# Elastic Coin: A Decentralized, Crypto Currency Driven Market for Computational Resources

Evil-Knievel and many others

## ABSTRACT

Elastic Coin provides the infrastructure for a decentralized and distributed computation of arbitrary tasks over the internet. In this context, Elastic Coin is built on-top of a crypto currency and provides a market-based mechanism to buy and sell computational resources. Buyers, those who need computational resources, model their problem using Elastic Coin's software development kit and broadcast it, along with a certain amount of ELC coins, to the network. The so-called miners are then motivated to offer their computational resources in exchange for a portion of those ELC coins. The size of this portion depends on the amount of work a miner has contributed in relation to the rest of the network. Using ELC as the driving force, Elastic Coin offers potential buyers a large parallel computation cluster composed of many CPUs, GPUs, FPGAs and other devices supplied by the miners. All at a fair and market-driven price.

## 1. INTRODUCTION

The number of internet users has increased more than tenfold from 1999 to 2016. With an increasing number of users, the number of devices—each equipped with a processor—that is connected to the Internet increases as well. As most of these users are within the scope of non-techincal and normal user activity, it can be assumed that, at any given moment, most of these devices are idle. It is not a new idea to bundle these idle processors and distribute the computational resources among various problems that require large computational power [1, 2]. In fact, this movement has enabled previously infeasible research to be accomplished. Furthermore, with the growing adoption of crypto currencies such as Bitcoin [6], where solving puzzles that are intentionally designed to be resource-intensive is rewarded with a certain number of coins, *supercomputing* becomes more and more relevant to the regular user.

Because each of these devices is owned by a different entity with different motivations and goals, there is no a-priori motivation for cooperation. Current approaches such as [1, 2] rely on the enthusiasm that users may develop for one or more interesting projects and so voluntarily share their computational resources with them. However, this scheme does not work well (efficently) in general. More precisely, there must be some kind of motivation for the ordinary user to contribute their computational resources to a global supercomputer that solves arbitrary tasks that do not necessarily pursue their own particular interests.

In this paper, we suggest a system that is built around a crypto currency termed *Elastic Coin* and addresses these difficulties. More precisely, Elastic Coin constitutes a market that matches buyers and sellers of computational resources according to economic criteria. Buyers, in this context, are those who demand computational resources in order to solve some arbitrary and computationally intensive task. They submit their task to the network and attach a certain amount of ELC—that is the currency of Elastic Coin—to it. Then, sellers, in the remainder of the paper termed *miners*, are then motivated to offer their computational resources for solving these tasks in exchange for a portion of the attached ELC. The size of this portion depends on the amount of work a miner has contributed in relation to the rest of the network. All this happens "behind the scenes" in a process termed *mining* without the requirement of manual intervention.

The remainder of this paper is structured as follows: Section 2 presents the general idea of a decentralized and market-based infrastructure for decentralized and distributed computation of arbitrary tasks over the internet. Section 3 and Section 4 discuss how these ideas can be embedded into the context of crypto currencies where security and reliability require special attention. Section 5 concludes the paper.

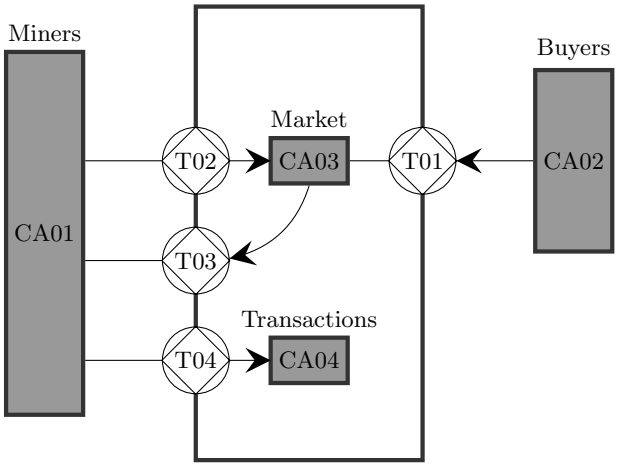
## 2. DESIGN (PENDING WORK)

### 2.1 Big Picture

See Figure 2.1.

### 2.2 Mining and the Faster Algorithm Attack

The security of Bitcoin relies on the distributed consensus achieved by a process termed *mining*. In the original Bitcoin scheme, *miners* essentially are users who spend a large amount of computational resources to solve a certain type of cryptographic puzzle [6]. The solutions to these cryptographic puzzles allow for the generation of so-called *blocks*. Users, in this context, are encouraged to mine blocks by rewarding each block with a payment of currently 25 BTC. Blocks are ordered in a linked list and both verify the cor-




| TID | Transaction Description |
|---|---|
| T01 | Submits work to market to be solved by the network |
| T02 | Proof-of-work issued periodically by miners |
| T03 | Rewarding the miners proportionally to the work done |
| T04 | Regular transaction (such as sending ELC) |

**Figure 1: Global ATD**

rectness of bitcoin transactions and form a consensus in terms of a history of transactions that all participants agree upon. In Bitcoin this history is termed *Blockchain* and aims for preventing fraudulent behavior such as double spending a coin multiple times [5] or charging back already done transactions. More precisely, this history is defined to be the *longest chain* of blocks from the genesis block (that is, the first block) to the current block. Of course, it may happen here that multiple side chains emerge from one block of which the longest eventually survives reverting the transactions included in the shorter. In order to be on the safe sides in terms of random reorganizations, users are suggested to wait until the transaction has been added to a block and $z$ blocks have been linked to it [6]. That means, the transaction has received $z+1$ confirmations. The only way for an attacker to remove one of his transactions from the longest chain in order to take back money he recently spent is to create an alternative longest chain himself that does not contain the particular transaction. In this context the attacker would start calculating an alternative chain from a point where he still owned the spent coins (in this case more than $z+1$ behind the current block). In order to be successful, he must (on average) calculate blocks faster than the rest of the network working on the original longest chain. More formally, the probability of an attacker catching up is analogous to the Gambler's Ruin problem [4, 6]. That is, let $q$ be the probability the attacker finds the next block and $p$ denote the probability that the rest of the network finds the next block.

$$q_z = \begin{cases} 1 & \text{for } p < q \\ (\frac{q}{p})^z & \text{for } p \geq q \end{cases}$$

denotes the probability that an attacker can catch up building an alternative chain from $z$ blocks behind. As the probability of finding a block increases linearly with the calculation power invested in solving the cryptographic puzzles, it can be assumed that an attacker can successfully perform this attack (in literature referred to as the 51 % attack [3, 7]) once he controls more than 50 % of the entire network's calculation power. Putting aside large mining pools, it is not economical from an attacker's point of view to acquire more than 50 % of the networks calculation power. In this paper we suggest a crypto currency-driven market for computational resources. As described in Section 1 the key idea is to working towards solving arbitrary computationally expensive tasks and in exchange get rewarded with a certain amount of ELC. This scheme is very similar to the Bitcoin mining process. An obvious and appealing idea is to utilize the immense amount of work put into solving these arbitrary tasks and use it to form the consensus, i.e., ensure the security of the blockchain, itself. In this particular case, however, the 51 % attack could be pulled off by an attacker with considerably less resources than required in the case of Bitcoin. This attack has been identified and thoroughly discussed by members of the Elastic Coin community and called *The Faster Algorithm Attack* (FAA). Without loss of generality, imagine an attacker hand crafts an algorithm with a complexity of $\mathcal{O}(P)$ with $P \in \Omega(N)$ bounded below by $N$ asymptotically. Furthermore, assume he knows an alternative algorithm which produces the same results but can be run in $\mathcal{O}(Z)$ with $Z \in \mathrm{o}(N)$ being dominated by $N$ asymptotically. When such an algorithm is submitted to the Elastic Coin network to be solved by the miners, the attacker needs significantly less resources to have the equivalent of 50 % of the networks calculation power. More precisely, he needs only $O(\frac{P}{Z})$ of the network's calculation power. To give a concrete example: imagine an attacker submits an algorithm with a complexity of $\mathcal{O}(2^N)$ while at the same time he knows an algorithm producing the same output with a complexity of $O(1)$. He then only needs $O(\frac{1}{2^N})$ of the entire network's calculation power to successfully perform the 51 % attack. In the remainder of the paper we will show how this attack can be mitigated.

## Acknowledgement

I would like to express my sincere gratitude to everyone who has contributed to this paper with his encouragement and consistent involvement. This includes bitcointalk members ...

## 3. REFERENCES

[1] D. P. Anderson. Boinc: A system for public-resource computing and storage. In *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*, pages 4–10. IEEE, 2004.

[2] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. Seti@home: an experiment in public-resource computing. *Communications of the ACM*, 45(11):56–61, 2002.

[3] I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.

[4] W. Feller. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 2008.

[5] G. O. Karame, E. Androulaki, and S. Capkun. Double-spending fast payments in bitcoin. In *Proceedings of the 2012 ACM conference on Computer*

*and communications security*, pages 906–917. ACM, 2012.

[6] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.

[7] M. Vasek, M. Thornton, and T. Moore. Empirical analysis of denial-of-service attacks in the bitcoin ecosystem. In *Financial cryptography and data security*, pages 57–71. Springer, 2014.