

Faculty of Engineering, University of Porto
BRPX - Bright Development Studio, S.A.



Trustable Oracles Towards Trustable Blockchains

Pedro Duarte da Costa
pedro.duarte costa@fe.up.pt

Supervisors

Filipe Figueiredo Correia -
Hugo Sereno Ferreira -

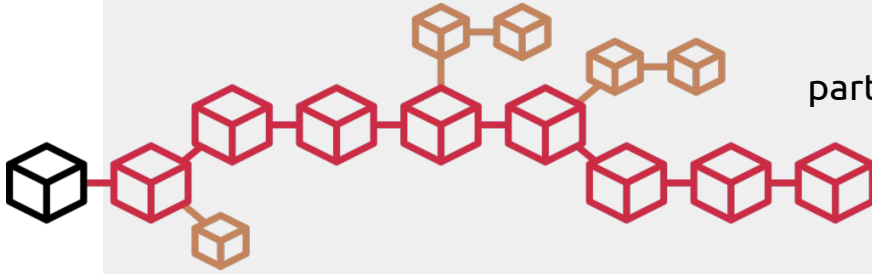
Proponent

Mário Ribeiro Alves -

Context

Blockchain

A distributed ledger, shared by untrusted participants, with strong guarantees about accuracy and consistency.



Smart Contracts

Applications that run on the blockchain.

Are self-verifying, self-executing and immutable.

The terms are directly written in lines of code which persist on the blockchain

The Smart Contract Connectivity Problem

The blockchain is deterministic. Meaning that verified in different points in time it always has the same state.

But the internet is non-deterministic, thus contracts cannot directly query the internet.

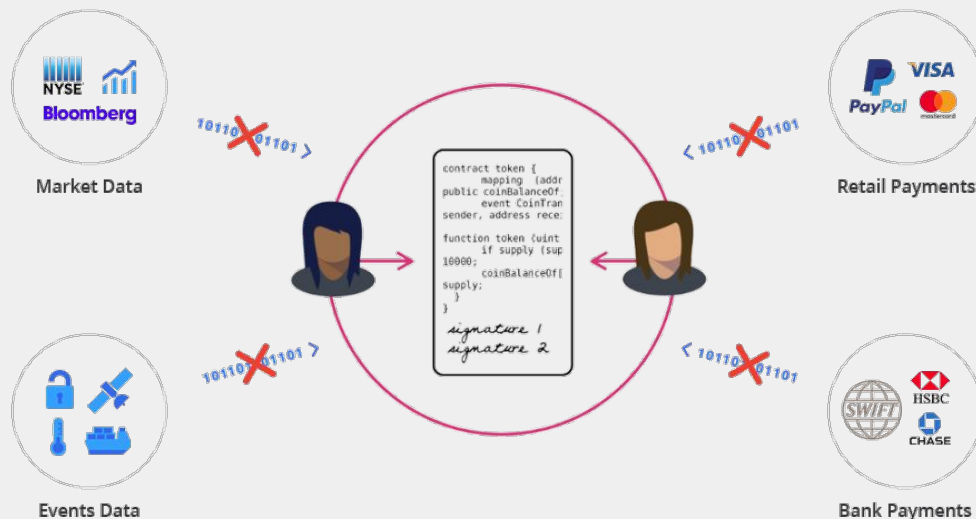


Image from <https://chain.link/>

Smart Contracts

Space and Computation limits

- On the Ethereum platform, developers pay “Gas” for the amount of computation the contract requires.
- On EOS, the deployer of the contract must hold the amount of EOS necessary for the computation, memory and bandwidth required by the contract.

Smart Contracts

Space and Computation limits

Example:

Adding two numbers together 1 million times.

Ethereum:

It costs around 0.09 ETH or \$9,61 (02/02/2019)^[1]

Amazon EC2 Instance:

In python takes around 0.04 seconds. Amazon charges \$0.0057/hour for their cheapest EC2 instance—t3.nano. This costs \$0,000001583/second or \$0,000000063 for the operation.

Oracles

Oracles run the smart contracts queries and input the answer in the blockchain.

The problem? We must trust oracles to be reliable and honest.

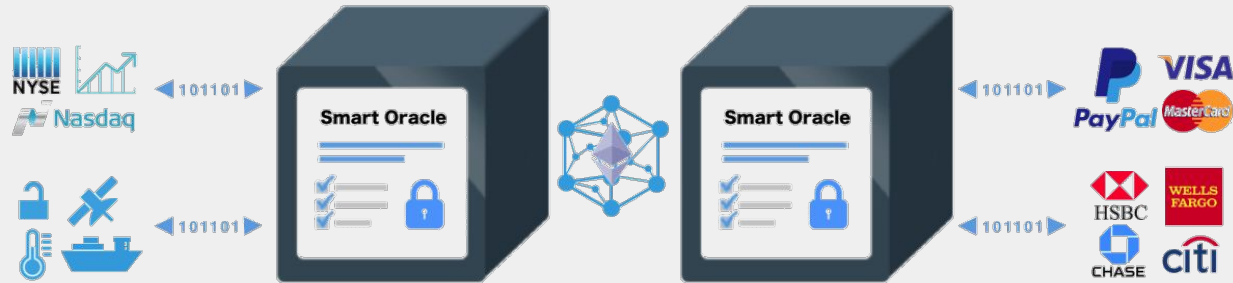
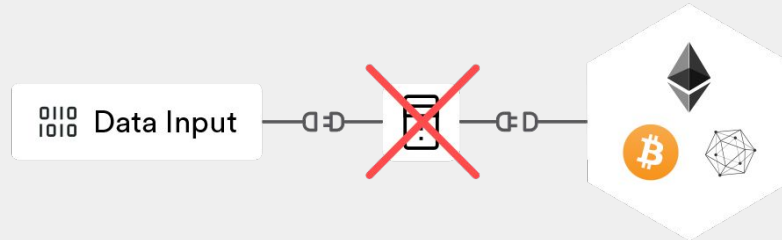


Image from <https://chain.link/>

Achieving Oracle Trust

Availability

Using one oracle can be a single point of failure.



Achieving Oracle Trust

Availability

- Achieved by using a network of oracles

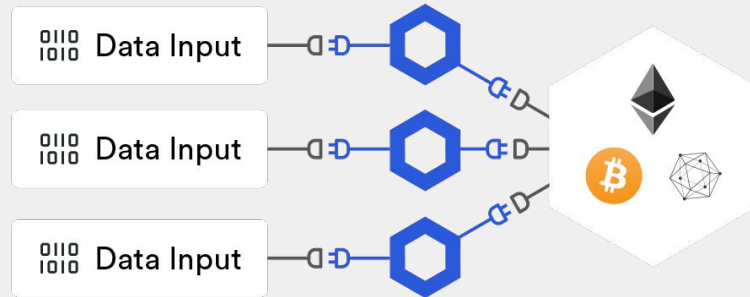


Image from <https://chain.link/>

Achieving Oracle Trust

Availability

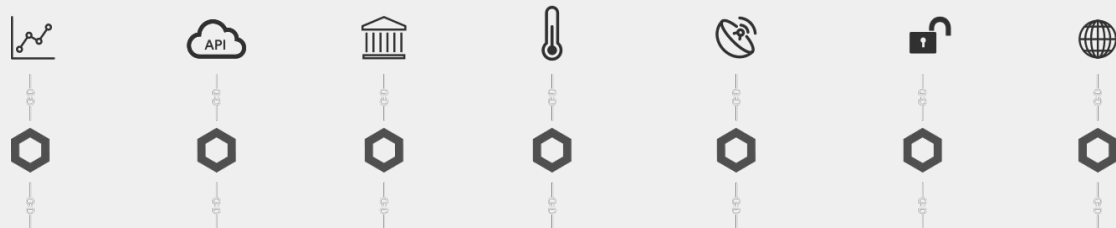
- Achieved by using a network of oracles

Resistance to tampering

- Providing data authenticity proofs
- Through oracle consensus

Challenges

The ability of clients to not only retrieve the result and proof of correctness for an outsourced operation, but to also verify the proof of correctness with less computational power than is required by the actual operation.



State of the Art

Data Carrier Oracles

Relay query results from a trusted data source to a smart contract. Commonly use cryptographic proofs to provide data authenticity guarantees.

Examples:

- **Oraclize**^[2] offers a number of authenticity proof options depending on the data source being used including TLSNotary and Android remote attestation based proofs.
- **TownCrier**^[3] uses signed attestations by trusted hardware (specifically Intel SGX).
- **Chainlink**^[4] 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.

State of the Art

Computation Oracles

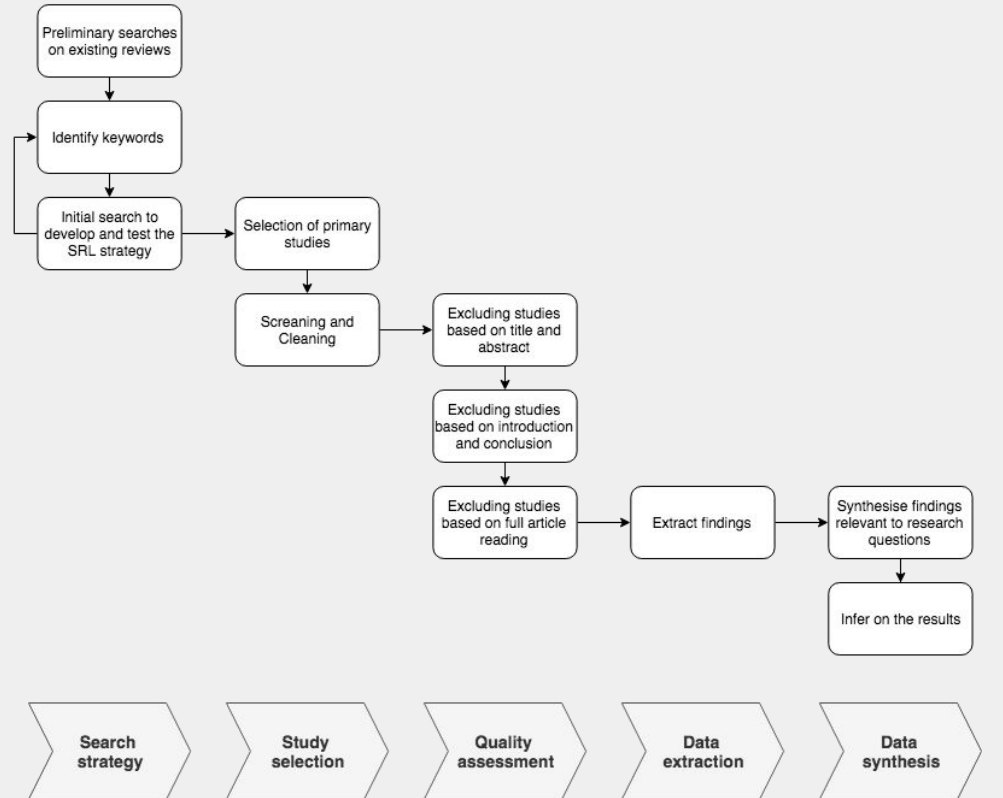
Not only relay query results, but also perform relevant computation.

Examples:

- **SchellingCoin**^[5] 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**^[6] 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.

State of the Art

A **Systematic Literature Review** is under development to further search for academia work on trustable blockchain oracles.



Motivation

The lack of academia research of defining a standard for trustable oracles as well as the growing interest from business and governments in the use of blockchain and smart contracts which need oracles to become feasible.

This gap and the opportunity bestowed with the Taikai project are the moving forces behind the work committed to this thesis.

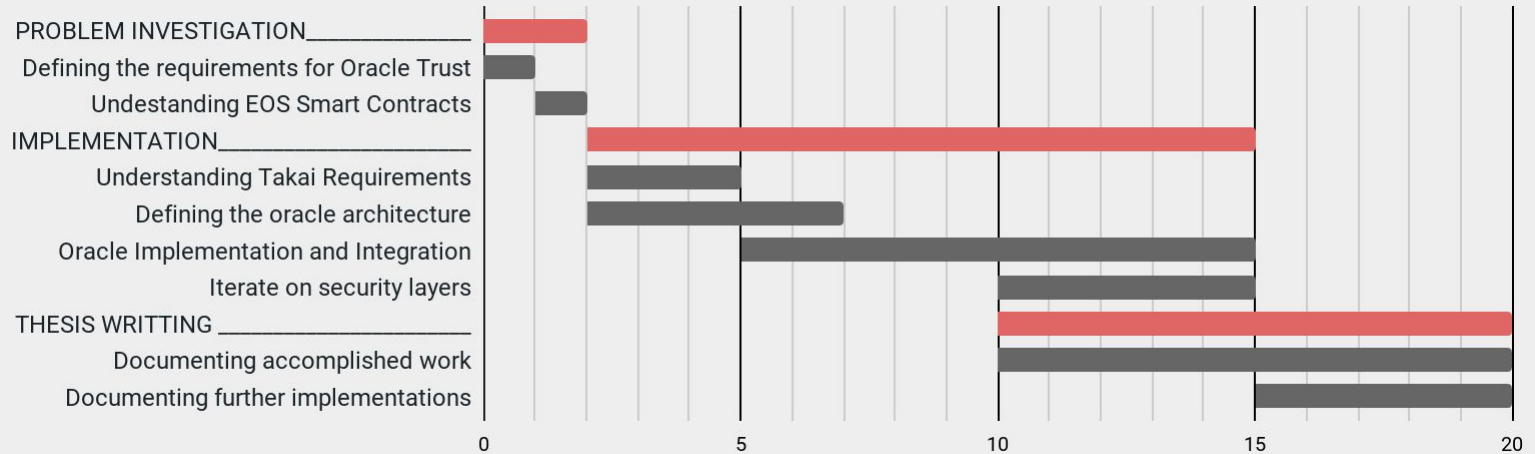


Image from <https://taikai.network/>

Objectives & Expected Results

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

Objectives & Expected Results



Work Validation

By the end of the implementation period, the following requirements should be implemented in order to solve the problem stated:

- A distributed and always available oracle network, guaranteeing service availability
- At least one method which should present sufficient proof of oracle good behaviour

References

- [1] Calculating Costs in Ethereum Contracts, Danny Ryan, <https://hackernoon.com/ether-purchase-power-df40a38c5a2f>
- [2] Oraclize.it. Oraclize Documentation, 2018
- [3] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. Town Crier: An Authenticated Data Feed for Smart Contracts. Technical report, 2016.
- [4] Steve Ellis, Ari Juels, and Sergey Nazarov. ChainLink A Decentralized Oracle Net-work. Technical report, 2017.
- [5] Vitalik Buterin. SchellingCoin: A Minimal-Trust Universal Data Feed, 2014.
- [6] Jason Teutsch and Christian Reitwießner. A scalable verification solution for blockchains.

Thanks

`pedro.duarte costa@fe.up.pt`

Faculty of Engineering, University of Porto
BRPX - Bright Development Studio, S.A.