FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Bo Li

Blockchain and smart contracts in health-related MyData scenario

Master's Thesis
International Master's in Computer Science and Engineering
(Ubiquitous Computing)
April 2017

Li B. (2017) Blockchain and smart contracts in health-related MyData Scenario. University of Oulu, Degree Programme in Computer Science and Engineering. Master's Thesis, 58 p.

ABSTRACT

The MyData is concept framework that refers to human-centric ways of personal data management. Personal data gained significant attention recently. As the developing of Ubicomp technology, more and more particularly personal data are generating and collecting. Personal data own increasingly important economic, social, and practical value. However, individuals have little or no power to control when and how their data being created or processed by companies, organizations or governments. The MyData aim to provide individuals with practical methods to obtain, access, and utilize their personal datasets and to encourage organizations to give users control over their personal data. In this way, access and trade personal data can expect to build an open data market. Two challenges to achieve this goal is how to gain the individuals trust and permission and how to provide a more human-centric way to support personal data management and utilization.

To explore a novel and reliable way to address the challenges in MyData, this thesis utilizes blockchain technology to support MyData framework. Blockchain is a decentralized transparent ledger with the transaction information that shared among all peer-to-peer network nodes. It has the potential to gain users trust and provide a solution to gain users permission in data trade.

This thesis work focuses on studying blockchain and smart contract performance in MyData architecture. An Ethereum blockchain based MyData system that combined AWARE platform designed and implemented. The system deploys smart contract that provides users' account management, personal data access, trade services, and information inquiry services in the Ethereum blockchain. Based on this system, two experiments designed to evaluate the performance of the integrated MyData system. The experiments results demonstrate how blockchain can facilitate MyData concept and how gas price influences the system performance. The thesis work shows that the blockchain and smart contract have the potential to provide the necessary technology support to solve the challenge in gain users' trust and permission and support new business models and open data market to benefit both the data consumer and data producer. Additionally, blockchain and the smart contract can provide a more fine-grained and transparent way to help individuals to manage and utilize their personal data.

Keywords: Ethereum, Smart Contract, AWARE, Personal Data

TABLE OF CONTENTS

FOREWORD

I would like to appreciate my supervisors Dr. Marko Jurmu and Prof. Jukka Riekki for the guidance and support. I also thank M.Sc. Lauri Loven, M.Sc. Marta Cortes and Docent Denzil Ferreira gave me the opportunity to join this project and wrote the thesis. I also thank my parents and friends gave me this unforgettable experience in Finland.

Oulu, 20.4.2017
Bo Li

ABBREVIATIONS

API	Application programming interface
QR	Code Quick Response Code
ESM	Experience Sampling Method
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
MQTT	Message Queue Telemetry Transport
GPS	Global Positioning System
IoT	Internet of Things
ANR	Android Not Responding
JSON	JavaScript Object Notation
RPC	Remote Procedure Call
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
P2P	Peer to Peer
PoW	Proof of Work
PoS	Proof of Stake
PoB	Proof of Burn
PoET	Proof of Elapsed Time
BTF	Byzantine Fault Tolerant
EVM	Ethereum Virtual Machine
ASIC	Application-Specific Integrated Circuit
EOA	Externally Owned Account
GUI	Graphical User Interface
APP	Application
SATA	Serial Advanced Technology Attachment
MD5	Message-Digest Algorithm
iOS	iPhone Operation System

1. INTRODUCTION

1.1. Motivation

The amount of personal data in our society rapidly increasing. Collecting and generating personal data based on people and their activities becom easier and more convenient. One reason to leading this situation is the continuous reduction of the smart devices price allows equip it everywhere. In addition, more and more specific and powerful smart devices are designing and popularizing, such as smart wearable, unmanned aerial vehicles, and smart home devices. The service provider and organizations gained more options to collect even customize users' data. Meanwhile, an increasing amount of data is becoming available on the internet. According to a report in 2014 [72], Facebook had collected 300 petabytes of personal data. When we connecting the Internet, no matter us using the browser or map application, even we just open it without logging. We continuously producing and publishing data about ourselves.

The continuously collected and analyzed data significantly promoting the innovation and economic growth. Companies, organizations and governments use the data to optimizing services, making better decisions, predicting trends and more [73]. While we obtain the benefits of this data-driven world, a more human centric way of data management and user privacy draw continuing public attention. Poikola etc. [24] proposed MyData that is an infrastructure-level approach, a consent-based data management. They hold a view that individuals should have the legal right and practical means to manage their personal data and privacy, not be passive targets but empower actors. Meanwhile, organizations should have practical methods for gain individuals' permission to use their personal data.

No real mechanisms allow individuals to share their data at present. Ali etc. [2] implemented a Health Records Retrieval system HER, it effective for local healthcare service, but HER's global implementation still an issue. For example, when people cross countries, her/his personal data cannot share with foreign health organization through HER. Individuals have the power to manage their data provides a potential solution to this situation.

Although MyData creates new possibilities and provides a great vision for the future personal data management and utilization. To achieve its goal presents multiple challenges. MyData aims to enable decentralized management of personal data to build an open business environment. The first challenge is how to protect the privacy. The data carries sensitive personal information such as health situation, location, movement, photos, videos etc. How MyData to achieve this decentralized management and how MyData protect users' privacy under the decentralized management?

The second challenge is how to gain users' trust and permission. MyData intends to give individuals the right to share or sell personal data to third parties, based on this to build an open data market. How MyData provide a trusty approach to verify the data consumers and how MyData provide an appropriate way for individuals to give the permission to data consumers? The core is MyData should provide a safely and trusty verification approach to prevent users' property would not be damaged by the malicious data consumer.

1.2. Research Question

The increasing amount of personal data can arise the new data-driven digital economy [10]. The data trade's security plays most important role in a data-driven digital ecosystem. The ecosystem should enable the personal data safely stored and transferred. In addition, data can flow and create value within the ecosystem, the data trade requires transparent and unchangeable to protect users' property. Moreover, the ecosystem can provide high performance transaction rate and high quality services. Hence, the blockchain technology has drawn MyData researchers' attention. In the new MyData paradigm, there has 'MyData operators' that govern the data for the individual. In this thesis, MyData Operator modeled by an Ethereum smart contract. The thesis aims to investigate the benefits and issues in deploy blockchain and smart contract technologies in MyData paradigm to explore a more fine-grained and transparent way for users to manage their data. Meanwhile, provide a more efficient way to utilize the fragment of personal data.

More specifically, we focus on the research questions as follow:

- How can a person manage MyData in a more fine-grained and transparent way?

- How can a person define who gets to use the MyData, and how?

Contributions of this thesis are as follows: first, a MyData system combined Ethereum and AWARE platform developed to study the benefits of applying blockchain and smart contract with personal data management and utilization in data trade. The system can manage user accounts, personal data and handle the personal data trade between data producers and data consumers. Second, two experiments designed to research the performance of the data trade and management. Experiments results give constructive suggestions to the future blockchain based MyData systems design and implementation.

1.3. Structure of Thesis

The rest of the thesis organized as follows. Chapter 2 describe the background and basic concepts about MyData, AWARE platform and Blockchain and Chapter 3 present a blockchain based MyData system. In Chapter 4, we describe the experiments and the evaluation results. Chapter 5 discuss the findings and future works and Chapter 6 conclude the thesis.

2. BACKGROUND

This research based on the three technologies: MyData, Blockchain, AWARE. This chapter provides details on the related concepts, and technologies for these three fields. MyData section descript the motivation, concept and benefits of MyData. Blockchain section introduce the related technologies, Ethereum framework and smart contract. AWARE section introduces AWARE, AWARE's data share strategies and the related frameworks.

2.1. MyData

2.1.1. Motivation

As the Ubicomp technologies developing, people's personal data can be easily collecting and record used to build the quantified self. Rich of computers, sensors, mobile devices, and the Internet enable devices to record a large amount of personal data which including the physiological even psychological data [1]. Personal data can be collected and used by different areas and organizations, which include social media platforms, search engines, supermarkets, healthcare providers and so on. Companies and organizations insights into those data to gain better understanding about how to design suitable services for users and how to advertising potential users. All these data can benefit in self-reflection to help users make better decisions [4] and be more aware of their behavior [3]. In addition, change behaviors in the health field [5]. People are continually developing tools and apps for personal informatics, which aiming for providing self-knowledge based on these personal data [7]. Many Ubicomp tools are working for personal informatics, like Nike+ [6], Fish'n'Steps [8], and diabetes mobile apps [9]. Those tools in their area can help users to draw up their personalized plans that adapt and improve users' life.

However, these systems without interoperability and companies in a vertical market running the isolated business. They work independently. Collecting data used by themselves and stored data in access-controlled silos [10]. Those IoT solutions did not efficiently connect together. Besides, after used the data once, almost tools will throw their data away or just store it in independent data silos. We can image if we can share those be abandoned data that collected by different IoT tools, which indexes our movements, habits, preferences, life patterns, and resource consumption [1]. It can provide a considerable amount of valued information that can help us to improve our lives.

Meanwhile, personal data has more and more important social, economic and practical value [12]. It plays a vital role in many aspects of our everyday lives [16]. The European Union's general data protection laws and regulations changed the operation pattern of organizations in 2016 [13]. Organizations around the world are busy exploring people-centric personal data. The entire industry, such as energy, health and welfare or financial, has been impacted by this trend.

Under this trend, Poikola etc. [24] proposed MyData in personal data management that combines industry needs with digital human rights. They intend to provide a fundamental and can benefit widely framework to handle the challenges that come from as the data becoming the business resources meanwhile raised privacy problem and how efficiently utilize fragments of data.

2.1.2. Concept

MyData is a conceptual framework. It refers to human-centric ways of organizing personal data [12]. As the developing of the IoT technologies, personal data has become a more and more valued business resource. Meanwhile as the currently increasing public awareness on privacy problems and the challenges on how to effectively utilize data fragmentation. Besides, most of those collect process are happens behind users back. Users even do not know what kind of data collected nor the value of the data that they give away. To handle these challenges, MyData to bring the value of personal data for people and propose the individual manage and control of their own data.

In this model, individual users can granted to access their personal data with a machine-readable form. This makes users using their own data as they see fit. Those data can shared, exchanged or perhaps even sold to other services by the users or with his/her permission [12]. It can apply in different sectors, like public service and health [25]. Figure 1 illustrates the flow of data between sectors, applications or services people using, organizations that requiring personal data. It provides a good way to organize data logistics between data producers, which would enable collecting more data about individuals to help in personalizing and optimizing services and make informed decisions. Hence, the MyData-model can create entirely new business models. In fact, World Economic Forum has estimated that the utilization of personal data is going to be one of the most significant business trends [16].

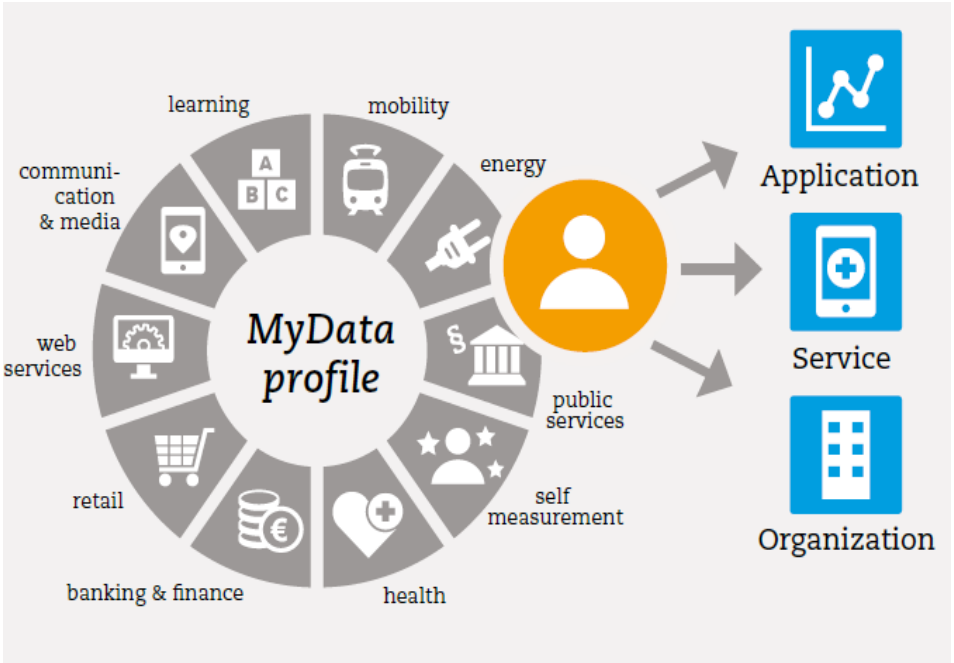

Figure 1. Personal data is across all sectors and can share selected parts [24].

2.1.3. Benefits

Personal data holds great potential for individuals, organizations and the entire society [12]. Databased feedback system can benefit the **individual** users in their daily life. Fing [37] proposed a MesInfos project, which shared the personal data between the organizations and customers. They find user can utilize their data among areas like management, control, life experience, decisions and action, contribution, conscience, self-knowledge, which presents in Figure 2. MyData-principles can make it easier to manage and utilize personal information collected from individuals. Increased the transparency when user access to data and customers can choose a more suitable way to utilize their data. Individuals can also enjoy the increased choice of services via data portability and the enhanced power when interacting with companies and public organizations who want to use their personal data.

For **organizations**, besides established companies can benefit from MyData but new business opportunities can also create. Under current businesses models, most of the personal data automatically gathered by service providers behind users [17]. This leading the unbalance between users and service providers that data producers cannot control the way their data used by the data consumers even have no power to act on the uploaded data. This issue undermining people's trust [16, 17]. The benefits using MyData-principle can gain greater trust from customers, which provides us more open business environment and reduce the transaction costs, easier ways to meet the privacy requirements. Besides, as the MyData-services would be compatible with each other makes it easier transfer the data and companies can draw benefits from customer data with a smaller user base.

MyData can have significant social effects [12]. MyData allows individuals making their decisions have a more environmentally friendly, fit and ethical choice. It would increase the positive impact on the **society** development. Besides, MyData would develop the human rights and information technologies together, and create a more sustainable basis data-driven technology. This can help us to mitigate the risk of monopolization when developing future society.

Figure 2. Utilize personal data among different areas [37].

2.1.4. Examples

MyData aim to enable the data flow for different sectors, it should be able utilized to different areas and these sectors can use the same personal data. To make MyData approach more clearly for readers, Poikola etc. [24] proposed few use cases that apply MyData, even some data types are specific to sectors, like clinical health data, but they believe the design of MyData can also applied in manage healthcare data to provide new sorts of services, and researchers can use it to create new research databases. The schematic (Figure 3, Figure 4) shows the flow of data, authorizations, money among different actors.

Use case 1: MyData and Occupational health

The occupational information included in the clinical data, public occupational health care data and individual's profile data are the Data Source in this use case. Occupational health provider is the Data Sink. Employers using the Data Sink's services by sending money to them. After authorized the consents from MyData Operator, Data Sink sends money to MyData operator to purchase occupational data. MyData operator sends money to account owners and send consent to Data Source and Data Sink. The Data Source receive the money and authorize data flows with consents. In this way, MyData provides the standardized and reliable manner for across organization data management between different professional and public health organizations and personal data source.

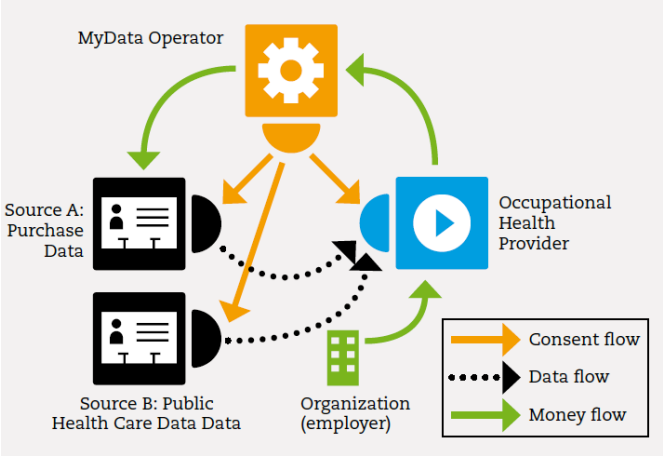

Figure 3. MyData and Occupational health [24].

Use case 2: MyData and research data banks

Recently, more and more individuals willing donate their personal data for research purpose, but integrating data from multiple sources it a big challenge in against privacy invasion. To handle this challenge, MyData can offer a universal framework for different sorts of research data banks to acquire consent to collect data. Then research data banks can avoid violating individual privacy rights when provide access to their data, also can retain the ability to cross-reference data.

In this use case, different kind of data set or individuals' profile data are the Data Source, the research data bank is the Data Sink. Public sector and researcher organizations provide financial assistance to help researchers using the data bank anonymization services. Research organization sends money to the data bank. After

data bank authorized the consents from the operator, it sends money to MyData operator to purchase data. MyData operator delivers the money to multiple Data Sources and sends consents out. Data Sources receive the money and authorize data flows with consents.

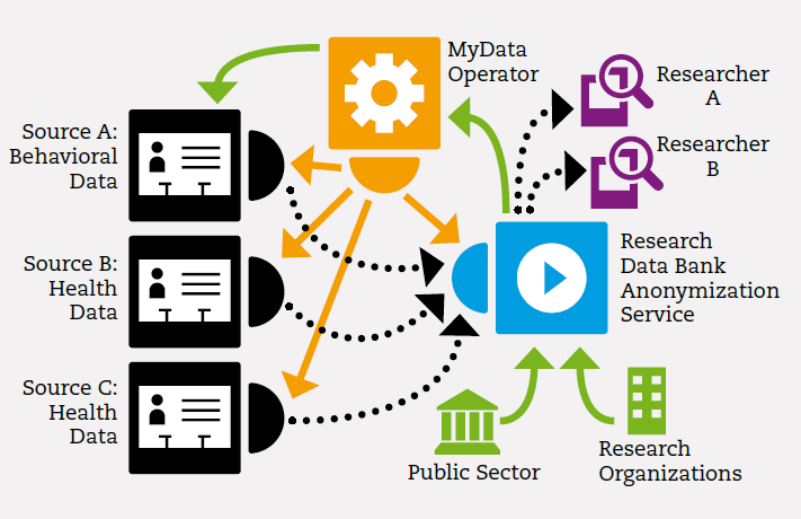

Figure 4. MyData and research data banks [24].

2.2. AWARE

AWARE [19] is an open-source framework that allows application developers, researchers and smartphone users collect hardware-, software-, and human-based data and can use AWARE plugins to transfer those data to a human-readable information. Researchers using that information can (re)construct the participant's context and study the participant's behaviors. AWARE's infrastructure (Figure 3) provides a variety of usages and benefits for individuals, researchers and application developers.

Individuals without any programming skills can use this application to record personal data by enable or disable sensors or plugins to visualize and contextualized information on the mobile phone. AWARE save the data collected locally and it does not log personal information like phone number or contacts information to enforce the privacy.

AWARE described as "an open-source effort to develop an extensible and reusable platform for capturing, inferring, and generating context on mobile devices" [18], it promises researchers the ability to capture the context of participants through their mobile phone and log their device usage. To fulfill this promise, the framework can activate a wide variety of smartphone sensors and collect data accordingly, like accelerometer data, network activity, and phone keyboard usages. Research also can publish study-specific or public plugins to collect research-required data when AWARE not record the data they need. To make a study easier, AWARE provide researchers such conveniently features like check participants' participation, using QR Code (Quick Response Code) or direct link enrolls participants, remotely trigger mobile ESM (Experience Sampling Method) questionnaires, label and group sets of participants' devices, remotely request or clear study data.

For developers, AWARE provides abundant of AWARE API (Application Programming Interface) and it can embed be an Android library to promote developing richer context-aware applications.

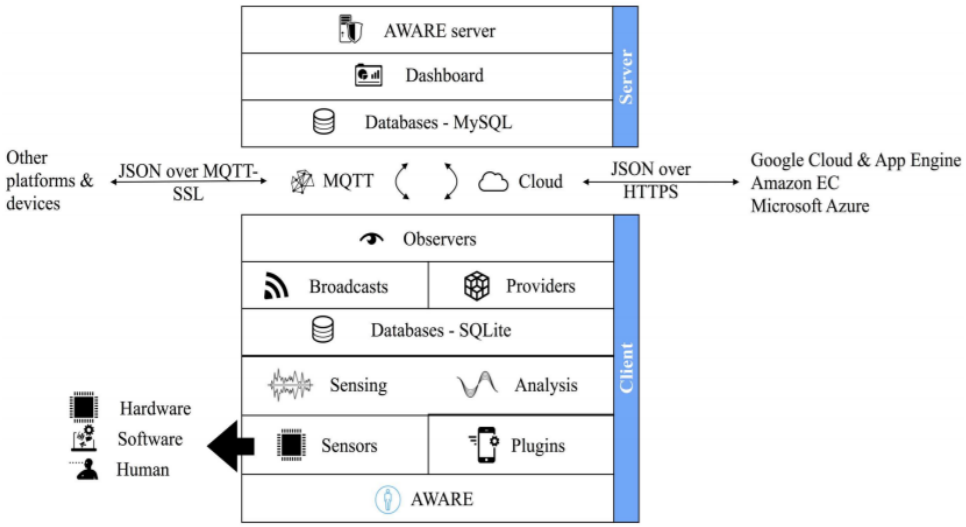

Figure 3. Overview of AWARE's infrastructure [18].

2.2.1. *Data Sharing*

The AWARE share collected context via three different strategies: broadcasts, observers and providers [19]. All of them are available for AWAREs' sensors, plugins and applications installed on the smart phone. Next, each strategy will introduce in detail.

Broadcasts used rapidly update other plugins, sensors, and applications of the users' context [18]. Using this lightest way, the application does not need to install AWARE to get notified of context. Users just need to use Android's BroadcastReceivers to broadcast current context, no necessary worry about the data captured with such context [19]. Multiple broadcasts can capture from sensors, plugins, and applications at the same time. There has one thing need to be noticed, as Android will interrupt broadcast with ANR (Android Not Responding) messages if it does not return under 15 seconds, so should make the work short inside onReceive() which implement in the context receiver side.

The sensors and plugins context data using **providers** to store. It achieved by Android's ContentProviders [19]. The data stored on the user's mobile phone locally or on a remote MySQL server. Thus, AWARE provide sensors and plugins two way to request (i.e., pull) the data, using Android's Cursor querying data in SQLite database file based on providers' unique content URIs (i.e., path) or using Observers to subscribe it. Besides, based on the AWARE's API and cloud services may also access and utilize the context data, for example, Google, Amazon, and Microsoft's cloud service.

When system required tracking the context and sensors change in real-time, AWARE provide **observers** to monitoring changes. It implemented as Android's ContentObservers and using message queue telemetry transport (MQTT) message callbacks to remotely sharing information to other devices. Observers offer active

(i.e., push) and event-driven access to the context. Observers enabling energy efficiency and provide real-time context labeling.

In AWARE, data packaged as JSON (JavaScript Object Notation) objects to send and replicate the users' data into MySQL database, to guarantee the privacy and security, data transfer via HTTPS (hypertext transfer protocol secure) protocol. To achieve better performance, MQTT applied when exchanging context messages between users' smart phone and other servers and devices in a publish/subscribe approach. It is designed for constrained devices and low-bandwidth, high-latency or unreliable networks [32], using minimize network bandwidth and device resource requirements to ensure reliability and assurance of delivery, when the device or server unreachable, the data would be queued locally waiting for delivery later.

2.2.2. *Related Frameworks*

Two frameworks, Paco [20] and Purple Robot [21], they both have a similar goal as AWARE: to establish a platform allows researchers to collect both sensor and human-contributed data based on the personal device of a study's participants. Paco, which stands for Personal Analytics Companion, promises its users the ability to build "Personal Science Experiments in minutes". It is actively developing and it works on both the iOS and Android platform. It provides usage of the application for both personal and research. Paco allows users to create a new experiment based on a simple to use web-interface. Through this interface, the researcher can create a signal schedule (either random or time-based), set trigger events and the possible user-inputs at each stage. Other data that can be collected is like device location (GPS), application and browser history, device information (e.g. the phone model), and a list of applications installed on the device.

Purple Robot is a full real-time sensor data-acquiring platform for collecting information about the user and their immediate surroundings. It is more similar to AWARE than Paco, in the sense that it requires the researcher to actively code and compile the desired application. Furthermore, Purple Robot provides full access to the Android sensor framework and other device information such as battery level, running software & apps, and hardware information, as well as providing integration with external data sources such as weather conditions, it implemented a full trigger framework used to defining conditions and providing rich interactions. In addition, researchers can use JavaScript API to construct their application, which compared to native Android code development probably easier to grasp for those without a software engineering background. However, the collection and immediate insight provided in the AWARE dashboard not implemented in this framework, therefore requiring extra effort from the researchers using this framework. Purple Robot is actively developed and similar to Paco and AWARE, is completely open source.

The overview shows in Table 1. This table based on the features in different frameworks, combined with the content of each framework's online documentation. It is an overview a completely truthful representation of reality. For completion, the Funf framework [22] included. Funf, aimed to offer an easy to use open sensing framework for mobile devices. Using the website of the framework, a research can create a sensing application by binding users' Dropbox account that without any coding. Researchers can distribute this application manually or through the Android market, and receive both the application and the research data in her personal

Dropbox folder. Funf also provides basic survey system for manual data collection and use probes which are the basic data collection objects used by this framework.

Table 1. Comparison between related frameworks

	AWARE	**Paco**	**Purple Robot**	**Funf**
Platform Support	Android/iOS	Android/iOS	Android	Android
Development Method	Writing JAVA code	Writing JAVA code	Writing JavaScript code	Writing JAVA code
Open source	Yes	Yes	Yes	Yes
Documentation	Detailed tutorials and technical documentation	Detailed tutorials and technical documentation	Necessary technical documentation	Necessary technical documentation
Access to activity data	Full access to the Android sensor framework, device information, Personalized behavior	Personalized behavior	Full access to the Android sensor framework, device information	Full access to the Android sensor framework, device information
Data export	Dashboard	Dashboard	External server	Email,bluetooth, or Dropbox
External Data	Weather conditions	No	Solar timing and weather	No

2.3. Blockchain

Since bitcoin was presented in a white paper by Satoshi Nakamoto [28], this digital cryptocurrency system inspiring people's interest in blockchain technology continuously. Modern cryptocurrency general structure contains three layers are blockchain, protocol, and currency. Each coin such as Bitcoin, Litecoin, Dogecoin typically both a currency and a protocol, and it may create its own blockchain or apply the Bitcoin blockchain. Blockchain is the underlying innovation, which enables seamless transactions between parties to make cryptocoins work and success. The blockchain is a decentralized transparent ledger with the transaction information that shared among all peer-to-peer network nodes but owned and controlled by no one [26]. More importantly, data entered to a blockchain is immutable: It cannot altered afterward.

The data stored in Bitcoin blockchain are the transactions information collected by miners, these data will put into blocks, which distributed among multiple nodes and all of them uses cryptographic proof to validate transactions. Generally, a block contains information such as previous block reference, summary of transactions (root

hash), timestamp, and proof of work (nonce) show as Figure 4. Blocks chained by reference to the previous block, the set of transactions included in a block happen in the same period. When generating a new block, it will be timestamped and the store the timestamp in the blockhead, thus the timeline in the blockchain network defined by this chain of references, the earlier transactions in the previous blocks. Users can follow the reference back to check every block until the first block knows as genesis block. Transactions placed in blocks and ordered in blockchain solved the double-spending problem [28] caused by transaction generated randomly. The blockchain allows no inconsistencies exist in the system, each transaction before entered are necessarily valid and verified by all nodes in the network, based on this checking and confirming methods, an attacker require controls at least 50% of the network's computing power can reverse transactions which called 51% problem [28], it ensured effectively self-regulation and fully secure. This enables users can trust this public ledger system stored worldwide on large amounts of distributed nodes maintained by "miner-accountants", single storage point database for the data and/or a trust third party (like a bank) to validate transactions can be replaced [26, 28].

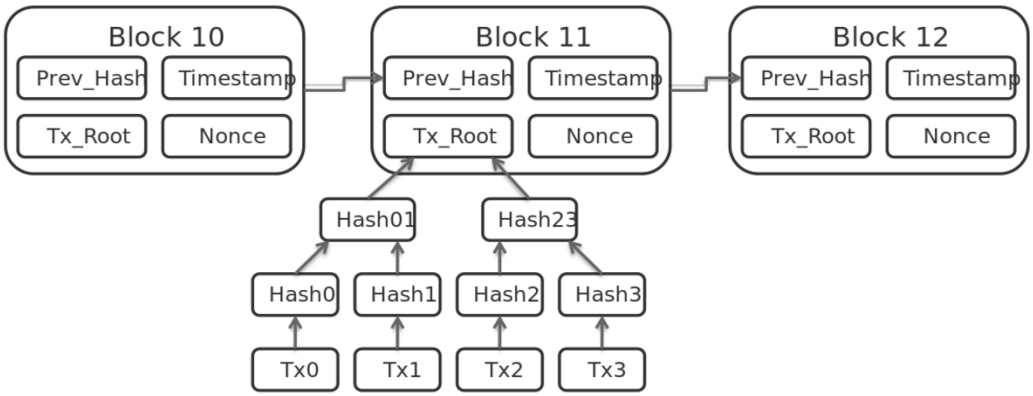

Figure 4. Graphic of data fields in Bitcoin block chain [36].

Thus, blockchain owns the potential to change the way transfer value on a global basis fundamentally [29]. Blockchain already be the cash for internet, and it can become "Internet of Money" and also "Internet of Trust", similar with Internet of Things, it connecting finances in the way that IoTs connects machines [26, 33], Babitt and Pilkington believes this technology creating a new type of economy: crypto-economy [30, 31]. Margaret [35] discussed the potential of blockchain to disrupt even replace the European Central Bank's Target2Securities project, a platform aims to standardize European cross-border trade settlement. Besides, The World Economic Forum [27] identified blockchain as one of six computing "mega-trends" over the coming decades. They believe blockchain can provide trust for financial, contract and voting activities. Applying blockchain can gain obviously benefits such as reducing credit card payment fees, avoid waiting days for receive funds, it completely redefines the current way for currency, transaction, and commerce.

Cryptocoins just are the first step of the future crypto-economy, now the other users like smart contracts have been developed [32]. It is a computable contract that does payouts between parties, without involving the third party. These smart contracts as code which tamper resistant are stored in the blockchain as "self-executing contractual states", it provides real-time track performance and can be triggered immediately by predefined conditions, can replace costly human error and

avoid the risk of based on third party's commitments [27]. Oracles [34] offers a simple way to connect to a smart contract from the off-chain world, providing the data needed to ensure performance and make valued payments. The smart contract deeply introduced in the section 2.3.2.

2.3.1. Background

Blockchain considered developed on numerous technologies. Major technologies such as peer-to-peer network, hash functions, public-private key cryptography and cryptographic signatures, Proof of Work outlined below.

Peer-to-Peer Networks

Peer-to-Peer network is a distributed application architecture, which shares tasks and workloads between all participating nodes so called peers [41]. Different with the client-server model that divided and fixed to resource providers (server) and service requesters (client), P2P network nodes not fixed as a server or client, peers are both acts as suppliers and consumers of resources [40].

Search and data transmission are necessary need to consider when applying a P2P network [39]. Search methods mean to manage nodes and data locations, and representative examples like installation index server, and conducting management super node. Data transmission methods mean data transmitting among nodes. It divided into direct transmission and related transmission through another node. P2P technology providing a complete distributed network and eliminate the single point of failure in the blockchain, and all normal nodes in the P2P network of blockchain and miner nodes have the same data.

Hash Function

Hash function is a mathematical algorithm, its output called hash values, hashes, digests, or hash codes [38]. It can be used to map arbitrary size of data to a fixed size string, and the same data value can always obtain the same hash, a slight different between the input data can results completely differences between hash codes. In addition, as designed to be a one-way function, to recreate the input data by knowing the hash value is extremely difficult. In blockchain, this is used for guarantee integrity of transmitted data and applied in Proof of Work to prevent data falsification when creating new blocks.

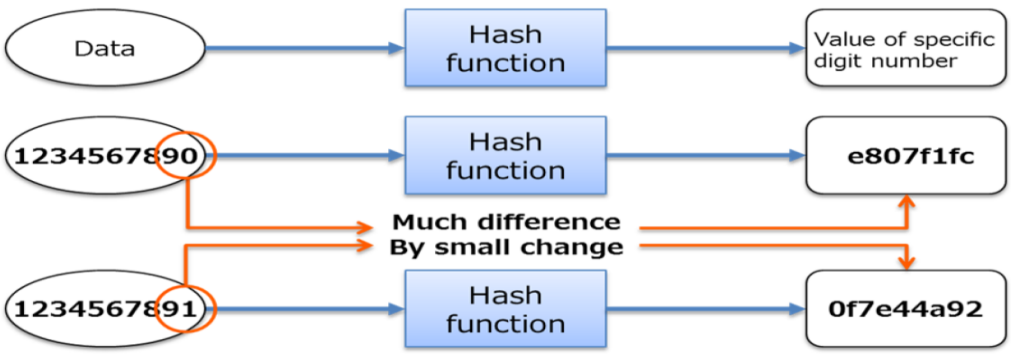

Figure 5. Mechanism of hash function [39].

Public-Privacy Key Cryptography & Cryptographic Signature

Blockchain uses cryptographic proof to replace trust third party for processing transaction online between parties. Each transaction sent to the receiver digitally signed by sender's "private key". The receiver receiving the transaction and verifies the digital signature by using sender's "public key", bitcoin address as the public key in Bitcoin [43]. When involved spend money, this used to verify the ownership of the currency.

Public key cryptography also called asymmetric cryptography is a cryptographic method using pairs of keys for encryption and decryption. The public keys which available for anyone used in authentication function to verify that a message sent by the holder of the paired private key, and the private keys known only to one person used in encryption function to decrypt the message encrypted with the public key [42]. Receiver required prepares both private key and public key enables security of the data transmission, and contrast to symmetric-key cryptography using one key for decryption and encryption which delivering the key required various safety measures, public-key cryptography advanced in delivers the public key to the users. Receiver through properly manages private key to maintain the safety.

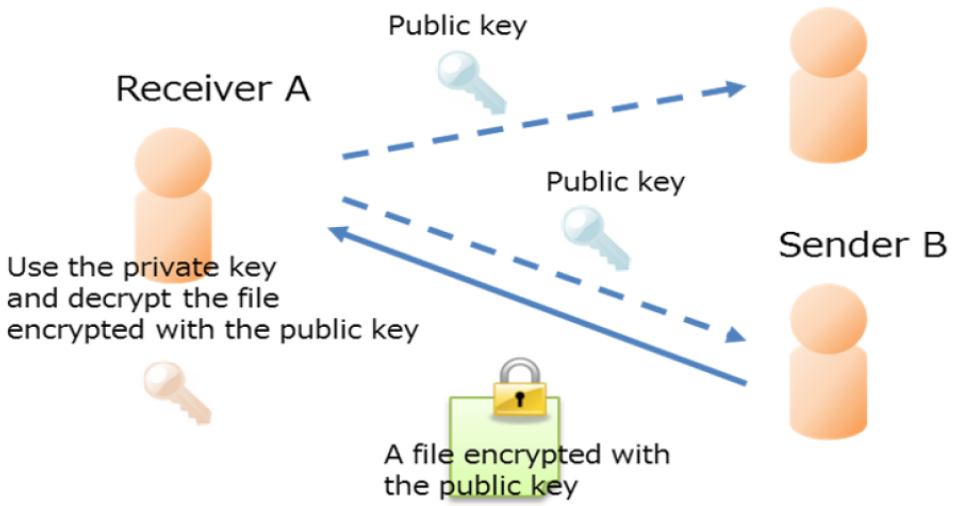

Figure 6. Mechanism of Public-Key cryptography [39].

Digital signature refers to a mathematical scheme for proving the authenticity of the digital messages or files. It offers a recipient reason for the receiver to believe the message sent by known sender [44]. It commonly applied in cryptographic protocol suites. Generally, it made by encrypting the file to a hash value with sender's private key and sent hash value and the file together. After received the message, the receiver decrypts the sender's digital signature with sender's public key to get the hash value and file contained in the message, and obtain the hash of the file through same hash function as sender used, cross-checks these two hashes to verify the sender's digital signature.

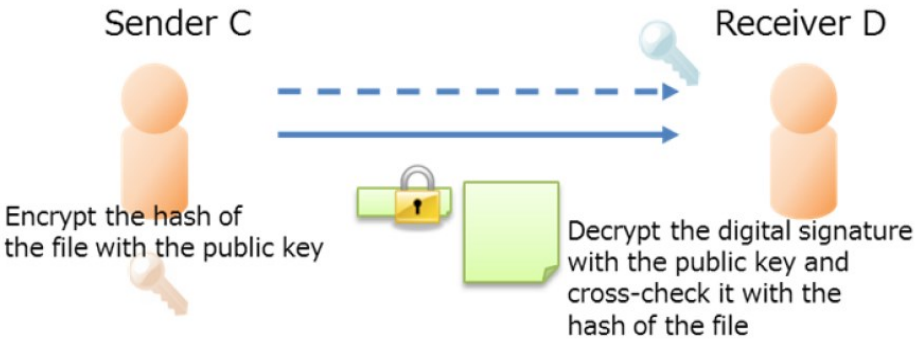

Figure 7. Mechanism of digital signature [39].

Merkle Tree

As mentioned above, the blocks contain the summary of the transactions included, which called Merkle Root Hash in Bitcoin [28]. Applying Merkle tree avoided store the interior transaction hashes, it highly saves the disk space, but also allow transaction fast verification. Merkle or Hash tree is a tree that leaves are hash value and non-leaf nodes labeled with the hash labels of its child nodes. It is a generalization of hash chains and hash lists. For example, in Figure 8 hash 01 is the result of hashing its child nodes hash 0 and hash 1. It enables efficient and secure verification of large data structures' contents [45]. To verify a leaf node is included in the given hash tree just need computing proportional to log2N times, N is the number of nodes of the tree.

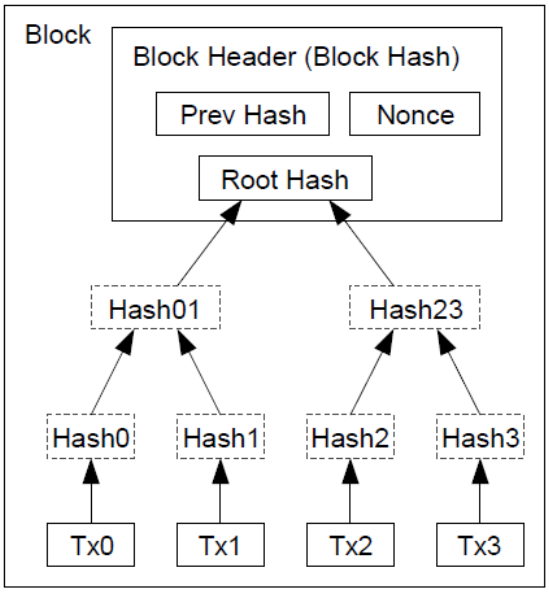

Figure 8. Example of a binary hash tree [28].

Consensus Mechanism

To achieve that all entries in blockchain are valid and consistent, blockchain network asked to make a consensus about the blocks in a decentralized way. Consensus mechanism generally refers method of action for the decentralized network to decide on make significant changes [46]. In blockchain, the most prominently used

consensus algorithm are Proof of Work and Proof of Stack, those consensus mechanisms and other mechanisms described as follow.

Proof of Work

PoW (proof-of-work) was introduced by Bitcoin blockchain [28], it generally refers a method to confirm a person's innocence (or courage right action) by requires do a certain work. The work is moderately hard (but feasible) on the worker side but easy to verify for the others. In Bitcoin blockchain, when adding new block to the blockchain, a miner requires to solve a computationally difficulty problem that given by the previous block, and when block created success it cannot be changed unless regenerating all successors and redoing the works they contain [47]. This problem can be calculated by all miners in the network at the same time, but only the first miner who solved the problem (find the nonce) have the right create a new block. And blockchain is designed one new block will find every ten minutes with the unpredictable miner regardless of how hard the computation required. In another word, the difficulty of the PoW updates to require more rigorous conditions for future blocks. Which protects the blockchain against greedy miners' attacks.

As mentioned above, each blockhead contains a nonce, which is a short string of arbitrary number. For generating a new block, the miner is required to complete a PoW which use nonce to vary the output of SHA-256 hash of the block results the certain number of leading zeros. As introduced before, the hashes are results of the one-way function. There is no easy way to find a right nonce to generate hash correct. Simply computing randomly times is the only know way find a suitable nonce to end the work. For a simple example, Figure 9 shows nonce attached to input "Proof of Work" and the resulting SHA-256 function output, each phrase attached a nonce can results completely different output but if user knows the nonce, the work can be verified by once hash operate.

Proof of Work0 → dda386f573065923a93196dad9a51d8088741af6c6ef0658c0c94b8f431e565f
Proof of Work1 → f93d801465aae4d284f457b20959a694aa51ec23c447d5b5d69e2c62a6605d65
Proof of Work2 → 9cd5ec83b99d4821e0d3dae2a18e0474dc237e93ad3f55d8ba2633437942b6b7
Proof of Work3 → 88dac60acd7ca9580d881153a34cee56aefa3da9282784d0250d2d55034f564b
Proof of Work4 → ce4b109a3f3623229bfc7b4306bcf1106ec1e3510d763fb83d97ef3d06cb7641
Proof of Work5 → 57415534158c49973e4ffd2e5f4249794c01f14d9d85326ee9abc6df8d05c8a7

Figure 9. Outputs of SHA-256 hash inputs attached a nonce.

To achieve consensus in a distributed computing systems need enable Byzantine fault tolerance. Byzantine general's problem [48] develops around a hypothetical general must communicate his decision that to attack or retreat with other generals in the army and make a consensus. A given number of these actors are traitors. Traitors cannot be relied upon to properly communicate orders, for worse, those traitors may actively alter information during the transfer the messages attempt to subvert the decision [52]. The main idea to solve this problem is that make the processors submitting message "expensive". Thus, Bitcoin proof-of-work blockchain viewed as a practical solution to the Byzantine General's Problem and reach a consistent global view of the system state. The chain owns greatest proof of work accepts as the valid one can represent the majority decision [28]. Under the condition that majority of CPU power controlled by honest nodes. The malicious attacker to create a fraudulent

transaction by modifying a past block becomes very difficult. As it has not only to recreate that block by redo the proof of work, but it requires to redo proof of work for all subsequent blocks and race against with all honest nodes to generate a newest block to make the network accept its transaction and block are valid. As Satoshi says that if the honest nodes outnumber the bad actors, the Byzantine General's Problem is solved [26]. But it is still not perfect, Juan etc. [49] observe the proof of work mechanism used in bitcoin falls short of solving Byzantine General's Problem as it under a simplifying assumption that the attackers are computationally limited. And Eyal and Sirer described selfish mining attacks which require less than 50% mining power can also execute attacks on the bitcoin blockchain [50], Ayelet etc. extended the underlying model of selfish mining [51].

Proof of Stake

As PoW require miner solve the computationally expensive problem, this makes its throughput not scaling well [54]. Now mining in Bitcoin requires more equipment, electricity, optional hosting cost. Bitcoin network expends massive computation power but presently only handles 7 transactions per second. Other centralized payment systems like Papal handled around 144 transactions per second in early 2016, and VISA handles 2000 transactions per second, we can easily find the demand from practice applications is at least two orders of magnitude more [53].

The most important alternative consensus algorithm called PoS (proof-of-stake). It proposed a "virtual mining". The chance of creating a new block in PoS blockchain depends on network nodes' coins. Different with PoW based on scarcity of computer hardware to prevent Sybil attacks, PoS based on coins inside the network itself [55]. For example, PoW user can take $1000 buy a mining computer become a miner, then participating the network and producing blocks to getting rewards, PoS user can take $1000 buy coins in blockchain, deposit coins in PoS mechanism, those coin holders regard as voters. Given the latest block in blockchain, the PoS algorithm would randomly select one voter and assign him/her authority to generate next block, the voter creates block successfully can get rewards, if that voter failed to create a block within a period, the algorithm can select secondary voter that can create the block instead. Much like the PoW, who owns more computation power would gain higher probability to create a new block, in PoS, the randomness weighted by coin holders' deposit size, a voter own 100 coins has 10 times the chance of a voter with 10 coins. In this way, similar in PoW, the chain with the highest collateral is valid [55].

PoS algorithm has some major benefits, the first one is avoid consume massive energy costs, make a more sustainable and greener blockchain, furthermore, enables scalability. Secondly, it is potentially safer as possibly reduced vulnerability to selfish-mining and makes attacks more expensive than PoW. Thirdly, without solving a computationally expensive problem, it is highly enhanced the ability to add new blocks, allow to handle more transactions per second and reduced centralization risks, as in PoS, the problem of economies of scale is solved.

The key problem of PoS is "nothing at stake" [55]. Many early PoS blockchain, including Peercoin, producing blocks only get rewards but no penalties. In that case, if there have multiple competing chains, the voter can vote on all blockchain histories at once (Figure 10), unlike in PoW would splitting computing power in half results not be lucrative. In PoS voter doing so have nothing to lose but get more rewards, which results to consensus never resolving. This is the core of "nothing at stake".

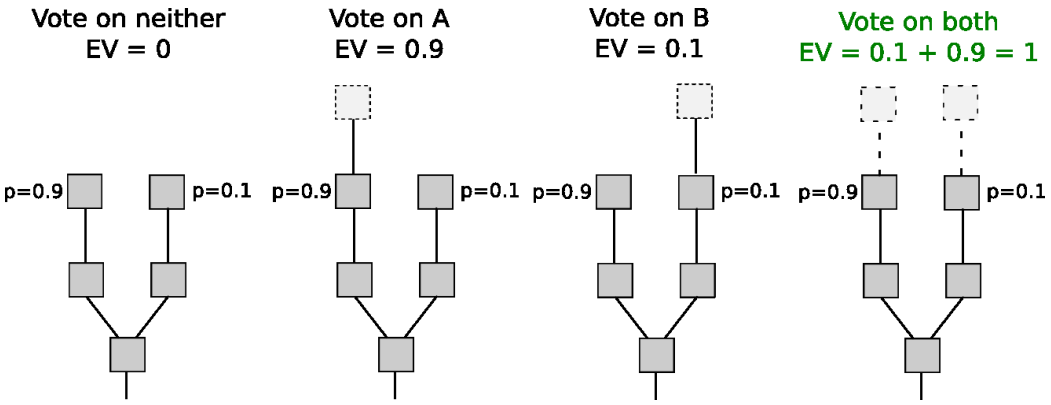

Figure 10. Core of Nothing at Stake [55].

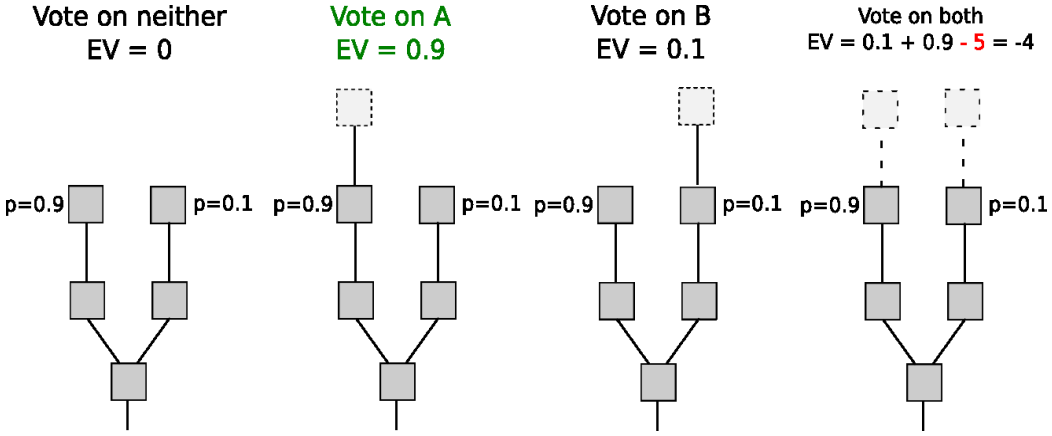

Figure 11. Slasher 1.0 for Nothing at Stake [55].

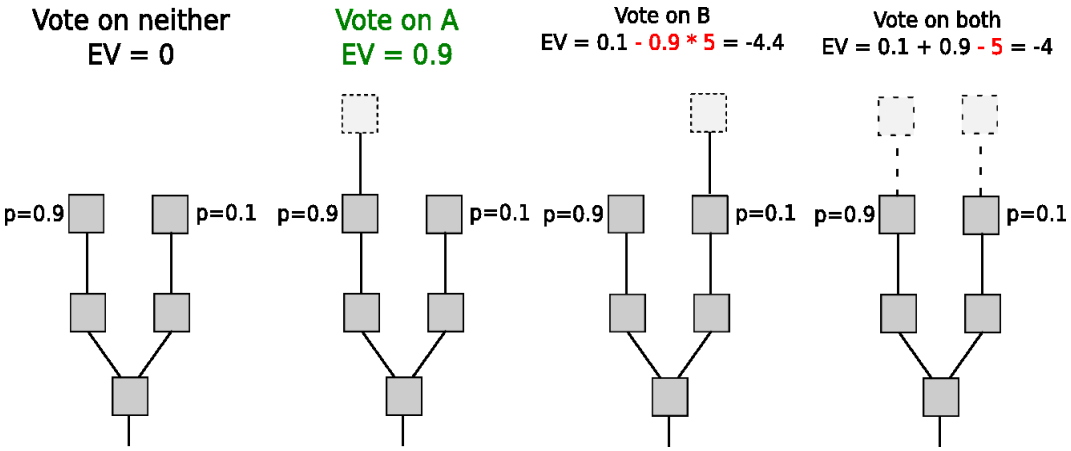

Figure 12. Slasher 2.0 for Nothing at Stake [55].

To solve this problem, Vitalik [57] proposed a strategy called “Slasher” which combine PoS with PoW to penalize voters if they simultaneously general blocks on multiple chains. It would enter the misbehavior into blockchain as a later timepoint and the malicious voter’s deposit would be deducted at that point (Figure 11), but in this algorithm, the voter set needs to be confirmed well ahead of time. Vitalik’s work developed further by Iddo Bentov [56], so called “Slasher 2.0” (Figure 12), this strategy is to simply punish voter for vote on the wrong chain, which without the

necessary of voters be known ahead of time but imposes more risk on voters than PoW by making the penalties explicit.

Other Mechanisms

There also existing other consensus mechanisms. Some based on BTF (Byzantine Fault Tolerant) protocols [60] trying to improve consensus efficiency and scalability or combine the PoW and BFT to enhance blockchain security and performance [58]. Proof of Burn (PoB) [61] as an alternative algorithm to PoW and PoS by proof burned coins, in which mechanism users required send some coins to an unspendable address. However, all existing PoB based cryptocurrencies work rely on burning coins from other PoW blockchains so they cannot stand alone. PoET (proof-of-elapsed-time) [59] proposed by Intel recently, which run in Intel's SGX a trusted execution environment no need cost massive energy and specialized hardware, it achieves consensus based on guaranteed wait time provided by SGX.

2.3.2. Ethereum

Satoshi Nakamoto [28] not limit the blockchain technology to Bitcoin or other cryptocurrencies, in his original plan proposed three steps. The first is the public transaction ledger, as we know the blockchain, the second step is the transaction system used to transfer valve between parties without trust third party which known as Bitcoin protocol. These two steps have achieved in bitcoin 1.0. As we see now, the implementation blockchain 1.0 for currency and payment transactions works well, but in more complicated blockchain 2.0 applications, more complex assets such as property and smart contracts need to record and transfer. Thus, we need to go ahead achieve the third step, a robust scripting system that Turing completeness allows run any protocol, coin, or blockchain. Nakamoto envisioned is a full feature set to enable a programmable money but not only transfer money between parties. There has a framework aiming to provide a blockchain with a built-in full-fledged Turing complete scripting language is Ethereum [26].

Ethereum is a blockchain platform for building and publishing decentralized applications, which is a Turing complete machine that can run any script, coin, or cryptocurrency project [62]. Ethereum is relying on Proof of Work (PoW) blockchain at present, it will implement PoS consensus mechanism called Casper in the future version [64]. Ethereum provides a fundamental underlying infrastructure platform to building decentralized applications especially for that rapid development time, security for small and rarely used applications, and applications that required efficiently interact with different applications. Ethereum intends to deliver not only a digital currency, more important is to allows the execution of "smart contracts", based on its blockchain and Turing complete programming language which can enable for users to create own rules for transaction formats, ownership and state transition, allow anyone create and use any of decentralized applications run on this blockchain platform.

Ethereum blockchain and Bitcoin blockchain similar in many ways, but the main difference is the blockchain architecture. For overcome the limitation of Bitcoin light client, Ethereum uses several previous transactions to determine the effect of a transaction which used to handle more complex applications, it contributed the Merkle tree one step further. Ethereum block headers (Figure 13) contain not just one Merkle tree, it contains three trees of the transaction list, the most recent state and the

transactions' receipts, which are binary data, represent the effect of each transaction. This highly enhanced light client's ability to get verifiable answers for many queries like account exist or not, the current balance of my account, has a particular transaction been entered in the particular block etc. [65].

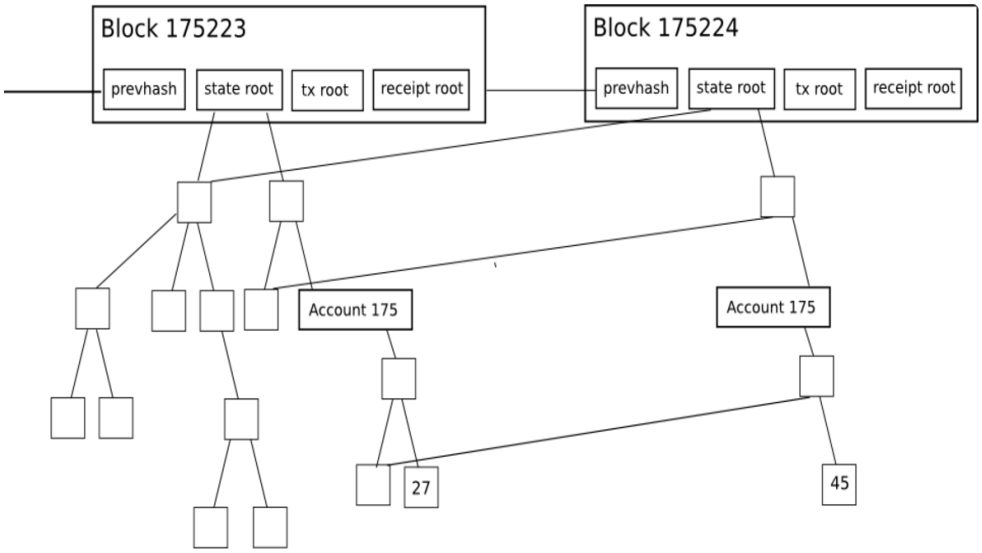

Figure 13. Block header in Ethereum [65].

"Smart Contracts" that was proposed in the year 1994 by Nick Szabo [15]. It designed for automatically execute contracts between parties. However, this great idea did not get well popularization until the cryptocurrencies and programmable payment invented. Before to discuss the Ethereum smart contract, the related concepts within Ethereum introduce first, those concepts briefly explained what the smart contracts' run environment is, how it works and what can we do with smart contract.

Ethereum Virtual Machine

As a programmable blockchain, users' activity in Ethereum not fixed within pre-defined operations, the complexity operations can created by their own. This achieves by executing code that with arbitrary algorithmic complexity in the EVM (Ethereum Virtual Machine) [63] which is the runtime environment for smart contracts. EVM is isolated, running code inside it without access to filesystem, network or other processes. Even for smart contracts also have limited access to each other. Ethereum is Turing complete, provides users developing applications the chance using programming languages, which modeled on languages like JavaScript and Python. EVM offers developers compiler to compile smart contracts written in high-level language (typically Solidity, Serpent) into byte code and upload to the blockchain by client, the EVM bytecode format is the way that smart contracts exists on the Ethereum blockchain [14].

EVM runs and executes the same instruction in every node in Ethereum peer-to-peer network, which described as "world computer". The computation on this massive parallelization Ethereum network not efficient but becomes more expensive and slower than on a normally "computer". However, EVM runs on every Ethereum node maintains the blockchain consensus. The decentralized consensus across the blockchain ensures zero downtime and allow entered data unchangeable and censorship-resistant.

Transaction and Message

Transaction in Ethereum refers to the signed data package which stores a message from EOA sent to another Ethereum account. It contains a set of data such like recipient (address), signature to identify the sender (address), value amount (ether), optional data field etc. The data field can contain the message sent to a contract or even contain the contract that user wants to create.

Essentially, the message is like the transaction except it generate from contract account but not from EOA, contracts using "sending messages" to connect each other, a message also can activate the recipient account executing its code, which makes contracts can interact with other contracts as EOAs can. Message is virtual object also contains some quantity of ether, a byte-array of data of any size, the sender and recipient's addresses, which never serialized and only exist in the EVM. Send messages regarded as function calls in Ethereum network, it produced when a contract executes the "CALL" or "DELEGATECALL" opcodes. After one contract receives a message, it enables returning some data for the message sender immediately use which are exactly the way calling a function.

Ethash Algorithm

Ethereum running a PoW algorithm called Ethash (modified Dagger-Hashimoto algorithm) now [63], it is similar to other PoW algorithms that require finding a nonce to solve a computational problem. Its difficulty dynamically changes rely on the average one block generated by Ethereum network every 15 seconds. To avoid the situation that like Bitcoin mining dominated by specialized hardware and provide a more decentralized distribution of security, Ethereum chose a memory-hard computational problem to make it ASIC (Application-specific integrated circuit) resistant, in this case, memory requires as well as CPU, the general computer can gain better performance.

Ethash PoW involves choosing random subsets of fixed resource to rely on the nonce and block header and hashing them together. The resource DAG (directed acyclic graph) represented in the file as a matrix, which created every epoch (roughly 30000 blocks per 5.2 days) using a version of the Dagger-Hashimoto Algorithm combining Vitalik Buterin's Dagger algorithm [67] and Thaddeus Dryja's Hashimoto algorithm [66]. The seed to generate DGA only depends on the block height, so DGA can pre-created to avoid wait a long time for each epoch transition. Pre-generate and cache DAGs ahead of time require large memory as the update once 30000 blocks and datasets grow linearly with time, makes most miners' work will be reading dataset, but not alter it. Which also means the large-scale miners get super-linear benefit hardly. The high bandwidth requirement makes sharing the memory among many super-fast processing units earns little benefit than a single unit. This result the mining pool have no reason to exist in Ethereum platform unlike Bitcoin, which discourage centralization.

Ethereum Accounts

In contrast Bitcoin consist of a list of transactions, the basic unit of Ethereum is the account, which is the innovation of the Ethereum. It tracks every account's state. Transfer value and information results in state transitions. The two type of accounts

in Ethereum are externally owned accounts (EOA) and contract accounts [63]. EOA controlled by private keys and another controlled by its contract code and only EOAs and other contract accounts can “fire” it.

In Ethereum, EOA can used to send three kinds of transaction inside the network. Firstly, the basic usage is transfer value (ether) to other EOAs or smart contracts; secondly, send transaction to invoke the functions inside the smart contracts; thirdly, the most important is allow users send transaction to create new contract.

Contract account in Ethereum is a contract also has its own balance and has associated code. It allows pass messages between themselves which is the only way to trigger the code execution except for sent transaction to it, contract able to perform arbitrary complexity operations as Turing completeness, such like managing its persistent storage, call other contracts, and own permanent state etc.

The basic difference between them is human users control the private key to controlling EOAs, but contract accounts managed by internal code. Which can just triggered by transactions sent from EOA or triggered by messages received from other contracts. For contract accounts, Ethereum asks nodes enable agree on the result of computation, which achieves by strictly deterministic execution. Thus, contract accounts cannot perform operations when without instructions from EOA or other contract accounts. Every little step prompted by transactions or messages. Ethereum charge transaction fees like Bitcoin, which used to protect its blockchain from the malicious attacks. That means sender of a transaction must pay for each step of the “program” they activated, including computation and memory storage. We can find the cost fee for each instruction on the Ethereum [63]. Which can prevent the DDos attacks or infinite loops effectively.

Gas

The gas is the special unit for running a transaction or contract in Ethereum. It is to decouple the unit of Ether and its market value from the unit to measure computational use. Every operation that performed by a transaction or contract on the Ethereum network costs a fixed number of gas, operations which require more computational resources cost more than operations which require few resources. For example, calculate SHA3 cryptographic hash once take 30 gas, plus a cost of 6 more gas for hash every 256 bits of data. The price of gas can be increased or decreased according to miners’ decision. Which avoiding increase in the price of Ether would cause change all the need of gas use.

Code Execution

The contract code in Ethereum written in a low-level, stack-based bytecode language refers as EVM bytecode. It uses a series of bytes to represents the operations. The operations provided store data in three types of space, the first is stack, which designed as a last-in-first-out container, can push and pop values. Another is auxiliary memory. It is a byte array can infinitely expand. The last one is the contract’s long-term key/value storage, unlike the mentioned above, which reset after computation ends, for long-term storage. It can also get the value, sender’s address, incoming message data, and return a byte array as output.

EVM code has simply execution model when EVM is running, its full computational state can be defined by the tuple which consists of block-state, code, transaction, message, stack, memory, gas, program counter. The block-state is the global state contains all accounts and include balances and storage. When execution

start, based on the program counter to find current instruction and each instruction has predefined how it affects the tuple. For example, ADD consumes three gas and increments program counter by one, which pops two items out of stack and pushes their sum in. Ethereum Blockchain built-in Turing complete programming language, which allows achieve a bare-bones version of Namecion in two lines of code, and built other protocols like currencies and reputation systems under twenty, a basic implementation of Ethereum in few hundred lines of code.

Integrates the concepts above, we can know that Ethereum smart contract created by EOAs send transaction which involved mining processes, associated with contract account, exits on the blockchain with EVM bytecode format and running its functions inside the EVM on each node, and only the transactions and messages can trigger its code execution. Each operates in contract much cost fee that pays by Ether the coin used within Ethereum. In a word, a contract is a collection of code and data exist at a specific address on the Ethereum blockchain with an Ethereum-specific binary format [63].

Smart contract regard as cryptographic "boxes" which contain value and only can be open if certain conditions are met [62]. It allows read/write to its own internal storage that is a database mapping 32-byte keys to 32-byte values, read the received message's storage, sends messages out to other contracts to trigger their code execution. Once the contract receives a transaction or message, it executes internal code as instructed by input parameters included in the transaction or message inside the Ethereum Virtual Machine. If no user or contract trigger the action, the Ethereum execution environment would keep everything as the same, and every account remains the same state, but when contract in turn activates, it would automatically run its code.

In general, the contracts serve for four purpose:

- Maintain a data storage stand for something that useful to other contracts or the outside world. For instance, the contract simulates a currency or contract records membership organization.
- Serve as kind of EOA but access policy is more complicated. Which called "forwarding contract", usually involves under certain conditions to resend incoming messages to those desired destinations. For example, the simplest conditions are requiring private keys to confirm the message before resending.
- Manage ongoing contract or multiple users' relationship. The instances include financial contracts, insurance and so on, it also used to create open contract that any other party can engage with whenever they want.
- Provide functions to other accounts, acting as software library in Ethereum blockchain network.

There have three kinds of application exist on the network at present. Which can classify three categories. The financial applications are major. It allows users to managing and entering contracts using money with more powerful ways. This includes sub-currencies, hedging contracts, wills, financial derivatives, savings wallets etc. Second category is the semi-financial applications, where involved money but still have a heavy non-money side to do. The self-enforcing reward for solving the computational problems. Peer-to-peer gambling can be good examples. The last category exists on Ethereum now, are not financial related applications like online voting and decentralized governance, decentralized File Storage.

3. PROYOTYPE DESIGN & IMPLEMENTATION

This chapter introduces Ethereum blockchain and smart contract based health-related MyData system in detail. It contains different layers like Node.js server, Ethereum smart contract and mobile client. This system combined the AWARE as a library that allows offering various data to meet the requirements from different parties. In the following sections, scenario introduced first, then specific the system requirements and describe the system architecture, the implementation of the whole system is the last part.

3.1. Scenario

Target audience

In modern medicine, the time for doctor determines the patient's condition is very short and have such limits to observe the situations and direct patients' activities after the patients left their sight. The staffs within the health organizations like hospital, health insurance, health care administration, medical rehabilitation unit etc. Those people desiring of patients' personal data, especially real-time and long lasting data for analysis and research. Meanwhile, there are some people willing to selling their personal data to gain benefits from those activities which including the patients who would like to report their situation to doctor to get correct instructions and useful feedbacks. This thesis implemented a demo system called ETHEREUM-AWARE that target audiences are those people. Furthermore, the target audience is generally people not experienced in the development of software, and just expected a basic understanding of blockchain among users, although they might not have used this technology in their life before.

To make a concrete case, a persona and a scenario introduced, in which this fictitious person looking for a solution that can get lasting real-time personal data support in a recovery healing case. Lucas is a physical therapist that providing treatment to people whose abilities to move and perform functional activities limited by the illnesses, injuries or environmental factors. Just like others, Lucas hard to monitor his patients' conditions in real-time after they go home. In addition, he currently working on a research that wonders to know whether his patients with leg injury walks 30 mins per day after dinner would get any positive effect in the treatment, which requires collecting trusty data from patients. Even Lucas as a heavy user of today's digital technology, like contact friends with social media, pay stuff by Bitcoin wallet. Despite his heavy reliance on technology in life, he has limited background with digital technology.

Scenario

For the completion of his research, Lucas requires long lasting personal data from his patients. Lucas expect his patients' privacy and security can be protected when running this research and the data can be easily transferred and apply in other researches, and to make the win-win situation, Lucas willing to pay for these data as incentive to encourage patients to provide trusty real-time data for a long period to support his research. Following this thinking, Lucas investigates various possible options exiting that meet his requirements that allow individuals to collect and sell

their personal data by themselves and to using security highly promised mechanism to host the trades between them.

After discovering the ETHEREUM-AWARE, Lucas found this is what he wants, and as he has experience in Bitcoin, he understands the concept of smart contract quickly. Then he introduced this system to his patients who willing to participate this research. After the system installed on patients' phones, they easily create a new data producer account and done the authorizations. Lucas also creates his data consumer account, transfer enough coins in Ethereum and decide to buy one-month data from their patients. Lucas can get data from his own computer when participants running the system and confirm this transaction, and the data with a human readable format provide by AWARE. Lucas satisfied these data, and patients rewarding and well recovered with Lucas's treatments.

3.2. System Requirements

The system aims to combine MyData and Ethereum Smart Contract to provide cost-efficient and stable services to support better data self-management and storage. There formulate a list of requirements for the system.

Scalability: The scalability of this system including two parts. The first part is the data type which produced by individual users. To reach more complicated research requirements, this system requires providing several of data with a general format that can easily use and migrate. Thus, easily expand more types of data, which in human readable format such as end device sensors are necessary. The second part is the data sets produced by all end users. Those end users gather together to act as a distributed database network. To achieve the goal of efficiently utilize and manage the data fragments, the network required able to generate and handle enough amount data sets. Which requires this system can provide the convenience for new end users join the network and can allow them to stand as an isolated personal database stored different data.

Security: Security is the most important requirements in this system. It involves all processes during collect, store, sells users' personal data and dispatches the rewards. The system required to ensure all users' data and rewards well protected.

Consent: Consent for a system involved personal information is necessary, and it becomes more desired for users. It usually refers individual notifications for possible uses of data when starting data collection or signing up for a service. In generally, users will get explicit notice. However, in the current notice model, individuals usually must seek their consent which written by legal term and hide in long privacy policies. In this way, users often treat consent as a tick-box exercise. The complicated information does not effectively noticed individuals. To improve this situation, the system required to provides more simple and efficient notification.

Usability: One goal of this system is to explore a better way to utilize and manage personal data. This goal reflected in the usability requirements. Not only should the application be easy to use and provide a clear and to-the-point UI, but it should actively support those individuals' data collection and management to gain more benefits. This means, that the several of data to be collected in the system should be

clearly presented, and the user should be well informed when she/he start collecting or selling these data.

Stability: Individuals collect and manage their data even sell the data by themselves. In addition, this system makes each node run as an isolated database and interacting with smart contract by send transactions. Which requires the system can continue operation properly when the delay happening. As we know the blockchain network requires time to generate a new block. Thus, confirming the authenticity or whether continue the transaction also need wait for the transactions included in the valid chain. During this waiting period, the system should have enough ability to keep the stable and proper execution.

Consistency: This thesis combined the AWARE framework and Ethereum platform, the system requires meeting the external interface requirements from these two frameworks. The system should follow the same JSON structure in AWARE when expanding data type, and use the same data structure in Ethereum when sending new transactions or messages to smart contract. Moreover, the interface requirements with the Android operating system need to follow the Android API restrictions.

3.3. Prototype Architecture

The system is designed for allowing individual user have more power on control and manage their personal data, and enable users to gain more explicit and implicit benefits through selling their personal data to the service provider under a mode with highly protected security and privacy. Individual users regard as data producer in this system, can be well noticed and control their personal data collection process, and can through the blockchain network to publish their data sets' size to blockchain network. End devices under the users' control collect sensor data and encode the raw data into JSON format, then hash the sensor data and deliver the hash code to smart contract to increase the variable of dataset size in their producer object, which stored in the blockchain. After confirmed consumer bought their data, producer start following the instructions defined in the transaction sends data to the consumer. Considering about the real situation, entering the confirm request transaction into blockchain need to finish the PoW, which needs to spend at least on new block time waiting for the response. Individuals not willing to waiting such long time to confirm the transaction and send data. The processes will automatically run on the behind. When it is ready, it just shows a confirm notification and asks the permissions from users.

Meanwhile, service providers in different areas act as data consumers in this system. Based on this system they can reduce the cost of collect personal data and easier gain the specific and enough datasets their want from users. The data distributed on end users' devices. Therefore, consumers do not need to worry about how to manage and store the data, what they only need to do is pay for the specific data sets they want whenever they need it. Nevertheless, the data consumer categorized by their area and assigned with the different priority. If the consumers who cannot get enough priority, they not allowed to get some private data such as GPS etc.

Based on the requirements, the architecture designed. Similar to the MyData MyData examples, this is a typical architecture with MyData Operator. It involved

three main components. As Figure 14 shows, the first one is Smart Operator contains the Node.js server which acting as the forwarding server also consumers' database in this thesis and the smart contract deployed on the Ethereum which serves as a bridge to establish the connection between the end users and different service providers and provide protection for users' security and privacy. The data consumer part contains different organizations such like health organization, insurance organization, and supermarket and so on. They create consumer account and transfer the money in the real word to coins using in Ethereum to buy producers' data. These transactions processed by smart contract, they have own database to handle the data from data producers. The data producer consisted by the individuals who want to collect and manage their personal data by themselves. They have their smart phone installed the application. Moreover, the AWARE framework, which has been embedded as a library inside of the application. They store the data on their smart phone's local database. Producer relies on smart contract to get all information they need and upload hash code as proof of their data size. The component developed by this project will be descript in detail in section 3.4. The third party like Ethereum and AWARE already introduced in background chapter. The main process in each part described as follow.

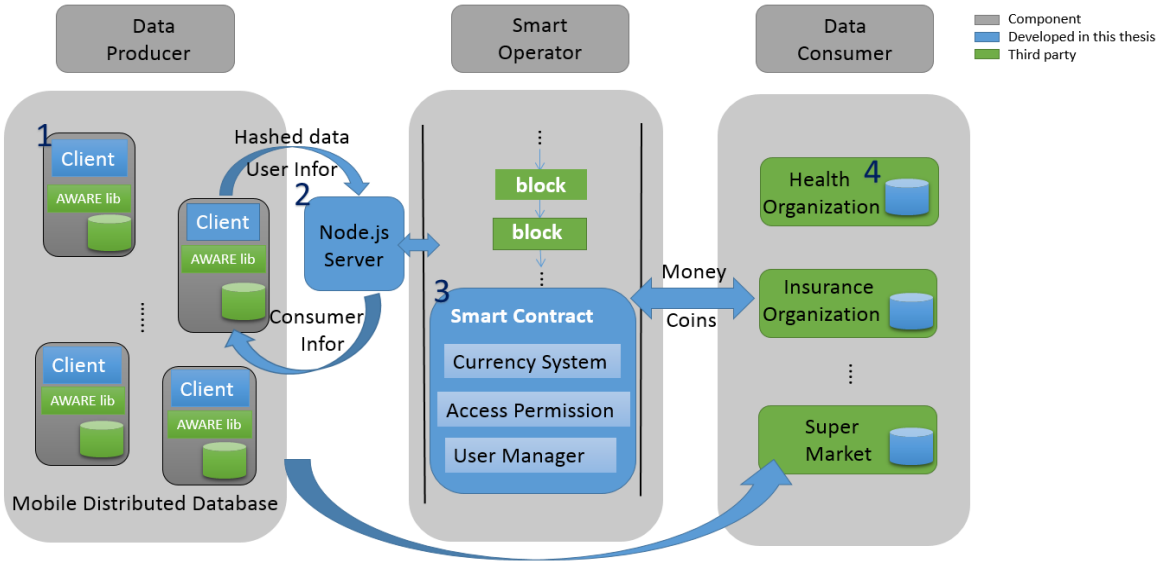

Figure 14. Overview of system architecture.

Data producer

- Creating producer account and using AWARE to collect and store data to local database on the phone.
- Upload hash value to blockchain when collect enough data to make a data block.
- Periodic send request to get produce object property to check whether exist any new transaction.
- If received response from blockchain network, parse the response to get the latest status and store the information in the system.

Smart Operator

- Manage and store users' (data consumers and producers) information and balance.

- Provide producers' database property including database name, owner, and size.
- Handle the request from consumer, complete transactions between consumers and producers (involves check the balance of the data consumer, adjust consumer and producer's balance etc.), consumer between smart contract (create consumer account, transfer coins),
- Handle request from client, the response gets request with the latest status, consumer's information to producer, parse the variable from post request to alter producer object property or create producer account.

Data consumer

- Create new consumer account and transfer enough coins.
- Check the database list and choose one database from this list to buy the data set they want.
- Send enough coins to smart contract to buy users' personal data and send smart contract the information of destination as well.
- Receive the data from the users and store it.

3.4. Implementation

3.4.1. *Hardware*

Ethereum node and Node.js server in this project host by a laptop. The module is HP EliteBook 840 G3 Notebook PC, which has Intel Core i5 6300 (2.40 GHz) CPU and 8GB memory, 500 GB SATA (Serial Advanced Technology Attachment) storage. 64-bit Windows7 Operating System installed on it. The Android client used LG G3 phone with Android OS version 5.0. It has four CPU cores, which consists of four Quad-core 2.5 GHz Krait-400 processors, running on Qualcomm MSM8974AC Snapdragon 801 Chipset. In addition, embedded accelerometer, gyro, proximity, compass sensors. It has 2GB RAM and 16GB storage, and provide dedicated microSD card slot, which can expand its memory up to 256 GB.

3.4.2. *Smart Contract*

MyData approach proposed MyData Operator, which used to provide MyData accounts and related services [24]. Considering the concurrency and security, smart contract to running as the Smart Operator. The concept of the smart contract already explained in the previous chapter. Smart contract in this thesis deployed on the Ethereum blockchain network, any Ethereum node can through the general contract address to get access to it. Based on this smart contract, a series of simple but valid rules built to allow convenient and efficient transaction between data producers and data consumers.

Smart contract assigned with number 3 in Figure 14. In this system, the smart contract provides four main functions: create account for data producer and consumer; manage and store users' profiles; provide transaction services among producer, consumer and itself; provide information inquiry services for producers and consumers. These functions described as follow.

When creating a producer account, user just needs to fill their user name and send the request in client side, the smart contract and client will automatically store the information and create data producer account for them. After create account success user will be notified in the client, then producers can start selling their data to get rewards. Creating data consumer account require providing more information like their database address, their property of organization and so on, which used to assign priority to them. Priority required when consumer want to buy different personal data. Another part for creating consumer account needed is transfer money using in the real world to Ethereum coins.

All accounts organized by the hash of their username, so user unable to create account with same username in smart contract. Smart contract will create a user profile for each account, which stores the information about users in contract's persistent storage. The mapping of each instance uses the hash code of its username, in which Sha3() function used to execute hash operation. All the data in smart contract is transparent for everyone, which used to ensure the transaction is valid. Everyone can check consumers' balance and priority if they want, which can prevent some forbidden or deceive activities happens.

In data transaction part, consumer just needs to fill the database name and pay enough coins. They can send a transaction to buy the personal data they want. After confirm consumer have enough coins to complete this transaction, smart contract operates consumer's and producer's balance, store the necessary information such as priority, period, consumer address in producer's profile and log the "bought" and "sell" event. Furthermore, contract can specific temporal constraints to data. Which means consumer can receive real-time data continually in a period which Specify by themselves, like 10 days or one month.

Moreover, smart contract allows all users query the information. For consumers, it provides the function to get databases name, dataset size, owners. Which used to help consumers to select the suitable databases and the necessary information to start a trade. Another function used to get their balance, consumer confirm the money transfer process completed or failed, or they need transfer more coins to start the next trade. Like consumer, contract also provides the similar functions to get producer's profile including balance, dataset size etc. However, the most important are the buyer's information, based on this, producers can know whether exist any new buyer, how many data and how long should they provide and where to send their data.

To help data producer know how and when their data used, a function for consumers to record their research events deployed, which involved producers' personal data. Consumer can call this function by simply fill their name, the time of the event, databases name, event description. They can use the minimal gas to send this transaction; it can be very helpful for producers to know what happens with their person data. However, it is not a mandatory function. All users can check the specify consumer's events by call checkEvent() function.

3.4.3. *Node.js Server*

Node.js is an open-source, cross-platform JavaScript runtime built on Chrome's V8 JavaScript engine [69]. It created because concurrency is a big problem in many server-side programming languages, and easily with poor performance. Node.js is lightweight and efficient as its event-driven, non-blocking I/O design. The non-blocking model in Node.js refers the commands execute in it are parallel, and use

callbacks to signal failure or completion [68]. Which allows the single-threaded approach as performing I/O in any function must use a callback. Operations in Node.js run on a single thread can reduce the cost of thread context switching when a lot of concurrent connections happening. It also used to optimize the Web application's throughput and scalability with many I/O operations. These design choices enable developing diverse variety of server tools and applications with highly concurrent and scalable ability in JavaScript.

The implementation of Node.js server aims to solve the problem that the outside world cannot directly communicate with Ethereum blockchain. It assigned with number 2 in Figure 14. As the system requires check the transactions between producer and consumer and get the necessary information, and the only way is via the JSON-RPC interface provided by Ethereum to communicate to a local Ethereum node. Thus, the Node.js server component performing as the forwarding server on the computer running Ethereum node, the major work is to forward the transaction from end users to Ethereum network and return the response from smart contract to end users. Node.js server deployed on the Window operating system in this experiment, while it can also deployed on other platforms like Mac OS and Unix etc.

This Node.js server also has another two functions: receive data from end users, store and manage the data from users. It acting as consumers' databases to handle the data sent from users, it assigned with number 4 in Figure 14. It use the HTTP protocol to communication. The workflows of Node.js server showed in Figure 15. After server start running, the server keeps waiting for post request from users. The server handles different request with different operations. If user post transaction to check the status in blockchain, the server forwards this transaction to Ethereum node running on the same computer. When users post their personal data, then server store it in different file by their user name. All these operations can execute simultaneously, which allows multi-user can post their request and data to one same Node.js server at the same time.

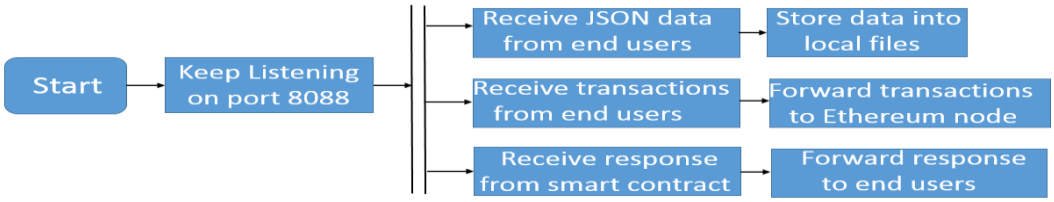

Figure 15. Node.js server workflow.

3.4.4. *Android Client*

The client runs on Android mobile devices, which can quickly to use even the users does not typically excel in technical expertise. It is a highly usable and extendable application that provides user a clearly GUI to guide them. It assigned with number 1 in Figure 14. A simple way to create a new account deployed in client side, user just need to enter their username and click create button (Figure 18). After confirm to create new account, create new account request will send to smart contract. User will receive a notification when it failed or completed.

The client mainly works with four steps: collecting the data, packaging and processing the data, sending the data and waiting for response. The most important step in the client is collecting data when tracking of users' situation, e.g., speed and

motion, location. For this end, GPS sensor and Accelerometer sensor are utilized in Android client app which handled by AWARE library. Moreover, GPS and Accelerometer provide the speed along with position data such as longitude and latitude, and accelerometer data respectively. The raw data collected from smart phone. The raw data will be encoded in JSON format and be stored in local SQLite database. When the GPS or Accelerometer sensors of the user generate a new value, AWARE encode the data into a JSON object and call it an individual record or individual data. In sensors data, using the default setting provide by AWARE, the sensors' data structure shows as Figure 16 and Figure 17. One accelerometer individual data consists of eight properties: an index data used as primary key, a timestamp when the data generated, a device id of this smart phone, an acceleration includes X, Y, Z-axis, accuracy of the sensor, a customizable label. One GPS individual record own 11 properties: an index, a timestamp when the data is generated, a device id of this smart phone, a coordination includes longitude, latitude and bearing, a data of speed, provider such like GPS or network, a latitude data, accuracy of the sensor; a customizable label.

All the data above mentioned is stored in the local database and processed by Android application, gather 60 GPS records and 60 Accelerometer records as one data block, which based on the evaluation. When generating one data block, used MD5 (Message-Digest Algorithm) function to hash it and send the result to Node.js server. Using the AWARE default sensor frequency to collect data but users can set the frequency or using the default frequency to send hash code to server.

Table field	Field type	Description
_id	INTEGER	primary key, auto incremented
timestamp	REAL	unixtime milliseconds since 1970
device_id	TEXT	AWARE device UUID
double_values_0	REAL	value of X axis
double_values_1	REAL	value of Y axis
double_values_2	REAL	value of Z axis
accuracy	INTEGER	Sensor's accuracy level (see SensorManager)
label	TEXT	Customizable label. Useful for data calibration or traceability

Figure 16. Accelerometer data structure [19].

Table field	Field type	Description
_id	INTEGER	primary key, auto incremented
timestamp	REAL	unixtime milliseconds since 1970
device_id	TEXT	AWARE device UUID
double_latitude	REAL	the location's latitude, in degrees
double_longitude	REAL	the location's longitude, in degrees
double_bearing	REAL	the location's bearing, in degrees
double_speed	REAL	the speed if available, in meters/second over ground
double_altitude	REAL	the altitude if available, in meters above sea level
provider	TEXT	gps or network
accuracy	INTEGER	the estimated location accuracy
label	TEXT	Customizable label. Useful for data calibration or traceability

Figure 17. Location data structure [19].

The communication components applied to deal with the producers' queries and send data (hash data or real-time data) requests. At the beginning, the user opens the client, then send_data service will start running and activating the sensors based the default value or user's previous setting. User navigated to the main screen. From here user can choose the sensors that he/she want to active or reactive. For convenient user purpose, few methods implemented to allow user can quickly choose the sensors, like select all, cancel select, de-select. After user made his decision and select save button, client will ask the permissions to start to collect data and store the data. Only gain user authorization, the result can be stored in the shared preference for other methods to read or next open this screen to read. Besides, from the side menu, user can move to the interval screen to set the interval to send data to server.

In the sen_data service, main activity transfers the sensor's status and interval to service when it executes. Service based on the sensors' status to choose query database or skip. The interval used to wake up the service to send new sensor data to Ethereum smart contract or data consumer. Meanwhile, client will continuous request Ethereum node to get the latest status. After get response, it parses response and updates the transaction status stored in the client. The transaction status used for the conditional judgment before user sends real-time data to consumer. The detailed workflow shows in Figure 19. Screenshots from each fragment seen in Figure 18, which displays the main screens the users can access in the application.

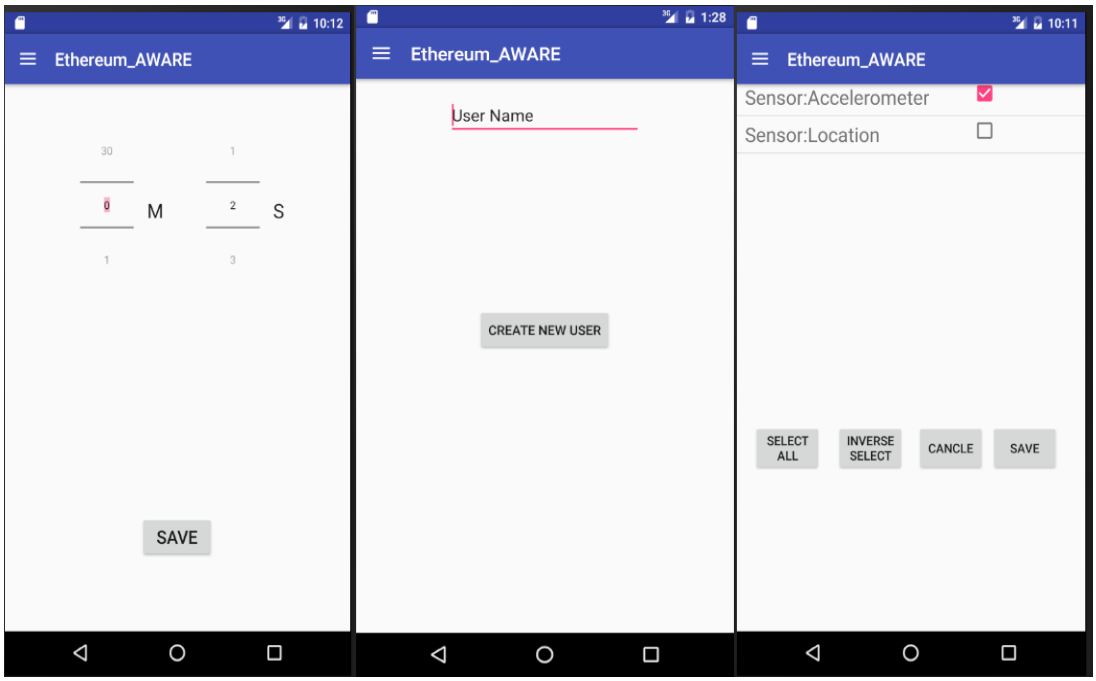

Figure 18. Android client screenshots.

Each function in smart contract has its identify code, when users need to call this function, this identify code used to locate this function, and the parameters users want to pass also encoded to binary code following the rules defined by Ethereum. For all users in this system, using Ethereum JSON-RPC (JavaScript Object Notation-Remote Procedure Call) API send the transaction to call the functions in smart contract, which based on JSON-RPC 2.0, a stateless, lightweight RPC protocol [70]. The main methods used described as follow. The first method is "eth_sendTransaction", which used to create new transaction or new contract. Its

data field contains binary code, and it returns a HEX string that is a 32 Bytes transaction hash. The gas parameter is optional, its default value is 90000. It is an integer of the gas provided for the transaction execution. It will return unused gas. The gasPrice parameter is the integer of the gasPrice used for each paid gas. Example:

```
//Request
{
"jsonrpc":"2.0",
"method":"eth_sendTransaction",
"params": [{
"from": "0xb60e8dd61c5d32be8058bb8eb970870f07233155",
"to": "0xd46e8dd67c5d32be8058bb8eb970870f07244567",
"gas": "0x76c0",
"gasPrice": "0x9184e72a000",
"value": "0x9184e72a",
"data":"0xd46e8dd67c5d32be8d46e8dd67c5d32be8058bb8eb970870f077278598924456
75058b"}],
"id":1
}
//Result
{
"id":1,
"jsonrpc": "2.0",
"result":
"0xe670ec64341771606e55d6b4ca35a1a6b75ee3d5145a99d05921026d1527331"
}
```

Another method is “ethe_getTransactionReceipt”. It used in two conditions. First, when you created a contract, after the transaction mined, which used to get the contract address. Second, it used to get the events data trigged by the transaction. It returns a transaction receipt object, or null when no receipt found based on a transaction hash. Example:

```
//Request
{
"jsonrpc":"2.0",
"method":"eth_getTransactionReceipt",
"params”:["0xb903239f8543d04b5dc1ba6579132b143087c68db1b2168786408fcbce568
238"],
"id":1
}
//Results
{
"transactionHash":
"0x9fc76417374aa880d4449a1f7f31ec597f00b1f6f3dd2d66f4c9c6c445836d8b",
"transactionIndex": 0,
"blockHash":
"0xef95f2f1ed3ca60b048b4bf67cde2195961e0bba6f70bcbea9a2c4e133e34b46",
"blockNumber": 3,
"contractAddress": "0xa94f5374fce5edbc8e2a8697c15331677e6ebf0b",
"cumulativeGasUsed": 314159,
"gasUsed": 30234,
"logs": [{
         // logs as returned by getFilterLogs, etc.
     }, ...]
}
```

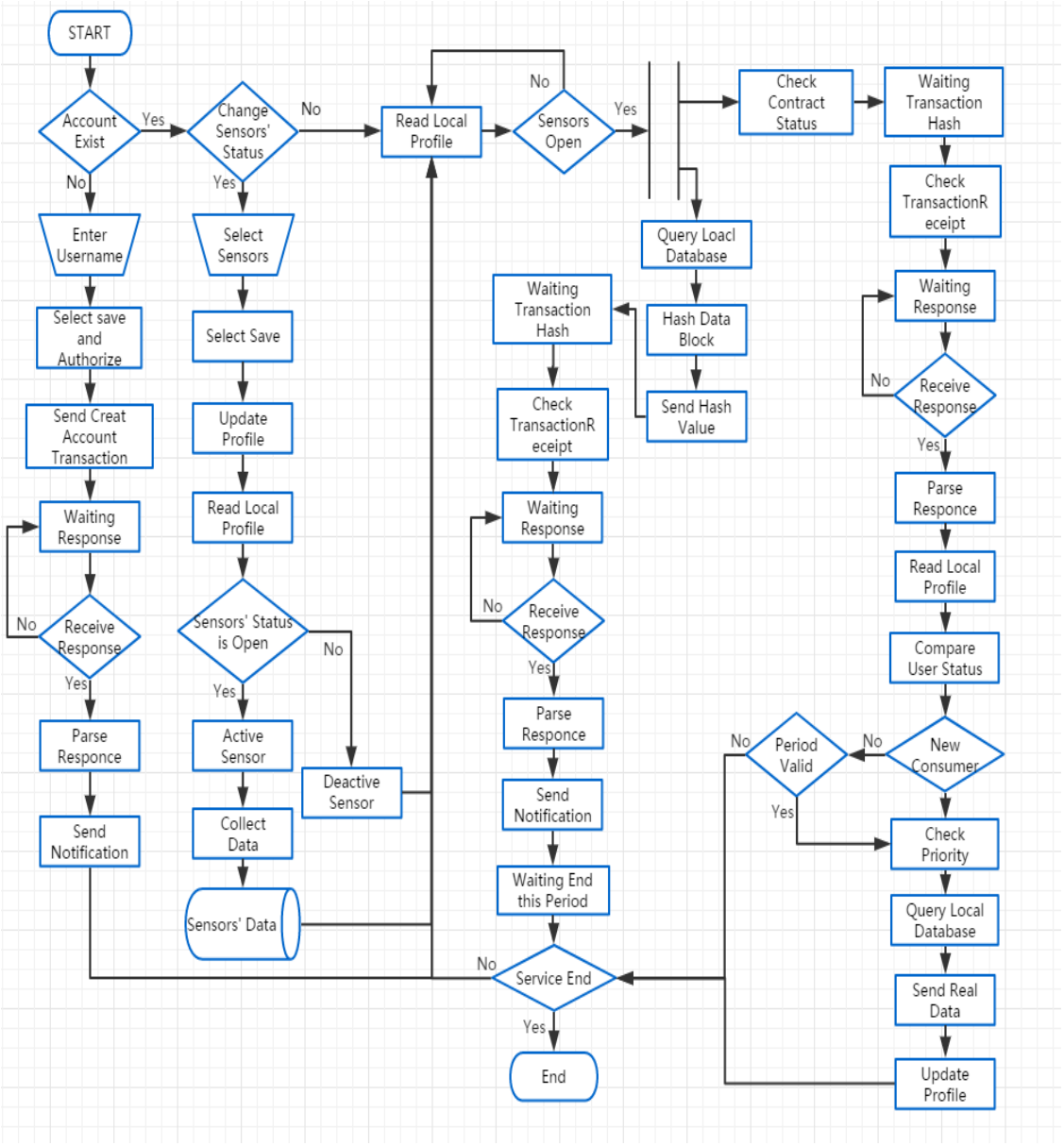

Figure 19. Android client workflow.

4. EVALUATION & RESULTS

This chapter describes the experiments setup, result and analysis. Two experiments designed to study the performance and related issue of the blockchain based MyData systems. The experiments show how Ethereum gas price, GPS data generation rate influence the performance of the system. The first experiment establish the static wait time between send a transaction and miner included a transaction to Ethereum blockchain. The second experiment evaluate the GPS data generation rate, which used establish the records quota in the data block and the interval to send add new data block transaction to smart contract.

4.1. Setup

The evaluation consists of two parts: the Android client, the Ethereum node. The Android client deployed on LG G3 phone, its configurations introduced in the previous chapter. The Ethereum node running on the HP EliteBook 840 G3 Notebook PC, which contains 8 GB memory and 500 GB SATA storage. The GPS data collected from LG G3 phone and stored in its local database. Each individual GPS record contains 11 elements: the index, the timestamp, a coordination includes longitude, latitude and bearing, the speed, the provider, the latitude, the accuracy, the customizable label. The data enveloped into JSON format. In addition, MD5 used on client side decoded GPS data to 32 bits hash code. All transactions send to Ethereum node processed by Android client automatically.

4.2. Experiment

4.2.1. *Static Wait Time Comparison*

Each transaction in blockchain needs to waiting for confirmed by the whole network, the waiting time including the network transmission time, queuing time and server processing time. Different with the typical server client network, not only waiting for the node to verifying the transaction request, transaction also needs to wait for the blockchain network generate a new block that contains this transaction. The block generates rate was discussed in the background chapter, which limited the transaction rate in blockchain. Furthermore, when implementing smart contract on the public network, there can have hundreds or thousands of transactions happening per second, which has considerable influence on the system's performance. The only way to fast the confirmation time is providing higher gas price to attract miners collect transactions as soon as possible. However, transaction price makes economic sense is another reason to effect the setting of gas price.

The first experiment analyzes and compare the confirmation time with different gas price under the Test-net and Main-net. The Main-net called Frontier network, it launched on July 30, 2015. At the beginning, Frontier intended for use by developers as a beta version. As developers writing smart contacts, decentralized apps to deploy on this live Ethereum network, and miners joining the Ethereum network to help secure this blockchain and earn ether from mining blocks. It turned out to be more

capable and reliable, the ether in the Frontier network can trade in for fiat or BTC or anything with real value. The Ethereum test network moved from Morden to Ropsten in 2016. The Morden Test-net has been running since the launch of the Ethereum blockchain (July 2015) and its block number was scheduled at block 1885000. It functionally equivalent to the main net as following the same rules as Frontier. It pushed to its limits in order to test scalability and block propagation times. Due to in order to make full synchronizing simpler for new users and less resource intensive, and the low difficulty of the Morden Test-net. Ethereum starts a new test network Ropsten without changing the protocol.

The gas price is 20 Gwei in Test-net at the time of writing the thesis, which equals 0.00000002 Ether, and 1 Ether equal 46 euro. If people sending transactions with gas price less than 20 Gwei, those transactions would own low priority. Miners prefer collect transactions with normal or higher gas price. In this experiment, 100 same transactions with same gas price sent to establish the static transaction confirmation time (see Appendix 1 for an example transaction). In Figure 20-1, the result shows the transactions pending time categorical grouped with different gas price in Test-net. Figure 20-2 shows the result in Main-net. The X-axis represents the gas price and the Y-axis represents the pending time. The average, median, mode, max, min value used different color to represent. As illustrated in the Figure 20-1, the average, max pending time with 25 Gwei gas price is 2.43 seconds, which less than other two conditions. With the normal gas price, the average pending time is 13.19 seconds, which nearly equal the new block generation time. The average pending time when gas price is 30 Gwei is 7.2 seconds, it less than the transaction with 20 Gwei but higher than transaction with 25 Gwei. The min pending time less than 1 second in different groups. The max pending time in these 300 transactions nearly one and quarter minute. Few transactions take significant time, but have little effect to respective median values.

The gas price in Main-net is 21.25Gwei at present, the experiment sends first 100 transactions with 21.25 Gwei gas price. Similar to the experiment in Test-net, divide this sub-experiment to three groups by add 5 Gwei and 10 Gwei. As illustrated in the Figure 20-2, the average, max pending time with 26.25 Gwei gas price is 15.35 seconds, which also less than other two conditions. With the normal gas price, the average pending time is 26.63 seconds, which nearly equal the two new blocks' generation time. The average pending time when gas price is 31.25 Gwei is 16.63 seconds, it less than the transaction with 21.25 Gwei but higher than transaction with 26.25 Gwei. The min pending time less than 1 second in different groups. The max pending time in these 300 transactions nearly two minutes. In the Main-net, the max transaction's confirmation time longer than Test-net, and the average confirmation time also longer than Test-net. Similar to the situation in Test-net, few transactions take significant time, but also have little effect on respective median values.

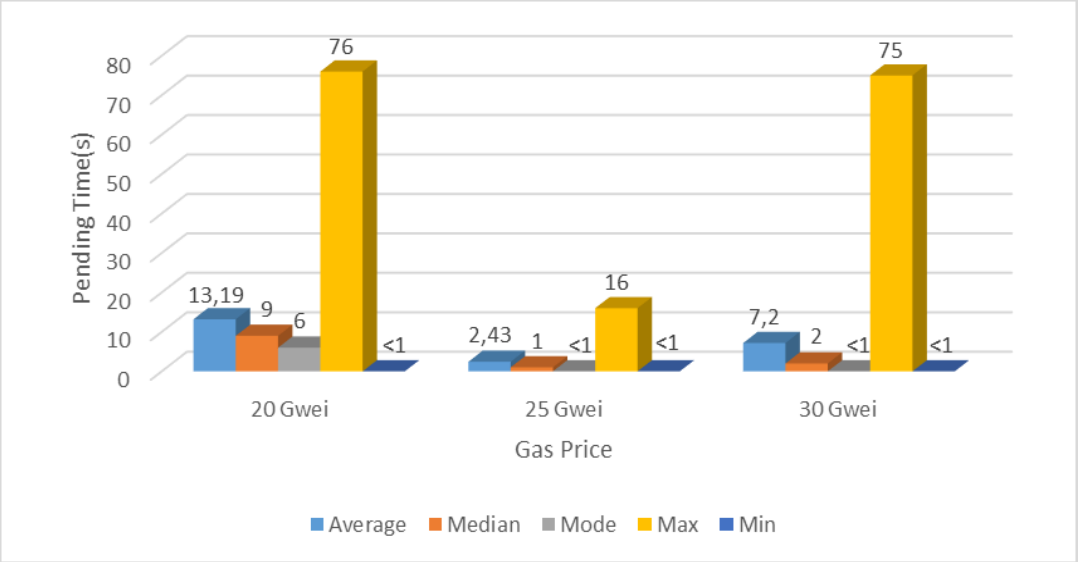

Figure 20-1. Pending time with different gas price in Test-net.

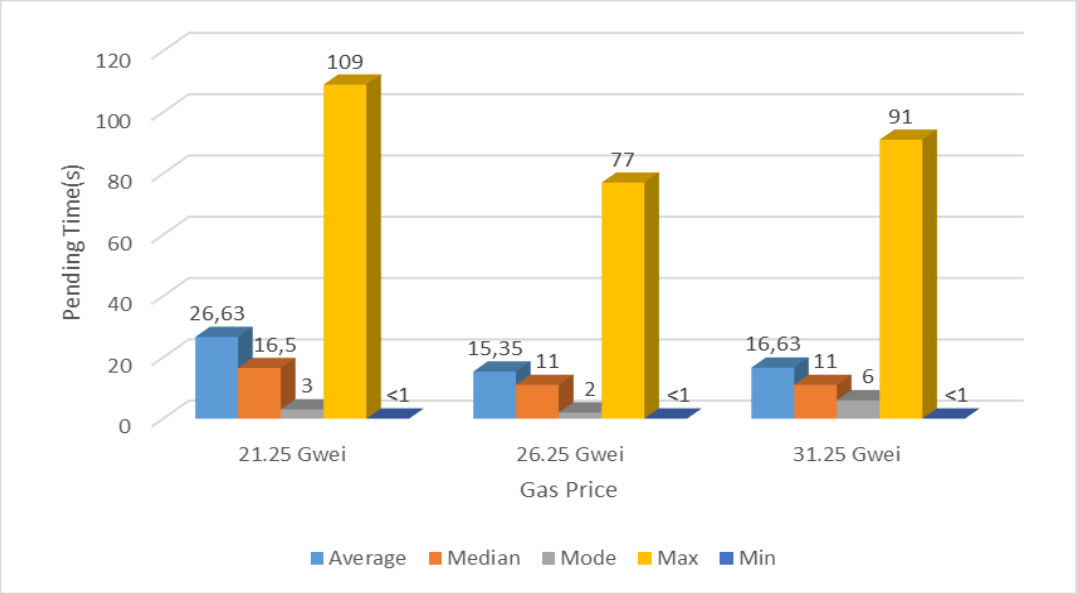

Figure 20-2. Pending time with different gas price in Main-net.

In Figure 21 to 23, the left figure shows the result of 300 transactions' pending time in Test-net and the right one shows the frequency of the pending time. In right figure, the X-axis represents the list of groups that dividing the range of min to max time value to 10 group, the Y-axis represents the number of the transactions.

As presented by the figures above, 43 out 100 of the transactions' pending time less than 7.7 seconds when gas price is 20 Gwei, and 80 out 100 of the transactions' confirmation time less than 7.7 seconds when gas price is 25 Gwei, nearly 91% of transactions' pending time less than 6.8 seconds when gas price is 25 Gwei. Two transactions used more than 60 seconds to be verified with 20 Gwei, and 4 transactions verifying more than 60 seconds with 30 Gwei, and 6 transaction spent more than 10 seconds to be verified with 25 Gwei.

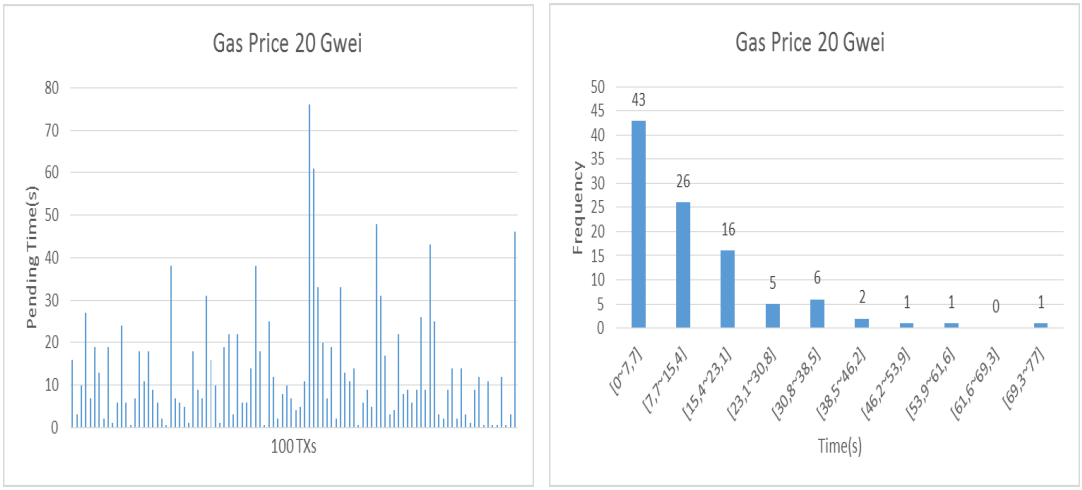

Figure 21. Pending time and frequency with 20 Gwei gas price in Test-net.

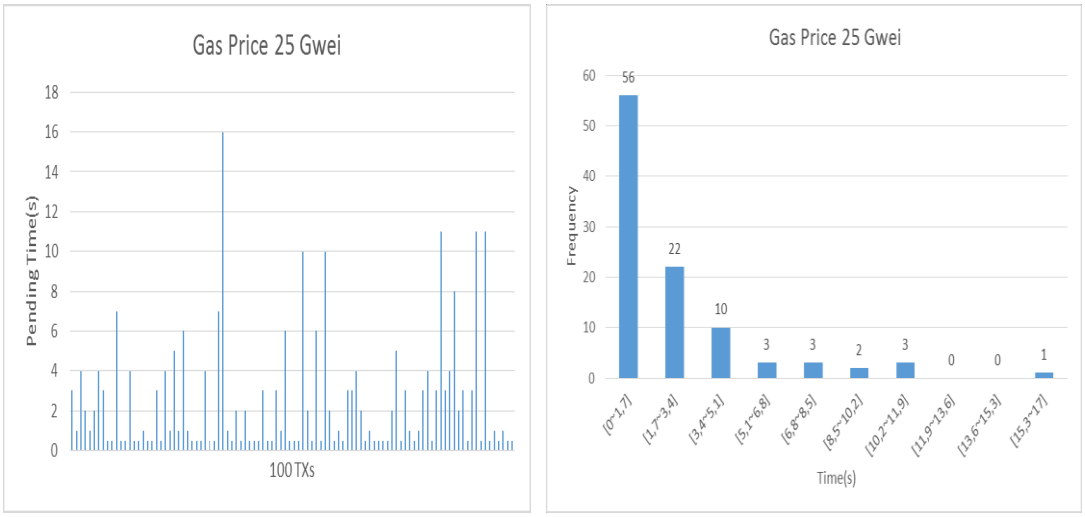

Figure 22. Pending time and frequency with 25 Gwei gas price in Test-net.

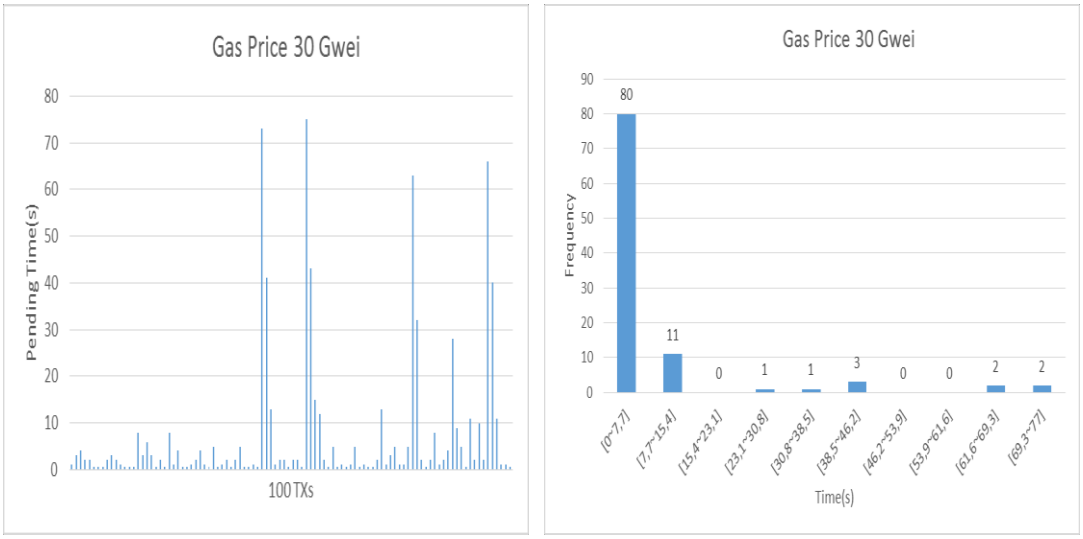

Figure 23. Pending time and frequency with 30 Gwei gas price in Test-net.

In Figure 24 to 26, the left figure shows the result of 300 transactions' pending time in Main-net and the right one shows the frequency of the pending time. As

presented by the figures above, 35 out 100 of the transactions' pending time less than 11 seconds when gas price is 21.25 Gwei, and 61 out 100 of the transactions' confirmation time less than 15.4 seconds when gas price is 26.25 Gwei, 48 out 100 of transactions' pending time less than 10 seconds when gas price is 31.25 Gwei. Six transactions used more than 80 seconds to be verified with 21.25 Gwei, and 4 transactions verifying more than 50 seconds with 26.25 Gwei, and 4 transaction spent more than 70 seconds to be verified with 31.25 Gwei.

The Figure 21 to 26 shows the long tail effect is stronger in Main-net than in Test-net. Long tail effect refers to a distribution having a large number of occurrences far from the central part of the distribution [74]. In Ethereum, there has larger number of online miners and users in Main-net than in Test-net. Besides, earn ether in Test-net is easier because of the lower difficulty and shorter blockchain. Different users could attach different amount of gas based on the priority of their transactions. That means users only willing pay more money to enhance the confirmation of the transactions with high priority. However, in Test-net user can get a huge amount of free ether, users in Test-net willing to pay more gas on any transactions to test their applications or smart contracts quickly. This result most transactions in Test-net occurrences more concentrated than in Main-net.

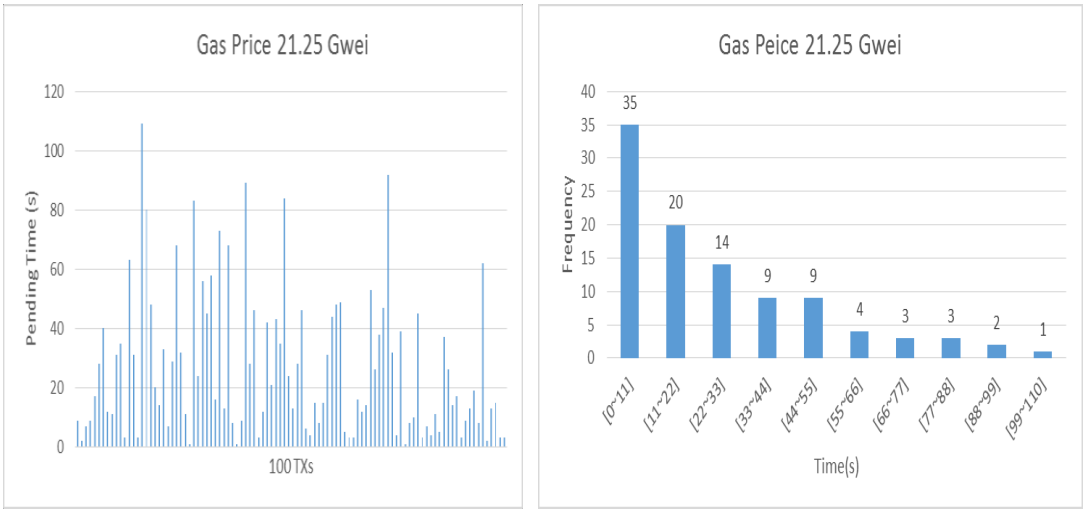

Figure 24. Pending time and frequency with 21.25 Gwei gas price in Main-net.

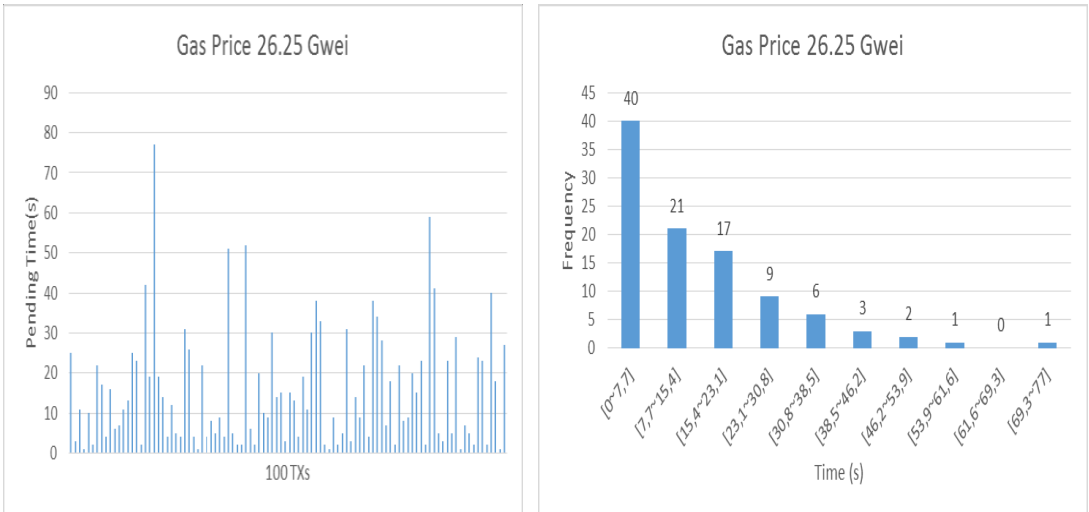

Figure 25. Pending time and frequency with 26.25 Gwei gas price in Main-net.

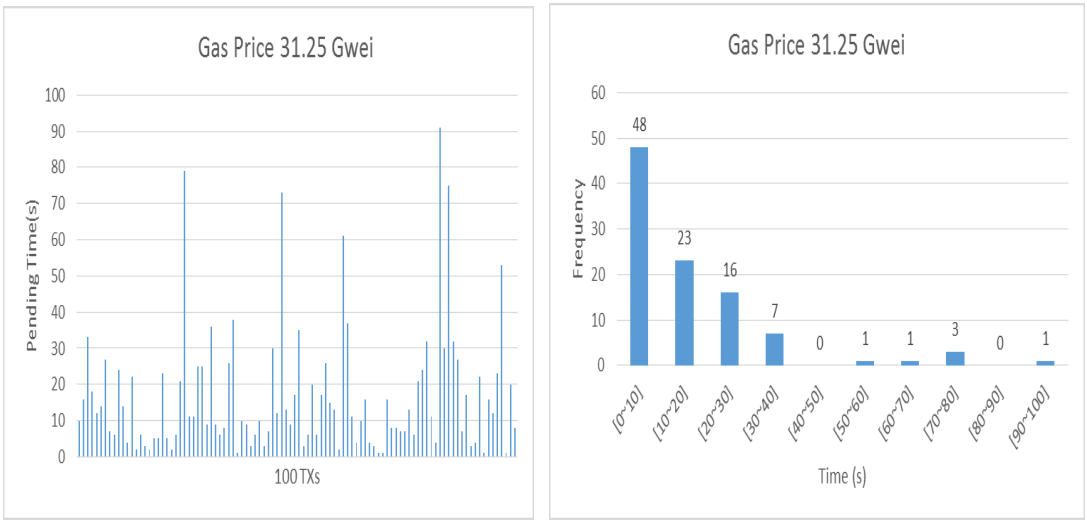

Figure 26. Pending time and frequency with 31.25 Gwei gas price in Main-net.

4.2.2. *GPS Generation Rate*

This experiment tests the stability of GPS data generation in the system. As the Ether has its real value, choosing the appropriate interval to send the new data transaction should consider both the usability and economic reason. As accelerometer sensor will generate hundreds of real time data per second, its data generating time ignored in this thesis. Only the GPS data generation rate measured. At the client side, 10 minutes of the generating time of the GPS data and the number of the GPS records measured. Those GPS data collected by deploy the AWARE inside the Android client as a library, and store the data in local database. The experiment queries the local database and prints the 60 latest records per minute, the result shows in Figure 27.

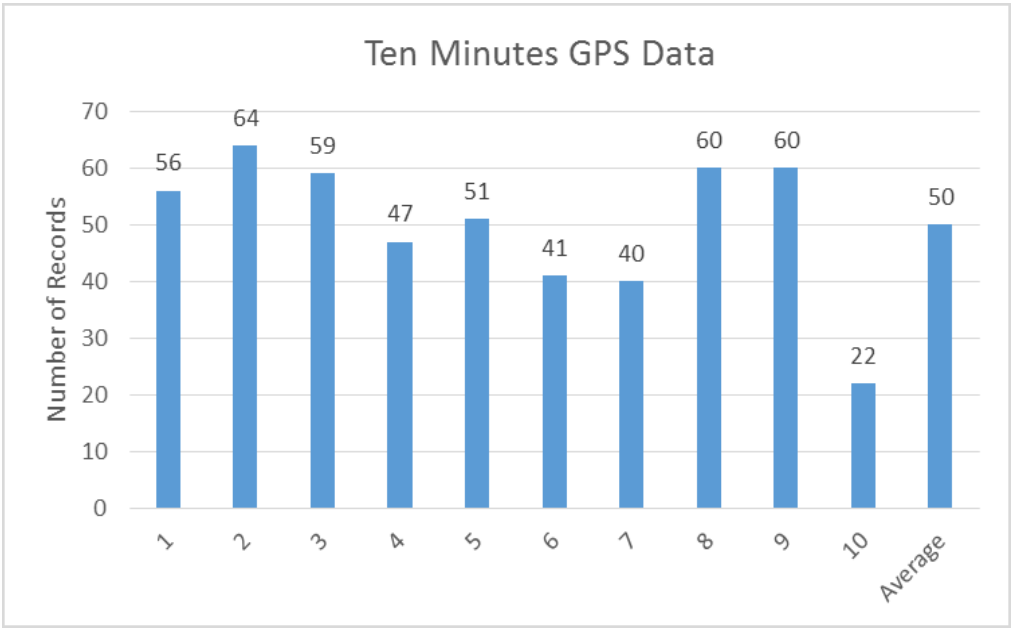

Figure 27. Number of GPS records in 10 minutes.

As presented by the figure above, the X-axis represents the minutes and the average number of the record. The Y-axis represents the number of the GPS record. Based on the figure, the max number of the GPS record is 64 per minute, and min number of the GPS record is 22 per minute, the average generation rate is 50 records per minute.

For the max time interval between generate two continuously GPS record in each minute, the result is the Figure 28. The max time interval is 22 seconds, and the min max time interval is 1 second, the average max interval is 6.6 seconds.

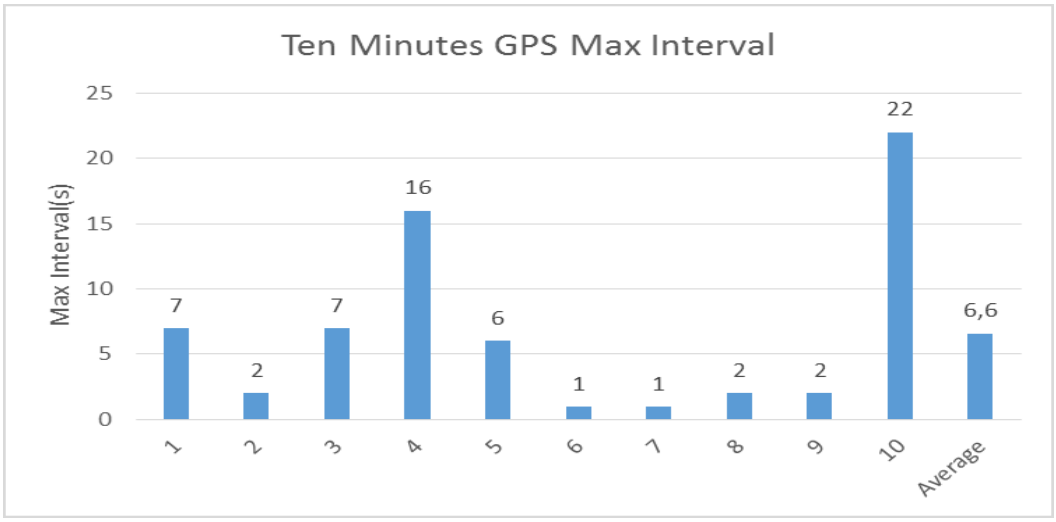

Figure 28. Max time interval in each minute.

The data about all time intervals in 10 minutes are as presented in Figure 29. The time intervals in GPS generation nearly equal 1 record per second. Several records can have large time interval or time interval less than 1 second.

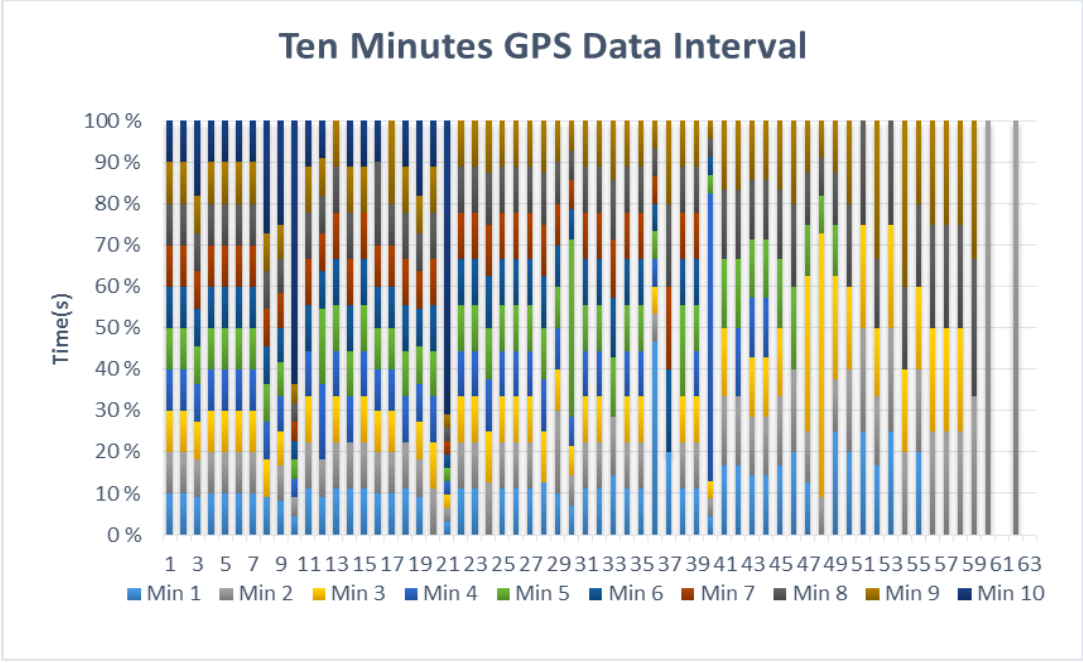

Figure 29. Time intervals in each minute.

4.3. Analysis

From the Static Wait Time Comparison section, the results shows the gas price affects the transaction confirmation time. In the Test-net, when the gas price is more than 20 Gwei, the transactions' pending time smaller than transactions with the normal gas price. To verify the correlation between gas price and transaction confirmation time, SPSS Pearson Correlations Analysis used to analysis the gas price and the transaction pending time, the result shows in Table 2. The VAR1 present the gas price, the VAR2 present the transaction pending time. As shown in the table, there has significant relationship between gas price and transaction pending time at the 0.01 level ($p=0.001<0.05$). It means selecting a high gas price can enhance the system performance through reduce the transaction pending time. However, the results shows that the transactions' average pending time with 25 Gwei gas price lower than 30 Gwei gas price. It shows the transaction pending time not only affected by the gas price. The network latency, the number of online miners, the difficulty of the current block etc. would affect the transaction confirmation time. That means the relationship between transaction pending time and gas price not strict linearly proportional.

In Main-net, SPSS Pearson Correlations Analysis also used to analysis the gas price and the transaction pending time, the result shows in Table 3. The VAR1 present the gas price, the VAR2 presents the transaction pending time. As same as the Test-net, there has significant relationship between gas price and transaction pending time at the 0.01 level ($p=0.000<0.05$). In addition, the results show the transaction's average confirmation time with 26.25 Gwei gas price lower than other two conditions. This situation is similar as Test-net. As mentioned before, the Test-net deployed same protocol as Main-net, the result correspond this fact. Through compare the average pending time. The results shows the average pending time in Main-net longer than in Test-net. This can caused by the longer blockchain in Main-net and the different difficulty between Main-net and Test-net.

Considering the system will send this transaction periodically, and the Ether has its real value, 1 Ether equal 46 euro when writing this thesis. According to the result shows in the Test-net, confirm transactions with normal gas price require less than one block time averagely. In addition, the result in Main-net shows the confirm transactions with normal gas price require less than two block time averagely. One block time in Ethereum platform is 15 seconds. As confirm a transaction within two block-time would not significantly affect the system performance, the normal gas price selected as the default gas price to send transactions.

Based on the GPS Generation Rate section, AWARE in this LG G3 phone's GPS data generation rate is 0.8 record per second. Many people preferred walking speed at about 1.4 m/s [71]. The max time interval between generating two GPS data is 22 seconds that means users can have nearly 31 meters disconnected distance. Considering the use case, the patients are treating by physical therapist. Their walk speed can be slower than healthy people can, and the research not focuses on where are patients going or went, the GPS data generation rate is acceptable in this system.

Considering the generation rate and the economic reason, one transaction contain only one GPS record is too expensive. However, envelop all data in one transaction to save transaction fee also not doable. Which can result another expensive situation when the data is miss. The transaction confirmation time nearly 15 seconds when send transaction with normal gas price, and AWARE can collect 50 records per

minute. The GPS data block consist of 60 GPS records, and the default interval set as 60 seconds is appropriate choice. When send add new data to smart contract, client query 60 latest GPS records in local database, this approach can result few redundant data but guarantee the data continuity. In addition, the transaction can confirm before sending the next transaction.

Table 2. Correlation analysis result in Test-net

Correlations

				VAR00001	VAR00002
VAR00001	Pearson Correlation			1	-,195**
	Sig. (2-tailed)				,001
	N			300	300
	Bootstrap[c]	Bias		0	-,007
		Std. Error		0	,074
		95% Confidence Interval	Lower	1	-,359
			Upper	1	-,065
VAR00002	Pearson Correlation			-,195**	1
	Sig. (2-tailed)			,001	
	N			300	300
	Bootstrap[c]	Bias		-,007	0
		Std. Error		,074	0
		95% Confidence Interval	Lower	-,359	1
			Upper	-,065	1

**. Correlation is significant at the 0.01 level (2-tailed).

c. Unless otherwise noted, bootstrap results are based on 1000 bootstrap samples

Table 3. Correlation analysis result in Main-net

Correlations

				VAR00001	VAR00002
VAR00001	Pearson Correlation			1	-,207**
	Sig. (2-tailed)				,000
	N			300	300
	Bootstrap[c]	Bias		0	,001
		Std. Error		0	,057
		95% Confidence Interval	Lower	1	-,318
			Upper	1	-,091
VAR00002	Pearson Correlation			-,207**	1
	Sig. (2-tailed)			,000	
	N			300	300
	Bootstrap[c]	Bias		,001	0
		Std. Error		,057	0
		95% Confidence Interval	Lower	-,318	1
			Upper	-,091	1

**. Correlation is significant at the 0.01 level (2-tailed).

c. Unless otherwise noted, bootstrap results are based on 1000 bootstrap samples

5. DISCUSSION & FUTURE WORK

5.1. Discussion

The developing of Ubicomp technologies allow collecting personal data easily and conveniently. As personal data more and more important and valued, new approaches and technologies required enable bring the value of personal data for people and enable the individual to manage and control of their own data. This thesis work utilizes blockchain technologies and AWARE platform in MyData paradigm to explore the personal data management and utilization. In this thesis, through an experiment that based on the scenario to analyzing the performance applying blockchain technology in MyData paradigm. A blockchain and smart contract based MyData system built for the research. In the system, smart contract used to act as MyData Operator to manage the users' account and profile, handle the transaction among data producer, data consumer and smart contract itself, and provide information inquiry services for data producers and data consumers. Personal data generate, manage, storage and delivery are implement on the Android client, which deployed the AWARE library. The blockchain platform used in this thesis is Ethereum. A Node.js server running on a local machine which running the Ethereum node to act the proxy between Android client and Ethereum blockchain and the data consumer's database.

In the different blockchain platform, the parameters can differ greatly. For example, Bitcoin average transaction confirmation time is over 10 minutes and in the worst case can be an hour, which would not work in many practical implementations. If the PoW takes too much computing power smaller applications usually cannot apply it. Moreover, if the transaction fee too expensive also can affect the users and developers choose. In Ethereum, the experiments start in the Test-net to avoid unnecessary waste. After built the whole system, experiments moved to the Main-net to evaluate the system performance in the real world with real ether. In addition, compare the result in two networks to get the conclusion would be more convincing.
To estimate the performance of the system, two experiments carried out. To figure out the static transaction confirmation time in Ethereum blockchain and research the influence of different gas price, the experiments test 300 transactions' pending time with different gas price both in Test-net and Main-net. For confirm the gas price has significant relationship with transaction confirmation time, SPSS Pearson Correlations Analysis used to analysis the gas price and the transaction pending time. In addition, this thesis compares the average confirmation time with different gas price and under the different network to establish the static wait time. To establish the data block size in the system, this thesis design experiments to test the GPS data generation rate.

This study shows that the gas price and the transaction confirmation time has significant relationship. No matter in the Test-net or Main-net network, the transaction pending time would decrease when the gas price increased. However, this two parameter not strict linearly negative. The result shows the highest gas price does not perform best. That is because that the transaction pending time not only influenced by gas price, but including the online miners, network latency, current difficulty etc. The result shows the transaction confirmation time in Test-net nearly one block time when the gas price is normal price, the transaction confirmation time

in Main-net nearly two blocks time when the gas price is normal price. As a recommendation, the developers should use the normal gas price considering the economy requirement when real time system requirements with lower priority. Moreover, developers can appropriate increase gas price when requires high communication performance.

This study also finds out the GPS generation rate in LG G3 phone that deployed AWARE library nearly 0.8 records per second. To reduce the bandwidth usage and choose an acceptable frequency to send add new data transaction to smart contract. This thesis set the default data block size is 60 GPS records. That also set the default interval to send add new data transaction to blockchain is 60 seconds. Each time send the new data transaction to blockchain, system query 60 latest GPS records in local database and hash it then send out. User can manual change the interval in the Android client as their want, but shorter interval would result in more cost in sending transaction and more data that are redundant.

5.2. Future Work

In the current study, this thesis just categorizes the consumers as health organization and other service providers. A simple approach deployed to achieve this by requiring consumers to fill their property and based on this property the system assigned them different priority to buy users personal data. Only the health organizations can buy the GPS data from producers, even this property is transparent for all users in Ethereum blockchain network, but consumers can still adventure to fake their profiles to gain higher priority. In the future work, a more powerful certificate system will implement to prevent consumer fake profiles, require stronger proof like only consumers submit their business license can gain the corresponding priority, or consumer can only gain lower priority at first, after 5 trusty organizations confirm its property then the system will improve its priority.

This thesis deployed the GPS sensor and Accelerometer sensor at present. In the use case, data will provide to health organization for research purpose. However, for a valuable personal data, requires more sensor sources to be included. This thesis deployed AWARE as library in the Android client. More sensors and useful data can be included to the system in the future work. The next step the system will provide all phone's sensors that following the same graphic user interface for users to select.

One more limitation in this system is it only focus on record the data transaction between consumers and producers. In this way, producers can know who bought their personal data and when the data sold. This thesis implements a function for consumers when they want to utilize the data they bought. It requires consumer provide when and where they use those personal data, and whose data they will use. Nevertheless, this thesis cannot promise consumers will call this function because the system does not bind any constraint on it at present. In next step, this system can record the all events when involved producers' data. Each time consumer utilizes the personal data generated by the system, they required to record the event to blockchain, and send notification to data producer. If producer disagrees this research or event, they can send request to stop this event. As the blockchain is transparent for everyone, other users can rely on these events to decide whether they sell their personal data to which organization. This can be more correspond the MyData concepts, which aiming to provide users more power to control their data and choose more suitable ways to utilize their data.

6. CONCLUSION

This thesis explored the research challenges in building MyData Operator to enable personal data access and sharing. To study the performance of blockchain and smart contracts in health-related MyData scenario, a blockchain based MyData system designed and implemented. This system can automatically help individual user to collect personal data and sell their data to service providers to gain reward after obtaining the users' permission. With the system, the impact of gas price on the transaction confirmation time of Ethereum PoW blockchain evaluated. The confirmation time of different blockchain network objectively compared. In addition, the GPS data generation rate in the system evaluated. The experiments designed to figure out how gas price and GPS generation rate can affect the system performance.

When revisiting the original research questions. It can concluded that the blockchain and smart contract technology can facilitate personal data management and utilization. It is the key technologies to building open data market. By utilizing blockchain and smart contract, personal data can protect in high security and make them available to trusted parties for rewards. The system and experiments show the performance that applying blockchain in the MyData can provide a new way to represent and understand manage and utilize personal data.

The first research question is about exploring a more fine-grained and transparent way to manage personal data. The system using the blockchain and smart contract successfully addressed these requirements. In blockchain network, each transaction is transparent for everyone. However, this not means the user in system loses the privacy. All personal data hashed in the blockchain. The smart contract cannot only allow data consumer to buy want they want but also set the priority to limit consumer to buy personal data. After evaluating the transaction confirmation time, the system shows good performance in interacting with smart contract, data consumer and data producer.

The second research question studies how to gain users' trust and permission. For this question, we can conclude that the blockchain and smart contract can make individual users to trust the transaction that confirmed by blockchain network. For example, the information in the smart contract can only alter by consumer send a real transaction. When system informed data producer that a data consumer brought his/her data, the whole blockchain network must confirm this transaction at least once. That specific consumer's information can found based on that transaction receipt. In another word, the data consumer through blockchain and smart contract can gain users' trust and permission, and data producer through blockchain and smart contract can know who brought their data and when they bought it.

The research focuses on the integration of Ethereum blockchain, MyData and AWARE platform. Although the blockchain, and personal data management technologies are hot topic, but this explore rarely done before. This thesis work demonstrates that the blockchain and smart contract can facilitate MyData concept.
The personal data have been collected and utilized with little or non-existent user control. The research shows the potential of blockchain technology in personal data management and utilization area. Ideally, based on the blockchain and smart contract can be able to building an open data market. Data producers can receive rewards after giving their personal data. In addition, this data market can also meet the data consumer's expectations and satisfaction.

7. REFERENCES

[1] Li, I., Dey, A. K., & Forlizzi, J. (2011). Understanding my data, myself: supporting self-reflection with ubicomp technologies. In Proceedings of the 13th international conference on Ubiquitous computing (pp. 405-414). ACM. DOI: 10.1145/2030112.2030166.

[2] Latif, A. I., Othman, M., Ali, N. A., Suliman, A., & Mahdi, O. A. (2016). An Investigation of IoT Importance and Viability of Health Records Retrieval using Electronic Tags in Pilgrimage. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 8(10), 95-98. ISSN: 2180-1843.

[3] Carver, C. S., & Scheier, M. F. (2001). On the self-regulation of behavior. Cambridge University Press. ISBN 0 521 57204 5.

[4] Endsley, M. R. (1997). The role of situation awareness in naturalistic decision making.

[5] DiClemente, C. C., Marinilli, A. S., Singh, M., & Bellino, L. E. (2001). The role of feedback in the process of health behavior change. American journal of health behavior, 25(3), 217-227. DOI: 10.5993 /AJHB.25.3.8.

[6] Saponas, T. S., Lester, J., Hartung, C., & Kohno, T. (2006). Devices That Tell On You: The Nike+ iPod Sport Kit.

[7] Li, I., Dey, A., & Forlizzi, J. (2010). A stage-based model of personal informatics systems. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 557-566). ACM. DOI: 10.1145/1753326.1753409.

[8] Lin, J. J., Mamykina, L., Lindtner, S., Delajoux, G., & Strub, H. B. (2006). Fish'n'Steps: Encouraging physical activity with an interactive computer game. In International Conference on Ubiquitous Computing (pp. 261-278). Springer Berlin Heidelberg.DOI: 10.1007/11853565_16.

[9] Preuveneers, D., & Berbers, Y. (2008). Mobile phones assisting with health self-care: a diabetes case study. In Proceedings of the 10th international conference on Human computer interaction with mobile devices and services (pp. 177-186). ACM.DOI: 10.1145/1409240.1409260.

[10] Perera, C., Wakenshaw, S. Y., Baarslag, T., Haddadi, H., Bandara, A. K., Mortier, R & Crowcroft, J. (2017). Valorising the IoT databox: creating value for everyone. Transactions on Emerging Telecommunications Technologies, 28(1). DOI: 10.1002/ett.3125.

[11] Perera, C., Ranjan, R., Wang, L., Khan, S. U., & Zomaya, A. Y. (2015). Big data privacy in the internet of things era. IT Professional, 17(3), 32-39. DOI: 10.1109/MITP.2015.34.

[12] MYDATA: THE BASICS. URL: http://mydata2016.org/2016/08/12/mydata-the-basics/. Retrieved March 03, 2017.

[13] European Commission. Protection of personal data. URL: http://ec.europa.eu/justice/data-protection/document/index_en.htm. Retrieved March 03, 2017.

[14] WÃ¼st, K. (2016). Ethereum Eclipse Attacks. DOI: 10.3929/ethz-a-010724205.

[15] Szabo, N. (1994). Smart contracts. Unpublished manuscript.

[16] Rose, J., & Kalapesi, C. (2012). Rethinking personal data: strengthening trust. BCG Perspectives, 16(05), 2012.

[17] Szczekocka, E., Gromada, J., Filipowska, A., Jankowiak, P., Kałuzny, P., Brun, A. & Staiano, J. Managing Personal Information: A Telco Perspective.

[18] Ferreira, D., Kostakos, V., & Dey, A. K. (2015). AWARE: mobile context instrumentation framework. Frontiers in ICT, 2, 6. DOI: 10.3389/fict.2015.00006.

[19] AWARE: Android Mobile Context Instrumentation Framework. URL: http://www.awareframework.com. Retrieved March 18, 2017.

[20] Google Research. PACO - The Personal Analytics Companion. URL: https://www.pacoapp.com/. Retrieved March 18, 2017.

[21] Karr C. Purple Robot. URL: http://tech.cbits.northwestern.edu/purple-robot/. Retrieved March 18, 2017.

[22] funf | Open Sensing Framework. URL: http://www.funf.org/. Retrieved March 18, 2017.

[23] MQTT V3.1 Protocol Specification. URL: http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html. Retrieved March 23, 2017.

[24] Poikola, K. A., & Honko, H. (2010). Mydata a nordic model for human-centered personal data management and processing. tech. rep., Ministry of Transport Finland.

[25] Su, X., Hyysalo, J., Rautiainen, M., Riekki, J., Sauvola, J., Maarala, A. I. & Honko, H. (2016). Privacy as a Service: Protecting the Individual in Healthcare Data Processing. Computer, 49(11), 49-59. DOI: 10.1109/MC.2016.337.

[26] Swan, M. (2015). Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.". ISBN: 978-1-491-92049-7.

[27] Espinel, V., O'Halloran, D., Brynjolfsson, E., & O'Sullivan, D. (2015). Survey Report:"Deep Shift: Technology Tipping Points and Societal Impact.". In World Economic Forum, September.

[28] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[29] Brown, William L., "An Analysis of Bitcoin Market Efficiency Through Measures of Short-Horizon Return Predictability and Market Liquidity" (2014). *CMC Senior Theses*. 864. URL:http://scholarship.claremont.edu/cmc_theses/864. Retrieved March 23, 2017.

[30] Babbitt, D., & Dietz, J. (2014). Crypto-Economic Design: A Proposed Agent-Based Modeling Effort. In English. Conference Talk. University of Notre Dame, Notre Dame, USA.

[31] Pilkington, M. (2015). Blockchain technology: principles and applications. DOI: 10.4337/9781784717766.

[32] Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In Security and Privacy (SP), 2016 IEEE Symposium on (pp. 839-858). IEEE. DOI: 10.1109/SP.2016.55.

[33] Peters, G. W., & Panayi, E. (2016). Understanding Modern Banking Ledgers through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money. In Banking Beyond Banks and Money (pp. 239-278). Springer International Publishing. DOI: 10.1007/978-3-319-42448-4_13.

[34] Smart Contracts are self-executing contractual states, stored on the blockchain. URL: https://smartcontract.com/. Retrieved February 21, 2017.

[35] Jones, M. H. Blockchain and T2S: A potential disruptor. URL: https://www.sc.com/BeyondBorders/wp-content/uploads/2016/06/2016-06-16-BeyondBorders-Report-SCB_Nema_Block-Chain-Paper-Final.pdf. Retrieved February 22, 2017.

[36] Wander, M. File: Bitcoin Block Data.svg. URL: https://commons. Wiki media.org/wiki/File:Bitcoin_Block_Data.svg#/media/File:Bitcoin_Block_Data.svg. Retrieved February 22, 2017.

[37] The MesInfos project on Self Data. URL: http://mydata2016.org /2016/07/27/if-i-can-use-your-data-you-can-too-however-you-please/. Retrieved February 22, 2017.

[38] Hash function. URL: https://en.wikipedia.org/wiki/Hash_function#Hash_function_algorithms. Retrieved February 23, 2017.

[39] Survey on Blockchain Technologies and Related Services. URL: http://www.meti.go.jp/english/press/2016/pdf/0531_01f.pdf. Retrieved February 23, 2017.

[40] Schollmeier, R. (2001). A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In Peer-to-Peer Computing, 2001. Proceedings. First International Conference on (pp. 101-102). IEEE. DOI: 10.1109/P2P.2001.990434.

[41] Peer-to-peer. URL: https://en.wikipedia.org/wiki/Peer-to-peer#References. Retrieved February 23, 2017.

[42] Public-key cryptography. URL: https://en.wikipedia.org/wiki/Public-key_cryptography. Retrieved February 24, 2017.

[43] Badev, Anton I. and Chen, Matthew, Bitcoin: Technical Background and Data Analysis (2014). FEDS Working Paper No. 2014-104. SSRN: 2544331. DOI: 10.2139/ssrn.2544331.

[44] Digital signature. URL: https://en.wikipedia.org/wiki/Digital_signature. Retrieved February 24, 2017.

[45] Becker, G. (2008). Merkle signature schemes, merkle trees and their cryptanalysis. Ruhr-University Bochum, Tech. Rep.

[46] Seibold, S., & Samman, G. Consensus: Immutable Agreement for the Internet of Value. URL: https://assets.kpmg.com/content/dam/kpmg/pdf/2016/06/kpmg-blockchain-consensus-mechanism.pdf. Retrieved February 27, 2017.

[47] Proof of work. URL: https://en.bitcoin.it/wiki/Proof_of_work. Retrieved February 27, 2017.

[48] Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. ACM Transactions on Programming Languages and Systems (TOPLAS), 4(3), 382-401. DOI: 10.1145/357172.357176.

[49] Garay, J., Kiayias, A., & Leonardos, N. (2015). The bitcoin backbone protocol: Analysis and applications. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 281-310). Springer Berlin Heidelberg. DOI: 10.1007/978-3-662-46803-6_10.

[50] Eyal, I., & Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. In International Conference on Financial Cryptography and Data Security (pp. 436-454). Springer Berlin Heidelberg. DOI: 10.1007/978-3-662-45472-5_28.

[51] Sapirshtein, A., Sompolinsky, Y., & Zohar, A. (2015). Optimal selfish mining strategies in bitcoin. arXiv preprint arXiv:1507.06183. Bibliographic Code: 2015arXiv150706183S.

[52] Nelson, M. The Byzantine Generals Problem. URL: http://mark nelson.us/2007/07/23/byzantine/. Retrieved February 28, 2017.

[53] Bitcoin wiki. Scalability. URL: https://en.bitcoin.it/wiki/Scalability. Retrieved February 28, 2017.

[54] Luu, L., Narayanan, V., Baweja, K., Zheng, C., Gilbert, S., & Saxena, P. (2015). SCP: A Computationally-Scalable Byzantine Consensus Protocol For Blockchains. IACR Cryptology ePrint Archive, 2015, 1168. Version: 20160823:024020.

[55] Ethereum. Wiki. URL: https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ#what-is-proof-of-stake. Retrieved March 01, 2017.

[56] Bentov, I., Gabizon, A., & Mizrahi, A. (2016). Cryptocurrencies without proof of work. In International Conference on Financial Cryptography and Data Security (pp. 142-157). Springer Berlin Heidelberg. DOI: 10.1007/978-3-662-53357-4_10.

[57] Buterin V. Slasher: A punitive proof-of-stake algorithm. URL: https://blog. ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm, Retrieved March 01, 2017.

[58] Kogias, E. K., Jovanovic, P., Gailly, N., Khoffi, I., Gasser, L., & Ford, B. (2016). Enhancing bitcoin security and performance with strong consistency via collective signing. In 25th USENIX Security Symposium (USENIX Security 16) (pp. 279-296). USENIX Association. ISBN 978-1-931971-32-4.

[59] Sawtooth Lake documentation. URL: https://intelledger.github.io/. Retrieved March 03, 2017.

[60] Vukolić, M. (2015). The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In International Workshop on Open Problems in Network Security (pp. 112-125). Springer International Publishing. DOI: 10.1007/978-3-319-39028-4_9.

[61] Stewart, I. (2012). Proof of burn.bitcoin.it.

[62] Ethereum. Ethereum/wiki. URL: https://github.com/ethereum/wiki/wiki White-Paper. Retrieved March 03, 2017.

[63] Ethereum Homestead Documentation. URL: http://www.ethdocs.org/en/latest/. Retrieved March 03, 2017.

[64] Zamfir, V. Introducing Casper "the Friendly Ghost". URL: https://blog. ethereum.org/2015/08/01/introducing-casper-friendly-ghost/. Retrieved March 04, 2017.

[65] Buterin, V. Merkling in Ethereum. URL: https://blog.ethereum.org/2015/11/15/merkling-in-ethereum/. Retrieved March 04, 2017.

[66] Dryja, T. (2014). Hashimoto: I/O bound proof of work.

[67] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 151.

[68] Howard, D. (2012). Node. js for PHP Developers. "O'Reilly Media, Inc.". ISBN: 978-1-449-33360-7.

[69] Node.js. URL: https://nodejs.org/en/. Retrieved March 11, 2017.

[70] Ethereum JSON RPC API. URL: https://github.com/ethereum/wiki/wiki/JSON-RPC. Retrieved March 14, 2017.

[71] Browning, R. C., Baker, E. A., Herron, J. A., & Kram, R. (2006). Effects of obesity and sex on the energetic cost and preferred speed of walking. Journal of Applied Physiology, 100(2), 390-398. DOI: 10.1152/japplphysiol.00767.2005.

[72] Vagata, P., & Wilfong, K. (2014). Scaling the Facebook data warehouse to 300 PB. Facebook, heattu, 30, 2016.

[73] Zyskind, G., & Nathan, O. (2015). Decentralizing privacy: Using blockchain to protect personal data. In Security and Privacy Workshops (SPW), 2015 IEEE (pp. 180-184). IEEE. DOI: 10.1109/SPW.2015.27.

[74] Alpheus Bingham and Dwayne Spradlin (2011). The Long Tail of Expertise. Pearson Education. p. 5. ISBN 9780132823135.

8. APPENDICES

Appendix 1. Transaction Example


```
{
  "jsonrpc":  "2.0",
  "method":  "eth_sendTransaction",
  "params":
  [{
  "from":  "0x77cA9b000Fda6Fe7A6c3F7a63593B102B7c55fF6",
  "to":  "0x89278265892281a3f74FaFE9522f596F183bD6C7",
  "gas":  "0xFDE8",
  "gasPrice":  "0x746BBBDC5",
  "value":  "0x0",
  "data":
  "0x9090e6330000000000000000000000000000000000000000000000000000000000000020000000000000000000000000000000000000000000000000000000000000046f6c690000000000000000000000000000000000000000000000000000000000000000"
  }],
  "id":  1
}
```