2021

Using Distributed Ledger Technologies in VANETs to Achieve Trusted Intelligent Transportation Systems

Fares Nabil ElAmine
fnelamine@mix.wvu.edu

Using Distributed Ledger Technologies in VANETs To Achieve Trusted Intelligent Transportation Systems

Fares Nabil El-Amine

Thesis Submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at
West Virginia University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science
in
Electrical Engineering

Roy Nutter, Ph.D., Chair
Powsiri Klinkhachorn, Ph.D.,
Muhammad Choudhry, Ph.D.

Lane Department of Computer Science and Electrical Engineering
Morgantown, West Virginia
2021

Keywords:
Vehicular Ad-Hoc Network, Internet of Things, Intelligent Transportation Systems, Distributed Ledger Technology, Blockchain, Directed Acyclic Graphs

Abstract

Using Distributed Ledger Technologies in VANETs To Achieve a Trusted Intelligent Transportation System

Fares Nabil El-Amine

With the recent advancements in the networking realm of computers as well as achieving real-time communication between devices over the Internet, IoT (Internet of Things) devices have been on the rise; collecting, sharing, and exchanging data with other connected devices or databases online, enabling all sorts of communications and operations without the need for human intervention, oversight, or control. This has caused more computer-based systems to get integrated into the physical world, inching us closer towards developing smart cities.

The automotive industry, alongside other software developers and technology companies have been at the forefront of this advancement towards achieving smart cities. Currently, transportation networks need to be revamped to utilize the massive amounts of data being generated by the public's vehicle's on-board devices, as well as other integrated sensors on public transit systems, local roads, and highways. This will create an interconnected ecosystem that can be leveraged to improve traffic efficiency and reliability. Currently, Vehicular Ad-hoc Networks (VANETs) such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-grid (V2G) communications, all play a major role in supporting road safety, traffic efficiency, and energy savings.

To protect these devices and the networks they form from being targets of cyber-related attacks, this paper presents ideas on how to leverage distributed ledger technologies (DLT) to establish secure communication between vehicles that is decentralized, trustless, and immutable. Incorporating IOTA's protocols, as well as utilizing Ethereum's smart contracts functionality and application concepts with VANETs, all interoperating with Hyperledger's Fabric framework, several novel ideas can be implemented to improve traffic safety and efficiency. Such a modular design also opens the possibility to further investigate use cases of the blockchain and distributed ledger technologies in creating a decentralized intelligent transportation system (ITS).

Acknowledgements

I would first like to thank Dr. Powsiri Klinkhachorn for convincing me to pursue my master’s degree, as well as offering me the opportunity to work as a teaching assistant for the Lane Department of Computer Science and Electrical Engineering. My time here has been nothing but great as I grew and learned more about the topics that I am passionate about. I am also grateful for the time spent with Drs. Mathew Valenti and Mohamed Hefeida in giving me general advice throughout my time working for the department. A special thank you to Dr. Roy Nutter for being my advisor and primary instructor for the last couple of years and offering me his wisdom, guidance, and support with my work; thank you for your continuous encouragement and belief in me. I would also like to thank Drs. Muhammad Choudhry, Powsiri Klinkhachorn, and Roy Nutter for serving as members of my committee. Thanks to my colleagues and friends, who always motivated me to keep pushing through, and never doubted what I could accomplish. Last but certainly not least, a huge thank you to my parents, sisters, and uncles for their patience, motivation, and continuous support. I would not be where I am today without them, and I will forever be grateful for that.

Table of Contents

List of Figures

Chapter 1 – Introduction

1.1 Motivation

Computing has helped automate many tasks by digitizing the information it gathers from the physical world. Because of that, our environment has been rapidly shifting into an all-encompassing digitalized one. Digitization has caused almost everything that we use or interact through to be embedded with or operated by some sort of an electronic device. Recently, these devices have been slowly getting integrated into the cyberspace. This makes real-time communication achievable with almost any other device. This also allows Internet of Things (IoT) devices to start collecting, sharing, and exchanging data with others to enable all sorts of operations and services which aim to make our life a little bit easier. This instead shifts responsibility and oversight from humans towards the electronics and the networks connecting them.

During the last couple of decades, vehicle manufacturers have been increasingly adopting more digital automotive technologies into their products and digitizing most parts of a vehicle. This will eventually allow them to offer services such as lane-keeping, parking assistance, adaptive cruise control (ACC), and many others. In addition, vehicles are now also equipped with many connectivity technologies such as Wi-Fi, Bluetooth, and cellular; enabling them to communicate with a wider range of devices located both in other vehicles, and in the public infrastructure [7]. This sort of technology represents the general class of cyber-physical systems that will enable us to enter the era of developing so called “smart cities”. One notable and worthwhile feature of smart cities would be to establish and develop intelligent transportation systems through smart traffic & information management, smart governance, as well as smart grid and energy distribution. This opens the possibility for greater benefits such as making driving easier and safer by giving drivers (and their vehicles) insight towards the state of traffic and the respective automotive responses of others, as well as providing the ability to “self-heal” in response to scheduled and unscheduled events. It also enables the possibility of more effective usage and exploitation of the supporting infrastructure while possibly maximizing the efficiency of the energy (gas and/or electric) utilized while operating.

This has been a motivating factor for the automotive industry, alongside other software developers and technology companies. They are beginning to put in a lot of effort and research into continuously enhancing the efficiency and security of Vehicular Ad-hoc Networks (VANETs). VANETs in this case would correspond to the dynamic network of vehicles equipped with some types of IoT devices that are actively deployed on the road itself. One of the key requirements of a VANET is to have efficient wireless intra- and inter-vehicle communication mechanisms to collect and exchange the data along with other drivers, vehicles, and road-side units (RSU) [1]. The goal of deploying VANETs would be to eventually attain the most sustainable form of mobility with the least amount of energy and time spent on the road, i.e., sustained mobility.

The most traditional architecture of a VANET can be described as a network of wireless, self-organizing, autonomous mobile devices/nodes that act as both a sender and a receiver as seen in Figure 1 below. Most of the communication between the nodes would be identified as either a data transfer between vehicles i.e., Vehicle-to-Vehicle (V2V), or between vehicles and the infrastructure that the RSU’s are part of i.e., Vehicle-to-Infrastructure (V2I) [2]. This is all accomplished with the help of intelligent coordination of information within the transporting networks themselves, alongside other more centralized resources located within the RSU’s. V2V, V2I, and Vehicle-to-Grid (V2G) communication all play a major role in supporting road safety, traffic efficiency, and energy savings. An example of using V2V communication to support traffic efficiency would be through providing an adaptive communication medium that allows vehicles to broadcast and/or share information with the network so that other vehicles can rely on this information to make more informed decisions with security being of upmost importance.

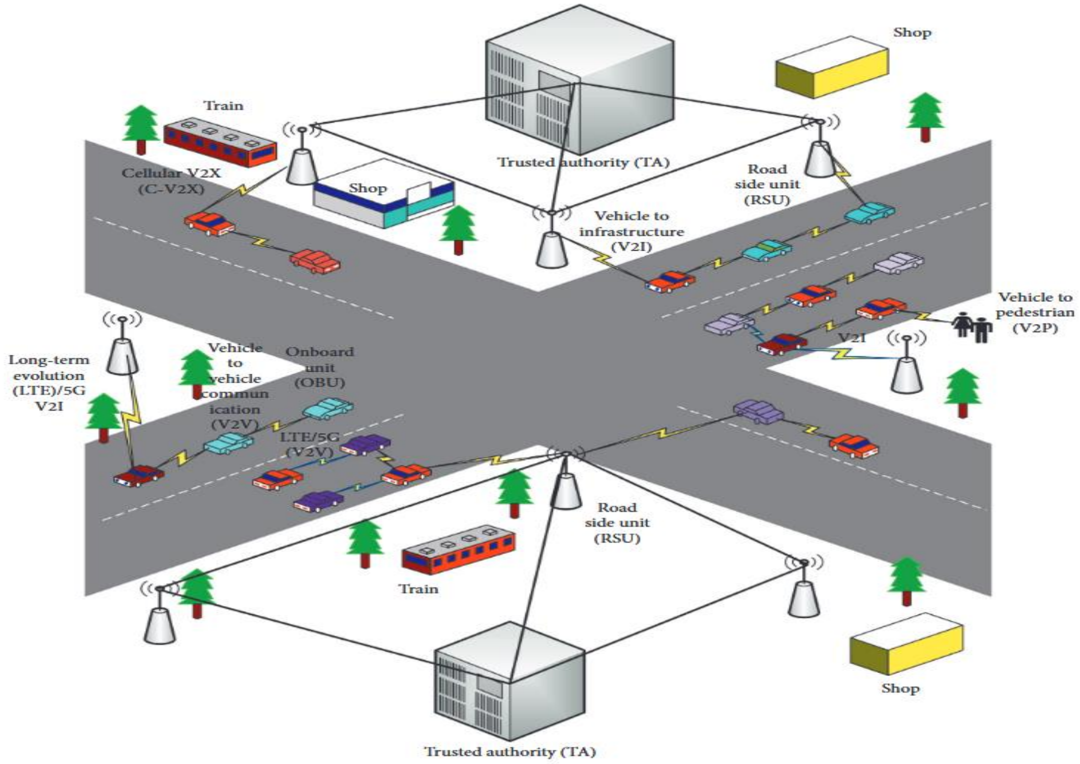


Figure 1 VANET Model [50]

1.2 Background

Developing a heterogenous network architecture that enables wire-like speeds, as well as being robust, seamless, and secure is not an easy task [1]. Several different methods have been proposed and used over the years ([8], [9], [10], [11]), but the common theme behind them seems to be that they all rely on an architectural topology that enables conventional access control, consisting of some centralized authority/node designated to monitor the entire system, distribute the messages, and authenticate the sources. This framework is prone to load concentration and single points of failure and some potentially serious drawbacks though; with the increasing number of vehicles equipped with sensors and wireless connectivity abilities, the load on centralized systems and/or nodes can get overwhelming thus slowing down the response rate [3]. In addition to that, messages in such a network architecture are prone to several security threats such as Sybil, Wormhole, and Illusion attacks [4].

Reliability of the information relayed on such networks is of upmost importance for safety reasons, this means that establishing effective trust management protocols is a must. From a research perspective, to help intelligent transportation systems (ITS) in maintaining an overall stable, effective, and profitable networking environment, there is a dire need to create a secure, trusted, and decentralized architecture to allow smooth interoperability and communication through reliable flow of data among any of the network participants [5] shown in Figure 2. Although distributed access control schemes that utilize blockchain technology have been proposed before to try and tackle these issues, they inherit the underlying drawbacks of it, such as high transaction fee and low throughput

[12]. Thus, the goal would be to build some framework that can create, disseminate, and secure messages that are communicated across vehicles, IoT's, RSU's, and other devices in a fashion that eliminates the risk of compromising any of the nodes' personal information, behavior, or control [3], with a high degree of throughput and cheap transactional fees in a modularized fashion.

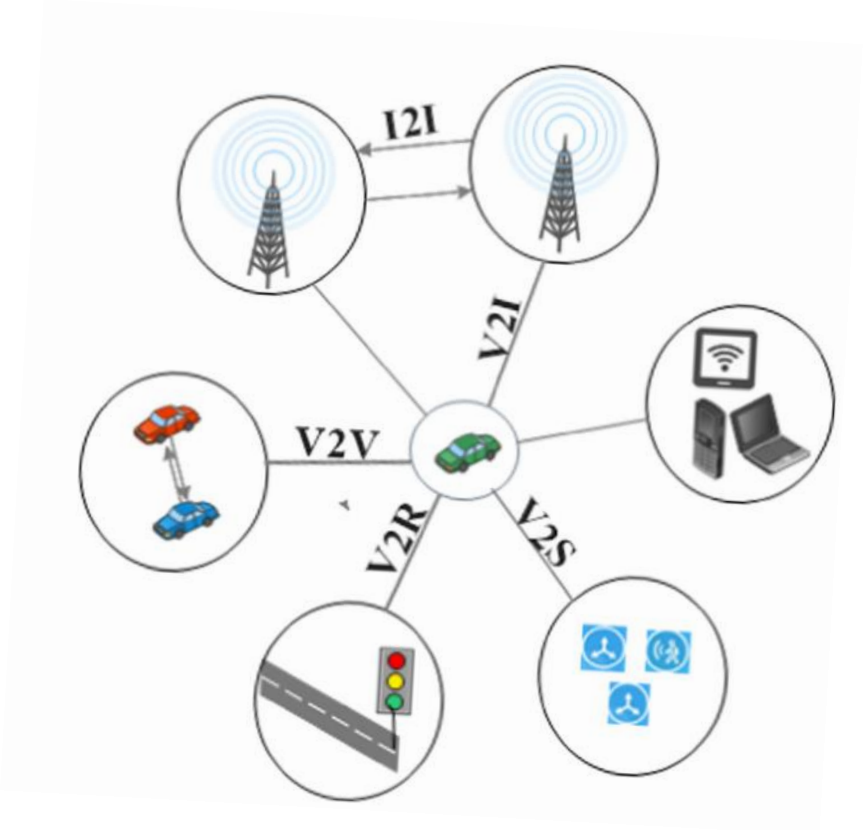


Figure 2 VANET Participants [52]

1.3 Problem Statement

To mitigate the risks from centralized authorities of classical network topologies, in this paper we propose developing a transportation-based network model that has a modular architectural topology. This hybrid design consists of several independent networks that rely on distributed ledger technologies based on blockchain and directed acyclic graphs with cross communication capabilities. The motivation behind this design is to make the transportation system decentralized, while still having a form of governance and access control for the whole ecosystem against potential abusers and attackers. After describing the design of the model and showing the feasibility of it in the real world by referencing other related publications, the experimental setup of the different networks is laid out for future simulations and further development.

1.4 Outline

The rest of the paper will be split up into four chapters. Chapter 2 defines Vehicular Ad-hoc Networks (VANETs). This includes background information needed regarding the various distributed ledger technologies and their properties (Blockchain, Directed Acyclic Graphs), as well as the respective projects (Ethereum, IOTA, and Hyperledger Fabric). These projects will be used to enhance security and efficiency of the network, while also adding an extra layer of functionality and customizability to what can be leveraged through them. Chapter 3 describes previously published, but related studies and implementations around improving the security of mobile networks and intelligent transportation systems. This is illustrated through examining works done on designing new protocols and architectures that also leverage decentralization through blockchain-based solutions.

Chapter 4 details the proposed ideas in terms of system overview and design. It also explains how the model can further improve existing network topologies. In addition, it details the steps taken to set up the basis of the simulation. Chapter 5 analyzes the feasibility of such a model, while also pointing out what can be improved upon further in future works by referencing several ongoing, publicly available development projects. This chapter also marks the end of the paper by laying out a few concluding remarks.

Chapter 2 – Background Information

2.1.1 VANETs: Overview

With the rapid increase in our societies' mobility, Information and Communication Technology (ICT) solutions have provided us with a much greater Internet connectivity on the move, most notably in vehicles. One such wireless connected-vehicular technology that has been designed to continuously work on improving the efficiency and safety of today's transportation network is through designing and setting up vehicular ad-hoc networks (VANETs).

VANETs are nothing more than an interpretation of wireless, or mobile ad-hoc networks (W/MANET). These types of networks are by design a type of decentralized network, analogous to peer-to-peer (P2P) self-forming networks in the sense that they do not rely on any sort of pre-existing infrastructure, central authority, or controller to keep the network maintained and stable. Routers and access point functionalities are taken care of by having every participating node in the network to also have a duty to forward packets to other participating nodes. One of the main reasons why VANETs developed to what they are, is due to the nature of how vehicles operate; they are mobile, independent, and free to move in any direction, dynamically changing the network's layout at any given moment. As such, VANETs can be characterized as being a dynamic topology with intermittent connectivity capability while still following some mobility patterns due to the predictable nature of how vehicles move around and where to expect them to be operating in.

Since VANETs provide a communication medium between neighboring vehicles, the architecture of such a network can be split up into three different domains:

1) Mobile Domain: Comprised of the vehicle domain (cars, buses, trucks, etc.) as well as portable devices such as smartphones and laptops within the vehicles.
2) Infrastructure Domain: This domain includes all units that are stationary (traffic lights, variable-message signs, toll booths, and cellular towers) with predictable locations.
3) Generic Domain: Consists of the Internet's public and private infrastructures that connect the different nodes and servers alongside other computing resources.

The dynamics of such a network can then be easily described. First, the mobile domains produce data that is leveraged and transmitted to the infrastructure domain, which processes the data and does its own modulation on them before communicating the results back to the generic domain. The generic domain then broadcasts them back to the mobile domain for insight on road and traffic conditions that can be eventually used by the devices inter-connected on a vehicle to make better decision-making actions.

2.1.2 VANETs: Data Transmission & Security

Currently, data exchange in a VANET happens with the help of routing protocols. Several types of routing protocols have been developed or proposed since 2005 [15], categorized into topology-based and position-based. All these protocols share a common theme in which nodes must cast their messages to the wireless medium to be received by others via unicast or broadcast type of dissemination. Unicast is designed to exchange data from one source to another either via a single-hop or multi-hop transmission(s), whereas broadcasting protocols are meant to deliver the data packets to as many nodes as possible at a time. The common goal between these two types of data casting is to utilize maximum bandwidth and reachability which is attainable through probability, area, and neighbor knowledge-based techniques.

Security of such a network is important due to how sensitive the information being disseminated is. VANET's communication can be either established over IEEE's 802.11p standard (extension of 802.11-WLAN's implementation to support Wireless Access for Vehicular Environments i.e., WAVE) by operating on the Dedicated Short-Range Communication (DSRC) band [15], or cellular such as WiMAX or LTE. Since VANETs are a type of an open network, any node that meets the minimum requirements can join. From a security perspective, this is critical as there is no defined mechanism that ensures the trustworthy nature of all nodes. However, it is highly important to still have some security measure in place to protect the network from any form of malicious manipulation that might cause harm to the environment where these networks exist, such as automobile accidents. Methods of attacking a network are plentiful, but they can all be categorized into three main groups. Attacks can either threaten the authenticity of a network’s activity, leak out confidential information, or disrupt the availability of its resources [13]. Most traditional ITS communication protocols focus on increasing the efficiency of communication to minimize disruption, packet-loss, and delay propagation while mitigating security risks by setting up centralized authorities responsible for the processing of the data and managing the trustfulness of the data being routed. The key challenge in providing secure communication channels in VANETs relates back to creating a robust system that can keep nodes’ messages authenticated and tamper-proof, while still being able to offer a reliable and scalable model. This is where decentralized models that are built using blockchain technology and graph theory come into play.

2.2 Decentralized Network Architecture

When a network topology relies on some central authority or node to maintain a stable and secure system, participants in that network must inherently trust that the central node is genuine and will not act maliciously. However, this presumption has already been proven to not always hold true. In organizations, personal data stored in some centralized server is susceptible to getting hacked and eventually leaked, and when that happens, it breaks the trust given to them in preserving the data’s confidentiality. Same can be even said regarding financial markets and governments that fail to provide economic security, peace, or freedom. Networks in the sense of governance and trust are no different.

In the world of computers, participants could interact with a large set of other participants to acquire certain services. These interactions are efficiently supported through overlaying network-topology maintenance protocols and message routing protocols [15]. In terms of securing such networks, trust must be shifted away from common authoritative third parties and instead establish a direct trust between all participants in a decentralized manner.

2.2.1 Blockchain Technology

Trusted and governed networks are not new but so far, with the current architectural frameworks deployed, they have all relied on some level of trust in human agents or centralized servers. These networks are also organized by means of the usual system of implicit and explicit rules that control the decision-making process. This is where decentralized trust management systems have shown to help in securing networks, VANETs included [19]. In a decentralized trust management system, data and tasks would be stored and evaluated in the vehicle itself or within RSU's. Naturally though, with such a highly dynamic and decentralized system, vehicles constitute separate entities and are considered strangers with no easy way to establish trust with each other and everyone else [18]. To overcome the problems of a centralized model as well as the trustfulness between the nodes, blockchain technology might be the tool that eventually solves this problem.

Technically speaking, blockchain as a data structure is a form of a decentralized shared ledger which is based on chronologically time-stamped blocks that are encrypted with ways of verifying and synchronizing the chained blocks, as well as the data within them in a peer-to-peer (P2P) network as shown in Figure 3 below. A blockchain can be thought of as a decentralized public database or ledger that keeps a permanent record of digital transactions or messages, with no way of reversing any record already published within it, making it tamper-proof. All participants of this network also have a full copy of the ledger for verifiability and transparency. Reaching consensus on what gets recorded, as well as establishing a trustful cooperation of nodes in a network without the need for a trusted central authority or a third party can now be further investigated for making this technology as the backbone of a VANET’s infrastructure.

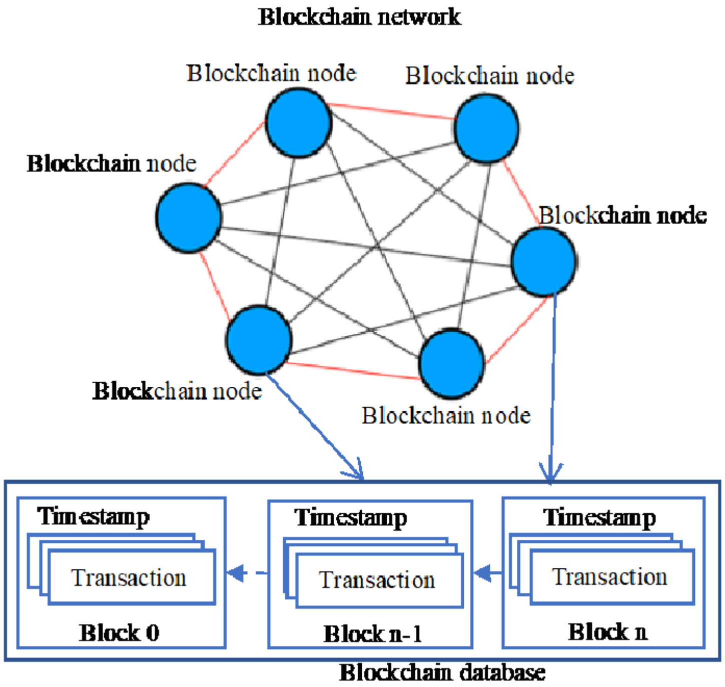


Figure 3 Blockchain Network Structure
© *2018 IEEE Blockchain Simplified Structure [51]*

2.2.1.1 Bitcoin

The term blockchain became popularized after Nakamoto released a whitepaper [20] that describes a P2P electronic cash system, Bitcoin, that fundamentally works on utilizing blocks of data chained together using hash functions to track the minting and transfer of tokens. The main motivation behind it was to establish a decentralized payment network without the inherent risks of trust-based models such as the traditional banking system. In a blockchain-based networking architecture, the nodes all work at once with little coordination while still offering the ability to leave and rejoin the network at will, which makes it suitable for VANET-like topologies.

For transactions to be uploaded onto the ledger, a process of validation must first take place. The validation process is achieved through a consensus algorithm, proving that the data published to the ledger is true and consistent. This can be implemented in several different ways; one notable example would be through Proof of Work (PoW). PoW works by making nodes complete a certain amount of work to validate a new block. This occurs by solving a computationally expensive function, brute-forcing calculations to find a particular hash. This system is dynamic as the difficulty of the function can be varied by changing the requirements of the answer (number of 0’s that the hash begins with). PoW can severely mitigate threats of malicious actors who wish to inject malicious data into the network due to the need to control at least 51% of all participating nodes to eventually approve the malicious data, which can get quite expensive [21].

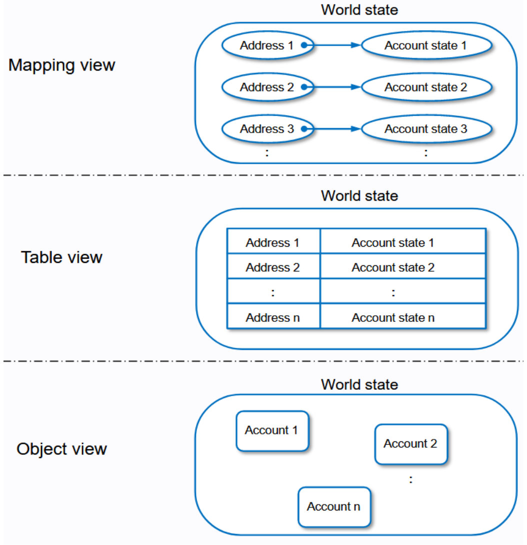
Note that most of the underlying ideas that Bitcoin runs on are not new. Nakamoto was the first that imagined and established a system that utilizes several ideas together within one project that has been functioning as intended, while still being fault-tolerant for the last 12 years as of writing this paper. David Lee Chaum is considered by many to be the grandfather of blockchain as his PhD Dissertation in June 1982 titled “Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups” [21] laid most of the foundational work. It was not until later when Stuart Haber and W. Scott Stornetta came up with the concept of digital timestamping [23] which made altering dates on digital messages infeasible, and later when Hal Finney described “Reusable Proofs of Work” system in 2004 [24] were established that blockchain was introduced to the world in 2009.

Utilizing blockchain as it was originally designed to function does not meet our proposed networking infrastructure requirements quite yet though. Since it was first introduced as a decentralized monetary tool, there is little emphasis on identifying nodes, making it hard to track down different vehicles, RSU’s, and other IoT devices that are part of this network. Another reason why such a network implementation cannot be directly enforced for VANETs is that the underlying protocol uses a scripting system that is not Turing complete, restricting the possibility to develop further use-cases that enforce some computational tasks besides securing digital transfers of tokens.

2.2.1.2 Ethereum

After Bitcoin's inception, the cyberpunk and crypto-enthusiast community became to unfold, and several forums started emerging to further analyze and discuss the underlying protocol. One of the early adopters was Vitalik Buterin, a 17-year-old Russian Canadian programmer that eventually went on to co-found "Bitcoin Magazine" after becoming fascinated by the blockchain technology. Realizing the limitations of Bitcoin's protocol in terms of Turing-completeness as mentioned before, Buterin spent three years developing his own platform alongside several others and eventually released it on July 30th, 2015, under the name Ethereum.

The purpose of Ethereum was to produce a blockchain network that has a built-in, fully fledged Turing-complete programming language with the capability of creating "contracts" to encode arbitrary state transition functions [25]. Buterin's vision for Ethereum was to create a neutral and open access infrastructure where different applications can efficiently interact with each other through the abstract foundational layer that it is. Instead of comparing its structure to just a distributed ledger, Ethereum acts more like a distributed message-based state machine [26] as shown in Figures 4 and 5. At any time, Ethereum can only have one world state that represents a snapshot of all participants' account's state, with each new state being linked to the previous state as shown in Figure 6. An account's state changes whenever new transactions calling it are issued, waiting to be uploaded onto the network.



Address 1	Account state 1
Address 2	Account state 2
:	:
Address n	Account state n

Figure 4

World State Contents [53]

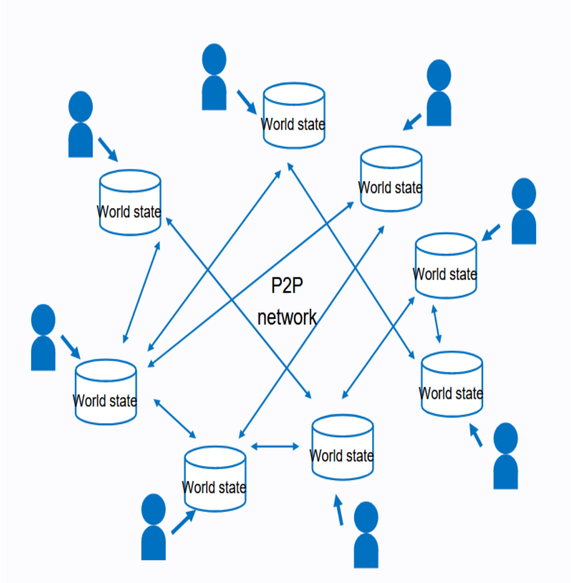


Figure 5

Ethereum: Distributed State Machine [53]

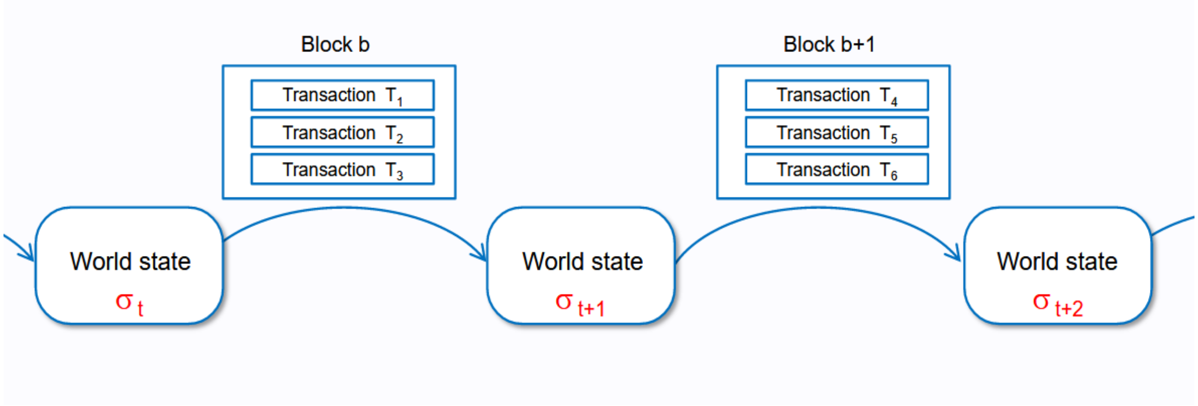


Figure 6 Blockchain Structure Mapping Ethereum World States [53]

Illustrated in Figure 7, Ethereum accounts can be of two forms: externally owned accounts (EOA) or contract accounts (CA). The address, balance, storage, and code are what constitutes an account’s state. An EOA consists only of an address and a balance (analogous to a wallet) that are controlled by an external actor, whereas contract accounts also include a storage and code hash that are used to execute whatever logic the deployer/developer made it to do. One thing to note about CA's though is that they cannot be controlled externally once deployed to the network – code is law.

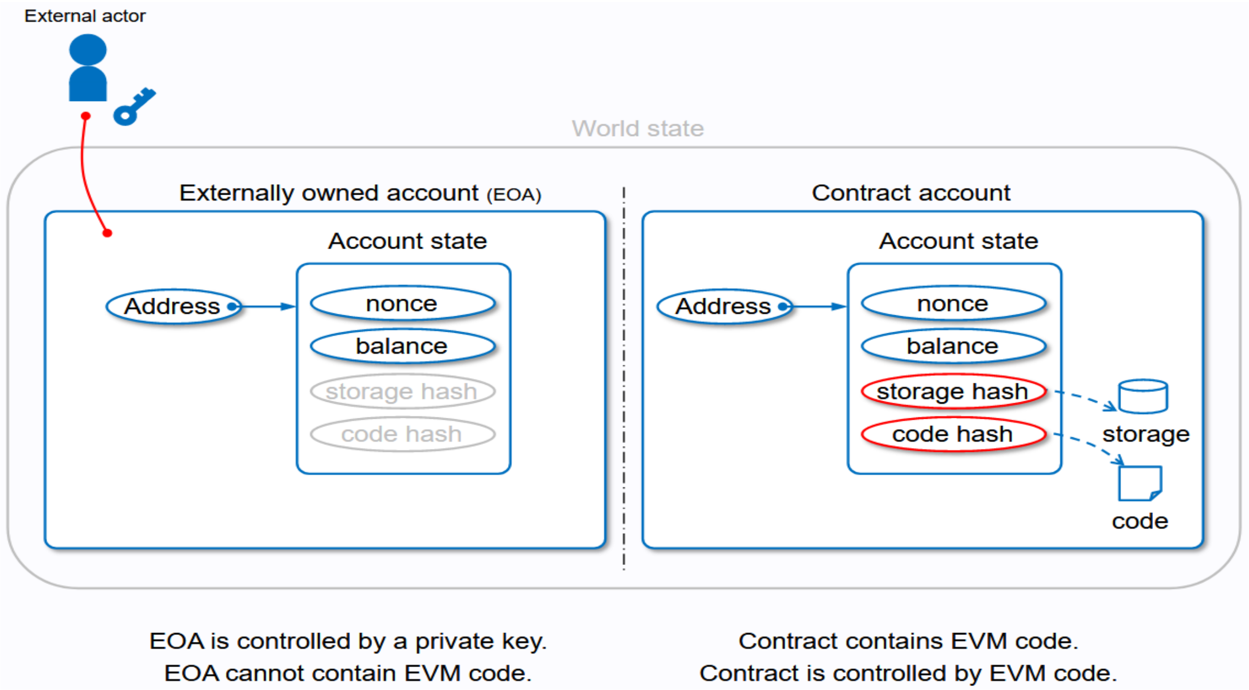


Figure 7 Account State Details [53]

Accounts interact with the network by issuing transactions that specify whether it is a message call to another account or meant for creating a new CA. This is how Ethereum was built, it works by allowing anyone who holds an Ethereum account to issue a transaction, and if the transaction's recipient is a specific address, the network interprets it as a request to create a new CA that holds a smart contract. A more in-depth explanation of the rulesets that describe how the global state transitions is explained in Ethereum's developer's documents under "Ethereum Virtual Machine" [54]. This allows users to develop and deploy decentralized applications with their own rules leveraged by the inherent components of this network compared to Bitcoin's, such as value-awareness, state, and Turing-completeness. Consequently, CA's can thus be labelled as autonomous, program-driven agents that can be trusted without any central authority since the code that drives it is public (copy of the world state is shared with all Ethereum participants in a P2P manner as shown previously in Figure 5) and no one can modify it once the CA has been deployed on the network, or else it will be seen as malicious data.

Data published on the network will always affect at least one of the accounts, so any inconsistency within will affect the data structure that describes Ethereum's world state through a modified Merkle Patricia Trie (MPT). An MPT is a data structure that maps key-value pairs and are used as paths to verify data integrity by checking the tree's root hash. The root node of this structure is cryptographically dependent on all the internal data within it (similar to how Merkle trees work), and as such the root's hash can be used as a secure identity to verify the whole system's state. This is considered secure, verifiable, and immutable since the root hash of the world state is included in each block (list of transactions), and the computational work needed to include a block (PoW) is the proof for verifying and trusting it. With all of Ethereum's complexity, [55] managed to elegantly describe its mechanism in one illustration shown in Figure 8 at the end of this section.

With such a robust framework available, Ethereum seems to be a much more suitable framework to use compared to Bitcoin's for deploying new and upgrading old VANET infrastructures. The fact that one can type high level code using Solidity, which is like JavaScript but designed with blockchain in mind and specifically for Ethereum, it enables a safe, open, autonomously governed, trustable, and reliable mechanism to build any rulesets on top of. An example of how smart contracts can be utilized in VANETs include a variety of transport-related payments such as toll, parking, car insurance, and vehicle registration [27].

Although Ethereum has proven itself to be a robust system, its design is not well-suited to be the basis protocol for all the underlying foundations of how VANETs are designed and the services they offer. One of the main drawbacks of it is that sending transactions on the network and the processing power required for clients to validate new transactions is not cheap nor fast enough. Ethereum also faces scalability issues as do all blockchain-based networks since blocks have to be sequentially linked. This can cause overload when there is an excess of messages being transmitted, increasing the load on the system and slowing it down to the point of not being suitable for rapid message dissemination on the roads. Although Ethereum's innovations might still be useful for VANETs in some situations, scalability and cost still need to be addressed. One proposed way to do so would be through rethinking how transactions and messages get validated and stored onto the network with the removal of mining nodes, making it far more accessible to low powered and mobile devices to also issue transactions onto the ledger.

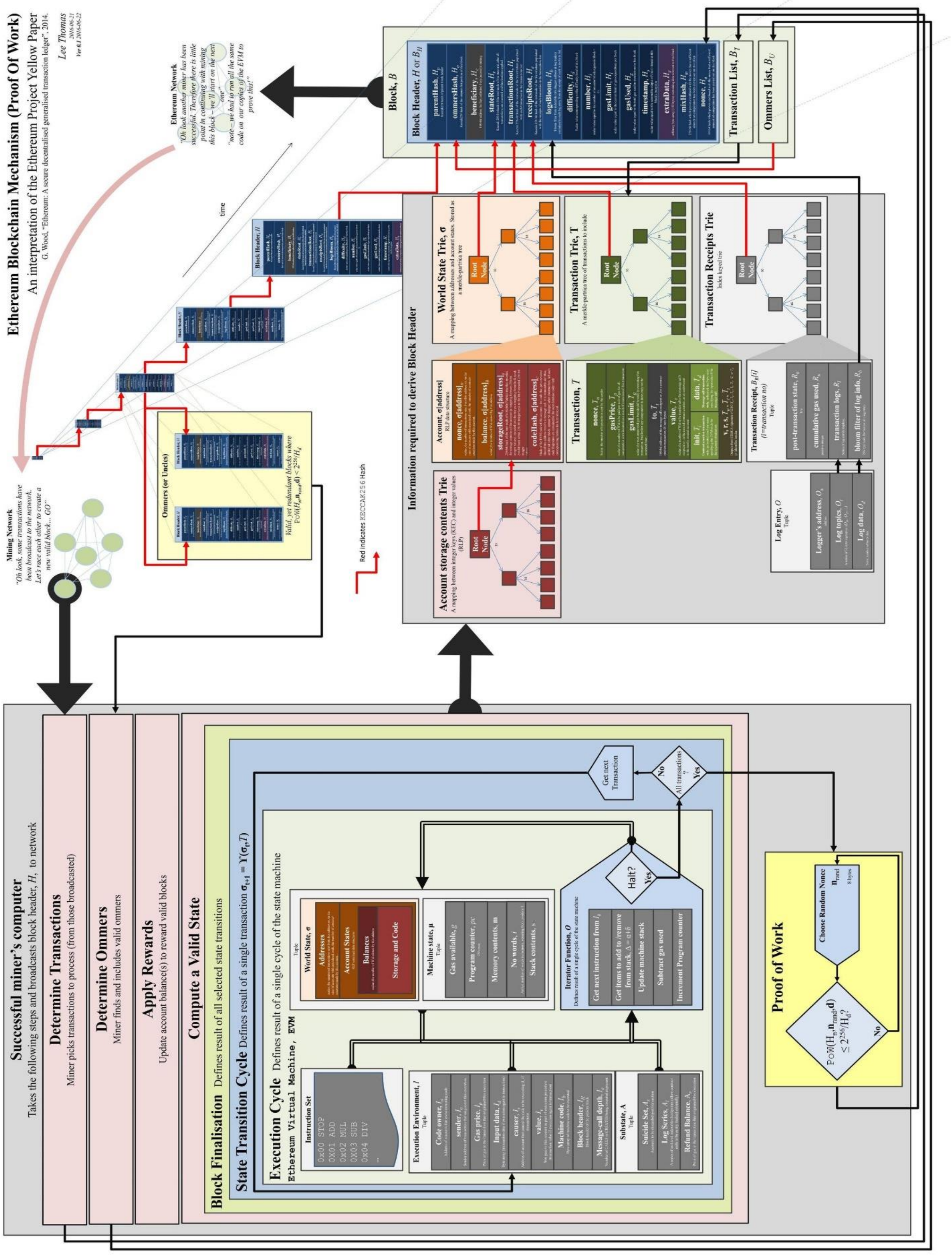

Ethereum Blockchain Mechanism (Proof Of Work)
An interpretation of the Ethereum Project Yellow Paper
G. Wood, "Ethereum: A secure decentralised generalised transaction ledger", 2014.
Lee Thomas
2016-06-21
Ver 0.1 2016-06-22
Mining Network
"Oh look, some transactions have been broadcast to the network. Let's race each other to create a new valid block... GO"
Ethereum Network
"Oh look another miner has been successful. Therefore there is little point in continuing with mining this block – we'll start on the next one"
"note – we had to run all the same code on our copies of the EVM to prove this!"
time
Block, B
Block Header, H or B_H
parentHash, H_p
ommersHash, H_o
beneficiary, H_c
stateRoot, H_r
transactionsRoot, H_t
receiptsRoot, H_e
logsBloom, H_b
difficulty, H_d
number, H_i
gasLimit, H_l
gasUsed, H_g
timestamp, H_s
extraData, H_x
mixHash, H_m
nonce, H_n
Transaction List, B_T
Ommers List, B_U
Ommers (or Uncles)
Valid, yet redundant blocks where PoW(H_n,n_rand,d) < 2^256/H_d
Red indicates KECCAK256 Hash
Information required to derive Block Header
World State Trie, σ
A mapping between addresses and account states. Stored as a merkle-patricia tree
Transaction Trie, T
A merkle-patricia tree of transactions to include
Transaction Receipts Trie
Index keyed trie
Root Node
Account storage contents Trie
A mapping between integer keys (KEC) and integer values (RLP)
Account, σ[address]
RLP data structure
nonce, σ[address]_n
balance, σ[address]_b
storageRoot, σ[address]_s
codeHash, σ[address]_c
Transaction, T
nonce, T_n
gasPrice, T_p
gasLimit, T_g
to, T_t
value, T_v
init, T_i
data, T_d
v, r, s, T_w, T_r, T_s
Transaction Receipt, B_R[i]
(i=transaction no)
Tuple
post-transaction state, R_σ
cumulative gas used, R_u
transaction logs, R_l
bloom filter of log info, R_b
Log Entry, O
Tuple
Logger's address, O_a
Log topics, O_t
Log data, O_d
Successful miner's computer
Takes the following steps and broadcasts block header, H, to network
Determine Transactions
Miner picks transactions to process (from those broadcasted)
Determine Ommers
Miner finds and includes valid ommers
Apply Rewards
Update account balance(s) to reward valid blocks
Compute a Valid State
Block Finalisation Defines result of all selected state transitions
State Transition Cycle Defines result of a single transaction σ_{t+1} = Υ(σ_t,T)
Execution Cycle Defines result of a single cycle of the state machine
Ethereum Virtual Machine, EVM
Instruction Set
0x00 STOP
0x01 ADD
0x02 MUL
0x03 SUB
0x04 DIV
...
Execution Environment, I
Code owner, I_a
sender, I_o
Gas price, I_p
Input data, I_d
causer, I_s
value, I_v
Machine code, I_b
Block header, I_H
Message-call depth, I_e
Substate, A
Tuple
Suicide Set, A_s
Log Series, A_l
Refund Balance, A_r
World State, σ
Tuple
Addresses
Account States
Balances
Storage and Code
Machine state, μ
Tuple
Gas available, g
Program counter, pc
Memory contents, m
No words, i
Stack contents, s
Iterator Function, O
Defines result of a single cycle of the state machine
Get next instruction from I_b
Get items to add to / remove from stack, Δ = α–δ
Update machine stack
Subtract gas used
Increment Program counter
Halt?
Yes
Get next Transaction
All transactions ?
No
Yes
Proof of Work
Choose Random Nonce
n_rand
8 bytes
PoW(H_n,n_rand,d) ≤ 2^256/H_d?
No

Figure 8 Ethereum's Yellow Paper Interpretation [55]

2.2.1.3 Hyperledger Fabric

Blockchain-based networks can have different architectures and use cases, but so far it sounds like any individual with the proper tools and resources can interact freely with these types of networks. As the popularity of Bitcoin, Ethereum and other networks grew, special interest started to emerge in applying this technology to enterprise use cases. The problem with many enterprises is that they might require certain characteristics that the public networks cannot fulfil such as controlled identity management, low latency, high throughput, as well as privacy and confidentiality of specific transactions and data pertaining to their businesses. This resulted in several different projects to design a new type of blockchain-based network that is permissioned, where approval must be granted to any participant prior to accessing or publishing data onto the ledger. Instead of having participants be anonymous as is the case in classic public permissionless networks, the participants in a permissioned network are known to each other which reduces the overhead required in the consensus process to establish trust.

One of the most mature projects nowadays in the space of permissioned networks is a framework developed under the Hyperledger project known as Hyperledger Fabric (HLF). HLF is an open source, enterprise-grade permissioned DLT platform established under the Linux Foundation in 2015. It was designed with a modular and extensible architecture in mind that delivers high degrees of confidentiality, resiliency, and scalability. Interpreted from [58], the main components that constitute this framework are the distributed ledger, organizations, peers, and chaincodes which are more widely known as smart contracts. Organizations are represented by a group of peers that hold instances of the ledger as well as the chaincodes, who enroll through a Membership Service Provider (MSP) that acts as the gateway for accepting new peers into the network, enforcing the idea that HLF is a permissioned network. The MSP maintains identities of all peers within the system and is responsible for issuing peer credentials that are used for authentication and authorization. The MSP handles standard public-key infrastructure (PKI) methods based on digital signatures and has the ability to accommodate commercial certificate authorities (CA) as well [57]. HLF can have more than one organization, so separate MSPs must be present for each one.

Since HLF supports private transactions within the network between specific organizations or peers, channels add an extra layer of privacy between the different participants, with the ability to establish sub-networks within the permissioned network. Every member or peer associated to a specific channel has visibility over the particular transactions occurring within, enabling a form of privacy and confidentiality layer against other members outside the channel, but still within the network. Since HLF relies on a modular architecture, transaction processing workflow is separated into three stages: chaincode (smart contract) invocation via proposal submission, transaction ordering, and transaction validation/commitment. To achieve this, participants within the HLF framework fall into three distinct roles labelled as endorsers, committers (ordering service), and consenters. Messages across this network can either be a type of signature or read/write instructions through a chain code. When a transaction is proposed, it is submitted to the endorser peers, which after a sufficient number of endorses approve the messages, they are sent to the committers who construct the blocks. To ensure that transactions are valid, transactions created using smart contracts typically need to be signed by multiple organizations to be committed to the channel ledger. Multiple signatures are integral to the trust model of HLF. Requiring multiple endorsements for a transaction prevents one organization on a channel from tampering with the ledger or using business logic that was not agreed to. To sign a transaction, each organization needs to invoke and execute the smart contract on their peer, which then signs the output of the transaction. If the output is consistent and has been signed by enough organizations, the transaction can be committed to the ledger. The committers then validate that the predefined policies (certain number of endorsing peers validated the message) were followed without any conflicting transactions, and then finally send them to the consenters who commit them to the ledger. Figure 9 shows an overview of the framework's components.

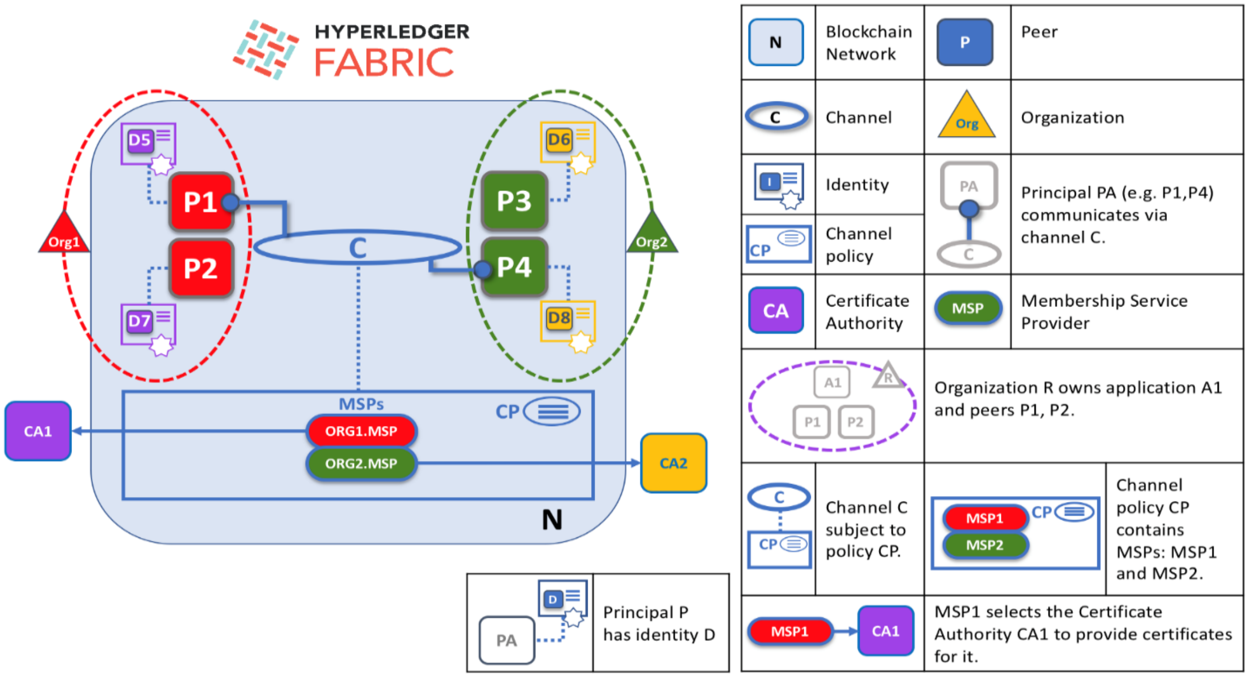

HYPERLEDGER
FABRIC
D5
P1
P2
D7
Org1
C
P3
P4
D6
D8
Org2
MSPs
CP
ORG1.MSP
ORG2.MSP
CA1
CA2
N
PA
D
Principal P has identity D
N
Blockchain Network
P
Peer
C
Channel
Org
Organization
I
Identity
CP
Channel policy
PA
C
Principal PA (e.g. P1,P4) communicates via channel C.
CA
Certificate Authority
MSP
Membership Service Provider
A1
R
P1
P2
Organization R owns application A1 and peers P1, P2.
C
CP
Channel C subject to policy CP.
MSP1
MSP2
CP
Channel policy CP contains MSPs: MSP1 and MSP2.
MSP1
CA1
MSP1 selects the Certificate Authority CA1 to provide certificates for it.

Figure 9 HLF Network Layout [58]

2.2.2 Directed Acyclic Graphs

Graphs represent a network between a finite set of points or vertices, together with a finite set of pairs linking the vertices together called edges. One specific type of such a network is the Directed Acyclic Graph (DAG). DAGs are characterized by edges that are directional only from one vertex to another, with no path that connects any vertex back to itself. From a conceptual level, both blockchains and DAGs have some similarities. They can both be described as distributed ledger technologies to obtain a consensus of the total system, but their underlying mechanism is what changes their scaling properties alongside some potential use cases. A DAG-based architectural networking allows to bypass some of the drawbacks of standard blockchain architectures including scalability and division of users into different groups such as users and validators within a network. This is further illustrated through examining another decentralized network that seems to fit our model for a VANET framework called the Tangle, which resides within the IOTA network.

2.2.2.1 The Tangle (IOTA)

IOTA [28] is one of many projects that aims to build on top of what Ethereum had created, but IOTA's main goal was to establish a new network optimized for IoT devices with micro-transactions in mind. As mentioned before, IoT devices will need to have a frictionless environment that is cheap for issuing out transactions and messages without sacrificing speed and security. To establish that, IOTA had to redesign the infrastructure to be as lightweight as possible to accommodate all forms of devices trying to participate in this network including RSU's, vehicles, and mobile devices on the go. This meant diverging away from the blockchain perspective while keeping the principal ideas behind it as a form of a distributed consensus mechanism.

Motivated by Lerner's project DagCoin [32], a group of researchers and developers designed the Tangle as a natural successor to the blockchain in terms of evolving it to meet the increased demand of its usage in a decentralized network. The Tangle as seen in Figure 10 below, represents a growing ordered set of messages that are cryptographically linked and replicated to all nodes in a P2P network. The Tangle is meant to work as the backbone data structure used by the IOTA network to store transactions, tailored for establishing a machine-to-machine micropayment system [28]. The Tangle is just the name of the DAG based transaction settlement and data integrity layer of IOTA's network with scalability being a major factor into establishing the framework. Two of the drawbacks of blockchain that the Tangle works on solving are the necessity for all participating nodes to reach a common consensus before releasing a new block, as well as the heterogenous nature of the system for having two different types of participants within it (users and miners). Reaching consensus between the huge number of devices interconnected within an ITS would drastically affect the throughput of the entire system.

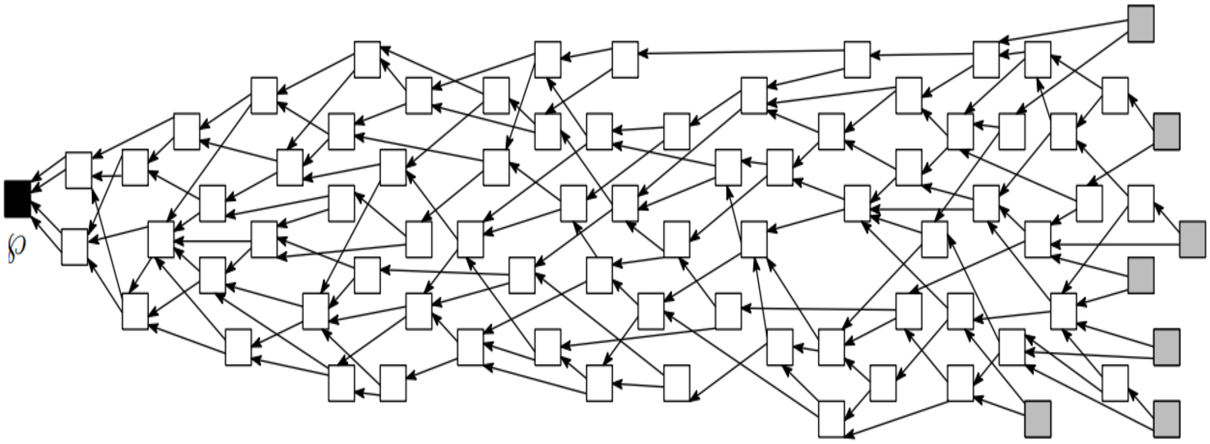


Figure 10 The Tangle: A DAG-based Distributed Ledger [31]

To overcome these setbacks, the Tangle works by being collaborative. A transaction is added to the Tangle as a vertex after referencing two other vertices by linking to them through a directed edge, making the transaction reside at the end of the Tangle, commonly referred to as a tip which can be observed as gray squares in Figure 10. The links constitute a hash of the referenced vertices to form a verifiable and immutable decentralized ledger, like Ethereum’s MPT. As a transaction starts receiving more approvals directly or indirectly (getting deeper within the structure), the system accepts the state with a higher level of confidence, reasonably considering it being confirmed, shown as a white square in Figure 10. Nodes can only issue a transaction once a cryptographic puzzle is solved (PoW) albeit in an adaptive way to meet the requirements of IoT devices, rather than needing accelerated hardware such as GPUs or application-specific integrated circuits (ASICs). Another thing to keep note of is that the

IOTA network does not have a global state like Ethereum; it instead can be viewed as an asynchronous network. Nodes do not necessarily see the same set of transactions i.e., the same Tangle structure at the same time. Although the Tangle might contain conflicting transactions in the form of double-spending or inconsistent data, a consensus algorithm is put in place for nodes to use for eventually deciding which transactions are discarded. To filter out these types of transactions with the goal of reaching consensus, the nodes can run tip-selection algorithms such as the Markov Chain Monte Carlo (MCMC), or the Fast Probabilistic Consensus (FPC) [33] protocol which evidently shows how malicious tips are discarded with a high probability as described in the papers published by Popov et al. [29] [30] [31].

Chapter 3 – Historical Work

Gaur, Scotney, Par, & McClean [38], Chen [39], Foschini, Taleb, Corradi, & Bottazzi [40], as well as Wang, Cho, Lee, & Ma's [41] works are some of the earliest publications that realize the growing field of IoT and the need to develop a machine-to-machine communications platform to leverage the data IoT devices produce; enabling a common medium to establish smart city services within an intelligent transportation system (ITS). The paper of Foschini et al. [40] reviews existing protocols to form a distributed communications medium between multimedia systems that can enable a highly scalable heterogenous terminal between devices and central application servers, specifically designed for municipality services such as video surveillance, traffic condition management, and automatic detection and sanctioning of cars that pass a red light. Kubota, Okamoto, & Oda's article [43] describes implementing a system to warn drivers of nearby pedestrians and bicyclists close to intersections using LF signal generators and RFID tag readers on RSU's that communicate with on-board devices in vehicles.

Chen's [39] publication combines networking frameworks to establish an architecture for IoT devices composed of four layers with the idea that service-oriented applications would be triggered by mobile code that is delivered from one machine to another through RFID tags and readers. Wang et al. [41] propose an infrastructure architecture to facilitate the interactions between vehicle drivers and the outside world through a vehicle cloud solution, which was then updated by Gaur et al. [38] to include other domains that would eventually form a smart city architecture that can be incorporated into the health, environment, and transportation industries. One of the novel ideas proposed in the article of Gaur et al. [38] was to utilize Dempster-Shafer rules [59] to deal with uncertainty aspects of the data being exchanged to resolve trust. Misra's dissertation [44] described and implemented a protocol (RoAdNet) for dynamic roadway traffic conditions to maximize throughput of broadcast messages supported over large distances, while Ding and Xiao's research paper [11] proposed to deploy static nodes at road intersections to help relay data when there are no vehicles to deliver the packet to.

Pajooh, Rashid, Alam, & Demidenko [48] utilized Hyperledger Fabric's framework to implement a permissioned blockchain for the traceability of the data generated by IoT devices and tested the performance of such a blockchain-based multi-layered security model. This work proved that blockchain-based frameworks for data traceability and security produce an optimal throughput for many IoT applications. Z. Yang, K. Yang, Lei, Zheng, & Leung [18] proposed to incorporate blockchain into the trust management protocol that establishes the trust value of vehicles producing data in a VANET. This scheme enables RSU's to participate in the consensus layer to determine the trust value of vehicles from ratings uploaded by neighboring vehicles. George, Jaekel, & Saini [36] propose a similar framework for authentication, by having an authentication party maintain a public key infrastructure (PKI) that holds public and private keys of registered vehicles to make the procedure of validating messages faster if a vehicle is already listed. Javed, Jamal, Javaid, Haider, & Imran [45] developed a comparable system to the framework designed by George et al. [36] but added a monitoring authority layer as a gap between a vehicles' pseudonym published on the network as source of a message, and the central authority that handles the PKI of all participating nodes. This then tailored the service to an advertising dissemination medium between vehicles and advertisers in a region of interest.

The works of Hassan, Habiba, Ghani, & Shoaib [3], Yuan and Wang [5], Iftikhar, Javaid, Samuel, Shoaib, & Imran [34], as well as Jabbar, Kharbeche, Al-Khalifa, Krichen, & Barkaoui [35] incorporated blockchain within different layers of the OSI model in VANET topologies, by either residing at the top of the stack in the application layer to have a reliable database as reference for vehicles' application interfaces, or part of the physical and transport layers to establish a connection between messages disseminated by the vehicles' on-board units (OBUs). Jabbar et al. [35] also proposed to use smaller blockchain-trees for faster access and communication with the ledger based on traffic density in certain areas, having the blockchain structure reside within the application layer for direct access by the central servers that analyze and process the obtained data. Similarly, Iftikhar et al. [34] designed a framework where a central authority decides on what traffic event data residing in an inter-planetary file system (IPFS) that was proposed and designed by Benet [6], gets eventually published onto the blockchain thus removing the need for third parties' involvement in the data storing process. Akhtar et al. [47] offer detailed research into exploring the use of the IOTA platform alongside real-time IoT applications, specifically within the machine-to-machine (M2M) economy by leveraging IOTA's masked-authenticated-messaging (MAM) protocol to establish secure and verifiable data channels on the public Tangle.

Jiang, C. Wang, Y. Wang, & Gao [49] proposed a cross-chain framework utilizing Hyperledger's Fabric to integrate several distinct Tangle networks or "sidechains", suitable for certain IoT data management services, while having interoperability between them through permissioned access given to notary nodes that act as a gateway to access a decentralized data storage layer. Zichichi, Ferretti, & D'Angelo [46] proposed an architecture that uses IOTA's Tangle alongside Ethereum's network and IPFS to aggregate and collect crowd-sensed and user-managed data in VANETs by having the different projects able to communicate with the application unit and an authorization service. Cintron [56] modelled a consortium based distributed ledger network (DLN) infrastructure to serve as the back end for multiple ITS applications within the same instance where all stakeholders within it can provide services, access, and report data on the network. The infrastructure proposed by Cintron [56] is analogous to the OSI model, shown in Figure 11 below.

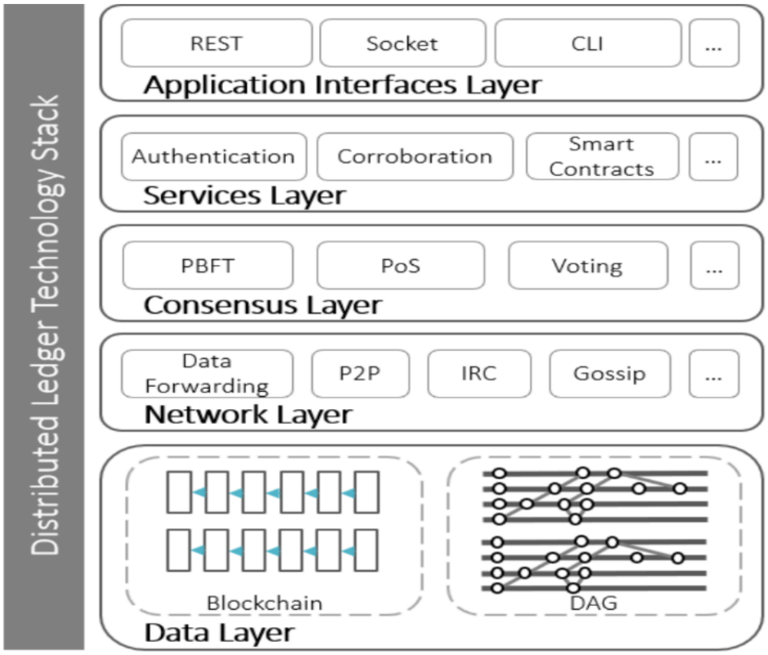


Figure 11 DLN-based Network Architecture [56]

This chapter highlighted several previous publications that offer insight and propose new frameworks and architectures related to enhancing a VANET's topology in terms of data dissemination, gathering, and storage. The motivation behind these works was to establish a dynamic and fault-tolerant system that can handle the ever-growing number of IoT devices that are being incorporated into all forms of the transportation system. Another common factor between the referenced works was to establish a machine-to-machine communication medium that does not rely on any middleman, shifting the network's structure to a more open and decentralized nature. To do so, several studies analyzed the feasibility of utilizing distributed ledger technologies in different layers of the network to better secure the data, while also offering additional services through autonomous agents residing within the network in the form of smart contracts. The consensus in all the previously mentioned publications show that it is entirely possible to incorporate blockchain and other distributed ledger-based technologies into a VANET's architecture, without affecting the throughput of messages being published by the different IoT devices within it. The takeaway from the ideas suggested and tested out in this chapter for this paper's work was to incorporate several different types of DLT networks within the transportation system. This was mainly based off of Cintron's model [56] to set up Hyperledger Fabric as the base permissioned network for all the organizations involved in the transportation system, as well as the proposed models done by Jiang et al. [49] and Zichichi et al. [46] to have different types of permissionless DLT networks that offer a wide variety of services to any IoT device that interacts with the VANET.

Chapter 4 – System Overview & Design

4.1.1 Proposed Architecture

The proposed architecture for this framework revolves around a multi-layered model that integrates multiple different distributed ledger-based networks (DLNs), capable of managing data coming from a multitude of IoTs within an intelligent transportation system (ITS). At the foundational layer, this model consists of a consortium blockchain network that is set up to act as a transitory medium for all the organizations within an ITS to interact and share data with each other in a collaborative, governed, and distributed manner. The key point is to have a governed base network to mitigate any risk of exploitation or manipulation by any potential outside attacker while keeping the participants within it known.

Rather than having the foundational layer be built upon an open, permissionless system such as Ethereum or IOTA that allow unknown identities to participate in the network, a permissioned blockchain network with a modular architecture has been favored instead, specifically Hyperledger's Fabric. The consortium blockchain network would allow multiple different sub-networks within it to be established, while also providing private data transferring capabilities between the sub-networks through special nodes, referred to as notary nodes. This layer would act as the decentralized access control station for the organizations within an ITS that are involved in maintaining a stable and safe ecosystem. The organizations involved are plentiful and might include government and law enforcement agencies, vehicle manufacturers, insurance companies, toll booths, parking authorities, and several other service providers. The foundational layer must be able to provide an integrated ecosystem in which all the different organizations can have access to and share new data under a set of predetermined rules. Transportation services can then further leverage the data on a need-to-know basis, thus enabling a collaborative network between transportation-related organizations and the various participants of the ITS such as vehicles and RSUs.

From the Fabric's point of view, every organization involved within the ITS would have its own certificate authority (MSP) that it can trust and have at least one participating peer node on the network that can verify, publish, and read data from the network. In addition to the peer node, each organization will have a notary node that enables data sharing between the different organizations via specific channels. Each organization can have its own set of rules that can be leveraged to interact with permissionless distributed ledger-based networks, referred to as sidechains in which vehicles and RSU's publish data about themselves, or their surroundings outside the consortium network. This approach enables different organizations to offer their unique services by interacting with the ITS participants, while still having a dedicated and trusted medium to interact with other organizations for more complex scenarios. This creates a cross-chain interactive data access model.

The task of each sidechain is to record sensor data generation events in its cluster network, setup accounts for each participant, and allow participants to invoke smart contracts that offer specific services. Meanwhile, the consortium blockchain is responsible for maintaining a log of successful or failed data access requests from a consortium member. In addition, the base HLF network has the ability to add specific data about the participants on the shared ledger between the consortium members to establish an extra layer of authentication for the participating vehicles and RSU’s. Figure 12 shows a generic overview of how the system would be organized. The different IoT clusters formed by the various vehicles and RSU’s interact with specific DLNs through either publishing data or requesting specific services from the system. The notary nodes are able to interact with the sidechains independently to verify data or initiate requests, while also being a part of the consortium network for exchanging data between each other privately. Each notary node is controlled by one of the many entities involved within the transportation system and would act as the gateway for transferring data from one organization to another.

Storing every piece of data published on the blockchain networks can cause a bottleneck in the system’s throughput, as well as require a lot of storage capacity. This makes it hard for some nodes, specifically IoT devices to participate in the distributed network as they would need to have a full copy of each ledger they wish to interact with. To mitigate the high cost in terms of storage for using this network, an inter-planetary file system (IPFS) [6] is used to help keep the data stored within the ledgers to a minimum. IPFS is a distributed, peer-to-peer (P2P) system for storing and accessing files. IPFS solves the storage problem by acting as the intermediary phase between data producers or consumers and the DLNs. ITS participants would publish data to IPFS nodes, which in turn would split, index, and hash the data before distributing them across the P2P network. Once the data gets distributed across the IPFS network, the index or starting point of the hash would be published within a transaction to a DLN so it can be stored on-chain. By utilizing IPFS as our storage medium for the actual data, there would be no need for nodes to store large chunks of data due to the fact that they are required to have a copy of the ledger for each network they participate with. The fact that content within the IPFS is also hashed and sorted out in a Merkle DAG tree, similar to what IOTA and certain version control software like GitHub use, data is also considered immutable the same way it would be within a DLN.

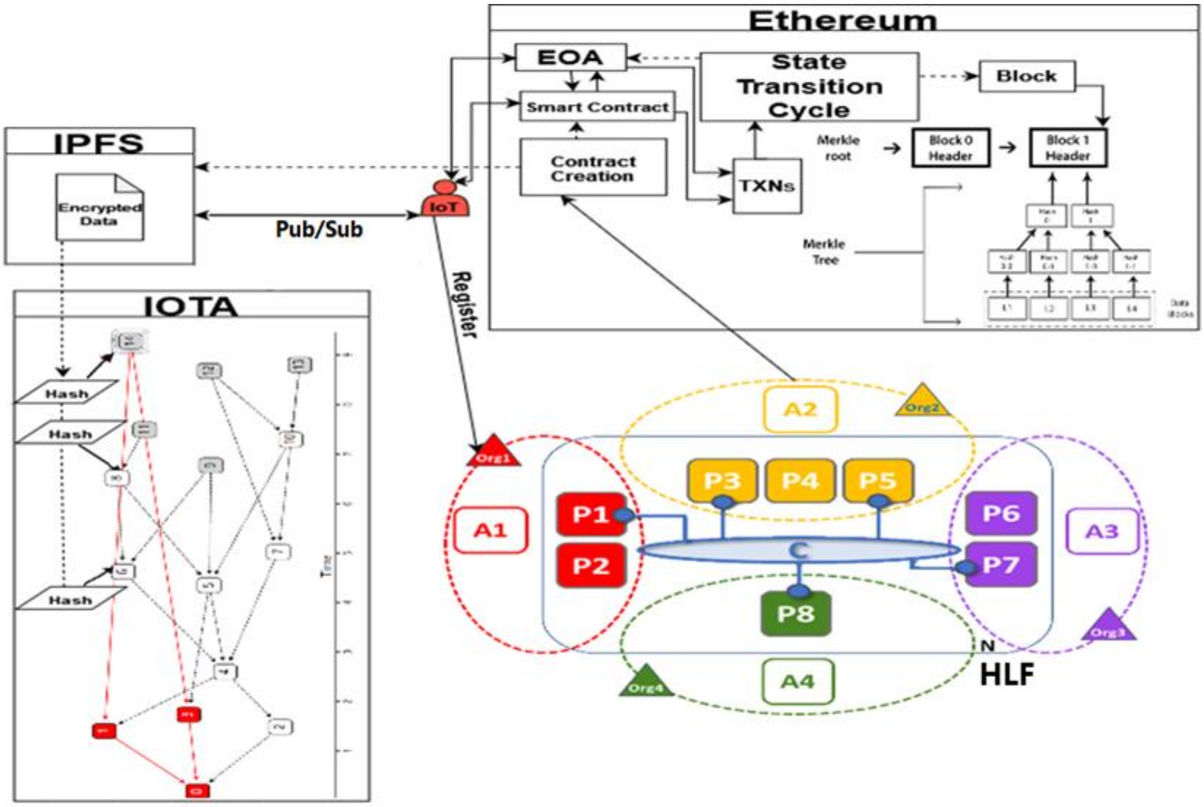


Figure 12 Proposed Network Layout

4.1.2 Example Workflow

To illustrate how this model behaves, this section demonstrates a couple of different scenarios where the interaction between the different modules becomes more apparent. The first scenario covers the interaction of all participants when an accident occurs between two vehicles, while the second scenario shows how the system works when a vehicle is sold and repurchased by another individual.

In the case of when a vehicular accident occurs, sensors within the vehicles involved would trigger warnings that are broadcasted to the networks they are connected to. Since both vehicles are located within the same region, they should be communicating to the same distributed ledger-based network. This allows the warning messages that were transmitted upon impact to propagate within the network, which would notify emergency response units (paramedics, firefighters) about it to immediately and rush to the scene. Upon confirming the accident, another accident message would be transmitted to the consortium network that will notify law enforcement agencies about the issue. This interaction can be all autonomous based on sensors and applications built into the vehicles to minimize response time. Once the accident is verified, law enforcement agencies can investigate further by checking with the department of motor vehicles (DMV) that both drivers and vehicles are up to date with inspections, registration, etc. before introducing both insurance companies to resolve the liability and financial problems. Assuming both vehicle's passengers are put in safe environments, the insurance companies can interact with each other through smart contracts to come out with a conclusion before relaying the results back to law enforcement agencies for further judgement. Throughout this whole process, the vehicles had to interact with a distributed ledger-based network, which upon a specific type of message got several transportation related organizations involved to further assess what happened and how to proceed further with the judgement. All the details about accident warning messages generated by such an event would be continuously distributed within the IPFS network, allowing other vehicles to be informed about the accident if requested, and either make their own rerouting decisions, or let the smart system do that for them instead.

In the case of when a vehicle is set to be repurchased, the buyer can first request access to the vehicle's information stored within the consortium network's ledger with the seller's permission. This allows the buyer to ensure that the seller owns the vehicle in question by validating the information stored within the ledger that is handled by the DMV, insurance company, the vehicle's manufacturer, and inspection stations on the consortium network. The ledger also provides the ability to check out all of the vehicle's history in terms of registration, inspections, maintenance, and accident history. Afterall, since the information that is stored on the ledger is immutable, this provides a more transparent and robust detailed overview of the vehicle's history from a single source of truth. Through this interaction between the two parties and the ledger, if the buyer is convinced, a smart contract can be invoked on the Ethereum network to start the process of transferring ownership and quite possibly the payment transaction. The smart contract would be triggered once the asking price for the vehicle has been transferred into its own account, another request would be initiated to the consortium network to request transfer of ownership thus removing the inherent risks and delays that usually accompany third parties and human intervention. A similar workflow can be used for car rentals or electric vehicle charging stations where a smart contract handles the communication between the parties involved as well as the infrastructure. Another promising example for the use of such a system would be through integrating the IOTA network into vehicle charging stations since it is designed for achieving seamless and fee-less machine-to-machine transactions.

By utilizing the several interconnected distributed ledger-based networks, this architecture can become the foundational layer that opens up a world of innovative opportunities for the automotive and intelligent transportation system. Transportation organizations and individuals could use this system as a platform to enhance the overall trustfulness for the various IoT devices participating in the network, validate transactions between different parties, enable secure micropayments, strengthen identity management, and improve data validation almost all autonomously. This shifts the responsibility and oversight from humans towards the electronics and the networks connecting them instead, inching us closer towards the concept of achieving an intelligent transportation system.

4.2 Experiment Setup

The following section provides the basic steps to set up some of the different components of the system presented in the previous section, specifically the IOTA and Hyperledger Fabric networks.

4.2.1 Chrysalis Private Network Setup:

From Amazon's Web Service's marketplace, an IOTA (Chrysalis) Private Tangle was initialized over an elastic computing (EC2) virtual server instance with the following configurations to set up and run it:

1) Creating a Virtual Private Cloud (VPC) with a private subnet associated to a security group that allows inbound traffic on ports 22, 4000, 8081, 8082, and 14265. According to the documentation found at https://github.com/iotaledger/one-click-tangle/blob/chrysalis/hornet-private-net/README_AWS.md these TCP ports offer the following services: 4000 (Tangle Explorer API), 8081 (Hornet's dashboard), 8082 (Tangle Explorer Frontend), and 14265 (IOTA protocol).

2) SSH into the machine and installing docker-compose first via the following command:

```
"sudo curl -L https://github.com/docker/compose/releases/download/1.21.0/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose"
```

3) Adding the required executable permissions to the binary via the following command:

```
"sudo chmod +x /usr/local/bin/docker-compose"
```

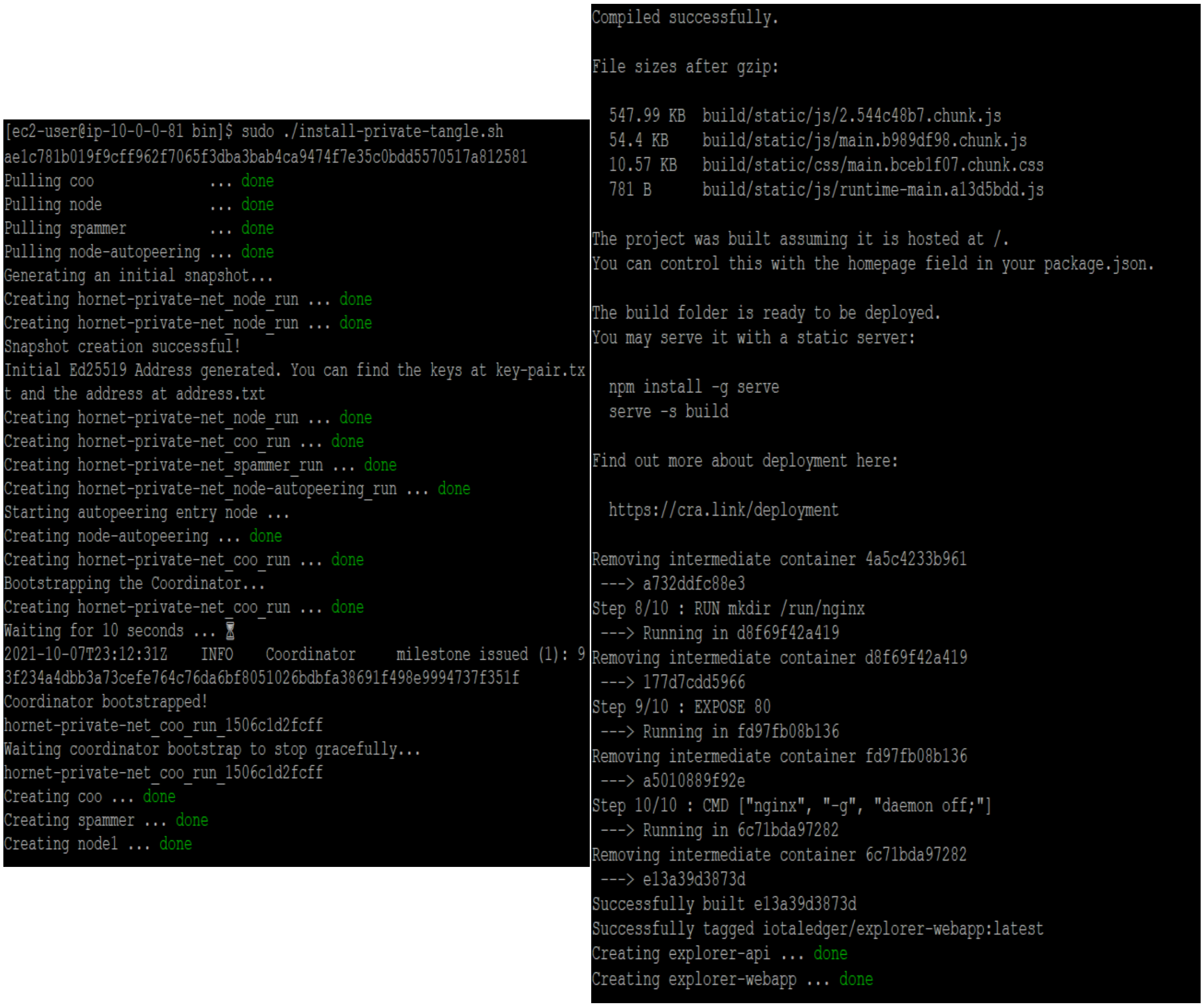
4) Creating a symbolic link via the following command:

```
"sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose"
```

5) Running the installation script via the following command:

"sudo ./install-private-tangle.sh" inside /bin/ directory.

The bootstrap and installation process will be initiated, as shown on the next page in Figure 13.

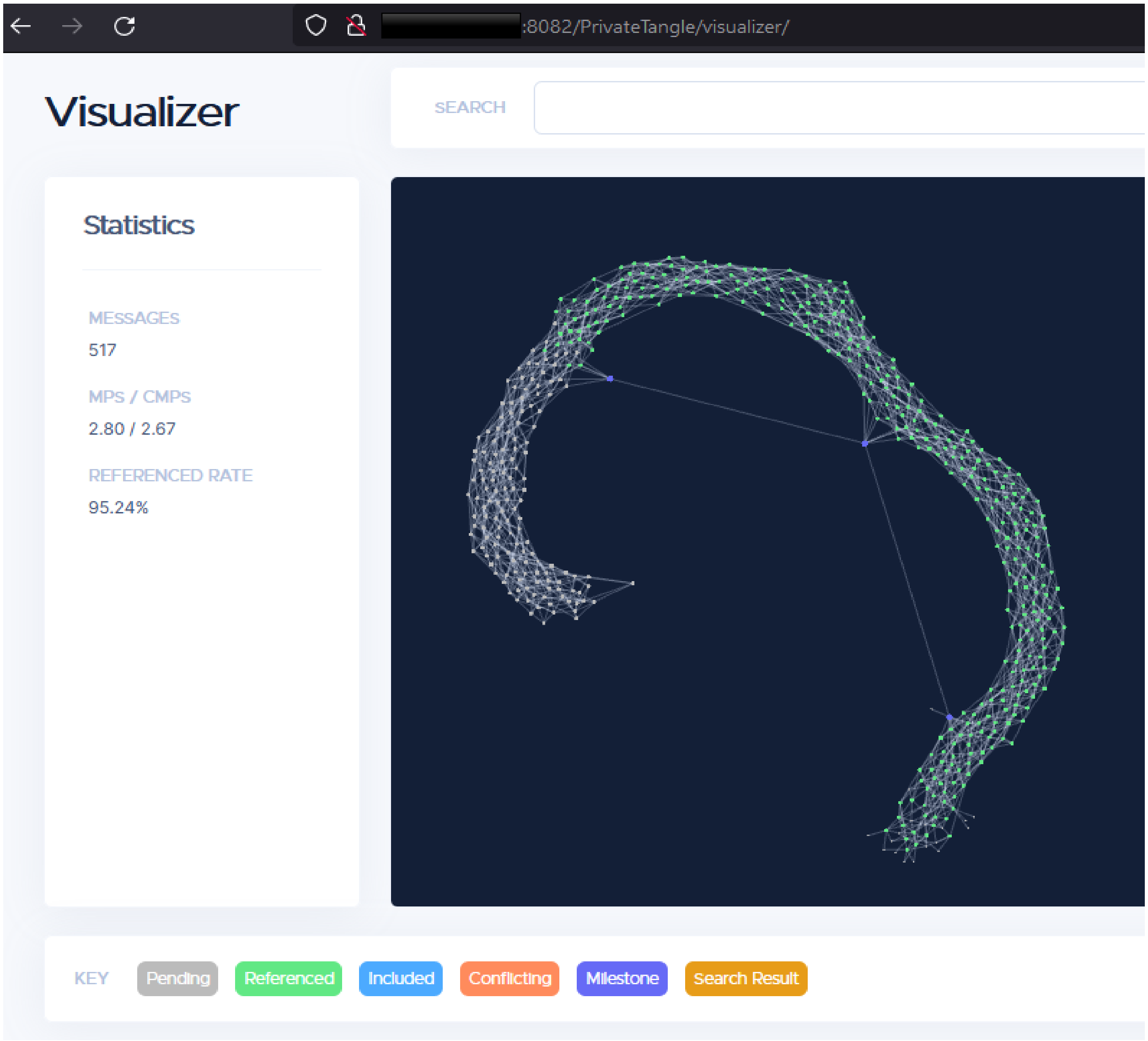


```
[ec2-user@ip-10-0-0-81 bin]$ sudo ./install-private-tangle.sh
ae1c781b019f9cff962f7065f3dba3bab4ca9474f7e35c0bdd5570517a812581
Pulling coo              ... done
Pulling node             ... done
Pulling spammer          ... done
Pulling node-autopeering ... done
Generating an initial snapshot...
Creating hornet-private-net_node_run ... done
Creating hornet-private-net_node_run ... done
Snapshot creation successful!
Initial Ed25519 Address generated. You can find the keys at key-pair.tx
t and the address at address.txt
Creating hornet-private-net_node_run ... done
Creating hornet-private-net_coo_run ... done
Creating hornet-private-net_spammer_run ... done
Creating hornet-private-net_node-autopeering_run ... done
Starting autopeering entry node ...
Creating node-autopeering ... done
Creating hornet-private-net_coo_run ... done
Bootstrapping the Coordinator...
Creating hornet-private-net_coo_run ... done
Waiting for 10 seconds ... ⌛
2021-10-07T23:12:31Z    INFO    Coordinator     milestone issued (1): 9
3f234a4dbb3a73cefe764c76da6bf8051026bdbfa38691f498e9994737f351f
Coordinator bootstrapped!
hornet-private-net_coo_run_1506c1d2fcff
Waiting coordinator bootstrap to stop gracefully...
hornet-private-net_coo_run_1506c1d2fcff
Creating coo ... done
Creating spammer ... done
Creating node1 ... done
```

```
Compiled successfully.

File sizes after gzip:

  547.99 KB  build/static/js/2.544c48b7.chunk.js
  54.4 KB    build/static/js/main.b989df98.chunk.js
  10.57 KB   build/static/css/main.bceb1f07.chunk.css
  781 B      build/static/js/runtime-main.a13d5bdd.js

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment

Removing intermediate container 4a5c4233b961
 ---> a732ddfc88e3
Step 8/10 : RUN mkdir /run/nginx
 ---> Running in d8f69f42a419
Removing intermediate container d8f69f42a419
 ---> 177d7cdd5966
Step 9/10 : EXPOSE 80
 ---> Running in fd97fb08b136
Removing intermediate container fd97fb08b136
 ---> a5010889f92e
Step 10/10 : CMD ["nginx", "-g", "daemon off;"]
 ---> Running in 6c71bda97282
Removing intermediate container 6c71bda97282
 ---> e13a39d3873d
Successfully built e13a39d3873d
Successfully tagged iotaledger/explorer-webapp:latest
Creating explorer-api ... done
Creating explorer-webapp ... done
```

Figure 13 IOTA Network's Initialization

Afterwards, the Private Tangle should be up and running. One can double check if everything was set up correctly by visiting the Tangle Explorer's webpage at http://<aws_dns_name>:8082 as shown in the figure below taken after 3 minutes of it being up, showing the Tangle spammer in action posting transactions onto the ledger. At this point, the private IOTA network setup is done and running.



Visualizer

Statistics

MESSAGES
517

MPS / CMPS
2.80 / 2.67

REFERENCED RATE
95.24%

Figure 14 The Tangle Visualized

4.2.2 Hyperledger Fabric Network Setup

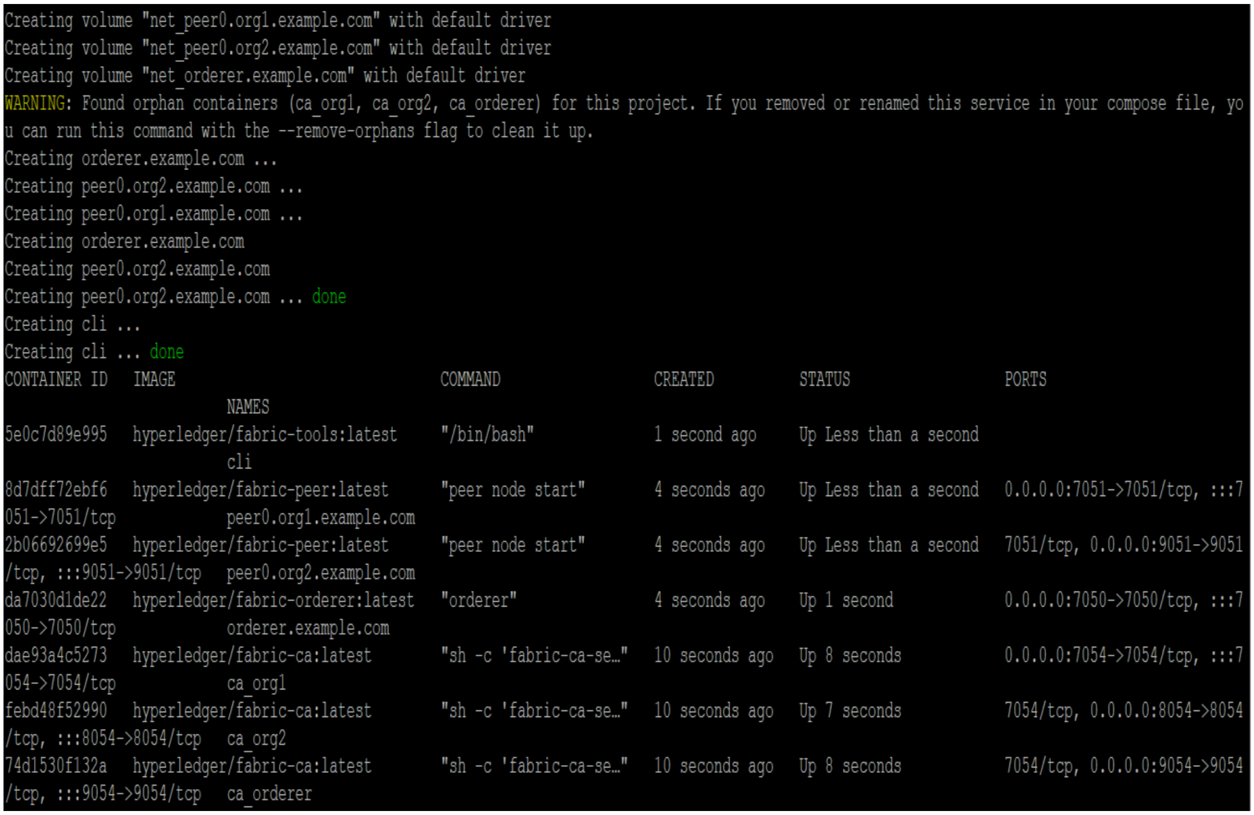
From Amazon's Web Service's (AWS) Marketplace, a Hyperledger Fabric virtual machine (HVM) was setup over an elastic computing (EC2) server instance with the following configurations to set up and run it:

1) Creating a VPC with a private subnet associated to a security group that allows inbound traffic on ports 22, 80, 443, 7051, 7052, and 7053 based on the recommended settings provided by the vendor.

2) SSH into the machine and updating the version of Go language to 1.17 before proceeding with deploying the network (machine ships with Go version 1.10 and is outdated).

3) Running the included script that generates the test network of two organizations with one peer in each, alongside a Fabric CA client that registers the nodes with the CA of each organization and generates an MSP for each identity. The following command also creates a channel within the network:

```
"./network.sh up createChannel -ca"
```

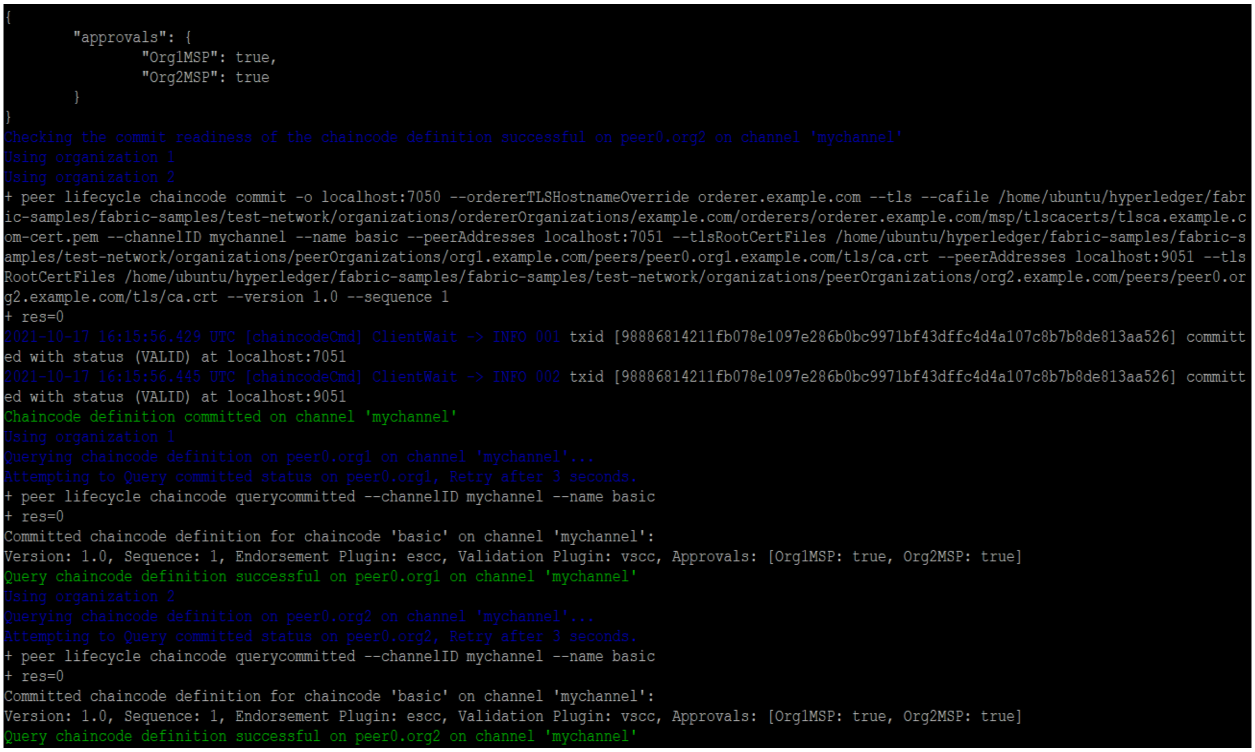


```
Creating volume "net_peer0.org1.example.com" with default driver
Creating volume "net_peer0.org2.example.com" with default driver
Creating volume "net_orderer.example.com" with default driver
WARNING: Found orphan containers (ca_org1, ca_org2, ca_orderer) for this project. If you removed or renamed this service in your compose file, yo
u can run this command with the --remove-orphans flag to clean it up.
Creating orderer.example.com ...
Creating peer0.org2.example.com ...
Creating peer0.org1.example.com ...
Creating orderer.example.com
Creating peer0.org2.example.com
Creating peer0.org2.example.com ... done
Creating cli ...
Creating cli ... done
CONTAINER ID   IMAGE                               COMMAND                  CREATED          STATUS                  PORTS
                          NAMES
5e0c7d89e995   hyperledger/fabric-tools:latest     "/bin/bash"              1 second ago     Up Less than a second
                          cli
8d7dff72ebf6   hyperledger/fabric-peer:latest      "peer node start"        4 seconds ago    Up Less than a second   0.0.0.0:7051->7051/tcp, :::7
051->7051/tcp             peer0.org1.example.com
2b06692699e5   hyperledger/fabric-peer:latest      "peer node start"        4 seconds ago    Up Less than a second   7051/tcp, 0.0.0.0:9051->9051
/tcp, :::9051->9051/tcp   peer0.org2.example.com
da7030d1de22   hyperledger/fabric-orderer:latest   "orderer"                4 seconds ago    Up 1 second             0.0.0.0:7050->7050/tcp, :::7
050->7050/tcp             orderer.example.com
dae93a4c5273   hyperledger/fabric-ca:latest        "sh -c 'fabric-ca-se…"   10 seconds ago   Up 8 seconds            0.0.0.0:7054->7054/tcp, :::7
054->7054/tcp             ca_org1
febd48f52990   hyperledger/fabric-ca:latest        "sh -c 'fabric-ca-se…"   10 seconds ago   Up 7 seconds            7054/tcp, 0.0.0.0:8054->8054
/tcp, :::8054->8054/tcp   ca_org2
74d1530f132a   hyperledger/fabric-ca:latest        "sh -c 'fabric-ca-se…"   10 seconds ago   Up 8 seconds            7054/tcp, 0.0.0.0:9054->9054
/tcp, :::9054->9054/tcp   ca_orderer
```

Figure 15 Generating HLF Network

4) Creating a chaincode to allow peers to interact with the channel's ledger. In this example, the provided "Basic Asset Transfer" chaincode was installed on both peers and deployed on the channel created via the following command:

"./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-go -ccl go"



```
{
        "approvals": {
                "Org1MSP": true,
                "Org2MSP": true
        }
}
Checking the commit readiness of the chaincode definition successful on peer0.org2 on channel 'mychannel'
Using organization 1
Using organization 2
+ peer lifecycle chaincode commit -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile /home/ubuntu/hyperledger/fabr
ic-samples/fabric-samples/test-network/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.c
om-cert.pem --channelID mychannel --name basic --peerAddresses localhost:7051 --tlsRootCertFiles /home/ubuntu/hyperledger/fabric-samples/fabric-s
amples/test-network/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tls
RootCertFiles /home/ubuntu/hyperledger/fabric-samples/fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/peers/peer0.or
g2.example.com/tls/ca.crt --version 1.0 --sequence 1
+ res=0
2021-10-17 16:15:56.429 UTC [chaincodeCmd] ClientWait -> INFO 001 txid [98886814211fb078e1097e286b0bc9971bf43dffc4d4a107c8b7b8de813aa526] committ
ed with status (VALID) at localhost:7051
2021-10-17 16:15:56.445 UTC [chaincodeCmd] ClientWait -> INFO 002 txid [98886814211fb078e1097e286b0bc9971bf43dffc4d4a107c8b7b8de813aa526] committ
ed with status (VALID) at localhost:9051
Chaincode definition committed on channel 'mychannel'
Using organization 1
Querying chaincode definition on peer0.org1 on channel 'mychannel'...
Attempting to Query committed status on peer0.org1, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID mychannel --name basic
+ res=0
Committed chaincode definition for chaincode 'basic' on channel 'mychannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org1 on channel 'mychannel'
Using organization 2
Querying chaincode definition on peer0.org2 on channel 'mychannel'...
Attempting to Query committed status on peer0.org2, Retry after 3 seconds.
+ peer lifecycle chaincode querycommitted --channelID mychannel --name basic
+ res=0
Committed chaincode definition for chaincode 'basic' on channel 'mychannel':
Version: 1.0, Sequence: 1, Endorsement Plugin: escc, Validation Plugin: vscc, Approvals: [Org1MSP: true, Org2MSP: true]
Query chaincode definition successful on peer0.org2 on channel 'mychannel'
```

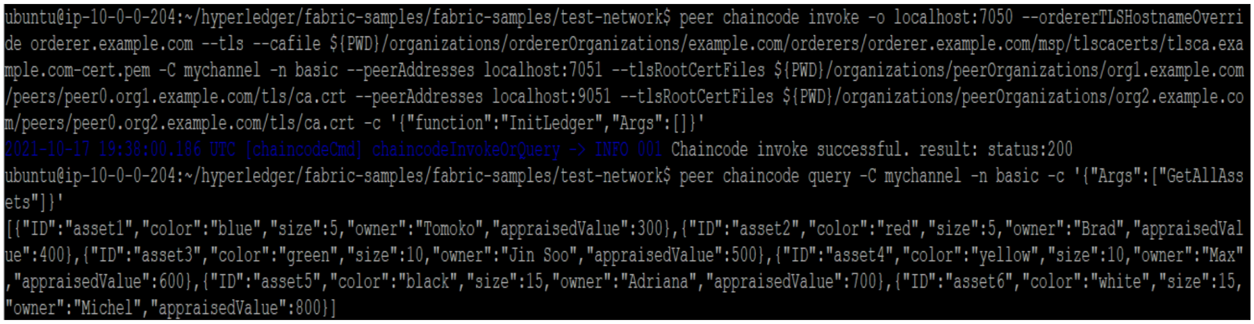
Figure 16 Installing Chaincode

5) Once the chaincode is deployed on the network, InitLedger function can be invoked to initialize the ledger with some assets provided as part of the fabric-samples directory within the HLF instance via the following command:

```
"peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com
--tls --cafile ${PWD}/organizations/ordererOrganizations/example.com/orderers/
orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem -C mychannel -n basic --
peerAddresses localhost:7051 --tlsRootCertFiles ${PWD}/organizations/peerOrganizations/
org1.example.com/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --
tlsRootCertFiles ${PWD}/organizations/peerOrganizations/org2.example.com
/peers/peer0.org2.example.com/tls/ca.crt -c '{"function":"InitLedger","Args":[]}'"
```

6) Afterwards, one can query the ledger by operating as one of the peers to get the list of assets that were added to the ledger from the previous step via the following command:

```
"peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAssets"]}'"
```



```
ubuntu@ip-10-0-0-204:~/hyperledger/fabric-samples/fabric-samples/test-network$ peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverri
de orderer.example.com --tls --cafile ${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.exa
mple.com-cert.pem -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles ${PWD}/organizations/peerOrganizations/org1.example.com
/peers/peer0.org1.example.com/tls/ca.crt --peerAddresses localhost:9051 --tlsRootCertFiles ${PWD}/organizations/peerOrganizations/org2.example.co
m/peers/peer0.org2.example.com/tls/ca.crt -c '{"function":"InitLedger","Args":[]}'
2021-10-17 19:38:00.186 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke successful. result: status:200
ubuntu@ip-10-0-0-204:~/hyperledger/fabric-samples/fabric-samples/test-network$ peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAss
ets"]}'
[{"ID":"asset1","color":"blue","size":5,"owner":"Tomoko","appraisedValue":300},{"ID":"asset2","color":"red","size":5,"owner":"Brad","appraisedVal
ue":400},{"ID":"asset3","color":"green","size":10,"owner":"Jin Soo","appraisedValue":500},{"ID":"asset4","color":"yellow","size":10,"owner":"Max"
,"appraisedValue":600},{"ID":"asset5","color":"black","size":15,"owner":"Adriana","appraisedValue":700},{"ID":"asset6","color":"white","size":15,
"owner":"Michel","appraisedValue":800}]
```

Figure 17 Chaincode Invoked

Chapter 5 – Conclusion

5.1 Overview

This chapter summarizes the work done in this paper by restating the concepts and the framework's architecture that were introduced, described, and proposed for improving the infrastructure of intelligent transportation systems (ITS). The focus of this paper was to analyze and design a new network model by leveraging distributed ledger technologies (DLT) to create a more collaborative and trustful model for an ITS. The model is designed to better improve scalability and decentralize the governance across a VANET topology, while making sure that the data and access control are secure against malicious actors and attacks. Due to the nature of model's modular layout, it can be used as a foundation to further develop additional smart city features. In addition, this design has the capability to upgrade pre-existing services such as autonomous traffic management, registration and transferring of vehicle ownership, as well as power grid contribution for safety, efficiency, and profitability. To further improve upon the model's design and interoperability, this chapter also highlights several ongoing developmental projects that can help make the framework a reality.

5.2 Summary

IoT devices have been getting integrated into cyberspace, more specifically within vehicles and road-side units, allowing them to establish autonomous and real-time communications with each other, shifting the responsibility away from humans in terms of decision making and control. Since 2001, vehicular ad-hoc networks (VANETs) have been established to allow vehicles to spontaneously form wireless networks, so that information can be relayed between the network participants in a distributed manner. The problem with these types of networks is that the participants must inherently trust each other, regardless of whether there are verification or authentication mechanisms in place for the data produced. This affects the safety of the whole transportation system with the rise and concern over cyberattacks.

To mitigate the risks behind the lack of trust between participants within a distributed network, blockchain technology was used in developing new network topologies. Blockchain can be described as a digitally shared, immutable ledger that facilitates the process of recording and tracking assets in a trustful manner. Some networks aimed to establish a more robust framework by utilizing blockchain technology to create an open access infrastructure for applications to be built on, most notably the Ethereum network. This led to another advancement in the world of blockchain and distributed ledger-based technologies, where autonomous agents in the form of computer code can be leveraged to establish decentralized, decision-making protocols under a predefined set of rules more formally known as smart contracts.

However, running and using the Ethereum network is computationally expensive, which makes it not suitable for IoT devices within a VANET alone. Instead, the IOTA network was developed with IoTs and a machine-to-machine economy in mind that can scale efficiently for low-powered devices with a fee-less structure. This paper suggested composing an intelligent transportation system network that interconnects IOTA and Ethereum for vehicles, roadside units, and business providers. In addition to that, transportation-related organizations would be a part of another distributed-ledger based network that is permissioned, allowing for network participants' verification as well as a way for providing their own services to the whole ecosystem. The permissioned consortium network was setup using Hyperledger's Fabric framework. Since data storage can become a bottleneck for IoT devices with time as the ledgers grow, an inter-planetary file system (IPFS) was proposed to store data off-chain while keeping a hash of the data on the shared ledger instead. The four networks used in this model for data acquisition and storage, autonomous service providing, access control and governance, as well as payments are IPFS, Ethereum, Hyperledger Fabric, and IOTA.

5.3 Discussion and Future Work

Given the continuous development and rapid advancement in blockchain and transportation system technologies, as well as the growing interest by major enterprises and countries to establish intelligent transportation systems, implementing DLTs within VANETs should be explored and experimented with further. Previous works stated in chapter 3 confirm that the use of DLTs by IoT devices in a VANET does not cause latency or bottlenecks in terms of system throughput and is convenient for deployment across multiple networks. In addition to that, the nature of such a distributed model ensures constant availability, integrity, and security of services deployed on the network.

While production-level solutions for distributed ledger-based networks implemented with an ITS are still in development and experimenting stages, several project upgrades seem promising in making the designed system a reality. IOTA is currently in a transitory phase, shifting towards a new architecture under the name Coordicide (IOTA 2.0), which will bring in smart contract capability to the network. When Coordicide is finalized, there will be no need to also have Ethereum as part of the model for autonomous service providing, decreasing the complexity of the system. Coordicide will also have the capability to seamlessly connect with Hyperledger's Fabric through IOTA's connector. Proof of concept was established with previous version in 2019 but was put on hold for now after the decision to entirely redesign IOTA was decided. Future work can also include setting up the networks over a hardware-based environment such as Raspberry Pi's as well as creating smart contracts that are triggered by real-time physical events.

References

1) Faezipour M, Nourani M, Saeed A, and Addepalli S. 2012. “Progress and Challenges in Intelligent Vehicle Area Networks.” Commun. ACM 55, 2 (2012), 90–100

2) Channappagouda M, and Venkataram P. “Mobile Agent Based Node Monitoring Protocol for MANETs”. In: 2013 National Conference on Communications (NCC) (2013), pp. 1–5.

3) Hassan M.A, Habiba U, Ghani U, Shoaib M. 2019. “A Secure Message-Passing Framework for Inter-Vehicular Communication Using Blockchain”. *International Journal of Distributed Sensor Networks*, *15*(2). https://doi.org/10.1177/1550147719829677

4) Al-Kahtani MS. "Survey on Security Attacks in Vehicular Ad Hoc Networks (VANETS)", In: 2012 6th international conference on signal processing and communication systems (ICSPCS), Gold Coast, QLD, Australia, 12-14

5) Yong Yuan, and Fei-Yue Wang, “Towards Blockchain-based Intelligent Transportation Systems”, 2016 IEEE 19thInternational Conference on Intelligent Transportation Systems (ITSC), Windsor Oceanico Hotel, Rio de Janerio, Brazil, Nov.1-4, 2016

6) Benet, J. (2014). *IPFS -Content Addressed, Versioned, P2P File System (DRAFT 3)*. https://arxiv.org/pdf/1407.3561.pdf

7) D’anna, Gloria D. "Cybersecurity for Commercial Vehicles". Warrendale, Pennsylvania, Usa, SAE International, 2019.

8) Han, M.-S., Lee, S. J., & Bae, W.-S. (2017). “A secure and efficient V2V authentication method in heavy traffic environment”. Wireless Personal Communications, 93(1), 245–254.

9) S. Ucar, S. C. Ergen and O. Ozkasap, "Multihop-Cluster-Based IEEE 802.11p and LTE Hybrid Architecture for VANET Safety Message Dissemination," in IEEE Transactions on Vehicular Technology, vol. 65, no. 4, pp. 2621-2636, April 2016, doi: 10.1109/TVT.2015.2421277.

10) Remy, G., Senouci, S., Jan, F., & Gourhant, Y. (2011). LTE4V2X: LTE for a Centralized VANET Organization. 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011.

11) Y. Ding and L. Xiao, "SADV: Static-Node-Assisted Adaptive Data Dissemination in Vehicular Networks," in IEEE Transactions on Vehicular Technology, vol. 59, no. 5, pp. 2445-2455, Jun 2010, doi: 10.1109/TVT.2010.2045234.

12) R. Nakanishi, Y. Zhang, M. Sasabe and S. Kasahara, "IOTA-Based Access Control Framework for the Internet of Things," 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), 2020, pp. 87-95, doi: 10.1109/BRAINS49436.2020.9223293.

13) M.T. Goodrich and R. Tamassia. ”Introduction to Computer Security”. Pearson Education Inc. Boston, 2011.

14) L. Chen, H. Tang and J. Wang, "Analysis of VANET security based on routing protocol information," 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP), Beijing, 2013, pp. 134-138.doi: 10.1109/ICICIP.2013.6568055.

15) A. Ahamed and H. Vakilzadian, "Issues and Challenges in VANET Routing Protocols," 2018 IEEE International Conference on Electro/Information Technology (EIT), 2018, pp. 0723-0728, doi: 10.1109/EIT.2018.8500180.

16) Liu, Ling, and Weisong Shi. “Trust and Reputation Management.” Ieee Internet Computing, vol. 14, no. 5, 2010, doi:10.1109/MIC.2010.124.

17) Morrison, R., Mazey, N. C. H. L., & Wingreen, S. C. (2020). The DAO Controversy: The Case for a New Species of Corporate Governance? Frontiers in Blockchain, 3. https://doi.org/10.3389/fbloc.2020.00025

18) Z. Yang, K. Yang, L. Lei, K. Zheng and V. C. M. Leung, "Blockchain-Based Decentralized Trust Management in Vehicular Networks," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1495-1505, April 2019, doi: 10.1109/JIOT.2018.2836144.

19) S. Gurung, D. Lin, A. Squicciarini, and E. Bertino, “Information-oriented trustworthiness evaluation in vehicular ad-hoc networks,” inProc. International Conference Network System Security, Madrid, Spain, Jun. 2013, pp. 94–108.

20) Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf

21) Karame, G., “On the security and scalability of bitcoin’s blockchain” Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1861-1862, 2016

22) David, C. “Computer Systems Established, Maintained and Trusted by Mutually Suspicious Groups.” 1982. URL

{https://nakamotoinstitute.org/static/docs/computer-systems-by-mutually-suspicious-groups.pdf}

23) S. Haber and W.S. Stornetta, “How to Time-Stamp a Digital Document,” Advances in Cryptology—CRYPTO ’90, A.J. Menezes and S.A. Vanstone, eds., LNCS 537, Springer, 1991, pp. 437–455.

24) H. Finney, “Reusable Proofs of Work”, 2004. URL {https://nakamotoinstitute.org/finney/rpow/index.html}

25) Buterin, V. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.2013a. URL {http://ethereum.org/ethereum.html}

26) Wood, G. Ethereum: “A Secure Decentralised Generalised Transaction Ledger.” 2021 URL {https://ethereum.github.io/yellowpaper/paper.pdf}

27) Bagloee A., Saeed, et al. “Tradable Mobility Permit with Bitcoin and Ethereum - a Blockchain Application in Transportation.” Vol. 8, 2019, doi: 10.1016/j.iot.2019.100103.

28) “IOTA: A Cryptocurrency for Internet-of-Things.” See http://www.iotatoken.com/, and https://bitcointalk.org/index.php?topic=1216479.0

29) Popov, S. (2017, October 1). “The Tangle”. Retrieved from https://www.whitepaperdatabase.com/wp-content/uploads/2018/03/IOTA-MIOTA-Whitepaper.pdf

30) Popov, S., & Lu, Q. (2019, January). “IOTA: Feeless and Free”. IEEE Blockchain Technical Briefs. Retrieved from https://blockchain.ieee.org/technicalbriefs/january-2019/iota-feeless-and-free

31) Popov, S., Saa, O., & Finardi, P. (2019, July 4). “Equilibria in the Tangle”. Retrieved from https://arxiv.org/pdf/1712.05385.pdf

32) Lerner, S.D. (2015) “DagCoin: A Cryptocurrency Without Blocks.” https://bitslog.wordpress.com/2015/09/11/dagcoin/

33) Popov, S. and Buchanan, W. “FPC-BI: Fast probabilistic consensus within byzantine infrastructures”. CoRR, abs/1905.10895, 2019.

34) Iftikhar, M. S., Javaid, N., Samuel, O., Shoaib, M., Imran, M “An Incentive Scheme for Vanets Based on Traffic Event Validation Using Blockchain”. 2020 International Wireless Communications and Mobile Computing (Iwcmc), IEEE, 2020, pp. 2133–2137.

35) Jabbar, R., Kharbeche, M., Al-Khalifa, K., Krichen, M., & Barkaoui, K. “Blockchain for the Internet of Vehicles: A Decentralized Iot Solution for Vehicles Communication Using Ethereum.” Sensors, vol. 20, no. 3928, 2020.

36) S. A. George, A. Jaekel and I. Saini, "Secure Identity Management Framework for Vehicular Ad-hoc Network using Blockchain," 2020 IEEE Symposium on Computers and Communications (ISCC), 2020, pp. 1-6, doi: 10.1109/ISCC50000.2020.9219736.

37) Yang, Yao-Tsung, Li-Der Chou, Chia-Wei Tseng, Fan-Hsun Tseng, and Chien-Chang Liu. “Blockchain-based traffic event validation and trust verification for VANETs.” IEEE Access 7 (2019): 30868-30877.

38) Gaur, A., Scotney, B., Parr, G., & McClean, S. “Smart City Architecture and Its Applications Based on Iot.” Procedia Computer Science, vol. 52, 2015, pp. 1089–1094., doi: 10.1016/j.procs.2015.05.122.

39) Chen, Min. “Towards Smart City: M2m Communications with Software Agent Intelligence.” Multimedia Tools and Applications, vol. 67, no. 1, 2013, pp. 167–178., doi:10.1007/s11042-012-1013-4.

40) Foschini, L., Taleb, T., Corradi, A., & Bottazzi, D. “M2m-Based Metropolitan Platform for Ims-Enabled Road Traffic Management in Iot.” Ieee Communications Magazine, vol. 49, no. 11, 2011, doi:10.1109/MCOM.2011.6069709.

41) Wang J, Cho J, Lee S, Ma T. “Real time services for future cloud computing enabled vehicle networks.” 2011 Int Conf Wirel Commun Signal Process. 2011:1-5. doi:10.1109/WCSP.2011.6096957.

42) Ferraro, P., King, C., Shorten R. “Distributed Ledger Technology for Smart Cities, the Sharing Economy, and Social Compliance.” Ieee Access, vol. 6, 2018, pp. 62728–62746.

43) Kubota, S., Okamoto, Y. and Oda, H. “Safety driving support system using RFID for prevention of pedestrian-involved accidents.” In Proceedings of the 6th International IEEE Conference on ITS Telecommunication (June 2006), 226-229.

44) Misra, Priyashraba, "RoAdNet: Robust Adaptive Network for Information Diffusion in VANET" (2019). Graduate Theses, Dissertations, and Problem Reports. 7476. https://researchrepository.wvu.edu/etd/7476

45) Javed, M. U., Jamal, A., Javaid, N., Haider, N., Imran, M., “Conditional Anonymity Enabled Blockchain-Based Ad Dissemination in Vehicular Ad-Hoc Network”. 2020 International Wireless Communications and Mobile Computing (IWCMC), IEEE, 2020, pp. 2149–2153.

46) Zichichi, M., Ferretti, S., D'Angelo, G. “A Distributed Ledger Based Infrastructure for Smart Transportation System and Social Good”. 2020 Ieee 17th Annual Consumer Communications & Networking Conference (Ccnc), IEEE, 2020, pp. 1–6.

47) Akhtar, M.M.; Rizvi, D.R.; Ahad, M.A.; Kanhere, S.S.; Amjad, M.; Coviello, G. “Efficient Data Communication Using Distributed Ledger Technology and IOTA-Enabled Internet of Things for a Future Machine-to-Machine Economy”. Sensors 2021,21, 4354.https://doi.org/10.3390/s21134354.

48) Pajooh, H., Rashid, M., Alam, F., Demidenko, S. “Hyperledger Fabric Blockchain for Securing the Edge Internet of Things”. Sensors 2021,21, 359. https://doi.org/10.3390/s21020359

49) Jiang, Y., Wang, C., Wang, Y., & Gao, L. “A Cross-Chain Solution to Integrating Multiple Blockchains for IoT Data Management.” Sensors (Basel, Switzerland) vol. 19,9 2042. 1 May. 2019, doi:10.3390/s19092042

50) Sheikh, M.S.; Liang, J.; Wang, W. “A Survey of Security Services, Attacks, and Applications for Vehicular Ad Hoc Networks (VANETs).” *Sensors* 2019, *19*, 3589. https://doi.org/10.3390/s19163589

51) Salman, Tara, Jain, Raj, Gupta, & Lav. (2018). “Probabilistic Blockchains: A Blockchain Paradigm for Collaborative Decision-Making.” 10.1109/UEMCON.2018.8796512.

52) Saleem, M. A., Zhou, S., & Sharif, A. “Data Transmission Using Iot in Vehicular Ad-Hoc Networks in Smart City Congestion.” Mobile Networks and Applications, vol. 24, no. 1, 2019, pp. 248–258., doi:10.1007/s11036-018-1205-x.

53) Tani, Takenobu. “Ethereum-EVM Illustrated”. 2018. URL {https://github.com/takenobu-hs/ethereum-evm-illustrated}

54) Wackerow, P., Richards, S.; & Cordell, R. “Ethereum Development Documentation”. Edited: May 5, 2021. URL {https://ethereum.org/en/developers/docs/}

55) Lee, T. “Ethereum Blockchain Mechanism, An Interpretation of the Ethereum Project Yellow Paper”. Ethereum Stack Exchange. 21 June. 2016, URL {https://ethereum.stackexchange.com/questions/268/ethereum-block-architecture}

56) Cintron, L.A., "Modeling a Consortium-based Distributed Ledger Network with Applications for Intelligent Transportation Infrastructure" (2019). Theses and Dissertations. 2252. https://scholar.afit.edu/etd/2252

57) Androulaki, Elli et al. 2018. “Hyperledger fabric: a distributed operating system for permissioned blockchains.” In Proceedings of the Thirteenth EuroSys Conference (EuroSys '18). Association for Computing Machinery, New York, NY, USA, Article 30, 1–15. DOI: https://doi.org/10.1145/3190508.3190538

58) O’Dowd, Anthony; et al. 2021. “A Blockchain Platform for the Enterprise.” Hyperledger Fabric: A Blockchain Platform for the Enterprise, https://hyperledger-fabric.readthedocs.io/en/latest/index.html.

59) Dempster AP. A Generalization of Bayesian Inference. J R Stat Soc Ser B. 1968; 30:205-247.