ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

ΠΡΟΓΡΑΜΜΑ ΔΙΔΑΚΤΟΡΙΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ασφαλής διαχείριση πόρων σε δίκτυα νέας γενιάς

Μιχαήλ Γ. Ξευγένης

ΑΙΓΑΛΕΩ

ΑΠΡΙΛΙΟΣ 2023

UNIVERSITY OF WEST ATTICA

SCHOOL OF ENGINEERING
DEPARTMENT OF INDUSTRIAL DESIGN AND PRODUCTION ENGINEERING

PROGRAM OF DOCTORAL STUDIES

PhD THESIS

Secure resource management in Next Generation Networks (NGNs)

Michael G. Xevgenis

ATHENS-EGALEO

APRIL 2023

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ασφαλής διαχείριση πόρων σε δίκτυα νέας γενιάς

Μιχαήλ Γ. Ξευγένης

ΕΠΙΒΛΕΠΟΥΣΑ ΚΑΘΗΓΗΤΡΙΑ: Ελένη Αικατερίνη Λελίγκου, Αναπληρώτρια Καθηγήτρια Τμ. ΜΒΣΠ, ΠαΔΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Ελένη Αικατερίνη Λελίγκου, Αναπληρώτρια Καθηγήτρια Τμ. ΜΒΣΠ, ΠαΔΑ
Παναγιώτης Καρκαζής, Αναπληρωτής Καθηγητής, Τμ. ΜΠΥ, ΠαΔΑ
Χαράλαμπος Πατρικάκης, Καθηγητής Τμ. ΗΗΜ, ΠαΔΑ

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

(Υπογραφή)

Eleni Aikaterini Leligkou

Digitally signed by Eleni Aikaterini Leligkou
Date: 2023.05.12
08:06:50 +03'00'

Ελένη Αικατερίνη Λελίγκου,
Αναπληρώτρια Καθηγήτρια ΠαΔΑ

(Υπογραφή)

Panagiotis Karkazis

Digitally signed by Panagiotis Karkazis
Date: 2023.05.12
11:11:56 +03'00'

Παναγιώτης Καρκαζής,
Αναπληρωτής Καθηγητής ΠαΔΑ

(Υπογραφή)

Digitally signed by Charalampos Patrikakis
Date: 2023.05.13 13:51:15
+03'00'

Χαράλαμπος Πατρικάκης,
Καθηγητής ΠαΔΑ

(Υπογραφή)

THEOFANIS ORFANOUDAKIS

THEOFANIS ORFANOUDAKIS
May 15 2023 3:08 PM

Θεοφάνης Ορφανουδάκης,
Αναπληρωτής Καθηγητής ΕΑΠ

(Υπογραφή)

Evangelos Pallis

Digitally signed by Evangelos Pallis
Date: 2023.05.15
11:58:38 +03'00'

Ευάγγελος Πάλλης,
Καθηγητής ΠαΔΑ

(Υπογραφή)

ANDREAS PAPADAKIS
15.05.2023 14:10

Ανδρέας Παπαδάκης,
Καθηγητής ΑΣΠΑΙΤΕ

(Υπογραφή)

Lampros Sarakis
15.05.2023 14:34

Λάμπρος Σαράκης,
Αναπληρωτής Καθηγητής ΕΚΠΑ

Ημερομηνία εξέτασης 11/04/2023

PhD THESIS

Secure resource management in Next Generation Networks (NGNs)

Michael G. Xevgenis

SUPERVISOR: Helen Catherine Leligou, Associate Professor Department of IDPE, UniWA

THREE-MEMBER ADVISORY COMMITTEE:

Helen Catherine Leligou, Associate Professor, Dep. IDPE, UniWA
Panagiotis Karkazis, Associate Professor, Dep. ICE, UniWA
Charalampos Patrikakis, Professor, Dep. EEE, UniWA

SEVEN-MEMBER EXAMINATION COMMITTEE

(Signature)

Eleni Aikaterini Leligkou

Digitally signed by Eleni Aikaterini Leligkou
Date: 2023.05.12 08:07:29 +03'00'

Helen Catherine Leligou,
Associate Professor UniWA

(Signature)

Panagiotis Karkazis

Digitally signed by Panagiotis Karkazis
Date: 2023.05.12 11:12:23 +03'00'

Panagiotis Karkazis,
Associate Professor UniWA

(Signature)

Digitally signed by Charalampos Patrikakis
Date: 2023.05.15 11:49:16 +03'00'

Charalampos Patrikakis,
Professor UniWA

(Signature)

THEOFANIS ORFANOUDAKIS

THEOFANIS ORFANOUDAKIS
May 15 2023 3:08 PM

Theofanis Orfanoudakis,
Associate Professor H.O.U

(Signature)

Evangelos Pallis

Digitally signed by Evangelos Pallis
Date: 2023.05.15 11:59:26 +03'00'

Evangelos Pallis,
Professor UniWA

(Signature)

ANDREAS PAPADAKIS
15.05.2023 14:11

Andreas Papadakis,
Professor ASPETE

(Signature)

Lampros Sarakis
15.05.2023 14:36

Lambros Sarakis,
Associate Professor NKUA

Examination Date 11/04/2023

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΔΑΚΤΟΡΙΚΗΣ ΔΙΑΤΡΙΒΗΣ

Ο/η κάτωθι υπογεγραμμένος Μιχαήλ Ξευγένης του Γεωργίου, υποψήφιος διδάκτορας του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι συγγραφέας και δικαιούχος των πνευματικών δικαιωμάτων επί της διατριβής και δεν προσβάλω τα πνευματικά δικαιώματα τρίτων. Για τη συγγραφή της διδακτορικής μου διατριβής δεν χρησιμοποίησα ολόκληρο ή μέρος έργου άλλου δημιουργού ή τις ιδέες και αντιλήψεις άλλου δημιουργού χωρίς να γίνεται αναφορά στην πηγή προέλευσης (βιβλίο, άρθρο από εφημερίδα ή περιοδικό, ιστοσελίδα κ.λπ.). Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

ΠΕΡΙΛΗΨΗ

Μία νέα εποχή στον τομέα των δικτύων έχει ξεκινήσει με την έλευση της πέμπτης γενιάς δικτύων (5G) που προσφέρουν υψηλές δικτυακές επιδόσεις για την υποστήριξη ιδιαιτέρως απαιτητικών εφαρμογών. Υψηλός ρυθμός μετάδοσης δεδομένων, ελάχιστη καθυστέρηση, υψηλή κάλυψη υπηρεσιών 5G και αυξημένη διαθεσιμότητα υπηρεσιών είναι τα βασικά στοιχεία που χαρακτηρίζουν τα δίκτυα νέας γενιάς. Τα χαρακτηριστικά αυτά είναι απόρροια των τεχνολογικών εξελίξεων τόσο σε επίπεδο υλισμικού όσο και σε επίπεδο λογισμικού. Έχοντας ως δεδομένο πλέον αυτά τα δικτυακά χαρακτηριστικά που φάνταζαν ένα άπιαστο όνειρο τα προηγούμενα χρόνια, νέες εφαρμογές άρχισαν να αναπτύσσονται για να καλύψουν διάφορες ανάγκες της ανθρωπότητας. Σκοπός των εφαρμογών αυτών αλλά και του δικτύου που τις υποστηρίζει είναι να προσφέρουν υψηλή ποιότητα υπηρεσίας και εμπειρίας στον τελικό χρήστη.

Έχοντας κατά νου πως οι εφαρμογές αυτές πρέπει να είναι διαθέσιμες στον χρήστη κατ' απαίτηση και σε παγκόσμιο επίπεδο, η δέσμευση και διάθεση των δικτυακών πόρων που τις υποστηρίζουν αποτελεί μια πρόκληση για του παρόχους δικτυακών υπηρεσιών. Είναι οικονομικά αδύνατον για έναν πάροχο να εγκαταστήσει και να προσφέρει ιδίους δικτυακούς πόρους πέμπτης γενιάς σε παγκόσμιο επίπεδο. Γι' αυτό λοιπόν οι συνεργασία των παρόχων δικτύου είναι απαραίτητη για την εξασφάλιση υπηρεσιών υψηλής ποιότητας προς τον χρήστη. Ταυτόχρονα η δυνατότητα της προσφοράς δικτυακών υπηρεσιών μέσα από νεφοϋπολογιστικά περιβάλλοντα, επιτρέπει την είσοδο νέων παρόχων δικτύου και μεταμορφώνει την υπάρχουσα αγορά. Λαμβάνοντας υπόψιν λοιπόν την ανάγκη συνεργασίας των παρόχων οι οποίοι είναι ανταγωνιστές μέσα σε αυτή την αγορά γεννώνται ερωτήματα όπως: πόσο ασφαλής μπορεί να είναι η διαδικασία διαχείρισης πόρων μεταξύ των ανταγωνιστικών παρόχων δικτύου σε ένα περιβάλλον όπου δεν υπάρχει εμπιστοσύνη μεταξύ τους; Πως θα μπορέσουμε να εγκαθιδρύσουμε ένα επίπεδο εμπιστοσύνης ανάμεσα στους παρόχους σε αυτή την ανταγωνιστική αγορά χωρίς την χρήση κεντρικοποιημένων συστημάτων; Πώς θα μπορέσουμε να ενσωματώσουμε έναν τέτοιο μηχανισμό σε ένα περιβάλλον που συνεχώς μεταβάλλεται ώστε να συνεχίζεται απρόσκοπτα και χωρίς καθυστερήσεις η διαδικασία της αυτό-οργάνωσης και αυτό-βελτιστοποίησης του δικτύου;

Η απάντηση στα ανωτέρω ερωτήματα παρουσιάζεται στην παρούσα διατριβή όπου μελετάται η εφαρμογή τεχνολογιών κατανεμημένου καθολικού, και πιο συγκεκριμένα της τεχνολογίας blockchain στη διαχείριση πόρων των δικτύων νέας γενιάς. Η τεχνολογία blockchain είναι μια από τις πλέον δημοφιλείς των τελευταίων ετών κυρίως λόγω της εφαρμογής της σε οικονομικούς τομείς. Στην παρούσα έρευνα αναλύουμε τον τρόπο λειτουργίας, τα βασικά χαρακτηριστικά, τα πλεονεκτήματα και τα μειονεκτήματα της τεχνολογίας αυτής και παρουσιάζουμε εφαρμογές της

σε τομείς της καθημερινότητάς μας πέραν των οικονομικών. Σκοπός μας είναι να αναδείξουμε τα βασικά χαρακτηριστικά του blockchain τα οποία μπορούν να επιλύσουν βασικά θέματα ασφάλειας στον τομέα της διαχείρισης πόρων των δικτύων νέας γενιάς. Κάποια από αυτά τα στοιχεία είναι: η δημιουργία ενός δικτύου εμπιστοσύνης μεταξύ των συμμετεχόντων, η χρήση αμετάβλητου κώδικα για την εκτέλεση διεργασιών διαχείρισης δικτύου κ.α. Παράλληλα, εξετάζουμε την ανάπτυξη σημαντικών εργαλείων και τεχνολογιών σχετικών με την διαχείριση των δικτύων νέας γενιάς, δίνοντας έμφαση στην προσπάθεια προτυποποίησης του ETSI με το όνομα Zero touch network and Service Management (ZSM). Αντικείμενο του ZSM είναι η διαχείριση των δικτύων νέας γενιάς τα οποία αποτελούνται από πολλούς διαφορετικούς παρόχους, με έναν αυτοματοποιημένο τρόπο που επιτρέπει την αυτό-διαχείριση και αυτό-βελτιστοποίηση των σύγχρονων δικτύων. Στόχος είναι να μειωθεί η ανθρώπινη παρέμβαση στις διεργασίες αυτές ώστε να αποφευχθούν ανθρώπινα λάθη και καθυστερήσεις. Το ZSM βασίζεται στην εφαρμογή καινοτόμων τεχνολογιών όπως η μηχανική μάθηση και η τεχνητή νοημοσύνη. Σύμφωνα με την ομάδα προτυποποίησης, υπάρχουν αρκετά σημαντικά ζητήματα ασφάλειας που παρουσιάζει αυτό το εγχείρημα τα οποία πρέπει να διευθετηθούν.

Ο συνδυασμός τεχνολογιών κατανεμημένου καθολικού και σύγχρονων εργαλείων διαχείρισης δικτύων όπως το ZSM, σκιαγραφεί και αναδεικνύει μια νέα ερευνητική περιοχή η οποία συζητείται εκτενώς στην παρούσα διατριβή και μπορεί να αποτελέσει πεδίο παραγωγής νέων καινοτόμων ερευνητικών έργων. Αντικείμενο της περιοχής αυτής είναι η δημιουργία ενός περιβάλλοντος ασφαλούς διαχείρισης δικτυακών πόρων σε δίκτυα νέας γενιάς. Παρόλο που σε θεωρητικό επίπεδο η τεχνολογία blockchain μπορεί να συνδυαστεί με εργαλεία όπως το ZSM για την διευθέτηση ζητημάτων ασφάλειας, στην έρευνά μας προχωρούμε στη διεξαγωγή πειραμάτων ώστε να εξετάσουμε αν η χρήση blockchain για την διαχείριση πόρων των δικτύων νέας γενιάς είναι δυνατή έχοντας κατά νου τις απαιτητικές προδιαγραφές των σύγχρονων δικτύων. Στο πλαίσιο αυτό τα πειράματά μας χωρίζονται σε δύο φάσεις όπου εξετάζουμε την συμπεριφορά των δικτύων blockchain όσον αφορά τον αριθμό των συναλλαγών που μπορούν να επιβεβαιωθούν το δευτερόλεπτο καθώς και τον χρόνο που χρειάζεται η επιβεβαίωση μιας συναλλαγής. Τα πειράματά μας πραγματοποιούνται πάνω σε πραγματικά δίκτυα blockchain τα οποία δημιουργούμε σε νεφοϋπολογιστικά περιβάλλοντα και στα οποία μεταβάλλουμε συγκεκριμένα χαρακτηριστικά όπως τον αριθμό των κόμβων, τη δομή του έξυπνου συμβολαίου και τον μηχανισμό συναίνεσης, ώστε να παρακολουθήσουμε την συμπεριφορά του δικτύου και να ελέγξουμε αν μπορεί να εφαρμοστεί η τεχνολογία αυτή σε σύγχρονα δίκτυα.

Τα αποτελέσματά μας είναι ενθαρρυντικά και μας ωθούν στον σχεδιασμό μια αρχιτεκτονικής όπου η τεχνολογία blockchain συνδυάζεται με το ZSM. Σκοπός της έρευνάς μας είναι η

αυτοματοποιημένη και ασφαλής διαχείριση πόρων στα σύγχρονα δίκτυα και γι’ αυτό το λόγο προχωρούμε στον καθορισμό συγκεκριμένων χαρακτηριστικών που θα πρέπει να έχει μια λύση κατανεμημένου καθολικού για να μπορεί να εξυπηρετήσει το σενάριο μας. Στη συνέχεια προχωρούμε στην εξέταση διάφορων υποσχόμενων λύσεων blockchain καθώς και στην παρουσίαση των Directed Acyclic Graphs (DAGs) που ανήκουν στις τεχνολογίες κατανεμημένου καθολικού και παρουσιάζουν αρκετά χρήσιμα χαρακτηριστικά. Στο τέλος της έρευνας αυτής συμπεραίνουμε πως οι υφιστάμενες λύσεις, είτε blockchain είτε DAG δεν πληρούν τα κριτήρια για το σενάριό μας ενώ παρουσιάζονται διάφορες ερευνητικές κατευθύνσεις που μπορούν να ακολουθηθούν για την ανάπτυξη και δημιουργία καινοτόμων λύσεων που μπορούν να συντελέσουν στην υλοποίηση ενός αυτοματοποιημένου συστήματος ασφαλούς διαχείρισης πόρων σε δίκτυα νέας γενιάς,

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: χρήση blockchain σε δίκτυα νέας γενιάς, ασφαλής διαχείριση πόρων σε δίκτυα νέας γενιάς, αυτό-διαχείριση και αυτό-βελτιστοποίηση σύγχρονων δικτύων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: blockchain, διαχείριση πόρων, τεχνολογίες κατανεμημένου καθολικού, δικτυακοί πόροι, αυτό-οργάνωση.

ABSTRACT

A new era in the computer networks has begun with the advent of 5G which promises to support demanding applications by facilitating high-performance numbers. Ultra-high data rates, high availability, ultra-low latency and wide area coverage are the main characteristics of modern networks which aim to ensure the high QoE level of the end-user. The Network Providers (NPs) are responsible for the deployment and maintenance of modern networks' infrastructure which consists of novel hardware devices and software components developed in the last few years under the umbrella of 5G. As the infrastructure becomes more sophisticated and expensive, the NPs seek to maximize the efficiency of its utilization by flexibly sharing resources. Therefore, NPs should cooperate and perform resource management processes in this multivendor ecosystem to guarantee the network requirements needed for the support of demanding applications. This introduces the potential of formation of a new marketplace where competitive NPs (with no established trust) trade resources. Hence, several security related questions arise such as: how secure could be this marketplace that consists of various competitive NPs since no one trusts each other? How could we form a network of trust among participants in this vast competitive market without the existence of a trusted third party in order to avoid the drawbacks of centralized approaches? How can we achieve the desired level of security in networks with highly automated management functionalities?

In this thesis, we answer to these questions by studying the use of emerging decentralized technologies in modern networks in order to facilitate the secure resource management in Next Generation Networks (NGNs). To this end, a comprehensive study of blockchain technology takes place, where we examine the characteristics of this hyped technology, and we discuss its main advantages and disadvantages. Furthermore, we discuss the major advancements of modern networks focusing on the software developments (i.e., MANOs, VNFs) and we present the standardization effort of the ETSI with the Zero touch networks and Service Management (ZSM) framework. ZSM aims to enable self-management and self-optimization of modern multivendor networks. The goal is to reduce or even eliminate the human intervention in the network management processes of modern networks in order to avoid human errors and delays. However, this framework presents significant security issues mentioned by the standardization team and examined in this study. In this framework, we explored whether blockchain technology can be the answer to the ZSM's security issues which automatically unveils a novel research area where the blockchain technology and the ZSM framework are combined to provide secure and dynamic resource management in NGNs.

To check the feasibility of implementing a blockchain solution to enable NPs share resources, we designed and implemented such an approach. Apart from the qualitative assessment, to check the performance of the blockchain network, we perform experiments in real testbeds by implementing a private/consortium permissioned blockchain network. Our experiments are conducted in two phases in order to check how the network behaves when certain parameters are changing (number of nodes, SC's structure, consensus mechanism). The metrics we focus on are: the network's throughput (number of transactions per second), the latency (time needed for transaction's validation) and success rate (transactions that have been successfully validated). The results of our experiments are encouraging and motivate us to continue our study and design the architecture of a novel blockchain-based ZSM approach. This architecture presents how blockchain can be integrated with ZSM and address the security issues highlighted by the ZSM standardization team. To provide guidance to prospective adopters of the proposed approach, we identify the most suitable blockchain solution for this particular use case and we also proceed to the definition of certain requirements that the Distributed Ledger Technology (DLT) should fulfil so that emerging solutions can be easily assessed. These requirements draw the profile of the ideal DLT that can be adopted to implement the secure resource management scenario and harness the benefits of both DLTs and ZSM framework without jeopardizing the proper operation of NGNs. Therefore, beyond the study of blockchain technology we examine the Directed Acyclic Graphs which is another promising DLT that presents high performance numbers and is considered more scalable than typical blockchains. At the end of this thesis, we performed a qualitative assessment of existing blockchain and DAG solutions in order to compare them with the characteristics that the ideal solution should present. The result of this assessment showed that none of the existing solutions is the ideal one although each one of them contains at least one valuable characteristic.

SUBJECT AREA: blockchain in next generation networks, secure resource management in next generation networks, self-management and self-optimization of next generation networks

KEYWORDS: blockchain, distributed ledger technologies, next generation networks, zero touch and service management, smart contracts, consensus, virtual network functions

In the loving memory of my father.

ΕΥΧΑΡΙΣΤΙΕΣ

First and foremost, I would like to express my gratitude and appreciation to my supervisor Professor Eleni Aikaterini Leligkou. Her inspiring words, her guidance and kindness helped me and motivated me to finish my PhD studies. Moreover, I would like to thank her for trusting me to perform lectures in her class. This experience was extremely valuable to me.

Additionally, I would like to thank Professor Panagiotis Karkazis for assisting me with his knowledge in the area of modern networks and providing me with useful information regarding the advancements in next generation networks. His accurate comments throughout this study helped me to stay on track and find answers to difficult questions.

I would like also to thank Professor Charalampos Patrikakis, who has inspired me all these years with his novel ideas and research skills. Since I was a student, he was always available and willing to provide guidance when needed.

Special thanks to my dear friend Dr. Dimitrios Kogias for his assistance and guidance when I was studying the distributed ledger technologies and more specific the blockchain. Dimitris was always eager to provide answers to my questions regarding the DLTs and their applicability in various sectors. His comments and suggestions were very useful and increased the quality of my research.

Also, I would like to thank my dear friend Ioannis Christidis for assisting me with his extremely valuable programming skills. His ability to find alternatives and smart programming solutions helped me to overcome many obstacles.

Finally, I would like to thank my family; my fiancée Efi, my brother Christos and my mother Maria for supporting me throughout my studies and for encouraging me all these years. In addition, this study could not be completed without the company of my dear dog Ektoras. Thank you for your company and support my best friend!

ΛΙΣΤΑ ΔΗΜΟΣΙΕΥΣΕΩΝ / List of Publications

- Xevgenis, M.; Kogias, D.G.; Karkazis, P.A.; Leligou, H.C. Addressing ZSM Security Issues with Blockchain Technology. Future Internet 2023, 15, 129. https://doi.org/10.3390/fi15040129
- Xevgenis, M., Kogias, D., Christidis, I., Patrikakis, C., & Leligou, H. C. (2022). Evaluation of a blockchain-enabled resource management mechanism for NGNs. International Journal of Network Security & Its Applications (IJNSA) Vol.13, No.5, September 2021, DOI: 10.5121/ijnsa.2021.13501.
- Xevgenis, M., Kogias, D. G., Karkazis, P., Leligou, H. C., & Patrikakis, C. (2020). Application of blockchain technology in dynamic resource management of next generation networks. Information, 11(12), 570, DOI: 10.3390/info11120570.
- Xevgenis, M. G., Kogias, D., Leligou, H. C., Chatzigeorgiou, C., Feidakis, M., & Patrikakis, C. Z. (2020, May). A Survey on the Available Blockchain Platforms and Protocols for Supply Chain Management. In IOT4SAFE@ ESWC.
- Kogias, D. G., Leligou, H. C., Xevgenis, M., Polychronaki, M., Katsadouros, E., Loukas, G., ... & Patrikakis, C. Z. (2019). Toward a blockchain-enabled crowdsourcing platform. IT Professional, 21(5), 18-25.
- Maria Polychronaki, Nick Kaftantzis, Michael Xevgenis, Dimitrios Kogias, and Nelly Leligou, “(Demystifying) Blockchain Development: The BLER Use Case”, CUTTER BUSINESS TECHNOLOGY JOURNAL, 2019, Vol 32, No. 10, pp. 26-31.

Table of Contents

LIST OF FIGURES 27

LIST OF TABLES........ 29

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ........ 31

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ........ 33

1. INTRODUCTION 37

1.1 Hardware and Software improvements for the development of 5G 38

1.1.1 The growth of virtualization and cloud computing........ 39

1.1.2 Software Defined Networks and Network Function Virtualization 41

1.1.3 Network Slicing in 5G 42

1.1.4 Management and Orchestration (MANO) tools in modern networks 43

1.2 Challenges and limitations of 5G networks........47

1.3 Blockchain fundamentals 49

1.3.1 Types of blockchain........ 52

1.3.2 Advantages and disadvantages of blockchain technology 56

1.4 Problem statement, identification of our research area and structure of thesis........58

2. STATE OF THE ART AND RELATED WORK........ 63

2.1 The rise of Blockchain technology........63

2.1.1 The Bitcoin, the Ethereum and the role of Smart Contract (SC)........ 64

2.1.2 Consensus mechanisms........ 67

2.1.3 Indicative blockchain application sectors 70

2.1.4 Blockchain in NGNs........ 79

2.2 The Zero Touch Network and Service Management (ZSM) framework for NGNs........87

2.2.1 The role of AI/ML in the ZSM concept implementation 87

2.2.2 Basic principles of ZSM architecture........ 89

2.2.3 Functional and Non-Functional requirements of ZSM architecture........ 91

2.2.4 Security requirements of ZSM architecture 94

2.2.5 The ZSM reference architecture 95

2.2.6 Open security issues of ZSM framework 98

3. DESIGN, IMPLEMENTATION AND EVALUATION OF A BLOCKCHAIN-BASED APPLICATION IN DYNAMIC RESOURCE MANAGEMENT OF NGNS 103

3.1 Application of blockchain technology in dynamic resource management of NGNs 103

3.1.1 System architecture and use case scenario 104

3.1.2 The experimental testbed and the evaluation results of the blockchain-based solution 108

3.1.3 Conclusions of our initial attempt in developing a blockchain based resource management mechanism for modern networks 113

3.2 Development and re-evaluation of a blockchain-enabled resource management mechanism for NGNs 114

3.2.1 SC version 2: Analysis of its structure and functionality 115

3.2.2 Candidate consensus algorithms 118

3.2.3 Description of the testbed and experiment methodology 119

3.2.4 Conclusions of the evaluation of the blockchain-enabled resource management mechanism using different consensus algorithms and modified SC structure 126

4. A BLOCKCHAIN-BASED ZSM APPROACH 129

4.1 Exploiting blockchain to address ZSM's security issues: Analysis and architecture overview 129

4.2 Discussion regarding the use of blockchain in ZSM 135

4.3 Guidelines for DLT solution design and deployment in ZSM scenario 138

4.3.1 Requirements of the blockchain-based ZSM approach and definition of the characteristics of the ideal DLT solution 138

4.3.2 Distributed Ledger Technologies: Blockchain vs DAG 142

4.3.3 Candidate blockchain solutions for the blockchain-based ZSM scenario 144

4.3.4 Candidate DAG solutions for the DLT-based ZSM scenario 148

4.3.5 Modifications of the discussed blockchain and DAG solutions to fit to our scenario 151

5. CONCLUSIONS AND FUTURE WORK 155

5.1 Conclusions 155

5.2 Future work 161

6. REFERENCES 165

APPENDIX A 173

APPENDIX B 191

List of Figures

Figure 1. NVF reference architectural framework [24] 45

Figure 2. 5G ecosystem 47

Figure 3. Blockchain overview [35] 52

Figure 4. Types of blockchain networks [40]...56

Figure 5. Identification of research area 59

Figure 6. Deployment of SC in the Ethereum network..66

Figure 7. Finding the PoW solution 68

Figure 8. The BLER platform architecture [50] 71

Figure 9. SCM's circle with stages and benefits/possibilities from using blockchain [42] 74

Figure 10. ZSM reference architecture [91]...97

Figure 11. The proposed system architecture 105

Figure 12. The experimental testbed..110

Figure 13. The throughput per SC function. 112

Figure 14. The latency per SC function for different transaction loads. 112

Figure 15.Testbed overview 120

Figure 16. Success rate vs ITR for Raft and IBFT 121

Figure 17. Throughput vs ITR in Raft testbed...122

Figure 18. Throughput vs ITR in IBFT testbed 122

Figure 19. Throughput vs ITR focusing in requestResources function 123

Figure 20. Latency vs ITR in Raft testbed 124

Figure 21. Latency vs ITR in IBFT testbed 124

Figure 22. Latency vs ITR in the requestResources function 125

Figure 23.The architecture of the blockchain-based ZSM 131

Figure 24. Lifecycle of a blockchain enabled ZSM scenario 134

Figure 25. Blockchain vs DAG [122] 143

List of Tables

Table 1. Functional requirements of ZSM architecture 92

Table 2. Non-functional requirements of ZSM architecture 94

Table 3. Solutions to security issues: Current ZSM vs Blockchain-based ZSM 136

Table 4. Blockchain vs Directed Acyclic Graphs (DAGs) 144

Table 5. Suitability of the examined blockchain and DAG solutions 152

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	**Ελληνικός Όρος**
Network resources	Δικτυακοί πόροι
Virtualization	Εικονικοποίηση
Immutability	Μη μεταβλητότητα
Software Defined Networks	Δίκτυα καθορισμένα από λογισμικό
High automation	Υψηλά επίπεδα αυτοματοποίησης
Distributed Ledger Technologies	Τεχνολογίες Κατανεμημένου Καθολικού
Smart Contract	Έξυπνο συμβόλαιο
Consensus	Συναίνεση

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

3GPP	3rd Generation Partnership Project
4G	Fourth Generation of internet
5G	Fifth Generation of internet
AI	Artificial Intelligence
AIRO	Artificial Intelligence based Resource aware Orchestration
API	Application Programming Interface
AR	Augmented Reality
BC	Blockchain
BFT	Byzantine Fault Tolerant
BTC	Bitcoin
CAPEX	Capital Expenditure
CDC	Content Delivery Contract
CIA	Confidentiality Integrity Availability
CLA	Closed Loop Automation
CP	Content Provider
CSC	Cross Service Communication
DAG	Directed Acyclic Graph
DApps	Distributes Applications
DBB	distributed blockchain-based broker
DC	Data Center
DDoS	Distributed Denial of Service
DDPG	Deep Deterministic Policy Gradient
DFF	Discouraging Free-riding and False-Reporting
DLT	Distributed Ledger Technologies
DRL	Deep RL
E2E	End-to-End
EC	Edge Cloud
EFF	Eliminating Free-riding and False-Reporting
eMBB	enhanced Mobile Broadband
ENI	Experiential Network Intelligence

EQF	European Qualifications Framework
ETSI	European Telecommunication Standards Institute
EVM	Ethereum Virtual Machine
FL	Federated Learning
FPC	Fast Probabilistic Consensus
GDPR	General Data Protection Regulation
GTIN	Global Trade Item Number
HLF	Hyperledger Fabric
HR	Human Resources
IaaS	Infrastructure as a Service
IBFT	Istanbul Byzantine Fault Tolerant
IBFT	Istanbul Byzantine Fault Tolerant
ILP	Inter Ledger Protocol
IoT	Internet of Things
IP	Internet Protocol
IPFS	Inter Planetary File System
ISC	IOTA Smart Contract
ITR	Input Transaction Rate
JVM	Java Virtual Machine
KPIs	Key Performance Indicators
MANO	Management and Operation
MD	Management Domain
MEC	Mobile Edge Computing
MIMO	Multiple Input Multiple Output
mIoT	massive Internet of Things
ML	Machine Learning
MNOs	Mobile Network Operators
MnS	Management Services
MoUs	Memorandums of Understanding
NBI	Northbound Interface
NESAS	Network Equipment Security Assurance Scheme
NF	Network Function
NFT	Non-Fungible Token

NFV	Network Function Virtualization
NFVI	Network Function Virtualization Infrastructure
NFVO	Network Function Virtualization Orchestrator
NGNs	Next Generation Networks
NP	Network Provider
NS	Network Service
NSB	Network Slice Broker
NSM	Network Service Marketplace
NTP	Network Time Protocol
OPEX	Operational Expenditure
OS	Operating System
OSS/BSS	Operational Support Systems and Business Support System
P2P	Peer to Peer
PaaS	Platform as a Service
pBFT	practical Byzantine Fault Tolerant
PoF	Proof of Formulation
PoS	Proof of Stake
PoW	Proof of Work
QoE	Quality of Experience
QoS	Quality of Service
RL	Reinforcement Learning
SaaS	Software as a Service
SBC	Single Board Computer
SC	Smart Contract
SCM	Supply Chain Management
SDNs	Software Defined Networks
SECAM	Security Assurance Methodology
SFC	Service Function Chaining
SLA	Service Level Agreement
SNs	Social Networks
SPoF	Single Point of Failure
SUT	System Under Testing
TE	Technical Enabler

TPS	Transaction Per Second
UPC	Universal Product Code
uRLLC	ultra-Reliable Low Latency Communications
V2X	Vehicle to X
VIM	Virtualized Infrastructure Manager
VM	Virtual Machine
VMAF	Virtual MEC Application Functions
VNF	Virtual Network Function
VNFM	Virtual Network Function Manager
ZSM	Zero touch Service Management

1. Introduction

A new era in the computer networks science has begun with advancements both in hardware and software. In the last few years, we have experienced the transition from 4G networks to 5G while currently researchers are focusing on the development of more sophisticated network structures beyond 5G, characterized as Next Generation Networks (NGNs). Nowadays 5G, offers high network performance characteristics, such as ultra-low latency, high throughput, wide area coverage and high service availability. These characteristics are extremely beneficial for several sectors of our lives such as, healthcare, industry, entertainment, and others. The ability to support demanding applications led the way for the development of solutions that increase the quality of our life by overcoming obstacles and limitations previously considered unbeatable.

To provide these characteristics, 5G should fulfill specific requirements such as:

- Support an increased number of connected devices up to 100 times higher than its predecessor 4G [1, 2].
- Support a mobile data volume per area 1000 times higher than 4G [1, 2].
- Offer increased data rate up to 100 times higher than the previous network generation [1-3].
- Reduce the end-to-end latency, reaching 5ms [4].
- Guarantee approximately 100% availability [4].
- Provide 5G capabilities globally and achieve close to 100% geographical coverage [4-6].
- Offer increased levels of security and privacy [3].
- Decrease the energy consumption in low levels by reaching 10 times less than 4G [1, 2].
- Support real-time processing and transmission.
- Easy integration with current wireless infrastructures and technologies.
- Increase the flexibility, intelligence, dynamicity, and openness of the network.
- Cost-efficient in terms of CAPital and OPerational EXpenditures (CAPEX, OPEX).

Having in mind the characteristics of 5G networks, use cases where advanced networks are used in a very beneficiary manner are discussed. In healthcare sector, the development of critical applications such as remote surgery is now feasible. 5G's main characteristics such as ultra-low latency and high reliability, allow us to implement the scenario of performing surgery while the doctor can be miles away from the patient as it is presented in [7,8]. These solutions directly affect the quality of life of the mankind by leveraging the well-being of humans in every corner of the earth. People who live in distant settlements could benefit the most from these lifesaving solutions in emergency situations, where the time to reach a hospital can be proved fatal. Beyond the remote surgery application, doctors and health organizations are capable of accessing critical data from patients who are in danger and proceed in time to the necessary actions without extra time-consuming procedures, thanks to the reliability, robustness and high data rates of modern networks.

The advent of 5G has a significant impact on the industry sector also as it accelerates the evolution towards Industry 4.0 [9]. Hyped technologies such as Internet of Things (IoT), Artificial Intelligence (AI) and Augmented Reality (AR) in combination with 5G reshape the industry environment. As it is stated in [10,11] automation in factories, warehouses, businesses and other organizations begins with 5G. As the technology expands, so does industrial automation, enabling organizations to deploy customized systems that will solve their production and business problems in real time. The role of 5G in Industry 4.0 is to facilitate that the proper network conditions are met and maintain the communication layer over which IoT, AI and other technologies communicate for the development of the Factories of the Future [12,13].

1.1 Hardware and Software improvements for the development of 5G

The development of 5G networks was not an easy task and required improvements in both hardware and software. Focusing on hardware components, many of them were used also in 4G networks but in 5G there are three major differentiators: the massive multiple-input multiple-output (MIMO) systems, the integrated radio, and edge computing [14]. Massive MIMOs are the evolution of the well-known MIMO antennas and provide greater network capacity and improved coverage in contrast to their predecessors. Nevertheless, massive MIMO requires computational power which means that they are power consuming. Moreover, the high data rates offered by 5G are available via the ultra-high frequency band used in the network. This band is in the millimeter frequency area which is not the preferred one when we are dealing with large distance communications. To overcome this issue the integrated radio unit was developed. This unit is a device that includes a 5G antenna, radio and a digital unit that is very easy to be installed. As a result, carriers are able to install multiple radio units within locations that need 5G millimeter wave coverage. Finally, the edge computing, is the attempt to bring computing resources closer to the end user in order to minimize the latency and increase the service coverage [15]. This is achieved with the installation of edge computing devices, which are practically high-performance computing devices in places near to the user. For example, a streaming service inside a stadium could be supported via an edge computing infrastructure deployed to provide the service to customers inside the building.

Major improvements were also made in the software of modern networks with the introduction of new technologies such as Software Defined Networks (SDNs) and Network Function Virtualization (NFV). The keystone technologies used for the development of SDNs and NFVs is the virtualization and cloud computing.

1.1.1 The growth of virtualization and cloud computing

Virtualization is the logical abstraction of the physical resources and is used extensively today. When we refer to resources, we mean the amount of memory and processors, the amount of storage capacity as well as the amount of networking elements. In a virtualized environment we allocate resources according to our needs. This means that we do not have to restrain all the available resources to operate for one purpose. For example, if we use a physical server which consists of a large amount of processing power and memory and we want to operate a web server, which do not require all of our resources, we are able to dedicate the amount of resources that are needed and the rest unused resources can be reserved for other purposes. As a result, we avoid the underutilization of resources. Virtualization can be implemented at all the aforementioned types of resources of the system. Therefore, we may use processor virtualization, memory virtualization, storage virtualization and network virtualization. When processor virtualization is used, a processor can be shared across several application instances. In memory virtualization, the memory resources are aggregated into a pool of single memory which is managed by multiple applications. The storage virtualization can be divided into block virtualization (storage of data at the device level) or file virtualization (at the file level). When network virtualization is used, virtual IP management and segmentation are supported. The use of virtualization technology in legacy datacenters led to their evolution towards modern cloud computing infrastructures which automatically increased the utilization of resources, reduced the operating costs, enabled server consolidation, increased the uptime, and allowed faster disaster recovery [16].

The development of cloud computing which is one of the main technologies that supports modern networks was based on virtualization. The cloud infrastructures are able to provide their clients with various services. These services form the cloud service pyramid model and are classified into three main categories: the Infrastructure as a Service (IaaS), the Platform as a Service (PaaS) and the Software as a Service (SaaS). A service is characterized as IaaS when the user of the cloud is able to provision resources such as processing, memory, networks, storage and is able to deploy and operate arbitrary software (operating system and applications). It is worth mentioning that the control and manageability of the user is restrained strictly on the OS, the storage and the applications that he has acquired and has no control to the underlying cloud infrastructure. However, when a user has only the ability to deploy onto the cloud applications, programming tools, libraries and services, and has no control over the underlying infrastructure then the service is characterized as PaaS. In a PaaS model the user is capable to control and possibly configure settings for the application-hosting environment. In the SaaS model, the user of the cloud is able to use the provider's applications running on a cloud infrastructure. These applications are accessible via various client devices such as a web browser or a program interface. However, the user has no control over the underlying infrastructure like servers, OS, storage, networks, or even individual application capabilities, with a possible exception of limited user specific application configuration settings. [16]

Relevant to the cloud deployment there are four deployment models used. The private cloud, the community cloud, the public cloud, and the hybrid cloud. In the private cloud model, the cloud infrastructure is used exclusively by one company or organization and its members. In the community cloud, the infrastructure is used exclusively by the community and its members. The cloud infrastructure may be managed and controlled by a number of organizations which belong to the community. In the public cloud model, the cloud infrastructure can be used by the general public. The cloud infrastructure may be controlled and managed by government organizations or companies. Finally, in the hybrid cloud, the cloud infrastructure is a composition of two or more distinct cloud infrastructures that remain unique entities but are bound together by standardized or proprietary technology which enables data and application portability. In the last few decades cloud computing infrastructures were used to be developed in the core of the network and provided resources for the implementation of numerous software solutions. In 5G networks, small cloud computing infrastructures are being developed at the edges of the network, closer to the end user in order to achieve ultra-low latency which is one of the main characteristics of 5G. The development of cloud at the edge of the network is called edge or fog computing and is one of the main technologies that allowed modern networks to provide high performance network metrics. [16]

1.1.2 Software Defined Networks and Network Function Virtualization

The rapid growth of virtualization and cloud computing had a major impact on the networking sector as new technologies came up. It is known that a network device consists of a data plane and a control plane. The data plane is often a switch connecting various network ports on a device and a control plane is the brain of a device. An idea of a logically distributed control plane was born which led to the development of Software-Defined Networking (SDN). SDNs exploited the virtualization technology characteristics and achieved the decoupling of control plane and data plane. The early steps of the SDN development took place at Stanford University where the OpenFlow protocol was created. OpenFlow was designed for a number of devices containing only data planes to respond to commands sent to them from a logically centralized controller which housed the single control plane for that network. According to Thomas D. Nadeau and Ken Gray in their book SDN: Software Defined Networks "An Authoritative Review of Network Programmability Technologies" the SDNs are defined as "an architectural approach that optimizes and simplifies network operations by more closely binding the interaction (i.e., provisioning, messaging, and alarming) among applications and network services and devices, whether they are real or virtualized. It is often achieved by employing a point of logically centralized network control- which is often realized as an SDN controller- which then orchestrates, mediates, and facilitates communication between applications wishing to interact with network elements and network elements wishing to convey information to those applications. The controller then exposes and abstracts network functions and operations via modern, application-friendly and bidirectional programmatic interfaces" [17]. The development of SDNs was a breakthrough in the computer networks field as it introduced centralized network provisioning, holistic enterprise management, more granular security, lower operating costs, reduced capital expenditure and cloud abstraction. The ability to host and manage various SDN controllers in the form of Virtual Machine (VM) in cloud environments gave the opportunity to network administrators to embrace the advantages of cloud technology and improve the manageability and performance of network structures.

At the same time the idea of describing and implementing network functionalities using programming languages was gaining ground. Functions such as routing and firewalling which were previously implemented by hardware networking devices now could be implemented using code. Having in mind the rapid growth of virtualization, cloud computing and SDN, the hosting of network functions in the form of code in virtual environments such as VMs was presented. This resulted to the birth of Network Function Virtualization (NFV) and the creation of Virtual Network Functions (VNFs). NFV allows network operators and service providers to implement network functions in software, leveraging standard servers and virtualization technologies, instead of run on purpose-built hardware [18]. The development of NFV and SDN increased the manageability of modern networks, decreased their CAPEX and OPEX, increased their flexibility and network resource utilization and increased the network performance metrics. Focusing on the software advances in modern networks, the use of cloud and edge computing along with the development of SDN and NFVs gave an extra boost to facilitate the 5G network requirements. VNFs can be implemented in every corner of the network to support demanding applications thanks to the presence of cloud infrastructures. When a cloud infrastructure is used for hosting VNFs it is characterized as a NFV Infrastructure (NFVI) point of presence, that deploys VMs based on VNF instances. In modern cloud environments VNFs can be implemented also in the form of containers which are less demanding virtual environments in terms of resources comparing to VMs and can be deployed faster than typical virtual instances.

1.1.3 Network Slicing in 5G

The ability to operate Network Functions (NFs) in the form of VNFs in virtual instances where the reserved resources (compute, memory) are controlled, urged researchers to take advantage of the virtualization technology and create virtual network slices to support numerous use cases. These slices have been defined by 3GPP as “a logical network that provides specific network capabilities and network characteristics” [19]. A network slice is implemented by a slice instance, which consists of NFs and their corresponding computing, storage, and networking resources. The description of the structure and the configuration details of a slice are captured in the so-called network slice template. Essentially, with network slicing a Network Provider (NP) can deploy multiple logical networks over the same physical infrastructure. Network slices use VNFs to provide the ideal network environment for each use case. A network slice consists of one or many VNFs which are chained together in order to meet predefined requirements for specific applications. The chaining of network services is called Service Function Chaining (SFC) which is one of the most interesting research domains in the computer networks field since it affects the network performance of the slice and the overall operation of the application [20,21]. This results to the formation of many different network slices implemented in the same hardware which can be used to support different use cases. Having in mind that modern networks must support several vertical industries, the "one size fits all" approach is not acceptable. 5G networks support applications such as smart factories, remote surgery, autonomous driving, which belong to different verticals with different requirements and key performance indicators (KPIs). Currently 3GPP has grouped all application verticals in four categories namely: enhanced Mobile Broadband (eMBB), Critical Communications, massive Internet of Things (mIoT), Vehicle to X (V2X) communications. In eMBB the very high data rate is the main priority while in critical communications, characteristics such as low latency and ultra-high reliability are extremely important. In mIoT, the ability to establish massive numbers of connections in environments with high user density is vital while in V2X use cases there is a need for high reliability, low latency, high speed, and high positioning accuracy. [22]

1.1.4 Management and Orchestration (MANO) tools in modern networks

Having discussed the development of technologies such as cloud computing, SDN, VNFs and network slicing, we proceed to the presentation of management and orchestration frameworks used for the proper functionality of 5G networks. These components are called Management and Orchestration (MANO) and as their name denotes, they are responsible for the management and orchestration of network slices and VNFs. According to authors in [23], MANO frameworks can be considered to be a management and orchestration suite for physical and virtual resources related to the life cycle of the deployed Network Service (NS). MANOs interact with cloud infrastructures owned by Network Providers and manage their resources to deploy and manage the requested NSs. The cloud infrastructures managed by MANOs are the NFVI points of presence, which can be owned by different Network Providers (NPs). These NFVIs may differ as every NP is free to build its own infrastructure by using the tools of its preference. As a result, modern networks are characterized by heterogenicity. Therefore, MANOs should be interoperable by providing standard software connections to interact with various cloud technologies. The majority of cloud software used for the implementation and management of clouds, provides these software connections usually in the form of Application Programming Interfaces (APIs) in order to easily communicate with MANOs and other external entities.

The development and design of MANOs has triggered the interest of the research community and Network Providers. More specific, the ETSI framework is open source and has been developed by the ETSI ISG NFV. Several frameworks have been created based on ETSI-NFV standards having in mind the NFV reference architectural framework as it is depicted in Figure 1. NVF reference architectural framework [24] [24]. On the bottom left part of the figure, the NFVI is presented which consists of the virtualized resources of the infrastructure (compute, storage, network) used for the creation of VNFs. Above the VNFs, the Element Management System is placed which is responsible for performing typical management functionality for one or more VNFs. Additionally, on the upper part of the figure the component called “Service, VNF and Infrastructure descriptions” is depicted that provides information regarding the VNF deployment template, the VNF Forwarding Graph, and other infrastructure and service-related functionalities. On upper left part of Figure 1. NVF reference architectural framework [24] the Operational Support Systems and Business Support System (OSS/BSS) of the provider are presented. Finally, on the right side of the figure, the Management and Orchestration mechanism is depicted, that consists of three main elements: the NFV orchestrator, the VNF Manager and the Virtualized Infrastructure Manager (VIM). The NFV orchestrator is responsible for the management and orchestration of NSs deployed on the NFVIs while the VNF manager is responsible for the life cycle management of one or many VNFs. VIM is responsible for the coordination of functionalities used for the interaction and management of virtualized resources, such as compute, storage, and network, reserved to support VNFs. Multiple VIMs instances may be deployed, one per different type of NFVI technology.

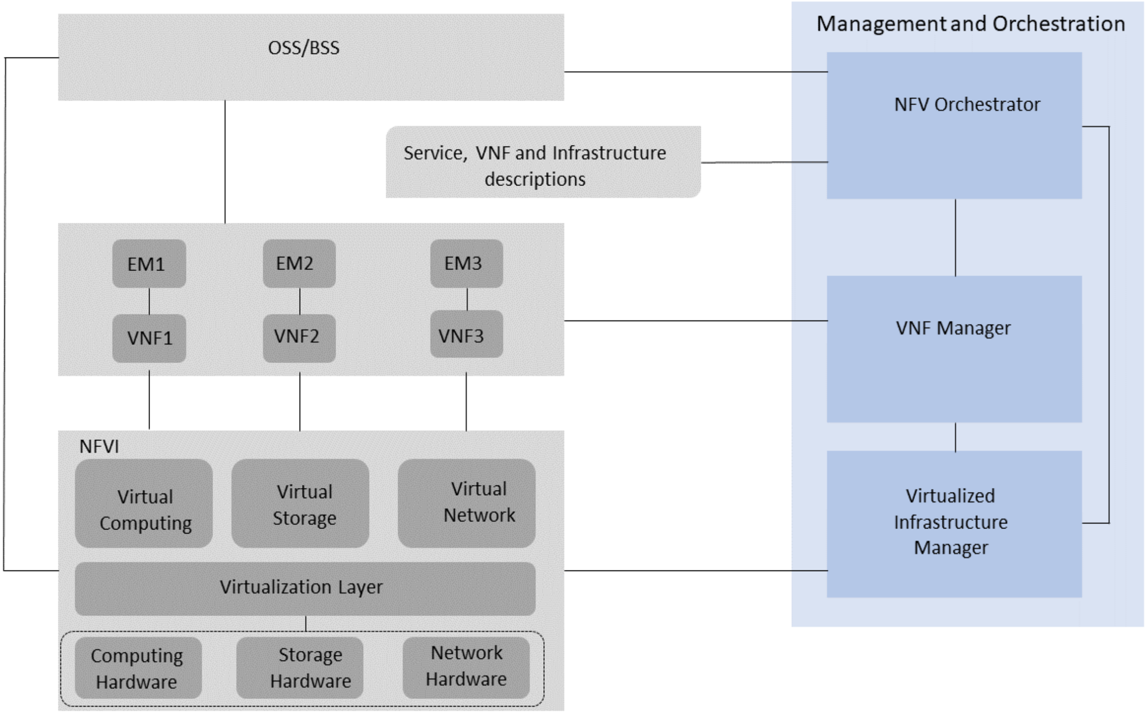

Figure 1. NVF reference architectural framework [24]

In the current research we survey three well known MANOs to better understand the flexibilities and functionalities they offer: the Open-Source MANO (OSM – MANO), SONATA and Cloudify.

OSM MANO [25] is an open-source framework created under the umbrella of ETSI and provides a Management and Orchestration stack for the development of commercial NFV applications. The goal of ETSI OSM is the creation of a community-driven production-quality end to end Network Service Orchestrator (E2E NSO) for telecommunication services, capable of modelling and automating real telco-grade services, with all the intrinsic complexity of production environments. OSM boosts the rapid development of NFV technologies and standards and enables a broad ecosystem of VNF vendors. The OSM key aspects allow its rapid and easy integration. These aspects are: the Information Model aligned with ETSI NFV, the unified Northbound Interface (NBI), the extended concept of Network Service in OSM and the fact that OSM can manage the lifecycle of Network Slices. [23,26]

SONATA [27] is another open-source MANO solution developed by the 5GTANGO project [28] and is aligned with the ETSI NFV. SONATA provides VNF management as well as resource and service orchestration. Moreover, it is customizable as it includes swappable modular plugins, such as life-cycle management, service monitoring, conflict resolution, network slice management, policies enforcement and run-time Service-Level Agreement (SLA) contracts. Also, SONATA offers open interfaces for supporting multi-vendor scenarios, independent of the supporting orchestration stacks. [23]

Cloudify [29] is also an open-source cloud-based orchestration platform, that provides a commercial release targeting vendors. The orchestrator of this solution is stable and adopted in many production environments. Cloudify uses a powerful core engine which manages the life cycle of the services across many cloud environments. This characteristic along with the fact that it supports many plugins that make integration with other platforms easier, makes Cloudify extremely attractive to the community. Additionally, it designs and deploys NSs based on a descriptive language that can be considered to be a NFVO as well as a VNFM under the perspective of ETSI-NFV architecture. [23]

Concluding the analysis of the major hardware and software advancements which led to the development of 5G, we proceed to the presentation of the 5G ecosystem in Figure 2. 5G ecosystem, emphasizing on the key technologies discussed previously. At the hardware layer, the use of powerful MIMO antennas improved the wireless propagation values and guaranteed 5G characteristics such as high throughput, wide area coverage and low latency. At the same time one of the most hyped technologies, the cloud computing was used to support software defined network structures and programmable network functions. In order to facilitate the promised 5G metrics, small cloud infrastructures were implemented at the edge of the network, near to the end user. These cloud environments support technologies such as network function virtualization by hosting virtual network functions and therefore they were also called NFV Infrastructures (NFVIs). NFVs and Network Slices are managed by management and orchestration (MANO) frameworks in 5G networks which interact with many different cloud infrastructures. The combination of these emerging technologies guarantee that the 5G network requirements are met and support many different demanding use cases which belong to different industry verticals. Considering that cloud infrastructures are playing a crucial role in 5G operation by supporting various network services, it is safe to come to the concussion that new Network Providers are entering the telco market. In contrast to legacy network structures where only certain organizations and companies could become NPs by installing their hardware and software infrastructure, in modern networks every owner of a cloud environment can possibly become a network provider.

Therefore, the legacy NPs' marketplace is reshaped and a new one is formed where the number of providers can be significantly increased. Having in mind the benefits of combining these hyped technologies, the research community proceed to the design and development of more sophisticated network structures called as Next Generation Networks (NGNs).

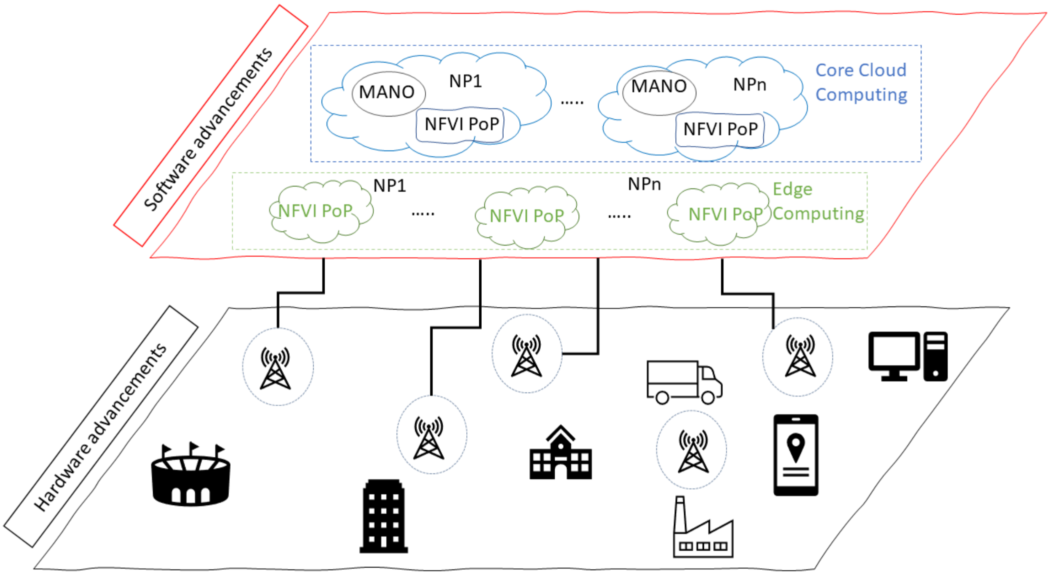

Figure 2. 5G ecosystem

1.2 Challenges and limitations of 5G networks

5G promises to offer high quality network services to every corner of the earth to support demanding applications for several use cases. To achieve that, 5G infrastructure must be implemented globally which is one of the biggest challenges, while NPs must cooperate in order to fulfill the clients' requests and at the same time maintain the cost in reasonably levels. The deployment of 5G infrastructure cannot be accomplished by one NP, nor the management of this vast ecosystem. The network must be divided into management domains which are the responsibility/administrative areas of the NP. An NP can be responsible for more than one domain and should be able to interact with other domains which may belong to different NP. Similar to legacy networks where the user of the network could use functionalities implemented by one or many NPs in case of roaming, in the 5G scenario the user should be able to access the 5G network in a provider-agnostic manner. Therefore, NPs should cooperate to deploy and manage the 5G network globally and offer the desired Quality of Service (QoS) levels to the end-user.

Furthermore, the increased number of NPs and the formation of a new competitive marketplace makes trust among participants a challenge. Modern networks demand the cooperation of NPs which should be able to easily trade or lease resources to support network services. In other words, it would be highly beneficial for all networking resource providers to be able to dynamically lend/borrow resources, instead of currently employed semi-static coarse Service Level Agreements (SLAs). However, this is difficult to achieve in an untrusted environment at the absence of a trusted 3rd party.

To clarify the situation, we present an example of such collaboration. NP1 and NP2 are the network providers of our scenario and each one is responsible and owns a management domain with specific resources (virtual resources, VNFs, Network Slices etc.). The end-user of NP1 took a flight and arrived at the airport that is out of the reach of NP1 and inside the management area of NP2. The end-user wants to access a streaming service with specific QoS parameters that is guaranteed by the 5G network. Since NP1 will not be able to support the end-user's request, a collaboration with NP2 takes place. NP1 asks NP2 to fulfill the request of the end-user with the predefined QoS level and NP2 sets the price for this action. When these two entities agree they sign an SLA where the terms of their transaction are described. Then NP2 fulfills the request of the end-user on behalf of NP1. But what happens if NP2 does not honor the agreement deliberately and fail to fulfill the request of the end user? Obviously, it is a case of an SLA violation, and the proper penalty should be applied to NP2, however the result from the end user's perspective is service failure. In a marketplace that consists of competitive NPs the likelihood of a deliberately SLA violation is high considering that the trust among participants is absent. Therefore, trust among participants is one of main challenges of modern networks. One of the well-known mechanisms to establish trust among two entities is the introduction of a trusted third party [30], but other solutions can also be investigated.

As the 5G network infrastructure grows in an extremely rapid pace, new limitations and challenges arise. Focusing on the management and orchestration domain, the increased network complexity, the introduction of new business-oriented services, the need for performance improvement and the constant research towards the development of future networks beyond 5G, forced us to examine more sophisticated MANO approaches [31]. In the sequel, we examine each of the aforementioned factors as follows:

- Increased network complexity: The heterogenicity and complexity of modern mobile networks increases since massive IoT connectivity is introduced and many emerging services and new 5G/6G technologies are being developed. As a result, the overall complexity of the network orchestration and management increases.
- New business-oriented services: Various services will be developed and rapidly implemented in modern networks, which aim to meet business opportunities. To this end, new management and orchestration frameworks should be designed that will cooperate with other key technologies such as NFV, NS and edge computing infrastructures.
- Performance improvement: NPs should be able to fulfil diverse QoS requirements and at the same time reduce the operational cost and improve network performance. This can be achieved by using efficient solutions responsible for the network operation and service management.
- Development of future networks: Up until now 5G networks are not fully available in every corner of the earth, and therefore many research efforts take place for the development of NGNs. Many new technologies, services, applications, and IoT connections will be available, which will make the future network very complex. As a result, conventional MANO approaches cannot efficiently manage modern network structures and therefore the need for the development of more sophisticated MANO frameworks is present [32].

Considering the above factors, it is clear that new MANO approaches should be designed and implemented, which should present characteristics such as full automation, self-management and self-orchestration. The European Telecommunication Standard (ETSI) moved to this direction by creating the ETSI ZSM Group in December 2017 [33]. The goal of this standardization team is the design and development of the Zero Touch Network and Service Management (ZSM) framework. In the next chapters of this thesis, we will further discuss the structure and operation of the framework.

1.3 Blockchain fundamentals

Another technology that has triggered the interest of academia and industry in the last few years is the blockchain. In parallel with the advances in networking architectures and management solutions, there are advances in distributed trust systems which come under the umbrella of blockchain technologies or distributed ledger technologies (DLT). Blockchain/DLT is one of the most hyped technologies, as they introduce trust in untrusted environments. The way blockchain technology can achieve that is presented next.

Blockchain was firstly presented in public in Satoshi Nakamoto's well known paper in 2008 where blockchain's most popular application was described [34]. In this paper, an electronic transaction system was presented, which uses a brand-new coin called Bitcoin. Due to Bitcoin's popularity the blockchain technology was considered for the general public as a synonym to this cryptocurrency although they are two different things. Blockchain is a technology while Bitcoin is an application of blockchain.

Nakamoto's paper presented not only a new cryptocurrency but also described how blockchain technology could be used to design and implement a complete electronic transaction system that consists of a Peer-to-Peer (P2P) network, a distributed ledger and users who perform transactions without the presence of a trusted third party (i.e., bank). The P2P network connects the participants of the application directly, which automatically reduces the time needed for a transaction to be implemented, while the ledger stores these transactions. The ledger is distributed as it is replicated across all nodes. Essentially, it is a common, continuously updated ledger that provides information to participants regarding the transactions. The distributed nature of the ledger automatically increases the integrity of the data written in it as the content of transactions cannot be altered. The ledger consists of a chain of blocks where the transactions of the users are stored. Users perform transactions through specific interfaces, called "wallets" which store and use their digital security keys. Therefore, it can be stated that blockchain consists of three major elements: the P2P network, the distributed ledger, and the wallets.

Furthermore, the system is characterized by:

- *The use of unique addresses for the implementation of transactions:* Every user of the network acquires a unique private key used for identification and communication purposes. This key corresponds to a unique network address used by the user in order to communicate with other entities in the network and be identified.

- *Permanent storage and immutability of data:* Data included in a valid transaction in the blockchain cannot be deleted or altered afterwards. When a transaction gets validated, it is inserted in a block, and it is stored in the ledger. Then the blocks are linked one another using cryptographic techniques which make impossible the modification or removal of the already inserted data. Cryptography is used not only to link blocks by implementing a cryptographic chain, but also in the process of block creation.
- *Time discrimination of transactions, which are collected and stored in the form of a block linked one another using a cryptographic chain:* Data are inserted in the form of transactions in the block in a serial manner.
- *The use of digital signatures to prove the authenticity of transactions:* Digital signatures ensure that the transaction has been implemented by the actual user, and therefore they contribute to the establishment of trust between users and the system. For the creation of digital signatures, the user's private key that has been used for the creation of the network address, is required.
- *The operation of consensus mechanisms that allow every node of the distributed system to take decisions that comply with the rules of the network:* The consensus mechanisms used in a blockchain network usually are related to the adopted blockchain solution.

In summary, blockchain technology handles data created from transactions among the users of the network. The data are grouped in the form of transactions into blocks which are chained together in a unique manner, using strong cryptography. The term "strong cryptography" denotes the presence of mathematical tools such as hash functions and other techniques which are not something new as they have been developed decades ago. Blockchain uses these mathematical tools to provide immutability of data and increase the security level of the system. For example, if we apply a hash function to a text document, the result will be a specific output with specific length. It is extremely difficult to find the content of the text from the hash value while the hashing process of the text can be done easily. However, if the content of the text changes, the output of the hash function will be completely different and irrelevant to the output of the original message. Therefore, using the hash function we can guarantee that the data used as fuel are not tampered and the integrity of data is ensured. Similar to this example, blockchain links the blocks using mathematical tools to increase the security and immutability of the system. Figure 3. presents how blocks are formed and linked together using the hash function and other cryptographic tools.

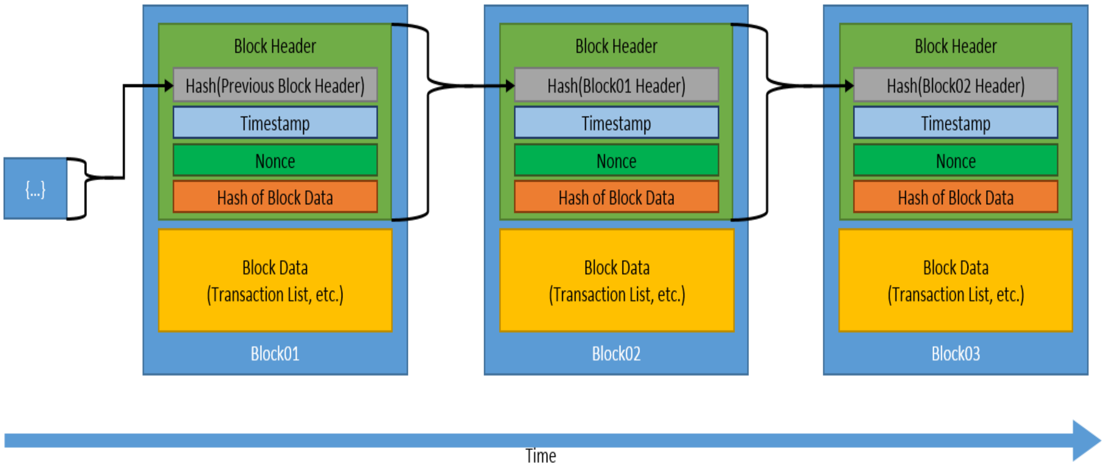

Figure 3. Blockchain overview [35]

Nevertheless, the verification of transactions, the validation of blocks and the growth of the blockchain are based on the consensus mechanism used in the network. Since there is absence of a trusted third party in these distributed environments the role of consensus is extremely important. In blockchain the nodes that form the network are responsible to decide for the validity of a block based on their own judgment. Having in mind the lack of trust among blockchain nodes, there is a need for establishing common rules for the proper operation of the network. Moreover, in order to ensure the normal functionality of the network in many cases the participation of the majority of nodes is required. The set of rules and the way these rules are applied in the network are defined by the consensus mechanism adopted in the blockchain. Some of the most popular consensus mechanisms are: Proof of Work (PoW), Proof of Stake (PoS) and Proof of Authority (PoA).

1.3.1 Types of blockchain

One of the most important decisions during the design of a blockchain solution is the selection of the blockchain type. Blockchains are divided into different types based on who has access to these networks. There are four types of blockchains: public, private, consortium and hybrid. The first and most popular applications of blockchain, Bitcoin and Ethereum, were public networks. Anyone could acquire a blockchain address and become active part of the network. Nevertheless, the exponential growth of blockchain applications called Distributed Apps (DApps) [36], led us to design applications for scenarios that require a small and controlled number of participants such as companies, organizations, and institutions. This resulted to the development of private blockchain networks with different characteristics than the public ones, where the participation is controlled. As new scenarios were examined and new applications were developed, new types of blockchains were also introduced which present some characteristics of both private and public.

Another significant decision regarding the operation of the network is related to the permissions of the user. For example, a user is permitted to actively participate in the consensus process and validate blocks or is allowed only to use the network for performing transactions. Therefore, two additional categories are formed: the permissionless networks and the permissioned. In the permissionless networks anyone can join and participate in the network's procedures while on the permissioned ones only the permitted members can actively participate.

Public blockchain networks

Everyone can participate in public blockchains by connecting to the network. Once the participant is connected, he/she can download a copy of the ledger, perform transactions, and get involved in the process of block validation following the consensus rules. Usually, public blockchains do not require any kind of permission for performing actions in the network, such as write on the ledger, hence they are considered as permissionless. It is worth mentioning however that, in some cases of public blockchains some of the participating nodes may have extended rights regarding the validation process and the storage of the entire ledger.

The nodes of the public blockchains are responsible for the collection of transactions and the validation of blocks in this fully distributed network. The number of connected nodes plays a significant role in the consensus process. The higher the number of participating nodes, the higher the robustness level in cases of a malicious behavior. Nevertheless, the high number of nodes affects the block validation time of the network as it reduces the transaction per second metric (tps). This metric presents how many transactions can be validated in a second. The higher the tps, the faster the network gets.

The advantages of public networks are listed as follows:

- Free access to the blockchain and participation in the consensus and block validation process.
- Increased level of trust as the network is controlled by the users.
- Increased level of trust as the network provides incentives to participants to obey to consensus rules. In many case the network rewards the participating nodes.
- Absence of third party and formation of a completely distributed network.
- Increased security due to the increased number of participants. The consensus process is implemented by a high number of nodes which makes extremely difficult for a malicious entity to attack to the network. Consensus solves the Byzantines Generals Problem, considering that the likelihood for a malicious user to control above the 50% of the network is the minimum [37].
- Increased transparency due to the fact that any participant is able to download the ledger and check the transactions. The transactions are not encrypted but the name of the user is not visible. Instead of the name, transactions use the address of the user which is unique in the network.

The disadvantages of public networks are:

- The small number of valid transactions per second. For example, in Bitcoin a new block is formed every 10 minutes which means that only 7 transactions are validated per second, that is an extremely low number.
- A factor that affects the tps is the consensus used which in the public networks like Bitcoin is the PoW and PoS.
- Network extensibility is an issue since the network is vast, and the ledger has become huge. Considering the Bitcoin blockchain, it is extremely difficult for a new node to join, because the ledger that has to be downloaded is now hundreds of GBs.

Private blockchain networks

The popularity of blockchain was increased over the years, and its application in scenarios where the access to the network is restricted was examined. This resulted to the creation of a new type of blockchain, the private networks where only entities with permission are allowed to access the network and the ledger. These networks are created and managed by one authority, which can control who can participate in the network and access the ledger. Usually, these networks use a mechanism to authenticate users which participate in the private network. It should be noted that since the network is controlled by a single organization/authority, the decentralization characteristic of this blockchain implementation is questioned. Also, the limited number of participants in private networks automatically decreases its size but maintains the basic blockchain characteristics which are: transparency, security of transactions and establishment of trust in the network. Some popular private blockchain networks are the Hyperledger Fabric and Corda [38, 39].

The main advantages of private blockchain networks are as follows:

- High number of valid transactions per second which leads to faster block creation and validation. The limited number of participating nodes allows the consensus process to be completed faster and this decreases the block validation time.
- Increased network extensibility: Since the number of nodes are smaller than those in public networks, the addition of a new node is not a difficult task. The newly added node can quickly be synchronized with the other nodes by downloading the ledger which is significantly smaller than the ledger of public networks.

Disadvantages of private implementations of blockchain:

- The decentralization level of the network is questioned since a single entity controls and manages the network.
- Decreased level of trust since the network is controlled by one authority.
- Security issues which are introduced due to the limited number of participants. If a malicious user manages to become a participant, it is easier to affect the consensus process and the overall network operation.

Beyond the two main categories that was described above, there are two more blockchain implementations which present characteristics of both public and private blockchains. These are the consortium and hybrid networks. In consortium, the blockchain is governed by a group of organizations and the participation in the network is controlled (permissioned blockchain). The consortium blockchain is not an easy task as it requires cooperation between several entities which presents logistical challenges. Additionally, some implementations of this type of blockchain present nodes with different roles. Some of them actively participate in the consensus process by validating transactions and blocks, while others only initiate transactions in the network.

In hybrid networks, there is a combination of the most powerful characteristics of both private and public implementations. In this type of blockchain we do not observe distinguished roles of the nodes like those in consortium blockchains. In addition, in hybrid implementations the access to the network is controlled and only the permitted entities can use the network and gain full access to the data and the functionalities. Figure 4. Types of blockchain networks illustrates the four main types of blockchain that have been discussed in this section.

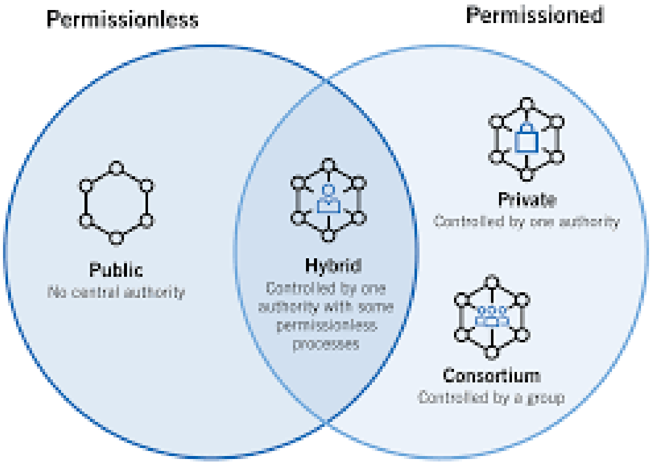

Figure 4. Types of blockchain networks [40]

1.3.2 Advantages and disadvantages of blockchain technology

Having discussed the main characteristics and features of the blockchain technology in the previous subsections, we proceed to the identification of the advantages and disadvantages of this technology. The advantages of blockchain are:

- ✓ *Absence of a trusted third party:* The participants of the network have the full control of its operation and growth, as they participate in the consensus process to create and validate new blocks.
- ✓ *No Single Point of Failure (SPoF):* The distributed nature of blockchain automatically eliminates the SPoF problem. Even if a node goes down for

several reasons (maintenance, attack etc.) the network continues to operate normally.

- ✓ *Data integrity and immutability:* The use of cryptography when the transactions are stored and validated in a block, and the strong cryptographic links used for the chaining of blocks, strengthens the security of the information. Once the information is stored in the ledger it cannot be erased or altered.
- ✓ *Transparency:* The information is stored in the ledger of the blockchain and is available to the participants. It should be noted that transactions contain the information and the network addresses of the parties that perform this transaction. There is no information on the network regarding the correspondence of a physical identity to a network address.
- ✓ *Credibility:* The rules of a blockchain network cannot be altered by one entity. Changes to the rules of the network are proposed by the participants. If the majority of participants vote in favor of a change, then the change is applied in the network. As the number of participants grows so does the credibility of the system.
- ✓ *Traceability:* The increased transparency of data, the integrity and immutability of the information automatically increases the traceability of a transaction written in the ledger.
- ✓ *Trust:* The characteristics of data integrity and immutability, data transparency and traceability, as well as the existence of the consensus mechanism, increase the user's feeling of trust towards the blockchain network.

The main disadvantages of blockchain technology are as follows:

- *Scalability:* Information stored in the ledger cannot be deleted, which means that the chain is growing as more and more transactions are validated. Also, the blockchain network grows as new nodes are entering, which should acquire a copy of the ledger to be synced with other participants. Therefore, the scalability of the systems is a major issue for this technology.
- *Use of private keys:* The private keys of the user are used for the creation of a blockchain address. This means that if the keys of the user are lost or stolen the content of his/her account is lost.

- *Speed of transactions:* To maintain blockchain technology an attractive solution for several use cases, the speed of transaction validation must be high. Already, many DLT systems offer high speed of transactions to satisfy demanding scenarios.
- *Increased cost:* When a company or organization actively participates in a blockchain network, creates one or many blockchain nodes. The implementation and maintenance of nodes increases the CAPEX and OPEX.

Having the pros and cons of blockchain in mind, it is obvious that this technology is not panacea. There are use case scenarios where blockchain could be proved a powerful weapon to overcome and even eliminated security issues. However, the consequences of using this technology should be considered when the designing of a solution takes place.

1.4 Problem statement, identification of our research area and structure of thesis

Networks beyond 5G should support demanding services worldwide which require the cooperation of NPs and the implementation of resource management processes in multi-administrative domains. Since NPs form a competitive marketplace characterized by lack of trust, the collaboration of NPs is a challenging task considering that new NPs are entering the market as it was stated in previous sections. To establish trust in this trustless environment two options are available: the introduction of a trusted third party which is a centralized approach and the examination of a decentralized approach by using distributed ledger technologies such as the blockchain. The latter will be thoroughly examined in the current thesis, aiming to take advantage of the benefits of this hyped technology and at the same time avoid the drawbacks of centralized approaches. The design of a solution based on blockchain technology attracts the interest of NPs as it:

- ✓ Guarantees transaction security. A resource management process is considered as a transaction.
- ✓ Embraces the benefits of blockchain by introducing characteristics such as data immutability and increased traceability. Therefore, the non-repudiation problem is solved as the participants are able to search the transactions written in the ledger if necessary.

- ✓ Performs resource management tasks in the form of Smart Contracts (SCs) which are executed automatically when needed. This results to increase of the security of management tasks and improves the dynamicity of resource management processes.

Moreover, the resource management processes between NPs should be implemented in an automated and secure manner in NGNs, since the network requirements of modern applications and the network conditions change dynamically. As a result, NGNs should be characterized by self-manageability and self-orchestration in order to rapidly adapt to changes and applications' demands. Researchers currently investigate the implementation of the ZSM framework in networks beyond 5G although they highlight various security issues that we will discuss in the following sections. These issues can be addressed by using blockchain technology as it will be presented in the next chapters of the thesis. Concluding, the current thesis highlights a new research area by focusing on the combination of blockchain with ZSM framework to achieve secure and automated resource management in NGNs, as it is illustrated in Figure 5. Identification of research area.

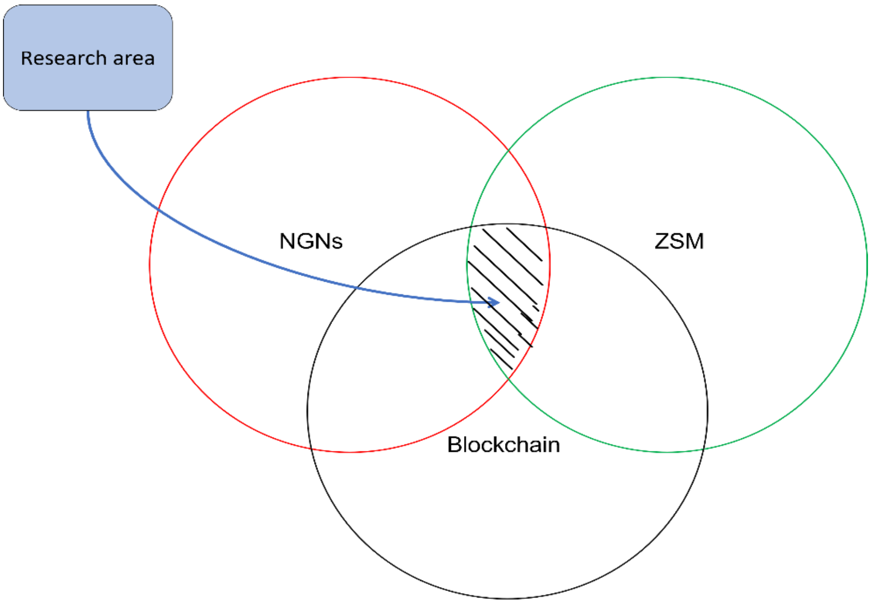

Figure 5. Identification of research area

Concluding, the presented thesis is structured as follows:

- *Chapter 1 – Introduction:* This chapter presents the hardware and software advancements that led to the development of 5G ecosystem and presents the limitations and challenges of current network structures. In addition, the need for more sophisticated network management and orchestration mechanisms that will contribute to the network's evolution is highlighted. ETSI confirms this need as it has kicked off a new standardization attempt for the development of Zero Touch and Service Management (ZSM) framework for NGNs. The trust and security issues of resource management processes are discussed, and possible approaches are presented. Furthermore, an analysis of blockchain fundamentals takes place. This technology can be used to tackle the security and trust issues of resource management processes and enhance the ZSM security. This results to the identification of a new research topic which is thoroughly examined in this thesis and can be described as, research for secure resource management in NGNs.

- *Chapter 2 – State of the art and related work:* An analysis of the ZSM framework is presented in this chapter, focusing on the main principles, the requirements, and the architecture of the framework. Also, the key components of ZSM architecture are discussed in detail. The main security issues and challenges of the framework are presented in order to highlight the weak points of the standard. At the same time, two well-known blockchain networks are presented, the Bitcoin and Ethereum. Moreover, the definition and analysis of the Smart Contract (SC) and Distributed App (DApp) takes place. It should be mentioned that blockchain is a technology that is not limited only to the cryptocurrency sector and can be applied in many use cases. To justify this statement, this chapter presents use case scenarios where blockchain can be combined with other technologies, such as crowdsourcing, while the application of blockchain in the human resources sector is presented in the BLER use case. Moreover, to present large scale blockchain applications, we focus on the supply chain sector by presenting many applications of this technology in detail. Having discussed the ability of blockchain to interoperate with other technologies, we focus on research papers where blockchain is used in the networking sector to increase the level of trust and enhance the overall security.
- *Chapter 3 – Design, implementation and evaluation of a blockchain-based application in dynamic resource management of NGNs:* In this chapter, we take advantage of the benefits that blockchain inherently provides and we

design and evaluate an application of blockchain in resource management scenarios of NGNs. A detailed analysis of this blockchain application takes place, that is implemented and tested in real testbeds. The blockchain network used in these experiments is Ethereum-based and it is called Quorum. The performance of the blockchain network is measured using the Hyperledger Caliper tool where we focus on metrics such as throughput (transactions per second), latency and success rate of the transaction. The application is evaluated using two different types of consensus mechanism, the Raft and the IBFT. Raft belongs to the crash fault tolerant consensus family while IBFT belong to byzantine fault tolerant. The functionality of the consensus mechanisms used is presented in detail as well as the operation of the application. Finally, the strong and weak points of this solution are presented. This blockchain application focuses on static resource management processes and does not examine scenarios where the network is self-managed and self-orchestrated.

- *Chapter 4 – A blockchain-based ZSM approach:* The combination of blockchain technology and ZSM framework takes place in this chapter. Having discussed the structure and operation of ZSM and having examined how blockchain can be applied in resource management scenarios, we present a detailed architecture of a blockchain-based ZSM framework. The functionality of our approach is analyzed and the way it addresses the ZSM's security issues is highlighted. Also, the challenges and limitations of this approach are discussed as well as possible improvements. Nevertheless, in order to increase the security of ZSM and at the same time maintain the performance level high, we proceed to the definition of requirements that the ideal blockchain should fulfill. These requirements are extremely valuable as they help us draw the profile of the ideal blockchain solution for the implementation of the blockchain-enabled ZSM scenario. Our research findings urged us to continue our study in order to find the most suitable DLTs for our scenario. Therefore, beyond the study of blockchain solutions, we proceed to the investigation of another promising DLT category, the Directed Acyclic Graphs (DAGs).

- *Chapter 5- Conclusions and future work:* This final chapter of the thesis presents the main findings of our research. Moreover, the main challenges and limitations of using blockchain technology in NGNs are discussed. Finally, this chapter concludes the current thesis by presenting future research paths that lead to the development of networks beyond 5G.

2. State of the art and related work

The high network performance metrics offered by 5G and the development of demanding applications made the design and implementation of sophisticated MANO frameworks mandatory. The ETSI standardization body has proceeded to the creation of the Zero touch network and Service Management (ZSM) framework to enable self-orchestration and self-management in MANO systems. The goal is to eliminate human intervention in the management and orchestration process of NGNs in order to achieve full automation, decrease the time needed for management actions and avoid errors caused by humans. In the following sections of this chapter, we present studies where blockchain technology and AI/ML are used in resource management and ZSM networks in order to show the growing interest in this field of study. Moreover, an analysis of the ZSM takes place where we discuss the main principles, the requirements, the architecture, and the security challenges of this novel framework.

At the same time the adoption of blockchain technology in computer networks gains more ground. Beyond the famous cryptocurrency applications, as well as other domains like human resources and recruitment solutions [41], Supply Chain Management (SCM) sector [42] and the crowdsourcing systems [43], the use of blockchain in NGNs can prove beneficial. In this chapter, we explore several blockchain applications in the aforementioned sectors, with emphasis in networks that adopt the Zero Touch Service Management framework.

2.1 The rise of Blockchain technology

The inherent characteristics of blockchain led to the rapid adoption of this technology initially in the finance sector as it was mentioned in the introduction of this thesis. Bitcoin and Ethereum are the two most popular applications of blockchain and are presented in this section. Furthermore, the development of a new feature of this technology, the Smart Contract, gave an extra boost to the creation of many different blockchain applications called DApps. In this chapter of the thesis a description of the Smart Contract takes place in order to realize the significance of this feature and how it can be used for the creation of DApps. In addition, the ability to develop customized DApps using SCs allowed us to use blockchain technology in various sectors and use cases. Therefore, in this chapter the use of blockchain in many scenarios (recruitment, SCM, crowdsourcing) is examined in order to highlight the wide adoption of this hyped technology. Finally, interesting research works are presented focusing on the use of blockchain in NGNs to enhance the security and trust among participants. The growing interest for using blockchain in NGNs urged us to investigate this research area and proceed to the study and design of secure resource management mechanisms in networks beyond 5G.

2.1.1 The Bitcoin, the Ethereum and the role of Smart Contract (SC)

In contrast to traditional banking systems, Bitcoin does not require the use of a central trusted authority to establish trust among participants. Bitcoin is based on decentralized trust guaranteed by the blockchain network which is characterized as public and permissionless. Anyone can access the Bitcoin and join the network by hosting a Bitcoin node. The nodes that form the P2P network are equal, but they may have different roles. According to Dr. Antonopoulos in his book [44], a Bitcoin node is a collection of functions: routing, the blockchain database, mining, and wallet services. A full node includes all four functionalities while all nodes support the routing function to participate in the network. A full node has the full copy of the ledger and verifies any transaction autonomously without any external reference while the lightweight nodes maintain a subset of the blockchain that allows them to verify transactions using the simplified payment verification method. The creation of new blocks is a responsibility of the mining nodes which compete for the insertion of a new block by solving complex mathematical problems as they run the PoW consensus algorithm. A mining node can be either a full or a lightweight node. Additionally, a full node and a lightweight node can support user wallets to perform transactions on the network.

Every node of the Bitcoin network should be able to:

- Access and download the common ledger that contains the chain of blocks. These blocks include validated and verified transactions of the users of the blockchain. As new transactions form new blocks the state of the ledger continuously changes.
- Obey the consensus rules which are used for the validation of transactions that will be executed and included to the next block. These rules describe the

procedures that need to be implemented for the validation of the proposed block.

- Confirm the mining of new cryptocurrency in the network.
- Follow the PoW consensus algorithm for the selection of the next block that will be added to the chain.

A block consists of a set of transactions, that require a significant amount of computational power to prove, but only a small amount of computation to be proven. The mining process validates the transactions according to consensus rules described by Bitcoin. Therefore, invalid or malformed transactions are rejected which automatically increases the security of Bitcoin's transactions. Also, the mining results to the generation of new Bitcoins created when a new block is formed. This Bitcoin is the reward of the miner that has successfully inserted the block to the blockchain network. As a result, the reward incentivizes the mining nodes to compete for the creation of new blocks. It is worth mentioning that the reward is given to the miner only if the miner has validated the transaction according to the rules defined by the consensus mechanism. [44]

Another popular blockchain application is the Ethereum platform which is a public blockchain network. In contrast to Bitcoin, which is used only for cryptocurrency transactions, Ethereum is used also for several other use cases such as the execution of computational programs written in the form of code called Smart Contracts and the support of Distributed Applications (DApps). Therefore, it is considered to be a general-purpose blockchain network. Ethereum acts as one powerful computer implemented by a global distributed computing system that consists of nodes which run the Ethereum Virtual Machine (EVM). The nodes of Ethereum are synchronized and maintain the same state regarding the ledger of the blockchain and the operation of EVM. Also, the nodes can initiate new transactions and observe the status of the submitted ones. The EVMs that form the Ethereum network update their information continuously in order to follow the changes in the network. [45]

In order to achieve consensus among EVMs and maintain high levels of synchronization, the Ethereum uses the blockchain technology for the implementation of the network and a cryptocurrency used as a fuel to initiate transactions, called Ether. The changes of the EVM state are stored in the blocks of the blockchain while the Ether is used for the execution of SCs and for performing transactions among participants. The EVM environment allows the execution of SCs as it translates the code written by the developer into machine language called bytecode. Then the bytecode can be executed in the EVM environment of any node in the Ethereum network. Focusing on the term SC, it can be described as a set of promises written in code that include protocols which force the involved parties to fulfill these promises. A SC is an immutable computer program that runs deterministically inside a EVM as part of the Ethereum protocol. In contrast to traditional software, the content of the contract cannot be altered when it is deployed. The only way to modify the functionality of the contract is to deploy a new one. The deterministic nature of SC denotes that the outcome of SC's execution is the same regardless of who runs it, given the context of the transaction that initiated its execution and the state of the Ethereum blockchain at the moment of execution. Moreover, SC once it is deployed it is available in every EVM of the Ethereum network as it is depicted in Figure 6. Deployment of SC in the Ethereum network. Since every EVM instance has the same initial state and produces the same final state, the execution of the SC can be implemented in every Ethereum node. [45]

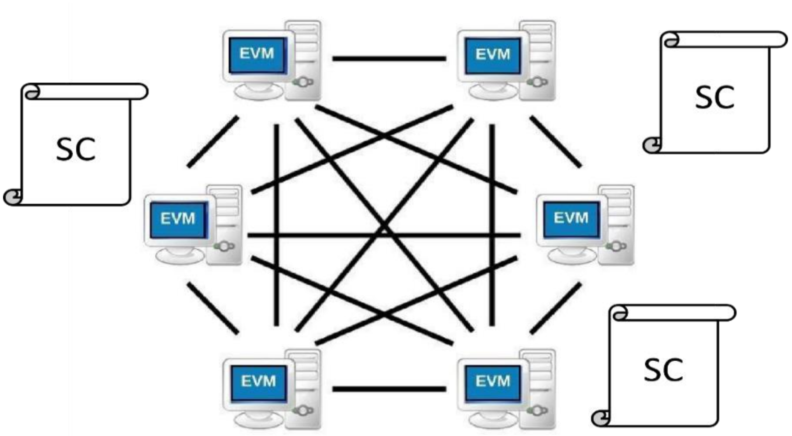

Figure 6. Deployment of SC in the Ethereum network

The ability to develop computer programs in blockchain networks in the form of SC led the way for the design and creation of more sophisticated DApps. The development of SC can be done using Touring complete languages such as Solidity and by using programming tools such as the Remix and Truffle. When the SC that implements the logic of the application is combined with a web user interface then the result is the creation of a DApp. The use of SC for the creation of custom DApps broadens the application area of blockchain technology.

However, DApps can access and use data that are already in the blockchain and cannot by themselves interact with entities outside of the network (i.e., web services) in a secure manner. Therefore, a new blockchain component is introduced, the oracle. Similar to the ancient Greek world, where oracles were communicating directly to the gods to provide valid information, in blockchain oracles are used to receive and transmit valid data to entities outside the network. To create a secure communication channel, they use cryptography to protect the information retrieved from a valid source (signed data), while some oracles use a consensus mechanism implemented outside the blockchain to evaluate the validity of information.

2.1.2 Consensus mechanisms

One of the key elements of blockchain technology is the consensus mechanism, that essentially defines a set of rules based on which the network operates. In systems where there is absence of a trusted authority, the consensus algorithm establishes a layer of trust among participants. Participants decide for the future of a block (accept or reject) via the consensus process and for the selection of the next block to be validated. Moreover, the consensus process is based on the fact that the content of the blocks is immutable and final, which means that the content of the block has not been changed in the past and will not change in the future. It is safe to say that the consensus is the keystone of the blockchain network, and the selection of the proper mechanism is a very important task for the operation of the blockchain. As it aforementioned in the introduction section various consensus algorithms have been developed where the most popular are the PoW and the PoS.

The Proof of Work (PoW) mechanism

The PoW uses the mining process which is applied to a block as soon as all transactions have been verified. Every blockchain node that has a copy of the ledger can perform mining. During the mining, the participating nodes receive a difficult computational problem that must be solved. It is a speed race among miners and the one that finishes first generates the new block and receives the reward. Although, the initial though is that mining is used for the creation of new BTC in the case of Bitcoin, the whole process practically strengthens the security of the blockchain. The reward is used to incentivize miners to join the process and increase the decentralized security of the network. In PoW consensus there is also a corresponding “punishment”, which is the cost of energy required to participate in mining. If participants do not follow the rules and earn the reward, they risk the funds they have already spent on electricity to mine.

Thus, PoW consensus is a careful balance of risk and reward that drives participants to behave honestly out of self-interest. [45]

The mining process can be divided in three steps as Figure 7. Finding the PoW solution shows:

1. Continuous fragmentations of the block's header to find the PoW solution.
2. Repeat the first step while the hash changes at least by 1 bit. The hash changes based on the Nonce variable of the block that defines the difficulty level of the PoW.
3. In every repetition, the output value that has been found is compared to the difficulty level of the network to check its validity. If the value has reached the difficulty level, it is considered as a successful mining.

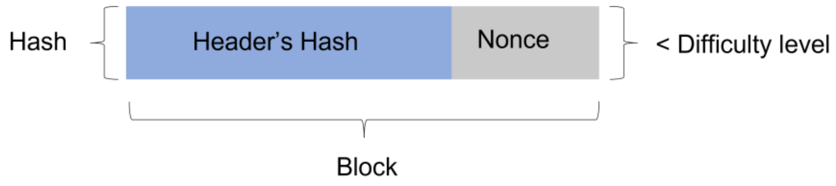

Figure 7. Finding the PoW solution

Additionally, this consensus algorithm solves the double spending problem discussed in [44] while there is justice among participants as the result is based on cryptography. However, PoW presents significant disadvantages as the effort of nodes to solve the hard computational problem has an impact on the energy. Also, the increased difficulty of the PoW process results to a low number of verified transactions per second, which means that new blocks are generated in an extremely low pace.

The Proof of Stake (PoS) mechanism

In order to reduce the energy footprint caused by the PoW a new consensus mechanism is used, called the Proof of Stake. PoS does not perform mining, instead every node that is willing to participate in the process of finding the new block transact with a SC. The nodes deposit to this contract address, the amount of cryptocurrency they are willing to offer in order to receive the right to generate the new block. As a result, the amount of cryptocurrency that the node deposits is used as a stake in this algorithm. When the SC has received the stakes, it randomly selects one node among the candidates to propose the new block. Then the node generates the new block and verifies the transactions in it. Once the verification of transactions has been completed, the node presents the new block to the other peers (nodes) of the network which proceed to its verification. If the verification process finishes successfully, the node that generated the block receives the rewards which is the fees of the transactions included in the block. However, if the new block contains invalid transactions, then it is cancelled by the other nodes of the network and the node that has proposed the block loses its stake. The stake is at the same time a guarantee of the node that is responsible for the proper generation of the new block. The popularity of PoS continuously increases as many blockchains are using it, such as Cardano, and it will be used in Ethereum 2.0.

The Byzantines Generals Problem and the Byzantine Fault Tolerant consensus family

The consensus process in a blockchain should not only tackle trust issues but should also guarantee the successful operation of the network in cases of nodes' failure or malicious activity. According to Lamport et.al in [46], a computing system should be able to properly operate even if parts of its system present failures, which may be caused by technical failures or malicious actions. As a result, there is a need for achieving consensus in a distributed system and at the same time guarantee the proper functionality of the system when problems that affect its performance occur. This problem is described in [46] as the Byzantines Generals Problem, where generals of Byzantium want to conquer a hostile city and must reach to a consensus regarding the plan of invasion. Generals should communicate with messages to reach to an agreement, however there is a possibility that messages may get lost or tampered by the enemies or by traitor generals who want to sabotage the attack.

In order to address the Byzantines Generals Problem, algorithms that are Byzantine Fault Tolerant (BFT) have been developed and used in consensus mechanisms. According to [46], it has been proven that in the case of oral messages, to defend against m malicious generals (nodes), there must be at least $3 * m + 1$ generals in the network. Furthermore, there are many BFT consensus mechanisms available which present variations one another and can be used in different use cases. Some of the most common are the practical Byzantine Fault Tolerant (pBFT) [47] and the Istanbul Byzantine Fault Tolerant (IBFT) [48].

Beyond the BFT algorithms, there are other consensus mechanisms which are more tolerant to failures. This family of consensus is called Crash Fault Tolerant (CFT) and it is more resilient to failures than the BFT. However, in this case CFT algorithms are not tolerant to malicious nodes which means that they are susceptible to Byzantines Generals Problem. CFT mechanism usually present increased consensus speed comparing to BFT ones and can operate in hazardous situations by maintaining the performance of the blockchain network in high levels. A popular CFT mechanism is the Raft [49] used in Ethereum based blockchain networks. In the following chapters of this thesis, experiments using both consensus families are conducted in order to realize how the consensus process affects the overall performance of the blockchain. In the following subsections of this chapter many applications of blockchain in different areas are presented and discussed in order to highlight the increasing interest in the adoption of this hyped technology.

2.1.3 Indicative blockchain application sectors

2.1.3.1 Blockchain in human resources – The BLER platform

Recruitment is one of the most critical parts of human resources (HR) management; it is where trust must be established before candidates proceed to an interview and/or employment. A unified platform, where trusted evidence of earned certificates and academic degrees is safely kept, would be enormously useful and would increase HR's efficiency. Such a solution is described in [50]. The scope of BLER is twofold: first, to ensure the integrity of information and, more specifically, to use blockchain to verify the validity of qualifications/ certifications, thus outperforming current (professional) social networks (e.g., LinkedIn) while obviating the need for validated copies; and, second, to give users control over the visibility of their personal information (i.e., qualifications, certificates, degrees, other) for each job application, adding an extra level of confidentiality.

Figure 8 depicts the operation of the BLER system. When any applicant/student successfully finishes a training course or acquires an academic degree, the academic institution or training organization inserts information regarding the qualification or degree into a common distributed ledger. When applying for a job, the applicant grants access to his or her profile details in the ledger to specific recruiters to ensure them that the information is genuine. Recruiters can then filter the received applications based on job requirements. Applicants have full control over the visibility of their profiles, which are not publicly available. BLER provides access to an applicant's complete academic history and eliminates the possibility of fraud through false certificates and degrees.

BLER currently supports three user types: applicant, recruiter, and academic institution. The role of each type of actor is explained below.

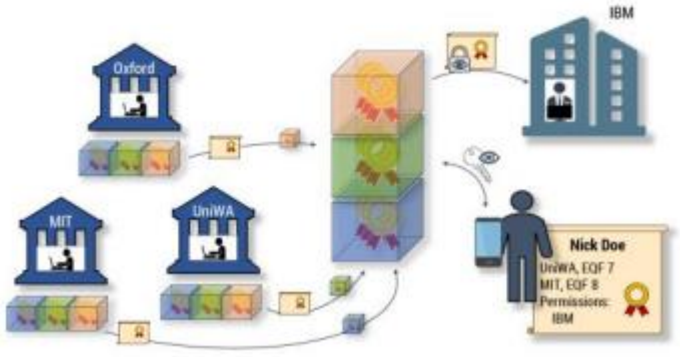

Figure 8. The BLER platform architecture [50]

Applicants (e.g., Nick Doe in Figure 8) are all individuals who are (potentially) seeking a job and have educational qualifications. This includes students at any educational organization, employees that undertake training, individuals with degrees, and so on. Once registered in the platform, they have full visibility into and control over their profile. Profiles contain any qualifications/credits, degrees, or other training certificates that the corresponding authorities have entered, along with demographic data (also entered by these authorities). Applicants cannot alter their data; they can only alter the permissions they grant to specific accounts. Thus, applicants can either allow open access to their data, exactly as they would do in a professional social network or provide access only to specific organizations or agencies to which they have applied for a job or turned to for help in finding a job. This action is almost equivalent to applying for a job and sending a CV. However, with BLER, the recruiter has access to this verified information for only a specific time span determined by the applicant. In addition, the use of blockchain technology — which mandates the maintenance of the information in multiple nodes (see Figure 8), with no node able to alter already inserted information — guarantees the information's integrity.

Recruiters may be a company (e.g., IBM in Figure 8) or a public administration or HR agency. Recruiters can view an applicant's profile, provided the applicant has granted the recruiter access; recruiters can then easily trace and confirm the applicant's specific skills and education. BLER offers an applicant profile that contains all academic titles, certifications, qualifications, or other data inserted by organizations that support the

BLER solution. While in the current implementation we focus on education-related information, the information included in an applicant's profile can be expanded by other organizations that support BLER. Such organizations may, for example, insert work experience certificates. This capability would allow the creation of a record of professional experience.

Academic institutions (e.g., the Massachusetts Institute of Technology [MIT], Oxford University in Figure 8) are the BLER actors representing universities or training organizations, or any organization that can certify skills, knowledge, and competencies. All academic institutions are able to register applicants and add academic qualifications to a profile, including information such as degree title, grade, European Qualifications Framework (EQF) level,3 and graduation year. Most important, the academic institution is the only BLER actor that can insert information into the blockchain.

These three BLER roles — applicants, recruiters, and academic institutions — exist to add educational qualifications, give or revoke visibility permissions, and query for educational qualifications. In blockchain, this kind of logic is implemented via a smart contract mechanism, which is computer code programmed to be triggered by certain events and to digitally facilitate the negotiation or contractual terms directly between users when certain conditions are met. Smart contracts allow the performance of credible transactions without the need for or presence of third parties, making them a perfect fit for a system, such as blockchain, that enforces trust between the participants.

2.1.3.2 Blockchain in Supply Chain Management (SCM)

Another sector that could benefit the most by adopting the blockchain technology is the Supply Chain Management (SCM) [42]. SCM, particularly, is one of the areas whose performance will be substantially affected by applying blockchain technology [51-53]. At the same time, blockchain is not a panacea nor should it be applied to all domains just because it is at a hype. The authors in [54] propose a flow chart to help people/organizations decide whether a blockchain-enabled solution should be considered for implementation, anticipating a substantial boost in their use case and, also, guide them to define the kind of solution that could be applied (e.g., private vs. public blockchain). This chart, in line with others [55], [56], suggests that SCM is such a case where blockchain technology can offer a significant boost. For SCM, we consider multiple organizations maintaining and processing information currently in isolated data silos which are difficult to interconnect due to multiple reasons from lack of trust among the involved parties, implementation of heterogeneous proprietary solutions to vulnerability to attacks (e.g., attack the database of a product provider).

Blockchain is expected to offer a unified framework, to be used by all the many participants in different stages of the SCM [57-59], with many possibilities and numerous benefits including, but not limited to:

- ✓ The creation of an immutable system where information is stored and protected by cryptography, consensus, and timestamps. As a result, this immutable nature of the SCM ledger enhances the willingness of the suppliers to participate in the process and add their data to the blockchain.
- ✓ The introduced transparency across all the stages of the SCM system increases the level of trust in its performance which leads to increased trust both between the partners and between partners and end users/consumers.
- ✓ The use of the ledger for detection and tracking of any token/asset (i.e., packet or animal in the blockchain) along with detection of any anomalies or gaps in the management process throughout its life-circle. Origin verification can, also, be applied in the chain since the trace can be followed back to its roots from any user in the blockchain. An example where such an approach is needed is for tracing the farm where specific animals have been affected by a virus that has, also, harmed humans.
- ✓ The use of IoT devices that connect and fuel the blockchain directly with data, without any human intervention in the process increasing the integrity of the process.
- ✓ Increased system security since any device can enter the system, encouraged to follow the rules, because there will be no gain going against them, while strengthening the overall system defense. The latter is enforced by the number of participating nodes due to its decentralized nature and the need to control over 50% of the nodes in case of an attack [60], for it to be able to succeed a breach in the system.
- ✓ Smart contracts implementation to trigger actions based on the data that are stored in the blockchain. Usually, those contracts apply to initiate instant payments or/and alerts, supporting the automation in the system based on secured and processed data in it.

- ✓ New digital experience and services for the products and SCM with the end user’s role and control over the process being enhanced significantly.

At the same time, in order to use blockchain to address SCM’s (or, otherwise, asset chain’s) operations, a process that represents any asset to the digital world, as a token that can be stored, processed and transferred on a blockchain is needed. This process, also known as tokenization [61], is of fundamental importance for all the solutions that deal with SCM. Keeping in mind that supply chain includes various smaller stages ranging from raw materials, suppliers, manufacturers, distributors, retailers to end-user/ customers, a blockchain platform might be designed to cover the whole process, at least for a specific use case scenario. Otherwise, co-operation between different blockchain implementations is needed to provide for a complete solution for SCM. Figure 9 illustrates the stages that are included in SCM, along with possibilities/benefits that are born from the use of blockchain.

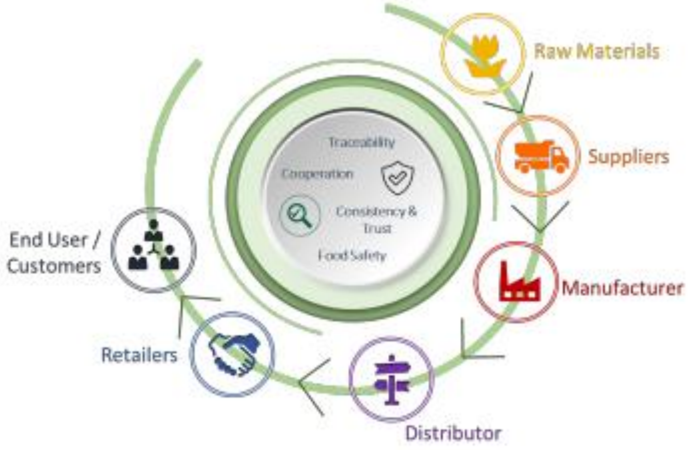

Figure 9. SCM's circle with stages and benefits/possibilities from using blockchain [42]

With the number of available solutions for SCM rising quickly, several ones are attempting to cover a specific use case all over the supply chain while others to propose a common framework and bridge the gap to create a common understanding, blockchain-based “language”. While a blockchain platform consists of all the software and hardware required to deploy the distributed ledger, a blockchain protocol is any tool enriching the functionality of the blockchain platform (e.g., Ambrosus protocol).

Concluding the presentation of blockchain application in SCM, it is safe to claim that blockchain can be used in SCM to satisfy diverse purposes. Each purpose imposes different challenges and requirements which can be met by blockchain technology.

Three main characteristics of the most popular blockchain platforms and protocols for SCM need to be carefully addressed based on the intended use:

- the right to access the Blockchain,
- the support for an IoT ecosystem, and
- the support for a unifying Blockchain.

Design choice 1: Private vs. Public and its business - economy relevant challenge. To decide between a private or public blockchain platform, the solution designer has to answer the question:" Is there a single organization that is responsible for the operation of the blockchain and for authenticating the nodes? Who owns and operates the nodes?".

When a private solution is implemented every member of the Blockchain is authorized to join the network and is permitted to read the ledger, transact and participate to the consensus procedure. The capabilities of a member are restricted by the rules that have been set by the system. On the other hand, in public blockchains, everyone can join and participate in the network without needing for a permission. Public blockchains have also the advantage of being popular solutions due to the presence of Bitcoin and Ethereum. However, the interest regarding the design and implementation of private blockchain platforms, such as HLF, presents a significant growth.

While a public blockchain should cover all the stages of the SCM with basic details, private blockchains seem appropriate to be used in a single-stage of the SCM, focusing on more performance-specific data of the involved actors of that stage that could be used for improving their performance and, therefore, the system's. Both blockchains need to work supplementary with primary focus given in the public implementation to be able to satisfy the needs of all the actors in the SCM.

Design choice 2: IoT-generated information in the supply chain relevant to elaborate tracking and food security. To decide whether an IoT-capable blockchain must be implemented, the prospective designer must answer the question:" are IoT-generated information automatically stored in the ledger?". A positive answer is more likely in the food and pharmaceutical supply chains and mandate that large amounts of "transactions" must be supported, which challenges the scalability and energy efficiency aspects of the blockchain technology. This has fueled efforts for developing blockchains (e.g., Ambrosus) that store the data captured by sensing/actuating devices combining IoT systems with blockchain. However, the designer must be careful as many of them support proprietary IoT devices which currently impede their wide deployment. The challenge thus moves in the interconnection of IoT-fueled ecosystems with blockchains to serve SCM requirements and operations.

Design Choice 3: Need for communication between different blockchains, supporting various use cases of SCM or not. Most of the solutions found during our research can support a reported use case throughout the life cycle of the asset on demand. Even though different kinds of data are appended at each stage on the blockchain, interoperation actions are offered from many solutions. Furthermore, the need to interconnect two (or more) discrete blockchains, covering different use cases of SCM, is starting to arise and blockchains that play the role of a middleware are being studied and developed. While one could think of the challenge to be already solved with the Inter Ledger Protocol (ILP), this is not the case because current implementations of ILP focus on transactions which are much simpler to handle (i.e., the transaction type remains the same and thus it is a transformation of currency) while in SCM more complex information management is needed. On the contrary, Waltonchain offers a parent blockchain (public) and child chains that interoperate through the parent one. It is easy to understand that a one-solution-to-fit-all approach cannot be expected, but there are options that provide increased possibilities and benefits, depending on the case.

2.1.3.3 Blockchain in crowdsourcing

Crowdsourcing systems can also benefit the most by adopting blockchain technology [43]. Crowdsourcing systems have been victim to a number of cyber-attacks, most of which aimed to compromise and steal data or render systems unavailable. For example, in March 2014 the freelancer platform Elance experienced a Distributed Denial-of-Service (DDoS) attack [62] which kept the systems unavailable for more than a day. More precisely, attackers employed the use of a Network Time Protocol (NTP) reflection attack. Another large-scale cyber-attack against a well-known crowdsourcing system happened in October 2016 affecting UBER. According to Bloomberg [63], hackers were able to steal personal data by gaining access to Uber’s private Github account which contained developers’ credentials for their Amazon Web Services (AWS) platform. This ultimately provides access to Uber’s AWS databases containing driver’s personal data.

Furthermore, free-riding (e.g., benefiting from crowd-sourced task output without having contributed to its production) and false-reporting (e.g., to avoid the payment the employer lies regarding the task's status) are common attacks on crowdsourcing platforms, therefore, the need to propose and apply countermeasures is of great importance for maintaining the data integrity and utility of crowdsource platforms. For example, the use of Eliminating Free-riding and False-Reporting with arbitration (EFF) and Discouraging Free-riding and False-Reporting with arbitration (DFF) auction-based mechanisms and the development of reputation protocols in those untrusted environments are solutions able to prevent these problems [64, 65]. The EFF and the DFF are based on any existing truthful double auction scheme for winner selection and pricing. The auction winner is required to deposit a warranty and then submit a report regarding the status of the corresponding task. The payment is determined by the platform and is based on these reports. While these mechanisms are useful tools, they don't use any kind of encryption to guarantee the integrity of the process.

Finally, the crowdsourcing platforms should perform regularly security assessments regarding their status. These assessments should be carried on by experienced security officers who will, at the end of the process, provide a report that highlights the vulnerable points of the system. To this end, the platform should apply the best practices regarding the storage of sensitive information (i.e., encryption) and should be compliant with the General Data Protection Regulation (GDPR).

Blockchain technology addresses efficiently the weaknesses of crowdsourcing systems, this way boosting their attractiveness to solve several problems and widening their application potential. A blockchain database retains the complete, indelible, and immutable history of all transactions, assets, and instructions executed since the very first one. With this, blockchain allows participating parties—and only those parties—to share accessible, transparent, and trusted information. The main characteristics to remember are: a) decentralized and distributed ledger storage and integrity, b) the ledger is irreversible and immutable, c) its operation is near real time (i.e. transactions verified and settled in minutes vs. days) and in any case satisfies the speed requirements of crowdsourcing which are significantly looser than those of the financial sector initially targeted by blockchain and d) it respects privacy (no personal data need to be registered). Users are identified by digital identities (exactly as credit cards) and only when physical world personal data are linked to those digital identities, is the linkage in place.

Adopting blockchain technology, the ledger of all transactions can be kept in a set of nodes (belonging either to workers or to requesters) obviating the need for a central authority/entity. The node resources are thus contributed by the peers that benefit from the platform and a small reward is granted to them. Such a system is proposed in [66], where a distributed system (entitled CrowdBC) is organized into three layers: the application layer, the blockchain layer and the storage layer. The blockchain layer is where the attributes of a transaction are kept (i.e., the ledger) while the storage layer includes the details and the content of the work produced by the workers. The application layer implements the business logic which, in the considered use case, is the user manager, the task manager and the program compiler. An important element of the CrowdBC is the use of smart contracts which follow the concept of smart contracts defined in Ethereum. The smart contract [67] is a self-executing digital contract in a secure environment with no intervention, which is verified through network peers. In crowdsourcing systems, a smart contract can be used by the system to describe the request-worker relationship (where the task ID, the task owner, the relevant deposit and task status are kept).

With respect to crowdsourcing, targeting information collection for different purposes ranging from facts (as e.g., Waze Carpool), opinions on events, products, and solutions to collection of pieces of evidence and verdicts, blockchain makes possible the involvement of a larger number of people, which increases the quality of the data and thus of the offered service. Blockchain has been proposed for judgement produce to increase the quality of justice in [68]. Blockchain technology is also leveraged to improve other crowd-sourcing cases, like crowdfunding. Equity crowdfunding is considered a new channel of raising money for start-ups encouraging innovation and the adoption of blockchain based solutions has important advantages as reported in [69].

The crowd sourcing Blockchain-enabled systems, until now, have been comprised of approaches that include the creation of a platform for advertising crowd-working tasks that initiate partnerships between possible “workers” and employers.

It is worth stressing that one of the drawbacks attributed to blockchain technology is the energy consumption increase which is caused primarily by the mining and consensus process. While before implementing such a system the energy consumption should be considered as well, we anticipate that the volume of “transactions” in a crowd-sourcing blockchain solution is by far less than in case of a blockchain solution is used for money transactions. Additionally, a consensus algorithm different than Proof of Work (PoW) can be used (e.g., Proof of Stake, PoS) that leads to significantly lower energy consumption. For example, employing a private blockchain solution, the energy consumed is significantly lower due to lower intensity of processing thanks to the lightweight consensus mechanisms (see Hyperledger Fabric solution). On top of it, Hyperledger allows each node to hold more than one ledger allowing the creation of different channels to host each ledger. Therefore, each studied SN could use each own ledger to decrease the growth rate of each applied solution.

2.1.4 Blockchain in NGNs

The continuous increase of blockchain’s popularity and its adoption in various sectors of our life, urged the research community to examine possible applications of blockchain in modern networks. The idea is to take advantage of blockchain’s inherent characteristics and solve major security issues identified in NGNs. At the same time, the goal is to minimize the impact of blockchain’s drawbacks in order to fully embrace the goods of this technology. To this end, numerous research works have been conducted and published in the community. In this section of the thesis, we survey the most important ones focusing on the resource management and orchestration processes of NGN.

Both for inter- and intra-administrative domain resource negotiation and allocation, two main options exist [70]: centralized and decentralized. Typical centralized approaches have been studied and used in many technologies while decentralized solutions are becoming extremely popular in the technological arena. Centralized approaches [71-73] have the advantage that they can achieve high performance due to the availability of information regarding the status of the whole network/domain. However, this comes with certain drawbacks: The centralized nature of the brokering mechanism automatically labels it as a SPoF (Single Point of Failure). If this centralized entity is out of service, then the operation of the whole system is disrupted. Furthermore, the communication between the entities participating in the resource brokering should be secured so that the data cannot be altered (which would cause service unavailability).

On the other hand, by adopting the blockchain concept, the different resource providers would be members of the DLT network hosting one (or more) nodes which obviates the SPoF attack. Each of these nodes keeps a copy of the ledger and is participating in the consensus procedure for validating the information registered in the form of transactions. The consensus mechanism used in blockchain discourages any node from performing malicious actions and validating false transactions.

Herbaut et al. in [74], present a model for collaborative blockchain-based video delivery. This work studies how the combination of a smart contract stored in a blockchain, and network service chaining can be used for supporting collaboration schemes. A keystone of this study is the introduction of a decentralized brokering mechanism for the creation of content sessions through the collaboration of CP (Content Provider) and a TE (Technical Enabler). Then, an attempt for using dynamic service chains takes place in order to benefit from link diversity of different TEs. The decentralized brokering mechanism is established among a CP and a TE which compete and collaborate for the instantiation of the best content delivery session. This decentralized mechanism is based on the blockchain technology and the various stages of this model are described by the use of Smart Contracts (SCs). One of the most critical aspects of the proposed solution is the time needed to converge toward the optimal Content Delivery Contract (CDC), involving the end user, the CP, and the TE. Therefore, authors chose the Hyperledger Fabric blockchain solution, that uses the practical Byzantine Fault Tolerance (pBFT) consensus algorithm, due to its high performance in terms of throughput and latency [75]. Additionally, Hyperledger Fabric (HLF) is a permissioned platform (i.e., every node is known to the other), which is something useful for this particular use case. Although authors present encouraging results in terms of convergence, the scalability of their proposed model based on HLF is questioned, as the experiment nodes were located in the same availability zone of the cloud infrastructure. However, the proposed solution does not specify where these blockchain nodes are hosted nor how they use their wallets for performing transactions in the blockchain network. Furthermore, after ending up with the optimal CDC it is not clear if the SCs in the blockchain are responsible for placement of the required network service function chain for supporting the content delivery.

Rebello et al. in [76], propose a blockchain based solution for secure orchestration operations in virtualized networks, ensuring auditability, non-repudiation and integrity. BSec-NFV Orchestrator (BSec-NFV) aims to protect the creation, management and termination of virtual machines, virtual network functions, and service chains. The contribution of this study lies in the introduction of blockchain and transaction models that provide traceability in a multi-tenant and multi-domain NFV environment. Their use case scenario is based on four key assumptions: (i) limited number of identified providers, as each provider takes part in service level agreements with tenants and other providers; (ii) low number of crash failures, due to the high availability of big data centers; (iii) high throughput and low latency in end-to-end communication, as VNFs are implemented in the network core; and (iv) tolerance to malicious behavior between competing providers and tenants. The authors develop their solution using the HLF that utilizes the pBFT consensus. Their evaluation shows that the overhead added by blockchain is not significant (causes an additional 3% delay with a confidence interval of 95%) while the throughput is considered by the authors to remain in acceptable levels. However, the evaluation is conducted in a data center environment where the various blockchain modules are placed in nearby virtual machines. In NGNs the number of providers may increase and therefore the tolerance to malicious participants should be higher. Additionally, this work does not focus on the resource negotiation among providers and how this can be achieved using blockchain.

Nour et al. in [77] propose the use of DLT in Network Slicing by presenting a blockchain-enabled Network Slice Broker (NSB). The purpose of the NSB is to guarantee the construction of secure end-to-end network slices in order to support applications of 5G vertical industries, using resources from different stakeholders of the 5G network. When a slice provider receives a request to build an end-to-end slice, it publishes in the blockchain a request for resources regarding each sub-slice composing the end-to-end slice. After receiving the different offers for each sub-slice, the slice provider selects the best offer in terms of cost and the capabilities to meet the requested performance. The proposed solutions introduce the use of two blockchains, one permission-less and one permissioned. The negotiation regarding the resources takes place on the permission-less blockchain, where the prices and capabilities of all offers are visible to everyone. Once the selection of the provider has been made, the permissioned blockchain is used for the creation of the end-to-end service chain. This work examines the use of Hashcash blockchain which utilizes a Proof of Work (PoW) consensus, and the results present its poor performance in terms of time needed to instantiate a slice. Additionally, authors do not mention which platform they recommend for the public blockchain and which for the private. Moreover, the use of wallets is not examined, and the experiments take place in machines located in the same area which automatically excludes network related parameters in performance evaluation.

Rebello et al. in [78] propose a blockchain solution for network slicing, where they introduce the use of different blockchains for different slice requirements. So, in this work, the network slices are categorized based on their requirements and, the blockchain data structure, the consensus, and the communication protocol are tailored to each specific network slice functionality. The goal of this work is to present a blockchain architecture for the creation of secure network slices for each end-to-end use case in 5G. The implementation of this solution is based also in the HLF software. Similar to previews studies, here the authors propose their solution in data center environments where there is no restriction regarding the resources. To ensure justice in consensus, each data center of the NPs may host at most one blockchain node per blockchain (it is reminded that each slice type is associated with a different blockchain). Blockchain nodes in a slice type are invisible to anyone outside the slice. This study proposes the use of a management blockchain where all VNF orchestration operations are logged in order to provide auditability and management regarding the slice creation. The management of various VNFs is accomplished by using SCs to introduce transparency and automation in this decentralized system. The architecture of this system is composed by four components: a user interface, the NFV MANO module, a blockchain creation server module, and a management blockchain server module. However, the evaluation of the prototype is conducted in one physical machine where the HLF nodes are running inside a container. As a result, we cannot be sure how this solution would operate if the nodes were in different networks and locations. Additionally, this work assumes that the blockchain runs in a data center environment. Furthermore, the scalability of the presented solution is not well defined, although in contrast to previews works, here a detailed analysis of blockchain’s operation is illustrated. Finally, authors are not focusing on the resource negotiation procedure that takes place among providers in this multi-tenant and multi-domain NFV environment.

The interest in using blockchain for resource management in modern networks is increasing and resulting in many interesting works as it is presented in [79]. Togou et al. in [80], present a distributed blockchain-based broker (DBB) for the dynamic leasing of resources among different network operators to support end-to-end services in a multi-administrative network. DBB includes a biding mechanism used for the management of incoming requests and the construction of Memorandums of Understanding (MoUs) among operators. This solution guarantees that SLAs among operators are fulfilled. The biding process requires the proposition of bids by the operators, which are inserted into the blockchain as transactions. Then the operator who requests resources selects the cheapest one. However, the insertion of all proposed bids and not only the winning ones in the blockchain arise scalability issues. Also, the introduction of auctioning can lead to time variations, while the requests of resources must be served as soon as possible. The monitoring of the leased resources is based on a QoS matrix that is not included in the blockchain, which means that it is not fully protected from a malicious operator who may try to cheat. Moreover, the experimental part of this approach consists of a simulation that does not take into account the impact of blockchain on the performance of the system.

Maksymyuk et al. in [81] discuss the potential benefits and challenges of the integration of blockchain in the mobile network infrastructure in terms of spectrum and infrastructure sharing. Authors propose a blockchain-based framework for decentralized 6G mobile networks to ensure cooperative network management by multiple Mobile Network Operators (MNOs). Due to the potentially huge number of transactions per second produced by mobile networks, the performance of the framework is very sensitive to the “speed” of the blockchain network. The speed of the network is influenced by the underlying consensus algorithms. Therefore, this work presents the use of a new consensus, the Proof of Formulation (PoF) used in the FLETA blockchain which according to authors can reach more than 10,000 transactions per second. They propose a combination of permissionless (public) and permission (consortium) blockchains. However, the authors have not clearly indicated which platform they have used to achieve these results and do not give the evaluation details.

In [82], Xu et al. present use cases of blockchain in next generation networks in a very abstract manner. Focusing on network slicing and resource management, authors propose the use of blockchain and SCs to introduce transparency and fairness to the system. The trading of a network slice is based on blockchain, where the SC orders the slice orchestration based on the agreed SLA described in the 5G network slice broker. The blockchain is integrated to store the usage of each leased resource and check the performance of a service provider against the SLA. According to authors, the key benefit that is introduced through the blockchain is the establishment of a trust layer, which lowers the collaboration/cooperation barrier and enables an effective and efficient ecosystem. Also, blockchain prevents the SPoF problem and thus improves systems’ security. Moreover, one of the elements that play a significant role according to authors to the performance of this solution is the consensus mechanism. However, they stay in a theoretical level.

Papadakis et al. propose in [83] a blockchain-based Network Service Marketplace (NSM) and a resource orchestration mechanism that enables the Cross Service Communication (CSC) in edge cloud (EC) for the creation of NSM. The authors present a complete solution for a multi-tenant edge cloud ecosystem described by an architectural diagram. The main functionalities of the NSM are the registration, the advertisement, the discovery, the lease, the usage, and the billing. In the registration phase, the tenant of an EC enters the solution and offers its services which are advertised in the network. In the discovery phase, the users can browse and select the desired services and proceed to the lease. The usage of the services is monitored to perform the billing at the end of the lease. The blockchain layer handles through SCs all information required regarding users, services, etc. Authors select the Hyperledger Fabric platform for the implementation of their solution and conduct experiments to test the performance regarding the transaction per second (TPS) and latency of the transactions implemented in the blockchain network. However, the tested network is deployed on a single VM which means that the impact on the node's communication through the internet (e.g. introduction of latency) is not taken into consideration.

Hewa et al. in [84] present the role of blockchain in 6G networks and the benefits introduced by this technology such as privacy, integrity, and accountability. The authors focus on the application areas of this technology in 6G systems, for example, in industrial applications beyond industry 4.0, smart healthcare, decentralized and seamless environmental monitoring and protection. Also, this paper discusses the use of blockchain for achieving decentralized network management to achieve better resource management, enhance SLA management and spectrum sharing. In [85] Praveen et al., describe the idea of a blockchain-enabled slice broker and how the use of SCs can leverage the negotiation process among NPs in terms of automation and security. Also, this work examines the use of blockchain technology in spectrum allocation, sharing and management by the implementation of Dynamic Spectrum Sharing. Furthermore, the interest of academia and industry for the combination of blockchain and NGNs can be proven by the participation of companies such as Intracom and Atos, in research projects like 5G Zorro [86]. The main concept of this project is to use blockchain SCs for network and security management.

In their survey paper, Liyanage et.al [31] present the progress of the ZSM standardization and highlight the main goals and challenges. While ZSM's goal is to provide high quality E2E services to the end user by automating the functional1ities of the core network, there are several security threats to be addressed especially in cross domain scenarios. Since ZSM relies on AI and ML to achieve full automation by implementing closed loop procedures and operate core management services, components that implement, these two technologies need to be secured. The security threats highlighted in this work by the authors are: ML/AI-based attacks, open API security threats, intent-based security threats, automated Closed-Loop network based security threats, and threats due to programmable network technologies. Moreover, the multidomain and heterogeneous nature of modern networks labels trust among different entities as a major issue. According to authors these open issues have not been sufficiently explored, although there are some published ideas where the use of blockchain is discussed as a solution.

In [87], authors discuss the considerations regarding trust in modern multi-stakeholder networks and propose the use of blockchain technology to deal with trust issues. Smart Contracts (SCs) deployed in blockchain networks are ideal to create Service Level Agreements (SLAs) among stakeholders and control SLA violations in a transparent and secure manner. Based on the table presented by the authors, blockchain can be combined with many other technologies to solve trust and security issues in modern networks. Some of these technologies are: VNFs, AI and ML. Moreover, sensitive data in modern networks can be protected using the blockchain technology in order to guarantee their integrity and provenance. Authors discuss a use case where data are used as fuel for AI and ML focusing on the importance of data security and highlight that data security is extremely important in AI/ML based solutions. Data must be untampered and protected in order to avoid the dataset poisoning which may lead to wrong decisions taken by the AI and ML mechanisms. In this use case, the data can be relevant to the service deployment parameters and the measured quality and blockchain technology could solve the security and trust issues, as stated in [87].

Benzaid et.al [88], describe the concept of Zero Touch Networks (ZTNs) and how AI can be used to automate the service management of modern networks. The presented research highlights the benefits derived by using AI technology to form ZTNs where the main characteristics are: self-management, self-healing and minimum human intervention. However, beyond the advantages of AI-driven ZTNs, there are certain limitations highlighted by the authors. Security and trust are considered open issues by the authors when AI is used. According to authors, it has been proven that ML techniques are vulnerable to several attacks targeting both the training phase and the test phase. Since data are used by the AI mechanism, their integrity and provenance are important for the proper operation of the mechanism. Authors claim that blockchain technology can be the antidote to these security limitations, due to its immutability and distributed nature.

Authors in [89], present a combination of AI technology and DLTs in order to increase the security and trust in multi-operator mobile/cellular networks. Authors highlight the ability of AI to offer characteristics such as self-adaptation and self-reaction to next generation networks which are susceptible to changes regarding the network conditions. This research is part of the 5GZORRO project, and its goal is to present a conceptual architecture of a solution that uses AI and DLTs. The advantages of this solution are highlighted while, authors present several use cases where the use of these technologies could offer significant advantages. Another work of the same project [90] proposes the use of Smart Contracts (SCs) coupled with Cloud-Native operational Data Lakes to provide a zero-touch solution for the automated service assurance of multi-domain network slices. The SLAs which define the proper performance of the services are applied in the form of SCs deployed in a blockchain network to increase the transparency of the process and to facilitate the integrity of the agreement. Additionally, the AI technology is used to predict SLA violations which may lead to service degradation. Concluding, this research presents an architecture for a Smart Contract-based service assurance mechanism for network slices in a multi-domain environment which is SLA-driven. Also, this work aims to present a definition of AI-driven SLA breach detection and mitigation mechanisms implemented as modular Cloud-native services. The validation of this solution takes place through the deployment of a CDN scenario on a large-scale 5G testbed. However, this work does not elaborate on the definition of the resources that should be allocated to each service to prevent the SLA breach detected by the AI-mechanism.

Concluding, the interest in the adoption of blockchain technology in various sectors beyond cryptocurrency is growing. Considering the advancements in the networking sector and the need for automated and secure resource management in NGNs, we examine the use of blockchain technology in modern networks. The already published works show us that there is room for further research and investigation towards this approach. Moreover, to the best of our knowledge none of the existing works clearly propose an architecture to answer how cross domain Network Service Management could be implemented in a secure manner using both blockchain technology and ZSM framework. In the next chapters of our thesis, the development and evaluation of a blockchain-based solution for resource management in modern networks is proposed. Then, based on the ZSM framework, we proceed to the design of a blockchain enabled ZSM architecture in order to address the main security issues defined by the standardization team. Our ultimate goal is to examine how the blockchain technology can be used in an efficient manner in order to guarantee the secure resource management in NGNs which may lead the way to the development and management of networks beyond 5G.

2.2 The Zero Touch Network and Service Management (ZSM) framework for NGNs

In this section a presentation of research works focusing on the use of AI/ML in modern networks takes place, followed by a detailed presentation of the ZSM framework based on the ZSM's reference architecture [91]. The main principles and requirements of ZSM are discussed and the architecture of the framework is illustrated. Furthermore, an analysis of the architecture's main components is presented followed by a description of the ZSM's main security challenges as they are stated by the ETSI team in [92].

2.2.1 The role of AI/ML in the ZSM concept implementation

Artificial intelligence and Machine Learning techniques have been pursued to support the profiling of a service, the forecasting of the quality a service will experience for a given deployment scenario, and the placement of an NFV among others.

Uzunidis et.al [93], focus on the resource management process in NGNs and the proper network service profiling and placement in order to offer high QoE to the end user. In this work authors present a framework to address the problem of service profiling and to predict the system's "critical points", focusing on complex services running over containers. In order to ensure the proper QoE by avoiding SLA violations and at the same time increase the efficiency of utilized resources (achieve minimum or even zero underutilization) authors use AI and ML technology to predict the critical points which indicate a change regarding the NSs characteristics (i.e. profile, placement).To evaluate their framework, they conduct experiments in a Hadoop environment in order to perform service profiling and performance predictions by monitoring an extensive number of critical system metrics (e.g., CPU usage, memory usage, service throughput etc.) from three layers, namely the physical, virtual and service layers. The results of this work show that AI/ML technology can increase the efficient resource utilization in NGNs without affecting the end user's QoE.

In [94], authors focus on the problem of choosing the proper amount of resources to support applications based on VNFs, especially in Mobile Edge Computing (MEC) environments where the computational resources are limited. They argue that AI technology can solve this problem in 5G networks, and they use it to develop a predictive autoscaling mechanism in NFV MANO that could beforehand automatically adapt the resources to the workload used by the application without any human intervention. The autoscaling feature of modern virtualized networks is the key to leverage the resource efficiency of the system. Having in mind that communication networks are administered by different entities/organisations, the multidomain scenario is also discussed in this research. As a result, Service Level Agreements (SLA) among network stakeholders are formed where the desired QoS must be guaranteed at the agreed price. Therefore, authors in this paper leverage on Federated Learning (FL) techniques to design deep learning models for predictive Virtual MEC Application Functions (VMAF) autoscaling in a multi-domain setting that can better react to the changing service requirements, optimize the network resource usage, and also comply with data protection policies. This research concludes with a comparison of the predictive autoscaling approach to a reactive approach, based on results gathered by experiments in which centralized and federated learning techniques have been applied in a Kubernetes testbed.

Dalgkitsis et.al [95], examine the use of Reinforcement Learning (RL) and more specifically, they leverage a Deep Deterministic Policy Gradient (DDPG) RL algorithm to solve the NFV placement problem in a scenario that consists of a Data Center (DC) and multiple Mobile Edge Computing (MECs) infrastructures. The goal is to minimize latency for ultra-Reliable Low Latency Communications (uRLLC). This research follows the definition of ETSI Experiential Network Intelligence (ENI) and Zero-touch Service Management (ZSM) standards. Authors use the Deep RL (DRL) to automate the live migration of VNFs and add the self-adaptation characteristic to the system which is beneficial for both vendors and end-users. The experimental results of this work are encouraging and therefore we may claim that RL can be a feasible solution for the resource management problem in modern networks regarding the accurate prediction of resources needed to support a service, as well as the proper management of resources (i.e. VNF placement) to avoid SLA violations.

Authors in [96] present a framework for Zero Touch Networks that uses the AI technology and microservices to perform self-orchestration of end-to-end network services. The presented research is part of the European H2020 program called CHARITY. The goal is to increase the QoE by respecting Key Performance Indicators (KPIs), which are based on NGNs characteristics such as, high availability and ultra-low latency. The outcome of this research is an Artificial Intelligence based Resource aware Orchestration (AIRO) framework in Cloud Native Environment that has been tested through simulation. The network services are implemented in the form of microservices using containers managed by a Kubernetes implementation. Finally, the results of the performance evaluation of the framework proposed by authors measured through simulation are very close to the results observed in a real testbed, which lead us to the conclusion that it is safe to use the simulator to easily examine the performance of AIRO. However, authors do not address the security issues when AI technology is used.

2.2.2 Basic principles of ZSM architecture

Zero touch Service Management (ZSM) is designed to enable zero-touch automated network service and management in a multivendor environment. The reference architecture of this framework was based on a number of principles in order to achieve high levels of full automation and service [91]. These principles are as follows:

1. *Modularity:* The monolithic architecture is avoided by using self-contained, loosely coupled services with specific roles which interact via well-defined interfaces. This results to the easy addition of new services, the update or removal of existing ones without performing major changes on the system.

Also, the modularity property reduces the troubleshooting time in case of a malfunction which increases the availability of the framework.

2. *Extensibility:* This property is related to the previous one and highlights the ability of the system to easily accept new services, service functionalities and endpoints without any backward-compatibility problems which require modifications to already existing service designs, implementations, and interactions.
3. *Scalability:* A modern MANO framework should be able to adapt rapidly to changes in order to satisfy the increasing or decreasing demands of managed entities (i.e., Network Providers, end-users). This means that deployments should be able to scale both in terms of resources (to satisfy the network requirements) and in terms of geographical distribution (offer services globally).
4. *Model-driven:* A model-driven architecture uses information models for the service management. Information models capture the definition of managed entities in terms of attributes and supported operations. The goal of models is to facilitate portability, reusability and to enable vendor-neutral management of resources and services.
5. *Closed-loop management automation:* This property is based on a feedback-driven process implemented in the framework, aiming at increasing the performance of the network by leveraging on the efficient resource utilization and on characteristics such as self-optimization and automated service assurance and fulfilment.
6. *Support for stateless management functions:* Management functions which separate processing from data storage are included.
7. *Resilience:* The management services, which are the heart of ZSM, are designed to overcome any functionality issues when degradation of infrastructure or of other critical services occurs. When the degradation has been resolved, the management services return to their initial and normal state. To reach the desired level of resilience, management services provide and maintain configurable stages of their offered functionalities.
8. *Separation of concerns in management:* The ZSM framework uses two management concerns, the management domain, and the end-to-end service management across multiple domains. On the one hand, a management

domain consists of its resources and the services they support. On the other hand, end-to-end cross-domain service management manages and orchestrates end-to-end services implemented across multiple domains. The separation of these two management concerns reduces the complexity of the system and boosts the independent evolution of management domain and of end-to-end management.

9. *Service composability:* The services offered by management domains are called management services and can be combined to create new management services.
10. *Intent-based interfaces:* The scope of these interfaces is to provide a high level of abstraction to the user in order to conceal the complexity, technology- and vendor-specific details.
11. *Functional abstraction:* This principle denotes the ability of the framework to provide a simple description of the behavior of the system's entities. More specific, the details of those entities are encapsulated into a single one.
12. *Simplicity:* The complexity level of the architecture is the minimum without making any discount to the fulfillment of functional and non-functional requirements.
13. *Designed for automation:* The automation of network and services management and the integration of technology advancements are supported by framework's components and functionalities.

2.2.3 Functional and Non-Functional requirements of ZSM architecture

Beyond the main principles used as a guide for the development of ZSM architecture, certain requirements were also defined by the ZSM team in [91]. The architecture requirements are divided in two main categories: the functional and the non-functional. Table 1 presents the functional requirements while Table 2 illustrates the non-functional ones.

Table 1. Functional requirements of ZSM architecture

Functional Requirements			
General functional requirements	**Functional requirements for data collection**	**Functional requirements for cross-domain data services**	**Functional requirements for cross-domain service integration and access**
Manage resources and services exposed from management domain and across multiple management domains.	Collecting up-to-date data which can be telemetry data (monitoring infrastructure resources), logs, data for ML.	Support cross-domain data services.	Registration of the provided management services.
Cross domain management of end-to-end services.	Storing of collected data.	Separation of data storage and data processing.	Support discovery of the provided management services.
Support adaptive closed-loop management.	Common access to collected data across management domains.	Sharing of data within ZSM framework architecture.	Information about the means to access a discovered service.
Support bounding the automated decision-making mechanisms by rules and policies set by the operator.	Enforcement of data governance for shared data.	Enable of data recovery in an automated manner.	Support synchronous and asynchronous communication between service producers and service consumers.
Hide the management complexity of domain and services.	Aggregation of the collected data cross-domain, and pre-processing of the data.	Management of consistency of redundant-stored data in an automated manner.	Support indirect invocation of the management services.

All domains should be able to implement an end-to-end Service	Support of different degrees of cadence, velocity and volume of data collection.	Enable data service failover in an automated manner.	Allow the direct invocation of discovered management services by the service consumer.
Management services shall support automation of operational lifecycle management functions as applicable to the resources and services.	Management of collected data distribution and maintenance of distributed data consistency.	Support automated overload handling of data services.	-
Definition of standard interfaces within the management domains to achieve fully automated management.	Provide data to data consumer according to the data consumer's requirements.	Support capabilities that allow logically centralized storage and processing of data, as well as the automatic provisioning of these capabilities.	-
Support access control to services exposed by the management domains.	Ability to attach metadata to collected data.	Support for automated policy-based data processing.	-
Support open interfaces.	-	Support processing of several data services with different data types in an automated manner.	-
Management of end-to-end services that cross boundaries between different domains.	-	-	-

Table 2. Non-functional requirements of ZSM architecture

Non-functional requirements		
General non-functional requirements	**Non-functional requirements for cross-domain data services**	**Non-functional requirements for cross-domain service integration**
Achieve a specified level of availability of the ZSM.	Handling of different data services QoS (throughput, delay) requirements.	Support integration of new and legacy management functions.
Management actions are complied with regulatory requirements.	Interoperability of data services across different management domains.	Integration of management services into the ZSM framework should not require changes to the management functions.
Energy efficiency.	Interoperability of data services provided by the ZSM framework with data services outside of the ZSM framework.	Support on-demand addition or removal of management services.
Vendor, operator and service provider agnostic.	Data processing within pre-defined processing time.	Support coexistence of different management service versions at the same time.
-	Execute management task within pre-defined processing time.	-
-	High data availability.	-

2.2.4 Security requirements of ZSM architecture

Significant role in the design of ZSM architecture play the security requirements. ZSM team based on the CIA model (Confidentiality, Integrity, and Availability) defined certain requirements in order to facilitate the proper operation of ZSM's core functions and to protect the data used in the framework. These requirements are listed as follows:

1. The ZSM architecture shall provide security of data at rest, in transit and in use, infrastructure resources, managed services and management functions.

The general term “data” is used for the management data and the data related to management functions (i.e., logs).

2. Support confidentiality of management data at rest, in transit and in use.
3. Guarantee integrity of data at rest, in transit and in use.
4. Guarantee integrity of managed services and management functions.
5. Offer high availability of data, infrastructure resources, managed services and management functions, in so far as security measures to handle availability threats are concerned.
6. Enable privacy of personal data using mechanisms such as privacy-by-design and privacy-by-default.
7. The building blocks of the ZSM architecture shall include the necessary safeguards and features to ensure security of operation as well as data protection appropriate to mitigate the risks.
8. Allow authorization of service access by authenticated service consumers.
9. Apply security policies in an automated manner, according to the compliance status of management services regarding to the security requirements.
10. Provide capabilities for automated incident detection, identification, prevention and mitigation.
11. The ZSM architecture shall support capabilities to audit/supervise AI/ML decisions against security and privacy criteria to prevent the proliferation of vulnerabilities and attacks.

2.2.5 The ZSM reference architecture

The architecture of the ZSM framework is defined by a set of architectural building blocks which collaborate in order to build and support complex management services and functions. The management of ZSM and its data services are developed in a distributed manner and organized into management domains. The integration of management and data services is a responsibility of the integration fabric that is used to enable management service consumption, communication, and integration with third party systems. The data sharing among different management domains is performed by the cross-domain data service. The key components and services defined in the ZSM architecture allow the delivery of end-to-end zero touch management of network services and infrastructure.

The main blocks in ZSM architecture are illustrated in Figure 10. and are the following: the management services, the management functions, the management domains (MD), the end-to-end (E2E) service management domain, the cross-domain integration fabric and the data services. Management services are the core component as they can be offered and consumed by other services and ZSM participants to support network services and applications. The management services consumption and/or offering is done using management functions as it is presented in Figure 10. . A management function can either be a "management service producer", a management "service consumer", or both at the same time. Moreover, management domains are used to define different areas of responsibility that belong to a different ZSM participant. Each management domain can use its own management services or services offered for consumption by other management domains using the ZSM framework. The E2E service management domain depicted at the upper part of Figure 10. , is a special management domain that provides end-to-end management of customer-facing services, composed from the customer-facing or resource-facing services provided by one or more management domains. The “cross-domain integration fabric” located at the center of Figure 10. is responsible for the interoperation and communication between the management functions within or across different domains. The registration, discovery and invocation of management services and the communication between management functions are implemented by the integration fabric. Finally, data services enable consistent means of shared management data access and persistence by authorized consumers across management services within or across management domains. [91]

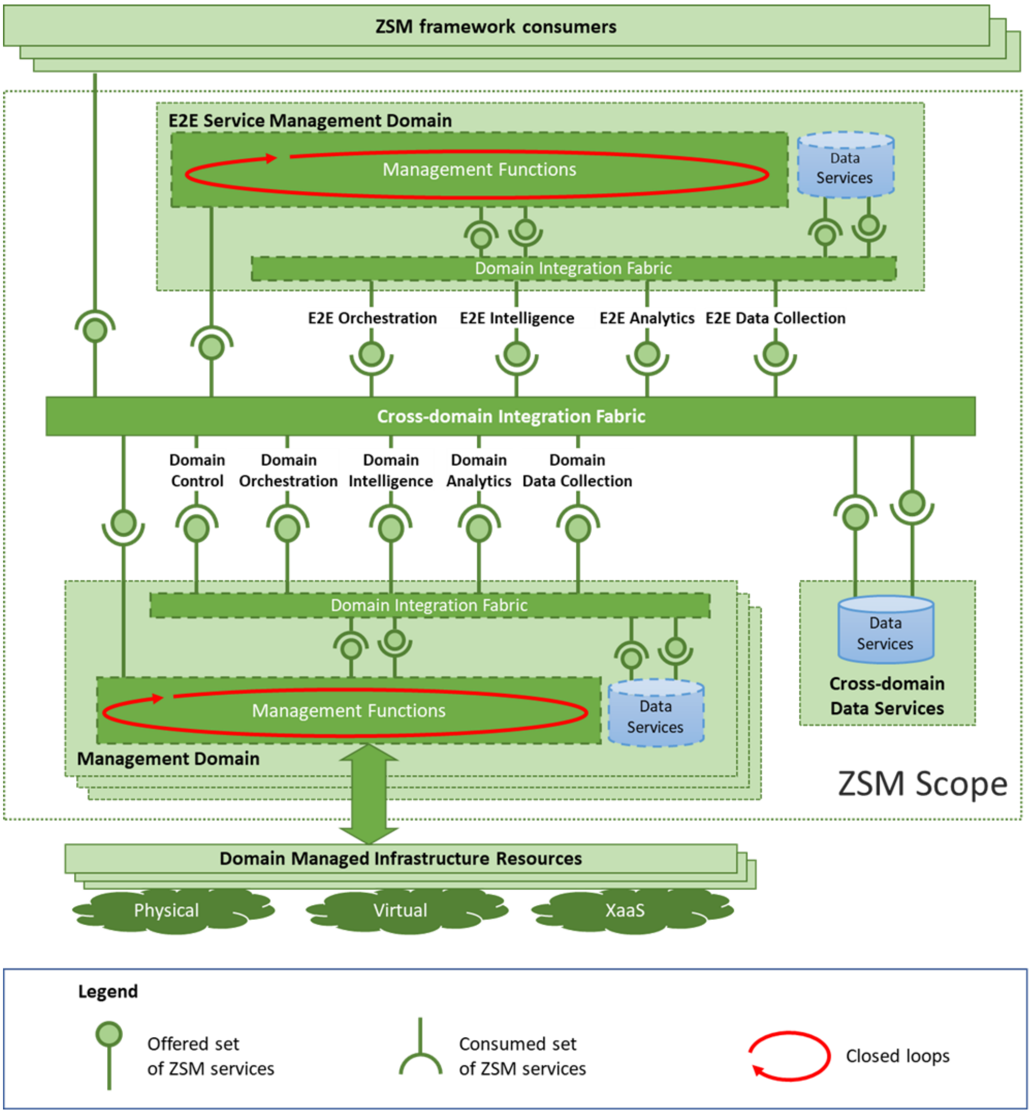

Figure 10. ZSM reference architecture [91]

Considering the role of the aforementioned blocks, an example of ZSM operation is examined: supposing there is a multi-domain network where NP1 is responsible to support a demanding network service with characteristics that require a specific set of resources to provide the necessary QoS level. In the presented scenario NP1's resources are unavailable in the area where the service must be deployed. According to standard's functionality, NP1 exploits the ZSM elements, such as management functions and cross-domain integration fabric, to find and consume a management service offered by NP2, which implements the necessary actions to cover the needs that NP1 has defined.

Focusing on the ZSM architecture figure, a crucial role for its functionality plays the closed loops. In Figure 10. we observe the red arrow that indicates the presence of Closed Loop Automation (CLA). CLA is the combination of closed loop stages that create automated processes which are based on feedback received from monitoring data.

CLA can manage the network reducing or even eliminating human involvement from the operation and management of the system. CLA in management systems can be implemented with the combination and chaining of management services (data, analytics, etc.), and it creates fully autonomous systems that are able to constantly monitor and assess the network and proceed to corrective actions when the goals are not fulfilled. This implies the presence of advanced technologies such as AI and Machine Learning (ML) used for the development of closed loops. Although the purpose of CLA is to reduce the direct human intervention, it is important for any autonomous system to allow interactions with human operators. Such interactions can be used for the specification and modification of the goals of the CL, as well as for monitoring the performance of the autonomous system and eventual approve or reject actions taken by it. [97]

2.2.6 Open security issues of ZSM framework

However, the ZSM standardization team has identified various security issues which should be addressed in order to take advantage of the benefits of this framework and avoid hazardous situations. The main security issues are discussed in [92] and are listed as follows:

- *Trust relationship between multiple management domains:* As new NPs form their management domain and embrace the ZSM concept, the collaboration among different domains in an automated manner requires a level of trust. E2E services in cross domain scenarios should be supported sufficiently regardless of the heterogenicity of the framework. Their proper operation is based on service level agreement (SLA) signed among NPs, which must facilitate the proper network conditions for the desired operation of E2E service.
- *Security risks introduced by the vulnerability of management function and security assurance of ZSM management function:* Since the core functionality of ZSM is based on management services, the possibility of a security threat breach in the operation of those functions would be catastrophic. Therefore, the immutability and the high security level of management functions is extremely important in ZSM networks.
- *Security isolation and security requirement fulfilment in multi-tenancy environment of ZSM framework:* The multitenant nature of ZSM networks

should not affect the security of services supported by virtualized resources. The isolation feature inherited by the virtualization technology that is used in modern networks, increases the security level which should be high in every tenant of the network.

- *Access control for management service provided by multiple domain service producers of ZSM framework:* taking into consideration that numerous NPs provide a management service, the access over this service should be controlled and supervised in order to identify any malicious activity and avoid service malfunction. The normal functionality of these services should be safeguarded since they are the heart of the ZSM framework.
- *Leverage existing security specifications to identify security risk of AI/ML model and protect AI/ML models in ZSM framework:* Although AI/ML are key technologies of the ZSM and increase the automation level of modern networks by introducing characteristics such as self-adaptation and self-optimization, their susceptibility in malicious attacks is a major issue. Models used in these technologies are trained using data sets which might be tampered. This type of attack is called dataset poisoning and may lead to wrong AI/ML decisions which are major threats for the framework's proper functionality [98].

In [92], the standardization team proposes countermeasures to overcome the security issues mentioned above. Regarding the trust relationship among entities, they propose a reflective and adaptive trust model to build mutual trust among entities in the ZSM framework. The goal of this process is to ensure the confidentiality, integrity, availability, and regulation compliance of every MD. To accomplish this, each entity that owns an MD needs to evaluate the trustworthiness of the other entity also owning an MD, based on threat and risk analysis and by examining the security policies applied in the entity. The outcome of this process leads to the building of a trust relationship among entities, followed by authentication procedures between parties and the formation of a secure channel where the behavior of each entity is tracked. Although this solution seems to tackle the trust problem, other approaches can also be investigated.

The safeguarding of management functions which are crucial for the operation of ZSM is addressed by authors in [92] using the GSMA Network Equipment Security Assurance Scheme (NESAS). This methodology defines security requirements and performs an assessment for secure product development and product lifecycle processes, using 3GPP's defined security processes for the evaluation of network equipment. Although this solution is tested for the security assurance of network equipment following the 3GPP's Security Assurance Methodology (SECAM), other technologies could be studied to protect management functions.

Additionally, the multitenancy issue is answered in [92] using policies applied to each tenant that uses the ZSM framework. The policy mechanism aims to provide a sufficient security layer for the users of the ZSM framework to avoid the exploitation of multi-tenancy which may lead to loss of sensitive data of E2E services and loss of frameworks' reputation. However, this solution is based on security requirements defined by authors in [92] and cannot by itself be considered as a high security level solution. Moreover, the access control of management services (MnS) is another major issue, as the exhaustive usage of management resources by a malicious entity may cause mis-operation of these critical services. Robust access control mechanism, including identification processes, authentication, authorization and audit of MnS usage, should be applied to prevent MnSs and other management resources of ZSM framework being misused by MnS consumers, according to authors. Considering that ZSM is implemented in a multi domain environment, the standardization team proposes techniques to enhance the security of MnSs by introducing authentication and authorization mechanisms, which check the trust relationship among entities in ZSM.

AI and ML are the main technologies used in ZSM framework and their reliability and robustness should not be left open to dispute. The decisions of AI/ML affect the ZSM network operation as they perform closed loop operations for the efficient deployment of E2E cross-domain services. The need of providing high security level to the components that implement AI/ML is highlighted by the authors in [92]. A comprehensive risk assessment for the AI/ML vulnerability issue based on the Adversarial ML Threat Matrix takes place and possible countermeasures are proposed. Nevertheless, there are more solutions that could be examined in order to increase the security of the system.

In the next chapters of this thesis, we propose the use of blockchain technology in ZSM scenario to address the security issues and challenges of the system and at the same time maintain the complexity in reasonably low levels. The goal of our research is to take advantage of blockchain's inherent characteristics to enhance the security and automation of ZSM's framework with respect to frameworks requirements.

3. Design, implementation and evaluation of a blockchain-based application in dynamic resource management of NGNs

The increased interest of researchers in the adoption of blockchain in modern networks and the main characteristics of this hyped technology urged us to design, implement and evaluate solutions for the resource management among competitive NPs in multi-domain scenarios. This chapter of the thesis presents the results of our study published in [70,79]. The main goal of our research is to check the feasibility of using blockchain for resource management in modern networks and therefore we proceed to the identification of specific metrics that play a significant role for the performance of the system. Such metrics are the success rate, the throughput and latency introduced by the blockchain network which are thoroughly discussed in the following subsections. Another major goal of our study is to examine how the consensus algorithm used in a blockchain network affects its performance. To this end, we conduct experiments using different consensus algorithms from different consensus families as it is illustrated in this chapter. The results of our study lead us to useful conclusions and encourage us to continue our research in using blockchain in NGNs.

3.1 Application of blockchain technology in dynamic resource management of NGNs

A blockchain-based solution for resource management in modern networks has been published in our paper [70]. The design and implementation of a trusted framework which provides the ability to the infrastructure providers to trade their computational and networking resources and to the service providers to negotiate, in real time, and purchase/access the resources with certain SLAs in a trusted environment is extremely valuable. The goal is to enrich the next generation networks with (re-) programmability and configurability, agile resource management optimizing network resource utilization while safeguarding user quality of service. The rapid growth of blockchain and the characteristics of this technology motivated researchers to examine useful use case scenarios in the area of Next Generation Networks (NGNs). More specifically, blockchain applications in 5G have attracted the interest of academia and industry, as many ideas have been published, [99–101]. Some of those ideas use the blockchain as an immutable database for storing crucial data (i.e., billing, roaming charges), while some others use this technology to guarantee that certain SLAs among systems' entities are met using Smart Contracts (SCs). Our research presented in [70] aims at: (a) investigating the suitability of blockchain/DLT technologies for flexible and distributed resource management in NGNs, (b) provide a definition of such a blockchain-enabled solution (c) the description of our evaluation testbed, and (d) the presentation of the initial evaluation results which prove its feasibility and current limitations. To serve this aim, we conducted a survey of existing (centralized) approaches, targeting flexible resource management in NGNs, as well as existing distributed blockchain-based approaches that have been proposed up to now. The blockchain-based solutions have been discussed in the previous section of the thesis. In the following subsections, we define the system architecture under consideration, and we present a blockchain-enabled solution based on the Ethereum-Quorum [102] platform, targeting inter-administrative island operation. Then, we present the test bed we deployed to evaluate the approach and present the first results accompanied by the conclusions of our study.

3.1.1 System architecture and use case scenario

Infrastructure operators are becoming totally separated from service providers, while the life cycle of each network service is becoming shorter and services become more and more demanding in terms of network dynamicity, computational capabilities, and flexibility [103]. In order to offer high quality services under highly varying load patterns due to high mobility and data-intensive tasks, the deployment of services over network infrastructures should be decided as dynamically and flexibly as possible and, more importantly, across the boundaries of networks belonging to different administrative areas.

The system architecture we are proposing is shown in Figure 11. We assume that several NPs exist, each operating a MANO instance to orchestrate the use of its own resources which consist of one or more NFV Infrastructures (NFVIs). In each one of the MANO instances, the corresponding monitoring component [104–106] is aware of the level of resource utilization, as well as of the quality of service experienced by the deployed services. Currently, this component can trigger the re-configuration of the resources that the specific MANO administers, including “leased’ resources which are statically defined upon agreement. For the different administrative areas to “trade” resources in real time, either a central trusted authority or a distributed solution should be in place. In Figure 11, a solution where each area (using a MANO instance) is maintaining a blockchain node (BC i) is shown. Each BCi shown in the figure is assumed to host the wallet of the blockchain solution and the blockchain node that contains the digital ledger. This entity is triggered by the MANO when a need for additional resources or the availability of resources is detected by the monitoring component. This way resource pooling across administrative areas becomes possible without the need for any trusted 3rd party.

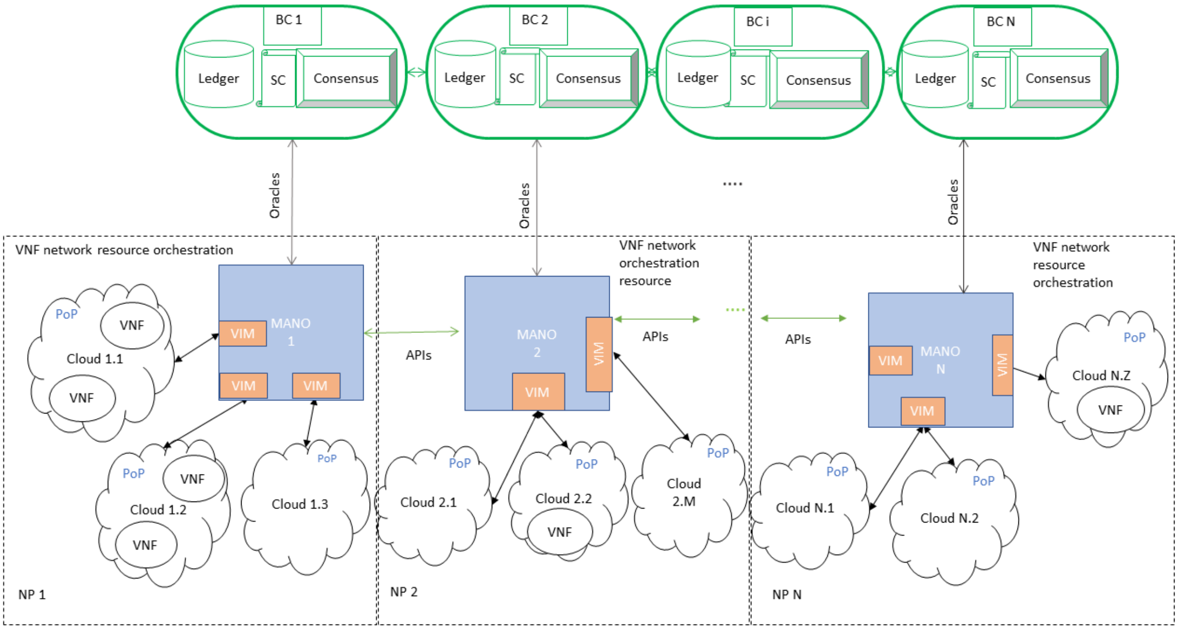

Figure 11. The proposed system architecture

The role of each component depicted in Figure 11 is presented in detail in the following bullets:

- *Blockchain nodes:* The blockchain nodes form the blockchain network and hold the digital ledger that contains all the history of transactions and information regarding the NPs. Each node has, also, access to the wallet of the NP that contains the keys of the node which are needed to access the blockchain network and make the required calls to the SCs deployed there. The blockchain network is responsible for the proper functionality of the resource management mechanism, that includes the trade of resources and their billing. Each NP supports one node as it is presented in Figure 11.
- *Oracles:* A blockchain oracle [107,108] is an entity that connects a blockchain with off-chain data. Oracles are known as blockchain middleware and enter every data input through an external transaction. To maintain the deterministic

validation of blocks, normally smart contracts can only access data previously stored on the blockchain and cannot use external data. The use of oracles makes communication possible from the external world to the blockchain, for example by recording external data on the blockchain in transactions. In the presented solution, oracles are used for the interaction of the blockchain network with the MANO components and the VNF network resource orchestration.

- *MANOs:* These components are responsible for performing the necessary actions for the implementation of the resource management. The resource management mechanism is implemented inside the blockchain using the SC and the MANO components execute the decisions derived by the blockchain network. Additionally, this component is responsible for monitoring the resource utilization of the virtual infrastructure and hosting images of virtual network functions (VNFs). The MANOs interact each other to reserve resources and implement the necessary network functions specified by the blockchain network which acts as a decentralized brokering system.
- *VIMs:* Virtual Infrastructure Manager (VIM) is responsible for managing the virtual infrastructures, usually cloud environments, and is hosted inside the MANO component. Through VIM, MANOs can manage these resources by launching, modifying, and terminating VMs that support various VNFs.
- *Clouds:* Cloud infrastructures offer computational resources to support various VNFs. These infrastructures are geographically staggered in order to cover regions and cities, trying to provide services near to customer.
- *VNFs:* The VMs are used for hosting the VNFs and consist of virtual resources such as VCPUs, RAM, storage, and network links (bandwidth). The resources used by the VM are based on the characteristics of the VNF.

The proposed solution introduces the benefits of Blockchain (BC) technology in a federated environment consisting of NPs. The basis of the solution is a Smart Contract (SC) written in solidity and deployed in a Quorum network. Quorum is a fork of Ethereum and was selected because it can support private transactions using the Tessera tool [109] and can be easily implemented using different consensus mechanisms. The SC consists of three main functions:

a) *addNetworkProvider:* This function is triggered each time a new NP joins the network. For every NP, the information kept in the BC includes: the name of the NP, the types and number of its offered resources, e.g., bandwidth, processing, memory, the cost of the resources per unit, other attributes of the resources like the region they cover and (most importantly) the Service Level Agreement (SLA) that the NP can support. A unique blockchain account address is also associated with each NP and is used as a wallet for interacting with other NPs and entities of the blockchain. This function performs a transaction and inserts the result in the ledger which is kept in the blockchain.

b) *GetBestMatch:* When an NP needs additional resources to satisfy the needs of its users, it searches the BC network in order to find another NP that can offer these resources. The GetBestMatch function is triggered in this case. It takes as an input the type, number and attributes of the needed resources and searches to find the NP that can fulfil them. In case more than one NPs can satisfy the request, the one incurring the lower cost is selected. It is worth stressing here that (a) our focus is not on the optimization algorithm but on the evaluation of the feasibility of such a solution offering adequate performance, and (b) the “cost” that we assume in the proposed solution can be the actual financial cost or any other metric, whose value is designed to be minimized. It should be noted that this function reads data from the blockchain and does not write any information in the ledger.

c) *ResourceReservationTransaction:* Once the GetBestMatch function ends up with the id of the NP that offers the required resources at the lowest price, the ResourceReservationTransaction is triggered so that the decision and relevant payment are enacted. The NP that has requested resources pays the amount specified by the cost field of the NP who offers the resources and the resources of the provider that purchased the resources increase. The balance of the NP that has offered resources increases and the transaction is completed. This transaction function writes data into the blockchain.

In a commercial solution, an additional function that will trigger the release of the resources would be implemented. It should be noted that, the use of cryptocurrency for the billing is not examined in this solution but is a mechanism that could be included as a feature in future extensions. The billing process of the solution uses as input the utilization of resources and the features of those resources described by the SLA.

In order to illustrate the functionality of the proposed solution, a use case scenario is described as follows. Each NP registers into this solution by using the addNetworkProvider function of the SC and becomes member of the blockchain by maintaining a node. Each NP uses its own resources to support its own customers and the resources that are not in use are available to the network. When a NP needs resources to cover an increased demand, creates a call at the GetBestMatch function of our Smart Contract to select the proper NP among candidates. The proper NP is the one that offers the required resources in the lowest price. The outcome of the GetBestMatch function is used by the third function (ResourceReservationTransaction) in order to initiate a transaction between the NP that has requested resources and the one that offers them. When the transaction function is triggered, the NP that has requested resources (NPreq) transfers an amount of digital money, which are called ethers for our Quorum implementation, to the NP that provides (NPprov) resources. The NPprov lends resources to the NPreq and immediately uses an Oracle mechanism in blockchain terms to implement the necessary services using the MANO components. MANO components are responsible for managing (e.g., launching, terminating) the required network services using the resources specified by the SC. The computational resources are offered by the cloud infrastructures through the VIMs (Virtual Infrastructure Managers) and are used for the creation of VMs which support the network services described by MANO. When the NPreq does no longer need the borrowed resources, these resources are released back to the original owner while the responsible MANO terminates the reserved services.

Having in mind that next generation networks use microservices and the lifetime of those services vary (from ms to seconds or even minutes) the above process should be performed with the minimum latency while the number of supported transactions should be high. Therefore, the next section evaluates this solution and checks its feasibility and its performance in matters of latency and throughput. It is worth mentioning that the current work illustrates a proof of concept of a blockchain application for dynamic resource management in next generation networks.

3.1.2 The experimental testbed and the evaluation results of the blockchain-based solution

To evaluate the proposed approach, we have deployed a custom Quorum network. Useful information regarding the implementation of our testbed is presented in

Appendix A. The goal of this section is to evaluate the feasibility of this solution and its performance in terms of transaction latency and transaction throughput, as well as the number of transactions that can be handled by the network to check the load burden. The test bed that we set up included three nodes which are not hosted on the same physical machine (as is done in all the surveyed articles discussed in the literature review section) but in machines interconnected through the Internet. We also deployed a fourth VM used for running Hyperledger Caliper, which is our benchmarking tool [110]. The Hyperledger Caliper tool is one of the most popular benchmarking solutions for blockchain applications. Caliper uses an adapter to connect to the System Under Testing (SUT) which, in this case, is the private Quorum network based on the RAFT consensus. Figure 12 illustrates the basic components used in the presented experiment. The Load Generator produces the load applied to the SUT, while the Configuration file describes the experiment. The Adapter is used for the interaction with the SUT which in our case is the Quorum network. The outcome of the whole experiment is the report file which contains information related to the behavior of the network. The Quorum Nodes that form the network are hosted in the Okeanos cloud [111] infrastructure offered by GRNET. The characteristics of the VMs are:

- Operating system: Ubuntu 16.04 LTS server,
- 4 CPU cores,
- 8 GB RAM,
- 30 GB storage, and
- public IP addresses

The performance evaluation was conducted as a function of three different parameters which were configured through a YAML file. These parameters included: (a) the number of workers, (b) the rate controllers, and (c) transaction number (txNumber). The workers are docker containers which generate the workload in the network. The rate controllers are two parameters affecting the rate at which load is inserted in the blockchain network. They take under consideration TPS which is the number of transactions to be sent in a second and txDuration which specifies the duration till which we will be sending the transaction. The txNumber is the number of transactions to be executed and represents the amount of transactions initiated when the functions of the SC are executed. In our experiments, we used one worker and the txNumber was set at very high values to ensure that we measure the steady state at different input loads (different TPS values).

The output metrics of the test bed which we measure are three: (a) Success or Fail of a transaction, (b) the average Transaction Latency (s) and (c) the average Transaction Throughput (TPS).

a) Success or Fail: Once a transaction has been successfully proposed, verified and inserted to a block is considered as a success.
b) Transaction latency: The time elapsed between the submission of a transaction to the time the transaction has been verified and inserted into the blockchain. Once the transaction has been inserted to the blockchain, it is available to all nodes of the network. The transaction latency is measured in seconds and can be described by the following equation:

$$Transaction\ Latency = Confirmation\ time(s) - Submission\ time(s)$$

c) Transaction throughput: The rate at which valid transactions are committed into blocks in the blockchain and become available across all nodes of the blockchain. Throughput is measured in completed transactions per second (TPS) and can be described by the following equation,

$$Transaction\ Throughput = Total\ commited\ transactions/Total\ time\ (s)$$

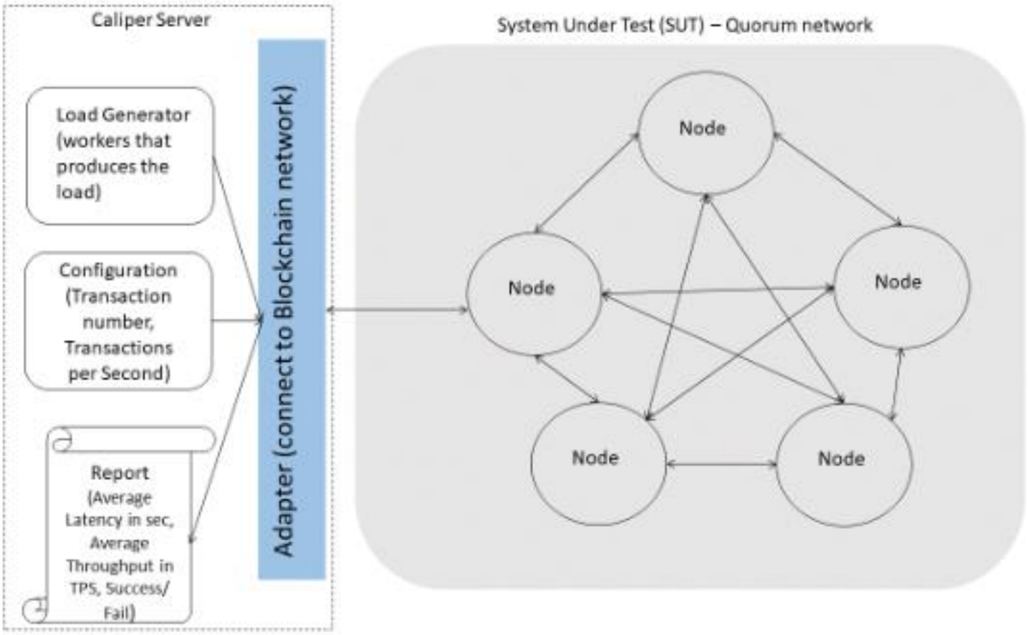
Caliper Server
Load Generator (workers that produces the load)
Configuration (Transaction number, Transactions per Second)
Report (Average Latency in sec, Average Throughput in TPS, Success/ Fail)
Adapter (connect to Blockchain network)
System Under Test (SUT) – Quorum network
Node
Node
Node
Node
Node

Figure 12. The experimental testbed

We have run a set of scenarios changing the number of (input) transaction per second parameter per smart contract function to assess (a) how many transactions the solution can handle, (b) the time required for a resource transaction to be decided and stored in the blockchain and (c) on the processing and memory resources required for the implementation of the solution.

For the first metric of interest, throughput (i.e., transactions stored in the blockchain per second), the results are shown in Figure 13. As the number of transactions (submitted to the system) per second increases the throughput (i.e., transactions that were successful and stored in the blockchain) increases as well. This is true up to 10TPS while from this point on, the number of transactions stored in the blockchain (reflected in the vertical axe) do not increase any further. The fail is attributed to the continuously increased latency of each transaction to be successfully executed. The response time of the transaction exceeds the Caliper's acceptable time limit and, therefore, is characterized as failed. As a result, the number of failed transaction increases inevitably as the (input) transaction rate becomes higher.

This is also proven by the results for latency, which are shown in Figure 14. The latency (in seconds) of the two functions, that do not require any reading or processing (addNetwork provider and Reservation transaction mentioned in the figure as transaction), is kept very low irrespective of the TPS. This is not the case for the getBestMatch which requires significant processing. For the getBestMatch function, the latency is very low as soon as the TPS is below 4 and increases to 15s when TPS becomes 10.

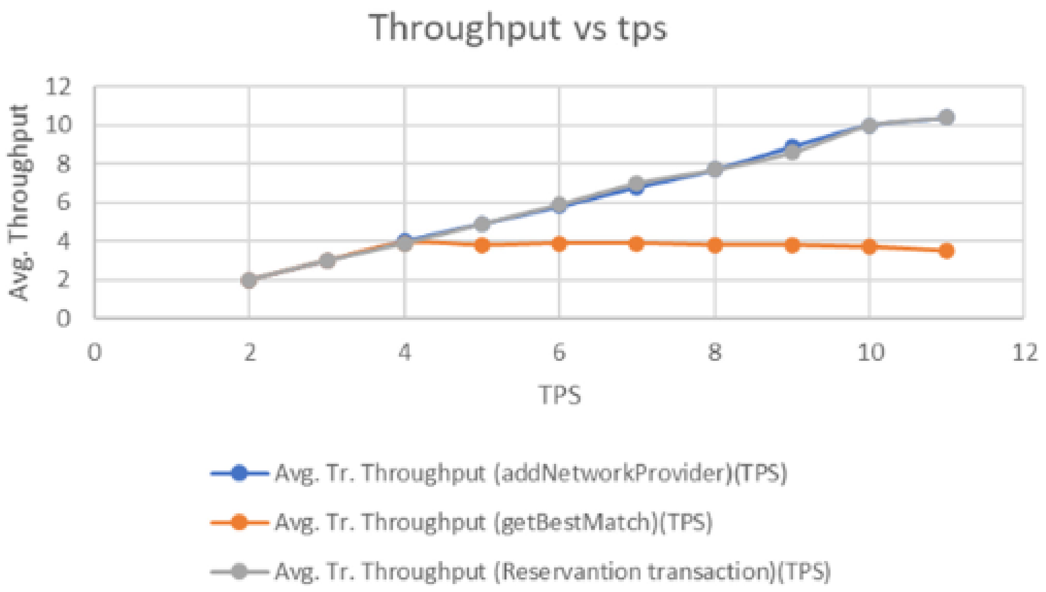

Figure 13. The throughput per SC function.

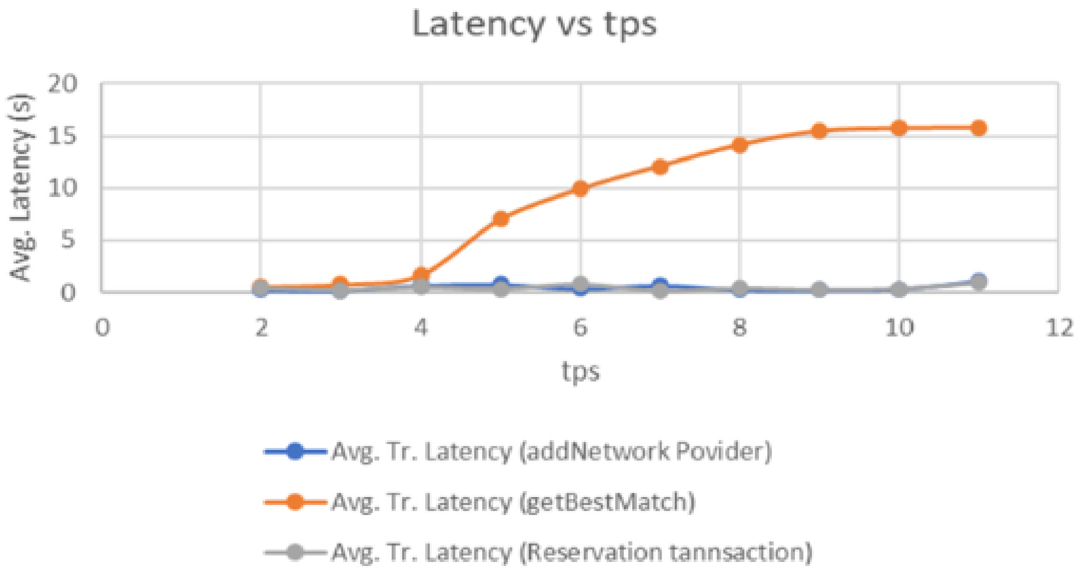

Figure 14. The latency per SC function for different transaction loads.

This is a very important result. It means that if a group of NPs decide to allow for up to 4 resource reallocations per second, the re-configuration of the resource allocation will be decided in less than 1s which is a very low latency result and leads to agile network reconfiguration. When the number of TPS increases to 10 the situation becomes like the one expected with more NPs joining the system or having fewer providers but with more than one resource requests per second. In this case, the latency becomes 15s which can still be considered acceptable assuming these requests correspond to pipes of traffic among service providers and not as single end-user services. To consider flows of finer granularity, we need to improve the solution. From the value of 10TPS and above, fails in the transaction occur (5% at TPS equal to 10 and rising with TPS). So, the presented solution can offer adequate performance up to 10 TPS. In both latency and throughput results, we presented the average values i.e., the average over the multiple runs we executed.

With respect to the resources needed for the implementation of the solution, in the aforementioned test bed, a max CPU utilization of 35% and of 1.5 GB memory is measured which is definitely affordable. Finally, the results produced in this section show that the impact of the SC on the Quorum blockchain network is not significant in terms of latency and throughput. The benefits provided by introducing the blockchain technology and the results of the presented evaluation show that the use of blockchain technology for resource management is feasible and promising.

3.1.3 Conclusions of our initial attempt in developing a blockchain based resource management mechanism for modern networks

Summarizing the work presented so far, the idea of adopting distributed, blockchain-enabled solutions in modern networks has triggered the interest of research community. The application of blockchain technology adds valuable characteristics to modern networks as it forms a network of trust among participants, while it guarantees the integrity of the information stored and used by the system.

In the presented solution, we proposed the use of a distributed broker mechanism by describing an architecture that includes the NPs as nodes, oracles for interaction within and out of the blockchain network and wallets to send and receive transactions. MANOs, VIMs, VMs and cloud instances provide for a complete view of the overall architecture that aims to showcase the strength of distributed solutions following a described use-case scenario that underlies its potential. Following the described architecture and the use case in hand, basic blockchain characteristics such as transparency, immutability, non-repudiation are examined as to whether they can provide for a safe multitenant environment for the NPs to perform resource management processes without relying on a trusted, centralized third party. Additionally, this concept opens the road for the formation of new business models between NPs which can reduce their cost and at the same time optimize management of their resources. The proposed solution requires from each of the NPs to host a blockchain node, to support the presented logic. In contrast to other related works, this study describes and evaluates the blockchain based resource management solution and produces results regarding its feasibility and performance by applying Caliper, a well-known blockchain emulator, to perform this task. The results show that the cost of the solution is more than affordable on one hand, while on the other the achieved performance, even when the solutions is not optimized, is adequate. According to the results, the latency of resource reconfiguration decisions remains below 15s for high loads, and the throughput is also adequate.

However, the proposed approach can be further developed and reevaluated by examining the behavior of the system in a larger blockchain network with more nodes to test the scalability of the solution. Also, changes should be made in the structure and logic of the SC to implement more sophisticated resource allocation algorithms and to evaluate their impact on the latency. It should be mentioned that the code of SC's functions affects the performance of the system according to our previously described experiments. Additionally, the impact of the adopted consensus algorithm should be also studied by applying different consensus mechanisms and test the performance changes on each one of them in an effort to identify an optimum implementation for our solution. To this end, we continue our study published in [79] which is presented in the following sections of this chapter.

3.2 Development and re-evaluation of a blockchain-enabled resource management mechanism for NGNs

Based on our previous study and the architecture illustrated in Figure 11, we proceed to changes in the structure of the SC and we create a new blockchain network that consists of more nodes (5 nodes) than the one used in our initial attempt to test the scalability of the solution. Moreover, we operate the blockchain network using two different consensus algorithms (i.e., Raft and the Istanbul Byzantine Fault Tolerant-IBFT) and we examine its performance in terms of success rate, throughput and latency. Then based on the evaluation results we suggest the most suitable consensus algorithm among these two for this particular use case scenario and we present useful conclusions regarding the performance and operation of this solution.

3.2.1 SC version 2: Analysis of its structure and functionality

The blockchain enabled resource management mechanism is implemented through a SC that consists of functions written in Solidity as it illustrated in Pseudocode 1, which is a language used for the creation of SC in Ethereum-based blockchains. In Appendix B a useful solidity file is presented as well as python scripts used for in experiments. The presented SC is deployed in a Quorum network which is a variation of Ethereum. Quorum is ideal for the creation of private networks and supports various consensus algorithms, which is the main reason for its selection. There are three main functions in this version of the SC: addNetworkProvider, requestResources, returnResources. It is worth mentioning that every function writes in the digital ledger of the blockchain. The role of each function is described as follows:

- *addNetworkProvider:* This function is similar to the one defined in our initial approach and is triggered by the administrator entity of the system to insert a new Network Provider (NP) to the blockchain. The NP is described by the following features: a) Name: the name of the NP, b)Computational resources: these resources consist of the amount of CPU, RAM, and storage the NP offers which change over time based on their utilization; c) Cost: the cost of the resources offered by the NP defined as the cost per resource; d) Domain: the area where the NP can offer the resources,; e) SLAs: the Service Level Agreements (SLAs) a provider can guarantee; f) VNF images: the Virtual Network Function (VNFs) a provider can support; g) Address: the blockchain address associated to the NP, which is used for implementing transactions in the blockchain network. The SLA describes the requirements that should be met when the resources are offered. More specific, characteristics such as latency, throughput and packet loss tolerance are defined in this field.
- *requestResources:* The NP who needs resources triggers this function that searches the ledger to find the NP who meets certain criteria. The criteria are based on the features analyzed above, while the requester sets the desired values of these properties. Moreover, this function uses another variable to set the time of using the resources. Summarizing, the request of resources contains the following attributes: a) Computational resources: amount of CPU, RAM, Storage; b) Domain; c) SLA; d) VNF image and e) Lend time. The execution of this request may return more than one results. In that case, the cheapest NP is selected based on the cost value. Then a transaction is initiated among the NP who called this function (requester) and the selected NP

(supplier), where the supplier lends the defined number of resources to the requester for a specified time period. The requester prepays the supplier based on a cost function that takes into account the total amount of resources lent, the lend time period and the cost value. It is worth mentioning that this function includes mechanisms to ensure that the requester has the necessary balance to execute this transaction. If the requester does not have enough balance, then the transaction is reverted.

- *returnResources:* this function is executed when the predefined time has passed and is responsible for returning the lent resources to the original owner. This function includes an oracle mechanism to check the time and then it uses the values contained in the request transaction to return the correct number of resources to the original owner. The properties used are: a) Supplier ID: is the id of the provider who offered the resources and corresponds to a blockchain address assigned to this particular NP; b) Computational resources: amount of CPU, RAM, Storage and c) time: the time when the requestResources function was validated, which is used to check if the time has passed or not. If the time has not passed the transaction is reverted.

```
  function addNetworkProvider( _name, _cpu, _ram, _storage, _cost, _domain, _slas, _vnfImages, _address) public {
   networkProviders[networkProviderIndex].name = _name;  networkProviders[networkProviderIndex].cpu = _cpu;
   networkProviders[networkProviderIndex].ram = _ram;   networkProviders[networkProviderIndex].storage = _storage;
   networkProviders[networkProviderIndex].cost = _cost;   networkProviders[networkProviderIndex].domain = _domain;
   networkProviders[networkProviderIndex].sla = _slas;  networkProviders[networkProviderIndex].vnfImage = _vnfImages;
   networkProviderToOwner[networkProviderIndex] = _address; ownerToNetworkProvider[_address] = networkProviderIndex;
   networkProviderIndex = networkProviderIndex + 1;   }
  function requestResources( _cpu, _ram, uint8 _storage, _domain, _sla, _vnfImage, _time) public returns {
     uint i = 1;  uint bestNetworkProvider;
     while( i < networkProviderIndex ) {
        if ( networkProviders[i].cpu >= _cpu && networkProviders[i].ram >= _ram && networkProviders[i].storage >= _storage && networkProviders[i].domain == _domain){
           if ( getBestSla(i, _sla) == true && getBestVnfImage(i, _vnfImage) == true) {
              if ( bestNetworkProvider == 0 ) {
                 bestNetworkProvider = i;
              }
              else if ( networkProviders[i].cost <= networkProviders[bestNetworkProvider].cost ) {
                 bestNetworkProvider = i;}
           }
        }
        i++;
     }
  require(msg.value >= calculateBestCost(bestNetworkProvider, _cpu, _ram, _storage, _time), "The Ether was not enough (3)");
     uint j = 0;
     while (providerResourcesTime[ownerToNetworkProvider[msg.sender]][bestNetworkProvider][j] != 0){
        j++;  }
     transfer_resources(msg.sender,bestNetworkProvider);  withdraw(networkProviderToOwner[bestNetworkProvider]);
  }
  function returnResourcses( _id,  _cpu,  _ram, _storage, _time) public returns{
require(providerResourcesTime[ownerToNetworkProvider[msg.sender]][_id][_timeId] != 0, "The users did not made any transaction (1)");
     require(providerResourcesTime[ownerToNetworkProvider[msg.sender]][_id][_timeId] <= block.timestamp, "The time has not passed yet (2)");
           transfer_resources(msg.sender, original_owner);  }
```

Pseudocode 1. The functionality of the SC

3.2.2 Candidate consensus algorithms

One of the main elements of blockchain that has a significant impact on the network's performance is the underlying consensus mechanism. In this study, we focus on two different consensus mechanisms: Raft and IBFT. These two consensus mechanisms were selected because both can be applied in consortium blockchains and perform better (faster block time, higher fault tolerance) than other popular mechanisms, such as PoW and PoS. Moreover, focusing on the impact of consensus, in this solution we decided to maintain the same blockchain characteristics (i.e., number of nodes, blockchain platform, SC structure) in order to facilitate a fair comparison of the consensus mechanisms. On the one hand, Raft [112] is suitable for consortium blockchains where byzantine fault tolerance (BFT) is not a requirement and the key characteristics that should be met is the fast block generation times and the transaction finality. It is worth mentioning that there is no creation of empty blocks in Raft, as it creates blocks on demand. This consensus mechanism is member of the crash fault tolerance algorithms, like Paxos [113], which can guarantee that if a subset of nodes in the decentralized system goes offline the same state of truth is maintained. On the other hand, IBFT [114] consensus mechanism is suitable for private/consortium blockchains where the byzantine fault tolerance is a requirement. This algorithm is member of the BFT consensus family and inherits from the pBFT the 3-phase consensus, PRE-PREPARE, PREPARE and COMMIT. IBFT can tolerate at most F faulty nodes in a N validator network, where $N = 3 \times F + 1$. In addition, using this mechanism no forks can be implemented and all valid blocks are appended in the main chain. It should be noted that although IBFT can tolerate the byzantine problem, the block generation times are higher than Raft's because of the use of the 3-phase feature and the BFT characteristic.

The selection between these two consensus algorithms in the considered use case is not easy as there is a tradeoff between security on one hand and speed and fault tolerance of the network on the other. Towards a qualitative comparison, the following should be taken into consideration. NGNs must offer network services to support intensive applications on demand, which means that transactions should be verified very fast, and the block generation time should be low. In addition, the crash fault tolerant attribute is vital for this solution as NGNs support critical applications. Moreover, the use of a consortium blockchain which consists of NPs added by an administration entity, although it reduces the sentiment of decentralization, it also reduces the possibility of the participation of malicious nodes. Furthermore, the identity of each NP is known, which is a fact that discourages NPs from performing malicious actions. As a result, we argue that Raft consensus is more suitable than IBFT in this qualitative approach. However, in the next section the quantitative evaluation of these two consensus algorithms takes place in order to assess if the above rationale is justified by the results and the actual performance limits.

3.2.3 Description of the testbed and experiment methodology

In this section, we aim to evaluate the performance of the blockchain-based marketplace in terms of transaction throughput, latency, and success rate. We first present the testbed that we have set up and then present the results. Our evaluation testbed adopts the architecture described in Figure 11. In this experiment, we deploy the SCs analyzed previously to a Quorum network, and we measure their performance under the two different consensus mechanisms in study. The experiments are conducted in two different Quorum networks characterized as Systems Under Testing (SUT). The one is using the Raft consensus mechanism and the other IBFT. The Quorum Raft network consists of five nodes in total, where four nodes are hosted in Okeanos cloud while the other one is hosted in an OpenStack infrastructure in University of West Attica premises (around 500km away) in a dedicated VM. The Quorum IBFT network consists of five nodes as well with the same deployment characteristics. All the VMs that host the blockchain nodes have the same characteristics:

- Operating system: Ubuntu 16.04 LTS server, Ubuntu 20.04 respectively,
- 4 vCPU cores,
- 8 GB RAM,
- 30 GB storage, and
- public IP addresses

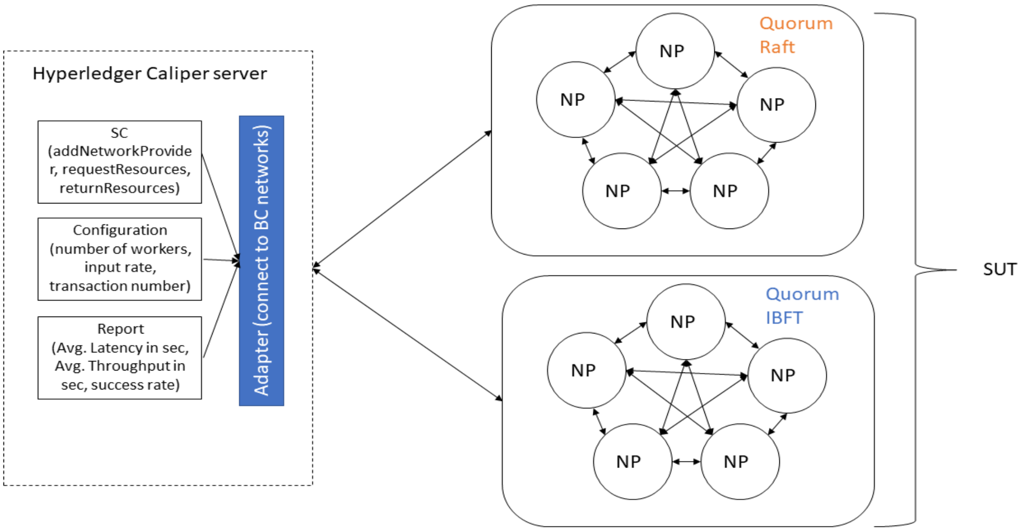
Hyperledger Caliper server
SC (addNetworkProvider, requestResources, returnResources)
Configuration (number of workers, input rate, transaction number)
Report (Avg. Latency in sec, Avg. Throughput in sec, success rate)
Adapter (connect to BC networks)
Quorum Raft
Quorum IBFT
NP
SUT

Figure 15.Testbed overview

In both networks, the previously described SC is tested. The blockchain nodes communicate via internet and no internal network is used. The scope of our experiment is to create small, private, geographically distributed blockchain networks which use the Raft and the IBFT consensus respectively and extract useful information in terms of throughput, latency, and success rate. The SC and the behavior of the blockchain network are tested using the Hyperledger Caliper tool, hosted in a VM on Okeanos cloud.

As presented in Figure 15, this tool (Caliper) connects to blockchain networks using a specified adapter compatible with Ethereum and runs tests based on a configuration file created by the user. In our experiment, we focus on the input transaction rate (denoted as ITR). This is the rate at which these transactions are proposed to the blockchain network and is measured in input transaction number per second. Caliper offers controllers which regulate the input transaction pattern. In our experiments we use the fixed rate controller for several ITR values to evaluate the behavior of these two blockchain networks and examine the impact of consensus to the systems' performance. At the end of each experiment round, Caliper produces a report where the Average Transaction Throughput, the Average Transaction Latency and the Success rate are displayed. The same experiments were conducted in both networks and we proceed to the presentation and comparison of the collected results. It should be noted that thanks to Caliper, we managed to check the performance of each function of the SC separately and display the outcome in the following figures.

Evaluation results

We first focus on the success rate for different values of ITR which is depicted in Figure 16. It is obvious that Raft outperforms IBFT. As we can see, the success rate drops below 100% for IBFT when the ITR is 5 transaction per second while for Raft the success rate remains high for significantly higher ITR values. This was expected due to the crash fault tolerance characteristic of Raft mechanism and the low throughput presented in the IBFT testbed when the ITR increases. Low throughput means low number of transactions that can be validated in a second.

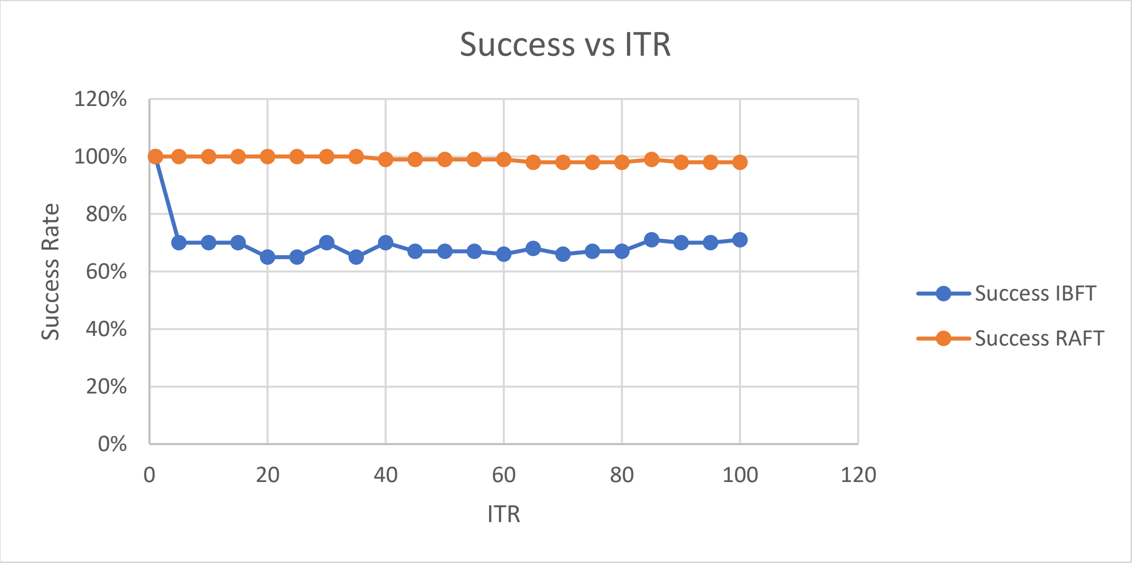

Figure 16. Success rate vs ITR for Raft and IBFT

We then present the throughput rate shown in Figure 17Figure 18Figure 19. For the selected consensus algorithms, we first try to shed light to each of the functions of the SC. The function addNetworkProvider and returnResources behave similarly although the latter presents higher throughput. However, the requestResources function presents a very low throughput as the ITR increases which is caused due to the nature of the function. This function performs recursive queries to find the most suitable NP candidate and select the cheapest one. This process requires more time than the other two functions of the SC. Therefore, we observe high latency in both networks when we focus on the requestResources function. In our experiments we maintain the same ITR for all functions executed in the Raft and IBFT testbed in order to perform a fair comparison. Nevertheless, it is worth mentioning that when we perform an experiment for low ITR value (i.e. ITR =

2) the requestResources function performed better and none failed transaction was presented.

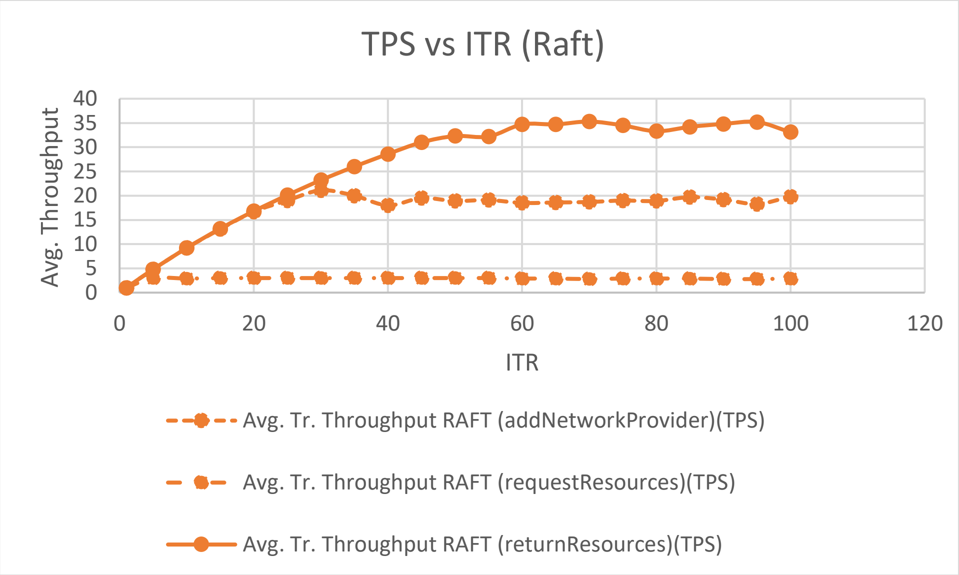

Figure 17. Throughput vs ITR in Raft testbed

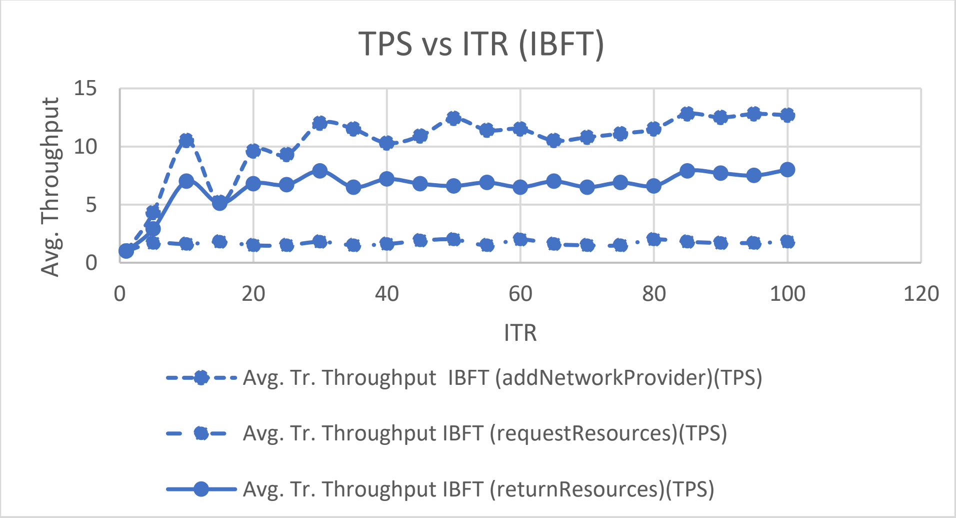

Figure 18. Throughput vs ITR in IBFT testbed

Turning our attention to the comparison of the two consensus algorithms, it is evident that for IBFT the throughput is lower which has caused the high number of losses. For this comparison we take into account the function with the worse performance because this affects the overall performance of the solution. For Raft, the throughput increases with the ITR as expected. From ITR equal to 40 and above, the increase is not linear which indicates that the blockchain network can sustain 40 transaction per second. It is obvious that Raft performs better than IBFT as it presents higher throughput values comparing the performance of the function in each testbed. Figure 19 shows that IBFT is exhibiting half throughput compared to Raft.

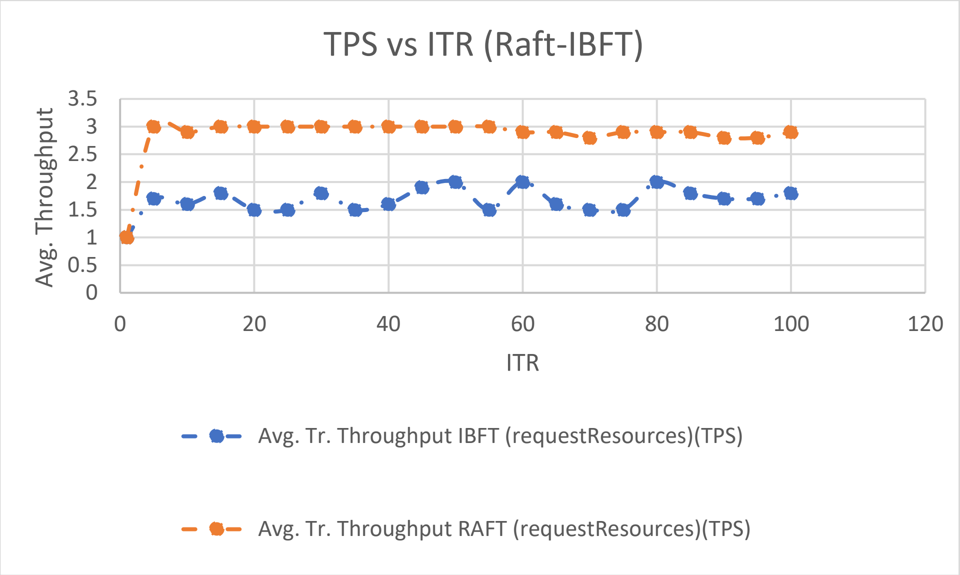

Figure 19. Throughput vs ITR focusing in requestResources function

We finally proceed to the latency which is presented in the following figures for different values of ITR. These figures illustrate the performance of the functions of SC for the two different networks. The blue color corresponds to the IBFT network while the orange represents the Raft. A first observation is that the requestResources function displays higher latency values than any other function of the SC regardless of the consensus used. The functionality of the requestResources described in the pseudocode in the previous section, is more intensive than any other function as it was discussed previously. As such, the latency of this function increases with ITR. The situation is different for the rest two functions which experience almost fixed latency, which depends on the consensus used in the system. IBFT has significant higher latency values than Raft. This was expected due to the characteristics of these two consensus mechanisms described in the previous sections.

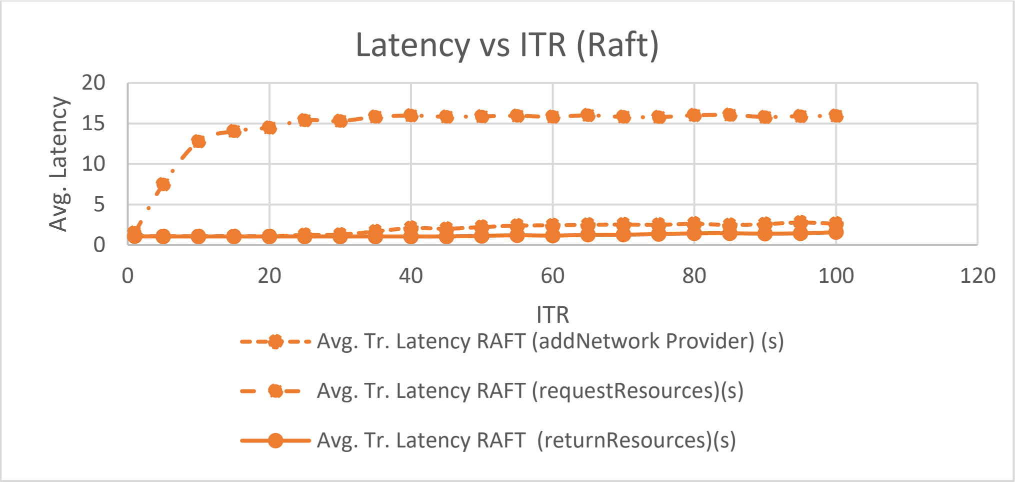

Figure 20. Latency vs ITR in Raft testbed

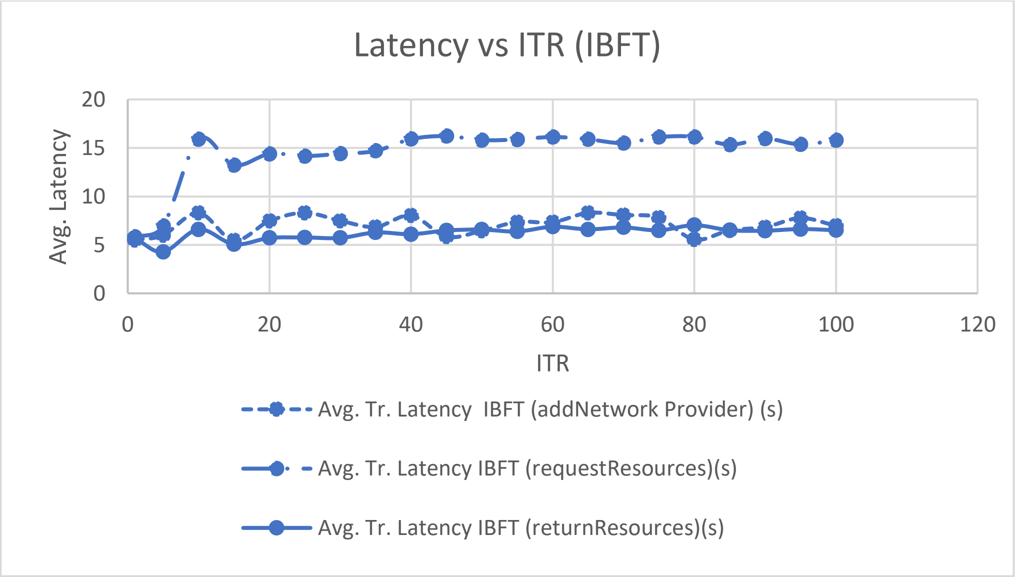

Figure 21. Latency vs ITR in IBFT testbed

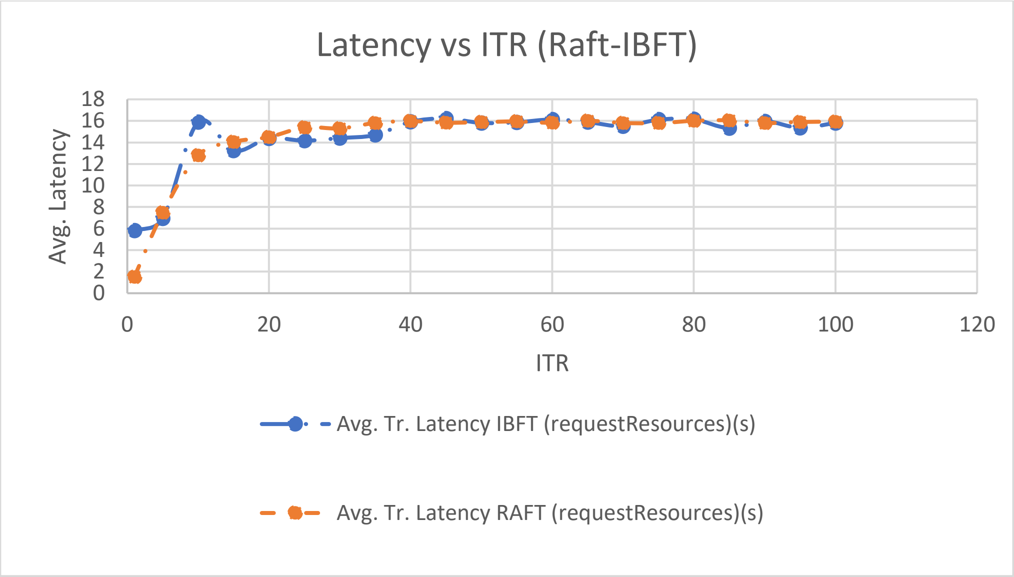

Figure 22. Latency vs ITR in the requestResources function

Raft achieves significantly higher TPS values than IBFT. The behavior of the requestResources function is expected due to the high latency it introduces. The high latency and low throughput values combined with high ITR lead to transaction failures. In Figure 16, IBFT presents lower success rate than Raft. Raft maintains 100% success rate due to the fact that it belongs to the crash fault tolerant consensus family.

The results of the experiment are in line with the theory considering the nature of these two different consensus mechanisms described previously. Consequently, the most suitable consensus among these two is the Raft as it presents better results than IBFT. It is important to be able to compare the latency to the service lifetime so that we can decide whether such a solution can work considering resource negotiation per micro-service or per aggregate. For the two SC functions and for RAFT the latency is in the order to seconds which means that the resource request can be performed on a per service basis in the future. This is not the case for the requestResources function which is about 17s. The latency for both consensus algorithms for this function is almost the same because the processing time dominates the consensus algorithm execution time. Therefore, from the perspective of the solution designer, it is important to either further optimize the code of the SC regarding the request resources transaction function, or to have this function executed outside the blockchain and register in the blockchain only the result of the function.

To improve the situation, the use of oracles and the development of oracle-enabled services is suggested. The oracle acts as a middleware that connects the blockchain world with the outside services in a secure manner. Recently, many oracle mechanisms have been developed [115,116] and can be characterized based on the data source, the trust model, the design pattern and the interaction with the blockchain [117]. Focusing on the security and integrity of the oracle, the examination of the trust model that should be adopted is crucial. There are two main categories: the centralized trust model (which use a mechanism to prove the authenticity of the data they exchange with the blockchain network) and the decentralized, which use many oracles which in turn use consensus mechanisms to safeguard the interaction with the blockchain. However, although the latter seems to be closer to the nature of blockchain, it adds latency to the system since it introduces an extra consensus mechanism.

3.2.4 Conclusions of the evaluation of the blockchain-enabled resource management mechanism using different consensus algorithms and modified SC structure

One of the most interesting research topics is the role of blockchain technology in NGNs and how it can contribute to the evolution of networking sector. The use of blockchain for the efficient resource management in modern NGNs marketplaces has triggered the interest of industry and academia as many works have been published. This research focuses on this field of study and presents a blockchain-based solution which is implemented in a SC. This SC is deployed in a blockchain environment and presents a resource management scenario in NGNs marketplace. In contrast to other related works, we proceed to the examination and testing of two consensus mechanisms Raft and IBFT to identify which is the most suitable for this use case. The main metrics we focused on are the transaction throughput, transaction latency and the success rate. The experiments were conducted and described in detail in order to justify the selection of the most suitable consensus and check the feasibility of this solution. After the evaluation of the results derived from the experimental process, we proceed to the identification of the points that need to be improved.

NGNs are responsible to provide high quality network services on demand with features such as ultra-low latency and high throughput. Therefore, the resource management process is extremely sensitive to time variations and latency. After the evaluation of our experiments, we may claim that the use of an AI-assisted prediction mechanism could improve the overall performance and increase the feasibility of the solution. This radically changes the scene compared to the traditional resource management, which was performed in each network sectors separately, putting emphasis and implementing intelligence in each domain considering the domain resource inelastic [118]. Moreover, the use of oracles in the blockchain can lead to the development of more efficient applications which can interact with many other web services. The introduction of such mechanisms and their impact on the systems’ efficiency is a topic that we will examine in our future work as well as, the combination of AI and blockchain in NGNs. In our case, an oracle service could be an AI-assisted prediction mechanism which could be used in this scenario to reduce the overall time needed for a network service to be offered. The idea is the use of a prediction mechanism that will notify NPs about the upcoming network demands. Then NPs could trigger the SC’s functions in time and acquire the necessary resources.

4. A blockchain-based ZSM approach

The results of the experiments conducted in our previous studies showed that the use of blockchain for resource management in NGNs is feasible and can be the answer to major security issues identified in modern networks. In this chapter, we examine how blockchain can be used to enhance the security and trust of the resource management process, supporting high levels of automation and dynamicity. More specifically, we propose the use of blockchain technology in the ZSM framework presented in the second chapter of the thesis to solve its main security issues. The security issues identified by the standardization team mostly derive from the lack of trust among network providers. Although the standardization team presents several security techniques to strengthen the security of the system, we argue that our approach provides a less complex solution using the blockchain technology and the goods it provides. In this chapter we present our approach to tackle the security issues of ZSM in a less complex manner by combining ZSM and blockchain technology. After presenting our proposal at high level, the requirements that a DLT-based solution should meet are clearly identified to provide guidance to prospective designers/users of such solutions. The rationale behind providing this list of requirements is that new DLT approaches are continuously emerging and selecting one today may not prove the best choice. This study is complemented by the exploration of a set of currently available approaches to provide further insights on the topic.

4.1 Exploiting blockchain to address ZSM’s security issues: Analysis and architecture overview

ZSM is expected to become one of the dominating frameworks of NGNs according to ETSI which presents a reference architecture of the framework in [91]. This architecture, analyzed in the second chapter of the thesis, enables the definition of the functionality and of the requirements that should be met in any ZSM implementation. In a multi-stakeholder scenario, a Management Domain (MD) is usually the administrative area of an NP that is responsible for the proper functionality of services running in this area. When E2E cross-domain services are deployed, the ZSM framework should guarantee the proper collaboration of MDs in order to support the E2E service with appropriate resources.

One of the main factors that affects the performance of the service is the time needed for the management tasks to be completed. The management tasks related to the deployment of E2E services should be executed within the limited processing time according to the ZSM reference architecture requirements. Functional and non-functional requirements defined by the ETSI standardization team and presented in chapter 2 of the thesis, determine the successful operation of the framework. The satisfaction of those requirements ensures the efficient operation of the network and allows modern networks to achieve high performance and support demanding applications.

However, the security level of the framework is extremely crucial for its proper functionality and the smooth operation of the entire network. In order to reduce or even eliminate the possibility of a malicious action that may lead to hazardous situations, a security assessment has been conducted and the identification of the main security issues of the ZSM framework are highlighted. To this end, the standardization team has identified the main security issues of the framework discussed in chapter 2 of the thesis and listed as follows:

- Trust relationship between multiple management domains,
- Security risks introduced by the vulnerability of management function and security assurance of ZSM management function,
- Security isolation and security requirement fulfilment in multi-tenancy environment of ZSM framework,
- Access control for management service provided by multiple domain service producers of ZSM framework,
- Leverage existing security specifications to identify security risk of AI/ML model and protect AI/ML models in ZSM framework.

To this end, we propose the use of blockchain technology in ZSM scenario to address the security issues and challenges of the system and at the same time maintain the complexity in reasonably low levels. The rest of this chapter presents the architecture of our novel idea and discusses how this solution could be implemented to safeguard the ZSM framework and contribute to the development of modern secure networks beyond 5G.

We propose the adoption of blockchain technology and its combination with ML technology towards increasing the automation level and the security of the ZSM framework while maintaining the complexity level low. Focusing on the scenario of E2E service deployment in a multi-domain environment and having in mind the architecture described in chapter 2, we propose the introduction of blockchain technology in the cross-domain integration fabric component as it is depicted in Figure 23.

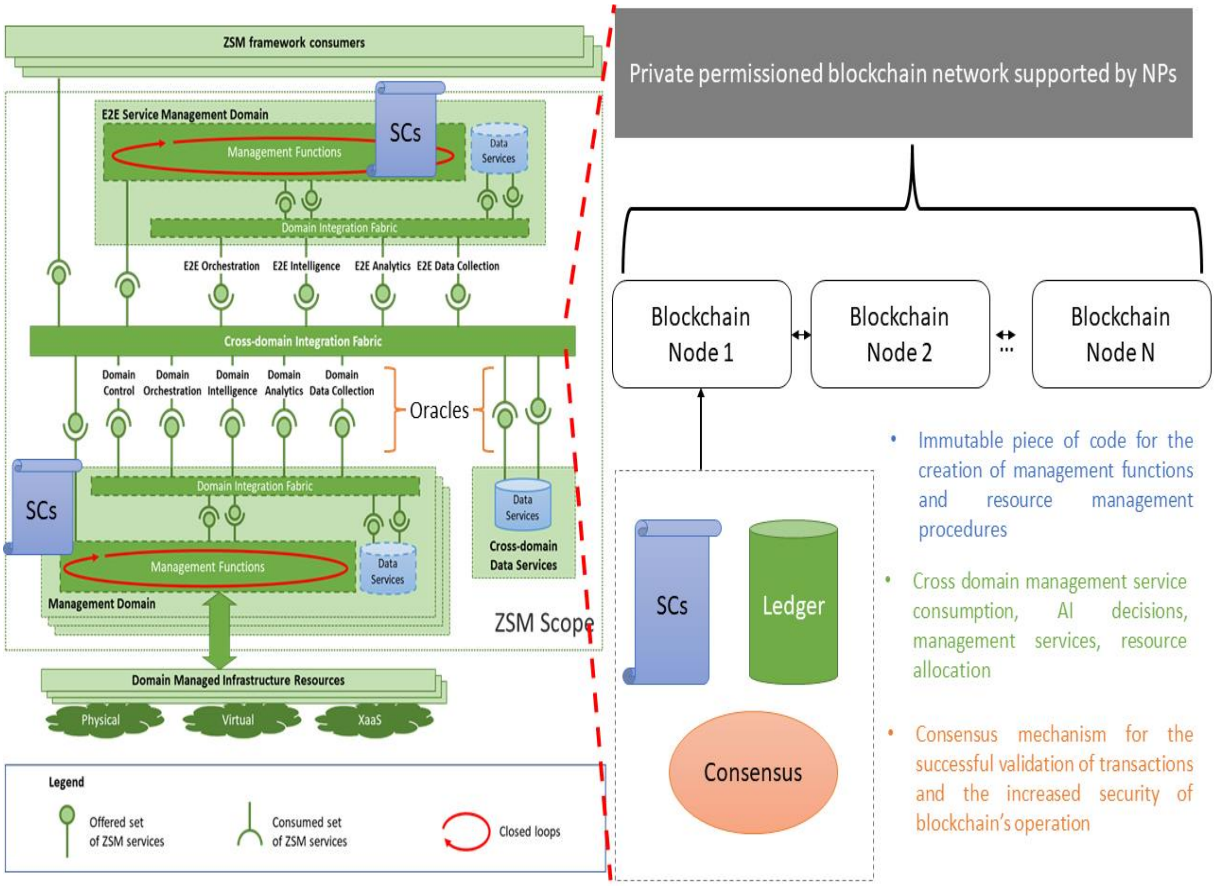
ZSM framework consumers
E2E Service Management Domain
SCs
Management Functions
Data Services
Domain Integration Fabric
E2E Orchestration
E2E Intelligence
E2E Analytics
E2E Data Collection
Cross-domain Integration Fabric
Domain Control
Domain Orchestration
Domain Intelligence
Domain Analytics
Domain Data Collection
Oracles
SCs
Domain Integration Fabric
Management Functions
Data Services
Management Domain
Data Services
Cross-domain Data Services
ZSM Scope
Domain Managed Infrastructure Resources
Physical
Virtual
XaaS
Legend
Offered set of ZSM services
Consumed set of ZSM services
Closed loops
Private permissioned blockchain network supported by NPs
Blockchain Node 1
Blockchain Node 2
Blockchain Node N
SCs
Ledger
Consensus
Immutable piece of code for the creation of management functions and resource management procedures
Cross domain management service consumption, AI decisions, management services, resource allocation
Consensus mechanism for the successful validation of transactions and the increased security of blockchain's operation

Figure 23.The architecture of the blockchain-based ZSM

In the presented approach, *each NP is part of the ZSM framework and hosts a blockchain node that belongs to a private permissioned blockchain network* as it is depicted in the figure above. The private and permissioned characteristic of the network increases the security of this approach as we are able to control which NP participate in the network and at the same time minimize the possibility of a malicious participant. The NPs are registered in the blockchain and obtain a unique address used as their identification in the network. In addition, the ledger of the blockchain includes not only the IDs of the NPs but also the addresses of the SCs deployed in the network. *Both the E2E service management domain and the management domain of the ZSM participant* *create and execute management functions which are deployed in the form of SC in the blockchain network.* According to ZSM standard, the development and execution of management functions is implemented using closed loops. Closed loops are based on AI/ML technology which use mathematical models trained by secure dataset of the framework. Within each MD, the architecture defined in [93] can be deployed so that the needs for additional resources is automatically detected and triggers the request for additional resources which is then handled by the blockchain-enabled solution outlined above.

It is worth mentioning that no AI/ML code runs inside the blockchain. Every change in the ZSM network in a cross-domain scenario, which in our case can be the consumption of a management function (i.e., a management function could be the deployment on a NP's premises of a network CDN service to support a streaming application), is considered a transaction and is stored in the ledger. The registration of an NP, the creation of a SC that utilizes a management function and the outcome of a SC are considered blockchain transactions and are permanently written in the ledger of the blockchain. Moreover, the blockchain interacts with other ZSM components using oracle mechanisms to ensure that valid information is exchanged from and towards the blockchain. Oracles in our case are software mechanisms developed to provide a secure interface between the blockchain network (including the SCs deployed in it) and ZSM services. To accomplish that, oracles use cryptography or/and consensus techniques applied on-chain or off-chain, to establish a secure connection between blockchain and other services outside of it. In the current research, oracles are used by the cross-domain integration fabric component as it is presented in Figure 23.

The way our solution addresses the security issues identified by ZSM is briefly presented in Table 3 and elaborated in the sequel. As new NPs join the ZSM framework, the number of blockchain nodes increases and the network grows, assuming that each NP hosts/deploys at least one blockchain node. The private and permissioned characteristics of the network minimize the possibility of the existence of a malicious player which is also tackled by the applied consensus mechanism. Since the blockchain network is private and permissioned, we assume that a trusted governance entity is responsible for registering the NPs to the network (i.e. the addNetworkProvider function presented in the previous chapter). Automatically, a trust layer among competitive NPs is created and the trust issue among multiple management domains highlighted by the ETSI team is addressed.

In addition, to reduce the vulnerabilities of management functions we take advantage of the immutability feature of Smart Contracts (SCs). *We propose the use of SCs for the implementation of the management functions defined in [91].* The rationale behind this is the following: a SC is an immutable deterministic piece of code stored and used in the blockchain network. SC's functionality cannot be undermined, and its content cannot be tampered as it is stored in the form of a transaction in the network. When a SC is created, it is related to a unique blockchain address used by other entities in the network in order to execute its functions. Additionally, a SC is a set of promises that is executed when predefined conditions are met. This feature allows SCs to execute functions automatically *without human intervention*. Given the security concerns regarding the vulnerability of management functions in ZSM, the use of SCs for their implementation is ideal. Moreover, the ability to control a SC's visibility to other blockchain participants is supported in various blockchain solutions and can be used to increase the confidentiality of a transaction or the non-disclosure of SC's information in multitenant environments, if this is required. As a result, we can achieve access control to sensitive information, such as management functions, stored in the network.

Having discussed the way blockchain addresses the ZSM's security issues, we examine how the multi-domain scenario described above changes with the integration of blockchain technology. Assuming that NP1 (which adopts/deploys the ZSM framework) has a request to support a demanding streaming application based on predefined network services and that NP1 cannot support the application using its own resources. Using the management services, NP1 finds another management service in a different domain that can fulfill the request. NP1 decides to consume the management service of the other provider (i.e., NP2) by executing a management function in the form of SC. The consumption of NP2's service by NP1 is registered as a transaction in the blockchain and the details of this transaction are defined by the SC which is also stored in the ledger. Figure 24 illustrates in an abstract manner the lifecycle of the discussed example in order to better understand the integration of ZSM and blockchain technology.

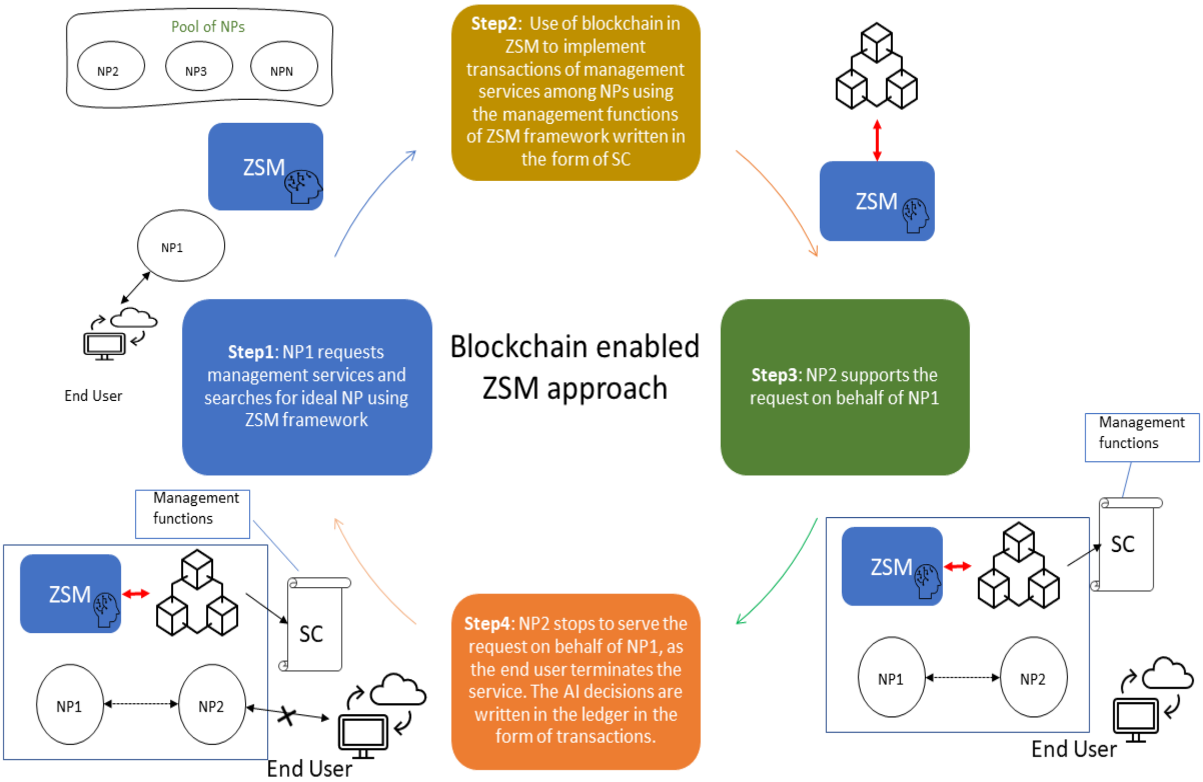
Pool of NPs
NP2
NP3
NPN
ZSM
NP1
End User
Step2: Use of blockchain in ZSM to implement transactions of management services among NPs using the management functions of ZSM framework written in the form of SC
ZSM
Step1: NP1 requests management services and searches for ideal NP using ZSM framework
Blockchain enabled ZSM approach
Step3: NP2 supports the request on behalf of NP1
Management functions
Management functions
ZSM
SC
NP1
NP2
End User
Step4: NP2 stops to serve the request on behalf of NP1, as the end user terminates the service. The AI decisions are written in the ledger in the form of transactions.
ZSM
SC
NP1
NP2
End User

Figure 24. Lifecycle of a blockchain enabled ZSM scenario

With respect to the machine learning techniques used to foresee the resources required to service a request, it is worth mentioning that:

a) this can be applied autonomously in each MD and when the need for additional resources is detected, this MD attempts to purchase the additional resources (exploiting the ZSM framework messages and blockchain technology). In this case, the results of the ML affect only this specific MD.

b) This can be applied over the whole set of MDs participating in the framework. In this case, the results of the ML can (in principle) be falsified in favor of a specific MD, e.g., suggesting to all the rest they need to purchase resources. To ensure that such a possibility will be minimized, blockchain can be used to ensure attributability of the resource needs' decision. Blockchain offers the possibility to store in an immutable manner a) the details of the training of the algorithm as proposed in [119], including the nodes that offered the updates of the models; b) links to the datasets (most probably stored in a (inherently distributed) Inter Planetary File System (IPFS) system and c) the node that decides to issue a resource request (to purchase resources) based on an ML-model. The traceability feature of blockchain allows us to examine the decisions of AI/ML components during their operation and identify any suspicious activity. At the same time, the credibility of the decisions' history cannot be questioned since it is a valid blockchain transaction registered in the ledger. As a result, the origin and quality of data is guaranteed, and an extra layer of security is added to the AI/ML components of ZSM.

Finally, the presented work published in [120] illustrates and assesses the idea of adopting blockchain to tackle the security concerns and, at the same time, guarantee the automated resource trading between NPs (as required to achieve the desired end user QoE) ensuring highly secure operation in cross domain scenarios.

4.2 Discussion regarding the use of blockchain in ZSM

ZSM networks are expected to lead the way towards the development of self-managed and self-optimized networks to form NGNs. The increased automation in combination with the minimum human intervention increases the performance of the network and at the same time exposes several security concerns. The investigation of several solutions to answer to these open issues with an efficient and less-complex manner, is a very tricky task. To this end, the current research examined the use of blockchain for the secure implementation of E2E cross domain services in ZSM scenarios and proposed an architecture for the integration of ZSM with blockchain. The idea is to take advantage of the main blockchain attributes to leverage the security of ZSM, having in mind the main drawbacks of blockchain technology. As a result, we tried to answer to the main security issues identified in the framework using the blockchain technology and at the same time keep the complexity in low levels. Table 3 summarizes the current work by illustrating how blockchain addresses the security issues highlighted by the ZSM standardization team.

Table 3. Solutions to security issues: Current ZSM vs Blockchain-based ZSM

Security issues and complexity	Trust relationship between multiple management domains	Security risks due to the vulnerability of management functions	Access control for management services in a multi domain scenario	Security risk of AI/ML model and protection of AI/ML models in ZSM	Complexity level
Current ZSM approach - Solutions	Reflective and adaptive trust model	GSMA Network Equipment Security Assurance Scheme (NESAS)	Authentication and authorization mechanisms, which check the trust relationship among entities	Risk assessment based on the Adversarial ML Threat Matrix	High – Use a bunch of technologies to increase ZSM security
Blockchain-based ZSM approach - Solutions	Achieve trust in trustless environment using blockchain	Use of SCs to eliminate the vulnerabilities which are automatically and securely executed	Use SCs for management services and apply visibility rules to achieve access control	Store the AI/ML decisions in the ledger and guarantee dataset integrity using both blockchain and IPFS	Low – Use blockchain technology and take advantage of its inherent characteristics

Considering the blockchain-enabled ZSM framework's architecture and functionality, we may claim that the proposed solution leverages the security of the system while it is less complex than the solutions defined in the ETSI whitepapers. Our approach aims to eliminate these issues by using only the blockchain technology. In the current research the complexity level of the system is defined based on the number of parts definition mentioned in [121]. According to the definition, a system that uses a smaller number of parts is less complex than a system that uses higher number of parts. In [92] the solutions proposed by the standardization team use many different technologies and techniques (e.g., NESAS, Adversarial ML Threat Matrix) to tackle the security issues. The complexity level in the proposed (our) solution is lower because the ZSM is combined with one key technology, the blockchain supported by a network that grows as new NPs join the framework while the main principles of the system remain the same. The use of oracle mechanisms and SCs boost the functionality and adaptability of blockchain which can interact with ZSM components in a secure manner.

Although blockchain introduces valuable characteristics such as the ability to form a network of trust in a trustless environment without the existence of a trusted third party, the immutability of data recorded as validated transactions in the ledger and the traceability feature, no technology is weakness-free. There are some drawbacks in this technology, which should be carefully considered during the design of the blockchain-based solution. Some of the most crucial ones are the following:

- *Scalability* is one of the main drawbacks of this technology since transactions written in the ledger cannot be deleted afterwards. However, as blockchain solutions nowadays attempt to compare in speed and storage volume with banking transactions, it is worth noticing that the environment considered in this work (i.e. the environment of network provider) is issuing “transactions” at a significantly lower pace that worldwide finance transactions occur.
- *Time needed for the transaction validation*. The proper operation of blockchain networks and more specifically, the transaction validation time is affected by the consensus mechanism used in the blockchain software. The consensus process in the network requires time that depends on the consensus type (Byzantine Fault Tolerant, Crash Fault Tolerant etc.), the size of the blockchain network, and the consensus protocol specifications. Time is crucial in modern networks, and it is considered as one of the key requirements of ZSM framework. In our previous work [70,79], we examined the performance of blockchain networks in terms of throughput and latency and we observed delays in the unit of seconds. A solution could be the use of a blockchain technology with fast consensus convergence time which could leverage the functionality of the solution. It should be noted that the selection of the suitable blockchain solution is a very important task for the development of time critical blockchain applications.

In addition, the introduction of blockchain technology may add extra functionalities to modern networks. The ability to use tokens in blockchain solutions introduces extra utilities that can be added in almost every use case. As both ZSM and blockchain are being developed we may claim that there is a very interesting research topic where these two technologies can be combined and lead to the evolution of next generation networks beyond 5G.

4.3 Guidelines for DLT solution design and deployment in ZSM scenario

The selection of the most suitable DLT solution for the development of a blockchain-based ZSM scenario should be based on specific requirements which should be clearly defined. The following section of this chapter aims to present the characteristics that a DLT solution should present in order to satisfy the needs of the ZSM use case. Furthermore, we proceed to the identification of the most suitable blockchain and DAG solutions, and we analyze their main functionalities and characteristics. Then we evaluate their suitability for the particular use case and we propose modifications to fulfil the requirements of our scenario.

4.3.1 Requirements of the blockchain-based ZSM approach and definition of the characteristics of the ideal DLT solution

Considering that modern networks promise to deliver high quality services in every corner of the world, on demand and almost instantly, the performance of the overall system that implements a blockchain-based ZSM scenario is of significant importance. The system's performance depends on the performance of each entity. In our scenario we combine two major technologies the blockchain supported by the blockchain network and the ZSM. The performance of the blockchain network directly affects the performance of the overall system which means that characteristics of the network such as latency and throughput affect the end user's QoE. Furthermore, other characteristics such as the accessibility of the network, the resiliency, the scalability and the network's ability to easily accept new features, are vital for the systems' successful operation and future growth. To this end we define certain characteristics that a DLT solution should present in order to build a secure, scalable and high-performance environment ready to be integrated with the ZSM framework.

The DLT solution integrated with the ZSM framework must be:

- ✓ *Access controlled:* the access and the ability to perform actions in the network should be allowed only to permitted members and should be restricted to anyone else. Therefore, only a predefined group of entities, in our case the NPs, should be able to access the network.
- ✓ *Scalable:* the nodes of the network are hosted by the NPs only. Every provider wishes to join the network should be able to easily deploy a node and become active member of the solution.
- ✓ *Resilient:* the proper functionality of the network should not be affected if a node or a number of nodes become unavailable for a period of time. The network should be resilient to network failures in order to present high availability which is one of the main requirements of NGNs.
- ✓ *Very fast:* the network should be able to perform multiple actions in a short period of time (in the unit of seconds or even milliseconds). Since modern networks should provide high quality services instantly, the network used in our scenario should not affect the experience of the end user.
- ✓ *Programmable:* the DLT adopted in this particular use case should be able to support the development of code to implement the necessary functions. The code is the logic of the system which utilizes all the functionalities needed for the successful operation of the entire solution.
- ✓ *Extensible:* the DLT chosen for the implementation of this scenario should be able to accept new features that will upgrade its functionality and cover needs that may occur in the future. This characteristic is relevant to the previous one as it depends on the programmability of the network.
- ✓ *Interoperable:* the DLT should be able to support interaction with services outside the network in order to successfully communicate with ZSM services. This characteristic increases the ability of the network to interact with the outside world and use services that may leverage the functionality of the whole solution.

The above characteristics are translated in blockchain terms as follows:

- ✓ *Permissioned network:* only the permitted entities can access the network and perform actions in the form of transaction. This feature increases the security of the system since every entity is known to the others and therefore it is less possible to act maliciously. The implementation of the permissioned network requires the presence of a governance entity that allows/authorizes new participants to join. It should be noted that the governance entity does not control the operation of the network and its role is restricted only to the authorization of the NPs.

- ✓ *Private/Consortium network:* the network is supported by the NPs only, which means that the nodes of the network are created, managed and maintained by the participants. Although this feature seems to question the sentiment of decentralization in the network, it increases the security of the system since only the NPs are responsible for the proper operation of the network. In this particular use case, a NP who wishes to become active member of the network should be able to easily deploy a node which will automatically become part of the network. This results to the growth of the network as new NPs with new nodes can easily become part of the solution. As a result, the scalability feature of the entire solution is highlighted, which is a factor that attracts new members.
- ✓ *Crash fault tolerance:* the network should support fault tolerant mechanisms in order to ensure that its operation is not affected if a number of nodes becomes unavailable. Considering that the ZSM framework uses the DLT network to perform crucial tasks, the resiliency of the network is vital in our scenario. The main element that is responsible for the network's proper operation is the consensus, as it was discussed in previous chapters. Therefore, the network should be able to use consensus mechanisms that will increase its fault tolerance and guarantee the functionality of the system in hazardous situations.
- ✓ *Low consensus convergence time:* the proposed solution is expected to receive large number of transactions which must be validated/executed with the minimum possible delay. As a result, the selection of a consensus algorithm with low convergence time that is able to cope with high number of transactions is crucial. This characteristic is related to the previous one, as the consensus mechanism used is able to increase the resiliency of the network and the speed of transaction validation. Therefore, the DLT solution should be able to support a consensus mechanism that can validate transactions fast, and at the same time tolerate failures.

- ✓ *Support of SCs:* the development of code in DLTs can be accomplished by the creation of SCs which can implement various functionalities of the network in a secure manner. Since the network should interact with ZSM services and provide solution to many and different problems regarding the management of modern networks, the capability of the DLT to support SC's is extremely important.
- ✓ *Support of tokens:* this feature is related to the previous one and highlights the need to support tokens implemented in the form of special SC functions, which improve the functionality of the solution. Tokens can be used to represent assets (i.e., a car can be defined as token), currency (i.e., the Ether in Ethereum network is practically a token) and rights (i.e., use a token to access a website). The development of tokens is based on standards that ensure their smooth integration in the network. There are two main well-known token categories, the Fungible Tokens (ERC 20) and Non-Fungible Tokens (ERC 721). In our use case a Fungible token could be used to create a currency used for the transactions among NPs in order to build a modern marketplace. Additionally, Non-Fungible Tokens (NFTs) could be developed to represent the reputation of a NP which could be related to the successful completion of a number of requests. This feature could be used by NPs as a criterion during the process of NP selection to support their request. It should be mentioned that these scenarios are only examples of how tokens could be used and have not yet been designed for our use case. However, the ability of a network to accept and use new programming features to enhance its functionality and solve any future issues is a very important characteristic.
- ✓ *Interaction with oracle mechanisms:* the DLT solution should be able to interact with entities outside the network in a secure manner to leverage the functionality of the whole system and support the ZSM processes. To this end, oracle mechanisms must be supported by the network, while the selection of the most suitable solution should be made based on the security level and the performance. On the one hand the information from and towards the network should be well protected while on the other hand the latency introduced by the oracle should be the minimum. Having in mind that some oracle solutions use consensus, which automatically introduces extra latency to the system, the selection of other oracles that use different tools seems to be preferred. There

are many oracle mechanisms available which use encryption to protect the content of their data and guarantee the origin of the information. The adoption of such a solution may not affect the overall system's performance dramatically.

Having defined the main characteristics that a DLT solution should present to become the ideal choice for our scenario, we proceed to the presentation of a specific DLT category that is different to the blockchain discussed in the previous chapters. In the following section we proceed to the presentation of Directed Acyclic Graphs (DAGs), and we compare them to the blockchain technology to identify DAGs' strong points and weaknesses. Then we proceed to the selection and examination of the most promising blockchain and DAG solutions for our use case and we highlight the points that need to be improved/modified in order to become the ideal networks for our solution.

4.3.2 Distributed Ledger Technologies: Blockchain vs DAG

The DLTs contain two main subcategories, the blockchain and the directed acyclic graphs. Although blockchain is a very popular specialization of DLTs and often mistaken as a synonym of DLT, DAGs present interesting characteristics. In contrast to blockchain technology, where transactions are bundled in cryptographically linked blocks creating a single chain, DAGs use a non-linear graph that consists of nodes which represent a transaction. Figure 25 illustrates an overview of both DLT specializations. The nodes depicted in the right side of the figure which belong to the DAG section, do not store the entire ledger but they are obligated to verify at least two previously inserted transactions. This results to the formation of a directed graph where on the one side are the older nodes (orange nodes) while on the other are the new ones (blue nodes). The acyclic term in this DLT category means that no nodes can reference back to an older one and are considered as mother nodes to the new nodes that are becoming members. In addition, when a new node is added, the path towards the old nodes increases. The longest the path is towards an old node, the more credible the node is regarding the transaction validation.

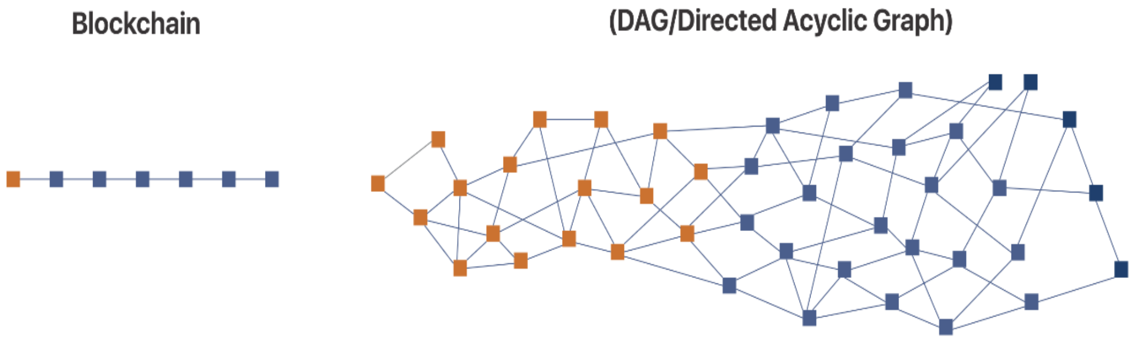

Figure 25. Blockchain vs DAG [122]

The validation of transactions in both DLTs is based on the consensus process. However, *the consensus mechanism used in DAGs are significantly faster than the one used in blockchain networks*. Since DAGs do not use blocks, they can proceed immediately to the verification of a transaction in contrast to typical blockchain networks that collect a number of transactions to form a block and then proceed to its verification and finally validation. In the following subsections of this chapter the functionality of two DAG consensus algorithms, the Tangle and the Hashgraph, are presented.

Furthermore, DAGs are considered to be more scalable than blockchain since the nodes do not have to store the whole ledger. A node is able to store only certain parts of the graph and not the entire path. *This feature automatically increases the scalability and allows devices with limited computational power to become active parts of the DAG network.* Therefore, a wide application area becomes available for the development of novel DAG based solutions considering the growth of IoT environments which are based on the use of Single Board Computers (SBCs) and sensors.

Beyond the differences of these two DLTs, there are also common characteristics such as the support of two different deployment types: the permissioned and permissionless networks. Both blockchain and DAG can be implemented as publicly available networks or as networks available only to the authorized/permitted users. The selection among these two types is based on the nature of the solution to be developed. Another common characteristic is the support of SCs which opens the way to the development of DApps. This feature automatically broadens the application area of these two DLTs and leads to the development of sophisticated and secure applications. Additionally, in terms of popularity, the blockchain is way more popular than the DAG technology, as it has been introduced to the public in 2009 with the launch of its well-known application, the Bitcoin. Table 4 tabulates the similarities and differences of these two DLTs.

Table 4. Blockchain vs Directed Acyclic Graphs (DAGs)

DLTs characteristics	Blockchain	Directed Acyclic Graphs
Use of consensus	Yes	Yes
Data structure	Validation of blocks	Validation of transactions
Transaction throughput	Lower	Higher
Scalability	Lower	Higher
Permissioned/Permissionless	Yes	Yes
Support of SCs	Yes	Yes
Popularity	Higher	Lower

4.3.3 Candidate blockchain solutions for the blockchain-based ZSM scenario

In this subsection of the thesis, we proceed to the presentation of three promising blockchain networks and we examine their suitability for the blockchain-based ZSM use case having in mind the requirements defined earlier. The blockchains we examine is the Hyperledger fabric, the Ethereum Quorum and the R3 Corda. The reason of their selection is that they present characteristics that are more likely to fulfil the requirements of our scenario.

Hyperledger Fabric (HLF)

This blockchain solution as it is described in [123], is an open-source permissioned platform that is established and maintained under the umbrella of the Linux Foundation. Hyperledger Fabric has been developed mainly for enterprise purposes and is governed by a diverse set of maintainers from multiple organizations. Currently the Hyperledger community has grown to over 35 organizations and almost 200 developers since its earliest commits.

Hyperledger Fabric's architecture is modular and configurable in order to easily become adopted to a wide spectrum of industry use cases. The versatility of this platform makes it ideal for several sectors such as healthcare, supply chain and others, while its ability to support SCs written in general-purpose programming languages (i.e., Java, Go and Node.js) makes it very attractive to organizations. Most enterprises already have the skill set needed to develop smart contracts, and no additional training to learn a new language nor the recruitment of experts is needed. In addition, this platform is permissioned therefore the participants are known to each other which automatically grows a sentiment of security. This means that while the participants may not fully trust one another (they may, for example, be competitors in the same industry), a network can be operated under a governance model. [123]

Another important characteristic of this platform is its ability to support pluggable consensus mechanisms. This feature allows HLF to be effectively customized in order to fit in various use cases. For instance, in the ZSM scenario where only known NPs are members of the network, a fully byzantine fault tolerant mechanism might be considered unnecessary and an excessive drag on performance and throughput. In order to maintain high performance standards and increase the availability of the solution, a crash fault-tolerant consensus might be the preferred option. Nevertheless, in the examined scenario if the parties involved were not only NPs, the use of a more traditional byzantine fault tolerant consensus might be required. The pluggable feature of HLF is achieved thanks to a modular component used to implement the consensus and transaction ordering, that is logically decoupled from the peers that perform transaction and maintain the ledger. This modular architecture allows the platform to rely on well-established toolkits for crash fault-tolerant or byzantine fault-tolerant ordering. Fabric currently offers a crash fault tolerant ordering service implementation based on the etcd library of the Raft protocol. Moreover, a HLF network can have multiple ordering services supporting different applications or application requirements. [123]

Fabric can leverage consensus protocols that do not require a native cryptocurrency to incent costly mining or to fuel smart contract execution. Avoidance of a cryptocurrency reduces some significant risk/attack vectors, and absence of cryptographic mining operations means that the platform can be deployed with roughly the same operational cost as any other distributed system. The aforementioned design features make HLF one of the better performing platforms both in terms of transaction processing and transaction confirmation latency, and it enables privacy and confidentiality of transactions and the smart contracts that implement them. It should be mentioned that many research papers have been published where the performance metrics of the HLF is studied and tested using the Hyperledger Caliper. Authors in [124] scaled HLF to 20,000 transactions per second [123]. Concluding the presentation of HLF, this solution supports the creation and management of tokens and is able to use several oracle mechanisms in order to become suitable for several use cases that demand the interaction of blockchain with the outside world.

Ethereum Quorum

Quorum is a permissioned implementation of Ethereum and it was initially developed by JP Morgan. The goal of this blockchain is to cover the needs of scenarios designed to operate in a controlled network where the identity of the members is known and the access to the public is restricted. Therefore, it is considered as an ideal solution for the implementation of private and consortium networks. An example of this characteristics is illustrated in chapter 3 of this thesis where an implementation of a Quorum network is presented and evaluated for the scenario of resource management among different network providers in NGNs.

In contrast to traditional Ethereum network, Quorum supports two different types of consensus mechanisms: the Raft and the IBFT. This feature allows developers to use a mechanism that suits better to their use case. For example, the Raft which belongs to the CFT consensus family is preferred in cases where the existence of a malicious participant is unlikely and the need for fault tolerance is high. Nevertheless, in cases where many different entities are participating in the Quorum network and the likelihood of a malicious member is high, the IBFT mechanism is preferred since it introduces byzantine fault tolerance. As it was illustrated in our experiments in chapter 3, a “one fits all” solution is not feasible, therefore the ability to use different consensus mechanisms is an extra advantage for a blockchain platform.

Similar to Ethereum, the network of Quorum supports the use of tokens and SCs that allow the creation of distributed secure applications. This feature broadens the application area of Quorum as many DApps can be developed to implement various scenarios. However, a major difference of these two blockchains is that Quorum supports privacy which was one of its main design goals. More specific, it allows subsets of parties in a consortium to transact with one another without making the transactions public to members of the larger consortium. Quorum practically splits the ledger into a public and a private ledger. All nodes of the network can observe the public ledger, while the private ledger is visible only to the transacting parties. Only a hash of the private transaction appears on the public ledger and is visible to other nodes that are not counterparties to the transactions. It should be noted that only the counterparties to the private transaction have the keys to decipher and observe it. This process can be done also for the deployment of private smart contracts which would be visible only to the transacting parties. [125]

Moreover, another significant difference of Quorum and Ethereum is the fact that Quorum does not adopt the concept of adding cost to a transaction using gas. Although it is a fork of Ethereum and supports the use of gas, it sets this value to zero to run transactions without gas fees. Since Quorum is usually deployed in a consortium or private blockchain, the use of gas in Ethereum terms is not mandatory [125]. Also, Quorum platform can be combined with oracle mechanism in order to become part of a solution that is not limited only to the blockchain world.

R3 Corda

Another popular blockchain platform is the R3 Corda, which allows the implementation of private permissioned networks ready to support various use cases. Corda consists of the following entities: the nodes, the identity service, the network map service, the notary service and the oracle service. A member of the network in order to deploy and manage a node, has to first acquire an identity certificate from the identity service. The existence of an identity service reduces the likelihood of malicious activity triggered by one or many nodes and therefore it increases the sentiment of security. In addition, the network map service contains and publishes information about each Corda node such as the supported version protocol, the active IP addresses and the identity certificates it hosts. Each data structure describing a node is signed by the identity keys it claims to host. The network map service is therefore not trusted to specify node data correctly, only to distribute it. [126]

Furthermore, the notary service in the Corda network is responsible to perform transaction ordering and timestamping. It actively participates in the consensus process which can be implemented using either crash, or byzantine fault tolerant algorithm. The selection of the desired algorithm depends on the use case scenario as it is stated earlier. Similar to the previous platforms, Corda supports both BFT and Raft mechanism. Considering our use case presented previously, a notary that uses Raft between nodes that form a network of NPs will present extremely good performance in terms of throughput and latency, at the cost of being more vulnerable to malicious attack by whichever node has been elected as a leader. [126]

Moreover, the Corda platform supports SCs for the development of several solutions called Cor-Dapps. SCs are defined using a restricted form of Java Virtual Machine (JVM) bytecode, which automatically allows developers to implement the logic of their solution by writing code in a variety of programming languages. Developers are able to use well developed toolchains and to reuse code written in Java or other JVM compatible languages, which is a fact that widens the application area of this platform. Also, Corda supports the development and use of tokens which can be used according to our use case to represent resources, like CPU, memory and others. [126]

Additionally, the privacy feature is supported in this blockchain as Corda uses several techniques to achieve this functionality. This means that the implementation of private transactions or the execution of private SCs is feasible. At the same time, Corda supports the use of oracles, defined as a network service that is trusted to sign transactions containing statements about the world outside the ledger only if the statements are true. This characteristic allows the secure communication of blockchain with entities outside the network such as the ZSM framework and its services. The use of oracles increases the adaptability of this blockchain platform and thus its attractiveness from a developer's perspective. Also, Corda presents high performance values as it can reach up to 20,000 transactions per second according to their benchmarking results illustrated in platform's website. [126, 127]

4.3.4 Candidate DAG solutions for the DLT-based ZSM scenario

Beyond the blockchain-based platforms there are other promising solutions based on Directed Acyclic Graphs (DAGs), which have been discussed in previous subsection of this chapter. In this subsection, we focus on two key DAG applications, the IOTA and the Hedera Hashgraph and we examine their functionality and main characteristics having in mind our use case scenario.

IOTA

IOTA is a very popular DLT solution, which is based on the Tangle DAG. It is supported by the IOTA Foundation which aims at the development of new DLT-based solutions. On July 2016 the IOTA main net was activated and it is considered as a public permissionless network. Some of IOTA's main characteristics are the increased scalability, the increased sentiment of decentralization and the zero transaction fees. In contrast to typical blockchain networks which present scalability issues as the transaction number increases, IOTA becomes more efficient and more powerful when the transaction number grows. Since IOTA does not use miners in the network, a node is at the same time, the creator and the validator of a transaction. This means that everyone in the network contributes to the consensus process, which is a fact that highlights the decentralized nature of this particular solution.

Moreover, IOTA uses the DAG technology at its core and therefore inherits the operation of DAGs presented in a previous section. The consensus mechanism implemented in the latest IOTA version is a probabilistic leaderless binary voting protocol called fast probabilistic consensus (FPC). This mechanism is responsible for Tangle's validity by addressing issues such as the double spending problem. FPC allows a node to update its opinion by querying a set size subset of other nodes on the network and then choosing the majority opinion. This is done multiple times (rounds) until an unchanging opinion is decided or a maximum round threshold is reached. It should be mentioned that the FPC mechanism presents high performance metrics and is considered as a scalable consensus, ideal for large numbers of transactions. [128]

In addition, the IOTA supports the creation of smart contracts called ISC and hence the development of several applications for various use case scenarios. ISC is agnostic regarding the virtual machine which executes the SC code. IOTA currently supports two types of SCs: the Rust/Wasm-based and Solidity/EVM-based. Nevertheless, all kinds of virtual machines can be supported in an ISC chain depending on the use case. It should be noted that IOTA smart contracts are more complex than typical SCs, but they provide freedom and flexibility to the developer to create and use SCs in a broad range of use cases. [129]

Furthermore, IOTA allows the use of tokens which can be exchanged among entities in this DAG-based network. A well-known token used as a cryptocurrency in this solution is the MIOTA which can be purchased by a user in order to buy assets in the network. MIOTA is an example of a fungible token, however the use of NFTs is also supported. Moreover, the use of oracles is supported in this DLT. Oracles bring off-chain data to decentralized applications and smart contracts on the IOTA network. These mechanisms provide blockchains with outside information, typically for use in smart contracts, or provide interoperability between different distributed ledgers. The IOTA Tangle oracle presents several key advantages compared to a conventional blockchain oracle solution which are listed as follows [130]:

- Implementation of feeless transactions,
- Transactions can hold a large amount of data,
- IOTA’s network operates in near real-time,
- Data retrieval using an IOTA node is lightweight and efficient,
- IOTA Oracles support diverse security and data structuring capabilities.

Hedera Hashgraph

Another DAG-based solution is presented by Hedera, named Hedera Hashgraph [131], which uses a distributed network, cryptographic tools and timestamps to store data in the form of transactions. This platform is considered a public permissioned network, although it is currently governed by the Hedera Council that deploys and supports the network nodes. In the future anyone will be able to host and operate a Hedera Hashgraph node.

The consensus mechanism used in this DLT is called Hashgraph and is based on the gossip protocol. Every node that transacts with another one, sends information regarding the current state of the network which is based on information previously received by other nodes. As a result, the information regarding the current state is spreading like a gossip among the nodes of the network. Therefore, every node in the network contributes to the consensus process and every node is aware of the current state. Hashgraph achieves high-throughput with 10,000+ transactions per second today and low-latency finality in seconds from its innovative gossip about gossip protocol and virtual voting. Once consensus is reached, the transaction is immutable and available on the public ledger for everyone to transparently see. The nodes in Hashgraph are divided in two types: the consensus nodes and the mirror nodes. The consensus nodes are actively participating in the consensus process while the mirror ones offer developers a flexible and cost-effective way to store and query historical data for analytics and explorers. It should be mentioned that the nodes store only the latest state of the network in their ledger, which automatically increases the scalability of the network. [131]

Hedera Hashgraph offers a set of so-called Hedera Services that allow users to perform various tasks such as the creation of SCs and tokens. The Smart Contract service of

Hedera allows the creation of contracts using the Solidity language similar to Ethereum-based networks. Hedera supports Solidity in order to take advantage of community-driven standards, development tools, frameworks and support, considering that Solidity is one of the most popular SC languages in the world with a huge community. At the same time Hedera promises fast SC execution with lower cost than blockchain alternatives.

Moreover, tokens are supported by the Hedera Token service which allow the configuration and management of native fungible and non-fungible tokens. Therefore, this service can support financial applications with the creation and management of tokens (fungible) which can be used for secure and real-time payments. Having in mind our use case, a NP could receive a payment in the form of token for lending its resources to another NP. In addition, NFTs could be used as a reputation badge to highlight the reliability of a NP in our scenario.

The support of SCs and tokens widens the application area of Hedera Hashgraph which can be used in various use cases. Nevertheless, an additional factor that adds extra flexibility to this platform is its ability to cooperate with oracle mechanisms. Chainlink and Hedera Hashgraph announced in 2019 their collaboration to integrate Chainlink's decentralized oracle solution with Hedera's network. Chainlink is a well-known oracle mechanism which allows smart contracts to securely access and retrieve off-chain information when needed. It uses a similar model to a blockchain, as it implements a decentralized network of independent entities, called oracles, that collectively retrieve data from multiple sources, aggregate it, and deliver a validated, single data point to the smart contract to trigger its execution, removing any centralized point of failure. [132]

4.3.5 Modifications of the discussed blockchain and DAG solutions to fit to our scenario

Having examined the blockchain and DAG DLTs in the previous subsection, we proceed to the identification of their main characteristics in Table 5, which directly affect their suitability for our use case scenario. The columns of the table present the main attributes that describe the DLT's functionality while at the bottom of the table the row with the ideal solution defines the properties of the most suitable solution for our scenario.

Table 5. Suitability of the examined blockchain and DAG solutions

Solution	Public – Private/Consortium	Permissioned - Permissionless	Consensus type	Support of SCs	Support of tokens	Support of oracles
HLF	Private / Consortium	Permissioned	CFT (Raft) or BFT (pBFT)	Yes	Yes	Yes
Quorum	Private / Consortium	Permissioned	CFT (Raft) or BFT (IBFT)	Yes (private SCs)	Yes	Yes
R3 Corda	Private / Consortium	Permissioned	CFT (Raft) or BFT (pBFT)	Yes (private SCs)	Yes	Yes
IOTA	Public	Permissionless	FPC	Yes	Yes	Yes
Hedera Hashgraph	Public	Permissioned	Hashgraph	Yes	Yes	Yes
Ideal solution	Private/Consortium	Permissioned	CFT – rapid convergence	Yes	Yes	Yes

Comparing each of the discussed solutions with the ideal one, we observe that every solution support SCs, tokens and oracle mechanisms. However, in the Quorum and R3 Corda we are able to use private SCs and implement private transactions if necessary. This is an extra feature that can be used to add extra functionalities to our solution in the future. For instance, some NPs in the network may sign an agreement of cooperation and fulfil requests with special terms which they might not want to unveil to other NPs in the network.

Furthermore, the blockchain solutions implement private/consortium and permissioned blockchains which are the ideal characteristics of the network to be adopted in our scenario. On the contrary, IOTA and Hedera Hashgraph support only public implementations while the IOTA network allows access to anyone as it is a permissionless network. These characteristics decrease the suitability level of those solutions for our use case and should be modified.

In addition, the ideal solution should use a crash fault tolerant consensus mechanism with high convergence time in order to ensure the high availability of the system and achieve high transaction validation numbers, considering that the likelihood of a malicious participant is low. As it is illustrated in Table 5 the blockchain-based solutions can use consensus mechanisms with these features by implementing the Raft algorithm. However, DAG-based solutions use the FPC and Hashgraph mechanisms which present higher transaction validation speed which seems ideal for our use case. In terms of performance, the DAG solutions present higher numbers (transaction throughput) than the blockchain ones and therefore they are considered more suitable for our use case where the transaction number are expected to be extremely high.

The scalability of the DLT solution is also a significant factor which plays a crucial role for the selection of the ideal solution. As it was mentioned in previous subsections, the blockchain solutions present scalability issues as every node holds a full copy of the ledger which increases as new transactions are validated. Nevertheless, DAG nodes store only parts of the graphs and therefore they scale well when the transaction number increases. Hence, they are considered more scalable than the blockchain solutions.

Concluding the qualitative assessment of blockchain and DAG solutions, it is clear that none of the examined networks fulfil the criteria to become the ideal solution for our use case. The development of a private/consortium and permissionless DAG network could be the most suitable solution, having in mind the main characteristics of the presented DAG-based solutions.

5. Conclusions and future work

In this last chapter of the thesis, the main conclusions of our study are presented accompanied by the description of future research paths which may lead to the generation of novel works in the field of security designs for NGNs. To this end, we summarize and highlight the most valuable information included in this study, which focuses on the use of blockchain technology in modern networks to ensure the secure resource management in NGNs.

5.1 Conclusions

It is evident that a new era in the computer networks sector has begun with the advent of 5G and Software Defined Networks. To offer high QoE, modern networks promise to offer ultra-high network speeds, high availability, world-wide coverage and other high-performance characteristics. In parallel, Network providers seek to maximise the efficiency of resource usage exploiting SDN technologies and to share resources/infrastructures in a highly dynamic and secure way. Thus, a marketplace of infrastructural resources is created. This marketplace introduces a new paradigm which led to the research questions: “how secure could be this marketplace that consists of various competitive NPs since no one trusts each other?” and “How could we form a network of trust among participants in this vast competitive market without the existence of a trusted third party?” Considering also the advancements in modern networks, and more specific the development of ZSM framework, an additional question to be answered was: “how this technology/solution can be adapted in NGNs to ensure security and trust in a self-managed dynamic environment such as the one created by applying the ZSM?”

The answer to these questions is the use of a decentralized and distributed technology that establishes a network of trust among participants. We chose to study a decentralized solution in order to avoid all the drawbacks of centralized approaches, such as the SPoF problem. Therefore, we focused on a distributed ledger technology and more specifically the blockchain which is one of the most hyped technologies in the last few years. Blockchain’s inherent characteristics such as the formation of a network of trust in a trustless environment, the data integrity and immutability, as well as its ability to support SCs, which are immutable pieces of code used for the development of DApps, motivated us to examine further this promising technology. Our research initially focused on the functionality of blockchain and its applicability in various sectors of our lives, while at the same time we identified the limitations and drawbacks of this technology. The initial goal was to fully understand the pros and cons of blockchain in order to decide on its selection for the target use case.

During our research we studied in depth the functionality and main characteristics of blockchain such as: the deployment of the network (public or private/consortium), the accessibility of the blockchain (permissionless or permissioned), the consensus mechanism (CFT, BFT and others), the support of SCs and tokens, as well as the support of oracle mechanisms. The results of our study showed the potential of this technology and the benefits it provides. For instance, the immutability of data contained in the ledger which can be easily accessed due to their transparency is extremely valuable for the supply chain management sector in order to guarantee proof of origin. This feature adds extra value to the product and increases the competitiveness of the company that adopts a blockchain-based solution.

However, blockchain technology presents some major limitations which were presented and studied during our research. The time needed for a transaction to be validated is a factor that plays a significant role in time critical applications. The consensus mechanism directly affects this parameter, as the transactions are bundled into blocks which are verified and validated based on the consensus. Another drawback of this technology is the scalability. Having in mind that data written in the ledger cannot be deleted afterwards, the size of the ledger grows as new transactions are inserted. In cases where the transaction rate is high the size of the ledger increases in a very rapid pace. Therefore, the nodes of the blockchain network should present significant storage capacity and the information contained in the transactions should be valuable. Information that is not critical for the network should be stored in alternative storage systems which may use various techniques such as the IPFS.

We also examined the advancements in the networking field and more specifically the development of ZSM framework and the applicability of blockchain technology in this framework. The ETSI has proceeded to the design and development of ZSM standard to enable the self-management and self-orchestration of modern networks. The goal is to achieve end-to-end management automation and eliminate the human intervention which could introduce human errors and delays. During our research we analyzed the

ZSM's functionality and we discussed the benefits this framework introduces that targets to the development of NGNs beyond 5G.

Although ZSM promises to revolutionize the way networks are managed, there are significant security issues highlighted by the standardization team which should be taken under consideration. The lack of trust among multiple management domains belonging to different NPs is one of them. Also, according to researchers the security risks introduced by the vulnerabilities of management functions and the security assurance of ZSM management functions are major threats for the framework's proper functionality. Additionally, the security isolation and security requirement fulfilment in the multi-tenancy environment of ZSM framework and the access control for management services provided by multiple domains are severe security issues that need to be addressed. Despite the fact that AI/ML leverage the functionality of the framework, these technologies present additional security risks which should be mitigated. The standardization team suggests countermeasures to these threats and urge the research community to examine alternatives to decrease and possibly eliminate these threats. Considering the architecture and functionality of ZSM illustrated in Figure 10, as well as the benefits provided by the blockchain, the idea of combining these two technologies was born. This resulted to the identification of a new research area presented in Figure 5, which can lead to the development of novel solutions that will contribute to the evolution of modern networks beyond 5G.

While theoretically blockchain and ZSM could be combined to implement a secure resource management scenario in NGNs, practical implementations should take place in order to evaluate the feasibility of such an approach. Therefore, we conducted experiments to test the performance of the blockchain network in a statically configured resource management scenario. In order to quantify assess the performance of the solution, we implemented it and performed measurements on a test bed we have set up in the cloud infrastructure of the University of West Attica, department of Industrial Design and Production Engineering. The results validated our decisions and helped us identify points of potential improvement.

Firstly, we proceeded to the selection of the proper blockchain solution for the implementation of our testbed. Since the blockchain network should support only NPs we deployed a private/consortium blockchain network where only permitted NPs would be members. The private permissioned network was created using the Ethereum

Quorum software that supports two types of consensus algorithms the Raft that belongs to the CFT family and the IBFT which belongs to the BFT mechanisms.

The blockchain nodes that supported the testbed had the same computational resources and were hosted in different cloud environments located in different data centers (geographically staggered). In order to measure the performance metrics of the network we used the Hyperledger Caliper tool that generated transactions according to the configuration we provided. This tool measured the transaction throughput (tps), the transaction latency (sec) and the success rate of the network. Our experiments were divided in two phases. In the first phase we tested the first version of the SC that was developed to perform resource management actions in a blockchain network based on the Raft mechanism. This network had fewer nodes than the one created for the second phase. The results of our first phase were adequate as it is illustrated in the third chapter of this thesis.

In the second phase of our experiment, we expanded our test networks and procced to changes in the structure of the SC. Moreover, we experimented with two consensus types: the Raft and the IBFT. In this phase of the experiment, we wanted to observe how the number of nodes and the structure of the SC (less complex implementation of functions) affect the performance of the network. Additionally, we conducted our experiments using two different consensus mechanism to realize the impact of consensus to the blockchain's performance.

The results of the experiments were extremely valuable and encouraging. The addition of an extra node didn't affect the network while the changes in the SC's structure had an impact on the performance. It should be mentioned that SC functions which implement recursive tasks (i.e., for loops) limit the performance (low tps, high latency, low success rate) of the blockchain while other functions which avoid these tasks perform well. Furthermore, our initial thoughts regarding the impact of consensus mechanisms in the performance of the network were verified. The same network with the same SC structure performs better when the Raft CFT mechanism is used. When the IBFT mechanism is applied the network presents significantly lower performance numbers as the figures in chapter 3 illustrate. Although IBFT reduces the performance of the network it introduces byzantine fault tolerance which is a feature that Raft does not support. Raft is tolerant to failures but is not tolerant to Byzantines generals' problem, which means that if a number of nodes in the network act maliciously, Raft cannot guarantee the proper functionality of the solution.

Considering the experimental results of the Raft mechanism, the functions that do not contain recursive commands present low latency and adequate transaction throughput (approximately 30 transactions per second). These results are encouraging and make the implementation of a blockchain-based resource management mechanism feasible. It should be noted that these results are based on an experiment that uses a non-BFT consensus which implies that we are willing to sacrifice the BFT feature. Having in mind that the network is private and permissionless the likelihood of multiple malicious nodes is low and therefore the selection of a non BFT consensus is an acceptable choice.

The encouraging results of our experiment motivated us to proceed to the design of a blockchain-based ZSM approach as it is displayed in Figure 23. The concept of this approach is discussed in chapter 4, where we describe how the blockchain technology could be integrated with the framework. More specific, we analyzed our proposal that is based on the inherent characteristics of blockchain technology used to address the security concerns identified by the ZSM standardization team. Furthermore, we discussed the complexity of the proposed system based on the number of parts definition and we argued that our approach solves the security issues of ZSM in a less complex manner than the solutions proposed by the standardization team.

Nevertheless, in order to increase the security of ZSM and at the same time maintain the performance level high, we proceed to the definition of requirements that the ideal blockchain should fulfill. These requirements are as follows: implementation of private/consortium permissioned network, present of crash fault tolerance, achieve low consensus convergence time, support of SCs, support of tokens and interaction with oracle mechanisms. The rationale behind the identification of the afore mentioned requirements is discussed in detail in chapter 4 of the current thesis. These requirements are extremely valuable as they help us draw the profile of the ideal blockchain solution for the implementation of the blockchain-enabled ZSM scenario.

Our research findings urged us to continue our study in order to find the most suitable DLTs for our scenario. Therefore, beyond the study of blockchain solutions, we proceed to the investigation of another promising DLT category, the Directed Acyclic Graphs (DAGs). This technology presents similarities and differences with the blockchain technology which are presented in Table 4. DAGs guarantee the transaction validity and display the same valuable security characteristics with blockchain. However, they present high-performance numbers in terms of transaction throughput and latency, while they are considered more scalable than traditional blockchains.

To this end, we examined both blockchain and DAG solutions in order to check their suitability for our use case. The Hyperledger Fabric, the Ethereum Quorum and the R3 Corda were the three examined blockchains, while the DAG category was represented by the IOTA and the Hedera Hashgraph. The conclusion of our study showed that none of the existing solutions is the ideal one for our scenario. Nevertheless, the combination of the characteristics of each DLT fulfils the requirements that the ideal solution should meet.

Concluding the current thesis, the main research findings and contributions of our study are listed as follows:

- ✓ Identification of a new research area illustrated in Figure 5
- ✓ Analysis of the blockchain technology and presentation of its non-financial applications
- ✓ Discussion and analysis of the advancements in the networking sector
- ✓ Presentation of the growing interest of academia in the use of blockchain in modern networks
- ✓ Showcase the feasibility of using blockchain for resource management in modern networks by implementing a proof-of-concept (PoC) prototype
- ✓ Evaluate the performance of the proposed solution based on the developed PoC
- ✓ Analysis of the ZSM framework and discussion of the main security issues presented by the ETSI standardization team,
- ✓ Design a novel architecture of the blockchain-based ZSM approach to address the security issues of the framework and analysis regarding the integration of blockchain with ZSM,
- ✓ Definition of requirements that a DLT solution should fulfill and examination of blockchain and DAG solutions regarding their suitability for our use case scenario,
- ✓ Identification of future research paths.

5.2 Future work

The current research opens the way for further studies in the use of DLTs for the secure resource management in NGN. Although the research area identified in this thesis is new, the interest of academia in the use of blockchain in NGNs is growing. Our study aims to increase the interest in this new research area and unveil new research directions discussed in this final chapter of the thesis. These new research paths are listed as follows:

- Design and development of a blockchain-based marketplace of resources available for sharing among NPs with the use of cryptocurrency and tokens,
- Development and optimization of SCs for the efficient management of ZSM functionalities with respect to the requirements of modern networks,
- Study, development and testing of oracle mechanisms for the interaction of the blockchain with the outside world with respect to security and requirements of NGNs,
- Development and testing of the ideal DLT solution for the implementation of secure resource management in NGNs.

Since the current study presents the concept and architecture of the blockchain-based ZSM use case in a technological perspective, where the NPs host and support the nodes of the network, the full potential of the blockchain technology can be harnessed. Therefore, a new marketplace of NPs can be designed and developed in a financial perspective, where the use of cryptocurrency and tokens could be introduced. The design and development of mechanisms that would motivate NPs to join this new market would be an extremely interesting topic, having in mind that a blockchain network becomes more resilient as the number of participants grows. The cryptocurrency could be utilized based on an already known crypto or on a new fungible token that could be used as a digital currency in this solution. Also, the use of non-fungible tokens (NFTs) could be proven very beneficial for the operation and increase of competitiveness among NPs in this new market. The development of tokens that could be used as reputation badges could play a significant role in the process of NP selection. For example, when an NP complies fully with the rules of the network and completes a large number of resource requests, a badge could be assigned to the provider that is a digital proof of NP's credibility. As a result, when a NP is searching for another one to fulfill a request, the

NP with a badge (NFT) would be preferred. This is fully in line with the reputation approach which is one of the focal points of ETSI PDL group.

Considering the impact of the SC's functionality in the performance of the network, the SC's code optimization topic is very important. The best practices regarding the development of SCs are already known to the community and should be followed by every developer, nevertheless more sophisticated solutions could be studied and developed. Having in mind that the network conditions are changing in a rapid pace and new challenges arise, the ability of the network to automatically adjust should be guaranteed. In a ZSM environment, the network management should be performed with the minimum or even zero human intervention and in our approach the functions that perform these management tasks are implemented in the form of SCs. Therefore, the ability to generate optimized SCs in an automated manner to cover the needs of the ZSM network would be a very interesting research area. This research path would include hyped technologies such as AI and ML, while the use of DLTs can be proved lifesaving for the networks. The ledger of the DLTs could store or point to useful data that can be used by the AI/ML systems to perform code optimization and to construct powerful SCs with the minimum performance impact. This would result to a completely autonomous self-managed and self-optimized network that follows the principles of ZSM and is secured by the blockchain technology.

The design of a blockchain enabled ZSM approach is based on the use of oracle mechanisms that allow the network to communicate with entities of the outside world. In the current research, the ZSM services interact with the blockchain via oracles which affect the overall system's performance. The selection of the most suitable oracle is a very tricky task as it should present increased security level and at the same time high performance numbers. It could be considered as the Achilles' heel of our system, since blockchain trusts the information received by the oracles and cannot choose when to accept or reject the data. There is always the risk of feeding the blockchain with false or garbage information which directly affects the operation and performance of the system. Therefore, a very promising research topic is the comprehensive study, development and testing of an oracle mechanism to be used in the discussed scenario. Up until now, various oracles have been developed which use different techniques to transfer in a secure manner valid information to the blockchain network. Nevertheless, new mechanisms can be created in order to cover the needs of our use case and fulfill the requirements of modern networks.

Although, at the end of this study, we investigated several DLT mechanisms, none of them presented the characteristics of the ideal solution for our use case. This automatically illustrates a new research direction towards the development and testing of the ideal solution. This research can be based on the results of the current study in order to develop a new DLT solution or to modify an existing one. Both blockchains and DAGs present significant characteristics while at the same time the requirements of modern networks can be used as a criterion to accept or reject a solution. In addition, this research path may lead to the development of new consensus mechanisms that will present crash fault tolerance, high validation times and resiliency to malicious parties. These characteristics, as well as those presented in Table 5, would lead to the development of the ideal DLT which can be adopted to guarantee the secure resource management in NGNs.

6. References

1. Al-Falahy, N., & Alani, O. Y. (2017). Technologies for 5G networks: Challenges and opportunities. It Professional, 19(1), 12-20.
2. Onoe, S. (2016, January). 1.3 Evolution of 5G mobile technology toward 1 2020 and beyond. In 2016 IEEE International Solid-State Circuits Conference (ISSCC) (pp. 23-28). IEEE.
3. Klaine, P. V., Imran, M. A., Onireti, O., & Souza, R. D. (2017). A survey of machine learning techniques applied to self-organizing cellular networks. IEEE Communications Surveys & Tutorials, 19(4), 2392-2431.
4. Monserrat, J. F., Mange, G., Braun, V., Tullberg, H., Zimmermann, G., & Bulakci, Ö. (2015). METIS research advances towards the 5G mobile and wireless system definition. EURASIP Journal on Wireless Communications and Networking, 2015(1), 1-16.
5. Intelligence, G. S. M. A. (2014). Understanding 5G: Perspectives on future technological advancements in mobile. White paper, 1-26.
6. Agiwal, M., Roy, A., & Saxena, N. (2016). Next generation 5G wireless networks: A comprehensive survey. IEEE Communications Surveys & Tutorials, 18(3), 1617-1655.
7. Site: "UK 5G Innovation Network" https://uk5g.org/discover/5g-industry/health-social-care/5g-in-medical-treatment-UK/5g-remote-robotic-surgery-UK/#:~:text=Thanks%20to%20the%20extremely%20low,the%20area%20being%20operated%20on , Accessed on 10-2022
8. Site: "HUAWEI" https://www.huawei.com/en/technology-insights/industry-insights/outlook/mobile-broadband/wireless-for-sustainability/cases/worlds-first-remote-operation-using-5g-surgery, Accessed on 10-2022
9. Gilchrist, A. (2016). Introducing Industry 4.0. In Industry 4.0 (pp. 195-215). Apress, Berkeley, CA.
10. Site: "Forbes: 5G: Moving Industry 4.0 To The Next Level" https://www.forbes.com/sites/forbestechcouncil/2021/12/01/5g-moving-industry-40-to-the-next-level/?sh=6332a0c82801 , Accessed on 10-2022
11. Site: "Ericson.com 5G for manufacturing - 5G Industry automation: A robust opportunity for operators" https://www.ericsson.com/en/reports-and-papers/5g-for-manufacturing#:~:text=5G%20networks%20offer%20manufacturers%20and,Internet%20of%20Things%20(IoT) , Accessed on 10-2022
12. Rao, S. K., & Prasad, R. (2018). Impact of 5G technologies on industry 4.0. Wireless personal communications, 100(1), 145-159.
13. Gundall, M., Strufe, M., Schotten, H. D., Rost, P., Markwart, C., Blunk, R., ... & Wübben, D. (2021). Introduction of a 5G-enabled architecture for the realization of industry 4.0 use cases. IEEE access, 9, 25508-25521.
14. Site: "IEEE Future Networks Enabling 5G and Beyond" https://futurenetworks.ieee.org/topics/5g-hardware-components-advancements-and-future-trends , Accessed on 10-2022
15. Hassan, N., Yau, K. L. A., & Wu, C. (2019). Edge computing in 5G: A review. IEEE Access, 7, 127276-127289.

16. Bauer E, Adams R. Reliability and availability of cloud computing. John Wiley & Sons; 2012 Jul 20.

17. Nadeau, T. D., & Gray, K. (2013). SDN: Software Defined Networks: an authoritative review of network programmability technologies. " O'Reilly Media, Inc.".

18. Li, Y., & Chen, M. (2015). Software-defined network function virtualization: A survey. IEEE Access, 3, 2542-2553.

19. 3GPP, TS 23.501 "System Architecture for the 5G System," Release-15, v. 2.0.1, Dec. 2017

20. Qu, L., Assi, C., Shaban, K., & Khabbaz, M. J. (2017). A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks. IEEE Transactions on Network and Service Management, 14(3), 554-568.

21. Kaur, K., Mangat, V., & Kumar, K. (2020). A comprehensive survey of service function chain provisioning approaches in SDN and NFV architecture. Computer Science Review, 38, 100298.

22. Kaloxylos, A. (2018). A survey and an analysis of network slicing in 5G networks. IEEE Communications Standards Magazine, 2(1), 60-65.

23. Trakadas, P., Karkazis, P., Leligou, H. C., Zahariadis, T., Vicens, F., Zurita, A., ... & Kyriazis, D. (2020). Comparison of management and orchestration solutions for the 5G era. Journal of Sensor and Actuator Networks, 9(1), 4.

24. Network Functions Virtualisation (NFV)—Architectural Framework. Available online: https:// www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf (accessed on 5 November 2022).

25. Open Source MANO. Available online: https://osm.etsi.org/ (accessed on 10 November 2022)

26. Reid, A., González, A., Armengol, A. E., de Blas, G. G., Xie, M., Grønsund, P., ... & Salguero, F. J. R. (2019). Osm scope, functionality, operation and integration guidelines. ETSI, White Paper.

27. SONATA - AGILE DEVELOPMENT, TESTING AND ORCHESTRATION OF SERVICES IN 5G VIRTUALIZED NETWORKS. Available online: https://www.sonata-nfv.eu/ (accessed on 10 November 2022)

28. Soenen, T.; Rossem, S.V.; Tavernier, W.; Vicens, F.; Valocchi, D.; Trakadas, P.; Karkazis, P.; Xilouris, G.;
Eardley, P.; Kolometsos, S.; et al. Insights from SONATA: Implementing and Integrating a Microservice-Based
NFV Service Platform with a DevOps Methodology. In Proceedings of the NOMS 2018—IEEE/IFIP Network
Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018.

29. Cloudify - Bridging the Gap Between Applications & Cloud Environments. Available online: https://cloudify.co/ (accessed on 10 November 2022)

30. Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. Future Generation computer systems, 28(3), 583-592.

31. Liyanage, M., Pham, Q. V., Dev, K., Bhattacharya, S., Maddikunta, P. K. R., Gadekallu, T. R., & Yenduri, G. (2022). A survey on Zero touch network and Service (ZSM) Management for 5G and beyond networks. Journal of Network and Computer Applications, 103362.

32. De Alwis, C., Kalla, A., Pham, Q. V., Kumar, P., Dev, K., Hwang, W. J., & Liyanage, M. (2021). Survey on 6G frontiers: Trends, applications, requirements, technologies and future research. IEEE Open Journal of the Communications Society, 2, 836-886.
33. ETSI Zero Touch & Service Management (ZSM). Available online: https://www.etsi.org/technologies/zero-touch-network-service-management (accessed on 13 November 2022)
34. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, 21260.
35. Blockchain overview. Available online: https://www.nist.gov/blockchain (accessed on 25 November 2022)
36. State of DApps – Dapp Statistics. Available online: https://www.stateofthedapps.com/stats (accessed on 24 November 2022)
37. Lamport, L., Shostak, R., & Pease, M. (2019). The Byzantine generals problem. In Concurrency: the works of leslie lamport (pp. 203-226).
38. Hyperledger Fabric. Available online: https://www.hyperledger.org/use/fabric (accessed on 25 November 2022)
39. Corda. Available online: https://www.corda.net/ (accessed on 25 November 2022)
40. Types of Blockchain: Public, Private, or Something in Between, Available online: https://www.foley.com/en/insights/publications/2021/08/types-of-blockchain-public-private-between (accessed on 25 November 2022)
41. (Demystifying) Blockchain Development: The BLER Use Case, Available online: https://www.cutter.com/article/demystifying-blockchain-development-bler-use-case-505691 (accessed on 08 December 2022)
42. Xevgenis, M. G., Kogias, D., Leligou, H. C., Chatzigeorgiou, C., Feidakis, M., & Patrikakis, C. Z. (2020, May). A Survey on the Available Blockchain Platforms and Protocols for Supply Chain Management. In IOT4SAFE@ ESWC.
43. Kogias, D. G., Leligou, H. C., Xevgenis, M., Polychronaki, M., Katsadouros, E., Loukas, G., ... & Patrikakis, C. Z. (2019). Toward a blockchain-enabled crowdsourcing platform. IT Professional, 21(5), 18-25.
44. Antonopoulos, A. M. (2017). Mastering Bitcoin: Programming the open blockchain. " O'Reilly Media, Inc.".
45. Antonopoulos, A. M., & Wood, G. (2018). Mastering ethereum: building smart contracts and dapps. O'reilly Media.
46. Lamport, L. (1983). The weak Byzantine generals problem. Journal of the ACM (JACM), 30(3), 668-676.
47. Sukhwani, H., Martínez, J. M., Chang, X., Trivedi, K. S., & Rindos, A. (2017, September). Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric). In 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS) (pp. 253-255). IEEE.
48. Moniz, H. (2020). The Istanbul BFT consensus algorithm. arXiv preprint arXiv:2002.03613.
49. Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm. In 2014 USENIX Annual Technical Conference (Usenix ATC 14) (pp. 305-319).

50. (Demystifying) Blockchain Development: The BLER Use Case, Available online: https://www.cutter.com/article/demystifying-blockchain-development-bler-use-case-505691 (accessed on 21 December 2022)
51. T. Bocek, B. B. Rodrigues, T. Strasser, and B. Stiller. Blockchains everywhere - a use-case of blockchains in the pharma supply-chain. In 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pages 772–777, May 2017.
52. Guido Perboli, Stefano Musso, and Mariangela Rosano. Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases. IEEE Access, 6:62018–62028, 2018.
53. Horst Treiblmaier. The impact of the blockchain on the supply chain: a theorybased research framework and a call for action. Supply Chain Management: An International Journal, 23(6):545–559, sep 2018.
54. Morgen E. Peck. Blockchain world - do you need a blockchain? this chart will tell you if the technology can solve your problem. IEEE Spectrum, 54(10):38–60, oct 2017.
55. Sin Kuang Lo, Xiwei Xu, Yin Kia Chiam, and Qinghua Lu. Evaluating suitability of applying blockchain. In 2017 22nd International Conference on Engineering of Complex Computer Systems (ICECCS). IEEE, nov 2017.
56. Karl Wust and Arthur Gervais. Do you need a blockchain? In 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). IEEE, jun 2018.
57. Qinghua Lu and Xiwei Xu. Adaptable blockchain-based systems: A case study for product traceability. IEEE Software, 34(6):21–27, nov 2017.
58. Zhijie Li, Haoyan Wu, Brian King, Zina Ben Miled, John Wassick, and Jeffrey Tazelaar. A hybrid blockchain ledger for supply chain visibility. In 2018 17th International Symposium on Parallel and Distributed Computing (ISPDC). IEEE, jun 2018.
59. Kaiwen Zhang and Hans-Arno Jacobsen. Towards dependable, scalable, and pervasive distributed ledgers with blockchains. In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). IEEE, jul 2018.
60. Congcong Ye, Guoqiang Li, Hongming Cai, Yonggen Gu, and Akira Fukuda. Analysis of security in blockchain: Case study in 51%-attack detecting. In 2018 5th International Conference on Dependable Systems and Their Applications (DSA). IEEE, sep 2018.
61. John Hargrave, Navroop K. Sahdev, and Olga Feldmeier. How value is created in tokenized assets. SSRN Electronic Journal, 2018.
62. D. Mayer, “Elance and oDesk hit by major DDoS attacks, downing services for many freelancers”, Available online: https://gigaom.com/2014/03/18/elance-hit-by-major-ddos-attack-downing-service-for-many-freelancers/ (accessed on 19 December 2022)
63. E. Newcomer, “Uber paid hackers to delete stolen data on 57 Million People”, Available online: https://www.bloomberg.com/news/articles/2017-11-21/uber-concealed-cyberattack-that-exposed-57-million-people-s-data (accessed on 19 December 2022)
64. Zhang, X., Xue, G., Yu, R., Yang, D., & Tang, J. (2015). Keep your promise: Mechanism design against free-riding and false-reporting in crowdsourcing. IEEE Internet of Things Journal, 2(6), 562-572.
65. Zhang, Y., & Van der Schaar, M. (2012, March). Reputation-based incentive protocols in crowdsourcing applications. In 2012 Proceedings IEEE INFOCOM (pp. 2140-2148). IEEE.

66. Li, M., Weng, J., Yang, A., Lu, W., Zhang, Y., Hou, L., ... & Deng, R. H. (2018). CrowdBC: A blockchain-based decentralized framework for crowdsourcing. IEEE Transactions on Parallel and Distributed Systems, 30(6), 1251-1266.
67. Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. Ieee Access, 4, 2292-2303.
68. Ast, F., & Sewrjugin, A. (2015). The crowdjury, a crowdsourced justice system for the collaboration era.
69. Yang, P., Li, Q., Yan, Y., Li, X. Y., Xiong, Y., Wang, B., & Sun, X. (2015). “Friend is treasure”: exploring and exploiting mobile social contacts for efficient task offloading. IEEE Transactions on Vehicular Technology, 65(7), 5485-5496.
70. Xevgenis, M., Kogias, D. G., Karkazis, P., Leligou, H. C., & Patrikakis, C. (2020). Application of blockchain technology in dynamic resource management of next generation networks. Information, 11(12), 570.
71. Agarwal, S., Malandrino, F., Chiasserini, C. F., & De, S. (2019). VNF placement and resource allocation for the support of vertical services in 5G networks. IEEE/ACM Transactions on Networking, 27(1), 433-446.
72. Taleb, T., Afolabi, I., Samdanis, K., & Yousaf, F. Z. (2019). On multi-domain network slicing orchestration architecture and federated resource control. IEEE Network, 33(5), 242-252.
73. Samdanis, K., Costa-Perez, X., & Sciancalepore, V. (2016). From network sharing to multi-tenancy: The 5G network slice broker. IEEE Communications Magazine, 54(7), 32-39.
74. Herbaut, N., & Negru, N. (2017). A model for collaborative blockchain-based video delivery relying on advanced network services chains. IEEE Communications Magazine, 55(9), 70-76.
75. Vukolić, M. (2015, October). The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In International workshop on open problems in network security (pp. 112-125). Springer, Cham.
76. Rebello, G. A. F., Alvarenga, I. D., Sanz, I. J., & Duarte, O. C. M. (2019, May). BSec-NFVO: A blockchain-based security for network function virtualization orchestration. In ICC 2019-2019 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.
77. Nour, B., Ksentini, A., Herbaut, N., Frangoudis, P. A., & Moungla, H. (2019). A blockchain-based network slice broker for 5G services. IEEE Networking Letters, 1(3), 99-102.
78. Rebello, G. A. F., Camilo, G. F., Silva, L. G., Guimarães, L. C., de Souza, L. A. C., Alvarenga, I. D., & Duarte, O. C. M. (2019, May). Providing a sliced, secure, and isolated software infrastructure of virtual functions through blockchain technology. In 2019 IEEE 20th International Conference on High Performance Switching and Routing (HPSR) (pp. 1-6). IEEE.
79. Xevgenis, M., Kogias, D., Christidis, I., Patrikakis, C., & Leligou, H. C. (2022). Evaluation of a blockchain-enabled resource management mechanism for NGNs. arXiv preprint arXiv:2211.00457.
80. Togou, M. A., Bi, T., Dev, K., McDonnell, K., Milenovic, A., Tewari, H., & Muntean, G. M. (2020, June). A distributed blockchain-based broker for efficient resource provisioning in 5g networks. In 2020 International wireless communications and mobile computing (IWCMC) (pp. 1485-1490). IEEE.
81. Maksymyuk, T., Gazda, J., Volosin, M., Bugar, G., Horvath, D., Klymash, M., & Dohler, M. (2020). Blockchain-empowered framework for decentralized network management in 6G. IEEE Communications Magazine, 58(9), 86-92.

82. Xu, H., Klaine, P. V., Onireti, O., Cao, B., Imran, M., & Zhang, L. (2020). Blockchain-enabled resource management and sharing for 6G communications. Digital Communications and Networks, 6(3), 261-269.
83. Papadakis-Vlachopapadopoulos, K., Dimolitsas, I., Dechouniotis, D., Tsiropoulou, E. E., Roussaki, I., & Papavassiliou, S. (2021, February). On blockchain-based cross-service communication and resource orchestration on edge clouds. In Informatics (Vol. 8, No. 1, p. 13). MDPI.
84. Hewa, T., Gür, G., Kalla, A., Ylianttila, M., Bracken, A., & Liyanage, M. (2020). The role of blockchain in 6G: Challenges, opportunities and research directions. 2020 2nd 6G Wireless Summit (6G SUMMIT), 1-5.
85. Praveen, G., Chamola, V., Hassija, V., & Kumar, N. (2020). Blockchain for 5G: A prelude to future telecommunication. Ieee Network, 34(6), 106-113.
86. 5G Zorro, Available online: https://www.5gzorro.eu/ (accessed on 22 December 2022)
87. Benzaïd, C., Taleb, T., & Farooqi, M. Z. (2021). Trust in 5G and beyond networks. IEEE Network, 35(3), 212-222.
88. Benzaid, C., & Taleb, T. (2020). AI-driven zero touch network and service management in 5G and beyond: Challenges and research directions. IEEE Network, 34(2), 186-194.
89. Carrozzo, G., Siddiqui, M. S., Betzler, A., Bonnet, J., Perez, G. M., Ramos, A., & Subramanya, T. (2020, June). AI-driven zero-touch operations, security and trust in multi-operator 5G networks: a conceptual architecture. In 2020 European Conference on Networks and Communications (EuCNC) (pp. 254-258). IEEE.
90. Theodorou, V., Lekidis, A., Bozios, T., Meth, K., Fernández-Fernández, A., Tavlor, J., ... & Behravesh, R. (2021, June). Blockchain-based Zero Touch Service Assurance in Cross-domain Network Slicing. In 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit) (pp. 395-400). IEEE.
91. ETSI, G. (2019). Zero-touch network and service management (zsm); reference architecture. Group Specification (GS) ETSI GS ZSM, 2.
92. ETSI, G. (2021). Zero-touch network and Service Management (ZSM); General Security Aspects. Tech. Rep.
93. Uzunidis, D., Karkazis, P., Roussou, C., Patrikakis, C., & Leligou, H. C. (2021). Intelligent Performance Prediction: The Use Case of a Hadoop Cluster. Electronics, 10(21), 2690.
94. Subramanya, T., & Riggio, R. (2021). Centralized and federated learning for predictive VNF autoscaling in multi-domain 5G networks and beyond. IEEE Transactions on Network and Service Management, 18(1), 63-78.
95. Dalgkitsis, A., Mekikis, P. V., Antonopoulos, A., Kormentzas, G., & Verikoukis, C. (2020, December). Dynamic Resource Aware VNF Placement with Deep Reinforcement Learning for 5G Networks. In GLOBECOM 2020-2020 IEEE Global Communications Conference (pp. 1-6). IEEE.
96. Boudi, A., Bagaa, M., Pöyhönen, P., Taleb, T., & Flinck, H. (2021). AI-based resource management in beyond 5G cloud native environment. IEEE Network, 35(2), 128-135.
97. ETSI, G. (2021). Zero-touch network and Service Management (ZSM); Closed-Loop Automation Part 1: Enablers.

98. Siriwardhana, Y., Porambage, P., Liyanage, M., & Ylianttila, M. (2021, June). AI and 6G security: Opportunities and challenges. In 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit) (pp. 616-621). IEEE.
99. Chaer, A., Salah, K., Lima, C., Ray, P. P., & Sheltami, T. (2019, December). Blockchain for 5G: Opportunities and challenges. In 2019 IEEE Globecom Workshops (GC Wkshps) (pp. 1-6). IEEE.
100. Tahir, M., Habaebi, M. H., Dabbagh, M., Mughees, A., Ahad, A., & Ahmed, K. I. (2020). A review on application of blockchain in 5G and beyond networks: Taxonomy, field-trials, challenges and opportunities. IEEE Access, 8, 115876-115904.
101. Nguyen, D. C., Pathirana, P. N., Ding, M., & Seneviratne, A. (2020). Blockchain for 5G and beyond networks: A state of the art survey. Journal of Network and Computer Applications, 166, 102693.
102. Quorum. Available online: https://docs.goquorum.consensys.net/en/latest/ (accessed on 10 January 2023).
103. Trakadas, P., Karkazis, P., Leligou, H. C., Zahariadis, T., Vicens, F., Zurita, A., ... & Kyriazis, D. (2020). Comparison of management and orchestration solutions for the 5G era. Journal of Sensor and Actuator Networks, 9(1), 4.
104. Al-Hazmi, Y., Gonzalez, J., Rodriguez-Archilla, P., Alvarez, F., Orphanoudakis, T., Karkazis, P., & Magedanz, T. (2014, September). Unified representation of monitoring information across federated cloud infrastructures. In 2014 26th International Teletraffic Congress (ITC) (pp. 1-6). IEEE.
105. Theodore, Z.; Panagiotis, K.; Sotiris, K.; George, X. A Monitoring Framework for Heterogeneous NFV/SDN-enabledCloud Environments. In Proceedings of the EuCNC 2016, Athens, Greece, 27–30 June 2016
106. Trakadas, P., Karkazis, P., Leligou, H. C., Zahariadis, T., Tavernier, W., Soenen, T., ... & Miguel Contreras Murillo, L. (2018). Scalable monitoring for multiple virtualized infrastructures for 5G services. In SoftNetworking 2018, The International Symposium on Advances in Software Defined Networking and Network Functions Virtualization (pp. 1-4).
107. Lo, S. K., Xu, X., Staples, M., & Yao, L. (2020). Reliability analysis for blockchain oracles. Computers & Electrical Engineering, 83, 106582.
108. Ladleif, J., Weber, I., & Weske, M. (2020, September). External data monitoring using oracles in blockchain-based process execution. In International Conference on Business Process Management (pp. 67-81). Springer, Cham.
109. Tessera. Available online: https://docs.tessera.consensys.net/en/stable/ (accessed on 10 January 2023)
110. Hyperledger Caliper. Available online: https://www.hyperledger.org/learn/publications/blockchain-performancemetrics# (accessed on 12 January 2023)
111. Okeanos. Available online: https://okeanos.grnet.gr/home/ (accessed on 12 January 2023)
112. Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm. In 2014 USENIX Annual Technical Conference (Usenix ATC 14) (pp. 305-319).
113. Lamport, L. (2001). Paxos made simple. ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001), 51-58.
114. Moniz, Henrique. "The Istanbul BFT consensus algorithm." arXiv preprint arXiv:2002.03613 (2020).
115. Provable. Available online: https://provable.xyz/ (accessed on 13 January 2023)

116. Chainlink. Available online: https://chain.link/ (accessed on 13 January 2023)
117. Al-Breiki, H., Rehman, M. H. U., Salah, K., & Svetinovic, D. (2020). Trustworthy blockchain oracles: review, comparison, and open research challenges. IEEE Access, 8, 85675-85685.
118. Leligou, H. C., Kanonakis, K., Angelopoulos, J., Pountourakis, I., & Orphanoudakis, T. (2006). Efficient burst aggregation for QoS-aware slotted OBS systems. European Transactions on Telecommunications, 17(1), 93-98.
119. Andrew R. Short, Theofanis G. Orfanoudakis, Helen C. Leligou, "Improving Security and Fairness in Federated Learning systems", International Journal of Network Security and its applications, Vol.13, No. 6, 2021, DOI: 10.5121/ijnsa.2021.13604
120. Xevgenis, M.; Kogias, D.G.; Karkazis, P.A.; Leligou, H.C. Addressing ZSM Security Issues with Blockchain Technology. Future Internet 2023, 15, 129. https://doi.org/10.3390/fi15040129
121. Standish, R. K. (2008). Concept and definition of complexity. In Intelligent complex adaptive systems (pp. 105-124). IGI Global.
122. Central Blockchain Council of America: DAG the DLT! Directed Acyclic Graph for Enterprise Blockchain! Available online: https://www.cbcamerica.org/blockchain-insights/dag-the-dlt-directed-acyclic-graph-for-enterprise-blockchain (accessed on 28 January 2023)
123. Hyperledger Fabric Documentation: Latest Release 25/01/2023. Available online: https://hyperledger-fabric.readthedocs.io/_/downloads/vi/latest/pdf/ (accessed on 28 January 2023)
124. Gorenflo, C., Lee, S., Golab, L., & Keshav, S. (2020). FastFabric: Scaling hyperledger fabric to 20 000 transactions per second. International Journal of Network Management, 30(5), e2099.
125. Baliga, A., Subhod, I., Kamat, P., & Chatterjee, S. (2018). Performance evaluation of the quorum blockchain platform. arXiv preprint arXiv:1809.03421.
126. Corda: A distributed ledger. Available online: https://www.r3.com/blog/corda-technical-whitepaper/ (accessed on 28 January 2023)
127. R3 Corda. Available online: https://www.r3.com/products/corda/ (accessed on 1 February 2023)
128. Sealey, N., Aijaz, A., & Holden, B. (2022, November). IOTA Tangle 2.0: Toward a Scalable, Decentralized, Smart, and Autonomous IoT Ecosystem. In 2022 International Conference on Smart Applications, Communications and Networking (SmartNets) (pp. 01-08). IEEE.
129. IOTA Smart Contracts. Available online: https://wiki.iota.org/shimmer/smart-contracts/overview/ (accessed on 2 February 2023)
130. IOTA Oracles. Available online: https://blog.iota.org/introducing-iota-oracles/ (accessed on 2 February 2023)
131. Hedera How it works. Available online: https://hedera.com/how-it-works (accessed on 3 February 2023)
132. What is Chainlink: A beginner's guide. Available online: https://blog.chain.link/what-is-chainlink/?_ga=2.209069778.1120344513.1675428709-842934195.1673628852 (accessed on 3 February 2023)

Appendix A

In this section of the thesis, information regarding the implementation of our testbed is presented. In our experiments, we used an Ethereum-based blockchain solution the Quorum in order to form a private and permissioned network. Below a comprehensive guide is available in the form of documentation which is the result of our technical effort. This guide has been used for the successful setup of the blockchain network.

The files used for the implementation of our testbed are available in the following github site: https://github.com/mxevgenis/quorum-network

# quorum-network

Quorum Net files

This is the content of fromscratch file. Use this files and folders which are based in the deployment of Quorum using Tessera.

Below you may find some usefull information regarding the Quorum platform and its components. Also you will find a step by step guide for the deployment of a Quorum network. The files and folders contained in this project are the result of a deployment that followed this guide.

Quorum is an Ethereum-based distributed ledger protocol that has been developed to provide industries such as finance, supply chain, retail, real estate, etc. with a permissioned implementation of Ethereum that supports transaction and contract privacy.

Quorum includes a minimalistic fork of the Go Ethereum client (a.k.a geth), and as such, leverages the work that the Ethereum developer community has undertaken.

The primary features of Quorum, and therefore extensions over public Ethereum, are:

- Transaction and contract privacy
- Multiple voting-based consensus mechanisms
- Network/Peer permissions management
- Higher performance

Quorum currently includes the following components:

- Quorum Node (modified Geth Client)
- Privacy Manager (Constellation/Tessera)

- Transaction Manager
- Enclave

Constellation & Tessera

Constellation and Tessera are Haskell and Java implementations of a general-purpose system for submitting information in a secure way. They are comparable to a network of MTA (Message Transfer Agents) where messages are encrypted with PGP. It is not blockchain-specific, and are potentially applicable in many other types of applications where you want individually-sealed message exchange within a network of counterparties. The Constellation and Tessera modules consist of two sub-modules:

- The Node (which is used for Quorum’s default implementation of a PrivateTransactionManager)
- The Enclave
- Transaction Manager

Quorum’s Transaction Manager is responsible for Transaction privacy. It stores and allows access to encrypted transaction data, exchanges encrypted payloads with other participant’s Transaction Managers but does not have access to any sensitive private keys. It utilizes the Enclave for cryptographic functionality (although the Enclave can optionally be hosted by the Transaction Manager itself.)

The Transaction Manager is restful/stateless and can be load balanced easily.

The Enclave

Distributed Ledger protocols typically leverage cryptographic techniques for transaction authenticity, participant authentication, and historical data preservation (i.e. through a chain of cryptographically hashed data.) In order to achieve a separation of concerns, as well as to provide performance improvements through parallelization of certain crypto-operations, much of the cryptographic work including symmetric key generation and data encryption/decryption is delegated to the Enclave.

The Enclave works hand in hand with the Transaction Manager to strengthen privacy by managing the encryption/decryption in an isolated way. It holds private keys and is essentially a “virtual HSM” isolated from other components.

Setup of a Quorum testbed using Raft consensus

There are two different ways to setup a fully functional Quorum blockchain. The easiest way is to use quorum examples where the development of the network is done in a fully automated manner. The most popular example is the 7nodes where a VM is created using vagrant (requires the existence of Virtual Box program). Within this VM 7 fully functional nodes are deployed and they form the Quorum network. For more information regarding the setup of 7nodes example please refer here.

However in this documentation we will present a step by step deployment, where we create a network from scratch. The goal of this process is to setup a Quorum network consisted from at least two nodes, where each node is hosted in a VM. These VMs will communicate over the internet and are members of the Quorum. This guide includes every single step from the moment we access the vanilla VM instance to the moment we deploy a Smart Contract. The characteristics of each VM are as follows:

OS: Ubuntu Server 16.04 LTS

CPU: 2vCPUs

RAM: 4GB

Storage: 30GB

IP: 1 Public IP

Firewall status: OFF

Setup the first VM

Access to VMs: The access to the VMs should be made by using the public key for security reasons. When we configure the access using the public key we should disable the access using password.

ssh-copy-id <username>@<domain or IP> : Copy your public key to the VM

sudo nano /etc/ssh/sshd_config : Access the configuration file and change Password authentication to no.

When you are in your VM enable your Ubuntu firewall:

sudo ufw enable : Enable firewall

sudo ufw allow 22 : Allow port 22 for ssh

sudo ufw allow 35570 : Allow this port for Quorum sudo ufw allow 50000 : Allow this port for Quorum

sudo ufw allow 21000 : Allow this port for Quorum

sudo ufw allow 9001 : Allow this port for Tessera

sudo ufw allow 9003 : Allow this port for Tessera

sudo ufw allow 9081 : Allow this port for Tessera

sudo ufw reload : Reload firewall

Then edit the source list to use the global:

sudo nano /etc/apt/sources.list : Edit this file and replace every gr. or us. to empty space.

Then proceed to the installation of several packages (Ethereum, docker) as it is displayed below:

```
############# Basic packets ###########################
sudo apt-get install -y software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get -y install ethereum
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce
sudo usermod -a -G docker $USER
sudo reboot
######### Docker images GO Geth env ################
docker pull quorumengineering/quorum
docker pull quorumengineering/tessera
docker pull quorumengineering/constellation
sudo apt-get update
sudo apt-get -y upgrade
wget https://dl.google.com/go/go1.12.7.linux-amd64.tar.gz
sudo tar -xvf go1.12.7.linux-amd64.tar.gz
sudo mv go /usr/local
export GOROOT=/usr/local/go
export GOPATH=$HOME/Projects/Proj1
export PATH=$GOPATH/bin:$GOROOT/bin:$PATH
go version
go env
git clone https://github.com/jpmorganchase/quorum.git
sudo apt-get install -y make
sudo apt-get install -y build-essential
cd quorum
make all
##################### Add at the end of ~/.bashrc the following ############
export GOROOT=/usr/local/go
export GOPATH=$HOME/Projects/Proj1
export PATH=$GOPATH/bin:$GOROOT/bin:$PATH
export PATH=/home/user/quorum/build/bin:$PATH

####################### Create the network #########################
mkdir fromscratch
cd fromscratch
mkdir new-node-1
geth --datadir new-node-1 account new
ls new-node-1/keystore
nano genesis.json

{

  "alloc": {

    "0x<Replace with the account id you created above>": {

      "balance": "1000000000000000000000000000"

    }

},

 "coinbase": "0x0000000000000000000000000000000000000000",

 "config": {

   "homesteadBlock": 0,

   "byzantiumBlock": 0,

   "constantinopleBlock": 0,
      "chainId": 10,

      "eip150Block": 0,

      "eip155Block": 0,

      "eip150Hash":
"0x0000000000000000000000000000000000000000000000000000000000000000",

      "eip158Block": 0,

      "maxCodeSize": 35,

      "maxCodeSizeChangeBlock" : 0,

      "isQuorum": true

    },

    "difficulty": "0x0",

    "extraData":
"0x0000000000000000000000000000000000000000000000000000000000000000",

    "gasLimit": "0xE0000000",

    "mixhash":
"0x00000000000000000000000000000000000000647572616c65787365646c6578",

    "nonce": "0x0",

    "parentHash":
"0x0000000000000000000000000000000000000000000000000000000000000000",

    "timestamp": "0x00"

   }
   bootnode --genkey=nodekey
   cp nodekey new-node-1/
   bootnode --nodekey=new-node-1/nodekey --writeaddress > new-node-1/enode
   cat new-node-1/enode
```

```
nano static-nodes.json

 [

  "enode://<Replace with the above node ID>@<Replace with the Public IP of your VM>:21000?discport=0&raftport=50000"

]

cp static-nodes.json new-node-1

geth --datadir new-node-1 init genesis.json

nano startnode1.sh

#!/bin/bash

PRIVATE_CONFIG=/yourpath/new-node-1t/tm.ipc nohup geth --datadir new-node-1 --nodiscover --verbosity 5 --networkid 31337 --raft --raftport 50000 --rpc --rpcaddr 0.0.0.0 --rpcport 22000 --rpcapi admin,db,eth,debug,miner,net,shh,txpool,personal,web3,quorum,raft --emitcheckpoints --port 21000 >> node.log 2>&1 &
```

The above configuration uses the tessera component which is configures below. If you do not want to use tessera add the following lines in this file and ignore the tessera section.

```
#!/bin/bash

PRIVATE_CONFIG=ignore nohup geth --datadir new-node-1 --nodiscover --verbosity 5 --networkid 31337 --raft --raftport 50000 --raftjoinexisting 2 --rpc --rpcaddr 0.0.0.0 --rpcport 22000 --rpcapi admin,db,eth,debug,miner,net,shh,txpool,personal,web3,quorum,raft --emitcheckpoints --port 21000 2>>node2.log &

chmod +x startnode1.sh

./startnode1.sh  // DO NOT EXECUTE THIS if you are using tessera. First we should set up tessera and then execute it,

geth attach new-node-1/geth.ipc
```

Tessera deployment node 1

In order to install tessera you have first to install Java and the proper JDK. The selection of the correct JDK depends on the tessera version.In our use case we have downloaded the tessera-app-0.10.4-app.jar and therefore we will install the JDK 11.

Install Java:

```
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update -q
sudo apt install -y openjdk-11-jdk
```

Then download the tessera app.

```
cd~
wget
https://oss.sonatype.org/service/local/repositories/releases/content/com/jpmorgan/quorum/tessera-app/0.10.4/tessera-app-0.10.4-app.jar
mv tessera-app-0.10.4-app.jar tessera.jar
```

if you want to know the path of tessera.jar run pwd.

Inside the directory fromscratch do the following:

```
mkdir new-node-1t
cd new-node-1t
java -jar <put tessera.jar path>/tessera.jar -keygen -filename new-node-1
```

Then create the config.json:

(the path with different format should be replaced by yours if it is different)

```
cd /home/user/quorum/fromscratch/new-node1t/
nano config.json
{

  "useWhiteList": false,

  "jdbc": {

     "username": "sa",

     "password": "",

     "url": "jdbc:h2/home/user/quorum/fromscratch/new-node-1t/db1;MODE=Oracle;TRACE_LEVEL_SYSTEM_OUT=0",

     "autoCreateTables": true

  },

  "serverConfigs":[

     {

        "app":"ThirdParty",
        "enabled": true,
        "serverAddress": "http://ip_of_this_node:9081",
        "communicationType" : "REST"
     },
     {
        "app":"Q2T",
        "enabled": true,
        "serverAddress":"unix: /home/user/quorum/fromscratch/new-node-1t/tm.ipc",
        "communicationType" : "REST"
     },
     {
        "app":"P2P",
        "enabled": true,
        "serverAddress":"http://ip_of_this_node:9001",
        "sslConfig": {
            "tls": "OFF"
        },
        "communicationType" : "REST"
     }
  ],
  "peer": [
    {
      "url": "http://ip_of_this_node:9001"
    },
    {
      "url": "http://ip_of_the_other_node:9003"
    }
  ],
  "keys": {
    "passwords": [],
    "keyData": [
      {
        "privateKeyPath": "/home/user/quorum/fromscratch/new-node-1t/new-node-1.key",
        "publicKeyPath": "/home/user/quorum/fromscratch/new-node-1t/new-node-1.pub"
      }
    ]
  },
  "alwaysSendTo": []
}
```

To start your Tessera node go to:

```
cd  /home/user/quorum/fromscratch/new-node-1t
```

```
java -jar /home/user/tessera.jar -configfile config.json >> tessera.log 2>&1 &
```

Then start your node :

```
cd /home/user/quorum/fromscratch
./startnode1.sh
geth attach new-node-1/geth.ipc
```

Setup the second VM

Access to VMs: The access to the VMs should be made by using the public key for security reasons. When we configure the access using the public key we should disable the access using password.

```
ssh-copy-id <username>@<domain or IP> : Copy your public key to the VM
sudo nano /etc/ssh/sshd_config : Access the configuration file and change Password authentication to no.
```

When you are in your VM enable your Ubuntu firewall:

```
sudo ufw enable : Enable firewall
sudo ufw allow 22 : Allow port 22 for ssh
sudo ufw allow 35570 : Allow this port for Quorum
sudo ufw allow 50000 : Allow this port for Quorum
sudo ufw allow 21000 : Allow this port for Quorum
sudo ufw allow 9001 : Allow this port for Tessera
sudo ufw allow 9003 : Allow this port for Tessera
sudo ufw allow 9081 : Allow this port for Tessera
sudo ufw reload : Reload firewall
```

Then edit the source list to use the global:

```
sudo nano /etc/apt/sources.list : Edit this file and replace every gr. or us. to empty space
```

Then proceed to the installation of several packages (Ethereum, docker) as it is displayed below:

```
############# Basic packets ###########################

sudo apt-get install -y software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get -y install ethereum
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt-get update
sudo apt-get install -y docker-ce
sudo usermod -a -G docker $USER
sudo reboot
######### Docker images GO Geth env ################
```

```
docker pull quorumengineering/quorum
docker pull quorumengineering/tessera
docker pull quorumengineering/constellation
sudo apt-get update
sudo apt-get -y upgrade
wget https://dl.google.com/go/go1.12.7.linux-amd64.tar.gz
sudo tar -xvf go1.12.7.linux-amd64.tar.gz
sudo mv go /usr/local
export GOROOT=/usr/local/go
export GOPATH=$HOME/Projects/Proj1
export PATH=$GOPATH/bin:$GOROOT/bin:$PATH
go version
go env
git clone https://github.com/jpmorganchase/quorum.git
sudo apt-get install -y make
sudo apt-get install -y build-essential
cd quorum
make all
##################### Add at the end of ~/.bashrc the following ############

export GOROOT=/usr/local/go

export GOPATH=$HOME/Projects/Proj1

export PATH=$GOPATH/bin:$GOROOT/bin:$PATH

export PATH=/home/user/quorum/build/bin:$PATH

####################### Create the network  #########################
mkdir fromscratch
cd fromscratch
mkdir new-node-2
bootnode --genkey=nodekey2
cp nodekey2 new-node-2/nodekey
bootnode --nodekey=new-node-2/nodekey –writeaddress
nano genesis.json

{

  "alloc": {
      "0x<Replace with the account id you entered in the genesis file of node 1>": {
        "balance": "1000000000000000000000000000"
      }
    },
    "coinbase": "0x0000000000000000000000000000000000000000",
    "config": {
      "homesteadBlock": 0,
      "byzantiumBlock": 0,
      "constantinopleBlock": 0,
      "chainId": 10,
      "eip150Block": 0,
      "eip155Block": 0,
      "eip150Hash":
"0x0000000000000000000000000000000000000000000000000000000000000000",
      "eip158Block": 0,
      "maxCodeSize": 35,
      "maxCodeSizeChangeBlock" : 0,
      "isQuorum": true
    },
    "difficulty": "0x0",
    "extraData":
"0x0000000000000000000000000000000000000000000000000000000000000000",
    "gasLimit": "0xE0000000",

    "mixhash": "0x00000000000000000000000000000000000000647572616c65787365646c6578",

    "nonce": "0x0",

    "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000",

    "timestamp": "0x00"

    }
```

```
nano static-nodes.json
```

```
 [

  "enode://<Replace with the node ID of node 1>@<Replace with the Public IP of node 1>:21000?discport=0&raftport=50000", "enode://<Replace with the above node ID which is result of bootnode --nodekey=new-node-2/nodekey –writeaddress >@<Replace with the Public IP of your VM>:21000?discport=0&raftport=50000"

]
```

Copy the content of the static-nodes.json that you created and paste it to the static-nodes.json files of your first node. Every static-nodes.json file should be updated!!!

```
geth --datadir new-node-2 init genesis.json
```

```
nano startnode2.sh
#!/bin/bash
PRIVATE_CONFIG=/yourpath/new-node-1t/tm.ipc nohup geth --datadir new-node-2 --nodiscover --verbosity 5 --networkid 31337 --raft --raftport 50000 --rpc --rpcaddr 0.0.0.0 --rpcport 22000 --rpcapi admin,db,eth,debug,miner,net,shh,txpool,personal,web3,quorum,raft --emitcheckpoints --port 21000 >> node.log 2>&1 &
```

The above configuration uses the tessera component which is configured below. If you do not want to use tessera add the following lines in this file and ignore the tessera section.

```
#!/bin/bash
PRIVATE_CONFIG=ignore nohup geth --datadir new-node-2 --nodiscover --verbosity 5 --networkid 31337 --raft --raftport 50000 --raftjoinexisting 2 --rpc --rpcaddr 0.0.0.0 --rpcport 22000 --rpcapi
```

```
admin,db,eth,debug,miner,net,shh,txpool,personal,web3,quorum,raft --emitcheckpoints --port 21000 2>>node2.log &
chmod +x startnode2.sh
```

First you should add this node to an active peer that is the first you have created. Then start your node :

Go to the first node:

```
cd /home/user/quorum/fromscratch
geth attach new-node-1/geth.ipc
raft.addPeer(‘enode://<node id of the second node>@<IP of the second node> :21000?discport=0&raftport=50000')
exit
```

Then go to the second node and do the following:

```
cd /home/user/quorum/fromscratch
./startnode2.sh  // DO NOT EXECUTE THIS if you are using tessera. First we should set up tessera and then execute it,
geth attach new-node-2/geth.ipc
raft.cluster
```

Tessera deployment node 2

In order to install tessera you have first to install Java and the proper JDK. The selection of the correct JDK depends on the tessera version. In our use case we have downloaded the tessera-app-0.10.4-app.jar and therefore we will install the JDK 11.

Install Java:

```
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update -q
sudo apt install -y openjdk-11-jdk
```

Then download the tessera app.

```
cd~
wget https://oss.sonatype.org/service/local/repositories/releases/content/com/jpmorgan/quorum/tessera-app/0.10.4/tessera-app-0.10.4-app.jar
mv tessera-app-0.10.4-app.jar tessera.jar
```

If you want to now the path of tessera.jar run pwd.

Inside the directory fromscratch do the following:

```
mkdir new-node-2t
cd new-node-2t
java -jar <put tessera.jar path>/tessera.jar -keygen -filename new-node-2
```

Then create the config.json:

(the path that with different format should be replaced by yours if it is different)

```
cd /home/user/quorum/fromscratch/new-node2t/
nano config.json
{
    "useWhiteList": false,

    "jdbc": {

      "username": "sa",

      "password": "",

      "url": "jdbc:h2/home/user/quorum/fromscratch/new-node-
1t/db1;MODE=Oracle;TRACE_LEVEL_SYSTEM_OUT=0",

      "autoCreateTables": true

    },

    "serverConfigs":[

      {

        "app":"ThirdParty",

        "enabled": true,

        "serverAddress": "http://ip_of_this_node:9081",

        "communicationType" : "REST"

      },

      {

        "app":"Q2T",

        "enabled": true,

         "serverAddress":"unix: /home/user/quorum/fromscratch/new-node-
1t/tm.ipc",

        "communicationType" : "REST"

      },
    {
        "app":"P2P",
        "enabled": true,
        "serverAddress":"http://ip_of_this_node:9003",
        "sslConfig": {
            "tls": "OFF"
        },
        "communicationType" : "REST"
    }
],
"peer": [
    {
        "url": "http://ip_of_the_other_node:9001"
    },
    {
        "url": "http://ip_of_this_node:9003"
    }
],
"keys": {
    "passwords": [],
    "keyData": [
            {
               "privateKeyPath": "/home/user/quorum/fromscratch/new-node-1t/new-node-1.key",
               "publicKeyPath": "/home/user/quorum/fromscratch/new-node-1t/new-node-1.pub"
            }
         ]
      },
      "alwaysSendTo": []
   }
```

To start your Tessera node go to:

```
cd  /home/user/quorum/fromscratch/new-node-2t
java -jar /home/user/tessera.jar -configfile config.json >> tessera.log 2>&1 &
```

First you should add this node to an active peer that is the first you have created. Then start your node :

Go to the first node:

```
cd /home/user/quorum/fromscratch
geth attach new-node-1/geth.ipc
raft.addPeer(‘enode://<node id of the second node>@<IP of the second node>:21000?discport=0&raftport=50000')
exit
```

Then go to the second node and do the following:

```
cd /home/user/quorum/fromscratch
./startnode2.sh
geth attach new-node-2/geth.ipc
raft.cluster
```

If you have completed all the steps as it was described then you should be able to see the raft cluster with two active nodes.

Appendix B

This section of the thesis presents the code that implements the scenario of resource management in NGNs. This code is applied in a quorum blockchain network and uses the web3 py library. The configuration files and the code used in our research can be found in the following github site: https://github.com/mxevgenis/blockchain_NPs

The code contained in this project uses the web3 py library for interacting with a Quorum blockchain network. The deploy script performs a connection to the blockchain network and deploys the smart contract defined by the ABI and the Bytecode.

When the SC is deployed, its address is stored in a json file which is later used for calling the SC's functions. The interact contract is used for creating accounts for the Network Providers and fund them with 100 ether.

A NP is characterized by:

a) name,

b) offered resources,

c) reserved resources,

d) cost,

e) domain,

f)sla

In order to conduct our experiments, we select an NP who wants extra resources, and we search among the NPs in order to select the valid candidate from which the NP should borrow resources. The SC firstly searches which NP can fulfil the requirements in matters of resources and focuses on the cost parameter. The cheapest provider with the required resources wins.

# blockchain_NPs

```
web3_SCNP_deploy.py

import  json
from web3 import Web3
from web3.middleware import geth_poa_middleware



#ganache_url = "http://127.0.0.1:7545"
```

```
vbox_url= "http://192.168.1.41:22000"

#web3 = Web3(Web3.HTTPProvider(ganache_url))

web3 = Web3(Web3.HTTPProvider(vbox_url))

#print(web3.isConnected())

web3.middleware_onion.inject(geth_poa_middleware, layer=0)

web3.eth.defaultAccount = web3.eth.accounts[0]

web3.parity.personal.unlock_account(web3.eth.defaultAccount,"", 3600)




#abi
=json.loads('[{"constant":true,"inputs":[{"name":"demand_resources","typ
e":"uint256"}],"name":"getBestMatch","outputs":[{"name":"","type":"uint2
56[]"},{"name":"","type":"uint256[]"},{"name":"","type":"uint256"},{"nam
e":"","type":"address"}],"payable":false,"stateMutability":"view","type"
:"function"},{"constant":true,"inputs":[],"name":"np_count","outputs":[{
"name":"","type":"uint256"}],"payable":false,"stateMutability":"view","t
ype":"function"},{"constant":true,"inputs":[{"name":"","type":"uint256"}
],"name":"NetProvtoOwner","outputs":[{"name":"","type":"address"}],"paya
ble":false,"stateMutability":"view","type":"function"},{"constant":false
,"inputs":[{"name":"_result","type":"uint256"},{"name":"_demand_resource
s","type":"uint256"}],"name":"transaction","outputs":[],"payable":true,"
stateMutability":"payable","type":"function"},{"constant":false,"inputs"
:[{"name":"name","type":"string"},{"name":"offered_resources","type":"ui
nt256"},{"name":"reserved_resources","type":"uint256"},{"name":"cost","t
ype":"uint256"},{"name":"domain","type":"string"},{"name":"sla","type":"
uint256"},{"name":"_address","type":"address"}],"name":"addNetworkProvid
er","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"
function"},{"constant":true,"inputs":[{"name":"demand_resources","type":
"uint256"}],"name":"get_request_resources","outputs":[{"name":"","type":
"bool"}],"payable":false,"stateMutability":"view","type":"function"},{"c
onstant":true,"inputs":[{"name":"","type":"address"}],"name":"HasNetProv
","outputs":[{"name":"","type":"bool"}],"payable":false,"stateMutability
":"view","type":"function"},{"constant":true,"inputs":[{"name":"","type"
:"uint256"}],"name":"NetworkProviders","outputs":[{"name":"name","type":
"string"},{"name":"offered_resources","type":"uint256"},{"name":"reserve
d_resources","type":"uint256"},{"name":"cost","type":"uint256"},{"name":
"domain","type":"string"},{"name":"sla","type":"uint256"}],"payable":fal
se,"stateMutability":"view","type":"function"},{"inputs":[],"payable":fa
lse,"stateMutability":"nonpayable","type":"constructor"}]')
#bytecode =”40451000dsd…..013601fghv015”

#### Below contract cost per resource #####################

abi
=json.loads('[{"constant":true,"inputs":[{"name":"demand_resources","typ
e":"uint256"}],"name":"getBestMatch","outputs":[{"name":"","type":"uint2
56[]"},{"name":"","type":"uint256[]"},{"name":"","type":"uint256"},{"nam
e":"","type":"address"}],"payable":false,"stateMutability":"view","type"
:"function"},{"constant":true,"inputs":[],"name":"np_count","outputs":[{
"name":"","type":"uint256"}],"payable":false,"stateMutability":"view","t
ype":"function"},{"constant":true,"inputs":[{"name":"","type":"uint256"}
],"name":"NetProvtoOwner","outputs":[{"name":"","type":"address"}],"paya
ble":false,"stateMutability":"view","type":"function"},{"constant":false
,"inputs":[{"name":"_result","type":"uint256"},{"name":"_demand_resource
s","type":"uint256"}],"name":"transaction","outputs":[],"payable":true,"
stateMutability":"payable","type":"function"},{"constant":false,"inputs"
:[{"name":"name","type":"string"},{"name":"offered_resources","type":"ui
nt256"},{"name":"reserved_resources","type":"uint256"},{"name":"cost","t
ype":"uint256"},{"name":"domain","type":"string"},{"name":"sla","type":"
uint256"},{"name":"_address","type":"address"}],"name":"addNetworkProvid
er","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"
function"},{"constant":true,"inputs":[{"name":"demand_resources","type":
"uint256"}],"name":"get_request_resources","outputs":[{"name":"","type":
"bool"}],"payable":false,"stateMutability":"view","type":"function"},{"c
onstant":true,"inputs":[{"name":"","type":"address"}],"name":"HasNetProv
","outputs":[{"name":"","type":"bool"}],"payable":false,"stateMutability
":"view","type":"function"},{"constant":true,"inputs":[{"name":"","type"
:"uint256"}],"name":"NetworkProviders","outputs":[{"name":"name","type":
"string"},{"name":"offered_resources","type":"uint256"},{"name":"reserve
d_resources","type":"uint256"},{"name":"cost","type":"uint256"},{"name":
"domain","type":"string"},{"name":"sla","type":"uint256"}],"payable":fal
se,"stateMutability":"view","type":"function"},{"inputs":[],"payable":fa
lse,"stateMutability":"nonpayable","type":"constructor"}]')
bytecode = “0084561……..43240jf023”


def _initialize_NPcon(abi,bytecode):
    NPs_match = web3.eth.contract(abi=abi, bytecode=bytecode)
    tx_hash = NPs_match.constructor().transact()
    tx_receipt = web3.eth.waitForTransactionReceipt(tx_hash)
    address=tx_receipt.contractAddress
    return address



address =_initialize_NPcon(abi,bytecode)
print(address)


### Write abi and address of contract to json file and call it when it
is necessary ###
data = {
         'abi':abi,
         'contract_address': address
        }
#print(data)
with open("data.json", "w", encoding= 'utf8') as outfile:
     json.dump(data, outfile, indent=4, sort_keys=True)
web3interact_contract.py

import  json
from web3 import Web3
from web3.middleware import geth_poa_middleware

#ganache_url = "http://127.0.0.1:7545"

vbox_url= "http://192.168.1.41:22000"

#web3 = Web3(Web3.HTTPProvider(ganache_url))

web3 = Web3(Web3.HTTPProvider(vbox_url))

web3.middleware_onion.inject(geth_poa_middleware, layer=0)
```

```
###### Use the JSON file to retrieve abi and address ######
with open('data.json') as data_json:
    data = json.loads(data_json.read())
    abi = data['abi']
    address = data['contract_address']
#    print(address)


web3.eth.defaultAccount = web3.eth.accounts[0]

contract = web3.eth.contract(address=address, abi=abi)

generateProv = input('Generate account for providers (Y/N): ')


if generateProv == 'Y':
    numProv = int(input('Enter number of providers: '))
    for i in range(numProv):
        web3.parity.personal.unlock_account(web3.eth.defaultAccount, "",
3600)
        web3.parity.personal.new_account("")
        web3.eth.sendTransaction({'from':web3.eth.defaultAccount,
'to':web3.eth.accounts[i], 'value': web3.toWei(100, "ether")})
print(web3.eth.accounts)



addProv = input('Add new NP (Y/N): ')

if addProv == 'Y':
    for i in range(numProv):
        #print(i)
        name = input('Enter Providers Name: ')
        offered_res = int(input('Enter offered resources: '))
        reserved_res = int(input('Enter reserved resources: '))
        cost = int(input('Enter resources cost: '))
        region = input('Enter Region: ')
        sla = int(input('Enter SLA number: '))
        addressNP = web3.eth.accounts[i+1]
        tx_hash = contract.functions.addNetworkProvider(name,
offered_res, reserved_res, cost, region, sla, addressNP).transact()
        web3.eth.waitForTransactionReceipt(tx_hash)


count = int(format(contract.functions.np_count().call()))
#print(count)

NetworkProviderName =list()
NetworkProviderAddresses =list()
NPinfos = list()

for i in range(1,count):
    NPaddress = format(contract.functions.NetProvtoOwner(i).call())
    NPinformation =
format(contract.functions.NetworkProviders(i).call())
    NPinfos.append(contract.functions.NetworkProviders(i).call())
    NPname = contract.functions.NetworkProviders(i).call()
    NetworkProviderName.append(NPname[0])
    NetworkProviderAddresses.append(NPaddress)

    #print('Updated NPs : ', NPaddress)
    #print('Network Provider Info : ', NPinformation)
    #print('Network Provider: ', NPname[0])

#print('List of Names', NetworkProviderName)
#print('List of Addresses', NetworkProviderAddresses)
print('Infos as list',NPinfos)
ProviderToAddress = dict(
zip(NetworkProviderName,NetworkProviderAddresses ))
print(ProviderToAddress)


#request_res =
format(contract.functions.get_request_resources(2).call())
#print('Borrow : ',request_res)


#BestMatch = format(contract.functions.getBestMatch(5).call())

demand_resources = int(input('Enter number of resources needed: '))

BestMatch = contract.functions.getBestMatch(demand_resources).call()




results= BestMatch[1]
#print(results)
_result = results[0]
id = _result -1
print('Result',_result)
if id >0:
    name= NetworkProviderName[id]
    address = NetworkProviderAddresses[id]
    NPinfo = NPinfos[id]
    cost = NPinfo[3]
    print(name, address, cost)
    final_cost = cost * demand_resources
    print(final_cost)


make_transaction=input('Proceed to transaction (Y/N): ')

if make_transaction== 'Y' and id >0:
    prov_req= input('Enter the name of the provider that request
resources: ')
    web3.eth.defaultAccount = ProviderToAddress[prov_req]
    # HasNetProv =
format(contract.functions.HasNetProv(web3.eth.defaultAccount).call())

    print(web3.eth.defaultAccount)
    destination_address = ProviderToAddress[name]
    print(destination_address)
    if web3.eth.defaultAccount != destination_address:
        web3.parity.personal.unlock_account(web3.eth.defaultAccount,"",
3600)
        tx_hash = contract.functions.transaction(_result,
demand_resources).transact({'from':web3.eth.defaultAccount,'value':
web3.toWei(final_cost, 'ether')})
```

```
        web3.eth.waitForTransactionReceipt(tx_hash)
    print('Infos as list',NPinfos)
```

NP_contact_cost_per_resource.sol

```
pragma solidity >=0.4.22 <0.7.0;


contract NPcontract {
// An array is created where information regarding the NPs will be 
stored///////
    NetworkProvider[] public NetworkProviders;

// Ether var
    uint value = 1 ether;

// The counter is used in order to find easier the number of 
participants///////
    uint256 public np_count = 0;
// When an Network Provider is accepted and want to join the network, 
the NP gets an address that gives him the ability to add a Network 
Provider. The NP can execute the  addNetworkProvider function to add 
himself
     address  admin;
//The modifier is used for perfoming the afore mentioned action
    // modifier onlyParticipant(){
    //     require(msg.sender == accepted_participant);
    //     _;
    // }
// The model of the Network Provider is created, that defines the 
information regarding the NP, which is contained in the array
    struct NetworkProvider{
        string name;                   // The name of the NP
        uint offered_resources;        // The amount of resources the 
NP offers to the network
        uint reserved_resources;       // The amount of resources the 
NP has for his own needs
        uint cost;                     // The cost of resources
        string domain;                 // The domain where these 
resources can be deployed
        uint sla;                      // A number that corresponds to 
certain SLA profiles

    }

    // mapping (address => NetworkProvider) public NPs;

    mapping (uint => address payable) public NetProvtoOwner;

    mapping (address => uint) OwnertoNetProv;

    mapping (address => bool) public HasNetProv;

//The constructor is used for perfoming the afore mentioned action 
regarding the ownership of the addNetworkProvider function
    constructor() public {
    admin =msg.sender;

addNetworkProvider("",0,0,1000000000000000000000000000000000000000000000
000000000000000000,"",0,0x55f8AFc0681fd701E2f43D145f71f3594b95eD5B);
```

```
//addNetworkProvider("OTE",10,8,7,"Athens",6,0x14723A09ACff6D2A60DcdF7aA
4AFf308FDDC160C);

//addNetworkProvider("Vodafone",10,5,5,"Athens",5,0x4B0897b0513fdC7C541B
6d9D7E929C4e5364D2dB);
    }

    modifier onlyAdmin(address _address) {
        require (_address== admin);
        _;

    }


    //This functions add Network Provider to the array, with the
predifined information
    function addNetworkProvider(string memory name, uint
offered_resources, uint reserved_resources, uint  cost, string memory
domain, uint  sla, address payable _address) public
onlyAdmin(msg.sender) {
    uint   id=
NetworkProviders.push(NetworkProvider(name,offered_resources,reserved_re
sources,cost,domain,sla)) - 1 ;
        NetProvtoOwner[id] =_address;
        OwnertoNetProv[_address] = id;
        HasNetProv[_address] = true;
        np_count +=1;
    }






// The following contract checks if there is a need for performing a
request for resources



// The output is boolean. True is request for resources should be made
and false if not.
    // function get_request_resources(uint demand_resources, uint i)
public view returns (bool) {
    //     if (NetworkProviders[i].reserved_resources >
demand_resources) {
    //         return  false;
    //     }else {
    //         return  true;
    //     }
    // }
// The output is boolean. True is request for resources should be made
and false if not.
    function get_request_resources(uint demand_resources) public view
returns (bool) {
        if
(NetworkProviders[OwnertoNetProv[msg.sender]].reserved_resources >
demand_resources) {
            return  false;
        }else {
            return  true;
        }
    }


//This function returns the best match when there is  demand of
resources. This function checks every NP that meets the demands and
selects the one (or many) with the minimum cost. ////

    function getBestMatch(uint demand_resources) external view
returns(uint[] memory,uint[] memory, uint, address) {
    uint counter = 0;     //set first counter
    uint counter2 = 0;    // set second counter

    uint[] memory result = new uint[](np_count);   //create an array
(used in storing resources result) in memory that is not stored in the
blockchain. The array is empty it is a uint type and the length is taken
by previous contract and is equal to the number of NPs stored.
    uint[] memory newresult = new uint[](np_count); //create an array
(used in storing cost result) in memory that is not stored in the
blockchain. The array is empty it is a uint type and the length is taken
by previous contract and is equal to the number of NPs stored.
// This loop is executed until we reach the number that denotes the
length of NetworkProviders array.
// If the field of offered resources of each Network Provider is larger
or equal to the given demand_resources the it stores the id of this
element in the result array and increases the counter by 1.
    for (uint i = 0; i < NetworkProviders.length; i++) {
      if (NetworkProviders[i].offered_resources >= demand_resources) {

        result[counter] = i;
        counter++;

      }
    }
// This loop is executed until we reach the number that denotes the
length of results array, constructed previously. Also we set a new empty
uint leastPrice used as a value inside the if.
// The if compares the field cost of each element of the Network
Providers array with the leastPrice and if the values is smaller or
equal to leastPrice or the leastPrice is 0, then the leastPrice is ste
to the value included in the cost field of the specified Network
Provider.
    uint leastPrice;
    for (uint j = 0; j < result.length; j++) {
// The output of this if is the minimum value stored in leastPrice
      if (NetworkProviders[result[j]].cost <= leastPrice || leastPrice
== 0) {


        leastPrice = NetworkProviders[result[j]].cost;
      }
    }
// This loop is executed until we reach the number that denotes the
length of results array, constructed previously.
// The if compares the field cost of each element of the Network
Providers array with the leastPrice computed previously and if it is
identical it stores the id of this network provider to the newresult
array and then increments the counter.
    for (uint k = 0; k < result.length; k++) {
        if (NetworkProviders[result[k]].cost == leastPrice){
            newresult[counter2] = result[k];
            counter2++;
        }
    }
    return (result,newresult, leastPrice, msg.sender);
  }
  function transaction(uint _result, uint _demand_resources) external
payable {
   require(HasNetProv[msg.sender]== true);
   require (NetworkProviders[_result].offered_resources >=
_demand_resources);
   //uint payFee = NetworkProviders[_result].cost ether;
   require(msg.value == NetworkProviders[_result].cost * value *
_demand_resources);
   NetworkProvider  storage NPRequest =
NetworkProviders[OwnertoNetProv[msg.sender]];
   NetworkProvider  storage NPReply = NetworkProviders[_result];
   NPRequest.reserved_resources = NPRequest.reserved_resources +
_demand_resources;
   NPReply.offered_resources = NPReply.offered_resources -
_demand_resources;
   withdraw(NetProvtoOwner[_result]);
   //emit transaction_event//



    }


    function withdraw(address payable _address_rec) internal
returns(bool) {

    //uint payFee = 0.001 ether;
    _address_rec.transfer(address(this).balance);
    return true;
  }

}
```