Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

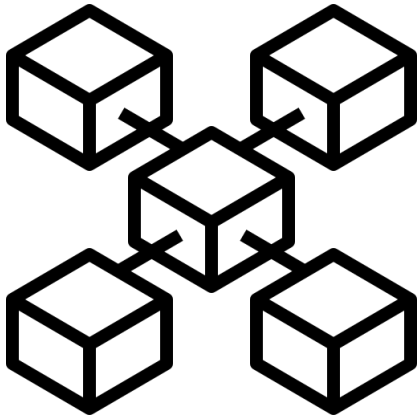
Αποκεντρωμένο υπολογιστικό νέφος: Μια προσέγγιση βασισμένη στο Blockchain

Μελέτη και υλοποίηση

Διπλωματικη Εργασια

του

ΓΡΗΓΟΡΑΤΟΥ Δ. ΣΠΥΡΙΔΩΝΟΣ



Επιβλέπων: Νεκτάριος Κοζύρης
Καθηγητής

Αθήνα, Οκτώβριος 2023

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Αποκεντρωμένο υπολογιστικό νέφος: Μια προσέγγιση βασισμένη στο Blockchain

Μελέτη και υλοποίηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΓΡΗΓΟΡΑΤΟΥ Δ. ΣΠΥΡΙΔΩΝΟΣ

Επιβλέπων: Νεκτάριος Κοζύρης
Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 32α Οκτωβρίου 2023.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Νεκτάριος Κοζύρης	Γεώργιος Γκούμας	Τσουμάκος Δημήτριος
Καθηγητής	Αναπληρωτής Καθηγητής	Αναπληρωτής Καθηγητής

Αθήνα, Οκτώβριος 2023

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....
Σπυρίδων Γρηγοράτος

32 Οκτωβρίου 2023

Περίληψη

TO BE ADDED

Λέξεις Κλειδιά

TO BE ADDED

Abstract

TO BE ADDED
TO BE ADDED

Keywords

TO BE ADDED

στους γονείς μου

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. Νεκτάριο Κοζύρη για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Υπολογιστικών Συστημάτων. Επίσης ευχαριστώ ιδιαίτερα την Δρ. Κατερίνα Δόκα για την καθοδήγησή της και την εξαιρετική συνεργασία που είχαμε. Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Αθήνα, Οκτώβριος 2023

Σπυρίδων Γρηγοράτος

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Στην σύγχρονη ψηφιακή εποχή, το υπολογιστικό νέφος (cloud computing) έχει αναδειχθεί σε ακρογωνιαίο λίθος της υποδομής πληροφοριακών συστημάτων, προσφέροντας κλιμακούμενους υπολογιστικούς πόρους κατά παραγγελία. Παραδοσιακά, οι υπηρεσίες υπολογιστικού νέφους κυριαρχούνται από τεχνολογικούς κολοσσούς, όπως οι Amazon (Amazon Web Services), IBM (IBM Cloud), Microsoft (Microsoft Azure) και Google (Google Cloud). Αυτά τα συγκεντρωτικά μοντέλα, αν και αποτελεσματικά, συχνά στερούνται διαφάνειας στην πολιτική τιμολόγησης, την κατανομή των πόρων και τις διαδικασίες λήψης των αποφάσεων. Επιπλέον, εισάγουν πιθανές μονοπωλιακές συμπεριφορές στην τιμολόγηση και την προσφορά υπηρεσιών.

Καθώς ο κόσμος κινείται προς ένα πιο αποκεντρωμένο πρότυπο σε διάφορους τομείς, από τα χρηματοοικονομικά έως την αλυσίδα εφοδιασμού, το πεδίο υπολογιστικού νέφους δεν αποτελεί εξαίρεση. Η υπόσχεση της αποκέντρωσής του προσφέρει την δυνατότητα μεγαλύτερης διαφάνειας, ενισχυμένης ασφάλειας και δικαιότερης κατανομής των πόρων. Εξαλείφοντας του μεσάζοντες και αξιοποιώντας τους εγγενείς μηχανισμούς εμπιστοσύνης της τεχνολογίας Blockchain, το αποκεντρωμένο υπολογιστικό νέφος μπορεί να εκδημοκρατίσει την πρόσβαση στην υπολογιστική ισχύ και να προωθήσει μια ανταγωνιστική αγορά.

1.1 Αντικείμενο της διπλωματικής

Η παρούσα εργασία παρουσιάζει μια νέα προσέγγιση για την αποκέντρωση των υπηρεσιών υπολογιστικού νέφους μέσω της ανάπτυξης μια αποκεντρωμένης εφαρμογής (decentralized App – dApp) στο Ethereum Blockchain. Το προτεινόμενο σύστημα επιτρέπει στους χρήστες να αναθέτουν υπολογιστικές εργασίες γραμμένες σε Java σε ένα αποκεντρωμένο δίκτυο παρόχων. Μέσω ενός μηχανισμού δημοπρασιών, οι πάροχοι των υπολογιστικών πόρων υποβάλλουν προσφορές για την ανάληψη εργασιών και οι πελάτες επιλέγουν τους παρόχους που επιθυμούν βάσει ενός συνδυασμού της προσφοράς που έχει κατατεθεί και του ιστορικού των επιδόσεων τους. Το σύστημα διασφαλίζει την ακεραιότητα της εκτέλεσης των εργασιών και προσφέρει μια ασφαλή διαδικασία πληρωμής μετά την επιτυχή ολοκλήρωσή τους.

Τα επόμενα κεφάλαια θα εμβαθύνουν στον σχεδιασμό, την αρχιτεκτονική και τις λεπτομέρειες υλοποίησης της dApp, αξιολογώντας τις δυνατότητές της να αναδιαμορφώσει το τοπίο του υπολογιστικού νέφους και να προσφέρει ένα πιο διαφανές, αξιόπιστο και αποτελεσματικό σύστημα για την ανάθεση υπολογιστικών εργασιών σε τρίτους.

1.2 Οργάνωση του τόμου

Η παρούσα εργασία είναι οργανωμένη σε επτά κεφάλαια. Στο κεφάλαιο 2 δίνεται το θεωρητικό υπόβαθρο του cloud computing και του Blockchain, με έμφαση στο Ethereum και τις λύσεις που προσφέρει. Στο κεφάλαιο 3 παρουσιάζεται η προεπισκόπηση, ο σχεδιασμός και αρχιτεκτονική του προτεινόμενου συστήματος. Στα κεφάλαια 4 και 5 αναλύονται οι επιμέρους μηχανισμοί των δημοπρασιών και εκτέλεσης των εργασιών αντίστοιχα, ενώ στο κεφάλαιο 6 περιγράφεται η υλοποίηση του συστήματος. Τέλος, στο κεφάλαιο 7 παρουσιάζονται τα συμπεράσματα, η σημασία τις εργασία, καθώς και οι μελλοντικές επεκτάσεις της.

Μέρος Ι

Θεωρητικό Μέρος

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό παρουσιάζονται η εξέλιξη του Cloud Computing, η τεχνολογία Blockchain, καθώς και οι αποκεντρωμένες εφαρμογές. Επίσης, γίνεται αναφορά σε προηγούμενες εργασίες που ασχολούνται με την ιδέα του αποκεντρωμένου υπολογιστικού νέφους.

2.1 Cloud Computing

2.1.1 Εξέλιξη Cloud Computing

Το υπολογιστικό νέφος αφορά την παροχή διαφόρων υπηρεσιών μέσω του διαδικτύου, συμπεριλαμβανομένων της αποθήκευσης και της υπολογιστικής ισχύος. Η έννοια, αν και έγινε ευρέως διαδεδομένη τα τελευταία χρόνια, έχει τις ρίζες της στην δεκαετία του 1960 με την έλευση του utility computing. Η ιδέα βασιζόταν στο εγχείρημα ότι οι υπολογιστικοί πόροι, όπως το νερό και το ηλεκτρικό ρεύμα, μπορούν να παρέχονται σε οικίες και επιχειρήσεις ως υπηρεσία κοινής ωφέλειας. Ωστόσο, μόλις την δεκαετία του 2000, με την άνοδο του διαδικτύου υψηλής ταχύτητας, τις σημαντικές εξελίξεις στις τεχνολογίες της εικονικοποίησης (virtualization) και την αύξηση της υπολογιστικής ισχύος, το cloud computing άρχισε να παίρνει την σύγχρονη μορφή του.

Εταιρείες όπως η Amazon, η Microsoft, η IBM και η Google, ήταν από τις πρώτες που αναγνώρισαν τις δυνατότητες ενοικίασης των τεράστιων υπολογιστικών πόρων τους [1] . Η Amazon Web Services (AWS) ξεκίνησε το 2006, σηματοδοτώντας την έναρξη της σύγχρονης εποχής του νέφους. Τα επόμενα χρόνια παρατηρήθηκε μια έκρηξη υπηρεσιών cloud computing [2], οι οποίες καλύπτουν διάφορες τεχνολογικές ανάγκες, από την υποδομή ως υπηρεσία (Infrastructure as a Service – IaaS) έως το λογισμικό ως υπηρεσία (Software as a Service – SaaS).

2.1.2 Υφιστάμενα συγκεντρωτικά μοντέλα και οι περιορισμού τους

Τα συγκεντρωτικά μοντέλα υπολογιστικού νέφους, προσφερόμενα από τεχνολογικούς κολοσσούς όπως το AWS, το Google Cloud και το Microsoft Azure κυριαρχούν στην τρέχουσα αγορά. Οι υπηρεσίες αυτές παρέχουν στους χρήστες αξιόπιστες, κλιμακούμενες και συχνά οικονομικά αποδοτικές λύσεις.

Ωστόσο, συνοδεύονται από εγγενείς περιορισμούς:

- Έλλειψη διαφάνειας: Ο τρόπος τιμολόγησης μπορεί να είναι πολύπλοκος και οι χρήστες συχνά δεν έχουν σαφή εικόνα για την κατανομή των πόρων [3, 4, 5] .

- Ενιαίο σημείο αποτυχίας (single point of failure): Τα συγκεντρωτικά συστήματα, εκ κατασκευής, έχουν πιθανά σημεία συμφόρησης. Εάν ένας μεγάλος πάροχος υπηρεσιών υπολογιστικού νέφους αντιμετωπίσει διακοπή λειτουργίας, αυτό μπορεί να επηρεάσει μεγάλο πλήθος χρηστών.

- Ανησυχίες σχετικά με το απόρρητο των δεδομένων: Με τα δεδομένα συγκεντρωμένα σε λίγες εταιρείες, υπάρχουν βάσιμες ανησυχίες σχετικά με την κατάχρηση των δεδομένων, την παρακολούθηση των χρηστών, καθώς και τις παραβιάσεις των υπαρχόντων συστημάτων.

- Δυνατότητα μονοπωλιακής συμπεριφοράς: Η κυριαρχία λίγων εταιρειών στην αγορά μπορεί να οδηγήσει σε μείωση του ανταγωνισμού και της καινοτομίας.

Παράλληλα με τους παραπάνω περιορισμούς, τα data centers που χρησιμοποιούνται για την παροχή των υπηρεσιών cloud computing καταναλώνουν μεγάλες ποσότητες ενέργειας, με αρνητικές επιπτώσεις στο περιβάλλον [6, 7].

2.1.3 Serverless Computing

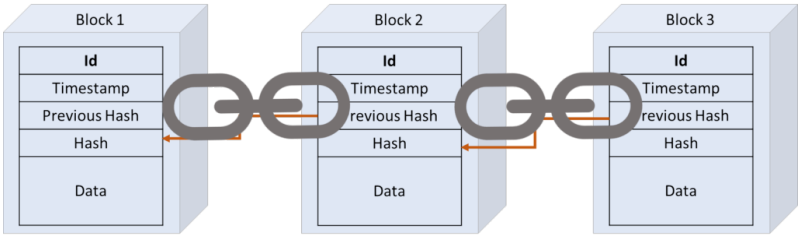
Με τον όρο serveless computing αναφερόμαστε σε ένα μοντέλο υπολογιστικού νέφους, στο οποίο οι προγραμματιστές δεν χρειάζεται να διαχειρίζονται οι ίδιοι τους πόρους και την υποδομή του server τους, επιτρέποντάς τους να επικεντρώνονται στον κώδικα του προγράμματός τους [8]. Αντί να υπάρχει μονίμως ενεργός ένας server, αυτός εκκινείται όταν ληφθεί ένα συγκεκριμένο event, εκτελεί την επιθυμητή συνάρτηση και, μόλις αυτή ολοκληρωθεί, τερματίζει την λειτουργία του. Το μοντέλο αυτό χαρακτηρίζεται από αυτόματη κλιμάκωση και μειωμένο κόστος εκτέλεσης, καθώς οι χρήστες χρεώνονται μόνο για την πραγματική τους χρήση, συνήθως με βάση τον αριθμό και τον χρόνο εκτέλεσης της συνάρτησής τους. Έτσι, η διαδικασία ανάπτυξης εφαρμογών επιταχύνεται, μειώνοντας ταυτόχρονα το κόστος και την λειτουργική πολυπλοκότητα [9, 10].

2.2 Blockchain

2.2.1 Εισαγωγή στην τεχνολογία Blockchain

Το 2008, μια οντότητα - άνθρωπος ή ομάδα ανθρώπων - με το ψευδώνυμο Satoshi Nakamoto παρουσίασε το Bitcoin, ένα αποκεντρωμένο ψηφιακό νόμισμα [11]. Πέρα από τη νομισματική του λειτουργία, το Bitcoin εισήγαγε την τεχνολογία του Blockchain. Το Blockchain είναι μια αλυσίδα από blocks, στα οποία αποθηκεύονται οι συναλλαγές που πραγματοποιούνται στο δίκτυο. Αποτελεί, δηλαδή, μια βάση δεδομένων συναλλαγών σε ένα δίκτυο, η οποία λειτουργεί ως αποκεντρωμένο λογιστικό βιβλίο. Προσφέρει διαφάνεια, ασφάλεια και απουσία κεντρικού ελέγχου. Στον πυρήνα του, το Blockchain, χρησιμοποιώντας κρυπτογραφικές αποδείξεις και έναν αλγόριθμο συναίνεσης (consensus algorithm) [12]

είναι ανθεκτικό στην τροποποίηση των δεδομένων του, διατηρώντας με τον τρόπο αυτό την ακεραιότητά τους, και εξασφαλίζοντας την εμπιστοσύνη μεταξύ των συμμετεχόντων.



Εικόνα 2.1: *Γραφική απεικόνιση του Blockchain* [13]

2.2.2 Ethereum

Το Ethereum, το οποίο παρουσιάστηκε το 2013 από τον Vitalik Buterin και δημοσιεύθηκε το 2015, επέκτεινε την ιδέα του Blockchain και δημιούργησε μια δημόσια,ανοικτού κώδικα, πλατφόρμα κατανεμημένου υπολογισμού, η οποία διαθέτει την λειτουργικότητα των έξυπνων συμβολαίων [14, 15]. Με τον τρόπο αυτό, παρέχει στους προγραμματιστές την Ethereum Virtual Machine (EVM), μια αποκεντρωμένη εικονική μηχανή που είναι Turing complete, μετατρέποντας τη δημιουργία εφαρμογών στο Blockchain σε πολύ απλούστερη και εύκολη διαδικασία [16].

2.2.3 Έξυπνα συμβόλαια και Αποκεντρωμένες εφαρμογές

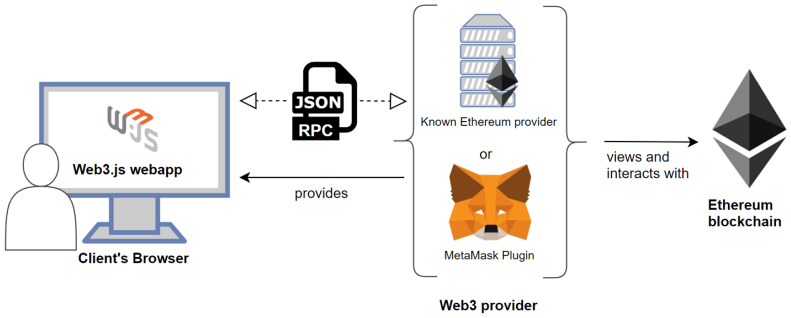
Μία από τις βασικές καινοτομίες του Ethereum είναι το έξυπνο συμβόλαιο (smart contract). Τα έξυπνα συμβόλαια είναι ντετερμινιστικά αυτοεκτελούμενα συμβόλαια όπου οι όροι γράφονται απευθείας σε κώδικα και διανέμονται στο Blockchain [16, 17]. Εκτελούν αυτόματα αξιόπιστες συναλλαγές, χωρίς την ανάγκη διαμεσολάβησης τρίτου, με τρόπο ανιχνεύσιμο και μη αναστρέψιμο. Η προσθήκη των έξυπνων συμβολαίων επεκτείνει την χρήσης της τεχνολογίας Blockchain πέρα από τις χρηματοοικονομικές συναλλαγές - όπως αυτές στις οποίες απευθύνεται το Bitcoin - σε οποιονδήποτε τομέα η εμπιστοσύνη είναι απαραίτητη, όπως τα συστήματα ψηφοφορίας και οι εφαρμογές Internet of Things (IoT) [18].

Οι αποκεντρωμένες εφαρμογές, ή dApps [19, 20], είναι ένα άμεσο προϊόν της λειτουργικότητας των έξυπνων συμβολαίων, που εκτελούνται σε ένα δίκτυο Blockchain με ομότιμο τρόπο. Αυτές οι εφαρμογές δεν απαιτούν κεντρική αρχή, είναι ανοικτού κώδικα και δίνουν κίνητρα στους χρήστες μέσω κρυπτογραφικών tokens. Αξιοποιούν τα οφέλη της τεχνολογίας Blockchain για να διασφαλίσουν ότι καμία μεμονωμένη οντότητα δεν έχει τον έλεγχο της εφαρμογής, προσφέροντας ένα νέο επίπεδο ασφάλειας και εμπιστοσύνης για τους χρήστες.

2.2.4 Web 3.0

Με βάση τα dApps, το Web3 προτάθηκε το 2014 από τον συνιδρυτή του Ethereum, Gavin Wood, ως ένα νέο μοντέλο χρήσης του διαδικτύου. Αναγνωρίζοντας ότι το παρόν μοντέλο του

διαδικτύου (Web 2.0) προϋποθέτει την εμπιστοσύνη των χρηστών προς κεντρικούς παρόχους, προτείνεται η χρήση του Blockchain για την κατασκευή μιας νέας, αποκεντρωμένης έκδοσής του, στην οποία τον έλεγχο των δεδομένων κατέχουν οι ίδιοι οι χρήστες [21].



Εικόνα 2.2: *Αρχιτεκτονική του Web3*
[16]

2.2.5 Ether και Gas Fees

Το token του Ethereum, το Ether (ETH), εξυπηρετεί δύο βασικούς σκοπούς: την αποζημίωση των κόμβων του δικτύου για τους υπολογισμούς που εκτελούν και τη διαπραγμάτευσή του ως ψηφιακό νόμισμα σε διάφορα ανταλλακτήρια κρυπτονομισμάτων [22]. Στο δίκτυο του Ethereum, οι αμοιβές των συναλλαγών μετρώνται με βάση την υπολογιστική πολυπλοκότητα, τη χρήση εύρους ζώνης και τις ανάγκες αποθήκευσης, οι οποίες υπολογίζονται σε όρους gas και πληρώνονται σε ETH. Αυτό διασφαλίζει ότι κακόβουλα προγράμματα ή αναποτελεσματικός κώδικας δεν φράσσουν το δίκτυο [23].

Για να διευκολύνει τις συναλλαγές και τους υπολογισμούς, το Ethereum χρησιμοποιεί και μονάδες μικρότερης αξίας. Η μικρότερη μονάδα του Ether είναι γνωστή ως wei. Ένα Ether ισοδυναμεί με ένα πεντάκις εκατομμύριο wei ($1 wei = 10^{-18} eth$). Η αμέσως επόμενη μονάδα που χρησιμοποιείται ονομάζεται gwei (Gigawei) και ισοδυναμεί με ένα δισεκατομμύριο wei ή 0,000000001 Ether.

Αυτές οι μικρότερες μονάδες Ether είναι σημαντικές για την ακρίβεια στις συναλλαγές, ειδικά όταν πρόκειται για χρεώσεις gas, καθώς επιτρέπουν στους χρήστες να καθορίζουν το ακριβές ποσό που είναι διατεθειμένοι να πληρώσουν ανά μονάδα gas, χωρίς να χρειάζεται να χρησιμοποιούν εξαιρετικά μικρής αξίας δεκαδικά ψηφία. Αυτό το σύστημα όχι μόνο παρέχει μια πιο κατανοητή κλίμακα για τους χρήστες, αλλά επιτρέπει επίσης στο δίκτυο Ethereum να χειρίζεται τις συναλλαγές και τις αλληλεπιδράσεις έξυπνων συμβολαίων με ακρίβεια, εξασφαλίζοντας δικαιοσύνη για όλα τα εμπλεκόμενα μέρη.

2.2.6 Ethereum 2.0

Αρχικά, το Ethereum Blockchain χρησιμοποιούσε ως αλγόριθμο συναίνεσης (consensus algorithm) το Proof of Work (PoW) που χρησιμοποιεί και το Bitcoin. Το γεγονός αυτό εισήγαγε περιορισμούς στην επεκτασιμότητα του δικτύου, την ενεργειακή αποδοτικότητά του, καθώς και υψηλές χρεώσεις στις συναλλαγές. Στις 15 Σεπτεμβρίου 2022, ολοκληρώθηκε

η μετάβαση στο Blockchain Ethereum 2.0, το οποίο χρησιμοποιεί για consensus algorithm το Proof of Stake (PoS), μειώνοντας έτσι την ενεργειακή κατανάλωση κατά 99% [24].

2.2.7 Layer 2

Οι τρεις επιθυμητές ιδιότητες της τεχνολογίας Blockchain είναι η αποκέντρωση, η ασφάλεια και η επεκτασιμότητα. Ωστόσο, σύμφωνα με το Blockchain Trilemma [25], μια απλή αρχιτεκτονική του μπορεί να πετύχει μόνο δύο από τις τρεις. Έτσι, προς επίτευξη ενός ασφαλούς και αποκεντρωμένου δικτύου, πρέπει να θυσιαστεί η επεκτασιμότητα.

Καθώς το Ethereum επεξεργάζεται σήμερα περισσότερες από 1 εκατομμύριο συναλλαγές την ημέρα, με δυνατότητα επεξεργασίας 15 συναλλαγές το δευτερόλεπτο, η συνεχώς αυξανόμενη ζήτηση για χρήση του μπορεί να προκαλέσει υψηλές χρεώσεις στις συναλλαγές. Στο σημείο αυτό εισάγονται τα δίκτυα επιπέδου 2 (layer 2), τα οποία, ενώ χρησιμοποιούν το δίκτυο του επιπέδου 1 (mainnet Ethereum), παρέχουν τρόπους για επεξεργασία των συναλλαγών εκτός αυτού (off-chain computations). Έτσι, αφαιρούν το βάρος επιβεβαίωσης κάθε συναλλαγής από το επίπεδο 1, επιτρέποντάς του να γίνει λιγότερο συμφορημένο και όλα να γίνονται πιο κλιμακούμενα, χωρίς να θυσιάζεται η αποκέντρωση και η ασφάλεια [26].

2.3 Προηγούμενες εργασίες σχετικά με το αποκεντρωμένο cloud computing και τις dApps

Οι περιορισμοί των συγκεντρωτικών μοντέλων υπολογιστικού νέφους και οι δυνατότητες της τεχνολογίας Blockchain, οδήγησαν ερευνητές και προγραμματιστές να εξερευνήσουν το αποκεντρωμένο υπολογιστικό νέφος [27, 28, 29, 30, 31, 32]. Έργα τα οποία έχουν αποτολμήσει το εγχείρημα αυτό, με στόχο την δημιουργία μιας αποκεντρωμένης αγοράς υπολογιστικής ισχύος και κατανεμημένου αποθηκευτικού χώρου είναι:

- Το SONM: Μια αποκεντρωμένη πλατφόρμα που προσφέρει σε χρήστες την ευκαιρία να νοικιάσουν τους αδρανείς υπολογιστικούς πόρους τους, δημιουργώντας ένα δίκτυο ομότιμων κόμβων. Ο κεντρικός στόχος της πλατφόρμας είναι η παροχή υποδομής (IaaS) κυρίως για εφαρμογές μηχανικής μάθησης και video rendering, τα οποία απαιτούν μεγάλη υπολογιστική ισχύ [33].

- Το Golem: Μια πλατφόρμα που έχει σκοπό την δημιουργία ενός αποκεντρωμένου υπερυπολογιστή. Απευθύνεται κυρίως σε χρήστες με απαιτητικές εργασίες, όπως η βαθιά μάθηση και η επεξεργασία γραφικών [34].

- Το iExec: Μια αποκεντρωμένη πλατφόρμα βασισμένη στο Ethereum, η οποία χρησιμοποιεί του αδρανείς πόρους του υπολογιστή για την επεξεργασία δεδομένων. Για την επιλογή των παρόχων που θα εκτελέσουν την εργασία χρησιμοποιείται task scheduler [35].

- Το Filecoin: Μια αποκεντρωμένη πλατφόρμα και πρωτόκολλο για την αποθήκευση δεδομένων, η οποία προσφέρει την Filecoin Virtual Machine για την ανάπτυξη εφαρμογών στο δίκτυό της [36].

- Το CloudAgora: Μια πλατφόρμα που στοχεύει να σπάσει το μονοπώλιο των παραδοσιακών παρόχων υπολογιστικού νέφους. Αξιοποιεί την τεχνολογία Blockchain για να προσφέρει μια αποκεντρωμένη αγορά υπολογιστικών πόρων και αποθήκευσης, επιτρέποντας σε οποιονδήποτε δυνητικό πάροχο πόρων να χρησιμοποιεί τους αδρανείς πόρους του και να ανταγωνίζεται με τους υπόλοιπους με ίσους όρους. Στην περίπτωση της εκτέλεσης εργασίας με υπολογιστικούς πόρους, για την επαλήθευση της ορθής εκτέλεσής της, χρησιμοποιεί το TrueBit Protocol [37] το οποίο αποτελεί μια σύνθετη επιλογή που προσθέτει καθυστέρηση στην εκτέλεση των εργασιών και αυξάνει το κόστος των συναλλαγών. Για την εκτέλεση των εργασιών, επιλέγεται πάντοτε ο πάροχος που θα έχει υποβάλει την καλύτερη προσφορά [38, 39].

- Το ChainFaas: Μια πλατφόρμα που στοχεύει στην αξιοποίηση της υπολογιστικής ικανότητας των προσωπικών υπολογιστών για εργασίες υπολογιστικής ισχύος. Δεν ελέγχει την ορθή εκτέλεση της εργασίας από τον πάροχο, αλλά χρησιμοποιεί ιδιωτικό (private) Blockchain για την σύνδεση παρόχων και πελατών. Η επιλογή του παρόχου γίνεται με task scheduler [40, 41].

Μέρος II

Πρακτικό Μέρος

Κεφάλαιο 3

Σχεδιασμός και αρχιτεκτονική του συστήματος

Στο κεφάλαιο αυτό περιγράφεται η αρχιτεκτονική της dApp και τα σενάρια εκτέλεσης που καλύπτει. Αρχικά, παρουσιάζεται μια επισκόπηση της λειτουργίας της εφαρμογής και των συμμετεχόντων σε αυτήν. Στην συνέχεια, αναλύονται η αρχιτεκτονική της εφαρμογής τα σενάρια εκτέλεσης που καλύπτει και οι διαδικασίες που ακολουθούνται σε κάθε ένα από αυτά. Τέλος, παρουσιάζονται τα μέτρα ασφαλείας που έχουν ληφθεί για την την διασφάλιση της αξιοπιστίας της.

3.1 Επισκόπηση της προτεινόμενης dApp

Στο πλαίσιο του αποκεντρωμένου υπολογιστικού νέφους, αναδύονται δύο κρίσιμες προκλήσεις: η διασφάλιση της ειλικρινούς εκτέλεσης των εργασιών και η προστασία του περιβάλλοντος του παρόχου. Οι παραδοσιακές μέθοδοι συχνά περιλαμβάνουν δυσκίνητες ή δαπανηρές διαδικασίες επαλήθευσης, οι οποίες μπορεί να αποτελέσουν αποτρεπτικό παράγοντα για πολλούς χρήστες. Επιπλέον η εκτέλεση άγνωστου κώδικα στο μηχάνημα ενός παρόχου ενέχει σημαντικούς κινδύνους ασφαλείας.

Αντιμετωπίζοντας αυτές τις προκλήσεις, η dApp εισάγει δύο απλές λύσεις:

- Μηχανισμός επαλήθευσης: Αντί να βασίζεται σε βαριές και δαπανηρές υπολογιστικές αποδείξεις, η dApp χρησιμοποιεί μια προκαθορισμένη Java κλάση. Οι πελάτες παρέχουν τον κώδικά τους ως μεταγλωττισμένη Java κλάση, διασφαλίζοντας με τον τρόπο αυτό ότι οι πάροχοι δεν μπορούν να έχουν άμεση πρόσβαση στον κώδικα ή να τροποποιήσουν την λογική της υλοποίησής του. Η κλάση αυτή, με την ονομασία *Code*, πρέπει να περιέχει δύο βασικές public μεθόδους: την *getComputation*, η οποία περιέχει την λογική της εκτέλεσης της υπολογιστικής εργασίας και την *getVerification*, η οποία επιστρέφει μια συμβολοσειρά που έχει οριστεί από τον πελάτη. Αυτή η συμβολοσειρά επαλήθευσης λειτουργεί ως απόδειξη της πραγματικής εκτέλεσης της εργασίας, διασφαλίζοντας ότι οι πάροχοι έχουν όντως εκτελέσει τους υπολογισμούς του πελάτη.

- Εκτέλεση εργασίας σε Docker Container: Για να διασφαλιστεί η ασφάλεια του μηχανήματος του παρόχου και να διατηρηθεί η ακεραιότητα της διαδικασίας εκτέλεσης, οι εργασίες εκτελούνται εντός ενός docker container. Αυτό το containerized περιβάλλον απομονώνει την εκτέλεση από το πρωτεύον σύστημα του παρόχου, αποτρέποντας την πρόκληση βλάβης από πιθανό κακόβουλο κώδικα. Επιπλέον, εξασφαλίζει ένα συνεπές

περιβάλλον εκτέλεσης και αποτρέπει κάθε εξωτερική παρέμβαση, διασφαλίζοντας ότι η εργασία εκτελείται όπως προβλέπεται από τον πελάτη

Με αυτές τις απλές και οικονομικά αποδοτικές λύσεις, η dApp στοχεύει στον εκδημοκρατισμό και την ενίσχυση της προσβασιμότητας του υπολογιστικού νέφους, δρώντας ως γέφυρα μεταξύ των πελατών που έχουν υπολογιστικές εργασίες και των παρόχων που κατέχουν τους πόρους για την εκτέλεσή τους. Εξασφαλίζοντας την διαφανή και επαληθεύσιμη εκτέλεση των εργασιών, η dApp προσφέρει και αξιοπιστία στις διαδικασίες εκτέλεσης και πληρωμής.

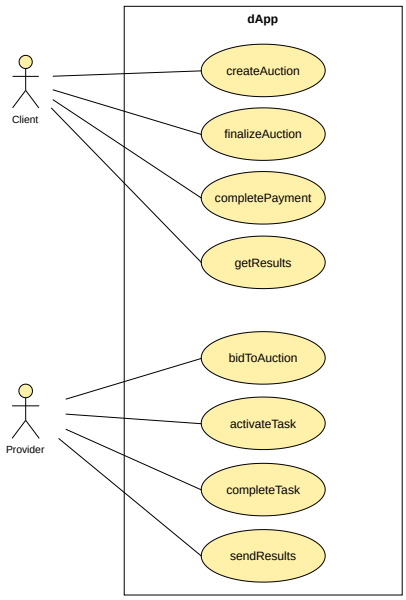
Για να κατανοηθεί καλύτερα ο τρόπος λειτουργίας του αποκεντρωμένου συστήματος, παρακάτω προσδιορίζονται οι κύριοι συμμετέχοντες σε αυτό.

3.2 Περιγραφή των συμμετεχόντων

Οι οντότητες που συμμετέχουν στην εφαρμογή είναι οι ακόλουθες:

- Πελάτης (client): Οντότητα που απαιτεί την εκτέλεση υπολογιστικών εργασιών. Παρέχει την εργασία με την μορφή μεταγλωττισμένης κλάσης Java και ξεκινά την διαδικασία δημοπρασίας για την εύρεση κατάλληλου παρόχου. Στην συνέχεια επιλέγει τον πάροχο που επιθυμεί από αυτούς που έχουν υποβάλλει προσφορά και, όταν η εκτέλεση της εργασίας του ολοκληρωθεί, πραγματοποιεί την πληρωμή και λαμβάνει τα αποτελέσματα των υπολογισμών του.

- Πάροχος (provider): Οντότητα με υπολογιστικούς πόρους που επιθυμεί να εκτελέσει εργασίες για τους πελάτες. Υποβάλλουν προσφορές για εργασίες σε δημοπρασία και, αφού επιλεγούν, εκτελούν τις εργασίες σε ασφαλές περιβάλλον.



Εικόνα 3.1: *Use case diagram της dApp*

Με σαφή κατανόηση των ρόλων των πελατών και των παρόχων, παρακάτω αναλύεται η τεχνική αρχιτεκτονική που διευκολύνει τις αλληλεπιδράσεις τους και διασφαλίζει την σταθερότητα του συστήματος.

3.3 Αρχιτεκτονική του συστήματος

Η αρχιτεκτονική της dApp έχει σχεδιαστεί για να διευκολύνει την απρόσκοπτη αλληλεπίδραση μεταξύ πελατών και παρόχων, διασφαλίζοντας παράλληλα την ακεραιότητα και την ασφάλεια των υπολογιστικών εργασιών. Ακολουθεί μια ανάλυσή της:

1. Smart Contracts:

   - AuctionsManager: Έξυπνο συμβόλαιο, το οποίο διαχειρίζεται τη διαδικασία δημοπρασίας, από την έναρξή της από τον πελάτη έως την επιλογή ενός παρόχου με βάση τις προσφορές. Χειρίζεται τις προθεσμίες και άλλες παραμέτρους που αφορούν τη δημοπρασία.
   - TasksManager: Έξυπνο συμβόλαιο, το οποίο επιβλέπει τον κύκλο ζωής μιας υπολογιστικής εργασίας. Εξασφαλίζει τις διαδικασίες σωστής εκτέλεσης, επαλήθευσης και πληρωμής και διαχειρίζεται επίσης τις χρηματικές εγγυήσεις και τις προθεσμίες που σχετίζονται με κάθε εργασία.

2. Ενσωμάτωση IPFS:

   - Οι πελάτες μεταφορτώνουν τη μεταγλωττισμένη κλάση Java (Code.class) στο IPFS (InterPlanetary FileSystem) και παρέχουν το CID (Content Identifier) κατά τη δημιουργία της δημοπρασίας. Αυτό διασφαλίζει ότι ο κώδικας παραμένει αμετάβλητος και προσβάσιμος στον επιλεγμένο πάροχο.
   - Οι τυποποιημένες boilerplate κλάσεις (Main.class και Time.class) αποθηκεύονται επίσης στο IPFS και ανακτώνται χρησιμοποιώντας προκαθορισμένα CID.

3. Docker Container:

   - Το Docker container χρησιμεύει ως περιβάλλον εκτέλεσης των υπολογιστικών εργασιών. Αποτελείται από δύο πρωταρχικά images:
     - Java Image: Εκτελεί την Main.class, η οποία ενορχηστρώνει ολόκληρη τη διαδικασία, από την εκτέλεση της υπολογιστικής εργασίας του πελάτη έως τον υπολογισμό διάρκειάς της και τέλος την εκτύπωση των αποτελεσμάτων.
     - Node Image: Μια ειδική ελαφριά έκδοση της dApp εκτελείται μέσα σε αυτό το docker image. Αναλύει τα αποτελέσματα από την εκτέλεση της Java class και τα στέλνει στο έξυπνο συμβόλαιο TasksManager. Επίσης αποθηκεύει το αποτέλεσμα της εκτέλεσης της εργασίας σε ένα αρχείο στο μηχάνημα του παρόχου ώστε στην συνέχεια να σταλεί στο IPFS από την dApp.

4. Γραφική διεπαφή:

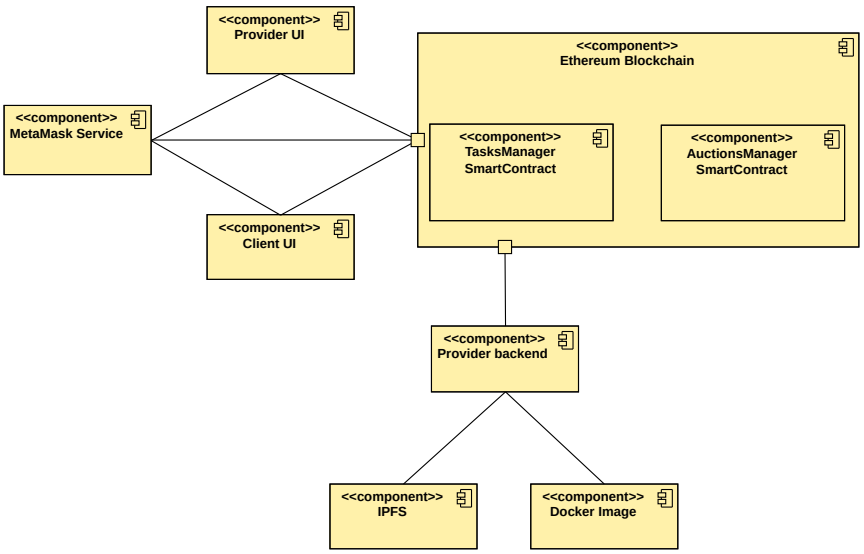
- Κατασκευασμένη με τη χρήση του React Framework, η διεπαφή της dApp διευκολύνει τις αλληλεπιδράσεις του χρήστη με το σύστημα. Επιτρέπει στους πελάτες να ξεκινούν δημοπρασίες, στους παρόχους να υποβάλλουν προσφορές και στα δύο μέρη να διαχειρίζονται και να παρακολουθούν τις εργασίες που συμμετέχουν. Η διεπαφή χειρίζεται επίσης τον κατακερματισμό της συμβολοσειράς επαλήθευσης και αλληλεπιδρά με το Ethereum blockchain μέσω events που στέλνονται από τα smart contracts.

5. Μηχανισμός επαλήθευσης:

   - Όπως αναλύθηκε προηγουμένως, ο μηχανισμός επαλήθευσης διασφαλίζει ότι οι πάροχοι εκτελούν πραγματικά την υπολογιστική εργασία του πελάτη. Η μέθοδος getVerification στην Code.class επιστρέφει μια συμβολοσειρά που ορίζεται από τον πελάτη, η οποία, όταν κατακερματιστεί και συγκριθεί με τον αποθηκευμένο κατακερματισμό που δηλώθηκε από τον πελάτη στο συμβόλαιο AuctionsManager, επαληθεύει την ορθή εκτέλεση της εργασίας.

6. Σύστημα πληρωμών και εγγυήσεων:

   - Τόσο οι πελάτες όσο και οι πάροχοι τοποθετούν χρηματικές εγγυήσεις για να διασφαλίσουν τη δέσμευσή τους στην εργασία. Μετά την επιτυχή ολοκλήρωση και επαλήθευση της εργασίας, ο πελάτης πληρώνει τον πάροχο με βάση το συμφωνημένο ποσο (wei ανά δευτερόλεπτο εκτέλεσης) και την διάρκεια της εκτέλεσης. Τα προαναφερθέντα έξυπνα συμβόλαια χειρίζονται αυτές τις συναλλαγές, εξασφαλίζοντας διαφάνεια και ασφάλεια.



Εικόνα 3.2: *Component diagram της dApp*

Με την ενσωμάτωση αυτών των στοιχείων, η αρχιτεκτονική της dApp εξασφαλίζει ένα στιβαρό, ασφαλές και αποτελεσματικό σύστημα για αποκεντρωμένο υπολογιστικό νέφος.

Ο σχεδιασμός δίνει προτεραιότητα στην αξιοπιστία, διασφαλίζοντας ότι τόσο οι πελάτες όσο και οι πάροχοι μπορούν να συμμετέχουν σε υπολογιστικές εργασίες με εμπιστοσύνη.

Η αρχιτεκτονική παρέχει ένα σχέδιο του συστήματος, δίνοντας έμφαση στα συστατικά του και στις αλληλεπιδράσεις τους.

Παρακάτω παρουσιάζεται η συμπεριφορά του συστήματος τόσο στο πρωταρχικό σενάριο εκτέλεσής του, όσο και στα εναλλακτικά μη ιδανικά σενάρια.

3.4 Ροές εργασιών του συστήματος

3.4.1 Επιτυχής ολοκλήρωση εργασίας

Τα βήματα που ακολουθούνται έως την επιτυχή ολοκλήρωση μιας εργασίας είναι τα ακόλουθα:

1. Δημιουργία εργασιών: Ο πελάτης γράφει και μεταγλωττίζει την κλάση Java με το όνομα *Code*. Η κλάση αυτή περιέχει δύο public μεθόδους: την *getComputation* που περιέχει τη λογική της εργασίας και την *getVerification* που επιστρέφει μια προκαθορισμένη συμβολοσειρά για επαλήθευση.

2. IPFS Upload: Ο πελάτης μεταφορτώνει την μεταγλωττισμένη κλάση του στο IPFS (InterPlanetary FileSystem) και λαμβάνει ένα CID (Content Identifier).

3. Έναρξη δημοπρασίας: Χρησιμοποιώντας την dApp, ο πελάτης ξεκινά μια δημοπρασία, παρέχοντας τις απαραίτητες λεπτομέρειες, όπως η προθεσμία της δημοπρασίας, η προθεσμία εκτέλεσης της εργασίας, το CID της Java κλάσης και την συμβολοσειρά επαλήθευσης.

4. Υποβολή προσφοράς: Οι πάροχοι υποβάλλουν προσφορά για την εργασία καθορίζοντας την τιμή (σε wei ανά δευτερόλεπτο εκτέλεσης) που προτείνουν.

5. Επιλογή παρόχου: Ο πελάτης επιλέγει τον πάροχο που επιθυμεί να εκτελέσει την εργασία του, βασιζόμενος στην τιμή προσφοράς και στην βαθμολογία που έχει προκύψει από προηγούμενες επιδόσεις του παρόχου. Στο βήμα αυτό, ο πελάτης στέλνει ως εγγύηση το ποσό που αντιστοιχεί σε 2 δευτερόλεπτα εκτέλεσης της εργασίας.

6. Ενεργοποίηση της εργασίας: Ο πάροχος ενεργοποιεί την εργασία στέλνοντας την δική του εγγύηση, η οποία αντιστοιχεί σε 10 δευτερόλεπτα εκτέλεσης της εργασίας. Στην συνέχεια, το dApp ενορχηστρώνει την διαδικασία εκτέλεσης της εργασίας, λαμβάνοντας από το IPFS τις απαραίτητες κλάσεις, δημιουργώντας την εικόνα *Docker* και εκκινώντας τον docker container.

7. Εκτέλεση εργασίας: Εντός του docker container, εκτελείται η κλάση Java. Η συμβολοσειρά επαλήθευσης, μαζί με την διάρκεια εκτέλεσης και τον χρόνο ολοκλήρωσης της εργασίας στέλνονται στο smart contract *TasksManager* για τον έλεγχο της διαδικασίας. Τα αποτελέσματα του υπολογισμού εγγράφονται σε ένα αρχείου κειμένου. Το συμβόλαιο επαληθεύει την ορθή εκτέλεση της εργασίας με βάση την παρεχόμενη

συμβολοσειρά επαλήθευσης και διασφαλίζει ότι ολοκληρώθηκε εντός της συμφωνημένης προθεσμίας. Στην περίπτωση ορθής και έγκαιρης εκτέλεσης, η βαθμολογία του παρόχου αυξάνεται και οι χρηματικές εγγυήσεις του επιστρέφονται.

8. Υποβολή αποτελέσματος: Το dApp μεταφορτώνει εκ μέρους του παρόχου το αποτέλεσμα της εργασίας στο IPFS και υποβάλλει το CID στο smart contract *TasksManager*.

9. Πληρωμή: Μετά την επιτυχή εκτέλεση και υποβολή των αποτελεσμάτων από τον πάροχο, ο πελάτης ενημερώνεται για το ποσό που οφείλει να πληρώσει, με βάση την διάρκεια εκτέλεσης της εργασίας. Στην συνέχεια αποστέλλει στο smart contract *TasksManager* το ποσό της πληρωμής που εκκρεμεί και τότε μπορεί να αποκτήσει πρόσβαση στο CID των αποτελεσμάτων. Με την επιτυχημένη ολοκλήρωση της πληρωμής η βαθμολογία του πελάτη αυξάνεται, ως δείκτης φερεγγυότητας.

3.4.2 Εναλλακτικές ροές εργασιών

Ανεπιτυχής ολοκλήρωση της εργασίας λόγω ασυμφωνίας επαλήθευσης ή καθυστερημένης εκτέλεσης

Κατά τη λήψη των αποτελεσμάτων από τον πάροχο, το smart contract TasksManager κατακερματίζει και ελέγχει την συμβολοσειρά επαλήθευσης. Εάν η κατακερματισμένη συμβολοσειρά επαλήθευσης δεν ταιριάζει με αυτήν που παρέχεται από τον πελάτη, η εργασία θεωρείται ανεπιτυχής.

Ταυτόχρονα, ελέγχεται και ο χρόνος ολοκλήρωσης της εκτέλεσης. Εάν η εργασία ολοκληρώθηκε μετά τη συμφωνηθείσα προθεσμία, θεωρείται επίσης ανεπιτυχής.

Και στις δύο περιπτώσεις ανεπιτυχούς επαλήθευσης ή καθυστερημένης εκτέλεσης, η βαθμολογία του παρόχου μειώνεται, ενώ οι χρηματικές εγγυήσεις του χάνονται και παραμένουν στο έξυπνο συμβόλαιο TasksManager. Στον πελάτη επιστρέφονται οι δικές του χρηματικές εγγυήσεις.

Καθυστέρηση του παρόχου στην υποβολή των αποτελεσμάτων

Μετά την επιτυχημένη εκτέλεση της εργασίας, ο πάροχος έχει περιθώριο μίας ημέρας για να υποβάλει τα αποτελέσματα στο έξυπνο συμβόλαιο TasksManager. Εάν ο πάροχος δεν υποβάλει τα αποτελέσματα εντός αυτού του χρονικού διαστήματος, η βαθμολογείται αρνητικά και χάνει τις χρηματικές εγγυήσεις του, οι οποίες παραμένουν στο συμβόλαιο. Παράλληλα, επιστρέφονται στον πελάτη οι δικές του χρηματικές εγγυήσεις.

Δικαίωμα ακύρωσης του πελάτη πριν από την ενεργοποίηση της εργασίας

Μέχρι ο πάροχος να ενεργοποιήσει την υπολογιστική εργασία, ο πελάτης έχει το δικαίωμα να την ακυρώσει. Κατά την ακύρωση, οι εξασφαλίσεις του πελάτη επιστρέφονται. Στο βήμα αυτό ο πάροχος δεν έχει αποστείλει ακόμα τις δικές του χρηματικές εγγυήσεις ώστε να του επιστραφούν.

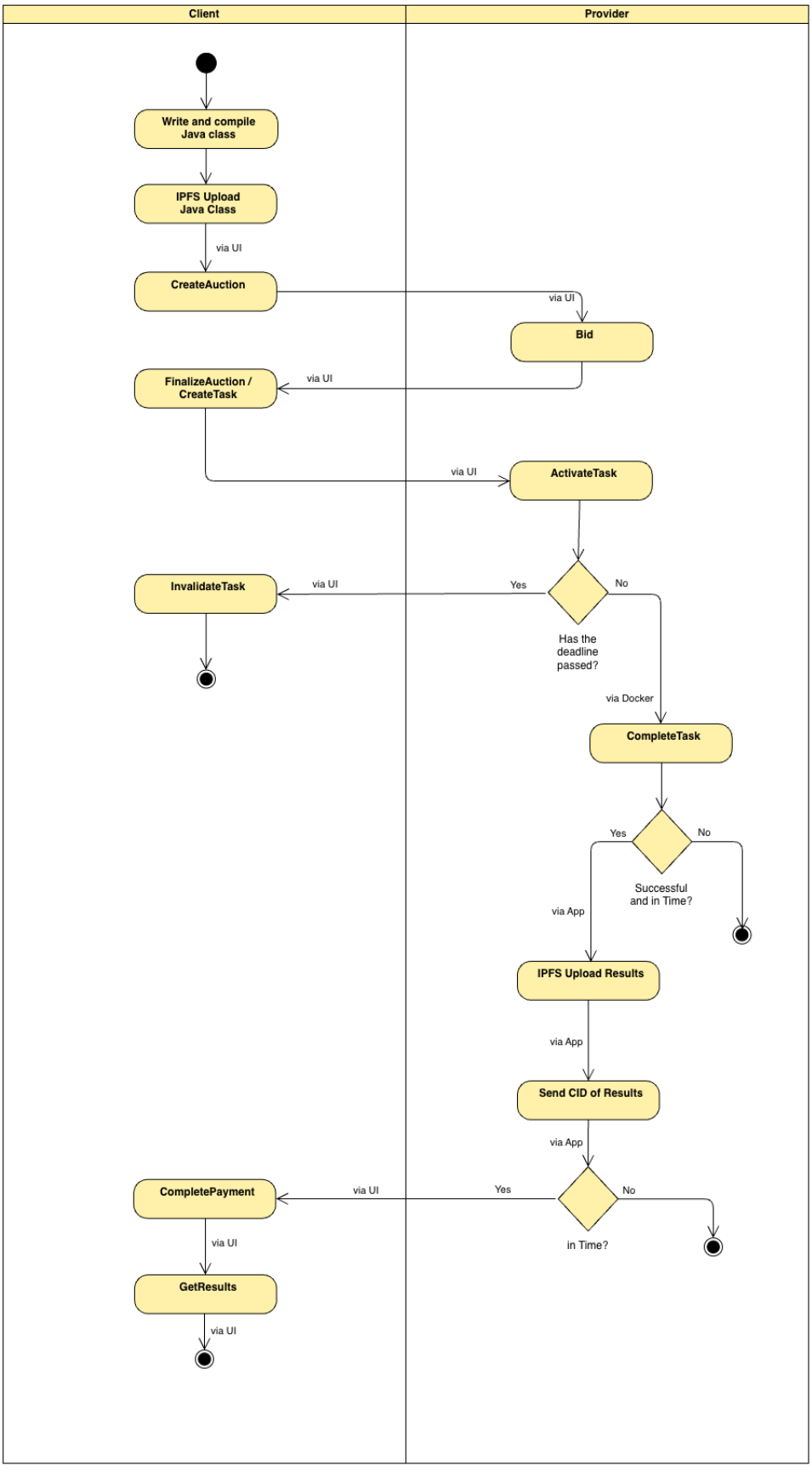
Δικαίωμα ακύρωσης του πελάτη για καθυστερημένη ολοκλήρωση

Στην περίπτωση που έχει παρέλθει ο χρόνος της προσυμφωνημένης προθεσμίας για την εκτέλεση της υπολογιστικής εργασίας, δίνοντας ακόμη περιθώριο μιας ημέρας, ο πελάτης έχει το δικαίωμα να ακυρώσει την εργασία. Στην περίπτωση αυτή, μεταφέρεται στον πελάτη τόσο η δική του χρηματική εγγύηση, όσο και του παρόχου.

Μη ολοκλήρωση πληρωμής από τον πελάτη

Στην περίπτωση επιτυχημένης εκτέλεσης της εργασίας εκ μέρους του παρόχου, ο πελάτης έχει το περιθώριο μιας ημέρας για να ολοκληρώσει την πληρωμή. Αν αυτό δεν συμβεί, ο πάροχος έχει το δικαίωμα να τον καταγγείλει. Τότε η βαθμολογία του πελάτη μειώνεται, ως δείγμα αφερεγγυότητας.

Παρακάτω παρουσιάζεται το Activity Diagram της dApp για τις βασικές ροές εκτέλεσής της.



Εικόνα 3.3: *Activity diagram της dApp*

Ενώ οι εναλλακτικές ροές εξασφαλίζουν ευελιξία και ευρωστία σε διάφορα σενάρια, το θεμέλιο αυτής της dApp έγκειται στα μέτρα ασφαλείας της. Τα μέτρα αυτά όχι μόνο προστατεύουν τα συμφέροντα των πελατών και των παρόχων, αλλά διασφαλίζουν επίσης τη συνολική αξιοπιστία του συστήματος.

Παρακάτω παρουσιάζονται οι μηχανισμοί ασφάλειας της dApp.

3.5 Μέτρα ασφαλείας για τη διασφάλιση της αξιοπιστίας

Η αξιοπιστία του συστήματος διασφαλίζεται εφαρμόζοντας τα ακόλουθα μέτρα:

- Συμβολοσειρά επαλήθευσης: Ο πελάτης παρέχει μια συμβολοσειρά επαλήθευσης που επιθυμεί, η οποία κατακερματίζεται (πραγματοποιείται hash με τον αλγόριθμο keccak256 που χρησιμοποιεί το Ethereum virtual machine) και αποθηκεύεται στο έξυπνο συμβόλαιο. Αυτό διασφαλίζει ότι ο πάροχος έχει εκτελέσει πραγματικά την εργασία, καθώς για την επαλήθευση πρέπει να επιστρέψει στο συμβόλαιο την σωστή συμβολοσειρά στην αρχική της μορφή (πριν κατακερματιστεί) χωρίς να έχει άμεση πρόσβαση σε αυτή.

- Docker containerization: Για να διασφαλιστεί η ασφάλεια του μηχανήματος του παρόχου και να αποτραπεί η αλλοίωση της εκτέλεσης από αυτόν, οι εργασίες εκτελούνται σε docker containers. Η ενθυλάκωση αυτή εξασφαλίζει ένα συνεπές και απομονωμένο περιβάλλον για την εκτέλεση των εργασιών.

- Blockchain: Όλες οι αλληλεπιδράσεις, από την έναρξη της δημοπρασίας έως την ολοκλήρωση της εργασίας, καταγράφονται στο Blockchain. Αυτό διασφαλίζει τη διαφάνεια και επιτρέπει στους ενδιαφερόμενους να επαληθεύουν όλες τις ενέργειες.

- Μηχανισμός εγγυήσεων: Τόσο ο πελάτης όσο και ο πάροχος αποστέλλουν χρηματικές εγγυήσεις στο έξυπνο συμβόλαιο. Αυτό λειτουργεί ως δέσμευση και διασφαλίζει ότι και τα δύο μέρη ενεργούν με ειλικρίνεια και έχουν κίνητρο για την επιτυχή ολοκλήρωση της διαδικασίας, καθώς σε περίπτωση ασυμφωνίας στην οποία ευθύνονται μπορεί να χάσουν τις εγγυήσεις τους.

- Επικοινωνία μέσω events: Η dApp και τα έξυπνα συμβόλαια επικοινωνούν μέσω events που παρέχουν τα συμβόλαια. Αυτό διασφαλίζει ότι όλα τα ενδιαφερόμενα μέρη ενημερώνονται άμεσα για τα διάφορα στάδια του κύκλου ζωής της εργασίας.

Με την ενσωμάτωση αυτών των μέτρων ασφαλείας, η dApp διασφαλίζει ένα αξιόπιστο περιβάλλον τόσο για τους πελάτες όσο και για τους παρόχους, προωθώντας τη δίκαιη και διαφανή εκτέλεση των υπολογιστικών εργασιών.

Κεφάλαιο 4

Μηχανισμός δημοπρασίας

Ο μηχανισμός δημοπρασίας λειτουργεί ως βασικό συστατικό της προτεινόμενης αποκεντρωμένης εφαρμογής (dApp), δημιουργώντας μια διαφανή και δίκαιη πλατφόρμα για την κατανομή των υπολογιστικών εργασιών. Η παρούσα ενότητα διευκρινίζει την περίπλοκη δυναμική της διαδικασίας δημοπρασίας, τις στρατηγικές υποβολής προσφορών που χρησιμοποιούν οι πάροχοι και τα κριτήρια που καθοδηγούν την επιλογή ενός παρόχου από έναν πελάτη.

4.1 Επισκόπηση της διαδικασίας δημοπρασίας

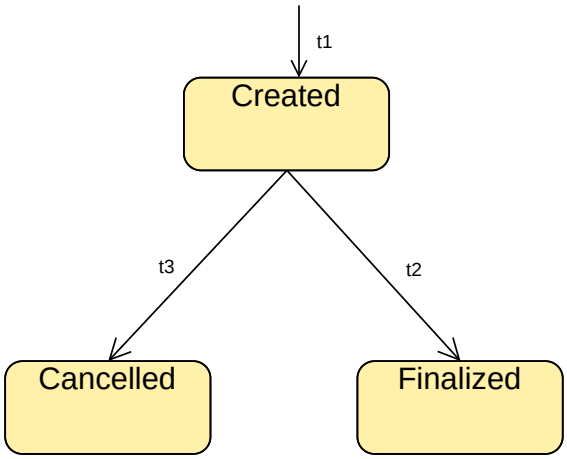
Η διαδικασία δημοπρασίας ξεκινά όταν ένας πελάτης υποβάλλει μια υπολογιστική εργασία στην dApp. Η διαδικασία αυτή είναι δομημένη ώστε να διασφαλίζεται η δικαιοσύνη, η διαφάνεια και η βέλτιστη κατανομή εργασιών:

1. Υποβολή εργασίας: Οι πελάτες ξεκινούν τη δημοπρασία αναφέροντας τις λεπτομέρειες της υπολογιστικής εργασίας και συγκεκριμένα την αναμενόμενη προθεσμία ολοκλήρωσης και τη σχετική συμβολοσειρά επαλήθευσης.

2. Υποβολή προσφορών: Μετά την υποβολή της εργασίας, οι πάροχοι έχουν την ευκαιρία να εξετάσουν τις λεπτομέρειες της εργασίας και να υποβάλλουν προσφορές με την προτεινόμενη τιμή τους (σε wei ανά δευτερόλεπτο εκτέλεσης της εργασίας).

3. Επιλογή παρόχου: Όσο η δημοπρασία παραμένει ενεργή, ο πελάτης μπορεί να επιλέξει τον πάροχο που επιθυμεί, βασιζόμενος σε διάφορους παράγοντες που θα αναλυθούν παρακάτω. Η απόφαση αυτή σηματοδοτεί την ολοκλήρωση της δημοπρασίας και την έναρξη της φάσης εκτέλεσης της εργασίας.

Παρακάτω παρουσιάζεται ο κύκλος ζωής μιας δημοπρασίας, με βάση την κατάσταση AuctionState της οντότητας Auction.



Εικόνα 4.1: *State machine του Auction, με βάση το AuctionState*
Οι μεταβάσεις t1, t2 και t3 πραγματοποιούνται από τον client.

4.2 Μηχανισμός βαθμολόγησης

Στο smart contract TasksManager αποθηκεύεται το ιστορικό των επιδόσεων των παρόχων και των πελατών στην διαδικασία μέσω ενός μηχανισμού βαθμολόγησης. Ο μηχανισμός αυτός είναι καθοριστικής σημασίας για τη διατήρηση της ακεραιότητας της dApp, διασφαλίζοντας ότι τόσο οι πάροχοι όσο και οι πελάτες λογοδοτούν για τις πράξεις και τις επιδόσεις τους.

- Βαθμολογία του παρόχου: Σε κάθε πάροχο αποδίδεται μια βαθμολογία, η οποία προκύπτει από τον συνδυασμό των ανεπιτυχών (downVotes) και επιτυχών (upVotes) εκτελέσεων εργασιών, η οποία αντικατοπτρίζει το ιστορικό των επιδόσεων τους. Ο υπολογισμός της βαθμολογίας χρησιμοποιεί τη μεθοδολογία 'ταξινόμησης εμπιστοσύνης' που βασίζεται στο διάστημα βαθμολογίας Wilson, έναν διάσημο αλγόριθμο που χρησιμοποιείται από πλατφόρμες όπως το Reddit για την ταξινόμηση των σχολίων στην πλατφόρμα. Με τον τρόπο αυτό διασφαλίζεται ότι η βαθμολογία δεν είναι απλώς ένας μέσος όρος, αλλά μια αναπαράσταση της αξιοπιστίας του παρόχου με βάση τον όγκο και την αναλογία των upvotes προς τα downvotes του.

- Βαθμολογία πελάτη: Στους πελάτες αποδίδεται επίσης μια αντίστοιχη βαθμολογία, ενδεικτική των ιστορικών αλληλεπιδράσεών τους και της συνέπειάς τους στην αμοιβή των παρόχων για επιτυχώς εκτελεσμένες εργασίες. Η βαθμολογία αυτή παίζει καθοριστικό ρόλο στη διαμόρφωση της στρατηγικής προσφορών του παρόχου.

4.3 Κριτήρια που καθοδηγούν τον πελάτη στην επιλογή παρόχου

Η επιλογή του παρόχου από τον πελάτη επηρεάζεται από μια σειρά παραγόντων, διασφαλίζοντας ότι η απόφαση είναι τόσο τεκμηριωμένη όσο και βέλτιστη:

- Προσφορά τιμής: Η προτεινόμενη τιμή του παρόχου παραμένει πρωταρχικής σημασίας, με τους πελάτες να κλίνουν φυσικά προς τις ανταγωνιστικές τιμές.
- Το ιστορικό επιδόσεων του παρόχου: Η βαθμολογία του παρόχου, όπως υπολογίζεται με την προαναφερθείσα μεθοδολογία, προσφέρει πληροφορίες σχετικά με την αξιοπιστία και τις προηγούμενες επιδόσεις του.
- Προηγούμενες συνεργασίες: Προηγούμενες συνεργασίες με τον συγκεκριμένο πάροχο που κατέληξαν σε θετικά αποτελέσματα μπορούν να επηρεάσουν σημαντικά την απόφαση ενός πελάτη.

4.4 Παράγοντες που διαμορφώνουν τη στρατηγική προσφορών του παρόχου

Οι πάροχοι, κατά τη διαμόρφωση των προσφορών τους, εξετάζουν πληθώρα παραγόντων για να βελτιστοποιήσουν τις πιθανότητές τους να επιλεγούν, εξασφαλίζοντας παράλληλα την κερδοφορία τους:

- Η αξιοπιστία του πελάτη: Η βαθμολογία ενός πελάτη, ενδεικτική των προηγούμενων αλληλεπιδράσεών του και της συνέπειας των πληρωμών του, μπορεί να επηρεάσει το ποσό της προσφοράς του παρόχου.
- Διαθεσιμότητα πόρων του παρόχου: Οι πάροχοι με άφθονους πόρους ενδέχεται να έχουν την προδιάθεση να υποβάλουν χαμηλότερη προσφορά για να εξασφαλίσουν την εργασία.
- Ιστορικό αλληλεπιδράσεων με τον πελάτη: Οι θετικές αλληλεπιδράσεις του παρελθόντος με τον πελάτη μπορούν να παρακινήσουν έναν πάροχο να υποβάλει μια πιο ανταγωνιστική προσφορά.

Συνοψίζοντας, ο μηχανισμός δημοπρασίας, ο οποίος υποστηρίζεται από τη διαφανή διαδικασία υποβολής προσφορών, τα κριτήρια αξιολόγησης και το σύστημα βαθμολόγησης, ενισχύει τη δέσμευση της dApp για την προώθηση ενός αξιόπιστου και αποτελεσματικού περιβάλλοντος για την αποκεντρωμένη εφαρμογή υπολογιστικής νέφους.

Κεφάλαιο 5

Εκτέλεση εργασίας και επαλήθευση

Το στάδιο εκτέλεσης και επαλήθευσης της εργασίας αποτελεί τον πυρήνα της αποκεντρωμένης εφαρμογής (dApp), διασφαλίζοντας ότι οι υπολογιστικές εργασίες όχι μόνο εκτελούνται με ακρίβεια και αποτελεσματικότητα από τον επιλεγμένο πάροχο, αλλά και ότι τα αποτελέσματα είναι επαληθεύσιμα και γνήσια. Αυτό το κεφάλαιο εμβαθύνει στις ιδιαιτερότητες της εκτέλεσης εργασιών, στον μηχανισμό που χρησιμοποιείται για την επαλήθευση της ορθότητας της εκτέλεσης και στην ασφαλή και διαφανή διαδικασία πληρωμής που διευκολύνεται από την τεχνολογία Blockchain.

5.1 Εκτέλεση εργασιών

Η εκτέλεση μιας υπολογιστικής εργασίας είναι μια σχολαστική διαδικασία, η οποία διασφαλίζει ότι ο πάροχος τηρεί τις προκαθορισμένες απαιτήσεις και εκτελεί την εργασία εντός του καθορισμένου περιβάλλοντος.

1. Ενεργοποίηση της εργασίας: Μετά την επιλογή του από τον πελάτη, ο πάροχος ενεργοποιεί την εργασία, σηματοδοτώντας τη δέσμευσή του για την εκτέλεσή της. Στα πλαίσια της δέσμευσης αυτής, στο σημείο αυτό στέλνει στο συμβόλαιο χρηματική εγγύηση ίση με την αξία εκτέλεσης της εργασίας για 2 δευτερόλεπτα.

2. Περιβάλλον απομόνωσης: Για να διασφαλιστεί το σύστημα του παρόχου και να εξασφαλιστεί ένα συνεπές περιβάλλον εκτέλεσης, οι εργασίες εκτελούνται εντός Docker container. Αυτή η ενθυλάκωση απομονώνει την εκτέλεση από εξωτερικές παρεμβάσεις και πιθανές απειλές ασφαλείας.

3. Ανάκτηση κώδικα: Ο πάροχος ανακτά τη μεταγλωττισμένη κλάση Java του πελάτη από το IPFS χρησιμοποιώντας το CID της.

4. Εκτέλεση εργασιών: Εντός του Docker container, η κλάση Java εκτελείται, τηρώντας τη λογική που περιέχεται στη μέθοδο *getComputation* από τον πελάτη.

5. Αποστολή αποτελεσμάτων: Τα αποτελέσματα εκτέλεσης της υπολογιστικής εργασίας αποθηκεύονται σε ένα αρχείο, μεταφορτώνονται στο IPFS και το CID τους αποστέλλεται στο smart contract ώστε να το λάβει ο πελάτης μετά την ολοκλήρωση της πληρωμής του.

5.2 Μηχανισμός επαλήθευσης

Η διασφάλιση της πραγματικής εκτέλεσης της εργασίας είναι καθοριστικής σημασίας και η dApp χρησιμοποιεί έναν ισχυρό μηχανισμό επαλήθευσης για να εξακριβώσει τη γνησιότητα των αποτελεσμάτων.

1. Συμβολοσειρά επαλήθευσης: Η μέθοδος *getVerification* εντός της εκτελούμενης κλάσης Java επιστρέφει μια προκαθορισμένη συμβολοσειρά, η οποία είναι άγνωστη στον πάροχο.

2. Σύγκριση κατακερματισμένης συμβολοσειράς: Η επιστρεφόμενη συμβολοσειρά επαλήθευσης κατακερματίζεται και συγκρίνεται με τον αρχικό κατακερματισμό που δόθηκε από τον πελάτη κατά την έναρξη της δημοπρασίας.

3. Επικύρωση αποτελέσματος: Εάν οι κατακερματισμένες συμβολοσειρές ταιριάζουν, το αποτέλεσμα θεωρείται γνήσιο και η εργασία κρίνεται επιτυχημένη. Αντίθετα, μια αναντιστοιχία υποδηλώνει ασυμφωνία, ενεργοποιώντας την ανεπιτυχή ροή της εργασίας.

4. Επαλήθευση χρόνου: Επιπλέον, ο χρόνος εκτέλεσης επαληθεύεται σε σχέση με τη συμφωνηθείσα προθεσμία, ώστε να διασφαλιστεί η έγκαιρη ολοκλήρωση της εργασίας.

5.3 Διαδικασία πληρωμής

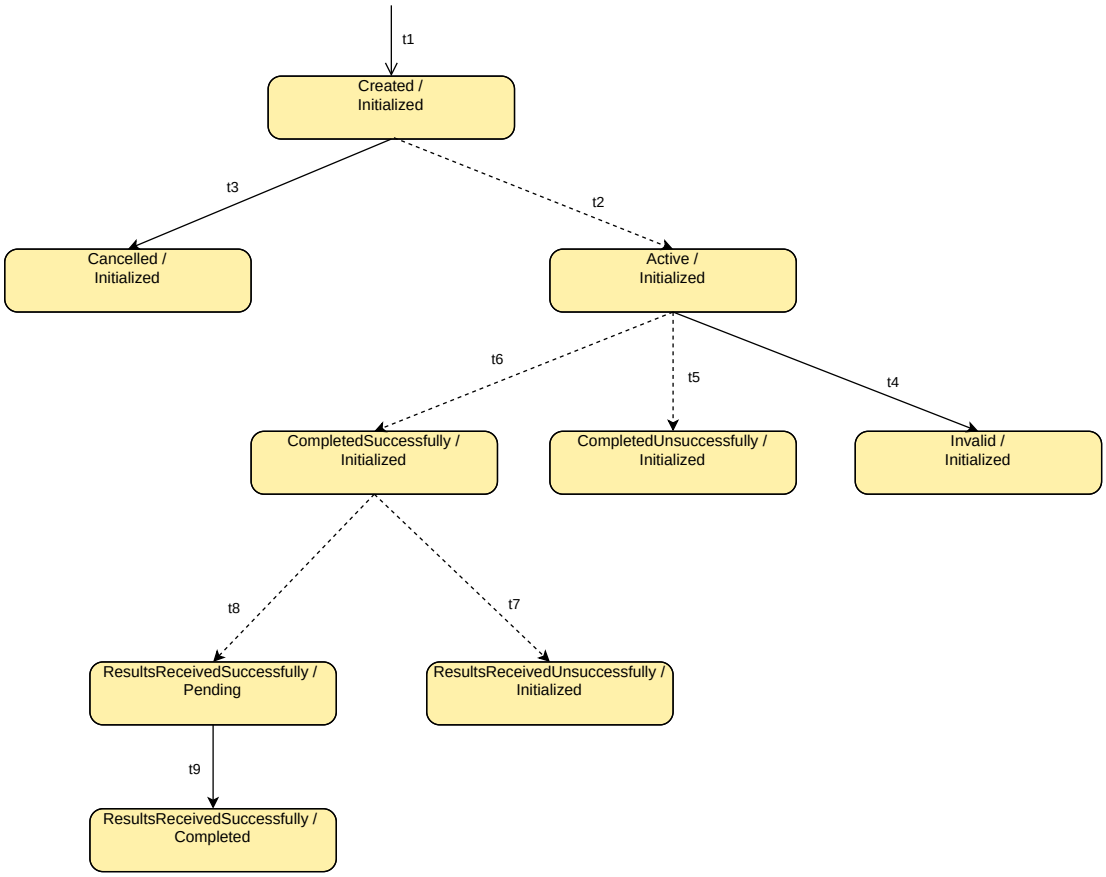
Μετά την επιτυχή επαλήθευση, ξεκινά η διαδικασία πληρωμής, διασφαλίζοντας μια διαφανή συναλλαγή μεταξύ του πελάτη και του παρόχου.

1. Υπολογισμός πληρωμής: Το ποσό πληρωμής υπολογίζεται με βάση τη συμφωνηθείσα τιμή (wei ανά δευτερόλεπτο) και τον πραγματικό χρόνο εκτέλεσης. Από αυτό αφαιρείται το ποσό που έχει ήδη στείλει στο συμβόλαιο ο πελάτης ως χρηματική εγγύηση.

2. Συναλλαγή blockchain: Η πληρωμή πραγματοποιείται μέσω του έξυπνου συμβολαίου TasksManager στο Ethereum blockchain, εξασφαλίζοντας ασφάλεια και διαφάνεια.

3. Χειρισμός εγγυήσεων: Μετά την επιτυχή εκτέλεση της εργασίας, επιστρέφονται στον πάροχο οι χρηματικές εγγυήσεις που είχε αποστείλει κατά την έναρξη της διαδικασίας. Σε περιπτώσεις ανεπιτυχούς εκτέλεσης της εργασίας, οι χρηματικές εγγυήσεις του υπεύθυνου χάνονται, όπως περιγράφεται λεπτομερώς στο προηγούμενο κεφάλαιο.

Παρακάτω παρουσιάζεται ο κύκλος ζωής μιας εργασίας (Task), με βάση τις καταστάσεις TaskState και PaymentState της οντότητας Task.



Εικόνα 5.1: *State machine του Task, με βάση το TaskState/PaymentState*
Η μετάβαση t1 πραγματοποιείται από το AuctionsManager μετά από την αντίστοιχη ενέργεια του client. Οι μεταβάσεις t3, t4 και t9 πραγματοποιούνται από τον client, ενώ οι μεταβάσεις t2, t5, t6, t7, t8 και t10 από τον provider.

5.4 Ασφάλεια κατά την εκτέλεση και την πληρωμή

Η dApp διακατέχεται από διάφορα μέτρα ασφαλείας για τη διασφάλιση των συμφερόντων τόσο των πελατών όσο και των παρόχων κατά την εκτέλεση και την πληρωμή.

- Αμετάβλητες λεπτομέρειες εργασιών: Οι λεπτομέρειες των εργασιών, αφού μεταφορτωθούν στο IPFS, είναι αμετάβλητες, διασφαλίζοντας ότι η εκτέλεση τηρεί τις αρχικές απαιτήσεις.

- Απομονωμένο περιβάλλον εκτέλεσης: Το Docker container διασφαλίζει ότι το σύστημα του παρόχου παραμένει απομονωμένο από πιθανό κακόβουλο κώδικα και ότι το περιβάλλον εκτέλεσης είναι συνεπές.

- Διαφανείς συναλλαγές: Όλες οι συναλλαγές, συμπεριλαμβανομένων των πληρωμών και των τοποθετήσεων εγγυήσεων, καταγράφονται στο blockchain, εξασφαλίζοντας διαφάνεια και επαληθευσιμότητα.

- Μηχανισμός εγγυήσεων: Ο μηχανισμός εγγυήσεων διασφαλίζει τη δέσμευση και των δύο μερών και χρησιμεύει ως αποτρεπτικός παράγοντας έναντι κακόβουλων δραστηριοτήτων ή μη συμμόρφωσης.

Ανακεφαλαιώνοντας, το στάδιο εκτέλεσης και επαλήθευσης των εργασιών είναι σχολαστικά σχεδιασμένο ώστε να διασφαλίζεται ότι οι εργασίες εκτελούνται με ακρίβεια, τα αποτελέσματα είναι επαληθεύσιμα και οι πληρωμές είναι ασφαλείς και διαφανείς. Αυτή η φάση, η οποία υποστηρίζεται από την τεχνολογία blockchain και τους μηχανισμούς που περιγράφηκαν, ενισχύει την αξιοπιστία της dApp στο αποκεντρωμένο υπολογιστικό νέφος.

Κεφάλαιο 6

Υλοποίηση του συστήματος

Η ανάπτυξη της αποκεντρωμένης εφαρμογής (dApp) περιλαμβάνει ένα συνδυασμό τεχνολογιών αιχμής και καινοτόμων λύσεων για την αντιμετώπιση των προκλήσεων του αποκεντρωμένου υπολογιστικού νέφους. Αυτό το κεφάλαιο εμβαθύνει στις τεχνικές ιδιαιτερότητες της dApp, τα εργαλεία και τις πλατφόρμες που χρησιμοποιήθηκαν και τις προκλήσεις που αντιμετωπίστηκαν κατά την ανάπτυξή της.

6.1 Τεχνικές ιδιαιτερότητες της dApp

Η αρχιτεκτονική της dApp συνδυάζει την τεχνολογία του Blockchain, του containerization και των κατανεμημένων συστημάτων αρχείων, εξασφαλίζοντας ένα αποτελεσματικό και ασφαλές σύστημα.

1. Έξυπνα συμβόλαια: Αναπτυγμένα με τη χρήση της Solidity, τα έξυπνα συμβόλαια AuctionsManager και TasksManager αναπτύσσονται στα blockchains Ethereum και Polygon. Διαχειρίζονται τη διαδικασία δημοπρασίας, τον κύκλο ζωής των εργασιών, τις πληρωμές και τα χρηματικές εγγυήσεις.

2. Ενσωμάτωση του IPFS: Το Διαπλανητικό Σύστημα Αρχείων (IPFS) είναι ένα αποκεντρωμένο σύστημα αποθήκευσης, με βάση την διεύθυνση του περιεχομένου, το οποίο διανέμει αποθηκευμένα αρχεία μεταξύ ομότιμων χρηστών σε ένα δίκτυο P2P. Το περιεχόμενο κάθε αρχείου κατακερματίζεται και ο κατακερματισμός αυτός (Content Identifier - CID) χρησιμοποιείται για την ταυτοποίηση και ανάκτηση του συγκεκριμένου αρχείου [42]. Το σύστημα της εργασίας χρησιμοποιεί το IPFS για την αποθήκευση και ανάκτηση των μεταγλωττισμένων κλάσεων Java και των αποτελεσμάτων της εκτέλεσης. Αυτό διασφαλίζει την αμεταβλητότητα των δεδομένων και την αποκεντρωμένη πρόσβαση.

3. Docker Containerization: Το Docker χρησιμοποιείται για τη δημιουργία απομονωμένων περιβαλλόντων για την εκτέλεση εργασιών, εξασφαλίζοντας σταθερή απόδοση και ασφάλεια έναντι πιθανών απειλών.

4. Γραφική διεπαφή της εφαρμογής: Αναπτυγμένη με τη χρήση του React, η γραφική διεπαφή διευκολύνει τις αλληλεπιδράσεις του χρήστη, από την έναρξη δημοπρασίας έως την παρακολούθηση εργασιών.

6.2 Εργαλεία, γλώσσες προγραμματισμού και πλατφόρμες ανάπτυξης

Για την ανάπτυξη της dApp αξιοποιήθηκε πλήθος εργαλείων και πλατφορμών με σκοπό την βέλτιστη απόδοση και εμπειρία των χρηστών.

1. Γλώσσες προγραμματισμού

   - Solidity: Για την ανάπτυξη έξυπνων συμβολαίων στο blockchain του Ethereum και του Polygon.
   - Java: Για την υλοποίηση της υπολογιστικής εργασίας τους πελάτη.
   - Javascript: Για την γραφική διεπαφή (frontend) της dApp.
   - Typescript: Για τις λειτουργίες του backend της dApp.

2. Εργαλεία και πλατφόρμες ανάπτυξης

   - Hardhat: Περιβάλλον ανάπτυξης που χρησιμοποιείται για την ανάπτυξη εφαρμογών στο Ethereum blockchain [43].
   - Metamask: Επέκταση (plugin) του προγράμματος περιήγησης (browser) που λειτουργεί ως πορτοφόλι για το Ethereum και επιτρέπει οικονομικές αλληλεπιδράσεις με την dApp [44].
   - Ganache: Προσωπικό τοπικό blockchain που χρησιμοποιείται για το testing κατά την ανάπτυξη εφαρμογών στο Ethereum.
   - Web3.js: Βιβλιοθήκη της Javascript που επιτρέπει τις αλληλεπιδράσεις μεταξύ κόμβων του Ethereum blockchain.
   - React: Framework της Javascript που χρησιμοποιείται για την ανάπτυξη της γραφικής διεπαφής εφαρμογών.
   - Testnets: Για σκοπούς πειραματισμού και ελέγχου ορθής λειτουργίας, η dApp αναπτύχθηκε στα πειραματικά live blockchains (testnets) του Ethereum (Sepolia testnet) και του Polygon (Mumbai testnet), τα οποία προσομοιώνουν την λειτουργία των πραγματικών αντίστοιχων blockchains (Mainnets).

6.3 Προκλήσεις και λύσεις

Κατά την ανάπτυξη της dApp, αντιμετωπίστηκαν οι εξής προκλήσεις:

1. Αμεταβλητότητα των δεδομένων: Η διασφάλιση ότι οι υπολογιστικές εργασίες παρέμεναν αμετάβλητες ήταν πρωταρχικής σημασίας. Η πρόκληση αντιμετωπίστηκε με την ενσωμάτωση του IPFS, το οποίο εξασφαλίζει τον αμετάβλητο χαρακτήρα των δεδομένων και παρέχει αποκεντρωμένη αποθήκευση.

2. Μηχανισμός επαλήθευσης: Ο σχεδιασμός ενός ελαφρύ αλλά αποτελεσματικού μηχανισμού επαλήθευσης αποτέλεσε πρόκληση. Η λύση ήταν η εισαγωγή της μεθόδου getVerification, η οποία παρέχει μια απλή αλλά ισχυρή διαδικασία επαλήθευσης.

3. Συνέπεια του περιβάλλοντος εκτέλεσης: Πρόκληση αποτέλεσε και η εξασφάλιση ενός συνεπούς περιβάλλοντος εκτέλεσης σε διαφορετικούς παρόχους. Η λύση ήταν το Docker containerization, το οποία προσέφερε ένα απομονωμένο και συνεπές περιβάλλον για την εκτέλεση κάθε εργασίας.

4. Ανησυχίες σχετικά με την ασφάλεια: Η προστασία του συστήματος του παρόχου από πιθανό κακόβουλο κώδικα ήταν μια σημαντική πρόκληση. Η προσέγγιση του Docker containerization, σε συνδυασμό με το απομονωμένο περιβάλλον εκτέλεσης, αντιμετώπισε αποτελεσματικά και αυτή την ανησυχία.

5. Κόστος συναλλαγών: Οι συναλλαγές στο Ethereum συνοδεύονται από κόστος σε gas. Η βελτιστοποίηση των λειτουργιών των έξυπνων συμβολαίων ήταν απαραίτητη για την ελαχιστοποίηση αυτών των εξόδων, εξασφαλίζοντας την όσο το δυνατόν πιο προσιτή τιμή για τους χρήστες.

Εν κατακλείδι, για την υλοποίηση της dApp παρουσιάστηκαν πολλές προκλήσεις, η επίλυση των οποίων ήταν απαραίτητη για την απόδειξη των δυνατοτήτων του αποκεντρωμένου υπολογιστικού νέφους. Ο συνδυασμός τεχνολογιών και οι στρατηγικές λύσεις που χρησιμοποιήθηκαν κατέληξαν σε μια πλατφόρμα που υπόσχεται πραγματική εκτέλεση, ασφάλεια και διαφάνεια στο πεδίο της αποκεντρωμένης πληροφορικής.

Μέρος III

Επίλογος

Κεφάλαιο 7

Επίλογος

7.1 Συμπεράσματα

Στην παρούσα εργασία ερευνήθηκε ο χώρος των αποκεντρωμένων εφαρμογών, εστιάζοντας ειδικά σε μια dApp σχεδιασμένη για υπολογιστικό νέφος. Η dApp αυτή αντιπροσωπεύει μια σημαντική αλλαγή στον τομέα του υπολογιστικού νέφους, διαφοροποιημένο από τα παραδοσιακά συγκεντρωτικά μοντέλα, εισάγοντας ένα εκδημοκρατισμένο, διαφανές και ασφαλές οικοσύστημα. Αξιοποιώντας έναν απλό μηχανισμό επαλήθευσης, το docker containerization και ένα σύστημα κατανομής εργασιών με βάση τη δημοπρασία, η dApp αντιμετωπίζει βασικές προκλήσεις στον τομέα του υπολογιστικού νέφους. Η δυνατότητά του να αναδιαμορφώσει το τοπίο, καθιστώντας το cloud computing πιο προσιτό, οικονομικά αποδοτικό και αξιόπιστο, υπογραμμίζει τη σημασία του στην εξελισσόμενη ψηφιακή εποχή.

7.2 Μελλοντικές Επεκτάσεις

Το τοπίο του αποκεντρωμένου υπολογιστικού νέφους είναι γεμάτο δυνατότητες και ευκαιρίες ανάπτυξης. Μελλοντικά, διάφοροι τομείς αναδεικνύονται ως κομβικοί για την εξέλιξη και την ενίσχυση αυτού του τομέα:

- Έλεγχος κλιμακωσιμότητας: Με την αυξανόμενη υιοθέτηση αποκεντρωμένων συστημάτων, η αντιμετώπιση της επεκτασιμότητας του blockchain καθίσταται ζωτικής σημασίας. Οι καινοτομίες σε λύσεις επιπέδου (layer) 2 του Ethereum blockchain [26] ή η διερεύνηση εναλλακτικών αλγορίθμων συναίνεσης θα μπορούσαν να είναι καθοριστικές.

- Βελτιωμένες διεπαφές χρήστη: Για να προωθηθεί η ευρύτερη υιοθέτηση, η εμπειρία του χρήστη των αποκεντρωμένων εφαρμογών πρέπει να είναι διαισθητική και φιλική προς τον αυτόν. Μια καλύτερη και πιο αποδοτική υλοποίηση της γραφικής διεπαφής της εφαρμογής κρίνεται απαραίτητη για την βελτίωση της εμπειρίας χρήσης της.

- Προηγμένα πρωτόκολλα ασφαλείας: Τα μέτρα ασφαλείας του dApp θέτουν ισχυρά θεμέλια, αλλά η δυναμική φύση των απειλών στον κυβερνοχώρο σημαίνει ότι τα πρωτόκολλα ασφαλείας πρέπει να εξελίσσονται συνεχώς. Αυτό θα μπορούσε να περιλαμβάνει την ενσωμάτωση προηγμένων κρυπτογραφικών τεχνικών κατά την εκτέλεση της υπολογιστικής εργασίας ή την λήψη των αποτελεσμάτων των υπολογισμών της από τον

πελάτη. Επιπλέον, η παρακολούθηση των εξελίξεων και εφαρμογή των σύγχρονων μέτρων ασφαλείας για το docker containerization κρίνεται απαραίτητη [45, 46].

- Διαλειτουργικότητα: Καθώς το αποκεντρωμένο οικοσύστημα επεκτείνεται, η εξασφάλιση απρόσκοπτων αλληλεπιδράσεων μεταξύ διαφορετικών πλατφορμών και συστημάτων blockchain θα είναι σημαντική. Αυτό απαιτεί την ανάπτυξη λύσεων cross-chain και τυποποιημένων πρωτοκόλλων.

- Έρευνα για οικονομικά αποδοτικές τεχνολογίες blockchain: Ένας από τους αποτρεπτικούς παράγοντες υιοθέτησης του σημερινού αποκεντρωμένου τοπίου είναι το υψηλό κόστος κρατήσεων (gas fees) που συνδέεται με ορισμένες τεχνολογίες blockchain, όπως το Ethereum. Η μελλοντική έρευνα θα μπορούσε να εμβαθύνει σε blockchains που προσφέρουν χαμηλότερα τέλη συναλλαγών, διασφαλίζοντας ότι το αποκεντρωμένο υπολογιστικό νέφος παραμένει προσιτό σε ένα ευρύτερο κοινό. Και εδώ θα μπορούσε να χρησιμοποιηθεί το Layer 2 του Ethereum με την χρήση Optimistic Rollups [26] και να δοκιμαστούν περισσότερο οι επιδόσεις σε αλυσίδες όπως το Polygon που χρησιμοποιήθηκε στην παρούσα εργασία.

- Επικοινωνία με τον έξω κόσμο: Καθώς τα smart contracts είναι ντετερμινιστικά, δεν επικοινωνούν ή επηρεάζονται από τον έξω κόσμο με real-time δεδομένα. Για την επίλυση αυτού του περιορισμού θα μπορούσαν να χρησιμοποιηθούν Oracles όπως το Chainlink [47], τα οποία μέσω ενός δικού τους blockchain χρησιμοποιούνται ως αποκεντρωμένα API μεταξύ των smart contracts και του έξω κόσμου. Με τον τρόπο αυτό θα μπορούσαν οι διαπραγματεύσεις για την εκτέλεση των εργασιών, αντί για wei, να γίνονται σε δολάρια, ευρώ ή οποιοδήποτε άλλο νόμισμα, και το smart contract να είναι υπεύθυνο για την μετατροπή των ισοτιμιών.

- Πλήρης αποκέντρωση: Για πλήρη αποκέντρωση, μπορούν τα apps που αφορούν το front end να γίνουν deploy στο Swarm, ένα peer-to-peer σύστημα αποθήκευσης αρχείων, αναπτυγμένο από το Ethereum Foundation, το οποίο επιτρέπει και την πρόσβαση σε ιστοσελίδες, αντικαθιστώντας τους κεντρικούς servers [48].

- Αλγόριθμος βαθμολόγησης επιδόσεων: Περαιτέρω μελέτη μπορεί να πραγματοποιηθεί και για την εύρεση ενός αλγορίθμου υπολογισμού της επίδοσης και φερεγγυότητας πελατών και παρόχων, ο οποίος θα αντιπροσωπεύει πιο ολοκληρωμένα την συμπεριφορά τους.

Βιβλιογραφία

[1] Greg Boss; Padma Malladi; Dennis Quan; Linda Legregni; Harold Hall. *Cloud Computing*. *IBM white paper*, 2007.

[2] Joe Baguley. *How cloud computing is changing the world ... without you knowing*. https://www.theguardian.com/media-network/media-network-blog/2013/sep/24/cloud-computing-changing-world-healthcare. Ημερομηνία πρόσβασης: 20-10-2022.

[3] Hong Xu; Baochun Li. *A Study of Pricing for Cloud Resources*. *Performance Evaluation Review*, 40(4), 2013.

[4] Bhanu Sharma; Ruppa K. Thulasiram; Parimala Thulasiraman; Saurabh K. Garg. *Pricing Cloud Compute Commodities: A Novel Financial Economic Model*. *12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2012.

[5] Hongyi Wang; Qingfeng Jing; Rishan Chen; Bingsheng He; Zhengping Qian; Lidong Zhou. *Distributed Systems Meet Economics: Pricing in the Cloud*. 2009.

[6] Dr. Rado Danilak. *Why Energy Is A Big And Rapidly Growing Problem For Data Centers*. https://www.forbes.com/sites/forbestechcouncil/2017/12/15/why-energy-is-a-big-and-rapidly-growing-problem-for-data-centers/. Ημερομηνία πρόσβασης: 21-10-2023.

[7] Nicola Jones. *How to stop data centres from gobbling up the world's electricity*. 2018.

[8] Michael Maximilien; David Hadas; Angelo Danducci II; Simon Moser. *The future is serverless*. https://developer.ibm.com/blogs/the-future-is-serverless, 2022. Ημερομηνία πρόσβασης: 14-10-2023.

[9] Robert Cordingly; Hanfei Yu; Varik Hoang; Zohreh Sadeghi; David Foster; David Perez; Rashad Hatchett; Wes Lloyd. *The Serverless Application Analytics Framework: Enabling Design Trade-off Evaluation for Serverless Software*. 2020.

[10] Ioana Baldini; Paul Castro; Kerry Chang; Perry Cheng; Stephen Fin; Vatche Ishakian; Nick Mitchell; Vinod Muthusamy; Rodric Rabbah; Aleksander Slominski και Philippe Suter. *Serverless Computing: Current Trends and Open Problems*. 2017.

[11] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. https://bitcoin.org/bitcoin.pdf.

[12] Christian Cachin; Marko Vukolic. *Blockchain Consensus Protocols in the Wild*. 2017.

[13] *Blockchain explained and its application to payments*. https://www.paiementor.com/blockchain-explained-application-payments/. Ημερομηνία πρόσβασης: 14-10-2023.

[14] V. Buterin. *Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform*. 2014. https://ethereum.org/en/whitepaper/.

[15] Dr. Gavin Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. 2023. [Version: 2bcdb2d] https://ethereum.github.io/yellowpaper/paper.pdf.

[16] Andreas M. Antonopoulos και Gavin Wood. *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media, Incorporated, 2018.

[17] Ethereum Foundation. *Introduction to smart contracts*. https://ethereum.org/en/smart-contracts/. Ημερομηνία πρόσβασης: 14-10-2023.

[18] Konstantinos Christidis; Michael Devetsikiotis. *Blockchains and Smart Contracts for the Internet of Things*. 2016.

[19] Ethereum Foundation. *What is a DApp?* https://ethereum.org/en/dapps/#beginner. Ημερομηνία πρόσβασης: 14-10-2023.

[20] Kaidong Wu; Yun Ma; Gang Huang; Xuanzhe Liu. *A First Look at Blockchain-based Decentralized Applications*. 2019.

[21] Ethereum Foundation. *What is Web3?* https://ethereum.org/en/web3/. Ημερομηνία πρόσβασης: 21-10-2023.

[22] Ethereum Foundation. *What is Ether (ETH)?* https://ethereum.org/en/eth/. Ημερομηνία πρόσβασης: 14-10-2023.

[23] Ethereum Foundation. *Gas Fees*. https://ethereum.org/en/gas/. Ημερομηνία πρόσβασης: 14-10-2023.

[24] Ethereum Foundation. *Ethereum's energy expenditure*. https://ethereum.org/en/energy-consumption/. Ημερομηνία πρόσβασης: 21-10-2023.

[25] Mohammad Musharraf. *What is the Blockchain Trilemma?* https://www.ledger.com/academy/what-is-the-blockchain-trilemma. Ημερομηνία πρόσβασης: 14-10-2023.

[26] Ethereum Foundation. *Layer 2*. https://ethereum.org/en/layer-2/. Ημερομηνία πρόσβασης: 14-10-2023.

[27] Magnus Westerlund; Nane Kratzke. *Towards Distributed Clouds*. 2018.

[28] Eric Jonas; Qifan Pu; Shivaram Venkataraman; Ion Stoica; Benjamin Recht. *Occupy the Cloud: Distributed Computing for the 99%*. 2017.

[29] Alex Kaplunovich; Karuna P. Joshi; Yelena Yesha. *Scalability Analysis of Blockchain on a Serverless Cloud*. 2019.

[30] Vladimir Yussupov; Ghareeb Falazi; Uwe Breitenbucher; Frank Leymann. *On the Serverless Nature of Blockchains and Smart Contracts*. 2020.

[31] Ch. V. N. U. Bharathi Murthy; M. Lawanya Shri; Seifedine Kadry. *Blockchain Based Cloud Computing: Architecture and Research Challenges*. 2020.

[32] Jin Ho Park; Jong Hyuk Park. *Blockchain Security in Cloud Computing: Use Cases, Challenges, and Solutions*. 2017.

[33] *SONM*. https://docs.sonm.com. Ημερομηνία πρόσβασης: 21-10-2023.

[34] *Golem*. https://www.golem.network. Ημερομηνία πρόσβασης: 21-10-2023.

[35] *iExec*. https://iex.ec. Ημερομηνία πρόσβασης: 21-10-2023.

[36] *Filecoin*. https://filecoin.io. Ημερομηνία πρόσβασης: 21-10-2023.

[37] Jason Teutsch; Christian Reitwießner. *A scalable verification solution for blockchains*. 2017.

[38] Katerina Doka; Tasos Bakogiannis; Ioannis Mytilinis και Georgios Goumas. *CloudAgora: Democratizing the Cloud*. 2019.

[39] Tasos Bakogiannis; Ioannis Mytilinis; Katerina Doka και Georgios Goumas. *Leveraging Blockchain Technology to Break the Cloud Computing Market Monopoly*. 2020.

[40] Sara Ghaemi. *Analysis and Design of Open Decentralized Serverless Computing Platforms*. 2020.

[41] Sara Ghaemi; Hamzeh Khazaei; Petr Musilek. *ChainFaaS: An Open Blockchain-Based Serverless Platform*. 2020.

[42] *IPFS*. https://ipfs.tech. Ημερομηνία πρόσβασης: 21-10-2023.

[43] *Hardhat*. https://hardhat.org. Ημερομηνία πρόσβασης: 21-10-2023.

[44] *MetaMask*. https://metamask.io. Ημερομηνία πρόσβασης: 21-10-2023.

[45] Fotis Loukidis – Andreou; Ioannis Giannakopoulos; Katerina Doka και Nectarios Koziris. *Docker-sec: A Fully Automated Container Security Enhancement Mechanism*. 2018.

[46] Ioannis Giannakopoulos; Konstantinos Papazafeiropoulos; Katerina Doka; Nectarios Koziris. *Isolation in Docker through Layer Encryption*. 2020.

[47] *Chainlink*. https://chain.link. Ημερομηνία πρόσβασης: 21-10-2023.

[48] *Swarm Ethereum*. https://www.ethswarm.org. Ημερομηνία πρόσβασης: 21-10-2023.