

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Trustable oracles towards trustable blockchains

Pedro Duarte da Costa

DISSERTATION PLANNING



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Filipe Figueiredo Correia

Second Supervisor: Hugo Sereno Ferreira

February 7, 2019

Trustable oracles towards trustable blockchains

Pedro Duarte da Costa

Mestrado Integrado em Engenharia Informática e Computação

February 7, 2019

Abstract

Resumo

Acknowledgements

Pedro Duarte da Costa

“If I have seen further it is by standing on ye sholders of Giants.”

Isaac Newton

Contents

1	Introduction	1
1.1	The smart contract connectivity problem	1
1.2	Smart contracts space and computation limits	2
1.3	Oracles as a solution	3
1.4	Oracles trust	3
1.5	Motivation and Objectives	3
1.6	Document Structure	4
2	State of the art	5
2.1	Academia Research	5
2.1.1	Research Questions	5
2.1.2	Review strategy and data sources	5
2.1.3	Study Selection and Quality assessment	7
2.1.4	Data extraction and synthesis	8
2.2	Non-Academia Research	8
2.2.1	Findings	9
2.3	Summary and conclusions	9
2.3.1	Data carrier oracles	9
2.3.2	Computation oracles	9
2.3.3	Conclusions	10
3	Trustable Oracles	11
3.1	Defining a trustable oracle	11
3.2	Implementation considerations	11
3.3	Summary and conclusions	11
4	Implementation	13
4.1	Summary and conclusions	13
5	Conclusions and Future Work	15
	References	17
A		19

CONTENTS

List of Figures

1.1	Smart contract connectivity problem.	2
1.2	Oracle integration.	4
2.1	Review strategy.	6
2.2	Screening stages.	8

LIST OF FIGURES

List of Tables

2.1	Search results per database	7
2.2	Search results per year	7

LIST OF TABLES


Abbreviations

SLR Systematic Literature Review

Chapter 1

Introduction

Satoshi Nakamoto's introduction of Bitcoin, in 2009 [Nak09], revolutionized money and currency, setting the first example of a digital asset which has no backing or intrinsic value and more importantly no centralized issuer or controller. In order to require no third party to verify each transaction and prevent double-spending, he introduced a distributed ledger mechanism in which transactions are recorded in an ongoing chain of hash-based proof-of-work, creating a public record that cannot be changed without redoing the proof-of-work. Nodes leave and rejoin the network and have incentives to work on the CPU intensive proof-of-work, extending the chain, and so, for as long as the majority of nodes are trustworthy, the longest and honest chain will thrive. This became known as blockchain, a tool for distributed consensus, in a byzantine fault-tolerant approach, without requiring to trust in centralized parties.

In 2015, Ethereum was launched as an alternative protocol for building decentralized applications, smart contract. Introduced as applications that run on the blockchain, smart contracts are self-verifying, self-executing and immutable contracts whose terms are directly written in lines of code which persist on the blockchain, promising to replace real world contracts. Contracts are the building blocks of our identity, economy and society. They enforce agreements between multiple parties and ensure trust in the compliance of the rules of the agreement but traditional contracts lack on automation and decentralization. Smart Contracts provide the ability to execute tamper-proof digital agreements, which are considered highly secure and highly reliable. 

1.1 The smart contract connectivity problem

The Ethereum blockchain is designed to be entirely deterministic [Gav14], meaning that if someone downloads the whole network history and replays it they should always end up with the correct state. Bearing this, smart contracts cannot directly query URLs for a certain information since everyone must be able to independently validate the outcome of running a given contract making it impossible to guarantee that everyone would retrieve the same information since the internet

Introduction

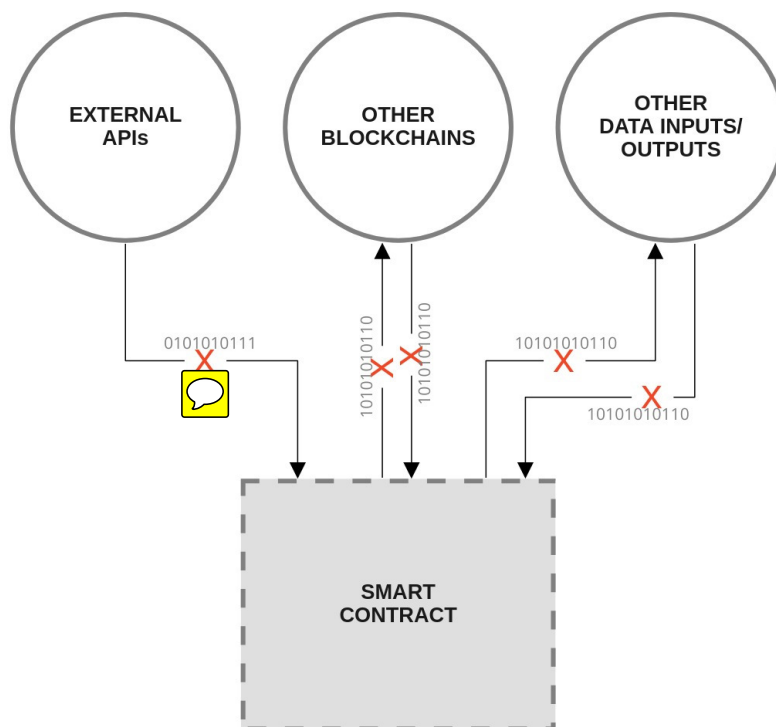


Figure 1.1: Smart contract connectivity problem.

is non-deterministic and changes over time. Determinism is necessary so that nodes can come to a consensus. In order for smart contracts to gain traction they need access information of the real world, outside of the blockchain. For example, the current price of the US dollar. However smart contracts cannot directly query the internet for information due to the non-deterministic nature of the internet. Meaning that the information retrieved at some point in time cannot be entrusted to be available or equal in another point in the future, which may result in different states when validating smart contracts by querying the internet in different moments. Oracles solve the non-deterministic problem, of querying the internet, by inputting external information on the blockchain through a transaction making sure that the blockchain contains all the information required to verify itself.

1.2 Smart contracts space and computation limits

Another problem for smart contracts, is performing long and costly operations in terms of computation and space. Several platforms are implementing smart contracts, also called DAPPs, Distributed Applications, namely Ethereum and EOS, among others.

On the Ethereum platform, smart contracts pay "Gas" to run. "Gas" is a unit that measures the amount of computation effort that certain operations require to execute. "Gas" is basically the fees

Introduction

paid to the network in order to execute an operation. Therefore, the longer the application runs the more "Gas" the smart contract has to pay.

EOS, on the opposite of Ethereum, works on an ownership model whereby users own and are entitled to use of resources in proportion to their stake. Basically, instead of paying transaction fees, the owner who holds N tokens is entitled to $N \cdot k$ transactions. While Ethereum rents out computational power on the network, EOS gives ownership of the resources in accordance to the amount of EOS held. The mentioned resources are RAM, corresponding to the used state on the network, CPU measuring the average consumption of computing resources and NET which measures used bandwidth. With increasing prices of EOS tokens, staking these resources becomes very costly.

All in all, either for users of smart contracts or the teams deploying them, keeping smart contracts efficient and performing non-costly operation is the key. Nonetheless, sometimes applications require costly operations and outsourcing them to an oracle outside of the blockchain is the answer.

1.3 Oracles as a solution

The solution to the smart contract connectivity problem and to outsourcing computation from the blockchain is the use of a secure blockchain middle-ware, mentioned before as, an oracle. Oracles, can query data from APIs, datafeeds, other blockchains or perform their own calculations and input that data on the smart contract. This way the blockchain has all the necessary information to verify the result of running a smart contract, independently of the point in time in which that verification is ran.

1.4 Oracles trust

Trust in oracles comprises two main components, service availability, trusting that the service will always return a response to our query, and untampered relay of information, meaning the information the oracle computed or queried is original and not tampered with. If the oracles are compromised we risk compromising the trust of the underlying blockchain by inputting falsified information in a system that is trusted to always have a valid state. Therefore it is imperative that oracles provide a proof for the information they provide, as well, as keep a decentralized approach by having a network of oracles always available to answer the queries.

1.5 Motivation and Objectives

The research hereby exposed was proposed by Takai, a blockchain start-up born in Porto, Portugal with the purpose to be the first blockchain open innovation platform. Sponsored by Bright Pixel, a innovation hub and venture investment house, who supports promising startups in their early years. Takai is building a platform that connects talent and entrepreneurs with the challenges of

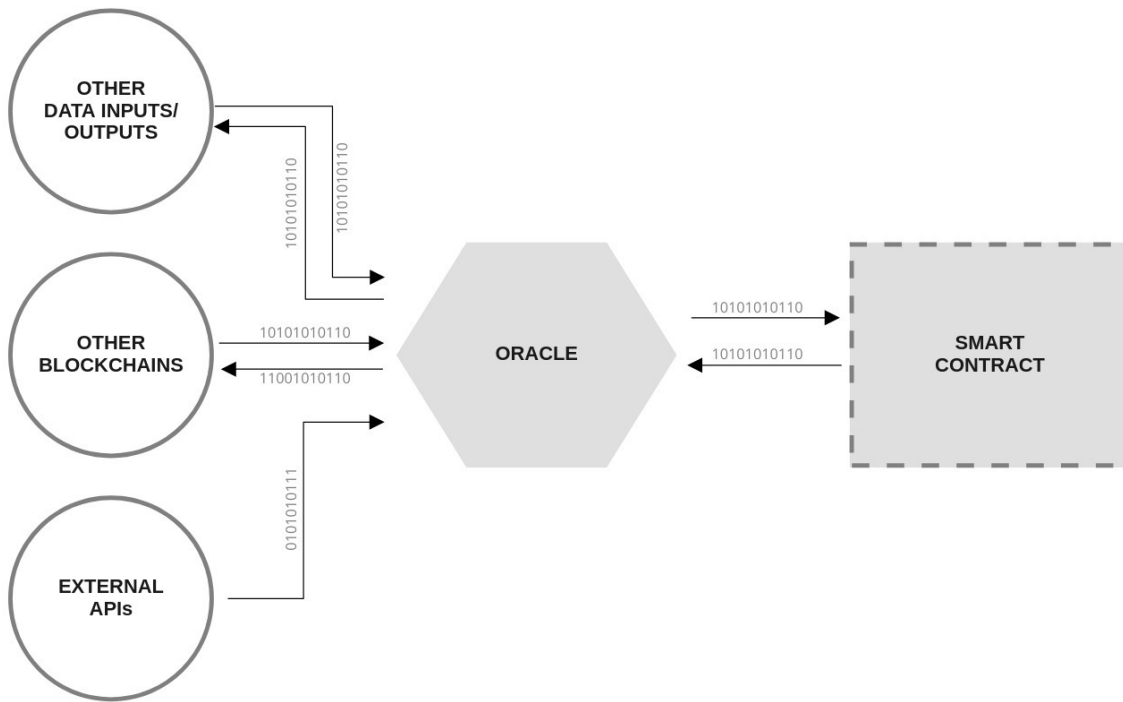


Figure 1.2: Oracle integration.

the corporate players, through the power of the **sharing economy** and blockchain trust. Taikai's project will serve as proof-of-concept for the implementation of trustable oracles **according to the principles here described.**

The growing interest in blockchain technology and specially in the potential of Smart Contracts **correlating** with the lack of research on trustable oracles creates a gap on the general adoption of blockchain by business and governments. This gap and the opportunity bestowed with the Taikai project **present a clear motivation** to pursue the construction of a bridge to overcome the oracle trust problem and further empower existing and future blockchain projects.

The proposed objectives for this work are as follows:

- Identifying the necessary components for end-to-end reliability between smart contracts and outside blockchain information;
- Providing a general framework for guaranteeing oracle trust;
- Implementing a proof-of-concept in the Taikai project on the EOS blockchain;

1.6 Document Structure

Chapter 2

State of the art

The topic of blockchain oracles is still unexplored territory mostly investigated by start-up companies and individuals thriving to solve a new problem. Therefore, research related to oracles may not yet be documented on peer-reviewed papers but, nonetheless, is invaluable in an early phase of the technology. Consequently, the state of the art cannot be complete without reviewing the research developed by the academia and also by start-ups, enterprises, governments and individuals.

2.1 Academia Research

In terms of academic research a systematic literature review was performed. It's main components and finding are described in this section. A literature review allows scholars not to step on each other's shoes but to climb on each other's shoulders, in [KKC07], meaning, not duplicating existing research but find gaps and strive to discover something new. To conduct a non-biased, methodic and reproducibile review, to the extent that a human can, it is necessary to clarify and identify at the beginning of the research its methodology, what are the data sources and what is the selection criteria.

2.1.1 Research Questions

First of all and to guide the focus of the research, the research question was defined:



- RQ: What kind of blockchain oracles have been proposed?

2.1.2 Review strategy and data sources

Figure 2.1, presents the predefined review strategy used in order to achieve such a goal and maintain unbiased, transparent and reproducibile research.

The following 5 electronic databases were used to query for such information.

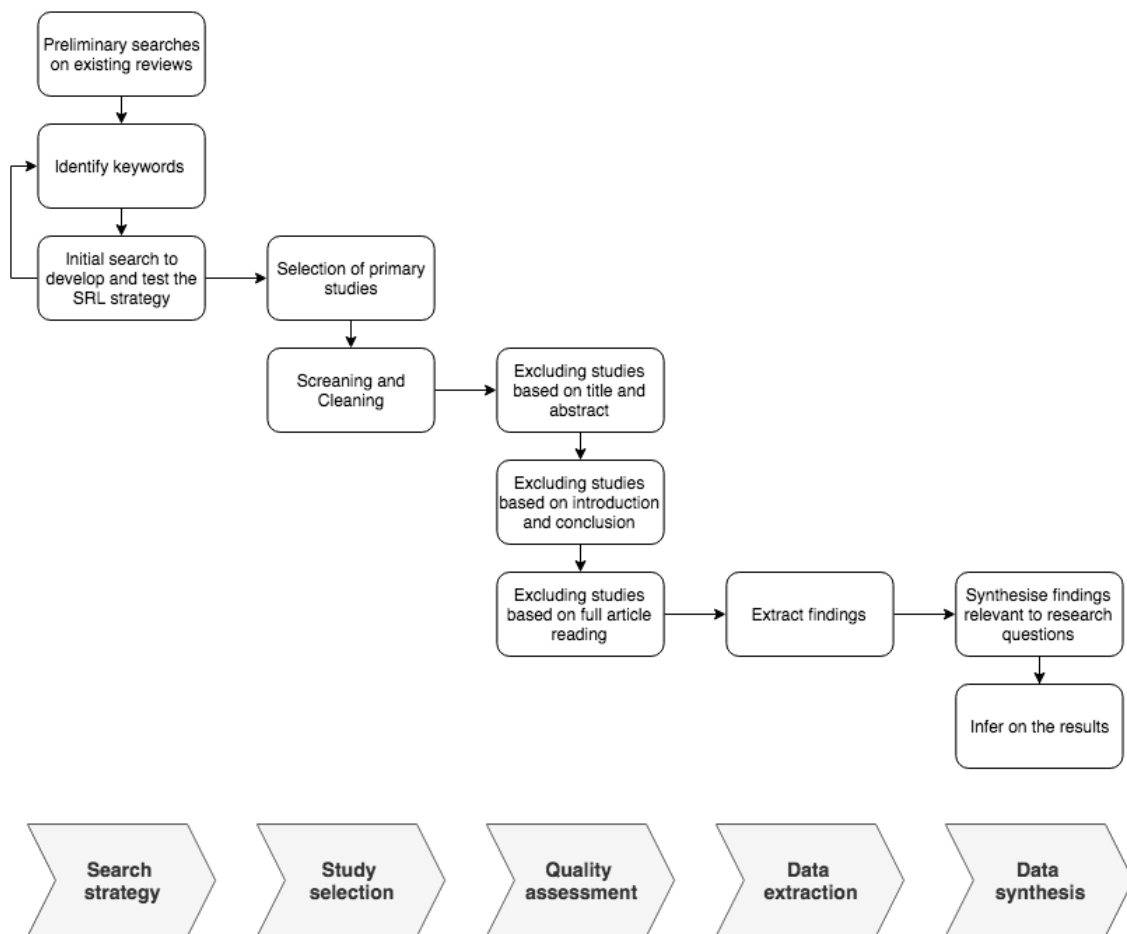


Figure 2.1: Review strategy.

- ACM Digital Library
- IEEE Xplore
- Scopus
- Google Scholar

The defined search query for the search of the relevant paper was the following:

((("blockchain" OR "block chain" OR "block-chain") AND ("oracles" OR "oracle" OR "middle-ware" OR "middleware" OR "middle ware" OR "datafeed" OR "data feed" OR "data-feed"))

The search was performed on the 5th of February 2019 and revealed the results presented in 2.1.

Database	Filters	Results
ACM Digital Library	Title, abstract and keywords	34
IEEE Xplore	Title, abstract and index terms	24
Scopus	Title, abstract and keywords	57
Google Scholar	Title	8
Total		123

Table 2.1: Search results per database

Since the concept of smart contracts on the blockchain was only introduced in 2015, with the introduction of the Ethereum blockchain, only results after 2015 were considered. Analysing the initial search results per year, in 2.2, we can infer the growing popularity of oracle related academic research.

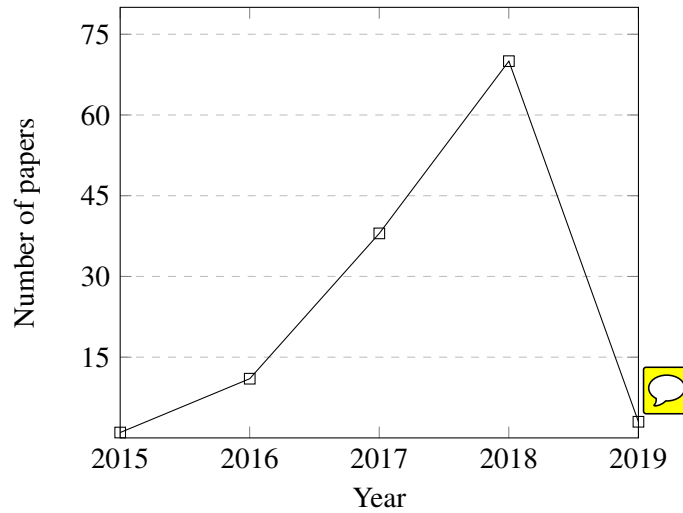


Table 2.2: Search results per year

2.1.3 Study Selection and Quality assessment

The first iteration consisted on removing all the duplicates. Second iteration, exclusion by carefully reading the title but most importantly the abstract. After this stage was finished, only 14 of the 123 papers were either describing specific trustable oracle implementations or mentioning the use of oracles. The next stage analysis the introduction and conclusions in order to remove papers which do not describe a implementation of a trustable oracle, or secure methods for the development and deployment of trustable oracles. The final step, comprises a full reading of the article to assess if the final bucket of articles answer the research question. Figure 2.2 displays the iterations described.

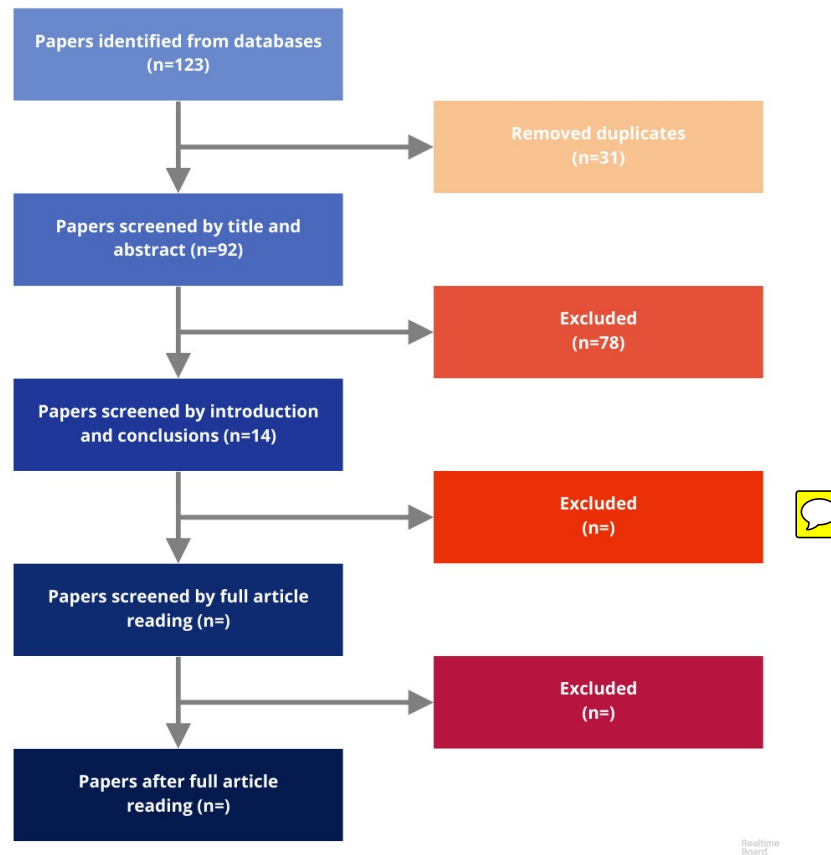


Figure 2.2: Screening stages.

2.1.4 Data extraction and synthesis

As of the moment, only two working implementations were found and are described in the 14 papers. The first and most known by the blockchain community is Town Crier [ZCC⁺16], taking advantage of trusted hardware to scrape HTTPS-enabled websites and serve source-authenticated data to the smart contracts. The second is Astraea, "a decentralized oracle based on a voting game that decides the truth or falsity of propositions." [ABV⁺18].

A more in-depth analysis will be described once the SLR is finished.

2.2 Non-Academia Research

To search for non-academia research the google search engine and medium, a platform for blog-posting used widely by developers and the start-up community, were used as a mean to find new projects. Most of these projects are documented on white-papers or on the company's website documentation page.

2.2.1 Findings

There are a few projects trying to solve the oracles trust problem. Oraclize.it provides Authenticity Proofs[[Ora18](#)] for the data it fetches guaranteeing that the original data-source is genuine and untampered and can even make use of several data sources in order to gather trustable data, but its centralized model does not guarantee an always available service. ChainLink[[EJN17](#)] and other Consensus-based oracles, such as Hivemind (previously Truthcoin[[Szt15](#)]), Augur[[PKZ⁺18](#)] and Gnosis[[20117](#)], although with varying architectures, base their data feed on "Wisdom of the Crowd" and incentives, in which participant behaviour effectively acts as the data source, and then report the result.

A more in-depth "Findings" section will be developed later once the SLR is completed.

2.3 Summary and conclusions

As far as the research has been completed, two main oracles categorizations were found,

2.3.1 Data carrier oracles

Oracles that relay query results from a trusted data source to a smart contract.

Summarizing the examples of data carrier oracles found:

- Oraclize offers a number of authenticity proof options depending on the data source being used including TLSNotary and Android remote attestation based proofs.
- TownCrier uses signed attestations by trusted hardware (specifically Intel SGX).
- Chainlink, a decentralized oracle which can be used to provide external data to smart contracts. Multiple Chainlinks to evaluate the same data before it becomes a trigger, eliminating points of failure.

2.3.2 Computation oracles

Oracles that not only relay query results, but also perform the relevant computation themselves. Computation oracles can be used as building blocks to construct off-chain computation markets.

Summarizing the examples of computation oracles found:

- SchellingCoin protocol incentivizes a decentralized network of oracles to perform computation by rewarding participants who submit results that are closest to the median of all submitted results in a commit-reveal process.
- TrueBit introduces a system of solvers and verifier. Solvers are compensated for performing computation and verifiers are compensated for detecting errors in solutions submitted by solvers.

2.3.3 **Conclusions**

The literature review points out the lack of research done by the academia in trying to solve one of the most important motives for blockchain general adoption. Only second, maybe, to scalability. The oracle trust problem, efficiently solved, opens doors to the contracts of the futures. Start-ups and sole developers are for now the main force in solving this problem which launches the challenge and motivation for the next chapter of this research.

Chapter 3

Trustable Oracles

3.1 Defining a trustable oracle

Here I will define what are the necessary components that are required in a oracle implementation to considered it as secure. This chapter will be the most important and extensive of the thesis. Followed by a proof-of-concept described in the next chapter.

3.2 Implementation considerations

3.3 Summary and conclusions

Chapter 4

Implementation

On this section, I will discuss in detail the implementation of a trustable oracle, following the standard described in the previous chapter, on the Takai project.

4.1 Summary and conclusions

Implementation

Chapter 5

Conclusions and Future Work

The oracle trust problem, is still rather recent, having emerged in 2015 with the deployment of smart contracts on the Ethereum blockchain. The systematic literature review, as far as it has been performed, and the non-academia research review reveal that most of the research and development is being done by the growing and excited blockchain community. Mostly by startups and single interested researchers.

Solving the trust problem and creating a standard for a secure middle-ware between blockchains and outside world information and application provides limitless range of future applications in terms of contracts. Creating, therefore, the grounds and the motivation for the work that will be developed on this thesis.

I hope that the work that will be developed in investigating the necessary requirements and research how oracle trust can be achieved allied with a proof-of-concept implementation on the Taikai projects will solidify the academia position on newly and ground breaking technologies such as the blockchain.

Conclusions and Future Work

References

- [20117] Gnosis Whitepaper. Technical report, Gnosis, 2017.
- [ABV⁺18] John Adler, Ryan Berryhill, Andreas Veneris, Zissis Poulos, Neil Veira, and Anastasia Kastania. ASTRAEA: A Decentralized Blockchain Oracle. Technical report, 2018.
- [EJN17] Steve Ellis, Ari Juels, and Sergey Nazarov. ChainLink A Decentralized Oracle Network. Technical report, 2017.
- [Gav14] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. Technical report, Ethereum, 2014.
- [KKC07] B. Kitchenham, B. Kitchenham, and S Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. 2007.
- [Nak09] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical report, Bitcoin, 2009.
- [Ora18] Oraclize.it. Oraclize Documentation, 2018.
- [PKZ⁺18] Jack Peterson, Joseph Krug, Micah Zoltu, Austin K Williams, and Stephanie Alexander. Augur: a Decentralized Oracle and Prediction Market Platform. Technical report, 2018.
- [Szt15] Paul Sztorc. Truthcoin Peer-to-Peer Oracle System and Prediction Marketplace. Technical report, 2015.
- [ZCC⁺16] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. Town Crier: An Authenticated Data Feed for Smart Contracts. Technical report, 2016.

REFERENCES

Appendix A