



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΠΜΣ “Ψηφιακά Συστήματα & Υπηρεσίες”
ΚΑΤΕΥΘΥΝΣΗ: Μεγάλα Δεδομένα & Αναλυτική

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΙΣΤΟΥ ΓΙΑ ΤΗ ΔΙΑΧΕΙΡΙΣΗ
ΑΛΛΗΛΕΠΙΔΡΑΣΕΩΝ ΦΑΡΜΑΚΩΝ ΜΕ ΤΗ ΧΡΗΣΗ
ΣΥΓΧΡΟΝΩΝ ΤΕΧΝΙΚΩΝ ΜΕΓΑΛΩΝ ΔΕΔΟΜΕΝΩΝ**

ΚΩΝΣΤΑΝΤΙΝΟΣ ΤΣΙΜΑΣ

Επιβλέπων:

Καθηγητής Γεώργιος Βασιλακόπουλος

ΦΕΒΡΟΥΑΡΙΟΣ 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ

ΠΜΣ: ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΥΠΗΡΕΣΙΕΣ

Κατεύθυνση: Μεγάλα Δεδομένα και Αναλυτική
(Big Data and Analytics)



ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη Εφαρμογής Ιστού για τη Διαχείριση Αλληλεπιδράσεων Φαρμάκων
με τη χρήση συγχρόνων τεχνικών μεγάλων δεδομένων

Εκπόνηση:

Κωνσταντίνος Τσίμας

Επιβλέπων:

Καθηγητής Γεώργιος Βασιλακόπουλος

Τριμελής Εξεταστική Επιτροπή:

Καθηγητής Γεώργιος Βασιλακόπουλος,
Αναπληρωτής Καθηγητής Μιχαήλ Φιλιππάκης
Επίκουρος Καθηγητής Δημοσθένης Κυριαζής

ΦΕΒΡΟΥΑΡΙΟΣ 2019

Ευχαριστίες

Η παρούσα εργασία αποτελεί διπλωματική εργασία στα πλαίσια του μεταπτυχιακού προγράμματος «Ψηφιακά Συστήματα & Υπηρεσίες» με κατεύθυνση «Μεγάλα Δεδομένα και Αναλυτική» του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς.

Πριν την παρουσίαση των αποτελεσμάτων της παρούσας διπλωματικής εργασίας, αισθάνομαι την υποχρέωση να ευχαριστήσω κάποιους ανθρώπους που γνώρισα, εξελίχθηκα μαζί τους και έπαιξαν καθοριστικό ρόλο στην πραγματοποίηση και ολοκλήρωσή της.

Πρώτα από όλους θα ήθελα να ευχαριστήσω τον Καθηγητή του Πανεπιστημίου Πειραιώς, Διευθυντή του Προγράμματος Μεταπτυχιακών Σπουδών και επιβλέποντα αυτής της εργασίας κ. Βασιλακόπουλο Γεώργιο για την αμέριστη βοήθειά του, τις υποδείξεις και την καθοδήγηση καθ' όλη την διάρκεια της εκπόνησης της διπλωματικής μου εργασίας.

Στη συνέχεια θα ήθελα να ευχαριστήσω όλους τους διδάσκοντες του μεταπτυχιακού προγράμματος οι οποίοι με τα πλούσια πνευματικά – ακαδημαϊκά προσόντα και το ήθος τους συνέβαλαν ουσιαστικά στην μεταπτυχιακή εκπαίδευσή μου και εμμέσως στην ολοκλήρωση της παρούσας εργασίας.

Τις ευχαριστίες μου εκφράζω και στον Αναπληρωτή Καθηγητή κ. Μιχαήλ Φιλιππάκη και στον Επίκουρο Καθηγητή κ. Δημοσθένη Κυριαζή που δέχτηκαν να είναι μέλη της τριμελούς επιτροπής αξιολόγησης της μεταπτυχιακής εργασίας.

Τέλος θα ήθελα να ευχαριστήσω τους δικούς μου ανθρώπους και κυρίως τη σύζυγό μου Ελένη και την κόρη μου Θεοδοσία για την υποστήριξή τους, την υπομονή τους και την κατανόηση τους για τον περιορισμένο χρόνο που τους αφιέρωσα, ώστε να ολοκληρώσω το μεταπτυχιακό πρόγραμμα σπουδών.

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη εφαρμογής ιστού για τις αλληλεπιδράσεις φαρμάκων με τη χρήση σύγχρονων τεχνικών μεγάλων δεδομένων. Στην σύγχρονη εποχή της πολυφαρμακίας, της αυξημένης μόλυνσης του περιβάλλοντος, της υπερκατανάλωσης τροφίμων-φαρμάκων και συμπληρωμάτων διατροφής, οι αλληλεπιδράσεις των φαρμάκων είναι στο επίκεντρο της Ιατρικής πληροφορικής και της Βιοπληροφορικής και η γνώση τους αποτελεί επιτακτική ανάγκη για τους επιστήμονες υγείας. Από την άλλη πλευρά τα ιατροφαρμακευτικά δεδομένα αυξάνονται με ραγδαίους ρυθμούς λόγω της εισόδου της τεχνολογίας στον τομέα της Ιατρικής. Στην χώρα μας αναμένεται τεράστια αύξηση των ιατροφαρμακευτικών δεδομένων καθώς έγινε υποχρεωτική η χρήση του Ατομικού Ηλεκτρονικού Φακέλου Υγείας (Α.Η.Φ.Υ.) και της ηλεκτρονικής συνταγογράφησης στο σύστημα Πρωτοβάθμιας Φροντίδας Υγείας. Έτσι γίνεται επιτακτική η ανάγκη νέων σύγχρονων εργαλείων και τεχνικών τόσο για τη διαχείριση των μεγάλων δεδομένων όσο και για την ανάπτυξη εφαρμογών ιστού για την υγεία.

Έτσι για την ανάλυση του σκοπού της εργασίας εισάγουμε στο Κεφάλαιο 2 τον όρο της «Αλληλεπίδρασης φαρμάκου», καθώς και τις κατηγορίες των αλληλεπιδράσεων. Στη συνέχεια στο Κεφάλαιο 3 αναλύουμε τον όρο «Μεγάλα δεδομένα» και αναφερόμαστε στα ιατροφαρμακευτικά δεδομένα. Στο Κεφάλαιο 4 παρουσιάζουμε τις χρησιμοποιούμενες τεχνικές και εργαλεία ανάλυσης και διαχείρισης μεγάλων δεδομένων. Για την δημιουργία της εφαρμογής ιστού αναλύουμε τα εργαλεία, τις υπηρεσίες ιστού, τις γλώσσες προγραμματισμού και πλατφόρμες ανάπτυξης εφαρμογών που χρησιμοποιήσαμε στο Κεφάλαιο 5, ορίζοντας την αρχιτεκτονική (υψηλού επιπέδου) και αναλύοντας την διαδικασία που ακολουθήσαμε για την επίτευξη του σκοπού μας στο επόμενο κεφάλαιο (Κεφάλαιο 6). Τέλος στο Κεφάλαιο 7 αναφέρουμε τα συμπεράσματα και τα επόμενα πιθανά βήματα.

Λέξεις Κλειδιά: Αλληλεπίδραση φαρμάκων, MongoDB, Apache Hadoop, Apache Spark, REST API.

Abstract

This dissertation presents the development of a web application for drug interactions using modern big data techniques. In the modern age of multifarmacy, increased environmental pollution, over-consumption of food-medicine and nutritional supplements, drug interactions are at the heart of Medical Informatics and Bioinformatics, and their knowledge is an imperative for health scientists. On the other hand, medical data are rising rapidly due to the entry of technology into the medical field. In our country a huge increase in medical data is expected as the use of the Personal Electronic Health File and the electronic prescription in the Primary Health Care System was mandatory. This makes it imperative to have new modern tools and techniques for managing large data as well as for developing web applications for health.

Thus, to analyze the purpose of the thesis, we enter in Chapter 2 the term "Drug Interaction", as well as the categories of interactions. Then in Chapter 3 we analyze the term "Big Data" and we refer to the medical data. In Chapter 4 we present the techniques and tools used to analyze and manage large data. In order to create the web application, we analyze the tools, web services, programming languages and application development platforms in Chapter 5, defining in the next chapter (Chapter 6) the architecture (high level) and analyzing the process we followed to achieve the goal us. Finally, in Chapter 7 we report the conclusions and the next possible steps.

Keywords: Drug Interaction, MongoDB, Apache Hadoop, Apache Spark, REST API.

Πίνακας περιεχομένων

Περίληψη	3
Abstract.....	4
Πίνακας περιεχομένων.....	5
Πίνακας εικόνων	7
1. Εισαγωγή	9
2. Αλληλεπίδραση φαρμάκου	10
2.1 Αλληλεπίδραση φαρμάκου – φαρμάκου (ων)	11
2.1.1 Φαρμακοκινητικές αλληλεπιδράσεις	11
2.1.2 Φαρμακοδυναμικές αλληλεπιδράσεις	13
2.1.3 Φαρμακευτικές αλληλεπιδράσεις	14
2.2 Αλληλεπίδραση τροφής – φαρμάκων	14
2.3 Αλληλεπίδραση φαρμάκου και ατομικοί παράγοντες	16
3. Ιατροφαρμακευτικά δεδομένα – Μεγάλα Δεδομένα	17
3.1 Μεγάλα δεδομένα	18
3.2 Ιατρικά δεδομένα	22
3.2.1 Πρότυπο HL7	24
3.2.2 XML	25
3.2.3 JSON.....	26
3.3 Φαρμακευτικά δεδομένα	26
Drugbank.....	27
4. Εργαλεία και τεχνικές διαχείρισης μεγάλων δεδομένων	28
4.1 NoSQL συστήματα διαχείρισης βάσεων δεδομένων.....	29
4.2 MongoDB.....	35

4.3	Υπολογιστικά Νέφη - Okeanos	39
4.4	Apache Hadoop – HDFS	40
4.5	Apache Spark	42
4.6	MongoDB Connectors	43
5.	Εφαρμογή Ιστού (Web Application).....	44
5.1	Υπηρεσίες Ιστού (Web Services).....	44
5.2	REST API	45
5.3	Postman.....	46
5.4	Γλώσσες προγραμματισμού	47
5.4.1	HTML (HyperText Markup Language) και HTML5	47
5.4.2	CSS (Cascading Style Sheets)	48
5.4.3	Javascript	48
5.4.4	React.js	49
5.4.5	Node.js – Express.js.....	49
6.	Αρχιτεκτονική και Διαδικασία	50
7.	Συμπεράσματα – Προοπτικές.....	58
	Βιβλιογραφία - Πηγές	59

Πίνακας εικόνων

Εικόνα 1 Αλληλεπιδράσεις φαρμάκων και αποτελέσματα	10
Εικόνα 2 Οι τρεις βασικές απαιτήσεις για την υλοποίηση ενός ολοκληρωμένου πληροφοριακού συστήματος.....	18
Εικόνα 3 Μοντέλο 3 Vs χαρακτηριστικών μεγάλων δεδομένων	19
Εικόνα 4 Μοντέλο 4 Vs χαρακτηριστικών μεγάλων δεδομένων	20
Εικόνα 5 Μοντέλο 6 Vs χαρακτηριστικών μεγάλων δεδομένων	21
Εικόνα 6 Βασικές πηγές από τις οποίες προέρχονται τα ιατρικά δεδομένα ενός ασθενούς	22
Εικόνα 7 Πρότυπο HL7 v3	24
Εικόνα 8 Παράδειγμα JSON	26
Εικόνα 9 Ιδιότητες ACID.....	30
Εικόνα 10 Θεώρημα CAP (Brewer).....	32
Εικόνα 11 Μορφή συνταγής φαρμάκων	35
Εικόνα 12 MongoDB - Replica Set.....	37
Εικόνα 13 Εκλογή πρωτεύοντα κόμβου - Replica set	38
Εικόνα 14 Παράδειγμα χρήσης MongoDB connector	43
Εικόνα 15 Κατηγορίες κωδικών κατάστασης.....	46
Εικόνα 16 Διεπαφή χρήστη Postman.....	47
Εικόνα 17 Αρχιτεκτονική Συστήματος	50
Εικόνα 18 Ιδιωτικό δίκτυο.....	51
Εικόνα 19 Σύμπλεγμα Hadoop.....	52

Εικόνα 20 MongoDB με κόμβο διαιτητή	53
Εικόνα 21 Σελίδα στοιχείων κάθε φαρμάκου	56
Εικόνα 22 Σελίδα αλληλεπιδράσεων ανά φάρμακο	56
Εικόνα 23 Σελίδα αλληλεπιδράσεων φαρμάκου- φαρμάκου	57
Εικόνα 24 Σελίδα αλληλεπιδράσεων φαρμάκου- τροφίμου	57

1. Εισαγωγή

Αντικείμενο της παρούσας εργασίας αποτελεί η ανάπτυξη εφαρμογής ιστού με τη χρήση σύγχρονων τεχνικών μεγάλων δεδομένων για τη διαχείριση των αλληλεπιδράσεων φαρμάκων.

Στην σημερινή εποχή η αλληλεπίδραση φαρμάκων βρίσκεται διεθνώς στο επίκεντρο της Ιατρικής Πληροφορικής και της Βιοπληροφορικής λόγω της αυξημένης χρήσης φαρμάκων και συμπληρωμάτων διατροφής καθώς και της αύξησης του αριθμού των φαρμάκων που χορηγούνται ανά άτομο - ασθενή. Η πολυφαρμακία, η υπερκατανάλωση τροφίμων και ποτών, η αυξημένη μόλυνση του περιβάλλοντος και ιδιαίτερα η ατμοσφαιρική ρύπανση και η ευρύτατη διάδοση του καπνίσματος αποτελούν τους σημαντικότερους λόγους, για τους οποίους η γνώση των αλληλεπιδράσεων είναι, πλέον, επιτακτική ανάγκη για τους επιστήμονες υγείας. Τα ανεπιθύμητα επεισόδια φαρμάκων παραμένουν η κύρια αιτία νοσηρότητας και θνησιμότητας σε ολόκληρο τον κόσμο και επιπλέον έχουν οδηγήσει σε πρόσθετο ιατρικό κόστος και αυξημένη νοσηλεία. Η αλληλεπίδραση φαρμάκων αναφέρεται σε ανεπιθύμητες ενέργειες που προκαλούνται από τον συνδυασμό δύο ή περισσότερων φαρμάκων, από τον συνδυασμό φαρμάκων - τροφής και από άλλους ατομικούς παράγοντες για κάθε ασθενή.

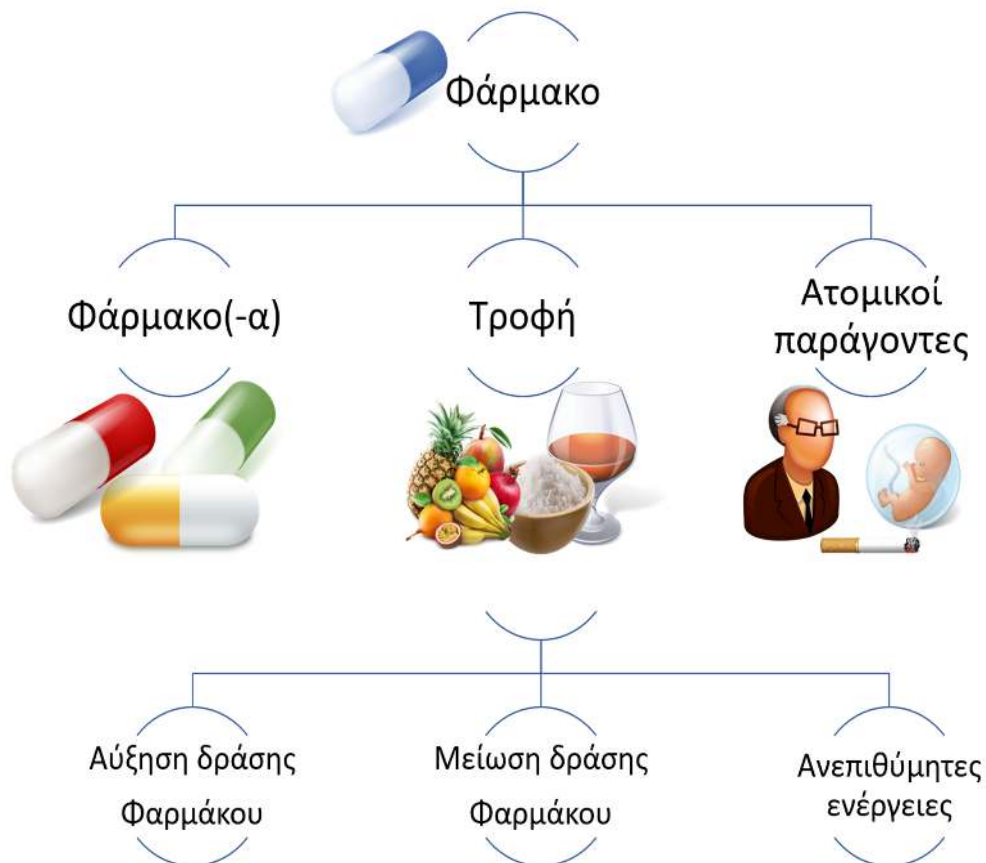
Από την άλλη πλευρά η τεχνολογία έχει εισχωρήσει στον τομέα της Ιατρικής με την ηλεκτρονική συνταγογράφηση και τον Ατομικό Ηλεκτρονικό Φάκελο Υγείας (Α.Η.Φ.Υ.). Από το 1960 εμφανίστηκαν οι πρώτοι ηλεκτρονικοί φάκελοι υγείας ενώ στην χώρα μας το 2014 [1] θεσπίστηκε με νόμο και από το 2019 γίνεται υποχρεωτική η ύπαρξη και χρήση των Α.Η.Φ.Υ. Από το 1980-85 οι Γερμανοί και Σουηδοί ξεκίνησαν την ηλεκτρονική συνταγογράφηση, ενώ στην χώρα μας ξεκίνησε από το 2010 [2]. Επιπλέον παγκοσμίως υπάρχουν και διατίθενται από διάφορους οργανισμούς (όπως U.S.A Food and Drug Administration, DrugBank, SNOMED International, European Medicines Agency) βάσεις δεδομένων που αφορούν τα φάρμακα, τις ανεπιθύμητες ενέργειες, τις αλληλεπιδράσεις τους, την χημική ανάλυσή τους, τα σκευάσματα στα οποία εμπεριέχονται και λοιπές πληροφορίες. Κι όσο ο όγκος όλων αυτών

των ιατρικών και φαρμακευτικών δεδομένων συνεχώς αυξάνεται καθημερινά, η επεξεργασία τους απαιτεί τη χρήση σύγχρονων εργαλείων και τεχνικών διαχείρισης μεγάλων δεδομένων.

2. Αλληλεπίδραση φαρμάκου

Με τον όρο «Αλληλεπίδραση φαρμάκου» αναφερόμαστε στην αντίδραση μεταξύ δύο ή περισσότερων φαρμάκων (drug-drug interactions) ή μεταξύ ενός φαρμάκου κι ενός τροφίμου (drug-food interactions), ενός ποτού ή ενός συμπληρώματος διατροφής. Η αλληλεπίδραση φαρμάκου εξαρτάται επίσης από ατομικούς παράγοντες του ασθενούς.

Μια αλληλεπίδραση φαρμάκου(ων) μπορεί να κάνει ένα φάρμακο λιγότερο αποτελεσματικό, να αυξήσει την δράση του ή να προκαλέσει ανεπιθύμητες ενέργειες. (Εικόνα 1)



Εικόνα 1 Αλληλεπιδράσεις φαρμάκων και αποτελέσματα

Στη συνέχεια γίνεται ανάλυση των κατηγοριών της αλληλεπίδρασης φαρμάκου και των αποτελεσμάτων κάθε κατηγορίας.

2.1 Αλληλεπίδραση φαρμάκου – φαρμάκου (ων)

Με τον όρο «Αλληλεπίδραση φαρμάκου - φαρμάκου(ων)» αναφερόμαστε σε κάθε μεταβολή των ιδιοτήτων ενός ή περισσότερων φαρμάκων από κάποιο άλλο ή άλλα φάρμακα που χρησιμοποιούνται ταυτόχρονα από κάποιον ασθενή, παράγοντας μια επίδραση διαφορετική από το άθροισμα των επιδράσεων που θα παρήγαγαν τα φάρμακα αυτά σε μεμονωμένη χρήση. Οι αλληλεπιδράσεις αυτές κατηγοριοποιούνται, ανάλογα με τον μηχανισμό πρόκλησής τους, σε αλληλεπιδράσεις εντός του οργανισμού και αφορούν τις φαρμακοκινητικές και φαρμακοδυναμικές παραμέτρους των φαρμάκων, στις φαρμακευτικές αλληλεπιδράσεις που εντοπίζονται συνήθως εκτός του οργανισμού, αλλά και εντός αυτού, και σε διάφορες αλληλεπιδράσεις, οι οποίες δεν μπορούν να ενταχθούν στις προηγούμενες κατηγορίες

2.1.1 Φαρμακοκινητικές αλληλεπιδράσεις

Στην κατηγορία αυτή περιλαμβάνεται το μεγαλύτερο ποσοστό αλληλεπιδράσεων φαρμάκων. Σχετίζονται με τις μεταβολές της συγκέντρωσης του φαρμάκου στο πλάσμα αίματος. Ένα φάρμακο επηρεάζει την ένταση και διάρκεια ενός άλλου εξαιτίας της μεταβολής της συγκέντρωσής του στον τόπο δράσης του. Ως τόπο δράσης αναφερόμαστε είτε στους τόπους απορρόφησης των φαρμάκων, είτε στους τόπους σύνδεσης των φαρμάκων με τις πρωτεΐνες του πλάσματος και ιστών, είτε στους τόπους μεταβολισμού των φαρμάκων, είτε στους τόπους απέκκρισης των φαρμάκων από τους νεφρούς.

2.1.1.1 Αλληλεπιδράσεις στους τόπους απορρόφησης

Οι αλληλεπιδράσεις αυτές αφορούν την ταχύτητα απορρόφησης ή το συνολικό ποσό του απορροφούμενου φαρμάκου, δηλαδή τη βιοδιαθεσιμότητά του και αναφέρονται σε:

1. Φάρμακα που αλλάζουν τις φυσιολογικές και χημικές ιδιότητες τους στην περιοχή του γαστρεντερικού συστήματος (ΓΕΣ)
2. Φάρμακα που επηρεάζουν το βλεννογόνο του εντέρου
3. Φάρμακα που επηρεάζουν μηχανισμούς μεταφοράς στο έντερο διαμέσου του εντερικού βλεννογόνου
4. Φάρμακα που μεταβάλλουν την εντερική μικροχλωρίδα
5. Φάρμακα που επηρεάζουν την κένωση του στομάχου ή την κινητικότητα του λεπτού εντέρου
6. Φάρμακα που επηρεάζουν την τοπική κυκλοφορία του αίματος

2.1.1.2 Αλληλεπιδράσεις στους τόπους σύνδεσης

Όταν δύο φάρμακα συναγωνίζονται για να δεσμεύσουν περιοχές στις πρωτεΐνες του πλάσματος, το ένα μπορεί να εκτοπίσει το άλλο και να οδηγήσει σε μια υψηλότερη συγκέντρωση του αδέσμευτου ελεύθερου φαρμάκου. Αυτό μπορεί να οδηγήσει σε αύξηση της έντασης και της διάρκειας της φαρμακολογικής του δράσης.

2.1.1.3 Αλληλεπιδράσεις στους τόπους μεταβολισμού

Αυτές οι αλληλεπιδράσεις είναι υπεύθυνες για την επιτάχυνση ή την επιβράδυνση του ρυθμού μεταβολισμού των φαρμάκων. Πολλά φάρμακα ή ουσίες του περιβάλλοντος μπορούν να αυξήσουν τη δραστηριότητα των ηπατικών ενζύμων με αποτέλεσμα την επιτάχυνση του μεταβολισμού τους αλλά και όσων φαρμάκων χορηγούνται ταυτόχρονα. Αντίθετα όταν ένα φάρμακο αναστέλλει τη δραστηριότητα του μεταβολικού συστήματος ή όταν εξαιτίας μεγαλύτερης συγγένειας έχει προτεραιότητα στη σειρά μεταβολισμού, τότε μπορεί να είναι υπεύθυνο για την επιβράδυνση του ρυθμού μεταβολισμού του συγχωρηγούμενου φαρμάκου το οποίο μπορεί να προκαλέσει τοξικές ενέργειες αν δεν ελαττωθούν οι δόσεις του.

2.1.1.4 Αλληλεπιδράσεις στους τόπους απέκκρισης

Οι περισσότερες αλληλεπιδράσεις μεταξύ φαρμάκων που αφορούν τα σημεία απέκκρισης από τους νεφρούς, σχετίζονται με μεταβολές της ενεργητικής απέκκρισης και της επαναρρόφησης των φαρμάκων αυτών. Η απέκκριση των φαρμάκων επηρεάζεται από το pH των ούρων. Τα φάρμακα βρίσκονται στα ουροφόρα σωληνάρια σε ιονισμένη και μη ιονισμένη μορφή. Μόνο η μη ιονισμένη μορφή είναι λιποδιαλυτή και μπορεί να επιστρέψει στο αίμα ενώ το pH των ούρων προσδιορίζει το ποσό κάθε μορφής φαρμάκου που υπάρχει στα σωληνάρια, άρα και το ποσό που θα απεκκριθεί και θα επαναπροσληφθεί. Μεταβολές του pH αυξάνουν την επαναπρόσληψη (όξινα ούρα για όξινα φάρμακα και αλκαλικά για τις βάσεις). Φάρμακα που χρησιμοποιούν το ίδιο σύστημα ενεργητικής μεταφοράς στα νεφρικά σωληνάρια μπορούν να συναγωνιστούν για την προτεραιότητα της απέκκρισης.

2.1.2 Φαρμακοδυναμικές αλληλεπιδράσεις

Οι φαρμακοδυναμικές αλληλεπιδράσεις αλλάζουν την απάντηση του οργάνου ή συστήματος στόχου. Τα φάρμακα μπορεί να μεταβάλουν το περιβάλλον της δράσης του άλλου ή ενεργούν στους ίδιους ή σε διαφορετικούς υποδοχείς και να προκαλούν αλληλεπιδράσεις που διακρίνονται σε:

1. Ανταγωνιστικές αλληλεπιδράσεις :
Ένα φάρμακο – ανταγωνιστής των υποδοχέων – ανταγωνίζεται τη δράση ενός φαρμάκου - αγωνιστή των υποδοχέων.
2. Προσθετικές ή συνεργιστικές αλληλεπιδράσεις:
Ταυτόχρονη χορήγηση φαρμάκων με ίδια δράση
3. Αλληλεπιδράσεις λόγω ηλεκτρολυτικών διαταραχών
4. Έμμεσες φαρμακοδυναμικές αλληλεπιδράσεις

2.1.3 Φαρμακευτικές αλληλεπιδράσεις

Αφορούν τις φυσικοχημικές ιδιότητες των φαρμάκων. Παρατηρούνται εκτός οργανισμού, όπως κατά την ανάμειξη δύο ή περισσότερων φαρμάκων σε σύριγγα ή φιάλη, αλλά και μέσα στον οργανισμό, όπως στην ίδια ενδοφλέβια οδό. Έχουν ως αποτέλεσμα την εξουδετέρωση των φαρμακολογικών ενεργειών των φαρμάκων ή τη δημιουργία παραγώγων ή συμπλοκών ουσιών, επικίνδυνων για τη ζωή του ασθενή. Οι αλληλεπιδράσεις αυτές, είναι αποτέλεσμα φυσικοχημικής ασυμβατότητας.

2.2 Αλληλεπίδραση τροφής – φαρμάκων

Η ανάμειξη φαρμάκων με τρόφιμα και ποτά μπορεί να προκαλέσει ποσοτικές και ποιοτικές μεταβολές στο αναμενόμενο θεραπευτικό αποτέλεσμα. Η τροφή μπορεί να επηρεάσει την απορρόφηση ορισμένων φαρμάκων. Συστατικά της τροφής σε συνδυασμό με φάρμακα μπορούν να προκαλέσουν σοβαρές ανεπιθύμητες ενέργειες ή να επηρεάσουν τη βιοδιαθεσιμότητα των φαρμάκων.

Η απορρόφηση πολλών αντιμικροβιακών και αντιφλεγμονωδών φαρμάκων όπως η ασπιρίνη μειώνεται σημαντικά με τη λήψη τροφής, παρόλο που χορηγούνται μετά από την πρόληψη τροφής, γιατί ερεθίζουν το γαστρικό βλεννογόνο προκαλώντας σε αντίθετη περίπτωση σοβαρές βλάβες.

Γαλακτοκομικά προϊόντα σε συνδυασμό με τετρακυκλίνες ή κινολόνες εμφανίζουν σημαντικές αλληλεπιδράσεις γεγονός που οφείλεται στο ασβέστιο που περιέχουν τα τρόφιμα αυτά, δημιουργώντας σύμπλοκα τα οποία δεν μπορούν να απορροφηθούν με αποτέλεσμα να μην επιτυγχάνονται θεραπευτικές συγκεντρώσεις των αντιμικροβιακών στο αίμα.

Ανεπαρκής ή υπερβολική λήψη νατρίου (αλατιού) ή καλίου μπορεί να επηρεάσουν το αναμενόμενο αποτέλεσμα ή να προκαλέσουν ηλεκτρολυτικές διαταραχές. Συνίσταται η αποφυγή και ο περιορισμός αλατιού σε ασθενείς που παίρνουν αντιυπερτασικά, διουρητικά ή κορτικοειδή. Επίσης σε κατηγορίες

ασθενών που λαμβάνουν καλιοσυντηρητικά διουρητικά δεν πρέπει να χρησιμοποιούνται συμπληρώματα καλίου γιατί υπάρχει σοβαρός κίνδυνος υπερκαλιαιμίας, που μπορεί να οδηγήσει ακόμα και σε θάνατο.

Ο χυμός grapefruit που χρησιμοποιείται κατά κόρον σε δίαιτες λόγω των λιποδιαλυτικών ιδιοτήτων του, περιέχει μια ουσία η οποία αναστέλλει το μεταβολισμό κάποιων φαρμάκων και αυξάνει τα επίπεδά τους με αποτέλεσμα να παρατείνεται η δράση τους καθώς και οι ανεπιθύμητες ενέργειες.

Ορισμένα φυσικά προϊόντα και θεραπευτικά βότανα που λαμβάνονται από αρκετά άτομα μπορεί να εμφανίσουν αλληλεπίδραση με φάρμακα. Ορισμένα από αυτά είναι το Ginkgo biloba, το οποίο σε συνδυασμό με αντιπηκτικά και αντιαιμοπεταλιακά φάρμακα αυξάνει τον κίνδυνο αιμορραγίας, το Ginseng που μπορεί να επιτείνει την υπογλυκαιμική δράση αντιδιαβητικών φαρμάκων, το σκόρδο που σε συνδυασμό με αντιπηκτικά και αντιαιμοπεταλιακά φάρμακα αυξάνει τον κίνδυνο αιμορραγίας, αλλά και η βαλεριάνα, η οποία επιτείνει τη δράση ορισμένων κατασταλτικών φαρμάκων του κεντρικού νευρικού συστήματος.

Πολλά λαχανικά και βότανα, όπως το κουνουπίδι, το μπρόκολο, το σπανάκι, το λάχανο και το πράσινο τσάι, είναι πλούσια σε βιταμίνη Κ, η οποία ευνοεί την πήξη του αίματος. Όταν τα τρόφιμα αυτά καταναλώνονται μαζί με αντιπηκτικά φάρμακα όπως η βαρφαρίνη (χορηγούνται για την αραιώση του αίματος και έτσι την πρόληψη των θρομβώσεων), η συνέπεια είναι να μην δρουν τα φάρμακα.

Ιδιαίτερη προσοχή χρήζει η χρήση της καφεΐνης σε συνδυασμό με ορισμένα φάρμακα. Η καφεΐνη που περιέχεται στον καφέ, το τσάι, σε ορισμένα αναψυκτικά, στη σοκολάτα σε συνδυασμό με φάρμακα που διεγείρουν το ΚΝΣ μπορεί να προκαλέσουν ανεπιθύμητες ενέργειες όπως ταχυκαρδία, αρρυθμία, αϋπνία, αίσθημα παλμών. Τέλος η καφεΐνη ενισχύει την ερεθιστική δράση στο γαστρικό βλεννογόνο ορισμένων φαρμάκων όπως της ασπιρίνης και της κορτιζόνης.

Τέλος η ταυτόχρονη λήψη αιθανόλης (οινοπνευματωδών ποτών) με φάρμακα μπορεί να ενισχύσει ή να ελαττώσει την δράση των φαρμάκων, να ενισχύσει την τοξικότητα ή να εμφανίσει ανεπιθύμητες ενέργειες, όπως

κεφαλαλγία, εφίδρωση, υπνηλία, βαριά υπατική βλάβη. Για τους παραπάνω λόγους συνίσταται η διακοπή της κατανάλωσης αλκοόλ και την διάρκεια της χορήγησης φαρμάκων επεκτείνοντας την διακοπή αυτή και για 2-3 μέρες μετά την διακοπή των φαρμάκων.

2.3 Αλληλεπίδραση φαρμάκου και ατομικοί παράγοντες

Πέρα από τις προηγούμενες κατηγορίες, σημαντικό ρόλο στις αλληλεπιδράσεις των φαρμάκων διαδραματίζουν και κάποιοι ατομικοί παράγοντες του ασθενούς. Τέτοιοι είναι η ηλικία, το φύλο, το βάρος, η μυϊκή μάζα, η κατάσταση της ηπατικής και νεφρικής λειτουργίας, η διάρκεια λήψης φαρμάκου, το κάπνισμα, η εγκυμοσύνη, η αντισύλληψη, η νηστεία, ανορεξία, ψυχικές - διατροφικές διαταραχές, η κοινωνικοοικονομική κατάσταση, η φυλή, η πολιτισμική ταυτότητα καθώς και οι προτιμήσεις στον τρόπο μαγειρέματος.

Η πολυνοσηρότητα και η υπερχορήγηση φαρμάκων σε ηλικιωμένα άτομα σε συνδυασμό με τις φυσιολογικές μεταβολές (συνήθως εξασθένιση) του οργανισμού λόγω γήρατος προκαλούν συνήθως σοβαρά προβλήματα τα οποία οφείλονται αποκλειστικά στα συνταγογραφούμενα και μη φάρμακα που λαμβάνουν. Συνήθη τέτοια προβλήματα είναι η υπέρταση, η καρδιακή ανεπάρκεια, ο διαβήτης και η κατάθλιψη.

Το κάπνισμα ενισχύει τη δραστικότητα των ηπατικών ενζύμων, με αποτέλεσμα την αύξηση του μεταβολισμού πολλών φαρμάκων. Μειώνει την αποτελεσματικότητα πολλών φαρμάκων όπως κάποιων αντικαταθλιπτικών και οιστρογόνων. Αυξάνει συνήθως τις ανεπιθύμητες ενέργειες από το καρδιαγγειακό σύστημα, στις γυναίκες που λαμβάνουν αντισυλληπτικά χάπια. Επιπλέον στους καπνιστές απαιτείται συνήθως χορήγηση μεγαλύτερων δόσεων αναλγητικών ή ινσουλίνης στην περίπτωση των διαβητικών καπνιστών.

Στους παχύσαρκους και τους ηλικιωμένους η αναλογία του λιπώδους ιστού σε σχέση με τη μυϊκή μάζα είναι μεγαλύτερη, με αποτέλεσμα την αύξηση της κατανομής των λιποδιαλυτών φαρμάκων στους ιστούς. Η υπερβολική

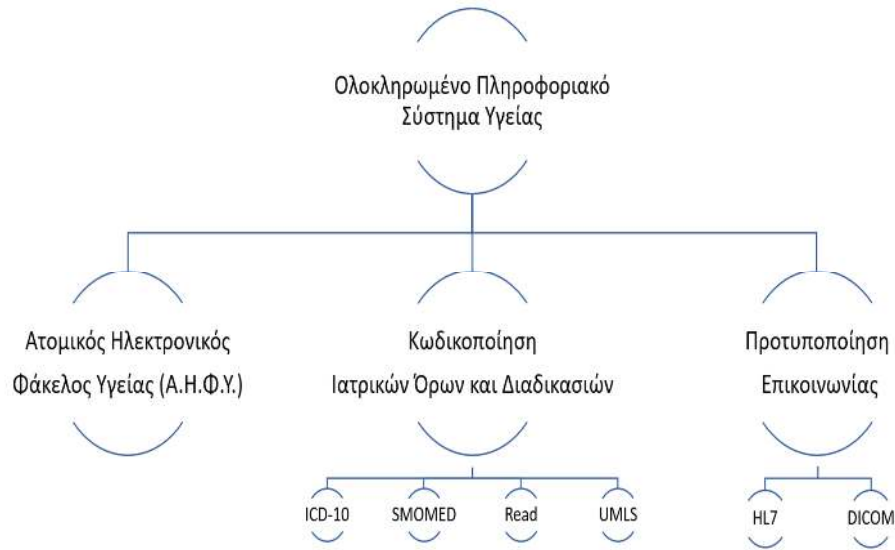
συγκέντρωση του φαρμάκου στο λιπώδη ιστό μπορεί να οδηγήσει σε επιμήκυνση της κάθαρσης και αύξηση της τοξικής δράσης του φαρμάκου [3].

Τέλος ακόμα και ο τρόπος μαγειρέματος μπορεί να επηρεάσει την φυσιολογική λειτουργία κάποιων φαρμάκων, όπως το ψητό κρέας στα κάρβουνα περιέχει πολυκυκλικούς υδρογονάνθρακες που παρεμβαίνουν στην ηπατική λειτουργία αυξάνοντας το μεταβολισμό ενός μεγάλου αριθμού φαρμάκων [4].

3. Ιατροφαρμακευτικά δεδομένα – Μεγάλα Δεδομένα

Στην εποχή που η τεχνολογία έχει πραγματοποιήσει τεράστια βήματα και έχει εισχωρήσει στον τομέα της Υγείας, τα ιατροφαρμακευτικά δεδομένα αυξάνονται με ταχείς ρυθμούς. Λόγω της ανάπτυξης της τεχνολογίας και της πληροφορικής είναι πλέον μαζικότερη η συλλογή και η αξιοποίηση ιατροφαρμακευτικών πληροφοριών από μεγαλύτερο φάσμα πληθυσμών, ασθενειών και θεραπειών τους. Στα πλαίσια υλοποίησης ενός ευρύτερου συστήματος διαχείρισης υγείας παρουσιάζονται πολλές απαιτήσεις σχετικές με το περιεχόμενο, την ομοιογένεια και τη διάδοση της ιατρικής πληροφορίας.

Το αντικείμενο της παρούσας εργασίας αποτελεί μέρος ενός ολοκληρωμένου πληροφοριακού συστήματος υγείας (Εικόνα 2), πιο συγκεκριμένα αφορά στο στάδιο της ηλεκτρονικής συνταγογράφησης κι επομένως απαιτείται να ληφθεί σοβαρά υπόψιν, τόσο στον σχεδιασμό όσο και στην υλοποίηση της εφαρμογής, η προτυποποίηση των ιατροφαρμακευτικών δεδομένων και ιατρικών διαδικασιών. Για το σχεδιασμό και την υλοποίηση της εφαρμογής μας είναι απαραίτητο να ληφθούν υπόψιν τα πρότυπα μετάδοσης των ιατροφαρμακευτικών δεδομένων τόσο για την επιλογή της NoSQL βάσης δεδομένων που θα χρησιμοποιηθεί όσο για τις τεχνικές διαχείρισης των ιατροφαρμακευτικών δεδομένων που με την πάροδο του χρόνου μετατρέπονται σε μεγάλα δεδομένα.



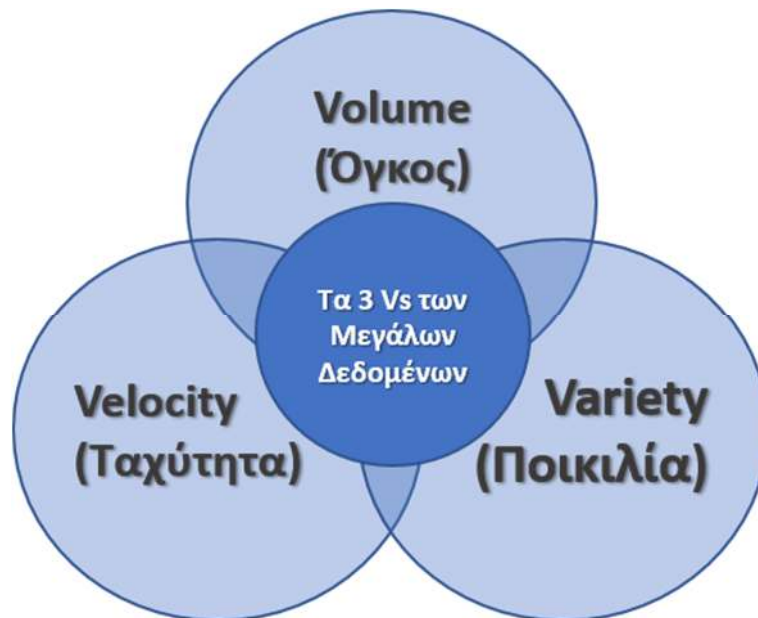
Εικόνα 2 Οι τρεις βασικές απαιτήσεις για την υλοποίηση ενός ολοκληρωμένου πληροφοριακού συστήματος

Πηγή: Ιατρική Πληροφορική, Ψηφιακή Επεξεργασία Βιοσημάτων, Μάρκος Γ. Τσίπουρας Αλέξανδρος Τ. Τζάλλας Ευάγγελος Χ. Καρβούνης Νικόλαος Γιαννακάς σελ 34

3.1 Μεγάλα δεδομένα

Ο όρος «Μεγάλα δεδομένα (Big data)» πρωτοεμφανίστηκε το 1997 από τους επιστήμονες της NASA, για να δηλώσουν την αδυναμία τους να διαχειριστούν, να αναπαραστήσουν και να αποθηκεύσουν στην κύρια μνήμη και σε σκληρό δίσκο τα σύνολα δεδομένων που κατείχαν, καθώς ήταν πολύ μεγάλος ο όγκος τους[5]. Το 2001, ο Doug Laney, τότε αναλυτής στη συμβουλευτική εταιρεία Meta Group Inc., διεύρυνε την έννοια των μεγάλων δεδομένων, εισάγοντας τα χαρακτηριστικά της ποικιλομορφίας και της ταχύτητας εκτός από το χαρακτηριστικό του όγκου των δεδομένων. Από τους σημαντικότερους ορισμούς για τα Μεγάλα δεδομένα είναι αυτός της Gartner (η κορυφαία ερευνητική και συμβουλευτική εταιρεία), το 2012: «Τα μεγάλα δεδομένα είναι στοιχεία υψηλής αξίας, υψηλής ταχύτητας και / ή μεγάλης ποικιλίας που απαιτούν οικονομικά αποδοτικές και καινοτόμες μορφές επεξεργασίας πληροφοριών που επιτρέπουν την καλύτερη γνώση, τη λήψη αποφάσεων και την αυτοματοποίηση της διαδικασίας» [6]. Με αυτόν τον ορισμό

η Gartner πρότεινε ένα μοντέλο των τριών Vs (Volume, Velocity, Variety) δηλαδή τον όγκο, την ταχύτητα και την ποικιλομορφία. (Εικόνα 2)



Εικόνα 3 Μοντέλο 3 Vs χαρακτηριστικών μεγάλων δεδομένων

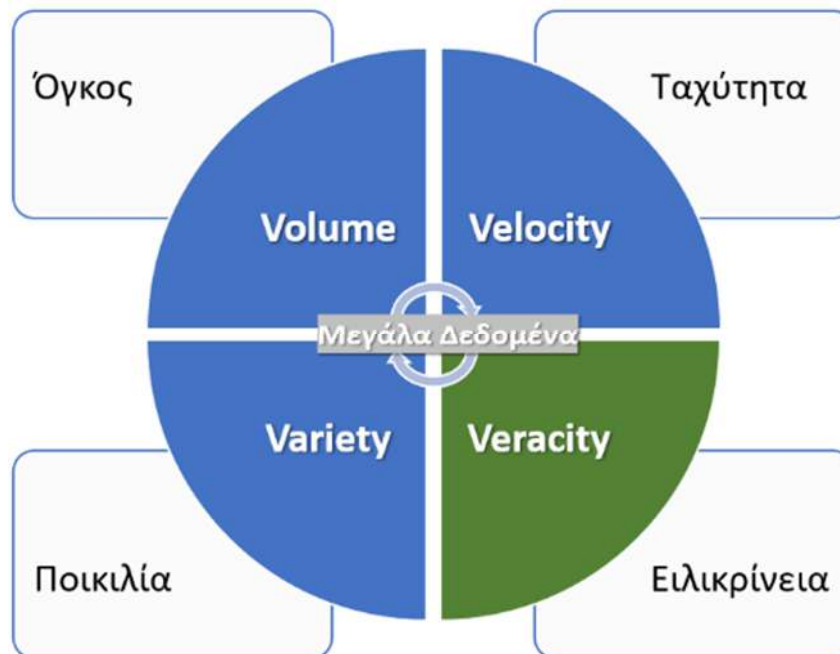
Με τον όρο «Όγκος (Volume)» αναφερόμαστε στον όγκο των δεδομένων που καλούμαστε να διαχειριστούμε. Υπολογίζεται ότι σήμερα με το συγκεκριμένο όρο αναφερόμαστε συνήθως σε όγκους δεδομένων που κυμαίνονται από μερικά terabytes έως δεκάδες ή και εκατοντάδες petabytes (1.024 terabytes) ή exabytes (1.024 petabytes) ή zetabytes(1.024 exabytes). Δεδομένα που παράγονται όχι μόνο από οργανισμούς , αλλά και από τους απλούς ανθρώπους, λόγω της ανάπτυξης των ψηφιακών μέσων, του διαδικτύου, των κοινωνικών δικτύων καθώς επίσης παράγονται και από διάφορους τομείς όπως της Υγείας λόγω των νέων μηχανημάτων, των νέων αισθητήρων, των νέων τεχνολογιών και των νέων πληροφοριακών συστημάτων υγείας.

Με τον όρο «Ταχύτητα (Velocity)» αναφερόμαστε στον ταχύτερο ρυθμό εισαγωγής νέων δεδομένων και ανανέωσης των ήδη υπαρχόντων. Πολλοί τύποι δεδομένων έχουν περιορισμένο χρόνο αποθήκευσης, όπου η αξία τους μπορεί να διαβρωθεί με το χρόνο - σε ορισμένες περιπτώσεις μάλιστα πολύ

γρήγορα. Σε πρώτο στάδιο βρίσκεται η αποθήκευση τους, η διαχείριση τους, η ανάλυση τους και σε δεύτερο στάδιο η ανάλυσή τους και η εξαγωγή χρήσιμων και αξιοποιήσιμων πληροφοριών από τα δεδομένα. Πρόκληση αποτελεί και η εκτέλεση όλων αυτών των διαδικασιών σε πραγματικό χρόνο. Η γρήγορη εκτέλεση όλων των παραπάνω σταδίων μπορεί να επιταχύνει την διαδικασία λήψης αποφάσεων ώστε να συμβαδίσει με τις εκάστοτε υπάρχουσες ανάγκες.

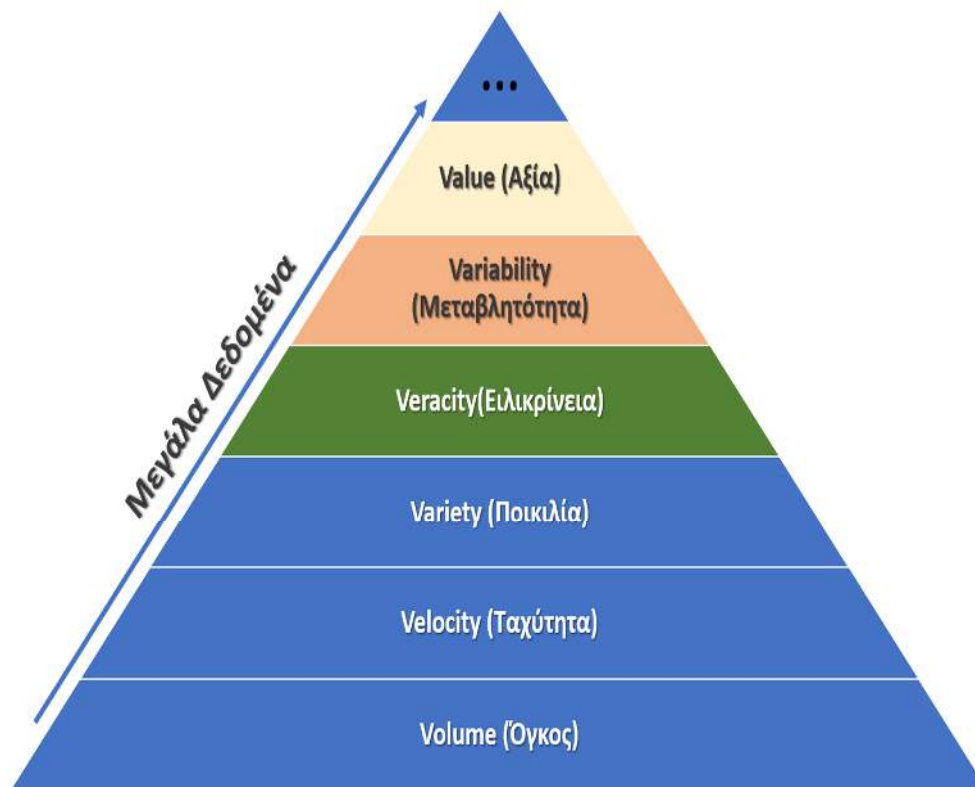
Με τον όρο «Ποικιλία ή Ποικιλομορφία (Variety) αναφερόμαστε στο μεγάλο εύρος διαφορετικών τύπων δεδομένων που απαιτείται να αποθηκεύσουμε, να διαχειριστούμε και να αναλύσουμε. Αυτό περιλαμβάνει διαφορετικές μορφές δεδομένων, σημασιολογία δεδομένων και τύπους δομών δεδομένων. Τα δεδομένα μπορεί να είναι δομημένα (structured) όπως τα σχεσιακά δεδομένα, ημιδομημένα (semi structured) όπως xml, json, αδόμητα (unstructured) όπως κείμενα, φωτογραφίες, βίντεο καθώς και συνδυασμός όλων των προηγούμενων.

Με το πέρασμα των χρόνων στο μοντέλο των 3 Vs προστέθηκαν και νέα χαρακτηριστικά, όπως αυτό της Ειλικρίνειας – Εγκυρότητας (Veracity) (Εικόνα 3), της Αξίας (Value) και της Μεταβλητότητας (Variability)(Εικόνα 4).



Εικόνα 4 Μοντέλο 4 Vs χαρακτηριστικών μεγάλων δεδομένων

Με τον όρο «Ειλικρίνεια - Εγκυρότητα (Veracity)» αναφερόμαστε στην ακρίβεια και την αξιοπιστία των δεδομένων. Όταν πρόκειται για την ακρίβεια των μεγάλων δεδομένων, δεν είναι μόνο η ποιότητα των δεδομένων, αλλά πόσο αξιόπιστη είναι η πηγή δεδομένων, ο τύπος και η επεξεργασία της. Η αφαίρεση στοιχείων όπως προκατάληψη, ανωμαλίες ή ασυνέπειες και αλληλοεπικάλυψη είναι μόνο μερικές από τις πτυχές που επηρεάζουν τη βελτίωση της ακρίβειας των μεγάλων δεδομένων. Η ερμηνεία μεγάλων δεδομένων με τον σωστό τρόπο διασφαλίζει ότι τα αποτελέσματα είναι συναφή και μπορούν να ασκηθούν.



Εικόνα 5 Μοντέλο 6 Vs χαρακτηριστικών μεγάλων δεδομένων

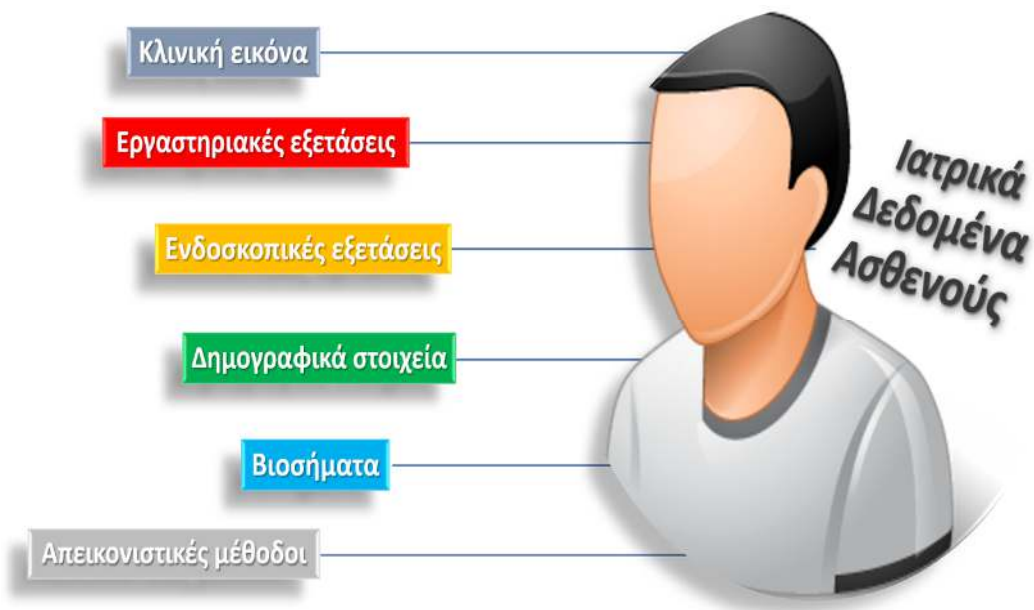
Στο μοντέλο των 6 Vs ένα χαρακτηριστικό είναι η «Μεταβλητότητα (Variability)» των δεδομένων και των μορφών – τύπων δεδομένων. Η μεταβλητότητα μπορεί επίσης να αναφέρεται στην ασυνεπή ταχύτητα με την οποία φορτώνονται μεγάλα δεδομένα στη βάση δεδομένων.

Σημαντικό χαρακτηριστικό είναι η Αξία (Value) των δεδομένων. Αναφερόμαστε στην αξία των πληροφοριών που εξάγονται από την ανάλυση των δεδομένων. Είναι ο τρόπος με τον οποίο χρησιμοποιούνται δεδομένα που επιτρέπουν να αναγνωριστεί πλήρως η πραγματική τους αξία και οι δυνατότητες βελτίωσης των δυνατοτήτων λήψης αποφάσεων. ίσως είναι πλέον το σημαντικότερο χαρακτηριστικό. Είναι σημαντική η ανάλυση της αξίας των εξαγόμενων πληροφοριών σε σχέση με το κόστος της αποθήκευσης, της επεξεργασίας και της ανάλυσης των μεγάλων δεδομένων.

Τέλος στο αρχικό μοντέλο προστίθενται και άλλα χαρακτηριστικά, όπως η Ισχύς (Validity), η Οπτικοποίηση (Visualization), η Ευπάθεια (Vulnerability), η Μεταβλητότητα (Voyability) και όσο περνάει ο χρόνος θα εμφανίζονται και περισσότερα χαρακτηριστικά, πέρα από τα κύρια.

3.2 Ιατρικά δεδομένα

Με τον όρο «Ιατρικά δεδομένα» αναφερόμαστε στα δεδομένα που αφορούν κάποιον ασθενή, τα οποία μπορούν να συλλεχθούν από διαφορετικές πηγές και τα οποία έχουν μια μεγάλη ποικιλομορφία (Εικόνα 6).



Εικόνα 6 Βασικές πηγές από τις οποίες προέρχονται τα ιατρικά δεδομένα ενός ασθενούς

Βασικές κατηγορίες από τις οποίες μπορούν να συλλεχθούν τα ιατρικά δεδομένα είναι:

1. Κλινική εικόνα: Κατά την επίσκεψη του ασθενούς στον ιατρό, καταχωρούνται τα στοιχεία - αποτελέσματα της εξέτασης καθώς και το ιστορικό των συμπτωμάτων του ασθενούς.
2. Εργαστηριακές εξετάσεις: Τα αποτελέσματα από τις εργαστηριακές εξετάσεις (πχ αιματολογικές) αποτελούν μέρος της κλινικής εικόνας του ασθενούς.
3. Ενδοσκοπικές εξετάσεις: Στις περιπτώσεις των ενδοσκοπικών εξετάσεων (πχ γαστροσκόπηση και η κολonosκόπηση) απαιτείται η καταχώρηση των αποτελεσμάτων σε μορφή βίντεο.
4. Δημογραφικά στοιχεία: Σημαντικό ρόλο παίζουν στοιχεία που αφορούν τον ασθενή, όπως ο τρόπος ζωής του (πχ καπνιστής, αλκοολικός), η κληρονομικότητα και όσοι άλλοι παράγοντες μπορούν να επιδράν στην νοσηρότητα του εκάστοτε ασθενή.
5. Βιολογικά σήματα: Αποτελούν στοιχεία που συλλέγονται ως μια ακολουθία τιμών στη διάρκεια του χρόνου (πχ καρδιογράφημα)
6. Απεικονιστικές μέθοδοι: Τα στοιχεία αυτά είναι υπό την μορφή εικόνας και συλλέγονται από τις απεικονιστικές μορφές εξετάσεων όπως είναι η ακτινογραφία, ο υπέρηχος, η μαγνητική τομογραφία και πυρηνικές μέθοδοι απεικόνισης.

Τα παραπάνω στοιχεία, καθώς και η φαρμακευτική αγωγή που δίνεται στον ασθενή, καταχωρούνται στον Ατομικό Ηλεκτρονικό Φάκελο Υγείας (Α.Η.Φ.Υ) του ασθενούς. Για την αξιοποίηση των δεδομένων αυτών από πολλούς διαφορετικούς ειδικούς (πχ ιατρούς, φαρμακοποιούς) έχουν θεσπιστεί πρότυπα για το περιεχόμενο των Α.Η.Φ.Υ αλλά και για τους ιατρικούς όρους και τις ιατρικές διαδικασίες ώστε να γίνονται αντιληπτά από όλους τους ειδικούς. Τα πρότυπα που έχουν καθιερωθεί και χρησιμοποιούνται κατά την καταχώρηση των ιατρικών στοιχείων είναι η Διεθνής Ταξινόμηση Νοσημάτων (ICD-10), η Συστηματοποιημένη ονοματολογία της Ιατρικής (SNOMED), οι Κώδικες Read και το Σύστημα Ενοποιημένης Ιατρικής Γλώσσας (Unified Medical Language System – UMLS).

Πέρα όμως από τα παραπάνω πρότυπα που απευθύνονται κυρίως στους ειδικούς της υγείας, υπάρχουν και τα πρότυπα μετάδοσης ιατρικών δεδομένων τα οποία στοχεύουν στην ομοιογένεια και την ασφάλεια των μεταδιδόμενων ιατρικών δεδομένων και τα οποία αφορούν κυρίως τους ειδικούς για την δημιουργία των πληροφοριακών συστημάτων υγείας. Τα δύο πιο ευρέως χρησιμοποιούμενα πρότυπα είναι το HL7 (Health Level Seven) πρότυπο μετάδοσης ιατρικής πληροφορίας και το DICOM (Digital Imaging and Communications in Medicine) πρότυπο μετάδοσης ιατρικής εικόνας. Για τις ανάγκες της παρούσας εργασίας θα αναλύσουμε παρακάτω το πρότυπο HL7 και κυρίως την μορφή και τον τύπο του και όχι τόσο στο περιεχόμενο του.

3.2.1 Πρότυπο HL7

Το πρότυπο HL7 (Health Level Seven) είναι ένα πρωτόκολλο μεταφοράς ιατρικών δεδομένων σε τυποποιημένα μηνύματα που υλοποιείται στο 7^ο επίπεδο (επίπεδο εφαρμογής) του μοντέλου διασύνδεσης ανοιχτών συστημάτων OSI (Open System Interconnection model). Αποσκοπεί στην εύκολη μεταφορά ιατρικών δεδομένων ανεξαρτήτως του πληροφοριακού και λειτουργικού συστήματος. Το πρότυπο HL7 έχει πιστοποιηθεί από το Εθνικό Ινστιτούτο Τυποποίησης της Αμερικής (American National Standards Institute – ANSI) και από τον οργανισμό ISO. Από το 2004 δημιουργήθηκε η 3^η έκδοση του προτύπου η οποία είναι βασισμένη στο πρότυπο XML (Εικόνα 7).

```
<?xml version="1.0" encoding="UTF-8" ?>
- <PRPA_IN403001UV01 xmlns="urn:hl7-org:v3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:hl7-org:v3 PRPA_IN403001UV01.xsd">
  <id root="1.1.2.3.4.6" extension="5929" assigningAuthorityName="Lifeware Inc." />
  <creationTime value="20050303180027" />
  <versionCode code="3981" />
  <interactionId root="1.1.6.7.8" extension="PRPA_IN403001UV01" assigningAuthorityName="HL7" />
  <!-- profileId root="1.1.1.1" -->
  <processingCode code="D" />
  <processingModeCode code="T" />
  <acceptAckCode code="AA" />
  <receiver typeCode="RCV" />
  <device classCode="DEV" determinerCode="INSTANCE">
    <id root="1.4.7.8.3" />
  </device>
  </receiver>
  <sender typeCode="SND">
    <device classCode="DEV" determinerCode="INSTANCE">
      <id root="1.45.6.7.98" />
    </device>
  </sender>
  <controlActProcess classCode="CACT" moodCode="EVN">
    <subject typeCode="SUBJ" contextConductionInd="False">
      <encounterEvent classCode="ENC" moodCode="EVN">
        <id root="1.56.3.4.7.5" extension="122345" assigningAuthorityName="Maple Hospital Emergency" />
        <code code="EMER" codeSystem="2.16.840.1.113883.5.4" />
        <statusCode code="active" />
      </encounterEvent>
      <subject contextControlCode="OP">
        <patient classCode="PAT">
          <id root="1.56.3.4.7.9" extension="55321" assigningAuthorityName="Maple Hospital Patients" />
          <patientPerson classCode="PSN" determinerCode="INSTANCE">
            <name>
              <given>Rob</given>
              <given>P</given>
              <family>Young</family>
            </name>
            <administrativeGenderCode code="M" codeSystem="2.16.840.1.113883.5.1" />
            <birthTime value="19800309" />
          </patientPerson>
        </patient>
      </subject>
    </encounterEvent>
  </subject>
</controlActProcess>
</PRPA_IN403001UV01>
```

Εικόνα 7 Πρότυπο HL7 v3

Πηγή: Oracle [7]

Τα μοντέλα HL7 v3 χωρίζονται σε 3 κύριους τύπους - περιτυλίγματα, ωφέλιμα φορτία και τύπους κοινών στοιχείων (CMETs). Η HL7 έχει δημοσιεύσει πρόσφατα το πρότυπο HL7 FHIR-R4 ως πρότυπο για δοκιμαστική χρήση [8]. Το FHIR περιγράφεται ως "RESTful" με τυπικούς τύπους MIME (δηλαδή με πρότυπα που υποδεικνύουν τη φύση και τη μορφή ενός εγγράφου, αρχείου ή συλλογής bytes) τους XML, JSON ή RDF.

3.2.2 XML

Η γλώσσα XML (eXtensible Markup Language) αποτελεί σήμερα ένα ευρέως διαδεδομένο πρότυπο για την περιγραφή και ανταλλαγή δεδομένων. Η XML έχει αναπτυχθεί και συντηρείται από το W3C (World Wide Web Consortium), και αποτελεί υποσύνολο ενός προτύπου κατά ISO, της γλώσσας SGML (Standard Generalized Markup Language). Σχεδιάστηκε για να υποστηρίξει την ανταλλαγή δεδομένων μέσω του Παγκόσμιου Ιστού. Δίνει έμφαση στο περιεχόμενο των δεδομένων μιας ιστοσελίδας, αγνοώντας τη μορφοποίησή τους. Η γλώσσα XML βασίζεται στην αναπαράσταση των δεδομένων μέσω αμιγούς κειμένου. Η βασική μονάδα του συντακτικού της XML είναι το στοιχείο (element). Το στοιχείο είναι ένα τμήμα κειμένου που περιβάλλεται από ταιριαστές ετικέτες (tags). Κάθε κείμενο XML έχει ένα στοιχείο, μέσα στο οποίο φωλιάζονται όλα τα υπόλοιπα. Το στοιχείο αυτό είναι μοναδικό και καλείται ρίζα του κειμένου. Η γλώσσα XML επιτρέπει την ανάθεση γνωρισμάτων (attributes) στα στοιχεία. Για τη σωστή δόμηση των δεδομένων μιας εφαρμογής σε ένα κείμενο XML, πρέπει να τηρηθούν οι περιορισμοί που αφορούν στα δεδομένα. Αυτό επιτυγχάνεται με τη χρήση ενός βοηθητικού κειμένου, το οποίο καθορίζει τις έγκυρες ετικέτες, το φώλιασμα των στοιχείων και τα γνωρίσματα αυτών, τους τύπους των δεδομένων, κ.α. Το βοηθητικό αυτό κείμενο δομείται σύμφωνα με τις προδιαγραφές που ορίζουν δύο πρότυπα: (α) το Document Type Definition ή DTD, και (β) το XML Schema Definition ή XSD. Η γλώσσα XML συνοδεύεται από μια μεγάλη και διαρκώς αναπτυσσόμενη οικογένεια εργαλείων (π.χ., XML Schemas, XSL, XSLT, XLink, RDF, DOM, XPath, XQuery, κ.ά.), που διευκολύνουν μια σειρά λειτουργιών, όπως τη μοντελοποίηση δεδομένων στη γλώσσα XML βάσει δομημένων σχημάτων, την

υποβολή ερωτήσεων (σε εκφραστικές γλώσσες) και ανάκτηση δεδομένων από μεγάλα κείμενα XML, την ευρετηριοποίηση κειμένων XML, τη μορφοποίηση και οπτικοποίησή τους, τον εμπλουτισμό της σημασιολογίας τους.

3.2.3 JSON

Το JSON (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολα αναγνωρίσιμο και διαχειρίσιμο και από τους ανθρώπους και από τις μηχανές – προγράμματα. Βασίζεται σε υποσύνολο της JavaScript, αλλά είναι πλήρως ανεξάρτητο της γλώσσας προγραμματισμού. Οι αλγόριθμοι ανάλυσης (parsing) είναι απλοί, γρήγοροι και συνεπείς λόγω του απλού συντακτικού. Η JSON συντάσσεται ως εξής: τα δεδομένα κατατάσσονται σε ζευγάρια ονόματος||τιμής, χωρίζονται με κόμμα και περικλείονται από άγκιστρα, ενώ οι πίνακες από αγκύλες. (Εικόνα 8)

```
{ "name": {  
  "id": "file",  
  "value": "file",  
  "popup": {  
    "items": [  
      { "value": "New", "onclick": "CreateNewDoc()" },  
      { "value": "Open", "onclick": "OpenDoc()" },  
      { "value": "Close", "onclick": "CloseDoc()" }  
    ]  
  }  
}
```

Εικόνα 8 Παράδειγμα JSON

3.3 Φαρμακευτικά δεδομένα

Σε ένα ολοκληρωμένο πληροφοριακό σύστημα υγείας για τη σωστή λειτουργία χρειάζονται διάφορα στοιχεία από φάρμακα, όπως η δραστική ουσία, οι παρενέργειες, οι αλληλεπιδράσεις των φαρμάκων, οι χημικές δομές, οι στόχοι των φαρμάκων, οι αλληλουχίες, τα φαρμακευτικά σκευάσματα, κ.α. Αυτά τα στοιχεία των φαρμάκων χρησιμοποιούνται στα διάφορα στάδια του πληροφοριακού συστήματος, όπως στο ιστορικό της φαρμακευτικής αγωγής για να μπορεί ο εκάστοτε ιατρός να είναι ενημερωμένος κατά την κλινική

εξέταση για παρενέργειες που τυχόν εμφανίζονται στον ασθενή, στο στάδιο της συνταγογράφησης για να προειδοποιείται ο ιατρός για τυχόν αλληλεπιδράσεις φαρμάκων μεταξύ των συνταγογραφούμενων φαρμάκων ή/και με φάρμακα ή τρόφιμα που ήδη λαμβάνει ο ασθενής, στο στάδιο της ανάλυσης από τους ειδικούς που συγκεντρώνουν στοιχεία από το πληροφοριακό σύστημα για περαιτέρω ανάλυση τόσο για ενημερωτικούς όσο και για πειραματικούς σκοπούς, στο στάδιο της πώλησης των σκευασμάτων από τους φαρμακοποιούς.

Τα παραπάνω δεδομένα για τα φάρμακα συνήθως διατίθενται από διάφορους οργανισμούς και εταιρείες παγκοσμίως είτε ελεύθερα είτε σε μορφή εφαρμογής ιστού κατόπιν αμοιβής. Μερικές από τις ευρέως αναγνωρισμένες εταιρείες και οργανισμούς που διαθέτουν βάσεις δεδομένων φαρμάκων είναι η Drugbank, την οποία χρησιμοποιούμε στην παρούσα εργασία και θα την αναλύσουμε παρακάτω, το RxNorm [10] και το DailyMed [11] από την Εθνική Βιβλιοθήκη Ιατρικής των ΗΠΑ (NIH), το DGIdb [12], ο FDA (Εθνική Υπηρεσία Τροφίμων και Φαρμάκων των ΗΠΑ) [13], το KEGG DRUG [14] του Πανεπιστημίου του Κιότο της Ιαπωνίας, το Reactome [15], το SIDER [16], ενώ στην Ελλάδα υπάρχει κατόπιν απόκτησης ειδικής άδειας η βάση δεδομένων του Γαληνού [17].

Drugbank

Η βάση δεδομένων Drugbank είναι ένα πόρος βιοπληροφορικής και χημειοπληροφορικής που περιέχει λεπτομερή δεδομένα για φάρμακα και στόχους φαρμάκων. Είναι ολοκληρωμένη, καθώς περιέχει λεπτομερή δεδομένα, χημικά, φαρμακολογικά και φαρμακευτικά, και περιέχει πάνω από 200 πεδία δεδομένων σε κάθε καρτέλα φαρμάκου (DrugCard) από τα οποία περίπου τα μισά αναφέρονται σε δεδομένα φαρμάκου/χημικών ενώ το άλλο μισό σε στόχους φαρμάκων ή πρωτεϊνών, όπως πληροφορίες αλληλουχίας, δομής και οδών. Είναι αρκετά αξιόπιστη, καθώς σχεδόν κάθε στοιχείο αναφέρεται στην πρωτότυπη πηγή και είναι πλήρως εξακριβώσιμο προτού καταχωρηθεί στην βάση.

Η βάση Drugbank ανήκει στους Διοικητές του Πανεπιστημίου της Αλμπέρτα που εδρεύει στον Καναδά. Εκδόθηκε για πρώτη φορά το 2006 περιείχε πληροφορίες για 954 φάρμακα εγκεκριμένα από την Υπηρεσία Τροφίμων και Φαρμάκων (FDA) και 133 στόχους φαρμάκων. Η τελευταία έκδοση της (έκδοση 5.1.2 με ημερομηνία έκδοσης την 20η Δεκεμβρίου 2018) περιέχει 12.064 καταχωρήσεις φαρμάκων και 5.142 στόχους φαρμάκων/ ένζυμα/ μεταφορείς / φορείς που συνδέονται με τις καταχωρήσεις φαρμάκων.

Τέλος διατίθεται ελεύθερα για μη εμπορικούς σκοπούς σε μορφή xml και έχει μέγεθος 758 MB.

4. Εργαλεία και τεχνικές διαχείρισης μεγάλων δεδομένων

Μελετώντας τα ιατροφαρμακευτικά προϊόντα στο προηγούμενο κεφάλαιο διαπιστώνεται ότι έχουν πολλά χαρακτηριστικά από τα μεγάλα δεδομένα, όπως η ποικιλία, καθώς συλλέγονται από πολλές διαφορετικές πηγές και σε διαφορετικούς τύπους, η ειλικρίνεια, καθώς οι πηγές που συλλέγονται είναι αξιόπιστες, ο όγκος και η ταχύτητα που με την πάροδο των χρόνων και με την αυξανόμενη, υποχρεωτική και καθολική χρήση των πληροφοριακών συστημάτων υγείας αυξάνονται ραγδαία. Ενδεικτικά για την χώρα μας, απαιτείται η εγγραφή στις Τοπικές Μονάδες Υγείας (ΤΟΜΥ) όλων των πολιτών της χώρας μας στους επόμενους μήνες και η χρήση του πληροφοριακού συστήματος υγείας της ΗΔΙΚΑ (Ηλεκτρονική Διακυβέρνηση Κοινωνικής Ασφάλισης) για την πρωτοβάθμια φροντίδα υγείας. Επιπλέον αυτών η μεταβλητότητα είναι ένα χαρακτηριστικό που εμφανίζεται καθώς θα ενημερώνονται συνεχώς για κάθε ασθένεια και τέλος η αξία των δεδομένων καθώς αυτά αφορούν την υγεία των ασθενών και αποσκοπούν στην βελτίωση του συστήματος υγείας τόσο από διοικητική, οργανωτική άποψη όσο και από οικονομική άποψη.

Από την άλλη πλευρά ο ρυθμός ανάπτυξης των δυνατοτήτων των υλικών των ηλεκτρονικών συστημάτων (hardware) είναι μικρότερος από τη ραγδαία αύξηση των δεδομένων, η οποία περιορίζει τη δυνατότητα διαχείρισης τους.

Έτσι δημιουργείται η ανάγκη για οριζόντια κλιμακούμενες λύσεις με ταυτόχρονη χρήση της νεφοϋπολογιστικής (cloud computing). Με την οριζόντια κλιμάκωση (horizontal scalability) επιτυγχάνεται η πρόσθεση νέων εξυπηρετητών (server) σε υπάρχοντα συστήματα με μικρή προσπάθεια και χαμηλό κόστος, η υποστήριξη ετερογενών συστημάτων και η εύκολη εξυπηρέτηση παράλληλων αιτημάτων σε αντίθεση με την κάθετη κλιμάκωση (vertical scalability) που αυξάνονται οι επιδόσεις ενός εξυπηρετητή (server), έχοντας έτσι περιορισμένες δυνατότητες, ευπάθειες σε αστοχίες υλικών και επιρρέπεια σε down times.

Σε συνδυασμό με τα παραπάνω, μελετώντας την προτυποποίηση των ιατροφαρμακευτικών δεδομένων διαπιστώνουμε ότι ο διαμοιρασμός και η εγγραφή τους πραγματοποιείται με πρότυπα που είναι κατά το πλείστον σε ημιδομημένες μορφές και χωρίς καθορισμένα σχήματα δεδομένων (schema-less), γεγονός που μας κατευθύνει στην χρήση NoSQL συστημάτων διαχείρισης βάσεων δεδομένων.

Τέλος στην παρούσα εργασία κάναμε χρήση της βάσης δεδομένων Drugbank, μιας βάσης με 12.064 φάρμακα, στην οποία όμως εμπεριέχονται 745.070 αλληλεπιδράσεις φαρμάκων με άλλα φάρμακα και 2.995 αλληλεπιδράσεις φαρμάκων με τρόφιμα, γεγονός που δυσκολεύει την χρήση του τελεστή συνένωσης (join).

4.1 NoSQL συστήματα διαχείρισης βάσεων δεδομένων

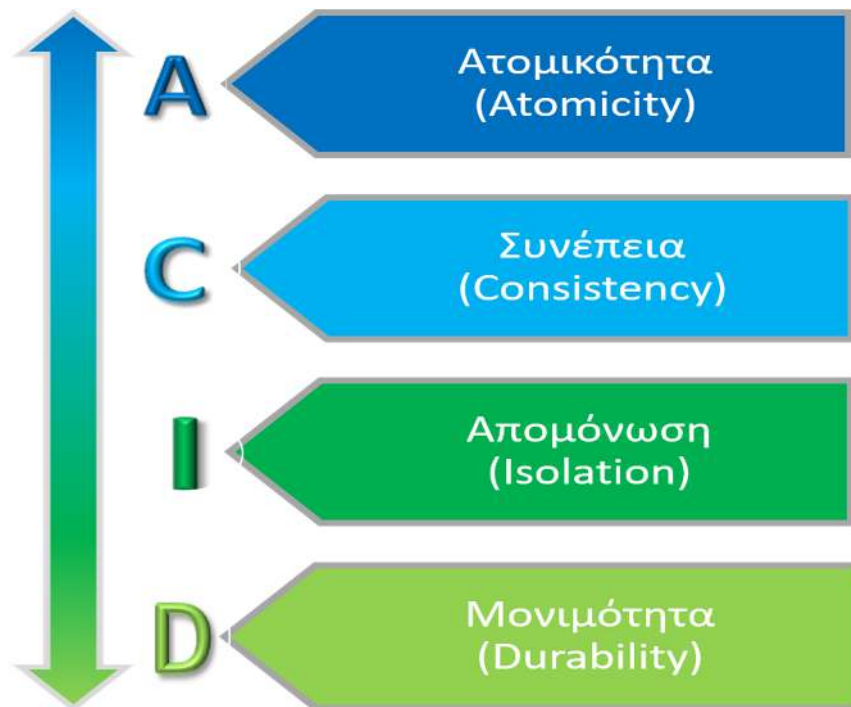
Τα τελευταία χρόνια έχει εδραιωθεί η χρήση των Σχεσιακών Συστημάτων Διαχείρισης Βάσεων Δεδομένων (RDBMS) και η χρήση της Δομημένης Γλώσσας Επερωτήσεων (SQL). Όμως η εκθετική αύξηση του όγκου των δεδομένων σε συνδυασμό με την αυξανόμενη αλληλεπίδραση μεταξύ των δεδομένων και την πολυπλοκότητα των δεδομένων επιταχύνονται από την χρήση κατανεμημένων συστημάτων και από την χρήση του διαδικτύου. Τα παραπάνω οδήγησαν τις σχεσιακές βάσεις δεδομένων στα όριά τους. Το αυστηρά σχεσιακό σχήμα αντιμετωπίζει προβλήματα στις νέες εφαρμογές ιστού που έχουν να διαχειριστούν μεγάλη ποικιλία δεδομένων, όπως κείμενα, φωτογραφίες, βίντεο, πηγαίο κώδικα. Μειονεκτήματα των σχεσιακών βάσεων

δεδομένων είναι ο εγγενής καθορισμός του σχήματος της βάσης και του όγκου των δεδομένων, η κακή κλιμάκωση και η κανονικοποίηση που οδηγεί σε τεμαχισμό των δεδομένων σε πολλούς πίνακες και στην εκτέλεση πολλών χρονοβόρων πράξεων σύνδεσης (join).

Τα χαρακτηριστικά που ισχύουν για την αξιοπιστία των συστημάτων είναι γνωστά με το ακρωνύμιο ACID (Atomicity, Consistency, Isolation, Durability) (Εικόνα 9).

Ατομικότητα (Atomicity): Η ατομικότητα ακολουθεί τον κανόνα όλα ή τίποτα. Δηλαδή όταν κάτι αποτύχει σε μια συναλλαγή, τότε η συναλλαγή αποτυγχάνει ολόκληρη και η βάση επιστρέφει στην αρχική της εικόνα.

Συνέπεια (Consistency): Με την συνέπεια εξασφαλίζεται η συνέπεια των δεδομένων τόσο στην αρχή όσο και στην ολοκλήρωση μιας συναλλαγής.



Εικόνα 9 Ιδιότητες ACID

Απομόνωση (Isolation): Με την απομόνωση, οι λοιπές εργασίες δεν μπορούν να δουν/επεξεργαστούν τα δεδομένα που τροποποιούνται μέχρι να ολοκληρωθεί η συναλλαγή.

Μονιμότητα (Durability): Με την μονιμότητα διασφαλίζεται ότι από την στιγμή που θα ολοκληρωθεί μία συναλλαγή τα δεδομένα δεν μπορούν να αναιρεθούν.

Όμως με την εμφάνιση των κατανεμημένων συστημάτων παρουσιάστηκε ένας προβληματισμός σχετικά με την πραγματοποίηση/τήρηση όλων των χαρακτηριστικών. Λύση στον παραπάνω προβληματισμό δόθηκε με το Θεώρημα CAP, γνωστό επίσης και ως Θεώρημα Brewer, που δημοσιεύτηκε το 2000. Σύμφωνα με αυτό ένα κατανεμημένο σύστημα διαχείρισης βάσης δεδομένων σε μια δεδομένη στιγμή μπορεί να επιτύχει μόνο δύο από τις τρεις ιδιότητες:

- **Consistency** : Συνεκτικότητα δεδομένων
- **Availability** : Διαθεσιμότητα συστήματος
- **Network Partitions**: Ανοχή σε κατανεμημένα δεδομένα

Με βάση το παραπάνω θεώρημα δημιουργούνται τρεις δυνατοί συνδυασμοί ιδιοτήτων (Εικόνα 10):

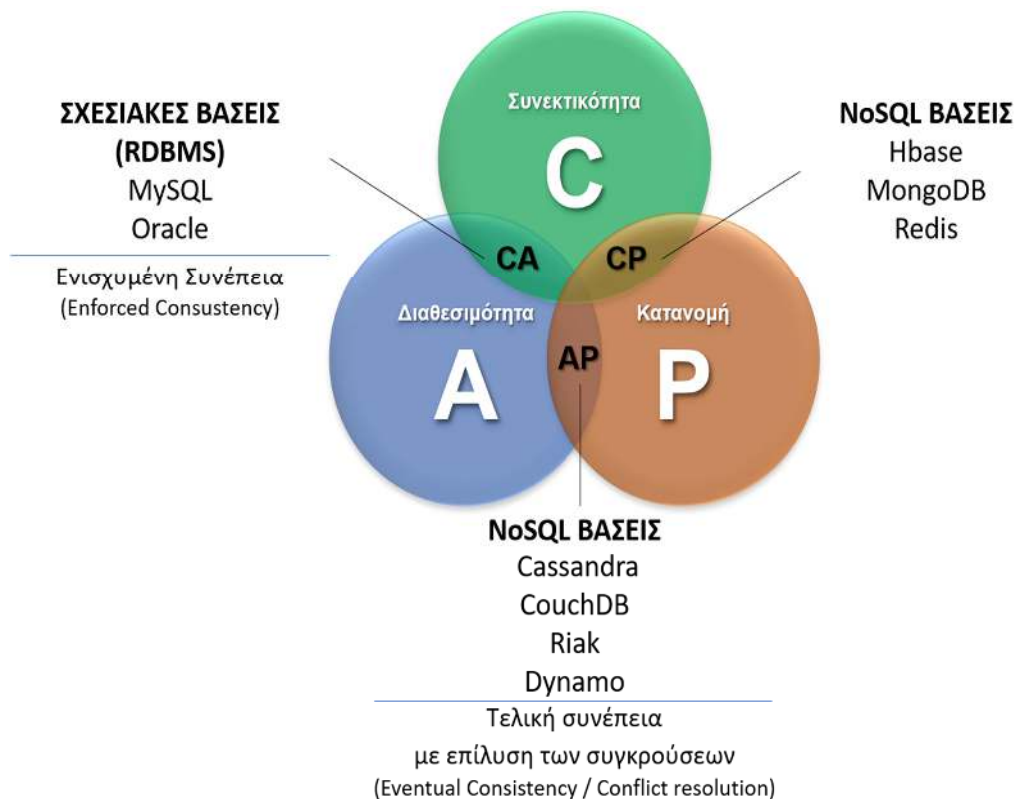
- ❖ **CA**: Εδώ τα δεδομένα έχουν μικρή έως καθόλου κατανομή, αλλά εγγυούνται την διαθεσιμότητα και την συνέπεια.
- ❖ **CP**: Εδώ τα δεδομένα δεν είναι συνεχώς διαθέσιμα, αλλά εγγυούνται την συνέπεια και την αντοχή στην κλιμάκωση.
- ❖ **AP**: Εδώ τα δεδομένα είναι συνεχώς διαθέσιμα και αντέχουν στην κατανομή, χαλαρώνοντας όμως τον όρο της συνέπειας.

Το 2009 πρωτοεμφανίστηκε ο όρος των NoSQL (Not only SQL) συστημάτων βάσεων δεδομένων. Είναι ένα είδος μη σχεσιακών (non relational) συστημάτων βάσεων δεδομένων, συνεπώς δεν απαιτείται ο καθορισμός του σχήματος των δεδομένων και δεν υποστηρίζεται εγγενώς η χρήση του τελεστή (join). Τα συστήματα NoSQL δεν αντικαθιστούν τα παραδοσιακά σχεσιακά συστήματα RDBMS (η ονομασία είναι **Not only SQL** και όχι **No SQL**) , αλλά τα συμπληρώνουν, όταν απαιτείται αύξηση των επιδόσεων και ευελιξία δεδομένων.

Πλεονεκτήματα των NoSQL συστημάτων είναι:

- Υποστήριξη του μοντέλου των Μεγάλων Δεδομένων
- Εγγενής υποστήριξη της οριζόντιας κλιμάκωσης (scalability)
- Ευελιξία στη μεταβαλλόμενη μορφή δεδομένων
- Εύκολη διαχειρισιμότητα

Σε σχέση με το θεώρημα CAP τα NoSQL συστήματα για να υποστηρίξουν τα κατανεμημένα δεδομένα επιλέγουν συνήθως να χαλαρώσουν την συνεκτικότητα (συνδυασμός AP και κάποια CP), σε αντίθεση με τα RDBMS συστήματα που αγνοούν τα Partitions για να επιτύχουν τις ACID ιδιότητες (συνδυασμός CA) (Εικόνα 10).



Εικόνα 10 Θεώρημα CAP (Brewer)

Με την χαλάρωση της συνεκτικότητας όμως, για την επίτευξη της αξιοπιστίας των NoSQL συστημάτων που δεν μπορούσαν να τηρήσουν τις ACID ιδιότητες, χρησιμοποιήθηκαν οι ιδιότητες BASE (**B**asically **A**vailable, **S**oft state, **E**ventual consistency).

➤ Basically Available

Το σύστημα είναι διαθέσιμο ακόμα κι αν ένας ή περισσότεροι κόμβοι δεν είναι διαθέσιμοι. Αυτό επιτυγχάνεται με την ύπαρξη αντιγράφων δεδομένων σε πολλούς κόμβους, χωρίς όμως να εγγυούνται ότι τα δεδομένα είναι συνεπή.

➤ Soft state

Η κατάσταση του συστήματος μπορεί να μεταβληθεί με την πάροδο του χρόνου, ακόμα κι αν δεν υπάρχει αλληλεπίδραση με κάποιον χρήστη, όπως είναι στην περίπτωση της επαναφοράς της συνέπειας στο σύστημα. Έτσι το σύστημα βρίσκεται πάντα σε μια «μαλακή» κατάσταση.

➤ Eventual consistency

Για την επίτευξη πολύ υψηλής διαθεσιμότητας στην κατανεμημένη υπολογιστική χρησιμοποιείται αυτό το μοντέλο συνεκτικότητας. Όταν δεν πραγματοποιούνται εγγραφές για μεγάλο χρονικό διάστημα, διαδίδονται όλες οι ενημερώσεις στο σύστημα και ενημερώνονται όλοι οι κόμβοι που περιέχουν αντίγραφα δεδομένων. Όμως με τον τρόπο αυτό μπορεί να εμφανιστούν αντιθέσεις δεδομένων (conflicts). Η ακριβής διαδικασία επίλυσής τους εξαρτάται κάθε φορά από την υλοποίηση του κάθε προϊόντος. Συνήθως όμως χρησιμοποιείται η μέθοδος διατήρησης της νεότερης τιμής που εντοπίζεται στους επιμέρους κόμβους.

Τα NoSQL συστήματα διαχείρισης βάσεων δεδομένων χρησιμοποιούνται σε μεγάλους όγκους δεδομένων και με μεγάλη πολυπλοκότητα. Ανάλογα με το μοντέλο δεδομένων δημιουργήθηκαν και τα αντίστοιχα προϊόντα. Τα μοντέλα δεδομένων που διαχειρίζονται τα NoSQL συστήματα [18] και τα αντίστοιχα προϊόντα είναι:

- Ζεύγη κλειδιού – τιμής (key-value): Τα δεδομένα αποθηκεύονται σε ζεύγη κλειδιού – τιμής. Προτιμάται για δεδομένα απλής φύσης χωρίς πολύπλοκη δομή και υψηλά επίπεδα συσχετίσεων. Μερικά από τα διαθέσιμα προϊόντα είναι τα Redis, Membase, Memcached.
- Οργάνωση σε στήλες (Column oriented) : Τα δεδομένα αποθηκεύονται σε πίνακες όπως στις σχεσιακές βάσεις, οι οποίοι αποθηκεύονται όμως σε στήλες και όχι σε γραμμές. Προτιμάται για πίνακες δεδομένων πολύ μεγάλου μεγέθους και επιτυγχάνεται καλύτερη συμπίεση των δεδομένων καθώς τα δεδομένα που βρίσκονται στην ίδια στήλη είναι του ίδιου τύπου. Μερικά από τα διαθέσιμα προϊόντα είναι τα Cassandra, Hbase, BigTable.
- Οργάνωση σε Γράφους (Graph oriented): Τα δεδομένα οργανώνονται σε κόμβους και στις σχέσεις που τα συνδέουν. Χρησιμοποιείται σε δεδομένα με υψηλές συσχετίσεις που πρέπει να αποτυπωθούν, όπως σε συστήματα κοινωνικής δικτύωσης. Μερικά από τα προϊόντα είναι τα Neo4j, OrientDB, GraphDB.
- Αντικειμενοστραφείς (Object oriented): Αποθηκεύουν τα δεδομένα σε μορφή αντικειμένων. Έχουν αυξημένες επιδόσεις με τις αντικειμενοστραφείς γλώσσες προγραμματισμού. Χρησιμοποιούνται σε εφαρμογές που αποθηκεύουν δεδομένα που αφορούν φυσικά ή ιδεατά αντικείμενα, όπως τα video games. Μερικά από τα προϊόντα είναι τα VelocityDB, HSS Database, Objectivity.
- Οργάνωση σε έγγραφα (Document oriented): Αποτελεί την πιο δημοφιλή κατηγορία, μια κατηγορία που μπορεί να εφαρμοστεί στην ηλεκτρονική συνταγογράφηση. Τα δεδομένα ακολουθούν τη μορφή εγγράφων και αποθηκεύονται σε μορφή JSON ή παραλλαγές αυτού (BSON). Τα δεδομένα δεν είναι σε κανονικοποιημένη μορφή καθώς έχουμε ίδια δεδομένα σε πολλά έγγραφα. Δίνεται όμως η δυνατότητα ευρετηρίασης για οποιοδήποτε χαρακτηριστικό (attribute) του εγγράφου. Μερικά από τα διαθέσιμα προϊόντα είναι τα MongoDB, CouchDB, Terrastore, Elasticsearch.

4.2 MongoDB

Σε ένα σύστημα ηλεκτρονικής συνταγογράφησης, οι συνταγές φαρμάκων ακολουθούν μορφή εγγράφων (Εικόνα 11), οπότε επιτρέπεται η χρήση βάσης δεδομένων οργανωμένη σε έγγραφα (document-oriented).

```
Prescription:
  id: 162635,
  Drug: "Paracetamol/500mg",
  Doctor: {
    Doctor_id: 1048,
    Doctor_name: "Efstathios",
    Doctor_surname: "Avery",
    *****
  },
  Patient: {
    Patient_id: 883457,
    Patient_name: "Konstantinos",
    Patient_surname: "Smith",
    *****
  }
}
```

Εικόνα 11 Μορφή συνταγής φαρμάκων

Η πιο δημοφιλής NoSQL βάση δεδομένων [19] στην κατηγορία αυτή είναι η MongoDB [20]. Είναι βάση δεδομένων ανοιχτού κώδικα, διατίθεται δωρεάν και είναι γραμμένη σε γλώσσα προγραμματισμού C++. Δημιουργήθηκε το 2007 από την εταιρεία 10gen. Το 2009, η 10gen κυκλοφόρησε τη MongoDB ως έργο ανοιχτού κώδικα, ενώ το 2013 η εταιρεία μετονομάστηκε σε MongoDB Inc.

Η MongoDB βάση δεδομένων διαχειρίζεται δεδομένα με δομή οργανωμένη σε έγγραφα. Το έγγραφο είναι δομή δεδομένων που αποτελείται από ζεύγη πεδίων – τιμών. Οι τιμές πεδίων μπορούν να περιλαμβάνουν άλλα έγγραφα, συστοιχίες και πίνακες εγγράφων, γεγονός που μειώνει την ανάγκη για δαπανηρές και χρονοβόρες πράξεις σύνδεσης (join). Σχεδιάστηκε για να καλύπτει αποτελεσματικά τις ανάγκες των εφαρμογών ιστού. Τα δεδομένα αποθηκεύονται σε μορφή BSON (ένα δυαδικό μορφότυπο σειριοποίησης που αντιπροσωπεύει το δυαδικό JSON και χρησιμοποιείται για την αποθήκευση εγγράφων και την πραγματοποίηση κλήσεων απομακρυσμένης διαδικασίας)

και η επικοινωνία με τις εφαρμογές ιστού γίνεται μέσω JSON REST-ful APIs. Στη MongoDB οι συλλογές (collections) είναι σαν πίνακες και τα έγγραφα (documents) σαν εγγραφές (records). Η διαφορά είναι ότι κάθε αρχείο σε μια συλλογή μπορεί να έχει εντελώς διαφορετικά πεδία από τα άλλα έγγραφα. Το μόνο schema που απαιτείται στη MongoDB είναι ένα '_id' πεδίο με μοναδική τιμή (non-array).

Τα βασικά χαρακτηριστικά του συστήματος της MongoDB [21] είναι:

- Υψηλή απόδοση
- Οριζόντια κλιμάκωση με ισοκατανομή δεδομένων στους κόμβους
- Πλούσια γλώσσα ερωτημάτων
- Υψηλή διαθεσιμότητα
- Υποστήριξη για πολλαπλούς μηχανισμούς αποθήκευσης

Η MongoDB επιτυγχάνει την οριζόντια κλιμάκωση μέσω του sharding. Με αυτήν την διαδικασία το σύνολο τεμαχίζεται σε υποσύνολα και διαμοιράζεται σε όλους τους κόμβους του συστήματος.

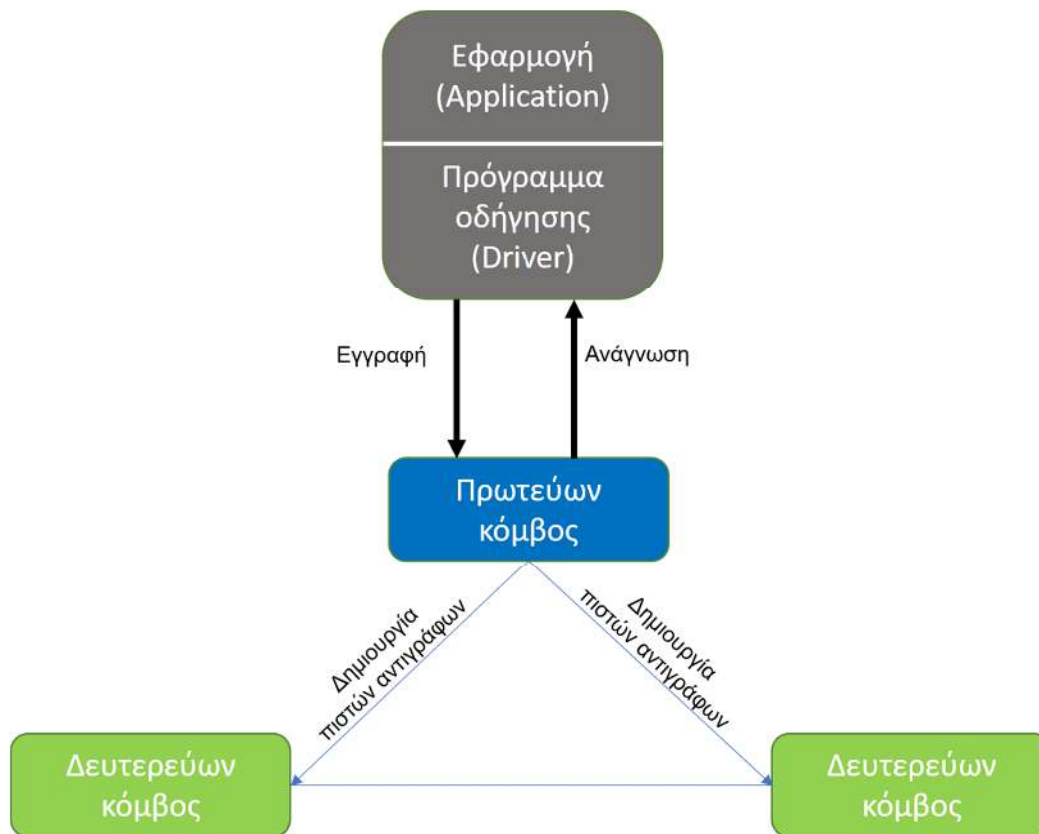
Κύριο χαρακτηριστικό της MongoDB είναι η υψηλή διαθεσιμότητα. Αυτό επιτυγχάνεται με την δημιουργία replica set (σύνολο πιστών αντιγράφων), δίνοντας την δυνατότητα αυτόματης αντιγραφής των δεδομένων σε κάποιους κόμβους και αυτόματης διαχείρισης περιπτώσεων μη διαθεσιμότητας κάποιων κόμβων.

Ένα replica set αποτελείται υποχρεωτικά από δύο ειδών κόμβους (Εικόνα 12):

- Ένα πρωτεύοντα που ανταποκρίνεται σε όλες τις ενέργειες εγγραφής – ανάγνωσης για τα δεδομένα που αποθηκεύει
- Έναν ή περισσότερους δευτερεύοντες κόμβους που αντιγράφουν τα δεδομένα.

Ο πρωτεύων κόμβος ή κύριος κόμβος αντιγράφει όλες τις αλλαγές των δεδομένων του στο ημερολόγιο λειτουργίας του – αρχείο καταγραφής

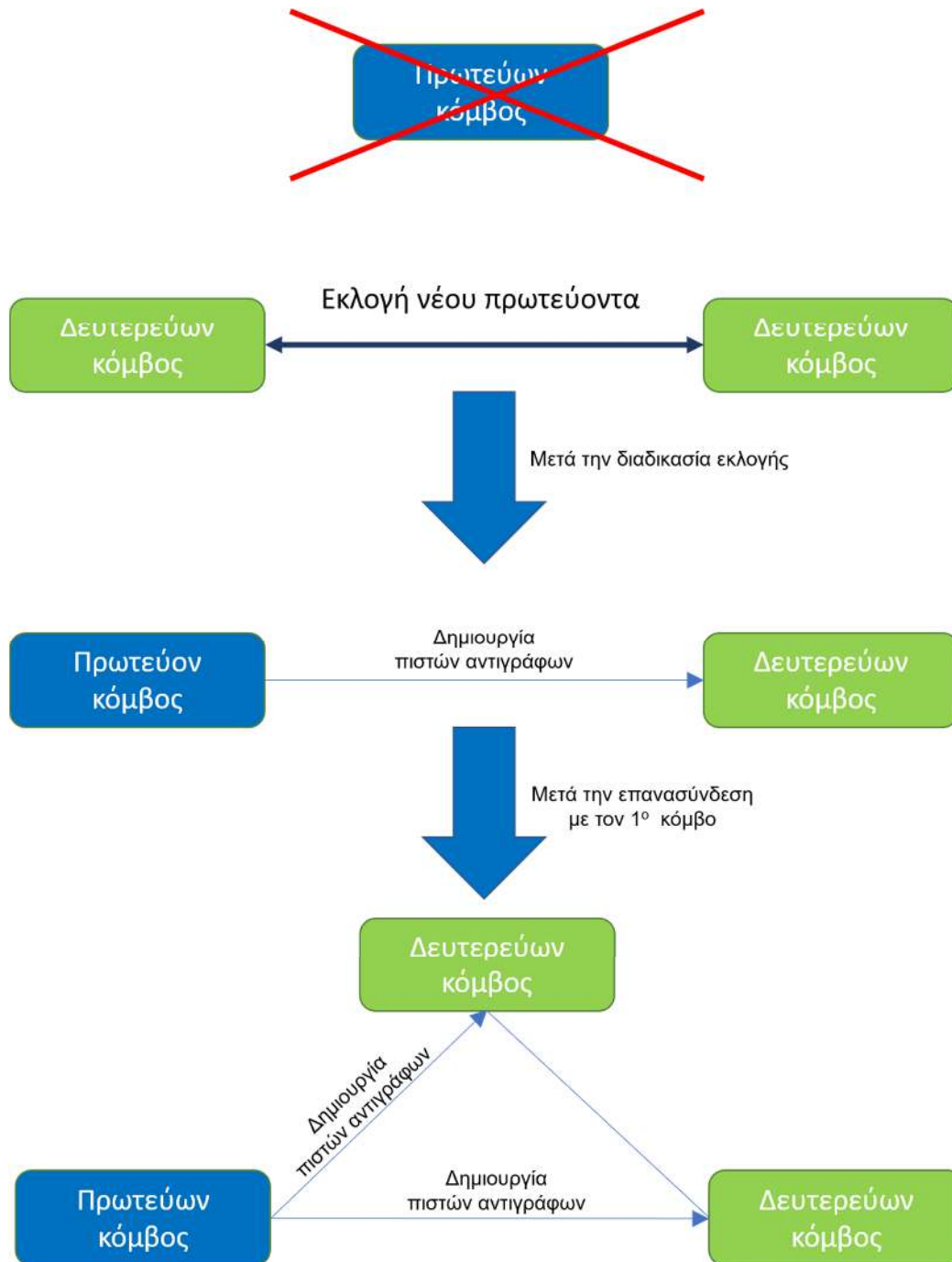
ενεργειών, δηλαδή orlog. Οι δευτερεύοντες κόμβοι αντιγράφουν ασύγχρονα το ημερολόγιο του πρωτεύοντα κόμβου και κάνουν τις πράξεις στα δεδομένα τους ώστε να είναι ίδια με τα δεδομένα του πρωτεύοντα.



Εικόνα 12 MongoDB - Replica Set

Σε περίπτωση που ο πρωτεύοντας κόμβος πάψει να επικοινωνεί με τους δευτερεύοντες για κάποιο χρονικό διάστημα (`electionTimeoutMillis` - από προεπιλογή 10 δευτερόλεπτα) τότε ξεκινάει η διαδικασία εκλογής νέου πρωτεύοντα με ψηφοφορία. Η διαδικασία εκλογής ξεκινάει και εκτελείται είτε από κάποιον προκαθορισμένο δευτερεύοντα είτε από κάποιον κόμβο που παίζει το ρόλο του διαιτητή (Εικόνα 13). Προαιρετικά μπορεί να γίνει η ανάθεση του ρόλου του διαιτητή σε κάποιον κόμβο. Αυτός ο κόμβος δεν αποθηκεύει αντίγραφο του συνόλου των δεδομένων αλλά διασφαλίζει την ακρίβεια των δεδομένων και είναι υπεύθυνος για την εκλογή του νέου πρωτεύοντα. Επίσης

ο διαιτητής είναι μια καλή λύση για την ύπαρξη πλειοψηφίας στην ψηφοφορία σε περίπτωση ζυγού αριθμού δευτερευόντων κόμβων και αποτελεί μια καλή και φθηνή λύση καθώς δεν απαιτεί ειδικό υλικό. Ένας διαιτητής κόμβος θα είναι πάντοτε διαιτητής ενώ ένας πρωτεύων μπορεί να γίνει δευτεύων και το αντίθετο.



Εικόνα 13 Εκλογή πρωτεύοντα κόμβου - Replica set

4.3 Υπολογιστικά Νέφη - Okeanos

Με τον όρο «Υπολογιστικά Νέφη» (Cloud Computing) αναφερόμαστε στην παροχή υπολογιστικών πόρων μέσω του διαδικτύου. Η παροχή των πόρων είναι βασισμένη στη ζήτηση, δηλαδή παρέχονται οι υπολογιστικοί πόροι που απαιτούνται κάθε στιγμή και μπορούν να μεταβληθούν ανάλογα με τις απαιτήσεις, ακόμα και να ακυρωθούν. Στα υπολογιστικά νέφη γίνεται χρήση εικονικού υλικού και ένας φυσικός εξυπηρετητής μπορεί να διαθέσει πολλούς εικονικούς εξυπηρετητές που μοιράζονται τους φυσικούς πόρους. Έτσι επιτυγχάνεται σωστή διαχείριση και μέγιστη χρήση των φυσικών πόρων, κλιμάκωση, ομοιογένεια και επεκτασιμότητα.

Ανάλογα με τους παρεχόμενους πόρους τα υπολογιστικά νέφη διακρίνονται σε τρεις κατηγορίες:

- Παροχή υποδομής ως υπηρεσία - Infrastructure as a Service (IaaS)
Παρέχεται στους χρήστες πρόσβαση σε υπολογιστικούς πόρους όπως διακομιστές, αποθήκευση και δικτύωση. Ο τελικός χρήστης χρησιμοποιεί τις δικές του πλατφόρμες και εφαρμογές. Το όφελος του τελικού χρήστη είναι εξοικονόμηση χρημάτων από την μη αγορά του υλικού και την μη συντήρησή του, η κλιμάκωση ανάλογα με τις ανάγκες του και η μη ύπαρξη αποτυχίας.
- Παροχή πλατφόρμας ως υπηρεσία - Platform as a Service (PaaS)
Παρέχεται μια συλλογή προεγκατεστημένων εργαλείων, επιπλέον των αποθηκευτικών και υπολογιστικών πόρων, με τα οποία μπορούν να αναπτύξουν, να διαχειριστούν, να δοκιμάσουν και να φιλοξενήσουν τις εφαρμογές τους. Ο τελικός χρήστης ασχολείται μόνο με την ανάπτυξη της εφαρμογής, ενώ οι πάροχοι είναι υπεύθυνοι για την ασφάλεια, τα λειτουργικά συστήματα, τις ενημερώσεις και τα αντίγραφα ασφαλείας.
- Παροχή εφαρμογών ως υπηρεσία - Software as a Service (SaaS)
Παρέχεται στους χρήστες λογισμικό και εφαρμογές με συνδρομή, χωρίς να ασχολούνται για την εγκατάστασή τους, τις ενημερώσεις τους και την αναβάθμισή τους. Ο τελικός χρήστης ασχολείται μόνο με την αποθήκευση και ανάλυση των δεδομένων του. Τα δεδομένα

είναι ασφαλή και προσβάσιμα από οποιαδήποτε συσκευή έχει πρόσβαση στο διαδίκτυο.

Για την ολοκλήρωση της παρούσας εργασίας σε υπολογιστικά νέφη χρησιμοποιήθηκε ο Okeanos [22], που είναι μια υπηρεσία IaaS (Infrastructure as a Service) του ΕΔΕΤ για την Ελληνική Έρευνα και Ακαδημαϊκή Κοινότητα. Παρέχεται δωρεάν σε φοιτητές και σε εκπαιδευτικό προσωπικό.

4.4 Apache Hadoop – HDFS

Το Apache Hadoop [23] είναι ένα κατακευματισμένο πλαίσιο επεξεργασίας ανοικτού κώδικα που διαχειρίζεται την επεξεργασία και αποθήκευση δεδομένων για εφαρμογές μεγάλων δεδομένων που εκτελούνται σε συστήματα συστοιχιών (clusters). Βρίσκεται στο επίκεντρο ενός αναπτυσσόμενου οικοσυστήματος τεχνολογιών μεγάλων δεδομένων που χρησιμοποιούνται κυρίως για τη υποστήριξη προηγμένων πρωτοβουλιών ανάλυσης, συμπεριλαμβανομένων προγνωστικών αναλύσεων, εξόρυξης δεδομένων και εφαρμογών μηχανικής μάθησης .

Το Hadoop μπορεί να χειριστεί διάφορες μορφές δομημένων και αδόμητων δεδομένων, παρέχοντας στους χρήστες περισσότερη ευελιξία στη συλλογή, επεξεργασία και ανάλυση δεδομένων από τις σχετικές βάσεις δεδομένων και τις αποθήκες δεδομένων . Είναι λογισμικό υποδομής γραμμένο σε Java για να “τρέχει” εφαρμογές σε μεγάλα clusters τυπικών υπολογιστών και να ενσωματώνει χαρακτηριστικά παρόμοια με αυτά του Google File System και του MapReduce. Δημιουργήθηκε για να διαχειριστεί μεγάλες ποσότητες δεδομένων τα οποία, λόγω του μεγάλου όγκου τους, δεν είναι δυνατόν να τοποθετηθούν στον δίσκο ενός υπολογιστή και έτσι τόσο τα δεδομένα όσο και η ανάλυσή τους χρειάζεται να κατακευματισθούν σε μεγάλες συστάδες υπολογιστών. Το Hadoop είναι, επίσης, γνωστό και από το MapReduce και από το κατακευματισμένο σύστημα αρχείων του Hadoop (Distributed File System, HDFS).

Το HDFS είναι ένα καταναμημένο σύστημα αρχείων, που παρέχει υψηλό βαθμό εξυπηρέτησης από κάθε μονάδα του cluster σε ένα μεγάλο σύνολο δεδομένων. Μια συστάδα με HDFS έχει δύο τύπους κόμβων, που λειτουργούν κάτω από το μοντέλο «αφέντη-σκλάβου» (master-slave): ένα namenode (αφέντη) και μια σειρά από datanodes (σκλάβοι).

- Το namenode διαχειρίζεται τον χώρο ονομάτων αρχείων. Με λίγα λόγια, διατηρεί το δέντρο του συστήματος αρχείων και τα μεταδεδομένα για όλα τα αρχεία και τους καταλόγους στο δέντρο. Αυτές οι πληροφορίες αποθηκεύονται μόνιμα στον τοπικό δίσκο του namenode με τη μορφή δύο αρχείων: την «εικόνα ονομάτων» (namespace image) και το αρχείο καταγραφής των λειτουργιών επεξεργασίας (edit log). Το namenode γνωρίζει, επίσης, σε ποια datanodes βρίσκονται τα blocks κάθε αρχείου. Η πληροφορία αυτή, όμως, δεν αποθηκεύεται μόνιμα, γιατί ανακατασκευάζεται από τα datanodes κατά την εκκίνηση του συστήματος. Μια εφαρμογή-πελάτης έχει πρόσβαση στο σύστημα αρχείων για λογαριασμό του χρήστη, επικοινωνώντας με το namenode και τα datanodes. Η εφαρμογή-πελάτης χρησιμοποιεί μια προγραμματιστική διεπαφή (Application Programming Interface ή API), παρόμοια με αυτή που ορίζει το πρότυπο Portable Operating System Interface for Unix (POSIX) για τα κοινά συστήματα αρχείων, έτσι ώστε ο κώδικας του χρήστη να μην χρειάζεται να γνωρίζει την ύπαρξη του namenode και των datanodes, για να λειτουργήσει σωστά.
- Τα datanodes αποτελούν τον ακρογωνιαίο λίθο του συστήματος αρχείων HDFS. Αποθηκεύουν και ανακτούν block όποτε τους ζητείται (από της εφαρμογές-πελάτες ή το namenode), ενώ ενημερώνουν περιοδικά το namenode με το σύνολο των blocks, που έχουν αποθηκευμένα.

Από την άλλη, χωρίς το namenode δεν μπορεί να χρησιμοποιηθεί το σύστημα αρχείων. Για την ακρίβεια, αν για κάποιον λόγο καταστραφεί ο κόμβος της συστάδας που λειτουργεί ως namenode, όλα τα αρχεία στο σύστημα

αρχείων θα χαθούν, καθώς δεν θα υπάρχει τρόπος να ανακατασκευαστούν τα αρχεία από τα blocks που υπάρχουν στα datanodes. Το HDFS, επίσης, έχει την έννοια block με προεπιλεγμένο μέγεθος να είναι 128 MB.

4.5 Apache Spark

Το Apache Spark [24] είναι ένα πλαίσιο επεξεργασίας μεγάλων δεδομένων ανοικτού κώδικα που βασίζεται στην ταχύτητα, την ευκολία χρήσης και τα εξελιγμένα αναλυτικά στοιχεία. Επιτρέπει στις εφαρμογές σε ομάδες συμπλέγματος Hadoop να τρέχουν έως και 100 φορές ταχύτερα στη μνήμη και 10 φορές πιο γρήγορα ακόμα και όταν εκτελούνται σε δίσκο. Επίσης επιτρέπει την δημιουργία, και μάλιστα γρήγορα, εφαρμογών σε Java, Scala ή Python. Το Spark τρέχει πάνω από την υπάρχουσα υποδομή του Hadoop Distributed File System (HDFS) για να παρέχει βελτιωμένες και πρόσθετες λειτουργίες. Παρέχει υποστήριξη για την ανάπτυξη εφαρμογών Spark σε υπάρχον σύμπλεγμα Hadoop v1 (με SIMR - Spark - Inside - MapReduce) ή στο σύμπλεγμα Hadoop v2 YARN ή ακόμα και στο Apache Mesos .

Το Spark διατηρεί ενδιάμεσα αποτελέσματα στη μνήμη αντί να τα γράφει στον δίσκο, κάτι που είναι πολύ χρήσιμο, ειδικά όταν απαιτείται η εργασία στην ίδια ομάδα δεδομένων πολλές φορές. Είναι σχεδιασμένο να είναι μια μηχανή εκτέλεσης που λειτουργεί τόσο σε μνήμη όσο και σε δίσκο. Οι χειριστές Spark εκτελούν εξωτερικές λειτουργίες όταν τα δεδομένα δεν ταιριάζουν στη μνήμη. Μπορεί να χρησιμοποιηθεί για την επεξεργασία συνόλων δεδομένων μεγαλύτερων από τη συσσωρευμένη μνήμη σε ένα σύμπλεγμα. Μπορεί να αποθηκεύσει ένα μέρος ενός συνόλου δεδομένων στη μνήμη και τα υπόλοιπα δεδομένα στο δίσκο.

Η βασική δομή δεδομένων του Spark είναι το Resilient Distributed Dataset (RDD). Πρόκειται για ένα αντικείμενο χαμηλού επιπέδου που επιτρέπει στο Spark να διαιρεί τα δεδομένα σε πολλαπλούς κόμβους του συμπλέγματος. Ωστόσο, οι RDDs είναι δύσκολο να εργαστούν απευθείας, έτσι χρησιμοποιούμε το Spark DataFrame που είναι χτισμένο πάνω από RDDs. Το Spark DataFrame σχεδιάστηκε για να συμπεριφέρεται πολύ όπως ένας πίνακας SQL (ένας

πίνακας με μεταβλητές στις στήλες και παρατηρήσεις στις σειρές) και είναι πιο βελτιστοποιημένο για πολύπλοκες λειτουργίες από RDDs.

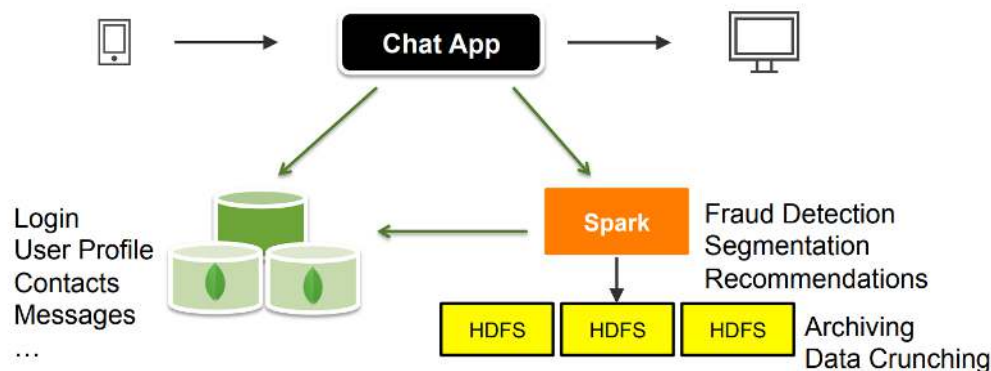
Το Spark Python API (PySpark) [25] εκθέτει το μοντέλο προγραμματισμού Spark στη Python.

4.6 MongoDB Connectors

Τα Apache Hadoop και Apache Spark μπορούν να συνεργαστούν με την βάση δεδομένων MongoDB. Ήδη πολλές εταιρείες και οργανισμοί παγκοσμίως εκμεταλλεύονται αυτήν την δυνατότητα για να αναπτύξουν πλήρεις εφαρμογές δεδομένων. Η λογική ταυτόχρονης εκμετάλλευσης (Εικόνα 14) των παραπάνω συστημάτων είναι:

- Η MongoDB τροφοδοτεί με δεδομένα την εφαρμογή σε πραγματικό χρόνο
- Τα Apache Hadoop και Apache Spark έχουν σαν είσοδο / έξοδο την MongoDB, συνδυάζουν τα δεδομένα με άλλα λειτουργικά συστήματα και τροφοδοτούν με αυτά άλλα πιο εξελιγμένα συστήματα για ανάλυση και μηχανική μάθηση.

Για να επιτευχθεί η παραπάνω σύνδεση η εταιρεία MongoDB διαθέτει δύο “plugins”, το MongoDB Hadoop Connector [26] για σύνδεση με το Hadoop και ένα το MongoDB Connector for Spark για τη σύνδεση με το Apache Spark.



Εικόνα 14 Παράδειγμα χρήσης MongoDB connector

Πηγή: https://gotocon.com/dl/goto-berlin-2015/slides/NorbertoLeite_DistributedProcessingForDistributedDatabase.pdf

5. Εφαρμογή Ιστού (Web Application)

Μια εφαρμογή ιστού είναι ένα πρόγραμμα υπολογιστή πελάτη - διακομιστή και το οποίο πρόγραμμα εκτελείται από την πλευρά του υπολογιστή-πελάτη σε ένα πρόγραμμα περιήγησης ιστού. Για την δημιουργία εφαρμογών ιστού απαιτείται ένα σύνολο υποπρογραμμάτων, υπηρεσιών, πρωτοκόλλων και εργαλείων, γνωστό με τον όρο API (Application programming interface). Το API είναι ένα είδος διεπαφής το οποίο διαθέτει ένα σύνολο λειτουργιών που επιτρέπουν στους προγραμματιστές να έχουν πρόσβαση σε συγκεκριμένα χαρακτηριστικά ή δεδομένα μιας εφαρμογής, λειτουργικού συστήματος ή άλλων υπηρεσιών. Με τον όρο Web API αναφερόμαστε σε αυτά τα σύνολα που παρέχονται μέσω του διαδικτύου και μπορεί να προσεγγιστεί χρησιμοποιώντας πρωτόκολλο HTTP. Με τα APIs επιδιώκεται η παροχή στους τελικούς χρήστες, μέσω του διαδικτύου, πόρων - πληροφοριών που υπάρχουν σε κάποια εφαρμογή - λογισμικό.

Στην παρούσα εργασία χρησιμοποιήθηκαν κάποιες υπηρεσίες, γλώσσες προγραμματισμού και εργαλεία για την ανάπτυξη της εφαρμογής και για τον έλεγχο των υπηρεσιών.

5.1 Υπηρεσίες Ιστού (Web Services)

Με τον όρο «Υπηρεσίες Ιστού» αναφερόμαστε στη διάθεση ενός λογισμικού – εφαρμογής μέσω του Διαδικτύου χρησιμοποιώντας ένα τυποποιημένο τρόπο ενσωμάτωσης για την ανταλλαγή μηνυμάτων XML. Καθώς η ανταλλαγή μηνυμάτων γίνεται με XML, δεν υπάρχει κάποια σύνδεση με λειτουργικό σύστημα και γλώσσα προγραμματισμού, γεγονός που παρέχει διαλειτουργικότητα και ευελιξία στην επικοινωνία και την ανταλλαγή μηνυμάτων μεταξύ εφαρμογής – εφαρμογής.

Μια υπηρεσία ιστού είναι μια συλλογή ανοιχτών πρωτοκόλλων και προτύπων. Πρωτόκολλα επικοινωνίας που χρησιμοποιούνται είναι:

- SOAP - Simple Object Access Protocol [28], για το οποίο ο Διεθνής Οργανισμός Προτύπων (W3C) έχει συντάξει σύσταση (recommendation).

- WSDL–Web Services Description Language [29].
- UDDI–Universal Description, Discovery, and Integration [30], τα οποία δεν υιοθετήθηκαν ευρέως, αλλά βρίσκονται ακόμα μέσα σε κάποιες εταιρίες.

5.2 REST API

Μια εξέλιξη των υπηρεσιών ιστού είναι το «RESTful web service». Το RESTful σημαίνει REpresentational State Transfer, δηλαδή βασίζεται στην τεχνολογία αναπαράστασης κρατικής μεταφοράς. Είναι ένα αρχιτεκτονικό στυλ και δεν έχει κάποιο επίσημο πρότυπο. Έχει γίνει η πιο ευρέως διαδεδομένη τεχνολογία για δημιουργία εφαρμογής ιστών, που είναι ελαφριές, επεκτάσιμες και εύκολα συντηρήσιμες. Επιπλέον, όπως προαναφέρθηκε, χρησιμοποιείται και για την επικοινωνία της MongoDB με εφαρμογές ιστού.

Με την χρήση των REST API οι πόροι προσδιορίζονται απλά μέσω υπερσυνδέσεων (URL's) και ανακτώνται ή τροποποιούνται μέσω τεσσάρων HTTP λειτουργιών (POST, PUT, GET, DELETE). Επιπλέον οι πόροι μπορούν να προσδιορίζονται ταυτόχρονα σε πολλούς πελάτες.

Η χρήση των HTTP ερωτημάτων είναι απλή:

- POST : Δημιουργία νέου πόρου
- GET : Ανάκτηση διαθέσιμου πόρου
- PUT : Ενημέρωση υπάρχοντος πόρου
- DELETE : Διαγραφή πόρου

Το REST API χρησιμοποιεί ένα τμήμα (γραμμή κατάστασης) σύμφωνα με το πρωτόκολλο RFC 2616 [32] για να ενημερώσει τους υπολογιστές – πελάτες σχετικά με την κατάσταση του αποτελέσματος του ερωτήματος. Οι κωδικοί κατάστασης χωρίζονται σε πέντε κατηγορίες (Εικόνα 15).

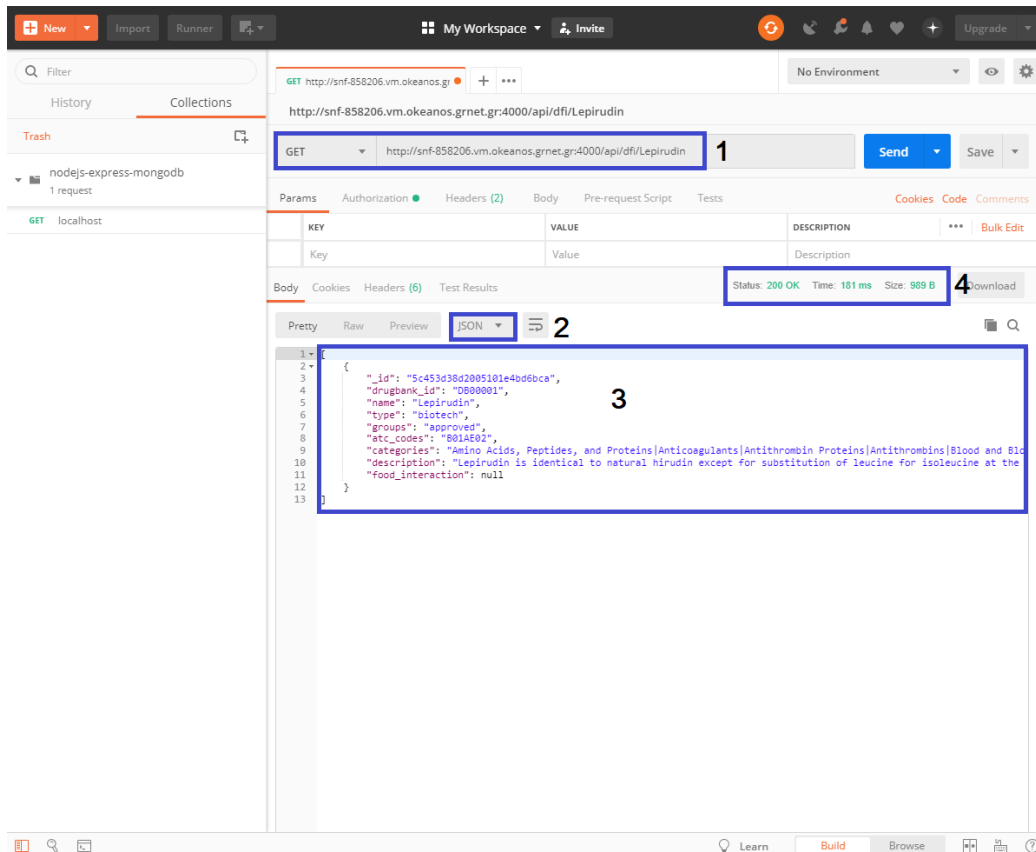
ΚΑΤΗΓΟΡΙΑ	ΠΕΡΙΓΡΑΦΗ
1xx: Ενημερωτικό	Επικοινωνεί με πληροφορίες πρωτοκόλλου μεταφοράς.
2xx: Επιτυχία	Υποδεικνύει ότι το αίτημα του πελάτη έγινε δεκτό με επιτυχία.
3xx: Ανακατεύθυνση	Υποδεικνύει ότι ο πελάτης πρέπει να λάβει κάποια επιπλέον ενέργεια για να ολοκληρώσει το αίτημά του.
4xx: Σφάλμα πελάτη	Αυτή η κατηγορία κωδικών κατάστασης σφαλμάτων δείχνει το δάκτυλο στους πελάτες.
5xx: Σφάλμα διακομιστή	Ο διακομιστής αναλαμβάνει την ευθύνη για αυτούς τους κωδικούς κατάστασης σφαλμάτων.

Εικόνα 15 Κατηγορίες κωδικών κατάστασης

Πηγή: <https://restfulapi.net/http-status-codes/>

5.3 Postman

Το Postman [33] είναι ένα εργαλείο για τη δοκιμή της λειτουργικότητας των RESTful APIs, καθώς και την ανάλυση τους. Παρέχει μία εύκολη, τόσο σε κατανόηση όσο και σε χρήση, διεπαφή χρήστη (Εικόνα 16) μέσω της οποίας γίνεται η χρήση των HTML ερωτημάτων. Ο χρήστης επιλέγει τη μορφή του ερωτήματος και τοποθετεί το url του REST API [Εικόνα 16 (1)]. Έχει τη δυνατότητα να επιλέξει τρόπο αυθεντικοποίησης (δηλαδή να εισαγάγει τα διαπιστευτήρια που χρειάζονται για την είσοδο στο REST API) καθώς και να επιλέξει τη μορφή – τύπο των δεδομένων που θα εξαχθούν (Pretty δηλαδή JSON, XML, HTML, text, auto και Raw ή Preview) [Εικόνα 16 (2) και (3)]. Μαζί με τα αποτελέσματα – πόρους δίνεται και ο κωδικός κατάστασης του ερωτήματος, ο χρόνος που χρειάστηκε για την ολοκλήρωση του και το μέγεθος των αποτελεσμάτων [Εικόνα 16 (4)]. Το Postman διατίθεται δωρεάν και σε Pro έκδοση που παρέχει περισσότερες, ποιοτικά και ποσοτικά, δυνατότητες χρήσης, όπως το monitoring. Τέλος δίνει τη δυνατότητα δημιουργίας τεκμηρίωσης (documentation) ενός REST API.



Εικόνα 16 Διεπαφή χρήστη Postman

5.4 Γλώσσες προγραμματισμού

Για την ανάπτυξη της εφαρμογής ιστού χρησιμοποιήθηκαν κάποιες γλώσσες προγραμματισμού για τις οποίες γίνεται αναφορά παρακάτω.

5.4.1 HTML (HyperText Markup Language) και HTML5

Η HTML είναι μια γλώσσα συγγραφής εγγράφων (ιστοσελίδων) για τον παγκόσμιο ιστό. Με την HTML καθορίζεται η δομή και η διάταξη των δεδομένων στο έγγραφο με την χρήση ετικετών, όπως <html>, <head>, <body>, <div>, <section>, <h1>, <p>. Η πέμπτη έκδοση της HTML είναι πολύ σημαντική και αποτελεί προδιαγραφή για τον Διεθνή Οργανισμό Προτύπων (W3C), γνωστή ως HTML5.

5.4.2 CSS (Cascading Style Sheets)

Το CSS είναι υπεύθυνο για την μορφοποίηση και την εμφάνιση των εγγράφων – ιστοσελίδων στον παγκόσμιο ιστό. Μέχρι την έκδοση HTML 3.0 η μορφοποίηση περιλαμβανόταν μέσα στο ίδιο αρχείο που ήταν και η δομή και η διάταξη. Από την έκδοση HTML5 η μορφοποίηση αποθηκεύεται σε ξεχωριστό αρχείο με κατάληξη .css, δηλαδή σε αρχείο css.

5.4.3 Javascript

Η Javascript είναι μια διερμηνευμένη και δομημένη γλώσσα προγραμματισμού. Με τον όρο «διερμηνευμένη» αναφερόμαστε στο γεγονός ότι η υλοποίηση της γίνεται από ένα διερμηνέα και συγκεκριμένα από τους φυλλομετρητές ιστού. Με τον όρο «δομημένη» αναφερόμαστε στο γεγονός ότι αποτελείται από ένα σύνολο εντολών (διαδικασία) προς εκτέλεση όταν υπάρξει κλήση της διαδικασίας.

Η Javascript αναπτύχθηκε από τον Brendan Eich που εργαζόταν για την Netscape, με την επωνυμία Mocha. Αργότερα ονομάστηκε LiveScript και τέλος ονομάστηκε Javascript. Ήταν βασισμένη στην γλώσσα προγραμματισμού C. Στα διάφορα στάδια ανάπτυξης της πήρε κάποια στοιχεία από την Java, όπως λέξεις-κλειδιά της Java, τις συμβάσεις ονοματολογίας για τις κύριες βιβλιοθήκες, κάποια αντικείμενα πχ Date και Time. Πέρα όμως από αυτά τα στοιχεία είναι διαφορετική γλώσσα από την Java, παρά τις ομοιότητές τους. Αυτές οφείλονται επειδή και οι δύο γλώσσες βασίστηκαν στην C.

Η Javascript δημιουργήθηκε για να εκτελείται από την πλευρά του υπολογιστή – πελάτη (client-side) και όχι από την πλευρά του εξυπηρετητή ιστού και δεν απαιτεί την εγκατάσταση ειδικού υλικού ή λογισμικού. Χρησιμοποιείται για την ανάπτυξη ιστοσελίδων εφαρμογών ιστού, εφαρμογών και προγραμμάτων. Η πιο συνηθισμένη χρήση της JavaScript είναι να προσθέσει συμπεριφορά και προσωπικότητα στην HTML στην πλευρά του χρήστη, που είναι γνωστό και ως Δυναμική HTML.

Η Javascript [34] στην αρχή της ανάπτυξής της υποβαθμίστηκε από πολλούς προγραμματιστές. Με την εμφάνιση του AJAX [35] (Asynchronous

JavaScript and XML), η JavaScript βγήκε στο προσκήνιο και πλέον είναι από τις πιο διαδεδομένες γλώσσες προγραμματισμού. Με το AJAX οι εφαρμογές ιστού επιτυγχάνουν την αποστολή και ανάκτηση δεδομένων ασύγχρονα από ένα διακομιστή (στο παρασκήνιο) χωρίς να επηρεάζεται η συμπεριφορά της υπάρχουσας σελίδας.

Κι ενώ η JavaScript ήταν client-side γλώσσα προγραμματισμού, τα τελευταία χρόνια αναπτύχθηκαν διάφορες πλατφόρμες ανάπτυξης λογισμικού βασισμένες στην JavaScript με σκοπό την ανάπτυξη εφαρμογών ιστού στην πλευρά του εξυπηρετητή – διακομιστή. (server-side)

5.4.4 React.js

Το React [36] είναι μια δηλωτική, αποδοτική και ευέλικτη βιβλιοθήκη JavaScript για τη δημιουργία διεπαφών χρήστη (UI). Επιτρέπει τη δημιουργία σύνθετων UI από μικρά και απομονωμένα κομμάτια κώδικα που ονομάζονται "components".

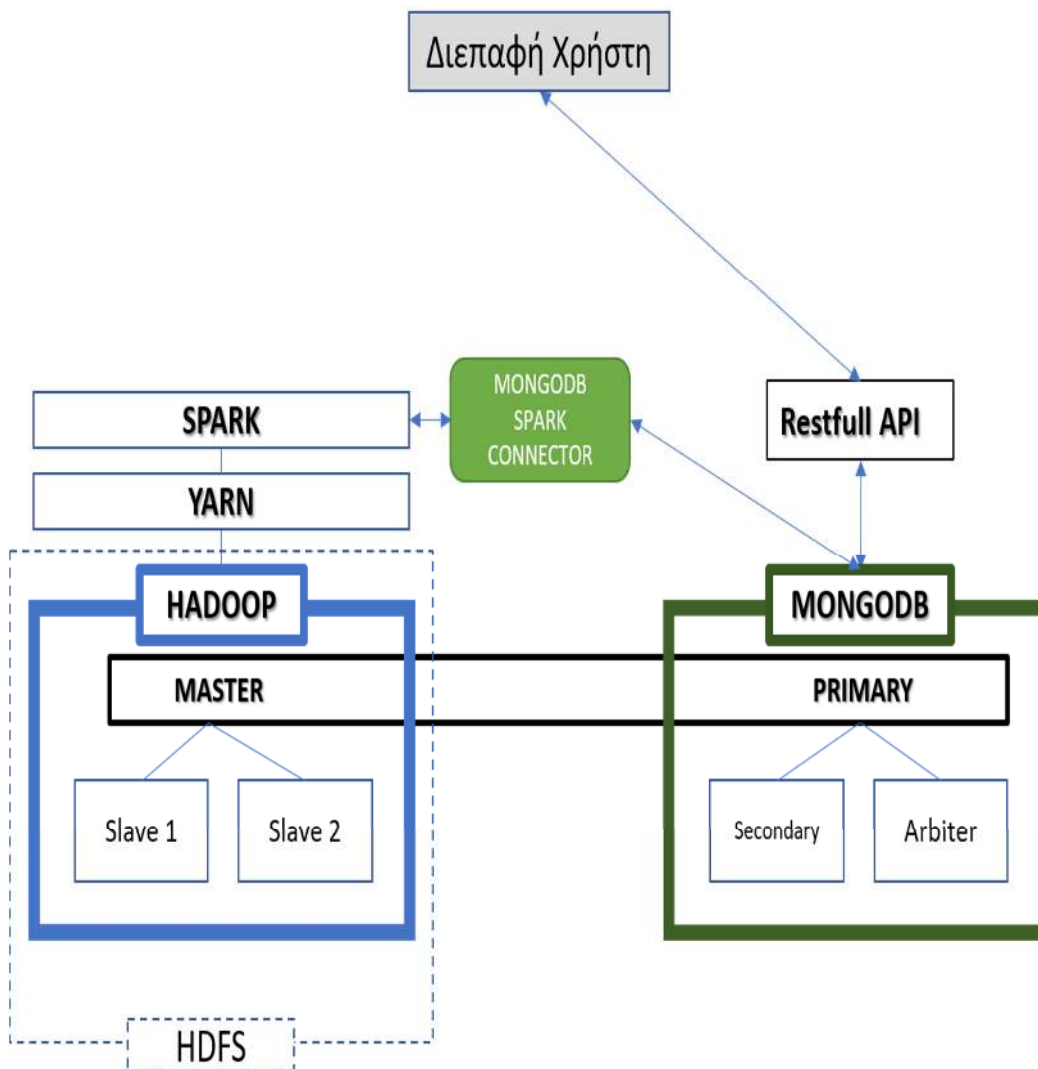
5.4.5 Node.js – Express.js

Το Node.js είναι μια πλατφόρμα ανάπτυξης λογισμικού στην πλευρά του διακομιστή. Είναι βασισμένη πάνω στην γλώσσα προγραμματισμού Javascript. Έχει εστιάσει στην ασύγχρονη επικοινωνία και επιτυγχάνεται η δημιουργία δυναμικών ιστοσελίδων από την αποστολή της ιστοσελίδας από τον εξυπηρετητή Ένα ελάχιστο και ευέλικτο πλαίσιο εφαρμογών Node.js είναι το Express που είναι κατάλληλο για την δημιουργία εφαρμογές ιστού αλλά και κινητών.

6. Αρχιτεκτονική και Διαδικασία

Σε αυτό το κεφάλαιο θα αναλυθεί η αρχιτεκτονική (Εικόνα 17) και η διαδικασία ανάπτυξης της εφαρμογής ιστού. Μπορούμε να διακρίνουμε 4 βασικές κατηγορίες:

- Η δημιουργία συμπλέγματος Hadoop με τη χρήση του υπολογιστικού νέφους Okeanos
- Η εγκατάσταση και η ανάπτυξη της βάσης δεδομένων MongoDB
- Η δημιουργία RESTful API
- Η δημιουργία της διεπαφής του τελικού χρήστη



Εικόνα 17 Αρχιτεκτονική Συστήματος

Για την υλοποίηση του σκοπού μας χρησιμοποιήσαμε υπολογιστικούς πόρους από τον Οκεανο για να εκμεταλλευτούμε τα πλεονεκτήματα του cloud computing. Από τον Οκεανο διατίθενται σε κάθε εγγεγραμμένο χρήστη οι παρακάτω πόροι:

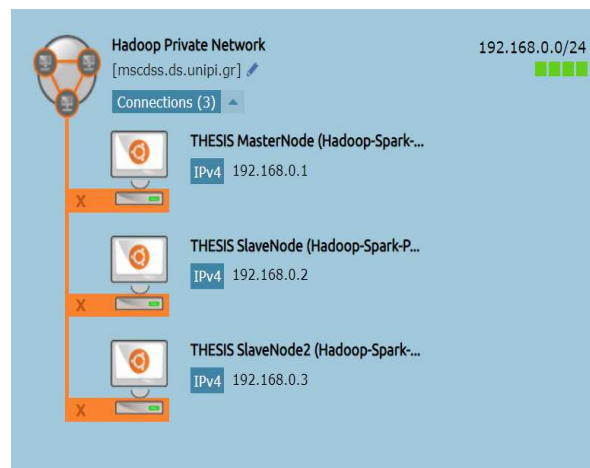
- ❖ CPUs: 20 cores
- ❖ RAM: 50 Gb
- ❖ Disk : 600 Gb
- ❖ Public ip: 1
- ❖ Private Networks: 10

Έχοντας ήδη κάνει ακαδημαϊκό λογαριασμό στον Οκεανο και συμμετέχοντας στο project του Τμήματος Ψηφιακών Συστημάτων δημιουργήσαμε τρεις εικονικές μηχανές με τα εξής χαρακτηριστικά η καθεμία:

- ❖ CPUs: 4 cores
- ❖ RAM: 8 Gb
- ❖ Disk : 40 Gb
- ❖ Public ip: 1 (μόνο για τον Master κόμβο)
- ❖ Private Networks: 1
- ❖ OS: Ubuntu Server 16.04.03 LTS

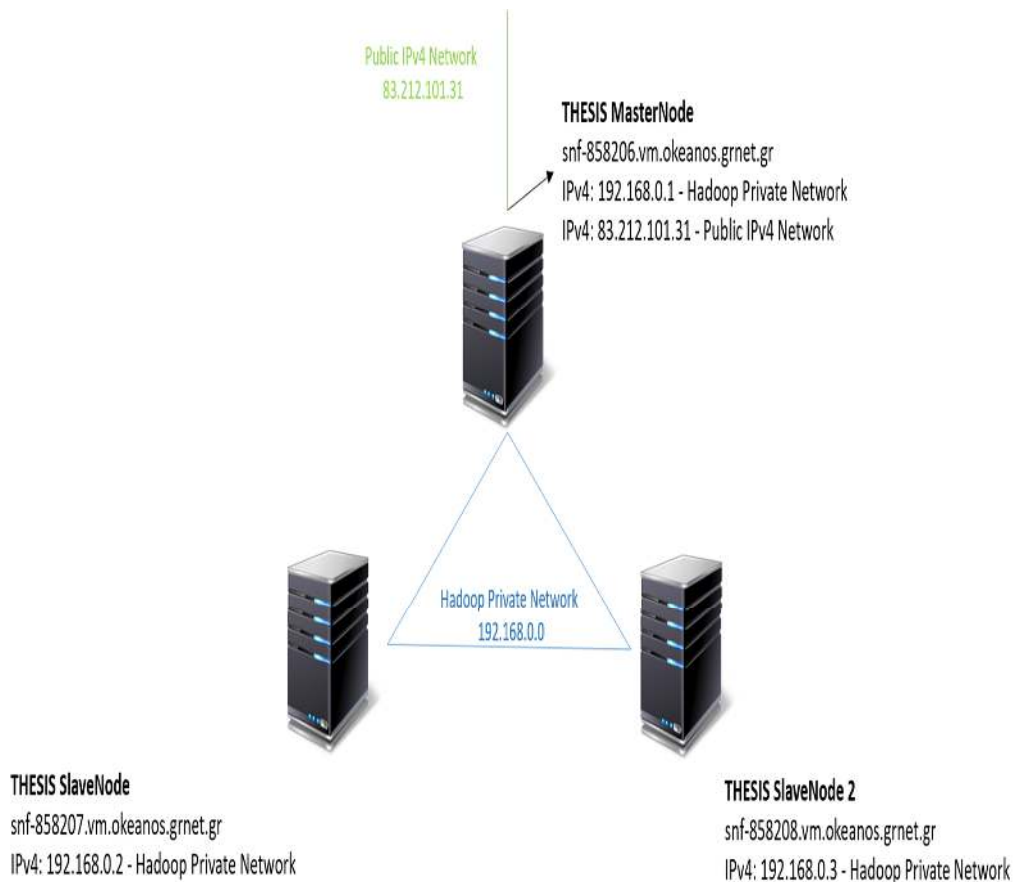
Για την δημιουργία του συμπλέγματος Hadoop χρησιμοποιήσαμε την μία εικονική μηχανή ως Master κόμβο, στον οποίο διαθέσαμε και την δημόσια ip, και τις άλλες δύο εικονικές μηχανές ως slave κόμβους.

Επιπλέον δημιουργήσαμε ένα ιδιωτικό δίκτυο (Hadoop Private Network – Εικόνα 18) για την επικοινωνία μεταξύ των κόμβων.



Εικόνα 18 Ιδιωτικό δίκτυο

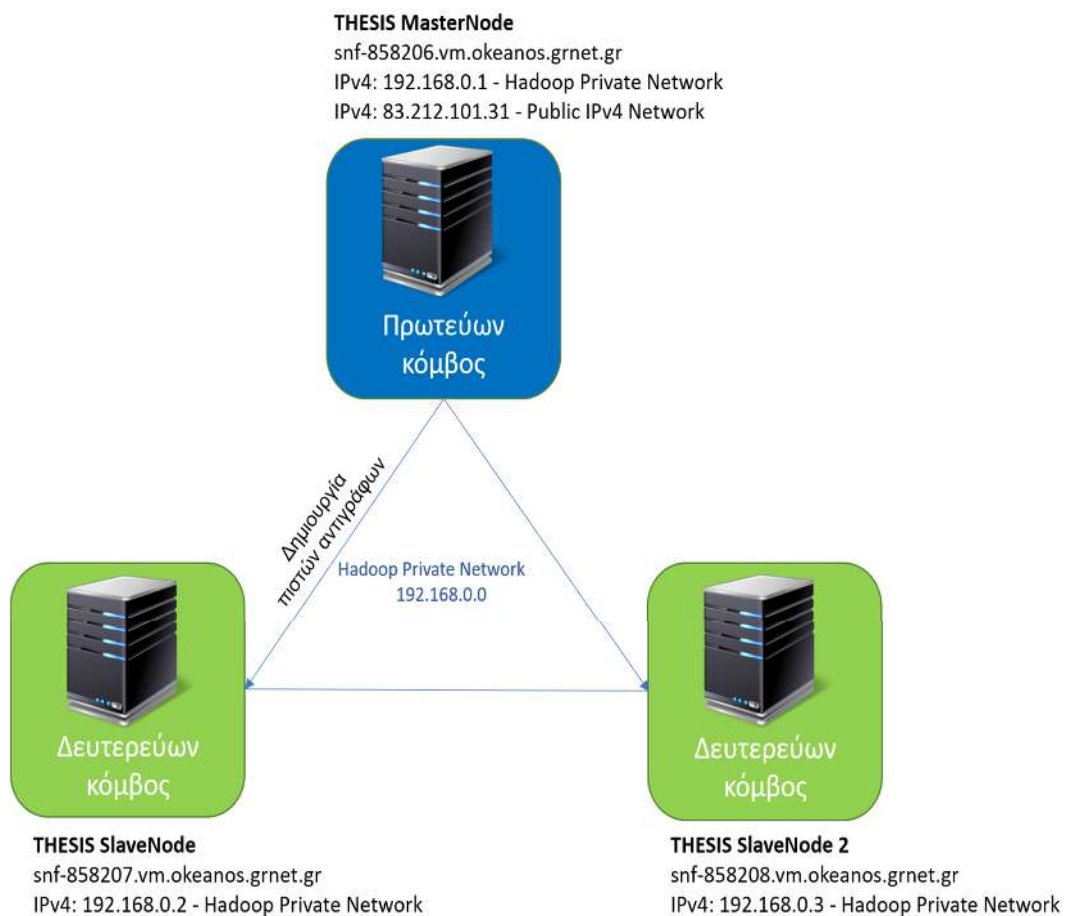
Έτσι η αρχιτεκτονική του συμπλέγματος Hadoop φαίνεται στην παρακάτω εικόνα (Εικόνα 19).



Εικόνα 19 Σύμπλεγμα Hadoop

Εγκαταστήσαμε τα Hadoop, Yarn και Spark στους τρεις κόμβους. Εγκαταστήσαμε το Hadoop για να είναι σε θέση το σύστημά μας να αποθηκεύσει, να ευρετηριάσει, να διαχειριστεί και να αναλύσει αργότερα τα ιατροφαρμακευτικά δεδομένα. Με την βοήθεια του Apache Yarn, που είναι υπεύθυνο για την κατανομή των πόρων του συστήματος στις διάφορες εφαρμογές που εκτελούνται και για τον προγραμματισμό εργασιών που πρέπει να εκτελούνται στους διαφορετικούς κόμβους, εγκαταστήσαμε το Spark για να εκμεταλλευτούμε την ταχύτητα και την δυνατότητα επεξεργασίας δεδομένων σε πραγματικό χρόνο.

Στο επόμενο στάδιο εγκαταστήσαμε την MongoDB στους τρεις κόμβους. Θέλοντας να εκμεταλλευτούμε το χαρακτηριστικό της υψηλής διαθεσιμότητας που παρέχει η MongoDB δημιουργήσαμε ένα replica set με τις τρεις υπάρχουσες εικονικές μηχανές. Έτσι στον Master κόμβο που διαθέτε και την δημόσια ip δώσαμε τον ρόλο του Πρωτεύοντα (Primary) κόμβου. Στους άλλους δύο κόμβους έπρεπε να αποφύγουμε την ύπαρξη ζυγού αριθμού δευτερευόντων κόμβων, ώστε να υπάρχει πλειοψηφία κατά την ψηφοφορία στην διαδικασία εκλογής νέου πρωτεύοντα κόμβου. Έτσι στον ένα κόμβο δώσαμε ρόλο δευτερεύοντα κόμβου (secondary), ενώ στον άλλο κόμβο δώσαμε τον ρόλο του διαιτητή (Arbiter) (Εικόνα 20), με αποτέλεσμα στο σύστημα να διατηρείται ένα πιστό αντίγραφο της βάσης δεδομένων, αφού ο κόμβος – διαιτητής δεν τηρεί αντίγραφο.



Εικόνα 20 MongoDB με κόμβο διαιτητή

Αφού εγκαταστήσαμε την βάση δεδομένων, στο επόμενο στάδιο, κάναμε λήψη του συνόλου φαρμακευτικών δεδομένων Drugbank, αποσυμπιέσαμε το αρχείο και πήραμε την τελική μορφή του συνόλου σε xml μεγέθους 758 MB. Επειδή όπως έχουμε προαναφέρει το σύνολο δεδομένων περιέχει πάνω από 200 πεδία για κάθε φάρμακο, με τη χρήση της γλώσσας προγραμματισμού Python, απομονώσαμε τα πεδία του συνόλου που μας είναι χρήσιμα για την υλοποίηση της εφαρμογής μας, τα οποία φαίνονται παρακάτω:

- `drugbank_id` : Ο αναγνωριστικός αριθμός κάθε φαρμάκου στην βάση Drugbank
- `name`: Το όνομα του φαρμάκου
- `type`: Ο τύπος του φαρμάκου (Biotech ή Small Molecule)
- `groups`: Η ομάδα του φαρμάκου ανάλογα με την έγκρισή του
- `atc_codes`: Ο κωδικός ανατομικής, θεραπευτικής, χημικής ταξινόμησης που έχει δοθεί σε κάθε φάρμακο από τον Παγκόσμιο Οργανισμό Υγείας (World Health Organization – W.H.O)
- `categories`: Η κατηγορία σύνδεσης του φαρμάκου (στόχος, ενζυμο, μεταφορέας, φορέας)
- `inchikey`: Το διεθνές χημικό αναγνωριστικό
- `inchi description`: Η περιγραφή του διεθνές χημικού αναγνωριστικού
- `drug_interaction_drug_id`: Το `drugbank` αναγνωριστικό του αλληλοεπιδρώντος φαρμάκου
- `drug_interaction_drug_name`: Το όνομα του αλληλοεπιδρώντος φαρμάκου
- `drug_interaction_drug_description`: Η περιγραφή της αλληλεπίδρασης
- `food_interaction`: Η αλληλεπίδραση του φαρμάκου με κάποιο τρόφιμο

Μετά την επιλογή των παραπάνω πεδίων, εξάγουμε το αρχείο Drugbank.tsv μεγέθους ~85 MB. Κατόπιν εισάγουμε το αρχείο αυτό στην MongoDB. Παρατηρώντας την βάση, τα πεδία `drug_interaction_drug_id`, `drug_interaction_drug_name` και `drug_interaction_drug_description` είναι τύπου πίνακα (array). Έτσι αποσυνθέτουμε τους πίνακες αυτούς με

αποτέλεσμα να εμφανιστούν 745.070 αλληλεπιδράσεις – έγγραφα και ο όγκος της βάσης μας να αυξηθεί σε 1.1 Gb.

Στη συνέχεια εγκαθιστούμε το node.js και express.js στον master κόμβο για την δημιουργία του RESTful service και κατόπιν εγκαθιστούμε την react.js για την δημιουργία της διεπαφής. Με το RESTful API δημιουργήσαμε και τα URIs που απαιτούνται για τα ερωτήματα. Αυτά είναι :

(στη συνέχεια του url του server: <http://snf-858206.vm.okeanos.grnet.gr:4000/api/>)

➤ GET

- ❖ /drug/:input_drug :Επιστρέφει πληροφορίες και όλες τις αλληλεπιδράσεις του φαρμάκου που καταχωρούμε
- ❖ /dfi/:input_drug : Επιστρέφει μόνο τις αλληλεπιδράσεις του φαρμάκου
- ❖ /drugs_info/:input_drug : Επιστρέφει πληροφορίες του φαρμάκου
- ❖ /ddi/ ?drug1&drug2 : Επιστρέφει την αλληλεπίδραση μεταξύ δύο φαρμάκων

➤ POST

- ❖ /newdrug : Εισάγουμε νέο φάρμακο

➤ UPDATE

- ❖ /drug/:id : Ενημερώνουμε ένα υπάρχον φάρμακο

➤ DELETE

- ❖ /drug/:id: Διαγράφουμε ένα υπάρχον φάρμακο

Αφού ελέγξαμε την κανονική λειτουργία των URIs μέσω του Postman, συνεχίσαμε με την εγκατάσταση του react.js για την δημιουργία της διεπαφής χρήστη.

Δημιουργήσαμε τέσσερις σελίδες, όπως φαίνονται παρακάτω :

- ✚ Στοιχεία φαρμάκων: Εισάγοντας το όνομα ή το atc κωδικό ή το Drugbank_id ενός φαρμάκου μας επιστρέφει τα πεδία Name, Drugbank id, Atc code, Type, Categories και Description (Εικόνα 21)

The screenshot shows the 'Drugs App - a REST API' interface. At the top, there are navigation tabs: 'Στοιχεία Φαρμάκων', 'Αλληλεπιδράσεις ανά Φάρμακο', 'Αλληλεπίδραση Φαρμάκων', and 'Αλληλεπίδραση Τροφών - Φαρμάκων'. The main heading is 'Drugs App - a REST API'. Below it is a 'Search a Drug' section with an input field containing 'Leparudin' and a 'Search..' button. The search results are displayed in a table with the following columns: Name, Drugbank id, Atc code, Type, Categories, and Description.

Name	Drugbank id	Atc code	Type	Categories	Description
Leparudin	DB00001	B01AE02	biotech	Amino Acids, Peptides, and Proteins Anticoagulants Antithrombin Proteins Antithrombins Blood and Blood Forming Organs Cardiovascular Agents Enzyme Inhibitors Fibrin Modulating Agents Hematologic Agents Peptides Protease Inhibitors Proteins Serine Proteinase Inhibitors Serpins	Leparudin is identical to natural hirudin except for substitution of leucine for isoleucine at the N-terminal end of the molecule and the absence of a sulfate group on the tyrosine at position 63. It is produced via yeast cells. Bayer ceased the production of leparudin (Refludan) effective May 31, 2012.

Εικόνα 21 Σελίδα στοιχείων κάθε φαρμάκου

- ✚ Αλληλεπιδράσεις ανά φάρμακο: Εισάγοντας το όνομα ή το atc κωδικό μας επιστρέφει όλες τις αλληλεπιδράσεις του φαρμάκου με άλλα φάρμακα (Εικόνα 22)

The screenshot shows the 'Each Drug Interactions - App' interface. At the top, there are navigation tabs: 'Στοιχεία Φαρμάκων', 'Αλληλεπιδράσεις ανά Φάρμακο', 'Αλληλεπίδραση Φαρμάκων', and 'Αλληλεπίδραση Τροφών - Φαρμάκων'. The main heading is 'Each Drug Interactions - App'. Below it is a 'Find interactions per Drug!' section with an input field containing 'Leparudin' and a 'Search..' button. The search results are displayed in a table with the following columns: Drug, Drug 2, and Interaction.

Drug	Drug 2	Interaction
Leparudin	with (1R)-limonene	= (1R)-limonene may increase the anticoagulant activities of Leparudin.
Leparudin	with Abacavir	= The serum concentration of Abacavir can be decreased when it is combined with Leparudin.
Leparudin	with Abciximab	= Abciximab may increase the anticoagulant activities of Leparudin.
Leparudin	with Aceclofenac	= Aceclofenac may increase the anticoagulant activities of Leparudin.
Leparudin	with Acemetacin	= The risk or severity of bleeding can be increased when Leparudin is combined with Acemetacin.
Leparudin	with Acenocoumarol	= Acenocoumarol may increase the anticoagulant activities of Leparudin.
Leparudin	with Acetylsalicylic acid	= Acetylsalicylic acid may increase the anticoagulant activities of Leparudin.

Εικόνα 22 Σελίδα αλληλεπιδράσεων ανά φάρμακο

- Αλληλεπιδράσεις φαρμάκων: Εισάγοντας στο κάθε πεδίο το όνομα ή το Drugbank_id του φαρμάκου, μας επιστρέφει την αλληλεπίδραση μεταξύ αυτών των δύο φαρμάκων, αν υπάρχει (Εικόνα 23)

Drug	Drug 2	Interaction
Abacavir	with Lepirudin	= The serum concentration of Abacavir can be decreased when it is combined with Lepirudin.

Εικόνα 23 Σελίδα αλληλεπιδράσεων φαρμάκου- φαρμάκου

- Αλληλεπίδραση Τροφών- Φαρμάκων: Εισάγοντας το όνομα ή το atc κωδικό ή το Drugbank_id ενός φαρμάκου μας επιστρέφει τα πεδία Name, Type, Atc_code, Categories και Food reaction (Εικόνα 24)

Name	Type	Atc code	Categories	Food reaction
Pravastatin	small molecule	C10AA03C 10BX02J1 08403	Anticholinergic Agents BCRP ABCG2 Inhibitors BCRP ABCG2 Substrates Cardiovascular System Cytochrome P-450 CYP3A Substrates Cytochrome P-450 CYP3A4 Substrates Cytokine Inhibitors P-290 CYP3A5 Substrates Cytokine P-450 L1 enzyme Inhibitors L1 enzyme Inhibitors Hydralazine Hydralazine, Aromatic Hydralazine, Cyclic-β-Hydroxypropyl Glutaryl-CoA Reductase Inhibitors Hydrophilic Agents Lipid Modifying Agents Lipid Modifying Agents, PPARα Lipid Regulating Agents Naphthalenes Norepinephrine OAT3 Substrates OATP1B1/SLC01B1 Substrates P-glycoprotein ABCB1 Substrates Polycyclic Compounds Polycyclic Hydrocarbons, Aromatic Toxic Acetates	Avoid alcohol Avoid drastic changes in dietary habit Take without regard to meals.

Εικόνα 24 Σελίδα αλληλεπιδράσεων φαρμάκου- τροφίμου

7. Συμπεράσματα – Προοπτικές

Σκοπός της συγκεκριμένης διπλωματικής ήταν η ανάπτυξη μιας εφαρμογής ιστού για την εύκολη ενημέρωση από τους επαγγελματίες υγείας των αλληλεπιδράσεων φαρμάκων τόσο μεταξύ τους όσο και με τρόφιμα. Τα δεδομένα που διαχειριστήκαμε ήταν από την βάση της Drugbank που ενώ εκ πρώτης όψεως εμφανίζεται σαν μια μικρή βάση δεδομένων κατά την επεξεργασία της εμφανίζεται σε ένα μικρό ποσοστό το πρόβλημα διαχείρισης των μεγάλων δεδομένων. Κι ενώ αναμένεται ραγδαία αύξηση των ιατροφαρμακευτικών δεδομένων τα επόμενα χρόνια, γίνεται αντιληπτό ότι απαιτείται να χρησιμοποιηθούν στην ανάπτυξη των νέων πληροφοριακών συστημάτων υγείας τεχνικές και εργαλεία διαχείρισης και ανάλυσης μεγάλων δεδομένων.

Η εφαρμογή θα μπορούσε να αποτελεί μέρος ενός πληροφοριακού συστήματος υγείας, ώστε να μπορεί ο κάθε επαγγελματίας υγείας να ενημερώνεται για τις αλληλεπιδράσεις φαρμάκων που αποτελεί ένα πολύ σημαντικό στοιχείο στην ηλεκτρονική συνταγογράφηση. Ως επόμενο βήμα θα μπορούσε να χρησιμοποιηθεί το σύστημα μας για την αποθήκευση ιατρικών δεδομένων και για την ανάλυση τους. Τέλος με την ύπαρξη των ιατρικών δεδομένων κάποιου ασθενούς, όπως το ιστορικό φαρμακευτικής αγωγής, μπορεί να γίνει έλεγχος των αλληλεπιδράσεων των συνταγογραφούμενων φαρμάκων σε πραγματικό χρόνο κατά την διαδικασία της ηλεκτρονικής συνταγογράφησης.

Βιβλιογραφία - Πηγές

- [1] Νόμος υπ. αριθμ 4238 Πρωτοβάθμιο Εθνικό Δίκτυο Υγείας (Π.Ε.Δ.Υ.), αλλαγή σκοπού Ε.Ο.Π.Υ.Υ. και λοιπές διατάξεις.
<https://www.taxheaven.gr/laws/law/index/law/578>
- [2] Νόμος υπ. αριθμ. 3892, Ηλεκτρονική καταχώριση και εκτέλεση ιατρικών συνταγών και παραπεμπτικών ιατρικών εξετάσεων, ΦΕΚ 189, 4 Νοεμβρίου 2010, σσ. 4225-4232
- [3] Leibovich ER, Deamer RL, Sanserson LA. Food-drug interactions: careful drug selection and patient counseling can reduce the risk in older patients. *Geriatrics*. 2004; 59:19-33
- [4] Teresi ME, Morgan DE. Attitudes of health-care professionals toward patient counseling on drug-nutrient interactions. *Annals of Pharmacotherapy*. 1994; 28:576-580.
- [5] Προδρομίτη Γερασιμούλα, Μεγάλα Δεδομένα - Η Εξόρυξη τους και η συμβολή τους στην Επιχειρηματική Ευφύια, 2017, 14-25
- [6] Big Data – Gartner , <https://www.gartner.com/it-glossary/big-data/>
- [7] Oracle, [Working With the Schematron HL7 V3 Sample Project](#)
- [8] FHIR R4, [Introduction HL7 FHIR](#)
- [9] Drugbank, <https://www.drugbank.ca/>
- [10] RxNorm U.S. National Library of Medicine, <https://www.nlm.nih.gov/research/umls/rxnorm/>
- [11] DailyMed, <https://dailymed.nlm.nih.gov/dailymed/>
- [12] DGIdb [12], <http://www.dgidb.org/>
- [13] FDA – OpenFDA, <https://open.fda.gov/tools/downloads/>
- [14] KEGG DRUG, <https://www.genome.jp/kegg/drug/>

- [15] Reactome, <https://reactome.org/>
- [16] SIDER, <http://sideeffects.embl.de/download/>
- [17] Γαληνός, <https://www.galinos.gr>
- [18] Catell, R. (2010): Scalable SQL and NoSQL data stores. ACM SIGMOD Record, Volume 39 Issue 4, December 2010, pages 12-27.
- [19] DB-Engines ranking, <https://db-engines.com/en/ranking>
- [20] MongoDB, <https://www.mongodb.com/>
- [21] Βασικά χαρακτηριστικά MongoDB, <https://docs.mongodb.com/manual/introduction/#key-features>
- [22] Okeanos - IaaS, <https://okeanos.grnet.gr/home/>
- [23] Apache Hadoop, <https://hadoop.apache.org/>
- [24] Apache Spark <https://spark.apache.org/>
- [25] Spark Python API (PySpark) <https://spark.apache.org/docs/latest/api/python/index.html>
- [26] MongoDB Hadoop Connector, <https://docs.mongodb.com/ecosystem/tools/hadoop/>
- [27] MongoDB Connector for Spark, <https://docs.mongodb.com/spark-connector/current/>
- [28] SOAP- <http://www.w3.org/TR/soap12-part0/>
- [29] WSDL 2.0 - <http://www.w3.org/TR/wsdl20-primer>
- [30] UDDI - <http://uddi.xml.org/>
- [31] RESTful API, <https://restfulapi.net>
- [32] RFC 2616, <https://www.ietf.org/rfc/rfc2616.txt>

- [33] Postman, <https://www.getpostman.com/>
- [34] JavaScript, <https://el.wikipedia.org/wiki/JavaScript>
- [35] AJAX, [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))
- [36] React, <https://reactjs.org/>
- [37] Node.js, <https://nodejs.org/>
- [38] Express.js, <https://expressjs.com/>