Whitepaper

P2P Digital Asset Distribution

aphelion.org
support@aphelion.org
October 2017 v.08

Abstract: Built to solve the inherent problems of the crypto exchanges and trading platforms, Aphelion advances distributed ledger technology (DLT) as an open source, peer-to-peer (P2P), decentralized asset distribution application protocol built on the NEO blockchain. Aphelion is designed to power a tokenized transaction called a Distributed Exchange Asset Ledger or DEAL. Aphelion DEAL transactions are facilitated through smart contracts as set forth by users and are independent of exchanges or trading platforms and the constraints they inherently create. An Aphelion token is the Liquidity Verification Device (LVD) powering the DEAL directly between users: instantly, securely and freely.

Disclaimer: This white paper does not constitute an offer or solicitation to sell securities or shares and is for informational purposes only. The APH token is considered a utility tool built within blockchain technology. The Aphelion token (APH) offering does not represent a stock or sale of securities; the Aphelion token does not grant equity or voting rights; the Aphelion token does not grant ownership rights directly or indirectly to the Aphelion company, it's physical, virtual or intellectual properties; the Aphelion token does not grant a debt security and is not an instrument of debt; the Aphelion token does not pay a distribution, disbursement or interest payment to token holders. If any future offers become available they will be made through confidential and appropriate channels and follow all necessary legal requirements. In compliance with recent sec announcements, Aphelion will not market to or accept contributions from any US citizen or resident. In compliance with China Securities and Regulatory Commission (CSRC) and the People's Bank of China (PBOC) regulations, Aphelion will not market to or accept contributions from any citizens or resident of the People's Republic of China (PRC). In compliance with the monetary authority of Singapore, Aphelion will not market to or accept contributions from any citizens or resident of Singapore.

Notice to citizens and residents of the United States of America: This website and the offering memorandum has not been filed with the Securities and Exchange Commission (SEC) as part of a registration statement. Accordingly, this website and the offering memorandum and any other document or material in connection with the offer or sale, or invitation for subscription or purchase of the APH tokens may not be circulated or distributed, nor may the APH tokens be offered or sold, or be made the subject of an invitation for subscription or purchase, whether directly or indirectly, to persons in the United States of America. For residents and citizens of the People's Republic of China (which, for the purposes of this document and offering memorandum, does not include Hong Kong, Macau, and Taiwan): APH tokens may not be marketed, offered or sold directly or indirectly to the public in China and neither this document nor the offering memorandum, which has not been submitted to the Chinese securities and regulatory commission, nor any offering material or information contained herein relating

to APH tokens, may be supplied to the public in China or used in connection with any offer for the subscription or sale of APH tokens to the public in China. The information contained in this website and the offering memorandum will not constitute an offer to sell or an invitation, advertisement or solicitation of an offer to buy any APH tokens within the PRC.

Notice to prospective subscribers in Singapore: This website and the offering memorandum has not been registered as a prospectus with the monetary authority of Singapore under the Securities and Futures Act (SFA) (Chapter 289). Accordingly, this website and the offering memorandum and any other document or material in connection with the offer or sale, or invitation for subscription or purchase of the APH tokens may not be circulated or distributed, nor may the APH tokens be offered or sold, or be made the subject of an invitation for subscription or purchase, whether directly or indirectly, to persons in Singapore.

Table of Contents

1. Introduction

Distributed ledgers, blockchain technology, cryptocurrencies and their smart contracts are disrupting a multitude of industries. In fact, experts argue that it has the expectation to disrupt the world more than any other industry in history. We are already seeing the applications especially growing in finance. As part of this new technology, developers, at an incredible rate, are building new tools and the race is on to find mainstream, secure solutions that the general public and its institutions will embrace.

1.1 Background

As part of the blockchain eco-system, cryptocurrencies such as Bitcoin (BTC), NEO (formerly AntShares) and Ethereum (ETH) have emerged as early leaders in digital asset distribution. Based on blockchain technology and distributed ledgers, Satoshi Nakamoto developed the first cryptocurrency in 2008 called Bitcoin (BTC) [1]. Since then, many cryptocurrencies have been created and the market cap is growing like nothing ever seen before (up 1000%+ in 2017). Entrepreneurs, venture capitalists, bankers, and other experts speculate that cryptocurrencies will eventually be the new norm and a bevy of businesses are blooming as a result. But, blockchain and distributed ledger technology behind the emerging cryptocurrencies could prove much more significant.

1.2 Blockchain Technology

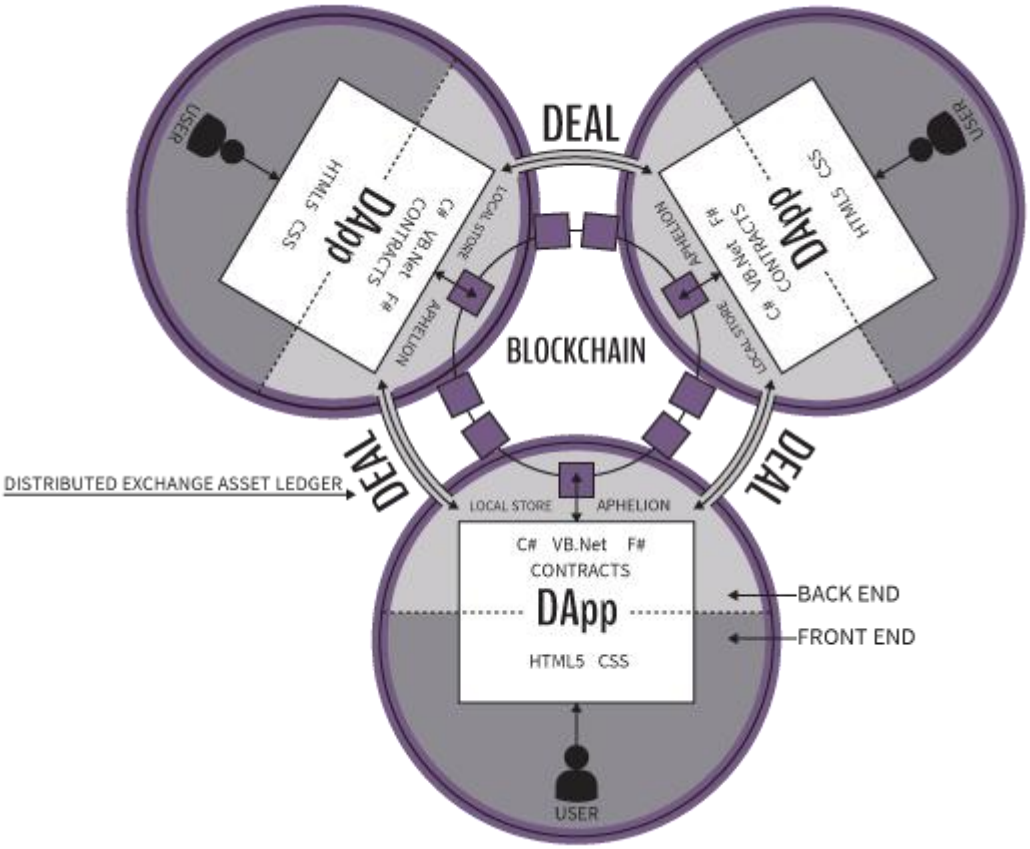
Cryptocurrency is made possible by blockchain technology. “The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value.”[2] What is Blockchain Technology? “Blockchain as a historical fabric underneath recording everything that happens exactly as it occurs. Then the chain stitches that data into encrypted blocks that can never be modified and scatters the pieces across a worldwide network of distributed computers or "nodes." Blockchain always has an immutable "ledger" that you can see, verify, and control. At the same time, it has no single point of failure from which records or digital assets can be hacked or corrupted. Because of its distributed-ledger technology, blockchain has applications across every kind of digital record and transaction, and we're already beginning to see major industries leaning into the shift.”[3]

1.3 Distributed Ledger Technology

"A distributed ledger is a type of database spread across multiple sites, regions, or participants. As one would expect, a distributed ledger has to be decentralized, otherwise it would resemble a centralized database like most companies use today. Removing the intermediary party from the equation is what makes the concept of distributed ledger technology so appealing. Moreover, enterprises use distributed ledger technology to process, validate or authenticate transactions or other types of data exchanges. Records are stored in the ledger once consensus is achieved by the majority of parties. Every record stored in the distributed ledger is timestamped and has its very own cryptographic signature. All of the participants on the distributed ledger can view all of the records in question. The technology provides a verifiable and auditable history of all information stored on that particular dataset. Distributed ledger technology will often to be referred to as DLT in financial and government circles."[4] Leveraging DLT, Aphelion will decentralize the P2P transactions in a safe, secure and truly decentralized manner powering the process through its token rather than an exchange. Bypassing the exchanges and allowing the DEAL to happen on a distributed ledger on the NEO blockchain is a giant step forward in the future of crypto trades.

1.4 Decentralized Application (DApp)

A decentralized application or DApp, as it is abbreviated, has its backend code running on a decentralized peer-to-peer network. A DApp can have frontend code and user interfaces written in any language (just like an app) that can make calls to its backend. Furthermore, its front end can be hosted on decentralized storage such as Swarm or IPFS. As illustrated in the graphic below, if an app=frontend+server, then DApp = frontend + community + contracts. Aphelion contracts are code that runs on the global Aphelion decentralized peer-to-peer protocol.



By design, Aphelion is a DApp and by decentralizing the application it becomes: peer-to-peer, open source, operated autonomously and cannot be controlled by a single operator or entity. The Aphelion (APH) cryptographic token will be stored in a public decentralized blockchain. Consensus Node installation ultimately derives value for the application.

1.5 PoW, PoS and Next Generation dBFT

From PoW to PoS "Proof-of-Work (PoW), Bitcoin's consensus algorithm responsible for the network's high energy demand, renders the system's bookkeeping mechanism artificially resource intensive. Bitcoin nodes, mining blocks and verifying transactions, have to proof the performance of cryptographic tasks in order to be eligible for the sought-after block reward. As a result, anyone trying to forge BTC transactions, or otherwise compromise the blockchain records, would have to outcompete all other miners and the energy they're investing in keeping bitcoin nice and clean. According to the energy estimates stated above, this means that thanks to PoW, an attacker would have to invest the aggregated energy consumption of a small North-American city, just to enforce their will on the Bitcoin blockchain. The most popular alternative to PoW, used by most alternative cryptocurrency systems, is called Proof-of-Stake, or PoS. PoS is highly promising in the sense that it doesn't require blockchain nodes to perform arduous, and otherwise useless, cryptographic tasks in order to render potential attacks costly and infeasible. Hence, this algorithm cuts the power requirements of PoS blockchains down to sane and manageable amounts, allowing them to be more scalable without guzzling up the planet's energy reserves. PoS is a viable alternative to PoW, which although highly energy inefficient, has proven itself as trustworthy during the last eight years. However, both systems have a crucial flaw, rarely addressed in the still somewhat countercultural crypto community. PoS, as well as PoW, simply allows the blockchain to fork into

two alternative versions if for some reason consensus breaks. In fact, most blockchains fork most of the time, only to converge back to a single source of truth a short while afterwards, as it is depicted in the image above. By many crypto enthusiasts, this obvious bug is very often regarded as a feature, allowing several versions of the truth to survive and compete for public adoption until a resolution is generated. This sounds nice in theory, but if we want to see blockchain technology seriously disrupt and/or augment the financial sector, this ever lurking possibility of the blockchain splitting into two alternative versions cannot be tolerated. Byzantine Fault Tolerance and dBFT The term Byzantine Fault Tolerance (BFT) derives its name from the Byzantine Generals problem in Game Theory and Computer Science, describing the problematic nature of achieving consensus in a distributed system with suboptimal communication between agents which do not necessarily trust each other. The BFT algorithm arranges the relationship between blockchain nodes in such a way that the network becomes as good as resilient to the Byzantine Generals problem, and allows the system to remain consensus even if some nodes bare malicious intentions or simply malfunction. To achieve this, NEO's version of the delegated BFT (or dBFT) algorithm acknowledges two kinds of players in the blockchain space: professional node operators, called bookkeeping nodes, who run nodes as a source of income, and users who are interested in accessing blockchain advantages. Theoretically, this differentiation does not exist in PoW and most PoS environments, practically, however, most Bitcoin users do not operate miners, which are mostly located in specialized venues run by professionals. Accordingly, block verification is achieved through a consensus game held between specialized bookkeeping nodes, which are appointed by ordinary nodes through a form of delegated voting process. In every verification round one of the bookkeeping nodes is pseudo-randomly appointed to broadcast its version of the blockchain to the rest of the network. If ⅔ of the remaining nodes agree with this version, consensus is secured and the blockchain marches on. If less than ⅔ of the network agrees, a different node is appointed to broadcast its version of the truth to the rest of the system, and so forth until consensus is established. In this way, successful system attacks are almost impossible to execute unless the overwhelming majority of the network is interested in committing financial suicide. Additionally, the system is fork proof, and at every given moment only one version of the truth exists. Without complicated cryptographic puzzles to solve, nodes operate much faster and are able to compete with centralized transaction methods."[5]

1.6 Aphelion Built on NEO dBFT

Because dBFT solves for the challenges identified and outlined above in Bitcoin PoW and subsequent alternative PoS technologies Aphelion will be built on NEO as an eco-friendly, open source, completely decentralized digital asset application creating the most secure and decentralized application for digital asset distribution. This will allow users to transact a DEAL P2P and independent of the exchanges, trading platforms and the limitations/challenges they bring. Aphelion is a tokenized DApp protocol. **Why NEO?**"NEO supports faster development and deployment of smart contracts and projects, as it enables developers to build on programming languages already familiar with them. We provide various advanced languages in the form of compiler," says Da Hongfei (founder). "Besides .Net and Java, we will support Python and Go in the future which can cover more than 90 percent of developers. Compared with Ethereum, development has more smooth learning curve and shorter learning cycle, allowing for fast introduction of projects."[6]

	bitcoin	ethereum	NEO
Efficiency	POW on ASIC machines uses vast amounts of energy	GPU miners collectively using more energy than an entire country*	dBFT ensures finality through highly efficient method
Secure Contracts	Pseudo-anonymity creates lack of integrity in transactions	Vulnerable contract code prone to hacker attacks**	Integrated digital identity allows for real world applications
Dev Languages	C++	Solidity	C#, .Net, Java, Python and Go coming which can cover 90% of developers
Scalability	Peak transaction per second is limited to 3-4	Current peak transaction per second is 20	Up to 10,000 transactions per second

1.7 The Cryptocurrency Market

"As of April 2017, the combined market value of all cryptocurrencies is $27 billion, which represents a level of value creation on the order of Silicon Valley success stories like AirBnB."[9] In late August 2017, the Market cap surpassed $180 billion, meaning the cryptocurrency total market cap has risen nearly 1000% this year, according to bitcoin.com.[12. The Problem Blockchain technology and subsequent cryptocurrencies are so new that many crippling challenges exist across trading platforms and exchanges. Currently, digital currencies do not connect to each other in the same way that information networks do. The current exchange model for currencies has a critical barrier to linking small-scale currencies to other popular currencies using a market-determined exchange rate. Also, the exchanges and trading platforms are in essence acting as a centralized system that inherently brings associated faults and defeats the purpose of decentralization. Challenges facing crypto-exchanges and trading platforms today: Centralization: Rules, fees, non-liquid assets, exchanges control private keys to user wallets allowing the exchange to have full custodial rights of the funds. Complexities: Trading platforms & exchanges lack any cross-consistency in virtually every aspect of their technology. Barriers to Entry: There are different rules to join each platform, delays in approval, traditional currency deposits vs. digital only deposits, lack of instant deposits. Challenges of Use: Trades blocked without explanation, daily limits, poor UI, buggy software, not user-friendly. Latency: Incessant lack of speed and performance issues. Lack of Support: There is a complete lack of customer support and inability to respond across most big-name platforms; It is not uncommon to wait weeks or months for a reply. Lack of Security: Multiple hacks, lost funds, privacy breaches, shut down sites. Lack of Privacy: Required verification, credit card, driver's license scans, passports.

2. The Problem

2.1 Cryptocurrency Challenges

Because Bitcoin is a relatively lite blockchain system, it requires additional development protocols to make it functional for transactional exchanges. NEO is also compatible with several coding languages, whereas ETH is only compatible with Solidity. "For instance, while you might think that the current proof-of-work (POW) consensus mechanism used by Bitcoin and Ethereum is a benefit, it actually comes with a cost. There is an issue with the lack of finality. Bitcoin transactions are final, you say? Not really. The protocol favors availability over finality — this means forks and lone blocks are a possibility, and we have previously observed how Bitcoin projects tend to "fork" whenever there are serious security concerns or when developers have disagreements regarding the standard. POW is also very energy-intensive, which means nodes spend a lot of money on electrical bills."[6]

2.2 Centralized Exchanges

There is widespread use of several cryptocurrency trading platforms and exchanges. They are the clear mechanism for P2P trading, but they are not decentralized. They act as intermediaries between traders initiating trades and this poses a number of inherent challenges. First, exchanges set the rules for who can trade, what can be traded and when. There are countless stories of user's accounts and even initiated trades being deleted or frozen without explanation. We've also had numerous security breaches resulting in hundreds of millions (USD) being stolen. On top of these inherent challenges the exchanges are facing there is a complete lack of support facing many users today. These so-called decentralized exchanges are not decentralized at all, in fact quite the opposite. "P2P exchanges aren't better than the regular ones in every regard - longer trade times, less intuitive use cases and lower liquidity are some of their comparative disadvantages. Most flaws of decentralized exchanges are caused simply by the fact that they are a relatively new kind of service. For example, Bitsquare, arguably one of the oldest of such exchanges, has been around for just three years and most of that was the development period. As such, these exchanges have to deal with a number of problems. For example, most of them are currently aimed at small, specific audiences of crypto enthusiasts and haven't had the need to cater to newcomers - because of that, they tend to be less intuitive to use. For the same reasons - small audience and early stage of existence, decentralized exchanges usually have much lower trading volumes than the regular ones. Longer trade times, on the other hand, are likely a disadvantage that will take a while to fix, if ever. They are caused by the manner in which the trades are conducted - with traders having to wait for actual Bitcoin and fiat transactions to complete before a trade is concluded. This last issue, coupled with the lower liquidity, means that P2P exchanges are not at all in demand with, for example, professional traders, who need fast transactions to make timely deals. In their current state, these exchanges can only be useful to people interested in the specific advantages they offer - the increased resilience, privacy, security and freedom of payments." [11]

	Decentralized Exchanges	Centralized Exchanges
Equanimity between buyer & seller		
Loss of funds from exchange shutdown		
Potential of frozen accounts		

Income for exchange from transaction	
Trading security risks	
Deposits required	

2.3 Decentralized Exchanges

Several projects make the claim of being a P2P Decentralized Exchange (DEX). However there are very few built as dApps, completely within a blockchain. Some are centralized client to server operations that rely on an organization's hardware and proprietary software and others are simply a protocol that requires integration into existing centralized exchanges to function properly. Aphelion aims to be one of the pioneers of DEX residing completely within the blockchain as dApp, requiring only an open source user interface to access data and control smart contracts to trade digital assets.

Areas of concern:

Ripple

Ripple[12] is a protocol that offers a real-time, settlement system, currency exchange, and remittance network. It requires an existing network to plug into and by design is programed to work within the central banking system. Ripple's protocol could help revolutionize the banking industry by bringing blockchain technology to the world's largest financial institutions. However, it does not offer a P2P decentralized exchange system.

Shapeshift

Shapeshift[13] is a server based operation that is highly dependant on corporate hardware and software to remain functional. Shapeshift makes a remarkable promise to trade peer to peer, instantly without having to deposit funds to an exchange platform. A quick search will reveal that the centralized server infrastructure of shapeshift can leave users with missing coins and lost transactions without support to remedy tough situations.

Loopring

Loopring[14] is an exchange protocol that is currently in development (as of Sept 2017). The loopring protocol requires existing cryptocurrency exchanges to plug into, including user authorization and corporate integration between the exchange and loopring. If loopring can overcome the challenges of integrating with existing exchanges, it could prove to be a promising intermediary.

Bitshares

Bitshares[15], [16] is an industrial grade financial blockchain smart contract platform. It is an excellent example of a truly decentralized technology. Some nuances that one might point out on the Bitshares DEX is the fact that as deposits are made, your assets are stored as collateral by Bitshares while you are issued Bitshares' own version of the currencies you might know in the real world, called smart tokens. Users must trade derivative tokens that

replicate real world currencies and assets. Some examples are bitUSD, Bitshares version of the US Dollar or bitGold, Bitshares version of gold.

OpenLedger

OpenLedger[17] Dex is a cryptocurrency exchange. Much like Bitshares, it allows users to exchange real world assets into derivative tokens, also known as smart tokens which reside in the OpenLedger network. For example, with Openledger, users trade Open.BTC and Open.ETH which are OpenLedger's own version of Bitcoin and Ethereum, respectively.

Bancor

Bancor [18] protocol enables built-in price discovery and a liquidity mechanism for tokens on smart contract blockchains. Like Bitshares and Openledger, Bancor uses "smart tokens" to hold one or more real world tokens in reserve to enable any party to instantly purchase or liquidate the smart token in exchange for any of its reserve tokens. This is done directly through the smart token's contract, at a continuously calculated price according to a formula which balances buy and sell volumes.

Ox

Ox[19] (Zero X) is a protocol that facilitates peer-to-peer exchange of ERC20 tokens on the Ethereum blockchain. The protocol is intended to be used within an existing dApp to facilitate Ethereum based token trading.

3. The Solution

Aphelion's breakthrough token-driven DApp allows for peer-to-peer asset distributions and smart contracts via a DEAL and solves the issues plaguing current exchanges and platforms. The solution is to eliminate the centralization of those mechanisms by allowing users to freely set their own smart contracts and exchange digital assets on their terms in a open source, secure, fast and truly decentralized process directly on the blockchain. The Aphelion DApp and protocol token will solve for latency, frozen or stolen assets and finally free crypto trading forever.

3.1 P2P Digital Asset Distribution and Protocol

Aphelion is a next generation DApp and token protocol that will integrate with any other DApp. Aphelion is truly open source, owned or controlled by no entity, organization or agent. By leveraging the smart contract technology as a protocol with its own tokenized systems of escrow or building blocks, Aphelion users can finally eliminate the barriers and controls of the cryptocurrency exchanges and trading platforms. Aphelion empowers users to trade directly between themselves on the contract terms they choose. It delivers an innovative, tokenized escrow solution for users to instantly trade, transfer, send and receive Aphelion approved currencies to anyone they want and anywhere they want.

3.2 Mission & Vision Statement

Mission: To build collaborative, open source blockchain technology that truly decentralizes asset distribution. Vision: A world powered by decentralized applications.

3.3 Aphelion Technology

NEO Technology: Through technologies such as P2P networking, dBFT consensus, digital certificates, Superconducting Transactions, and cross-chain interoperability the blockchain enables management of smart assets in an efficient, safe and legally binding manner. Digital Assets: Digital assets are programmable assets that exist in the form of electronic data. With blockchain technology, the digitization of assets can be decentralized, trustful, traceable, highly transparent, and free of intermediaries. On the blockchain, users can register, trade, and circulate multiple types of assets such as BTC, ETH, XRP, LTC and NEO to name a few.

3.4 Key Differentiators

True Decentralization: Aphelion trades are transacted P2P and node-based without third-party control or influence. Users can set their own rules in the truest definition of decentralization. It's impossible to take the site down, because there is no site. The transactions only complete when both sides enter into the DEAL (Distributed Exchange Asset Ledger) and the ledger logs it across potentially millions of machines. Cross-Language Scalability: Completely unlike other tokens, Aphelion will be open and buildable across languages such as Python, .Net, C#, F#, Go & Java; making it highly scaleable and conducive to onboarding diverse coding talent. Next Gen DApp: A NEO tokenized system using the DEAL protocol to power a true P2P exchange totally decentralized from the exchanges. Ease of Entry: Aphelion only requires access to an open source Aphelion portal built in-browser, in-app and on desktop. Security: Because the data is truly decentralized across the distributed blockchain ledger it cannot be stolen or corrupted. Control: Aphelion users initiate the DEAL transactions and have total control over the conditions of their individual smart contracts, freeing the transactions from fees and rules.

3.5 Roadmap

Q1 2017 - Concept & Research

- R&D blockchain options
- Identified industry leaders
- Cryptocurrency market research
- Cryptocurrency exchange platform comparison analysis

Q2 2017 - Strategy & Design

- Retained Legal Council
- Created Concept
- Coined Aphelion name and messaging
- Design mockups
- SWOT Analysis

- Crafted our mission
- Vetted concept with blockchain developers

Q3 2017 - Initial Business Rollout & Pre-Marketing

- Incorporated business unit
- Identified markets
- Built partnership networks
- Launched Landing Website
- Compliance framework
- NEO is the One
- Formed Founders alliance agreement
- Recruited and vetted advisers
- Q4 2017 - Marketing & ICO
- Dev kickoff
- Rollout Marketing Efforts
- Develop influencer network
- Build relationships with liquidity providers
- Deploy Testnet
- Manage GitHub repo
- Website enhancements & backend
- KYC verification entity integration
- Finalize and release white paper
- Open private offer
- Decstack channel
- Smart contract testing & auditing
- ICO transaction testing
- Inital dApp development

Q4 2017 - Marketing & ICO (continued)

- Dev kickoff
- Rollout Marketing Efforts
- Develop influencer network
- Build relationships with liquidity providers
- Initial NEO access
- Deploy Testnet
- Manage GitHub repo
- Website enhancements & backend
- KYC verification entity integration
- Finalize and release white paper
- Open private offer
- Telegram channel
- Smart contract testing & auditing
- ICO transaction testing

- Inital dApp development
- KYC audits completed
- ICO kicks-off
- ICO closes
- Tokens distributed
- PR begins
- Regulatory compliance updates

Q1 2018 - A NEO Year Begins

- Full dApp development commences
- Cross blockchain transactions
- Solve for liquidity verification
- Marketing continues
- Exchange registrations begin
- Audits
- APH traded on exchanges
- Launch initial version Aphelion DApp
- Aphelion dApp community development & growth
- Continued market analysis
- Champion advancement of the NEO smart economy

2018 and into the future - 2018 and into the future [tbd]

- Continue to build dev team
- Build brand loyalty & create raving fans
- Expand market reach to all continents
- Leverage partnerships to promote innovation and integration
- Achieve preeminence

3.6 Aphelion Tokens - How It Works...

APH tokens represent a new breed of digital asset distribution. Acting as a digital escrow or Liquidity Verification Device (LVD), the Aphelion token simultaneously captures terms from buyer and seller, reconciles the proposed smart contract, instantly verifies liquidity and settles the DEAL. Aphelion's Distributed Exchange Asset Ledger (or DEAL) is advancing P2P by bypassing the exchanges into a direct and truly node-based decentralized ledger. The APH tokenized DEAL is a protocol DApp residing directly on the blockchain, thus bypassing exchanges and allowing APH to be the Liquidity Verification Device and ultimately unlocking the promises of instant, secure and total decentralization.

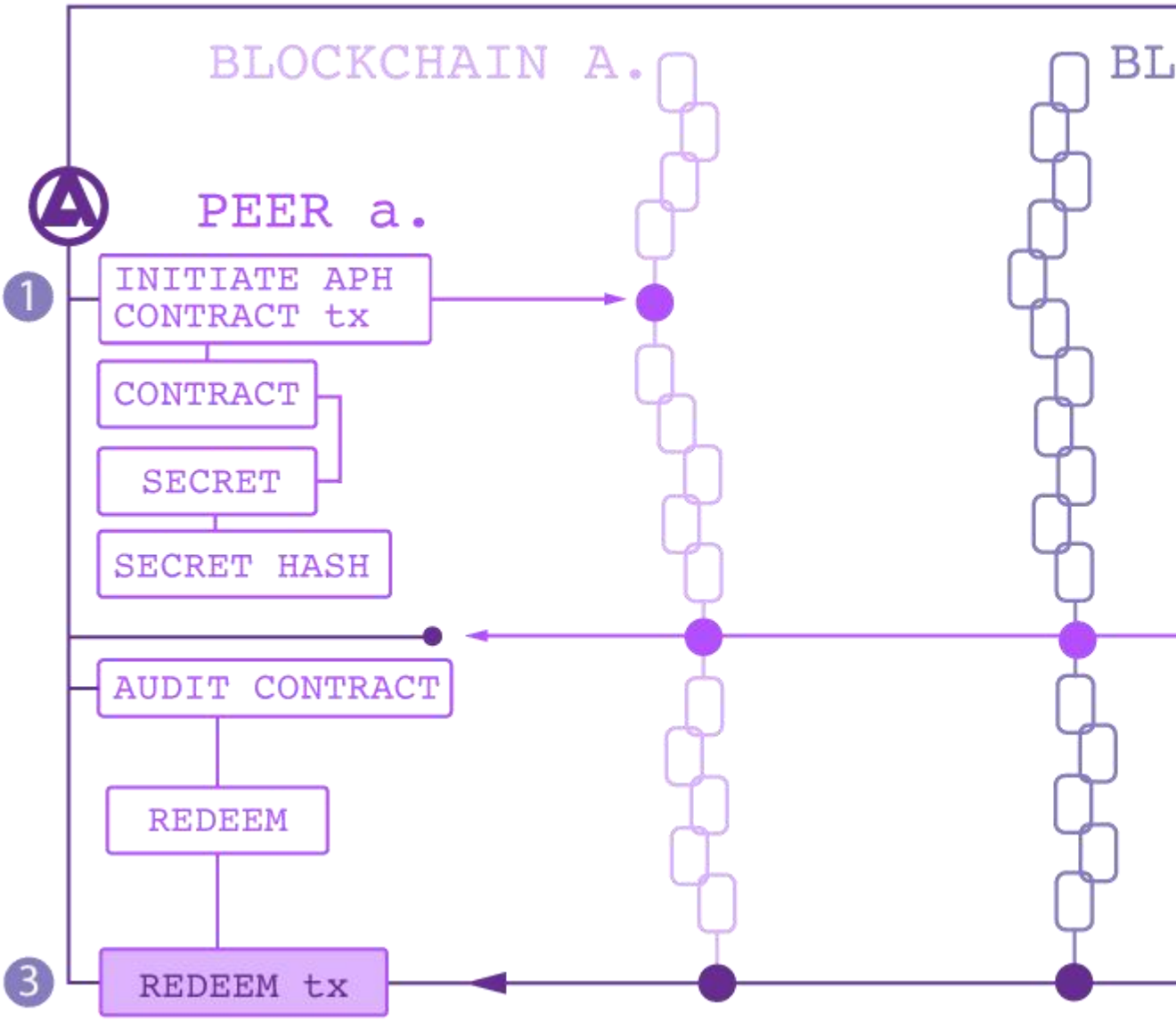
Let's consider two peers who wish to exchange digital assets that exist on separate blockchains. Peer A, we'll call Alex, is willing to trade some of his assets on Blockchain A (B.A) for some minimum amount of assets on Blockchain B (B.B). Peer B, we'll call Bob, is willing to trade some of his assets on B.B for some minimum amount of B.A. Both peers already have addresses on both blockchains and either side can initiate the exchange contract. This distributed cross-chain transaction is executed in multiple phases but is treated as a single unit of work.

In the end, the exchange succeeds on both sides or everything reverts back to its original state.

1. Here it will be Alex who initiates the cross-chain digital asset exchange using an Aphelion token. This creates a contract transaction containing the contract, secret code (often just called the secret) and a hash of that secret code. This also locks down Alex's necessary assets on B.A and provides the address on B.B to where he wishes to receive assets.

2. Next, Bob takes a look at the contract (called auditing) and liking the terms decides to participate. He agrees by making another contract transaction, which uses the secret hash from Alex's contract transaction. This also locks down Bob's required assets on B.B. and provides the address on B.A to where Alex shall send assets to Bob.

3. Alex has a look (audit) at what Bob has sent and decides to close the DEAL. Alex collects Bob's payment by creating a redeeming transaction. This automatically releases Alex's secret to Bob and triggers a second redeeming transaction (4.) that allows Bob to receive Alex's payment. The dual redeeming transactions are similar to a traditional relational database two phase commit where if any part of the meta transaction fails the individual transactions fail and rollback.



A key element that is not being shown in this simplified diagram is that assets can also be refunded, or returned to the original wallet, at certain points during the exchange. Alex's contract transaction will contain a locktime that expires after the transaction has been mined but has not yet been redeemed. Bob's contract transaction will also contain a locktime, which will be half the time of Alex's locktime. If these locktimes expire then the given party can initiate a refund and all relevant assets will be returned.

Whats Next? Aphelion is only just beginning on the NEO blockchain. The ultimate vision is a decentralized node-based bridge connecting communities across blockchains. Aphelion begins on NEO for the intrinsic values of that blockchain and aims to spread its protocol and reach to ETH, BTC and other future blockchains for the eventual utility of the token: complete blockchain agnostic, direct, P2P, cross-dimensional, decentralized exchange, finally bringing the promise of blockchain into its full potential. By building the bridge as a truly decentralized DApp any one point becomes irrelevant in the strength and utility of the DApp and Aphelion token protocol; The whole will exceed the sum of its parts.

3.7 Aphelion Initial Coin Offering

Aphelion ICO is in pre-sale. Early contributors, advisors, and owners have been allotted tokens. The countdown to the official ICO is scheduled for November 15, 2017. Deposits can be made with NEO, BTC and ETH directly at Aphelion.org.

Allotment Breakdown

- 45% Sold ICO
- 5% Incentive Program
- 5% Pre-ICO Contributors
- 15% Advisors
- 30% Organization

3.8 Pricing Structure & Timeline

Aphelion ICO Token price is $0.20

The NEO exchange rate will be determined on Nov 13, 2017 based on a 3 day moving average.

The moving average is determined using the SMA method derived from coinmarketcap.com historical data.

Stage one starts on the first block of November 15, 2017

The ICO Ends on the last block of December 7, 2017

The entire 50M ICO token allocation is available through each round. It is possible that all ICO tokens are sold in round 1.

Any tokens that are unsold through the end of round 3 will be burned.

Example exchange rate at $30 NEO:

Round 1: 1 NEO = 150 APH + 75 APH [225 APH total]

Round 2: 1 NEO = 150 APH + 38 APH [188 APH total]

No Bonus: 1 NEO = 150 APH

Round	Start Date	End Date	Duration	Bonus	Effective Price
1	Nov 15 2017	Nov 15 2017	24hrs	50%	$0.13

	First Block	Last Block			
2	Nov 16 2017 First Block	Nov 22 2017 Last Block	7 Days	25%	$0.16
3	Nov 23 2017 First Block	Dec 7 2017 Last Block	14 Days	No Bonus	$0.20

Use of Proceeds:

- 65% Blockchain & DApp Development
- 10% Marketing
- 15% Operations
- 10% R&D

3.9 Aphelion Smart Contract Moratorium

To preserve the project and protect ICO contributors there will be a mandatory 6 month moratorium on selling Aphelion tokens for all founders and advisors. This policy will be built into the blockchain smart contract for total transparency.

4. Aphelion Team

The Aphelion Team is made up of a global network of successful entrepreneurs, experts, and visionaries with a successful track record in blockchain technology, finance, economics, marketing, security, software engineering and development.

ICO Advisors

Chris Mitchell

LinkedIn

Adi Benari

LinkedIn

Andrew Morrell

LinkedIn

Colan Sewell

LinkedIn

Aaron Levin

LinkedIn

Matt Brozovich

LinkedIn

Joshua Finkleman

LinkedIn

Shannon Hardin

LinkedIn

Jeff Solinsky

LinkedIn

Astrid Baldissera

LinkedIn

Natalie Wilcox

LinkedIn

Eric Liss

LinkedIn

Joe Debuzna

LinkedIn

Michael Jaltuch

LinkedIn

Ian Holtz

LinkedIn

James Hollister

LinkedIn

5. Conclusion

Aphelion is building a next generation, tokenized and blockchain built mechanism to solve for the challenges plaguing the centralized cryptocurrency exchanges and trading platforms. This protocol will allow for a truly peer-to-peer smart contract called a Distributed Exchange Asset Ledger (DEAL). An Aphelion DEAL is a new breed of DApp built on the NEO blockchain that is open source, available across programming languages, transits instantly and frees DEAL makers from: rules, latency and security breaches. Join in our mission to build a collaborative, open source, P2P blockchain technology that finally decentralizes asset distribution and brings blockchain into the future.

6. References

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system (2008) https://bitcoin.org/bitcoin.pdf

[2] Don & Alex Tapscott, authors Blockchain Revolution (2016)

[3] Rob Marvin, Blockchain: The Invisible Technology That's Changing the World (2017)

[4] JP Buntinx, Distributed Ledger Technology Vs Blockchain Technology (March 25, 2017) https://themerkle.com/distributed-ledger-technology-vs-blockchain-technology/

[5] Blockchain project Antshares explains reasons for choosing dBFT over PoW and PoS (July 17, 2017) https://www.econotimes.com/Blockchain-project-Antshares-explains-reasons-for-choosing-dBFT-over-PoW-and-PoS-659275

[6] Daan Pepijn, Here's how NEO plans to top Ethereum and Bitcoin (August 11, 2017) https://thenextweb.com/contributors/2017/08/17/heres-neo-plans-top-ethereum-bitcoin/

[7] Christopher Malmo, Ethereum Is Already Using a Small Country's Worth of Electricity (June 26, 2017)https://motherboard.vice.com/en_us/article/d3zn9a/ethereum-mining-transaction-electricity-consumption-bitcoin

[8] Haseeb Qureshi, A hacker stole $31M of Ether (July 20, 2017) https://medium.freecodecamp.org/a-hacker-stole-31m-of-ether-how-it-happened-and-what-it-means-for-ethereum-9e5dc29e33ce

[9] Dr Garrick Hileman & Michel Rauchs, Global Cryptocurrency Benchmarking Study, The Cambridge Centre for Alternative Finance (2017) https://www.jbs.cam.ac.uk/fileadmin/user_upload/research/centres/alternative-finance/downloads/2017-global-cryptocurrency-benchmarking-study.pdf

[10] Jamie Redman, Another All Time High – Bitcoin Breaks Through 5,000 USD on Asian Exchanges (September 2, 2017) https://news.bitcoin.com/bitcoin-hits-5000-usd-new-all-time-high/

[11] Andrew Marshall, P2P Cryptocurrency Exchanges, Explained (APR 07, 2017) https://cointelegraph.com/explained/p2p-cryptocurrency-exchanges-explained

[12] Peter Todd, The ripple protocol consensus algorithm Review. Ripple Labs Inc White Paper (May, 2015) https://raw.githubusercontent.com/petertodd/ripple-consensus-analysis-paper/master/paper.pdf

[13] Shapeshift Reviews http://bittrust.org/shapeshift

[14] Loopring Project Ltd., LOOPRING Decentralized Token Exchange Protocol (Sept 26, 2017) https://github.com/Loopring/whitepaper/raw/master/en_whitepaper.pdf
[15] Fabian Schuh and Daniel Larimer. Bitshares 2.0: Financial smart contract platform (Nov 12, 2015) http://docs.bitshares.org/_downloads/bitshares-financial-platform.pdf
[16] Fabian Schuh and Daniel Larimer. Bitshares 2.0: General overview (2015) http://docs.bitshares.org/_downloads/bitshares-general.pdf
[17] Open ledger (2017) https://openledger.io/
[18] Eyal Hertzog, Guy Benartzi & Galia Benartzi, Bancor Protocol Continuous Liquidity and Asynchronous Price Discovery for Tokens through their Smart Contracts; aka "Smart Tokens" (March 30, 2017) https://www.bancor.network/static/bancor_protocol_whitepaper_en.pdf
[19] Amir Bandeali. Introducing 0x -An Open Protocol For Decentralized Exchange On The Ethereum Blockchain (February 22, 2017) https://blog.0xproject.com/introducing-0x-d51d5231ba53
[20] Binance GAS(was Antcoin), (July 2017) https://binance.zendesk.com/hc/en-us/articles/115000967291-GAS-was-Antcoin-

7. Appendix

Aphelion DApp Pseudo-Code Algorithm

Parties

• {COIN_A} holder

• {COIN_B} holder

Process

• The 'superconducting transaction' (also 'on-chain atomic swap') proceeds through two transactions, one on the {COIN_A} blockchain, the other on the {COIN_B} blockchain.

◦ [1]:{COIN_A} holder has an unspent amount, A, of {COIN_A} in an address recorded in a transaction on the {COIN_A} blockchain. {COIN_A} holder will pay this unspent amount into a {COIN_A} address controlled by {COIN_B} holder through a transaction on the {COIN_A} blockchain.

◦ [2]:{COIN_B} holder has an unspent amount, B, of {COIN_B} in an address recorded in a transaction on the {COIN_B} blockchain. {COIN_B} holder will pay this unspent amount into a {COIN_B} address controlled by {COIN_A} holder through a transaction on the {COIN_B} blockchain.

Steps

◦ {COIN_A} holder 'initiates'.

▪ Obtains following information from {COIN_B} holder:

• {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} payment will be made.

▪ Creates and publishes contract transaction on {COIN_A} blockchain, with a locktime set by the seller sometime in the future (user set expiry date/time).

▪ This step returns the secret, the secret hash, the contract script, the contract transaction, and a refund transaction that can be sent after (user set expiry date/time) if necessary.

◦ {COIN_B} holder 'audits contract'.

▪ Obtains following information from {COIN_A} holder:

• Swap-script, the output script that may be redeemed on the {COIN_A} blockchain by one of two signature scripts.

• Trans, superconducting transaction for {COIN_A} blockchain.

▪ Inspects {COIN_A} blockchain superconducting transaction contract script to review addresses that may claim the output, the locktime, and the secret hash. Also validates that contract transaction pays to the contract and reports the contract output amount.

◦ {COIN_B} holder 'participates'.

▪ Obtains following information from {COIN_A} holder:

• {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} payment will be made.

• Secret-hash, the hash of the secret key for the {COIN_A} blockchain contract transaction.

▪ Creates and publishes contract transaction on {COIN_B} blockchain, incorporating also the secret hash from the {COIN_A} blockchain contract transaction 'initiated', above, and with a locktime of (user set expiry date/time).

▪ This step returns the the contract script, the contract transaction, and a refund transaction that can be sent after (user set expiry date/time) if necessary.

◦ {COIN_A} holder 'audits contract'.

▪ Obtains following information from {COIN_B} holder:

• Swap-script, the output script that may be redeemed on the {COIN_B} blockchain by one of two signature scripts.

• Trans, superconducting transaction for {COIN_B} blockchain.

▪ Inspects {COIN_B} blockchain superconducting transaction contract script to review addresses that may claim the output, the locktime, and the secret hash. Also validates that contract transaction pays to the contract and reports the contract output amount.

◦ {COIN_A} holder 'redeems'.

▪ Will already have obtained (see prior step) the following information from {COIN_B} holder:

• Swap-script, the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts.

• Trans, the superconducting transaction for the {COIN_B} blockchain.

▪ Redeems {COIN_B} coins paid into the contract in {COIN_B} blockchain by {COIN_B} holder. Redeeming requires the secret, known only to the {COIN_A} holder up to this point.

◦ {COIN_B} holder 'extracts secret'.

▪ Extracts secret from {COIN_A} holder's redemption transaction. With the secret known, the {COIN_B} holder may claim the {COIN_A} coins paid into the contract in the {COIN_A} blockchain by {COIN_A} holder.

◦ {COIN_B} holder 'redeems'.

▪ Will already have obtained (see 'audit contract' step) the following information from {COIN_A} holder:

• Swap-script, the output script that may be redeemed on the {COIN_A} blockchain by either of two signature scripts.

• Trans, the superconducting transaction for the {COIN_A} blockchain.

▪ Redeems {COIN_A} coins paid into the contract in {COIN_A} blockchain by {COIN_A} holder.

Refunds

◦ If a period of time equal to the time-lock (i.e. (user set expiry date/time), in the case of the {COIN_A} blockchain superconducting transaction, and (user set expiry date/time), in the {COIN_B} case) expires after the transaction has been mined but has not been redeemed, the contract output can be redeemed back to the holder's wallet.

Pseudo-code

'initiate', by {COIN_A} holder

```
{COIN_A} holder runs:

$ 'initiate', with parameters

- [{COIN_A} blockchain, i.e. the blockchain on which {COIN_A} holder's payment will be made]

- [string representing {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} payment will be made]

- [string representing A, amount of {COIN_A} to be paid to this address]

{

Decode parameter [string representing {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} payment will be made].

If it conforms with a valid address for the {COIN_A} blockchain, return this address, their-address.

Decode [string representing A, amount of {COIN_A} to be paid to this address]. If it conforms to a valid double-precision floating-point

number (i.e. binary64), and is not NaN or +/- infinity, return this number, amount.

Open JSON-RPC connection with the {COIN_A} blockchain.

Generate [secret], a new secret key for the {COIN_A} blockchain.

Calculate [secret-hash], the hash of [secret].

Calculate [lock-time], a locktime (user set expiry date/time) from current time.

Calculate [refund-address], a {COIN_A} address for the refund transaction.


Build the superconducting contract on the {COIN_A} blockchain, with parameters:

- [their-address]

- [lock-time]

- [secret-hash]

- [refund-address]

Return [swap-script], the output script that may be redeemed on the {COIN_A} blockchain by one of two signature scripts:

- [{COIN_B} holder's sig] [{COIN_B} holder's pub key] [{COIN_A} holder's secret], or

- [{COIN_A} holder's sig] [{COIN_A} holder's pub key]

Calculate [swap-address-script-hash], a new address script hash of [swap-script].

Calculate [tx-script], a new script to pay the transaction output to [swap-address-script-hash].

Calculate [fee], the fees associated with the transaction.

Calculate [trans], superconducting transaction for {COIN_A} blockchain, with parameters:

- [A, amount]

- [tx-script]

- [refund-address]

- [fee]

- [lock-time]

Sign [trans].

Calculate:

- [refund trans], the refund transaction

- [refund fee], the fee associated with the refund transaction.


Return and Display:

- [secret]

- [secret-hash]

- [swap-script]

- [trans]

- [refund-trans]

- [lock-time]
```

```
Publish transaction.

}


'audit contract', by {COIN_B} holder

{COIN_B} holder runs:

$ 'auditcontract', with parameters

- [{COIN_B} blockchain, i.e. the blockchain on which {COIN_B} holder's payment will be made]

- [string representing swap-script, output script that may be redeemed on the {COIN_A} blockchain by one of two signature scripts]

- [string

representing trans, superconducting transaction for {COIN_A} blockchain]

{

Decode parameter [string representing swap-script, output script that may be redeemed on the {COIN_A} blockchain by one of two signature

scripts]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.

Decode parameter [string representing trans, superconducting transaction for {COIN_A} blockchain]. If it conforms to a valid

hexadecimal string of the right length, return the bytes, swap-script.


Open JSON-RPC connection with the {COIN_A} blockchain.



Calculate superconducting transaction data pushes, with parameters:

- [swap-script], the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts

Return

- [address], {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} payment will be made

- [secret-hash], the hash of the secret key for the {COIN_A} blockchain contract transaction

- [lock-time]

Calculate pay to address, with parameters:

- [trans], the superconducting transaction for the {COIN_A} blockchain


Return

- [PubKeyTx], address on {COIN_A} blockchain into which {COIN_A} holder will make payment


Display

- [swap-script-hash], address on {COIN_A} blockchain of superconducting contract

- [amount], value of {COIN_A} to be paid into {COIN_B} holder's address on {COIN_A} blockchain

- [address], {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} will be paid

- [refund-address], {COIN_A} holder's address on {COIN_A} blockchain for payment of refund of {COIN_A}

- [lock-time]

}



'participate', by {COIN_B} holder

{COIN_B} holder runs:

$ 'participate', with parameters

- [{COIN_B} blockchain, i.e. the blockchain on which {COIN_B} holder's payment will be made]

- [string representing {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} payment will be made]
```

- [string representing B, amount of {COIN_B} to be paid to this address]

- [string representing secret-hash, the hash of the secret key for the {COIN_A} blockchain contract transaction]

{

Decode parameter [string representing {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} payment will be made] .

If it conforms with a valid address for the {COIN_B} blockchain, return this address, their-address.

Decode [string representing B, amount of {COIN_B} to be paid to this address]. If it conforms to a valid double-precision floating-point

number (i.e. binary64), and is not NaN or +/- infinity, return this number, amount.

Decode [string representing secret-hash, the hash of the new secret key for the {COIN_A} blockchain contract transaction]. If it

conforms to a valid hexadecimal string of the right length, return the bytes, their-secret-hash.

Open JSON-RPC connection with the {COIN_B} blockchain.

Calculate [lock-time], a locktime (user set expiry date/time) from current time.

Calculate [refund-address], a {COIN_B} address for the refund transaction.

Build the superconducting contract on the {COIN_B} blockchain, with parameters:

- [their-address]

- [lock-time]

- [their-secret-hash]

- [refund-address]

Return [swap-script], the output script that may be redeemed on the {COIN_B} blockchain by one of two signature scripts:

- [{COIN_A} holder's sig] [{COIN_A} holder's pub key] [{COIN_A} holder's secret], or

- [{COIN_B} holder's sig] [{COIN_B} holder's pub key]

Calculate [swap-address-script-hash], a new address script hash of [swap-script].

Calculate [tx-script], a new script to pay the transaction output to [swap-address-script-hash].

Calculate [fee], the fees associated with the transaction.

Calculate [trans], superconducting transaction for {COIN_B} blockchain, with parameters:

- [B, amount]

- [tx-script]

- [refund-address]

- [fee]

- [lock-time]

Sign [trans].

Calculate: - [refund trans], the refund transaction

- [refund fee], the fee associated with the refund transaction.

Return and Display:

- [secret]

- [secret-hash]

- [swap-script]

- [trans]

- [refund-trans]

- [lock-time]

Publish transaction.

}

'audit contract', by {COIN_A} holder

{COIN_A} holder runs:

$ 'auditcontract', with parameters

- [{COIN_B} blockchain, i.e. the blockchain on which {COIN_B} holder's payment will be made]

- [string representing swap-script, output script that may be redeemed on the {COIN_B} blockchain by one of two signature scripts]

- [string representing trans, superconducting transaction for {COIN_B} blockchain]

{

Decode parameter [string representing swap-script, output script that may be redeemed on the {COIN_B} blockchain by one of two signature scripts]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.

Decode parameter [string representing trans, superconducting transaction for {COIN_B} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.

Open JSON-RPC connection with the {COIN_B} blockchain.

Calculate superconducting transaction data pushes, with parameters:

- [swap-script], the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts

Return

- [address], {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} payment will be made

- [secret-hash], the hash of the secret key for the {COIN_B} blockchain contract transaction

- [lock-time]

Calculate pay to address, with parameters:

- [trans], the superconducting transaction for the {COIN_B} blockchain

Return

- [PubKeyTy], address on {COIN_B} blockchain into which {COIN_B} holder will make payment

Display

- [swap-script], address on {COIN_B} blockchain of superconducting contract

- [amount], value of {COIN_B} to be paid into {COIN_A} holder's address on {COIN_B} blockchain

- [address], {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} will be paid

- [refund-address], {COIN_B} holder's address on {COIN_B} blockchain for payment of refund of {COIN_B}

- [lock-time]

}

'redeem', by {COIN_A} holder

{COIN_A} holder runs:

$ 'redeem', with parameters

- [{COIN_B} blockchain, i.e. the blockchain on which {COIN_B} holder's payment will be made]

- [string representing swap-script, the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts:

- [{COIN_A} holder's sig] [{COIN_A} holder's pub key] [{COIN_A} holder's secret], or

- [{COIN_B} holder's sig] [{COIN_B} holder's pub key]]

- [string representing trans, the superconducting transaction for the {COIN_B} blockchain]

- [string representing secret, the secret key for the {COIN_A} blockchain]

{

Decode parameter [string representing swap-script, the output script that may be redeemed on the {COIN_B} blockchain by either of

two signature scripts]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.

Decode [string representing trans, the superconducting transaction for the {COIN_B} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, trans.

Decode [string representing secret, the secret key for the {COIN_A} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, secret.

Open JSON-RPC connection with the {COIN_B} blockchain.

Calculate superconducting transaction data pushes, with parameters:

- [swap-script], the output script that may be redeemed on the {COIN_B} blockchain by either of two signature scripts

Return

- [address], {COIN_A} holder's address on {COIN_B} blockchain, into which {COIN_B} payment will be made
- [secret-hash], the hash of the secret key for the {COIN_A} blockchain contract transaction

Calculate pay to address, with parameters:

- [trans], the superconducting transaction for the {COIN_B} blockchain

Return

- [PubKeyTy], address on {COIN_B} blockchain into which {COIN_B} holder will make payment

Verify [address] and [PubKeyTy] are equal.

Calculate [pay-script], script to pay a transaction output to [PubKeyTy].

Create [redeemTx], redeem transaction.

Sign [redeemTx].

Publish [redeemTx]

}

'extract secret', by {COIN_B} holder

{COIN_B} holder runs:

$ 'extractsecret', with parameters:

- [string representing redeemTx, the redeem transaction published by {COIN_A} holder on the {COIN_B} blockchain]
- [string representing secret-hash, the hash of the secret key for the {COIN_A} blockchain contract transaction]

{

Decode [string representing redeemTx, the redeem transaction published by {COIN_A} holder on the {COIN_B} blockchain]. If it conforms to a valid hexadecimal string of the right length, return the bytes, redeemTx.

Decode [string representing secret-hash, the hash of the new secret key for the {COIN_A} blockchain contract transaction]. If it conforms to a valid hexadecimal string of the right length, return the bytes, their-secret-hash.

Open JSON-RPC connection with the {COIN_B} blockchain.

Loop over all pushed data, searching for one that hashes to the expected hash. Return [secret].

Display [secret].

}

'redeem', by {COIN_B} holder

{COIN_B} holder runs:

$ 'redeem', with parameters

- [{COIN_A} blockchain, i.e. the blockchain on which {COIN_A} holder's payment will be made]
- [string representing swap-script, the output script that may be redeemed on the {COIN_A} blockchain by either of two signature

```
scripts:
- [{COIN_B} holder's sig] [{COIN_B} holder's pub key] [{COIN_A} holder's secret], or
- [{COIN_A} holder's sig] [{COIN_A} holder's pub key]]
- [string representing trans, the superconducting transaction for the {COIN_Aa} blockchain]
- [string representing secret, the secret key for the {COIN_A} blockchain]
{
Decode parameter [string representing swap-script, the output script that may be redeemed on the {COIN_A} blockchain by either of
two signature scripts]. If it conforms to a valid hexadecimal string of the right length, return the bytes, swap-script.
Decode [string representing trans, the superconducting transaction for the {COIN_A} blockchain]. If it conforms to a valid hexadecimal
string of the right length, return the bytes, trans.
Decode [string representing secret, the secret key for the {COIN_A} blockchain]. If it conforms to a valid hexadecimal string of
the right length, return the bytes, secret.
Open JSON-RPC connection with the {COIN_A} blockchain.
Calculate superconducting transaction data pushes, with parameters:
- [swap-script], the output script that may be redeemed on the {COIN_A} blockchain by either of two signature scripts

Return
- [address], {COIN_B} holder's address on {COIN_A} blockchain, into which {COIN_A} payment will be made
- [secret-hash], the hash of the secret key for the {COIN_A} blockchain contract transaction
Calculate pay to address, with parameters:
- [trans], the superconducting transaction for the {COIN_A} blockchain

Return
- [PubKeyTy], address on {COIN_A} blockchain into which {COIN_A} holder will make payment
Verify [address] and [PubKeyTy] are equal.
Calculate [pay-script], script to pay a transaction output to [PubKeyTy].
Create [redeemTx], redeem transaction.
Sign [redeemTx].
Publish [redeemTx]
}

'refund', by either holder
Either holder runs:
$ 'refund', with parameters
- [B, blockchain, i.e. the blockchain on which refund will be made]
- [string representing swap-script, for the superconducting transaction to be refunded]
- [string representing trans, the superconducting transaction to be refunded]
{
Decode [string representing swap-script, for the superconducting transaction to be refunded]. If it conforms to a valid hexadecimal
string of the right length, return the bytes, redeemTx.
Decode [string representing swap-script, for the superconducting transaction to be refunded]. If it conforms to a valid hexadecimal
string of the right length, return the bytes, their-secret-hash.
Open JSON-RPC connection with the blockchain, B.
Calculate superconducting transaction data pushes, with parameters:
- [swap-script], the output script that may be redeemed on the {COIN_A} blockchain by either of two signature scripts
```

```
Return

- [amount], value to be refunded on blockchain, B

- [fees], fees associated with the transaction

- [refund-address], the address on blockchain, B, into which refund will be made

Calculate [pay-script], script to pay a transaction output to [refund-address].

Create [refundTx], refund transaction.

Sign [refundx].

Publish [refundTx]

}
```