Elastos-coin(Ela)

Elastos.org

Elastos Coin (ELA) will be a virtual currency marketed in conjunction with, and optimized to work with, the [Elastos secure operating system](http://learningchain.cn/read/elastos) invented by Rong [Chen](http://learningchain.cn/read/chen-rong) . The BTC community of the PRC have already expressed hope and support for the Elastos family of developments, there are some high expectations in store for ELA. In particular, the currency and its associated technical system must satisfy the dual goals of somehow developing DApp while also working alongside BTC, serving to bolster and to stabilize it.   
  
The currency is set to benefit from certain abstract reconsiderations in the design of virtual currencies, such as have been imagined by Dr Gang Xu, and are slated to be further explored along the course of the project, as part of its overlapping phases for development and research. In order to secure the timely delivery of a usable product, such as will be deployed both to server farms under the American company Yuan Engineering or under the management of Feng Han, as well as to interested large-scale commercial parties, it is necessary to focus the development team's efforts upon a clean, well defined, well tested core program written in C, which is targeted towards the platform of choice among the entrepreneurial and mining community of the PRC, namely, Windows AMD64 (also known as sixty-four bit Windows), among others.   
  
This core program will access a modular framework whose policies may be implemented in a language agnostic manner, such as with the help of mixed compilation technologies, in order to experiment with different concepts accrued from the research section of this project, which as mentioned above, will be overlapping with the development portion and largely instigated by a separate team. This team may be led, or at least closely advised, by Dr Xu, in its efforts to specify, and to verify, well defined mathematical functions or subroutines, such as may constitute an implementation of: smart contracts, side-chains, NAT hole punching, internet filtration avoidance, inflationary economy, founder's reward mechanisms, merge-mining, quasi-hard-fork, and other sub-goals which may be raised by interested parties during the project's course.  
  
The schedule of the project as a whole, which is to say, its time-table for public deployment, and associated efforts on the marketing front, has not been exactly specified, owing to ongoing discussions and the open-ended scope of features. However, every attempt will be made from the technical side, led by Yuan Xun, to have 1) a deployable product no later than 25 August 2017 with all of the basic features implemented, 2) a testbed for merge-mining, and a testnet, active by no later than 11 August 2017, 3) executables suitable for regression testing no later than 18 August 2017. Every attempt will be made to neatly modularize the sub-goals mentioned above, such as UDP hole punching and support for side-chains, so that these features may be cleanly deferred as required by schedule, and in accordance with the order of priorities of the marketing campaign.   
  
Ideally, every sub-goal should be modularized to the extent that the entry of new teams possessing minimal communication with existing teams may be dispatched to work on sub-goals individually. In fact, seeing that the project at hand is a community-based venture, and that the PRC is a nation renowned for the breadth of its human resources, and not to say the least, human merits, it would make sense that we are working to create the world's first truly modular virtual currency platform, one which is all-at-once accessible to the masses by means of its transparency and wide-spread support, and which has no need of the knowledge-control of a particular interest group to continue to function and to thrive, an area where BTC and ETH have sadly both fallen short. With the ubiquity and safety of Elastos the operating system, combined with the modularity, scalability, and correctness inherent in the design of ELA, the road may finally be paved for the technical and financial community of the PRC to unleash, in the full view of the world, and to their amazement, the next generation of advancements foretelling the smart economy's arrival.   
  
## Explication   
  
The sole viable implementation of Bitcoin, the first version of which was written by Nakamoto Satoshi, suffers from an absence of conciseness and atomicity, which hinders its portability between platforms. At the time of writing of the first version, there had existed an acceptable reason to some extent. The concept of the block-chain and of digital currencies in general were not yet known to the wider public. If the implementation were not to squeeze a range of predictably desired features into a unitary package, then early users may have had difficulty understanding or accepting the concept as a whole.   
  
However, as the concept of the block-chain has matured, there is no more reason to complexify the code by bundling together numerous features unrelated to the full-node implementation's core reason of existence, which is to store and verify the block-chain in the most efficient manner. The full-node implementation should concentrate on the two areas of systemic function which it primarily has access to, namely, control of the network and of the file-system. Both of these require relatively complex routines which would benefit from simplification and from formal verification of the states and substates of operating-system services between crucial code passages, repetitions, and invocations.   
  
The program ought to behave as deterministically as possible. To that effect, it should not engage in techniques which obscure its visible operation, such as reliance upon environment variables which are not easily accounted for in preliminary calculations of the deploymen process, or upon configuration files whose syntaxes and range of options may differ, often in subtle ways as observed in practice, across the span of an upgrade. Instead, all settings must be specified by command line arguments. In practice, the invocation of the program may be simplified by the use of aliases, prior environment variables (which are parsed before invocation on account of rules provided by the shell, as opposed to afterward), shell-scripts which are saved to a file and run in batch-mode, and last of all, customzied shells or log-on routines which would be suitable for high-performance environments and embedded environments.   
  
The last means is particularly important because Elastos Coin is the companion to the operating system Elastos as developed by Chen Rong. Since this operating system is intended to run on a variety of mobile and embedded systems, it cannot suffer from the same design flaws of bitcoind(1) which would compromise its use in deployment environments of low supervision, such as the passive, semi-automated or fully automated execution and maintenance of a node within a television box (also known as set-top box) or an automotive internet-of-things (IoT) module. The hard drive requirement of bitcoind(1) makes it particularly unsuited for use upon smaller files. In contrast, Elastos Coin should be usable even in devices with limited hard disk, such as Android phones or the coming equivalent system by Chen Rong, phones based upon Elastos.   
  
## Implementation Details   
  
The entire program will be statically linked. It will make use of the following libraries: libtomcrypt(7), libtommath(7), tomsfastmath(7), base58(7), sha3(7), and musl(7). In particular, it will not use glibc(7). This is because glibc(7) is not suitable for static linking or for use in embedded environments. This will represent a major improvement in speed over bitcoind(1) in terms of start-up and also routine operation, as well as a shrinking of the overall executable size.   
  
There will also be a static script interpreter which takes policy statements using a quasi-object-oriented system and turns them into statically linked C functions with reflective attribute indicators for type information and usage of system-calls. That way, it will be possible during future revisions of the protocol to specify the new set of policies using a revised syntax reminiscent of denotational semantics or of simplified or orthogonal scripting languages, such as JS. This is the meaning of the portable policy framework.   
  
As for merge-mining, also known as equivocation, there will be Protocol B entry-points and special rules composed of extensions of the binary block-format in order to validate blocks of the auxiliary type. An open area of concern is the relationship between directly mined blocks and merge-mined blocks. As a result of certain considerations of economics, and the strategies of interested parties, it may be desirable to ban direct mining in favor of exclusive merge-mining. However, this will compromise the use of Elastos Coin as an alternative infrastructure in the event that the BTC block-chain should fork into several competing block-chains, especially where each of them suffers from subtle errors caused by similarities in protocol, as evidenced by possible replay attacks, and to a lesser extent, transaction malleability attacks. The two may even be combined. A single transaction may be malleated and then replayed, slowing down its detection among a wider wave of attacks by a hostile hybrid-node-operator until it is too late. Notwithstanding the difficulty of such an attack, the important thing is that an attack of such sort can always happen because nothing in the protocol prevents its consistent execution on a logical level of consideration.   
  
## Amazing Eight   
  
The implementation will be framed by four major goals and four minor goals.   
  
The first major goal is equivocation. It means that ELA nodes accept not only ELA proofs-of-work, but also BTC proofs-of-work. Thus, if a miner performs the proof-of-work of BTC, it is recognized and rewarded by both BTC and ELA. Using this feature, ELA can benefit from the increased mining competition, and therefore the increased security, of the BTC blockchain. The miners of BTC will submit their proofs-of-work to both BTC and ELA, thus enjoying a dual reward. The development team of ELA will work closely with miners from the BTC community to ensure concise and accurate integration of ELA rewards into their respective mining workflows. In a similar fashion to Dogecoin, there will be a public-facing function in Protocol B known as "Get Aux Work" which retrieves a magic string (from any ELA node) to be written into a BTC block. Once the BTC block is mined, which is to say, the proof-of-work completed, the BTC block is submitted to both BTC and to ELA. Thus, on both sides, a reward will be dealt.   
  
The second major goal is inheritance. It means that all people who possessed BTC before a given cut-off date will be in possession of an equal number of ELA, at no cost to themselves. There will be a one-time process necessary per UTXO to retrieve these balances. The verification will be done in SPV fashion, to avoid requiring ELA nodes to store the BTC blockchain directly, which would be a burden upon storage space. By providing this bonus for BTC possessors, the ELA community can bolster its level of initial participation and better served the shared interests established by BTC in the era preceeding.   
  
The third major goal is gratuity. It means that at ELA's inception, there will be a relatively generous one-time reward of 16M or 32M units bestown to the development team of ELA, led by Chen Rong. These funds will be stored in a special account similar to DASH's development fund, and will be used to support ELA's further revision, expansion, and technical support toward its users.   
  
The fourth major goal is inflation. For the sake of the continual maintenance of the community, there ought to be an inflationary economy which can be used to compensate miners in addition to mere transaction fees. This stream of income can also be used for further development and management costs. Unlike with a centralized, traditional, and monolithic corporation, this income will be held in an account similar to DASH's, as mentioned above, and possessors of ELA above a certain threshold will be able to vote upon its use.   
  
BTC was known for being the first digital currency, but hardly for being the most cohesive. As it happened, while Mr Nakamoto and others pioneered the underlying mechanics of digital monetization and commodification, it is owing to the contributions of later figures that we achieved a cohesive community and mutual respect. BTC was a toolbox, but we needed social understandings, and digital mechanisms underlying those social understandings to establish an economy using those tools. To this, we owe the creators and maintainers of NMC, DOGE, STEEM, DASH, and ETH. NMC pioneered the sharing and reuse of chains through its policy and mechanism of merge-mining. DOGE pioneered community understanding and frictionless protocol upgrades. STEEM gave rise to the reward economy and inflationary policy. DASH gave rise to community governance and just compensation to node-operators. ETH generalized the DASH decentralized fund into a Turing-complete system where anyone can create new contracts.   
  
ELA, having been conceived during the midst of BTC's blocksize disputations, being based on the convictions of its team members, advocates, and advisors, with decades of cumulative experience spanning institutions such as Tsinghua University and the Chinese Academy of Sciences, seeks to learn from the precedents of these aforementioned inventors and to create a sharing economy and reliable asset system based upon responsible decentralized governance, mutual respect, and not to say the least, modular cryptographic fundaments such as enabled the economy in the first place. ELA seeks to be the basis of an economy rather than a simple collection of shell utilities. Having the four major goals secures the ability of ELA to protect the interests of the community as a whole.   
  
  
There are also four minor goals meant to expand upon ELA's economic foundation to serve the next generation of internet consumers and suppliers, and preferrably to blur the line between the two, as more and more individuals enjoy access to the economy made possible with ELA, and the advanced tools provided by the Elastos operating system.   
  
The first minor goal is avoidance of filtration. Simply stated, the ELA implementation will have a wire protocol meant to avoid most types of filtration by avoiding the use of universally fixed constants, also known as magic constants. Thus, the operator of a private subnet or intranet will not be able to allow its members general access to the internet while discriminating against ELA specifically.   
  
The second minor goal is support for NAT hole punching. This will allow people without enterprise-grade internet connections, but only with modest home internet connections, to become node operators. This is a feature which every major digital currency has been absent of. Thus, in order to become a node operator, a person had to rent a cloud computer from Amazon Web Services or Aliyun and proceed to install the node software upon it. He did not have the luxury of simply downloading the program to his own computer and leaving the computer on during days and nights at his discretion. This has been a noticeable obstacle in the proliferation of nodes, as many everyday users wish to contribute to the community at large but are left hanging, as if relegated to being a consumer, not a supplier, on account of them not being professional engineers. ELA, in contrast, will open the opportunity for anyone, using nearly any device, to become a node operator.   
  
The third minor goal is support for sidechains. By using a mechanism similar to Segregated Witness (Segwit), it will be possible to create long-term or short-term sidechains with low, almost negligible fees, allowing new branches of computation to occur without incuring the overhead of the entire network of nodes. Moreover, sidechains are essential to effective execution of micropayments.   
  
The fourth minor goal is support for smart contracts. By extending sidechains to include smart contracts, we can save bandwidth on the main chain and allow contracts only to execute between necessary parties, in a simliar manner to the projected operation of BTS or EOS. Unlike with ETH, our choice to focus upon sidechains means that we can experiment with numerous possible instruction architectures, and that consumers and producers alike have a choice in their instruction architecture, which, while being admittedly an advanced concern, may have important ramifications upon the domain of usage in question.