

# **FEASIBILITY STUDY USING BLOCKCHAIN TO IMPLEMENT PROOF OF LOCATION**

by

**Kristina Lister-Gruesbeck**

**A Thesis**

*Submitted to the Faculty of Purdue University*

*In Partial Fulfillment of the Requirements for the Degree of*

**Master of Science**



Department of Computer and Information Technology

West Lafayette, Indiana

December 2018

**THE PURDUE UNIVERSITY GRADUATE SCHOOL**  
**STATEMENT OF COMMITTEE APPROVAL**

Dr. Baijian Yang, Chair

Department of Computer and Information Technology

Dr. J. Eric Dietz

Department of Computer and Information Technology

Prof. Anthony H. Smith

Department of Computer and Information Technology

**Approved by:**

Dr. Eric T. Matson

Head of the Graduate Program

My Parents: Thank you for giving me the dump truck when I was the 3 and the radio-controlled car when I was 8 as Christmas Presents. Thank you for telling me that I could be anything that I wanted to be and always believing in me even when I didn't believe in myself.

Mom: Thank you for being my mom, telling me the stories about growing up poor with your three brothers, teaching me about motivation, and always being there even when I didn't think that I needed you (1949-2015).

Harrison Latchaw (Al)- Thank you for all the stories growing up and being one of the few who could see my full potential (1917-1998).

Dad and JoeAnn: Thank you for being the ones who listened and questioned everything.

Michael: Thank you for convincing me to apply to Purdue, starting graduate school before me, being the calm one in my life, having patience with me on the nights that I was stuck late in a lab, or doing research, as well as listening (willingly or not) to me whine, cry, and complain about how research and testing were not going according to planned, and your love and support. WE did it!!!

Angel, and Kaitlyn: Thank you both for being able to make me laugh so many times over the past 23 months when I all I wanted to do was cry.

Matt: Thank you for taking the time to explain network switches and routers to me.

Manufacturers of Stouffers Animal Crackers and Twizzlers Cherry Nibs: I am not positive if this research would have ever been successfully completed if it was not for the combination of these two food items.

David and Debbie- Thank you for you both for your ideas, feedback, and willingness to answer numerous random questions.

Jim Lerums- Thank you for your help as well as all of the tips and ideas that you have given me this semester.

Aaron, Bai Lynn, Cate, Chris, Christine, Ethan, Jerry, Katie, Madeline, Russell, Shannon, and Tae- There is no way I can thank or repay each one of you for how much you have helped this semester so I hope THANK YOU covers it!!!!

Adit, Dean, Jennine, and Nithika- Thank for everything this semester especially for listening to me when I was trying to figure out something.

## **ACKNOWLEDGMENTS**

I wish to gratefully acknowledge my thesis committee for their insightful comments and guidance and my family for their support and encouragement.

I would also like to acknowledge my undergraduate as well as graduate professors that I have had at Purdue who have provided me with several experiences which got me to where I am today.

Professor Smith: Thank you for approving my Small Devices paper in your IOT class because it started my research.

Dr. Yang: I know at some points this semester you were wondering if this thesis was going to be finished on time because I know that I was but I am actually submitting it early!!! Thank you for everything.



## TABLE OF CONTENTS

LIST OF FIGURES . . . . .	ix
LIST OF ABBREVIATIONS . . . . .	xvi
GLOSSARY . . . . .	xviii
ABSTRACT . . . . .	xxviii
CHAPTER 1. INTRODUCTION . . . . .	1
1.1 Background . . . . .	1
1.2 Problem Statement . . . . .	1
1.3 Research Question . . . . .	4
1.4 Significance . . . . .	4
1.5 Assumptions . . . . .	5
1.6 Limitations . . . . .	6
1.7 Delimitations . . . . .	6
1.8 Summary . . . . .	6
CHAPTER 2. REVIEW OF LITERATURE . . . . .	7
2.1 Literature Review Background . . . . .	8
2.1.1 CPS Device Safety . . . . .	10
2.1.2 CPS Device Energy Sustainability . . . . .	11
2.1.3 CPS Security . . . . .	11
2.1.4 Internet of Things . . . . .	14
2.1.5 Differentiating IOT from CPS . . . . .	15
2.1.6 Blockchain . . . . .	16
2.1.7 Hashes and their functions . . . . .	17
2.1.8 Public Blockchain . . . . .	20
2.1.8.1 Etheruem Blockchain . . . . .	20
2.1.8.2 Advantages . . . . .	21
2.1.8.3 Disadvantages . . . . .	21
2.1.9 Private Blockchain . . . . .	21

2.1.9.1	Advantages . . . . .	21
2.1.9.2	Disadvantages . . . . .	22
2.1.10	Consortium or Hybrid Blockchains . . . . .	23
2.1.10.1	Advantages . . . . .	23
2.1.10.2	Disadvantages . . . . .	24
2.1.11	Smart Contracts . . . . .	24
2.1.12	Blockchain Network . . . . .	25
2.1.13	Byzantine fault tolerance . . . . .	26
2.1.14	Tendermint . . . . .	26
2.1.15	FOAM Protocol . . . . .	27
2.1.16	Proof of Location . . . . .	28
2.1.17	Crypto-Spatial Coordinates . . . . .	29
2.2	Completed Research . . . . .	30
2.3	Not Completed Research . . . . .	35
2.3.1	Internet of Things (IoT) . . . . .	35
2.3.2	Digital Notary . . . . .	36
2.3.3	Interorganizational Recordkeeping . . . . .	36
2.3.4	Lightweight financial systems . . . . .	37
2.3.5	Multiparty aggregation . . . . .	38
2.3.6	Provenance tracking . . . . .	38
2.4	Needs . . . . .	39
2.5	Issues . . . . .	40
2.6	Conclusions . . . . .	44
2.7	Summary . . . . .	44
CHAPTER 3. RESEARCH METHODOLOGY . . . . .		45
3.1	Research Approach and Hypothesis . . . . .	45
3.2	Testing Equipment . . . . .	46
3.3	Testing conditions . . . . .	47
3.4	Data and Measurements . . . . .	48
3.5	Threats to Validity . . . . .	48

3.6	Summary . . . . .	49
CHAPTER 4. RESULTS, AND ANALYSIS . . . . .		50
4.1	Introduction . . . . .	50
4.1.1	Equipment Selection . . . . .	50
4.1.2	Feasibility Criteria . . . . .	51
4.1.3	Blockchain Feasibility . . . . .	53
4.1.4	Software Installation Feasibility . . . . .	56
4.1.5	Analysis of Feasibility . . . . .	57
4.2	Testing Environment . . . . .	57
4.2.1	Testing Conditions . . . . .	58
4.3	Discussion . . . . .	58
4.3.1	Data Analysis . . . . .	59
4.3.2	FOAM Dependencies . . . . .	95
4.3.3	Purescript Dependencies . . . . .	96
4.3.4	Haskell Dependencies . . . . .	96
4.3.5	Kubernetes Dependencies . . . . .	97
4.3.6	How the dependencies work together . . . . .	97
4.3.7	FOAM Map . . . . .	99
4.3.8	FOAM Technical Architecture . . . . .	99
4.3.9	How does the FOAM Map Work . . . . .	102
4.3.10	Successes . . . . .	103
4.3.10.1	Ethereum Raspberry Pi 3B . . . . .	103
4.3.10.2	Ethereum Raspberry Pi 3B+ . . . . .	104
4.3.10.3	Tendermint Raspberry Pi 3B . . . . .	104
4.3.10.4	Tendermint Raspberry Pi 3B+ . . . . .	104
4.3.10.5	Ethereum Virtual Raspberry Pi . . . . .	104
4.3.10.6	Tendermint Virtual Raspberry Pi . . . . .	105
4.3.10.7	Tendermint Virtual Server . . . . .	105
4.3.11	Lessons Learned . . . . .	105
4.3.11.1	Ethereum Raspberry Pi 3B . . . . .	105

4.3.11.2	Ethereum Raspberry Pi 3B+ . . . . .	106
4.3.11.3	Tendermint Raspberry Pi 3B . . . . .	106
4.3.11.4	Tendermint Raspberry Pi 3B+ . . . . .	107
4.3.11.5	Ethereum Virtual Raspberry Pi . . . . .	107
4.3.11.6	Tendermint Virtual Raspberry Pi . . . . .	108
4.3.11.7	Tendermint Virtual Server . . . . .	108
4.4	Procedures . . . . .	109
4.4.1	Ethereum Raspberry Pi 3B and Raspberry Pi 3B+ . . . . .	109
4.4.2	Tendermint Raspberry Pi 3B and Raspberry Pi 3B+ . . . . .	110
4.4.3	Ethereum Virtual Raspberry Pi . . . . .	110
4.4.4	Tendermint Virtual Raspberry Pi . . . . .	111
4.4.5	Virtual Tendermint Ubuntu Server . . . . .	112
4.5	Results . . . . .	112
4.5.1	Raspberry Pi 3B and Raspberry Pi 3B+ Issues . . . . .	112
4.5.2	Tendermint . . . . .	114
4.5.3	Ethereum Raspberry Pi 3B and Raspberry Pi 3B+ Part 2 . . . . .	115
4.5.4	FOAM Token-Server Implementation . . . . .	127
4.6	Summary . . . . .	130
CHAPTER 5. RECOMMENDATIONS, AND CONCLUSIONS . . . . .		131
5.1	Introduction . . . . .	131
5.2	Summary . . . . .	131
5.3	Recommendations and Future Work . . . . .	135
5.4	Conclusions . . . . .	137
REFERENCES . . . . .		138
APPENDIX A. . . . .		150
A.1	Introduction . . . . .	150
A.1.1	Code that was used to collect data . . . . .	150

## LIST OF FIGURES

4.1	Chart Comparing the Raspberry Pi 3B and Raspberry Pi 3B+ . . . . .	51
4.2	Image of a chart comparing various blockchains including EOS, Ethereum, Cardano, Tendermint, and Tomochain . . . . .	54
4.3	Network Diagram of the Testing environment . . . . .	59
4.4	Overhead of testing Environment . . . . .	60
4.5	Screenshot showing some of the higher temperatures that were observed on 9/23 on the Raspberry Pi 3B . . . . .	63
4.6	Screenshot showing some of the higher temperatures that were observed on 9/23 on the Raspberry Pi 3B . . . . .	63
4.7	Screenshot showing some of the higher temperatures that were observed on 9/23 on the Raspberry Pi 3B . . . . .	64
4.8	Temperature Spike on the Raspberry Pi 3B . . . . .	65
4.9	Screenshot showing some of the higher temperatures that were documented on the Raspberry Pi 3B+ . . . . .	66
4.10	Temperature Spike on the Raspberry Pi 3B+ . . . . .	67
4.11	Image of a graph comparing the temperatures readings of the Raspberry Pi 3B and the Raspberry Pi 3B+ . . . . .	68
4.12	Image of a chart comparing the voltage outputs of the Raspberry Pi 3B and the Raspberry Pi 3B+ . . . . .	69
4.13	Graph showing the time and temperature of readings that were taken on 10/20 for the Raspberry Pi 3B . . . . .	70
4.14	Graph showing the time and voltage of readings that were taken on 10/20 for the Raspberry Pi 3B . . . . .	70
4.15	Graph showing the time and temperature of readings that were taken on 10/21 for the Raspberry Pi 3B . . . . .	70
4.16	Graph showing the time and voltage of readings that were taken on 10/21 for the Raspberry Pi 3B . . . . .	71

4.17 Graph showing the time and temperature of readings that were taken on 10/22 for the Raspberry Pi 3B . . . . .	71
4.18 Graph showing the time and voltage of readings that were taken on 10/22 for the Raspberry Pi 3B . . . . .	71
4.19 Graph showing the time and temperature of readings that were taken on 10/23 for the Raspberry Pi 3B . . . . .	72
4.20 Graph showing the time and voltage of readings that were taken on 10/23 for the Raspberry Pi 3B . . . . .	72
4.21 Graph showing the time and temperature of readings that were taken on 10/24 for the Raspberry Pi 3B . . . . .	72
4.22 Graph showing the time and voltage of readings that were taken on 10/24 for the Raspberry Pi 3B . . . . .	73
4.23 Graph showing the time and temperature of readings that were taken on 10/25 for the Raspberry Pi 3B . . . . .	73
4.24 Graph showing the time and voltage of readings that were taken on 10/25 for the Raspberry Pi 3B . . . . .	73
4.25 Graph showing the time and temperature of readings that were taken on 10/26 for the Raspberry Pi 3B . . . . .	74
4.26 Graph showing the time and voltage of readings that were taken on 10/26 for the Raspberry Pi 3B . . . . .	74
4.27 Graph showing the time and temperature of readings that were taken on 10/27 for the Raspberry Pi 3B . . . . .	74
4.28 Graph showing the time and voltage of readings that were taken on 10/27 for the Raspberry Pi 3B . . . . .	75
4.29 Graph showing the time and temperature of readings that were taken on 10/28 for the Raspberry Pi 3B . . . . .	75
4.30 Graph showing the time and voltage of readings that were taken on 10/28 for the Raspberry Pi 3B . . . . .	75
4.31 Graph showing the time and temperature of readings that were taken on 10/29 for the Raspberry Pi 3B . . . . .	76

4.32 Graph showing the time and voltage of readings that were taken on 10/29 for the Raspberry Pi 3B . . . . .	76
4.33 Graph showing the time and temperature of readings that were taken on 10/30 for the Raspberry Pi 3B . . . . .	76
4.34 Graph showing the time and voltage of readings that were taken on 10/30 for the Raspberry Pi 3B . . . . .	77
4.35 Graph showing the time and temperature of readings that were taken on 10/31 for the Raspberry Pi 3B . . . . .	77
4.36 Graph showing the time and voltage of readings that were taken on 10/31 for the Raspberry Pi 3B . . . . .	77
4.37 Graph showing the time and temperature of readings that were taken on 11/1 for the Raspberry Pi 3B . . . . .	78
4.38 Graph showing the time and voltage of readings that were taken on 11/1 for the Raspberry Pi 3B . . . . .	78
4.39 Graph showing the time and temperature of readings that were taken on 11/2 for the Raspberry Pi 3B . . . . .	78
4.40 Graph showing the time and voltage of readings that were taken on 11/2 for the Raspberry Pi 3B . . . . .	79
4.41 Graph showing the time and temperature of readings that were taken on 11/3 for the Raspberry Pi 3B . . . . .	79
4.42 Graph showing the time and voltage of readings that were taken on 11/3 for the Raspberry Pi 3B . . . . .	79
4.43 Graph showing the time and temperature of readings that were taken on 11/4 for the Raspberry Pi 3B . . . . .	80
4.44 Graph showing the time and voltage of readings that were taken on 11/4 for the Raspberry Pi 3B . . . . .	80
4.45 Graph showing the time and temperature of readings that were taken on 11/5 for the Raspberry Pi 3B . . . . .	80
4.46 Graph showing the time and voltage of readings that were taken on 11/5 for the Raspberry Pi 3B . . . . .	81

4.47	Graph showing the time and temperature of readings that were taken on 11/6 for the Raspberry Pi 3B . . . . .	81
4.48	Graph showing the time and voltage of readings that were taken on 11/6 for the Raspberry Pi 3B . . . . .	81
4.49	Graph showing the time and temperature of readings that were taken on 10/20 for the Raspberry Pi 3B+ . . . . .	82
4.50	Graph showing the time and voltage of readings that were taken on 10/20 for the Raspberry Pi 3B+ . . . . .	82
4.51	Graph showing the time and temperature of readings that were taken on 10/21 for the Raspberry Pi 3B+ . . . . .	82
4.52	Graph showing the time and voltage of readings that were taken on 10/21 for the Raspberry Pi 3B+ . . . . .	83
4.53	Graph showing the time and temperature of readings that were taken on 10/22 for the Raspberry Pi 3B + . . . . .	83
4.54	Graph showing the time and voltage of readings that were taken on 10/22 for the Raspberry Pi 3B+ . . . . .	83
4.55	Graph showing the time and temperature of readings that were taken on 10/23 for the Raspberry Pi 3B + . . . . .	84
4.56	Graph showing the time and voltage of readings that were taken on 10/23 for the Raspberry Pi 3B+ . . . . .	84
4.57	Graph showing the time and temperature of readings that were taken on 10/24 for the Raspberry Pi 3B+ . . . . .	85
4.58	Graph showing the time and voltage of readings that were taken on 10/24 for the Raspberry Pi 3B+ . . . . .	85
4.59	Graph showing the time and temperature of readings that were taken on 10/25 for the Raspberry Pi 3B+ . . . . .	85
4.60	Graph showing the time and voltage of readings that were taken on 10/25 for the Raspberry Pi 3B+ . . . . .	86
4.61	Graph showing the time and temperature of readings that were taken on 10/26 for the Raspberry Pi 3B+ . . . . .	86



4.62	Graph showing the time and voltage of readings that were taken on 10/26 for the Raspberry Pi 3B+ . . . . .	86
4.63	Graph showing the time and temperature of readings that were taken on 10/27 for the Raspberry Pi 3B + . . . . .	87
4.64	Graph showing the time and voltage of readings that were taken on 10/27 for the Raspberry Pi 3B+ . . . . .	87
4.65	Graph showing the time and temperature of readings that were taken on 10/28 for the Raspberry Pi 3B+ . . . . .	87
4.66	Graph showing the time and voltage of readings that were taken on 10/28 for the Raspberry Pi 3B+ . . . . .	88
4.67	Graph showing the time and temperature of readings that were taken on 10/29 for the Raspberry Pi 3B+ . . . . .	88
4.68	Graph showing the time and voltage of readings that were taken on 10/29 for the Raspberry Pi 3B+ . . . . .	88
4.69	Graph showing the time and temperature of readings that were taken on 10/30 for the Raspberry Pi 3B+ . . . . .	89
4.70	Graph showing the time and voltage of readings that were taken on 10/30 for the Raspberry Pi 3B+ . . . . .	89
4.71	Graph showing the time and temperature of readings that were taken on 10/31 for the Raspberry Pi 3B+ . . . . .	89
4.72	Graph showing the time and voltage of readings that were taken on 10/31 for the Raspberry Pi 3B+ . . . . .	90
4.73	Graph showing the time and temperature of readings that were taken on 11/1 for the Raspberry Pi 3B+ . . . . .	90
4.74	Graph showing the time and voltage of readings that were taken on 11/1 for the Raspberry Pi 3B+ . . . . .	90
4.75	Graph showing the time and temperature of readings that were taken on 11/2 for the Raspberry Pi 3B+ . . . . .	91
4.76	Graph showing the time and voltage of readings that were taken on 11/2 for the Raspberry Pi 3B+ . . . . .	91

4.77 Graph showing the time and temperature of readings that were taken on 11/3 for the Raspberry Pi 3B+ . . . . .	91
4.78 Graph showing the time and voltage of readings that were taken on 11/3 for the Raspberry Pi 3B+ . . . . .	92
4.79 Graph showing the time and temperature of readings that were taken on 11/4 for the Raspberry Pi 3B+ . . . . .	92
4.80 Graph showing the time and voltage of readings that were taken on 11/4 for the Raspberry Pi 3B+ . . . . .	92
4.81 Graph showing the time and temperature of readings that were taken on 11/5 for the Raspberry Pi 3B+ . . . . .	93
4.82 Graph showing the time and voltage of readings that were taken on 11/5 for the Raspberry Pi 3B+ . . . . .	93
4.83 Graph showing the time and temperature of readings that were taken on 11/6 for the Raspberry Pi 3B+ . . . . .	93
4.84 Graph showing the time and voltage of readings that were taken on 11/6 for the Raspberry Pi 3B+ . . . . .	94
4.85 FOAM Architecture . . . . .	100
4.86 Technical Architecture Layers . . . . .	101
4.87 Chart comparing the voltages on the Raspberry Pi 3B and Raspberry Pi 3B+ . . . .	113
4.88 Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B was set equal to 32 MB, 64 MB, and 128 MB . . . . .	117
4.89 Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B was set equal to 256 MB, 512 MB, and 1024 MB . . . . .	118
4.90 Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B was set equal to 1024 MB . . . . .	119
4.91 Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B+ was set equal to 32 MB, 64 MB, and 128 MB . . . . .	120
4.92 Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B was set equal to 256 MB, 512 MB, and 1024 MB . . . . .	121

4.93	Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B was set equal to 2048 MB . . . . .	122
4.94	Incremental Time and Temperature Data that was obtained on the Raspberry Pi 3B . . . . .	123
4.95	Incremental Time and Temperature Data that was obtained on the Raspberry Pi 3B+ . . . . .	123
4.96	Incremental Time and Voltage Data that was obtained on the Raspberry Pi 3B . . . . .	124
4.97	Incremental Time and Voltage Data that was obtained on the Raspberry Pi 3B+ . . . . .	124
4.98	Image of a chart showing all of the data that was collected on the Raspberry Pi 3B during a temperature spike . . . . .	128
4.99	Image of a chart comparing all of the data that was collected on the Raspberry Pi 3B+ during a temperature spike . . . . .	129

## LIST OF ABBREVIATIONS

ABCI	Application BlockChain Interface
BLE	Bluetooth Low Energy
APIs	Application Programming Interface
BAN	Body Area Network
BFT	Byzantine Fault Tolerance
CCTV	Closed-circuit television
CSC	Crypto-Spatial Coordinate
CPS	Cyber-physical systems
dApps	Decentralized applications
DDoS	Distributed Denial-of-Service Attack
DNS	Domain Name Servers
DVR	Digital Video Recorder
EDCCS	Executable Distributed Code Contract
ERC	Ethereum Request for Comment
FFG	Friendly Finality Gadget
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
GSM	Global System for Mobile communication
HTTP	HyperText Transfer Protocol
ICO	Initial Coin Offering
IoT	Internet of Things
IP	Internet Protocol
IPFS	InterPlanetary File System
IT	Information Technology

LPWAN	Low Power Wide Area Networks
LTE	Long-Term Evolution
NID	Network Intrusion Detection
OCX	Operational Control System
ORDBMS	Object-Relational Database Management System
OS	Operating System
OSM	OpenStreetMap
P2P	Peer-to-peer
PGP	Pretty Good Privacy
PLCS	Programmable Logic Controllers
PKI	Public Key Infrastructures
POI	Point of Interest
POW	Proof of Work
ROI <sub>m</sub>	Region of Impact
ROI <sub>n</sub>	Region of Interest
RTUs	Remote Telemetry Units
SCADA	Supervisory control and data acquisition systems
SI	Smart Infrastructure
SIV	Spatial Index and Visualizer
SLAs	Service Level Agreement
SQL	Structured Query Language
SSH	Secure Shell
TCRs	Token Curated Registries
USB	Universal Serial Bus
UTXO	Unspent Transaction Output

## GLOSSARY

**Actors-** Within the FOAM TCR there are three kinds of actors which are Consumers, candidates, and cartographers ("Foamspace Corp", 2018b).

**Application Blockchain Interface (ABCI)-** It is an interface that allows transactions to be completed using any programming language (Rehman, 2018).

**Asset-** Can be tangible or an intangible item (Gupta, 2017).

**Blockchain-** "is a shared, distributed ledger that facilitates the process of recording transactions and tracking assets in a business network" (Gupta, 2017, p.3). It is also "a technology that allows people who don't know each other to trust a shared record of events" which is distributed to all of the network participants who will use "their computers to validate transactions" and therefore ensure that there is no need for a third party intermediary ("Foamspace Corp", 2018b, p. 3). It can be divided into three different types which are public, private, and consortium or hybrid (Deloitte LLP, 2016).

**Blockchain Network-** a network that has the ability to be able to track or trade virtually anything that has a value which will decrease the risks as well as cut costs for all of those that are involved (Gupta, 2017).

**Candidates-** One of the three kinds of actors that are found within the FOAM TCR which their goal is "to be on the list" ("Foamspace Corp", 2018b, p. 14).

**Cartographers-** One of the three kinds of actors that are found within the FOAM TCR and they are the curators of the list as well as being the FOAM Token Holders ("Foamspace Corp", 2018b).

**Clients-** Machines that can interact with servers that are local or remote via terminals. They are used to monitor the conditions of the network as well as having the ability to be able to stop and start processes that are being executed within the network (Slay & Miller, 2007).

**Consensus Engine-** Ensures that every transaction is documented in the same order to all of the connected machines using the Proof of Stake consensus (Rehman, 2018).

Consumers- One of the three kinds of actors that are found within the FOAM TCR and they are the users “that want to utilize the list” (“Foamspace Corp”, 2018b, p. 14).

Controllers- They are used to perform actions based upon the sensor data and control algorithms (Slay & Miller, 2007).

Crypto-Spatial Coordinates (CSC)- is “an open and interoperable standard for location” services “in ‘Ethereum smart contracts’ that supports the use of a variety of hardware and software (Josefsson, 2017, para. 2). It is a starting point for a shared location standard that permits any smart contract to make an immutable claim to an address on the blockchain as well as a corresponding physical location on the map that can be verified on- or off-chain (Josefsson, 2017, para. 4).

Cyber-Physical Systems (CPS) - They are used to create a connection between physical objects and their computing resources (Chadwick, Betzig, & Hu, 2011). They are also used for the integration and processing of computations, and the creation of networks. The monitoring and controlling of the physical processes are completed using embedded computers and networks which contain feedback loops when the physical processes are affected by computations as well as having the computations affect the physical processes. There are several vast and unrealized economic and societal impacts that CPS provides and they have only recently begun to receive financial support. CPS builds upon the firmly established technology of embedded systems, computers as well as the software that is embedded in the system which have primary uses other than computation. The integration of the physical process properties with the networking and software while providing a simulated model containing a design and an analysis of the techniques that are used to integrate the processes together (Asare et al., 2012).

ERC20 – is an address scheme that uses a technical standard for implementing tokens that is “used for smart contracts on the Ethereum Blockchain” (*ERC-20*, 2018; *ERC-20 Token Standard*, 2018, para. 1). “ERC stands for Ethereum Request for Comment, and 20 is the number that was assigned to this request” (*ERC-20*, 2018, para. 1). It was proposed by

Fabian Vogelsteller on November 19, 2015 (*ERC-20*, 2018; “ERC: Token standard #20”, 2015). ERC-20 is used to define “a common list of rules that an Ethereum token” is required “to implement, giving developers the ability to program how new tokens will function within the Ethereum ecosystem” (*ERC-20*, 2018, para. 1). According to Etherscan there were “more than 103,621 ERC-20 token Contracts as of July 26th 2018” (*ERC-20*, 2018, para. 4) (*Etherscan Token Tracker Page*, 2018).

Field devices- Include sensors and a variety of controllers including remote telemetry units (RTUs) and programmable logic controllers (PLCS) (Slay & Miller, 2007).

FOAM- is a proof of location protocol that permits an “autonomous network of radio beacons that” are permissionless to offer a variety of secure independent location services as well as “independent of external centralized sources such as GPS through time synchronization” (*The Future of Proof of Location*, 2018, para. 1). Additionally it “is an open protocol for decentralized, geospatial data markets”, and it “is designed to empower users to build a consensus-driven map of the world that can be trusted for every application” (“Foamspace Corp”, 2018b, p. 2).

Global Navigation Satellite System (GNSS)- Consists of 31 satellites that are available to be used by civilians as well as commercial users and they have been launched by the United States military (“Foamspace Corp”, 2018b).

Haskell- is a programming language that is purely functional and similar to other functional languages its functions have no side-effects meaning that the only task a function can complete is to calculate a value and return it as a result. Unless it is told to execute a particular function, it will not complete the task on its own, and must be specifically told to do so (Lipovača, 2011; Snoyman, 2015). Haskell is also statically typed which means that it will complete the type checking of a script at compile time and it will not compile until all of the errors that are found within the script are corrected (Lipovača, 2011). Additionally, Haskell’s type system supports type inference which means that not every piece of code needs to have its type specified because the typing system is able to



determine the appropriate resulting value based upon the information that it is provided (Lipovača, 2011; Snoyman, 2015).

Hybrid/Consortium Blockchain - Blockchains which have a group of preselected nodes that are used in the consensus process (Buterin, 2015). Consortium blockchains are also thought of as being hybrid blockchains because they are partially decentralized and do not have the stringent separation requirements that have been established for public and private blockchains (Pilkington, 2015). Consortium blockchains must have one of three types of rights to be able to read it which are public, limited to the participants, or hybrid routes. One example of a hybrid route is where the root hashes from the blocks and API are publicly shared together in order to allow public members of the consortium blockchain to be able to make a certain number of queries and get the cryptographic proofs of certain parts of the blockchain status. Consortium blockchains could be thought of as being “partially decentralized” (Buterin, 2015, para. 4). Recently there has been some confusion about the difference between private and consortium blockchains but consortium blockchains are somewhat private (Thompson, 2016).

Intangible assets- An asset that cannot be physically touched such as intellectual property including “patents, copyrights, or branding” (Gupta, 2017, p. 3).

Low Power Wide Area Networks (LPWAN)- Wireless technology that is able to operate efficiently long range as well as in a low power situation on an unlicensed radio spectrum (“Foamspace Corp”, 2018b; Raza, Kulkarni, & Sooriyabandara, 2017). It has a potentially lower data speed as well as a higher latency than other wireless connection options (“Foamspace Corp”, 2018b; Raza et al., 2017).

Main-Net Node- It is the main or primary network where the transactions occur on a distributed ledger (*Glossary - Mainnet*, 2018, para. 1). Additionally the mainnet is a blockchain that is specifically tasked with moving the digital currency from the senders to the recipients (Rhodes, 2018, para. 2).

**MetaMask Account-** These are accounts that permit users to be able to access decentralized applications (dApps) that run on Etheruem “such as the Spatial Index right from” their web browser without needing to run “a full Ethereum node” (“Foamspace Corp”, 2018b, p. 11). It also includes “a secure identity vault,” which will provide users the ability to be able to manage their identities on different websites and sign blockchain transactions (“Foamspace Corp”, 2018b). It is an extension that is available to be installed on a variety of popular browsers including “Chrome, Firefox, Opera,” as well as the new Brave browser (“Foamspace Corp”, 2018b, p. 11).

**Node.JS-** an open source software that was originally designed to be used to write internet applications that are highly scalable such as web application servers (Cuomo, 2013).

**NOOBS-** a software installation manager that contains the standard operating systems of Raspbian as well as LibreELEC and supports the ability to be able to download alternative operating systems that are also supported on a Raspberry Pi (*Download NOOBS for Raspberry Pi*, 2018).

**OpenStreetMap (OSM)-** It is collaborative open source mapping project, alternative to Google, free to use, and supports the use of “other proprietary mapping data” (“Foamspace Corp”, 2018b, p. 5). It has millions of creators located around the world. It’s continued improvement means that the value of the mapping data that Google can provide will rapidly decrease. There is one negative aspect of OpenStreetMap which it does not make it easy to enforce previously agreed upon principles which is a hindrance to future development of the blockchain infrastructure because it is a location standard that is needed for a variety of blockchain applications. Since it “is free, open source, and interoperable” there are a variety of protocols that are able to “securely connect offline spaces to online assets” (“Foamspace Corp”, 2018b, p. 5).

**Point of Interest (POI)-** The combination of CSCs and TCRs are used to create a new type “of mapping and maintaining” (“Foamspace Corp”, 2018b, p. 13). They are the most valuable points on a map because they show the locations of “stores, cafes, restaurants and

mall,” as well as “where a fleet of vehicles in a ride sharing program like Uber should” anticipate a change in demand, or the location of some traffic bottlenecks that drivers should be aware of (“Foamspace Corp”, 2018b, p. 14).

PostgreSQL- It is “an object-relational database management system (ORDBMS) based on POSTGRES, Version 4.2, developed at the University of California” in the “Berkeley Computer Science Department” (*What is PostgreSQL?*, 2018, para. 1). Additionally, PostgreSQL is “an open-source descendant of” the “original Berkeley code”, and it is able to support a majority of the standard SQL commands as well as a several modern features including “complex queries, foreign keys, triggers, updatable views,”, and provide “transactional integrity as well as multiversion concurrency control” (*What is PostgreSQL?*, 2018, para. 2). There are several ways that users are able to extend PostgreSQL such as “by adding new data types, functions, operators, aggregate functions, index methods” as well as new procedural languages (*What is PostgreSQL?*, 2018, para. 3). Lastly, since the license for PostgreSQL is open-source this means that anyone is able to modify, use, and distribute the software for free no matter the desired purpose (*PostgreSQL 11.0 Documentation*, 2018).

Presence Claim- The digital authentication certificate that has the ability to provide verified of proof of location as to whether an agent or an event is or was at a particular location “at a certain point in time and space” (“Foamspace Corp”, 2018b, p. 16).

Private blockchain - Private blockchains have the most restrictions on them because the ability to be able to write to the blockchain is kept in a location that is central to one organization (Buterin, 2015). Private blockchains are closely analogized with permissioned ledgers which allows for organizational processes which will allow user ids to be whitelisted or blacklisted (Pilkington, 2015). The public may or may not have the ability to be able to read the information that is contained with a private blockchain. A few of the ways that private blockchains can be used are for database management, or completing an internal audit of a company which would mean that the public would not need to have the ability

to be able to access the information but there cases where having public accountability would be a good thing (Buterin, 2015).

**Proof of Location-** Allows users as well as self-governing agents the ability to be able to authenticate their location data when they choose too as well as revealing their personal information at a later time and/or date by providing a location claim that is fraud-proof (King, 2018a). It uses Crypto-Spatial Coordinates (CSC) to create a claim to a physical location and an address on the smart contract for that blockchain (Josefsson, 2017).

**Public blockchain -** Blockchains that anyone can read as well as send transactions too (Buterin, 2015). The public nature of public blockchains originates in the restriction free participation that it permits in order to determine what blocks are appended to the chain as well as determining the current state of the chain (Pilkington, 2015). After the transactions have been validated a receipt of a blockchain transaction should expect to see them included in the chain. In addition to being able to read the blockchain, anyone in the world can participate in the consensus process. The consensus process is a process that is used to determine what blocks will be added to the chain as well as determining what is its current status. Since public blockchains are frequently used to substitute for trusts that are centralized in nature they are secured with cryptoeconomics which is a combination of cryptographic verification and economic incentives which will use either proof of stake or proof of work to validate the information. For a blockchain to follow the universal principle that states the degree to which a person is able to influence the consensus process is related to the quality of the economic resources that they are able to provide. Public blockchains are usually thought to be “fully decentralized” (Buterin, 2015, para. 4).

**Purescript-** is functional programming language that is strongly typed and is able to be compiled to JavaScript (*PureScript*, 2017).

**Raspbian** is the standard operating system of the Raspberry Pi.

**Sensors-** They are used to collect data from a variety of sources (Slay & Miller, 2007).

**Servers-** They are responsible for collecting and analyzing the various field inputs as well as creating alerts, starting and stopping processes and enforcing the logic that is required to automate the control processes (Slay & Miller, 2007).

**Signal-** it “is a mechanism that is designed to” permit cartographers to be encouraged to grow the FOAM network by staking FOAM tokens to a particular location to demonstrate the demand for Proof of location as well as improve the geographic coverage of the FOAM network (“Foamspace Corp”, 2018b, p. 15).

**Smart Contracts-** Self-executing contracts that provide the details of a deal that was reached between two or more people (typically a buyer and a seller) that is written directly into the code. The most commonly known smart contract is Ethereum (*Smart Contracts Definition* — *Investopedia*, 2017).

**Spatial Index-**It is currently being used to visualize the Proof of Location Protocol, but it also has the ability to be a front-end or user facing decentralized application which has the ability to create a visual representation of smart contracts on a map (*FOAM Spatial Index*, 2018; King, 2018b, para. 1). It is thought of as a “cross between Google Maps and a Bloomberg trading terminal” (*FOAM Spatial Index*, 2018, para. 1). It also has two primary job duties the first one is to permit users to collaborate, understand, engage with as well as act upon CSC contracts. The second duty of the spatial index is to “serve as the foundation of the large variety of decentralized applications that can be built on top of our protocols” ((King, 2018b, para. 1), (*FOAM Spatial Index*, 2018, para. 1)).

**Spatial Index and Visualizer (“SIV”)-** When the spatial index and visualizer are combined together they create a front-end interface (or the user facing interface) that is used by a variety of decentralized applications to see the locations of smart contracts on a map (“Foamspace Corp”, 2018b, p. 9).

**Supervisory control and data acquisition (SCADA) systems-** Systems that are used for collecting real-time data, monitoring the equipment and controlling the processes that occur in industrial as well as public facilities. Examples include chemical plants, chemical

refineries, electrical power transmission and generation systems, oil and gas pipelines, sewage and water treatment plants (Slay & Miller, 2007).

**Tangible Assets-** physical assets that can be seen and/or felt such as “a house, a car, cash, or land” (Gupta, 2017, p. 3).

**Testnet-** This is “where new dApps and EDCCs can be tested and developed” (*Glossary - Mainnet*, 2018, para. 1).

**Token Curated Registries (TCRs)-** it is a crypto-economic model that is used to create lists that can be read by humans and contain underlying financial “incentives for independent token holders to” create the contents of the list (“Foamspace Corp”, 2018b, p. 13).

**Verifiers-** are the computers that are used to check the zones and ensure that there is no fraud being committed with the location of a person or thing as well as use the time data to calculate the location algorithms. They work with a specific zone or nearby zones to permit the mining of triangulations that will be used to verify a location as well as the resulting data output that is used to compute additional triangulations and in exchange for this process these actors are eligible to receive the newly issued tokens (“Foamspace Corp”, 2018b).

**Visualizer-** permits users to interact with the smart contracts that are referenced by the SIV in a normal web “browser via an independent identity solution” such as Metamask or uPort (“Foamspace Corp”, 2018b, p. 11).

**Zone-** a particular area that has been mapped and contains zone anchors and zone authorities (“Foamspace Corp”, 2018b).

**Zone Anchors-** The development of “an overlaying peer-to-peer network of radio beacons” that are specialized hardware beacons that were setup by zone operators to increase participation in the protocol by broadcasting coverage (“Foamspace Corp”, 2018b, p. 17). Additionally, zone anchors are able to achieve a consensus to determine if something was at a particular location at a “certain point in time and space based” upon the radio beacons (“Foamspace Corp”, 2018b, p. 17).

Zone Authorities- They are the radio gateways and they are used to establish zones once the synchronization between them and the zone anchors has been completed. They also guarantee that they will offer location services that will use smart contract safety deposits to enforce the agreement ("Foamspace Corp", 2018b, p. 21).

## ABSTRACT

Author: Lister-Gruesbeck, Kristina. M.S.

Institution: Purdue University

Degree Received: December 2018

Title: Feasibility Study Using Blockchain to Implement Proof of Location

Major Professor: Baijian Yang

The purpose of this thesis is to determine the feasibility of using blockchain to implement proof of location. There has been an increasing demand for a way to create a validated proof of location that is economical, and easy to deploy as well as portable. There are several reasons for an increased demand in this technology including the ever-increasing number of mobile gamers that have been able to spoof their location successfully, the increasing number of on demand package shipments from companies such as Amazon, and the desire to reduce the occurrence of medical errors as well as holding hospitals accountable for their errors. Additional reasons that this technology is gaining popularity and increasing in demand is due the continually increasing number of lost baggage claims that airlines are receiving, as well as insurance companies desire to reduce the number of fraud cases that are related to high-value goods as well as increasing the probability of their recovery. Within the past year, there has been an extensive amount of research as well as work that has been completed to create an irrefutable method of location verification, which will permit a user to be able to create time-stamped documentation validating that they were at a particular location at a certain day and time. Additionally, the user is then permitted to release the information at a later date and time that is convenient for them. This research was completed using a Raspberry Pi 3B, a Raspberry Pi 3B+, two virtual Raspberry Pi's as well as two virtual servers in which the goal was to download, and setup either Ethereum and/or Tendermint Blockchain on each piece of equipment. After completely synchronizing the blockchain it be used to store the verified location data that been time-stamped. There was a variety of issues that were encountered during the setup and installation of the blockchains on the



equipment including overclocking processors, which negatively affected the computational abilities of the devices as well as causing overheating and surges in voltage as well as a variety of software and hardware incompatibilities. These issues when looked at individually appear to not have much of an impact on the results of this research but when combined together it is obvious that they reduced the results that could be obtained. In conclusion, the combination of hardware and software issues when combined with the temperature and voltage issues that were due to the overheating of the processor resulted in several insurmountable issues that could not be overcome. There are several recommendations for continuing this work including presyncing the blockchain using a computer, using a device that has more functionality and computational abilities, connecting a cooling device such as a fan or adding a heat sink, increasing the available power supply, utilizing an externally power hard drive for data storage, recreate this research with the goal in mind of determining what process or application was causing the high processor usage, or creating a distributed system that utilizes both physical and virtual equipment to reduce the amount of work on one type of device.

## **CHAPTER 1. INTRODUCTION**

This chapter provides an overview of the research study. It introduces the research that has been completed on cyber-physical systems (CPS), blockchain, and network security by presenting a background of the problem areas and research question. In addition, it covers the research significance, assumptions, limitations and delimitations which define the extent of the study.

### 1.1 Background

The purpose of this research and thesis is to determine what improvements can be made to the existing security of CPS when a blockchain is implemented. The blockchain will be used to detect unauthorized changes to information as well as unauthorized attempts to gain access to a particular CPS, its device, or the network on which it is found. The unauthorized access attempt will be confirmed by the other connected CPS because their portions of the blockchain will not contain the modifications that are found in the attacked system. The proposed idea is limited to building upon the existing security of CPS by implementing a blockchain. Upon completion; this project will be able to be implemented in a variety of CPS's to aide in improving the security of the systems by decreasing the successful physical attacks while maintaining an efficiently running CPS. Currently it is not known what will be the procedure for securing the system that was attacked.

### 1.2 Problem Statement

Recently there has been an increase in the number of successful attacks that have been directed at a variety of CPSs including those devices that are used by power plants, central transit systems, industry production environments, and smart meters (Montalbano, 2017). Due to the negative impacts that are being felt across a variety of industries, there has been an increase in the

need for improving the security of existing CPSs as well as those that are being developed. The National CPS to be:

Cyber-Physical Systems or smart systems are co-engineered interacting networks of physical and computational components. These systems will provide the foundation of our critical infrastructure, form the basis of emerging and future smart services, and improve our quality of life in many areas (Sanger, 2016, para. 1).

There are several unavoidable challenges that have been previously encountered when attempting to secure a CPS or its data. They include disgruntled employees, data spillage, hackers, and cyber bombs. The reason that a disgruntled employee would pose a challenge to securing the data is they could voluntarily share information about the CPSs data with an unauthorized person. Data spillage poses a challenge when the data of a higher classification is accidentally moved to an area of lower classification, thus allowing those who do not have authorization to view the information. The third risk to the security of CPS devices are hackers because once they can gain access to a system there is very little that can be done to prevent them from retaining access to that system in question. Additionally, if a hacker is able to be locked out of a system it is very possible that they will be able to regain access to that system. The final unavoidable challenge that has been recently encountered are cyberbombs, which are a combination of software and hardware that can be used to disrupt or physically damage a variety of technological devices including communications platforms, wired and wireless networks, and physical servers (Lloyd, 2015).

The four security challenges mentioned above are not the only reasons that the data security of CPS has been recently brought into question but they are a sizable portion of the problem. Recently NIST has been researching what improvements can be made to the security of CPS and one of the research areas that has not received much attention has been the use blockchain to aid in the monitoring of the data from the CPS as well as its security (Forno & Joshi, 2016; Sanger, 2016). The issue that this research addresses is which of the available blockchains will best improve the security of CPS and how they transport data?

Blockchain is a digital ledger that contains the transactional history of the various operations that it has processed (*Blockchain*, 2017). In addition to being a digital ledger, blockchain is able to confirm the integrity of a particular piece of information or data as well as validate that it has not been modified (Ivezic, 2016).

There are three types of blockchains; they are public, private, and consortium blockchains. Public blockchains were originally designed to reduce the chances of a middleman being needed in order to complete a peer-to-peer exchange of assets. A public blockchain can be assessed by anyone who has a computer that will support running a node as well as an internet connection that will support the volume of data that will be needed in order to synchronize the entire blockchain history. Due to the numerous amount of servers that are running blockchain, syncing the data is part of the reason that it is a slow and time-consuming task. It is also this redundancy that makes public blockchains extremely secure. Public blockchain servers are decentralized, meaning that they are not stored in one central location but in fact spread out, which also increases the amount of time it takes to synchronize the data (Gupta, 2017).

Private blockchains are centralized blockchains that limit who can make modifications to them by keeping the decision-making central to the location of those who are making the decisions. The company is able to write and verify every transaction, which allows for the transactions to be completed faster and more efficiently. Companies that run private blockchains have the ability to control who has access to the blockchain, which is why the blockchain is primarily used in traditional businesses and governments (Thompson, 2016).

Consortium blockchains are blockchains that fall in between public and private blockchains because they are partly private. There has been some confusion about what a consortium blockchain exactly is and Vitalik Buterin has provided the definition of:

So far there has been little emphasis on the distinction between consortium blockchains and fully private blockchains, although it is important: the former provides a hybrid between the 'low-trust' provided by public blockchains and the 'single highly-trusted entity' model of private blockchains, whereas the latter can be

more accurately described as a traditional centralized system with a degree of cryptographic auditability attached (Buterin, 2015, para. 3).

Consortium blockchains are able to provide several of the same benefits that have been connected with private blockchains including efficiency and transactional privacy while ensuring the one entity does not have all of the modification rights to a blockchain. The management of consortium blockchains has been closely analogized with a council of elders because they allow a predetermined group or section of nodes to make decisions.

### 1.3 Research Question

This research contributes answers for following question:

- Can a Raspberry Pi support implementing the FOAM token protocol?

### 1.4 Significance

It is hoped that the research that is completed while doing this thesis will be used to improve the existing security of CPS as well as its overall performance. By improving the security and the performance of CPS it is theorized that there will be a decrease in the number of successful attacks on the CPS devices and the networks that are used to connect them. Even though there has been an extensive amount of research completed on the security of CPS there have still been attacks that have been successful in physically damaging CPS and hindering the transmission of data. By preventing the transmission of data as well as causing physical damage to the CPS devices there is an increased risk of security issues going undetected. The combination of the blockchains use of records that are securely linked using cryptography will decrease the chances that a data transmission for a CPS device will be interrupted without being noticed because each block of data that is sent typically contains a hash pointer which serves as a link to the previous block as well as a timestamp and transactional data. The combination of the hash

pointer, timestamp, and transactional data is what makes blockchains so impervious to being modified (Raval, 2016). The primary reason that the combination of the hash pointer, timestamp, and transactional data provides a blockchain with such strength in security is that a block of data cannot be modified retroactively without altering the blocks of data that follow it in addition to having complicit support from others in the network (Armstrong, 2016). Because transactions are not only audited but verified by participants of the blockchain there is a decreased chance that an invalid or unsecured transaction will be able to be completed successfully (Catalini & Gans, 2017). Lastly the number of lost devices will decrease because Proof of Location allows users as well as self-governing agents the ability to be able to authenticate their location data when they choose too as well as reveling their personal information when they choose by providing a location claim that is fraud-proof (King, 2018a).

### 1.5 Assumptions

The assumptions of this study are:

- Updates will be released to Tendermint Blockchain and Ethereum which might prevent them working at anytime.
- Ethereum Blockchain will be running on one Raspberry Pi and Tendermint Blockchain will be running on the other.
- The physical devices that will be used for testing purposes will also have a cyber presence.
- There will continue to be security vulnerabilities that will be discovered and utilized in CPS.
- There are a number of differences in the internet security between CPS devices that are being used by home users and the US Federal Government.

### 1.6 Limitations

The limitations of this study are:

- There will be two Raspberry Pi's that will be used for testing which are Raspberry Pi 3B and Raspberry Pi 3B+.
- The devices that will be used for testing will be CPS home devices.
- The CPS devices that will be used for testing will be used from beginning to end.

### 1.7 Delimitations

The delimitations of this study are:

- CPS devices that are currently being used by the US Federal Government will not be used.

### 1.8 Summary

This chapter provided the scope, significance, research question, assumptions, limitations, delimitations, definitions, and other background information for the research that has been completed on CPS, security, and blockchain. The next chapter will provide a review of relevant literature that is related to the problems of CPS, network security, and blockchain.

## CHAPTER 2. REVIEW OF LITERATURE

This chapter provides a review of the literature relevant to the problems of CPS, security, and blockchain.

The term CPS was originally used by Helen Gill in 2006 when she was at the “at the National Science Foundation in the United States” while the term cyberspace was originally used in the novel *Neuromancer* by William Gibson in 2000 (Lee & Cheng, 2015, p. 4838). Even though the term CPS is a relatively recently developed term, the technology has been available since World War II when Norbert Wiener developed the term cybernetics. Mr. Wiener was an American mathematician who initially developed the technology that would lay the foundation for the automated processes that were required in the “automatic aiming and firing of the anti-aircraft guns” (Lee & Cheng, 2015, p. 4838). The automated processes that Mr. Wiener developed in World War II are closely associated with the theory of control systems that are used in the feedback systems of present day computerized control systems. The control logic that he used was essentially a computation, notwithstanding the fact that it was executed using mechanical parts and analog circuits. It is for this reason that cybernetics is the combination of computation, communication and processes that are physical. The term cybernetics is a derivation of the “Greek term κυβερνήτης which translates to kybernetes and means helmsman, governor, pilot or rudder” and it is thought to be an appropriate metaphor for control systems (Lee & Cheng, 2015, p. 4838).

With the technical knowledge for CPS dating back well over seventy years, the question then becomes why are we experiencing the security issues that we are today? Some of the security issues that have been experienced include the Maroochy wastewater breach, several power outages that have occurred in Brazil, the SQL Slammer attack on the Davis-Besse Nuclear plant, the StuxNet computer worm as well as a variety of industrial security incidents (Pasqualetti, Dorfler, & Bullo, 2013).



Blockchain has been used by Bitcoin in order to process financial transactions that have caused it to receive some bad press recently. What gives entrepreneurs some grief is that Bitcoin has been closely associated with the dark web and websites like the Silk Road. This means that blockchain has been giving those that are involved with the processes of buying and selling of illegal materials on the dark web a bad feeling. This involvement in illegal activities has led to some questions as well as debates about what are some additional opportunities that are available for not only blockchain but its protocols that support Bitcoin and other cryptocurrencies (Deloitte LLP, 2016). Blockchain was initially defined by Vitalik Buterin as:

... a magic computer that anyone can upload programs to and leave the programs to self-execute, where the current and all previous states of every program are always publicly visible, and which carries a very strong cryptoeconomically secured guarantee that programs running on the chain will continue to execute in exactly the way that the blockchain protocol specifies (Pilkington, 2015, p. 8).

Blockchain completed its most recent fork or split on August 1, 2017 because there were some members of the bitcoin community who were not happy with the direction that bitcoin was taking. This split created a new “digital token called Bitcoin Cash” while allowing the existing blockchain to remain intact (Graham, 2017, para. 1). Blockchain has been used in the past as a method of tracking a variety of transaction histories including financial, shipping, and ownership changes (*Blockchain*, 2017). Based upon the above information further research is needed in order to determine how blockchain could be used to improve the security of CPS.

## 2.1 Literature Review Background

There are a variety of definitions about what CPS are but three things are consistent. CPS support the use of integrated computational systems, have physical capabilities that allow them to interact with humans in a variety of ways, and there are a variety of opportunities and research challenges that are available for them (Baheti & Gill, 2011). Some examples of CPS systems that

we frequently encounter are networks for transportation, gas and water distribution as well as advanced communication systems (Pasqualetti et al., 2013). The integrated computational systems are also known as embedded systems. The integrated computational systems are used to analyze the data that is physically collected by a variety of systems. Once the data is analyzed then a corresponding action or response is determined by the physical portion of the system. The computational system contains the software and the algorithms that are used to complete the analysis as well as a processor and corresponding microcontroller. The “region of interest (ROIn)” as well as the “region of impact (ROIm)” are two physical aspects that are contained within the embedded system. Region of impact (ROIm) is a region that is computationally modified to support the results of the analysis (Guo, Hu, & Hong, 2013). Region of interest (ROIn) is a region or rectangle that is defined by the user with the goal of limiting the calculations that will be completed to within certain parameters (Brinkmann, 1999). CPS devices have been able to solve a variety of issues but they are still susceptible to their surrounding conditions which have caused several of their vital systems including safety, security, and sustainability to experience failure (S. Guy, Boyle, & Hu, 2013).

The physical infrastructure of CPS devices has long been shown to be extremely prone to failure as well as attacks and their data management and communications layers are prone to cyber-attacks. There have been a variety of attacks on CPS devices and networks but some of the more commonly known attacks include the Maroochy wastewater breach, several power outages that occurred in Brazil, the SQL Slammer attack on the Davis-Besse Nuclear plant, the StuxNet computer worm as well as a variety of industrial security incidents (Pasqualetti et al., 2013).

In March of 2000 the Maroochy Shire Council noticed that they were receiving an increase in the number of reported issues with the newly installed wastewater system. The communications that were being sent to the wastewater pumping stations via radio links were getting lost, pumps were experiencing issues working properly, and the alarms that were installed with the goal of alerting staff to failures were not working properly. The issues were originally attributed to problems with the new system but while monitoring it an engineer observed someone

was repeatedly breaking into the system and intentionally causing the issues. The perpetrator turned out to be a former contractor Vitek Boden who was eventually arrested and put in jail (Slay & Miller, 2007).

The Stuxnet computer worm attacked the Iranian nuclear facility Natanz in June of 2010. The worm was able to infect over 60,000 computers of which more than half were located in Iran. Other countries that felt the sting of the Stuxnet Worm included India, Indonesia, China, Azerbaijan, South Korea, Malaysia, the United States, the United Kingdom, Australia, Finland, and Germany. Even though there are numerous known antidotes as well as having a built-in expiration date of June 24th, 2012, the Stuxnet worm continues to be discovered. Stuxnet primarily targeted devices that “were air-gapped” or not directly connected to the public internet, which meant that in order for them to be infected the worm had to be introduced using an intermediary device such as a universal serial bus (USB) device. The worm used four “zero-day vulnerabilities” which allowed it to employ “Siemens’ default pass-words to access Windows operating systems that run the WinCC and PCS 7 programs” such as programmable logic controllers (PLC) which executed a variety of programs including those that are used to manage industrial environments (Farwell & Rohozinski, 2011, p. 3).

### 2.1.1 CPS Device Safety

Safety is a fundamental requirement for the systems that are contained within CPS devices. Previously the goal of ensuring the safety of CPS devices was done by effective software maintenance but this does not allow for all safety measures to be included. The idealized goal of optimizing the safety of CPS devices should be to focus on the interactions that occur among the computing devices and their corresponding physical systems. There are a variety of safety issues that have been recently addressed in connection with CPS devices and should be taken into consideration when designing and testing new CPS devices with the goal of preventing the issues from happening. Some of the more commonly discussed safety issues that are related to CPS

devices is the possibility of a computing device negatively affecting another devices ability to monitor devices, the unknown impacts that a single device could have on its surrounding environment, and the physical or environmental effects that could result due to the cyber devices monitoring of the physical conditions (S. Guy et al., 2013).

### 2.1.2 CPS Device Energy Sustainability

The next problem that has been determined to be of substantial concern is the energy sustainability of CPS devices. Energy sustainability according to S. Guy et al. (2013), and Banerjee, Venkatasubramanian, Mukherjee, and Gupta (2012) is defined as "the balance between the power required for computation and the power available from renewable or green sources"(Banerjee et al., 2012, p. 287; S. Guy et al., 2013, p. 112). There have been several calls to move from using batteries and generators to more environmentally friendly sources of energy such as solar, or wind power. There has been one significant challenge that has been frequently encountered when attempting to implement more environmentally friendly energy sources and that is how can they be effectively implemented in the system design (S. Guy et al., 2013).

### 2.1.3 CPS Security

The third and final condition that researchers are expressing concern about in relation to CPS devices is the security of the devices. S. Guy et al. (2013) as well as Banerjee et al. (2012) have both defined security as being "the ability to ensure that both data and the operational capabilities of the system can only be accessed when authorized" (Banerjee et al., 2012, p. 286; S. Guy et al., 2013, p. 112). There are a variety of threats that can be posed to CPS devices due to security vulnerabilities such as compromised privacy as well as the damaging effects that can be posed to the physical environment. It is due to the vulnerabilities that are posed that the security of CPS devices should be a primary concern of researchers (S. Guy et al., 2013).

There are several unknown as well as known differences between the technology security of those at the corporate level and the security of the technology within CPS devices. Some of the more commonly known differences between corporate level information technology (IT) security and the security of CPS devices or their control systems are the frequency of updates and patches, real-time availability requirements, how well legacy systems are able to support cryptography, and simpler network requirements (Cárdenas et al., 2009).

The first property that differentiates the security of CPS devices with those in the standard technology sector is that it is not advised to have frequent software patches and updates process because they are not frequently supported by control systems. The next property that differentiates control systems from corporate IT is the requirement to have a continuous availability so that decisions can be made in real time. Another property that will separate large corporate IT systems from control systems is the fact that there is a substantial amount of control systems that are being used in industrial environments that are running legacy systems and they will not support standard cryptography methods. There is research that is being completed to improve the lightweight cryptography options that are available. The last property that distinguishes control systems from corporate IT is the fact that they have simpler requirements for networking such as rarely having servers change, standardized layout requirements, a user population that has remained stable, consistent communication patterns, and the number of available protocols is restricted. Since control systems have been shown to have simpler networking requirements, it would mean that it is possible that the implementation of a network intrusion detection (NID) system would be easier than on standard corporate IT devices (Cárdenas et al., 2009).

There have been a variety of concerns expressed about the security issues of control systems such as detecting system faults, isolation, and recovery which can be attested too but CPS also suffers from security vulnerabilities that are specific to them and do not affect standard control systems. It is these vulnerabilities that are increasing the demand for the development of detection and identification techniques that would be useful in decreasing the successes of these threats (Pasqualetti et al., 2013). One of the more common types of attacks that can be directed at

control systems are resonance attacks in which an attacker will compromise the sensors and/or controllers thereby forcing “the physical system to oscillate at its resonant frequency” (Cárdenas et al., 2009, p. 2). The demand for improved detection and identification of vulnerabilities is substantiated when you look at how much of a reliance there is on unsecured communication networks as well as the standard protocols that are used in communication to send the measurements and control packets. This method of transmission increases the chances of intentional and worst-case scenario attacks against physical plants. In contrast, the standardized information security methods of access control, authentication, and message integrity appear to be insufficient to provide a suitable amount of protection to CPS. Due to the fact that these security methods do not capitalize upon the weaknesses of the physical processes or the control mechanisms; they have been shown to be ineffective at preventing insider attacks that target the physical dynamics of systems (Pasqualetti et al., 2013).

The security of CPS systems is a highly debated topic because there are a variety of opinions which state that the security should be handled from the cyber side, or the physical side; but the most common opinion is the security should be incorporated in the both cyber and the physical side. The implementation of security on the cyber side as well as the physical holds so much appeal because it allows both sides to work together with the goal of preventing, detecting, and handling attacks that originate from an outside source. Despite the popularity of the idea integrating security into the cyber and physical there is a broader area of security that contains a larger base of opinions. There are several security methods that can be used with a monitoring system that creates a connection between the cyber and physical systems which will allow the security to be integrated into the system rather than being applied as a patch after the systems has been completely setup. CPS affects several industries including smart infrastructure (SI), body area networking (BAN), and the efficiency of the energy of data centers. Body area networks (BAN) are a group of applications which are primarily used by those in the medical field and are involved with systems that are used to monitor the health of others. Smart infrastructure (SI) are systems that are primarily used in connection with the infrastructure of smart electrical grid

systems. The processes that are involved in the monitoring of the idle and active modes in data centers with the goal of ensuring energy efficiency by making sure that unnecessary functions are disabled unless they are needed whereby the consumption of power will be reduced (S. Guy et al., 2013).

There has been an increase in the amount of analyses that are being completed on the external vulnerabilities of CPS due to the increasing amount of attacks that they have been experiencing recently (Pasqualetti et al., 2013). The standard approach has been to study the how a system is affected by an attack on it. There have been numerous attacks, issues, and problems that have been detected but some of the more common ones are deception and denial of service attacks, false data injections, stealthy deception attacks, replay attacks, covert attacks as well as several security issues that have been specifically directed at CPS devices and networks (Pasqualetti et al., 2013). “Security solutions can exploit the physical nature of CPS by leveraging location-based, time-based and tag-based mechanisms” (Rajkumar, Lee, Sha, & Stankovic, 2010, p. 734).

#### 2.1.4 Internet of Things

There has been a recent push towards creating a decentralized architecture in order to expand the network of the Internet of Things (IoT) devices and ensure its success. The current centralized model is causing the manufactures of the devices to have a high maintenance cost such as the costs that are incurred when a company must roll out software updates for devices that are out of date. Consumers are expressing several negative opinions about the trust that they are willing to place in devices that can initiate contact with other devices on their own as well as having a device that will support the use of transparent security. The above issues can be resolved with the use of a scalable model that also supports the use of a peer-to-peer model that is also trustless and is able to function in a transparent manner and securely distribute data (Christidis & Devetsikiotis, 2016).

Additionally, IoT has faced several challenges in internet security including the continuous demand for servers in order to be able to determine what can be done to prevent distributed denial-of-service (DDOS) attacks that have originated from IoT devices. One of the more recently known attacks was the Mirai virus that attacked unsecured devices using HTTP, an SSH server, and tent with the goal of not only gaining remote control of the devices but being able to load malware into the devices memory. In 2016 Rotem Kirner was able to successfully take control of an Internet Protocol (IP) Closed-circuit television (CCTV) as well as a variety of Digital Video Recorder (DVR) controllers (McKee, Clement, Almutairi, & Xu, 2017).

#### 2.1.5 Differentiating IOT from CPS

Recently there has been a great deal of debate about the differences as well as the similarities between CPS and IoT. According to IBM they have defined the Internet of Things (IoT) to be a “dynamic and distributed environment that is composed of numerous smart devices that sense their environment and that are able to act in that environment” and it is due to the above-mentioned “devices, it is possible to monitor the external environment, gather information about the real world, and create a type of ubiquitous computing, which enables every device to communicate with any other device in the world, from everywhere” (Zanni, 2015, p. 2). The goal of IoT is to allow the internet to become more universal by allowing devices to be able to connect with each other as well as work together as individualized sensors or swarms of sensors that will create not only large endpoints but act as an entire system (Zanni, 2015). CPS uses several elements of computing with the goal of getting them to coordinate the use of sensors as well as communicate with them in order to monitor both cyber and physical systems as well as actuators. Actuators are able to modify the environment that surrounds both the cyber and physical systems where they are running. Not only do CPS devices attempt to control their surrounding environment, but they can also use sensors to create connections in it so they may in turn use the distributed intelligence that is contained within the sensors to obtain a more thorough knowledge



of its environment. This will allow it to move in a more precise manner (Zanni, 2015). Some of the more recent advancements in IoT have been due to the combination of the computational and physical components of CPSs that have been integrated with the computational resources. Lastly “CPSs often support real life processes and provide operational control of Internet of Things objects, which allow physical devices to sense the environment and modify it” (Zanni, 2015, p. 3). Some of the more commonly known CPS device types are “medical devices and systems, aerospace systems, transportation vehicles and intelligent highways, defense systems, robotic systems, process control, factory automation, building and environmental control and smart spaces” (Rajkumar et al., 2010, p. 731).

#### 2.1.6 Blockchain

Blockchain is a digital ledger that contains the transactional history of the various operations that it has processed (*Blockchain*, 2017). In addition to being a digital ledger blockchain is able to provide the ability to be confirm the integrity of a particular piece of information or data as well as validate that it has not been modified (Ivezic, 2016). A blockchain can be grouped into one of three different types of blockchains which are public, private, and hybrid or consortium (Deloitte LLP, 2016).

Bitcoin was the first digital alternative to cash which was developed in 2008 by Satoshi Nakamoto (Deloitte LLP, 2016). It is a peer-to-peer decentralized online currency that is able to maintain its value without any financial sponsorship, inherent worth, or central issuer (Buterin & Vitalik, 2014). Despite its checkered past Bitcoin has proven that cryptocurrencies are a viable alternative to cash as well as other monetary exchanges in present society (Deloitte LLP, 2016).

Bitcoin was developed by Satoshi Nakamoto in 2008 as the first decentralized digital currency that could also be used virtually. It was fully operational in 2009. It is a “crypto Peer to Peer currency” which means that it is an “electronic payment system” that is “based upon cryptographic proof instead of trust” (Flavio, 2013, p. 3). Despite no financial institutions

managing Bitcoin it is still experiencing growth. The four key concepts of Bitcoin are transactions, proof of work, digital wallet, and mining (Nakamoto, 2008). Bitcoin transactions are public but anonymous transactions that occur between the owner and the recipient and they are broadcasted throughout the P2P network. The process of mining nodes is when the transactions are collected in blocks and each block contains a proof of work (POW). A new block is started and linked to the existing blockchain when there is either a special transaction, or a new first transaction in the block of the blockchain. After the new block is created then the network receives a broadcast with status of the new block that has been added to the blockchain (Nakamoto, 2008).

#### 2.1.7 Hashes and their functions

The primary aspect of blockchain is that it will be used for informational purposes over economic or monetary benefits which is part of the reason that it is conducive to the newer token-free blockchains that are gaining popularity. Blockchain has a heavy reliance on hash functions and their corresponding hashes. A hash is the output which results when the original information or input is transformed. The hash function is the mathematical computation or algorithm that takes the original input values and calculates its output value. The primary characteristic of a cryptographic hash function is that it is extremely hard to reverse or recreate the input value from the hash values and it is known as collision resistance (Pilkington, 2015).

Proof of work is a protocol that has the primary goal of preventing cyber-attacks such as a distributed denial-of-service attack (DDoS) (BlockGeeks, 2017).

Proof of stake is another way that is used to validate transactions as well as accomplish the goal of a distributed consensus. Proof of stake rewards the creator of the block using a deterministic method that is based upon the wealth or the stake of the block (BlockGeeks, 2017).

Even though there is not a centralized entity the process of mining bitcoins is completed when the proof of work from a transaction block has been solved which allows for the confirmation of transactions and it also increases the security of the bitcoin (Nakamoto, 2008).

When a transaction is completed using the blockchain, it is added on to the end of it. Since blockchain is required to keep a complete list of all the transactions that have been completed with it; the length of the blockchain is rapidly growing because each transaction that is processed becomes a block that is added on to the end of the blockchain (*Blockchain*, 2017). As the block is added onto the end of the blockchain it is also added to be included in the ledger and then the ledger is able to be programmed so that it can automatically generate transactions (Iansiti & Lakhani, 2017). Blockchain also provides another service which is that it permits people who do not know or trust each other to share a record of events (Deloitte LLP, 2016). The information contained within the record is shared with all the users on the network who can validate the transactions which will reduce the need for a middleman (Deloitte LLP, 2016). In order to validate the transactions, each node which is a computer that is connected to the Bitcoin network verifies and downloads the transactions as well as relaying them to the next node (*Blockchain*, 2017). Each block in the blockchain contains a timestamp of when the transaction occurred as well as a link to the previous block (*Blockchain*, 2017). The information that they are validating is contained with a shared record or a ledger (Deloitte LLP, 2016). Every node that is located in the Bitcoin network or similar network automatically receives a downloaded copy of the blockchain when they join (*Blockchain*, 2017). By storing the contracts in a shared database that is also transparent they are protecting the information that is within the blockchain from being deleted, tampered with or modified (Iansiti & Lakhani, 2017). The addresses as well as the balances for each location in the blockchain start with the genesis block and contain the information of the most current block (*Blockchain*, 2017). The genesis block is the first or originating block in the blockchain (*Blockchain*, 2017). If there is a modification or change made to a block in the blockchain then its change affects all the blocks which are located after the modified block. In the blockchain world every transaction, payment processed, signed agreement

and task would have a connected digital record as well as verified signature connected to it.

Though it can also be thought of as a form of nonrepudiation blockchain decreases need for those individuals employed in the financial, investment, or law sectors to function as intermediaries because people, machinery, organizations, and algorithms would be able to independently complete transactions between each other with a minimal amount of resistance which shows the possibilities that could be available with blockchain (Iansiti & Lakhani, 2017). It is due to the increasing possibilities that are available for using blockchain that it has experienced such a rapid increase in popularity as well as increased usage in horizontal market. There are several companies that have been active in the various markets but some of the more popular ones are Chain of Things, Ethereum, Coinbase, Itbit, Petamine, and EasyWallet.org (Deloitte LLP, 2016). As the popularity of blockchain continues to spread there have been several security issues that have been encountered such as the collapse of the bitcoin exchange in 2014 as well as the hacking that has been done on a few others. As blockchain evolves so do the barriers that it will encounter such as those in technology, government, organizational, as well as society which it has already encountered. It is due to the previously mentioned issues that opportunities to implement security using blockchain is experiencing so much interest especially with Internet of Things (IoT) devices (Iansiti & Lakhani, 2017). Blockchain was originally started in October of 2008 as a peer to peer network that would sit on top of the internet and would serve as a portion of proposal bitcoin. Bitcoin is a virtualized currency system that prevented the use of sanctions in order to issue currency, complete transfers of ownership, and to confirm of the processed transactions. Bitcoin was the first of many applications that made use of the technology that is available to blockchain (Iansiti & Lakhani, 2017). Blockchain can be used for both cryptographic and non-cryptographic functions that occur in the real world. The issue of security arises when you take into consideration that the ledger is shared across the network which means that everyone who receives the shared ledger is able to see every transaction in the ledger. This shared ledger can be used to aid in completing several tasks such as processing payment and exchange which are in the

financial sector or the on-chain tokens can be used to represent one of several financial assets such as stocks, bonds, cash, etc.

### 2.1.8 Public Blockchain

Public blockchains are blockchains that anyone can read as well as allow transactions to be sent to (Buterin, 2015). The public nature of public blockchains originates in the restriction free participation that it permits in order to determine what blocks are appended to the chain as well as determining the current state of the chain (Pilkington, 2015). After the transactions have been validated a receipt of a blockchain transaction should expect to see them included in the chain. In addition to being able to read the blockchain anyone in the world is able to participate in the consensus process. The consensus process is a method that is used to determine what blocks will be added to the chain as well as determining its current status. Since public blockchains are frequently used to substitute for trusts that are centralized in nature they are secured with cryptoeconomics which is a combination of cryptographic verification and economic incentives which will use either proof of stake or proof of work to validate the information. Cryptoeconomics allows blockchain to follow the universal principle that states the degree to which a person is able to influence the consensus process is related to the quality of the economic resources that they are able to provide. Public blockchains are usually thought to be “fully decentralized” (Buterin, 2015, para. 3).

#### 2.1.8.1 Etheruem Blockchain

Etheruem Blockchain is a public blockchain network that has been optimized for use with smart contracts and it has its own cryptocurrency which is called Ether (Christidis & Devetsikiotis, 2016).

#### 2.1.8.2 Advantages

Public blockchains have two major advantages over their private counterparts. The first advantage is that public blockchains protect the application's users from the developer provided that there are certain tasks that the developers of the application are not able to complete. By giving up the ability to be able to modify the application the developers increased the trust that the users had with them as well as reducing the chances that they would be coerced or pressured by another entity into changing or modifying the application in some way. The second benefit of public blockchains is they are open which means that they are likely to be used by multiple entities and will be able to acquire network effects (Buterin, 2015).

#### 2.1.8.3 Disadvantages

There are a few disadvantages that are related to public chain in that they have longer confirmation times, and transactions can cost up to three cents each to process (Buterin, 2015).

### 2.1.9 Private Blockchain

Private blockchains have the most restrictions on them because the ability to be able to write to the blockchain is kept in a location that is central to one organization (Buterin, 2015). Private blockchains are closely analogized with permissioned ledgers which allows for organizational processes which will allow user ids to be whitelisted or blacklisted (Pilkington, 2015). The public may or may not have the ability to be able to read the information that is contained with a private blockchain. A few of the ways that private blockchains can be used are for database management, or completing an internal audit of a company which would mean that the public would not need to have the ability to be able to access the information but there cases where having public accountability would be a good thing (Buterin, 2015).

#### 2.1.9.1 Advantages

Private blockchains has five advantages over public blockchains. The first advantage that public blockchain has is that the consortium or company which is running the blockchain is able to easily modify the rules of the blockchain, reverse transactions, modify balances as well as a few other financial transactions. The next advantage that users of a private blockchain will gain is the knowledge that validators are known which will reduce the chances of a 51% attack being initiated by another miner. Another benefit that users of private blockchains will experience is that the processing of transactions is cheaper because the transactions only need to be verified by a few nodes which have more power in their higher-powered processors than several thousand laptops would. The reduction in power consumption for private blockchains is a major concern currently because public blockchains have transaction fees that are over a penny a transaction but the long-term forecast is that there will be a reduction in the transaction costs as there are improvements made to the scalable blockchain costs which has promised to significantly reduce the costs of public blockchains so they are closer to private blockchains. The next advantage that users of private blockchains will experience is that nodes have proven to be very well connected and if a system failure is experienced it can be easily repaired with manual intervention as well as allowing the use of consensus algorithms which reduces the block times. The fifth advantage that users of private blockchains will experience is that if the read permissions are restricted then there is a greater level of privacy (Buterin, 2015).

#### 2.1.9.2 Disadvantages

There are very few disadvantages to private blockchains but the ones that I could find are they have fewer participants due to their smaller size as well as having an increased chance of being taken over. The reason that they have a smaller size is that private blockchains are not as spread out as public blockchains. Additionally since private blockchains are smaller it also means that their smaller size makes them more likely to be a victim of an attack because they do not have a lot of people on the network who will be able to validate transactions thus increasing the chances of a 51% attack (Christidis & Devetsikiotis, 2016).

### 2.1.10 Consortium or Hybrid Blockchains

Consortium blockchains are also known as hybrid blockchains which have a group of pre-selected nodes that are used in the consensus process (Buterin, 2015). Consortium blockchains are also thought of as being hybrid blockchains because they are partially decentralized and do not have the stringent separation requirements that have been established for public and private blockchains (Pilkington, 2015). Consortium blockchains must have one of three types of rights to read it which are public, limited to the participants, or hybrid routes. One example of a hybrid route is where the root hashes from the blocks and API are publicly shared together in order to allow public members of the consortium blockchain to be able to make a certain number of queries and get the cryptographic proofs of certain parts of the blockchain status. Consortium blockchains could be thought of as being “partially decentralized” (Buterin, 2015, para. 4). Recently there has been some confusion about the difference between private and consortium blockchains but consortium blockchains are somewhat private (Thompson, 2016). The best definition for consortium blockchains has been provided by Vitalik Buterin and it is:

So far there has been little emphasis on the distinction between consortium blockchains and fully private blockchains, although it is important: the former provides a hybrid between the “low-trust” provided by public blockchains and the “single highly-trusted entity” model of private blockchains, whereas the latter can be more accurately described as a traditional centralized system with a degree of cryptographic auditability attached (Buterin, 2015, para. 3).

#### 2.1.10.1 Advantages

Consortium or Hybrid Blockchains do not allow just any person who has a connection to the internet to verify the transactions that have been processed nor do they allow complete control to be given to one company but they use a selected grouping of predetermined nodes. Additionally, consortium blockchains also allow for several benefits that are associated with



private blockchains such as the ability to process transactions privately and efficiently without needing to consolidate power within one company. In fact, the predetermined nodes that validate transactions can be thought of as being akin to a council of elders; where the council members are usually recognized as individual entities who decide who is allowed to have the ability to read the blockchain ledger. It has been recommended that the best use for a consortium or hybrid blockchain would be for organizational collaboration which makes the available uses of blockchain to be close to limitless (Buterin, 2015).

#### 2.1.10.2 Disadvantages

There are few disadvantages to consortium blockchains. The first issue is that there is not a consistency on who can read a consortium blockchain as well as how it can be accessed. Some consortium blockchains might be shared publicly and others that are restricted to the participants of that blockchain (Buterin, 2015). Additionally, there are some consortium blockchains that support:

Hybrid routes such as the root hashes of the blocks being public together with an API that allows members of the public to make a limited number of queries and get back cryptographic proofs of some parts of the blockchain state (Buterin, 2015, p. 4).

#### 2.1.11 Smart Contracts

Smart contracts are scripts that have the ability to be self-executing and they are located on the blockchain. They also support workflows which are distributed as well as automatic workflows. Nick Szabo defined smart contracts as meaning “a computerized transaction protocol that executes the terms of a contract” (Christidis & Devetsikiotis, 2016, p. 2296).

### 2.1.12 Blockchain Network

A blockchain network is comprised of a set of nodes which are also known as clients and function on the same blockchain through the copy that each one contains. A node is frequently used as a point of entry into the network for users of that particular blockchain and these nodes will frequently form a peer to peer network. Since a peer-to-peer network is formed it permits the users the ability to complete four tasks. The first task is that they will be able to interact with the blockchain using a set of public and private keys which they will use to sign each of their transactions and the public key will be used in order to ensure that they are able to be found on the network. Each time a user signs a transaction it will be shared from the user's node to its peers that are within one-hop of its current location. This use of asymmetric cryptography provides the network with authentication, integrity as well as non-repudiation that is needed to validate transactions. The next task that can be completed is the peers that are connected to the node in question are tasked with validating the transaction before allowing it to proceed farther on the network. If the transaction is determined to be fraudulent then it will be discarded. Using neighboring nodes, the transaction will be shared across the network. The third task that will be completed after the transactions have been grouped and determined to be valid by the network is they will be ordered and packaged into a candidate block that has been timestamped. This process is known as mining which will share the block with the rest of the network. The final task that is completed is the node will validate that the block contains a valid transaction as well as a reference to the hash of the appropriate preceding block that is located on the chain. If the transaction is determined to be valid and well, containing a connection to the previous block, then it is added to the chain. Then transactions are applied and the world view is updated to reflect the changes. If the transaction is determined to be invalid, then the suggested block is discarded. Once the transaction is added to the chain; it indicates that a round has been added to the blockchain (Christidis & Devetsikiotis, 2016).

Bitcoin transactional model is a transaction configured to transfer the bitcoin values that are broadcast to the network so they can be collected into blocks. The transaction frequently references the output values of the previously completed transactions as input values for new transactions and all the bitcoin values that were used as inputs will also be used as dedicated outputs. Since bitcoin transactions are unencrypted; then it is probable that every transaction once it is collected into a block is browsable as well as viewable. Since transactions that have a standardized output can nominate addresses this means that in order to exchange any additional inputs and outputs there must be an applicable signature on file. Due to all the transactions being visible in the blockchain this means that they are also viewable with hex editor. A blockchain browser is a location where every transaction that is contained in the blockchain is able to be viewed by anyone. This is a valuable feature to have in order to be able to see the technical specifications of the transactions as well as being able to verify payments have processed (*Bitcoin Wiki:Transaction*, 2016). Unspent Transaction Output (UTXO) is the output transaction that has not been spent as an input into a new transaction (*Unspent Transaction Output, UTXO - Bitcoin Glossary*, 2017).

#### 2.1.13 Byzantine fault tolerance

Byzantine fault tolerance (BFT) is “the ability to be able to tolerate machines failing in arbitrary ways, including becoming malicious” (“What is Tendermint ?”, 2018, para. 2). The theory of BFT is decades old, but software implementations have only become popular recently, due largely to the success of “blockchain technology” like Bitcoin and Ethereum.

#### 2.1.14 Tendermint

Tendermint is a piece of software that uses BFT to securely and consistently replicate an application to several machines provided that one third of the machines do not experience a random failure. The ability to be able to consistently replicate an application is important because

there is a need to be able to see the same transaction logs as the other machines on the network because this will permit the same state to be consistently computed. The ability to be able to consistently and securely replicate machines that are located in distributed systems has been a cumbersome task which is why it poses such a significant role in the ensuring fault tolerance in a variety of applications, currency exchanges, elections, infrastructure organization as well as many more things (“What is Tendermint ?”, 2018). Tendermint contains two technical components which are “a blockchain consensus engine and a generic application interface” (“What is Tendermint ?”, 2018, para. 2). The consensus engine which is known as Tendermint Core is used to guarantee “that the same transactions are recorded” in the same order on every machine (“What is Tendermint ?”, 2018, para. 3). The Application BlockChain Interface (ABCI) is an application interface and it permits the processing of transactions in any programming language (“What is Tendermint ?”, 2018). In contrast with other blockchains as well as consensus options Tendermint does not contain pre-packaged state machines which means that developers are able to utilize the BFT state machine with Tendermint to replicate a variety of applications “written in whatever programming language and development environment is right for them” (“What is Tendermint ?”, 2018, para. 3). Tendermint was “designed to be easy-to-use, simple-to-understand, highly performant, and useful for a wide variety of distributed applications” (“What is Tendermint ?”, 2018, para.4).

#### 2.1.15 FOAM Protocol

FOAM presents a protocol that utilizes geospatial technology in combination with the Ethereum blockchain that will be used by smart contracts as well as dApps as the to provide location services. The FOAM Protocol is an “interoperable standard for location on Ethereum” and it “is an open protocol for decentralized, geospatial data markets” (*Introducing the FOAM Protocol*, 2017, para. 1). Not only is FOAM backwards compatible with OpenStreetMap (OSM) it is also connected to their application programming interface (API) and a “monetization layer to

open source mapping to enable new geospatial markets of exchange” was added (*Introducing the FOAM Protocol*, 2017, para. 6). The addition of the “monetization layer to open source mapping” will allow developers to be able to create applications that will be connected to a physical address that has been verified as well as allowing the coordinates to be able to be “turned into a blockchain wallet that can hold a balance and be tagged with crowdsourced data” (*Introducing the FOAM Protocol*, 2017, para. 6). Maps have evolved over the years from the classic hand drawn map to those that use a variety of digital technology tools such as GPS devices, or smartphones that use a variety of applications to track your location. The process that is used to navigate the world is primarily controlled by the map makers which is starting to cause an issue because there has been an increase in the number of applications that require geospatial data that is not only verified but also consensus-driven. It is due to the increased demand for consensus driven location data that blockchain technology is beginning to replace networks with markets which will ensure that the newly designed applications will be able to access this data which is not only verified but trusted. There have been several attempts previously made to develop maps that can be read by humans as well as machines and have been verified as being accurate which has dealt a crippling blow to open source projects that are not receiving as much funding as others (*Introducing the FOAM Protocol*, 2017). In the end, “while Bitcoin creates digitally scarce value, and Ethereum creates programmable money, FOAM will enable programmable space” (*Introducing the FOAM Protocol*, 2017, para. 6).

#### 2.1.16 Proof of Location

Proof of Location affords users as well as self-governing agents the ability to be able to authenticate their location data when they choose too as well as reveling their personal information when they choose by providing a location claim that is fraud-proof (King, 2018a). There are five primary features of FOAM’s Proof of Location and they are it is trustless, independent, open, accountable, and incentivized. The first feature of FOAM’s Proof of Location

is that it is trustless which means that it supports synchronization with a Byzantine fault tolerant clock. The next feature of Proof of Location is that it is independent which means that it is not dependent on GPS. The ability for anyone to be able to utilize the network as well as be able to offer utility services is another feature of the Proof of Location from FOAM. The ability to be held accountable is the next feature that is part of Proof of Location and it means that the economics are structured in a way so as to ensure behavior that is honest and is able to be verified with a proof that verifies that it is not fraudulent. The final feature that is part of Proof of Location from FOAM is the fact that it is incentivized which means that service providers are repaid for extending the parameters of the localization and verification zones (King, 2018a).

#### 2.1.17 Crypto-Spatial Coordinates

Crypto-Spatial Coordinates (CSC) are an open standard for location based services that supports the use of a variety of hardware and software. The world that we interact with is beginning to experience a variety of changes and blockchains are going to make those changes more noticeable especially in relation to internet of things (IoT), supply chain management, gaming that makes use of your current location as well as augmented reality games in addition to how land registration and real estate deals are completed (Josefsson, 2017). “Currently, there are no standards for embedded locations, physical addresses, or coordinates in smart contracts” (Josefsson, 2017, para. 3). Since there are a variety of different blockchains that use smart contracts they will need a consistent language that will allow them to be able to exchange information with each other as well as being able to reference and index a variety of things in the physical world. The CSC that FOAM is using will be the foundation for the sharing of the FOAM new location standard as well as permitting any smart contract to be able to make a permanent claim to a specific physical location and connect it to an address on the blockchain (Josefsson, 2017).

## 2.2 Completed Research

There has been an extensive amount of work and research that has been completed in the various IoT deployment options. Some of the more common ones are logistics, sharing of property, marketplace options, peer-to-peer marketing where machines would be able to buy and sell and energy, peer-to-peer exchanges for tracking, peer-to-peer exchanges using atomic tokens to complete the process, and automating the logistics movements (Christidis & Devetsikiotis, 2016).

There are four primary types of transactions that blockchain has been able to record. They are financial, public, reputation, and security (Zheng, Xie, Dai, & Wang, 2016). Some of the financial transactions that blockchain has been able process include sending and receiving money using one of several different cryptocurrencies as a method of payment (Zheng et al., 2016). Blockchain was originally designed to allow currency transactions to be completed in a trustless environment but it has evolved to so much more (Zheng et al., 2016).

Additionally, Blockchain has been used by those in the public sector in order to manage the information that is related to a piece of land or property including its physical status as well as management rights. Blockchain also has the ability to be able to record and manage any changes that are made to a piece of property which will improve the efficiency of the services that the public sector will be able to provide. It will also decrease the amount of time it takes to complete the registration process for buying and selling land. Blockchain has the ability to be used for a variety of additional applications in the public sector such as recording marriage licenses, completing patent management, and maintaining income tax records. As the number of public services that support the integration of blockchain increase there should be a corresponding decrease in the amount of paperwork that will need to be completed (Zheng et al., 2016).

Blockchain can also be used to manage the reputation that a person or company has within a community. Each transaction or interaction that is completed will be added to the blockchain with the goal of allowing others to be able to evaluate their reputation within the community. By

monitoring reputation that a person or company has with a community; it will decrease the probability of a successful falsification of a company or a person's reputational records. Several companies have falsified their reputational records in the past by creating a large number of customer accounts and have them leave high feedback ratings on public reputational websites. In order for blockchain to be able to successfully manage the reputational records of a person or a company, it will need to be able to create a system that is distributed across a variety of hosts with the goal of consistently maintaining a consensus of the data that is contained within the various hosts. In addition to maintaining the reputational records across several hosts blockchain must be able to allow transactions to be completed rapidly. Blockchain could also be used to improve the reliability of data encryption and decryption that is known as "pretty good privacy (pgp)" which is used to transfer information. PGP is conceptually based upon a "web of trust" whereby "everyone is linked with a public and private key" and uses them in order to exchange information (Zheng et al., 2016, p. 14). A proposal has been made where a distributed PGP key server would be created with the goal of having the blockchain not only store the certificates but also support retrieving them which would increase the reliability of that particular PGP system (Zheng et al., 2016).

The use of blockchain with information technology as well as blockchain with IoT are a couple of areas that have been receiving much interest recently (McKee et al., 2017; Zheng et al., 2016). Combining blockchain with IoT will allow for improvements to be made in the fields of finance and healthcare (McKee et al., 2017). There are several sectors in information technology in which blockchain would benefit such as the Internet of Things (IoT), Privacy management, and software copyright management (Zheng et al., 2016).

The goal of improving the security of distributed networks is another application of blockchain in the information technology sector that has received some interest. The anti-malware application that was created was BitAV in which the users distributed the portions of the malware viruses across the blockchain with the goal of enhancing the fault tolerance. BitAV was able to increase the speed at which it scanned the network for viruses as well as making it less susceptible to a denial of service attack. Another implementation of blockchain in



the information technology sector is the supporting and upgrading of the network security infrastructure while using public key infrastructures (PKI). In 2015, Axon (2015) it was proven that blockchain could be used in order to build a PKI that was privacy-aware as well as improved the dependability of the currently established PKI. Lastly blockchain has been used to improve the security and dependability of “decentralized systems such as cloud systems and distributed databases” (Zheng et al., 2016, p. 15).

Blockchains can be used in logistics or supply chain management in order to track the location of a container that consists of a variety of assets as it moves from the manufacturer to the distributor or its final destination over one of several modes of transport. The blockchain network is used to monitor the location of the each of the assets that are stored within the container. In the past, each of the stakeholders would be responsible for tracking its own assets that are located in the container but if a blockchain were to be implemented then all of the information that is stored within each of the databases could be stored in a single distributed database which is shared among the stakeholders. This shared database would be able to receive updates that have been verified cryptographically, automatically propagated on the network, and are able to complete a trail of information that is auditable. Since the transaction has a cryptographical digital signature it will be able to be used as a delivery receipt by the shipping company to verify that container has reached its destination port. Whoever is receiving the container at the destination port will be able to amend the smart contract to reflect they are in possession of the container (Christidis & Devetsikiotis, 2016).

When a manufacturer produces a new IoT device it will function on the same blockchain network as the rest of that manufactures devices. The manufacturer will deploy a smart contract which will allow them to retain the hashes of the most recent software updates that are available on the network. When the devices ship they will either have the address of the smart contract permanently imprinted in the blockchain or it will be able to obtain the address through discovery. The contract is then queried in order to obtain the new firmware as well as requesting its hash via a distributed filesystem that supports the use of peer-to-peer. The initial request for this file will

be answered by the manufacturer's node which will broadcast the binary code to the nodes until it has been broadcast to a sufficient number of nodes whereby the manufacturer's node will stop broadcasting. The assumption is that the devices will be configured to share the code that they originally received. When another device joins the network after the device manufacturer has terminated its support it will continue to receive correct updates. This process of receiving updates will occur automatically and without any user collaboration. This process of centralized distribution can be compared with decentralized distribution implementation with IoT where the device will scan the manufacturer's server for an update and it will receive a 404 error. By using a blockchain network in which cryptocurrency is exchanged, it will provide an appropriate billing layer as well as creating a "marketplace of services between devices" (Christidis & Devetsikiotis, 2016, p. 2292). By having a cryptocurrency established it allows each device to have a bank account of its own on the internet which it can share with the other devices or users and get paid for their usage through microtransactions (Christidis & Devetsikiotis, 2016).

Since the device has access to its own bank account on the internet it is better equipped to support the sharing of services as well as property. One company that has become successful in the sharing of services as well as property is Slock.it. Slock.it is a device that supports the use of tokens which can be purchased on the Ethereum blockchain and used to unlock a smart electronic lock ("Slocks") (Christidis & Devetsikiotis, 2016, p. 2298). The owner of a slock that would like to rent their home or vehicle establishes the price of the scheduled access to the electronic door lock in which an interested party is able to identify using a mobile application, pay the amount that has been requested in Ethers and then communicate with the lock in order to unlock the door. The billing for slocks is simple because all of the slocks function on an identical blockchain (Christidis & Devetsikiotis, 2016).

The use of blockchains in the energy sector would allow for a peer-to-peer market which will allow machines to automatically purchase or sell energy based upon predetermined criteria. Transactive Grid is a company that has been testing the idea of using blockchain technology to enable a peer-to-peer energy market. Solar panels are used to document the energy that they

absorb, notate the disproportionate amount of input, report it on blockchain, and make it available for sale to neighboring power companies via smart contracts (Christidis & Devetsikiotis, 2016).

Another way that blockchain is currently being used is to implement an atomic communications model which is decentralized and allowing each party to initiate communications where each party has identical capabilities and either party can initiate communications. The process of atomic communications will support the exchange of tokens if the Bitcoin transactional model is followed. The manufacturer is allowed to circulate a “I have the container” token where the other stakeholders are permitted to respond with a “I have received the container” token (Christidis & Devetsikiotis, 2016, p. 2299). When the manufacturer transfers the carton to a freight transportation company which will transfer the container from one location to another; that company will create two sets of inputs and outputs where the first input originates from the UTXO of the manufacturer and its corresponding output will create a new UTXO that will lock the token to prevent it from being used by the transporter. The second input points towards the UTXO of the transporter whereby the second output will create a second UTXO token that will be transferred to the manufacturer. Once the resulting token is received the manufacturer signs their portion then sends the incomplete transaction back to the transporter for them to sign their portion and it is then added on to the blockchain. Once the transaction is appended to the blockchain, the “I have received the container” is delivered to the manufacturer from the transporter and the “I have the container token” is being held by the transporter (Christidis & Devetsikiotis, 2016, p. 2299). There will be a comparable atomic exchange that will be completed at the next few points as the carton moves from the shipping company to the distribution facilities to the location of the retailer when the retailer receives “I have the container token” (Christidis & Devetsikiotis, 2016, p. 2299). After the carton arrives at the destination there is a complete, timestamped trail that has been proven cryptographically which leaves a limited amount of room for the stakeholders to be able to dispute anything that has happened (Christidis & Devetsikiotis, 2016).

Filament is a company that has experienced success using sensors that are connected to “Taps” which are long-range radios (Christidis & Devetsikiotis, 2016, p. 2299). The Taps can

form a wireless mesh network that is able to communicate with the other radios using a technique called telehash that is secure as well as distributed. Taps also supports the ability to interact with each other using smart contracts on a standard blockchain. To reduce the deployment costs the sensors will not connect to the internet but they will use gateway nodes that support that kind of connectivity (Christidis & Devetsikiotis, 2016).

### 2.3 Not Completed Research

The above scenario can be greatly improved if a blockchain were to be implemented so as to allow the process to become completely automated using IoT. Each stakeholder must carry one of three internet connectible smart trackers which are BLE, GSM, or a LTE radio as well as having a blockchain client that is installed on the device. The container that is being tracked will have a corresponding tracker attached to it which will send out signed transactions when two of the stakeholders meet in one location to complete a handoff of the container. The transactions that the stakeholders will be sending will be able to be completed without any input from the users which will allow the container to be moved onto the next location as soon as the tokens have been swapped. There are several opportunities for research that focus on solely on CPS including the design and development of the next-generation of air and space transportation, hybrid vehicles that support the use of gas and electric together, completely automated urban driving, and the ability to integrate brain signals with prosthesis to allow them to physically move objects (Baheti & Gill, 2011).

#### 2.3.1 Internet of Things (IoT)

Recently there has been a considerable amount of interest in expanding the use of Blockchain to non-financial sectors such as the Internet of Things (IoT), provenance tracking, recordkeeping across different organizations, and the process of writing data to a centrally managed record or location. There are several ways that blockchain can be used with IOT and

some of them are security, device tracking, device security, and authentication. Additionally, there are several additional possibilities when you consider the combination of IoT, blockchains and other devices, applications, or services. There are several abilities and exchanges that IoT could help with implementing such as improving the security of IoT devices that are located on a network as well as continuing to ensure that the network itself remains secure (Ahluwalia, 2016; Samaniego & Deters, 2017).

### 2.3.2 Digital Notary

Imagine that instead of being required to go to the bank to get something notarized that you could use the internet to complete the same task. The process would be similar to having a notary validate every transaction that you complete on the internet to ensure that nothing is wrong but it would be computerized. This would provide businesses with a frictionless environment seeing as through the control is rooted in the transaction which may open the door to a variety of new possibilities for blockchain (Duivestijn, van Doorn, van Manen, Bloem, & van Ommeren, 2015).

### 2.3.3 Interorganizational Recordkeeping

Even though blockchain has been primarily used for financial transaction processing it has gained some interest in the non-financial sectors such as the collecting and recording of data as well as providing the ability to authenticate it. One type of data that blockchain could be used to authenticate is communications that occur between companies such as the signing of contracts or the exchange of information. The ability to authenticate information across the organizations would prevent issues because an individual company would not be the sole holder of all the information. By distributing the information across the companies that are involved would prevent one of the companies from falsifying or deleting information that would provide significant damage to the other companies that are involved. To solve the issue of ensuring that

one company does not delete information that would help the other company a shared database will be created. The shared database will contain all of the timestamped records as well as proof of origin for each record. An alternative to having the database shared between the two companies is to create a database with a trusted third-party who will store and maintain the records as well as allow the companies in question to be able to view the records without fear that the other company will damage them (Greenspan, 2016).

#### 2.3.4 Lightweight financial systems

Blockchain has been used to process a variety financial transactions from Bitcoin since its inception (Popper, 2015). There are two issues that have frequently reared their heads that must be solved in order for blockchain to be used for lightweight financial systems. The two issues are the double spending issue and creation of forgery coins (Greenspan, 2016). Blockchain has been able to resolve the issue of double spending bitcoin coins in which the coins from one transaction are also spent in another transaction (Pilkington, 2015). The second issue of preventing the creation of forgery cryptocurrencies is a little bit more difficult to resolve because blockchain uses virtual currency not physical currency so it is impossible to accept one for payment and make sure that you are accepting one that is not a forgery (Greenspan, 2016). Both of the above issues could be resolved by using public-key cryptography where every agent was given a public and a private key that were used in every transaction. The public key was shared with the other agents and original agent kept the private key to themselves and didn't share it with others. A blockchain transaction is started when the recipient of the cryptocash sends that their public key to the originating owner of the cash. The cryptocash is then transferred from the originating owner to the recipient using a hash of the digital signature. The public keys are addresses that are cryptographically generated and stored in the blockchain. Every coin or cryptocash has an address that is associated with it and each transaction that occurs in the crypto-economy is a transfer of coins from one address to another address. One of the most outstanding features of

blockchain is that there is no connection between the public keys that are used and the real-world identity of the originating user which means that even though the transactions are traceable they will not reveal the identity of the sender. Preventing blockchain from connecting the user ids to specific users is a major difference between it and fiat currencies (Pilkington, 2015).

### 2.3.5 Multiparty aggregation

Multiparty aggregation would be used in order to manage written data for a large number of sources so that it can be easily combined and analyzed as needed. One example of this is two banks notice that they have a large number of overlap between their customers and they agree to share information in order to reduce the volume of verifications that each needs to complete. Each of the banks would need to have read only access to the other banks database in order to query it in search of existing information on the customers as well as being able to query its own database. This would work if it was a small number of entities that need to share information but if the number of banks that wanted to share information with each other became too large then it is very likely that the system could break down due to being overloaded (Greenspan, 2016).

### 2.3.6 Provenance tracking

The ability to track a highly valued item as it traverses a supply chain, or a critical document have been tracking issues that have plagued those in a variety of markets for an extended amount of time. Most common issue that is encountered by those working in these markets is the risk of counterfeit and theft as they relate to shipping and document validity. These issues can be resolved by using blockchain by connecting a digital token to the item or document in question and every time it changes hands the digital token is moved in parallel which will provide a real-world chain of custody and transactional chain will be created on the blockchain. Additionally, it is possible that the physical item might not even need to be sent since the

blockchain will support creating a digital tag and attaching it to the document in question (Greenspan, 2016).

## 2.4 Needs

Due to the fact that there are several known issues with the testing of the CPS devices at the physical component level as well as the software level the ability to be able to test both aspects of the devices is a feature that needs to be researched and made available. Another need that should be made available is the ability to integrate the control, communication, and computational aspects in order to aid in the design and deployment of CPS. One major challenge is the processes that are used to design and implement the computing, timing, software, reconfiguration, control, and distributed decision-making support systems that are used to network the various systems in CPS devices together. The next aspect of CPS that needs to be improved is the current software and hardware components, the corresponding middleware, and operating systems need to be researched and developed so they are dependable, easy to reconfigure, and supported across multiple systems which means that the trustworthiness of the systems needs to be improved. Additionally, by identifying the current challenges, needs and opportunities that are available in a variety of industrial sectors as well as encouraging research collaboration between those in industry and academic research will make advancements in CPS research that will be easier to complete. By working together those that are researching in industry as well as academia will be able to create new systems that will be supported in science as well as engineering fields while building systems that will support the use of integrated cyber and physical systems which will work synergistically. Research has revealed that there are several opportunities and challenges in both biomedical and medical engineering, as well as neuroscience and cognition, air traffic control and management systems, disgruntled employees, cybercriminals, nation states, renewable energy research focusing on smart grid development, and terrorist, activists, and other organized crime groups (Baheti & Gill, 2011; Cárdenas et al., 2009). Cybercriminals have been



able to threaten the security of computers including the control systems no matter where they are located. Despite these attacks not having any intention of harming the system or systems that are being attacked there are several negative side effects that might be felt such as the control system might become infected with malware causing it to not operate properly (Cárdenas et al., 2009). Disgruntled employees have been a primary source of attacks that have been directed at control systems as evidenced by the attack in 2000 on the Maroochy sewage control system that is located in Queensland, Australia (Cárdenas et al., 2009; Slay & Miller, 2007). Despite the fact that there is not a substantial amount of evidence to prove that control systems have been successfully threatened or damaged by criminal organizations there is evidence that shows possible connections. The attacks that are primarily completed for terror or extortion reasons are physical attacks. The next step after physical attacks is cyber-attacks which can be completed from anywhere and means that they expose the attacker to less danger making them easier to organize as well as repeat and they are cheaper and easier to carry out (Cárdenas et al., 2009).

## 2.5 Issues

There have been numerous issues since blockchain, CPS, and IoT were first created but they have each managed to grow and evolve in their own way.

Even though CPS research is the youngest of the three it has probably encountered the most barriers. Most of the barriers that it has encountered have originated with research institutions as well as professional researchers which has caused a narrow definition in research as well as being very discipline specific and having specific educational venues that are directed at those in the academic fields of science and engineering. Research of CPS has been subdivided into groups of sensors, networks and communications, mathematics, control theories, computer science, and software engineering. The reason that subdividing the groups has proven to be such an issue is that systems are designed and analyzed using formalized modeling as well as modeling tools. Each model or tool focuses on particular features and ignores other features so as to ensure

that the analysis is properly connected to the original information. Research and testing have been able to show that a tool or modeling scheme is able to provide representation of either the processes that are occurring on the cyber or the physical side but not together. It is this divide and conquer mindset that is slowing the research and development of CPS as well as posing a serious issue for validating the research and verifying the safety features that are supported at the system level as well as how well do the physical and behavioral components work together (Baheti & Gill, 2011).

There are several anticipated as well as known issues that are related to the implementations of various blockchains and IoT devices (Christidis & Devetsikiotis, 2016). Ethereum has several known issues but they are technically related issues such as scalability, efficiency and security (Peck, 2016). Some of the issues with other blockchains that have been anticipated are increased transaction processing times, privacy, transactional privacy or confidentiality, legal enforcement of smart contracts, determining an appropriate miner to use, expected value of assets, the ability to self-govern itself, and the need for a Domain Name Servers (DNS) service, as well as communication and file exchanges that are secure (Christidis & Devetsikiotis, 2016).

The first issue that has been anticipated is that transaction processing time will be increased due the blockchain having a decentralized database which means that they will have to figure out the preferred paths to take in processing transactions (Christidis & Devetsikiotis, 2016).

The next difficult issue is the ability to maintain the privacy of those that are participating in the blockchain. Part of the reason that securing the privacy of the users is such a difficult task is that each device that is on the blockchain has a public key or hash which is used to identify it and it is all that an interested party needs to locate its counterpart. Since all blockchain transactions occur in the open then an interested party can examine the transactions to determine patterns and create associations among addresses in order to make conclusions about a person's actual identity. The privacy issues with public blockchains can be mitigated by having the IoT device use a new

key for every transaction or a different key for every party with which it has a transaction and if you are using a private blockchain then it is advised that different blockchains be used for different transactions in order to prevent another user from being able to track your devices activity in order to gain an advantage (Christidis & Devetsikiotis, 2016).

Another issue that has been forecasted is transactional privacy because in and of itself confidentiality is hard to attain due to content of each transaction being exposed to all nodes that are located on the network in order to achieve validation. Two ways to assist transactional privacy are to use homomorphic encryption, and zero-knowledge proofs but they are both very resource intensive so the ability to use them in IoT is questionable (Christidis & Devetsikiotis, 2016). Homomorphic encryption is a technique of executing calculations on information that is encrypted prior to decrypting it and it has been used in the past to improve the security of cloud computing (Greenberg, 2014). Zero-knowledge proofs are used when a statements or fact needs to be verified or proven without sharing unnecessary information with your adversary (Lynn, 2001). A blockchain can be established for a specific purpose such as maintaining privacy and discarded after its usefulness as a temporary workaround has ended (Christidis & Devetsikiotis, 2016).

A fourth issue that has been foreseen in relation to blockchain privacy is determining which miner set will provide your blockchain with the most functionality (Christidis & Devetsikiotis, 2016). Blockchain mining is where a review of the distributed computations is completed on each block of data that is located within the blockchain and it also allows a consensus to be achieved in an environment in which neither party knows or has expectations for their counterparts (Blockchain Technologies, 2016). Additionally, a miner is not able to falsify a transaction or edit a transaction that has already been completed but it can censor a transaction by preventing a valid transaction from being included in the blockchain. The consensus mechanisms have limited tolerance against nodes that are extremely complicated and if the number of miners is greater than the supported threshold then there is a greater risk that the transaction will be censored. The nodes that are used in a blockchain mining set need to be carefully selected so as to

reduce the chances of them conspiring together. If a private block is being used then a legal contract should be signed so as to ensure that if any collusions occur that they are punished in an appropriate manner (Christidis & Devetsikiotis, 2016).

Currently there is another issue which is the limited legal enforceability of smart contracts, but work is being done to rectify this as well as making them binding to all parties that are involved. One recommendation that has been made which will increase the chances that a smart contract is able to be enforced legally is that a reference to the actual contract be made in the smart contract and the actual contract should have a reference to the smart contract within it which is also known as dual integration. If a question of legality is brought up then having the cross reference between the two, will establish the link between the blockchain and the outcome that is actually expected (Christidis & Devetsikiotis, 2016).

The sixth anticipated issue is that several questions have been raised about the expected values of the assets that have been tokenized. Blockchains have been used to exchange the tokens due to the value that they hold but what happens if a device of yours believes that it owns the tokens that are on that chain and you decide that you want to cash in one of the tokens. How do you know that the exchange will be able to be completed or not? If the blockchain does not support smart contracts then it will not support the use of dual integration but it might support the hashing of an actual contract, by having the hash of the contract be included in the metadata of the token that will be exchanged. Anyone who is involved with an exchange of tokens using blockchain should complete a thorough examination of the assets in order to make their own assessment of the token values (Christidis & Devetsikiotis, 2016).

The next issue that has been anticipated is that prior to releasing a smart contract on a blockchain network someone should thoroughly inspect its logic as well as including preventative measures in the code that will avert issues. The reason that the code should be thoroughly reviewed is that someone else might be able to remote in, access the contract, destroy it, and thus remove it from the distributed virtual machine of the blockchain. If none of the above tasks are completed, then the blockchain will never be able to be modified which might not be a bad thing;

but what if there is a portion of the smart contract that has code that was written poorly or incorrectly; then the interactions that are completed cannot be reversed (Christidis & Devetsikiotis, 2016).

The final issue that should be considered prior to implementing a blockchain is that Domain Name Servers (DNS) should be created as well as options for communication and file exchange that are both secure. The reason that a DNS should be created is that it contains the pointers to the resources that are available. A communication option that is secure should be created because messages that are in the blockchain can be read by everyone who is on the network. If someone needs to communicate privately then a more secure communications protocol such as telehash or Whisper should be implemented (Christidis & Devetsikiotis, 2016). If there needs to be a more secure network for file sharing then InterPlanetary File System (IPFS) which is a distributed, peer-to-peer (p2p) file system with a goal of connecting the computing devices within one system of files (Christidis & Devetsikiotis, 2016; J. Guy, 2016).

## 2.6 Conclusions

The use of blockchain with CPS devices will improve the current security implementations as well as decrease the chances of further attacks.

## 2.7 Summary

This chapter provided a review of the literature relevant to the problems of CPS, security, and blockchain. The next chapter provides the framework and methodology to be used in the research of CPS, security, and blockchain.

## CHAPTER 3. RESEARCH METHODOLOGY

This chapter provides the methodologies and evaluations that are related to the research of CPS, security, and blockchain.

The purpose of this study is to decide what if any improvements can be made to the existing security of cyber-physical systems (CPS) by adding blockchain. CPS have several security vulnerabilities that have been used in the past few years to create cyber-attacks that have been directed at the US and its allies with the goal of crippling the cyber infrastructure and or causing damage to the physical systems that protect it. The issue is that if the security vulnerability is found then it is often done after the cyber-attack has been completed and the damage is done. The major benefit that would be realized from this research is that since blockchain is able to track modifications that are done to files then the receiving system would be able to review the changes that have been made and decide if a file has been changed in route as well who changed it. Additionally since the blockchain would be able to review the file modification history it would be able to release an alert in the blockchain network advising of the modification of the file and to validate any updates that have been made as well as transmissions that have been received. Lastly the ability of being able to obtain as well as verify the coordinates of a device will decrease the chances of a device being lost, damaged, or stolen because the devices coordinates will be appended to the blockchain at intervals that will vary based upon needs.

### 3.1 Research Approach and Hypothesis

The research methods that were used in this study were designed to improve the security of the existing CPS network by creating a connection to an existing blockchain with the goal of tracking the information as it traverses the network from its origin point to its destination and can

provide documentation showing the history of changes and the network path that it took. The hypothesis that are related to the research question in Section 1.3 are:

- $H_0$ : FOAM implemented on Tendermint will be able to run more efficiently than FOAM on Ethereum.
- $H_A$ : FOAM implemented on Tendermint will not be able to run more efficiently than FOAM on Ethereum.

### 3.2 Testing Equipment

The summer of 2017 an independent research project was completed on blockchain whereby one of the research assignments was to create a couple of virtualized blockchains to determine which of the existing blockchains were the most efficient to set up and the procedures that were needed to complete the implementation. Additionally, the security options that were currently available for CPS devices were reviewed and an analysis of the results was completed. Additionally, a literature review of the FOAM protocol as well as associated research has been started. The second task that has been completed is to select Tendermint Blockchain as the one that will be used for testing the FOAM protocol. After thoroughly researching the computational requirements for Tendermint Blockchain as well as those that are needed for Ethereum Smart Contracts the hardware that has been determined to be appropriate for testing purposes is a Raspberry Pi 3B and a Raspberry Pi 3B+ which have both been purchased. The fifth task that has been completed is to download and install Raspbian on both the Raspberry Pi 3B and the Raspberry Pi 3B+. Another task that needs to be done is completing the run time tests on both of the Raspberry Pi's. The next task that will need to be completed after the run time tests are done is to install Ethereum on the Raspberry Pi 3B and Tendermint Blockchain on the Raspberry Pi 3B+. After the installations are finished the post Blockchain installation run time tests will need to be done. The next set of steps that will need to be completed is to download and install the Proof of Location on both of the Raspberry Pi's. Completing the post-installation of Proof of Location run

time tests on both of the Raspberry Pi's is the next set of tasks that will have to be completed. The next to the last set of research and testing tasks that will be completed is to run the Proof of Location and test its accuracy. The last set of tasks that will need to be completed for testing is to swap the memory cards from each of Raspberry Pi's into the other Raspberry Pi with the goal of deciding if the Blockchains run any differently on the other Raspberry Pi. The last task that will need to be completed is to compare the data and analyze results that were obtained in testing.

### 3.3 Testing conditions

Several experiments will have to be created and ran in a virtual environment using one of several blockchains. By running the experiment in a virtualized environment, it will be easier to maintain a consistent test environment, a secure network, and ensure that the data is not corrupted by outside forces. Using several blockchains will allow with the goal of determining which blockchain will work the most efficiently with each CPS because they will be able to be tested individually without any negative impacts from the other blockchains. The goal of each experiment will be to provide a confirmation of the feasibility of the implementation as well as the verifying if the security implementations will be successful. Additionally, by using these testing conditions, it will allow for a more refined analysis of the successes and failures of the security and highlight the cases where there is an anomaly in the results, which will lead to a review of the testing methods to determine if there was a variation in the testing that was used. Due to the current knowledge that has been obtained through research, there are several factors that are related to the methods that will be used for testing that have yet to be determined but the current plan will be a series of steps. These steps will be to: (1) create the blockchain, (2) determine the information that will be sent to the CPS, (3) create a connection between the blockchain and the data that needs to be transmitted, (4) transmit the data, (5) verify that the data is received by the proper device in the network, and (6) analyze the transmitted data for any changes that were made. If the data is not delivered to the proper device then an analysis will be



completed on the path that the data took to determine what changed its route and why it was changed. Lastly, the path that the data takes will be compared with the blockchain to confirm the data transmission that the payload says that it took.

### 3.4 Data and Measurements

Since the goal of this research is to decide if implementing blockchain as a form of validation and tracking will decrease the number of successful attacks on CPS then there are a few aspects that need to be determined including what type of data will be used, how will it be obtained and what result would provide a favorable outcome. The data that will be collected and analyzed will be based upon two factors. The first set of measurements that will be taken and analyzed will be used to determine if the location was able to be verified successfully or not and did the blockchain notate the time and coordinates of the location. The second set of measurements that will be collected will be the runtime data from each step of the setup as well as the installation processes on each of the Raspberry Pi's. After the data from the two-separate test groups is reviewed and analyzed then it will be combined to decide if there are any connections can be determined between the runtime data and the location verification on either one of the Raspberry Pi's. Depending on the results from these analyses there is further research and testing that may have to be completed. The outcomes that would provide the most favorable results would be able to determine if one or both of the blockchains is able to successfully run the proof of location protocol on one or both of the Raspberry Pi's as well as being able to correctly report the locations coordinates.

### 3.5 Threats to Validity

Due to the continuous changes that are made to the CPS devices there are a variety of updates that will need to be made to software as well as devices and this could cause some inconsistencies in the testing and results. Additionally, since the technology is continuously

changing the devices that would be available in a year, two years or even three years have not only not been thought of or designed yet.

When a device keeps receiving the same kind of data transmissions or payloads its security software will stop providing alerts on those that contain malware thus causing alerts that are false negatives to attack methods. Additionally, it may start to flag those transmissions that have valid information causing false negatives in security scans which will prevent the payload of data from properly traversing the network. The payload could have anti-malware updates, launch codes, or other forms of data that would need to be transmitted in a secure method.

When a new blockchain is created there are certain properties that are required to be included, some of these properties are recommendations and due to the inconsistencies in the properties that are included in each blockchain determining what blockchains will provide the largest amount of properties will be a challenge.

### 3.6 Summary

This chapter described the research methodologies and evaluations that have been completed on CPS, security, and blockchain.

## CHAPTER 4. RESULTS, AND ANALYSIS

### 4.1 Introduction

The purpose of this research was to determine if it was feasible for a blockchain to be able to implement proof of location. This portion of the document contains sections that review the equipment that was selected as well as why it was selected, the feasibility criteria, the testing environment, data analysis, FOAM dependencies, successes and lessons learned during testing, and implementation procedures.

#### 4.1.1 Equipment Selection

There were several reasons that the Raspberry Pi 3B and Raspberry Pi 3B+ were selected for the testing environment but the five primary reasons were financial, ease of replacement, portability, the specifications on the Raspberry Pi 3B corresponded to the minimum requirements that were needed to run the location verification protocol, and the operating systems that they were able to support. The first and second reasons that the Raspberry Pi's were selected were mainly financial reasons. The first reason that the Raspberry Pi's were selected for testing was their ability to be purchased cheaply, and if something were to happen to one or both of them, they could be easily, quickly, and efficiently replaced with minimal inconvenience. The second financial reason that they were significantly cheaper than purchasing another similar device such as a BeagleBoard, PixelPro, or other equipment such as a server (Kurve, 2017). Currently a Raspberry Pi 3B can be purchased on Amazon for \$35.00 - \$37.00 by itself or it can be purchased as the Ultimate Starter Kit with a 32 GB SD Card for approximately \$90 (*Amazon.com: CanaKit Raspberry Pi 3 B+ Ultimate Starter Kit*, 2018; *Raspberry Pi 3 Model B Motherboard*, 2016). Additionally, a Raspberry Pi 3B+ can be purchased for around \$50 by itself or it can be purchased

as a Raspberry Pi 3B+ Ultimate Starter Kit for \$90.00 (*Amazon.com: CanaKit Raspberry Pi 3 B+ Ultimate Starter Kit*, 2018; *Raspberry Pi 3 B+ Motherboard*, 2018). Another reason that the Raspberry Pi's were selected to be used is because they are highly portable and would have been significantly easier to move than a server or a computer. The computational specifications on the Raspberry Pi 3B corresponded to the minimum specifications that were needed to run "the BFT time-sync protocol" which is the clock cycle that will be used to calculate the distance and therefore enable the location verification protocol ("Foamspace Corp", 2018b, p. 26). The last reason that the Raspberry Pi's were chosen for testing was the operating system that they were running on which the researcher was able to review some of the code for FOAM's time-sync protocol and determined that it was running on a Linux based operating system.

	<b>Raspberry Pi 3B</b>	<b>Raspberry Pi 3B+</b>
<b>Chip</b>	Broadcom BCM2837 64-bit	Broadcom BCM2837B0 64-bit
<b>Processor</b>	ARM Cortex-A53 quad core	ARM Cortex-A53 quad core
<b>Processor Speed</b>	1.2 GHz	1.4 GHz
<b>Voltage and Power Draw</b>	750mA @ 5V	750mA @ 5V
<b>GPU</b>	Dual Core VideoCore IV	Dual Core VideoCore IV
<b>Size</b>	85x56mm	85x56mm
<b>Memory</b>	1 GB SDRAM @ 900 MHz	1 GB SDRAM @ 900 MHz
<b>Storage</b>	Micro SD Card	Micro SD Card
<b>GPIO</b>	40	40
<b>USB 2.0</b>	4	4
<b>Ethernet</b>	10/100mb Ethernet RJ45 Jack	10/100/1000mb Ethernet RJ45
<b>Wireless LAN</b>	Integrated 802.11 b/g/n	Integrated 2.4GHz and 5GHz 802.11 b/g/n/ac
<b>Bluetooth</b>	Integrated Bluetooth 4.1 (Classic & Low Energy)	Integrated Bluetooth 4.2 (Classic & Low Energy)
<b>Audio</b>	Multi-Channel HD Audio over HDMI,	Multi-Channel HD Audio over HDMI,

*Figure 4.1. Chart Comparing the Raspberry Pi 3B and Raspberry Pi 3B+ (Raspberry Pi comparison chart, 2018)*

#### 4.1.2 Feasibility Criteria

The feasibility criteria for this research was determined in three phases which were to review the computation abilities of the Raspberry Pi 3B, the hardware requirements of the

blockchains to run as well as the Raspberry Pi's ability to be able to install software that would be required for testing. In the first phase the feasibility criteria that were considered was the hardware, software, and computational abilities of a Raspberry Pi 3B. The goal of reviewing this information was to determine a standard set of specifications that would be used for testing. Additionally, using the specifications of the Raspberry Pi 3B as the minimum computational requirements the theory was that if something failed on the Raspberry Pi 3B then it would be substantially easier to select better equipment such as the Raspberry Pi 3B+ since it had a few improvements over the Raspberry Pi 3B. The next phase that was used to determine the feasibility of this research was to review the hardware, software, and computational requirements of a variety of blockchains. Additionally, each of the use case implementations of the blockchains was reviewed to help in determining which blockchains would work best with this research. The ability to be able to successfully install and update software that would be needed to complete this research was the third set of criteria that were reviewed to determine feasibility.

Since the Raspberry Pi 3B had been previously purchased it was determined that the blockchains as well as the software would have to be able to be supported on its specifications. The hardware specifications for the Raspberry Pi 3B were a 1.2 GHz processor, 1 GB of Ram, it supported lightweight Linux-based systems, it used a MicroSD card for its internal storage, and it also supported using external storage if the internal storage became too full. Additionally, there were other devices that were reviewed including a Raspberry Pi 2B, Raspberry Pi 1B+, Asus Tinker Board, Parallela, BeagleBoard, and a PixelPro that were not selected (Kurve, 2017). The Raspberry Pi 1B+ and the Raspberry Pi 2B were not selected because their processors were less than 1 GHz (*Products Archive - Raspberry Pi*, 2015; *Raspberry Pi 2 Model B*, 2015). The BeagleBoard, and the PixelPro were not selected to be used in testing because of their higher price points (Kurve, 2017). Lastly, there were a few devices such as the Asus Tinker Board and the Parallela that had excellent hardware specifications, but the researcher felt that these devices had specifications that were too good and for this reason they would not be supported in this research (*ASUS SBC Tinker board RK3288*, 2017; Kurve, 2017).

#### 4.1.3 Blockchain Feasibility

In addition to completing a review of the feasibility criteria for the hardware that would be used in the research, a review of the feasibility criteria for the available blockchains was completed. The goal of completing a review of the current blockchains was to determine which of them would be the most feasible for implementation. Some of the feasibility requirements that were used to select the blockchains that were used for testing were their supported operating systems, processor and computational abilities, memory, and storage requirements. Additionally, the maximum storage space that could be required to store the blockchain was 32 GB. Some of the other blockchains that were reviewed were Neo, Waves Platform, and Nxt Platform which is similar to Ethereum as well as Casper Friendly Finality Gadget (FFG), Cardano, and Tomochain which were similar to Tendermint (Cam PHAM, 2018; Ciobanu, 2017). Casper FFG was not selected because its use case was not similar to this research and it “is an overlay atop a proposal mechanism” (Cam PHAM, 2018, para. 10). The reason that Cardano was not selected to be used in testing was because it utilizes stakeholders over validators and the researcher determined that having validators was going to be more advantageous to completing the research. The third blockchain that was considered as an alternative to Tendermint was Tomochain which had not released a stable version of its MainNet when testing started (Cam PHAM, 2018). Please view image 4.2 since it provides a more detailed comparison of EOS, Ethereum, Cardano, Tendermint, and Tomochain blockchains.

The first blockchain that was similar to Ethereum in which the feasibility of its usage was reviewed was Neo and the reason it was not selected was because it was an international based blockchain. The next blockchain in which its feasibility was reviewed was Waves Platform which is based in Europe and its token value is tied to the European (EURO Symbol) but it did not support being used with the US dollar. The final blockchain in which its feasibility was reviewed was Nxt Platform and despite its claims that it could run a full node on a Raspberry Pi it was not selected to be used in this research because it was programmed in Java (Ciobanu, 2017).

COMPARISON CRITERIA	EOS	ETHEREUM	CARDANO	Tendermint	TOMOCHAIN
CONSENSUS	DPoS	Chain-based PoS with slashing	Ouroboros PoS	BFT-based PoS	PoS
DECENTRALIZATION	21 validators	Unlimited number of validators joining block creation	Unlimited number of stakeholders joining block creation	Number of validators > 4	99 masternodes
SECURITY	BFT	BFT	BFT	BFT	BFT
PERFORMANCE	Potential to scale to millions of transactions per second	No information	On average 257.6 per second in experimentation with 40 nodes but the potential is more than that	Up to 10,000 transactions per second for 250 byte transactions	Potential to many thousands of transactions per second
ROADMAP	Mainnet launched in Q2 2018	No information	Centralized mainnet released on 29/9/2017, decentralized mainnet before Q3 2018	Tendermint core 0.20.0 released on 7/6/2018, 1.0.0 release date not clear	Mainnet will be released by Q4 2018
ECOSYSTEM	Refer to [ 1 ]	Refer to [ 2 ]	Refer to [ 3 ]	Refer to [ 4 ]	Refer to [ 5 ]

Figure 4.2. Image of a chart comparing various blockchains including EOS, Ethereum, Cardano, Tendermint, and Tomochain (Cam PHAM, 2018, fig. 1)

There were several reasons why Ethereum was selected to be used as a blockchain for this testing including the existing connection that Ethereum has with FOAM, its successful implementation on a Raspberry Pi, the researcher's previous knowledge and success with it, the ability to be able to implement Ethereum on a Linux based system with minimal difficulty as well as its minimal storage requirement, its use of smart contracts, and it supports the trading of cryptocurrency. The first reason that Ethereum was selected over other blockchains is because of the fact that FOAM was currently using it and it was a known working factor that would not provide issues. The next reason that Ethereum was selected is because of the fact the researcher had previously completed testing and research with it and felt comfortable using it. The third reason that Ethereum was selected over other blockchains is the number of substantiated reports that it had been able to be installed on a Raspberry Pi with minimal issues. The fourth reason that Ethereum was selected was the fact that it used smart contracts to update its blockchain. Another reason that Ethereum was selected over other blockchains was because of the fact it had an established track record of being able to run on Linux based systems with minimal difficulties. The next reason that Ethereum was selected was that there were several reports that it was able to completely sync the Ethereum blockchain to it despite reports that it was over 100 GB using Geth with Fast sync

There were several reasons why Tendermint was selected to be used as an alternative blockchain for this research including its existing connection with FOAM, its reports of being a lightweight blockchain as well as its limited storage requirements, reports of being more fault tolerant than other blockchains, its ability to be supported on Linux based systems, the use of BFT in order to achieve consensus as well as its ability to trade cryptocurrency. The first reason that Tendermint was selected to be used for this research is the fact that FOAM has already been successfully implemented on Tendermint. The next reason that it was selected is because of its claims to be a lightweight blockchain and it would take up less storage space than other blockchains. According to the documentation that has been released by Tendermint the minimum processor and hardware specifications that are required to run Tendermint are as follows: "1 GB



RAM, 25GB of disk space, and 1.4 GHz CPU” (*Tendermint core: Running in production*, 2018, para. 31). Additionally according to the Operating System Documentation that is available for Tendermint it “can be compiled for a wide range of operating systems thanks to Go language” (*Tendermint core: Running in production*, 2018, para. 32). There are several reports of Tendermint being more fault tolerant than other blockchains because it used Byzantine Fault Tolerance (Konstantopoulos, 2017). The lowest Operating System that Tendermint claims to be supported on is Linux with an ARM architecture (*Installing Go from source*, 2018; *Tendermint core: Running in production*, 2018). Based upon the minimal operating system requirements as well as the processor and hardware needs that were previously mentioned Tendermint should have been able to successfully run on the Raspberry Pi 3B+. The process that it used order to achieve consensus is another factor that was taken into consideration when determining if it would be used. Tendermint’s ability to be supported on a variety of Linux based systems is another reason why it was selected to be used in this testing. The final reason that Tendermint was selected to be used in this research is the fact that it has a cryptocurrency that can be traded. After comparing the previously mentioned blockchains with Ethereum and Tendermint it was decided that Ethereum and Tendermint would be used for completing the testing.

#### 4.1.4 Software Installation Feasibility

After completing a review of the feasibility criteria for the hardware that would be used in the research as well as the blockchains that were available the software requirements were reviewed. There were a variety of the applications that would need to be installed on the Raspberry Pi 3B as well as the protocols that would need to be able to run or sync including Go version 1.10, and the BFT time-sync protocol.

There were minimal installation requirements for the BFT time-sync protocol. The BFT time-sync protocol is a protocol in which the network will utilize a “high-precision BFT clock signal” to calculate “the relative geometry between beacons to compute” the distance to a node

thereby enabling a system that is secure as well as a “spatially distributed location system” (“Foamspace Corp”, 2018a, 2018b, p. 3). “The BFT time-sync protocol” itself is “hardware agnostic” but it does specify that the hardware is required to have a few features including a radio transceiver, power, the ability to send and receive messages, ability to be able to connect to the internet, “clock source with a frequency of at least 1 GHz”, as well as the ability to be able to make “its logs publicly available to validators” (“Foamspace Corp”, 2018a, 2018b, p. 26). Since there were no additional hardware requirements that were found the assumption was made that these were the computational requirements that would be needed to run the location verification protocol.

#### 4.1.5 Analysis of Feasibility

The previously stipulated feasibility requirements of this research were reviewed including the computational abilities of the Raspberry Pi 3B, the hardware and computational requirements of the blockchains that would be needed to sync them as well as the additional software and applications that would need to be installed. After reviewing the previously mentioned requirements and information it was determined that this research was feasible to be completed but that a Raspberry Pi 3B+ should be purchased to ensure that Tendermint would have an adequate testing environment.

#### 4.2 Testing Environment

This section will provide details about the testing environment including the testing conditions information such as location, how long the testing was done for, the equipment that was used during testing, as well as a network diagram of the testing environment and a picture of the equipment that was used in the testing.

#### 4.2.1 Testing Conditions

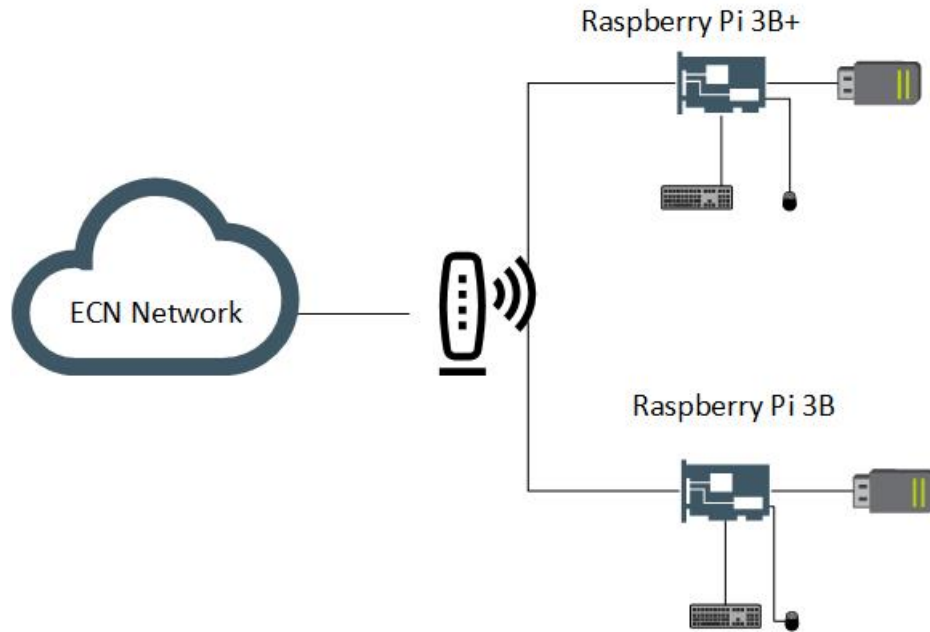
The testing was completed inside Wang hall on the fourth floor. The testing was completed over eighteen days from October 20th to November 6th in which the first five days of testing the Ethereum Blockchain synced intermittently on the Raspberry Pi 3B and the Raspberry Pi 3B+. The reason that the Ethereum blockchain synced intermittently was the researcher wanted to monitor the temperature as well as the voltage of both of the devices because there had been a few temperature and voltage spikes that were detected. From October 25th to November 6th the Ethereum Blockchain was permitted to continuously sync on the Raspberry Pi 3B as well as the Raspberry Pi 3B+ because the temperature and the voltage readings had significantly stabilized within the previous couple of days. The equipment that was used for testing is listed below:

1. Raspberry Pi 3B
2. Raspberry Pi 3B+
3. 4 Patriot 128GB SuperSonic Rage 2 USBs
4. 4 Sandisk Ultra 32GB Micro SDHC
5. NetGear AC1200 Dual Band Smart Wifi Router, Gigabit Ethernet (R6230)
6. HDMI Switch 4K HDMI Splitter
7. Cat 7 Ethernet Cables

The image 4.3 is a network diagram of the testing environment. The image 4.4 is a picture that was taken of the testing equipment.

#### 4.3 Discussion

In this section there are a variety of topics that are discussed including the data analysis, the FOAM dependencies, Purescript dependencies, Haskell dependencies, Kubernetes



*Figure 4.3.* Network Diagram of the Testing environment

dependencies, how the dependencies work together, FOAM Map, functionality of the FOAM Map, successes as well as the lessons that were learned in research.

#### 4.3.1 Data Analysis

This feasibility study was completed with the goal in mind of determining what was the probability of being able to use a blockchain to implement proof of location. The goal of this section is to review the data that was obtained during research and testing and connect it back to the feasibility study that was completed.

At the beginning of testing there were several instances of high temperatures as well as high voltages that were detected on Raspberry Pi 3B and the Raspberry Pi 3B+ while they were syncing the Ethereum Blockchain. As they continued to sync the Ethereum Blockchain there were significantly fewer instances of the temperature spikes on the Raspberry Pi 3B+ while the Raspberry Pi 3B continued to experience the occasional temperature and voltage spike. Due to



*Figure 4.4.* Overhead of testing Environment

having observed some of the temperature and voltage spikes on both of the devices the researcher decided to write a script to monitor the date and time as well as the temperature, voltage, the system performance, and the processes that were running. When the researcher reviewed the data that was collected from the Raspberry Pi 3B as well as the Raspberry Pi 3B+ the researcher immediately noticed that there was a strong connection between the temperature and the voltage increases. After completing a more in-depth review of the data that was collected it was discovered that the temperature and voltage spikes were not dependent upon each other but other factors. After reviewing the data, it was determined that there were seven factors that were caused the temperature and voltages to increase on the Raspberry Pi 3B+ and especially on the Raspberry Pi 3B. The seven factors were number of processes that were waiting to gain access to the processor, the large amount of memory that had been swapped out to another location, the limited availability of free memory, increase in the number of data blocks that were written to the disk as well as an increase in the utilization of the memory cache, and the number of system interrupts that were being seen as well as the context switches that the system was needing to complete. The varying combinations of the factors caused differing impacts to temperature as well as voltage. Some of the causes of the factors were easy to spot such as an increase in the processor utilization which was connected to the continuous syncing of the Ethereum Blockchain. As the Ethereum was syncing it caused an increase in the power that was needed as well as it increased how hard the processor was having to work to catch-up to the current block in the Ethereum node and remain caught-up. The images 4.5, 4.6, and 4.7 are three screenshots that were taken of command line text and its output showing the date and time that the readings were taken as well as the temperature and voltage of the Raspberry Pi 3B. The image 4.8 shows the highest recorded temperatures on the Raspberry Pi 3B and the temperature reading of 84.4 degrees Celsius was 0.6 degrees Celsius under the maximum safe operating temperature for a Raspberry Pi 3B. The image 4.9 is a figure of a screenshot that was taken of the command line text and its output showing the date and time that the readings were taken as well as the temperature and voltage of the Raspberry Pi 3B+. The image 4.10 is a figure that shows the temperature and voltage readings that were

taken on the Raspberry Pi 3B+ that were at the same time as the readings on the Raspberry Pi 3B. As you can see the Raspberry Pi 3B figure 4.8 has a temperature that is approximately 20 to 25 degrees Celsius higher than the same time frame as the Raspberry Pi 3B+ figure 4.10. The figure 4.11 is an image of a graph that contains a comparison of the temperature data that was collected on the Raspberry Pi 3B and the Raspberry Pi 3B+ from October 20th to November 6th. The figure 4.12 is an image of a graph that contains a comparison of the voltage data that was collected on the Raspberry Pi 3B and the Raspberry Pi 3B+ from October 20th to November 6th. Additionally, the figures 4.11, and 4.12 have been separated into images that contain the daily graphs of the time and temperature as well as the time and voltage for the Raspberry Pi 3B and the Raspberry Pi 3B+ which are detailed below. The figures 4.13, 4.15, 4.17, 4.19, 4.21, and 4.23 show the initial instability of the daily temperature measurements that were taken on the Raspberry Pi 3B from October 20th to October 25th. The daily time and voltage measurements that were taken on the Raspberry Pi 3B can be found within the figures 4.14, 4.16, 4.18, 4.20, 4.22, and 4.24 are from October 20th to October 25th and show the initial instability of the daily voltage measurements. The researcher would like to point out that that the daily images establish the instability of the temperatures and the voltage measurements that were taken as well as the overall figures. The figures 4.49, 4.51, 4.53, 4.55, 4.57, and 4.59 show the minor instability of the daily temperature measurements that were taken on the Raspberry Pi 3B+ from October 20th to October 25th. The figures 4.50, 4.52, 4.54, 4.56, 4.58, and 4.60 shows the instability of the daily voltage measurements that were taken on the Raspberry Pi 3B+ from October 20th to October 25th. The figures 4.25, 4.27, 4.29, 4.31, 4.33, 4.35, 4.37, 4.39, 4.41, 4.43, 4.45, and 4.47 show the decreasing instability of the daily temperature measurements that were taken on the Raspberry Pi 3B from October 26th to November 6th. The figures 4.26, 4.28, 4.30, 4.32, 4.34, 4.36, 4.38, 4.40, 4.42, 4.44, 4.46, and 4.48 shows the decreasing instability of the daily voltage measurements that were taken on the Raspberry Pi 3B from October 26th to November 6th. The figures 4.61, 4.63, 4.65, 4.67, 4.69, 4.71, 4.73, 4.75, 4.77, 4.79, 4.81, and 4.83 establishes the daily temperature measurements that were taken from October 26th to November

6th on the Raspberry Pi 3B+ were remaining fairly stable. The figures 4.62, 4.64, 4.66, 4.68, 4.70, 4.72, 4.74, 4.76, 4.78, 4.80, 4.82, and 4.84 were from October 26th to November 6th which can be used to establish that the daily voltage measurements from on the Raspberry Pi 3B+ continued to remain stable.

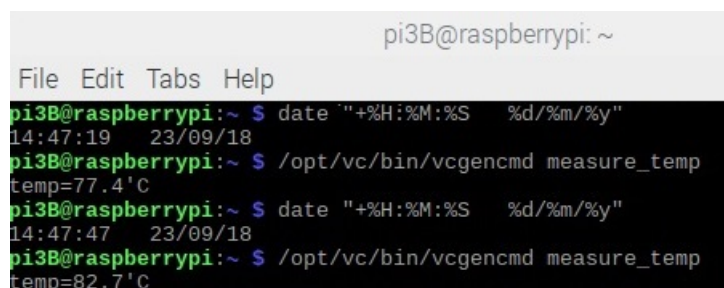


```

pi3B@raspberrypi: ~
File Edit Tabs Help
pi3B@raspberrypi:~ $ /opt/vc/bin/vcgenclmd measure_temp
temp=63.4'C
pi3B@raspberrypi:~ $ /opt/vc/bin/vcgenclmd measure_temp
temp=64.5'C
pi3B@raspberrypi:~ $ /opt/vc/bin/vcgenclmd measure_temp
temp=68.8'C
pi3B@raspberrypi:~ $ /opt/vc/bin/vcgenclmd measure_temp
temp=78.4'C

```

*Figure 4.5.* Screenshot showing some of the higher temperatures that were observed on 9/23 on the Raspberry Pi 3B



```

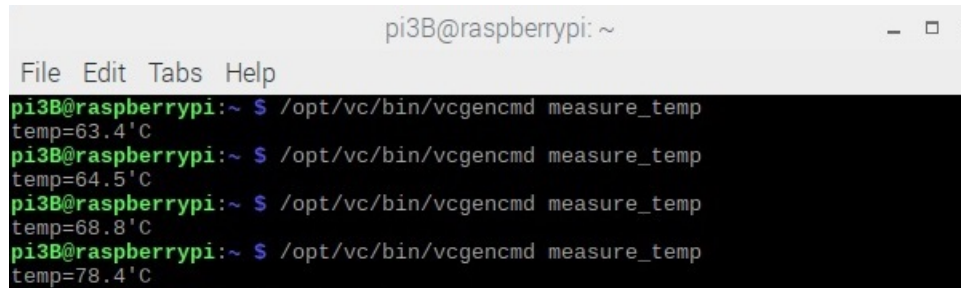
pi3B@raspberrypi: ~
File Edit Tabs Help
pi3B@raspberrypi:~ $ date "+%H:%M:%S %d/%m/%y"
14:47:19 23/09/18
pi3B@raspberrypi:~ $ /opt/vc/bin/vcgenclmd measure_temp
temp=77.4'C
pi3B@raspberrypi:~ $ date "+%H:%M:%S %d/%m/%y"
14:47:47 23/09/18
pi3B@raspberrypi:~ $ /opt/vc/bin/vcgenclmd measure_temp
temp=82.7'C

```

*Figure 4.6.* Screenshot showing some of the higher temperatures that were observed on 9/23 on the Raspberry Pi 3B

Additionally, once the Raspberry Pi 3B and the Raspberry Pi 3B+ had synced the Ethereum to what is known as the caught-up point then the processors on both of them were not having to work as hard and there was a significant stabilization in the temperature and voltage measurements.





A screenshot of a terminal window titled "pi3B@raspberrypi: ~". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The terminal shows four consecutive executions of the command `/opt/vc/bin/vcgencmd measure_temp`. Each execution is preceded by a green prompt `pi3B@raspberrypi:~` and a blue dollar sign `$`. The output of each command is `temp=` followed by a temperature value in degrees Celsius. The temperatures shown are 63.4, 64.5, 68.8, and 78.4.

```
pi3B@raspberrypi:~ $ /opt/vc/bin/vcgencmd measure_temp
temp=63.4'C
pi3B@raspberrypi:~ $ /opt/vc/bin/vcgencmd measure_temp
temp=64.5'C
pi3B@raspberrypi:~ $ /opt/vc/bin/vcgencmd measure_temp
temp=68.8'C
pi3B@raspberrypi:~ $ /opt/vc/bin/vcgencmd measure_temp
temp=78.4'C
```

*Figure 4.7.* Screenshot showing some of the higher temperatures that were observed on 9/23 on the Raspberry Pi 3B

Raspberry Pi 3B		
Date/Time	Temperature in Celsuis	Voltage
10/20/18 16:20:03	53.7000	1.2000
10/20/18 16:21:03	64.5000	1.2750
10/20/18 16:22:03	67.1000	1.2750
10/20/18 16:23:03	54.8000	1.2000
10/20/18 16:24:03	56.9000	1.2000
10/20/18 16:25:03	55.8000	1.2000
10/20/18 16:26:03	55.8000	1.2000
10/20/18 16:27:03	53.7000	1.2000
10/20/18 16:28:03	54.8000	1.2750
10/20/18 16:29:03	63.4000	1.2750
10/20/18 16:30:03	75.8000	1.2750
10/20/18 16:31:03	80.6000	1.2750
10/20/18 16:32:04	82.7000	1.2750
10/20/18 16:33:04	82.2000	1.2750
10/20/18 16:34:04	80.6000	1.2750
10/20/18 16:35:04	83.8000	1.2750
10/20/18 16:36:05	83.3000	1.2750
10/20/18 16:37:05	83.8000	1.2750
10/20/18 16:38:06	83.8000	1.2750
10/20/18 16:39:07	84.4000	1.2750
10/20/18 16:40:07	68.8000	1.2000
10/20/18 16:41:07	65.0000	1.2000
10/20/18 16:42:07	62.3000	1.2000
10/20/18 16:43:07	60.1000	1.2000
10/20/18 16:44:07	58.0000	1.2000
10/20/18 16:45:07	58.5000	1.2000
10/20/18 16:46:07	61.2000	1.2000
10/20/18 16:47:07	59.1000	1.2000
10/20/18 16:48:07	58.0000	1.2000

Figure 4.8. Temperature Spike on the Raspberry Pi 3B

```
pi3B+@raspberrypi:~ $ vcgencmd measure_temp
temp=55.8'C
pi3B+@raspberrypi:~ $ date "+%H:%M:%S %m/%d/%y"
21:08:19 09/26/18
pi3B+@raspberrypi:~ $ vcgencmd measure_temp
temp=79.5'C
pi3B+@raspberrypi:~ $ date "+%H:%M:%S %m/%d/%y"
21:30:19 09/26/18
pi3B+@raspberrypi:~ $ vcgencmd measure_temp
temp=80.1'C
pi3B+@raspberrypi:~ $ date "+%H:%M:%S %m/%d/%y"
21:40:45 09/26/18
pi3B+@raspberrypi:~ $ vcgencmd measure_temp
temp=69.3'C
pi3B+@raspberrypi:~ $ date "+%H:%M:%S %m/%d/%y"
21:42:58 09/26/18
pi3B+@raspberrypi:~ $ vcgencmd measure_temp
temp=56.4'C
pi3B+@raspberrypi:~ $ date "+%H:%M:%S %m/%d/%y"
22:11:01 09/26/18
pi3B+@raspberrypi:~ $ □
```

*Figure 4.9.* Screenshot showing some of the higher temperatures that were documented on the Raspberry Pi 3B+

Raspberry Pi 3B+		
Date/Time	Temperature in Celsius	Voltage
10/20/18 16:20:28	55.8000	1.2000
10/20/18 16:21:28	55.8000	1.2000
10/20/18 16:22:28	55.8000	1.2000
10/20/18 16:23:28	55.8000	1.2000
10/20/18 16:24:28	56.4000	1.2000
10/20/18 16:25:28	55.8000	1.2000
10/20/18 16:26:28	55.8000	1.2000
10/20/18 16:27:28	55.8000	1.2000
10/20/18 16:28:29	56.9000	1.2000
10/20/18 16:29:29	56.4000	1.2000
10/20/18 16:30:29	56.4000	1.2000
10/20/18 16:31:29	56.4000	1.2000
10/20/18 16:32:29	54.8000	1.2000
10/20/18 16:33:29	56.4000	1.2000
10/20/18 16:34:29	55.8000	1.2000
10/20/18 16:35:29	56.9000	1.2000
10/20/18 16:36:29	55.8000	1.2000
10/20/18 16:37:29	56.4000	1.2000
10/20/18 16:38:29	56.4000	1.2000
10/20/18 16:39:30	55.8000	1.2000
10/20/18 16:40:30	56.9000	1.2000
10/20/18 16:41:30	55.8000	1.2000
10/20/18 16:42:30	56.9000	1.2000
10/20/18 16:43:30	56.9000	1.2000
10/20/18 16:44:30	56.4000	1.2000
10/20/18 16:45:30	55.8000	1.2000
10/20/18 16:46:30	55.3000	1.2000
10/20/18 16:47:30	54.8000	1.2000
10/20/18 16:48:30	55.3000	1.2000

Figure 4.10. Temperature Spike on the Raspberry Pi 3B+

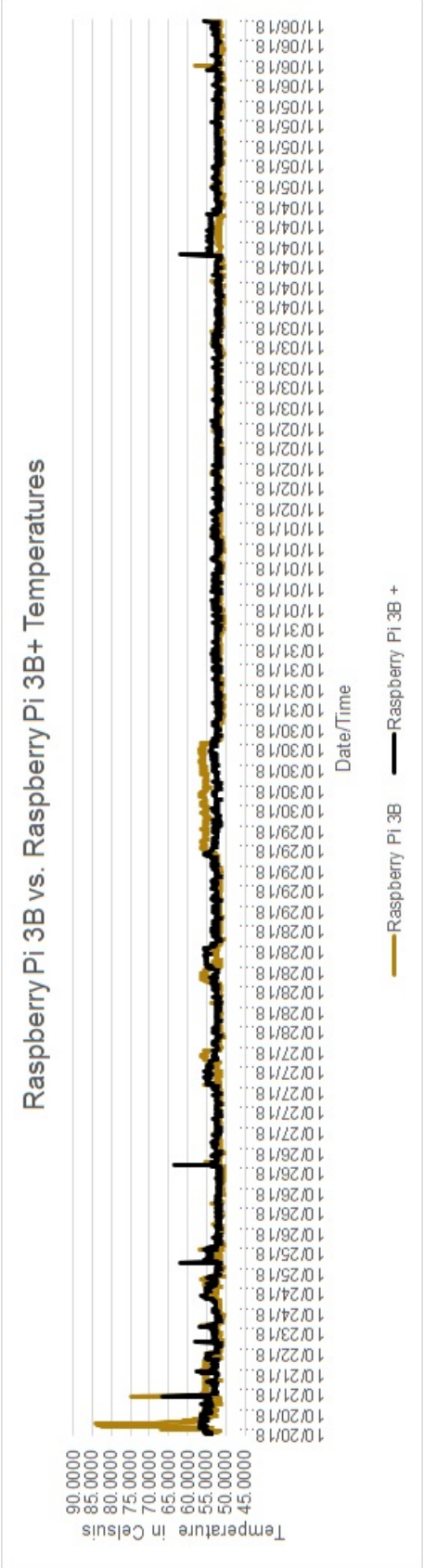


Figure 4.11. Image of a graph comparing the temperatures readings of the Raspberry Pi 3B and the Raspberry Pi 3B+

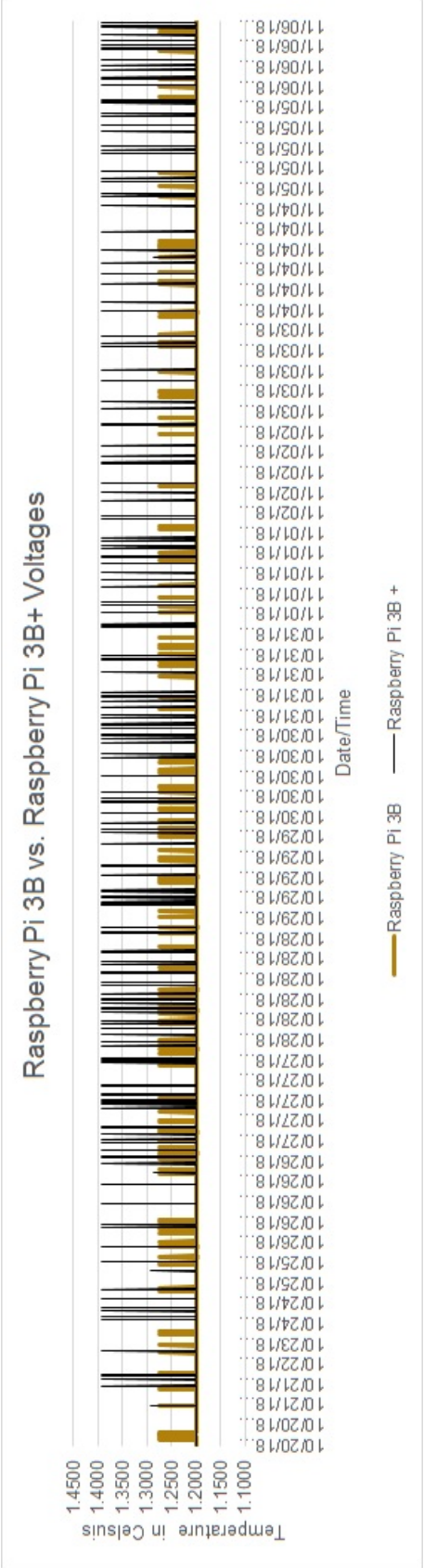


Figure 4.12. Image of a chart comparing the voltage outputs of the Raspberry Pi 3B and the Raspberry Pi 3B+



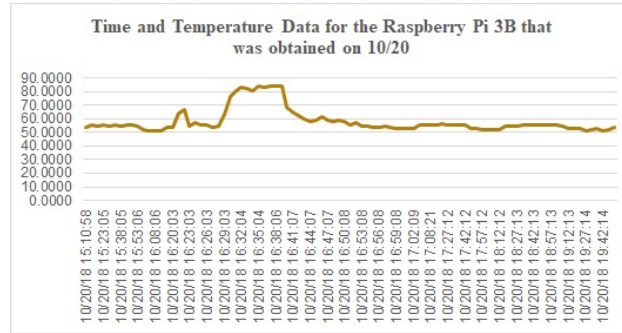


Figure 4.13. Graph showing the time and temperature of readings that were taken on 10/20 for the Raspberry Pi 3B

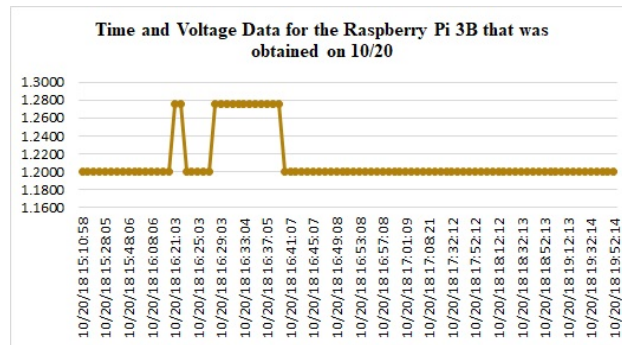


Figure 4.14. Graph showing the time and voltage of readings that were taken on 10/20 for the Raspberry Pi 3B

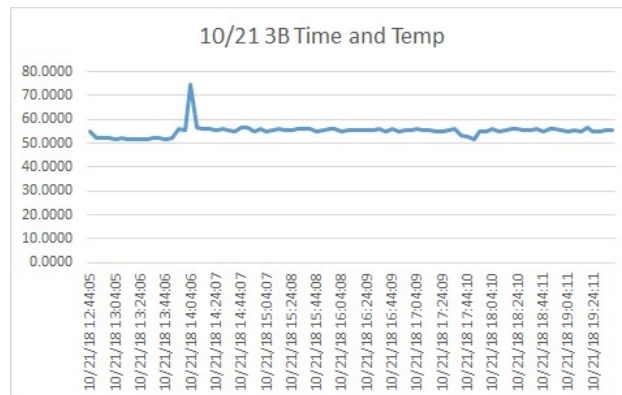


Figure 4.15. Graph showing the time and temperature of readings that were taken on 10/21 for the Raspberry Pi 3B

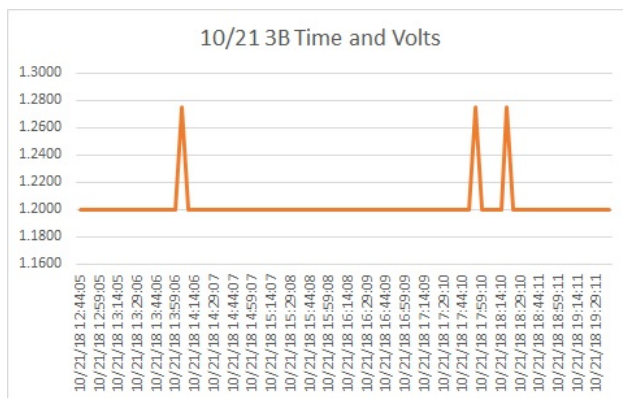


Figure 4.16. Graph showing the time and voltage of readings that were taken on 10/21 for the Raspberry Pi 3B

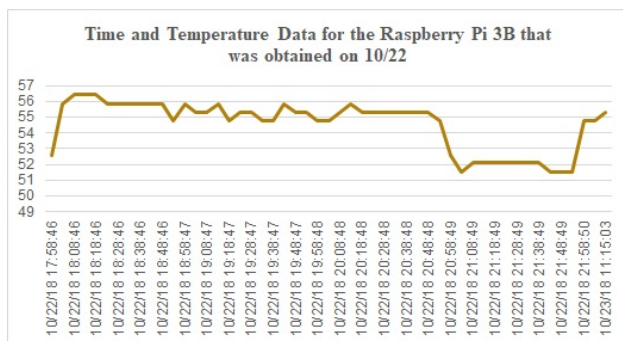


Figure 4.17. Graph showing the time and temperature of readings that were taken on 10/22 for the Raspberry Pi 3B

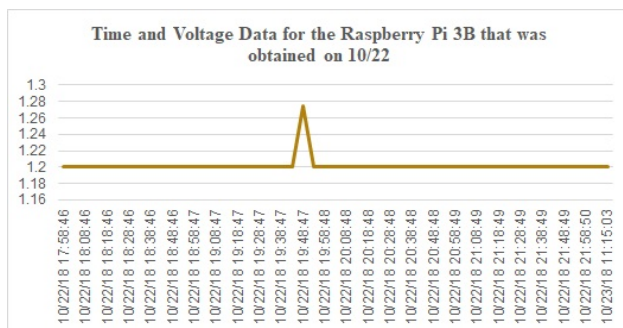


Figure 4.18. Graph showing the time and voltage of readings that were taken on 10/22 for the Raspberry Pi 3B



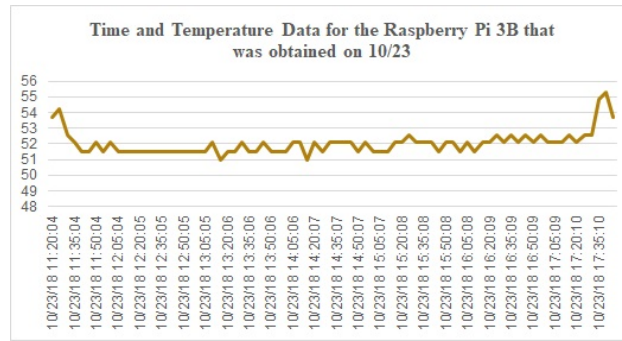


Figure 4.19. Graph showing the time and temperature of readings that were taken on 10/23 for the Raspberry Pi 3B

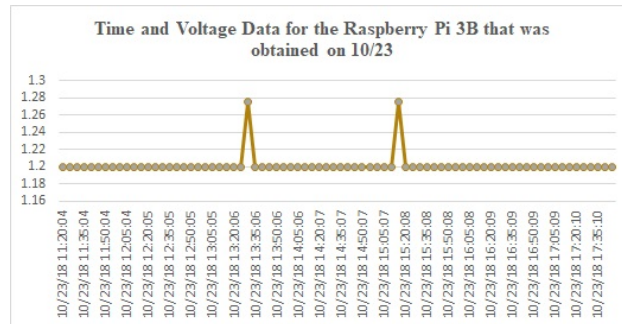


Figure 4.20. Graph showing the time and voltage of readings that were taken on 10/23 for the Raspberry Pi 3B

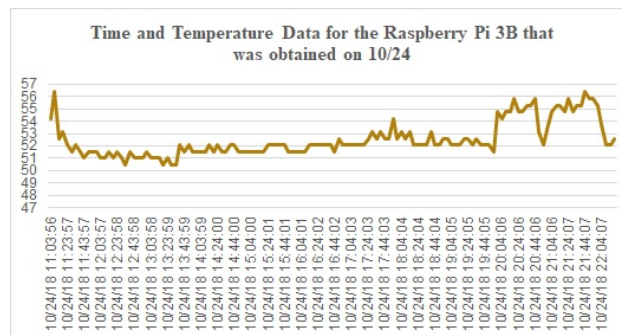


Figure 4.21. Graph showing the time and temperature of readings that were taken on 10/24 for the Raspberry Pi 3B



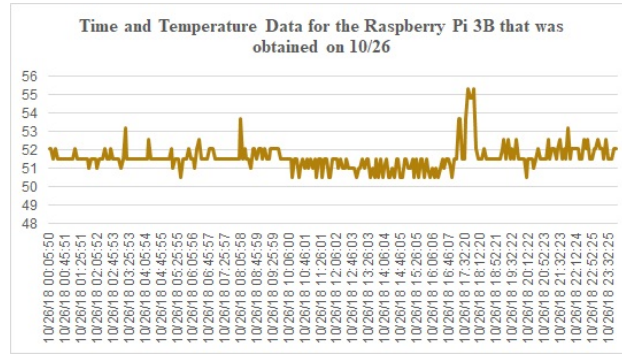


Figure 4.25. Graph showing the time and temperature of readings that were taken on 10/26 for the Raspberry Pi 3B

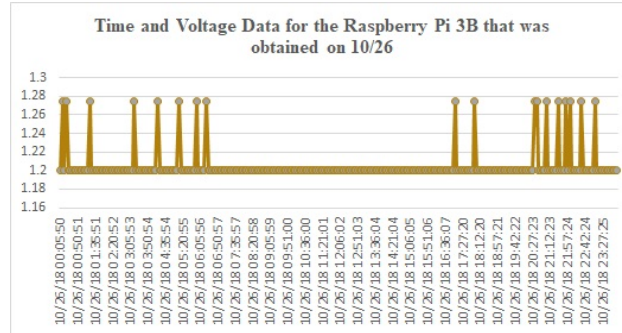


Figure 4.26. Graph showing the time and voltage of readings that were taken on 10/26 for the Raspberry Pi 3B

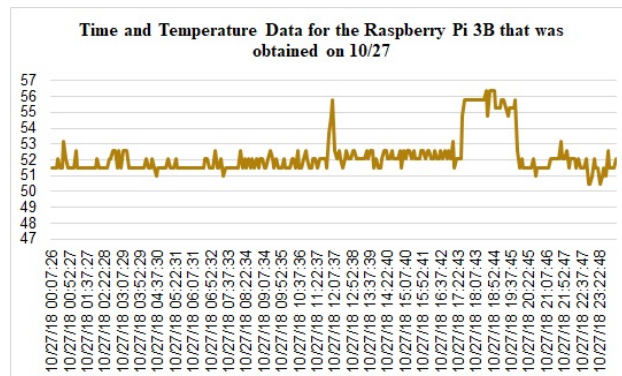


Figure 4.27. Graph showing the time and temperature of readings that were taken on 10/27 for the Raspberry Pi 3B

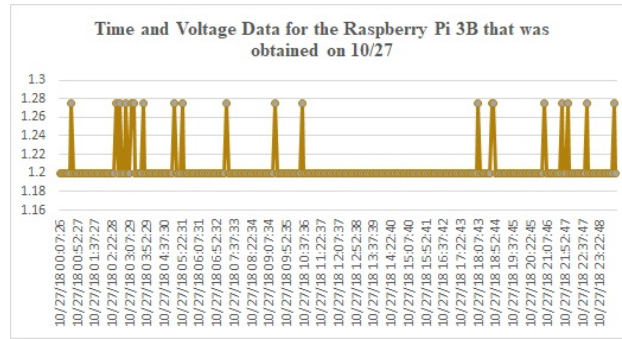


Figure 4.28. Graph showing the time and voltage of readings that were taken on 10/27 for the Raspberry Pi 3B

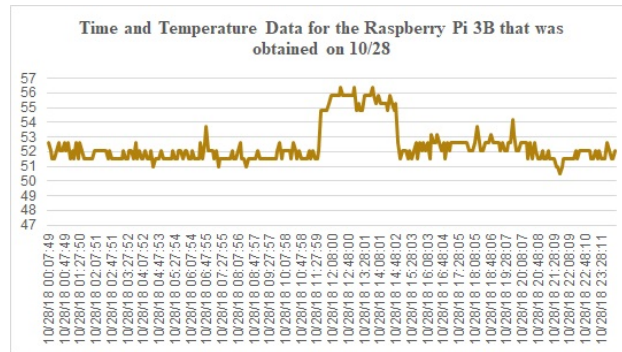


Figure 4.29. Graph showing the time and temperature of readings that were taken on 10/28 for the Raspberry Pi 3B

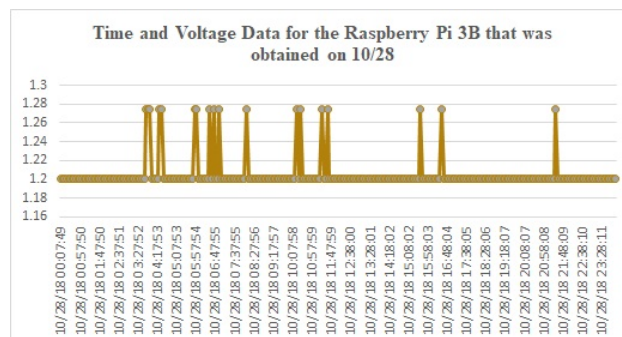


Figure 4.30. Graph showing the time and voltage of readings that were taken on 10/28 for the Raspberry Pi 3B

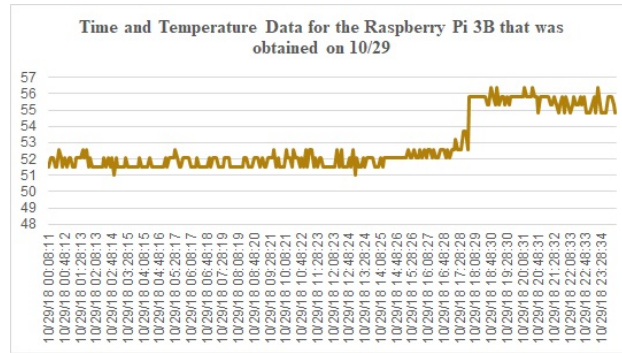


Figure 4.31. Graph showing the time and temperature of readings that were taken on 10/29 for the Raspberry Pi 3B

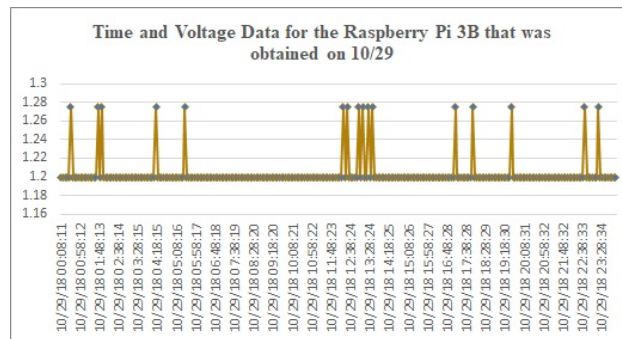


Figure 4.32. Graph showing the time and voltage of readings that were taken on 10/29 for the Raspberry Pi 3B

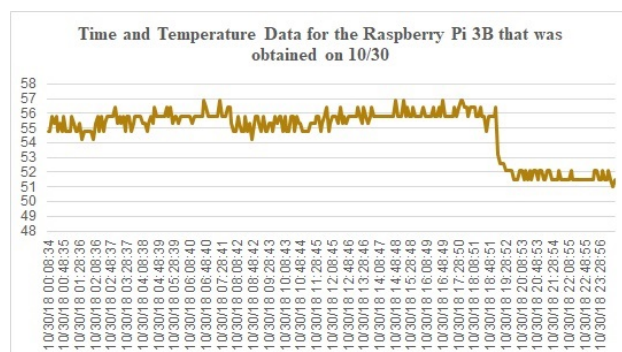


Figure 4.33. Graph showing the time and temperature of readings that were taken on 10/30 for the Raspberry Pi 3B



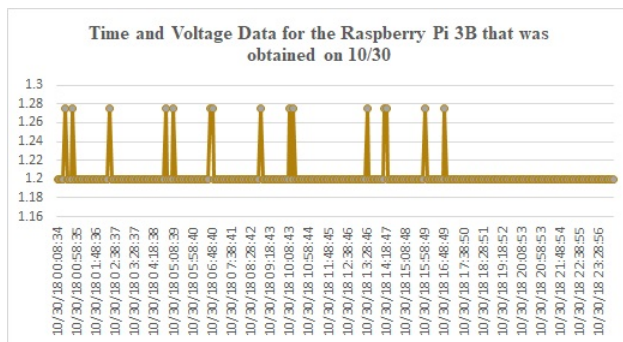


Figure 4.34. Graph showing the time and voltage of readings that were taken on 10/30 for the Raspberry Pi 3B

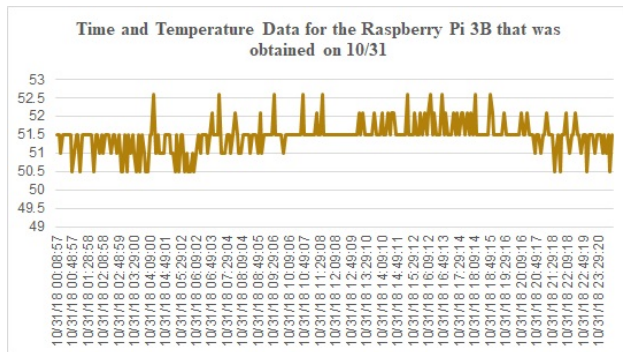


Figure 4.35. Graph showing the time and temperature of readings that were taken on 10/31 for the Raspberry Pi 3B

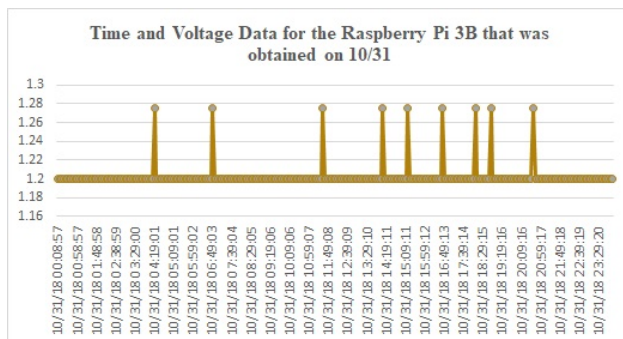


Figure 4.36. Graph showing the time and voltage of readings that were taken on 10/31 for the Raspberry Pi 3B

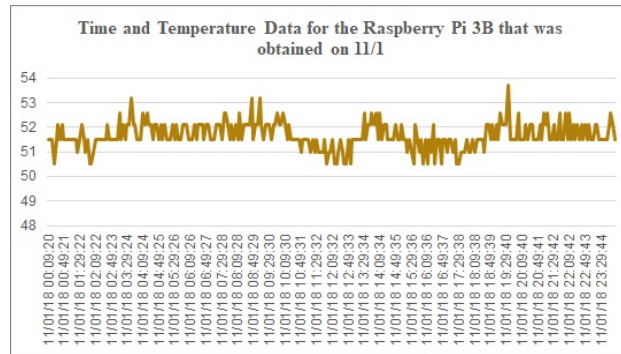


Figure 4.37. Graph showing the time and temperature of readings that were taken on 11/1 for the Raspberry Pi 3B

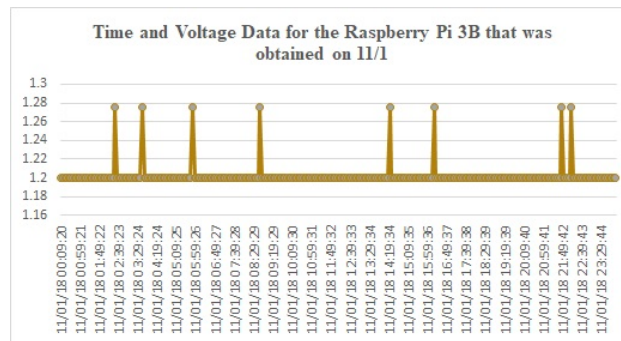


Figure 4.38. Graph showing the time and voltage of readings that were taken on 11/1 for the Raspberry Pi 3B

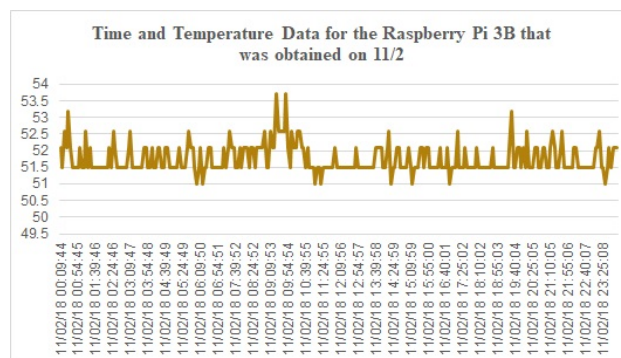


Figure 4.39. Graph showing the time and temperature of readings that were taken on 11/2 for the Raspberry Pi 3B

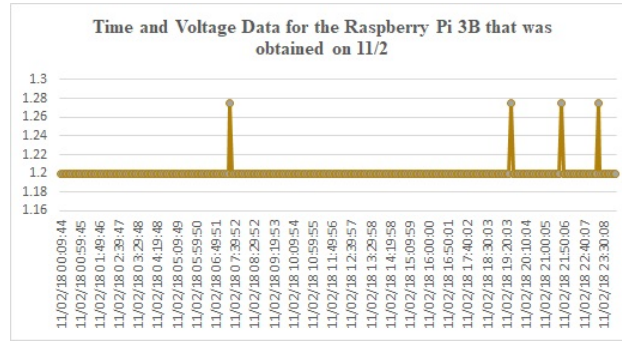


Figure 4.40. Graph showing the time and voltage of readings that were taken on 11/2 for the Raspberry Pi 3B

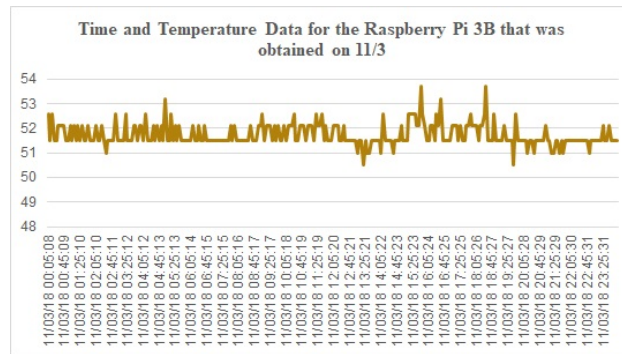


Figure 4.41. Graph showing the time and temperature of readings that were taken on 11/3 for the Raspberry Pi 3B

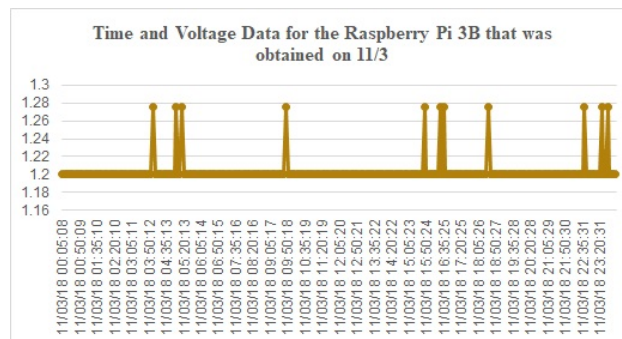


Figure 4.42. Graph showing the time and voltage of readings that were taken on 11/3 for the Raspberry Pi 3B



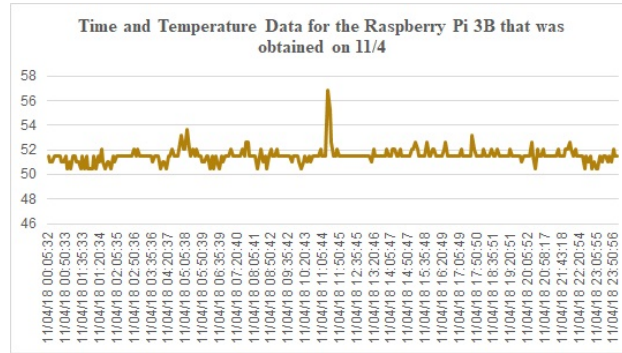


Figure 4.43. Graph showing the time and temperature of readings that were taken on 11/4 for the Raspberry Pi 3B

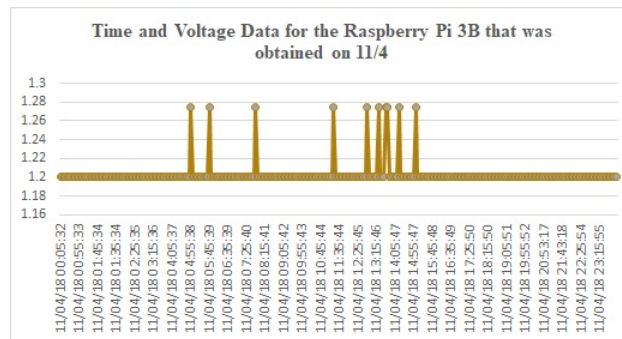


Figure 4.44. Graph showing the time and voltage of readings that were taken on 11/4 for the Raspberry Pi 3B

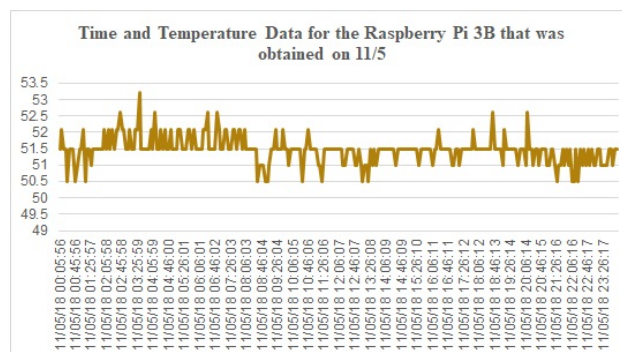


Figure 4.45. Graph showing the time and temperature of readings that were taken on 11/5 for the Raspberry Pi 3B

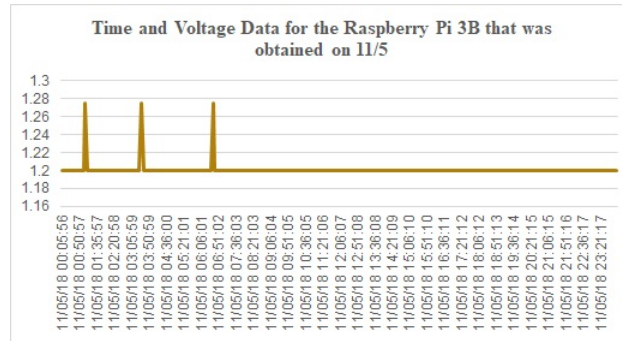


Figure 4.46. Graph showing the time and voltage of readings that were taken on 11/5 for the Raspberry Pi 3B

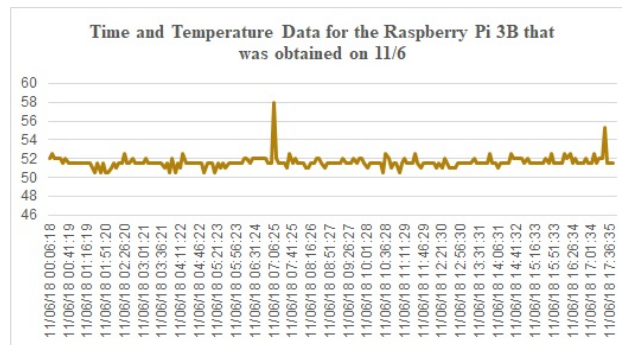


Figure 4.47. Graph showing the time and temperature of readings that were taken on 11/6 for the Raspberry Pi 3B

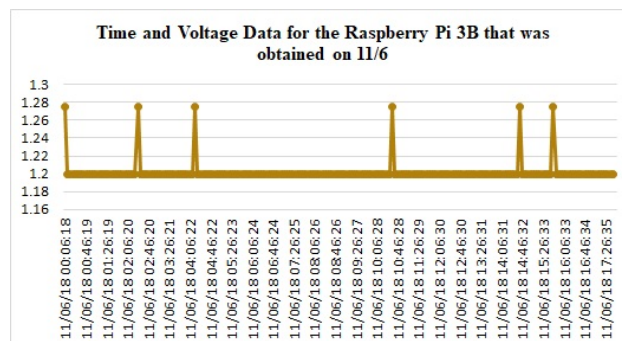


Figure 4.48. Graph showing the time and voltage of readings that were taken on 11/6 for the Raspberry Pi 3B

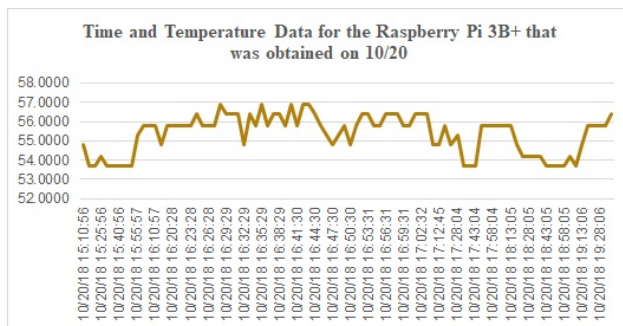


Figure 4.49. Graph showing the time and temperature of readings that were taken on 10/20 for the Raspberry Pi 3B+

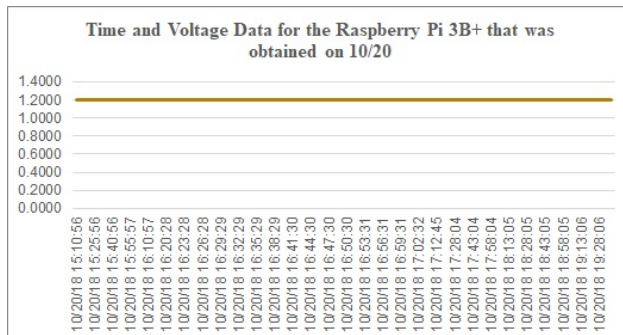


Figure 4.50. Graph showing the time and voltage of readings that were taken on 10/20 for the Raspberry Pi 3B+

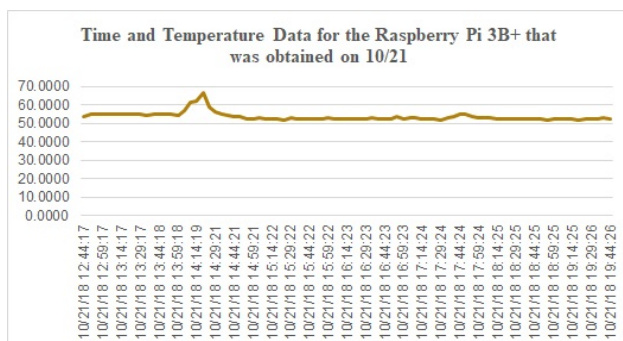


Figure 4.51. Graph showing the time and temperature of readings that were taken on 10/21 for the Raspberry Pi 3B+

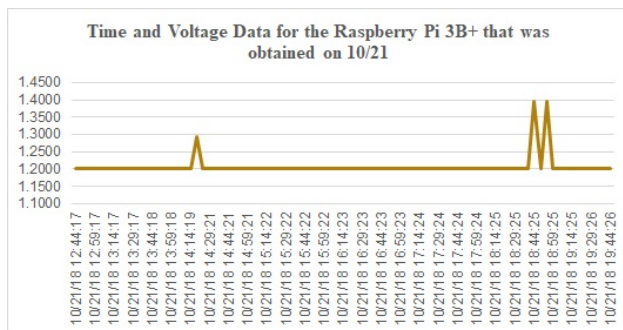


Figure 4.52. Graph showing the time and voltage of readings that were taken on 10/21 for the Raspberry Pi 3B+

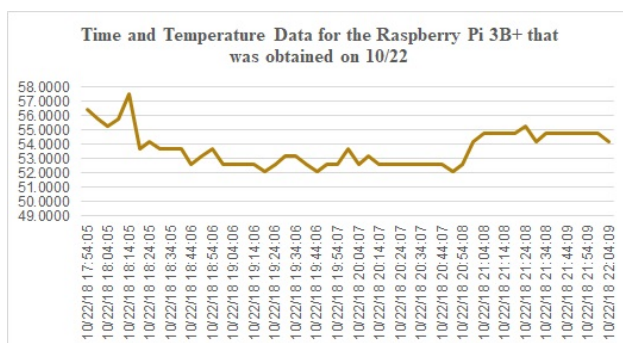


Figure 4.53. Graph showing the time and temperature of readings that were taken on 10/22 for the Raspberry Pi 3B +

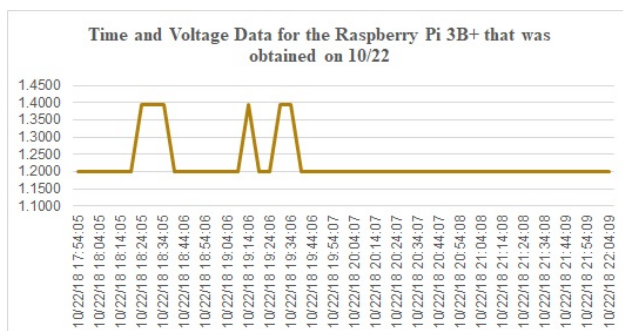


Figure 4.54. Graph showing the time and voltage of readings that were taken on 10/22 for the Raspberry Pi 3B+

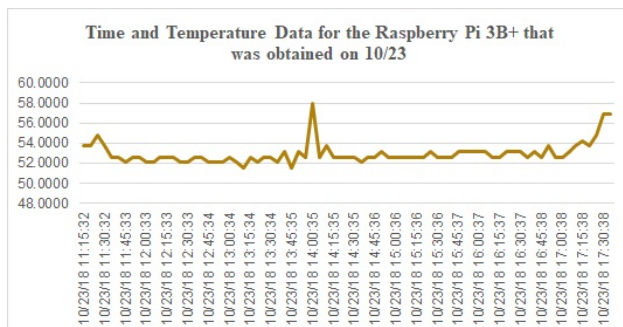


Figure 4.55. Graph showing the time and temperature of readings that were taken on 10/23 for the Raspberry Pi 3B +

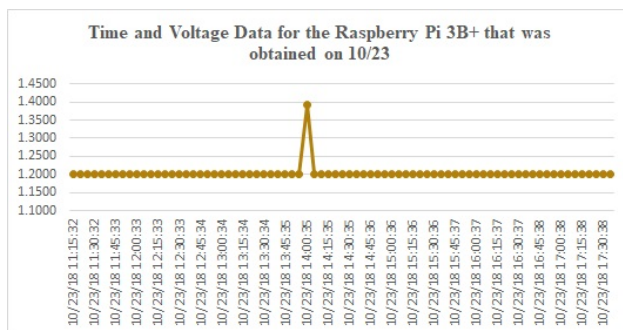


Figure 4.56. Graph showing the time and voltage of readings that were taken on 10/23 for the Raspberry Pi 3B+

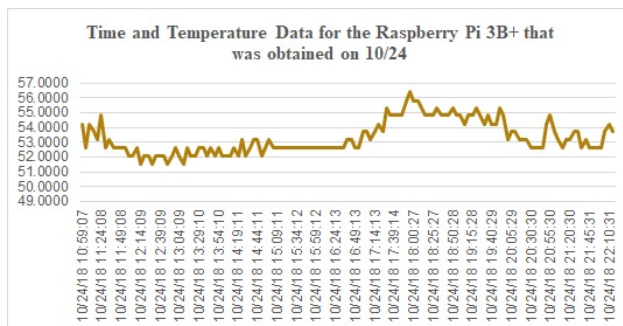


Figure 4.57. Graph showing the time and temperature of readings that were taken on 10/24 for the Raspberry Pi 3B+

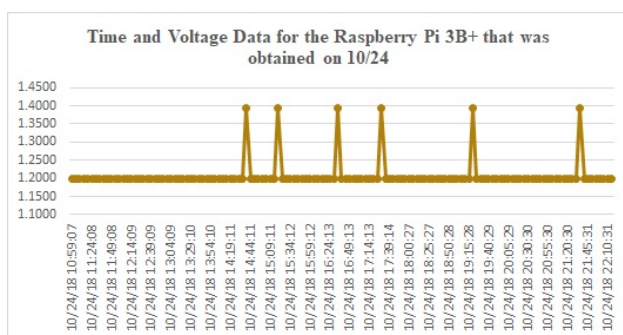


Figure 4.58. Graph showing the time and voltage of readings that were taken on 10/24 for the Raspberry Pi 3B+

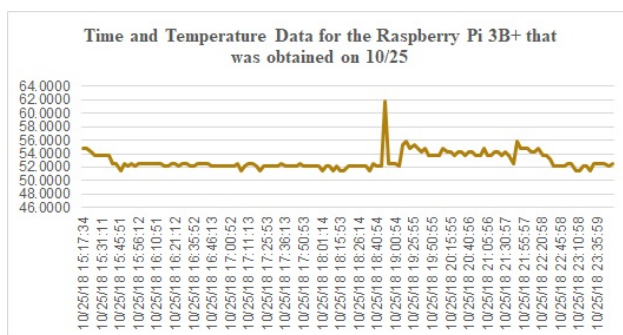


Figure 4.59. Graph showing the time and temperature of readings that were taken on 10/25 for the Raspberry Pi 3B+



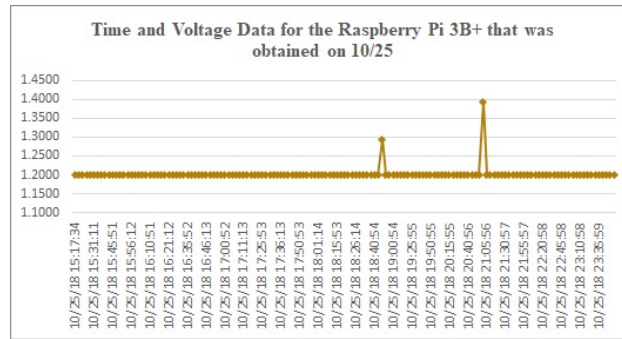


Figure 4.60. Graph showing the time and voltage of readings that were taken on 10/25 for the Raspberry Pi 3B+

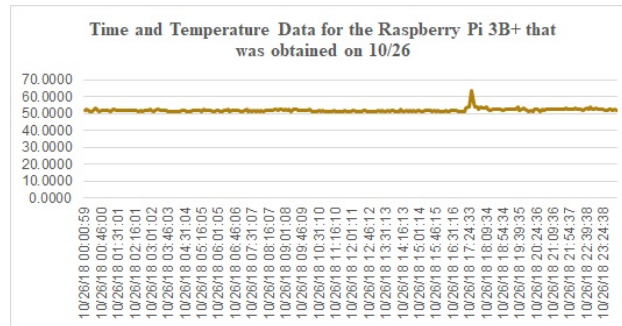


Figure 4.61. Graph showing the time and temperature of readings that were taken on 10/26 for the Raspberry Pi 3B+

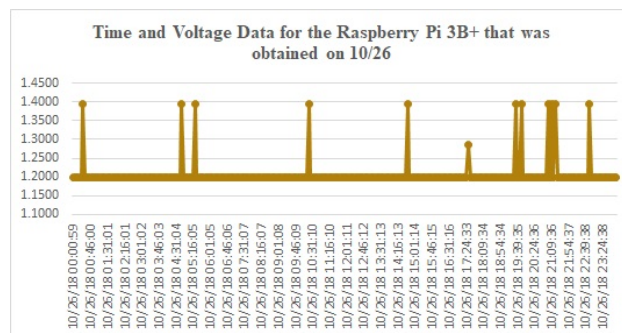


Figure 4.62. Graph showing the time and voltage of readings that were taken on 10/26 for the Raspberry Pi 3B+

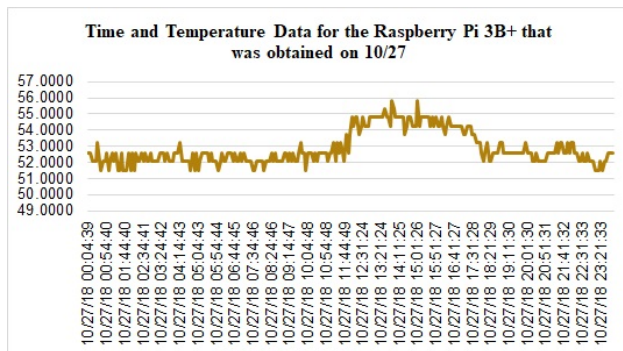


Figure 4.63. Graph showing the time and temperature of readings that were taken on 10/27 for the Raspberry Pi 3B +

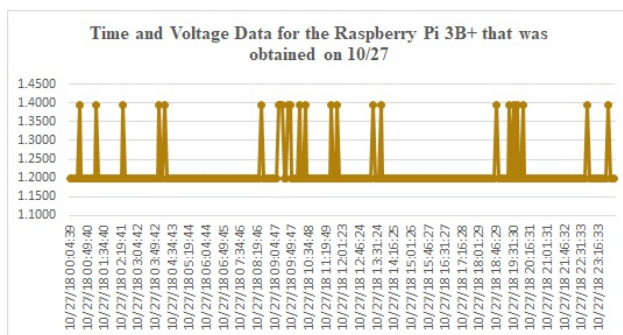


Figure 4.64. Graph showing the time and voltage of readings that were taken on 10/27 for the Raspberry Pi 3B+

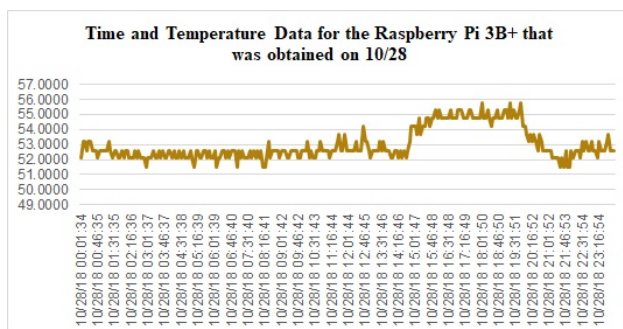


Figure 4.65. Graph showing the time and temperature of readings that were taken on 10/28 for the Raspberry Pi 3B+



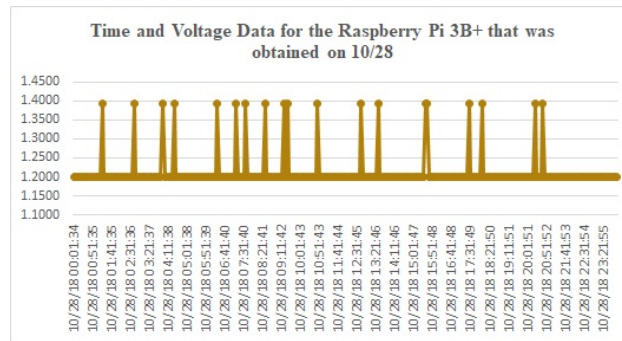


Figure 4.66. Graph showing the time and voltage of readings that were taken on 10/28 for the Raspberry Pi 3B+

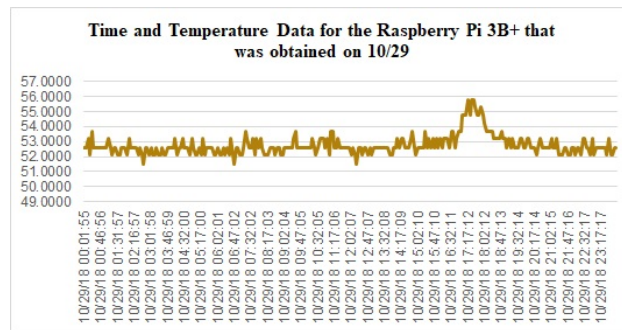


Figure 4.67. Graph showing the time and temperature of readings that were taken on 10/29 for the Raspberry Pi 3B+

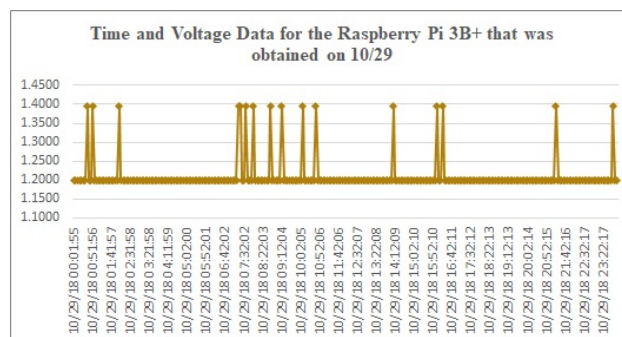


Figure 4.68. Graph showing the time and voltage of readings that were taken on 10/29 for the Raspberry Pi 3B+

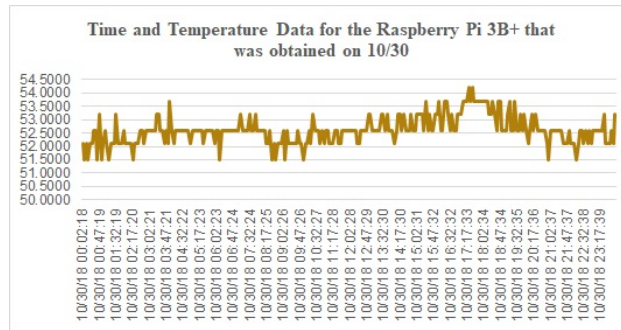


Figure 4.69. Graph showing the time and temperature of readings that were taken on 10/30 for the Raspberry Pi 3B+

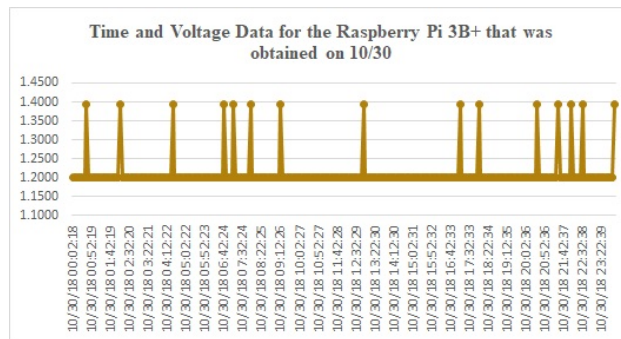


Figure 4.70. Graph showing the time and voltage of readings that were taken on 10/30 for the Raspberry Pi 3B+

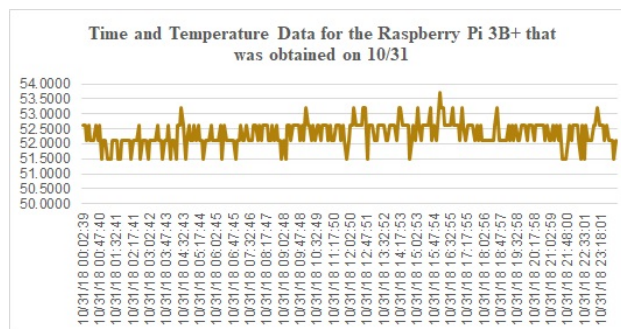


Figure 4.71. Graph showing the time and temperature of readings that were taken on 10/31 for the Raspberry Pi 3B+

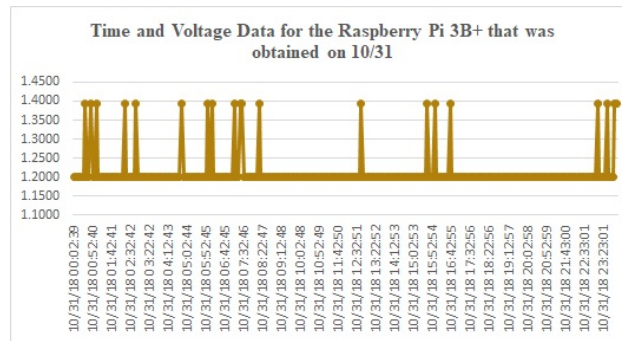


Figure 4.72. Graph showing the time and voltage of readings that were taken on 10/31 for the Raspberry Pi 3B+

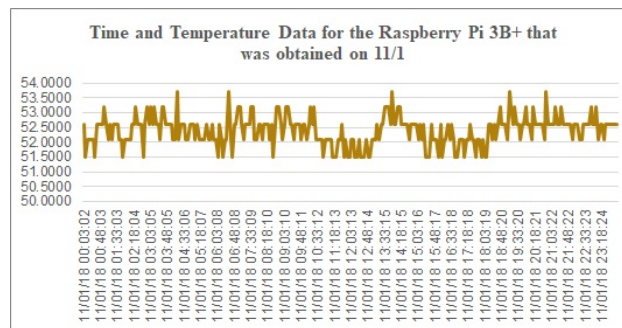


Figure 4.73. Graph showing the time and temperature of readings that were taken on 11/1 for the Raspberry Pi 3B+

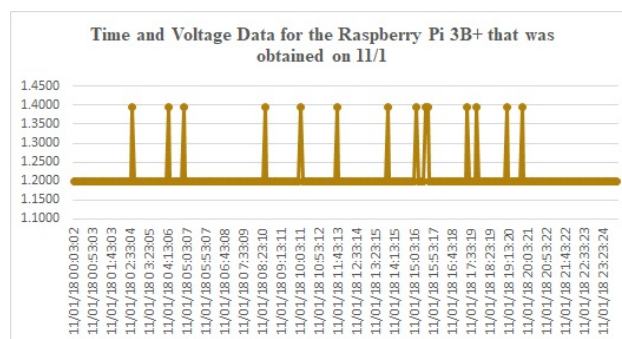


Figure 4.74. Graph showing the time and voltage of readings that were taken on 11/1 for the Raspberry Pi 3B+

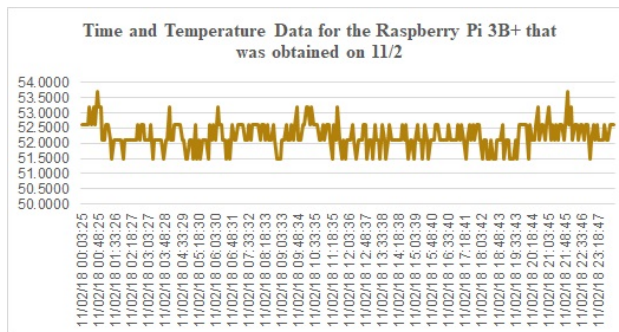


Figure 4.75. Graph showing the time and temperature of readings that were taken on 11/2 for the Raspberry Pi 3B+

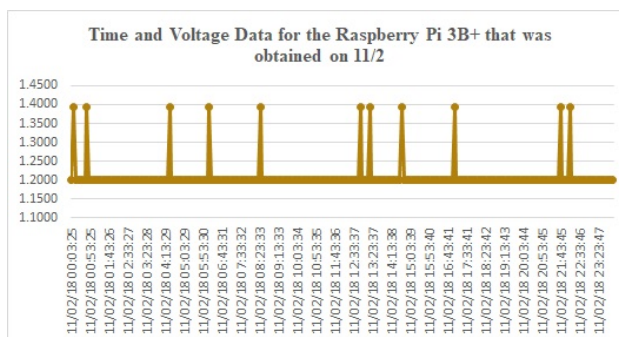


Figure 4.76. Graph showing the time and voltage of readings that were taken on 11/2 for the Raspberry Pi 3B+

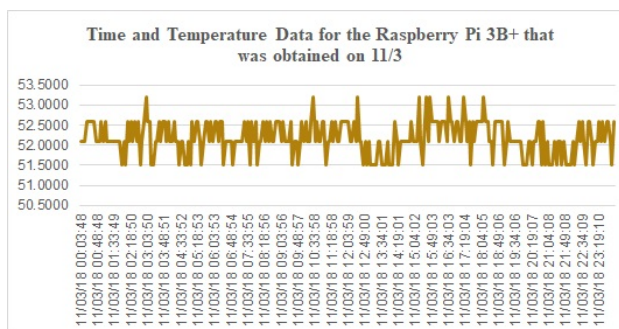


Figure 4.77. Graph showing the time and temperature of readings that were taken on 11/3 for the Raspberry Pi 3B+

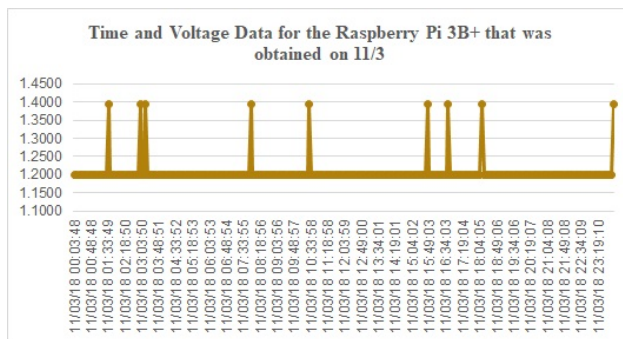


Figure 4.78. Graph showing the time and voltage of readings that were taken on 11/3 for the Raspberry Pi 3B+

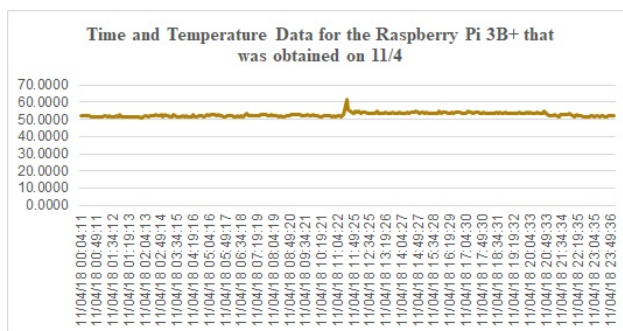


Figure 4.79. Graph showing the time and temperature of readings that were taken on 11/4 for the Raspberry Pi 3B+

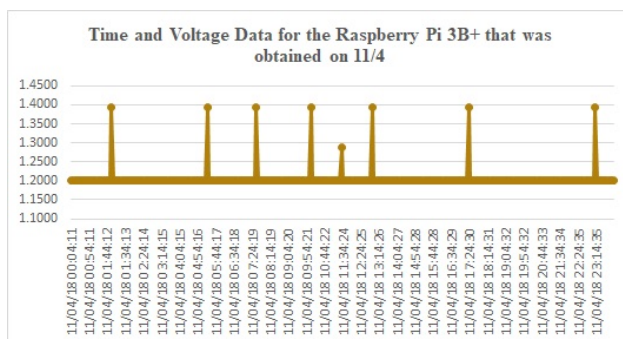


Figure 4.80. Graph showing the time and voltage of readings that were taken on 11/4 for the Raspberry Pi 3B+



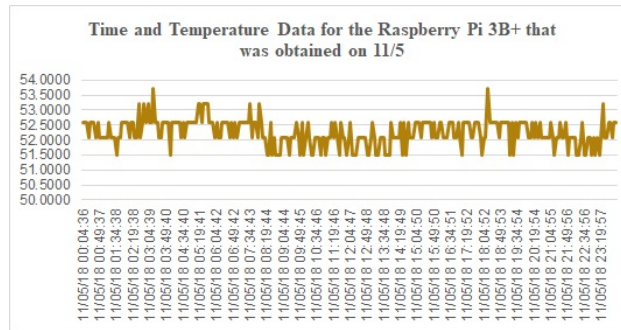


Figure 4.81. Graph showing the time and temperature of readings that were taken on 11/5 for the Raspberry Pi 3B+

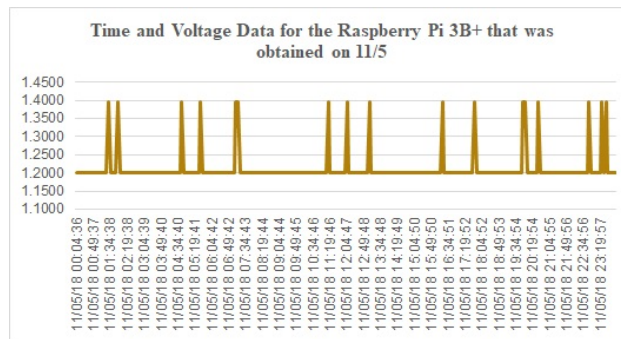


Figure 4.82. Graph showing the time and voltage of readings that were taken on 11/5 for the Raspberry Pi 3B+

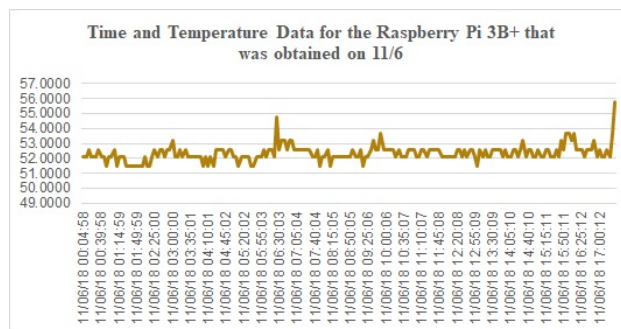


Figure 4.83. Graph showing the time and temperature of readings that were taken on 11/6 for the Raspberry Pi 3B+

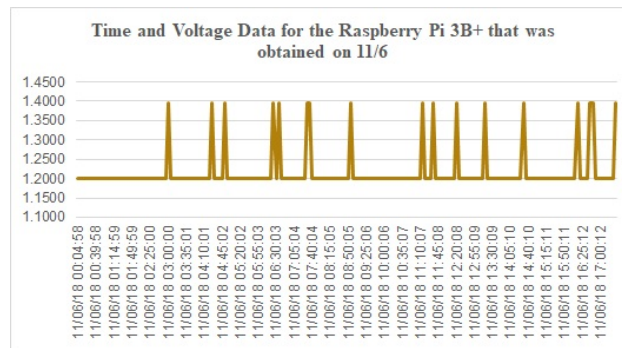


Figure 4.84. Graph showing the time and voltage of readings that were taken on 11/6 for the Raspberry Pi 3B+

### 4.3.2 FOAM Dependencies

In order for FOAM to run properly it is dependent upon three things which are the hardware that it will be running on, the blockchain that will be used to store data for the locations as well as applications and software that are being used in the background. The two blockchains that were used for this research were Tendermint and Ethereum because both had an established history with FOAM and FOAM had been previously implemented on each of them using a server setup. Additionally, there were several differences between Tendermint and Ethereum including their properties, application dependency upon as well as their setup procedures.

There are several applications as well as backend processes that FOAM needed to have installed and/or running on the hardware in order for it to work properly. There were three applications that were required to run FOAM which are Proof of Location Protocol, the Spatial Index, and the Crypto-Spatial Coordinate (CSC) Standard (Josefsson, 2017; King, 2018a, 2018b). The Spatial index is “a general purpose visual blockchain explorer” (King, 2018b, para. 1). The Spatial Index is able to serve as the public facing portion for any application that is decentralized and needs to be able to see where smart contracts are located on a map (King, 2018b). The spatial index is built on top of the purescript-web3 library (Josefsson, 2017). The Proof of Location Protocol is a decentralized protocol which permits users as well as autonomous agents to be able to privately document location data that has been authenticated when they choose as well as reveal this personal information at a later date and time that they select by providing a location claim that is fraud proof (King, 2018a, para. 2). Crypto-Spatial Coordinates (CSC) are “an open and interoperable standard for location in Ethereum smart contracts” (Josefsson, 2017, para. 2). Not only is the FOAM CSC a good “starting point for this location standard” it will also permit any smart contract to be able to make a permanent claim to a particular address on the blockchain as well as its corresponding physical location on a map (Josefsson, 2017, para. 4). CSC’s “are Ethereum smart contract addresses” that have a



corresponding physical address that is able to be verified on as well as off-chain (Josefsson, 2017, para. 11).

There are additional applications that are needed to run FOAM including Go, Go-Ethereum, Haskell, Kubernetes, react, ABI, CSC-Explorer, Purescript-eth-core, Purescript, Visualizer, SIV, MetaMask, foam.token-server, Smart contracts, cliquebait, database, servant swagger, servant client, servant client, and REST API. Each of the previously listed applications has its own functionalities as well as dependencies on other applications. The previously mentioned FOAM dependencies have been classified into three major dependency groups depending upon what their highest dependency to FOAM is and this can be seen in Figure 4.85 as a diagram. The three groups of dependencies are Haskell, Purescript, and Kubernetes ("Foamspace Corp", 2018b, fig. 6).

#### 4.3.3 Purescript Dependencies

The front-end application which is also frequently referred to as the map is the third application which is "a visual explorer that is written in Purescript" and is supported by the purescript-web3 library (King, 2018b, para. 11). The combination of the interactive deck.gl and the D3 visualization components are used for visualizing the network activity that is used by the FOAM protocol. The final application that FOAM is dependent upon is can be either Uport or Metamask and it is used to securely sign transactions as well as provide application authentication (King, 2018b). The researcher created a MetaMask account for testing purposes.

#### 4.3.4 Haskell Dependencies

The second group of FOAM dependencies are the applications that utilize Haskell and they are REST API, Servant mock, servant client, servant swagger, indexer, and DB. Haskell is the language that the FOAM REST API is written in (*Architecture of FOAM*, 2018). The first application that Haskell is dependent is REST API and "it is written in Haskell using the servant

framework” which means that the Haskell client libraries are “provided by the servant framework” (King, 2018b, para. 7). By using the servant framework REST API is able to create documentation by utilizing swagger on demand by using the type system (King, 2018b). Servant Mock is the testing server that was created by the designers of FOAM and it runs on Haskell (*FOAM Servant Mock*, 2018). Servant Client is a library that permits users to be able to derive Haskell functions that will let them query each of the endpoints on the associated web service (*FOAM Servant Client*, 2018). Servant swagger is a module that provides the ability to be able to “to describe and document RESTful APIs” (Kudasov, 2018, para. 2). The next application that FOAM is dependent upon is the Ethereum log indexer which is used for indexing logs that are “produced by spatial index contacts” therefore making them available for performance queries (King, 2018b, para. 8). The database is used to store a variety of information including coordinates for locations (King, 2018b).

#### 4.3.5 Kubernetes Dependencies

There are two primary dependencies for Kubernetes which are the previously mentioned applications that are dependent upon Haskell as well as the Ethereum clients (King, 2018b, para. 8). Kubernetes are a portable, open-source platform that is used for managing containerized workloads as well as services that are able to facilitate configurations that are both declarative and automated (*What is Kubernetes?*, 2018). Ethereum clients are the nodes that are used to parse and verify the blockchain as well as everything that is connected to it including the smart contracts (*Ethereum Homestead: Choosing a client*, 2016; King, 2018b; *What exactly is an Ethereum client and what clients are there?*, 2016).

#### 4.3.6 How the dependencies work together

The database is combined with the indexer in order to capture all of the applicable data that is being “emitted by the FOAM contracts via the Ethereum logs” (King, 2018b, para. 10).

Some of the information that is being released by the Ethereum logs includes data about the deployment of CSCs, where localizations are being performed, who is completing token transfers as well as a variety of other events and all of this information is indexed together to improve performance of queries and decrease the amount of data that is transmitted over websockets. The only remaining application that is needed to properly run this component is an Ethereum client that has a web3 API enabled on it (King, 2018b). A customized Kubernetes cluster is running in the background and it contains a combination of Geth and parity nodes that has been scaled according to the demands of the user or the customer (King, 2018b). The combination of the indexer and the database that was previously mentioned is available as “a standalone docker service” that other developers who are wanting “access to the underlying data” that is stored in the Postgres database and Elastic Search (King, 2018b, para. 10). The REST API uses the data store for the FOAM applications as well as acting like an open API for additional applications that want access to the data that was created by FOAM (King, 2018b). “It is written in Haskell” and uses the servant framework which means that current API documentation as well as client libraries are able to be created from the type system when they are initially requested (King, 2018b, para. 11). The visual explorer is the customer facing portion of the application that is used to display the data that is obtained from the REST API in a useful way while utilizing D3.Js and deck.gl (King, 2018b). Additionally the visualizer is able to permit users “to interact with FOAM contracts via MetaMask or Uport” (King, 2018b, para. 12). The visualizer is written in Purescript programming language (King, 2018b). The ability of this design to permit users to be able to not only interact with the blockchain itself but also deploy smart contracts that contain geospatial parameters from a web browser is not only the primary contribution but also an incredible accomplishment (King, 2018b). The architecture of the spatial index creates “a smooth loop of events” that moves from the web-application, to the blockchain, to the indexer and returns back to the application (King, 2018b, para. 13).

#### 4.3.7 FOAM Map

CSCs which act as spatially specific smart contracts “are displayed directly in the application” and users are able to utilize filters in order to display the CSCs that they want to see (King, 2018b, para. 14). A CSC is able to be directly deployed as a smart contract through “the Spatial Index by using MetaMask” (King, 2018b, para. 15). FOAM checks “when a new smart contract is deployed, and automatically” “visualizes “it in the spatial index” (King, 2018b, para. 16).

#### 4.3.8 FOAM Technical Architecture

The technical architecture of FOAM is best thought of as a series of layers consisting of the FOAM Coordinate System, the FOAM Verification, and the Spatial Index. The Figure 4.85 is an image of the layers of the FOAM architecture. The architectural base is the FOAM Coordinate System because it acts like a registry for the CSCs that will be used to record the geospatial data on the blockchain (*Introducing the FOAM Protocol*, 2017). As it was previously stated a “CSC is an on-chain verifiable Ethereum smart contract address” that has a matching geohash address that has a physical location and can be verified online as well as offline (*Introducing the FOAM Protocol*, 2017, para. 11). The ability to be able to combine physical addresses that are crowdsourced with publicly owned and accessible blockchain smart contracts that can not only hold but receive the value of its current balance (*Introducing the FOAM Protocol*, 2017). The next layer of the technical architecture is the FOAM verification layer which is used to incentive users to create a reputation by utilizing the CSCs (*Introducing the FOAM Protocol*, 2017). The users are able to stake either “reputation or utility tokens to” be able to vouch for the accuracy of a CSC as well as verify its location (*Introducing the FOAM Protocol*, 2017, para. 12). The third and highest level is the Spatial Index which is being used “as a real time visualization tool” (*Introducing the FOAM Protocol*, 2017, para. 13). It is “a cross between Google Maps and a Bloomberg Terminal” and it has “a visual interface to the FOAM Protocol that will” permit users

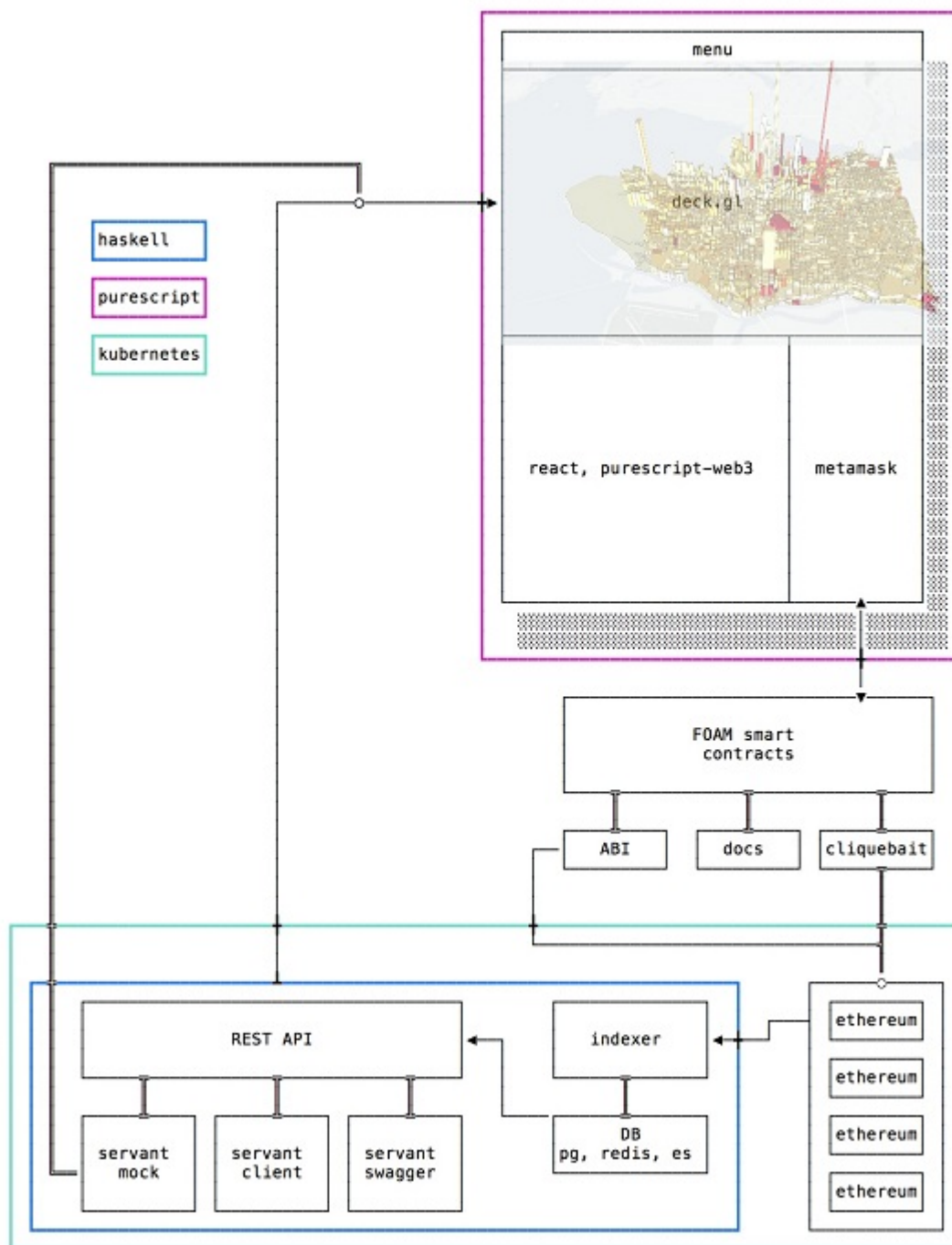


Figure 4.85. FOAM Architecture

(Architecture of FOAM, 2018; King, 2018b, fig. 1)

to be able to use the coordinate system to understand, engage as well as use their data (*FOAM Spatial Index*, 2018; *Introducing the FOAM Protocol*, 2017, para. 13). By introducing a economic layer that will be able to connect the coordinate systems, geospatial data as well as hard assets together the Spatial index will permit users to be able to track and view analytics about all of the applications that are utilizing the FOAM Protocol (*Introducing the FOAM Protocol*, 2017).



*Figure 4.86. Technical Architecture Layers*  
(*Introducing the FOAM Protocol*, 2017, fig. 1)

#### 4.3.9 How does the FOAM Map Work

There are several additional things that are needed in order to make the FOAM Proof of Location work including a Byzantine fault tolerant (BFT) clock, Zone anchors, beacons, Zone authorities, verifiers, and a location customer or user. Byzantine fault tolerant (BFT) clocks are the clocks that are used to keep the time between the zones and the zone anchors consistent. Zone anchors are radio transceivers that provide time synchronization (King, 2018a). “Four or more Zone Anchors form a Zone, the quorum that maintains clock sync for a given region” (King, 2018a, para. 10). “Zone authorities are essentially internet gateways” (Goblirsch, 2018, para. 6). There are eleven steps that are needed to obtain a valid proof of location and these steps are listed below.

1. To become a FOAM Proof of location service provider a radio beacon pledges foam tokens as a safety deposit to initiate its participation in the network as a zone anchor.
2. The radio beacon searches for three other beacons or zone anchors that are nearby and once it finds them then messages are exchanged between the zone anchors with the goal of synchronizing their clocks.
3. After their clocks are able to be successfully synchronized and consensus is achieved about time then a zone is created.
4. After the zone is established then each of the zone anchors receives a reward in the form of FOAM Tokens.
5. A user which is also known as a location customer enters the zone and submits a request for a presence claim.
6. Each of the zone authorities receives a fee payment from the customer in the form of FOAM tokens.

7. Each of the zone authorities calculates its distance from the customer by measuring the time it takes to receive a message.
8. After each of the zone authorities has been able to calculate how far it is from the customer then the information is stored on local blockchain to help them in reaching consensus to verify the information.
9. The presence claim is then sent to verifiers who will compare the data that was just received against the data that is located in other zones.
10. After the data is verified then it will be printed/added to the Ethereum blockchain and made available to the public.
11. The location customer receives its verified location through a decentralized application.

#### 4.3.10 Successes

This section consists of the successes that were encountered while setting up the Raspberry Pi 3B and the Raspberry Pi 3B+, the installation as well as setting up the additional software, compiling, and syncing the Ethereum and Tendermint blockchains.

##### 4.3.10.1 Ethereum Raspberry Pi 3B

The first application that needed to be downloaded, compiled, and setup on the Raspberry Pi 3B was Go which was successfully completed. The next application that also needed to be downloaded, compiled, and setup was the Ethereum blockchain. After setting up the environments for Go as well as Ethereum they were compiled together to make Go-Etheruem which is also known as Geth. Once Go-Ethereum had been compiled then the syncing of the Ethereum Blockchain began so that it would be completely caught up. The reason that the Ethereum Blockchain had to be completely synced was FOAM needed to have access to a



ERC-20 chain. CSC-Explorer, FOAM, Purescript, and the Purescript-eth-core were all able to be successfully downloaded to the Raspberry Pi 3B+.

#### 4.3.10.2 Ethereum Raspberry Pi 3B+

After completing the setup and installing the most recent updates on the Raspberry Pi 3B+ Go was the first application that was downloaded, compiled, and setup. The next application to be downloaded and setup was the Ethereum blockchain. Ethereum and Go were compiled together to create Go-Ethereum which is also known as Geth. Once Go-Ethereum had completely compiled the Ethereum blockchain was permitted to completely sync so that it had obtained the most recent blocks in its blockchain. The final successes that was observed in completing the setup of the Raspberry Pi 3B+ was completing the download of CSC-Explorer, FOAM, Purescript, and the Purescript-eth-core.

#### 4.3.10.3 Tendermint Raspberry Pi 3B

Once the Raspberry Pi 3B was setup, and the most recent updates were downloaded to it then Go was downloaded, compiled and it was setup. The second success that was seen in this iteration of installations was the successful download of Tendermint.

#### 4.3.10.4 Tendermint Raspberry Pi 3B+

After completing the setup as well as downloading the most recent updates to the Raspberry Pi 3B+ Go was then downloaded, compiled, and setup. The second and final success that was seen during the setup of the Raspberry Pi 3B+ was completing the download of Tendermint.

#### 4.3.10.5 Ethereum Virtual Raspberry Pi

Once the virtual environment for the Raspberry Pi was completely setup and it had received the most recent updates then Go was downloaded, compiled, and setup in the Virtual Raspberry Pi. Ethereum successfully downloaded to the Virtual Raspberry Pi as well as FOAM, Haskell-Opaleye, and the Foam-Token-Server.

#### 4.3.10.6 Tendermint Virtual Raspberry Pi

Once the virtual environment for the Raspberry Pi was completely setup and it had received the most recent updates then Go was downloaded, compiled, and the setup was completed for the Go environment on the Virtual Raspberry Pi. Tendermint was able to be successfully downloaded to the Virtual Raspberry Pi.

#### 4.3.10.7 Tendermint Virtual Server

After completing the initial setup of the Ubuntu Server it was completely updated then Go was downloaded, compiled, and its environment was setup. The second step that was completed to create the Tendermint Virtual Server was to download, compile, and setup the Tendermint Blockchain. The Tendermint blockchain like the Ethereum Blockchain on the Raspberry Pi 3B and the Raspberry Pi 3B+ was permitted to sync so that it would have knowledge of the most recent blocks on the Tendermint blockchain.

### 4.3.11 Lessons Learned

This section contains the lessons that were learned while setting up the Raspberry Pi 3B and the Raspberry Pi 3B+, the installation of software as well as setting up, compiling, and syncing the Ethereum and Tendermint blockchain.

#### 4.3.11.1 Ethereum Raspberry Pi 3B

Despite that fact that the downloads for the CSC-Explorer, FOAM, Purescript, and the Purescript-eth-core were able to be completed on the Raspberry Pi 3B there were additional issues that prevented the applications from being able to be compiled and setup. The primary reason that the CSC-Explorer, FOAM, and Purescript were not able to compile correctly is they were dependent upon several additional applications that were either not supported on the Raspberry Pi 3B or could not be downloaded. Additionally, there were several interdependencies between the CSC-Explorer, FOAM, and Purescript applications, which meant that if one could

not compile correctly then there would be several issues with the other two applications. There were two issues that prevented the Purescript-Eth-Core from being compiled as well as setup and the first issue was that it was like the other three applications in the fact that it was dependent upon the other two applications in order for it to work properly. The other reason that the Purescript-Eth-Core would not compile is that it was not supported on the Raspbian Operating System and the earliest operating system that it was supported on was the Ubuntu Server Trusty version (14.04) (Kejace, 2018). Lastly, the Raspberry Pi 3B experienced several temperature and voltage spikes due to the processor working extremely hard while it was syncing the Ethereum Blockchain.

#### 4.3.11.2 Ethereum Raspberry Pi 3B+

Notwithstanding the fact that the Raspberry Pi 3B+ has a better processor than the Raspberry Pi 3B it was still plagued by the same issues as the Raspberry Pi 3B. The first issue that was encountered on the Raspberry Pi 3B+ was the fact that FOAM would not compile even though its software dependencies were downloaded to the device. The second issue that was encountered on the Raspberry Pi 3B+ was due to the processor continuously working to download the Ethereum blockchain which resulted in occasional temperature and voltage spikes on the device. Lastly, the processor on the Raspberry Pi 3B+ also started to experience a slight amount of lag due to the work it was doing to continuously sync the Ethereum blockchain.

#### 4.3.11.3 Tendermint Raspberry Pi 3B

Even though Go was able to be compiled on the Raspberry Pi 3B and Tendermint blockchain had been able to be downloaded to it the fact remains that it did not have a fast enough processor to be able to run the Tendermint Blockchain. The minimum processor speed that was needed to sync the Tendermint Blockchain is 1.4 GHz and the Raspberry Pi 3B had a 1.2 GHz processor.

#### 4.3.11.4 Tendermint Raspberry Pi 3B+

In spite of successfully downloading Tendermint, completing the setting up of the Go environment, and having a 1.4 GHz processor the Tendermint Blockchain was not able to be compiled on the the Raspberry Pi 3B+. The researcher has completed troubleshooting and believes that there are three possible causes for Tendermint Blockchain not compiling properly on the Raspberry Pi 3B+. The first and most probable cause for Tendermint not compiling on the Raspberry 3B+ is that Tendermint required additional applications that were not supported on the Raspbian Operating System. The second reason that Tendermint blockchain would not compile is that there were insufficient system resources available when the compile was attempted. The final possible reason that Tendermint was not able to be compiled on the Raspberry Pi 3B+ is that it needed to have a processor that was slightly higher than 1.4 GHz.

#### 4.3.11.5 Ethereum Virtual Raspberry Pi

Despite the fact that the Virtual Raspberry Pi was running on a laptop and it had been created with specifications that were either identical to a Raspberry Pi 3B or better there were two issues which serves as lessons learned. The first issue is that despite being able to download, compile, and setup Go as well as being able to download Ethereum the two applications would not compile together to create Go-Ethereum. It is believed that the issue of not getting Go and Ethereum to compile together was caused by one or a combination of two issues. The first issue that is theorized to have caused Go and Ethereum to not compile together correctly is that the Debian operating system that was being used for testing would not support one or both of the applications. The possible reason that Debian would not support either Go or Ethereum is because one or both of them were too old or the applications were just too new for the operating system and they were not yet supported. The second issue that was encountered on the Virtual Raspberry Pi was that the Go environment was either being reset or Go was being completely removed or uninstalled from the system when it was rebooted. Based upon the troubleshooting that has been completed there are three possible reasons that Go would be uninstalled or removed

from the system as well as having the Go environment be reset. The reasons are that Go was uninstalling itself, the operating system was being rolled back for some reason, or the Raspbian Operating system determined that Go was a threat and automatically removed it or reset it's environment. This is an identical issue that was encountered when attempting to setup the Tendermint Virtual Raspberry Pi and the researcher believes that they are related.

#### 4.3.11.6 Tendermint Virtual Raspberry Pi

Despite the fact that the Virtual Raspberry Pi was created using specifications that were either equal to or better than a Raspberry Pi 3B it would not compile the Tendermint Blockchain that had been downloaded. Since the device as well as the operating system had the most recent updates and met the minimum specifications to compile the Tendermint Blockchain the installation history was reviewed and prior to rebooting the system everything looked fine and normal. Once the operating system was rebooted it was discovered that entire setup for Go was reset and needed to be reset up again which it was and another reboot was required and it was discovered that the same issue was encountered again. Tendermint as well as Go were uninstalled, removed from the virtual Raspberry Pi, the server was rebooted, the process was attempted again, and the same error was received. After completing the troubleshooting, no determination has been made as to why Tendermint would not compile on the Virtual Raspberry Pi since it was running an AMD 64bit version of the software (*Debian – Details of package tendermint in sid*, 2018). Lastly this is the same issue that was encountered on the Ethereum Virtual Raspberry Pi and the researcher feels that it is an issue with the Debian Operating System having an issue with the version of Go (1.10) that was used.

#### 4.3.11.7 Tendermint Virtual Server

Despite having completed the setup of the Ubuntu Server, including downloading and installing its most recent updates as well as being able to completely set up the Go environment, compiling and initiating the synchronization of the Tendermint Blockchain there was one issue

that could not be resolved. Tendermint was not able to completely sync all of the blocks in its blockchain which prevented moving forward with compiling the rest of the applications that were need to run the Location Protocol for FOAM.

#### 4.4 Procedures

This section will contain the procedures that were used to setup the Raspberry Pi 3B, the Raspberry Pi 3B+, the Virtual Raspberry Pi's as well as the installation of the additional software and the steps that were taken to compile the Ethereum and Tendermint blockchains on each of the physical and virtual devices as well as the steps that were taken in the installation and setup of the Tendermint Virtual Server.

##### 4.4.1 Ethereum Raspberry Pi 3B and Raspberry Pi 3B+

NOOBS was downloaded to a laptop, unpacked, copied to two different 32 GB MicroSD Cards. One MicroSD Card was inserted into a Raspberry Pi 3B and the other MicroSD Card was inserted into the Raspberry Pi 3B+. After the MicroSD cards were inserted into their respective Raspberry Pi's then each of the Raspberry Pi's were powered on, Raspbian Operating System (OS) was installed, standard setup was completed, and pending updates were installed. The next step to be completed was formatting USB drives to support the FAT32 file system on a Windows computer. Then each of the USB drives was plugged into their respective Raspberry Pi's and reformatted so that they supported the ext4 file system. After the USB drives were reformatted so that they could be read, as well as written to by the Raspberry Pi's; then, they were mounted as an external storage location for the Ethereum Data. Then next step that was completed on both Raspberry Pi's was to download, install, and setup Go. Then Ethereum was downloaded as well as installed and Geth was compiled. After Ethereum was installed on the Raspberry Pi 3B as well as the Raspberry Pi 3B+ each of them was permitted to mine the Ethereum Blockchain until it had downloaded all the blocks in the Blockchain. Haskell, and Node.js were downloaded and installed

on the Raspberry Pi 3B as well as the 3B+. Several errors were encountered while attempting to install csc-explorer, FOAM, foam.developer, purescript, and purescript-eth-core applications after downloading each of them to the Raspberry Pi 3B and Raspberry Pi 3B+ from GitHub.

#### 4.4.2 Tendermint Raspberry Pi 3B and Raspberry Pi 3B+

Since the FOAM Protocol was not able to successfully run on the Raspberry Pi's with Ethereum it was decided that the Raspbian OS would be reinstalled and Tendermint would be installed instead of Ethereum. The initial setup of the two Raspberry Pi's that was mentioned previously was completed again using an additional two 32 GB MicroSD cards as well two more 128 GB USB drives from the initial setup to the installation of GO. After completing the installation of Go on the Raspberry Pi's, Tendermint was downloaded from GitHub and when it was compiled there were several errors that displayed indicating that it needed additional software packages to be able to run. While attempting to download the additional software packages it was discovered that most of the applications that were required to run Tendermint were either not available on GitHub or were not supported on the operating system of the Raspberry Pi's. It was determined that running Tendermint on a Raspberry Pi 3B or 3B+ was not a feasible option.

#### 4.4.3 Ethereum Virtual Raspberry Pi

A Virtual Raspberry Pi was created using a 32 GB virtual hard drive and one GB of ram was allocated to the virtual drive. The Raspberry Pi Desktop which is also known as Debian had been previously downloaded to the laptop that contained the virtual hard drive, it was installed, the standard setup was completed, updates were downloaded, and installed. After the Virtual Raspberry Pi was setup and updated then Go was downloaded and it was also setup. After completing the setup and installation of Go then PostgreSQL database, Haskell, Haskell-Opaleye, and the foam-token-server were downloaded to the virtual Ethereum Raspberry Pi from GitHub. The last task that needed to be completed in the setup of the virtual Raspberry Pi was to download

and install Ethereum but unfortunately after downloading Ethereum an attempt to compile Go and Ethereum together was made and several errors were encountered indicating that there was an issue with the operating system. The minimum specifications that were needed to run Ethereum were reviewed and no determination could be made as to why it refused to compile.

#### 4.4.4 Tendermint Virtual Raspberry Pi

Due to the issues that were encountered when attempting to install Ethereum on the Virtual Raspberry Pi it was decided that a second Virtual Raspberry Pi be created using Debian again as the operating system. The second Virtual Raspberry Pi was created using the same specifications as the Ethereum Virtual Raspberry Pi including a 32 GB virtual hard drive and it was also allocated one GB of RAM. Additionally, the setup that was completed for the Virtual Tendermint Raspberry Pi environment was identical to the setup of the Ethereum Virtual Raspberry Pi environment including the standard setup being completed as well as the installation of the most recent operating system updates. While research was being completed on Tendermint it was discovered that it needed to have Go installed differently than it was on Ethereum. After Go was downloaded then an attempt to compile it was made and several errors were encountered indicating that Go application could not be found even though it was downloaded, and the researcher verified that the compile was being done from the correct file location. The Virtual Raspberry Pi was rebooted, the setup process was attempted again, and this time it successfully completed. The virtual environment was rebooted again and the researcher attempted to install Tendermint only to get an error. It was discovered that the Go environment had been completely reset so it was reset up again, another reboot was completed, the Go environment settings were checked and it was discovered that the settings for the Go environment had been removed again. Go was uninstalled, and removed from the virtual environment and Tendermint was removed from the virtual Raspberry Pi, the server was rebooted, the process was attempted again, and the same error was received. The processing and computational requirements for Go were reviewed



and no determination has been made as to why the Go environment on the Tendermint Virtual Raspberry Pi was being reset.

#### 4.4.5 Virtual Tendermint Ubuntu Server

A virtual server was created for Tendermint by creating a 160 GB hard drive and allocating one GB of RAM to it then the Ubuntu Server was downloaded and installed. The Virtual Tendermint Server was powered on, pending updates were installed and Go was downloaded, installed, and setup. Tendermint was then downloaded and permitted to sync but it has not downloaded all of the blocks in its blockchain.

### 4.5 Results

This section will provide a brief overview of the results that were obtained in the implementation and installation procedures for the Ethereum and Tendermint Blockchains' on the Raspberry Pi 3B, the Raspberry Pi 3B+, the Virtual Raspberry Pi's as well as the Virtual Ubuntu Server in which Tendermint Blockchain was installed on.

#### 4.5.1 Raspberry Pi 3B and Raspberry Pi 3B+ Issues

There were several issues that were encountered with this research including temperature and voltage issues that were later connected to issues with the overclocking of the processor, implementation issues with the Tendermint and Ethereum blockchains that were used as well as them not having accurate hardware, and software requirements.

There were several issues with the Raspberry Pi 3B and Raspberry Pi 3B+ that were related to the overclocking of the processor. While installing and initially syncing Ethereum the researcher noticed that some of the blocks in the blockchain were not syncing or downloading correctly and the area around the Raspberry Pi's was getting and staying very warm. Due to the

warmth of the area around the Raspberry Pi's temperature and voltage readings were taken and it was discovered that both of them were getting very hot. When initial measurements were taken it was noticed that the Raspberry Pi 3B+ was getting warmer than the Raspberry Pi 3B. The fact that the Raspberry Pi 3B+ was having more issues with controlling its temperature as well as voltage the researcher found to be surprising as well as not surprising. The reason that the Raspberry Pi 3B+ overheating issues were not a surprise was due to the fact that it had a faster processor which meant that it had the ability to get warmer than the Raspberry Pi 3B and since there was no external fan attached to the device to provide it with additional cooling it had to cool itself off. The temperature and voltage control issues of the Raspberry Pi 3B+ were a surprise because several other users had tested the temperature as well as the voltage and power needs on the Raspberry Pi 3B and the Raspberry Pi 3B+ and found that the Raspberry Pi 3B+ was able to stay approximately cooler 10 degrees Celsius cooler than the Raspberry Pi 3B (Manuel, 2018). Later tests would substantiate and validate the claims that the Raspberry Pi 3B+ is able to run approximately 10 degrees cooler than the Raspberry Pi 3B. The image 4.87 contains a comparison of the watts and temperature of the Raspberry Pi 3B and Raspberry Pi 3B+ while idle as well as under load.

	RPI 3 B+	RPI 3 B
Watt, Idle	2,3 W	1,3 W
Watt, Load	5,4 W	3,7 W
Temperature, Idle	44 °C	40 °C
Temperature, Load	66 °C	75 °C

*Figure 4.87.* Chart comparing the voltages on the Raspberry Pi 3B and Raspberry Pi 3B+ (Manuel, 2018, fig. 2)

Due to the temperature and voltage regulation issues that were encountered on the Raspberry Pi 3B and Raspberry Pi 3B+ it was decided that Tendermint would be installed on both of the Raspberry Pi's with the goal of determining if the issues were caused by the equipment itself or the demands that the Ethereum Blockchain was putting on the equipment.

#### 4.5.2 Tendermint

Several issues were encountered when trying to complete the installation and setup Tendermint including unclear as well as incomplete installation instructions, hardware and software requirements were hard to find as well as requirements for several additional applications that were not documented. The first issue that was encountered when trying to install Tendermint was finding the correct installation instructions that pertained to installing it on a Raspberry Pi 3B+. The installation steps that were used were the standard ones that were found on the Tendermint Website (*Tendermint Core: Install Tendermint*, 2018). Another issue that was encountered when trying to install Tendermint was obtaining the minimum specifications that would be needed to ensure that the blockchain was able to properly sync. There were a variety of specifications that were obtained that stated that Tendermint had been able to successfully run on a 1.0 GHz processor as well as a 1.4 GHz processor which is the computing requirements that were officially published on the Tendermint Website on August 18, 2018 (*Tendermint core: Running in production*, 2018).

The third issue that was encountered when installing Tendermint on the Raspberry 3B+ was the fact that there is no documentation that provided information that the additional applications would be needed to complete the installation. When attempting to install the additional applications there were several errors that were displayed which meant that the applications were not supported on a Raspberry Pi 3B+ or they were not available to be downloaded. Due to the numerous issues that were encountered when attempting to install Tendermint on the Raspberry Pi 3B as well as the Raspberry Pi 3B+ a decision was made to start

looking at virtual options as well as resuming the syncing of Ethereum Blockchain on the Raspberry Pi 3B as well as the Raspberry Pi 3B+. Additionally, Tendermint claims to be a lightweight blockchain but based upon the numerous issues that were encountered when trying to set it up on a Raspberry Pi 3B+ it is the opinion of the researches that it is not as lightweight as it is claims to be.

#### 4.5.3 Ethereum Raspberry Pi 3B and Raspberry Pi 3B+ Part 2

Despite the temperature and voltage issues that were encountered while initially setting up the Raspberry Pi 3B and Raspberry Pi 3B+ to sync the Ethereum blockchain it was decided that this was an avenue that needed additional research. The goal of resuming the syncing of the Ethereum on the Raspberry Pi 3B as well as the Raspberry Pi 3B+ was to determine what was causing the voltage and temperature to remain elevated.

When the Raspberry Pi's initially began to resync it was noticed that they were experiencing the same issues with temperature control as they were in the first phase of testing with the Ethereum Blockchain. Like the first phase of testing with the Ethereum Blockchain as the Raspberry Pi's began to get caught up to the current blocks on the blockchain it was noticed that the temperature and voltage readings began to stabilize but they still occasionally experienced intermittently high values. The temperature and voltage readings were closely monitored for a few days and it was discovered that the irregular temperature and voltage measurements were due the blockchain not being able to sync properly and both of the Raspberry Pi's processors were having to work harder to sync the blocks. While monitoring the blockchain sync progress it was discovered that some of the blocks might have had issues with syncing due to their size because they kept being dropped and the cache size was currently 64 MB. The cache size was modified in order to determine what the optimum cache size was that would reduce the syncing issues with the blockchain. The cache sizes that were used for testing were 32 MB, 64 MB, 128 MB, 256 MB, 512 MB, 1024 MB as well as 2048 MB and the readings were taken in one, three, and

five-minutes after the cache size was modified on each of the Raspberry Pi's. The figure 4.94 contains the time and temperature data for the Raspberry Pi 3B which can be compared to the data that was obtained from the Raspberry Pi 3B+ which can be found in figure 4.95. The figure 4.96 contains the time and voltage data for the Raspberry Pi 3B which can be compared to the voltage data that was obtained from the Raspberry Pi 3B+ which is located in figure 4.97. Additionally the data that was collected in the one, three, and five minute incremental test runs on the Raspberry Pi 3B is displayed in figures 4.88, 4.92, and 4.90 and the corresponding data for the Raspberry Pi 3B+ can be viewed in figures 4.91, 4.89, and 4.93. The data is grouped into three figures for the Raspberry Pi 3B and the Raspberry Pi 3B+ with the first set of figures on each of the devices containing the 32 MB, 64 MB, and 128 MB cache sizes. The second set of images for the Raspberry Pi 3B and the Raspberry Pi 3B+ contain the data from the 256 MB, 512 MB, and 1024 MB cache sizes. The final set of figures from each of the Raspberry Pi's contains the data from the when the cache size was set equal to 2048 MB.

Below are the steps that were taken to determine what cache size permitted the most efficient syncing of the Ethereum Blockchains on the Raspberry Pi 3B as well as the Raspberry Pi 3B+:

1. Resumed syncing Ethereum on both of the Raspberry Pi's with cache sizes at 64 MB.
2. Noticed high temperature and irregular voltage measurements that were matching up with blockchain sync errors.
3. Modified the cache size so that it had a cache size of 32 MB, 64 MB, 128 MB, 256 MB, 512 MB, 1024 MB, and 2048 MB.
4. The blockchain was permitted to synchronize while measurements were taken at one, three, and five minutes after each of the caches were modified.
5. Reviewed results and found that 128 MB was the most stable cache size.
6. Permanently modified the cache value in the syncing command to be 128 MB.

Raspberry Pi 3B 32 MB Cache Size																						
Time	Temperature in Celsuis	Volts	Proc		Memory				Swap		IO		System		CPU				Difference in Cache	In>Out		
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa			st	
20:17:26	56.900	1.200	1	0	0	101,088	47,908	484,532	0	0	70	46	254	187	6	1	93	0	0	484,532	24	
20:17:39	60.100	1.200	2	0	0	52,136	48,068	516,428	0	0	70	49	254	187	6	1	93	0	0	31,896	21	
20:18:39	58.000	1.200	1	0	0	64,672	48,612	494,144	0	0	74	64	262	202	6	1	93	0	0	-22,284	10	
20:20:39	58.000	1.200	1	0	0	49,192	48,808	506,600	0	0	70	74	260	196	6	1	93	0	0	12,456	-4	
20:23:39	58.000	1.200	1	0	0	48,664	48,932	506,680	0	0	64	69	256	189	6	1	93	0	0	80	-5	

Raspberry Pi 3B 64 MB Cache Size																						
Time	Temperature in Celsuis	Volts	Proc		Memory					Swap		IO		System		CPU				Difference in Cache	In>Out	
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st			
20:31:21	57.500	1.200	1	0	0	58,404	49,284	508,352	0	0	53	58	244	168	5	1	94	0	0	1,672	-5	
20:32:21	56.400	1.200	1	0	0	34,536	49,332	508,324	0	0	52	56	250	178	5	1	94	0	0	-28	-4	
20:34:21	56.900	1.200	1	0	0	36,680	49,352	500,892	0	0	49	54	248	174	5	1	94	0	0	-7,432	-5	
20:37:21	56.900	1.200	1	0	0	51,008	49,468	485,576	0	0	47	51	246	169	5	1	95	0	0	-15,316	-4	

Raspberry Pi 3B 128 MB Cache Size																						
Time	Temperature in Celsuis	Volts	Proc		Memory					Swap		IO		System		CPU				Difference in Cache	In>Out	
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st			
20:41:50	56.900	1.200	1	0	0	76,092	49,624	485,648	0	0	43	47	241	161	4	1	95	0	0	72	-4	
20:42:50	56.900	1.200	1	0	0	58,128	49,652	485,648	0	0	42	46	246	170	4	1	95	0	0	0	-4	
20:44:50	56.400	1.200	1	0	0	54,452	49,664	485,648	0	0	41	45	244	166	4	0	95	0	0	0	-4	
20:47:50	56.400	1.200	1	0	0	52,348	49,812	485,720	0	0	38	43	243	163	4	0	95	0	0	72	-5	
20:49:50	55.800	1.200	1	0	0	85,008	49,876	485,724	0	0	37	41	241	159	4	0	95	0	0	4	-4	

Figure 4.88. Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B was set equal to 32 MB, 64 MB, and 128 MB



Raspberry Pi 3B 256 MB Cache Size																				
Time	Temperature in Celsius	Volts	Proc r	Memory				Swap		IO		System		CPU				Difference in Cache	In>Out	
				b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id			wa
20:53:37	56.900	1.200	2	0	0	71,928	49,972	485,796	0	0	35	39	237	153	4	0	96	0	0	-4
20:54:38	58.000	1.275	0	0	0	39,404	49,992	485,800	0	0	35	38	242	163	4	0	95	0	0	-3
20:56:38	56.400	1.200	1	0	0	38,548	50,020	485,796	0	0	34	37	243	163	4	0	96	0	0	-3
20:59:38	55.800	1.200	1	0	0	44,580	50,148	478,608	0	0	32	36	242	160	4	0	96	0	0	-4
21:01:38	55.800	1.200	0	0	0	91,152	50,188	478,612	0	0	31	35	241	158	4	0	96	0	0	-4

Raspberry Pi 3B 512 MB Cache Size																				
Time	Temperature in Celsius	Volts	Proc r	Memory				Swap		IO		System		CPU				Difference in Cache	In>Out	
				b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id			wa
21:04:40	58.000	1.200	2	0	0	60,816	50,352	488,784	0	0	30	34	238	154	4	0	96	0	0	-4
21:05:41	57.500	1.200	2	0	0	61,184	46,288	454,464	0	0	30	39	243	162	4	0	96	0	0	-9
21:07:41	56.400	1.200	1	0	0	59,316	46,312	454,476	0	0	29	38	242	160	4	0	96	0	0	-9
21:10:41	56.400	1.200	1	0	0	58,220	46,384	454,516	0	0	28	36	241	158	3	0	96	0	0	-8

Raspberry Pi 3B 1024 MB Cache Size																				
Time	Temperature in Celsius	Volts	Proc r	Memory				Swap		IO		System		CPU				Difference in Cache	In>Out	
				b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id			wa
21:16:21	58.000	1.200	2	0	0	88,876	46,516	454,536	0	0	26	34	238	153	3	0	96	0	0	-8
21:17:21	56.900	1.200	1	0	0	73,708	46,800	454,564	0	0	26	34	242	160	3	0	96	0	0	-8
21:19:21	56.900	1.200	1	0	0	69,280	46,844	454,568	0	0	25	33	241	158	3	0	96	0	0	-8
21:22:21	56.900	1.200	1	0	0	68,268	46,872	454,588	0	0	25	32	240	156	3	0	96	0	0	-7
21:23:21	56.400	1.200	1	0	0	109,116	46,912	454,596	0	0	24	32	240	155	3	0	96	0	0	-8

Figure 4.89. Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B was set equal to 256

MB, 512 MB, and 1024 MB

Raspberry Pi 3B 2048 MB Cache Size																					
Time	Temperature in Celsuis	Volts	Proc		Memory				Swap		IO		System		CPU			Difference in Cache	In>Out		
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id			wa	st
21:24:57	58.500	1.275	0	0	0	84,568	47,000	454,608	0	0	24	31	240	155	3	0	96	0	0	12	-7
21:25:57	59.100	1.275	0	0	0	63,232	47,020	454,608	0	0	24	31	244	163	3	0	96	0	0	0	-7
21:27:57	56.900	1.200	1	0	0	51,316	47,032	454,608	0	0	23	30	249	173	3	0	96	0	0	0	-7
21:30:57	56.900	1.200	1	0	0	48,836	47,064	454,608	0	0	23	30	254	182	3	0	96	0	0	0	-7
21:31:57	56.900	1.200	3	0	0	106,592	47,092	454,636	0	0	22	29	256	185	3	0	96	0	0	28	-7

Figure 4.90. Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B was set equal to 1024 MB



Raspberry Pi 3B+ 32 MB Cache Size																						
Time	Temperature in Celsius	Volts	Procs		Memory					Swap		IO		System			CPU				Difference in Cache	In>Out
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st			
21:42:28	55.300	1.2000	0	0	7,424	74,608	44,772	458,580	0	0	0	1	5	7	1	0	99	0	0	458,580	-1	
21:42:38	55.800	1.2000	8	0	7,424	63,604	44,812	458,700	0	0	0	1	5	7	1	0	99	0	0	120	-1	
21:43:38	56.400	1.2000	2	0	7,424	48,956	44,836	458,724	0	0	0	1	5	7	1	0	99	0	0	24	-1	
21:45:38	54.800	1.2000	2	0	7,424	48,460	44,844	458,724	0	0	0	1	5	7	1	0	99	0	0	0	-1	
21:48:38	55.300	1.2000	0	0	7,424	39,592	44,940	464,376	0	0	0	1	5	7	1	0	99	0	0	5,652	-1	
21:50:38	55.300	1.2000	3	0	7,424	67,972	44,964	464,372	0	0	0	1	5	7	1	0	99	0	0	-4	-1	

Raspberry Pi 3B+ 64 MB Cache Size																						
Time	Temperature in Celsius	Volts	Procs		Memory					Swap		IO		System			CPU				Difference in Cache	In>Out
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st			
21:55:44	57.500	1.3938	0	1	8,192	91,476	45,232	433,812	0	0	0	1	5	7	1	0	99	0	0	-30,560	-1	
21:56:45	56.900	1.2000	0	0	12,800	127,888	16,940	407,100	0	0	1	1	5	7	1	0	99	0	0	-26,712	0	
21:58:45	55.800	1.2000	2	0	12,800	125,832	16,964	407,928	0	0	1	1	6	7	1	0	99	0	0	828	0	
22:01:45	60.700	1.2000	2	0	12,800	105,912	17,020	411,060	0	0	1	1	6	7	1	0	99	0	0	3,132	0	
22:03:45	58.500	1.2000	2	0	12,800	93,632	14,372	474,064	0	0	1	1	6	7	1	0	99	0	0	63,004	0	

Raspberry Pi 3B+ 128 MB Cache Size																						
Time	Temperature in Celsius	Volts	Procs		Memory					Swap		IO		System			CPU				Difference in Cache	In>Out
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st			
22:11:47	55.800	1.2000	0	0	12,800	94,632	14,512	473,092	0	0	1	1	6	7	1	0	99	0	0	-972	0	
22:12:47	56.900	1.2000	2	0	12,800	62,052	14,720	473,120	0	0	1	1	6	7	1	0	99	0	0	28	0	
22:14:48	56.900	1.3938	0	0	12,800	60,252	14,740	473,120	0	0	1	1	6	7	1	0	99	0	0	0	0	
22:17:48	56.400	1.2000	4	0	12,800	59,576	14,876	473,136	0	0	1	1	6	7	1	0	99	0	0	16	0	
22:19:48	56.400	1.2000	2	0	12,800	97,156	14,904	473,136	0	0	1	1	6	7	1	0	99	0	0	0	0	

Figure 4.91. Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B+ was set equal to 32

MB, 64 MB, and 128 MB

Raspberry Pi 3B+ 256 MB Cache Size																						
Time	Temperature in Celsius	Volts	Procs		Memory				Swap		IO			System			CPU				Difference in Cache	In>Out
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st			
22:21:16	55.800	1.2000	1	0	12,800	97,404	14,956	473,160	0	0	1	1	6	7	1	0	99	0	0	24	0	
22:22:16	58.000	1.2000	4	0	12,800	118,492	11,704	401,956	0	0	1	1	6	7	1	0	99	0	0	-71,204	0	
22:24:16	56.900	1.2000	4	0	12,800	108,380	11,724	401,956	0	0	1	1	6	7	1	0	99	0	0	0	0	
22:27:17	58.000	1.2000	2	0	12,800	58,228	12,024	447,512	0	0	1	1	7	7	1	0	99	0	0	45,556	0	
22:29:17	56.400	1.2000	2	0	12,800	126,252	12,060	447,524	0	0	1	1	7	8	1	0	99	0	0	12	0	

Raspberry Pi 3B+ 512 MB Cache Size																					
Time	Temperature in Celsius	Volts	Procs		Memory				Swap		IO		System		CPU				Difference in Cache	In>Out	
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa			st
22:32:02	56.400	1.2000	1	0	12,800	113,084	12,128	447,692	0	0	1	1	7	8	1	0	99	0	0	168	0
22:33:02	56.900	1.2000	1	0	12,800	89,424	12,136	447,692	0	0	1	1	7	8	1	0	99	0	0	0	0
22:35:02	56.400	1.2000	2	0	12,800	86,948	12,144	447,692	0	0	1	1	7	8	1	0	99	0	0	0	0
22:38:03	56.400	1.2000	1	0	12,800	85,684	12,152	447,692	0	0	1	1	7	8	1	0	99	0	0	0	0
22:40:03	55.800	1.2000	3	0	12,800	125,372	12,180	447,704	0	0	1	1	7	8	1	0	99	0	0	12	0

Raspberry Pi 3B+ 1024 MB Cache Size																					
Time	Temperature in Celsius	Volts	Procs		Memory				Swap		IO		System		CPU				Difference in Cache	In>Out	
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa			st
22:42:04	56.400	1.3938	2	0	12,800	108,448	12,276	447,736	0	0	1	1	7	8	1	0	99	0	0	32	0
22:43:04	56.900	1.2000	2	0	12,800	87,368	12,300	447,728	0	0	1	1	7	8	1	0	99	0	0	-8	0
22:45:04	56.400	1.2000	2	0	12,800	65,928	12,312	447,728	0	0	1	1	7	8	1	0	99	0	0	0	0
22:48:05	55.800	1.2000	2	0	12,800	62,616	12,408	447,808	0	0	1	1	7	8	1	0	99	0	0	80	0
22:49:05	56.400	1.2000	3	0	12,800	121,752	12,452	447,816	0	0	1	1	7	8	1	0	99	0	0	8	0

Figure 4.92. Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B was set equal to 256 MB, 512 MB, and 1024 MB



Raspberry Pi 3B+ 2048 MB Cache Size																					
Time	Temperature in Celsius	Volts	Procs		Memory				Swap		IO		System		CPU				Difference in Cache	In>Out	
			r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa			st
22:51:12	57.500	1.3938	1	1	12,800	99,896	12,584	457,944	0	0	1	1	8	8	1	0	99	0	0	10,128	0
22:52:12	58.000	1.2000	4	0	12,800	115,620	10,548	404,164	0	0	1	1	8	8	1	0	99	0	0	-53,780	0
22:54:12	56.900	1.3938	2	0	12,800	100,768	10,612	404,184	0	0	1	1	8	8	1	0	99	0	0	20	0
22:57:12	56.900	1.2000	2	0	12,800	107,072	11,128	404,692	0	0	1	1	8	8	1	0	99	0	0	508	0
22:58:12	56.900	1.2000	4	0	12,800	153,992	12,080	406,108	0	0	1	1	8	8	1	0	99	0	0	1,416	0

Figure 4.93. Image of a chart containing the data that was collected when the cache size on the Raspberry Pi 3B was set equal to 2048 MB

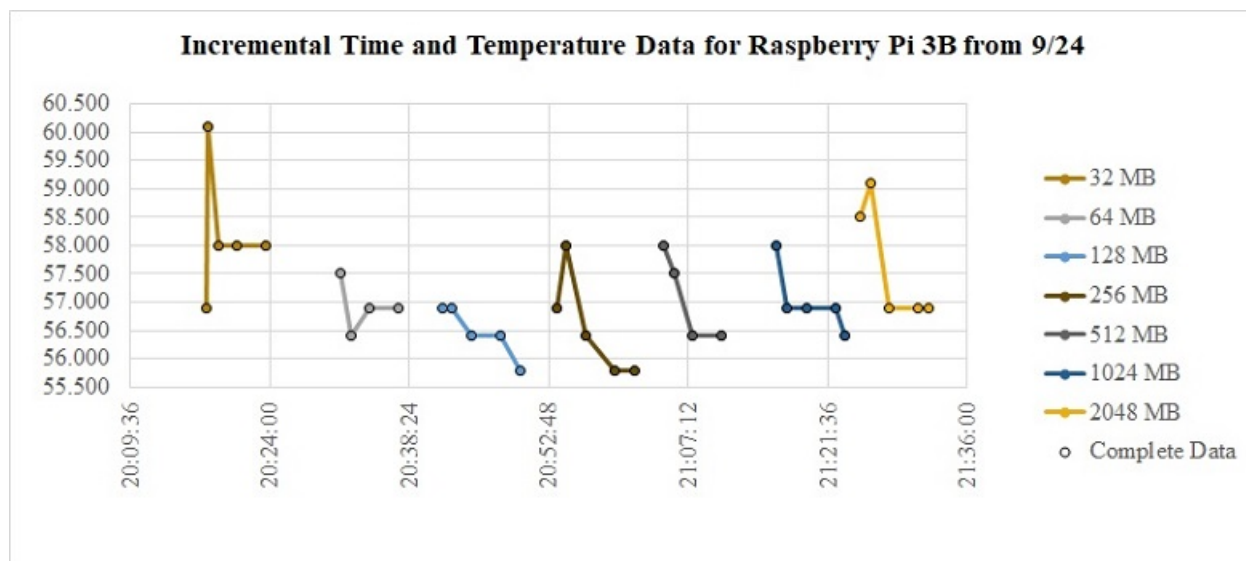


Figure 4.94. Incremental Time and Temperature Data that was obtained on the Raspberry Pi 3B

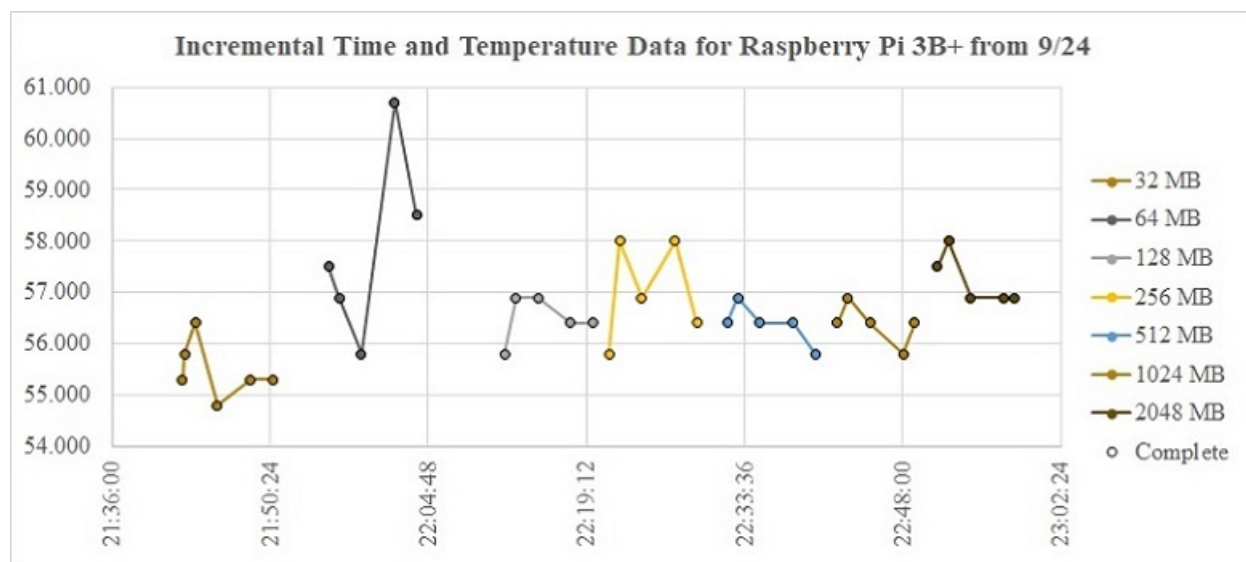


Figure 4.95. Incremental Time and Temperature Data that was obtained on the Raspberry Pi 3B+

After the above time, temperature, and voltage readings were taken while modifying the cache size it was theorized that the issues were related and possibly negatively impacting the function of the Raspberry Pi's. After discovering this relationship, the Raspberry Pi's were permitted to resume intermittently syncing the Ethereum Blockchain from October 20th to

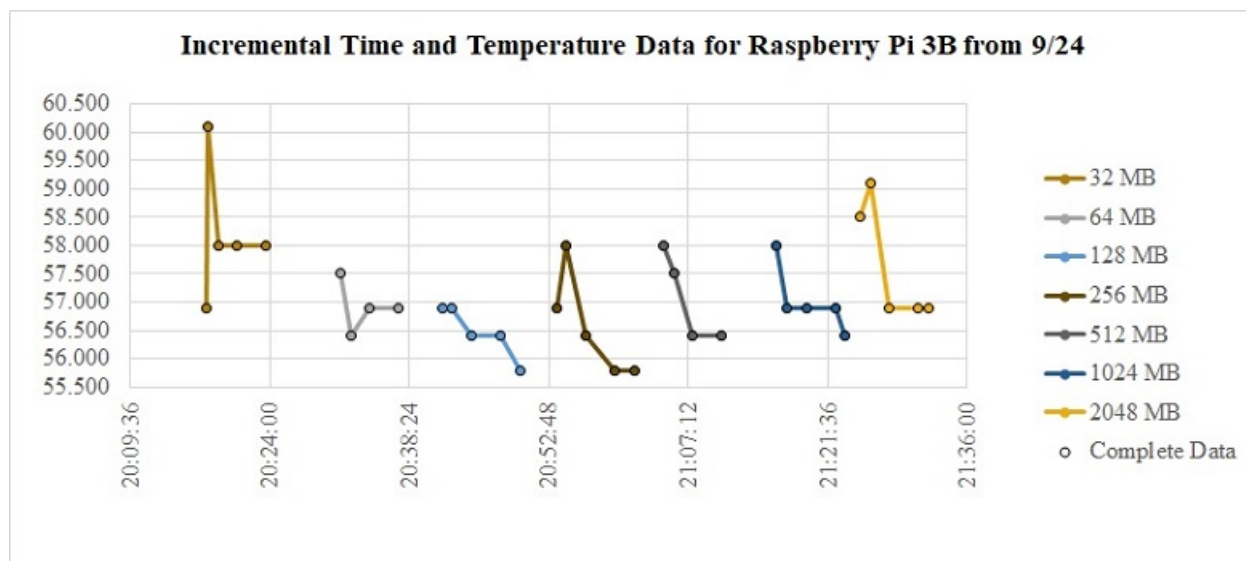


Figure 4.96. Incremental Time and Voltage Data that was obtained on the Raspberry Pi 3B

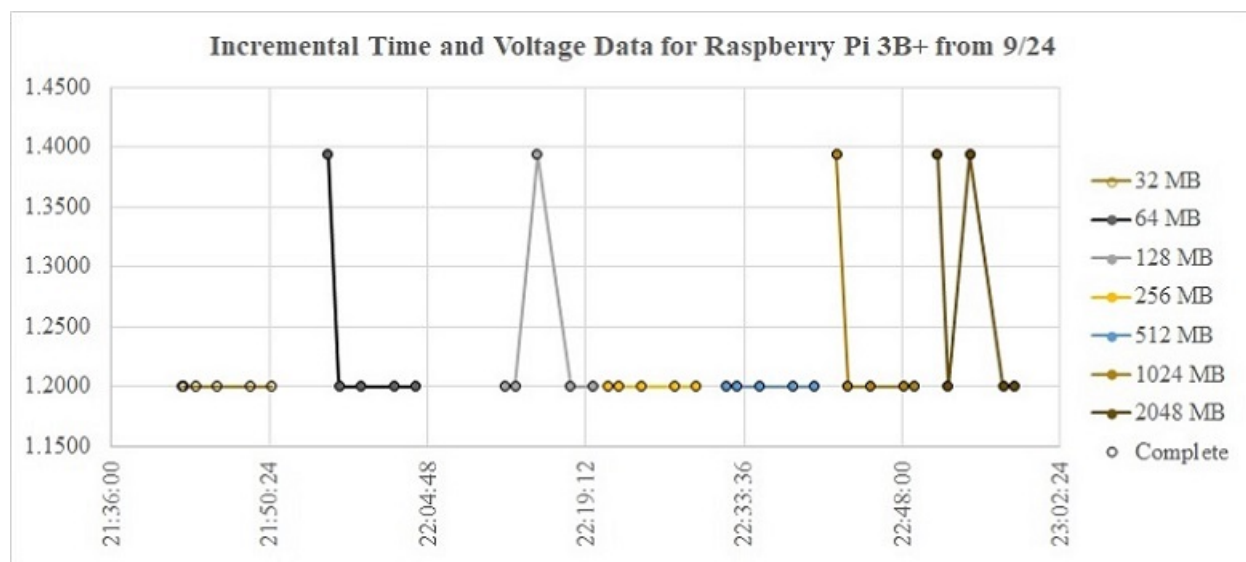


Figure 4.97. Incremental Time and Voltage Data that was obtained on the Raspberry Pi 3B+

October 25th while the researcher was able to monitor it. The Raspberry Pi's were not permitted to sync constantly for this portion of the testing due to the risk of fire because of unstable temperature and voltage readings. On October 25th the researcher determined that the temperature and voltage had experienced a significant stabilization and decided that both of the

Raspberry Pi's would be permitted to sync overnight and they would be checked in the morning to determine if there were any issues and the next morning there appeared to be none. While the Ethereum Blockchain was syncing on both of the Raspberry Pi's a script was running on each of the devices that took measurements of the temperature, voltage, recorded the active processes as well as the date and time of the measurements in five minute increments. The researcher noticed that the initial temperature and voltage measurements that were taken were unstable but as the Raspberry Pi 3B and the Raspberry Pi 3B+ were permitted to sync the Ethereum Blockchain that there were fewer and fewer issues that were being experienced with the temperature and the voltage measurements on the devices. As the temperature and voltages began to stabilize the researcher was able to establish a connection between the processor usages that are listed below and the temperature and voltage readings.

Primarily the Raspberry Pi 3B was the one that was the device that was more negatively impacted by the overclocking of the processor. The data has been reviewed and it was determined that several factors were involved in the overclocking of the processor including the number of running processes, the use of swapped memory, the amount of memory that was currently free, the blocks of data that were being written to the disk, increase and decrease in the use of the memory cache, the number of system interrupts that were seen and the number of context switches that were needing to be completed. The combination of the previously mentioned issues can be seen when reviewing the data from the Raspberry Pi 3B that is contained within 4.98 that was obtained on October 20th from 16:27:03 (4:27:03 PM) to 16:47:07 (4:47:07 PM) while it was syncing the Ethereum Blockchain. During this timeframe the voltage reached a maximum of 1.275 volts, and the temperature was measured at 84.4 degrees Celsius. The swapped (swpd) memory column is used to report the amount of "memory that has been swapped out to a swap file or disk" and in this case it was 25,088 MB of data (Linode, 2017, para. 11). The free memory is used to report the amount of memory that is unallocated and in this case there was 30,684 MB of available space (Linode, 2017, para. 11). The buffer value was 2,880 and it is the amount of allocated memory that is currently in use (Linode, 2017). The cache value is used to report "the

amount of allocated memory that could be swapped to a disk or unallocated if the resources are needed for another task” and at the time of the high temperature it was 133,996 MB (Linode, 2017, para. 11). The blocks in (bi) value is used to report the number of inbound “blocks that have been recieved” per second from the disk and at the time of the peak temperature it was 39 (Linode, 2017, para. 13). The value of 50 that is in the blocks out (bo) column is used to indicate the number of blocks that have been sent out to the disk per second (Linode, 2017). The in column is used to report “the number of system interrupts” that are being reported per second including those that are received from the system clock and the value that was reported at the time of the temperature spike was 211 (Linode, 2017, para. 14). The value of 73 that is in the context switches (cs) column is used to indicate “the number of context switches that the system” is making to process all of the tasks (Linode, 2017, para. 14). The 10 value that is located in the us column is used to report “the amount of time that the processor spends on” non-kernal processes or userland tasks (Linode, 2017, para. 15). The 89 value that is located in the idle (id) column is used to indicate the amount of time that the processor is sitting in an idle state (Linode, 2017). The cache value experienced a 8,168 MB increase which when combined with the negative number in the in vs out column it meant that there were more blocks being written to the disk than were being received from the disk. The in vs out column is the difference of the blocks out (bo) and the blocks in (bi) columns. Lastly combination of the above factors resulted in a variety of issues on the Raspberry Pi 3B due to processor overheating including the slowdown of the computational abilities as well as the temperature and voltage issues. If the above values are compared with the measurements that were taken one minute later then there is a significant drop in the values resulting in a normalization of measurements.

The primary reason that these issues caused such concern is because there was a possibility that one if not both of the Raspberry Pi’s could become damaged due to each of their processors being overclocked for an extended period of time and possibly causing damage to the devices. Additionally the inconsistent temperatures and voltage measurements that primarily occurred on the Raspberry Pi 3B as well as the occasional measurements that were obtained on

the Raspberry Pi 3B+ were an issue because the high probability that the functionality of the processor will be negatively impacted. Lastly the erratic voltage and temperature measurements had a secondary concern with them and that was the possibility that one or both of the Raspberry Pi's would experience a surge in voltage or a sudden increase in temperature due to the processor being overclocked and possibly damaging the board itself.

Based upon the processor utilization history as well as the temperature and voltage measurements that was obtained during this research it is the opinion of the researcher that the Ethereum Blockchain required additional processor utilization while it was syncing. This additional processor utilization caused an increase in the temperature as well as the voltage of the Raspberry Pi 3B. Lastly it is possible that since this issue has not been documented by other users that much that the Raspberry Pi 3B that was being used for testing had an issue with its processor that had not been detected prior to this testing.

The testing portion of this research was completed in the following order: Create Ethereum Raspberry Pi's, Create Tendermint Raspberry Pi's, Resume research and testing with Ethereum Raspberry Pi's, Create virtual Ethereum Raspberry Pi, Create Virtual Tendermint Raspberry Pi, and Create Virtual Tendermint Server.

#### 4.5.4 FOAM Token-Server Implementation

There were several issues that were encountered when trying to implement the Foam Token-Server on the Raspberry Pi's as well as in the Ethereum Virtual Raspberry Pi including incomplete requirements for operating systems, hardware, and software as well as not being able to purchase tokens that would be needed to stake a location. The FOAM website as well as the FOAM GitHub Repository provided no information as to what operating system would be able to run FOAM. A review of the existing code was completed, and it was determined that it most likely would be supported on some variation of a Linux Operating System. The FOAM Token-Server listed the following requirements:



Raspberry Pi 3B Date Time	Temperature in Celsius	Procs			Memory			Swap		IO		System			CPU			Difference in Cache	In vs. Out		
		Voltage	r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id			wa	st
10/20/18 16:27:03	53.7000	1.2000	1	0	0	212,916	40,604	380,172	0	0	19	9	187	44	1	0	99	0	0	0	10
10/20/18 16:28:03	54.8000	1.2750	0	0	0	208,528	40,636	380,168	0	0	19	9	187	45	1	0	99	0	0	-4	10
10/20/18 16:29:03	63.4000	1.2750	5	1	0	133,388	40,748	403,824	0	0	18	10	190	48	1	0	99	0	0	23,656	8
10/20/18 16:30:03	75.8000	1.2750	6	0	0	89,952	41,100	388,412	0	0	18	14	192	51	2	0	98	0	0	-15,412	4
10/20/18 16:31:03	80.6000	1.2750	4	0	1,024	31,816	38,688	219,056	0	0	19	14	195	54	3	0	97	0	0	-169,356	5
10/20/18 16:32:04	82.7000	1.2750	6	0	1,024	31,600	38,736	197,888	0	0	18	15	197	56	4	0	96	0	0	-21,168	3
10/20/18 16:33:04	82.2000	1.2750	6	0	3,072	28,552	316	232,236	0	0	20	19	199	58	5	0	95	0	0	34,348	1
10/20/18 16:34:04	80.6000	1.2750	1	0	3,072	30,552	388	219,296	0	0	20	24	200	59	6	0	94	0	0	-12,940	-4
10/20/18 16:35:04	83.8000	1.2750	4	0	4,608	34,716	3,420	118,708	0	0	26	31	203	62	7	0	93	0	0	-100,588	-5
10/20/18 16:36:05	83.3000	1.2750	4	0	4,608	29,864	3,664	122,684	0	0	36	41	206	68	8	0	92	0	0	3,976	-5
10/20/18 16:37:05	83.8000	1.2750	5	0	4,608	31,736	3,668	120,264	0	0	36	41	208	70	9	0	91	0	0	-2,420	-5
10/20/18 16:38:06	83.8000	1.2750	5	0	11,264	28,424	3,140	125,828	0	1	38	47	209	71	10	0	90	0	0	5,564	-9
10/20/18 16:39:07	84.4000	1.2750	5	0	25,088	30,684	2,880	133,996	0	1	39	50	211	73	10	0	89	0	0	8,168	-11
10/20/18 16:40:07	68.8000	1.2000	5	0	25,088	424,228	2,940	180,416	0	1	40	51	213	77	11	0	89	0	0	46,420	-11
10/20/18 16:41:07	65.0000	1.2000	1	0	25,088	422,956	2,968	180,416	0	1	40	51	215	81	11	0	89	0	0	0	-11
10/20/18 16:42:07	62.3000	1.2000	4	0	25,088	482,348	3,040	180,424	0	1	40	50	217	85	10	0	89	0	0	8	-10
10/20/18 16:43:07	60.1000	1.2000	1	0	25,088	477,324	3,068	180,444	0	1	39	49	220	91	10	0	89	0	0	20	-10
10/20/18 16:44:07	58.0000	1.2000	1	0	25,088	473,504	3,084	180,444	0	1	39	49	220	91	10	0	89	0	0	0	-10
10/20/18 16:45:07	58.5000	1.2000	1	0	25,088	472,528	3,112	180,448	0	1	38	48	220	92	10	0	89	0	0	4	-10
10/20/18 16:46:07	61.2000	1.2000	3	0	25,088	350,344	3,820	285,160	0	1	42	55	221	93	10	0	89	0	0	104,712	-13
10/20/18 16:47:07	59.1000	1.2000	1	0	25,088	349,264	3,836	285,160	0	1	42	54	221	93	10	0	89	0	0	0	-12

Figure 4.98. Image of a chart showing all of the data that was collected on the Raspberry Pi 3B during a temperature spike

Raspberry Pi 3B+			Procs		Memory				Swap		IO		System		CPU				Difference in Cache	In vs. Out
Date/Time	Temperature in Celsius	Voltage	r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st	
10/20/18 16:26:28	55.8000	1.2000	2	0	0	432,752	37,332	166,736	0	0	11	1	581	50	1	0	99	0	0	0
10/20/18 16:27:28	55.8000	1.2000	2	0	0	430,768	37,352	166,740	0	0	11	1	581	51	1	0	99	0	0	4
10/20/18 16:28:29	56.9000	1.2000	3	0	0	184,220	38,144	384,992	0	0	17	5	582	53	1	0	99	0	0	218,252
10/20/18 16:29:29	56.4000	1.2000	1	0	0	272,332	38,224	295,656	0	0	17	5	583	54	1	0	99	0	0	-89,336
10/20/18 16:30:29	56.4000	1.2000	1	0	0	272,204	38,244	295,660	0	0	17	5	583	54	1	0	99	0	0	4
10/20/18 16:31:29	56.4000	1.2000	1	0	0	272,460	38,260	295,664	0	0	16	5	583	54	1	0	99	0	0	4
10/20/18 16:32:29	54.8000	1.2000	1	0	0	271,924	38,280	295,668	0	0	16	5	583	54	1	0	99	0	0	4
10/20/18 16:33:29	56.4000	1.2000	0	0	0	272,508	38,312	295,664	0	0	16	5	583	54	1	0	99	0	0	-4
10/20/18 16:34:29	55.8000	1.2000	1	0	0	276,556	38,340	295,668	0	0	16	5	583	55	1	0	99	0	0	4
10/20/18 16:35:29	56.9000	1.2000	1	0	0	276,380	38,352	295,672	0	0	16	5	584	55	1	0	99	0	0	4
10/20/18 16:36:29	55.8000	1.2000	2	0	0	276,460	38,380	295,672	0	0	15	5	584	55	1	0	99	0	0	0
10/20/18 16:37:29	56.4000	1.2000	1	0	0	310,900	38,400	295,672	0	0	15	5	584	55	1	0	99	0	0	0
10/20/18 16:38:29	56.4000	1.2000	1	0	0	310,976	38,436	295,668	0	0	15	5	584	55	1	0	99	0	0	-4
10/20/18 16:39:30	55.8000	1.2000	1	0	0	349,904	38,516	295,452	0	0	15	5	585	58	1	0	99	0	0	-216
10/20/18 16:40:30	56.9000	1.2000	3	0	0	313,380	38,592	295,472	0	0	15	5	588	64	1	0	99	0	0	20
10/20/18 16:41:30	55.8000	1.2000	6	0	0	293,004	38,628	295,472	0	0	15	5	592	71	1	0	99	0	0	0
10/20/18 16:42:30	56.9000	1.2000	1	0	0	312,204	38,692	295,560	0	0	14	5	596	78	1	0	99	0	0	88
10/20/18 16:43:30	56.9000	1.2000	1	0	0	311,148	38,712	295,568	0	0	14	5	596	78	1	0	99	0	0	8
10/20/18 16:44:30	56.4000	1.2000	1	0	0	311,104	38,728	295,568	0	0	14	5	596	78	1	0	99	0	0	0
10/20/18 16:45:30	55.8000	1.2000	2	0	0	310,980	38,756	295,572	0	0	14	5	596	78	1	0	99	0	0	4
10/20/18 16:46:30	55.3000	1.2000	1	0	0	310,764	38,780	295,568	0	0	14	5	596	78	1	0	99	0	0	-4
10/20/18 16:47:30	54.8000	1.2000	1	0	0	310,388	38,792	295,572	0	0	14	5	596	78	1	0	99	0	0	4

Figure 4.99. Image of a chart comparing all of the data that was collected on the Raspberry Pi 3B+ during a temperature spike

1. “PostGres database”
2. “Access to a synced main-net node where you have permission to install filters”
3. “An ERC20 address (the default is the OmiseGo token address)” (*GitHub Foam Token Server*, 2018, para. 2)

Another issue that was encountered while attempting to setup the FOAM token-Server was determining the system requirements that would be needed for it. The only information that was provided on GitHub about the system requirements of the Foam Token-Server is that the “required environment variables” could be found within the Makefile (*GitHub Foam Token Server*, 2018). The makefile was reviewed and there was not any information about the system requirements provided in the makefile only application setup parameters (blinkey3713, 2018).

The next issue proved to be the most significant which was in order to be able to set up a beacon you needed to have tokens that you could stake. The tokens were available for sale over the summer in the FOAM ICO sale but due to not owning Bitcoin the researcher was not able to purchase them (*Token Sale*, 2018). Additionally, it was highly unlikely that approval could have been obtained to purchase them with the department’s money. Lastly, the FOAM tokens were needed to completely set up a MetaMask Account which would be used to add Beacons to the map.

#### 4.6 Summary

This chapter contains a review of the the equipment that was selected as well as why it was selected, the feasibility criteria, the testing environment, data analysis, FOAM dependencies, successes and lessons learned during testing, and implementation procedures.

## CHAPTER 5. RECOMMENDATIONS, AND CONCLUSIONS

### 5.1 Introduction

The purpose of this research was to determine the feasibility of using blockchain to implement proof of location. To conclude this document this chapter summarizes the research that was completed, provides recommendations for additional research, and conclusions.

### 5.2 Summary

This section will provide a summary of the research that was completed, the successes as well as the lessons learned, limitations of the study, the problems that arose during research, the implications of the researcher's findings, and recommendations for further research. The purpose of this research was to determine the feasibility of using blockchain to implement proof of location. The research question that the researcher had the goal of answering was:

- Can a Raspberry Pi support implementing the FOAM token protocol?

The two main goals of this research were to install Ethereum and Tendermint on a Raspberry Pi 3B and a Raspberry 3B+ then download, install, and implement Proof of location on the Raspberry Pi 3B and the Raspberry Pi 3B+. It was because of the overarching goals that the researcher made two hypotheses about this research. The first or null hypothesis was FOAM implemented on Tendermint will be able to run more efficiently than FOAM on Ethereum. The second or alternate hypothesis was FOAM implemented on Tendermint will not be able to run more efficiently than FOAM on Ethereum. The reason that the first hypothesis stated that FOAM would be able to run more efficiently on Tendermint was that during the literature review it was discovered that Tendermint claimed to be able to support lightweight blockchain clients (*Tendermint core: Running in production*, 2018). To determine which of the hypotheses was

correct an extensive amount of testing as well as research had to be completed. There were a variety of successes as well as problems that were encountered while completing the research and testing phase. There were five iterations of testing that were completed and there were:

- Ethereum Raspberry Pi 3B and Raspberry Pi 3B+
- Tendermint Raspberry Pi 3B and Raspberry Pi 3B+
- Virtual Raspberry Pi Ethereum
- Virtual Raspberry Pi Tendermint
- Virtual Server Tendermint.

Each iteration of research and testing contained several subtasks that the researcher completed. In the first iteration of testing there were several tasks that were completed including setting up both Go and Ethereum on the Raspberry Pi 3B and Raspberry Pi 3B+ then they were compiled together to create Geth, and the Ethereum Blockchain was completely synced. Additionally, in the first iteration the CSC Explorer, FOAM, Purescript and the Purescript-eth-core were able to be downloaded to both the Raspberry Pi 3B and the Raspberry Pi 3B+. The next set of tasks that was completed was to download and set up Go in preparation for the installation of Tendermint and download Tendermint to the Raspberry Pi 3B and the Raspberry Pi 3B+. The third and fourth phases of testing and research is where the Go environment was created and setup in each of the Virtual Raspberry Pi's as well as having FOAM, Haskell-Opalyeye, and Foam-Token-Server downloaded to both of them. Ethereum was downloaded to one of the Virtual Raspberry Pi's and Tendermint was downloaded to the other one. The fifth and final iteration of testing is where Go was completely setup on the Virtual Ubuntu Server as well as the Tendermint Blockchain. Lastly the Tendermint blockchain was able to completely sync.

It should be stressed that this research has primarily focused upon determining the feasibility of using a blockchain to implement proof of location. The results that were obtained in testing were limited to three parameters. The use of Raspberry Pi 3B, and Raspberry Pi 3B+ as

testing equipment was the primary constraint that was driving this research. The consistent use of the same non-commercial or home CPS devices from the beginning of testing to the end were the second and third driving elements of this research. Additionally, the researcher would like to make it clear that that CPS devices that are currently being used by the US Federal Government were not used during this testing.

There were two types of problems that cropped up during the research and testing phases. The most significant issue that was encountered was related to the possible overclocking of the processor on the Raspberry Pi 3B causing temperature to be elevated. The Raspberry Pi 3B has a maximum safe operating temperature of 85 degrees Celsius and on October 20th of 2018 at 4:39 PM as shown in 4.8 a temperature recording of 84.4 degrees Celsius was obtained. After reviewing the data for that timeframe it is believed that there were several things that combined together to cause the increase in temperature including the number of running processes that were waiting to get access to the processor, the large amount of memory that had been swapped out to another location, the small amount of free memory that was currently available, increase in the blocks of data that were sent out to the disk as well as an 8,168 MB increase in the memory cache, and the numerous system interrupts that were being seen as well as the context switches that the system was needing to complete. It is the belief of the researcher that if one of the previously mentioned causes could have been prevented or controlled than the temperature and voltage issues that were encountered on the Raspberry Pi 3B would be minimized. It is also possible that the temperature and voltage issues that were seen on the Raspberry Pi 3B would have been closer to those that were seen on the Raspberry Pi 3B+ as shown in figure 4.10.

The second issue that consistently occurred during research and testing was related to hardware and software issues. The first set of issues that were encountered was the fact that FOAM would not compile on the Raspberry Pi 3B or the Raspberry Pi 3B+ due to an interdependence with other applications that were not supported on either piece of equipment or the application was not available to be downloaded. It is believed that part of the reason that FOAM would not compile properly is the fact that one or more of the programs or applications



that it was dependent such as Purescript-eth-core were not designed to be ran on Raspbian. In the next iteration of testing Tendermint was not able to be installed on the Raspberry Pi 3B due to insufficient computational power. It is believed that even though the Raspberry Pi 3B+ met the minimum computational specifications to run Tendermint it needed to have a slightly stronger processor.

In the next iteration of testing Go and Ethereum would not compile together to make Geth on the Virtual Raspberry Pi and the researcher believes that this issue was caused by one or two issues. The first reason that could have caused that Go and Ethereum could not compile together is the possibility that one or both of the applications were either too new or too old to be supported on the Debian Operating System. The second issue that could have prevented Geth from being compiled on the Virtual Raspberry Pi is that Go was being completely uninstalled for one of three reasons when the virtual Raspberry Pi was power cycled after setting up the Ethereum environment. The three reasons that GO could have been removed are it was uninstalling itself, the Virtual Raspberry Pi was being rolled back or reset for some reason, or the Raspbian Operating system determined that it was a threat and automatically removed it. The fourth phase of testing experienced a similar issue to the previous phase of testing except for in this phase of testing Tendermint was not able to be compiled on Debian OS and Go was also mysteriously reset. The fact that Tendermint was not able to be ran on the Raspberry Pi 3B+ as well as in a virtual Raspberry Pi leads the researcher to believe that Tendermint minimum specifications are higher than what is currently being stated (Authors, 2018). The final iteration of testing is where the most progress was seen wherein Tendermint was able to be installed and synced but FOAM was not able to be compiled on the virtual Ubuntu Server. The researcher believes that due to Tendermint using up a large amount of computational abilities in the virtual environment that FOAM was not able to run

In conclusion despite the successes that were encountered during the various iterations of research and testing the issues that were encountered including a variety of hardware and software issues which when combined with the processor issues causing temperature and voltage

irregularities proved to be too much to overcome. It is for this reason that both the null hypothesis and the alternative hypothesis were found to be not true.

### 5.3 Recommendations and Future Work

It is the opinion of the researcher that there are additional research avenues including presyncing the blockchain from a computer, using the next version of a Raspberry Pi, or a similar device such as an Asus Tinker Board, Parallela, BeagleBoard, or a PixelPro, connect a fan to the device, or add a heat sink, increase power supply that the Raspberry Pi is using, use an externally power hard drive to store the blockchain data, creating a distributed system that contains physical as well as virtual equipment, use a lightweight server application, and recreate this research and monitor all of the background and foreground processes.

The first recommendation that the researcher would like to make concerning possible avenues that could be used to continue this research is to predownload or presync the blockchain to the SD card where it will be stored. The SD card would be inserted into a computer and the blockchain would be downloaded and synced.

By predownloading or presyncing the blockchain to the SD card where it will be stored then inserting it back into the Raspberry Pi or similar device means that there will be less work that the device will have to complete to obtain the most recent blocks in the blockchain. Additionally, by presyncing the blockchain on a computer it will decrease the amount of time that it takes to sync the blockchain. Lastly since the temperature issues primarily occurred at the beginning of when the blockchains were being synced it is the opinion of the researcher that if the Raspberry Pi or similar device did not have to completely sync the blockchain then it is probably that there would be fewer temperature and voltage issues that originated with the processor working so hard to get caught up with the current blocks in the blockchain.

The next set of recommendations that could be used to continue this research is to connect a fan or add a heatsink onto the Raspberry Pi or similar device. By adding a heat sink onto the



device it would be able to transfer the heat from the device itself to the air where it will be able to be dissipated away from the device and aid in the regulation of the device's temperature.

Additionally, if a cooling fan were connected to the device and it was set to trigger when the temperature reaches around 65-70 degrees Celsius it is possible that this would reduce some of the issues with the processor overclocking. Lastly it was around 65-70 degrees Celsius that the researcher first noticed the issues with the processor overclocking and causing the temperature of the Raspberry Pi 3B to increase.

Replacing the standard power supply with a slightly higher power supply is the next recommendation that the researcher would like to make concerning ways to continue this research. The reason for this recommendation is that the Raspberry Pi 3B "uses between 700-1000mA depending on what peripherals are connected" (*Raspberry Pi Power Supply*, 2013, para. 2). By replacing the standard power supply that is 5.0 volt/2.5 Amp with a newer power supply that has 5.1 volt/2.5 Amp means that there will be a slight increase in the available watts that will be able to power the device.

Another way that this research could be continued would be to use a harddrive that has an external power supply attached to it to store the blockchain data. By utilizing a harddrive that has an external power supply on it there will be a decrease in the amount of power that is being drawn off of the Raspberry Pi or a similar device. The reason for this recommendation is that the researcher noticed that when large amounts of data were being transferred to the externally mounted USB that there was an increase in the voltage that was being used by the Raspberry Pi 3B and this was caused because USB thumbdrives do pull a small amount of power from the devices that they are connected too.

The sixth option that can be implemented in order to continue this research is to create a distributed system containing physical as well as virtual equipment. The distributed system should contain at least one Raspberry Pi 3B+ or a similar cheap device functioning as a sensor that will be used to collect data and it will be connected a virtual Ubuntu Server. The virtual Ubuntu server will be syncing either the Tendermint blockchain or the Ethereum Blockchain.

This option also resolves the issue of not being able to implement all of FOAM dependencies on one device because it separates the device that is collecting the data from the server that is syncing the blockchain.

The next possible research avenue to be explored is to recreate the research installing a lightweight Linux virtual server such as Puppy Linux, Lubuntu, or Linux Lite on the Raspberry Pi or a similar device. The researcher feels that using a lightweight Linux distribution such as one of those that was previously mentioned would prevent several of the issues that were encountered particularly those that were related to needing to have a full Linux server or a distribution to install applications on as well as being able to access the appropriate package applications.

The final avenue that the researcher feels that this research can be continued by using is to recreate this research and monitor all of the processes that are active as well as those that are running in the background to determine which application is causing the backup of the data in the application.

In conclusion a variety of additional research avenues was just reviewed including presyncing the blockchain from a computer, using the next version of a Raspberry Pi, or a similar device such as an Asus Tinker Board, Parallela, BeagleBoard, or a PixelPro, connect a fan to the device, or add a heat sink, increase power supply that the Raspberry Pi is using, use an externally power hard drive to store the blockchain data, creating a distributed system that contains physical as well as virtual equipment, use a lightweight server application, and recreate this research and monitor all of the background and foreground processes.

## 5.4 Conclusions

The purpose of this research was to determine the feasibility of using blockchain to implement proof of location. To conclude this document this chapter summarizes the research that was completed, provides recommendations for additional research, and conclusions.

## REFERENCES

- Ahluwalia, G. (2016). *IoT on the blockchain*. Retrieved from  
<http://www.w3.org/2016/04/blockchain-workshop/interest/ahluwalia.html>
- Amazon.com: *CanaKit Raspberry Pi 3 B+ Ultimate Starter Kit*. (2018). Retrieved from  
[https://www.amazon.com/dp/B07BC567TW?axitk=nCbDXWa9vGH-pT30iBfiig&pd\\_rd\\_i=B07BC567TW&pf\\_rd\\_m=ATVPDKIKX0DER&pf\\_rd\\_p=3ff6092e-8451-438b-8278-7e94064b4d42&pf\\_rd\\_s=desktop-sx-top-slot&pf\\_rd\\_t=301&pf\\_rd\\_i=Raspberry+Pi+3B%2B+Ultimate+Starter+Kit&hsa\\_cr\\_id=4057](https://www.amazon.com/dp/B07BC567TW?axitk=nCbDXWa9vGH-pT30iBfiig&pd_rd_i=B07BC567TW&pf_rd_m=ATVPDKIKX0DER&pf_rd_p=3ff6092e-8451-438b-8278-7e94064b4d42&pf_rd_s=desktop-sx-top-slot&pf_rd_t=301&pf_rd_i=Raspberry+Pi+3B%2B+Ultimate+Starter+Kit&hsa_cr_id=4057)
- Architecture of FOAM. (2018). Retrieved from  
<https://f-o-a-m.github.io/foam.developer/tutorials/architecture.html>
- Armstrong, S. (2016, 11). Move over Bitcoin, the blockchain is only just getting started. *Wired Magazine*. Retrieved from  
<https://www.wired.co.uk/article/unlock-the-blockchain>
- Asare, P., Broman, D., Lee, E. A., Prinsloo, G., Torngren, M., & Sunder, S. S. (2012). *Cyber-Physical Systems - a Concept Map*. Retrieved from  
<http://cyberphysicalsystems.org/>
- ASUS SBC Tinker board RK3288. (2017). Retrieved from [https://www.amazon.com/Tinker-board-RK3288-1-8GHz-Mali-T764/dp/B06VSBVQWS/ref=sr\\_1\\_3?tag=beebom-20&ie=UTF8&qid=1501393542&sr=8-3&keywords=Asus+Tinker+Board](https://www.amazon.com/Tinker-board-RK3288-1-8GHz-Mali-T764/dp/B06VSBVQWS/ref=sr_1_3?tag=beebom-20&ie=UTF8&qid=1501393542&sr=8-3&keywords=Asus+Tinker+Board)
- Authors, T. (2018). Tendermint Documentation. Retrieved from  
<https://media.readthedocs.org/pdf/tendermint/master/tendermint.pdf>
- Axon, L. (2015). *Privacy-awareness in blockchain-based PKI*. Retrieved from  
<http://ora.ox.ac.uk/objects/uuid:f8377b69-599b-4cae-8df0-f0cded53e63b>

Baheti, R., & Gill, H. (2011). Cyber-physical systems. In *The impact of control technology* (pp. 161–166). doi: 10.1145/1795194.1795205

Banerjee, A., Venkatasubramanian, K. K., Mukherjee, T., & Gupta, S. K. S. (2012). Ensuring safety, security, and sustainability of mission-critical cyber-physical systems. *Proceedings of the IEEE*, 100(1), 283–299. doi: 10.1109/JPROC.2011.2165689

*Bitcoin Wiki:Transaction*. (2016). Retrieved from  
[https://en.bitcoin.it/wiki/Transaction#Principle\\_example\\_of\\_a\\_Bitcoin\\_transaction\\_with\\_1\\_input\\_and\\_1\\_output\\_only](https://en.bitcoin.it/wiki/Transaction#Principle_example_of_a_Bitcoin_transaction_with_1_input_and_1_output_only)

blinke3713. (2018). *MakeFile for Foam Token Server*. Retrieved from  
<https://github.com/f-o-a-m/foam.token-server/blob/master/Makefile>

*Blockchain*. (2017). Retrieved from  
<http://www.investopedia.com/terms/b/blockchain.asp>

Blockchain Technologies. (2016). *Blockchain Mining Explained*. Retrieved from  
<http://www.blockchaintechnologies.com/blockchain-mining>

BlockGeeks. (2017). *Proof of Work vs Proof of Stake: Basic Mining Guide - Blockgeeks*. Retrieved from  
<https://blockgeeks.com/guides/proof-of-work-vs-proof-of-stake/>

Brinkmann, R. (1999). *The art and science of digital compositing*. Morgan Kaufmann. Retrieved from <https://books.google.com/books?id=DSouFSW56C4C&pg=PA184#v=onepage&q&f=true>

Buterin, & Vitalik. (2014). Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform. *Ethereum*(January), 1–36. Retrieved from  
<https://github.com/ethereum/wiki/wiki/White-Paper>

- Buterin, V. (2015). *On public and private blockchains* -. Retrieved from <https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/>
- Cam PHAM, V. (2018). *Blockchain comparison overview: EOS.IO, Casper FFG, Cardano, Tendermint and Tomochain*. Retrieved from <https://medium.com/tomochain/blockchain-comparison-overview-eos-io-casper-ffg-cardano-tendermint-and-tomochain-2b0df68806b6>
- Cárdenas, A. A., Amin, S., Sinopoli, B., Giani, A., Perrig, A., & Sastry, S. (2009). *Challenges for securing cyber physical systems* (Tech. Rep.). Berkeley: University of California, Carnegie Mellon University. Retrieved from <https://pdfs.semanticscholar.org/d514/97e5827cc00d9d00c26e27a769d42284cfba.pdf>
- Catalini, C., & Gans, J. S. (2017, 9). Some Simple Economics of the Blockchain. *SSRN Electronic Journal*. Retrieved from <http://www.ssrn.com/abstract=2874598> doi: 10.2139/ssrn.2874598
- Chadwick, C., Betzig, S., & Hu, F. (2011). Cyber-physical Systems Concepts. In *Cyber-physical systems: Integrated computing and engineering design* (pp. 3–14). Boca Raton: CRC Press. doi: 10.1145/1795194.1795205
- Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *IEEE Access*, 4, 2292–2303. doi: 10.1109/ACCESS.2016.2566339
- Ciobanu, M.-G. (2017). *Ethereum's Competition: Blockchains that Aim to Beat Ethereum*. Retrieved from <https://cryptostreet.co/cryptocurrency-news/ethereum-news/eth-competition-blockchains-beat-ethereum>
- Cuomo, J. (2013). *JavaScript Everywhere and the Three Amigos (Into the wild BLUE yonder!)*. Retrieved from [https://www.ibm.com/developerworks/community/blogs/gcuomo/entry/javascript\\_everywhere\\_and\\_the\\_three\\_amigos?lang=en](https://www.ibm.com/developerworks/community/blogs/gcuomo/entry/javascript_everywhere_and_the_three_amigos?lang=en)

*Debian – Details of package tendermint in sid.* (2018). Retrieved from

<https://packages.debian.org/sid/tendermint>

Deloitte LLP. (2016). *Blockchain: Enigma. Paradox. Opportunity.* London. Retrieved from

<https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/Innovation/deloitte-uk-blockchain-full-report.pdf>

*Download NOOBS for Raspberry Pi.* (2018). Retrieved from

<https://www.raspberrypi.org/downloads/noobs/>

Duivestijn, S., van Doorn, M., van Manen, T., Bloem, J., & van Ommeren, E. (2015). *Design to*

*Disrupt Blockchain: cryptoplatform for a frictionless economy.* Sogeti, Fr. Retrieved from

[http://labs.sogeti.com/wp-content/uploads/2015/08/D2D-3\\_EN-web.pdf](http://labs.sogeti.com/wp-content/uploads/2015/08/D2D-3_EN-web.pdf)

*ERC-20.* (2018). Retrieved from <https://en.wikipedia.org/wiki/ERC-20>

*ERC-20 Token Standard.* (2018). Retrieved from

[https://theethereum.wiki/w/index.php/ERC20\\_Token\\_Standard](https://theethereum.wiki/w/index.php/ERC20_Token_Standard)

ERC: Token standard #20. (2015). *Github.com.* Retrieved from

<https://github.com/ethereum/eips/issues/20>

*Ethereum Homestead: Choosing a client.* (2016). Retrieved from

<http://ethdocs.org/en/latest/ethereum-clients/choosing-a-client.html>

*Etherscan Token Tracker Page.* (2018). Retrieved from <https://etherscan.io/tokens>

Farwell, J. P., & Rohozinski, R. (2011, 2). Stuxnet and the future of cyber. *Survival*, 53(1),

23–40. Retrieved from

<https://www.tandfonline.com/doi/full/10.1080/00396338.2011.555586> doi:  
10.1080/00396338.2011.555586

Flavio, V. (2013). *Bitcoin A Peer-to-Peer Electronic Cash System* (Tech. Rep.). Albany, NY: The

State University of New York (SUNY) Polytechnic Institute. Retrieved from

<https://web.cs.sunyit.edu/~salerne/CS538/Papers/Bitcoin%20A%20Peer%20to%20Peer%20Electronic%20Cash%20System.pdf>

*FOAM Servant Client*. (2018). Retrieved from

<https://github.com/haskell-servant/servant/tree/master/servant-client>

*FOAM Servant Mock*. (2018). Retrieved from

<https://github.com/haskell-servant/servant-mock>

"Foamspace Corp". (2018a). *FOAM Technicial Whitepaper*. Retrieved from

<https://github.com/f-o-a-m/public-research/blob/master/FOAMTechincalWhitepaperDraft.pdf>

"Foamspace Corp". (2018b). *FOAM Whitepaper*. Retrieved from

<https://foam.space/publicAssets/FOAM.Whitepaper.pdf>

*FOAM Spatial Index*. (2018). Retrieved from <https://beta.foam.space/welcome>

Forno, R., & Joshi, A. (2016). How U.S. "Cyber Bombs" Against Terrorists Really Work.

*Scientific American*. Retrieved from <https://www.scientificamerican.com/article/how-u-s-cyber-bombs-against-terrorists-really-work/>

*The Future of Proof of Location*. (2018). Retrieved from <https://www.foam.space/>

*GitHub Foam Token Server*. (2018). Retrieved from

<https://github.com/f-o-a-m/foam.token-server>

*Glossary - Mainnet*. (2018). Retrieved from <https://www.ethnews.com/glossary/mainnet>

Goblirsch, B. (2018). *Community questions for Ryan John King, FOAM Co-founder & CEO*.

Retrieved from <https://blog.foam.space/community-questions-for-ryan-john-king-foam-co-founder-ceo-335cef4b17f8>

Graham, L. (2017). *Blockchain fork will create new digital currency called Bitcoin Cash*.

Retrieved from <https://www.cnn.com/2017/07/31/blockchain-fork-will-create-new-digital-crypto-currency-bitcoin-cash.html>

Greenberg, A. (2014, 11). Hacker Lexicon: What Is Homomorphic Encryption? — WIRED.

*Wired*. Retrieved from

<https://www.wired.com/2014/11/hacker-lexicon-homomorphic-encryption/>

Greenspan, G. (2016). *Four Genuine Blockchain Use Cases*. Retrieved from

<http://www.coindesk.com/four-genuine-blockchain-use-cases/>

Guo, M., Hu, F., & Hong, Y.-k. (2013). Cyber-Physical System Modeling on Cognitive

Unmanned Aerial Vehicle Communications. In *Cyber-physical systems: Integrated computing and engineering design* (1st ed., pp. 87–110). Boca Raton, FL: CRC Press.

Gupta, M. (2017). *Blockchain for Dummies, IBM Limited Edition* (V. Koilraj & S. Hayes, Eds.).

Hoboken, NJ: John Wiley & Sons, Inc.

Guy, J. (2016). *InterPlanetary File System (IPFS)*. Retrieved from

<http://wiki.p2pfoundation.net/IPFS>

Guy, S., Boyle, E., & Hu, F. (2013). Cyber-Physical System Security. In *Cyber-physical systems:*

*Integrated computing and engineering design* (1st ed., pp. 111–120). Boca Raton, FL:

CRC Press. Retrieved from [https://](https://www.taylorfrancis.com/ezproxy.lib.purdue.edu/books/e/9781466577015)

[www-taylorfrancis-com.ezproxy.lib.purdue.edu/books/e/9781466577015](https://www.taylorfrancis.com/ezproxy.lib.purdue.edu/books/e/9781466577015)

Iansiti, M., & Lakhani, K. R. (2017). *The Truth About Blockchain*. Retrieved from

<https://hbr.org/2017/01/the-truth-about-blockchain>

*Installing Go from source*. (2018). Retrieved from

<https://golang.org/doc/install/source#environment>



*Introducing the FOAM Protocol.* (2017). Retrieved from

<https://blog.foam.space/introducing-the-foam-protocol-2598d2f71417>

Ivezic, M. (2016). *Cybersecurity of Blockchain and Blockchain for Cybersecurity.* Retrieved from

<https://www.linkedin.com/pulse/cybersecurity-blockchain-marin-ivezic>

Josefsson, K. (2017). *Crypto-Spatial Coordinates – the open location standard on Ethereum.*

Retrieved from

<https://blog.foam.space/crypto-spatial-coordinates-fe0527816506>

Kejace. (2018). *Purescript-eth-core travis.yml file.* Retrieved from [https://www.github.com/](https://www.github.com/f-o-a-m/purescript-eth-core/blob/master/.travis.yml)

[f-o-a-m/purescript-eth-core/blob/master/.travis.yml](https://www.github.com/f-o-a-m/purescript-eth-core/blob/master/.travis.yml)

King, R. J. (2018a). *Introduction to Proof of Location.* Retrieved from

<http://blog.foam.space/introduction-to-proof-of-location-6b4c77928022>

King, R. J. (2018b). *The Spatial Index.* Retrieved from

<https://blog.foam.space/the-spatial-index-9793f42c46c8>

Konstantopoulos, G. (2017). *Understanding Blockchain Fundamentals, Part 1: Byzantine Fault Tolerance.* Retrieved from

<https://medium.com/loom-network/understanding-blockchain-fundamentals-part-1-byzantine-fault-tolerance-245f46fe8419/>

Kudasov, N. (2018). *Servant.Swagger.* Retrieved from [http://haskell-servant.github.io/](http://haskell-servant.github.io/servant-swagger/haddock/Servant-Swagger.html)

[servant-swagger/haddock/Servant-Swagger.html](http://haskell-servant.github.io/servant-swagger/haddock/Servant-Swagger.html)

Kurve, A. (2017). *10 Best Raspberry Pi 3 Alternatives You Can Buy.* Retrieved from

<https://beebom.com/best-raspberry-pi-3-alternatives/>

Lee, E. A., & Cheng, A. M. K. (2015, 2). The past, present and future of cyber-physical systems: a focus on models. *Sensors*, 15(3), 4837–4869. Retrieved from

<https://www.mdpi.com/journal/sensors> doi: 10.3390/s150304837

- Linode. (2017). *Use vmstat to Monitor System Performance*. Retrieved from <https://www.linode.com/docs/uptime/monitoring/use-vmstat-to-monitor-system-performance/>
- Lipovača, M. (2011). Introduction. In A. Staton & S. Yang (Eds.), *Learn you a haskell for great good!* (First ed., p. 400). San Francisco, CA: No Starch Press. Retrieved from <http://learnyouahaskell.com/introduction#so-whats-haskell>
- Lloyd, T. (2015). *Top 5 Cybersecurity Failures in Financial Services*. Retrieved from <http://wmtoday.com/2015/04/04/top-5-cyber-security-failures-in-financial-services/>
- Lynn, B. Y.-S. (2001). *Cryptography - Zero-Knowledge Proofs*. Retrieved from <https://crypto.stanford.edu/pbc/notes/crypto/zk.html>
- Manuel. (2018). *Raspberry Pi 3B+ and 3B in comparison*. Retrieved from <https://www.datenreise.de/en/raspberry-pi-3b-and-3b-in-comparison/>
- McKee, D. W., Clement, S. J., Almutairi, J., & Xu, J. (2017). Massive-Scale Automation in Cyber-Physical Systems: Vision & Challenges. *Proceedings - 2017 IEEE 13th International Symposium on Autonomous Decentralized Systems, ISADS 2017*, 5–11. doi: 10.1109/ISADS.2017.56
- Montalbano, E. (2017). *Six Cyber-Physical Attacks the World Could Live Without*. Retrieved from <https://securityledger.com/2017/01/six-cyber-physical-attacks-the-world-could-live-without/>
- Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System* (Tech. Rep.). Retrieved from <https://bitcoin.org/bitcoin.pdf>

- Pasqualetti, F., Dorfler, F., & Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11), 2715–2729. doi: 10.1109/TAC.2013.2266831
- Peck, M. (2016). *The Uncanny Mind That Built Ethereum*. Retrieved from <https://www.wired.com/2016/06/the-uncanny-mind-that-built-ethereum/>
- Pilkington, M. (2015). Blockchain Technology: Principles and Applications. *Research Handbook on Digital Transformations*, 1–39. Retrieved from <http://papers.ssrn.com/abstract=2662660> doi: 10.4337/9781784717766.00019
- Popper, N. (2015). *Digital Gold: Bitcoin and the Inside Story of the Misfits and Millionaires Trying to Reinvent Money* (1st ed.). New York: Harper.
- PostgreSQL 11.0 Documentation*. (2018). Retrieved from <https://www.postgresql.org/docs/11/static/index.html>
- Products Archive - Raspberry Pi*. (2015). Retrieved from <https://www.raspberrypi.org/products/>
- PureScript*. (2017). Retrieved from <http://www.purescript.org/>
- Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010). Cyber-physical systems: The next computing revolution. *47th ACM/IEEE Design Automation Conference (DAC)*, 731–736. doi: 10.1145/1837274.1837461
- Raspberry Pi 2 Model B*. (2015). Retrieved from <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>
- Raspberry Pi 3 B+ Motherboard*. (2018). Retrieved from [https://www.amazon.com/RS-Components-Raspberry-Pi-Motherboard/dp/B07BFH96M3/ref=dp\\_ob\\_title\\_ce](https://www.amazon.com/RS-Components-Raspberry-Pi-Motherboard/dp/B07BFH96M3/ref=dp_ob_title_ce)

*Raspberry Pi 3 Model B Motherboard*. (2016). Retrieved from [https://www.amazon.com/dp/B01CD5VC92/ref=nav\\_timeline\\_asin?encoding=UTF8&psc=1](https://www.amazon.com/dp/B01CD5VC92/ref=nav_timeline_asin?encoding=UTF8&psc=1)

*Raspberry Pi comparison chart*. (2018). Retrieved from <http://www.thepishop.com.au/raspberry-pi-comparison-chart>

*Raspberry Pi Power Supply*. (2013). Retrieved from <https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md>

Raval, S. (2016). *Decentralized Applications: Harnessing Bitcoin's Blockchain Technology - Siraj Raval - Google Books* (1st ed.; T. McGovern, Ed.). Boston. Retrieved from <https://books.google.com/books?id=fvywDAAAQBAJ&pg=PA1#v=onepage&q&f=false>

Raza, U., Kulkarni, P., & Sooriyabandara, M. (2017). Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys & Tutorials*, 19(2), 855–873. Retrieved from <http://ieeexplore.ieee.org/document/7815384/> doi: 10.1109/COMST.2017.2652320

Rehman, S. (2018). *Create and deploy a blockchain chat application on the IBM Cloud using Tendermint and Lotionjs*. Retrieved from <https://developer.ibm.com/tutorials/cl-create-blockchain-chat-app-tendermint-ibm-cloud/>

Rhodes, D. (2018). *What is a Cryptocurrency Mainnet?* Retrieved from <https://coincentral.com/what-is-a-mainnet/>

Samaniego, M., & Deters, R. (2017, 12). Hosting virtual IoT resources on edge-hosts with blockchain. *Proceedings - 2016 16th IEEE International Conference on Computer and Information Technology, CIT 2016, 2016 6th International Symposium on Cloud and Service Computing, IEEE SC2 2016 and 2016 International Symposium on Security and Privacy in Social Netwo*, 116–119. Retrieved from <http://ieeexplore.ieee.org/document/7876325/> doi: 10.1109/CIT.2016.71

- Sanger, D. E. (2016). *U.S. Cyberattacks Target ISIS in a New Line of Combat*. Retrieved from <https://www.nytimes.com/2016/04/25/us/politics/us-directs-cyberweapons-at-isis-for-first-time.html>
- Slay, J., & Miller, M. (2007). Lessons learned from the maroochy water breach. *Critical Infrastructure Protection*, 253, 73–82. Retrieved from [https://link.springer.com/content/pdf/10.1007%2F978-0-387-75462-8\\_6.pdf](https://link.springer.com/content/pdf/10.1007%2F978-0-387-75462-8_6.pdf) doi: 10.1007/978-0-387-75462-8{\\_}6
- Smart Contracts Definition — Investopedia*. (2017). Retrieved from <https://www.investopedia.com/terms/s/smart-contracts.asp>
- Snoyman, M. (2015). *Developing web apps with Haskell and Yesod : safety-driven web development* (2nd ed.; S. St. Laurent & A. MacDonald, Eds.). Sebastopol, CA: O'Reilly Media. Retrieved from <https://proquestcombo-safaribooksonline-com.ezproxy.lib.purdue.edu/book/web-development/9781491915585>
- Tendermint Core: Install Tendermint*. (2018). Retrieved from <https://tendermint.com/docs/introduction/install.html#compile-with-cleveldb-support>
- Tendermint core: Running in production*. (2018). Retrieved from <https://tendermint.com/docs/tendermint-core/running-in-production.html#corruption>
- Thompson, C. (2016). *The Difference Between a Private, Public, & Consortium Blockchain*. Retrieved from [http://www.blockchaindailynews.com/The-difference-between-a-Private-Public-Consortium-Blockchain\\_a24681.html](http://www.blockchaindailynews.com/The-difference-between-a-Private-Public-Consortium-Blockchain_a24681.html)
- Token Sale*. (2018). Retrieved from <https://www.foam.space/token>
- Unspent Transaction Output, UTXO - Bitcoin Glossary*. (2017). Retrieved from <https://bitcoin.org/en/glossary/unspent-transaction-output>

*What exactly is an Ethereum client and what clients are there?* (2016). Retrieved from

[https://ethereum.stackexchange.com/questions/269/  
what-exactly-is-an-ethereum-client-and-what-clients-are-there](https://ethereum.stackexchange.com/questions/269/what-exactly-is-an-ethereum-client-and-what-clients-are-there)

*What is Kubernetes?* (2018). Retrieved from

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

*What is PostgreSQL?* (2018). Retrieved from

<https://www.postgresql.org/docs/11/static/intro-what-is.html>

*What is Tendermint ?* (2018). , 1–8. Retrieved from

<https://tendermint.com/docs/introduction/what-is-tendermint.html>

Zanni, A. (2015). *Cyber-physical systems and smart cities*. Retrieved from

[https://www.ibm.com/developerworks/library/  
ba-cyber-physical-systems-and-smart-cities-iot/](https://www.ibm.com/developerworks/library/ba-cyber-physical-systems-and-smart-cities-iot/)

Zheng, Z., Xie, S., Dai, H.-N., & Wang, H. (2016). *Blockchain Challenges and Opportunities: A Survey*. Retrieved from [https://www.researchgate.net/publication/](https://www.researchgate.net/publication/319058582_Blockchain_Challenges_and_Opportunities_A_Survey)

319058582\_Blockchain\_Challenges\_and\_Opportunities\_A\_Survey

## APPENDIX A.

### A.1 Introduction

The code script that is below was executed on the Raspberry Pi 3B and the Raspberry Pi 3B+ from October 20th to November 6th. This script was ran every 5 minutes intermittently from October 20th to October 25th and it was ran constantly starting on October 26th to November 6th. There were over 90,000 lines of output data that were received from the Raspberry Pi 3B and Raspberry Pi 3B+ during this time frame.

#### A.1.1 Code that was used to collect data

Run File

```
pi3B_TnT2 |& tee -a /home/pi3B/Desktop/pi3B_Data  
pi3B+_TnT2 |& tee -a /home/pi3B+/Desktop/pi3B+_Data
```

Create File

```
sudo nano /usr/bin/pi3B_TnT2  
sudo nano /usr/bin/pi3B+_TnT2
```

Paste into Created file

```
#!/bin/sh  
  
while true  
do  
  
echo
```

```
date "+%H:%M:%S %m/%d/%y"
```

```
vcgencmd measure_temp
```

```
vcgencmd measure_volts
```

```
vmstat
```

```
ps
```

```
sleep 30
```

```
done
```

```
Ctrl 0
```

```
Ctrl X
```

```
sudo chmod a+x /usr/bin/pi3B_TnT2
```

```
sudo chmod a+x /usr/bin/pi3B+_TnT2
```