



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΤΜΗΜΑ ΛΟΓΙΣΤΙΚΗΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

**ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ, ΛΟΓΙΣΤΙΚΗ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΑ
ΣΥΣΤΗΜΑΤΑ**

**Διασφάλιση ποιότητας με την τεχνολογία
του Blockchain**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΕΥΘΥΜΙΑ ΠΑΠΑΧΡΥΣΟΠΟΥΛΟΥ
02721**

Επιβλέπων : ΠΑΝΑΓΙΩΤΗΣ ΝΟΤΟΠΟΥΛΟΣ

Θεσσαλονίκη, Μάρτιος 2023

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ	
ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ	III
ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ	IV
ΠΕΡΙΛΗΨΗ	V
ABSTRACT	VI
ΚΕΦΑΛΑΙΟ 1: Εισαγωγή	1
1.1 Κίνητρα και στόχοι της διπλωματικής.....	3
1.2 Διάρθρωση της Εργασίας.....	7
ΚΕΦΑΛΑΙΟ 2: Θεωρητικό υπόβαθρο.....	9
2.1 Εφοδιαστική αλυσίδα	9
2.1.1 Διαχείριση εφοδιαστικής αλυσίδας.....	9
2.1.2 Η ροή της εφοδιαστικής αλυσίδας.....	9
2.1.3 Γαλακτοκομικά προϊόντα	10
2.2 Ιχνηλασιμότητα τροφίμων	11
2.2.1 Εισαγωγή Ιχνηλασιμότητας	11
2.2.2 Είδη Ιχνηλασιμότητας	12
2.2.3 Συστήματα Ιχνηλασιμότητας	14
2.2.4 Εφαρμογή ιχνηλασιμότητας και τεχνολογίες.....	15
2.2.5 Διασφάλιση ποιότητας	16
2.3 Blockchain.....	16
2.3.1 Τι είναι το Blockchain	16
2.3.2 Πως λειτουργεί το Blockchain.....	17
2.3.3 Χαρακτηριστικά του Blockchain	18
2.3.4 Peer-to-Peer δίκτυα	19
2.3.5 Bitcoin και Ethereum.....	20
2.3.6 Hyperledger Project.....	21
2.3.7 Permissioned και Permissionless Blockchains	22

2.3.8 Κατανεμημένο καθολικό (Distributed Ledger)	24
2.3.9 Έξυπνα συμβόλαια (Smart Contracts)	25
2.3.10 Συναίνεση (Consensus)	27
2.3.11 Εφαρμογές blockchain στην εφοδιαστική αλυσίδα σήμερα	29
ΚΕΦΑΛΑΙΟ 3: Τεχνολογικό πλαίσιο	33
3.1 Blockchain Technologies	33
3.1.1 Hyperledger Fabric & Composer	34
3.2 Παράλληλες Τεχνολογίες και Εργαλεία	36
3.2.1 JavaScript	37
3.2.2 Web Playground	37
3.2.3 CLI Utilities	38
3.2.4 VSCode και Extensions	38
3.2.5 Docker	39
3.2.6 Debugging	39
ΚΕΦΑΛΑΙΟ 4: Δημιουργία συστήματος αποκεντρωμένης ιχνηλασιμότητας στη βιομηχανία γάλακτος	40
4.1 Στόχοι	40
4.2 Προσέγγιση	41
4.3 Μοντέλο	42
ΚΕΦΑΛΑΙΟ 5: Υλοποίηση και Πειραματικά αποτελέσματα	44
5.1 Δομή	44
5.2 Λειτουργίες	45
5.3 Συμμετέχοντες	47
5.4 Έννοιες	48
5.5 Περιουσιακά στοιχεία	48
5.6 Συναλλαγές και επιχειρησιακή λογική	50
ΚΕΦΑΛΑΙΟ 6: Επίλογος	54
6.1 Σύνοψη και συμπεράσματα	54
6.2 Όρια και περιορισμοί της έρευνας	56
6.3 Μελλοντικές επεκτάσεις	57
ΒΙΒΛΙΟΓΡΑΦΙΑ	59
ΠΑΡΑΡΤΗΜΑ Ι: Εγκατάσταση και Ανάπτυξη	64
ΠΑΡΑΡΤΗΜΑ ΙΙ: Κώδικας	71

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

Πίνακας 1 Σύγκριση χαρακτηριστικών Permissioned με Permissionless Blockchains (International Bank for Reconstruction and Development, 2017).....	23
Πίνακας 2 Σύγκριση χαρακτηριστικών Bitcoin, Ethereum και Hyperledger (Friebe, 2017)	24
Πίνακας 3 Πλαίσιο βασικών αναγκών συμμετεχόντων.....	43
Πίνακας 4 Composer Modeling Language συστήματος	45
Πίνακας 5 Πίνακας μεταβλητών Milk Asset	49

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

Εικόνα 1 Εξέλιξη πωλήσεων γαλακτοκομικών (Μανιφάβα, 2018)	5
Εικόνα 2 Ροή της εφοδιαστικής αλυσίδας.....	10
Εικόνα 3 Δομή του Blockchain (International Bank for Reconstruction and Development, 2017).....	18
Εικόνα 4 Επικοινωνία Peer-2-Peer.....	19
Εικόνα 5 Blockchain Frameworks και Εργαλεία του Hyperledger.....	21
Εικόνα 6 Κατανεμημένο καθολικό (International Bank for Reconstruction and Development, 2017).....	25
Εικόνα 7 Διάρκεια ζωής του οχήματος μέσω Smart Contracts σε blockchain. Παραγωγός-Πελάτης-Απόσυρση.....	26
Εικόνα 8 Μηχανισμός συναίνεσης PoW (Li, et al., 2017).....	28
Εικόνα 9 Διάγραμμα αρχιτεκτονικής ενός κόμβου που εξυπηρετεί διάφορες εφαρμογές blockchain. (IBM, 2018).....	35
Εικόνα 10 Η ροή της συναλλαγής στο Hyperledger	36
Εικόνα 11 Hyperledger Composer Tools-Stack.....	36
Εικόνα 12 Composer Playground.....	38
Εικόνα 13 Αρχιτεκτονική προτεινομένης λύσης.....	41
Εικόνα 14 Διάγραμμα γενικών λειτουργιών συστήματος.....	46
Εικόνα 15 Οθόνη έναρξης πλατφόρμας με το λογότυπο	Error! Bookmark not defined.
Εικόνα 16 Βασική οθόνη φάρμας	Error! Bookmark not defined.
Εικόνα 17 Οθόνη μεταφορέα κατά τη συμπλήρωση φόρμας.....	Error! Bookmark not defined.
Εικόνα 18 Βασική οθόνη Manufacturer.....	Error! Bookmark not defined.
Εικόνα 19 Βασική οθόνη εμπόρου σε mobile συσκευή	Error! Bookmark not defined.
Εικόνα 20 Βασική οθόνη τελικού καταναλωτή.....	Error! Bookmark not defined.

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια, ένα θέμα που προβληματίζει τόσο ακαδημαϊκό όσο και εμπορικά είναι η ασφάλεια των τροφίμων. Η ταχύτητα με την οποία εξελίσσεται η τεχνολογία αλλά και ο γοργός ρυθμός με τον οποίο εισέβαλε στη ζωή του σύγχρονου ανθρώπου, έφερε μία επανάσταση στον τρόπο που αντιλαμβάνεται τον κόσμο. Αποτελεί πλέον αναπόσπαστο κομμάτι της καθημερινότητάς του. Πιο συγκεκριμένα, στον τομέα της εφοδιαστικής αλυσίδας, με την τεχνολογική εξέλιξη, έχουν εφαρμοστεί πολλά συστήματα ιχνηλασιμότητας. Ωστόσο μέχρι σήμερα σχεδόν όλα αυτά τα συστήματα είναι κεντρικοποιημένα και συνήθως αφορούν την κάθε επιχείρηση μεμονωμένα. Αυτό μπορεί να σημαίνει πως είναι αρκετά ευάλωτα ώστε να καταρρεύσουν ακόμα και από ένα μόνο σημείο σφάλματος. Τα συστήματα αυτά είναι μονοπωλιακά, αδιαφανή και έχουν ασύμμετρη πληροφόρηση. Αυτό μπορεί να οδηγήσει σε απάτες, διαφθορές, παραβιάσεις και παραποίηση πληροφοριών. Παράλληλα, όταν απαιτηθεί μια συνολική ιχνηλασιμότητα θα πρέπει ο κάθε κρίκος της εφοδιαστικής αλυσίδας να επικοινωνήσει με τον άλλο για να μπορέσει να διαμοιραστεί τις κεντρικοποιημένες γνώσεις του. Αυτή η χρονοβόρα επικοινωνία μπορεί να επιφέρει κόστος, το οποίο μετακυλιέται στην ασφάλεια των τροφίμων. Σήμερα, μια νέα αποκεντρωμένη τεχνολογία που ονομάζεται blockchain αποτελεί πρωτοποριακή καινοτομία. Τα τελευταία χρόνια το blockchain μπαίνει σε πολλές βιομηχανίες. Η διαχείριση της εφοδιαστικής αλυσίδας έχει αντιληφθεί τη δυναμική του. Η χρήση του, μπορεί να ενεργοποιήσει στη βιομηχανία αυτή, τη διαφάνεια, τον καλύτερο διαμοιρασμό πληροφορίας όπως ακόμη και να ενισχύει την ασφάλεια των τροφίμων.

Στο πλαίσιο αυτής της διπλωματικής ερευνήθηκε ο τρόπος της λειτουργίας του blockchain και των έξυπνων συμβολαίων. Χρησιμοποιήθηκαν τεχνολογίες blockchain, για την ανάπτυξη μιας αποκεντρωμένης πλατφόρμας. Σκοπός της παρούσας εργασίας είναι η παροχή μιας ποιοτικής και αδιάβλητης πληροφορίας στην εφοδιαστική αλυσίδα γαλακτοκομικών προϊόντων, από το σημείο της παραγωγής μέχρι το ράφι του σουπερμάρκετ. Η εφαρμογή δίνει τη δυνατότητα στις συνεργαζόμενες επιχειρήσεις να ανταλλάσσουν πληροφορίες και να παρακολουθούν το ίχνος των προϊόντων που διακινούν. Παράλληλα ο τελικός καταναλωτής, έχει τη γνώση της προέλευσης, της διανομής, των συνθηκών μεταφοράς και της συσκευασίας του προϊόντος. Μέσω της αδιαβλητότητας της πληροφορίας, η αξιοπιστία της ποιότητας του τελικού προϊόντος είναι πραγματική, επομένως η επιλογή του καταναλωτή μπορεί να γίνει σε συνάρτηση ποιότητας-τιμής ή προσωπικών επιλογών του. Τέλος, καταγράφονται και αναλύονται τα συνολικά οφέλη και οι περιορισμοί της τεχνολογίας του blockchain.

Λέξεις κλειδιά: Διαχείριση εφοδιαστικής αλυσίδας, Ιχνηλασιμότητα, Blockchain

ABSTRACT

Over the last few years, an issue that concerns both academic and commercial world, is food safety. The speed at which technology evolves, but also the rapid rhythm that has invaded modern life, revolutionized the way humans perceive the world. It is now an integral part of everyday life. More specifically, in the supply chain, with technological development, many traceability systems have been implemented. However, these days, all these systems are centralized and usually involve each business individually. This might mean that they are vulnerable enough to collapse even from a single point of failure. These systems are opaque, monopolistic and have asymmetric information. This can lead to frauds, corruption, violations and falsification of information. At the same time, when a comprehensive traceability is required, each member in the supply chain must communicate with each other so that it can share its centralized knowledge. This communication is time-consuming and can lead to costs, which are passed on to food safety. Nowadays, a new decentralized technology called blockchain is a pioneering innovation. Recently, blockchain has entered many industries. Supply Chain Management has realized its potential. Its use can trigger transparency, better sharing of information, as well as enhancing food safety in this industry.

The technology of blockchain and smart contracts have been studied during this thesis. Blockchain technologies were used to develop a decentralized platform. The purpose of this paper is to provide quality and immutable information in the dairy supply chain, from the farm to the supermarket shelf. This application enables collaborating companies to exchange information and track the trace of their products. At the same time the final consumer has the knowledge of the provenance, the distribution, the conditions of transport and the packaging of the product. The quality of the final product is credible due to the integrity of information; therefore, the choice of the consumer can be based on economic, quality and personal criteria. Finally, the overall benefits and limitations of the blockchain technology are recorded and analyzed.

Keywords: *Supply chain management, Traceability, Blockchain*

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

Η πρόοδος στην τεχνολογία συνεχίζει να μετασχηματίζει μια ευρεία γκάμα βιομηχανιών, συμπεριλαμβανομένης της τεράστιας βιομηχανίας της εφοδιαστικής αλυσίδας. Οι συνεχώς μεταβαλλόμενες και αυξανόμενες απαιτήσεις των καταναλωτών πλάθουν τη βιομηχανία αυτή σε άμεσα εξελισσόμενη ώστε να μπορεί να ανταποκρίνεται στις ανάγκες τους.

Οι χρήσεις του blockchain σε βάθος χρόνου θα είναι αμέτρητες. Σύμφωνα με προβλέψεις ειδικών σε λιγότερο από 10 χρόνια οι εφαρμογές του blockchain θα εφαρμόζονται σε όλα τα φάσματα της ζωής του σύγχρονου ανθρώπου. Θα είναι εύκολο να γίνεται η συλλογή φόρων από τους πολίτες, οι μεταφορές χρημάτων από οποιοδήποτε μέρος του πλανήτη, θα υλοποιηθούν αυτόνομες επιχειρηματικές λειτουργίες, θα μπορούν άμεσα να ταυτοποιούνται τα άτομα με τα προσωπικά τους στοιχεία διασφαλίζοντας όμως τα προσωπικά τους δεδομένα. Ακόμα και οι οικονομικές απάτες θα μειωθούν καθώς κάθε συναλλαγή θα καταγράφεται σε ένα κατανεμημένο καθολικό. Η γραφειοκρατία θα μειωθεί μέσω της ψηφιοποίησης και αυτοματοποίησης των συστημάτων. Το blockchain είναι μία πηγή εμπιστοσύνης με παγκόσμια εμβέλεια.

Στα μέσα του 2018 παρουσιάστηκε μια πρωτοφανής έκρηξη της δημοτικότητας των τεχνολογιών του blockchain μέσω της αύξησης της ζήτησης του Bitcoin από τα τέλη του 2017 (Fiorillo, 2018). Η αυξημένη ζήτηση του κρυπτονομίσματος εκτίναξε την τιμή αγοράς στα ανώτατα επίπεδα σε βάθος δεκαετίας. Παράλληλα υπήρξε η ολοένα και αυξανόμενη δημιουργία νέων κρυπτονομισμάτων τα ICO (Initial Coin Offering). Αυτό είχε σαν αποτέλεσμα οι κυβερνήσεις να προσπαθήσουν να βρουν λύσεις ώστε να καταφέρουν να ρυθμίσουν τη ροή και τη συγκέντρωση χρήματος ώστε να περιορίσουν τις οικονομικές απάτες, μπαίνοντας έτσι βαθύτερα στο χώρο του blockchain (O'Neal, 2018).

Σήμερα οι τεχνολογίες του blockchain εισάγονται στην καθημερινότητα σε διάφορους τομείς. Μεγάλες βιομηχανίες αυτοματοποιούν συστήματα, παραγωγής, ταυτοποίησης

ποιότητας προϊόντος ακόμη και παροχής υπηρεσίας. Η τεχνολογία blockchain του Hyperledger έχει φτάσει τα 250 μέλη, δείχνοντας το μεγάλο ενδιαφέρον που υπάρχει για τις blockchain εφαρμογές. Εταιρείες κολοσσοί, όπως η Samsung, ενσωματώνουν εφαρμογές πορτοφολιών κρυπτονομισμάτων στα νέα μοντέλα κινητών (theverge.com, 2019), προσπαθώντας να προλάβουν τις εξελίξεις και να γίνουν μέρος αυτών. Άλλο ένα χαρακτηριστικό παράδειγμα είναι, η Walmart, η μεγαλύτερη εταιρεία supermarket παγκοσμίως που ενσωμάτωσε στο δίκτυό της, τεχνολογίες blockchain με την εφαρμογή FoodTrust. Η υιοθέτηση αυτών των τεχνολογιών παρατηρείται από πλευράς επιχειρήσεων στην εφοδιαστική αλυσίδα, στα χρηματοοικονομικά, στον τομέα υγείας αλλά και σε κυβερνητικό επίπεδο.

Συνοπτικά υπάρχουν δύο βασικοί τύποι blockchain:

- Τα δημόσια (permissionless) στα οποία ο καθένας μπορεί να διαβάσει και να γράψει. Είναι ανοιχτού κώδικα και όλοι μπορούν να λάβουν μέρος σε αυτά, να εξερευνήσουν και να στείλουν συναλλαγές ή συμβάσεις. Παραδείγματα δημοσίων blockchain είναι το Bitcoin και το Ethereum. Τα ιδιωτικά blockchain (permissioned) στα οποία οι συμμετέχοντες είναι γνωστοί και αξιόπιστοι εκ των προτέρων.
- Τα δικαιώματα και η εγγραφή διατηρούνται κεντρικά σε έναν οργανισμό, ενώ για ανάγνωση είναι με περιορισμένη έκταση.

Στην εφοδιαστική αλυσίδα για την διασφάλιση της καλής ποιότητας του προϊόντος και της ασφαλούς κατανάλωσής του, πληρώντας τις απαραίτητες προϋποθέσεις υγιεινής, είναι χρήσιμο να εφαρμοστούν permissioned blockchain εφαρμογές. Οι εφαρμογές αυτές, θα φέρουν το μέγιστο δυνατό αποτέλεσμα σε κάθε στάδιο μέχρι το τελικό προϊόν να φτάσει στο ράφι του καταναλωτή.

Ωστόσο υπάρχουν και προκλήσεις από την χρήση της τεχνολογίας αυτής. Στα δημόσια δίκτυα το βασικότερο εμπόδιο για την ευρεία υιοθέτηση του blockchain είναι το κόστος της ενέργειας που απαιτείται. Ο όγκος των συναλλαγών και η αποθήκευση των δεδομένων είναι βασικές ανησυχίες για την επεκτασιμότητα του blockchain. Ο κίνδυνος από κυβερνοεπιθέσεις και το θεσμικό πλαίσιο στο οποίο λειτουργεί είναι δύο ακόμα

παράγοντες που πρέπει να συνυπολογιστούν. Μια ακόμη μεγάλη πρόκληση για την υλοποίηση τέτοιων τεχνολογιών σε βιομηχανικό επίπεδο είναι η εναρμόνιση των διαφορετικών συμβαλλομένων, οι οποίοι έχουν πολύπλοκες και διαφορετικές διαδικασίες, σε μια ενιαία πατροναρισμένη ροή, η οποία απαιτεί την σωστή κατανόηση όλων των προηγούμενων σταδίων.

Παρά το γεγονός ότι η σταδιακή ανάπτυξη της ποιότητας των μεταποιημένων προϊόντων τα τελευταία χρόνια έχει δημιουργήσει μια ευκαιρία για την υλοποίηση ασφαλούς και καλής ποιότητας παραγωγής με βάση εναρμονισμένους νόμους της ΕΕ - οι καταναλωτές έχουν όλο και περισσότερες ανησυχίες και δίνουν μεγαλύτερη προσοχή στην ποιότητα των προϊόντων. Ωστόσο, η καλή ποιότητα των τροφίμων μπορεί να παραχθεί μόνο από υλικά καλής ποιότητας στα οποία γίνεται σωστή μεταχείριση σε όλα τα στάδια της εφοδιαστικής αλυσίδας.

1.1 Κίνητρα και στόχοι της διπλωματικής

Στην παρούσα διπλωματική πρωταγωνιστικό ρόλο λαμβάνει η τεχνολογία του blockchain η οποία έρχεται να επιλύσει προβλήματα της εφοδιαστικής αλυσίδας. Προσεγγίζεται καθολικά και αντικειμενικά, από τους συμμετέχοντες της εφοδιαστικής μέχρι τον καταναλωτή.

Η ψηφιακή πρόοδος όπως τα μεγάλα δεδομένα και τα εργαλεία βελτιστοποίησης παραμένουν οι πιο εμφανείς παράγοντες αλλαγής στην αλυσίδα εφοδιασμού. Παράλληλα η τεχνολογία blockchain επαναπροσδιορίζει την συλλογή και επεξεργασία των μεγάλων δεδομένων. Οι θεμελιώδεις μεταβολές στον κλάδο επηρεάζουν σημαντικά τον ρυθμό αλλαγών. Η υπερπληθώρα προσφερόμενων προϊόντων σε παγκόσμιο επίπεδο και η ενοποίηση των αγορών τροφίμων μεταξύ κρατών δημιουργεί την ανάγκη για διασφάλιση της ποιότητας και την ασφάλεια κατανάλωσής του. Η προέλευση των προϊόντων στην εφοδιαστική αλυσίδα είναι σημαντική σήμερα, καθώς βάση στατιστικών, έρευνας που πραγματοποιήθηκε σε 600 εκατομμύρια ανθρώπους, ένας στους δέκα παγκοσμίως έχει αρρωστήσει μετά από κατανάλωση μολυσμένου τροφίμου. Ετησίως 420 χιλιάδες από

αυτούς πεθαίνουν (Gutierrez, 2017). Το συνολικό κόστος που προκαλείται από την τροφική ασθένεια, έχει επιπτώσεις σε οικονομικό επίπεδο με αισθητή μείωση της παραγωγικότητα λόγω ανικανότητας προς εργασία ή και θανάτου. Επίσης έχει επιπτώσεις στην αύξηση των ιατρικών δαπανών για αντιμετώπιση των καταστάσεων. Μόνο στις ΗΠΑ ετησίως η δαπάνη αγγίζει τα \$55.5 έως \$93.2 δισεκατομμύρια (Galvin, 2018). Με 20% αύξηση την τετραετία 2014-2018. Στην παγκόσμια ιχνηλασιμότητα τροφίμων υπολογίζεται ότι έχουν επενδυθεί \$11.25 δισεκατομμύρια το 2017 και αναμένεται να αγγίξει τα \$20.95 δισεκατομμύρια μέχρι το 2026. Το σύνθετο ετήσιο ποσοστό ανάπτυξης (CAGR) θα αυξηθεί κατά 7.15% για την προαναφερθείσα περίοδο πρόβλεψης (Newswire, 2018). Στις αναδυόμενες αγορές υπάρχει μεγαλύτερος ρυθμός του ετήσιου ποσοστού ανάπτυξης, όπως για παράδειγμα στην Κίνα αγγίζοντας το ποσοστό 18.9% μέχρι το 2019. Ακόμα ένα πρόβλημα στην τροφική αλυσίδα είναι οι απάτες τροφίμων. Υπολογίζεται ετησίως το κόστος στη βιομηχανία να αγγίζει παγκοσμίως τα \$10-15 δισεκατομμύρια (IFIS, 2016). Άλλο ένα βασικό πρόβλημα που εμφανίστηκε στην τροφική αλυσίδα το 2006 ήταν η εμφάνιση του μικροβίου E. coli στο συσκευασμένο σπανάκι στις αγορές της Αμερικής (Galvin, 2018). Το διανεμημένο σπανάκι σε όλες τις πολιτείες της Αμερικής έπρεπε να ανακληθεί καθώς είχε επηρεάσει 26 από αυτές. Υπήρξαν 199 περιπτώσεις ατόμων. Από αυτούς οι 109 νοσηλεύτηκαν και οι 31 υπέφεραν από νεφρική ανεπάρκεια που ονομάζεται σύνδρομο hemolytic-uremic. Υπήρξαν τρεις επιβεβαιωμένοι θάνατοι. Καταστροφικές ήταν οι συνέπειες σε ολόκληρη την Αμερικάνικη αγορά σπανακιού η οποία έκανε να ανακάμψει 4 χρόνια. Μετά από χρονοβόρες έρευνες που διενεργήθηκαν, κατέληξαν πως το κρούσμα προήλθε από έναν προμηθευτή, μιας συγκεκριμένης παρτίδας, μίας ημέρας.

Η αυξανόμενη ανησυχία των καταναλωτών για την ασφάλεια των τροφίμων, η υψηλή ζήτηση στις αναδυόμενες οικονομίες, οι αυστηροί εξαγωγικοί κανονισμοί και η αυξανόμενη διεύθυνση των σύγχρονων λιανικών πωλήσεων, όπως τα σούπερ μάρκετ και οι υπεραγορές, αποτελούν μερικούς από τους βασικούς παράγοντες που οδηγούν στην ανάπτυξη της αγοράς. Εντούτοις, οι παράγοντες όπως το σχετικό κόστος, το πρόσθετο κόστος για τα συστήματα ιχνηλασιμότητας, οι διαφορετικές ανάγκες για διαφορετικά

προϊόντα, η έλλειψη αυστηρών νόμων στις αναπτυσσόμενες χώρες και τα ζητήματα ιδιωτικού απορρήτου για την ανταλλαγή δεδομένων εμποδίζουν την ανάπτυξη της αγοράς.

Τα γαλακτοκομικά προϊόντα αποτελούν αναπόσπαστο κομμάτι της καθημερινής διατροφής του Έλληνα. Για να καλυφθούν οι ημερήσιες ανάγκες σε ασβέστιο και βιταμίνη D, πρέπει να καταναλώνονται συνολικά τουλάχιστον 1-2 μερίδες γαλακτοκομικών. Η εξέλιξη πωλήσεων των γαλακτοκομικών στην Ελλάδα είχε ορισμένες διακυμάνσεις, αλλά από το 2016 και έπειτα ακολουθεί μια σαφή αύξηση των πωλήσεων με αύξοντα ρυθμό (Μανιφάβα, 2018). Ειδικότερα, σύμφωνα με τα στοιχεία της εταιρείας ερευνών αγοράς IRI (βλ. **Εικόνα 1**) το πρώτο εξάμηνο του 2018 καταγράφηκε αύξηση, σε σύγκριση με το πρώτο εξάμηνο του 2017, των πωλήσεων γαλακτοκομικών προϊόντων στα σούπερ μάρκετ κατά 4,4% σε αξία και κατά 1,9% σε όγκο.



Εικόνα 1 Εξέλιξη πωλήσεων γαλακτοκομικών (Μανιφάβα, 2018)

Ειδικότερα, σύμφωνα με τη Nielsen, το πρώτο τετράμηνο του 2018 σε σχέση με του 2017, διαπιστώθηκε αύξηση των πωλήσεων σε όγκο γαλακτοκομικών προϊόντων, στο παγωτό

στα 10.051,1 χιλ. κιλά από 7.346,7 χιλ. κιλά, στα επιδόρπια γιαουρτιού στα 1.086 χιλ. κιλά από 953,5 χιλ. κιλά και στο γιαούρτι στα 17.076,5 χιλ. κιλά από 16.093,2 χιλ. κιλά (Σκούφου, 2018). Σημειώνεται επίσης, πως το πρώτο τετράμηνο του 2018 έναντι του 2017, είχε συνολικά θετικό πρόσημο των πωλήσεων γάλακτος σε αξία, καθώς διαμορφώθηκαν στα 104,28 εκατ. ευρώ από 101,78 εκατ. ευρώ, μια αύξηση της τάξης του 2,5%.

Ο βασικός στόχος είναι ο σχεδιασμός μιας αποκεντρωμένης εφαρμογής η οποία θα βοηθήσει στην απλοποίηση της διαχείρισης της εφοδιαστικής αλυσίδας γαλακτοκομικών προϊόντων, η οποία είναι σε ανοδικό ρυθμό τα τελευταία χρόνια. Οι επιχειρήσεις θα είναι σε θέση να ενοποιήσουν αποτελεσματικά τις κοινές πληροφορίες για το γάλα, τις πληροφορίες αποθεμάτων τους, τις ψηφιακές συμβάσεις, τις πληροφορίες ποιότητας του προϊόντος, τις τιμολογιακές πολιτικές ακόμα και την ταχύτητα στην ανάκληση κάποιου ελαττωματικού προϊόντος.

Επιγραμματικά η υλοποίηση αυτής της εφαρμογής στοχεύει:

- Να παρέχει ασφαλή και καλής ποιότητας προϊόντα από τα οποία ο τελικός καταναλωτής αντλεί μόνος του τις πληροφορίες. Δηλαδή η επέκταση της αλυσίδας εφοδιασμού στον χώρο κατανάλωσης των πελατών.
- Διαφάνεια των διαδικασιών σε όλα τα επίπεδα της εφοδιαστικής αλυσίδας γαλακτοκομικών προϊόντων, καθώς όλες οι συναλλαγές είναι καταγεγραμμένες στο μητρώο και είναι προσβάσιμες ανά πάσα στιγμή.
- Γρηγορότερη ιχνηλασιμότητα και ανάκληση προϊόντων. Βελτίωση συντονισμού μεταξύ συνεργατών.
- Μείωση ποσοτήτων καταστροφής γαλακτοκομικών από λανθασμένες ανακλήσεις. Μείωση οικονομικών καταστροφών.
- Αποκεντρωμένη/διανεμημένη τήρηση του μητρώου των δεδομένων από τις εμπλεκόμενες επιχειρήσεις. Η ανταλλαγή δεδομένων αποτελεί βασικό στοιχείο για την καταπολέμηση της απάτης στα τρόφιμα.

- Ασφάλεια των πληροφοριών καθώς είναι εξαιρετικά δύσκολο να τροποποιηθούν ήδη καταχωρημένες συναλλαγές.
- Μετάβαση από μεμονωμένες γραμμικές αλυσίδες εφοδιασμού γαλακτοκομικών προϊόντων σε ένα αποκεντρωμένο οικοσύστημα εφοδιαστικής.
- Μετάβαση από περιβάλλον ανταγωνισμού σε περιβάλλον συνεργασίας και συν-δημιουργίας. Εμπιστοσύνη μεταξύ των χρηστών οι οποίοι είναι σε θέση ανά πάσα στιγμή να επαληθεύσουν οποιαδήποτε καταχώρηση ή συναλλαγή στο δίκτυο.
- Καινοτομία με την χρήση νέων τεχνολογιών.
- Μείωση κόστους εργασίας.

Καταλήγοντας, η τεχνολογία του blockchain είναι μία αποκεντρωμένη και διανεμημένη βάση δεδομένων, όπου οι καταχωρήσεις κρυπτογραφούνται και επιβεβαιώνονται δημιουργώντας μία αλληλουχία με της προηγούμενες καταχωρήσεις. Κοινή συνισταμένη των χαρακτηριστικών αυτών είναι η αποκατάσταση της εμπιστοσύνης, η οποία πλέον εκφεύγει από τις κεντρικές αρχές, των εμπλεκομένων, ως αρμοδιότητα και διαμοιράζεται μεταξύ τους.

1.2 Διάρθρωση της Εργασίας

Η παρούσα μεταπτυχιακή εργασία θα αναπτυχθεί σε έξι συνολικά κεφάλαια. Η διάρθρωση της εργασίας είναι η εξής:

1. Στο **Κεφάλαιο 2** δίνονται οι ορισμοί της εφοδιαστικής αλυσίδας, της ιχνηλασιμότητας των τροφίμων και της τεχνολογίας του blockchain. Παράλληλα αναπτύσσονται οι προεκτάσεις τους και πραγματοποιείται η ανασκόπηση της διεθνούς βιβλιογραφίας.
2. Στο **Κεφάλαιο 3** γίνεται εκτεταμένη αναφορά στο τεχνολογικό πλαίσιο που χρησιμοποιήθηκε για την υλοποίηση της πλατφόρμας. Των τεχνολογιών blockchain και των παράλληλων εργαλείων που χρησιμοποιήθηκαν.

3. Στο **Κεφάλαιο 4** αναλύθηκε το σύστημα αποκεντρωμένης ιχνηλασιμότητας στη βιομηχανία γάλακτος με τους στόχους, την προσέγγιση και το μοντέλο.
4. Στο **Κεφάλαιο 5** πραγματοποιείται η υλοποίηση του συστήματος.
5. Τέλος, στο **Κεφάλαιο 6** διατυπώνονται επιγραμματικά τα συμπεράσματα που προκύπτουν από τη μελέτη, οι μελλοντικές εφαρμογές καθώς και οι περιορισμοί.

Στο τέλος της εργασίας παρατίθεται **παράρτημα** με:

- τα βήματα εγκατάστασης και ανάπτυξης του κώδικα
- τον κώδικα έξυπνων συμβολαίων, ειδικών ερωτημάτων, δικαιωμάτων και σημείων της web εφαρμογής
- και μια σύντομη επίδειξη της λειτουργικότητας της πλατφόρμας πάνω στην τεχνολογία του blockchain.

ΚΕΦΑΛΑΙΟ 2: Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό διατυπώνεται το θεωρητικό υπόβαθρο σχετικά με την εφοδιαστική αλυσίδα. Αναλύεται η σημαντικότητα της ιχνηλασιμότητας και οι προσεγγίσεις που υπάρχουν γι' αυτή. Η σύνδεσή της με τη διασφάλιση ποιότητας ιδίως στα ευπαθή προϊόντα, όπως τα γαλακτοκομικά. Επίσης, παρουσιάζονται οι τεχνολογίες του blockchain μαζί με συστατικά που το αποτελούν. Τέλος, γίνεται μια διερεύνηση και σύνδεση με blockchain εφαρμογές που έχουν υλοποιηθεί και αγγίζουν την εφοδιαστική αλυσίδα.

2.1 Εφοδιαστική αλυσίδα

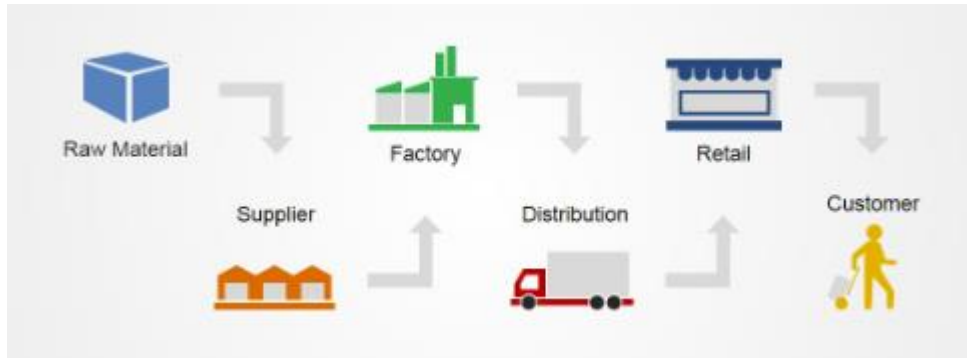
2.1.1 Διαχείριση εφοδιαστικής αλυσίδας

Η διαχείριση της εφοδιαστικής αλυσίδα περιλαμβάνει όλες εκείνες τις δραστηριότητες και τις διαδικασίες που σχετίζονται με τον σχεδιασμό, την παραγωγή και τις υπηρεσίες προϊόντων μέχρι τον τελικό χρήστη. Αυτό περιλαμβάνει τον προγραμματισμό της παραγωγής, τις προμήθειες των πρώτων υλών, την συλλογή, την διανομή και μεταφορά, την μεταποίηση και επεξεργασία, την αποθήκευση και διαχείριση αποθέματος και την εξυπηρέτηση των πελατών. Στην πραγματικότητα είναι συνδεδεμένα δίκτυα εταιρειών και συνεργατών με σκοπό την ροή των υλικών και τον πληροφοριών από τους παραγωγούς και τους προμηθευτές στον καταναλωτή (Larson & Rogers, 1998).

2.1.2 Η ροή της εφοδιαστικής αλυσίδας

Υπάρχουν αρκετά βήματα μέσα στην αλυσίδα για να φτάσει ένα προϊόν από την παραγωγή της πρώτης ύλης στο τραπέζι του καταναλωτή. Οι παραγωγοί που είναι οι αγρότες είτε καλλιεργούν, είτε εκτρέφουν τις πρώτες ύλες. Οι βιομηχανικοί παραγωγοί παίρνουν από τους προμηθευτές ή απευθείας τις ύλες και δημιουργούν μεταποιημένα και

συσκευασμένα προϊόντα. Οι χονδρέμποροι και οι διανομείς αναλαμβάνουν να μεταφέρουν τα προϊόντα στους εμπόρους για να τα πουλήσουν με την σειρά τους στους καταναλωτές (βλ. **Εικόνα 2**) (Fredriksson & Liljestrand, 2015).



Εικόνα 2 Ροή της εφοδιαστικής αλυσίδας

2.1.3 Γαλακτοκομικά προϊόντα

Η εφοδιαστική αλυσίδα των τροφίμων κρίνεται ιδιαίτερα κρίσιμη και επιτάσσει αυστηρές διαδικασίες όταν πρόκειται για ευπαθή προϊόντα. Στον τομέα των γαλακτοκομικών αν δεν ακολουθηθούν σωστές διαδικασίες ενδέχεται τα αγαθά να φθαρούν, να αλλοιωθούν ή να μην είναι ασφαλή προς κατανάλωση. Γι' αυτό το λόγο έχουν κάποιες ιδιαιτερότητες στην ροή της αλυσίδας. Αρχικά οι παραγωγοί συγκεντρώνουν το γάλα σε σταθμούς συγκέντρωσης και ψύξεως, τις παγολεκάνες. Στην συνέχεια γίνεται η μεταφορά τους με ανοξείδωτα βυτία, τα οποία περιέχουν διαμερίσματα διαχωρισμού γάλακτος. Η μεταφορά τους γίνεται σε θερμοκρασία περίπου 4-6 βαθμών. Κατά την παραλαβή από την βιομηχανία και τηρώντας όλες τις προδιαγραφές υγιεινής και ασφάλειας του γάλακτος, με αντλίες εισέρχεται στο εσωτερικό της μονάδας και αποθηκεύεται προσωρινά σε δεξαμενές (Ανοπ., χ.χ.) (ΕΦΕΤ, 2012). Σε αυτό το στάδιο είναι απαραίτητος ο ποιοτικός έλεγχος, ο οποίος περιλαμβάνει τα εξής:

- Έλεγχος θερμοκρασίας
- Έλεγχος οξύτητας

- Έλεγχος αντιβιοτικών
- Περικτικότητα λίπους, πρωτεϊνών, λακτόζης, στερεού υπολείμματος και στερεού υπολείμματος άνευ λίπους.

Τέλος, ακολουθεί η επεξεργασία και η τυποποίηση του προϊόντος.

2.2 Ιχνηλασιμότητα τροφίμων

Η ιχνηλασιμότητα (traceability) των τροφίμων ορίζεται ως η δυνατότητα παρακολούθησης (track) και ανίχνευσης της προέλευσης (trace) κατά τα στάδια παραγωγής, μεταποίησης και διανομής τους (Anon., 2005). Τα στάδια είναι προκαθορισμένα και ακολουθούν συγκεκριμένους κανόνες οι οποίοι ορίζονται από το είδος του προϊόντος. Ουσιαστικά είναι ο τρόπος να βλέπεις την πραγματική διαδρομή του προϊόντος μέσω της ιστορίας του και του «ίχνους» που αυτό αφήνει κατά την εναλλαγή των σταδίων στην εφοδιαστική αλυσίδα.

2.2.1 Εισαγωγή Ιχνηλασιμότητας

Βάση της θεωρίας των Aung και Chang, 2014 (Aung & Chang, 2014) η ιχνηλασιμότητα είναι άμεσα συνυφασμένη με το υποκείμενο προϊόν κατά μήκος της εφοδιαστικής αλυσίδας και απαντά στα ερωτήματα πότε, που, πως, τι και γιατί. Το σωστό σύστημα ιχνηλασιμότητας, πρέπει να μας δίνει απαντήσεις στην επίλυση των ζητημάτων που αφορούν στη διαχείριση των κινδύνων ποιότητας των τροφίμων αλλά και να είναι αποτελεσματικό στη διαχείριση της αλυσίδας εφοδιασμού (Manzini & Accorsi, 2013).

Η ιχνηλασιμότητα στη βιομηχανία τροφίμων εφαρμόζεται κατά βάση σε δύο κατηγορίες:

- i. της προόδου, η οποία είναι η σχέση προϊόντος – διαδικασίας και
- ii. της τοποθεσίας, η οποία είναι η σχέση προϊόντος – εντοπισμού.

Η ιχνηλασιμότητα ουσιαστικά περιγράφει τον συνδυασμό της ροής των συστατικών και της πληροφορίας (Food Marketing Research and Information Center, 2008).

Η βασική υπεροχή της εφαρμογής του συστήματος ιχνηλασιμότητας στην εφοδιαστική αλυσίδα είναι πως δίνει τη δυνατότητα στον τελικό καταναλωτή να έχει στη διάθεση του όλες τις απαραίτητες πληροφορίες για την προέλευση, την τοποθεσία, τη συλλογή, τη διανομή και το γενικό ιστορικό της ζωής του προϊόντος. Αυτό παρέχει μια εγγύηση και ασφάλεια στην ποιότητα του προϊόντος κατά την διαδρομή του (Bechini, et al., 2007). Σύμφωνα με την έρευνα των Panta et al. (Pant, et al., 2015), οι καταναλωτές των προϊόντων διατροφής, τα διαχωρίζουν σε τέσσερις κατηγορίες ελέγχου. Τα αισθητήρια χαρακτηριστικά, τα χαρακτηριστικά υγείας, τα χαρακτηριστικά ευκολίας και τα χαρακτηριστικά διεργασίας.

Για να υπάρχει ολοκληρωμένη ιχνηλασιμότητα στα προϊόντα διατροφής, πρέπει να είναι δυνατή η πληροφόρηση σε πρωτογενές επίπεδο. Για παράδειγμα στα γεωργικά προϊόντα η γνώση για την διαδικασία των σταδίων παραγωγής που αφορά τη χρήση λιπασμάτων, φαρμάκων, την ποιότητα του νερού, τη συχνότητα των ποτισμάτων κ.α.. Σε μία άλλη κατηγορία προϊόντων, που είναι πιο ευπαθή, όπως τα γαλακτοκομικά η ανάγκη για πληροφόρηση είναι ακόμα μεγαλύτερη καθώς οι συνθήκες διανομής, συλλογής, αποθήκευσης και τελικής τυποποίησης είναι ακόμη πιο απαιτητικές και καθορίζουν ουσιαστικά την ποιότητα του τελικού προϊόντος. Μια άλλη κατηγορία, αυτή του φρέσκου γάλακτος έχει να κάνει με την ποσότητα συγκέντρωσης λιπαρών τη συγκέντρωση αντιβιοτικών αλλά και τη σωστή θερμοκρασία συλλογής και διανομής του.

2.2.2 Είδη Ιχνηλασιμότητας

Στην εποχή μας ο παράγοντας που αφορά την ακριβή και έγκαιρη ιχνηλασιμότητα των προϊόντων και των δραστηριοτήτων στην αλυσίδα εφοδιασμού αποτελεί προϋπόθεση. Με βάση το άρθρο του Oraga (Oraga, 2003), για να επιτευχθεί ένα ολοκληρωμένο σύστημα ιχνηλασιμότητας στα τρόφιμα υπάρχουν έξι παράγοντες.

i. **Ιχνηλασιμότητα προϊόντος** – η οποία καθορίζει τη φυσική θέση ενός προϊόντος σε οποιοδήποτε στάδιο της αλυσίδας εφοδιασμού προκειμένου να διευκολυνθεί στη

διαχείριση των αποθεμάτων, την ανάκληση των προϊόντων και τη μετάδοση της πληροφορίας στους τελικούς καταναλωτές ακόμη και σε άλλα ενδιαφερόμενα μέρη.

ii. **Ιχνηλασιμότητα διαδικασίας** – η οποία εξακριβώνει τον τύπο και την σειρά των δραστηριοτήτων που επηρέασαν το προϊόν κατά την παραγωγή του. Αυτές μπορούν να αφορούν και τις αλληλεπιδράσεις μεταξύ του προϊόντος και των φυσικών / μηχανικών / χημικών / περιβαλλοντικών παραγόντων. Εδώ περιλαμβάνεται και η απουσία ή παρουσία προσμείξεων.

iii. **Γενετική ιχνηλασιμότητα** – η οποία καθορίζει τη γενετική σύσταση του προϊόντος. Αυτή περιλαμβάνει πληροφορίες σχετικά με τον τύπο και την προέλευση (πηγή, προμηθευτές) γενετικών τροποποιημένων οργανισμών / υλικών ή συστατικών καθώς και πληροφορίες σχετικά με τα υλικά φύτευσης (σπόροι, μοσχεύματα, στελέχη, κόνδυλοι κ.α.) που χρησιμοποιούνται για τη δημιουργία του ακατέργαστου προϊόντος.

iv. **Ιχνηλασιμότητα εισροών** – η οποία προσδιορίζει τον τύπο και την προέλευση (πηγή, προμηθευτές) εισροών όπως λιπάσματα, χημικά σπρέι, ύδατα άρδευσης, ζώα, ζωοτροφές και την παρουσία προσθέτων και χημικών ουσιών που χρησιμοποιούνται για τη διατήρηση ή κατεργασία του αρχικού υλικού σε μεταποιημένα προϊόντα διατροφής.

v. **Ιχνηλασιμότητα ασθενειών και επιβλαβών οργανισμών** – η οποία εντοπίζει την επιδημιολογία των παρασίτων και τους κινδύνους όπως τα βακτήρια, τους ιούς και άλλα αναδυόμενα παθογόνα που μπορεί να μολύνουν τα τρόφιμα.

vi. **Ιχνηλασιμότητα μέτρησης** – η οποία σχετίζεται με αποτελέσματα μέτρησης μέσω μιας αδιάσπαστης αλυσίδας βαθμονόμησης με βάση συγκεκριμένα αποδεκτά πρότυπα.

Εναλλακτικά ανάλογα την κατηγοριοποίηση των στοιχείων και των πληροφοριών διακρίνεται μια μάρκο προσέγγιση της ιχνηλασιμότητας (Zhang & Bhatt, 2014) σε:

i. **Εξωτερική ιχνηλασιμότητα** – η οποία εντοπίζει όλα τα ανιχνεύσιμα στοιχεία και προϋποθέτει το διαμοιρασμό της πληροφορίας σε όλους τους συμμετέχοντες στο κανάλι διανομής. Αυτό μπορεί να επιτευχθεί με την κοινοποίηση στους συνεργάτες της

εφοδιαστικής αλυσίδας είτε με κάποιο μοναδικό αριθμό ταυτοποίησης ανά προϊόν, είτε με αριθμό ανά παρτίδα (π.χ. lot no).

ii. **Εσωτερική ιχνηλασιμότητα** – η οποία αφορά όλες τις διαδικασίες που γίνονται εντός μιας επιχείρησης για να συνδέσει τις ταυτότητες των πρώτων υλών και τις ταυτότητες των τελικών προϊόντων ανά γραμμή παραγωγής.

iii. **Ιχνηλασιμότητα εφοδιαστικής αλυσίδας** – η οποία δίνει έμφαση στην σύνδεση των πληροφοριών από το ένα άκρο της αλυσίδας στο άλλο. Δηλαδή σε όλα τα στάδια παραγωγής, επεξεργασίας και διανομής του προϊόντος. Οι πληροφορίες αυτές περιλαμβάνουν έγγραφα που αποδεικνύουν την πηγή των πρώτων υλών, αρχεία που καταρτίστηκαν κατά τη διάρκεια της τεχνολογικής διαδικασίας, εντοπισμός συστατικών και πρόσθετων υλών, τους τελικούς εμπόρους κ.α.

2.2.3 Συστήματα Ιχνηλασιμότητας

Ένα σύστημα ιχνηλασιμότητας περιλαμβάνει τους μηχανισμούς που αφορούν την ταυτοποίηση, τη σύνδεση, την καταγραφή, την αποθήκευση, την επαλήθευση και την προσπέλαση της πληροφορίας. Το σύστημα αποτελείται από κανόνες, διαδικασίες και τεκμηριώσεις και μπορεί να απαιτεί πιθανών υλικοτεχνικό εξοπλισμό και ανθρώπινους πόρους.

Υπάρχουν δύο βασική τύποι συστημάτων:

- **Προς τα εμπρός** (downstream) ιχνηλασιμότητα: ξεκινάει από μία συγκεκριμένη παρτίδα πρώτης ύλης (lot), με σκοπό να φτάσει στον εντοπισμό όλων των παρτίδων τελικών προϊόντων που παρήχθησαν.

- **Προς τα πίσω** (upstream) ιχνηλασιμότητα: ξεκινάει γνωρίζοντας την παρτίδα του τελικού προϊόντος (lot), με σκοπό να ανιχνεύσει τις πρώτες ύλες που χρησιμοποιήθηκαν για την παραγωγή της.

Στην εποχή μας τα συστήματα ιχνηλασιμότητας, αν και δεν είναι απαραίτητο, χρησιμοποιούν τις τεχνολογίες πληροφοριακών συστημάτων για να εκμεταλλευτούν δεδομένα που προϋπάρχουν ή είναι μέρος αυτού (Food Marketing Research and Information Center, 2008).

2.2.4 Εφαρμογή ιχνηλασιμότητας και τεχνολογίες

Το πρώτο βήμα για την ιχνηλασιμότητα είναι η τυποποίηση και η χρήση διαδικασιών. Διαδικασίες όπως έκδοση παραστατικών, καταγραφή παραλαβών, διαχείριση αποθεμάτων έχουν μια σειρά από πληροφορίες που μπορούν να οδηγήσουν σε αυτή. Υπάρχουν διαφορετικές εφαρμογές αξιοποίησης και συγκέντρωσης πληροφοριών. Πολλοί ερευνητές βλέπουν εφαρμογές που προσεγγίζουν ή πετυχαίνουν την ιχνηλασιμότητα με χρήση παράλληλων εργαλείων όπως τα internet of things, χρήση RFID (Radio-Frequency Identification), WSN (Wireless Sensor Network), GPS (Global Positioning System) και GIS (Geographic Information System), κ.λπ., στην διαχείριση της εφοδιαστικής αλυσίδας (Sari, 2010).

Υπάρχουν παγκόσμιοι και ανά χώρα κανονισμοί που αφορούν την ιχνηλασιμότητα και την εφαρμογή της, όπως το έγγραφο Codex που αναφέρεται σε ένα σύνολο αρχών και κανονισμός που πρέπει να ακολουθηθούν σχετικά με αυτή (International Union of FOOD SCIENCE and Technology, 2012). Επίσης μια σειρά από πρότυπα βοηθούν στην εκτέλεση τέτοιων συστημάτων. Το Global Traceability Standard (GTS), μέσω του μη κερδοσκοπικού οργανισμού GS1, συμβάλει σε πάνω από 100 χώρες να δουλεύουν συστήματα ιχνηλασιμότητας. Το ISO 22005:2007 είναι άλλο ένα πρότυπο για ιχνηλασιμότητα τροφίμου βασισμένο στις γενικές αρχές υλοποίησης και σχεδιασμού συστημάτων. Η πρωτοβουλία Produce Traceability Initiative (PTI) που υποστηρίζεται από τον Καναδική CPMA, τον GS1 και άλλους οργανισμούς. Πολλές φορές γίνεται λόγος πως η υιοθέτηση κάποιων προτύπων βοηθάει στην βελτίωση των εσωτερικών διαδικασιών μιας επιχείρησης και στην αύξηση της αποδοτικότητας στο σύνολο της εφοδιαστικής αλυσίδας.

2.2.5 Διασφάλιση ποιότητας

Υπάρχει αυξημένη απαίτηση από τους καταναλωτές για υψηλή ποιότητα και ασφάλεια στα τρόφιμα. Έχουν υλοποιηθεί διάφορα συστήματα διαχείρισης όπως το ISO 9000, HACCP, TQM (El-Hofi, et al., 2010). Η ανάλυση κινδύνου και τα κρίσιμα σημεία ελέγχου (HACCP) είναι η πιο γνωστή εφαρμογή διασφάλισης ποιότητας και έχει εφαρμοστεί αποτελεσματικά στα τρόφιμα. Χρησιμοποιείται για την πλήρη ανάλυση των διαφόρων πιθανών παραγόντων κινδύνου που θα μπορούσαν να προκαλέσουν μόλυνση σε ολόκληρη την αλυσίδα εφοδιασμού γαλακτοκομικών προϊόντων. Το σύστημα HACCP αναπτύχθηκε αρχικά ως μικροβιολογικό σύστημα για να εξασφαλίσει την ασφάλεια των τροφίμων για αστροναύτες από τη NASA στη δεκαετία του 1960. Ακολουθώντας τέτοια συστήματα αλλάζουν τον τρόπο παραγωγής μετακίνησης και τυποποίησης των προϊόντων (El-Hofi, et al., 2010). Παράλληλα συμβάλουν στην καταγραφή του ίχνους των προϊόντων και περιλαμβάνουν διαδικασίες ανάκλησης. Στην Ευρωπαϊκή Ένωση ισχύει και το Σύστημα Έγκαιρης Προειδοποίησης για τα τρόφιμα και τις ζωοτροφές (RASFF) (European Commission, n.d.), που είναι ένα εργαλείο διαμοιρασμού πληροφόρησης μεταξύ κεντρικών αρμόδιων αρχών για τους κανονισμούς στα τρόφιμα σε περιπτώσεις κινδύνων και λήψης μέτρων για την ανθρώπινη υγεία. Βάση της γραφειοκρατίας, των ελέγχων και της πολυπλοκότητας των συστημάτων αυτών υπάρχει επιτακτική ανάγκη για καλύτερη και πιο οργανωμένη διαχείριση της πληροφορίας όπου είναι εφικτό.

2.3 Blockchain

2.3.1 Τι είναι το Blockchain

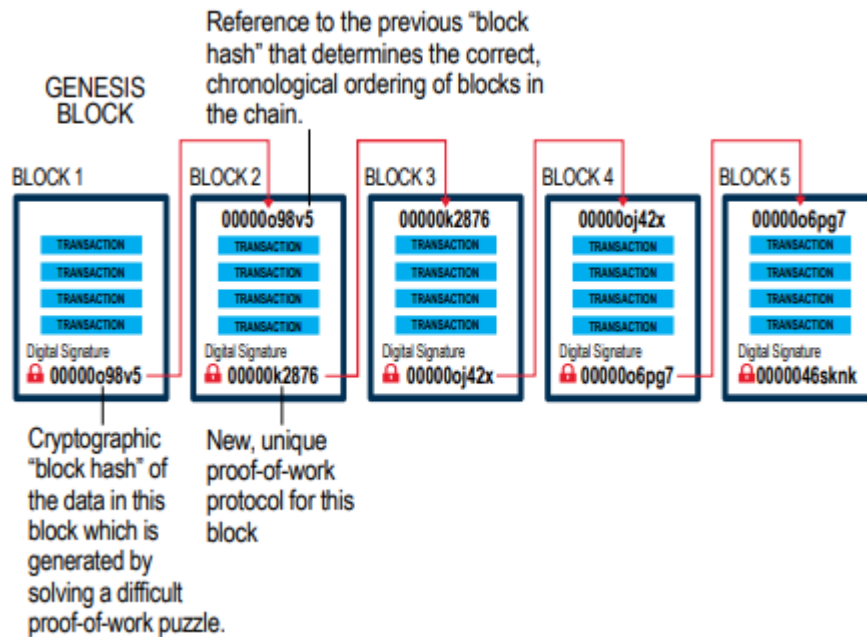
Η δομή δεδομένων blockchain είναι μια ταξινομημένη και αυξανόμενη λίστα συναλλαγών. Οι εγγραφές ονομάζονται μπλοκ και συνδέονται μεταξύ τους με κρυπτογραφία (Antonopoulos, 2017). Κάθε μπλοκ περιέχει ένα κρυπτογραφημένο κωδικό κατακερματισμού του προηγούμενου μπλοκ (hash), στίγματα χρόνου των συναλλαγών και δεδομένα. Το αρχείο συναλλαγών διατηρείται σε διάφορους υπολογιστές οι οποίοι είναι

συνδεδεμένοι σε ένα peer-to-peer (p2p) δίκτυο (Nakamoto, 2008). Έτσι η τεχνολογία Blockchain επιτρέπει να υπάρχουν καταναμημένα και αμετάβλητα δεδομένα με ασφαλή και κρυπτογραφημένο τρόπο. Διασφαλίζοντας παράλληλα ότι οι συναλλαγές δεν μπορούν ποτέ να τροποποιηθούν.

2.3.2 Πως λειτουργεί το Blockchain

Το blockchain, όπως προαναφέρθηκε, είναι μια αποκεντρωμένη δομή δεδομένων στην οποία αποθηκεύονται ομάδες συναλλαγών σε ένα καταναμημένο δίκτυο. Οι συμμετέχοντες του δικτύου έχουν πρόσβαση σε όλα τα ιστορικά δεδομένα και τις συναλλαγές, συμπεριλαμβανομένου της ώρας δημιουργίας των μπλοκ. Κάθε μπλοκ αποτελείται από συγκεκριμένες πληροφορίες, μια λίστα με τις συναλλαγές που έγιναν σε κάποιο χρονικό διάστημα, μια σφραγίδα του χρόνου δημιουργίας του και μια ψηφιακή αναφορά (hash) στο προηγούμενο μπλοκ. Με την πάροδο του χρόνου δημιουργείται μια ταξινομημένη σειρά από μπλοκ (βλ. **Εικόνα 3**). Για να επαληθευτεί η εγκυρότητα των μπλοκ χρησιμοποιούνται αλγόριθμοι συναίνεσης όπου μπορούν να το επιτύχουν σχεδόν σε πραγματικό χρόνο. Αυτό γίνεται σε συνδυασμό με την κρυπτογραφημένη αναφορά, το hash, που υπάρχει και τον αλγόριθμο που το επικυρώνει. Στα δημόσια blockchains η διαδικασία αυτή ονομάζεται εξόρυξη (mining) (Acheson, 2018). Η εξόρυξη περιλαμβάνει μια μορφή ανταμοιβής για κάποιον που θα επαληθεύσει ή θα επικυρώσει ένα μπλοκ. Για παράδειγμα στο Bitcoin όποιος επαληθεύσει ένα μπλοκ θα λάβει μια συγκεκριμένη αμοιβή από Bitcoin (Transaction Fees + Block Reward). Αυτού του είδους η διαδικασία ονομάζεται απόδειξης της εργασίας (Proof-of-Work, PoW) και ο σκοπός της είναι να επικυρώσει ότι ο αλγόριθμος είναι αληθής και η συναλλαγή είναι νόμιμη (Wu, et al., 2017). Αν κάποιος προσπαθήσει να αλλάξει ή να παραβιάσει τις πληροφορίες συναλλαγής που καταγράφονται σε ένα μπλοκ, το hash για το συγκεκριμένο μπλοκ θα αλλάξει και δεν θα δείχνει πια το hash του προηγούμενου, επαληθευμένου μπλοκ. Τέλος, το πρώτο block που δημιουργείται στην αλυσίδα ονομάζεται Genesis μπλοκ, είναι ειδική περίπτωση, γι' αυτό

δεν έχει κάποια αναφορά σε προηγούμενο μπλοκ και είναι γραμμένο στον κώδικα που ξεκινάει το blockchain.



Εικόνα 3 Δομή του Blockchain (International Bank for Reconstruction and Development, 2017)

2.3.3 Χαρακτηριστικά του Blockchain

Το blockchain λόγω της αρχιτεκτονικής και του τρόπου που λειτουργεί έχει κάποια πολύ σημαντικά χαρακτηριστικά:

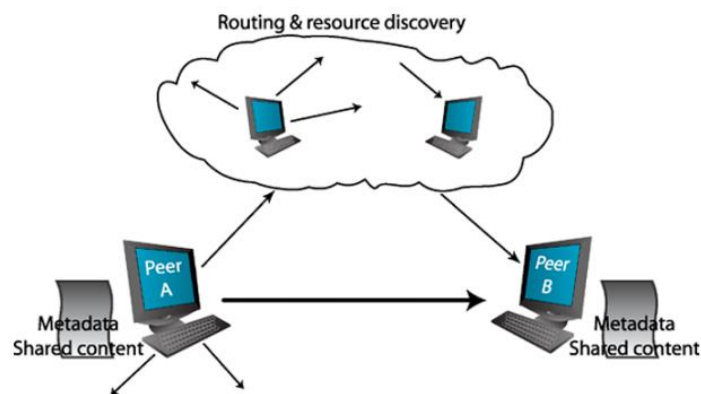
- **Αμετάβλητο (Immutable)** – (permanent and tamper-proof) Το blockchain είναι ένα μόνιμο αρχείο συναλλαγών. Εφόσον προστεθεί ένα block στον κόμβο δεν μπορεί να τροποποιηθεί. Αυτό οδηγεί στην ασφάλεια και στην εμπιστοσύνη των συναλλαγών.
- **Αποκεντρωμένο** – αυτό αποθηκεύεται σε ένα αρχείο το οποίο είναι προσβάσιμο και αντιγράφεται από οποιονδήποτε κόμβο του δικτύου. Το οποίο και δημιουργεί την αποκέντρωση.
- **Διαφανές (Transparent)** – (ολόκληρο ιστορικό συναλλαγών) κάθε μέρος μπορεί να έχει πρόσβαση και να ελέγχει τις συναλλαγές. Αυτό δημιουργεί την έννοια της

προέλευσης και της παρακολούθησης ολόκληρης της ζωής του περιουσιακού στοιχείου.

- **Βασίζεται στη Συναίνεση** (Consensus Driven) – κάθε μπλοκ στο blockchain επαληθεύεται ανεξάρτητα μέσω μοντέλων συναίνεσης που παρέχουν κανόνες για την επικύρωση ενός μπλοκ. Πολύ συχνά στα δημόσια δίκτυα χρειάζονται πολλοί πόροι για να το πετύχουν αυτό (π.χ. επεξεργαστική ισχύ). Στο bitcoin η διαδικασία αυτή ονομάζεται εξόρυξη. Ο μηχανισμός αυτός λειτουργεί χωρίς κάποια κεντρική αρχή.

2.3.4 Peer-to-Peer δίκτυα

Τα ομότιμα ή γνωστά ως peer-to-peer (P2P) δίκτυα επιτρέπουν δύο ή περισσότερους υπολογιστές να διαμοιράζουν ισοδύναμα τους πόρους τους. Λειτουργεί ταυτόχρονα ο κάθε υπολογιστής ως πελάτης και ως εξυπηρετητής (Schollmeier, 2002). Οι πληροφορίες που βρίσκονται στον κάθε κόμβο διαμοιράζονται και συγχρονίζονται κατ' επιλογή μεταξύ τους δίχως κεντρικό εξυπηρετητή. Αυτό διαφέρει από ένα μοντέλο κεντρικού διακομιστή που επιβραδύνεται όταν περισσότεροι χρήστες συμμετέχουν, καθώς το δίκτυο P2P μπορεί να βελτιώσει την ισχύ του με περισσότερες συσκευές ή κόμβους που ενώνουν το δίκτυο.



Εικόνα 4 Επικοινωνία Peer-2-Peer

Αυτή η μέθοδος μεταφοράς πληροφοριών αποτελεί σημαντικό συστατικό στοιχείο του blockchain επειδή τα δεδομένα δεν αποθηκεύονται σε ένα συγκεντρωτικό σημείο,

καθιστώντας την, πολύ λιγότερο ευάλωτη σε εκμετάλλευση, αλλοίωση ή απώλεια. Για παράδειγμα στην **Εικόνα 4** ο Peer A ζητάει κάποια δεδομένα που έχει ο Peer B. Πρώτα πρέπει μέσω του P2P δικτύου να εντοπίσει πως το B είναι αυτός που έχει τα δεδομένα. Στη συνέχεια εφόσον εντοπιστεί δημιουργείται απευθείας σύνδεση του Peer A με τον Peer B.

2.3.5 Bitcoin και Ethereum

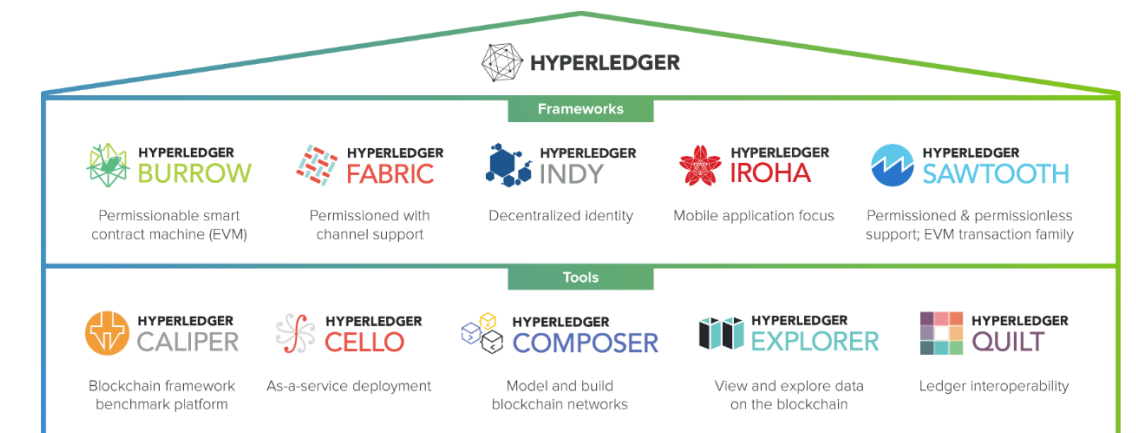
Το Bitcoin δημιουργήθηκε το 2008 από έναν άνθρωπο ή ομάδα (The Economist, 2015) που αποκαλείται με το ψευδώνυμο Satoshi Nakamoto. Υλοποίησε ένα αποκεντρωμένο σύστημα ηλεκτρονικού νομίσματος βασισμένο στις Peer-to-Peer επικοινωνίες [8] [9]. Η υποκείμενη τεχνολογία με την οποία λειτουργεί το Bitcoin ονομάστηκε Blockchain. Η καινοτομία που παρείχε ο Satoshi είναι η ιδέα του να συνδυάσει ένα αποκεντρωμένο πρωτόκολλο συναίνεσης βασισμένο σε κόμβους που συλλέγουν συναλλαγές σε μπλοκ κάθε δέκα λεπτά. Έτσι δημιουργείται μια αυξανόμενη αλυσίδα από μπλοκ όπου με τον μηχανισμό απόδειξης της εργασίας οι κόμβοι κερδίζουν το δικαίωμα να συμμετέχουν στο δίκτυο. Ταυτόχρονα οι κόμβοι με μεγάλη υπολογιστική ισχύ έχουν αναλογικά μεγαλύτερη επιρροή. Με αυτό τον τρόπο επιτυγχάνεται η αξιόπιστη και η σωστή λειτουργία του αποκεντρωμένου νομίσματος, χωρίς κάποιο διαχειριστή ή μεσάζοντες, όπως κάποια κεντρική τράπεζα.

Το Ethereum είναι μια ανοιχτή πλατφόρμα blockchain η οποία δίνει τη δυνατότητα να δημιουργούνται, να εκτελούνται και να χρησιμοποιούνται αποκεντρωμένες εφαρμογές (DApps) και έξυπνα συμβόλαια (smart contracts) [12]. Χαρακτηριστική είναι η διευκόλυνση και η αυτοματοποίηση που προσφέρει στην αλληλεπίδραση μεταξύ των συμμετεχόντων του δικτύου. Όπως και με το Bitcoin, το Ethereum επιτρέπει τη δημιουργία ενός συστήματος πληρωμών χωρίς μεσάζοντες. Το ether (ETH) είναι το νόμισμα της πλατφόρμας και λειτουργεί επίσης ως αντίτιμο για τις εξορύξεις στο δίκτυο Ethereum. Για την κατασκευή των αποκεντρωμένων εφαρμογών, η πλατφόρμα προσφέρει μια ειδική γλώσσα προγραμματισμού turing-complete. Στη συνέχεια αυτή μεταφράζεται και τρέχει

σε εικονικό σύστημα του (Ethereum Virtual Machine). Τεχνικά, λόγω του turing-complete συστήματος, το μεγάλο πλεονέκτημα είναι πως οι εφαρμογές και τα έξυπνα συμβόλαια που υποστηρίζει η πλατφόρμα μπορούν να γραφτούν σε μια πληθώρα από σύγχρονες και παλιές γλώσσες προγραμματισμού, όπως C++, Python, Ruby, Go, Java, JavaScript, Rust κ.α..

2.3.6 Hyperledger Project

Το Hyperledger Project ξεκίνησε το 2015 από το Linux Foundation. Είναι ένα σύνολο από blockchains και εργαλεία πάνω σε αυτό (βλ. **Εικόνα 5**) που έχουν ως στόχο να δημιουργήσουν σε επιχειρηματικό επίπεδο μια δομή ανοιχτού κώδικα για καταναμημένα καθολικά (Distributed Ledger) (Cachin, 2016). Μέλη και υποστηρικτές σε αυτό το εγχείρημα είναι μεγάλες εταιρείες του κλάδου όπως η IBM, η Intel, η Cisco και η SAP.



Εικόνα 5 Blockchain Frameworks και Εργαλεία του Hyperledger

Το Hyperledger Fabric είναι μια υλοποίηση πλατφόρμας καταναμημένου καθολικού όπου τρέχει έξυπνα συμβόλαια, αξιοποιεί αποδεδειγμένες τεχνολογίες και έχει μια αρχιτεκτονική που επιτρέπει να «κουμπώνουν» και να υλοποιούνται διάφορες άλλες λειτουργίες. Γι' αυτό το λόγο είναι ευέλικτο για μια πληθώρα χρήσεων blockchain (Gaur, et al., 2018). Επίσης επιτρέπει την δημιουργία ξεχωριστών επιπέδων ασφαλείας και αδειοδοτεί μόνο πιστοποιημένους χρήστες. Λόγω της κρυπτογράφησης των συναλλαγών

και σε συνδυασμό με τα προηγούμενα είναι ιδανικό για επιχειρηματικά περιβάλλοντα μιας και πετυχαίνει την εμπιστευτικότητα των συναλλαγών και την επιλεκτική πρόσβαση μεταξύ των συμμετεχόντων. Για παράδειγμα σε μια υλοποίηση για βιομηχανίες αυτοκινήτων, οι διαφορετικές βιομηχανίες δεν θα έχουν πρόσβαση σε ανταγωνιστικές πληροφορίες. Αλλά μόνο σε αυτές που έχουν διαμοιράσει σε αυτούς που πρέπει και αμφίδρομα. Έτσι επιτυγχάνεται η εμπιστοσύνη των συμμετεχόντων στο διαμοιρασμό της πληροφορίας, σε συνδυασμό με όλα τα οφέλη του blockchain. Το Hyperledger Fabric βασίζεται στον αλγόριθμο συναίνεσης BFT (Byzantine Fault Tolerant) σε αντίθεση με το PoW του Bitcoin (Rilee, 2018). Η υπηρεσία εντολών (orderer) του Hyperledger πρέπει να ελέγχεται από κοινού, από τα μέλη του δικτύου, χρησιμοποιώντας έναν αλγόριθμο BFT που αντιστέκεται σε κακόβουλες δραστηριότητες (Gaur, et al., 2018).

2.3.7 Permissioned και Permissionless Blockchains

Όπως φαίνεται στις προηγούμενες ενότητες, υπάρχουν πολλές υλοποιήσεις blockchain. Αυτές μπορούν να κατηγοριοποιηθούν σε δύο μεγάλα σύνολα, τα δημόσια blockchains χωρίς δικαιώματα και τα ιδιωτικά ή επιχειρηματικά blockchains με δικαιώματα (Bauerle, 2017). Στα δημόσια οποιοσδήποτε ενδιαφερόμενος μπορεί να συμμετάσχει στο δίκτυο. Η πρόσβαση είναι ανοιχτή στα δεδομένα τους διαβάζοντας την αλυσίδα και επαληθεύοντας τα μπλοκ δημιουργώντας διαφάνεια στην πληροφορία. Η επαλήθευση των μπλοκ γίνεται από τους εξορυκτές (miners) και υπάρχει η δυνατότητα να εξορύξουν όλοι αναζητώντας την ανταμοιβή τους. Έτσι επιτυγχάνεται η ασφαλής αποκέντρωση του συστήματος μιας και δεν χρειάζεται τα μέλη να εμπιστεύονται το ένα τον άλλο. Από την άλλη πλευρά ένα πρόβλημα για τις επιχειρήσεις προέκυψε λόγω του δημόσιου blockchains, οι συναλλαγές είναι εντελώς διαφανείς για όλους. Αυτό εξαλείφει την ανταγωνιστικότητα των επιχειρήσεων, επειδή δεν θέλουν να παρουσιάζουν όλες τις πληροφορίες τους (Terzi & Stamelos, 2018). Υπάρχουν πολλές περιπτώσεις εφαρμογών όπου οι συναλλαγές ή τα περιουσιακά στοιχεία δεν πρέπει να κοινοποιούνται ή να είναι προσβάσιμα από όλους, αλλά από επιλεγμένους συμμετέχοντες, π.χ. συναλλαγές ανταγωνιστών, ιατρικό ιστορικό

και μεταφορά αγαθών. Γι' αυτό το λόγο δημιουργήθηκαν τα ιδιωτικά blockchains. Είναι χρήσιμα σε περιπτώσεις όπου η ακεραιότητα του ίχνους δεν είναι το πιο σημαντικό και υπάρχει η ανάγκη να τυποποιηθεί η ανταλλαγή πληροφοριών με ασφαλή τρόπο μεταξύ εταιρών, όπως ανάμεσα στις βιομηχανίες.

Στον παρακάτω πίνακα (βλ. **Πίνακας 1**) (International Bank for Reconstruction and Development, 2017) παρατηρείται συνοπτικά η σύγκριση των κύριων χαρακτηριστικών ανάμεσα στα Blockchains με δικαιώματα και χωρίς. Ενώ παρακάτω (βλ. **Πίνακας 2**) συγκρίνονται τα βασικά χαρακτηριστικά από τρία μεγάλα blockchains, το Bitcoin, το Ethereum και το Hyperledger Fabric.

Πίνακας 1 Σύγκριση χαρακτηριστικών Permissioned με Permissionless Blockchains (International Bank for Reconstruction and Development, 2017)

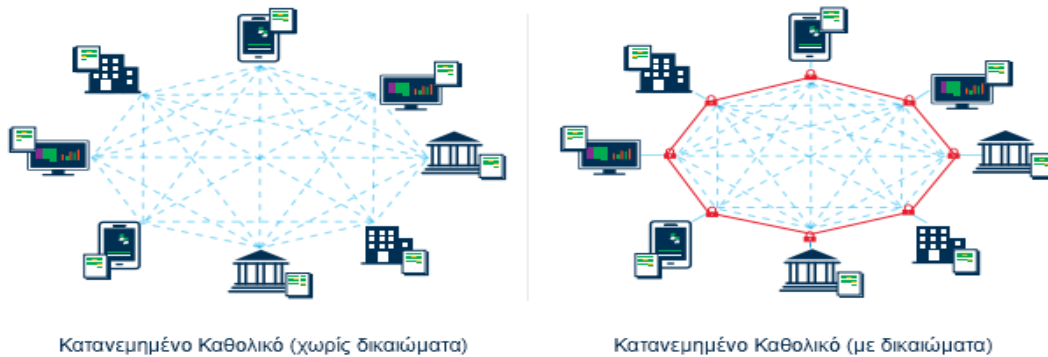
	Blockchains χωρίς δικαιώματα	Blockchain με δικαιώματα
Κεντρικό μέρος	Χωρίς κεντρικό ιδιοκτήτη ή διαχειριστή	Υπάρχει σε κάποιο βαθμό εξωτερικός διαχειριστής/ές ή έλεγχος
Πρόσβαση	Μπορεί οποιοσδήποτε να συμμετάσχει	Μόνο επιλεγμένοι συμμετέχοντες έχουν πρόσβαση στο δίκτυο
Επίπεδο εμπιστοσύνης	Δεν χρειάζεται τα μέλη του δικτύου να έχουν εμπιστοσύνη μεταξύ τους	Υψηλός βαθμός εμπιστοσύνης μεταξύ των μελών
Ανοιχτό	Το καθολικό είναι ανοιχτό και διαφανές αφού διαμοιράζεται μεταξύ όλων των συμμετεχόντων	Δυνατότητα σε διαφορετικό βαθμό ανοιχτού και διαφανές δικτύου
Ασφάλεια	Διαμέσου ευρείας κατανομής	Έλεγχος πρόσβασης σε συνδυασμό με τεχνολογία καταμεμημένου καθολικού
Ταχύτητα	Αργή επεξεργασία συναλλαγών	Γρήγορη επεξεργασία συναλλαγών
Ταυτότητα	Κυρίως ανώνυμα ή ψευδοανώνυμα	Ταυτότητα μέσω ιδιοκτήτη ή διαχειριστή
Συναίνεση (Consensus)	Δυσκολίας Proof-of-Work (συνήθως)	Διάφοροι αλγόριθμοι π.χ. PBFT, Kafka (τυπικά πιο εύκολοι από τον PoW)
Περιουσιακό στοιχείο	Συνήθως η φύση τους είναι κρυπτονομίσματα αλλά υπάρχει η δυνατότητα μέσω tokens να αντιπροσωπευθεί οποιοδήποτε περιουσιακό στοιχείο	Οποιοδήποτε περιουσιακό στοιχείο
Ιδιοκτησία	Υπάρχουν νομικά θέματα σχετικά με την ιδιοκτησία διότι κανένα νομικό πρόσωπο δεν ελέγχει ή κατέχει το καθολικό	Υπάρχει μεγαλύτερη νομική σαφήνεια ως προς την ιδιοκτησία. Συνήθως είναι ο ιδιοκτήτης ή ο διαχειριστής
Παραδείγματα	Bitcoin, Ethereum	Hyperledger Fabric

Πίνακας 2 Σύγκριση χαρακτηριστικών Bitcoin, Ethereum και Hyperledger (Friebe, 2017)

Σύγκριση χαρακτηριστικών			
Χαρακτηριστικά	Bitcoin	Ethereum	Hyperledger Fabric
Πρόσβαση	Permissionless	Permissionless	Permissioned
Ιδιωτικότητα δεδομένων	Δημόσια	Δημόσια ή Ιδιωτικά	Ιδιωτικά
Συναίνεση	Proof-of-Work	Proof-of-Work	PBFT
Επεκτασιμότητα	Υψηλή κόμβου Χαμηλή απόδοσης	Υψηλή κόμβου Χαμηλή απόδοσης	Χαμηλή κόμβου Υψηλή απόδοσης
Κεντρικός έλεγχος	Χαμηλός, αποκεντρωμένες αποφάσεις που λαμβάνονται από την κοινότητα/miners	Μέτριος, κύρια ομάδα ανάπτυξης αλλά και EIP διαδικασία	Χαμηλός, ανοιχτής διακυβέρνησης μοντέλο βασισμένο στο μοντέλο Linux
Ανωνυμία	Ψευδοανωνυμία, χωρίς κρυπτογράφηση	Ψευδοανωνυμία, χωρίς κρυπτογράφηση	Όχι, με κρυπτογράφηση
Νόμισμα	Ναι, το bitcoin	Ναι, το ether	Όχι
Κόστος συναλλαγής	Ναι	Ναι	Όχι
Ψευδογλώσσα	Περιορισμένη δυνατότητα, stack- based	Υψηλή δυνατότητα, turing-complete, υψηλού επιπέδου	Υψηλή δυνατότητα, turing-complete, υψηλού επιπέδου

2.3.8 Κατανεμημένο καθολικό (Distributed Ledger)

Η τεχνολογία του κατανεμημένου καθολικού (distributed ledger technology ή DLT) είναι μια νέα τεχνολογία στα πρώτα στάδια ανάπτυξής της. Είναι ένας τύπος βάσης δεδομένων που έχει την ιδιότητα να αναπαράγει, να διαμοιράζεται και να συγχρονίζει τα δεδομένα του σε πολλαπλά σημεία, κόμβων και συσκευών (UK Office for Science, 2016). Χαρακτηριστικό του κατανεμημένου καθολικού είναι ότι δεν υπάρχει κεντρικός διαχειριστής ή κεντρική αποθήκευση δεδομένων. Μόλις υπάρξει η συναίνεση, το κατανεμημένο καθολικό ενημερώνεται και όλοι οι κόμβοι διατηρούν το δικό τους όμοιο αντίγραφο του (Bauerle, 2017).



Εικόνα 6 Κατανεμημένο καθολικό (International Bank for Reconstruction and Development, 2017)

Με το blockchain η τεχνολογία αυτή έκανε την πιο σημαντική της παρουσία εφόσον ανέδειξε τις δυνατότητες της. Υπάρχουν δημόσια και ιδιωτικά κατανεμημένα καθολικά (βλ. **Εικόνα 6**). Στα δημόσια ο κάθε κόμβος κατέχει ένα ολόκληρο αντίγραφο του καθολικού. Στα ιδιωτικά ο κάθε κόμβος έχει δικαιώματα, μέσα από τον έλεγχο πρόσβασης μπορεί να συνδεθεί στο δίκτυο και να κάνει αλλαγές στο καθολικό (International Bank for Reconstruction and Development, 2017).

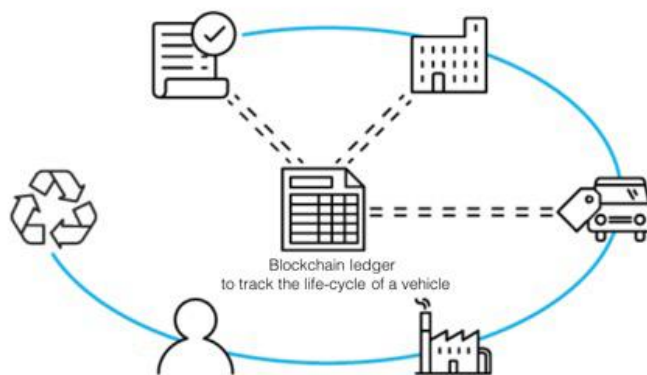
2.3.9 Έξυπνα συμβόλαια (Smart Contracts)

Ο όρος των έξυπνων συμβολαίων χρησιμοποιήθηκε αρχικά από τον Αμερικάνο επιστήμονα πληροφορικής και κρυπτογράφο Nick Szabo (Szabo, 1997). Χαρακτηριστικά αναφέρει το παράδειγμα της ενοικίασης ενός αυτοκινήτου με έξυπνο συμβόλαιο έτσι ώστε να αποτραπεί η κλοπή του αν δεν ικανοποιηθεί το πρωτόκολλο παράδοσης (Szabo, 1996). Το ίδιο θα μπορούσε να συμβεί σε μία περίπτωση τραπεζικού δανείου για αγορά αυτοκινήτου όπου ο ιδιοκτήτης δεν μπορούσε να πραγματοποιήσει τις πληρωμές, το έξυπνο συμβόλαιο αυτομάτως θα έκανε κατάσχεση επισωρεύοντας τα κλειδιά του αυτοκινήτου στην τράπεζα.

Τα έξυπνα συμβόλαια είναι μικρά αυτόνομα προγράμματα που λαμβάνουν μια συναλλαγή ως είσοδο, επεξεργάζονται και παράγουν μια νέα έξοδο. Η εκτέλεσή τους ξεκινάει αυτόματα, υπό συγκεκριμένες συνθήκες και παράγει ένα αποτέλεσμα. Αυτό

πραγματοποιείται στο επίπεδο του blockchain όπου ζει η επιχειρηματική λογική και συγκεκριμένες προγραμματικές λειτουργίες που εξυπηρετούν μια περίπτωση χρήσης. Το κυριότερο όφελος του συμβολαίου είναι ότι το blockchain εγγυάται πως οι συμβατικοί όροι δεν μπορούν να τροποποιηθούν και είναι αδύνατη η παραβίαση τους. Με την εφαρμογή αυτών αναμένεται να μειωθεί το κόστος εκτέλεσης, επαλήθευσης, ελέγχου και αποφυγής απάτης μια σύμβασης. Τέλος, με τα έξυπνα συμβόλαια ξεπερνιέται το πρόβλημα του ηθικού κινδύνου.

Η χρήση έξυπνων συμβολαίων σε ένα blockchain ξεκίνησε με το Bitcoin, το οποίο προσέφερε μια περιορισμένη ψευδο-γλώσσα (scripting language). Αυτή η γλώσσα έχει «κλαδευτεί» ακόμη περισσότερο μετά την εισαγωγή της καθώς εντοπίστηκαν ευπάθειες όταν εκτελούνταν ορισμένες λειτουργίες. Το Ethereum επέκτεινε αυτή την έννοια, των έξυπνων συμβολαίων, με την επεξεργασία των συναλλαγών σε μια ειδικά δημιουργημένη εικονική μηχανή. Τα έξυπνα συμβόλαια γράφονται σε γλώσσα υψηλότερου επιπέδου, πιο συχνά την Solidity, τα οποία στη συνέχεια μεταγλωττίζονται σε ψηφιακό κώδικα Ethereum Virtual Machine (EVM). Η λογική του συμβόλαιου και ο κώδικας διαμοιράζεται στο δίκτυο ούτως ώστε να χρησιμοποιηθεί από τους συμμετέχοντες για να επικυρώσουν και να επεξεργαστούν τις συναλλαγές. Το αποτέλεσμα της εσωτερικής κατάστασης του συμβολαίου γράφεται κατανεμημένα.



Εικόνα 7 Διάρκεια ζωής του οχήματος μέσω Smart Contracts σε blockchain. Παραγωγός-Πελάτης-Απόσυρση.

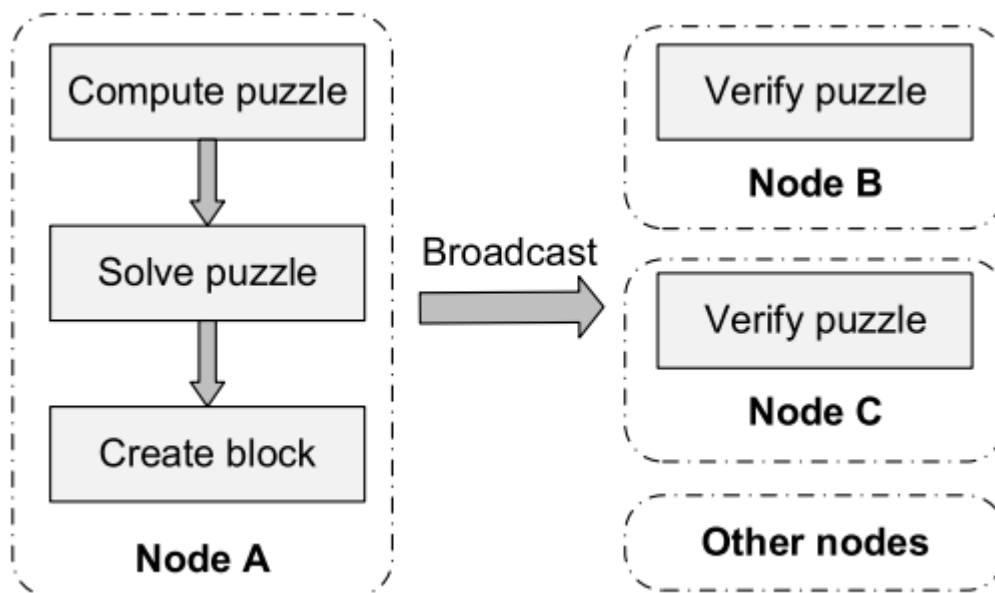
Σε σύγχρονες εφαρμογές blockchain για επιχειρήσεις αξιοποιείται μια ακόμα υλοποίηση από έξυπνα συμβόλαια, αυτά του Hyperledger Fabric. Έχει αναπτυχθεί ο chaincode που

είναι ο κώδικας στον οποίο γράφονται τα έξυπνα συμβόλαια. Περιλαμβάνει την ερμηνεία αυτών σε λογική μεθόδων και αλγορίθμων μαζί με επιπρόσθετες λειτουργίες (Hyperledger Fabric Docs, 2018). Με συγκεκριμένες ρυθμίσεις στην πολιτική ορίζεται ποιοι ακριβώς κόμβοι ή χρήστες, ή πόσοι από αυτούς θα εκτελέσουν ένα smart contract. Επόμενος η κάθε συναλλαγή εκτελείται μόνο από ένα υποσύνολο κόμβων. Αυτό επιτρέπει παράλληλες εκτελέσεις, αυξάνοντας τη συνολική απόδοση και την κλίμακα του συστήματος. Η γλώσσα προγραμματισμού που μπορούν να συνταχθούν είναι είτε Go είτε Node.js. Παράλληλα υπάρχει η δυνατότητα ανάπτυξης εφαρμογών blockchain στο Hyperledger Composer όπου για την δημιουργία έξυπνων συμβολαίων χρησιμοποιείτε διερμηνέας σε γλώσσα JavaScript που εκτελεί τη λογική για την επεξεργασία της συναλλαγής σε chaincode του HyperLedger Fabric (IBM, 2018). Στην **Εικόνα 7** παρατηρείται ένα χαρακτηριστικό παράδειγμα της IBM (IBM, 2017) σχετικά με τον κύκλο ζωής των οχημάτων. Πολλά μέρη εμπλέκονται στον κύκλο ζωής ενός οχήματος, συμπεριλαμβανομένου του κατασκευαστή, των αντιπροσώπων αυτοκινήτων, της ρυθμιστικής αρχής των οχημάτων, των ασφαλιστικών εταιρειών, των ιδιοκτητών και των αντιπροσώπων των απορριμμάτων. Όλα τα μέρη στο επιχειρηματικό δίκτυο επωφελούνται από την ύπαρξη αξιόπιστης, κατανεμημένης πηγής αλήθειας για την ιστορία και την ιδιοκτησία ενός οχήματος. Με την ενσωμάτωση της λογικής επιχειρηματικών αποφάσεων σε έξυπνες συμβάσεις που τρέχουν στο blockchain διαφαίνεται η αξία και η διαφάνεια μιας τέτοιας υλοποίησης.

2.3.10 Συναίνεση (Consensus)

Ως συναίνεση στην τεχνολογία αυτή ορίζεται η διαδικασία κατά την οποία επικυρώνεται η αξιοπιστία των συναλλαγών σε ολόκληρο το δίκτυο και η απόρριψη των ελαττωματικών διαδικασιών. Υπάρχουν πολλοί αλγόριθμοι συναινέσεις με διαφορετικά χαρακτηριστικά αλλά εξυπηρετώντας τον ίδιο σκοπό. Σε ένα ανοιχτό blockchain, όπως το bitcoin χρειάζεται μεγάλη επεξεργαστική ισχύ για τον PoW. Οι εξορυκτές (miners) ανταγωνίζονται μεταξύ τους για να ολοκληρώσουν τις συναλλαγές στο δίκτυο και να ανταμειφθούν (βλ. **Εικόνα 8**)

(Li, et al., 2017). Δείχνοντας μεγαλύτερη εμπιστοσύνη για τις επικυρώσεις των συναλλαγών, π.χ. το παγκόσμιο σύστημα χρηματοπιστωτικών συναλλαγών Ripple, επιλέγει μια λίστα επικυρωτών (validators) γνωστών ή μερικώς γνωστών. Κάθε λίγα δευτερόλεπτα εφαρμόζεται ο αλγόριθμος RPCA (Ripple Consensus Algorithm) από όλους τους κόμβους. Είναι βασισμένος στην έννοια του proof of correctness και είναι βελτιστοποιημένος και ταχύτερος από αυτόν του PoW (Schwartz, et al., 2014). Ένα ακόμη διαδεδομένο μοντέλο κατανεμημένης συναίνεσης είναι το Proof-of-Stake, στο οποίο επιλέγεται ο δημιουργός του επόμενου μπλοκ βάση συνδυασμών τυχαίας επιλογής π.χ. πλούτου, ηλικίας. Αυτό είναι το ποντάρισμα (stake) για την επικύρωση του μπλοκ. Τέλος ακόμα ένας ευρέως διαδεδομένος αλγόριθμος είναι ο Byzantine Fault Tolerance (BFT) που εγγυάται την κάλυψη ή την ικανότητα να επιτύχει συναίνεση, ακόμα και αν υπάρχουν αντίπαλοι κόμβοι (κακόβουλοι) ή αν οι κόμβοι βρεθούν εκτός δικτύου. Δεν χρειάζεται εξορυκτές και βρίσκει μεγάλη εφαρμογή σε ιδιωτικά blockchain όπως το Hyperledger Fabric.



Εικόνα 8 Μηχανισμός συναίνεσης PoW (Li, et al., 2017)

2.3.11 Εφαρμογές blockchain στην εφοδιαστική αλυσίδα σήμερα

Ο Christian Catalini, κύριος ερευνητής του MIT στον τομέα Digital Currencies Research Study, αναφέρει πως η τεχνολογία του blockchain έχει τη δυνατότητα να εξορθολογήσει και να απλοποιήσει την αλυσίδα εφοδιασμού, ιδίως στις περιπτώσεις όπου εμπλέκονται πολλές οντότητες, οι συμμετέχοντες είναι διασκορπισμένοι γεωγραφικά και υπάρχει συνολική πολυπλοκότητα (Catalini & Gans, 2017) (Church, 2017). Σε αυτή τη βιομηχανία παρατηρούνται σήμερα αρκετές υλοποιήσεις που στοχεύουν να λύσουν χρόνια προβλήματα.

Η πιο μεγάλη και γνωστή περίπτωση είναι η συνεργασία της IBM με την Walmart για την υλοποίησης του έργου «Food Safety». Τον Οκτώβριο του 2016 ξεκινάει η πιλοτική έρευνα της ιχνηλασιμότητας προϊόντων μέσω του blockchain (Kamath, 2018). Στόχος ήταν η επένδυση \$24 εκατομμυρίων σε βάθος πέντε ετών για να ερευνηθεί η παγκόσμια ασφάλεια τροφίμων. Χρησιμοποιώντας την πλατφόρμα της IBM, το Hyperledger Fabric, ολοκλήρωσε με επιτυχία δύο πιλοτικά. Το χοιρινό κρέας στην Κίνα και τα μάνγκο στην Αμερική. Με έναν προσανατολισμό από την φάρμα στο τραπέζι πέτυχε την μείωση της ιχνηλασιμότητας σε όλο το εύρος της εφοδιαστικής αλυσίδας από επτά μέρες σε 2,2 δευτερότατα. Ενώ παράλληλα προώθησε την διαφάνεια της διαδικασίας σε όλα τα επίπεδα. Τον Αύγουστο του 2017 προχώρησε στην ενσωμάτωση μεγάλων κολοσσών όπως Dole, Nestlé, Unilever, Driscoll's, Golden State Foods, Kroger, McLane Company, και Tyson Foods (IBM, 2017).

Η Αμερικάνικη εταιρία IBM, σε ακόμα μια επένδυση της στο blockchain, συνεργάστηκε με την Maersk από την Δανία για να υλοποιήσει το «TradeLens». Η λύση αυτή ψηφιοποιεί και διαχειρίζεται το ίχνος χαρτιού του μεταφέρουν εμπορευματοκιβώτια σε ολόκληρο τον πλανήτη. Η αποστολή μεταξύ ηπείρων απαιτεί γραμματόσημα και εγκρίσεις από περισσότερα των 30 ατόμων και οργανώσεων, όπως τελωνειακές αρχές, μεταφορείς, πράκτορες, φορτωτές κ.λπ. Επιπροσθέτως περιλαμβάνει πάνω από 200 επικοινωνίες μεταξύ διαφορετικών συμμετεχόντων (IBM, 2017). Τα γεγονότα, τα έγγραφα και τα δεδομένα της εφοδιαστικής αλυσίδας ανταλλάσσονται σε πραγματικό χρόνο. Η ψηφιακή

υποδομή συνδέει τους συμμετέχοντες σε ένα οικοσύστημα, ενσωματώνοντας τις διαδικασίες και τους εταίρους της ναυτιλίας και δημιουργεί ένα πλαίσιο αξιοπιστίας, αυξημένες διαφάνειας και εμπιστοσύνης.

Τον Φεβρουάριο του 2018 το EXIMCHAIN κάνει το ντεμπούτο του ως κρυπτονόμισμα και καταφέρνει να συγκεντρώσει \$20 εκατομμύρια ως ICO (Initial Coin Offering) (ICODROPS, 2018). Η αρχική ιδέα και η ανάπτυξη του ξεκίνησε το 2015, ενώ το 2016 ιδρύεται η startup. Ο σκοπός του είναι να παρέχει χρηματοοικονομικές λύσεις στην εφοδιαστική αλυσίδα μέσω ενός υβριδικού δημόσιου και ιδιωτικού blockchain. Στοχεύει μικρομεσαίες επιχειρήσεις που δραστηριοποιούνται στις αγορές και στις προμήθειες. Η πλατφόρμα παρέχει εργαλεία βελτιστοποίησης που ταιριάζουν στις ανάγκες του κλάδου. Μέσω των έξυπνων συμβολαίων δίνει την δυνατότητα προσιτών πηγών κεφαλαίου για να αναπτύξουν των επιχειρήσεων και δίνουν στους χρηματοδότες ορατότητα στην ταμειακή ροή της αλυσίδας εφοδιασμού (Huertas, et al., 2018).

Το κρυπτονόμισμα Tael γνωστό και ως WABI δημιουργήθηκε από την Κινεζική εταιρεία Techrock. Λειτουργεί ως μέσο επιβράβευσης για τον καταναλωτή και παράλληλα επιβεβαιώνει την αυθεντικότητα των προϊόντων. Οι καταναλωτές μπορούν να ξοδέψουν τα κέρματα τους με αγορές ασφαλών προϊόντων της Techrock (Techrock, n.d.). Η γένεση του συστήματος αυτού απορρέει από το σκάνδαλο του κινέζικου γάλακτος το 2008. Το σκάνδαλο περιλάμβανε το γάλα και τη βρεφική φόρμουλα μαζί με άλλα υλικά και συστατικά τροφίμων που είχαν αλλοιωθεί με μελαμίνη. Αυτό είχε σαν αποτέλεσμα να νοσηλευτούν 50.000 βρέφη, εκ των οποίων 6 κατέληξαν (Wikipedia, 2008). Ο στόχος της ομάδας είναι να αποτρέψει τους καταναλωτές από θάνατο ή σοβαρή κακουχία εξαιτίας αυτών των παραποιημένων αγαθών.

Τον Νοέμβριο του 2016 ιδρύθηκε το Waltonchain (WaltonChain, 2018). Είναι Κινεζικών και Κορεατικών συμφερόντων. Στόχος είναι ο συνδυασμός hardware και software με την χρήση RFIDs και Blockchain. Το όραμα τους είναι να υλοποιήσουν αυτό που αποκαλούν Value Internet of Things (VIoT). Με την τεχνολογία του blockchain θέλουν να φέρουν στην ανθρωπότητα μια αξιόπιστη ψηφιακή ζωή, να οικοδομήσουν το Internet of Everything

(IoT) και μια υγιή ανάπτυξη ενός οικοσυστήματος. Τα τσιπ RFID, που έχουν σχεδιαστεί από τους κατασκευαστές Waltonchain, αποθηκεύουν πληροφορίες σχετικά με τα φυσικά στοιχεία των προϊόντων. Η ομάδα ανέπτυξε επίσης ένα σαρωτή που διαβάζει δεδομένα από τα τσιπ RFID και μεταφορτώνει δεδομένα απευθείας στο blockchain. Τα τσιπ RFID και οι σαρωτές είναι πατενταρισμένη τεχνολογία. Υπήρξε υλοποίηση στην εφοδιαστική αλυσίδα του ρούχου. Πετυχαίνοντας υψηλής ποιότητας σύστημα ιχνηλασιμότητας και ελέγχου αυθεντικότητας των ενδυμάτων, όπως επίσης μείωση της επικοινωνίας, της ανάλυσης και του κόστους αποθήκευσης των δεδομένων.

Όπως λοιπόν γίνεται σαφές, η τεχνολογία του blockchain δημιουργήθηκε αρχικά για την λειτουργία του κρυπτονομίσματος Bitcoin, όμως, συνεχώς εξελίσσεται και τώρα το πεδίο εφαρμογής του έχει γίνει πολύ πιο ευρύ. Πρόκειται για μία σειρά καταχωρίσεων που αφορούν συναλλαγές, σε ένα δημόσιο σημειωματάριο (ledger) λαμβάνοντας χώρα σε ένα δημόσιο ή ιδιωτικό peer-to-peer δίκτυο. Εκτός από την λειτουργία κρυπτονομισμάτων, μπορεί να αποτελέσει πυλώνα για την δημιουργία και την λειτουργία αποκεντρωμένων εφαρμογών, που βασίζονται σε ένα κατακεντρωμένο δίκτυο ομότιμων κόμβων και όχι στους εξυπηρετητές κάποιου οργανισμού. Το blockchain μπορεί να έχει τεράστια επιρροή σε αυτοματοποιήσεις διεργασιών ανάμεσα σε επιχειρήσεις όταν συνδυάζεται με καινοτόμες τεχνολογίες όπως η μηχανική μάθηση, η τεχνητή νοημοσύνη και το Internet of Things. Η χρήση του ενεργοποιεί στη βιομηχανία της εφοδιαστικής αλυσίδας την διαφάνεια, τον καλύτερο διαμοιρασμό πληροφορίας και ενισχύει την ασφάλεια των τροφίμων. Ένα από τα χαρακτηριστικά παράδειγμα της χρήσης αυτής, είναι η εφαρμογή της IBM Food Trust για την εφοδιαστική αλυσίδα φρέσκου τροφίμου.

Σύμφωνα με όλα τα προαναφερθέντα, παρατηρείται πως η ιχνηλασιμότητα και η πληροφόρηση στην εφοδιαστική αλυσίδα είναι αναγκαία και σημαντική. Ωστόσο, τα περισσότερα από τα σημερινά συστήματα είναι κεντροποιημένα και δεν υπάρχουν αποκεντρωμένα συστήματα στην αλυσίδα εφοδιασμού γαλακτοκομικών προϊόντων για την ασφάλεια των τροφίμων. Σε αυτή τη διπλωματική, αναπτύσσεται μια αποκεντρομένη πλατφόρμα διαμοιρασμού, παρακολούθησης πληροφοριών και ιχνηλασιμότητας για την εφοδιαστική αλυσίδα γαλακτοκομικών βασισμένη στις τεχνολογίες του blockchain.

Συγκριτικά με τα κεντρικά συστήματα, το νέο σύστημα θα μπορούσε να γίνει μια καινοτόμα πλατφόρμα που θα παρέχει πληροφορία για το γάλα με αξιοπιστία, διαφάνεια και ασφάλεια, από τους συμμετέχοντες μέχρι και τον τελικό καταναλωτή. Το σύστημα αυτό μπορεί να δώσει μια νέα προοπτική και ιδέα για το πως μπορεί να διαμοιραστεί η πληροφορία και να βελτιώσει σημαντικά την ασφάλεια των τροφίμων στις αλυσίδες εφοδιασμού.

ΚΕΦΑΛΑΙΟ 3: Τεχνολογικό πλαίσιο

Οι περισσότερες από τις καθημερινές εργασίες γίνονται με τη βοήθεια κάποιας τεχνολογίας, με αποτέλεσμα να υπάρχει μεγαλύτερη πρόσβαση σε χρήσιμη πληροφορία, η οποία χρειάζεται επεξεργασία. Ένα από τα βασικά χαρακτηριστικά του διαδικτύου, το οποίο έχει για μεγάλο χρονικό διάστημα σταθερότητα, είναι η κεντροποιημένη αρχιτεκτονική στην αμφίδρομη σχέση πελάτη-εξυπηρετητή με την οποία είναι φτιαγμένη η πλειοψηφία των εφαρμογών και των υπηρεσιών. Αυτή λοιπόν τη σχέση προσπαθούν να την αλλάξουν ορισμένες νέες τεχνολογίες, πρωταγωνιστικό ρόλο παίζει το blockchain με τον αποκεντρωμένο χαρακτήρα του. Σε αυτό το κεφάλαιο θα αναλυθούν οι τεχνολογίες που επιλέχθηκαν για την υλοποίηση της πλατφόρμας που προτείνεται.

3.1 Blockchain Technologies

Μιλώντας γενικά, η εγκυρότητα των πληροφοριών βασίζεται στην εμπιστοσύνη της κεντροποιημένης μονάδας του συστήματος ή κάποιου τρίτου ισχυρού οργανισμού που την διαχειρίζεται. Ωστόσο, υπάρχει ασύμμετρη πληροφόρηση μεταξύ των οργανισμών και των συμμετεχόντων, η οποία και δημιουργεί μονοπώλιο στην κεντροποιημένη μονάδα. Αυτό έχει σαν αποτέλεσμα να γίνεται ευάλωτη και στόχος για hacking, δωροδοκία ή παραπληροφόρηση. Για παράδειγμα κάποιος διαχειριστής του κεντροποιημένου συστήματος μπορεί να δωροδοκηθεί για να αλλάξει έγκυρες πληροφορίες, με αποτέλεσμα να μετατρέψει σε αναξιόπιστο το σύστημα. Χάρη στο blockchain, αυτού του είδους τα προβλήματα μπορούν να επιλυθούν.

Το blockchain ουσιαστικά φαίνεται ως ένα τεχνικό σχήμα μιας αξιόπιστης βάσης δεδομένων, η οποία διατηρείται συλλογικά και αποκεντρωμένα με αξιόπιστες μεθόδους. Η τεχνολογία του blockchain είναι παρόμοια με το NoSQL (Not Only Structured Query

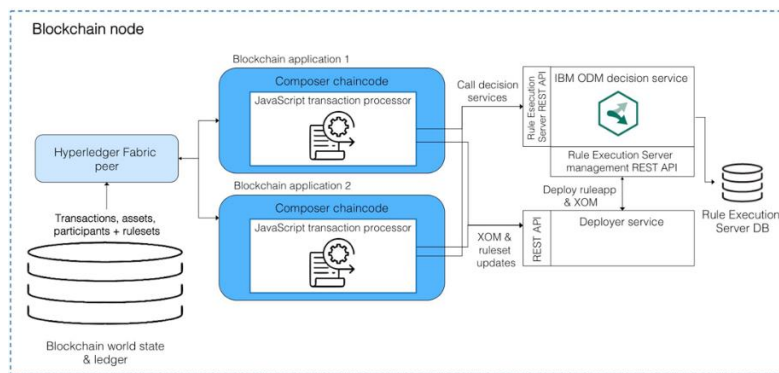
Language) (Meunier, 2016) και μπορεί να υλοποιηθεί με πολλά είδη γλωσσών προγραμματισμού όπως C++, Javascript, Python, Go, Solidity κ.α..

Η τεχνολογία του blockchain επιτρέπει την εμπιστοσύνη ακόμα και εκεί που δεν υπάρχει, την ασφάλεια και την ταυτοποίηση συστημάτων εφοδιαστικής αλυσίδας και ανταλλαγής αυτών. Λειτουργεί σε αποκεντρωμένο δίκτυο δίνοντας την δυνατότητα με τα έξυπνα συμβόλαια να αυτοματοποιούνται εκτελέσεις και επικυρώσεις διαδικασιών όταν συμβούν συγκεκριμένες συνθήκες.

3.1.1 Hyperledger Fabric & Composer

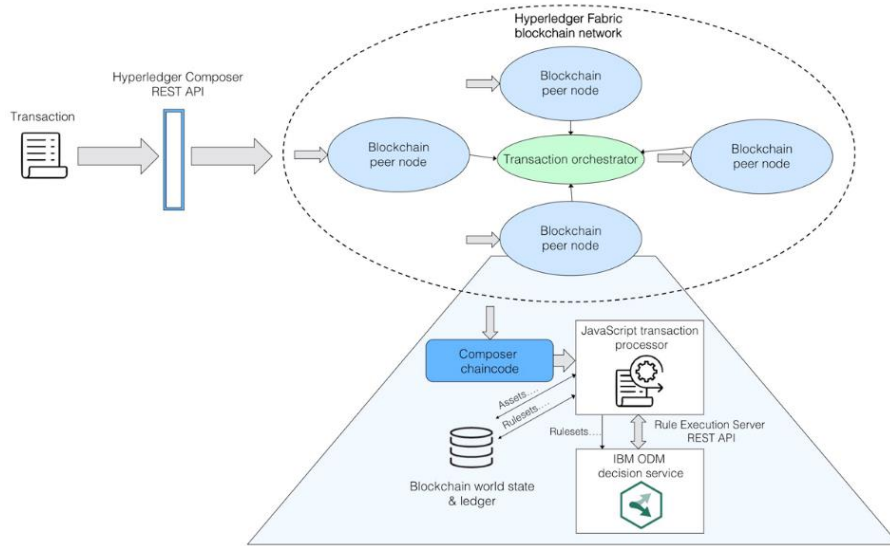
Ένα blockchain που αναπτύσσεται χρησιμοποιώντας το Hyperledger Fabric αποθηκεύει δεδομένα με τη μορφή chaincode, έναν προγραμματικό κώδικα στο δίκτυο που λειτουργεί παρόμοια με έξυπνες συμβάσεις σε άλλα blockchains. Το δίκτυο υποστηρίζει επί του παρόντος τις Golang, node.js ή Java ως τις γλώσσες για τον chaincode. Από προεπιλογή, το δίκτυο δεν περιλαμβάνει το δικό του κρυπτονόμισμα. Ωστόσο, οι χρήστες μέσω του chaincode μπορούν να αναπτύξουν το δικό τους.

Το Hyperledger Composer είναι ένα σύνολο από εργαλεία που παρέχουν ένα αφαιρετικό επίπεδο για την υλοποίηση μια επαγγελματικής λύσης blockchain σε πραγματικές περιπτώσεις χρήσης [16]. Σε αυτό το πλαίσιο δημιουργούνται εφαρμογές blockchain και έξυπνα συμβόλαια. Το Composer εκτελείτε πάνω από το Hyperledger Fabric και σε συνδυασμό με τα εργαλεία του όπως είναι η JavaScript, node.js, npm, Loopback API, CLI και δημοφιλή περιβάλλοντα ανάπτυξης κώδικα. Έτσι δίνει μια καλύτερη και εύχρηστη διαχείριση των τριών στοιχείων που ενδιαφέρουν μια επιχείρηση, τα περιουσιακά στοιχεία (assets), τις συναλλαγές (transactions) και του συμμετέχοντες (participants). Με αυτή την μοντελοποίηση ομαδοποιεί την διαχείριση των δεδομένων, τις λογικές λειτουργίες και την διαχείριση των μελών του δικτύου (Hyperledger Foundation, 2019) (IBM, 2018). Καταλήγοντας σε μια τυπική λειτουργία ενός κόμβου (βλ. **Εικόνα 9**) blockchain που λειτουργεί μέσα σε docker containers.



Εικόνα 9 Διάγραμμα αρχιτεκτονικής ενός κόμβου που εξυπηρετεί διάφορες εφαρμογές blockchain. (IBM, 2018)

Ένα εξωτερικό σύστημα, που θα μπορούσε να είναι μια διαδικασία μιας επιχείρησης, ενεργοποιεί μια συναλλαγή μέσω του REST API (βλ. **Εικόνα 10**). Η συναλλαγή κατανέμεται από το Hyperledger Fabric σε όλους τους κόμβους του blockchain. Ο κάθε κόμβος ενεργοποιεί το chaincode (έξυπνο συμβόλαιο) για να εφαρμοστεί η λογική πίσω από αυτό και να την ελέγξει. Το chaincode ενεργοποιεί την συναλλαγή του Hyperledger Composer όπου γίνεται η διερμηνεία του κώδικα JavaScript. Εκτελούνται οι απαραίτητες ενέργειες με της επιχειρηματικής λογικής και επιστρέφει μια απάντηση μέσω του REST API. Με αυτό τον τρόπο μπορούν να αυτοματοποιηθούν συναλλαγές της εφοδιαστικής αλυσίδας. Συλλέγονται οι πληροφορίες που αφορούν τις διαφορετικές εταιρείες της εφοδιαστικής που συμμετέχουν και δημιουργούνται γεγονότα που είναι το σημείο εκκίνησης κάποιας άλλης διαδικασίας. Επίσης με την συγκέντρωση πληροφοριών με αυτό τον τρόπο ενισχύεται η ορατότητα της ροής του προϊόντος.



Εικόνα 10 Η ροή της συναλλαγής στο Hyperledger

3.2 Παράλληλες Τεχνολογίες και Εργαλεία

Για να λειτουργήσει ένα ολοκληρωμένο σύστημα βασισμένο στο blockchain χρειάζονται πολλά εργαλεία και γλώσσες προγραμματισμού ώστε να συνδυαστούν και να δημιουργήσουν μια παραγωγική λύση. Έτσι θα αναλυθεί η σύγχρονη συλλογή εργαλείων που χρειάζεται από το Hyperledger Composer, όπως φαίνεται και στην εικόνα (βλ. **Εικόνα 11**). Συμβάλλοντας στην δόμηση του εμπρός, του πίσω και του ενδιάμεσου τμήματος μιας τέτοιας αρχιτεκτονικής υλοποίησης.



Εικόνα 11 Hyperledger Composer Tools-Stack

3.2.1 JavaScript

Η JavaScript είναι μια γλώσσα που πρωτοεμφανίστηκε το 1995 από την Netscape και Sun (Netscape Communications Corporation, 1995). Είναι μια γλώσσα υψηλού επιπέδου κατηγορίας διερμηνέα. Με τις τεχνολογικές εξελίξεις και την ραγδαία ανάπτυξη του internet και των browsers η JavaScript παράλληλα με την HTML και την CSS δομούν τεχνολογικά το World Wide Web. Ακολουθεί τις προδιαγραφές και εξελίσσεται με βάση το ECMAScript. Ενώ ταυτόχρονα έχει γνωρίσει μέσα στα χρόνια μεγάλες επεκτάσεις και υποκατάστατα όπως την VBScript από την Microsoft, την εκδοχή της Adobe, το Silverlight από την Microsoft, την εκδοχή της Google και φυσικά την TypeScript (Aston, 2015).

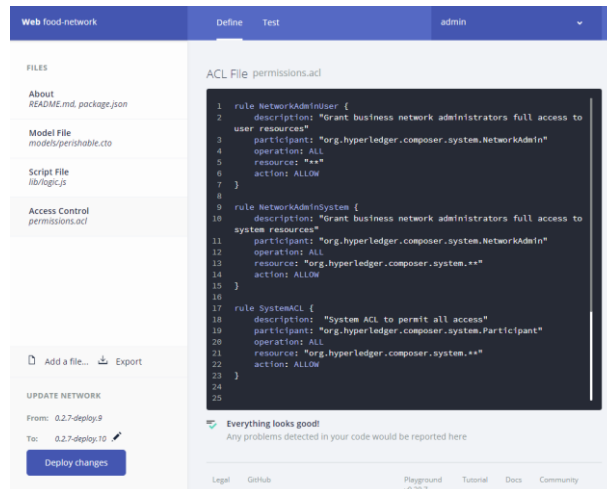
Η Angular είναι μια μοντέρνα cross-platform και ανοιχτού κώδικα γλώσσα για ανάπτυξη front-end web εφαρμογών. Είναι βασισμένη στην TypeScript και αναπτύσσεται από την Google. Ένα παράδειγμα αυτής, αποτελεί η παρούσα διπλωματική, στα πλαίσια της οποίας αναπτύχθηκαν δυο web εφαρμογές για τις επιχειρήσεις και για τον τελικό καταναλωτή. Παράλληλα υπάρχει το Node.JS όπου είναι ασύγχρονο περιβάλλον εκτέλεσης κώδικα JavaScript. Έχει σχεδιαστεί για την δημιουργία επεκτάσιμων εφαρμογών δικτύου. Το 2010 παρουσιάστηκε μαζί με το Node ο διαχειριστής πακέτων NPM. Ο οποίος διευκολύνει τους προγραμματιστές να δημοσιεύουν και να μοιράζονται πηγαίο κώδικα βιβλιοθηκών JavaScript. Στόχος του είναι να απλοποιεί την εγκατάσταση, την ενημέρωση και την απεγκατάσταση των βιβλιοθηκών.

Τέλος, σε γλώσσα JavaScript συντάσσεται όλη η επιχειρηματική λογική για τον ορισμό του δικτύου. Ο chaincode στο Hyperledger Composer έχει έναν διερμηνέα JavaScript που εκτελεί τη λογική για την επεξεργασία της συναλλαγής.

3.2.2 Web Playground

Το εργαλείο Composer Playground παρέχει μια web διεπαφή στον χρήστη για την διαμόρφωση, την εγκατάσταση και τις δοκιμές ενός επιχειρηματικού δικτύου blockchain. Έχει λειτουργίες που επιτρέπουν στους χρήστες να σχεδιάσουν τα επίπεδα ασφάλειας του

δικτύου, να προσκαλούν συμμετέχοντες, να συνδέονται και να προσομοιώνουν πολλά blockchain (βλ. **Εικόνα 12**).



Εικόνα 12 Composer Playground

3.2.3 CLI Utilities

Οι σύγχρονες τάσεις στον κώδικα ανοιχτού λογισμικού δεν θα μπορούσαν να λείπουν από εδώ. Υπάρχει μια πληθώρα εργαλείων και βιβλιοθηκών που συνθέτουν το περιβάλλον ανάπτυξης του blockchain. Τα εργαλεία CLI (command-line interface) έχουν σκοπό την αλληλεπίδραση προγραμμάτων όπου ο χρήστης δίνει εντολές με μορφή διαδοχικών γραμμών κειμένου, όπως σε μια κλασική κονσόλα. Μερικά από τα οποία είναι Yeoman Generator, που είναι γραμμένος σε Node.js, βοηθάει στην παραγωγή κώδικα με βάση κάποιο πρότυπο και τις δυναμικές παραμέτρους, για έλεγχο μονάδων κ.α. Μια σειρά από εργαλεία τα fabric-tools για την ρύθμιση και διαχείριση του Hyperledger Fabric. Τα πακέτα διαχειρίσεις του composer. Το composer-rest-server που είναι η εφαρμογή για το Backend API και τρέχει το Hyperledger Composer LoopBack Connector.

3.2.4 VSCode και Extensions

Το Visual Studio Code είναι ένα ελεύθερο περιβάλλον ανάπτυξης λογισμικού, της Microsoft, το οποίο είναι ανοιχτού κώδικα. Στης εξέλιξή του έχει συμβάλει σημαντικά η

κοινότητα των προγραμματιστών. Ένα δυνατό του σημείο είναι η υποστήριξη επεκτάσεων. Υπάρχει ειδική επέκταση από την Hyperledger για το Composer. Με αυτό τον τρόπο βοηθάει στις καλύτερη και ευκολότερη σύνταξη κώδικα. Προσφέρει επικύρωση, ορισμούς και δομή στον κώδικα. Επίσης αναφέρει τυχόν σφάλματα και εξετάζει να επικυρώσει τα αρχεία που δομούν την ασφάλεια και την πρόσβαση. Έτσι κατά την μοντελοποίηση μπορούν να αποφευχθούν λάθη και να βελτιστοποιηθεί ο χρόνος.

3.2.5 Docker

Το docker είναι ένα πρόγραμμα που πετυχαίνει εικονικότητα σε επίπεδο λειτουργικού συστήματος το οποίο ορίζεται ως containerization. Μέσω του docker δημιουργούνται και συνδέονται όλα τα κομμάτια του Blockchain όπως τα εργαλεία, η βάση δεδομένων CouchDB, ο orderer, η αρχή έκδοσης πιστοποιητικών (certificate authority), οι peers και ο chaincode. Το docker έχει μια σειρά από οφέλη όπως ασφάλεια και απομόνωση των κομματιών. Δίνει τη δυνατότητα για ταχεία και συνεχή ανάπτυξη κώδικα. Τέλος, δίνει ευελιξία στην ανάπτυξη και τις δοκιμές του κάθε τμήματος.

3.2.6 Debugging

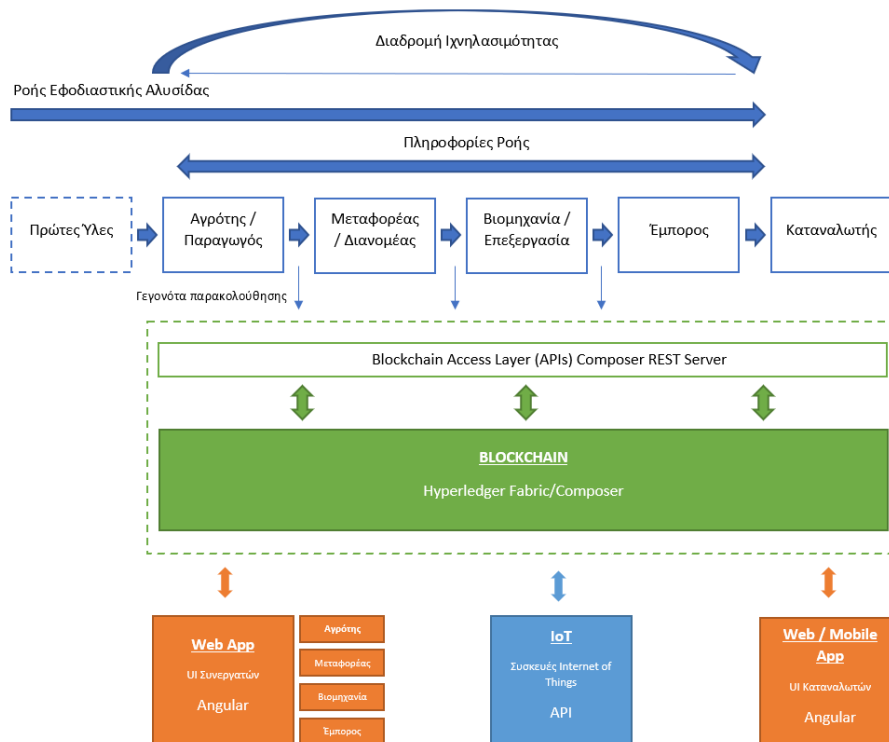
Η διαδικασία του debugging είναι πολύπλοκη, προβληματική και περιορισμένη σε υλοποιήσεις blockchain. Στα έργα δικτύων blockchain η διαχείριση των σφαλμάτων του κώδικα είναι πολύ σημαντική (Stone, 2017). Ένα σφάλμα σε μια συναλλαγή μπορεί να σημαίνει απώλεια περιουσίας για τους συμμετέχοντες, την ακύρωση κάποιας σύμβασης ή το να παγώσουν κάποια εκατομμύρια δολάρια σε κρυπτονομίσματα. Ο κλασικός τρόπος αποσφαλμάτωσης στις περισσότερες περιπτώσεις είναι δοκιμή και σφάλμα, με παρακολούθηση των logs για γραμμένες του κώδικα με εκτύπωση αναφορών, π.χ. τιμές μεταβλητών και καταστάσεων. Ευτυχώς, το Hyperledger Composer σε συνδυασμό με το Visual Studio Code έχουν αναπτύξει μια επικοινωνία για ζωντανή αποσφαλμάτωση με σημεία. Το μόνο που απαιτείται είναι μια λεπτομερής ρύθμιση παραμέτρων για να τρέξει.

ΚΕΦΑΛΑΙΟ 4: Δημιουργία συστήματος αποκεντρωμένης ιχνηλασιμότητας στη βιομηχανία γάλακτος

Σε αυτό το κεφάλαιο θα καταγραφούν λεπτομέρειες που αφορούν τον σκοπό του συστήματος, την προσέγγιση που έγινε, τις κατηγορίες των χρηστών, τις παραδοχές, καθώς και τις λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος. Στη βιομηχανία των τροφίμων διακρίνεται μια δυναμική για το blockchain και επί του παρόντος υπάρχουν λίγες υλοποιήσεις. Σκοπός της παρούσας εργασίας είναι να μελετηθούν τα οφέλη και οι περιορισμοί του. Θα επιχειρηθεί να αναπτυχθεί και να εξεταστεί μια περίπτωση χρήσης στην εφοδιαστική αλυσίδα γάλακτος.

4.1 Στόχοι

Ένα σημαντικό κομμάτι στην διαχείριση της εφοδιαστικής αλυσίδας και στην ασφάλεια των τροφίμων είναι η έγκυρη και έγκαιρη ιχνηλασιμότητα. Επίσης, η εκμετάλλευση, η οργάνωση και ο διαμοιρασμός του όγκου πληροφοριών που υπάρχει, από άκρη σε άκρη, μπορεί να φέρει αξία στις συμβαλλόμενες επιχειρήσεις αλλά και στους καταναλωτές. Με γνώμονα αυτούς τους δύο παράγοντες προτάθηκε μια αποτελεσματική λύση. Σκοπός είναι να δημιουργηθεί ένα περιβάλλον ασφαλές, έμπιστο και αποκεντρωμένο που να εξυπηρετεί την ροή της αλυσίδας (βλ. **Εικόνα 13**). Επίσης, να δίνει με διαφάνεια ανεκμετάλλευτες πληροφορίες και να εγγυάται την ασφάλεια του καταναλωτή.



Εικόνα 13 Αρχιτεκτονική προτεινομένης λύσης

4.2 Προσέγγιση

Υπάρχουν διάφοροι τύποι οντοτήτων σε όλη την ροή της εφοδιαστικής που θα είναι οι κόμβοι του δικτύου blockchain και θα έχουν την δυνατότητα με διαφάνεια να κάνουν τις συναλλαγές και να γράφουν πληροφορίες. Αρχικά, υπάρχουν πέντε διαφορετικούς τύπους χρηστών· τους Αγρότες που είναι οι γαλακτοπαραγωγοί, τους Μεταφορείς γάλακτος με τα βυτία, τις Βιομηχανίες επεξεργασίας γάλακτος, τους Εμπόρους που είναι τα supermarket και τους τελικούς Καταναλωτές. Με βάση αυτές τις ανάγκες και τις απαιτήσεις προτείνεται μια λύση. Η λύση που εφαρμόζεται έχει χαρακτηριστικά αποκέντρωσης, επιλεκτικής ιδιωτικότητας, αμεταβλητότητας και εμπιστοσύνης. Μέσω του ιδιωτικού μας blockchain, εκτελούνται έξυπνα συμβόλαια που διαχειρίζονται αυτόματα τις πληροφορίες. Συγκεντρώνουν και δομούν τα στοιχεία της αλυσίδας, αναδεικνύουν την προέλευση και διαχειρίζονται την ιχνηλασιμότητα. Βάση της φύσης, της δουλειάς, συμπεριλήφθηκαν οι ανάγκες για ασφάλεια των δεδομένων, η διαχείριση

ρίσκου και η δημιουργία ανταγωνιστικού πλεονεκτήματος. Ορίστηκαν κάποιες απαιτήσεις για την λύση ενός διαφανούς συστήματος αποκεντρωμένης ιχνηλασιμότητας στη βιομηχανία γάλακτος:

A1: Κάθε πληροφορία που μπαίνει στο σύστημα να είναι κρυπτογραφημένη, αμετάβλητη, ασφαλής και με δυναμικά δικαιώματα.

A2: Τα εμπλεκόμενα μέρη της εφοδιαστικής να έχουν εικόνα προς τα εμπρός και προς τα πίσω στις συναλλαγές που εμπλέκονται αποκτώντας ιχνηλασιμότητα των περιουσιακών τους στοιχείων.

A3: Οι ιδιοκτήτες των περιουσιακών στοιχείων να έχουν ολοκληρωμένη πληροφόρηση για κάθε στάδιο της εφοδιαστικής αλυσίδας.

A4: Δυνατότητα διασύνδεσης με υπάρχοντα συστήματα των εμπλεκόμενων (ERP, Control Management Systems κ.λπ).

A5: Με στόχο τη μείωση του επιχειρηματικού κινδύνου, ο κάθε συμμετέχων πρέπει να βλέπει τα ίχνη, τις κινήσεις και το ιστορικό από τις συναλλαγές και αυτά που μοιράζεται.

A6: Web και Mobile εφαρμογές για την διαχείριση των διαδικασιών στην πορεία της εφοδιαστικής. Όπως επίσης παρουσίαση για τον καταναλωτή, σε responsive web, με έμφαση στο marketing, προωθώντας την φρεσκάδα, τη διατροφική ασφάλεια, την ποιότητα και την ιστορία γύρω από το τελικό προϊόν.

4.3 Μοντέλο

Η ροή της εφοδιαστικής αλυσίδας του γάλακτος (βλ. **Εικόνα 13**) απαιτεί διαχείριση της πληροφορίας στους τέσσερις βασικούς πυλώνες. Έτσι γίνεται καταγραφή των γεγονότων μέσω κλήσεων στον διακομιστή Composer REST και αποθηκεύονται οι πληροφορίες στο δίκτυο του blockchain (π.χ. υπάρχοντα συστήματα, ERP, IoT συσκευές) αποκεντρωμένες και ασφαλείς. Παράλληλα σε αυτό, λειτουργούν οι εφαρμογές διαχείρισης των συμμετεχόντων, τις οποίες εμπλουτίζουν με στοιχεία των διαδικασιών στην πλατφόρμα.

Τέλος, παρέχεται η εφαρμογή που απευθύνεται στους καταναλωτές για την πληροφόρησή τους σχετικά με τα προϊόντα, ενισχύοντας την διαφάνεια της εφοδιαστικής αλυσίδας γαλακτοκομικών και την ασφάλεια της ποιότητάς τους.

Με βάση τους στόχους, τις απαιτήσεις, την έρευνα και κάποιες συνεντεύξεις από άτομα στον χώρο με πολυετή εμπειρία δομήθηκε το πλαίσιο και η αρχιτεκτονική του συστήματος που υλοποιήθηκε. Καταγράφηκαν οι βασικές ανάγκες των συμβαλλόμενων (βλ. **Πίνακας 3**) και χρησιμοποιήθηκαν για την ανάπτυξη της δομής του μοντέλου, των διαδικασιών και των έξυπνων συμβολαίων. Στον **Πίνακας 3** συνοψίζονται οι βασικές ανάγκες σε κάθε κατηγορία μέλους.

Πίνακας 3 Πλαίσιο βασικών αναγκών συμμετεχόντων

Αγρότης	Μεταφορέας	Βιομηχανία	Έμπορος	Καταναλωτής
Διαχείριση φήμης	Διαχείριση φήμης	Συγκέντρωση δεδομένων	Διαχείριση εγγράφων, Χωρίς χαρτιά και ευελιξία	Ψηφιακή ταυτότητα
Προβολή φρεσκάδας	Καταγραφικά και αποθήκευση θερμοκρασιών (πίεση, υγρασία)	Εξασφάλιση ασφάλειας τροφίμων	Ελαχιστοποίηση χρόνου ιχνηλασιμότητας	Ασφάλεια και υγεία
Αυτοματοποίηση μετρήσεων	Διαχείριση εγγράφων, Χωρίς χαρτιά και ευελιξία	Διαχείριση μετρήσεων και ελέγχων	Συγκέντρωση στοιχείων που μπορούν να βοηθήσουν στην ανάλυση για πρόβλεψη ζήτησης	Γνώση προέλευσης φαγητού και συστατικών
		Γρήγορη διερεύνηση (ιχνηλασιμότητα)		
		Ψηφιακά πιστοποιητικά		
		Ελαχιστοποίηση ανακλήσεων		

ΚΕΦΑΛΑΙΟ 5: Υλοποίηση και Πειραματικά αποτελέσματα

Σε αυτό το κεφάλαιο θα παρουσιαστούν λεπτομέρειες που αφορούν τον σχεδιασμό και την υλοποίηση της αποκεντρωμένης πλατφόρμας. Παρέχονται πληροφορίες σε πραγματικό χρόνο σε όλα τα μέλη της εφοδιαστικής αλυσίδας σχετικά με την κατάσταση των προϊόντων. Μειώνοντας εξαιρετικά τον κίνδυνο των κεντροποιημένων πληροφοριακών συστημάτων, προσφέρει μια ασφαλέστερη, κατανεμημένη, διαφανή και συνεργατική λύση. Ξεκινώντας την υλοποίηση ρυθμίστηκε ένα εικονικό μηχάνημα με λειτουργικό σύστημα Ubuntu Linux και εγκαταστάθηκαν τα προαπαιτούμενα πακέτα, οι βιβλιοθήκες και τα εργαλεία ούτως ώστε να δημιουργηθεί το περιβάλλον ανάπτυξης.

5.1 Δομή

Με βάση την έρευνα και τις απαιτήσεις που ορίστηκαν, σχεδιάστηκε το σχήμα του δικτύου των συμμετεχόντων, των περιουσιακών στοιχείων, των συναλλαγών και των γεγονότων έτσι όπως ορίζεται στην Composer Modeling Language (βλ. πάνω μέρος **Πίνακας 4**). Οι ροές για κάθε τύπο συναλλαγής υλοποιούνται σε ένα API μέσω κώδικα JavaScript. Τα αρχεία ελέγχου πρόσβασης χρησιμοποιούνται για τον περιορισμό των δικαιωμάτων των συμμετεχόντων σε ορισμένους πόρους. Τα εξειδικευμένα και πιο πολύπλοκα ερωτήματα για τα δεδομένα στο blockchain συντάχθηκαν στην γλώσσα ερωτημάτων του Composer, μια γλώσσα τύπου SQL (βλ. κάτω μέρος **Πίνακας 4**).

Πίνακας 4 Composer Modeling Language συστήματος

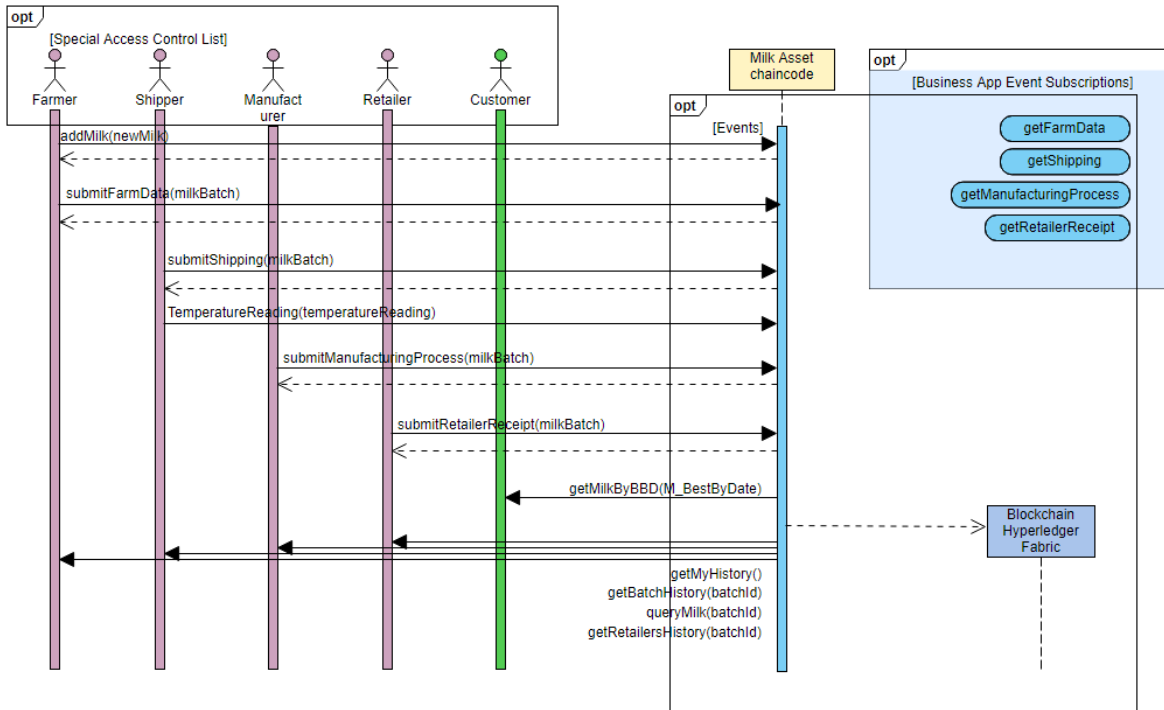
Participants	Assets	Transactions	Events
Farmer	Milk	addMilk	getFarmData
Shipper		transferMilk	getShipping
Manufacturer		submitFarmData	getManufacturingProcess
Retailer		TemperatureReading	getRetailerReceipt
Customer		submitShipping	
		submitManufacturingProcess	
		submitRetailerReceipt	
		CreateDemoParticipants	

Concepts & Enums	Queries	Special Access Control List
State	queryMilk	SystemACL
ProductType	getBatchHistory	NetworkAdminUser
MilkType	getMilkByBBD	NetworkAdminSystem
Address	getRetailersHistory	ReadCommodity
	getMyHistory	CanDelete
		historianAccess

Όλα τα απαραίτητα αρχεία συγκεντρώνονται σε ένα αρχείο, με ορισμό BND (Business Network Definition) και στη συνέχεια εγκαθίσταται στο δίκτυο Fabric.

5.2 Λειτουργίες

Μετά την ανάλυση των απαιτήσεων, της δομής και του μοντέλου που αναφέρονται παραπάνω, σχεδιάστηκε το διάγραμμα γενικών λειτουργιών για κάθε εμπλεκόμενο ρόλο (βλ. **Εικόνα 14**). Αναπτύχθηκαν οι λειτουργίες ανά διαδικασία συμμετέχοντα με κώδικα JavaScript η οποία μεταφράζεται σε chaincode. Υλοποιήθηκαν συμβάντα στα οποία εγγράφεται η εφαρμογή του κάθε συμμετέχοντα για να λαμβάνει αυτόματα τα γεγονότα και να συνεχίζει η ροή των εργασιών. Γράφτηκαν τα ειδικά ερωτήματα στη γλώσσα CQL (Composer Query Language) έτσι ώστε να ανταποκρίνονται στις επιπλέον απαιτήσεις της εφαρμογής.



Εικόνα 14 Διάγραμμα γενικών λειτουργιών συστήματος

Ο chaincode του Fabric πρέπει να παρέχει τις ακόλουθες λειτουργίες:

- Δημιουργία νέου γάλακτος από τους αγρότες με μοναδικό αριθμό παρτίδας (batch number), τον τύπο του γάλακτος, τα λίτρα, το όνομα του αγρότη, τον αριθμό των ζώων που αρμέχθηκαν.
- Μεταφορά γάλακτος στις βιομηχανίες με καταγραφή καιρικών συνθηκών, χιλιομέτρων, αριθμό και ημερομηνία παραστατικού, όχημα.
- Παρακολούθηση θερμοκρασιών γάλακτος κατά την μεταφορά.
- Παραλαβή γάλακτος από τις βιομηχανίες με θερμοκρασία, στοιχεία ελέγχου, ώρα.
- Τυποποίηση προϊόντος με τον τύπο, το lot, την ημερομηνία λήξης και την ποσότητα.
- Παραλαβή από τους εμπόρους με καταγραφή παραστατικών και ποσοτήτων.

- Πρόσβαση επιλεγμένων δεδομένων στον τελικό καταναλωτή με ερώτημα βάση ημερομηνία λήξης (best by date)

5.3 Συμμετέχοντες

Στην περίπτωση αυτή, οι κατηγορίες των ενεργών συμμετεχόντων του δικτύου είναι οι Αγρότες, οι Μεταφορείς, οι Βιομηχανίες και οι Έμποροι. Δημιουργούνται λογαριασμοί χρηστών των εταιρειών της, στην κατηγορία που ανήκουν. Ενώ παράλληλα υπάρχει μια γενική κατηγορία, ο Customer, που θα λειτουργεί απρόσωπα για πολύ συγκεκριμένη πρόσβαση και ερωτήματα. Πιο συγκεκριμένα, η υπηρεσία που θα επιστρέφει δεδομένα στους πελάτες, θα λειτουργεί με αυτή την ταυτότητα χρήση.

Το Hyperledger Composer διαθέτει ένα μητρώο που είναι γεμάτο με ιστορικές συναλλαγές. Κάθε φορά που υποβάλλεται μια συναλλαγή, το HistorianRecord ενημερώνεται και διαχρονικά διατηρεί ιστορικό συναλλαγών εντός ενός επιχειρηματικού δικτύου. Οι συμμετέχοντες και οι ταυτότητες εμπλέκονται και καταγράφονται στην υποβολή των συναλλαγών. Έτσι είτε περιορίζεται η πρόσβαση σε αυτό είτε συντάσσονται ερωτήματα. Η πιο βασική λειτουργία ασφάλειας και πρόσβασης στα δεδομένα ορίστηκε πως τα μέλη θα έχουν πρόσβαση στο συστημικό HistorianRecord για τις συναλλαγές τις οποίες αυτοί καταχώρησαν και συμμετείχαν.

```
rule historianAccess{
  description: "Only allow members to read historian records referencing
transactions they submitted."
  participant(p): "org.foodnet.milk.Business"
  operation: READ
  resource(r): "org.hyperledger.composer.system.HistorianRecord"
  condition: (r.participantInvoking.getIdentifier() == p.getIdentifier())
  action: ALLOW
}
```

Σύνταξη δικαιωμάτων πρόσβασης

5.4 Έννοιες

Κατά την εκτέλεση των συναλλαγών ορίστηκε μια αριθμημένη σήμανση πέντε καταστάσεων (βλ. [Error! Reference source not found.](#)) για το περιουσιακό στοιχείο του γάλακτος. Με αυτή εξασφαλίζει η σωστή λειτουργία του κατά την εκτέλεση των έξυπνων συμβολαίων. Το γάλα, στον κύκλο ζωής του, ξεκινάει από την κατάσταση «συλλογή». Περνάει στην ολοκλήρωση της συλλογής του και είναι «έτοιμο για κατανομή» από τους μεταφορείς. Στη συνέχεια «μεταφέρεται» στις βιομηχανίες για να «επεξεργαστεί» και να τυποποιηθεί στις συσκευασίες πώλησης. Όταν γίνει η παραλαβή των πρώτων συσκευασιών από τους εμπόρους είναι πλέον «έτοιμο» για πώληση. Παράλληλα, ορίζονται τα είδη γάλακτος και οι τύποι προϊόντων παραγωγής.

```
enum State {  
    /**  
     * The state of milk  
     */  
    MILK_COLLECTION  
    READY_FOR_DISTRIBUTION  
    TRANSPORT  
    PROCESSING  
    READY_FOR_SALE  
}
```

Καταστάσεις Milk Asset

5.5 Περιουσιακά στοιχεία

Ο ορισμός των περιουσιακών στοιχείων είναι κομβικός για την ανάπτυξη αυτού του μοντέλου. Ένα περιουσιακό στοιχείο μπορεί να συνδεθεί με κάποιον συμμετέχοντα του δικτύου. Δεν μπορεί να υπάρξει ένα περιουσιακό στοιχείο που δεν θα ανήκει σε κάποιον. Στον σχεδιασμό που αναπτύχθηκε το μοναδικό περιουσιακό στοιχείο είναι η πρώτη ύλη, αυτή του γάλακτος. Για να μπορέσει να επιτευχθεί η διαφάνεια και η ασφάλεια της

πληροφορίας αυτής, θα πρέπει αυτά που αναλύθηκαν παραπάνω να συμπεριληφθούν σε μια ολοκληρωμένη δομή. Στο συγκεντρωτικό πίνακα που ακολουθεί (βλ. Πίνακας 5) περιλαμβάνονται όλες οι μεταβλητές της δομής με τον τύπο του και την περιγραφή του περιεχομένου τους.

Πίνακας 5 Πίνακας μεταβλητών Milk Asset

Μεταβλητή	Τύπος	Περιεχόμενο πεδίου
batchId	String	Αριθμός παρτίδας γάλακτος
milkType	MilkType	Τύπος γάλακτος (αγελαδινό, πρόβειο, κατσικίσιο)
status	State	Κατάσταση περιουσιακού στοιχείου στην ροή
owner	Business	Κάτοχος παρτίδας
F_Timestamp	DateTime	Χρονική στιγμή ολοκλήρωσης συλλογής
F_Name	String	Επωνυμία Φάρμας
F_PersonMilked	String	Άτομο που έκανε την συλλογή
F_AnimalCount	Double	Πλήθος ζώων που αρμέχθηκαν
F_LitersOfMilk	Double	Λίτρα γάλακτος
SH_ShipmentStart	DateTime	Χρονική στιγμή έναρξης μεταφοράς
SH_Plate	String	Πινακίδα βυτιοφόρου
SH_Invoice_No	String	Αριθμός παραστατικού παρτίδας
SH_IssueDate	DateTime	Ημ/νία παραστατικού
SH_WeatherCelsius	Double	Εξωτερική θερμοκρασία την ώρα φόρτωσης
SH_TemperatureReadings	TemperatureReading[]	Πίνακας θερμοκρασιών γάλακτος κατά τη διάρκεια μεταφοράς του
SH_KilometerDistance	Double	Χιλιόμετρα μεταφοράς
M_ReceiptTimestamp	DateTime	Χρονική στιγμή παραλαβής
M_PersonReceipt	String	Άτομο που έκανε παραλαβή
M_Lot	String	Αριθμός LOT παρτίδας μεταποιημένου
M_BestByDate	DateTime	Ημ/νία λήξης
M_ProductType	ProductType	Τύπος προϊόντος
M_Kilograms	Double	Κιλά που χρησιμοποιήθηκαν
M_Liters	Double	Λίτρα που χρησιμοποιήθηκαν
M_Fat	Double	Τιμή πάχους
M_Protein	Double	Τιμή πρωτεΐνης
M_Inhibitors	Double	Τιμή από ανασταλτικούς παράγοντες (αντιβιοτικά κλπ)
M_Temperature	Double	Θερμοκρασία επεξεργασίας
R_Date	DateTime	Ημ/νία έναρξης παραδόσεων

R_PiecesQty	Double	Ποσότητα που έχει παραδοθεί σε σημεία πώλησης (δυναμικό)
--------------------	--------	--

Για την απρόσκοπτη λειτουργία του blockchain είναι σημαντικό να αναφερθεί πως πρέπει να οριστεί σε αυτό το σημείο το πρωτεύον κλειδί που έχει νόημα, στην περίπτωση αυτή είναι ο αριθμός παρτίδας γάλακτος. Παράλληλα πρέπει να οριστούν τα απαραίτητα και τα προαιρετικά πεδία για την δημιουργία νέου στοιχείου.

```
asset Milk identified by batchId {
  /**
   * Standard attributes
   */
  o String batchId
  o MilkType milkType
  o State status

  /**
   * Farm Data Collection
   */
  o DateTime F_Timestamp optional
  o String F_Name optional
  o String F_PersonMilked optional
  o Double F_AnimalCount optional
  o Double F_LitersOfMilk optional

  /**
   * Shipper Details
   */
  o DateTime SH_ShipmentStart optional
  ...
}
```

Απόσπασμα Milk Asset

5.6 Συναλλαγές και επιχειρησιακή λογική

Για την ορθή λειτουργία της εφαρμογής, δόθηκε εξαιρετική προσοχή στο στάδιο ανάπτυξης των έξυπνων συμβολαίων λόγω της σπουδαιότητάς τους και της δυσκολίας ορθής ανάπτυξής τους. Ο πηγαίος κώδικας των συμβολαίων βρίσκεται στο Παράρτημα.

Με βάση τις λειτουργίες που ορίστηκαν παραπάνω υλοποιήθηκαν έξι βασικές συναλλαγές της επιχειρηματικής λογικής που αντιπροσωπεύει η προσέγγιση αυτή. Αρχικά είναι η

έναρξη νέας συλλογής γάλακτος με την δημιουργία της συνάρτησης **addMilk**. Η συναλλαγή αυτή δημιουργεί ένα νέο περιουσιακό στοιχείο με αυτοματοποιημένη αρίθμηση μοναδικής παρτίδας και καταχώρηση του ιδιοκτήτη της. Στη συνέχεια αναπτύχθηκε η συνάρτηση **submitFarmData** με την οποία ολοκληρώνεται η συλλογή γάλακτος, ενημερώνοντας την κατάσταση του asset σε έτοιμο για κατανομή και τα απαραίτητα πεδία που αφορούν την φάρμα έτσι όπως αναλύθηκαν παραπάνω. Η συνάρτηση αυτή μπορεί να εκτελεστεί μόνο από τον ιδιοκτήτη του περιουσιακού στοιχείου αυτής της παρτίδας γάλακτος. Στο τέλος της συνάρτησης ενημερώνεται το στοιχείο και ενεργοποιείται ένα γεγονός (event) που περιλαμβάνει ολόκληρη την πληροφορία. Αυτό το event το εκμεταλλεύεται η πλατφόρμα για να μπορεί σε ζωντανό χρόνο να λάβει και να ενημερώσει την διεπαφή της. Παράλληλα, αυτό το event αποτελεί ισχυρό πλεονέκτημα της πλατφόρμας μιας και οποιοδήποτε άλλο σύστημα, εξωτερικό, εγγραφεί στην ενημέρωση των γεγονότων, με την απαραίτητη εξουσιοδότηση του ιδιοκτήτη, ενεργοποιεί την δυνατότητα περαιτέρω αυτοματοποιήσεων ή αλυσιδωτών λειτουργιών.

```
event getRetailerReceipt {
  --> Business retailer

  o DateTime R_Date optional
  o String R_Invoice_No optional
  o DateTime R_IssueDate optional
  o Double R_PiecesQty optional

  o String batchId
  o Milk asset
}
```

Γεγονός παραλαβής εμπορών getRetailerReceipt

Ακολούθως, στην ίδια λογική αναπτύχθηκαν οι **submitShipping** και **submitManufacturingProcess** με τις πληροφορίες που αφορούν τους μεταφορείς και την βιομηχανία. Αλλάζοντας την ιδιοκτησία του περιουσιακού στοιχείου και ενεργοποιώντας ένα event. Έτσι, επιτυγχάνεται η συγκέντρωση και ο διαμοιρασμός πληροφοριών μεταξύ

των συμμετεχόντων της εφοδιαστικής αλυσίδας με ασφαλή και αξιόπιστο τρόπο, άμεσα προσβάσιμων.

```
async function submitShipping(milkBatch) {
  if (milkBatch.batchId.length <= 0) {
    throw new Error('Please enter the batchId');
  }

  const assetRegistry = await getAssetRegistry('org.foodnet.milk.Milk');

  const exists = await assetRegistry.exists(milkBatch.batchId);

  if (exists) {
    const milk = await assetRegistry.get(milkBatch.batchId);

    milk.owner = milkBatch.shipper;
    milk.status = 'TRANSPORT';
    milk.SH_ShipmentStart = new Date();
    milk.SH_Plate = milkBatch.SH_Plate;
    milk.SH_Invoice_No = milkBatch.SH_Invoice_No;
    milk.SH_IssueDate = milkBatch.SH_IssueDate;
    milk.SH_WeatherCelsius = milkBatch.SH_WeatherCelsius;
    milk.SH_TemperatureReadings = milkBatch.SH_TemperatureReadings;
    milk.SH_KilometerDistance = milkBatch.SH_KilometerDistance;

    ...
  }
}
```

Απόσπασμα συναλλαγής submitShipping

Για να επιτευχθεί η περαιτέρω διαφάνεια της εφοδιαστικής αλυσίδας γάλακτος υλοποιήθηκε η **temperatureReading** με την οποία και καταγράφονται στο asset οι θερμοκρασίες κατά την μεταφορά της πρώτης ύλης στην βιομηχανία. Αυτό συμβαίνει με συσκευές IoT οι οποίες επικοινωνούν ζωντανά με το blockchain διαμέσου αυτής της συνάρτησης. Κατά την ολοκλήρωση της μεταφοράς του γάλακτος υπολογίζεται ο μέσος όρος της θερμοκρασίας μεταφοράς, ενώ υπάρχει και η πλήρης ανάλυσή της. Έτσι ο τελικός καταναλωτής έχει ακόμη μεγαλύτερη αίσθηση της ασφαλούς μετακίνησης άρα και μη αλλοίωσης της πρώτης ύλης. Η τελευταία συνάρτηση είναι η **submitRetailerReceipt** η οποία αναλαμβάνει να συγκεντρώσει τις ποσότητες παράδοσης των προϊόντων στους εμπόρους με τις αντίστοιχες χρονικές στιγμές. Ακόμη, σε αυτό το στάδιο υπάρχει μια ιδιαίτερη χρήση των πληροφοριών. Αν και δεν καταλήγουν όλες στο περιουσιακό στοιχείο,

καταγράφονται στην επιστροφή του event οι πληροφορίες των καταστήματος, αριθμός παραστατικού, ημερομηνία παραστατικού και ποσότητα.

Για την επίτευξη της ιχνηλασιμότητας όλων των βημάτων μέσα στην εφοδιαστική αλυσίδα γάλακτος αναπτύχθηκαν τα ειδικά ερωτήματα **queryMilk** με το οποίο λαμβάνεται ολόκληρο το ιστορικό μιας παρτίδας και **getBatchHistory** με το οποίο λαμβάνονται όλοι οι συμμετέχοντες μιας συγκεκριμένης παρτίδας. Επίσης αναπτύχθηκε το ερώτημα **getRetailersHistory** όπου ένας βιομήχανος μπορεί να πάρει το ιστορικό όλων των παραδόσεων μιας παρτίδας. Σε περίπτωση ανάκλησης σε ελάχιστα δευτερόλεπτα επιστρέφονται οι ποσότητες και τα παραστατικά που εκδόθηκαν.

```
query getMyHistory {
  description: "see all history transactions"
  statement:
    SELECT org.hyperledger.composer.system.HistorianRecord
      WHERE (participantInvoking == _$participantInvoking AND
transactionType == _$transactionType)
}
```

Ειδικό ερώτημα getMyHistory

Με το ερώτημα **getMyHistory** ο κάθε συμμετέχων μπορεί να δει τις συναλλαγές που εμπλεκόταν μαζί με τις πληροφορίες που μετέφεραν. Τέλος, με το ερώτημα **getMilkByBBD** ο καταναλωτής συμπληρώνοντας την ημερομηνία λήξης του προϊόντος, θα μπορεί να λάβει με λεπτομέρεια, διαφάνεια και αξιοπιστία τη ροή του από την φάρμα μέχρι το ράφι.

Το σύστημά μπορεί να βελτιώσει σημαντικά την αποτελεσματικότητα και τη διαφάνεια των γαλακτοκομικών τροφίμων στην εφοδιαστική αλυσίδα, η οποία προφανώς θα ενισχύσει την ασφάλεια των τροφίμων και θα θεμελιώσει την εμπιστοσύνη των καταναλωτών στη βιομηχανία τροφίμων.

ΚΕΦΑΛΑΙΟ 6: Επίλογος

Στο κεφάλαιο αυτό συνοψίζεται η έρευνα που έγινε σχετικά με τις τεχνολογίες του blockchain στην εφοδιαστική αλυσίδα γαλακτοκομικών. Διατυπώνονται τα συμπεράσματα, οι προβληματισμοί και οι περιορισμοί της έρευνας. Τέλος, προτείνονται κάποιες μελλοντικές επέκτασης του συστήματος που υλοποιήθηκε.

6.1 Σύνοψη και συμπεράσματα

Η σύγχρονη τεχνολογία μπορεί να παράσχει τα απαραίτητα εργαλεία για την μετατροπή των λειτουργιών και την κάλυψη των σημερινών αλλά και των μελλοντικών αναγκών της ψηφιακής οικονομίας. Σημαντικές τεχνολογίες πλέον βρίσκονται σε έξυπνες μονάδες παραγωγής και σε τμήματα αλυσίδων εφοδιασμού, όπου οι αισθητήρες και τα χειριστήρια είναι συνδεδεμένα μεταξύ τους. Η αύξηση του όγκου και της ποιότητας των δεδομένων που παράγονται, σε ένα συνδεδεμένο περιβάλλον, έχουν ως αποτέλεσμα να εξελιχθούν οι δυνατότητες ανάλυσης και υπολογισμού για την κατανόηση των δεδομένων.

Στην αλυσίδα εφοδιασμού, οι περισσότερες συναλλαγές συνδέουν διαφορετικά μέρη που δεν έχουν λόγο να εμπιστεύονται το ένα το άλλο και η τεχνολογία Blockchain μπορεί να προσφέρει μια λύση. Την τεχνολογία για την ανταλλαγή πληροφοριών, που επιτρέπει σε πολλαπλά μέρη, των οποίων οι εγγραφές επαληθεύονται και επομένως μπορούν να θεωρηθούν αξιόπιστες, να δημιουργήσουν ένα δημόσιο αρχείο ορατό σε όλους.

Στην παρούσα διπλωματική εργασία έγινε προσπάθεια να αναπτυχθεί μια πλατφόρμα για την ασφαλή, αξιόπιστη και διαφανή ροή πληροφοριών στην εφοδιαστική αλυσίδα γαλακτοκομικών προϊόντων. Ο στόχος της εργασίας ήταν να βρεθούν τρόποι ώστε να επιλυθούν τα ισχύοντα μέχρι σήμερα ζητήματα που ταλαιπωρούσαν μέχρι τώρα καταναλωτές αλλά και παραγωγούς. Η εφαρμογή προσεγγίζει τη λύση από την φάρμα μέχρι το τραπέζι.

Τα βασικά κριτήρια που έπρεπε να καλυφθούν για την πλατφόρμα ενοποίησης και ανταλλαγής πληροφοριών στην εφοδιαστική αλυσίδα γαλακτοκομικών ήταν αρχικά η αποκέντρωση, η διαφάνεια, η ασφάλεια και η αδιαβλητότητα των πληροφοριών. Η ιδανική τεχνολογία γι' αυτή την λύση κρίθηκε το blockchain. Εξετάστηκαν πολλές διαφορετικές τεχνολογίες υλοποίησης blockchain, ώστε να βρεθεί η πιο κατάλληλη. Τελικά, αξιολογήθηκε και επιλέχθηκε η αποκεντρωμένη ιδιωτική πλατφόρμα του Hyperledger Fabric με ανάπτυξη σε Hyperledger Composer. Κομβικό ρόλο για την επιλογή της ήταν μια σειρά από χαρακτηριστικά που υπερτερούσαν από τις υπόλοιπες. Είναι ιδανική για επιχειρησιακές λύσεις, με δικαιώματα χρηστών και φιλικό περιβάλλον ανάπτυξης. Επίσης, είναι η πιο δημοφιλής υλοποίηση σε permissioned blockchain.

Έτσι όπως είναι σχεδιασμένη η αρχιτεκτονική της λύσης αυτής, κάθε χρήσιμη κίνηση, στις διαδικασίες των συμμετεχόντων, καταγράφεται. Οι πληροφορίες είναι αμετάβλητες καθώς κάθε πρότερη καταγραφή κίνησης δεν μπορεί να διαγραφεί. Με αυτόν τον τρόπο διατίθεται όλη η ιστορικότητα των κινήσεων διότι αφήνει πάντοτε το αποτύπωμά του. Χρησιμοποιεί συστήματα που προϋπάρχουν και προσθέτοντας νέες IoT συσκευές και web υπηρεσίες (api services) επιτυγχάνεται ο αυτοματισμός της συλλογής πληροφοριών. Με την χρήση των έξυπνων συμβολαίων υλοποιήθηκε η επιχειρηματική λογική για τον κάθε συμμετέχοντα κρίκο (αγρότες, μεταφορείς, βιομηχανίες και έμποροι). Η εφαρμογή αυτή φέρνει μια σύγχρονη εποχή με την ριζική αλλαγή των σχέσεων μεταξύ των συμμετεχόντων της εφοδιαστικής αλυσίδας γάλακτος. Δεν υπάρχουν αντικρουόμενα συμφέροντα και σχέσεις αμφισβήτησης αλλά σχέσεις συνεργασία και εμπιστοσύνης για τον διαμοιρασμό της πληροφορίας μεταξύ τους. Πλέον το τελικό προϊόν διαθέτει τη σωστή και όσο το δυνατόν ολοκληρωμένη πληροφορία η οποία είναι αποτέλεσμα σύνθεσης τμημάτων αδιάβλητης πληροφόρησης κάθε σταδίου. Ο αριθμός παρτίδας του τελικού προϊόντος (lot) είναι ουσιαστικά ένα ευρετήριο πληροφοριών. Λειτουργεί σαν εργαλείο για οποιαδήποτε χρήση, είτε έκτακτης ανάγκης όπως για παράδειγμα ενδεχόμενη ανάκληση, είτε αναβάθμιση προϊόντος λόγω ποιότητας. Χαρακτηριστική χρήση αυτής της εφαρμογής είναι η πλήρης ιχνηλασιμότητα της προέλευσης του βασικού συστατικού, του τελικού προϊόντος, που είναι το γάλα, μέσα σε λίγα δευτερόλεπτα. Στα συμβατικά συστήματα που

υπάρχουν ο χρόνος και η διαδικασία για την πλήρη ιχνηλασιμότητα μπορεί να διαρκέσει μέρες και εξαρτάται και από το επίπεδο συνεννόησης και πληροφόρησης της κάθε επιχείρησης. Οι στοχοθετημένες ανακλήσεις έχουν σαν αποτέλεσμα την μείωση του κόστους, σε εργατώρες, σπατάλη ή καταστροφή λάθος προϊόντος και χρόνου. Η κάθε επιχείρηση επενδύει, μέσω της εφαρμογής, όχι μόνο στην παροχή ποιοτικού προϊόντος αλλά και στην παροχή μιας ποιοτικότερης συνολικής υπηρεσίας. Επίσης, ο τελικός καταναλωτής αποκτά διαφάνεια όλων των πληροφοριών του γάλακτος. Γνωρίζοντας λεπτομέρειες όπως, ο τόπος συλλογής του, το πλήθος των ζώων, η ώρα συλλογής, οι συνθήκες μεταφοράς, οι εξωτερικές καιρικές θερμοκρασίες, τα χιλιόμετρα που έχει διανύσει, ο χρόνος από την παραλαβή της πρώτης ύλης μέχρι την συσκευασία κ.α. Έτσι γίνεται ουσιαστικά ένας πραγματικός ποιοτικός έλεγχος από τον ίδιο τον καταναλωτή δίνοντας του την δυνατότητα να θέτει μόνος του τα κριτήρια επιλογής του καταλληλότερου, γι' αυτόν, προϊόντος. Από την άλλη πλευρά δίνει τη δυνατότητα στον παραγωγό να επενδύσει στο ίδιο το προϊόν, κάνοντας την ποιότητα του προϊόντος του διαφήμιση, με αποτέλεσμα να μειώνει τις δαπάνες για διαφήμιση. Τέλος, οι επιχειρήσεις που επενδύουν σε νέες τεχνολογίες έχουν ένα σαφές ανταγωνιστικό πλεονέκτημα.

6.2 Όρια και περιορισμοί της έρευνας

Γεγονός είναι πως η τεχνολογία του blockchain είναι μια σχετικά νέα τεχνολογία της οποίας οι εφαρμογές δεν έχουν ακόμη δοκιμαστεί εκτενώς. Ανοίγοντας λοιπόν τον καινούργιο δρόμο σε πολλές υλοποιήσεις, δεν υπάρχουν σαφείς βέλτιστες πρακτικές. Ο ανταγωνισμός των εφαρμογών που εξυπηρετούν την εφοδιαστική αλυσίδα είναι πολύ μεγάλος. Οι επιχειρήσεις έχουν επενδύσει σε δαπανηρές εφαρμογές και λύσεις όπως συστήματα ERP, CRM, WMS, Tracking το οποίο συνεπάγεται την απαίτηση για την ορθή και ολοκληρωμένη κατανόηση, του εκάστοτε συστήματος, της κάθε εταιρείας για να μπορέσει να ενταχθεί με επιτυχία στην πλατφόρμα. Σαφής προϋπόθεση για να είναι εφαρμόσιμες τέτοιες καινοτόμες λύσεις είναι ο εκσυγχρονισμός των υλικοτεχνικών μέσων και υποδομών αλλά και οι γνώσεις των άμεσα εμπλεκόμενων ατόμων. Ακόμα μια

συνισταμένη της πολυπλοκότητας αυτής είναι η ανάγκη για πλήρη απεικόνιση όλων των επιχειρησιακών λειτουργιών και των πολλαπλών παραμέτρων σε κάθε τμήμα της εφοδιαστικής αλυσίδας. Για την ανάλυση όλων αυτών των καταστάσεων απαιτείται αρκετός χρόνος ο οποίος μεταφράζεται σε δαπάνη και κόστος για την επιχείρηση. Ο όγκος των εξόδων για την υλοποίηση και τις δοκιμές της εφαρμογής δεν είναι εύκολα υπολογίσιμος, έτσι αποτελεί έναν αστάθμητο παράγοντα στην συνάρτηση για την απόδοση της επένδυσης (Return of Investment).

6.3 Μελλοντικές επεκτάσεις

Επενδύοντας κάποια επιχείρηση στο blockchain είναι σαν να συντηρεί έναν ζωντανό οργανισμό ο οποίος τροφοδοτείται συνεχώς με τις νέες εξελίξεις. Οι τεχνολογίες πάνω στις οποίες βασίζεται η εφαρμογή παρότι είναι νέες, αλλάζουν ραγδαία. Προτείνονται ορισμένες μελλοντικές επεκτάσεις της πλατφόρμας:

- **Προσθήκη IoT συσκευών.** Αναμένεται με την εφαρμογή των IoT συσκευών στην βιομηχανία να παίξουν καθοριστικό ρόλο αντικαθιστώντας όλα τα προϋπάρχοντα συστήματα.
- **Προσθήκη συστημάτων RFID για περισσότερη αυτοματοποίηση.**
- **Ανάλυση δεδομένων που συγκεντρώνονται.** Φαίνεται να υπάρχει μεγάλη συνοχή με πολλά οφέλη από τον συνδυασμό του blockchain με IoT, AI, Big Data, autonomous robotics. Με την σωστή ανάλυση και τον συνδυασμό αυτών των τεχνολογιών μπορεί να επιτευχθεί βελτίωση της ακρίβειας των προβλέψεων, αυξημένη επίτευξη στόχων των πωλήσεων, πρόβλεψη ζήτησης, να δημιουργηθούν μαθηματικά μοντέλα που θα φέρουν επιπλέον ανταγωνιστικό πλεονέκτημα κ.α.
- **Εξάλειψη χρήσης των web εφαρμογών** για καταχωρήσεις πληροφοριών από ανθρώπους. Πρέπει να είναι όσο το δυνατόν πιο αυστηρές και αυτόματες οι συναλλαγές.

- **Υλοποίηση πιλοτικού σε πραγματικές επιχειρήσεις (Proof-of-Concept).** Να οριστούν ΚΡΙ's για την αξιολόγηση της πλατφόρμας του blockchain. Το μεγαλύτερο πλεονέκτημα θα υπάρξει αν υιοθετηθεί από πολλούς οργανισμούς και εταιρείες για τις πολύπλοκες αλληλεπιδράσεις τους.
- **Ενσωμάτωση περισσότερων ποιοτικών στοιχείων** στα έξυπνα συμβόλαια του blockchain για διενέργεια αυτοματοποιημένων ποιοτικών ελέγχων. Μακροπρόθεσμα η χρήση του blockchain θα αποτελεί ποιοτική σφραγίδα του προϊόντος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Acheson, N., 2018. *How Bitcoin Mining Works*. [Ηλεκτρονικό]
Available at: <https://www.coindesk.com/information/how-bitcoin-mining-works>
- Anon., 2005. *International Standards Organization 9000*. s.l.:s.n.
- Anon., χ.χ. ΚΑΤΑΓΡΑΦΗ ΣΤΟΙΧΕΪΩΝ ΠΑΡΑΓΩΓΗΣ ΝΩΠΟΥ ΓΑΛΑΚΤΟΣ. [Ηλεκτρονικό]
Available at: <http://www.elgo.gr/index.php/quality-assurance-of-agricultural-products/check-milk?lang=el>
- Antonopoulos, A. M., 2017. *Mastering Bitcoin: Programming the Open Blockchain*. s.l.:O'Reilly.
- Aston, B., 2015. *A brief history of JavaScript*. [Ηλεκτρονικό]
Available at: <https://medium.com/@benastontweet/lesson-1a-the-history-of-javascript-8c1ce3bffb17>
[Πρόσβαση 2019].
- Aung, M. M. & Chang, Y. S., 2014. Traceability in a food supply chain: Safety and quality perspectives. Τόμος 39, pp. 172-184.
- Bauerle, N., 2017. *What is a Distributed Ledger?*. [Ηλεκτρονικό]
Available at: <https://www.coindesk.com/information/what-is-a-distributed-ledger>
- Bauerle, N., 2017. *What is the Difference Between Public and Permissioned Blockchains?*. [Ηλεκτρονικό]
Available at: <https://www.coindesk.com/information/what-is-the-difference-between-open-and-permissioned-blockchains>
[Πρόσβαση 2019].
- Bechini, A., Cimino, M. G., Marcelloni, F. & Tomasi, A., 2007. Patterns and technologies for enabling supply chain traceability through collaborative e-business. *Information and Software Technology*, 50(4), pp. 342-359.
- Buterin, V., 2013. *Ethereum White Paper*. [Ηλεκτρονικό]
Available at:
https://web.archive.org/web/20161021061647/https://www.weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf
- Cachin, C., 2016. Architecture of the Hyperledger Blockchain Fabric. p.
https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf.
- Catalini, C. & Gans, J. S., 2017. Some Simple Economics of the Blockchain.
- Church, Z., 2017. *Blockchain, explained*. [Ηλεκτρονικό]
Available at: <https://mitsloan.mit.edu/ideas-made-to-matter/blockchain-explained>

El-Hofi, M., El-Tanboly, E.-S. & Ismail, A., 2010. IMPLEMENTATION OF THE HAZARD ANALYSIS CRITICAL CONTROL POINT (HACCP) SYSTEM TO UF WHITE CHEESE PRODUCTION LINE.

European Commission, χ.χ. *RASFF - Food and Feed Safety Alerts*. [Ηλεκτρονικό]
Available at: https://ec.europa.eu/food/safety/rasff_en
[Πρόσβαση 2019].

Fiorillo, S., 2018. *Bitcoin History: Timeline, Origins and Founder*. [Ηλεκτρονικό]
Available at: <https://www.thestreet.com/investing/bitcoin/bitcoin-history-14686578>

Food Marketing Research and Information Center, 2008. *Handbook for Introduction of Food Traceability Systems*. s.l.:s.n.

Fredriksson, A. & Liljestrand, K., 2015. Capturing food logistics: a literature review and research agenda. *A Leading Journal of Supply Chain Management* , 18(1), pp. 16-34.

Friebe, T., 2017. *Bitcoin, Ethereum, and Hyperledger Fabric — which one wins?*. [Ηλεκτρονικό]
Available at: <https://medium.com/blockchainspace/3-comparison-of-bitcoin-ethereum-and-hyperledger-fabric-cd48810e590c>

Galvin, D., 2018. *IBM and Walmart: Blockchain for Food Safety*, s.l.: s.n.

Gaur, N. και συν., 2018. *Hands-on blockchain with Hyperledger*. s.l.:Packt.

Gutierrez, C., 2017. *Blockchain at Walmart: Tracking Food from Farm to Fork*. [Ηλεκτρονικό]
Available at: <https://www.altoros.com/blog/blockchain-at-walmart-tracking-food-from-farm-to-fork/>

Huertas, J., Liu, H. & Robinson, S., 2018. *Eximchain Whitepaper: Supply Chain Finance solutions on a secured public, permissioned blockchain hybrid*. [Ηλεκτρονικό]
Available at: <https://www.eximchain.com/documents>

Hyperledger Fabric Docs, 2018. *Smart Contracts*. [Ηλεκτρονικό]
Available at: <https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html?highlight=smart%20contracts#smart-contracts>

Hyperledger Foundation, 2019. *Hyperledger Composer Intro*. [Ηλεκτρονικό]
Available at: <https://hyperledger.github.io/composer/latest/introduction/introduction>

IBM, 2017. *Maersk and IBM Unveil First Industry-Wide Cross-Border Supply Chain Solution on Blockchain*. [Ηλεκτρονικό]
Available at: <https://www-03.ibm.com/press/us/en/pressrelease/51712.wss>

IBM, 2017. *Major Blockchain Collaboration*. [Ηλεκτρονικό]
Available at: <https://www.prnewswire.com/news-releases/ibm-announces-major-blockchain-collaboration-with-dole-driscolls-golden-state-foods-kroger-mccormick-and-company-mclane-company-nestle-tyson-foods-unilever-and-walmart-to-address-food-safety-worldwide-300507604.htm>

- IBM, 2017. *Make your blockchain smart contracts smarter with business rules*. [Ηλεκτρονικό]
Available at: <https://www.ibm.com/developerworks/library/mw-1708-mery-blockchain/1708-mery.html>
- IBM, 2018. *Topology of a blockchain network with Hyperledger Fabric, Hyperledger Composer, and IBM ODM*. [Ηλεκτρονικό]
Available at: <https://www.ibm.com/developerworks/library/mw-1708-mery-blockchain/1708-mery.html>
- ICODROPS, 2018. *EximChain Token Sale Info*. [Ηλεκτρονικό]
Available at: <https://icodrops.com/eximchain/>
[Πρόσβαση 2019].
- IFIS, 2016. *Food Fraud Costs the Global Food Industry \$10-15 Billion Annually*. [Ηλεκτρονικό]
Available at: <https://blog.ifis.org/global-food-fraud>
- International Bank for Reconstruction and Development, 2017. *Distributed Ledger Technology (DLT) and Blockchain*, s.l.: FinTech Note, No. 1.
- International Union of FOOD SCIENCE and Technology, 2012. *Food Traceability*. [Ηλεκτρονικό]
Available at: <https://www.yumpu.com/en/document/read/39546561/iufost-scientific-information-bulletin-sib-food-traceability>
- Kamath, R., 2018. Food Traceability on Blockchain: Walmart's Pork and Mango Pilots. *The Journal of The British Blockchain Association*, 1(1), pp. 1-12.
- Larson, P. D. & Rogers, D. S., 1998. Supply Chain Management: Definition, Growth and Approaches. *Supply Chain Management*, 6(4), pp. 1-5.
- Li, X. και συν., 2017. A survey on the security of blockchain systems. *Future Generation Computer Systems*.
- Manzini, R. & Accorsi, R., 2013. The new conceptual framework for food supply chain assessment. *Journal of Food Engineering*, 115(2), pp. 251-263.
- Meunier, S., 2016. *Blockchain technology - a very special kind of Distributed Database*. [Ηλεκτρονικό]
Available at: <https://medium.com/@sbmeunier/blockchain-technology-a-very-special-kind-of-distributed-database-e63d00781118>
[Πρόσβαση 2019].
- Nakamoto, S., 2008. *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Ηλεκτρονικό]
Available at: <https://bitcoin.org/bitcoin.pdf>
- Nakamoto, S., 2009. [Ηλεκτρονικό]
Available at: <http://p2pfoundation.ning.com/forum/topics/bitcoin-open-source>
- Netscape Communications Corporation, 1995. *NETSCAPE AND SUN ANNOUNCE JAVASCRIPT, THE OPEN, CROSS-PLATFORM OBJECT SCRIPTING LANGUAGE FOR ENTERPRISE NETWORKS AND THE INTERNET*. [Ηλεκτρονικό]

Available at:

<https://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsreleas e67.html>

Newswire, C. P., 2018. *The Global Food Traceability Market*. [Ηλεκτρονικό]

Available at: <https://www.prnewswire.com/news-releases/the-global-food-traceability-market-is-accounted-for-11-25-billion-in-2017-and-is-expected-to-reach-20-95-billion-by-2026-growing-at-a-cagr-of-7-15-during-the-forecast-period-300707133.html>

O'Neal, S., 2018. *Central Bank-Issued Digital Currencies: Why Governments May (or May Not) Need Them*. [Ηλεκτρονικό]

Available at: <https://cointelegraph.com/news/central-bank-issued-digital-currencies-why-governments-may-or-may-not-need-them>

Opara, U. L., 2003. Traceability in agriculture and food supply chain: A review of basic concepts, technological implications, and future prospects. *European Journal of Operational Research*.

Pant, R., Prakash, G. & Farooque, J. A., 2015. A Framework for Traceability and Transparency in the Dairy Supply Chain Networks. *Procedia - Social and Behavioral Sciences*, Τόμος 189, pp. 385-394.

Rilee, K., 2018. *Understanding Hyperledger Fabric — Byzantine Fault Tolerance*. [Ηλεκτρονικό]

Available at: <https://medium.com/kokster/understanding-hyperledger-fabric-byzantine-fault-tolerance-cf106146ef43>

Sari, K., 2010. Exploring the impacts of radio frequency identification (RFID) technology on supply chain performance. *European Journal of Operational Research*, Issue 207, pp. 174-183.

Schollmeier, R., 2002. A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications.

Schwartz, D., Youngs, N. & Britto, A., 2014. *The Ripple Protocol Consensus Algorithm*.

[Ηλεκτρονικό]

Available at: https://ripple.com/files/ripple_consensus_whitepaper.pdf

Stone, S., 2017. *Debug your Blockchain business network using Hyperledger Composer*.

[Ηλεκτρονικό]

Available at: <https://medium.com/@mrsimonstone/debug-your-blockchain-business-network-using-hyperledger-composer-9bea20b49a74>

Szabo, N., 1996. *Smart Contracts: Building Blocks for Digital Markets*. s.l.:s.n.

Szabo, N., 1997. *Formalizing and Securing Relationships on Public Networks*. s.l.:s.n.

Techrock, χ.χ. *Taelpay Website*. [Ηλεκτρονικό]

Available at: <https://taelpay.com/>

Terzi, S. & Stamelos, I., 2018. Permissioned Blockchains and Smart Contracts into Agile Software Processes. *Communications in Computer and Information Science*, Τόμος 918.

- The Economist, 2015. *Who is Satoshi Nakamoto?*. [Ηλεκτρονικό]
Available at: <https://www.economist.com/the-economist-explains/2015/11/02/who-is-satoshi-nakamoto>
- The Linux Foundation, 2018. *Blockchain for Business - An Introduction to Hyperledger Technologies*. [Ηλεκτρονικό]
Available at: <https://www.edx.org/course/blockchain-for-business-an-introduction-to-hyperledger-technologies>
- theverge.com, 2019. *The Samsung Galaxy S10 has a cryptocurrency wallet built in*. [Ηλεκτρονικό]
Available at: <https://www.theverge.com/2019/2/25/18233131/samsung-galaxy-s10-bitcoin-cryptocurrency-wallet-features>
- UK Office for Science, 2016. *Distributed Ledger Technology: beyond block chain*. [Ηλεκτρονικό]
Available at:
https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distributed-ledger-technology.pdf
- WaltonChain, 2018. *WaltonChain White Paper v2*. [Ηλεκτρονικό]
Available at: <https://www.waltonchain.org/en/Uploads/2018-12-10/5c0e41aa64a69.pdf>
[Πρόσβαση 2019].
- Wikipedia, 2008. *2008 Chinese milk scandal*. [Ηλεκτρονικό]
Available at: https://en.wikipedia.org/wiki/2008_Chinese_milk_scandal
[Πρόσβαση 2019].
- Wu, H. και συν., 2017. A Distributed Ledger for Supply Chain Physical. *Information*.
- Zhang, J. & Bhatt, T., 2014. A Guidance Document on the Best Practices in Food Traceability. *Food Science and Food Safety*, 13(5).
- ΕΦΕΤ, 2012. *ΓΕΝΙΚΟΣ ΟΔΗΓΟΣ ΓΙΑ ΤΗΝ ΕΦΑΡΜΟΓΗ ΣΥΣΤΗΜΑΤΟΣ ΒΑΣΕΙ ΤΩΝ ΑΡΧΩΝ ΤΟΥ HACCP ΣΕ ΜΙΚΡΕΣ ΓΑΛΑΚΤΟΚΟΜΙΚΕΣ ΕΠΙΧΕΙΡΗΣΕΙΣ*, s.l.: s.n.
- Μανιφάβα, Δ., 2018. «Πόλεμος» γαλακτοβιομηχανιών για την αγορά του γιαουρτιού. [Ηλεκτρονικό]
Available at: <http://www.kathimerini.gr/981043/article/oikonomia/epixeirhseis/polemos-galaktoviomhxiwn-gia-thn-agera-toy-giaourtiou>
- Σκούφου, Δ., 2018. Έντονη κινητικότητα σε όλα τα πεδία της Γαλακτοβιομηχανίας. *σελφ σέρβις*, 8, Issue 485, pp. 38-40.

ΠΑΡΑΡΤΗΜΑ Ι: Εγκατάσταση και Ανάπτυξη

Στο παράρτημα αυτό θα γίνει παρουσίαση οδηγιών για την προετοιμασία και την εγκατάσταση του συστήματος σε Ubuntu Linux ούτως ώστε να γίνει η χρήση του. Παρουσιάζονται αναλυτικά τα προγράμματα που χρειάστηκαν. Αναλύονται κάποια από τα βήματα που εκτελέστηκαν τακτικά κατά την ανάπτυξη της εφαρμογής και του συστήματος. Τα στάδια πραγματοποιήθηκαν με επιτυχή εγκατάσταση και εκτέλεση της εφαρμογής σε Oracle VM VirtualBox v5.2.8 με λειτουργικό σύστημα Ubuntu Linux 18.04 LTS.

Βήμα 1

Εγκαταστάθηκαν τα πακέτα, τα κλειδιά και τα repositories που χρειάζονται για την σωστή ανάπτυξη και χρήση του συστήματος αφού έγινε πρώτα μια ενημέρωση του συστήματος. Οι εντολές που εκτελέστηκαν στην κονσόλα ήταν για την εγκατάσταση των: Docker, NodeJS, NPM, GoLang, CURL και κάποιον ακόμα πακέτων.

```
sudo apt-get update
sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
# Docker
sudo apt-get install docker-ce
sudo groupadd docker
sudo usermod -aG docker $USER
docker run hello-world
sudo apt install docker-compose

# Node.JS
sudo apt-get install -y nodejs
sudo apt-get install -y build-essentials

# NPM
sudo apt install aptitude
sudo snap install node --classic --channel 9/stable
sudo aptitude install npm
sudo apt update
```



```
# Go
sudo curl -O https://storage.googleapis.com/golang/go1.10.2.linux-amd64.tar.gz
sudo tar -xvf go1.10.2.linux-amd64.tar.gz
sudo mv go /usr/local
echo 'export PATH=$PATH:/usr/local/go/bin' >> ~/.profile
source ~/.profile
go version
```

Βήμα 2

Έγινε έλεγχος των εκδόσεων που εγκαταστάθηκαν

```
docker --version
Docker version 18.03.1-ce, build 9ee9f40

docker-compose --version
docker-compose version 1.17.1, build unknown

node --version
v8.9.0

npm --version
5.6.0

git --version
git version 2.17.1

python --version
Python 2.7.15rc1
```

Βήμα 3

Διαμέσου του NPM έγινε η global εγκατάσταση των απαραίτητων βιβλιοθηκών για την προετοιμασία των εργαλείων του Hyperledger Fabric και Hyperledger Composer.

```
npm i -g composer-cli

npm i -g generator-hyperledger-composer

npm i -g composer-rest-server

npm install -g composer-playground
```

```
# YeoMan (current @2.0.5)
npm i -g yo
```

Βήμα 4

Έλεγχος ο οποίος γίνεται τακτικά κατά την ανάπτυξη της εφαρμογής σχετικά με τα ανοιχτά sessions του docker στο μηχάνημα. Πρέπει να μην είναι ενεργό κάποιο παλιό container κατά την έναρξη νέου.

```
docker kill $(docker ps -q)
docker rm $(docker ps -aq)
docker rmi $(docker images dev-* -q)
clear
```

Βήμα 5

Έγινε η εγκατάσταση του Visual Studio Code IDE στον οποίο αναπτύχθηκε ο κώδικας (<https://code.visualstudio.com/download>) και των επεκτάσεων από το Extensions Marketplace του για το Hyperledger Composer.

Βήμα 6

Εγκαταστάθηκε το Hyperledger Fabric τοπικά και των εργαλείων του, για να μπορέσει να δημοσιευτεί το δίκτυο του blockchain και να δοκιμαστεί. Με τα εργαλεία που κατεβαίνουν υπάρχουν προγράμματα που κατεβάζουν, εκκινούν και δημιουργούν απαραίτητες ρυθμίσεις για την λειτουργία του Fabric.

```
mkdir fabric-tools
cd fabric-tools
curl -O https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers/fabric-dev-servers.zip --output fabric-dev-servers.zip

unzip fabric-dev-servers.zip
./downloadFabric.sh
```

```
./startFabric.sh  
./createPeerAdminCard.sh
```

Βήμα 7

Σε αυτό το σημείο έγινε η διάρθρωση των φακέλων και των αρχείων του project «food-network» ορίστηκαν δυο υπο-κατάλογοι για την ανάπτυξη, ο «blockchain» και ο «business-app». Στον «blockchain» εγκαταστάθηκε το template της Hyperledger Composer το Perishables Goods Network. Στον «business-app» έγινε ανάπτυξη των angular εφαρμογών με τη βοήθεια του angular-cli. Ρυθμίστηκαν τα package.json για τον εκάστοτε κατάλογο και επιλέχθηκε το bitbucket για version control όλου του project με μικρές ρυθμίσεις (gitignore).

```
git clone https://github.com/hyperledger/composer-sample-networks.git  
cp -r composer-sample-networks/packages/perishable-network/ food-network/blockchain  
  
npm install -g @angular/cli@1.4
```

Στο δίκτυο του blockchain προετοιμάζεται και δημιουργεί το πρώτο network με την εντολή *npm install* η οποία και θα εκτελέσει μια σειρά από εντολές

```
mkdirp ./dist && composer archive create --sourceType dir --sourceName . -a  
./dist/food-network.bna
```

Βήμα 8

Εφόσον η ανάπτυξη είναι στο επιθυμητό επίπεδο θα δημιουργηθεί μια νέα έκδοση του δικτύου.

```
composer archive create -a dist/food-network@0.3.3.bna --sourceType dir --  
sourceName .
```

Βήμα 9

Γίνεται εγκατάσταση και έναρξη του δικτύου

```
composer network install --card PeerAdmin@hlfv1 --archiveFile food-network@0.3.3.bna  
  
composer network start --networkName food-network --networkVersion 0.3.2 --  
networkAdmin admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 -  
-file networkadmin.card
```

ή μπορεί γίνεται η αναβάθμιση σε νέα έκδοση του δικτύου

```
composer network upgrade --networkName food-network --networkVersion 0.3.3 --  
card PeerAdmin@hlfv1
```

Βήμα 10

Γίνεται η εισαγωγή της ταυτότητας του διαχειριστή δικτύου ως μια χρησιμοποιήσιμη κάρτα του δικτύου μας και ελέγχεται ότι δουλεύει.

```
composer card import --file networkadmin.card  
composer network ping --card admin@food-network
```

Βήμα 11

Για να επικοινωνήσει το δίκτυο μπορεί είτε να χρησιμοποιηθεί ο Composer SDK είτε να παραχθεί ο REST API server με το εργαλείο composer-rest-server. Παράγεται ο REST Server και δοκιμάζεται τοπικά στη <http://localhost:3000> για την επικοινωνία με το blockchain

```
user@blockchain-vm:~/food-network$ composer-rest-server  
? Enter the name of the business network card to use: admin@food-network  
? Specify if you want namespaces in the generated REST API: never use namespaces  
? Specify if you want to use an API key to secure the REST API: No  
? Specify if you want to enable authentication for the REST API using Passport: No  
? Specify if you want to enable the explorer test interface: Yes  
? Specify a key if you want to enable dynamic logging: I
```

? Specify if you want to enable event publication over WebSockets: **Yes**
 ? Specify if you want to enable TLS security for the REST API: **No**

The image displays two screenshots of the Hyperledger Composer REST server interface. The left screenshot shows the 'addMilk' transaction details, including endpoints for GET /addMilk, POST /addMilk, and GET /addMilk/{id}. The right screenshot shows the 'submitFarmData' transaction details, including a successful response with a JSON object containing fields like 'F.Timestamp', 'F.Name', 'F.PersonLiked', 'F.AdminCount', 'F.LitersOfMilk', 'batchId', 'transactionId', and 'timestamp'.

Βήμα 12

Σε μια νέα κονσόλα γίνεται έναρξη της υπηρεσίας του τοπικού Hyperledger Composer Playground (composer-playground) με προσπέλαση στο <http://localhost:8080>. Να σημειωθεί πως υπάρχει και online έκδοση αυτού του εργαλείου που μπορεί να βοηθήσει στην ανάπτυξη και δοκιμή ενός δικτύου στη διεύθυνση: <https://composer-playground.mybluemix.net>

Βήμα 13

Με την χρήση του yeoman generator και της εντολής «yo hyperledger-composer:angular» μπορεί να δημιουργηθεί ένας σκελετός από κώδικα επικοινωνίας angular με το blockchain.

Βήμα 14

Εκτελείται ο κώδικας των web εφαρμογών με την εντολή «npm run start» και γίνεται η προσπέλαση <http://localhost:4200>.

Βήμα 15

Για το debugging χρησιμοποιήθηκαν δυο τεχνικές. Είτε μέσα από την προσπέλαση των logs του docker είτε μέσω των debug points κατά την ανάπτυξη στο visual studio code.

```
docker logs dev-peer0.org1.example.com-food-network-0.2.6-deploy.6 >&1 | grep waiting  
docker logs peer0.org1.example.com 2>&1 | grep 'err'
```

ΠΑΡΑΡΤΗΜΑ ΙΙ: Κώδικας

Στο παράρτημα αυτό βρίσκεται ο κώδικας του chaincode για τις λειτουργίες των έξυπνων συμβολαίων που αναπτύχθηκαν στα πλαίσια της παρούσας διπλωματικής. Τα ειδικά ερωτήματα στη γλώσσα Composer Query Language, ο κώδικας εγγραφής των δικαιωμάτων των συμμετεχόντων για τους πόρους του συστήματος όπως ορίζεται στην Composer Access Control Language. Τέλος, γίνεται μια αναφορά στη δομή των web εφαρμογών με παραδείγματα της Angular.

A. Chaincode Logic

Σε αυτή την ενότητα βρίσκεται ο πηγαίος κώδικας όλων των επιχειρησιακών διαδικασιών και λειτουργιών που αναπτύχθηκαν για τις συναλλαγές που γίνονται με το Blockchain.

```
/**
 * A shipment has been received by an importer
 * @param {org.foodnet.milk.addMilk} shipmentReceived - the addMilk transaction
 * @transaction
 */
async function addMilk(newMilk) {
  // eslint-disable-line no-unused-vars
  const participantRegistry = await getParticipantRegistry(
    'org.foodnet.milk.Farmer'
  );
  var NS = 'org.foodnet.milk';
  var milk = getFactory().newResource(
    NS,
    'Milk',
    Math.random()
      .toString(36)
      .substring(3)
  );
  milk.milkType = newMilk.milkType;
  milk.owner = newMilk.farmer;
  milk.status = 'MILK_COLLECTION';

  const assetRegistry = await getAssetRegistry('org.foodnet.milk.Milk');
  await assetRegistry.add(milk);
  await participantRegistry.update(newMilk.farmer);
}

/**
 * Submit Farm data after milk Collection process.
 * @param {org.foodnet.milk.submitFarmData} milkBatch - the milk collection ends.
 * @transaction
 */
```

```

*/
async function submitFarmData(milkBatch) {
  if (milkBatch.batchId.length <= 0) {
    throw new Error('Please enter the batchId');
  }

  const assetRegistry = await getAssetRegistry('org.foodnet.milk.Milk');

  const exists = await assetRegistry.exists(milkBatch.batchId);

  if (exists) {
    const milk = await assetRegistry.get(milkBatch.batchId);

    // milk.owner = milkBatch.farmer;
    milk.status = 'READY_FOR_DISTRIBUTION';
    milk.F_Timestamp = new Date();
    milk.F_Name = milkBatch.F_Name;
    milk.F_PersonMilked = milkBatch.F_PersonMilked;
    milk.F_AnimalCount = milkBatch.F_AnimalCount;
    milk.F_LitersOfMilk = milkBatch.F_LitersOfMilk;

    // Create and emit a regulation event
    var event = getFactory().newEvent('org.foodnet.milk', 'getFarmData');
    event.batchId = milkBatch.batchId;
    event.F_Timestamp = milkBatch.F_Timestamp;
    event.F_Name = milkBatch.F_Name;
    event.F_PersonMilked = milkBatch.F_PersonMilked;
    event.F_AnimalCount = milkBatch.F_AnimalCount;
    event.F_LitersOfMilk = milkBatch.F_LitersOfMilk;

    emit(event);

    // publish update
    await assetRegistry.update(milk);
  } else {
    throw new Error('the batch you specified does not exist!');
  }
}

/**
 * Submit packing details for Shipping the Milk.
 * @param {org.foodnet.milk.submitShipping} milkBatch - the batch we are transfer
 * @transaction
 */
async function submitShipping(milkBatch) {
  if (milkBatch.batchId.length <= 0) {
    throw new Error('Please enter the batchId');
  }

  const assetRegistry = await getAssetRegistry('org.foodnet.milk.Milk');

  const exists = await assetRegistry.exists(milkBatch.batchId);

  if (exists) {
    const milk = await assetRegistry.get(milkBatch.batchId);

    milk.owner = milkBatch.shipper;
    milk.status = 'TRANSPORT';
    milk.SH_ShipmentStart = new Date();
    milk.SH_Plate = milkBatch.SH_Plate;
    milk.SH_Invoice_No = milkBatch.SH_Invoice_No;
  }
}

```



```

milk.SH_IssueDate = milkBatch.SH_IssueDate;
milk.SH_WeatherCelsius = milkBatch.SH_WeatherCelsius;
milk.SH_TemperatureReadings = milkBatch.SH_TemperatureReadings;
milk.SH_KilometerDistance = milkBatch.SH_KilometerDistance;

// Create and emit a regulation event
var event = getFactory().newEvent('org.foodnet.milk', 'getShipping');
event.batchId = milkBatch.batchId;
event.farmer = milkBatch.farmer;
event.shipper = milkBatch.shipper;
event.SH_ShipmentStart = milkBatch.SH_ShipmentStart;
event.SH_Plate = milkBatch.SH_Plate;
event.SH_Invoice_No = milkBatch.SH_Invoice_No;
event.SH_IssueDate = milkBatch.SH_IssueDate;
event.SH_WeatherCelsius = milkBatch.SH_WeatherCelsius;
event.SH_TemperatureReadings = milkBatch.SH_TemperatureReadings;
event.SH_KilometerDistance = milkBatch.SH_KilometerDistance;

emit(event);

var participantRegistry = await getParticipantRegistry(
  'org.foodnet.milk.Shipper'
);
await participantRegistry.update(milkBatch.shipper);
//update ownership
milk.owner = milkBatch.shipper;
participantRegistry = await getParticipantRegistry(
  'org.foodnet.milk.Shipper'
);
await participantRegistry.update(milkBatch.shipper);

// publish update
await assetRegistry.update(milk);
} else {
  throw new Error('the batch you specified does not exist!');
}
}
}

/**
 * A temperature reading has been received for a shipment
 * @param {org.foodnet.milk.TemperatureReading} temperatureReading - the
TemperatureReading transaction
 * @transaction
 */
async function temperatureReading(temperatureReading) {
  if (temperatureReading.batchId.length <= 0) {
    throw new Error('Please enter the batchId');
  }
}

const assetRegistry = await getAssetRegistry('org.foodnet.milk.Milk');

const exists = await assetRegistry.exists(temperatureReading.batchId);

if (exists) {
  const milk = await assetRegistry.get(temperatureReading.batchId);

  console.log(
    'Adding temperature ' +
    temperatureReading.centigrade +
    ' to shipment ' +
    milk.$identifier +

```

```

        ' with batchId ' +
        milk.batchId
    );

    if (milk.SH_TemperatureReadings) {
        milk.SH_TemperatureReadings.push(temperatureReading);
    } else {
        milk.SH_TemperatureReadings = [temperatureReading];
    }

    await assetRegistry.update(milk);
} else {
    throw new Error('the batch you specified does not exist!');
}
}

/**
 * Get data about Manufacturing Process.
 * @param {org.foodnet.milk.submitManufacturingProcess} milkBatch - milk batch
 * @transaction
 */
async function submitManufacturingProcess(milkBatch) {
    if (milkBatch.batchId.length <= 0) {
        throw new Error('Please enter the batchId');
    }

    const assetRegistry = await getAssetRegistry('org.foodnet.milk.Milk');

    const exists = await assetRegistry.exists(milkBatch.batchId);

    if (exists) {
        const milk = await assetRegistry.get(milkBatch.batchId);

        milk.M_ReceiptTimestamp = new Date();
        milk.status = 'PROCESSING';
        milk.M_PersonReceipt = milkBatch.M_PersonReceipt;
        milk.M_Lot = milkBatch.M_Lot;
        milk.M_BestByDate = milkBatch.M_BestByDate;
        milk.M_ProductType = milkBatch.M_ProductType;
        milk.M_Kilograms = milkBatch.M_Kilograms;
        milk.M_Liters = milkBatch.M_Liters;
        milk.M_Fat = milkBatch.M_Fat;
        milk.M_Protein = milkBatch.M_Protein;
        milk.M_Inhibitors = milkBatch.M_Inhibitors;
        milk.M_Temperature = milkBatch.M_Temperature;

        // Create and emit a regulation event
        var event = getFactory().newEvent(
            'org.foodnet.milk',
            'getManufacturingProcess'
        );
        event.batchId = milkBatch.batchId;
        event.shipper = milkBatch.shipper;
        event.manufacturer = milkBatch.manufacturer;
        event.timeStamp = new Date();

        event.M_ReceiptTimestamp = milkBatch.M_ReceiptTimestamp;
        event.M_PersonReceipt = milkBatch.M_PersonReceipt;
        event.M_Lot = milkBatch.M_Lot;
        event.M_BestByDate = milkBatch.M_BestByDate;
        event.M_ProductType = milkBatch.M_ProductType;
    }
}

```

```

    event.M_Kilograms = milkBatch.M_Kilograms;
    event.M_Liters = milkBatch.M_Liters;
    event.M_Fat = milkBatch.M_Fat;
    event.M_Protein = milkBatch.M_Protein;
    event.M_Inhibitors = milkBatch.M_Inhibitors;
    event.M_Temperature = milkBatch.M_Temperature;

    emit(event);

    var participantRegistry = await getParticipantRegistry(
        'org.foodnet.milk.Manufacturer'
    );
    await participantRegistry.update(milkBatch.manufacturer);
    //update ownership
    milk.owner = milkBatch.manufacturer;
    participantRegistry = await getParticipantRegistry(
        'org.foodnet.milk.Manufacturer'
    );
    await participantRegistry.update(milkBatch.manufacturer);

    // publish update
    await assetRegistry.update(milk);
} else {
    throw new Error('the batch you specified does not exist!');
}
}

/**
 * Get data about Retailer Milks Receipt.
 * @param {org.foodnet.milk.submitRetailerReceipt} milkBatch - milk batch
 * @transaction
 */
async function submitRetailerReceipt(milkBatch) {
    if (milkBatch.batchId.length <= 0) {
        throw new Error('Please enter the batchId');
    }

    const assetRegistry = await getAssetRegistry('org.foodnet.milk.Milk');

    const exists = await assetRegistry.exists(milkBatch.batchId);

    if (exists) {
        const milk = await assetRegistry.get(milkBatch.batchId);

        // Start of delivery
        milk.R_Date = new Date();
        milk.status = 'READY_FOR_SALE';
        if (milk.R_PiecesQty) {
            milk.R_PiecesQty = milk.R_PiecesQty + milkBatch.R_PiecesQty;
        } else {
            milk.R_PiecesQty = milkBatch.R_PiecesQty;
        }

        // Create and emit a regulation event
        var event = getFactory().newEvent('org.foodnet.milk', 'getRetailerReceipt');
        event.batchId = milkBatch.batchId;
        event.retailer = milkBatch.retailer;

        event.R_Date = new Date();
        event.R_Invoice_No = milkBatch.R_Invoice_No;
        event.R_IssueDate = milkBatch.R_IssueDate;
    }
}

```

```

    event.R_PiecesQty = milkBatch.R_PiecesQty;
    event.asset = milk;

    emit(event);

    // publish update
    await assetRegistry.update(milk);
  } else {
    throw new Error('the batch you specified does not exist!');
  }
}

/**
 * Create the participants needed for the demo
 * @param {org.foodnet.milk.CreateDemoParticipants} createDemoParticipants - the
 * CreateDemoParticipants transaction
 * @transaction
 */
async function createDemoParticipants() {
  // eslint-disable-line no-unused-vars
  const factory = getFactory();
  const NS = 'org.foodnet.milk';

  const globalAddress = factory.newConcept(NS, 'Address');
  globalAddress.country = 'GR';

  // create the farmer
  const farmerRegistry = await getParticipantRegistry(NS + '.Farmer');
  const farmer1 = factory.newResource(NS, 'Farmer', 'FARM01@foodnet');
  farmer1.address = globalAddress;
  farmer1.organizationName = 'Farma Kavalariou';
  await farmerRegistry.add(farmer1);

  // create the shipper
  const shipperRegistry = await getParticipantRegistry(NS + '.Shipper');
  const shipper1 = factory.newResource(NS, 'Shipper', 'TRANSF01@foodnet');
  shipper1.address = globalAddress;
  shipper1.organizationName = 'Metaforiki Thessalonikis';
  await shipperRegistry.add(shipper1);

  // create the manufacturer
  const manufacturerRegistry = await getParticipantRegistry(
    NS + '.Manufacturer'
  );
  const manufacturer1 = factory.newResource(NS, 'Manufacturer', 'GAL@foodnet');
  manufacturer1.address = globalAddress;
  manufacturer1.organizationName = 'Galaktoviomihania Thessalonikis';
  await manufacturerRegistry.add(manufacturer1);

  // create the retailer
  const retailerRegistry = await getParticipantRegistry(NS + '.Retailer');
  const retailer1 = factory.newResource(NS, 'Retailer', 'SM109@foodnet');
  retailer1.address = globalAddress;
  retailer1.organizationName = 'Supermarket 109 - Thermi';
  await retailerRegistry.add(retailer1);

  const retailer2 = factory.newResource(NS, 'Retailer', 'SM124@foodnet');
  retailer2.address = globalAddress;
  retailer2.organizationName = 'Supermarket 124 - Kalamaria';
  await retailerRegistry.add(retailer2);
}

```

```
const retailer3 = factory.newResource(NS, 'Retailer', 'SM119@foodnet');
retailer3.address = globalAddress;
retailer3.organizationName = 'Supermarket 119 - Lagadas';
await retailerRegistry.add(retailer3);
}
```

B. Composer Query Language

Τα ερωτήματα που συντάχθηκαν στην CQL έχουν να κάνουν με τον τομέα στον οποίο απευθύνονται και τα κριτήρια τα οποία δίνονται. Σε μια πιο προχωρημένη και σύνθετη λογική παρατηρείται το ερώτημα `getMyHistory` όπου χρησιμοποιεί το συστημικό `HistorianRecord` και δείχνει τη δυναμική του φιλτράροντας με τον συμμετέχοντα της κλήσης των συναλλαγών. Έτσι επιστρέφει όλο το ιστορικό των συναλλαγών με τα δεδομένα τους στα οποία συμμετείχε ο χρήστης. Άλλα ειδικά ερωτήματα είναι οι συναλλαγές γάλακτος βάση παρτίδας, τα περιουσιακά στοιχεία γάλακτος βάση ημερομηνίας λήξης (καλύτερης κατανάλωσης πριν από) κ.λπ.

```
/**
 * Extra Queries
 */
query queryMilk {
  description: "get history by batchId"
  statement:
    SELECT org.foodnet.milk.Milk
      WHERE (batchId == _$batchId)
}

query getBatchHistory {
  description: "see all of the participants batch"
  statement:
    SELECT org.foodnet.milk.transferMilk
      WHERE (batchId == _$batchId )
}

query getMilkByBBD {
  description: "get history by best by date"
  statement:
```

```

        SELECT org.foodnet.milk.Milk
           WHERE (M_BestByDate == _$M_BestByDate)
    }

    query getRetailersHistory {
        description: "see all retailers stores with this batch"
        statement:
            SELECT org.foodnet.milk.submitRetailerReceipt
               WHERE (batchId == _$batchId)
    }

    query getMyHistory {
        description: "see all history transactions"
        statement:
            SELECT org.hyperledger.composer.system.HistorianRecord
               WHERE (participantInvoking == _$participantInvoking AND
transactionType == _$transactionType)
    }

```

C. Composer Access Control Language

Τα δικαιώματα που συντάχθηκαν στην CACL ορίζουν τις συστημικές προσβάσεις ανά τομέα, ανά επιχειρησιακό διαχειριστή, ανά πόρους συστήματος, ανά συμμετέχων σε κατηγορία και περιουσιακό στοιχείο. Είναι το σημείο που αποκτά αξία η ασφάλεια και η πρόσβαση στην πληροφορία. Παρέχοντας πλήρη δυνατότητες για τροποποίηση και παραμετροποίηση ανά οντότητα.

```

/**
 * Access control list.
 */
rule SystemACL {
    description: "System ACL to permit all access"
    participant: "org.hyperledger.composer.system.Participant"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}

rule NetworkAdminUser {

```

```

    description: "Grant business network administrators full access to USER
resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "*"
    action: ALLOW
}

rule NetworkAdminSystem {
    description: "Grant business network administrators full access to
SYSTEM resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.*"
    action: ALLOW
}

rule ReadCommodity {
    description: "All participants can read its own goods"
    participant(p): "org.foodnet.milk.*"
    operation: READ
    resource(r): "org.foodnet.milk.Milk"
    condition: (r.owner.getIdentifier() == p.getIdentifier())
    action: ALLOW
}

rule CanDelete {
    description: "Farmer example can delete contracts that he is a part of"
    participant(p): "org.foodnet.milk.Farmer"
    operation: READ, DELETE
    resource(r): "org.foodnet.milk.Milk"
    condition: (r.owner.getIdentifier() === p.getIdentifier())
    action: ALLOW
}
/** etc... */

rule historianAccess{
    description: "Only allow members to read historian records referencing
transactions they submitted."
    participant(p): "org.foodnet.milk.Business"
    operation: READ
    resource(r): "org.hyperledger.composer.system.HistorianRecord"
    condition: (r.participantInvoking.getIdentifier() == p.getIdentifier())
    action: ALLOW
}

```

```
}
```

D. Angular Web Application Component

Η αρχιτεκτονική της γλώσσας Angular είναι Component Oriented. Αναπτύχθηκαν components για κάθε λειτουργία της web εφαρμογής όπως για την κεντρική σελίδα, τον τελικό καταναλωτή, τους αγρότες, τους μεταφορείς, τους βιομήχανους, τους λιανέμπορους αλλά και παράπλευρα επαναχρησιμοποιούμενα components. Θα παρουσιαστεί σε αυτή την ενότητα ο κώδικας του component για την εφαρμογή των Αγροτών. Περιλαμβάνονται οι δυνατότητες δημιουργίας νέας συλλογής γάλακτος, η ολοκλήρωση της συλλογής του με την καταχώρηση των απαραίτητων πεδίων, η προβολή των πληροφοριών και τα επιπλέον σύνθετα ερωτήματα.

```
import { Component, OnInit, Input } from '@angular/core';
import {
  FormGroup,
  FormControl,
  Validators,
  FormBuilder
} from '@angular/forms';
import { submitFarmDataService } from './submitFarmData.service';
import 'rxjs/add/operator/toPromise';
import { addMilkService } from './addMilk.service';
import { MilkService } from 'app/Milk/Milk.service';
import { QueriesService } from 'app/Queries/Queries.service';
import * as _ from 'lodash';

@Component({
  selector: 'app-submitfarmdata',
  templateUrl: './Farm.component.html',
  styleUrls: ['./Farm.component.css'],
  providers: [
    submitFarmDataService,
    addMilkService,
    MilkService,
    QueriesService
  ]
})
export class FarmComponent implements OnInit {
  myForm: FormGroup;
  myFormAdd: FormGroup;

  private allTransactions;
  private Transaction;
```



```

private currentId;
private errorMessage;
private milkAssets;
private history;
private jsonString;
user = 'resource:org.foodnet.milk.Farmer#FARM01@foodnet';

F_Timestamp = new FormControl('', Validators.required);
F_Name = new FormControl('', Validators.required);
F_PersonMilked = new FormControl('', Validators.required);
F_AnimalCount = new FormControl('', Validators.required);
F_LitersOfMilk = new FormControl('', Validators.required);
batchId = new FormControl('', Validators.required);
transactionId = new FormControl('', Validators.required);
timestamp = new FormControl('', Validators.required);

// Add
milkType = new FormControl('', Validators.required);
status = new FormControl('', Validators.required);
farmer = new FormControl('', Validators.required);
transactionIdAdd = new FormControl('', Validators.required);
timestampAdd = new FormControl('', Validators.required);

constructor(
  private servicesubmitFarmData: submitFarmDataService,
  private serviceaddMilk: addMilkService,
  private serviceMilk: MilkService,
  private serviceQuery: QueriesService,
  fb: FormBuilder
) {
  this.myFormAdd = fb.group({
    milkType: this.milkType,
    status: this.status,
    farmer: this.farmer,
    transactionId: this.transactionId,
    timestamp: this.timestamp
  });

  this.myForm = fb.group({
    F_Timestamp: this.F_Timestamp,
    F_Name: this.F_Name,
    F_PersonMilked: this.F_PersonMilked,
    F_AnimalCount: this.F_AnimalCount,
    F_LitersOfMilk: this.F_LitersOfMilk,
    batchId: this.batchId,
    transactionId: this.transactionIdAdd,
    timestamp: this.timestampAdd
  });
}

ngOnInit(): void {
  this.loadAll().then(() => {
    this.loadAllAdd().then(() => {
      this.loadMilks().then(() => {
        this.loadHistory();
      });
    });
  });
}

loadAll(): Promise<any> {

```

```

const templist = [];
return this.servicesubmitFarmData
  .getAll()
  .toPromise()
  .then(result => {
    this.errorMessage = null;
    result.forEach(transaction => {
      templist.push(transaction);
    });
    this.allTransactions = templist;
  })
  .catch(error => {
    if (error === 'Server error') {
      this.errorMessage =
        'Could not connect to REST server. Please check your configuration
details';
    } else if (error === '404 - Not Found') {
      this.errorMessage =
        '404 - Could not find API route. Please check your available APIs.';
    } else {
      this.errorMessage = error;
    }
  });
}

loadAllAdd(): Promise<any> {
  const templist = [];
  return this.serviceaddMilk
    .getAll()
    .toPromise()
    .then(result => {
      this.errorMessage = null;
      result.forEach(transaction => {
        templist.push(transaction);
      });
      this.allTransactions = templist;
    })
    .catch(error => {
      if (error === 'Server error') {
        this.errorMessage =
          'Could not connect to REST server. Please check your configuration
details';
      } else if (error === '404 - Not Found') {
        this.errorMessage =
          '404 - Could not find API route. Please check your available APIs.';
      } else {
        this.errorMessage = error;
      }
    });
}

loadMilks(): Promise<any> {
  const templist = [];
  return this.serviceMilk
    .getAll()
    .toPromise()
    .then(result => {
      this.errorMessage = null;
      result.forEach(asset => {
        templist.push(asset);
      });
    });
}

```

```

        this.milkAssets = tempList.filter(i => {
            return i.owner === this.user;
        });
        console.log(this.milkAssets);
    })
    .catch(error => {
        if (error === 'Server error') {
            this.errorMessage =
                'Could not connect to REST server. Please check your configuration
details';
        } else if (error === '404 - Not Found') {
            this.errorMessage =
                '404 - Could not find API route. Please check your available APIs.';
        } else {
            this.errorMessage = error;
        }
    });
}

loadHistory(): Promise<any> {
    return this.serviceQuery
        .getMyHistory(
            'resource:org.hyperledger.composer.system.NetworkAdmin#admin',
            // 'org.foodnet.milk.submitFarmData'
            'org.foodnet.milk.submitShipping'
        )
        .toPromise()
        .then(result => {
            this.errorMessage = null;
            this.history = _.flatMap(result, st => st.eventsEmitted);
            this.history = _.orderBy(this.history, 'F_Timestamp', 'desc');
            console.log('History: ', this.history);
        })
        .catch(error => {
            if (error === 'Server error') {
                this.errorMessage =
                    'Could not connect to REST server. Please check your configuration
details';
            } else if (error === '404 - Not Found') {
                this.errorMessage =
                    '404 - Could not find API route. Please check your available APIs.';
            } else {
                this.errorMessage = error;
            }
        });
}

showJsonAsset(milk): void {
    this.jsonString = milk;
}

changeBatchId(batchId): void {
    this.myForm.setValue({
        F_Timestamp: null,
        F_Name: null,
        F_PersonMilked: null,
        F_AnimalCount: null,
        F_LitersOfMilk: null,
        batchId: batchId,
        transactionId: null,
        timestamp: null
    });
}

```

```

    });
  }

  /**
   * Event handler for changing the checked state of a checkbox (handles array
   enumeration values)
   * @param {String} name - the name of the transaction field to update
   * @param {any} value - the enumeration value for which to toggle the checked state
   */
  changeArrayValue(name: string, value: any): void {
    const index = this[name].value.indexOf(value);
    if (index === -1) {
      this[name].value.push(value);
    } else {
      this[name].value.splice(index, 1);
    }
  }

  hasArrayValue(name: string, value: any): boolean {
    return this[name].value.indexOf(value) !== -1;
  }

  addTransactionAdd(form: any): Promise<any> {
    this.Transaction = {
      $class: 'org.foodnet.milk.addMilk',
      status: 'MILK_COLLECTION',
      farmer: 'resource:org.foodnet.milk.Farmer#FARM01@foodnet',
      milkType: this.milkType.value
    };

    this.myFormAdd.setValue({
      milkType: null,
      status: null,
      farmer: null,
      transactionId: null,
      timestamp: null
    });

    return this.serviceaddMilk
      .addTransaction(this.Transaction)
      .toPromise()
      .then(() => {
        this.errorMessage = null;
        this.myFormAdd.setValue({
          milkType: null,
          status: null,
          farmer: null,
          transactionId: null,
          timestamp: null
        });
        this.loadMilks();
      })
      .catch(error => {
        if (error === 'Server error') {
          this.errorMessage =
            'Could not connect to REST server. Please check your configuration
details';
        } else {
          this.errorMessage = error;
        }
      })
      .finally(() => {});
  });
}

```

```

}

addTransaction(form: any): Promise<any> {
  this.Transaction = {
    $class: 'org.foodnet.milk.submitFarmData',
    F_Timestamp: new Date().toISOString(),
    F_Name: 'Farma Kavalariou',
    F_PersonMilked: 'Maria K.',
    F_AnimalCount: this.F_AnimalCount.value,
    F_LitersOfMilk: this.F_LitersOfMilk.value,
    batchId: this.batchId.value,
    timestamp: new Date().toISOString()
  };

  this.myForm.setValue({
    F_Timestamp: null,
    F_Name: null,
    F_PersonMilked: null,
    F_AnimalCount: null,
    F_LitersOfMilk: null,
    batchId: null,
    transactionId: null,
    timestamp: null
  });

  return this.services.submitFarmData
    .addTransaction(this.Transaction)
    .toPromise()
    .then(() => {
      this.errorMessage = null;
      this.myForm.setValue({
        F_Timestamp: null,
        F_Name: null,
        F_PersonMilked: null,
        F_AnimalCount: null,
        F_LitersOfMilk: null,
        batchId: null,
        transactionId: null,
        timestamp: null
      });
      this.loadMilks();
    })
    .catch(error => {
      if (error === 'Server error') {
        this.errorMessage =
          'Could not connect to REST server. Please check your configuration
details';
      } else {
        this.errorMessage = error;
      }
    });
}

updateTransaction(form: any): Promise<any> {
  this.Transaction = {
    $class: 'org.foodnet.milk.submitFarmData',
    F_Timestamp: this.F_Timestamp.value,
    F_Name: this.F_Name.value,
    F_PersonMilked: this.F_PersonMilked.value,
    F_AnimalCount: this.F_AnimalCount.value,
    F_LitersOfMilk: this.F_LitersOfMilk.value,

```

```

        batchId: this.batchId.value,
        timestamp: this.timestamp.value
    };

    return this.servicessubmitFarmData
        .updateTransaction(form.get('transactionId').value, this.Transaction)
        .toPromise()
        .then(() => {
            this.errorMessage = null;
        })
        .catch(error => {
            if (error === 'Server error') {
                this.errorMessage =
                    'Could not connect to REST server. Please check your configuration
details';
            } else if (error === '404 - Not Found') {
                this.errorMessage =
                    '404 - Could not find API route. Please check your available APIs.';
            } else {
                this.errorMessage = error;
            }
        });
    }

    deleteTransaction(): Promise<any> {
        return this.servicessubmitFarmData
            .deleteTransaction(this.currentId)
            .toPromise()
            .then(() => {
                this.errorMessage = null;
            })
            .catch(error => {
                if (error === 'Server error') {
                    this.errorMessage =
                        'Could not connect to REST server. Please check your configuration
details';
                } else if (error === '404 - Not Found') {
                    this.errorMessage =
                        '404 - Could not find API route. Please check your available APIs.';
                } else {
                    this.errorMessage = error;
                }
            });
    }

    setId(id: any): void {
        this.currentId = id;
    }

    getForm(id: any): Promise<any> {
        return this.servicessubmitFarmData
            .getTransaction(id)
            .toPromise()
            .then(result => {
                this.errorMessage = null;
                const formObject = {
                    F_Timestamp: null,
                    F_Name: null,
                    F_PersonMilked: null,
                    F_AnimalCount: null,
                    F_LitersOfMilk: null,

```

```

        batchId: null,
        transactionId: null,
        timestamp: null
    };

    if (result.F_Timestamp) {
        formObject.F_Timestamp = result.F_Timestamp;
    } else {
        formObject.F_Timestamp = null;
    }

    if (result.F_Name) {
        formObject.F_Name = result.F_Name;
    } else {
        formObject.F_Name = null;
    }

    if (result.F_PersonMilked) {
        formObject.F_PersonMilked = result.F_PersonMilked;
    } else {
        formObject.F_PersonMilked = null;
    }

    if (result.F_AnimalCount) {
        formObject.F_AnimalCount = result.F_AnimalCount;
    } else {
        formObject.F_AnimalCount = null;
    }

    if (result.F_LitersOfMilk) {
        formObject.F_LitersOfMilk = result.F_LitersOfMilk;
    } else {
        formObject.F_LitersOfMilk = null;
    }

    if (result.batchId) {
        formObject.batchId = result.batchId;
    } else {
        formObject.batchId = null;
    }

    if (result.transactionId) {
        formObject.transactionId = result.transactionId;
    } else {
        formObject.transactionId = null;
    }

    if (result.timestamp) {
        formObject.timestamp = result.timestamp;
    } else {
        formObject.timestamp = null;
    }

    this.myForm.setValue(formObject);
})
.catch(error => {
    if (error === 'Server error') {
        this.errorMessage =
            'Could not connect to REST server. Please check your configuration
details';
    } else if (error === '404 - Not Found') {

```

```

        this.errorMessage =
            '404 - Could not find API route. Please check your available APIs.';
    } else {
        this.errorMessage = error;
    }
    });
}

resetForm(): void {
    this.myForm.setValue({
        F_Timestamp: null,
        F_Name: null,
        F_PersonMilked: null,
        F_AnimalCount: null,
        F_LitersOfMilk: null,
        batchId: null,
        transactionId: null,
        timestamp: null
    });
}

resetFormAdd(): void {
    this.myFormAdd.setValue({
        milkType: null,
        status: null,
        farmer: null,
        transactionId: null,
        timestamp: null
    });
}
}
}

```

E. Angular Web Application Service

Η Angular χρησιμοποιεί τις οντότητες των services για να επαναχρησιμοποιεί και να επικοινωνεί η εφαρμογή με εξωτερικές συνήθως πηγές δεδομένων. Παρακάτω περιλαμβάνεται ένα παράδειγμα επικοινωνίας για τα στοιχεία της φάρμας που υλοποιεί βασικές CRUD (create, read, update, delete) λειτουργίες. Στην πλατφόρμα έχουν υλοποιηθεί εννέα οντότητες service με τις λειτουργίες που χρειάζονται για την καθεμία: CreateDemoParticipants, addMilk, submitFarmData, submitManufacturingProcess, Queries, submitRetailerReceipt, submitShippingService, TemperatureReading, dataService.

```

import { Injectable } from '@angular/core';
import { DataService } from '../data.service';
import { Observable } from 'rxjs/Observable';
import { submitFarmData } from '../org.foodnet.milk';
import 'rxjs/Rx';

```



```
// Can be injected into a constructor
@Inject()
export class submitFarmDataService {

  private NAMESPACE = 'submitFarmData';

  constructor(private dataService: DataService<submitFarmData>) {
  };

  public getAll(): Observable<submitFarmData[]> {
    return this.dataService.getAll(this.NAMESPACE);
  }

  public getTransaction(id: any): Observable<submitFarmData> {
    return this.dataService.getSingle(this.NAMESPACE, id);
  }

  public addTransaction(itemToAdd: any): Observable<submitFarmData> {
    return this.dataService.add(this.NAMESPACE, itemToAdd);
  }

  public updateTransaction(id: any, itemToUpdate: any): Observable<submitFarmData> {
    return this.dataService.update(this.NAMESPACE, id, itemToUpdate);
  }

  public deleteTransaction(id: any): Observable<submitFarmData> {
    return this.dataService.delete(this.NAMESPACE, id);
  }
}
```