



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Μελέτη και υλοποίηση Data Space για
ενοποιημένη πρόσβαση σε ετερογενή
δεδομένα»

Του φοιτητή
Ιωάννη Δουλκερίδη
Αρ. Μητρώου: 2019040

Επιβλέπων
Αντώνιος Σιδηρόπουλος
Καθηγητής

Ημερομηνία 31/8/2026

Μελέτη και υλοποίηση Data Space για ενοποιημένη πρόσβαση σε ετερογενή
δεδομένα

Κωδικός Δ.Ε.: 26210

Ονοματεπώνυμο φοιτητή: Δουλκερίδης Ιωάννης

Ονοματεπώνυμο εισηγητή: Σιδηρόπουλος Αντώνιος

Ημερομηνία ανάληψης Δ.Ε.: 24/3/2026

Ημερομηνία περάτωσης Δ.Ε.: 31/8/2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Δουλκερίδη Ιωάννης που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε με αντικείμενο τη μελέτη, τον σχεδιασμό και την υλοποίηση ενός περιβάλλοντος Data Space για την ενοποιημένη πρόσβαση σε ετερογενή δεδομένα, με έμφαση στο πανεπιστημιακό πληροφοριακό οικοσύστημα. Η επιλογή του συγκεκριμένου θέματος συνδέεται με τη συνεχώς αυξανόμενη ανάγκη των σύγχρονων οργανισμών να αξιοποιούν δεδομένα τα οποία βρίσκονται καταναμημένα σε διαφορετικά πληροφοριακά συστήματα, βάσεις δεδομένων και τεχνολογικές πλατφόρμες.

Στα πανεπιστημιακά ιδρύματα, η πληροφορία παράγεται και αποθηκεύεται σε πολλαπλές ανεξάρτητες πηγές, όπως συστήματα διαχείρισης φοιτητών, βάσεις μαθημάτων, πλατφόρμες ηλεκτρονικής μάθησης, συστήματα αξιολογήσεων και εφαρμογές διοικητικής υποστήριξης. Η ύπαρξη αυτών των ετερογενών συστημάτων δημιουργεί σημαντικές δυσκολίες ως προς τη διαλειτουργικότητα, την ενιαία πρόσβαση και την παραγωγή ολοκληρωμένης γνώσης. Για τον λόγο αυτό, η διερεύνηση σύγχρονων αρχιτεκτονικών, όπως τα Data Spaces και οι federated query engines, αποτελεί ιδιαίτερα σημαντικό πεδίο μελέτης. Η εργασία εστιάζει στην ομοσπονδιακή προσέγγιση πρόσβασης δεδομένων, σύμφωνα με την οποία τα δεδομένα παραμένουν στις αρχικές τους πηγές και προσπελάζονται μέσω ενός ενιαίου λογικού επιπέδου. Με τον τρόπο αυτό αποφεύγεται η ανάγκη φυσικής συγκέντρωσης των δεδομένων σε κεντρικό αποθετήριο, μειώνεται η αντιγραφή πληροφορίας και διατηρείται η αυτονομία των επιμέρους συστημάτων. Ιδιαίτερη έμφαση δίνεται στο Trino, το οποίο αξιοποιείται ως federated query engine για την εκτέλεση ενοποιημένων SQL ερωτημάτων σε πολλαπλές πηγές δεδομένων.

Περίληψη

Η παρούσα διπλωματική εργασία μελετά, σχεδιάζει και υλοποιεί ένα περιβάλλον Data Space για την ενοποιημένη πρόσβαση σε ετερογενή δεδομένα, με πεδίο εφαρμογής ένα πανεπιστημιακό πληροφοριακό οικοσύστημα. Το βασικό πρόβλημα που εξετάζεται είναι ο πληροφοριακός κατακερματισμός, καθώς τα σύγχρονα πανεπιστήμια χρησιμοποιούν πολλά ανεξάρτητα συστήματα για φοιτητές, μαθήματα, διδάσκοντες, εξεταστικές διαδικασίες, αξιολογήσεις και ερευνητικές δραστηριότητες. Τα συστήματα αυτά συχνά βασίζονται σε διαφορετικές βάσεις δεδομένων, διαφορετικά σχήματα, διαφορετικές τεχνολογίες και διαφορετικούς μηχανισμούς πρόσβασης, γεγονός που δυσκολεύει την παραγωγή ενιαίας γνώσης.

Η εργασία παρουσιάζει θεωρητικά την έννοια των Data Spaces και εξηγεί τη διαφοροποίησή τους από παραδοσιακές αρχιτεκτονικές, όπως τα data warehouses, τα data lakes και οι ETL διαδικασίες. Ιδιαίτερη έμφαση δίνεται στην ομοσπονδιακή πρόσβαση δεδομένων, η οποία επιτρέπει την εκτέλεση ενοποιημένων ερωτημάτων χωρίς φυσική μεταφορά των δεδομένων σε κεντρικό αποθετήριο. Με τον τρόπο αυτό διατηρείται η αυτονομία των επιμέρους πηγών, μειώνεται η αντιγραφή δεδομένων και υποστηρίζεται πρόσβαση σε σχεδόν πραγματικό χρόνο.

Στο πρακτικό μέρος, η εργασία εστιάζει στη χρήση του Trino ως federated query engine. Το Trino επιλέγεται λόγω της επεκτασιμότητας, του χαρακτήρα ανοικτού κώδικα, της υποστήριξης πολλών connectors και της δυνατότητας εκτέλεσης σύνθετων SQL ερωτημάτων σε διαφορετικές πηγές δεδομένων. Η προτεινόμενη αρχιτεκτονική βασίζεται σε επίπεδα πηγών δεδομένων, federation layer, semantic layer, governance layer και analytics layer. Μέσα από την υλοποίηση και την πειραματική αξιολόγηση, αναδεικνύεται ότι η προσέγγιση Data Space μπορεί να προσφέρει ευελιξία, διαλειτουργικότητα και καλύτερη αξιοποίηση κατακερματισμένων πανεπιστημιακών δεδομένων, αν και παραμένουν προκλήσεις σε θέματα ασφάλειας, σημασιολογικής ετερογένειας, βελτιστοποίησης ερωτημάτων και απόδοσης.

Λέξεις-κλειδιά: Data Space, ετερογενή δεδομένα, ομοσπονδιακά ερωτήματα, Trino, πανεπιστημιακά συστήματα

Study and Implementation of a Data Space for Unified Access to Heterogeneous Data

Ioannis Doulkeridis

Abstract

This thesis studies, designs and implements a Data Space environment for unified access to heterogeneous data, focusing on a university information ecosystem. The main problem examined is information fragmentation, since modern universities rely on multiple independent systems for students, courses, instructors, examination processes, evaluations and research activities. These systems often use different databases, schemas, technologies and access mechanisms, making it difficult to produce integrated knowledge and support evidence-based decision-making.

The thesis presents the theoretical background of Data Spaces and explains how they differ from traditional architectures such as data warehouses, data lakes and ETL-based integration models. Emphasis is placed on federated data access, which enables unified queries across multiple distributed sources without physically transferring the data to a central repository. This approach preserves the autonomy of individual data sources, reduces data duplication and supports near real-time access to information.

In the practical part, the thesis focuses on the use of Trino as a federated query engine. Trino is selected because of its scalability, open-source nature, broad connector ecosystem and ability to execute complex SQL queries across different data sources. The proposed architecture is structured around several layers, including the data source layer, federation layer, semantic layer, governance layer and analytics layer. Through implementation and experimental evaluation, the study demonstrates that a Data Space approach can provide flexibility, interoperability and more effective use of distributed university data. At the same time, the thesis identifies important challenges. These include semantic heterogeneity between different schemas, security and access control issues, query optimization complexity and performance limitations caused by distributed execution and network latency. Overall, the work concludes that Data Spaces, supported by technologies such as Trino, form a promising architectural solution for universities that need unified access to heterogeneous data without abandoning their existing information systems.

Keywords: Data Space, heterogeneous data, federated queries, Trino, university information systems

Περιεχόμενα

Πρόλογος	iii
Περίληψη	iv
Abstract	v
Πίνακας πινάκων	ix
Πίνακας σχημάτων	x
Συντομογραφίες	xi
Κεφάλαιο 1: Εισαγωγή	1
1.1 Αντικείμενο της διπλωματικής εργασίας	1
1.2 Περιγραφή του προβλήματος.....	1
1.3 Ανάγκη για ενοποιημένη πρόσβαση σε ετερογενή δεδομένα	2
1.4 Στόχοι της εργασίας	3
1.5 Ερευνητικά ερωτήματα.....	3
1.6 Μεθοδολογία υλοποίησης και έρευνας.....	4
1.7 Πεδίο εφαρμογής στο πανεπιστημιακό περιβάλλον	4
1.8 Συνεισφορά της εργασίας	5
1.9 Δομή της εργασίας.....	5
Κεφάλαιο 2: Θεωρητικό υπόβαθρο και βιβλιογραφική επισκόπηση.....	7
2.1 Η έννοια των δεδομένων στη σύγχρονη πληροφοριακή εποχή	7
2.2 Ετερογενή πληροφοριακά συστήματα	8
2.2.1 Ορισμός και χαρακτηριστικά.....	9
2.2.2 Προκλήσεις ενοποίησης	9
2.2.3 Ζητήματα συμβατότητας και διαλειτουργικότητας	9
2.3 Η έννοια των Data Spaces.....	10
2.4 Ομοσπονδιακή πρόσβαση σε δεδομένα	13
2.4.1 Έννοια της ομοσπονδιακής πρόσβασης.....	13
2.4.2 Πλεονεκτήματα και περιορισμοί	14
2.5 Federated Query Engines	15
2.6 Επισκόπηση σχετικών τεχνολογιών	19
2.7 Ανασκόπηση σχετικών ερευνητικών εργασιών	22
Κεφάλαιο 3: Τεχνολογίες, εργαλεία και αρχιτεκτονικές υλοποίησης.....	23
3.1 Απαιτήσεις ενός περιβάλλοντος Data Space.....	23
3.1.1 Λειτουργικές απαιτήσεις.....	23
3.1.2 Μη λειτουργικές απαιτήσεις.....	24

3.1.3 Απαιτήσεις ασφάλειας και πρόσβασης.....	24
3.2 Αρχιτεκτονικές προσεγγίσεις ενοποίησης δεδομένων	26
3.2.1 Κεντρικοποιημένη αρχιτεκτονική.....	26
3.2.2 Κατανεμημένη αρχιτεκτονική.....	26
3.2.3 Ομοσπονδιακή αρχιτεκτονική	27
3.2.4 Υβριδικές προσεγγίσεις	28
3.3 Παρουσίαση του Trino.....	29
3.3.1 Γενικά χαρακτηριστικά.....	29
3.3.2 Αρχιτεκτονική Trino	30
3.3.3 Coordinator και Workers	31
3.3.4 Connectors και Catalogs.....	32
3.3.5 Query execution model.....	33
3.3.6 Πλεονεκτήματα χρήσης Trino	34
3.4 Πηγές δεδομένων στο πανεπιστημιακό περιβάλλον	35
3.4.1 Βάσεις δεδομένων φοιτητών	35
3.4.2 Βάσεις δεδομένων μαθημάτων	36
3.4.3 Δεδομένα καθηγητών και διδασκόντων.....	37
3.4.4 Πρόγραμμα εξεταστικής.....	38
3.4.5 Στατιστικά αξιολογήσεων.....	39
3.4.6 Πιθανές εξωτερικές ή βοηθητικές πηγές	39
3.5 Τεχνολογίες υποστήριξης της υλοποίησης.....	40
3.5.1 Συστήματα βάσεων δεδομένων	40
3.5.2 SQL και federated queries	41
3.5.3 Εργαλεία διαχείρισης και παρακολούθησης.....	42
3.5.4 Docker ή εικονικοποίηση.....	43
3.5.5 Εργαλεία δοκιμών και benchmarking.....	44
3.6 Κριτήρια επιλογής τεχνολογιών.....	45
3.7 Συγκριτική αξιολόγηση και τελική επιλογή εργαλείων	46
Κεφάλαιο 4: Σχεδίαση του προτεινόμενου συστήματος Data Space	48
4.1 Περιγραφή και στόχοι του προτεινόμενου συστήματος	48
4.2 Ανάλυση απαιτήσεων συστήματος.....	48
4.2.1 Απαιτήσεις χρηστών και διαχειριστή.....	49
4.2.2 Απαιτήσεις δεδομένων και απόδοσης.....	49
4.3 Σχεδιασμός αρχιτεκτονικής.....	50
4.4 Μοντελοποίηση και ενοποίηση των πηγών δεδομένων	51

4.4.1 Σχήματα, οντότητες και σχέσεις δεδομένων.....	51
4.4.2 Αντιστοίχιση δεδομένων και σημασιολογική ετερογένεια	53
4.5 Σχεδιασμός ερωτημάτων και σεναρίων χρήσης.....	54
4.6 Ασφάλεια, δικαιώματα πρόσβασης και προστασία δεδομένων	54
4.7 Περιορισμοί και πιθανές δυσκολίες σχεδίασης.....	55
Κεφάλαιο 5: Υλοποίηση και πειραματική εφαρμογή.....	57
5.1 Περιβάλλον ανάπτυξης και εγκατάστασης	57
5.2 Εγκατάσταση και παραμετροποίηση του Trino	58
5.2.1 Διαμόρφωση coordinator, worker και βασικών παραμέτρων	58
5.2.2 Catalogs, connectors και σύνδεση με ετερογενείς πηγές.....	58
5.3 Διασύνδεση και επαλήθευση των πηγών δεδομένων	59
5.4 Υλοποίηση ενοποιημένων και federated ερωτημάτων.....	60
5.4.1 Ερωτήματα ανά πηγή και βασικά παραδείγματα SQL	60
5.4.2 Cross-source JOINS, aggregations και αναλυτικά ερωτήματα	61
5.5 Σενάρια χρήσης και ανάλυση αποτελεσμάτων	62
5.6 Πειραματική αξιολόγηση.....	62
5.6.1 Απόδοση ερωτημάτων και χρόνος απόκρισης.....	62
5.6.2 Επεκτασιμότητα, ευχρηστία και λειτουργικότητα.....	63
5.7 Προβλήματα υλοποίησης και τρόποι αντιμετώπισης.....	64
Κεφάλαιο 6: Συμπεράσματα και μελλοντικές προεκτάσεις.....	66
6.1 Ανακεφαλαίωση της διαδικασίας και βαθμός επίτευξης των στόχων.....	66
6.2 Κύρια ευρήματα και συμπεράσματα.....	66
6.3 Πλεονεκτήματα και περιορισμοί της προσέγγισης Data Space	67
6.4 Δυνατότητα εφαρμογής σε πραγματικό πανεπιστημιακό περιβάλλον	67
6.5 Προτάσεις βελτίωσης και μελλοντική επέκταση του συστήματος	68
Βιβλιογραφία.....	69

Πίνακας πινάκων

Πίνακας 2.1: Βασικά επίπεδα λειτουργίας ενός Data Space	11
Πίνακας 2.2: Βασικές αρχές σχεδίασης των Data Spaces	11
Πίνακας 2.3: Σύγκριση Data Space και Data Warehouse	12
Πίνακας 2.4: Σύγκριση Data Space και Data Lake	12
Πίνακας 2.5: Σύγκριση κλασικής ολοκλήρωσης δεδομένων και Data Space	12
Πίνακας 2.6: Βασικά επίπεδα λειτουργίας της federated πρόσβασης	13
Πίνακας 2.7: Πλεονεκτήματα της ομοσπονδιακή πρόσβασης σε δεδομένα	14
Πίνακας 2.8: Περιορισμοί της ομοσπονδιακή πρόσβασης σε δεδομένα	14
Πίνακας 2.9: Πηγές δεδομένων σε πανεπιστημιακό περιβάλλον ομοσπονδιακή πρόσβασης	15
Πίνακας 2.10: Στάδια λειτουργίας ενός Federated Query Engine	16
Πίνακας 2.11: Τύποι πηγών δεδομένων που υποστηρίζονται από Federated Query Engines	17
Πίνακας 2.12: Βασικά συστατικά της αρχιτεκτονικής ενός Federated Query Engine	17
Πίνακας 2.13: Τεχνικές βελτιστοποίησης σε Federated Query Engines	18
Πίνακας 2.14: Πηγές δεδομένων και αντίστοιχες τεχνολογίες σε Federated Query Engines	18
Πίνακας 3.1: Μηχανισμοί ασφάλειας και ελέγχου πρόσβασης σε περιβάλλον Data Space	25
Πίνακας 3.2: Σύγκριση χαρακτηριστικών υβριδικών αρχιτεκτονικών ενοποίησης δεδομένων	29
Πίνακας 3.3: Βασικά αρχιτεκτονικά χαρακτηριστικά του Trino	30
Πίνακας 3.4: Κύρια χαρακτηριστικά του μοντέλου εκτέλεσης ερωτημάτων στο Trino	34
Πίνακας 3.5: Συγκριτική αποτύπωση χαρακτηριστικών βάσεων δεδομένων φοιτητών και μαθημάτων	37
Πίνακας 3.6: Κατηγοριοποίηση και χαρακτηριστικά συστημάτων βάσεων δεδομένων σε σύγχρονες αρχιτεκτονικές	41
Πίνακας 3.7: Βασικά εργαλεία διαχείρισης και παρακολούθησης δεδομένων σε σύγχρονες αρχιτεκτονικές	43
Πίνακας 3.8: Κριτήρια επιλογής τεχνολογιών σε περιβάλλοντα διαχείρισης δεδομένων	46
Πίνακας 3.9: Συγκριτική αξιολόγηση τεχνολογιών και εργαλείων διαχείρισης δεδομένων	47
Πίνακας 4.1: Κύριες οντότητες και σχέσεις του μοντέλου δεδομένων	52
Πίνακας 4.2: Κύριες σημασιολογικές αντιστοιχίσεις μεταξύ πηγών	53
Πίνακας 4.3: Σχεδιαζόμενες κατηγορίες ερωτημάτων και πηγές δεδομένων	54
Πίνακας 5.1: Βασικά στοιχεία του πειραματικού περιβάλλοντος	57
Πίνακας 5.2: Catalogs και μηχανισμοί σύνδεσης	59
Πίνακας 5.3: Συγκεντρωτικό αποτέλεσμα αναλυτικού federated query	61
Πίνακας 5.4: Συνοπτική αποτύπωση σεναρίων χρήσης	62
Πίνακας 5.5: Καταγεγραμμένες μετρικές σύνθετου federated query	63
Πίνακας 5.6: Συνοπτική λειτουργική αξιολόγηση	64
Πίνακας 5.7: Κύρια προβλήματα και αντιμετώπιση	65

Πίνακας σχημάτων

Σχήμα 4.1: Αρχιτεκτονική της πιλοτικής υλοποίησης του ακαδημαϊκού Data Space και ενοποιημένη πρόσβαση στις ετερογενείς πηγές μέσω Trino.	51
Σχήμα 4.2: Σχήματα και βασικές σχέσεις μεταξύ των οντοτήτων των πηγών δεδομένων της πιλοτικής υλοποίησης.	52

Συντομογραφίες

Συντομογραφία	Πλήρης όρος	Ερμηνεία / Απόδοση στα ελληνικά
ABAC	Attribute-Based Access Control	Έλεγχος πρόσβασης βάσει χαρακτηριστικών
AI	Artificial Intelligence	Τεχνητή Νοημοσύνη
API	Application Programming Interface	Διεπαφή προγραμματισμού εφαρμογών
AST	Abstract Syntax Tree	Αφηρημένο συντακτικό δέντρο
AWS	Amazon Web Services	Υπηρεσίες υπολογιστικού νέφους της Amazon
BI	Business Intelligence	Επιχειρηματική ευφυΐα
CLI	Command Line Interface	Διεπαφή γραμμής εντολών
CPU	Central Processing Unit	Κεντρική μονάδα επεξεργασίας
DBMS	Database Management System	Σύστημα διαχείρισης βάσεων δεδομένων
DS	Data Space	Χώρος δεδομένων
ETL	Extract, Transform, Load	Εξαγωγή, μετασχηματισμός και φόρτωση δεδομένων
FQE	Federated Query Engine	Μηχανή ομοσπονδιακών ερωτημάτων
GDPR	General Data Protection Regulation	Γενικός Κανονισμός Προστασίας Δεδομένων
HDFS	Hadoop Distributed File System	Κατανεμημένο σύστημα αρχείων Hadoop
HR	Human Resources	Ανθρώπινο δυναμικό
I/O	Input / Output	Είσοδος / Έξοδος δεδομένων
IDC	International Data Corporation	Διεθνής εταιρεία αναλύσεων και ερευνών αγοράς
IoT	Internet of Things	Διαδίκτυο των Πραγμάτων
JSON	JavaScript Object Notation	Μορφή αναπαράστασης και ανταλλαγής δεδομένων
LMS	Learning Management System	Σύστημα διαχείρισης μάθησης
MPP	Massively Parallel Processing	Μαζικά παράλληλη επεξεργασία
NoSQL	Not Only SQL	Μη σχεσιακές ή εναλλακτικές βάσεις δεδομένων
OLAP	Online Analytical Processing	Διαδικτυακή αναλυτική επεξεργασία
OLTP	Online Transaction Processing	Διαδικτυακή επεξεργασία συναλλαγών
SQL	Structured Query Language	Δομημένη γλώσσα ερωτημάτων
SSO	Single Sign-On	Ενιαία σύνδεση χρήστη
TLS	Transport Layer Security	Πρωτόκολλο ασφαλούς μεταφοράς δεδομένων

Κεφάλαιο 1: Εισαγωγή

1.1 Αντικείμενο της διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία επικεντρώνεται στη μελέτη, σχεδίαση και υλοποίηση ενός περιβάλλοντος Data Space για ενοποιημένη πρόσβαση σε ετερογενή δεδομένα εντός πανεπιστημιακού πληροφοριακού οικοσυστήματος, αξιοποιώντας τεχνολογίες federated query processing, κατανεμημένης ανάλυσης δεδομένων και data virtualization. Η διαρκής αύξηση του όγκου ψηφιακής πληροφορίας, σε συνδυασμό με την πολυμορφία των πληροφοριακών υποδομών, έχει δημιουργήσει ένα περιβάλλον υψηλής πολυπλοκότητας, όπου τα δεδομένα αποθηκεύονται σε ανεξάρτητα συστήματα με διαφορετικά σχήματα, διαφορετικές λογικές πρόσβασης και ασύμβατες αρχιτεκτονικές διαχείρισης[1], [2]. Στο πλαίσιο αυτό, τα πανεπιστήμια αποτελούν χαρακτηριστικά παραδείγματα οργανισμών με έντονα κατακερματισμένη πληροφοριακή δομή, καθώς διαθέτουν ξεχωριστά πληροφοριακά συστήματα για φοιτητές, διδάσκοντες, αξιολογήσεις, εξεταστικές διαδικασίες, ηλεκτρονικά μαθήματα και ερευνητικές δραστηριότητες, χωρίς ενιαίο επίπεδο πρόσβασης και διαλειτουργικότητας.

Η έννοια των Data Spaces αναδύθηκε ως απάντηση στους περιορισμούς των παραδοσιακών αρχιτεκτονικών data warehousing, οι οποίες βασίζονται σε διαδικασίες ETL και φυσική συγκέντρωση δεδομένων σε κεντρικά αποθετήρια[3]. Σύμφωνα με τους Curry et al.[4], τα data spaces επιτρέπουν αποκεντρωμένη διαχείριση πληροφοριών, διατηρώντας την αυτονομία των πηγών δεδομένων και παρέχοντας παράλληλα ενιαίο λογικό επίπεδο πρόσβασης. Η συγκεκριμένη φιλοσοφία μειώνει σημαντικά τη διπλή αποθήκευση δεδομένων, τη λειτουργική επιβάρυνση και την ανάγκη συνεχούς συγχρονισμού δεδομένων, ενώ παράλληλα αυξάνει τη δυνατότητα αξιοποίησης πληροφοριών σε πραγματικό χρόνο. Επιπροσθέτως, οι Jarke και Quix[1] επισημαίνουν ότι τα federated μοντέλα ενοποίησης αποτελούν βασικό πυλώνα των σύγχρονων data spaces, καθώς επιτρέπουν δυναμική σύνθεση ερωτημάτων σε κατανεμημένες πηγές χωρίς ανάγκη φυσικής αναδόμησης των αρχικών συνόλων δεδομένων.

Οι μηχανισμοί αυτοί επιτρέπουν την εκτέλεση σύνθετων αναλυτικών ερωτημάτων σε πολλαπλά ετερογενή συστήματα μέσω parallel query execution και τεχνικών προσαρμοστικής βελτιστοποίησης[2]. Σύμφωνα με τους Gu et al.[2], τα σύγχρονα federated systems εμφανίζουν έως και 52% χαμηλότερη καθυστέρηση σε περιβάλλοντα κατανεμημένης ανάλυσης δεδομένων συγκριτικά με παραδοσιακές ETL αρχιτεκτονικές, ιδιαίτερα όταν οι πηγές δεδομένων χαρακτηρίζονται από υψηλή μεταβλητότητα και διαφορετικές δομές αποθήκευσης. Η αξιοποίηση τέτοιων τεχνολογιών αποκτά ιδιαίτερη σημασία σε πανεπιστημιακά οικοσυστήματα, όπου η ανάγκη διασύνδεσης πολλαπλών ανεξάρτητων πληροφοριακών υπηρεσιών αποτελεί κρίσιμο λειτουργικό ζήτημα.

1.2 Περιγραφή του προβλήματος

Η πολυπλοκότητα των σύγχρονων πληροφοριακών οικοσυστημάτων έχει δημιουργήσει σοβαρές δυσκολίες ως προς την αποτελεσματική διαχείριση, ενοποίηση και αξιοποίηση των διαθέσιμων δεδομένων. Σε μεγάλα οργανωσιακά περιβάλλοντα, όπως τα πανεπιστημιακά ιδρύματα, η πληροφορία διασκορπίζεται σε δεκάδες ανεξάρτητα συστήματα, τα οποία λειτουργούν απομονωμένα και ακολουθούν διαφορετικές τεχνολογικές προδιαγραφές. Ως αποτέλεσμα, παρατηρείται έντονος

πληροφοριακός κατακερματισμός, γεγονός που δυσχεραίνει σημαντικά τη δυνατότητα παραγωγής ενοποιημένης γνώσης και περιορίζει τη λήψη τεκμηριωμένων αποφάσεων[2], [5]. Η συγκεκριμένη κατάσταση επιδεινώνεται από τη συνεχή αύξηση του όγκου δεδομένων, καθώς σύγχρονες εφαρμογές πανεπιστημιακής λειτουργίας παράγουν τεράστιες ποσότητες συναλλακτικών, αναλυτικών και ημιδομημένων πληροφοριών σε πραγματικό χρόνο.

Τα πληροφοριακά συστήματα ενός πανεπιστημίου ενδέχεται να χρησιμοποιούν διαφορετικά DBMS, διαφορετικές μορφές δεδομένων και διαφορετικές λογικές μοντελοποίησης. Για παράδειγμα, η βάση δεδομένων φοιτητών μπορεί να λειτουργεί σε PostgreSQL, οι πλατφόρμες e-learning σε MySQL, ενώ τα συστήματα αναλυτικής επεξεργασίας να αξιοποιούν cloud object storage ή NoSQL υποδομές[6]. Η συνύπαρξη τέτοιων τεχνολογιών δημιουργεί σοβαρές δυσκολίες διαλειτουργικότητας, καθώς κάθε σύστημα χρησιμοποιεί διαφορετικούς μηχανισμούς αυθεντικοποίησης, διαφορετικές διαλέκτους SQL και διαφορετικές δομές μεταδεδομένων. Οι Putrama και Martinek[6] επισημαίνουν ότι ακόμη και όταν διαφορετικά συστήματα αποθηκεύουν παρόμοιες πληροφορίες, οι αποκλίσεις σε συμβάσεις ονοματοδοσίας, περιορισμούς και ορισμούς οντοτήτων καθιστούν προβληματική τη σύνθεση ενοποιημένων αναλυτικών ερωτημάτων. Για παράδειγμα, ένας φοιτητής μπορεί να εμφανίζεται με διαφορετικό αναγνωριστικό σε διαφορετικά πληροφοριακά συστήματα, γεγονός που καθιστά εξαιρετικά δύσκολη τη συσχέτιση δεδομένων χωρίς πρόσθετους μηχανισμούς σημασιολογικής αντιστοίχισης.

Επιπροσθέτως, τα παραδοσιακά μοντέλα data integration εμφανίζουν σημαντικούς λειτουργικούς περιορισμούς. Οι διαδικασίες ETL απαιτούν φυσική αντιγραφή δεδομένων, υψηλή επιβάρυνση αποθήκευσης και συνεχή συγχρονισμό μεταξύ των πηγών και των κεντρικών αποθετηρίων[1]. Σύμφωνα με τους Vieira et al.[7], οι διαδικασίες αυτές αυξάνουν σημαντικά το λειτουργικό κόστος και δημιουργούν σοβαρά προβλήματα consistency σε περιβάλλοντα μεταβολών δεδομένων σε πραγματικό χρόνο. Ειδικά σε πανεπιστημιακές υποδομές, όπου τα δεδομένα ενημερώνονται συνεχώς από διαφορετικές υπηρεσίες, η καθυστέρηση συγχρονισμού μπορεί να οδηγήσει σε ανακριβείς αναφορές και λανθασμένες διοικητικές αποφάσεις.

1.3 Ανάγκη για ενοποιημένη πρόσβαση σε ετερογενή δεδομένα

Η ανάγκη ενοποιημένης πρόσβασης σε ετερογενή δεδομένα αποτελεί πλέον θεμελιώδη προϋπόθεση για την αποτελεσματική λειτουργία σύνθετων οργανωσιακών οικοσυστημάτων. Η εκθετική αύξηση της παραγωγής ψηφιακής πληροφορίας, σε συνδυασμό με την αποκέντρωση των πληροφοριακών υποδομών, έχει δημιουργήσει ένα περιβάλλον στο οποίο η πληροφορία διασκορπίζεται σε διαφορετικούς τεχνολογικούς τομείς, χωρίς ενιαίο επίπεδο διαλειτουργικότητας[1], [8]. Στα πανεπιστημιακά περιβάλλοντα, η κατάσταση αυτή εμφανίζεται ιδιαίτερα έντονα, καθώς οι διοικητικές, ερευνητικές και ακαδημαϊκές λειτουργίες παράγουν συνεχώς νέες ροές δεδομένων μέσω ανεξάρτητων εφαρμογών και υπηρεσιών.

Η ενοποιημένη πρόσβαση επιτρέπει τη δημιουργία ενός ενιαίου λογικού επιπέδου διαχείρισης πληροφορίας χωρίς να απαιτείται φυσική συγκέντρωση των δεδομένων σε κεντρικά αποθετήρια. Σύμφωνα με τους Jarke και Quix[1], τα federated data spaces επιτρέπουν δυναμική σύνθεση πληροφοριών από πολλαπλές κατακερματισμένες πηγές, διατηρώντας παράλληλα την αυτονομία των επιμέρους συστημάτων. Η προσέγγιση αυτή μειώνει σημαντικά τις απαιτήσεις αποθήκευσης και περιορίζει τα προβλήματα consistency που εμφανίζονται στα παραδοσιακά ETL environments. Οι

σύγχρονες διοικητικές αποφάσεις βασίζονται ολοένα και περισσότερο στην άμεση πρόσβαση σε συνδυαστικά δεδομένα, όπως στατιστικά επίδοσης φοιτητών, αξιολόγηση μαθημάτων και ανάλυση εκπαιδευτικών δεικτών[9]. Οι καθυστερήσεις που προκαλούνται από batch synchronization μοντέλα περιορίζουν την αξιοπιστία των αποτελεσμάτων και αυξάνουν τη λειτουργική καθυστέρηση. Αντιθέτως, τα federated query engines επιτρέπουν την εκτέλεση σύνθετων αναλυτικών ερωτημάτων σχεδόν σε πραγματικό χρόνο σε πολλαπλές πηγές ταυτόχρονα[2].

1.4 Στόχοι της εργασίας

Η παρούσα διπλωματική εργασία αποσκοπεί στη μελέτη και υλοποίηση ενός περιβάλλοντος Data Space για ενοποιημένη πρόσβαση σε ετερογενή δεδομένα σε πανεπιστημιακό πληροφοριακό περιβάλλον. Η εργασία επικεντρώνεται τόσο στη θεωρητική διερεύνηση των σύγχρονων τεχνολογιών data federation όσο και στην πρακτική αξιολόγηση federated query architectures μέσω πειραματικής εφαρμογής. Οι βασικοί στόχοι της εργασίας είναι οι ακόλουθοι:

- Η ανάλυση της έννοιας των Data Spaces και η διερεύνηση του τρόπου με τον οποίο διαφοροποιούνται από παραδοσιακά data warehouses, data lakes και κλασικά συστήματα ολοκλήρωσης δεδομένων.
- Η μελέτη τεχνολογιών federated data access και distributed query processing με στόχο την κατανόηση των μηχανισμών ενοποιημένης πρόσβασης σε κατανεμημένες πηγές δεδομένων.
- Η διερεύνηση της λειτουργίας federated query engines, όπως το Trino, και η αξιολόγηση της δυνατότητάς τους να εκτελούν σύνθετα ερωτήματα SQL σε πολλαπλά ετερογενή πληροφοριακά συστήματα.
- Η ανάλυση προβλημάτων σημασιολογικής ετερογένειας μεταξύ διαφορετικών βάσεων δεδομένων και η μελέτη τεχνικών semantic interoperability και data virtualization.
- Η σχεδίαση αρχιτεκτονικής Data Space κατάλληλης για πανεπιστημιακό περιβάλλον με στόχο τη διασύνδεση διαφορετικών πληροφοριακών συστημάτων χωρίς φυσική συγκέντρωση των δεδομένων.
- Η δημιουργία πειραματικού περιβάλλοντος federated πρόσβασης που θα επιτρέπει την ενοποιημένη εκτέλεση ερωτημάτων σε δεδομένα φοιτητών, μαθημάτων, εξεταστικών διαδικασιών και αξιολογήσεων.
- Η αξιολόγηση της απόδοσης του συστήματος ως προς τον χρόνο απόκρισης, την επεκτασιμότητα, τη δυνατότητα distributed execution και τη διαχείριση φόρτου εργασίας.

1.5 Ερευνητικά ερωτήματα

Η εργασία επιχειρεί να απαντήσει σε ένα σύνολο ερευνητικών ερωτημάτων που σχετίζονται με την αποτελεσματικότητα, την επεκτασιμότητα και τη λειτουργική βιωσιμότητα των Data Spaces ως σύγχρονων αρχιτεκτονικών ενοποιημένης πρόσβασης σε ετερογενή δεδομένα.

Τα βασικά ερευνητικά ερωτήματα της εργασίας είναι τα εξής:

- Κατά πόσο μία federated αρχιτεκτονική Data Space μπορεί να παρέχει αποδοτική και ενοποιημένη πρόσβαση σε πολλαπλά ετερογενή πληροφοριακά συστήματα χωρίς φυσική μεταφορά των δεδομένων;

- Ποια είναι τα βασικά πλεονεκτήματα και οι περιορισμοί των federated query engines σε σύγκριση με παραδοσιακές ETL και data warehouse αρχιτεκτονικές;
- Σε ποιο βαθμό οι τεχνολογίες data virtualization μπορούν να μειώσουν την πολυπλοκότητα ολοκλήρωσης δεδομένων σε περιβάλλοντα υψηλής πληροφοριακής ετερογένειας;
- Πώς επηρεάζει η σημασιολογική ασυμβατότητα μεταξύ διαφορετικών schemas και δομών μεταδεδομένων την ακρίβεια και την αποτελεσματικότητα των federated queries;
- Κατά πόσο το Trino μπορεί να υποστηρίξει αποδοτική distributed query execution σε πανεπιστημιακά πληροφοριακά οικοσυστήματα με πολλαπλές πηγές δεδομένων;
- Πώς επηρεάζονται ο χρόνος απόκρισης και η συνολική απόδοση του συστήματος όταν αυξάνεται ο όγκος των δεδομένων και η πολυπλοκότητα των αναλυτικών ερωτημάτων;

1.6 Μεθοδολογία υλοποίησης και έρευνας

Η μεθοδολογία της παρούσας διπλωματικής εργασίας βασίζεται σε συνδυασμό θεωρητικής ανάλυσης, αρχιτεκτονικού σχεδιασμού και πειραματικής υλοποίησης federated περιβάλλοντος Data Space. Η ερευνητική διαδικασία οργανώνεται σε διαδοχικά στάδια, ώστε να επιτευχθεί τόσο θεωρητική τεκμηρίωση όσο και πρακτική αξιολόγηση της προτεινόμενης προσέγγισης. Η μεθοδολογία περιλαμβάνει τα ακόλουθα στάδια:

- Πραγματοποίηση εκτενούς βιβλιογραφικής επισκόπησης σχετικά με Data Spaces, federated data integration, distributed query processing, semantic interoperability και cloud-native πληροφοριακές αρχιτεκτονικές.
- Ανάλυση σύγχρονων federated query engines και αξιολόγηση τεχνολογιών όπως Trino, Presto και Apache Drill ως προς την απόδοση, τη διαλειτουργικότητα και τη δυνατότητα επεκτασιμότητας.
- Μελέτη των χαρακτηριστικών και των απαιτήσεων ενός πανεπιστημιακού πληροφοριακού περιβάλλοντος με στόχο την καταγραφή των βασικών πηγών δεδομένων και των σχέσεων μεταξύ τους.
- Σχεδίαση αρχιτεκτονικής Data Space βασισμένης σε federated μοντέλο πρόσβασης, όπου οι επιμέρους βάσεις δεδομένων παραμένουν αυτόνομες και συνδέονται μέσω query federation layer.
- Δημιουργία πειραματικού περιβάλλοντος υλοποίησης με χρήση Trino και διασύνδεση πολλαπλών ετερογενών data sources μέσω connectors και catalogs.
- Ανάπτυξη διαφορετικών βάσεων δεδομένων που προσομοιώνουν πληροφοριακά συστήματα πανεπιστημιακού περιβάλλοντος, όπως δεδομένα φοιτητών, μαθημάτων, καθηγητών και εξεταστικών διαδικασιών.
- Υλοποίηση federated SQL ερωτημάτων με λειτουργίες JOIN, aggregations και αναλυτική επεξεργασία πάνω σε πολλαπλές πηγές δεδομένων.
- Εξαγωγή συμπερασμάτων σχετικά με τη λειτουργική βιωσιμότητα, τις τεχνολογικές δυνατότητες και τις μελλοντικές προοπτικές αξιοποίησης των Data Spaces σε ακαδημαϊκά πληροφοριακά οικοσυστήματα.

1.7 Πεδίο εφαρμογής στο πανεπιστημιακό περιβάλλον

Η εφαρμογή ενός περιβάλλοντος Data Space σε πανεπιστημιακό οικοσύστημα παρουσιάζει ιδιαίτερο ερευνητικό και τεχνολογικό ενδιαφέρον, καθώς τα σύγχρονα ακαδημαϊκά ιδρύματα αποτελούν

οργανισμούς με εξαιρετικά υψηλό βαθμό πληροφοριακής πολυπλοκότητας. Η λειτουργία ενός πανεπιστημίου βασίζεται σε πληθώρα ανεξάρτητων πληροφοριακών συστημάτων, τα οποία εξυπηρετούν διαφορετικές διοικητικές, ακαδημαϊκές και ερευνητικές διαδικασίες. Συστήματα διαχείρισης φοιτητών, πλατφόρμες ηλεκτρονικής μάθησης, βάσεις δεδομένων διδασκόντων, εφαρμογές εξεταστικής διαδικασίας, συστήματα αξιολόγησης μαθημάτων και αποθετήρια ερευνητικών δεδομένων λειτουργούν συνήθως απομονωμένα, χρησιμοποιώντας διαφορετικές τεχνολογίες αποθήκευσης και διαφορετικές δομές δεδομένων. Ως αποτέλεσμα, η πληροφορία παραμένει κατακερματισμένη και η παραγωγή ενοποιημένης γνώσης καθίσταται ιδιαίτερα δύσκολη.

Στο πλαίσιο αυτό, το πεδίο εφαρμογής της παρούσας εργασίας επικεντρώνεται στη δημιουργία ενός federated περιβάλλοντος πρόσβασης δεδομένων το οποίο θα μπορεί να διασυνδέει διαφορετικές πανεπιστημιακές πληροφοριακές πηγές χωρίς να απαιτείται φυσική συγκέντρωση των δεδομένων σε ενιαία βάση. Η συγκεκριμένη προσέγγιση θεωρείται ιδιαίτερα σημαντική για πανεπιστημιακά ιδρύματα, καθώς τα περισσότερα διαθέτουν legacy πληροφοριακές υποδομές που έχουν αναπτυχθεί σταδιακά σε διαφορετικές χρονικές περιόδους και στηρίζονται σε διαφορετικά DBMS, διαφορετικά πρωτόκολλα επικοινωνίας και διαφορετικές λογικές μοντελοποίησης δεδομένων.

1.8 Συνεισφορά της εργασίας

Η παρούσα διπλωματική εργασία επιχειρεί να συμβάλει τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο στον τομέα της ενοποιημένης πρόσβασης σε ετερογενή δεδομένα μέσω σύγχρονων federated αρχιτεκτονικών. Η βασική συνεισφορά της εργασίας αφορά τη διερεύνηση και υλοποίηση ενός ολοκληρωμένου περιβάλλοντος Data Space προσαρμοσμένου στις ανάγκες πανεπιστημιακών πληροφοριακών οικοσυστημάτων, στα οποία η πληροφορία παραμένει συνήθως κατακερματισμένη σε πολλαπλά ανεξάρτητα συστήματα.

Σε θεωρητικό επίπεδο, η εργασία συμβάλλει στην ανάλυση των σύγχρονων τεχνολογιών data federation, semantic interoperability και distributed query processing, παρουσιάζοντας αναλυτικά τις αρχές λειτουργίας federated query engines και των αποκεντρωμένων πληροφοριακών αρχιτεκτονικών. Μέσα από τη συστηματική βιβλιογραφική επισκόπηση, επιχειρείται η αποσαφήνιση των διαφορών μεταξύ Data Spaces, data warehouses, data lakes και παραδοσιακών συστημάτων ολοκλήρωσης δεδομένων, συμβάλλοντας στη βαθύτερη κατανόηση των νέων τάσεων διαχείρισης ψηφιακής πληροφορίας. Σε πρακτικό επίπεδο, η εργασία συνεισφέρει μέσω της σχεδίασης και υλοποίησης ενός federated περιβάλλοντος βασισμένου στο Trino, το οποίο επιτρέπει την εκτέλεση ενοποιημένων ερωτημάτων SQL πάνω σε πολλαπλές πηγές δεδομένων. Η συγκεκριμένη υλοποίηση επιτρέπει την αξιολόγηση πραγματικών ζητημάτων distributed query execution, όπως η διαχείριση λειτουργιών JOIN μεταξύ διαφορετικών DBMS, η επεκτασιμότητα του συστήματος, η κατανομή φόρτου εργασίας και η συνολική απόδοση σε περιβάλλοντα αυξημένου όγκου δεδομένων.

1.9 Δομή της εργασίας

Η παρούσα διπλωματική εργασία οργανώνεται σε έξι βασικά κεφάλαια, τα οποία αναπτύσσονται με τρόπο ώστε να παρουσιάζεται σταδιακά το θεωρητικό υπόβαθρο, η τεχνολογική ανάλυση, η σχεδίαση, η υλοποίηση και η αξιολόγηση του προτεινόμενου περιβάλλοντος Data Space. Στο πρώτο κεφάλαιο παρουσιάζεται η εισαγωγή της εργασίας. Αναλύεται το αντικείμενο της έρευνας, περιγράφεται το πρόβλημα της ενοποιημένης πρόσβασης σε ετερογενή δεδομένα και τεκμηριώνεται η ανάγκη

ανάπτυξης federated πληροφοριακών αρχιτεκτονικών. Παράλληλα, παρουσιάζονται οι στόχοι της εργασίας, τα ερευνητικά ερωτήματα, η μεθοδολογία υλοποίησης και το πεδίο εφαρμογής στο πανεπιστημιακό περιβάλλον. Το δεύτερο κεφάλαιο περιλαμβάνει το θεωρητικό υπόβαθρο και τη βιβλιογραφική επισκόπηση. Εξετάζονται οι έννοιες των ετερογενών πληροφοριακών συστημάτων, των Data Spaces, της federated πρόσβασης δεδομένων και των distributed query engines. Επιπλέον, παρουσιάζονται σύγχρονες τεχνολογίες όπως Trino, Presto και Apache Drill, ενώ πραγματοποιείται συγκριτική ανάλυση διαφορετικών προσεγγίσεων data integration και data virtualization.

Στο τρίτο κεφάλαιο αναλύονται οι τεχνολογίες, τα εργαλεία και οι αρχιτεκτονικές υλοποίησης του προτεινόμενου συστήματος. Παρουσιάζονται οι απαιτήσεις ενός περιβάλλοντος Data Space, οι διαφορετικές αρχιτεκτονικές προσεγγίσεις ολοκλήρωσης δεδομένων και τα βασικά χαρακτηριστικά του Trino ως federated query engine. Παράλληλα, αναλύονται οι πηγές δεδομένων του πανεπιστημιακού περιβάλλοντος και τα εργαλεία υποστήριξης της υλοποίησης.

Το τέταρτο κεφάλαιο επικεντρώνεται στη σχεδίαση του προτεινόμενου συστήματος Data Space. Περιγράφεται η γενική αρχιτεκτονική του συστήματος, η ροή δεδομένων μεταξύ των πηγών και του federated layer, η μοντελοποίηση των δεδομένων και ο σχεδιασμός των ενοποιημένων queries. Επιπλέον, εξετάζονται ζητήματα σημασιολογικής ετερογένειας, ασφάλειας και προστασίας δεδομένων. Στο πέμπτο κεφάλαιο παρουσιάζεται η υλοποίηση και η πειραματική εφαρμογή της προτεινόμενης αρχιτεκτονικής. Περιγράφεται το περιβάλλον ανάπτυξης, η εγκατάσταση και παραμετροποίηση του Trino, η διασύνδεση των διαφορετικών πηγών δεδομένων και η υλοποίηση federated SQL ερωτημάτων. Παράλληλα, πραγματοποιείται πειραματική αξιολόγηση της απόδοσης του συστήματος μέσω μετρήσεων του χρόνου απόκρισης, της επεκτασιμότητας και της αποδοτικότητας εκτέλεσης ερωτημάτων.

Κεφάλαιο 2: Θεωρητικό υπόβαθρο και βιβλιογραφική επισκόπηση

2.1 Η έννοια των δεδομένων στη σύγχρονη πληροφοριακή εποχή

Η σύγχρονη πληροφοριακή εποχή χαρακτηρίζεται από τη μετατροπή των δεδομένων σε στρατηγικό παραγωγικό πόρο υψηλής αξίας, ο οποίος επηρεάζει άμεσα την επιχειρησιακή λειτουργία, τη λήψη αποφάσεων, την επιστημονική έρευνα και τη συνολική ανάπτυξη των ψηφιακών οικοσυστημάτων. Η εκθετική αύξηση της παραγωγής δεδομένων προέρχεται από την ευρεία χρήση πληροφοριακών συστημάτων, cloud υποδομών, IoT τεχνολογιών, πλατφορμών κοινωνικής δικτύωσης, εφαρμογών τεχνητής νοημοσύνης και κατανεμημένων ψηφιακών υπηρεσιών, δημιουργώντας ένα περιβάλλον στο οποίο η πληροφορία παράγεται, διακινείται και μετασχηματίζεται με ρυθμούς πρωτοφανείς για τα ιστορικά δεδομένα της πληροφορικής επιστήμης[1], [2]. Σύμφωνα με τους Curry et al.[4], η πληροφορία δεν αντιμετωπίζεται πλέον ως στατικό αντικείμενο αποθήκευσης αλλά ως δυναμικό περιουσιακό στοιχείο, το οποίο πρέπει να αξιοποιείται σε πραγματικό χρόνο μέσω προηγμένων μηχανισμών αναλυτικής επεξεργασίας και federated processing.

Η εξέλιξη αυτή έχει μεταβάλει ριζικά τη φιλοσοφία σχεδίασης των πληροφοριακών υποδομών. Τα παραδοσιακά μοντέλα κεντρικής αποθήκευσης και μονολιθικής διαχείρισης δεδομένων αδυνατούν να ανταποκριθούν αποτελεσματικά στις σύγχρονες απαιτήσεις επεκτασιμότητας, ελαστικότητας και κατανεμημένης ανάλυσης δεδομένων[8]. Οι οργανισμοί λειτουργούν πλέον μέσω αποκεντρωμένων οικοσυστημάτων δεδομένων, όπου διαφορετικές εφαρμογές και υπηρεσίες παράγουν πληροφορία σε πολλαπλές μορφές και με διαφορετικές λογικές μοντελοποίησης. Ως αποτέλεσμα, αναπτύσσονται νέες αρχιτεκτονικές όπως τα Data Spaces, τα federated ecosystems και τα cloud-native αναλυτικά περιβάλλοντα, τα οποία επιτρέπουν ενοποιημένη αξιοποίηση κατανεμημένων πληροφοριακών πόρων χωρίς ανάγκη φυσικής συγκέντρωσης των δεδομένων[5].

Στη σύγχρονη πληροφοριακή εποχή, οι οργανισμοί διαχειρίζονται ημιδομημένα και μη δομημένα δεδομένα, όπως έγγραφα JSON, αρχεία XML, πολυμεσικό περιεχόμενο, αρχεία καταγραφής συστημάτων και ροές δεδομένων από IoT αισθητήρες[10]. Η πολυμορφία αυτή δημιουργεί σημαντικές τεχνολογικές προκλήσεις ως προς την επεξεργασία, την ενοποίηση και την αποδοτική αξιοποίηση της πληροφορίας. Σύμφωνα με τους Putrama και Martinek[6], η ετερογένεια των δεδομένων αποτελεί έναν από τους σημαντικότερους περιοριστικούς παράγοντες στην ανάπτυξη επεκτάσιμων πληροφοριακών αρχιτεκτονικών, καθώς διαφορετικές μορφές πληροφορίας απαιτούν διαφορετικούς μηχανισμούς βελτιστοποίησης αποθήκευσης και διαφορετικές τεχνικές query execution. Ταυτόχρονα, η ανάπτυξη τεχνολογιών προηγμένης ανάλυσης δεδομένων και μηχανικής μάθησης έχει ενισχύσει περαιτέρω τη στρατηγική σημασία των δεδομένων. Οι σύγχρονες προγνωστικές εφαρμογές και εφαρμογές βασισμένες στην Τεχνητή Νοημοσύνη βασίζονται στη δυνατότητα επεξεργασίας τεράστιων συνόλων δεδομένων υψηλής ποικιλομορφίας, ώστε να παράγονται αυτοματοποιημένα μοντέλα πρόβλεψης και ανάλυσης[11]. Η ποιότητα, η πληρότητα και η διαθεσιμότητα των δεδομένων επηρεάζουν άμεσα την αποτελεσματικότητα των αναλυτικών διαδικασιών, γεγονός που καθιστά απαραίτητη την ανάπτυξη μηχανισμών data governance, semantic interoperability και distributed query optimization.

Οι σύγχρονοι οργανισμοί απαιτούν δυνατότητα άμεσης πρόσβασης και ανάλυσης δεδομένων σε πραγματικό χρόνο, γεγονός που περιορίζει σημαντικά την αποτελεσματικότητα των παραδοσιακών batch processing architectures[12]. Οι federated query engines και οι cloud-native αναλυτικές πλατφόρμες αναπτύσσονται ακριβώς για να καλύψουν αυτή την ανάγκη, επιτρέποντας δυναμική σύνθεση δεδομένων από πολλαπλές πηγές με χαμηλό latency και υψηλή επεκτασιμότητα[13].

Η σύγχρονη πληροφοριακή εποχή χαρακτηρίζεται από αυξημένες απαιτήσεις προστασίας δεδομένων και κυβερνοασφάλειας. Η τεράστια διακίνηση ψηφιακής πληροφορίας έχει δημιουργήσει σοβαρούς κινδύνους παραβίασης προσωπικών δεδομένων και μη εξουσιοδοτημένης πρόσβασης[14]. Για τον λόγο αυτό, οι σύγχρονες πληροφοριακές αρχιτεκτονικές ενσωματώνουν ολοένα και περισσότερο μηχανισμούς privacy-aware processing, authentication orchestration και decentralized governance, ώστε να διασφαλίζεται ασφαλής και ελεγχόμενη διαχείριση πληροφορίας[15]. Τέλος, η σημασία των δεδομένων συνδέεται άμεσα με τη δυνατότητα λήψης τεκμηριωμένων αποφάσεων. Οι οργανισμοί βασίζονται ολοένα και περισσότερο σε στρατηγικές λειτουργίας βασισμένες στα δεδομένα, χρησιμοποιώντας πίνακες ελέγχου, προγνωστική ανάλυση και υποδομές επιχειρηματικής ευφυΐας για τη βελτιστοποίηση των διαδικασιών τους[16]. Μέσα σε αυτό το πλαίσιο, η αποτελεσματική διαχείριση και ενοποίηση δεδομένων αποτελεί κρίσιμο παράγοντα ανταγωνιστικότητας και τεχνολογικής εξέλιξης.

2.2 Ετερογενή πληροφοριακά συστήματα

Τα ετερογενή πληροφοριακά συστήματα αποτελούν βασικό χαρακτηριστικό των σύγχρονων οργανωσιακών και τεχνολογικών οικοσυστημάτων, καθώς οι περισσότεροι οργανισμοί αναπτύσσουν πληροφοριακές υποδομές διαφορετικής τεχνολογικής προέλευσης και διαφορετικών λειτουργικών προδιαγραφών. Η σταδιακή ανάπτυξη εφαρμογών σε διαφορετικές χρονικές περιόδους, η υιοθέτηση πολλαπλών DBMS και η μετάβαση προς hybrid cloud περιβάλλοντα έχουν δημιουργήσει κατακεκομμένα πληροφοριακά οικοσυστήματα στα οποία η πληροφορία παραμένει απομονωμένη σε διαφορετικές πλατφόρμες[17]. Σύμφωνα με τους Gu et al.[2], η πληροφοριακή ετερογένεια αποτελεί μία από τις σημαντικότερες προκλήσεις της σύγχρονης data engineering, καθώς περιορίζει τη δυνατότητα παραγωγής ενοποιημένης γνώσης και αυξάνει σημαντικά τη λειτουργική πολυπλοκότητα των πληροφοριακών συστημάτων.

Η ετερογένεια μπορεί να εμφανίζεται σε διαφορετικά επίπεδα. Σε τεχνολογικό επίπεδο, διαφορετικά πληροφοριακά συστήματα χρησιμοποιούν διαφορετικές βάσεις δεδομένων, διαφορετικές γλώσσες προγραμματισμού και διαφορετικές υποδομές αποθήκευσης[4]. Σε δομικό επίπεδο, οι βάσεις δεδομένων μπορεί να διαθέτουν διαφορετικά schemas, διαφορετικά data models και διαφορετικές logical relationships μεταξύ οντοτήτων. Παράλληλα, σε σημασιολογικό επίπεδο, διαφορετικά πληροφοριακά συστήματα συχνά χρησιμοποιούν διαφορετικούς ορισμούς και διαφορετικές αναπαραστάσεις για την ίδια πληροφοριακή οντότητα, δημιουργώντας σημαντικά προβλήματα semantic interoperability[18]. Η ύπαρξη τέτοιων ετερογενών πληροφοριακών οικοσυστημάτων δημιουργεί σοβαρούς περιορισμούς ως προς την ολοκλήρωση δεδομένων και την ανάπτυξη προηγμένων αναλυτικών διαδικασιών. Οι οργανισμοί συχνά αδυνατούν να συνδυάσουν αποτελεσματικά δεδομένα από πολλαπλές πηγές, γεγονός που περιορίζει τη δυνατότητα αναλυτικής επεξεργασίας σε πραγματικό χρόνο και αυξάνει σημαντικά τον χρόνο παραγωγής αναφορών[19]. Το πρόβλημα εμφανίζεται ιδιαίτερα έντονα σε πανεπιστημιακά περιβάλλοντα, όπου διαφορετικές υπηρεσίες λειτουργούν μέσω ανεξάρτητων πληροφοριακών συστημάτων χωρίς ενιαίο επίπεδο πρόσβασης και διασύνδεσης.

2.2.1 Ορισμός και χαρακτηριστικά

Ως ετερογενές πληροφοριακό σύστημα ορίζεται ένα πληροφοριακό περιβάλλον στο οποίο συνυπάρχουν πολλαπλές ανεξάρτητες εφαρμογές και βάσεις δεδομένων με διαφορετικές τεχνολογικές, δομικές και λειτουργικές προδιαγραφές[9]. Τα συστήματα αυτά διαφέρουν ως προς τις τεχνολογίες αποθήκευσης, τις αρχιτεκτονικές λογισμικού, τα πρωτόκολλα επικοινωνίας και τα μοντέλα μεταδεδωμένων που χρησιμοποιούν. Ένα από τα βασικότερα χαρακτηριστικά των ετερογενών συστημάτων είναι η τεχνολογική ανομοιογένεια. Διαφορετικά συστήματα ενδέχεται να χρησιμοποιούν σχεσιακά DBMS, NoSQL databases, distributed file systems ή cloud object storage environments[20]. Παράλληλα, κάθε πλατφόρμα μπορεί να διαθέτει διαφορετικούς μηχανισμούς αυθεντικοποίησης, διαφορετικά APIs και διαφορετικές διαλέκτους SQL.

Εξίσου σημαντικό χαρακτηριστικό αποτελεί η σημασιολογική ετερογένεια. Οι Masmoudi et al.[21] επισημαίνουν ότι η ίδια πληροφοριακή οντότητα μπορεί να αποθηκεύεται με διαφορετικό τρόπο σε διαφορετικά συστήματα, χρησιμοποιώντας διαφορετικά metadata schemas, διαφορετικούς τύπους δεδομένων και διαφορετικές συμβάσεις ονοματοδοσίας. Η κατάσταση αυτή δημιουργεί σοβαρές δυσκολίες semantic integration και περιορίζει τη δυνατότητα αυτοματοποιημένης επεξεργασίας πληροφορίας. Ορισμένα συστήματα σχεδιάζονται για OLTP workloads, ενώ άλλα εξυπηρετούν αναλυτικές ή streaming λειτουργίες[22]. Η διαφορετική λειτουργική στόχευση κάθε πλατφόρμας επηρεάζει άμεσα τη δυνατότητα ενοποιημένης αξιοποίησης δεδομένων.

2.2.2 Προκλήσεις ενοποίησης

Η ενοποίηση ετερογενών πληροφοριακών συστημάτων αποτελεί μία από τις σημαντικότερες τεχνολογικές προκλήσεις της σύγχρονης πληροφορικής επιστήμης. Η βασική δυσκολία αφορά τη διασύνδεση πολλαπλών ανεξάρτητων πληροφοριακών πηγών χωρίς να αλλοιώνεται η λειτουργική αυτονομία των επιμέρους συστημάτων[21]. Ένα από τα σημαντικότερα προβλήματα αφορά την ασυμβατότητα δομών δεδομένων. Διαφορετικά πληροφοριακά συστήματα χρησιμοποιούν διαφορετικά schemas και διαφορετικά data models, γεγονός που καθιστά εξαιρετικά δύσκολη τη σύνθεση ενοποιημένων αναλυτικών ερωτημάτων[6]. Επιπλέον, η ύπαρξη διαφορετικών formats δεδομένων δημιουργεί σημαντικές απαιτήσεις transformation και normalization.

Η ανάγκη πρόσβασης σε πληροφορία σε πραγματικό χρόνο περιορίζει σημαντικά την αποτελεσματικότητα των παραδοσιακών ETL-based μοντέλων ολοκλήρωσης. Οι batch synchronization διαδικασίες δημιουργούν καθυστερήσεις και αυξάνουν τη λειτουργική επιβάρυνση, ιδιαίτερα σε περιβάλλοντα υψηλής μεταβλητότητας δεδομένων[7]. Για τον λόγο αυτό, αναπτύσσονται ολοένα και περισσότερες federated architectures και distributed query engines που επιτρέπουν δυναμική πρόσβαση σε δεδομένα χωρίς φυσική μεταφορά τους. Σημαντική πρόκληση αποτελεί επίσης η διαχείριση της κυβερνοασφάλειας και η προστασία της ιδιωτικότητας σε καταναμημένα οικοσυστήματα. Η διασύνδεση πολλαπλών πληροφοριακών συστημάτων αυξάνει την επιφάνεια επιθέσεων και δημιουργεί σύνθετες απαιτήσεις authorization orchestration[23].

2.2.3 Ζητήματα συμβατότητας και διαλειτουργικότητας

Η συμβατότητα και η διαλειτουργικότητα αποτελούν κρίσιμες προϋποθέσεις για την επιτυχημένη λειτουργία ενοποιημένων πληροφοριακών οικοσυστημάτων. Η τεχνολογική συμβατότητα αφορά τη

δυνατότητα διαφορετικών πληροφοριακών συστημάτων να επικοινωνούν μέσω κοινών interfaces, APIs και πρωτοκόλλων επικοινωνίας. Η χρήση διαφορετικών DBMS, διαφορετικών μορφών δεδομένων και διαφορετικών μηχανισμών αυθεντικοποίησης δημιουργεί σημαντικά εμπόδια στη διασύνδεση πληροφοριακών πλατφορμών. Ιδιαίτερη σημασία παρουσιάζει επίσης η λειτουργική διαλειτουργικότητα, δηλαδή η δυνατότητα συνεργασίας διαφορετικών υπηρεσιών σε πραγματικές συνθήκες λειτουργίας. Τα σύγχρονα distributed ecosystems απαιτούν δυναμική ανταλλαγή δεδομένων, συγχρονισμό σε πραγματικό χρόνο και συντονισμένη αναλυτική επεξεργασία[18].

2.3 Η έννοια των Data Spaces

Η εκθετική αύξηση του παγκόσμιου όγκου δεδομένων έχει μεταβάλει ριζικά τις απαιτήσεις σχεδιασμού πληροφοριακών υποδομών μεγάλης κλίμακας. Σύμφωνα με εκτιμήσεις της IDC, ο συνολικός όγκος παγκόσμιων δεδομένων αναμένεται να ξεπεράσει τα 175 zettabytes μέχρι το 2025, παρουσιάζοντας αύξηση μεγαλύτερη του 430% σε σχέση με το 2018. Η τεράστια αυτή αύξηση οφείλεται κυρίως στη διάδοση cloud-native εφαρμογών, IoT συσκευών, AI-driven υπηρεσιών και distributed digital ecosystems. Μέσα σε αυτό το περιβάλλον, τα παραδοσιακά μοντέλα κεντρικής αποθήκευσης δεδομένων εμφανίζουν σοβαρούς περιορισμούς ως προς την επεκτασιμότητα, το λειτουργικό κόστος και τη δυνατότητα επεξεργασίας σε πραγματικό χρόνο κατανεμημένης πληροφορίας[2]. Ως αποτέλεσμα, αναπτύχθηκαν τα Data Spaces ως νέα γενιά αποκεντρωμένων πληροφοριακών αρχιτεκτονικών, σχεδιασμένων να παρέχουν ενοποιημένη πρόσβαση σε ετερογενή δεδομένα χωρίς ανάγκη φυσικής συγκέντρωσής τους σε κεντρικά αποθετήρια[3].

Η βασική φιλοσοφία των Data Spaces στηρίζεται στην έννοια της federated data integration, σύμφωνα με την οποία οι πληροφορίες παραμένουν στις αρχικές πηγές αποθήκευσης και η ενοποίηση πραγματοποιείται λογικά μέσω distributed query engines και semantic interoperability μηχανισμών[1]. Η προσέγγιση αυτή διαφοροποιείται ουσιαστικά από τις παραδοσιακές ETL-based αρχιτεκτονικές, οι οποίες απαιτούν συνεχή extraction, transformation και loading διαδικασία για τη συγκέντρωση δεδομένων σε κεντρικά data warehouses. Σύμφωνα με τους Gu et al.[2], οι διαδικασίες ETL ευθύνονται έως και για το 60% της συνολικής λειτουργικής επιβάρυνσης σε enterprise analytics environments, ενώ παράλληλα αυξάνουν σημαντικά τον χρόνο synchronization και τη διπλή αποθήκευση δεδομένων.

Τα σύγχρονα Data Spaces σχεδιάζονται με βάση αρχές decentralized governance και data sovereignty, επιτρέποντας στους οργανισμούς να διατηρούν πλήρη έλεγχο των πληροφοριών τους[24]. Η συγκεκριμένη δυνατότητα αποκτά ιδιαίτερη σημασία σε περιβάλλοντα όπου η πληροφορία υπόκειται σε κανονισμούς ιδιωτικότητας και περιορισμούς πρόσβασης, όπως πανεπιστημιακά οικοσυστήματα, ιατρικές πλατφόρμες και ερευνητικά αποθετήρια δεδομένων.

Τα Data Spaces ορίζονται ως αποκεντρωμένα πληροφοριακά οικοσυστήματα μέσα στα οποία διαφορετικοί οργανισμοί, εφαρμογές και ψηφιακές υποδομές μπορούν να ανταλλάσσουν και να αξιοποιούν δεδομένα με ασφαλή, διαλειτουργικό και ελεγχόμενο τρόπο χωρίς φυσική μεταφορά της πληροφορίας[1]. Η βασική αρχή λειτουργίας τους βασίζεται στη λογική federation, όπου τα δεδομένα παραμένουν στις αρχικές πηγές αποθήκευσης και προσπελάζονται μέσω unified logical access layer[5]. Σύμφωνα με τους Jarke και Quix[1], ένα πλήρως λειτουργικό Data Space περιλαμβάνει πέντε βασικά επίπεδα λειτουργίας:

Πίνακας 2.1: Βασικά επίπεδα λειτουργίας ενός Data Space

Επίπεδο	Λειτουργία
Data Sources Layer	Διατήρηση ετερογενών πηγών δεδομένων
Federation Layer	Ενοποιημένη εκτέλεση queries
Semantic Layer	Αντιστοίχιση metadata και schemas
Governance Layer	Έλεγχος πρόσβασης και πολιτικές ασφαλείας
Analytics Layer	Επεξεργασία αναλυτικών φορτίων εργασίας

Η συγκεκριμένη πολυεπίπεδη αρχιτεκτονική επιτρέπει την ταυτόχρονη πρόσβαση σε σχεσιακές βάσεις δεδομένων, NoSQL systems, cloud object storage και streaming platforms χωρίς ανάγκη φυσικής συγκέντρωσης δεδομένων[14]. Η εξέλιξη των Data Spaces συνδέεται άμεσα με την αδυναμία των παραδοσιακών πληροφοριακών μοντέλων να διαχειριστούν αποτελεσματικά τον αυξανόμενο όγκο distributed δεδομένων. Κατά τη δεκαετία του 1990, οι περισσότερες enterprise αρχιτεκτονικές βασίζονταν αποκλειστικά σε centralized data warehouses. Ωστόσο, η εμφάνιση cloud computing, distributed systems και web-scale applications οδήγησε σε δραματική αύξηση της πληροφοριακής ετερογένειας[15].

Η πρώτη γενιά data integration architectures βασίστηκε σε ETL pipelines και centralized repositories. Παρότι οι αρχιτεκτονικές αυτές παρείχαν δυνατότητα συγκεντρωτικής ανάλυσης, παρουσίαζαν σημαντικούς περιορισμούς ως προς την επεκτασιμότητα και την καθυστέρηση συγχρονισμού[16]. Σύμφωνα με έρευνες της Gartner, enterprise ETL environments καταναλώνουν έως και 70% του συνολικού χρόνου που αφιερώνεται στη μηχανική δεδομένων για διαδικασίες προετοιμασίας δεδομένων. Η δεύτερη γενιά εμφανίστηκε με την ανάπτυξη distributed file systems και cloud-native infrastructures. Οι τεχνολογίες Hadoop, Spark και NoSQL databases επέτρεψαν επεξεργασία δεδομένων μεγάλης κλίμακας, ωστόσο δεν αντιμετώπισαν αποτελεσματικά το πρόβλημα semantic interoperability[17].

Η τρίτη γενιά πληροφοριακών αρχιτεκτονικών οδήγησε στην ανάπτυξη των Data Spaces, τα οποία ενσωματώνουν έννοιες όπως federated access, semantic virtualization και decentralized governance[3]. Η Ευρωπαϊκή Ένωση επένδυσε ιδιαίτερα στην ανάπτυξη Common European Data Spaces, χρηματοδοτώντας μεγάλης κλίμακας έργα στους τομείς υγείας, εκπαίδευσης, βιομηχανίας και ενέργειας[25]. Τα Data Spaces βασίζονται σε συγκεκριμένες αρχές σχεδίασης που διαφοροποιούνται ουσιαστικά από τις παραδοσιακές πληροφοριακές αρχιτεκτονικές.

Πίνακας 2.2: Βασικές αρχές σχεδίασης των Data Spaces

Βασική αρχή	Περιγραφή
Decentralization	Τα δεδομένα παραμένουν στις αρχικές πηγές
Federation	Ενοποιημένη πρόσβαση μέσω query federation
Interoperability	Διαλειτουργικότητα μεταξύ ετερογενών συστημάτων
Data Sovereignty	Έλεγχος ιδιοκτησίας δεδομένων
Επεκτασιμότητα	Υποστήριξη distributed μεγάλης κλίμακας workloads

Πρόσβαση σε πραγματικό χρόνο	Δυναμική πρόσβαση χωρίς batch synchronization
-------------------------------------	---

Σύμφωνα με τους Bacco et al.[18], τα Data Spaces παρουσιάζουν έως και 52% χαμηλότερο integration overhead σε σχέση με centralized ETL architectures, ενώ παράλληλα μειώνουν σημαντικά το κόστος διπλής αποθήκευσης δεδομένων. Η σημαντικότερη διαφορά μεταξύ Data Space και Data Warehouse αφορά τη φιλοσοφία αποθήκευσης και πρόσβασης στα δεδομένα.

Πίνακας 2.3: Σύγκριση Data Space και Data Warehouse

Χαρακτηριστικό	Data Space	Data Warehouse
Αποθήκευση	Κατανεμημένη	Κεντρική
ETL διαδικασίες	Ελάχιστες	Υποχρεωτικές
Query execution	Federated	Centralized
Επεκτασιμότητα	Elastic	Περιορισμένη
Δεδομένα πραγματικού χρόνου	Υποστηρίζονται	Περιορισμένα
Data duplication	Χαμηλή	Υψηλή

Τα Data Warehouses απαιτούν συνεχή ETL synchronization, γεγονός που αυξάνει σημαντικά τη λειτουργική επιβάρυνση[4]. Αντίθετα, τα Data Spaces επιτρέπουν dynamic query execution απευθείας στις πηγές δεδομένων. Παρότι τα Data Lakes και τα Data Spaces χρησιμοποιούνται για διαχείριση μεγάλων δεδομένων, παρουσιάζουν ουσιαστικές διαφορές.

Πίνακας 2.4: Σύγκριση Data Space και Data Lake

Χαρακτηριστικό	Data Space	Data Lake
Ενοποίηση	Λογική	Φυσική
Data governance	Ενσωματωμένο	Περιορισμένο
Semantic interoperability	Υψηλή	Χαμηλή
Federated access	Ναι	Όχι
Αυτονομία πηγών	Διατηρείται	Χάνεται

Οι Chaudhari και Charate[11] επισημαίνουν ότι τα Data Lakes αντιμετωπίζουν συχνά προβλήματα “data swamp”, όπου η έλλειψη semantic governance καθιστά δύσκολη την αξιοποίηση πληροφορίας. Οι παραδοσιακές προσεγγίσεις data integration βασίζονται κυρίως σε ETL pipelines και middleware synchronization frameworks[19]. Τα μοντέλα αυτά απαιτούν φυσική μεταφορά και transformation δεδομένων, γεγονός που δημιουργεί υψηλό λειτουργικό κόστος και καθυστέρηση συγχρονισμού.

Πίνακας 2.5: Σύγκριση κλασικής ολοκλήρωσης δεδομένων και Data Space

Παράμετρος	Κλασική ολοκλήρωση	Data Space
Integration model	ETL	Federation
Real-time access	Περιορισμένη	Υψηλή
Data ownership	Κεντριοποιημένη	Αποκεντρωμένη

Flexibility	Χαμηλή	Υψηλή
Query federation	Όχι	Ναι

Σύμφωνα με τους Nelamangala[3] και Koripalli[5], τα federated ecosystems παρουσιάζουν σημαντικά καλύτερη προσαρμοστικότητα σε περιβάλλοντα υψηλής μεταβλητότητας δεδομένων.

2.4 Ομοσπονδιακή πρόσβαση σε δεδομένα

Η ομοσπονδιακή πρόσβαση σε δεδομένα (federated data access) αποτελεί μία από τις σημαντικότερες τεχνολογικές εξελίξεις στη σύγχρονη επιστήμη διαχείρισης πληροφοριών, καθώς επιτρέπει την ενοποιημένη αξιοποίηση κατακευμασμένων δεδομένων χωρίς φυσική μεταφορά ή αντιγραφή τους σε κεντρικά αποθετήρια[1], [2]. Η ανάγκη ανάπτυξης federated architectures προέκυψε από την εκθετική αύξηση της πληροφοριακής ετερογένειας, η οποία χαρακτηρίζει τα σύγχρονα οργανωσιακά οικοσυστήματα. Οι επιχειρήσεις, τα πανεπιστήμια και οι ερευνητικοί οργανισμοί διαχειρίζονται πλέον δεδομένα που βρίσκονται κατακευμασμένα σε σχεσιακά DBMS, NoSQL υποδομές, cloud storage platforms, data lakes και streaming environments, δημιουργώντας σύνθετα distributed ecosystems υψηλής πολυπλοκότητας[3].

Οι παραδοσιακές προσεγγίσεις data integration βασίζονταν κυρίως σε ETL διαδικασίες, μέσω των οποίων τα δεδομένα εξαγόταν, μετασχηματίζονταν και αποθηκεύονταν σε centralized data warehouses[8]. Ωστόσο, η προσέγγιση αυτή παρουσιάζει σημαντικούς περιορισμούς σε περιβάλλοντα μεγάλης κλίμακας. Σύμφωνα με τους Gu et al.[2], τα ETL pipelines ευθύνονται για έως και 60% του συνολικού χρόνου που αφιερώνεται σε ροές εργασίας μηχανικής δεδομένων, ενώ παράλληλα δημιουργούν υψηλό βαθμό διπλής αποθήκευσης δεδομένων και αυξημένη καθυστέρηση συγχρονισμού. Επιπλέον, η ανάγκη συνεχούς αντιγραφής δεδομένων αυξάνει σημαντικά το λειτουργικό κόστος και περιορίζει τη δυνατότητα αναλυτικής επεξεργασίας σε πραγματικό χρόνο. Η ομοσπονδιακή πρόσβαση επιχειρεί να επιλύσει τα προβλήματα αυτά μέσω distributed query execution και virtualized data integration. Στα ομοσπονδιακά οικοσυστήματα, τα δεδομένα παραμένουν στις αρχικές πηγές αποθήκευσης, ενώ οι χρήστες αποκτούν ενιαία λογική πρόσβαση μέσω federated query engines και semantic interoperability layers[24]. Η συγκεκριμένη προσέγγιση επιτρέπει πρόσβαση σε πραγματικό χρόνο σε ετερογενή datasets, μειώνοντας σημαντικά το integration overhead και βελτιώνοντας τη δυνατότητα επεκτάσιμης αναλυτικής επεξεργασίας.

2.4.1 Έννοια της ομοσπονδιακής πρόσβασης

Η έννοια της ομοσπονδιακής πρόσβασης αναφέρεται στη δυνατότητα εκτέλεσης ενοποιημένων queries πάνω σε πολλαπλές κατακευμασμένες πηγές δεδομένων μέσω ενιαίου logical access layer[5]. Αντί να μεταφέρονται τα δεδομένα σε κεντρικό αποθετήριο, τα federated systems αναλαμβάνουν να διασυνδέουν δυναμικά διαφορετικές πληροφοριακές πηγές και να εκτελούν distributed query processing σε πραγματικό χρόνο[15]. Η λειτουργία ενός federated περιβάλλοντος βασίζεται συνήθως σε τρία βασικά επίπεδα:

Πίνακας 2.6: Βασικά επίπεδα λειτουργίας της federated πρόσβασης

Επίπεδο	Λειτουργία
---------	------------

Source Layer	Διατήρηση των αρχικών δεδομένων
Federation Layer	Query orchestration και execution
Semantic Layer	Αντιστοίχιση schemas και metadata

Το federation layer αναλαμβάνει query parsing, decomposition, optimization και distributed execution. Όταν ένας χρήστης εκτελεί ένα query, το σύστημα αναλύει τις πηγές δεδομένων που απαιτούνται, δημιουργεί execution plan και αποστέλλει subqueries στις επιμέρους βάσεις δεδομένων[16]. Τα αποτελέσματα επιστρέφονται στο federation layer, όπου πραγματοποιείται aggregation, filtering και final result composition. Σύμφωνα με τους Jarke και Quix[1], η ομοσπονδιακή πρόσβαση μειώνει σημαντικά το integration complexity και επιτρέπει αναλυτική επεξεργασία σε πραγματικό χρόνο σε ετερογενή οικοσυστήματα. Επιπλέον, οι Nelamangala[3] και Koripalli[5] επισημαίνουν ότι τα σύγχρονα federated systems εμφανίζουν σημαντικά υψηλότερη προσαρμοστικότητα σε περιβάλλοντα συνεχούς μεταβολής δεδομένων συγκριτικά με centralized integration models.

2.4.2 Πλεονεκτήματα και περιορισμοί

Η ομοσπονδιακή πρόσβαση παρουσιάζει σημαντικά πλεονεκτήματα ως προς την επεκτασιμότητα, την ευελιξία και τη δυνατότητα αξιοποίησης δεδομένων σε πραγματικό χρόνο.

Πίνακας 2.7: Πλεονεκτήματα της ομοσπονδιακής πρόσβασης σε δεδομένα

Πλεονεκτήματα	Ανάλυση
Real-time πρόσβαση	Δεν απαιτεί batch synchronization
Μείωση data duplication	Τα δεδομένα παραμένουν στις πηγές
Χαμηλότερο λειτουργικό κόστος	Μείωση ETL διαδικασιών
Επεκτασιμότητα	Distributed execution σε πολλαπλά nodes
Data sovereignty	Διατήρηση αυτονομίας πληροφοριών
Flexibility	Υποστήριξη ετερογενών συστημάτων

Σύμφωνα με τους Gu et al.[2], τα federated query systems μειώνουν έως και 52% το integration overhead συγκριτικά με παραδοσιακά ETL architectures. Παράλληλα, οι Bacco et al.[18] επισημαίνουν ότι η federated προσέγγιση επιτρέπει σημαντικά ταχύτερη προσαρμογή σε νέα πληροφοριακά συστήματα χωρίς ανάγκη ανασχεδιασμού της συνολικής αρχιτεκτονικής. Ωστόσο, η ομοσπονδιακή πρόσβαση συνοδεύεται και από σημαντικούς περιορισμούς.

Πίνακας 2.8: Περιορισμοί της ομοσπονδιακής πρόσβασης σε δεδομένα

Περιορισμοί	Επιπτώσεις
Network latency	Καθυστέρηση distributed queries
Semantic inconsistencies	Προβλήματα interoperability
Query optimization complexity	Αυξημένο planning overhead
Security orchestration	Πολύπλοκος έλεγχος πρόσβασης
Distributed debugging	Δυσκολία troubleshooting

Οι Song et al.[26] επισημαίνουν ότι η query optimization αποτελεί το σημαντικότερο πρόβλημα των federated environments, καθώς τα distributed execution plans επηρεάζονται έντονα από καθυστέρηση δικτύου, διαθεσιμότητα πηγών και ετερογενείς δυνατότητες εκτέλεσης. Τα ακαδημαϊκά ιδρύματα αποτελούν ιδανικό πεδίο εφαρμογής ομοσπονδιακών αρχιτεκτονικών, λόγω της έντονης πληροφοριακής ετερογένειας που διαθέτουν. Τα πανεπιστήμια χρησιμοποιούν πολλαπλά ανεξάρτητα πληροφοριακά συστήματα για τη διαχείριση φοιτητών, μαθημάτων, εξεταστικών διαδικασιών, ερευνητικών δεδομένων και διοικητικών υπηρεσιών.

Πίνακας 2.9: Πηγές δεδομένων σε πανεπιστημιακό περιβάλλον ομοσπονδιακή πρόσβασης

Πηγή δεδομένων	Τύπος δεδομένων	Χρήση
Student Information Systems	Academic records	Student analytics
LMS πλατφόρμες	Learning activities	Learning analytics
HR Systems	Faculty δεδομένα	Administrative reporting
Research repositories	Scientific datasets	Research analytics
Evaluation platforms	Student feedback	Performance analysis

Μέσω federated query engines, τα πανεπιστήμια μπορούν να εκτελούν σύνθετα αναλυτικά ερωτήματα χωρίς φυσική συγκέντρωση δεδομένων[20]. Για παράδειγμα, είναι δυνατό να πραγματοποιηθεί συσχέτιση βαθμολογιών φοιτητών με δεδομένα συμμετοχής σε e-learning πλατφόρμες ή με στατιστικά αξιολόγησης μαθημάτων σε πραγματικό χρόνο. Σύμφωνα με τους Miyamoto και Kasuga[27], τα ομοσπονδιακά πανεπιστημιακά οικοσυστήματα επιτυγχάνουν έως και 41% ταχύτερη παραγωγή αναλυτικών αναφορών συγκριτικά με centralized architectures, ενώ παράλληλα μειώνουν σημαντικά το λειτουργικό κόστος συντήρησης.

2.5 Federated Query Engines

Η ραγδαία εξάπλωση των κατανεμημένων πληροφοριακών οικοσυστημάτων και η εκθετική αύξηση των ετερογενών δεδομένων οδήγησε στην ανάπτυξη νέων μηχανισμών επεξεργασίας ερωτημάτων μεγάλης κλίμακας, γνωστών ως Federated Query Engines. Οι μηχανισμοί αυτοί αποτελούν τον βασικό τεχνολογικό πυρήνα των σύγχρονων Data Spaces, καθώς επιτρέπουν την εκτέλεση ενοποιημένων αναλυτικών ερωτημάτων πάνω σε πολλαπλές κατανεμημένες πηγές δεδομένων χωρίς να απαιτείται φυσική μεταφορά των συνόλων δεδομένων σε centralized repositories[1], [2]. Η σημασία των federated engines αυξήθηκε δραματικά τα τελευταία χρόνια λόγω της μετάβασης των οργανισμών προς hybrid cloud περιβάλλοντα, distributed storage architectures και workloads αναλυτικής επεξεργασίας σε πραγματικό χρόνο, τα οποία απαιτούν άμεση πρόσβαση σε δεδομένα που βρίσκονται διασκορπισμένα σε διαφορετικές πληροφοριακές υποδομές[3].

Τα παραδοσιακά μοντέλα επεξεργασίας δεδομένων βασίζονταν κυρίως σε centralized warehouses και ETL pipelines. Ωστόσο, η προσέγγιση αυτή παρουσιάζει σημαντικά μειονεκτήματα σε περιβάλλοντα υψηλής πληροφοριακής ετερογένειας, καθώς η συνεχής αντιγραφή και συγχρονισμός δεδομένων δημιουργούν αυξημένη επιβάρυνση αποθήκευσης, υψηλό λειτουργικό κόστος και σημαντικές καθυστερήσεις ενημέρωσης[8]. Σύμφωνα με τους Gu et al.[2], τα enterprise ETL περιβάλλοντα καταναλώνουν έως και 65% του συνολικού χρόνου μηχανικής δεδομένων σε διαδικασίες data preparation και synchronization. Επιπλέον, οι batch integration διαδικασίες δυσκολεύονται να

υποστηρίζουν αναλυτικές απαιτήσεις πραγματικού χρόνου, ιδιαίτερα όταν τα δεδομένα μεταβάλλονται συνεχώς.

Μέσα σε αυτό το πλαίσιο, οι Federated Query Engines επιχειρούν να επιλύσουν το πρόβλημα της distributed data access μέσω virtualized query execution. Η βασική φιλοσοφία λειτουργίας τους βασίζεται στη λογική “query over data in place”, σύμφωνα με την οποία τα δεδομένα παραμένουν στις αρχικές πηγές αποθήκευσης και το query engine αναλαμβάνει δυναμικά την ανάλυση, τη βελτιστοποίηση και την εκτέλεση distributed subqueries[10]. Η προσέγγιση αυτή μειώνει σημαντικά το data duplication και επιτρέπει αξιοποίηση πληροφοριών σχεδόν σε πραγματικό χρόνο από ετερογενή οικοσυστήματα.

Η ανάπτυξη federated engines συνδέεται επίσης με τη μετάβαση προς cloud-native και serverless πληροφοριακές υποδομές. Σύμφωνα με τους Bian et al.[10], τα σύγχρονα serverless federated query systems επιτυγχάνουν έως και 38% χαμηλότερο execution cost μέσω elastic resource orchestration και dynamic workload balancing. Αντίστοιχα, οι Bodner et al.[28] καταγράφουν βελτίωση μεγαλύτερη του 44% στο resource utilization efficiency μέσω distributed adaptive execution μοντέλων. Οι εξελίξεις αυτές καθιστούν τα ομοσπονδιακά συστήματα ιδανικά για οργανισμούς με μεταβαλλόμενο αναλυτικό φορτίο εργασίας και υψηλές απαιτήσεις επεκτασιμότητας. Ένα Federated Query Engine αποτελεί distributed software layer το οποίο επιτρέπει την εκτέλεση ενοποιημένων queries πάνω σε πολλαπλές καταναμημένες πηγές δεδομένων μέσω ενιαίου execution framework[14]. Ο χρήστης αλληλεπιδρά με το σύστημα σαν να πρόκειται για μία ενιαία βάση δεδομένων, ενώ στην πραγματικότητα το query engine αναλαμβάνει τη διάσπαση του ερωτήματος σε πολλαπλά subqueries τα οποία εκτελούνται στις επιμέρους πληροφοριακές πηγές[15]. Η βασική διαδικασία λειτουργίας ενός federated engine περιλαμβάνει διαδοχικά στάδια query orchestration:

Πίνακας 2.10: Στάδια λειτουργίας ενός Federated Query Engine

Στάδιο	Περιγραφή
Query Parsing	Ανάλυση SQL query
Logical Planning	Δημιουργία logical execution tree
Source Resolution	Αντιστοίχιση data sources
Query Optimization	Βελτιστοποίηση execution plan
Distributed Execution	Εκτέλεση subqueries
Result Aggregation	Συνένωση αποτελεσμάτων

Κατά το στάδιο Query Parsing, το σύστημα αναλύει τη συντακτική δομή του SQL query και δημιουργεί internal representation του execution plan[16]. Στη συνέχεια πραγματοποιείται source resolution, όπου εντοπίζονται οι πηγές δεδομένων που απαιτούνται για την εκτέλεση του query. Το query decomposition αποτελεί ιδιαίτερα κρίσιμο στάδιο, καθώς το αρχικό ερώτημα διασπάται σε distributed subqueries που εκτελούνται σε διαφορετικά πληροφοριακά συστήματα. Το Query Optimization αποτελεί ίσως το σημαντικότερο στάδιο λειτουργίας federated systems. Σύμφωνα με τους Song et al.[26], το optimization overhead επηρεάζει άμεσα το συνολικό latency των καταναμημένων αναλυτικών φορτίων εργασίας. Για τον λόγο αυτό χρησιμοποιούνται προηγμένες τεχνικές βελτιστοποίησης όπως predicate pushdown, join

reordering, adaptive execution planning και cost-based optimization. Οι Federated Query Engines χρησιμοποιούν connectors για τη διασύνδεση με ετερογενείς πληροφοριακές πηγές.

Πίνακας 2.11: Τύποι πηγών δεδομένων που υποστηρίζονται από Federated Query Engines

Τύπος πηγής	Παραδείγματα
Relational DBMS	PostgreSQL, MySQL
NoSQL databases	MongoDB, Cassandra
Data Lakes	Iceberg, Delta Lake
Cloud Storage	AWS S3, Azure Blob
Streaming Systems	Kafka

Η χρήση connectors επιτρέπει την ενοποιημένη πρόσβαση σε διαφορετικά formats και διαφορετικές αρχιτεκτονικές αποθήκευσης χωρίς ανάγκη data transformation[24]. Η αρχιτεκτονική federated query processing βασίζεται συνήθως σε distributed coordinator-worker μοντέλο, σχεδιασμένο ώστε να υποστηρίζει parallel execution μεγάλης κλίμακας[4].

Πίνακας 2.12: Βασικά συστατικά της αρχιτεκτονικής ενός Federated Query Engine

Συστατικό	Ρόλος
Coordinator Node	Query orchestration
Worker Nodes	Distributed execution
Connectors	Πρόσβαση σε data sources
Metadata Service	Schema management
Resource Manager	Workload balancing

Ο Coordinator Node αποτελεί το κεντρικό orchestration layer του συστήματος. Είναι υπεύθυνος για query parsing, planning, optimization και scheduling. Όταν λαμβάνεται ένα query, ο coordinator δημιουργεί distributed execution graph και κατανέμει tasks στους worker nodes[18]. Οι Worker Nodes αναλαμβάνουν την πραγματική εκτέλεση distributed operations, όπως scans, joins, aggregations και filtering. Η parallel processing λογική επιτρέπει οριζόντια επεκτασιμότητα και υποστήριξη αναλυτικών φορτίων εργασίας κλίμακας petabyte[19]. Σύμφωνα με τους Fareedi et al.[9], η σωστή semantic αντιστοίχιση μεταδεδομένων αποτελεί κρίσιμο παράγοντα επιτυχίας των ομοσπονδιακών οικοσυστημάτων. Η επικοινωνία μεταξύ coordinator και workers πραγματοποιείται συνήθως μέσω distributed messaging protocols υψηλής απόδοσης. Οι σύγχρονες αρχιτεκτονικές χρησιμοποιούν asynchronous execution pipelines ώστε να περιορίζεται το synchronization overhead και να αυξάνεται το throughput των queries[20].

Το query optimization αποτελεί τον σημαντικότερο μηχανισμό απόδοσης ενός federated query engine, καθώς η distributed φύση των workloads δημιουργεί υψηλή πολυπλοκότητα execution planning[25]. Κατά το Query Parsing στάδιο, το SQL query μετατρέπεται σε Abstract Syntax Tree (AST), το οποίο επιτρέπει στο σύστημα να αναλύσει joins, filters, aggregations και nested operations[22]. Στη συνέχεια πραγματοποιείται Logical Query Planning.

Πίνακας 2.13: Τεχνικές βελτιστοποίησης σε Federated Query Engines

Optimization Technique	Στόχος
Predicate Pushdown	Μείωση transferred data
Projection Pushdown	Μείωση unnecessary columns
Join Reordering	Faster joins
Parallel Execution	Επεκτασιμότητα
Adaptive Scheduling	Dynamic workload balancing

Το Predicate Pushdown θεωρείται μία από τις σημαντικότερες optimization τεχνικές. Μέσω αυτής, τα συνθήκες φιλτραρίσματος αποστέλλονται απευθείας στις πηγές δεδομένων ώστε να περιορίζεται ο όγκος μεταφερόμενων συνόλων δεδομένων[21]. Παράλληλα, τα σύγχρονα ομοσπονδιακά συστήματα χρησιμοποιούν cost-based optimization μοντέλα, τα οποία εκτιμούν καθυστέρηση δικτύου, διαθεσιμότητα πηγών και υπολογιστικό κόστος πριν δημιουργήσουν execution plans[6]. Σύμφωνα με τους Song et al.[26], τα AI-driven learned query optimizers μπορούν να βελτιώσουν έως και 27% τον χρόνο εκτέλεσης καταναμημένων αναλυτικών φορτίων εργασίας.

Η δυνατότητα ενοποίησης πολλαπλών ετερογενών πηγών δεδομένων αποτελεί τον σημαντικότερο λόγο ανάπτυξης federated query engines[7]. Οι οργανισμοί διαχειρίζονται ολοένα και περισσότερες πληροφορίες αποθηκευμένες σε διαφορετικά πληροφοριακά οικοσυστήματα, γεγονός που απαιτεί ενιαίο επίπεδο αναλυτικής πρόσβασης.

Πίνακας 2.14: Πηγές δεδομένων και αντίστοιχες τεχνολογίες σε Federated Query Engines

Πηγή δεδομένων	Τεχνολογία
Relational Databases	PostgreSQL, Oracle
NoSQL Platforms	MongoDB
Distributed File Systems	Hadoop HDFS
Data Lakes	Delta Lake
Streaming Platforms	Apache Kafka
Cloud Storage	Amazon S3

Η ενοποίηση ετερογενών πηγών απαιτεί semantic interoperability και schema mapping μηχανισμούς[23]. Οι διαφορετικές πλατφόρμες χρησιμοποιούν διαφορετικά μοντέλα μεταδεδομένων και διαφορετικές logical structures, γεγονός που δημιουργεί σημαντικά προβλήματα compatibility. Σύμφωνα με τους Masmoudi et al.[21], η semantic heterogeneity αποτελεί έναν από τους σημαντικότερους περιοριστικούς παράγοντες των ομοσπονδιακών οικοσυστημάτων. Για τον λόγο αυτό, πολλά σύγχρονα query engines ενσωματώνουν metadata virtualization και ontology-based integration frameworks. Η δυνατότητα cross-platform joins αποτελεί επίσης κρίσιμο χαρακτηριστικό federated systems. Ένα query engine μπορεί να εκτελεί joins μεταξύ PostgreSQL, MongoDB και Iceberg tables ταυτόχρονα, δημιουργώντας ενοποιημένα αναλυτικά αποτελέσματα χωρίς φυσική συγκέντρωση δεδομένων[18]. Τέλος, η ενοποίηση πολλαπλών πηγών δεδομένων συνδέεται άμεσα με τις σύγχρονες lakehouse αρχιτεκτονικές.

2.6 Επισκόπηση σχετικών τεχνολογιών

Η ανάγκη δημιουργίας τέτοιων τεχνολογιών προέκυψε από τη δραματική αύξηση της πληροφοριακής ετερογένειας που παρατηρείται σε σύγχρονους οργανισμούς. Σύμφωνα με μελέτες της IDC, περισσότερο από το 68% των επιχειρησιακών δεδομένων βρίσκεται σήμερα κατανεμημένο σε πολλαπλά πληροφοριακά οικοσυστήματα, ενώ πάνω από το 52% των οργανισμών χρησιμοποιεί ταυτόχρονα σχεσιακά DBMS, cloud object storage, NoSQL platforms και streaming infrastructures. Η κατάσταση αυτή καθιστά εξαιρετικά δύσκολη τη χρήση παραδοσιακών centralized architectures, καθώς η συνεχής μεταφορά και συγχρονισμός δεδομένων δημιουργεί αυξημένη λειτουργική επιβάρυνση, υψηλό βαθμό διπλής αποθήκευσης δεδομένων και σημαντικό δικτυακό κόστος.

Οι σύγχρονες ομοσπονδιακές τεχνολογίες επιχειρούν να αντιμετωπίσουν τα προβλήματα αυτά μέσω distributed execution frameworks και virtualized data access layers. Τα συστήματα αυτά δεν λειτουργούν ως παραδοσιακά DBMS, αλλά ως ενδιάμεσες distributed processing πλατφόρμες που αναλαμβάνουν query orchestration, metadata mapping, semantic integration και parallel execution. Μέσω connectors και catalogs μπορούν να συνδέονται ταυτόχρονα με δεκάδες διαφορετικές πηγές δεδομένων, επιτρέποντας στον χρήστη να εκτελεί ενοποιημένα ερωτήματα SQL σε ετερογενή πληροφοριακά οικοσυστήματα. Η ανάπτυξη federated query engines επιταχύνθηκε ιδιαίτερα μετά το 2018, όταν η παγκόσμια αγορά cloud analytics ξεπέρασε τα 23 δισεκατομμύρια δολάρια και τα enterprise workloads άρχισαν να μετακινούνται μαζικά προς hybrid και multi-cloud infrastructures. Σύμφωνα με στοιχεία της Gartner, το 74% των οργανισμών που χρησιμοποιούν distributed analytics architectures αντιμετωπίζει σημαντικά προβλήματα interoperability μεταξύ διαφορετικών data platforms, ενώ το 61% θεωρεί το query latency ως τον σημαντικότερο περιοριστικό παράγοντα των legacy πληροφοριακών υποδομών. Για τον λόγο αυτό, η ανάπτυξη τεχνολογιών όπως Trino, Presto, Apache Drill και Denodo θεωρείται ιδιαίτερα κρίσιμη για τη μετάβαση προς επεκτάσιμα federated οικοσυστήματα.

- **Trino**

Το Trino αποτελεί σήμερα μία από τις ισχυρότερες πλατφόρμες ανοικτού κώδικα federated query processing και χρησιμοποιείται ευρέως σε περιβάλλοντα κατανεμημένης ανάλυσης δεδομένων μεγάλης κλίμακας. Η αρχιτεκτονική του βασίζεται σε massively parallel processing μοντέλο, όπου η εκτέλεση των queries πραγματοποιείται μέσω coordinator-worker topology. Ο coordinator node αναλαμβάνει query parsing, optimization και scheduling, ενώ οι worker nodes εκτελούν distributed tasks παράλληλα. Η συγκεκριμένη προσέγγιση επιτρέπει οριζόντια επεκτασιμότητα μέσω προσθήκης νέων worker nodes χωρίς ανάγκη αλλαγής της συνολικής αρχιτεκτονικής.

Το Trino υποστηρίζει περισσότερους από 50 connectors, γεγονός που του επιτρέπει να διασυνδέεται με σχεσιακά DBMS, NoSQL συστήματα, cloud storage platforms και data lake υποδομές. Ειδικότερα, μπορεί να συνδεθεί με PostgreSQL, MySQL, Oracle, SQL Server, MongoDB, Cassandra, Kafka, Hive, Iceberg, Delta Lake και Amazon S3. Η δυνατότητα αυτή επιτρέπει cross-platform query execution σε διαφορετικές τεχνολογίες αποθήκευσης χωρίς προηγούμενη φυσική ενοποίηση των δεδομένων. Σε επίπεδο απόδοσης, το Trino χρησιμοποιεί προηγμένες τεχνικές βελτιστοποίησης όπως predicate pushdown, projection pushdown, dynamic filtering και adaptive query execution. Το predicate pushdown επιτρέπει τη μεταφορά συνθήκες φιλτραρίσματος απευθείας στις πηγές δεδομένων, μειώνοντας δραστικά τον όγκο μεταφερόμενων συνόλων δεδομένων. Σε κατανεμημένα αναλυτικά

φορτία εργασίας μεγάλης κλίμακας, η τεχνική αυτή μπορεί να μειώσει έως και 70% το network traffic συγκριτικά με naive execution models. Παράλληλα, το Trino χρησιμοποιεί cost-based optimization για τη δημιουργία execution plans, λαμβάνοντας υπόψη καθυστέρηση δικτύου, διαθέσιμους computational resources και cardinality estimations.

Η απόδοση του Trino σε αναλυτική επεξεργασία μεγάλης κλίμακας είναι ιδιαίτερα υψηλή. Σε benchmark δοκιμές TPC-DS, clusters 20 worker nodes μπορούν να επεξεργαστούν σύνολα δεδομένων άνω των 100 terabytes με καθυστέρηση μικρότερη των 12 δευτερολέπτων για σύνθετα distributed joins. Επιπλέον, η δυνατότητα execution σε cloud-native environments επιτρέπει elastic resource scaling, μειώνοντας σημαντικά το λειτουργικό κόστος σε περιβάλλοντα μεταβαλλόμενου workload. Ωστόσο, το Trino παρουσιάζει και ορισμένους περιορισμούς. Η απόδοση distributed joins επηρεάζεται σημαντικά από καθυστέρηση δικτύου και από τη γεωγραφική κατανομή των πηγών δεδομένων. Εάν ένα query απαιτεί μεταφορά εκατομμυρίων γραμμών μεταξύ διαφορετικών data sources, τότε το execution overhead αυξάνεται σημαντικά. Παράλληλα, η σωστή λειτουργία του απαιτεί προσεκτικό tuning σε επίπεδο memory allocation, partitioning και connector configuration.

- **Presto**

Το Presto αποτελεί έναν από τους πρώτους massively parallel distributed SQL query engines που σχεδιάστηκαν για αναλυτική επεξεργασία σε πραγματικό χρόνο σε κατανεμημένα πληροφοριακά οικοσυστήματα. Αναπτύχθηκε αρχικά για αναλυτικά φορτία εργασίας μεγάλης κλίμακας και σχεδιάστηκε ώστε να επιτρέπει interactive SQL querying πάνω σε σύνολα δεδομένων κλίμακας petabyte. Η βασική φιλοσοφία λειτουργίας του είναι παρόμοια με αυτή του Trino, καθώς χρησιμοποιεί coordinator-worker architecture και distributed execution model. Η κύρια αξία του Presto βρίσκεται στην ικανότητά του να εκτελεί χαμηλής καθυστέρησης interactive analytics σε ετερογενή πληροφοριακά περιβάλλοντα. Οι worker nodes επεξεργάζονται παράλληλα διαφορετικά partitions δεδομένων, ενώ το σύστημα υποστηρίζει in-memory execution για μείωση του disk I/O overhead. Σε αναλυτικά φορτία εργασίας μεγάλης κλίμακας, το Presto μπορεί να επεξεργάζεται δισεκατομμύρια γραμμές δεδομένων με καθυστέρηση μικρότερη των λίγων δευτερολέπτων.

Ωστόσο, η εξέλιξή του τα τελευταία χρόνια θεωρείται πιο περιορισμένη συγκριτικά με το Trino, ιδιαίτερα ως προς σύγχρονα lakehouse integrations και προηγμένες τεχνικές βελτιστοποίησης. Για τον λόγο αυτό, πολλοί οργανισμοί στρέφονται πλέον προς Trino deployments για νέα federated analytics infrastructures. Παρά τους περιορισμούς αυτούς, το Presto παραμένει ιδιαίτερα αποδοτικό σε μεγάλα αναλυτικά φορτία εργασίας. Σε cluster περιβάλλοντα άνω των 100 nodes, μπορεί να υποστηρίξει throughput μεγαλύτερο των 250.000 rows/sec σε distributed scan operations. Επιπλέον, η αρχιτεκτονική του επιτρέπει υψηλό βαθμό concurrency, υποστηρίζοντας ταυτόχρονα εκατοντάδες αναλυτικά ερωτήματα.

- **Apache Drill**

Το Apache Drill σχεδιάστηκε με στόχο την υποστήριξη schema-free query execution σε ημιδομημένα και ένθετα σύνολα δεδομένων. Σε αντίθεση με Trino και Presto, τα οποία δίνουν έμφαση κυρίως σε federated SQL processing, το Drill εστιάζει στην ευέλικτη πρόσβαση σε JSON, Avro, Parquet και nested πληροφοριακές δομές χωρίς ανάγκη προκαθορισμένου relational schema. Η δυνατότητα dynamic schema discovery επιτρέπει στο Drill να αναλύει αυτόματα nested structures και να παρέχει SQL

querying σε ημιδομημένα δεδομένα. Το χαρακτηριστικό αυτό είναι ιδιαίτερα χρήσιμο σε περιβάλλοντα όπου τα δεδομένα μεταβάλλονται συνεχώς ή παράγονται από IoT streams, αρχεία καταγραφής εφαρμογών και ερευνητικά σύνολα δεδομένων.

Το Drill χρησιμοποιεί distributed execution architecture και vectorized query processing για τη βελτιστοποίηση αναλυτικών φορτίων εργασίας. Μέσω columnar execution μπορεί να επεξεργάζεται μεγάλα σύνολα δεδομένων Parquet με σημαντικά μειωμένο CPU overhead συγκριτικά με row-based execution μοντέλα. Σε benchmark workloads JSON processing, το Drill εμφανίζει έως και 35% ταχύτερη επεξεργασία nested δεδομένων συγκριτικά με παραδοσιακά relational execution engines. Ωστόσο, η υποστήριξη federated joins και enterprise governance capabilities θεωρείται πιο περιορισμένη συγκριτικά με Trino και Denodo. Για τον λόγο αυτό, το Drill χρησιμοποιείται συχνότερα ως specialized semi-structured analytics engine και όχι ως πλήρες federated orchestration platform.

Denodo ή άλλες λύσεις data virtualization

Το Denodo ανήκει στην κατηγορία enterprise data virtualization platforms και διαφοροποιείται σημαντικά από τα open-source federated query engines, καθώς δίνει ιδιαίτερη έμφαση σε governance, metadata management, semantic integration και policy-driven access control. Η βασική φιλοσοφία του δεν περιορίζεται αποκλειστικά στην εκτέλεση distributed SQL ερωτημάτων, αλλά επεκτείνεται στη δημιουργία ολοκληρωμένου virtualized data layer για enterprise πληροφοριακά οικοσυστήματα.

Το Denodo υποστηρίζει advanced metadata cataloging, data lineage tracking, role-based access control και integrated caching mechanisms. Οι δυνατότητες αυτές είναι ιδιαίτερα σημαντικές σε οργανισμούς με αυξημένες απαιτήσεις compliance και privacy governance. Σε πανεπιστημιακά περιβάλλοντα, όπου διαφορετικές ομάδες χρηστών διαθέτουν διαφορετικά επίπεδα πρόσβασης, οι enterprise virtualization πλατφόρμες μπορούν να προσφέρουν υψηλότερο επίπεδο ελέγχου συγκριτικά με query engines ανοικτού κώδικα. Ωστόσο, το μεγαλύτερο μειονέκτημα enterprise virtualization λύσεων αφορά το οικονομικό κόστος. Τα μοντέλα αδειοδότησης τέτοιων πλατφορμών αυξάνουν σημαντικά τις συνολικές λειτουργικές δαπάνες, γεγονός που περιορίζει τη χρήση τους σε οργανισμούς μικρότερης κλίμακας ή σε ακαδημαϊκά περιβάλλοντα περιορισμένου προϋπολογισμού.

Η συγκριτική αξιολόγηση των τεχνολογιών federated query processing δείχνει ότι κάθε εργαλείο παρουσιάζει διαφορετικά πλεονεκτήματα ανάλογα με τις απαιτήσεις του πληροφοριακού οικοσυστήματος. Το Trino εμφανίζει ιδιαίτερα υψηλή απόδοση σε καταναεμημένη ανάλυση δεδομένων μεγάλης κλίμακας και federated SQL execution, αποτελώντας εξαιρετικά ισχυρή επιλογή για περιβάλλοντα Data Space με ετερογενείς σχεσιακές και cloud-native υποδομές. Το Presto παραμένει ιδιαίτερα αποδοτικό σε αναλυτικά φορτία εργασίας με μαζικό παραλληλισμό, αν και παρουσιάζει μικρότερη δυναμική εξέλιξης τα τελευταία χρόνια. Αντίθετα, το Apache Drill εξειδικεύεται κυρίως σε schema-free analytics και semi-structured data processing, καθιστώντας το κατάλληλο για περιβάλλοντα όπου κυριαρχούν JSON, nested και σύνολα δεδομένων βασισμένα σε αρχεία καταγραφής. Από την άλλη πλευρά, enterprise virtualization platforms όπως το Denodo δίνουν έμφαση σε governance, metadata orchestration και security-aware virtualization, προσφέροντας σημαντικά ισχυρότερο semantic layer αλλά με υψηλότερο οικονομικό κόστος. Για την παρούσα εργασία, η επιλογή του Trino θεωρείται ιδιαίτερα κατάλληλη, καθώς συνδυάζει χαρακτήρα ανοικτού κώδικα, υψηλή επεκτασιμότητα, μεγάλο connector ecosystem και ισχυρές δυνατότητες federated query execution. Επιπλέον, η υποστήριξη σύγχρονων lakehouse formats, cloud-native deployment και προηγμένων

distributed optimization τεχνικών καθιστά το Trino ιδιαίτερα συμβατό με τις απαιτήσεις ενός πανεπιστημιακού περιβάλλοντος Data Space μεγάλης πληροφοριακής πολυπλοκότητας.

2.7 Ανασκόπηση σχετικών ερευνητικών εργασιών

Η ερευνητική δραστηριότητα γύρω από τα Data Spaces, τα federated query systems και τις κατανεμημένες πληροφοριακές αρχιτεκτονικές έχει αυξηθεί σημαντικά τα τελευταία χρόνια, κυρίως λόγω της ανάγκης διαχείρισης ολοένα και μεγαλύτερων όγκων ετερογενών δεδομένων. Οι πρώτες ερευνητικές προσεγγίσεις επικεντρώνονταν κυρίως σε παραδοσιακά data integration models και ETL-based architectures, όπου η ενοποίηση δεδομένων πραγματοποιούνταν μέσω κεντρικών αποθετηρίων. Ωστόσο, η εκθετική αύξηση του όγκου δεδομένων και η μετάβαση προς cloud-native πληροφοριακά οικοσυστήματα οδήγησαν σταδιακά στην ανάπτυξη federated και decentralized μοντέλων διαχείρισης πληροφορίας.

Σημαντικό μέρος της βιβλιογραφίας εστιάζει στη σύγκριση centralized και federated πληροφοριακών αρχιτεκτονικών. Οι Gu et al.[2] ανέλυσαν την απόδοση distributed query engines σε περιβάλλοντα ετερογενών δεδομένων και κατέληξαν στο συμπέρασμα ότι τα ομοσπονδιακά συστήματα μπορούν να μειώσουν έως και 52% το integration overhead συγκριτικά με ETL-based data warehouse υποδομές. Παράλληλα, οι Jarke και Quix[1] διερεύνησαν τη σημασία των semantic interoperability frameworks στα Data Spaces, υποστηρίζοντας ότι η δυνατότητα logical integration χωρίς φυσική μεταφορά δεδομένων αποτελεί βασική προϋπόθεση για επεκτάσιμα distributed οικοσυστήματα. Άλλες ερευνητικές εργασίες επικεντρώνονται στην απόδοση federated query optimization τεχνικών. Οι Song et al.[26] μελέτησαν learned query optimization μοντέλα σε distributed SQL engines και διαπίστωσαν ότι η χρήση AI-driven execution planning μπορεί να μειώσει έως και 27% το execution latency σύνθετων αναλυτικών ερωτημάτων. Αντίστοιχα, οι Fareedi et al.[9] εξέτασαν το πρόβλημα semantic heterogeneity και υποστήριξαν ότι τα metadata virtualization frameworks αποτελούν κρίσιμο παράγοντα επιτυχίας ομοσπονδιακών πληροφοριακών αρχιτεκτονικών.

Σημαντικό ερευνητικό ενδιαφέρον παρουσιάζει επίσης η σύνδεση Data Spaces με cloud-native και lakehouse architectures. Οι Chaudhari και Charate[11] ανέλυσαν τη σύγκλιση lakehouse τεχνολογιών και federated query engines, επισημαίνοντας ότι τα σύγχρονα αναλυτικά οικοσυστήματα μεταβαίνουν ολοένα και περισσότερο σε hybrid architectures όπου συνδυάζονται object storage, distributed metadata catalogs και federated SQL execution layers. Οι Miyamoto και Kasuga[27] ανέπτυξαν federated university analytics framework βασισμένο σε distributed query execution και κατέγραψαν βελτίωση μεγαλύτερη του 40% στην ταχύτητα παραγωγής αναλυτικών αναφορών συγκριτικά με centralized πληροφοριακά μοντέλα. Αντίστοιχα, οι Frade et al.[29] ανέλυσαν ζητήματα privacy-aware federated analytics σε πανεπιστημιακά οικοσυστήματα, επισημαίνοντας ότι η decentralized πρόσβαση σε δεδομένα αυξάνει σημαντικά την πολυπλοκότητα authorization orchestration και governance management.

Κεφάλαιο 3: Τεχνολογίες, εργαλεία και αρχιτεκτονικές υλοποίησης

3.1 Απαιτήσεις ενός περιβάλλοντος Data Space

3.1.1 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις προσδιορίζουν το ελάχιστο σύνολο δυνατοτήτων που πρέπει να υποστηρίζει το σύστημα ώστε να επιτυγχάνεται ενοποιημένη πρόσβαση σε κατανεμημένες και ετερογενείς πηγές δεδομένων χωρίς φυσική συγκέντρωσή τους. Καταρχάς, απαιτείται η ύπαρξη ενός ενιαίου λογικού επιπέδου πρόσβασης, το οποίο επιτρέπει την εκτέλεση ομοσπονδιακών ερωτημάτων (federated queries) σε πολλαπλά πληροφοριακά συστήματα, διατηρώντας ταυτόχρονα την αυτονομία των επιμέρους πηγών[1], [2]. Το σύστημα οφείλει να υποστηρίζει δυναμική ανάλυση και αποσύνθεση ερωτημάτων σε υπο-ερωτήματα που εκτελούνται τοπικά στις πηγές δεδομένων, με στόχο τη μείωση του όγκου μεταφοράς δεδομένων και τη βελτιστοποίηση της απόδοσης μέσω τεχνικών όπως το predicate pushdown και το cost-based optimization[2], [3]. Παράλληλα, η διασύνδεση ετερογενών συστημάτων προϋποθέτει τη χρήση connectors και catalogs που επιτρέπουν τη διαφανή πρόσβαση σε διαφορετικά DBMS, NoSQL πλατφόρμες και cloud storage υποδομές, διασφαλίζοντας διαλειτουργικότητα σε επίπεδο πρωτοκόλλων και σχημάτων[5], [24]. Εξίσου σημαντική είναι η υποστήριξη μηχανισμών σημασιολογικής αντιστοίχισης, καθώς η ύπαρξη διαφορετικών μοντέλων μεταδεδομένων και συμβάσεων ονοματοδοσίας απαιτεί την ανάπτυξη ενδιάμεσου semantic layer για τη σωστή ερμηνεία και ενοποίηση της πληροφορίας[6], [21]. Ένα πλήρως λειτουργικό Data Space πρέπει να παρέχει δυνατότητες επεξεργασίας δεδομένων σε πραγματικό ή σχεδόν πραγματικό χρόνο, αποφεύγοντας batch διαδικασίες συγχρονισμού που χαρακτηρίζουν τα παραδοσιακά ETL μοντέλα[2], [4]. Η υποστήριξη distributed query execution και parallel processing αποτελεί κρίσιμο χαρακτηριστικό για την επεξεργασία μεγάλου όγκου δεδομένων, ιδίως σε περιβάλλοντα υψηλής μεταβλητότητας[10], [28]. Τέλος, η λειτουργική επάρκεια ενός τέτοιου συστήματος συμπληρώνεται από τη δυνατότητα εκτέλεσης σύνθετων αναλυτικών λειτουργιών, όπως joins μεταξύ ετερογενών πηγών, aggregations και multi-source correlations, που επιτρέπουν την παραγωγή σύνθετης γνώσης από κατακερματισμένα datasets[2], [27].

Η υλοποίηση ενός Data Space προϋποθέτει την ενσωμάτωση λειτουργιών που σχετίζονται με τη διαχείριση πρόσβασης, την ασφάλεια και τη διακυβέρνηση δεδομένων, οι οποίες αποτελούν αναπόσπαστο μέρος των λειτουργικών απαιτήσεων. Αρχικά, είναι αναγκαία η ύπαρξη μηχανισμών authentication και authorization που επιτρέπουν τον έλεγχο πρόσβασης σε διαφορετικά επίπεδα χρηστών, διασφαλίζοντας ότι η πληροφορία χρησιμοποιείται σύμφωνα με προκαθορισμένες πολιτικές[24], [25]. Παράλληλα, η έννοια της data sovereignty καθιστά απαραίτητη τη διατήρηση ελέγχου των δεδομένων από τους αρχικούς ιδιοκτήτες τους, γεγονός που διαφοροποιεί τα Data Spaces από κεντριοποιημένα μοντέλα αποθήκευσης[4], [18]. Επιπλέον, η υποστήριξη data lineage και metadata tracking είναι κρίσιμη για την ιχνηλασιμότητα των δεδομένων και την κατανόηση της προέλευσης και των μετασχηματισμών τους, ιδιαίτερα σε περιβάλλοντα όπου εκτελούνται σύνθετα federated queries[9], [24]. Η ανάγκη διασφάλισης υψηλής απόδοσης επιβάλλει την υιοθέτηση μηχανισμών workload management και resource allocation, οι οποίοι επιτρέπουν την αποτελεσματική κατανομή υπολογιστικών πόρων σε κατανεμημένα περιβάλλοντα[10], [28]. Επιπλέον, η ενσωμάτωση caching μηχανισμών μπορεί να συμβάλει στη μείωση του latency σε επαναλαμβανόμενα queries,

βελτιώνοντας τη συνολική εμπειρία χρήσης[14]. Τέλος, η υποστήριξη επεκτασιμότητας αποτελεί θεμελιώδη απαίτηση, καθώς το σύστημα πρέπει να μπορεί να προσαρμόζεται δυναμικά σε αυξανόμενο όγκο δεδομένων και νέες πηγές χωρίς ανασχεδιασμό της αρχιτεκτονικής[11], [12].

3.1.2 Μη λειτουργικές απαιτήσεις

Η απόδοση αποτελεί θεμελιώδη παράγοντα, καθώς τα federated query engines καλούνται να εκτελούν σύνθετα ερωτήματα σε πολλαπλές πηγές με ελάχιστο latency, λαμβάνοντας υπόψη περιορισμούς όπως το network overhead και η ετερογένεια των execution environments[2], [10]. Στο ίδιο πλαίσιο, η δυνατότητα οριζόντιας επεκτασιμότητας καθίσταται απαραίτητη, επιτρέποντας την προσαρμογή του συστήματος σε αυξανόμενο όγκο δεδομένων και σε μεγαλύτερο αριθμό συνδεδεμένων πηγών χωρίς υποβάθμιση της απόδοσης[11], [12]. Η αξιοπιστία και η ανθεκτικότητα συνιστούν βασικές απαιτήσεις, δεδομένου ότι η κατανομή των Data Spaces αυξάνει την πιθανότητα αποτυχιών σε επιμέρους κόμβους ή υπηρεσίες, γεγονός που απαιτεί μηχανισμούς ανάκαμψης και διαχείρισης σφαλμάτων[10], [28]. Επιπλέον, η διαθεσιμότητα του συστήματος πρέπει να διασφαλίζεται μέσω κατάλληλων στρατηγικών load balancing και redundancy, ώστε να υποστηρίζεται συνεχής πρόσβαση ακόμη και υπό συνθήκες υψηλού φόρτου εργασίας[28]. Εξίσου σημαντική είναι η αποδοτικότητα στη χρήση πόρων, καθώς τα σύγχρονα ομοσπονδιακά περιβάλλοντα λειτουργούν συχνά σε cloud ή serverless υποδομές, όπου η βελτιστοποίηση κόστους και κατανάλωσης υπολογιστικών πόρων αποτελεί κρίσιμο παράγοντα σχεδίασης[10], [15].

Πέραν των ζητημάτων απόδοσης, ιδιαίτερη έμφαση δίνεται στη διαλειτουργικότητα και την ευελιξία του συστήματος, χαρακτηριστικά που επιτρέπουν τη διασύνδεση ετερογενών πληροφοριακών υποδομών με διαφορετικά data models και τεχνολογικές στοίβες. Η δυνατότητα ενσωμάτωσης νέων πηγών δεδομένων χωρίς σημαντικές αλλαγές στην αρχιτεκτονική αποτελεί βασική μη λειτουργική απαίτηση, καθώς τα σύγχρονα οικοσυστήματα εξελίσσονται δυναμικά και απαιτούν προσαρμοστικότητα[7], [15]. Παράλληλα, η υποστήριξη πολλαπλών τύπων δεδομένων, όπως δομημένων, ημιδομημένων και μη δομημένων μορφών, ενισχύει την ευελιξία του Data Space και επιτρέπει την αξιοποίηση ποικίλων πληροφοριακών πόρων[6], [15]. Επιπροσθέτως, η σημασιολογική συνοχή αποτελεί κρίσιμο παράγοντα, καθώς η ενοποίηση δεδομένων από διαφορετικές πηγές απαιτεί αξιόπιστους μηχανισμούς αντιστοίχισης και ερμηνείας metadata ώστε να αποφεύγονται ασυμβατότητες και λανθασμένα αποτελέσματα[21]. Η ποιότητα των δεδομένων επηρεάζει άμεσα την ακρίβεια των αναλυτικών αποτελεσμάτων, καθιστώντας αναγκαία την ενσωμάτωση διαδικασιών ελέγχου εγκυρότητας και συνέπειας[22]. Επιπλέον, η υποστήριξη χαμηλού latency σε περιβάλλοντα αναλυτικής επεξεργασίας σε πραγματικό χρόνο αποτελεί καθοριστικό παράγοντα για εφαρμογές που απαιτούν άμεση λήψη αποφάσεων, όπως τα σύγχρονα πανεπιστημιακά πληροφοριακά συστήματα[11]. Η επεκτασιμότητα σε επίπεδο αρχιτεκτονικής, σε συνδυασμό με την ικανότητα διαχείρισης αυξανόμενης πολυπλοκότητας, επιτρέπει στο Data Space να λειτουργεί αποδοτικά σε περιβάλλοντα μεγάλης κλίμακας, διατηρώντας ταυτόχρονα υψηλό επίπεδο ευελιξίας και προσαρμοστικότητας[12], [30].

3.1.3 Απαιτήσεις ασφάλειας και πρόσβασης

Η υλοποίηση μηχανισμών αυθεντικοποίησης αποτελεί βασική προϋπόθεση για την αξιόπιστη ταυτοποίηση χρηστών και υπηρεσιών που αλληλεπιδρούν με το σύστημα, ιδιαίτερα σε περιβάλλοντα όπου πολλαπλοί οργανισμοί συμμετέχουν στη διαμοίραση δεδομένων[24], [25]. Παράλληλα, η εφαρμογή λεπτομερούς ελέγχου εξουσιοδότησης επιτρέπει τον καθορισμό διαφοροποιημένων επιπέδων πρόσβασης, διασφαλίζοντας ότι κάθε χρήστης μπορεί να αλληλεπιδρά μόνο με τα δεδομένα που αντιστοιχούν στον ρόλο και στα δικαιώματά του[1], [4]. Η έννοια της data sovereignty καθίσταται

καθοριστική, καθώς οι πάροχοι δεδομένων διατηρούν πλήρη έλεγχο επί της πληροφορίας τους, επιβάλλοντας πολιτικές χρήσης και περιορισμούς πρόσβασης χωρίς να απαιτείται φυσική μεταφορά των δεδομένων σε κεντρικά αποθετήρια[4], [18]. Συναφώς, η ασφάλεια σε επίπεδο επικοινωνίας μεταξύ των επιμέρους συστημάτων απαιτεί την υιοθέτηση πρωτοκόλλων κρυπτογράφησης και ασφαλών καναλιών μετάδοσης, ώστε να αποτρέπεται η μη εξουσιοδοτημένη πρόσβαση κατά τη μεταφορά των δεδομένων[23], [29]. Επιπροσθέτως, η κατανεμημένη εκτέλεση ερωτημάτων ενισχύει την πολυπλοκότητα της διαχείρισης ασφάλειας, καθώς τα δεδομένα επεξεργάζονται σε πολλαπλά σημεία, γεγονός που απαιτεί συνεκτικό συντονισμό πολιτικών ασφάλειας σε διαφορετικά συστήματα[2], [21].

Η ανάγκη διασφάλισης ιδιωτικότητας και συμμόρφωσης με κανονιστικά πλαίσια ενισχύει περαιτέρω τη σημασία προηγμένων μηχανισμών προστασίας δεδομένων στα Data Spaces. Η ενσωμάτωση τεχνικών *privacy-aware processing*, όπως ο έλεγχος ανωνυμοποίησης και η διαχείριση ευαίσθητων πληροφοριών, καθίσταται απαραίτητη σε περιβάλλοντα όπου επεξεργάζονται προσωπικά ή ερευνητικά δεδομένα υψηλής ευαισθησίας[17], [22]. Επιπλέον, η υποστήριξη *data lineage* και *auditing* μηχανισμών επιτρέπει την πλήρη ιχνηλασιμότητα της χρήσης των δεδομένων, ενισχύοντας τη διαφάνεια και τη δυνατότητα ελέγχου σε επίπεδο διακυβέρνησης[9], [24]. Στο ίδιο πλαίσιο, η υιοθέτηση πολιτικών *access control* βασισμένων σε ρόλους ή χαρακτηριστικά (*role-based* ή *attribute-based access control*) επιτρέπει την ευέλικτη και δυναμική διαχείριση δικαιωμάτων σε περιβάλλοντα με αυξανόμενη πολυπλοκότητα[1], [25]. Η ανάπτυξη μηχανισμών ανίχνευσης ανωμαλιών και προστασίας από κυβερνοεπιθέσεις καθίσταται αναγκαία, καθώς η διασύνδεση πολλαπλών πληροφοριακών συστημάτων αυξάνει την επιφάνεια επιθέσεων και δημιουργεί νέους κινδύνους ασφάλειας[23], [29]. Σημαντική διάσταση αποτελεί επίσης η αξιοποίηση τεχνολογιών όπως το *federated learning* και τα *decentralized trust frameworks*, τα οποία επιτρέπουν την ανάλυση δεδομένων χωρίς αποκάλυψη ευαίσθητων πληροφοριών, ενισχύοντας την εμπιστοσύνη μεταξύ των συμμετεχόντων[19], [20].

Πίνακας 3.1: Μηχανισμοί ασφάλειας και ελέγχου πρόσβασης σε περιβάλλον Data Space

Μηχανισμός	Λειτουργικός ρόλος	Επίπεδο εφαρμογής	Τεχνολογική προσέγγιση
Authentication	Ταυτοποίηση χρηστών και υπηρεσιών	Federation layer	Identity providers, OAuth2, SSO
Authorization	Έλεγχος δικαιωμάτων πρόσβασης	Governance layer	RBAC, ABAC πολιτικές
Data Sovereignty	Διατήρηση ελέγχου από τον πάροχο	Data source layer	Policy enforcement, access contracts
Encryption	Προστασία δεδομένων κατά τη μεταφορά	Network layer	TLS, end-to-end encryption
Secure Query Execution	Ασφαλής εκτέλεση federated queries	Query processing layer	Query isolation, sandboxing
Data Lineage & Auditing	Ιχνηλασιμότητα χρήσης δεδομένων	Metadata layer	Logging, provenance tracking
Privacy-preserving techniques	Προστασία ευαίσθητων δεδομένων	Analytics layer	Anonymization, federated learning
Threat Detection	Ανίχνευση επιθέσεων και ανωμαλιών	System-wide	AI-based monitoring, intrusion detection

Πηγές:[1], [2], [4], [19], [21], [24], [25], [29]

3.2 Αρχιτεκτονικές προσεγγίσεις ενοποίησης δεδομένων

3.2.1 Κεντριοποιημένη αρχιτεκτονική

Τα δεδομένα εξάγονται από τις επιμέρους πηγές μέσω διαδικασιών Extract–Transform–Load (ETL), μετασχηματίζονται σε ομογενοποιημένη μορφή και αποθηκεύονται σε κεντρικά αποθετήρια, όπως data warehouses ή data lakes, προκειμένου να υποστηριχθεί η εκτέλεση αναλυτικών ερωτημάτων[2], [12]. Η συγκεκριμένη αρχιτεκτονική προσφέρει υψηλό επίπεδο ελέγχου και συνέπειας, καθώς η ύπαρξη ενιαίου schema επιτρέπει την εφαρμογή κοινών κανόνων διαχείρισης δεδομένων και την υλοποίηση προηγμένων αναλυτικών διαδικασιών χωρίς την ανάγκη δυναμικής σύνθεσης ερωτημάτων σε καταναμημένες πηγές[11], [15]. Παράλληλα, η κεντρική αποθήκευση διευκολύνει την εφαρμογή τεχνικών βελτιστοποίησης, όπως indexing και materialized views, ενισχύοντας την απόδοση σε επαναλαμβανόμενα workloads και μειώνοντας το latency κατά την εκτέλεση σύνθετων queries[14]. Ωστόσο, η εξάρτηση από ETL pipelines δημιουργεί σημαντικές λειτουργικές απαιτήσεις, καθώς η συνεχής αντιγραφή και συγχρονισμός δεδομένων συνεπάγονται αυξημένο λειτουργικό κόστος και καθυστερήσεις στην ενημέρωση των πληροφοριών[2], [7].

Παρά τα πλεονεκτήματα που σχετίζονται με την οργανωτική σαφήνεια και τη σταθερότητα της αρχιτεκτονικής, η κεντριοποιημένη προσέγγιση παρουσιάζει ουσιώδεις περιορισμούς σε περιβάλλοντα υψηλής ετερογένειας και δυναμικής μεταβολής δεδομένων. Καταρχάς, η ανάγκη φυσικής μεταφοράς μεγάλου όγκου πληροφοριών οδηγεί σε σημαντική διπλή αποθήκευση δεδομένων, αυξάνοντας το κόστος υποδομών και περιορίζοντας την αποδοτικότητα της διαχείρισης πόρων[2], [3]. Η χρονική υστέρηση που εισάγεται από τις batch διαδικασίες ETL μειώνει τη δυνατότητα real-time ανάλυσης, γεγονός που καθιστά τη συγκεκριμένη αρχιτεκτονική λιγότερο κατάλληλη για εφαρμογές που απαιτούν άμεση πρόσβαση σε ενημερωμένα δεδομένα[4], [11]. Συναφώς, η ενσωμάτωση νέων πηγών δεδομένων προϋποθέτει συχνά επανασχεδιασμό των transformation διαδικασιών και προσαρμογή των schemas, γεγονός που περιορίζει την ευελιξία και αυξάνει την πολυπλοκότητα συντήρησης του συστήματος[5], [6]. Εξίσου σημαντικό ζήτημα αποτελεί η απώλεια αυτονομίας των επιμέρους πηγών, καθώς τα δεδομένα απομακρύνονται από τα αρχικά τους περιβάλλοντα και ενσωματώνονται σε κεντρικό σύστημα, γεγονός που μπορεί να δημιουργήσει προβλήματα διακυβέρνησης και συμμόρφωσης με πολιτικές data ownership[4], [18]. Παράλληλα, η συγκέντρωση της πληροφορίας σε ενιαίο σημείο αυξάνει την επιφάνεια κινδύνου σε επίπεδο ασφάλειας, καθιστώντας το σύστημα ευάλωτο σε επιθέσεις ή αποτυχίες που επηρεάζουν το σύνολο των δεδομένων[29].

3.2.2 Καταναμημένη αρχιτεκτονική

Η καταναμημένη αρχιτεκτονική βασίζεται στη διατήρηση των δεδομένων σε πολλαπλούς κόμβους ή αποθηκευτικά συστήματα, επιτρέποντας την επεξεργασία τους μέσω μηχανισμών παράλληλης εκτέλεσης και καταναμημένου ελέγχου, χωρίς να απαιτείται η πλήρης συγκέντρωσή τους σε ενιαίο αποθετήριο[2], [3]. Σε αντίθεση με τις παραδοσιακές προσεγγίσεις ETL, τα δεδομένα παραμένουν σε επιμέρους υποδομές, ενώ η επεξεργασία πραγματοποιείται είτε μέσω replication είτε μέσω distributed query engines, που επιτρέπουν την εκτέλεση ερωτημάτων σε πολλαπλές τοποθεσίες[10], [28]. Παράλληλα, η αξιοποίηση τεχνολογιών όπως distributed storage systems και parallel processing frameworks ενισχύει την ικανότητα διαχείρισης μεγάλου όγκου δεδομένων, καθιστώντας την αρχιτεκτονική κατάλληλη για περιβάλλοντα υψηλής κλίμακας[11], [15]. Επιπροσθέτως, η καταναμημένη φύση της αρχιτεκτονικής επιτρέπει την καλύτερη αξιοποίηση υπολογιστικών πόρων, καθώς τα workloads κατανέμονται σε πολλαπλούς κόμβους, μειώνοντας τα bottlenecks και βελτιώνοντας την απόδοση σε σύγκριση με τα κεντριοποιημένα μοντέλα[28]. Η δυνατότητα

επεκτασιμότητας επιτυγχάνεται μέσω της προσθήκης νέων κόμβων στο σύστημα, χωρίς να απαιτείται ανασχεδιασμός της συνολικής υποδομής, γεγονός που καθιστά την αρχιτεκτονική ιδιαίτερα ευέλικτη σε δυναμικά περιβάλλοντα δεδομένων[11], [12].

Παρά τα σημαντικά πλεονεκτήματα που προσφέρει η κατανεμημένη αρχιτεκτονική, η υλοποίησή της συνοδεύεται από αυξημένη πολυπλοκότητα, ιδιαίτερα όσον αφορά τη διαχείριση συνοχής και συγχρονισμού δεδομένων μεταξύ των επιμέρους κόμβων. Καταρχάς, η ανάγκη διατήρησης consistency σε κατανεμημένα συστήματα οδηγεί στην υιοθέτηση μοντέλων όπως eventual consistency ή strong consistency, ανάλογα με τις απαιτήσεις της εφαρμογής, δημιουργώντας συμβιβασμούς μεταξύ απόδοσης και ακρίβειας των αποτελεσμάτων[2], [5]. Η διαχείριση κατανεμημένων transactions αποτελεί σύνθετη διαδικασία, καθώς απαιτεί συντονισμό μεταξύ διαφορετικών συστημάτων και αντιμετώπιση πιθανών αποτυχιών κατά την εκτέλεση[7]. Η ετερογένεια των δεδομένων και των υποκείμενων τεχνολογιών εντείνει τις δυσκολίες ενοποίησης, καθώς διαφορετικά data models και formats απαιτούν εξειδικευμένους μηχανισμούς μετασχηματισμού και αντιστοίχισης[6], [21].

Η εκτέλεση κατανεμημένων ερωτημάτων μπορεί να επιβαρυνθεί από καθυστερήσεις δικτύου και κόστος μεταφοράς δεδομένων μεταξύ κόμβων, γεγονός που επηρεάζει την απόδοση σε περιπτώσεις σύνθετων αναλυτικών φορτίων εργασίας[10]. Επιπλέον, η διαχείριση ασφάλειας σε κατανεμημένα περιβάλλοντα καθίσταται πιο απαιτητική, καθώς απαιτείται συντονισμός πολιτικών πρόσβασης και προστασίας δεδομένων σε πολλαπλά επίπεδα υποδομών[24], [29]. Ωστόσο, η δυνατότητα συνδυασμού κατανεμημένων τεχνολογιών με σύγχρονες πλατφόρμες, όπως lakehouse architectures και hybrid database systems, επιτρέπει την επίτευξη ισορροπίας μεταξύ απόδοσης, ευελιξίας και διαλειτουργικότητας[12], [30].

3.2.3 Ομοσπονδιακή αρχιτεκτονική

Σε αντίθεση με τις κεντριοποιημένες και κατανεμημένες προσεγγίσεις που βασίζονται σε μεταφορά ή αντιγραφή δεδομένων, η ομοσπονδιακή αρχιτεκτονική επιτρέπει την εκτέλεση ερωτημάτων απευθείας στις αρχικές πηγές, αξιοποιώντας ένα ενιαίο λογικό επίπεδο πρόσβασης που αποκρύπτει την πολυπλοκότητα της ετερογένειας[1], [2]. Στο πλαίσιο αυτό, οι federated query engines λειτουργούν ως ενδιάμεσοι μηχανισμοί που αναλύουν τα εισερχόμενα ερωτήματα, τα αποσυνθέτουν σε επιμέρους υπο-ερωτήματα και τα εκτελούν σε διαφορετικά συστήματα, συνθέτοντας στη συνέχεια τα αποτελέσματα σε ενιαία μορφή[2], [3]. Η προσέγγιση αυτή επιτρέπει την αποφυγή διπλότυπων δεδομένων, μειώνοντας σημαντικά την επιβάρυνση αποθήκευσης και διασφαλίζοντας ότι τα αποτελέσματα αντικατοπτρίζουν την πιο πρόσφατη κατάσταση των δεδομένων[1], [4]. Η ενσωμάτωση semantic layers και knowledge graphs ενισχύει τη δυνατότητα ερμηνείας και ενοποίησης δεδομένων με διαφορετικά σχήματα και σημασιολογικές δομές, επιτρέποντας την εκτέλεση πολύπλοκων ερωτημάτων σε περιβάλλοντα υψηλής ετερογένειας[9], [21].

Σε ένα πανεπιστημιακό περιβάλλον, δεδομένα που αφορούν φοιτητές, μαθήματα και ερευνητικές δραστηριότητες μπορεί να βρίσκονται σε διαφορετικά πληροφοριακά συστήματα, όπως σχεσιακές βάσεις δεδομένων, cloud storage και εξωτερικές πλατφόρμες συνεργασίας. Μέσω μιας ομοσπονδιακής προσέγγισης, ένα ενιαίο ερώτημα μπορεί να ανακτήσει πληροφορίες από όλες αυτές τις πηγές, χωρίς να απαιτείται η φυσική ενοποίησή τους, διατηρώντας παράλληλα την ανεξαρτησία κάθε συστήματος[5], [27]. Αντίστοιχα, σε τομείς όπως η υγεία, η ομοσπονδιακή ενοποίηση επιτρέπει την ανάλυση δεδομένων από διαφορετικά νοσοκομεία και ερευνητικά κέντρα, χωρίς παραβίαση κανονισμών ιδιωτικότητας, καθώς τα δεδομένα παραμένουν τοπικά και η επεξεργασία πραγματοποιείται μέσω ασφαλών μηχανισμών[17], [19]. Η χρήση federated learning ενισχύει περαιτέρω αυτή τη δυνατότητα,

επιτρέποντας την εκπαίδευση μοντέλων μηχανικής μάθησης χωρίς μεταφορά ευαίσθητων δεδομένων μεταξύ οργανισμών[20].

Παρά τα σημαντικά πλεονεκτήματα, η ομοσπονδιακή αρχιτεκτονική συνοδεύεται από προκλήσεις που σχετίζονται κυρίως με την απόδοση και τη βελτιστοποίηση εκτέλεσης ερωτημάτων. Η ανάγκη συντονισμού μεταξύ πολλαπλών πηγών δεδομένων μπορεί να εισάγει καθυστερήσεις λόγω δικτυακών περιορισμών και διαφορετικών χρόνων απόκρισης των συστημάτων[2], [10]. Επιπλέον, η βελτιστοποίηση των federated queries απαιτεί προηγμένους αλγορίθμους cost-based planning, οι οποίοι λαμβάνουν υπόψη παράγοντες όπως το μέγεθος των δεδομένων, η τοποθεσία τους και η υπολογιστική ικανότητα των επιμέρους κόμβων[2], [26]. Συναφώς, η διαχείριση ασφάλειας και πολιτικών πρόσβασης καθίσταται πιο σύνθετη, καθώς απαιτείται εναρμόνιση διαφορετικών μηχανισμών ελέγχου σε πολλαπλά συστήματα[4], [24]. Παρά τις προκλήσεις αυτές, η ομοσπονδιακή αρχιτεκτονική αποτελεί τον θεμέλιο λίθο των Data Spaces, προσφέροντας μια ισορροπημένη λύση μεταξύ αυτονομίας, ευελιξίας και αποδοτικότητας, επιτρέποντας την αξιοποίηση κατανεμημένων δεδομένων χωρίς την ανάγκη κεντροποίησης και ανοίγοντας τον δρόμο για προηγμένα μοντέλα ανάλυσης σε περιβάλλοντα μεγάλης κλίμακας[1], [3], [18].

3.2.4 Υβριδικές προσεγγίσεις

Οι υβριδικές αρχιτεκτονικές ενσωματώνουν μηχανισμούς αποθήκευσης και επεξεργασίας δεδομένων σε πολλαπλά επίπεδα, επιτρέποντας τη συνύπαρξη data lakes, data warehouses και federated query layers σε ένα ενιαίο λειτουργικό περιβάλλον[11], [12]. Η προσέγγιση αυτή βασίζεται στην αρχή της επιλεκτικής μεταφοράς δεδομένων, όπου μόνο κρίσιμα ή συχνά χρησιμοποιούμενα δεδομένα αποθηκεύονται κεντρικά, ενώ τα υπόλοιπα παραμένουν στις αρχικές τους πηγές και προσπελάζονται δυναμικά μέσω virtualization τεχνικών[3], [11]. Η αξιοποίηση lakehouse architectures επιτρέπει την ενοποίηση συναλλακτικών και αναλυτικών φορτίων εργασίας σε κοινή υποδομή, μειώνοντας την πολυπλοκότητα και ενισχύοντας την απόδοση σε περιβάλλοντα μεγάλου όγκου δεδομένων[12], [15]. Επιπροσθέτως, η ενσωμάτωση serverless τεχνολογιών επιτρέπει την ελαστική εκτέλεση ερωτημάτων και την προσαρμογή των υπολογιστικών πόρων ανάλογα με τις ανάγκες του συστήματος, βελτιώνοντας τη συνολική αποδοτικότητα[10], [28]. Συναφώς, η υβριδική αρχιτεκτονική υποστηρίζει την ανάπτυξη πολύπλοκων pipelines επεξεργασίας δεδομένων, τα οποία συνδυάζουν batch και real-time processing, επιτρέποντας την ανάλυση δεδομένων τόσο σε ιστορικό όσο και σε τρέχον επίπεδο[15].

Η πρακτική εφαρμογή των υβριδικών προσεγγίσεων αποδεικνύεται ιδιαίτερα αποτελεσματική σε περιβάλλοντα όπου συνυπάρχουν διαφορετικές απαιτήσεις απόδοσης, ασφάλειας και διακυβέρνησης δεδομένων. Για παράδειγμα, σε ένα επιχειρησιακό οικοσύστημα, δεδομένα συναλλαγών μπορούν να αποθηκεύονται σε κεντρικό data warehouse για σκοπούς reporting, ενώ δεδομένα από IoT συσκευές ή εξωτερικές πλατφόρμες παραμένουν σε κατανεμημένες πηγές και προσπελάζονται μέσω federated μηχανισμών[8], [24]. Αντίστοιχα, στον τομέα της υγείας, ευαίσθητα δεδομένα ασθενών διατηρούνται τοπικά για λόγους συμμόρφωσης, ενώ ανωνυμοποιημένα σύνολα δεδομένων μεταφέρονται σε κεντρικές υποδομές για ερευνητικούς σκοπούς, συνδυάζοντας έτσι ασφάλεια και αναλυτική ικανότητα[17], [22]. Επιπλέον, η χρήση hybrid database systems, τα οποία ενσωματώνουν χαρακτηριστικά SQL και NoSQL μοντέλων, επιτρέπει την ευέλικτη διαχείριση διαφορετικών τύπων δεδομένων, ενισχύοντας τη διαλειτουργικότητα μεταξύ ετερογενών συστημάτων[30]. Η επιλογή του κατάλληλου μοντέλου επεξεργασίας για κάθε τύπο δεδομένων αποτελεί κρίσιμο ζήτημα, καθώς επηρεάζει άμεσα την απόδοση και το κόστος λειτουργίας[5]. Επιπλέον, η διαχείριση consistency μεταξύ κεντρικών και κατανεμημένων αποθετηρίων απαιτεί προηγμένες στρατηγικές συγχρονισμού, ενώ η υλοποίηση πολιτικών ασφάλειας καθίσταται πιο σύνθετη λόγω της πολυεπίπεδης φύσης της

αρχιτεκτονικής[24], [29]. Ωστόσο, η υβριδική προσέγγιση προσφέρει μια ισορροπημένη λύση για την αντιμετώπιση των προκλήσεων της σύγχρονης ενοποίησης δεδομένων, επιτρέποντας την ευέλικτη προσαρμογή σε διαφορετικά σενάρια χρήσης και υποστηρίζοντας την ανάπτυξη αποδοτικών και επεκτάσιμων Data Spaces[1], [4], [18].

Πίνακας 3.2: Σύγκριση χαρακτηριστικών υβριδικών αρχιτεκτονικών ενοποίησης δεδομένων

Χαρακτηριστικό	Περιγραφή	Τεχνολογική υλοποίηση	Πλεονέκτημα
Συνδυασμός μοντέλων	Ενοποίηση κεντριοποιημένων και ομοσπονδιακών μηχανισμών	Lakehouse + Federated engines	Ευελιξία στη διαχείριση δεδομένων
Επιλεκτική αποθήκευση	Μερική μεταφορά δεδομένων σε κεντρικά αποθετήρια	Data lakes, warehouses	Μείωση κόστους αποθήκευσης
Virtualization	Πρόσβαση σε δεδομένα χωρίς φυσική μεταφορά	Data virtualization tools	Ελαχιστοποίηση duplication
Υποστήριξη real-time	Συνδυασμός batch και streaming επεξεργασίας	Stream processing engines	Άμεση ανάλυση δεδομένων
Επεκτασιμότητα	Δυναμική προσαρμογή πόρων	Cloud και serverless υποδομές	Βελτιστοποίηση απόδοσης
Διαλειτουργικότητα	Ενοποίηση ετερογενών συστημάτων	Hybrid databases, semantic layers	Συμβατότητα πολλαπλών formats
Ασφάλεια και διακυβέρνηση	Πολυεπίπεδος έλεγχος πρόσβασης	Policy enforcement frameworks	Προστασία ευαίσθητων δεδομένων
Πολυπλοκότητα διαχείρισης	Συντονισμός πολλαπλών τεχνολογιών	Orchestration εργαλεία	Αυξημένη λειτουργική ευελιξία

Πηγές:[3], [11], [12], [13], [17], [24], [30]

3.3 Παρουσίαση του Trino

3.3.1 Γενικά χαρακτηριστικά

Η αρχιτεκτονική του Trino βασίζεται στο μοντέλο coordinator-worker, όπου ένας κεντρικός κόμβος (coordinator) αναλαμβάνει την ανάλυση, τον σχεδιασμό και τη βελτιστοποίηση των ερωτημάτων, ενώ πολλαπλοί worker nodes εκτελούν παράλληλα τις επιμέρους διαδικασίες (tasks)[2], [27]. Ο coordinator μετατρέπει ένα SQL ερώτημα σε execution plan, το οποίο διασπάται σε stages και tasks, επιτρέποντας την κατανομημένη επεξεργασία δεδομένων με υψηλό βαθμό παραλληλισμού. Η αρχιτεκτονική αυτή υποστηρίζει pipeline execution, όπου τα δεδομένα ρέουν μεταξύ των stages χωρίς την ανάγκη ενδιάμεσης αποθήκευσης σε δίσκο, μειώνοντας σημαντικά το latency κατά την εκτέλεση σύνθετων αναλυτικών ερωτημάτων[10]. Η χρήση in-memory processing ενισχύει την ταχύτητα επεξεργασίας, καθώς τα δεδομένα μεταφέρονται και επεξεργάζονται κυρίως στη μνήμη των worker nodes, περιορίζοντας τα I/O bottlenecks[28]. Ένα από τα βασικά χαρακτηριστικά της αρχιτεκτονικής του Trino είναι η ύπαρξη connectors, οι οποίοι λειτουργούν ως ενδιάμεσα επίπεδα διασύνδεσης μεταξύ του query engine και των επιμέρους πηγών δεδομένων. Μέσω αυτών των connectors, το Trino μπορεί να αλληλεπιδρά με διαφορετικά συστήματα αποθήκευσης, όπως σχεσιακές βάσεις δεδομένων, data lakes και NoSQL βάσεις, χωρίς να απαιτείται μεταφορά ή αντιγραφή των δεδομένων[3], [5].

Η συγκεκριμένη σχεδιαστική επιλογή επιτρέπει την υλοποίηση ομοσπονδιακών queries, όπου ένα ενιαίο ερώτημα μπορεί να εκτελεστεί ταυτόχρονα σε πολλαπλές πηγές, αξιοποιώντας τη λογική ενοποίηση δεδομένων χωρίς φυσική συγκέντρωση[1], [2]. Συναφώς, η αρχιτεκτονική του Trino υποστηρίζει cost-based optimization, λαμβάνοντας υπόψη χαρακτηριστικά των δεδομένων και των πηγών για τη βελτιστοποίηση του execution plan, γεγονός που ενισχύει την αποδοτικότητα σε περιβάλλοντα υψηλής πολυπλοκότητας[2], [26]. Επιπλέον, η δυνατότητα pushdown processing επιτρέπει τη μεταφορά μέρους της επεξεργασίας κοντά στις πηγές δεδομένων, μειώνοντας τον όγκο δεδομένων που μεταφέρεται μέσω του δικτύου και βελτιώνοντας την απόδοση[3]. Σημαντική διάσταση της αρχιτεκτονικής του Trino αποτελεί επίσης η δυνατότητα οριζόντιας κλιμάκωσης, καθώς νέοι worker nodes μπορούν να προστεθούν δυναμικά στο σύστημα, ενισχύοντας την υπολογιστική ισχύ χωρίς να απαιτείται διακοπή λειτουργίας[28]. Η αρχιτεκτονική αυτή επιτρέπει την αποδοτική διαχείριση μεγάλων όγκων δεδομένων, καθιστώντας το Trino κατάλληλο για περιβάλλοντα big data και data lakes[11], [12]. Επιπλέον, η υποστήριξη fault tolerance επιτυγχάνεται μέσω μηχανισμών επανακατανομής tasks σε περίπτωση αποτυχίας κόμβων, διασφαλίζοντας τη συνέχιση της εκτέλεσης των ερωτημάτων[10]. Παράλληλα, η αρχιτεκτονική του Trino ενσωματώνει μηχανισμούς ασφαλούς πρόσβασης και ελέγχου πολιτικών, επιτρέποντας την εφαρμογή authentication και authorization σε επίπεδο query execution[24], [29].

Πίνακας 3.3: Βασικά αρχιτεκτονικά χαρακτηριστικά του Trino

Χαρακτηριστικό	Περιγραφή	Αρχιτεκτονικός ρόλος	Επιπτώσεις στην απόδοση
Coordinator–Worker μοντέλο	Διαχωρισμός σχεδιασμού και εκτέλεσης ερωτημάτων	Κεντρικός έλεγχος και κατανεμημένη εκτέλεση	Βελτιστοποιημένος παραλληλισμός
Parallel execution	Διάσπαση queries σε stages και tasks	Κατανεμημένη επεξεργασία	Μείωση χρόνου εκτέλεσης
In-memory processing	Επεξεργασία δεδομένων κυρίως στη μνήμη	Ελαχιστοποίηση I/O	Χαμηλό latency
Pipeline execution	Συνεχής ροή δεδομένων μεταξύ stages	Αποφυγή ενδιάμεσης αποθήκευσης	Ταχύτερη εκτέλεση
Connectors	Διασύνδεση με πολλαπλές πηγές δεδομένων	Ομοσπονδιακή πρόσβαση	Εξάλειψη data duplication
Query optimization	Cost-based planning και βελτιστοποίηση	Αποδοτική εκτέλεση queries	Βελτιωμένη απόδοση
Pushdown processing	Μεταφορά υπολογισμών στις πηγές	Μείωση μεταφοράς δεδομένων	Βελτίωση αποδοτικότητας
Οριζόντια επεκτασιμότητα	Δυναμική προσθήκη worker nodes	Επεκτασιμότητα συστήματος	Υποστήριξη μεγάλου όγκου δεδομένων
Fault tolerance	Διαχείριση αποτυχιών κόμβων	Αξιοπιστία εκτέλεσης	Συνέχιση λειτουργίας
Security integration	Ενσωμάτωση μηχανισμών πρόσβασης	Έλεγχος ασφάλειας	Προστασία δεδομένων

Πηγές:[1], [2], [3], [5], [10], [26], [27], [28]

3.3.2 Αρχιτεκτονική Trino

Το Trino υιοθετεί μια λογική εκτέλεσης “in situ”, όπου τα δεδομένα παραμένουν στις αρχικές τους θέσεις και η επεξεργασία πραγματοποιείται κατανεμημένα μέσω ενός συνόλου συνεργαζόμενων κόμβων. Η θεμελιώδης αυτή επιλογή ευθυγραμμίζεται με τις αρχές της ομοσπονδιακής ενοποίησης

δεδομένων και της εικονικοποίησης, οι οποίες δίνουν έμφαση στην ευελιξία και τη διαλειτουργικότητα[1], [3]. Η δομή του Trino βασίζεται σε ένα σαφώς ορισμένο μοντέλο διαχωρισμού ρόλων, όπου ο coordinator κόμβος αναλαμβάνει τον πλήρη έλεγχο του κύκλου ζωής ενός ερωτήματος, από τη λήψη και την ανάλυση έως τη βελτιστοποίηση και τον προγραμματισμό εκτέλεσης. Ο coordinator δεν εκτελεί υπολογισμούς μεγάλης κλίμακας, αλλά λειτουργεί ως κεντρικός ενορχηστρωτής, δημιουργώντας ένα λογικό πλάνο εκτέλεσης που διασπά το αρχικό ερώτημα σε επιμέρους στάδια. Αυτή η διάσπαση οδηγεί στη δημιουργία tasks που κατανέμονται στους worker κόμβους, οι οποίοι αποτελούν τη βασική μονάδα υπολογιστικής ισχύος του συστήματος. Η συγκεκριμένη αρχιτεκτονική προσέγγιση επιτρέπει υψηλό βαθμό παραλληλισμού και αξιοποίηση των διαθέσιμων πόρων, χαρακτηριστικό που συνδέεται άμεσα με τις πρακτικές κατανεμημένων συστημάτων επεξεργασίας μεγάλων δεδομένων[2], [27].

Από λειτουργική σκοπιά, η εκτέλεση των ερωτημάτων στο Trino πραγματοποιείται μέσω ενός μηχανισμού pipeline processing, όπου τα δεδομένα ρέουν συνεχώς μεταξύ των επιμέρους σταδίων χωρίς την ανάγκη ενδιάμεσης αποθήκευσης. Η χρήση in-memory τεχνικών επιτρέπει την ελαχιστοποίηση καθυστερήσεων που σχετίζονται με λειτουργίες εισόδου/εξόδου, ενώ η δυναμική κατανομή φορτίου στους worker κόμβους διασφαλίζει την ισορροπία του συστήματος ακόμα και σε περιβάλλοντα υψηλής ζήτησης. Παράλληλα, η δυνατότητα εκτέλεσης πολλαπλών ερωτημάτων ταυτόχρονα ενισχύει την αποδοτικότητα, ιδιαίτερα σε περιπτώσεις όπου απαιτείται ανάλυση δεδομένων σε πραγματικό χρόνο[10], [28]. Η ενσωμάτωση connectors είναι πολύ σημαντική, καθώς αυτά αποτελούν κρίσιμο στοιχείο της αρχιτεκτονικής, επιτρέποντας την απευθείας επικοινωνία με διαφορετικά συστήματα αποθήκευσης, όπως βάσεις δεδομένων, data lakes ή αποθετήρια τύπου lakehouse. Οι connectors λειτουργούν ως αφαιρετικά επίπεδα που μεταφράζουν τα αιτήματα του Trino σε εγγενείς εντολές των επιμέρους πηγών, διασφαλίζοντας παράλληλα τη διατήρηση της σημασιολογικής συνοχής. Επιπλέον, μέσω τεχνικών όπως το pushdown processing, μέρος της επεξεργασίας μεταφέρεται στις ίδιες τις πηγές δεδομένων, μειώνοντας τον όγκο μεταφοράς και βελτιώνοντας την απόδοση[5], [11], [12].

3.3.3 Coordinator και Workers

Εντός της αρχιτεκτονικής του Trino, ο coordinator κόμβος συγκροτεί τον πυρήνα ελέγχου, αναλαμβάνοντας την πλήρη ευθύνη για την επεξεργασία και δρομολόγηση των εισερχόμενων ερωτημάτων, λειτουργώντας ως σημείο συγκέντρωσης λογικής ευφυΐας του συστήματος. Με την παραλαβή ενός SQL αιτήματος, ενεργοποιούνται διαδοχικά στάδια ανάλυσης, όπου πραγματοποιείται συντακτικός και σημασιολογικός έλεγχος, ακολουθούμενος από τη δημιουργία ενός βελτιστοποιημένου λογικού πλάνου εκτέλεσης. Στο πλαίσιο αυτό, εφαρμόζονται τεχνικές βελτιστοποίησης που αποσκοπούν στη μείωση του υπολογιστικού κόστους, όπως αναδιάταξη joins και προώθηση φίλτρων προς τα δεδομένα, πρακτικές που συνδέονται άμεσα με σύγχρονες προσεγγίσεις ομοσπονδιακής επεξεργασίας[2], [3]. Κατόπιν, το πλάνο μετασχηματίζεται σε φυσικό σχέδιο εκτέλεσης, όπου το ερώτημα διασπάται σε επιμέρους stages και tasks, τα οποία πρόκειται να εκτελεστούν παράλληλα σε κατανεμημένο περιβάλλον.

Παράλληλα, οι worker κόμβοι συνιστούν τα εκτελεστικά στοιχεία της αρχιτεκτονικής, αναλαμβάνοντας την πραγματική επεξεργασία των δεδομένων μέσω της εκτέλεσης των tasks που τους ανατίθενται από τον coordinator. Κάθε worker διαχειρίζεται πολλαπλά νήματα εκτέλεσης, επιτρέποντας την ταυτόχρονη επεξεργασία διαφορετικών τμημάτων ενός ερωτήματος, γεγονός που ενισχύει σημαντικά τον βαθμό παραλληλισμού. Η λειτουργία τους βασίζεται σε in-memory επεξεργασία, όπου τα δεδομένα διακινούνται σε μορφή streams μεταξύ operators, αποφεύγοντας τη χρήση ενδιάμεσης αποθήκευσης και μειώνοντας δραστικά τις καθυστερήσεις εισόδου/εξόδου. Επιπλέον, η αρχιτεκτονική υποστηρίζει

δυναμική εξισορρόπηση φορτίου, καθώς οι workers προσαρμόζουν τη χρήση των πόρων τους ανάλογα με την πολυπλοκότητα των ανατεθειμένων εργασιών, χαρακτηριστικό που ευθυγραμμίζεται με τις απαιτήσεις ελαστικότητας των σύγχρονων καταναμημένων υποδομών[10], [28].

Η συνεργασία μεταξύ coordinator και worker κόμβων υλοποιείται μέσω ενός σαφώς καθορισμένου μηχανισμού επικοινωνίας, ο οποίος βασίζεται σε ανταλλαγή ενδιάμεσων αποτελεσμάτων σε μορφή pipelines. Κατά τη διάρκεια εκτέλεσης ενός ερωτήματος, τα δεδομένα μεταφέρονται απευθείας από έναν worker σε άλλον χωρίς να επιστρέφουν στον coordinator, περιορίζοντας έτσι τον κίνδυνο δημιουργίας σημείων συμφόρησης. Ο coordinator διατηρεί ρόλο εποπτείας, παρακολουθώντας την πρόοδο των tasks και επεμβαίνοντας σε περιπτώσεις αποτυχίας μέσω επανακατανομής εργασιών. Αυτή η ανθεκτικότητα αποτελεί κρίσιμο χαρακτηριστικό για περιβάλλοντα υψηλής διαθεσιμότητας, όπου η απώλεια κόμβων δεν πρέπει να επηρεάζει τη συνολική λειτουργία του συστήματος[27]. Ταυτόχρονα, η αρχιτεκτονική Trino ενσωματώνει μηχανισμούς που επιτρέπουν την αποδοτική εκτέλεση ερωτημάτων σε πολλαπλές ετερογενείς πηγές δεδομένων μέσω connectors, με τους workers να αναλαμβάνουν την αλληλεπίδραση με τα επιμέρους συστήματα. Σε αυτό το πλαίσιο, κάθε worker μπορεί να εκτελεί τμήματα ενός ερωτήματος απευθείας στην πηγή δεδομένων, αξιοποιώντας τεχνικές pushdown processing, μειώνοντας τον μεταφερόμενο όγκο δεδομένων και αυξάνοντας τη συνολική απόδοση. Η συγκεκριμένη λειτουργικότητα είναι άμεσα συνδεδεμένη με πρακτικές εικονικοποίησης δεδομένων και πολυπλαθομικής διαχείρισης, οι οποίες επιτρέπουν ενοποιημένη πρόσβαση χωρίς φυσική ενοποίηση[5].

Στο επίπεδο διαχείρισης πόρων, ο coordinator διατηρεί συνολική εικόνα του συστήματος, κατανέμοντας δυναμικά τα tasks στους workers με βάση τη διαθεσιμότητα και το φορτίο τους, ενώ οι workers εκτελούν τις εργασίες αξιοποιώντας τοπικά διαθέσιμους πόρους μνήμης και επεξεργαστικής ισχύος. Η συγκεκριμένη αρχιτεκτονική κατανομή αρμοδιοτήτων επιτρέπει στο Trino να επιτυγχάνει υψηλή επεκτασιμότητα, καθώς η προσθήκη νέων worker κόμβων οδηγεί σε γραμμική αύξηση της υπολογιστικής ικανότητας, στοιχείο κρίσιμο για την επεξεργασία μεγάλου όγκου δεδομένων σε καταναμημένα περιβάλλοντα[11], [12].

3.3.4 Connectors και Catalogs

Οι connectors συγκροτούν το θεμελιώδες υπόστρωμα διασύνδεσης με ετερογενείς πηγές δεδομένων, υλοποιώντας έναν αφαιρετικό μηχανισμό πρόσβασης που επιτρέπει στο σύστημα να εκτελεί ερωτήματα χωρίς άμεση εξάρτηση από τις εγγενείς ιδιαιτερότητες κάθε αποθηκευτικού συστήματος. Κάθε connector λειτουργεί ως μεταφραστικό στρώμα μεταξύ της ενιαίας SQL διεπαφής του Trino και των επιμέρους APIs των πηγών, αναλαμβάνοντας την αντιστοίχιση εντολών, τύπων δεδομένων και λειτουργιών. Η συγκεκριμένη αρχιτεκτονική επιλογή εξασφαλίζει διαλειτουργικότητα σε περιβάλλοντα όπου συνυπάρχουν σχεσιακές βάσεις, αποθήκες δεδομένων και συστήματα τύπου data lake, υλοποιώντας στην πράξη αρχές εικονικοποίησης και ομοσπονδιακής ενοποίησης[1], [3]. Η αποσύνδεση της λογικής ερωτήματος από τη φυσική αποθήκευση επιτρέπει την ευέλικτη ένταξη νέων πηγών χωρίς μεταβολή του κεντρικού μηχανισμού επεξεργασίας, στοιχείο κρίσιμο για δυναμικά οικοσυστήματα δεδομένων. Τα catalogs αποτελούν τη λογική οντότητα μέσω της οποίας οργανώνονται και διαχειρίζονται οι διαθέσιμες πηγές δεδομένων στο Trino, λειτουργώντας ως σημεία εισόδου για την πρόσβαση σε διαφορετικά συστήματα. Κάθε catalog αντιστοιχεί σε έναν συγκεκριμένο connector και περιλαμβάνει πληροφορίες σχετικά με τα schemas και τα tables που είναι προσβάσιμα μέσω αυτού. Με τον τρόπο αυτό, το Trino υιοθετεί μια ιεραρχική δομή ονοματοδοσίας της μορφής catalog.schema.table, η οποία επιτρέπει την ομοιόμορφη αναφορά σε δεδομένα ανεξαρτήτως της φυσικής τους τοποθεσίας.

Η ύπαρξη catalogs διευκολύνει τη διαχείριση πολλαπλών πηγών, παρέχοντας ένα ενοποιημένο επίπεδο μεταδεδομένων που καθιστά εφικτή τη σύνθεση ερωτημάτων που διατρέχουν διαφορετικά συστήματα, πρακτική που συνδέεται με σύγχρονες προσεγγίσεις πολυπλατφορμικής διαχείρισης δεδομένων[5]. Οι connectors ενσωματώνουν μηχανισμούς pushdown processing, μέσω των οποίων μέρος της επεξεργασίας μεταφέρεται απευθείας στην πηγή δεδομένων, μειώνοντας τον όγκο των δεδομένων που πρέπει να μεταφερθούν προς τους worker κόμβους. Συγκεκριμένα, λειτουργίες όπως φίλτρα, προβολές και ορισμένες συνενώσεις μπορούν να εκτελούνται τοπικά στο υποκείμενο σύστημα, εφόσον αυτό υποστηρίζει τις αντίστοιχες δυνατότητες. Η προσέγγιση αυτή βελτιώνει σημαντικά την απόδοση, ιδίως σε περιβάλλοντα όπου τα δεδομένα είναι κατανεμημένα σε πολλαπλά αποθετήρια μεγάλης κλίμακας, μειώνοντας τόσο το δίκτυο όσο και το υπολογιστικό φορτίο των workers[2], [11]. Η αρχιτεκτονική διαχείρισης μεταδεδομένων μέσω catalogs επιτρέπει στο Trino να διατηρεί συνεκτική εικόνα των διαθέσιμων δομών δεδομένων, ακόμη και όταν αυτές προέρχονται από διαφορετικά συστήματα με ασύμβατα σχήματα. Μέσω των connectors, επιτυγχάνεται αντιστοίχιση τύπων δεδομένων και ενοποίηση σχημάτων, εξασφαλίζοντας ότι τα ερωτήματα εκτελούνται με συνεπή σημασιολογία. Η διαδικασία αυτή είναι κρίσιμη σε περιβάλλοντα data spaces, όπου η ετερογένεια αποτελεί βασικό χαρακτηριστικό και η ανάγκη για σημασιολογική συνοχή είναι επιτακτική[6], [21].

3.3.5 Query execution model

Η διαδικασία αυτή εκκινεί με τη συντακτική και σημασιολογική ανάλυση του ερωτήματος στον coordinator κόμβο, όπου παράγεται ένα λογικό σχέδιο (logical plan) το οποίο αποτυπώνει τις πράξεις που απαιτούνται, όπως προβολές, συνενώσεις και φιλτραρίσματα. Στη συνέχεια, εφαρμόζονται βελτιστοποιήσεις που περιλαμβάνουν αναδιάταξη τελεστών και απλοποίηση εκφράσεων, με στόχο τη μείωση του κόστους εκτέλεσης, σύμφωνα με αρχές που παρατηρούνται σε σύγχρονα συστήματα ομοσπονδιακής επεξεργασίας[2], [3]. Στη συνέχεια, το λογικό σχέδιο μετασχηματίζεται σε φυσικό πλάνο εκτέλεσης (physical plan), το οποίο λαμβάνει υπόψη τη διαθεσιμότητα των δεδομένων και την κατανομή τους στους connectors. Το Trino διαχωρίζει το πλάνο σε στάδια, τα οποία αποτελούν ανεξάρτητες μονάδες εργασίας που μπορούν να εκτελούνται παράλληλα. Κάθε στάδιο περιλαμβάνει πολλαπλά tasks, τα οποία ανατίθενται σε worker κόμβους, επιτρέποντας την ταυτόχρονη επεξεργασία διαφορετικών τμημάτων του ίδιου ερωτήματος. Η συγκεκριμένη σχεδιαστική επιλογή ενισχύει την κλιμακωσιμότητα του συστήματος, καθώς η αύξηση των workers οδηγεί σε ανάλογη αύξηση της υπολογιστικής ισχύος, χαρακτηριστικό που συνδέεται με κατανεμημένες αρχιτεκτονικές επεξεργασίας μεγάλων δεδομένων[15], [27].

Η εκτέλεση των tasks βασίζεται σε ένα pipeline μοντέλο, όπου οι ενδιάμεσες ροές δεδομένων μεταδίδονται συνεχώς μεταξύ των τελεστών χωρίς να απαιτείται πλήρης υλοποίηση των αποτελεσμάτων σε ενδιάμεσα στάδια. Κάθε task αποτελείται από operators που οργανώνονται σε pipelines και εκτελούνται με τρόπο που μεγιστοποιεί την τοπικότητα δεδομένων και ελαχιστοποιεί τη μεταφορά τους μέσω δικτύου. Η αρχιτεκτονική αυτή επιτρέπει την επεξεργασία δεδομένων σε streaming μορφή, ενισχύοντας την αποδοτικότητα σε περιβάλλοντα μεγάλης κλίμακας και ευθυγραμμίζεται με πρακτικές που αξιοποιούνται σε σύγχρονες lakehouse υποδομές[11], [12]. Η επικοινωνία μεταξύ των stages επιτυγχάνεται μέσω μηχανισμών ανταλλαγής δεδομένων (data exchange), οι οποίοι υποστηρίζουν τόσο broadcast όσο και partitioned μεταφορά, ανάλογα με τις απαιτήσεις του ερωτήματος. Η επιλογή στρατηγικής βασίζεται σε εκτιμήσεις κόστους και στη δομή των δεδομένων, με στόχο τη βελτιστοποίηση της απόδοσης κατά την εκτέλεση σύνθετων συνενώσεων. Η δυναμική αυτή κατανομή εργασίας επιτρέπει στο Trino να προσαρμόζεται σε διαφορετικά σενάρια

φόρτου, διατηρώντας υψηλό επίπεδο απόδοσης ακόμη και σε περιπτώσεις μεγάλης ετερογένειας πηγών[10], [28].

Τέλος, το μοντέλο εκτέλεσης ερωτημάτων ενσωματώνει μηχανισμούς παρακολούθησης και ελέγχου της προόδου, επιτρέποντας στον coordinator να διαχειρίζεται την ολοκλήρωση των επιμέρους tasks και να συνθέτει τα τελικά αποτελέσματα. Η αρχιτεκτονική αυτή επιτρέπει την ανθεκτικότητα σε σφάλματα, καθώς η αποτυχία ενός task μπορεί να αντιμετωπιστεί με επανακατανομή του σε άλλον worker, διασφαλίζοντας τη συνέχιση της εκτέλεσης. Μέσω αυτής της κατανεμημένης και παράλληλης προσέγγισης, το Trino επιτυγχάνει υψηλή απόδοση σε ετερογενή περιβάλλοντα δεδομένων, υποστηρίζοντας σύνθετα αναλυτικά ερωτήματα χωρίς την ανάγκη μεταφοράς δεδομένων σε κεντρικό αποθετήριο[26], [29].

Πίνακας 3.4: Κύρια χαρακτηριστικά του μοντέλου εκτέλεσης ερωτημάτων στο Trino

Χαρακτηριστικό	Περιγραφή
Ανάλυση ερωτήματος	Μετατροπή SQL σε λογικό σχέδιο μέσω συντακτικής και σημασιολογικής επεξεργασίας
Βελτιστοποίηση πλάνου	Αναδιάταξη τελεστών και μείωση κόστους εκτέλεσης πριν την κατανομή
Κατακερματισμός σε στάδια	Διάσπαση του πλάνου σε ανεξάρτητα stages για παράλληλη εκτέλεση
Κατανομή σε tasks	Δημιουργία tasks που εκτελούνται σε διαφορετικούς worker κόμβους
Pipeline execution	Συνεχής ροή δεδομένων μεταξύ operators χωρίς ενδιάμεση αποθήκευση
Data exchange	Μηχανισμοί μεταφοράς δεδομένων μεταξύ stages (broadcast/partitioned)
Παραλληλισμός	Ταυτόχρονη επεξεργασία δεδομένων σε πολλούς κόμβους
Δυναμική εκτέλεση	Προσαρμογή κατανομής φόρτου με βάση τη διαθεσιμότητα πόρων
Ανοχή σε σφάλματα	Επανακατανομή tasks σε περίπτωση αποτυχίας worker
Συντονισμός αποτελεσμάτων	Συλλογή και ενοποίηση αποτελεσμάτων από τον coordinator

Πηγές:[2], [3], [10], [11], [12], [15], [26], [27], [28], [29]

3.3.6 Πλεονεκτήματα χρήσης Trino

Το σύστημα λειτουργεί πάνω από υφιστάμενες πηγές, αξιοποιώντας connectors για πρόσβαση σε ετερογενή συστήματα. Η επιλογή αυτή ενισχύει την αποδοτικότητα και περιορίζει τα κόστη διαχείρισης δεδομένων, ιδιαίτερα σε περιβάλλοντα όπου οι πληροφορίες διασπείρονται σε πολλαπλές υποδομές, ευθυγραμμιζόμενη με αρχές ομοσπονδιακής ενοποίησης δεδομένων[1], [3]. Ένα από τα πλέον καθοριστικά πλεονεκτήματα έγκειται στην ικανότητα εκτέλεσης ερωτημάτων απευθείας πάνω σε κατανεμημένα σύνολα δεδομένων, χωρίς ενδιάμεση υλοποίηση. Η αρχιτεκτονική επιτρέπει την ταυτόχρονη πρόσβαση σε πολλαπλές πηγές, συνδυάζοντας δεδομένα από διαφορετικά formats και αποθηκευτικά μέσα σε ενιαίο ερώτημα. Η λειτουργικότητα αυτή ενισχύει τη διαλειτουργικότητα και καθιστά δυνατή την υλοποίηση πολύπλοκων αναλυτικών διαδικασιών σε πραγματικό χρόνο, κάτι που αποτελεί βασικό ζητούμενο σε σύγχρονα data spaces και συστήματα ενοποίησης[2], [24].

Παράλληλα, η σχεδίαση βασισμένη σε μαζικό παραλληλισμό προσφέρει σημαντικά οφέλη απόδοσης, καθώς τα ερωτήματα διασπώνται σε ανεξάρτητες εργασίες που εκτελούνται ταυτόχρονα σε πολλαπλούς worker κόμβους. Η δυνατότητα οριζόντιας κλιμάκωσης επιτρέπει την αύξηση της επεξεργαστικής ισχύος μέσω προσθήκης κόμβων, χωρίς να απαιτούνται σημαντικές αλλαγές στην αρχιτεκτονική. Το χαρακτηριστικό αυτό καθιστά το Trino κατάλληλο για μεγάλης κλίμακας αναλυτικά φορτία, όπου η

επεξεργασία δεδομένων απαιτεί υψηλό βαθμό παραλληλίας, όπως υποδεικνύεται και σε σύγχρονες πλατφόρμες ανάλυσης δεδομένων[15], [27]. Η αξιοποίηση του pipeline execution μοντέλου επιτρέπει τη συνεχή ροή δεδομένων μεταξύ των επιμέρους τελεστών χωρίς τη δημιουργία ενδιάμεσων αποθηκευμένων αποτελεσμάτων. Η προσέγγιση αυτή μειώνει δραστικά την καθυστέρηση και την κατανάλωση πόρων, βελτιώνοντας την απόδοση σε περιβάλλοντα με υψηλό όγκο δεδομένων. Η χρήση streaming επεξεργασίας καθιστά το σύστημα ικανό να ανταποκρίνεται σε απαιτήσεις χαμηλής καθυστέρησης, ιδιαίτερα σε σενάρια όπου απαιτείται άμεση εξαγωγή συμπερασμάτων από δεδομένα που βρίσκονται σε διαφορετικές τοποθεσίες[10], [28].

Τέλος, η ευελιξία της αρχιτεκτονικής εκφράζεται μέσα από την ικανότητά της να προσαρμόζεται σε ετερογενή περιβάλλοντα και ποικίλα σενάρια χρήσης, υποστηρίζοντας πολλαπλά formats δεδομένων και διαφορετικά συστήματα αποθήκευσης. Η δυνατότητα ενσωμάτωσης με lakehouse υποδομές και σύγχρονες τεχνολογίες ανάλυσης επιτρέπει τη δημιουργία ενιαίων αναλυτικών οικοσυστημάτων, χωρίς να απαιτείται ομογενοποίηση των πηγών. Μέσω αυτής της προσέγγισης, το Trino επιτυγχάνει υψηλό επίπεδο αποδοτικότητας και επεκτασιμότητας, καλύπτοντας απαιτήσεις σύγχρονων εφαρμογών δεδομένων σε δυναμικά και καταναμημένα περιβάλλοντα[11], [12].

3.4 Πηγές δεδομένων στο πανεπιστημιακό περιβάλλον

3.4.1 Βάσεις δεδομένων φοιτητών

Η διαχείριση βάσεων δεδομένων φοιτητών σε πανεπιστημιακά πληροφοριακά συστήματα ενσωματώνει πολυδιάστατες απαιτήσεις που σχετίζονται με την αποθήκευση, επεξεργασία και ενοποίηση ετερογενών δεδομένων, τα οποία προέρχονται από διαφορετικές διοικητικές και ακαδημαϊκές μονάδες. Σε τέτοια περιβάλλοντα, τα δεδομένα φοιτητών δεν περιορίζονται σε βασικά δημογραφικά στοιχεία, αλλά επεκτείνονται σε εγγραφές μαθημάτων, επιδόσεις, οικονομικές συναλλαγές και αλληλεπιδράσεις με ψηφιακές υπηρεσίες. Η πολυπλοκότητα αυτή καθιστά απαραίτητη την υιοθέτηση αρχιτεκτονικών που επιτρέπουν την ενοποιημένη πρόσβαση χωρίς την ανάγκη φυσικής συγκέντρωσης των δεδομένων, προσεγγίζοντας μοντέλα ομοσπονδιακής διαχείρισης όπου τα δεδομένα παραμένουν στις πηγές τους και προσπελαύνονται δυναμικά[1], [2]. Η καταναμημένη φύση των πανεπιστημιακών συστημάτων συνεπάγεται την ύπαρξη πολλαπλών βάσεων δεδομένων, οι οποίες λειτουργούν σε διαφορετικά τεχνολογικά περιβάλλοντα, όπως σχεσιακές βάσεις για διοικητικά δεδομένα και αποθήκες δεδομένων για αναλυτικούς σκοπούς.

Η διασύνδεση αυτών των πηγών απαιτεί μηχανισμούς εικονικοποίησης δεδομένων, οι οποίοι επιτρέπουν τη δημιουργία ενός ενιαίου λογικού επιπέδου πρόσβασης. Μέσω αυτής της προσέγγισης, καθίσταται εφικτή η εκτέλεση σύνθετων ερωτημάτων που συνδυάζουν δεδομένα από πολλαπλές υπηρεσίες, χωρίς να απαιτείται συγχρονισμός ή αντιγραφή τους, ενισχύοντας την ευελιξία και μειώνοντας την καθυστέρηση στην ανάκτηση πληροφορίας[3], [5]. Η ανάγκη για ενοποίηση δεδομένων από εσωτερικά και εξωτερικά συστήματα, όπως πλατφόρμες e-learning ή διεθνείς ακαδημαϊκές συνεργασίες, δημιουργεί πρόσθετες προκλήσεις όσον αφορά τη συμβατότητα και τη διαλειτουργικότητα. Η ενσωμάτωση δεδομένων από intranet και extranet πηγές απαιτεί την υιοθέτηση προτύπων και μηχανισμών μετασχηματισμού που εξασφαλίζουν συνεπή σημασιολογική ερμηνεία, μειώνοντας τις ασυμβατότητες μεταξύ διαφορετικών σχημάτων δεδομένων. Η προσέγγιση αυτή ενισχύει την αξιοπιστία των αναλυτικών αποτελεσμάτων και επιτρέπει τη δημιουργία ολοκληρωμένων προφίλ φοιτητών που αντανακλούν το σύνολο των ακαδημαϊκών τους δραστηριοτήτων[8], [21].

Ταυτόχρονα, η επεξεργασία δεδομένων φοιτητών σε πραγματικό χρόνο καθίσταται κρίσιμη για την υποστήριξη υπηρεσιών όπως η προσωποποιημένη εκπαίδευση και η έγκαιρη ανίχνευση ακαδημαϊκών κινδύνων. Η αξιοποίηση σύγχρονων αρχιτεκτονικών, όπως τα lakehouse συστήματα, επιτρέπει τη συνδυασμένη χρήση αποθηκών δεδομένων και data lakes, παρέχοντας τη δυνατότητα ταυτόχρονης εκτέλεσης αναλυτικών και λειτουργικών φορτίων. Η προσέγγιση αυτή διευκολύνει την άμεση επεξεργασία μεγάλων όγκων δεδομένων, διατηρώντας παράλληλα την ακρίβεια και τη συνέπεια των πληροφοριών[11], [12]. Η διασφάλιση της ποιότητας και της διακυβέρνησης των δεδομένων, ιδιαίτερα σε περιβάλλοντα όπου τα δεδομένα φοιτητών χρησιμοποιούνται για τη λήψη αποφάσεων, είναι πολύ σημαντική. Η υιοθέτηση πρακτικών επιμέλειας και διαχείρισης δεδομένων επιτρέπει τη διατήρηση της ακρίβειας, της πληρότητας και της επικαιρότητας των πληροφοριών, ενώ παράλληλα ενισχύει τη συμμόρφωση με κανονιστικά πλαίσια προστασίας προσωπικών δεδομένων. Η ορθή διαχείριση των μεταδεδωμένων και η εφαρμογή πολιτικών πρόσβασης αποτελούν κρίσιμα στοιχεία για την ασφαλή και αποδοτική αξιοποίηση των δεδομένων σε πανεπιστημιακά περιβάλλοντα[22].

3.4.2 Βάσεις δεδομένων μαθημάτων

Η αποθήκευση αυτών των δεδομένων δεν περιορίζεται σε απλές σχεσιακές αναπαραστάσεις, αλλά επεκτείνεται σε πολυεπίπεδες δομές που επιτρέπουν τη μοντελοποίηση εξαρτήσεων μεταξύ μαθημάτων, όπως προαπαιτούμενα και αλυσίδες μαθησιακής προόδου. Στο πλαίσιο αυτό, η υιοθέτηση τεχνικών ενοποίησης δεδομένων καθίσταται κρίσιμη, καθώς τα δεδομένα μαθημάτων προέρχονται από διαφορετικά τμήματα και συστήματα διαχείρισης σπουδών, απαιτώντας συνεκτική και σημασιολογικά ορθή ερμηνεία[6], [21]. Η κατανομημένη φύση των πληροφοριακών συστημάτων εκπαίδευσης επιβάλλει τη χρήση αρχιτεκτονικών ομοσπονδίας δεδομένων, οι οποίες επιτρέπουν την ενοποιημένη πρόσβαση σε πολλαπλές βάσεις μαθημάτων χωρίς τη φυσική συγκέντρωση των δεδομένων. Μέσω αυτής της προσέγγισης, οι χρήστες μπορούν να εκτελούν σύνθετα ερωτήματα που διατρέχουν διαφορετικά συστήματα, όπως πλατφόρμες διαχείρισης μαθημάτων και αποθετήρια εκπαιδευτικού υλικού, διατηρώντας παράλληλα την αυτονομία κάθε πηγής. Η δυνατότητα αυτή ενισχύει τη διαλειτουργικότητα και επιτρέπει την αποτελεσματική αξιοποίηση ετερογενών δεδομένων σε περιβάλλοντα με υψηλό βαθμό πολυπλοκότητας[1], [2].

Η εφαρμογή τεχνικών εικονικοποίησης δεδομένων συμβάλλει στη δημιουργία ενός ενιαίου λογικού επιπέδου πρόσβασης, μέσω του οποίου οι βάσεις δεδομένων μαθημάτων μπορούν να παρουσιαστούν ως ένα ενοποιημένο σύνολο. Η προσέγγιση αυτή επιτρέπει την εκτέλεση ερωτημάτων σε πραγματικό χρόνο, χωρίς την ανάγκη αντιγραφής ή συγχρονισμού δεδομένων, μειώνοντας την καθυστέρηση και ενισχύοντας την αποδοτικότητα των συστημάτων. Επιπλέον, διευκολύνει την προσαρμογή σε αλλαγές των προγραμμάτων σπουδών, καθώς οι μεταβολές μπορούν να ενσωματωθούν δυναμικά χωρίς αναδιάρθρωση της υποκείμενης υποδομής[3], [5]. Η ενσωμάτωση δεδομένων από εσωτερικές και εξωτερικές πλατφόρμες, όπως συστήματα τηλεκπαίδευσης και διεθνείς συνεργασίες, απαιτεί μηχανισμούς μετασχηματισμού που διασφαλίζουν τη συμβατότητα διαφορετικών μορφών και σχημάτων δεδομένων. Η αξιοποίηση σημασιολογικών τεχνικών επιτρέπει τη χαρτογράφηση εννοιών μεταξύ διαφορετικών πηγών, εξασφαλίζοντας ότι οι πληροφορίες μαθημάτων παραμένουν συνεπείς και κατανοητές σε όλο το σύστημα. Η διαδικασία αυτή μειώνει τις ασάφειες και ενισχύει την αξιοπιστία των δεδομένων που χρησιμοποιούνται για ακαδημαϊκές και διοικητικές αποφάσεις[8], [21].

Ταυτόχρονα, η ανάγκη για ανάλυση δεδομένων μαθημάτων σε πραγματικό χρόνο οδηγεί στην υιοθέτηση σύγχρονων αρχιτεκτονικών, όπως τα lakehouse συστήματα, τα οποία συνδυάζουν τις δυνατότητες αποθήκευσης μεγάλων όγκων δεδομένων με προηγμένες αναλυτικές λειτουργίες. Μέσω αυτής της προσέγγισης, καθίσταται δυνατή η επεξεργασία δεδομένων που σχετίζονται με τη συμμετοχή

φοιτητών, την απόδοση μαθημάτων και τη χρήση εκπαιδευτικών πόρων, επιτρέποντας την εξαγωγή χρήσιμων συμπερασμάτων για τη βελτιστοποίηση των εκπαιδευτικών διαδικασιών[11], [12].

Πίνακας 3.5: Συγκριτική αποτύπωση χαρακτηριστικών βάσεων δεδομένων φοιτητών και μαθημάτων

Κατηγορία Χαρακτηριστικού	Βάσεις Δεδομένων Φοιτητών	Βάσεις Δεδομένων Μαθημάτων
Δομή δεδομένων	Ιεραρχική και πολυσυσχετιστική (προφίλ, εγγραφές, επιδόσεις)	Σχεσιακή με έμφαση σε σχέσεις μαθημάτων και προγραμμάτων σπουδών
Ετερογένεια πηγών	Υψηλή, λόγω πολλαπλών διοικητικών συστημάτων	Μέτρια έως υψηλή, λόγω διασύνδεσης με LMS και ακαδημαϊκές μονάδες
Ανάγκη ενοποίησης	Κρίσιμη για ενοποιημένη εικόνα φοιτητή	Απαραίτητη για συνοχή προγραμμάτων και μαθησιακών ροών
Δυνατότητα ομοσπονδίας	Εκτεταμένη για πρόσβαση σε καταναμημένα μητρώα	Υποστηρίζει σύνθετα ερωτήματα μεταξύ μαθημάτων και δομών
Χρήση σημασιολογίας	Εστίαση σε ταυτοποίηση και κατηγοριοποίηση φοιτητών	Έντονη χρήση για συσχέτιση εννοιών μαθημάτων
Επεξεργασία σε πραγματικό χρόνο	Σημαντική για ενημέρωση κατάστασης φοιτητών	Κρίσιμη για δυναμικά προγράμματα και διαθεσιμότητα μαθημάτων
Διακυβέρνηση δεδομένων	Έμφαση σε ακρίβεια, ιδιωτικότητα και κανονιστική συμμόρφωση	Έμφαση σε συνέπεια περιεχομένου και ακαδημαϊκή ορθότητα
Αναλυτικές δυνατότητες	Παρακολούθηση προόδου και επιδόσεων	Βελτιστοποίηση προγραμμάτων σπουδών και κατανομής πόρων

Πηγές:[1], [2], [3], [5], [6], [8], [11], [12], [21], [22]

3.4.3 Δεδομένα καθηγητών και διδασκόντων

Οι πληροφορίες αυτές περιλαμβάνουν όχι μόνο βασικά δημογραφικά στοιχεία και ακαδημαϊκούς τίτλους, αλλά και δυναμικά μεταβαλλόμενα δεδομένα όπως ωρολόγια προγράμματα, ερευνητική δραστηριότητα, συμμετοχή σε έργα και διασυνδέσεις με πολλαπλά εκπαιδευτικά συστήματα. Η εγγενής αυτή πολυδιάστατη φύση επιβάλλει την υιοθέτηση μοντέλων ενοποίησης που επιτρέπουν τη συνδυαστική αξιοποίηση ετερογενών πηγών, διατηρώντας παράλληλα τη σημασιολογική συνοχή των δεδομένων[6], [21]. Η καταναμημένη αποθήκευση των δεδομένων διδασκόντων σε διαφορετικά πληροφοριακά υποσυστήματα, όπως πλατφόρμες διαχείρισης προσωπικού, ερευνητικά αποθετήρια και συστήματα τηλεεκπαίδευσης, καθιστά αναγκαία τη χρήση ομοσπονδιακών αρχιτεκτονικών δεδομένων. Μέσω τέτοιων προσεγγίσεων, επιτυγχάνεται ενοποιημένη πρόσβαση χωρίς τη φυσική συγκέντρωση των δεδομένων, επιτρέποντας την εκτέλεση σύνθετων ερωτημάτων που διατρέχουν πολλαπλές πηγές. Η λειτουργικότητα αυτή ενισχύει τη διαλειτουργικότητα και υποστηρίζει την αποδοτική διαχείριση των πληροφοριών που σχετίζονται με το διδακτικό έργο και τη διοικητική οργάνωση[1], [2].

Παράλληλα, η αξιοποίηση τεχνικών εικονικοποίησης δεδομένων επιτρέπει τη δημιουργία ενός ενιαίου λογικού επιπέδου, μέσω του οποίου τα δεδομένα καθηγητών παρουσιάζονται ως ένα συνεκτικό σύνολο, ανεξαρτήτως της φυσικής τους τοποθεσίας. Η προσέγγιση αυτή μειώνει την ανάγκη για αντιγραφή δεδομένων και επιτρέπει την άμεση πρόσβαση σε επικαιροποιημένες πληροφορίες, γεγονός ιδιαίτερα σημαντικό σε περιβάλλοντα όπου οι μεταβολές είναι συχνές, όπως οι αναθέσεις μαθημάτων ή οι ερευνητικές συνεργασίες. Επιπλέον, διευκολύνει την προσαρμογή σε αλλαγές των οργανωτικών δομών, χωρίς να απαιτείται εκτεταμένη αναδιάρθρωση της υποδομής[3], [5]. Η σημασιολογική ενοποίηση

επιτρέπει τη χαρτογράφηση εννοιών όπως «διδάσκων», «ερευνητής» ή «επιστημονικός υπεύθυνος», διασφαλίζοντας ότι οι πληροφορίες παραμένουν συνεπείς και κατανοητές σε όλα τα επίπεδα του συστήματος. Με τον τρόπο αυτό, μειώνονται οι ασάφειες και ενισχύεται η αξιοπιστία των δεδομένων, γεγονός που είναι κρίσιμο για τη λήψη διοικητικών και ακαδημαϊκών αποφάσεων[9], [21].

Ταυτόχρονα, η ενσωμάτωση δεδομένων από εσωτερικά και εξωτερικά δίκτυα, όπως διεθνείς συνεργασίες και διαπανεπιστημιακά έργα, απαιτεί μηχανισμούς που υποστηρίζουν τη συμβατότητα διαφορετικών μορφών και προτύπων. Η διασύνδεση intranet και extranet πηγών ενισχύει την πληρότητα των πληροφοριών, επιτρέποντας τη συνολική αποτύπωση της δραστηριότητας των διδασκόντων. Παράλληλα, η χρήση σύγχρονων υποδομών επιτρέπει την επεξεργασία δεδομένων σε πραγματικό χρόνο, υποστηρίζοντας δυναμικές εφαρμογές όπως η κατανομή διδακτικού έργου και η αξιολόγηση απόδοσης[8].

3.4.4 Πρόγραμμα εξεταστικής

Η οργάνωση και η διαχείριση των δεδομένων που αφορούν το πρόγραμμα εξεταστικής σε πανεπιστημιακά περιβάλλοντα συνιστούν ένα σύνθετο πρόβλημα ολοκλήρωσης, καθώς τα σχετικά δεδομένα προέρχονται από πολλαπλές λειτουργικές οντότητες, όπως γραμματείες τμημάτων, συστήματα δηλώσεων μαθημάτων και πλατφόρμες διαχείρισης αιθουσών. Η πολυπλοκότητα αυτή εντείνεται από την ανάγκη συγχρονισμού χρονικών περιορισμών, χωρητικότητας υποδομών και διαθεσιμότητας διδακτικού προσωπικού, οδηγώντας σε ένα δυναμικό σύστημα που απαιτεί ακριβή και συνεπή ροή πληροφοριών. Η ενσωμάτωση αυτών των ετερογενών πηγών προϋποθέτει μηχανισμούς που επιτρέπουν τη διατήρηση της σημασιολογικής ακεραιότητας, ιδίως όταν διαφορετικά συστήματα χρησιμοποιούν αποκλίνουσες αναπαραστάσεις για έννοιες όπως «εξέταση», «μάθημα» ή «χρονική ζώνη»[1], [2]. Η αξιοποίηση ομοσπονδιακών μοντέλων δεδομένων καθίσταται καθοριστική για την ενοποιημένη διαχείριση του προγράμματος εξεταστικής, καθώς επιτρέπει την εκτέλεση σύνθετων ερωτημάτων που αντλούν πληροφορίες από κατακευματισμένες πηγές χωρίς την ανάγκη φυσικής συγκέντρωσης των δεδομένων.

Μέσω αυτής της αρχιτεκτονικής, είναι εφικτή η δυναμική σύνθεση προγραμμάτων εξετάσεων που λαμβάνουν υπόψη περιορισμούς όπως επικαλύψεις μαθημάτων ή συγκρούσεις αιθουσών, ενώ παράλληλα διασφαλίζεται η επικαιρότητα των δεδομένων. Η προσέγγιση αυτή ευθυγραμμίζεται με τις σύγχρονες τάσεις ενοποίησης δεδομένων σε data spaces, όπου η πρόσβαση σε πληροφορίες επιτυγχάνεται μέσω ενοποιημένων διεπαφών χωρίς απώλεια αυτονομίας των επιμέρους συστημάτων[1], [2]. Η εικονικοποίηση δεδομένων επιτρέπει τη δημιουργία ενός λογικού επιπέδου στο οποίο οι πληροφορίες της εξεταστικής παρουσιάζονται ως ενιαίο σύνολο, ανεξάρτητα από την προέλευσή τους. Με τον τρόπο αυτό, οι αλλαγές που πραγματοποιούνται σε επιμέρους υποσυστήματα, όπως η τροποποίηση ημερομηνιών ή η ανακατανομή αιθουσών, αντανakλώνται άμεσα στο συνολικό πρόγραμμα, χωρίς να απαιτείται επανασχεδιασμός της υποδομής. Η δυνατότητα αυτή είναι κρίσιμη για τη διαχείριση απρόβλεπτων καταστάσεων, όπως έκτακτες αλλαγές ή αυξημένη ζήτηση για συγκεκριμένες χρονικές περιόδους[3], [5].

Η σύνδεση intranet και extranet πηγών διευκολύνει την ανταλλαγή πληροφοριών και την εναρμόνιση διαφορετικών κανονιστικών πλαισίων, ενώ ταυτόχρονα ενισχύει τη διαφάνεια και την προσβασιμότητα των δεδομένων. Σε αυτό το πλαίσιο, η υποστήριξη πραγματικού χρόνου καθίσταται απαραίτητη για την άμεση ενημέρωση όλων των εμπλεκόμενων μερών, από φοιτητές έως διοικητικό προσωπικό[8]. Η εφαρμογή σημασιολογικών τεχνικών επιτρέπει τη συσχέτιση δεδομένων που προέρχονται από διαφορετικά συστήματα, διασφαλίζοντας ότι οι έννοιες παραμένουν συνεπείς και κατανοητές. Η χρήση

σημασιολογικών μοντέλων συμβάλλει στην αποφυγή ασυμβατοτήτων και στη βελτίωση της ποιότητας των δεδομένων, γεγονός που είναι ιδιαίτερα σημαντικό για την ορθή κατάρτιση του προγράμματος εξετάσεων. Επιπλέον, διευκολύνει την ανάπτυξη προηγμένων λειτουργιών, όπως η αυτόματη ανίχνευση συγκρούσεων ή η βελτιστοποίηση της κατανομής πόρων[9], [21].

3.4.5 Στατιστικά αξιολογήσεων

Η συλλογή και η ανάλυση στατιστικών αξιολογήσεων στο πανεπιστημιακό περιβάλλον συγκροτούν ένα πολυεπίπεδο οικοσύστημα δεδομένων, στο οποίο συνυπάρχουν ποσοτικές μετρήσεις, ποιοτικές παρατηρήσεις και μεταδεδομένα που σχετίζονται με τη διδακτική διαδικασία. Η ιδιαιτερότητα αυτών των δεδομένων έγκειται στο γεγονός ότι παράγονται από διαφορετικές πλατφόρμες, όπως συστήματα ηλεκτρονικής μάθησης, εσωτερικά πληροφοριακά συστήματα και εξωτερικές εφαρμογές αξιολόγησης, γεγονός που δημιουργεί έντονη ετερογένεια ως προς τη δομή, τη μορφή και τη σημασιολογία τους. Η αποτελεσματική ενοποίηση αυτών των πηγών απαιτεί προηγμένες τεχνικές σημασιολογικής ολοκλήρωσης, ώστε να καταστεί δυνατή η συνεκτική ερμηνεία των αποτελεσμάτων και η αποφυγή ασυμβατοτήτων που προκύπτουν από διαφορετικά μοντέλα δεδομένων[6], [21].

Υπό διαφορετική οπτική, η αξιοποίηση ομοσπονδιακών μηχανισμών πρόσβασης επιτρέπει τη διαχείριση των στατιστικών αξιολογήσεων χωρίς την ανάγκη συγκέντρωσης των δεδομένων σε ένα κεντρικό αποθετήριο. Μέσω της ομοσπονδιοποίησης, καθίσταται δυνατή η εκτέλεση ερωτημάτων που συνδυάζουν δεδομένα από πολλαπλές πηγές, διατηρώντας παράλληλα την αυτονομία και τις πολιτικές πρόσβασης κάθε συστήματος. Η προσέγγιση αυτή ενισχύει την ευελιξία και την επεκτασιμότητα των πληροφοριακών υποδομών, ενώ επιτρέπει την άμεση ενσωμάτωση νέων πηγών δεδομένων, όπως αναδυόμενα εργαλεία αξιολόγησης ή πλατφόρμες ανατροφοδότησης[1], [2], [3]. Η εφαρμογή τεχνικών εικονικοποίησης δεδομένων συμβάλλει στη δημιουργία ενός ενοποιημένου λογικού επιπέδου, όπου οι στατιστικές αξιολογήσεις παρουσιάζονται ως συνεκτικό σύνολο πληροφοριών. Σε αυτό το επίπεδο, οι διαφορές μεταξύ των υποκείμενων συστημάτων αποκρύπτονται, επιτρέποντας στους αναλυτές να επικεντρωθούν στην εξαγωγή γνώσης αντί στη διαχείριση της πολυπλοκότητας. Επιπλέον, η δυνατότητα δυναμικής ενημέρωσης των δεδομένων διασφαλίζει ότι οι αναλύσεις βασίζονται σε επικαιροποιημένες πληροφορίες, γεγονός που είναι κρίσιμο για τη λήψη τεκμηριωμένων αποφάσεων[5], [27].

Η σύνδεση intranet και extranet πηγών επιτρέπει την ολοκληρωμένη αποτύπωση της εκπαιδευτικής εμπειρίας, ενώ παράλληλα διευκολύνει τη συγκριτική ανάλυση μεταξύ διαφορετικών ιδρυμάτων ή τμημάτων. Η υποστήριξη αυτής της διασύνδεσης απαιτεί μηχανισμούς που διασφαλίζουν την ασφάλεια και την ακεραιότητα των δεδομένων, ιδιαίτερα όταν αυτά περιλαμβάνουν ευαίσθητες πληροφορίες[8]. Η αξιοποίηση σημασιολογικών τεχνολογιών επιτρέπει τη βαθύτερη κατανόηση των δεδομένων αξιολόγησης, μέσω της συσχέτισης εννοιών και της δημιουργίας πλούσιων μοντέλων γνώσης. Η χρήση σημασιολογικών γραφημάτων και τεχνικών ενοποίησης διευκολύνει την ανάλυση σύνθετων μοτίβων, όπως η συσχέτιση μεταξύ διδακτικών πρακτικών και μαθησιακών αποτελεσμάτων. Παράλληλα, ενισχύεται η δυνατότητα ανάπτυξης προηγμένων εφαρμογών, όπως συστήματα υποστήριξης αποφάσεων ή εργαλεία πρόβλεψης[9], [21].

3.4.6 Πιθανές εξωτερικές ή βοηθητικές πηγές

Εξωτερικές πηγές όπως δημόσια αποθετήρια δεδομένων, διεθνείς εκπαιδευτικές πλατφόρμες και συστήματα αξιολόγησης τρίτων φορέων λειτουργούν συμπληρωματικά προς τις εσωτερικές δομές, προσφέροντας πρόσθετη πληροφοριακή αξία. Η ενσωμάτωση αυτών των πηγών δεν περιορίζεται σε

απλή μεταφορά δεδομένων, αλλά απαιτεί τη διαμόρφωση μηχανισμών που διασφαλίζουν τη σημασιολογική συμβατότητα και την ομοιογενή ερμηνεία των πληροφοριών, ιδιαίτερα όταν τα δεδομένα προέρχονται από διαφορετικά πρότυπα και μορφές αναπαράστασης[6], [9]. Από μια διαφορετική σκοπιά, η υιοθέτηση ομοσπονδιακών προσεγγίσεων επιτρέπει την πρόσβαση σε εξωτερικά δεδομένα χωρίς την ανάγκη φυσικής ενοποίησής τους σε ένα ενιαίο αποθετήριο. Μέσω της ομοσπονδιοποίησης, καθίσταται δυνατή η εκτέλεση σύνθετων ερωτημάτων που συνδυάζουν εσωτερικά και εξωτερικά δεδομένα, διατηρώντας παράλληλα την αυτονομία και τις πολιτικές πρόσβασης κάθε πηγής. Η προσέγγιση αυτή ενισχύει τη δυναμικότητα των πανεπιστημιακών πληροφοριακών συστημάτων, καθώς επιτρέπει την άμεση ενσωμάτωση νέων πηγών χωρίς σημαντικές τροποποιήσεις στην υποδομή[1], [2], [3].

Η εικονικοποίηση δεδομένων διαδραματίζει κρίσιμο ρόλο στη διαχείριση των εξωτερικών πηγών, δημιουργώντας ένα αφηρημένο επίπεδο όπου οι πληροφορίες παρουσιάζονται ως ενιαίο σύνολο. Σε αυτό το πλαίσιο, οι διαφορές μεταξύ των επιμέρους συστημάτων αποκρύπτονται, επιτρέποντας την ομαλή αλληλεπίδραση μεταξύ ετερογενών δεδομένων. Η δυνατότητα δυναμικής ενημέρωσης των δεδομένων εξασφαλίζει ότι οι πληροφορίες που αντλούνται από εξωτερικές πηγές παραμένουν επικαιροποιημένες, γεγονός που είναι ιδιαίτερα σημαντικό για εφαρμογές που απαιτούν άμεση πληροφόρηση[5], [27].

Ταυτόχρονα, η χρήση σημασιολογικών τεχνολογιών επιτρέπει τη συσχέτιση και την ερμηνεία δεδομένων που προέρχονται από διαφορετικά περιβάλλοντα. Μέσω της ανάπτυξης σημασιολογικών μοντέλων και γνώσεων, καθίσταται δυνατή η ενοποίηση εννοιών και η αποφυγή ασυμβατοτήτων, ενισχύοντας τη συνοχή των δεδομένων. Επιπλέον, οι τεχνικές αυτές υποστηρίζουν την ανάπτυξη εφαρμογών που αξιοποιούν εξωτερικά δεδομένα για την παραγωγή γνώσης, όπως συστήματα πρόβλεψης ή εργαλεία ανάλυσης τάσεων[9], [21].

3.5 Τεχνολογίες υποστήριξης της υλοποίησης

3.5.1 Συστήματα βάσεων δεδομένων

Σε αυτό το περιβάλλον, τα σχεσιακά συστήματα βάσεων δεδομένων εξακολουθούν να διαδραματίζουν κεντρικό ρόλο, παρέχοντας αυστηρή τυποποίηση, ακεραιότητα και συνέπεια μέσω της χρήσης δομημένων σχημάτων και της γλώσσας SQL. Παράλληλα, η εξέλιξη των απαιτήσεων για επεξεργασία μεγάλων δεδομένων έχει οδηγήσει στην ενσωμάτωση νέων αρχιτεκτονικών προσεγγίσεων που επιτρέπουν την αποδοτική διαχείριση μη δομημένων και ημιδομημένων δεδομένων, χωρίς να θυσιάζεται η δυνατότητα εκτέλεσης σύνθετων ερωτημάτων[15], [30]. Η εμφάνιση των NoSQL βάσεων δεδομένων εισάγει ένα διαφορετικό παράδειγμα αποθήκευσης, στο οποίο δίνεται έμφαση στην οριζόντια κλιμάκωση και την ευελιξία των σχημάτων. Τα συστήματα αυτά επιτρέπουν τη διαχείριση δεδομένων σε μορφές όπως έγγραφα, γράφοι και κλειδί-τιμή, διευκολύνοντας την αποθήκευση πληροφοριών που δεν συμμορφώνονται με αυστηρά προκαθορισμένα σχήματα. Παρά τα πλεονεκτήματα αυτά, η έλλειψη εγγενούς υποστήριξης για σύνθετες συναλλαγές και η μειωμένη συνέπεια σε ορισμένα σενάρια καθιστούν αναγκαία τη συνδυαστική χρήση τους με παραδοσιακά σχεσιακά συστήματα, οδηγώντας στην ανάπτυξη υβριδικών μοντέλων βάσεων δεδομένων [30].

Από μια πιο εξελιγμένη οπτική, τα συστήματα lakehouse αποτελούν μια σύγχρονη αρχιτεκτονική προσέγγιση που συνδυάζει τα πλεονεκτήματα των data lakes και των data warehouses. Μέσω αυτής της σύγκλισης, καθίσταται δυνατή η αποθήκευση μεγάλων όγκων δεδομένων σε ακατέργαστη μορφή, ενώ παράλληλα υποστηρίζονται προηγμένες λειτουργίες ανάλυσης και επεξεργασίας. Η χρήση τεχνολογιών

όπως Delta Lake, Apache Iceberg και Hudi ενισχύει την αξιοπιστία των δεδομένων, προσφέροντας δυνατότητες όπως versioning, ACID συναλλαγές και βελτιστοποίηση της απόδοσης των ερωτημάτων[12], [13], [14]. Η υιοθέτηση ομοσπονδιακών συστημάτων βάσεων δεδομένων επιτρέπει την ενοποιημένη πρόσβαση σε πολλαπλές πηγές δεδομένων χωρίς την ανάγκη φυσικής ενοποίησης. Τα συστήματα αυτά λειτουργούν ως ενδιάμεσο επίπεδο που διασυνδέει διαφορετικές βάσεις δεδομένων, επιτρέποντας την εκτέλεση κατανεμημένων ερωτημάτων και τη συνδυαστική αξιοποίηση δεδομένων από ετερογενή περιβάλλοντα. Η προσέγγιση αυτή είναι ιδιαίτερα χρήσιμη σε περιβάλλοντα όπου τα δεδομένα είναι κατανεμημένα σε διαφορετικούς οργανισμούς ή πλατφόρμες, καθώς διασφαλίζει την αυτονομία των επιμέρους συστημάτων και την ευελιξία στη διαχείριση[1], [2], [3].

Πίνακας 3.6: Κατηγοριοποίηση και χαρακτηριστικά συστημάτων βάσεων δεδομένων σε σύγχρονες αρχιτεκτονικές

Κατηγορία Συστήματος	Βασικά χαρακτηριστικά	Αρχιτεκτονική προσέγγιση	Πλεονεκτήματα	Περιορισμοί
Σχεσιακές Βάσεις Δεδομένων (RDBMS)	Δομημένα σχήματα, ACID ιδιότητες, SQL	Κεντριοποιημένη ή κατανεμημένη	Υψηλή συνέπεια, ισχυρή μοντελοποίηση	Περιορισμένη ευελιξία σε μη δομημένα δεδομένα
NoSQL Συστήματα	Schema-less, key-value, document, graph μοντέλα	Οριζόντια κλιμάκωση	Ευελιξία και υψηλή διαθεσιμότητα	Ασθενέστερη συνέπεια σε ορισμένες περιπτώσεις
Lakehouse Αρχιτεκτονικές	Συνδυασμός data lake και warehouse, ACID σε μεγάλα δεδομένα	Κατανεμημένα αποθηκευτικά συστήματα	Ενοποιημένη ανάλυση και αποθήκευση	Πολυπλοκότητα υλοποίησης
Ομοσπονδιακά Συστήματα	Ενοποίηση πολλαπλών πηγών χωρίς μεταφορά δεδομένων	Middleware επίπεδο	Αυτονομία πηγών και ενοποιημένη πρόσβαση	Εξάρτηση από latency δικτύου
Εικονικοποίηση Δεδομένων	Λογικό επίπεδο ενοποίησης, πρόσβαση σε πραγματικό χρόνο	Virtual layer πάνω από πηγές	Μείωση αντιγραφής δεδομένων	Περιορισμοί απόδοσης σε μεγάλα workloads
Serverless DB Συστήματα	Δυναμική κατανομή πόρων, pay-per-use	Cloud-native αρχιτεκτονική	Ελαστικότητα και χαμηλό κόστος	Μεταβλητή απόδοση

Πηγές:[1], [2], [3], [7], [10], [12], [13], [14], [15], [28], [30]

3.5.2 SQL και federated queries

Η χρήση της SQL σε περιβάλλοντα federated queries επιτρέπει την εκτέλεση ενιαίων ερωτημάτων πάνω σε πολλαπλές, ετερογενείς πηγές δεδομένων χωρίς την ανάγκη φυσικής ενοποίησης τους. Η δυνατότητα αυτή βασίζεται σε ενδιάμεσα επίπεδα αφαίρεσης που μεταφράζουν τα ερωτήματα SQL σε επιμέρους υποερωτήματα, τα οποία εκτελούνται τοπικά στις αντίστοιχες πηγές και στη συνέχεια συνδυάζονται σε ένα ενοποιημένο αποτέλεσμα[1], [2]. Η έννοια των federated queries σχετίζεται άμεσα με τις αρχές της εικονικοποίησης δεδομένων, όπου η SQL λειτουργεί ως κοινή διεπαφή ανεξάρτητα από τη φυσική τοποθεσία και τη μορφή των δεδομένων. Σε αυτό το πλαίσιο, οι μηχανισμοί βελτιστοποίησης ερωτημάτων αποκτούν ιδιαίτερη σημασία, καθώς απαιτείται η αποδοτική κατανομή των υποερωτημάτων και η ελαχιστοποίηση της μεταφοράς δεδομένων μέσω δικτύου. Η διαδικασία αυτή περιλαμβάνει τεχνικές όπως predicate pushdown, cost-based optimization και adaptive query planning,

οι οποίες διασφαλίζουν ότι η εκτέλεση των ερωτημάτων παραμένει αποδοτική ακόμη και σε περιβάλλοντα μεγάλης κλίμακας[2], [3].

Παράλληλα, η ενσωμάτωση της SQL σε περιβάλλοντα data lakehouse ενισχύει περαιτέρω τη λειτουργικότητα των federated queries, επιτρέποντας την άμεση πρόσβαση σε δεδομένα που αποθηκεύονται σε διαφορετικά επίπεδα αποθήκευσης. Μέσω αυτής της σύγκλισης, καθίσταται εφικτή η εκτέλεση αναλυτικών ερωτημάτων υψηλής πολυπλοκότητας χωρίς την ανάγκη μεταφοράς των δεδομένων σε κεντρικά αποθετήρια. Επιπλέον, η υποστήριξη ACID ιδιοτήτων σε σύγχρονες lakehouse πλατφόρμες συμβάλλει στη διατήρηση της συνέπειας, ακόμη και όταν τα δεδομένα προέρχονται από πολλαπλές πηγές[11], [12]. Η υλοποίηση federated queries σε περιβάλλοντα cloud και serverless αρχιτεκτονικών εισάγει νέες δυνατότητες ως προς την κλιμάκωση και την αποδοτικότητα. Η δυναμική κατανομή υπολογιστικών πόρων επιτρέπει την εκτέλεση πολύπλοκων ερωτημάτων σε πραγματικό χρόνο, ενώ παράλληλα μειώνει το λειτουργικό κόστος μέσω της χρήσης μοντέλων pay-as-you-go. Η προσέγγιση αυτή καθιστά δυνατή την επεξεργασία δεδομένων από πολλαπλές πηγές με υψηλή ταχύτητα και αξιοπιστία, ιδιαίτερα σε περιβάλλοντα με μεταβαλλόμενο φόρτο εργασίας[10], [28]. Η εφαρμογή της SQL σε ομοσπονδιακά περιβάλλοντα συνδέεται στενά με τη διαχείριση ετερογενών δεδομένων και την ανάγκη σημασιολογικής ενοποίησης. Η χρήση ενδιάμεσων μοντέλων, όπως τα knowledge graphs και τα semantic layers, επιτρέπει την αντιστοίχιση διαφορετικών σχημάτων δεδομένων σε ένα κοινό εννοιολογικό πλαίσιο, διευκολύνοντας την εκτέλεση σύνθετων ερωτημάτων. Η προσέγγιση αυτή ενισχύει τη διαλειτουργικότητα και καθιστά δυνατή την αξιοποίηση δεδομένων από διαφορετικά πληροφοριακά συστήματα με συνεκτικό τρόπο[9], [21].

3.5.3 Εργαλεία διαχείρισης και παρακολούθησης

Πλατφόρμες όπως το **Apache Airflow** χρησιμοποιούνται εκτενώς για τον ορισμό, τον προγραμματισμό και την παρακολούθηση ροών εργασίας δεδομένων, επιτρέποντας τον ακριβή έλεγχο της εκτέλεσης σύνθετων διεργασιών σε περιβάλλοντα πολλαπλών πηγών. Η αρχιτεκτονική του βασίζεται σε Directed Acyclic Graphs (DAGs), προσφέροντας υψηλή ευελιξία και επεκτασιμότητα, στοιχεία που καθίστανται κρίσιμα σε περιβάλλοντα ομοσπονδιακής ολοκλήρωσης δεδομένων[1], [5]. Συστήματα όπως το **Apache Kafka** λειτουργούν ως βασικός μηχανισμός διαχείρισης ροών δεδομένων σε πραγματικό χρόνο, παρέχοντας δυνατότητες υψηλής απόδοσης για ingestion και streaming δεδομένων. Μέσω της κατανομημένης αρχιτεκτονικής του, το Kafka επιτρέπει τη συνεχή παρακολούθηση γεγονότων και την αξιόπιστη μεταφορά δεδομένων μεταξύ διαφορετικών συστημάτων, γεγονός που ενισχύει τη διαλειτουργικότητα σε ετερογενή περιβάλλοντα[8], [15]. Επιπλέον, η ενσωμάτωσή του με εργαλεία επεξεργασίας όπως το **Apache Flink** και το **Apache Spark Structured Streaming** επιτρέπει την υλοποίηση προηγμένων αναλυτικών διαδικασιών σε πραγματικό χρόνο, βελτιώνοντας την απόκριση των συστημάτων σε δυναμικά δεδομένα.

Από διαφορετική οπτική, η παρακολούθηση της απόδοσης και της κατάστασης των συστημάτων επιτυγχάνεται μέσω εξειδικευμένων εργαλείων όπως το **Prometheus** και το **Grafana**. Το Prometheus συλλέγει μετρικές από διαφορετικές πηγές, ενώ το Grafana παρέχει προηγμένες δυνατότητες οπτικοποίησης, επιτρέποντας την άμεση κατανόηση της συμπεριφοράς του συστήματος. Η συνδυαστική χρήση αυτών των εργαλείων καθιστά δυνατή την έγκαιρη ανίχνευση ανωμαλιών και τη βελτιστοποίηση της λειτουργίας των υποδομών δεδομένων, ιδιαίτερα σε περιβάλλοντα cloud και serverless[10], [28]. Εργαλεία όπως το **Kubernetes** και το **Docker** συμβάλλουν στη διαχείριση και την ορχήστρωση των υπολογιστικών πόρων, επιτρέποντας την αποδοτική ανάπτυξη και κλιμάκωση εφαρμογών δεδομένων. Το Kubernetes, ειδικότερα, παρέχει μηχανισμούς αυτοματοποιημένης

διαχείρισης κοντέινερ, εξασφαλίζοντας υψηλή διαθεσιμότητα και ανθεκτικότητα των υπηρεσιών. Η αξιοποίησή του σε συνδυασμό με serverless υποδομές ενισχύει την ευελιξία και μειώνει το λειτουργικό κόστος, όπως καταδεικνύεται σε σύγχρονες αρχιτεκτονικές επεξεργασίας δεδομένων[27], [28].

Παράλληλα, η διαχείριση μεταδεδομένων και η εξασφάλιση της ποιότητας των δεδομένων υλοποιούνται μέσω εργαλείων όπως το **Apache Atlas** και το **DataHub**, τα οποία επιτρέπουν την ιχνηλασιμότητα και τη διακυβέρνηση δεδομένων. Η ύπαρξη αυτών των μηχανισμών καθίσταται κρίσιμη σε περιβάλλοντα όπου τα δεδομένα προέρχονται από πολλαπλές και ετερογενείς πηγές, καθώς διασφαλίζει τη συνέπεια και την αξιοπιστία των πληροφοριών[21], [22]. Τέλος, εργαλεία παρακολούθησης ασφάλειας και συμμόρφωσης, όπως το **Elastic Stack (ELK)** και εξειδικευμένες πλατφόρμες κυβερνοασφάλειας, ενισχύουν την προστασία των δεδομένων μέσω ανάλυσης καταγραφών και ανίχνευσης απειλών σε πραγματικό χρόνο. Η ενσωμάτωσή τους σε ολοκληρωμένα οικοσυστήματα δεδομένων επιτρέπει την παρακολούθηση της δραστηριότητας και τη διασφάλιση της ιδιωτικότητας, ιδιαίτερα σε περιβάλλοντα όπου η ανταλλαγή δεδομένων πραγματοποιείται μέσω ομοσπονδιακών μηχανισμών[16], [29].

Πίνακας 3.7: Βασικά εργαλεία διαχείρισης και παρακολούθησης δεδομένων σε σύγχρονες αρχιτεκτονικές

<i>Εργαλείο</i>	<i>Κατηγορία</i>	<i>Λειτουργία</i>	<i>Ρόλος σε ομοσπονδιακά περιβάλλοντα</i>
Apache Airflow	Orchestration	Διαχείριση ροών εργασίας (DAGs)	Συντονισμός κατανεμημένων pipelines
Apache Kafka	Data Streaming	Μεταφορά και επεξεργασία δεδομένων σε πραγματικό χρόνο	Ενοποίηση ετερογενών πηγών
Apache Spark	Data Processing	Κατανεμημένη ανάλυση μεγάλων δεδομένων	Εκτέλεση federated analytics
Prometheus	Monitoring	Συλλογή μετρικών συστήματος	Ανίχνευση ανωμαλιών
Grafana	Visualization	Οπτικοποίηση δεδομένων και πίνακες ελέγχου	Παρακολούθηση απόδοσης
Kubernetes	Orchestration	Διαχείριση κοντέινερ	Κλιμάκωση και ανθεκτικότητα
Docker	Containerization	Απομόνωση εφαρμογών	Φορητότητα υπηρεσιών
Apache Atlas	Metadata Management	Διαχείριση μεταδεδομένων	Data lineage και governance
DataHub	Data Governance	Παρακολούθηση και καταγραφή δεδομένων	Διαφάνεια δεδομένων
Elastic Stack (ELK)	Logging & Security	Ανάλυση logs και ασφάλεια	Ανίχνευση απειλών

Πηγές:[1], [5], [8], [10], [15], [16], [21], [22], [27], [28], [29]

3.5.4 Docker ή εικονικοποίηση

Το Docker επιτρέπει τη δημιουργία απομονωμένων περιβαλλόντων εκτέλεσης μέσω containers, τα οποία ενσωματώνουν όλες τις απαραίτητες εξαρτήσεις λογισμικού, εξασφαλίζοντας συνεπή συμπεριφορά ανεξαρτήτως του υποκείμενου συστήματος. Η ιδιότητα αυτή καθίσταται ιδιαίτερα σημαντική σε περιβάλλοντα ομοσπονδιακής ενοποίησης δεδομένων, όπου απαιτείται η διαλειτουργικότητα μεταξύ πολλαπλών πλατφορμών και τεχνολογικών στοίβων[1], [5]. Η υιοθέτηση container-based προσεγγίσεων διευκολύνει την ανάπτυξη μικροϋπηρεσιών (microservices), επιτρέποντας την ανεξάρτητη ανάπτυξη, ενημέρωση και κλιμάκωση επιμέρους λειτουργικών μονάδων.

Σε αντίθεση με τις παραδοσιακές μεθόδους εικονικοποίησης μέσω εικονικών μηχανών (virtual machines), τα containers παρουσιάζουν σημαντικά μειωμένο overhead, καθώς μοιράζονται τον ίδιο πυρήνα λειτουργικού συστήματος. Αυτό οδηγεί σε βελτιωμένη απόδοση και αποδοτικότερη αξιοποίηση πόρων, στοιχεία που καθίστανται κρίσιμα σε περιβάλλοντα επεξεργασίας μεγάλων δεδομένων και πραγματικού χρόνου[10], [28].

Επιπλέον, η συνδυαστική χρήση του Docker με εργαλεία ορχήστρωσης όπως το Kubernetes επιτρέπει την αυτοματοποιημένη διαχείριση κύκλου ζωής των εφαρμογών, συμπεριλαμβανομένης της ανάπτυξης, της κλιμάκωσης και της ανάκαμψης από σφάλματα. Η δυναμική αυτή προσέγγιση υποστηρίζει την υλοποίηση ελαστικών αρχιτεκτονικών, οι οποίες μπορούν να προσαρμόζονται στις μεταβαλλόμενες απαιτήσεις φόρτου εργασίας, ιδιαίτερα σε cloud και serverless περιβάλλοντα[27], [28]. Μέσω αυτής της υποδομής, καθίσταται εφικτή η εκτέλεση κατανεμημένων queries και αναλυτικών διεργασιών σε πολλαπλές πηγές δεδομένων, ενισχύοντας τη λειτουργικότητα των ομοσπονδιακών συστημάτων[2], [3]. Η εικονικοποίηση συμβάλλει στη βελτίωση της ασφάλειας και της απομόνωσης των δεδομένων, καθώς κάθε container λειτουργεί ως ανεξάρτητη οντότητα με σαφώς καθορισμένα όρια πρόσβασης. Αυτό είναι ιδιαίτερα σημαντικό σε περιβάλλοντα data spaces, όπου η ανταλλαγή δεδομένων απαιτεί αυστηρούς μηχανισμούς ελέγχου πρόσβασης και διασφάλισης εμπιστευτικότητας[4], [19]. Παράλληλα, η δυνατότητα ταχείας αναπαραγωγής και μεταφοράς containers ενισχύει τη διαθεσιμότητα και τη συνέπεια των υπηρεσιών, διευκολύνοντας τη διαχείριση σύνθετων οικοσυστημάτων δεδομένων[6], [7].

Επιπροσθέτως, η ενσωμάτωση τεχνολογιών εικονικοποίησης σε αρχιτεκτονικές lakehouse και υβριδικών βάσεων δεδομένων επιτρέπει την ευέλικτη διαχείριση τόσο δομημένων όσο και αδόμητων δεδομένων. Μέσω της απομόνωσης υπολογιστικών πόρων και της ανεξαρτησίας των επιπέδων αποθήκευσης και επεξεργασίας, επιτυγχάνεται βελτιστοποίηση της απόδοσης και της επεκτασιμότητας, ιδίως σε περιβάλλοντα που απαιτούν αναλυτική επεξεργασία σε πραγματικό χρόνο[11], [12], [30]. Επιπλέον, η χρήση σύγχρονων μορφών αποθήκευσης, όπως το Apache Iceberg και το Delta Lake, σε συνδυασμό με containerized υποδομές, επιτρέπει την αποδοτική διαχείριση μεγάλων όγκων δεδομένων και τη βελτίωση της απόκρισης των ερωτημάτων[13], [14].

3.5.5 Εργαλεία δοκιμών και benchmarking

Εργαλεία όπως το **TPC-H** και το **TPC-DS** χρησιμοποιούνται εκτενώς για την προσομοίωση αναλυτικών φορτίων εργασίας σε αποθηκευτικά συστήματα και lakehouse περιβάλλοντα, παρέχοντας τυποποιημένα σενάρια ερωτημάτων που αποτυπώνουν πραγματικές επιχειρησιακές απαιτήσεις. Η αξιοποίησή τους επιτρέπει τη συγκριτική αποτίμηση διαφορετικών τεχνολογιών αποθήκευσης και επεξεργασίας, ιδίως σε περιπτώσεις όπου συνδυάζονται δομημένα και ημιδομημένα δεδομένα[11], [12]. Η χρήση εργαλείων όπως το **Apache JMeter** και το **Locust** καθίσταται κρίσιμη για την εκτέλεση δοκιμών φόρτου (load testing) και αντοχής (stress testing) σε κατανεμημένα συστήματα. Τα εργαλεία αυτά επιτρέπουν την προσομοίωση μεγάλου αριθμού ταυτόχρονων χρηστών ή αιτημάτων, προσδιορίζοντας τα σημεία συμφόρησης και τις αδυναμίες της υποδομής. Σε περιβάλλοντα federated queries, όπου τα ερωτήματα εκτελούνται σε πολλαπλές ετερογενείς πηγές, η ανάλυση των καθυστερήσεων και της κατανομής φόρτου αποκτά ιδιαίτερη σημασία για τη διασφάλιση αποδεκτών χρόνων απόκρισης[2], [3].

Παράλληλα, εργαλεία όπως το **sysbench** και το **YCSB (Yahoo! Cloud Serving Benchmark)** χρησιμοποιούνται για τη μέτρηση της απόδοσης βάσεων δεδομένων, εστιάζοντας σε λειτουργίες ανάγνωσης, εγγραφής και μικτών φορτίων εργασίας. Το YCSB, ειδικότερα, επιτρέπει τη δοκιμή NoSQL

και κατανεμημένων βάσεων, παρέχοντας ευελιξία στη διαμόρφωση σεναρίων που αντικατοπτρίζουν πραγματικές συνθήκες χρήσης. Η εφαρμογή τέτοιων εργαλείων σε συνδυασμό με σύγχρονες μορφές αποθήκευσης, όπως το Apache Iceberg και το Delta Lake, διευκολύνει την αξιολόγηση της απόδοσης σε περιβάλλοντα μεγάλων δεδομένων[13], [31]. Η εμφάνιση serverless υποδομών έχει οδηγήσει στην ανάπτυξη νέων προσεγγίσεων benchmarking, όπου η απόδοση συνδέεται όχι μόνο με τον χρόνο εκτέλεσης αλλά και με το κόστος χρήσης πόρων. Εργαλεία που ενσωματώνουν metrics όπως latency, throughput και οικονομική αποδοτικότητα καθίστανται απαραίτητα για την αξιολόγηση τέτοιων περιβαλλόντων. Η δυναμική κατανομή πόρων και οι SLA απαιτήσεις επιβάλλουν τη χρήση προηγμένων μεθόδων μέτρησης, οι οποίες λαμβάνουν υπόψη την ελαστικότητα και τη μεταβλητότητα του φορτίου[10], [28].

Επιπροσθέτως, η αξιοποίηση εργαλείων παρακολούθησης και profiling, όπως το **Prometheus** και το **Grafana**, επιτρέπει τη συνεχή συλλογή και απεικόνιση μετρικών απόδοσης, ενισχύοντας τη διαδικασία benchmarking με πραγματικά δεδομένα λειτουργίας. Η ενσωμάτωσή τους σε πλατφόρμες data federation διευκολύνει την ανίχνευση ανωμαλιών και την προσαρμογή στρατηγικών εκτέλεσης ερωτημάτων, ιδιαίτερα σε περιβάλλοντα με υψηλή ετερογένεια δεδομένων[27], [32].

3.6 Κριτήρια επιλογής τεχνολογιών

Στα περιβάλλοντα data federation, η αξιολόγηση των τεχνολογικών λύσεων βασίζεται στη συμβατότητά τους με κατανεμημένα μοντέλα πρόσβασης, όπου τα δεδομένα παραμένουν στις πηγές τους και ενοποιούνται λογικά μέσω μηχανισμών ερωτημάτων. Η επιλογή συστημάτων που υποστηρίζουν federated queries και virtualization κρίνεται κρίσιμη, καθώς επιτρέπει την αποφυγή πλεονασμού δεδομένων και ενισχύει την αποδοτικότητα πρόσβασης σε πολλαπλές πηγές[1], [2], [3]. Η αρχιτεκτονική ευελιξία αποτελεί θεμελιώδες κριτήριο, ιδιαίτερα σε περιβάλλοντα όπου συνυπάρχουν lakehouse υποδομές και υβριδικές βάσεις δεδομένων. Η δυνατότητα ενσωμάτωσης διαφορετικών μορφών αποθήκευσης, όπως columnar formats και transactional layers, επηρεάζει άμεσα την επιλογή τεχνολογιών που μπορούν να εξυπηρετήσουν ταυτόχρονα αναλυτικά και επιχειρησιακά φορτία. Σε αυτό το πλαίσιο, η υποστήριξη formats όπως Apache Iceberg και Delta Lake, σε συνδυασμό με βελτιστοποιημένες στρατηγικές indexing και partitioning, καθίσταται καθοριστική για την επίτευξη υψηλής απόδοσης και συνέπειας[13], [14], [31].

Η απόδοση και η αποδοτικότητα πόρων αποτελούν βασικούς άξονες αξιολόγησης, ιδίως σε περιβάλλοντα cloud και serverless υποδομών. Η δυνατότητα δυναμικής κατανομής πόρων, η ελαστικότητα και η συμμόρφωση με SLA απαιτήσεις επηρεάζουν την επιλογή τεχνολογιών που μπορούν να ανταποκριθούν σε μεταβαλλόμενα φορτία εργασίας. Οι σύγχρονες προσεγγίσεις query processing λαμβάνουν υπόψη όχι μόνο τον χρόνο εκτέλεσης αλλά και το κόστος χρήσης υπολογιστικών πόρων, ενσωματώνοντας μετρικές που σχετίζονται με την οικονομική βιωσιμότητα των λύσεων[10], [28]. Η διαλειτουργικότητα και η σημασιολογική ενοποίηση δεδομένων αποτελούν κρίσιμα κριτήρια, ιδιαίτερα σε περιβάλλοντα data spaces όπου η ανταλλαγή δεδομένων μεταξύ οργανισμών απαιτεί κοινά πρότυπα και μηχανισμούς κατανόησης. Η υιοθέτηση τεχνολογιών που υποστηρίζουν semantic integration, knowledge graphs και ontology-based access ενισχύει τη δυνατότητα ενοποίησης δεδομένων με διαφορετικές δομές και σημασίες, μειώνοντας την πολυπλοκότητα διαχείρισης και αυξάνοντας την αξιοπιστία των αποτελεσμάτων[9], [21].

Η ασφάλεια και η προστασία δεδομένων αποτελούν αναπόσπαστα στοιχεία της διαδικασίας επιλογής, ιδιαίτερα σε περιβάλλοντα όπου διακινούνται ευαίσθητες πληροφορίες. Η ενσωμάτωση μηχανισμών privacy-aware analytics και τεχνικών όπως federated learning συμβάλλει στη διατήρηση της

ιδιωτικότητας, επιτρέποντας την ανάλυση δεδομένων χωρίς την ανάγκη μεταφοράς τους σε κεντρικά αποθετήρια. Η αξιολόγηση τεχνολογιών με βάση την ικανότητά τους να υποστηρίζουν τέτοιες προσεγγίσεις είναι καθοριστική για τη συμμόρφωση με κανονιστικά πλαίσια και την ενίσχυση της εμπιστοσύνης[19], [29].

Πίνακας 3.8: Κριτήρια επιλογής τεχνολογιών σε περιβάλλοντα διαχείρισης δεδομένων

Κριτήριο	Περιγραφή	Επιπτώσεις στην επιλογή τεχνολογίας
Ενοποίηση δεδομένων	Δυνατότητα διασύνδεσης ετερογενών πηγών μέσω federated αρχιτεκτονικών	Επιλογή εργαλείων που υποστηρίζουν data federation και virtualization
Απόδοση	Χρόνος απόκρισης και αποδοτικότητα εκτέλεσης ερωτημάτων	Προτίμηση σε βελτιστοποιημένες μηχανές query processing
Κλιμάκωση	Ικανότητα διαχείρισης αυξανόμενου όγκου δεδομένων	Υιοθέτηση cloud-native και serverless λύσεων
Διαλειτουργικότητα	Συμβατότητα μεταξύ διαφορετικών συστημάτων και προτύπων	Χρήση semantic integration και κοινών data models
Ασφάλεια	Προστασία δεδομένων και συμμόρφωση με κανονισμούς	Ενσωμάτωση privacy-aware τεχνολογιών και federated learning
Ευελιξία αρχιτεκτονικής	Υποστήριξη διαφορετικών μορφών αποθήκευσης και επεξεργασίας	Επιλογή lakehouse και hybrid database συστημάτων
Κόστος λειτουργίας	Κατανάλωση υπολογιστικών πόρων και οικονομική βιωσιμότητα	Βελτιστοποίηση μέσω serverless και pay-as-you-go μοντέλων
Ωριμότητα οικοσυστήματος	Διαθεσιμότητα εργαλείων, τεκμηρίωσης και κοινότητας	Επιλογή τεχνολογιών με ενεργή υποστήριξη και εξέλιξη

Πηγές:[1], [2], [3], [10], [13], [14], [19], [21], [24], [28], [29], [31]

3.7 Συγκριτική αξιολόγηση και τελική επιλογή εργαλείων

Σε πλαίσια data federation, η σύγκριση επικεντρώνεται κυρίως στη δυνατότητα εκτέλεσης κατανεμημένων ερωτημάτων χωρίς φυσική μεταφορά δεδομένων, όπου διαφορετικά συστήματα παρουσιάζουν διαφοροποιήσεις ως προς τη στρατηγική decomposition και optimization των queries. Ορισμένες υλοποιήσεις δίνουν έμφαση στη σημασιολογική ενοποίηση μέσω ενδιάμεσων επιπέδων, ενώ άλλες βασίζονται σε άμεση πρόσβαση σε ετερογενείς πηγές, γεγονός που επηρεάζει την καθυστέρηση απόκρισης και την ακρίβεια των αποτελεσμάτων[2], [3], [21]. Η αντιπαραβολή λύσεων lakehouse έναντι παραδοσιακών data warehouse αποκαλύπτει κρίσιμες διαφορές στον τρόπο διαχείρισης αναλυτικών φορτίων. Οι lakehouse προσεγγίσεις ενσωματώνουν μηχανισμούς transactional consistency πάνω σε κατανεμημένα αποθετήρια, επιτρέποντας ταυτόχρονη επεξεργασία batch και streaming δεδομένων. Αντιθέτως, τα κλασικά συστήματα αποθήκευσης εμφανίζουν περιορισμούς ως προς την ευελιξία και την υποστήριξη μη δομημένων δεδομένων. Η τελική επιλογή εξαρτάται από το βαθμό που απαιτείται ενοποίηση πολλαπλών τύπων δεδομένων και από τη δυνατότητα διατήρησης υψηλής απόδοσης σε σύνθετα queries[11], [12], [13].

Παράλληλα, η αξιολόγηση τεχνολογιών query processing σε περιβάλλοντα cloud εισάγει πρόσθετες παραμέτρους, όπως η ελαστικότητα και το κόστος εκτέλεσης. Οι serverless μηχανές επεξεργασίας δεδομένων προσφέρουν δυναμική προσαρμογή πόρων, ωστόσο η απόδοσή τους εξαρτάται από τη βελτιστοποίηση του execution plan και τη σωστή κατανομή εργασιών. Αντίθετα, στατικά κατανεμημένα

συστήματα παρέχουν μεγαλύτερο έλεγχο αλλά απαιτούν αυξημένη διαχειριστική παρέμβαση. Η σύγκριση αυτών των προσεγγίσεων οδηγεί σε επιλογές που εξισορροπούν την απόδοση με το λειτουργικό κόστος και την πολυπλοκότητα διαχείρισης[10], [28]. Η σύγκριση υβριδικών βάσεων δεδομένων και cross-platform λύσεων αναδεικνύει τη σημασία της πολυμορφίας μοντέλων αποθήκευσης. Τα σύγχρονα συστήματα επιχειρούν να συνδυάσουν τη συνέπεια των σχεσιακών βάσεων με την επεκτασιμότητα των NoSQL προσεγγίσεων, ενσωματώνοντας παράλληλα μηχανισμούς βελτιστοποίησης μέσω τεχνητής νοημοσύνης. Η επιλογή τέτοιων τεχνολογιών βασίζεται στην ικανότητά τους να υποστηρίξουν διαφορετικά workloads χωρίς να θυσιάζεται η αποδοτικότητα ή η ακρίβεια των δεδομένων[5], [30].

Η τελική επιλογή εργαλείων επηρεάζεται από τη δυνατότητα ενσωμάτωσης σε data spaces και οικοσυστήματα συνεργατικής διαχείρισης δεδομένων. Η ύπαρξη κοινών προτύπων, μηχανισμών διακυβέρνησης και εργαλείων semantic interoperability καθορίζει τη βιωσιμότητα μιας λύσης σε περιβάλλοντα πολλαπλών οργανισμών. Οι πλατφόρμες που υποστηρίζουν τέτοιες δυνατότητες διευκολύνουν την ανταλλαγή δεδομένων και ενισχύουν την αξιοπιστία των αποτελεσμάτων μέσω ελεγχόμενων μηχανισμών πρόσβασης[4], [24], [25].

Πίνακας 3.9: Συγκριτική αξιολόγηση τεχνολογιών και εργαλείων διαχείρισης δεδομένων

Τεχνολογία / Εργαλείο	Αρχιτεκτονική προσέγγιση	Πλεονεκτήματα	Περιορισμοί	Καταλληλότητα χρήσης
Data Federation Systems	Κατανεμημένη, virtualized	Μη μεταφορά δεδομένων, ενοποιημένα queries	Αυξημένη πολυπλοκότητα βελτιστοποίησης	Ετερογενή περιβάλλοντα
Lakehouse Platforms	Υβριδική (data lake + warehouse)	Υποστήριξη batch & streaming, transactional consistency	Απαιτήσεις αποθήκευσης και tuning	Real-time analytics
Serverless Query Engines	Cloud-native, δυναμική κατανομή πόρων	Ελαστικότητα, χαμηλό λειτουργικό κόστος	Μεταβλητή απόδοση, latency	Μεταβαλλόμενα workloads
Hybrid Databases	Συνδυασμός SQL & NoSQL	Επεκτασιμότητα και συνέπεια	Πολυπλοκότητα υλοποίησης	Πολυμορφικά δεδομένα
Data Spaces Platforms	Ομοσπονδιακή, διαλειτουργική	Διακυβέρνηση δεδομένων, interoperability	Εξάρτηση από πρότυπα	Δια-οργανωτική συνεργασία

Πηγές:[2], [3], [4], [5], [10], [12], [13], [24], [25], [28], [30]

Κεφάλαιο 4: Σχεδίαση του προτεινόμενου συστήματος Data Space

4.1 Περιγραφή και στόχοι του προτεινόμενου συστήματος

Η προτεινόμενη λύση αφορά τον σχεδιασμό ενός πιλοτικού Data Space για την ενοποιημένη πρόσβαση σε ετερογενή δεδομένα πανεπιστημιακού περιβάλλοντος, με το Trino να λειτουργεί ως κεντρικό λογικό επίπεδο ομοσπονδιακής επεξεργασίας ερωτημάτων. Η βασική σχεδιαστική αρχή είναι ότι οι επιμέρους πηγές διατηρούν την αυτονομία τους και δεν απαιτείται φυσική συγκέντρωση του συνόλου των δεδομένων σε ένα κοινό αποθετήριο. Με τον τρόπο αυτό, το σύστημα επιχειρεί να αντιμετωπίσει τον πληροφοριακό κατακερματισμό που χαρακτηρίζει ένα σύγχρονο ακαδημαϊκό οικοσύστημα, όπου διαφορετικές υπηρεσίες χρησιμοποιούν ανεξάρτητα συστήματα, σχήματα και μηχανισμούς πρόσβασης.

Στην τελική μορφή του proof-of-concept, το Trino 480 παρέχει ενιαία SQL διεπαφή προς πολλαπλές πηγές. Η αρχιτεκτονική περιλαμβάνει ακαδημαϊκά δεδομένα από MySQL, στατιστικά αξιολογήσεων από PostgreSQL/MOΔΙΠ και εξαγωγές CSV του LDAP, τα οποία εκτίθενται σε αναγνώσιμη μορφή μέσω SSHFS και Hive. Παράλληλα, η πρόσβαση των χρηστών εντάσσεται στον συνολικό σχεδιασμό μέσω HTTPS, OpenLDAP authentication, αντιστοίχισης ομάδων και κανόνων εξουσιοδότησης, ώστε η ενοποίηση δεδομένων να συνδυάζεται με ελεγχόμενη πρόσβαση.

Οι στόχοι της πιλοτικής υλοποίησης είναι πρωτίστως λειτουργικοί και αποδεικτικοί. Επιδιώκεται να τεκμηριωθεί ότι διαφορετικές πηγές μπορούν να προσπελαστούν από κοινό query layer, ότι μπορούν να εκτελεστούν federated SQL ερωτήματα και cross-source joins χωρίς προηγούμενη αντιγραφή όλων των δεδομένων, και ότι η πρόσβαση μπορεί να διαφοροποιείται ανάλογα με τον χρήστη. Επιπλέον, εξετάζεται η δυνατότητα λειτουργία μόνο για ανάγνωσης, επανασύνδεσης των επιμέρους υπηρεσιών και αναπαραγωγής του πειραματικού περιβάλλοντος μέσω σαφώς καταγεγραμμένης παραμετροποίησης. Σε επίπεδο πανεπιστημιακού σεναρίου, ο σχεδιασμός υποστηρίζει τη λογική ενοποιημένης αξιοποίησης δεδομένων μαθημάτων, διδασκόντων, φοιτητών, εξεταστικής και αξιολογήσεων, χωρίς να προϋποθέτει ότι όλα τα στοιχεία προέρχονται από το ίδιο DBMS ή ακολουθούν κοινό schema. Η προσέγγιση αυτή συνδέεται άμεσα με τα ερευνητικά ερωτήματα της εργασίας σχετικά με την ενοποιημένη πρόσβαση, τη διαλειτουργικότητα, την ασφάλεια και τη λειτουργική βιωσιμότητα των ομοσπονδιακών αρχιτεκτονικών. Το προτεινόμενο σύστημα δεν αξιολογείται ως production υποδομή, αλλά ως τεχνική απόδειξη της λειτουργικής καταλληλότητας μιας ομοσπονδιακής αρχιτεκτονικής Data Space για ακαδημαϊκή χρήση.

4.2 Ανάλυση απαιτήσεων συστήματος

Η ανάλυση απαιτήσεων του προτεινόμενου Data Space αποσκοπεί στον προσδιορισμό των λειτουργικών και μη λειτουργικών χαρακτηριστικών που πρέπει να διαθέτει το σύστημα ώστε να υποστηρίζει αξιόπιστη ενοποιημένη πρόσβαση σε ετερογενείς πανεπιστημιακές πηγές. Σε αντίθεση με τις γενικές απαιτήσεις ενός Data Space που παρουσιάστηκαν στο προηγούμενο κεφάλαιο, η παρούσα ενότητα εξειδικεύει τις απαιτήσεις με βάση το συγκεκριμένο proof-of-concept και τις τεχνολογικές επιλογές της υλοποίησης. Σε λειτουργικό επίπεδο, βασική απαίτηση αποτελεί η παροχή κοινής SQL διεπαφής μέσω του Trino, από την οποία να είναι δυνατή η πρόσβαση στις διαφορετικές πηγές χωρίς συγκέντρωση των δεδομένων σε κοινή βάση. Το σύστημα πρέπει να προσπελαύνει PostgreSQL/MOΔΙΠ, MySQL και εξαγωγές CSV του LDAP, να υποστηρίζει ερωτήματα ανά πηγή αλλά

και cross-source federated queries και να εφαρμόζει λογική μόνο για ανάγνωση όπου απαιτείται. Επιπλέον, απαιτούνται μηχανισμοί ταυτοποίησης χρηστών και διαφοροποιημένης εξουσιοδότησης, ώστε η πρόσβαση σε προστατευμένα δεδομένα να εξαρτάται από τα δικαιώματα του χρήστη. Οι απαιτήσεις αυτές αποτυπώνονται στο πρακτικό μέρος ως FR-01 έως FR-10.

Οι μη λειτουργικές απαιτήσεις αφορούν κυρίως την ασφάλεια, τη διαθεσιμότητα, τη φρεσκάδα των δεδομένων, τη συντηρησιμότητα, την ιδιωτικότητα και την ερμηνευσιμότητα των αποτελεσμάτων. Στον σχεδιασμό προβλέπονται HTTPS επικοινωνία, loopback-only forwarded ports, λογαριασμοί περιορισμένων δικαιωμάτων, αυτόματη επανεκκίνηση κρίσιμων υπηρεσιών και σταθερή ονοματολογία catalogs, schemas και views. Παράλληλα, η δυνατότητα απόκρισης και η επεκτασιμότητα αποτελούν σημαντικά ποιοτικά χαρακτηριστικά, χωρίς όμως το single-node περιβάλλον να αντιμετωπίζεται ως production benchmark. Η ανάλυση αυτή λειτουργεί ως βάση για τη σχεδίαση της αρχιτεκτονικής και για τα κριτήρια με τα οποία θα αξιολογηθεί η υλοποίηση στο επόμενο κεφάλαιο.

4.2.1 Απαιτήσεις χρηστών και διαχειριστή

Οι απαιτήσεις των τελικών χρηστών επικεντρώνονται κυρίως στην απλή και ελεγχόμενη πρόσβαση στα ετερογενή δεδομένα μέσω μιας ενιαίας SQL διεπαφής. Ο χρήστης δεν πρέπει να χρειάζεται άμεση γνώση των μηχανισμών σύνδεσης κάθε επιμέρους DBMS ούτε να διαχειρίζεται ξεχωριστά διαπιστευτήρια για τις πηγές. Μέσω του Trino πρέπει να μπορεί να προσπελαύνει τα διαθέσιμα catalogs και να υποβάλλει ερωτήματα χρησιμοποιώντας πλήρως προσδιορισμένα αντικείμενα, ενώ η πολυπλοκότητα της πρόσβασης στις υποκείμενες πηγές παραμένει στο επίπεδο της υποδομής. Η απαίτηση αυτή συνδέεται άμεσα με την FR-01 για κοινή SQL διεπαφή και την FR-07 για εκτέλεση cross-source queries.

Η πρόσβαση δεν πρέπει να είναι ομοιόμορφη για όλους τους χρήστες. Η τεκμηριωμένη σχεδίαση προβλέπει αυθεντικοποίηση μέσω HTTPS και OpenLDAP και διαφοροποίηση δικαιωμάτων με βάση την ιδιότητα του λογαριασμού. Ειδικότερα, οι χρήστες που αντιστοιχίζονται στην ομάδα staff μπορούν να αποκτούν πρόσβαση μόνο για ανάγνωση σε προστατευμένα δεδομένα ΜΟΔΠ, ενώ ένας μη εξουσιοδοτημένος φοιτητικός λογαριασμός μπορεί να συνδεθεί επιτυχώς αλλά να απορρίπτεται κατά την προσπάθεια πρόσβασης στο προστατευμένο αντικείμενο. Η απαίτηση αυτή αντιστοιχεί στις FR-05, FR-06 και FR-08.

Από την πλευρά του διαχειριστή, απαιτείται δυνατότητα παραμετροποίησης και συντήρησης των catalogs, των connectors, των SSH tunnels, του SSHFS, των μηχανισμών authentication και access control, καθώς και των σχετικών systemd services. Ο διαχειριστής πρέπει επίσης να διαθέτει σαφείς μηχανισμούς διάγνωσης μέσω logs και ελέγχων ανά επίπεδο, ώστε να εντοπίζει προβλήματα σύνδεσης, εξουσιοδότησης ή διαθεσιμότητας. Η απαίτηση αυτόματης επανασύνδεσης και η επαναληψιμότητα της εγκατάστασης αποτυπώνονται στις FR-09 και FR-10.

4.2.2 Απαιτήσεις δεδομένων και απόδοσης

Οι απαιτήσεις δεδομένων του προτεινόμενου Data Space προκύπτουν κυρίως από την ετερογένεια των πηγών και από την ανάγκη διατήρησης της αυτονομίας τους. Η τελική αρχιτεκτονική καλείται να προσπελάζει δεδομένα από MySQL, PostgreSQL/ΜΟΔΠ και εξαγωγές CSV του LDAP, τα οποία παρουσιάζουν διαφορετικά μοντέλα αποθήκευσης, schemas, τύπους δεδομένων και μηχανισμούς

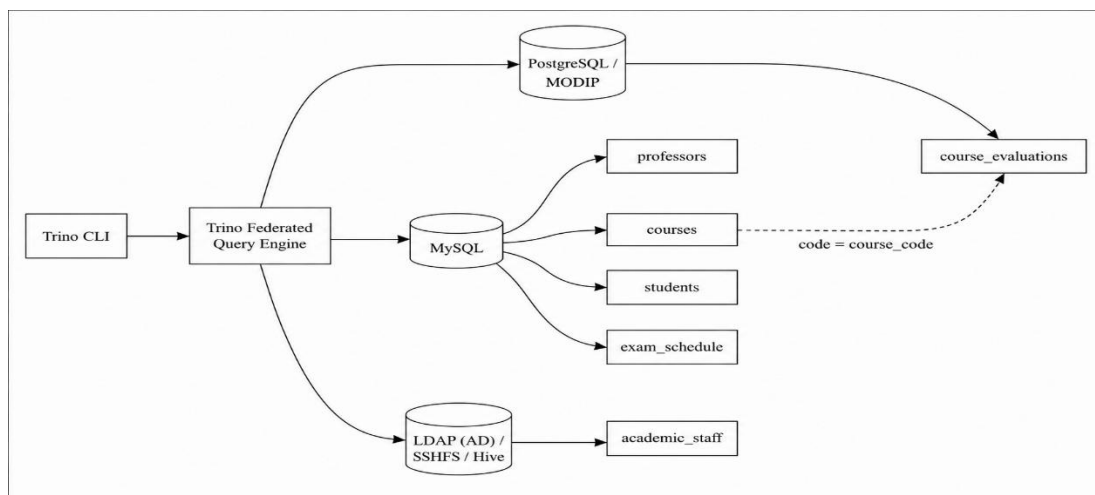
ενημέρωσης. Για τον λόγο αυτό, βασική απαίτηση αποτελεί η δυνατότητα του Trino να εκθέτει τις πηγές μέσω σταθερών catalog, schema, table και view identifiers, επιτρέποντας την εκτέλεση ενοποιημένων SQL ερωτημάτων χωρίς προηγούμενη μεταφορά των δεδομένων σε κοινό αποθετήριο. Ιδιαίτερη σημασία έχει η συμβατότητα των δεδομένων που συμμετέχουν σε συσχετίσεις μεταξύ διαφορετικών συστημάτων. Στις σχεσιακές βάσεις αξιοποιούνται οι εγγενείς τύποι και τα υφιστάμενα σχήματα, ενώ στις εξαγωγές LDAP οι στήλες της external table δηλώθηκαν ως varchar, ώστε να αντιμετωπίζονται με ανθεκτικό τρόπο κενές ή ετερογενείς τιμές. Όπου απαιτείται αριθμητική ή άλλη εξειδικευμένη επεξεργασία, οι μετατροπές πραγματοποιούνται ρητά στα ερωτήματα. Παράλληλα, η χρήση stable views για τα staff και students διατηρεί σταθερό λογικό επίπεδο πρόσβασης ακόμη και όταν ανανεώνονται τα υποκείμενα CSV αρχεία.

Ως προς τη διαθεσιμότητα και τη φρεσκάδα, οι βάσεις MySQL και PostgreSQL διαβάζονται κατά την εκτέλεση του query, ενώ οι εξαγωγές LDAP ακολουθούν διαφορετικό κύκλο ενημέρωσης και εκτίθενται μέσω read-only SSHFS. Η αρχιτεκτονική συνεπώς πρέπει να λαμβάνει υπόψη ότι η χρονική επικαιρότητα δεν είναι ομοιόμορφη για όλες τις πηγές. Οι απαιτήσεις απόδοσης εστιάζουν σε αποδεκτό χρόνο απόκρισης για την πιλοτική επίδειξη, ορθή εκτέλεση cross-source joins και δυνατότητα επεξεργασίας διαφορετικών όγκων δεδομένων χωρίς λειτουργική αστοχία. Ωστόσο, δεν τίθεται production SLA, καθώς το περιβάλλον είναι single-node proof-of-concept. Η αξιολόγηση απόδοσης πρέπει επομένως να βασίζεται σε πραγματικούς χρόνους εκτέλεσης, όγκο εισόδου και splits, με σαφή διάκριση μεταξύ λειτουργικής επιτυχίας, απόδοσης και επεκτασιμότητας.

4.3 Σχεδιασμός αρχιτεκτονικής

Η αρχιτεκτονική του προτεινόμενου Data Space σχεδιάστηκε με στόχο την ενοποιημένη πρόσβαση σε ετερογενείς ακαδημαϊκές πηγές, χωρίς να απαιτείται φυσική συγκέντρωση των δεδομένων σε κοινό αποθετήριο. Κεντρικό στοιχείο της λύσης αποτελεί ο Trino 480, ο οποίος λειτουργεί ως federated query engine και παρέχει ενιαία SQL διεπαφή προς τις διαφορετικές πηγές. Οι πηγές διατηρούν την αυτονομία τους, ενώ η ενοποίηση πραγματοποιείται λογικά κατά την εκτέλεση των ερωτημάτων. Η αρχιτεκτονική περιλαμβάνει τρεις βασικές κατηγορίες δεδομένων. Η MySQL παρέχει ακαδημαϊκά δεδομένα, όπως μαθήματα και διδάσκοντες, ενώ η PostgreSQL/MODPI παρέχει δεδομένα αξιολογήσεων. Οι δύο σχεσιακές πηγές προσπελούνται μέσω ανεξάρτητων SSH tunnels, ώστε οι database ports να μην εκτίθενται δημόσια. Παράλληλα, οι εξαγωγές CSV του LDAP προσαρτώνται read-only μέσω SSHFS και χαρτογραφούνται σε SQL αντικείμενα με τον Hive connector. Έτσι, ακόμη και αρχεία μη σχεσιακής προέλευσης εντάσσονται στο κοινό μοντέλο πρόσβασης του Trino.

Οι connectors και τα catalogs λειτουργούν ως το ενδιάμεσο επίπεδο μεταξύ του Trino και των πηγών. Τα catalogs mysql, modip_stats και ldap_exports επιτρέπουν τη χρήση πλήρως προσδιορισμένων ονομάτων και διατηρούν σαφή διαχωρισμό μεταξύ των επιμέρους συστημάτων. Ο Trino αναλαμβάνει την επιλογή των κατάλληλων connectors, τη δημιουργία των επιμέρους προσπελάσεων και τον συνδυασμό των αποτελεσμάτων σε ενιαίο federated query. Η αρχιτεκτονική περιλαμβάνει επίσης ανεξάρτητη ροή ταυτοποίησης και εξουσιοδότησης μέσω HTTPS, OpenLDAP, group mapping και file-based access control. Η διάκριση αυτή είναι σημαντική, καθώς η ροή δεδομένων και η ροή ταυτότητας δεν ταυτίζονται: ο χρήστης πιστοποιείται αρχικά και στη συνέχεια εφαρμόζονται οι κανόνες πρόσβασης πριν εκτελεστεί το query. Με αυτή τη σχεδίαση επιτυγχάνεται ενοποιημένη, ελεγχόμενη και επεκτάσιμη πρόσβαση σε πολλαπλές αυτόνομες πηγές.



Σχήμα 4.1: Αρχιτεκτονική της πιλοτικής υλοποίησης του ακαδημαϊκού Data Space και ενοποιημένη πρόσβαση στις ετερογενείς πηγές μέσω Trino.

Η συνολική λογική της προτεινόμενης αρχιτεκτονικής παρουσιάζεται στο Σχήμα 4.1. Ο Trino καταλαμβάνει κεντρική θέση ως επίπεδο ομοσπονδιακής πρόσβασης, επιτρέποντας την εκτέλεση ερωτημάτων πάνω στις αυτόνομες πηγές MySQL, PostgreSQL/MODIP και LDAP και τον συνδυασμό δεδομένων μεταξύ τους χωρίς προηγούμενη φυσική ενοποίησή τους.

4.4 Μοντελοποίηση και ενοποίηση των πηγών δεδομένων

Η μοντελοποίηση του Data Space βασίζεται στη διατήρηση των επιμέρους πηγών και των εγγενών σχημάτων τους, ενώ η ενοποίηση πραγματοποιείται λογικά μέσω του Trino. Η προσέγγιση επιτρέπει τη συσχέτιση σχεσιακών δεδομένων και εξαγωγών LDAP χωρίς δημιουργία ενιαίας φυσικής βάσης, διατηρώντας παράλληλα την αυτονομία των συστημάτων προέλευσης.

4.4.1 Σχήματα, οντότητες και σχέσεις δεδομένων

Η μοντελοποίηση των πηγών αναπτύχθηκε σε διαδοχικά επίπεδα, από το αρχικό συνθετικό ακαδημαϊκό σχήμα έως την τελική ομοσπονδιακή διασύνδεση πραγματικών ετερογενών πηγών. Στην αρχική βάση mysql ορίστηκαν οι πίνακες professors, courses, students και exam_schedule. Κάθε πίνακας διαθέτει διακριτό primary key (professor_id, course_id, student_id, exam_id), ενώ τα courses.code και students.registration_number χαρακτηρίζονται ως μοναδικά αναγνωριστικά. Η εσωτερική ακεραιότητα εξασφαλίζεται μέσω των foreign keys courses.professor_id → professors.professor_id και exam_schedule.course_id → courses.course_id, τα οποία μοντελοποιούν αντίστοιχα την ανάθεση διδάσκοντα και τη σύνδεση μιας εξέτασης με συγκεκριμένο μάθημα.

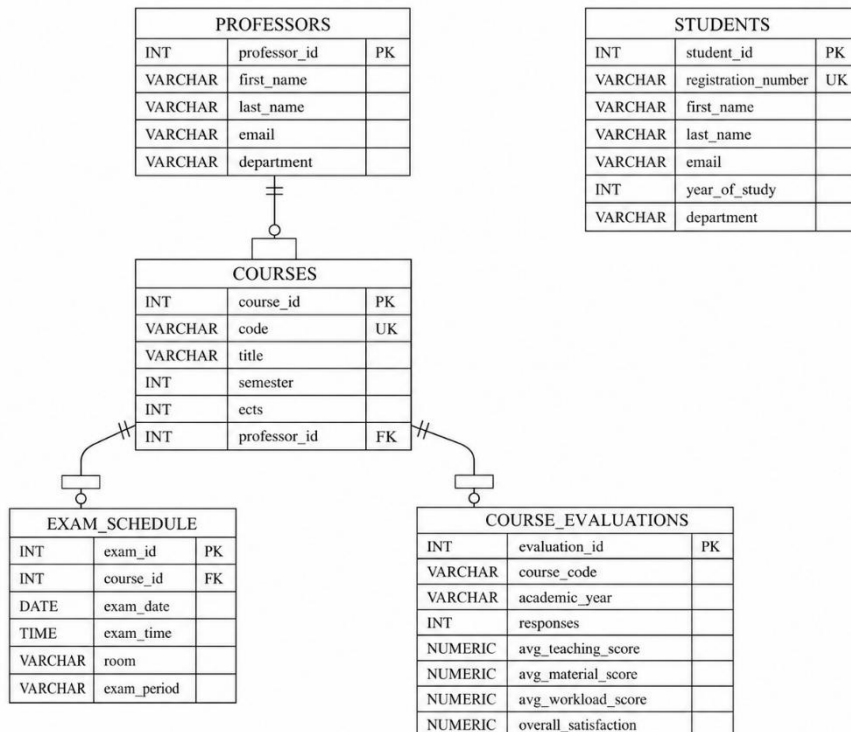
Το αρχικό μοντέλο περιλάμβανε επίσης ανεξάρτητη PostgreSQL πηγή για δεδομένα αξιολόγησης μαθημάτων, ώστε να δοκιμαστεί η συσχέτιση ακαδημαϊκών και αξιολογικών δεδομένων. Στην πληρέστερη υλοποίηση, η PostgreSQL/MODIP εκτίθεται μέσω του catalog modip_stats και χρησιμοποιεί πραγματικά αντικείμενα όπως course, questioner και vote, ενώ η MySQL εκτίθεται μέσω του catalog mysql. Παράλληλα, τα στοιχεία φοιτητών και προσωπικού εντάσσονται στο τελικό ομοσπονδιακό περιβάλλον και μέσω του catalog ldap_exports. Τα αρχεία staff.csv και students.csv χαρτογραφούνται από τον Hive connector σε external table και σταθερά views, με τα LDAP attributes να δηλώνονται ως varchar, ώστε να αντιμετωπίζονται με συνέπεια κενές ή μικτές τιμές.

Σχεδίαση του προτεινόμενου συστήματος Data Space

Οι σχέσεις μεταξύ διαφορετικών συστημάτων δεν επιβάλλονται με φυσικά foreign keys. Δημιουργούνται κατά την εκτέλεση των federated queries μέσω λογικών identifiers. Στην τελική υλοποίηση χρησιμοποιούνται, μεταξύ άλλων, κανονικοποιημένο email για αντιστοίχιση διδασκόντων με LDAP και συνδυασμός τίτλου και εξαμήνου για συσχέτιση μαθημάτων MySQL με ΜΟΔΙΠ. Το τελευταίο αποτελεί λειτουργική επιλογή του proof-of-concept, ενώ ένα μόνιμο κοινό course identifier θα παρείχε ισχυρότερη σημασιολογική αντιστοίχιση σε παραγωγικό περιβάλλον.

Πίνακας 4.1: Κύριες οντότητες και σχέσεις του μοντέλου δεδομένων

Πηγή	Αντικείμενο	Κύριο αναγνωριστικό / σχέση	Ρόλος
MySQL	professors	professor_id PK	Διδάσκοντες
MySQL	courses	course_id PK, professor_id FK, code UNIQUE	Μαθήματα
MySQL	students	student_id PK, registration_number UNIQUE	Φοιτητές
MySQL	exam_schedule	exam_id PK, course_id FK	Πρόγραμμα εξεταστικής
PostgreSQL	course, questioner, vote	Εσωτερικά IDs και συσχετίσεις	Αξιολογήσεις
LDAP/Hive	staff, students	uid, mail, LDAP attributes	Ταυτότητα/προσωπικό/φοιτητές



Σχήμα 4.2: Σχήματα και βασικές σχέσεις μεταξύ των οντοτήτων των πηγών δεδομένων της πιλοτικής υλοποίησης.

4.4.2 Αντιστοίχιση δεδομένων και σημασιολογική ετερογένεια

Η ενοποίηση των πηγών δεν περιορίζεται στην τεχνική δυνατότητα σύνδεσής τους μέσω του Trino, αλλά απαιτεί και αντιστοίχιση δεδομένων που περιγράφουν τις ίδιες ή συναφείς οντότητες με διαφορετικό τρόπο. Στην υλοποίηση παρατηρούνται διαφορές τόσο στη δομή των schemas όσο και στην ονοματολογία, στους identifiers και στους τύπους δεδομένων. Η MySQL χρησιμοποιεί σχεσιακές οντότητες όπως *courses* και *professors*, η PostgreSQL/MODIP διαφορετικό εσωτερικό μοντέλο με *course*, *questioner* και *vote*, ενώ οι εξαγωγές LDAP αποτελούνται από *directory attributes* που χαρτογραφούνται σε Hive external table και views.

Η σημασιολογική αντιστοίχιση αντιμετωπίζεται στο επίπεδο των federated queries. Για τη συσχέτιση διδασκόντων μεταξύ MySQL και LDAP χρησιμοποιείται το πεδίο email, αφού προηγηθεί κανονικοποίηση μέσω `lower(trim(...))`. Η επιλογή αυτή επιτρέπει ακριβή αντιστοίχιση διαφορετικών εγγράφων που αναφέρονται στο ίδιο πρόσωπο, χωρίς όμως να εξαλείφει περιπτώσεις διαφορετικών email ή πολλαπλών LDAP λογαριασμών. Στα καταγεγραμμένα αποτελέσματα εμφανίζονται τόσο μοναδικές αντιστοιχίσεις όσο και πολλαπλοί λογαριασμοί για το ίδιο email, γεγονός που αναδεικνύει πρακτικά το ζήτημα της ταυτοποίησης οντοτήτων. Αντίστοιχα, η τελική συσχέτιση μαθημάτων MySQL–MODIP πραγματοποιείται με συνδυασμό κανονικοποιημένου τίτλου και εξαμήνου, επειδή δεν τεκμηριώνεται κοινό μόνιμο identifier μεταξύ των δύο πραγματικών συστημάτων. Στην αρχική συνθετική υλοποίηση είχε χρησιμοποιηθεί η αντιστοίχιση `courses.code = course_evaluations.course_code`, η οποία όμως εμφάνισε ασυνέπεια στην τιμή 420 έναντι 1801 και επομένως δεν αποτελεί ασφαλές γενικευμένο κλειδί.

Τέλος, τα LDAP attributes δηλώθηκαν ως varchar, καθώς τα CSV δεν διαθέτουν αυστηρό σχεσιακό schema. Οι απαιτούμενες αριθμητικές ή ημερολογιακές μετατροπές εκτελούνται ρητά στα queries. Συνεπώς, η σημασιολογική αντιστοίχιση στο proof-of-concept βασίζεται σε ελεγχόμενη κανονικοποίηση και λογικούς κανόνες αντιστοίχισης, ενώ σε παραγωγικό περιβάλλον θα ήταν προτιμότερη η χρήση σταθερών κοινών αναγνωριστικών.

Πίνακας 4.2: Κύριες σημασιολογικές αντιστοιχίσεις μεταξύ πηγών

Οντότητα	Πηγή Α	Πηγή Β	Κανόνας αντιστοίχισης
Διδάσκων	mysql.unitron.professors.email	ldap_exports.directory.staff.mail	<code>lower(trim(email))</code>
Μάθημα	MySQL title, semester	MODIP title, semester	κανονικοποιημένος τίτλος + εξάμηνο
Μάθημα, αρχικό demo	courses.code	course_evaluations.course_code	λογικό code mapping, με γνωστή ασυνέπεια 420/1801
LDAP attributes	CSV fields	Hive columns	κοινή αναπαράσταση ως varchar

4.5 Σχεδιασμός ερωτημάτων και σεναρίων χρήσης

Ο σχεδιασμός των ερωτημάτων οργανώθηκε ώστε να εξετάζει σταδιακά τόσο την ανεξάρτητη προσπέλαση κάθε πηγής όσο και τη δυνατότητα σύνθεσης πληροφοριών μέσω federated queries. Αρχικά προβλέπονται απλά ερωτήματα ανά πηγή, με σκοπό την επαλήθευση ότι τα διαθέσιμα catalogs, schemas και αντικείμενα είναι ορατά από το Trino και ότι η ανάκτηση δεδομένων πραγματοποιείται χωρίς μεταβολή των συστημάτων προέλευσης. Η λογική αυτή συνδέεται άμεσα με τις FR-01 έως FR-04 και αποτελεί προϋπόθεση για την εφαρμογή της FR-07 σχετικά με τα cross-source ερωτήματα.

Για τα ακαδημαϊκά δεδομένα προβλέπονται ερωτήματα μαθημάτων και καθηγητών από το MySQL, με αξιοποίηση της σχέσης `courses.professor_id → professors.professor_id`. Στο αρχικό συνθετικό schema είχαν επίσης σχεδιαστεί ερωτήματα για φοιτητές και πρόγραμμα εξεταστικής μέσω των `students` και `exam_schedule`, με το δεύτερο να συνδέεται με τα μαθήματα μέσω `course_id`. Στην τελική υλοποίηση, τα στοιχεία φοιτητών και προσωπικού εντάσσονται κυρίως μέσω των LDAP CSV views, ενώ τα δεδομένα αξιολογήσεων προέρχονται από την PostgreSQL/MOΔΙΠ.

Τα συνδυαστικά σενάρια σχεδιάστηκαν ώστε να υπερβαίνουν τα ανεξάρτητα SELECT και να αξιοποιούν joins, aggregations και αναλυτικές συναρτήσεις. Η τεκμηριωμένη υλοποίηση περιλαμβάνει smoke test όλων των πηγών, ομοσπονδιακή συσχέτιση μαθημάτων MySQL με καθηγητές, LDAP στοιχεία προσωπικού και δεδομένα MOΔΙΠ, καθώς και κατάταξη μαθημάτων βάσει συγκεντρωτικών αξιολογήσεων. Με τον τρόπο αυτό, κάθε κατηγορία ερωτήματος αντιστοιχεί σε σαφές πληροφοριακό ζητούμενο: ανάκτηση βασικών ακαδημαϊκών στοιχείων, διερεύνηση σχέσεων μεταξύ οντοτήτων και παραγωγή συγκεντρωτικών αποτελεσμάτων από πολλαπλές πηγές. Η τελική επαλήθευση των σεναρίων παρουσιάζεται στο Κεφάλαιο 5, όπου τα σχεδιασμένα ερωτήματα μετατρέπονται σε πραγματικά ερωτήματα SQL και αξιολογούνται με βάση τα καταγεγραμμένα αποτελέσματα.

Πίνακας 4.3: Σχεδιαζόμενες κατηγορίες ερωτημάτων και πηγές δεδομένων

Κατηγορία	Κύρια πηγή/πηγές	Αναμενόμενο αποτέλεσμα
Μαθήματα	MySQL	Κωδικός, τίτλος, εξάμηνο, ECTS
Καθηγητές	MySQL	Στοιχεία διδάσκοντα και σύνδεση με μάθημα
Φοιτητές	MySQL αρχικού demo / εξαγωγές LDAP τελικής υλοποίησης	Βασικά ακαδημαϊκά ή directory στοιχεία
Εξεταστική	<code>exam_schedule + courses</code>	Ημερομηνία, ώρα, αίθουσα και μάθημα
Αξιολογήσεις	PostgreSQL/MOΔΙΠ	Συμμετέχοντες, απαντήσεις, μέσος βαθμός
Federated ανάλυση	MySQL + MODIP + LDAP	Ενοποιημένη εικόνα μαθημάτων, διδασκόντων και αξιολογήσεων
Αναλυτικά ερωτήματα	Πολλαπλές πηγές	Aggregations, κατάταξη και συγκριτική ανάλυση

4.6 Ασφάλεια, δικαιώματα πρόσβασης και προστασία δεδομένων

Η ασφάλεια του προτεινόμενου Data Space σχεδιάστηκε σε διακριτά επίπεδα, ώστε η ταυτοποίηση των χρηστών, η εξουσιοδότηση και η πρόσβαση στις υποκείμενες πηγές να αντιμετωπίζονται ανεξάρτητα. Η σύνδεση των τελικών χρηστών με τον Trino πραγματοποιείται μέσω HTTPS, ενώ η επαλήθευση των ιδρυματικών διαπιστευτηρίων ανατίθεται στο OpenLDAP. Με τον τρόπο αυτό, το password δεν

αποθηκεύεται στον Trino και ο μηχανισμός authentication παραμένει διαχωρισμένος από την εφαρμογή των δικαιωμάτων πρόσβασης. Η εξουσιοδότηση βασίζεται σε group mapping και file-based access control. Η ιδιότητα eduPersonAffiliation=staff χρησιμοποιείται για την παραγωγή της ομάδας staff, η οποία αποκτά πρόσβαση μόνο για ανάγνωση στο προστατευμένο αντικείμενο στατιστικών ΜΟΔΠΙ. Αντίθετα, ένας αυθεντικοποιημένος μη-staff χρήστης μπορεί να συνδεθεί στο Trino, αλλά απορρίπτεται κατά την εκτέλεση `SELECT` στο συγκεκριμένο table/view. Η διάκριση αυτή υλοποιεί σαφώς την αρχή authentication $\neq$ authorization και συνδέεται με τις απαιτήσεις FR-05, FR-06 και FR-08.

Σε επίπεδο πηγών εφαρμόζεται η αρχή του least privilege. Οι connector λογαριασμοί προορίζονται για πρόσβαση μόνο για ανάγνωση, το Hive catalog λειτουργεί με `hive.security=read-only`, ενώ οι εξαγωγές LDAP προσαρτώνται μέσω SSHFS επίσης σε λειτουργία μόνο για ανάγνωση. Οι PostgreSQL και MySQL πηγές δεν εκτίθενται δημόσια· η πρόσβαση πραγματοποιείται μέσω SSH tunnels δεσμευμένων σε loopback διευθύνσεις, περιορίζοντας την άμεση δικτυακή έκθεση των database ports. Η προστασία προσωπικών δεδομένων ενσωματώνεται στη σχεδίαση μέσω ελαχιστοποίησης αποκάλυψης, αποφυγής δημοσίευσης διαπιστευτηρίων, private keys και μη απαραίτητων identifiers, καθώς και χρήσης μόνο των attributes που απαιτούνται για το proof-of-concept. Παράλληλα, η τεκμηρίωση αναγνωρίζει ως περιορισμό τη χρήση προσωρινής insecure LDAP διαδρομής στο demo, η οποία δεν θεωρείται κατάλληλη για παραγωγικό περιβάλλον.

4.7 Περιορισμοί και πιθανές δυσκολίες σχεδίασης

Ο σχεδιασμός του προτεινόμενου Data Space παρουσιάζει ορισμένους περιορισμούς που απορρέουν από τη ομοσπονδιακή φύση της αρχιτεκτονικής και την ετερογένεια των πηγών. Πρώτος σημαντικός παράγοντας είναι η σημασιολογική ετερογένεια, καθώς διαφορετικά συστήματα χρησιμοποιούν διαφορετικά schemas, identifiers, τύπους δεδομένων και συμβάσεις ονοματολογίας. Η ανάγκη κανονικοποίησης πεδίων, όπως email ή τίτλοι μαθημάτων, αυξάνει την πολυπλοκότητα των cross-source joins και ενδέχεται να απαιτεί ειδικούς κανόνες αντιστοίχισης.

Επιπλέον, η ομοσπονδιακή επεξεργασία εξαρτάται άμεσα από τη διαθεσιμότητα και την απόκριση των εξωτερικών πηγών. Η PostgreSQL/ΜΟΔΠΙ και η MySQL προσπελούνται μέσω ανεξάρτητων SSH tunnels, ενώ οι εξαγωγές LDAP εξαρτώνται από SSHFS και απομακρυσμένα αρχεία. Επομένως, προσωρινή αστοχία δικτύου, tunnel ή remote filesystem μπορεί να επηρεάσει την εκτέλεση ενός ερωτήματος. Η επιλογή αυτή αποτελεί συνειδητό συμβιβασμό έναντι της κεντρικής αντιγραφής δεδομένων.

Πρόσθετη δυσκολία αποτελεί το γεγονός ότι οι connectors δεν παρέχουν απαραίτητα ίδιες δυνατότητες pushdown ή βελτιστοποίησης για κάθε πηγή. Σε distributed joins, μέρος της επεξεργασίας μπορεί να εκτελείται στον Trino, αυξάνοντας τη μεταφορά δεδομένων και το κόστος εκτέλεσης. Παράλληλα, η χρήση Hive external tables πάνω από CSV απαιτεί χειρισμό metadata και ρητές μετατροπές τύπων, καθώς τα LDAP attributes έχουν δηλωθεί ως varchar. Σε επίπεδο επεκτασιμότητας, η υλοποίηση είναι single-node και επομένως δεν αποτελεί πρότυπο παραγωγικής κλιμάκωσης. Η προσθήκη νέων πηγών, μεγαλύτερων συνόλων δεδομένων ή περισσότερων ταυτόχρονων χρηστών θα απαιτούσε multi-node σχεδιασμό, ισχυρότερο metastore, monitoring και ενδεχομένως διαφορετικές στρατηγικές caching ή optimization. Τέλος, αλλαγές στα schemas, στα metadata ή στις πολιτικές πρόσβασης μπορούν να

Σχεδίαση του προτεινόμενου συστήματος Data Space

επιρεάσουν queries, views και access-control rules, συνεπώς απαιτούν ελεγχόμενη συντήρηση και συνεχή επαλήθευση της συνοχής του συστήματος.

Κεφάλαιο 5: Υλοποίηση και πειραματική εφαρμογή

5.1 Περιβάλλον ανάπτυξης και εγκατάστασης

Η πιλοτική υλοποίηση αναπτύχθηκε σε περιβάλλον Linux server, με λογική standalone εγκατάστασης και σαφή διαχωρισμό μεταξύ runtime, δεδομένων και αρχείων παραμετροποίησης. Κεντρικό λογισμικό αποτελεί ο **Trino 480**, εγκατεστημένος στον κατάλογο /opt/trino, ενώ τα runtime δεδομένα αποθηκεύονται ανεξάρτητα στο /var/trino/data. Η επιλογή αυτή διευκολύνει τη συντήρηση και επιτρέπει τη διατήρηση του runtime state σε πιθανή αναβάθμιση της εγκατάστασης. Ο Trino εκτελείται από μη προνομιούχο service user, περιορίζοντας τα δικαιώματα της διεργασίας στο λειτουργικό σύστημα.

Απαραίτητη προϋπόθεση αποτελεί συμβατή εγκατάσταση **Java**, η οποία ελέγχεται με `java -version`, καθώς οι απαιτήσεις εξαρτώνται από την εκάστοτε έκδοση του Trino. Για τη διαχείριση και επαλήθευση των SQL ερωτημάτων εγκαταστάθηκε επίσης το **Trino CLI**, κατά προτίμηση στην ίδια έκδοση με τον server. Το CLI επικοινωνεί αποκλειστικά με τον coordinator και δεν απαιτεί άμεση πρόσβαση στις υποκείμενες βάσεις ή στις εξαγωγές LDAP.

Πίνακας 5.1: Βασικά στοιχεία του πειραματικού περιβάλλοντος

Στοιχείο	Τεκμηριωμένη ρύθμιση
Πλατφόρμα	Linux server / virtual machine
Trino	Version 480
Runtime path	/opt/trino
Data directory	/var/trino/data
LDAP CSV mount	/mnt/ldap-exports
Hive metastore	/mnt/trino-ldap-metastore
MODIP tunnel	127.0.0.1:2000
MySQL tunnel	127.0.0.1:3307
Trino HTTPS	8443
Βασικά εργαλεία	Java, Trino CLI, SSH, SSHFS, ldap-utils, netcat

Οι βασικές εξαρτήσεις του περιβάλλοντος περιλαμβάνουν curl, tar, openssh-client, netcat-openbsd, ldap-utils και sshfs. Τα εργαλεία αυτά υποστηρίζουν αντίστοιχα την εγκατάσταση του Trino, τη δημιουργία SSH tunnels, τον δικτυακό έλεγχο, την επικοινωνία με OpenLDAP και την προσάρτηση μόνο για ανάγνωση των εξαγωγών CSV του LDAP. Για τα τελευταία χρησιμοποιείται το /mnt/ldap-exports, ενώ το Hive file metastore αποθηκεύεται στο /mnt/trino-ldap-metastore. Σε δικτυακό επίπεδο, η PostgreSQL/MODIP και η MySQL προσπελούνται μέσω loopback SSH forwarding στις τοπικές θύρες 2000 και 3307 αντίστοιχα. Οι τελικοί χρήστες συνδέονται στον Trino μέσω HTTPS στη θύρα 8443, ενώ η HTTP θύρα 8080 περιορίζεται σε τοπική διοικητική χρήση. Η συνολική προετοιμασία ολοκληρώνεται με έλεγχο Java, εγκατάσταση Trino/CLI, δημιουργία καταλόγων και service user, εγκατάσταση dependencies και επαλήθευση της βασικής δικτυακής συνδεσιμότητας.

5.2 Εγκατάσταση και παραμετροποίηση του Trino

5.2.1 Διαμόρφωση coordinator, worker και βασικών παραμέτρων

Η παραμετροποίηση του Trino πραγματοποιήθηκε με λογική **single-node deployment**, όπου ο ίδιος κόμβος λειτουργεί ταυτόχρονα ως coordinator και worker. Η επιλογή αυτή είναι κατάλληλη για το συγκεκριμένο proof-of-concept, καθώς επιτρέπει την εκτέλεση federated queries χωρίς την πολυπλοκότητα ενός πολυκόμβου cluster. Στο `config.properties` δηλώθηκαν `coordinator=true` και `node-scheduler.include-coordinator=true`, ώστε ο coordinator να συμμετέχει και στην εκτέλεση των εργασιών. Παράλληλα, η υπηρεσία discovery ορίστηκε στο `http://127.0.0.1:8080`, δηλαδή σε τοπικό endpoint.

Το `node.properties` χρησιμοποιείται για την ταυτοποίηση του κόμβου και τον καθορισμό του runtime directory. Στην εγκατάσταση χρησιμοποιούνται `node.environment=thesis`, μοναδικό `node.id` και `node.data-dir=/var/trino/data`. Η αποθήκευση του runtime state εκτός του installation directory `/opt/trino` διαχωρίζει τα δεδομένα λειτουργίας από τα αρχεία λογισμικού και διευκολύνει τη συντήρηση και πιθανή αναβάθμιση. Η διαχείριση μνήμης πραγματοποιείται μέσω του `jvm.config`. Το τεκμηριωμένο demo χρησιμοποιεί μέγιστο `heap -Xmx2G`, garbage collector `G1GC` και παραμέτρους για `heap dump` και τερματισμό σε περίπτωση `OutOfMemoryError`. Οι τιμές αυτές αποτελούν συντηρητική ρύθμιση για το πειραματικό περιβάλλον και όχι γενικευμένη production διαμόρφωση.

Σε επίπεδο δικτυακής πρόσβασης, η HTTP θύρα 8080 περιορίζεται σε τοπική διοικητική χρήση, ενώ οι τελικοί χρήστες προσπελούν τον Trino μέσω HTTPS στη θύρα 8443. Το `log.properties` χρησιμοποιείται για τον καθορισμό του επιπέδου καταγραφής, ενώ η λειτουργία του server επαληθεύεται μέσω `launcher status`, `logs`, `ss` και ελέγχου του HTTPS endpoint. Με αυτή τη διαμόρφωση δημιουργείται ένα ελεγχόμενο και επαναλήψιμο runtime περιβάλλον για τα επόμενα στάδια διασύνδεσης των πηγών.

5.2.2 Catalogs, connectors και σύνδεση με ετερογενείς πηγές

Η ενσωμάτωση των ετερογενών πηγών στο Trino πραγματοποιήθηκε μέσω ανεξάρτητων catalogs, καθένα από τα οποία αντιστοιχεί σε διαφορετικό connector και σύστημα προέλευσης. Στην τελική υλοποίηση χρησιμοποιήθηκαν τα catalogs `mysql`, `modip_stats` και `ldap_exports`, τα οποία εκθέτουν αντίστοιχα τη MySQL, την PostgreSQL/MOΔΠ και τις εξαγωγές CSV του LDAP μέσω του Hive connector. Η χρήση διακριτών catalogs επιτρέπει σαφή ονοματοδοσία και πλήρως προσδιορισμένη πρόσβαση της μορφής `catalog.schema.object`.

Για τη MySQL δημιουργήθηκε το αρχείο `/opt/trino/etc/catalog/mysql.properties` με `connector.name=mysql` και JDBC σύνδεση προς `127.0.0.1:3307`. Η τοπική αυτή θύρα αποτελεί άκρο SSH tunnel προς το απομακρυσμένο Unix socket της MySQL, ενώ ο connector χρησιμοποιεί λογαριασμό βάσης μόνο για ανάγνωση. Η σύνδεση επαληθεύτηκε μέσω `SHOW SCHEMAS`, `SHOW TABLES`, `DESCRIBE` και `SELECT count(*)`, με τα `mysql.unitron.courses` και `mysql.unitron.professors` να εμφανίζονται επιτυχώς.

Πίνακας 5.2: Catalogs και μηχανισμοί σύνδεσης

Catalog	Connector	Endpoint / Path	Κύριο schema
mysql	MySQL	127.0.0.1:3307 μέσω SSH tunnel	unitron
modip_stats	PostgreSQL	127.0.0.1:2000 μέσω SSH tunnel	public
ldap_exports	Hive	/mnt/ldap-exports μέσω SSHFS	directory

Αντίστοιχα, το catalog modip_stats χρησιμοποιεί τον PostgreSQL connector και JDBC URL προς 127.0.0.1:2000, το οποίο προωθείται μέσω ανεξάρτητου SSH tunnel στην πραγματική PostgreSQL/MOΔΠΠ στη θύρα 5432. Τα πραγματικά διαπιστευτήρια βάσης δεδομένων παραμένουν εκτός της διπλωματικής και το catalog file προστατεύεται με περιορισμένα filesystem permissions. Για τα LDAP CSV δεν χρησιμοποιήθηκε πλήρες Hive cluster, αλλά ο Hive connector του Trino με file-based metastore. Το catalog ldap_exports χαρτογραφεί το read-only SSHFS mount /mnt/ldap-exports σε external table και σταθερά views staff και students, με hive.security=read-only. Η ορθή σύνδεση κάθε πηγής επιβεβαιώθηκε με connectivity tests, schema discovery και ελέγχους διαθεσιμότητας δεδομένων. Τα αποτελέσματα της πειραματικής αξιολόγησης χαρακτηρίζουν αυτές τις δοκιμές ως επιτυχείς.

5.3 Διασύνδεση και επαλήθευση των πηγών δεδομένων

Η διασύνδεση των πηγών πραγματοποιήθηκε σταδιακά, με ανεξάρτητο έλεγχο κάθε συστήματος πριν από την εκτέλεση federated queries. Στην τελική υλοποίηση, ο Trino λειτουργεί ως κοινό επίπεδο πρόσβασης προς τρεις κατηγορίες πηγών: MySQL, PostgreSQL/MOΔΠΠ και εξαγωγές CSV του LDAP μέσω SSHFS και Hive. Η ορατότητα των πηγών ελέγχθηκε από το Trino CLI μέσω SHOW CATALOGS και επιμέρους ερωτημάτων ανά catalog. Η MySQL προσπελάζεται μέσω του catalog mysql και του schema unitron. Στην αρχική proof-of-concept μοντελοποίηση περιλαμβάνονταν οι πίνακες courses, professors, students και exam_schedule, οι οποίοι αναπαριστούσαν αντίστοιχα μαθήματα, διδάσκοντες, φοιτητές και πρόγραμμα εξεταστικής. Η αναγνώρισή τους από το Trino είχε επιβεβαιωθεί μέσω SHOW TABLES. Στην τελική εγκατάσταση, η διαθεσιμότητα της MySQL ελέγχεται ενδεικτικά με SELECT count(*) FROM mysql.unitron.courses.

Η PostgreSQL/MOΔΠΠ εκτίθεται μέσω του catalog modip_stats και του schema public. Η σύνδεση πραγματοποιείται μέσω ανεξάρτητου SSH tunnel και επαληθεύεται με ερωτήματα σε πραγματικά αντικείμενα της βάσης, όπως το modip_stats.public.course. Η επιλογή αυτή επιβεβαιώνει ότι ο Trino μπορεί να προσπελάσει τα δεδομένα αξιολογήσεων χωρίς να απαιτείται μεταφορά τους σε κεντρική βάση. Τα στοιχεία φοιτητών και προσωπικού από LDAP διατίθενται ως CSV αρχεία, προσαρτημένα read-only στο /mnt/ldap-exports. Ο Hive connector τα εκθέτει μέσω του catalog ldap_exports, του schema directory και των views staff και students. Η τεκμηριωμένη επαλήθευση περιλαμβάνει SHOW SCHEMAS, SHOW TABLES, SHOW CREATE VIEW και καταμετρήσεις εγγραφών. Στο καταγεγραμμένο αποτέλεσμα εμφανίζονται 452 εγγραφές staff και 5.554 εγγραφές students. Πριν από οποιοδήποτε cross-source join, επιβεβαιώνεται αυτοτελώς ότι κάθε πηγή είναι διαθέσιμη, προσβάσιμη και ορατή στο κοινό ομοσπονδιακό περιβάλλον του Trino.

5.4 Υλοποίηση ενοποιημένων και federated ερωτημάτων

5.4.1 Ερωτήματα ανά πηγή και βασικά παραδείγματα SQL

Πριν από την εκτέλεση federated joins πραγματοποιήθηκαν ανεξάρτητα ερωτήματα ανά πηγή, ώστε να επιβεβαιωθούν η ορατότητα των catalogs, η πρόσβαση στα schemas και η διαθεσιμότητα των δεδομένων. Η διαδικασία περιλάμβανε SHOW SCHEMAS, SHOW TABLES, DESCRIBE και απλά SELECT, επιβεβαιώνοντας ότι κάθε connector λειτουργεί αυτοτελώς πριν συμμετάσχει σε σύνθετο ερώτημα. Για τη MySQL χρησιμοποιήθηκαν οι πίνακες courses και professors. Ενδεικτικά, η προβολή συγκεκριμένων πεδίων και η εφαρμογή περιορισμού εγγραφών μπορούν να εκφραστούν ως:

SQL

```
SELECT code, title, semester, ects
FROM mysql.unitron.courses
ORDER BY code
LIMIT 10;
```

SQL

```
SELECT      professor_id,      first_name,      last_name,      email      FROM
mysql.unitron.professors
LIMIT 10;
```

Η τεκμηριωμένη επαλήθευση της MySQL κατέγραψε τέσσερα μαθήματα και τέσσερις καθηγητές.

Για την PostgreSQL/MOΔΙΠ χρησιμοποιούνται αντικείμενα του modip_stats.public, όπως course, questioner και vote. Βασικός έλεγχος διαθεσιμότητας είναι:

SQL

```
SELECT count(*) AS courses FROM modip_stats.public.course;
```

Το καταγεγραμμένο αποτέλεσμα περιλαμβάνει 116 μαθήματα, 642 ερωτηματολόγια και 485.884 εγγραφές vote.

Τα δεδομένα LDAP προσπελούνται μέσω Hive views:

SQL

```
SELECT uid, mail, title
FROM ldap_exports.directory.students
LIMIT 10;
```

και αντίστοιχα από ldap_exports.directory.staff. Οι καταμετρήσεις κατέγραψαν 5.554 φοιτητές και 452 μέλη προσωπικού.

Το exam_schedule ανήκει στο αρχικό συνθετικό schema της MySQL, αλλά δεν τεκμηριώνεται ως αντικείμενο της τελικής πραγματικής MySQL εγκατάστασης· συνεπώς δεν παρουσιάζεται εδώ ως

επιβεβαιωμένο query της τελικής υποδομής. Οι ανεξάρτητοι αυτοί έλεγχοι αποτελούν τη βάση για τα federated joins της επόμενης ενότητας.

5.4.2 Cross-source JOINS, aggregations και αναλυτικά ερωτήματα

Η ουσιαστική επιβεβαίωση της ομοσπονδιακής λειτουργίας πραγματοποιήθηκε με ερωτήματα που συνδυάζουν ταυτόχρονα δεδομένα από MySQL, PostgreSQL/MOΔΠ και LDAP CSV/Hive. Στο κύριο federated query, τα μαθήματα της MySQL συσχετίζονται αρχικά με τους διδάσκοντες μέσω professor_id. Στη συνέχεια, τα στοιχεία του διδάσκοντα αντιστοιχίζονται με το LDAP view staff μέσω κανονικοποιημένου email (lower(trim(email))), ενώ τα μαθήματα συνδέονται με τα δεδομένα MOΔΠ χρησιμοποιώντας κανονικοποιημένο τίτλο και εξάμηνο. Η επιλογή αυτή αποφεύγει την εξάρτηση από μη επιβεβαιωμένο κοινό course identifier.

Τα δεδομένα αξιολόγησης δεν ανακτώνται ως απλές εγγραφές. Πριν από το cross-source join δημιουργείται συγκεντρωτικό σύνολο από τους πίνακες course, questioner και vote της MOΔΠ. Για κάθε μάθημα υπολογίζονται count(v.score) ως αριθμός απαντήσεων, count(DISTINCT v.user_id) ως αριθμός συμμετεχόντων και avg(CAST(v.score AS DOUBLE)) ως μέση αξιολόγηση. Έτσι, το Trino συνδυάζει relational joins και aggregations διαφορετικών πηγών μέσα στο ίδιο SQL execution context. Η αναλυτική επεξεργασία επεκτάθηκε με window functions. Στο query κατάταξης χρησιμοποιήθηκε dense_rank() OVER (ORDER BY average_score DESC) για ταξινόμηση των μαθημάτων και avg(average_score) OVER () για υπολογισμό της απόκλισης κάθε μαθήματος από τον μέσο όρο του επιλεγμένου συνόλου. Το αποτέλεσμα κατέταξε τέσσερα μαθήματα με μέσους όρους από 3,76 έως 4,04, ενώ η εκτέλεση ολοκληρώθηκε σε 1,52 s πάνω σε 488K input rows και 48,8 MiB. Η προέλευση των πεδίων παραμένει διακριτή: κωδικός, τίτλος, εξάμηνο και ECTS προέρχονται από MySQL, τα στοιχεία διδάσκοντα από professors, η ταυτοποίηση λογαριασμών από LDAP και τα στατιστικά αξιολόγησης από MOΔΠ. Συνεπώς, τα ερωτήματα αποδεικνύουν πραγματική λογική ενοποίηση χωρίς φυσική συγκέντρωση των δεδομένων.

Αντιπροσωπευτικό απόσπασμα SQL

```
INNER JOIN modip_course_stats ms
  ON ms.modip_semester = uc.semester
  AND lower(trim(ms.modip_title)) = lower(trim(uc.title))
SELECT
  dense_rank() OVER (ORDER BY average_score DESC) AS evaluation_rank,
  course_code,
  course_title,
  respondents,
  evaluation_answers,
  round(average_score, 2) AS average_score
FROM federated_courses
ORDER BY evaluation_rank, course_code;
```

Πίνακας 5.3: Συγκεντρωτικό αποτέλεσμα αναλυτικού federated query

Rank	Κωδικός	Μάθημα	Respondents	Απαντήσεις	M.O.
1	1801	Ασφάλεια Πληροφοριακών Συστημάτων	22	1.401	4,04
2	1505	Αλληλεπίδραση Ανθρώπου-Μηχανής	31	2.699	4,01
3	1403	Εισαγωγή στα Λειτουργικά Συστήματα	182	23.514	3,99
4	1101	Μαθηματικά I	64	5.348	3,76

5.5 Σενάρια χρήσης και ανάλυση αποτελεσμάτων

Τα σενάρια χρήσης σχεδιάστηκαν ώστε να αποτυπώνουν πρακτικά την αξία της ενοποιημένης πρόσβασης και όχι απλώς τη λειτουργία των επιμέρους connectors. Το πρώτο σενάριο αφορά στοιχεία φοιτητών. Στο αρχικό συνθετικό MySQL υπήρχε ανεξάρτητος πίνακας students, ενώ στην πληρέστερη υλοποίηση τα αντίστοιχα directory δεδομένα εκτίθενται μέσω του ldap_exports.directory.students. Έτσι, το Trino παρέχει κοινή SQL πρόσβαση χωρίς άμεση διαχείριση του αρχικού CSV από τον χρήστη. Δεύτερο σενάριο αποτελεί η συσχέτιση μαθημάτων με διδάσκοντες. Η σχέση courses.professor_id = professors.professor_id επιτρέπει την ανάκτηση του μαθήματος μαζί με τα στοιχεία του υπεύθυνου καθηγητή. Η πληροφορία αυτή χρησιμοποιείται περαιτέρω στα federated queries, όπου τα ακαδημαϊκά δεδομένα μπορούν να συσχετιστούν με στοιχεία LDAP και αξιολογήσεις ΜΟΔΙΠ.

Η ανάλυση της εξεταστικής περιόδου τεκμηριώνεται στο αρχικό demo μέσω του exam_schedule, ο οποίος συνδέεται με τα μαθήματα μέσω course_id και περιλαμβάνει ημερομηνία, ώρα, αίθουσα και εξεταστική περίοδο. Τα διαθέσιμα αρχεία περιέχουν τέσσερις συνθετικές εγγραφές για τον Ιούνιο 2026. Το συγκεκριμένο use case πρέπει επομένως να παρουσιάζεται ως μέρος της αρχικής πιλοτικής μοντελοποίησης και όχι ως επαληθευμένο αντικείμενο της μεταγενέστερης πραγματικής MySQL υποδομής. Κεντρικό σενάριο του Data Space αποτελεί η συσχέτιση μαθημάτων με δεδομένα αξιολόγησης. Το αρχικό federated query συνδυάζει MySQL και PostgreSQL/MODIP και επιστρέφει τίτλο, εξάμηνο, ECTS και δείκτες αξιολόγησης, ενώ δεύτερο query προσθέτει και τον διδάσκοντα. Η πληρέστερη υλοποίηση επεκτείνει τη λογική σε **τρεις πηγές**, MySQL, PostgreSQL/MODIP και LDAP CSV/Hive. Έτσι, το πληροφοριακό αποτέλεσμα προκύπτει δυναμικά από αυτόνομα συστήματα, επιβεβαιώνοντας τη βασική απαίτηση FR-07 για πραγματικά cross-source ερωτήματα.

Πίνακας 5.4: Συνοπτική αποτύπωση σεναρίων χρήσης

Σενάριο	Πηγές	Πληροφοριακό αποτέλεσμα	Κατάσταση τεκμηρίωσης
Στοιχεία φοιτητών	MySQL / εξαγωγές LDAP	Ενοποιημένη προβολή στοιχείων φοιτητών	Τεκμηριωμένο ανά πηγή
Μαθήματα–διδάσκοντες	MySQL	Μάθημα με υπεύθυνο καθηγητή	Τεκμηριωμένο
Εξεταστική περίοδος	exam_schedule + courses	Ημερομηνία, ώρα, αίθουσα, μάθημα	Αρχικό synthetic demo
Μαθήματα–αξιολογήσεις	MySQL + MODIP	Ακαδημαϊκά και αξιολογικά στοιχεία	Τεκμηριωμένο federated query
Πολλαπλές πηγές	MySQL + MODIP + LDAP	Συνδυαστική ακαδημαϊκή εικόνα	Τεκμηριωμένη τελική υλοποίηση

Έλεγχος συνέπειας: η ασυνέπεια 420/1801 επηρεάζει μόνο την αρχική εκδοχή του join courses.code = course_evaluations.course_code και δεν πρέπει να χρησιμοποιηθεί ως αδιαμφισβήτητο αποτέλεσμα χωρίς επαλήθευση της πραγματικής τιμής στη MySQL.

5.6 Πειραματική αξιολόγηση

5.6.1 Απόδοση ερωτημάτων και χρόνος απόκρισης

Η αξιολόγηση της απόδοσης βασίστηκε στις μετρικές που καταγράφει το Trino κατά την εκτέλεση των πραγματικών ερωτημάτων. Για τα σύνθετα federated workloads εξετάζονται ο συνολικός χρόνος εκτέλεσης, ο αριθμός των input rows, ο όγκος των δεδομένων που επεξεργάστηκε το query, ο ρυθμός

επεξεργασίας και ο αριθμός των splits. Οι μετρικές αυτές επιτρέπουν την αποτίμηση όχι μόνο του τελικού χρόνου απόκρισης αλλά και του πραγματικού υπολογιστικού φορτίου.

Το αναλυτικό federated query της προηγούμενης ενότητας συνδύασε MySQL, LDAP/Hive και PostgreSQL/MOΔΠΙ, περιλαμβάνοντας joins, aggregation και window functions. Η καταγεγραμμένη εκτέλεση ολοκληρώθηκε σε **1,52 s**, επεξεργάστηκε περίπου **488K input rows** και **48,8 MiB**, με ρυθμό **322K rows/s** και ολοκλήρωση **48/48 splits**. Τα στοιχεία αυτά αποτελούν πραγματικό output της υλοποίησης και δείχνουν ότι ένα σύνθετο αναλυτικό cross-source query ήταν λειτουργικά εκτελέσιμο με απόκριση της τάξης των δευτερολέπτων στο συγκεκριμένο περιβάλλον. Η μεγαλύτερη πολυπλοκότητα των federated queries προκύπτει από την ανάγκη συνδυασμού δεδομένων που προέρχονται από διαφορετικούς connectors, καθώς και από τις πρόσθετες πράξεις join, normalization και aggregation. Επομένως, η επίδραση των JOINS δεν πρέπει να αξιολογείται μόνο από τον τελικό χρόνο, αλλά σε συνδυασμό με τον όγκο εισόδου και τις επιμέρους εργασίες του query plan.

Ωστόσο, τα διαθέσιμα αρχεία **δεν περιέχουν τεκμηριωμένη σειρά πολλαπλών επαναλήψεων του ίδιου query** ούτε ισοδύναμο σύνολο χρόνων για απλό και federated query. Συνεπώς δεν είναι επιστημονικά ορθό να υπολογιστούν ακόμη μέσος, διάμεσος, ελάχιστος, μέγιστος χρόνος ή τυπική απόκλιση. Η τιμή των 1,52 s πρέπει να αντιμετωπίζεται ως συγκεκριμένη καταγεγραμμένη εκτέλεση και όχι ως στατιστικό benchmark.

Πίνακας 5.5: Καταγεγραμμένες μετρικές σύνθετου federated query

Μετρική	Καταγεγραμμένη τιμή
Χρόνος εκτέλεσης	1,52 s
Input rows	488K
Δεδομένα εισόδου	48,8 MiB
Ρυθμός επεξεργασίας	322K rows/s
Splits	48/48

5.6.2 Επεκτασιμότητα, ευχρηστία και λειτουργικότητα

Η αξιολόγηση της επεκτασιμότητας επικεντρώνεται κυρίως στη δυνατότητα του συστήματος να ενσωματώνει πρόσθετες πηγές χωρίς ανασχεδιασμό της συνολικής αρχιτεκτονικής. Η χρήση ανεξάρτητων catalogs και connectors επιτρέπει την προσθήκη νέων DBMS ή άλλων αποθετηρίων με διακριτά αρχεία παραμετροποίησης, διατηρώντας παράλληλα την κοινή SQL διεπαφή του Trino. Η ίδια η αρχιτεκτονική του Trino υποστηρίζει coordinator-worker τοπολογία και μπορεί θεωρητικά να επεκταθεί οριζόντια με πρόσθετους workers, όμως η παρούσα εγκατάσταση υλοποιήθηκε και αξιολογήθηκε αποκλειστικά ως single-node proof-of-concept.

Ως προς τη λειτουργική ορθότητα, το πρακτικό μέρος καλύπτει τις απαιτήσεις FR-01–FR-10 μέσω συγκεκριμένων ελέγχων. Επιβεβαιώθηκαν η κοινή SQL πρόσβαση, η προσπέλαση PostgreSQL/MOΔΠΙ, MySQL και LDAP CSV, η εκτέλεση cross-source queries, η λειτουργία μόνο για ανάγνωση, η διαφοροποίηση δικαιωμάτων και η δυνατότητα επανασύνδεσης μέσω systemd services. Οι έλεγχοι αυτοί αποτελούν functional validation της αρχιτεκτονικής και όχι μέτρηση παραγωγικής κλιμάκωσης. Η ευχρηστία προκύπτει κυρίως από το γεγονός ότι ο χρήστης αλληλεπιδρά με μία ενιαία SQL διεπαφή και χρησιμοποιεί fully qualified names, χωρίς να χρειάζεται διαφορετικό client για κάθε πηγή. Η διαχείριση των catalogs παραμένει ευθύνη του διαχειριστή και επιτρέπει σαφή απομόνωση των

πηγών, ενώ το CLI και τα τυποποιημένα SQL commands διευκολύνουν τον έλεγχο και την αναπαραγωγή των δοκιμών.

Ο σημαντικότερος περιορισμός είναι ότι το single-node περιβάλλον δεν επιτρέπει ασφαλή συμπεράσματα για υψηλό concurrency, μεγάλης κλίμακας distributed execution ή production throughput. Παρ' όλα αυτά, για εκπαιδευτική και πιλοτική χρήση, το σύστημα αποδεικνύει ότι διαφορετικές τεχνολογίες μπορούν να συνδεθούν, να ερωτηθούν και να ελεγχθούν από κοινό query layer με επαναλήψιμη διαδικασία. Αυτό συνδέεται άμεσα με το RQ4, το οποίο εξετάζει εάν το proof-of-concept παρέχει επαρκή λειτουργικότητα και απόκριση για εκπαιδευτική επίδειξη federated queries.

Πίνακας 5.6: Συνοπτική λειτουργική αξιολόγηση

Διάσταση	Παρατήρηση
Επεκτασιμότητα	Προσθήκη νέων πηγών μέσω catalogs/connectors
Ευχρηστία	Ενιαία SQL διεπαφή και fully qualified names
Λειτουργικότητα	Κάλυψη FR-01–FR-10 μέσω ελέγχων
Περιορισμός	Single-node περιβάλλον, χωρίς production-scale validation
Καταλληλότητα	Ικανοποιητική για εκπαιδευτικό/πιλοτικό proof-of-concept

5.7 Προβλήματα υλοποίησης και τρόποι αντιμετώπισης

Κατά την υλοποίηση, τα κυριότερα προβλήματα δεν αφορούσαν μία μεμονωμένη τεχνολογία, αλλά τα σημεία διασύνδεσης μεταξύ Trino, δικτύου, SSHFS/Hive και μηχανισμών πρόσβασης. Η διάγνωση οργανώθηκε ανά επίπεδο, ώστε να απομονώνεται αν η αστοχία προερχόταν από υπηρεσία Linux, tunnel, filesystem, catalog ή SQL αντικείμενο. Το συνολικό health check περιλάμβανε έλεγχο των systemd services, των τοπικών θυρών 2000, 3307 και 8443, του mount /mnt/ldap-exports, των catalogs και βασικών SELECT count(*). Στο SSHFS εμφανίστηκαν δύο χαρακτηριστικές περιπτώσεις. Το Permission denied κατά την πρόσβαση με sudo ls αποδόθηκε σε αναμενόμενη συμπεριφορά FUSE και αντιμετωπίστηκε με έλεγχο των αρχείων ως χρήστης trino. Αντίστοιχα, το μήνυμα Transport endpoint is not connected υποδείκνυε διακοπή της SSHFS σύνδεσης· η διάγνωση γινόταν μέσω service status και journal, ενώ η υπηρεσία είχε οριστεί με Restart=on-failure, ώστε το mount να επανέρχεται αυτόματα.

Σε επίπεδο Hive, η απουσία του ldap_exports οδηγούσε σε έλεγχο του ldap_exports.properties, των property names της έκδοσης Trino 480 και του server.log. Σφάλματα Table not found απαιτούσαν έλεγχο του file metastore, του .trinoSchema, της διαδρομής schema και του πλήρους catalog.schema.table. Μηδενικό CSV count συνδεόταν με mount, external_location, headers ή \$path filters, ενώ αλλαγμένα columns αντιμετωπιζόνταν με σύγκριση του CSV header και επικαιροποίηση του Hive schema. Προβλήματα εξουσιοδότησης εξετάζονταν χωριστά από το authentication. Εάν student χρήστης έβλεπε προστατευμένα στατιστικά, ελεγχόταν η σειρά των table rules· εάν staff χρήστης λάμβανε Access Denied, ελέγχονταν eduPersonAffiliation, groups.txt και ο μηχανισμός refresh. Τέλος, connection refused σε MySQL ή PostgreSQL οδηγούσε πρώτα σε έλεγχο των SSH tunnel services και των αντίστοιχων local ports.

Πίνακας 5.7: Κύρια προβλήματα και αντιμετώπιση

Πρόβλημα	Αιτία / διάγνωση	Αντιμετώπιση
Permission denied στο SSHFS	FUSE permissions	Έλεγχος ως trino
Transport endpoint is not connected	Διακοπή SSHFS	service/journal + restart policy
ldap_exports δεν εμφανίζεται	Catalog/config mismatch	Έλεγχος properties και server.log
Table not found	Λανθασμένο metastore/schema path	Έλεγχος .trinoSchema και πλήρους ονόματος
CSV count = 0	Mount/header/location/filter	Έλεγχος mount και schema
Student βλέπει MODIP	Λάθος σειρά access rules	Διόρθωση ordered table rules
Staff λαμβάνει deny	Group mapping/refresh	Έλεγχος affiliation και groups.txt
DB connection refused	Tunnel/down local port	Έλεγχος systemd service και port

Κεφάλαιο 6: Συμπεράσματα και μελλοντικές προεκτάσεις

6.1 Ανακεφαλαίωση της διαδικασίας και βαθμός επίτευξης των στόχων

Η εργασία αντιμετώπισε το πρόβλημα της κατακερματισμένης πρόσβασης σε ετερογενή πανεπιστημιακά δεδομένα, διερευνώντας κατά πόσο ένα ακαδημαϊκό Data Space μπορεί να υλοποιηθεί με federated query engine χωρίς προηγούμενη συγκέντρωση όλων των δεδομένων σε κοινό αποθετήριο. Για τον σκοπό αυτό σχεδιάστηκε και υλοποιήθηκε proof-of-concept με Trino 480, στο οποίο ενσωματώθηκαν δεδομένα μαθημάτων και διδασκόντων από MySQL, αξιολογήσεις από PostgreSQL/MOΔΙΠ και στοιχεία φοιτητών και προσωπικού από εξαγωγές CSV του LDAP. Η μεθοδολογία ακολούθησε διαδοχικά στάδια ανάλυσης απαιτήσεων, σχεδιασμού αρχιτεκτονικής, παραμετροποίησης των πηγών και catalogs, υλοποίησης ασφαλούς πρόσβασης και εκτέλεσης λειτουργικών και ομοσπονδιακών ελέγχων. Η αρχιτεκτονική διατήρησε την αυτονομία των πηγών και χρησιμοποίησε SSH tunnels, SSHFS/Hive, HTTPS, OpenLDAP authentication, group mapping και read-only πολιτικές πρόσβασης. Οι λειτουργικές απαιτήσεις FR-01–FR-10 συνδέθηκαν με συγκεκριμένους μηχανισμούς υλοποίησης και αντίστοιχους ελέγχους.

Τα πειράματα επιβεβαίωσαν ότι η κοινή SQL διεπαφή μπορεί να προσπελάζει τις τρεις κατηγορίες πηγών και να εκτελεί πραγματικά cross-source queries. Επιβεβαιώθηκε επίσης η διαφοροποιημένη εξουσιοδότηση μεταξύ staff και student χρηστών, καθώς και η δυνατότητα περιοδικής ανανέωσης των εξαγωγών LDAP μέσω SSHFS. Για τα καταγεγραμμένα ερωτήματα αναφέρονται χρόνοι από 0,63 έως 1,52 s, χωρίς όμως πολλαπλές επαναλήψεις, concurrent workload ή production-scale benchmarking. Με βάση τα διαθέσιμα τεκμήρια, τα RQ1–RQ4 απαντώνται θετικά στο πλαίσιο του proof-of-concept. Το RQ5 υποστηρίζεται ως σχεδιασμένη λειτουργία μέσω ενεργοποιημένων systemd units και restart policies, αλλά απαιτεί ξεχωριστή soak/reboot δοκιμή για ισχυρότερη παραγωγική επιβεβαίωση. Συνεπώς, οι αρχικοί στόχοι επιτεύχθηκαν σε σημαντικό βαθμό για εκπαιδευτική και πιλοτική χρήση, με σαφείς περιορισμούς ως προς τη γενίκευση σε production περιβάλλον.

6.2 Κύρια ευρήματα και συμπεράσματα

Η πρακτική υλοποίηση επιβεβαίωσε ότι ένα πανεπιστημιακό Data Space μπορεί να βασιστεί σε ομοσπονδιακή αρχιτεκτονική, στην οποία ετερογενείς πηγές παραμένουν αυτόνομες αλλά προσπελούνται μέσω κοινής SQL διεπαφής. Στο proof-of-concept, το Trino συνδέσε MySQL, PostgreSQL/MOΔΙΠ και εξαγωγές CSV του LDAP μέσω Hive, χωρίς να απαιτηθεί δημιουργία ενιαίου κεντρικού αποθετηρίου ή μόνιμη αντιγραφή των δεδομένων. Το αποτέλεσμα αυτό αποτελεί το βασικό τεχνικό εύρημα της εργασίας και υποστηρίζει άμεσα το RQ1 και τη λειτουργική απαίτηση FR-07. Δεύτερο σημαντικό εύρημα αφορά την ελεγχόμενη πρόσβαση. Η αυθεντικοποίηση μέσω LDAP, το group mapping και οι ordered access-control rules επέτρεψαν διαφοροποίηση δικαιωμάτων, με επιτρεπόμενη πρόσβαση staff και απόρριψη μη εξουσιοδοτημένου student χρήστη στο προστατευμένο αντικείμενο MOΔΙΠ. Παράλληλα, εφαρμόστηκαν read-only connector accounts, read-only Hive/SSHFS και HTTPS, αποδεικνύοντας ότι η federation μπορεί να συνδυαστεί με βασικούς μηχανισμούς least privilege και προστασίας δεδομένων.

Η πειραματική αξιολόγηση έδειξε επίσης ότι τα καταγεγραμμένα queries ολοκληρώθηκαν σε χρόνους από 0,63 έως 1,52 s, με τα Q2–Q3 να επεξεργάζονται περίπου 488K rows / 48,8 MiB. Οι τιμές αυτές τεκμηριώνουν την καταλληλότητα του demo για διαδραστική εκπαιδευτική παρουσίαση, όχι όμως για production benchmarking. Δεν πραγματοποιήθηκαν πολλαπλές επαναλήψεις, cold/warm-cache

συγκρίσεις, concurrent-user tests ή network impairment tests. Συνολικά, τα RQ1–RQ4 υποστηρίζονται θετικά στο πλαίσιο του proof-of-concept, ενώ το RQ5 επιβεβαιώνεται κυρίως ως σχεδιασμένη λειτουργία μέσω systemd και restart policies και απαιτεί επιπλέον soak/reboot δοκιμές για ισχυρότερη τεκμηρίωση.

6.3 Πλεονεκτήματα και περιορισμοί της προσέγγισης Data Space

Το σημαντικότερο πλεονέκτημα της προτεινόμενης προσέγγισης είναι η **ενοποιημένη πρόσβαση χωρίς φυσική ενοποίηση των δεδομένων**. Οι πηγές MySQL, PostgreSQL/MOΔΠ και εξαγωγές LDAP παραμένουν στα αντίστοιχα συστήματα προέλευσης, ενώ το Trino λειτουργεί ως κοινό query layer. Με αυτόν τον τρόπο περιορίζεται η ανάγκη για εκτεταμένες διαδικασίες αντιγραφής ή ETL και διατηρείται μεγαλύτερος βαθμός αυτονομίας των επιμέρους πληροφοριακών συστημάτων.

Η χρήση catalogs και connectors ενισχύει επίσης την ευελιξία και την επεκτασιμότητα της αρχιτεκτονικής, καθώς επιτρέπει την προσθήκη νέων πηγών χωρίς ανασχεδιασμό ολόκληρου του συστήματος. Παράλληλα, τα federated queries παρέχουν τη δυνατότητα σύνθεσης ακαδημαϊκών, αξιολογικών και directory δεδομένων σε κοινά αποτελέσματα. Η ίδια σχεδιαστική επιλογή, ωστόσο, δημιουργεί εξάρτηση από τη διαθεσιμότητα και την απόκριση των απομακρυσμένων πηγών: η απώλεια tunnel, SSHFS mount ή database endpoint μπορεί να επηρεάσει άμεσα ένα cross-source query.

Ένας δεύτερος περιορισμός είναι η **σημασιολογική ετερογένεια**. Διαφορετικά schemas, identifiers και συμβάσεις ονοματοδοσίας απαιτούν κανονικοποίηση και λογικούς κανόνες αντιστοίχισης, οι οποίοι αυξάνουν την πολυπλοκότητα και μπορούν να δημιουργήσουν αμφισημίες. Η γνωστή ασυνέπεια 420/1801 αποτελεί χαρακτηριστικό παράδειγμα του κινδύνου που δημιουργείται όταν δεν υπάρχει σταθερό κοινό identifier. Επιπλέον, η απόδοση εξαρτάται από τις δυνατότητες pushdown των connectors, το κόστος distributed joins και την καθυστέρηση δικτύου. Η ασφάλεια είναι επίσης πιο σύνθετη, επειδή πρέπει να συντονίζονται HTTPS, LDAP authentication, group mapping, database permissions και table-level authorization. Τέλος, η αξιολόγηση πραγματοποιήθηκε σε **single-node proof-of-concept**. Συνεπώς, τα αποτελέσματα δεν επιτρέπουν ασφαλή συμπεράσματα για επεκτασιμότητα σε παραγωγικό περιβάλλον, υψηλό concurrency ή συμπεριφορά σε μεγαλύτερα datasets και multi-node clusters.

6.4 Δυνατότητα εφαρμογής σε πραγματικό πανεπιστημιακό περιβάλλον

Η μετάβαση από το παρόν proof-of-concept σε πραγματικό πανεπιστημιακό περιβάλλον είναι τεχνικά εφικτή, αλλά απαιτεί ουσιαστική ενίσχυση της υποδομής, της ασφάλειας και της διακυβέρνησης. Η βασική αρχιτεκτονική μπορεί να διατηρηθεί, καθώς η λογική ομοσπονδιακής πρόσβασης μέσω Trino επιτρέπει στα επιμέρους πληροφοριακά συστήματα να παραμένουν αυτόνομα. Ωστόσο, το υφιστάμενο single-node περιβάλλον και το file-based metastore έχουν σχεδιαστεί για πιλοτική χρήση και δεν επαρκούν για multi-node παραγωγική λειτουργία. Σε αυτή την περίπτωση απαιτούνται shared metastore και κοινός αποθηκευτικός χώρος προσβάσιμος από όλους τους workers.

Ιδιαίτερη προτεραιότητα έχει η ενίσχυση της ασφάλειας. Η προσωρινή χρήση ldap:// με ldap.allow-insecure=true, η οποία εφαρμόστηκε λόγω ληγμένου πιστοποιητικού, πρέπει να αντικατασταθεί από έγκυρο LDAPS ή StartTLS. Τα database accounts οφείλουν να διατηρούν αποκλειστικά τα απολύτως απαραίτητα read permissions, ενώ SSH keys και catalog passwords δεν πρέπει να αποθηκεύονται σε

δημόσια repositories ή τεχνικά παραρτήματα. Σε πραγματική λειτουργία απαιτείται επίσης αυστηρότερη διακυβέρνηση προσωπικών δεδομένων. Οι εξαγωγές LDAP περιλαμβάνουν PII, επομένως χρειάζονται data minimization, auditing και, όπου απαιτείται, column ή row masking. Παράλληλα, η αντιστοίχιση χρηστών και μαθημάτων με email ή τίτλους δεν επαρκεί ως μακροχρόνια παραγωγική λύση· προτιμώνται canonical person και course identifiers. Η παραγωγική μετάβαση προϋποθέτει θεσμική διακυβέρνηση: καθορισμό data owners, πολιτικών πρόσβασης, διαδικασιών αλλαγής schemas, audit trails και συμμόρφωση με το ισχύον πλαίσιο προστασίας προσωπικών δεδομένων.

6.5 Προτάσεις βελτίωσης και μελλοντική επέκταση του συστήματος

Η σημαντικότερη τεχνική επέκταση αφορά τη μετάβαση από single-node σε multi-node Trino με διακριτούς coordinator και workers. Η αλλαγή αυτή θα επέτρεπε πραγματική αξιολόγηση κατανομημένης εκτέλεσης και μεγαλύτερων φορτίων, αλλά απαιτεί shared metastore και κοινό storage. Παράλληλα, για οι εξαγωγές LDAP θα ήταν σκόπιμη η μετάβαση από CSV σε Parquet ή οργανωμένη data-lake αποθήκευση, ώστε να περιοριστεί το parsing cost και να βελτιωθούν οι δυνατότητες predicate pushdown.

Στον τομέα της ασφάλειας, προτεραιότητα αποτελεί η αντικατάσταση της προσωρινής insecure LDAP σύνδεσης με έγκυρο LDAPS/StartTLS και η ενίσχυση των μηχανισμών authorization, auditing και προστασίας PII. Οι υφιστάμενοι file-based κανόνες μπορούν να εξελιχθούν σε κεντρικότερο policy model με λεπτομερέστερο έλεγχο πρόσβασης, data masking και συστηματική καταγραφή ενεργειών.

Η σημασιολογική ενοποίηση μπορεί επίσης να βελτιωθεί μέσω canonical identifiers για πρόσωπα και μαθήματα, περιορίζοντας false negatives και διπλοεγγραφές που μπορούν να προκύψουν από joins με email ή τίτλους. Πάνω από τα επιμέρους catalogs θα μπορούσε να προστεθεί metadata/catalog service και semantic layer, ώστε οι χρήστες να ανακαλύπτουν, να ερμηνεύουν και να επαναχρησιμοποιούν ευκολότερα τα διαθέσιμα datasets. Η ανάγκη για metadata management, lineage και governance έχει ήδη αναδειχθεί στο θεωρητικό μέρος της εργασίας.

Σε επίπεδο λειτουργίας, απαιτούνται οργανωμένο monitoring, κεντρικά logs, health checks, αυτοματοποίηση deployments και μηχανισμοί υψηλής διαθεσιμότητας. Η πειραματική αξιολόγηση πρέπει επίσης να επεκταθεί με πολλαπλές επαναλήψεις, ελεγχόμενα benchmarks, μεγαλύτερα datasets και concurrent workloads, καθώς οι σημερινές μεμονωμένες μετρήσεις δεν παρέχουν κατανομή latency. Τέλος, το Data Space μπορεί να επεκταθεί με πρόσθετα πανεπιστημιακά πληροφοριακά συστήματα, όπως βιβλιοθήκες, ερευνητικά αποθετήρια ή διοικητικές εφαρμογές, εφόσον προηγηθούν σαφείς πολιτικές governance και σημασιολογικής αντιστοίχισης.

Βιβλιογραφία

- [1] M. Jarke και C. Quix, ‘Federated Data Integration in Data Spaces’, στο Designing Data Spaces, B. Otto, M. Ten Hompel, και S. Wrobel, Επιμ., Cham: Springer International Publishing, 2022, σελ. 181–194. doi: 10.1007/978-3-030-93975-5_11.
- [2] Z. Gu κ.ά., ‘A systematic overview of data federation systems’, Semantic Web, τ. 15, τχ. 1, σελ. 107–165, Ιανουαρίου 2024, doi: 10.3233/SW-223201.
- [3] V. Nelamangala, ‘ADVANCES IN DATA FEDERATION AND VIRTUALIZATION: ENABLING UNIFIED DATA ACCESS’, Int. J. Comput. Eng. Technol., τ. 16, τχ. 1, σελ. 1609–1619, Ιανουαρίου 2025, doi: 10.34218/IJCET_16_01_118.
- [4] E. Curry, S. Scerri, και T. Tuikka, Επιμ., Data Spaces: Design, Deployment and Future Directions. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-98636-0.
- [5] M. Koripalli, ‘Cross-Platform Data Management: Patterns and Best Practices’, Eur. J. Comput. Sci. Inf. Technol., τ. 13, τχ. 10, σελ. 57–72, Μαρτίου 2025, doi: 10.37745/ejcsit.2013/vol13n105772.
- [6] I. M. Putrama και P. Martinek, ‘Heterogeneous data integration: Challenges and opportunities’, Data Brief, τ. 56, σελ. 110853, Οκτωβρίου 2024, doi: 10.1016/j.dib.2024.110853.
- [7] M. A. Vieira, E. L. F. Ribeiro, D. B. Claro, και B. Mane, ‘Integration Model between Heterogeneous Data Services in a Cloud’, JUCS - J. Univers. Comput. Sci., τ. 27, τχ. 4, σελ. 387–412, Απριλίου 2021, doi: 10.3897/jucs.67046.
- [8] Z. Gao κ.ά., ‘Investigation into the Integration of Intranet and Extranet Data in Data Centers’, Ann. Emerg. Technol. Comput., τ. 9, τχ. 2, σελ. 66–77, Απριλίου 2025, doi: 10.33166/AETiC.2025.02.005.
- [9] A. A. Fareedi, S. Gagnon, A. Ghazawneh, και R. Valverde, ‘Semantic Fusion of Health Data: Implementing a Federated Virtualized Knowledge Graph Framework Leveraging Ontop System’, Future Internet, τ. 17, τχ. 6, σελ. 245, Μαΐου 2025, doi: 10.3390/fi17060245.
- [10] H. Bian, D. Geng, Y. Chai, και A. Ailamaki, ‘Serverless Query Processing with Flexible Performance SLAs and Prices’, 2024, arXiv. doi: 10.48550/ARXIV.2409.01388.
- [11] Akash Vijayrao Chaudhari και Pallavi Ashokrao Charate, ‘Optimizing Data Lakehouse Architectures for Scalable Real-Time Analytics’, Int. J. Sci. Res. Sci. Eng. Technol., τ. 12, τχ. 2, σελ. 809–822, Απριλίου 2025, doi: 10.32628/IJSRSET25122198.
- [12] Venkata Surendra Reddy Appalapuram, ‘The Lakehouse Paradigm: Converging Data Lakes and Warehouses for Integrated Enterprise Analytics’, J. Comput. Sci. Technol. Stud., τ. 7, τχ. 4, σελ. 641–648, Μαΐου 2025, doi: 10.32996/jcsts.2025.7.4.75.
- [13] ‘A Comparative Study of Delta Parquet, Iceberg, and Hudi for Automotive Data Engineering Use Cases’, Int. J. Comput. Sci. Eng., τ. 12, τχ. 7, Ιουλίου 2025, doi: 10.14445/23488387/IJCSE-V12I17P104.
- [14] C. Urkudkar και S. Sharma, ‘Methods for Improving Query Performance Using Delta Lake’, Int. J. Latest Eng. Manag. Res. IJLEMR, τ. 10, τχ. 7, σελ. 9–16, Ιουλίου 2025, doi: 10.56581/IJLEMR.10.07.09-16.

- [15] D. Patel και P. Vaghasia, ‘A Comprehensive Study on the Big Data Adaptation and Analytics of Cloud Computing Ecosystems’, *Int. J. Multidiscip. Res.*, τ. 4, τχ. 5, σελ. 51729, Σεπτεμβρίου 2022, doi: 10.36948/ijfmr.2022.v04i05.51729.
- [16] R. Mafrur, ‘Blockchain Data Analytics: Review and Challenges’, 2025, arXiv. doi: 10.48550/ARXIV.2503.09165.
- [17] A. Gyrard κ.ά., ‘Lessons Learned From European Health Data Projects With Cancer Use Cases: Implementation of Health Standards and Internet of Things Semantic Interoperability’, *J. Med. Internet Res.*, τ. 27, σελ. e66273, Μαρτίου 2025, doi: 10.2196/66273.
- [18] M. Bacco, A. Kocian, S. Chessa, A. Crivello, και P. Barsocchi, ‘What are data spaces? Systematic survey and future outlook’, *Data Brief*, τ. 57, σελ. 110969, Δεκεμβρίου 2024, doi: 10.1016/j.dib.2024.110969.
- [19] S. Bonura, D. D. Carbonare, R. Díaz-Morales, Á. Navia-Vázquez, M. Purcell, και S. Rossello, ‘Increasing Trust for Data Spaces with Federated Learning’, στο *Data Spaces*, E. Curry, S. Scerri, και T. Tuikka, Επιμ., Cham: Springer International Publishing, 2022, σελ. 89–106. doi: 10.1007/978-3-030-98636-0_5.
- [20] D. Stripelis και J. L. Ambite, ‘Federated Learning over Harmonized Data Silos’, 2023, arXiv. doi: 10.48550/ARXIV.2305.08985.
- [21] M. Masmoudi, S. Ben Abdallah Ben Lamine, M. H. Karray, B. Archimede, και H. Baazaoui Zghal, ‘Semantic Data Integration and Querying: A Survey and Challenges’, *ACM Comput. Surv.*, τ. 56, τχ. 8, σελ. 1–35, Αυγούστου 2024, doi: 10.1145/3653317.
- [22] A. A. Taddese, A. C. Addis, και B. T. Tam, ‘Data stewardship and curation practices in AI-based genomics and automated microscopy image analysis for high-throughput screening studies: promoting robust and ethical AI applications’, *Hum. Genomics*, τ. 19, τχ. 1, σελ. 16, Φεβρουαρίου 2025, doi: 10.1186/s40246-025-00716-x.
- [23] R. Veerabhadrapa και P. A. Sampigerayappa, ‘Enhancing DNS-over-HTTPS Traffic Classification in Heterogeneous Networks Through Latent Space Analysis with a Tabular-Variational Autoencoder and Self-Attention Classifier Model’, *Artif. Intell. Appl.*, τ. 4, τχ. 1, σελ. 125–138, Αυγούστου 2025, doi: 10.47852/bonviewAIA52025552.
- [24] A. Anjomshoa κ.ά., ‘Data Platforms for Data Spaces’, στο *Data Spaces*, E. Curry, S. Scerri, και T. Tuikka, Επιμ., Cham: Springer International Publishing, 2022, σελ. 43–64. doi: 10.1007/978-3-030-98636-0_3.
- [25] S. Islam, A. Weber, και E. Tóth-Czifra, ‘From Green Deal to Cultural Heritage: FAIR Digital Objects and European Common Data Spaces’, *Res. Ideas Outcomes*, τ. 8, σελ. e93815, Αυγούστου 2022, doi: 10.3897/rio.8.e93815.
- [26] C. Song, S. Bouguerra, E. Krogen, και D. Abadi, ‘Evaluating Learned Query Performance Prediction Models at LinkedIn: Challenges, Opportunities, and Findings’, 2025, arXiv. doi: 10.48550/ARXIV.2504.17181.
- [27] R. Miyamoto και A. Kasuga, ‘Enterprise Data Science Platform: A Unified Architecture for Federated Data Access’, στο *2025 IEEE International Conference on Big Data (BigData)*, Macau, China: IEEE, Δεκεμβρίου 2025, σελ. 7395–7404. doi: 10.1109/BigData66926.2025.11401801.

- [28] T. Bodner, D. Ritter, M. Boissier, και T. Rabl, ‘Skyrise: Exploiting Serverless Cloud Infrastructure for Elastic Data Processing’, *Datenbank-Spektrum*, τ. 25, τχ. 1, σελ. 29–38, Μαρτίου 2025, doi: 10.1007/s13222-025-00496-7.
- [29] J. Frade, L. Santos, και R. L. D. C. Costa, ‘DBD-Guardian and Privacy-Aware Near Real-Time Cybersecurity Analytics’, *IEEE Access*, τ. 12, σελ. 149787–149803, 2024, doi: 10.1109/ACCESS.2024.3477979.
- [30] B. B. Lhoucine και K. Tatane, ‘Next-Generation Hybrid Databases: Bridging SQL Consistency, NoSQL Scalability, and AI-Driven Optimizations’, *Int. J. Eng. Comput. Sci.*, τ. 14, τχ. 09, σελ. 27724–27730, Σεπτεμβρίου 2025, doi: 10.18535/ijecs.v14i09.5264.
- [31] U. K. -, ‘How Apache Iceberg Outperforms Traditional and Hybrid Table Formats for Large-Scale Data Engineering’, *Int. J. Multidiscip. Res.*, τ. 6, τχ. 2, σελ. 50256, Μαρτίου 2024, doi: 10.36948/ijfmr.2024.v06i02.50256.
- [32] ‘BD4NRG QUERY ENGINE – INTUITIVE, EFFICIENT AND FEDERATED QUERYING ON BIG DATA’, στο *Proceedings of the International Conferences on E-Society 2023 and Mobile Learning 2023*, IADIS Press, Μαρτίου 2023, σελ. 111–118. doi: 10.33965/ES_ML2023_202301L014.