το σύνολο των δεδομένων και εντολών που εισάγει ο χρήστης για τα προϊόντα του, τεχνικά χαρακτηριστικά, περιβαλλοντικά δεδομένα και υπο-μοντέλα AAS κατά τη δημιουργία ή επεξεργασία ενός DPP και αιτήματα για δημοσίευση (Publish), αποδημοσίευση (Unpublish) ή διαγραφή των δικών του DPPs, καθώς και αιτήματα για εξαγωγή (Export) δεδομένων. Τέλος, καλύπτει και τις ενέργειες διαχείρισης του προσωπικού λογαριασμού του χρήστη (π.χ. αλλαγή κωδικού πρόσβασης, ενημέρωση στοιχείων προφίλ).

- Input: Search Criteria (Είσοδος: Κριτήρια Αναζήτησης)

Οι παράμετροι αναζήτησης που εισάγει ο χρήστης, όπως λέξεις-κλειδιά (Simple Search) ή φίλτρα πεδίων (Advanced Search) για την εύρεση προϊόντων.

- Output: DPP Details & Exported Files (Εξοδος: Λεπτομέρειες DPP & Αρχεία Εξαγωγής)

Η πληροφορία που λαμβάνει ο χρήστης από το σύστημα, δηλαδή η πλήρης προβολή των δικών του προϊόντων και προβολή των δημοσιευμένων προϊόντων τρίτων, ταξινομημένη βάσει ημερομηνίας δημιουργίας ή συνάφειας (σε περίπτωση αναζήτησης). Επίσης, περιλαμβάνει τα παραγόμενα αρχεία (JSON, PDF) αποκλειστικά όταν ο χρήστης ζητά την εξαγωγή (Export) ενός συγκεκριμένου DPP.

### ***Επισκέπτης (Viewer)***

Οι ροές για τον Επισκέπτη είναι περιορισμένες και αφορούν μόνο την ανάκτηση δημόσιας πληροφορίας:

- Input: Search Criteria (Είσοδος: Κριτήρια Αναζήτησης)

Λέξεις-κλειδιά και φίλτρα που εισάγει ο επισκέπτης για να αναζητήσει προϊόντα στη δημόσια βάση του συστήματος.

- Output: Public DPP Data (Εξοδος: Δημόσια Δεδομένα DPP)

Τα δεδομένα που επιστρέφει το σύστημα, τα οποία περιορίζονται αυστηρά σε δημοσιευμένα (Published) DPPs, δηλαδή μόνο τα προϊόντα που οι ιδιοκτήτες τους έχουν επιλέξει να κάνουν δημόσια σε read-only προβολή, οπότε ο επισκέπτης βλέπει τα δεδομένα χωρίς δυνατότητα επεξεργασίας, διατηρώντας όμως το δικαίωμα εξαγωγής (Export) για τα δημόσια στοιχεία.

Συνολικά, το διάγραμμα παρουσιάζει τη **DPP Management Platform** ως το κεντρικό σημείο ενοποίησης και επικοινωνίας ανάμεσα σε χρήστες και την πλατφόρμα. Η οπτική αυτή προσέγγιση βοηθά στην κατανόηση των βασικών λειτουργικών ροών και αποτελεί θεμελιώδες βήμα για τη μετάβαση στις επόμενες υπο-φάσεις σχεδίασης της εφαρμογής, οι οποίες καλύπτουν την εσωτερική δομή και την συμπεριφορά του.

### 4.3.2 Διάγραμμα συστατικών

Η αρχιτεκτονική της πλατφόρμας για τη διαχείριση Ψηφιακών Διαβατηρίων Προϊόντων (DPPs) βασίζεται σε ένα αρθρωτό, πολυεπίπεδο μοντέλο με σαφή διαχωρισμό μεταξύ frontend, backend, και σημασιολογικής υποδομής. Η σχεδίαση αυτή εξασφαλίζει επεκτασιμότητα, ευκολία συντήρησης και υψηλή επαναχρησιμοποίηση.

Βασικά συστατικά του συστήματος:

#### 1. Presentation Layer (Frontend Module)

Αποτελεί το κύριο σημείο αλληλεπίδρασης (Single Page Application) με τις εξωτερικές οντότητες, δηλαδή τα 3 είδη χρηστών που προδιαγράφηκαν προηγουμένως. Υποστηρίζει:

- **Διαχείριση Χρηστών (User Management):** Διεπαφές για Αυθεντικοποίηση (Login), Εγγραφή (Registration) και διαχείριση προφίλ.
- **Πίνακα Ελέγχου Διαχείρισης DPP (DPP Management Dashboard):** Παροχή πίνακα ελέγχου με δυνατότητες υποστήριξης λειτουργιών CRUD για τα ψηφιακά διαβατήρια προϊόντων, μεταφόρτωσης αρχείων βιομηχανικών προτύπων (AASX/ZIP) (κατά την δημιουργία DPP) και διαχείρισης της κατάστασης δημοσίευσης (Draft/Published).
- **Διεπαφή Προηγμένης Αναζήτησης (Advanced Search Interface):** Υποστηρίζει τη δυναμική αναζήτηση με χρήση απλών λέξεων-κλειδιών (Simple Search) ή σύνθετων κριτηρίων (Advanced Search) και την οπτικοποίηση των αποτελεσμάτων. Επιπλέον, παρέχει εξειδικευμένο περιβάλλον για την εκτέλεση σημασιολογικών ερωτημάτων (SPARQL Queries) απευθείας στον γράφο γνώσης, επιτρέποντας την ανάκτηση σύνθετων συσχετίσεων.
- **Οπτικοποίηση & Εξαγωγή Δεδομένων (Data Extraction & Visualization):** Εξαγωγή δεδομένων σε μορφή PDF/JSON και οπτικοποίηση του σημασιολογικού υπο-γράφου (Graph Visualization) για κάθε προϊόν.

#### 2. Business Logic Layer (Backend API Application)

Αποτελεί τον πυρήνα του συστήματος. Περιλαμβάνει τα εξής συστατικά (Components):

- **Διαχειριστής Ασφάλειας & Αυθεντικοποίησης (Security & Auth Manager):** Μεσισμικό (Middleware) για την έκδοση και επαλήθευση JWT tokens, διασφαλίζοντας ότι κάθε κλήση στο API του συστήματος είναι εξουσιοδοτημένη.
- **Ελεγκτής Χρηστών (User Controller):** Διαχειρίζεται τους λογαριασμούς χρηστών, τους ρόλους και τα προφίλ.
- **Ελεγκτής DPP (DPP Controller):** Διαχειρίζεται τη δημιουργία, ενημέρωση και τον κύκλο ζωής (Publish/Draft) των ψηφιακών διαβατηρίων, καθώς και τη δρομολόγηση των ερωτημάτων SPARQL προς τον Σημασιολογικό Σύνδεσμο (Semantic Connector).
- **Μηχανή Αναζήτησης & Κατάταξης (Search & Ranking Engine):** Υλοποιεί την επιχειρησιακή λογική για την απλή και προηγμένη αναζήτηση, αξιοποιώντας τις δυνατότητες ευρετηρίασης της PostgreSQL (JSONB) για βέλτιστη απόδοση. Υποστηρίζει Weighted Ranking (σταθμισμένη κατάταξη) και σύνθετα φίλτρα, λειτουργώντας συμπληρωματικά προς το σημασιολογικό αποθετήριο που εξυπηρετεί τα SPARQL ερωτήματα. Η επιλογή αυτής της υβριδικής προσέγγισης αποτελεί στρατηγική σχεδιαστική απόφαση: ενώ το Jena Fuseki υποστηρίζει εγγενώς αναζήτηση, η χρήση της

PostgreSQL προσφέρει σημαντικά ταχύτερη απόκριση και μεγαλύτερη ευελιξία στην υλοποίηση πολύπλοκων αλγορίθμων κατάταξης, αφήνοντας στο Fuseki τον ρόλο της εξυπηρέτησης αμιγώς σημασιολογικών ερωτημάτων.

- **Επεξεργαστής Αρχείων (File Processor):** Υποσύστημα που διαχειρίζεται τη μεταφόρτωση αρχείων και την επεξεργασία πακέτων **.aasx**, εκτελώντας αποσυμπίεση και αυτόματη εξαγωγή πεδίων μέσω τεχνικών Fuzzy Matching.
- **Σημασιολογικός Σύνδεσμος (Semantic Connector):** Λειτουργεί ως γέφυρα μεταξύ του σχεσιακού και του σημασιολογικού κόσμου, με κύριο σκοπό τον αυτόματο συγχρονισμό των δεδομένων. Μετασχηματίζει τα δεδομένα από τη βάση (JSONB) σε σημασιολογικές τριάδες RDF (Turtle format) και τα εισάγει ασύγχρονα στο Fuseki. Μέσω αυτής της διαδικασίας, διασφαλίζεται η ενημέρωση του Γράφου Γνώσης, επιτρέποντας την άμεση αξιοποίηση των δεδομένων για σημασιολογική αναζήτηση (SPARQL) και οπτικοποίηση γράφου.

### 3. Data & Infrastructure Layer

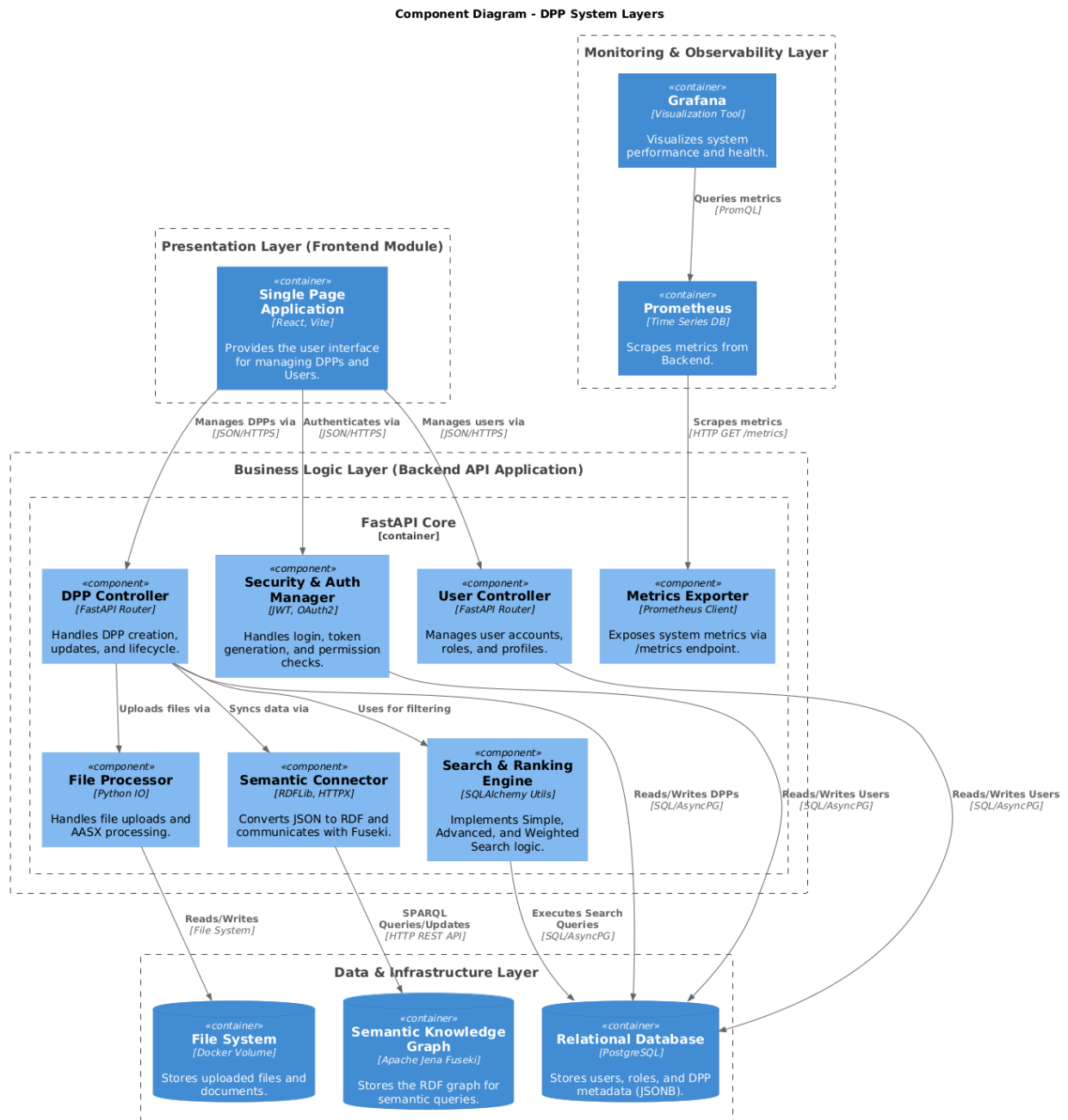
Περιλαμβάνει τις υποδομές αποθήκευσης δεδομένων:

- **Σχεσιακή Βάση Δεδομένων (Relational Database - PostgreSQL/Supabase):** η οποία λειτουργεί ως μέσο αποθήκευσης των διαφορετικών οντοτήτων του συστήματος ως εξής:
  - **Αποθήκη Χρηστών & Ρόλων (User & Role Store):** Αποθήκευση στοιχείων χρηστών και ρόλων για την εφαρμογή του RBAC.
  - **Αποθήκη DPP σε JSONB Μορφή (DPP JSONB Store):** Κεντρική αποθήκευση των Ψηφιακών Διαβατηρίων Προϊόντων όπου μεταβλητά δεδομένα του κάθε προϊόντος (π.χ. περιβαλλοντικό αποτύπωμα, οδηγίες ανακύκλωσης, τεχνικές προδιαγραφές) αποθηκεύονται σε στήλη τύπου JSONB. Η επιλογή αυτής της τεχνολογίας επιτρέπει στο σύστημα να διαχειρίζεται την ετερογένεια των DPPs διαφορετικών κλάδων. Η εσωτερική δομή των δεδομένων ακολουθεί το πρότυπο AAS (Asset Administration Shell), διασφαλίζοντας ότι το Ψηφιακό Διαβατήριο είναι τεχνικά συμβατό με τα πρότυπα της Βιομηχανίας 4.0, συνδυάζοντας έτσι την επιχειρησιακή απαίτηση (DPP) με την τεχνική τυποποίηση AAS.
- **Σημασιολογικός Γράφος Γνώσης (Semantic Knowledge Graph - Apache Jena Fuseki):** το συστατικό αυτό επιτρέπει την διαχείριση του σημασιολογικού γράφου γνώσης των DPPs ενώ την περιτυλίγει μέσω ενός API που παρέχει δυνατότητες επερώτησης του γράφου αυτού:
  - **Ονοματισμένοι Γράφοι (Named Graphs):** Κάθε DPP αποθηκεύεται ως ένας διακριτός γράφος με βάση το UUID του.
  - **SPARQL REST API:** Παρέχει τη δυνατότητα εκτέλεσης σύνθετων σημασιολογικών ερωτημάτων που επιτρέπουν τη διασύνδεση των προϊόντων με εξωτερικές πηγές γνώσης.
- **Σύστημα Αρχείων (File System - Docker Volume):** Χώρος για την προσωρινή αποθήκευση και επεξεργασία των ανεβασμένων αρχείων AASX πριν την εισαγωγή τους στη βάση.

### 4. Monitoring & Observability Layer

Λειτουργεί ως οριζόντιο επιχειρησιακό επίπεδο (operational layer) το οποίο καλύπτει τόσο την υποδομή όσο και τη διεπαφή χρήστη, για την εποπτεία της καλής λειτουργίας και της απόδοσης του συστήματος σε πραγματικό χρόνο. Περιλαμβάνει:

- **Prometheus (Time Series Database):** Συλλέγει αυτόματα μετρικές (metrics) από το Backend API ανά τακτά χρονικά διαστήματα (scraping). Καταγράφει δεδομένα όπως ο αριθμός των αιτημάτων (HTTP Requests), ο χρόνος απόκρισης (Latency), η χρήση πόρων (CPU/Memory) και τα σφάλματα συστήματος.
- **Grafana (Visualization Tool):** Συνδέεται με τον Prometheus και παρέχει οπτικοποίηση των μετρικών μέσω διαδραστικών πινάκων ελέγχου (Dashboards). Επιτρέπει στους διαχειριστές να παρακολουθούν την υγεία του συστήματος, να εντοπίζουν συμφορήσεις (bottlenecks) και να λαμβάνουν ειδοποιήσεις σε περίπτωση δυσλειτουργίας.



Εικόνα 5: Διάγραμμα συστατικών της πλατφόρμας διαχείρισης DPPs

### 4.3.3 Διαγράμματα περιπτώσεων χρήσης

Το Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram) αποτυπώνει τις λειτουργικές απαιτήσεις του συστήματος από την πλευρά των τελικών χρηστών. Περιγράφει τις αλληλεπιδράσεις μεταξύ των δρώντων (actors) και του συστήματος στο πλαίσιο των διαθέσιμων λειτουργιών, ορίζοντας το εύρος (scope) της πλατφόρμας διαχείρισης DPP.

Το πρώτο διάγραμμα εστιάζει στη διαχείριση των οντοτήτων πρόσβασης και την εποπτεία της πλατφόρμας.

#### Ανάλυση Διαγράμματος Περιπτώσεων Χρήσης: Διαχείριση Χρηστών (User & System Management)

##### 1. Δρώντες (Actors)

Στο σύστημα αναγνωρίζονται τρεις ιεραρχικοί ρόλοι, όπου κάθε ρόλος κληρονομεί τα δικαιώματα του προηγούμενου (σχέση ISA):

- **Viewer (Επισκέπτης):** Αποτελεί τον βασικό ρόλο εισόδου στο σύστημα. Έχει τη δυνατότητα να εγγραφεί στην πλατφόρμα και να πραγματοποιήσει σύνδεση. Έχει δικαιώματα μόνο Ανάγνωσης (Read-Only).
- **Registered User (Εγγεγραμμένος Χρήστης):** Είναι ο χρήστης με δικαιώματα Δημιουργίας (Create/Edit). Ο ρόλος αυτός αποδίδεται αποκλειστικά από τον Διαχειριστή (Admin) (είτε μέσω αναβάθμισης Viewer, είτε μέσω απευθείας δημιουργίας). Δεν μπορεί κάποιος να εγγραφεί μόνος του ως User. Στο πλαίσιο της διαχείρισης λογαριασμού, ο χρήστης αυτός διαθέτει πλήρη αυτονομία στην ενημέρωση των προσωπικών του στοιχείων και στην ασφάλεια του προφίλ του.
- **Administrator (Διαχειριστής):** Κληρονομεί όλα τα δικαιώματα του Registered User, αλλά διαθέτει επιπλέον προνομιακή πρόσβαση για τη διαχείριση τρίτων χρηστών και την εποπτεία της καλής λειτουργίας του συστήματος.

##### 2. Περιπτώσεις Χρήσης (Use Case)

Οι κύριες περιπτώσεις χρήσης κατηγοριοποιούνται ως εξής:

###### A. Ενότητα Διαχείρισης Χρηστών (User Management Module)

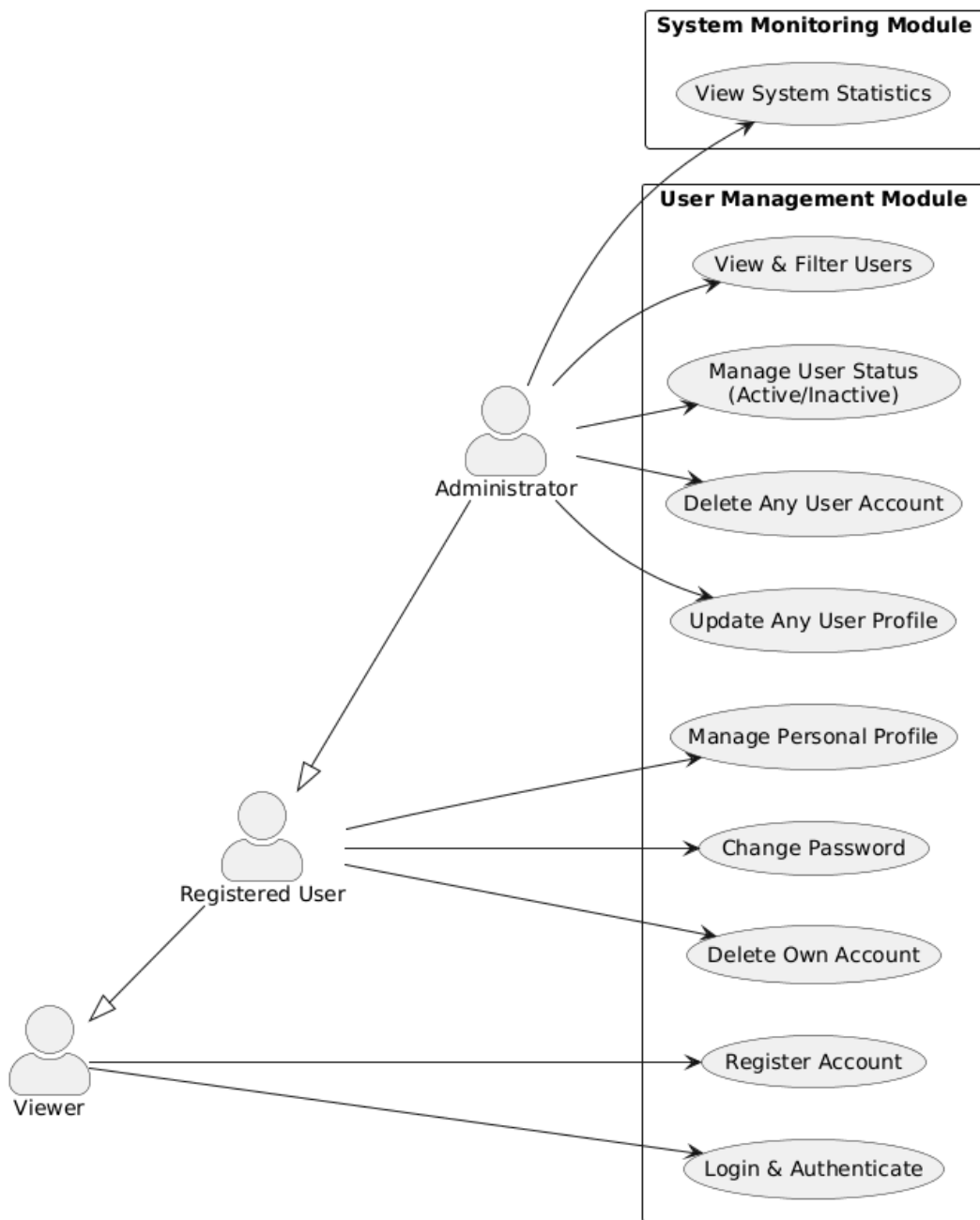
- **Register Account (Εγγραφή Λογαριασμού):** Επιτρέπει σε έναν νέο επισκέπτη να δημιουργήσει λογαριασμό στην πλατφόρμα, ορίζοντας τα στοιχεία του (username, email, password, κλπ.).
- **Login & Authenticate (Σύνδεση & Αυθεντικοποίηση):** Η διαδικασία πιστοποίησης των στοιχείων εισόδου και η συνεπακόλουθη έκδοση του JWT token, το οποίο απαιτείται για την πρόσβαση σε προστατευμένες λειτουργίες.
- **Manage Personal Profile (Διαχείριση Προσωπικού Προφίλ):** Επιτρέπει στον χρήστη να ενημερώσει τα προσωπικά του στοιχεία (π.χ. ονοματεπώνυμο, email).
- **Change Password (Αλλαγή Κωδικού):** Λειτουργία ασφαλείας για την τροποποίηση του κωδικού πρόσβασης από τον ίδιο τον χρήστη.
- **Delete Own Account (Διαγραφή Προσωπικού Λογαριασμού):** Δίνει τη δυνατότητα στον χρήστη να διαγράψει οριστικά τον λογαριασμό του από το σύστημα.

- View & Filter Users (Προβολή & Φιλτράρισμα Χρηστών): [Μόνο για Admin] Παρέχει πρόσβαση στη λίστα όλων των εγγεγραμμένων χρηστών με δυνατότητα αναζήτησης και φιλτραρίσματος.
- Update Any User Profile (Επεξεργασία Προφίλ Τρίτων): [Μόνο για Admin] Επιτρέπει την τροποποίηση στοιχείων οποιουδήποτε χρήστη (π.χ. διόρθωση email, αλλαγή ρόλου).
- Delete Any User Account (Διαγραφή Λογαριασμού Τρίτων): [Μόνο για Admin] Επιτρέπει την οριστική διαγραφή οποιουδήποτε χρήστη από το σύστημα.
- Manage User Status (Διαχείριση Κατάστασης Χρήστη): [Μόνο για Admin] Επιτρέπει την ενεργοποίηση ή απενεργοποίηση (ban) ενός λογαριασμού χρήστη, ελέγχοντας έτσι το δικαίωμα πρόσβασής του στην πλατφόρμα χωρίς να απαιτείται η διαγραφή των δεδομένων του.

## **B. Ενότητα Παρακολούθησης Συστήματος (System Monitoring Module)**

- View System Statistics (Προβολή Στατιστικών Συστήματος): [Μόνο για Admin] Παρέχει μια συνοπτική εικόνα της κατάστασης της πλατφόρμας, εμφανίζοντας μετρικές όπως το πλήθος των εγγεγραμμένων χρηστών και των ενεργών DPPs. Η λειτουργία αυτή υποστηρίζεται από το στρώμα Monitoring & Observability (Prometheus/Grafana) που περιγράφηκε προηγουμένως στην αρχιτεκτονική της πλατφόρμας.

### Use Case Diagram - User & System Management



Εικόνα 6: Διάγραμμα περιπτώσεων χρήσης: Διαχείριση Χρηστών

### Ανάλυση Διαγράμματος Περιπτώσεων Χρήσης: Διαχείριση Προϊόντων (DPP Management)

Το δεύτερο διάγραμμα (Εικόνα 7) εστιάζει στον πυρήνα της πλατφόρμας, περιγράφοντας τις λειτουργίες που αφορούν τον κύκλο ζωής των Ψηφιακών Διαβατηρίων Προϊόντων.

#### 1. Δρώντες (Actors)

Η ιεραρχία των ρόλων παραμένει ίδια με το προηγούμενο διάγραμμα, με τα δικαιώματα να κλιμακώνονται από τον Viewer στον Administrator.

## 2. Περιπτώσεις Χρήσης (Use Cases)

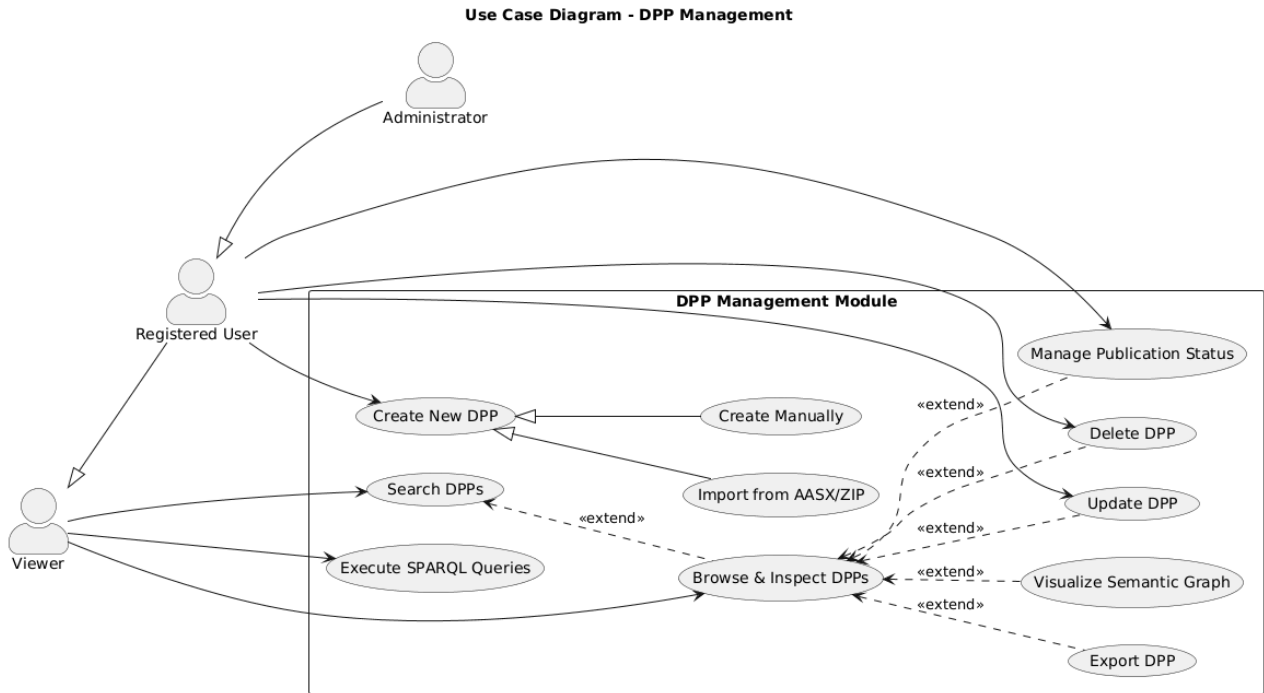
Οι λειτουργίες διακρίνονται σε αυτές που είναι δημόσια προσβάσιμες και σε αυτές που απαιτούν δικαιώματα ιδιοκτησίας ή διαχείρισης.

**A. Δημόσιες Λειτουργίες (Public Actions) - Διαθέσιμες σε όλους τους ρόλους (Viewer, Registered User, Admin), με πρόσβαση περιορισμένη στα Δημοσιευμένα (Published) προϊόντα για τους απλούς χρήστες, εκτός αν πρόκειται για τον Ιδιοκτήτη (Owner) του προϊόντος, ο οποίος διατηρεί πρόσβαση και στα δικά του Αδημοσίευτα (Draft) προϊόντα,**

- Search DPPs (Αναζήτηση DPPs): Επιτρέπει την εύρεση προϊόντων μέσω λέξεων-κλειδιών ή σύνθετων κριτηρίων.
- Browse & Inspect DPPs (Περιήγηση & Επισκόπηση DPPs): Η βασική λειτουργία προβολής της λίστας των προϊόντων και της επισκόπησης των βασικών τους στοιχείων μέσω του Πίνακα Ελέγχου (Dashboard).
  - Export DPP: Δυνατότητα εξαγωγής των δεδομένων σε μορφή JSON ή PDF.
  - Visualize Semantic Graph: Δυνατότητα γραφικής απεικόνισης του σημασιολογικού γράφου του προϊόντος.
- Execute SPARQL Queries (Εκτέλεση Ερωτημάτων SPARQL): Επιτρέπει την εκτέλεση σημασιολογικών ερωτημάτων στη βάση γνώσης.
  - Περιορισμός: Οι Viewers και Users μπορούν να αναζητήσουν μόνο σε δημοσιευμένους γράφους, ενώ οι Admins έχουν καθολική πρόσβαση.

**B. Λειτουργίες Διαχείρισης (User Actions) - Διαθέσιμες στους Εγγεγραμμένους Χρήστες (για τα δικά τους προϊόντα) και στους Διαχειριστές (για όλα τα προϊόντα).**

- Create New DPP (Δημιουργία Νέου DPP): Η διαδικασία καταχώρησης ενός νέου προϊόντος. Υλοποιείται μέσω δύο διακριτών μεθόδων (υπο-περιπτώσεις χρήσης):
  - Χειροκίνητη Δημιουργία (Create Manually): Εισαγωγή δεδομένων μέσω φόρμας.
  - Αυτόματη Δημιουργία (Import from AASX/ZIP): Εισαγωγή μέσω αρχείου AASX/ZIP.
- Update DPP (Ενημέρωση DPP): Τροποποίηση των στοιχείων ενός υφιστάμενου προϊόντος.
- Delete DPP (Διαγραφή DPP): Οριστική αφαίρεση του προϊόντος από το σύστημα.
- Manage Publication Status (Διαχείριση Κατάστασης Δημοσίευσης): Αλλαγή της κατάστασης ορατότητας του προϊόντος μεταξύ "Draft" (Πρόχειρο) και "Published" (Δημοσιευμένο).



Εικόνα 7: Διάγραμμα περιπτώσεων χρήσης: Διαχείριση Προϊόντων

#### 4.3.4 Διάγραμμα κλάσεων

Το διάγραμμα κλάσεων αποτυπώνει τις κυριότερες πληροφοριακές οντότητες του συστήματος, τα χαρακτηριστικά (attributes) και τις μεθόδους (operations) τους, καθώς και τις συσχετίσεις μεταξύ τους.

##### A. Σχεσιακό Μοντέλο (Relational Data Model)

Για το παρόν σύστημα, όπου ο στόχος είναι η διαχείριση και αναζήτηση Ψηφιακών Διαβατηρίων Προϊόντων (DPPs), οι κλάσεις μοντελοποιούν οντότητες ως εξής:

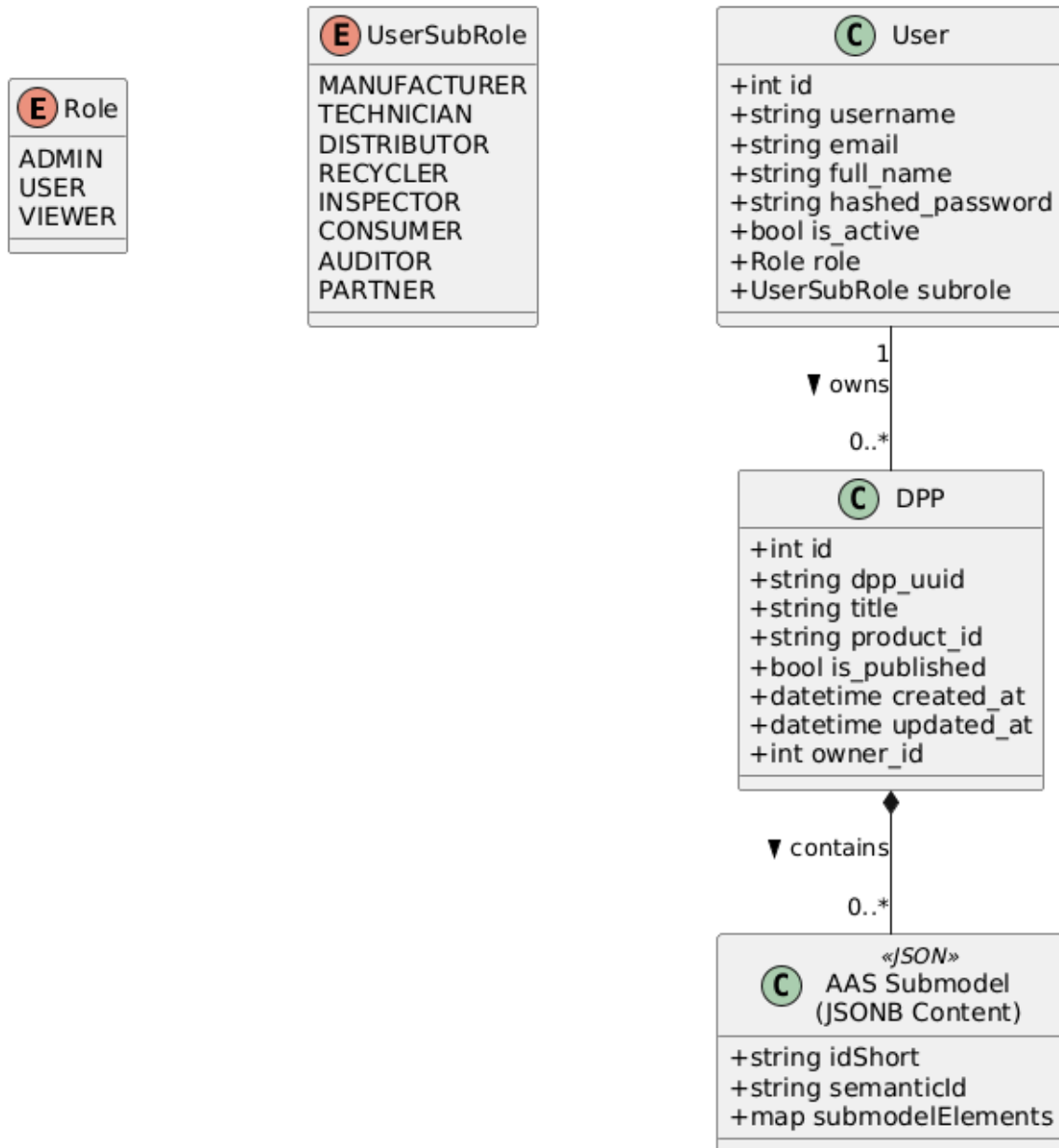
##### 1. Βασικές οντότητες (Entities)

- **Χρήστες (User):** Εγγεγραμμένοι χρήστες του συστήματος. Η κλάση αυτή περιλαμβάνει το ονοματεπώνυμο του χρήστη καθώς και τα απαραίτητα πεδία για την ταυτοποίηση (username, email, hashed\_password) και τον καθορισμό των δικαιωμάτων πρόσβασής του:
  - Role: Καθορίζει το επίπεδο πρόσβασης (Admin, User, Viewer).
  - Subrole: Προσδιορίζει την επαγγελματική ιδιότητα του χρήστη (π.χ. Manufacturer, Technician), παρέχοντας επιπλέον πληροφοριακό πλαίσιο.
  - is\_active: Boolean πεδίο που υποδηλώνει την κατάσταση του λογαριασμού. Επιτρέπει στον διαχειριστή να απενεργοποιεί (soft delete/ban) χρήστες χωρίς να διαγράφει τα δεδομένα τους, ενισχύοντας την ασφάλεια και τη διαχειρισσιμότητα.
- **DPP (Digital Product Passport):** Η κεντρική ψηφιακή αναπαράσταση ενός προϊόντος. Περιλαμβάνει τα εξής πεδία:
  - Ταυτοποίηση & Μεταδεδομένα: id (εσωτερικό κλειδί), dpp\_uuid (μοναδικό αναγνωριστικό διαβατηρίου), product\_id (αναγνωριστικό προϊόντος), title (τίτλος).

- Κατάσταση: `is_published` (Boolean πεδίο που καθορίζει την ορατότητα του διαβατηρίου, Draft vs Published).
- Διαχείριση: `created_at` (ημερομηνία δημιουργίας), `updated_at` (ημερομηνία τροποποίησης), `owner_id` (σύνδεση με τον χρήστη-ιδιοκτήτη).
- **AAS Submodel (JSONB Content):** Αναπαριστά τα επιμέρους τμήματα πληροφορίας που απαρτίζουν το διαβατήριο (π.χ. Τεχνικά Χαρακτηριστικά, Περιβαλλοντικό Αποτύπωμα). Αν και αποθηκεύονται ως ενιαίο JSON αντικείμενο στη βάση, στο μοντέλο αναπαριστώνται ως διακριτές οντότητες που συνθέτουν το DPP. Περιλαμβάνει πεδία όπως:
  - **idShort:** Σύντομο αναγνωριστικό (π.χ. "CarbonFootprint").
  - **semanticId:** Σημασιολογική αναφορά σε εξωτερικά πρότυπα (π.χ. ECLASS).
  - **submodelElements:** Τα πραγματικά δεδομένα (key-value pairs).

## 2. Σχέσεις (Relationships)

- **User - DPP (1-to-Many):** Ένας χρήστης μπορεί να δημιουργήσει και να διαχειρίζεται πολλαπλά DPPs, αλλά κάθε DPP ανήκει αποκλειστικά σε έναν ιδιοκτήτη. Η σχέση αυτή είναι θεμελιώδης για την εφαρμογή των κανόνων ελέγχου πρόσβασης.
- **DPP - AAS Submodel (Composition):** Η σχέση αυτή δεν είναι σχέση ξένου κλειδιού (Foreign Key) σε επίπεδο SQL, αλλά σχέση σύνθεσης (Composition) εντός του πεδίου JSONB. Το DPP "περιέχει" πολλαπλά Submodels (π.χ. Carbon Footprint, Technical Data), τα οποία αποτελούν αναπόσπαστο κομμάτι της δομής του.



Εικόνα 8: Διάγραμμα κλάσεων της πλατφόρμας διαχείρισης DPPs

## B. Σημασιολογικό Μοντέλο (Semantic Data Model / Ontology)

Το Σημασιολογικό Μοντέλο Δεδομένων αποτυπώνει τη δομή και τις σχέσεις των οντοτήτων όπως αυτές αναπαρίστανται στον Γράφο Γνώσης (Knowledge Graph) του συστήματος, χρησιμοποιώντας πρότυπα του Σημασιολογικού Ιστού (Semantic Web). Σε αντίθεση με το σχεσιακό μοντέλο που εστιάζει στην αποθήκευση πινάκων, το παρόν μοντέλο εστιάζει στη σημασία και τη διασύνδεση των δεδομένων μέσω τριάδων RDF (Subject-Predicate-Object).

Οι κύριες κλάσεις και οι μεταξύ τους σχέσεις περιγράφονται ως εξής:

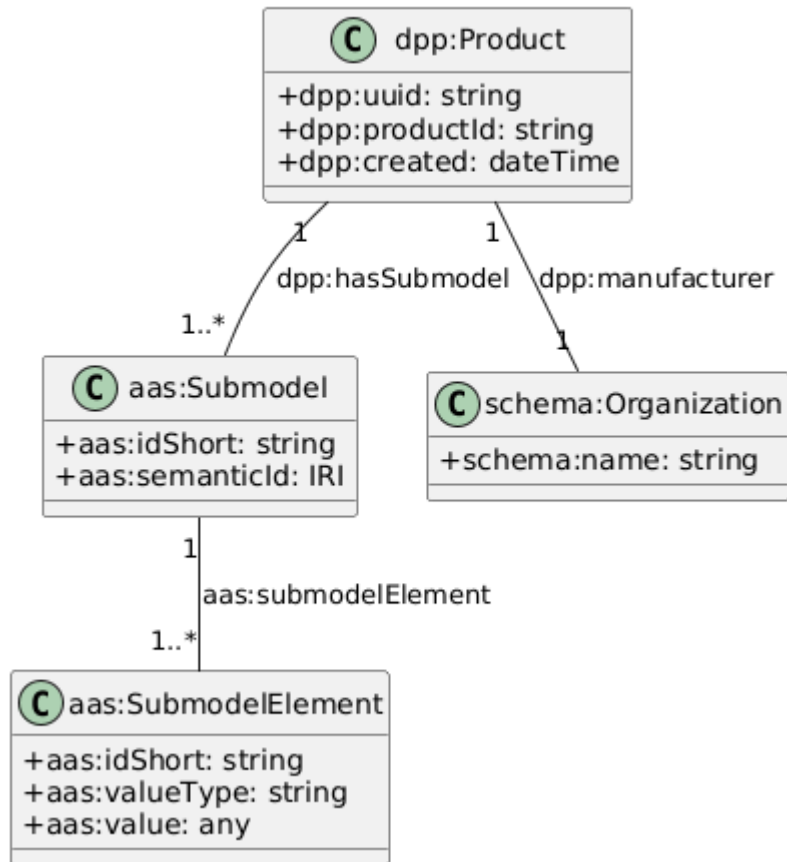
### 1. Βασικές Οντότητες (Entities):

- Κλάση *dpp:Product* (Προϊόν): Αποτελεί την κεντρική οντότητα του μοντέλου και αναπαριστά το ίδιο το Ψηφιακό Διαβατήριο Προϊόντος. Χρησιμοποιεί το πρόθεμα *dpp* (από το namespace της οντολογίας του συστήματος) και περιλαμβάνει βασικές ιδιότητες όπως:
  - *dpp:uuid*: Το μοναδικό αναγνωριστικό του προϊόντος (Universally Unique Identifier).
  - *dpp:title*: Ο τίτλος ή η ονομασία του προϊόντος.
  - *dpp:created*: Η ημερομηνία και ώρα δημιουργίας του διαβατηρίου.
- Κλάση *aas:Submodel* (Υπο-μοντέλο AAS): Αναπαριστά τα επιμέρους τμήματα πληροφορίας που απαρτίζουν το διαβατήριο, ακολουθώντας το πρότυπο Asset Administration Shell (AAS). Κάθε προϊόν μπορεί να συνδέεται με πολλαπλά υπο-μοντέλα (π.χ. Τεχνικά Χαρακτηριστικά, Περιβαλλοντικό Αποτύπωμα). Οι ιδιότητές του περιλαμβάνουν:
  - *aas:idShort*: Ένα σύντομο αναγνωριστικό όνομα για το υπο-μοντέλο (π.χ. "CarbonFootprint").
  - *aas:semanticId*: Ένα σημασιολογικό αναγνωριστικό που παραπέμπει σε εξωτερικά λεξικά ή πρότυπα (π.χ. ECLASS), διασφαλίζοντας την κοινή κατανόηση της σημασίας των δεδομένων.
- Κλάση *aas:SubmodelElement* (Στοιχείο Υπομοντέλου): Τα υπομοντέλα περιέχουν επιμέρους στοιχεία που φέρουν την πραγματική πληροφορία.
  - *aas:value*: Η τιμή του στοιχείου (π.χ., "100kg", "High Efficiency").
  - *aas:idShort*: Το όνομα του στοιχείου.
- Κλάση *schema:Organization* (Κατασκευαστής): Αναπαριστά τον οργανισμό ή την εταιρεία που κατασκεύασε το προϊόν. Χρησιμοποιεί το ευρέως διαδεδομένο λεξιλόγιο *schema.org* για μέγιστη διαλειτουργικότητα.
  - *schema:name*: Η επωνυμία του κατασκευαστή.

## 2. Συσχετίσεις (Relationships):

- *dpp:hasSubmodel*: Συνδέει την οντότητα *Product* με τα αντίστοιχα *AAS\_Submodel*. Η σχέση αυτή δηλώνει ότι ένα προϊόν "αποτελείται από" ή "περιέχει" συγκεκριμένα υπο-μοντέλα πληροφορίας.
- *aas:submodelElement*: Συνδέει την οντότητα *AAS\_Submodel* με τα *SubmodelElement* που περιέχει, δομώντας ιεραρχικά την πληροφορία.
- *dpp:manufacturer*: Συνδέει την οντότητα *Product* με την οντότητα *Manufacturer*, προσδιορίζοντας ποιος οργανισμός είναι υπεύθυνος για το προϊόν.

Η χρήση αυτών των σημασιολογικών συνδέσμων επιτρέπει την εκτέλεση σύνθετων ερωτημάτων SPARQL, όπως "Βρες όλα τα προϊόντα που κατασκευάστηκαν από την εταιρεία X και έχουν υπο-μοντέλο για το Ανθρακικό Αποτύπωμα".

**Simplified Semantic Model for DPP**

Εικόνα 9: Διάγραμμα κλάσεων του σημασιολογικού μοντέλου της πλατφόρμας DPP

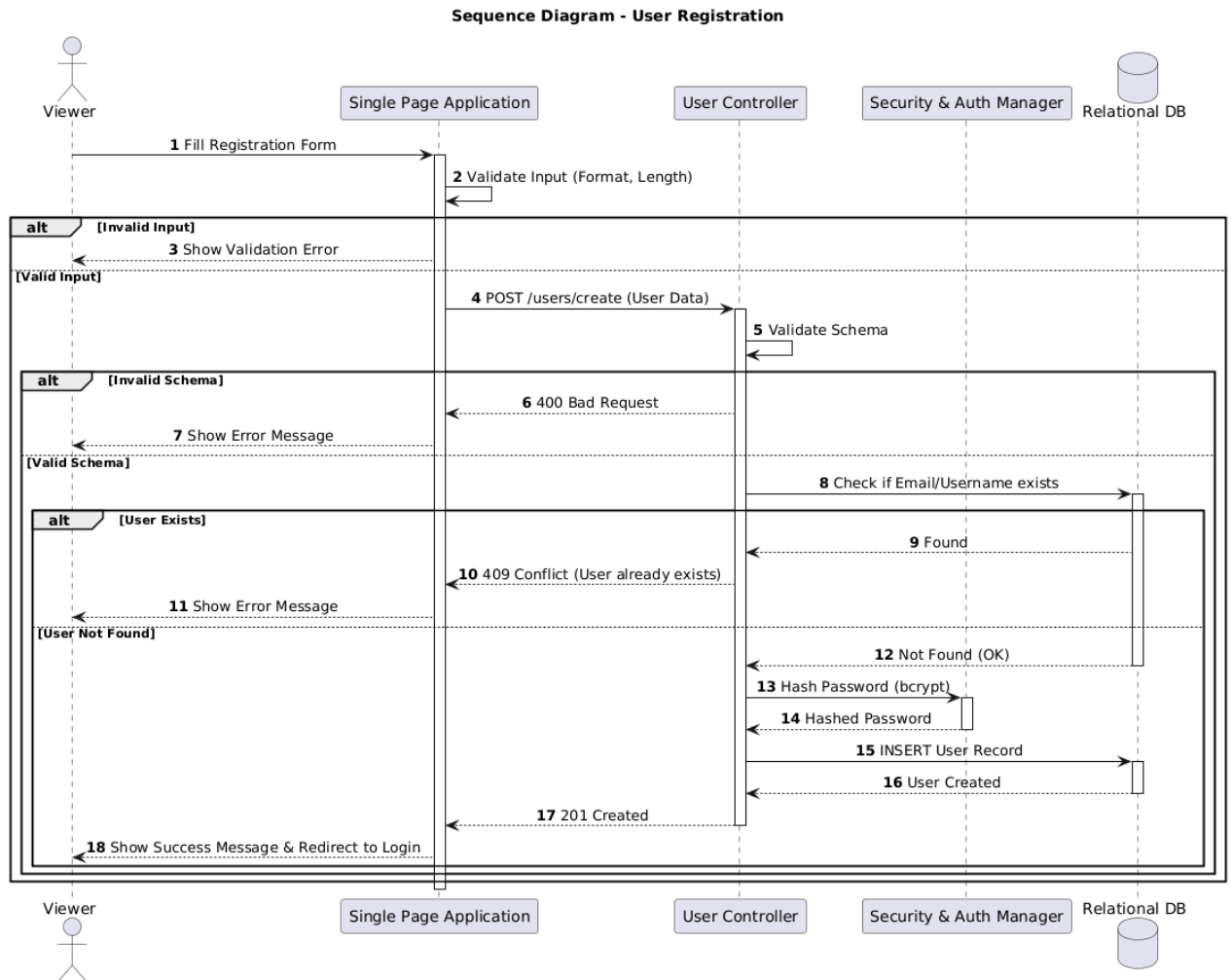
**4.3.5 Διαγράμματα ακολουθίας**

Το διάγραμμα ακολουθίας (sequence diagram) χρησιμοποιείται για την αποτύπωση της χρονικής ροής μηνυμάτων που ανταλλάσσονται μεταξύ διαφορετικών οντοτήτων του συστήματος. Μέσω αυτών των διαγραμμάτων, αναδεικνύεται η δυναμική συμπεριφορά της πλατφόρμας και ο τρόπος με τον οποίο τα επιμέρους υποσυστήματα (Frontend, Backend, Βάσεις Δεδομένων) συνεργάζονται για την εκτέλεση μιας λειτουργίας.

Στην παρούσα ενότητα, παραθέτουμε τέσσερα (4) αντιπροσωπευτικά διαγράμματα ακολουθίας που καλύπτουν τις κρίσιμότερες λειτουργίες του συστήματος.

**1. Εγγραφή Επισκέπτη (Viewer Registration)**

Το διάγραμμα ακολουθίας για την εγγραφή επισκέπτη περιγράφει τα βήματα που απαιτούνται για τη δημιουργία ενός νέου λογαριασμού στο σύστημα. Η διαδικασία έχει σχεδιαστεί με έμφαση στην ασφάλεια και την ακεραιότητα των δεδομένων.



Εικόνα 10: Διάγραμμα ακολουθίας εγγραφής

### Ανάλυση ροής εγγραφής

Η διαδικασία ξεκινά όταν ο Επισκέπτης (Viewer) συμπληρώσει τη φόρμα εγγραφής στη διεπαφή χρήστη (Βήμα 1), όπου η εφαρμογή (SPA) πραγματοποιεί έναν άμεσο έλεγχο εγκυρότητας της μορφής των δεδομένων (Βήμα 2).

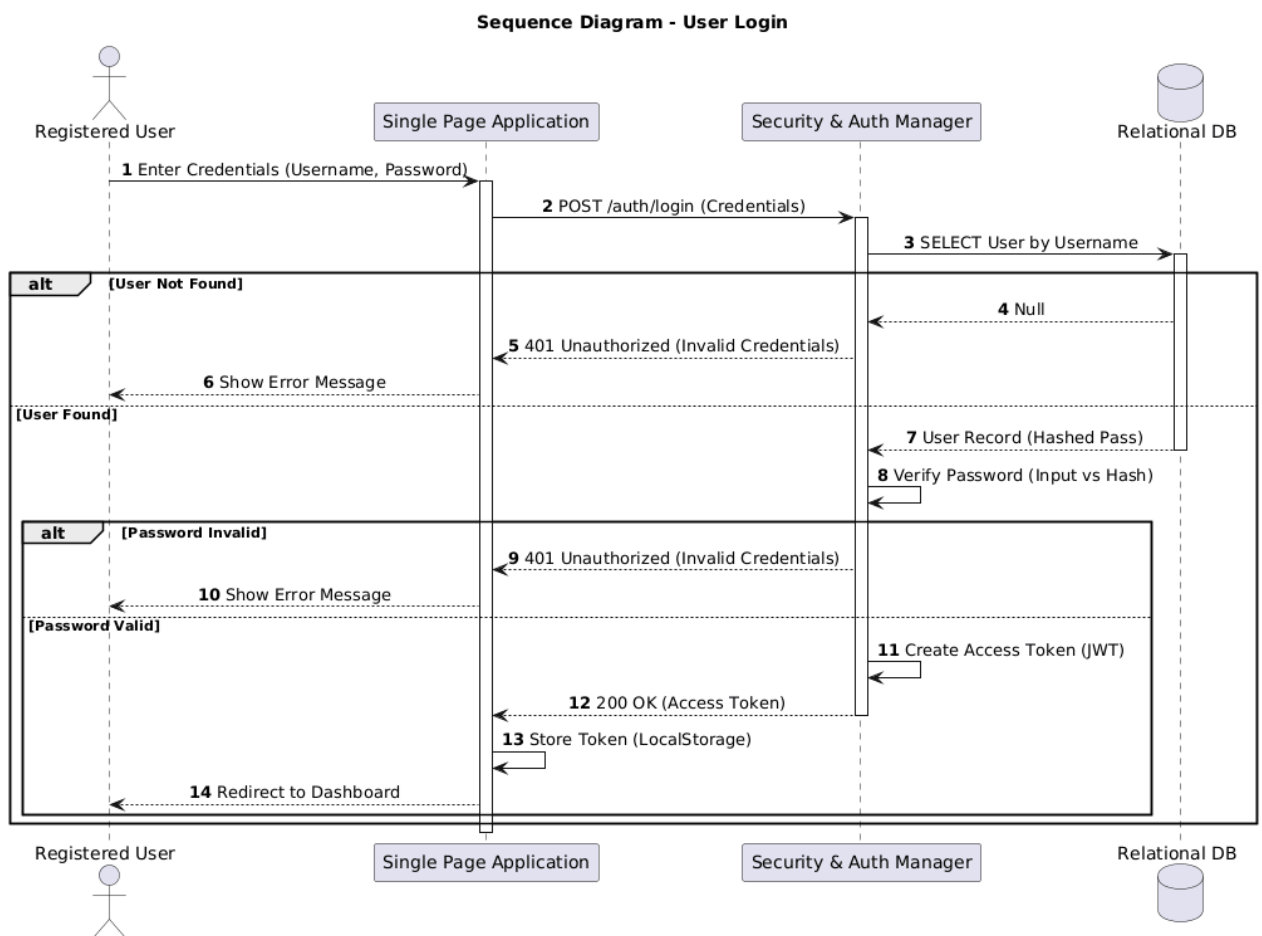
- **Υποβολή και Επικύρωση Backend (Βήματα 4-7):** Εφόσον τα δεδομένα είναι έγκυρα, αποστέλλονται στον **User Controller** (Βήμα 4), ο οποίος διενεργεί έλεγχο πληρότητας (Schema Validation) στο Βήμα 5. Σε περίπτωση ασυμβατότητας, η διαδικασία διακόπτεται με την επιστροφή αντίστοιχου μηνύματος σφάλματος στην εφαρμογή (Βήματα 6-7).
- **Έλεγχος Μοναδικότητας (Βήματα 8-12):** Ο Ελεγκτής επικοινωνεί με τη **Σχεσιακή Βάση Δεδομένων (Relational DB)** για να βεβαιώσει ότι τα στοιχεία (email/username) δεν ανήκουν σε υπάρχοντα χρήστη (Βήμα 8). Αν τα στοιχεία βρεθούν, επιστρέφεται σφάλμα (Βήματα 9-11), ενώ αν η εγγραφή δεν υπάρχει (Βήμα 12), η διαδικασία συνεχίζεται κανονικά.
- **Ασφάλεια και Καταχώρηση (Βήματα 13-16):** Για τη διασφάλιση των δεδομένων, ο κωδικός πρόσβασης αποστέλλεται στον **Security & Auth Manager** για κρυπτογράφηση (Βήμα 13). Μετά την παραλαβή του κρυπτογραφημένου κωδικού (Βήμα 14), ο Ελεγκτής

πραγματοποιεί την εισαγωγή του νέου χρήστη στη βάση δεδομένων (Βήμα 15) και λαμβάνει επιβεβαίωση της δημιουργίας (Βήμα 16).

- **Ολοκλήρωση και Ενημέρωση (Βήματα 17-18):** Η διαδικασία ολοκληρώνεται με την επιστροφή της απόκρισης επιτυχίας (201 Created) στην εφαρμογή (Βήμα 17), η οποία προβάλλει το τελικό μήνυμα στον χρήστη και τον κατευθύνει στη σελίδα σύνδεσης (Βήμα 18).

## 2. Αυθεντικοποίηση Χρήστη (User Login)

Το διάγραμμα ακολουθίας για την αυθεντικοποίηση περιγράφει τη διαδικασία εισόδου του χρήστη (Registered User) στο σύστημα και την έκδοση του διακριτικού ασφαλείας (JWT). Η ροή αυτή διαχειρίζεται απευθείας από το υποσύστημα ασφαλείας, παρακάμπτοντας τον User Controller για λόγους αποδοτικότητας και διαχωρισμού αρμοδιοτήτων.



Εικόνα 11: Διάγραμμα ακολουθίας σύνδεσης

### Ανάλυση ροής σύνδεσης

Η διαδικασία ξεκινά όταν ο Εγγεγραμμένος Χρήστης εισάγει τα διαπιστευτήριά του (Username και Password) στη διεπαφή χρήστη (Βήμα 1), η οποία στη συνέχεια αποστέλλει το αίτημα σύνδεσης στον Security & Auth Manager (Βήμα 2).

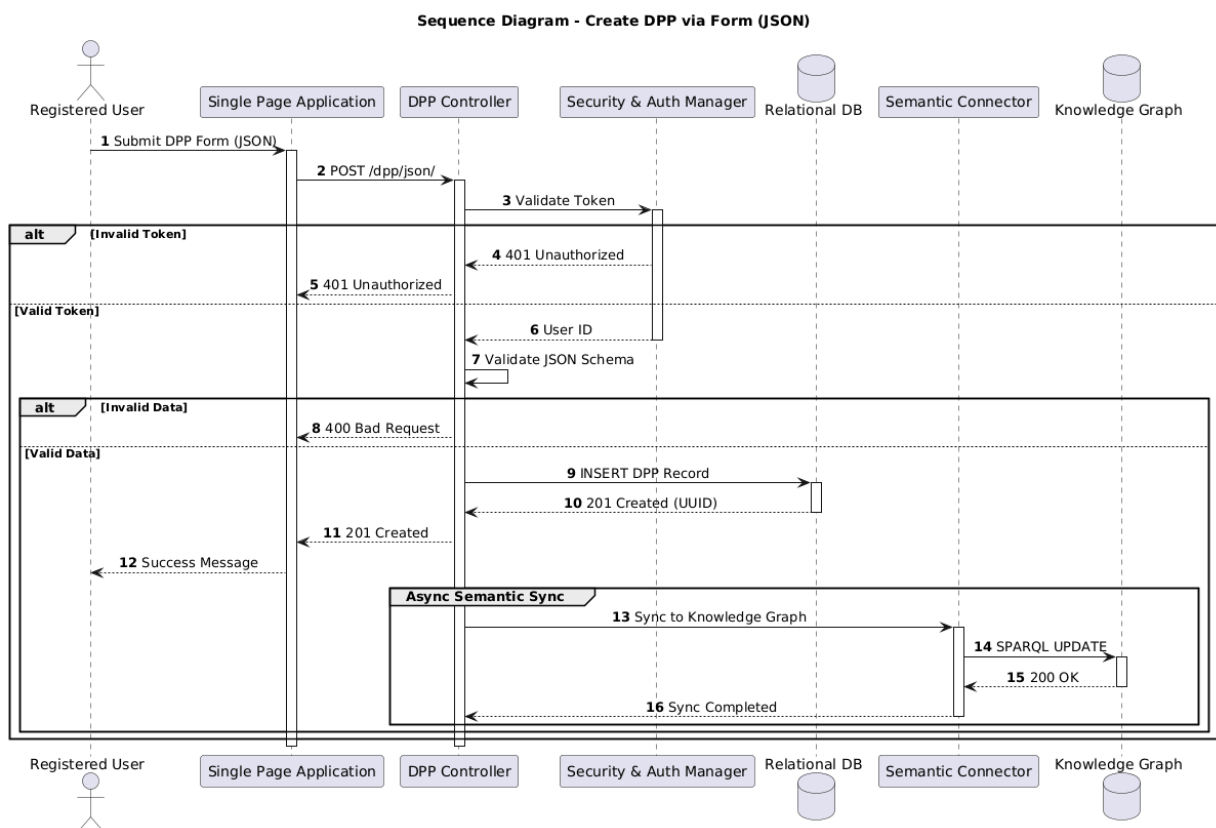
- **Ανάκτηση και Έλεγχος Χρήστη (Βήματα 3-7):** Ο Auth Manager επικοινωνεί με τη Relational DB για την αναζήτηση του χρήστη βάσει του ονόματος χρήστη (Βήμα 3). Εάν ο

χρήστης δεν βρεθεί (Βήμα 4), επιστρέφεται σφάλμα 401 Unauthorized και η εφαρμογή ενημερώνει τον χρήστη (Βήματα 5-6). Στην περίπτωση που ο χρήστης εντοπιστεί, η βάση επιστρέφει την εγγραφή του χρήστη που περιλαμβάνει τον κρυπτογραφημένο κωδικό (Βήμα 7).

- **Επαλήθευση και Έκδοση Token (Βήματα 8-12):** Ο Auth Manager εκτελεί τη διαδικασία επαλήθευσης συγκρίνοντας τον κωδικό εισόδου με την αποθηκευμένη τιμή (Βήμα 8). Εάν ο κωδικός είναι λανθασμένος, επιστρέφεται σφάλμα 401 (Βήματα 9-10). Εφόσον η επαλήθευση είναι επιτυχής, το σύστημα δημιουργεί ένα ψηφιακά υπογεγραμμένο Token πρόσβασης (JWT) στο Βήμα 11 και το αποστέλλει στην εφαρμογή (Βήμα 12).
- **Ολοκλήρωση και Πλοήγηση (Βήματα 13-14):** Η εφαρμογή αποθηκεύει το token τοπικά για τη διατήρηση της συνεδρίας (Βήμα 13) και ανακατευθύνει αυτόματα τον χρήστη στον Πίνακα Ελέγχου (Dashboard), επιτρέποντάς του την πρόσβαση στις προστατευμένες λειτουργίες της πλατφόρμας (Βήμα 14).

### 3. Χειροκίνητη δημιουργία νέου DPP μέσω Φόρμας (Create DPP via Form)

Το παρακάτω διάγραμμα περιγράφει τη ροή δημιουργίας ενός νέου Ψηφιακού Διαβατηρίου Προϊόντος με χειροκίνητο τρόπο μέσω φόρμας. Ιδιαίτερη έμφαση δίνεται στον διπλό μηχανισμό αποθήκευσης, όπου τα δεδομένα καταχωρούνται αρχικά στη σχεσιακή βάση (PostgreSQL) και στη συνέχεια συγχρονίζονται ασύγχρονα με τον Γράφο Γνώσης (Fuseki) μέσω του Semantic Connector.



Εικόνα 12: Διάγραμμα ακολουθίας δημιουργίας νέου DPP

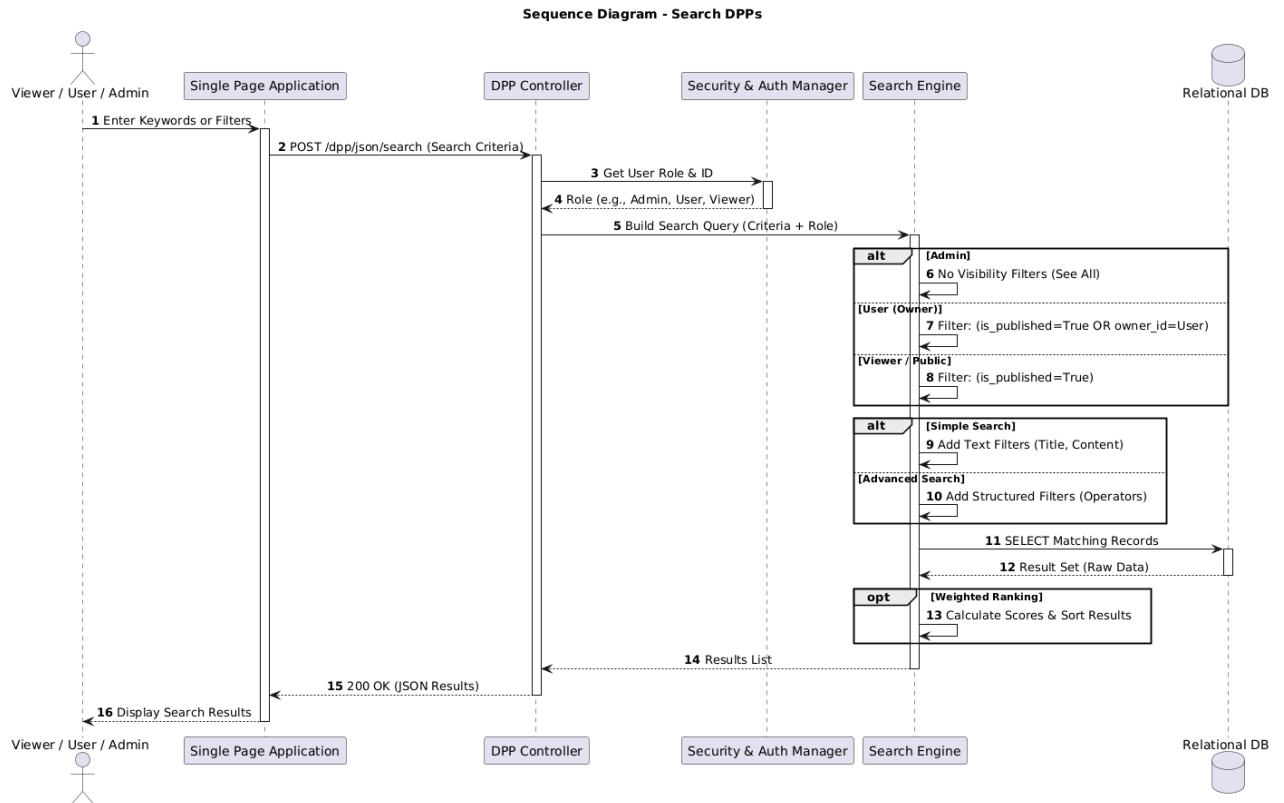
### Ανάλυση ροής δημιουργίας DPP

Η διαδικασία ξεκινά όταν ο Εγγεγραμμένος Χρήστης υποβάλει τη φόρμα με τα στοιχεία του νέου προϊόντος σε μορφή JSON μέσω της διεπαφής χρήστη (Βήμα 1), η οποία προωθεί το αίτημα στον DPP Controller (Βήμα 2).

- **Αυθεντικοποίηση & Επικύρωση (Βήματα 3-8):** Ο Ελεγκτής ζητά την επικύρωση του Token από τον Security & Auth Manager (Βήμα 3). Σε περίπτωση μη έγκυρου Token, επιστρέφεται σφάλμα 401 Unauthorized και η διαδικασία τερματίζεται (Βήματα 4-5). Αν η ταυτοποίηση είναι επιτυχής και επιστραφεί το User ID (Βήμα 6), ο Ελεγκτής προχωρά στον έλεγχο του σχήματος των δεδομένων JSON (Βήμα 7). Αν τα δεδομένα είναι λανθασμένα, επιστρέφεται σφάλμα 400 Bad Request (Βήμα 8).
- **Αποθήκευση και Επιβεβαίωση (Βήματα 9-12):** Εφόσον τα δεδομένα είναι έγκυρα, ο Ελεγκτής πραγματοποιεί την εισαγωγή της εγγραφής στη Relational DB (Βήμα 9), η οποία επιστρέφει το μοναδικό αναγνωριστικό UUID (Βήμα 10). Στη συνέχεια, αποστέλλεται η απόκριση 201 Created στην εφαρμογή (Βήμα 11) και προβάλλεται το μήνυμα επιτυχίας στον χρήστη (Βήμα 12).
- **Ασύγχρονος Σημασιολογικός Συγχρονισμός (Βήματα 13-15):** Μετά την επιβεβαίωση στον χρήστη, εκκινεί στο παρασκήνιο η διαδικασία ενημέρωσης του γράφου. Ο Semantic Connector αναλαμβάνει τον συγχρονισμό (Βήμα 13), εκτελώντας SPARQL UPDATE στον Knowledge Graph (Βήμα 14) και λαμβάνοντας επιβεβαίωση ολοκλήρωσης από τη βάση (Βήμα 15).

#### 4. Αναζήτηση DPP (Search DPPs)

Το παρακάτω διάγραμμα δείχνει πώς λειτουργεί η αναζήτηση (Simple & Advanced) και πώς επιστρέφονται τα αποτελέσματα από τη σχεσιακή βάση δεδομένων.



Εικόνα 13: Διάγραμμα ακολουθίας αναζήτησης DPP

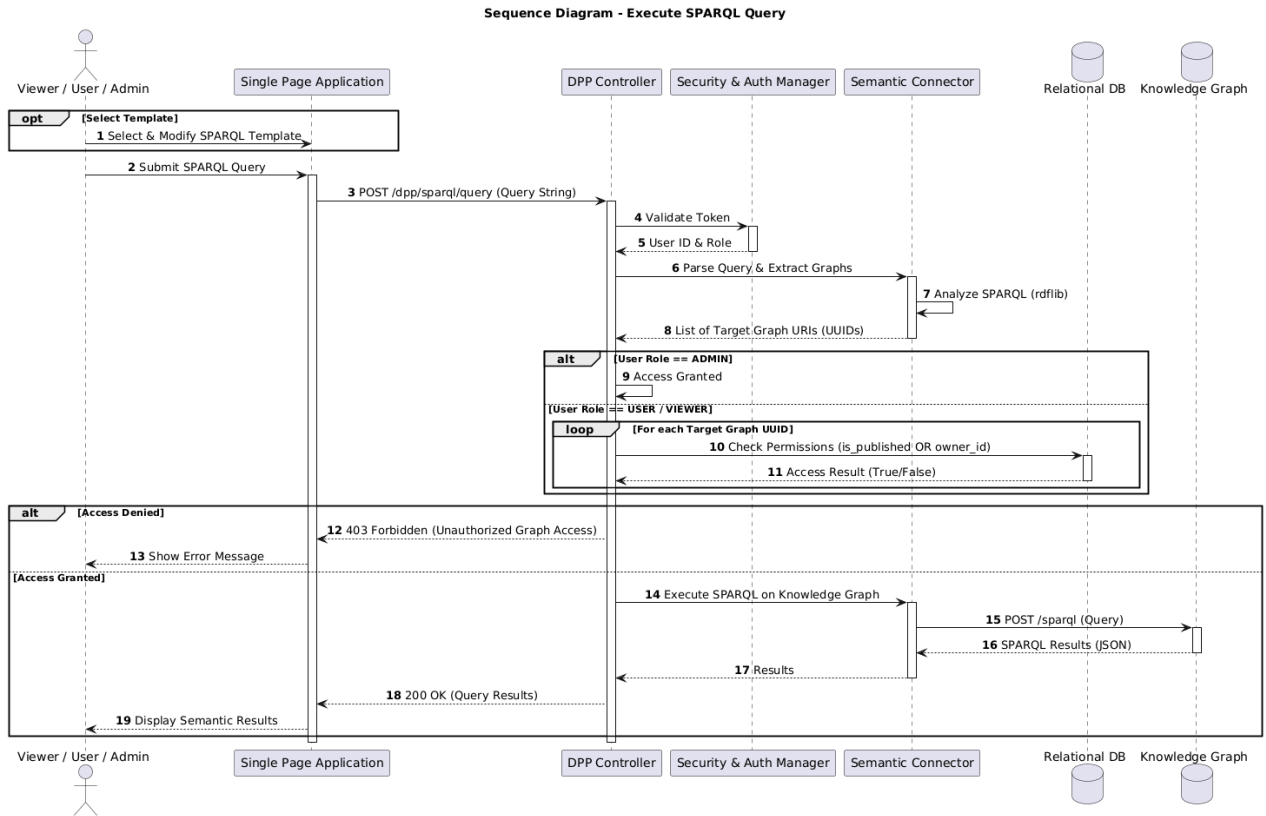
### Ανάλυση ροής αναζήτησης DPP

Η διαδικασία ξεκινά με την εισαγωγή κριτηρίων από τον χρήστη και την αποστολή του αιτήματος στον ελεγκτή (Βήματα 1-2).

- **Έλεγχος Ρόλου (Βήματα 3-4):** Πριν την εκτέλεση της αναζήτησης, ο DPP Controller επικοινωνεί με τον Security & Auth Manager για να ανακτήσει τον ρόλο και το ID του χρήστη. Αυτό είναι κρίσιμο βήμα για τον καθορισμό της ορατότητας.
- **Δυναμική Κατασκευή Ερωτήματος (Βήματα 5-10):** Η Search Engine κατασκευάζει το SQL ερώτημα συνδυάζοντας τα κριτήρια αναζήτησης του χρήστη με φίλτρα ασφαλείας (Βήματα 6-8) και φίλτρα τύπου αναζήτησης (Βήματα 9-10):
  - Admin: Βλέπει όλα τα DPPs.
  - User: Βλέπει όλα τα Δημοσιευμένα (Published) καθώς και τα δικά του Πρόχειρα (Drafts).
  - Viewer: Βλέπει μόνο τα Δημοσιευμένα DPPs.
- **Εκτέλεση & Κατάταξη (Βήματα 11-13):** Το ερώτημα εκτελείται στη βάση δεδομένων (Βήματα 11-12). Προαιρετικά, αν έχει ενεργοποιηθεί η σταθμισμένη αναζήτηση (κατάταξη), τα αποτελέσματα βαθμολογούνται και ταξινομούνται (Βήμα 13).
- **Ολοκλήρωση (Βήματα 14-16):** Η λίστα των αποτελεσμάτων επιστρέφεται στον ελεγκτή, ο οποίος στέλνει την τελική απόκριση στην εφαρμογή για την προβολή τους στον χρήστη.

## 5. Εκτέλεση SPARQL ερωτήματος (Execute SPARQL Query).

Το διάγραμμα αυτό περιγράφει τη διαδικασία εκτέλεσης σημασιολογικών ερωτημάτων, δίνοντας έμφαση στον μηχανισμό ελέγχου πρόσβασης σε επίπεδο γράφου (Graph-Level Security).



Εικόνα 14: Διάγραμμα ακολουθίας εκτέλεσης SPARQL ερωτήματος

### Ανάλυση ροής εκτέλεσης SPARQL ερωτήματος

Το διάγραμμα περιγράφει τη διαδικασία υποβολής και εκτέλεσης σημασιολογικών ερωτημάτων στον Γράφο Γνώσης, εστιάζοντας στους μηχανισμούς ασφαλείας.

- **Προετοιμασία (Προαιρετικό - Βήμα 1):** Ο χρήστης δύναται να επιλέξει ένα προκαθορισμένο πρότυπο (Template) ερωτήματος από τη βιβλιοθήκη του συστήματος και να το τροποποιήσει πριν την υποβολή, διευκολύνοντας τη σύνταξη πολύπλοκων SPARQL queries.
- **Αυθεντικοποίηση (Βήματα 2-5):** Με την υποβολή του ερωτήματος (Βήμα 2) και την αποστολή του αιτήματος στον ελεγκτή (Βήμα 3), το σύστημα επικυρώνει την ταυτότητα και τον ρόλο του χρήστη μέσω του Security & Auth Manager (Βήματα 4-5).
- **Ανάλυση Ερωτήματος (Βήματα 6-8):** Ο DPP Controller δεν εκτελεί τυφλά το ερώτημα. Αντίθετα, το προωθεί στον Semantic Connector (Βήμα 6), ο οποίος αναλύει τη σύνταξη SPARQL (Βήμα 7) και εξάγει τα Target Graph URIs, δηλαδή τα UUIDs των συγκεκριμένων DPPs που στοχεύει το ερώτημα και τα αποστέλλει στον DPP Controller (Βήμα 8).

- **Έλεγχος Πρόσβασης (Βήματα 9-11):**
  - Αν ο χρήστης είναι Admin, η πρόσβαση επιτρέπεται άμεσα σε όλα τα γραφήματα (Βήμα 9).
  - Αν είναι απλός User ή Viewer, ο Controller ελέγχει στη Σχεσιακή Βάση (Relational DB) αν ο χρήστης έχει δικαίωμα πρόσβασης για κάθε στοχευμένο γράφο (Βήματα 10-11), ελέγχοντας αν το DPP είναι δημοσιευμένο ή αν ο χρήστης είναι ο ιδιοκτήτης.
- **Εκτέλεση ή Απόρριψη (Βήματα 12-19):**
  - **Access Denied (Βήματα 12-13):** Αν έστω και ένας από τους στοχευμένους γράφους δεν είναι προσβάσιμος, η διαδικασία διακόπτεται και επιστρέφεται σφάλμα 403 Forbidden.
  - **Access Granted (Βήματα 14-19):** Μόνο εφόσον επιβεβαιωθούν τα δικαιώματα, το ερώτημα αποστέλλεται στον Γράφο Γνώσης προς εκτέλεση (Βήματα 14-15). Τα αποτελέσματα επιστρέφονται και προβάλλονται στον χρήστη (Βήματα 16-19).

#### 4.3.6 Διάγραμμα δραστηριοτήτων

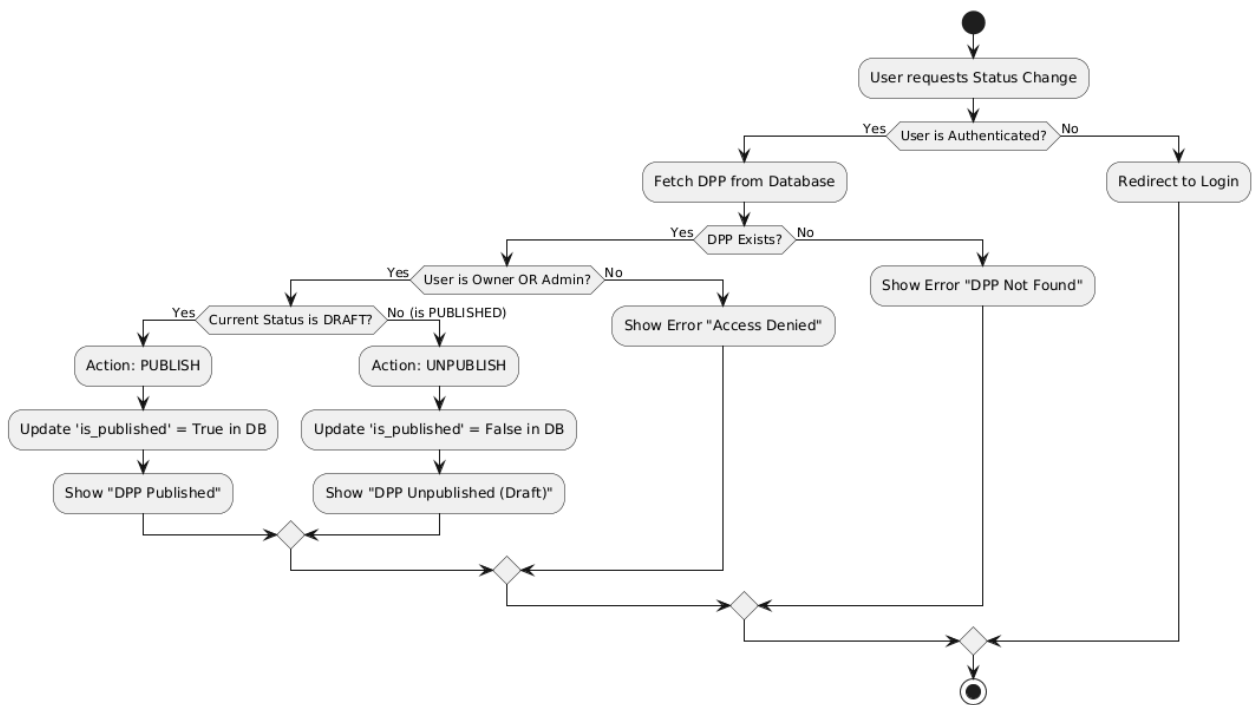
Το διάγραμμα δραστηριοτήτων (activity diagram) χρησιμοποιείται για τη μοντελοποίηση της ροής εργασιών (workflow) και της επιχειρησιακής λογικής του συστήματος. Επίσης, αποτυπώνει τη σειρά των ενεργειών, τις συνθήκες λήψης αποφάσεων (decision points) και τις παράλληλες διεργασίες που απαιτούνται για την ολοκλήρωση μιας σύνθετης λειτουργίας / περίπτωσης χρήσης του συστήματος.

Στην παρούσα ενότητα, αναλύουμε τέσσερις (4) διαδικασίες μέσω διαγραμμάτων δραστηριοτήτων που αντιστοιχούν σε 4 βασικές λειτουργίες του συστήματός μας, αντίστοιχα:

##### 1. Διαχείριση κατάστασης δημοσίευσης DPP (Manage DPP Status) - Publish/Unpublish

Το συγκεκριμένο διάγραμμα περιγράφει τη λογική που διέπει την αλλαγή κατάστασης ενός Ψηφιακού Διαβατηρίου. Καλύπτει την περίπτωση χρήσης "Manage Publication Status", δείχνοντας πώς το σύστημα αποφασίζει αν θα επιτρέψει τη δημοσίευση (Publish) ή την απόσυρση (Unpublish) ενός προϊόντος, ελέγχοντας την τρέχουσα κατάστασή του και τα δικαιώματα του χρήστη.

Activity Diagram - Manage DPP Publication Status



Εικόνα 15: Διάγραμμα δραστηριοτήτων διαχείρισης κατάστασης δημοσίευσης DPP

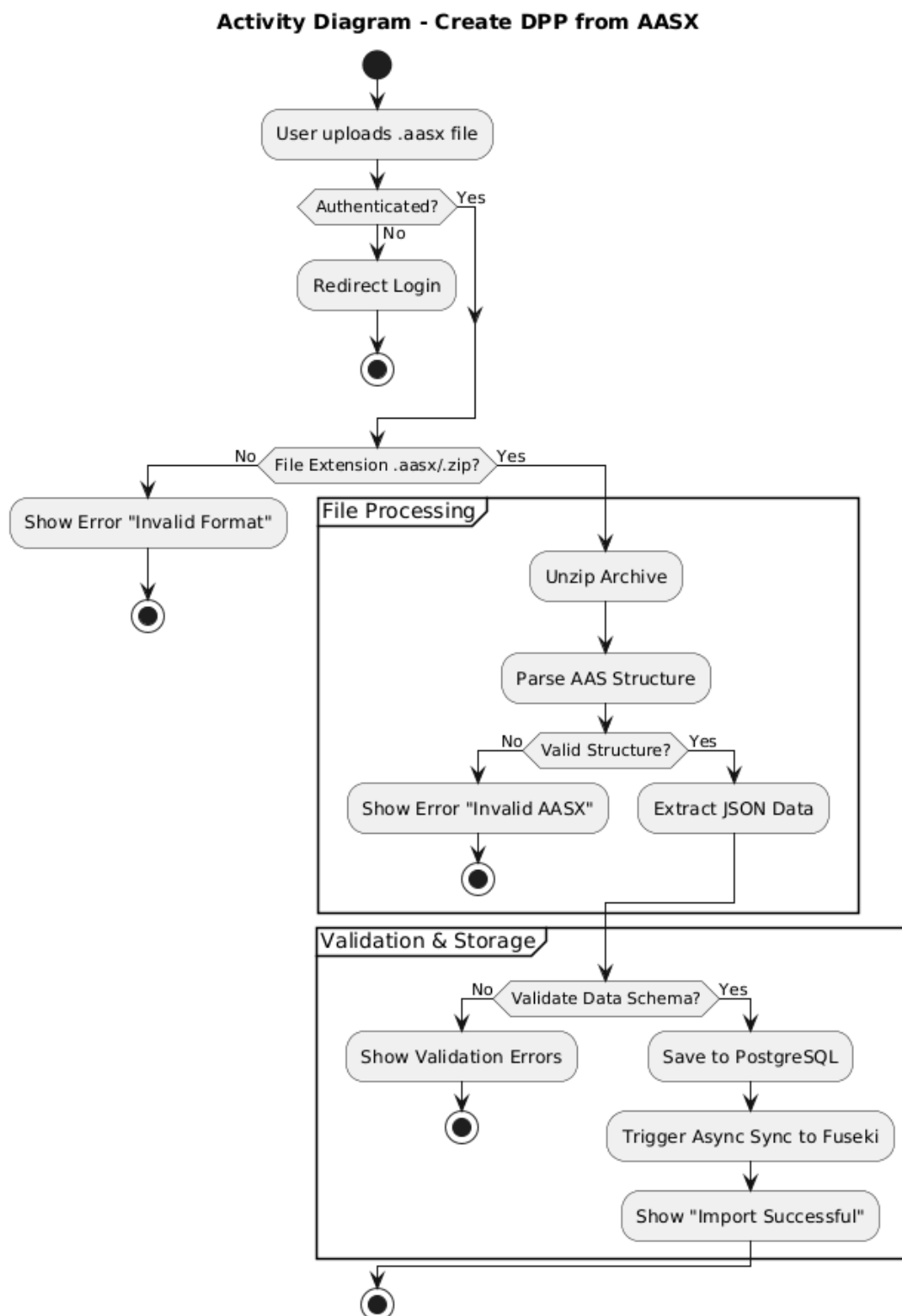
## Ανάλυση ροής διαχείρισης κατάστασης δημοσίευσης DPP

Το διάγραμμα δραστηριοτήτων απεικονίζει τη λογική λήψης αποφάσεων για την αλλαγή της κατάστασης ορατότητας ενός Ψηφιακού Διαβατηρίου. Η διαδικασία ξεκινά με την επαλήθευση της ταυτότητας του χρήστη. Εάν ο χρήστης δεν είναι συνδεδεμένος, ανακατευθύνεται άμεσα στη σελίδα εισόδου (Login). Εφόσον ο χρήστης είναι αυθεντικοποιημένος, το σύστημα επιχειρεί να ανακτήσει το DPP από τη βάση δεδομένων. Αν το προϊόν δεν βρεθεί, εμφανίζεται μήνυμα σφάλματος ("DPP Not Found"). Στη συνέχεια, ελέγχεται αν ο χρήστης έχει την απαραίτητη εξουσιοδότηση. Η ενέργεια επιτρέπεται μόνο αν ο χρήστης είναι ο Ιδιοκτήτης (Owner) του συγκεκριμένου DPP ή αν διαθέτει ρόλο Διαχειριστή (Admin). Σε αντίθετη περίπτωση, η πρόσβαση απορρίπτεται ("Access Denied"). Τέλος, το σύστημα ελέγχει την τρέχουσα κατάσταση του DPP. Αν το προϊόν είναι "Πρόχειρο", ενημερώνεται το πεδίο `is_published` σε `True`, καθιστώντας το δημόσια διαθέσιμο. Αν το προϊόν είναι ήδη "Δημοσιευμένο", η κατάσταση αλλάζει σε "Πρόχειρο" (Unpublish - το πεδίο `is_published` γίνεται `false`), αποσύροντάς το από τη δημόσια προβολή. Η ενημέρωση πραγματοποιείται στη Σχεσιακή Βάση Δεδομένων, η οποία λειτουργεί ως η κεντρική αρχή για τον έλεγχο της ορατότητας των δεδομένων.

## 2. Δημιουργία DPP μέσω Αρχείου (Create DPP via File Import)

Το παρόν διάγραμμα αποτυπώνει τη διαδικασία αυτοματοποιημένης δημιουργίας ενός DPP μέσω της μεταφόρτωσης αρχείων βιομηχανικών προτύπων. Καλύπτει την περίπτωση χρήσης "Import from AASX/ZIP", περιγράφοντας τα διαδοχικά στάδια επεξεργασίας: από τον έλεγχο

εγκυρότητας του αρχείου και την αποσυμπίεση, μέχρι την εξόρυξη δεδομένων και την τελική αποθήκευση στη βάση δεδομένων.



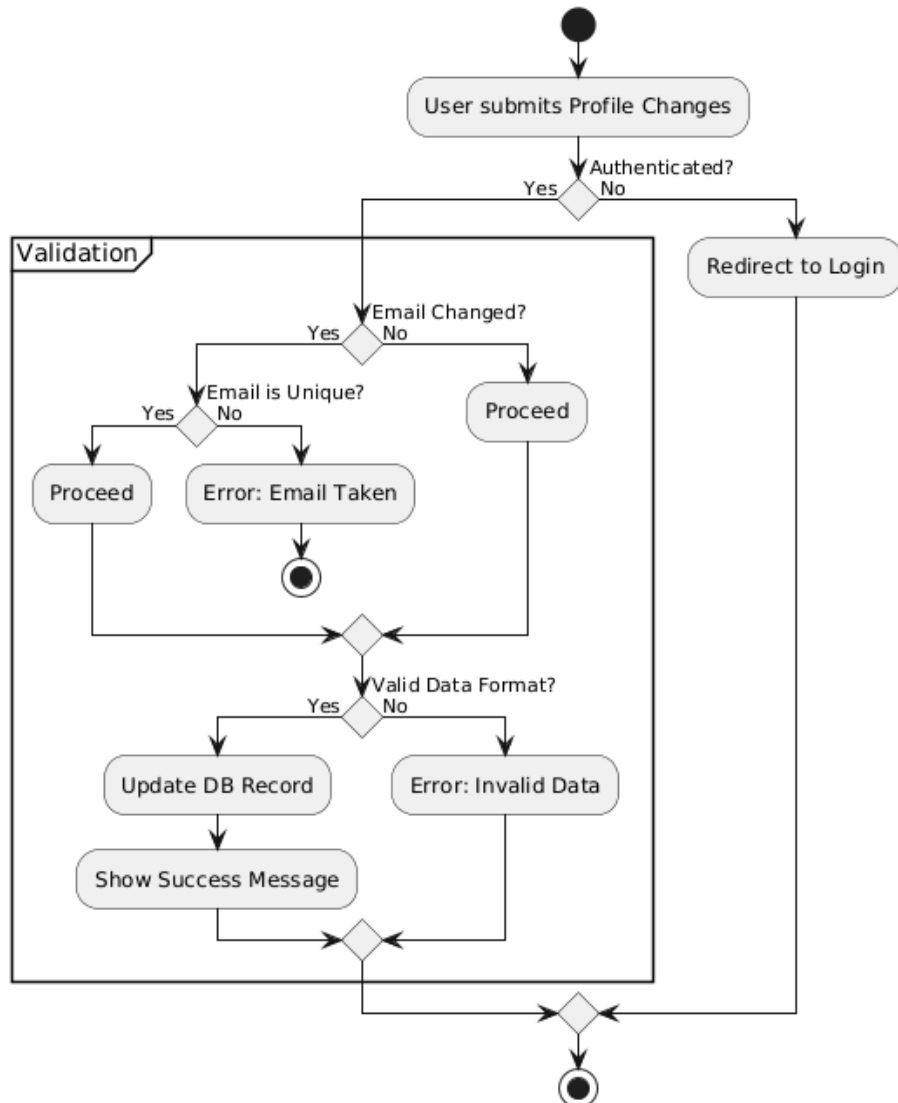
Εικόνα 16: Διάγραμμα δραστηριοτήτων εισαγωγής αρχείου AASX

### **Ανάλυση ροής εισαγωγής αρχείου AASX**

Το διάγραμμα δραστηριοτήτων περιγράφει την εναλλακτική διαδικασία μαζικής εισαγωγής δεδομένων μέσω αρχείων προτύπου AASX. Η διαδικασία ξεκινά με τη μεταφόρτωση του αρχείου από τον χρήστη. Το σύστημα ελέγχει αρχικά αν ο χρήστης είναι αυθεντικοποιημένος και στη συνέχεια επαληθεύει την επέκταση του αρχείου (πρέπει να είναι .aasx ή .zip). Εφόσον το αρχείο είναι έγκυρο, ο File Processor αναλαμβάνει την αποσυμπίεσή του και την ανάλυση της εσωτερικής δομής AAS. Ελέγχεται η ύπαρξη των απαραίτητων αρχείων (Manifest/JSON) και εξάγονται τα δεδομένα του προϊόντος και των υπο-μοντέλων του. Τα εξαγόμενα δεδομένα υποβάλλονται σε έλεγχο εγκυρότητας (Schema Validation). Αν είναι συμβατά με το μοντέλο δεδομένων του συστήματος, αποθηκεύονται στη Σχεσιακή Βάση (PostgreSQL). Με την επιτυχή αποθήκευση, ενεργοποιείται ο ασύγχρονος μηχανισμός συγχρονισμού με τον Γράφο Γνώσης (Fuseki) (όπως εμφανίστηκε και στο διάγραμμα ακολουθίας για την χειροκίνητη δημιουργία DPP) και εμφανίζεται μήνυμα επιτυχίας στον χρήστη.

### **3. Ενημέρωση προφίλ χρήστη (Update User Profile).**

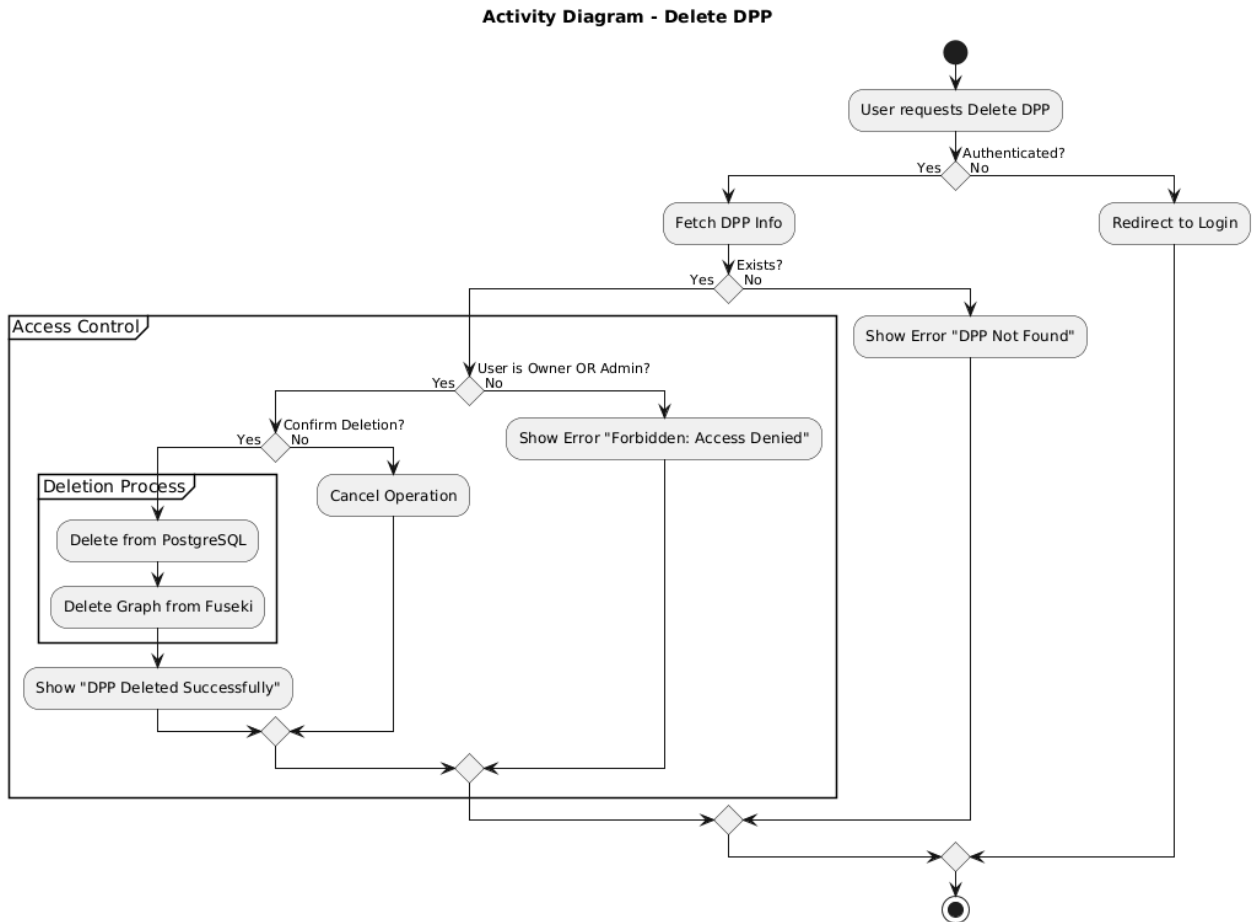
Το διάγραμμα δραστηριοτήτων εστιάζει στη διαδικασία ενημέρωσης των στοιχείων του χρήστη (π.χ. Όνομα, Email), εξαιρώντας την αλλαγή κωδικού πρόσβασης, η οποία αποτελεί ξεχωριστή περίπτωση χρήσης.

**Activity Diagram - Update User Profile***Εικόνα 17: Διάγραμμα δραστηριοτήτων ενημέρωσης προφίλ χρήστη***Ανάλυση ροής ενημέρωσης προφίλ χρήστη**

Η διαδικασία προχωρά μόνο εφόσον ο χρήστης είναι συνδεδεμένος. Το σύστημα ανιχνεύει αν έχει τροποποιηθεί το Email. Σε αυτή την περίπτωση, εκτελείται έλεγχος μοναδικότητας στη βάση δεδομένων. Αν το νέο email χρησιμοποιείται ήδη, η ενέργεια ακυρώνεται. Εφόσον τα δεδομένα είναι έγκυρα (σωστή μορφή, μοναδικό email), ενημερώνεται η εγγραφή του χρήστη στη βάση δεδομένων.

**4. Διαγραφή DPP (Delete DPP)**

Αυτό το διάγραμμα είναι σημαντικό γιατί δείχνει τη διπλή διαγραφή της πληροφορίας για ένα DPP (από PostgreSQL και Fuseki) και τους αντίστοιχους ελέγχους δικαιωμάτων που επιτελούνται.



Εικόνα 18: Διάγραμμα δραστηριοτήτων διαγραφής DPP

### Ανάλυση ροής διαγραφής DPP

Η διαδικασία διαγραφής ενός Ψηφιακού Διαβατηρίου περιλαμβάνει αυστηρούς ελέγχους ασφαλείας και επιβεβαίωσης για την αποφυγή ακούσιας απώλειας δεδομένων. Το σύστημα οφείλει να ελέγχει αν ο χρήστης είναι ταυτοποιημένος και αν το υπό διαγραφή προϊόν είναι διαθέσιμο στη βάση δεδομένων. Επιβεβαιώνεται ότι ο χρήστης διαθέτει τα απαραίτητα δικαιώματα, δηλαδή είναι ο Ιδιοκτήτης του συγκεκριμένου DPP ή Διαχειριστής του συστήματος. Πριν από την οριστική διαγραφή, απαιτείται η ρητή επιβεβαίωση της ενέργειας από τον χρήστη μέσω ειδικής διεπαφής. Εφόσον εγκριθεί η ενέργεια, εκτελείται η διαδικασία διαγραφής σε δύο επίπεδα: αφαιρείται η εγγραφή από τη σχεσιακή βάση δεδομένων και, παράλληλα, διαγράφεται ο αντίστοιχος ονοματισμένος γράφος (Named Graph) από τον Γράφο Γνώσης (Knowledge Graph), εξασφαλίζοντας ότι δεν παραμένουν "ορφανά" δεδομένα στο σύστημα.

## 4.4 Υλοποίηση συστήματος

Η υλοποίηση της πλατφόρμας για τη διαχείριση των Digital Product Passports (DPPs) βασίστηκε σε μια καθαρή αρχιτεκτονική διαχωρισμού μεταξύ του frontend και του backend, με στόχο την ανεξαρτησία, την επεκτασιμότητα και την ευκολία συντήρησης της εφαρμογής. Επιλέχθηκε το επίπεδο 3 για την ωριμότητα του RESTful API της εφαρμογής/συστήματος, ενώ εφαρμόστηκαν βέλτιστες πρακτικές ασφαλείας, αρθρωτός (modular) προγραμματισμός και τεκμηρίωση του API με χρήση εργαλείων όπως το Swagger. Η προσέγγιση αυτή διασφαλίζει ότι κάθε τμήμα του συστήματος μπορεί να εξελίσσεται αυτόνομα, ενώ η χρήση σύγχρονων εργαλείων αυτοματοποιεί τις διαδικασίες ελέγχου και διάθεσης της πληροφορίας.

### 4.4.1 Τεχνολογίες και περιβάλλον ανάπτυξης

#### Backend

Για την ανάπτυξη του Backend επιλέχθηκε η γλώσσα προγραμματισμού Python 3.11 σε συνδυασμό με το πλαίσιο ιστού (web framework) FastAPI<sup>10</sup>. Η επιλογή αυτή βασίστηκε στα εξής βασικά χαρακτηριστικά του πλαισίου αυτού: στην υψηλή ταχύτητα απόκρισης των αναπτυσσόμενων RESTful υπηρεσιών, την εγγενή υποστήριξη ασύγχρονου προγραμματισμού και την αυτόματη παραγωγή τεκμηρίωσης OpenAPI για τις αναπτυσσόμενες υπηρεσίες. Η διαχείριση των εξαρτήσεων και του περιβάλλοντος ανάπτυξης πραγματοποιείται μέσω του εργαλείου Poetry<sup>11</sup>, διασφαλίζοντας τη σταθερότητα των εκδόσεων των χρησιμοποιούμενων βιβλιοθηκών.

Για την υλοποίηση των επιμέρους λειτουργιών αξιοποιήθηκαν εξειδικευμένα εργαλεία: η επεξεργασία αρχείων AASX και η ασαφής ταύτιση πεδίων υποστηρίζονται από τις βιβλιοθήκες xmldict<sup>12</sup> και diffliib, η μετατροπή δεδομένων σε RDF γίνεται μέσω της rdflib<sup>13</sup>, ενώ η εξαγωγή αναφορών σε PDF υλοποιείται με τη χρήση της ReportLab<sup>14</sup>. Για την ασφάλεια του συστήματος, υλοποιήθηκε μηχανισμός ταυτοποίησης με χρήση JSON Web Tokens (JWT) (μέσω της python-jose) και εφαρμόστηκε έλεγχος πρόσβασης βάσει ρόλων (Role-Based Access Control - RBAC), ο οποίος υλοποιήθηκε εγγενώς μέσω του μηχανισμού Dependency Injection του FastAPI, διαχωρίζοντας τα προνόμια μεταξύ επισκεπτών, χρηστών και διαχειριστών.

Η διαχείριση των δεδομένων πραγματοποιείται μέσω ενός υβριδικού μοντέλου αποθήκευσης. Χρησιμοποιείται η σχεσιακή βάση δεδομένων PostgreSQL, η οποία συνδέεται με το Backend μέσω του ORM SQLAlchemy<sup>15</sup> (και του οδηγού asyncpg<sup>16</sup>), παρέχοντας ένα υψηλού επιπέδου επίπεδο αφαίρεσης (abstraction layer) για την εκτέλεση ερωτημάτων. Η βάση αυτή διαχειρίζεται τόσο τα αυστηρά δομημένα δεδομένα (όπως οι χρήστες και τα βασικά μεταδεδομένα των προϊόντων) όσο και τα ημι-δομημένα δεδομένα περιεχομένου. Συγκεκριμένα, αξιοποιείται ο τύπος δεδομένων JSONB για την αποθήκευση των αναλυτικών πληροφοριών των DPPs, προσφέροντας την απαραίτητη ευελιξία για την ενσωμάτωση των υπο-μοντέλων AAS χωρίς την ανάγκη αλλαγής του σχήματος της βάσης.

---

<sup>10</sup> FastAPI: <https://fastapi.tiangolo.com/>

<sup>11</sup> Poetry: <https://python-poetry.org/>

<sup>12</sup> xmldict: <https://github.com/martinblech/xmldict>

<sup>13</sup> rdflib: <https://rdflib.readthedocs.io/>

<sup>14</sup> ReportLab: <https://www.reportlab.com/>

<sup>15</sup> SQLAlchemy: <https://www.sqlalchemy.org/>

<sup>16</sup> asyncpg: <https://github.com/MagicStack/asyncpg>

Παράλληλα, το σύστημα ενσωματώνει τη σημασιολογική βάση/αποθετήριο γνώσης Apache Jena Fuseki. Αυτή λειτουργεί ως το SPARQL Endpoint για τη διαχείριση των σημασιολογικών δεδομένων (RDF), τα οποία προκύπτουν μέσω αυτόματης μετατροπής και συγχρονισμού των ημι-δομημένων δεδομένων που αποθηκεύονται πρωτογενώς στην PostgreSQL. Με τον τρόπο αυτό, καθίσταται δυνατή η εκτέλεση πολύπλοκων SPARQL ερωτημάτων πάνω στον Γράφο Γνώσης.

Τέλος, για την παρακολούθηση της υγείας του συστήματος, έχει ενσωματωθεί μηχανισμός καταγραφής συμβάντων μέσω της βιβλιοθήκης Loguru<sup>17</sup>, καθώς και σύστημα συλλογής μετρικών με τον Prometheus FastAPI Instrumentator<sup>18</sup>, τα οποία οπτικοποιούνται σε πραγματικό χρόνο μέσω Grafana<sup>19</sup>.

### Διεπαφή Προγραμματισμού Εφαρμογών (REST API)

Η πλατφόρμα προσφέρει μια σειρά από REST endpoints για την επικοινωνία του frontend με το backend, τα οποία τεκμηριώνονται αυτόματα μέσω του εργαλείου Swagger UI (Εικόνα 19). Η διαδραστική αυτή τεκμηρίωση επιτρέπει στους προγραμματιστές να δοκιμάζουν τις κλήσεις σε πραγματικό χρόνο, διασφαλίζοντας την ορθή ανταλλαγή δεδομένων.

Τα διαθέσιμα endpoints οργανώνονται στις εξής θεματικές ενότητες:

- *Authentication*: Λειτουργίες αυθεντικοποίησης χρηστών (Login) και έκδοσης διακριτικών ασφαλείας (JWT Access Tokens). Περιλαμβάνει επίσης endpoint για την ανάκτηση των στοιχείων του τρέχοντος συνδεδεμένου χρήστη (/me).
- *Users*: Ολοκληρωμένη διαχείριση χρηστών, συμπεριλαμβανομένης της εγγραφής (Create), της προβολής λίστας χρηστών (με σελιδοποίηση και φίλτρα), της ενημέρωσης προφίλ και της διαχείρισης ρόλων (Admin/User/Viewer).
- *DPP JSON*: Υλοποίηση του πλήρους κύκλου ζωής (CRUD) των Ψηφιακών Διαβατηρίων. Περιλαμβάνει τη δημιουργία, ανάγνωση, ενημέρωση και διαγραφή DPPs, καθώς και προηγμένες λειτουργίες αναζήτησης (Simple & Advanced Search) και διαχείρισης κατάστασης δημοσίευσης (Publish/Unpublish).
- *DPP Files*: Υπηρεσίες διαχείρισης αρχείων, με έμφαση στη μεταφόρτωση (Upload) και αυτόματη εξαγωγή δεδομένων από αρχεία προτύπου AAS (.aasx, .zip, .json), διευκολύνοντας τη μαζική εισαγωγή δεδομένων.
- *DPP SPARQL*: Διεπαφή για την εκτέλεση σημασιολογικών ερωτημάτων (SPARQL SELECT) απευθείας στον γράφο γνώσης (Fuseki). Παρέχει επίσης λειτουργικότητα για την ανάκτηση της δομής του γράφου (Nodes/Edges) προς οπτικοποίηση.
- *DPP Export*: Μηχανισμός εξαγωγής των δεδομένων του διαβατηρίου σε αναγνώσιμη μορφή PDF (για ανθρώπινη χρήση) και σε δομημένη μορφή JSON (για μηχανική επεξεργασία).
- *System*: Λειτουργίες ελέγχου καλής λειτουργίας (Health Check) για την παρακολούθηση της διαθεσιμότητας του Backend και της σύνδεσης με τη βάση δεδομένων.

Για την αναλυτική περιγραφή κάθε endpoint (διαδρομές, παράμετροι εισόδου, κωδικοί κατάστασης HTTP και δομή αποκρίσεων), ανατρέξτε στο Παράρτημα I: Τεκμηρίωση REST API.

<sup>17</sup> Loguru: <https://github.com/Delgan/loguru>

<sup>18</sup> Prometheus FastAPI Instrumentator: <https://github.com/trallnag/prometheus-fastapi-instrumentator>

<sup>19</sup> Grafana: <https://grafana.com/>

**DPP Management Platform** 0.1.0 OAS 3.1

/openapi.json

Digital Product Passport Management Backend

Authorize

Authentication

Users

DPP JSON

DPP Files

DPP SPARQL

DPP Export

System

**Εικόνα 19: Συνοπτική παρουσίαση των API Endpoints και των θεματικών ενοτήτων μέσω του εργαλείου Swagger UI.**

## Frontend

Στην πλευρά του Frontend χρησιμοποιήθηκε η βιβλιοθήκη React.js<sup>20</sup>, η οποία προσφέρει ένα ευέλικτο περιβάλλον για την ανάπτυξη της διεπαφής χρήστη. Μέσω αυτής, παρέχεται η δυνατότητα στους χρήστες να αλληλεπιδρούν με το σημασιολογικό υπόβαθρο της πλατφόρμας, εκτελώντας ερωτήματα SPARQL για την άντληση εξειδικευμένης πληροφορίας.

Η οπτικοποίηση των σημασιολογικών δεδομένων στο frontend πραγματοποιείται μέσω δυναμικών γράφων, αξιοποιώντας τη βιβλιοθήκη react-force-graph-2d<sup>21</sup>. Το σύστημα μετατρέπει τις τριάδες (triples) που ανακτώνται από το SPARQL endpoint σε ένα διαδραστικό δίκτυο κόμβων και ακμών, επιτρέποντας στον χρήστη να εξερευνήσει οπτικά τις συσχετίσεις του προϊόντος με τους προμηθευτές και τα υλικά του.

### 4.4.2 Διαχείριση αρχείων και εξαγωγή δεδομένων

Το σύστημα υποστηρίζει την εισαγωγή δεδομένων μέσω αρχείων **.aasx** και **.zip**. Κατά τη διαδικασία μεταφόρτωσης, το backend εκτελεί δυναμική αποσυμπίεση των πακέτων και εντοπίζει τα αρχεία προτύπου AAS (XML/JSON). Στην συνέχεια, πραγματοποιείται έλεγχος εγκυρότητας της δομής και αυτόματη εξαγωγή των πεδίων. Τα εξαγόμενα δεδομένα, τα οποία μπορεί να αντιστοιχούν σε πολλαπλά υπο-μοντέλα (Submodels) (π.χ. Τεχνικά Χαρακτηριστικά, Περιβαλλοντικό Αποτύπωμα), ενοποιούνται και αποθηκεύονται σε ένα ενιαίο πεδίο JSONB στη βάση δεδομένων, διατηρώντας την ιεραρχική τους δομή.

Επιπλέον, παρέχεται η δυνατότητα εξαγωγής των δεδομένων του διαβατηρίου σε μορφή PDF για χρήση από τον τελικό καταναλωτή, καθώς και σε JSON/RDF για διαλειτουργικότητα με άλλα πληροφοριακά συστήματα.

<sup>20</sup> React: <https://react.dev/>

<sup>21</sup> react-force-graph-2d: <https://github.com/vasturiano/react-force-graph>

#### 4.4.3 Υποδομή και οργάνωση πηγαίου κώδικα

Για τη διασφάλιση της ομαλής λειτουργίας και της εύκολης εγκατάστασης, η εφαρμογή ακολουθεί αρχιτεκτονική μικρο-υπηρεσιών (microservices logic), όπου το Backend και το Frontend αναπτύσσονται σε ξεχωριστά αποθετήρια κώδικα (Repositories), διατηρώντας πλήρη ανεξαρτησία. Η ενοποίηση και η κοινή εκτέλεση επιτυγχάνεται μέσω της "πακετοποίησης" ολόκληρης της εφαρμογής με τη χρήση του Docker.

Το αρχείο `docker-compose.yaml` λειτουργεί ως ο κεντρικός ενορχηστρωτής, ορίζοντας τις υπηρεσίες για το Backend, το Frontend και το Fuseki, και δημιουργώντας ένα απομονωμένο εσωτερικό δίκτυο (dpp-net) για την ασφαλή επικοινωνία τους. Αν και στην παρούσα φάση επιλέχθηκε η χρήση ενός ενιαίου δικτύου για λόγους διαχειριστικής απλότητας, η περαιτέρω τμηματοποίηση (network segmentation) μέσω της δημιουργίας ξεχωριστών δικτύων ανά ζεύγος υπηρεσιών αποτελεί προτεραιότητα για μελλοντική βελτίωση, προκειμένου να ενισχυθεί περαιτέρω το επίπεδο ασφάλειας του συστήματος.

Η δομή του κώδικα ακολουθεί τα εξής πρότυπα οργάνωσης:

- Στο Backend, τα αρχεία χωρίζονται σε routers για τα endpoints, models για τη βάση δεδομένων, schemas για την επικύρωση δεδομένων (Pydantic) και services για την επιχειρησιακή λογική.
- Αντίστοιχα, στο Frontend, ο κώδικας οργανώνεται σε σελίδες (pages), επαναχρησιμοποιήσιμα UI components (που ενσωματώνονται σε σελίδες) και υπηρεσίες επικοινωνίας (api services) για την αλληλεπίδραση με το backend.

Ο πηγαίος κώδικας του backend είναι οργανωμένος σε διακριτές ενότητες:

dpp-backend/

|                 |                                                         |
|-----------------|---------------------------------------------------------|
| └─ app/         | # Κύριος φάκελος εφαρμογής                              |
| └─ configs/     | # Ρυθμίσεις συστήματος και ορισμός ρόλων                |
| └─ db/          | # Ρυθμίσεις σύνδεσης με PostgreSQL (Session/Engine)     |
| └─ models/      | # Ορισμός οντοτήτων SQLAlchemy (User, DPP, SubModel)    |
| └─ routers/     | # Διαδρομές (Endpoints) API ανά λειτουργική ενότητα     |
| └─ schemas/     | # Μοντέλα Pydantic για επικύρωση δεδομένων (Schemas)    |
| └─ utils/       | # Βοηθητικά εργαλεία (RDF converter, SPARQL client)     |
| └─ constants.py | # Σταθερές τιμές εφαρμογής (π.χ. FIELD_MAPPING)         |
| └─ main.py      | # Σημείο εισόδου (Entry point) και αρχικοποίηση FastAPI |
| └─ alembic/     | # Φάκελος διαχείρισης εκδόσεων βάσης (Migrations)       |
| └─ logs/        | # Καταγραφή συμβάντων λειτουργίας (Runtime Logs)        |

|                       |                                                     |
|-----------------------|-----------------------------------------------------|
| — monitoring/         | # Ρυθμίσεις εποπτείας (Prometheus/Grafana configs)  |
| — scripts/            | # Σενάρια αυτοματοποίησης και συντήρησης            |
| — tests/              | # Σουίτα δοκιμών (Unit & Integration Tests)         |
| — uploads/            | # Προσωρινή αποθήκευση αρχείων AASX/JSON            |
| — alembic.ini         | # Ρυθμίσεις εργαλείου Alembic για τη βάση δεδομένων |
| — docker-compose.yaml | # Ορχήστρωση υπηρεσιών σε containers                |
| — pyproject.toml      | # Διαχείριση εξαρτήσεων και ρυθμίσεις του Poetry    |
| — README.md           | # Τεκμηρίωση εγκατάστασης και χρήσης του κώδικα     |

Αντίστοιχα, το frontend οργανώνεται ως εξής:

dpp-frontend/

|                    |                                                         |
|--------------------|---------------------------------------------------------|
| — src/             | # Κύριος κώδικας εφαρμογής (Source)                     |
| — assets/          | # Στατικά αρχεία (εικόνες, λογότυπα)                    |
| — components/      | # Επαναχρησιμοποιήσιμα δομικά στοιχεία (UI Components)  |
| — config/          | # Ρυθμίσεις εφαρμογής και σταθερές                      |
| — pages/           | # Κεντρικές σελίδες/οθόνες (Dashboard, New DPP, κ.λπ.)  |
| — services/        | # Επικοινωνία με το Backend API (api.js)                |
| — styles/          | # Αρχεία μορφοποίησης και CSS/SCSS                      |
| — tests/           | # Μονάδες ελέγχου (Unit & Component Tests)              |
| — App.jsx          | # Κεντρικό Component και ορισμός δρομολόγησης (Routing) |
| — i18n.js          | # Ρύθμιση Διεθνοποίησης (Ελληνικά & Αγγλικά)            |
| — main.jsx         | # Σημείο εκκίνησης (Entry Point) της React              |
| — public/          | # Δημόσια στατικά αρχεία (favicon, robots.txt)          |
| — .env             | # Μεταβλητές περιβάλλοντος (π.χ. URL του API)           |
| — Dockerfile       | # Ορισμός περιβάλλοντος για containerization            |
| — eslint.config.js | # Κανόνες ελέγχου ποιότητας κώδικα (Linting)            |
| — index.html       | # Το κεντρικό αρχείο HTML της εφαρμογής                 |

|                  |                                                |
|------------------|------------------------------------------------|
| — nginx.conf     | # Ρυθμίσεις διακομιστή Nginx για το deployment |
| — package.json   | # Διαχείριση εξαρτήσεων και scripts            |
| — README.md      | # Τεκμηρίωση του project                       |
| └ vite.config.js | # Ρυθμίσεις του εργαλείου Vite                 |

#### 4.4.4 Επικοινωνία frontend - backend

Η επικοινωνία μεταξύ frontend και backend γίνεται αποκλειστικά μέσω RESTful HTTP αιτήσεων με τα endpoints του backend να απαιτούν την ύπαρξη κεφαλίδας αυθεντικοποίησης (Authentication Header) με έγκυρο JWT token για την εκτέλεση προστατευμένων λειτουργιών, επιστρέφοντας απαντήσεις σε μορφή JSON.

Το frontend λαμβάνει το token κατά τη διαδικασία επιτυχούς σύνδεσης (Login) του χρήστη και το αποθηκεύει προσωρινά στον τοπικό χώρο αποθήκευσης (localStorage) του φυλλομετρητή. Στη συνέχεια, φροντίζει να το ενσωματώνει αυτόματα σε κάθε επόμενη αίτηση προς τον εξυπηρετητή, διασφαλίζοντας τη συνεχή ταυτοποίηση του χρήστη.

Λόγω της φύσης των δεδομένων των DPPs, η πλειονότητα των παρεχόμενων λειτουργιών απαιτεί αυθεντικοποίηση. Το σύστημα υλοποιεί το σχήμα **JWT Bearer Authentication**.

## 4.5 Επαλήθευση και αξιολόγηση

Η επικύρωση και επαλήθευση αποτελεί σημαντικό στάδιο της διαδικασίας ανάπτυξης λογισμικού, διασφαλίζοντας ότι το σύστημα όχι μόνο λειτουργεί όπως αναμένεται αλλά και συμμορφώνεται πλήρως με τις απαιτήσεις που τέθηκαν. Αυτή η ενότητα περιγράφει τις μεθόδους, τα εργαλεία και τα αποτελέσματα των δοκιμών που πραγματοποιήθηκαν για την αξιολόγηση της ποιότητας και αξιοπιστίας του αναπτυχθέντος συστήματος, με ιδιαίτερη έμφαση στις εξειδικευμένες λειτουργίες διαχείρισης των Ψηφιακών Διαβατηρίων Προϊόντων (DPPs).

### 4.5.1 Μέθοδος ελέγχου ποιότητας

Για την ολιστική αξιολόγηση της πλατφόρμας, εφαρμόστηκε μια στρατηγική δοκιμών πολλαπλών επιπέδων, καλύπτοντας τόσο την ορθότητα των υπολογιστικών διεργασιών όσο και τη συνοχή των δεδομένων στο υβριδικό μοντέλο αποθήκευσης.

Για την επαλήθευση της λειτουργικότητας του **backend**, αξιοποιήθηκε η τεχνική των δοκιμών μονάδων (unit testing) με χρήση του ενσωματωμένου πλαισίου unittest<sup>22</sup> της Python. Οι δοκιμές επικεντρώθηκαν σε βασικές συναρτήσεις που σχετίζονται με τη διαχείριση χρηστών καθώς και την διαχείριση και αναζήτηση των DPPs. Επιπρόσθετα, στο πλαίσιο του ελέγχου της εισαγωγής δεδομένων (data ingestion) των DPPs, δημιουργήθηκαν unittest για επαλήθευση της

<sup>22</sup> unittest: <https://docs.python.org/3/library/unittest.html>

ορθής αποσυμπίεσης των πακέτων και της επιτυχούς εξαγωγής των τεχνικών χαρακτηριστικών από τα υπομοντέλα (submodels).

Εφαρμόστηκαν δοκιμές ενοποίησης για τον έλεγχο της συνεργασίας μεταξύ των επιμέρους στοιχείων του συστήματος, ακολουθώντας μια υβριδική προσέγγιση. Αρχικά, αναπτύχθηκαν αυτοματοποιημένα σενάρια ελέγχου (automated test scripts) με τη χρήση του πλαισίου pytest<sup>23</sup> και του TestClient του FastAPI. Μέσω αυτών, προσομοιώθηκαν πλήρεις ροές χρήσης (π.χ. Εγγραφή -> Login -> Δημιουργία DPP), ελέγχοντας προγραμματιστικά την ορθή ενημέρωση της βάσης δεδομένων και την επιστροφή των αναμενόμενων κωδικών HTTP.

Συμπληρωματικά, πραγματοποιήθηκαν χειροκίνητοι έλεγχοι με τη χρήση του εργαλείου Postman<sup>24</sup> και του Swagger UI. Η διαδικασία αυτή περιελάμβανε την εκτέλεση διαδοχικών αιτημάτων (chaining requests), όπου το JWT token που λαμβανόταν από το Login endpoint αντιγραφόταν στις κεφαλίδες των επόμενων κλήσεων, επιτρέποντας την επαλήθευση της ροής αυθεντικοποίησης και του ελέγχου πρόσβασης (RBAC) σε πραγματικές συνθήκες. Ιδιαίτερη έμφαση δόθηκε στον έλεγχο του συγχρονισμού με το Fuseki, επιβεβαιώνοντας οπτικά (μέσω του Fuseki UI) ότι κάθε εγγραφή στην PostgreSQL δημιουργούσε τον αντίστοιχο γράφο.

Το Swagger UI, ειδικότερα, αποτέλεσε τον κύριο μηχανισμό για τη διενέργεια χειροκίνητων δοκιμών API (Manual API Testing) κατά τη φάση ανάπτυξης. Συγκεκριμένα, επέτρεψε την άμεση επαλήθευση της ορθότητας των endpoints και των σχημάτων δεδομένων (data schemas) πριν την υλοποίηση του Frontend, λειτουργώντας ως ένα πρώτο επίπεδο ελέγχου ενοποίησης.

Για τη δοκιμή των προστατευμένων endpoints, το Swagger UI έχει παραμετροποιηθεί ώστε να υποστηρίζει το σχήμα ασφαλείας OAuth2/JWT της πλατφόρμας. Η ροή ελέγχου ακολουθεί τα εξής βήματα:

1. **Αυθεντικοποίηση (Login):** Ο χρήστης αποστέλλει τα διαπιστευτήριά του στο endpoint `/auth/login`. Το σύστημα επαληθεύει τα στοιχεία στην σχεσιακή βάση PostgreSQL και επιστρέφει ένα `access_token` σε μορφή JWT.
2. **Εξουσιοδότηση (Authorization):** Το παραχθέν Bearer Token εισάγεται στο καθολικό πεδίο ασφαλείας του Swagger (μέσω της επιλογής 'Authorize'). Με αυτή την ενέργεια, το εργαλείο ενσωματώνει αυτόματα την κεφαλίδα `Authorization: Bearer <token>` σε κάθε επόμενη κλήση. Αυτό επιτρέπει στο backend να αναγνωρίσει τον χρήστη, να επαληθεύσει τον ρόλο του (RBAC) και να επιτρέψει ή να απορρίψει την πρόσβαση σε πραγματικό χρόνο.

---

<sup>23</sup> pytest: <https://docs.pytest.org/>

<sup>24</sup> Postman: <https://www.postman.com/>



PostgreSQL. Εάν πρόκειται για σημασιολογική αναζήτηση (SPARQL), το αίτημα προωθείται στο Apache Jena Fuseki για εκτέλεση πάνω στον Γράφο Γνώσης.

POST /dpp/json/search Search Dpp Entries

Performs search across DPP entries supporting Simple, Advanced, and SPARQL modes. Access control (is\_published vs. owner\_id) is applied automatically.

Parameters

No parameters

Request body <sup>required</sup>

application/json

Edit Value | Schema

```
{
  "search_mode": "advanced",
  "advanced_criteria": [
    {
      "field_key": "manufacturer",
      "field_value": "Siemens",
      "match_type": "partial"
    }
  ],
  "limit": 10
}
```

Execute Clear

Εικόνα 22: Παραμετροποίηση αιτήματος προηγμένης αναζήτησης (advanced search) με χρήση φίλτρων μερικού ταιριάσματος (partial match) στο Swagger UI.

Request URL

http://127.0.0.1:8000/dpp/json/search

Server response

Code Details

200

Response body

```
{
  "total_count": 1,
  "limit": 10,
  "offset": 0,
  "results": [
    {
      "product_id": null,
      "manufacturer": "Siemens AG",
      "model_number": "1FK7063-2AF71-1RA0",
      "serial_number": "SN-2024-M01-998811",
      "production_date": "2024-05-15",
      "attributes": {},
      "submodels": [
        {
          "idShort": "TechnicalData",
          "semanticId": "https://admin-shell.io/ZWEI/TechnicalData/1/1",
          "submodelElements": {
            "Voltage": "400 V",
            "MaxSpeed": "6000 rpm",
            "RatedPower": "2.3 kW",
            "CoolingType": "Natural Convection",
            "Ratedtorque": "11 Nm",
            "EfficiencyClass": "IE4 (Super Premium Efficiency)",
            "ProtectionClass": "IP65"
          }
        }
      ]
    },
    {
      "idShort": "CarbonFootprint",

```

Download

Εικόνα 23: Αποτέλεσμα επιτυχούς ανάκτησης δεδομένων σε μορφή JSON, μετά την εφαρμογή του μηχανισμού flattening για την ανάγνωση των υπομοντέλων (submodels).

Κατά την ανάκτηση δεδομένων, το Backend επιστρέφει τα αποτελέσματα σε μορφή JSON. Για τη διευκόλυνση της παρουσίασης στο Frontend, εφαρμόζεται ένας μηχανισμός ισοπέδωσης

(flattening), ο οποίος απλοποιεί την πολύπλοκη ιεραρχική δομή των υπο-μοντέλων (submodels) σε ένα πιο επίπεδο σχήμα, άμεσα αξιοποιήσιμο από τα UI components.

Παράλληλα, για την υποστήριξη της σημασιολογικής αναζήτησης, το Frontend επικοινωνεί με το endpoint `/dpp/sparql/query`. Μέσω αυτού, ο διαχειριστής δύναται να εκτελεί ερωτήματα SPARQL απευθείας από το Dashboard, λαμβάνοντας τις απαντήσεις σε δομημένη μορφή (JSON bindings) που προέρχονται από τον Γράφο Γνώσης.

Για τη διασφάλιση της ακεραιότητας των δεδομένων κατά την επικοινωνία, χρησιμοποιούνται Pydantic schemas στο backend που επικυρώνουν (validate) το JSON payload των εισερχόμενων αιτημάτων πριν αυτά υποβληθούν προς επεξεργασία. Σε περίπτωση λανθασμένου JSON περιεχομένου, το API επιστρέφει αυτόματα κατάλληλους κωδικούς σφάλματος (HTTP 422 Unprocessable Entity), εξασφαλίζοντας τη σταθερότητα του συστήματος.

Για την επαλήθευση της λειτουργικότητας του **frontend**, εφαρμόστηκε μια συνδυαστική προσέγγιση ελέγχου ποιότητας. Για την αυτοματοποιημένη επαλήθευση της ορθής λειτουργίας των συστατικών (components), ενσωματώθηκε το πλαίσιο δοκιμών Vitest<sup>25</sup> σε συνδυασμό με τη βιβλιοθήκη React Testing Library<sup>26</sup>. Μέσω αυτών, αναπτύχθηκαν σενάρια δοκιμών (test suites) που προσομοιώνουν τη συμπεριφορά του χρήστη και ελέγχουν:

1. Την ορθή απόδοση (rendering) κρίσιμων σελίδων όπως η φόρμα σύνδεσης (Login), ο πίνακας ελέγχου (Dashboard) και η φόρμα δημιουργίας DPP.
2. Τη διαχείριση καταστάσεων (state management), επιβεβαιώνοντας ότι οι αλληλεπιδράσεις (π.χ. συμπλήρωση πεδίων, πάτημα κουμπιών) ενημερώνουν σωστά την κατάσταση της εφαρμογής.
3. Την ορθή κλήση των υπηρεσιών API, χρησιμοποιώντας τεχνικές mocking (μέσω του `vi.mock`) για να απομονωθεί το frontend από το backend και να ελεγχθεί η συμπεριφορά του σε περιπτώσεις επιτυχίας αλλά και αποτυχίας (π.χ. εμφάνιση μηνυμάτων σφάλματος).

Επιπλέον, διενεργήθηκαν χειροκίνητες δοκιμές χρηστικότητας και αποκρισιμότητας (responsiveness) μέσω των εργαλείων προγραμματιστή (browser dev tools) των φυλλομετρητών Google Chrome και Mozilla Firefox. Συγκεκριμένα, χρησιμοποιήθηκε η λειτουργία Device Mode για την προσομοίωση της διεπαφής σε ποικίλες αναλύσεις (π.χ. iPhone SE, iPad Air, Desktop 1920x1080), επιβεβαιώνοντας την ορθή αναδιάταξη των στοιχείων (grid/flexbox) και την προσαρμογή του μενού πλοήγησης (hamburger menu σε mobile). Παράλληλα, μέσω του εργαλείου Lighthouse<sup>27</sup>, ελέγχθηκε η προσβασιμότητα (Accessibility) και η ορθή λειτουργία της διεθνοποίησης (i18n) σε διαφορετικά σενάρια χρήσης.

#### 4.5.2 Εργαλεία και περιβάλλον δοκιμών Backend

Σε συνέχεια της μεθοδολογίας που αναλύθηκε στην Ενότητα 4.5.1, παρουσιάζεται εδώ η τεχνική υλοποίηση των ελέγχων. Για την εξασφάλιση της ορθότητας του κώδικα, χρησιμοποιήθηκε το πλαίσιο pytest σε συνδυασμό με τον AsyncClient (httpx), ο οποίος επιτρέπει την εκτέλεση ασύγχρονων κλήσεων προς το API.

<sup>25</sup> Vitest: <https://vitest.dev/>

<sup>26</sup> React Testing Library: <https://testing-library.com/docs/react-testing-library/intro>

<sup>27</sup> Google Lighthouse: <https://developer.chrome.com/docs/lighthouse/overview/>

Η στρατηγική ελέγχου διαχωρίζεται σε δύο βασικές κατηγορίες (οι έλεγχοι Frontend αναλύονται σε επόμενη ενότητα, την 4.5.3):

1. Αυτοματοποιημένοι Έλεγχοι (Pytest): Χρησιμοποιήθηκαν τόσο για Δοκιμές Μονάδας (Unit Tests), ελέγχοντας μεμονωμένες συναρτήσεις, όσο και για Δοκιμές Ενοποίησης (Integration Tests), καλύπτοντας ολοκληρωμένα σενάρια χρήσης και την αλληλεπίδραση με τη βάση δεδομένων.
2. Χειροκίνητοι Έλεγχοι (Postman): Εστίασαν στην μεμονωμένη επαλήθευση κάθε endpoint (Unit Testing of API endpoints) και στον έλεγχο των απαντήσεων (status codes, payload).

### ***Παράδειγμα Δοκιμής Μονάδας (Unit Test)***

Οι δοκιμές μονάδας εστιάζουν σε μεμονωμένες συναρτήσεις, όπως η επαλήθευση των βοηθητικών συναρτήσεων ασφαλείας. Ακολουθεί παράδειγμα ελέγχου της συνάρτησης hashing κωδικών:

```
from app.utils.security import hash_password, verify_password

def test_password_hashing():
    """
    Unit Test: Verify that password hashing and verification work
    correctly.
    """
    password = "secret_password"
    hashed = hash_password(password)

    # Ensure the hashed password is not the same as the plain text
    assert hashed != password

    # Ensure the correct password verifies successfully
    assert verify_password(password, hashed) is True

    # Ensure a wrong password fails verification
```

```
assert verify_password("wrong_password", hashed) is False
```

**Εικόνα 24:** Παράδειγμα δοκιμής μονάδας (unit test) - Ο κώδικας ελέγχει την ορθή λειτουργία των συναρτήσεων κρυπτογράφησης (hash\_password) και επαλήθευσης (verify\_password) κωδικών πρόσβασης, διασφαλίζοντας ότι οι κωδικοί δεν αποθηκεύονται ως απλό κείμενο και ότι η διαδικασία ταυτοποίησης είναι αξιόπιστη.

### Παράδειγμα Δοκιμής Ενοποίησης (Integration Test)

Οι δοκιμές ενοποίησης ελέγχουν τη διαχείριση των DPPs και την αλληλεπίδραση με τη βάση. Όπως φαίνεται στο παρακάτω τμήμα κώδικα, προσομοιώνεται μια πλήρης ροή όπου ένας πιστοποιημένος χρήστης δημιουργεί ένα νέο διαβατήριο.

```
import pytest
from httpx import AsyncClient

@pytest.mark.asyncio
async def test_create_dpp_flow(client: AsyncClient):
    """
    Integration Test: Full flow of creating a DPP by an authenticated user.
    Steps:
    1. Login to get Access Token.
    2. Create a new DPP using the token.
    3. Verify the response and data integrity.
    """

    # --- Step 1: Login (Get Token) ---
    login_payload = {
        "username": "zoe",
        "password": "test"
    }

    login_res = await client.post("/auth/login", json=login_payload)

    if login_res.status_code == 400:
        pytest.skip("User 'zoe' not found. Skipping integration test.")

    assert login_res.status_code == 200
    token = login_res.json()["access_token"]
    headers = {"Authorization": f"Bearer {token}"}

    # --- Step 2: Create DPP ---
    dpp_payload = {
        "title": "Integration Test Product",
        "product_identifier": "IT-999",
        "dpp_data": {
            "manufacturer": "Test Corp",
            "weight": 50,
            "submodels": [
                {"idShort": "TechnicalData", "submodelElements": {"power":
"100W"}}}
    ]
    }
```

```

    }
}

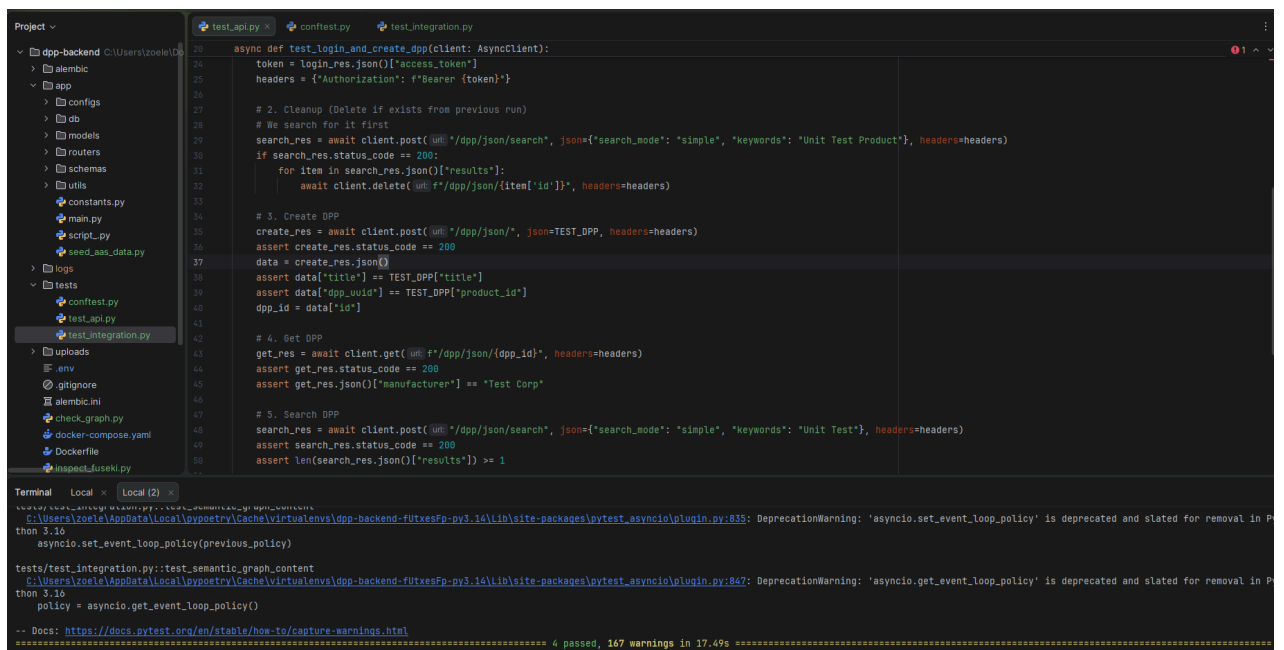
response = await client.post("/dpp/json/create", json=dpp_payload,
headers=headers)

# --- Step 3: Verify Response ---
assert response.status_code == 201
data = response.json()

assert data["title"] == dpp_payload["title"]
assert "dpp_uuid" in data
assert data["dpp_data"]["manufacturer"] == "Test Corp"
```

Εικόνα 25: Παράδειγμα δοκιμής ενοποίησης (integration test) - Ο κώδικας προσομοιώνει μια πλήρη ροή χρήστη (End-to-End scenario), ξεκινώντας από την αυθεντικοποίηση (login) για την απόκτηση token και προχωρώντας στη δημιουργία ενός νέου DPP. Στο τέλος, επαληθεύεται ότι η εγγραφή δημιουργήθηκε σωστά στη βάση δεδομένων και ότι η απάντηση του API περιέχει τα αναμενόμενα δεδομένα.

Η επιτυχής εκτέλεση αυτών των σεναρίων (όπως φαίνεται στην Εικόνα 26 με 4 passed tests) επιβεβαιώνει τη σταθερότητα του συστήματος κάτω από συνθήκες που προσομοιώνουν την πραγματική χρήση.

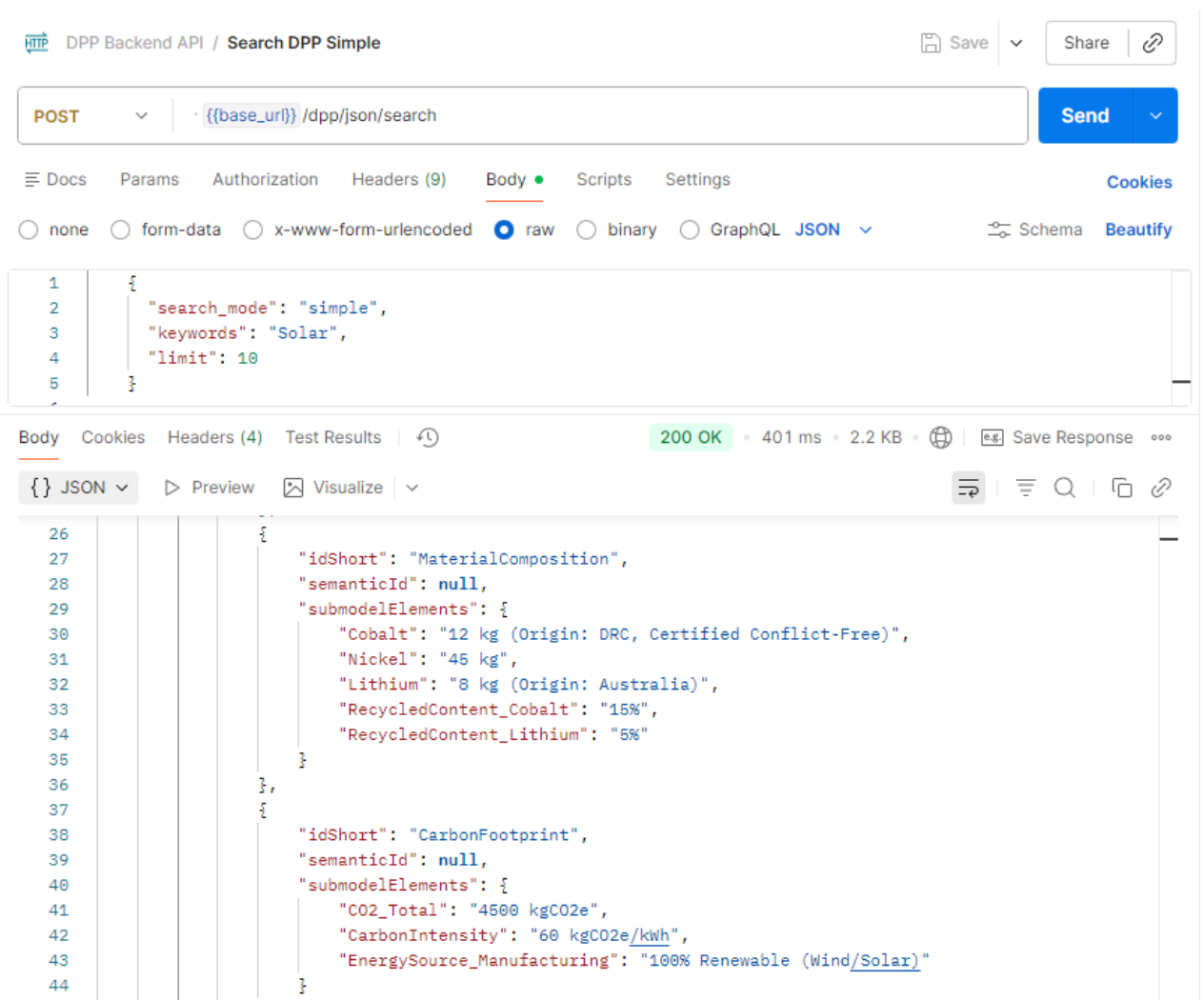


**Εικόνα 26: Περιβάλλον ανάπτυξης με τον κώδικα των integration tests και το αποτέλεσμα της επιτυχούς εκτέλεσης στο terminal.**

## *Postman*

Το Postman αποτέλεσε εργαλείο για τον έλεγχο των HTTP endpoints, της αυθεντικοποίησης και της διαχείρισης σφαλμάτων. Επιτρέπει την χειρωνακτική εκτέλεση σεναρίων χρήσης και την επαλήθευση των JSON αποκρίσεων σε πραγματικές συνθήκες. Η διαδικασία επαλήθευσης πραγματοποιήθηκε κυρίως μέσω οπτικού ελέγχου (Visual Inspection) των κωδικών κατάστασης HTTP (π.χ. 200 OK, 401 Unauthorized) και της δομής των δεδομένων. Επιπλέον, αξιοποιήθηκε η λειτουργικότητα των Environment Variables του Postman για την αυτόματη αποθήκευση και επαναχρησιμοποίηση του JWT token (Bearer Token) σε διαδοχικά αιτήματα, απλοποιώντας τη ροή ελέγχου (π.χ. Login  $\rightarrow$  Create DPP).

Όπως φαίνεται στην Εικόνα 27, το Postman χρησιμοποιήθηκε για την αποστολή αιτημάτων αναζήτησης και την επικύρωση ότι το backend επιστρέφει σωστά τα υπομοντέλα (submodels), όπως το *CarbonFootprint* και το *MaterialComposition*, διασφαλίζοντας την ακεραιότητα των δεδομένων. Η διασφάλιση της ορθότητας επιτεύχθηκε μέσω της σύγκρισης (cross-verification) των επιστρεφόμενων τιμών JSON με τα αρχικά δεδομένα που υποβλήθηκαν κατά τη δημιουργία του DPP (μέσω του Create endpoint), επιβεβαιώνοντας ότι δεν υπήρξε απώλεια ή αλλοίωση πληροφορίας κατά την αποθήκευση και ανάκτηση.

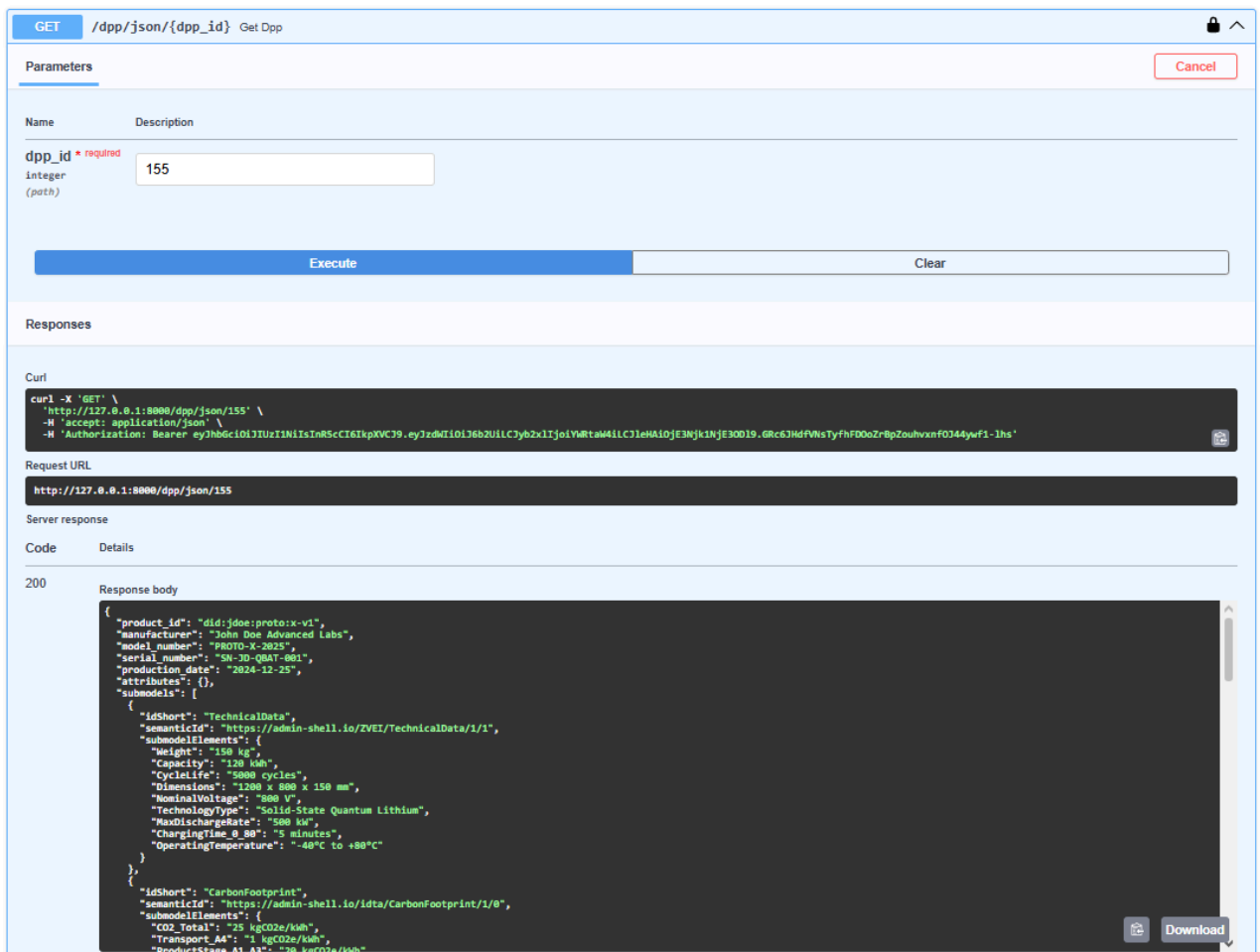


**Εικόνα 27: Επιτυχής εκτέλεση λειτουργικής δοκιμής (Functional Test) στο endpoint αναζήτησης /dpp/json/search μέσω του Postman, με επαλήθευση της ορθής δομής των δεδομένων του ψηφιακού διαβατηρίου (JSON response)**

### Swagger UI (OpenAPI 3.0)

Η πλατφόρμα αξιοποιεί την ενσωματωμένη υποστήριξη OpenAPI 3.0 μέσω του Swagger UI για την τεκμηρίωση και τη δοκιμή των υπηρεσιών (του backend) σε πραγματικό χρόνο. Λόγω της ευαίσθητης φύσης των δεδομένων των DPPs, η πλειονότητα των υπηρεσιών απαιτεί ταυτοποίηση μέσω του σχήματος JWT Bearer Authentication. Επίσης, το Swagger UI είναι πολύ χρήσιμο εργαλείο γιατί επιτρέπει τον έλεγχο των Pydantic schemas και την επαλήθευση των τύπων δεδομένων που ανταλλάσσονται, μεταξύ του frontend και του backend. Συγκεκριμένα, η πλατφόρμα διευκολύνει την επικύρωση της εισόδου (input validation), ενώ για την έξοδο των δεδομένων παρέχεται παθητική επικύρωση (passive validation), διευκολύνοντας τον εντοπισμό σφαλμάτων (debugging) κατά τη φάση της ανάπτυξης.

Επιπλέον, το Swagger χρησιμοποιήθηκε για την χειρωνακτική επαλήθευση λειτουργιών εξαγωγής δεδομένων. Στην Εικόνα 28, παρουσιάζεται η επιτυχής κλήση του endpoint /dpp/export/{dpp\_id}, όπου το σύστημα επιστρέφει το πλήρες περιεχόμενο του διαβατηρίου, συμπεριλαμβανομένων των τεχνικών χαρακτηριστικών και του αποτυπώματος άνθρακα (CarbonFootprint), σε δομημένη μορφή έτοιμη για επεξεργασία από τρίτα συστήματα.



Εικόνα 28: Διαδικασία εξαγωγής δεδομένων DPP και επαλήθευση της απόκρισης μέσω του Swagger UI.

### 4.5.3 Εργαλεία και περιβάλλον δοκιμών Frontend

Η διασφάλιση της ποιότητας του Frontend βασίστηκε σε σύγχρονα εργαλεία ελέγχου που ενσωματώνονται άμεσα στη ροή ανάπτυξης (development workflow). Για την εκτέλεση των δοκιμών χρησιμοποιήθηκε το Vitest, ένας ταχύτατος test runner συμβατός με το Vite, σε συνδυασμό με το React Testing Library για την προσομοίωση της αλληλεπίδρασης του χρήστη με το DOM.

Η στρατηγική ελέγχου του Frontend εστίασε στα εξής σημεία:

1. Δοκιμές Μονάδας & Συνιστωσών (Unit & Component Testing): Ελέγχθηκε η μεμονωμένη συμπεριφορά των React components (π.χ. Login, Dashboard, CreateDPP). Μέσω της τεχνικής mocking (χρησιμοποιώντας το vi.mock του Vitest), απομονώθηκαν οι εξωτερικές εξαρτήσεις (όπως οι κλήσεις API), επιτρέποντας τον έλεγχο της λογικής της διεπαφής ανεξάρτητα από τη διαθεσιμότητα του backend.
2. Έλεγχος Ροής Χρήστη (User Flow Testing): Επιβεβαιώθηκε ότι οι κρίσιμες διαδικασίες, όπως η υποβολή φόρμας και η πλοήγηση μετά από επιτυχή ενέργεια, εκτελούνται σωστά.

#### Παράδειγμα Δοκιμής Component (Login Test)

Το παρακάτω απόσπασμα κώδικα παρουσιάζει τον έλεγχο της φόρμας σύνδεσης. Αρχικά, επαληθεύεται ότι τα πεδία εισαγωγής (username, password) και το κουμπί υποβολής εμφανίζονται σωστά (Rendering Test). Στη συνέχεια, προσομοιώνεται η πληκτρολόγηση στοιχείων και το πάτημα του κουμπιού, ελέγχοντας αν η εφαρμογή καλεί τη σωστή συνάρτηση API με τα αναμενόμενα δεδομένα.

```
describe('Login Component', () => {
  // Clear mocks before each test to ensure isolation
  beforeEach(() => {
    vi.clearAllMocks();
  });
  // Verify that the login form renders correctly
  it('renders login form', () => {
    render(
      <BrowserRouter>
        <Login />
      </BrowserRouter>
    );
    // Check if username, password inputs and login button are present in the
    document
    expect(screen.getByLabelText(/username/i)).toBeInTheDocument();
    expect(screen.getByLabelText(/password/i)).toBeInTheDocument();
    expect(screen.getByRole('button', { name: /login/i })).toBeInTheDocument();
  });
  // Verify successful login flow
  it('handles successful login', async () => {
```

```
api.post.mockResolvedValueOnce({ data: { access_token: 'fake-token' } });

render(
  <BrowserRouter>
    <Login />
  </BrowserRouter>
);

fireEvent.change(screen.getByLabelText(/username/i), { target: { value: 'testuser' } });
fireEvent.change(screen.getByLabelText(/password/i), { target: { value: 'password123' } });
fireEvent.click(screen.getByRole('button', { name: /login/i }));

await waitFor(() => {
  expect(api.post).toHaveBeenCalledWith('/auth/login', {
    username: 'testuser',
    password: 'password123'
  }, expect.any(Object));
  expect(localStorage.getItem('token')).toBe('fake-token');
  expect(mockedNavigate).toHaveBeenCalledWith('/dashboard');
});
});
```

Εικόνα 29: : Δοκιμές Μονάδας Frontend (React Testing Library) - Έλεγχος της φόρμας εισόδου (Login Component).

**RERUN** rerun all tests

stderr | src/tests/Login.test.jsx > Login Component > handles login failure  
Login error: { response: { status: 401, data: { detail: 'Invalid credentials' } } }

✓ src/tests/App.test.jsx (1)  
✓ src/tests/Login.test.jsx (3)  
✓ src/tests/Dashboard.test.jsx (2)  
✓ src/tests/CreateDPP.test.jsx (1)

Test Files 4 passed (4)  
Tests 7 passed (7)  
Start at 03:31:41  
Duration 3.39s

**PASS** Waiting for file changes...  
press h to show help, press q to quit

Εικόνα 30: Στιγμιότυπο επιτυχούς εκτέλεσης των μονάδων ελέγχου (Unit Tests) του Frontend, όπου επιβεβαιώνεται η ορθή λειτουργία των βασικών συστατικών και η σωστή διαχείριση σφαλμάτων.

#### 4.5.4 Ενδεικτικά σενάρια δοκιμών

Για την αξιολόγηση της συμπεριφοράς του συστήματος, εκτελέστηκαν συγκεκριμένα σενάρια ελέγχου (Test Cases). Οι δοκιμές πραγματοποιήθηκαν μέσω της πλατφόρμας Postman, η οποία επέτρεψε τη διενέργεια ελέγχων ολοκλήρωσης (Integration Testing) σε πραγματικές συνθήκες λειτουργίας του API του backend. Τα ιδανικά αποτελέσματα των οποίων συνοψίζονται στον Πίνακα 3. Η διαδικασία περιλάμβανε τη χειροκίνητη αποστολή αιτημάτων για την επαλήθευση των αποκρίσεων του συστήματος, καθώς και την αυτόματη επικύρωση των δεδομένων εισόδου μέσω των Pydantic schemas.

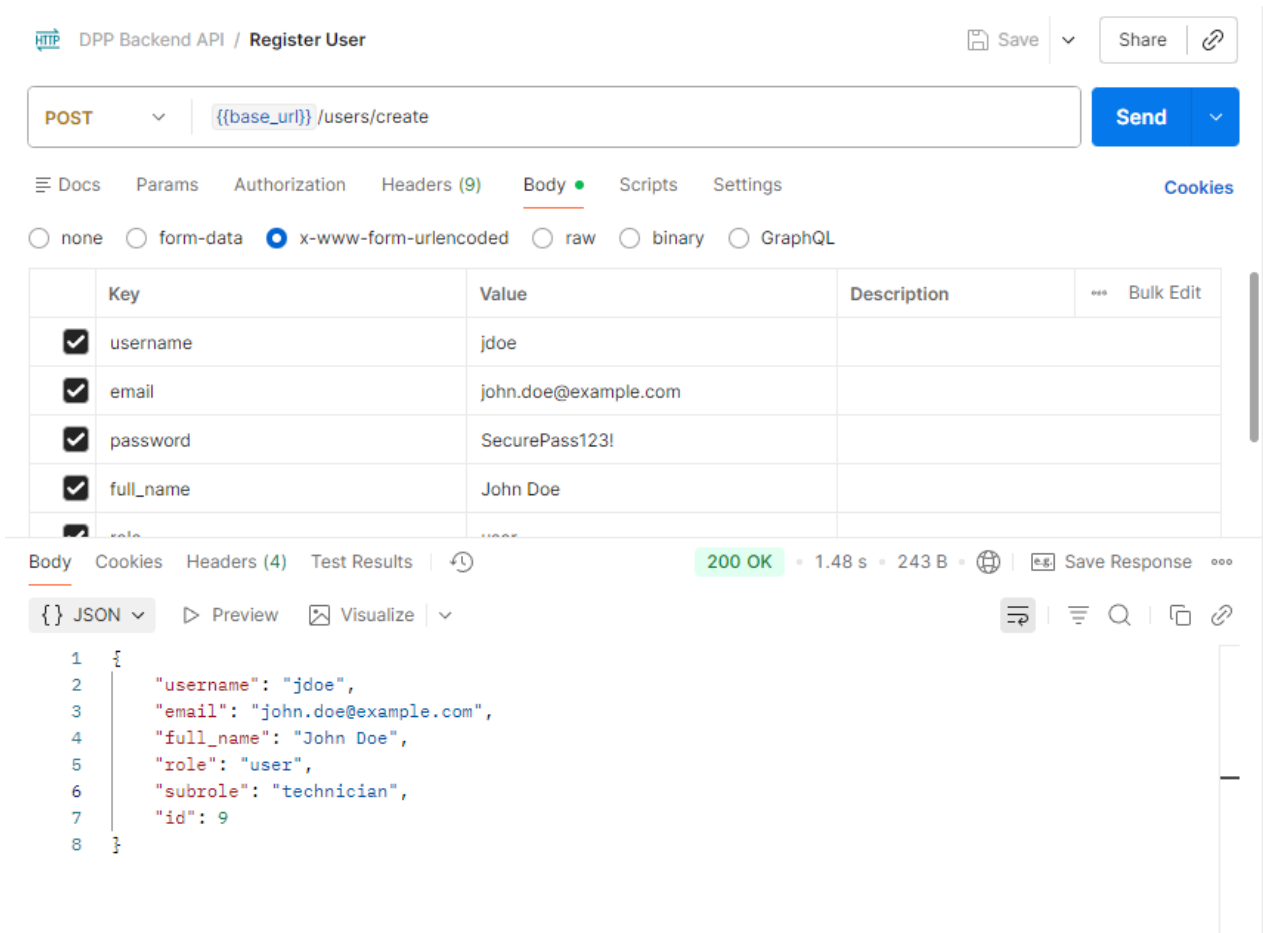
**Πίνακας 3: Σενάρια ελέγχου λειτουργικότητας και ασφάλειας**

| Αρ. | Σενάριο                             | Περιγραφή                                                                      | Αναμενόμενο Αποτέλεσμα                                 |
|-----|-------------------------------------|--------------------------------------------------------------------------------|--------------------------------------------------------|
| TC1 | Εγγραφή χρήστη με ήδη υπάρχον email | Ο χρήστης επιχειρεί εγγραφή με email που υπάρχει ήδη στη βάση (βλ. Εικόνα 31). | HTTP 400 + μήνυμα σφάλματος                            |
| TC2 | Σύνδεση με άκυρα στοιχεία           | Λανθασμένος κωδικός ή username (βλ. Εικόνα 32).                                | HTTP 401 Unauthorized                                  |
| TC3 | Δημιουργία DPP χωρίς τίτλο          | Αποστολή αιτήματος POST χωρίς τιμή στο πεδίο <b>title</b> (βλ. Εικόνα 33).     | HTTP 422 + validation error                            |
| TC4 | Προηγμένη Αναζήτηση                 | Εκτέλεση αναζήτησης με φίλτρο "Manufacturer: Siemens" (βλ. Εικόνα 34).         | Λίστα DPPs με σχετικές περιγραφές                      |
| TC5 | SPARQL query χωρίς δικαιώματα       | SPARQL από μη ιδιοκτήτη προς μη δημοσιευμένο DPP (βλ. Εικόνα 36).              | HTTP 403 Forbidden με συνοδευτικό μήνυμα Access Denied |

#### TC1: Έλεγχος Μοναδικότητας Ηλεκτρονικής Διεύθυνσης

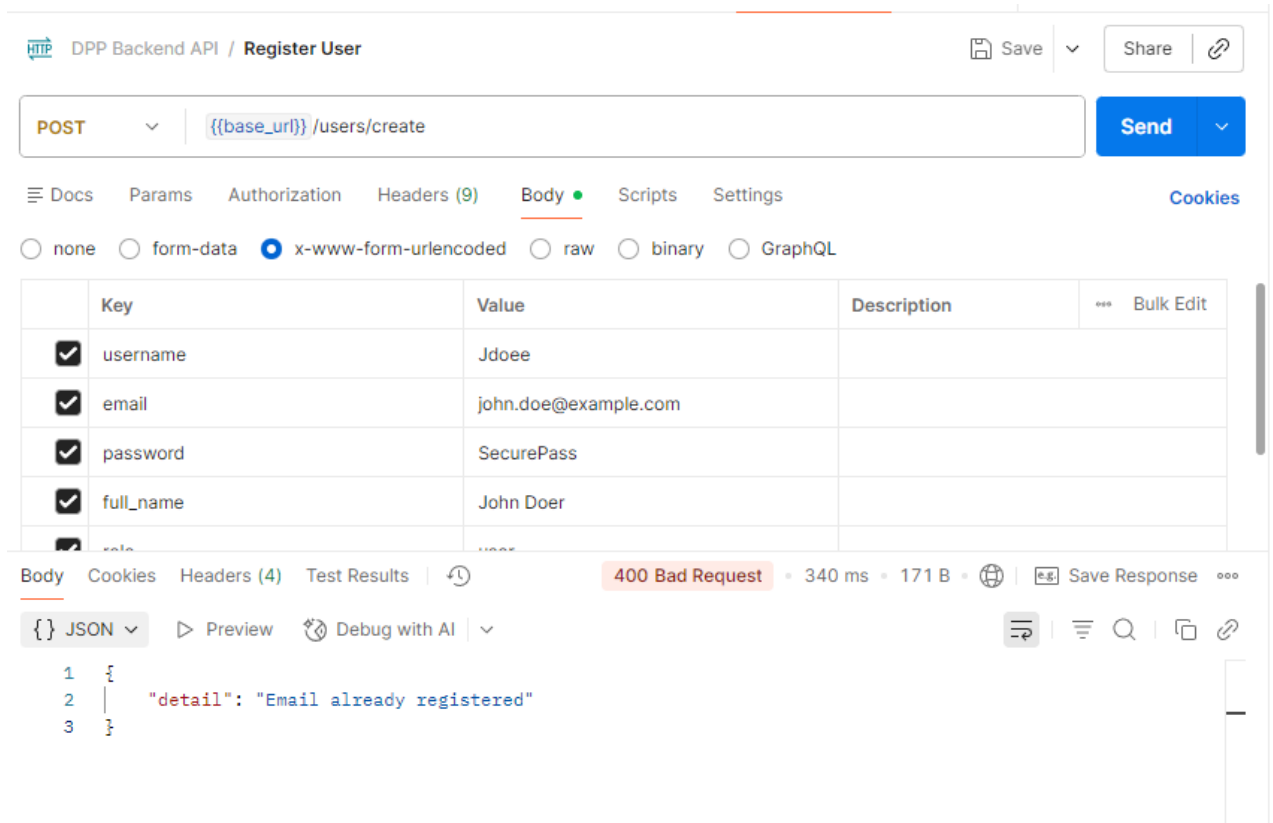
Σύμφωνα με τις προδιαγραφές ασφαλείας, το σύστημα πρέπει να αποτρέπει τη δημιουργία πολλαπλών λογαριασμών με την ίδια ηλεκτρονική διεύθυνση. Για την επαλήθευση αυτού του σεναρίου, πραγματοποιήθηκε αρχικά μια επιτυχής εγγραφή και εν συνεχεία μια απόπειρα εγγραφής με το ίδιο email.

Όπως φαίνεται στην Εικόνα 31, κατά το πρώτο βήμα στάλθηκε αίτημα **POST** στο endpoint **/users/create** με τα στοιχεία του χρήστη "jdoe" και τη διεύθυνση "john.doe@example.com". Το σύστημα επεξεργάστηκε επιτυχώς το αίτημα, επιστρέφοντας κωδικό κατάστασης 200 OK και το JSON αντικείμενο του νέου χρήστη, επιβεβαιώνοντας την ορθή αποθήκευση στην PostgreSQL.



**Εικόνα 31: Επιτυχής αρχική εγγραφή χρήστη μέσω του Postman, η οποία χρησιμοποιείται ως βάση για τον έλεγχο των περιορισμών μοναδικότητας.**

Στη συνέχεια, όπως απεικονίζεται στην Εικόνα 32, επιχειρήθηκε η εγγραφή ενός νέου χρήστη ("Jdoee") χρησιμοποιώντας την ίδια ηλεκτρονική διεύθυνση. Το backend, μέσω της λογικής επικύρωσης, εντόπισε τη σύγκρουση δεδομένων και απέρριψε το αίτημα επιστρέφοντας κωδικό 400 Bad Request. Η απόκριση περιλαμβάνει το μήνυμα **"Email already registered"**, το οποίο επιβεβαιώνει την επιτυχή εκτέλεση του σεναρίου ελέγχου TC1.

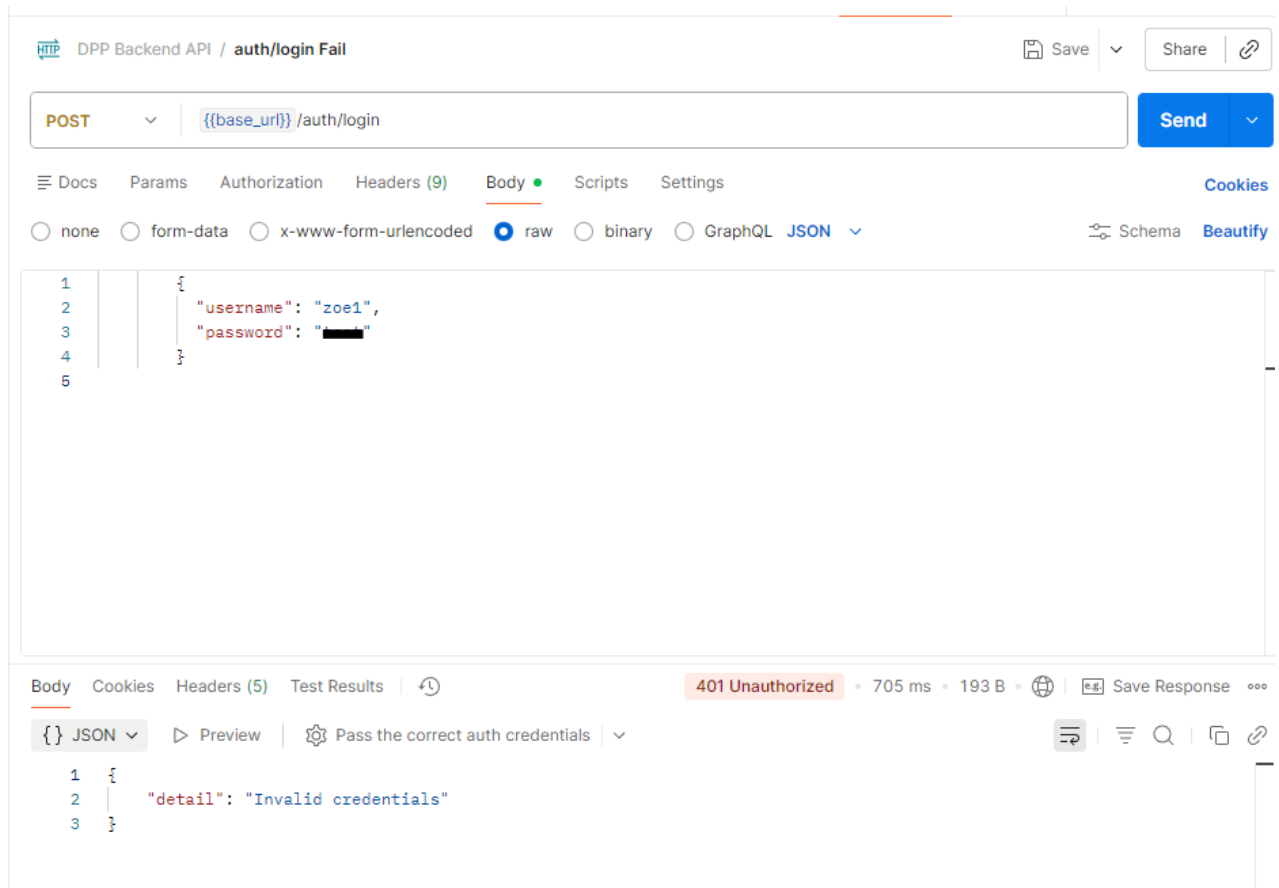


Εικόνα 32: Απόρριψη αίτησης εγγραφής από το API λόγω προϋπάρχουσας ηλεκτρονικής διεύθυνσης στη βάση δεδομένων (HTTP 400 Bad Request).

## TC2: Έλεγχος Αυθεντικοποίησης με Άκυρα Στοιχεία

Η διασφάλιση ότι μόνο έγκυροι χρήστες έχουν πρόσβαση στο σύστημα αποτελεί κρίσιμη απαίτηση για την προστασία των δεδομένων των ψηφιακών διαβατηρίων προϊόντων. Για την επικύρωση αυτής της λειτουργίας, εκτελέστηκε δοκιμή σύνδεσης με τη χρήση λανθασμένου κωδικού πρόσβασης ή ονόματος χρήστη.

Όπως αποτυπώνεται στην Εικόνα 33, στάλθηκε αίτημα **POST** στο endpoint `/auth/login` με λανθασμένα στοιχεία εισόδου, και συγκεκριμένα με μη έγκυρο όνομα χρήστη ("zoe1"). Το backend, κατόπιν ελέγχου στη βάση δεδομένων, απέρριψε την είσοδο επιστρέφοντας κωδικό κατάστασης 401 Unauthorized με την ένδειξη "Invalid credentials". Η συγκεκριμένη απόκριση επιβεβαιώνει ότι ο μηχανισμός ασφαλείας απομονώνει επιτυχώς τις μη εξουσιοδοτημένες προσπάθειες πρόσβασης, προστατεύοντας τα endpoints της πλατφόρμας.



**Εικόνα 33:** Αποτυχία αυθεντικοποίησης στο endpoint `/auth/login` λόγω παροχής λανθασμένων στοιχείων, με την επιστροφή κατάλληλου μηνύματος σφάλματος από το API.

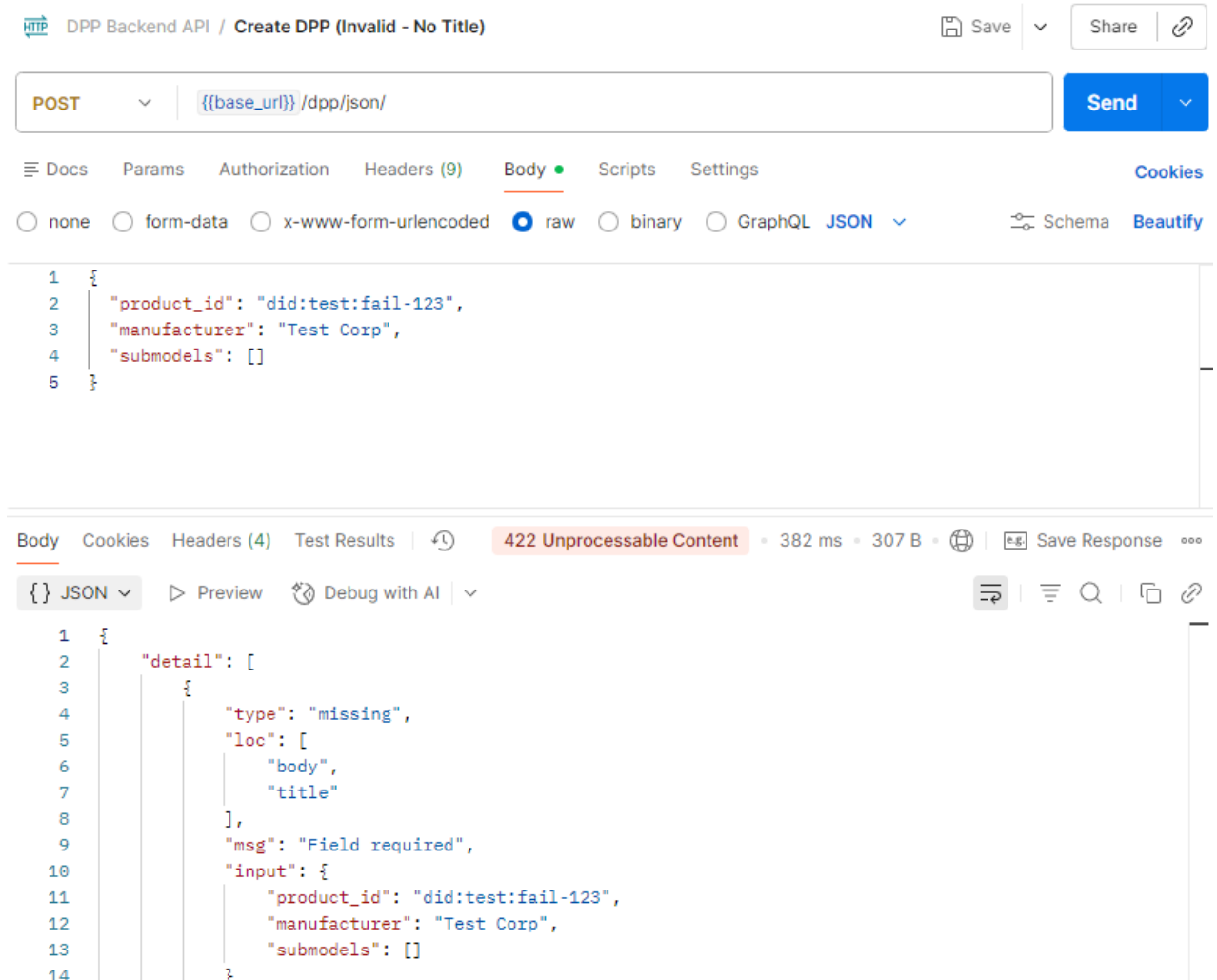
### TC3: Έλεγχος Επικύρωσης Δεδομένων κατά τη Δημιουργία DPP

Η διασφάλιση ότι κάθε Ψηφιακό Διαβατήριο Προϊόντος περιλαμβάνει το ελάχιστο σύνολο υποχρεωτικών πληροφοριών είναι κρίσιμη για τη σημασιολογική ορθότητα του συστήματος. Για τον έλεγχο αυτό, χρησιμοποιούνται τα σχήματα Pydantic, τα οποία εκτελούν αυτόματη επικύρωση των εισερχόμενων JSON αιτημάτων πριν αυτά προωθηθούν για επεξεργασία ή αποθήκευση.

Στην Εικόνα 34, παρουσιάζεται η εκτέλεση ενός αιτήματος POST στο endpoint `/dpp/json/` με σκοπό τη δημιουργία μιας νέας εγγραφής διαβατηρίου. Στο σώμα του αιτήματος παραλείφθηκαν σκόπιμα οι τιμές σε δύο υποχρεωτικά πεδία, μεταξύ των οποίων και το πεδίο `title`, τα οποία αποτελούν απαραίτητες παραμέτρους σύμφωνα με τις προδιαγραφές του μοντέλου δεδομένων.

Όπως ήταν αναμενόμενο, το API απέρριψε το αίτημα επιστρέφοντας κωδικό κατάστασης 422 Unprocessable Content (καθώς πρόκειται για σημασιολογικό και όχι συντακτικό λάθος). Η λεπτομερής απόκριση JSON που επιστρέφεται στον χρήστη προσδιορίζει με ακρίβεια την αιτία της αποτυχίας, αναφέροντας για κάθε παράλειψη ότι το πεδίο λείπει (`"type": "missing"`) και είναι απαραίτητο (`"msg": "Field required"`). Η επιτυχής εκτέλεση αυτού του σεναρίου

αποδεικνύει ότι το backend προστατεύει τη βάση δεδομένων από ελλειπείς καταχωρήσεις, διατηρώντας υψηλό επίπεδο ποιότητας δεδομένων.



Εικόνα 34: Αυτόματη επικύρωση σχήματος δεδομένων από το backend και απόρριψη αιτήματος λόγω παράλειψης υποχρεωτικών πεδίων (HTTP 422 Validation Error).

#### TC4: Προηγμένη Αναζήτηση και Ανάκτηση Σύνθετων Δεδομένων

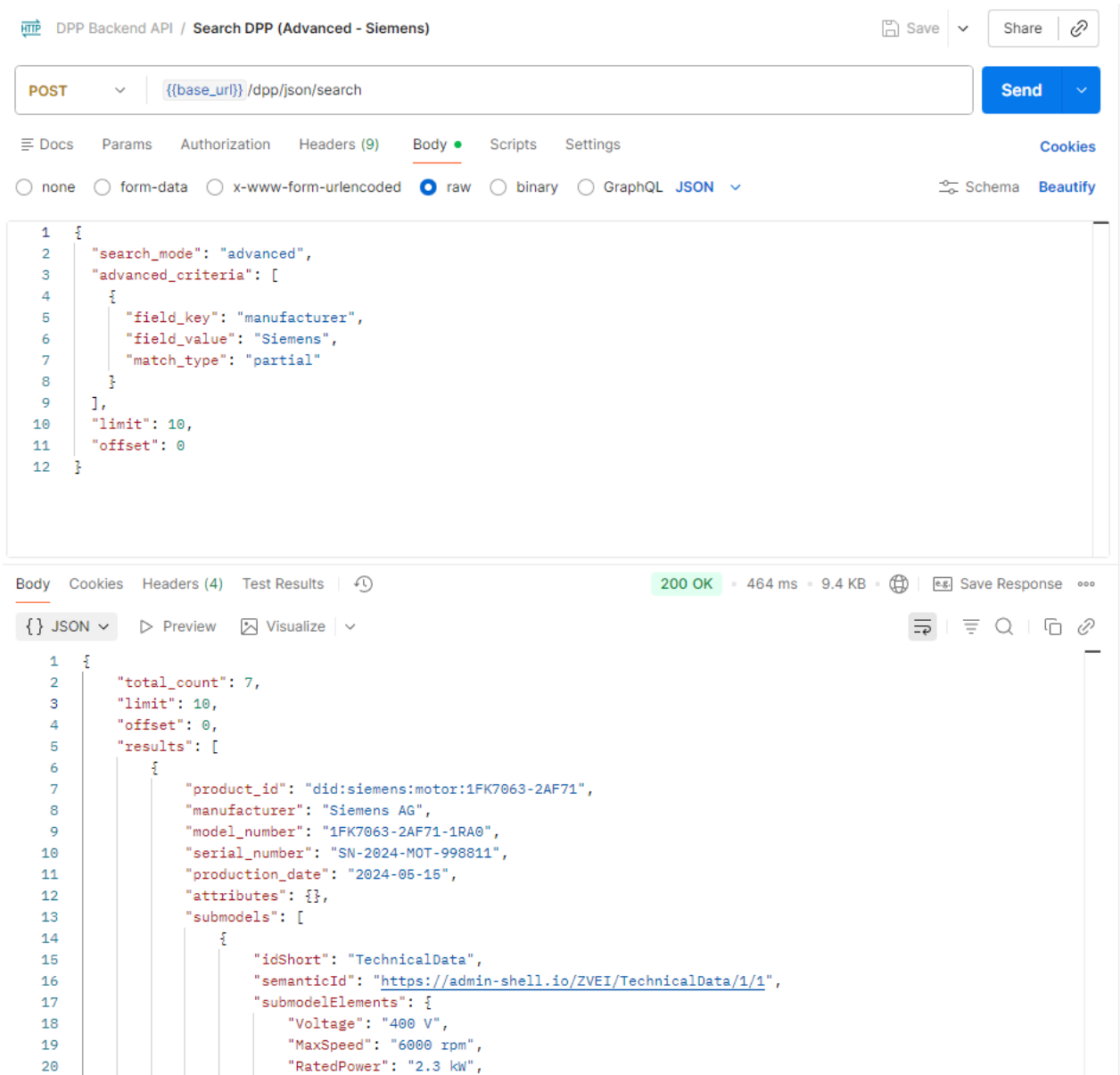
Η λειτουργία της προηγμένης αναζήτησης (Advanced Search) επιτρέπει στους χρήστες να φιλτράρουν τα διαβατήρια βάσει συγκεκριμένων παραμέτρων, όπως ο κατασκευαστής, το μοντέλο ή οι τεχνικές προδιαγραφές, προσφέροντας μεγαλύτερη ακρίβεια σε σχέση με την απλή αναζήτηση λέξεων-κλειδίων.

Στην Εικόνα 35, παρουσιάζεται η αποστολή ενός αιτήματος στο endpoint `/dpp/json/search` με την παράμετρο `search_mode` ορισμένη ως `"advanced"`. Το αίτημα περιλαμβάνει σύνθετα κριτήρια (`advanced_criteria`), όπου αναζητούνται εγγραφές με πεδίο

κατασκευαστή (`field_key: "manufacturer"`) που περιέχει την τιμή `"Siemens"`. Για την εκτέλεση της αναζήτησης ορίζεται ότι απαιτείται μερικό ταίριασμα (`match_type: "partial"`), επιτρέποντας στο σύστημα να εντοπίζει εγγραφές που περιέχουν τον όρο αναζήτησης ως υποσύνολο της πλήρους τιμής.

Το σύστημα επεξεργάστηκε επιτυχώς το φίλτρο και επέστρεψε κωδικό 200 OK, παρέχοντας μια λίστα αποτελεσμάτων που ταυτίζονται με τα κριτήρια, όπως η εγγραφή για την εταιρεία `"Siemens AG"`.

Η επιτυχής εκτέλεση του **TC4** επιβεβαιώνει ότι η πλατφόρμα υποστηρίζει αποτελεσματικά τη διαλειτουργικότητα και την ανάκτηση πληροφοριών υψηλής ανάλυσης, διευκολύνοντας την ιχνηλασιμότητα των προϊόντων.



**Εικόνα 35:** Εκτέλεση προηγμένης αναζήτησης με φιλτράρισμα βάσει κατασκευαστή και επιβεβαίωση της ορθής ανάκτησης των σχετικών εγγραφών.

## TC5: Έλεγχος Πρόσβασης σε Σημασιολογικά Δεδομένα (SPARQL RBAC)

Το συγκεκριμένο σενάριο ελέγχει την αποτελεσματικότητα του μοντέλου ελέγχου πρόσβασης (RBAC) όταν ένας μη εξουσιοδοτημένος χρήστης επιχειρεί να ανακτήσει πληροφορίες μέσω απευθείας ερωτημάτων SPARQL.

Όπως παρουσιάζεται στην Εικόνα 36, ο χρήστης "John Doe" (John Doe Advanced Labs) έχει δημιουργήσει ένα ψηφιακό διαβατήριο για το προϊόν με αναγνωριστικό **did:jdoe:proto:x-v1**. Το διαβατήριο αυτό βρίσκεται σε κατάσταση "Draft" (μη δημοσιευμένο), γεγονός που σημαίνει ότι η πρόσβαση στα σημασιολογικά του δεδομένα περιορίζεται αποκλειστικά στον ιδιοκτήτη και στους διαχειριστές του συστήματος.

DPP Backend API / Draft DPP (John's Product)

GET {{base\_url}}/dpp/json/:dpp\_id

Send

Docs Params Authorization Headers (9) Body Scripts Settings Cookies

Query Params

| Key | Value | Description | Bulk Edit |
|-----|-------|-------------|-----------|
| Key | Value | Description |           |

Path Variables

| Key    | Value | Description | Bulk Edit |
|--------|-------|-------------|-----------|
| dpp_id | 155   | Description |           |

Body Cookies Headers (4) Test Results 200 OK • 323 ms • 3.94 KB Save Response

{ } JSON Preview Visualize

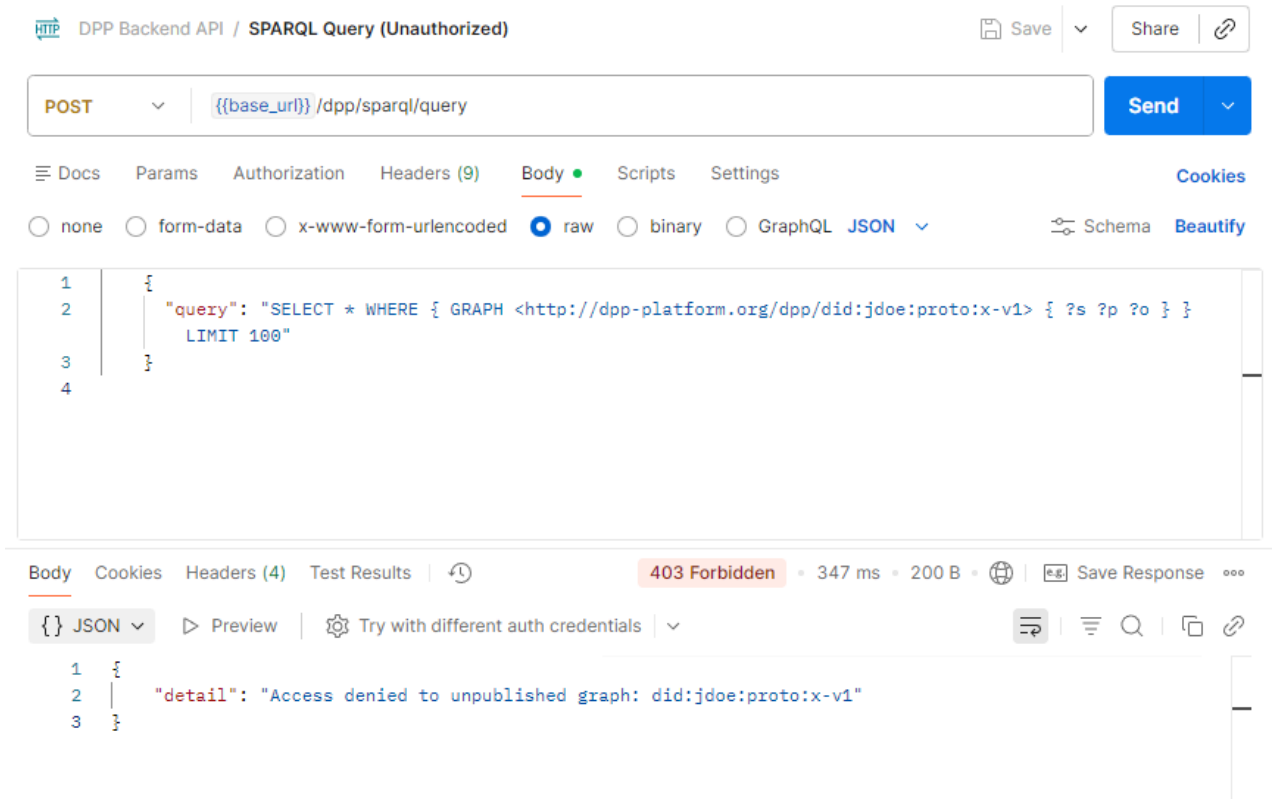
```
68   },
69   ],
70   "id": 155,
71   "dpp_uuid": "did:jdoe:proto:x-v1",
72   "is_owner": true,
73   "title": "Prototype X - NextGen Quantum Battery",
74   "is_published": false,
75   "created_at": "2026-01-27T22:59:41.492172",
76   "updated_at": "2026-01-27T22:59:41.492175",
77   "dpp_data": {
78     "submodels": [
79       {
80         "idShort": "TechnicalData",
81         "semanticId": "https://admin-shell.io/ZVEI/TechnicalData/1/1",
82         "submodelElements": {
83           "Weight": "150 kg",
84           "Capacity": "120 kWh",
85           "CycleLife": "5000 cycles",
86           "Dimensions": "1200 x 800 x 150 mm",
87         }
88       }
89     ]
90   }
91 }
```

Εικόνα 36: Διαδικασία ανάκτησης για το αναγνωριστικό **dpp\_id** από τον ιδιοκτήτη (John), το οποίο παραμένει σε μη δημοσιευμένη κατάσταση.

Στη συνέχεια, εξετάζεται η περίπτωση όπου ένας τρίτος χρήστης (π.χ. "Alice"), επιχειρεί να προσπελάσει ένα μη δημοσιευμένο ψηφιακό διαβατήριο. Στο πλαίσιο ενός σεναρίου ελέγχου διείσδυσης (Penetration Testing), υποτίθεται ότι ο κακόβουλος χρήστης έχει κατορθώσει να υποκλέψει ή να προβλέψει το μοναδικό αναγνωριστικό (URI/DID) του προϊόντος.

Όπως φαίνεται στην Εικόνα 37, το αίτημα αποστέλλεται στο endpoint `/dpp/sparql/query`. Παρά το γεγονός ότι το ερώτημα είναι συντακτικά ορθό, το backend αναγνωρίζει ότι ο αιτών χρήστης δεν διαθέτει τα απαραίτητα δικαιώματα για τον συγκεκριμένο γράφο δεδομένων. Το σύστημα απορρίπτει το αίτημα επιστρέφοντας κωδικό κατάστασης 403 Forbidden με το επεξηγηματικό μήνυμα: `"Access denied to unpublished graph: did:jdoe:proto:x-v1"`.

Η επιτυχής ολοκλήρωση του TC5 επιβεβαιώνει ότι η γνώση του αναγνωριστικού (ID) ενός προϊόντος δεν συνεπάγεται την πρόσβαση στα δεδομένα του, διασφαλίζοντας ότι τα μη δημοσιευμένα ψηφιακά διαβατήρια παραμένουν προστατευμένα από μη εξουσιοδοτημένη ανάγνωση.



**Εικόνα 37:** Απόρριψη μη εξουσιοδοτημένου SPARQL ερωτήματος από το API (HTTP 403 Forbidden), διασφαλίζοντας την ιδιωτικότητα των δεδομένων του John.

#### 4.5.5 Συμπεράσματα δοκιμών

Η διεξαγωγή των παραπάνω δοκιμών κατέδειξε ότι το σύστημα ανταποκρίνεται στις θεμελιώδεις λειτουργικές απαιτήσεις, προσφέρει συνεπή και τεκμηριωμένη συμπεριφορά μέσω του RESTful API, και διαχειρίζεται σωστά την αυθεντικοποίηση, τις εξαιρέσεις και την πρόσβαση βάσει ρόλων. Επιπλέον, η χρήση Swagger ενίσχυσε σημαντικά την ευκολία ελέγχου κατά το στάδιο της ανάπτυξης, ενώ η χρήση Postman παρείχε δυνατότητα αναπαραγωγής των ελέγχων και εκτέλεση σεναρίων πραγματικής χρήσης.

#### 4.5.6 Ικανοποίηση απαιτήσεων

Στην παρούσα ενότητα παρουσιάζεται η συσχέτιση των λειτουργικών και μη λειτουργικών απαιτήσεων που τέθηκαν κατά τη φάση ανάλυσης με τα παραδοτέα της υλοποίησης. Για καθεμία από τις απαιτήσεις αξιολογείται ο βαθμός ικανοποίησής της και τεκμηριώνεται η σχετική αιτιολόγηση. Η προσέγγιση αυτή διασφαλίζει τη διαφάνεια στην κάλυψη του αρχικού πλαισίου προδιαγραφών και αναδεικνύει τα τεχνικά σημεία συμμόρφωσης ή τυχόν περιορισμούς.

**Πίνακας 4: Βαθμός ικανοποίησης απαιτήσεων**

| A/A | Κατηγορία Απαίτησης | Περιγραφή Απαίτησης                                                       | Βαθμός Ικανοποίησης | Αιτιολόγηση                                                                                                                                                                           |
|-----|---------------------|---------------------------------------------------------------------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | Λειτουργική         | Υποστήριξη διαφορετικών τύπων χρηστών (επισκέπτης, χρήστης, διαχειριστής) | Πλήρης              | Υλοποιήθηκε RBAC μοντέλο με χρήση JWT. Διαχωρισμένα endpoints & δικαιώματα. Διασφαλίζεται πλήρης κάλυψη στο πληροφοριακό μοντέλο της εφαρμογής (συσχέτιση χρηστών-ρόλων-δικαιωμάτων). |
| 2   | Λειτουργική         | Διαχείριση χρηστών (CRUD & Security questions)                            | Πλήρης              | Υλοποίηση όλων των endpoints με έλεγχο πρόσβασης.                                                                                                                                     |
| 3   | Λειτουργική         | Δημιουργία, επεξεργασία, διαγραφή, δημοσίευση/αποδημοσίευση DPP           | Πλήρης              | Υλοποίηση όλων των λειτουργιών μέσω του REST API. Ελέγχονται δικαιώματα ιδιοκτησίας κατά την εξουσιοδότηση για το εκάστοτε DPP.                                                       |

|   |                |                                                                                       |                         |                                                                                                                                                                                                                                                                                                                                                                                       |
|---|----------------|---------------------------------------------------------------------------------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 | Λειτουργική    | Αναζήτηση DPP με 3 τρόπους (λέξεις-κλειδιά, key-value, SPARQL) & σταθμισμένη κατάταξη | Πλήρης                  | Υλοποιήθηκαν και τα τρία είδη αναζήτησης. SPARQL μέσω Fuseki endpoint. Φιλτράρισμα αποτελεσμάτων μέσω ελέγχου πρόσβασης με βάση τα δικαιώματα ιδιοκτησίας και την κατάσταση του DPP (δημοσιευμένο ή όχι). Περιλαμβάνεται μηχανισμός σταθμισμένης κατάταξης (Weighted Ranking) μέσω της search_dpps, όπου τα αποτελέσματα ταξινομούνται βάσει score υπολογισμένου από sorting weights. |
| 5 | Λειτουργική    | Οπτικοποίηση και εξαγωγή DPP σε RDF μορφή                                             | Πλήρης                  | Δυνατότητα οπτικοποίησης των σημασιολογικών σχέσεων μέσω διαδραστικού γράφου και εξαγωγής των δεδομένων σε πρότυπα RDF (Turtle/XML).                                                                                                                                                                                                                                                  |
| 6 | Μη Λειτουργική | Responsive διεπαφή χρήστη                                                             | Πλήρης                  | Υλοποιήθηκε με HTML/CSS/Bootstrap, υποστηρίζει πολλαπλές συσκευές.                                                                                                                                                                                                                                                                                                                    |
| 7 | Μη Λειτουργική | Σχεδιασμός και Τεκμηρίωση REST API                                                    | Μερική (Level 2 & Docs) | Υιοθέτηση αρχιτεκτονικής RESTful (Level 2) για τον διαχωρισμό Frontend-Backend και παροχή αυτόματης τεκμηρίωσης (OpenAPI). Η πλήρης συμμόρφωση με το Level 3 (HATEOAS) αποτελεί στόχο μελλοντικής επέκτασης, ώστε να ενισχυθεί η δυνατότητα αυτο-ανακάλυψης (discoverability) του API.                                                                                                |
| 8 | Μη Λειτουργική | Αυθεντικοποίηση και ασφάλεια                                                          | Πλήρης (μέσω Nginx)     | JWT, RBAC, hash κωδικών, CSRF-safe endpoints, validation. Ο βαθμός ικανοποίησης για                                                                                                                                                                                                                                                                                                   |

|    |                |                                             |                                 |                                                                                                                                                                                                                                                                                                                                       |
|----|----------------|---------------------------------------------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                |                                             |                                 | υποστήριξη TLS μπορεί να θεωρηθεί Πλήρης (με χρήση Nginx - παραγωγικό περιβάλλον / deployment mode) ή Μερική (σε Dev Mode)                                                                                                                                                                                                            |
| 9  | Μη Λειτουργική | Υποστήριξη $\geq 10$ ταυτόχρονων χρηστών    | Μερική (Λειτουργική Επαλήθευση) | Δεν εκτελέστηκε επίσημος έλεγχος φόρτου (Load/Stress Test) με εργαλεία προσομοίωσης.                                                                                                                                                                                                                                                  |
| 10 | Μη Λειτουργική | Καλή απόδοση (φόρτωση < 3 sec)              | Πλήρης                          | Η απόδοση επαληθεύτηκε εμπειρικά μέσω των εργαλείων προγραμματιστή (Browser DevTools) κατά τη διάρκεια των λειτουργικών δοκιμών, όπου οι χρόνοι απόκρισης ήταν συστηματικά κάτω από 3 δευτερόλεπτα. Ωστόσο, δεν διενεργήθηκαν μετρήσεις υπό συνθήκες υψηλού φόρτου (Load Testing) για την πιστοποίηση της απόδοσης σε ακραία σενάρια. |
| 11 | Μη Λειτουργική | Αξιοπιστία, Επικύρωση και Έλεγχος Ποιότητας | Πλήρης                          | Επιτυγχάνεται μέσω αυστηρής επικύρωσης δεδομένων (Pydantic/HTML) και φιλικής διαχείρισης σφαλμάτων που ελαχιστοποιεί τα λάθη χρήστη. Η σταθερότητα του συστήματος έχει πιστοποιηθεί μέσω εκτενών δοκιμών μονάδας (Unit Tests) και ενοποίησης (Integration Tests).                                                                     |
| 14 | Μη Λειτουργική | Φορητότητα                                  | Πλήρης                          | Η εφαρμογή είναι πλήρως containerized (Docker), επιτρέποντας την εκτέλεση σε οποιοδήποτε λειτουργικό σύστημα. Επιπλέον, το Frontend είναι σχεδιασμένο με responsive design (React), εξασφαλίζοντας                                                                                                                                    |

|  |  |  |  |                                                         |
|--|--|--|--|---------------------------------------------------------|
|  |  |  |  | συμβατότητα με διαφορετικές συσκευές και φυλλομετρητές. |
|--|--|--|--|---------------------------------------------------------|

Η συνολική υλοποίηση ικανοποιεί σχεδόν πλήρως το σύνολο των λειτουργικών και μη λειτουργικών απαιτήσεων, όπως αυτές προσδιορίστηκαν στην Ενότητα 4.2. Οι δοκιμές επαλήθευσης κατέδειξαν τη σταθερότητα και τη σωστή συμπεριφορά του συστήματος σε όλα τα βασικά σενάρια χρήσης. Επιπλέον, η χρήση προτύπων και σύγχρονων τεχνολογιών (π.χ. FastAPI, Swagger, SPARQL, RDF) υποστηρίζει την επεκτασιμότητα και την εύκολη συντήρηση της πλατφόρμας στο μέλλον.

# 5

## Εγκατάσταση και επίδειξη συστήματος

Το κεφάλαιο αυτό περιλαμβάνει αρχικά τα απαραίτητα βήματα για την εγκατάσταση και την προετοιμασία της πλατφόρμας ώστε να τεθεί σε πλήρη λειτουργία. Στη συνέχεια, πραγματοποιείται η επίδειξη του συστήματος, παρουσιάζοντας το περιβάλλον εργασίας και τον τρόπο με τον οποίο ο χρήστης διαχειρίζεται, αναζητά και οπτικοποιεί τα ψηφιακά διαβατήρια στην πράξη.

### 5.1 Εγκατάσταση Συστήματος

Η εγκατάσταση του συστήματος περιλαμβάνει την παραμετροποίηση και εκτέλεση του περιβάλλοντος τόσο για το backend (API server & Semantic Store) όσο και για το frontend (διεπαφή χρήστη). Η υλοποίηση ακολουθεί αρχιτεκτονική μικροϋπηρεσιών (microservices) με χρήση Docker για εύκολη αναπαραγωγή.

#### Προαπαιτούμενα

Για την ορθή εγκατάσταση και εκτέλεση της εφαρμογής απαιτούνται:

- Docker & Docker Compose<sup>28</sup> (Συνιστάται για πλήρη λειτουργία)
- Python<sup>29</sup>  $\geq 3.11$  (Για τοπική εκτέλεση Backend)
- Poetry<sup>30</sup> (Για διαχείριση εξαρτήσεων Python)
- Node.js<sup>31</sup>  $\geq 18.0$  (Για το Frontend)
- Git<sup>32</sup>

#### 5.1.1 Αυτόματη εγκατάσταση εφαρμογής

Η εφαρμογή έχει σχεδιαστεί ώστε να είναι εύκολα μεταφέρσιμη. Παρέχονται δύο μέθοδοι:

---

<sup>28</sup> Docker & Docker Compose: <https://docs.docker.com/get-docker/>

<sup>29</sup> Python: <https://www.python.org/downloads/>

<sup>30</sup> Poetry: <https://python-poetry.org/docs/#installation>

<sup>31</sup> Node.js: <https://nodejs.org/>

<sup>32</sup> Git: <https://git-scm.com/downloads>

### Γρήγορη Εγκατάσταση με Docker (Προτείνεται)

Η μέθοδος αυτή περιλαμβάνει και τη βάση δεδομένων (PostgreSQL) και το Fuseki ως containers, οπότε δεν απαιτείται εξωτερική υπηρεσία.

1. Κλωνοποίηση του αποθετηρίου του backend της εφαρμογής στο Github:

```
git clone https://github.com/zoeleve/dpp-backend.git  
cd dpp-backend
```

2. Ρύθμιση μεταβλητών περιβάλλοντος:

Δημιουργήστε ένα αρχείο .env στον ριζικό φάκελο (dpp-backend/) με τις παρακάτω ρυθμίσεις (προσαρμόστε ανάλογα με τη βάση δεδομένων σας, π.χ. Supabase):

```
# Database (Docker Service)  
POSTGRES_USER=postgres  
POSTGRES_PASSWORD=<your_secure_password>  
POSTGRES_SERVER=db  
POSTGRES_PORT=5432  
POSTGRES_DB=dpp_db  
DATABASE_URL=postgresql+asyncpg://postgres:<your_secure_password>@db:5432/  
dpp_db  
  
# Security  
SECRET_KEY=<your_super_secret_key>  
ALGORITHM=HS256  
ACCESS_TOKEN_EXPIRE_MINUTES=30  
  
# Semantic Web  
FUSEKI_HOST=localhost  
FUSEKI_PORT=3030  
FUSEKI_DATASET_NAME=dpp
```

**Εικόνα 38:** Απόσπασμα αρχείου ρυθμίσεων περιβάλλοντος (.env) Το αρχείο περιλαμβάνει τις απαραίτητες μεταβλητές για τη σύνδεση με τη βάση δεδομένων PostgreSQL (εντός Docker container), τις παραμέτρους ασφαλείας για την κρυπτογράφηση των JWT tokens, καθώς και τις ρυθμίσεις διασύνδεσης με τον σημασιολογικό γράφο Apache Jena Fuseki.

3. Εκτέλεση με Docker

Αυτή η μέθοδος εκκινεί αυτόματα τα εξής services: Frontend, Backend, Fuseki (Semantic Store), Nginx, Prometheus και το Grafana.

```
docker-compose up -d --build
```

Το Frontend της εφαρμογής είναι υλοποιημένο με React και Vite.

### **Εκτέλεση μέσω Docker (Προτείνεται)**

Το Frontend περιλαμβάνεται στο docker-compose.yaml και εκκινείται αυτόματα μαζί με το Backend (δείτε πρώτη εναλλακτική εγκατάστασης backend μέσω Docker προηγουμένως).

Πρόσβαση: <http://localhost> (μέσω Nginx Reverse Proxy).

### **Αρχικοποίηση Δεδομένων (Seeding)**

Για να γεμίσετε τη βάση με πρότυπα δεδομένα (AAS-compliant) και να συγχρονίσετε το σημασιολογικό γράφο:

#### **Μέσω Docker:**

```
docker-compose exec backend python app/seed_aas_data.py
```

## **5.1.2 Χειροκίνητη εγκατάσταση εφαρμογής**

### **5.1.2.1 Χειροκίνητη εγκατάσταση backend**

Για προγραμματιστές που επιθυμούν να επεξεργαστούν τον πηγαίο κώδικα χωρίς τη χρήση Docker για το Backend, ακολουθούνται τα εξής βήματα:

```
git clone https://github.com/zoeleve/dpp-backend.git
cd dpp-backend
python -m venv venv
source venv/bin/activate # (Windows: venv\Scripts\activate)
```

1. Εγκατάσταση εξαρτήσεων:

```
pip install poetry
poetry install
```

2. Εκκίνηση Fuseki (μέσω Docker):

```
docker-compose up -d fuseki
```

3. Εκτέλεση Server:

```
poetry run uvicorn app.main:app --reload
```

### **Αρχικοποίηση Δεδομένων (Seeding)**

Για να γεμίσετε τη βάση με πρότυπα δεδομένα (AAS-compliant) και να συγχρονίσετε το σημασιολογικό γράφο:

**Τοπικά (localhost):**

```
poetry run python app/seed_aas_data.py
```

### 5.1.2.2 Χειροκίνητη εγκατάσταση frontend

Για ανάπτυξη του Frontend ξεχωριστά:

1. Κλωνοποίηση του repo:

```
git clone https://github.com/zoeleve/dpp-frontend.git  
cd dpp-frontend
```

2. Εγκατάσταση εξαρτήσεων:

```
npm install
```

3. Η εφαρμογή χρειάζεται να γνωρίζει πού βρίσκεται το Backend API. Δημιουργήστε ένα αρχείο .env στον ριζικό φάκελο του project (δίπλα στο package.json). Προσθέστε την παρακάτω γραμμή, αντικαθιστώντας το URL με τη διεύθυνση του backend σας αν διαφέρει:

```
VITE_API_URL=http://localhost:8000
```

4. Εκκίνηση εφαρμογής:

```
npm run dev
```

Μόλις ολοκληρωθεί η εκκίνηση της εφαρμογής, θα δείτε στο τερματικό τη διεύθυνση πρόσβασης. Συνήθως είναι η: <http://localhost:5173> (Το Vite χρησιμοποιεί την πόρτα 5173 ως προεπιλογή. Αν αυτή η πόρτα χρησιμοποιείται ήδη από άλλη εφαρμογή, το Vite θα επιλέξει αυτόματα την επόμενη διαθέσιμη, π.χ. 5174, και θα το αναφέρει στο τερματικό). Ανοίξτε αυτό τον σύνδεσμο στον φυλλομετρητή (browser) σας για να δείτε και χρησιμοποιήσετε την εφαρμογή.

## 5.2 Επίδειξη συστήματος

Στην παρούσα ενότητα πραγματοποιείται η επίδειξη του συστήματος με βάση συγκεκριμένα σενάρια χρήσης, τα οποία συνοδεύονται από σχετικά στιγμιότυπα (screenshots) της διεπαφής χρήστη της εφαρμογής. Τα σενάρια αυτά καλύπτουν ορισμένες από τις πιο βασικές περιπτώσεις χρήσης του συστήματος, προσφέροντας μια ρεαλιστική απεικόνιση της λειτουργίας του.

### 5.2.1 Σενάριο 1: Εγγραφή και Σύνδεση Χρήστη (Ρόλος Viewer)

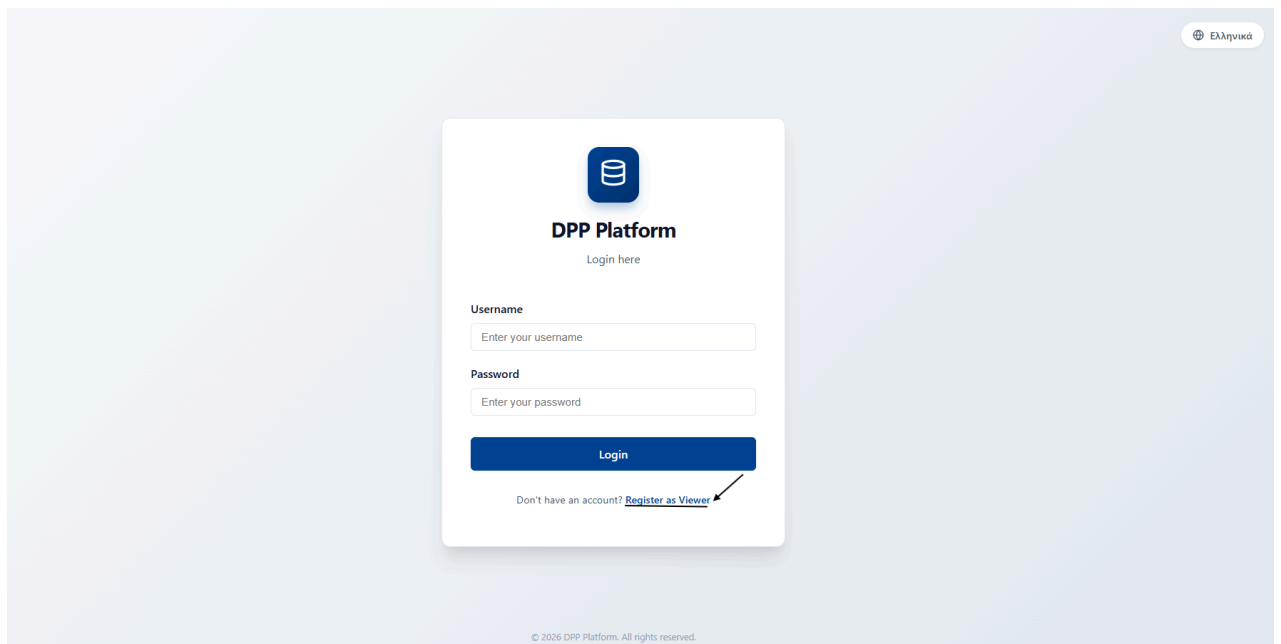
Το πρώτο σενάριο αφορά τη διαδικασία εισαγωγής ενός νέου χρήστη στο σύστημα. Συγκεκριμένα, εξετάζεται η εγγραφή ενός χρήστη με τον ρόλο του "Επισκέπτη" (Viewer), ο οποίος ενδιαφέρεται να αναζητήσει πληροφορίες προϊόντων χωρίς να έχει δικαιώματα καταχώρησης νέων δεδομένων.

#### Προϋποθέσεις:

- Ο χρήστης έχει πλοηγηθεί στην αρχική σελίδα της εφαρμογής και επιλέγει την σύνδεση/εγγραφή από το κάτω μέρος της εμφανιζόμενης φόρμας (δείτε Εικόνα 39).
- Δεν υπάρχει ενεργή συνεδρία χρήστη (logged-out state).

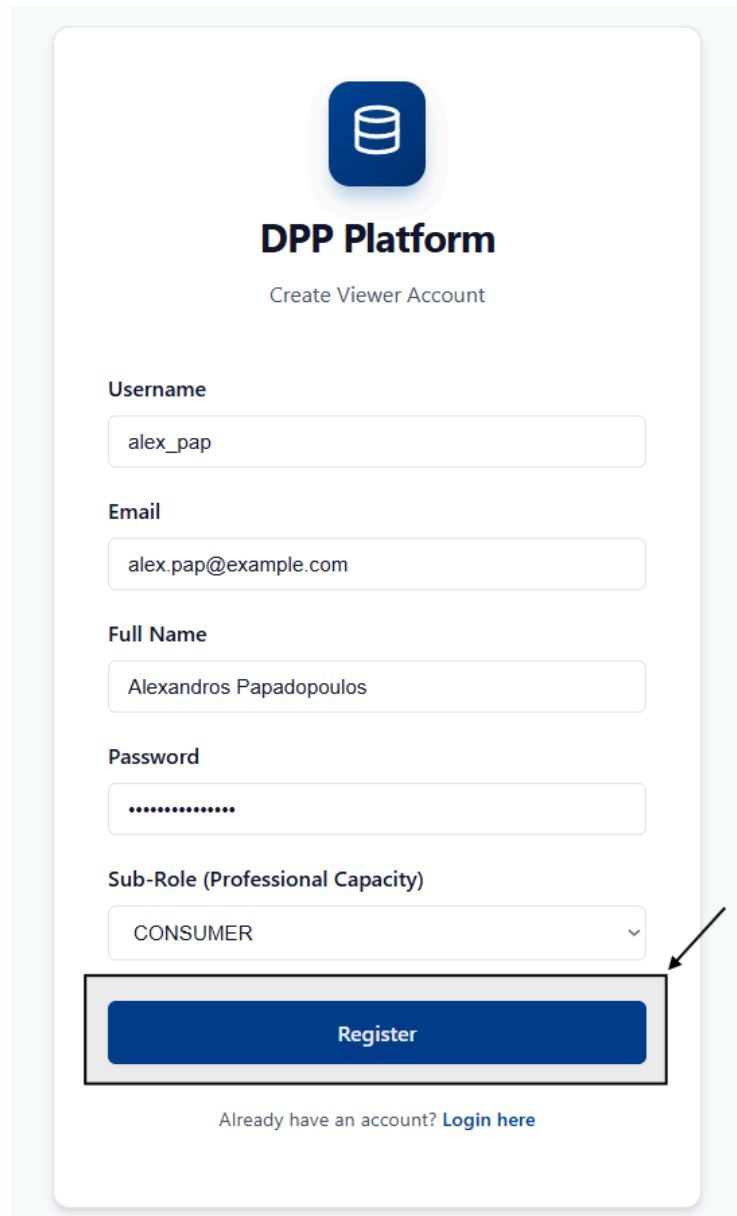
#### Ροή Ενεργειών:

1. Ο χρήστης επιλέγει την επιλογή εγγραφής από την οθόνη εισόδου (Εικόνα 39).



Εικόνα 39: Οθόνη εισόδου (Αρχική Σελίδα Εφαρμογής)

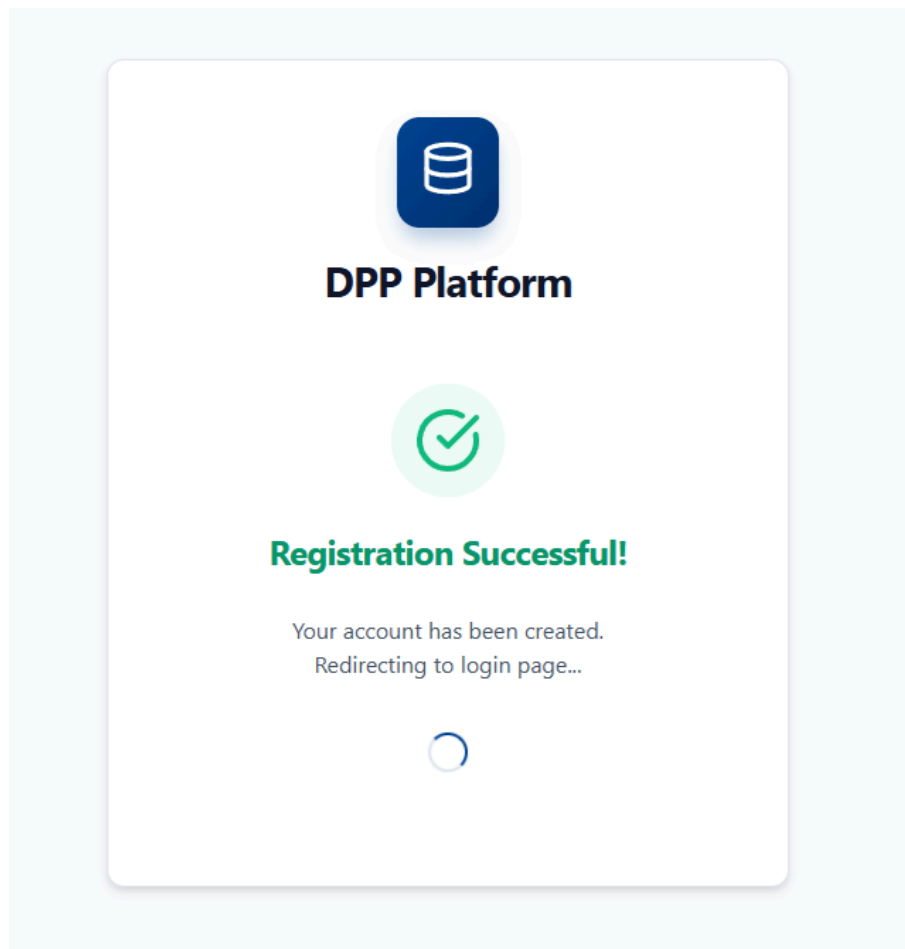
2. Στη φόρμα εγγραφής (Εικόνα 40), ο χρήστης καλείται να συμπληρώσει τα προσωπικά του στοιχεία (Όνομα Χρήστη, Email, Κωδικό Πρόσβασης και Επαγγελματική Ιδιότητα).



The image shows a registration form for the 'DPP Platform'. At the top, there is a blue database icon and the text 'DPP Platform' followed by 'Create Viewer Account'. The form contains several input fields: 'Username' with the value 'alex\_pap', 'Email' with 'alex.pap@example.com', 'Full Name' with 'Alexandros Papadopoulos', and 'Password' with masked characters. Below these is a 'Sub-Role (Professional Capacity)' dropdown menu set to 'CONSUMER'. A blue 'Register' button is at the bottom, highlighted with a black border and an arrow pointing to it from the right. Below the button, there is a link: 'Already have an account? [Login here](#)'.

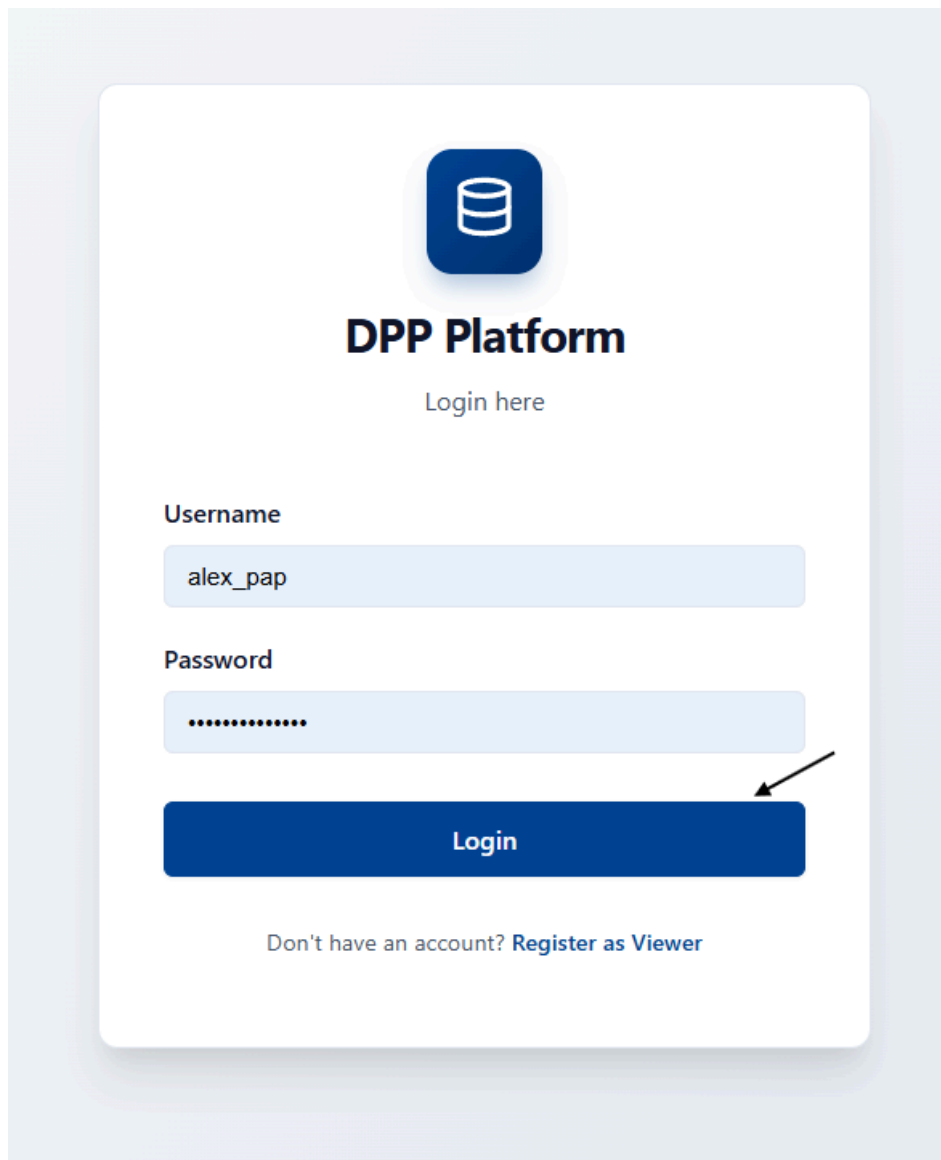
Εικόνα 40: Η διεπαφή εγγραφής νέου χρήστη με ρόλο Επισκέπτη (Viewer)

3. Με την υποβολή της φόρμας, το σύστημα επικυρώνει τα στοιχεία και, εφόσον είναι έγκυρα, εμφανίζει μήνυμα επιτυχίας ("Registration Successful"). Όπως φαίνεται στην Εικόνα 41, το σύστημα ενημερώνει τον χρήστη και τον ανακατευθύνει αυτόματα στη σελίδα εισόδου μετά από σύντομο χρονικό διάστημα, βελτιώνοντας την εμπειρία χρήστη.



*Εικόνα 41: Επιτυχής ολοκλήρωση εγγραφής και αυτόματη ανακατεύθυνση.*

4. Τέλος, ο χρήστης εισάγει τα διαπιστευτήριά του στη σελίδα εισόδου για να αποκτήσει πρόσβαση στην πλατφόρμα (Εικόνα 42).





## DPP Platform

Login here

Username

alex\_pap

Password

.....

Login

Don't have an account? [Register as Viewer](#)

*Εικόνα 42: Εισαγωγή διαπιστευτηρίων για σύνδεση στο σύστημα.*

**Αποτέλεσμα:** Ο χρήστης συνδέεται επιτυχώς και μεταφέρεται στον Πίνακα Ελέγχου (Dashboard). Λόγω του ρόλου του (Viewer), το μενού πλοήγησης προσαρμόζεται δυναμικά, παρέχοντας πρόσβαση μόνο στις λειτουργίες αναζήτησης και προβολής, ενώ οι επιλογές δημιουργίας DPP παραμένουν κρυφές (Εικόνα 43).

**DPP Platform** Dashboard

MENU: Dashboard, SPARQL Query

ACCOUNT: My Profile

Language: Ελληνικά

User: Alexandros Papadopoulos... Viewer

Logout

**Summary:**

- Total DPPs: 18
- Published: 6
- Drafts: 12

**Search:** Simple | Advanced

Search by keywords... **Search**

| ID   | TITLE                            | PRODUCT ID                      | STATUS    | ACTIONS |
|------|----------------------------------|---------------------------------|-----------|---------|
| #160 | Smart Digital Dosing Pump        | did:pump:smart-dose-ddc         | Published |         |
| #159 | Recycled Ocean Plastic Jacket    | did:textile:jacket:ocean-blue-1 | Published |         |
| #158 | EcoSmart Phone Pro 5             | did:tech:phone:pro5-128gb       | Published |         |
| #157 | EV Battery Module 75kWh          | did:gba:battery:75kwh-nmc-2024  | Published |         |
| #156 | High-Efficiency Servo Motor 1FK7 | did:siemens:motor:1FK7063-2AF71 | Published |         |
| #151 | Apple iPad Pro M4                | APL-IPD-M4-512                  | Published |         |

Εικόνα 43: Ο Πίνακας Ελέγχου για τον ρόλο Viewer, με περιορισμένες επιλογές πλοήγησης.

### 5.2.2 Σενάριο 2: Επισκόπηση και Πλοήγηση στο Σύστημα (Ρόλος User)

Το δεύτερο σενάριο περιγράφει την αρχική αλληλεπίδραση του χρήστη με το κεντρικό περιβάλλον της πλατφόρμας και την εξοικείωσή του με τα εργαλεία διαχείρισης.

**Σημείωση Ασφαλείας:** Η πλατφόρμα ακολουθεί αυστηρή πολιτική εγγραφών. Η δημόσια εγγραφή (Public Registration) επιτρέπει τη δημιουργία λογαριασμών αποκλειστικά με τον ρόλο "Viewer".

Η δημιουργία χρηστών με αναβαθμισμένα δικαιώματα (ρόλος "User" ή "Admin"), οι οποίοι έχουν τη δυνατότητα δημιουργίας και διαχείρισης DPPs, πραγματοποιείται μόνο από τον Διαχειριστή (Admin) μέσω του ειδικού περιβάλλοντος διαχείρισης χρηστών. Επιπλέον, μόνο ένας διαχειριστής μπορεί να αλλάξει τον ρόλο ενός χρήστη (πχ. από "Viewer" να τον μετατρέψει σε "User"). Αυτή η προσέγγιση διασφαλίζει ότι μόνο πιστοποιημένοι κατασκευαστές και συνεργάτες μπορούν να εισάγουν δεδομένα στο σύστημα, προστατεύοντας την ακεραιότητα του μητρώου προϊόντων.

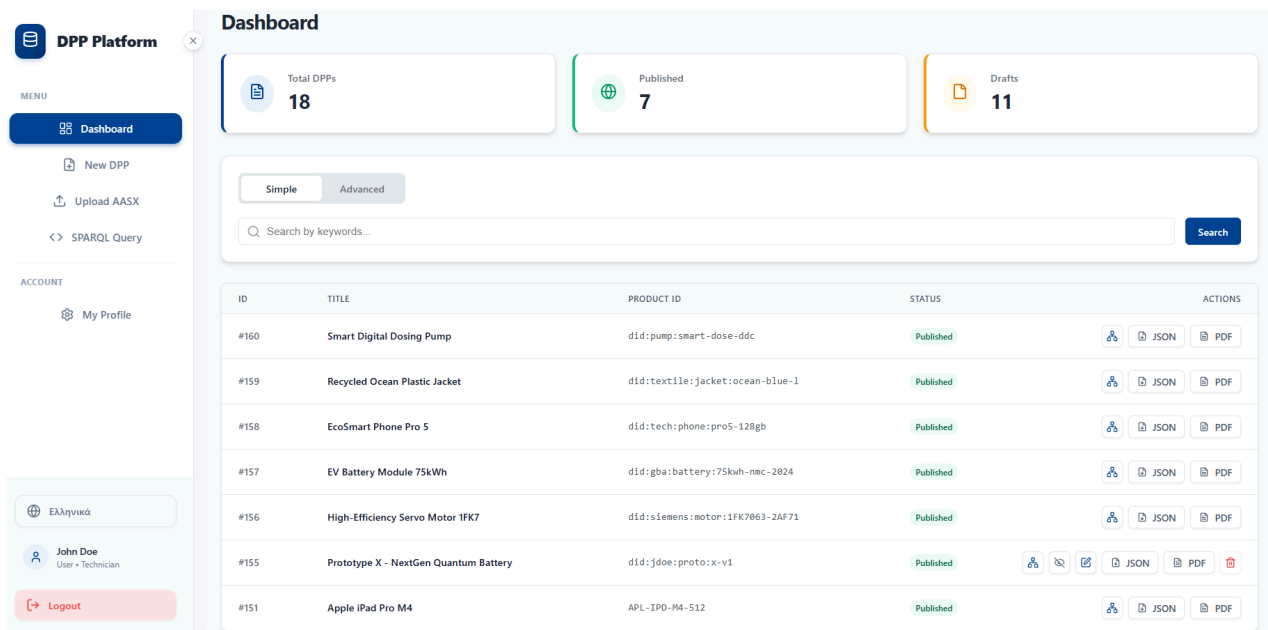
#### Προϋποθέσεις:

- Ο χρήστης έχει συνδεθεί επιτυχώς στην πλατφόρμα.

#### Ροή Ενεργειών:

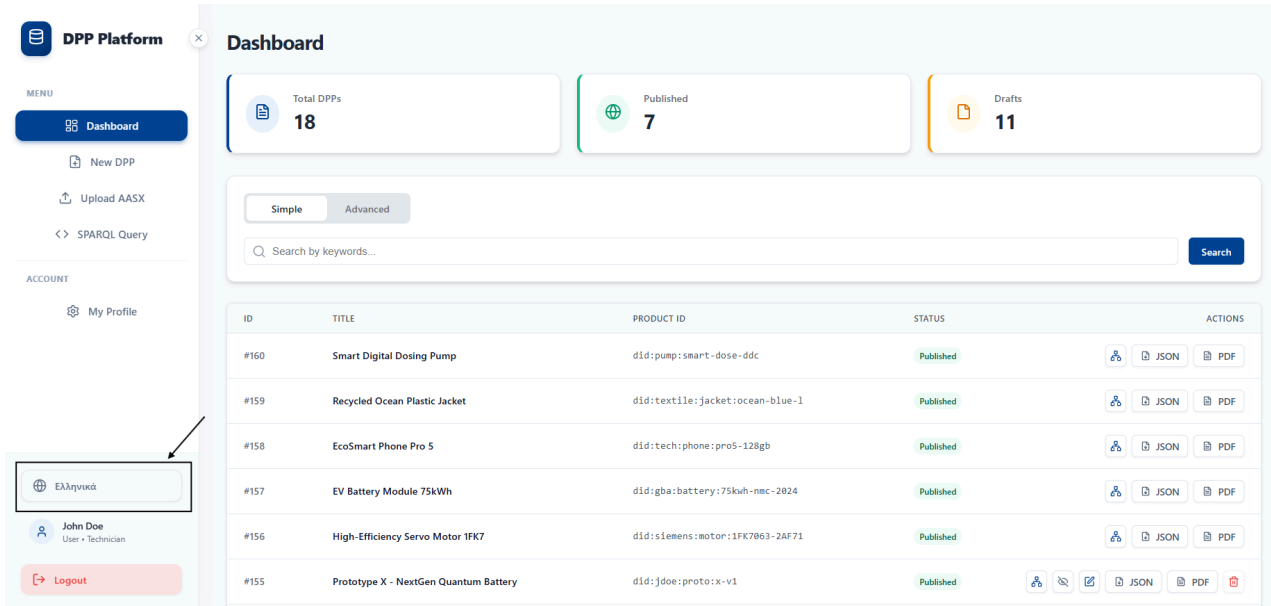
1. Με την είσοδο στο σύστημα, ο χρήστης (Registered User) μεταφέρεται στον κεντρικό Πίνακα Ελέγχου, ο οποίος προσφέρει μια συγκεντρωτική απεικόνιση της δραστηριότητάς του (δείτε Εικόνα 42). Συγκεκριμένα, ο χρήστης παρατηρεί:

- **Στατιστικά Στοιχεία (Cards):** Στο πάνω μέρος εμφανίζονται αριθμητικοί δείκτες για το σύνολο των διαθέσιμων διαβατηρίων (Total DPPs), τον αριθμό των δημοσιευμένων (Published) και τον αριθμό των πρόχειρων (Drafts).
- **Λειτουργίες Αναζήτησης:** Παρέχεται δυνατότητα επιλογής μεταξύ απλής αναζήτησης με λέξεις-κλειδιά και προηγμένης αναζήτησης (Advanced), διευκολύνοντας τον εντοπισμό συγκεκριμένων προϊόντων.
- **Πίνακας Διαχείρισης:** Στον κεντρικό πίνακα παρατίθενται τα ψηφιακά διαβατήρια με πληροφορίες όπως ο Τίτλος, το Product ID και το Status.
- **Πλευρική Γραμμή Πλοήγησης (Sidebar):** Επιτρέπει την ταχεία μετάβαση στις λειτουργίες δημιουργίας νέου DPP, μεταφόρτωσης αρχείων AASX και εκτέλεσης ερωτημάτων SPARQL.

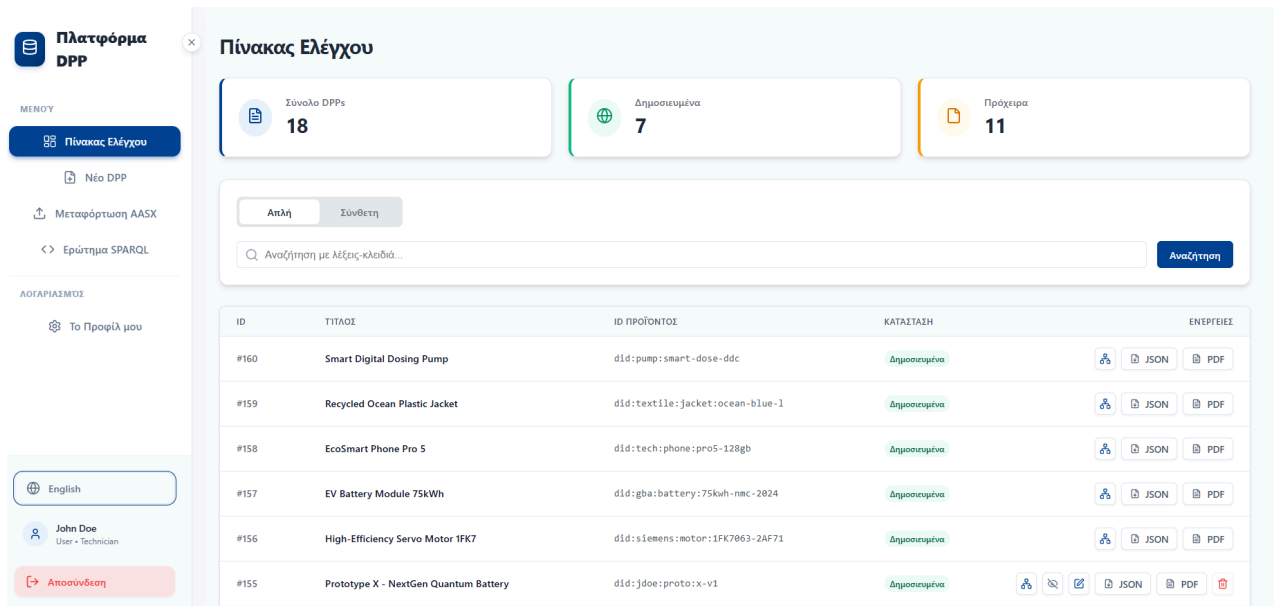


Εικόνα 44: Κεντρική διεπαφή χρήστη (ρόλος User) του συστήματος (Dashboard) με συγκεντρωτική απεικόνιση των DPPs και εργαλεία διαχείρισης.

- Ο χρήστης εστιάζει στη στήλη των ενεργειών (Actions), η οποία περιλαμβάνει:
  - **Semantic Graph View:** Ενεργοποιεί την οπτικοποίηση του διαβατηρίου ως σημασιολογικό γράφο.
  - **Visibility Toggle (Υδρόγειος/Μάτι):** Επιτρέπει την εναλλαγή της κατάστασης μεταξύ "Published" και "Draft", ελέγχοντας την εξωτερική ορατότητα.
  - **Edit (Μολύβι):** Επιτρέπει την τροποποίηση των στοιχείων του DPP.
  - **Export Functions:** Κουμπιά για την εξαγωγή των δεδομένων σε JSON (για διαλειτουργικότητα) και PDF (για τεκμηρίωση).
  - **Delete (Κάδος):** Για τη διαγραφή της εγγραφής.
- Για την ενίσχυση της προσβασιμότητας, ο χρήστης επιλέγει να αλλάξει τη γλώσσα της διεπαφής (Εικόνα 43). Εντοπίζει τον ειδικό επιλογέα στο κάτω μέρος της πλευρικής γραμμής και επιλέγει "Ελληνικά". Η αλλαγή εφαρμόζεται δυναμικά σε όλα τα στοιχεία του συστήματος (μενού, τίτλοι, μηνύματα), όπως φαίνεται στην Εικόνα 45.



Εικόνα 45: Διαδικασία επιλογής της ενότητας "Ελληνικά" από το πλευρικό μενού πλοήγησης.



Εικόνα 46: Δυνατότητα εναλλαγής γλώσσας (Localization) της πλατφόρμας μεταξύ Ελληνικών και Αγγλικών.

**Αποτέλεσμα:** Ο χρήστης έχει αποκτήσει πλήρη εποπτεία των διαθέσιμων εργαλείων και έχει προσαρμόσει το περιβάλλον εργασίας, όντας έτοιμος για τη διαχείριση των δεδομένων του.

### 5.2.3 Σενάριο 3: Δημιουργία και Διαχείριση Ψηφιακού Διαβατηρίου (Ρόλος User)

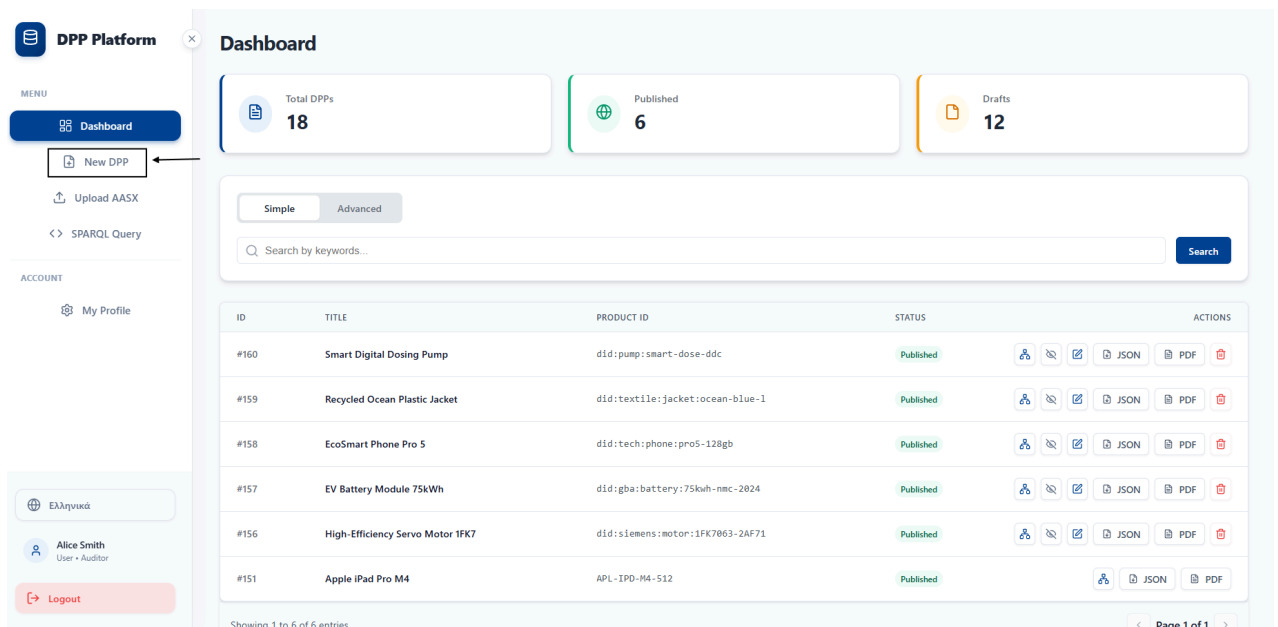
Το τρίτο σενάριο εστιάζει στη βασική λειτουργία της πλατφόρμας “DPP Platform”: τη δημιουργία ενός νέου Ψηφιακού Διαβατηρίου Προϊόντος (DPP). Εξετάζεται η περίπτωση ενός χρήστη με ρόλο "User" (π.χ. Ελεγκτής), ο οποίος έχει δικαιώματα δημιουργίας, επεξεργασίας και δημοσίευσης δεδομένων.

### Προϋποθέσεις:

- Ο χρήστης έχει συνδεθεί επιτυχώς στην πλατφόρμα με ρόλο "User".
- Ο χρήστης βρίσκεται στον Πίνακα Ελέγχου (Dashboard).

### Ροή Ενεργειών:

1. Από το μενού πλοήγησης, ο χρήστης επιλέγει "New DPP" (Εικόνα 47). Το σύστημα εμφανίζει τη φόρμα δημιουργίας, η οποία είναι σχεδιασμένη να υποστηρίξει τη δομή του προτύπου AAS (Εικόνα 48).



Εικόνα 47: Εκκίνηση της διαδικασίας δημιουργίας νέου διαβατηρίου από τον Πίνακα Ελέγχου.

The screenshot displays the 'DPP Platform' interface. On the left, a sidebar menu contains 'Dashboard', 'New DPP' (highlighted), 'Upload AASX', 'SPARQL Query', and 'My Profile'. The main content area is titled 'Create New DPP' and includes a 'Back to Dashboard' link. The form fields are as follows:

- Title \***: A text input field.
- Product ID (GlobalAssetId) \***: A text input field with a hint 'e.g. UUID, GTIN, DID' and a refresh icon.
- Manufacturer**: A text input field.
- Model Number**: A text input field.
- Serial Number**: A text input field.
- Production Date**: A date picker showing 'mm/dd/yyyy'.
- Description / Extra Data**: A text area with a placeholder 'This will be stored in attributes for now.'
- AAS Submodels**: A section with a '+ Add Submodel' button and a note 'No submodels added. Add one to structure your data (e.g. CarbonFootprint).'

At the bottom of the form is a large blue 'Create DPP' button. The user profile 'Alice Smith, User • Auditor' and a 'Logout' button are visible in the bottom left of the sidebar.

Εικόνα 48: Η φόρμα δημιουργίας νέου DPP, με πεδία για τα βασικά χαρακτηριστικά του προϊόντος.

- Ο χρήστης συμπληρώνει τα βασικά στοιχεία του προϊόντος (Τίτλος, Product ID, Κατασκευαστής, Ημερομηνία Παραγωγής). Στη συνέχεια, χρησιμοποιεί τον δυναμικό επεξεργαστή υπο-μοντέλων (AAS Submodel Editor) για να προσθέσει εξειδικευμένα δεδομένα, όπως το "Carbon Footprint" (Εικόνα 49). Ιδιαίτερη έμφαση δόθηκε στην αρθρωτή (modular) αρχιτεκτονική της φόρμας, η οποία επιτρέπει στον χρήστη να προσθέτει δυναμικά πολλαπλά υπο-μοντέλα (Submodels), όπως περιβαλλοντικά αποτυπώματα, τεχνικά εγχειρίδια ή δεδομένα κυκλικότητας. Αυτή η λειτουργία διασφαλίζει την πληρότητα της πληροφορίας και την ευελιξία του συστήματος, επιτρέποντας την εισαγωγή δομημένων δεδομένων χωρίς να απαιτούνται αλλαγές στο σχήμα της βάσης δεδομένων.

**DPP Platform**

MENU

- Dashboard
- New DPP**
- Upload AASX
- SPARQL Query

ACCOUNT

- My Profile

Ελληνικά

Alice Smith  
User • Auditor

Logout

### Create New DPP

Title \* High-Efficiency Motor M-200

Product ID (GlobalAssetId) \* urn:uuid:1234-5678-90ab-cdef

Manufacturer EcoMotors Ltd

Model Number M-200-X

Serial Number SN-2024-8892

Production Date 12/05/2024

Description / Extra Data  
Three-phase induction motor with IE4 efficiency class. Designed for industrial automation applications.

**AAS Submodels**

+ Add Submodel

**Submodel #1**

CarbonFootprint https://admin-shell.io/id/carbon\_footprint

Elements

|                   |                |   |
|-------------------|----------------|---|
| CO2_Equivalent    | 150 kg         | x |
| CalculationMethod | ISO 14067      | x |
| LifecycleStage    | Cradle-to-Gate | x |

+ Add Element

Εικόνα 49: Προσθήκη υπο-μοντέλων AAS (Submodels) για την καταγραφή εξειδικευμένων δεδομένων.

- Με την ολοκλήρωση της καταχώρησης, ο χρήστης πατάει το κουμπί “Create DPP” και υποβάλλει τη φόρμα (Εικόνα 50). Το σύστημα αποθηκεύει τα δεδομένα στην PostgreSQL και ταυτόχρονα εκκινεί τη διαδικασία συγχρονισμού με το σημασιολογικό αποθετήριο (Fuseki). Εμφανίζεται μήνυμα επιτυχίας (Εικόνα 51) και ο χρήστης ανακατευθύνεται στο Dashboard.

The screenshot displays the 'AAS Submodels' interface. At the top, there is a header with the 'AAS Submodels' logo and a '+ Add Submodel' button. Below this, two submodels are listed:

- Submodel #1**: Contains a 'Carbon Footprint' element with a value of 'CarbonFootprint'. Below it, a table of elements is shown:

| Element           | Value          | Action |
|-------------------|----------------|--------|
| CO2_Equivalent    | 150 kg         | ×      |
| CalculationMethod | ISO 14067      | ×      |
| LifecycleStage    | Cradle-to-Gate | ×      |

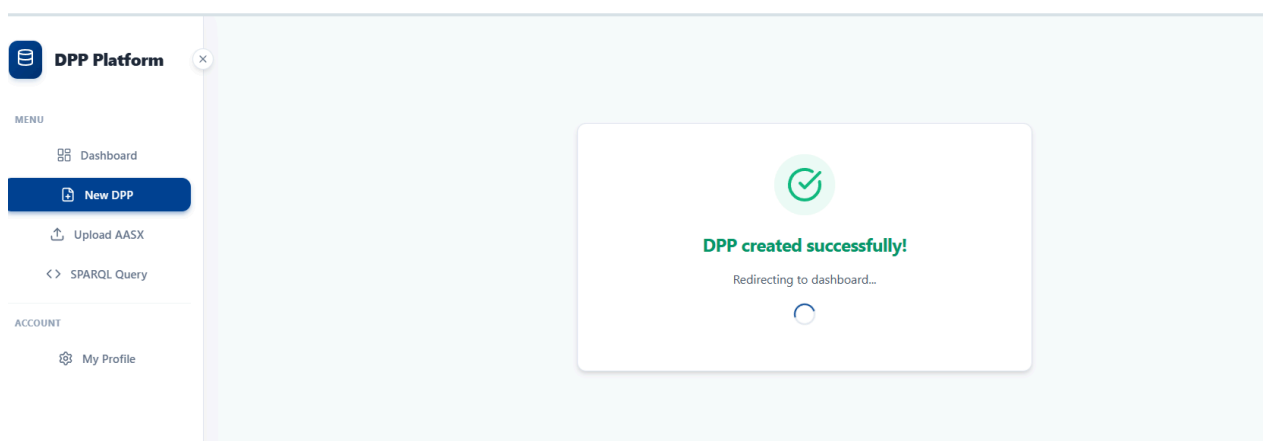
A '+ Add Element' button is located below the table.
- Submodel #2**: Contains a 'Technical Data' element with a value of '0173-1#01-AFZ615#016'. Below it, a table of elements is shown:

| Element | Value    | Action |
|---------|----------|--------|
| Power   | 5.5 kW   | ×      |
| Voltage | 400 V    | ×      |
| Speed   | 1450 rpm | ×      |

A '+ Add Element' button is located below the table.

At the bottom of the interface, a large blue button labeled 'Create DPP' is highlighted with a black box and an arrow pointing to it from the right.

Εικόνα 50: Χειροκίνητη εισαγωγή δεδομένων για τη δημιουργία νέου DPP.

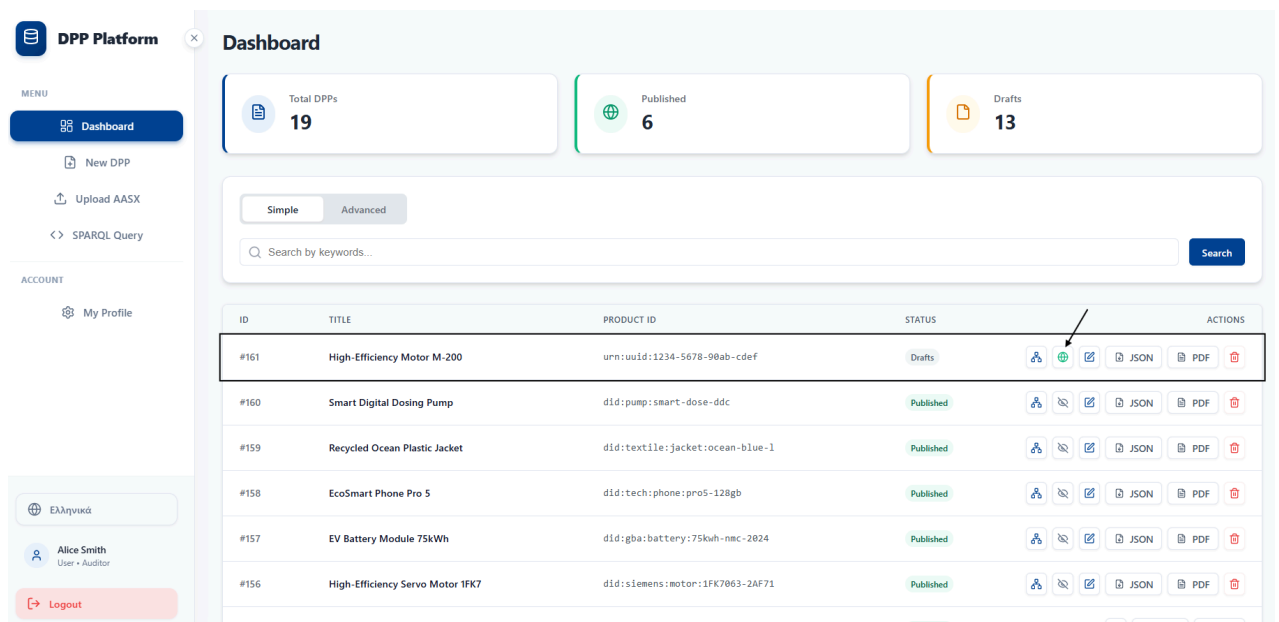


Εικόνα 51: Επιβεβαίωσης επιτυχούς καταχώρησης του νέου DPP, όπου προβάλλεται το αντίστοιχο μήνυμα ολοκλήρωσης και η ενημέρωση για την αυτόματη ανακατεύθυνση του χρήστη στο Dashboard.

4. Στο Dashboard, το νέο DPP εμφανίζεται στη λίστα με κατάσταση "Draft" (Πρόχειρο). Ως ιδιοκτήτης του DPP, ο χρήστης αποκτά πρόσβαση σε μια πλήρη εργαλειοθήκη διαχείρισης, η οποία περιλαμβάνει:

- Οπτικοποίηση (Semantic Graph): Μέσω του εικονιδίου δικτύου, ο χρήστης μπορεί να προβάλει τη σημασιολογική αναπαράσταση των δεδομένων σε μορφή γράφου.
- Δημοσίευση (Publish): Μέσω του εικονιδίου της υδρογείου, ο χρήστης αλλάζει την κατάσταση σε "Published", καθιστώντας το διαβατήριο ορατό σε τρίτους.
- Εξαγωγή (Export): Δυνατότητα λήψης των δεδομένων σε δομημένη μορφή JSON (για διαλειτουργικότητα) ή σε αρχείο PDF (για εκτύπωση/αναφορά).
- Διαγραφή (Delete): Για την αφαίρεση της εγγραφής από το σύστημα.

Όπως φαίνεται στην Εικόνα 52, οι επιλογές αυτές είναι διαθέσιμες στη στήλη "Actions", παρέχοντας άμεσο έλεγχο στον κύκλο ζωής του προϊόντος.



**Εικόνα 52: Εμφάνιση του νέου DPP στον Πίνακα Ελέγχου. Διακρίνονται τα εργαλεία διαχείρισης (Οπτικοποίηση, Δημοσίευση, Εξαγωγή, Διαγραφή) που είναι διαθέσιμα στον ιδιοκτήτη.**

5. Ο χρήστης επιλέγει να δημοσιεύσει το DPP πατώντας το εικονίδιο "Publish" (Υδρόγειος) (Εικόνα 52). Το σύστημα ενημερώνει την κατάσταση σε "Published", καθιστώντας το διαβατήριο ορατό στους Viewers (Εικόνα 53).

The screenshot shows the DPP Platform dashboard. At the top, there are three summary cards: 'Total DPPs' with a count of 19, 'Published' with a count of 7, and 'Drafts' with a count of 12. Below these is a search bar with a 'Search' button. The main content area is a table of DPPs. The table has columns for ID, TITLE, PRODUCT ID, STATUS, and ACTIONS. The first row is highlighted, showing a 'High-Efficiency Motor M-200' with a 'Published' status. The table lists 7 entries in total. On the left side, there is a sidebar with a 'MENU' section containing 'Dashboard', 'New DPP', 'Upload AASX', and 'SPARQL Query'. Below the menu is an 'ACCOUNT' section with 'My Profile' and a 'Logout' button. At the bottom left, there is a language selector set to 'Ελληνικά' and a user profile for 'Alice Smith' (User - Auditor).

| ID   | TITLE                            | PRODUCT ID                      | STATUS    | ACTIONS                                     |
|------|----------------------------------|---------------------------------|-----------|---------------------------------------------|
| #161 | High-Efficiency Motor M-200      | urn:uuid:1234-5678-90ab-cdef    | Published | [Icons: Share, Download, JSON, PDF, Delete] |
| #160 | Smart Digital Dosing Pump        | did:pump:smart-dose-ddc         | Published | [Icons: Share, Download, JSON, PDF, Delete] |
| #159 | Recycled Ocean Plastic Jacket    | did:textile;jacket:ocean-blue-1 | Published | [Icons: Share, Download, JSON, PDF, Delete] |
| #158 | EcoSmart Phone Pro 5             | did:tech:phone:pro5-128gb       | Published | [Icons: Share, Download, JSON, PDF, Delete] |
| #157 | EV Battery Module 75kWh          | did:gba:battery:75kwh-nmc-2024  | Published | [Icons: Share, Download, JSON, PDF, Delete] |
| #156 | High-Efficiency Servo Motor 1FK7 | did:siemens:motor:1FK7063-2AF71 | Published | [Icons: Share, Download, JSON, PDF, Delete] |
| #151 | Apple iPad Pro M4                | APL-IPD-M4-512                  | Published | [Icons: Share, Download, JSON, PDF, Delete] |

Showing 1 to 7 of 7 entries

Page 1 of 1

**Εικόνα 53: Ενημέρωση κατάστασης σε “Published”**

**Αποτέλεσμα:** Το Ψηφιακό Διαβατήριο έχει δημιουργηθεί, αποθηκευτεί δομημένα βάσει του προτύπου AAS και δημοσιευτεί επιτυχώς. Τα δεδομένα είναι πλέον προσβάσιμα τόσο μέσω του Dashboard όσο και μέσω σημασιολογικών ερωτημάτων (SPARQL).

#### 5.2.4 Σενάριο 4: Οπτικοποίηση Σημασιολογικού Γράφου (Semantic Graph Visualization)

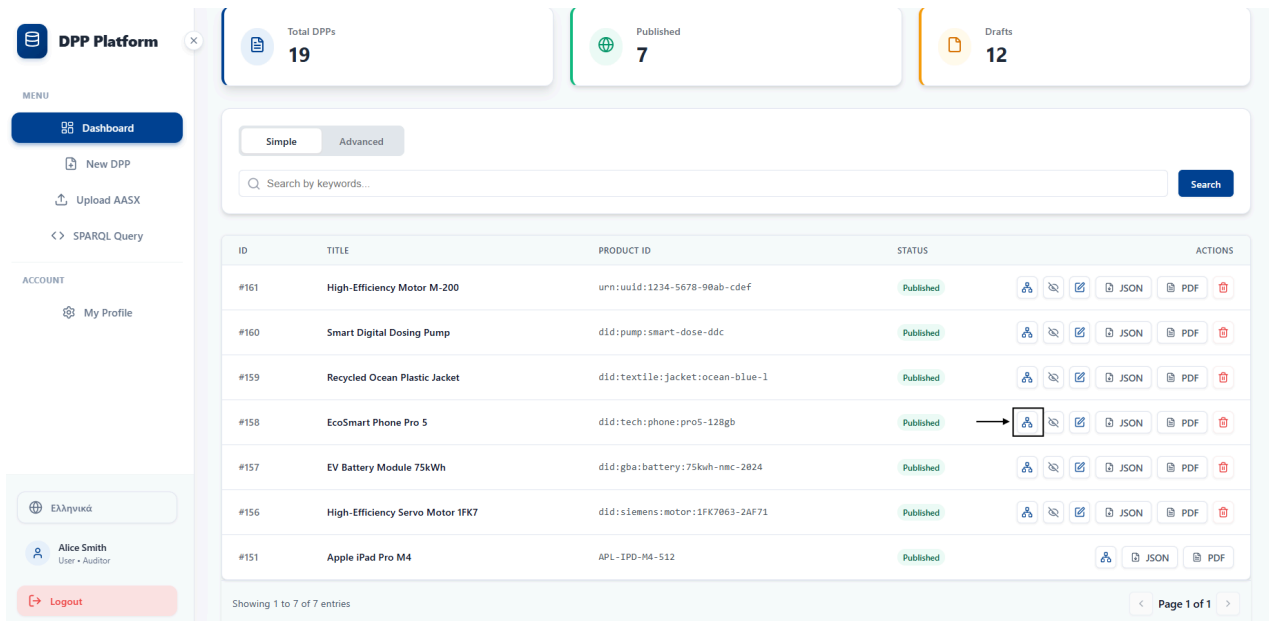
Το τέταρτο σενάριο αναδεικνύει την καινοτομία της πλατφόρμας στη διαχείριση σημασιολογικών δεδομένων. Επιδεικνύεται η δυνατότητα του συστήματος να μετατρέπει δυναμικά τα δεδομένα ενός DPP σε γράφο γνώσης (Knowledge Graph) και να τα οπτικοποιεί διαδραστικά.

##### Προϋποθέσεις:

- Υπάρχει τουλάχιστον ένα καταχωρημένο DPP στο σύστημα (π.χ. αυτό που δημιουργήθηκε στο Σενάριο 3).
- Ο χρήστης βρίσκεται στον Πίνακα Ελέγχου (Dashboard).

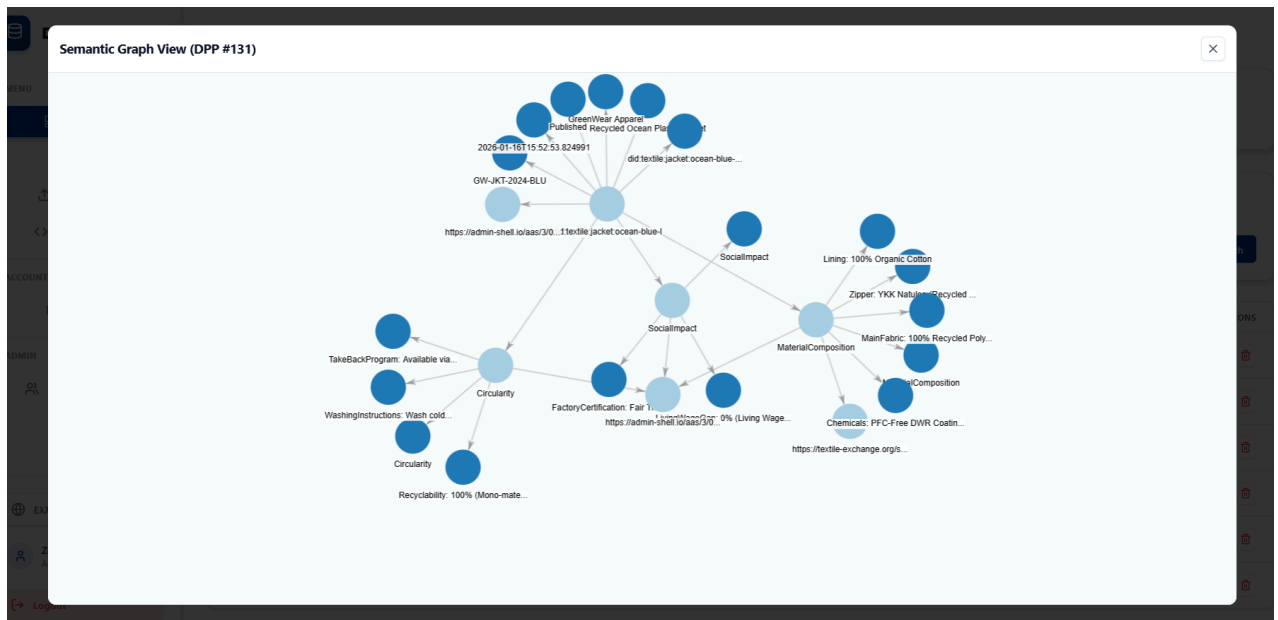
##### Ροή Ενεργειών:

- Ο χρήστης εντοπίζει το επιθυμητό DPP στη λίστα και επιλέγει το εικονίδιο "View Semantic Graph" (εικονίδιο δικτύου) από τη στήλη ενεργειών (Εικόνα 54).



Εικόνα 54: Επιλογή εικονιδίου δικτύου για προβολή του οπτικοποιημένου γράφου

2. Το σύστημα εκτελεί μια ασύγχρονη κλήση στο backend, το οποίο ανακτά τα RDF δεδομένα απευθείας από το σημασιολογικό αποθετήριο (Apache Jena Fuseki) μέσω SPARQL ερωτημάτων, διασφαλίζοντας ότι η οπτικοποίηση αντικατοπτρίζει την τρέχουσα κατάσταση του Γράφου Γνώσης.
3. Εμφανίζεται ένα αναδυόμενο παράθυρο (Modal) που περιέχει την οπτικοποίηση του γράφου (Εικόνα 55).



Εικόνα 55: Διαδραστική οπτικοποίηση του σημασιολογικού γράφου. Οι κόμβοι αναπαριστούν τις οντότητες (Προϊόν, Υπο-μοντέλα) και οι ακμές τις μεταξύ τους σχέσεις.

4. Ο χρήστης μπορεί να αλληλεπιδράσει με τον γράφο με τους εξής τρόπους:

- Να εστιάσει (zoom in/out) για να δει λεπτομέρειες.
- Να σύρει (drag) τους κόμβους για να αναδιατάξει τη δομή.
- Να διακρίνει ξεκάθαρα τη σχέση μεταξύ του κεντρικού προϊόντος (Product ID) και των υπο-μοντέλων του (π.χ. CarbonFootprint), καθώς και τις τιμές των επιμέρους στοιχείων.

**Αποτέλεσμα:** Ο χρήστης αποκτά μια εποπτική και κατανοητή εικόνα της δομής της πληροφορίας, επιβεβαιώνοντας την ορθή σημασιολογική διασύνδεση των δεδομένων σύμφωνα με το πρότυπο AAS.

### 5.2.5 Σενάριο 5: Σημασιολογικά ερωτήματα (SPARQL Query Editor)

Το έκτο σενάριο αναδεικνύει τις προηγμένες δυνατότητες αναζήτησης της πλατφόρμας μέσω του ενσωματωμένου επεξεργαστή SPARQL. Εξετάζεται η περίπτωση ενός χρήστη με ρόλο "User", ο οποίος επιθυμεί να εκτελέσει εξειδικευμένα ερωτήματα στα δεδομένα των προϊόντων του, αξιοποιώντας τη σημασιολογική τους δομή.

#### Προϋποθέσεις:

- Ο χρήστης έχει συνδεθεί επιτυχώς στην πλατφόρμα με ρόλο "User".
- Υπάρχει τουλάχιστον ένα καταχωρημένο DPP που ανήκει στον χρήστη (π.χ. με Product ID did:john:proto:x-v1).

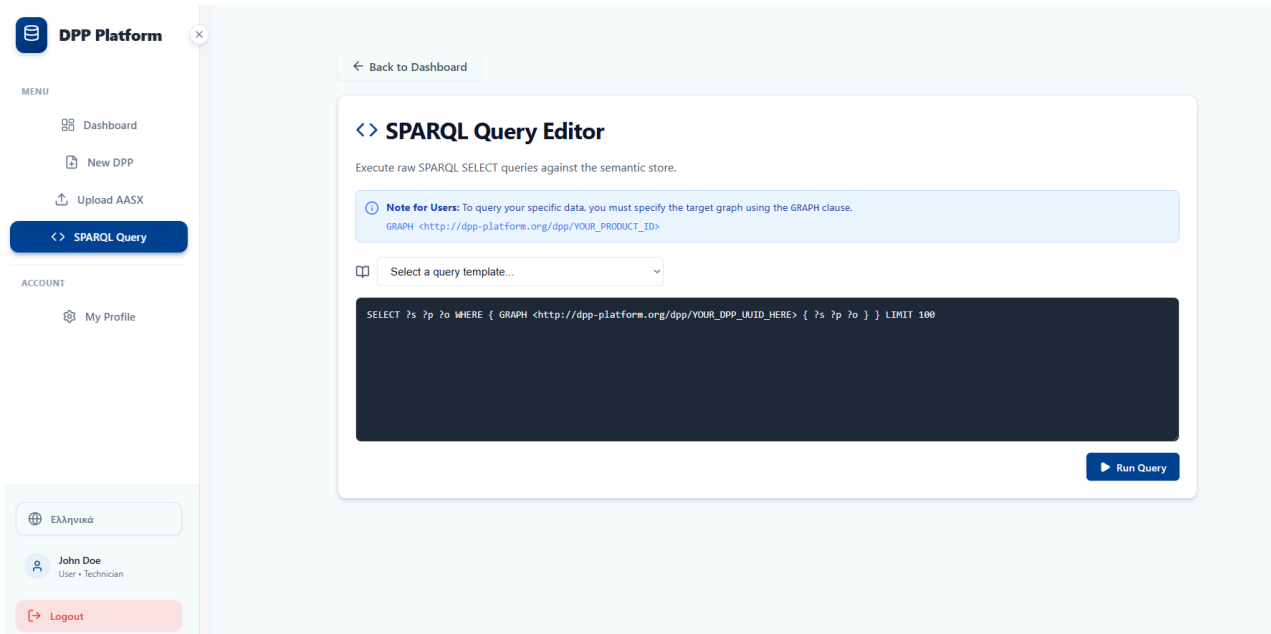
#### Ροή Ενεργειών:

1. Από το μενού πλοήγησης, ο χρήστης επιλέγει την ενότητα "SPARQL Query" (Εικόνα 56). Το σύστημα εμφανίζει τον επεξεργαστή ερωτημάτων (Εικόνα 57).

The screenshot displays the DPP Platform Dashboard. On the left, a sidebar menu includes options like 'Dashboard', 'New DPP', 'Upload AASX', and 'SPARQL Query' (highlighted with a red box and an arrow). The main area shows a 'Dashboard' with three summary cards: 'Total DPPs: 18', 'Published: 7', and 'Drafts: 11'. Below these is a search bar with 'Simple' and 'Advanced' tabs. The bottom section is a table of DPPs with columns for ID, Title, Product ID, Status, and Actions. The table lists several products, including 'Smart Digital Dosing Pump', 'Recycled Ocean Plastic Jacket', 'EcoSmart Phone Pro 5', 'EV Battery Module 75kWh', 'High-Efficiency Servo Motor 1FK7', 'Prototype X - NextGen Quantum Battery', and 'Apple iPad Pro M4'. The 'Prototype X' entry is highlighted in blue.

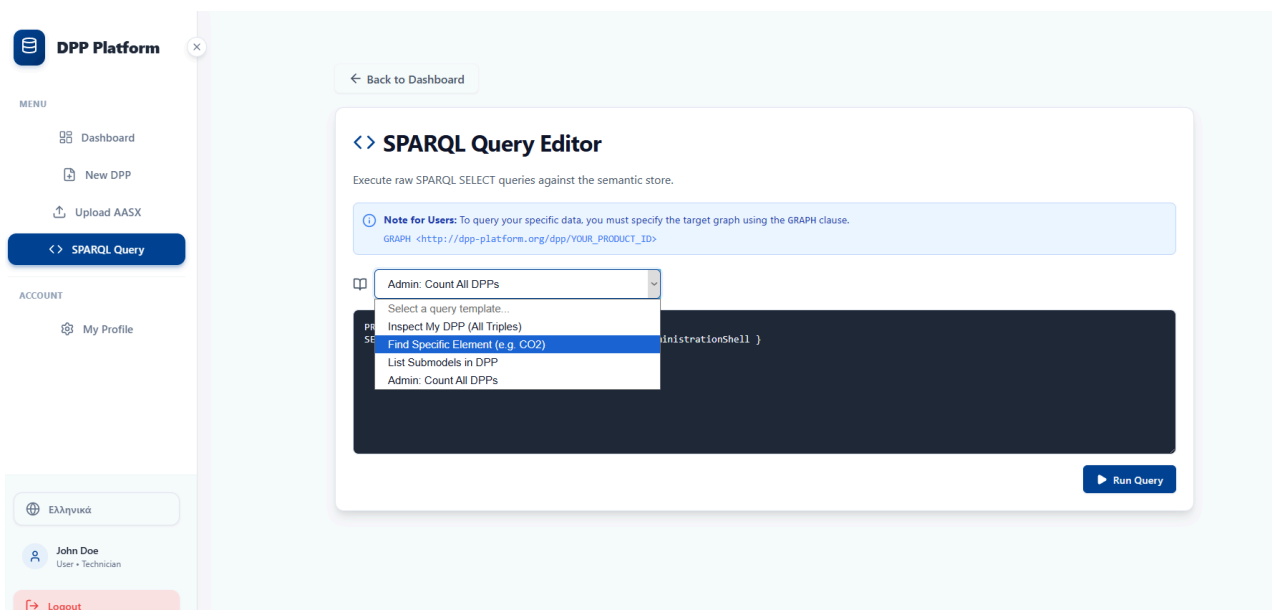
| ID   | TITLE                                 | PRODUCT ID                      | STATUS    | ACTIONS                     |
|------|---------------------------------------|---------------------------------|-----------|-----------------------------|
| #160 | Smart Digital Dosing Pump             | did:pump:smart-dose-ddc         | Published | [Icon] [JSON] [PDF]         |
| #159 | Recycled Ocean Plastic Jacket         | did:textile:jacket:ocean-blue-1 | Published | [Icon] [JSON] [PDF]         |
| #158 | EcoSmart Phone Pro 5                  | did:tech:phone:pro5-128gb       | Published | [Icon] [JSON] [PDF]         |
| #157 | EV Battery Module 75kWh               | did:gba:battery:75kwh-nmc-2024  | Published | [Icon] [JSON] [PDF]         |
| #156 | High-Efficiency Servo Motor 1FK7      | did:siemens:motor:1FK7063-2AF71 | Published | [Icon] [JSON] [PDF]         |
| #155 | Prototype X - NextGen Quantum Battery | did:john:proto:x-v1             | Published | [Icon] [JSON] [PDF] [Red X] |
| #151 | Apple iPad Pro M4                     | APL-IPD-M4-512                  | Published | [Icon] [JSON] [PDF]         |

**Εικόνα 56: Προβολή του κεντρικού πίνακα ελέγχου (Dashboard) του χρήστη και διαδικασία επιλογής της ενότητας "SPARQL Query" από το πλευρικό μενού πλοήγησης.**



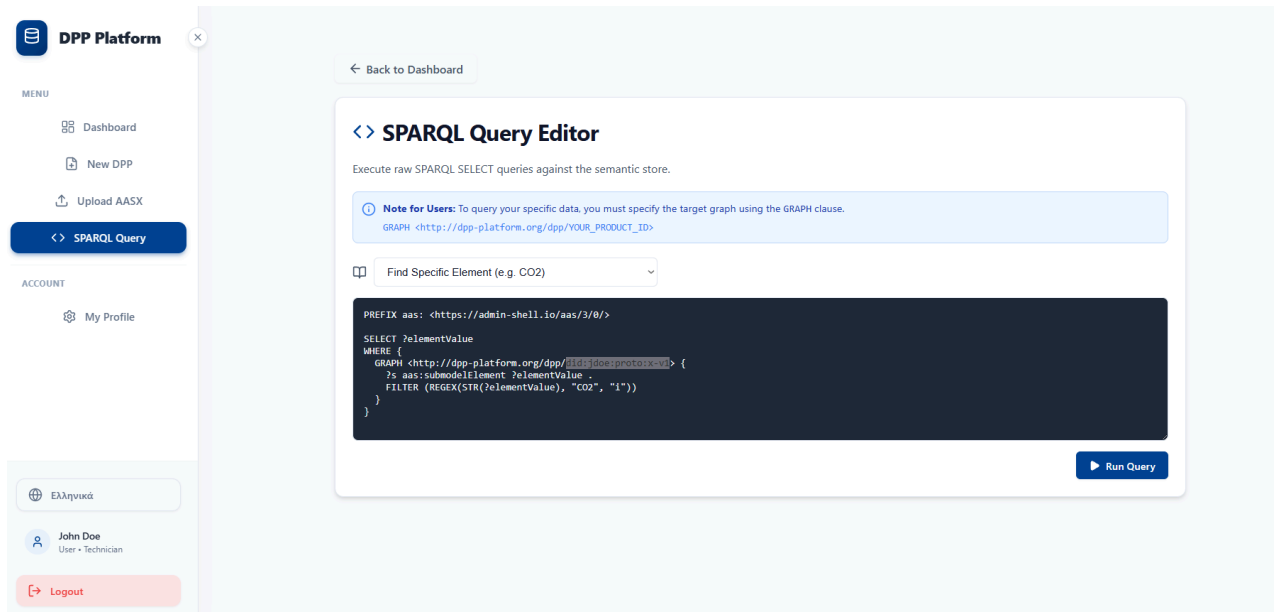
**Εικόνα 57: Ο επεξεργαστής ερωτημάτων SPARQL. Διακρίνεται το πλαίσιο οδηγιών για τη χρήση του GRAPH clause.**

2. Ο χρήστης επιθυμεί να εντοπίσει συγκεκριμένα περιβαλλοντικά δεδομένα (π.χ. εκπομπές CO<sub>2</sub>) μέσα στο πολύπλοκο δέντρο των υπο-μοντέλων του προϊόντος του. Αντί να γράφει το ερώτημα από την αρχή, επιλέγει από τη λίστα προτύπων (Templates) το "Find Specific Element (e.g. CO<sub>2</sub>)" (Εικόνα 58).



**Εικόνα 58: Χρήση προκαθορισμένων προτύπων (templates) στον SPARQL Query Editor για την ταχεία ανάκτηση συγκεκριμένων πληροφοριών, όπως περιβαλλοντικά δεδομένα, χωρίς την ανάγκη χειροκίνητης σύνταξης κώδικα.**

3. Το σύστημα συμπληρώνει αυτόματα τον κώδικα SPARQL. Ο χρήστης αντικαθιστά το placeholder YOUR\_DPP\_UUID\_HERE με το αναγνωριστικό του προϊόντος του (π.χ. did:jdope:proto:x-v1), όπως φαίνεται στην Εικόνα 59.



**Εικόνα 59: Σύνταξη ερωτήματος SPARQL με χρήση προτύπου. Ο χρήστης ορίζει ρητά το επιθυμητό γράφημα για λόγους ασφαλείας.**

4. Ο χρήστης πατάει το κουμπί "Run Query". Το σύστημα εκτελεί το ερώτημα στο σημασιολογικό αποθετήριο (Fuseki) και επιστρέφει τα αποτελέσματα σε μορφή πίνακα (Εικόνα 60).

**<> SPARQL Query Editor**

Execute raw SPARQL SELECT queries against the semantic store.

**Note for Users:** To query your specific data, you must specify the target graph using the GRAPH clause.  
GRAPH <http://dpp-platform.org/dpp/YOUR\_PRODUCT\_ID>

Find Specific Element (e.g. CO2)

```

PREFIX aas: <https://admin-shell.io/aas/3/0/>

SELECT ?elementValue
WHERE {
  GRAPH http://dpp-platform.org/dpp/did:jdoe:proto:x-v1 {
    ?s aas:submodelElement ?elementValue .
    FILTER (REGEX(STR(?elementValue), "CO2", "i"))
  }
}

```

**Run Query**

**Results**

| ELEMENTVALUE                      |
|-----------------------------------|
| ProductStage_A1_A3: 20 kgCO2e/kWh |
| CO2_Total: 25 kgCO2e/kWh          |
| Transport_A4: 1 kgCO2e/kWh        |

Total count: 3

**Εικόνα 60:** Αποτελέσματα του σημασιολογικού ερωτήματος. Εμφανίζεται και ο πίνακας των περιβαλλοντικών δεικτών (CO2).

**Αποτέλεσμα:** Ο χρήστης κατάφερε να ανακτήσει στοχευμένη πληροφορία μέσα από τη σημασιολογική δομή του DPP, παρακάμπτοντας την πολυπλοκότητα της βάσης δεδομένων και επιβεβαιώνοντας τη διαλειτουργικότητα των δεδομένων RDF.

### 5.2.6 Σενάριο 6: Διαχείριση Χρηστών και Δικαιωμάτων (Ρόλος Admin)

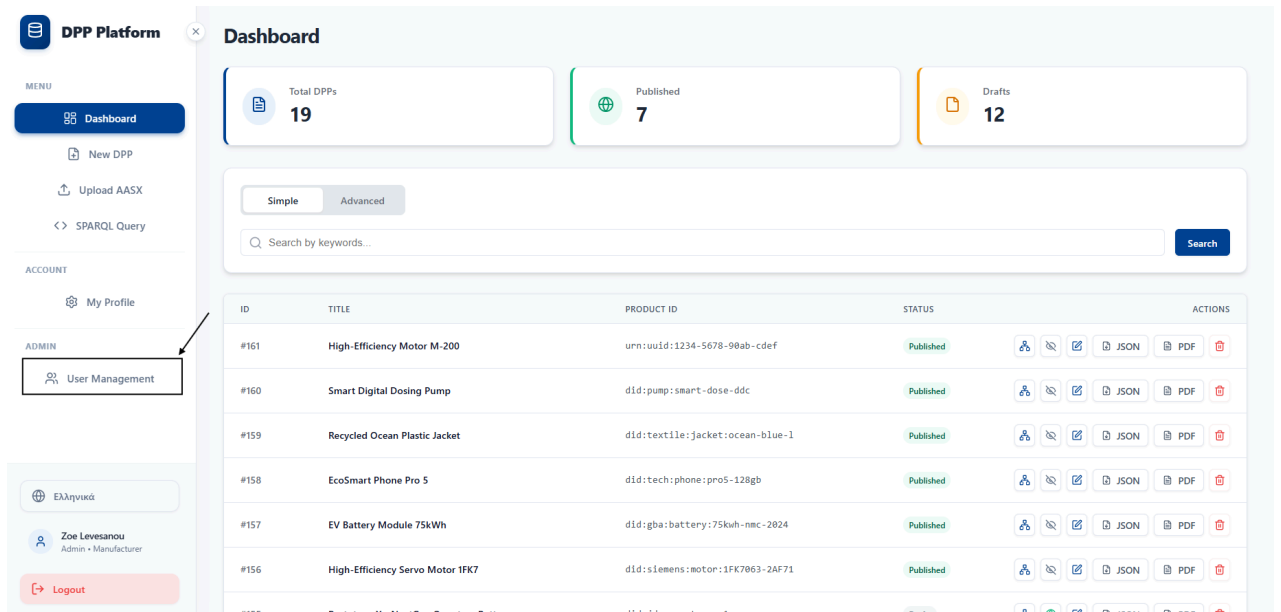
Το πέμπτο σενάριο εστιάζει στις διοικητικές λειτουργίες της πλατφόρμας, οι οποίες είναι προσβάσιμες αποκλειστικά από χρήστες με ρόλο "Admin". Επιδεικνύεται η δυνατότητα εποπτείας των εγγεγραμμένων χρηστών και η διαχείριση της κατάστασής τους (ενεργοποίηση/απενεργοποίηση).

#### Προϋποθέσεις:

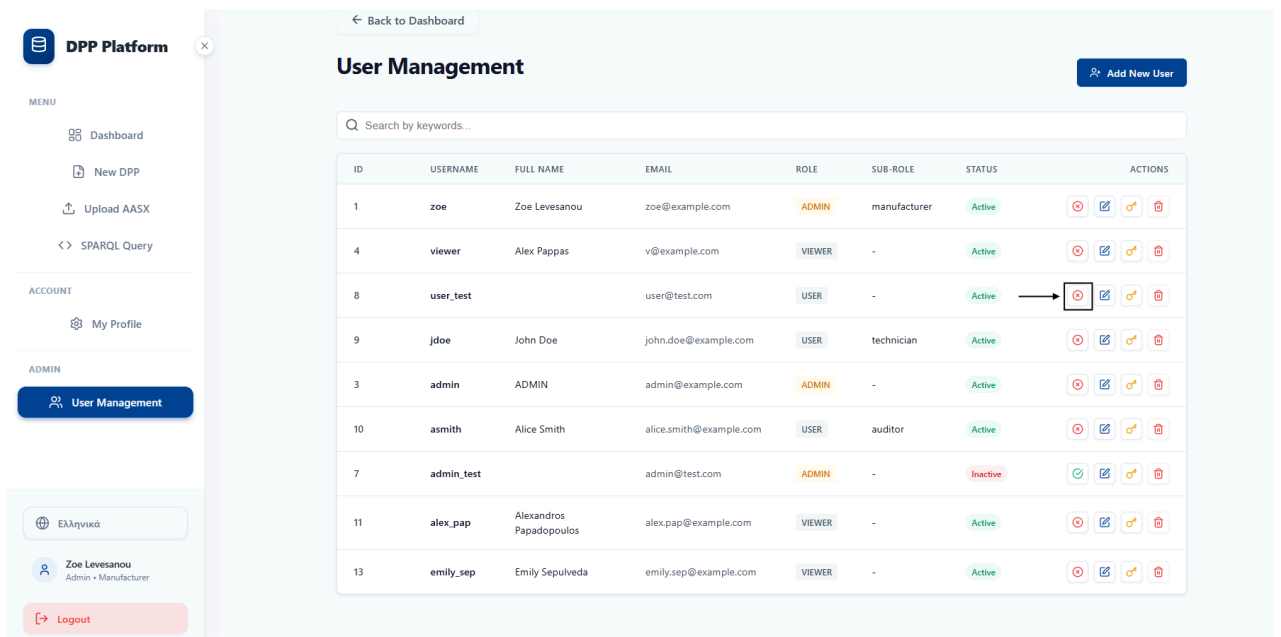
- Ο χρήστης έχει συνδεθεί επιτυχώς στην πλατφόρμα με ρόλο "Admin".
- Υπάρχουν εγγεγραμμένοι χρήστες στο σύστημα (π.χ. αυτοί που δημιουργήθηκαν στα προηγούμενα σενάρια).

#### Ροή Ενεργειών:

1. Ο διαχειριστής επιλέγει την ενότητα "User Management" από το μενού πλοήγησης (Εικόνα 61). Το σύστημα εμφανίζει έναν πίνακα με όλους τους εγγεγραμμένους χρήστες, τα στοιχεία τους (Username, Email, Role, Sub-role) και την τρέχουσα κατάστασή τους (Status) (Εικόνα 62).



Εικόνα 61: Προβολή του κεντρικού πίνακα ελέγχου (Dashboard) του διαχειριστή και διαδικασία επιλογής της ενότητας "User Management" από το πλευρικό μενού πλοήγησης.



Εικόνα 62: Η οθόνη διαχείρισης χρηστών, προσβάσιμη μόνο από διαχειριστές. Εμφανίζονται οι ρόλοι και η κατάσταση (Active/Inactive) κάθε χρήστη.

- Ο διαχειριστής εντοπίζει έναν χρήστη του οποίου την πρόσβαση επιθυμεί να ανακαλέσει (π.χ. λόγω λήξης σύμβασης ή παραβίασης όρων). Πατώντας το κουμπί "Deactivate" (εικονίδιο X) (δείτε Εικόνα 62), το σύστημα ενημερώνει άμεσα τη βάση δεδομένων και η κατάσταση του χρήστη αλλάζει σε "Inactive" (με κόκκινη ένδειξη), απαγορεύοντας άμεσα την είσοδό του στο σύστημα (Εικόνα 63).

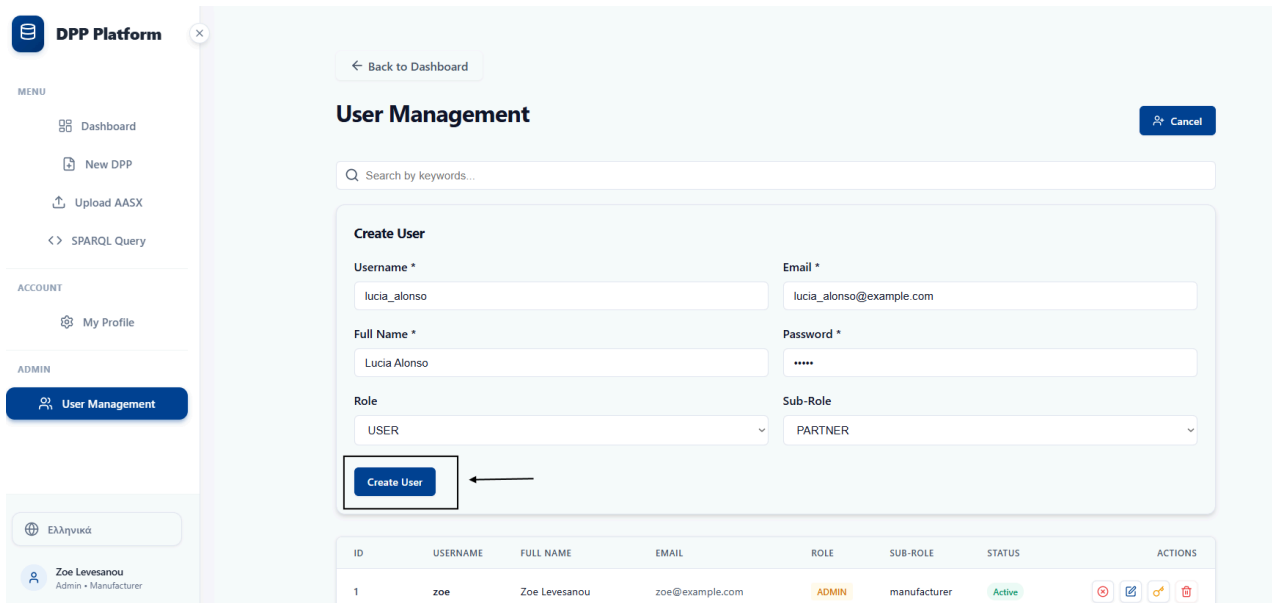
3. Λειτουργικότητες (Actions): Για κάθε χρήστη, ο διαχειριστής έχει στη διάθεσή του μια σειρά από ενέργειες μέσω των εικονιδίων στη στήλη "Actions":
  - Προβολή/Επεξεργασία: Δυνατότητα ελέγχου και τροποποίησης των στοιχείων του προφίλ.
  - Επαναφορά Κωδικού: Άμεση έκδοση νέων διαπιστευτηρίων για λόγους ασφαλείας.
  - Απενεργοποίηση (Deactivate): Πατώντας το εικονίδιο διακοπής (X), η κατάσταση του χρήστη αλλάζει σε "Inactive" (δείτε προηγούμενο βήμα (2)).
  - Διαγραφή (Delete): Οριστική αφαίρεση του χρήστη από τη βάση δεδομένων.
4. Προσθήκη Χρήστη: Μέσω του κουμπιού "Add New User" (Δείτε Εικόνα 63, πάνω δεξιά), ο διαχειριστής μπορεί να προβεί σε χειροκίνητη εγγραφή νέου μέλους, ορίζοντας εξαρχής τα δικαιώματά του (Εικόνα 64).

**User Management**

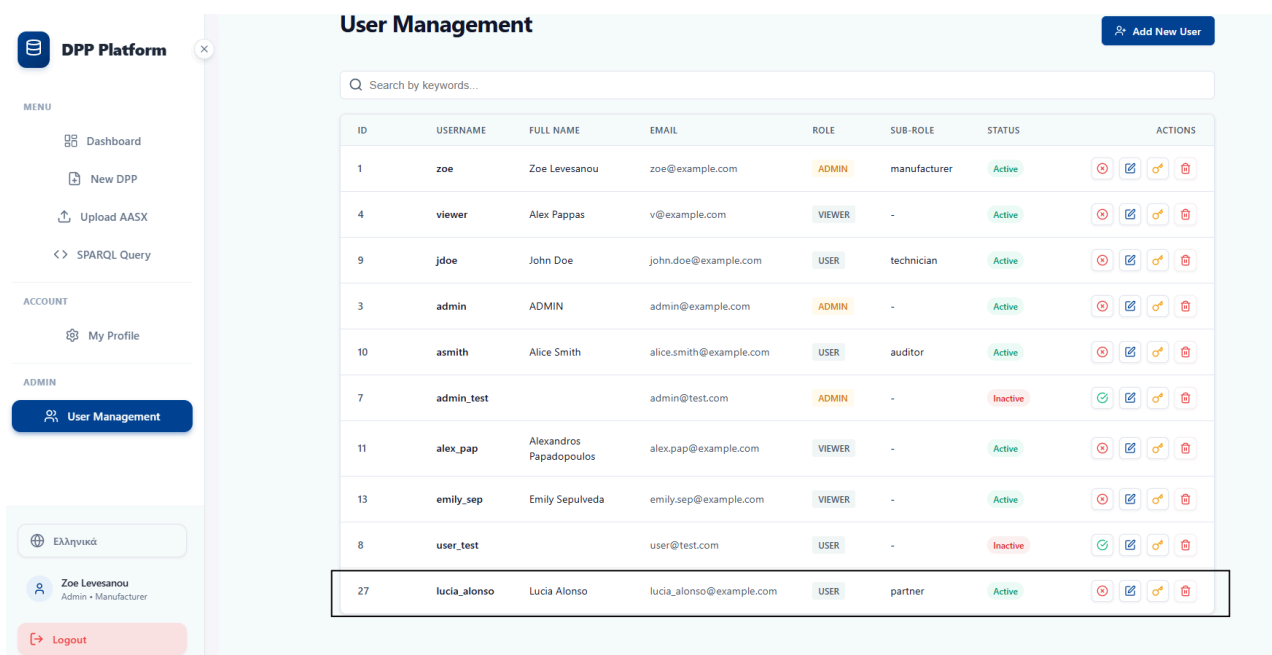
Search by keywords...

| ID | USERNAME   | FULL NAME               | EMAIL                   | ROLE   | SUB-ROLE     | STATUS   | ACTIONS |
|----|------------|-------------------------|-------------------------|--------|--------------|----------|---------|
| 1  | zoe        | Zoe Levesanou           | zoe@example.com         | ADMIN  | manufacturer | Active   | [Icons] |
| 4  | viewer     | Alex Pappas             | v@example.com           | VIEWER | -            | Active   | [Icons] |
| 9  | jdoe       | John Doe                | john.doe@example.com    | USER   | technician   | Active   | [Icons] |
| 3  | admin      | ADMIN                   | admin@example.com       | ADMIN  | -            | Active   | [Icons] |
| 10 | asmith     | Alice Smith             | alice.smith@example.com | USER   | auditor      | Active   | [Icons] |
| 7  | admin_test |                         | admin@test.com          | ADMIN  | -            | Inactive | [Icons] |
| 11 | alex_pap   | Alexandros Papadopoulos | alex.pap@example.com    | VIEWER | -            | Active   | [Icons] |
| 13 | emily_sep  | Emily Sepulveda         | emily.sep@example.com   | VIEWER | -            | Active   | [Icons] |
| 8  | user_test  |                         | user@test.com           | USER   | -            | Inactive | [Icons] |

**Εικόνα 63: Απενεργοποίηση λογαριασμού χρήστη. Η κατάσταση έχει αλλάξει σε "Inactive", απαγορεύοντας την πρόσβαση στο σύστημα.**



Εικόνα 64: Επιλογή του πλήκτρου "Create User" για την οριστικοποίηση της εγγραφής και την εισαγωγή των στοιχείων του νέου μέλους.



Εικόνα 65: Επιβεβαίωσης επιτυχούς δημιουργίας χρήστη.

**Αποτέλεσμα:** Ο διαχειριστής έχει ασκήσει επιτυχώς έλεγχο πρόσβασης, αποδεικνύοντας την αποτελεσματικότητα του μηχανισμού RBAC (Role-Based Access Control) και τη δυνατότητα άμεσης παρέμβασης για τη διασφάλιση της ασφάλειας του συστήματος.

### 5.2.7 Λοιπές Λειτουργίες και Σενάρια

Πέραν των βασικών σεναρίων που αναλύθηκαν διεξοδικά, η πλατφόρμα υποστηρίζει πλήθος συμπληρωματικών λειτουργιών που ολοκληρώνουν την εμπειρία χρήστη και την επιχειρησιακή λογική:

- *Μαζική Εισαγωγή Δεδομένων (Bulk Upload)*: Χρήστες μπορούν να μεταφορτώσουν αρχεία .aasx ή .zip, επιτρέποντας την ταυτόχρονη δημιουργία πολλαπλών DPPs και την αυτόματη εξαγωγή των μεταδεδομένων τους.
- *Σύνθετη Αναζήτηση (Advanced Search)*: Δυνατότητα φιλτραρίσματος DPPs με βάση πολλαπλά κριτήρια (π.χ. Κατασκευαστής ΚΑΙ Έτος Παραγωγής > 2023), αξιοποιώντας τις δυνατότητες του JSONB indexing.
- *Διαχείριση Προφίλ (My Profile)*: Κάθε χρήστης έχει τη δυνατότητα να ενημερώσει τα προσωπικά του στοιχεία και να αλλάξει τον κωδικό πρόσβασής του μέσω της σελίδας "My Profile".

Η συνολική λειτουργικότητα της πλατφόρμας καλύπτει τον πλήρη κύκλο ζωής των δεδομένων (CRUD), διασφαλίζοντας παράλληλα την ασφάλεια, την ακεραιότητα και τη διαλειτουργικότητα σε κάθε βήμα.

# 6

## Συμπεράσματα και μελλοντική επέκταση

### 6.1 Συμπεράσματα

Η παρούσα διπλωματική εργασία είχε ως στόχο τη μελέτη, σχεδίαση και υλοποίηση ενός συστήματος Digital Product Passport (DPP), το οποίο επιτρέπει την αποθήκευση, διαχείριση και αναζήτηση πληροφοριών σχετικών με τον κύκλο ζωής ενός προϊόντος, υποστηρίζοντας τόσο τις απαιτήσεις της κυκλικής οικονομίας όσο και τις ανάγκες της βιομηχανίας για διαφάνεια και ιχνηλασιμότητα.

Κατά τη φάση της υλοποίησης, υιοθετήθηκε μια υβριδική αρχιτεκτονική προσέγγιση αποθήκευσης, συνδυάζοντας την ισχύ μιας σχεσιακής βάσης δεδομένων (της PostgreSQL) (με υποστήριξη διαχείρισης και ημι-δομημένων δεδομένων τύπου JSONB) για δομημένα δεδομένα και την ευελιξία μιας σημασιολογικής βάσης γνώσης (του Apache Jena Fuseki) για τη σημασιολογική αναπαράσταση και ανάκτηση πληροφοριών σε μορφή RDF. Η χρήση του FastAPI για την ανάπτυξη του backend επέτρεψε τη δημιουργία ενός υψηλής απόδοσης ασύγχρονου API, ενώ η ενσωμάτωση του προτύπου JWT (JSON Web Tokens) και του μοντέλου RBAC (Role-Based Access Control) εξασφάλισε ένα στιβαρό πλαίσιο ασφαλείας και διαβαθμισμένης πρόσβασης.

Το παραχθέν σύστημα αποτελεί ένα λειτουργικό Proof of Concept (PoC), το οποίο επιτυγχάνει:

- Την αποτελεσματική διαχείριση του κύκλου ζωής των διαβατηρίων, από την εισαγωγή και την επεξεργασία έως την εξαγωγή σε διαλειτουργικούς μορφότευπους (Turtle, RDF/XML).
- Την παροχή πολλαπλών μεθόδων αναζήτησης (keyword-based, εξεζητημένη με φίλτρα και σημασιολογική μέσω SPARQL), διευκολύνοντας την ανάκτηση πληροφοριών από διαφορετικές κατηγορίες χρηστών.
- Την πλήρη ευθυγράμμιση με διεθνή βιομηχανικά πρότυπα, όπως το Asset Administration Shell (AAS), διασφαλίζοντας τη μελλοντική συμβατότητα με το οικοσύστημα του Industry 4.0.

Η ολοκλήρωση της εργασίας ανέδειξε τη σκοπιμότητα και την τεχνική εφικτότητα της υλοποίησης ενός αρθρωτού συστήματος DPP. Διαπιστώθηκε ότι η χρήση ανοιχτών προτύπων και σημασιολογικών τεχνολογιών είναι καθοριστικής σημασίας για την επίτευξη διαλειτουργικότητας μεταξύ ετερογενών συστημάτων. Η αρχιτεκτονική που αναπτύχθηκε επιτρέπει την εύκολη επεκτασιμότητα και διαλειτουργικότητα, διευκολύνοντας την απρόσκοπτη ενσωμάτωση της

πλατφόρμας με υφιστάμενα βιομηχανικά πληροφοριακά συστήματα (όπως ERP ή PLM). Η ευελιξία αυτή επιτρέπει στο σύστημα να ανταποκρίνεται αποτελεσματικά σε αυξανόμενους όγκους δεδομένων και φορτία χρηστών (scalability), καθιστώντας το ικανό να εξυπηρετήσει ένα ευρύ φάσμα οργανισμών (από μικρομεσαίες επιχειρήσεις έως μεγάλες βιομηχανικές μονάδες), διασφαλίζοντας με αυτό τον τρόπο την πλήρη συμμόρφωση με τις απαιτήσεις του κανονιστικού πλαισίου ESPR (Ecodesign for Sustainable Products Regulation).

## 6.2 Μελλοντική επέκταση

Παρά την επιτυχή επίτευξη των αρχικών στόχων, η πλατφόρμα DPP δύναται να εξελιχθεί περαιτέρω σε πολλαπλά επίπεδα, ενισχύοντας τη λειτουργικότητα, απόδοση και αξιοπιστία της. Αρχικά, μια κρίσιμη προοπτική αφορά τη **Δυναμική Ενημέρωση Δεδομένων (Live DPP)** μέσω της σύνδεσης με βιομηχανικές υποδομές IoT και middleware συστήματα, γεγονός που θα επιτρέπει την αυτόματη επικαιροποίηση περιβαλλοντικών δεικτών, όπως το πραγματικό αποτύπωμα άνθρακα κατά την παραγωγική διαδικασία, σε πραγματικό χρόνο. Σε αυτό το πλαίσιο, εξετάζεται και η δυνατότητα **Υβριδικής Εγκατάστασης (Hybrid Deployment)**<sup>33</sup>, όπου η πλατφόρμα θα μπορεί να λειτουργεί σε τοπικές υποδομές (On-Premises / Edge) εντός του εργοστασίου για άμεση διασύνδεση με συστήματα AAS (όπως το Eclipse BaSyx), συγχρονίζοντας μόνο τα απαραίτητα δεδομένα στο νέφος (cloud).

Παράλληλα με την αυτοματοποίηση των δεδομένων, κρίνεται σκόπιμη η **Ενσωμάτωση Τεχνολογίας Blockchain** με την αξιοποίηση κατακεντρωμένων καθολικών (Distributed Ledgers) για την πιστοποίηση της ακεραιότητας των πληροφοριών και τη διασφάλιση ότι οι ιστορικές καταγραφές του προϊόντος παραμένουν αμετάβλητες και επαληθεύσιμες από τρίτους φορείς.

Επιπροσθέτως, η στρατηγική ανάπτυξη ενός **Οικοσυστήματος Εφαρμογών (Application Ecosystem)** θα επιτρέψει τη δημιουργία προστιθέμενης αξίας πάνω από την πλατφόρμα. Μέσω ανοικτών APIs, εξειδικευμένες εφαρμογές τρίτων (π.χ. εργαλεία υπολογισμού LCA, πλατφόρμες διαχείρισης ανακύκλωσης) θα μπορούν να καταναλώνουν δεδομένα και να παρέχουν προηγμένες υπηρεσίες στους ιδιοκτήτες των προϊόντων, προωθώντας τη συνεργατικότητα.

Η προσπάθεια αυτή συμπληρώνεται από την σημασιολογική λειτουργικότητα που μπορεί να αναβαθμιστεί μέσω της **Ομοσπονδιακής Σημασιολογικής Αναζήτησης (Federated Semantic Search)** και της **Συλλογιστικής (Reasoning)**. Στόχος είναι η δυνατότητα εκτέλεσης σύνθετων ερωτημάτων που διασυνδέουν δεδομένα της πλατφόρμας με εξωτερικά, απομακρυσμένα αποθετήρια γνώσης (π.χ. Wikidata, άλλα DPP Systems) σε πραγματικό χρόνο, καθώς και η εφαρμογή κανόνων λογικής για την αυτόματη εξαγωγή συμπερασμάτων που δεν είναι ρητά καταγεγραμμένα στα δεδομένα.

Επιπρόσθετα, η ύπαρξη των δεδομένων σε μορφή Γράφου Γνώσης (Knowledge Graph) ανοίγει τον δρόμο για την **Αξιοποίηση Τεχνητής Νοημοσύνης και Μηχανικής Μάθησης (AI/ML)**. Μέσω αλγορίθμων που εκτελούνται απευθείας πάνω στον γράφο (Graph Neural Networks ή Knowledge Graph Embeddings), η πλατφόρμα μπορεί να εξελιχθεί από ένα σύστημα

<sup>33</sup> Υβριδική εγκατάσταση (hybrid deployment): νοείται μια κατακεντρωμένη αρχιτεκτονική όπου ορισμένα υποσυστήματα (π.χ. η διασύνδεση με τοπικά συστήματα AAS ή βιομηχανικές υποδομές) λειτουργούν σε τοπική υποδομή (On-Premises / Edge) για άμεση απόκριση, ενώ η κύρια πληροφοριακή υποδομή της εφαρμογής φιλοξενείται κεντρικά στο υπολογιστικό νέφος (Cloud).

καταγραφής σε ένα σύστημα πρόβλεψης. Συγκεκριμένα, η ΑΙ μπορεί να αναλύει τα ιστορικά δεδομένα του διαβατηρίου για την **πρόβλεψη αναγκών συντήρησης (Predictive Maintenance)** ή για την αυτόματη **βελτιστοποίηση των διαδικασιών ανακύκλωσης**, προτείνοντας την ιδανική μέθοδο ανάκτησης υλικών βάσει της χημικής σύστασης και της κατάστασης του προϊόντος.

Εν συνεχεία, η **Επέκταση Ρόλων και Διακυβέρνησης** με την προσθήκη εξειδικευμένων ρόλων για ρυθμιστικές αρχές (Regulators) και ελεγκτές (Auditors) θα ενισχύσει το πλαίσιο διαφάνειας, προσφέροντας δυνατότητες ψηφιακής υπογραφής και επίσημης επικύρωσης των δεδομένων του διαβατηρίου. Τέλος, ιδιαίτερη βαρύτητα πρέπει να δοθεί στη **Φορητότητα και Πρόσβαση στο Πεδίο** μέσω της ανάπτυξης εγγενούς (native) mobile εφαρμογής που θα επιτρέπει τη σάρωση QR codes ή NFC tags απευθείας πάνω στο φυσικό προϊόν. Η εξέλιξη αυτή θα προσφέρει άμεση πρόσβαση στις πληροφορίες του διαβατηρίου τόσο στους τελικούς καταναλωτές όσο και στους εργαζομένους σε κέντρα διαλογής, βελτιστοποιώντας τις διαδικασίες ανάκτησης και ανακύκλωσης υλικών.

Κλείνοντας, η δυναμική που προσφέρει η έννοια του Digital Product Passport, σε συνδυασμό με τις τεχνολογίες που αξιοποιήθηκαν στην παρούσα εργασία, θέτει τις βάσεις για τη δημιουργία ενός διαφανούς, βιώσιμου και ψηφιακά προηγμένου βιομηχανικού μοντέλου, ικανού να ανταποκριθεί στις προκλήσεις του μέλλοντος.

## Βιβλιογραφία

[1] Battery Pass Consortium. *Battery Passport Content Guidance – Version 1.0*. 2023.

[https://thebatterypass.eu/assets/images/content-guidance/pdf/2023\\_Battery\\_Passport\\_Content\\_Guidance.pdf](https://thebatterypass.eu/assets/images/content-guidance/pdf/2023_Battery_Passport_Content_Guidance.pdf). Accessed January 12, 2026.

[2] Battery Pass Consortium. *BatteryPassDataModel*. GitHub. n.d.

<https://github.com/batterypass/BatteryPassDataModel>. Accessed January 12, 2026.

[3] Catena-X. *DPP Catena-X – Final Report*. 2025.

[https://catena-x.net/wp-content/uploads/2025/05/DPP\\_Catena-X\\_FINAL.pdf](https://catena-x.net/wp-content/uploads/2025/05/DPP_Catena-X_FINAL.pdf). Accessed January 12, 2026.

[4] Circuits Project. "Sustainable products initiative: Building a framework for circular design." *Circuits Project*. December 9, 2025.

<https://circuitsproject.eu/2025/12/09/sustainable-products-initiative-building-a-framework-for-circular-design/>. Accessed January 12, 2026.

[5] CIRPASS Project. *Digital Product Passport Blueprint – Deliverable D3.1: DPP Use Cases, Data Requirements and Architecture*. 2023.

[https://cirpassproject.eu/wp-content/uploads/2023/03/CIRPASS\\_Benchmark-of-existing-DPP-oriented-reference-architectures.pdf](https://cirpassproject.eu/wp-content/uploads/2023/03/CIRPASS_Benchmark-of-existing-DPP-oriented-reference-architectures.pdf). Accessed January 12, 2026.

[6] CIRPASS Project. "Digital Product Passport Blueprint." *CIRPASS Project*. n.d.

<https://cirpassproject.eu>. Accessed January 12, 2026.

[7] ECHA (European Chemicals Agency). "REACH Regulation – Understanding REACH."

*European Chemicals Agency*. n.d.

<https://www.echa.europa.eu/regulations/reach/understanding-reach>. Accessed January 12, 2026.

[8] Ellen MacArthur Foundation. *Circular Economy – Completing the Picture*. 2019.

<https://www.ellenmacarthurfoundation.org/completing-the-picture>. Accessed January 12, 2026.

[9] European Commission. *A new Circular Economy Action Plan For a cleaner and more*

*competitive Europe*. COM(2020) 98 final. 2020.

[https://eur-lex.europa.eu/resource.html?uri=cellar:9903b325-6388-11ea-b735-01aa75ed71a1.0017\\_02/DOC\\_1&format=PDF](https://eur-lex.europa.eu/resource.html?uri=cellar:9903b325-6388-11ea-b735-01aa75ed71a1.0017_02/DOC_1&format=PDF). Accessed January 12, 2026.

[10] European Commission. "Sustainable products initiative." *European Commission*. June 9,

2021.

[https://ec.europa.eu/info/law/better-regulation/have-your-say/initiatives/12567-Sustainable-products-initiative\\_en](https://ec.europa.eu/info/law/better-regulation/have-your-say/initiatives/12567-Sustainable-products-initiative_en). Accessed January 12, 2026.

[11] European Commission. *Proposal for a Regulation on Ecodesign for Sustainable Products*

(ESPR). 2022. <https://data.consilium.europa.eu/doc/document/ST-7854-2022-INIT/en/pdf>.

Accessed January 12, 2026.

[12] European Commission. *Regulation (EU) 2023/1542 on Batteries and Waste Batteries*.

*Official Journal of the European Union*. 2023. <https://eur-lex.europa.eu/eli/reg/2023/1542/oj/eng>.

Accessed January 12, 2026.

[13] European Commission. *Regulation (EU) 2024/1781 establishing a framework for the*

*setting of ecodesign requirements for sustainable products*. *Official Journal of the European*

*Union*. 2024. <https://eur-lex.europa.eu/legal-content/EL/TXT/?uri=CELEX:32024R1781>.

Accessed January 12, 2026.

[14] European Commission. *Commission Implementing Decision C(2024) 5423 final on a standardisation request regarding the digital product passport*. 2024.

[https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=PI\\_COM:C\(2024\)4006](https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=PI_COM:C(2024)4006). Accessed January 12, 2026.

[15] European Commission. "Circular Economy Strategy." *European Commission – Environment*. n.d. [https://environment.ec.europa.eu/strategy/circular-economy\\_en](https://environment.ec.europa.eu/strategy/circular-economy_en). Accessed January 12, 2026.

[16] European Commission. "Ecodesign for Sustainable Products Regulation." *European Commission - Energy, Climate Change, Environment*. n.d. [https://commission.europa.eu/energy-climate-change-environment/standards-tools-and-labels/products-labelling-rules-and-requirements/ecodesign-sustainable-products-regulation\\_en](https://commission.europa.eu/energy-climate-change-environment/standards-tools-and-labels/products-labelling-rules-and-requirements/ecodesign-sustainable-products-regulation_en). Accessed January 12, 2026.

[17] European Commission. "WEEE (Waste Electrical and Electronic Equipment)." *European Commission – Environment*. n.d. [https://environment.ec.europa.eu/topics/waste-and-recycling/waste-electrical-and-electronic-equipment-weee\\_en](https://environment.ec.europa.eu/topics/waste-and-recycling/waste-electrical-and-electronic-equipment-weee_en). Accessed January 12, 2026.

[18] European Green Deal. "Strategic Priorities." *European Commission*. n.d. [https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal\\_en](https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en). Accessed January 12, 2026.

[19] European Parliament. *Briefing: Digital Product Passport*. European Parliamentary Research Service, Study No. 757808. 2024. [https://www.europarl.europa.eu/RegData/etudes/STUD/2024/757808/EPRS\\_STU\(2024\)757808\\_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/STUD/2024/757808/EPRS_STU(2024)757808_EN.pdf). Accessed January 12, 2026.

- [20] European Union / Data.europa.eu. "EU's Digital Product Passport: Advancing transparency and sustainability." *Data.europa.eu*. 2025.  
<https://data.europa.eu/en/news-events/news/eus-digital-product-passport-advancing-transparency-and-sustainability>. Accessed January 12, 2026.
- [21] FEAD. "REACH Revision – Position Paper." *FEAD*. 2023.  
<https://fead.be/position/reach-revision>. Accessed January 12, 2026.
- [22] Finboot. "From linear to circular economy: the role of Digital Product Passports in sustainable production and consumption." *Finboot Blog*. 2022.  
<https://www.finboot.com/post/from-linear-to-circular-economy-the-role-of-digital-product-passports-in-sustainable-production-and-consumption>. Accessed January 12, 2026.
- [23] GS1. "Digital Link Standard Overview." *GS1*. n.d.  
<https://www.gs1.org/standards/gs1-digital-link>. Accessed January 12, 2026.
- [24] IDSA (International Data Spaces Association). *Bridging the Gap: Between IDS and Industry 4.0*. 2024.  
[https://internationaldataspaces.org/wp-content/uploads/dlm\\_uploads/IDSA-Position-Paper-Bridging-the-Gap-Between-IDS-and-Industry-4.0.pdf](https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Position-Paper-Bridging-the-Gap-Between-IDS-and-Industry-4.0.pdf). Accessed January 12, 2026.
- [25] IEC (International Electrotechnical Commission). *Smart manufacturing – Reference architecture model Industry 4.0 (RAMI 4.0)*. IEC PAS 63088:2017. 2017.  
[https://www.digitale-technologien.de/DT/Redaktion/DE/Downloads/Publikation/PAiCE\\_Leitfaden\\_Reference\\_Architecture.pdf?\\_\\_blob=publicationFile&v=1](https://www.digitale-technologien.de/DT/Redaktion/DE/Downloads/Publikation/PAiCE_Leitfaden_Reference_Architecture.pdf?__blob=publicationFile&v=1). Accessed January 12, 2026.
- [26] International Organization for Standardization. *Information technology — Automatic identification and data capture techniques — Unique identifiers — Part 1: Individual packages*.

ISO/IEC 15459-1:2014. 2015. <https://www.iso.org/standard/54600.html>. Accessed January 12, 2026.

[27] International Organization for Standardization. *Information technology — Automatic identification and data capture techniques — EPC Information Services (EPCIS) Standard*.

ISO/IEC 19987:2021. 2021. <https://www.iso.org/standard/79717.html>. Accessed January 12, 2026.

[28] iPoint-systems. *Digital Product Passport Whitepaper: Turning Circular Economy into Reality*. 2022. <https://www.ipoint-systems.com>. Accessed January 12, 2026.

[29] OPC Foundation. "Industry 4.0 Asset Administration Shell – General Information / AAS Metamodel." *OPC Foundation Reference*. n.d.

<https://reference.opcfoundation.org/I4AAS/v100/docs/4.1>. Accessed January 12, 2026.

[30] Plattform Industrie 4.0. "Details of the Asset Administration Shell." *Plattform Industrie 4.0*. n.d. <https://www.plattform-i40.de>. Accessed January 12, 2026.

[31] Plociennik, C., M. Pourjafarian, S. Saleh, T. Hagedorn, A. d. Carmo Precci Lopes, M. Vogelgesang, J. Baehr, B. Kellerer, M. Jansen, H. Berg, M. Ruskowski, L. Schebek, and A. Ciroth. "Requirements for a Digital Product Passport to Boost the Circular Economy." In *INFORMATIK 2022*, Gesellschaft für Informatik, Bonn (2022): 1485–1494.

[https://doi.org/10.18420/inf2022\\_127](https://doi.org/10.18420/inf2022_127). Accessed January 12, 2026.

[32] Right to Repair Campaign. "What's My Right to Repair." *Repair.eu*. n.d.

<https://repair.eu/whats-my-right-to-repair>. Accessed January 12, 2026.

[33] Royce, W. "Managing the Development of Large Software Systems." *Proceedings of IEEE WESCON* (1970): 1–9. <https://ieeexplore.ieee.org/document/9357688>. Accessed January 12, 2026.

- [34] TechTarget. "Waterfall Model – Definition and Overview." *SearchSoftwareQuality*. n.d. <https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model>. Accessed January 12, 2026.
- [35] W3C. "RDF and OWL Standards." *W3C — Semantic Web Standards*. n.d. <https://www.w3.org/standards/semanticweb/>. Accessed January 12, 2026.
- [36] European Union. "Regulation (EU) 2023/1542 concerning batteries and waste batteries." Official Journal of the European Union. 2023. <https://eur-lex.europa.eu/eli/reg/2023/1542/oj/eng>. Accessed January 23, 2026.
- [37] ISO. "ISO 8000-1:2022 Data quality." *International Organization for Standardization*. 2022.
- [38] ISO/IEC. "ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection." 2022.
- [39] ISO/IEC. "ISO/IEC 20248:2022 Information technology — Automatic identification and data capture techniques." 2022.
- [40] W3C. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML." *W3C Member Submission*. 2004. <https://www.w3.org/Submission/SWRL/>. Accessed January 23, 2026.
- [41] W3C. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML." *W3C Member Submission*. 2004. <https://www.w3.org/Submission/SWRL/>. Accessed January 23, 2026.
- [42] ECLASS e.V. "ECLASS Standard for Classification and Product Description." ECLASS. n.d. <https://www.eclass.eu/en/standard.html>. Accessed January 23, 2026.
- [43] IEC. "IEC CDD (Common Data Dictionary) – IEC 61360-4 DB." International Electrotechnical Commission. n.d. <https://cdd.iec.ch/>. Accessed January 23, 2026.
- [44] TextileGenesis. "Digital Traceability Platform for the Apparel and Home Textile Industry." TextileGenesis. 2024. <https://textilegenesis.com>. Accessed January 23, 2026.

[45] Kovacic, I., et al. "Digital Ecosystem to enable Circular Buildings – The Circular Twin Framework Proposal." *Journal of Sustainable Development of Energy Water and Environment Systems*. 2024. <https://doi.org/10.13044/j.sdewes.d12.0493>. Accessed January 23, 2026.

[46] Product Ontology. "The Open Product Data project." [productontology.org](http://www.productontology.org/). n.d. <http://www.productontology.org/>. Accessed January 23, 2026.

## Παράρτημα I: Τεκμηρίωση REST API

Το παρόν παράρτημα παρέχει αναλυτική περιγραφή των διαθέσιμων endpoints του συστήματος, όπως αυτά τεκμηριώνονται αυτόματα μέσω του προτύπου OpenAPI (Swagger UI).

### I.1 Authentication (Αυθεντικοποίηση)

Η ενότητα αυτή περιλαμβάνει τις λειτουργίες για την πιστοποίηση χρηστών και τη διαχείριση συνεδριών.

- **POST /auth/login:** Ελέγχει τα διαπιστευτήρια του χρήστη (username, password) και επιστρέφει ένα JWT Access Token.
  - Είσοδος: username, password.
  - Έξοδος: access\_token, token\_type (Bearer).
  - Σφάλματα: 401 Unauthorized (Λάθος κωδικός).
- **GET /auth/me:** Επιστρέφει τα στοιχεία του τρέχοντος συνδεδεμένου χρήστη βάσει του token.
  - Είσοδος: -
  - Έξοδος: JSON αντικείμενο με τα στοιχεία του χρήστη (π.χ. username, email, role, is\_active).
  - Απαιτεί: Bearer Token.

**DPP Management Platform** 0.1.0 OAS 3.1  
/openapi.json  
Digital Product Passport Management Backend

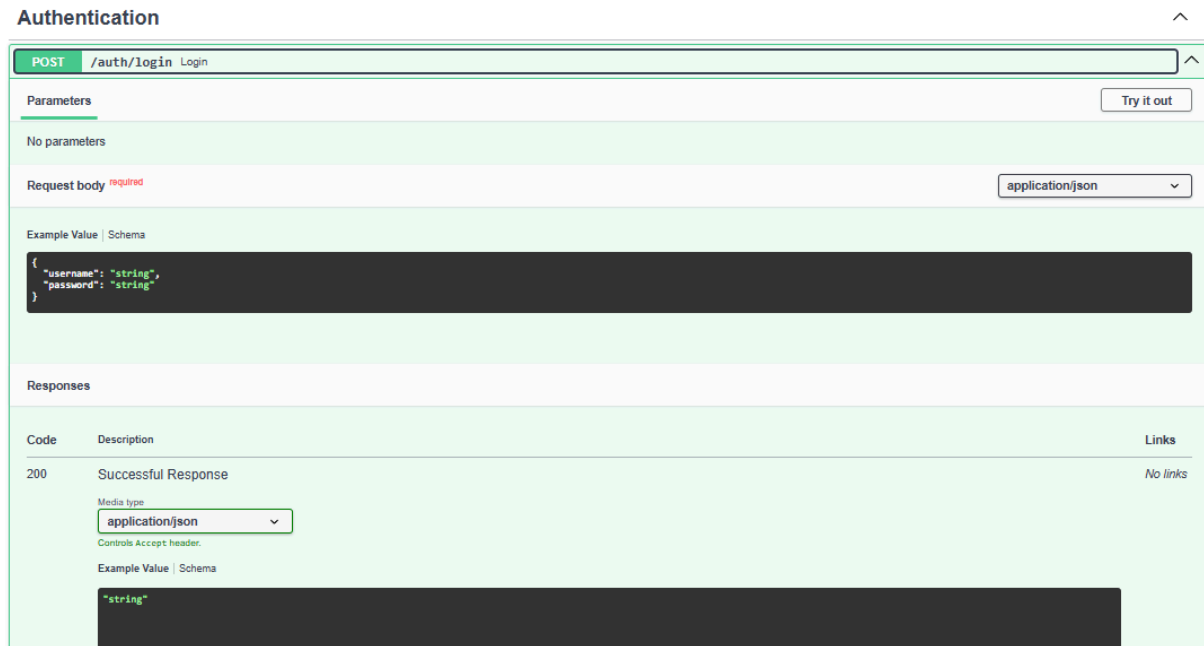
Authorize

#### Authentication

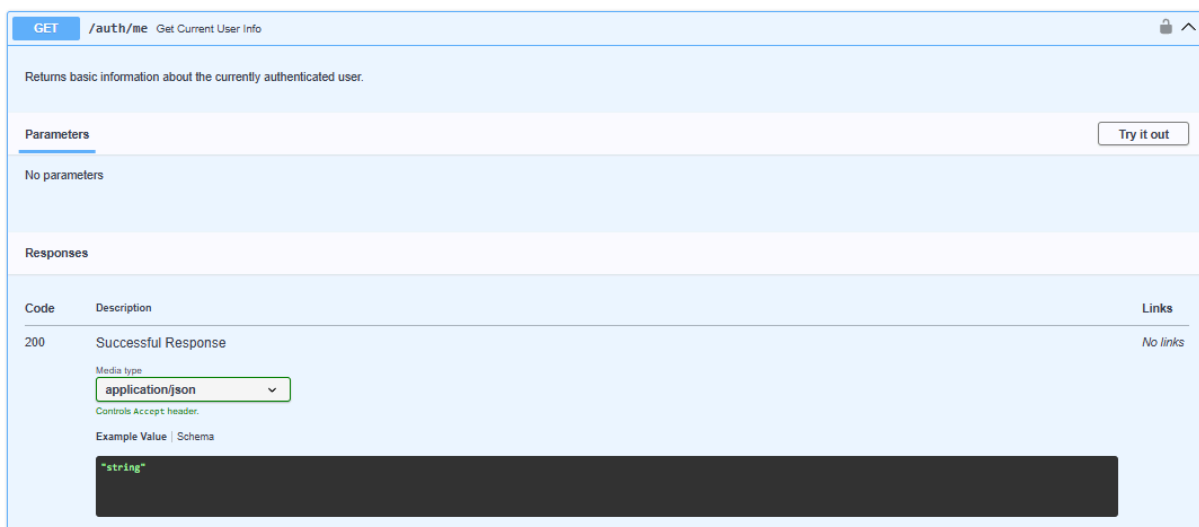
POST /auth/login Login

GET /auth/me Get Current User Info

Εικόνα 66: Τα endpoints της ενότητας Authentication στο Swagger UI.



Εικόνα 67: Τεκμηρίωση της μεθόδου `POST /auth/login` στο Swagger UI για τη διαδικασία αυθεντικοποίησης και εισόδου του χρήστη στην πλατφόρμα.



Εικόνα 68: Τεκμηρίωση της μεθόδου `GET /auth/me` στο Swagger UI για την ανάκτηση των βασικών πληροφοριών του τρέχοντος συνδεδεμένου χρήστη.

## I.2 Users (Διαχείριση χρηστών)

Παρέχει λειτουργίες για την πλήρη διαχείριση (CRUD) των λογαριασμών χρηστών.

- **GET /users/all:** Επιστρέφει λίστα με όλους τους εγγεγραμμένους χρήστες.
  - Είσοδος: -

- Έξοδος: Λίστα από JSON αντικείμενα (Array of JSON), όπου το καθένα περιέχει τα στοιχεία ενός χρήστη (id, username, email, role, κλπ.).
- Πρόσβαση: Admin.
- **GET /users/paginated:** Επιστρέφει χρήστες με σελιδοποίηση (pagination).
  - Είσοδος: Παράμετροι Query (Query Parameters): skip (ακέραιος, προεπιλογή 0 - παράλειψη εγγραφών), limit (ακέραιος, προεπιλογή 10 - μέγιστο όριο εγγραφών προς επιστροφή).
  - Έξοδος: JSON αντικείμενο που περιέχει δύο κλειδιά: total (ο συνολικός αριθμός χρηστών) και items (λίστα JSON με τα στοιχεία των χρηστών για τη συγκεκριμένη σελίδα).
  - Πρόσβαση: Admin
- **GET /users/{user\_id}:** Ανάκτηση στοιχείων συγκεκριμένου χρήστη.
  - Είσοδος: user\_id (Path Parameter στο URL).
  - Έξοδος: JSON αντικείμενο με τα πλήρη στοιχεία του ζητούμενου χρήστη (id, username, email, role, subrole, is\_active)
  - Πρόσβαση: Ο ίδιος ο χρήστης ή Admin.
- **PUT /users/{user\_id}:** Ενημέρωση στοιχείων χρήστη (email, full name, role).
  - Είσοδος: user\_id (Path Parameter στο URL). Στο Request Body (JSON) μπορούν να αποσταλούν τα προς ενημέρωση πεδία (π.χ. email, full\_name, role, subrole).
  - Έξοδος: JSON αντικείμενο με τα ενημερωμένα πλέον στοιχεία του χρήστη.
  - Πρόσβαση: Ο ίδιος ο χρήστης (για βασικά στοιχεία) ή Admin.
- **DELETE /users/{user\_id}:** Διαγραφή χρήστη.
  - Είσοδος: user\_id (Path Parameter στο URL).
  - Έξοδος: Καμία επιστροφή δεδομένων (Status Code 204 No Content) επί επιτυχούς διαγραφής.
  - Πρόσβαση: Ο ίδιος ο χρήστης ή Admin.
- **GET /users/filter:** Αναζήτηση χρηστών με κριτήρια.
  - Είσοδος: Παράμετροι Query (Query Parameters) στο URL. Διαθέσιμα φίλτρα: role, subrole, email, username (μπορούν να συνδυαστούν).
  - Έξοδος: Λίστα από JSON αντικείμενα (Array of JSON) που περιέχουν τους χρήστες που ικανοποιούν τα δοθέντα κριτήρια.
  - Παράμετροι: role, subrole, email, username.
  - Πρόσβαση: Admin.
- **POST /users/create:** Δημόσια εγγραφή νέου χρήστη (Sign Up).
  - Είσοδος: username, email, password, role (User/Viewer), subrole (π.χ. Consumer).
  - Έξοδος: Τα στοιχεία του νέου χρήστη (χωρίς τον κωδικό του).
  - Περιορισμοί: Επιτρέπει τη δημιουργία μόνο απλών χρηστών (User) ή επισκεπτών (Viewer). Απαγορεύεται η δημιουργία διαχειριστών (Admin).
  - Πρόσβαση: Δημόσιο (Public).
- **POST /users/admin/create:** Δημιουργία χρηστών με ειδικά δικαιώματα (Admin, User κλπ.).
  - Είσοδος: JSON αντικείμενο στο Request Body που περιέχει τα στοιχεία του νέου χρήστη (username, email, password, role, προαιρετικά subrole και full\_name).
  - Έξοδος: JSON αντικείμενο με τα στοιχεία του νεοδημιουργηθέντος χρήστη (χωρίς τον κωδικό πρόσβασης), με HTTP Status Code 201 Created.
  - Πρόσβαση: Admin.
- **PUT /users/{user\_id}/password:** Αλλαγή κωδικού πρόσβασης.
  - Είσοδος: user\_id (Path Parameter στο URL). Το Request Body (JSON) περιέχει το new\_password.

- Έξοδος: JSON αντικείμενο με μήνυμα επιτυχίας (π.χ. {"msg": "Password updated successfully"}).
- Πρόσβαση: Ο ίδιος ο χρήστης ή Admin.
- **PATCH /users/{user\_id}/status:** Ενεργοποίηση/Απενεργοποίηση λογαριασμού (is\_active).
  - Είσοδος: user\_id (Path Parameter στο URL). Το Request Body (JSON) περιέχει τη νέα κατάσταση, π.χ. {"is\_active": false}.
  - Έξοδος: JSON αντικείμενο με τα ενημερωμένα στοιχεία του χρήστη, επιβεβαιώνοντας την αλλαγή του is\_active.
  - Πρόσβαση: Admin.

| Users  |                           |                      | ^   |
|--------|---------------------------|----------------------|-----|
| GET    | /users/all                | Get Users            | 🔒 ▼ |
| GET    | /users/paginated          | Get Paginated Users  | 🔒 ▼ |
| GET    | /users/{user_id}          | Get User By Id       | 🔒 ▼ |
| PUT    | /users/{user_id}          | Update User          | 🔒 ▼ |
| DELETE | /users/{user_id}          | Delete User          | 🔒 ▼ |
| GET    | /users/filter             | Filter Users         | 🔒 ▼ |
| POST   | /users/create             | Create User          | ▼   |
| POST   | /users/admin/create       | Create User By Admin | 🔒 ▼ |
| PUT    | /users/{user_id}/password | Update Password      | 🔒 ▼ |
| PATCH  | /users/{user_id}/status   | Update User Status   | 🔒 ▼ |

*Εικόνα 69: Λίστα endpoints για τη διαχείριση χρηστών (Users Controller).*

Εικόνα 70: Τεκμηρίωση της μεθόδου **POST /users/create** στο Swagger UI για τη δημιουργία νέου χρήστη και τον ορισμό των αντίστοιχων ρόλων και υπο-ρόλων (roles & sub-roles).

### 1.3 DPP JSON (Διαχείριση διαβατηρίων)

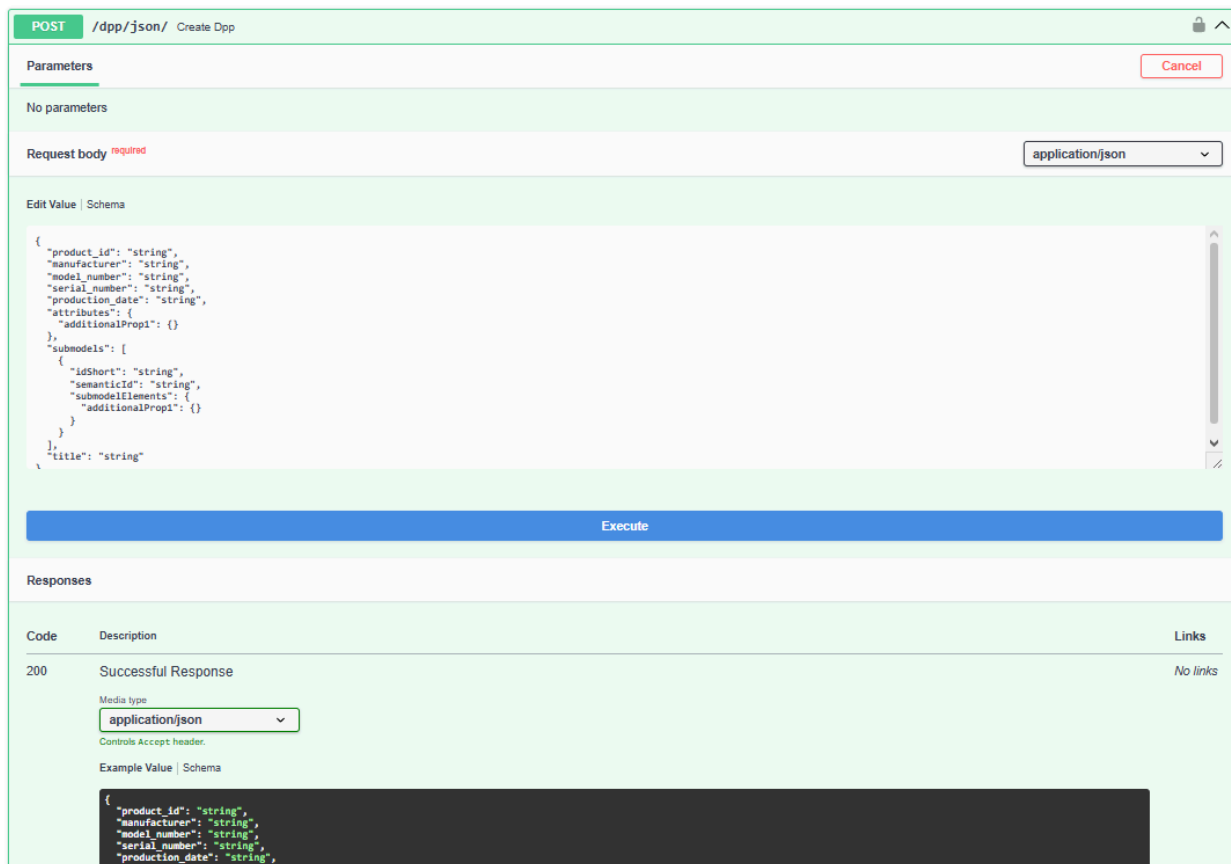
Η βασική ενότητα για τη διαχείριση των Ψηφιακών Διαβατηρίων Προϊόντων.

- **GET /dpp/json/stats:** Επιστρέφει στατιστικά στοιχεία (Σύνολο DPPs, Published, Drafts, My DPPs).
  - Είσοδος: -
  - Έξοδος: JSON αντικείμενο με τα αθροίσματα (π.χ. {"total\_dpps": 100, "published": 80, "drafts": 20, "my\_dpps": 5}).
  - Πρόσβαση: Όλοι οι αυθεντικοποιημένοι χρήστες.
- **POST /dpp/json/:** Δημιουργία νέου DPP.
  - Είσοδος: JSON αντικείμενο στο Request Body που περιλαμβάνει title, product\_id, και το βασικό φορτίο δεδομένων στο πεδίο dpp\_data (το οποίο περιέχει και τα submodels).
  - Έξοδος: JSON αντικείμενο με τα στοιχεία του νέου DPP (περιλαμβάνει το νέο dpp\_uuid, id, ημερομηνίες δημιουργίας), με HTTP Status Code 201 Created.
  - Πρόσβαση: Όλοι οι αυθεντικοποιημένοι χρήστες (εκτός Viewer).
- **GET /dpp/json/{dpp\_id}:** Ανάκτηση πλήρων στοιχείων ενός DPP.
  - Είσοδος: dpp\_id (Path Parameter).
  - Έξοδος: JSON αντικείμενο με όλα τα δεδομένα και μεταδεδομένα του DPP.

- Πρόσβαση:
  - Published DPP: Όλοι οι χρήστες.
  - Draft DPP: Μόνο ο Ιδιοκτήτης (Owner) και οι Admins.
- **PUT /dpp/json/{dpp\_id}**: Ενημέρωση στοιχείων DPP (τίτλος, δεδομένα).
  - Είσοδος: dpp\_id (Path Parameter). Στο Request Body (JSON) αποστέλλονται τα νέα δεδομένα (μπορεί να είναι μερική ή πλήρης ανανέωση των πεδίων title, product\_id, dpp\_data).
  - Έξοδος: JSON αντικείμενο με την ενημερωμένη μορφή του DPP.
  - Πρόσβαση: Μόνο ο Ιδιοκτήτης και οι Admins.
- **DELETE /dpp/json/{dpp\_id}**: Οριστική διαγραφή DPP.
  - Είσοδος: dpp\_id (Path Parameter).
  - Έξοδος: Καμία επιστροφή δεδομένων (Status Code 204 No Content) επί επιτυχούς διαγραφής.
  - Πρόσβαση: Μόνο ο Ιδιοκτήτης και οι Admins.
- **PUT /dpp/json/{dpp\_id}/publish**: Δημοσίευση DPP (γίνεται ορατό σε όλους).
  - Είσοδος: dpp\_id (Path Parameter).
  - Έξοδος: JSON αντικείμενο με το ενημερωμένο DPP, όπου το πεδίο is\_published είναι πλέον true (ή false στο unpublish).
  - Πρόσβαση: Μόνο ο Ιδιοκτήτης και οι Admins.
- **PUT /dpp/json/{dpp\_id}/unpublish**: Απόσυρση DPP (γίνεται Draft).
  - Είσοδος: dpp\_id (Path Parameter στο URL).
  - Έξοδος: JSON αντικείμενο με τα ενημερωμένα στοιχεία του DPP, όπου το is\_published είναι false.
  - Πρόσβαση: Μόνο ο Ιδιοκτήτης και οι Admins.
- **POST /dpp/json/search**: Αναζήτηση DPPs (Simple & Advanced).
  - Είσοδος: JSON αντικείμενο στο Request Body που περιλαμβάνει: keywords (λίστα λέξεων για αναζήτηση κειμένου), filters (λεξικό με ακριβή κριτήρια, π.χ. {"product\_id": "123"}), και προαιρετικά weights (για σταθμισμένη αναζήτηση).
  - Έξοδος: Λίστα από JSON αντικείμενα (Array of JSON) που αναπαριστούν τα DPPs που ικανοποιούν τα κριτήρια, ταξινομημένα (αν έχουν εφαρμοστεί βάρη).
  - Πρόσβαση: Όλοι οι αυθεντικοποιημένοι χρήστες (βλέπουν μόνο Published και τα δικά τους Drafts).

| DPP JSON |                              |                    | ^   |
|----------|------------------------------|--------------------|-----|
| GET      | /dpp/json/stats              | Get Dpp Stats      | 🔒 ▼ |
| POST     | /dpp/json/                   | Create Dpp         | 🔒 ▼ |
| GET      | /dpp/json/{dpp_id}           | Get Dpp            | 🔒 ▼ |
| PUT      | /dpp/json/{dpp_id}           | Update Dpp         | 🔒 ▼ |
| DELETE   | /dpp/json/{dpp_id}           | Delete Dpp         | 🔒 ▼ |
| PUT      | /dpp/json/{dpp_id}/publish   | Publish Dpp        | 🔒 ▼ |
| PUT      | /dpp/json/{dpp_id}/unpublish | Unpublish Dpp      | 🔒 ▼ |
| POST     | /dpp/json/search             | Search Dpp Entries | 🔒 ▼ |

Εικόνα 71: Endpoints για CRUD λειτουργίες και αναζήτηση DPPs.



Εικόνα 72: Τεκμηρίωση της μεθόδου `POST /dpp/json/` για τη χειροκίνητη καταχώρηση και δημιουργία ενός νέου DPP μέσω της υποβολής δομημένων δεδομένων σε μορφή JSON.

**PUT** /dpp/json/{dpp\_id} Update Dpp

**Parameters**

| Name                                   | Description |
|----------------------------------------|-------------|
| dpp_id * required<br>integer<br>(path) | dpp_id      |

**Request body** required application/json

Edit Value | Schema

```
{
  "product_id": "string",
  "manufacturer": "string",
  "model_number": "string",
  "serial_number": "string",
  "production_date": "string",
  "attributes": {
    "additionalProp1": {}
  },
  "submodels": [
    {
      "idShort": "string",
      "semanticId": "string",
      "submodelElements": {
        "additionalProp1": {}
      }
    }
  ],
  "title": "string"
}
```

**Execute**

**Responses**

| Code | Description         | Links     |
|------|---------------------|-----------|
| 200  | Successful Response | API links |

Εικόνα 73: Τεκμηρίωση της μεθόδου **PUT /dpp/json/{dpp\_id}**. Εμφανίζεται η διαδικασία ενημέρωσης (Update) των στοιχείων ενός ήδη καταχωρημένου DPP, χρησιμοποιώντας το μοναδικό αναγνωριστικό (**dpp\_id**) ως παράμετρο διαδρομής.

## 1.4 DPP SPARQL (Σημασιολογικά ερωτήματα)

Ενότητα για την αλληλεπίδραση με τον Γράφο Γνώσης (Fuseki).

- **POST /dpp/sparql/query:** Εκτέλεση raw SPARQL SELECT ερωτημάτων.
  - Είσοδος: JSON αντικείμενο στο Request Body που περιλαμβάνει το query (το ίδιο το SPARQL string σε μορφή κειμένου).
  - Έξοδος: JSON αντικείμενο στη standard μορφή απάντησης SPARQL Results (π.χ. {"head": {...}, "results": {"bindings": [...]}}).
  - Πρόσβαση:
    - Admin: Πλήρης πρόσβαση σε όλα τα γραφήματα (Graphs).
    - User/Viewer: Περιορισμένη πρόσβαση μόνο σε Published γραφήματα ή στα δικά τους (Drafts). Απαιτείται ρητή αναφορά του Graph URI στο query.
  - Ασφάλεια: Επιτρέπει μόνο SELECT queries. Οι απλοί χρήστες περιορίζονται σε συγκεκριμένους γράφους.
- **GET /dpp/sparql/graph/{dpp\_id}:** Επιστρέφει τη δομή του γράφου (Nodes/Edges) για οπτικοποίηση στο Frontend.
  - Είσοδος: dpp\_id (Path Parameter στο URL).
  - Έξοδος: JSON αντικείμενο που περιέχει δύο λίστες: nodes (οι κόμβοι του γράφου, π.χ. οντότητες, ιδιότητες) και edges (οι ακμές/σχέσεις μεταξύ τους).
  - Πρόσβαση:
    - Published DPP: Όλοι οι αυθεντικοποιημένοι χρήστες.

- Draft DPP: Μόνο ο Ιδιοκτήτης και οι Admins.

| DPP SPARQL |                            |                             | ^ |
|------------|----------------------------|-----------------------------|---|
| POST       | /dpp/sparql/query          | Sparql Query Execution      | 🔒 |
| GET        | /dpp/sparql/graph/{dpp_id} | Get Dpp Graph Visualization | 🔒 |

Εικόνα 74: Endpoints εκτέλεσης SPARQL ερωτημάτων.

**DPP SPARQL** ^

**POST** /dpp/sparql/query Sparql Query Execution 🔒

Executes a raw SPARQL SELECT query against the semantic store (Fuseki). Only SELECT queries are permitted.

**SECURITY:**

- Admins can query everything.
- Regular users can ONLY query graphs that are PUBLISHED.
- Queries MUST specify target graphs explicitly (via URIs in the query).

**Parameters** Try it out

No parameters

**Request body** required application/json

**Example Value** | **Schema**

```
{  "query": "string",  "limit": 100,  "offset": 0}
```

**Responses**

| Code | Description         | Links    |
|------|---------------------|----------|
| 200  | Successful Response | No links |

Media type: application/json

Controls Accept header.

**Example Value** | **Schema**

```
{  "total_count": 0,  "limit": 0,  "offset": 0,  "results": [    {      "additionalProp1": {}    }  ]}
```

Εικόνα 75: Τεκμηρίωση της μεθόδου **POST /dpp/sparql/query** για την εκτέλεση σημασιολογικών ερωτημάτων **SELECT** στη βάση *Fuseki*, με ενσωματωμένους περιορισμούς ασφαλείας βάσει του ρόλου του χρήστη.

## 1.5 DPP export & files

Λειτουργίες εισαγωγής και εξαγωγής δεδομένων.

- **POST /dpp/files/upload:** Μεταφόρτωση αρχείων .aasx ή .json για μαζική δημιουργία DPPs.

- Είσοδος: Αρχείο (Multipart/Form-Data).
- Έξοδος: JSON αντικείμενο με τα στοιχεία του νεοδημιουργηθέντος DPP (περιλαμβάνει το νέο dpp\_uuid, id, title κλπ.), με HTTP Status Code 201 Created
- Πρόσβαση: Όλοι οι αυθεντικοποιημένοι χρήστες (εκτός Viewer).
- **GET /dpp/export/{dpp\_id}**: Εξαγωγή των δεδομένων του DPP σε μορφή JSON (για χρήση από εξωτερικά συστήματα).
  - Είσοδος: dpp\_id (Path Parameter στο URL).
  - Έξοδος: Αρχείο τύπου application/json (ως downloadable file attachment).
  - Πρόσβαση:
    - Published DPP: Όλοι οι αυθεντικοποιημένοι χρήστες.
    - Draft DPP: Μόνο ο Ιδιοκτήτης και οι Admins.
- **GET /dpp/export/{dpp\_id}/pdf**: Παραγωγή και λήψη αρχείου PDF με τα στοιχεία του DPP.
  - Είσοδος: dpp\_id (Path Parameter στο URL).
  - Έξοδος: Αρχείο τύπου application/pdf.
  - Πρόσβαση:
    - Published DPP: Όλοι οι αυθεντικοποιημένοι χρήστες.
    - Draft DPP: Μόνο ο Ιδιοκτήτης και οι Admins.

#### DPP Files

|      |                   |             |   |
|------|-------------------|-------------|---|
| POST | /dpp/files/upload | Upload Aasx | ▼ |
|------|-------------------|-------------|---|

#### DPP Export

|     |                          |                |   |
|-----|--------------------------|----------------|---|
| GET | /dpp/export/{dpp_id}     | Export Dpp     | ▼ |
| GET | /dpp/export/{dpp_id}/pdf | Export Dpp Pdf | ▼ |

*Εικόνα 76: Endpoints για εξαγωγή (Export) και μεταφόρτωση (Files).*

**DPP Files**

**POST** /dpp/files/upload Upload Aasx

Upload and process a .aasx or .zip file containing DPP data. The endpoint extracts XML/JSON, validates, and stores unified data in DB.

**Parameters** Cancel Reset

No parameters

**Request body** *required* multipart/form-data

**file** *required*  
string(binary) Choose File No file chosen

**Execute**

**Responses**

| Code | Description         | Links    |
|------|---------------------|----------|
| 200  | Successful Response | No links |

Media type: application/json

Controls: Accept header.

Example Value | Schema

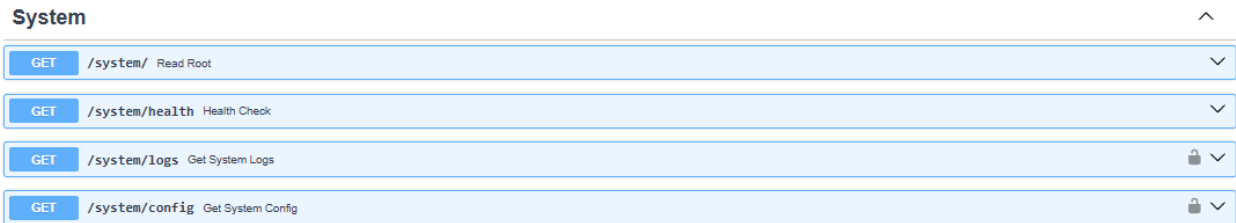
**Εικόνα 77:** Τεκμηρίωση της μεθόδου **POST /dpp/files/upload** στο **Swagger UI** για τη μεταφόρτωση και αυτόματη επεξεργασία αρχείων τύπου **.aasx** ή **.zip** που περιέχουν δεδομένα διαβατηρίου.

## 1.6 System (Σύστημα)

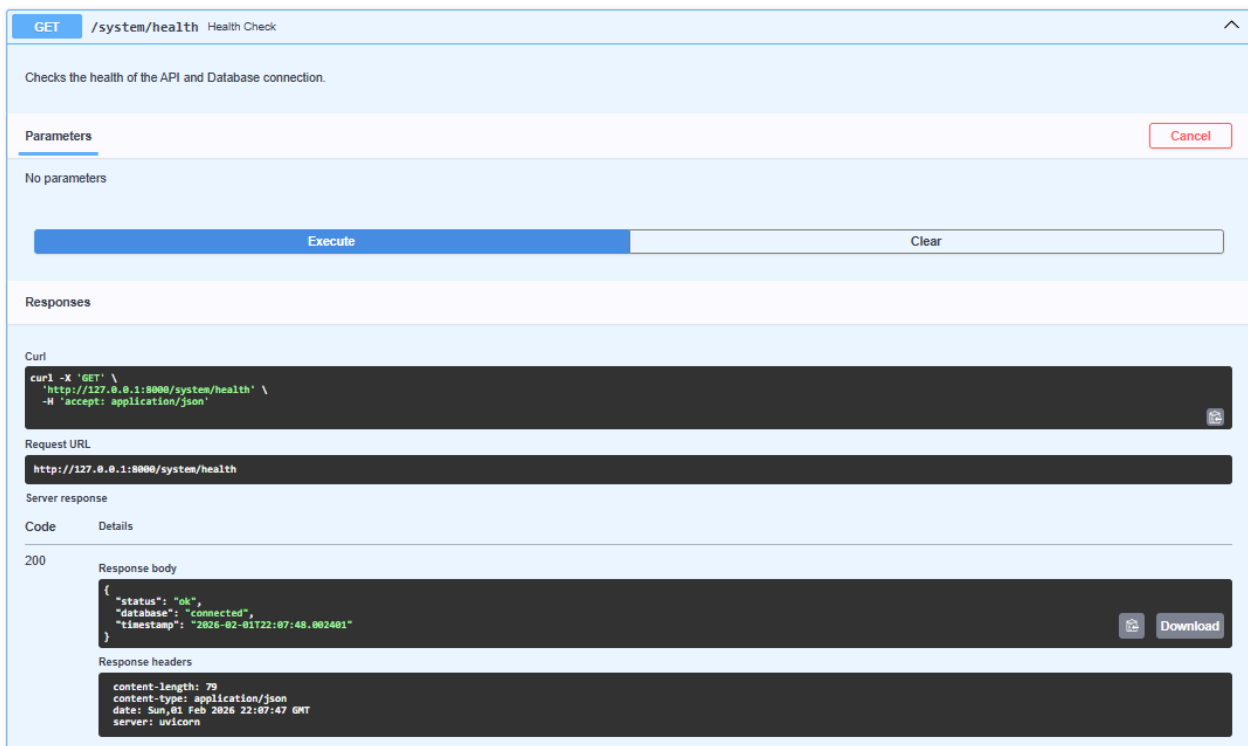
Λειτουργίες ελέγχου κατάστασης και υγείας του συστήματος.

- **GET /system/:** Απλό μήνυμα καλωσορίσματος για επιβεβαίωση ότι το API τρέχει.
  - Είσοδος: -
  - Έξοδος: JSON αντικείμενο με μήνυμα.
  - Πρόσβαση: Admins.
- **GET /system/health:** Ελέγχει τη συνδεσιμότητα με τη βάση δεδομένων εκτελώντας ένα απλό query (SELECT 1).
  - Είσοδος: -
  - Έξοδος: JSON αντικείμενο με τη συνολική κατάσταση και την επιμέρους κατάσταση κάθε υποσυστήματος (π.χ. {"status": "ok", "database": "ok", "fuseki": "ok"}). Σε περίπτωση σφάλματος επιστρέφει το είδος του σφάλματος.
  - Πρόσβαση: Admins.
- **GET /system/logs:** Επιστρέφει τις τελευταίες 50 γραμμές από το αρχείο καταγραφής (logs) του συστήματος.
  - Είσοδος: ( Προαιρετικά query parameters για πλήθος γραμμών, π.χ. ?lines=50).
  - Έξοδος: JSON αντικείμενο που περιέχει μία λίστα με τις γραμμές του αρχείου καταγραφής (π.χ. {"logs": ["ERROR:...", "INFO:..."]}).
  - Πρόσβαση: Admins.
- **GET /system/config:** Επιστρέφει τις ρυθμίσεις του συστήματος (χωρίς ευαίσθητα δεδομένα όπως κωδικούς).
  - Είσοδος: -

- Έξοδος: JSON αντικείμενο με ζεύγη κλειδιού-τιμής που αφορούν μη ευαίσθητες ρυθμίσεις (π.χ. {"ENVIRONMENT": "production", "DATABASE\_POOL\_SIZE": 10}).
- Πρόσβαση: Admins.



Εικόνα 78: Endpoint ελέγχου κατάστασης και υγείας συστήματος.

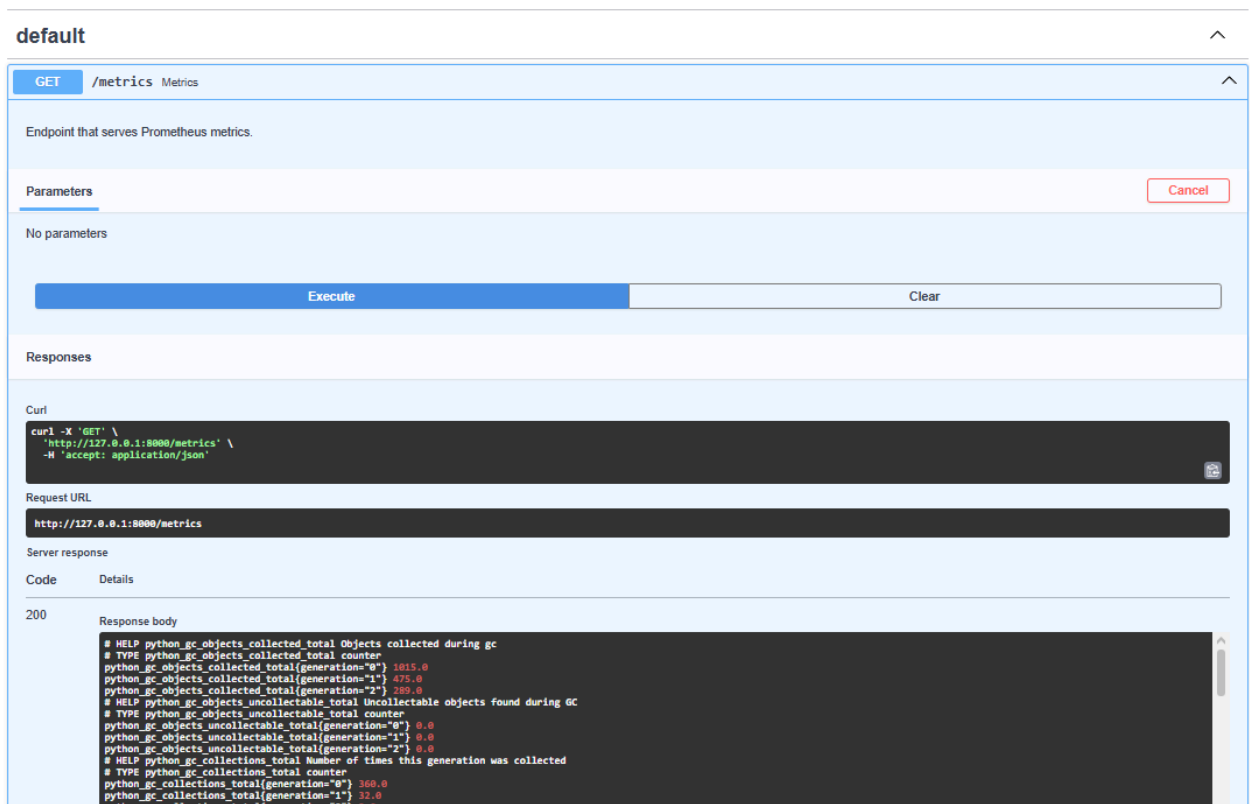


Εικόνα 79: Τεκμηρίωση της μεθόδου GET /system/health στο Swagger UI για τον έλεγχο της κατάστασης λειτουργίας (Health Check) του API και της σύνδεσης με τη βάση δεδομένων.

## I.7 Metrics & Monitoring (Παρακολούθηση)

Λειτουργίες για τη συλλογή μετρήσεων απόδοσης και την παρακολούθηση του συστήματος από τον διαχειριστή (admin).

- **GET /metrics:** Επιστρέφει μετρήσεις σε μορφή συμβατή με Prometheus (π.χ. CPU usage, Memory usage, HTTP Request rates, Latency). Καταναλώνεται αυτόματα από τον Prometheus Server για την τροφοδοσία των γραφημάτων στην Grafana.
  - Είσοδος: -
  - Έξοδος: Αρχείο απλού κειμένου (text/plain) όπου κάθε γραμμή αναπαριστά μια μέτρηση με τη μορφή `metric_name {label="value"} value` (π.χ. `http_requests_total{method="GET", status="200"} 154`).
  - Πρόσβαση: Admins.



Εικόνα 80: Δείγμα των μετρήσεων που παράγει το endpoint `/metrics`.

## ***Παράρτημα II: Παρακολούθηση Επιδόσεων και Υγείας***

### ***Συστήματος (Grafana Monitoring Dashboard)***

Στο παρόν παράρτημα παρουσιάζεται το dashboard παρακολούθησης που υλοποιήθηκε στο περιβάλλον Grafana. Το εργαλείο αυτό προσφέρει οπτικοποίηση σε πραγματικό χρόνο των κρίσιμων παραμέτρων της πλατφόρμας DPP, όπως η χρήση πόρων (CPU/RAM), ο ρυθμός αιτημάτων (RPS) και η απόκριση των endpoints.

Η διάταξη έχει σχεδιαστεί με γνώμονα την υψηλή πληροφοριακή πυκνότητα, ακολουθώντας βιομηχανικά πρότυπα (όπως το TrueNAS style), επιτρέποντας στον διαχειριστή να εντοπίζει άμεσα τυχόν σφάλματα ή καθυστερήσεις στη σημασιολογική επεξεργασία των δεδομένων. Το dashboard χωρίζεται σε δύο κύριες νοητές ενότητες: το System Health για την υποδομή και το API Performance για την ανάλυση της κίνησης.

#### ***II.1 System Health: Κατάσταση και Πόροι Συστήματος***

Στην κορυφή του dashboard τοποθετήθηκε μια γραμμή κατάστασης (Status Bar) για την άμεση εποπτεία της διαθεσιμότητας:

- **Athens Clock:** Ρολόι πραγματικού χρόνου συγχρονισμένο με την τοπική ώρα, υλοποιημένο με Stat Panel και τη συνάρτηση `time()`×1000 για την ορθή απεικόνιση της ημερομηνίας.
- **System Status:** Ψηφιακός δείκτης που επιστρέφει την τιμή **ONLINE** (πράσινο) ή **OFFLINE** (κόκκινο), συνδεδεμένος άμεσα με την κατάσταση του backend.
- **Uptime:** Stat Panel που υπολογίζει τη διάρκεια αδιάλειπτης λειτουργίας της εφαρμογής, παρέχοντας κρίσιμη πληροφορία για τη σταθερότητα του συστήματος.

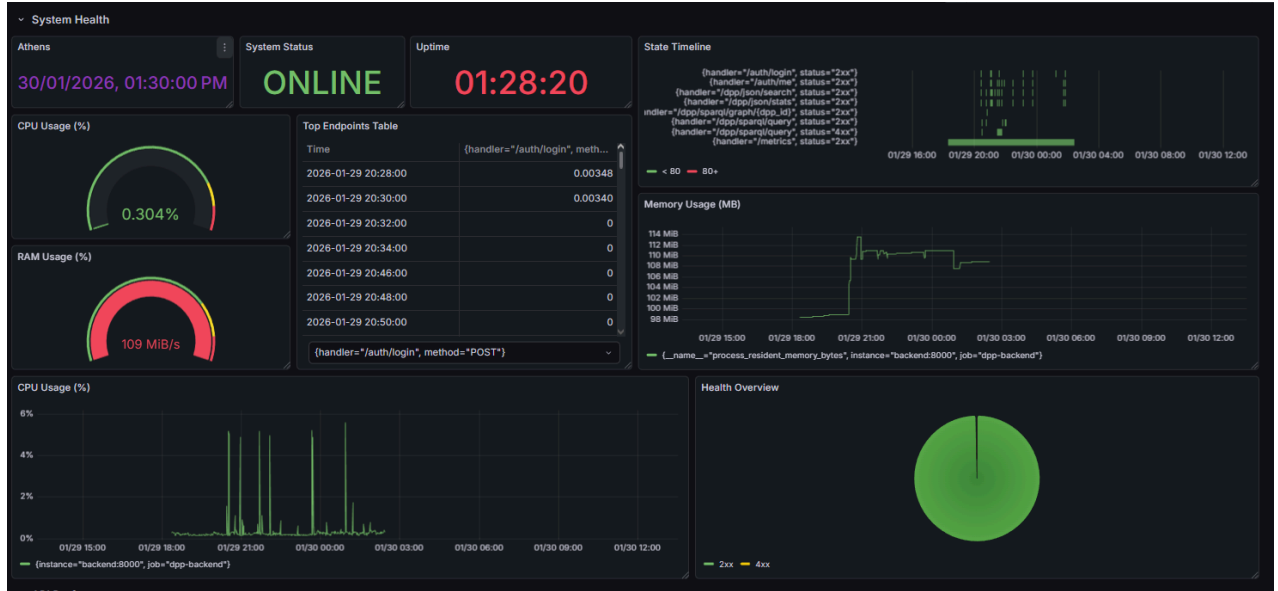
#### ***II.2 Παρακολούθηση Υλικών Πόρων (CPU & RAM)***

Η παρακολούθηση της CPU επιτρέπει τον εντοπισμό περιόδων έντονης υπολογιστικής δραστηριότητας, ειδικά κατά την εκτέλεση σύνθετων ερωτημάτων SPARQL στη βάση Fuseki. Παράλληλα, η μέτρηση της Memory Usage (RAM) διασφαλίζει ότι η εφαρμογή δεν παρουσιάζει διαρροές μνήμης (memory leaks) κατά τη μαζική επεξεργασία αρχείων .aasx.

Για την οπτικοποίηση χρησιμοποιήθηκαν **Gauges** με ενσωματωμένα κατώφλια (thresholds) βάσει της λογικής της προληπτικής διαχείρισης:

- **Thresholds RAM:** Το πρώτο επίπεδο προειδοποίησης ορίστηκε στο 75% (384MB) για τον εντοπισμό ασυνήθιστης φόρτωσης, ενώ το κρίσιμο επίπεδο στο 90% (460MB) για την αποφυγή σφαλμάτων Out-of-Memory (OOM).

- **Resource Limits:** Το ανώτατο όριο (Max) των μετρικών παρακολούθησης ορίστηκε βάσει των ορίων της εφαρμογής (512MB) και όχι της συνολικής μνήμης του εξυπηρετητή (16GB), διασφαλίζοντας ότι ακόμα και μικρές διακυμάνσεις στη χρήση των πόρων, όπως της μνήμης RAM και της CPU, είναι άμεσα ορατές στον διαχειριστή μέσω του dashboard.



Εικόνα 81: Συνολική διάταξη της ενότητας System Health στο Grafana, όπου απεικονίζονται οι δείκτες κατάστασης, τα Gauges πόρων και το ιστορικό χρήσης μνήμης.

## II.3 API Performance: Ανάλυση Κίνησης και Απόκρισης

Η δεύτερη ενότητα του dashboard επικεντρώνεται στη συμπεριφορά του REST API υπό συνθήκες φόρτου, παρέχοντας δεδομένα για την ποιότητα των παρεχόμενων υπηρεσιών (QoS).

### Ρυθμός Αιτημάτων (Requests per Second - RPS)

Αντί για την απλή καταγραφή των συνολικών αιτημάτων, χρησιμοποιήθηκε η συνάρτηση rate για τον υπολογισμό του αριθμού των κλήσεων ανά δευτερόλεπτο. Η προσέγγιση αυτή επιτρέπει τον εντοπισμό απότομων αυξήσεων στην κίνηση (traffic spikes), οι οποίες οπτικοποιούνται με διαφορετικά χρώματα ανά endpoint (π.χ. `/auth/login`, `/dpp/sparql/query`).

### Καθυστέρηση Απόκρισης (Latency Analysis)

Στο panel του Average Latency, εφαρμόστηκαν PromQL queries για τον υπολογισμό του 95th percentile (P95). Ενώ η μέση τιμή μπορεί να αποκρύπτει μεμονωμένες μεγάλες καθυστερήσεις, το P95 αποκαλύπτει το χρόνο απόκρισης για το 95% των χρηστών, διασφαλίζοντας ότι ακόμα και τα πιο «βαριά» σημασιολογικά ερωτήματα SPARQL εκτελούνται εντός αποδεκτών ορίων.

### Διαχείριση Σφαλμάτων (Error Rate)

Για την παρακολούθηση της σταθερότητας, υλοποιήθηκε ένας δυναμικός υπολογισμός του ποσοστού σφαλμάτων. Η φόρμουλα βασίζεται στον λόγο των αποτυχημένων αιτημάτων προς το σύνολο των αιτημάτων σε ένα κυλιόμενο χρονικό παράθυρο:

$$\frac{\sum \text{rate}(\text{http\_requests\_total}\{\text{status} = \sim "4..|5.." \}[1m])}{\sum \text{rate}(\text{http\_requests\_total}[1m])} \times 100$$

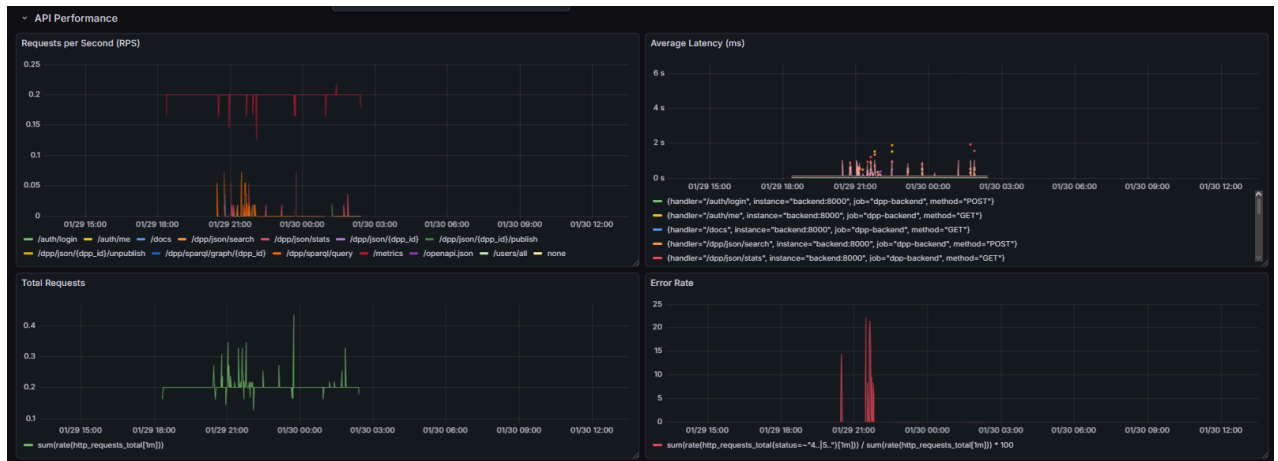
Πιο συγκεκριμένα, ο αριθμητής υπολογίζει τον ρυθμό των HTTP αιτημάτων που επέστρεψαν κωδικούς σφάλματος των κατηγοριών 4xx (client errors) και 5xx (server errors). Ο παρονομαστής περιλαμβάνει το συνολικό πλήθος όλων των αιτημάτων που απαντήθηκαν από το σύστημα, ανεξαρτήτως αποτελέσματος. Η μέτρηση πραγματοποιείται με βάση τον μέσο όρο μεταβολής ανά 1 λεπτό, επιτρέποντας την άμεση ανίχνευση απότομων αυξήσεων (spikes) στα σφάλματα.

Το αποτέλεσμα οπτικοποιείται με έντονο κόκκινο χρώμα, επιτρέποντας στον διαχειριστή να αντιληφθεί αν μια νέα έκδοση του κώδικα ή μια δυσλειτουργία της βάσης δεδομένων προκαλεί αποτυχίες στις κλήσεις HTTP.

### Σημασιολογική Εποπτεία και Ιστορικό Κατάστασης

Για την ολοκληρωμένη εικόνα της πλατφόρμας, ενσωματώθηκαν δύο εξειδικευμένα εργαλεία ανάλυσης:

- **Health Overview (Pie Chart):** Χρησιμοποιήθηκαν κανονικές εκφράσεις (wildcards όπως `2.*`, `4.*`, `5.*`) για την ομαδοποίηση των status codes. Αυτό επιτρέπει τη γρήγορη αξιολόγηση της συνολικής υγείας, όπου το πράσινο τμήμα αντιπροσωπεύει το ποσοστό των επιτυχών κλήσεων έναντι των σφαλμάτων.
- **State Timeline:** Το panel αυτό λειτουργεί ως «μαύρο κουτί» του συστήματος, καταγράφοντας την κατάσταση κάθε endpoint (handler) σε μια οριζόντια λωρίδα χρόνου. Η χρήση της ίδιας χρωματικής κωδικοποίησης διασφαλίζει τη γνωστική συνοχή, επιτρέποντας τη σύνδεση μιας καθυστέρησης στο latency με ένα συγκεκριμένο χρονικό σημείο σφάλματος σε κάποιο endpoint.

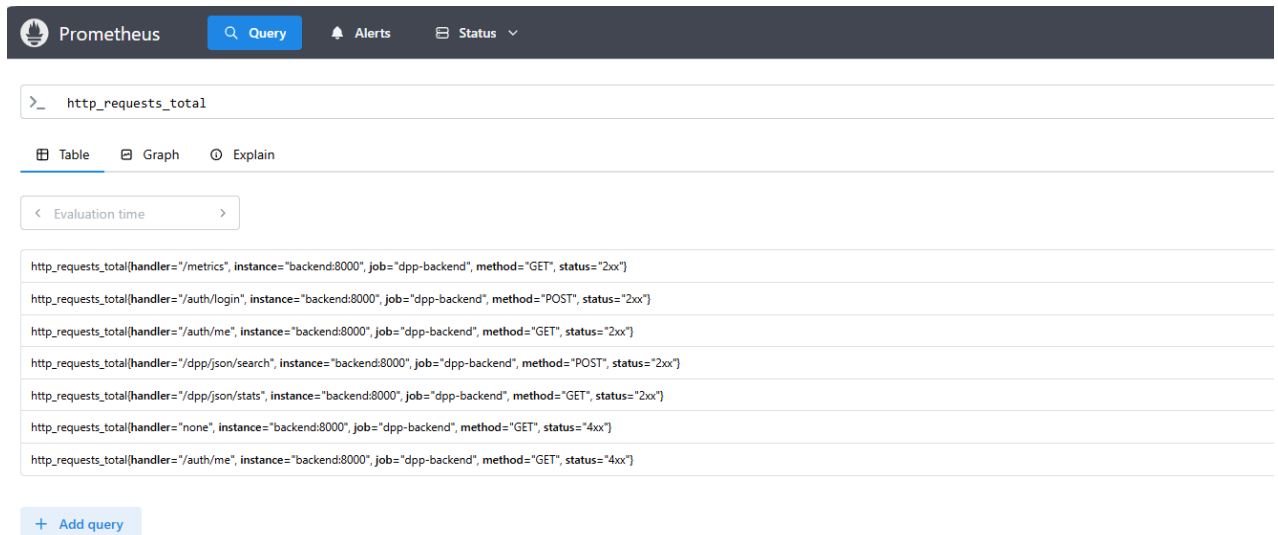


Εικόνα 82: Λεπτομερής απεικόνιση των panels *API Performance* και *State Timeline*, όπου καταγράφεται η σταθερή απόκριση του συστήματος κάτω από 5ms και η απουσία σφαλμάτων κατά τη διάρκεια των μετρήσεων.

### Συλλογή Δεδομένων και Υποδομή Prometheus

Η τροφοδοσία του dashboard με δεδομένα βασίζεται στη στενή συνεργασία του backend με τον εξυπηρετητή **Prometheus**. Η εφαρμογή εκθέτει ένα ειδικό endpoint (`/metrics`), το οποίο ο Prometheus «σαρώνει» (scrape) σε τακτά χρονικά διαστήματα (κάθε 5 δευτερόλεπτα).

Όπως φαίνεται στην Εικόνα 83, ο Prometheus καταγράφει τα αιτήματα ως μετρητές (counters) συνοδευόμενους από ετικέτες (labels) όπως handler, method και status. Αυτή η πολυδιάστατη δομή δεδομένων είναι που επιτρέπει στο Grafana να εκτελεί σύνθετα φίλτραρίσματα και να διαχωρίζει, για παράδειγμα, την κίνηση των SPARQL ερωτημάτων από τις απλές κλήσεις ελέγχου υγείας.



**Εικόνα 83: Προβολή των ακατέργαστων μετρήσεων (raw metrics) στη διεπαφή του Prometheus. Διακρίνεται ο μετρητής `http_requests_total` και ο διαχωρισμός των δεδομένων ανά `endpoint` και `status code`.**