ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη έξυπνων συμβολαίων σε αποκεντρωμένα δίκτυα αλυσίδας κορμού με επίτρεψη για την προστασία ιδιωτικότητας και την εξουσιοδότηση χρήσης KYC δεδομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Φοίβος Νηχωρίτης

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Μάρτιος 2022

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη έξυπνων συμβολαίων σε αποκεντρωμένα δίκτυα αλυσίδας κορμού με επίτρεψη για την προστασία ιδιωτικότητας και την εξουσιοδότηση χρήσης KYC δεδομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Φοίβος Νηχωρίτης

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16η Μαρτίου 2022.

.....	.....	.....
Θεοδώρα Βαρβαρίγου	Συμεών Παπαβασιλείου	Εμμανουήλ Βαρβαρίγος
Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2022

Αυτή η σελίδα σκοπίμως αφέθηκε κενή

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

.....
Φοίβος Νηχωρίτης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Αυτή η σελίδα σκοπίμως αφέθηκε κενή

Περίληψη

Η υιοθέτηση νέων μορφών συναλλαγών και η ψηφιοποίηση της οικονομίας έχουν προκαλέσει την ανάγκη για πιο ταχύ και αποτελεσματικό έλεγχο της ταυτότητας των πελατών ενός χρηματοπιστωτικού ιδρύματος, προκειμένου να αποφευχθεί η νομιμοποίηση εσόδων από παράνομες δραστηριότητες, τον λεγόμενο Know Your Customer (KYC) έλεγχο. Η παρούσα διπλωματική εργασία μελετάει την δυνατότητα χρήσης της τεχνολογίας Blockchain για την αυτοματοποίηση της διαδικασίας KYC. Προτείνεται μία εφαρμογή σε πλατφόρμα αποκεντρωμένου δικτύου αλυσίδας κορμού με επίτρεψη, και συγκεκριμένα στο R3 Corda η οποία επιτρέπει την κοινοποίηση ευαίσθητων δεδομένων προς επαλήθευση από τον πελάτη, με γνώμονα την προστασία της ιδιωτικότητας. Αναπτύσσονται τα έξυπνα συμβόλαια που διέπουν την λειτουργία της εφαρμογής και εξασφαλίζουν την προστασία των δεδομένων του πελάτη. Γίνεται μελέτη και σύγκριση δύο διαφορετικών τρόπων αποθήκευσης των δεδομένων, με χρήση ή όχι του InterPlanetary File System, καθώς και μελέτη σχετικά με την χρονική απόδοση του προτεινόμενου συστήματος ανάλογα με τον αριθμό των συμμετεχόντων οργανισμών και του μεγέθους των προς κοινοποίηση δεδομένων. Προτείνεται, τέλος, μία επέκταση της εφαρμογής η οποία επιτυγχάνει τον ταχύ και αξιόπιστο διαμοιρασμό του κόστους της διαδικασίας KYC ανάμεσα στους εμπλεκόμενους χρηματοπιστωτικούς οργανισμούς. Συμπεραίνεται ότι η χρήση της τεχνολογίας Blockchain, με έμφαση τις πλατφόρμες με επίτρεψη, μπορεί να βοηθήσει σημαντικά στην αυτοματοποίηση και τον περιορισμό του κόστους της επαλήθευσης των ευαίσθητων KYC δεδομένων των πελατών.

Λέξεις – κλειδιά : Blockchain, Permissioned blockchain, Corda, αποκεντρωμένα δίκτυα αλυσίδας κορμού, Know Your Customer, DLT στα χρηματοοικονομικά

Αυτή η σελίδα σκοπίμως αφέθηκε κενή

Abstract

The adoption of new forms of transactions and the digitization of the economy have created the need for a faster and more efficient control of a financial institution's client's identity. This process of validating a customer's identity to eschew money laundering activities is called Know Your Customer (KYC). In this Diploma Thesis, we study the possibility of using blockchain technology towards the automatization of the KYC process. An application in the permissioned blockchain platform R3 Corda is proposed and fully developed. The application allows the sharing of a client's sensitive data to a number of financial institutions of his choice, focusing on privacy. A Smart Contract which ensures privacy protection is proposed and developed. A comparative analysis between two different methods of data storage, with or without the usage of InterPlanetary File System, is conducted, focusing on time needed for the application to be fully executed. The parameters which are changed during the analysis are the number of involved financial institutions and the size of the documents to be shared. Effectively, an extension of the application is proposed, which seeks to achieve a faster and more efficient sharing of the KYC process cost among the involved financial institutions.

Keywords: Blockchain, Permissioned blockchain, Corda, Know Your Customer, privacy in blockchain, Decentralized finance

Ευχαριστίες

Με την εκπόνηση της παρούσας διπλωματικής εργασίας ο κύκλος των σπουδών μου στην Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου φτάνει στο τέλος του.

Θα ήθελα να ευχαριστήσω, πρωτίστως, την Καθηγήτρια κ. Βαρβαρίγου για την ευκαιρία που μου προσέφερε να μελετήσω ένα ιδιαίτερα ενδιαφέρον αντικείμενο αλλά και για την συνολική επίβλεψη και καθοδήγηση της εργασίας. Ένα βαθύ ευχαριστώ οφείλω και στον διδακτορικό ερευνητή Νίκο Καψούλη για την διαρκή βοήθεια και την πολύτιμη υποστήριξη που μου προσέφερε καθ' όλη την διάρκεια της εκπόνησης της διπλωματικής εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω ιδιαίτερα τόσο την οικογένεια μου η οποία μου παρείχε απεριόριστη στήριξη κατά την διάρκεια των σπουδών μου όσο και τους φίλους και συμφοιτητές μου, ιδιαιτέρως τους Πέτρο Σ., Άννα Μαρία Β. και Δημήτρη Μ., οι οποίοι υπήρξαν ιδανικοί συνοδοιπόροι στο πεναταετές αυτό ταξίδι.

Φοίβος Νηχωρίτης

Αθήνα, 8η Μαρτίου 2022

Αυτή η σελίδα σκοπίμως αφέθηκε κενή

Περιεχόμενα

Ευρετήριο Εικόνων

Ευρετήριο Διαγραμμάτων

Κεφάλαιο 1

Εισαγωγή

1.1. Εισαγωγή στο αντικείμενο της διπλωματικής εργασίας

Η σημερινή εποχή χαρακτηρίζεται από μία σειρά ριζικών αλλαγών σε πληθώρα τομέων της κοινωνικής και οικονομικής ζωής του ανθρώπου. Οι φρενήρεις ρυθμοί τεχνολογικής ανάπτυξης οδηγούν στην δημιουργία καινούργιων αναγκών και διαδικασιών αλλά και στην αναθεώρηση του τρόπου υλοποίησης των υφιστάμενων διαδικασιών. Ένας από τους τομείς που έχει επηρεαστεί, ιδιαίτερα, από την ανάπτυξη αυτή είναι και ο χρηματοοικονομικός τομέας. Οι αλλαγές που συντελούνται στις χρηματοοικονομικού τύπου συναλλαγές έχουν δύο βασικές αιτιάσεις: την ίδια την ανάπτυξη της οικονομίας με την είσοδο περισσότερων παικτών, δηλαδή επιχειρήσεων και πελατών, καθώς και την απαίτηση για ταχύτερες και γρηγορότερες συναλλαγές ως καθοριστικό παράγοντα για τον εναρμονισμό με το ευρύτερο τεχνολογικό οικοσύστημα.

Σύμφωνα με την **Παγκόσμια Τράπεζα,** καθημερινά πραγματοποιούνται περισσότερες από 1 δισεκατομμύρια χρηματοοικονομικές συναλλαγές. Ήδη από το 1986 και γινόμενη ολοένα και εντονότερη στις μέρες μας, έχει ξεκινήσει να εντατικοποιείται ο έλεγχος και η ρύθμιση των συναλλαγών στις περισσότερες τράπεζες. Η παγκοσμιοποιήμενη οικονομία και η δυνατότητα που δίνεται μέσω αυτής για νομιμοποίηση κερδών από παράνομες δραστηριότητες (ξέπλυμα μαύρου χρήματος – money laundering) σε συνδυασμό με τον μεγάλο αριθμό συναλλαγών που λαμβάνουν χώρα κάθε μέρα έχουν δημιουργήσει την ανάγκη για ταχύτερο και αποδοτικότερο έλεγχο των πελατών. Ο έλεγχος αυτός που αποσκοπεί στην επαλήθευση της ταυτότητας του πελάτη και της προέλευσης των χρημάτων του, μεταξύ άλλων, καλείται Know Your Customer (KYC).

Η παρούσα διπλωματική πραγματεύεται την δυνατότητα χρήσης της τεχνολογίας Blockchain ως εργαλείου για την αυτοματοποίηση της διαδικασίας KYC με γνώμονα την προστασία της ιδιωτικότητας του χρήστη. Τα βασικά χαρακτηριστικά των εφαρμογών blockchain που είναι το αδιάβλητο των συναλλαγών και η δυσκολία αλλοίωσης των δεδομένων μπορεί να δείχνουν ακατάλληλα για χρήση σε εφαρμογές διαμοιρασμού ευαίσθητων και ιδιωτικών δεδομένων, χωρίς κάτι τέτοιο να ισχύει. Αντιθέτως, η ανάπτυξη πλατφορμών blockchain με επίτρεψη, όπου συμμετέχουν μόνο ορισμένοι εξουσιοδοτημένοι χρήστες επιτρέπουν, πλέον, την χρήση της τεχνολογίας αυτής σε περιπτώσεις που επιδιώκεται η μεγιστοποίηση της ιδιωτικότητας των δεδομένων. Μία από αυτές τις πλατφόρμες είναι το Corda της εταιρείας R3.

Προτείνεται, επομένως, στην παρούσα εργασία μία εφαρμογή στην πλατφόρμα blockchain με επίτρεψη Corda, όπου ο χρήστης ενός δικτύου έχει την δυνατότητα να κοινοποιήσει τα ιδιωτικά του δεδομένα ώστε αυτά να χρησιμοποιηθούν για επαλήθευση Know Your Customer. Ο χρήστης επιλέγει σε ποιόν ή ποιους οργανισμούς θα κοινοποιήσει τα δεδομένα αυτά. Τα έξυπνα συμβόλαια που αναπτύσσονται στην παρούσα διπλωματική εξασφαλίζουν την ιδιωτικότητα και αυτοματοποιούν την διαδικασία κοινοποίησης των δεδομένων. Δίνουν τη δυνατότητα στον πελάτη να παρέχει ανά πάσα στιγμή τα πιο ενημερωμένα δεδομένα του σε όποιους οργανισμούς-συμμετέχοντες του δικτύου επιλέξει. Ταυτόχρονα, λόγω της ιδιότητας του Corda να μην χρησιμοποιεί gossip protocol, όλοι οι συμμετέχοντες του δικτύου στους οποίους δεν έχει δοθεί έγκριση από τον πελάτη δεν έχουν πρόσβαση στα KYC δεδομένα.

Η αυτοματοποίηση της διαδικασίας παρουσιάζεται με δύο διαφορετικούς τρόπους αποθήκευσης των δεδομένων, με ή χωρίς χρήση του IPFS, ενώ γίνονται πειράματα για την μέτρηση της απόδοσης της εφαρμογής όταν μεταβάλλονται παράμετροι όπως ο αριθμός των οργανισμών που λαμβάνουν τα δεδομένα αλλά και το μέγεθος των ίδιων των δεδομένων. Τέλος προτείνεται μία επέκταση της εφαρμογής η οποία θα επιτρέπει τον διαμοιρασμό του κόστους της διαδικασίας επαλήθευσης των KYC δεδομένων.

1.2. Οργάνωση της διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία χωρίζεται σε τέσσερα βασικά μέρη, (1) το θεωρητικό υπόβαθρο των εξεταζόμενων εννοιών, (2) η παρουσίαση της αρχιτεκτονικής της προτεινόμενης εφαρμογής και (3) συγκριτικά αποτελέσματα της απόδοσης της εφαρμογής και (4) μία πρόταση για περαιτέρω επέκταση της εφαρμογής.

Αρχικά, στο Κεφάλαιο 2 αναλύονται οι βασικές έννοιες: η εξέλιξη του διαδικτύου από την πρώτη σύλληψη της ιδέας μέχρι την σημερινή και μελλοντική επέκταση του ως σημασιολογικό ιστό (Web 3.0), η έννοια των δικτύων ομότιμων κόμβων τα οποία οδήγησαν στον αποκεντρωμένο ιστό και το blockchain, τα είδη των blockchain και, τέλος, την χρήση της τεχνολογίας αυτής στον χρηματοοικονομικό τομέα και, συγκεκριμένα, στην διαδικασία KYC. Στο Κεφάλαιο 3 περιγράφεται αναλυτικά η αρχιτεκτονική της προτεινόμενης εφαρμογής (CordApp) με δύο προτεινόμενες υλοποιήσεις, με χρήση IPFS ή με χρήση Corda attachments. Στο Κεφάλαιο 4 παρουσιάζονται τα αποτελέσματα της συγκριτικής μελέτης για την απόδοση της εφαρμογής με διάφορες παραμέτρους να διαφοροποιούνται. Τέλος, στο Κεφάλαιο 5 περιγράφεται μία δυνητική επέκταση του προτεινόμενου συστήματος η οποία επιτρέπει τον διαμοιρασμό του κόστους της διαδικασίας KYC.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο και σχετικές εργασίες

Στο ακόλουθο κεφάλαιο θα παρατεθεί το θεωρητικό και τεχνολογικό υπόβαθρο στο οποίο στηρίχθηκε η εκπόνηση της παρούσας εργασίας. Αρχικά γίνεται μια αναφορά στην αναδυόμενη νέα μορφή του διαδικτύου που χαρακτηρίζεται από τις αποκεντρωμένες λειτουργίες του, στην συνέχεια γίνεται εκτενής αναφορά στην τεχνολογία του Blockchain και, τέλος, συγκρίνεται η τεχνολογία δημόσιου και ιδιωτικού Blockchain, όπου στην τελευταία στηρίχτηκε η υλοποίηση που θα αναφερθεί στην συνέχεια.

2.1. Η εξέλιξη του διαδικτύου

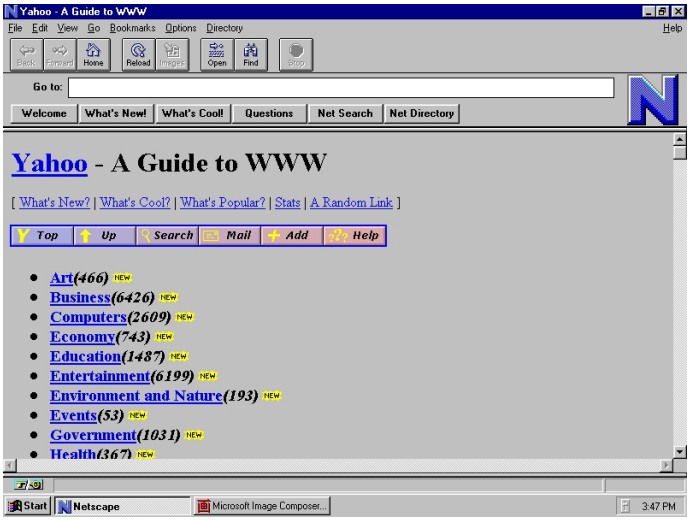
Ο όρος “World Wide Web” και διαδίκτυο συγχέονται πολύ συχνά μεταξύ τους. Το διαδίκτυο αποτελεί, ουσιαστικά, την δικτυακή υποδομή η οποία συνδέει συσκευές μεταξύ τους, ενώ ο παγκόσμιος ιστός ή World Wide Web αποτελεί έναν τρόπο πρόσβασης στην πληροφορία δια μέσου του διαδικτύου.

Ο Tim Berners-Lee πρότεινε την ιδέα του ιστού της πληροφορίας (web of information) για πρώτη φορά το 1989. Βασιζόταν σε υπερσυνδέσμους (hyperlinks) για την διασύνδεση μεταξύ εγγράφων. Γραμμένο σε Hypertext Markup Language (HTML) ένα hyperlink μπορεί να δείχνει σε οποιαδήποτε σελίδα HTML ή αρχείο που βρίσκεται στο διαδίκτυο. Στην συνέχεια, και κατά την διάρκεια της τελευταίας δεκαετίας του 20ου αιώνα η έννοια του παγκόσμιου ιστού πέρασε πολλές φάσεις μέχρι να πάρει την σημερινή του μορφή. [1]

2.1.1 Web 1.0

Η πρώιμη εποχή του παγκόσμιου ιστού που ονομάστηκε, εκ των υστέρων, Web 1.0 αποτελεί την εποχή όπου εισήχθησαν η πλειονότητα των χαρακτηριστικών του παγκόσμιου ιστού στην σημερινή επικρατούσα μορφή του. Σύμφωνα με τους Cormode και Krishnamurthy [2] οι δημιουργοί περιεχομένου στην πρώιμη αυτή περίοδο του διαδικτύου ήταν λιγοστοί, ενώ η πλειονότητα των χρηστών αποτελούσαν, απλώς, καταναλωτές του περιεχομένου. Βασικό χαρακτηριστικό, επομένως, της εποχής αυτή ήταν η στατικότητα των σελίδων στον Παγκόσμιο ιστό καθώς και η μειωμένη αλληλεπίδραση μεταξύ του συνόλου των χρηστών μιας σελίδας.

Από τεχνική άποψης, οι περισσότερες σελίδες ήταν δομημένες με χρήση Server Side Includes ή Common Gateway Interfaces και όχι γλωσσών δυναμικού προγραμματισμού όπως η Pearl ή η Python. [3]



Εικόνα 1 Η αρχική σελίδα της Yahoo, δείγμα Web 1.0 ιστοσελίδων [2]

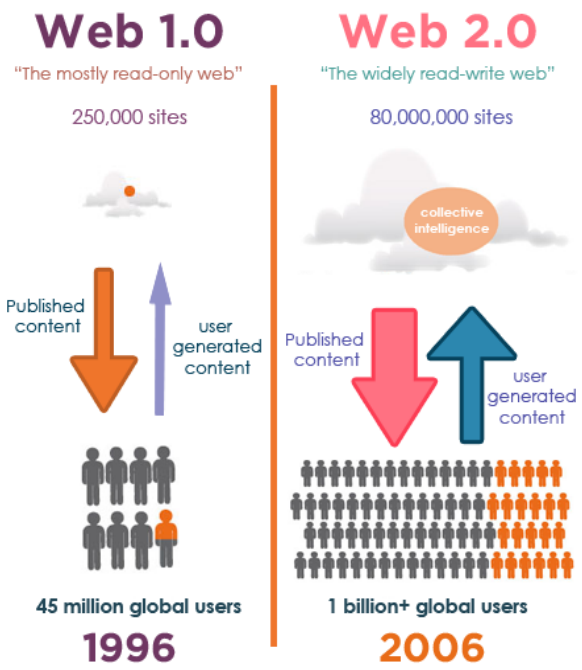
2.1.2 Web 2.0

Λειτουργίες

Η ανάπτυξη των τεχνολογικών εφαρμογών του διαδικτύου οδήγησε σε μία αλλαγή όχι μόνον του περιεχόμενου των σελίδων που αποτελούσαν μέρη του Παγκόσμιου Ιστού αλλά και των δυνατοτήτων των χρηστών - συμμετεχόντων. Η έννοια του Web 2.0 διατυπώθηκε πρώτη φορά από τον Darcy DiNucci το 1999 ενώ στην συνέχεια έγινε, ευρέως, γνωστός από τους Tim O'Reilly και Dale Dougherty το 2004, οι οποίοι παρατήρησαν πως, αντίθετα με την κυρίαρχη άποψη που αμφισβητούσε την αξία του διαδικτύου κυρίως λόγω του κραχ των dot coms, ο παγκόσμιος ιστός ήταν πιο κυρίαρχος από ποτέ. [4]

Συγκεκριμένα παρατηρήθηκε μία γρήγορη ανάπτυξη των υπαρχουσών ιστοσελίδων οι οποίες εισήγαγαν όλο και περισσότερες δυνατότητες για τους χρήστες, εκμεταλλευόμενες υπάρχουσες τεχνολογίες του διαδικτύου. Δημιουργήθηκαν, έτσι, οι λεγόμενες ιστοσελίδες Web 2.0 οι οποίες είχαν ως κυρίαρχο γνώρισμα, που τις διαφοροποιούσε από αυτές του Web 1.0, την αυξημένη εμπλοκή του χρήστη στην δημιουργία του περιεχομένου του παγκόσμιου ιστού. Πλέον, ο χρήστης αντί να αποτελεί απλώς έναν παρατηρητή μιας ιστοσελίδας με περιεχόμενο δημιουργημένο από τρίτους, ήταν σε θέση να εμπλέκεται ενεργά στην δημιουργία νέου περιεχομένου. Συγκεκριμένα οι ιστοσελίδες Web 2.0 παροτρύνουν τους χρήστες να συμβάλουν στην διαμόρφωση του περιεχόμενου τους, είτε μέσω σχολίων, είτε μέσω γραφής

άρθρων είτε δημιουργώντας έναν λογαριασμό ο οποίος και δημιουργεί ξεχωριστό μικρό-οικοσύστημα δημιουργίας περιεχομένου για κάθε χρήστη.



Εικόνα 2 Η διαφορά μεταξύ Web 1.0 και Web 2.0 [4]

Σύμφωνα με τον Best, τα χαρακτηριστικά του Web 2.0 αποτελούν «η πιο πλούσια εμπειρία του χρήστη, συμμετοχή του χρήστη, το δυναμικό περιεχόμενο, η κλιμακωσιμότητα». [5] .Οι ιστοσελίδες του Web 2.0 παροτρύνουν τους χρήστες να βασίζονται περισσότερο στον περιηγητή ώστε να κάνουν χρήση της διεπαφής χρήστη, των εφαρμογών αλλά και υπηρεσιών αποθήκευσης αρχείων. Δύο χαρακτηριστικά παραδείγματα των αυξημένων δυνατοτήτων των σελίδων του παγκόσμιου ιστού είναι τόσο η ανάπτυξη του υπολογιστικού νέφους το οποίο κατά πολλούς αποτελεί απόρροια του Web 2.0 όσο και η αυξημένη χρήστη ιστοσελίδων που χρησιμοποιούνται ως εφαρμογές, για παράδειγμα για μετατροπή αρχείων. Στην πρότερη εποχή του Παγκόσμιου Ιστού, οι αυτόνομες εφαρμογές ήταν οι μόνες που μπορούν να διεκπεραιώσουν τόσο σύνθετες λειτουργίες.

Τεχνολογίες

Η υλοποίηση των ανωτέρω λειτουργιών που καθόρισαν την μετάβαση στην εποχή Web 2.0 έγινε δυνατή μέσω της αυξημένης υιοθέτηση συγκεκριμένων τεχνολογιών, με επίκεντρο τις client-side τεχνολογίες. Βασικές τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του Web 2.0 περιλαμβάνουν τα frameworks Ajax και Javascript. Καινοτόμα και καθοριστική για τον μετασχηματισμό των ιστοσελίδων σε πιο αλληλεπιδραστικές οντότητες υπήρξε η δυνατότητα του Ajax να χρησιμοποιεί Javascript και Document Object Model [6] ώστε να ενημερώνει συγκεκριμένα τμήματα της ιστοσελίδας χωρίς να χρειάζεται ολική ανανέωση.

Στην ουσία, έγινε χρήση και νέων τεχνολογιών όπως είναι η Enterprise Java (J2EE), και microsoft.NET Framework, ώστε να γίνεται δυναμική εξαγωγή δεδομένων από βάσεις δεδομένων και αρχεία. Οι τεχνολογίες αυτές χρησιμοποιήθηκαν εκτενώς και στην μετέπειτα εξέλιξη του διαδικτύου.

2.1.3 Web 3.0

Ο όρος Web 3.0 χρησιμοποιείται για να περιγράψει την επόμενη, "τρίτη" γενιά των υπηρεσιών διαδικτύου και των εφαρμογών τους, οι οποίες επικεντρώνονται κυρίως στην χρήση «μηχανοκεντρικής» (machine-based) αντίληψης των δεδομένων με σκοπό την παροχή του λεγόμενου Σημασιολογικού Ιστού (semantic web). Ο απώτερος σκοπός του νέου Ιστού είναι η δημιουργία ενός συνόλου από ευφυή, συνδεδεμένα και ανοιχτά κύτταρα στο διαδίκτυο. Ο χρόνος που θα χρειαστεί προκειμένου να γίνει η πλήρης ενσωμάτωση του Web 3.0 στην καθημερινότητα των χρηστών (όπως έγινε με το Web 2.0) εικάζεται ότι θα είναι αντίστοιχης διάρκειας με αυτόν των προηγούμενων γενιών. [7]

Κυρίαρχο ρόλο στην νέα αυτή τεχνολογική πραγματικότητα θα κατέχει η διαχείριση των δεδομένων. Οι εφαρμογές του Web 2.0 επέτρεψαν την δημιουργία πολύ μεγάλων μεγεθών δεδομένων με αποτέλεσμα να προκύψει η ανάγκη για πιο πρακτική αποθήκευση αυτών καθώς και εύκολη πρόσβαση των χρηστών σε αυτά. Παράλληλα, τα δεδομένα, με την κυριαρχία του Web 3.0 βρίσκονται πλέον "απλωμένα" σε μία πληθώρα συσκευών της καθημερινότητας. Συσκευές όπως τηλέφωνα, ηλεκτρονικοί υπολογιστές, οικιακές συσκευές, αισθητήρες και οχήματα χρησιμοποιούν και παράγουν τόσο μεγάλο όγκο δεδομένων έτσι ώστε εκτιμήσεις να κάνουν λόγο για πάνω από 160 φορές περισσότερα δεδομένα ανά χρήστη το 2025 σε σχέση με το 2010.

Δύο κυρίαρχα χαρακτηριστικά του Web 3.0 αποτελεί η χρήση τεχνολογιών τεχνητής νοημοσύνης καθώς και η τάση για αποκεντροποίηση των δικτύων δεδομένων. Η αποκεντροποίηση (decentralization) καθιστά εφικτό στους δημιουργούς της πληροφορίας (είτε αυτή είναι αγροτικά δεδομένα ενός καλλιεργητή ή δεδομένα υγείας π.χ. ΗΚΓ), [8] να τα πουλήσουν ή να τα ανταλλάξουν χωρίς να χάνουν την κυριότητα σε αυτά. Με την κυριότητα εννοείται η ιδιωτικότητα των δεδομένων, ο έλεγχος πρόσβασης σε αυτά και κυρίως η αποφυγή εμπλοκής ενός τρίτου μέρους στην διαχείριση τους. Η αποκεντροποίηση δύναται να δημιουργήσει πολύ σημαντικά οφέλη στην κοινωνία επιτρέποντας στους χρήστες να αποτελούν και αυτοί μέρος της μεγάλης αλυσίδας δεδομένων χωρίς να έχουν σχέση εξάρτησης από άλλους. Παράλληλα, η ανάπτυξη της τεχνητής νοημοσύνης (Artificial intelligence) και η βελτιστοποίηση των αλγορίθμων μηχανικής μάθησης σε συνδυασμό με τον πλούτο των δεδομένων θα επιτρέψουν στο Web 3.0 να αποτελέσει την καταλυτική τεχνολογία διαδικτύου στα αμέσως επόμενα χρόνια.

2.2 Δίκτυα ομότιμων κόμβων

Η αποτύπωση του αποκεντρωμένου χαρακτήρα που λαμβάνει το διαδίκτυο αντικατοπτρίζεται και από την διαδεδομένη αρχιτεκτονική ομότιμων κόμβων, η οποία και αποτελεί θεμέλιο λίθο της τεχνολογίας blockchain.

Τα δίκτυα ομότιμων κόμβων, παρά το γεγονός ότι είχαν χρησιμοποιηθεί και κατά τα πρώτα χρόνια του διαδικτύου, εδραιωθήκαν μέσω εφαρμογών διαμοιρασμού αρχείων όπως είναι το Napster.[9] Τα συστήματά ομότιμων κόμβων αποτελούν μια αρχιτεκτονική τεχνοτροπία η οποία χρησιμοποιείται για την δημιουργία κατανεμημένων συστημάτων αλλά και εφαρμογών, στα οποία δεδομένα και υπολογιστικοί πόροι διαμοιράζονται από δύο ή περισσότερους υπολογιστές ισοδύναμα. Το σύνολο των υπολογιστών αυτών συντελούν στην προσφορά μιας ενιαίας υπηρεσίας. Σε αντίθεση με τα συστήματα πελάτη-εξυπηρετητή, τα συστήματα ομότιμων κόμβων δεν διαθέτουν κάποιον ενεργό υπολογιστή - εξυπηρετητή ο οποίος λαμβάνει αιτήσεις, τις επεξεργάζεται και απαντά σε αυτές με κάποιον τρόπο. Στην περίπτωση P2P architecture, οι κόμβοι έχουν άμεση επικοινωνία μεταξύ τους δίχως να είναι απαραίτητη η ύπαρξη ξεχωριστά διαχειριζόμενων εξυπηρετητών και της υποδομής που αυτοί απαιτούν. Στην ουσία σε ένα δίκτυο ομότιμων κόμβων, κάθε συμμετέχων είναι ταυτόχρονα και προμηθευτής και καταναλωτής των πόρων.

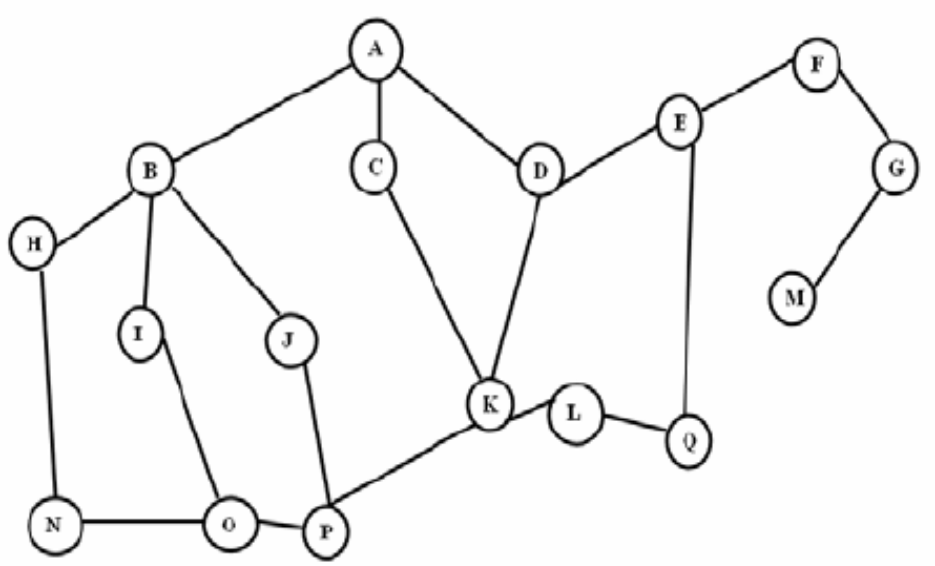
Αρχιτεκτονική

Το σύνολο των P2P αρχιτεκτονικών δικτύου έχουν ένα κοινό χαρακτηριστικό, η πραγματική ανταλλαγή των δεδομένων γίνεται πάντοτε peer-to-peer: δημιουργείται μία άμεση σύνδεση δεδομένων (data connection) μεταξύ του κόμβου που ζητάει τα δεδομένα και αυτού που τα προσφέρει. Όσον αφορά το πρωτόκολλο στρώματος μεταφοράς, χρησιμοποιείται το πρωτόκολλο TCP ενώ για το στρώμα δικτύου IP, σε επίπεδο εφαρμογής, οι συμμετέχοντες του δικτύου επικοινωνούν απευθείας μεταξύ μέσω των **λογικών ανώτερων στρωμάτων δικτύου.**

Το υπερκείμενο δίκτυο (Overlay Network) χρησιμοποιείται προκειμένου να δημιουργηθεί μία διάταξη των υφιστάμενων κόμβων του δικτύου. Ανάλογα από τον τρόπο που έχουν δομηθεί οι κόμβοι στο overlay network, ένα P2P δίκτυο μπορεί να διακριθεί σε δομημένο και αδόμητο δίκτυο ομότιμων κόμβων. Στην συνέχεια παρουσιάζονται αναλυτικά τα δύο διαφορετικά ήδη αρχιτεκτονικής οργάνωσης των δικτύων αυτών.

2.2.1 Αδόμητο δίκτυο ομότιμων κόμβων

Σε αυτή την κατηγορία δικτύων ομότιμων κόμβων, δεν υπάρχει συγκεκριμένη διάταξη (δομή) των κόμβων που απαρτίζουν το δίκτυο. Το δίκτυο ουσιαστικά αποτελείται από κόμβους οι οποίοι είναι συνδεδεμένοι μεταξύ τους με τυχαίο τρόπο. Τα δίκτυα σχηματίζουν έναν γράφο ο οποίος είναι τις περισσότερες φορές συνδετικός. Η απουσία μία συγκεκριμένης ιεραρχίας επιτρέπει την βελτιστοποίηση μίας υποπεριοχής του δικτύου εάν αυτό κρίνεται απαραίτητο σε μία συγκεκριμένη περίπτωση. Παράλληλα, το γεγονός ότι οι κόμβοι είναι πλήρως ομότιμοι και επιτελούν τον ίδιο ακριβώς ρόλο στο δίκτυο, δίνει στα αδόμητα δίκτυα P2P την δυνατότητα να είναι ιδιαίτερα ανεκτικά σε μεγάλους ρυθμούς προσθηκών ή αναχωρήσεων από το δίκτυο. Το αρνητικό ενός αδόμητου δικτύου P2P έγκειται στο γεγονός ότι ελλείψει δομής, όταν ένας κόμβος επιθυμεί να στείλει σε έναν άλλον κόμβο, είναι υποχρεωμένος να κάνει χρήση των λεγόμενων τεχνικών πλημμύρας (flooding). Με αυτόν τον τρόπο το μήνυμα/αίτημα διαβιβάζεται στο σύνολο των κόμβων μέχρι να βρεθεί ο κόμβος που κατέχει την πληροφορία. Αυτή η κατάσταση δημιουργεί αυξημένη κίνηση στους διαύλους μεταξύ των κόμβων ενώ δεν εξασφαλίζει ότι θα επιτελεστεί η επιθυμητή λειτουργία (π.χ. query) αφού δεν υπάρχει κάποια ένδειξη εξαρχής ότι υπάρχει κόμβος με τα δεδομένα που αναζητούνται.



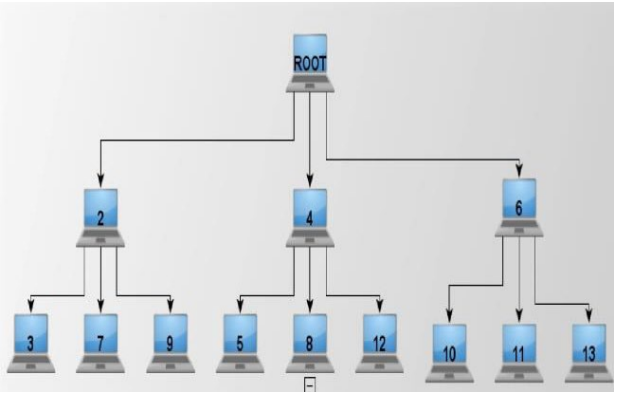
Εικόνα 3 Ένα αδόμητο δίκτυο P2P. Φαίνεται ότι δεν υπάρχει κάποια συγκεκριμένη δομή στον τρόπο όπου είναι συνδεδεμένοι οι κόμβοι-υπολογιστές

Παραπάνω αποτυπώνεται ένα αδόμητο δίκτυο P2P. Αν υποθέσουμε ότι ο κόμβος N θέλει να εκτελέσει ένα query ζητώντας την πληροφορία X, την οποία έχει μόνον ο κόμβος M, τότε θα κατακλύσει το δίκτυο με πακέτα αιτήσεων, μέχρι αυτά να φτάσουν τελικώς στον κόμβο M οποίος με την σειρά του θα απαντήσει. Διαπιστώνεται εύκολα ότι στην συγκεκριμένη περίπτωση θα δημιουργηθεί μεγάλος αριθμός αχρείαστων πακέτων με αποτέλεσμα να επιβαρύνεται ιδιαίτερα ο δίαυλος.

2.2.2 Δομημένο δίκτυο ομότιμων κόμβων

Σε ένα δομημένο δίκτυο P2P, το overlay δίκτυο παρουσιάζει συγκεκριμένη οργάνωση. Το πρωτόκολλο εξασφαλίζει ότι κάθε κόμβος μπορεί να αναζητήσει αποδοτικά, οποιονδήποτε πόρο με την προϋπόθεση ότι αυτός είναι διαθέσιμος σε κάποιον άλλον κόμβο του δικτύου, ανεξαρτήτως από το πόσο σπάνιος είναι αυτός. Η πιο συχνή υλοποίηση των δομημένων δικτύων P2P, αποτελούν αυτά που υλοποιούν ένα DHT (Distributed Hash Table). Σε αυτά χρησιμοποιείται κάποια εκδοχή consistent hashing προκειμένου κάθε αρχείο να "ανατεθεί" σε κάποιον κόμβο. Αυτή η αντιστοίχιση κόμβων του δικτύου με αρχεία επιτρέπει στους ίδιους τους κόμβους να ψάχνουν κάποιο αρχείο/πόρο/δεδομένο στο δίκτυο χρησιμοποιώντας τον πίνακα κερματισμού (hash table). **Πίνακας κερματισμού ή hash table** ονομάζεται ένας πίνακας που περιέχει ζεύγη κλειδιών-τιμών (key, value) και ο οποίος χρησιμοποιείται για την γρήγορη αντιστοίχιση κάποιας τιμής (που αντιστοιχεί σε κάποιο αρχείο για παράδειγμα) στον αντίστοιχο κόμβο που την έχει.

Ενώ η αποδοτικότητα της συγκεκριμένης δομής όσον αφορά την αναζήτησή είναι αυξημένη σε σύγκριση με το αδόμητο δίκτυο, προκύπτουν ορισμένα προβλήματα τα οποία έγκεινται κυρίως στην μεγάλη επιβάρυνση που δέχεται το δίκτυο σε περιπτώσεις προσθήκης ή αναχώρησης κάποιου κόμβου. Σε μία τέτοια περίπτωση, χρειάζεται να γίνει αναδιάταξη των περιεχομένων κάθε κόμβου ενώ και ο πίνακας κατακερματισμού του DHT πρέπει να ενημερωθεί κατάλληλα. [10]



Εικόνα 4 Ένα δομημένο δίκτυο P2P

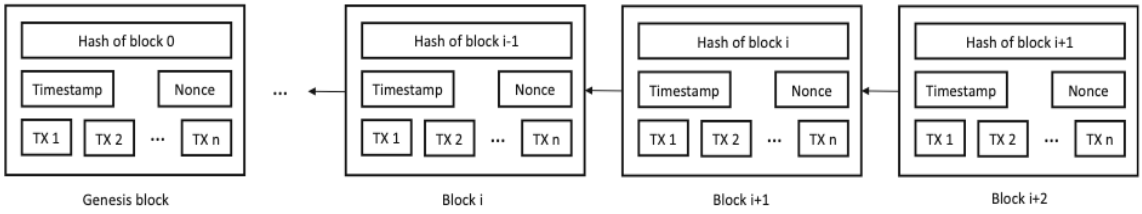
2.3 Τεχνολογία Blockchain

Ως **blockchain** καλείται ένας συνεχώς αυξανόμενος κατάλογος από εγγραφές, οι οποίες καλούνται block, και τα οποία ενώνονται μεταξύ τους μέσω κρυπτογραφίας. Ένα blockchain αποτελεί ουσιαστικά ένα ψηφιακό βιβλίο καταχωρήσεων -ο όρος που χρησιμοποιείται ευρέως είναι ledger- το οποίο δεν δύναται να τροποποιηθεί ως προς το ιστορικό του. Ο όρος blockchain

στα ελληνικά αποδίδεται με ποικίλες ονομασίες όπως αλυσίδα μπλοκ ή αλυσίδα ομάδων συναλλαγών κ.ά. .

Η βασική ιδέα πίσω από το blockchain αποτελεί η δημιουργία ενός καταλόγου- μπλοκ εγγραφών το οποίο να είναι προσβάσιμο σε μία ,όσο διευρυμένη επιθυμείται , κοινότητα έτσι ώστε, υπό φυσιολογική λειτουργία τους blockchain, να μην μπορεί να γίνει οποιαδήποτε αλλαγή στις καταχωρημένες στο μπλοκ αυτό συναλλαγές. Συνήθως, η λειτουργία του blockchain είναι πλήρως αποκεντρωμένη. [11]

Ένα τυπικό παράδειγμα blockchain φαίνεται στο ακόλουθο σχήμα 1. Μία αλυσίδα από blocks, όπου το κάθε block αποτελείται από μία σειρά συναλλαγών (TX 1-n). Το blockchain επεκτείνεται κάθε φορά που προστίθεται ένα νέο block, και έτσι δημιουργείται η αλυσίδα συναλλαγών. [12]



Εικόνα 5 Παράδειγμα ενός blockchain

Εκτός από τις συναλλαγές, κάθε block περιέχει μία χρονοσφραγίδα, το hash value του προηγούμενου block καθώς και έναν τυχαίο αριθμό για να επικυρωθεί (verify) το hash. Αυτή, ακριβώς, η διαδικασία εξασφαλίζει την ακεραιότητα ολόκληρου του blockchain μέχρι και του πρώτου block. (“genesis block”). Οποιαδήποτε αλλαγή στις τιμές των hash θα μπορούσε πολύ εύκολα να ανιχνευτεί αφού αυτή θα προκαλούσε άμεση αλλαγή της επόμενης τιμής hash. Προκειμένου να προστεθεί άλλο ένα block στην αλυσίδα blockchain θα πρέπει πρώτα να συμφωνήσει η πλειονότητα των κόμβων του δικτύου. Αυτό επιτυγχάνεται μέσω ενός μηχανισμού συναίνεσης (consensus mechanism). Σύμφωνα με τον Swanson (2015) [11], μηχανισμός συναίνεσης (consensus mechanism) ονομάζεται η διαδικασία κατά την οποία η πλειονότητα, και σε ορισμένες περιπτώσεις το σύνολο, των κόμβων στο δίκτυο έρχονται σε συμφωνία σχετικά με την κατάσταση του ledger. Επομένως οι συναλλαγές δεν αποτελούν απευθείας μέρος τους blockchain. Αντιθέτως υπάρχει ένα χρονικό διάστημα, σύμφωνα με τον Swanson (2015), [13] κατά το οποίο οι καινούργιες συναλλαγές παραμένουν στο block προτού αποτελέσουν μέρος του blockchain. Στην περίπτωση του Bitcoin, τα καινούρια blocks δημιουργούνται από τους λεγόμενους miners, οι οποίοι επιβραβεύονται με Bitcoins για την συνεισφορά τους στην επικύρωση (validation) των blocks. [14]

Πολύ σημαντικό χαρακτηριστικό ενός δικτύου blockchain αποτελεί το μοντέλο επίτρεψης (permission model) του. Αυτό καθορίζει, κυρίως, ποιος μπορεί να διατηρήσει ένα αντίγραφο του blockchain. Εάν οποιοσδήποτε μπορεί να εκδώσει ένα καινούργιο block προσθέτοντάς το στην αλυσίδα, τότε το blockchain καλείται χωρίς επίτρεψη ή permissionless. Εάν μόνον συγκεκριμένοι χρήστες μπορούν να εκδώσουν μπλοκ τότε το δίκτυο ονομάζεται με επίτρεψη ή permissioned. Σε αναλογία με την λειτουργία του δικτύου μίας επιχείρησης, στο με επίτρεψη

Blockchain δεν μπορεί να συμμετέχει οποιοσδήποτε υπολογιστής. Αντιθέτως σε πλήρη αντιστοιχία με το ελεύθερο διαδίκτυο, το permissioneless blockchain επιτρέπει σε -σχεδόν- όλους τους χρήστες να αποτελέσουν μέρος του. Παρακάτω γίνεται αναλυτική σύγκριση των δυο τεχνολογιών. Ιδιαίτερη αναφορά θα γίνει στο permsissioned blockchain το οποίο αποτελεί και την βάση της συγκεκριμένης εργασίας. [15]

2.3.1 Δίκτυα blockchain χωρίς επίτρεψη

Τα δίκτυα αλυσίδας κορμού χωρίς επίτρεψη αποτελούν αποκεντρωμένες πλατφόρμες πρόσβασης στην αλυσίδα δεδομένων (ledger) οι οποίες επιτρέπουν σε όλους τους συμμετέχοντες να εκδίδουν blocks χωρίς την ανάγκη να ληφθεί κάποια ειδική άδεια από κάποια κεντρική αρχή. Αποτελούν, συνήθως, λογισμικά ανοιχτού κώδικα ελεύθερα στον οποιοδήποτε να τα κατεβάσει.

Το γεγονός ότι οποιοσδήποτε έχει το δικαίωμα να εκδώσει μπλοκ οδηγεί, αυτόματα, στην ιδιότητα ο οποιοσδήποτε να είναι σε θέση να διαβάσει τα περιεχόμενα του blockchain καθώς και να εκδώσει συναλλαγές πάνω σε αυτό. Το τελευταίο, δηλαδή η έκδοση συναλλαγών πάνω στο blockchain, γίνεται μέσω της συμπερίληψης της εκάστοτε συναλλαγής στα προς έκδοση μπλοκ.

Στην ουσία, οποιοσδήποτε χρήστης του δικτύου μέσα στο permissionless blockchain είναι σε θέση να διαβάσει και να γράψει στο ledger. Από την στιγμή που δεν τίθεται κάποιος περιορισμός σχετικά με το ποιος συμμετέχει στο δίκτυο, δίνεται η δυνατότητα και σε κακόβουλους χρήστες να προσπαθήσουν να εκδώσουν μπλοκ με τρόπο τέτοιο ώστε να μην συνάδει με τα ήδη υπάρχοντα blocks. Για αποφυγή τέτοιων ενεργειών, όπως αναφέρθηκε παραπάνω, τα δημόσια blockchain χρησιμοποιούν κάποιο ή κάποια πρωτόκολλα συναίνεσης. Προκειμένου να επιβραβευτούν χρήστες οι οποίοι δημιουργούν νέα blocks τα οποία υπακούν στα συγκεκριμένα πρωτόκολλα, και ως αποτέλεσμα δεν υποκρύπτουν κακόβουλη συμπεριφορά, συνήθως τους δίνεται κάποιο κρυπτονόμισμα, για παράδειγμα στο Blockchain του bitcoin παρέχεται 1 bitcoin. [15]

2.3.2 Δίκτυα blockchain με επίτρεψη

Τα δίκτυα blockchain με επίτρεψη αποτελούν τα δίκτυα blockchain αυτά, στα οποία οι συμμετέχοντες προκειμένου να εκδώσουν block(s) πρέπει να λάβουν την άδεια μιας συγκεκριμένης αρχής (είτε αυτή είναι κεντρική είτε αποκεντρωμένη). Από την στιγμή που μόνον εξουσιοδοτημένοι συμμετέχοντες διατηρούν το blockchain, είναι δυνατόν να περιοριστεί και η δυνατότητα ανάγνωσης των δεδομένων του blockchain. Ακόμη μπορεί να περιοριστεί και η δυνατότητα συναλλαγής μόνο σε συγκεκριμένους συμμετέχοντες του δικτύου. Γενικότερα, υπάρχει πλήρης ελευθερία από τον σχεδιαστή-διαχειριστή του δικτύου σχετικά με το ποιος/ ποιοι θα μπορούν είτε να διαβάζουν τα δεδομένα ή να εκτελούν

συναλλαγές οι με σκοπό αυτές να αποτελέσουν μέρος της αλυσίδας. Ακόμη, τα permissioned blockchain μπορούν να διατηρούνε είτε μέσω open source ή μη λογισμικού. [11]

Τα permissioned μοιράζονται αρκετά κοινά χαρακτηριστικά με τα δημόσια blockchain, αφού αμφότερα μπορούν να έχουν το ίδιο κατανεμημένο και ανθεκτικό χώρο αποθήκευσης των δεδομένων, όπως τα public blockchains. Ακόμη, και τα ιδιωτικά χρησιμοποιούν μοντέλα συναίνεσης. Παρόλα αυτά η βασική διαφορά έγκειται στο ότι τα μοντέλα συναίνεσης αυτά δεν έχουν τόσο μεγάλο κόστος συντήρησης και λειτουργίας όσο αυτά των δημόσιων. Αυτό συμβαίνει καθώς στο σύνολο των περιπτώσεων, προκειμένου να γίνει ένας κόμβος-χρήστης μέρος του δικτύου, απαιτείται μια τυπική ταυτοποίηση του από την διαχειριστική αρχή. Επομένως, η διασύνδεση των κόμβων και η ανταλλαγή των δεδομένων γίνεται πάνω σε μία ελάχιστη βάση εμπιστοσύνης η οποία δεν υπάρχει στο δημόσιο blockchain. Η ύπαρξης αναγνωρισμένης ταυτότητας κάθε συμμετέχοντος δρα ως αντικίνητρο κακόβουλης συμπεριφοράς καθώς σε τέτοια περίπτωση ο "δράστης" δύναται να εκδιωχθεί από το δίκτυο ή να αντιμετωπίσει ακόμη και νομικές κυρώσεις. [16]

Τα μοντέλα συναίνεσης (consensus models) στα permissioned blockchains είναι, συνεπώς, σαφώς γρηγορότερα, αποδοτικότερα και φθηνότερα από τα αντίστοιχα των δημόσιων. Τα permissioned δίκτυα μπορούν να χρησιμοποιηθούν από οργανισμούς οι οποίοι επιθυμούν να χρησιμοποιήσουν την τεχνολογία blockchain προσθέτοντας έναν βαθμό προστασίας και ελέγχου. Συνήθως, η κεντρική αρχή που είναι υπεύθυνη να ελέγχει τους συμμετέχοντες του δικτύου και να καθορίζει τις αρμοδιότητες τους, είναι αξιόπιστη από την πλευρά των συμμετεχόντων. Συχνή είναι και η χρήση των permissioned blockchain δικτύων σε περιπτώσεις κατά τις οποίες διαφορετικοί οργανισμοί επιθυμούν να συνεργαστούν αλλά δεν εμπιστεύονται πλήρως ο ένας τον άλλον. Το μοντέλο συναίνεσης που χρησιμοποιείται σε αυτές τις περιπτώσεις εξαρτάται σε μεγάλο βαθμό από το επίπεδο εμπιστοσύνης μεταξύ των εμπλεκόμενων οργανισμών.

Ορισμένα permissioned blockchain επιτρέπουν την δυνατότητα επιλεκτικής αποκάλυψης των πληροφοριών βάσει της ταυτότητας και των διαπιστευτηρίων κάθε εμπλεκόμενου οργανισμού [17]. Για παράδειγμα, οι πληροφορίες μία συναλλαγής μπορεί να είναι ορατές μόνο στους εμπλεκόμενους οργανισμούς.

Κυριότερα Permissioned blockchains

Όπως αναφέρθηκε παραπάνω, παρόλο το γεγονός ότι το blockchain αρχικά δημιουργήθηκε για την δημιουργία δημόσιων [12] δικτύων χωρίς εμπιστοσύνη των εμπλεκόμενων πλευρών [8] και χωρίς να υπάρχει η ανάγκη για μία κεντρική διαχειριστική αρχή, τα τελευταία χρόνια γίνεται ευρεία χρήση του σε ιδιωτικές πλατφόρμες. Στην συνέχεια θα αναφερθούν ορισμένα από τα βασικότερα από τα frameworks ιδιωτικού – permissioned blockchain. [18]

Το **Hyperledger Fabric**, αποτελεί μία ανοιχτού κώδικα (open source) πλατφόρμα permissioned blockchain (Distributed Ledger Technology – DLT), η οποία έχει σχεδιαστεί

ώστε να χρησιμοποιείται για επαγγελματικούς σκοπούς. [19] Η πλατφόρμα Hyperledger Fabric, η οποία δημιουργήθηκε από το Linux Foundation, επιτρέπει την εκτέλεση κατανεμημένων (distributed) εφαρμογών που καλούνται Dapps και είναι γραμμένες σε γενικού σκοπού γλώσσες προγραμματισμού. Το Hyperledger Fabric χρησιμοποιεί smart contracts τα οποία καλούνται chaincodes και καθορίζουν την λογική εκτέλεσης της εκάστοτε εφαρμογής. Το πρωτόκολλο συναίνεσης που χρησιμοποιείται είναι το voting based πρωτόκολλο Solo. [19]

Το **Quorum** [22] αναπτύχθηκε από την JPMorgan για χρήση σε χρηματοπιστωτικές συναλλαγές, αλλά μπορεί να χρησιμοποιηθεί σε ποικίλες εφαρμογές. Βασίζεται στο Ethereum, και συγκεκριμένα στο go-ethereum με ορισμένες διαφορές. Συγκεκριμένα, επιτρέπει την ιδιωτικότητα (privacy) καθώς είναι δυνατόν να δημιουργηθούν ιδιωτικά smart contracts και συναλλαγές (transactions), των οποίων τα δεδομένα είναι ορατά μόνο σε όσους συμμετέχοντες έχουν αυστηρά καθοριστεί. Ακόμη δίνει την δυνατότητα χρήσης και άλλων πρωτοκόλλων συναίνεσης όπως το Istanbul BFT [23], ενώ έχει και σαφώς καλύτερη απόδοση.

Το **R3 Corda** αποτελεί μία πλατφόρμα ανοιχτού κώδικα που ακολουθεί μία λογική Know Your Customer, δηλαδή κάθε κόμβος προκειμένου να αποτελέσει μέρος τους δικτύου πρέπει να αποδείξει την ταυτότητά του. Το δίκτυο περιλαμβάνει πολλαπλούς notary (σ.σ. συμβολαιογράφους) οι οποίοι είναι κόμβοι υπεύθυνοι για την επικύρωση των συναλλαγών και την εξασφάλιση της μοναδικότητας τους χωρίς να χρειάζεται πολυεκπομπή στο σύνολο των κόμβων του δικτύου. Θα αναφερθούμε το Corda ενδελεχώς στην συνέχεια.

Μορφοποίηση permissionless blockchains

Το **Ethereum** αποτελεί μία δημόσια πλατφόρμα ανοιχτού κώδικα, βασισμένη σε blockchain που επιτρέπει την ανάπτυξη αποκεντρωμένων (decentralized) εφαρμογών. Η βασική ιδέα δημιουργίας του Ethereum το καθιστούσε μία δημόσια permissionless blockchain πλατφόρμα που κάνει χρήση του πρωτοκόλλου συναίνεσης Ethash, το οποίο είναι Proof-of-Work based. Το Ethereum χρησιμοποιείται, όμως, και ως permissioned πλατφόρμα με την κατάλληλη διαμόρφωση. Η διαδικασία αυτή «μετατροπής» του σε permissioned blockchain γίνεται στα ανώτερα στρώματα ανάπτυξης, κυρίως στο στρώμα εφαρμογής [20]. Ορισμένες εταιρίες και οργανισμοί γενικότερα έχουν προσαρμόσει το ανοιχτού κώδικα πρωτόκολλο του Ethereum προκειμένου να τρέξουν το δικό τους ιδιωτικό permissioned δίκτυο.[21] Σε αυτές τις περιπτώσεις, μόνο οι κόμβοι-συμμετέχοντες στους οποίους επιτρέπεται η πρόσβαση στο δίκτυο, μπορούν να δουν τις συναλλαγές.

2.4 Η τεχνολογία Blockchain στον χρηματοοικονομικό τομέα

Η τεχνολογία Blockchain έχει επηρεάσει σημαντικά και τον χρηματοοικονομικό τομέα. Οι διάφορες πλατφόρμες blockchain προσφέρονται για την πραγματοποίηση συναλλαγών, εξασφαλίζοντας την εγκυρότητά τους, κάνοντας χρήση των πρωτοκόλλων συναίνεσης που απαιτούν την επικύρωση από πολλαπλούς κόμβους. Επιτρέπουν στα συμβαλλόμενα μέρη να παρακολουθούν δυναμικά τα περιουσιακά τους στοιχεία και τις διάφορες συναλλαγές-συμφωνίες χρησιμοποιώντας κοινό πρωτόκολλο, αυτοματοποιώντας, ακόμη και πλήρως, οποιεσδήποτε διαδικασίες επικύρωσης ακόμα και τρίτων μερών.

Τα συστήματα blockchain διαθέτουν πολλά χαρακτηριστικά που ευνοούν την χρήση τους σε ένα χρηματοοικονομικό περιβάλλον. Τα συστήματα αυτά είναι ιδιαιτέρως ευέλικτα και μπορούν να λειτουργήσουν ως αποκεντρωμένα δίκτυα χωρίς την απαίτηση ύπαρξης ενός κεντρικού εξυπηρετητή ούτε κάποιας αξιόπιστης, τρίτης ρυθμιστικής αρχής. Εξασφαλίζεται σε μεγάλο βαθμό η αξιοπιστία των δεδομένων στο ledger. Στον τραπεζικό και χρηματοοικονομικό τομέα, η τεχνολογία blockchain παρέχει την δυνατότητα της απλούστευσης των ροών εργασιών (business processes), δημιουργώντας παράλληλα αξιόπιστα αρχεία των συναλλαγών και των συμφωνιών μεταξύ των εμπλεκομένων πλευρών

Έχουν δημοσιευτεί πολλές εργασίες με αντικείμενο την πιθανή χρήση της τεχνολογίας blockchain στον χρηματοοικονομικό τομέα. Οι Aste et al. [24] θεωρούν ότι το blockchain προκαλεί μία μεταστροφή της πηγής εμπιστοσύνης των συστημάτων από τον άνθρωπο στην μηχανή. Εκκινώντας από την μελέτη των χαρακτηριστικών του bitcoin, περιγράφουν την δυνητική μετάβαση της οικονομίας με βάση το συνάλλαγμα, χάρη στο Bitcoin. Οι William et al. [25] ήδη από το 2017 είχαν δει την δυνατότητα χρήσης των εφαρμογών που βασίζονται σε blockchain smart contracts να αλλάξουν ριζικά τον τρόπο εκτέλεσης χρηματοοικονομικών συναλλαγών, νομικών υπηρεσιών και διαδικασιών του κυβερνητικού τομέα. Τέλος, οι Chris Kan et. al. [26] περιγράφουν τις προσπάθειες που είχαν καταβληθεί από την R3 να δημιουργήσει μία κοινοπραξία με πάνω από 80 μέλη (της βιομηχανίας) προκειμένου να βρεθεί κοινό σημείο με γνώμονα την υιοθέτηση τεχνολογιών αλυσίδας συστοιχιών (blockchain) στον χρηματοοικονομικό τομέα. Αναγνωρίζοντας τις βασικές απαιτήσεις της τεχνολογίας Distributed Ledger (DLT), ευνοήθηκε η δημιουργία μία νέας πλατφόρμας blockchain με επίτρεψη η οποία προορίζεται μόνο για χρηματοοικονομικού τύπου συναλλαγές, του Corda με το οποίο θα ασχοληθούμε, εκτενώς, στην παρούσα εργασία στην συνέχεια.

2.5 Blockchain, Know Your Customer και σχετικές εργασίες

Μία από τις διαδικασίες που εκτελούν οι τράπεζες και γενικότερα οι χρηματοπιστωτικοί οργανισμοί αποτελεί η διαδικασία ταυτοποίησης – επαλήθευσης τους πελάτη. Η διαδικασία αυτή, η οποία είναι γνωστή ως Know Your Customer είναι ιδιαίτερα οικονομικά επιζήμια για τους οργανισμούς. Σύμφωνα με τον Thomson Reuters [27] το μέσο ετήσιο κόστος που προκύπτει για τις τράπεζες είναι περίπου 60 εκατομμύρια δολάρια. Το παραπάνω κόστος δύναται να αυξηθεί σημαντικά εξαιτίας των προστίμων που επιβάλλονται στους χρηματοπιστωτικούς οργανισμούς εξαιτίας πιθανής αδυναμίας συμμόρφωσης με τους κανονισμούς της εκάστοτε χώρας (ή κοινοτικούς κανονισμούς). Σύμφωνα με τους Benedict N. Nolens, [28] τα πρόστιμα σε περιπτώσεις εντοπισμού ελλιπών ελέγχων κατά του ξεπλύματος μαύρου χρήματος (AML policies), φτάνουν τα 10 δις. δολάρια. Γίνεται, συνεπώς, συνεχής αναζήτηση για νέους τρόπους απλοποίησης της διαδικασίας κοινοποίησης των δεδομένων KYC από τους πελάτες προς τους οργανισμούς.

Παράλληλα, παρατηρείται και μία ολοένα και αυξανόμενη χρήση της τεχνολογίας Blockchain σε συναλλαγές οικονομικού χαρακτήρα, με κυρίαρχη από αυτές την χρήση κρυπτονομισμάτων για την ανταλλαγή αξίας μεταξύ συμμετεχόντων στο δίκτυο. Μία από τις συχνότερες κριτικές που δέχεται η τεχνολογία των κρυπτονομισμάτων αφορά την έλλειψη μίας αρχής η οποία ελέγχει την προέλευση των πόρων και εντοπίζει περιπτώσεις ξεπλύματος μαύρου χρήματος. Ειδικά η Ευρωπαϊκή Επιτροπή έχει τονίσει την ανάγκη μηχανισμών Know Your Customer μηχανισμών στο blockchain.

Η χρήση της τεχνολογίας Blockchain με σκοπό την απλοποίηση της διαδικασίας Know your Customer έχει προταθεί από πολλούς ερευνητές. Οι Shabair et al. [29] πρότειναν ένα σύστημα proof-of-concept based με σκοπό την διαχείριση private blockchain περιβαλλόντων σε μεγάλης κλίμακας ελέγχους- tests, τονίζοντας την ανάγκη περαιτέρω έρευνας σχετικά με την ασφάλεια και την ιδιωτικότητα των εφαρμογών με χρήση DLT. Ακόμη οι Norvill et al. [30] έχουν παρουσιάσει ένα σύστημα που επιτρέπει την αυτοματοποίηση της κοινοποίησης ιδιωτικών εγγράφων (σχετικών με την διαδικασίας KYC) επιχειρώντας την ελαχιστοποίηση του κόστους και του χρόνου που απαιτεί η διαδικασία Know Your Customer. Οι Kapsoulis et al. [31] έχουν προτείνει ένα αποκεντρωμένο schema το οποίο επιτρέπει την προστασία της ιδιωτικότητας του χρήστη σε Enterprise Blockchains.

Η διαδικασία με την οποία γίνεται η κοινοποίηση των KYC δεδομένων από τους χρήστες προς τους οργανισμούς αποτελεί ιδιαίτερα σημαντικό παράγοντα που καθορίζει όχι μόνο τον βαθμό στον οποίο εξασφαλίζεται η ιδιωτικότητα και η προστασία των ευαίσθητων δεδομένων, αλλά και τις δυνατότητες περαιτέρω προσθήκης λειτουργιών στα προτεινόμενα συστήματα. Μία πιθανή προσθήκη σε μία εφαρμογή ανταλλαγής KYC δεδομένων στο Blockchain προτείνεται από τους Parra Moyano et al.. [32] Στην δημοσίευσή τους προτείνεται ένα σύστημα διαμοιρασμού του κόστους της διαδικασίας επαλήθευσης των KYC δεδομένων η οποία, όπως αναφέρθηκε νωρίτερα, είναι ιδιαίτερα επιζήμια για τους εμπλεκόμενους χρηματοπιστωτικούς οργανισμούς. Το παραπάνω σύστημα βασίζεται στο Blockchain, ενώ μία προτεινόμενη εκδοχή του με χρήση της πλατφόρμας Corda, παρουσιάζεται στο **Κεφάλαιο 5.**

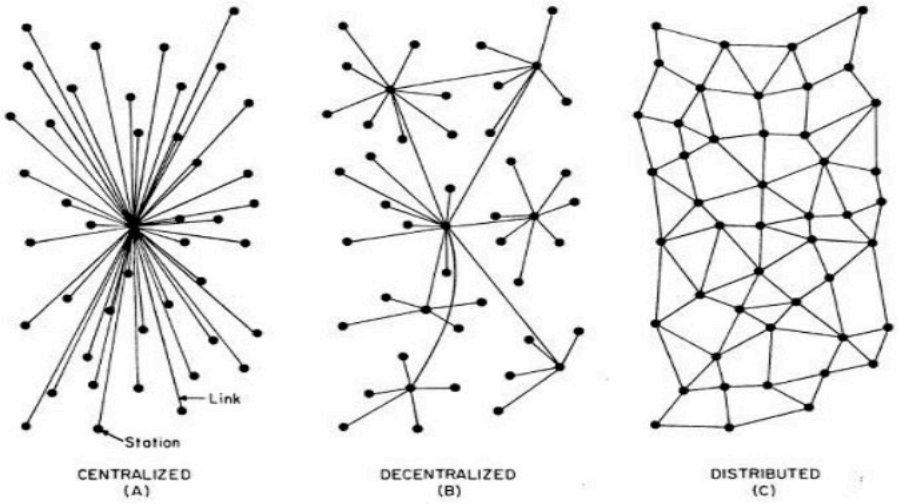
2.6 Τεχνολογίες

2.6.1 R3 Corda

Σε πάρα πολλές βιομηχανίες, σημαντική προσπάθεια απαιτείται προκειμένου να υπάρχει αποτελεσματική οργάνωση και σωστός συγχρονισμός των δεδομένων. Για την ακρίβεια, μεγάλο μέρος της δουλείας που απαιτείται σε έναν οργανισμό αφορά την διαδικασία συγχρονισμού των επιμέρους ροών εργασίας και των δεδομένων που αυτές παράγουν. Η χρήση σχεσιακών βάσεων δεδομένων οι οποίες θα είναι προσβάσιμες από πολλούς οργανισμούς θα επίλυε ορισμένα προβλήματα αλλά θα δημιουργούσε ακόμη περισσότερα. Το κύριο πρόβλημα έγκειται στην ύπαρξη ενός (ή λίγων) διαχειριστών αυτών των βάσεων για τους οποίους δεν εξασφαλίζεται ότι θα κάνουν καλή χρήση.

Αποκεντρωμένες βάσεις δεδομένων vs κατανεμημένες βάσεις δεδομένων:

Μία κατανεμημένη βάση δεδομένων σαν την BigTable μπορεί να χρησιμοποιηθεί για την διαχείριση μεγάλου όγκου δεδομένων, "απλώνοντας" τα δεδομένα σε πολλούς υπολογιστές. Παρόλα αυτά, βασίζεται στην παραδοχή ότι το σύνολο αυτών των υπολογιστών λειτουργούν υπό την διαχείριση μιας οντότητας η οποία είναι και η πλέον αξιόπιστη. Σε μια αποκεντρωμένη βάση δεδομένων, από την άλλη, όπως το Bitcoin, οι κόμβοι κάνουν πολύ λιγότερες παραδοχές εμπιστοσύνης και ελέγχουν την αξιοπιστία των δεδομένων. Αυτό φιλοδοξεί να πετύχει και το R3 Corda.



Εικόνα 6 Η διαφορά μεταξύ μίας Centralized, μίας Decentralized (αποκεντρωμένης) και μίας distributed (κατανεμημένης) βάσης δεδομένων [33]

Δίκτυο ομότιμων κόμβων Corda:

Ένα P2P δίκτυο στην πλατφόρμα Corda αποτελείται από τα εξής χαρακτηριστικά:

- nodes (κόμβοι), λειτουργούν ως parties, επικοινωνούν με χρήση AMQP/1.0 over TLS
- Identity service
- Network map που παρέχει πληροφορίες για το πως θα γίνει η σύνδεση με κόμβους μέσα στο δίκτυο
- Μία ή παραπάνω notary services
- Oracle services, υπηρεσίες που υπογράφουν τις συναλλαγές δίνοντας την εξασφαλίζοντας ότι τα δεδομένα σε αυτές (π.χ. χρόνος ή συναλλαγματική ισοτιμία) είναι ακριβή. Μέσω αυτών γίνεται η επικοινωνία με τον έξω κόσμο.

Δίκτυο - Network:

Ένα δίκτυο Corda είναι στην ουσία ένα δίκτυο ομότιμων κόμβων. Κάθε κόμβος αντιπροσωπεύει μία νομική οντότητα, και εκτελεί το λογισμικό του Corda. Στην ουσία σε κάθε κόμβο είναι εγκατεστημένη μία ή περισσότερες εφαρμογές στο Corda, που ονομάζονται CordApps. Κάθε είδους επικοινωνία μεταξύ των κόμβων γίνεται point-to-point κωδικοποιημένα, χρησιμοποιώντας την ασφάλεια του επιπέδου στρώματος μεταφοράς. Το βασικότερο χαρακτηριστικό του δικτύου είναι το ότι η ανταλλαγή των δεδομένων γίνεται σε μια need-to-know βάση, δηλαδή η πληροφορία μεταφέρεται MONO σε αυτόν που πρέπει να την μάθει. Δεν υπάρχουν, συνεπώς, πολυεκπομπές (broadcasts).

Κάθε κόμβος έχει μία μοναδική, ευρέως γνωστή ταυτότητα. Η ταυτότητα αυτή χρησιμοποιείται για να προσδιοριστεί ένας κόμβος στις συναλλαγές (transactions). Για παράδειγμα, καθώς το Corda χρησιμοποιείται ευρέως για χρηματοπιστωτικές συναλλαγές, όταν πρέπει να προσδιοριστεί ο αγοραστής ενός προϊόντος. Ο χάρτης του δικτύου (network map) συνδέει κάθε οντότητα με μια διεύθυνση IP ώστε να υπάρχει και η δυνατότητα ανταλλαγής μηνυμάτων μεταξύ των κόμβων.

Συμμετοχή στο δίκτυο:

Σε αντίθεση με τις κλασικές μορφές blockchain, τα δίκτυα Corda αποτελούν υβριδικές μορφές, semi-private blockchain στην ίδια λογική με τον ορισμό της συγκεκριμένης κατηγορίας στο **κεφάλαιο 2.3.** Η ένταξη ενός κόμβου-χρήστη στο δίκτυο απαιτεί την έκδοση ενός πιστοποιητικού (certificate) από τον διαχειριστή του δικτύου (network operator). Το συγκεκριμένο certificate, εκδίδεται μετά από διαδικασία ταυτοποίησης του χρήστη και αντιστοιχίζει κάθε κόμβο σε :

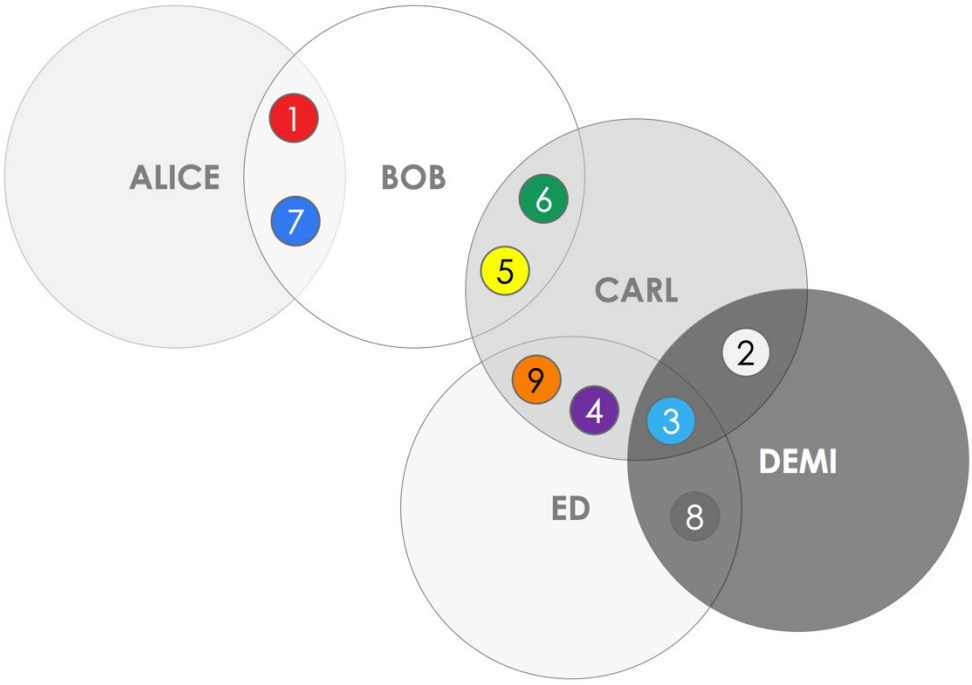
- Μία νομική οντότητα (legal entity) , φυσικό πρόσωπο ή νομικό πρόσωπο
- Ένα public key

Δεδομένα στο blockchain:

Στο Corda, δεν υπάρχει μια μοναδική κεντρική αποθήκη δεδομένων. Σε αντίθεση με τα παραδοσιακά blockchain, κάθε κόμβος διατηρεί την δική του μοναδική βάση δεδομένων η οποία ανά περίπτωση περιέχει το σύνολο των δεδομένων τα οποία ο κόμβος "γνωρίζει", στην ουσία έχει πρόσβαση.

Στην πράξη, το σύνολο των δεδομένων για τα οποία έχει γνώση ο κάθε κόμβος είναι αυτά στα οποία εμπλέκεται. Υπενθυμίζεται ότι η διάδοση των δεδομένων γίνεται σε μια need-to-know βάση. Επομένως δεν αποθηκεύεται τοπικά σε κάθε κόμβο οποιαδήποτε πληροφορία δεν τον αφορά (σε ορισμένες περιπτώσεις θα δούμε παρακάτω ότι μπορεί να υπάρξουν εξαιρέσεις). Για παράδειγμα μια συναλλαγή μεταξύ του Bob και της Alice, όπου ο Bob επιθυμεί την αγορά ενός χρηματοπιστωτικού προϊόντος από την Alice, ο Carl από την στιγμή που δεν είναι ούτε αγοραστής ούτε πωλητής, δεν γνωρίζει τίποτα για την συγκεκριμένη συναλλαγή.

Το ledger, συνεπώς, στο Corda δεν αποτελεί ένα **υποκειμενικό κατασκεύασμα** το οποίο διαφέρει ανάλογα από το ποιος το βλέπει. Στην συνέχεια βλέπουμε σε ένα διάγραμμα Venn ποια δεδομένα είναι ορατά σε κάθε χρήστη.



Εικόνα Τα δεδομένα στο Corda [34]

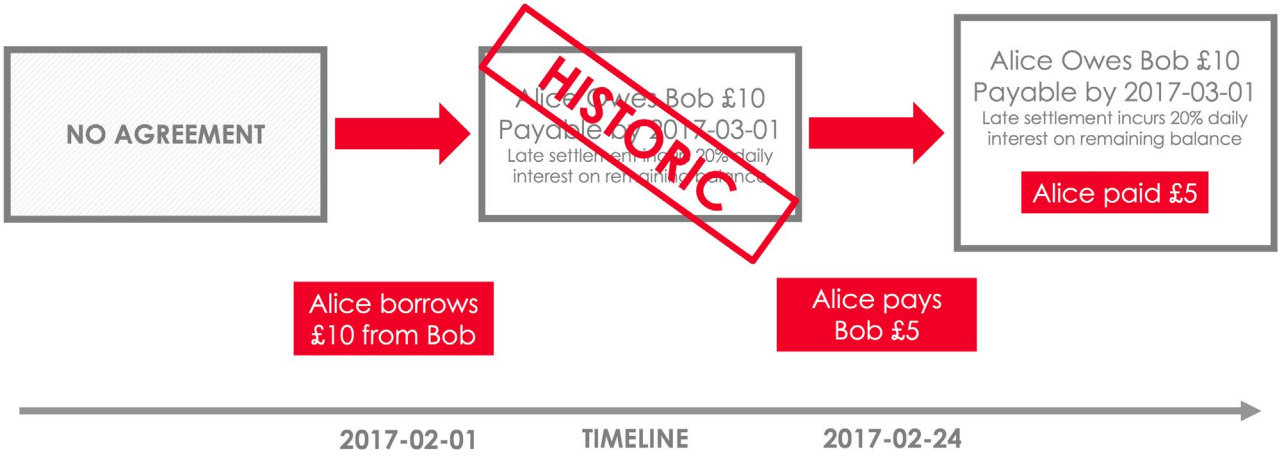
Τα δεδομένα 1,7 είναι ορατά μόνο στους Alice και Bob, τα δεδομένα 5,6 μόνο στον Bob και τον Carl.

Δεν υπάρχει κεντρικό ή γενικό ledger που να περιέχει το σύνολο της πληροφορίας. Αντίθετα, κάθε κόμβος διατηρεί το δικό του, μοναδικό, vault που περιέχει μόνον την πληροφορία εκείνη στην οποία έχει πρόσβαση.

States:

Τα States αποτελούν τις ατομικές μονάδες πληροφορίας στο Corda. Δεν μπορούν να μεταβληθούν με κανέναν τρόπο. Δηλώνονται είτε unconsumed (δεν έχουν καταναλωθεί) είτε consumed (δηλαδή έχουν καταναλωθεί) και επομένως δεν είναι πλέον σε ισχύ. Σε αντίθεση με το Bitcoin blockchain, η βάση δεδομένων του Corda μπορεί να περιέχει πληθώρα δεδομένων (π.χ. πληροφορίες μετοχών, ομολόγων) και όχι μόνον ένα ζεύγος (key,value). Για αυτόν τον λόγο μιλάμε για State και όχι row. Σε αντιστοιχία με το bitcoin, τα States είναι συνδεδεμένα με bytecodes programs τα οποία είναι υπεύθυνα να απορρίψουν ή να δεχτούν μία συναλλαγή (βλ. Συναλλαγές). Όμως σε αντίθεση με το bitcoin, στο Corda ελέγχονται αμφότερες τα States που εισέρχονται και εξέρχονται από μια συναλλαγή.

Ο προσδιορισμός ενός State γίνεται μέσω της αναφοράς στην συναλλαγή (transaction) της οποίας το output ήταν το συγκεκριμένο State.



Εικόνα 7 Το ledger στο Corda [29]

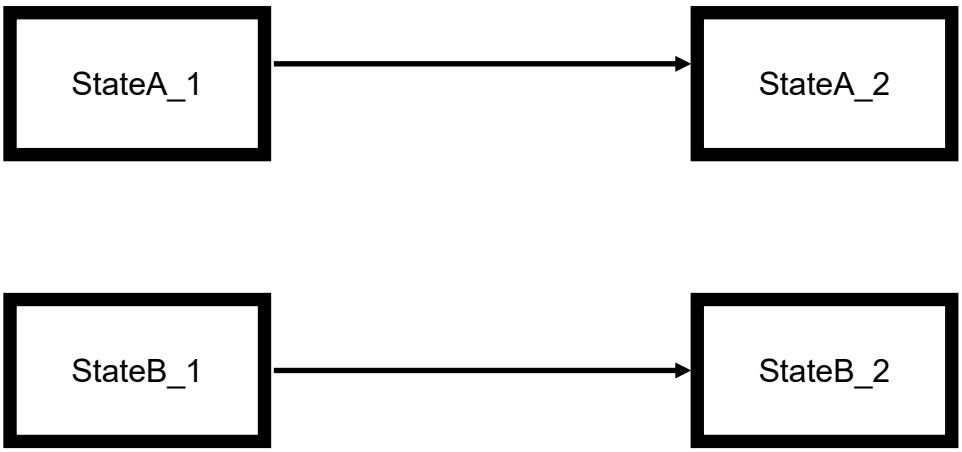
Κάθε χρήστης-κόμβος αποθηκεύει το σύνολο των States τα οποία είναι unconsumed αλλά και αυτά που θεωρούνται historic, δηλαδή έχουν καταναλωθεί, στο δικό του vault. Στην ουσία το vault κάθε χρήστη αποτελεί την εικόνα που αυτός έχει για το ledger.

Συναλλαγές στο Corda:

Το Corda χρησιμοποιεί το μοντέλο UTXO (unspent transaction output) στο οποίο κάθε State δεν μπορεί να μεταβληθεί. Οι συναλλαγές (transactions) χρησιμεύουν ώστε να ενημερώσουν το ledger κάνοντας τα παρακάτω:

- Λαμβάνοντας 0 ή περισσότερες unconsumed States οι οποίες βρίσκονται στο ledger μέχρι την στιγμή εκείνη και σημειώνοντας αυτές ως historic

- Δημιουργώντας 0 ή περισσότερα καινούργια States.



Εικόνα 8 Μία συναλλαγή μπορεί να περιλαμβάνει παραπάνω από μία μετάβαση States [29]

Μία συναλλαγή στο Corda περιέχει τα ακόλουθα:

References στα input States που καταναλώνονται. Αποτελούν ζεύγη (hash, output index) που δείχνουν στα States που καταναλώνονται από την συναλλαγή.

Output States. Τα καινούργια States που δημιουργούνται. Μαζί ορίζεται και το notary που θα επικυρώσει την συναλλαγή καθώς και το contract (βλ. contract) που καθορίζει ποιες αλλαγές είναι αποδεκτές όσον αφορά την δημιουργία του νέου State.

Attachments. Οι συναλλαγές μπορούν να περιέχουν και μία λίστα από hashes αρχείων zip. Τα αρχεία αυτά δύναται να περιέχουν είτε κώδικα και δεδομένα για την συναλλαγή, όπως ημερολόγια, πληροφορίες ζωνών ώρα κτλ. Επίσης μπορούν να περιέχουν και αρχεία που δεν επηρεάζουν την έκβαση της συναλλαγής.

Commands. Κάθε συναλλαγή ενδέχεται να μπορεί να κάνει μία σειρά από αλλαγές σε κάθε input State. Μέσω των commands, καθορίζεται ποια στοιχεία της συναλλαγής θα ελεγχθούν προκειμένου αυτή να κριθεί έγκυρη ή άκυρη.

Signatures. Οι υπογραφές που απαιτούνται προκειμένου να πραγματοποιηθεί η συναλλαγή. Οι υπογραφές αναφέρονται σε parties που πρέπει να συμφωνήσουν.

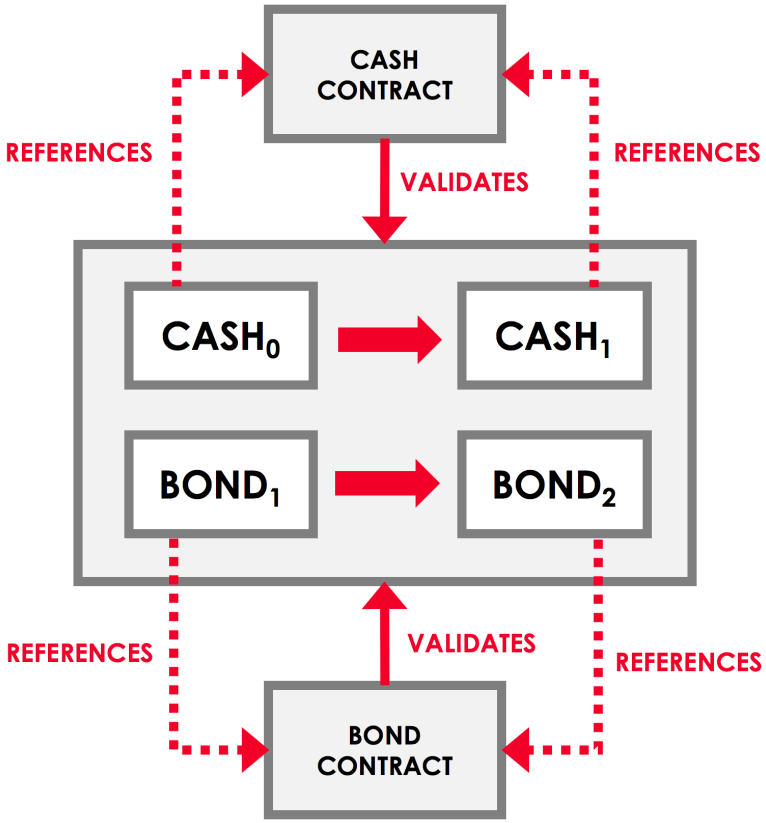
Notary. Τα notaries (ελλ. συμβολαιογράφοι) είναι οι υπηρεσίες αυτές που καθιστούν το corda μία εφαρμογή βασισμένη σε blockchain. Επιτρέπουν στην εφαρμογή να εξασφαλίσει ότι σε κάθε συναλλαγή τα States που καταναλώνονται ΔΕΝ έχουν ήδη καταναλωθεί. Επιτρέπουν έτσι την αποφυγή double spending.

Contracts:

Μία συναλλαγή στο Corda είναι έγκυρα αν και μόνο αν έχει λάβει το σύνολο των υπογραφών που πρέπει να λάβει και αν είναι έγκυρη σύμφωνα με το **contract.** Υπενθυμίζεται ότι στο Corda γίνεται χρήση του UTXO μοντέλου. Ο ορισμός του smart contract που χρησιμοποιείται στο Corda ταιριάζει με τον ορισμό των Clack, Bakshi, Braine, [35] δηλαδή τα έξυπνα συμβόλαια αποτελούν μια συμφωνία της οποίας η εκτέλεση είναι αυτοματοποιήσιμη από κώδικα υπολογιστή που εκτελείται με είσοδο καθορισμένη από έναν ή περισσότερους ανθρώπους αλλά ταυτόχρονα τα δικαιώματα και οι υποχρεώσεις του συμβολαίου είναι νομικά καθορισμένες.

Η εγκυρότητα σύμφωνα με το smart contract καθορίζει ότι:

- Κάθε συναλλαγή πρέπει να υπακούει σε ένα συγκεκριμένο συμβόλαιο.
- Κάθε συμβόλαιο λαμβάνει ως είσοδο μία συναλλαγή και αποφαίνεται αν αυτή είναι έγκυρη ή όχι
- Μία συναλλαγή είναι έγκυρη αν και μόνο αν κάθε input και κάθε output αυτής είναι έγκυρο



Εικόνα 9 Μία συναλλαγή και το αντίστοιχο Smart Contract που καθορίζει τις επιτρεπόμενες μεταβάσεις [29]

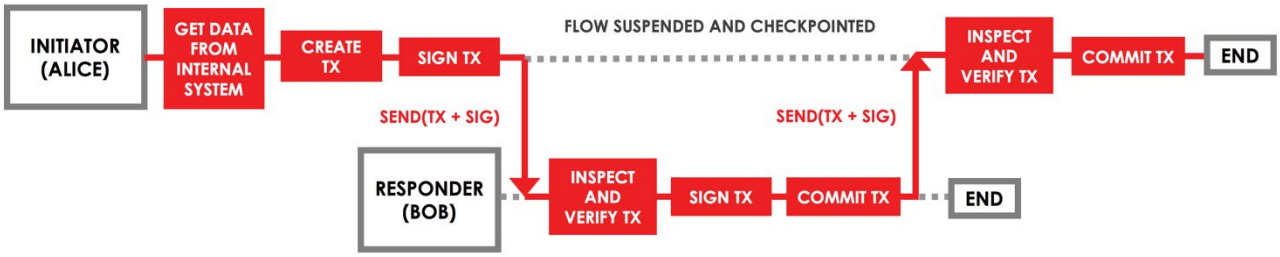
Η επικύρωση μιας συναλλαγής πρέπει να είναι ντετερμινιστική. Αυτό σημαίνει ότι θα πρέπει πάντοτε να αποδέχεται ή πάντοτε να απορρίπτει οποιαδήποτε συναλλαγή. Δεν επηρεάζεται η έκβαση της επικύρωσης από κάποιον εξωτερικό παράγοντα όπως για παράδειγμα η ώρα που συνέβη η συναλλαγή. Η παραπάνω προϋπόθεση είναι απαραίτητη προκειμένου να εξασφαλιστεί ότι όλοι οι συμμετέχοντες ενός δικτύου ομότιμων κόμβων μπορούν να φτάσουν σε συναίνεση σχετικά με την εγκυρότητα μίας ενημέρωσης του ledger.

Δεδομένου ότι πρέπει όλοι κόμβοι ανά πάσα στιγμή να συμφωνούν εάν μία συναλλαγή είναι έγκυρη ή όχι, πρέπει το JVM bytecode που χρησιμοποιείται να είναι πλήρως ντετερμινιστικό. Καθώς το κλασικό JVM δεν είναι πλήρως ντετερμινιστικό (π.χ. external input, random number generators, διαφορές στην αριθμητική floating point σε διαφορετικά συστήματα), το Corda κάνει χρήση ενός JVM sandbox το οποίο είναι πλήρως ντετερμινιστικό.

Flows:

Τα δίκτυα του Corda, όπως έχει αναφερθεί και παραπάνω, δεν χρησιμοποιούν το λεγόμενο global broadcasting, δηλαδή την αποστολή δεδομένων σε όλους τους κόμβους του δικτύου, αλλά αντιθέτως στέλνουν μηνύματα σε συγκεκριμένους επιλεγμένους κόμβους του δικτύου.

Ακριβώς αυτή η ιδιότητα της πλατφόρμας Corda, καθιστά απαραίτητο ο προγραμματιστής να καθορίσει αυστηρά ποια πληροφορία θα σταλεί, σε ποιους κόμβους και με ποια σειρά.
Το API του Corda παρέχει την δυνατότητα στον προγραμματιστή να αυτοματοποιήσει τις παραπάνω διαδικασίες χρησιμοποιώντας καθορισμένες ροές εργασιών, τα **flows.** Flow ονομάζεται στο Corda μία αλληλουχία από βήματα τα οποία καθορίζουν πως ένας κόμβος θα πετύχει μία συγκεκριμένη ενημέρωση του ledger. Ακολούθως φαίνεται ένα παράδειγμα αλληλουχίας των βημάτων



Εικόνα 10 Μία ροή εργασιών όπως ορίζεται στο Corda API [29]

Αφού καθοριστεί η σειρά των βημάτων που απαιτούνται για την εκτέλεση μίας συγκεκριμένης ενημέρωσης του ledger μέσω της δημιουργίας ενός flow, η αντίστοιχη εφαρμογή CordApp η οποία περιλαμβάνει το (ή τα) flow, εγκαθίσταται μέσα σε έναν κόμβο. Ο διαχειριστής του κόμβου, στην συνέχεια, μπορεί ανά πάσα στιγμή να ξεκινήσει την διαδικασία που ορίζει το flow μέσω μίας κλήσης RPC.
Σε αντίθεση με τα Contracts, τα flows δεν εκτελούνται σε ένα sandbox, επομένως οι κόμβοι μπορούν να προβούν σε διαδικασίες όπως η δικτύωση ή το I/O.

Subflows:

Τα flows μπορούν να περιλαμβάνουν ως κάποιο βήμα στην εκτέλεσή τους, την εκκίνηση ενός ή περισσοτέρων διαφορετικών flows. Κάθε flow που εκκινείται μέσα σε ένα άλλο flow ονομάζεται subFlow. Το αρχικό flow που εκκίνησε το subFlow θα περιμένει μέχρι να ολοκληρωθεί η εκτέλεση του subFlow.

Όπως θα δούμε και στην ανάπτυξης της προτεινόμενης από την παρούσα διπλωματική εφαρμογής, το API του Corda παρέχει μία βιβλιοθήκη από πολλές διαφορετικές Subflows που εκτελούν κοινές εργασίες που απαιτούνται συχνά στα flows του Corda. Αυτές περιλαμβάνουν:

- Την επικύρωση (notarizing) μία συναλλαγής (transaction)
- Την συλλογή υπογραφών από ένα σύνολο κόμβων

Vault:

Κάθε κόμβος που συμμετέχει στο δίκτυο Corda διαθέτει έναν τοπικό χώρο αποθήκευσης που ονομάζεται Vault (θησαυροφυλάκιο). Το Vault κάθε κόμβου περιέχει το σύνολο των δεδομένων που έχουν εξαχθεί από το ledger και θεωρείται σχετιζόμενο με τον συγκεκριμένο κόμβο. Το σύνολο των δεδομένων, αυτό, αποθηκεύεται σε μία τοπική βάση δεδομένων που κάνει χρήση του σχεσιακού μοντέλου ώστε να μπορεί εύκολα να πραγματοποιήσει αιτήματα (queries). Το Vault, συνεπώς, αποθηκεύει τόσο τις CONSUMED όσο και τις UNCONSUMED States που αφορούν τον εκάστοτε κόμβο. [14] [28]

Cordapp Design Language (CDL):

Η Cordapp Design Language (CDL) αποτελεί ένα σύνολο από διαγράμματα που καθορίζουν με πολύ συγκεκριμένο τρόπο την σχεδίαση μίας εφαρμογής σε Corda (CorDapp). Υπάρχουν 3 διαφορετικές όψεις του CDL: η Smart Contract View, η Ledger Evolution View και η Business Process Modelling Notation (BPMN). Στην παρούσα διπλωματική, παρουσιάζονται αρκετά διαγράμματα σε CDL. Παρακάτω παρατίθεται μία επεξήγηση για πιο εύκολη κατανόηση των διαγραμμάτων.

Smart Contract View

Αποτελεί την βασική όψη του CDL. Συνοψίζει την βασική σχεδίαση των Smart Contracts στο Corda. Στο Corda, τα smart contracts διαφέρουν από τα κλασικά Smart Contracts σε διαφορετικά blockchain. Αποτελούνται από μία ή περισσότερες States (όπως αναφέρθηκε και νωρίτερα) που αντιπροσωπεύουν δεδομένα στο ledger καθώς και από τα Corda contracts τα οποία καθορίζουν τι μπορεί να συμβεί με αυτά τα States. Η όψη Smart Contract προσομοιάζει με μία μηχανή (αυτόματο) πεπερασμένων καταστάσεων (Finite State Machine). Προτού γίνει επέκταση στον τρόπο που είναι δομημένο το συγκεκριμένο διάγραμμα, τονίζεται, ξανά ο τρόπος με τον οποίο λειτουργεί η πλατφόρμα Corda. Σε άλλα Blockchain, για παράδειγμα το Ethereum [20], οι δυνατότητες του χρήστη αναφορικά με το τί επιτρέπεται να κάνει με το Solidity Smart Contract εξαρτώνται, πλήρως, από τις συναρτήσεις (μεθόδους) που είναι διαθέσιμες στον χρήστη του Smart Contract. Αντιθέτως, το Corda παρέχει πολύ μεγαλύτερη ευελιξία στον προγραμματιστή. Συγκεκριμένα δεν καθορίζει τι επιτρέπεται αλλά επιτρέπει στον χρήστη οποιαδήποτε συναλλαγή με βασική προϋπόθεση να υπακούει στους κανόνες που ορίζει το Contract.

Η δομή του διαγράμματος CDL Smart Contract View στοχεύει στο να βοηθήσει τον σχεδιαστή των Contracts να εξασφαλίσουν ότι δεν υπάρχουν «κενά» ασφαλείας στην σχεδίαση του. Βασικό στοιχείο του διαγράμματος αποτελεί το State, του οποίου η εξέλιξη παρουσιάζεται στο ίδιο το διάγραμμα. Στο State εκτός από το **όνομα** του, παρουσιάζεται και η κατάσταση ή status στην οποία βρίσκεται. Σημαντικό στοιχείο αποτελεί η πολλαπλότητα των συναλλαγών. Συγκεκριμένα κατά την σύνδεση δύο States ο αριθμός στις δύο άκρες του καθορίζει τον αριθμό

των States που μπορεί να αποτελούν input και output αντίστοιχα στην κάθε συναλλαγή. Όσον αφορά τα πεδία με τους περιορισμούς παρουσιάζονται στον ακόλουθο πίνακα.

Constraint	Περιορισμοί που καθορίζονται
Signing Constraints (SC)	Καθορίζει ποιο ή ποια Parties πρέπει να υπογράψουν την συναλλαγή (transaction) που οδηγεί στην συγκεκριμένη μετάβαση της κατάστασης. Περιγράφεται υπό την μορφή input.<Party> ή output.<Party>
Universal Constraints	Καθορίζει τους καθολικούς περιορισμούς που πρέπει να ικανοποιούνται ανεξάρτητα από το Status που βρίσκεται το State.
Status Constraints	Καθορίζει τους περιορισμούς στους οποίους υπόκειται το State όταν βρίσκεται σε μία συγκεκριμένη κατάσταση (status). Είναι συνήθως διαφορετικοί οι περιορισμοί για κάθε διαφορετικό Status.
Flow Constraints	Καθορίζει την γενικότερη λειτουργία του flow στο οποίο ανήκει η συγκεκριμένη συναλλαγή που παρουσιάζεται στο διάγραμμα. Δεν περιέχει περιορισμούς που υλοποιούνται στο Smart Contract.

Ledger Evolution View

Η Ledger Evolution όψη μας δείχνει την ακριβή σειρά των συναλλαγών (transactions) του State οι οποίες υπακούν στο σύνολο των κανόνων που καθορίζονται από τα Contracts. Αποτελούνται από τα States στα οποία είναι ορατά τα πεδία τους (με τυχαίες συνήθως τιμές οι οποίες υπακούν στους περιορισμούς). Σκοπός της συγκεκριμένης όψης είναι η αποτύπωση της εξέλιξης των δεδομένων που βρίσκονται, υπό την μορφή States, στο Ledger.

2.6.2 Interplanetary File System (IPFS)

Το Inter Planetary File System ή IPFS, όπως καλείται συχνότερα, αποτελεί ένα κατανεμημένο σύστημα για την αποθήκευση και την πρόσβαση σε δεδομένα, αρχεία ιστοσελίδες και εφαρμογές. Χρησιμοποιεί την λεγόμενη content-addressable storage, η οποία επιτρέπει την αναφορά και τον εντοπισμό ενός αρχείου βάσει του περιεχομένου του και όχι του ονόματος του. Το IPFS λειτουργεί με τρόπο αρκετά παρόμοιο με το BitTorrent, αποτελώντας μία peer to peer πλατφόρμα. Σε αντίθεση με έναν centralized σύστημα, η

αρχιτεκτονική του IPFS βασίζεται στον κατακερματισμό του συνόλου των δεδομένων. Στο σύστημα αυτό, οι χρήστες-συμμετέχοντες κατέχουν οι ίδιοι μία μερίδα από τα συνολικά δεδομένα, δημιουργώντας ένα ανεκτικό σε σφάλματα σύστημα αποθήκευσης και διαμοιρασμού δεδομένων. Οποιοσδήποτε χρήστης μπορεί να ανατρέξει σε οποιοδήποτε αρχείο που υπάρχει στο δίκτυο, κάνοντας ένα αίτημα. Η χρήση ενός DHT (Distributed Hash Table), δηλαδή ενός πίνακα κατακερματισμού, επιτρέπει την εύρεση των κόμβων που διαθέτουν το εκάστοτε αναζητούμενο αρχείο.

Η χρήση του IPFS στις τεχνολογίες blockchain παρέχει σημαντικά πλεονεκτήματα. Παρότι το πρωτόκολλο του IPFS παρέχει σχετικά απλή ασφάλεια στα δεδομένα αυτά καθαυτά, η προστασία της ιδιωτικότητας παρέχεται μέσω των μηχανισμών κρυπτογράφησης που παρέχει. Συγκεκριμένα, κάθε μονάδα πληροφορίας που αποθηκεύεται στο IPFS, για παράδειγμα αρχείο ή εφαρμογή, αποκτά ένα μοναδικό hash που καλείται content id ή CID (π.χ. QmQ3GriRSNUrmF67GN7f3dq8x2UDku8KYWZbhvrTtt29fW) το οποίο οδηγεί στα δεδομένα. Επομένως, προκειμένου να έχει κάποιος πρόσβαση στα αποθηκευμένα δεδομένα, πρέπει να γνωρίζει το μοναδικό CID του αρχείου. Σημειώνεται ότι η διαδικασία απόκτησης του hash key αποτελείται από μία μονόπλευρη συνάρτηση (one-way function) και δεν είναι δυνατόν, σε ρεαλιστικές συνθήκες, να βρεθεί αντίστροφη της.

Κεφάλαιο 3

Αρχιτεκτονική Εφαρμογής

3.1 Γενική περιγραφή αρχιτεκτονικής εφαρμογής

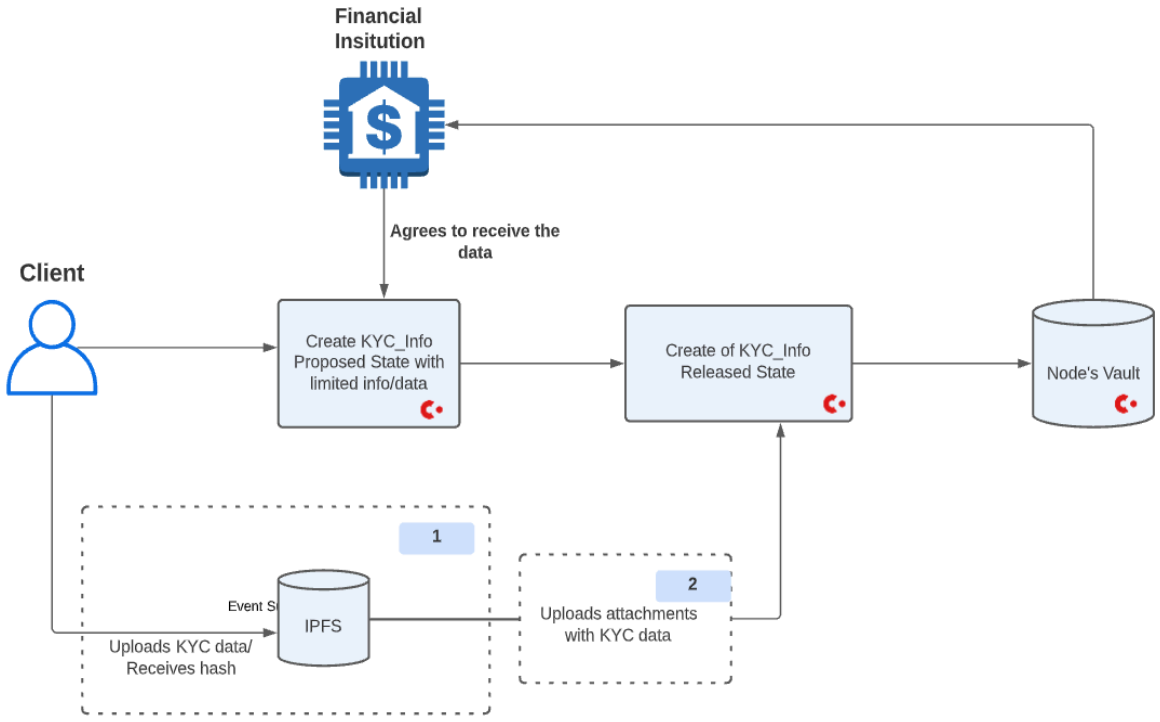
Σκοπός της παρούσας εργασίας είναι η ανάπτυξη μίας εφαρμογής **(CordApp)** με χρήση blockchain στην πλατφόρμα με επίτρεψη Corda, η οποία επιτρέπει τον διαμοιρασμό ευαίσθητων KYC δεδομένων μεταξύ συμμετεχόντων σε ένα δίκτυο.

Η πλατφόρμα Corda είναι πλατφόρμα blockchain με επίτρεψη. Συνεπώς οι συμμετέχοντες στο εκάστοτε δίκτυο P2P προκειμένου να συμμετέχουν σε αυτό, λαμβάνουν πρώτα την έγκριση της κεντρικής οντότητας. Στην παρούσα εργασία, δεν γίνεται αναφορά σχετικά με την αυθεντικοποίηση των ίδιων των συμμετεχόντων ενώ θεωρείται πως αυτοί έχουν εισέλθει στο δίκτυο πληρώντας τα όποια κριτήρια έχουν θεσπιστεί από τον διαχειριστή του δικτύου. Συγκεκριμένα, αρχικά αναλύεται η βασική υλοποίηση της εφαρμογή με την χρήση 2 διαφορετικών τρόπων αποθήκευσης και διαμοιρασμού των δεδομένων, στην συνέχεια μελετάται η απόδοση του συστήματος καθώς ορισμένοι παράμετροι μεταβάλλονται, ενώ τέλος προτείνεται μία εφαρμογή που διαμοιράζει το κόστος της αυθεντικοποίησης KYC μεταξύ των χρηματοπιστωτικών οργανισμών που κάνουν χρήση αυτών.

Ένα δίκτυο Corda αποτελείται από κόμβους οι οποίοι απαρτίζουν ένα P2P δίκτυο. Στην παρούσα διπλωματική θεωρείται ότι το δίκτυο αποτελείται από ένα σύνολο ενός ή περισσοτέρων χρηματοπιστωτικών οργανισμών, καθώς και ενός ή παραπάνω πελατών. Μία από τις συχνότερες συναλλαγές μεταξύ ενός πελάτη και ενός χρηματοπιστωτικού οργανισμού αποτελεί η αποστολή – κοινοποίηση εκ μέρους του πελάτη, μίας σειράς εγγράφων τα οποία έχουν ως σκοπό την νομική αυθεντικοποίηση του χρήστη. Η συγκεκριμένη διαδικασία είναι γνωστή ως Know Your Customer (KYC). Καθώς οι πελάτες επιθυμούν να κάνουν χρήση κάποιων εκ των υπηρεσιών που προσφέρουν οι χρηματοπιστωτικοί οργανισμοί, η παραπάνω διαδικασία είναι απαραίτητη.

Ο πελάτης, αρχικά, συλλέγει το σύνολο των KYC δεδομένων που επιθυμεί να κοινοποιήσει. Αυτά μπορεί να είναι αντίγραφα των φορολογικών του δηλώσεων, φορολογικές ενημερότητες, η ταυτότητα κλπ. Επίσης, ο πελάτης πρέπει να αποφασίσει σε ποιόν ή ποιους χρηματοπιστωτικούς επιθυμεί να κοινοποιήσει αυτά τα έγγραφα. Η ταυτότητα των οργανισμών είναι γνωστή στον πελάτη. Δεν έχει νόημα να κοινοποιήσει τα έγγραφα σε χρηματοπιστωτικούς οργανισμούς με τους οποίους δεν έχει έρθει σε συνεννόηση, καθώς όπως θα γίνει κατανοητό και στην συνέχεια, αυτοί θα απορρίψουν την συναλλαγή.

Εξετάζονται δύο βασικοί τρόποι κοινοποίησης των δεδομένων μέσω Corda. Στον πρώτο τρόπο, ο πελάτης κάνει χρήση του InterPlanataryFileSystem (IPFS), αποθηκεύοντας σε αυτό το σύνολο των δεδομένων προς κοινοποίηση. Στο ledger του Corda δεν κοινοποιούνται τα δεδομένα αυτά καθαυτά. Αντιθέτως, αποθηκεύεται και αντιστοίχως κοινοποιείται κατά βούληση του πελάτη, το hash του IPFS το οποίο περιλαμβάνει το σύνολο των δεδομένων. Στον δεύτερο τρόπο, γίνεται χρήση των attachments, μίας λειτουργίας του Corda, που επιτρέπει την αποθήκευση δεδομένων τοπικά στους κόμβους. Το ακόλουθο διάγραμμα μας δείχνει συνολικά την υλοποίηση.



Διάγραμμα 1 Η αρχιτεκτονική του κορμού της εφαρμογής με υλοποίηση μέσω IPFS (1) και μέσω attachments (2)

Στην συνέχεια αναλύονται οι δύο βασικές αρχιτεκτονικές σχετικά με τον τρόπο αποθήκευσης των δεδομένων KYC. Τόσο η αρχιτεκτονική μέσω IPFS όσο και η αρχιτεκτονική με αποστολή δεδομένων μέσω attachments εξασφαλίζουν την ιδιωτικότητα που επιδιώκεται για την συγκεκριμένη εφαρμογή.

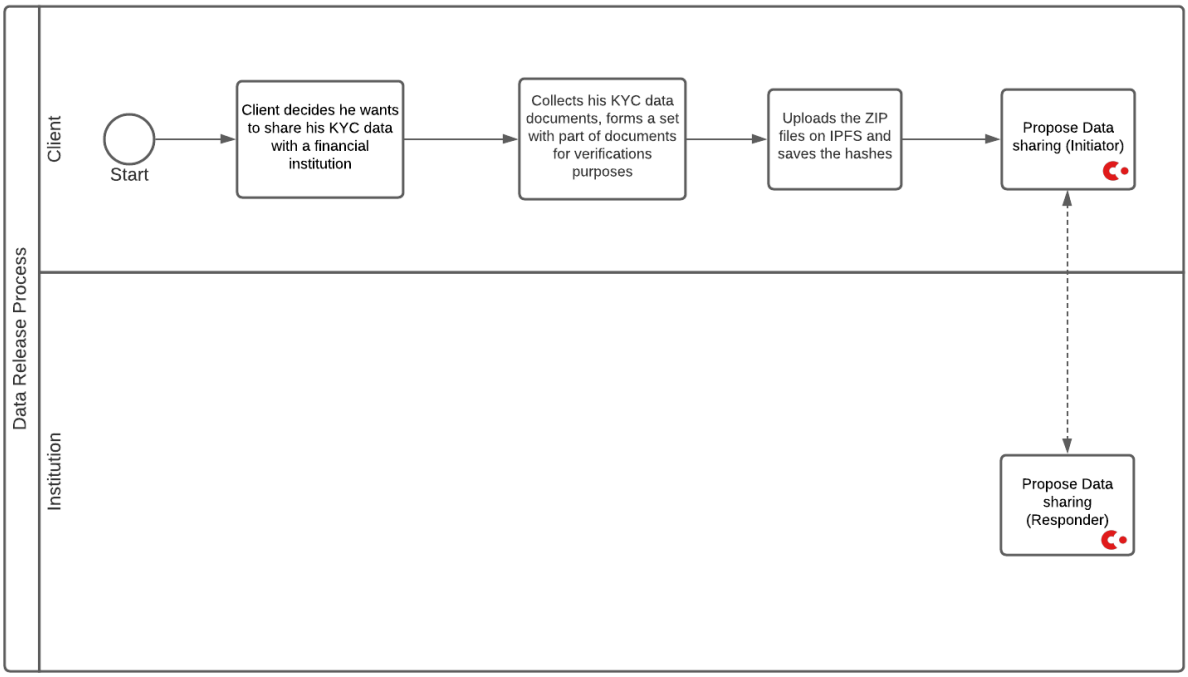
3.2 Αρχιτεκτονική εφαρμογής (Cordapp) με χρήση IPFS

3.2.1 Περιγραφή της δομής και λειτουργίας της εφαρμογής

Βασικοί συμμετέχοντες του δικτύου αποτελούν ο πελάτης (client) που επιθυμεί να διαμοιράσει τα δεδομένα του, και ο οργανισμός (στο εξής institution) που επιθυμεί να λάβει τα δεδομένα αυτά, να τα επεξεργαστεί υλοποιώντας έλεγχο KYC και να αυθεντικοποιήσει ή όχι τον πελάτη.

Αρχικά, ο πελάτης αποθηκεύει δύο σύνολα με αντίγραφα των δεδομένων του στο σύστημα διαμοιρασμού αρχείων IPFS. Στην συνέχεια, ξεκινάει μία ροή εργασιών στο corda (βλ. flows) προκειμένου να διαθέσει τα δεδομένα αυτά στους εξουσιοδοτημένους από τον ίδιο συμμετέχοντες του δικτύου. Γίνεται χρήση, της ιδιότητας του Corda να μην χρησιμοποιεί gossip protocol αλλά να γίνεται διαμοιρασμός των δεδομένων σε μία need-to-know basis μόνο.

1ο βήμα: Ο πελάτης αποθηκεύει ένα **υποσύνολο** των δεδομένων προς παραχώρηση, στο IPFS και στην συνέχεια λαμβάνει ένα hash για τα συγκεκριμένα δεδομένα. Στην συνέχεια δημιουργεί ένα State για τα δεδομένα αυτά στο Corda βάζοντας ως participants (δηλαδή συμμετέχοντες) τον εαυτό του (client) καθώς και τον χρηματοπιστωτικό οργανισμό. Η διαδικασία δημιουργίας τους State διέπεται από το πρώτο contract (βλ. Propose_contract). Σκοπός της διαδικασίας αυτής είναι να γίνει η αναγνώριση του πελάτη από κάθε οργανισμό. (βλ. Propose_flow)



Διάγραμμα 2 Το BPMN διάγραμμα που απεικονίζει την αρχή της διαδικασίας κοινοποίησης των KYC δεδομένων

2ο βήμα: Αφού ο οργανισμός έχει κάνει δεκτή την έκδοση του State που περιέχει τα αναγνωριστικά δεδομένα για τον πελάτη, ο πελάτης έχει την δυνατότητα να αλλάξει το State, αντικαθιστώντας το IPFS hash που αυτό περιέχει, με το IPFS hash που αντιστοιχεί στο σύνολο των δεδομένων προς έλεγχο. Με αυτόν τον τρόπο, δεδομένου ότι οι μόνοι participants είναι ο πελάτης και ο οργανισμός, εξασφαλίζεται η ιδιωτικότητα των KYC δεδομένων.

Διάγραμμα 3 Ο χρήστης, αφού λαμβάνει το IPFS hash, δημιουργεί το νέο State

3ο βήμα: Στην συνέχεια, ο πελάτης έχει την δυνατότητα να προσθέσει έναν ή παραπάνω συμμετέχοντες στο δίκτυο, δίνοντας τους πρόσβαση στα δεδομένα του. Αυτοί, εάν κάνουν την συναλλαγή, αποκτούν πρόσβαση στα δεδομένα και σε οποιαδήποτε ενημέρωση αυτών.

Στην συνέχεια αναλύονται αναλυτικά οι σχεδιαστικές επιλογές σχετικά με τα βασικά στοιχεία της εφαρμογής CordApp, δηλαδή το State, το Contract και τα Flows.

3.2.2 To State

Όπως αναφέρθηκε στην ενότητα **States,** στο Corda δεν αποθηκεύεται στο ledger μόνον ένα ζεύγος τιμών όπως γίνεται στο bitcoin αλλά, αντιθέτως, αποθηκεύεται μία δομή πληροφορίας που ονομάζεται State. Στην προτεινόμενη εφαρμογή από την παρούσα διπλωματική εργασία, υλοποιείται ένα State το οποίο και εξελίσσεται μέσα από τις 3 διαφορετικές ροές εργασιών.

Το State ονομάζεται **KYC_info** και υλοποιεί το **LinearState** του API του Corda. Τα στοιχεία από τα οποία αποτελείται το State είναι :

Πίνακας 1 Τα πεδία του KYC_Info State

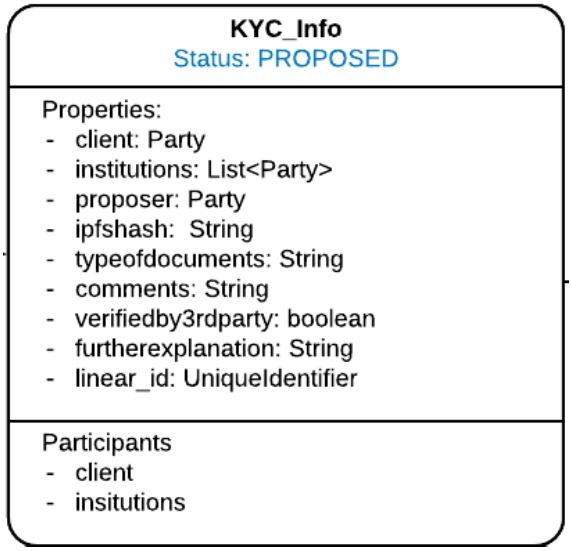
Client	ο πελάτης ο οποίος είναι αυτός που ξεκινά την διαδικασία δημιουργίας του State.
Institution	ο χρηματοπιστωτικός οργανισμός στον οποίο γίνεται η αρχική κοινοποίηση των KYC δεδομένων.
Proposer	αναφέρεται στην πλευρά εκείνη η οποία ξεκινάει την ροή εργασιών
IPFShash	το hash του IPFS που αντιστοιχεί στα δεδομένα που διαμοιράζονται στο εκάστοτε στάδιο
TypeofDocuments	Καθορίζει το είδος των εγγράφων τα οποία κοινοποιούνται από τον πελάτη προς τον οργανισμό.
Comments	περιλαμβάνει τα σχόλια τα οποία επιθυμεί ο χρήστης να συνοδεύσει την κοινοποίηση, στο οποίο μπορεί να παρέμβει και ο οργανισμός
Further Information	πεδίο στο οποίο ο πελάτης μπορεί να προσθέσει παραπάνω πληροφορίες σχετικές με τα υπό διαμοιρασμό δεδομένα
Verifiedby3rdParty	καθορίζει το κατά πόσο τα υπάρχοντα δεδομένα έχουν κριθεί έγκυρα από τον

	πρώτο οργανισμό στον οποίο κοινοποιήθηκαν
Participants	Η πιο βασική ιδιότητα του State. Καθορίζει τους χρήστες αυτούς οι οποίοι είναι σχετικοί με το συγκεκριμένο State. Αυτόματα, όπως θα φανεί στην συνέχεια, κάθε State που είναι unconsumed βρίσκεται στο vault όλων των participants. Επομένως, ορίζοντας συγκεκριμένους participants, εξασφαλίζεται ότι αυτοί θα λάβουν γνώση για την ύπαρξη του State.

Παρακάτω παρατίθεται και ο constructor (σε java) για το συγκεκριμένο State:

```
public MyState(@NotNull String status, @NotNull Party clientParty,
@NotNull Party institution, @NotNull Party proposer, @NotNull String
typeOfDocuments, String IPFShash, @NotNull String furtherexplanation,
boolean verifiedby3rdparty, String comments, @NotNull UniqueIdentifier
linear_id) {
        //this.invoiceAttachementID = invoiceAttachementID;
        //noinspection ConstantConditions
        if(clientParty == null) throw new NullPointerException("client
cannot be null");
        //noinspection ConstantConditions
        if(institution == null) throw new
NullPointerException("institution cannot be null");
        this.status = status;
        this.clientParty = clientParty;
        this.institution = institution;
        this.proposer = proposer;
        this.typeOfDocuments = typeOfDocuments;
        this.IPFShash = IPFShash;
        this.furtherexplanation = furtherexplanation;
        this.verifiedby3rdparty = verifiedby3rdparty;
        this.comments = comments;
        this.linear_id = linear_id;
    }
```

Για την αναπαράσταση της κατάστασης (State) θα κάνουμε χρήση της Corda Design Language (CDL), και πιο συγκεκριμένα της Smart Contract View:



Διάγραμμα 4 Το KYC_Info State με τα πεδία (CDL Smart Contract View)

Καταστάσεις (status) του State:

- **PROPOSED**: Η κατάσταση ***proposed*** αποτελεί την πρώτη κατάσταση (status) στην οποία μεταβαίνει το KYC_info την στιγμή που δημιουργείται. Περιέχει το IPFS hash που αντιστοιχεί στην πρώτη συλλογή πληροφοριών που παρέχει ο πελάτης (client) στον οργανισμό (institution) που γίνεται για σκοπούς επιβεβαίωσης.

- **RELEASED**: Η κατάσταση **released** αποτελεί την δεύτερη κατάσταση (status) στην οποία μεταβαίνει το KYC_Info, αφού έχει δοθεί μέσω της συλλογής υπογραφών (βλ. Συλλογή υπογραφών), η "άδεια" από τον οργανισμό να δημιουργηθεί η proposed State. ***Σημείωση***: Στην παρούσα υλοποίηση, προτιμήθηκε σχεδιαστικά, να εισαχθεί μία ενδιάμεση κατάσταση, η proposed, η οποία θα περιέχει μέρος της υπό κοινοποίηση πληροφορίας, ώστε ο πελάτης να έχει μία δεύτερη ευκαιρία να επεξεργαστεί τα υπό κοινοποίηση δεδομένα αφού λάβει έγκριση από το institution (ως έγκριση θεωρείται αυτόματα η **υπογραφή** που επιτρέπει την δημιουργία της πρώτης State).

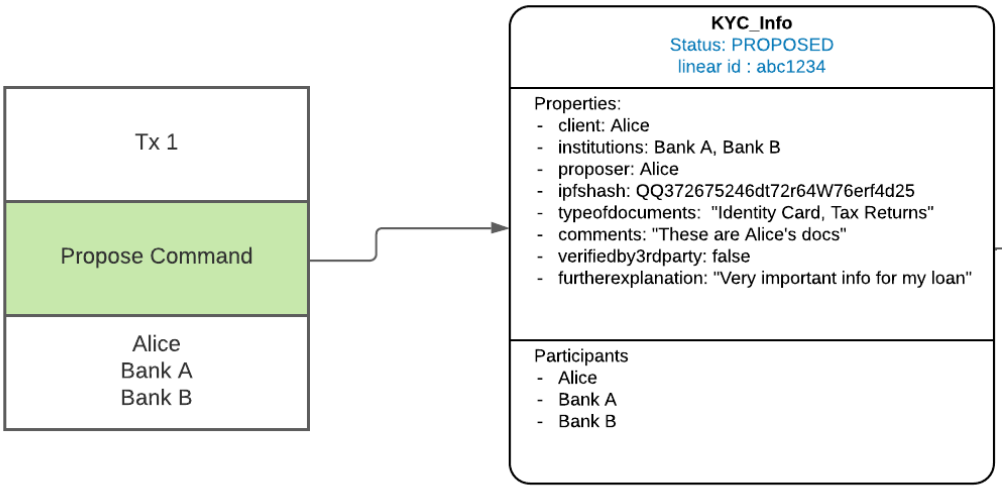
- **UPDATED**: Η κατάσταση **updated** αποτελεί την τελική κατάσταση στην οποία μπορεί να βρεθεί η κατάσταση KYC_Info. Από την στιγμή που βρίσκεται σε αυτή την κατάσταση μπορεί να μεταβεί πάλι μόνο στην ίδια κατάσταση. Επιτρέπονται πολύ συγκεκριμένες αλλαγές στα πεδία του State όταν αυτό μεταβαίνει από την κατάσταση released σε updated ή από την κατάσταση updated σε updated.

3.2.3 Το Contract

Το Contract (συμβόλαιο) που καθορίζει ποιες συναλλαγές θεωρούνται έγκυρες κατά την εκτέλεση μία εφαρμογής Cordapp αποτελεί τον πυρήνα των μεταβάσεων. Συνολικά οι ροές εργασιών (βλ. flows) της εφαρμογής είναι τρεις. Αυτές είναι η δημιουργία του KYC_Info, δηλαδή η αποθήκευση στο ledger της πληροφορίας στην πρωταρχική της μορφή (propose), η κοινοποίηση του συνόλου των δεδομένων (release), και τέλος η ενημέρωση των πληροφοριών (update). Συνολικά, απαιτείται ένα contract το οποίο όμως αποτελείται από 3 commands, ένα command για κάθε μετάβαση- ροή εργασιών.

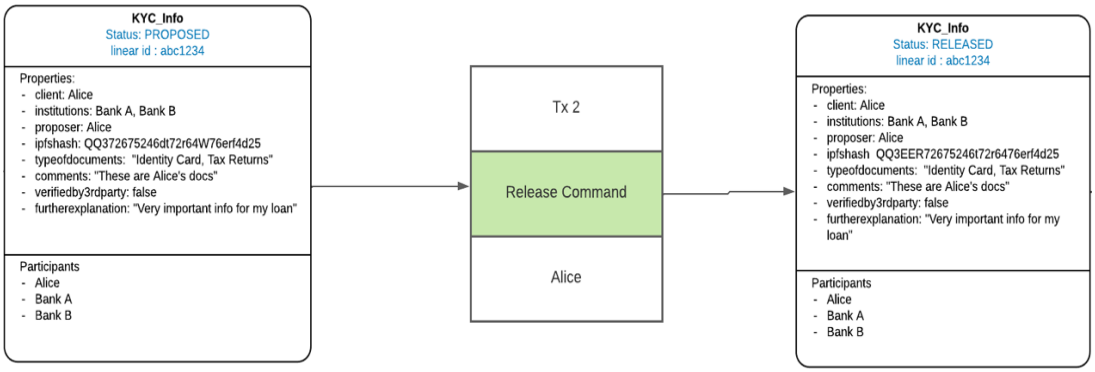
Τα **Commands** στο Corda καθορίζουν την πρόθεση που έχει μία συναλλαγή. Στην προτεινόμενη υλοποίηση υπάρχουν 3 commands που πρέπει να υλοποιηθούν και αντιπροσωπεύουν τις 3 δυνατές μεταβάσεις του State.

- **Propose**: αποτελεί στην ουσία το command που αφορά την πρώτη ροή εργασιών-συναλλαγή την οποία εκκινεί ο client και δημιουργείται το KYC_Info. Οι περιορισμοί που τίθενται στο συγκεκριμένο κομμάτι :
  - Πρέπει να υπάρχει μόνο ένα State που δημιουργείται
  - Διατήρηση του linear id
  - Οι τιμές ορισμένων πεδίων είναι αυστηρά καθορισμένες
  - Δεν καταναλώνεται κάποιο ήδη υπάρχον State αλλά δημιουργείται ένα καινούργιο
  - Ορισμένα πεδία δεν πρέπει να είναι κενά
  - Καθορίζεται ότι πρέπει να υπογράψει τόσο ο πελάτης όσο και ο αντίστοιχος οργανισμός



Διάγραμμα 5 Ledger Evolution View για το Propose Command

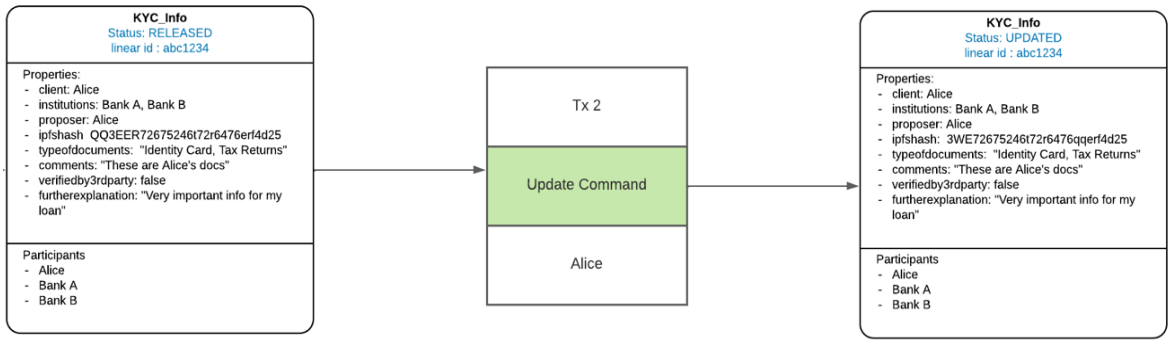
- **Release**: αποτελεί το command που καθορίζει την μετάβαση ενός State από κατάσταση "PROPOSED" σε "CONSUMED". Δηλαδή, ελέγχει την συναλλαγή που πραγματοποιείται προκειμένου ο πελάτης να κοινοποιήσει το σύνολο των δεδομένων KYC στον αντίστοιχο οργανισμό. Οι περιορισμοί που τίθενται περιλαμβάνουν τους ακόλουθους:
  - Καθορίζεται ότι μόνο ένα State μπορεί να είναι το input και το output της αντίστοιχης συναλλαγής
  - Διατήρηση του linear id
  - Οι τιμές ορισμένων πεδίων πρέπει να είναι συγκεκριμένες
  - Ορισμένα πεδία δεν πρέπει να είναι κενά
  - Όλα τα πεδία εκτός από το comments, further explanation, status, IPFS hash πρέπει να είναι ίδια. Ο συγκεκριμένος περιορισμός τίθεται προκειμένου το μόνο που να δύναται να αλλάξει να είναι το περιεχόμενο των προς κοινοποίηση πληροφοριών
  - Συγκεκριμένα, το IPFS hash επιβάλλεται να αλλάξει. Με αυτόν τον τρόπο εξασφαλίζεται ότι ο πελάτης (client) δεν παρέχει τις ίδιες πληροφορίες που παρείχε στην πρώτη εκδοχή του State, αλλά τις νέες, ενημερωμένες, πλήρεις KYC πληροφορίες
  - Καθορίζεται ότι αρκεί να υπογράψει μόνο ο πελάτης. Υπενθυμίζεται, ότι στο πρώτο στάδιο, μέσω της παροχής της υπογραφής, ο οργανισμός (institution) είχε παρέχει συναινέσει ώστε να του κοινοποιηθούν τα δεδομένα.



Διάγραμμα 6 Ledger Evolution View για το Release Command

- Update: αποτελεί το command που καθορίζει την ενημέρωση του State είτε από "**released**" σε "**updated**" είτε από "**updated**" σε "**updated**". Ελέγχει με αυτόν τον τρόπο τις συναλλαγές που αφορούν μεταβολές στην "τελική" κατάσταση κάθε State. Οι αλλαγές αυτές περιλαμβάνουν τόσο αλλαγές στα δεδομένα προς κοινοποιούνται, επομένως αλλαγές στα πεδία IPFS hash, comments, further explanation, typeofdocuments αλλά και αλλαγές στους participants, δηλαδή προσθήκες (ή αφαιρέσεις) οργανισμών που έχουν πρόσβαση στα πιο πρόσφατα δεδομένα. Οι περιορισμοί που ορίζει το contract στην συγκεκριμένη εντολή είναι:

- Καθορίζεται ότι μόνο ένα State μπορεί να είναι το input και το output της αντίστοιχης συναλλαγής
- Ορισμένα πεδία δεν πρέπει να είναι κενά
- Διατήρηση του linear id
- Οι τιμές ορισμένων πεδίων πρέπει να είναι συγκεκριμένες
- Καθορίζεται ότι αρκεί να υπογράψει μόνο ο πελάτης. Υπενθυμίζεται, ότι στο πρώτο στάδιο, μέσω της παροχής της υπογραφής, ο οργανισμός (institution) είχε παρέχει συναίνεση ώστε να του κοινοποιηθούν τα δεδομένα.



Διάγραμμα 7 Ledger Evolution View για το Update Command

3.2.4 Τα Flow

Όπως αναφέρθηκε στο κομμάτι των **flows** στο Corda, χρησιμοποιείται point to point messaging. Επομένως, πρέπει να καθορίζεται αυστηρά από τον προγραμματιστή της εφαρμογής το ποιος στέλνει μία πληροφορία, σε ποιόν την στέλνει καθώς και με ποια σειρά. Στην υλοποίηση που προτείνεται στην συγκεκριμένη εργασία, έχουν οριστεί συνολικά 3 flows. Αυτά αποτελούν την ραχοκοκαλιά της εφαρμογής, και καθορίζουν την αλληλουχία των γεγονότων. Η αλληλουχία των γεγονότων που συμβαίνουν κατά την διάρκεια κάθε ροής απεικονίζεται μέσω BPMN διαγραμμάτων.

3.2.4.1 First Flow

Το πρώτο flow, ή **First Flow** όπως ονομάστηκε, έχει σκοπό να πραγματοποιήσει μία συναλλαγή (transaction) προκειμένου να εκδοθεί το State KYC_Info. Το party που εκκινεί το συγκεκριμένο flow είναι ο **client (proposer).**

Αρχικά, ο client υλοποιεί έναν **TransactionBuilder** προκειμένου να ξεκινήσει την διαδικασία δημιουργίας της συναλλαγής. Προφανώς, εφόσον πραγματοποιείται έκδοση νέου State, χωρίς να τίθεται ως **CONSUMED** κάποιο προηγούμενο State, δεν υπάρχει input State, πάρα μόνον output State. Κατά την διαδικασία αυτή δημιουργείται το νέο State σε status "**PROPOSED**" που θα περιέχει μέρος των πληροφοριών που επιθυμεί να κοινοποιήσει ο πελάτης (client). Ο πελάτης δημιουργεί το KYC_Info State εισάγοντας τις πληροφορίες που επιθυμεί στα πεδία του State. Ειδικότερα στο πεδίο του **IPFShash,** εισάγει το hash που αναφέρεται στο σύνολο εκείνο των δεδομένων που αποσκοπούν απλώς στην αρχική επιβεβαίωση του πελάτη, όχι στο σύνολο τους. Η πρώτη ροή εργασιών, όπως αναφέρθηκε και παραπάνω, έχει ως βασικό σκοπό την επιβεβαίωση ότι και οι δύο πλευρές (πελάτης και οργανισμός) επιθυμούν από κοινού την κοινοποίηση των δεδομένων.
Στην συνέχεια, προστίθεται το απαραίτητο **Command** (Propose Command) το οποίο και αντανακλά το σημείο του Contract το οποίο καθορίζει τι επιτρέπεται κατά την διάρκεια αυτής της συναλλαγής.

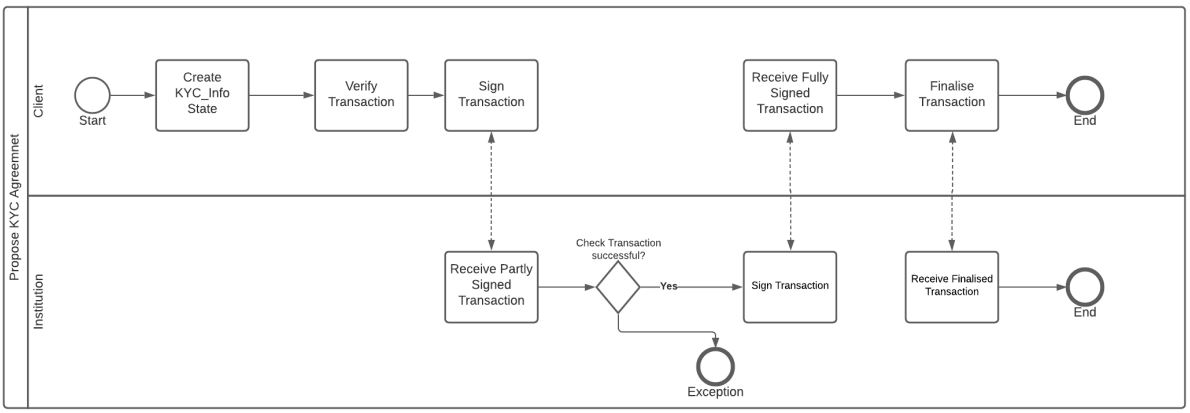
```
final Command<Propose> txCommand = new Command<>(new
Propose(),requiredSigners);

            final TransactionBuilder txbuilder = new
TransactionBuilder(notary)
                    .addCommand(txCommand)
                    .addOutputState(KYCInfo, MyContract.MY_CONTRACT_ID);

            txbuilder.verify(getServiceHub());

            final SignedTransaction partlySignedtx =
getServiceHub().signInitialTransaction(txbuilder);
```

Μετά την ολοκλήρωση της δημιουργίας της συναλλαγής (transaction), ελέγχει αν είναι valid, γίνεται, δηλαδή, το verification. Αξίζει να σημειωθεί ότι το verification γίνεται κατά την διάρκεια της δημιουργίας της συναλλαγής **(transaction builder).** Το transaction builder δεν αποθηκεύει καμία υπογραφή. Αντιθέτως, η υπογραφή του transaction γίνεται αφού έχει δημιουργηθεί, πλήρως, το transaction. **Το contract, στο Corda, επιβεβαιώνει την παρουσία των απαιτούμενων Parties που πρέπει να υπογράψουν (required signers), και όχι την παρουσία των υπογραφών αυτών καθαυτών.** Επομένως, το verification, από το Contract, γίνεται ΠΡΙΝ υπογραφεί η συναλλαγή.

Τέλος, όπως καθορίζεται και από το **Issue command**, απαιτείται και η συλλογή της υπογραφής της άλλης πλευράς η οποία συμμετέχει στην συναλλαγή, αυτή του Institution. Δίνεται το σήμα για εκκίνηση ενός **subFlow**, το οποίο έχει ως στόχο την συλλογή της υπογραφής από την πλευρά του Institution. Σε περίπτωση που ο οργανισμός δεν γνωρίζει τον πελάτη (client) ή δεν επιθυμεί για οποιοδήποτε άλλο λόγο να λάβει τα δεδομένα του, δεν

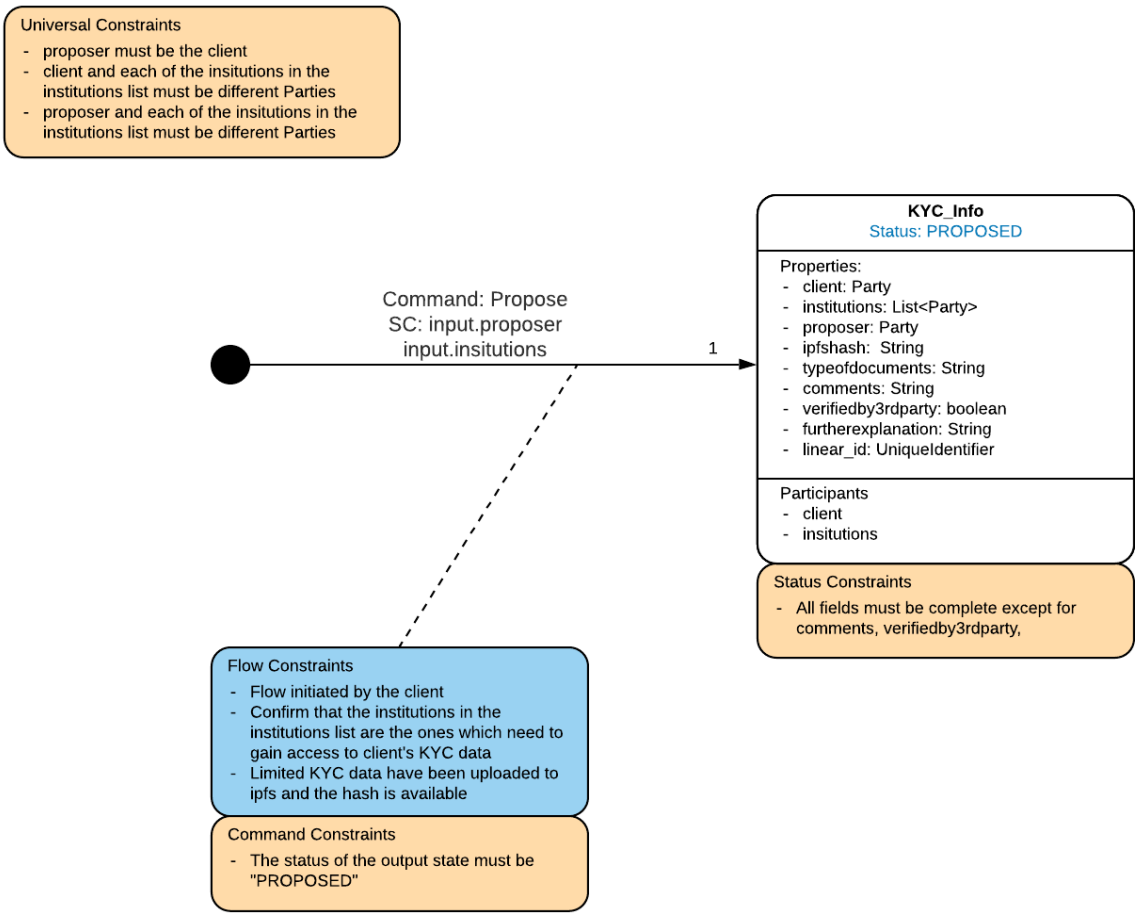
αποδέχεται την συναλλαγή, **μη υπογράφοντάς την**. Με αυτόν τον τρόπο, δεν θα δημιουργηθεί το KYC_Info State. Ο παραπάνω έλεγχος γίνεται μέσω του checkTransaction όπως φαίνεται παρακάτω. Στην συγκεκριμένη περίπτωση ελέγχεται μόνον εάν πράγματι στο πεδίο Institution βρίσκεται το όνομα του σωστού οργανισμού. Σε διαφορετικές εκδοχές θα μπορούσαν να γίνονται πληθώρα ελέγχων σχετικά με τον πελάτη.

```
@Override
                protected void checkTransaction(@NotNull
SignedTransaction stx) throws FlowException {
                    // I must be relevant
                    requireThat(require -> {
                        // Client is a member of Institution's client
                        // We are part of the Institution list to give
out KYCInfo
                        KYC_info KYCInfo = (KYC_info)
stx.getTx().getOutputs().get(0).getData();
                        require.using("Must be a transaction between
Institution and the Client",

KYCInfo.getInstitution().equals(getOurIdentity()));
                        return null;
                    });
                }
```



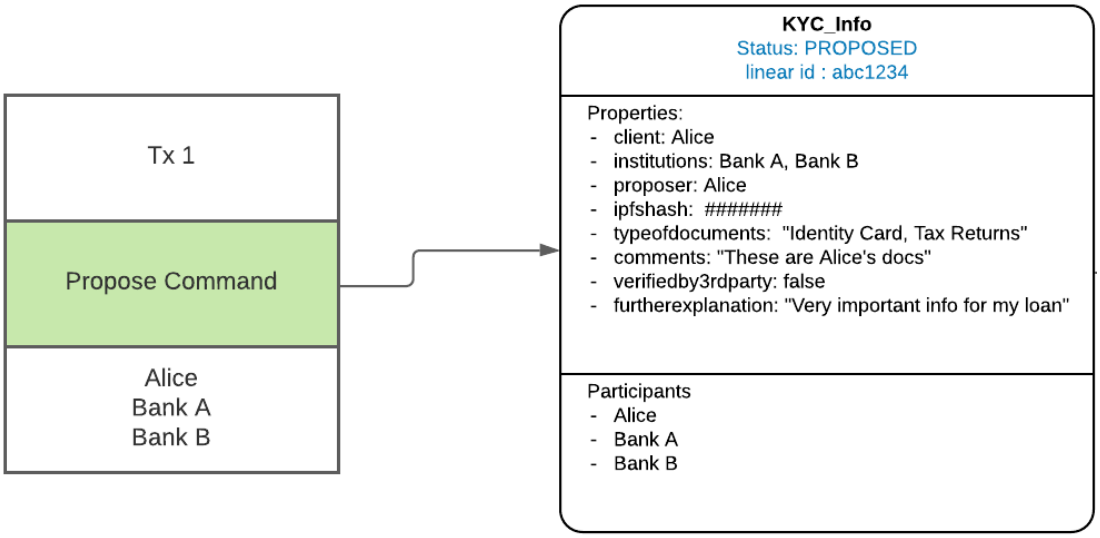
Διάγραμμα 8 Το BPMN διάγραμμα για το First Flow

Ακολούθως φαίνεται μέρος του Corda Design Language Smart Contract View. Το συγκεκριμένο τμήμα απεικονίζει την αρχή του Cordapp, δηλαδή το First Flow. Φαίνονται όλοι οι περιορισμοί που διέπουν την δημιουργία του αρχικού State.



Διάγραμμα 9 Το διάγραμμα Smart Contract View του CDL που δείχνει την δημιουργία του KYC_Info State μετά την εκτέλεση του First Flow

Παρακάτω είναι εμφανής η εξέλιξη της κατάστασης στο Ledger του Corda, μέσω της απεικόνισης του Cordapp Ledger Evolution View. Για το συγκεκριμένο παράδειγμα βλέπουμε και την απεικόνισή του μέσω Corda Node Explorer στην συνέχεια. Το Corda Node Explorer μας δίνει την δυνατότητα να εκκινούμε flows αλλά και να έχουμε μία γραφική άποψη του Vault και των Transactions κάθε κόμβου (node).



Διάγραμμα 10 Ledger Evolution View για το Propose Command που ορίζει την συναλλαγή (transaction) του First Flow

3.2.4.2 Second Flow

Το δεύτερο flow, ή **Second Flow (Release Flow)** όπως ονομάστηκε, έχει σκοπό να πραγματοποιήσει μία συναλλαγή (transaction) προκειμένου να αλλάξει ορισμένα πεδία του State KYC_Info. Το party που εκκινεί το συγκεκριμένο flow είναι και πάλι ο **client (proposer).** Συγκεκριμένα, ο πελάτης (client) εκκινεί μία ροή εργασιών (flow) προκειμένου να θέσει το ήδη υπάρχον στο blockchain State KYC_Info ως CONSUMED, και να εκδώσει ένα **νέο State** το οποίο θα περιέχει το **IPFS hash** που ανταποκρίνεται στο **σύνολο** των KYC δεδομένων προς κοινοποίηση. Το συγκεκριμένο flow έχει ως αποτέλεσμα την κοινοποίηση των πλήρων δεδομένων στους οργανισμούς που επιλέγει ο πελάτης.

Αρχικά είναι απαραίτητο να «βρεθεί» το State εκείνο που πρέπει να καταναλωθεί και να δηλωθεί ως CONSUMED και να δημιουργηθεί ένα άλλο στην θέση του. Ο κόμβος client πραγματοποιεί ένα Vault Query, δηλαδή μία αναζήτηση στο Vault του, προκειμένου να βρεθεί το State αυτό. Η αναζήτηση μπορεί να γίνει με διάφορα κριτήρια. Στην συγκεκριμένη υλοποίηση γίνεται αναζήτηση με κριτήριο το linearKYCInfo, δηλαδή το id hash του State. Εναλλακτικά, για μεγαλύτερη ευκολία, θα μπορούσε να πραγματοποιηθεί ένα Vault Query με κριτήριο τους participants του State. Το Vault Query φαίνεται παρακάτω:

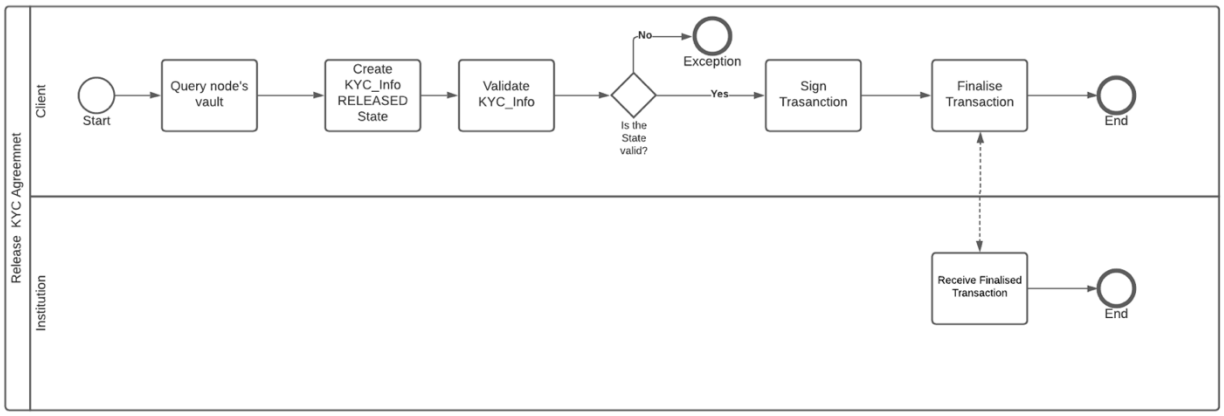
```
QueryCriteria criteriaforProposedKYC = new
QueryCriteria.LinearStateQueryCriteria(
                    null,
                    ImmutableList.of(linearIdKYCInfo),
                    Vault.StateStatus.UNCONSUMED,
                    null
            );

            List<StateAndRef<KYC_info>> inputStateList =
getServiceHub().getVaultService().queryBy(KYC_info.class,
                    criteriaforProposedKYC).getStates();
            if (inputStateList.isEmpty()) {
                // may add extra null
                throw new IllegalArgumentException("State cannot be found"
+ inputStateList.size() + " " + linearIdKYCInfo);
            }
```

Ο πελάτης (client) υλοποιεί ένα **TransactionBuilder** προκειμένου να ξεκινήσει την διαδικασία δημιουργίας της συναλλαγής. Κατά την διαδικασία αυτή, δημιουργείται το νέο State, που θα αποτελέσει το output του νέου Transaction. Ακόμη, επιλέγεται το Notary το οποίο είναι υπεύθυνο για την αυθεντικοποίηση της συναλλαγής. Εδώ σημειώνεται ότι το Notary πρέπει να είναι το ίδιο με το First Flow. Το Corda επιβάλλει ότι ένα State πρέπει να αυθεντικοποιηθεί από το ίδιο Notary Service καθ' όλη την διάρκεια εξέλιξής του.

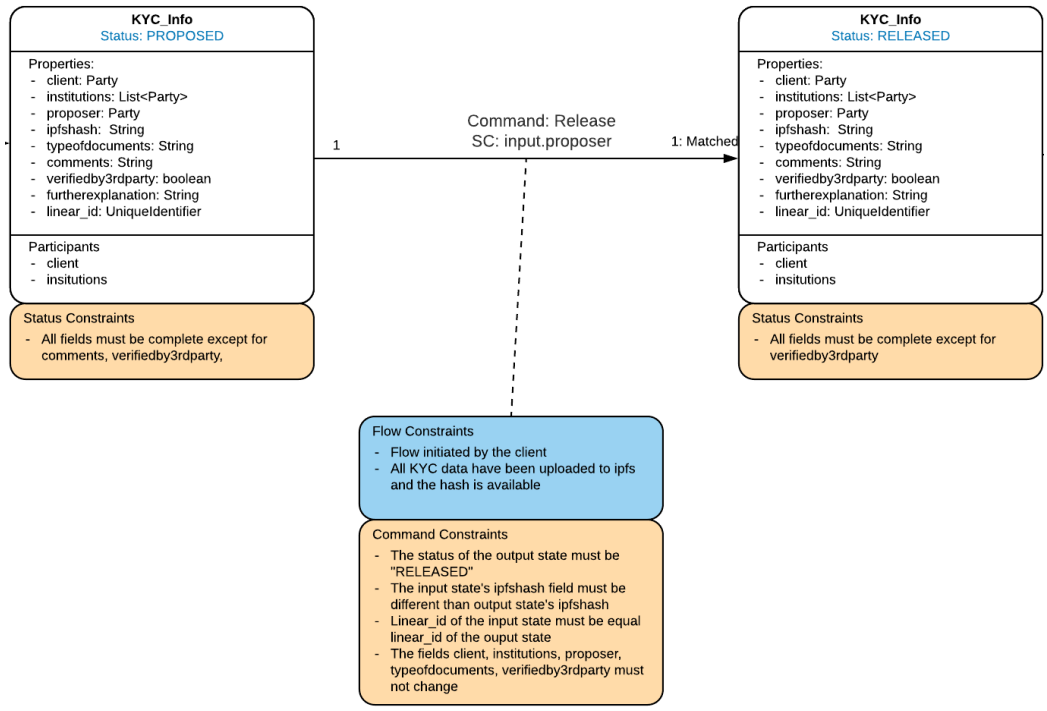
Το status του νέου State θα είναι **"RELEASED"** καθώς έχουν κοινοποιηθεί πλέον όλα τα δεδομένα. Στην συνέχεια, προστίθεται το απαραίτητο Command (Release Command) το οποίο αντανακλά το σημείο του Contract το οποίο και καθορίζει τι επιτρέπεται κατά την διάρκεια της συγκεκριμένης συναλλαγής (βλ. Commands). Το flow περιέχει, επίσης, την διαδικασία του verification.

Στην συγκεκριμένη υλοποίηση, όπως αναφέρθηκε και νωρίτερα, δεν απαιτείται συλλογή υπογραφής από το Institution, καθώς αυτό το Party αποδέχθηκε την έκδοση του State κατά την διάρκεια του FirstFlow. Ξεκινάει μια sub-flow με μοναδικό σκοπό την ενημέρωση των participants (institutions).

Ακολούθως φαίνεται μέρος του Corda Design Language Smart Contract View. Το συγκεκριμένο τμήμα απεικονίζει την πορεία του Second Flow, που περιλαμβάνει την κοινοποίηση (Release) των KYC δεδομένων από τον πελάτη προς τον κάθε οργανισμό (institution).



Διάγραμμα 11 Το BPMN διάγραμμα για το First Flow



Διάγραμμα 12 Το CDL Smart Contract View που απεικονίζει το Second Flow

3.2.4.3 Third Flow

Το τρίτο flow, ή Third Flow όπως ονομάστηκε, έχει σκοπό να πραγματοποιήσει την ενημέρωση (update) του State που περιέχει το σύνολο των KYC δεδομένων. Συγκεκριμένα λαμβάνει ως input ένα State που έχει κοινοποιηθεί σε ένα ή περισσότερα institutions, δηλαδή

έχει ως State “RELEASED” ή “UPDATED” και την συνέχεια το θέτει ως CONSUMED, παράγοντας ένα νέο State το οποίο έχει ως status “UPDATED”.

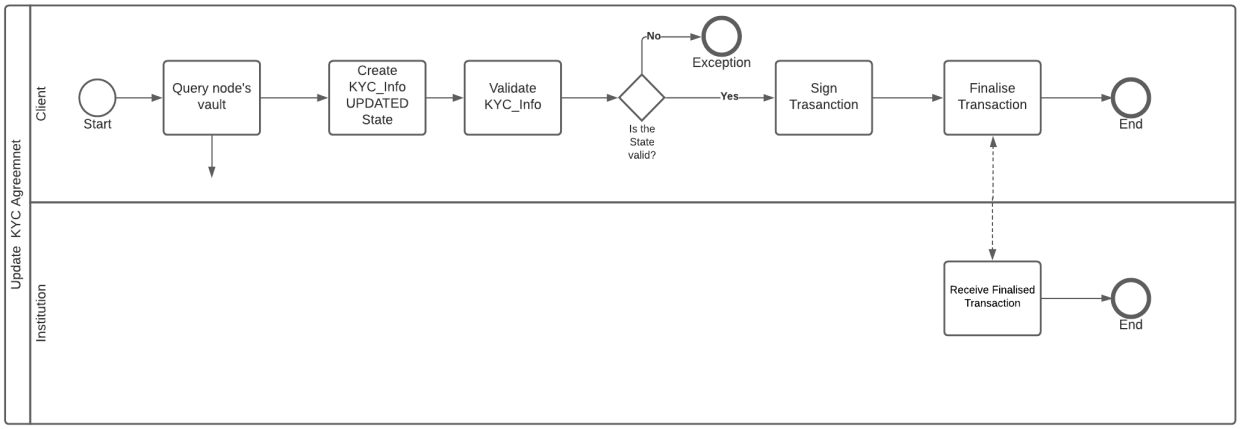
Σκοπός της συγκεκριμένης ροής εργασιών είναι να δίνεται η δυνατότητα στον πελάτη να επεξεργάζεται το σύνολο των δεδομένων που κοινοποιεί. Παράλληλα, η αλλαγή αυτή γίνεται ορατή στο σύνολο των συμμετεχόντων του συγκεκριμένου State σε πολύ σύντομο χρονικό διάστημα. Επομένως, επιτυγχάνεται η **ανανέωση των KYC δεδομένων** που αφορούν έναν συγκεκριμένο πελάτη ταυτόχρονα σε όλους τους institutions στους οποίους έχουν κοινοποιηθεί αυτά. Παράλληλα, η τεχνολογία του Corda εξασφαλίζει ότι πάντοτε στα Vaults τους, όλοι οι συμμετέχοντες (participants) θα έχουν την πιο ενημερωμένη έκδοση του State.
Η δομή της 3ης ροής εργασιών (Third Flow – Update Flow) είναι παρόμοια με εκείνη της Second Flow. Η κύρια διαφορά έγκειται στο γεγονός ότι κατά την μεταβολή του KYC_Info επιτρέπονται παραπάνω τροποποιήσεις.

Πιο αναλυτικά, ο πελάτης (client) είναι ο Initiator, δηλαδή αυτός που εκκινεί την ροή εργασιών. Αρχικά εκτελεί ένα Vault Query προκειμένου να βρει το State το οποίο πρόκειται να αποτελέσει το input State της συναλλαγής (transaction), όπως φαίνεται παρακάτω:

```
QueryCriteria criteriaforReleasedKYC = new QueryCriteria.LinearStateQueryCriteria(
        participants: null,
        ImmutableList.of(linearIdKYCInfo),
        Vault.StateStatus.UNCONSUMED,
        contractStateTypes: null);

List<StateAndRef<MyState>> inputStateList = getServiceHub().getVaultService().queryBy(MyState.class,
        criteriaforReleasedKYC).getStates();
if (inputStateList.isEmpty()) {
    // maybe an extra check for null
    throw new IllegalArgumentException("State cannot be found" + inputStateList.size() + " " + linearIdKYCInfo);
}
```

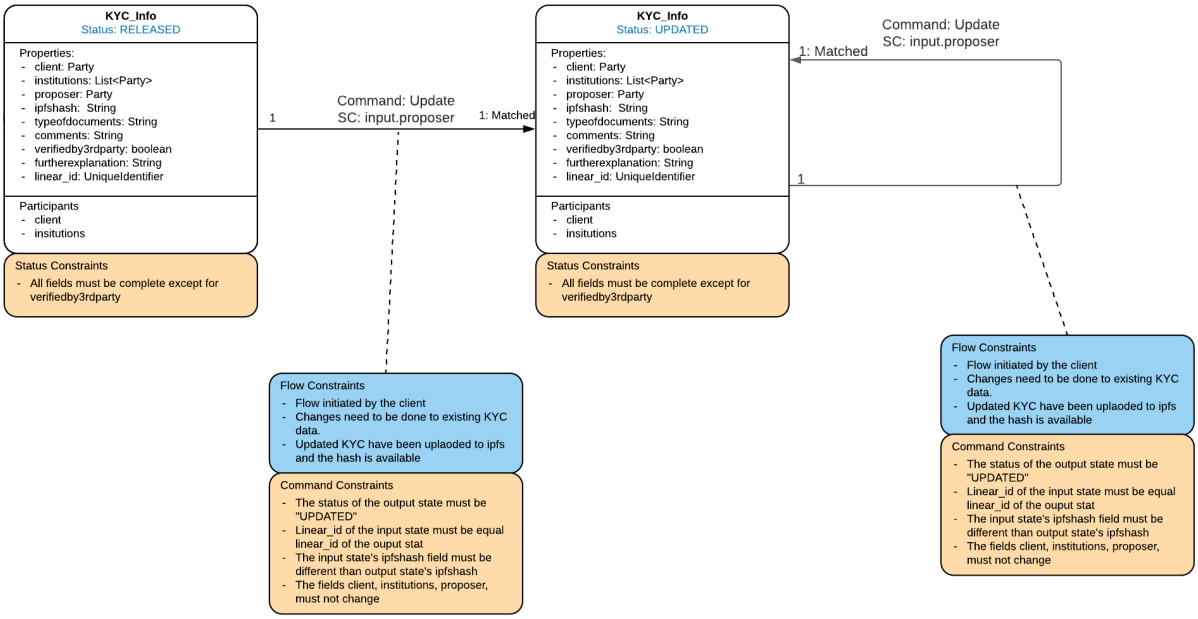
Αφού αποκτηθεί το State Id - που όπως έχει αναλυθεί και πολλές φορές προηγουμένως αποτελείται από το id της συναλλαγής στην οποία ήταν output καθώς και ένα offset αριθμό - δημιουργείται το TransactionBuilder. Στην συνέχεια προστίθεται το παραπάνω input State και το output State του οποίου τα πεδία περιέχουν τις καινούργιες-ενημερωμένες πληροφορίες ενώ τέλος προστίθεται και το Command. Στην συγκεκριμένη περίπτωση το Command είναι το Update. Αφού γίνει το verification και εξασφαλιστεί ότι η μετάβαση υπακούει στα προστάγματα του Contract (Update Command), ο client υπογράφει την συναλλαγή. Η συγκεκριμένη υλοποίηση δεν απαιτεί συλλογή υπογραφής από την άλλη πλευρά, δηλαδή αυτή των institutions. Θεωρείται δεδομένη η συναίνεση τους από την στιγμή που είχαν υπογράψει το First Flow transaction.
Ειδικό ενδιαφέρον στο συγκεκριμένο workflow έχουν οι περιορισμοί τους οποίους επιβάλει το Contract. Συγκεκριμένα αν ανατρέξουμε στον κώδικα του Contract, θα παρατηρήσουμε ότι καθορίζεται η υποχρεωτική αλλαγή της τιμής του IPFShash μεταξύ άλλων. Ο παραπάνω περιορισμός, όπως αναφέρθηκε και παραπάνω, εξασφαλίζει ότι δεν θα γίνεται άσκοπη εκκίνηση της συγκεκριμένης workflow. Θυμίζουμε ότι το IPFS έχει την ιδιότητα να παρέχει

διαφορετικό hash για κάθε διαφορετική αποθήκευση ή τροποποίηση αρχείων. Επομένως η απαίτηση για διαφορετικό IPFShash εξασφαλίζει ότι πράγματι τα προς κοινοποίηση δεδομένα έχουν αλλάξει.
Τέλος, η ροή εκκινεί μία sub-flow με σκοπό την ενημέρωση του(των) institution(s) για τις αλλαγές, δίχως να ζητείται υπογραφή.



Διάγραμμα 13 Το BPMN διάγραμμα για το Second Flow

Παρακάτω απεικονίζεται το κομμάτι του CDL Smart Contract View διαγράμματος το οποίο αναφέρεται στην διαδικασία ενημέρωσης του State.



Διάγραμμα 14 CDL Smart Contract View for Update Flow

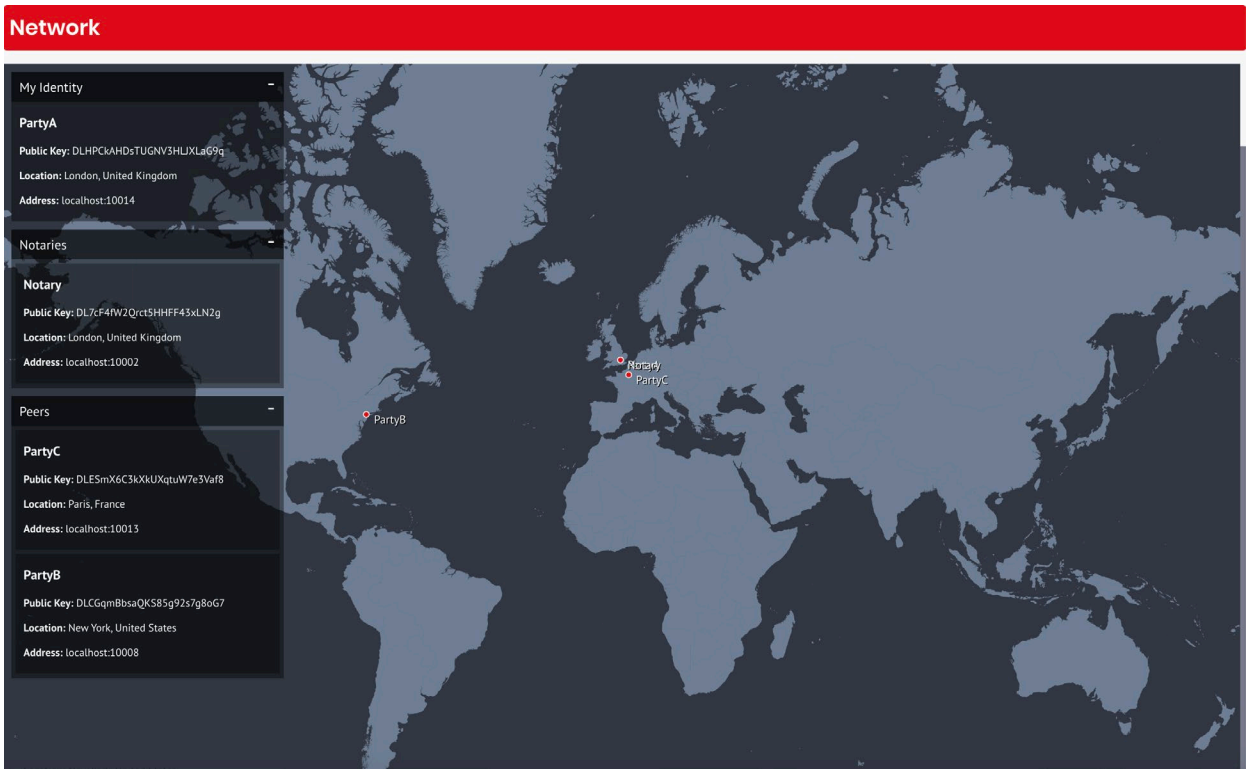
3.2.5 Λειτουργία της εφαρμογής

Στην συνέχεια θα δούμε την εκτέλεση της εφαρμογής KYC CordApp σε πραγματικές συνθήκες λειτουργίας. Για την καλύτερη κατανόηση θα γίνει χρήση της εφαρμογής Corda Node Explorer (GUI).

Ο πελάτης στην συγκεκριμένη περίπτωση (client), ονομάζεται PartyA ενώ ο οργανισμός ονομάζεται PartyB. Ο πελάτης-PartyA επιθυμεί να κοινοποιήσει τα δεδομένα KYC στον οργανισμό. Αρχικά, ανεβάζει ένα πρώτο υποσύνολο των δεδομένων του στο IPFS. Αφού λάβει το IPFS hash, ξεκινάει την First Flow.

Εικόνα 8 Χρήση IPFS για uploading μέρους των KYC δεδομένων

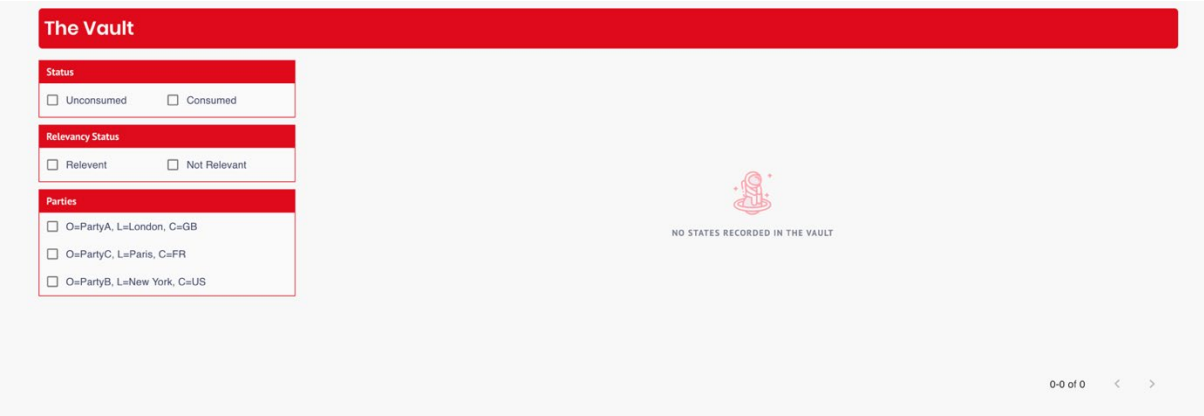
Το δίκτυο των Nodes στο Corda φαίνεται ακολούθως:



Εικόνα 9 Το δίκτυο Corda όπως φαίνεται στο Corda Node Explorer

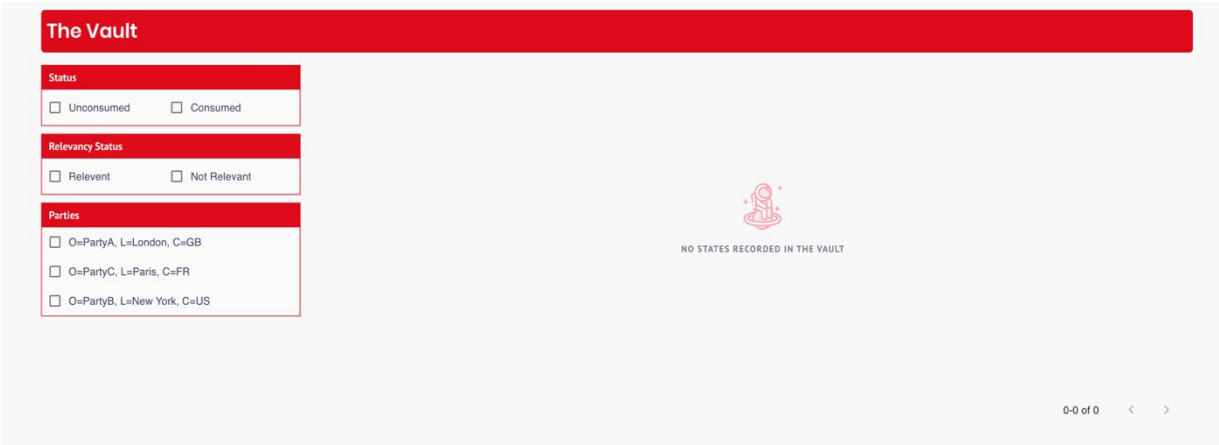
Χρησιμοποιείται το Corda Node Explorer [37] προκειμένου να αποκτηθεί πρόσβαση στις πληροφορίες που είναι αποθηκευμένες σε κάθε κόμβο του δικτύου.

Όπως φαίνεται και στην παρακάτω εικόνα, τόσο το Vault του client (πάνω), όσο και το Vault του institution (κάτω) δεν περιέχουν κάποιο State.



Εικόνα 10 Το Vault του πελάτη όπως φαίνεται στο Corda Node Explorer

Ο πελάτης (client) εκκινεί μία ροή εργασιών **First Flow - Propose** προκειμένου να ξεκινήσει η διαδικασία κοινοποίησης των KYC δεδομένων. Η δημιουργία του KYC_ Info γίνεται με τα ακόλουθα ορίσματα:



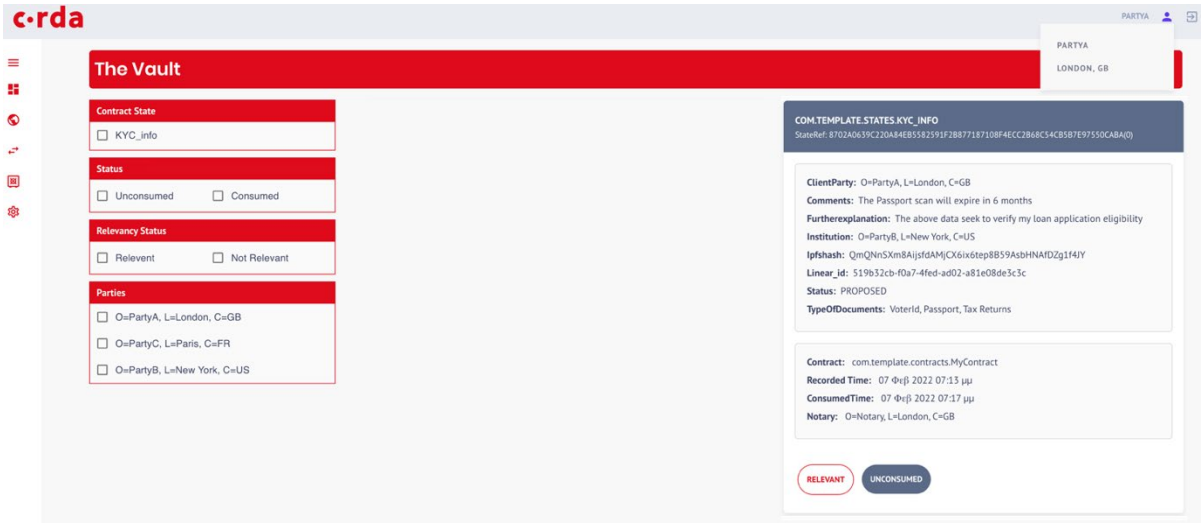
Εικόνα 11 Το Vault του institution όπως φαίνεται στο Corda Node Explorer

Πίνακας 2 Τα πεδία της δοκιμαστικής λειτουργίας

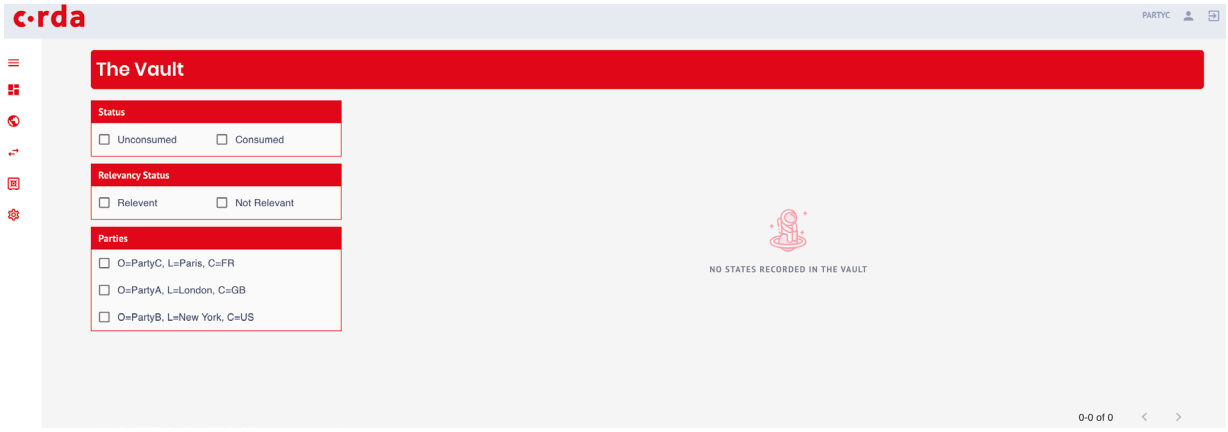
Client	PartyA
Institution	PartyB
TypeofDocuments	VoterId, Passport, Tax Returns

IPFShash	QmQNnSXm8AijsfdAMjCX6ix6tep8B59AsbHNAfDZg1f4JY
FurtherInformation	The above data seek to verify my loan application eligibility
Comments	The Passport scan will expire in 6 months
IsVerified	false

Στην συγκεκριμένη περίπτωση γίνεται αποδεκτή η συναλλαγή από το PartyB (institution) καθώς ικανοποιούνται οι συνθήκες του checkTransaction αλλά και του Contract. Στην ακόλουθη εικόνα φαίνονται τα Vaults των PartyA, PartyB μετά την επιτυχή πραγματοποίηση της συναλλαγής. Ιδιαίτερο ενδιαφέρον έχει το Vault του PartyC. Από την στιγμή που ως participants του State έχουν οριστεί ο πελάτης (client) και ο οργανισμός (institution), ο κόμβος PartyC έχει πλήρη άγνοια για τα δεδομένα που περιλαμβάνονται στο KYC_Info State. Αυτή ακριβώς την ιδιότητα, δηλαδή η απουσία gossip protocol, εκμεταλλευόμαστε στην παρούσα εφαρμογή ώστε να εξασφαλιστεί η ιδιωτικότητα των KYC δεδομένων.



Εικόνα 12 Το Vault του πελάτη μετά το First Flow όπως φαίνεται στο Corda Node Explorer



Εικόνα 13 Το Vault του Party C μετά το First Flow όπως φαίνεται στο Corda Node Explorer. Δεν έχει καταγραφεί κανένα State, ως αποτέλεσμα της μη χρήσης gossip protocol απο το Corda

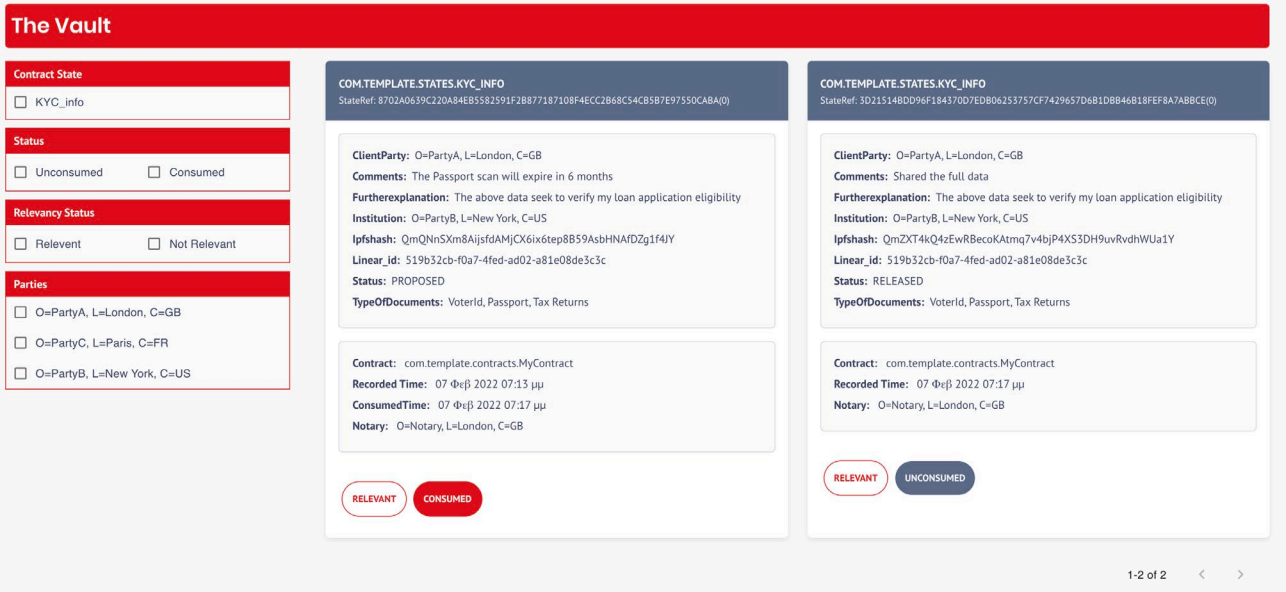
Αφού έχει δημιουργηθεί το KYC_Info State που περιλαμβάνει τα περιορισμένα δεδομένα KYC για λόγους επιβεβαίωσης, ο πελάτης (client) προχωρά στην αποθήκευση του πλήρους συνόλου δεδομένων στο IPFS, και λαμβάνει το IPFS hash.

```
added QmZXT4kQ4zEwRBecoKAtmq7v4bjP4XS3DH9uvRvdhWUa1Y KYC_data_full.pdf
 325.82 KiB / 325.82 KiB [=============================================================================================================]
foivos@192 Downloads %
```

Εικόνα 14 Χρήση IPFS για uploading του συνόλου των KYC δεδομένων

Στην συνέχεια, ο πελάτης (client) εκκινεί την ροή εργασιών **Second Flow** προκειμένου να κάνει Release το σύνολο των KYC δεδομένων.

Όπως φαίνεται και παρακάτω, πλέον το πρώτο State που είχε ως Status “PROPOSED” είναι πλέον CONSUMED και έχει δημιουργηθεί ένα νέο State με Status “RELEASED”. Την συγκεκριμένη αλλαγή έχουν καταγράψει στο Vault τους τόσο ο πελάτης (client) όσο και ο οργανισμός (Institution). Το PartyC εξακολουθεί να έχει πλήρη άγνοια σχετικά με τα δεδομένα που κοινοποιήθηκαν.



Εικόνα 15 Το Vault του πελάτη μετά και το Second Flow. Παρατηρούμε ένα CONSUMED State σε κατάσταση "PROPOSED" και ένα UNCONSUMED State σε κατάσταση "RELEASED"

3.3 Αρχιτεκτονική εφαρμογής με χρήση Corda attachments

3.3.1 Περιγραφή της δομής και λειτουργίας της εφαρμογής

Στην συγκεκριμένη προτεινόμενη υλοποίηση, η βασική διαφορά με την αρχική υλοποίηση (με χρήση IPFS) έγκειται στο γεγονός ότι το σύνολο των προς κοινοποίηση KYC δεδομένων δεν αποθηκεύονται εκτός Corda, αλλά γίνεται χρήση των attachments που παρέχεται από το ίδιο το Corda. Παραπάνω έγινε εκτενής αναφορά στα Attachments, την χρήση τους και τις ιδιότητές τους.

Ο λόγος που επιλέγεται να γίνει χρήση των Attachments για την κοινοποίηση των KYC δεδομένων, είναι η ασφάλεια και η εμπιστευτικότητα του συγκεκριμένου τρόπου αποθήκευσης. Η βασική σχεδιαστική χρήση των attachments στο Corda ήταν η αποθήκευση κώδικα των Contracts ή η αποθήκευση δεδομένων που προσπελλάσονται συχνά από τους συμμετέχοντες. Δεδομένου, όμως, ότι η αποθήκευση και κοινοποίηση των αρχείων ως attachments πληροί το σύνολο των προϋποθέσεων που έχουν τεθεί για την εφαρμογή που προτείνεται, καθιστά την χρήση τους κατάλληλη.

Η αρχιτεκτονική της συγκεκριμένης υλοποίησης είναι αρκετά παρόμοια με την πρώτη περίπτωση όπου τα δεδομένα αποθηκεύονται από τον χρήστη εξωτερικά του δικτύου Corda (στο IPFS). Η βασική αλλαγή έγκειται στο γεγονός ότι τα δεδομένα πλέον θα ανταλλάσσονται μέσω της πλατφόρμας Corda, ως **attachments** στις αντίστοιχες συναλλαγές που δημιουργούνται.

Βασικοί συμμετέχοντες του δικτύου παραμένουν ο πελάτης (client) που επιθυμεί να διαμοιράσει τα δεδομένα του, και ο οργανισμός (institution) που επιθυμεί να λάβει τα δεδομένα αυτά, να τα επεξεργαστεί υλοποιώντας έλεγχο KYC και να αυθεντικοποιήσει ή όχι τον πελάτη. Και στην συγκεκριμένη περίπτωση, η εφαρμογή βασίζεται στην ιδιότητα του δικτύου Corda να μην χρησιμοποιεί gossip protocol αλλά, αντιθέτως, να αποστέλλει μόνο τις απαραίτητες πληροφορίες σε όποιους εκ των συμμετεχόντων του δικτύου επιθυμεί. Παρακάτω ακολουθούν τα βήματα που αποτελούν την εφαρμογή:

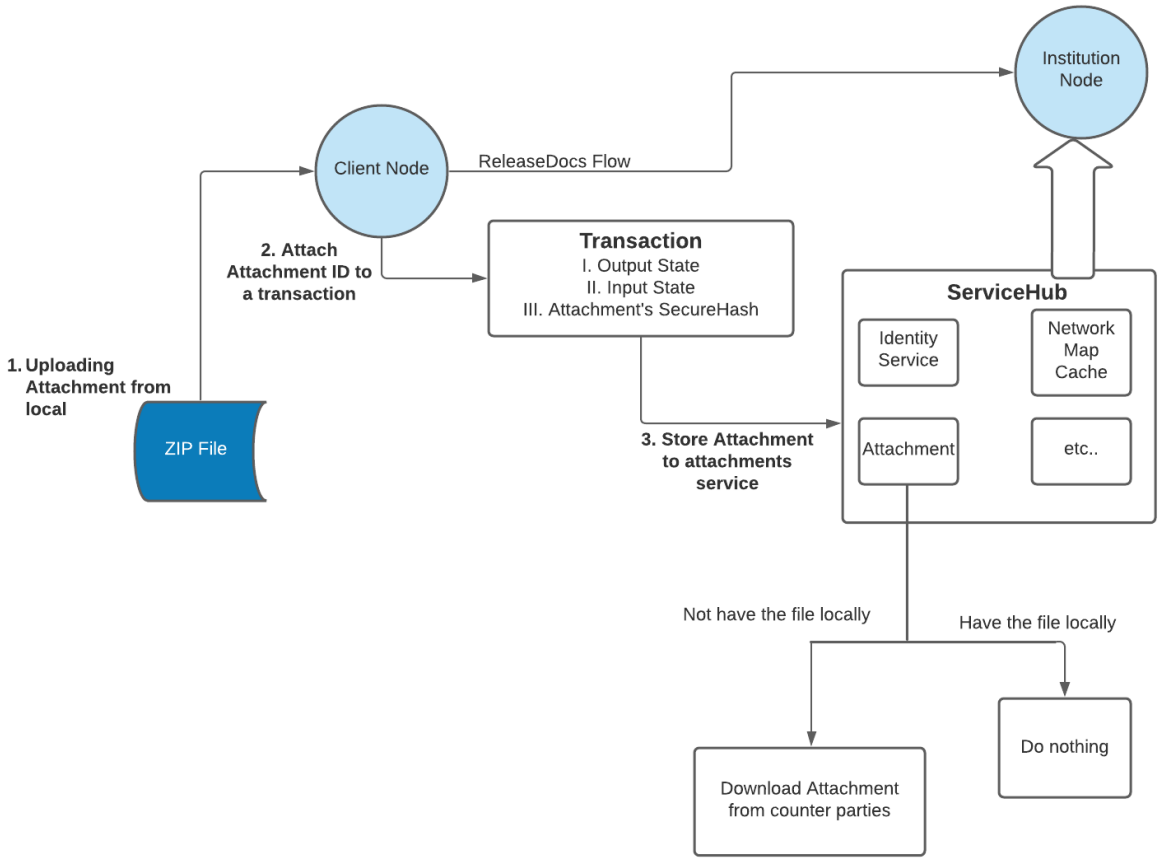
1ο βήμα: Ο πελάτης, μέσω της αντίστοιχης workflow που θα αναλυθεί στην συνέχεια, δημιουργεί ένα State το οποίο αποτελεί την αποτύπωση των δεδομένων KYC στο δίκτυο blockchain, ακριβώς ανάλογα με την πρώτη περίπτωση (αποθήκευσης μέσω IPFS). Συμμετέχοντες (participants) του συγκεκριμένου State είναι ο πελάτης (client) ο οποίος και το δημιούργησε καθώς και ο ένας ή περισσότεροι χρηματοπιστωτικοί οργανισμοί (institutions) στους οποίους επιθυμεί ο πελάτης να κοινοποιηθούν τα δεδομένα. Η δημιουργία τους State διέπεται από μία σειρά κανόνων που υπάρχουν στο πρώτο contract. Αφού δημιουργηθεί το State, αποστέλλεται μέσω subflows στους αντίστοιχους οργανισμούς. Σκοπός της διαδικασίας αυτής είναι να γίνει η αναγνώριση του πελάτη από κάθε οργανισμό.

2ο βήμα: Προκειμένου να προχωρήσει η διαδικασία, όλοι οι οργανισμοί οφείλουν να κάνουν δεκτή την έκδοση του State που δεν περιέχει ακόμα κάποιο attachment, παρά μόνον

αναγνωριστικές πληροφορίες σχετικά με τα προς κοινοποίηση δεδομένα. Σε περίπτωση που το σύνολο των οργανισμών συμφωνήσουν (δηλαδή υπογράψουν βλ. First Flow), ο πελάτης (client) έχει την δυνατότητα να ενημερώσει το State που έχει εκδώσει, προσθέτοντας στην συναλλαγή το 1 ή περισσότερα attachments που θα κοινοποιηθεί στους οργανισμούς. Στο σημείο αυτό γίνεται και η **κοινοποίηση των KYC δεδομένων.**

3ο βήμα: Στην συνέχεια, ο πελάτης έχει την δυνατότητα να προσθέσει έναν ή παραπάνω συμμετέχοντες (participants) στο State, δίνοντας τους πρόσβαση στα δεδομένα του. Αυτοί, εάν κάνουν την συναλλαγή, αποκτούν πρόσβαση στα attachments- δεδομένα.

Η βασική αρχιτεκτονική της δεύτερης (Second-Release) ροής εργασιών απεικονίζεται και παρακάτω:



Διάγραμμα 15 Η βασική αρχιτεκτονική της υλοποίησης με χρήση Corda Attachments

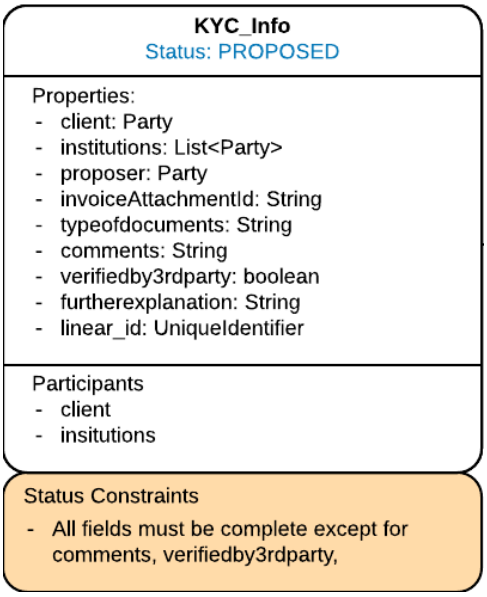
3.3.2 To State

Το State της συγκεκριμένης αρχιτεκτονικής έχει τα περισσότερα κοινά χαρακτηριστικά με αυτό της πρώτης προτεινόμενης αρχιτεκτονικής με IPFS. Το State ονομάζεται **KYC_info** και υλοποιεί το **LinearState** του API του Corda. Η βασική διαφορά έγκειται στο γεγονός ότι δεν υπάρχει πεδίο IPFS State, καθώς η αποθήκευση των πληροφοριών δεν γίνεται στο IPFS. Υπάρχει όμως ένα

Τα στοιχεία από τα οποία αποτελείται το State είναι :

Client	ο πελάτης ο οποίος είναι αυτός που ξεκινά την διαδικασία δημιουργίας του State.
Institution	ο χρηματοπιστωτικός οργανισμός στον οποίο γίνεται η αρχική κοινοποίηση των KYC δεδομένων.
Proposer	αναφέρεται στην πλευρά εκείνη η οποία ξεκινάει την ροή εργασιών
invoiceattachementID	Το ID του attachment που περιέχει τα δεδομένα KYC που κοινοποιούνται
TypeofDocuments	Καθορίζει το είδος των εγγράφων τα οποία κοινοποιούνται από τον πελάτη προς τον οργανισμό.
Comments	περιλαμβάνει τα σχόλια τα οποία επιθυμεί ο χρήστης να συνοδεύσει την κοινοποίηση, στο οποίο μπορεί να παρέμβει και ο οργανισμός
Further Information	πεδίο στο οποίο ο πελάτης μπορεί να προσθέσει παραπάνω πληροφορίες σχετικές με τα υπό διαμοιρασμό δεδομένα
Verifiedby3rdParty	καθορίζει το κατά πόσο τα υπάρχοντα δεδομένα έχουν κριθεί έγκυρα από τον πρώτο οργανισμό στον οποίο κοινοποιήθηκαν

Participants	Η πιο βασική ιδιότητα του State. Καθορίζει τους χρήστες αυτούς οι οποίοι είναι σχετικοί με το συγκεκριμένο State. Αυτόματα, όπως θα φανεί στην συνέχεια, κάθε State που είναι unconsumed βρίσκεται στο vault όλων των participants. Επομένως, ορίζοντας συγκεκριμένους participants, εξασφαλίζεται ότι αυτοί θα λάβουν γνώση για την ύπαρξη του State.



Διάγραμμα 16 Το KYC_Info State στην εφαρμογή με χρήση attachments (όπως φαίνεται στο CDL Smart Contract View)

Καταστάσεις (status) του State: (εντελώς ανάλογα με την αρχιτεκτονική μέσω IPFS)

PROPOSED: Η κατάσταση ***PROPOSED*** αποτελεί την πρώτη κατάσταση (status) στην οποία μεταβαίνει το KYC_info την στιγμή που δημιουργείται. Το State δεν περιέχει το id κάποιου attachment ακόμα. Η επιβεβαίωση από πλευράς οργανισμού γίνεται μέσω των υπόλοιπων στοιχείων του State όπως comments, furtherinformation κλπ.

RELEASED: Η κατάσταση **RELEASED** αποτελεί την δεύτερη κατάσταση (status) στην οποία μεταβαίνει το KYC_Info, αφού έχει δοθεί μέσω της συλλογής υπογραφών (βλ. Συλλογή υπογραφών), η "άδεια" από τον οργανισμό να δημιουργηθεί η proposed State.

UPDATED: Η κατάσταση **UPDATED** αποτελεί την τελική κατάσταση στην οποία μπορεί να βρεθεί το KYC_Info. Από την στιγμή που βρίσκεται σε αυτή την κατάσταση μπορεί να μεταβεί πάλι μόνο στην ίδια κατάσταση. Επιτρέπονται πολύ συγκεκριμένες αλλαγές στα πεδία του State όταν αυτό μεταβαίνει από την κατάσταση released σε updated ή από την κατάσταση updated σε updated.

3.3.3 Το Contract

Όπως αναφέρθηκε και στην πρώτη αρχιτεκτονική, το contract (συμβόλαιο) καθορίζει ποιες συναλλαγές θεωρούνται έγκυρες κατά την εκτέλεση μία εφαρμογής Cordapp. Συνολικά οι ροές εργασιών (βλ. workflows) και της συγκεκριμένης αρχιτεκτονικής της εφαρμογής είναι τρεις. Αυτές είναι η δημιουργία του KYC_Info, δηλαδή η αποθήκευση στο ledger του State το οποίο ακόμα δεν περιέχει την προς κοινοποίηση πληροφορία, η κοινοποίηση του συνόλου των δεδομένων (release), και τέλος η ενημέρωση των πληροφοριών (update). Συνολικά, απαιτείται ένα contract το οποίο όμως αποτελείται από 3 commands, ένα command για κάθε μετάβαση-ροή εργασιών.

Τα **Commands** στο Corda καθορίζουν την πρόθεση που έχει μία συναλλαγή. Στην προτεινόμενη υλοποίηση υπάρχουν 3 commands που πρέπει να υλοποιηθούν και αντιπροσωπεύουν τις 3 δυνατές μεταβάσεις του State. Τα contracts στην συγκεκριμένη αρχιτεκτονική με χρήση attachments είναι πολύ παρόμοια με τα αντίστοιχα contracts της αρχιτεκτονικής με IPFS που αναφέρθηκε παραπάνω:

- **Propose**: αποτελεί στην ουσία το command που αφορά την πρώτη ροή εργασιών-συναλλαγή την οποία εκκινεί ο client και δημιουργείται το KYC_Info. Οι περιορισμοί που τίθενται στο συγκεκριμένο κομμάτι :
  - Πρέπει να υπάρχει μόνο ένα State που δημιουργείται
  - Διατήρηση του linear id
  - Οι τιμές ορισμένων πεδίων είναι αυστηρά καθορισμένες
  - Δεν καταναλώνεται κάποιο ήδη υπάρχον State αλλά δημιουργείται ένα καινούργιο
  - Ορισμένα πεδία δεν πρέπει να είναι κενά
  - Καθορίζεται ότι πρέπει να υπογράψει τόσο ο πελάτης όσο και ο αντίστοιχος οργανισμός. Υπενθυμίζεται ότι το contract δεν ελέγχει την παρουσία υπογραφών αλλά την παρουσία των parties που υπογράφουν.

- **Release**: αποτελεί το command που καθορίζει την μετάβαση ενός State από κατάσταση "PROPOSED" σε "RELEASED". Δηλαδή, ελέγχει την συναλλαγή που πραγματοποιείται προκειμένου ο πελάτης να κοινοποιήσει το σύνολο των δεδομένων KYC στον αντίστοιχο οργανισμό. Στην συγκεκριμένη υλοποίηση αυτό επιτυγχάνεται μέσω της προσθήκης attachment στην συναλλαγή. Οι περιορισμοί που τίθενται περιλαμβάνουν τους ακόλουθους:
  - Καθορίζεται ότι μόνο ένα State μπορεί να είναι το input και το output της αντίστοιχης συναλλαγής
  - Διατήρηση του linear id
  - Οι τιμές ορισμένων πεδίων πρέπει να είναι αυστηρά καθορισμένες.
  - Ορισμένα πεδία δεν πρέπει να είναι κενά
  - Όλα τα πεδία εκτός από το comments, further explanation, status, invoiceattachementID πρέπει να παραμένουν αμετάβλητα. Ο συγκεκριμένος περιορισμός τίθεται προκειμένου το μόνο που να δύναται να αλλάξει να είναι το περιεχόμενο των προς κοινοποίηση πληροφοριών
  - Συγκεκριμένα, όπως καθορίστηκε και με την τιμή του IPFS hash, και εδώ η τιμή του invoiceattachementID καθορίζεται ότι πρέπει να αλλάξει. Από null που ήταν στο PROPOSED State (αφού δεν είχε κοινοποιηθεί ακόμα πληροφορία), πρέπει να αποκτήσει κάποια τιμή.
  - Καθορίζεται ότι αρκεί να υπογράψει μόνο ο πελάτης. Υπενθυμίζεται, ότι στο πρώτο στάδιο, μέσω της παροχής της υπογραφής, ο οργανισμός (institution) είχε παρέχει συναινέσει ώστε να του κοινοποιηθούν τα δεδομένα.

- **Update**: αποτελεί το command που καθορίζει την ενημέρωση του State είτε από "RELEASED" σε "UPDATED" είτε από "UPDATED" σε "UPDATED". Ελέγχει με αυτόν τον τρόπο τις συναλλαγές που αφορούν μεταβολές στην "τελική" κατάσταση κάθε State. Οι αλλαγές αυτές περιλαμβάνουν τόσο αλλαγές στα δεδομένα προς κοινοποιούνται, επομένως αλλαγές στα πεδία IPFS hash, comments, further explanation, typeofdocuments αλλά και αλλαγές στους participants, δηλαδή προσθήκες (ή αφαιρέσεις) οργανισμών που έχουν πρόσβαση στα πιο πρόσφατα δεδομένα. Οι περιορισμοί που ορίζει το contract στην συγκεκριμένη εντολή είναι:

  - Καθορίζεται ότι μόνο ένα State μπορεί να είναι το input και το output της αντίστοιχης συναλλαγής
  - Ορισμένα πεδία δεν πρέπει να είναι κενά
  - Διατήρηση του linear id
  - Καθορίζεται ότι αρκεί να υπογράψει μόνο ο πελάτης. Μόνο ο πελάτης πρέπει να συναινέσει στην αλλαγή των πληροφοριών που κοινοποιεί.

3.3.4 Τα Flows

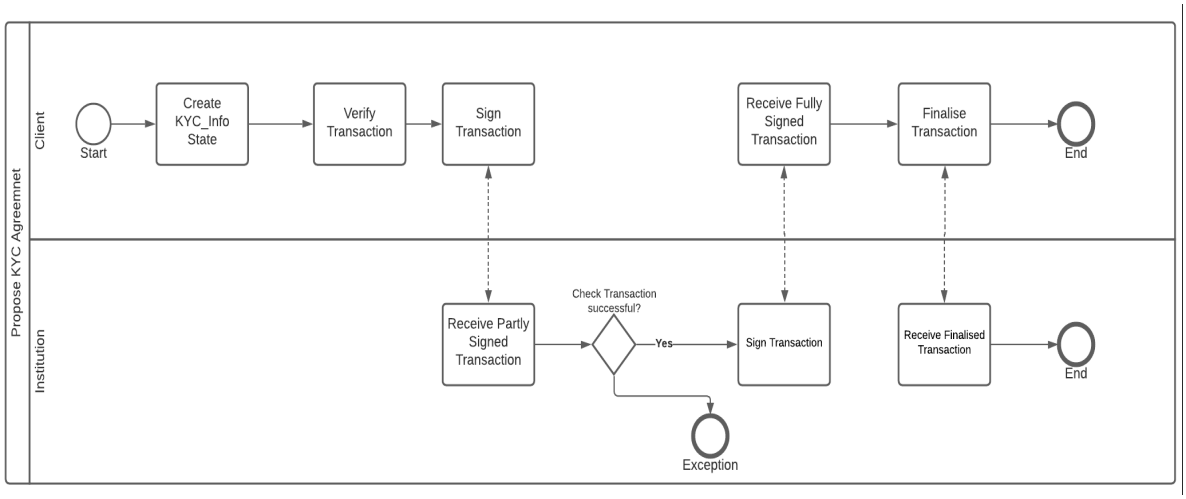
Στην αρχιτεκτονική της εφαρμογής που γίνεται χρήση attachments για την κοινοποίηση των δεδομένων, έχουν υλοποιηθεί 3 flows, όπως συνέβη και στην περίπτωση των IPFS hash. Υπενθυμίζεται ότι ως flow ορίζεται μία σειρά από ενέργειες που εκτελούνται στο δίκτυο Corda, με σειρά που έχει καθοριστεί αυστηρά από τον προγραμματιστή. Όπως αναφέρθηκε στο κομμάτι των **flows** στο Corda, χρησιμοποιείται point to point messaging. Επομένως, πρέπει να καθορίζεται αυστηρά από τον προγραμματιστή της εφαρμογής το ποιος στέλνει μία πληροφορία, σε ποιόν την στέλνει καθώς και με ποια σειρά. Στην υλοποίηση που προτείνεται στην συγκεκριμένη εργασία, έχουν οριστεί συνολικά 3 flows. Αυτές αποτελούν την ραχοκοκαλιά της εφαρμογής, και καθορίζουν την αλληλουχία των γεγονότων.

3.2.4.1 First Flow (Propose)

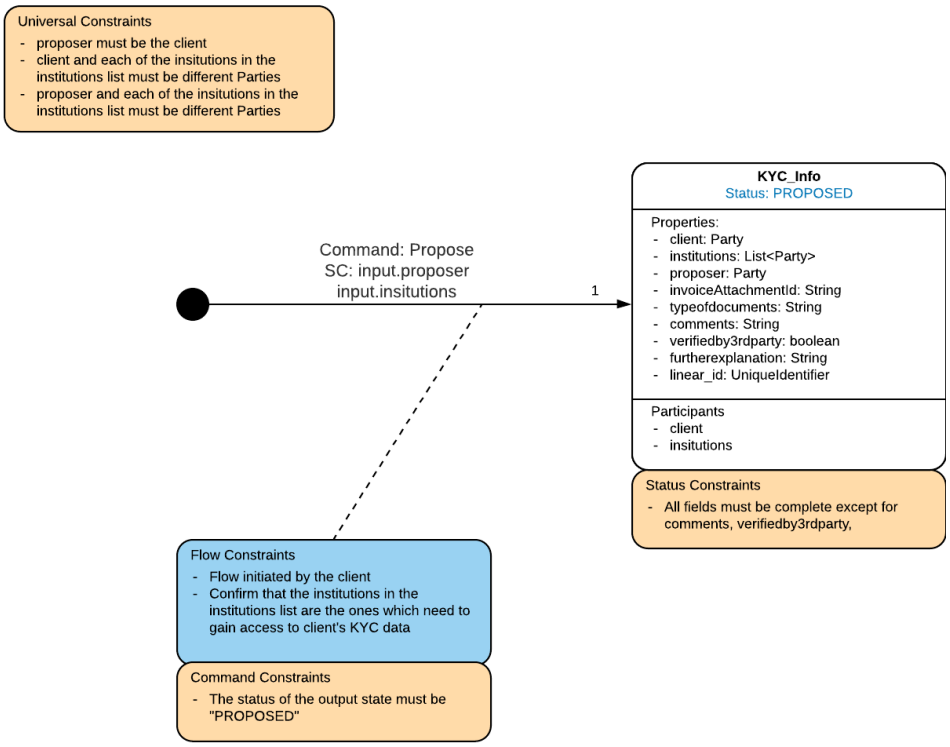
Όπως και στην αρχιτεκτονική με IPFS, βασικός σκοπός την πρώτης ροής εργασιών είναι η πραγματοποίηση μία συναλλαγής που θα εκδώσει το KYC_Info State, το οποίο αντιπροσωπεύει την κατάσταση της κοινοποίησης των πληροφοριών KYC. Το party που εκκινεί την First Flow είναι ο πελάτης (client).
Αρχικά, ο client υλοποιεί ένα Transaction Builder προκειμένου να ξεκινήσει την διαδικασία δημιουργίας της συναλλαγής. Προφανώς, επειδή δημιουργείται ένα καινούργιο State, αφού δεν έχει υπάρξει κάποια συναλλαγή μεταξύ πελάτη και οργανισμών πιο πριν (τουλάχιστον που να αφορά την κοινοποίηση των KYC data), δεν υπάρχει κάποιο input State στην συναλλαγή. Το State που δημιουργείται έχει ως Status “PROPOSED”. Το State αυτό, έχει ως πεδία βασικές πληροφορίες για την συναλλαγή τις οποίες έχει ορίσει ο πελάτης. Προγραμματιστικά, αυτό γίνεται μέσω των ορισμάτων που περνάει ο χρήστης-πελάτης.

Στην συνέχεια, προστίθεται το αντίστοιχο **Command** το οποίο δείχνει ουσιαστικά ποιο κομμάτι του συμβολαίου (Contract) διέπει την συγκεκριμένη συναλλαγή (έκδοση State).
Αφού δημιουργείται η συναλλαγή, ελέγχεται αν αυτή είναι valid (validation process), δηλαδή αν υπακούει το σύνολο των κανόνων που έχουν οριστεί στο Contract. Όπως αναφέρθηκε και στην αντίστοιχη περίπτωση του First Flow της πρώτης αρχιτεκτονικής (με IPFS), το σημείο αυτό δεν ελέγχει την παρουσία των υπογραφών, αλλά των Parties που θα υπογράψουν.
Τέλος, αφού απαιτείται και η συλλογή της υπογραφής της άλλης μεριάς, δηλαδή του (των) Institution, γίνεται εκκίνηση ενός (ή περισσοτέρων sub-Flows) τα οποία έχουν ως στόχο την συλλογή των απαιτούμενων υπογραφών.

Παρακάτω παρουσιάζονται τα διαγράμματα BPMN και Smart Contract View που αφορούν την First Flow, όπου γίνεται κατανοητή η διαδικασία που ακολουθείται.



Διάγραμμα 17 Το BPMN διάγραμμα για το First Flow (attachments)



Διάγραμμα 18 Το μέρος του CDL Smart Contract View που παρουσιάζει τις διαδικασίες της First Flow (attachments)

3.2.4.2 Second Flow (Release)

Το δεύτερο flow, ή **Second Flow (Release Flow)** όπως ονομάστηκε, σε πλήρη αντιστοίχιση με τη αρχιτεκτονική με IPFS έχει σκοπό να πραγματοποιήσει μία συναλλαγή (transaction) προκειμένου να αλλάξει ορισμένα πεδία του State KYC_Info. Το party που εκκινεί το συγκεκριμένο flow είναι και πάλι ο **client (proposer).**
Η μεγάλη διαφορά των υλοποιήσεων με αποθήκευση σε IPFS και με διαμοιρασμό μέσω attachments αντικατοπτρίζεται κυρίως στο Second Flow. Στην συγκεκριμένη υλοποίηση με attachments, κατά την διάρκεια του Second Flow γίνεται προσθήκη attachment στην συναλλαγή (transaction), τα οποία περιέχουν τα KYC δεδομένα προς κοινοποίηση. Παρακάτω παρουσιάζεται η διαδικασία αναλυτικά.

Αρχικά, πρέπει να «βρεθεί» το State εκείνο που πρέπει να καταναλωθεί και να δηλωθεί ως CONSUMED και να δημιουργηθεί ένα άλλο στην θέση του. Ο κόμβος client πραγματοποιεί ένα Vault Query, δηλαδή μία αναζήτηση στο Vault του, προκειμένου να βρεθεί το State αυτό. Η αναζήτηση μπορεί να γίνει με διάφορα κριτήρια. Στην συγκεκριμένη υλοποίηση, όπως και στην υλοποίηση με IPFS, γίνεται αναζήτηση με κριτήριο το linearKYCInfo, δηλαδή το id hash του State. Εναλλακτικά, για μεγαλύτερη ευκολία, θα μπορούσε να πραγματοποιηθεί ένα Vault Query με κριτήριο τους participants του State. Το Vault Query φαίνεται παρακάτω:

```
QueryCriteria criteriaforProposedKYC = new
QueryCriteria.LinearStateQueryCriteria(
                    null,
                    ImmutableList.of(linearIdKYCInfo),
                    Vault.StateStatus.UNCONSUMED,
                    null
            );
```

Αφού πραγματοποιηθεί το Vault Query και έχει βρεθεί το προς κατανάλωση State, συλλέγεται και το πλήθος των πληροφοριών που θα αποτελούν τις τιμές των πεδίων του νέου output State. Το πιο σημαντικό κομμάτι της διαδικασίας αποτελεί το ανέβασμα (uploading) του attachment. Ο client εισάγει ως όρισμα κατά την εκκίνηση της ροής Second Flow το path που αντιστοιχεί στο αρχείο JAR που περιέχει το σύνολο των προς κοινοποίηση δεδομένων. Καλώντας μία βοηθητική συνάρτηση uploadAttachment, γίνεται η αποθήκευση του αρχείου στην τοπική μνήμη του κόμβου. Στην συνέχεια ο client λαμβάνει την τιμή του attachment hash που αντιστοιχεί στο συγκεκριμένο αρχείο, όπως φαίνεται παρακάτω:

```
try {
                attachmentHash = SecureHash.parse(uploadAttachment(
                        path,
                        getServiceHub(),
                        getOurIdentity(),
                        path)
                );
            } catch (IOException e) {
                e.printStackTrace();
            }
```

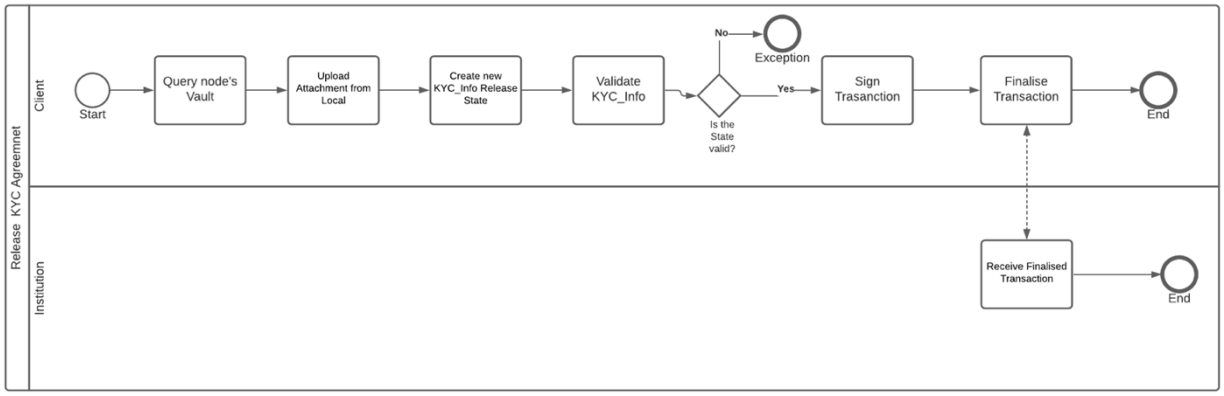
Μετά την ολοκλήρωση της διαδικασίας αυτής, κατασκευάζεται η συναλλαγή μέσω του TransactionBuilder. Ορίζεται ως input State το State που έχει ανακτηθεί από το vault με την διαδικασία που αναφέρθηκε παραπάνω. Ακόμη, κατασκευάζεται και το νέο KYC_info State το οποίο έχει πλέον Status "RELEASED", περιέχει τις ενημερωμένες πληροφορίες στα πεδία του καθώς και την τιμή του Attachment Id.

Αφού προστεθεί και το Output State, προστίθεται στο Transaction και το Attachment. Η διαδικασία αυτή έχει ως αποτέλεσμα όλοι οι συμμετέχοντες της συγκεκριμένης συναλλαγής να λάβουν το hash του attachment. **Δεν γίνεται άμεση αποστολή του αρχείου JAR**. Αντίθετα, οι participants (εδώ ο client και οι institutions) αποκτούν γνώση του attachment hash. Όταν στην συνέχεια θελήσουν να ανοίξουν το αρχείο, τότε και μόνο τότε θα αποθηκεύσουν το αρχείο στην τοπική τους μνήμη.

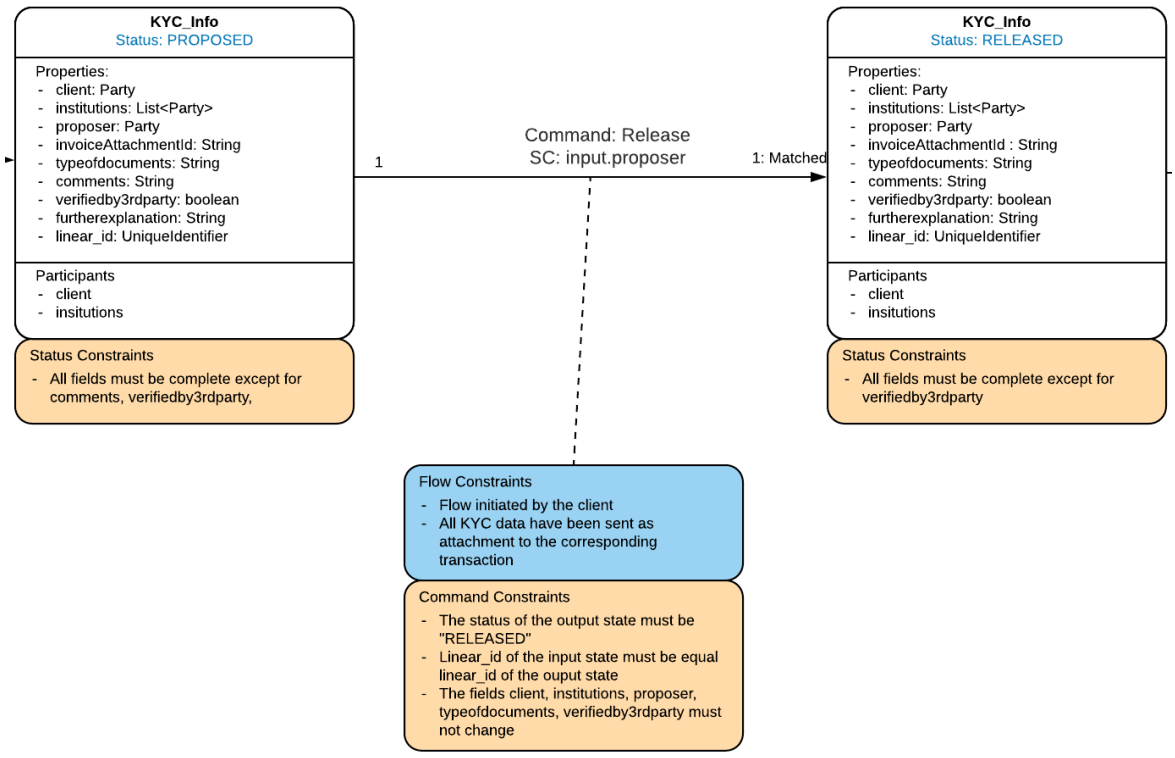
```
final TransactionBuilder txBuilder = new TransactionBuilder(notary)
                        .addCommand(txCommand)
                        .addInputState(inputState)
                        .addOutputState(KYCInfo_release,
KYCContract.KYC_CONTRACT_ID)
                        .addAttachment(attachmentHash);
```

Στην συνέχεια, γίνεται το verification της συναλλαγής. Το Command που καθορίζει ποιο σημείο του Contract διέπει το σύνολο των επιτρεπτών αλλαγών στο State, είναι το Release Command. Και στην συγκεκριμένη περίπτωση, όπως συμβαίνει και στην υλοποίηση με IPFS, δεν απαιτείται συλλογή υπογραφών. Για την ενημέρωση του State, δηλαδή στην ουσία για την κοινοποίηση των πληροφοριών KYC στα institutions δεν χρειάζεται κάποια έγκριση από τα Institutions Parties.

Τέλος, αφού έχει δημιουργηθεί μία πλήρως υπογεγραμμένη συναλλαγή (fully signed transaction), ξεκινάνε sub-flows, μια για κάθε institution, προκειμένου να ενημερωθούν (ως participants), σχετικά με το νέο State. Ακολούθως φαίνεται και το BPMN διάγραμμα που περιγράφει την διαδικασία της δεύτερης (Release Flow) μέσω attachments.



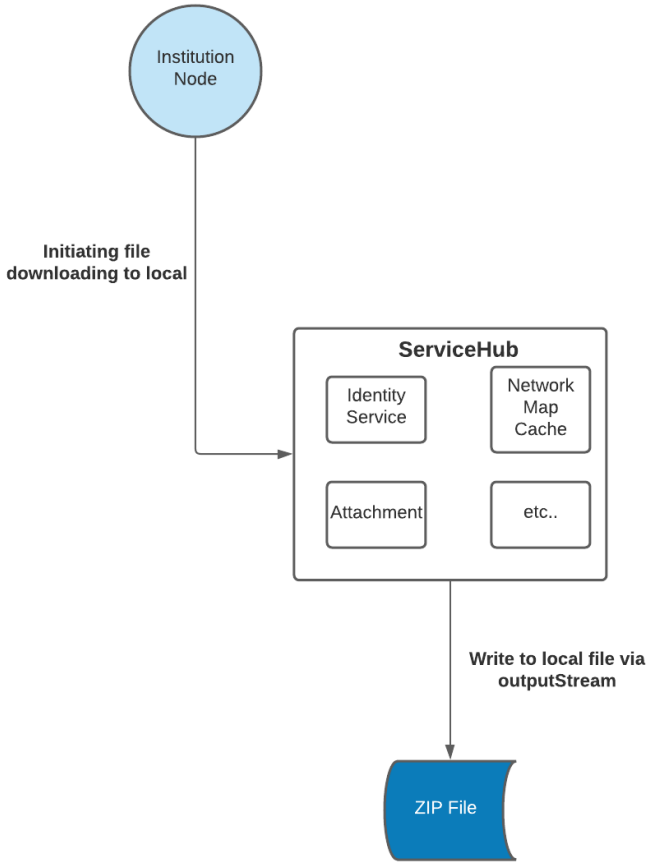
Διάγραμμα 17 Το BPMN διάγραμμα για το Second Flow



Διάγραμμα 18 Το CDL Smart Contract View για το Second Flow (attachments)

3.2.4.3 Download Flow

Μετά την εκτέλεση των δύο παραπάνω ροών εργασιών, εάν έχουν γίνει όλα με επιτυχία, τόσο ο πελάτης (client) όσο και το σύνολο των οργανισμών (institutions) έχουν στο attachments Service τους το AttachmentId του αρχείου που περιέχει τα KYC δεδομένα. Προκειμένου κάθε εμπλεκόμενο Party να μπορεί να αποθηκεύσει και τοπικά το αντίγραφο των δεδομένων, έχει δημιουργηθεί, πέραν των 3 βασικών flows (First-Propose, Second-Release, Third-Update) και μία επιπλέον βοηθητική ροή εργασιών προκειμένου αυτή η διαδικασία να αυτοματοποιηθεί. Η διαδικασία περιγράφεται από το ακόλουθο διάγραμμα:



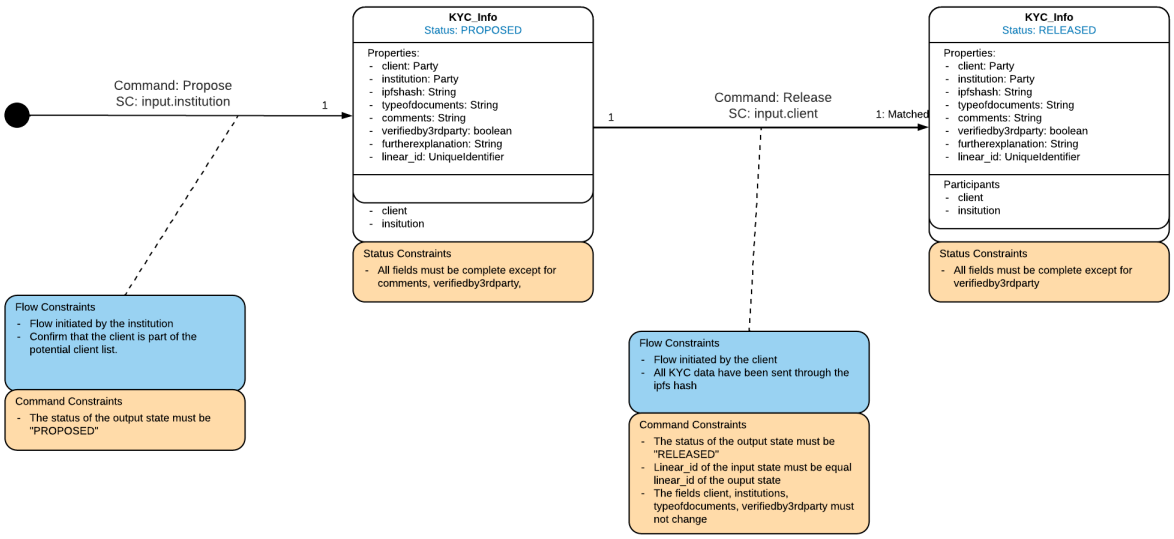
Διάγραμμα 19 Σχηματική αναπαράσταση της Third Flow (Download)

3.4 Εναλλακτική υλοποίηση με εκκίνηση διαδικασίας από τον οργανισμό

Οι παραπάνω δύο υλοποιήσεις βασίζονται στην υπόθεση ότι ο πελάτης είναι αυτός ο οποίος εκκινεί την διαδικασία κοινοποίησης των δεδομένων KYC. Η μοναδική αρμοδιότητα του οργανισμού είναι ο -αυτόματος εν μέρει- έλεγχος της αρχικής συναλλαγής (First Flow Transaction), προκειμένου να διαπιστωθεί αν έγινε η κοινοποίηση στον σωστό οργανισμό-institution. Σε περίπτωση που ο ίδιος ο χρηματοπιστωτικός οργανισμός είναι αυτός που επιθυμεί να εκκινήσει την διαδικασία διαμοιρασμού των KYC δεδομένων, προτείνεται η ακόλουθη υλοποίηση (με χρήση του IPFS για την αποθήκευση των δεδομένων).

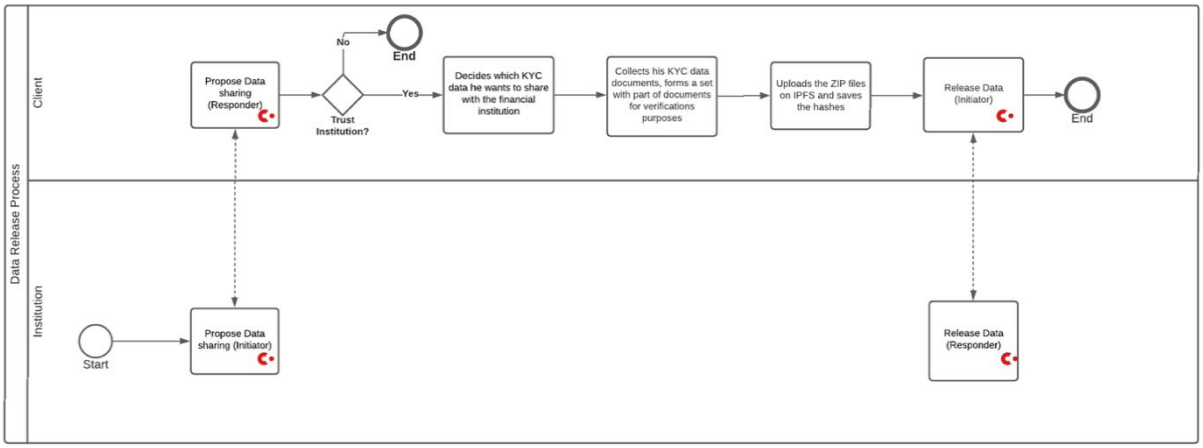
Αρχικά, ο οργανισμός αφού βεβαιωθεί για την ταυτότητα του πελάτη από τον οποίο θα ζητήσει τα δεδομένα, εκκινεί μία πρώτη ροή εργασιών (Request Flow), προκειμένου να ενημερώσει τον πελάτη για το αίτημα αυτό. Δημιουργείται έτσι ένα νέο KYC_Info State το οποίο δεν περιέχει κάποιο IPFShash, παρά μόνον πληροφορίες σχετικά με το αίτημα της τράπεζας στα πεδία Comments και Further Information.

Στην συνέχεια, ο πελάτης αποφασίζει (off-ledger) για το κατά πόσο επιθυμεί να κοινοποιήσει στον συγκεκριμένο οργανισμό τα δεδομένα του. Αν αποφασίσει ότι δεν επιθυμεί την κοινοποίηση των δεδομένων που ζητάει η τράπεζα τότε δεν προχωρά σε κάποια ενέργεια. Αντιθέτως, εάν συμφωνεί με τους όρους που έθεσε ο οργανισμός και επιθυμεί την κοινοποίηση των δεδομένων του, τότε εκκινεί μία ροή εργασιών (Release Flow), προκειμένου να προβεί σε αυτή την κοινοποίηση.



Διάγραμμα 20 To CDL Smart Contract View της εναλλακτικής υλοποίησης

Τελικά, ο ίδιος ο πελάτης έχει την δυνατότητα να ανανεώσει ο ίδιος τις πληροφορίες που περιέχονται στο KYC_Info State.



Διάγραμμα 21 Το BPMN διάγραμμα της εναλλακτικής υλοποίησης

Κεφάλαιο 4

Συγκριτική μελέτη - Αποτελέσματα

Η διαδικασία αποστολής των KYC δεδομένων από έναν πελάτη (client) σε έναν ή περισσότερους οργανισμούς (institutions) μπορεί να γίνει και μέσω της χρήσης συνημμένων (attachments), μίας δυνατότητας που παρέχεται από το Corda. Έγινε εκτενής ανάλυση της διαδικασίας αυτής στο Κεφάλαιο 3.

Στην συγκεκριμένη ενότητα συγκρίνεται η χρονική απόδοση αποστολής των δεδομένων από τον πελάτη σε 1 ή περισσότερους οργανισμούς ως προς 1) το πλήθος των οργανισμών στους οποίους κοινοποιούνται τα δεδομένα και 2) ως προς το μέγεθος των προς αποστολή δεδομένων. Διαπιστώνεται μία διαφοροποίηση στους χρόνους αποστολής των δεδομένων όταν οι παραπάνω παράμετροι αλλάζουν.

4.1 Μελέτη χρόνου συναρτήσει αριθμού institutions

Το Corda επιτρέπει τον διαμοιρασμό των δεδομένων σε μία need-to-know basis δηλαδή, όπως έχει αναφερθεί ήδη, τα δεδομένα προς αποστολή δεν μεταδίδονται στο σύνολο των συμμετεχόντων στο δίκτυο. Αντιθέτως, μεταδίδονται τα δεδομένα μόνον σε αυτούς οι οποίοι έχουν οριστεί να τα λάβουν.

Στην περίπτωση της εφαρμογής που πραγματεύεται η συγκεκριμένη διπλωματική εργασία, ο πελάτης αποφασίζει να κοινοποιήσει τα προς εξέταση KYC δεδομένα σε έναν χρηματοπιστωτικό (ή άλλο) οργανισμό. Είναι εύκολα κατανοητό το γεγονός ότι, υπό πραγματικές συνθήκες, ενδέχεται ο αριθμός των οργανισμών στους οποίους ο πελάτης θέλει να κοινοποιήσει τα δεδομένα να είναι μεγαλύτερος του ενός. Στο συγκεκριμένο σημείο μελετάμε την χρονική απόδοση της εφαρμογής σε περιπτώσεις που ο αριθμός των οργανισμών που θα λάβουν τα δεδομένα είναι παραπάνω από 1. Συγκεκριμένα μελετάμε την απόδοση για 1 έως και 5 οργανισμούς.

Δομή του δικτύου: Οι μετρήσεις έγιναν σε ένα τοπικό δίκτυο (local network) το οποίο περιλάμβανε έναν κόμβο (node) που λειτουργεί ως Notary, έναν κόμβο ο οποίος έχει το όνομα PartyA και αποτελεί τον πελάτη (client) καθώς και έως 5 κόμβους οι οποίοι ονομάζονται PartyB, PartyC, PartyD, PartyE, PartyF και οι οποίοι λειτουργούν ως οι οργανισμοί στους οποίους θα κοινοποιηθούν τα KYC δεδομένα.

First Flow

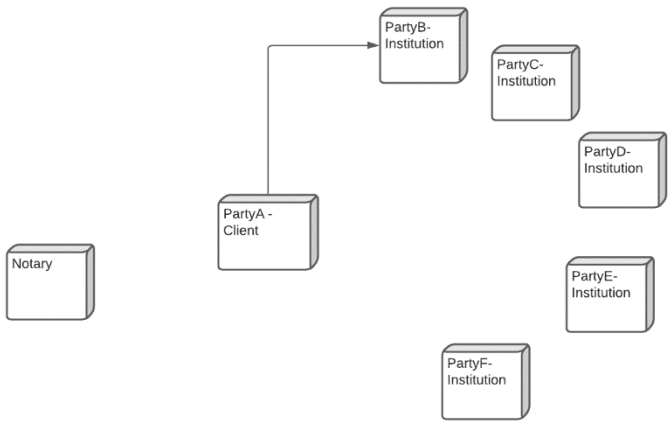
Έγιναν συνολικά 5 πειράματα. Στο πρώτο πείραμα, ο πελάτης ξεκινάει 3 ροές εργασιών First Flow, κάθε μία από τις οποίες περιλαμβάνει μία συναλλαγή (transaction) που έχει ως output ένα State. Στο κάθε State που περιλαμβάνεται ως output στην εκάστοτε συναλλαγή (transaction), ως participants ορίζονται ο πελάτης καθώς και **ένας οργανισμός (institution).**

Στο δεύτερο πείραμα, ο πελάτης ξεκινάει πάλι 3 ροές εργασιών First Flow, οι οποίες περιλάμβαναν συναλλαγές οι οποίες είχαν ως output ένα State, με participants τον πελάτη καθώς και **δύο (2) οργανισμούς (institutions).**

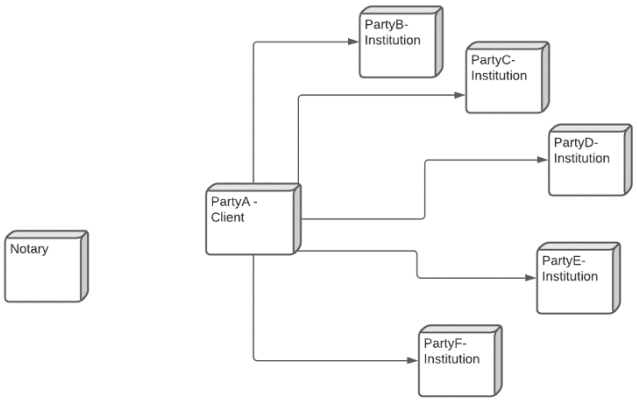
Στο τρίτο πείραμα, ο πελάτης ξεκινάει πάλι 3 ροές εργασιών First Flow, οι οποίες περιλάμβαναν συναλλαγές οι οποίες είχαν ως output ένα State, με participants τον πελάτη καθώς και **τρεις (3) οργανισμούς (institutions).**

Στο τέταρτο πείραμα, ο πελάτης ξεκινάει πάλι 3 ροές εργασιών First Flow, οι οποίες περιλάμβαναν συναλλαγές οι οποίες είχαν ως output ένα State, με participants τον πελάτη καθώς και **τέσσερις (4) οργανισμούς (institutions).**

Στο πέμπτο πείραμα, ο πελάτης ξεκινάει πάλι 3 ροές εργασιών First Flow, οι οποίες περιλάμβαναν συναλλαγές οι οποίες είχαν ως output ένα State, με participants τον πελάτη καθώς και **πέντε (5) οργανισμούς (institutions).**



Διάγραμμα 22 Καθώς το Corda χρησιμοποιεί κοινοποίηση δεδομένων σε μία need-to-know basis μόνο, ο πελάτης κάθε φορά επιλέγει ποιοι είναι οι participants του KYC_Info State, από 1 (πάνω) έως 5 (κάτω)



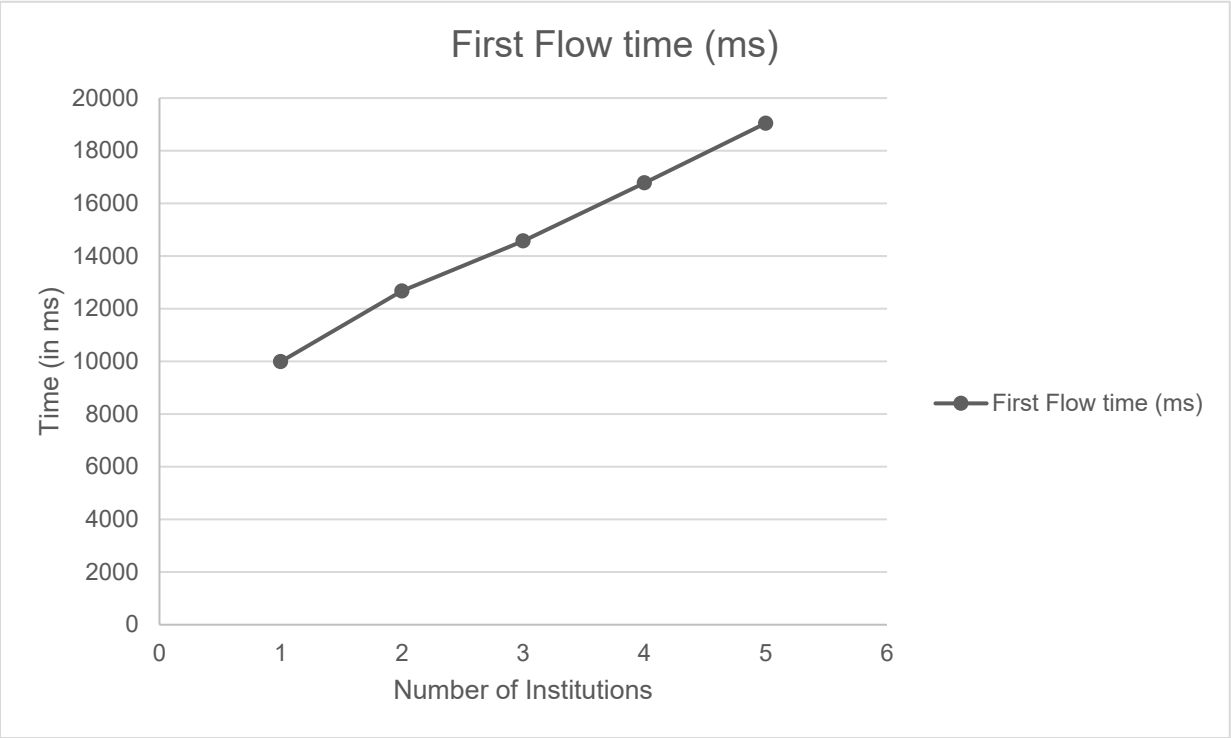
Προφανώς, όπως αναφέρθηκε και νωρίτερα στην συγκεκριμένα εργασία, το γεγονός ότι ορίζονται ως συμμετέχοντες (participants) οι κόμβοι-οργανισμοί (institutions), προκαλεί την αποθήκευση των States στα vault του κάθε κόμβου – participant. Ακόμη, κατά την διάρκεια της κάθε ροής εργασιών απαιτείται η συλλογή υπογραφών από το σύνολο των συμμετεχόντων – participants που ορίζεται. Σε καθεμία περίπτωση, προκαλείται η εκκίνηση επιμέρους ροών (sub- flows) που είναι υπεύθυνες για την συλλογή των υπογραφών από τους κατάλληλους κόμβους. Η διαδικασία αυτή έχει περιγράφει εκτενώς. Στην συγκεκριμένη ενότητα, αυτό που μας ενδιαφέρει είναι το γεγονός ότι σε καθεμία διαφορετική περίπτωση- πείραμα απαιτείται η συλλογή υπογραφών από 1 έως 5 κόμβους. Ακόμη, απαιτείται από 5 κόμβους να αποθηκεύσουν το εκάστοτε State στο vault τους.

Σε κάθε πείραμα, γίνεται η μέτρηση του συνολικού χρόνου εκκίνησης των 3 ροών (flow), της υπογραφής της συναλλαγής (transaction) από τον πελάτη και της αποστολής στα επιμέρους nodes- participants. Το ενδιαφέρον έγκειται στην μελέτη της σχέσης συνολικού χρόνου εκτέλεσης της ροής (flow) και του αριθμού των οργανισμών (institutions) στους οποίους πρόκειται να υλοποιηθούν τα δεδομένα.

Τα αποτελέσματα φαίνονται στον παρακάτω πίνακα:

Αριθμός Institutions	*First Flow time (in ms)*
1	9989
2	12674
3	14547
4	16779
5	19045

Γραφικά τα αποτελέσματα φαίνονται παρακάτω:



Διάγραμμα 23 Ο χρόνος που απαιτείται για να εκτελεστούν 3 First Flows για 1 έως Institutions

Παρατηρήσεις:

Το γεγονός ότι το δίκτυο ήταν ακριβώς το ίδιο και στις 5 περιπτώσεις, αλλά και το γεγονός ότι το μέγεθος των συνημμένων ήταν ακριβώς το ίδιο (400KB για κάθε διαφορετικό flow, 3 flows στο σύνολο), μας επιτρέπει να εξάγουμε χρήσιμα συμπεράσματα για τον ρόλο του αριθμού των οργανισμών (institutions) στην συνολική χρονική επιβάρυνση των flows.

Υπενθυμίζεται ότι ο αριθμός των διαφορετικών οργανισμών οι οποίοι είναι participants στο εκάστοτε State, **είναι ανάλογος με τον αριθμό των επιμέρους sub-flows που δημιουργούνται κατά την διάρκεια του First Flow που εκκινεί ο πελάτης**. Επομένως ενώ στην πρώτη περίπτωση απαιτείται η δημιουργία δύο sub-flows, ένα για συλλογή υπογραφής και ένα για την κοινοποίηση του τελικού transaction- αποθήκευση των output State στο Vault του counter party, όσο αυξάνεται ο αριθμός των institutions, αναλόγως αυξάνεται και ο αριθμός των sub-flows που δημιουργούνται. Επομένως στην $5^{η}$ περίπτωση, θα πρέπει να δημιουργηθούν 10 sub-flows. Ο αριθμός των sub-flows που δημιουργούνται επηρεάζεται από το πλήθος των Parties τα οποία βρίσκονται στην λίστα otherParties, όπως φαίνεται παρακάτω:

```
List<FlowSession> sessions = new ArrayList<>();
for (Party otherParty: otherParties) {
    sessions.add(initiateFlow(otherParty));
}


final SignedTransaction fullySignedtx = subFlow(new
CollectSignaturesFlow(
        partySignedtx,
        sessions
));
```

Κατά την ανάλυση των αποτελεσμάτων, παρατηρούμε ότι ο συνολικός χρόνος εκτέλεσης των 3 First Flows σε κάθε περίπτωση αυξάνεται με μία γραμμική τάση, ανάλογα με τον αριθμό των οργανισμών (institutions) στους οποίους κοινοποιούνται τα δεδομένα. Η συγκεκριμένη παρατήρηση φαίνεται αρκετά λογική.

Αξίζει να επισημανθεί ότι κατά την εκτέλεση ενός flow, το αρχικό flow περιμένει την εκτέλεση όλων των subflow τα οποία εκκίνησε προκειμένου να συνεχίσει την εκτέλεση της. Επομένως, η συλλογή υπογραφών (βλ. Διάγραμμα 17) πρέπει να έχει ολοκληρωθεί από το σύνολο των institutions (counter parties) προκειμένου να προχωρήσει το flow. Αρκεί ένας κόμβος να παρουσιάσει κάποια καθυστέρηση, ώστε να παρουσιαστεί μία συνολική καθυστέρηση.

Συμπερασματικά, σχετικά με την χρήση της συγκεκριμένης εφαρμογής σε πραγματικές συνθήκες, αξίζει να σημειωθεί ότι η διαφορά στους χρόνους είναι αμελητέα. Παρατηρούμε ότι τόσο η αυθεντικοποίηση από το Notary Service, όσο και η αποστολή και συλλογή υπογραφών γίνεται μέσα σε ελάχιστο χρόνο. Το χαρακτηριστικό αυτό είναι εκείνο που δίνει το πλεονέκτημα στο private blockchain έναντι του public blockchain, μεταξύ άλλων. Στην πραγματική αγορά, ένας πελάτης δεν θα χρειαστεί να κοινοποιήσει τα δεδομένα του σε έναν εξαιρετικά μεγάλο αριθμό οργανισμών – συμμετεχόντων στο δίκτυο. Σε περιπτώσεις που οι αριθμοί των συμμετεχόντων κινούνται στα λογικά πλαίσια, όπως δείξαμε στα πειράματα, δεν υπάρχει κάποια μεγάλη χρονική επιβάρυνση.

Second Flow

Στην συνέχεια κάθε πειράματος, αφού δημιουργήθηκαν τα 3 States που θα περιέχουν την πληροφορία με τα KYC δεδομένα, έγινε εκκίνηση από τον node πελάτη (client) των 3 **Second Flow** ροών εργασιών κατά την διάρκεια των οποίων αυτός (σ.σ. ο πελάτης) ανεβάζει τα attachments, δημιουργώντας μια συναλλαγή με input το output του πρώτου State και ως output το νέο State που περιέχει τα attachments (βλ. attachments).

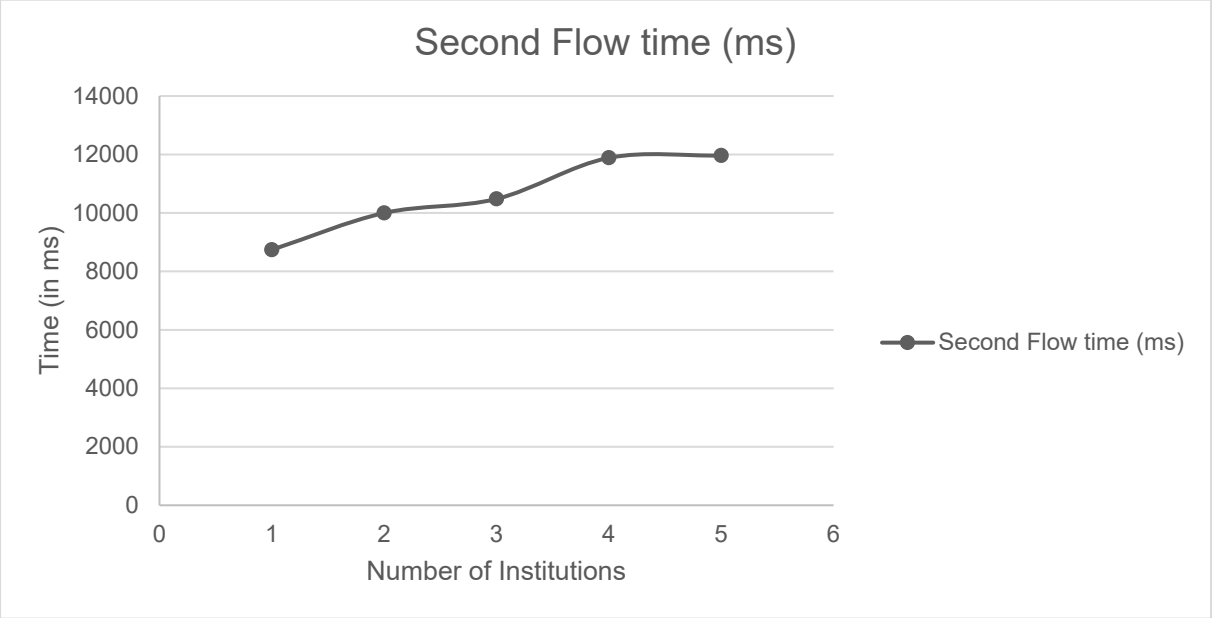
Στην συγκεκριμένη περίπτωση έγινε χρήση attachments μεγέθους 400KB. Μετρήθηκε ο χρόνος που απαιτήθηκε από τον πελάτη να ολοκληρώσει την δεύτερη ροή εργασιών. Αυτή περιλάμβανε την δημιουργία του νέου State, το ανέβασμα των attachments καθώς και την

ενημέρωση των institutions. Όπως έχει περιγράφει και παραπάνω, στο συγκεκριμένο σημείο **δεν απαιτείται συλλογή υπογραφών** από τα εμπλεκόμενα μέρη – institutions καθώς η εξουσιοδότηση εκ μέρους τους έχει ήδη δοθεί κατά την διάρκεια του First Flow.

Παρακάτω φαίνονται τα αποτελέσματα σε κάθε πείραμα. Οι περιπτώσεις, εντελώς ανάλογα με την πρώτη σειρά πειραμάτων, αφορούσαν ύπαρξη 1 έως και 5 institutions στα οποία κοινοποιήθηκαν τα δεδομένα:

Αριθμός Institutions	*Second Flow time (in ms)*
1	8740
2	9999
3	10479
4	11885
5	11964

Γραφικά τα αποτελέσματα φαίνονται παρακάτω:



Διάγραμμα 24 Ο χρόνος εκτέλεσης της Second Flow συναρτήσει του αριθμού των institutions

Παρατηρήσεις:

Η εκτέλεση των 3 Second Flows σε κάθε περίπτωση (που αντιστοιχούν στην «τροποποίηση» του State που δημιουργήθηκε στα First Flows) ήταν σαφώς πιο γρήγορη από αυτή των 3 First Flows. Αυτό το γεγονός οφείλεται στο ότι δεν απαιτείται έλεγχος και υπογραφή από την

πλευρά των οργανισμών. Η δημιουργία των sub-flows από την αρχική ροή εργασίας (workflow) έχει ως μοναδικό σκοπό την αποθήκευση των States στα επιμέρους vaults των οργανισμών (institutions) που έχουν οριστεί ως participants. Επομένως, κατά την διάρκεια εκτέλεσης του Second Flow, δημιουργούνται συνολικά τόσα sub-flows όσα ακριβώς και οι institutions στους οποίους θα κοινοποιηθούν τα KYC δεδομένα. Αυτός ο αριθμός αντιστοιχεί ακριβώς στα μισά sub-flows σε σχέση με την First Flow.

Ένα ενδιαφέρον στοιχείο που παρατηρούμε είναι ότι η αύξηση του χρόνου εκτέλεσης των τριών Second Flows σε κάθε περίπτωση, δεν φαίνεται να επηρεάζεται ιδιαίτερα από τον αριθμό των participants. Παρατηρούμε ότι ενώ για την εκτέλεση των πρώτων ροών εργασίας, που απαιτούσαν την συλλογή υπογραφών, ο χρόνος που απαιτήθηκε αυξανόταν σε σημαντικό ποσοστό όσο αυξάνονταν ο αριθμός των institutions, εδώ η αύξηση είναι πολύ πιο ήπια.

Ο χρόνος που απαιτείται στην βασική περίπτωση, η οποία περιλαμβάνει μόνο έναν (1) institution, αυξάνεται μόνο κατά 36% στην περίπτωση των 5 institutions. Η διαφορά φαίνεται στον παρακάτω πίνακα:

	1 institution	**5 institutions**	**% Increase**
First Flow (Propose)	9989	19045	90,65%
Second Flow (Release)	8740	11964	36,87%

Ενώ θεωρητικά **θα αναμέναμε μεγάλη αύξηση** του χρόνου εξαιτίας της μεταφοράς δεδομένων (των KYC data) η οποία συμβαίνει κατά την διάρκεια του Second Flow, κάτι τέτοιο δεν συμβαίνει τελικά.

Η παρατήρηση αυτή μπορεί εύκολα να εξηγηθεί αν ξαναδούμε την έννοια των συνημμένων (attachments) στο Corda. Όπως έχει αναφερθεί και σε προηγούμενη ενότητα, τα attachments αποτελούν ZIP/JAR αρχεία στα οποία γίνεται αναφορά στις συναλλαγές (transaction) μέσω του hash τους. Τα **attachments δεν περιλαμβάνονται στην συναλλαγή (transaction) αυτή καθαυτή**. Αντιθέτως, κάθε κόμβος αποθηκεύει την λίστα με τα hash των attachments που επιθυμεί και όταν τα χρειαστεί, τα «κατεβάζει», αποθηκεύοντας τα στην μνήμη cache του.

Επομένως, από την στιγμή που δεν γίνεται αποστολή των αρχείων με τα KYC δεδομένα αυτών καθαυτών, αλλά μόνον των hash που δείχνουν σε αυτά, δεν προστίθεται επιπλέον επιβάρυνση στο δίκτυο. Για αυτόν τον λόγο ακριβώς, παρατηρούμε ότι δεν επηρεάζεται ο συνολικός χρόνος ενημέρωσης των States (δηλαδή το Second Flow), εξαιτίας των πολλαπλών participants – institutions. Ναι μεν δημιουργείται περισσότερη κίνηση στο δίκτυο όταν δημιουργούνται 5 αντί για 1 ή 2 sub-flows, όμως δεν απαιτείται η αποστολή των αρχείων πολλαπλές φορές.
Όταν ο κάθε οργανισμός (institution) θελήσει να αναγνώσει τα δεδομένα, απλώς, θα κατεβάσει τα αντίστοιχα attachments με τους τρόπους που περιεγράφηκαν παραπάνω (βλ. Κεφάλαιο 4).

4.2 Μελέτη χρόνου συναρτήσει του μεγέθους δεδομένων προς κοινοποίηση

Η χρήση της αποκεντρωμένης με επίτρεψη πλατφόρμας Corda για την κοινοποίηση ευαίσθητων KYC δεδομένων μεταξύ συμμετεχόντων ενός δικτύου (π.χ. χρηματοπιστωτικών οργανισμών) πρέπει, εκτός από την ιδιωτικότητα, να πληροί ορισμένες προϋποθέσεις που ορίζονται από τις διαδικασίες αυτές καθαυτές. Το Corda δέχεται attachments μόνο σε μορφή JAR/ZIP. Παρόλα αυτά, ο πελάτης (client) ενδέχεται να συμπιέσει αρχεία πολλών διαφορετικών τύπων σε μία μορφή JAR/ZIP.

Ενώ τυπικά δεν υπάρχει μέγιστο μέγεθος (ή αριθμός) συνημμένων (attachments) που μπορούν να προστεθούν σε ένα transaction, κάθε compatibility zone έχει ένα δικό της σετ από παραμέτρους, μία εκ των οποίων είναι και το **maxTransactionSize,** που καθορίζει το μέγιστο μέγεθος κάθε συναλλαγής. Λογικά μεγέθη attachments τα οποία βρίσκονται στα λογικά πλαίσια μεγέθους μίας συναλλαγής σε κάθε compatibility zone (>10 MB), αλλά είναι και ρεαλιστικά μεγέθη για αρχεία που ενδέχεται να συμπιεστούν ως KYC δεδομένα (π.χ. pdf, jpeg, png, doc) είναι τα ακόλουθα:

- 100KB
- 200KB
- 400KB
- 1 MB
- 2MB

Για τις παραπάνω τιμές που αντιπροσωπεύουν το μέγεθος των attachments, πραγματοποιήθηκαν 5 πειράματα, προκειμένου να διαπιστωθεί το πόσο επηρεάζεται ο χρόνος ολοκλήρωσης κάθε ροής εργασιών ανάλογα με το attachment size.

Σε όλες τις μετρήσεις τα χαρακτηριστικά του δικτύου ήταν τα ίδια:

- 1 Notary
- Party A (client)
- Party B (institution)

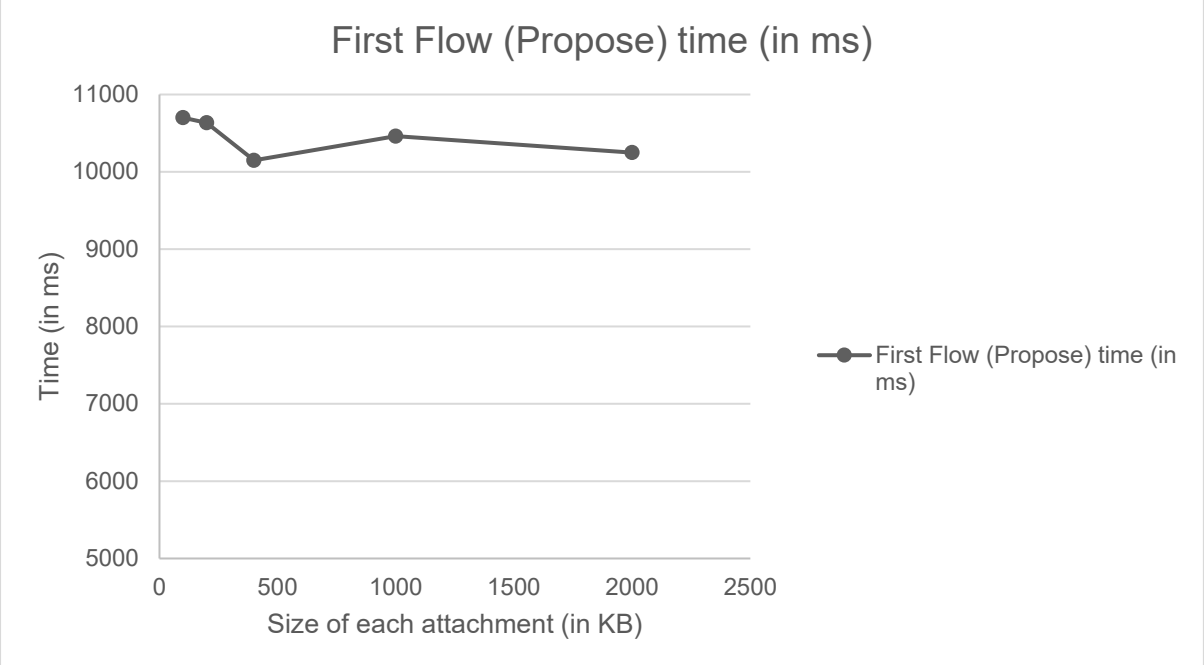
Αρχικά, ο πελάτης (client) εκκίνησε τέσσερις (4) ροές First Flow, με σκοπό την δημιουργία 4 States (Propose flows). Αφού στάλθηκαν στο institution και υπογράφηκαν, ο client "ενημέρωσε" τα States αυτά (σ.σ. δημιούργησε καινούργια States), αφού είχε ανεβάσει τα attachments και είχε περάσει στα νέα States, το αντίστοιχα hash. Προφανώς, εκκίνησε την Second State (Release) 4 φορές.

Οι μετρήσεις αφορούν τον συνολικό χρόνο εκτέλεσης των 4 ροών (που αντιστοιχούν στα 4 attachments) σε καθεμία από τις παρακάτω περιπτώσεις:

- 4 attachments x 100KB
- 4 attachments x 200KB
- 4 attachments x 400KB
- 4 attachments x 1MB
- 4 attachments x 2MB

First Flow (Propose)

Attachment size	**First Flow (Propose) time (in ms)**
4 x 100KB	10700
4 x 200KB	10634
4 x 400KB	10149
4 x 1MB	10461
4 x 2MB	10249



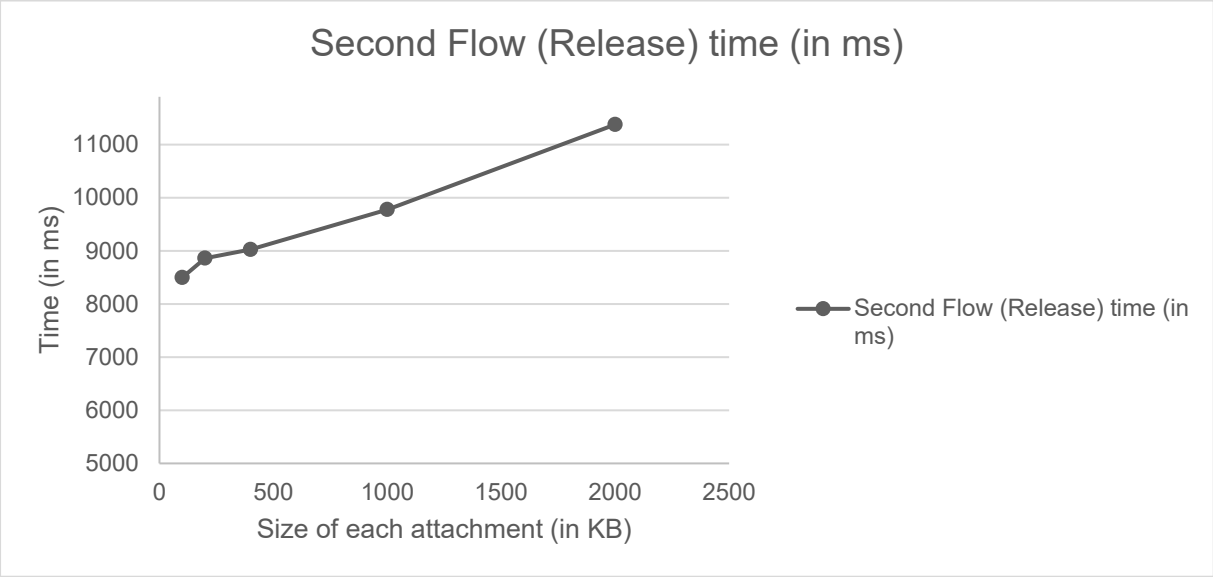
Διάγραμμα 25 Ο χρόνος εκτέλεσης του First Flow συναρτήσει του μεγέθους των attachments προς κοινοποίηση

Παρατήρηση: Προφανώς, κατά την διάρκεια των πρώτων ροών εργασιών, δεν γίνεται εισαγωγή των attachments στην συναλλαγή, επομένως προφανώς δεν φαίνεται καμία αξιόλογη

μεταβολής τους χρόνους εκτέλεσης των flows σε κάθε περίπτωση. Η απεικόνιση έγινε για συγκριτικούς λόγους.

Second Flow (Release)

Attachment size	**Second Flow (Release) time (in ms)**
4 x 100KB	8498
4 x 200KB	8862
4 x 400KB	9026
4 x 1MB	9778
4 x 2MB	11378



Διάγραμμα 26 Ο χρόνος εκτέλεσης του Second Flow συναρτήσει του μεγέθους των attachments προς κοινοποίηση

Παρατηρήσεις:

Εξετάζοντας τα αποτελέσματα σχετικά με τους χρόνους εκτέλεσης των Second Flows στην κάθε περίπτωση μπορούμε, εύκολα, να διαπιστώσουμε ότι το μέγεθος των attachments αποτελεί σημαντικό παράγοντα χρονικής επιβάρυνσης. Συγκεκριμένα, παρατηρούμε ότι στο ίδιο ακριβώς δίκτυο, οι ροές εργασιών που απαιτούν ανέβασμα (uploading) συνημμένων με μέγεθος 2MB απαιτούν περίπου 40% παραπάνω χρόνο από αυτές τις ροές εργασιών που απαιτούν ανέβασμα συνημμένων μεγέθους 100MB.
Παρόλο που αυτή η επιβάρυνση είναι αξιόλογη, δεν αποτελεί παράγοντα μεγάλης αύξησης του συνολικού χρόνου εκτέλεσης των ροών εργασιών σε περιπτώσεις μεγάλου δικτύου.

Υπενθυμίζεται ότι κατά την εκτέλεση της Second Flow (Release), ο πελάτης ανεβάζει στην μνήμη του κόμβου του το προς κοινοποίηση συνημμένο. Όταν στην συνέχεια αποστέλλει τα States στους οργανισμούς (institutions), δεν στέλνει μαζί και τα αρχεία αυτά καθαυτά. Αντίθετα στέλνει μόνο το hash που δείχνει σε αυτά. Επομένως, η επιβάρυνση προέρχεται μόνον από το ανέβασμα (uploading) των attachments, και δεν πολλαπλασιάζεται όσο αυξάνεται ο αριθμός των κόμβων – δεκτών (εδώ institutions).

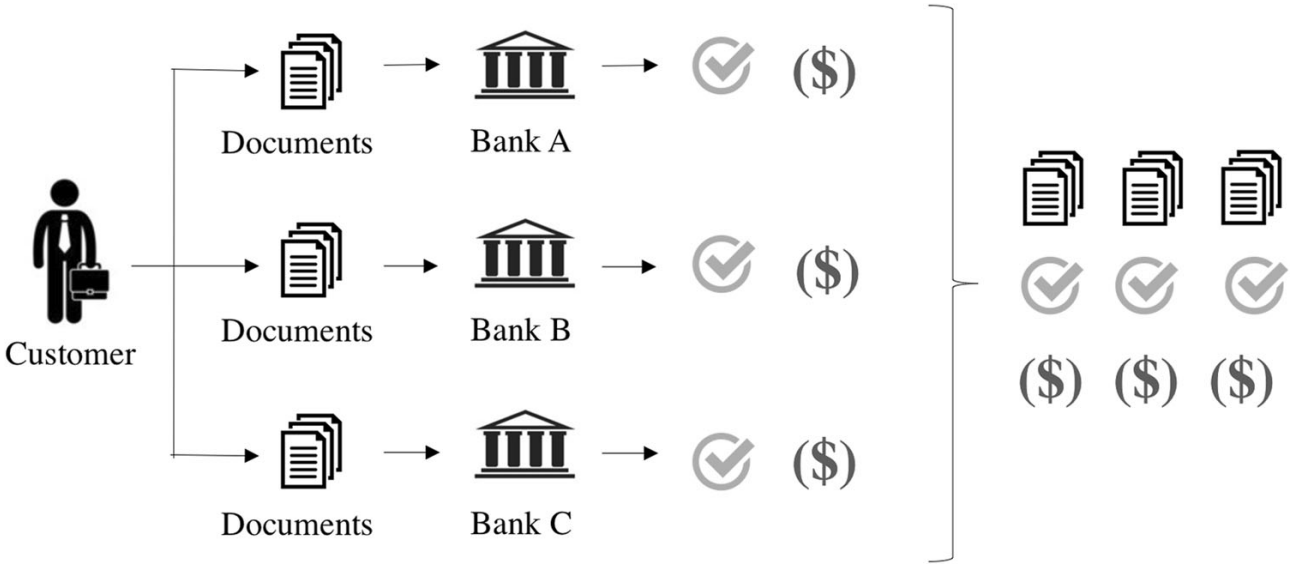
Κεφάλαιο 5

Μελλοντικές προεκτάσεις – Σύστημα διαμοιρασμού κόστους

Η διαδικασία κοινοποίησης των Know Your Customer δεδομένων που αναλύθηκε στην παρούσα διπλωματική διαθέτει ορισμένα χαρακτηριστικά τα οποία επιτρέπουν την περαιτέρω ανάπτυξη της και για άλλους σκοπούς. Συγκεκριμένα, το γεγονός ότι πελάτης και οργανισμοί είναι συμμετέχοντες του ίδιου κλειστού δικτύου blockchain ευνοεί την ανάπτυξη ροών εργασιών που επιτελούν και άλλες λειτουργίες, εκτός από την κοινοποίηση των KYC δεδομένων. Μία από αυτές, στην οποία θα αναφερθούμε στον συγκεκριμένο κεφάλαιο, και αποτελεί γόνιμο έδαφος για την ανάπτυξη εφαρμογής, είναι η διαδικασία διαμοιρασμού του κόστους επεξεργασίας, επαλήθευσης και έγκρισης των KYC δεδομένων που κοινοποιήθηκαν εκ του πελάτη. Στην συνέχεια θα αναπτυχθεί η προτεινόμενη αρχιτεκτονική για την συγκεκριμένη λειτουργία, η οποία βασίζεται στην ήδη αναπτυχθείσα εφαρμογή CordApp.
Για να έχει νόημα η προτεινόμενη εφαρμογή απαιτείται ένα ελάχιστο επίπεδο εμπιστοσύνης μεταξύ των εμπλεκόμενων χρηματοπιστωτικών οργανισμών. Αυτό περιλαμβάνει την εμπιστοσύνη ότι η διαδικασία επεξεργασίας και επαλήθευσης των KYC πληροφοριών του πελάτη από οποιονδήποτε οργανισμό είναι αξιόπιστη. Θα γίνει πλήρως κατανοητός ο λόγος αυτής της απαίτησης στην συνέχεια.

Το κόστος επαλήθευσης των πληροφοριών KYC είναι αρκετά υψηλό για μία τράπεζα [37]. Αυτό προκύπτει από την σύνθετη φύση των διαφορετικών ελέγχων που πρέπει να γίνουν προκειμένου να διαπιστωθεί η φερεγγυότητα του κάθε πελάτη. Η αποφυγή περιπτώσεων ξεπλύματος μαύρου χρήματος (anti-money laundering policies) κρίνεται απαραίτητη στο σύγχρονο χρηματοπιστωτικό σκηνικό. [39, p. 01046] Επομένως, λόγω του μεγάλου κόστους της διαδικασίας επαλήθευσης των πληροφοριών του πελάτη, προκύπτει η ανάγκη διαμοιρασμού του απαιτούμενου κόστους.

Πολύ συχνά, ένας πελάτης, ειδικά αν πρόκειται για νομική υπόσταση, χρειάζεται να κοινοποιήσει τα απαραίτητα δεδομένα για την διαδικασία KYC επαλήθευσης, σε παραπάνω από ένα χρηματοπιστωτικά ιδρύματα. Για παράδειγμα σε δύο τράπεζες, ένα fund και έναν ασφαλιστικό φορέα. Το κόστος για την επαλήθευση KYC το επωμίζεται ο χρηματοπιστωτικός οργανισμός. Σε περιπτώσεις όπως η παραπάνω, όπου εμπλέκονται πάνω από ένας οργανισμοί, το συνολικό κόστος πολλαπλασιάζεται.



Εικόνα 16 Η υφιστάμενη κατάσταση σχετικά με το κόστος επαλήθευσης των KYC δεδομένων [27]

Ακολούθως προτείνεται ένα σύστημα διαμοιρασμού του κόστους, σε περίπτωση που τόσο ο πελάτης όσο και οι οργανισμοί βρίσκονται στο ίδιο δίκτυο Corda και βασίζεται στην πρόταση του ***" KYC Optimization Using Distributed Ledger Technology"*** από τους Parra Moyano και Omri Ross (2017). [32]

Αρχικά ο πελάτης κοινοποιεί τα δεδομένα του σε έναν μόνο οργανισμό, όπως ακριβώς περιεγράφηκε στην παρούσα εργασία (βλ. Κεφάλαιο 4). Ο οργανισμός αφού λάβει τα δεδομένα, ξεκινά την διαδικασία επαλήθευσης τους. Η διαδικασία αυτή οδηγεί σε κάποιο πόρισμα, το οποίο και αντιπροσωπεύεται στο ledger του Corda ως ένα State, έστω **KYC_Results.** Στην συνέχεια, ο πελάτης δύναται να δημιουργήσει (Update Flow) νέο State προσθέτοντας οργανισμούς. Αυτοί οι επιπλέον οργανισμοί στους οποίους κοινοποιήθηκαν τα KYC δεδομένα, επιθυμούν με την σειρά τους το πόρισμα για την φερεγγυότητα του πελάτη (client).

Επομένως, αφού έρθουν σε επικοινωνία με τον αρχικό οργανισμό για το αρχικό κόστος της διαδικασίας επεξεργασίας και επαλήθευσης των δεδομένων, κάνουν ένα αίτημα κοινοποίησης σε αυτούς του πορίσματος. Για να είναι επιτυχημένη, όπως φαίνεται και στο **σχήμα,** η συναλλαγή πρέπει να υπάρξει η αντίστοιχη πληρωμή, η οποία γίνεται μέσω των Tokens (εδώ USD) που «αλλάζουν χέρια». Συγκεκριμένα στην συγκεκριμένη συναλλαγή:

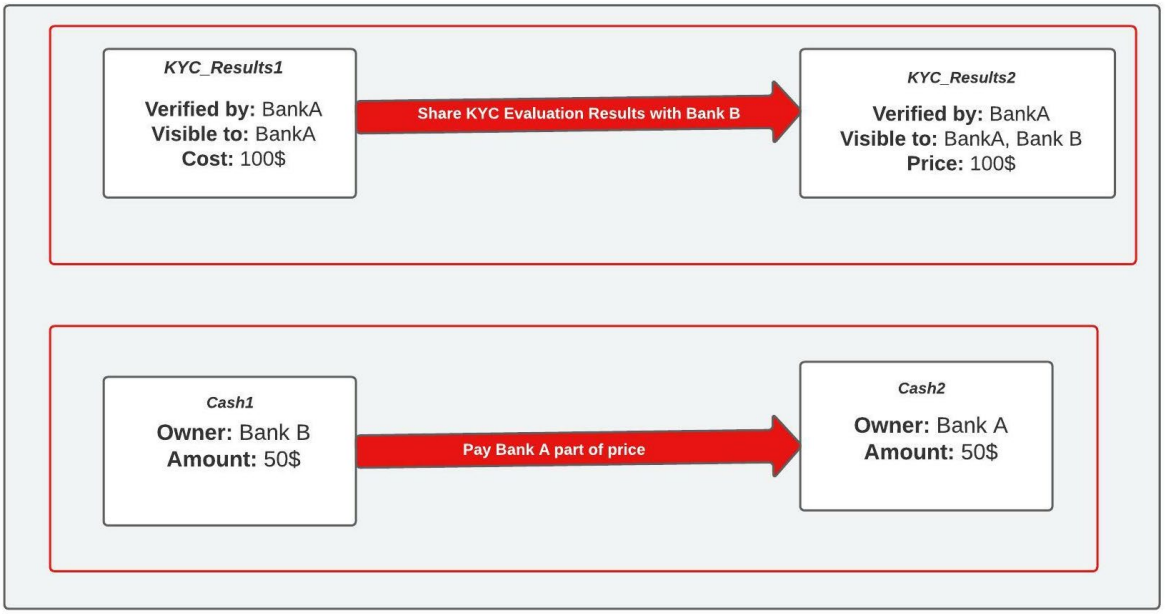
Inputs:

- KYC_Results States (with one institution as participants)
- TokenStateUSD (with Institution B as owner)

Outputs:

- KYC_Results (with two institutions as participants)
- TokenStateUSD (with Institution A as owner).

Στην ουσία, πραγματοποιείται μία συναλλαγή, η οποία προϋποθέτει, ώστε να είναι επιτυχής, την μεταβίβαση μονάδων αξίας (εδώ USD) από τον έναν οργανισμό στον άλλον. Ακολούθως φαίνεται το State το οποίο αντιπροσωπεύει το αποτέλεσμα της επαλήθευσης των KYC δεδομένων.



Διάγραμμα 27 Η συναλλαγή για τον διαμοιρασμού του κόστους μεταξύ του πρώτου και του δεύτερου οργανισμού

Προκειμένου να μπορεί να πραγματοποιηθεί η συναλλαγή, τίθενται ορισμένοι περιορισμοί από το Contract του Corda. Πρέπει ταυτόχρονα να γίνεται μεταφορά αξίας από τον οργανισμό που επιθυμεί να λάβει τα αποτελέσματα της KYC verification και να δημιουργηθεί ένα νέο KYC_Info με participants αμφότερους τους οργανισμούς (ενν. και τον πελάτη). Ο πιο σημαντικός περιορισμός που τίθεται σχετικά με την μεταφορά αξίας και το κόστος συνοψίζεται παρακάτω.

Ο προτεινόμενος μηχανισμός απαιτεί ότι όλοι οι χρηματοπιστωτικοί οργανισμοί που πρόκειται να λάβουν τα αποτελέσματα του KYC_verification πρέπει να συνεισφέρουν ισόποσα. Αυτό σημαίνει, σύμφωνα και με το προτεινόμενο σύστημα, ότι ο n-οστός κατά σειρά ο οποίος

επιθυμεί να συνεργαστεί με έναν πελάτη, πάντα πρέπει να συνεισφέρει κατά $\frac{m}{n}$ (δολάρια) όπου,
m : το συνολικό κόστος της διαδικασίας KYC
n : ο αύξων αριθμός του οργανισμού που θέλει να συνεργαστεί με τον πελάτη .

Είναι κατανοητό ότι όταν ο 2ος οργανισμός επιθυμεί να λάβει τα αποτελέσματα της KYC διαδικασίας επαλήθευσης, πρέπει να στείλει το ½ του κόστους στον 1ο οργανισμό, ο οποίος και πραγματοποίησε την επαλήθευση . Στην περίπτωση που n>2, δηλαδή στις περιπτώσεις όπου επιδιώκεται κοινοποίηση στον 3ο, τον 4ο κ.ο.κ. κατά σειρά οργανισμό, το κόστος που επωμίζεται ο νέος οργανισμός πρέπει να διαμοιραστεί ισόποσα στους n-1 ήδη συμμετέχοντες οργανισμούς.

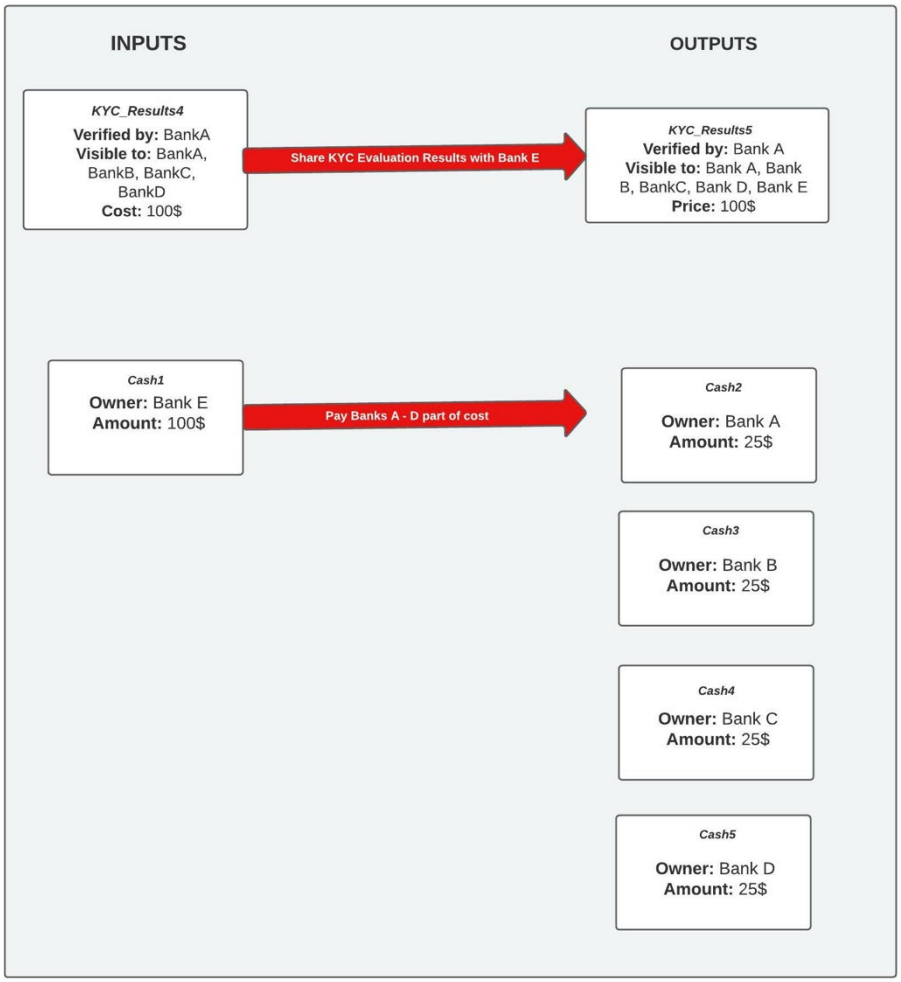
Ανά πάσα στιγμή, όλοι οι οργανισμοί που έχουν στην διάθεση τους τα αποτελέσματα της KYC διαδικασίας, έχουν συνεισφέρει ισόποσα. Την στιγμή που ο n-οστός οργανισμός επιθυμεί να γίνει participant του KYC_Results, το ποσό που πρέπει να δώσει, όπως αναφέρθηκε, είναι $\frac{m}{n}$.

Μέχρι την στιγμή εκείνη, οι n-1 οργανισμοί που έχουν ήδη στην διάθεση τους τα αποτελέσματα έχουν πληρώσει από $\frac{m}{(n-1)}$ δολάρια. Προκειμένου να γίνει ο διαμοιρασμός του κόστους, ο νέος οργανισμός πρέπει να πληρώσει $\frac{m}{n(n-1)}$ (1) σε κάθε οργανισμό που λαμβάνει το αποτέλεσμα.
Συνολικά, το κόστος είναι. $\sum_{m=1}^{n-1} \frac{m}{n(n-1)} = \frac{m}{n}$.
Το κόστος συνεπώς κάθε οργανισμού θα είναι $\frac{m}{(n-1)} - \frac{m}{n(n-1)} = \frac{m}{n}$.

Αρχιτεκτονική μεταφοράς αξίας

Έστω ότι το αποτέλεσμα της διαδικασίας KYC επαλήθευσης (την οποία έχει διενεργήσει ο institution A) έχουν κάποια δεδομένη χρονική στιγμή στην κατοχή τους ακόμη 3 οργανισμοί, δηλαδή συνολικά 4. Αυτό σημαίνει ότι την συγκεκριμένη χρονική στιγμή έχουν δαπανηθεί $\frac{m}{4}$ δολάρια από κάθε οργανισμό για να ληφθούν τα αποτελέσματα. Έστω ότι ακόμα ένας οργανισμός επιθυμεί τα αποτελέσματα της KYC επαλήθευσης. Εννοείται ότι τα δεδομένα έχουν κοινοποιηθεί και σε αυτόν τον οργανισμό με την συγκατάθεση του πελάτη (όπως αναφέρθηκε στις προηγούμενες ενότητες). Προκειμένου να αποκτήσει και ο 5ος οργανισμός πρόσβαση στα αποτελέσματα, πρέπει να πληρώσει ισόποσα τους υπόλοιπους 4 οργανισμούς με τέτοιο τρόπο ώστε, τελικά, και οι 5 οργανισμοί να έχουν συνεισφέρει με το ίδιο ποσό.

Το δολάριο, ως fiat currency, είναι fungible (ανταλλάξιμο). Αυτό σημαίνει ότι ένα asset σε δολάριο αξίας X δολαρίων μπορεί να διαχωριστεί σε επιμέρους αξίες υποπολλαπλάσιων του X. Επομένως το Cash State μπορεί να διαχωριστεί όπως φαίνεται παρακάτω.



Διάγραμμα 28 Η συναλλαγή (transaction) για τον διαμοιρασμό του κόστους όταν ο 5ος (Bank E) οργανισμός θέλει να αποκτήσει το αποτέλεσμα KYC_Results

Ως είσοδος της συναλλαγής που αποσκοπεί στην διαδικασία που αναφέρθηκε παραπάνω αποτελεί το KYC_Results State καθώς και το Cash State με Owner τον επιπλέον οργανισμό. Σύμφωνα με την παραπάνω σχέση (1) πρέπει σε καθένα από τους 4 υπόλοιπους οργανισμούς να προσφερθούν $\frac{m}{n(n-1)}$ δολάρια όπου m το κόστος της διαδικασίας KYC και n οι οργανισμοί (συμπεριλαμβανομένου του επιπλέον οργανισμού). Έστω ότι το συνολικό κόστος ήταν 500$. Σύμφωνα με την σχέση 1, εφόσον n = 5, πρέπει να δοθούν 25$ σε καθένα από τους 4 οργανισμούς. Προφανώς συνολικά το τίμημα θα ανέρχεται σε 100$, όσα ακριβώς θα έχουν καταβάλει όλοι οι οργανισμοί μετά το πέρας της συγκεκριμένη διαδικασίας.

Το Corda δεν επιτρέπει τον κατακερματισμό του. Input State. Επομένως, προκειμένου να γίνει ο διαμοιρασμός των χρημάτων που ο 5ος οργανισμός πρέπει να πληρώσει στους 4 προηγούμενους, αυτό που απαιτείται είναι η δημιουργία ακόμη 3 Cash States καθένα από τα οποία θα έχει ως Owner τους καθένα από τους 3 οργανισμούς (δεδομένου ότι η πρώτη State που αποτελεί εξέλιξη του Input State έχει ήδη δημιουργηθεί με 1 εκ των τεσσάρων οργανισμό ως Owner).

Βιβλιογραφία

[1] “http://info.cern.ch/Proposal.html.”

[2] G. Cormode and B. Krishnamurthy, “Key differences between Web 1.0 and Web 2.0,” *First Monday*, Apr. 2008, doi: 10.5210/fm.v13i6.2125.

[3] *W3C, (2010). Web 1.0. [image] Available at: http://www.w3.org/2010/Talks/0921-html5-plh/web10.html*.

[4] “DiNucci, Darcy (1999). ‘Fragmented Future’(PDF). Print. 53 (4): 32. Archived (PDF) from the original on 2011-11-10. Retrieved 2011-11-04.)”.

[5] “O’Reilly, Tim (2005-09-30). ‘What Is Web 2.0’. O’Reilly Network. Archived from the original on 2013-04-24. Retrieved 2006-08-06.)”.

[6] “Wood, L., Le Hors, A., Apparao, V., Byrne, S., Champion, M., Isaacs, S., Jacobs, I., Nicol, G., Robie, J., Sutor, R. and Wilson, C., 1998. Document object model (dom) level 1 specification. W3C recommendation, 1.”.

[7] O’Reily, “https://www.oreilly.com/radar/why-its-too-early-to-get-excited-about-web3/.”

[8] C. Lin *et al.*, “Artificial Intelligence–Assisted Electrocardiography for Early Diagnosis of Thyrotoxic Periodic Paralysis,” *J. Endocr. Soc.*, vol. 5, no. 9, p. bvab120, Sep. 2021, doi: 10.1210/jendso/bvab120.

[9] B. Carlsson and R. Gustavsson, “The Rise and Fall of Napster - An Evolutionary Approach,” in *Active Media Technology*, vol. 2252, J. Liu, P. C. Yuen, C. Li, J. Ng, and T. Ishida, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 347–354. doi: 10.1007/3-540-45336-9_40.

[10] G.Coulouris, J. Dollimore, T. Kindberg, G. Blair, Μετάφραση - Επιμέλεια Κωνσταντίνος Κοντογιάννης, *Κατανεμημένα Συστήματα Αρχές και Σχεδίαση*, 2nd ed. DA VINCI Μ.Ε.Π.Ε.

[11] D. Yaga, P. Mell, N. Roby, and K. Scarfone, “Blockchain technology overview,” National Institute of Standards and Technology, Gaithersburg, MD, NIST IR 8202, Oct. 2018. doi: 10.6028/NIST.IR.8202.

[12] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, “Blockchain,” *Bus. Inf. Syst. Eng.*, vol. 59, no. 3, pp. 183–187, Jun. 2017, doi: 10.1007/s12599-017-0467-3.

[13] “Swanson, Tim. ‘Consensus-as-a-service: a brief report on the emergence of

permissioned, distributed ledger systems.' Report, available online (2015)."

[14] "Nakamoto, S. and Bitcoin, A., 2008. A peer-to-peer electronic cash system. Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf."

[15] C. V. Helliar, L. Crawford, L. Rocca, C. Teodori, and M. Veneziani, "Permissionless and permissioned blockchain diffusion," *Int. J. Inf. Manag.*, vol. 54, p. 102136, Oct. 2020, doi: 10.1016/j.ijinfomgt.2020.102136.

[16] "Kulms, R., 2020. Blockchains: Private law matters. Sing. J. Legal Stud., p.63."

[17] "Hearn, M. and Brown, R.G., 2016. Corda: A distributed ledger. Corda Technical White Paper, 2016."

[18] J. Polge, J. Robert, and Y. Le Traon, "Permissioned blockchain frameworks in the industry: A comparison," *ICT Express*, vol. 7, no. 2, pp. 229–233, Jun. 2021, doi: 10.1016/j.icte.2020.09.002.

[19] "Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y. and Muralidharan, S., 2018, April. Hyperledger fabric: a distributed operating system for permissioned blockchains. In Proceedings of the thirteenth EuroSys conference (pp. 1-15).".

[20] F. Tjark, "Ethereum: Permission, Access, and Consensus? Better than Bitcoin?" (https://medium.com/blockchainspace/3-2-1-ethereum-in-detail-permission-access-and-consensus-234b7ee0f60e).

[21] "EEA MEMBERS :https://entethalliance.org/eea-members/."

[22] *https://consensys.net/quorum/*.

[23] H. Moniz, "The Istanbul BFT Consensus Algorithm," 2020, doi: 10.48550/ARXIV.2002.03613.

[24] "Aste, T., Tasca, P. and Di Matteo, T., 2017. Blockchain technologies: The foreseeable impact on society and industry. computer, 50(9), pp.18-28.".

[25] D. Magazzeni, P. McBurney, and W. Nash, "Validation and Verification of Smart Contracts: A Research Agenda," *Computer*, vol. 50, no. 9, pp. 50–57, 2017, doi: 10.1109/MC.2017.3571045.

[26] C. Khan, A. Lewis, E. Rutland, C. Wan, K. Rutter, and C. Thompson, "A Distributed-Ledger Consortium Model for Collaborative Innovation," *Computer*, vol. 50, no. 9, pp. 29–37, 2017, doi: 10.1109/MC.2017.3571057.

[27] “Thompson Reuters (2016) Know your customer (KYC) independent survey.”

[28] “(Benedict N.Nolens, at the MIT Technology Review Emtech conference, 2016.”

[29] “Shbair, W.; Steichen, M.; Fran ois, J. Blockchain orchestration and experimentation framework: A case study of KYC. In Proceedings of the First IEEE/IFIP International Workshop on Managing and Managed by Blockchain (Man2Block) colocated with IEEE/IFIP NOMS 2018, Jeju Island, Korea, 23–25 August 2018.”.

[30] “Norvill, R.; Steichen, M.; Shbair,W.M.; State, R. Blockchain for the Simplification and Automation of KYC Result Sharing. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea, 18–21 November 2019; pp. 9–10.”.

[31] N. Kapsoulis, A. Psychas, G. Palaiokrassas, A. Marinakis, A. Litke, and T. Varvarigou, “Know Your Customer (KYC) Implementation with Smart Contracts on a Privacy-Oriented Decentralized Architecture,” *Future Internet*, vol. 12, no. 2, p. 41, Feb. 2020, doi: 10.3390/fi12020041.

[32] J. Parra Moyano and O. Ross, “KYC Optimization Using Distributed Ledger Technology,” *Bus. Inf. Syst. Eng.*, vol. 59, no. 6, pp. 411–423, Dec. 2017, doi: 10.1007/s12599-017-0504-2.

[33] *https://medium.com/delta-exchange/centralized-vs-decentralized-vs-distributed-41d92d463868*.

[34] *R3 Corda Docs, docs.r3.com, Key Concepts*.

[35] C. D. Clack, V. A. Bakshi, and L. Braine, “Smart Contract Templates: foundations, design landscape and research directions,” 2016, doi: 10.48550/ARXIV.1608.00771.

[36] D. Mohanty, *R3 Corda for Architects and Developers: With Case Studies in Finance, Insurance, Healthcare, Travel, Telecom, and Agriculture*. Berkeley, CA: Apress, 2019. doi: 10.1007/978-1-4842-4529-3.

[37] *https://docs.r3.com/en/platform/corda/4.6/open-source/node-explorer.html*.

[38] R. F. Pol, “Anti-money laundering: The world’s least effective policy experiment? Together, we can fix it,” *Policy Des. Pract.*, vol. 3, no. 1, pp. 73–94, Jan. 2020, doi: 10.1080/25741292.2020.1725366.

[39] “https://www.legislation.gov.au/Details/F2016C01046.”