TRUSTABLE ORACLES TOWARDS TRUSTABLE BLOCKCHAINS

Pedro Duarte da Costa

Supervisor: *Filipe Figueiredo Correia*
Co-Supervisor: *Hugo Sereno Ferreira*
Institution: *Faculty of Engineering of the University of Porto*
Course: *Integrated Master in Informatics and Computing Engineering*

Keywords Blockchain, Oracles, Trusted Computation.

CCS Categories Computer systems organization - Architectures - Distributed architectures - Peer-to-peer architectures, Security and privacy - Formal methods and theory of security - Trust frameworks

1. Abstract

The Blockchain concept was proposed as a way of processing and recording financial transactions in a peer-to-peer network while avoiding the double-spending problem and without requiring any centralized authority. Later, smart contracts were introduced as immutable applications whose terms are directly written in lines of code that are persisted and run on the Blockchain.

However, currently, smart contracts lack an important feature: internet connectivity. Due to the deterministic nature of Blockchain and the incompatible indeterministic nature of the Web, smart contracts cannot directly query it.

Oracles solve the connectivity problem, by listening to events produced by smart contracts, they can insert the needed information on the Blockchain to later be used by the contracts. But oracles do not abide by the same rules and do not support the same guarantees given by Blockchain, so they must either be trusted without hard guarantees about the truthfulness of the data that they provide or we must find ways of guaranteeing their honesty.

In order to find out how blockchain oracles are being designed, a systematic literature review was performed. This review produced fewer oracle solutions than expected, and, therefore, the author also searched for projects created by the industry. Existing oracle solutions rely on two main solutions: authenticity proofs, which are cryptographic proofs that something actually happened, or wisdom-of-the-crowd solutions based on incentives and penalising bad behaviour.

Bearing this in mind, this dissertation contribution is threefold.

Firstly, it analyses and summarises the existing authenticity proofs and mechanisms for guaranteeing oracle trust, allowing a smart contract developer to be fully informed of the implications of using each proof and their limitations.

Secondly, it defines four possible architectures for oracle design and how each of them addresses different points of trust in the oracle. Starting from the use of a third-party the author identifies and describes two architectures: *Oracle-as-a-Service w/Single Data Feed* and *Oracle-as-a-service w/Multiple Data Feeds*. Then the author describes a self-hosted approach with another two architectures: *Single-Party Self Hosted Oracle* and *Multi-Party Self Hosted Oracle*. This way, the author creates a guide that can support those creating blockchain oracles in thinking about the many limitations, trade-offs and possibilities that are inherent to the design of such kinds of system.

Finally, as the author worked on each architecture he found existing oracle solutions that would fit in the first two, regarding third-party providers, and later a new project focusing on the development of the third architecture. Regarding this, the author decided to implement the fourth architecture, both because no existing solution was available and because doing so would demonstrate the viability of this new architecture.

In conclusion, this dissertation paves the way for oracle development and research summarising firstly in a broad sense existing solutions and later contributing with a systematic set of architectures and a solution that can be easily adopted by teams interested in deploying their own oracle to achieve higher standards of trust.