



### **Π10. Ηλεκτρονική Πλατφόρμα Επανεμπορίου REFASHION**

SUB1.1: Συμπράξεις Ερευνητικής Αριστείας / Clusters of Research Excellence (CREs) - ΟΠΣ ΤΑ 5180519

#### **ΤΙΤΛΟΣ ΕΡΓΟΥ**

«Καινοτόμα Πλατφόρμα Ηλεκτρονικού Επανεμπορίου βασισμένη στην Τεχνολογία Blockchain για την Ενίσχυση του Κυκλικού και Βιώσιμου Μοντέλου Οικονομίας στη Βιομηχανία της Μόδας»

**ΑΚΡΩΝΥΜΙΟ:** REFASHION

**ΚΩΔΙΚΟΣ ΕΡΓΟΥ ΣΤΟ ΠΣΚΕ:** ΥΠ3ΤΑ-0560500



The present research is carried out within the framework of the National Recovery and Resilience Plan Greece 2.0, funded by the European Union – NextGenerationEU – Project Code: TAEDR-0560500 - Acronym: REFASHION).

Η παρούσα έρευνα υλοποιείται στο πλαίσιο του Εθνικού Σχεδίου Ανάκαμψης και Ανθεκτικότητας «Ελλάδα 2.0», με χρηματοδότηση από την Ευρωπαϊκή Ένωση – NextGenerationEU – Κωδικός Έργου: TAEDR-0560500 – Ακρωνύμιο: REFASHION

**Ελλάδα 2.0 ΕΘΝΙΚΟ  
ΕΘΝΙΚΟ ΣΧΕΔΙΟ ΑΝΑΚΑΜΨΗΣ και ΑΝΘΕΚΤΙΚΟΤΗΤΑΣ**

Συμπράξεις Ερευνητικής Αριστείας – ΣΕΑ

**ΕΡΓΟ: «Καινοτόμα Πλατφόρμα Ηλεκτρονικού Επανεμπορίου βασισμένη στην Τεχνολογία Blockchain για την Ενίσχυση του Κυκλικού και Βιώσιμου Μοντέλου Οικονομίας στη Βιομηχανία της Μόδας»**

**ΚΩΔΙΚΟΣ ΕΡΓΟΥ: ΥΠ3ΤΑ-0560500**

**ΠΑΡΑΔΟΤΕΟ Π10: ΗΛΕΚΤΡΟΝΙΚΗ ΠΛΑΤΦΟΡΜΑ ΕΠΑΝΕΜΠΟΡΙΟΥ REFASHION**

Διάρκεια Έργου:	15 μήνες
Ημερομηνία Παράδοσης:	30/05/2026
Υπεύθυνος Παραδοτέου:	ΠΑ.ΠΕΛ.
Επίπεδο Διάχυσης:	Δημόσιο
Τρέχουσα Κατάσταση Παραδοτέου:	Τελικό
Έκδοση:	F
Ιστοσελίδα Έργου:	<a href="https://refashion.cyclefi.com/">https://refashion.cyclefi.com/</a>

**ΦΟΡΕΙΣ ΥΛΟΠΟΙΗΣΗΣ:**



**E.K.A.N.**  
HELLENIC RECYCLING CENTER S.A.

**NEO**  
TEXTILE

The present research is carried out within the framework of the National Recovery and Resilience Plan Greece 2.0, funded by the European Union – NextGenerationEU – Project Code: TAEDR-0560500 - Acronym: REFASHION).

Η παρούσα έρευνα υλοποιείται στο πλαίσιο του Εθνικού Σχεδίου Ανάκαμψης και Ανθεκτικότητας «Ελλάδα 2.0», με χρηματοδότηση από την Ευρωπαϊκή Ένωση – NextGenerationEU – Κωδικός Έργου: TAEDR-0560500 – Ακρωνύμιο: REFASHION

## Περιεχόμενα

<b>1.</b>	<b>Διοικητική Σύνοψη (Executive Summary)</b> .....	<b>9</b>
<b>2.</b>	<b>Πεδίο Εφαρμογής του Παραδοτέου</b> .....	<b>10</b>
<b>3.</b>	<b>Αρχιτεκτονική Πλατφόρμας</b> .....	<b>12</b>
<b>3.1.</b>	<b>Headless Commerce</b> .....	<b>12</b>
<b>3.2.</b>	<b>Composable Commerce</b> .....	<b>13</b>
<b>3.3.</b>	<b>Επιλογή της Τεχνολογικής Βάσης (MedusaJS &amp; Mercur)</b> .....	<b>14</b>
3.3.1	Συνολική Αρχιτεκτονική Συστήματος.....	16
3.3.2	Επισκόπηση Πολυεπίπεδης Αρχιτεκτονικής .....	17
3.3.3	Διάγραμμα Συνολικής Αρχιτεκτονικής.....	19
3.3.4	Συνιστώσες Frontend .....	21
3.3.4.1	Web Storefront .....	21
3.3.4.2	Εφαρμογή Κινητού.....	23
3.3.4.3	Seller Admin Panel .....	24
3.3.4.4	Medusa Admin Panel .....	26
3.3.5	Συνιστώσες Backend & Βάση Δεδομένων .....	28
3.3.5.1	Medusa/Mercur Backend .....	28
3.3.5.2	Επίπεδο Επιχειρησιακής Λογικής .....	30
3.3.5.3	Βάση Δεδομένων PostgreSQL.....	32
3.3.5.4	Μηχανή Αναζήτησης & Αποθήκευση Αρχείων.....	35
3.3.6	Εξωτερικές Υπηρεσίες & Microservices .....	37
3.3.6.1	Stripe Connect.....	38
3.3.6.2	AI Microservices.....	39
3.3.7	Ροές Επικοινωνίας.....	43
3.3.7.1	Σύγχρονη Επικοινωνία (Request/Response) .....	44
3.3.7.2	Ασύγχρονη Επικοινωνία (Event-Driven) .....	46
3.3.7.3	Παράδειγμα End-to-End Ροής .....	49
<b>3.4.</b>	<b>Μοντέλο Δεδομένων</b> .....	<b>54</b>
3.4.1	Βασικές Οντότητες Medusa/Mercur.....	55
3.4.2	Custom Οντότητες REFASHION .....	58
3.4.3	Επέκταση μέσω Metadata .....	61
<b>4.</b>	<b>Βασικές Λειτουργίες E-Commerce</b> .....	<b>65</b>
<b>4.1.</b>	<b>Διαχείριση Χρηστών</b> .....	<b>65</b>
4.1.1	Υποστήριξη Ρόλων (Ιδιώτες vs. Επιχειρήσεις) .....	65
4.1.2	Εγγραφή και Αυθεντικοποίηση.....	65
<b>4.2.</b>	<b>Διαχείριση Προϊόντων Μόδας</b> .....	<b>69</b>
4.2.1	Καταχώρηση Προϊόντων (List your Item).....	69
4.2.2	Χαρακτηριστικά Προϊόντων .....	72
4.2.3	Αποθήκευση Εικόνων.....	72

<b>4.3.</b>	<b>Διαχείριση Συναλλαγών</b> .....	<b>73</b>
4.3.1	Διαδικασία Αγοράς (Cart & Order Workflow) .....	73
4.3.2	Πληρωμές.....	75
4.3.3	Αποστολές.....	75
<b>4.4.</b>	<b>Αναζήτηση &amp; Φιλτράρισμα</b> .....	<b>76</b>
4.4.1	Αναζήτηση Προϊόντων .....	76
4.4.2	Φίλτρα Καταλόγου (Αριστερό Sidebar) .....	76
4.4.3	Τεχνική Ροή Εφαρμογής.....	77
<b>5.</b>	<b>Frontend Υλοποίηση</b> .....	<b>78</b>
<b>5.1.</b>	<b>Web Storefront</b> .....	<b>78</b>
<b>5.2.</b>	<b>Admin Panel</b> .....	<b>84</b>
5.2.1	Seller Admin Panel / Vendor Panel .....	84
5.2.2	Medusa Admin Panel .....	90
<b>5.3.</b>	<b>Σχεδιαστικές Προδιαγραφές (Design Specifications)</b> .....	<b>91</b>
5.3.1	Σχεδιαστική Φιλοσοφία & Visual Identity .....	91
5.3.2	Design System & Component Library .....	93
5.3.3	Πρότυπα Διεπαφής & Layout Patterns.....	96
5.3.4	Εργαλεία & Διαδικασία Σχεδιασμού.....	100
<b>6.</b>	<b>Υπηρεσίες Τεχνητής Νοημοσύνης (AI Services)</b> .....	<b>102</b>
<b>6.1.</b>	<b>Image Analysis Service</b> .....	<b>102</b>
<b>6.2.</b>	<b>Visual Product Intelligence Service</b> .....	<b>103</b>
<b>6.3.</b>	<b>Recommendation Service</b> .....	<b>104</b>
<b>6.4.</b>	<b>Υποδομή Δεδομένων</b> .....	<b>104</b>
<b>7.</b>	<b>Υποσύστημα Ανταμοιβών RFT - Υλοποίηση &amp; Διασύνδεση</b> .....	<b>106</b>
<b>7.1.</b>	<b>Σκοπός &amp; Αρχιτεκτονική Προσέγγιση</b> .....	<b>106</b>
<b>7.2.</b>	<b>Επισκόπηση Tokenomics</b> .....	<b>106</b>
<b>7.3.</b>	<b>Επέκταση Μοντέλου Δεδομένων</b> .....	<b>107</b>
<b>7.4.</b>	<b>RFT API Client &amp; Integration Layer</b> .....	<b>107</b>
<b>7.5.</b>	<b>Backend Endpoints &amp; Event-Driven Ροές</b> .....	<b>107</b>
<b>7.6.</b>	<b>Παρατηρήσεις Υλοποίησης</b> .....	<b>108</b>
<b>8.</b>	<b>Τεκμηρίωση API Endpoints</b> .....	<b>109</b>
<b>8.1.</b>	<b>Αυθεντικοποίηση &amp; Διαχείριση Χρηστών</b> .....	<b>109</b>
<b>8.2.</b>	<b>Διαχείριση Προϊόντων</b> .....	<b>109</b>
<b>8.3.</b>	<b>Ροή Αγοράς (Cart Flow)</b> .....	<b>110</b>
<b>8.4.</b>	<b>Σύστημα Ανταμοιβών</b> .....	<b>110</b>

8.5.	Βοηθητικά Endpoints .....	111
<b>9.</b>	<b>Ασφάλεια &amp; Συμμόρφωση GDPR .....</b>	<b>112</b>
9.1.	Σχεδιαστικές Αρχές Ασφάλειας .....	112
9.2.	Αυθεντικοποίηση & Εξουσιοδότηση .....	112
9.3.	Προστασία Επικοινωνιών & Δεδομένων .....	113
9.4.	Ασφάλεια Πληρωμών & PCI DSS .....	114
9.5.	Συμμόρφωση με GDPR.....	114
<b>10.</b>	<b>Ανάπτυξη &amp; Φιλοξενία (Deployment) .....</b>	<b>116</b>
10.1.	Τρέχουσα Υποδομή Φιλοξενίας .....	116
10.2.	Παράμετροι Πρόσβασης .....	116
10.3.	Σχεδιαστικές Στοχεύσεις για Μελλοντική Παραγωγική Λειτουργία .....	117
<b>11.</b>	<b>Μεθοδολογία Υλοποίησης.....</b>	<b>118</b>
11.1.	Φιλοσοφία Σταδιακής Υλοποίησης .....	118
11.2.	Δομή των Φάσεων Υλοποίησης.....	118
11.3.	11.3 Συντονισμός Ομάδων και Διαχείριση Έργου.....	120
<b>12.</b>	<b>Συμπεράσματα .....</b>	<b>121</b>
12.1.	Επισκόπηση Επιτευγμάτων .....	121
12.2.	Σημαντικές Αρχιτεκτονικές Συμβολές .....	122
12.3.	Επόμενα Βήματα και Μελλοντική Πορεία .....	122

## Κατάλογος Σχημάτων

Εικόνα 1: Η Αρχιτεκτονική της Πλατφόρμας REFASHION.....	19
Εικόνα 1 - Αρχική σελίδα Storefront.....	78
Εικόνα 2 - Σελίδα εισόδου χρήστη.....	79
Εικόνα 3 - Σελίδα εγγραφής χρήστη .....	79
Εικόνα 4 - Σελίδα καταλόγου προϊόντων .....	80
Εικόνα 5 - Σελίδα συγκεκριμένης κατηγορίας προϊόντων.....	80
Εικόνα 6 - Σελίδα αποτελεσμάτων αναζήτησης με τον όρο "shirt" .....	81
Εικόνα 7 - Σελίδα καλαθιού χρήστη .....	81
Εικόνα 8 - Σελίδα checkout.....	82
Εικόνα 9 - Σελίδα στατιστικών χρήστη .....	82
Εικόνα 10 - Σελίδα Επεξεργασίας προφίλ χρήστη που τα στοιχεία χρησιμοποιούνται για AI product suggestions .....	83
Εικόνα 11 - Προτεινόμενα προϊόντα .....	83
Εικόνα 12 - responsive σχεδιασμός.....	84
Εικόνα 13 - responsive σχεδιασμός.....	84
Εικόνα 14 - Dashboard σελίδας πωλητή.....	85
Εικόνα 15 – Οθόνη δημιουργίας προϊόντος – ανέβασμα εικόνας και εκτέλεση AI service: οπτική αναζήτηση και εμπλουτισμό πληροφοριών προϊόντος .....	86
Εικόνα 16 – Οθόνη δημιουργίας προϊόντος – αποτελέσματα εμπλουτισμού πληροφοριών προϊόντος..	87
Εικόνα 17 - Οθόνη δημιουργίας προϊόντος – αυτόματη συμπλήρωση πεδίων .....	87
Εικόνα 18 - Οθόνη δημιουργίας προϊόντος - εκτέλεση AI service: αφαίρεση background .....	88
Εικόνα 19 - Οθόνη προϊόντων .....	88
Εικόνα 20 - Σελίδα λεπτομερειών παραγγελίας.....	89
Εικόνα 21 - Σελίδα δημιουργίας campaigns.....	89
Εικόνα 22 - Σελίδα διαχείρισης πωλητών.....	90

## 1. Διοικητική Σύνοψη (Executive Summary)

Το παρόν έγγραφο αποτελεί το Παραδοτέο Π10 του έργου REFASHION και τεκμηριώνει αναλυτικά την υλοποίηση της Ηλεκτρονικής Πλατφόρμας Επανεμπορίου (Re-commerce) μεταχειρισμένων ειδών μόδας. Παρέχει μια ολοκληρωμένη καταγραφή της αρχιτεκτονικής, των τεχνολογικών επιλογών, των λειτουργικών υποσυστημάτων και της τρέχουσας κατάστασης ανάπτυξης, ενώ ταυτόχρονα λειτουργεί ως σημείο αναφοράς για τις επόμενες φάσεις του έργου.

Η σύνταξη του εγγράφου εξυπηρετεί ένα σύνολο διακριτών στόχων, οι οποίοι αντανακλούν τη διπλή του φύση τόσο ως τεχνικό κείμενο όσο και ως επίσημο παραδοτέο προς τον φορέα χρηματοδότησης. Ειδικότερα, επιδιώκεται η ολοκληρωμένη τεκμηρίωση της αρχιτεκτονικής της πλατφόρμας, με σαφή αποτύπωση των επιλογών σχεδιασμού και των αρχών που τη διέπουν, καθώς και η καταγραφή των τεχνολογικών συνιστωσών που έχουν ενσωματωθεί, σε συνάρτηση με τη λογική επιλογής τους έναντι των απαιτήσεων του έργου. Παράλληλα, παρουσιάζονται οι λειτουργικές ροές που υλοποιούν τις βασικές δυνατότητες του οικοσυστήματος, όπως η διαχείριση χρηστών, η καταχώρηση προϊόντων, η ολοκλήρωση αγοράς, η αποστολή και η πληρωμή.

Σημαντικό μέρος του παραδοτέου αφιερώνεται στην τεκμηρίωση των API endpoints που διασυνδέουν το backend με τα διαφορετικά frontend περιβάλλοντα - το storefront, την εφαρμογή κινητού, το vendor panel και το admin panel - καθώς και στην αποτύπωση των ενσωματωμένων υπηρεσιών Τεχνητής Νοημοσύνης, οι οποίες αποτελούν κομβικό στοιχείο διαφοροποίησης της πλατφόρμας έναντι παραδοσιακών e-commerce λύσεων. Επιπλέον, καταγράφεται η πρόοδος υλοποίησης, εντοπίζοντας τα ολοκληρωμένα τμήματα και τα ζητήματα που παραμένουν υπό ανάπτυξη ή απαιτούν περαιτέρω επίλυση, ώστε να σχηματίζεται μια διαφανής εικόνα της τρέχουσας κατάστασης.

Το έγγραφο απευθύνεται σε πολλαπλά ακροατήρια με διαφορετικές ανάγκες πληροφόρησης. Στην ομάδα ανάπτυξης προσφέρει έναν τεχνικό οδηγό για την κατανόηση της αρχιτεκτονικής και τη συνέχιση της υλοποίησης. Στους διαχειριστές και επιστημονικούς υπεύθυνους του έργου λειτουργεί ως εργαλείο εποπτείας της προόδου και λήψης αποφάσεων. Προς τον φορέα χρηματοδότησης και τους αξιολογητές αποτελεί την επίσημη απόδειξη εκτέλεσης του παραδοτέου εντός του προβλεπόμενου χρονοδιαγράμματος. Τέλος, για τους εξωτερικούς συνεργάτες και τους λοιπούς εμπλεκόμενους φορείς, παρέχει ένα σαφές πλαίσιο αναφοράς για τη συνεργασία και την ενσωμάτωση με τα υφιστάμενα υποσυστήματα.

Το παρόν κείμενο σχεδιάστηκε ώστε να αποτυπώνει την κατάσταση της πλατφόρμας με τρόπο διαφανή και αντικειμενικό, αναγνωρίζοντας τόσο τα στοιχεία που έχουν υλοποιηθεί και βρίσκονται σε λειτουργική κατάσταση, όσο και εκείνα που παραμένουν σε εξέλιξη ή θα ολοκληρωθούν σε μεταγενέστερες φάσεις του έργου. Με την προσέγγιση αυτή, λειτουργεί ως τεχνικός οδηγός και σημείο αναφοράς καθ' όλη τη διάρκεια της υλοποίησης, υποστηρίζοντας την ομαλή εξέλιξη του έργου REFASHION έως την τελική του παράδοση.

## 2. Πεδίο Εφαρμογής του Παραδοτέου

Το REFASHION αποτελεί ένα ερευνητικό και αναπτυξιακό έργο που εστιάζει στη δημιουργία μιας καινοτόμου, ψηφιακής υποδομής για την επανέμπορία (re-commerce) μεταχειρισμένων ειδών ένδυσης και υπόδησης. Η σύλληψη του έργου εδράζεται στην αναγκαιότητα αντιμετώπισης ενός από τα πιο πιεστικά περιβαλλοντικά ζητήματα της εποχής, καθώς η βιομηχανία της μόδας συγκαταλέγεται στους μεγαλύτερους ρυπαντές παγκοσμίως, ευθυνόμενη για σημαντικό μερίδιο εκπομπών αερίων του θερμοκηπίου, υπέρμετρη κατανάλωση νερού και τεράστιο όγκο κλωστοϋφαντουργικών αποβλήτων που καταλήγουν σε χωματερές. Το γραμμικό μοντέλο παραγωγής και κατανάλωσης που κυριαρχεί στον κλάδο - με την επικράτηση των πρακτικών της fast fashion - έχει επιδεινώσει το φαινόμενο, καθιστώντας επιτακτική τη μετάβαση σε ένα πιο κυκλικό και βιώσιμο μοντέλο οικονομίας. Ο πυρήνας του έργου συνίσταται στην ανάπτυξη μιας αποκεντρωμένης πλατφόρμας ηλεκτρονικού επανεμπορίου, η οποία επιδιώκει να παρατείνει σημαντικά τον κύκλο ζωής των προϊόντων μόδας και να ενθαρρύνει τους καταναλωτές να υιοθετήσουν πιο συνειδητές αγοραστικές συνήθειες. Η πλατφόρμα δεν λειτουργεί απλώς ως ένας ακόμη χώρος αγοραπωλησίας μεταχειρισμένων ειδών, αλλά φιλοδοξεί να αποτελέσει ένα ολοκληρωμένο ψηφιακό οικοσύστημα που μετασχηματίζει τη συμμετοχή στην κυκλική οικονομία σε μια ελκυστική, ασφαλή και τεχνολογικά προηγμένη εμπειρία. Πέρα από την κλασική αγοραπωλησία, υποστηρίζονται εναλλακτικές μορφές κυκλικής συναλλαγής, όπως οι δωρεές και οι ανταλλαγές προϊόντων, διευρύνοντας τις επιλογές των χρηστών και ενισχύοντας την επαναχρησιμοποίηση ως κοινωνική πρακτική.

Το παρόν παραδοτέο εστιάζει στην τεκμηρίωση της ηλεκτρονικής πλατφόρμας του έργου REFASHION, όπως αυτή έχει διαμορφωθεί **στο τέλος του έργου**. Καλύπτει το σύνολο των συνιστωσών που συνθέτουν την πλατφόρμα, από την αρχιτεκτονική σχεδίαση και την ανάπτυξη του backend, έως τα frontend περιβάλλοντα, τα custom plugins και τις βοηθητικές υπηρεσίες Τεχνητής Νοημοσύνης. Στόχος είναι η δημιουργία μιας ενιαίας και συνεκτικής εικόνας της πλατφόρμας, στην οποία να αποτυπώνονται όχι μόνο τα αυτόνομα δομικά στοιχεία, αλλά και ο τρόπος με τον οποίο αυτά αλληλεπιδρούν για να υποστηρίξουν τις λειτουργικές απαιτήσεις του έργου.

Στο επίπεδο του αρχιτεκτονικού σχεδιασμού, το έγγραφο παρουσιάζει τις θεμελιώδεις αρχές που διέπουν την υλοποίηση, εξηγώντας τις στρατηγικές επιλογές που οδήγησαν στην υιοθέτηση μιας headless και composable προσέγγισης. Αναλύονται οι λόγοι επιλογής της κάθε τεχνολογικής συνιστώσας, οι τρόποι σύνδεσής τους και οι συμβιβασμοί που έγιναν στο σχεδιαστικό επίπεδο, προσφέροντας στον αναγνώστη μια σαφή κατανόηση όχι μόνο του τι έχει υλοποιηθεί, αλλά και του γιατί έχει υλοποιηθεί με τον συγκεκριμένο τρόπο.

Η ανάπτυξη του backend καλύπτεται εκτενώς, με αναλυτική παρουσίαση των υποσυστημάτων διαχείρισης χρηστών, καταχώρησης και διαχείρισης προϊόντων, εκτέλεσης συναλλαγών, ολοκλήρωσης πληρωμών και διαχείρισης αποστολών. Δίνεται έμφαση στις προσαρμογές που πραγματοποιήθηκαν για την κάλυψη των ιδιαίτερων απαιτήσεων ενός marketplace επανεμπορίου, όπως η υποστήριξη πολλαπλών πωλητών και ο διαχωρισμός των ρόλων εντός του οικοσυστήματος. Αντίστοιχα, η frontend υλοποίηση καλύπτεται μέσω της παρουσίασης των διακριτών περιβαλλόντων που έχουν αναπτυχθεί, δηλαδή του Web Storefront που απευθύνεται στους τελικούς αγοραστές, της εφαρμογής κινητού που προσφέρει mobile-first εμπειρία, του Seller Admin Panel για τους εγκεκριμένους πωλητές και του Medusa Admin Panel για τους διαχειριστές της πλατφόρμας.

Ξεχωριστή ενότητα αφιερώνεται στις καινοτόμες λειτουργίες που διαφοροποιούν το

REFASHION από μια τυπική e-commerce λύση, συμπεριλαμβανομένων των μηχανισμών gamification και υπολογισμού του Eco-Impact Score, καθώς και των εναλλακτικών μορφών συναλλαγής όπως οι δωρεές και οι ανταλλαγές. Παράλληλα, το παραδοτέο παρουσιάζει τις ενσωματωμένες υπηρεσίες Τεχνητής Νοημοσύνης, οι οποίες λειτουργούν ως αυτόνομα *microservices* και υποστηρίζουν την αυτόματη επεξεργασία εικόνων, την οπτική αναζήτηση και την παραγωγή προσωποποιημένων προτάσεων.

Σημαντικό τμήμα του εγγράφου αποτελεί η τεκμηρίωση των API endpoints, η οποία λειτουργεί ως τεχνική αναφορά για τη διασύνδεση των διαφορετικών συνιστωσών της πλατφόρμας. Παρουσιάζονται οι ροές αυθεντικοποίησης, οι λειτουργίες διαχείρισης προϊόντων, η ροή αγοράς, οι μηχανισμοί ανταμοιβών και τα βοηθητικά endpoints που υποστηρίζουν τις υπόλοιπες λειτουργίες. Επιπλέον, καταγράφεται η τελική κατάσταση υλοποίησης, με σαφή αποτύπωση των ολοκληρωμένων τμημάτων και των εκκρεμοτήτων που εντοπίστηκαν κατά την ανάλυση διαθεσιμότητας API και την ενσωμάτωση μεταξύ των διαφορετικών συστημάτων.

Πέρα από τις τεχνικές πτυχές, το παραδοτέο αναφέρεται και σε οριζόντια ζητήματα που διέπουν το σύνολο της πλατφόρμας, όπως η ασφάλεια, η συμμόρφωση με τον Γενικό Κανονισμό Προστασίας Δεδομένων (GDPR), η στρατηγική deployment, καθώς και η μεθοδολογία που έχει υιοθετηθεί για τη σταδιακή υλοποίηση του έργου. Τα στοιχεία αυτά εξασφαλίζουν ότι η πλατφόρμα δεν αντιμετωπίζεται μόνο ως ένα σύνολο τεχνικών συνιστωσών, αλλά ως ένα ολοκληρωμένο προϊόν λογισμικού το οποίο πρέπει να ανταποκρίνεται σε λειτουργικές, κανονιστικές και επιχειρησιακές απαιτήσεις.

Συνολικά, το πεδίο εφαρμογής του Π10 διαμορφώνεται με γνώμονα την πλήρη και διαφανή απεικόνιση της τεχνικής διάστασης της πλατφόρμας, παρέχοντας στους εμπλεκόμενους φορείς όλα τα στοιχεία που χρειάζονται για την κατανόηση, την αξιολόγηση και τη συνέχιση του έργου, χωρίς να επεκτείνεται σε θεματικές που εξυπηρετούνται αυτοτελώς από άλλα παραδοτέα.

### 3. Αρχιτεκτονική Πλατφόρμας

Η αρχιτεκτονική φιλοσοφία που υιοθετήθηκε για την υλοποίηση της πλατφόρμας REFASHION βασίζεται σε δύο σύγχρονες αρχές σχεδιασμού ηλεκτρονικών αγορών: το Headless Commerce και το Composable Commerce. Η συνδυαστική εφαρμογή των δύο αυτών παραδειγμάτων αποτελεί τη θεμελιώδη επιλογή που καθορίζει τόσο τις δυνατότητες όσο και τα όρια εξέλιξης της πλατφόρμας, και συνεπώς αξίζει να αναλυθεί ως αυτόνομη θεματική προτού παρουσιαστούν οι επιμέρους τεχνολογικές συνιστώσες.

#### 3.1. Headless Commerce

Η αρχή του Headless Commerce αφορά τον αρχιτεκτονικό διαχωρισμό μεταξύ του εμπορικού πυρήνα μιας πλατφόρμας (commerce engine) και των διεπαφών χρήστη που λειτουργούν πάνω σε αυτόν. Σε μια παραδοσιακή, μονολιθική προσέγγιση ηλεκτρονικού εμπορίου, το backend - δηλαδή ο μηχανισμός που διαχειρίζεται τα προϊόντα, τις παραγγελίες, τους χρήστες και τις συναλλαγές - είναι άρρηκτα συνδεδεμένο με το frontend, την ιστοσελίδα δηλαδή που βλέπει ο τελικός χρήστης. Η σύζευξη αυτή έχει σημαντικές πρακτικές συνέπειες: οποιαδήποτε αλλαγή στην εμφάνιση, τη λειτουργικότητα ή την εμπειρία χρήστη απαιτεί παρεμβάσεις που επηρεάζουν το σύνολο του συστήματος, ενώ η υποστήριξη πολλαπλών διαφορετικών διεπαφών (π.χ. ιστοσελίδα και εφαρμογή κινητού) γίνεται δυσχερής και ασύμβατη με τη ραγδαία μεταβαλλόμενη φύση των ψηφιακών καναλιών.

Η headless προσέγγιση καταργεί αυτόν τον περιορισμό, μετατρέποντας το backend σε έναν αυτόνομο, ανεξάρτητο μηχανισμό που εκθέτει τις λειτουργίες και τα δεδομένα του μέσω ενός πλήρους και σταθερού API (Application Programming Interface). Με τον τρόπο αυτό, το backend ασχολείται αποκλειστικά με τη διαχείριση της εμπορικής λογικής - τη συντήρηση των δεδομένων, την επιβολή των επιχειρησιακών κανόνων, την εκτέλεση των workflows - ενώ το frontend λειτουργεί ως ένας ή περισσότεροι ανεξάρτητοι «καταναλωτές» του API, οι οποίοι ευθύνονται αποκλειστικά για την παρουσίαση και την αλληλεπίδραση με τον χρήστη. Η επικοινωνία μεταξύ των δύο επιπέδων γίνεται μέσω τυποποιημένων αιτημάτων HTTP, επιτρέποντας στις δύο πλευρές να εξελισσονται παράλληλα και ανεξάρτητα, αρκεί να διατηρείται η συμβατότητα στο επίπεδο του API contract.

Για το REFASHION, η υιοθέτηση της headless αρχιτεκτονικής δεν αποτελεί απλώς μια τεχνολογική επιλογή, αλλά μια στρατηγική προϋπόθεση για τη βιωσιμότητα του έργου. Η πλατφόρμα προβλέπεται να υποστηρίξει πολλαπλά frontend περιβάλλοντα, καθένα με διαφορετικό σκοπό και κοινό: το Web Storefront που απευθύνεται στους τελικούς αγοραστές μέσω browser, την εφαρμογή κινητού για την προσφορά μιας mobile-first εμπειρίας σε iOS και Android, το Seller Admin Panel για τους εγκεκριμένους πωλητές που διαχειρίζονται τον δικό τους ψηφιακό κατάλογο, και το Medusa Admin Panel για τους διαχειριστές της πλατφόρμας. Όλα τα παραπάνω περιβάλλοντα αντλούν δεδομένα και εκτελούν ενέργειες μέσω του ίδιου κεντρικού backend, χωρίς διπλασιασμό επιχειρησιακής λογικής και χωρίς απόκλιση στη συμπεριφορά μεταξύ των καναλιών.

Η αποσύνδεση αυτή προσφέρει επιπρόσθετα μια σειρά από πρακτικά οφέλη που είναι κρίσιμα για το REFASHION. Πρώτον, παρέχει δημιουργική ελευθερία στον σχεδιασμό της εμπειρίας χρήστη, καθώς η ομάδα frontend δεν περιορίζεται από τα templates ή τις σχεδιαστικές συμβάσεις μιας μονολιθικής πλατφόρμας. Δεύτερον, επιτρέπει την εξατομικευμένη βελτιστοποίηση κάθε καναλιού - η εφαρμογή κινητού μπορεί να αξιοποιήσει native δυνατότητες (κάμερα, push notifications, geolocation) χωρίς να επηρεαστεί ο σχεδιασμός του web περιβάλλοντος.

Τρίτον, διευκολύνει την τεχνολογική εξέλιξη του έργου: μελλοντικά κανάλια όπως φωνητικοί βοηθοί, smart displays ή IoT συσκευές μπορούν να ενσωματωθούν στο οικοσύστημα ως νέοι «πελάτες» του ίδιου API, χωρίς να απαιτείται αναδιάρθρωση του πυρήνα. Τέταρτον, η ανεξαρτησία των επιπέδων επιτρέπει την παράλληλη ανάπτυξη από διαφορετικές ομάδες - γεγονός που ισχύει στην περίπτωση του REFASHION, όπου το backend συντηρείται από την ομάδα του Πανεπιστημίου Πελοποννήσου ενώ η εφαρμογή κινητού αναπτύσσεται από άλλον εταίρο του έργου.

Τέλος, η headless προσέγγιση είναι καθοριστικής σημασίας για την ενσωμάτωση των καινοτόμων λειτουργιών του REFASHION. Τα custom υποσυστήματα της πλατφόρμας - όπως η ενσωμάτωση Blockchain, ο υπολογισμός του Eco-Impact Score και οι μηχανισμοί gamification - μπορούν να εκτεθούν ως νέα API endpoints δίπλα στα τυπικά endpoints του εμπορικού πυρήνα, χωρίς να επηρεαστεί η σταθερότητα ή η συμπεριφορά των υπόλοιπων λειτουργιών. Με τον τρόπο αυτό, η headless φιλοσοφία δεν αποτελεί απλώς μια αρχή σχεδιασμού, αλλά αναδεικνύεται σε οργανωτική και τεχνική προϋπόθεση που υποστηρίζει το σύνολο του οράματος του έργου.

### 3.2. Composable Commerce

Η αρχή του Composable Commerce αποτελεί τη φυσική εξέλιξη της headless αρχιτεκτονικής και αναφέρεται στον τρόπο με τον οποίο συντίθεται μια ολοκληρωμένη πλατφόρμα ηλεκτρονικού εμπορίου από διακριτές, εξειδικευμένες υπηρεσίες. Σε αντίθεση με την παραδοσιακή προσέγγιση, όπου μία ενιαία πλατφόρμα επιχειρεί να καλύψει αυτόνομα το σύνολο των απαιτήσεων ενός marketplace - από τη διαχείριση καταλόγου και την εκτέλεση συναλλαγών έως την αναζήτηση, τις πληρωμές και τις ειδοποιήσεις - η composable φιλοσοφία προτείνει τη δημιουργία ενός οικοσυστήματος που αποτελείται από τις βέλτιστες διαθέσιμες υπηρεσίες για κάθε επιμέρους λειτουργία, συνδεδεμένες μεταξύ τους μέσω σταθερών διεπαφών API. Η προσέγγιση αυτή βασίζεται σε μια θεμελιώδη αναγνώριση: καμία μονολιθική πλατφόρμα δεν μπορεί να είναι ταυτόχρονα η καλύτερη επιλογή για κάθε επιμέρους λειτουργία. Μια πλατφόρμα που διαπρέπει στη διαχείριση καταλόγου ενδέχεται να υστερεί στις δυνατότητες αναζήτησης, ενώ ένας ισχυρός μηχανισμός πληρωμών μπορεί να συνοδεύεται από περιορισμένες δυνατότητες προσωποποίησης. Η composable αρχιτεκτονική απαντά σε αυτή την πραγματικότητα, επιτρέποντας στον σχεδιαστή της πλατφόρμας να επιλέξει την καλύτερη υπηρεσία για κάθε επιμέρους ανάγκη (best-of-breed), και να συνθέσει το τελικό σύστημα σαν μια εξατομικευμένη συναρμολόγηση εξειδικευμένων δομικών στοιχείων.

Το MedusaJS, που αποτελεί τη βάση του REFASHION, έχει σχεδιαστεί ακριβώς γύρω από αυτή τη φιλοσοφία. Λειτουργεί ως ένας κεντρικός κόμβος ορχήστρωσης (orchestration hub), ο οποίος αναλαμβάνει τις βασικές εμπορικές λειτουργίες - διαχείριση προϊόντων, παραγγελιών, πελατών και καλαθιών - ενώ ταυτόχρονα παρέχει σαφή σημεία επέκτασης για την ενσωμάτωση τρίτων υπηρεσιών. Η ενσωμάτωση αυτή υλοποιείται μέσω plugin-based αρχιτεκτονικής, όπου κάθε εξειδικευμένη λειτουργία ενσωματώνεται ως αυτόνομο πακέτο κώδικα, χωρίς να επηρεάζει τον πυρήνα του συστήματος ή τη συμπεριφορά των υπόλοιπων components. Για το REFASHION, η composable προσέγγιση μεταφράζεται σε μια σειρά από συγκεκριμένες σχεδιαστικές επιλογές που αξιοποιούν τις πιο κατάλληλες τεχνολογίες για κάθε επιμέρους λειτουργία. Στον τομέα της αναζήτησης και του φιλτραρίσματος, το οικοσύστημα μπορεί να αξιοποιήσει εξειδικευμένες μηχανές όπως το MeiliSearch ή το Algolia, οι οποίες προσφέρουν δυνατότητες instant search και faceted search δραματικά ανώτερες από αυτές μιας τυπικής βάσης δεδομένων.

Στον τομέα των πληρωμών, η ενσωμάτωση με τον πάροχο Stripe - και ειδικότερα με το Stripe Connect για multi-vendor υποστήριξη - προσφέρει PCI-compliant ασφάλεια και

ευρεία υποστήριξη μεθόδων πληρωμής χωρίς την ανάγκη ανάπτυξης ιδιόκτητης λύσης. Στον τομέα της αποθήκευσης αρχείων και πολυμέσων, η χρήση εξωτερικών υπηρεσιών όπως το Amazon S3 ή το MinIO επιτρέπει την κλιμακώσιμη και αξιόπιστη διαχείριση εικόνων προϊόντων, αποσυνδέοντας τη λογική του backend από τις λεπτομέρειες της φυσικής αποθήκευσης. Αντίστοιχα, στον τομέα των ειδοποιήσεων, υπηρεσίες όπως το SendGrid καλύπτουν τις ανάγκες email και push notifications, ενώ για τη συνομιλία μεταξύ αγοραστών και πωλητών αξιοποιείται το TalkJS.

Η πραγματική όμως αξία της composable αρχιτεκτονικής για το REFASHION αναδεικνύεται στην ενσωμάτωση των καινοτόμων λειτουργιών που διαφοροποιούν το έργο. Αυτές οι λειτουργίες - η σύνδεση με το Ethereum Blockchain για την καταγραφή συναλλαγών και την έκδοση ReFashion Tokens (RFTs), ο υπολογισμός του Eco-Impact Score, οι μηχανισμοί gamification και οι υπηρεσίες Τεχνητής Νοημοσύνης - δεν θα μπορούσαν να υλοποιηθούν σε μια κλειστή, μονολιθική πλατφόρμα χωρίς εκτεταμένες και επικίνδυνες παρεμβάσεις στον πυρήνα της. Η composable προσέγγιση επιτρέπει την ομαλή ενσωμάτωση των λειτουργιών αυτών ως αυτόνομα plugins ή εξωτερικά microservices, τα οποία επικοινωνούν με τον πυρήνα μέσω σαφώς ορισμένων διεπαφών. Με τον τρόπο αυτό, η ομάδα ανάπτυξης μπορεί να εστιάσει στις πραγματικά καινοτόμες πτυχές του έργου, χωρίς να ξοδεύει χρόνο στην επανεφεύρεση βασικών δυνατοτήτων e-commerce.

Η composable αρχιτεκτονική προσφέρει επιπλέον σημαντικά πλεονεκτήματα συντηρησιμότητας και ανθεκτικότητας. Επειδή κάθε υπηρεσία λειτουργεί ως αυτόνομο component με σαφή όρια ευθύνης, η αναβάθμιση, η αντικατάσταση ή η απομόνωση μιας μεμονωμένης υπηρεσίας δεν επηρεάζει το υπόλοιπο σύστημα. Εάν για παράδειγμα στο μέλλον το REFASHION αποφασίσει να αλλάξει πάροχο πληρωμών ή να αναβαθμίσει τη μηχανή αναζήτησης, η αλλαγή αυτή μπορεί να γίνει εστιασμένα, χωρίς να απαιτείται γενικότερη αναδιάρθρωση της πλατφόρμας. Αντίστοιχα, σε περίπτωση αστοχίας ενός εξωτερικού service, το υπόλοιπο σύστημα μπορεί να συνεχίσει να λειτουργεί με υποβαθμισμένες δυνατότητες (graceful degradation), αυξάνοντας τη συνολική ανθεκτικότητα του οικοσυστήματος.

Παράλληλα, η composable προσέγγιση διευκολύνει σημαντικά την κλιμάκωση της πλατφόρμας. Διαφορετικά components μπορούν να κλιμακωθούν ανεξάρτητα, ανάλογα με τις πραγματικές ανάγκες φόρτου του καθενός. Στο REFASHION, για παράδειγμα, οι υπηρεσίες AI που εκτελούν επεξεργασία εικόνων και reverse image search ενδέχεται να απαιτούν GPU-enabled υποδομές και να αντιμετωπίζουν διαφορετικά μοτίβα φόρτου από τον εμπορικό πυρήνα. Η αυτονομία τους ως microservices επιτρέπει την ανεξάρτητη κλιμάκωση και τη βελτιστοποίηση κόστους, καθώς δεν χρειάζεται να επεκταθεί ολόκληρη η υποδομή για να καλυφθούν οι ανάγκες ενός μόνο τμήματος. Συνοψίζοντας, η composable αρχιτεκτονική αποτελεί για το REFASHION πολύ περισσότερο από μια τεχνική επιλογή. Συνιστά μια στρατηγική σχεδιασμού που επιτρέπει στην πλατφόρμα να συνδυάζει τις καλύτερες διαθέσιμες τεχνολογίες για κάθε επιμέρους λειτουργία, να ενσωματώνει με σαφή και βιώσιμο τρόπο τις καινοτόμες λειτουργίες του έργου, και να εξελίσσεται με τον χρόνο χωρίς να εγκλωβίζεται στις αρχικές της επιλογές. Σε συνδυασμό με τη headless φιλοσοφία που αναλύθηκε προηγουμένως, η composable προσέγγιση παρέχει το αρχιτεκτονικό υπόβαθρο πάνω στο οποίο οικοδομούνται όλες οι επιμέρους τεχνολογικές αποφάσεις που παρουσιάζονται στις επόμενες ενότητες του παρόντος παραδοτέου.

### 3.3. Επιλογή της Τεχνολογικής Βάσης (MedusaJS & Mercur)

Η επιλογή της τεχνολογικής βάσης για την υλοποίηση της πλατφόρμας REFASHION αποτέλεσε μία από τις πιο κρίσιμες αποφάσεις στη φάση του αρχιτεκτονικού σχεδιασμού. Η απόφαση αυτή δεν αφορά απλώς την επιλογή ενός εργαλείου ή ενός framework, αλλά

καθορίζει το σύνολο του οικοσυστήματος εντός του οποίου θα αναπτυχθεί, θα συντηρηθεί και θα εξελιχθεί η πλατφόρμα κατά τη διάρκεια του κύκλου ζωής της. Έπειτα από τη συστηματική αξιολόγηση των διαθέσιμων εναλλακτικών έναντι των ιδιαίτερων απαιτήσεων του έργου, η επιλογή κατέληξε στο συνδυασμό του MedusaJS ως κεντρικού εμπορικού πυρήνα και του Mercur ως επέκτασης που υποστηρίζει τη λειτουργία marketplace πολλαπλών πωλητών.

Το MedusaJS αποτελεί ένα open-source headless commerce framework το οποίο τα τελευταία χρόνια έχει αναδειχθεί ως μία από τις πιο αξιοσημείωτες σύγχρονες πλατφόρμες στον χώρο του ηλεκτρονικού εμπορίου. Έχει σχεδιαστεί εξ αρχής με βάση τις αρχές του headless και composable commerce που αναλύθηκαν στις προηγούμενες ενότητες, και προσφέρει μια σύγχρονη τεχνολογική στοίβα βασισμένη σε Node.js και TypeScript. Η αρχιτεκτονική του οργανώνεται γύρω από τρεις πυλώνες - plugins, services και modules - οι οποίοι επιτρέπουν την επέκταση και την προσαρμογή του πυρήνα χωρίς να απαιτείται τροποποίηση του βασικού κώδικα. Με τον τρόπο αυτό, ο πυρήνας παραμένει καθαρός και αναβαθμίσιμος, ενώ οι εξατομικευμένες υλοποιήσεις του έργου περιορίζονται σε ξεχωριστά, καλά απομονωμένα τμήματα.

Παράλληλα με τις τεχνικές του αρετές, το MedusaJS προσφέρει out-of-the-box ένα ολοκληρωμένο σύνολο βασικών λειτουργιών e-commerce. Συγκεκριμένα, καλύπτει τη διαχείριση προϊόντων και των παραλλαγών τους (variants), τη διαχείριση πελατών και των ομάδων πελατών (customer groups), τη ροή του καλαθιού και της παραγγελίας (cart και order workflow), τη διαχείριση περιοχών και νομισμάτων (regions), τις συλλογές (collections) και τις ετικέτες (tags) προϊόντων, καθώς και τις θεμελιώδεις λειτουργίες αυθεντικοποίησης. Η ύπαρξη αυτής της έτοιμης λειτουργικότητας μειώνει δραστικά τον χρόνο ανάπτυξης (time-to-market) και επέτρεψε στην ομάδα του REFASHION να εστιάσει στα πραγματικά καινοτόμα στοιχεία του έργου, αντί να αναλώνει χρόνο στην επανεφεύρεση βασικών μηχανισμών που απαντώνται σε κάθε e-commerce πλατφόρμα.

Ο ανοιχτός κώδικας (open-source) του MedusaJS αποτέλεσε επιπλέον έναν από τους καθοριστικούς παράγοντες επιλογής. Σε αντίθεση με κλειστές, εμπορικές πλατφόρμες, το MedusaJS παρέχει πλήρη πρόσβαση και έλεγχο στον πηγαίο κώδικα, επιτρέποντας στην ομάδα ανάπτυξης να κατανοεί σε βάθος τη συμπεριφορά του συστήματος, να εντοπίζει και να επιδιορθώνει προβλήματα και να υλοποιεί προσαρμογές χωρίς εξαρτήσεις από εξωτερικούς προμηθευτές. Στο πλαίσιο ενός ερευνητικού έργου όπως το REFASHION, αυτή η διαφάνεια και ευελιξία δεν αποτελεί απλώς πρακτικό πλεονέκτημα, αλλά θεμελιώδη προϋπόθεση. Επιπλέον, η αποφυγή του φαινομένου του vendor lock-in εξασφαλίζει ότι η πλατφόρμα θα παραμείνει βιώσιμη και ανεξάρτητη μετά τη λήξη της χρηματοδοτούμενης περιόδου, χωρίς εξαρτήσεις από συγκεκριμένο εμπορικό προμηθευτή.

Από τεχνολογική σκοπιά, η επιλογή του MedusaJS υποστηρίζεται από τη σύγχρονη τεχνολογική στοίβα που το συνοδεύει. Η συγγραφή σε Node.js με TypeScript στο backend εξασφαλίζει την υψηλή ποιότητα και τη συντηρησιμότητα του κώδικα μέσω του στατικού τυποποιημένου συστήματος, ενώ το admin panel που βασίζεται σε React χρησιμοποιεί ένα ευρέως διαδεδομένο και ώριμο frontend framework.

Οι τεχνολογίες αυτές διευκολύνουν σημαντικά τόσο τη φάση εύρεσης εξειδικευμένου προσωπικού ανάπτυξης όσο και τη μακροπρόθεσμη συντήρηση της πλατφόρμας, καθώς ανήκουν στις κυρίαρχες επιλογές της σύγχρονης βιομηχανίας λογισμικού. Παράλληλα, η ενεργή και διαρκώς αναπτυσσόμενη κοινότητα του MedusaJS εξασφαλίζει τη συνεχή εξέλιξη του οικοσυστήματος, τη δημοσίευση νέων plugins από τρίτους, καθώς και την πρόσβαση σε εκτενή τεκμηρίωση και υλικό υποστήριξης.

Παρά τις θεμελιώδεις αρετές του MedusaJS, η εφαρμογή του στο REFASHION παρουσίαζε

μια κρίσιμη πρόκληση: ο πυρήνας του MedusaJS είναι σχεδιασμένος για single-vendor καταστήματα, όπου ένας μοναδικός φορέας διαχειρίζεται το σύνολο του καταλόγου των προϊόντων. Το REFASHION ωστόσο, λόγω της φύσης του ως πλατφόρμας επανεμπορίου, απαιτεί ένα πραγματικά πολυ-πωλητικό (multi-vendor) μοντέλο, όπου πολλοί ανεξάρτητοι πωλητές - ιδιώτες και επιχειρήσεις - διαχειρίζονται τους δικούς τους καταλόγους, παραγγελίες και αποθέματα μέσα στο ίδιο οικοσύστημα. Η κάλυψη αυτής της απαίτησης μέσω εκτεταμένων προσαρμογών του πυρήνα θα ήταν χρονοβόρα, επιρρεπής σε σφάλματα και θα δημιουργούσε σημαντικό τεχνικό χρέος για την ομάδα ανάπτυξης.

Η λύση σε αυτό το ζήτημα ήρθε με την υιοθέτηση του Mercur, μιας open-source επέκτασης του MedusaJS που είναι ειδικά σχεδιασμένη για τη μετατροπή του σε πλήρες marketplace framework. Το Mercur διατηρεί τα πλεονεκτήματα και την αρχιτεκτονική φιλοσοφία του MedusaJS, ενώ προσθέτει τις απαραίτητες δυνατότητες που χαρακτηρίζουν μια αυθεντική πλατφόρμα πολλαπλών πωλητών. Συγκεκριμένα, εισάγει την έννοια του Seller ως ξεχωριστή οντότητα δίπλα στους τυπικούς Customers και Admins, εμπλουτίζει το backend με τα εξειδικευμένα /vendor/\* endpoints που υποστηρίζουν τις λειτουργίες των πωλητών, και ενσωματώνει τα κατάλληλα ownership checks ώστε κάθε πωλητής να έχει πρόσβαση αποκλειστικά στα δικά του δεδομένα. Επιπλέον, εισάγει νέες λειτουργίες προσαρμοσμένες στις ανάγκες ενός marketplace, όπως τη ροή έγκρισης (moderation) νέων πωλητών και προϊόντων, τη διαχείριση προμηθειών (commissions) ανά πωλητή ή κατηγορία, και την παραγωγή ενοποιημένων order sets που διαχωρίζουν αυτόματα μια ενιαία παραγγελία αγοραστή σε επιμέρους παραγγελίες ανά πωλητή. Παράλληλα, παρέχει εξειδικευμένη ενσωμάτωση με το Stripe Connect για την υποστήριξη payouts προς τους πωλητές και τη διαχείριση των οικονομικών ροών του marketplace.

Η συνδυαστική επιλογή MedusaJS και Mercur αξιοποιείται μέσω τριών διακριτών μηχανισμών επέκτασης που προσφέρει το οικοσύστημα. Τα Plugins λειτουργούν ως αυτόνομα πακέτα κώδικα (Node.js packages) που εγκαθίστανται πάνω στο backend και προσθέτουν νέα data models, services, API endpoints και subscribers, αποτελώντας τον κύριο μηχανισμό υλοποίησης των καινοτομιών του REFASHION. Τα Services αποτελούν την επιχειρησιακή λογική του συστήματος, οργανωμένη σε κλάσεις TypeScript που αλληλεπιδρούν μεταξύ τους μέσω Dependency Injection, και επιτρέπουν τη δημιουργία custom λογικής (όπως ο υπολογισμός του Eco-Impact Score) χωρίς παρέμβαση στον πυρήνα. Τέλος, τα Modules προσφέρουν τη δυνατότητα αντικατάστασης ή επέκτασης βασικών λειτουργιών - όπως οι πληρωμές, οι αποστολές και οι ειδοποιήσεις - με custom υλοποιήσεις ή με ενσωματώσεις τρίτων παρόχων, διατηρώντας τη σαφή απομόνωση των ευθυνών εντός του συστήματος.

Συνολικά, η επιλογή του συνδυασμού MedusaJS και Mercur αποτέλεσε για το REFASHION την ιδανική τεχνολογική βάση: συνδυάζει την ωριμότητα και την επεκτασιμότητα ενός σύγχρονου headless commerce framework με τις εξειδικευμένες λειτουργίες ενός marketplace πολλαπλών πωλητών, παρέχοντας στην ομάδα ανάπτυξης ένα στιβαρό υπόβαθρο πάνω στο οποίο μπορούν να οικοδομηθούν οι πραγματικά καινοτόμες πτυχές του έργου - η ενσωμάτωση Blockchain, ο υπολογισμός Eco-Impact, οι μηχανισμοί gamification και οι υπηρεσίες Τεχνητής Νοημοσύνης. Με τον τρόπο αυτό, η αρχιτεκτονική φιλοσοφία που αναλύθηκε στις προηγούμενες υποπαραγράφους μετουσιώνεται σε συγκεκριμένες τεχνολογικές επιλογές, οι οποίες παρουσιάζονται αναλυτικά στις επόμενες ενότητες του παρόντος κεφαλαίου.

### 3.3.1 Συνολική Αρχιτεκτονική Συστήματος

Η συνολική αρχιτεκτονική του REFASHION έχει δομηθεί ως ένα πολυεπίπεδο, αποσυνδεδεμένο σύστημα (multi-tier, decoupled system), του οποίου η οργάνωση

αντανακλά άμεσα τις αρχές της headless και composable προσέγγισης που αναλύθηκαν στην προηγούμενη ενότητα. Η επιλογή μιας πολυεπίπεδης δόμησης δεν αποτελεί απλώς μια σχεδιαστική σύμβαση, αλλά μια συνειδητή στρατηγική επιλογή που εξυπηρετεί τη συντηρησιμότητα, την κλιμακωσιμότητα και την εξελισιμότητα της πλατφόρμας. Κάθε επίπεδο έχει σαφώς οριοθετημένες ευθύνες και επικοινωνεί με τα γειτονικά του μέσω τυποποιημένων διεπαφών, εξασφαλίζοντας ότι οποιαδήποτε αλλαγή ή αναβάθμιση σε ένα επίπεδο δεν επηρεάζει τη λειτουργία των υπολοίπων.

### 3.3.2 Επισκόπηση Πολυεπίπεδης Αρχιτεκτονικής

Η αρχιτεκτονική διαρθρώνεται σε πέντε διακριτά επίπεδα, καθένα από τα οποία επιτελεί έναν σαφώς καθορισμένο ρόλο εντός του συνολικού οικοσυστήματος. Στο ανώτερο επίπεδο τοποθετείται το Επίπεδο Παρουσίασης (Presentation Layer), το οποίο περιλαμβάνει όλες τις διεπαφές χρήστη μέσω των οποίων οι διαφορετικές κατηγορίες χρηστών αλληλεπιδρούν με την πλατφόρμα. Το επίπεδο αυτό απαρτίζεται από το Web Storefront που απευθύνεται στους τελικούς αγοραστές, την εφαρμογή κινητού που προσφέρει mobile-first εμπειρία σε iOS και Android, το Seller Admin Panel που χρησιμοποιούν οι εγκεκριμένοι πωλητές για τη διαχείριση των ψηφιακών τους καταστημάτων, και το Medusa Admin Panel που εξυπηρετεί τους διαχειριστές της πλατφόρμας στις λειτουργίες εποπτείας και moderation. Κάθε ένα από αυτά τα frontend περιβάλλοντα αναπτύσσεται και εξελίσσεται αυτόνομα, ωστόσο μοιράζεται την ίδια κεντρική πηγή δεδομένων και επιχειρησιακής λογικής, εξασφαλίζοντας απόλυτη συνέπεια στη συμπεριφορά του συστήματος ανεξάρτητα από το κανάλι πρόσβασης.

Ακριβώς κάτω από το επίπεδο παρουσίασης βρίσκεται το Επίπεδο API (API Layer), το οποίο λειτουργεί ως η κύρια διεπαφή επικοινωνίας μεταξύ των διεπαφών χρήστη και του εμπορικού πυρήνα. Το επίπεδο αυτό υλοποιείται μέσω του REST API που εκθέτει το Medusa/Mercur backend, και οργανώνεται σε τρία διακριτά scopes: τα /store/\* endpoints για τις λειτουργίες των αγοραστών, τα /vendor/\* endpoints για τις λειτουργίες των πωλητών και τα /admin/\* endpoints για τις διαχειριστικές λειτουργίες. Ο διαχωρισμός αυτός εξασφαλίζει σαφή απομόνωση των ευθυνών και επιβολή των κατάλληλων κανόνων ασφαλείας σε κάθε επίπεδο πρόσβασης. Επιπλέον, σε αυτό το επίπεδο τοποθετούνται και τα custom endpoints που υλοποιούν τις καινοτόμες λειτουργίες του REFASHION, καθώς και τα proxy routes που διασυνδέουν την πλατφόρμα με τις εξωτερικές υπηρεσίες Τεχνητής Νοημοσύνης.

Στον πυρήνα της αρχιτεκτονικής βρίσκεται το Επίπεδο Επιχειρησιακής Λογικής (Business Logic Layer), το οποίο ενσωματώνει το σύνολο των κανόνων και των workflows που διέπουν τη συμπεριφορά της πλατφόρμας. Στο επίπεδο αυτό υλοποιούνται τα core services του MedusaJS και του Mercur - όπως το ProductService, το OrderService, το CartService, το CustomerService και το SellerService - καθώς και τα custom services που έχουν αναπτυχθεί ειδικά για τις ανάγκες του REFASHION, όπως οι υπηρεσίες διαχείρισης των ReFashion Tokens, υπολογισμού του Eco-Impact Score και απόδοσης των gamification στοιχείων. Παράλληλα, το επίπεδο αυτό φιλοξενεί τους subscribers που αντιδρούν σε events του συστήματος (π.χ. order.placed, customer.created) και ενεργοποιούν τις αντίστοιχες ροές, καθώς και τους custom workflows που ενορχηστρώνουν πολύπλοκες λειτουργίες όπως η δημιουργία πολλαπλών παραγγελιών σε ένα order set.

Στο κάτω μέρος της αρχιτεκτονικής βρίσκεται το Επίπεδο Δεδομένων (Data Layer), υπεύθυνο για την επίμονη αποθήκευση και ανάκτηση όλων των πληροφοριών που διαχειρίζεται η πλατφόρμα. Πρωταρχικός μηχανισμός αποθήκευσης είναι η σχεσιακή βάση δεδομένων PostgreSQL, η οποία φιλοξενεί όλες τις βασικές οντότητες - προϊόντα,

παραγγελίες, πελάτες, πωλητές, καλάθια - καθώς και τις custom οντότητες του REFASHION όπως οι SwapRequest, GamificationProfile και BlockchainTransaction. Πέρα από τη σχεσιακή βάση, το επίπεδο αυτό περιλαμβάνει επιπλέον εξειδικευμένους μηχανισμούς αποθήκευσης για συγκεκριμένους τύπους δεδομένων: τη μηχανή αναζήτησης (MeiliSearch) που διατηρεί ευρετηριασμένες αναπαραστάσεις των προϊόντων για ταχεία αναζήτηση και φιλτράρισμα, καθώς και την υπηρεσία αποθήκευσης αρχείων (Amazon S3 ή MinIO) που φιλοξενεί τις εικόνες των προϊόντων και τα λοιπά πολυμέσα.

Τέλος, περιφερειακά της κεντρικής δομής λειτουργεί το Επίπεδο Εξωτερικών Υπηρεσιών (External Services Layer), το οποίο περιλαμβάνει όλες τις υπηρεσίες τρίτων με τις οποίες διασυνδέεται η πλατφόρμα για την κάλυψη εξειδικευμένων αναγκών. Σε αυτό το επίπεδο εντάσσονται ο πάροχος πληρωμών Stripe (μέσω της επέκτασης Stripe Connect), η υπηρεσία αποστολής ειδοποιήσεων SendGrid, η υπηρεσία ζωντανής συνομιλίας TalkJS, καθώς και - κρίσιμο για τις καινοτομίες του REFASHION - το Ethereum Blockchain για την καταγραφή των συναλλαγών και την έκδοση των ReFashion Tokens. Παράλληλα, σε αυτό το επίπεδο τοποθετούνται και τα αυτόνομα AI microservices που έχουν αναπτυχθεί στο πλαίσιο του έργου - το Image Analysis Service, το Visual Product Intelligence Service και το Recommendation Service - τα οποία λειτουργούν ως εξωτερικές υπηρεσίες ως προς τον εμπορικό πυρήνα, παρότι αναπτύσσονται και συντηρούνται από την ίδια ομάδα του έργου.

Η αλληλεπίδραση μεταξύ των επιπέδων ακολουθεί ένα σαφές και κατευθυνόμενο πρότυπο επικοινωνίας. Τα επίπεδα παρουσίασης απευθύνονται αποκλειστικά στο επίπεδο API μέσω αιτημάτων HTTP, χωρίς ποτέ να έχουν άμεση πρόσβαση στη βάση δεδομένων ή στην επιχειρησιακή λογική. Το επίπεδο API διοχετεύει τα αιτήματα στο επίπεδο επιχειρησιακής λογικής, το οποίο εκτελεί την αντίστοιχη επεξεργασία και αλληλεπιδρά με το επίπεδο δεδομένων και, όπου απαιτείται, με τις εξωτερικές υπηρεσίες. Η μονόδρομη ροή αυτή - από τα ανώτερα επίπεδα προς τα κατώτερα - εξασφαλίζει σαφή ιεραρχία ευθυνών και αποτρέπει τη δημιουργία κυκλικών εξαρτήσεων που θα δυσχέραιναν τη συντήρηση του συστήματος. Παράλληλα, η ασύγχρονη επικοινωνία μέσω του event bus του Medusa επιτρέπει στα συστήματα του business logic επιπέδου να αντιδρούν σε γεγονότα χωρίς άμεση εξάρτηση μεταξύ τους, ενισχύοντας την αποσύζευξη και την επεκτασιμότητα.

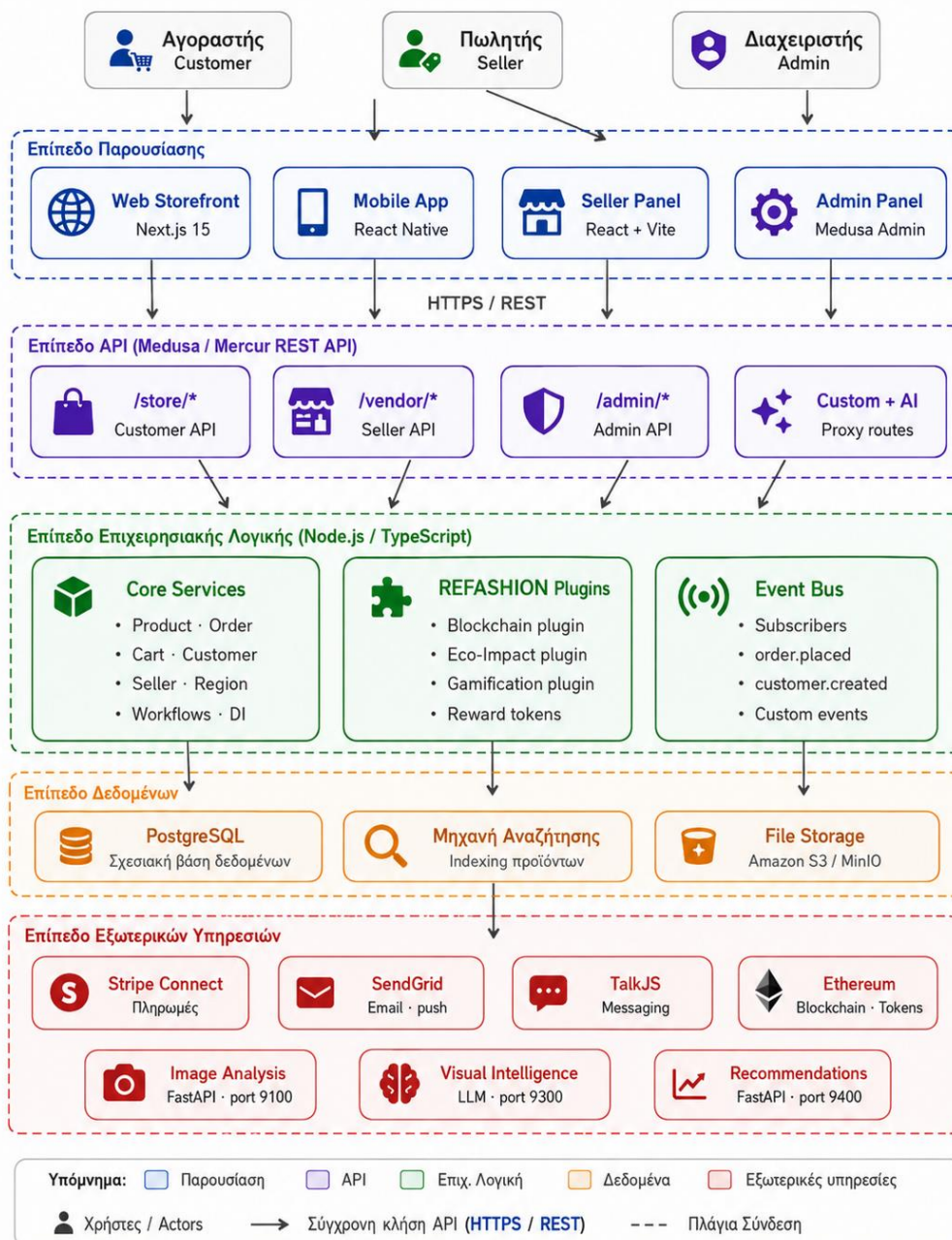
Η πολυεπίπεδη αυτή αρχιτεκτονική προσφέρει στο REFASHION μια σειρά από κρίσιμα πλεονεκτήματα. Πρώτον, η σαφής διαστρωμάτωση των ευθυνών (separation of concerns) καθιστά το σύστημα ευκολότερα κατανοητό και συντηρήσιμο, καθώς κάθε αλλαγή εστιάζεται στο επίπεδο όπου εντοπίζεται. Δεύτερον, η ανεξαρτησία των επιπέδων υποστηρίζει την παράλληλη ανάπτυξη από διαφορετικές ομάδες, καθώς οι ομάδες frontend και backend μπορούν να εργάζονται ταυτόχρονα στις δικές τους ευθύνες, αρκεί να τηρείται η συμβατότητα του API contract. Τρίτον, η μη απευθείας σύζευξη των frontend με τη βάση δεδομένων ενισχύει την ασφάλεια του συστήματος, αφού όλες οι προσβάσεις στα δεδομένα διέρχονται υποχρεωτικά μέσα από τα ελεγχόμενα σημεία του API, όπου εφαρμόζονται οι κατάλληλοι μηχανισμοί αυθεντικοποίησης και εξουσιοδότησης. Τέλος, η αυτονομία των επιπέδων επιτρέπει την ανεξάρτητη κλιμάκωση καθενός ανάλογα με τις πραγματικές απαιτήσεις φόρτου, μειώνοντας το συνολικό κόστος της υποδομής και βελτιστοποιώντας τη χρήση των πόρων.

Η αναλυτική δομή των επιμέρους συνιστωσών της πλατφόρμας, καθώς και ο τρόπος με τον οποίο αυτές εντάσσονται στο παραπάνω πολυεπίπεδο πλαίσιο, παρουσιάζεται στις επόμενες υποπαραγράφους του παρόντος κεφαλαίου. Πριν όμως από την λεπτομερή ανάλυση των επιμέρους components, η ενότητα που ακολουθεί παρουσιάζει το συνολικό διάγραμμα αρχιτεκτονικής, το οποίο αποτυπώνει με γραφικό τρόπο τη συνύπαρξη και τις

διασυνδέσεις όλων των συνιστωσών εντός του οικοσυστήματος REFASHION.

### 3.3.3 Διάγραμμα Συνολικής Αρχιτεκτονικής

Το ακόλουθο διάγραμμα αποτυπώνει με γραφικό τρόπο τη συνολική αρχιτεκτονική της πλατφόρμας REFASHION, αναδεικνύοντας τη διαστρωμάτωση των επιπέδων που περιγράφηκε στην προηγούμενη υποπαράγραφο, καθώς και τις κύριες συνιστώσες που εντάσσονται σε καθένα από αυτά. Η οπτική αυτή αναπαράσταση λειτουργεί ως σημείο αναφοράς για το σύνολο των τεχνικών αναλύσεων που ακολουθούν στις επόμενες υποπαρραγράφους, παρέχοντας μια ενοποιημένη εικόνα του οικοσυστήματος πριν από την αναλυτική παρουσίαση κάθε επιμέρους συνιστώσας.



Εικόνα 1: Η Αρχιτεκτονική της Πλατφόρμας REFASHION

Στο ανώτερο τμήμα του διαγράμματος αποτυπώνονται οι τρεις διακριτές κατηγορίες χρηστών που αλληλεπιδρούν με την πλατφόρμα - οι αγοραστές (customers), οι πωλητές (sellers) και οι διαχειριστές (administrators) - καθεμία από τις οποίες προσεγγίζει την πλατφόρμα μέσω του αντίστοιχου frontend περιβάλλοντος. Οι αγοραστές χρησιμοποιούν είτε το Web Storefront είτε την εφαρμογή κινητού, οι πωλητές αλληλεπιδρούν τόσο με τη δημόσια όψη του storefront όσο και με το εξειδικευμένο Seller Panel, ενώ οι διαχειριστές προσεγγίζουν αποκλειστικά το Medusa Admin Panel για τις λειτουργίες εποπτείας και moderation.

Το Επίπεδο Παρουσίασης φιλοξενεί τις τέσσερις διεπαφές χρήστη που έχουν αναπτυχθεί στο πλαίσιο του έργου: το Web Storefront βασισμένο στο Next.js 15, την εφαρμογή κινητού σε React Native, το Seller Panel που αξιοποιεί React με Vite, και το Medusa Admin Panel που εξυπηρετεί τους διαχειριστές. Κάθε ένα από αυτά τα περιβάλλοντα επικοινωνεί ανεξάρτητα με το backend μέσω αιτημάτων HTTPS/REST, σύμφωνα με την headless φιλοσοφία που υιοθετήθηκε.

Ακολουθεί το Επίπεδο API, το οποίο αποτυπώνει την οργάνωση των endpoints σε τέσσερα διακριτά scopes: τα /store/\* που εξυπηρετούν τις λειτουργίες του Customer API, τα /vendor/\* για τις λειτουργίες των πωλητών, τα /admin/\* για τις διαχειριστικές λειτουργίες και την κατηγορία των custom endpoints και AI proxy routes που υλοποιούν τις καινοτόμες λειτουργίες του REFASHION και διασυνδέουν την πλατφόρμα με τις εξωτερικές υπηρεσίες Τεχνητής Νοημοσύνης. Ο σαφής διαχωρισμός των scopes εξασφαλίζει την εφαρμογή των κατάλληλων μηχανισμών αυθεντικοποίησης και εξουσιοδότησης σε κάθε επίπεδο πρόσβασης.

Στον πυρήνα του διαγράμματος βρίσκεται το Επίπεδο Επιχειρησιακής Λογικής, το οποίο οργανώνεται σε τρεις λειτουργικές ενότητες. Τα Core Services του MedusaJS και του Mercury καλύπτουν τις βασικές οντότητες του συστήματος (προϊόντα, παραγγελίες, καλάθια, πελάτες, πωλητές, περιοχές) και τα αντίστοιχα workflows. Τα REFASHION Plugins ενσωματώνουν τις καινοτόμες λειτουργίες του έργου - Blockchain, Eco-Impact και Gamification - ως αυτόνομα πακέτα κώδικα που εκτείνουν τον πυρήνα χωρίς να τον τροποποιούν. Τέλος, το Event Bus φιλοξενεί τους subscribers και τις custom ροές που αντιδρούν ασύγχρονα σε γεγονότα του συστήματος, όπως το order.placed και το customer.created, ενεργοποιώντας τις αντίστοιχες λειτουργίες ανταμοιβής και ενημέρωσης.

Το Επίπεδο Δεδομένων περιλαμβάνει τους τρεις μηχανισμούς επίμονης αποθήκευσης που χρησιμοποιεί η πλατφόρμα: τη σχεσιακή βάση PostgreSQL που αποτελεί την κύρια πηγή αλήθειας για το σύνολο των οντοτήτων, μια εξειδικευμένη μηχανή αναζήτησης που διατηρεί ευρετηριασμένες αναπαραστάσεις των προϊόντων για ταχείες λειτουργίες αναζήτησης και φιλτραρίσματος, και την υπηρεσία αποθήκευσης αρχείων (Amazon S3 στην παραγωγή ή MinIO σε self-hosted περιβάλλον) που φιλοξενεί τις εικόνες των προϊόντων και τα λοιπά πολυμέσα.

Στο κατώτερο τμήμα του διαγράμματος αποτυπώνεται το Επίπεδο Εξωτερικών Υπηρεσιών, το οποίο περιλαμβάνει τόσο τους τρίτους παρόχους που έχουν ενσωματωθεί (Stripe

Connect για πληρωμές, SendGrid για ειδοποιήσεις, TalkJS για messaging, Ethereum Blockchain για συναλλαγές και ReFashion Tokens) όσο και τα τρία αυτόνομα AI microservices που έχουν αναπτυχθεί από την ίδια ομάδα του έργου - το Image Analysis Service για την επεξεργασία εικόνων, το Visual Product Intelligence Service για την οπτική αναζήτηση και τον εμπλουτισμό καταχωρήσεων, και το Recommendation Service για τις προσωποποιημένες προτάσεις προϊόντων.

Η κατευθυνόμενη ροή των βελών στο διάγραμμα αναδεικνύει τη βασική αρχή της αρχιτεκτονικής: τα αιτήματα κινούνται από τα ανώτερα επίπεδα (παρουσίαση) προς τα κατώτερα (δεδομένα), διέρχονται υποχρεωτικά μέσα από τα ενδιάμεσα επίπεδα και ποτέ δεν παρακάμπτουν τις προβλεπόμενες διεπαφές. Παράλληλα, οι πλάγιες συνδέσεις μεταξύ του επιπέδου επιχειρησιακής λογικής και των εξωτερικών υπηρεσιών αποτυπώνουν την ικανότητα του πυρήνα να εννορηστρώνει την επικοινωνία με τρίτα συστήματα, παραμένοντας ταυτόχρονα απομονωμένος από τις λεπτομέρειες υλοποίησης καθενός από αυτά.

Η αναλυτική περιγραφή των επιμέρους συνιστωσών κάθε επιπέδου, καθώς και των ροών επικοινωνίας μεταξύ τους, παρουσιάζεται στις υποπαραγράφους που ακολουθούν.

### 3.3.4 Συνιστώσες Frontend

Το επίπεδο παρουσίασης της πλατφόρμας REFASHION συγκροτείται από τέσσερα διακριτά frontend περιβάλλοντα, καθένα από τα οποία απευθύνεται σε διαφορετική κατηγορία χρηστών και εξυπηρετεί διαφορετικό σκοπό εντός του οικοσυστήματος. Παρά τις λειτουργικές διαφορές τους, και τα τέσσερα περιβάλλοντα μοιράζονται κοινά αρχιτεκτονικά χαρακτηριστικά: λειτουργούν ως αυτόνομες εφαρμογές που επικοινωνούν αποκλειστικά μέσω του REST API του Medusa/Mercur backend, δεν διατηρούν δική τους επιχειρησιακή λογική πέραν της παρουσίασης και της αλληλεπίδρασης, και υλοποιούνται με σύγχρονες, ευρέως διαδεδομένες τεχνολογίες web. Η ενότητα που ακολουθεί παρουσιάζει τα κύρια αρχιτεκτονικά χαρακτηριστικά κάθε frontend, ενώ η αναλυτική λειτουργική τους περιγραφή παρατίθεται στο Κεφάλαιο 5.

#### 3.3.4.1 Web Storefront

Το Web Storefront αποτελεί τη βασική και δημοσίως προσβάσιμη διεπαφή της πλατφόρμας, μέσω της οποίας οι τελικοί αγοραστές αλληλεπιδρούν με το οικοσύστημα REFASHION. Από αρχιτεκτονική σκοπιά, είναι σχεδιασμένο ως μια αυτόνομη web εφαρμογή που υλοποιεί τη φιλοσοφία του customer-facing scope, χωρίς να εκτίθεται σε λειτουργίες πωλητών ή διαχειριστών. Η επιλογή αυτή ακολουθεί την αρχή της ελάχιστης απαραίτητης πληροφορίας και προστατεύει τόσο την εμπειρία χρήσης όσο και την ασφάλεια της πλατφόρμας.

Η υλοποίηση βασίζεται στις τεχνολογίες Next.js 15, React 19 και TypeScript, αξιοποιώντας την αρχιτεκτονική του App Router για τη δρομολόγηση των αιτημάτων και τη διαχείριση του rendering. Η επιλογή του Next.js δεν είναι συμπτωματική: το framework προσφέρει server-side rendering (SSR) και static site generation (SSG), χαρακτηριστικά κρίσιμα για την απόδοση μιας πλατφόρμας e-commerce, καθώς εξασφαλίζουν γρήγορους χρόνους

φόρτωσης για τους χρήστες και βελτιστοποιημένη ορατότητα στις μηχανές αναζήτησης (SEO). Παράλληλα, η χρήση του Tailwind CSS ως utility-first styling framework διασφαλίζει συνεπή σχεδιαστική γλώσσα σε ολόκληρη την εφαρμογή, ενώ ταυτόχρονα διατηρεί χαμηλό το μέγεθος του τελικού CSS bundle.

Στο επίπεδο της αρχιτεκτονικής συνιστωσών (component architecture), το storefront ακολουθεί τη μεθοδολογία των επαναχρησιμοποιήσιμων components σε ιεραρχημένα επίπεδα (atoms, molecules, organisms), διασφαλίζοντας ότι η πλατφόρμα παραμένει modular και επεκτάσιμη. Η οργανωτική αυτή προσέγγιση επιτρέπει στην ομάδα ανάπτυξης να διατηρεί συνέπεια στη σχεδιαστική γλώσσα ενώ ταυτόχρονα επιταχύνει την προσθήκη νέων λειτουργιών, καθώς νέες οθόνες συντίθενται από υπάρχοντα δομικά στοιχεία.

Η επικοινωνία με το backend πραγματοποιείται αποκλειστικά μέσω του /store/\* scope του Medusa/Mercur REST API, είτε μέσω του επίσημου JavaScript SDK είτε μέσω απευθείας fetch κλήσεων. Καμία επιχειρησιακή λογική δεν εκτελείται τοπικά στο storefront - όλες οι αποφάσεις σχετικά με τιμολόγηση, διαθεσιμότητα, εγκυρότητα παραγγελιών και επεξεργασία πληρωμών λαμβάνονται από το backend. Το storefront περιορίζεται στον ρόλο του ως παρουσιαστή δεδομένων και διαμεσολαβητή αλληλεπίδρασης, διατηρώντας έτσι τη σαφή απομόνωση των ευθυνών που υπαγορεύει η headless αρχιτεκτονική.

Από αρχιτεκτονικής σκοπιάς, το storefront ενσωματώνει τη λογική region-aware παρουσίασης, μέσω εξειδικευμένου middleware που αναγνωρίζει την περιοχή του χρήστη και προσαρμόζει δυναμικά το νόμισμα, τις τιμές και τις διαθέσιμες επιλογές αποστολής. Παράλληλα, διασυνδέεται με τα AI proxy routes του backend για την ενεργοποίηση των υπηρεσιών Τεχνητής Νοημοσύνης (επεξεργασία εικόνων, οπτική αναζήτηση, προσωποποιημένες προτάσεις), χωρίς να εκθέτει απευθείας τα εσωτερικά microservices στο διαδίκτυο. Η προσέγγιση αυτή εξυπηρετεί τόσο την ασφάλεια όσο και την ενοποιημένη παρουσίαση ενός ενιαίου API surface προς τον τελικό χρήστη.

Ένα διακριτό αρχιτεκτονικό χαρακτηριστικό του storefront είναι η ενσωμάτωση του Eco-Impract και Gamification module στη διεπαφή χρήστη. Παρότι η λογική υπολογισμού και η αποθήκευση των σχετικών δεδομένων εκτελείται από τα αντίστοιχα plugins του backend, το storefront αναλαμβάνει την οπτική παρουσίαση των ReFashion Tokens, των περιβαλλοντικών δεικτών και των achievements των χρηστών, καθιστώντας τα καινοτόμα στοιχεία του REFASHION άμεσα ορατά και αντιληπτά. Με αυτόν τον τρόπο, το storefront λειτουργεί ως ο κύριος αγωγός επικοινωνίας της προστιθέμενης αξίας της πλατφόρμας προς τους τελικούς χρήστες.

Τέλος, ο responsive σχεδιασμός που υιοθετείται εξασφαλίζει ότι η εμπειρία χρήσης παραμένει συνεπής και βελτιστοποιημένη ανεξάρτητα από τη συσκευή πρόσβασης - desktop, tablet ή κινητό - ενισχύοντας την προσβασιμότητα στη βιώσιμη αγορά και διευρύνοντας το πιθανό κοινό της πλατφόρμας. Η αναλυτική λειτουργική περιγραφή των ροών εγγραφής, αναζήτησης, αγοράς και προβολής στατιστικών παρουσιάζεται στην ενότητα 5.1 του παρόντος παραδοτέου.

### 3.3.4.2 Εφαρμογή Κινητού

Η Εφαρμογή Κινητού του REFASHION αποτελεί τη native mobile διέξοδο του οικοσυστήματος, παρέχοντας στους χρήστες - αγοραστές και πωλητές - μια βελτιστοποιημένη εμπειρία πρόσβασης μέσω συσκευών iOS και Android. Από αρχιτεκτονική σκοπιά, η εφαρμογή κατατάσσεται στο ίδιο επίπεδο παρουσίασης με το Web Storefront, διατηρώντας ωστόσο την απόλυτη αυτονομία της ως ξεχωριστό frontend περιβάλλον με δικό του κύκλο ανάπτυξης και δικά του χαρακτηριστικά παρουσίασης.

Η ύπαρξη αυτή αυτόνομης mobile εφαρμογής, παράλληλα με το responsive web storefront, αποτελεί συνειδητή στρατηγική επιλογή που εξυπηρετεί συγκεκριμένους σκοπούς. Παρότι το web storefront προσφέρει responsive σχεδιασμό συμβατό με κάθε συσκευή, η native εφαρμογή κινητού επιτρέπει την αξιοποίηση πληθώρας λειτουργιών της συσκευής που δεν είναι διαθέσιμες - ή είναι σημαντικά περιορισμένες - μέσω browser. Στις λειτουργίες αυτές περιλαμβάνονται η απευθείας πρόσβαση στην κάμερα για ταχεία φωτογράφιση προϊόντων, η αξιοποίηση του geolocation για αναζήτηση γεωγραφικά κοντινών προσφορών, η αποστολή push notifications για ενημέρωση σχετικά με νέες παραγγελίες, μηνύματα ή ανταμοιβές, καθώς και η offline-first λειτουργία για την προβολή ήδη επισκεπτόμενων προϊόντων χωρίς ενεργή σύνδεση στο διαδίκτυο.

Η ανάπτυξη της εφαρμογής υλοποιείται από την τεχνική ομάδα της Cyclefi. Η συγκεκριμένη οργανωτική επιλογή - όπου το backend συντηρείται κεντρικά από την ομάδα του Πανεπιστημίου Πελοποννήσου ενώ το mobile frontend αναπτύσσεται από άλλο εταίρο του έργου - αποτελεί επιβεβαίωση της σχεδιαστικής επιλογής για τη headless αρχιτεκτονική που υιοθετήθηκε. Χάρη στον αυστηρό διαχωρισμό backend και frontend μέσω του REST API, οι δύο ομάδες μπορούν να εργάζονται παράλληλα και ανεξάρτητα, αρκεί να τηρείται η συμβατότητα του API contract, χωρίς να απαιτείται συντονισμένη συν-ανάπτυξη ή διαμοιραζόμενη πρόσβαση σε κώδικα. Η συνεργασία διαμεσολαβείται σε επίπεδο τεκμηρίωσης και σχεδιαστικών προδιαγραφών, με τη χρήση του Figma για το UI/UX και του OpenAPI/Swagger για τις διεπαφές του backend.

Από τεχνολογική σκοπιά, η εφαρμογή υλοποιείται σε React Native, ένα ευρέως διαδεδομένο cross-platform framework που επιτρέπει την ταυτόχρονη υποστήριξη iOS και Android μέσω ενιαίας κωδικοβάσης. Η επιλογή αυτή προσφέρει σημαντικά πλεονεκτήματα: επιταχύνει σημαντικά τον χρόνο ανάπτυξης σε σύγκριση με την παράλληλη συντήρηση δύο native εφαρμογών (Swift για iOS και Kotlin για Android), εξασφαλίζει συνέπεια στη συμπεριφορά και την εμφάνιση μεταξύ των δύο λειτουργικών συστημάτων, και επιτρέπει την αξιοποίηση κοινής εξειδίκευσης σε JavaScript/TypeScript με την ομάδα ανάπτυξης του web storefront. Παράλληλα, η αρχιτεκτονική του React Native παρέχει τη δυνατότητα ενσωμάτωσης native modules για λειτουργίες που απαιτούν άμεση πρόσβαση σε API του λειτουργικού συστήματος, εξασφαλίζοντας ότι η τελική εμπειρία χρήστη παραμένει αντίστοιχη με αυτή των native εφαρμογών.

Η επικοινωνία της εφαρμογής με το backend ακολουθεί την ίδια αρχή με όλα τα frontend περιβάλλοντα του REFASHION: πραγματοποιείται αποκλειστικά μέσω αιτημάτων HTTPS προς το REST API του Medusa/Mercur. Η εφαρμογή αξιοποιεί τόσο το /store/\* scope για

τις λειτουργίες των αγοραστών όσο και το `/vendor/*` scope για τις λειτουργίες των πωλητών, καθώς υποστηρίζει και τις δύο κατηγορίες χρηστών μέσω ενιαίας εφαρμογής με εναλλαγή ρόλου. Η αυθεντικοποίηση εκτελείται μέσω JWT tokens που αποθηκεύονται με ασφάλεια στο συσκευής storage (Keychain σε iOS, EncryptedSharedPreferences σε Android), εξασφαλίζοντας ότι τα διαπιστευτήρια χρήστη παραμένουν προστατευμένα ακόμη και σε περίπτωση κακόβουλης πρόσβασης στη συσκευή.

Σημαντικό αρχιτεκτονικό χαρακτηριστικό της εφαρμογής είναι η διασύνδεσή της με τα AI *microservices* της πλατφόρμας. Κατά τη ροή καταχώρησης ενός προϊόντος από πωλητή, η εφαρμογή αξιοποιεί την κάμερα της συσκευής για την άμεση λήψη φωτογραφιών, οι οποίες αποστέλλονται μέσω του backend στα proxy routes που διασυνδέονται με τα AI *microservices*. Ο μηχανισμός αυτός επιτρέπει την αυτόματη επεξεργασία της εικόνας (αφαίρεση φόντου), την αναγνώριση χαρακτηριστικών του προϊόντος μέσω *visual product intelligence*, και την αυτόματη συμπλήρωση των πεδίων της καταχώρησης. Με τον τρόπο αυτό, η εφαρμογή μετασχηματίζει μια διαδικασία που παραδοσιακά απαιτούσε λεπτά πληκτρολόγησης και επιλογών σε μια ροή λίγων δευτερολέπτων, αξιοποιώντας την κάμερα ως κύριο μηχανισμό εισόδου αντί του πληκτρολογίου.

Παράλληλα, η εφαρμογή υποστηρίζει την προβολή του Stats Module, μέσω του οποίου ο χρήστης παρακολουθεί τα συσσωρευμένα ReFashion Tokens, τους περιβαλλοντικούς δείκτες της δραστηριότητάς του (εξοικονόμηση CO<sub>2</sub> και νερού), καθώς και τα ανακτηθέντα badges. Σημειώνεται ότι, σύμφωνα με την τρέχουσα κατάσταση υλοποίησης, η ροή των Tokens και των Impact δεδομένων προς την εφαρμογή είναι υπό ανάπτυξη, ενώ η ενότητα Achievements βρίσκεται σε εξέλιξη - η αναλυτική καταγραφή της προόδου παρουσιάζεται στο Κεφάλαιο 8.

Η ενσωμάτωση των push notifications πραγματοποιείται μέσω της σύνδεσης με την υπηρεσία SendGrid του backend, ενώ η εφαρμογή λειτουργεί ως ο κύριος αγωγός παράδοσης εξειδικευμένων ειδοποιήσεων προς τους χρήστες - νέα μηνύματα από πωλητές, ενημερώσεις κατάστασης παραγγελιών, ανακοινώσεις απόδοσης ReFashion Tokens και επετειακές υπενθυμίσεις. Η real-time αυτή επικοινωνία αποτελεί κρίσιμο διαφοροποιητικό στοιχείο σε σύγκριση με το web περιβάλλον, ενισχύοντας την εμπλοκή των χρηστών και τη συχνότητα επαναφοράς τους στην πλατφόρμα.

Η αναλυτική λειτουργική περιγραφή της εφαρμογής, καθώς και των κύριων ροών αλληλεπίδρασης που υλοποιεί, παρατίθεται στην ενότητα 5.2 του παρόντος παραδοτέου.

### 3.3.4.3 Seller Admin Panel

Το Seller Admin Panel αποτελεί το εξειδικευμένο περιβάλλον διαχείρισης των πωλητών εντός του οικοσυστήματος REFASHION. Πρόκειται για μια διακριτή frontend εφαρμογή, πλήρως αποσυνδεδεμένη τόσο από το storefront των αγοραστών όσο και από το admin panel των διαχειριστών, η οποία απευθύνεται αποκλειστικά σε εγκεκριμένους πωλητές για την αυτόνομη λειτουργία των ψηφιακών τους καταστημάτων εντός του marketplace. Από αρχιτεκτονική σκοπιά, το panel αποτελεί την υλοποίηση της αρχής του διαχωρισμού των σκοπών που διέπει το σύνολο της πλατφόρμας: κάθε κατηγορία χρηστών εξυπηρετείται από ένα εξειδικευμένο περιβάλλον, σχεδιασμένο γύρω από τις δικές της ανάγκες και αρμοδιότητες, χωρίς διαρροή λειτουργικότητας μεταξύ των ρόλων.

Η ανάπτυξη του Seller Panel πραγματοποιείται με τη χρήση του συνδυασμού React, Vite και TypeScript, εμπλουτισμένου με τα components της βιβλιοθήκης Medusa UI. Η επιλογή του Vite ως build tool - αντί άλλων εναλλακτικών - εξυπηρετεί συγκεκριμένες απαιτήσεις του περιβάλλοντος: προσφέρει εξαιρετικά γρήγορους χρόνους εκκίνησης κατά την ανάπτυξη και ταχύτερη ανανέωση των αλλαγών στο browser, χαρακτηριστικά κρίσιμα για μια εφαρμογή με σύνθετα διαχειριστικά interfaces και πολλαπλά πεδία φόρμας. Παράλληλα, η χρήση της Medusa UI εξασφαλίζει σχεδιαστική συνέπεια με τα υπόλοιπα διαχειριστικά περιβάλλοντα του οικοσυστήματος, ενώ μειώνει σημαντικά τον χρόνο ανάπτυξης μέσω της επαναχρησιμοποίησης έτοιμων components για κοινές διαχειριστικές λειτουργίες (πίνακες δεδομένων, φόρμες, drawers, modals).

Το πιο κρίσιμο αρχιτεκτονικό χαρακτηριστικό του Seller Panel είναι η αυστηρή εφαρμογή του seller-scored δεδομένων (data scoring). Σε αντίθεση με ένα κεντρικό admin panel που έχει καθολική πρόσβαση, το Seller Panel επιβάλλει τον περιορισμό ότι κάθε πωλητής βλέπει και διαχειρίζεται αποκλειστικά τα δικά του δεδομένα - τα προϊόντα του, τις παραγγελίες που τον αφορούν, το απόθεμά του, τις επικοινωνίες του και τα οικονομικά του στοιχεία. Η επιβολή αυτής της απομόνωσης δεν αποτελεί απλώς ζήτημα παρουσίασης στο frontend, αλλά υλοποιείται σε βάθος επιπέδων μέσω των /vendor/\* endpoints του backend, τα οποία ενσωματώνουν εγγενώς ownership checks που εξασφαλίζουν ότι κάθε request φιλτράρεται αυτόματα στα όρια του συγκεκριμένου πωλητή. Με τον τρόπο αυτό, η αρχιτεκτονική εγγυάται ότι ακόμη και σε περίπτωση παράκαμψης των UI κανόνων μέσω άμεσων κλήσεων API, η ισολογιστική του συστήματος παραμένει αδιαπέραστη.

Η πρόσβαση στο Seller Panel ακολουθεί μια εξειδικευμένη ροή που αξιοποιεί τη μηχανική του token handoff μεταξύ storefront και vendor περιβάλλοντος. Η διαδικασία εκκινεί από το customer storefront, όπου ο χρήστης - εφόσον διαθέτει ενεργό και εγκεκριμένο seller account - μπορεί να ενεργοποιήσει τη μετάβαση στο vendor περιβάλλον μέσω του κουμπιού "Sell now". Στο παρασκήνιο, η ενέργεια αυτή πυροδοτεί τη ροή /auth/acceprt?token=..., η οποία μεταφέρει το seller token από το ένα περιβάλλον στο άλλο χωρίς να απαιτείται από τον χρήστη να εισάγει εκ νέου τα διαπιστευτήριά του. Η ροή αυτή προσφέρει απρόσκοπτη εμπειρία χρήστη, καθώς οι χρήστες με διπλή ιδιότητα (αγοραστή και πωλητή) δεν χρειάζεται να διαχειρίζονται δύο ξεχωριστούς λογαριασμούς ή δύο διαδικασίες σύνδεσης. Η αυθεντικοποίηση επικυρώνεται μέσω του endpoint GET /vendor/sellers/me, το οποίο εκτελεί ταυτότητα και επιβεβαιώνει την κατάσταση του λογαριασμού (active vs suspended).

Η οργάνωση του Seller Panel σε λειτουργικές ενότητες ακολουθεί τη φυσική ροή εργασίας ενός πωλητή στο marketplace. Στο κέντρο της εμπειρίας βρίσκεται το seller dashboard, που λειτουργεί ως πανοραμική επισκόπηση της εμπορικής δραστηριότητας, ενσωματώνοντας onboarding καθοδήγηση για νέους πωλητές και απευθείας πρόσβαση σε κρίσιμες ενέργειες. Γύρω από το dashboard, οι υπόλοιπες ενότητες καλύπτουν τη διαχείριση προϊόντων (καταχώρηση νέων ειδών, διαχείριση παραλλαγών, οργάνωση καταλόγου), τη διαχείριση αποθέματος (stock levels, reservations), τη διαχείριση αποστολών (stock locations, fulfillment providers, shipping profiles), τη διαχείριση παραγγελιών (συμπεριλαμβανομένων των returns, claims, exchanges και refunds), τα εργαλεία marketing και τιμολόγησης (campaigns, price lists), και την ενότητα επικοινωνίας μέσω TalkJS για άμεση αλληλεπίδραση με τους αγοραστές.

Η οικονομική διάσταση του πωλητή ενσωματώνεται μέσω της σύνδεσης με το Stripe Connect, η οποία αποτελεί κρίσιμη συνιστώσα για τη βιωσιμότητα του marketplace μοντέλου. Το panel εκθέτει τη ροή σύνδεσης λογαριασμού Stripe Connect του πωλητή, παρακολουθεί την κατάσταση της σύνδεσης σε πραγματικό χρόνο και επιτρέπει τη

διαχείριση των *raouts* προς τον τραπεζικό του λογαριασμό. Η αρχιτεκτονική αυτή προσέγγιση εξασφαλίζει ότι κάθε πωλητής διατηρεί αυτόνομη οικονομική σχέση με τον πάροχο πληρωμών, χωρίς η πλατφόρμα να χρειάζεται να διαχειρίζεται κεντρικά τα κεφάλαια όλων των πωλητών - μειώνοντας ταυτόχρονα τις κανονιστικές υποχρεώσεις (PSD2, ηλεκτρονικό χρήμα) και τους κινδύνους ρευστότητας.

Σημαντική αρχιτεκτονική επιλογή είναι ότι το Seller Panel δεν παρακάμπτει τη διαδικασία *moderation* για τις καταχωρήσεις προϊόντων. Παρότι ο πωλητής έχει πλήρη αυτονομία στη δημιουργία νέων καταχωρήσεων, αυτές υποβάλλονται είτε σε κατάσταση *draft* είτε ως *proposed* προς έγκριση, χωρίς να εμφανίζονται άμεσα στο δημόσιο *storefront*. Η οριστική δημοσίευσή τους εξαρτάται από τη θετική αξιολόγηση των διαχειριστών της πλατφόρμας μέσω του Medusa Admin Panel. Ο μηχανισμός αυτός εξασφαλίζει την ποιοτική συνέπεια του καταλόγου και αποτρέπει την εμφάνιση ακατάλληλου ή χαμηλής ποιότητας υλικού, στοιχεία κρίσιμα για την αξιοπιστία ενός *marketplace* επανεμπορίου.

Η αναλυτική περιγραφή των λειτουργικών ενότητων και των ροών εργασίας που υποστηρίζει το Seller Admin Panel παρατίθεται στην ενότητα 5.3.1 του παρόντος παραδοτέου.

#### 3.3.4.4 Medusa Admin Panel

Το Medusa Admin Panel αποτελεί το κεντρικό σημείο ελέγχου και διακυβέρνησης της πλατφόρμας REFASHION, μέσω του οποίου οι διαχειριστές εποπτεύουν, παραμετροποιούν και ελέγχουν το σύνολο των λειτουργιών του *marketplace*. Σε αντίθεση με το Seller Panel που εξυπηρετεί την αυτόνομη λειτουργία των επιμέρους πωλητών, το Medusa Admin Panel απευθύνεται στους κεντρικούς υπευθύνους της πλατφόρμας, προσφέροντάς τους καθολική εποπτεία σε όλους τους πωλητές, τα προϊόντα, τις παραγγελίες και τις διαδικασίες που λαμβάνουν χώρα εντός του οικοσυστήματος.

Από αρχιτεκτονική σκοπιά, το panel εδράζεται εντός του ίδιου του *backend project* του Medusa/Mercur, αξιοποιώντας τον επίσημο πυρήνα του Medusa Admin εμπλουτισμένο με τις *custom marketplace* επεκτάσεις που εισάγει το Mercur. Η συγκεκριμένη οργανωτική επιλογή - όπου το *admin panel* δεν αποτελεί ξεχωριστό *repository* αλλά συνυπάρχει με το *backend* - αποτελεί συνηθισμένο *pattern* σε διαχειριστικά περιβάλλοντα, καθώς εξασφαλίζει απόλυτο συγχρονισμό μεταξύ της εξέλιξης του *backend* και των διαθέσιμων διαχειριστικών εργαλείων. Κάθε νέα οντότητα, νέο πεδίο ή νέα ροή που προστίθεται στο *backend* μπορεί να εκτεθεί άμεσα στο *admin panel* χωρίς τη διαμεσολάβηση ξεχωριστής διαδικασίας *release*.

Η υλοποίηση του panel βασίζεται στο React ως *frontend framework*, ακολουθώντας τα σχεδιαστικά πρότυπα και τα *components* του Medusa Admin οικοσυστήματος. Η επιλογή αυτή προσφέρει σημαντικά πλεονεκτήματα: εξασφαλίζει συνέπεια στην εμπειρία διαχείρισης μέσω καθιερωμένων διαχειριστικών *patterns* που οι χρήστες ήδη γνωρίζουν από άλλες πλατφόρμες *e-commerce*, ενώ ταυτόχρονα διευκολύνει την προσθήκη νέων ενότητων μέσω της *extensible* αρχιτεκτονικής του Medusa Admin που επιτρέπει την ενσωμάτωση *custom pages*, *custom widgets* και *custom routes* χωρίς παρέμβαση στον πυρήνα.

Η επικοινωνία του panel με το *backend* πραγματοποιείται αποκλειστικά μέσω των */admin/endpoints\** του REST API, τα οποία αποτελούν το πιο προστατευμένο *scope* του συστήματος. Σε αντίθεση με τα */store/\** και */vendor/\** endpoints που επιτρέπουν περιορισμένη πρόσβαση σε *customer* και *seller tokens* αντίστοιχα, τα */admin/\** endpoints απαιτούν αυξημένα δικαιώματα αυθεντικοποίησης και είναι σχεδιασμένα για να φιλτράρουν αυστηρά τις εισερχόμενες κλήσεις, εξασφαλίζοντας ότι οι ευαίσθητες

διαχειριστικές λειτουργίες παραμένουν προσβάσιμες μόνο σε εξουσιοδοτημένο προσωπικό.

Η λειτουργική οργάνωση του Medusa Admin Panel στο REFASHION εκτείνεται πέρα από τις τυπικές δυνατότητες ενός e-commerce backoffice, ενσωματώνοντας τις εξειδικευμένες ανάγκες ενός marketplace επανεμπορίου. Κομβική θέση κατέχει η διαχείριση πωλητών, η οποία επιτρέπει στους διαχειριστές να αναζητούν και να αναλύουν στοιχεία των sellers, να παρακολουθούν τη δραστηριότητά τους και να εκτελούν ενέργειες suspend ή activate όταν κρίνεται αναγκαίο για την προστασία της ποιότητας του marketplace. Παράλληλα, η ενότητα Seller creation requests συντονίζει τη ροή έγκρισης νέων συμμετεχόντων, παρέχοντας στον admin τη δυνατότητα εξέτασης και αξιολόγησης των εκκρεμών αιτήσεων μέσω των endpoints GET /admin/requests?type=seller&status=pending και POST /admin/requests/{id}. Η οριστική ενεργοποίηση των seller accounts αποτελεί αποκλειστικό προνόμιο του admin, εξασφαλίζοντας την κεντρική επίβλεψη της εισόδου νέων μελών στην πλατφόρμα.

Αντίστοιχα κεντρική θέση στο panel κατέχει το product moderation. Κάθε νέα υποβολή προϊόντος από πωλητή δημιουργεί ένα product request, το οποίο εμφανίζεται στους διαχειριστές για επικύρωση πριν τη δημοσίευσή του στο storefront. Ο μηχανισμός αυτός αποτελεί κρίσιμη δικλίδα ασφαλείας για την ποιοτική συνέπεια του καταλόγου, καθώς αποτρέπει την εμφάνιση ακατάλληλου, παραπλανητικού ή χαμηλής ποιότητας υλικού στο δημόσιο μέρος της πλατφόρμας. Η λογική του ελέγχου επεκτείνεται και σε άλλες σχετικές οντότητες - κατηγορίες, συλλογές (collections), tags, καθώς και Product Attributes (brand, υλικό, χρώμα) - επιτρέποντας την κεντρική επιμέλεια των δομικών στοιχείων του καταλόγου. Ξεχωριστή σημασία για το επιχειρηματικό μοντέλο της πλατφόρμας έχει η ενότητα διαχείρισης των marketplace commissions. Μέσω των commission rules, οι διαχειριστές ορίζουν τα ποσοστά προμήθειας που εφαρμόζονται ανά πωλητή ή ανά κατηγορία προϊόντος, υποστηρίζοντας τη βιωσιμότητα και την κερδοφορία του οικοσυστήματος. Η ευελιξία του μηχανισμού επιτρέπει την εφαρμογή διαφοροποιημένων πολιτικών - για παράδειγμα, μειωμένα ποσοστά για ιδιώτες πωλητές έναντι εταιρικών χρηστών, ή προωθητικές περιόδους χωρίς προμήθεια για την προσέλκυση νέων πωλητών - χωρίς την ανάγκη παρέμβασης στον κώδικα.

Πέρα από τις διαχειριστικές λειτουργίες, το Medusa Admin Panel ενσωματώνει και ενότητες που αναδεικνύουν την προστιθέμενη αξία του REFASHION ως πλατφόρμας βιώσιμης κατανάλωσης. Το Impact Overview παρέχει συγκεντρωτικά δεδομένα για τη συνεισφορά της πλατφόρμας στην κυκλική οικονομία, όπως η συνολική εξοικονόμηση CO<sub>2</sub> και νερού, και τη συνολική αξία επανεμπορίου που έχει διακινηθεί. Αυτές οι μετρήσεις αποτελούν κρίσιμους δείκτες όχι μόνο για την εσωτερική αξιολόγηση της επιτυχίας του έργου, αλλά και για την επικοινωνία της περιβαλλοντικής επίδρασης σε εξωτερικούς φορείς, χρηματοδότες και την ίδια την κοινότητα των χρηστών.

Η ενότητα Product catalog settings επιτρέπει στους διαχειριστές να ορίζουν καθολικούς κανόνες λειτουργίας που επηρεάζουν τη συμπεριφορά ολόκληρου του συστήματος. Παραδείγματα τέτοιων ρυθμίσεων περιλαμβάνουν την υποχρεωτική έγκριση νέων προϊόντων πριν τη δημοσίευση, τον περιορισμό συγκεκριμένων κατηγοριών ή εμπορικών πρακτικών, και τους κανόνες ορατότητας προϊόντων ανά region. Με τον τρόπο αυτό, το admin panel λειτουργεί ως ο μηχανισμός διακυβέρνησης της πλατφόρμας, παρέχοντας στους διαχειριστές τη δυνατότητα να προσαρμόζουν τη συμπεριφορά του συστήματος στις εκάστοτε επιχειρησιακές, κανονιστικές ή στρατηγικές απαιτήσεις.

Παράλληλα με τις marketplace-specific επεκτάσεις, το Medusa Admin Panel διατηρεί το πλήρες σύνολο των τυπικών δυνατοτήτων ενός e-commerce backoffice: διαχείριση

πελατών (customers), περιοχών και νομισμάτων (regions, currencies), φορολογικών ρυθμίσεων (tax regions, tax rates), παρόχων πληρωμής και αποστολής (payment providers, fulfillment providers), επιστροφών και ανταλλαγών, καθώς και των γενικών παραμέτρων της πλατφόρμας. Αυτές οι τυπικές λειτουργίες, σε συνδυασμό με τις custom marketplace επεκτάσεις του REFASHION, καθιστούν το panel ολοκληρωμένο εργαλείο διακυβέρνησης που καλύπτει την πλήρη γκάμα των διαχειριστικών αναγκών.

Συνολικά, το Medusa Admin Panel λειτουργεί ως ο κεντρικός μηχανισμός εποπτείας και ελέγχου που εγγυάται τη λειτουργική αρτιότητα, την ασφάλεια και την ποιοτική συνέπεια ολόκληρου του οικοσυστήματος REFASHION. Η αναλυτική περιγραφή των επιμέρους ενότητων του και των ροών εργασίας που υποστηρίζει παρατίθεται στην ενότητα 5.3.2 του παρόντος παραδοτέου.

### 3.3.5 Συνιστώσες Backend & Βάση Δεδομένων

Πέρα από τα frontend περιβάλλοντα που παρουσιάστηκαν προηγουμένως, ο πραγματικός πυρήνας της πλατφόρμας REFASHION εδράζεται στο backend και στα σχετιζόμενα persistence υποσυστήματα (PostgreSQL, μηχανή αναζήτησης, file storage). Σε αυτά τα επίπεδα συγκεντρώνεται το σύνολο της επιχειρησιακής λογικής, διατηρείται η μοναδική πηγή αλήθειας για όλες τις οντότητες του συστήματος και εκτελούνται οι αλγόριθμοι που υλοποιούν τόσο τις βασικές λειτουργίες ηλεκτρονικού εμπορίου όσο και τις καινοτόμες επεκτάσεις του έργου. Η ενότητα που ακολουθεί παρουσιάζει τις κύριες αρχιτεκτονικές συνιστώσες του backend, την οργάνωση της επιχειρησιακής λογικής, καθώς και τους μηχανισμούς αποθήκευσης δεδομένων που υποστηρίζουν τη λειτουργία της πλατφόρμας.

#### 3.3.5.1 Medusa/Mercur Backend

Το **Medusa/Mercur Backend** αποτελεί τον **κεντρικό εμπορικό πυρήνα** της πλατφόρμας REFASHION και τη μοναδική συνιστώσα που έχει πρόσβαση στα δεδομένα του συστήματος. Όλα τα αιτήματα από όλα τα frontend περιβάλλοντα διοχετεύονται μέσω αυτού, καθιστώντας το **το πιο κρίσιμο σημείο** της αρχιτεκτονικής τόσο σε επίπεδο λειτουργικής αξιοπιστίας όσο και σε επίπεδο ασφάλειας. Η σταθερή λειτουργία και η σωστή σχεδίαση του backend είναι **καθοριστικής σημασίας** για τη βιωσιμότητα του συνόλου της πλατφόρμας.

Το backend υλοποιείται σε **Node.js** με χρήση της γλώσσας **TypeScript**, αποτελώντας το codebase που συνδυάζει τον πυρήνα του MedusaJS με τις επεκτάσεις του Mercur που υποστηρίζουν τη marketplace λειτουργικότητα. Η επιλογή του Node.js ως runtime προσφέρει πολλαπλά πλεονεκτήματα για την αρχιτεκτονική του REFASHION: η ασύγχρονη φύση του (event-driven, non-blocking I/O) είναι ιδανική για **API-heavy εφαρμογές** που εξυπηρετούν μεγάλο όγκο συγχρόνων αιτημάτων, ενώ το πλούσιο οικοσύστημα πακέτων (npm) διευκολύνει την ενσωμάτωση με τρίτα συστήματα. Η χρήση της TypeScript, αντί απλής JavaScript, εξασφαλίζει την **τυποποίηση των δεδομένων (static typing)** σε όλη την κωδικοβάση, μειώνοντας δραματικά τις πιθανότητες σφαλμάτων κατά τη φάση ανάπτυξης και αυξάνοντας τη συντηρησιμότητα του κώδικα σε βάθος χρόνου.

Από αρχιτεκτονική σκοπιά, το backend οργανώνεται γύρω από μια σαφή **διαστρωμάτωση εσωτερικών συνιστωσών**. Στο εξώτερο επίπεδο, η **API επιφάνεια** εκτίθεται μέσω του Express-based HTTP server που συνοδεύει το MedusaJS, εκθέτοντας τα διακριτά scopes /store/, /vendor/ και /admin/\* όπως αυτά παρουσιάστηκαν στις προηγούμενες ενότητες. Κάθε endpoint υλοποιείται μέσω εξειδικευμένων **route handlers** που εκτελούν την αρχική επικύρωση των αιτημάτων, εφαρμόζουν τους κανόνες αυθεντικοποίησης και

εξουσιοδότησης, και στη συνέχεια προωθούν την επεξεργασία στα αντίστοιχα services του εσωτερικού επιπέδου.

Στο **εσωτερικό επίπεδο** βρίσκεται η **επιχειρησιακή λογική** (business logic), η οποία οργανώνεται γύρω από τους τρεις πυλώνες που εισήγαγε το MedusaJS - **Plugins, Services και Modules** - και που αναλύθηκαν στην προηγούμενη υποπαράγραφο 2.1.3. Τα Services υλοποιούν τις core λειτουργίες κάθε οντότητας (ProductService, OrderService, CartService, CustomerService, SellerService), αλληλεπιδρούν μεταξύ τους μέσω **Dependency Injection**, και ευθύνονται για την εφαρμογή των επιχειρησιακών κανόνων που διέπουν τη συμπεριφορά του συστήματος. Τα Plugins, με τη σειρά τους, ενσωματώνουν τις εξειδικευμένες επεκτάσεις του REFASHION - τα refashion-blockchain-plugin, refashion-eco-impact-plugin και refashion-gamification-plugin - χωρίς να επηρεάζουν τον πυρήνα.

Σημαντική αρχιτεκτονική συνιστώσα είναι το **Event Bus**, ο εσωτερικός μηχανισμός που επιτρέπει την **ασύγχρονη επικοινωνία** μεταξύ διαφορετικών τμημάτων του backend. Όταν συμβαίνει ένα γεγονός σημαντικό για το σύστημα - όπως η ολοκλήρωση μιας παραγγελίας (`order.placed`) ή η δημιουργία νέου χρήστη (`customer.created`) - το event δημοσιεύεται στο bus και οι ενδιαφερόμενοι **Subscribers** το λαμβάνουν και εκτελούν τις αντίστοιχες ενέργειες, χωρίς να απαιτείται απευθείας coupling μεταξύ των συστατικών. Ο μηχανισμός αυτός αποτελεί τη **σπονδυλική στήλη** για την ενσωμάτωση των καινοτόμων λειτουργιών του REFASHION: η ολοκλήρωση μιας παραγγελίας πυροδοτεί ταυτόχρονα την καταγραφή της συναλλαγής στο blockchain, τον υπολογισμό του Eco-Impact, την απόδοση των ReWear Tokens και την ενεργοποίηση των gamification κανόνων - χωρίς ο κώδικας της παραγγελίας να γνωρίζει την ύπαρξη αυτών των δευτερευουσών λειτουργιών.

Το backend ενσωματώνει επίσης τον μηχανισμό των **Workflows** του Medusa/Mercur, που υλοποιεί τις **σύνθετες ροές εκτέλεσης πολλαπλών βημάτων**. Ένα workflow ενορχηστρώνει την εκτέλεση μιας ακολουθίας ενεργειών που πρέπει να ολοκληρωθούν με συνέπεια - όπως η δημιουργία ενός νέου προϊόντος (που περιλαμβάνει validation, αποθήκευση στη βάση, indexing για αναζήτηση, κλήση εξωτερικών services για AI ανάλυση), ή η ολοκλήρωση μιας παραγγελίας σε marketplace context (που παράγει διαχωρισμένες παραγγελίες ανά πωλητή, δεσμεύει το απόθεμα, ενεργοποιεί διαδικασίες πληρωμής και εκπέμπει τα σχετικά events). Ένα workflow υποστηρίζει την έννοια του **compensation** σε περίπτωση αποτυχίας, εξασφαλίζοντας την επαναφορά του συστήματος σε συνεπή κατάσταση όταν κάποιο βήμα αποτυγχάνει.

Από οργανωτική σκοπιά, το backend είναι ταυτόχρονα και η **κωδικοβάση που φιλοξενεί το Medusa Admin Panel** που παρουσιάστηκε στην προηγούμενη ενότητα. Αυτή η συν-υπαρξή τους σε κοινό repository αποτελεί έναν συνηθισμένο αρχιτεκτονικό σχηματισμό που εξασφαλίζει τον απόλυτο συγχρονισμό μεταξύ των διαθέσιμων διαχειριστικών εργαλείων και των ενεργών δυνατοτήτων του πυρήνα. Παρά τη φυσική τους συνύπαρξη, ωστόσο, λειτουργικά παραμένουν διακριτές συνιστώσες με διαφορετικούς ρόλους στην αρχιτεκτονική.

Η **σύνδεση του backend με τα persistence layers** (PostgreSQL, μηχανή αναζήτησης, file storage) πραγματοποιείται μέσω εξειδικευμένων **abstraction layers** που παρέχει το MedusaJS. Συγκεκριμένα, η σχεσιακή πρόσβαση στη βάση δεδομένων διαμεσολαβείται από έναν Object-Relational Mapper (ORM), που μεταφράζει τις σύνθετες SQL queries σε αντικειμενοστραφή προγραμματιστική γλώσσα, ενώ η αποθήκευση αρχείων διαχειρίζεται μέσω της αρχιτεκτονικής των **File Service plugins**, η οποία επιτρέπει την εναλλαγή μεταξύ διαφορετικών παρόχων αποθήκευσης (Amazon S3, MinIO, local file system) χωρίς

αλλαγές στην επιχειρησιακή λογική. Η ευελιξία αυτή είναι κρίσιμη για το REFASHION καθώς επιτρέπει διαφορετικά setups μεταξύ development, staging και production περιβαλλόντων με ελάχιστη παραμετροποίηση.

Τέλος, η **σύνδεση με εξωτερικές υπηρεσίες** - Stripe Connect, SendGrid, TalkJS, Ethereum Blockchain και τα AI microservices - υλοποιείται μέσω εξειδικευμένων **integration modules** που εκτελούν τη μετάφραση μεταξύ της εσωτερικής λογικής του Medusa και των πρωτοκόλλων κάθε εξωτερικού συστήματος. Η προσέγγιση αυτή εξασφαλίζει ότι ο πυρήνας του backend παραμένει **απομονωμένος** από τις λεπτομέρειες υλοποίησης κάθε εξωτερικού provider, και επιτρέπει τη μελλοντική αντικατάσταση ή προσθήκη νέων integrations με ελάχιστη παρέμβαση στον υπόλοιπο κώδικα.

Συνολικά, το Medusa/Mercur Backend λειτουργεί ως ο **κεντρικός ενορχηστρωτής** του οικοσυστήματος REFASHION, ενοποιώντας την επιχειρησιακή λογική, τη διαχείριση δεδομένων και τις διασυνδέσεις με εξωτερικά συστήματα κάτω από ένα συνεκτικό αρχιτεκτονικό μοντέλο. Η αναλυτική παρουσίαση της εσωτερικής οργάνωσης της επιχειρησιακής λογικής παρατίθεται στην επόμενη υποπαράγραφο.

### 3.3.5.2 Επίπεδο Επιχειρησιακής Λογικής

Το **Επίπεδο Επιχειρησιακής Λογικής (Business Logic Layer)** συγκεντρώνει το σύνολο των κανόνων, των αλγορίθμων και των ροών που υλοποιούν τη συμπεριφορά της πλατφόρμας REFASHION. Σε αυτό το επίπεδο αποτυπώνεται **η ουσία του τι κάνει το σύστημα**: πώς υπολογίζονται οι τιμές, πώς δημιουργούνται και διαχωρίζονται οι παραγγελίες, πώς αποδίδονται οι ανταμοιβές, πώς ελέγχεται η εγκυρότητα μιας ενέργειας και πώς συντονίζονται τα διάφορα υποσυστήματα μεταξύ τους. Από αρχιτεκτονική σκοπιά, το επίπεδο αυτό λειτουργεί ως ο **εγκέφαλος** της πλατφόρμας, διαμεσολαβώντας μεταξύ των αιτημάτων του API επιπέδου και των δεδομένων που βρίσκονται στο επίπεδο persistence.

Η οργάνωση της επιχειρησιακής λογικής στο REFASHION ακολουθεί την **τριμερή αρχιτεκτονική φιλοσοφία** του MedusaJS - Services, Plugins και Modules - με την προσθήκη ενός τέταρτου, εξίσου σημαντικού στοιχείου: του **Event Bus** που ενορχηστρώνει την ασύγχρονη επικοινωνία μεταξύ των επιμέρους components. Η συνδυαστική λειτουργία αυτών των τεσσάρων μηχανισμών εξασφαλίζει ότι η λογική του συστήματος παραμένει **μορφοποιημένη, επεκτάσιμη και συντηρήσιμη**, ακόμη και καθώς η πλατφόρμα εξελίσσεται με νέες λειτουργίες.

Στον **πρώτο πυλώνα** του επιπέδου βρίσκονται τα **Core Services** του MedusaJS και του Mercur. Πρόκειται για κλάσεις TypeScript που ενσωματώνουν τη βασική λογική κάθε οντότητας του συστήματος, ακολουθώντας το pattern του **service-oriented design**. Στο REFASHION αξιοποιούνται οι εξής βασικές υπηρεσίες: το **ProductService** που διαχειρίζεται τις λειτουργίες των προϊόντων (δημιουργία, ενημέρωση, ανάκτηση, αναζήτηση), το **OrderService** που υλοποιεί τη λογική των παραγγελιών συμπεριλαμβανομένης της δημιουργίας **order sets** για multi-vendor checkouts, το **CartService** που ευθύνεται για τη ροή του καλαθιού (προσθήκη/αφαίρεση line items, εφαρμογή εκπτώσεων, υπολογισμό φόρων), το **CustomerService** που διαχειρίζεται τους αγοραστές και τις διευθύνσεις τους, το **SellerService** για τη διαχείριση των πωλητών και των αντίστοιχων ownership checks, καθώς και βοηθητικές υπηρεσίες όπως το **RegionService** (γεωγραφικές περιοχές, νομίσματα, φορολογικοί συντελεστές) και το **InventoryService** (διαχείριση αποθεμάτων και reservations).

Η αλληλεπίδραση μεταξύ των services υλοποιείται μέσω του μηχανισμού του **Dependency Injection (DI)**, που αποτελεί θεμελιώδες αρχιτεκτονικό pattern του MedusaJS. Αντί κάθε

service να δημιουργεί απευθείας τα services που χρειάζεται για να εκτελέσει τη λειτουργία του, οι εξαρτήσεις του **εγγέονται αυτόματα** από το framework κατά τη φάση εκκίνησης. Η προσέγγιση αυτή προσφέρει σημαντικά πλεονεκτήματα: μειώνει τη σύζευξη μεταξύ των services, διευκολύνει το **unit testing** μέσω της δυνατότητας αντικατάστασης πραγματικών υπηρεσιών με mocks, και επιτρέπει την επέκταση ή αντικατάσταση συγκεκριμένων services χωρίς να επηρεάζεται ο υπόλοιπος κώδικας.

Στον **δεύτερο πυλώνα** βρίσκονται τα **REFASHION Plugins**, που αποτελούν τα αυτόνομα πακέτα κώδικα στα οποία ενσωματώνονται οι **καινοτόμες λειτουργίες** του έργου. Όπως αναλύθηκε στην ενότητα 2.1.3, η αρχιτεκτονική των plugins επιτρέπει την προσθήκη νέας λειτουργικότητας χωρίς παρέμβαση στον πυρήνα του Medusa, εξασφαλίζοντας τη συντηρησιμότητα της κωδικοβάσης και τη δυνατότητα αναβάθμισης του υποκείμενου framework. Στο πλαίσιο του REFASHION έχουν αναπτυχθεί τρία εξειδικευμένα plugins: το **refashion-blockchain-plugin** που υλοποιεί τη σύνδεση με το Ethereum Blockchain για την καταγραφή συναλλαγών και τη διαχείριση των ReWear Tokens, το **refashion-eco-impact-plugin** που διασυνδέεται με την εξωτερική υπηρεσία υπολογισμού του Eco-Impact Score και ενημερώνει τα metadata των προϊόντων με τους αντίστοιχους περιβαλλοντικούς δείκτες, και το **refashion-gamification-plugin** που διαχειρίζεται την απόδοση πόντων, την αναβάθμιση επιπέδων χρηστών και την απονομή badges. Κάθε plugin περιέχει τα δικά του custom services, custom data models, custom API endpoints και custom subscribers, λειτουργώντας ως μια **πλήρως αυτόνομη μονάδα** μέσα στο ευρύτερο οικοσύστημα.

Ο  **τρίτος πυλώνας** αφορά τα **Modules**, τα οποία αντιπροσωπεύουν αντικαταστάσιμες υλοποιήσεις βασικών λειτουργιών του Medusa. Σε αντίθεση με τα plugins που **προσθέτουν** νέα λειτουργικότητα, τα modules **αντικαθιστούν ή εξειδικεύουν** υπάρχουσες δυνατότητες με custom υλοποιήσεις. Στο REFASHION αξιοποιούνται modules για τη διαχείριση των **πληρωμών** (με ενσωμάτωση του Stripe Connect για multi-vendor payouts), των **αποστολών** (αρχικά μέσω του manual fulfillment provider, με προοπτική μελλοντικής επέκτασης για smart lockers), και των **ειδοποιήσεων** (μέσω SendGrid για emails και push notifications). Η αρχιτεκτονική αυτή εξασφαλίζει ότι η εναλλαγή μεταξύ διαφορετικών παρόχων ή η αναβάθμιση συγκεκριμένων λειτουργιών μπορεί να γίνει με **εστιασμένες αλλαγές**, χωρίς εκτεταμένη αναδιάρθρωση του κώδικα.

Ο **τέταρτος και ζωτικής σημασίας** συστατικός μηχανισμός του επιπέδου είναι το **Event Bus** του Medusa, μέσω του οποίου επιτυγχάνεται η **ασύγχρονη και αποσυζευγμένη επικοινωνία** μεταξύ των διαφορετικών components. Στον πυρήνα του Event Bus βρίσκονται οι **Subscribers** - εξειδικευμένες κλάσεις που "ακούν" συγκεκριμένα events και εκτελούν αντίστοιχες ενέργειες όταν αυτά πυροδοτούνται. Η αρχιτεκτονική αυτή επιτρέπει την υλοποίηση **σύνθετων ροών reaction** χωρίς να εισάγεται απευθείας coupling μεταξύ των συστατικών που τις παράγουν και εκείνων που τις καταναλώνουν.

Στο REFASHION, το Event Bus αποτελεί τη **σπονδυλική στήλη** για τις καινοτόμες λειτουργίες. Όταν ολοκληρώνεται μια παραγγελία, πυροδοτείται το event **order.placed**. Στο event αυτό είναι εγγεγραμμένοι πολλαπλοί subscribers: ο ένας αναλαμβάνει την **καταγραφή της συναλλαγής στο blockchain** μέσω κλήσης στο smart contract, ο άλλος ενεργοποιεί τον **υπολογισμό και την απόδοση των ReWear Tokens** στους εμπλεκόμενους χρήστες, ένας τρίτος ενημερώνει τα **gamification στοιχεία** του πωλητή (αύξηση transactions, πιθανή απονομή νέου badge), και ένας τέταρτος **εκπέμπει την ειδοποίηση** προς τους εμπλεκόμενους μέσω SendGrid. Όλα αυτά συμβαίνουν παράλληλα και ανεξάρτητα, χωρίς το αρχικό OrderService που δημιούργησε την παραγγελία να γνωρίζει την ύπαρξή τους. Η ίδια λογική εφαρμόζεται και σε άλλα events του συστήματος: το **customer.created** ενεργοποιεί τη δημιουργία του gamification profile και την απονομή του welcome badge,

το **product.published** ενεργοποιεί τον υπολογισμό του Eco-Impact Score και το indexing στη μηχανή αναζήτησης, κ.ο.κ.

Παράλληλα με τους Subscribers, το επίπεδο επιχειρησιακής λογικής αξιοποιεί τον μηχανισμό των **Workflows** για την ενορχήστρωση **σύνθετων ροών εκτέλεσης πολλαπλών βημάτων**. Σε αντίθεση με τους Subscribers που αντιδρούν ασύγχρονα σε events, τα Workflows εκτελούν **συγχρονισμένες ακολουθίες ενεργειών** που πρέπει να ολοκληρωθούν με συνέπεια, με δυνατότητα **compensation** σε περίπτωση αποτυχίας οποιουδήποτε βήματος. Παράδειγμα τέτοιου workflow είναι ο **createProductRequestWorkflow** που εκτελείται κατά την υποβολή νέου προϊόντος από πωλητή: επικυρώνει τα στοιχεία, καλεί τις AI υπηρεσίες για την ανάλυση εικόνας και την εξαγωγή metadata, αποθηκεύει τα δεδομένα στη βάση, δημιουργεί το request προς moderation και εκπέμπει τα σχετικά events - με κάθε βήμα να μπορεί να ακυρωθεί και να αντιστραφεί αν κάποιο επόμενο αποτύχει. Αντίστοιχα, το cart completion workflow στο marketplace context ευθύνεται για τον **διαχωρισμό της ενιαίας παραγγελίας** του αγοραστή σε επιμέρους παραγγελίες ανά πωλητή, τη δέσμευση των αποθεμάτων, την ενεργοποίηση των payment intents και την παραγωγή του τελικού order set.

Το επίπεδο επιχειρησιακής λογικής επιτελεί επίσης το ρόλο του **κεντρικού φορέα ελέγχου εξουσιοδότησης (authorization)**. Οι βασικοί κανόνες αυθεντικοποίησης εφαρμόζονται στο επίπεδο του API μέσω middleware, ωστόσο οι **πιο εξειδικευμένοι κανόνες ιδιοκτησίας** - όπως η επαλήθευση ότι ένας πωλητής έχει πρόσβαση μόνο στα δικά του προϊόντα και παραγγελίες - υλοποιούνται μέσα στα services. Αυτή η προσέγγιση εξασφαλίζει ότι ακόμη και σε περίπτωση παράκαμψης του API επιπέδου (π.χ. σε internal κλήσεις μεταξύ services), οι κανόνες ιδιοκτησίας εξακολουθούν να εφαρμόζονται, διατηρώντας την **ακεραιότητα του multi-vendor μοντέλου**.

Συνολικά, η αρχιτεκτονική του επιπέδου επιχειρησιακής λογικής του REFASHION συνδυάζει την **ωριμότητα και τη στιβαρότητα** του πυρήνα MedusaJS/Mercur με τη **σαφή απομόνωση** των εξατομικευμένων επεκτάσεων του έργου, εξασφαλίζοντας μια κωδικοβάση που είναι ταυτόχρονα **σταθερή και εξελισσόμενη**. Οι αλληλεπιδράσεις μεταξύ των διαφορετικών components γίνονται μέσω σαφώς ορισμένων διεπαφών, ενώ ο συνδυασμός σύγχρονης (services, workflows) και ασύγχρονης (event bus, subscribers) επικοινωνίας εξασφαλίζει τόσο τη **συνέπεια των δεδομένων** όσο και τη **διακριτή απομόνωση των ευθυνών** μεταξύ των διαφόρων υποσυστημάτων.

### 3.3.5.3 Βάση Δεδομένων PostgreSQL

Η **PostgreSQL** αποτελεί το **κύριο σύστημα persistence** της πλατφόρμας REFASHION και λειτουργεί ως η **μοναδική πηγή αλήθειας (single source of truth)** για το σύνολο των επιχειρησιακών δεδομένων του συστήματος. Σε αυτήν αποθηκεύονται όλες οι βασικές οντότητες - προϊόντα, παραγγελίες, χρήστες, πωλητές, καλάθια, διευθύνσεις - καθώς και οι εξειδικευμένες οντότητες που εισάγουν οι καινοτόμες λειτουργίες του REFASHION. Η επιλογή της PostgreSQL ως βάσης δεδομένων δεν είναι συμπτωματική: αποτελεί μία από τις **πιο ώριμες και αξιόπιστες** σχεσιακές βάσεις ανοιχτού κώδικα παγκοσμίως, με ισχυρή υποστήριξη από κοινότητα και εμπορικούς προμηθευτές, και με χαρακτηριστικά που την καθιστούν ιδανική για συστήματα ηλεκτρονικού εμπορίου.

Από τεχνικής σκοπιάς, η PostgreSQL προσφέρει πληθώρα χαρακτηριστικών που είναι **κρίσιμα για την αρχιτεκτονική του REFASHION**. Η πλήρης υποστήριξη του προτύπου **ACID (Atomicity, Consistency, Isolation, Durability)** εξασφαλίζει ότι οι συναλλαγές της πλατφόρμας - ιδιαίτερα οι οικονομικές, όπως η ολοκλήρωση παραγγελιών και η απόδοση ανταμοιβών - εκτελούνται με απόλυτη ακεραιότητα, ακόμη και υπό συνθήκες αστοχίας του συστήματος. Η υποστήριξη **complex transactions** με multi-statement και multi-table

operations είναι θεμελιώδης για τη λειτουργία του multi-vendor μοντέλου, όπου μια ενιαία αγοραστική ενέργεια μπορεί να επηρεάζει δεκάδες πίνακες ταυτόχρονα (παραγγελία, line items, αποθέματα, payments, fulfillments, blockchain transactions, gamification updates).

Παράλληλα, η PostgreSQL προσφέρει **εξαιρετική υποστήριξη JSON/JSONB** μέσω εγγενών τύπων δεδομένων, χαρακτηριστικό κρίσιμο για το REFASHION καθώς ο πυρήνας του Medusa αξιοποιεί εκτενώς το **metadata field** ως μηχανισμό επέκτασης χωρίς αλλαγή του schema. Στο μοντέλο δεδομένων του REFASHION, σχεδόν κάθε οντότητα διαθέτει ένα πεδίο **metadata** τύπου JSONB, στο οποίο αποθηκεύονται custom πληροφορίες όπως ο Eco-Impact Score ενός προϊόντος, το wallet address ενός πελάτη, τα gamification στοιχεία ενός χρήστη ή τα στοιχεία αναγνώρισης ενός seller account. Η χρήση JSONB έναντι του απλού JSON προσφέρει σημαντικά πλεονεκτήματα: επιτρέπει την **εκτέλεση ευρετηριασμένων ερωτημάτων** μέσω GIN indexes, υποστηρίζει εξειδικευμένους τελεστές για queries πάνω σε JSON πεδία, και εξασφαλίζει την **αποδοτική αποθήκευση** ακόμη και για μεγάλα και σύνθετα έγγραφα.

Σημαντική θέση κατέχει επίσης η υποστήριξη **full-text search** που προσφέρει η PostgreSQL μέσω των tsvector και tsquery τύπων, χαρακτηριστικό που εν δυνάμει μπορεί να αξιοποιηθεί ως **fallback μηχανισμός αναζήτησης** σε σενάρια όπου η εξειδικευμένη μηχανή αναζήτησης δεν είναι διαθέσιμη. Επιπρόσθετα, η υποστήριξη **stored procedures, triggers, και custom functions** μέσω PL/pgSQL επιτρέπει την υλοποίηση εξειδικευμένης λογικής σε επίπεδο βάσης δεδομένων, ενώ τα **table inheritance** και **partitioning** features παρέχουν τη δυνατότητα οριζόντιας κλιμάκωσης για μεγάλους πίνακες - λειτουργικότητα κρίσιμη για το REFASHION όταν η πλατφόρμα ξεπεράσει σε όγκο τα αρχικά πιλοτικά της στάδια.

Η **οργάνωση του schema** της βάσης δεδομένων ακολουθεί τη δομή των **οντοτήτων (entities)** που ορίζει το MedusaJS/Mercur, εμπλουτισμένη με τις custom οντότητες του REFASHION. Στον πυρήνα του schema βρίσκονται οι **θεμελιώδεις πίνακες**: ο **product** με τις παραλλαγές του (**product\_variant**) και τα συσχετιζόμενα χαρακτηριστικά (collections, tags, options), ο **customer** με τις διευθύνσεις και τα customer groups, ο **seller** που εισάγει το Mercur για την υποστήριξη του multi-vendor μοντέλου, ο **cart** με τα line items του, ο **order** (καθώς και η εξειδικευμένη οντότητα **order\_set** που ομαδοποιεί παραγγελίες ανά αγοραστή στο marketplace context), και πίνακες υποστηρικτικής φύσης όπως οι **region, currency, tax\_rate** και **shipping\_option**.

Πέρα από τις τυπικές οντότητες, το schema του REFASHION εμπλουτίζεται με **custom πίνακες** που εξυπηρετούν τις καινοτόμες λειτουργίες της πλατφόρμας. Ο πίνακας **swap\_request** διαχειρίζεται τη ροή των ανταλλαγών μεταξύ χρηστών, αποθηκεύοντας την κατάσταση κάθε ανταλλαγής (pending, accepted, rejected, completed), τους εμπλεκόμενους χρήστες και τα προϊόντα τους, καθώς και τις τελικές παραγγελίες που παράγονται. Ο πίνακας **gamification\_profile** φιλοξενεί τα στατιστικά συμμετοχής κάθε χρήστη στους μηχανισμούς παιχνιδιοποίησης - total points, current level, completed transactions, καθώς και το JSON array των ανακτηθέντων badges. Αντίστοιχα, ο πίνακας **blockchain\_transaction** καταγράφει τις συναλλαγές που έχουν προωθηθεί στο Ethereum blockchain, διατηρώντας την αντιστοίχιση μεταξύ των εσωτερικών παραγγελιών του Medusa και των επί της αλυσίδας transaction hashes, ενώ ο πίνακας **reward\_history** φυλάσσει το ιστορικό των ReWear Tokens που έχουν αποδοθεί σε κάθε χρήστη.

Η **διαχείριση των αλλαγών στο schema** πραγματοποιείται μέσω του μηχανισμού των **migrations** του MedusaJS. Κάθε αλλαγή στη δομή της βάσης - προσθήκη νέου πεδίου, δημιουργία νέου πίνακα, τροποποίηση constraint - εγγράφεται σε ένα versioned migration file που αποτελεί μέρος της κωδικοβάσης. Η προσέγγιση αυτή εξασφαλίζει ότι η εξέλιξη του

schema είναι **παρακολουθήσιμη, αναπαραγώγιμη και αντιστρέψιμη**, ενώ επιτρέπει τη συγχρονισμένη ανάπτυξη μεταξύ διαφορετικών περιβαλλόντων (development, staging, production) χωρίς χειροκίνητη παρέμβαση. Σε περίπτωση που μια αναβάθμιση εισάγει προβλήματα, το migration μπορεί να αντιστραφεί (rollback) μέσω της εκτέλεσης της αντίθετης ενέργειας που έχει προβλεφθεί στον αντίστοιχο κώδικα.

Η **αλληλεπίδραση των services με τη βάση δεδομένων** δεν πραγματοποιείται με απευθείας SQL queries, αλλά μέσω ενός **Object-Relational Mapper (ORM)** που παρέχει το MedusaJS. Το ORM μεταφράζει τις λειτουργίες σε αντικειμενοστραφή γλώσσα - δηλαδή σε χειρισμό instances κλάσεων TypeScript - σε αντίστοιχα SQL statements, εξασφαλίζοντας την **πληρότητα του τυπικού ελέγχου (type safety)** στις διεργασίες πρόσβασης σε δεδομένα. Η προσέγγιση αυτή προσφέρει πολλαπλά πλεονεκτήματα: μειώνει δραστικά τις πιθανότητες SQL injection ευπαθειών, βελτιώνει την αναγνωσιμότητα και τη συντηρησιμότητα του κώδικα, και επιτρέπει την **εναλλαγή της υποκείμενης βάσης δεδομένων** χωρίς εκτεταμένες αλλαγές στον κώδικα - αν και στην πράξη το REFASHION δεσμεύεται στην PostgreSQL λόγω των εξειδικευμένων χαρακτηριστικών της.

Από αρχιτεκτονική σκοπιά, η PostgreSQL αναπτύσσεται σε **container-based περιβάλλον**, ακολουθώντας τη συνολική προσέγγιση containerization που υιοθετεί η πλατφόρμα. Αυτό επιτρέπει την **εύκολη αναπαραγωγή** του ίδιου περιβάλλοντος σε development, staging και production, ενώ εξασφαλίζει την **απομόνωση** της βάσης από τα υπόλοιπα systems της πλατφόρμας. Σε production περιβάλλον, η βάση συνοδεύεται από εξειδικευμένες ρυθμίσεις **αντιγράφων ασφαλείας (backups), point-in-time recovery** και **monitoring**, εξασφαλίζοντας ότι σε περίπτωση αστοχίας τα δεδομένα μπορούν να ανακτηθούν με ελάχιστη απώλεια.

Σε επίπεδο **απόδοσης**, η βάση δεδομένων αξιοποιεί τον μηχανισμό των **indexes** της PostgreSQL για τη βελτιστοποίηση των πιο συχνά εκτελούμενων ερωτημάτων. Ευρετήρια εφαρμόζονται τόσο σε τυπικά πεδία (primary keys, foreign keys, λοιπά πεδία που εμπλέκονται σε JOIN λειτουργίες) όσο και σε εξειδικευμένα σενάρια αναζήτησης μέσω **GIN indexes** για πεδία JSONB. Παράλληλα, η βάση υποστηρίζεται από **connection pooling** στο επίπεδο του backend, χαρακτηριστικό κρίσιμο για την αποδοτική διαχείριση των ταυτόχρονων συνδέσεων από πολλαπλά API requests και background processes. Με την προσέγγιση αυτή, η PostgreSQL μπορεί να εξυπηρετήσει αξιόπιστα έναν σημαντικό όγκο ταυτόχρονης χρήσης χωρίς να αποτελέσει σημείο συμφόρησης (bottleneck) στη συνολική αρχιτεκτονική.

Η **PostgreSQL αξιοποιείται επιπρόσθετα** ως **πηγή δεδομένων** για ένα από τα AI microservices της πλατφόρμας - το Recommendation Service - το οποίο συνδέεται απευθείας στη βάση μέσω εξειδικευμένου connection pool για την ανάλυση των ιστορικών παραγγελιών και αλληλεπιδράσεων των χρηστών. Η προσέγγιση αυτή, παρότι σπάει την αυστηρή απομόνωση μεταξύ υπηρεσιών, εξυπηρετεί την **αποδοτικότητα του αλγορίθμου** συστάσεων, ο οποίος χρειάζεται απευθείας πρόσβαση σε μεγάλο όγκο δεδομένων χωρίς τη διαμεσολάβηση του Medusa backend. Η περιορισμένη πρόσβαση στο επίπεδο της βάσης ελέγχεται μέσω εξειδικευμένων database roles, εξασφαλίζοντας ότι το microservice διαθέτει μόνο τα δικαιώματα ανάγνωσης που χρειάζεται.

Συνολικά, η PostgreSQL λειτουργεί ως ο **σταθερός θεμέλιος λίθος** του persistence layer της πλατφόρμας REFASHION, παρέχοντας ένα αξιόπιστο, ώριμο και εξαιρετικά λειτουργικό σύστημα αποθήκευσης που υποστηρίζει τόσο τις τυπικές λειτουργίες ηλεκτρονικού εμπορίου όσο και τις εξειδικευμένες απαιτήσεις των καινοτομιών του έργου.

### 3.3.5.4 Μηχανή Αναζήτησης & Αποθήκευση Αρχείων

Πέρα από τη σχεσιακή βάση PostgreSQL που λειτουργεί ως ο **κύριος μηχανισμός persistence** της πλατφόρμας, η αρχιτεκτονική του REFASHION περιλαμβάνει δύο επιπλέον εξειδικευμένα συστήματα αποθήκευσης, καθένα από τα οποία εξυπηρετεί ένα **συγκεκριμένο σύνολο αναγκών** που δεν θα μπορούσαν να καλυφθούν αποδοτικά από μια σχεσιακή βάση. Πρόκειται για τη **μηχανή αναζήτησης** που υποστηρίζει τις λειτουργίες ταχείας ανάκτησης και φιλτραρίσματος προϊόντων, και την **υπηρεσία αποθήκευσης αρχείων** που διαχειρίζεται τις εικόνες και τα λοιπά πολυμέσα της πλατφόρμας. Η συνύπαρξη αυτών των τριών διακριτών persistence συστημάτων αποτελεί εφαρμογή της αρχής της **εξειδίκευσης** που διέπει τη composable αρχιτεκτονική: κάθε είδος δεδομένων αποθηκεύεται και διαχειρίζεται από το σύστημα που το εξυπηρετεί καλύτερα.

#### Μηχανή Αναζήτησης

Η ταχεία και ευφυής αναζήτηση προϊόντων αποτελεί **θεμελιώδες χαρακτηριστικό** μιας σύγχρονης πλατφόρμας ηλεκτρονικού εμπορίου και έχει άμεση επίπτωση στην εμπειρία του χρήστη. Παρότι η PostgreSQL υποστηρίζει εγγενώς λειτουργίες full-text search, η αξιοποίησή της για αναζήτηση σε μεγάλο όγκο προϊόντων με σύνθετα κριτήρια φιλτραρίσματος (κατηγορία, μάρκα, μέγεθος, χρώμα, εύρος τιμής, κατάσταση) εμφανίζει σημαντικούς **περιορισμούς απόδοσης**, ιδιαίτερα όταν τα ερωτήματα συνδυάζουν πολλαπλά κριτήρια ή απαιτούν δυναμικό υπολογισμό διαθέσιμων επιλογών (faceted search). Για τον λόγο αυτό, η αρχιτεκτονική του REFASHION περιλαμβάνει μια **εξειδικευμένη μηχανή αναζήτησης** που λειτουργεί παράλληλα με την κύρια βάση δεδομένων.

Η μηχανή αναζήτησης φιλοξενεί **ευρετηριασμένες αναπαραστάσεις των προϊόντων** της πλατφόρμας, οργανωμένες σε δομές δεδομένων που είναι βελτιστοποιημένες για ταχεία ανάκτηση και φιλτράρισμα. Σε αντίθεση με τη σχεσιακή βάση που οργανώνει τα δεδομένα σε πίνακες με σαφή σχήμα και συσχετίσεις, η μηχανή αναζήτησης αποθηκεύει κάθε προϊόν ως ένα **πλήρως αυτοτελές έγγραφο** που περιέχει όλα τα σχετικά πεδία αναζήτησης σε **εξομαλυμένη μορφή** (denormalized). Αυτή η προσέγγιση εξαλείφει την ανάγκη για σύνθετα JOIN λειτουργιών κατά την εκτέλεση των ερωτημάτων, μειώνοντας δραματικά τους χρόνους απόκρισης.

Από αρχιτεκτονική σκοπιά, η μηχανή αναζήτησης λειτουργεί ως **ξεχωριστή υπηρεσία** που διασυνδέεται με το Medusa/Mercur backend μέσω ενός εξειδικευμένου plugin. Η συγχρονισμένη ενημέρωσή της επιτυγχάνεται μέσω του **Event Bus**: όποτε ένα προϊόν δημιουργείται, ενημερώνεται ή διαγράφεται, τα αντίστοιχα events (**product.created**, **product.updated**, **product.deleted**) πυροδοτούν τους εξειδικευμένους subscribers που ενημερώνουν το αντίστοιχο έγγραφο στη μηχανή αναζήτησης. Με αυτόν τον τρόπο, η μηχανή αναζήτησης διατηρείται διαρκώς **συγχρονισμένη** με την κύρια βάση δεδομένων, χωρίς να απαιτείται συγχρονισμένη εγγραφή σε δύο συστήματα ταυτόχρονα - μηχανισμός που θα δημιουργούσε προβλήματα συνέπειας σε περίπτωση αστοχίας.

Η μηχανή αναζήτησης υποστηρίζει πλούσιο σύνολο λειτουργιών που είναι **κρίσιμες για την εμπειρία χρήστη**. Η λειτουργία **instant search** εμφανίζει αποτελέσματα καθώς ο χρήστης πληκτρολογεί τον όρο αναζήτησης, με χρόνους απόκρισης κάτω των 50 χιλιοστών του δευτερολέπτου ακόμη και για καταλόγους εκατοντάδων χιλιάδων προϊόντων. Η λειτουργία **faceted search** υπολογίζει δυναμικά τα διαθέσιμα φίλτρα με βάση τα τρέχοντα αποτελέσματα - όταν ο χρήστης φιλτράρει για παράδειγμα στην κατηγορία "Φορέματα", η μηχανή υπολογίζει αυτόματα ποια μεγέθη, χρώματα και μάρκες είναι διαθέσιμα σε αυτή την υποκατηγορία και τα εμφανίζει με τους αντίστοιχους counts. Παράλληλα, υποστηρίζονται **typo-tolerant queries**, που επιτρέπουν την εύρεση

προϊόντων ακόμη και όταν ο χρήστης πληκτρολογεί λανθασμένα τον όρο αναζήτησης, καθώς και **prefix matching** για αυτόματη συμπλήρωση αναζητήσεων.

Στο επίπεδο του frontend, η αναζήτηση υλοποιείται μέσω ενός **URL-driven μηχανισμού** στο storefront, σύμφωνα με τον οποίο κάθε επιλογή του χρήστη μεταφράζεται σε query parameters (color, size, min\_price, max\_price, page, limit, sortBy). Η προσέγγιση αυτή προσφέρει σημαντικά πλεονεκτήματα: εξασφαλίζει **διατήρηση κατάστασης** μετά από ανανέωση της σελίδας, υποστηρίζει **deep-linking** σε εξειδικευμένες όψεις του καταλόγου, και διαχωρίζει σαφώς το presentation layer από την κατάσταση των φίλτρων. Η αναλυτική παρουσίαση της ροής αναζήτησης και φιλτραρίσματος παρατίθεται στην ενότητα 3.4 του παρόντος παραδοτέου.

Σημειώνεται ότι, παρά την κρισιμότητά της, η μηχανή αναζήτησης **δεν αποτελεί την πηγή αλήθειας** για τα δεδομένα των προϊόντων. Σε περίπτωση που το ευρετήριο αναζήτησης χαθεί ή χρειαστεί ανακατασκευή, μπορεί να **αναδημιουργηθεί πλήρως** από τη βάση δεδομένων PostgreSQL μέσω μιας διαδικασίας re-indexing που τρέχει όλα τα προϊόντα και τα προωθεί στη μηχανή. Αυτή η αρχιτεκτονική επιλογή εξασφαλίζει ότι, ακόμη και σε ακραίες περιπτώσεις απώλειας του search index, η ακεραιότητα της πλατφόρμας παραμένει άθικτη.

### Αποθήκευση Αρχείων (File Storage)

Η **αποθήκευση αρχείων** αποτελεί την τρίτη και τελευταία συνιστώσα του persistence layer, υπεύθυνη για τη διαχείριση του **οπτικού και πολυμεσικού υλικού** της πλατφόρμας - κυρίως των φωτογραφιών των προϊόντων που ανεβάζουν οι πωλητές. Σε αντίθεση με τη σχεσιακή βάση που είναι σχεδιασμένη για δομημένα δεδομένα και τη μηχανή αναζήτησης που εξυπηρετεί τα ευρετηριασμένα queries, η υπηρεσία αποθήκευσης αρχείων εξειδικεύεται στη διαχείριση **binary content** μεγάλου μεγέθους, με τρόπο που εξασφαλίζει αξιοπιστία, κλιμακωσιμότητα και χαμηλό κόστος.

Από αρχιτεκτονική σκοπιά, η πλατφόρμα REFASHION αξιοποιεί δύο εναλλακτικούς παρόχους αποθήκευσης, η επιλογή των οποίων εξαρτάται από το περιβάλλον deployment. Σε **παραγωγικό περιβάλλον (production)** χρησιμοποιείται το **Amazon S3**, μια ώριμη και αξιόπιστη υπηρεσία cloud object storage που προσφέρει υψηλά επίπεδα διαθεσιμότητας, αυτόματη ανθεκτικότητα στα δεδομένα και απεριόριστη πρακτική χωρητικότητα. Σε **self-hosted ή development περιβάλλον** αξιοποιείται το **MinIO**, μια open-source εναλλακτική λύση που είναι πλήρως συμβατή με το API του S3 και επιτρέπει την υλοποίηση αντίστοιχης λειτουργικότητας σε ιδιωτικές υποδομές χωρίς εξάρτηση από εξωτερικό πάροχο.

Η ευελιξία αυτή της εναλλαγής μεταξύ διαφορετικών παρόχων υλοποιείται μέσω της αρχιτεκτονικής των **File Service plugins** του MedusaJS. Ο πυρήνας του Medusa δεν γνωρίζει τις λεπτομέρειες του εκάστοτε παρόχου αποθήκευσης - αλληλεπιδρά μέσω μιας **τυποποιημένης διεπαφής** που εκτελεί τις βασικές λειτουργίες (upload, download, delete) και μεταφράζεται από το αντίστοιχο plugin στις ειδικές κλήσεις του εκάστοτε provider. Με αυτόν τον τρόπο, η εναλλαγή μεταξύ S3 και MinIO (ή ακόμη και τοπικού file system σε development) γίνεται μέσω **απλής αλλαγής configuration**, χωρίς καμία τροποποίηση του υπόλοιπου κώδικα.

Η **ροή αποθήκευσης ενός αρχείου** ακολουθεί μια σαφή και εξορθολογισμένη αρχιτεκτονική. Όταν ένας πωλητής ανεβάζει μια εικόνα μέσω του Seller Panel ή της εφαρμογής κινητού, η εικόνα διοχετεύεται στο εξειδικευμένο endpoint **POST /vendor/uploads** υπό μορφή multipart/form-data. Στο επίπεδο του backend, εφαρμόζεται το pattern του **multiplexer memory storage** που επιτρέπει την άμεση επεξεργασία του payload στη μνήμη, αποφεύγοντας την προσωρινή εγγραφή σε τοπικό δίσκο που θα

μείωνε την απόδοση. Τα αρχεία διοχετεύονται στη συνέχεια στο `uploadFilesWorkflow` του Medusa Core, το οποίο ενορχηστρώνει τη μεταφόρτωση μέσω του ενεργού File Service provider, ορίζοντας τις τεχνικές παραμέτρους κάθε αρχείου (filename, mimeType) και την ιδιότητα `access: "public"` που εξασφαλίζει την παραγωγή συνδέσμων **έτοιμων για άμεση προβολή** στο storefront.

Σημαντικό αρχιτεκτονικό χαρακτηριστικό είναι ο **διαχωρισμός μεταξύ μεταφόρτωσης και συσχέτισης**. Τα αρχεία αποθηκεύονται **αυτόνομα** στο file storage και αποκτούν σταθερά δημόσια URLs **πριν** συσχετιστούν με κάποια οντότητα προϊόντος. Με τον τρόπο αυτό, η διαδικασία ανεβάσματος αρχείων μπορεί να εκτελεστεί σε πρώτη φάση - όταν ο πωλητής επιλέγει τις φωτογραφίες - και η συσχέτιση με το προϊόν να γίνει αργότερα, κατά την οριστική υποβολή της καταχώρησης. Η ευελιξία αυτή είναι κρίσιμη για την υποστήριξη ροών όπως η **AI-assisted επεξεργασία εικόνων**: η εικόνα μπορεί να ανέβει, να σταλεί στα AI microservices για επεξεργασία (αφαίρεση φόντου, εξαγωγή metadata), και ο πωλητής να επιλέξει μεταξύ της πρωτότυπης και της επεξεργασμένης έκδοχής πριν την οριστική καταχώρηση.

Η διαχείριση αρχείων του REFASHION αλληλεπιδρά επίσης με τα **AI microservices** της πλατφόρμας μέσω ενός εξειδικευμένου μηχανισμού **προσωρινής φιλοξενίας**. Όταν μια εικόνα χρειάζεται να επεξεργαστεί από το Visual Product Intelligence Service για reverse image search, αποθηκεύεται προσωρινά σε εξειδικευμένο φάκελο και εκτίθεται μέσω ενός προσωρινού δημόσιου URL με μοναδικό token. Το token αυτό επιτρέπει στις εξωτερικές υπηρεσίες (όπως το Serper Lens) να προσπελάσουν την εικόνα για ανάλυση, ενώ μετά την ολοκλήρωση του request τα προσωρινά αρχεία **διαγράφονται αυτόματα**. Η αρχιτεκτονική αυτή εξυπηρετεί τη **διπλή απαίτηση** της πλατφόρμας: από τη μία πρέπει να επιτρέπεται η εξωτερική πρόσβαση σε εικόνες για AI επεξεργασία, από την άλλη η πλατφόρμα οφείλει να ελαχιστοποιεί τον χρόνο αποθήκευσης ευαίσθητων ή προσωπικών δεδομένων εικόνας.

Σε επίπεδο **απόδοσης και προσβασιμότητας**, η χρήση cloud object storage όπως το S3 προσφέρει σημαντικά πλεονεκτήματα. Τα αρχεία είναι προσβάσιμα μέσω **Content Delivery Network (CDN)**, εξασφαλίζοντας ταχύτερη παράδοση των εικόνων σε χρήστες ανεξάρτητα από τη γεωγραφική τους θέση. Παράλληλα, η αρχιτεκτονική αυτή **αποσυνδέει** τη φόρτο της εξυπηρέτησης στατικών αρχείων από τους servers της πλατφόρμας - οι εικόνες δεν διέρχονται από το backend κατά την προβολή τους, αλλά παραδίδονται απευθείας από το object storage στους χρήστες. Η αποσύνδεση αυτή είναι κρίσιμη για την κλιμακωσιμότητα: ακόμη και αν το backend αντιμετωπίσει υψηλό φόρτο, η εμπειρία περιήγησης στον κατάλογο παραμένει αμετάβλητη.

Συνολικά, ο συνδυασμός των τριών αυτών συστημάτων persistence - της σχεσιακής PostgreSQL, της μηχανής αναζήτησης και της υπηρεσίας αποθήκευσης αρχείων - συγκροτεί μια **ολοκληρωμένη και ισχυρή υποδομή δεδομένων**, καθένα από τα οποία είναι βελτιστοποιημένο για τις ανάγκες που εξυπηρετεί, αλλά παράλληλα διασυνδεδεμένο με τα υπόλοιπα μέσω σαφών διεπαφών και συγχρονισμένων ροών δεδομένων.

### 3.3.6 Εξωτερικές Υπηρεσίες & Microservices

Πέρα από τις συνιστώσες που αναπτύσσονται και συντηρούνται εξ ολοκλήρου εντός του πυρήνα της πλατφόρμας, η αρχιτεκτονική του REFASHION αξιοποιεί ένα **σύνολο εξωτερικών υπηρεσιών** που καλύπτουν εξειδικευμένες λειτουργικές ανάγκες. Η επιλογή ενσωμάτωσης αυτών των υπηρεσιών - αντί της ανάπτυξης ιδιόκτητων αντίστοιχων λύσεων - αποτελεί άμεση εφαρμογή της **composable αρχιτεκτονικής φιλοσοφίας** που αναλύθηκε στην ενότητα 2.1.2: η πλατφόρμα συνθέτει τις καλύτερες διαθέσιμες λύσεις για

κάθε επιμέρους λειτουργία, εστιάζοντας τους πόρους ανάπτυξης στα **πραγματικά καινοτόμα στοιχεία** του έργου.

Οι εξωτερικές υπηρεσίες της πλατφόρμας κατηγοριοποιούνται σε δύο διακριτές ομάδες. Η πρώτη περιλαμβάνει τις **υπηρεσίες τρίτων παρόχων (third-party SaaS)** που ενσωματώνονται από εξωτερικές εταιρείες με τις οποίες υπάρχει εμπορική σχέση - Stripe Connect για πληρωμές, SendGrid για ειδοποιήσεις, TalkJS για messaging - καθώς και το **Ethereum Blockchain** ως δημόσιο δίκτυο για την καταγραφή συναλλαγών. Η δεύτερη ομάδα περιλαμβάνει τα **αυτόνομα AI microservices** που έχουν αναπτυχθεί από την ίδια την ομάδα του έργου ως ξεχωριστές υπηρεσίες, με δικό τους κύκλο ζωής και δικιά τους υποδομή. Παρά τη διαφορετική προέλευσή τους, και οι δύο ομάδες ενσωματώνονται στην πλατφόρμα μέσω **τυποποιημένων διεπαφών** που εξασφαλίζουν την απομόνωση του πυρήνα από τις λεπτομέρειες υλοποίησης κάθε εξωτερικού συστήματος.

### 3.3.6.1 Stripe Connect

Η **διαχείριση των πληρωμών** σε μια πλατφόρμα marketplace αποτελεί μία από τις πιο **σύνθετες και κρίσιμες λειτουργίες**, καθώς συνδυάζει τεχνικές, οικονομικές, κανονιστικές και ζητήματα ασφαλείας. Η ανάπτυξη μιας ιδιόκτητης λύσης πληρωμών θα απαιτούσε εκτεταμένη συμμόρφωση με τα πρότυπα PCI DSS, την εγγραφή της πλατφόρμας ως ιδρύματος πληρωμών στις αρμόδιες ρυθμιστικές αρχές, και την ανάληψη τραπεζικών συμφωνιών για την επεξεργασία συναλλαγών με κάρτες - απαιτήσεις που εκφεύγουν τόσο του τεχνικού όσο και του οργανωτικού πεδίου ενός ερευνητικού έργου. Για τον λόγο αυτό, το REFASHION αξιοποιεί το **Stripe Connect**, μια εξειδικευμένη επέκταση της πλατφόρμας Stripe που είναι **σχεδιασμένη ειδικά για multi-vendor marketplaces**.

Το Stripe Connect διαφοροποιείται από το τυπικό Stripe σε ένα **καθοριστικό σημείο**: αντί η πλατφόρμα να λειτουργεί ως ο μοναδικός λήπτης των πληρωμών (merchant of record), το Stripe Connect επιτρέπει την **κατευθείαν ροή χρημάτων** από τους αγοραστές προς τους ξεχωριστούς λογαριασμούς των πωλητών, ενώ η πλατφόρμα παρακρατά αυτόματα ένα **προκαθορισμένο ποσοστό προμήθειας** ως marketplace fee. Η αρχιτεκτονική αυτή προσέγγιση εξυπηρετεί πολλαπλούς σκοπούς: μειώνει δραστικά τις κανονιστικές υποχρεώσεις της πλατφόρμας (καθώς δεν λειτουργεί ως ίδρυμα πληρωμών), εξαλείφει τους κινδύνους ρευστότητας (καθώς η πλατφόρμα δεν διαχειρίζεται κεντρικά τα κεφάλαια όλων των πωλητών), και απλοποιεί σημαντικά τη φορολογική και λογιστική διαχείριση τόσο για την πλατφόρμα όσο και για τους πωλητές.

Η **ενσωμάτωση του Stripe Connect** στην αρχιτεκτονική του REFASHION πραγματοποιείται σε δύο διακριτά αρχιτεκτονικά επίπεδα. Στο πρώτο επίπεδο βρίσκεται η **σύνδεση των πωλητών** με την πλατφόρμα Stripe. Κατά τη φάση onboarding ενός νέου πωλητή, το Seller Panel καθοδηγεί τον χρήστη μέσω της διαδικασίας δημιουργίας ή σύνδεσης ενός υπάρχοντος Stripe Connect Express Account. Η διαδικασία αυτή εξυπηρετείται από εξειδικευμένα endpoints του backend που ενεργοποιούν τη ροή του Stripe (**POST /vendor/sellers/stripe-onboarding**), παράγουν εκτενείς σύνδεσμους που οδηγούν τον πωλητή στο Stripe-hosted onboarding flow, και αφού ολοκληρωθεί η σύνδεση, αποθηκεύουν την αναγνωριστική ταυτότητα (Stripe Account ID) του πωλητή στο μοντέλο δεδομένων. Καθ' όλη τη διάρκεια, η πλατφόρμα **δεν διαχειρίζεται απευθείας** ευαίσθητα οικονομικά δεδομένα του πωλητή - όλη η συμπληρωματική επαλήθευση (αναγνωριστικά έγγραφα, τραπεζικοί λογαριασμοί, φορολογικά στοιχεία) διεκπεραιώνεται από τη Stripe σύμφωνα με τα ισχύοντα ρυθμιστικά πρότυπα Know Your Customer (KYC) και Anti-Money Laundering (AML).

Στο **δεύτερο επίπεδο** εντοπίζεται η **εκτέλεση των ίδιων των συναλλαγών**. Κατά την ολοκλήρωση μιας παραγγελίας από αγοραστή, ο Medusa κύκλος checkout ενεργοποιεί

τη **δημιουργία ενός Payment Intent** στο Stripe, μέσω του medusa-payment-stripe plugin. Σε ένα multi-vendor σενάριο, όπου ο αγοραστής αγοράζει προϊόντα από πολλούς διαφορετικούς πωλητές σε μία ενιαία ενέργεια, το Stripe Connect επιτρέπει τη χρήση του μηχανισμού των **destination charges**, που αυτόματα κατανέμει τα αντίστοιχα ποσά στους λογαριασμούς των διαφορετικών πωλητών, ενώ παρακρατά το ποσοστό προμήθειας της πλατφόρμας. Με αυτόν τον τρόπο, μία και μόνη χρέωση στην κάρτα του αγοραστή μεταφράζεται αυτόματα σε **πολλαπλά payouts προς τους εμπλεκόμενους πωλητές**, χωρίς να απαιτείται χειροκίνητη παρέμβαση ή πολύπλοκη λογική στο επίπεδο του backend.

Σημαντικό αρχιτεκτονικό χαρακτηριστικό είναι η **εξειδικευμένη μεταχείριση των δωρεών και ανταλλαγών**, λειτουργιών που αποτελούν διακριτικά στοιχεία του REFASHION. Στις περιπτώσεις αυτές, όπου η συναλλαγή έχει **μηδενικό οικονομικό κόστος**, ο τυπικός μηχανισμός του Stripe δεν εφαρμόζεται. Για την υποστήριξή τους έχει αναπτυχθεί ένας **custom Payment Provider** στο Medusa, ο οποίος αναγνωρίζει τις παραγγελίες με μηδενικό συνολικό ποσό και τις εγκρίνει αυτόματα, ενεργοποιώντας ταυτόχρονα όλα τα downstream events του Medusa Event Bus. Με τον τρόπο αυτό, οι ροές των δωρεών και των ανταλλαγών συμμετέχουν κανονικά στο σύνολο του οικοσυστήματος (blockchain logging, gamification rewards, notifications), παρότι δεν εμπλέκουν πραγματική κίνηση χρημάτων.

Η **ασφάλεια των συναλλαγών** επιτυγχάνεται μέσω της απευθείας ενσωμάτωσης του **Stripe.js SDK** στο frontend. Όταν ο αγοραστής εισάγει τα στοιχεία της κάρτας του, αυτά **δεν διέρχονται ποτέ από τους servers** της πλατφόρμας REFASHION - αποστέλλονται απευθείας από τον browser στους servers της Stripe, οι οποίοι επιστρέφουν ένα **secure token** που αντιπροσωπεύει τη συναλλαγή. Το token αυτό χρησιμοποιείται στη συνέχεια από το backend για την ολοκλήρωση της παραγγελίας. Με τον μηχανισμό αυτό, η πλατφόρμα **απαλλάσσεται από τις απαιτήσεις PCI DSS** που θα ίσχυαν αν διαχειριζόταν απευθείας δεδομένα κάρτας, διατηρώντας ταυτόχρονα μια εμπειρία χρήσης που είναι απόλυτα συνδεδεμένη με τη συνολική εμπειρία της πλατφόρμας.

Η **παρακολούθηση της κατάστασης των συναλλαγών** πραγματοποιείται μέσω του μηχανισμού των **Stripe webhooks**. Όταν συμβαίνουν σημαντικά γεγονότα στη Stripe - επιτυχής ή αποτυχημένη πληρωμή, dispute, refund, κατάσταση connected account - η Stripe αποστέλλει ασύγχρονα events στο backend μέσω HTTPS POST requests σε εξειδικευμένα webhook endpoints. Το backend επαληθεύει την αυθεντικότητα κάθε event μέσω signature verification, και στη συνέχεια ενημερώνει την κατάσταση των αντίστοιχων εσωτερικών οντοτήτων (παραγγελίες, payments). Η αρχιτεκτονική αυτή εξασφαλίζει την **εκ νέου συγχρονισμένη κατάσταση** μεταξύ της πλατφόρμας και του Stripe, ακόμη και σε περιπτώσεις όπου ο χρήστης έχει εγκαταλείψει την εφαρμογή πριν την ολοκλήρωση της πληρωμής.

Συνολικά, η ενσωμάτωση του Stripe Connect παρέχει στο REFASHION μια **τραπεζοδυναμική υποδομή πληρωμών παγκόσμιας κλάσης**, χωρίς την ανάγκη ανάπτυξης ιδιόκτητων μηχανισμών οικονομικής διαμεσολάβησης. Η αρχιτεκτονική προσέγγιση εξασφαλίζει τη βιωσιμότητα του multi-vendor μοντέλου, ενώ ταυτόχρονα παρέχει στους πωλητές μια **αξιόπιστη και διαφανή σχέση** με την πλατφόρμα, βασισμένη σε ξεκάθαρους κανόνες προμηθειών και άμεσης απόδοσης των εσόδων τους.

### 3.3.6.2 AI Microservices

Οι **υπηρεσίες Τεχνητής Νοημοσύνης** αποτελούν ένα από τα **κρίσιμα διαφοροποιητικά στοιχεία** της πλατφόρμας REFASHION και υλοποιούν τις πιο σύγχρονες τεχνολογικά δυνατότητες του οικοσυστήματος. Σε αντίθεση με τους τρίτους παρόχους που αναλύθηκαν στις προηγούμενες υποπαραγράφους, οι AI υπηρεσίες **αναπτύσσονται και**

**συντηρούνται από την ίδια ομάδα του έργου**, αποτελώντας ένα πρωτότυπο ερευνητικό αποτέλεσμα. Ωστόσο, σε αρχιτεκτονικό επίπεδο εντάσσονται στο επίπεδο των εξωτερικών υπηρεσιών διότι λειτουργούν ως **πλήρως αυτόνομα microservices**, με δικό τους κύκλο ζωής, δικιά τους τεχνολογική στοίβα, δικιά τους υποδομή και δικιά τους διεπαφή προς τον υπόλοιπο κόσμο. Η διάκριση αυτή είναι θεμελιώδης: το REFASHION δεν αποτελεί μια μονολιθική εφαρμογή με ενσωματωμένα AI components, αλλά ένα **οικοσύστημα ανεξάρτητων υπηρεσιών** που συνεργάζονται μέσω καθαρά ορισμένων διεπαφών.

Η επιλογή της microservices αρχιτεκτονικής για τις AI υπηρεσίες αποτελεί **συνειδητή τεχνολογική απόφαση** που εξυπηρετεί συγκεκριμένες απαιτήσεις. Πρώτον, οι AI υπηρεσίες έχουν **εντελώς διαφορετικές υπολογιστικές απαιτήσεις** από τον πυρήνα του e-commerce: ενώ το Medusa backend είναι κυρίως I/O-bound και χειρίζεται μικρά request payloads, οι AI υπηρεσίες είναι **CPU-bound ή GPU-bound** και ενδέχεται να χρειάζονται εξειδικευμένο υλικό επιτάχυνσης. Η απομόνωσή τους ως ξεχωριστά microservices επιτρέπει την **ανεξάρτητη κλιμάκωση** καθεμίας ανάλογα με τον φόρτο της, καθώς και την ανάπτυξη τους σε εξειδικευμένα GPU clusters χωρίς να επηρεάζεται το κόστος υποδομής του υπόλοιπου συστήματος. Δεύτερον, η AI επιστήμη εξελίσσεται με ταχύτατους ρυθμούς - νέα μοντέλα, νέοι αλγόριθμοι και νέες προσεγγίσεις εμφανίζονται διαρκώς. Η οργάνωση των AI λειτουργιών σε ανεξάρτητες υπηρεσίες επιτρέπει την **εστιασμένη αναβάθμιση** καθεμίας χωρίς επιπτώσεις στις υπόλοιπες, μέσω εναλλαγής μοντέλων ή ολόκληρης της υλοποίησης πίσω από μια σταθερή διεπαφή.

Από τεχνολογική σκοπιά, και οι τρεις AI υπηρεσίες υλοποιούνται σε **Python**, την κυρίαρχη γλώσσα της επιστήμης δεδομένων και της Μηχανικής Μάθησης, χρησιμοποιώντας το **FastAPI** ως web framework. Η επιλογή του FastAPI εξυπηρετεί συγκεκριμένες απαιτήσεις: παρέχει εξαιρετική απόδοση μέσω της ασύγχρονης αρχιτεκτονικής του, υποστηρίζει εγγενώς το πρότυπο **OpenAPI** για την τυποποιημένη τεκμηρίωση των διεπαφών, και ενσωματώνεται φυσικά με τη Python τη ευρύτερη κοινότητα μηχανικής μάθησης (PyTorch, TensorFlow, scikit-learn, transformers). Κάθε υπηρεσία αναπτύσσεται σε εξειδικευμένη **container-based υποδομή** μέσω Docker, εξασφαλίζοντας την αναπαραγωγιμότητα του περιβάλλοντος εκτέλεσης μεταξύ development, staging και production.

Η **ενσωμάτωση των AI υπηρεσιών** με το Medusa/Mercur backend πραγματοποιείται μέσω εξειδικευμένων **AI proxy routes** που εκτίθενται από το backend, σύμφωνα με το αρχιτεκτονικό μοτίβο που αναφέρθηκε στις προηγούμενες ενότητες. Όταν ένα frontend περιβάλλον χρειάζεται μια AI λειτουργία, δεν επικοινωνεί απευθείας με το αντίστοιχο microservice - αντιθέτως, υποβάλει αίτημα στο backend, το οποίο διαμεσολαβεί την κλήση. Η προσέγγιση αυτή προσφέρει σημαντικά πλεονεκτήματα: εξασφαλίζει την **ενιαία αυθεντικοποίηση** μέσω του ίδιου μηχανισμού JWT tokens που χρησιμοποιείται για όλες τις υπόλοιπες λειτουργίες, αποκρύπτει τα εσωτερικά microservices από το διαδίκτυο (τα οποία λειτουργούν αποκλειστικά εντός του εσωτερικού δικτύου του deployment), και επιτρέπει στο backend να εκτελεί **ενδιάμεση επεξεργασία** (logging, monitoring, rate limiting, caching) σε όλες τις AI κλήσεις. Παράλληλα, αυτή η διαμεσολάβηση καθιστά εύκολη τη μελλοντική αντικατάσταση ή προσθήκη νέων AI υπηρεσιών χωρίς αλλαγές στα frontend περιβάλλοντα.

### Image Analysis Service

Το **Image Analysis Service** αποτελεί το **πρωτογενές AI σημείο εισόδου** της πλατφόρμας REFASHION και ευθύνεται για τη βασική επεξεργασία των εικόνων προϊόντων κατά τη φάση καταχώρησής τους από τους πωλητές. Λειτουργεί ως εξειδικευμένο microservice που εκτίθεται μέσω HTTP API στο εσωτερικό δίκτυο της πλατφόρμας, με κύριο σκοπό την **οπτική**

**προετοιμασία** του υλικού πριν αυτό προωθηθεί σε άλλες, πιο εξειδικευμένες υπηρεσίες ή αποθηκευτεί ως τελικό προϊόν.

Η κύρια λειτουργικότητα της υπηρεσίας περιστρέφεται γύρω από την **αυτόματη αφαίρεση φόντου (background removal)** από φωτογραφίες προϊόντων. Στο πλαίσιο μιας πλατφόρμας επανεμπορίου, οι φωτογραφίες που ανεβάζουν οι πωλητές παρουσιάζουν συνήθως μεγάλη ανομοιογένεια: τραβιούνται σε διαφορετικά περιβάλλοντα, με διαφορετικό φωτισμό και αντικείμενα στο φόντο. Η αυτόματη αφαίρεση του φόντου και η αντικατάστασή του με ενιαίο, ουδέτερο background **ομοιογεντοποιεί την οπτική παρουσίαση** του καταλόγου, αυξάνοντας την αισθητική ποιότητα της πλατφόρμας και τη συγκρισιμότητα μεταξύ προϊόντων. Παράλληλα, η αφαίρεση του φόντου διευκολύνει την επεξεργασία από τις υπόλοιπες AI υπηρεσίες (όπως η οπτική αναζήτηση), καθώς εξαλείφει οπτικό θόρυβο που μπορεί να επηρεάσει αρνητικά την ακρίβεια των αλγορίθμων.

Από τεχνολογική σκοπιά, η υπηρεσία αξιοποιεί σύγχρονα μοντέλα **τμηματοποίησης εικόνας (image segmentation)** βασισμένα σε νευρωνικά δίκτυα. Τα μοντέλα αυτά εκπαιδεύονται να αναγνωρίζουν το **κύριο αντικείμενο** μιας εικόνας και να το διαχωρίζουν από το υπόλοιπο περιεχόμενο με ακρίβεια pixel επιπέδου. Η ροή επεξεργασίας ξεκινά με τη λήψη της εικόνας σε μορφή binary upload, ακολουθούν τα στάδια κανονικοποίησης (resize, color normalization), η εκτέλεση του μοντέλου τμηματοποίησης, η εφαρμογή της μάσκας που παράγεται και τέλος η εξαγωγή της επεξεργασμένης εικόνας σε ζητούμενη μορφή (συνήθως PNG με διαφάνεια ή JPEG με ουδέτερο φόντο).

Η ενσωμάτωση με τον πυρήνα της πλατφόρμας πραγματοποιείται κατά τη ροή καταχώρησης νέου προϊόντος. Όταν ένας πωλητής ανεβάζει μια φωτογραφία μέσω της εφαρμογής κινητού ή του Seller Panel, η εικόνα αποθηκεύεται αρχικά στο file storage και ταυτόχρονα διαβιβάζεται στο Image Analysis Service. Η επεξεργασμένη έκδοχή επιστρέφεται και αποθηκεύεται ως **εναλλακτική έκδοχή** της αρχικής εικόνας, παρέχοντας στον πωλητή τη δυνατότητα να επιλέξει μεταξύ της πρωτότυπης και της επεξεργασμένης πριν την τελική δημοσίευση της καταχώρησης. Η ροή αυτή εξυπηρετείται στην παρούσα φάση από εξειδικευμένο endpoint της υπηρεσίας που τρέχει σε εσωτερική θύρα 9100, διαμεσολαβημένο μέσω του αντίστοιχου proxy route του Medusa backend.

### Visual Product Intelligence Service

Το **Visual Product Intelligence Service** αποτελεί την **πιο εκτεταμένη και πολυσύνθετη** από τις AI υπηρεσίες της πλατφόρμας, και συνιστά ένα από τα **κορυφαία καινοτόμα στοιχεία** του REFASHION. Σε αντίθεση με το Image Analysis Service που εστιάζει σε μια εξειδικευμένη λειτουργία βασικής επεξεργασίας εικόνας, το Visual Intelligence συνδυάζει **πολλαπλές τεχνολογίες αιχμής** - αναζήτηση οπτικής ομοιότητας, κατανεμημένη συλλογή δεδομένων από το διαδίκτυο και ευφυή εξαγωγή δομημένης πληροφορίας μέσω Large Language Models - για να επιτύχει έναν φιλόδοξο στόχο: τον **αυτόματο εμπλουτισμό των καταχωρήσεων** των πωλητών με πληροφορίες που θα ήταν χρονοβόρες ή και αδύνατες να συλλεχθούν χειροκίνητα.

Η **λειτουργική ροή** της υπηρεσίας ξεκινά όταν ένας πωλητής ανεβάζει την εικόνα ενός προϊόντος. Η υπηρεσία λαμβάνει την εικόνα και εκτελεί τρεις φάσεις επεξεργασίας. Στην **πρώτη φάση**, εφαρμόζεται **οπτική αναζήτηση ομοιότητας** μέσω εξειδικευμένης μηχανής που εξαγεί υψηλής διάστασης διανύσματα χαρακτηριστικών (feature vectors) από την εικόνα. Τα διανύσματα αυτά παράγονται από προεκπαιδευμένα **convolutional neural networks** (όπως οι αρχιτεκτονικές ResNet-50 ή EfficientNet-B4) που έχουν προσαρμοστεί για την κατανόηση της οπτικής γλώσσας της μόδας. Με την αξιοποίηση μιας **διανυσματικής βάσης δεδομένων** (όπως FAISS ή Milvus), η υπηρεσία εντοπίζει οπτικά παρόμοια προϊόντα τόσο εντός του εσωτερικού καταλόγου της πλατφόρμας όσο και σε

εξωτερικά καταστήματα μέσω **reverse image search APIs** (Google Vision API, Bing Visual Search, Serper Lens). Το αποτέλεσμα είναι ένα σύνολο **URLs εξωτερικών σελίδων** που περιέχουν οπτικά παρόμοια προϊόντα, τα οποία λειτουργούν ως σπόροι για την επόμενη φάση.

Στη **δεύτερη φάση**, ενεργοποιείται ένας **κατανεμημένος μηχανισμός web scraping**, υλοποιημένος ως pipeline βασισμένο σε Apache Spark. Τα URLs που εντοπίστηκαν προωθούνται σε ουρές Kafka, κατατημένες κατά domain για την επιβολή per-domain rate limiting και τη συμμόρφωση με τις πολιτικές robots.txt των επισκεπτόμενων ιστοσελίδων. Ένα Spark Streaming job επεξεργάζεται τις ουρές, χρησιμοποιώντας **headless browsers** (Playwright, Selenium) για ιστοσελίδες που απαιτούν εκτέλεση JavaScript ή απλούστερους HTTP clients για στατικό περιεχόμενο. Παράλληλα, εφαρμόζονται κανόνες **adaptive rate limiting** και exponential backoff σε περιπτώσεις HTTP 429 αποκρίσεων, εξασφαλίζοντας **ηθική και υπεύθυνη συλλογή δεδομένων**. Το ακατέργαστο HTML κάθε σελίδας επεξεργάζεται από έναν content extractor που συνδυάζει CSS selectors, XPath patterns και machine learning τεχνικές αναγνώρισης περιεχομένου για την απομόνωση των πραγματικά κρίσιμων τμημάτων: περιγραφή προϊόντος, προδιαγραφές, τιμή και εικόνες.

Στην **τρίτη και κρίσιμη φάση**, το ακατέργαστο κείμενο που εξάγεται από τις εξωτερικές σελίδες προωθείται σε ένα **Large Language Model (LLM)** για τη μετατροπή της αδόμητης πληροφορίας σε **δομημένα μεταδεδομένα προϊόντος**. Η αρχιτεκτονική υποστηρίζει multiple LLM backends, με προτίμηση σε ανοιχτά μοντέλα όπως τα **LLaMA-2 και Mistral** (καθώς και τις instruction-tuned παραλλαγές τους), τα οποία εκτελούνται σε εξειδικευμένη υποδομή GPU για αποδοτικό inference. Μέσω **προσεκτικά σχεδιασμένων prompts**, το LLM καθοδηγείται στην εξαγωγή συγκεκριμένων ιδιοτήτων: σύσταση υλικού (material composition με ποσοστιαία αναλογία), διαστάσεις, εμπορική επωνυμία, αρχική τιμή λιανικής, χώρα προέλευσης, πιστοποιήσεις βιωσιμότητας, και άλλα. Όταν πολλαπλές πηγές παρέχουν αντιφατική πληροφορία, εφαρμόζεται ένας **weighted voting μηχανισμός** που συνδυάζει την αξιοπιστία της κάθε πηγής με τη συχνότητα εμφάνισης κάθε τιμής, καταλήγοντας στην πιο πιθανή σωστή απάντηση.

Το τελικό αποτέλεσμα της επεξεργασίας - ένα δομημένο JSON με όλα τα εξαχθέντα μεταδεδομένα - επιστρέφεται στο Medusa backend, το οποίο **προσυμπληρώνει αυτόματα** τα αντίστοιχα πεδία της φόρμας καταχώρησης του πωλητή. Ο πωλητής βλέπει τα προτεινόμενα δεδομένα και μπορεί να τα επιβεβαιώσει, να τα διορθώσει ή να τα συμπληρώσει, μετατρέποντας μια διαδικασία που παραδοσιακά απαιτούσε λεπτά πληκτρολόγησης σε μια ροή λίγων δευτερολέπτων επιβεβαίωσης. Η αρχιτεκτονική αυτή μειώνει δραστικά το **κόστος συμμετοχής** των πωλητών στην πλατφόρμα, ιδιαίτερα των ιδιωτών που ανεβάζουν περιστασιακά λίγα αντικείμενα και που σε διαφορετική περίπτωση θα απέφευγαν να αναλάβουν τη χρονοβόρα διαδικασία καταχώρησης. Η υπηρεσία εκτίθεται μέσω εσωτερικής θύρας 9300 και διαμεσολαβείται από το Medusa backend με την υπόλοιπη πλατφόρμα.

### Recommendation Service

Το **Recommendation Service** συμπληρώνει τη συστάδα των AI υπηρεσιών του REFASHION, υλοποιώντας τη **λειτουργία προσωποποιημένων προτάσεων** που ανυψώνει την εμπειρία περιήγησης από έναν στατικό κατάλογο σε ένα **δυναμικό, εξατομικευμένο οικοσύστημα** ανακάλυψης προϊόντων. Σε ένα marketplace με συνεχώς ανανεούμενο και διαφοροποιημένο κατάλογο μεταχειρισμένων ειδών - όπου κάθε προϊόν είναι μοναδικό και η ανακάλυψη του "κατάλληλου" αντικειμένου αποτελεί κρίσιμο στοιχείο της εμπειρίας - η

ικανότητα έξυπνης σύνδεσης χρηστών με προϊόντα αποτελεί **καθοριστικό παράγοντα επιτυχίας**.

Η υπηρεσία προτάσεων του REFASHION συνδυάζει **πολλαπλές αλγοριθμικές προσεγγίσεις**, καθεμία από τις οποίες αξιοποιεί διαφορετική πτυχή των διαθέσιμων δεδομένων. Η **collaborative filtering** προσέγγιση εντοπίζει χρήστες με παρόμοιο ιστορικό συμπεριφοράς και προτείνει σε καθέναν προϊόντα που έλκυσαν την προσοχή των "ομοίων" του. Η **content-based filtering** αναλύει τα χαρακτηριστικά των προϊόντων που έχει αλληλεπιδράσει ο χρήστης (κατηγορία, μάρκα, χρώμα, στυλ) και προτείνει προϊόντα με παρόμοιες ιδιότητες. Η **visual similarity** προσέγγιση αξιοποιεί τα feature vectors που παράγει το Visual Product Intelligence Service για να εντοπίσει οπτικά παρόμοια προϊόντα με αυτά που έχουν προηγουμένως αλληλεπιδράσει με τον χρήστη. Τέλος, οι **business rules** ενσωματώνουν επιχειρησιακούς κανόνες όπως η προτεραιότητα σε προϊόντα με υψηλό Eco-Impact Score, η ανάδειξη νέων πωλητών ή η προβολή προϊόντων εντός γεωγραφικής εγγύτητας.

Από αρχιτεκτονική σκοπιά, η υπηρεσία **διαφοροποιείται από τα υπόλοιπα AI microservices** σε ένα σημαντικό σημείο: αξιοποιεί απευθείας σύνδεση με τη **βάση δεδομένων PostgreSQL** της πλατφόρμας μέσω εξειδικευμένου connection pool. Η επιλογή αυτή, παρότι σπάει την αυστηρή απομόνωση μεταξύ υπηρεσιών που τηρείται στις άλλες περιπτώσεις, εξυπηρετεί τη **βέλτιστη αποδοτικότητα του αλγορίθμου**: ο αλγόριθμος συστάσεων χρειάζεται να αναλύσει μεγάλο όγκο δεδομένων (όλο το ιστορικό αλληλεπιδράσεων όλων των χρηστών, όλο τον κατάλογο προϊόντων, τις σχέσεις μεταξύ τους) και θα ήταν εξαιρετικά αναποτελεσματικό να αντλεί αυτά τα δεδομένα μέσω επαναλαμβανόμενων HTTP κλήσεων στο Medusa backend. Η πρόσβαση στο επίπεδο της βάσης ελέγχεται μέσω εξειδικευμένων **database roles** που παρέχουν στην υπηρεσία **μόνο δικαιώματα ανάγνωσης** σε συγκεκριμένους πίνακες, εξασφαλίζοντας ότι η περιορισμένη παράκαμψη της απομόνωσης δεν παραβιάζει την ασφάλεια του συστήματος.

Η **παραγωγή των προτάσεων** πραγματοποιείται σε **δύο διακριτές χρονικές κλίμακες**. Σε **batch επεξεργασία** που εκτελείται περιοδικά (π.χ. ωριαία ή ημερήσια), παράγονται **προ-υπολογισμένες προτάσεις** για κάθε χρήστη, οι οποίες αποθηκεύονται και επιστρέφονται άμεσα όταν ζητηθούν, εξασφαλίζοντας χρόνους απόκρισης κάτω των 100 χιλιοστών του δευτερολέπτου. Παράλληλα, για **real-time εμπλουτισμό** των προτάσεων με βάση την τρέχουσα συνεδρία του χρήστη (π.χ. αν μόλις προέβαλε ένα συγκεκριμένο προϊόν), εφαρμόζεται **on-the-fly υπολογισμός** που συνδυάζει τις προ-υπολογισμένες προτάσεις με την άμεση συμπεριφορά. Η συνδυαστική αυτή προσέγγιση εξασφαλίζει την **ισορροπία μεταξύ απόδοσης και προσαρμοστικότητας**.

Η ενσωμάτωση με την πλατφόρμα πραγματοποιείται μέσω εξειδικευμένων **API endpoints** που εκτίθενται από την υπηρεσία στη θύρα 9400. Τα frontend περιβάλλοντα, μέσω των αντίστοιχων proxy routes του Medusa backend, καλούν τα endpoints αυτά σε κρίσιμα σημεία της εμπειρίας χρήστη: στην αρχική σελίδα του storefront για τη δημιουργία **προσωποποιημένης βιτρίνας**, στη σελίδα προϊόντος για τη **σύσταση συναφών προϊόντων** ("Σας ενδιαφέρει ίσως..."), στη σελίδα ολοκλήρωσης παραγγελίας για τη **σύσταση συμπληρωματικών προϊόντων**, και στις προωθητικές ειδοποιήσεις για τη **στοχευμένη ενημέρωση** χρηστών για νέες καταχωρήσεις που αντιστοιχούν στις προτιμήσεις τους.

### 3.3.7 Ροές Επικοινωνίας

Έχοντας παρουσιάσει στις προηγούμενες υποπαραγράφους τις επιμέρους συνιστώσες της αρχιτεκτονικής - frontend περιβάλλοντα, backend, persistence layers, εξωτερικές υπηρεσίες και AI microservices - η ενότητα που ακολουθεί εστιάζει στον τρόπο με τον

οποίο αυτές οι συνιστώσες **συνεργάζονται** μεταξύ τους. Η μελέτη των ροών επικοινωνίας είναι κρίσιμη διότι αποκαλύπτει την **πραγματική δυναμική** του συστήματος: ένα αρχιτεκτονικό διάγραμμα μπορεί να δείχνει ποια components υπάρχουν, αλλά μόνο η ανάλυση των ροών εξηγεί πώς αυτά **αλληλεπιδρούν** για να εξυπηρετήσουν τις λειτουργικές ανάγκες της πλατφόρμας. Η αρχιτεκτονική του REFASHION αξιοποιεί δύο θεμελιωδώς διαφορετικά μοτίβα επικοινωνίας - το σύγχρονο και το ασύγχρονο - καθένα από τα οποία εξυπηρετεί διαφορετικές κατηγορίες αλληλεπιδράσεων, ενώ η συνδυαστική τους χρήση επιτρέπει στο σύστημα να συνδυάζει **άμεση απόκριση** στα αιτήματα των χρηστών με **αποσυζευγμένη επεξεργασία** των δευτερευουσών ενεργειών.

### 3.3.7.1 Σύγχρονη Επικοινωνία (Request/Response)

Η **σύγχρονη επικοινωνία** αποτελεί το **πιο διαδεδομένο μοτίβο** στην αρχιτεκτονική του REFASHION και δέπει τη συντριπτική πλειονότητα των αλληλεπιδράσεων μεταξύ frontend και backend. Στο μοτίβο αυτό, ο **αιτών (client)** στέλνει ένα αίτημα στον **παρόχο (server)**, αναμένει την επεξεργασία του και λαμβάνει μια απάντηση πριν συνεχίσει με τις επόμενες ενέργειες. Η σύγχρονη φύση της επικοινωνίας εξασφαλίζει ότι ο χρήστης βλέπει **άμεσα τα αποτελέσματα** των ενεργειών του, χαρακτηριστικό κρίσιμο για μια διαδραστική εφαρμογή ηλεκτρονικού εμπορίου όπου η εμπειρία χρήστη εξαρτάται από την ταχύτητα και την προβλεψιμότητα των αποκρίσεων.

Στο τεχνικό επίπεδο, η σύγχρονη επικοινωνία του REFASHION υλοποιείται μέσω του πρωτοκόλλου **HTTPS** και ακολουθεί το αρχιτεκτονικό μοτίβο **REST (Representational State Transfer)**. Κάθε σύγχρονη αλληλεπίδραση αποτελείται από ένα **HTTP request** που αποστέλλεται από το frontend προς ένα συγκεκριμένο endpoint του backend, και ένα αντίστοιχο **HTTP response** που επιστρέφει το backend. Η ροή δεδομένων είναι κωδικοποιημένη σε **JSON**, ενώ η αυθεντικοποίηση πραγματοποιείται μέσω **JWT tokens** που μεταβιβάζονται στην HTTP κεφαλίδα **Authorization** των αιτημάτων.

Η **τυπική διαδρομή ενός σύγχρονου αιτήματος** στο REFASHION ακολουθεί μια σαφή και προβλέψιμη ακολουθία βημάτων που διασχίζει όλα τα επίπεδα της αρχιτεκτονικής. Στο **πρώτο βήμα**, ένα frontend περιβάλλον - Web Storefront, εφαρμογή κινητού, Seller Panel ή Admin Panel - εκκινεί το αίτημα ως αποτέλεσμα μιας ενέργειας του χρήστη (π.χ. αναζήτηση προϊόντος, προσθήκη στο καλάθι, υποβολή φόρμας). Το αίτημα συντάσσεται με βάση τους κανόνες του Medusa REST API, περιλαμβάνοντας το κατάλληλο HTTP method (GET για ανάγνωση, POST για δημιουργία, PUT/PATCH για ενημέρωση, DELETE για διαγραφή), το αντίστοιχο URL, τυχόν query parameters, τα κατάλληλα HTTP headers (Content-Type, Authorization, x-publishable-api-key) και το payload σε JSON μορφή όπου αυτό απαιτείται.

Στο **δεύτερο βήμα**, το αίτημα φτάνει στο **API επίπεδο** του Medusa/Mercur backend, όπου εκτελείται η αρχική επεξεργασία από μια αλληλουχία **middleware** συναρτήσεων. Τα middleware αυτά εκτελούν διαδοχικά τυποποιημένες λειτουργίες: επικύρωση του Content-Type, parsing του JSON payload, εκτέλεση CORS ελέγχων, εξαγωγή και επαλήθευση του JWT token, ταυτοποίηση του χρήστη και του scope του (customer, seller, admin), επιβολή των βασικών κανόνων εξουσιοδότησης και - όπου χρειάζεται - δομική επικύρωση του payload μέσω schemas. Εάν οποιοδήποτε από αυτά τα βήματα αποτύχει, το αίτημα **απορρίπτεται άμεσα** με κατάλληλο HTTP error code (401 για μη αυθεντικοποίηση, 403 για μη εξουσιοδότηση, 400 για κακή δομή), χωρίς να φτάσει στο επίπεδο της επιχειρησιακής λογικής.

Εφόσον το αίτημα περάσει με επιτυχία τα middleware checks, εισέρχεται στο **τρίτο βήμα**, στο **επίπεδο της επιχειρησιακής λογικής**. Εδώ ο αντίστοιχος route handler ταυτοποιεί την επιθυμητή λειτουργία και προωθεί την επεξεργασία στα κατάλληλα **Services** ή **Workflows**.

Τα services εκτελούν τους **επιχειρησιακούς κανόνες** που σχετίζονται με την κάθε λειτουργία: επικύρωση του ότι ο εξουσιοδοτημένος χρήστης έχει δικαίωμα στη συγκεκριμένη ενέργεια (ownership checks), έλεγχος των επιχειρησιακών constraints (π.χ. διαθεσιμότητα αποθέματος, εγκυρότητα τιμολόγησης), εφαρμογή των κανόνων προσαρμογής (π.χ. εκπτώσεις, φόροι ανά region), και τελική εκτέλεση της αιτούμενης ενέργειας.

Όπου η ενέργεια απαιτεί **πρόσβαση σε δεδομένα**, εισερχόμαστε στο **τέταρτο βήμα**, στο **επίπεδο persistence**. Τα services δεν αλληλεπιδρούν απευθείας με τη βάση δεδομένων μέσω SQL, αλλά αξιοποιούν τον **Object-Relational Mapper (ORM)** του MedusaJS, ο οποίος μεταφράζει τις αντικειμενοστραφείς λειτουργίες σε αντίστοιχα SQL queries προς την PostgreSQL. Οι queries εκτελούνται μέσω **connection pool** που διαχειρίζεται αποδοτικά τις ταυτόχρονες συνδέσεις, ενώ - όπου χρειάζεται - εκτελούνται μέσα σε **database transactions** που εξασφαλίζουν την ατομικότητα πολλαπλών σχετικών αλλαγών. Παράλληλα, για λειτουργίες που εμπλέκουν εξειδικευμένα persistence layers - όπως η αναζήτηση μέσω της μηχανής αναζήτησης ή η ανάκτηση εικόνων από το file storage - γίνονται αντίστοιχες κλήσεις προς τις εξειδικευμένες υπηρεσίες.

Στο **πέμπτο βήμα**, εφόσον η ενέργεια απαιτεί **εξωτερική επικοινωνία** - όπως κλήση στο Stripe για επεξεργασία πληρωμής, αποστολή email μέσω SendGrid ή κλήση σε AI microservice για ανάλυση εικόνας - εκτελούνται οι αντίστοιχες HTTP κλήσεις. Σε αυτές τις περιπτώσεις, η ροή σύγχρονης επικοινωνίας μπορεί να **παραταθεί** καθώς αναμένεται απόκριση από τους εξωτερικούς παρόχους. Για την αντιμετώπιση αυτής της κατάστασης, εφαρμόζονται μηχανισμοί **timeout, retry με exponential backoff και circuit breaker patterns**, που εξασφαλίζουν ότι η αναμονή δεν θα διαρκέσει υπερβολικά και ότι η αποτυχία μιας εξωτερικής υπηρεσίας δεν θα μπλοκάρει ολόκληρο το σύστημα.

Στο **έκτο και τελευταίο βήμα**, αφού η επιχειρησιακή λογική ολοκληρώσει την επεξεργασία, το service επιστρέφει το αποτέλεσμα στον route handler, ο οποίος συνθέτει την **HTTP απόκριση**. Η απόκριση περιλαμβάνει τον κατάλληλο HTTP status code (200 για επιτυχία, 201 για δημιουργία πόρου, 204 για επιτυχία χωρίς περιεχόμενο, 4xx για σφάλματα πελάτη, 5xx για σφάλματα server), τα κατάλληλα response headers και το JSON payload με τα δεδομένα ή τα μηνύματα σφάλματος. Η απόκριση επιστρέφεται μέσω του αντίστροφου της ίδιας διαδρομής μέχρι το αρχικό frontend, όπου επεξεργάζεται και προβάλλεται κατάλληλα στον χρήστη.

Σημαντικό αρχιτεκτονικό χαρακτηριστικό της σύγχρονης επικοινωνίας του REFASHION είναι η **σαφής διάκριση των scopes** των endpoints. Όπως αναλύθηκε σε προηγούμενες ενότητες, το API οργανώνεται σε τρία διακριτά namespace: το **/store/\*** που εξυπηρετεί τους αγοραστές με publishable API key authentication, το **/vendor/\*** που εξυπηρετεί τους πωλητές με JWT-based seller authentication, και το **/admin/\*** που εξυπηρετεί τους διαχειριστές με ισχυρότερους ελέγχους ασφαλείας. Κάθε scope έχει τους δικούς του κανόνες αυθεντικοποίησης, εξουσιοδότησης και rate limiting, εξασφαλίζοντας ότι κανένας χρήστης δεν μπορεί να αποκτήσει πρόσβαση σε λειτουργίες που δεν τον αφορούν, ακόμη και σε περίπτωση προσπάθειας απευθείας κλήσης endpoints παρακάμπτοντας τη διεπαφή χρήστη.

Η **κατηγοριοποίηση των σύγχρονων αλληλεπιδράσεων** της πλατφόρμας εμπίπτει σε τέσσερις γενικές ομάδες, καθεμία από τις οποίες παρουσιάζει χαρακτηριστικές απαιτήσεις. Οι **read operations** (αναζήτηση προϊόντων, ανάκτηση καταλόγου, εμφάνιση λεπτομερειών παραγγελίας) είναι οι πιο συχνές και απαιτούν εξαιρετικά χαμηλούς χρόνους απόκρισης. Για την επίτευξη αυτού του στόχου, οι σχετικές διαδρομές αξιοποιούν εκτεταμένα **caching strategies** σε πολλαπλά επίπεδα - από edge cache στο CDN, μέσω

application-level cache στο backend, έως query-level cache στη βάση δεδομένων. Οι **write operations** (δημιουργία παραγγελίας, ενημέρωση προφίλ, υποβολή νέου προϊόντος) είναι λιγότερο συχνές αλλά πιο σύνθετες, εμπλέκοντας database transactions και ενεργοποίηση events. Οι **search operations** εξυπηρετούνται από την εξειδικευμένη μηχανή αναζήτησης που προαναφέρθηκε, με χρόνους απόκρισης συνήθως κάτω των 100ms ακόμη και για σύνθετα queries. Τέλος, οι **integration operations** (πληρωμές, AI επεξεργασία) έχουν εγγενώς υψηλότερους χρόνους απόκρισης λόγω της εμπλοκής εξωτερικών συστημάτων και αντιμετωπίζονται με κατάλληλα timeouts και feedback προς τον χρήστη.

Η σύγχρονη επικοινωνία, παρά τα οφέλη της σε όρους απλότητας και άμεσης ανάδρασης, παρουσιάζει και **εγγενείς περιορισμούς**: δεν είναι κατάλληλη για χρονοβόρες λειτουργίες, δεν επιτρέπει την παράλληλη εκτέλεση πολλαπλών δευτερευουσών ενεργειών, και δημιουργεί ισχυρή σύζευξη μεταξύ του αρχικού αιτήματος και των συνεπαγόμενων ενεργειών. Για την αντιμετώπιση αυτών των περιορισμών, η αρχιτεκτονική του REFASHION συμπληρώνεται από έναν **ισχυρό μηχανισμό ασύγχρονης επικοινωνίας**, ο οποίος παρουσιάζεται στην επόμενη υποπαράγραφο.

### 3.3.7.2 Ασύγχρονη Επικοινωνία (Event-Driven)

Η **ασύγχρονη επικοινωνία** συμπληρώνει το σύγχρονο μοτίβο και αποτελεί τη **δεύτερη βασική νοοτροπία** που διέπει την αρχιτεκτονική του REFASHION. Σε αντίθεση με το σύγχρονο μοτίβο όπου ο αιτών αναμένει την ολοκλήρωση κάθε ενέργειας πριν συνεχίσει, στην ασύγχρονη επικοινωνία ο **παραγωγός (producer)** μιας πληροφορίας ή ενός γεγονότος απλώς το **δημοσιεύει** σε έναν κεντρικό μηχανισμό διανομής, χωρίς να γνωρίζει - ή να ενδιαφέρεται - ποιοι θα είναι οι **καταναλωτές (consumers)** αυτής της πληροφορίας. Η εκτέλεση των δευτερευουσών ενεργειών συμβαίνει **παράλληλα και ανεξάρτητα** από την αρχική αλληλεπίδραση, επιτρέποντας τη γρήγορη απόκριση στο χρήστη ενώ ταυτόχρονα ενεργοποιείται μια **καταρράκτη ενεργειών** στο παρασκήνιο.

Στην αρχιτεκτονική του REFASHION, η ασύγχρονη επικοινωνία υλοποιείται μέσω ενός **Event-Driven μοτίβου** που εφαρμόζεται σε τρία διαφορετικά επίπεδα. Το **πρώτο επίπεδο** αποτελεί το **εσωτερικό Event Bus** του Medusa/Mercur, που λειτουργεί μέσα στον ίδιο τον πυρήνα του backend και διασυνδέει τα διάφορα Services μεταξύ τους. Το **δεύτερο επίπεδο** εμπλέκει **εξωτερικά events** από τρίτα συστήματα, όπως τα Stripe webhooks και τα blockchain events, που εισέρχονται στην πλατφόρμα ως ασύγχρονα μηνύματα και πυροδοτούν αντίστοιχες αντιδράσεις. Το **τρίτο επίπεδο** αξιοποιεί **ασύγχρονες ροές μηνυμάτων (message queues)** για την επικοινωνία με τα AI microservices και τα καταναμημένα συστήματα επεξεργασίας. Καθένα από αυτά τα τρία επίπεδα εφαρμόζει το ίδιο θεμελιώδες μοτίβο, αλλά εξυπηρετεί διαφορετικές κατηγορίες αναγκών.

Στο **πρώτο επίπεδο**, ο πυρήνας του Medusa διαθέτει έναν ενσωματωμένο **Event Bus**, που αποτελεί τη **σπονδυλική στήλη** της ασύγχρονης επικοινωνίας εντός του backend. Ο μηχανισμός αυτός υλοποιείται μέσω του **publish-subscribe pattern (pub/sub)**: τα Services δημοσιεύουν events όταν συμβαίνει κάτι αξιοσημείωτο στο σύστημα, και ένας ή περισσότεροι **Subscribers** που έχουν εγγραφεί στο συγκεκριμένο event ενεργοποιούνται για να εκτελέσουν τις αντίστοιχες ενέργειες. Η αρχιτεκτονική αυτή επιτυγχάνει **απόλυτη αποσύζευξη** μεταξύ του παραγωγού και των καταναλωτών - το Service που δημιουργεί μια παραγγελία δεν γνωρίζει ποιοι Subscribers θα ενεργοποιηθούν από το event `order.placed`, ενώ νέοι Subscribers μπορούν να προστεθούν χωρίς καμία αλλαγή στον κώδικα που παράγει το event.

Τα events στο REFASHION ακολουθούν μια **τυποποιημένη σχεδίαση**. Κάθε event έχει ένα **όνομα** που ακολουθεί τη σύμβαση `entity.action` (π.χ. `order.placed`, `customer.created`,

product.updated, payment.captured), και ένα **payload** που περιέχει τα δεδομένα που είναι απαραίτητα για την επεξεργασία από τους Subscribers. Συνήθως το payload περιλαμβάνει τους αναγνωριστικούς αριθμούς των εμπλεκόμενων οντοτήτων (όχι ολόκληρα τα δεδομένα τους), ώστε οι Subscribers να ανακτούν την **τρέχουσα κατάσταση** των οντοτήτων από τη βάση όταν εκτελούνται, αποφεύγοντας την ύπαρξη **stale data** στα events. Η σχεδίαση αυτή είναι κρίσιμη διότι μεταξύ της δημοσίευσης ενός event και της εκτέλεσης ενός Subscriber μπορεί να μεσολαβήσει σημαντικός χρόνος, οπότε τα δεδομένα ενδέχεται να έχουν αλλάξει στο μεταξύ.

Η **εκτέλεση των Subscribers** πραγματοποιείται **εκτός του κρίσιμου μονοπατιού** του αρχικού αιτήματος. Όταν ένα Service δημοσιεύει ένα event, αυτό **τοποθετείται σε μια εσωτερική ουρά** και ο έλεγχος επιστρέφει αμέσως στο Service, το οποίο μπορεί να ολοκληρώσει την ίδια του τη λειτουργία και να επιστρέψει την απόκριση στον αρχικό αιτούντα. Ξεχωριστά **worker processes** καταναλώνουν τα events από την ουρά και ενεργοποιούν τους εγγεγραμμένους Subscribers, χωρίς να επηρεάζουν την ταχύτητα απόκρισης του βασικού API. Ο μηχανισμός αυτός εξασφαλίζει ότι ακόμη και αν ένας Subscriber χρειάζεται δευτερόλεπτα ή λεπτά για να ολοκληρώσει την εργασία του (π.χ. καταγραφή στο blockchain, επεξεργασία εικόνας από AI), ο τελικός χρήστης δεν αντιλαμβάνεται καμία καθυστέρηση.

Το πιο **εμβληματικό παράδειγμα** αξιοποίησης του Event Bus στο REFASHION είναι η ροή που ενεργοποιείται κατά την **ολοκλήρωση μιας παραγγελίας**. Μόλις ο αγοραστής ολοκληρώσει επιτυχώς την πληρωμή, το OrderService δημιουργεί την παραγγελία στη βάση και δημοσιεύει το event **order.placed**. Σε αυτό το event είναι εγγεγραμμένοι **πολλαπλοί Subscribers**, καθένας από τους οποίους αναλαμβάνει διαφορετική παράλληλη εργασία. Ο **BlockchainSubscriber** ενεργοποιεί την καταγραφή της συναλλαγής στο smart contract και την έκδοση των ReWear Tokens. Ο **EcoImpactSubscriber** ενημερώνει τα συγκεντρωτικά περιβαλλοντικά στατιστικά του χρήστη και της πλατφόρμας. Ο **GamificationSubscriber** προσθέτει πόντους στο προφίλ του χρήστη, ελέγχει αν πληρούνται οι προϋποθέσεις για απονομή νέου badge, και ενημερώνει το επίπεδο (level) του χρήστη. Ο **NotificationSubscriber** εκπέμπει την ειδοποίηση επιβεβαίωσης παραγγελίας μέσω email προς τον αγοραστή και ειδοποίηση νέας πώλησης προς τον πωλητή. Ο **InventorySubscriber** ενημερώνει τα stock levels και πυροδοτεί τη ροή fulfillment. Ο **SearchIndexSubscriber** ενημερώνει το ευρετήριο αναζήτησης ώστε το προϊόν να μην εμφανίζεται πλέον σε αναζητήσεις (καθώς είναι μοναδικό και έχει πωληθεί).

Όλες αυτές οι ενέργειες εκτελούνται **παράλληλα και αυτόνομα**. Κάθε Subscriber έχει τον δικό του κύκλο ζωής, τους δικούς του μηχανισμούς error handling και retry, και τη δική του απομονωμένη επιχειρησιακή λογική. Αν για παράδειγμα το blockchain δίκτυο είναι προσωρινά μη διαθέσιμο, ο BlockchainSubscriber θα επιχειρήσει επαναλαμβανόμενα να ολοκληρώσει την καταγραφή με exponential backoff, χωρίς να επηρεάσει την επιτυχή εκτέλεση των υπόλοιπων Subscribers - η παραγγελία θα έχει ήδη επιβεβαιωθεί στον χρήστη, οι ειδοποιήσεις θα έχουν αποσταλεί, και η blockchain καταγραφή θα ολοκληρωθεί ευκαιριακά όταν το δίκτυο επανέλθει.

Στο **δεύτερο επίπεδο** της ασύγχρονης επικοινωνίας, η αρχιτεκτονική του REFASHION χειρίζεται **εξωτερικά events** από τρίτα συστήματα. Το πιο χαρακτηριστικό παράδειγμα είναι τα **webhooks του Stripe**: όταν συμβαίνει κάτι σημαντικό στην πλευρά του Stripe - επιτυχής ολοκλήρωση πληρωμής, αποτυχία χρέωσης, dispute, refund, αλλαγή κατάστασης connected account - η Stripe **αποστέλλει αυτόνομα ένα HTTP POST request** σε εξειδικευμένο webhook endpoint του backend. Η πλατφόρμα δεν χρειάζεται να **κάνει**

**polling** για να ελέγξει την κατάσταση της συναλλαγής στο Stripe· αντίθετα, ενημερώνεται **proactively** μόλις συμβεί η αλλαγή.

Ο χειρισμός των webhooks ακολουθεί ένα τυποποιημένο μοτίβο. Πρώτα, εκτελείται **επαλήθευση της αυθεντικότητας** του εισερχόμενου webhook μέσω signature verification - η Stripe υπογράφει κρυπτογραφικά κάθε webhook με ένα shared secret, και το backend ελέγχει την υπογραφή πριν εμπιστευτεί το περιεχόμενο. Αυτό αποτρέπει την πλαστογράφηση webhooks από κακόβουλους τρίτους. Στη συνέχεια, το webhook μετατρέπεται σε **εσωτερικό event** στο Event Bus της πλατφόρμας (π.χ. ένα Stripe `payment_intent.succeeded` webhook πυροδοτεί ένα εσωτερικό event `payment.captured`), και η περαιτέρω επεξεργασία ακολουθεί την κανονική ροή των εσωτερικών events με τους αντίστοιχους Subscribers. Η αρχιτεκτονική αυτή εξασφαλίζει την **ενιαία αντιμετώπιση** όλων των ασύγχρονων events στο σύστημα, ανεξάρτητα από την προέλευσή τους.

Παρόμοιο μοτίβο εφαρμόζεται και για τα **blockchain events**. Όταν μια συναλλαγή που έχει υποβληθεί στο Ethereum δίκτυο τελικά **επιβεβαιώνεται** και ενσωματώνεται σε ένα block, ή όταν συμβαίνει κάποιο event που εκπέμπεται από το smart contract του REFASHION, ένας εξειδικευμένος **BlockchainListener** εντοπίζει το γεγονός μέσω polling ή event subscriptions και το μεταφράζει σε εσωτερικά events της πλατφόρμας. Έτσι, η πλατφόρμα μπορεί να αντιδράσει σε γεγονότα της αλυσίδας με την ίδια λογική όπως σε εσωτερικά της events.

Στο **τρίτο επίπεδο** της ασύγχρονης επικοινωνίας, η αρχιτεκτονική αξιοποιεί **message queues** για τη διαχείριση εξειδικευμένων ροών μεταξύ του πυρήνα και των microservices ή καταναμημένων συστημάτων επεξεργασίας. Παράδειγμα τέτοιας ροής είναι η διασύνδεση με το **Visual Product Intelligence Service**: όταν ένας πωλητής ανεβάζει μια εικόνα προϊόντος, το αίτημα προς το microservice δεν περιμένει συγχρονισμένα την επιστροφή των αποτελεσμάτων - αντίθετα, η εικόνα **τοποθετείται σε μια ουρά επεξεργασίας**, και το microservice την επεξεργάζεται όταν φτάσει η σειρά της. Όταν η επεξεργασία ολοκληρωθεί, τα αποτελέσματα επιστρέφονται στο backend μέσω callback ή μέσω αντίστροφης ουράς, και ένα εσωτερικό event ενεργοποιείται για να ενημερωθεί το frontend (μέσω push notification, real-time websocket update ή απλώς ενημέρωση κατάστασης στη βάση).

Η προσέγγιση αυτή είναι κρίσιμη για τη **σταθερότητα του συστήματος** σε σενάρια υψηλού φόρτου. Σε περιόδους μαζικών καταχωρήσεων προϊόντων - π.χ. όταν ένα μεγάλο vintage store ανεβάζει εκατοντάδες αντικείμενα σε σύντομο χρονικό διάστημα - το AI microservice δεν αναγκάζεται να εξυπηρετήσει όλα τα αιτήματα ταυτόχρονα, με κίνδυνο υπερφόρτωσης. Αντί αυτού, τα αιτήματα **σταθεροποιούνται στην ουρά** και επεξεργάζονται με σταθερό ρυθμό σύμφωνα με τη χωρητικότητα του microservice. Σε περίπτωση αύξησης του φόρτου, η αρχιτεκτονική επιτρέπει την **οριζόντια κλιμάκωση** του microservice (προσθήκη περισσότερων instances), τα οποία καταναλώνουν παράλληλα από την κοινή ουρά, αυξάνοντας τη συνολική απόδοση χωρίς αλλαγές στον κώδικα.

Η ασύγχρονη επικοινωνία υλοποιεί επίσης **scheduled jobs και background tasks** που εκτελούνται περιοδικά στο σύστημα. Παραδείγματα τέτοιων εργασιών είναι ο νυχτερινός υπολογισμός των συγκεντρωτικών στατιστικών της πλατφόρμας, η ωριαία ενημέρωση του ευρετηρίου του Recommendation Service, ο καθαρισμός προσωρινών αρχείων και sessions, και η αποστολή επετειακών email στους χρήστες. Όλες αυτές οι εργασίες ενορχηστρώνονται από έναν **scheduler** που τις τοποθετεί στις κατάλληλες ουρές την προβλεπόμενη στιγμή, ενώ τα ίδια worker processes που χειρίζονται τα events ασχολούνται και με την εκτέλεση τους.

Σημαντική πτυχή της ασύγχρονης αρχιτεκτονικής αποτελεί η **διαχείριση σφαλμάτων και η ανθεκτικότητα**. Όταν ένας Subscriber αποτυγχάνει να ολοκληρώσει την εργασία του (λόγω εξωτερικής αστοχίας, network issue ή απρόβλεπτου σφάλματος), εφαρμόζεται **retry policy με exponential backoff**: η εργασία επανατοποθετείται στην ουρά με αυξανόμενες καθυστερήσεις (π.χ. 1 λεπτό, 5 λεπτά, 30 λεπτά), δίνοντας χρόνο στις εξωτερικές υπηρεσίες να ανακάμψουν. Μετά από προκαθορισμένο αριθμό αποτυχιών, η εργασία μεταφέρεται σε μια ξεχωριστή **dead letter queue (DLQ)** όπου διατηρείται για χειροκίνητη εξέταση από τους διαχειριστές, χωρίς να μπλοκάρει την επεξεργασία των επόμενων events. Η αρχιτεκτονική αυτή εξασφαλίζει ότι **καμία ενέργεια δεν χάνεται οριστικά** χωρίς ανθρώπινη γνώση, ακόμη και σε ακραίες περιπτώσεις αστοχίας.

Συνολικά, η ασύγχρονη επικοινωνία αποτελεί τον **μηχανισμό που μετατρέπει το REFASHION από μια απλή εφαρμογή e-commerce σε ένα πραγματικά καταναμημένο και επεκτάσιμο οικοσύστημα**. Επιτρέπει τη συνύπαρξη δεκάδων υποσυστημάτων χωρίς ισχυρή σύζευξη μεταξύ τους, υποστηρίζει την προσθήκη νέων λειτουργιών χωρίς παρέμβαση στις υπάρχουσες, εξυπηρετεί υψηλούς φόρτους χωρίς υποβάθμιση της εμπειρίας χρήστη, και προσφέρει το πλαίσιο εντός του οποίου οι καινοτόμες λειτουργίες της πλατφόρμας - blockchain, gamification, AI - μπορούν να ενσωματωθούν με τρόπο **οργανικό και βιώσιμο**.

### 3.3.7.3 Παράδειγμα End-to-End Ροής

Για να γίνει **απλή και κατανοητή** η συνδυαστική λειτουργία των αρχιτεκτονικών αρχών που αναλύθηκαν στις προηγούμενες υποπαραγράφους, η παρούσα ενότητα παρουσιάζει μια **ολοκληρωμένη ροή από άκρο σε άκρο** που διασχίζει το σύνολο των επιπέδων της πλατφόρμας. Επιλέγεται ως case study η **ολοκλήρωση μιας παραγγελίας από έναν αγοραστή**, διότι αποτελεί τη ροή με **τη μεγαλύτερη πολυπλοκότητα** και ενεργοποιεί ταυτόχρονα όλα τα συστατικά της αρχιτεκτονικής - frontend, API, επιχειρησιακή λογική, persistence, εξωτερικές υπηρεσίες και αμφοτέρους τους μηχανισμούς επικοινωνίας (σύγχρονη και ασύγχρονη). Η αναλυτική παρακολούθηση της ροής αυτής δεν αποτελεί απλώς επεξηγηματικό παράδειγμα, αλλά **αποκαλύπτει τη δυναμική του οικοσυστήματος** και τις λεπτές αλληλεπιδράσεις που εξασφαλίζουν την ομαλή του λειτουργία.

**Σενάριο.** Ένας αγοραστής που έχει ήδη συμπληρώσει το καλάθι του με ένα μεταχειρισμένο φόρεμα από μια πωλήτρια-ιδιώτη, πατάει το κουμπί "Ολοκλήρωση Αγοράς" στο Web Storefront. Από αυτό το σημείο και μέχρι ο αγοραστής να λάβει την επιβεβαίωση της παραγγελίας του και η πωλήτρια να ενημερωθεί για τη νέα της πώληση, **εκτελείται μια αλληλουχία δεκάδων ενεργειών** που εμπλέκουν σχεδόν όλα τα υποσυστήματα της πλατφόρμας. Η ροή χωρίζεται σε **έξι διακριτές φάσεις**, καθεμία από τις οποίες αναλύεται παρακάτω.

#### Φάση Α: Προετοιμασία της Πληρωμής (Σύγχρονη Ροή)

Όταν ο αγοραστής επιλέγει τη μέθοδο πληρωμής με κάρτα και πατάει το κουμπί επιβεβαίωσης, το Web Storefront εκκινεί ένα σύγχρονο HTTP POST αίτημα προς το endpoint `/store/carts/{cart_id}/payment-sessions`, συμπεριλαμβάνοντας στο payload τον επιθυμητό πάροχο πληρωμής (Stripe). Το αίτημα διασχίζει το επίπεδο middleware του backend, όπου επικυρώνεται το publishable API key, εντοπίζεται η τρέχουσα συνεδρία του αγοραστή και εξακριβώνεται ότι το cart\_id ανήκει πραγματικά σε αυτόν. Στη συνέχεια, το αίτημα παραδίδεται στο **CartService**, το οποίο επιβεβαιώνει ότι το καλάθι περιέχει διαθέσιμα προϊόντα (έλεγχος αποθέματος στο InventoryService), υπολογίζει το τελικό κόστος εφαρμόζοντας τους φόρους που αντιστοιχούν στην περιοχή του αγοραστή και ενεργοποιεί τον αντίστοιχο **PaymentProcessorService**.

Ο `PaymentProcessorService`, μέσω του Stripe Connect plugin, πραγματοποιεί μια σύγχρονη κλήση προς το API της Stripe για τη **δημιουργία ενός Payment Intent**. Η κλήση αυτή περιλαμβάνει το συνολικό ποσό, το νόμισμα, την αναγνωριστική ταυτότητα της πωλήτριας στο Stripe (Stripe Account ID) και τα ποσοστά κατανομής που υπαγορεύονται από τους `commission rules`. Η Stripe δημιουργεί τον Payment Intent και επιστρέφει ένα **client\_secret**, το οποίο διοχετεύεται μέσω της απόκρισης του backend πίσω στο Web Storefront. Ολόκληρη η Φάση Α ολοκληρώνεται **σύγχρονα**, σε χρόνο της τάξης των 300-800ms, και ο αγοραστής βλέπει αμέσως τη φόρμα εισαγωγής στοιχείων κάρτας.

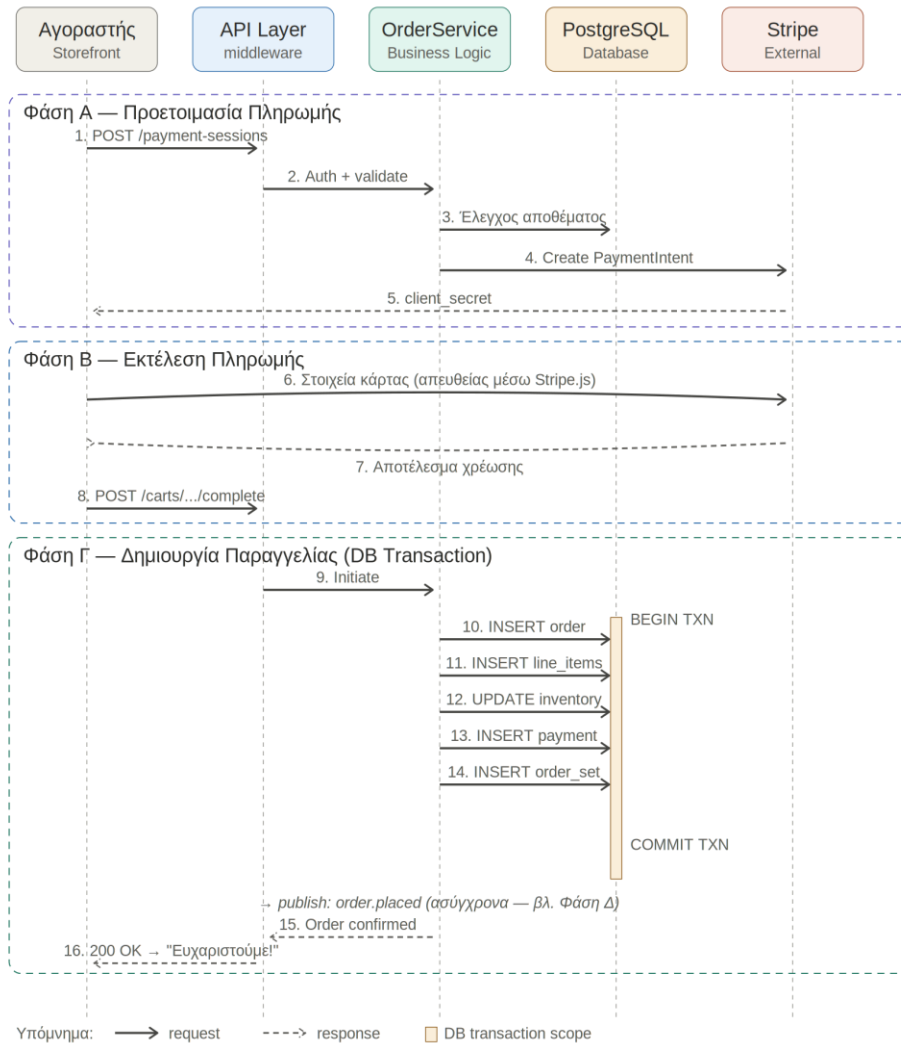
### Φάση Β: Εκτέλεση της Πληρωμής (Σύγχρονη με Εξωτερική Διαμεσολάβηση)

Ο αγοραστής εισάγει τα στοιχεία της κάρτας του στη φόρμα του Stripe.js SDK, η οποία είναι ενσωματωμένη στο storefront αλλά **εκτελείται απευθείας στον browser** χωρίς να διέρχονται τα ευαίσθητα δεδομένα από τους servers της πλατφόρμας. Όταν ο αγοραστής επιβεβαιώνει την πληρωμή, το Stripe.js αποστέλλει τα στοιχεία απευθείας στη Stripe χρησιμοποιώντας το `client_secret`, και η Stripe επιστρέφει το αποτέλεσμα της προσπάθειας χρέωσης. Εφόσον η χρέωση είναι επιτυχής, το frontend ενημερώνει το backend με ένα νέο σύγχρονο αίτημα προς το endpoint `/store/carts/{cart_id}/complete`. Αυτό αποτελεί το **σημείο εκκίνησης της πραγματικής δημιουργίας της παραγγελίας** στο σύστημα.

### Φάση Γ: Δημιουργία της Παραγγελίας (Σύγχρονη Ροή με Transaction)

Το αίτημα ολοκλήρωσης παραδίδεται στο **OrderService**, το οποίο εκκινεί ένα database transaction στην PostgreSQL για να εξασφαλίσει την ατομικότητα όλων των αλλαγών που πρόκειται να συμβούν. Εντός του transaction, το service εκτελεί τα ακόλουθα βήματα: δημιουργεί τη **βασική εγγραφή Order** στον αντίστοιχο πίνακα, αντιγράφει τα line items του καλαθιού στον πίνακα `order_line_item`, αποθηκεύει τη διεύθυνση αποστολής, καταχωρεί το payment record συσχετισμένο με τον Stripe Payment Intent ID, ενημερώνει το απόθεμα του προϊόντος ώστε να εμφανίζεται ως πωλημένο (καθώς πρόκειται για μοναδικό αντικείμενο μεταχειρισμένου είδους, το `inventory_quantity` πέφτει στο 0), και - επειδή το REFASHION λειτουργεί ως marketplace - δημιουργεί ένα **OrderSet** με την παραγγελία του πωλητή, ενορχηστρώνοντας τη μελλοντική κατανομή της πληρωμής. Όλες αυτές οι αλλαγές γίνονται **εντός του ίδιου transaction**, εξασφαλίζοντας ότι σε περίπτωση οποιασδήποτε αποτυχίας θα γίνει rollback και η παραγγελία δεν θα δημιουργηθεί σε μερική κατάσταση.

Σύγχρονη ροή ολοκλήρωσης παραγγελίας



Μόλις το transaction ολοκληρωθεί με επιτυχία, το OrderService εκτελεί δύο τελευταίες ενέργειες πριν επιστρέψει: **δημοσιεύει στο Event Bus το event `order.placed`** μαζί με τα κύρια αναγνωριστικά της παραγγελίας, και επιστρέφει στον route handler την επιβεβαίωση. Ο route handler συνθέτει την HTTP απόκριση επιτυχίας, η οποία επιστρέφει στο Web Storefront, και ο αγοραστής βλέπει την οθόνη "Ευχαριστούμε για την αγορά σας". Από τη σκοπιά του αγοραστή, η ροή έχει **ολοκληρωθεί**. Στην πραγματικότητα όμως, **μόλις τώρα ξεκινάει η πιο σύνθετη και ενδιαφέρουσα φάση** της αρχιτεκτονικής.

**Φάση Δ: Ασύγχρονος Καταρράκτης (Event-Driven)**

Στο παρασκήνιο, το event `order.placed` που δημοσιεύτηκε στο Event Bus ενεργοποιεί **παράλληλα και ανεξάρτητα** όλους τους εγγεγραμμένους Subscribers, καθένας από τους οποίους αναλαμβάνει διαφορετική παράλληλη εργασία. Το σύστημα δεν περιμένει την ολοκλήρωση κανενός από αυτούς πριν θεωρήσει την παραγγελία επιτυχημένη - αντιθέτως, καθένας εκτελεί τη δική του απομονωμένη λογική με τους δικούς του μηχανισμούς error handling και retry.

Ο **BlockchainSubscriber** ενεργοποιείται και καλεί τον BlockchainService του refashion-blockchain-plugin. Ο service ετοιμάζει τη συναλλαγή για το smart contract, υπογράφει

την κλήση με το ιδιωτικό κλειδί της πλατφόρμας και την υποβάλει στο Ethereum δίκτυο μέσω της βιβλιοθήκης ethers.js. Η συναλλαγή αυτή εκτελεί δύο ενέργειες ατομικά: **καταγράφει τα στοιχεία της παραγγελίας** στην αλυσίδα (transaction ID, διευθύνσεις αγοραστή και πωλήτριας, αναγνωριστικό προϊόντος, χρονοσφραγίδα) και **εκδίδει (mints) τα αντίστοιχα ReWear Tokens** προς τα ψηφιακά πορτοφόλια των δύο εμπλεκόμενων χρηστών. Η blockchain συναλλαγή θα χρειαστεί χρόνο για να επιβεβαιωθεί (από δευτερόλεπτα έως λεπτά), αλλά αυτό συμβαίνει **εκτός του κρίσιμου μονοπατιού** - ο αγοραστής έχει ήδη λάβει την επιβεβαίωσή του.

Ταυτόχρονα και παράλληλα, ο **EcoImpactSubscriber** ενεργοποιείται για να ενημερώσει τα περιβαλλοντικά στατιστικά. Ανακτά τον Eco-Impact Score του προϊόντος που έχει υπολογιστεί από την αντίστοιχη υπηρεσία (εξοικονόμηση CO<sub>2</sub>, νερού, αποφυγή αποβλήτων), και τα προσθέτει τόσο στο προσωπικό προφίλ του αγοραστή όσο και στα συγκεντρωτικά στατιστικά της πλατφόρμας που εμφανίζονται στο Admin Panel. Έτσι ο αγοραστής, όταν επόμενη φορά εισέλθει στο Stats Module της εφαρμογής κινητού, θα δει τους ανανεωμένους δείκτες της προσωπικής του περιβαλλοντικής συνεισφοράς.

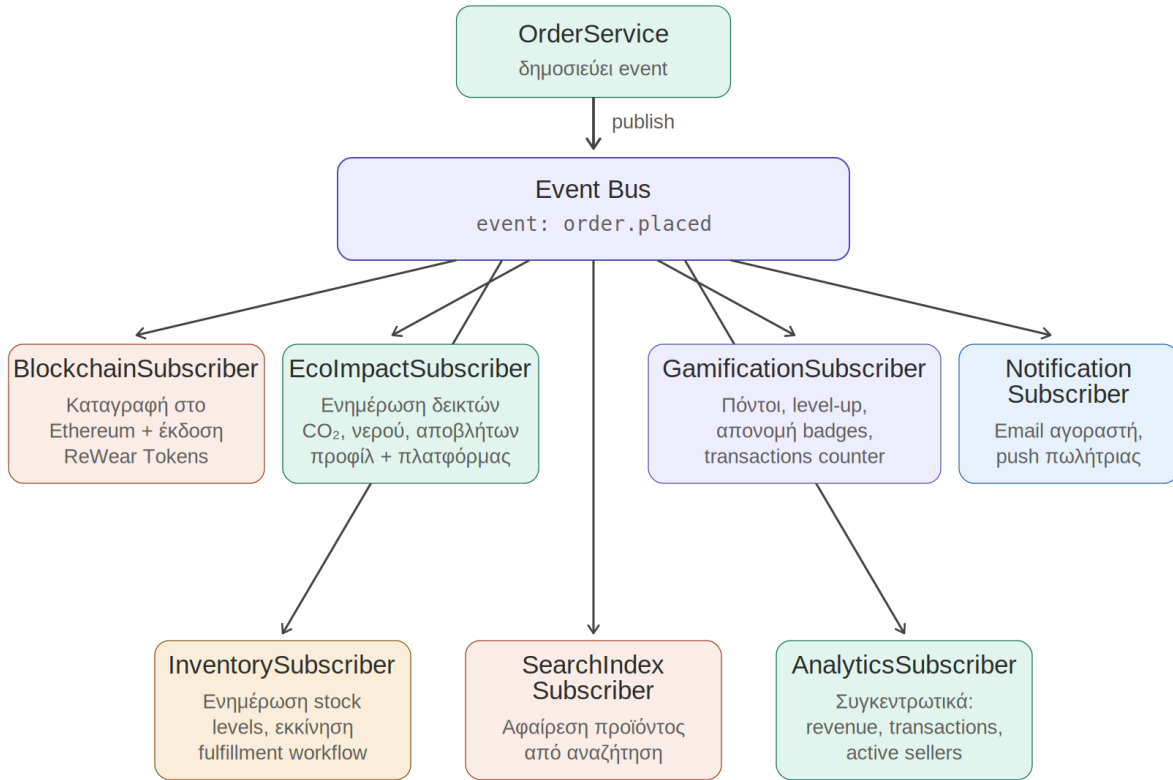
Παράλληλα, ο **GamificationSubscriber** ενημερώνει τα στοιχεία παιχνιδοποίησης. Προσθέτει τους αντίστοιχους πόντους στο **gamification\_profile** του αγοραστή και της πωλήτριας, αυξάνει τους μετρητές **completed\_transactions** της πωλήτριας, ελέγχει αν πληρούνται οι προϋποθέσεις για **απονομή νέου badge** (π.χ. "Εμπειρη Πωλήτρια" μετά από 10 πωλήσεις, "Σώτεια του Πλανήτη" όταν συσσωρευτεί συγκεκριμένη ποσότητα εξοικονόμησης CO<sub>2</sub>), και ενημερώνει το επίπεδο (level) των χρηστών αν συγκεντρώθηκαν αρκετοί πόντοι. Όπου εντοπίζεται απονομή badge ή level-up, ο subscriber δημοσιεύει ένα νέο event **badge.awarded** ή **level.upgraded**, το οποίο με τη σειρά του ενεργοποιεί άλλους Subscribers (π.χ. ειδοποίηση).

Ο **NotificationSubscriber** αναλαμβάνει την επικοινωνία με τους χρήστες. Στέλνει μέσω της υπηρεσίας email προς τον αγοραστή την επιβεβαίωση παραγγελίας με τα στοιχεία αποστολής, και μέσω push notification (στην εφαρμογή κινητού) προς την πωλήτρια την ενημέρωση "Νέα πώληση!" με σύνδεσμο για τη διαχείριση της παραγγελίας στο Seller Panel. Ο **SearchIndexSubscriber** ενημερώνει το ευρετήριο της μηχανής αναζήτησης, αφαιρώντας το προϊόν από τα δημόσια αποτελέσματα ώστε να μην εμφανίζεται πλέον σε άλλους αγοραστές, και προσθέτει σχετικές μετρικές πωλήσεων που θα αξιοποιηθούν από το Recommendation Service για μελλοντικές προτάσεις.

Τέλος, ένας **AnalyticsSubscriber** ενημερώνει τα συγκεντρωτικά στατιστικά της πλατφόρμας - total revenue, total transactions, active sellers - που χρησιμοποιούνται για το Impact Overview του Admin Panel και για τις περιοδικές αναφορές προς τους χρηματοδότες και τους ενδιαφερόμενους φορείς.

**Ασύγχρονη ροή: Event-driven fan-out (Φάση Δ)**

Παράλληλη ενεργοποίηση Subscribers από το event order.placed



**Παράλληλη και αποσυζευγμένη εκτέλεση**  
 Κάθε Subscriber εκτελείται ανεξάρτητα, με δικά του retry policies και error handling.  
 Η αποτυχία ενός Subscriber δεν επηρεάζει τους υπόλοιπους ούτε την επιτυχία της παραγγελίας.  
 Ο αγοραστής έχει ήδη λάβει την επιβεβαίωσή του από τη Φάση Γ.

Νέα events που πιθανώς πυροδοτούνται:

- blockchain.confirmed (όταν επιβεβαιωθεί η αλυσίδα)
- badge.awarded (αν πληρούνται προϋποθέσεις)
- payment.captured (από Stripe webhook)

Υπόμνημα: ■ Event Bus ■ Internal services ■ External integrations  Επόμενα events (cascade)

**Φάση Ε: Ολοκλήρωση Εξωτερικών Συναλλαγών (Ασύγχρονα με Webhooks)**

Παράλληλα με την εκτέλεση των Subscribers, οι **εξωτερικές συναλλαγές** που έχουν εκκινηθεί συνεχίζουν την πορεία τους και θα ολοκληρωθούν σε διαφορετικές χρονικές κλίμακες. Η Stripe, μετά την επιτυχή χρέωση της κάρτας, εκτελεί τις δικές της εσωτερικές διαδικασίες settlement και - όταν αυτές ολοκληρωθούν - αποστέλλει ένα **webhook payment\_intent.succeeded** στο εξειδικευμένο webhook endpoint του backend. Το backend επαληθεύει την υπογραφή του webhook, ανανεώνει την κατάσταση του payment record στη βάση και δημοσιεύει το εσωτερικό event **payment.captured**. Νέοι Subscribers ενεργοποιούνται από αυτό το event - ένας ενημερώνει τον πωλητή ότι τα χρήματα του είναι πλέον σε διαδικασία payout, ένας άλλος ενεργοποιεί τη ροή προετοιμασίας του fulfillment.

Αντίστοιχα, όταν η συναλλαγή blockchain που εκκίνησε ο BlockchainSubscriber **επιβεβαιωθεί τελικά στο Ethereum** (μετά από αρκετά block confirmations), ο **BlockchainListener** εντοπίζει την επιβεβαίωση και δημοσιεύει το εσωτερικό event **blockchain.confirmed**. Συναφείς Subscribers ενεργοποιούνται για να ενημερώσουν την κατάσταση της εγγραφής **blockchain\_transaction** στη βάση από "pending" σε "confirmed", να ενημερώσουν τα balances των ReWear Tokens στα προφίλ των χρηστών (από "pending" σε "available"), και να αποστείλουν ειδοποίηση προς τους εμπλεκόμενους ότι τα tokens τους είναι πλέον διαθέσιμα για χρήση.

### Φάση ΣΤ: Παρακολούθηση από τον Πωλητή και Συνέχεια της Ροής

Η πωλήτρια, μέσω της push notification που έλαβε, ανοίγει την εφαρμογή κινητού και βλέπει τη νέα της παραγγελία στο Seller Panel. Η εφαρμογή εκτελεί ένα σύγχρονο αίτημα **GET /vendor/orders** προς το backend, το οποίο επιστρέφει τη λίστα των παραγγελιών της πωλήτριας (φιλτραρισμένη μέσω των ownership checks του Mercur). Η πωλήτρια επιλέγει την παραγγελία, βλέπει τα στοιχεία αποστολής και εκκινεί τη διαδικασία fulfillment. Η ροή που ακολουθεί - προετοιμασία του αντικειμένου, εκτύπωση ετικέτας αποστολής μέσω του fulfillment provider, ενημέρωση κατάστασης σε "shipped", λήψη tracking number - αποτελεί **μια νέα αλληλουχία** σύγχρονων και ασύγχρονων αλληλεπιδράσεων που εμπλέκει ξανά πολλαπλά υποσυστήματα.

### Συνολική Παρατήρηση

Η ροή που μόλις παρουσιάστηκε αποκαλύπτει μερικά **κρίσιμα χαρακτηριστικά** της αρχιτεκτονικής του REFASHION. Πρώτον, η **σαφής διαστρωμάτωση επιπέδων** εξασφαλίζει ότι κάθε επίπεδο εστιάζει στις δικές του ευθύνες: το frontend στην παρουσίαση, το API στην αυθεντικοποίηση και το routing, η επιχειρησιακή λογική στους κανόνες, το persistence στην αποθήκευση, και τα εξωτερικά συστήματα στις εξειδικευμένες λειτουργίες. Δεύτερον, ο **συνδυασμός σύγχρονης και ασύγχρονης επικοινωνίας** επιτρέπει στην πλατφόρμα να παρέχει **άμεση απόκριση** στους χρήστες (η παραγγελία επιβεβαιώνεται σε δευτερόλεπτα) ενώ ταυτόχρονα εκτελεί δεκάδες δευτερεύουσες ενέργειες στο παρασκήνιο χωρίς να επηρεάζεται η εμπειρία τους. Τρίτον, η **απομονωμένη φύση των Subscribers** εξασφαλίζει την **ανθεκτικότητα του συστήματος**: η αποτυχία οποιασδήποτε δευτερεύουσας λειτουργίας (π.χ. προσωρινή μη διαθεσιμότητα του blockchain) δεν επηρεάζει την επιτυχία της κύριας συναλλαγής. Τέταρτον, η **εκτεταμένη χρήση events** ως μηχανισμού επικοινωνίας μεταξύ υποσυστημάτων επιτρέπει την προσθήκη νέων λειτουργιών (π.χ. ένας νέος αναλυτικός μηχανισμός, μια νέα κανονιστική αναφορά) **χωρίς καμία τροποποίηση** στον υπάρχοντα κώδικα της παραγγελίας - αρκεί η ανάπτυξη ενός νέου Subscriber που θα ακούει το ίδιο event.

Συνολικά, η ροή αυτή αποτελεί την **έμπρακτη απόδειξη** ότι οι αρχιτεκτονικές αρχές που υιοθετήθηκαν για το REFASHION - headless, composable, event-driven - δεν αποτελούν αφηρημένες θεωρητικές επιλογές, αλλά **υλοποιούνται σε κάθε καθημερινή λειτουργία** της πλατφόρμας με τρόπο που επιτρέπει τη συνύπαρξη πολλαπλών εξειδικευμένων υποσυστημάτων κάτω από μια ενιαία και συνεκτική εμπειρία χρήστη.

### 3.4. Μοντέλο Δεδομένων

Στις προηγούμενες ενότητες παρουσιάστηκε η αρχιτεκτονική φιλοσοφία και η οργανωτική δομή της πλατφόρμας REFASHION, καθώς και οι τρόποι με τους οποίους τα διαφορετικά επίπεδα συνεργάζονται για να εξυπηρετήσουν τις λειτουργικές της ανάγκες. Η ενότητα που ακολουθεί εστιάζει σε ένα **εξίσου θεμελιώδες ζήτημα**: το ίδιο το **μοντέλο δεδομένων** πάνω στο οποίο οικοδομείται το σύνολο της επιχειρησιακής λογικής. Ένα καλά σχεδιασμένο μοντέλο δεδομένων αποτελεί τη **σπονδυλική στήλη** οποιουδήποτε επιχειρησιακού

συστήματος, καθώς καθορίζει τις πληροφορίες που μπορούν να αποθηκευτούν, τις σχέσεις που μπορούν να εκφραστούν και τις λειτουργίες που μπορούν να υλοποιηθούν.

Στο REFASHION, το μοντέλο δεδομένων προκύπτει από τη **συμπληρωματική αλληλεπίδραση τριών συνιστώσων**. Η πρώτη συνιστώσα είναι το **σύνολο των βασικών οντοτήτων του MedusaJS και του Mercur**, το οποίο αποτελεί ένα ώριμο και πλήρες μοντέλο ηλεκτρονικού εμπορίου που καλύπτει τις απαιτήσεις ενός σύγχρονου marketplace πολλαπλών πωλητών. Η δεύτερη συνιστώσα είναι οι **custom οντότητες** που έχουν προστεθεί ειδικά για τις ανάγκες του REFASHION και υλοποιούν τις καινοτόμες λειτουργίες της πλατφόρμας - δωρεές, ανταλλαγές, gamification, blockchain καταγραφές. Η τρίτη συνιστώσα είναι ο **μηχανισμός των metadata** του MedusaJS, που επιτρέπει την ευέλικτη επέκταση των υφιστάμενων οντοτήτων με custom πεδία χωρίς αλλαγή του βασικού σχήματος.

Οι οντότητες που ακολουθούν αναλύουν αυτές τις τρεις συνιστώσες, ξεκινώντας από τις βασικές οντότητες που κληρονομούνται από τον πυρήνα MedusaJS/Mercur.

### 3.4.1 Βασικές Οντότητες Medusa/Mercur

Ο πυρήνας του MedusaJS και η επέκταση Mercur παρέχουν ένα **ολοκληρωμένο και θεμελιωμένο μοντέλο δεδομένων** που καλύπτει το σύνολο των κλασικών λειτουργιών ενός marketplace πολλαπλών πωλητών. Η αξιοποίηση αυτού του έτοιμου μοντέλου αποτελεί **κρίσιμη στρατηγική επιλογή** για το REFASHION: αντί η ομάδα ανάπτυξης να σχεδιάσει εκ νέου τις βασικές δομές δεδομένων του e-commerce - εργασία που θα απαιτούσε μήνες σχεδιασμού, υλοποίησης και δοκιμών - η πλατφόρμα κληρονομεί άμεσα ένα **δοκιμασμένο και βιομηχανικά τεκμηριωμένο μοντέλο**, εστιάζοντας τους πόρους της στις πραγματικά καινοτόμες πτυχές του έργου.

Η αρχιτεκτονική του μοντέλου δεδομένων ακολουθεί τη φιλοσοφία της **οντοκεντρικής σχεδίασης**, όπου κάθε σημαντική έννοια του συστήματος αντιστοιχίζεται σε μια διακριτή οντότητα (entity) με σαφώς ορισμένα πεδία και σχέσεις προς άλλες οντότητες. Στις παραγράφους που ακολουθούν αναλύονται οι **πιο κρίσιμες** από αυτές τις οντότητες, ομαδοποιημένες σε τέσσερις λειτουργικές περιοχές: **κατάλογος προϊόντων, πελάτες και πωλητές, παραγγελίες και πληρωμές, και οργανωτικές οντότητες**.

#### Κατάλογος Προϊόντων

Η κεντρική οντότητα του καταλόγου είναι το **Product**, που αντιπροσωπεύει ένα προϊόν προς πώληση στην πλατφόρμα. Στο REFASHION, κάθε **μεταχειρισμένο αντικείμενο μόδας** που καταχωρείται από έναν πωλητή αντιστοιχεί σε μία εγγραφή Product. Η οντότητα αυτή περιλαμβάνει τα βασικά πεδία περιγραφής (τίτλος, περιγραφή, υλικό, χαρακτηριστικά εικόνων), καθώς και πεδία διαχείρισης της κατάστασης (draft, proposed, published, rejected) που υποστηρίζουν τη ροή moderation. Σημαντικό χαρακτηριστικό είναι η ύπαρξη του πεδίου **metadata** τύπου JSONB, που λειτουργεί ως ευέλικτος container για τα custom πεδία του REFASHION (Eco-Impact Score, εξοικονόμηση CO<sub>2</sub> και νερού, αναγνωριστικά blockchain καταγραφών). Η εκτενής ανάλυση του μηχανισμού metadata παρατίθεται στην υποπαράγραφο 2.3.3.

Το **ProductVariant** αντιπροσωπεύει μια συγκεκριμένη παραλλαγή ενός Product. Στο τυπικό σενάριο ηλεκτρονικού εμπορίου, ένα προϊόν μπορεί να έχει πολλαπλές παραλλαγές - για παράδειγμα ένα φόρεμα διαθέσιμο σε διαφορετικά μεγέθη και χρώματα, καθένα ως ξεχωριστή variant με το δικό της SKU και inventory\_quantity. Στο REFASHION ωστόσο, λόγω της φύσης του ως marketplace μεταχειρισμένων ειδών, εφαρμόζεται κατά κανόνα το **μοντέλο "ενός variant"**: κάθε προϊόν είναι **μοναδικό** (ένα συγκεκριμένο μεταχειρισμένο φόρεμα, σε συγκεκριμένη κατάσταση, σε συγκεκριμένο μέγεθος) και διαθέτει μία και μόνο

ProductVariant με **inventory\_quantity**: 1. Όταν αυτό το αντικείμενο πωληθεί, η ποσότητα γίνεται 0 και το προϊόν δεν μπορεί να ξαναπωληθεί.

Η οντότητα **ProductCollection** χρησιμοποιείται για τη λειτουργική κατηγοριοποίηση των προϊόντων (π.χ. "Φορέματα", "Παπούτσια", "Αξεσουάρ"), λειτουργώντας ως κύριος μηχανισμός φιλτραρίσματος και πλοήγησης στον κατάλογο. Συμπληρωματικά, η οντότητα **ProductTag** προσφέρει έναν πιο ευέλικτο μηχανισμό ετικετοποίησης, επιτρέποντας τη συσχέτιση ενός προϊόντος με πολλαπλά χαρακτηριστικά παράλληλα (π.χ. εμπορικά σήματα, στυλ, εποχές, υλικά). Η συνδυαστική χρήση των δύο μηχανισμών - ιεραρχικές κατηγορίες μέσω Collections και ευέλικτες ετικέτες μέσω Tags - προσφέρει στην πλατφόρμα τη δυνατότητα **πολυδιάστατης οργάνωσης** του καταλόγου.

### Πελάτες και Πωλητές

Η οντότητα **Customer** αντιπροσωπεύει έναν εγγεγραμμένο χρήστη της πλατφόρμας. Περιέχει τα προσωπικά στοιχεία επικοινωνίας, τις προτιμήσεις του χρήστη και τις σχέσεις του προς άλλες οντότητες (διευθύνσεις, παραγγελίες, καλάθια). Σημαντικό χαρακτηριστικό είναι η σύνδεση κάθε Customer με ένα **CustomerGroup**, που στο REFASHION αξιοποιείται για τη **διάκριση μεταξύ ιδιωτών και επιχειρήσεων** μέσω των ομάδων **group\_individuals** και **group\_businesses** αντίστοιχα. Η κατηγοριοποίηση αυτή ενεργοποιείται αυτόματα μέσω ενός Subscriber που αντιδρά στο event **customer.created** και αναθέτει τον νέο χρήστη στην κατάλληλη ομάδα βάσει των δεδομένων εγγραφής του.

Η οντότητα **CustomerAddress** φιλοξενεί τις πολλαπλές διευθύνσεις που μπορεί να διατηρεί ένας πελάτης - διευθύνσεις χρέωσης, διευθύνσεις αποστολής, σπίτι, εργασία. Κάθε διεύθυνση συνδέεται με ένα συγκεκριμένο Customer μέσω foreign key, και μία εξ αυτών μπορεί να σημανθεί ως **προεπιλεγμένη** για κάθε τύπο χρήσης.

Η οντότητα **Seller**, που εισάγεται από την επέκταση Mercur, αποτελεί τη **διαφοροποιητική προσθήκη** του multi-vendor μοντέλου. Σε αντίθεση με τους Customers που είναι αγοραστές, οι Sellers αντιπροσωπεύουν τους **εμπορικούς συμμετέχοντες** της πλατφόρμας - ιδιώτες πωλητές, second-hand shops, vintage stores. Κάθε Seller διαθέτει το δικό του επιχειρησιακό προφίλ (επωνυμία, περιγραφή, λογότυπο), τη δική του κατάσταση (active, suspended, pending approval), και τις δικές του οικονομικές συνδέσεις (Stripe Account ID για payouts, ποσοστά προμηθειών). Η οντότητα συνδέεται με όλα τα δικά της προϊόντα και τις σχετικές παραγγελίες μέσω αυστηρών ownership relationships που επιβάλλονται σε επίπεδο service και API.

### Παραγγελίες και Πληρωμές

Η οντότητα **Cart** αντιπροσωπεύει ένα **καλάθι αγορών** ενός χρήστη πριν την οριστικοποίηση της παραγγελίας. Λειτουργεί ως **προσωρινός container** που συγκεντρώνει τα επιλεγμένα προϊόντα, τη διεύθυνση αποστολής, τη μέθοδο αποστολής και τη μέθοδο πληρωμής, χωρίς να δεσμεύει ακόμη το απόθεμα ή να ενεργοποιεί οικονομικές συναλλαγές. Συνδέεται με την οντότητα **LineItem** που αντιπροσωπεύει κάθε προϊόν εντός του καλαθιού, καθώς και με τις οντότητες **ShippingMethod** και **PaymentSession** που καταγράφουν τις σχετικές επιλογές.

Όταν ο χρήστης ολοκληρώνει την αγορά, το Cart μετατρέπεται σε **Order** - μια οριστικοποιημένη παραγγελία με συγκεκριμένα στοιχεία τιμολόγησης, διευθύνσεων και πληρωμής. Η Order συνδέεται με αντίστοιχα **OrderLineItem** εγγραφές, με την οντότητα **Payment** που καταγράφει τις λεπτομέρειες πληρωμής (συμπεριλαμβανομένου του Stripe Payment Intent ID), και με την οντότητα **Fulfillment** που παρακολουθεί την πορεία αποστολής.

Στο **marketplace context** του REFASHION, η ολοκλήρωση μιας παραγγελίας που περιλαμβάνει προϊόντα από **πολλούς διαφορετικούς πωλητές** δεν παράγει μία ενιαία Order, αλλά ένα **OrderSet**: μια συναθροιστική εγγραφή που ομαδοποιεί πολλαπλές παραγγελίες, μία ανά πωλητή. Η οντότητα αυτή, που εισάγεται από το Mercur, λειτουργεί ως **κοινός αναγνωριστικός κόμβος** για τον αγοραστή (που βλέπει μια ενιαία αγοραστική εμπειρία), ενώ ταυτόχρονα κάθε επιμέρους Order αντιπροσωπεύει την υποχρέωση ενός συγκεκριμένου Seller προς τον αγοραστή. Με τον τρόπο αυτό, το ίδιο σύστημα διατηρεί τη **συνεκτικότητα της αγοραστικής εμπειρίας** και την **αυτονομία διαχείρισης** κάθε πωλητή για τη δική του παραγγελία.

### Οργανωτικές Οντότητες

Πέρα από τις βασικές οντότητες, το μοντέλο δεδομένων περιλαμβάνει μια σειρά από **οργανωτικές οντότητες** που υποστηρίζουν τη συνολική λειτουργία της πλατφόρμας. Η οντότητα **Region** αντιπροσωπεύει μια γεωγραφική περιοχή λειτουργίας (π.χ. Ελλάδα, Ευρωπαϊκή Ένωση) και ορίζει τα χαρακτηριστικά της αγοράς σε αυτή την περιοχή: τα διαθέσιμα νομίσματα (Currency), τους ισχύοντες φορολογικούς συντελεστές (TaxRate), τις διαθέσιμες χώρες (Country) και τις μεθόδους αποστολής που υποστηρίζονται. Η συσχέτιση κάθε Cart και Order με μια Region εξασφαλίζει την **εφαρμογή των κατάλληλων κανόνων τιμολόγησης** ανάλογα με τη γεωγραφική θέση του αγοραστή.

Η οντότητα **ShippingOption** ορίζει μια διαθέσιμη μέθοδο αποστολής (π.χ. ταχυμεταφορά, κανονική αποστολή, παραλαβή από κατάστημα), συνδεδεμένη με συγκεκριμένη Region και συγκεκριμένο **FulfillmentProvider** που αναλαμβάνει την πραγματική εκτέλεση της αποστολής. Αντίστοιχα, η οντότητα **PaymentProvider** καταγράφει τους διαθέσιμους παρόχους πληρωμής (στο REFASHION πρωτίστως το Stripe Connect), με τις σχετικές παραμέτρους ρύθμισης.

Τέλος, η οντότητα **CommissionRule**, που εισάγεται από το Mercur, ορίζει τους κανόνες προμήθειας που εφαρμόζονται στις συναλλαγές του marketplace. Μια CommissionRule μπορεί να αφορά συγκεκριμένο Seller (π.χ. προωθητική περίοδος χωρίς προμήθεια για νέους πωλητές), συγκεκριμένη κατηγορία προϊόντων (π.χ. μειωμένη προμήθεια για προϊόντα υψηλού Eco-Impact Score), ή να ισχύει καθολικά. Η ευελιξία αυτή επιτρέπει στους διαχειριστές της πλατφόρμας να **διαμορφώνουν δυναμικά τη στρατηγική τιμολόγησης** του marketplace χωρίς παρέμβαση στον κώδικα.

### Σχέσεις και Ακεραιότητα

Το σύνολο των παραπάνω οντοτήτων συγκροτεί ένα **πυκνό δίκτυο σχέσεων** που εκφράζεται μέσω foreign keys και που εξασφαλίζει την **αναφορική ακεραιότητα** του συστήματος. Ένα Product ανήκει σε έναν Seller · πολλά ProductVariants συσχετίζονται με ένα Product · ένα Cart περιέχει πολλαπλά LineItems και ανήκει σε έναν Customer · μια Order παράγεται από ένα Cart και συνδέεται με ένα OrderSet στο marketplace context. Η PostgreSQL επιβάλλει την ακεραιότητα αυτών των σχέσεων σε επίπεδο βάσης δεδομένων, ενώ τα services της επιχειρησιακής λογικής επιβάλλουν επιπλέον τους **επιχειρησιακούς κανόνες** που δεν μπορούν να εκφραστούν με απλά foreign keys (π.χ. ότι ένας Seller βλέπει μόνο τα δικά του Products και Orders).

Συνολικά, οι βασικές οντότητες που κληρονομεί το REFASHION από το MedusaJS και το Mercur αποτελούν ένα **πλούσιο και αυτοτελές μοντέλο δεδομένων** που καλύπτει τις απαιτήσεις ενός marketplace πολλαπλών πωλητών χωρίς να χρειάζεται καμία ριζική επέκταση. Οι ειδικότερες ανάγκες του REFASHION - δωρεές, ανταλλαγές, gamification, blockchain - δεν αλλοιώνουν αυτές τις θεμελιώδεις οντότητες, αλλά υλοποιούνται μέσω **συμπληρωματικών custom οντοτήτων** που παρουσιάζονται στην επόμενη υποπαράγραφο.

### 3.4.2 Custom Οντότητες REFASHION

Πέρα από τις βασικές οντότητες που κληρονομούνται από τον πυρήνα MedusaJS και την επέκταση Mercur, η αρχιτεκτονική του REFASHION εισάγει ένα **σύνολο εξειδικευμένων οντοτήτων** που υλοποιούν τις καινοτόμες λειτουργίες της πλατφόρμας. Οι οντότητες αυτές δεν αποτελούν τροποποίηση ή αντικατάσταση των υφισταμένων, αλλά **συμπληρωματικές προσθήκες** που συνυπάρχουν αρμονικά με τις βασικές οντότητες, αξιοποιώντας την επεκτασιμότητα του μοντέλου δεδομένων του MedusaJS. Η σχεδιαστική αυτή προσέγγιση εξασφαλίζει ότι η κωδικοβάση του REFASHION παραμένει **αναβαθμίσιμη** σε σχέση με το υποκείμενο framework, καθώς οι core οντότητες διατηρούνται άθικτες και οι custom προσθήκες απομονώνονται σε ξεχωριστά plugins.

Από αρχιτεκτονική σκοπιά, οι custom οντότητες του REFASHION υλοποιούνται μέσω του **migration mechanism** του MedusaJS, που αποτυπώνει κάθε νέο πίνακα σε ένα versioned migration file. Με τον τρόπο αυτό, η εξέλιξη του custom schema παραμένει **παρακολουθήσιμη, αναπαραγωγίμη και αντιστρέψιμη**, ενώ ενσωματώνεται φυσικά στη ροή deployment της πλατφόρμας. Παράλληλα, κάθε νέα οντότητα ορίζεται μέσω εξειδικευμένων TypeScript model classes στο αντίστοιχο REFASHION plugin, εκθέτοντας τον τύπο της στο σύστημα Dependency Injection του Medusa και επιτρέποντας την αυτόματη ενσωμάτωσή της στα services και τα API endpoints.

Οι custom οντότητες του REFASHION ομαδοποιούνται σε **τέσσερις λειτουργικές περιοχές**, καθεμία από τις οποίες αντιστοιχίζεται σε μία από τις καινοτόμες δυνατότητες της πλατφόρμας: τις **εναλλακτικές μορφές συναλλαγής** (δωρεές, ανταλλαγές), τη **διαχείριση των ReWear Tokens και της blockchain διασύνδεσης**, τους **μηχανισμούς gamification**, και τις **υπηρεσίες Eco-Impact**.

#### Εναλλακτικές Μορφές Συναλλαγής

Η οντότητα **SwapRequest** αποτελεί την κύρια προσθήκη για την υποστήριξη του μηχανισμού ανταλλαγών μεταξύ χρηστών. Στο REFASHION, η ανταλλαγή αποτελεί μια εναλλακτική μορφή κυκλικής συναλλαγής όπου δύο χρήστες συμφωνούν να **ανταλλάξουν αντικείμενα** χωρίς οικονομική συναλλαγή, αξιοποιώντας τη φιλοσοφία της επαναχρησιμοποίησης σε ένα peer-to-peer περιβάλλον. Η οντότητα SwapRequest καταγράφει την κατάσταση μιας τέτοιας πρότασης ανταλλαγής, με τα παρακάτω βασικά πεδία: **status** (pending, accepted, rejected, completed, cancelled), **initiator\_id** που αναφέρεται στον χρήστη που εκκίνησε την πρόταση, **receiver\_id** που αναφέρεται στον αποδέκτη της πρότασης, **initiator\_product\_id** και **receiver\_product\_id** που προσδιορίζουν τα δύο εμπλεκόμενα προϊόντα, καθώς και πεδία χρονοσφραγίδων (created\_at, accepted\_at, completed\_at) που επιτρέπουν την πλήρη παρακολούθηση του κύκλου ζωής της πρότασης.

Η σχεδίαση της SwapRequest ως αυτόνομης οντότητας - και όχι ως ειδική περίπτωση μιας Order - αντικατοπτρίζει την **εννοιολογική διαφορά** μιας ανταλλαγής από μια αγορά. Όταν η πρόταση γίνει αποδεκτή, παράγονται **δύο αμοιβαίες Orders μηδενικού κόστους** που υλοποιούν την πραγματική μεταφορά των προϊόντων μεταξύ των χρηστών, ωστόσο η ίδια η SwapRequest παραμένει ως **αναγνωριστικός κόμβος** της συνολικής συναλλαγής, επιτρέποντας στους χρήστες να δουν το ιστορικό των ανταλλαγών τους ως διακριτές οντότητες, ξεχωριστά από τις αγορές τους.

Σημειώνεται ότι το REFASHION υποστηρίζει επίσης τη λειτουργία των **δωρεών**, η οποία ωστόσο **δεν απαιτεί ξεχωριστή οντότητα** - υλοποιείται απευθείας ως μια τυπική Order με συνολικό κόστος μηδέν, εξυπηρετούμενη από έναν custom Payment Provider που εγκρίνει αυτόματα τέτοιες παραγγελίες. Η επιλογή αυτή κρατάει το μοντέλο δεδομένων

**απλούστερο**, καθώς οι δωρεές διαφέρουν από τις αγορές μόνο στο οικονομικό σκέλος και όχι στη βασική τους δομή.

### Διαχείριση ReWear Tokens και Blockchain

Η οντότητα **BlockchainTransaction** καταγράφει κάθε αλληλεπίδραση της πλατφόρμας με το Ethereum δίκτυο, αποτελώντας τη **γέφυρα μεταξύ της εσωτερικής επιχειρησιακής λογικής και της αμετάβλητης on-chain πραγματικότητας**. Κάθε εγγραφή στην οντότητα αυτή αντιστοιχεί σε μία υποβληθείσα blockchain συναλλαγή και περιλαμβάνει τα ακόλουθα βασικά πεδία: **transaction\_hash** (το μοναδικό αναγνωριστικό της συναλλαγής στο Ethereum), **transaction\_type** (order\_record, token\_mint, token\_transfer), **from\_address** και **to\_address** (οι εμπλεκόμενες διευθύνσεις πορτοφολιών), **amount** (όπου αφορά token transactions), **status** (pending, confirmed, failed), **block\_number** (μετά την επιβεβαίωση), και αναγνωριστικά συσχέτισης με τις εσωτερικές οντότητες του Medusa (order\_id, customer\_id, seller\_id όπου ισχύει).

Η σχεδίαση της BlockchainTransaction υλοποιεί μια κρίσιμη **διπλή αναφορική σχέση**: αφενός συνδέεται με τις εσωτερικές οντότητες του REFASHION (μέσω foreign keys προς Order, Customer, Seller), αφετέρου διατηρεί τις on-chain αναφορές (μέσω του transaction\_hash) που επιτρέπουν την επαλήθευση της συναλλαγής στο δημόσιο δίκτυο. Αυτή η διπλή αναφορά εξασφαλίζει ότι κάθε σημαντικό γεγονός της πλατφόρμας μπορεί να **ιχνηλατηθεί** από την εσωτερική επιχειρησιακή πραγματικότητα έως την αμετάβλητη blockchain εγγραφή του.

Η οντότητα **RewardHistory** καταγράφει το πλήρες ιστορικό των ReWear Tokens που έχουν αποδοθεί σε κάθε χρήστη της πλατφόρμας. Κάθε εγγραφή αναφέρεται σε μία πράξη ανταμοιβής και περιλαμβάνει: **customer\_id** (ο δικαιούχος), **amount** (η ποσότητα των tokens), **reward\_type** (purchase, sale, donation, swap, badge\_bonus, level\_up), **source\_entity\_type** και **source\_entity\_id** (η οντότητα που προκάλεσε την ανταμοιβή, π.χ. μια συγκεκριμένη Order), **blockchain\_transaction\_id** (συσχέτιση με την αντίστοιχη BlockchainTransaction), και **status** (pending για unconfirmed blockchain transactions, available μετά την επιβεβαίωση).

Σημαντική αρχιτεκτονική απόφαση είναι ότι η RewardHistory διατηρείται ως **on-chain mirror entity** - δηλαδή ως **εσωτερικό αντιγραφο** της πραγματικότητας που υπάρχει on-chain. Τα tokens κάθε χρήστη υπάρχουν πραγματικά στο smart contract του Ethereum, ωστόσο η εσωτερική RewardHistory επιτρέπει στην πλατφόρμα να **εμφανίζει άμεσα** τα στοιχεία ανταμοιβών στους χρήστες χωρίς να απαιτείται κάθε φορά query στο blockchain, που θα ήταν αργό και δαπανηρό. Σε περίπτωση ασυνέπειας μεταξύ εσωτερικού και on-chain μοντέλου, η on-chain πραγματικότητα αποτελεί την **τελική πηγή αλήθειας**, και η εσωτερική RewardHistory συγχρονίζεται μέσω εξειδικευμένης διαδικασίας reconciliation.

### Μηχανισμοί Gamification

Η οντότητα **GamificationProfile** αποτελεί την κύρια προσθήκη για την υποστήριξη των μηχανισμών παιχνιδοποίησης της πλατφόρμας. Κάθε εγγεγραμμένος χρήστης διαθέτει ακριβώς ένα GamificationProfile, που λειτουργεί ως **συγκεντρωτικός χάρτης** της δραστηριότητας και των επιτευγμάτων του εντός της πλατφόρμας. Τα βασικά πεδία της οντότητας περιλαμβάνουν: **customer\_id** (foreign key προς τον αντίστοιχο χρήστη), **total\_points** (συνολικοί πόντοι gamification), **current\_level** (τρέχον επίπεδο), **completed\_transactions** (μετρητής ολοκληρωμένων συναλλαγών), **completed\_donations** (μετρητής δωρεών), **completed\_swaps** (μετρητής ανταλλαγών), καθώς και πεδία **last\_activity\_at** και **profile\_created\_at** που επιτρέπουν την παρακολούθηση της εμπλοκής διαχρονικά.

Συμπληρωματικά, η ίδια οντότητα περιλαμβάνει ένα πεδίο **badges** τύπου JSONB, που αποθηκεύει το σύνολο των ανακτηθέντων badges του χρήστη ως δομημένο JSON array. Κάθε στοιχείο του array αντιπροσωπεύει ένα διακριτικό επίτευγμα (π.χ. "Πρώτη Πώληση", "Σώτεια του Πλανήτη", "Εμπειρη Πωλήτρια") με τη χρονοσφραγίδα απονομής του. Η επιλογή αποθήκευσης ως JSONB - αντί της δημιουργίας ξεχωριστής οντότητας Badge με σχέση many-to-many - αντικατοπτρίζει μια **σχεδιαστική απόφαση εμπνευσμένη από την πραγματική χρήση** των δεδομένων: τα badges ενός χρήστη ανακτώνται **πάντοτε μαζί** με το υπόλοιπο gamification profile, επομένως η ομαδοποίησή τους στην ίδια εγγραφή βελτιστοποιεί τις σχετικές queries χωρίς να υπονομεύει τη λειτουργικότητα. Η ενσωματωμένη υποστήριξη της PostgreSQL για indexing JSONB πεδίων μέσω GIN indexes εξασφαλίζει αποδοτικές αναζητήσεις όπου χρειάζονται.

Η οντότητα **GamificationRule** αποτελεί τη **διαμορφωτική συνιστώσα** του μηχανισμού παιχνιδιοποίησης και ορίζει τους κανόνες σύμφωνα με τους οποίους αποδίδονται πόντοι και badges. Κάθε κανόνας περιγράφει μια συνθήκη και μια ανταμοιβή - για παράδειγμα "10 πόντοι ανά ολοκληρωμένη πώληση", "Badge 'Εμπειρη Πωλήτρια' μετά από 10 ολοκληρωμένες πωλήσεις", "Bonus 50 πόντων όταν συμπληρωθούν 100kg εξοικονόμηση CO<sub>2</sub>". Η ύπαρξη της οντότητας αυτής στη βάση δεδομένων - αντί της σκληρής κωδικοποίησης των κανόνων στον κώδικα - επιτρέπει στους διαχειριστές της πλατφόρμας να **τροποποιούν τους μηχανισμούς ανταμοιβής δυναμικά** μέσω του admin panel, χωρίς ανάγκη επανανάπτυξης του συστήματος.

### Υπηρεσίες Eco-Impact

Η αρχιτεκτονική του REFASHION αξιοποιεί κυρίως τον **μηχανισμό metadata** των Product entities για την αποθήκευση των δεικτών περιβαλλοντικού οφέλους (Eco-Impact Score, εξοικονόμηση CO<sub>2</sub>, νερού, αποβλήτων), σύμφωνα με την προσέγγιση που αναλύεται στην επόμενη υποπαράγραφο. Ωστόσο, για την υποστήριξη των **συγκεντρωτικών στατιστικών** της πλατφόρμας, εισάγεται η custom οντότητα **EcoImpactAggregate**.

Η οντότητα EcoImpactAggregate λειτουργεί ως **materialized snapshot** των συσσωρευμένων περιβαλλοντικών δεικτών της πλατφόρμας, ενημερωμένο σε τακτική βάση μέσω scheduled jobs. Περιλαμβάνει πεδία όπως: **aggregate\_type** (global, per\_user, per\_region, per\_period), **total\_co2\_saved**, **total\_water\_saved**, **total\_waste\_avoided**, **total\_items\_recirculated**, **period\_start** και **period\_end**, καθώς και references στις σχετικές οντότητες (customer\_id, region\_id) όπου ισχύει. Η χρήση μιας ξεχωριστής οντότητας - αντί της on-the-fly υπολογισμού των στατιστικών μέσω SQL aggregations - εξυπηρετεί την **απόδοση** του Impact Overview του Admin Panel και των αντίστοιχων ενοτήτων του Stats Module της εφαρμογής κινητού, οι οποίες απαιτούν άμεση εμφάνιση συγκεντρωτικών δεδομένων ακόμη και όταν η βάση περιέχει εκατοντάδες χιλιάδες παραγγελιών.

### Συμπληρωματικές Οντότητες Υποστήριξης

Πέρα από τις κύριες custom οντότητες, το μοντέλο δεδομένων του REFASHION περιλαμβάνει ορισμένες **βοηθητικές οντότητες** που εξυπηρετούν λειτουργικές ανάγκες υψηλότερου επιπέδου. Η οντότητα **ProductRequest** καταγράφει τις προτάσεις νέων προϊόντων που υποβάλλονται από τους πωλητές, διατηρώντας την κατάστασή τους (pending, approved, rejected) και τα στοιχεία της απόφασης (admin\_id που ενέκρινε ή απέρριψε, χρονοσφραγίδα, αιτιολογία). Παρόμοια, η οντότητα **SellerRequest** καταγράφει τις προτάσεις εγγραφής νέων πωλητών, υποστηρίζοντας την κεντρική εποπτεία της εισόδου νέων μελών στο marketplace.

Συνολικά, οι custom οντότητες του REFASHION προσφέρουν την **απαραίτητη συμπληρωματική δομή** για την υλοποίηση των καινοτόμων λειτουργιών της πλατφόρμας, χωρίς να αλλοιώνουν το βασικό μοντέλο που κληρονομείται από το MedusaJS/Mercur. Σε

συνδυασμό με τη χρήση των metadata πεδίων που παρουσιάζεται στην επόμενη υποπαράγραφο, σχηματίζουν ένα **ολοκληρωμένο, ευέλικτο και επεκτάσιμο μοντέλο δεδομένων** που υποστηρίζει το σύνολο των απαιτήσεων του έργου.

### 3.4.3 Επέκταση μέσω Metadata

Πέρα από τη δημιουργία αυτόνομων custom οντοτήτων που παρουσιάστηκαν στην προηγούμενη υποπαράγραφο, η αρχιτεκτονική του REFASHION αξιοποιεί έναν **εξίσου σημαντικό αλλά λιγότερο εμφανή** μηχανισμό επέκτασης του μοντέλου δεδομένων: τη χρήση του πεδίου **metadata** που υπάρχει σε σχεδόν κάθε βασική οντότητα του MedusaJS. Ο μηχανισμός αυτός αποτελεί μία από τις **πιο ευέλικτες και ισχυρές** αρχιτεκτονικές δυνατότητες του υποκείμενου framework, και η σωστή του αξιοποίηση είναι κρίσιμη για τη σχεδιαστική ομορφιά και τη συντηρησιμότητα ολόκληρης της πλατφόρμας.

Το πεδίο **metadata** είναι ένα πεδίο τύπου **JSONB** που υπάρχει εγγενώς σε όλες τις βασικές οντότητες του MedusaJS - Product, ProductVariant, Customer, Order, Cart, Seller, Region, και άλλες. Η ύπαρξή του δεν αποτελεί σχεδιαστική τυχαιότητα του framework, αλλά **σκοπίμη αρχιτεκτονική επιλογή** που αναγνωρίζει μια θεμελιώδη πραγματικότητα: κάθε υλοποίηση e-commerce έχει τις δικές της εξειδικευμένες απαιτήσεις που δεν μπορούν να καλυφθούν από ένα γενικό σχήμα. Αντί να υποχρεώνει κάθε υλοποιητή να επεκτείνει τις βασικές οντότητες - διαδικασία που θα δημιουργούσε **τεχνικό χρέος** και θα δυσχέραινε τη μελλοντική αναβάθμιση του core framework - το MedusaJS παρέχει αυτό το ευέλικτο πεδίο που δέχεται οποιαδήποτε δομή JSON, αφήνοντας στον υλοποιητή την ελευθερία να αξιοποιήσει τον χώρο αυτόν σύμφωνα με τις δικές του ανάγκες.

#### Πλεονεκτήματα του μηχανισμού

Η αξιοποίηση των metadata fields προσφέρει στο REFASHION σημαντικά πρακτικά πλεονεκτήματα. Πρώτον, εξασφαλίζει την **ομαλή αναβαθμισιμότητα** του υποκείμενου MedusaJS framework. Καθώς δεν τροποποιούνται οι ίδιες οι βασικές οντότητες, αναβαθμίσεις του Medusa σε νεότερες εκδόσεις μπορούν να εφαρμοστούν χωρίς conflicts ή απρόσμενα προβλήματα συμβατότητας. Δεύτερον, παρέχει **εξαιρετική ευελιξία** στη φάση ανάπτυξης: η προσθήκη νέου custom πεδίου σε μια οντότητα μπορεί να γίνει **χωρίς database migration**, αφού το JSONB πεδίο δέχεται οποιαδήποτε νέα δομή ή πεδίο χωρίς αλλαγή του σχήματος της βάσης. Τρίτον, μειώνει σημαντικά την **πολυπλοκότητα του μοντέλου δεδομένων**, αποφεύγοντας τη δημιουργία πληθώρας μικρών custom πινάκων για κάθε εξειδικευμένη ανάγκη.

Παράλληλα, η ισχυρή υποστήριξη του τύπου JSONB από την PostgreSQL εξασφαλίζει ότι αυτή η ευελιξία **δεν συνοδεύεται από κόστος απόδοσης**. Σε αντίθεση με τον απλό τύπο JSON που αποθηκεύεται ως κείμενο, ο JSONB αποθηκεύεται σε **δυναμική, ευρετηριασμένη μορφή**, επιτρέποντας ταχείες αναζητήσεις πάνω σε JSON πεδία μέσω εξειδικευμένων τελεστών (->, ->>, @>, ?, κ.ά.) και μέσω **GIN indexes** που μπορούν να εφαρμοστούν στο σύνολο ή σε επιλεγμένα πεδία του JSON. Με τη σωστή ευρετηρίαση, queries που φιλτράρουν προϊόντα βάσει custom metadata πεδίων (π.χ. "βρες όλα τα προϊόντα με Eco-Impact Score άνω του 80") εκτελούνται με επιδόσεις αντίστοιχες ή και καλύτερες από queries σε παραδοσιακά scalar πεδία.

#### Σχεδιαστική Φιλοσοφία και Όρια Χρήσης

Η σχεδιαστική φιλοσοφία του REFASHION ακολουθεί μια **σαφή διάκριση** για την επιλογή μεταξύ metadata extension και custom οντότητας: όταν η εξειδικευμένη πληροφορία αφορά **ιδιότητες ή χαρακτηριστικά** μιας υπάρχουσας οντότητας - που σχεδόν πάντα ανακτώνται μαζί με την οντότητα και δεν συσχετίζονται έντονα μεταξύ τους - προτιμάται το metadata field. Όταν, αντιθέτως, η πληροφορία αφορά **διακριτή έννοια** με δικό της κύκλο

ζωής και πολύπλοκες σχέσεις προς άλλες οντότητες (όπως η SwapRequest ή η BlockchainTransaction), δημιουργείται ξεχωριστή custom οντότητα. Αυτή η διάκριση εξασφαλίζει ότι το metadata δεν μετατρέπεται σε **καταβόθρα ασυσχέτιστων δεδομένων** που υπονομεύει τη συνεκτικότητα του μοντέλου, αλλά παραμένει ένας στοχευμένος μηχανισμός επέκτασης.

Στις παραγράφους που ακολουθούν παρουσιάζονται οι **κύριες χρήσεις** του μηχανισμού metadata στο REFASHION, οργανωμένες ανά οντότητα.

### Metadata στην οντότητα Product

Η οντότητα Product αποτελεί το **πιο εκτενώς εμπλουτισμένο σημείο** του μοντέλου δεδομένων μέσω metadata. Αυτό είναι αναμενόμενο, καθώς το προϊόν αποτελεί τον κεντρικό αντικείμενο διαπραγμάτευσης της πλατφόρμας και κάθε καινοτόμος λειτουργία του REFASHION σχετίζεται με κάποιο τρόπο με αυτό. Το metadata field κάθε Product περιέχει μια **πολυδιάστατη συλλογή δεδομένων** που οργανώνεται σε λογικές κατηγορίες.

Η **πιο εμβληματική κατηγορία** αφορά τα **δεδομένα Eco-Impact**, που υπολογίζονται κατά τη δημιουργία ή ενημέρωση ενός προϊόντος από την αντίστοιχη εξωτερική υπηρεσία. Στο metadata αποθηκεύεται το **eco\_impact\_score** ως αριθμητική τιμή που εκφράζει τον περιβαλλοντικό δείκτη του προϊόντος, μαζί με τα επιμέρους στατιστικά: **water\_saved\_liters** (εξοικονόμηση νερού σε λίτρα συγκριτικά με την παραγωγή ενός αντίστοιχου νέου προϊόντος), **co2\_reduced\_kg** (μείωση εκπομπών CO<sub>2</sub> σε χιλιόγραμμα), και **waste\_avoided\_kg** (αποφυγή κλωστοϋφαντουργικών αποβλήτων). Παράλληλα, αποθηκεύονται και τα **χαρακτηριστικά υπολογισμού** που χρησιμοποιήθηκαν: ο αναγνωρισμένος τύπος υφάσματος, το εκτιμώμενο βάρος, η κατηγορία και η εκτιμώμενη ηλικία, επιτρέποντας τόσο τη μελλοντική επαναμέτρηση όσο και την επαλήθευση του υπολογισμού.

Μια δεύτερη κατηγορία αφορά τα **AI-εμπλουτισμένα δεδομένα** που παράγονται από το Visual Product Intelligence Service. Όπως αναλύθηκε στην ενότητα 2.2.5.5, κατά την καταχώρηση ενός προϊόντος, η υπηρεσία αυτή εξάγει αυτόματα δομημένα μεταδεδομένα από εξωτερικές πηγές μέσω visual search και LLM-based εξαγωγής. Τα αποτελέσματα αυτής της διαδικασίας - **original\_brand**, **original\_retail\_price**, **material\_composition** ως δομημένο object με ποσοστά υλικών, **country\_of\_origin**, **sustainability\_certifications** - αποθηκεύονται στο metadata του Product, επιτρέποντας στο σύστημα να **διατηρεί ίχνη** της AI επεξεργασίας και να εμπλουτίζει την παρουσίαση του προϊόντος με πληροφορίες που ο πωλητής δεν θα μπορούσε να συλλέξει χειροκίνητα.

Μια τρίτη κατηγορία περιλαμβάνει τα **αναγνωριστικά συσχέτισης με εξωτερικά συστήματα**. Στο metadata αποθηκεύονται αναφορές προς τις σχετικές BlockchainTransaction εγγραφές (όπου εφαρμόζεται), αναγνωριστικά της επεξεργασμένης εικόνας στο Image Analysis Service (**processed\_image\_id**), καθώς και AI-generated similarity vectors hash που μπορούν να αξιοποιηθούν από το Recommendation Service. Η ομαδοποίηση αυτών των αναφορών στο ίδιο πεδίο εξασφαλίζει ότι **όλες οι εξωτερικές διασυνδέσεις** ενός προϊόντος μπορούν να ανακτηθούν με μία και μόνο query.

### Metadata στην οντότητα Customer

Στο metadata της οντότητας Customer αποθηκεύονται κυρίως δεδομένα που σχετίζονται με τις **καινοτόμες λειτουργίες** του REFASHION και δεν περιλαμβάνονται στις τυπικές πληροφορίες προφίλ του Medusa. Η πιο σημαντική κατηγορία αφορά τη σύνδεση με τα **ψηφιακά πορτοφόλια Ethereum**: το πεδίο **wallet\_address** αποθηκεύει τη blockchain διεύθυνση του χρήστη, ενώ συμπληρωματικά πεδία όπως **wallet\_type** (custodial, user-

managed) και `wallet_created_at` παρακολουθούν την κατάσταση και την προέλευση του πορτοφολιού. Η αξιοποίηση του metadata field για τα στοιχεία αυτά - αντί της δημιουργίας ξεχωριστής οντότητας Wallet - εξυπηρετεί την **ενότητα της παρουσίασης**: όταν ζητείται ένα Customer entity, ολόκληρο το blockchain context του ανακτάται μαζί, χωρίς ανάγκη επιπλέον JOIN λειτουργιών.

Επιπρόσθετα, στο metadata του Customer αποθηκεύονται οι **προτιμήσεις εμφάνισης** του χρήστη που αφορούν τις καινοτόμες λειτουργίες της πλατφόρμας: ρυθμίσεις προβολής των ReWear Tokens, προτιμήσεις εμφάνισης του Eco-Impact, επιλογές gamification (π.χ. αν επιθυμεί να εμφανίζεται στους κοινούς πίνακες κατάταξης ή να παραμένει ανώνυμος). Η ομαδοποίηση των προτιμήσεων αυτών στο metadata εξασφαλίζει την **ενιαία διαχείριση** των ρυθμίσεων χρήστη χωρίς τον πολλαπλασιασμό μικρών scalar πεδίων στον πίνακα Customer.

### Metadata στην οντότητα Order

Το metadata της Order αξιοποιείται για την αποθήκευση **εξειδικευμένων στοιχείων κάθε παραγγελίας** που σχετίζονται με τις καινοτομίες του REFASHION. Σε αυτό αποθηκεύονται το `reward_tokens_minted` (η ποσότητα ReWear Tokens που αποδόθηκε σε κάθε εμπλεκόμενο μέρος μέσω της παραγγελίας), το `eco_impact_contribution` (η συνεισφορά της συγκεκριμένης παραγγελίας στους συνολικούς περιβαλλοντικούς δείκτες), και το `blockchain_transaction_hash` (συντόμευση αναφοράς στη BlockchainTransaction χωρίς ανάγκη JOIN). Σημαντική ειδική περίπτωση αποτελεί το πεδίο `order_type`, που διαχωρίζει τις τυπικές αγορές από τις **εναλλακτικές μορφές συναλλαγής** - δωρεές, ανταλλαγές, business-to-business - επιτρέποντας στα συστήματα reporting και στο admin panel να φιλτράρουν εύκολα τις παραγγελίες ανά είδος χωρίς ανάγκη εξειδικευμένων queries.

Σε περιπτώσεις παραγγελιών που σχετίζονται με ανταλλαγές (SwapRequest), το metadata της Order περιέχει επίσης το `swap_request_id` ως αναφορά προς την αρχική SwapRequest οντότητα, καθώς και το `paired_order_id` που αναφέρεται στην αμοιβαία παραγγελία της ίδιας ανταλλαγής. Με τον τρόπο αυτό, ο διπλός χαρακτήρας της ανταλλαγής αποτυπώνεται **αμφίδρομα** στο μοντέλο δεδομένων, διευκολύνοντας τη συνεκτική παρουσίαση των δύο μερών της συναλλαγής.

### Metadata στην οντότητα Seller

Το metadata της οντότητας Seller - που εισάγεται από το Mercur - αξιοποιείται για την αποθήκευση **στοιχείων πιστοποίησης και βιωσιμότητας** του πωλητή που είναι χαρακτηριστικά της φιλοσοφίας REFASHION. Στο πεδίο αποθηκεύονται πιστοποιήσεις βιωσιμότητας (π.χ. B Corp certification, fair trade), μέλος συνεταιριστικών οργανισμών, στοιχεία επικοινωνίας με τρίτους πιστοποιητικούς φορείς, καθώς και χαρακτηριστικά δραστηριότητας που χρησιμοποιούνται από τους αλγόριθμους προτάσεων (γεωγραφική περιοχή λειτουργίας, εξειδικευμένες κατηγορίες, μέσος όρος χρόνου αποστολής). Παράλληλα, αποθηκεύονται και αναγνωριστικά διασύνδεσης με εξωτερικά συστήματα: το `stripe_account_id` (αν και αυτό αποθηκεύεται κυρίως ως κανονικό πεδίο), αναγνωριστικά συνεργαζόμενων logistics partners, καθώς και τα blockchain wallet addresses του πωλητή για την παραλαβή των ReWear Tokens.

### Σχεδιαστικές Συμβάσεις και Καλές Πρακτικές

Η αξιοποίηση του μηχανισμού metadata στο REFASHION ακολουθεί ορισμένες **σχεδιαστικές συμβάσεις** που εξασφαλίζουν τη συνεκτικότητα και τη μακροπρόθεσμη συντηρησιμότητα του μοντέλου. Πρώτον, η **ονομασία των πεδίων** ακολουθεί συνεπή σύμβαση snake\_case, διευκολύνοντας την αναγνωσιμότητα και την αυτόματη τεκμηρίωση. Δεύτερον, οι **πιο σύνθετες δομές** οργανώνονται σε **εμφωλωμένα objects**

που ομαδοποιούν λογικά συσχετισμένα πεδία (π.χ. όλα τα Eco-Impact πεδία υπό το object `eco_impact`), αποτρέποντας τον πολλαπλασιασμό flat fields που θα δυσχέρανε την κατανόηση της δομής. Τρίτον, για κάθε metadata extension διατηρείται **τεκμηρίωση στη μορφή TypeScript interfaces** στην κωδικοβάση των plugins, εξασφαλίζοντας την τυπολογική επαλήθευση των πεδίων κατά τη φάση ανάπτυξης.

Επιπλέον, η αρχιτεκτονική του REFASHION εφαρμόζει **εξειδικευμένα GIN indexes** στα metadata fields για τα πεδία που χρησιμοποιούνται συχνά σε queries. Παραδείγματος χάριν, ένα GIN index στο πεδίο `metadata->'eco_impact'->'eco_impact_score'` της Product οντότητας επιτρέπει αποδοτικές αναζητήσεις προϊόντων βάσει του δείκτη περιβαλλοντικού οφέλους, όπως απαιτείται για το φιλτράρισμα στο storefront και για τους αλγορίθμους προτάσεων. Με τον τρόπο αυτό, η ευελιξία του JSONB συνδυάζεται με την **απαραίτητη απόδοση** χωρίς συμβιβασμούς.

### **Συμπληρωματική Φύση Metadata και Custom Οντοτήτων**

Συνολικά, ο μηχανισμός metadata και η δημιουργία custom οντοτήτων **λειτουργούν συμπληρωματικά** στην αρχιτεκτονική του REFASHION. Οι custom οντότητες (SwapRequest, BlockchainTransaction, GamificationProfile, RewardHistory, EcoImpactAggregate, ProductRequest, SellerRequest, GamificationRule) αποτυπώνουν **διακριτές έννοιες** με δικό τους κύκλο ζωής και πλούσιες σχέσεις προς άλλες οντότητες. Τα metadata fields, από την άλλη, εμπλουτίζουν τις υπάρχουσες οντότητες με **εξειδικευμένα χαρακτηριστικά** που αποτελούν αναπόσπαστο μέρος αυτών των οντοτήτων.

Η συνδυαστική αξιοποίηση των δύο μηχανισμών διαμορφώνει ένα μοντέλο δεδομένων που είναι ταυτόχρονα **πλούσιο και ευέλικτο, αυστηρά δομημένο όπου χρειάζεται και ελαστικό όπου ωφελεί, αναβαθμισμό χωρίς να θυσιάζει εξειδίκευση**. Με την αρχιτεκτονική αυτή, το REFASHION αξιοποιεί στο μέγιστο τα πλεονεκτήματα του υποκείμενου MedusaJS/Mercur framework, ενώ ταυτόχρονα ενσωματώνει με συστηματικό τρόπο τις δικές του καινοτόμες απαιτήσεις, καθιστώντας το μοντέλο δεδομένων **μια από τις πιο επιμελημένες και εκλεπτυσμένες πτυχές** της πλατφόρμας.

## 4. Βασικές Λειτουργίες E-Commerce

### 4.1. Διαχείριση Χρηστών

#### 4.1.1 Υποστήριξη Ρόλων (Ιδιώτες vs. Επιχειρήσεις)

Η αρχιτεκτονική της πλατφόρμας REFASHION προβλέπει την υποστήριξη **διακριτών ρόλων χρηστών**, επιτρέποντας την ταυτόχρονη εξυπηρέτηση ιδιωτών και επαγγελματιών. Η διαφοροποίηση βασίζεται στον διαχωρισμό μεταξύ *customer accounts* (για χρήση στο storefront) και *seller accounts* (για πρόσβαση στο εξειδικευμένο vendor panel).

Με την ολοκλήρωση της εγγραφής ενός χρήστη από το frontend, ενεργοποιείται αυτόματα ένας λογαριασμός **Customer**, παρέχοντας δυνατότητες αγορών, διαχείρισης παραγγελιών, wishlist και αξιολογήσεων. Ταυτόχρονα, δρομολογείται η δημιουργία ενός παράλληλου λογαριασμού **Seller**, διασφαλίζοντας την προοπτική μελλοντικής δραστηριοποίησης του χρήστη ως πωλητή εντός του οικοσυστήματος.

Η ενεργοποίηση του πωλητή δεν πραγματοποιείται άμεσα, αλλά παραμένει σε κατάσταση *pending* προς αξιολόγηση. Η διαδικασία **moderation** εκτελείται από τους διαχειριστές μέσω του admin panel, οι οποίοι εγκρίνουν ή απορρίπτουν το αίτημα. Μετά την οριστική έγκριση, ο χρήστης αποκτά πρόσβαση στο vendor panel για τη διαχείριση του αποθέματος, των πωλήσεων και των στοιχείων του ψηφιακού του καταστήματος.

Στην περίπτωση των εταιρικών χρηστών, το backend αξιοποιεί την παράμετρο `registration_type: business`. Αυτή η δομή επιτρέπει την καταχώρηση εκτενών **εταιρικών στοιχείων** (Επωνυμία, ΑΦΜ, ΔΟΥ, εταιρική έδρα), επιτρέποντας στο σύστημα να διαφοροποιεί πλήρως τους ιδιώτες πωλητές από τις οργανωμένες επιχειρήσεις.

Για χρήστες με διπλή ιδιότητα (customer και εγκεκριμένος seller), η σύνδεση στο storefront πυροδοτεί αυτόματα και τη διαδικασία **seller authentication**. Εφόσον ο λογαριασμός είναι ενεργός, εκδίδεται vendor token, το οποίο επιτρέπει την απρόσκοπτη μετάβαση στο περιβάλλον διαχείρισης πωλήσεων χωρίς την απαίτηση νέας εισαγωγής διαπιστευτηρίων.

Αντίστοιχα, η διαδικασία αποσύνδεσης (logout) λειτουργεί καθολικά, εκκαθαρίζοντας ταυτόχρονα το customer session και το vendor session. Με αυτόν τον τρόπο διασφαλίζεται η πλήρης έξοδος του χρήστη από όλα τα υποσυστήματα της πλατφόρμας, αποτρέποντας την παραμονή ενεργών συνεδριών σε οποιοδήποτε επίπεδο.

#### 4.1.2 Εγγραφή και Αυθεντικοποίηση

Οι διαδικασίες **εγγραφής και αυθεντικοποίησης** στο οικοσύστημα REFASHION εδράζονται στο Backend API των Medusa/Mercur, διασφαλίζοντας την **ενοποιημένη διαχείριση χρηστών** μεταξύ του storefront και του εξειδικευμένου vendor panel. Η ροή εισόδου εκκινεί αποκλειστικά από το customer storefront, όπου με την αρχική εγγραφή ενός χρήστη ως *customer*, το σύστημα δρομολογεί αυτόματα τη δημιουργία ενός παράλληλου **seller account**, ο οποίος τίθεται σε κατάσταση *pending* (εκκρεμότητα).

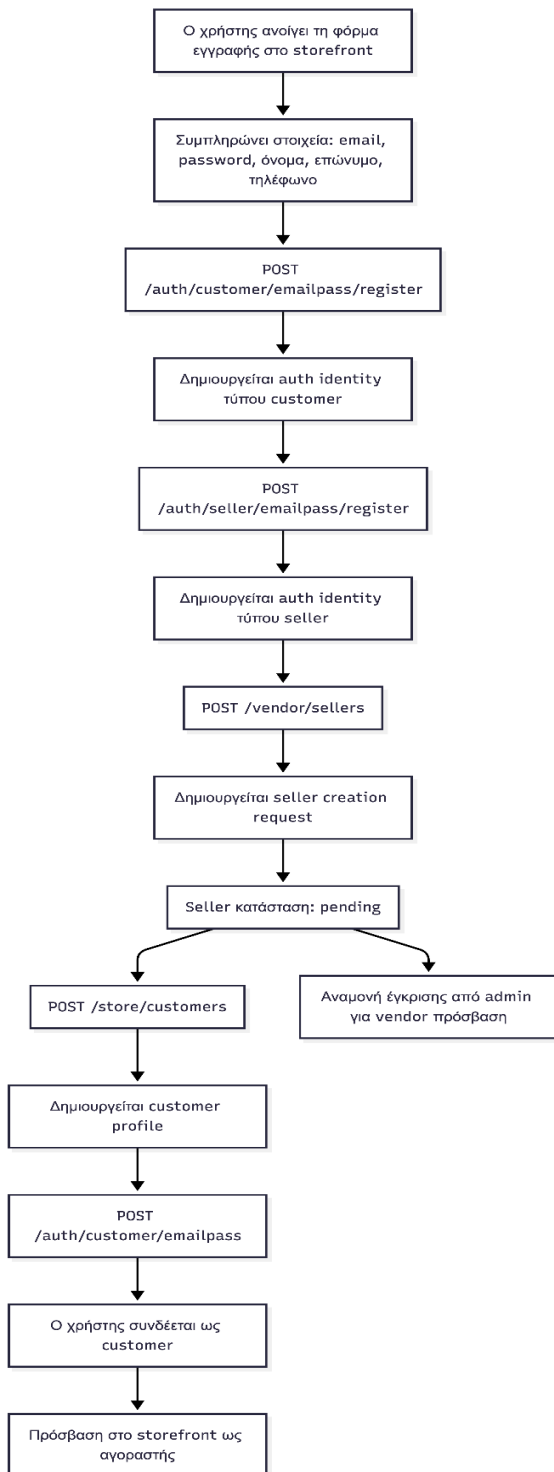
Αυτή η αρχιτεκτονική επιτρέπει σε κάθε νέο μέλος να αποκτά άμεση πρόσβαση στις λειτουργίες αγοραστή (περιήγηση, συναλλαγές, wishlist, αξιολογήσεις), ενώ ταυτόχρονα προετοιμάζει το έδαφος για τη μελλοντική του δραστηριοποίηση ως πωλητής. Ενώ το προφίλ **Customer** ενεργοποιείται ακαριαία, ο λογαριασμός **Seller** παραμένει ανενεργός για το vendor panel, απαιτώντας την οριστική **έγκριση και moderation** από τους διαχειριστές της πλατφόρμας.

Η δημιουργία της *auth identity* του πελάτη πραγματοποιείται μέσω της κλήσης:

POST /auth/customer/emailpass/register

Το συγκεκριμένο endpoint εγκαθιδρύει την ταυτότητα του χρήστη και εκδίδει το απαραίτητο token για την εκτέλεση προστατευμένων αιτημάτων. Ακολουθεί η αυτόματη παραγωγή της *auth identity* του πωλητή μέσω του:

POST /auth/seller/emailpass/register



Η ύπαρξη της *seller identity* αποτελεί προϋπόθεση για τη μελλοντική σύνδεση του λογαριασμού, δίχως όμως να παρέχει άμεση πρόσβαση στα εργαλεία διαχείρισης πωλήσεων προτού ολοκληρωθεί το **admin review**.

Στη συνέχεια, υποβάλλεται το επίσημο αίτημα δημιουργίας πωλητή (*seller creation request*) στο endpoint:

POST /vendor/sellers

Το request ενσωματώνει τα απαραίτητα στοιχεία (ονοματεπώνυμο, επικοινωνία κ.λπ.) και παραμένει σε *pending status*. Κατά τη διάρκεια αυτής της περιόδου, η χρήση του **vendor panel** για την καταχώρηση προϊόντων παραμένει αποκλεισμένη.

Παράλληλα, οριστικοποιείται το προφίλ του αγοραστή μέσω της κλήσης:

POST /store/customers

Με την ολοκλήρωση της παραπάνω διαδικασίας, ο χρήστης εκτελεί **login** ως customer μέσω του:

POST /auth/customer/emailpass

Το παραγόμενο *customer token* διατηρείται στο frontend ως **secure session**, επιτρέποντας την απρόσκοπτη αλληλεπίδραση με τις υπηρεσίες της αγοράς.

Για τη μετέπειτα διαχείριση του λογαριασμού, αξιοποιούνται τα κάτωθι endpoints:

GET /store/customers/me

Ανάκτηση των τρεχόντων στοιχείων του συνδεδεμένου μέλους.

POST /store/customers/me

Επικαιροποίηση βασικών πληροφοριών (όνομα, τηλέφωνο κ.α.).

POST /store/customers/me/addresses

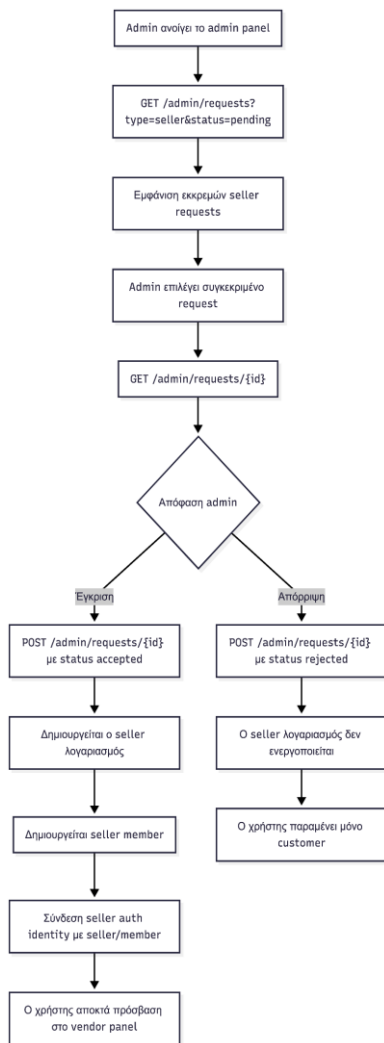
Προσθήκη νέας τοποθεσίας αποστολής στο ψηφιακό προφίλ.

POST /store/customers/me/addresses/{address\_id}

Τροποποίηση υφιστάμενης διεύθυνσης.

DELETE /store/customers/me/addresses/{address\_id}

Οριστική αφαίρεση διεύθυνσης από το αρχείο του χρήστη.



Η μετάβαση του πωλητή σε ενεργή κατάσταση αποτελεί προνόμιο του **Admin**, ο οποίος εποπτεύει τα εκκρεμή αιτήματα μέσω του κεντρικού πίνακα ελέγχου. Τα κρίσιμα endpoints διαχείρισης περιλαμβάνουν:

GET /admin/requests?type=seller&status=pending

Προβολή της λίστας με όλες τις υπό εξέταση αιτήσεις πωλητών.

GET /admin/requests/{id}

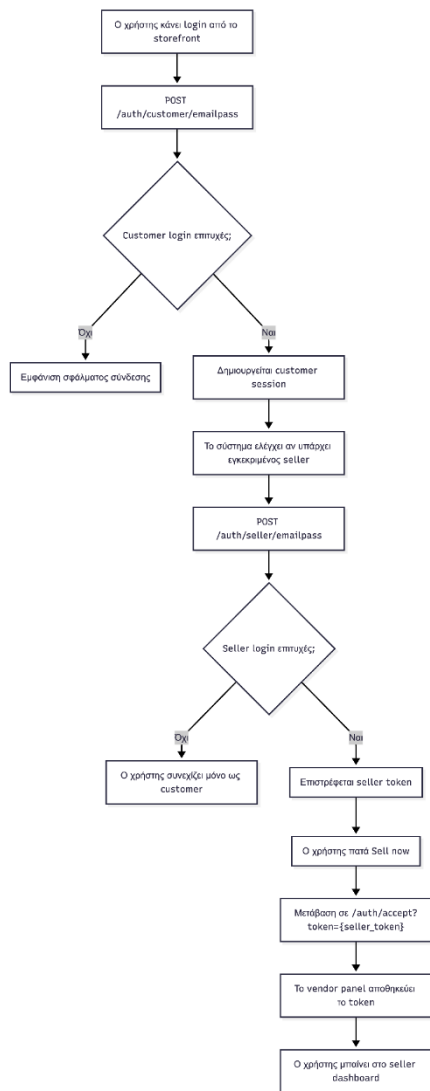
Λεπτομερής ανάλυση συγκεκριμένου αιτήματος.

POST /admin/requests/{id}

Τελική έγκριση (*accepted*) ή απόρριψη (*rejected*) της αίτησης.

Με την έγκριση, το backend οριστικοποιεί τον **seller account** και συνδέει την αρχική ταυτότητα με τον νέο ρόλο, επιτρέποντας πλέον την είσοδο στο περιβάλλον διαχείρισης.

Η αυθεντικοποίηση εκκινεί από το storefront login. Μόλις το σύστημα αναγνωρίσει τον χρήστη μέσω του:

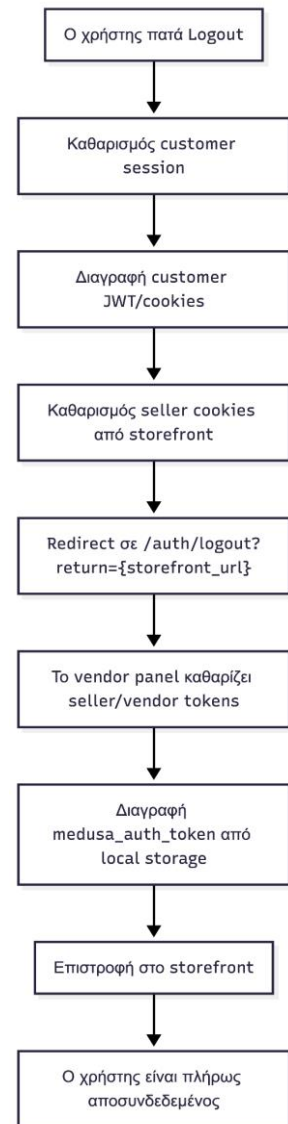


POST /auth/customer/emailpass

Εφόσον υφίσταται εγκεκριμένος λογαριασμός πωλητή, ενεργοποιείται αυτόματα η διαδικασία **seller authentication**:

POST /auth/seller/emailpass

Το εκδοθέν *seller token* διευκολύνει την άμεση μετάβαση στο vendor panel χωρίς την ανάγκη για επιπλέον διαπιστευτήρια, αξιοποιώντας το route:



/auth/accept?token={seller\_token}

Το σύστημα αποθηκεύει το medusa\_auth\_token και κατευθύνει τον χρήστη στο **seller dashboard**, ενοποιώντας την εμπειρία χρήσης.

Η διαδικασία **logout** λειτουργεί καθολικά. Με την αποσύνδεση από το storefront, εκκαθαρίζονται όλα τα customer sessions και ακολουθεί η έξοδος από το vendor panel μέσω του:

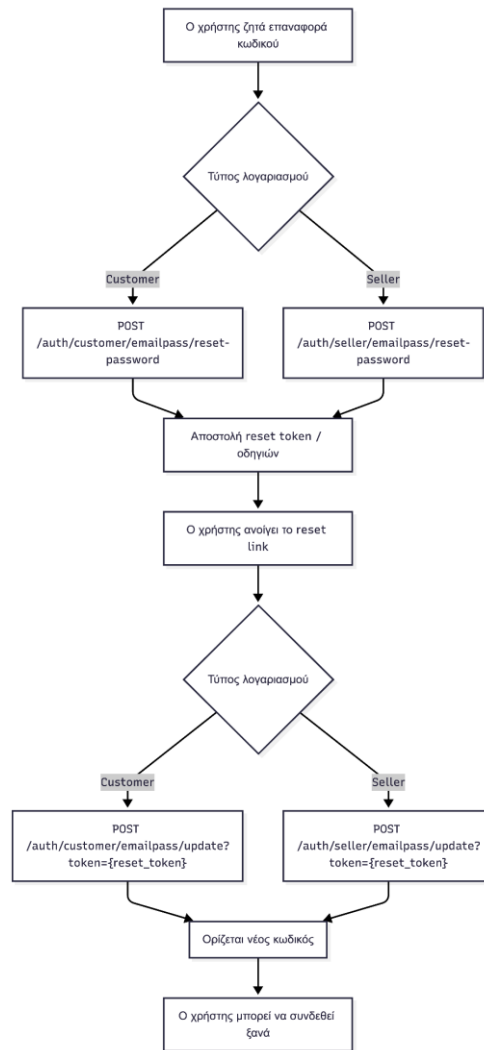
/auth/logout?return={storefront\_url}

Αυτό διασφαλίζει τον πλήρη τερματισμό των ενεργών συνεδριών σε όλα τα επίπεδα της πλατφόρμας.

Για την ανάκτηση πρόσβασης, το REFASHION αξιοποιεί τα εξειδικευμένα *auth endpoints*. Για τους αγοραστές:

POST /auth/customer/emailpass/reset-password

Αποστολή οδηγιών επαναφοράς.



POST/auth/customer/emailpass/update?token={reset\_token}

Καθορισμός νέου συνθηματικού.

Αντίστοιχα, για τους εγκεκριμένους πωλητές ισχύουν τα:

POST /auth/seller/emailpass/reset-password

POST /auth/seller/emailpass/update?token={reset\_token}

Συνοψίζοντας, η υλοποίηση προσφέρει μια **συγκροτημένη ροή εγγραφής**, όπου η ιδιότητα του πωλητή παραμένει υπό έλεγχο μέσω *approval process*, ενώ η αυθεντικοποίηση εγγυάται την απρόσκοπτη κίνηση μεταξύ των διαφορετικών περιβαλλόντων της πλατφόρμας.

## 4.2. Διαχείριση Προϊόντων Μόδας

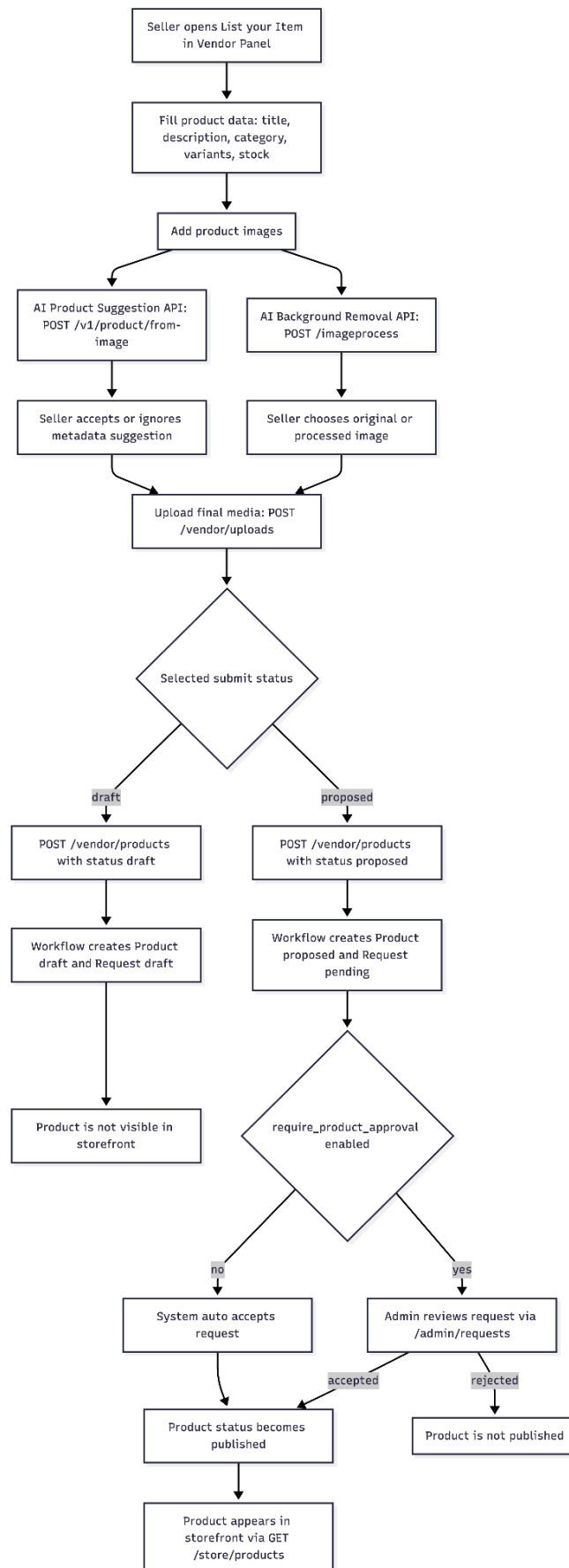
### 4.2.1 Καταχώρηση Προϊόντων (List your Item)

Η λειτουργία **List your Item** συνιστά τον πρωταρχικό μηχανισμό εισαγωγής μεταχειρισμένων ειδών μόδας από τον πωλητή στην πλατφόρμα. Η αρχιτεκτονική της ροής βασίζεται στον **αυστηρό διαχωρισμό ρόλων**, όπου η δραστηριότητα του πωλητή περιορίζεται στο *vendor scope*, ενώ ο αγοραστής αλληλεπιδρά αποκλειστικά με το *store scope*. Αυτή η προσέγγιση εγγυάται την πλήρη αυτονομία στη διαχείριση του καταλόγου,

διασφαλίζοντας παράλληλα ότι οι εσωτερικές λειτουργίες *backoffice* παραμένουν απομονωμένες από το δημόσιο *storefront*.

Η διαδικασία καταχώρησης εκκινεί από το **vendor panel** και περιλαμβάνει τα εξής στάδια:

1. Αρχικά, ο πωλητής συμπληρώνει στη φόρμα καταχώρησης τα **βασικά στοιχεία του προϊόντος**, όπως τον τίτλο, την περιγραφή, την κατηγορία, την τιμή, καθώς και τις διαθέσιμες παραλλαγές και το απόθεμα.
2. Με την προσθήκη φωτογραφικού υλικού, ενεργοποιείται η υπηρεσία **AI metadata suggestion** μέσω του endpoint `POST /v1/product/from-image`. Το σύστημα εκτελεί *reverse-search* και *LLM επεξεργασία*, προτείνοντας αυτόματα ονομασία, κατηγορία και περιγραφή, τα οποία ο πωλητής δύναται να επικαιροποιήσει.
3. Παράλληλα, παρέχεται η δυνατότητα **αυτόματης αφαίρεσης φόντου** μέσω της κλήσης `POST /imageprocess`. Η παραγόμενη *processed* έκδοση της εικόνας επιτρέπει στον χρήστη να επιλέξει την βέλτιστη απεικόνιση για το προϊόν του.
4. Τα τελικά αρχεία εικόνων μεταφορτώνονται στο **backend storage** μέσω του `POST /vendor/uploads`, από όπου και ανακτώνται τα αντίστοιχα δημόσια URLs.
5. Κατά την υποβολή, το *vendor panel* αποστέλλει το πλήρες *payload* στο `POST /vendor/products`, ορίζοντας την κατάσταση (*draft* ή *proposed*), τα thumbnails και τα λοιπά τεχνικά χαρακτηριστικά.
6. Το *backend* αναγνωρίζει αυτόματα την ταυτότητα του πωλητή μέσω του **authentication context**, διασφαλίζοντας τη σύνδεση του προϊόντος με τον ορθό λογαριασμό χωρίς την ανάγκη εξωτερικού *seller\_id*.
7. Στη συνέχεια, ενεργοποιείται το *createProductRequestWorkflow*, το οποίο δρομολογεί την επίσημη δημιουργία της οντότητας του προϊόντος στη βάση.
8. Με την οριστική **δημοσίευση (published)**, το είδος καθίσταται ορατό στο *storefront*, με τα δεδομένα να αντλούνται μέσω του endpoint `GET /store/products`.



## 4.2.2 Χαρακτηριστικά Προϊόντων

Η διαχείριση των ιδιοτήτων των ειδών βασίζεται στις εγγενείς λειτουργίες του οικοσυστήματος, διασφαλίζοντας την **ομοιόμορφη περιγραφή** και τη βέλτιστη αναζήτηση κάθε αντικειμένου. Τα θεμελιώδη χαρακτηριστικά περιλαμβάνουν:

- **Κατηγορία:** Καθορίζει το πρωταρχικό είδος για τη σωστή ταξινόμηση και το φιλτράρισμα στην πλατφόρμα.
- **Συλλογή:** Ενσωματώνει το προϊόν σε συγκεκριμένες θεματικές ενότητες (π.χ. Vintage, Casual).
- **Τύπος Προϊόντος:** Προσδιορίζει τη φύση του αντικειμένου σε πιο εξειδικευμένο τεχνικό επίπεδο.
- **Brand:** Καταγράφει τον κατασκευαστή, ενισχύοντας την εμπορική ταυτότητα και την ευκολία εντοπισμού.
- **Ετικέτες (Tags):** Δυναμικά keywords που ενισχύουν την ανακαλυψιμότητα και την ταχεία διήθηση αποτελεσμάτων.
- **Χρώμα:** Αποτυπώνει το βασικό οπτικό γνώρισμα του είδους μόδας.
- **Μέγεθος:** Δηλώνει τις φυσικές διαστάσεις για την ένδυση ή την υπόδηση.
- **Κατάσταση:** Περιγράφει τον βαθμό χρήσης του αντικειμένου (π.χ. Καινούργιο, Μεταχειρισμένο).
- **Παραλλαγές:** Συνδυαστικά χαρακτηριστικά (π.χ. Χρώμα και Μέγεθος) για την πληρέστερη απεικόνιση.
- **Τιμολόγηση Variants:** Παρέχει τη δυνατότητα ορισμού διαφορετικής αξίας για κάθε επιμέρους παραλλαγή.
- **Απόθεμα:** Εκφράζει τη διαθεσιμότητα, η οποία στις second-hand συναλλαγές αφορά συνήθως μοναδικά τεμάχια.

Μέσω αυτού του σχήματος, η πλατφόρμα συζεύγνυε τη δομημένη οργάνωση (category, collection) με τις απαραίτητες εμπορικές παραμέτρους, καθιστώντας τα προϊόντα πλήρως **λειτουργικά και αξιοποιήσιμα** εντός του marketplace.

## 4.2.3 Αποθήκευση Εικόνων

Η διαχείριση του οπτικού υλικού αποτελεί τη σπονδυλική στήλη της media στρατηγικής στο υποσύστημα **List your Item**, επιβάλλοντας έναν σαφή διαχωρισμό μεταξύ της μεταφόρτωσης αρχείων και της συσχέτισής τους με την οντότητα του προϊόντος. Η αρχιτεκτονική της ροής *Vendor Panel* → *Backend Upload API* → *File Service Plugin* → *Product Create API* διασφαλίζει ότι κάθε αρχείο αποκτά πρώτα ένα **σταθερό δημόσιο URL** προτού συνδεθεί οριστικά με το αντικείμενο πώλησης.

Κατά την επιλογή των αρχείων στο περιβάλλον του πωλητή, ο client υποβάλλει το υλικό στο εξειδικευμένο endpoint `POST /vendor/uploads` υπό μορφή *multipart/form-data*. Στο επίπεδο του backend, η χρήση middleware με `multer.memoryStorage` επιτρέπει την άμεση επεξεργασία του payload, αποφεύγοντας την προσωρινή εγγραφή σε τοπικό δίσκο. Τα αρχεία διοχετεύονται στο *uploadFilesWorkflow* του Medusa Core, το οποίο εποπτεύει την αποθήκευση στον ενεργό **File Service provider**. Για κάθε στοιχείο, ορίζονται οι τεχνικές παράμετροι (`filename`, `mimeType`) και η ιδιότητα `access: "public"`, εγγυώμενη την παραγωγή συνδέσμων έτοιμων για άμεση προβολή στο storefront.

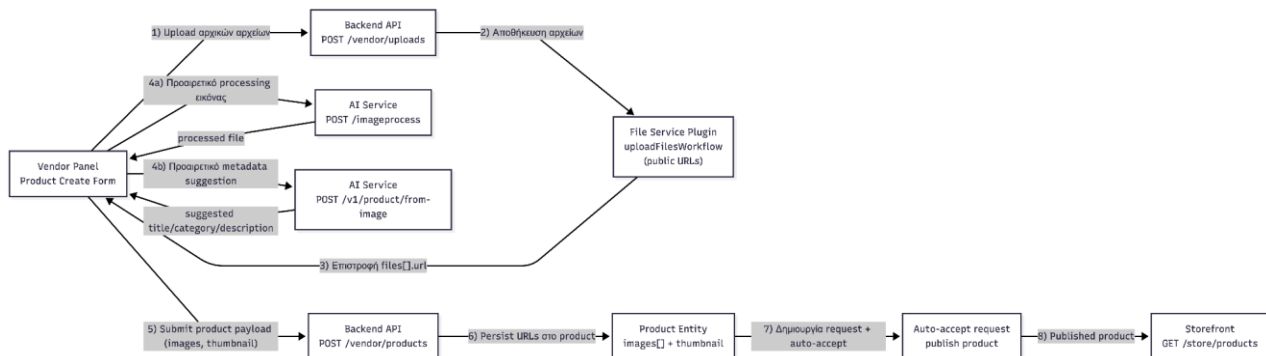
Με την επιτυχή ολοκλήρωση της μεταφόρτωσης, το σύστημα επιστρέφει το array `files[]` και η διεπαφή διατηρεί τα URLs ως υποψήφια πεδία για τις **εικόνες και το thumbnail**. Σε αυτή τη φάση, το υλικό θεωρείται αποθηκευμένο στο `file layer`, παραμένοντας ωστόσο σε αναμονή για την τελική σύνδεση, η οποία ενεργοποιείται μόνο με την υποβολή του πλήρους `product payload`.

Παράλληλα με την τυπική ροή, το σύστημα δύναται να ενεργοποιήσει **AI microservices** για τον εμπλουτισμό ή τον μετασχηματισμό των πολυμέσων:

- POST `/imageprocess`: Εκτελεί επεξεργασία (π.χ. αφαίρεση φόντου) παράγοντας μια νέα, βελτιστοποιημένη έκδοση του αρχείου.
- POST `/v1/product/from-image`: Αναλύει το οπτικό περιεχόμενο για την αυτόματη εξαγωγή στοιχείων όπως ο τίτλος, η κατηγορία και η περιγραφή.

Κατά την τελική υποβολή (POST `/vendor/products`), ο χρήστης επιλέγει για κάθε οντότητα μεταξύ της πρωτότυπης ή της επεξεργασμένης εκδοχής, συνθέτοντας το `payload` με τα πεδία `images` και `thumbnail`. Ο **backend validator** επικυρώνει τις παραμέτρους και το `workflow` δημιουργίας οριστικοποιεί τη συσχέτιση στη βάση δεδομένων, βασιζόμενο σε ήδη προσβάσιμα και έγκυρα URLs.

Βάσει της τρέχουσας πολιτικής του οικοσυστήματος REFASHION, το προϊόν καθίσταται **άμεσα ορατό** στο κοινό, ενσωματώνοντας το οπτικό του υλικό αμέσως μετά την ολοκλήρωση της συνδυαστικής ροής αποθήκευσης και εγγραφής.



### 4.3. Διαχείριση Συναλλαγών

#### 4.3.1 Διαδικασία Αγοράς (Cart & Order Workflow)

Η συναλλακτική δραστηριότητα εντός του οικοσυστήματος ακολουθεί τη θεμελιώδη αλληλουχία `Cart` → `Checkout` → `Order`, αξιοποιώντας τις δυνατότητες του Store API. Στην αρχιτεκτονική του REFASHION, η τυπική ροή εμπλουτίζεται με **marketplace επεκτάσεις**, οι οποίες διασφαλίζουν την εξειδικευμένη διαχείριση των αποστολών ανά πωλητή και την εννοποιημένη ολοκλήρωση μέσω της οντότητας `order_set`.

#### Βήματα Ροής (υλοποίηση συστήματος)

Βήμα	Ενέργεια	API Endpoint
1	Εύρεση προϊόντος & variant ID	GET <code>/store/products</code>

2	Εύρεση region ID	GET /store/regions
3	Δημιουργία καλαθιού	POST /store/carts
4	Προσθήκη προϊόντος στο καλάθι	POST /store/carts/{id}/line-items
5	Ανάκτηση μεθόδων αποστολής	GET /store/shipping-options?cart_id={id}
6	Προσθήκη μεθόδου αποστολής	POST /store/carts/{id}/shipping-methods
7	Δημιουργία payment session	POST /store/carts/{id}/payment-sessions
8	Ολοκλήρωση παραγγελίας	POST /store/carts/{id}/complete

Στο αρχικό στάδιο, ο πελάτης πλοηγείται στον κατάλογο και επιλέγει συγκεκριμένο *variant*. Η ακρίβεια της παραλλαγής (μέγεθος, κατάσταση, τεχνικά χαρακτηριστικά) είναι καθοριστική, καθώς η προσθήκη στο καλάθι εκτελείται αποκλειστικά σε επίπεδο *variant*. Ταυτόχρονα, πραγματοποιείται η αντιστοίχιση της περιοχής (**region resolution**) βάσει του κωδικού χώρας, ώστε να διασφαλιστεί η ορθότητα στο νόμισμα, την τιμολόγηση και τις διαθέσιμες υπηρεσίες *fulfillment*.

Κατά τη δημιουργία του καλαθιού, το σύστημα αποθηκεύει το *cart\_id* σε ασφαλές cookie, επιτρέποντας την επαναχρησιμοποίησή του στις επόμενες φάσεις. Εφόσον δεν υφίσταται ενεργό καλάθι, παράγεται νέα οντότητα. Στην περίπτωση που το υπάρχον καλάθι αντιστοιχεί σε διαφορετική γεωγραφική ζώνη από τη χώρα του χρήστη, το σύστημα επικαιροποιεί αυτόματα τη *region* για τη διατήρηση της οικονομικής συνέπειας.

Στο επίπεδο της αποστολής, η διαδικασία ανακτά τις διαθέσιμες επιλογές (*shipping options*) βάσει του περιεχομένου. Εδώ ενσωματώνεται η **marketplace φιλοσοφία**: το backend δεν αντιμετωπίζει τις αποστολές ως ενιαίο σύνολο, αλλά επικυρώνει τη συμβατότητα των μεθόδων με τους πωλητές και τα είδη του καλαθιού. Κατά την εγκαθίδρυση της μεθόδου:

- Αποκλείεται η ύπαρξη πλεοναζουσών ή διπλών επιλογών.
- Απορρίπτονται επιλογές που δεν συνάδουν με το προφίλ των ενεργών πωλητών.
- Η τροποποίηση της μεθόδου περιορίζεται αυστηρά εντός του πλαισίου (*scope*) του εκάστοτε πωλητή.

Στη φάση της πληρωμής, η διεπαφή επιλέγει τον πάροχο (*provider*) από το διαθέσιμο σύνολο της περιοχής και εγκαθιδρύει το *payment session*. Για τη χρήση του Stripe, προηγείται η επικύρωση του *payment intent* στην πλευρά του πελάτη (μέσω *client\_secret*) πριν την οριστική ολοκλήρωση. Σε δοκιμαστικούς ή χειροκίνητους παρόχους, η ροή

οδηγείται απευθείας στο τελικό στάδιο.

Η οριστικοποίηση της συναλλαγής (POST /store/carts/{id}/complete) στο REFASHION αποτελεί μια **εξειδικευμένη marketplace ροή**: αντί για μια μεμονωμένη παραγγελία, παράγεται ένα *order\_set* με επιμέρους παραγγελίες ανά πωλητή. Κατά την εκτέλεση, διενεργούνται έλεγχοι διαθεσιμότητας και εγκυρότητας, πραγματοποιείται ο διαχωρισμός των προϊόντων, δεσμεύεται το απόθεμα και εκπέμπονται τα σχετικά *events*. Με αυτόν τον τρόπο, διασφαλίζεται ο πλήρης διαχωρισμός της εμπορικής ευθύνης, ενώ ο αγοραστής απολαμβάνει μια ενοποιημένη εμπειρία.

Τέλος, εξειδικευμένοι *subscribers* στον κύκλο ζωής του *order\_set* αναλαμβάνουν την είσπραξη (*capture*) και την ενημέρωση των επιμέρους πληρωμών. Μετά την επιτυχή κατάληξη, το frontend εκκαθαρίζει τα cookies και κατευθύνει τον χρήστη στη σελίδα επιβεβαίωσης.

### 4.3.2 Πληρωμές

Η διαχείριση των οικονομικών συναλλαγών βασίζεται στο **Medusa Payment Module**, ενσωματώνοντας τον εξειδικευμένο πάροχο **Stripe Connect** (@mercurijs/payment-stripe-connect). Κατά τη φάση του checkout, η διεπαφή δρομολογεί την εγκαθίδρυση ή επικαιροποίηση του *payment session* για το τρέχον καλάθι. Το σύστημα ανακτά τα απαραίτητα δεδομένα, όπως το *client\_secret*, και ενεργοποιεί τη διαδικασία **confirmCardPayment** μέσω του Stripe.js. Με αυτόν τον τρόπο, η επεξεργασία των στοιχείων κάρτας παραμένει απομονωμένη από τους servers της πλατφόρμας, διασφαλίζοντας την απόλυτη **PCI-compliance** συμμόρφωση.

Μόλις οριστικοποιηθεί η επιβεβαίωση της καταβολής, πυροδοτείται η εντολή *complete* του καλαθιού, οδηγώντας στην παραγωγή του σχετικού **order set**. Στην περίπτωση που το τελικό ποσό είναι μηδενικό (για παράδειγμα, λόγω καθολικής χρήσης *gift cards*), η ροή του UI παρακάμπει αυτόματα το στάδιο εισαγωγής στοιχείων πληρωμής. Η συγκεκριμένη λειτουργία **free checkout** επιτρέπει την άμεση ολοκλήρωση της παραγγελίας, αξιοποιώντας, όπου κρίνεται απαραίτητο, έναν *manual* ή συστημικό *payment provider* ως εναλλακτική μέθοδο.

### 4.3.3 Αποστολές

Η στρατηγική **διαχείρισης αποστολών** εδράζεται στην αρχική χρήση του *fulfillment-manual* (provider id: manual\_manual), διασφαλίζοντας τη σταθερότητα της διαδικασίας checkout χωρίς την ανάγκη άμεσης διασύνδεσης με εξωτερικά courier APIs.

Στο επίπεδο της ροής υλοποίησης, το storefront εκτελεί ανάκτηση των διαθέσιμων επιλογών μέσω του endpoint GET /store/shipping-options?cart\_id={id}, ενώ η οριστικοποίηση της επιλεγμένης μεθόδου στο καλάθι πραγματοποιείται με την κλήση POST /store/carts/{id}/shipping-methods.

Εντός του **marketplace context**, η εφαρμογή των αποστολών διέπεται από **εξειδικευμένους ελέγχους**: το backend εκτελεί φιλτράρισμα και επικύρωση των *shipping options* ανά πωλητή, εγγυώμενο ότι κάθε είδος εξυπηρετείται από συμβατή μέθοδο. Η προσέγγιση αυτή επιτρέπει την ορθή διαχείριση *multi-seller carts* και αποτρέπει την εμφάνιση μη έγκυρων συνδυασμών μεταξύ προϊόντων και επιλογών αποστολής.

Στο πλαίσιο της μελλοντικής εξέλιξης, προβλέπεται η αναβάθμιση ή επέκταση του *manual provider* μέσω ενός **custom Fulfillment Provider plugin** για smart lockers. Η νέα αυτή αρχιτεκτονική δύναται να ενσωματώσει:

- **Δυναμικό έλεγχο:** Παρακολούθηση της διαθεσιμότητας των θυρίδων (lockers) σε πραγματικό χρόνο.
- **Κοστολόγηση ανά Ζώνη:** Αυτόματος υπολογισμός των εξόδων παράδοσης με βάση το locker zone.
- **Structured Metadata:** Παροχή δομημένων δεδομένων στο checkout για την επιλογή του σημείου παραλαβής από τον τελικό χρήστη.

#### 4.4. Αναζήτηση & Φιλτράρισμα

Το υποσύστημα **αναζήτησης και φιλτραρίσματος** εδράζεται σε μια *URL-driven* αρχιτεκτονική στο επίπεδο του frontend, διασφαλίζοντας ότι η εκάστοτε κατάσταση του καταλόγου παραμένει πλήρως αναπαραγωγίμη, διαμοιράσιμη και συνεπής.

Κάθε επιλογή του χρήστη μεταφράζεται άμεσα σε **query parameters** (ενδεικτικά: color, size, min\_price, max\_price, range, limit, sortBy), προσφέροντας τα ακόλουθα πλεονεκτήματα:

- Διατήρηση σταθερού *state* μετά από ανανέωση (refresh) της σελίδας.
- Δυνατότητα άμεσου **deep-linking** σε εξειδικευμένες όψεις του καταλόγου.
- Σαφή διαχωρισμό μεταξύ του *presentation layer* και της κατάστασης των φίλτρων (*filter state*).

Η τρέχουσα υλοποίηση επικεντρώνεται αποκλειστικά στο **frontend layer**, χωρίς να απαιτούνται παρεμβάσεις στη δομή του backend κώδικα.

##### 4.4.1 Αναζήτηση Προϊόντων

Η ροή αναζήτησης εκκινεί από το ειδικό *search input* του storefront, το οποίο μεταβιβάζει τον όρο αναζήτησης ως query parameter στη σελίδα του καταλόγου.

Στο επίπεδο της εμπειρίας χρήστη (UX), ο μηχανισμός εγγυάται τη χρήση **shareable URLs** και την απρόσκοπτη επαναφορά της επιθυμητής κατάστασης περιήγησης.

Στην υφιστάμενη έκδοση του *sidebar filtering pipeline*, η επιχειρησιακή εφαρμογή των κριτηρίων εστιάζει στις παραμέτρους **τιμής, χρώματος και μεγέθους**.

##### 4.4.2 Φίλτρα Καταλόγου (Αριστερό Sidebar)

Ο κατάλογος πλαισιώνεται από ένα λειτουργικό αριστερό sidebar που ενσωματώνει τα εξής φίλτρα:

- **Εύρος Τιμής (Price Range):** Ειδικός διπλός *range slider* για τον ορισμό ορίων (min\_price, max\_price) με οπτική ένδειξη ενεργού τμήματος.
- **Χρώμα (Color):** Δυνατότητα πολλαπλής επιλογής (multi-select) μέσω ειδικών *color swatches*.
- **Μέγεθος (Size):** Επιλογή μέσω *size chips* με υποστήριξη πολλαπλών κριτηρίων.

Η λογική εφαρμογής είναι συνδυαστική, ακολουθώντας τους κανόνες:

- **AND** μεταξύ διαφορετικών κατηγοριών (π.χ. τιμή ΚΑΙ χρώμα).
- **OR** εντός της ίδιας κατηγορίας (π.χ. χρώμα Κόκκινο Ή Μπλε).

Για τη βελτίωση της χρηστικότητας παρέχονται επιπλέον:

- **Active filter chips** για την άμεση ορατότητα των ενεργών κριτηρίων.

- Δυνατότητα αφαίρεσης μεμονωμένων επιλογών.
- *Clear all filters* για την ολική επαναφορά (*reset*) του καταλόγου.

#### 4.4.3 Τεχνική Ροή Εφαρμογής

Η λειτουργική ακολουθία του υποσυστήματος εξελίσσεται ως εξής:

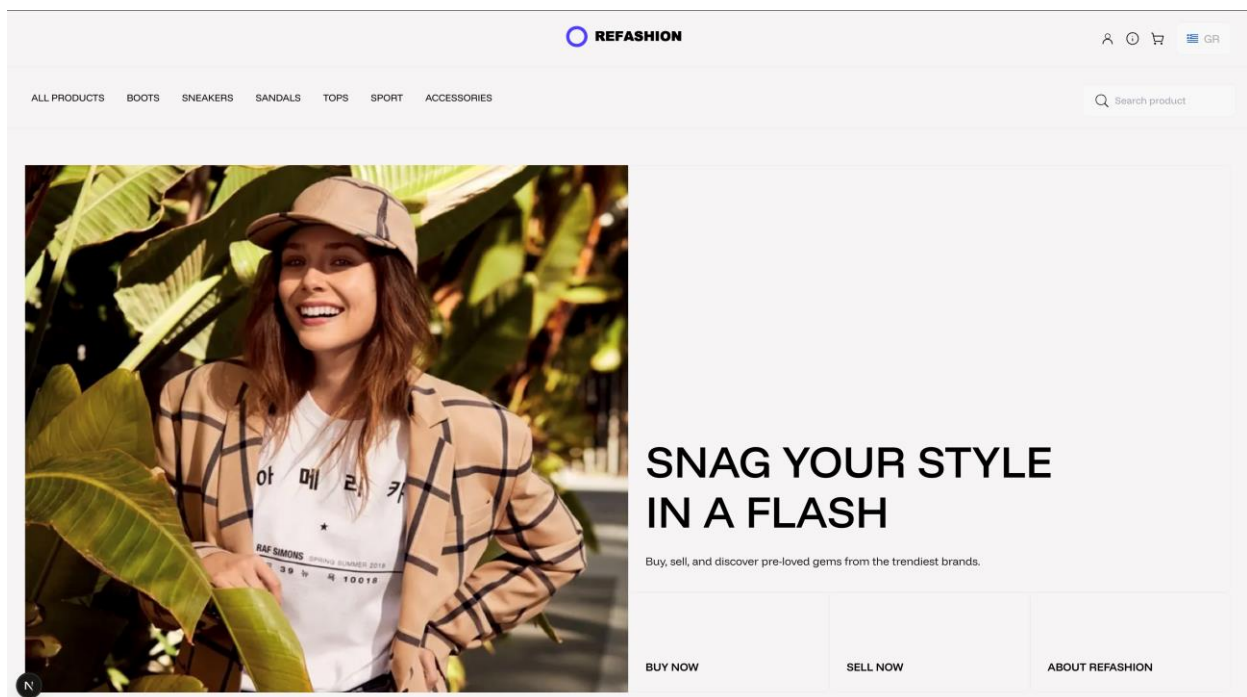
1. Αλληλεπίδραση του χρήστη με τα στοιχεία φιλτραρίσματος στο sidebar.
2. Επικαιροποίηση των **query parameters** στο URL από το frontend.
3. Πυροδότηση νέου *render* του καταλόγου λόγω της μεταβολής του URL.
4. Ανάκτηση δεδομένων από το **Store API** και εφαρμογή της λογικής *intersection* (τιμή/χρώμα/μέγεθος) στο Next.js layer (server-side).
5. Δυναμικός υπολογισμός των διαθέσιμων ορίων και επιλογών φιλτραρίσματος.
6. Εφαρμογή της σελιδοποίησης (*pagination*) επί του ήδη φιλτραρισμένου συνόλου αποτελεσμάτων.
7. Αυτόματο *reset* της σελίδας (*page=1*) σε κάθε αλλαγή κριτηρίου για τη διασφάλιση της λογικής συνέπειας.

Η συγκεκριμένη αρχιτεκτονική προσέγγιση εξασφαλίζει προβλέψιμη συμπεριφορά, καθαρό **URL state** και επιτρέπει τη μελλοντική ενσωμάτωση επιπλέον κριτηρίων (όπως *condition, rating, brand*) ή τη μετάβαση σε *backend-driven filtering*.

## 5. Frontend Υλοποίηση

### 5.1. Web Storefront

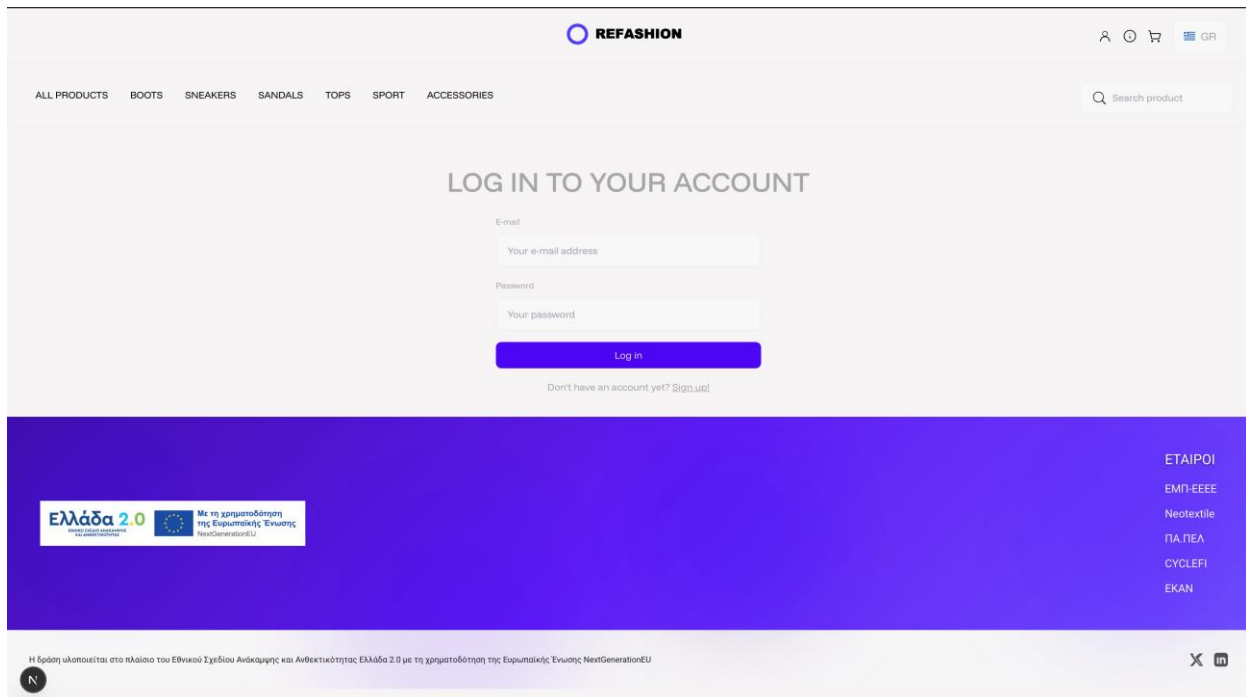
Η διεπαφή **Web Storefront** αποτελεί το κεντρικό σημείο αλληλεπίδρασης των χρηστών με το οικοσύστημα REFASHION. Η υλοποίηση βασίζεται στις τεχνολογίες **Next.js 15**, **React 19** και **TypeScript**, αξιοποιώντας την αρχιτεκτονική του *App Router* για τη βέλτιστη δρομολόγηση. Το περιβάλλον αφορά αποκλειστικά το *customer-facing score* και συνδέεται με το backend μέσω του **Medusa/Mercur REST API**, κάνοντας χρήση του επίσημου JS-SDK ή απευθείας *fetch* κλήσεων.



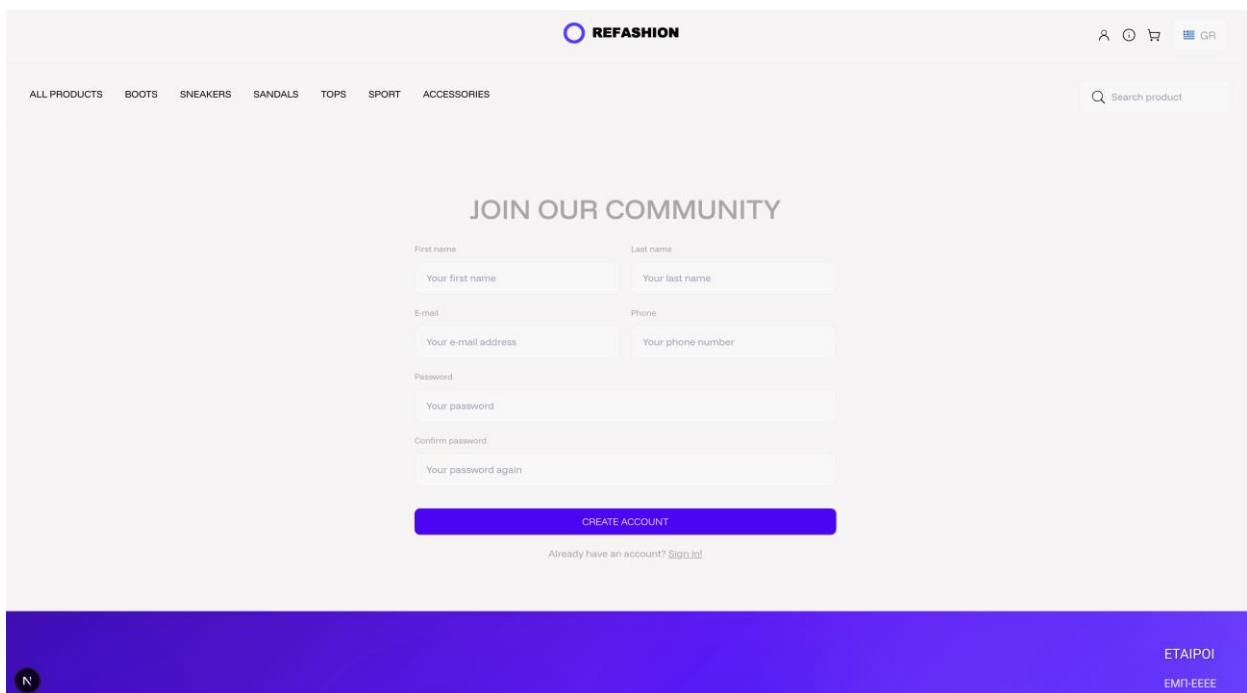
Εικόνα 2 - Αρχική σελίδα Storefront

Το υποσύστημα ενσωματώνει όλες τις κρίσιμες λειτουργίες ενός **B2C marketplace**, περιλαμβάνοντας την αυθεντικοποίηση, την καταλογογράφηση ειδών, τη διαχείριση καλαθιού και τη ροή ολοκλήρωσης παραγγελιών. Η frontend αρχιτεκτονική ακολουθεί τη μεθοδολογία των *reusable components* (*atoms, molecules, organisms*), διασφαλίζοντας ότι η πλατφόρμα παραμένει πλήρως **modular και επεκτάσιμη**.

Η εγγραφή των μελών δρομολογείται μέσω των routes */user* και */user/register*. Κατά την αρχική είσοδο εγκαθιδρύεται ένας *customer account*, ενώ πυροδοτείται ταυτόχρονα η αίτηση δημιουργίας **seller profile**. Μέσω του κεντρικού *dashboard*, ο χρήστης εποπτεύει το ψηφιακό του αποτύπωμα, τις διευθύνσεις, το ιστορικό συναλλαγών και τα προσωπικά του στατιστικά.

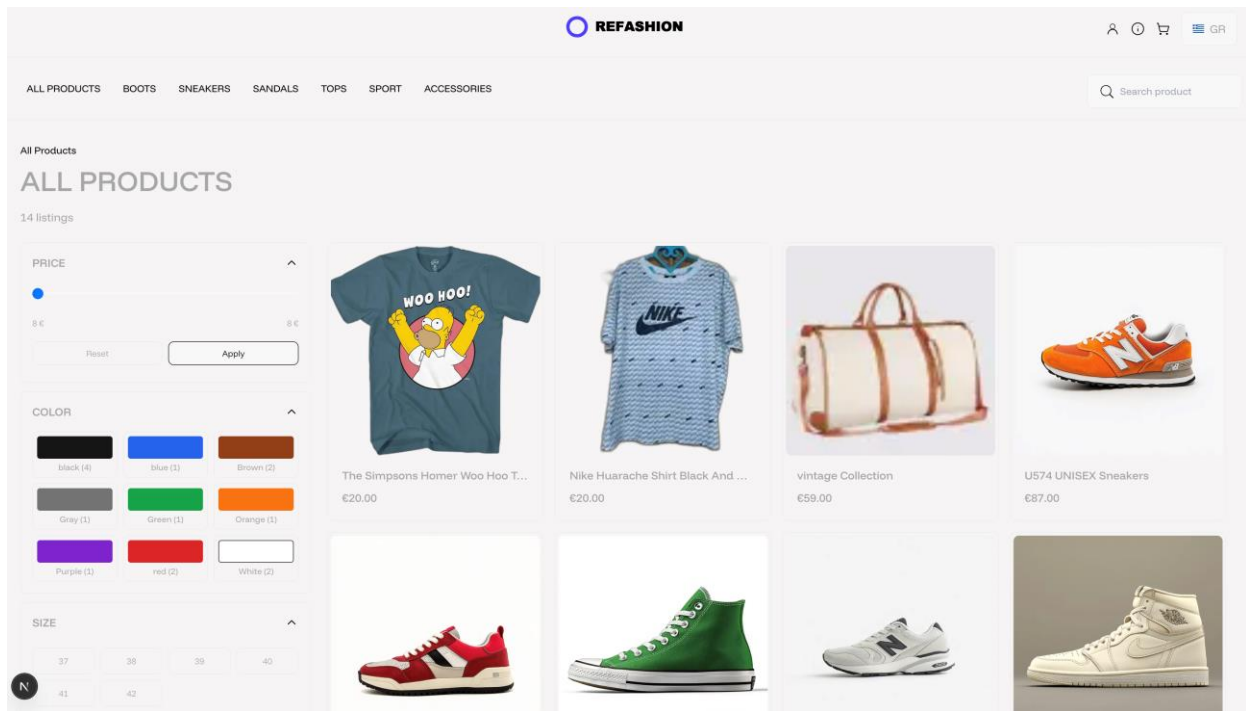


Εικόνα 3 - Σελίδα εισόδου χρήστη

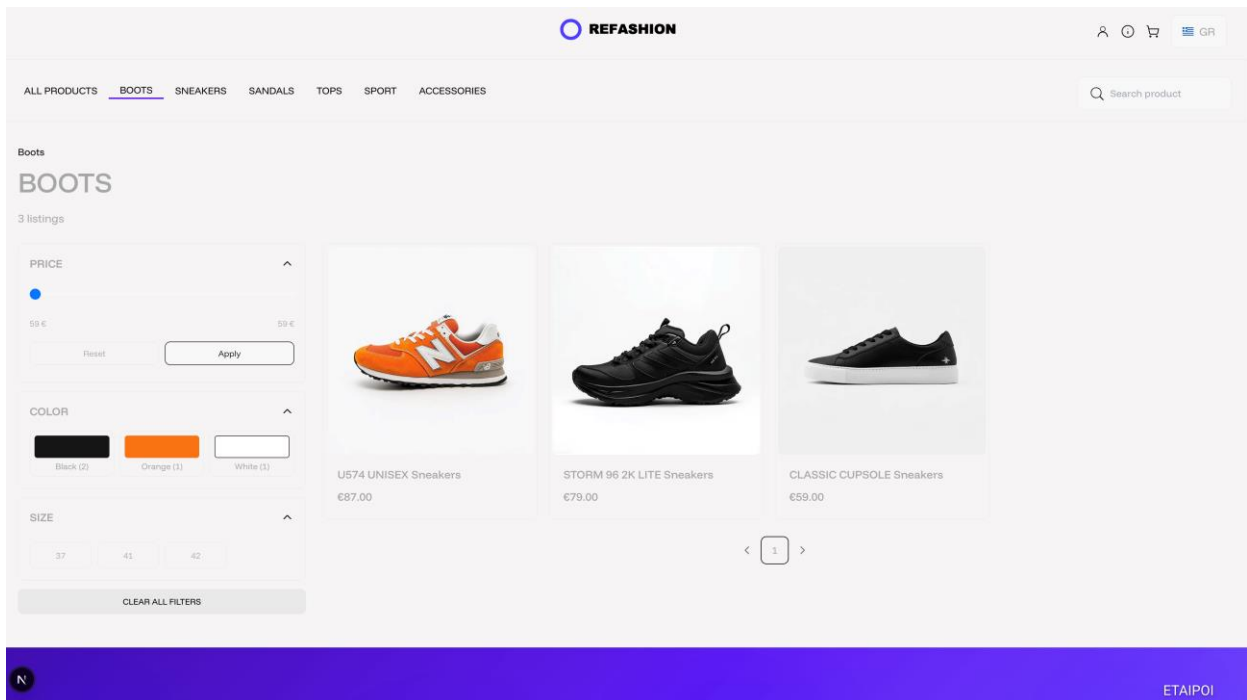


Εικόνα 4 - Σελίδα εγγραφής χρήστη

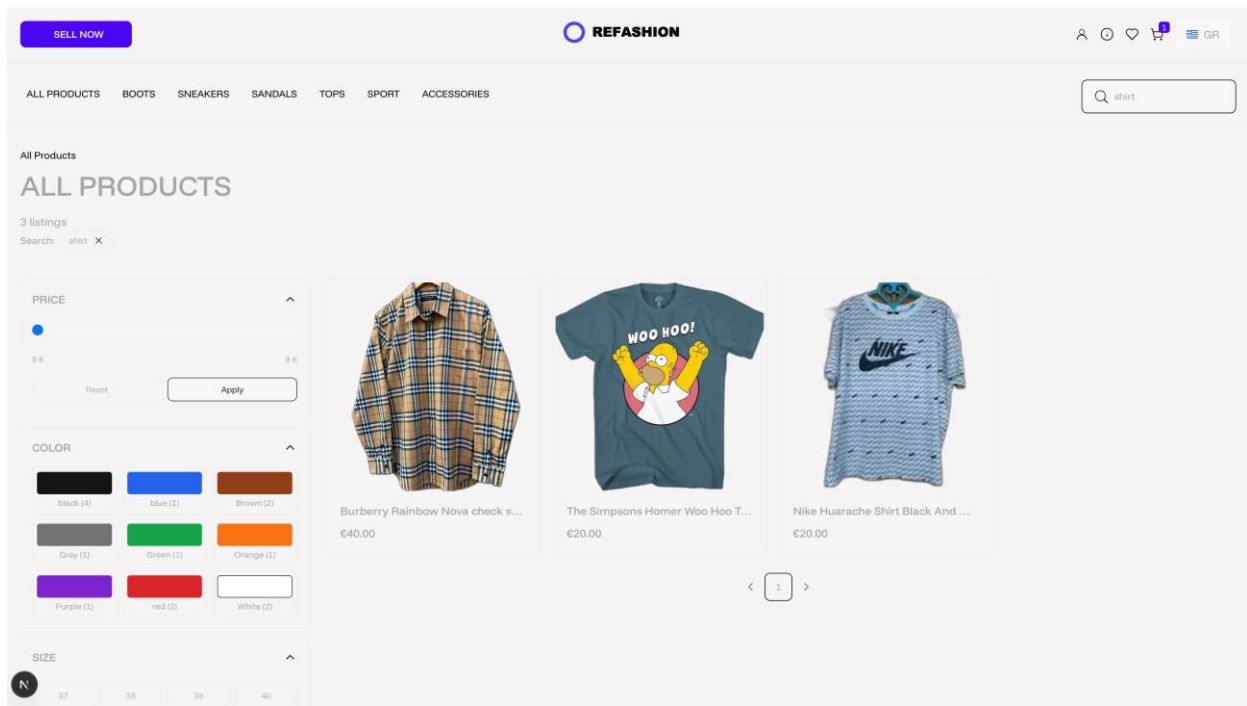
Η πλοήγηση στον κατάλογο υποστηρίζεται από δυναμικά routes (/categories, /products), με τα δεδομένα να αντλούνται από το endpoint GET /store/products. Το storefront προβάλλει εκτενή στοιχεία για κάθε είδος, όπως παραλλαγές, διαθεσιμότητα και **AI-powered προτάσεις**. Παρόλο που η πώληση εκκινεί με το κουμπί *Sell now*, η διαδικασία **List your Item** εκτελείται αποκλειστικά εντός του εξειδικευμένου vendor panel.



Εικόνα 5 - Σελίδα καταλόγου προϊόντων

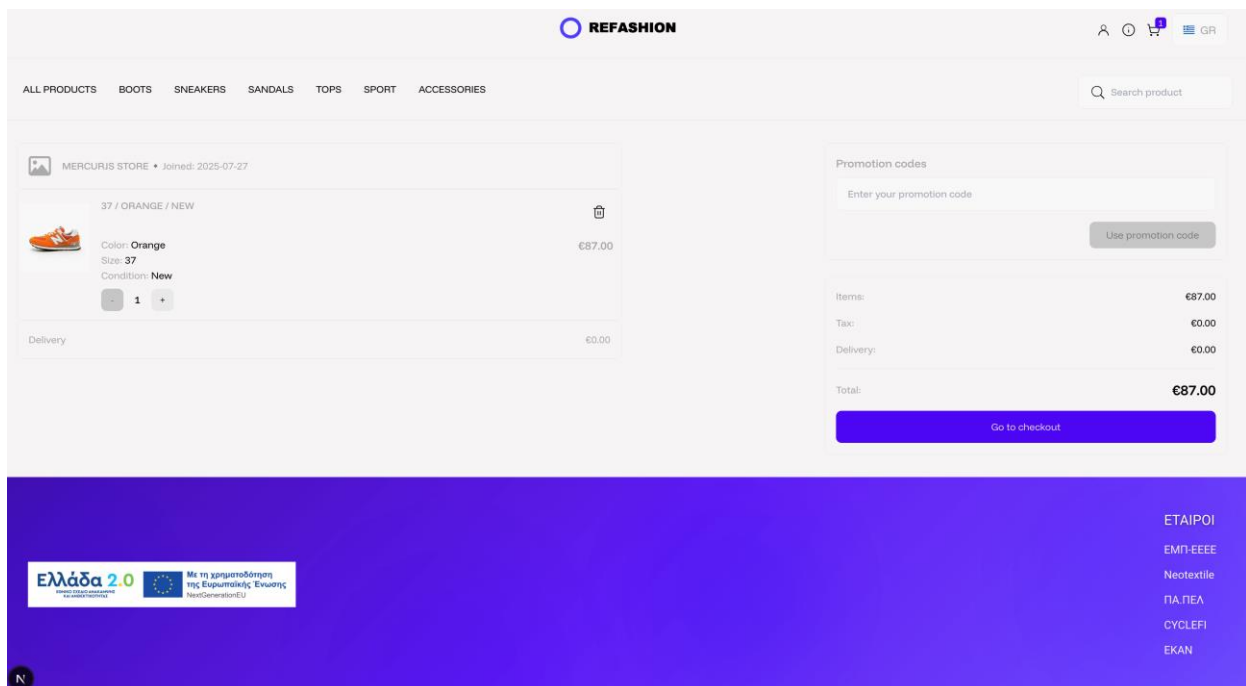


Εικόνα 6 - Σελίδα συγκεκριμένης κατηγορίας προϊόντων

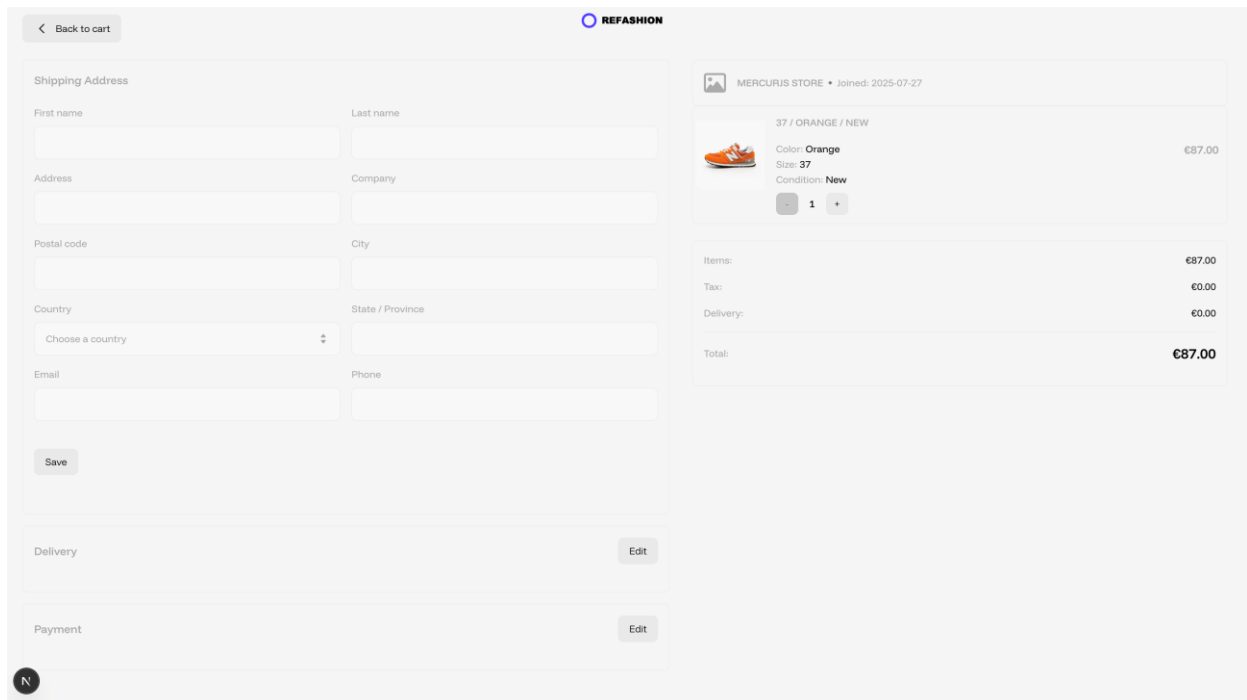


Εικόνα 7 - Σελίδα αποτελεσμάτων αναζήτησης με τον όρο "shirt"

Η εμπορική ροή ακολουθεί το σχήμα *Cart* → *Checkout* → *Order*. Παρέχεται η δυνατότητα χρήσης κουπονιών, επιλογής μεθόδων αποστολής και ασφαλούς πληρωμής. Η σελίδα **checkout** είναι οργανωμένη σε διακριτές ενότητες (*address*, *payment*, *review*), διασφαλίζοντας την ορθή εκτέλεση του *cart completion workflow* στο backend.

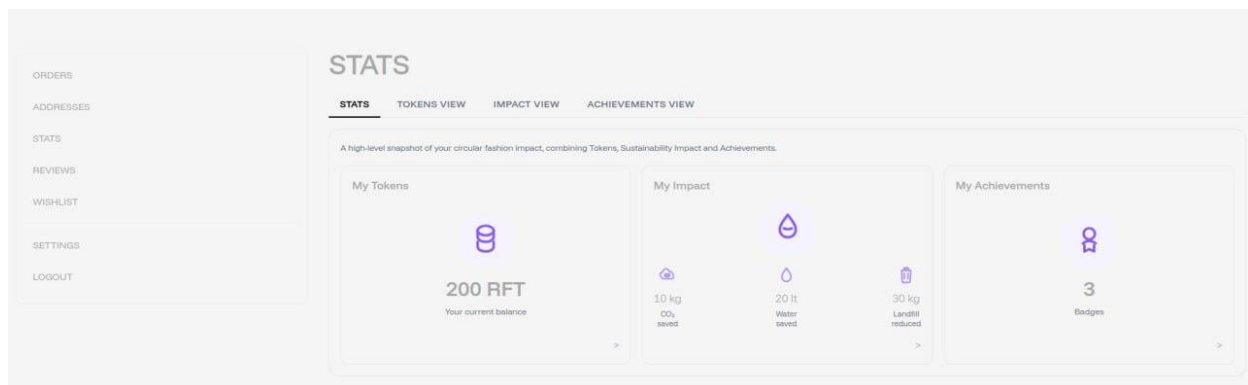


Εικόνα 8 - Σελίδα καλαθιού χρήστη

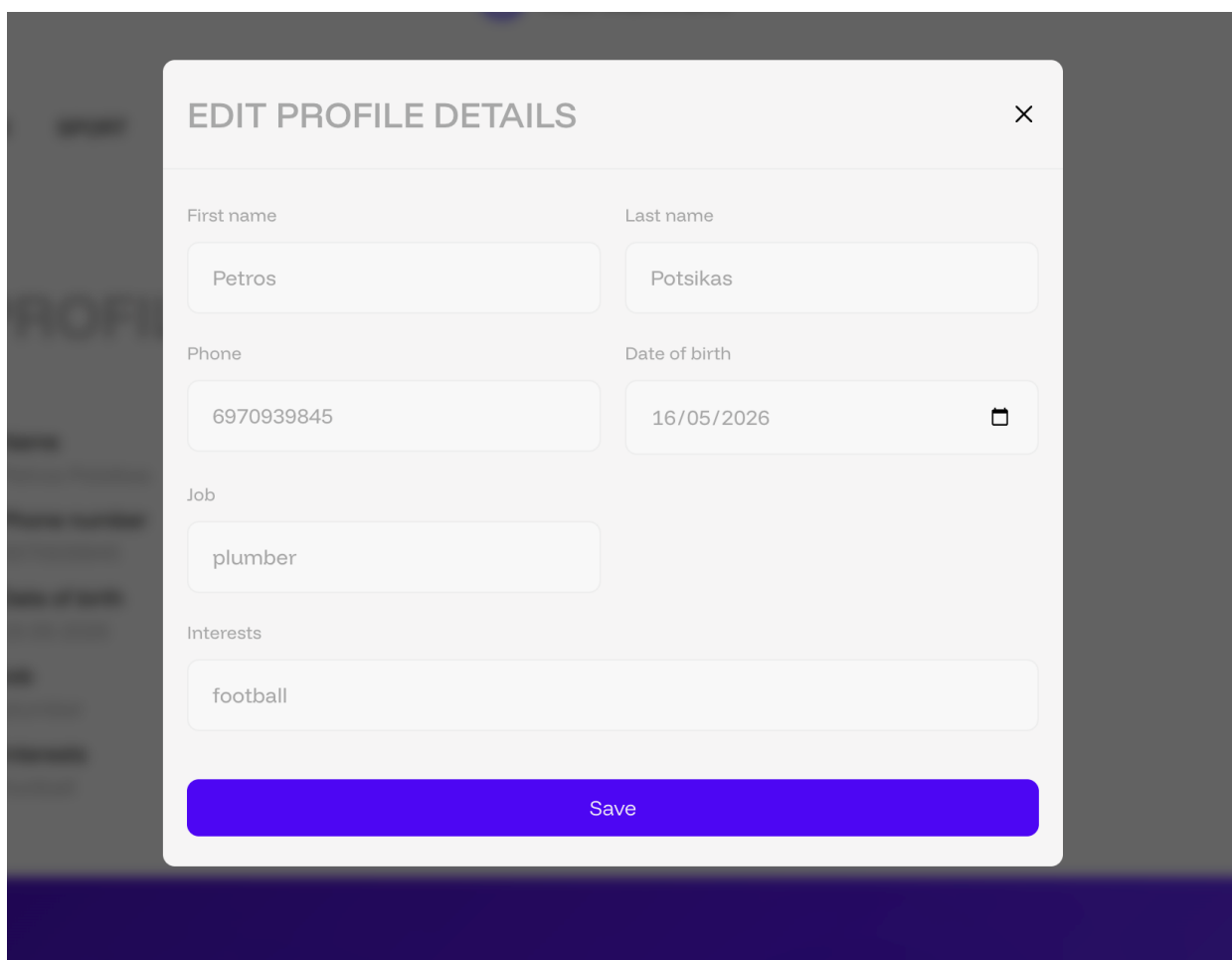


Εικόνα 9 - Σελίδα checkout

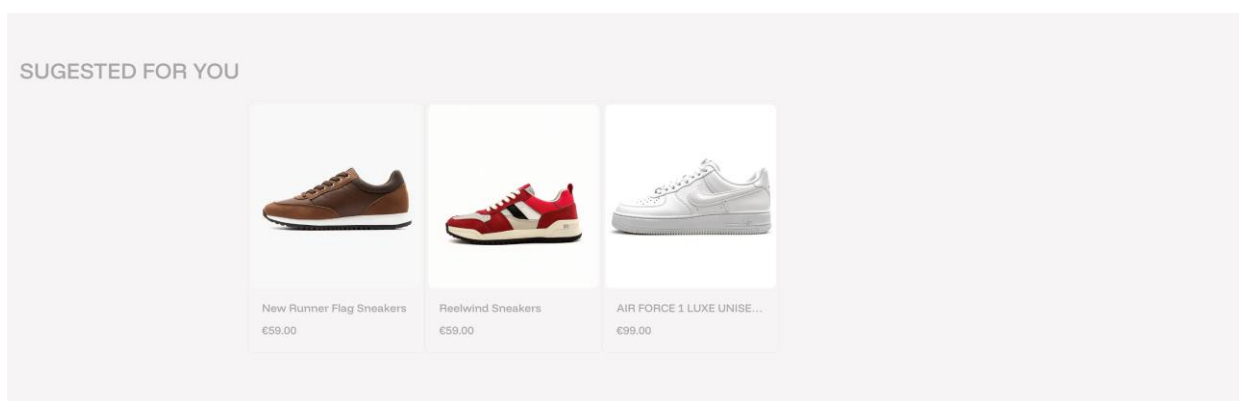
Καινοτόμο στοιχείο αποτελεί η ενσωμάτωση του **Eco-Impact & Gamification module**. Μέσω του /user/stats, προβάλλονται τα ReWear Tokens και οι περιβαλλοντικοί δείκτες (εξοικονόμηση CO<sub>2</sub> και νερού). Ταυτόχρονα, το σύστημα αποδίδει *badges* (Achievements) για την επιβράβευση της ενεργού συμμετοχής στην κυκλική οικονομία.



Εικόνα 10 - Σελίδα στατιστικών χρήστη



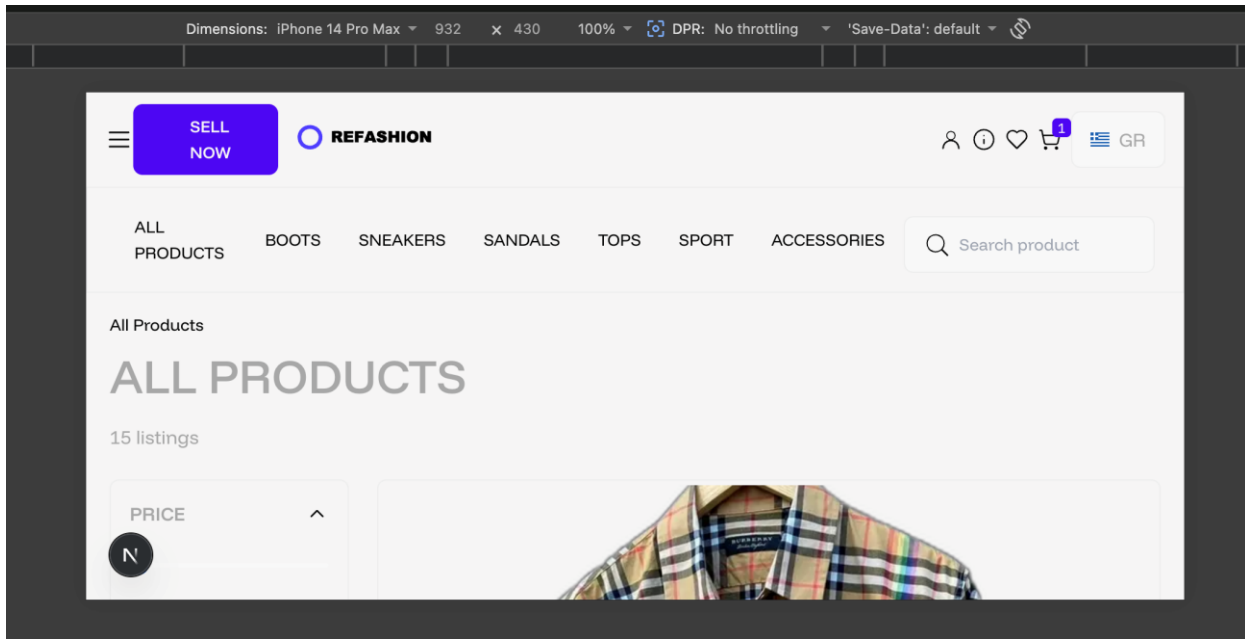
Εικόνα 11 - Σελίδα Επεξεργασίας προφίλ χρήστη που τα στοιχεία χρησιμοποιούνται για AI product suggestions



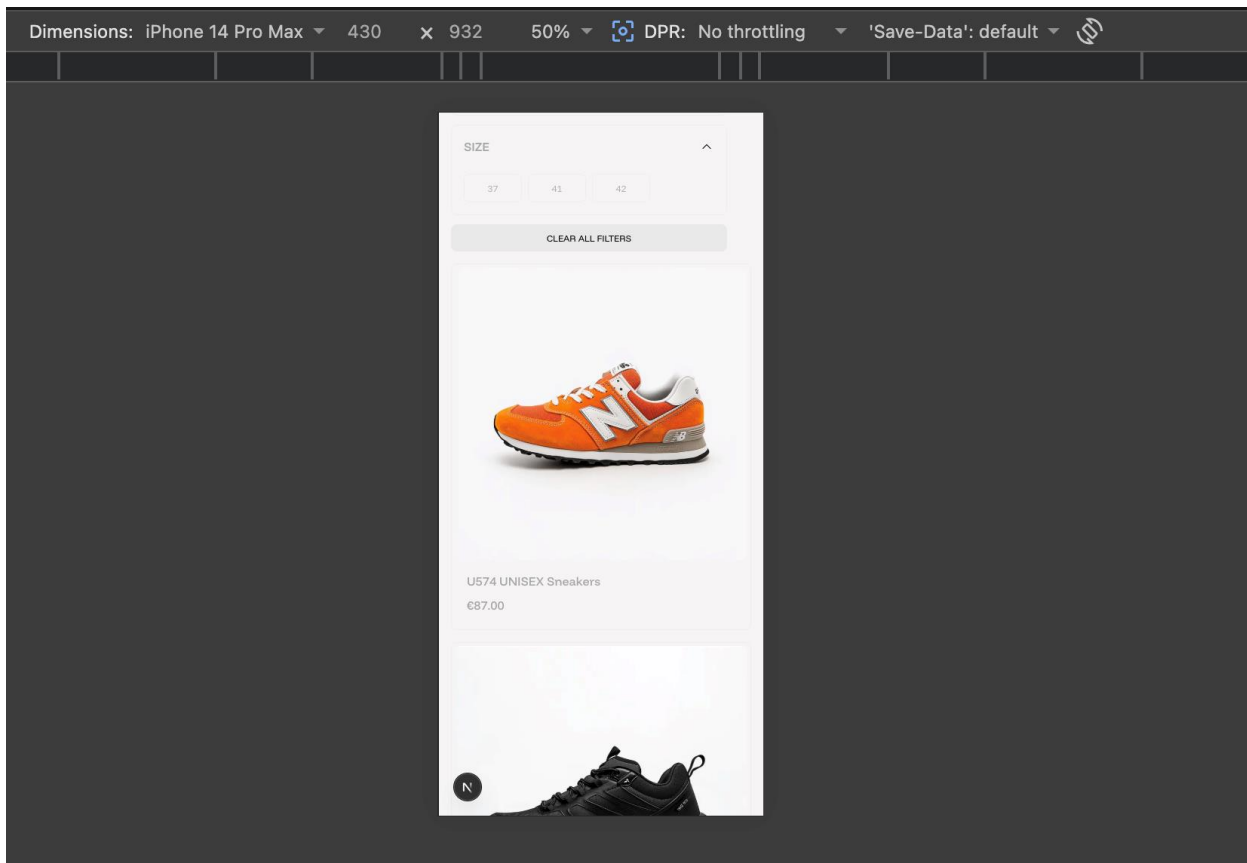
Εικόνα 12 - Προτεινόμενα προϊόντα

Η πλατφόρμα προσφέρει μια **region-aware** εμπειρία, όπου εξειδικευμένο middleware αναγνωρίζει την περιοχή του χρήστη και προσαρμόζει αυτόματα το νόμισμα, τις τιμές και τις επιλογές *fulfillment*. Η αλλαγή γεωγραφικής ζώνης μέσω του *country selector* επικαιροποιεί άμεσα το περιεχόμενο του καλαθιού και τη διαθεσιμότητα των προϊόντων.

Τέλος, η υλοποίηση διαθέτει **responsive σχεδιασμό** με χρήση **Tailwind CSS**, προσφέροντας ενοποιημένη και σύγχρονη εμπειρία χρήσης σε desktop και mobile περιβάλλοντα, ενισχύοντας την προσβασιμότητα σε όλα τα βιώσιμα χαρακτηριστικά του marketplace.



Εικόνα 13 - responsive σχεδιασμός



Εικόνα 14 - responsive σχεδιασμός

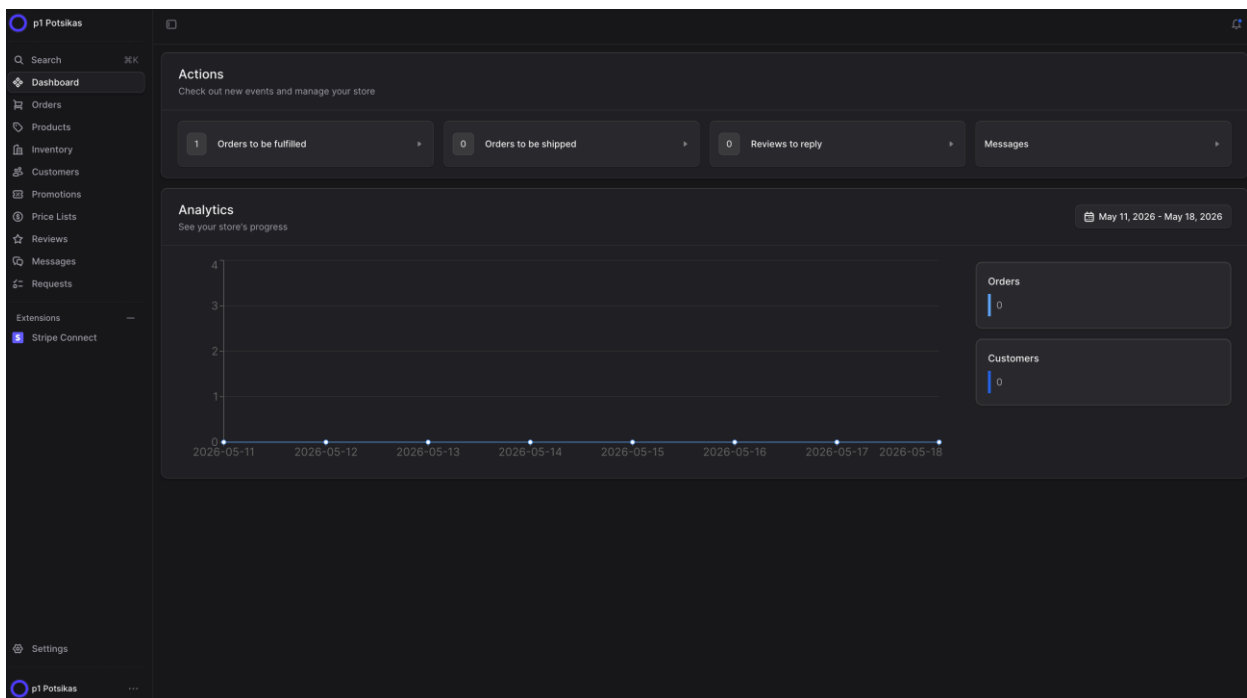
## 5.2. Admin Panel

### 5.2.1 Seller Admin Panel / Vendor Panel

Το **Seller Admin Panel** συνιστά το εξειδικευμένο περιβάλλον διαχείρισης των πωλητών

εντός του οικοσυστήματος ReFashion. Πρόκειται για μια διακριτή frontend εφαρμογή, πλήρως αποσυνδεδεμένη από το storefront των πελατών, η οποία έχει αναπτυχθεί με τη χρήση **React**, **Vite**, **TypeScript** και τη βιβλιοθήκη **Medusa UI**. Αποστολή του panel είναι ο εξοπλισμός των εγκεκριμένων πωλητών με τα απαραίτητα εργαλεία για την αυτόνομη λειτουργία του ψηφιακού τους καταστήματος εντός του marketplace.

Η πρόσβαση στο vendor panel περιορίζεται αποκλειστικά σε χρήστες που διαθέτουν ενεργό και εγκεκριμένο *seller account*. Κατά τη σύνδεση από το storefront, εφόσον υφίσταται η σχετική ιδιότητα, η μετάβαση ενεργοποιείται μέσω του κουμπιού *Sell now*, αξιοποιώντας τη διαδικασία **token handoff** στη διαδρομή `/auth/accept?token=...`. Το panel αποθηκεύει το *seller token* και κατευθύνει τον χρήστη στο dashboard, ενώ σε περίπτωση έλλειψης έγκυρων διαπιστευτηρίων, επιβάλλεται η ανακατεύθυνση στη σελίδα login.



Εικόνα 15 - Dashboard σελίδας πωλητή

Η ασφάλεια του περιβάλλοντος διασφαλίζεται μέσω **protected routes**, όπου κατά την είσοδο εκτελείται επαλήθευση της ταυτότητας μέσω του endpoint `GET /vendor/sellers/me`. Εάν το κατάστημα βρίσκεται σε κατάσταση *suspended* κατόπιν απόφασης του administrator, το σύστημα προβάλλει σχετική προειδοποίηση περιορισμού των λειτουργιών.

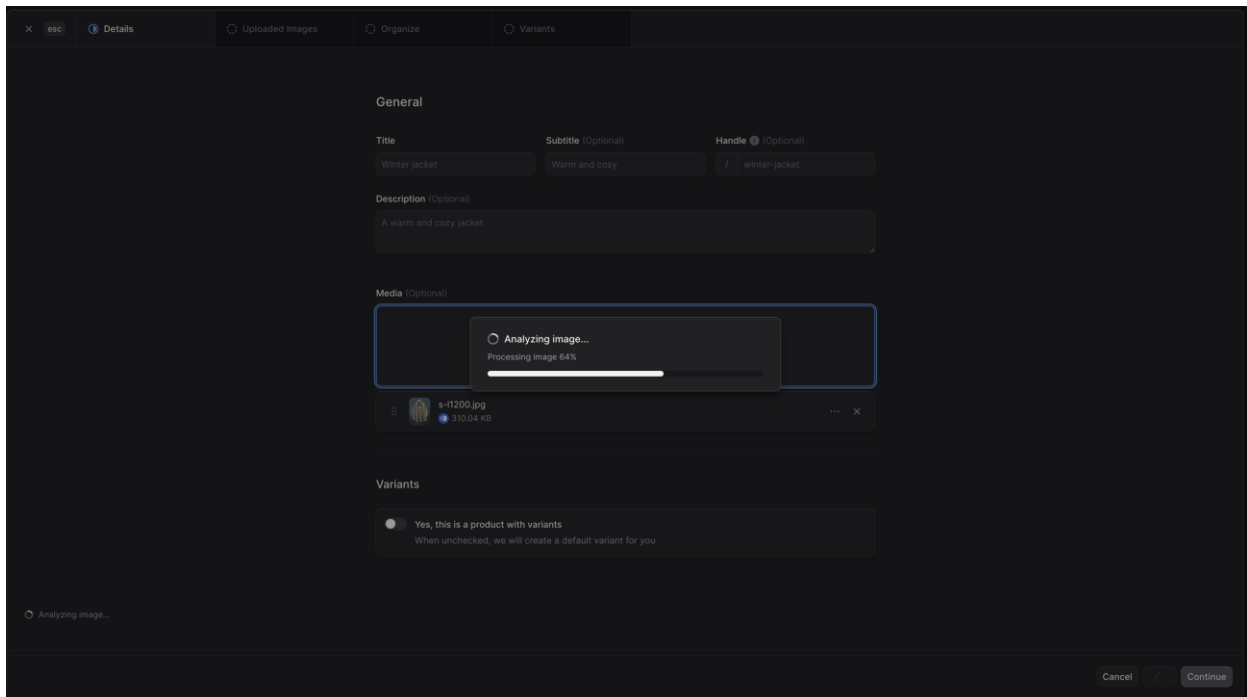
Θεμελιώδης αρχή του panel είναι η διαχείριση **seller-scoped δεδομένων**. Κάθε πωλητής έχει ορατότητα και έλεγχο αποκλειστικά στον δικό του κατάλογο προϊόντων, τις παραγγελίες, τα αποθέματα και τις επικοινωνίες του. Η αρχιτεκτονική αυτή εγγυάται τον διαχωρισμό της ιδιοκτησίας μέσω των `/vendor/*` endpoints και των ενσωματωμένων *ownership checks* στο backend.

Το **seller dashboard** προσφέρει μια πανοραμική εικόνα της εμπορικής δραστηριότητας, ενσωματώνοντας καθοδήγηση *onboarding* για την ολοκλήρωση των βασικών ρυθμίσεων (προϊόντα, στοιχεία καταστήματος, αποστολές). Παράλληλα, παρέχει άμεση πρόσβαση σε κρίσιμες ενέργειες, όπως η διαχείριση εκκρεμών *fulfillments*, η απόκριση σε reviews και η παρακολούθηση **analytics** σχετικά με την εξέλιξη των πωλήσεων σε

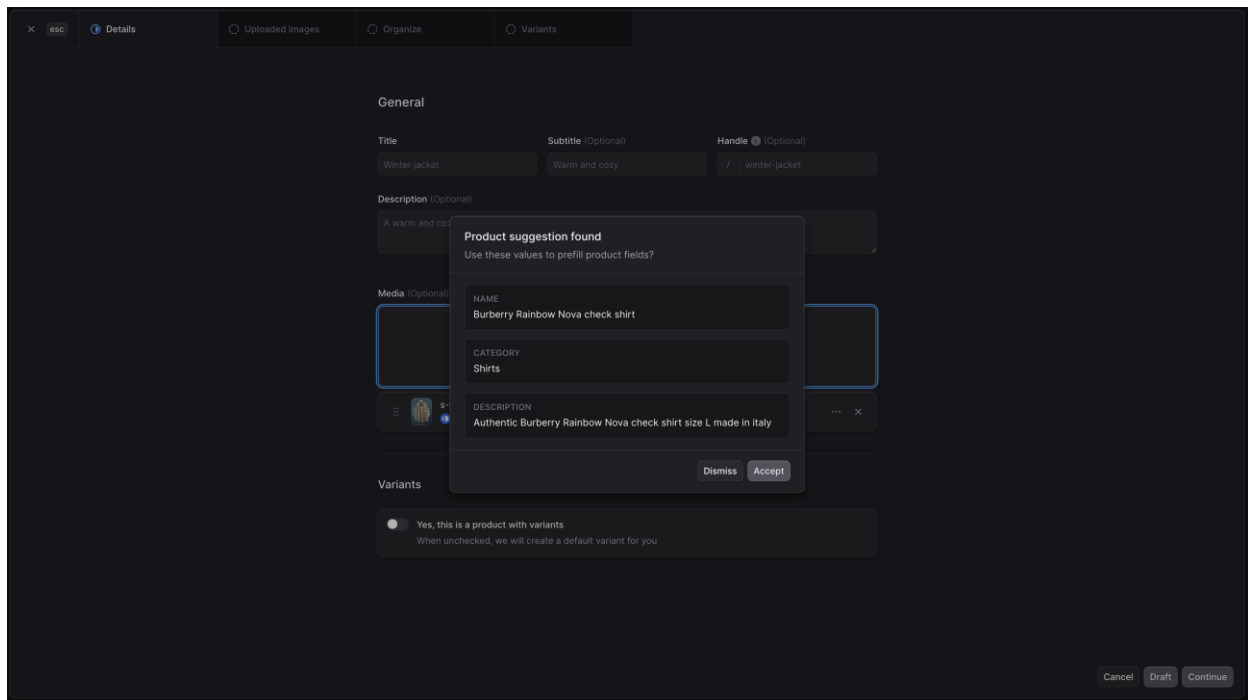
επιλεγμένα χρονικά διαστήματα.

Στο επίκεντρο της εφαρμογής βρίσκεται η **διαχείριση προϊόντων**. Ο πωλητής δύναται να καταχωρεί νέα είδη, να επεξεργάζεται παραλλαγές (variants) και να ορίζει τιμές, κατηγορίες και metadata. Η ροή δημιουργίας πραγματοποιείται μέσω του *vendor API*, επιτρέποντας την αποθήκευση ως *draft* ή την υποβολή ως *proposed*, υποστηρίζοντας έτσι τη διαδικασία **moderation** που απαιτείται για τη διασφάλιση της ποιότητας του marketplace.

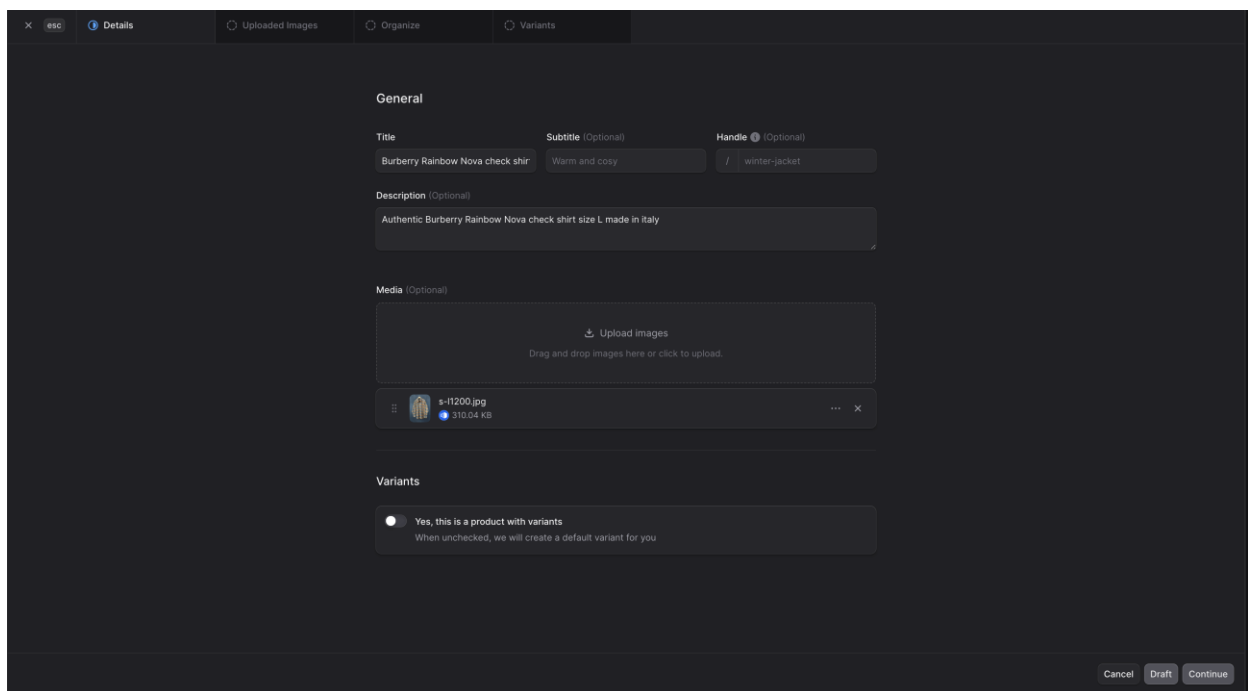
Επιπλέον, το panel ενσωματώνει λειτουργίες **inventory management**, επιτρέποντας τον έλεγχο των *stock levels* και των *reservations*. Λόγω της μοναδικότητας των μεταχειρισμένων ειδών μόδας, το σύστημα συνδέει άρρηκτα τα variants με το απόθεμα, αποτρέποντας την πώληση μη διαθέσιμων αντικειμένων.



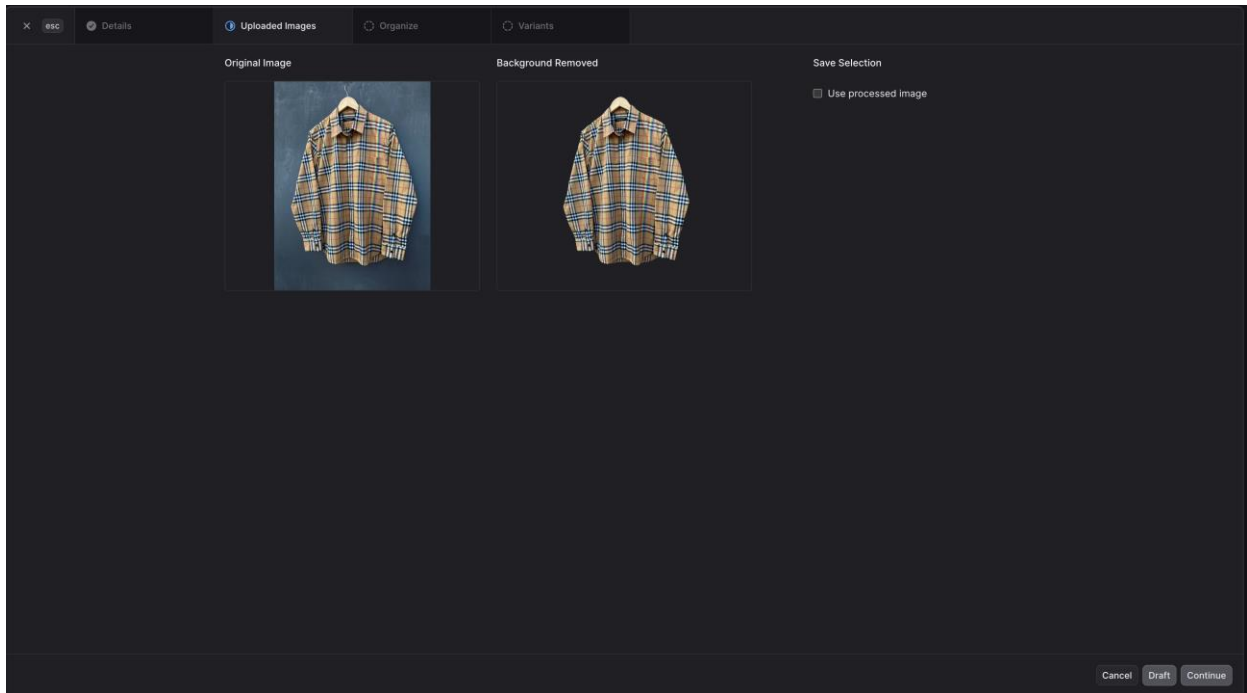
Εικόνα 16 – Οθόνη δημιουργίας προϊόντος – ανέβασμα εικόνας και εκτέλεση AI service: οπτική αναζήτηση και εμπλουτισμό πληροφοριών προϊόντος



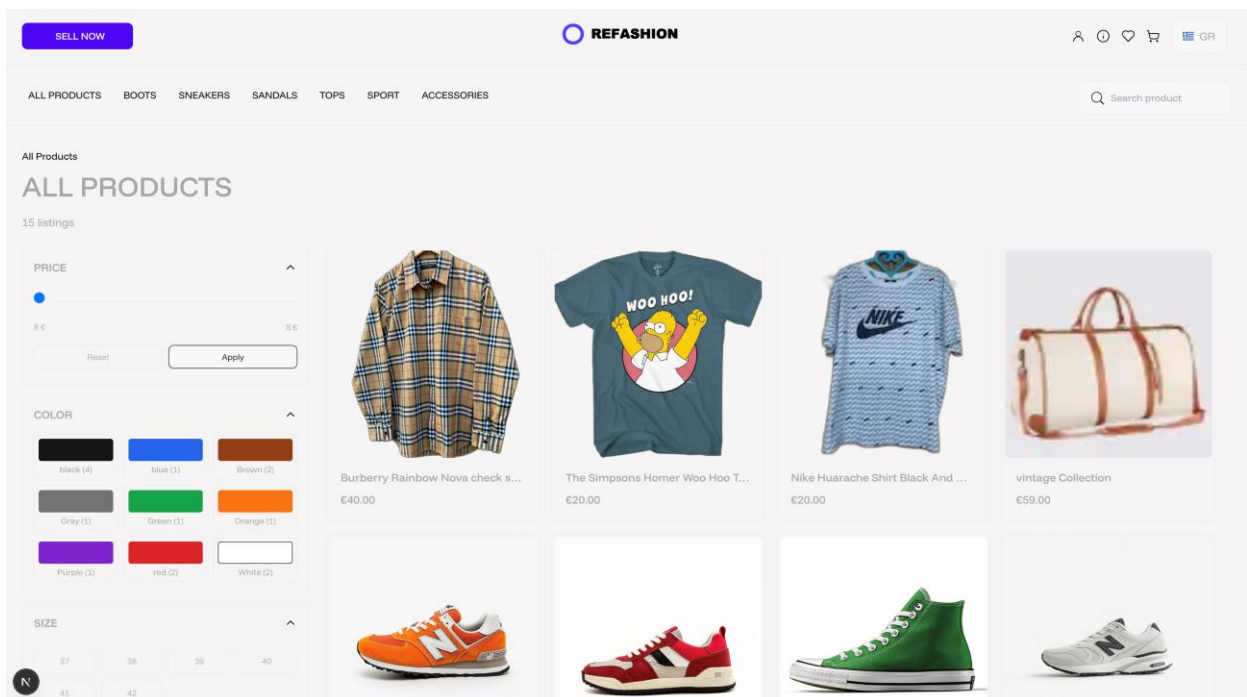
Εικόνα 17 – Οθόνη δημιουργίας προϊόντος – αποτελέσματα εμπλουτισμού πληροφοριών προϊόντος



Εικόνα 18 - Οθόνη δημιουργίας προϊόντος – αυτόματη συμπλήρωση πεδίων



Εικόνα 19 - Οθόνη δημιουργίας προϊόντος - εκτέλεση AI service: αφαίρεση background

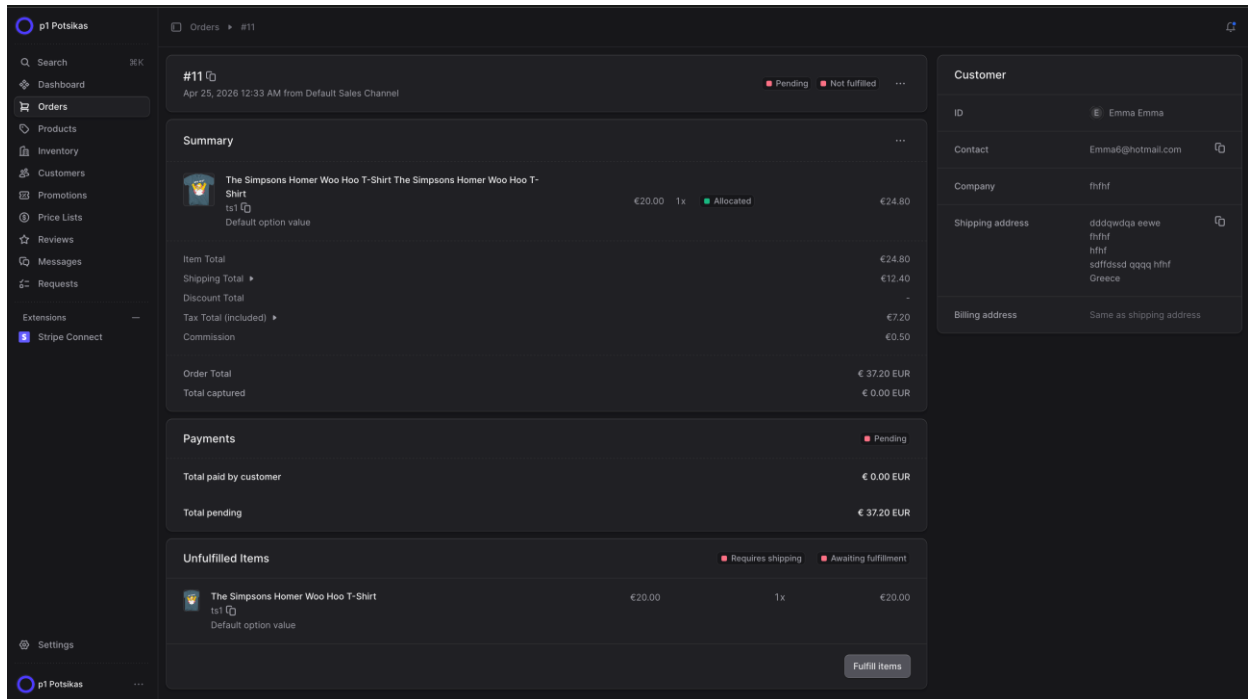


Εικόνα 20 - Οθόνη προϊόντων

Το υποσύστημα **αποστολών** επιτρέπει στον πωλητή να παραμετροποιεί τα *stock locations*, τους *fulfillment providers* και τα *shipping profiles* του. Μέσα από αυτές τις ρυθμίσεις, καθορίζεται ο τρόπος και η γεωγραφική κάλυψη της διανομής των προϊόντων προς τους τελικούς αγοραστές.

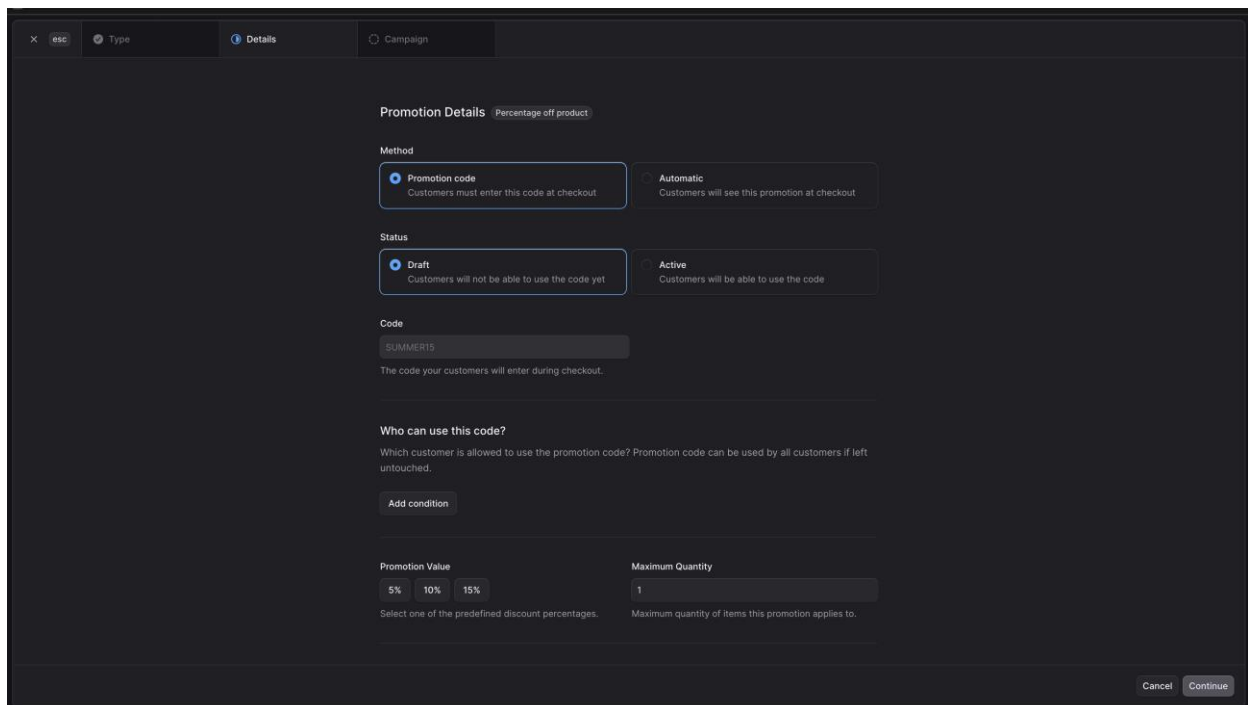
Η διαχείριση παραγγελιών είναι πλήρης, παρέχοντας πρόσβαση σε λεπτομερή στοιχεία για κάθε *order*, το καθεστώς πληρωμής και την κατάσταση αποστολής. Υποστηρίζονται όλες οι μετα-συναλλακτικές ενέργειες, συμπεριλαμβανομένων των **returns, claims, exchanges**

## και refunds.



Εικόνα 21 - Σελίδα λεπτομερειών παραγγελίας

Για την ενίσχυση της εμπορικής παρουσίας, παρέχονται εργαλεία **marketing και τιμολόγησης**, όπως η δημιουργία campaigns και price lists, επιτρέποντας την εφαρμογή στοχευμένων εκπτώσεων ή προωθητικών ενεργειών.



Εικόνα 22 - Σελίδα δημιουργίας campaigns

Η επικοινωνία και η εμπιστοσύνη διασφαλίζονται μέσω της ενότητας **reviews** και του συστήματος **messages** (με ενσωμάτωση **TalkJS**), διευκολύνοντας την άμεση

αλληλεπίδραση μεταξύ αγοραστών και πωλητών, στοιχείο κρίσιμο για τη second-hand αγορά.

Τέλος, το panel υποστηρίζει το **Stripe Connect**, επιτρέποντας τη διασύνδεση λογαριασμών πληρωμών για την είσπραξη των *royaltes* από τις πωλήσεις, με την κατάσταση σύνδεσης να αποτυπώνεται σε πραγματικό χρόνο.

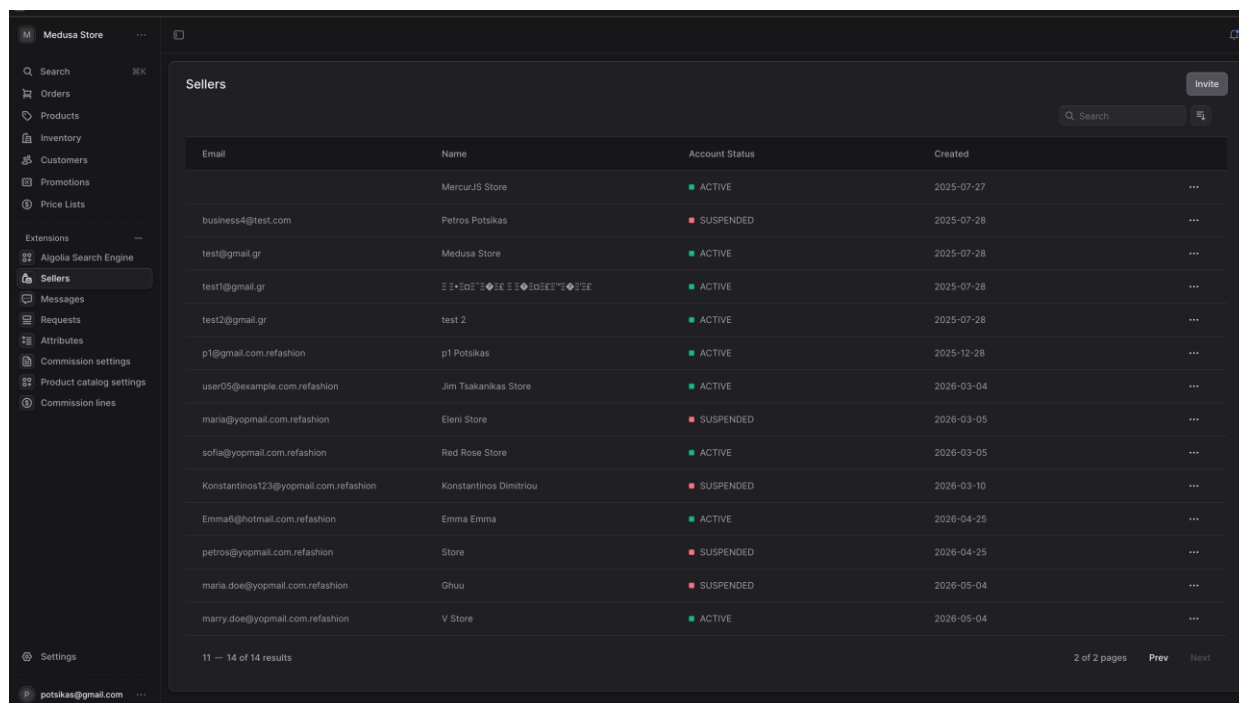
Συνολικά, το Seller Admin Panel λειτουργεί ως ένα ολοκληρωμένο επιχειρησιακό εργαλείο, το οποίο παρέχει πλήρη αυτονομία στη διαχείριση του καταστήματος χωρίς να εκθέτει global λειτουργίες του marketplace.

### 5.2.2 Medusa Admin Panel

Το **Medusa Admin Panel** αποτελεί το κεντρικό κέντρο ελέγχου και διακυβέρνησης της πλατφόρμας ReFashion. Σε αντίθεση με το vendor-specific panel, το Medusa Admin προορίζεται για τους administrators, προσφέροντας καθολική εποπτεία του marketplace. Εδράζεται στο backend project και αξιοποιεί τον πυρήνα του Medusa Admin, εμπλουτισμένο με custom **Mercur marketplace** λειτουργίες.

Η εφαρμογή αλληλεπιδρά με τα προστατευμένα `/admin/*` endpoints, επιτρέποντας στους διαχειριστές να εποπτεύουν πωλητές, αιτήματα έγκρισης, παραγγελίες, προμήθειες, καθώς και τα **περιβαλλοντικά στατιστικά** ολόκληρης της πλατφόρμας.

Κομβική λειτουργία αποτελεί η **διαχείριση πωλητών**. Ο administrator δύναται να αναζητά και να αναλύει στοιχεία των sellers, να παρακολουθεί τη δραστηριότητά τους και να προβαίνει σε ενέργειες *suspend* ή *activate*, διασφαλίζοντας την ασφάλεια και την ποιότητα των υπηρεσιών του marketplace.



Email	Name	Account Status	Created
	MercurJS Store	ACTIVE	2025-07-27
business4@test.com	Petros Potsikas	SUSPENDED	2025-07-28
test@gmail.gr	Medusa Store	ACTIVE	2025-07-28
test1@gmail.gr	test1@gmail.gr	ACTIVE	2025-07-28
test2@gmail.gr	test 2	ACTIVE	2025-07-28
p1@gmail.com.refashion	p1 Potsikas	ACTIVE	2025-12-28
user05@example.com.refashion	Jim Tsakanikas Store	ACTIVE	2026-03-04
maria@yopmail.com.refashion	Eleni Store	SUSPENDED	2026-03-05
sofia@yopmail.com.refashion	Red Rose Store	ACTIVE	2026-03-05
Konstantinos123@yopmail.com.refashion	Konstantinos Dimitriou	SUSPENDED	2026-03-10
Emma6@hotmail.com.refashion	Emma Emma	ACTIVE	2026-04-25
petros@yopmail.com.refashion	Store	SUSPENDED	2026-04-25
maria.doe@yopmail.com.refashion	Ghau	SUSPENDED	2026-05-04
marry.doe@yopmail.com.refashion	V Store	ACTIVE	2026-05-04

Εικόνα 23 - Σελίδα διαχείρισης πωλητών

Επιπλέον, το panel συντονίζει τη ροή έγκρισης νέων συμμετεχόντων. Μέσω της ενότητας *Seller creation requests*, ο admin εξετάζει τις αιτήσεις και αποφασίζει για την ενεργοποίηση των λογαριασμών, η οποία αποτελεί προϋπόθεση για την πρόσβαση στις λειτουργίες πωλητή.

Αντίστοιχα, εφαρμόζεται αυστηρό **product moderation**. Κάθε υποβολή νέου προϊόντος δημιουργεί ένα *product request*, το οποίο ο admin επικυρώνει ως προς την εγκυρότητα και την καταλληλότητα πριν τη δημοσίευσή του στο δημόσιο storefront. Αυτή η κεντρική λογική ελέγχου επεκτείνεται σε κατηγορίες, συλλογές (collections) και tags.

Το σύστημα ενσωματώνει τη διαχείριση των **marketplace commissions**. Μέσω των *commission rules*, ορίζονται τα ποσοστά προμήθειας ανά πωλητή ή κατηγορία, υποστηρίζοντας το επιχειρηματικό μοντέλο της πλατφόρμας.

Στις ρυθμίσεις του καταλόγου (*Product catalog settings*), ο administrator καθορίζει **global κανόνες λειτουργίας**, όπως η απαίτηση έγκρισης προϊόντων ή ο περιορισμός συγκεκριμένων εμπορικών πρακτικών, επηρεάζοντας τη συνολική συμπεριφορά του συστήματος.

Η οργάνωση των ειδών μόδας υποστηρίζεται από την ενότητα **Product Attributes**, όπου διαχειρίζονται χαρακτηριστικά όπως το brand, το υλικό και το χρώμα, ενισχύοντας την ακρίβεια στην αναζήτηση και το φιλτράρισμα.

Μέσω του **Impact Overview**, το admin panel παρέχει μετρήσιμα δεδομένα για τη συμβολή στην κυκλική οικονομία, όπως η συνολική εξοικονόμηση CO<sub>2</sub> και νερού, αναδεικνύοντας τον βιώσιμο χαρακτήρα του ReFashion.

Η ενότητα **Messages** (TalkJS) προσφέρει κεντρική εποπτεία των επικοινωνιών, επιτρέποντας την παρέμβαση της υποστήριξης όπου κρίνεται απαραίτητο.

Παράλληλα με τις marketplace επεκτάσεις, το panel διατηρεί όλες τις τυπικές δυνατότητες ενός *e-commerce backoffice* (πελάτες, περιοχές, φόροι, πληρωμές, επιστροφές), λειτουργώντας ως το απόλυτο εργαλείο διακυβέρνησης.

Συνοπτικά, το Medusa Admin Panel εγγυάται τη λειτουργική αρτιότητα, την ασφάλεια και την ποιοτική συνέπεια ολόκληρου του οικοσυστήματος ReFashion.

### 5.3. Σχεδιαστικές Προδιαγραφές (Design Specifications)

Πέρα από την τεχνική υλοποίηση και τις λειτουργικές δυνατότητες των διεπαφών χρήστη που παρουσιάστηκαν στις προηγούμενες ενότητες, η επιτυχία μιας πλατφόρμας όπως το REFASHION εξαρτάται σε σημαντικό βαθμό από τη **σχεδιαστική της ταυτότητα** και την **εμπειρία χρήστη** που προσφέρει. Οι σχεδιαστικές προδιαγραφές αποτυπώνουν το σύνολο των αρχών, των κανόνων και των οπτικών στοιχείων που διέπουν την παρουσίαση της πλατφόρμας σε όλα τα κανάλια αλληλεπίδρασης, εξασφαλίζοντας ένα **συνεκτικό και αναγνωρίσιμο** οπτικό προφίλ. Στις υποπαραγράφους που ακολουθούν, παρουσιάζονται οι κύριες σχεδιαστικές κατευθύνσεις του έργου, καλύπτοντας τη φιλοσοφία της οπτικής ταυτότητας, το σύστημα σχεδίασης που έχει υιοθετηθεί, τα πρότυπα διεπαφής που εφαρμόζονται και τη διαδικασία σχεδιασμού που ακολουθείται από την ομάδα ανάπτυξης.

#### 5.3.1 Σχεδιαστική Φιλοσοφία & Visual Identity

Η σχεδιαστική φιλοσοφία του REFASHION δεν αποτελεί απλώς ζήτημα αισθητικής, αλλά **στρατηγική επιλογή** που αντικατοπτρίζει τις αξίες και τους στόχους του έργου. Η πλατφόρμα δεν είναι μια συμβατική εφαρμογή ηλεκτρονικού εμπορίου, και η σχεδιαστική της προσέγγιση οφείλει να αποτυπώνει αυτή τη **διαφοροποίηση** σε κάθε επίπεδο της χρήστη-εμπειρίας. Η οπτική ταυτότητα του REFASHION καλείται να εκφράσει ταυτόχρονα τρεις διακριτές αλλά αλληλένδετες αξίες: τη **βιωσιμότητα και την περιβαλλοντική συνείδηση**, την **εμπιστοσύνη και τη διαφάνεια** σε ένα marketplace ομότιμων χρηστών, και τη **σύγχρονη τεχνολογική αρτιότητα** που πηγάζει από τις καινοτόμες λειτουργίες της

πλατφόρμας.

Η **κύρια σχεδιαστική κατευθυντήρια γραμμή** της πλατφόρμας ακολουθεί τις αρχές της σύγχρονης *minimalist* αισθητικής, με έμφαση στην καθαρότητα της οπτικής γλώσσας και στη λειτουργικότητα έναντι της οπτικής υπερφόρτωσης. Η προσέγγιση αυτή επιλέγεται ενσυνείδητα, καθώς ένα marketplace μεταχειρισμένων ειδών μόδας πρέπει να **αναδεικνύει τα ίδια τα προϊόντα** ως κεντρικό στοιχείο της εμπειρίας, χωρίς ο σχεδιασμός της πλατφόρμας να ανταγωνίζεται οπτικά τις φωτογραφίες και τις περιγραφές των αντικειμένων. Σε ένα οικοσύστημα όπου κάθε προϊόν είναι μοναδικό και έχει τη δική του ιστορία, η οπτική διεπαφή λειτουργεί ως **διακριτικό πλαίσιο** που υποστηρίζει την ανακάλυψη χωρίς να την επισκιάζει.

Η **χρωματική παλέτα** της πλατφόρμας οικοδομείται γύρω από ουδέτερες αποχρώσεις, οι οποίες προσφέρουν το κατάλληλο υπόβαθρο για την προβολή του ποικιλόμορφου υλικού των πωλητών. Η επιλογή αυτή εξυπηρετεί επίσης την **απομάκρυνση από την οπτική γλώσσα της fast fashion**, που τυπικά χαρακτηρίζεται από έντονα χρώματα και πιεστικά καλέσματα δράσης. Η ουδέτερη παλέτα του REFASHION αντικατοπτρίζει μια **πιο ώριμη και στοχαστική** αγοραστική εμπειρία, εναρμονισμένη με τη φιλοσοφία της κυκλικής οικονομίας. Πέρα από τις βασικές αποχρώσεις, ορισμένα **τονικά accent χρώματα** χρησιμοποιούνται με φειδώ για την επισήμανση κρίσιμων στοιχείων της εμπειρίας - των δεικτών βιωσιμότητας, των ReWear Tokens, των επιτευγμάτων gamification - λειτουργώντας ως **οπτικοί δείκτες αξίας** που τραβούν αμέσως την προσοχή του χρήστη χωρίς να κατακλύζουν τη συνολική εικόνα.

Η **τυπογραφική προσέγγιση** ακολουθεί την ίδια φιλοσοφία της καθαρότητας και της λειτουργικής αισθητικής. Επιλέγονται σύγχρονες, ευανάγνωστες γραμματοσειρές που εξασφαλίζουν εξαιρετική αναγνωσιμότητα τόσο σε desktop όσο και σε κινητές συσκευές, υποστηρίζοντας πλήρως την ελληνική και τη λατινική γραφή. Η ιεραρχία των κειμένων διαμορφώνεται με σαφή διαβάθμιση μεγεθών και βαρών, ώστε ο χρήστης να αντιλαμβάνεται εύκολα τη **σημασιολογική σχέση** μεταξύ τίτλων, υποτίτλων, σωματικού κειμένου και βοηθητικών πληροφοριών χωρίς να χρειάζεται να καταβάλει νοητική προσπάθεια. Η συνεπής εφαρμογή της τυπογραφικής ιεραρχίας σε όλα τα frontend περιβάλλοντα αποτελεί κρίσιμο στοιχείο για την **αντιλαμβανόμενη ποιότητα** της πλατφόρμας.

Ένα **διακριτικό σχεδιαστικό στοιχείο** του REFASHION αποτελεί η ανάγκη οπτικής αναπαράστασης των **καινοτόμων λειτουργιών** της πλατφόρμας, οι οποίες δεν συναντώνται σε συμβατικές εφαρμογές e-commerce. Ο Eco-Impact Score, τα ReWear Tokens και τα στοιχεία gamification απαιτούν εξειδικευμένη σχεδιαστική αντιμετώπιση, ώστε να **γίνουν αμέσως αντιληπτά** ως μοναδικά χαρακτηριστικά του REFASHION και να επικοινωνήσουν αποτελεσματικά την αξία τους στους χρήστες. Για τον σκοπό αυτό, σχεδιάζονται εξειδικευμένα **οπτικά γλωσσικά στοιχεία** - εικονογραφικά μοτίβα, χρωματικοί κωδικοί και διακριτά visual components - που λειτουργούν ως **σήματα κατατεθέντα** της πλατφόρμας και διαφοροποιούν την οπτική εμπειρία από αυτή των ανταγωνιστικών οικοσυστημάτων.

Η **εικονογραφική γλώσσα** της πλατφόρμας ακολουθεί ομοιόμορφες κατευθύνσεις σε όλα τα κανάλια. Επιλέγονται γραμμικά εικονίδια (line icons) με συνεπές πάχος γραμμής και ομοιόμορφο στυλ, αντί για γεμάτα ή υπερβολικά διακοσμητικά εικονίδια, εξυπηρετώντας τη γενικότερη *minimalist* φιλοσοφία. Η εικονογράφηση χρησιμοποιείται με φειδώ και πάντοτε για **λειτουργικούς σκοπούς** - υποδεικνύοντας ενέργειες, καταστάσεις, κατηγορίες - ποτέ ως αμιγώς διακοσμητικό στοιχείο. Όπου χρειάζεται πιο εκφραστική απεικόνιση, για

παράδειγμα στη σήμανση κατηγοριών προϊόντων ή στα badges του gamification, αξιοποιούνται συλιζαρισμένες εικονογραφήσεις που διατηρούν τη συνέπεια με τη συνολική οπτική γλώσσα.

Όσον αφορά τη **χρήση εικόνας**, ο σχεδιασμός της πλατφόρμας αναγνωρίζει ότι οι **φωτογραφίες των προϊόντων** αποτελούν τα κυριότερα οπτικά στοιχεία της εμπειρίας. Η πλατφόρμα σχεδιάζεται έτσι ώστε να **τις αναδεικνύει**, με μεγάλες περιοχές παρουσίασης, καθαρά πλαίσια και επαρκή λευκό χώρο γύρω από κάθε εικόνα. Παράλληλα, η ενσωμάτωση του Image Analysis Service (ενότητα 2.2.5.5) που αυτοματοποιεί την αφαίρεση φόντου από τις φωτογραφίες των πωλητών συμβάλλει σημαντικά στην **οπτική ομοιογένεια του καταλόγου**, εξασφαλίζοντας ότι ακόμη και φωτογραφίες λαμβανόμενες σε ποικίλα περιβάλλοντα παρουσιάζονται με συνεκτικό τρόπο στην πλατφόρμα.

Η συνολική σχεδιαστική φιλοσοφία του REFASHION αποτελεί επομένως ένα **προσεκτικό μείγμα** σύγχρονης αισθητικής, λειτουργικής σαφήνειας και ηθικής έκφρασης. Δεν επιδιώκεται η οπτική εντυπωσιακότητα ως αυτοσκοπός, αλλά η **σχεδιαστική επικοινωνία της ταυτότητας** μιας πλατφόρμας που στέκεται διαφορετικά απέναντι στη μόδα, στην τεχνολογία και στους χρήστες της. Η αναλυτική παρουσίαση των επιμέρους τεχνικών στοιχείων που υλοποιούν αυτή τη φιλοσοφία - design tokens, components, layouts - παρουσιάζεται στις επόμενες υποπαραγράφους.

### 5.3.2 Design System & Component Library

Η μετάφραση της σχεδιαστικής φιλοσοφίας που παρουσιάστηκε στην προηγούμενη υποπαραγραφή σε **απτή, εφαρμόσιμη πραγματικότητα** απαιτεί έναν συστηματικό και τεκμηριωμένο μηχανισμό: ένα **Design System** που λειτουργεί ως η **κοινή γλώσσα σχεδιασμού** μεταξύ όλων των εμπλεκόμενων ομάδων - designers, frontend developers, backend developers, project managers - και ως η **ενιαία πηγή αλήθειας** για τα οπτικά και αλληλεπιδραστικά πρότυπα της πλατφόρμας. Στο REFASHION, η συγκρότηση ενός τέτοιου συστήματος αποτελεί κρίσιμη στρατηγική επιλογή, καθώς η πλατφόρμα συγκροτείται από πολλαπλά frontend περιβάλλοντα που αναπτύσσονται από διαφορετικές ομάδες (το web storefront, το seller panel και το admin panel αναπτύσσονται από την ομάδα του ΠΑ.ΠΕΛ, ενώ η εφαρμογή κινητού από την Cyclefi), και η οπτική και εμπειρική συνέπεια μεταξύ τους απαιτεί την ύπαρξη κοινών σχεδιαστικών αναφορών.

#### Φιλοσοφία και Δομή του Design System

Το Design System του REFASHION ακολουθεί τη σύγχρονη μεθοδολογία οργάνωσης σε **τρία ιεραρχημένα επίπεδα αφαίρεσης**: τα **Design Tokens** ως θεμελιώδη ατομικά στοιχεία, τα **Components** ως επαναχρησιμοποιήσιμα δομικά μπλοκ διεπαφής, και τα **Patterns** ως σύνθετα μοτίβα αλληλεπίδρασης. Η ιεραρχία αυτή εξασφαλίζει ότι **κάθε αλλαγή στα θεμελιώδη στοιχεία** προπαγάγεται αυτόματα σε όλα τα παράγωγα επίπεδα, διατηρώντας τη συνεκτικότητα του συστήματος χωρίς διπλασιασμό προσπάθειας.

Η σχεδιαστική φιλοσοφία πίσω από το Design System ακολουθεί την προσέγγιση του **atomic design**, σύμφωνα με την οποία τα οπτικά στοιχεία οργανώνονται σε αυξανόμενες κλίμακες πολυπλοκότητας. Από τα μικρότερα και πιο θεμελιώδη στοιχεία (atoms - π.χ. ένα κουμπί, ένα input field) συντίθενται μεγαλύτερες ενότητες (molecules - π.χ. μια κάρτα προϊόντος), που με τη σειρά τους συγκροτούν πιο σύνθετες δομές (organisms - π.χ. μια λίστα προϊόντων με φίλτρα), και τελικά ολοκληρωμένες σελίδες. Η προσέγγιση αυτή επιτρέπει την **επαναχρησιμοποίηση** των κατώτερων επιπέδων σε πολλαπλά ανώτερα πλαίσια, και εξασφαλίζει ότι μια αλλαγή σε ένα atom (π.χ. στο border-radius των κουμπιών) εφαρμόζεται αυτόματα σε όλες τις σύνθετες δομές που το αξιοποιούν.

## Design Tokens

Τα **Design Tokens** αποτελούν το **Θεμέλιο** του Design System και ορίζουν τις βασικές οπτικές αξίες της πλατφόρμας με τη μορφή **ονοματισμένων μεταβλητών**. Αντί κάθε σχεδιαστικής απόφασης να εκφράζεται με σκληρά κωδικοποιημένες τιμές (π.χ. ένα συγκεκριμένο hex χρώμα ή μια συγκεκριμένη τιμή σε pixels), τα tokens παρέχουν **σημασιολογικά ονόματα** που εκφράζουν τον **σκοπό** της κάθε τιμής. Με τον τρόπο αυτό, ένα κουμπί δεν αναφέρεται στο χρώμα "πράσινο", αλλά στο token **color-action-primary**, το οποίο μπορεί κεντρικά να αντιστοιχίζεται σε διαφορετική τιμή χωρίς επιπτώσεις στον υπόλοιπο σχεδιαστικό κώδικα.

Τα Design Tokens του REFASHION οργανώνονται σε **έξι βασικές κατηγορίες**. Στην κατηγορία **Color tokens** ορίζονται οι ονοματισμένες αξίες της χρωματικής παλέτας, με σαφή διάκριση μεταξύ ουδέτερων χρωμάτων (που χρησιμοποιούνται για φόντα και επιφάνειες), accent χρωμάτων (που επισημαίνουν τις καινοτόμες λειτουργίες όπως Eco-Impact, ReWear Tokens, gamification), σημασιολογικών χρωμάτων (success, warning, error, info) και χρωμάτων κειμένου (primary text, secondary text, disabled text). Η οργάνωση αυτή επιτρέπει στους σχεδιαστές και τους developers να επιλέγουν χρώματα βάσει σκοπού αντί βάσει εμφάνισης, εξασφαλίζοντας **σημασιολογική συνέπεια** σε όλη την πλατφόρμα.

Στην κατηγορία **Typography tokens** ορίζονται οι ονοματισμένες αξίες της τυπογραφικής ιεραρχίας. Κάθε token εκφράζει μια σαφή σημασιολογική χρήση (π.χ. **text-heading-large**, **text-body-default**, **text-caption-small**) και αντιστοιχίζεται σε συγκεκριμένο συνδυασμό font-family, font-size, font-weight, line-height και letter-spacing. Η προσέγγιση αυτή εξασφαλίζει ότι όλα τα κείμενα της πλατφόρμας **ακολουθούν μια περιορισμένη και τεκμηριωμένη ιεραρχία**, αποτρέποντας τη χρήση αυθαίρετων τυπογραφικών αξιών που θα υπονόμωσαν τη συνεκτικότητα του συστήματος.

Η κατηγορία **Spacing tokens** ορίζει τις τυποποιημένες αξίες για τα κενά μεταξύ στοιχείων, τα paddings και τα margins. Στο REFASHION εφαρμόζεται μια **κλίμακα βασισμένη σε 4px ή 8px multiples** (π.χ. xs, sm, md, lg, xl), η οποία εξασφαλίζει τη **ρυθμική οπτική αρμονία** σε όλη την πλατφόρμα και αποτρέπει την εμφάνιση αυθαίρετων αποστάσεων που θα διασπούσαν την οπτική συνοχή. Η αυστηρή τήρηση αυτής της κλίμακας αποτελεί έναν από τους πιο αναγνωρίσιμους - αν και υποσυνείδητους - δείκτες της σχεδιαστικής ποιότητας μιας πλατφόρμας.

Στην κατηγορία **Sizing tokens** ορίζονται οι τυποποιημένες διαστάσεις για βασικά οπτικά στοιχεία (π.χ. ύψος κουμπιών, μέγεθος εικονιδίων, διαστάσεις avatars), εξασφαλίζοντας την οπτική συμβατότητα μεταξύ διαφορετικών components. Η κατηγορία **Border & Radius tokens** ορίζει τις αξίες για περιγράμματα και γωνίες, με ομοιόμορφες ακτίνες καμπύλωσης που δίνουν στην πλατφόρμα μια **σταθερή οπτική προσωπικότητα** (π.χ. ήπιες καμπύλες για μια φιλική αίσθηση ή ευθείες γωνίες για πιο επίσημο τόνο). Τέλος, η κατηγορία **Shadow & Elevation tokens** ορίζει τα επίπεδα οπτικής βαθύτητας μέσω σκιάσεων, υποστηρίζοντας την αντιλαμβανόμενη ιεραρχία μεταξύ επιφανειών (π.χ. μεταξύ ενός κανονικού στοιχείου και ενός dropdown menu που "πέταει" από πάνω του).

## Component Library

Πάνω στα Design Tokens οικοδομείται η **Component Library** του REFASHION - μια συλλογή από **επαναχρησιμοποιήσιμα οπτικά στοιχεία διεπαφής** που υλοποιούν τα κοινά μοτίβα αλληλεπίδρασης της πλατφόρμας. Η ύπαρξη μιας ενιαίας component library αποτελεί τη **μεγαλύτερη πρακτική εφαρμογή** του design system, καθώς εξασφαλίζει ότι όπου εμφανίζεται μια κάρτα προϊόντος, ένα κουμπί προσθήκης στο καλάθι, μια ένδειξη Eco-Impact ή ένα badge gamification, αυτά παρουσιάζονται **πανομοιότυπα** σε όλα τα

frontend περιβάλλοντα.

Τα **βασικά components** της βιβλιοθήκης οργανώνονται σε λογικές ομάδες ανάλογα με τον σκοπό τους. Τα **input controls** (κουμπιά, text inputs, dropdowns, checkboxes, radio buttons, toggles) αποτελούν τα στοιχεία αλληλεπίδρασης χρήστη και υλοποιούνται με συνεπή συμπεριφορά σε όλες τις καταστάσεις (default, hover, focus, active, disabled, error). Τα **display components** (κάρτες, λίστες, πίνακες, badges, tags, avatars) ευθύνονται για την παρουσίαση πληροφορίας και ακολουθούν συνεπή οργάνωση εσωτερικού περιεχομένου. Τα **feedback components** (alerts, toasts, modals, dialogs, progress indicators) υποστηρίζουν την επικοινωνία καταστάσεων και ανατροφοδότησης προς τον χρήστη. Τέλος, τα **navigation components** (menus, breadcrumbs, tabs, pagination) διαμορφώνουν τη δομή πλοήγησης της πλατφόρμας.

Πέρα από τα γενικά components, η Component Library του REFASHION περιλαμβάνει και ένα σύνολο **εξειδικευμένων components** που υλοποιούν τις καινοτόμες λειτουργίες της πλατφόρμας. Η **κάρτα προϊόντος (Product Card)** αποτελεί ίσως το πιο εμβληματικό από αυτά: σχεδιάζεται για να αναδεικνύει με ισορροπημένο τρόπο τη φωτογραφία του αντικειμένου, τα βασικά εμπορικά στοιχεία (τίτλος, τιμή, μέγεθος, κατάσταση), και τους **διακριτικούς δείκτες REFASHION** όπως το Eco-Impact Score και την ένδειξη εναλλακτικής συναλλαγής (donation/swap), χωρίς να γίνεται οπτικά υπερφορτωμένη. Το **Eco-Impact indicator** αποτελεί ένα δεύτερο εξειδικευμένο component που εμφανίζεται σε πολλαπλές παραλλαγές (compact για listings, expanded για σελίδες προϊόντος, summary για profile dashboards), παρέχοντας τη μετρήσιμη οπτικοποίηση του περιβαλλοντικού οφέλους κάθε προϊόντος. Αντίστοιχα εξειδικευμένα στοιχεία αποτελούν το **ReWear Token balance widget** που εμφανίζει την κατάσταση των ψηφιακών ανταμοιβών του χρήστη, καθώς και τα **gamification components** (badges, level indicators, progress bars για achievements) που αξιοποιούνται σε όλο το Stats Module και τα profile screens.

### Καταστάσεις και Παραλλαγές των Components

Κάθε component στη βιβλιοθήκη του REFASHION σχεδιάζεται με την αρχή της **πληρότητας των καταστάσεων (state completeness)**. Αυτό σημαίνει ότι για κάθε component δεν τεκμηριώνεται μόνο η **default εμφάνιση** του, αλλά **όλες οι πιθανές καταστάσεις** στις οποίες μπορεί να βρεθεί κατά τη χρήση: hover, focus, active, disabled, loading, error, empty, καθώς και τυχόν παραλλαγές μεγέθους (small, medium, large) ή σημασιολογικής χρήσης (primary, secondary, tertiary). Η συστηματική αυτή κάλυψη εξασφαλίζει ότι **καμία περίπτωση χρήσης δεν αφήνεται απροσχεδίαστη**, αποτρέποντας την εμφάνιση αυθαίρετων ή ασυνεπών συμπεριφορών κατά τη φάση υλοποίησης.

Παράλληλα, ιδιαίτερη προσοχή δίνεται στη συμπεριφορά των components σε **οριακές καταστάσεις δεδομένων**. Πώς εμφανίζεται μια λίστα προϊόντων όταν δεν υπάρχουν αποτελέσματα; Πώς συμπεριφέρεται ένα input όταν εισάγεται μη έγκυρη τιμή; Πώς απεικονίζεται ένα avatar όταν δεν έχει εικόνα; Πώς προβάλλεται μια κάρτα προϊόντος όταν λείπουν πληροφορίες όπως ο Eco-Impact Score (π.χ. σε εκκρεμή AI ανάλυση); Όλες αυτές οι περιπτώσεις αντιμετωπίζονται με σχεδιαστική πρόνοια, εξασφαλίζοντας ότι η πλατφόρμα παραμένει **οπτικά ομοιόμορφη και λειτουργικά κατανοητή** ακόμη και σε μη ιδανικές συνθήκες δεδομένων.

### Συνέπεια Σχεδιασμού και Υλοποίησης

Ένα από τα **κρισιμότερα ζητήματα** στην εφαρμογή ενός Design System αποτελεί η **διασφάλιση της συνέπειας** μεταξύ του σχεδιασμού (όπως αυτός εκφράζεται στα σχεδιαστικά εργαλεία) και της υλοποίησης (όπως αυτή προκύπτει στον κώδικα). Στο REFASHION, η συνέπεια αυτή διασφαλίζεται μέσω της **κωδικοποιημένης μετάφρασης** των

Design Tokens σε τιμές που μπορούν να αξιοποιηθούν απευθείας από τα frontend frameworks. Συγκεκριμένα, τα tokens εκφράζονται ως CSS variables στις web εφαρμογές (storefront, seller panel, admin panel), και ως αντίστοιχες δομές στις εφαρμογές κινητού. Η προσέγγιση αυτή εξασφαλίζει ότι ο σχεδιαστής και ο developer **εργάζονται με τις ίδιες σημασιολογικές αξίες**, και ότι οποιαδήποτε αλλαγή στα tokens εφαρμόζεται αυτόματα σε όλα τα κανάλια.

Επιπρόσθετα, τα components της βιβλιοθήκης υλοποιούνται ως **πραγματικός κώδικας** που μπορεί να αξιοποιηθεί από τις διάφορες εφαρμογές της πλατφόρμας. Για το οικοσύστημα του Medusa Admin Panel και του Seller Panel, αξιοποιείται η **Medusa UI library** που παρέχει έτοιμα React components ευθυγραμμισμένα με τη σχεδιαστική γλώσσα του Medusa, εμπλουτισμένα με τα **REFASHION custom components** που υλοποιούν τα εξειδικευμένα στοιχεία της πλατφόρμας. Για το web storefront, αναπτύσσεται μια **εξατομικευμένη συλλογή components** που υλοποιεί τη σχεδιαστική γλώσσα του REFASHION σε React/Next.js. Για την εφαρμογή κινητού, η Cyclefy αναπτύσσει την **αντίστοιχη συλλογή σε React Native**, αξιοποιώντας τα ίδια design tokens για να εξασφαλιστεί η οπτική συνέπεια με τα web περιβάλλοντα.

#### **Διαχείριση Αλληλεπίδρασης και Καταστάσεων UI**

Πέρα από τα στατικά οπτικά χαρακτηριστικά, το Design System του REFASHION ορίζει επίσης τα **πρότυπα αλληλεπίδρασης** που πρέπει να ακολουθούν τα components. Αυτά περιλαμβάνουν τις **animations και transitions** (διάρκεια, καμπύλες χρονισμού, ποια στοιχεία πρέπει να εμψυχωθούν), τα πρότυπα **focus management** (πώς διαχειρίζεται η εστίαση κατά την πλοήγηση με πληκτρολόγιο, ποια στοιχεία είναι εστίασιμα), τα πρότυπα **error handling** (πώς και πού εμφανίζονται τα μηνύματα σφάλματος), και τα πρότυπα **loading states** (πώς ενημερώνεται ο χρήστης για ασύγχρονες λειτουργίες όπως οι κλήσεις στις AI υπηρεσίες). Η τεκμηρίωση αυτών των προτύπων εξασφαλίζει ότι η αλληλεπίδραση με την πλατφόρμα είναι **προβλέψιμη και ομαλή**, ανεξάρτητα από το συγκεκριμένο περιβάλλον στο οποίο βρίσκεται ο χρήστης.

#### **Προσβασιμότητα ως Αναπόσπαστο Στοιχείο**

Η **προσβασιμότητα (accessibility)** ενσωματώνεται από την αρχή στον σχεδιασμό κάθε component, και όχι ως εκ των υστέρων προσθήκη. Όλα τα components σχεδιάζονται με τήρηση των βασικών αρχών των οδηγιών **WCAG (Web Content Accessibility Guidelines)**: επαρκής χρωματική αντίθεση μεταξύ κειμένου και φόντου, σαφής σήμανση όλων των διαδραστικών στοιχείων, υποστήριξη πλοήγησης με πληκτρολόγιο, κατάλληλες ετικέτες ARIA για screen readers, και διασφάλιση ότι κανένα στοιχείο πληροφορίας δεν εξαρτάται αποκλειστικά από το χρώμα για τη μετάδοσή του. Η προσβασιμότητα δεν αντιμετωπίζεται ως κανονιστική απαίτηση, αλλά ως **έκφραση της ηθικής διάστασης** του REFASHION ως πλατφόρμας ανοιχτής σε όλους τους χρήστες, ανεξάρτητα από τυχόν περιορισμούς που αντιμετωπίζουν.

Συνολικά, το Design System και η Component Library του REFASHION αποτελούν τους θεμελιώδεις **σχεδιαστικούς πυλώνες** πάνω στους οποίους οικοδομείται η συνεκτική εμπειρία χρήστη σε όλα τα κανάλια της πλατφόρμας. Η συστηματική και τεκμηριωμένη οργάνωσή τους εξασφαλίζει ότι η σχεδιαστική φιλοσοφία του έργου μεταφράζεται σε **απτή πραγματικότητα** σε κάθε διεπαφή, από τη μικρότερη ένδειξη μέχρι την πιο σύνθετη σελίδα.

#### **5.3.3 Πρότυπα Διεπαφής & Layout Patterns**

Πέρα από τα ατομικά οπτικά στοιχεία και τη βιβλιοθήκη components που παρουσιάστηκαν στην προηγούμενη υποπαράγραφο, η σχεδιαστική προδιαγραφή του REFASHION

εκτείνεται σε ένα **ανώτερο επίπεδο οργάνωσης**: τα **πρότυπα διεπαφής (interface patterns)** και τα **μοτίβα διάταξης (layout patterns)** που καθορίζουν πώς συνδυάζονται τα επιμέρους στοιχεία για να συγκροτήσουν ολοκληρωμένες οθόνες και εμπειρίες χρήστη. Τα πρότυπα αυτά αποτελούν τη **γέφυρα μεταξύ του Design System και των τελικών σελίδων** της πλατφόρμας, εξασφαλίζοντας ότι η οπτική συνεκτικότητα διατηρείται όχι μόνο στο επίπεδο των μικρών components, αλλά και στις σύνθετες δομές όπου ο χρήστης πραγματικά αλληλεπιδρά με την πλατφόρμα.

### Καθολική Δομή Σελίδων

Όλες οι σελίδες της πλατφόρμας REFASHION ακολουθούν μια **κοινή θεμελιώδη δομή** που εξασφαλίζει την προβλεψιμότητα της εμπειρίας και την ευκολία πλοήγησης. Η δομή αυτή αρθρώνεται σε **τρεις βασικές περιοχές**: το **header** που φιλοξενεί το logo της πλατφόρμας, τη βασική πλοήγηση, την κεντρική μπάρα αναζήτησης και τις προσβάσεις σε λογαριασμό, καλάθι και ειδοποιήσεις · το **κύριο περιεχόμενο (main content area)** όπου παρουσιάζεται η εξειδικευμένη πληροφορία κάθε σελίδας · και το **footer** που φιλοξενεί τις δευτερεύουσες πληροφορίες, τους νομικούς συνδέσμους, τα στοιχεία επικοινωνίας και τις πληροφορίες σχετικά με το έργο και τη φιλοσοφία του REFASHION.

Στις mobile εμφανίσεις, η δομή αυτή προσαρμόζεται κατάλληλα: το header συμπυκνώνεται σε **πιο συμπαγή μορφή** που διατηρεί τα κρίσιμα στοιχεία, η βασική πλοήγηση μετακινείται σε **bottom tab bar** ή σε hamburger menu ανάλογα με τη σημασία των επιλογών, και το footer απλοποιείται σημαντικά. Η ίδια λογική ακολουθείται και στην εφαρμογή κινητού, όπου η bottom navigation αποτελεί τον κύριο μηχανισμό πλοήγησης μεταξύ των βασικών εννοιών (Home, Search, Sell, Profile, Stats).

### Responsive Σχεδιασμός και Breakpoints

Ο σχεδιασμός των διεπαφών του REFASHION αξιοποιεί τη μεθοδολογία του **responsive design**, εξασφαλίζοντας ότι η εμπειρία προσαρμόζεται κατάλληλα σε όλο το φάσμα των συσκευών - από μικρές οθόνες κινητών έως μεγάλες οθόνες desktop. Η προσαρμογή αυτή βασίζεται σε **σταθερά breakpoints** που έχουν οριστεί στο επίπεδο των Design Tokens και εφαρμόζονται ομοιόμορφα σε όλα τα frontend περιβάλλοντα.

Σε κάθε breakpoint, ορισμένα οπτικά χαρακτηριστικά **προσαρμόζονται συστηματικά**: το πλάτος του κύριου περιεχομένου, ο αριθμός των στηλών σε grid layouts προϊόντων, η εμφάνιση ή απόκρυψη δευτερευόντων στοιχείων, η αναδιάταξη στοιχείων που σε στενότερες οθόνες πρέπει να εμφανίζονται κάθετα αντί οριζόντια, και η προσαρμογή των μεγεθών κειμένου για την εξασφάλιση αναγνωσιμότητας. Η αρχή που διέπει αυτές τις προσαρμογές είναι η **διατήρηση της λειτουργικότητας** σε κάθε μέγεθος οθόνης χωρίς θυσία της σχεδιαστικής ποιότητας, αναγνωρίζοντας ότι κάθε συσκευή εξυπηρετεί διαφορετικά σενάρια χρήσης.

### Πρότυπα Διάταξης Καταλόγου

Η παρουσίαση του καταλόγου προϊόντων αποτελεί ένα από τα **πιο κρίσιμα πρότυπα διάταξης** της πλατφόρμας, καθώς αντιπροσωπεύει την κύρια εμπειρία ανακάλυψης του χρήστη. Στο REFASHION εφαρμόζεται το πρότυπο του **grid layout**, με τις κάρτες προϊόντων διατεταγμένες σε στήλες με σταθερό gap μεταξύ τους. Ο αριθμός των στηλών προσαρμόζεται δυναμικά ανάλογα με το μέγεθος της οθόνης - από δύο στήλες σε mobile, έως τρεις ή τέσσερις σε tablet, και πέντε ή έξι σε ευρύχωρα desktop. Η σχεδιαστική επιλογή αυτή εξασφαλίζει ότι κάθε καρτέλα προϊόντος διατηρεί την **αναγκαία οπτική παρουσία** χωρίς να χάνεται μέσα σε υπερβολικά πυκνή προβολή.

Παράλληλα με το grid layout, η αρχιτεκτονική προβλέπει εναλλακτική εμφάνιση σε **list**

**layout** για περιπτώσεις όπου η περιεκτικότερη παρουσίαση κάθε προϊόντος εξυπηρετεί τη χρήση (π.χ. σε εμπορικές σελίδες με λεπτομερή στοιχεία). Η εναλλαγή μεταξύ των δύο εμφανίσεων ελέγχεται από τον χρήστη μέσω εξειδικευμένου control, με αυτόματη απομνημόνευση της επιλογής για επόμενες επισκέψεις.

Ένα **καθοριστικό σχεδιαστικό μοτίβο** για τον κατάλογο αποτελεί η οργάνωση των **φίλτρων αναζήτησης**. Σε ευρύχωρες desktop εμφανίσεις, τα φίλτρα προβάλλονται σε μόνιμη **πλαϊνή στήλη (sidebar)** που επιτρέπει στον χρήστη να βλέπει συνεχώς τις διαθέσιμες επιλογές. Σε mobile εμφανίσεις, τα φίλτρα μετακινούνται σε **modal panel ή bottom sheet** που ανοίγει με ειδικό κουμπί, εξοικονομώντας πολύτιμο χώρο στην οθόνη. Σε κάθε εμφάνιση, τα ενεργά φίλτρα παρουσιάζονται ως **διακριτά chips** πάνω από τα αποτελέσματα, επιτρέποντας την άμεση αφαίρεσή τους χωρίς επιστροφή στο πλήρες μενού επιλογών.

Η αρχιτεκτονική του καταλόγου ενσωματώνει εξ αρχής τα **καινοτόμα στοιχεία** του REFASHION ως **οπτικά διακριτά** χαρακτηριστικά των κάρτων προϊόντων. Ο Eco-Impact Score εμφανίζεται με συνεπή εικονογραφία και χρωματικό κωδικό σε κάθε κάρτα, οι ειδικοί τύποι συναλλαγών (δωρεές, ανταλλαγές) επισημαίνονται με σχετικά badges, ενώ τα νέα προϊόντα ή αυτά με υψηλή ζήτηση φέρουν αντίστοιχες σημάνσεις. Με τον τρόπο αυτό, ο χρήστης μπορεί να **σαρώσει γρήγορα** τον κατάλογο και να εντοπίσει τα προϊόντα που τον ενδιαφέρουν με βάση πολλαπλά κριτήρια, ακόμη και πριν αναζητήσει αναλυτικότερες πληροφορίες.

#### Πρότυπο Σελίδας Προϊόντος

Η **σελίδα προϊόντος** αποτελεί την πιο **πλούσια οπτικά και πληροφοριακά** οθόνη της πλατφόρμας, όπου ο χρήστης παίρνει την τελική αγοραστική απόφαση. Η σχεδίασή της ακολουθεί ένα διπλό πρότυπο: σε ευρύχωρες οθόνες, η σελίδα οργανώνεται σε **δύο στήλες**, με την αριστερή να φιλοξενεί τις φωτογραφίες του προϊόντος (κύρια εικόνα μεγάλου μεγέθους με thumbnails για πλοήγηση), και τη δεξιά να περιέχει όλες τις πληροφορίες αγοράς (τίτλος, τιμή, μέγεθος, κατάσταση, περιγραφή, στοιχεία πωλητή, κουμπιά ενεργειών). Σε στενότερες οθόνες, οι δύο περιοχές αναδιατάσσονται κάθετα, με τις φωτογραφίες στο πάνω μέρος και τις πληροφορίες αμέσως από κάτω.

Ξεχωριστή σχεδιαστική προσοχή δίνεται στην **παρουσίαση του Eco-Impact Score** εντός της σελίδας προϊόντος, όπου εμφανίζεται σε **εκτεταμένη μορφή** που δεν περιορίζεται στον αριθμητικό δείκτη αλλά αναλύεται στις επιμέρους συνιστώσες του (εξοικονόμηση CO<sub>2</sub>, νερού, αποβλήτων). Η ανάλυση αυτή μετατρέπει τον περιβαλλοντικό δείκτη από αόριστο αριθμό σε **απτό και κατανοητό μήνυμα**, ενισχύοντας την αξία της αγοραστικής επιλογής. Παρόμοια προσοχή δίνεται στην εμφάνιση των **ReWear Tokens** που θα αποδοθούν με την αγορά, καθώς και στις σχετικές προτάσεις προϊόντων που αξιοποιούν το Recommendation Service.

#### Πρότυπα Φορμών και Καταχώρησης Δεδομένων

Η καταχώρηση δεδομένων αποτελεί κρίσιμο σημείο της εμπειρίας χρήστη, ιδιαίτερα σε ένα marketplace όπου πωλητές καλούνται τακτικά να συμπληρώνουν εκτεταμένες πληροφορίες προϊόντων. Στο REFASHION εφαρμόζονται **τυποποιημένα πρότυπα φορμών** που εξασφαλίζουν τη συνεκτική και απρόσκοπτη εμπειρία καταχώρησης. Η οργάνωση των πεδίων ακολουθεί **λογική ομαδοποίηση** με χρήση visual sections που διαχωρίζουν θεματικά συναφή πεδία (π.χ. βασικές πληροφορίες, χαρακτηριστικά, φωτογραφίες, τιμολόγηση). Οι μεγάλες φόρμες σπάνε σε **πολλαπλά βήματα (multi-step forms)** με σαφή progress indicator που ενημερώνει τον χρήστη για τη θέση του στη συνολική διαδικασία.

Ένα **διακριτικό σχεδιαστικό στοιχείο** για τη ροή καταχώρησης προϊόντων είναι η ενσωμάτωση των **AI-assisted features** του Visual Product Intelligence Service στη φόρμα. Αντί ο πωλητής να συμπληρώνει αρχικά όλα τα πεδία από το μηδέν, η φόρμα ξεκινά με τη μεταφόρτωση φωτογραφιών, και στη συνέχεια **αυτο-συμπληρώνεται** σταδιακά καθώς οι AI υπηρεσίες επιστρέφουν τα εξαχθέντα δεδομένα. Η σχεδιαστική παρουσίαση αυτής της αυτόματης συμπλήρωσης γίνεται με **οπτικά διακριτικά στοιχεία** (π.χ. ειδικό χρωματισμό ή εικονίδιο) που υποδεικνύουν στον χρήστη ποια πεδία προτείνονται από την AI και χρειάζονται επιβεβαίωση, δημιουργώντας μια εμπειρία **συνεργατικής συμπλήρωσης** μεταξύ χρήστη και πλατφόρμας.

Η **επικύρωση πεδίων** εφαρμόζεται σε πραγματικό χρόνο όπου είναι δυνατό, παρέχοντας άμεση ανατροφοδότηση στον χρήστη για τυχόν σφάλματα ή ελλείψεις, χωρίς να χρειάζεται να υποβάλει τη φόρμα για να ανακαλύψει προβλήματα. Τα μηνύματα σφάλματος εμφανίζονται **κοντά στο σχετικό πεδίο** και είναι διατυπωμένα με **επικοινωνιακό τόνο** που εξηγεί πώς μπορεί να διορθωθεί το πρόβλημα, αντί απλώς να επισημαίνει την ύπαρξή του.

### Πρότυπα Πλοήγησης

Η πλοήγηση στο REFASHION ακολουθεί συνεκτικά πρότυπα σε όλη την πλατφόρμα, αναγνωρίζοντας ότι η **προβλεψιμότητα** των μηχανισμών πλοήγησης αποτελεί κρίσιμο στοιχείο της εμπειρίας χρήστη. Η **κύρια πλοήγηση** εμφανίζεται στο header σε όλες τις σελίδες, παρέχοντας πρόσβαση στις βασικές ενότητες της πλατφόρμας. Η **δευτερεύουσα πλοήγηση** χρησιμοποιείται όπου χρειάζεται για την εμφάθυνση εντός μιας ενότητας (π.χ. tabs στη σελίδα προφίλ χρήστη ή στο seller dashboard). Τα **breadcrumbs** εμφανίζονται σε σελίδες με βαθιά ιεραρχία, επιτρέποντας στον χρήστη να αντιλαμβάνεται τη θέση του και να επιστρέφει εύκολα σε ανώτερα επίπεδα.

Ξεχωριστή σχεδιαστική σημασία έχει η **αναγνώριση της τρέχουσας τοποθεσίας** του χρήστη εντός της πλατφόρμας. Σε κάθε επίπεδο πλοήγησης, η ενεργή επιλογή επισημαίνεται με σαφές οπτικό σήμα (διαφορετικό χρώμα, βάρος γραμματοσειράς, υπογράμμιση), εξασφαλίζοντας ότι ο χρήστης δεν χάνεται ποτέ μέσα στη δομή της πλατφόρμας. Παράλληλα, οι **back navigation** και **deep linking** συμπεριφορές σχεδιάζονται με προσοχή ώστε ο χρήστης να μπορεί ανά πάσα στιγμή να επιστρέψει σε προηγούμενη κατάσταση, να αποθηκεύσει ή να μοιραστεί συνδέσμους που οδηγούν ακριβώς στην επιθυμητή όψη της πλατφόρμας.

### Πρότυπα Ανατροφοδότησης και Καταστάσεων Συστήματος

Η πλατφόρμα σχεδιάζεται με την αρχή ότι ο χρήστης πρέπει **πάντοτε να γνωρίζει την κατάσταση του συστήματος** και τις συνέπειες των ενεργειών του. Για τον σκοπό αυτό, εφαρμόζονται συστηματικά **πρότυπα ανατροφοδότησης** σε διαφορετικά επίπεδα έντασης. Οι **inline messages** εμφανίζονται κοντά στο σχετικό στοιχείο για ήπιες ενημερώσεις και validation feedback. Τα **toast notifications** εμφανίζονται προσωρινά στην άκρη της οθόνης για επιβεβαιώσεις επιτυχών ενεργειών (π.χ. "Το προϊόν προστέθηκε στο καλάθι"). Τα **alerts** χρησιμοποιούνται για σημαντικότερες ενημερώσεις που χρειάζονται μεγαλύτερη οπτική έμφαση. Τέλος, τα **modal dialogs** επιφυλάσσονται για κρίσιμες ενέργειες που απαιτούν επιβεβαίωση από τον χρήστη (π.χ. διαγραφή λογαριασμού), σχεδιασμένα με σαφή προειδοποίηση των συνεπειών και διακριτά κουμπιά για τη συνέχιση ή την ακύρωση.

Για τις **ασύγχρονες λειτουργίες** της πλατφόρμας - που είναι ιδιαίτερα συχνές λόγω της ενσωμάτωσης με AI υπηρεσίες, blockchain και εξωτερικά συστήματα - εφαρμόζονται **κατάλληλα loading states** που ενημερώνουν τον χρήστη ότι μια ενέργεια εκτελείται στο

παρασκήνιο. Τα states αυτά κυμαίνονται από διακριτικά **progress indicators** για σύντομες λειτουργίες, μέχρι πιο επεξηγηματικά **skeleton screens** που προβάλλουν τη δομή της επερχόμενης σελίδας πριν φορτωθούν τα δεδομένα. Όπου μια ενέργεια διαρκεί ασυνήθιστα πολύ (π.χ. καταγραφή στο blockchain), εμφανίζεται **σαφές μήνυμα** που εξηγεί στον χρήστη ότι η διαδικασία θα ολοκληρωθεί ασύγχρονα και ότι θα ενημερωθεί όταν τελειώσει.

### Συνεκτικότητα Μεταξύ Καναλιών

Παρότι τα τέσσερα frontend περιβάλλοντα της πλατφόρμας - web storefront, εφαρμογή κινητού, seller panel, admin panel - αναπτύσσονται με διαφορετικές τεχνολογίες και απευθύνονται σε διαφορετικές κατηγορίες χρηστών, η σχεδιαστική προδιαγραφή τους εξασφαλίζει **οπτική και εμπειρική συνεκτικότητα**. Ένας χρήστης που αλληλεπιδρά με την πλατφόρμα τόσο μέσω web όσο και μέσω εφαρμογής κινητού πρέπει να αναγνωρίζει αμέσως ότι πρόκειται για **την ίδια πλατφόρμα** - μέσω της κοινής χρωματικής παλέτας, της κοινής τυπογραφίας, των κοινών εικονογραφικών στοιχείων και των κοινών μοτίβων αλληλεπίδρασης. Παράλληλα, κάθε περιβάλλον προσαρμόζεται κατάλληλα στις **ιδιαιτερότητες του καναλιού** του: η εφαρμογή κινητού αξιοποιεί native gestures και bottom navigation, το seller panel οργανώνεται με data-dense tables και επαγγελματικά dashboards, ενώ το admin panel ευνοεί την πληροφοριακή πυκνότητα και τις διαχειριστικές ροές.

### 5.3.4 Εργαλεία & Διαδικασία Σχεδιασμού

Η εφαρμογή των σχεδιαστικών προδιαγραφών που παρουσιάστηκαν στις προηγούμενες υποπαραγράφους στηρίζεται σε μια **δομημένη διαδικασία σχεδιασμού**, που περιλαμβάνει τόσο τα εργαλεία που αξιοποιεί η ομάδα όσο και τις πρακτικές συνεργασίας μεταξύ designers και developers. Η διαδικασία αυτή είναι κρίσιμη για ένα έργο όπως το REFASHION, όπου η ανάπτυξη κατανέμεται σε πολλαπλές ομάδες (το ΠΑ.ΠΕΛ για το backend και τα web περιβάλλοντα, η Cyclefy για την εφαρμογή κινητού) και η οπτική συνεκτικότητα απαιτεί αδιαπραγμάτευτη συνεργασία.

Ο σχεδιασμός των διεπαφών πραγματοποιείται σε **σύγχρονα συνεργατικά εργαλεία σχεδιασμού**, τα οποία υποστηρίζουν την παράλληλη εργασία πολλαπλών designers, τη διατήρηση κοινής βιβλιοθήκης components και styles, και την παρακολούθηση της εξέλιξης των σχεδιαστικών αποφάσεων μέσω version history. Σε αυτά τα εργαλεία αναπτύσσονται οι σχεδιαστικές προδιαγραφές σε διαφορετικά επίπεδα ωριμότητας - από αρχικά **wireframes** που εστιάζουν στη δομή και τη ροή, έως **high-fidelity mockups** που αποτυπώνουν την τελική οπτική εμφάνιση, και τελικά **interactive prototypes** που επιτρέπουν την προεπισκόπηση της εμπειρίας πριν την υλοποίηση.

Η **διαδικασία σχεδιασμού** ακολουθεί μια επαναληπτική προσέγγιση που εκκινεί από την κατανόηση των αναγκών των χρηστών μέσω **user research και personas**, προχωρά σε σχεδιαστικές προτάσεις, και ολοκληρώνεται με **εσωτερική κριτική και αναθεώρηση** πριν την παράδοση στις ομάδες ανάπτυξης. Η εμπλοκή των developers στη διαδικασία ξεκινά **νωρίς** μέσω εργαστηρίων ανασκόπησης, εξασφαλίζοντας ότι οι σχεδιαστικές προτάσεις είναι **τεχνικά υλοποιήσιμες** και ευθυγραμμισμένες με τις δυνατότητες των επιλεγμένων frameworks.

Η **παράδοση του σχεδιασμού (design handoff)** στην ομάδα ανάπτυξης πραγματοποιείται μέσω συστηματικής τεκμηρίωσης που περιλαμβάνει τις τελικές οθόνες, τα design tokens με τις ακριβείς αξίες τους, τις προδιαγραφές κάθε component σε όλες τις καταστάσεις του, και αναλυτικές οδηγίες για τη συμπεριφορά διαδραστικών στοιχείων. Τα σύγχρονα

εργαλεία σχεδιασμού επιτρέπουν στους developers να **επιθεωρούν απευθείας** τα στοιχεία της σχεδίασης, να εξάγουν χρώματα, διαστάσεις και CSS κώδικα, επιταχύνοντας σημαντικά τη μετάφραση του σχεδιασμού σε λειτουργικό κώδικα.

Καθώς η πλατφόρμα εξελίσσεται, η **σχεδιαστική προδιαγραφή παραμένει ζωντανός οργανισμός** που ενημερώνεται και εμπλουτίζεται διαρκώς. Νέα components προστίθενται όταν εμφανίζονται νέες απαιτήσεις, υφιστάμενα μοτίβα αναθεωρούνται όταν προκύπτουν καλύτερες προσεγγίσεις, και τα design tokens μπορούν να επαναπροσαρμοστούν για να βελτιωθεί η συνολική αισθητική. Η συστηματική αυτή εξέλιξη εξασφαλίζει ότι το design system του REFASHION δεν αποτελεί στατικό κατάλογο αλλά **δυναμικό εργαλείο** που υποστηρίζει τη μακροπρόθεσμη ανάπτυξη της πλατφόρμας.

## 6. Υπηρεσίες Τεχνητής Νοημοσύνης (AI Services)

Η πλατφόρμα REFASHION ενσωματώνει ένα σύνολο ανεξάρτητων υπηρεσιών Τεχνητής Νοημοσύνης, οι οποίες λειτουργούν ως εξωτερικά *microservices* γύρω από τον κεντρικό πυρήνα του Medusa backend. Η επιλογή αυτής της αρχιτεκτονικής επιτρέπει την ανεξάρτητη ανάπτυξη, δοκιμή και κλιμάκωση των AI λειτουργιών, χωρίς να επιβαρύνεται ο βασικός e-commerce μηχανισμός της πλατφόρμας.

Οι AI υπηρεσίες του REFASHION καλύπτουν τρεις βασικούς άξονες:

- προεπεξεργασία και ανάλυση εικόνων προϊόντων,
- οπτική αναζήτηση και εμπλουτισμό πληροφοριών προϊόντος,
- παραγωγή προσωποποιημένων προτάσεων προϊόντων.

Η επικοινωνία των υπηρεσιών με την πλατφόρμα γίνεται μέσω HTTP API endpoints. Τα AI *microservices* εκτελούνται τοπικά ή σε απομονωμένο server περιβάλλον, ενώ το frontend και το backend της πλατφόρμας επικοινωνούν μαζί τους μέσω proxy routes, ώστε να διατηρείται ενιαίο *origin* για τον χρήστη και να μη γίνεται απευθείας έκθεση των βοηθητικών υπηρεσιών στο διαδίκτυο.

### 6.1. Image Analysis Service

Το Image Analysis Service είναι υπεύθυνο για την επεξεργασία εικόνων που ανεβάζουν οι χρήστες κατά την καταχώρηση προϊόντων. Στην τρέχουσα υλοποίηση, η υπηρεσία λειτουργεί ως FastAPI *microservice* και παρέχει μηχανισμό αυτόματης αφαίρεσης φόντου από φωτογραφίες προϊόντων, ώστε οι καταχωρήσεις να εμφανίζονται με πιο ομοιόμορφο, καθαρό και επαγγελματικό τρόπο στο storefront.

Η υπηρεσία δέχεται εικόνα μέσω multipart/form-data, εφαρμόζει αλγόριθμο background removal και επιστρέφει αρχείο PNG. Με αυτόν τον τρόπο παράγεται πιο καθαρή και ομοιόμορφη απεικόνιση προϊόντος, η οποία μπορεί να χρησιμοποιηθεί τόσο στην καταχώρηση του marketplace όσο και σε μελλοντικά υποσυστήματα computer vision.

Η βασική ροή λειτουργίας είναι η εξής:

1. Ο χρήστης ανεβάζει φωτογραφία προϊόντος από το frontend.
2. Το backend λαμβάνει το αρχείο μέσω του endpoint /imageprocess.
3. Το backend προωθεί την εικόνα στο εσωτερικό FastAPI *microservice*.
4. Το *microservice* επεξεργάζεται την εικόνα και επιστρέφει PNG με αφαιρεμένο φόντο.
5. Το αποτέλεσμα συνδέεται με την εγγραφή προϊόντος και εμφανίζεται στο frontend.

Στο επίπεδο υλοποίησης χρησιμοποιούνται βιβλιοθήκες Python όπως FastAPI, rembg, Pillow και onnxruntime, γεγονός που επιτρέπει την εκτέλεση inference χωρίς ανάγκη άμεσης ενσωμάτωσης βαρύτερων ML components στον Medusa backend.

Η αρχιτεκτονική αυτή επιτρέπει τη μελλοντική επέκταση της υπηρεσίας για πιο σύνθετες λειτουργίες computer vision, όπως:

- εκτίμηση κατάστασης προϊόντος από εικόνα,
- αναγνώριση βασικών χαρακτηριστικών ένδυσης,
- εκτίμηση χρώματος, υφής και τύπου υλικού,
- αυτόματη πρόταση κατηγορίας προϊόντος.

Με αυτόν τον τρόπο, το Image Analysis Service αποτελεί το πρώτο επίπεδο του AI pipeline, καθώς προετοιμάζει τις εικόνες για περαιτέρω ανάλυση και βελτιώνει την ποιότητα των καταχωρήσεων.

## 6.2. Visual Product Intelligence Service

Η υπηρεσία Visual Product Intelligence Service έχει ως στόχο τον αυτόματο εμπλουτισμό καταχωρήσεων προϊόντων με βάση την εικόνα που ανεβάζει ο πωλητής. Η υπηρεσία αξιοποιεί reverse image search και LLM-based ανάλυση, ώστε από μία εικόνα προϊόντος να παραχθούν δομημένα πεδία που μπορούν να χρησιμοποιηθούν για την αυτόματη ή ημιαυτόματη συμπλήρωση της καταχώρησης.

Η υπηρεσία έχει υλοποιηθεί ως ανεξάρτητο FastAPI microservice και εκθέτει το endpoint:

POST /v1/product/from-image

Το endpoint δέχεται εικόνα μέσω multipart/form-data και επιστρέφει δομημένο JSON με τα ακόλουθα πεδία:

```
{
  "name": "...",
  "category": "...",
  "description": "..."}
}
```

Η υπηρεσία ακολουθεί pipeline πολλαπλών σταδίων:

- Λήψη εικόνας από τον χρήστη**  
Η εικόνα προϊόντος ανεβαίνει στο service ως αρχείο. Πραγματοποιείται έλεγχος τύπου αρχείου, ώστε να γίνονται δεκτά μόνο αρχεία εικόνας, καθώς και έλεγχος μεγέθους με μέγιστο όριο 10 MB.
- Προσωρινή φιλοξενία εικόνας**  
Η εικόνα αποθηκεύεται προσωρινά στον φάκελο tmp\_images και αντιστοιχίζεται σε μοναδικό token. Μέσω του endpoint /tmp/{token} δημιουργείται προσωρινό δημόσια προσβάσιμο URL, το οποίο απαιτείται για την εκτέλεση reverse image search από εξωτερική υπηρεσία. Τα προσωρινά αρχεία έχουν περιορισμένο χρόνο ζωής και διαγράφονται μετά την ολοκλήρωση του request.
- Reverse Image Search μέσω Serper Lens**  
Η υπηρεσία καλεί το endpoint <https://google.serper.dev/lens>, αποστέλλοντας το προσωρινό URL της εικόνας. Το Serper Lens επιστρέφει αποτελέσματα reverse image search, τα οποία περιλαμβάνουν οπτικά παρόμοια προϊόντα, τίτλους, συνδέσμους και περιγραφικά στοιχεία από τον ιστό.
- Περιορισμός και καθαρισμός αποτελεσμάτων**  
Τα αποτελέσματα περιορίζονται με βάση την παράμετρο top\_k, η οποία έχει προεπιλεγμένη τιμή 10 και μέγιστη τιμή 20. Με αυτόν τον τρόπο ελέγχεται ο όγκος πληροφορίας που αποστέλλεται στο επόμενο στάδιο επεξεργασίας.
- LLM-Based Extraction μέσω Ollama**  
Τα αποτελέσματα του reverse image search αποστέλλονται σε τοπικό ή απομακρυσμένο Ollama endpoint. Στην τρέχουσα υλοποίηση χρησιμοποιείται προεπιλεγμένα το μοντέλο llama3.1:8b. Το LLM λαμβάνει ως είσοδο αποκλειστικά τα δεδομένα του reverse search και παράγει αυστηρά δομημένο JSON με τρία πεδία: name, category, description.

Η υπηρεσία είναι σχεδιασμένη έτσι ώστε το LLM να μην χρησιμοποιεί εξωτερική γνώση πέρα από το JSON των αποτελεσμάτων. Αυτό μειώνει τον κίνδυνο αυθαίρετων απαντήσεων και διατηρεί την εξαγωγή πληροφορίας συνδεδεμένη με τα πραγματικά οπτικά αποτελέσματα αναζήτησης.

Σε επίπεδο παραμετροποίησης, η υπηρεσία χρησιμοποιεί μεταβλητές περιβάλλοντος:

- SERPER\_API\_KEY: κλειδί πρόσβασης στο Serper Lens API,
- PUBLIC\_BASE\_URL: δημόσια διεύθυνση μέσω της οποίας είναι προσβάσιμες προσωρινά οι εικόνες,
- OLLAMA\_URL: endpoint του Ollama service,
- OLLAMA\_MODEL: μοντέλο LLM που χρησιμοποιείται για την εξαγωγή πληροφορίας.

Η υλοποίηση αυτή αντικαθιστά πιο εύθραυστες προσεγγίσεις browser automation, καθώς βασίζεται σε API-based reverse image search και σε τοπική LLM επεξεργασία. Έτσι, η υπηρεσία είναι πιο σταθερή, πιο εύκολη στην ανάπτυξη και περισσότερο κατάλληλη για ενσωμάτωση σε παραγωγικό περιβάλλον.

### 6.3. Recommendation Service

Το Recommendation Service αποτελεί το υποσύστημα προσωποποιημένων προτάσεων προϊόντων της πλατφόρμας. Υλοποιείται επίσης ως ανεξάρτητο FastAPI microservice και συνδέεται απευθείας με τη βάση δεδομένων PostgreSQL της πλατφόρμας.

Η τρέχουσα λογική συστάσεων βασίζεται σε εύρεση κοντινών χρηστών με βάση τα χαρακτηριστικά του χρήστη όπως η ηλικία. Για έναν δεδομένο χρήστη, το service εντοπίζει άλλους χρήστες με παρόμοια χαρακτηριστικά και στη συνέχεια αναζητά προϊόντα που έχουν εμφανιστεί στο ιστορικό παραγγελιών ή αλληλεπιδράσεων αυτών των χρηστών. Τα προϊόντα αυτά χρησιμοποιούνται ως προτεινόμενα προϊόντα για τον αρχικό χρήστη.

Το service υποστηρίζει επίσης περιοδική εκτέλεση μέσω scheduler. Κατά την εκκίνηση ή σε προκαθορισμένα χρονικά διαστήματα, μπορεί να επαναυπολογίζει τις προτάσεις για όλους τους χρήστες, ώστε οι συστάσεις να παραμένουν ενημερωμένες. Η προεπιλεγμένη περίοδος ανανέωσης έχει οριστεί σε ημερήσια βάση.

Η υπηρεσία υποστηρίζει background execution μέσω ελεγχόμενου trigger. Το backend proxy endpoint είναι:

GET /recommend?neighbor\_limit=4&product\_limit=3&refresh\_seconds=86400

Η κλήση προστατεύεται με x-api-key, ώστε να μην μπορεί να ενεργοποιηθεί από μη εξουσιοδοτημένους χρήστες. Όταν καλείται το endpoint, ενεργοποιείται η διαδικασία υπολογισμού προτάσεων για όλους τους πελάτες της βάσης. Επιπλέον, υπάρχει endpoint για διακοπή του trigger:

POST /recommend/stop

### 6.4. Υποδομή Δεδομένων

Η υποδομή δεδομένων των AI Services έχει σχεδιαστεί με στόχο την απομόνωση των AI workloads από τον βασικό e-commerce πυρήνα, διατηρώντας όμως ενιαία ροή δεδομένων προς την πλατφόρμα REFASHION.

Τα AI microservices λειτουργούν ως ξεχωριστές υπηρεσίες στις ακόλουθες θύρες ανάπτυξης:

Image Processing Service: 127.0.0.1:9100

Visual Search Service: 127.0.0.1:9300

Recommendation Service: 127.0.0.1:9400

Η πρόσβαση από το frontend δεν γίνεται απευθείας στα παραπάνω services. Αντίθετα,

χρησιμοποιούνται proxy routes στον Medusa backend, όπως:

- /imageprocess
- /visualsearch
- /recommend

Με αυτό το μοντέλο, η πλατφόρμα διατηρεί ενιαίο API surface προς το frontend, ενώ τα εσωτερικά AI services μπορούν να τροποποιούνται ή να κλιμακώνονται ανεξάρτητα.

Η διαχείριση αρχείων γίνεται προσωρινά κατά την εκτέλεση του request. Οι εικόνες αποθηκεύονται σε προσωρινό αρχείο, υποβάλλονται σε επεξεργασία και στη συνέχεια διαγράφονται. Η επιλογή αυτή μειώνει τον κίνδυνο μη αναγκαίας αποθήκευσης προσωπικών ή ευαίσθητων δεδομένων εικόνας.

Για το Recommendation Service, η κύρια πηγή δεδομένων είναι η PostgreSQL βάση της πλατφόρμας. Η υπηρεσία χρησιμοποιεί connection pool για αποδοτική πρόσβαση στους πίνακες χρηστών, παραγγελιών και προϊόντων. Τα παραγόμενα recommendations αποθηκεύονται πίσω στη βάση, ώστε να είναι διαθέσιμα άμεσα από το backend και το frontend χωρίς να απαιτείται κάθε φορά εκτέλεση του αλγορίθμου σε πραγματικό χρόνο.

Το Visual Product Intelligence Service χρησιμοποιεί προσωρινή αποθήκευση εικόνων μόνο κατά τη διάρκεια του request. Κάθε εικόνα αποθηκεύεται σε προσωρινό αρχείο, συσχετίζεται με μοναδικό token και εκτίθεται προσωρινά μέσω του endpoint /tmp/{token}. Η διάρκεια ζωής των προσωρινών αρχείων είναι περιορισμένη, ενώ μετά την ολοκλήρωση της επεξεργασίας πραγματοποιείται αυτόματη διαγραφή.

Τα δεδομένα που παράγονται από την υπηρεσία δεν είναι ακατέργαστα δεδομένα εικόνας, αλλά δομημένα μεταδεδομένα προϊόντος. Συγκεκριμένα, το τελικό αποτέλεσμα περιλαμβάνει προτεινόμενη ονομασία προϊόντος, κατηγορία και περιγραφή. Τα δεδομένα αυτά χρησιμοποιούνται άμεσα στο frontend ως προτεινόμενα πεδία κατά τη δημιουργία νέας καταχώρησης.

Η αρχιτεκτονική διατηρεί σαφή διαχωρισμό μεταξύ:

- της εικόνας εισόδου,
- των αποτελεσμάτων reverse image search,
- της LLM-based εξαγωγής πληροφορίας,

και της τελικής δομημένης πληροφορίας που αξιοποιείται από την πλατφόρμα.

Με αυτόν τον τρόπο, η υπηρεσία μπορεί να εξελιχθεί ανεξάρτητα από τον κεντρικό Medusa backend, ενώ παράλληλα προσφέρει καθαρό API για ενσωμάτωση με το frontend και τη ροή καταχώρησης προϊόντων.

## 7. Υποσύστημα Ανταμοιβών RFT - Υλοποίηση & Διασύνδεση

### 7.1. Σκοπός & Αρχιτεκτονική Προσέγγιση

Το υποσύστημα ανταμοιβών **ReFashion Token (RFT)** υλοποιεί τη σύνδεση της πλατφόρμας REFASHION με το blockchain backend, μέσω του οποίου εκδίδονται, διαχειρίζονται και εξαργυρώνονται τα ψηφιακά tokens που αποδίδονται στους χρήστες. Από αρχιτεκτονική σκοπιά, η σχεδίαση του υποσυστήματος ακολουθεί μια **σαφή αρχή διαχωρισμού ευθυνών (separation of concerns)** μεταξύ των δύο εμπλεκόμενων συστημάτων: το **REFASHION backend** ευθύνεται για το σύνολο της **επιχειρησιακής λογικής** - υπολογισμός των ανταμοιβών, εφαρμογή κανόνων επιλεξιμότητας και ορίων, παρακολούθηση καταστάσεων χρηστών - ενώ το **εξωτερικό RFT backend** ευθύνεται αποκλειστικά για την **εκτέλεση των blockchain ενεργειών** - δημιουργία wallets, έκδοση και αφαίρεση tokens, ανάκτηση balance.

Ο διαχωρισμός αυτός εξυπηρετεί τόσο τη συντηρησιμότητα όσο και την ευελιξία του συστήματος. Η πλατφόρμα REFASHION παραμένει **απομονωμένη** από τις λεπτομέρειες υλοποίησης του blockchain πρωτοκόλλου, ενώ το RFT backend μπορεί να αναβαθμίζεται ή να μεταβαίνει σε διαφορετική αλυσίδα χωρίς αλλαγές στη βασική λογική του marketplace. Η ροή των αιτημάτων διασχίζει τρία διακριτά επίπεδα: από τα **frontend περιβάλλοντα** (storefront, mobile app), στο **REFASHION backend** που υπολογίζει και ενορχηστρώνει τις ανταμοιβές, και τελικά στο **RFT API** που εκτελεί τις αντίστοιχες blockchain συναλλαγές και επιστρέφει τα αναγνωριστικά τους (transaction hash, block number).

### 7.2. Επισκόπηση Tokenomics

Πριν την παρουσίαση των τεχνικών συνιστωσών της υλοποίησης, κρίνεται σκόπιμη μια **σύντομη επισκόπηση** των κανόνων ανταμοιβής που εφαρμόζει το REFASHION backend, ώστε να γίνει κατανοητό το πλαίσιο εντός του οποίου λειτουργούν οι μηχανισμοί που περιγράφονται στις επόμενες υποπαραγράφους. Η πλήρης τεκμηρίωση των tokenomics εκφεύγει του σκοπού του παρόντος παραδοτέου, που εστιάζει στην αρχιτεκτονική υλοποίηση και διασύνδεση παρά στους επιχειρησιακούς κανόνες αυτούς καθαυτούς.

Το σύστημα ανταμοιβής αξιοποιεί το **ReFashion Token (RFT)** ως ψηφιακή μονάδα αξίας με σταθερή ισοτιμία προς το ευρώ (1 RFT = 0,01 €), και αποδίδει tokens στους χρήστες μέσω **τριών βασικών κατηγοριών events**. Η πρώτη κατηγορία αφορά τις **εγγενείς ενέργειες χρήστη** εντός της πλατφόρμας — η αυτόματη επαλήθευση προφίλ κατά την εγγραφή και η έγκριση καταχωρήσεων προϊόντων από πωλητές παράγουν προκαθορισμένες ανταμοιβές, με διαφοροποιημένη αξία για την πρώτη εγκεκριμένη καταχώρηση κάθε πωλητή έναντι των επακόλουθων. Η δεύτερη κατηγορία αφορά τις **συναλλαγές εμπορίου**, όπου τόσο οι αγοραστές όσο και οι πωλητές λαμβάνουν ανταμοιβή ως ποσοστό της αξίας της παραγγελίας, με την πλευρά των πωλητών να υπόκειται σε ανώτατα όρια ανά συναλλαγή και ετήσια ανά πωλητή για την αποτροπή κατάχρησης. Η τρίτη κατηγορία αφορά τα **περιβαλλοντικά milestones**, σύμφωνα με τα οποία η συσσώρευση καθορισμένης ποσότητας εξοικονόμησης CO<sub>2</sub>-ισοδυνάμου από τις αγορές ενός χρήστη ενεργοποιεί αυτόματη ανταμοιβή, ενισχύοντας έμπρακτα τη σύνδεση μεταξύ περιβαλλοντικής συνεισφοράς και ψηφιακής επιβράβευσης.

Συμπληρωματικά προς τους μηχανισμούς απόδοσης, το σύστημα υποστηρίζει τη **διαδικασία εξαργύρωσης (redemption)** των tokens κατά την ολοκλήρωση παραγγελιών, με τους χρήστες να μπορούν να μειώσουν την τελική αξία της αγοράς τους χρησιμοποιώντας τα συσσωρευμένα tokens τους. Η εξαργύρωση υπόκειται σε **ανώτατο ποσοστιαίο όριο** ανά παραγγελία, εξασφαλίζοντας τη βιωσιμότητα του μηχανισμού και την αποτροπή χρήσης των tokens ως πλήρους υποκατάστατου χρηματικής συναλλαγής.

Όλοι οι κανόνες υπολογισμού, τα όρια, τα ποσοστά και τα milestones παραμετροποιούνται μέσω **τυποποιημένων σταθερών** στην κωδικοβάση του backend, επιτρέποντας στους διαχειριστές του έργου να **αναπροσαρμόζουν τη στρατηγική ανταμοιβής** χωρίς αλλαγές στη βασική αρχιτεκτονική. Με τον τρόπο αυτό, η υλοποίηση των tokenomics παραμένει ευέλικτη και εξελίξιμη, καθώς οι κανόνες ενδέχεται να αναθεωρηθούν με βάση τα ευρήματα της πιλοτικής λειτουργίας και τη συμπεριφορά των πραγματικών χρηστών.

### 7.3. Επέκταση Μοντέλου Δεδομένων

Για την υποστήριξη του υποσυστήματος, το μοντέλο δεδομένων της πλατφόρμας επεκτείνεται με **δύο νέες οντότητες** που υλοποιούνται ως custom tables στη βάση PostgreSQL. Η πρώτη οντότητα, **rft\_user\_reward\_state**, αποθηκεύει την τρέχουσα κατάσταση ανταμοιβών κάθε χρήστη - τη διεύθυνση του blockchain wallet του, τους συγκεντρωτικούς μετρητές των tokens που έχουν εκδοθεί ανά κατηγορία (buyer, seller, in-app), τα ετήσια όρια που έχουν συμπληρωθεί, καθώς και πεδία ελέγχου όπως οι ανακτηθείσες ορόσημες και η συνολική περιβαλλοντική επίπτωση. Η δεύτερη οντότητα, **rft\_reward\_ledger**, λειτουργεί ως **πλήρες audit trail** όλων των token events: κάθε έκδοση ή αφαίρεση tokens καταγράφεται ως διακριτή εγγραφή με συσχέτιση προς την πηγή ενεργοποίησης (παραγγελία, listing, milestone), το τύπο του event, την κατάσταση εκτέλεσης (PENDING, ISSUED, DEDUCTED, FAILED, DUPLICATE), και τα blockchain references (tx\_hash, block\_number) που επιστρέφονται από το RFT API.

Η σχεδιαστική επιλογή της διπλής αυτής δομής αντικατοπτρίζει μια θεμελιώδη αρχή: το **blockchain ledger αποτελεί την τελική αλήθεια** για τις κινήσεις των tokens, ενώ το **εσωτερικό ledger της πλατφόρμας τεκμηριώνει το γιατί** αυτές οι κινήσεις συνέβησαν. Η συνδυαστική τους αξιοποίηση παρέχει πλήρη διαφάνεια και ιχνηλασιμότητα, καθώς και τη δυνατότητα reconciliation σε περίπτωση αποκλίσεων.

### 7.4. RFT API Client & Integration Layer

Η επικοινωνία με το εξωτερικό RFT backend υλοποιείται μέσω ενός **εξειδικευμένου integration client (rft-client.ts)**, ο οποίος ενσωματώνει όλες τις διαθέσιμες λειτουργίες του RFT API σε αυτόνομες, τυποποιημένες κλήσεις. Ο client καλύπτει τη δημιουργία wallets (createWallet), την έκδοση και αφαίρεση tokens (issueReward, deductReward), την ανάκτηση balance (getBalance), και την απενεργοποίηση wallet (deactivateUser), προσφέροντας μια **καθαρή και τυπικά ελεγχόμενη διεπαφή** προς την υπόλοιπη κωδικοβάση. Η αυθεντικοποίηση των αιτημάτων εκτελείται μέσω **API Key** που εγχέεται ως environment variable, ενώ το base URL του RFT backend είναι παραμετροποιήσιμο για την υποστήριξη διαφορετικών περιβαλλόντων (development, staging, production).

Σημαντικό σχεδιαστικό χαρακτηριστικό αποτελεί η ενσωμάτωση **ισχυρού error handling** στον client. Κάθε αποτυχημένο αίτημα μεταφράζεται σε διακριτό exception με ενδεικτικό μήνυμα, επιτρέποντας στα ανώτερα επίπεδα της εφαρμογής να αντιμετωπίσουν διαφορετικά κάθε κατηγορία σφάλματος (network failure, authentication error, business logic rejection). Παράλληλα, η σχεδίαση του client ενσωματώνει την υποστήριξη της αρχής της **idempotency** μέσω του πεδίου rewardEventId που συνοδεύει κάθε αίτημα έκδοσης ή αφαίρεσης - μηχανισμός που αναλύεται περαιτέρω στη συνέχεια.

### 7.5. Backend Endpoints & Event-Driven Poés

Πάνω από τον RFT client, το REFASHION backend εκθέτει ένα σύνολο **εξειδικευμένων endpoints** που διαμεσολαβούν μεταξύ των frontend περιβαλλόντων και του RFT subsystem.

Τα endpoints αυτά οργανώνονται σε δύο κατηγορίες: **admin endpoints** (/admin/rft/events/\*) που ενεργοποιούνται από εσωτερικά events της πλατφόρμας και εκτελούν αυτόματα τους υπολογισμούς ανταμοιβών, και **store endpoints** (/store/rft/\*) που εξυπηρετούν τις αλληλεπιδράσεις των τελικών χρηστών με το σύστημα.

Στα **admin endpoints** περιλαμβάνονται τα account-created (που δημιουργεί wallet και εκδίδει την αρχική profile ανταμοιβή κατά την εγγραφή), listing-approved (που αποδίδει tokens σε πωλητές μετά την έγκριση καταχωρήσεων), και order-completed (που εκτελεί τους πιο σύνθετους υπολογισμούς - buyer reward, seller reward με ετήσια όρια, και milestone-based ανταμοιβές που σχετίζονται με την περιβαλλοντική επίπτωση κάθε αγοράς). Στα **store endpoints** περιλαμβάνονται τα state (που επιστρέφει στον χρήστη την τρέχουσα κατάσταση ανταμοιβών του), redemption/calculate και redemption/apply (που υλοποιούν τη ροή εξαργύρωσης tokens κατά το checkout), καθώς και το balance/sync (που συγχρονίζει το cached balance με το RFT backend).

Σε επόμενη φάση εξέλιξης, η ενεργοποίηση των admin endpoints αναμένεται να διασυνδεθεί απευθείας με το **Event Bus του Medusa** μέσω αντίστοιχων Subscribers, σύμφωνα με την αρχιτεκτονική που αναλύθηκε στο Κεφάλαιο 2. Με τον τρόπο αυτό, τα events του πυρήνα (customer.created, order.placed, product.approved) θα πυροδοτούν αυτόματα τη ροή ανταμοιβών χωρίς την ανάγκη ρητής κλήσης από εξωτερικό παράγοντα, εμπλουτίζοντας το event-driven μοτίβο της πλατφόρμας.

## 7.6. Παρατηρήσεις Υλοποίησης

Η υλοποίηση του υποσυστήματος ενσωματώνει ορισμένες **κρίσιμες αρχές σχεδιασμού** που εξασφαλίζουν τη σταθερότητα και την αξιοπιστία του σε παραγωγικό περιβάλλον. Η πιο σημαντική αποτελεί η εφαρμογή του **idempotency pattern** μέσω μοναδικών rewardEventId που συνοδεύουν κάθε αίτημα προς το RFT API. Με την αρχή αυτή, ακόμη και αν ένα αίτημα επαναληφθεί λόγω network retry ή σφάλματος της εφαρμογής, το σύστημα δεν θα εκδώσει διπλή ανταμοιβή - το RFT backend αναγνωρίζει το ήδη επεξεργασμένο event μέσω του rewardEventId και επιστρέφει την προηγούμενη συναλλαγή χωρίς να εκτελέσει νέα. Παράλληλα, η ίδια αρχή εφαρμόζεται και σε τοπικό επίπεδο μέσω της λειτουργίας ledgerExistsByRewardEventId, που ελέγχει την προϋπάρχουσα καταχώρηση πριν την κλήση του RFT API.

Δεύτερη κρίσιμη πτυχή αποτελεί η **πλήρης audit logging** όλων των ενεργειών στο εσωτερικό ledger, ανεξάρτητα από το αποτέλεσμα. Επιτυχίες συναλλαγές, αποτυχίες, διπλότυπα, ακυρώσεις - όλα καταγράφονται με τα κατάλληλα statuses και metadata, εξασφαλίζοντας ότι κάθε ενέργεια της πλατφόρμας είναι **πλήρως ιχνηλάσιμη** σε περίπτωση ελέγχου, αντιμετώπισης σφαλμάτων ή reconciliation με τις on-chain εγγραφές.

Τέλος, η αρχιτεκτονική υποστηρίζει μηχανισμούς **graceful degradation** σε περίπτωση μη διαθεσιμότητας του RFT backend. Καθώς η blockchain επικοινωνία εκτελείται **εκτός του κρίσιμου μονοπατιού** των κύριων ροών (παραγγελίες, εγγραφές), προσωρινή αστοχία του RFT API δεν επηρεάζει την ομαλή λειτουργία της πλατφόρμας - οι εκκρεμείς ανταμοιβές παραμένουν σε κατάσταση PENDING και μπορούν να επεξεργαστούν εκ νέου όταν η υπηρεσία αποκατασταθεί, σύμφωνα με το event-driven μοτίβο που χαρακτηρίζει την αρχιτεκτονική του REFASHION.

## 8. Τεκμηρίωση API Endpoints

Η πλατφόρμα εκθέτει ένα πλήρες REST API. Τα endpoints κατηγοριοποιούνται ως εξής:

### 8.1. Αυθεντικοποίηση & Διαχείριση Χρηστών

Μέθοδος	Endpoint	Περιγραφή
POST	/auth/seller/emailpass/register	Εγγραφή πωλητή (Βήμα 1)
POST	/vendor/sellers	Προσθήκη προφίλ πωλητή (Βήμα 2)
POST	/auth/seller/emailpass	Σύνδεση πωλητή
POST	/auth/customer/emailpass/register	Εγγραφή αγοραστή (Βήμα 1)
POST	/store/customers	Προσθήκη προφίλ αγοραστή (Βήμα 2)
GET	/store/customers/me	Εμφάνιση πληροφοριών συνδεδεμένου προφίλ
POST	/auth/customer/emailpass	Σύνδεση αγοραστή
DELETE	/admin/customers/{id}/force-delete	Διαγραφή λογαριασμού

### 8.2. Διαχείριση Προϊόντων

Μέθοδος	Endpoint	Περιγραφή
GET	/store/products	Λίστα προϊόντων
POST	/store/products	Καταχώρηση νέου προϊόντος (custom)
GET	/store/product-tags	Εμφάνιση tags προϊόντων
GET	/store/product-categories	Κατηγορίες προϊόντων (tree)
GET	/store/custom/product-filters	Φιλτράρισμα (material, collection, category, price)

### 8.3. Ροή Αγοράς (Cart Flow)

Μέθοδος	Endpoint	Περιγραφή
POST	/store/carts	Δημιουργία καλαθιού
POST	/store/carts/{id}/line-items	Προσθήκη προϊόντος
GET	/store/shipping-options?cart_id={id}	Μέθοδοι αποστολής
POST	/store/carts/{id}/shipping-methods	Ορισμός αποστολής
POST	/store/carts/{id}/payment-sessions	Δημιουργία πληρωμής
POST	/store/carts/{id}/complete	Ολοκλήρωση παραγγελίας
GET	/store/orders	Ιστορικό παραγγελιών
GET	/store/orders/{id}	Ανάκτηση παραγγελίας με βάση το id
GET	/store/payment-providers	Ανάκτηση των διαθέσιμων payment provider
GET	/store/currencies	Ανάκτηση των διαθέσιμων currencies
GET	/store/collections	Ανάκτηση των διαθέσιμων collections
GET	/store/collections/{id}	Ανάκτηση συγκεκριμένου collection με βάση το id
POST	/store/shipping-options/{id}/calculate	Υπολογισμός τιμής με βάση το shipping option

### 8.4. Σύστημα Ανταμοιβών

Μέθοδος	Endpoint	Περιγραφή
---------	----------	-----------

<b>GET</b>	/store/rft/state	Ανάκτηση τρέχουσας κατάστασης ανταμοιβών χρήστη (counters, milestones, balance)
<b>POST</b>	/store/rft/balance/sync	Συγχρονισμός cached balance με το RFT backend
<b>POST</b>	/store/rft/redemption/calculate	Υπολογισμός μέγιστης επιτρεπτής εξαργύρωσης για συγκεκριμένο καλάθι
<b>POST</b>	/store/rft/redemption/apply	Εφαρμογή εξαργύρωσης tokens κατά το checkout
<b>POST</b>	/admin/rft/events/account-created	Δημιουργία wallet & έκδοση profile reward
<b>POST</b>	/admin/rft/events/listing-approved	Απόδοση ανταμοιβής μετά από έγκριση καταχώρησης πωλητή
<b>POST</b>	/admin/rft/events/order-completed	Επεξεργασία buyer/seller rewards & milestone-based ανταμοιβών

### 8.5. Βοηθητικά Endpoints

Μέθοδος	Endpoint	Περιγραφή
GET	/store/regions	Διαθέσιμες περιοχές
POST	/store/customer/address	Προσθήκη διεύθυνσης
POST	/store/carts/{cart_id}/taxes	Υπολογισμός φόρου
POST	/store/wishlist	Προσθήκη προϊόντος σε wishlist
GET	/store/wishlist	Ανάκτηση wishlist
DELETE	/store/wishlist/{wishlist_id}/product/{product_id}	Διαγραφή προϊόντος από wishlist

## 9. Ασφάλεια & Συμμόρφωση GDPR

Σε ένα οικοσύστημα όπως το REFASHION, που διαχειρίζεται προσωπικά δεδομένα χρηστών, οικονομικές συναλλαγές και ψηφιακά περιουσιακά στοιχεία (digital assets), η **ασφάλεια αποτελεί θεμελιώδη σχεδιαστική αρχή** και όχι εκ των υστέρων προσθήκη. Η αρχιτεκτονική της πλατφόρμας ενσωματώνει μηχανισμούς ασφάλειας σε όλα τα επίπεδα - από την επικοινωνία frontend με backend έως την αποθήκευση δεδομένων στη βάση και την αλληλεπίδραση με εξωτερικές υπηρεσίες - υιοθετώντας μια **defense-in-depth προσέγγιση**, σύμφωνα με την οποία πολλαπλά αμυντικά στρώματα συνεργάζονται ώστε η αστοχία ενός να μην οδηγεί σε πλήρη παραβίαση του συστήματος. Παράλληλα, η πλατφόρμα σχεδιάζεται με γνώμονα την πλήρη συμμόρφωση προς τον **Γενικό Κανονισμό Προστασίας Δεδομένων (GDPR)** της Ευρωπαϊκής Ένωσης, αναγνωρίζοντας τόσο τη νομική υποχρέωση όσο και την ηθική δέσμευση προς τους χρήστες της.

### 9.1. Σχεδιαστικές Αρχές Ασφάλειας

Η αρχιτεκτονική ασφάλειας του REFASHION εδράζεται σε τέσσερις θεμελιώδεις αρχές που διέπουν όλες τις σχεδιαστικές αποφάσεις. Πρώτη και κυρίαρχη είναι η αρχή της **defense-in-depth**, σύμφωνα με την οποία η ασφάλεια δεν εξαρτάται από έναν και μοναδικό αμυντικό μηχανισμό αλλά από τη συνεργασία πολλαπλών στρωμάτων προστασίας. Σε πρακτικό επίπεδο, αυτό σημαίνει ότι ακόμη και αν παραβιαστεί το ένα επίπεδο - για παράδειγμα αν ένας κακόβουλος δράστης παρακάμψει τα frontend controls - τα επόμενα επίπεδα (επικύρωση στο API, ownership checks στο business logic, database constraints) θα αποτρέψουν τη μη εξουσιοδοτημένη πρόσβαση σε δεδομένα. Δεύτερη βασική αρχή είναι αυτή του **least privilege**, σύμφωνα με την οποία κάθε χρήστης, διεργασία ή υπηρεσία διαθέτει αποκλειστικά τα δικαιώματα που χρειάζεται για να εκτελέσει τη λειτουργία της - και τίποτα περισσότερο.

Η τρίτη αρχή αφορά τον **διαχωρισμό των ευθυνών (separation of duties)**, που αποτυπώνεται τόσο στην οργάνωση των API scopes (/store/, /vendor/, /admin/\*) όσο και στη διαχείριση των ευαίσθητων δεδομένων. Παράδειγμα εφαρμογής της αρχής αυτής είναι η ανάθεση της επεξεργασίας στοιχείων κάρτας πληρωμής στη Stripe, ώστε η ίδια η πλατφόρμα να μην αναλαμβάνει την ευθύνη διαχείρισης τέτοιας ευαίσθητης πληροφορίας. Η τέταρτη αρχή είναι αυτή της **fail-safe λειτουργίας**: όταν προκύπτει αμφισημία ή σφάλμα, το σύστημα παίρνει την πιο **συντηρητική απόφαση**, αρνούμενο πρόσβαση μέχρι να εξακριβωθεί η νομιμότητα της ενέργειας. Με αυτόν τον τρόπο, ένα bug ή μια απρόβλεπτη κατάσταση δεν μετατρέπεται σε ευπάθεια ασφαλείας.

### 9.2. Αυθεντικοποίηση & Εξουσιοδότηση

Η αυθεντικοποίηση και η εξουσιοδότηση των χρηστών αποτελούν τη **βασική πύλη ελέγχου** της πλατφόρμας και υλοποιούνται μέσω συνδυασμού σύγχρονων μηχανισμών. Η κύρια μέθοδος αυθεντικοποίησης βασίζεται σε **JSON Web Tokens (JWT)**, τα οποία εκδίδονται μετά από επιτυχή εισαγωγή διαπιστευτηρίων και ενσωματώνουν κρυπτογραφικά υπογεγραμμένες πληροφορίες για την ταυτότητα και τον ρόλο του χρήστη. Τα JWT tokens διαθέτουν περιορισμένη διάρκεια ζωής (expiration), εξασφαλίζοντας ότι ακόμη και σε περίπτωση υποκλοπής, η χρήση τους είναι χρονικά οριοθετημένη. Για τις εφαρμογές κινητού, τα tokens αποθηκεύονται σε ασφαλή χώρο της συσκευής (Keychain σε iOS, EncryptedSharedPreferences σε Android), αποτρέποντας την πρόσβαση από κακόβουλες εφαρμογές. Παράλληλα, η πλατφόρμα αξιοποιεί **publishable API keys** ως δεύτερο επίπεδο αναγνώρισης των εφαρμογών-πελατών, εξασφαλίζοντας ότι μόνο εξουσιοδοτημένα frontend περιβάλλοντα μπορούν να επικοινωνούν με τα /store/\*

endpoints.

Πάνω από τη βασική αυθεντικοποίηση εφαρμόζεται ένα ολοκληρωμένο σύστημα **Role-Based Access Control (RBAC)**, το οποίο καθορίζει τι μπορεί να κάνει κάθε αυθεντικοποιημένος χρήστης ανάλογα με τον ρόλο του. Το RBAC του REFASHION ορίζει τέσσερις βασικούς ρόλους - **Customer, Seller, Admin και System Service** - καθένας από τους οποίους έχει σαφώς οριοθετημένα δικαιώματα πρόσβασης. Ένας Customer μπορεί να δει και να διαχειριστεί μόνο τις δικές του παραγγελίες και τα στοιχεία προφίλ του· ένας Seller μπορεί να δει και να διαχειριστεί μόνο τα δικά του προϊόντα και τις παραγγελίες που τον αφορούν· ένας Admin διαθέτει εποπτικά δικαιώματα στο σύνολο της πλατφόρμας, αλλά με εξειδικευμένα όρια ανά κατηγορία ενέργειας· τέλος, οι System Services (όπως οι AI microservices) διαθέτουν αυστηρά περιορισμένα δικαιώματα, συχνά μόνο για ανάγνωση συγκεκριμένων δεδομένων.

Σημαντική αρχιτεκτονική επιλογή είναι ότι οι κανόνες εξουσιοδότησης **επιβάλλονται σε πολλαπλά επίπεδα**: στο API middleware (που απορρίπτει αιτήματα χωρίς έγκυρη αυθεντικοποίηση), στα services της επιχειρησιακής λογικής (που επιβάλλουν τους εξειδικευμένους ownership checks), και στα database roles της PostgreSQL (που εξασφαλίζουν ότι ακόμη και απευθείας συνδέσεις στη βάση δεν μπορούν να παρακάμψουν τους κανόνες). Η πολυεπίπεδη αυτή επιβολή αποτελεί έμπρακτη εφαρμογή της αρχής defense-in-depth που αναφέρθηκε προηγουμένως, και εξασφαλίζει ότι **καμία πτυχή του συστήματος** δεν αποτελεί μοναδικό σημείο αποτυχίας.

### 9.3. Προστασία Επικοινωνιών & Δεδομένων

Όλη η επικοινωνία μεταξύ των frontend περιβαλλόντων και του backend της πλατφόρμας REFASHION πραγματοποιείται **αποκλειστικά μέσω κρυπτογραφημένου καναλιού** με χρήση του πρωτοκόλλου **HTTPS/TLS**. Η επιβολή αυτή δεν αποτελεί απλή σύσταση αλλά **αυστηρή απαίτηση** του συστήματος: αιτήματα που υποβάλλονται μέσω μη κρυπτογραφημένου HTTP απορρίπτονται αυτόματα και ανακατευθύνονται στην ασφαλή έκδοσή τους. Η κρυπτογράφηση εξασφαλίζει ότι ακόμη και αν τα δεδομένα υποκλαπούν κατά τη διάρκεια της μεταφοράς, παραμένουν αναγνώσιμα μόνο από τους εξουσιοδοτημένους αποδέκτες, προστατεύοντας τόσο τα στοιχεία αυθεντικοποίησης όσο και κάθε ευαίσθητη πληροφορία. Συμπληρωματικά, εφαρμόζονται μηχανισμοί όπως οι **HTTP Strict Transport Security (HSTS) headers** που εξασφαλίζουν ότι οι browsers θα επικοινωνούν με την πλατφόρμα αποκλειστικά μέσω HTTPS ακόμη και αν ο χρήστης πληκτρολογήσει το URL χωρίς το πρόθεμα.

Πέρα από την προστασία της μετάδοσης, η αρχιτεκτονική του REFASHION ενσωματώνει μηχανισμούς **προστασίας των αποθηκευμένων δεδομένων (data at rest)**. Τα κρίσιμα ευαίσθητα στοιχεία όπως οι κωδικοί πρόσβασης δεν αποθηκεύονται ποτέ ως απλό κείμενο, αλλά μετά από εφαρμογή ισχυρών αλγορίθμων κρυπτογραφικής σύνοψης (hashing) με αλατοποίηση (salting), όπως ο bcrypt. Με τον τρόπο αυτό, ακόμη και σε περίπτωση πρόσβασης κακόβουλου παράγοντα στη βάση δεδομένων, η ανάκτηση των πραγματικών κωδικών παραμένει υπολογιστικά πρακτικά αδύνατη. Παράλληλα, η ίδια η βάση δεδομένων προστατεύεται από κρυπτογράφηση σε επίπεδο αποθηκευτικού μέσου (encryption at rest), εξασφαλίζοντας ότι το φυσικό υλικό της υποδομής δεν αποτελεί ευπάθεια. Σε επίπεδο εφαρμογής εφαρμόζονται επιπρόσθετα μηχανισμοί προστασίας από κοινούς τύπους επιθέσεων: **SQL injection** αποτρέπεται μέσω της χρήσης του ORM και των parameterized queries, **Cross-Site Scripting (XSS)** μέσω της σχολαστικής επικύρωσης και του sanitization των inputs, και **Cross-Site Request Forgery (CSRF)** μέσω εξειδικευμένων tokens επιβεβαίωσης για κρίσιμες ενέργειες.

#### 9.4. Ασφάλεια Πληρωμών & PCI DSS

Η ασφαλής διαχείριση των πληρωμών αποτελεί μια από τις **πιο ευαίσθητες πτυχές** μιας πλατφόρμας ηλεκτρονικού εμπορίου, καθώς η διαρροή στοιχείων κάρτας έχει άμεσες οικονομικές και νομικές συνέπειες τόσο για την πλατφόρμα όσο και για τους χρήστες της. Η αρχιτεκτονική του REFASHION αντιμετωπίζει αυτή την πρόκληση μέσω της **πλήρους εξωτερικεύσης** της διαχείρισης ευαίσθητων δεδομένων πληρωμής στην εξειδικευμένη πλατφόρμα **Stripe**, η οποία διαθέτει την υψηλότερη πιστοποίηση **PCI DSS Level 1**. Η προσέγγιση αυτή αξιοποιεί τη μηχανική του **Stripe.js SDK** στο frontend, μέσω της οποίας τα στοιχεία κάρτας του αγοραστή αποστέλλονται **απευθείας** από τον browser στους servers της Stripe, χωρίς να διέρχονται ποτέ από την υποδομή του REFASHION.

Πρακτικά, αυτό σημαίνει ότι η πλατφόρμα **απαλλάσσεται από το μεγαλύτερο μέρος των απαιτήσεων** του προτύπου PCI DSS, καθώς δεν αποθηκεύει, δεν επεξεργάζεται και ούτε καν μεταδίδει ευαίσθητα δεδομένα κάρτας. Όλα αυτά διεκπεραιώνονται από τη Stripe, η οποία επιστρέφει στην πλατφόρμα μόνο **ασφαλή tokens αναφοράς** που μπορούν να αξιοποιηθούν για τη συσχέτιση με τις παραγγελίες χωρίς να αντιπροσωπεύουν αυτά τα ίδια τα στοιχεία της κάρτας. Παράλληλα, η ενσωμάτωση με το **Stripe Connect** για το multi-vendor μοντέλο εξασφαλίζει ότι οι **πληρωμές προς τους πωλητές** πραγματοποιούνται απευθείας από τη Stripe προς τους τραπεζικούς λογαριασμούς τους, χωρίς να μεσολαβεί η πλατφόρμα ως οικονομικός φορέας. Η αρχιτεκτονική αυτή επιλογή έχει διττή αξία: μειώνει δραστικά τις **κανονιστικές υποχρεώσεις** της πλατφόρμας και ταυτόχρονα προσφέρει στους πωλητές τη μέγιστη δυνατή **διαφάνεια** για τις οικονομικές ροές τους. Επιπρόσθετα, η ασύγχρονη ενημέρωση της κατάστασης των συναλλαγών μέσω των **Stripe webhooks** συνοδεύεται από κρυπτογραφική επαλήθευση υπογραφής, αποτρέποντας τυχόν προσπάθειες πλαστογράφησης από κακόβουλους τρίτους.

#### 9.5. Συμμόρφωση με GDPR

Η συμμόρφωση με τον **Γενικό Κανονισμό Προστασίας Δεδομένων (GDPR)** αποτελεί όχι μόνο νομική υποχρέωση αλλά και **ηθική δέσμευση** του REFASHION προς τους χρήστες του. Η αρχιτεκτονική της πλατφόρμας ενσωματώνει αρχές **data governance** που υποστηρίζουν την πλήρη συμμόρφωση με τον κανονισμό σε όλες τις διαστάσεις του. Πρωταρχικός μηχανισμός εφαρμογής είναι το **data lineage tracking**: κάθε στοιχείο προσωπικού δεδομένου που εισέρχεται στο σύστημα συνοδεύεται από μεταδεδομένα που καταγράφουν την πηγή προέλευσής του, τη νόμιμη βάση επεξεργασίας (συγκατάθεση χρήστη, εκτέλεση σύμβασης, έννομο συμφέρον), και τις διαδρομές μέσω των οποίων αυτό μεταφέρεται και επεξεργάζεται από διαφορετικά υποσυστήματα.

Η πλατφόρμα υλοποιεί έμπρακτα τα **βασικά δικαιώματα** που παρέχει ο GDPR στους χρήστες. Το **δικαίωμα πρόσβασης (right of access)** εξυπηρετείται μέσω εξειδικευμένων προφίλ λειτουργιών που επιτρέπουν στον χρήστη να δει το σύνολο των δεδομένων που διατηρεί η πλατφόρμα για αυτόν. Το **δικαίωμα διόρθωσης (right to rectification)** ικανοποιείται μέσω της δυνατότητας ενημέρωσης των στοιχείων προφίλ μέσω των αντίστοιχων ροών αυτο-εξυπηρέτησης. Το **δικαίωμα διαγραφής (right to erasure / "right to be forgotten")** υλοποιείται μέσω εξειδικευμένης ροής διαγραφής λογαριασμού χρήστη που πυροδοτεί την αντίστοιχη διαδικασία αφαίρεσης ή ανωνυμοποίησης των προσωπικών δεδομένων από όλα τα υποσυστήματα της πλατφόρμας. Το **δικαίωμα φορητότητας (right to data portability)** εξυπηρετείται μέσω της δυνατότητας εξαγωγής των δεδομένων του χρήστη σε δομημένη, ευρέως χρησιμοποιούμενη μορφή.

Σημαντικό αρχιτεκτονικό χαρακτηριστικό αποτελεί η **auditability** όλων των κρίσιμων

ενεργειών της πλατφόρμας. Κάθε πρόσβαση σε ευαίσθητα δεδομένα, κάθε αλλαγή σε προφίλ χρήστη, κάθε διοικητική παρέμβαση από admin καταγράφεται σε ξεχωριστά **audit logs** που διατηρούνται για προκαθορισμένη περίοδο και δεν επιτρέπουν τη μεταβολή τους από κανέναν, ακόμη και από τους ίδιους τους διαχειριστές της πλατφόρμας. Τα audit logs αξιοποιούνται τόσο για την **ανίχνευση** ύποπτων συμπεριφορών όσο και ως **αποδεικτικό στοιχείο** συμμόρφωσης σε περίπτωση κανονιστικού ελέγχου.

Η αρχιτεκτονική εφαρμόζει επίσης την αρχή της **ελαχιστοποίησης δεδομένων (data minimization)**: από τους χρήστες ζητούνται μόνο τα **απολύτως απαραίτητα** στοιχεία για κάθε λειτουργία, και αποφεύγεται η συλλογή πληροφοριών "για παν ενδεχόμενο" που θα μπορούσαν να αξιοποιηθούν στο μέλλον. Παράλληλα, εφαρμόζονται **πολιτικές διατήρησης (data retention policies)** που εξασφαλίζουν την αυτόματη διαγραφή ή ανωνυμοποίηση των δεδομένων μετά τη λήξη της περιόδου για την οποία είναι απαραίτητα. Σε εξειδικευμένες περιπτώσεις όπου τα δεδομένα χρειάζονται για στατιστικούς ή αναλυτικούς σκοπούς πέραν της περιόδου διατήρησης, αυτά υφίστανται **ανωνυμοποίηση (anonymization)** ή **ψευδωνυμοποίηση (pseudonymization)**, ώστε να μην μπορούν να αποδοθούν σε συγκεκριμένα φυσικά πρόσωπα.

Τέλος, σε ότι αφορά τη διαχείριση **συγκαταθέσεων (consent management)**, η πλατφόρμα εφαρμόζει σαφείς και διακριτές διαδικασίες για τη συλλογή της ρητής συγκατάθεσης του χρήστη όπου αυτή απαιτείται - για παράδειγμα, για τη χρήση cookies, για την αποστολή marketing επικοινωνιών, ή για την εξαγωγή δεδομένων προς τρίτους παρόχους AI υπηρεσιών. Οι συγκαταθέσεις καταγράφονται με χρονοσφραγίδα και είναι **ανακαλέσιμες** ανά πάσα στιγμή μέσω των ρυθμίσεων του προφίλ, σύμφωνα με τη γενική αρχή του GDPR ότι **η συγκατάθεση πρέπει να είναι εξίσου εύκολο να αποσυρθεί όσο και να δοθεί**. Με τη συνδυαστική εφαρμογή όλων αυτών των μηχανισμών, η αρχιτεκτονική του REFASHION εξασφαλίζει ότι η συμμόρφωση με τον GDPR δεν αποτελεί απλή τυπική απαίτηση, αλλά **οργανική συνιστώσα** του τρόπου με τον οποίο σχεδιάζεται και λειτουργεί η πλατφόρμα.

## 10. Ανάπτυξη & Φιλοξενία (Deployment)

### 10.1. Τρέχουσα Υποδομή Φιλοξενίας

Η πλατφόρμα REFASHION φιλοξενείται σε **self-hosted server του Πανεπιστημίου Πελοποννήσου**, ο οποίος είναι διαθέσιμος στη δικτυακή διεύθυνση **195.251.8.72**. Η επιλογή της φιλοξενίας σε υποδομή του ίδιου του ακαδημαϊκού φορέα - αντί της προσφυγής σε εμπορικό cloud provider - αποτελεί συνειδητή επιλογή για την παρούσα φάση του έργου, καθώς εξυπηρετεί συγκεκριμένες ανάγκες: τον **πλήρη έλεγχο της υποδομής** από την ομάδα ανάπτυξης, την **απουσία επαναλαμβανόμενου κόστους** σε εμπορικές υπηρεσίες κατά τη διάρκεια του ερευνητικού σταδίου, και τη **διαθεσιμότητα ακαδημαϊκών δικτυακών πόρων** του ιδρύματος. Ο server λειτουργεί ως **ενιαίο σημείο φιλοξενίας** για το σύνολο των υπηρεσιών του οικοσυστήματος - Medusa/Mercur backend, βάση δεδομένων PostgreSQL, μηχανή αναζήτησης, AI microservices και βοηθητικές υπηρεσίες - εξυπηρετώντας τόσο τις development όσο και τις πιλοτικές παραγωγικές ροές της πλατφόρμας.

Σε επίπεδο **πακετοποίησης και διαχείρισης υπηρεσιών**, η αρχιτεκτονική βασίζεται στη χρήση **Docker containers** για κάθε διακριτή υπηρεσία της πλατφόρμας. Κάθε υπηρεσία - από το Medusa backend έως τα AI microservices - συσκευάζεται σε αυτόνομη εικόνα container που περιλαμβάνει τόσο τον κώδικα της εφαρμογής όσο και όλες τις εξαρτήσεις της. Η προσέγγιση αυτή εξασφαλίζει την **αναπαραγωγιμότητα του περιβάλλοντος εκτέλεσης** μεταξύ διαφορετικών μηχανημάτων και διαφορετικών developers, αποτρέποντας τις κλασικές δυσκολίες που προκύπτουν από διαφορές στις εκδόσεις βιβλιοθηκών, στις παραμέτρους λειτουργικού συστήματος ή στις τοπικές ρυθμίσεις. Η συνολική ενορχήστρωση των containers υλοποιείται μέσω **Docker Compose**, που επιτρέπει τη δηλωτική περιγραφή του συνόλου των υπηρεσιών, των δικτυακών τους διασυνδέσεων, των volumes για persistence δεδομένων και των μεταξύ τους εξαρτήσεων μέσα σε ένα ενιαίο αρχείο διαμόρφωσης. Η εκκίνηση, σταμάτημα και αναβάθμιση της πλατφόρμας μπορούν να εκτελεστούν με λίγες εντολές, διευκολύνοντας σημαντικά τις διαδικασίες ανάπτυξης και συντήρησης.

Η **πρόσβαση των frontend περιβαλλόντων** στις backend υπηρεσίες διαμεσολαβείται από έναν **reverse proxy** που τοποθετείται μπροστά από όλες τις εκτεθειμένες υπηρεσίες. Ο reverse proxy αναλαμβάνει πολλαπλούς κρίσιμους ρόλους: εξασφαλίζει την **τερματισμό TLS** για την κρυπτογραφημένη επικοινωνία HTTPS, εκτελεί **routing** των εισερχόμενων αιτημάτων στην κατάλληλη εσωτερική υπηρεσία ανάλογα με το hostname ή το URL path, εφαρμόζει βασικούς ελέγχους **rate limiting** για την αποτροπή κατάχρησης, και παρέχει ένα **ενιαίο σημείο εισόδου** προς όλο το οικοσύστημα, αποκρύπτοντας την εσωτερική δομή των υπηρεσιών από τους εξωτερικούς πελάτες. Η αρχιτεκτονική αυτή εξασφαλίζει ότι οι εσωτερικές υπηρεσίες (PostgreSQL, AI microservices, εσωτερικά endpoints του Medusa) **δεν εκτίθενται απευθείας στο διαδίκτυο**, αλλά είναι προσβάσιμες αποκλειστικά μέσω του reverse proxy, ενισχύοντας σημαντικά την ασφάλεια της εγκατάστασης.

### 10.2. Παράμετροι Πρόσβασης

Σε επίπεδο πρακτικής λειτουργίας, το backend της πλατφόρμας είναι διαθέσιμο μέσω του reverse proxy στη διεύθυνση **http://195.251.8.72:9000** για το τρέχον development environment. Όλα τα frontend περιβάλλοντα - Web Storefront, εφαρμογή κινητού, Seller Panel, Admin Panel - διαμορφώνονται κατά την εκτέλεσή τους ώστε να επικοινωνούν με αυτό το endpoint για την υποβολή των API αιτημάτων τους. Η αυθεντικοποίηση των αιτημάτων προς τα /store/\* endpoints απαιτεί τη χρήση **publishable API key**, το οποίο

παρέχεται μέσω του Medusa Admin Panel για κάθε deployment environment ξεχωριστά. Με τον τρόπο αυτό, εξασφαλίζεται ότι μόνο εξουσιοδοτημένα frontend περιβάλλοντα μπορούν να αλληλεπιδρούν με την πλατφόρμα, και ότι τα keys κάθε environment παραμένουν απομονωμένα μεταξύ τους.

Σε επίπεδο διαχείρισης παραμέτρων, οι ευαίσθητες ρυθμίσεις της πλατφόρμας - συνθηματικά βάσης δεδομένων, JWT secrets, API keys προς εξωτερικές υπηρεσίες, διευθύνσεις blockchain wallets - αποθηκεύονται σε **environment variables** που εγχέονται στα containers κατά την εκκίνησή τους, και ποτέ δεν αποθηκεύονται στον κώδικα της εφαρμογής. Η προσέγγιση αυτή εξασφαλίζει τόσο την ασφαλή διαχείριση των διαπιστευτηρίων όσο και την ευελιξία αλλαγής τους χωρίς επαναμεταγλώττιση του κώδικα.

### 10.3. Σχεδιαστικές Στοχεύσεις για Μελλοντική Παραγωγική Λειτουργία

Η τρέχουσα υποδομή φιλοξενίας εξυπηρετεί ικανοποιητικά τις ανάγκες της **πilotικής φάσης** του έργου και επιτρέπει την ομαλή ανάπτυξη και αξιολόγηση της πλατφόρμας. Ωστόσο, η μετάβαση σε **πλήρως παραγωγική λειτουργία** με μεγάλο όγκο πραγματικών χρηστών και συναλλαγών θα απαιτήσει συγκεκριμένες αρχιτεκτονικές αναβαθμίσεις, οι οποίες έχουν προβλεφθεί ως **σχεδιαστικές στοχεύσεις** της πλατφόρμας. Η containerized αρχιτεκτονική της υπάρχουσας υποδομής διευκολύνει σημαντικά τη μελλοντική μετάβαση σε πιο προηγμένα περιβάλλοντα ενορχήστρωσης, καθώς οι ίδιες εικόνες Docker που χρησιμοποιούνται σήμερα στο Docker Compose μπορούν να αναπτυχθούν σε πλατφόρμες όπως το Kubernetes χωρίς να απαιτούνται ουσιαστικές τροποποιήσεις στις επιμέρους υπηρεσίες.

Μεταξύ των σχεδιαστικών στοχεύσεων για την παραγωγική φάση περιλαμβάνεται η μετάβαση σε **container orchestration platform** (όπως Kubernetes ή Docker Swarm) που θα προσφέρει **ελαστική κλιμάκωση** των υπηρεσιών ανάλογα με τον φόρτο, **αυτοματοποιημένους μηχανισμούς ανάκαμψης** σε περίπτωση αστοχίας containers, και **υψηλή διαθεσιμότητα** μέσω κατανομής λειτουργίας σε πολλαπλούς nodes. Παράλληλα, για τα AI microservices που εκτελούν υπολογιστικά απαιτητικές λειτουργίες (επεξεργασία εικόνας, LLM inference), προβλέπεται η αξιοποίηση **GPU-enabled υποδομής** που θα επιτρέψει σημαντικά γρηγορότερη και πιο κλιμακώσιμη εκτέλεση των σχετικών αλγορίθμων. Επιπρόσθετα, αναμένεται η ενσωμάτωση ενός **πλήρους API Gateway layer** που, πέρα από τις ήδη υπάρχουσες λειτουργίες του reverse proxy, θα παρέχει κεντροποιημένα χαρακτηριστικά όπως αναβαθμισμένη αυθεντικοποίηση, λεπτομερές rate limiting ανά χρήστη και ενδεδειγμένο monitoring της κίνησης. Όλα αυτά τα στοιχεία θα μπορούν να ενσωματωθούν **σταδιακά** στην πλατφόρμα, καθώς οι ανάγκες του έργου εξελίσσονται από το pilotικό προς το παραγωγικό στάδιο.

## 11. Μεθοδολογία Υλοποίησης

### 11.1. Φιλοσοφία Σταδιακής Υλοποίησης

Η ανάπτυξη της πλατφόρμας REFASHION ακολουθεί μια **σταδιακή (phased) μεθοδολογία**, σύμφωνα με την οποία το σύνολο του έργου χωρίζεται σε διακριτές φάσεις με σαφώς οριοθετημένο περιεχόμενο, παραδοτέα και ορόσημα. Η επιλογή αυτής της προσέγγισης δεν είναι συμπτωματική: σε ένα έργο που συνδυάζει ένα σύνθετο οικοσύστημα τεχνολογιών - από headless commerce framework έως blockchain integration και υπηρεσίες Τεχνητής Νοημοσύνης - η **παράλληλη ανάπτυξη όλων των συνιστωσών** θα δημιουργούσε ανυπερέβλητο επίπεδο πολυπλοκότητας και θα δυσχέραινε τόσο τη συντονισμένη λειτουργία των ομάδων όσο και τον έλεγχο ποιότητας των επιμέρους αποτελεσμάτων.

Η σταδιακή μεθοδολογία προσφέρει σημαντικά πλεονεκτήματα έναντι μιας **μονολιθικής προσέγγισης**. Πρώτον, εξασφαλίζει ότι κάθε νέα φάση **εδράζεται σε ώριμα και σταθεροποιημένα θεμέλια** που έχουν εδραιωθεί στις προηγούμενες, αποτρέποντας τη συσσώρευση τεχνικού χρέους. Δεύτερον, επιτρέπει την **ενδιάμεση αξιολόγηση** των επιτευγμάτων και την επαναπροσαρμογή των επόμενων βημάτων με βάση τη συσσωρευμένη εμπειρία και τα ευρήματα της κάθε φάσης. Τρίτον, διευκολύνει τη **σαφή κατανομή ευθυνών** μεταξύ των διαφορετικών ομάδων του έργου, με κάθε ομάδα να γνωρίζει με ακρίβεια το αντικείμενο και τα όρια της δικής της εμπλοκής. Τέλος, υποστηρίζει την **εκπλήρωση των συμβατικών παραδοτέων** του έργου σε προβλεπόμενα χρονικά ορόσημα, εξυπηρετώντας τις απαιτήσεις τόσο της εσωτερικής διοίκησης όσο και του φορέα χρηματοδότησης.

Σημαντική παρατήρηση είναι ότι, παρά τη γραμμική παρουσίαση των φάσεων στον ακόλουθο πίνακα, η πραγματική εκτέλεση του έργου παρουσιάζει **σημαντικές χρονικές επικαλύψεις** μεταξύ διαδοχικών φάσεων. Ενώ μια φάση βρίσκεται σε προχωρημένο στάδιο υλοποίησης, η επόμενη ξεκινά παράλληλα τις αρχικές της εργασίες - σχεδιαστική προεργασία, τεχνολογική διερεύνηση, οργάνωση πόρων - εξασφαλίζοντας ομαλή μετάβαση και βέλτιστη αξιοποίηση των διαθέσιμων ομάδων. Η οργανωτική αυτή ευελιξία είναι κρίσιμη για την ομαλή ροή ενός σύνθετου ερευνητικού έργου, χωρίς ωστόσο να υπονομεύει τη σαφή λογική διαδοχή των φάσεων.

### 11.2. Δομή των Φάσεων Υλοποίησης

Η μεθοδολογία υλοποίησης του REFASHION αποτυπώνεται σε **πέντε διακριτές φάσεις**, καθεμία από τις οποίες εστιάζει σε διαφορετικό στρώμα της συνολικής λύσης. Ο ακόλουθος πίνακας συνοψίζει τη δομή των φάσεων:

Φάση	Τίτλος	Περιεχόμενο
<b>Φάση 1</b>	Έρευνα Σχεδιασμός	& Ανάλυση τεχνολογιών, ορισμός απαιτήσεων, αρχιτεκτονική προδιαγραφή
<b>Φάση 2</b>	Core Platform	UI πλατφόρμας, transaction management, API υποδομή
<b>Φάση 3</b>	AI Services	Image analysis, Visual Product Intelligence, predictive analytics

<b>Φάση 4</b>	Data Infrastructure	Streaming pipelines, distributed storage
<b>Φάση 5</b>	Testing & Optimization	Functional, performance & security testing, ML model optimization

Η **Φάση 1 (Έρευνα & Σχεδιασμός)** αποτελεί το θεμελιώδες προπαρασκευαστικό στάδιο του έργου, κατά το οποίο αναλύονται οι διαθέσιμες τεχνολογίες, οριστικοποιούνται οι λειτουργικές και μη-λειτουργικές απαιτήσεις, και διαμορφώνεται η συνολική αρχιτεκτονική προδιαγραφή της πλατφόρμας. Σε αυτή τη φάση λαμβάνονται κρίσιμες αρχιτεκτονικές αποφάσεις - όπως η επιλογή του MedusaJS/Mercur ως τεχνολογικής βάσης, η υιοθέτηση headless και composable αρχιτεκτονικής, και ο σχεδιασμός του μοντέλου δεδομένων - που επηρεάζουν το σύνολο του έργου και αποτυπώνονται στα τεχνικά παραδοτέα του ορόσημου. Η ποιότητα και η ακρίβεια των αποφάσεων αυτής της φάσης έχει **πολλαπλασιαστική επίπτωση** στις επόμενες, και για τον λόγο αυτό της αφιερώνεται ο απαραίτητος χρόνος ωρίμανσης.

Η **Φάση 2 (Core Platform)** αναλαμβάνει την υλοποίηση του **εμπορικού πυρήνα** της πλατφόρμας, που αποτελεί τη βάση πάνω στην οποία θα οικοδομηθούν οι καινοτόμες λειτουργίες των επόμενων φάσεων. Σε αυτή τη φάση αναπτύσσονται οι βασικές διεπαφές χρήστη (storefront, seller panel, admin panel, εφαρμογή κινητού), υλοποιείται η ροή των συναλλαγών (cart, checkout, order management, fulfillment), και διαμορφώνεται η συνολική API υποδομή που θα εξυπηρετεί όλους τους πελάτες της πλατφόρμας. Στο τέλος αυτής της φάσης, η πλατφόρμα είναι ήδη λειτουργική ως marketplace, χωρίς ωστόσο τις διαφοροποιητικές προεκτάσεις που αναπτύσσονται στις επόμενες φάσεις.

Η **Φάση 3 (AI Services)** εστιάζει στην ανάπτυξη των **υπηρεσιών Τεχνητής Νοημοσύνης** που διαφοροποιούν το REFASHION από συμβατικές πλατφόρμες ηλεκτρονικού εμπορίου. Σε αυτή τη φάση υλοποιούνται και ενσωματώνονται τα τρία βασικά AI microservices - το Image Analysis Service για την επεξεργασία εικόνων προϊόντων, το Visual Product Intelligence Service για την οπτική αναζήτηση και τον εμπλουτισμό καταχωρήσεων, και το Recommendation Service για τις προσωποποιημένες προτάσεις. Παράλληλα, εφαρμόζονται οι αλγόριθμοι predictive analytics που υποστηρίζουν τόσο τις δυνατότητες του Recommendation Service όσο και τα συγκεντρωτικά αναλυτικά εργαλεία της πλατφόρμας.

Η **Φάση 4 (Data Infrastructure)** ενισχύει την υποδομή δεδομένων της πλατφόρμας με τα συστήματα που θα επιτρέψουν την κλιμάκωσή της σε μεγάλους όγκους χρηστών και συναλλαγών. Η ενσωμάτωση **distributed storage** για την αποδοτική διαχείριση μεγάλων δεδομένων αποτελεί το κύριο αντικείμενο αυτής της φάσης. Σε αυτό το στάδιο, η πλατφόρμα αποκτά τα τεχνολογικά θεμέλια για βιώσιμη λειτουργία σε παραγωγικό περιβάλλον.

Η **Φάση 5 (Testing & Optimization)** ολοκληρώνει τον κύκλο ανάπτυξης με τη συστηματική επικύρωση και βελτιστοποίηση όλων των επιμέρους συνιστωσών. Εκτελείται **functional testing** που επαληθεύει την ορθή λειτουργία κάθε χαρακτηριστικού, **performance testing** που μετρά τη συμπεριφορά της πλατφόρμας υπό υψηλό φόρτο, και **security testing** που εντοπίζει τυχόν ευπάθειες πριν τη μετάβαση σε παραγωγική χρήση. Παράλληλα, εκτελείται **βελτιστοποίηση των ML μοντέλων** που χρησιμοποιούνται στα AI microservices, με στόχο τη βελτίωση τόσο της ακρίβειας των αποτελεσμάτων όσο και της αποδοτικότητας στη χρήση υπολογιστικών πόρων.

### 11.3. 11.3 Συντονισμός Ομάδων και Διαχείριση Έργου

Η εφαρμογή της σταδιακής μεθοδολογίας απαιτεί **συστηματικό συντονισμό** μεταξύ των διαφορετικών ομάδων που συμμετέχουν στο έργο. Στο REFASHION, η ανάπτυξη κατανέμεται σε πολλαπλές ομάδες με διακριτές ευθύνες: η κύρια ομάδα του Πανεπιστημίου Πελοποννήσου (Disyd Lab) αναλαμβάνει την αρχιτεκτονική σχεδίαση, την ανάπτυξη του backend και των web περιβαλλόντων.μ Ο συντονισμός αυτής της κατανομημένης ανάπτυξης διευκολύνεται σημαντικά από την **headless και composable αρχιτεκτονική** της πλατφόρμας, η οποία επιτρέπει στις διάφορες ομάδες να εργάζονται **παράλληλα και ανεξάρτητα** πάνω σε διαφορετικά τμήματα του οικοσυστήματος, με τη συνέπεια να εξασφαλίζεται μέσω σαφώς ορισμένων διεπαφών API.

Η οργάνωση του έργου ακολουθεί σύγχρονες πρακτικές διαχείρισης λογισμικού, με τη χρήση εργαλείων όπως version control συστήματα για τη συνεργατική ανάπτυξη του κώδικα, issue trackers για την παρακολούθηση των εργασιών και ταχείες κύκλους ανατροφοδότησης μεταξύ των μελών της ομάδας. Στο τέλος κάθε φάσης παράγονται **τεκμηριωμένα παραδοτέα** που αποτυπώνουν τα επιτεύγματα και τη συνολική πορεία, τα οποία αποτελούν τόσο εργαλείο εσωτερικής αξιολόγησης όσο και επίσημες αποδείξεις προς τον φορέα χρηματοδότησης και τους ενδιαφερόμενους φορείς.

## 12. Συμπεράσματα

### 12.1. Επισκόπηση Επιτευγμάτων

Η ολοκλήρωση του παρόντος παραδοτέου σηματοδοτεί ένα **κρίσιμο ορόσημο** στην εξέλιξη του έργου REFASHION και αποτυπώνει μια ουσιαστική μετάβαση από τη φάση του αρχιτεκτονικού σχεδιασμού στη φάση της απτής υλοποίησης. Κατά το διάστημα που οδήγησε στο ορόσημο M20, έχει επιτευχθεί η ανάπτυξη μιας **λειτουργικά πλήρους και αρχιτεκτονικά συνεκτικής ηλεκτρονικής πλατφόρμας επανεμπορίου**, η οποία ενσωματώνει με συστηματικό τρόπο τόσο τις θεμελιώδεις δυνατότητες ενός σύγχρονου marketplace πολλαπλών πωλητών όσο και τα διαφοροποιητικά καινοτόμα στοιχεία που αποτελούν την ταυτότητα του έργου.

Ο **εμπορικός πυρήνας** της πλατφόρμας, βασισμένος στον συνδυασμό MedusaJS και Mergur, λειτουργεί ολοκληρωμένα και υποστηρίζει το σύνολο των απαραίτητων ροών - από τη διαχείριση προφίλ χρηστών και πωλητών, την καταχώρηση και διαχείριση προϊόντων, την εκτέλεση συναλλαγών μέσω καλαθιού, την ολοκλήρωση πληρωμών μέσω Stripe Connect, έως τη διαχείριση αποστολών και επιστροφών. Η headless και composable αρχιτεκτονική που υιοθετήθηκε αποδείχθηκε **καθοριστική στρατηγική επιλογή**, καθώς επέτρεψε την παράλληλη ανάπτυξη του backend από την ομάδα του Πανεπιστημίου Πελοποννήσου και της εφαρμογής κινητού από την εξωτερική ομάδα Cyclefy, με τη συνεκτικότητα να εξασφαλίζεται μέσω καθαρά ορισμένων διεπαφών REST API.

Παράλληλα με τον εμπορικό πυρήνα, έχουν αναπτυχθεί και ενσωματωθεί τα **τρία frontend περιβάλλοντα** που εξυπηρετούν τις διαφορετικές κατηγορίες χρηστών - το Web Storefront σε Next.js, το Seller Admin Panel και το Medusa Admin Panel - ενώ η εφαρμογή κινητού βρίσκεται σε προχωρημένο στάδιο ενσωμάτωσης με το backend. Η συνύπαρξη όλων αυτών των διεπαφών κάτω από κοινό σχεδιαστικό πλαίσιο και κοινή λογική δεδομένων αποτελεί έμπρακτη απόδειξη της επιτυχούς εφαρμογής της αρχιτεκτονικής φιλοσοφίας που αναλύθηκε στα προηγούμενα κεφάλαια.

Σε επίπεδο **καινοτόμων λειτουργιών**, η πλατφόρμα έχει θέσει τις τεχνολογικές βάσεις για την ενσωμάτωση των διαφοροποιητικών στοιχείων του REFASHION. Η αρχιτεκτονική των custom plugins - refashion-blockchain-plugin, refashion-eco-impact-plugin, refashion-gamification-plugin - έχει σχεδιαστεί και υλοποιείται σταδιακά, εξασφαλίζοντας ότι οι μηχανισμοί καταγραφής συναλλαγών στο Ethereum blockchain, υπολογισμού του Eco-Impact Score και απονομής ReWear Tokens μέσω gamification θα ενσωματωθούν με τρόπο **οργανικό και βιώσιμο** στον υπάρχοντα πυρήνα.

Ξεχωριστή αναφορά αξίζει στις **υπηρεσίες Τεχνητής Νοημοσύνης** που αποτελούν ένα από τα πιο φιλόδοξα και διακριτικά καινοτόμα στοιχεία του έργου. Τα τρία αυτόνομα AI microservices - Image Analysis Service, Visual Product Intelligence Service, Recommendation Service - έχουν αρχιτεκτονικά σχεδιαστεί ως ανεξάρτητα συστατικά με τις δικές τους τεχνολογικές στοίβες (Python, FastAPI, container-based deployment), αναπτύσσονται σταδιακά και διασυνδέονται με τον πυρήνα μέσω εξειδικευμένων proxy routes. Με τον τρόπο αυτό, η πλατφόρμα αποκτά τη δυνατότητα να προσφέρει στους χρήστες της εμπειρίες - όπως η αυτόματη συμπλήρωση καταχωρήσεων μέσω visual intelligence ή οι προσωποποιημένες προτάσεις - που υπερβαίνουν σημαντικά τις δυνατότητες μιας συμβατικής πλατφόρμας ηλεκτρονικού εμπορίου.

## 12.2. Σημαντικές Αρχιτεκτονικές Συμβολές

Πέρα από την καταγραφή των επιμέρους επιτευγμάτων, αξίζει να αναδειχθούν ορισμένες **ευρύτερες αρχιτεκτονικές συμβολές** του έργου, οι οποίες αποτελούν διδάγματα τόσο για την ίδια την πλατφόρμα όσο και για ευρύτερα έργα της κατηγορίας. Πρώτη και κρίσιμη αποτελεί η απόδειξη ότι η **headless και composable αρχιτεκτονική** μπορεί να εξυπηρετήσει αποτελεσματικά τις σύνθετες απαιτήσεις ενός marketplace επανεμπορίου, ενσωματώνοντας ταυτόχρονα ευρύ φάσμα εξειδικευμένων υπηρεσιών - από blockchain έως AI - χωρίς να καταρρέει σε μονολιθική πολυπλοκότητα. Η επιτυχία αυτή υποστηρίζει την ευρύτερη τεχνολογική τάση προς αρχιτεκτονικές βασισμένες σε αυτόνομες υπηρεσίες που συντίθενται μέσω τυποποιημένων διεπαφών.

Δεύτερη σημαντική συμβολή είναι η **έμπρακτη απόδειξη της δυνατότητας ενσωμάτωσης blockchain τεχνολογιών** σε ένα παραδοσιακό περιβάλλον e-commerce, με τρόπο που αξιοποιεί τα πραγματικά πλεονεκτήματα της αποκεντρωμένης καταγραφής (διαφάνεια, αμετάβλητη ιχνηλασιμότητα, νέα μοντέλα ανταμοιβής) χωρίς να επιβάλλει στους τελικούς χρήστες την πολυπλοκότητα του υποκείμενου μηχανισμού. Η αρχιτεκτονική του refashion-blockchain-plugin και ο διαχωρισμός μεταξύ on-chain πραγματικότητας και εσωτερικού mirror entity αποτελούν προτυπικά παραδείγματα ενσωμάτωσης blockchain σε mainstream εφαρμογές.

Τρίτη συμβολή αποτελεί η **αρχιτεκτονική ένταξη υπηρεσιών Τεχνητής Νοημοσύνης** ως αυτόνομων microservices διασυνδεδεμένων με τον πυρήνα μέσω καθαρών διεπαφών. Η προσέγγιση αυτή αναγνωρίζει την ιδιαίτερη φύση των AI υπηρεσιών - διαφορετικές υπολογιστικές απαιτήσεις, ταχεία τεχνολογική εξέλιξη, ανάγκη ανεξάρτητης κλιμάκωσης - και την αντιμετωπίζει με αρχιτεκτονικό μοντέλο που εξασφαλίζει τη μακροπρόθεσμη βιωσιμότητα του οικοσυστήματος. Η οπτική αυτή αποτελεί χρήσιμο πρότυπο για ευρύτερες εφαρμογές που επιδιώκουν να ενσωματώσουν αλγόριθμους Μηχανικής Μάθησης στην επιχειρησιακή τους ροή.

Τέταρτη και εξίσου σημαντική αποτελεί η **ανάδειξη της περιβαλλοντικής αξίας ως οπτικού και λειτουργικού στοιχείου** της πλατφόρμας. Με τον Eco-Impact Score να υπολογίζεται αυτόματα για κάθε προϊόν και να εμφανίζεται με συνεκτικό τρόπο σε όλες τις διεπαφές, το REFASHION καταφέρνει να μετατρέψει την έννοια της βιωσιμότητας από αφηρημένη αξία σε **απτή, μετρήσιμη και εύκολα αντιληπτή** πραγματικότητα στο σημείο της αγοραστικής απόφασης. Η συμβολή αυτή εκτείνεται πέρα από τα στενά τεχνολογικά όρια και αφορά τη συνολική αναμόρφωση της σχέσης καταναλωτή-προϊόντος σε ένα κυκλικό μοντέλο οικονομίας.

## 12.3. Επόμενα Βήματα και Μελλοντική Πορεία

Η εξέλιξη του έργου από το παρόν ορόσημο M20 προς την τελική του ολοκλήρωση περιλαμβάνει μια σειρά από **σαφώς οριοθετημένα επόμενα βήματα**. Σε άμεσο επίπεδο, οι εργασίες της ομάδας ανάπτυξης θα επικεντρωθούν στην **ολοκλήρωση της ενσωμάτωσης** των καινοτόμων υποσυστημάτων με τη mobile εφαρμογή, αντιμετωπίζοντας τα εκκρεμή ζητήματα που έχουν εντοπιστεί από την ανάλυση διαθεσιμότητας API. Συγκεκριμένα, η ολοκλήρωση των μηχανισμών παρουσίασης των ReWear Tokens, των Eco-Impact δεικτών και των gamification στοιχείων στο Stats Module της εφαρμογής αποτελεί προτεραιότητα, καθώς θα ολοκληρώσει την εμπειρία χρήστη σε όλα τα κανάλια.

Σε μεσοπρόθεσμο επίπεδο, η πλατφόρμα θα προχωρήσει στη **σταδιακή υλοποίηση των φάσεων** που παρουσιάστηκαν στο κεφάλαιο 11, με έμφαση στην ολοκλήρωση των AI

services και της data infrastructure, που θα αναβαθμίσουν την πλατφόρμα σε ένα πραγματικά κλιμακώσιμο και ευφυές οικοσύστημα. Η ενσωμάτωση του πλήρους Visual Product Intelligence Service με τη ροή καταχώρησης προϊόντων αναμένεται να μειώσει δραστικά το χρόνο και την προσπάθεια που απαιτείται από τους πωλητές, διευκολύνοντας την υιοθέτηση της πλατφόρμας από ευρύτερο κοινό. Παράλληλα, η ενεργοποίηση του Recommendation Service θα μετασχηματίσει την εμπειρία περιήγησης από στατική σε εξατομικευμένη, ενισχύοντας τόσο τη συμμετοχή των χρηστών όσο και τους ρυθμούς μετατροπής σε αγορά.

Πριν τη μετάβαση σε ευρεία παραγωγική λειτουργία, η πλατφόρμα θα υποβληθεί σε **εκτεταμένη φάση δοκιμών** που θα περιλαμβάνει functional testing όλων των ροών, performance testing υπό συνθήκες υψηλού φόρτου, και security testing για τον εντοπισμό και την αντιμετώπιση τυχόν ευπαθειών. Από τεχνολογική σκοπιά, η μελλοντική εξέλιξη του REFASHION περιλαμβάνει την ωρίμανση της υποδομής φιλοξενίας προς μια **πραγματικά κλιμακώσιμη παραγωγική αρχιτεκτονική**, σύμφωνα με τις σχεδιαστικές στοχεύσεις που παρουσιάστηκαν στο κεφάλαιο 10. Η μετάβαση από την τρέχουσα self-hosted Docker Compose εγκατάσταση σε ένα ώριμο container orchestration περιβάλλον, η ενσωμάτωση GPU-enabled υποδομής για τις AI υπηρεσίες, και η ενίσχυση του API Gateway layer αποτελούν τα κύρια βήματα προς την κατεύθυνση αυτή.

Σε ευρύτερο πλαίσιο, η πλατφόρμα REFASHION φιλοδοξεί να αποτελέσει όχι απλώς ένα τεχνολογικό προϊόν, αλλά **έναν θεμελιώδη μηχανισμό μετασχηματισμού** της σχέσης των καταναλωτών με τη μόδα. Η συνεκτικότητα της αρχιτεκτονικής, η ποιότητα της υλοποίησης και η αρτιότητα των καινοτόμων λειτουργιών που έχουν αναπτυχθεί δημιουργούν στέρεα θεμέλια πάνω στα οποία μπορεί να οικοδομηθεί μια πραγματικά διαφορετική εμπειρία ηλεκτρονικού εμπορίου - μια εμπειρία όπου η περιβαλλοντική συνείδηση, η οικονομική βιωσιμότητα και η τεχνολογική αρτιότητα συνυπάρχουν αρμονικά. Η ολοκλήρωση του έργου θα συμβάλει ουσιαστικά στην προώθηση των αρχών της κυκλικής οικονομίας στον χώρο της μόδας, ενώ παράλληλα θα παράξει σημαντική **ερευνητική και τεχνολογική γνώση** που θα μπορεί να αξιοποιηθεί σε αντίστοιχες πρωτοβουλίες του μέλλοντος.

**Η ολοκλήρωση του παρόντος παραδοτέου δεν αποτελεί επομένως απλά ένα διοικητικό ορόσημο, αλλά έμπρακτη απόδειξη της προόδου ενός φιλόδοξου και πολυδιάστατου έργου, καθώς και σταθερή βάση για τις επόμενες φάσεις που θα ακολουθήσουν.**