Phoenix Whitepaper v1.1

Introduction

In early 2018 a cryptocurrency (Hydro) was created with a promise of being a complete set of decentralized, community built, open-source protocols, that could work hand in hand with proprietary fintech technology.

The initial decentralization plan included a Community Development Program where developers would be paid in tokens to build on the protocols and create a digital identity driven dApp Store. In 2020 it became clear that a number of factors including the creators majority ownership of the token supply (~70%), legal ramifications in the US, and the core developers leaving the project had created insurmountable problems with the project's leadership.

Development itself hadn't been subject to these problems. In fact a powerful ecosystem had been built by a hardworking community over the previous two years. The centralization of power by a corporate entity that was shackled by US legislation was the overwhelming issue. They couldn't release or burn tokens, they couldn't hand them off to individual developers, and they couldn't develop the open-source project. A fantastic project had been set adrift with no leadership from its founders.

The Phoenix fork has been launched by community members to correct the missteps the original creators took when attempting to create a decentralized financial ecosystem. The team has forked most of the original open-source smart contracts and created the PHNX token to power it. To ensure the longevity of the project they have put together a healthier token distribution plan, added use cases, and an updated token supply that includes the handing off of the entire project off to the community as a DAO within 12 months.

Project Hydro (HYDRO)	PhoenixDAO (PHNX)
Total Supply	
HYDRO has an 11.1 billion token supply and was trading around five sats which discouraged market activity.	PHNX has a 110 million max token supply which solves market activity issues created by the high supply of HYDRO tokens.
Distribution	
Founders own over 60% of the token supply with no restrictions on how much could be sold at any time or what the money could be used for. This did not sit well with current investors or would-be investors.	The foundation has initial control over only 30% of the supply (earmarked for ecosystem development only) with 70% in the hands of the community. All spending is transparent.
Tokenomics	
The token's only real life use case was as money paying other community members P2P or receiving payment for smart contract development.	The PHNX token has incorporated a vast number of additional use cases for each protocol including a DAO to drive token adoption and expand the ecosystem.
Governance	
A private corporation created the token and controlled all development until starting the decentralized ambassador's program without a well-defined plan for future success of the program or the development process.	Initial development of five flagship dApps, a DAO, and dApp Store upgrades will be overseen by the PhoenixDao Foundation. When the DAO finishes setting up, 100% of all governance will be handed over to the DAO.
Market Responsibility	
One of the core founding developers became disgruntled and dumped a large number of tokens over seven days crashing the market.	Smart contracts are being used to slowly release funds to core developers and founders over time. A market stability plan has been worked out and documented. Future controls on spending and dispersal of funds will be handled by the DAO.
Developer Community	
The three core founding developers left the project abruptly and set protocol development back months and without a core development leadership team.	The Phoenix Dao Foundation has developer recruiters in each continent and is partnering with developer initiatives around the world to keep expanding the developer community.
KYC	
The HYDRO token was de-listed from Bittrex US over questions about US citizens receiving the initial token airdrop.	Only KYC verified non-US residents can take part in the PHNX airdrop on Bittrex Global.

Much of the following document is a summary of the pre-existing Hydro Whitepaper documentation with some edits to make it relevant to Phoenix. The majority of the protocols require little change and so the original whitepaper documentation written by the original ecosystem developers remains relevant.[1]

We will update the core whitepapers as and when it is appropriate to do so.

The Ecosystem

The Phoenix ecosystem is currently made up of the following protocols and solutions however, will continue to evolve as we build:

- Identity
- Payments
- Storage
- Tokenization
- Authentication
- dApp Store
- Events Marketplace

Identity

Phoenix Identity is an identity protocol and the core of the entire Phoenix ecosystem. A modular and open framework, the Identity protocol allows individuals to both craft their digital identities and to share their data in a secure way and with anyone they want. Phoenix Identity is powered by ERC-1484 for decentralized on-/off-chain identity management.

Identity profiles that are created are owned by the person who created it and they are immutable so they cannot be changed by a third party. To ensure ownership remains secure all Phoenix identities are tied to cryptographic keys that are stored on each user's personal device, with a corresponding public ID stamped on the blockchain. It is up to the user who has access to this data and how it is used. They can share as they please without the fear that it will be mishandled or stolen by nefarious third parties.

The Phoenix Identity protocol is an important step towards ending many of the current problems so prevalent with identity management and theft, as summarised below.

- Paper-based and current digital identity management services are expensive to run and easy pickings for identity theft and hackers.
- Online account creation has made the verification of identities increasingly difficult.
- Smartphone technology is growing across the developed and developing world and will play an important role in identity creation and management.

The Phoenix Identity protocol allows for the construction of complex digital identities by leveraging ERC 1484, a global standard for digital identity aggregation. Using the Identity protocol will allow users to secure their online identity whilst complying with widespread KYC/AML standards across the globe.

What is Phoenix Identity (PHNX-ID)?

When creating an identity for use within the Phoenix ecosystem each user is minted a globally unique Ethereum Identification Number (EIN), or in this case a PHNX-ID. The user can then tie any ethereum address that they own to this PHNX-ID.

This structure supports the storage of data within external smart contracts called "Resolvers." Being able to tie this data back to a single PHNX-ID enables unlimited scope for how a user defines their identity, rather than restricting an identity to a predefined set of standards. Also as the Identity protocol acts as a simple data management tool, without encoding the data itself, it allows the user greater flexibility in who has access to that data.

An example use would be the use of decentralized applications or dApps. Traditionally when users want to interact with dApps on Ethereum, they must either transfer their private key to a wallet on the new device or create a new, entirely unassociated, address. Fundamentally, any transfer of a private key away from a local device exposes a user's Ether, tokens, and data tied to that address to phishing and malware. Even careful users are likely to fall victim to systems that structurally require the transfer of private keys across devices. Alternatively, creating new, unassociated keypairs doesn't offer much value to users, as all of their data is stratified among addresses on a separate device. The Phoenix Identity protocol allows users to provably tie multiple addresses back to their core PHNX-ID, extending any Resolver data across their devices without requiring any transfer of private keys. Procedurally, when linking a new address to their PHNX-ID, the owner address initiates a signed claim sign(EIN, addressToClaim, timestamp). Subsequently, the address to be claimed sends a transaction with the signed claim, addressToAdd, and approvingAddress. The initial claim is timestamped in order to prevent a malicious observer from recognizing which address an owner is trying to claim prior to the completion of the claim.

Identity Verification

The concept of identity is relatively meaningless without validation. In general, validators are third parties who can vouch for the validity of data tied to an identity. Since Phoenix Identity empowers users to self-associate with on-chain identity information, it seems apparent that Phoenix Identity only enables for self-attested data, which is relatively limited in scope. However, because the nature of a user's self-attested data is arbitrary, Phoenix Identity allows for norms to drive data attestation in fundamentally decentralized fashion. The logic goes as follows: Because entities observing a user's self-attested data will not derive value from this data without third-party attestations, users will find value in connecting with data structures that enable third-party attestation. Users are able to choose which third-party attestations to pursue, which is closely tied to the ability for entities to choose which third-party attestations they accept. In this way, Phoenix Identity enables a free-flowing ecosystem of third-party attestations under any set of standards adopted by corresponding parties, rather than a predefined attestation structure native to the identity protocol.

In practice, validation structures will likely develop whereby off-chain validators can act as parties trusted by individual actors in the decentralized ecosystem. A hypothetical privacy-preserving implementation is as follows: Once the validation happens off-chain, an on-chain merkle hash would be tied to a user’s PHNX-ID Because nobody can impersonate a validator without access to their private keys, business entities can incorporate on-chain validations into any relevant business logic by querying a user's PHNX-ID. If a user changes any of the data, it would lose its prior validations along with any corresponding significance due to the deviance from the on-chain record. Various implementations of similar validation structures can develop, each catered to their particular use-cases (KYC, reputation, identity implementations, etc) while all resolving to a user's PHNX-ID.

The Role of Phoenix Metadata Within an Ecosystem of Businesses and dApps

While immutable and standard data native to an identity built on top of the Phoenix Identity protocol can help a user establish a secure and global standard for core identity information, Phoenix Identity itself allows users to tie an unlimited range of metadata to their identities by setting third-party dApps as resolvers. As users' interactions with dApps helps build their on-chain identity, they may set a dApp as a resolver for their PHNX-ID, allowing businesses querying the associated metadata to incorporate it into the logic of their own applications.

Fundamentally, a third-party or dApp acts as a certain attribute on a user's PHNX-ID. An application prescribes meaning to a user based on associated data with their PHNX-ID. Let us more closely examine these meanings in the context of a few examples.

What does a Resolver mean from the perspective of the user? Resolvers establish a user's core data. Acquiring validations from reputable sources enhances the integrity of a user's Resolver data which can be relied on by businesses, governments, or decentralized applications querying any relevant identity information.

What does a Resolver mean from the perspective of an app? Since resolvers encode data about a user, a business can programmatically incorporate this data to drive business logic on its platform to improve user experience on its platform or offer users certain permissions. Requirements for third-party validations of this data could be encoded as part of the business logic. If an app were to recognize a given aggregated identity standard processed through Phoenix Identity, it could prescribe whatever meaning to Resolver data best suits its needs.

- Example 1, KYC validation: a government could (off-chain) allow users to register their PHNX-ID through a Resolver for a voting validation. The user could then cast their vote through a third-party voting dApp, which would only tally votes from registered PHNX-ID. This process would ensure that each person can only vote once, eliminating voter fraud or Sybil attacks, while providing complete transparency and much greater efficiency to elections. This can apply to any arbitrary set of digital voting structures and does not enforce a specific implementation.
- Example 2, Social Media: to prevent the creation of fake accounts, an app could implement a system that links a user's PHNX-ID to an account on the app. Any accounts that have their name validated by at least X existing users of the app could display a 'verified' checkmark on the UI of the app. Unlike the first example, this example factors validations from individuals instead of from large institutions. Also, this example observes validation for the PHNXID associated with the PHNX-ID itself rather than a resolver. The openness of the PHNX-ID identity framework allows for a wide range of use-cases

Fundamentally, both applications identify users taking an entirely different approach, but they remain globally interoperable from a user's perspective through a standard user-owned Ethereum Identification Number, created in the ERC 1484 identity registry, with interactions between the user and the application handled through the Identity Smart Contract.

What does being a Resolver mean from the perspective of the Resolver? Resolvers are able to leverage the native functionality of users' PHNX-ID. This offers two key points of value:

- Rather than existing as an independent dApp with which the user interacts, they exist as dApps within the ecosystem of Resolvers associated with a given PHNX-ID. This means a user can seamlessly manage their associated data from a dashboard, and they are not burdened with the struggle of aggregating data from multiple Ethereum addresses and various dApps whenever they want to leverage it.
- Resolvers are able to easily leverage token transfer functionality native to users' PHNX-ID. This streamlines the development process for dApps aiming to monetize without wanting to create sophisticated payment structures within their apps. Users are able to manage PHNX withdrawal balances for all of their resolvers from a single dashboard.

PHNX Tokens in the Identity Protocol

As Phoenix Identity provides out-of-the-box solutions for complex dApp development, one component of this framework is payments.

The PHNX token is not only intended to be a gateway into the ecosystem of Resolvers built on top of PHNX but also a centerpiece for convenient programmatic token transfers between users and resolvers. Through PHNX-ID, users can set specific allowances for resolvers to withdraw PHNX. The user sets the limits on allowed withdrawals on a per-resolver basis, streamlining replication of user-facing models that are currently managed through third-party financial institutions. The Phoenix Identity's flexibility with programmatic token transfers allows businesses to encode an arbitrary set of criteria into their business logic for recurrent processes, such as verifying membership within a particular group in order to offer discounted subscriptions for a product.

More specifically, Phoenix Identity provides a powerful set of tools for dApp developers to easily monetize their products. Currently, we have created a structure for direct payment gateways between users and dApps. This is structured to provide an intuitive and secure model for user-dApp interaction through the opening and closing of gateways called *allowances*. The allowance-based structure has a few advantages over direct token-transfer functionality.

- Allowances allow dApps to programmatically withdraw user funds as certain events are triggered instead of requiring user-initiated transactions every time. This significantly simplifies the user experience.

- Allowances let dApps escrow user funds on a user-driven basis. This prevents dApp/user-flow breakage whenever a user doesn't have the necessary balance to call a function on a dApp through a front-end.
- Allowances are handled on a per-identity basis rather than a per-address basis.
- Allowances are globally useful within a dApp. This means that rather than each function independently handling balance throw errors, global logic checks for allowance-handling can be put in place by a dApp.

What does all this mean for a user? A cleaner user-experience, more consistent with traditional payments rather than cryptocurrency-based payments.

Beyond opening simple user-dApp payments gateways, PHNX-ID allows dApps to open user-contract-dApp payments gateways through what we call Via Contracts. Via Contracts allow token transfers to undergo arbitrary logic before arriving at their end destination. This gives dApps an easy and consistent way to encode transactions that would otherwise be incredibly complicated and nonstandard to implement for simple applications. Examples include:

- Hot-swapping; dApps are able to allow their users to interface cleanly with the PHNX token while accepting or driving logic within their application within their own native token or a token of their choice. This lets them have complex tokens as-needed within their applications without disrupting the user-experience.
- Routing transfers through a side-chain. This allows developers to leverage complex scaling solutions within their application without having to build out custom integrations.
- Applying payments logic to a transaction. This empowers transactions that can do things like accept subscription payments, accrue interest for a loan, check for completion of an event before transferring funds to an end-destination, escrowing funds temporarily for reversible transactions, or more - all implemented at the discretion of the developer. These functionalities are taken for granted in most financial applications today.

Apps, dApps, products, or platforms built on top of Phoenix Idenity can also incorporate PHNX tokens into their processes. For example, certain kinds of validations or actions may require on-chain PHNX token transactions, where users would be required to maintain or transfer PHNX balances.

Open Framework

It is important to note that the proposed framework is an open protocol for identity management. Unlike other blockchain products, there will be no centralized decision on the strength or authenticity of data associations, attestations validations, or those who provide them. It will be up to the global community to identify and punish bad actors. Later in this paper, we examine potential apps, dApps, and platforms that can be built on top of or integrated into Phoenix Identity to increase the effectiveness of the ecosystem.

Phoenix Identity: Technical Details

There are four main entities generally important to Phoenix Identity implementations: users, validators, resolvers, and business entities.

1. Users

Users mint ERC 1484 EINs through the Phoenix Identity protocol to represent their Identities. They attach data to their PHNX-ID and set resolvers in order to tie any form of metadata to their base PHNX-ID. Users can also maintain balances of PHNX within PHNX-ID, creating an easy and intuitive mechanism by which any dApp can interact with a user's PHNX-ID. User data can take any arbitrary format and is associated with Smart Contracts called Resolvers on the Ethereum blockchain.

2. Resolvers

While validators attach validations to a PHNX-ID, resolvers are set by users themselves. Resolvers are dApps that contain identifying data about a user. A simple, but intuitive example is CryptoKitties - setting CryptoKitties as a resolver can tie the ownership of certain kitties back to the user's PHNX-ID identity, even if the Kitties are owned by various Ethereum addresses. dApps that then want to integrate a user's owned Kitties into their experience are able to reliably draw on and regenerate the Kitties.

3. Validators

While validators are not included in the core Phoenix Identity protocol, they are worth mentioning, as they may add significant value when introduced at the dApp level. Resolvers built on Phoenix Identity can encode any range of validation criteria. The simplest level would be a validation dApp in which a trusted KYC provider affirms whether a PHNX-ID is owned by a real person or not. Any PHNX-ID owner could complete KYC through the trusted party, set the validation dApp as a Resolver for their PHNX-ID, and prove that they exist to anybody else who also trusts the KYC party without ever meeting them. In practice, dApps can build nonbinary validation structures with a much more far-reaching range of implications in a variety of industries, with meaning derived from off-chain reputation, or their own native on-chain reputation protocols.

4. Identity Smart Contracts

The ERC 1484 Identity Registry stores, addresses, resolvers, and providers for a given identity. The Identity smart contract acts as a complex provider built on this registry. It allows users to mint identities, deposit PHNX, set resolvers, and prove ownership of multiple Ethereum addresses. Users, resolvers, or any interested party are able to deposit PHNX tokens to the Identity smart contract and interact with them. This facilitates payments, and other token functionality that will be integrated seamlessly into the PHNX ecosystem as it develops.

The Base PHNX-ID consists of the summation of:

- Multiple Ethereum addresses claimed by the user
- Resolvers - set by PHNX-ID owners through their Resolver dashboard
- Token allowances and transfer permissions set for each Resolver

Implications for Financial Services and beyond

The open framework of Phoenix Identity makes products, platforms, apps, and dApps built on top of the protocol integral to its long-term success. Because you can aggregate, not only on-chain identities, but different Ethereum addresses with ERC-1484, this allows a sole core identity (PHNX-ID), to be utilized across different financial products, resolvers, and on a macro level, across different industry verticals.

Payments

The Phoenix Payment is a powerful protocol that includes built-in identity and authentication systems. Built on the decentralized and public Ethereum blockchain, Phoenix Payments allows users to complete one-click authorization of debit and credit POS transactions, as well as instantaneous P2P, P2B, and B2B payments. In addition, all of these payments are secured by the Phoenix Authentication and Identity protocols, providing an unprecedented level of security.

Although Phoenix and the PHNX token lie at the core of the Phoenix Payment system, by incorporating other protocols and technologies, such as atomic swaps, Phoenix Payments is essentially cryptocurrency-agnostic.

All of the smart contracts are open source and free for anyone to use and build on, from individual developers, to large enterprises. Below we discuss Payment functionalities and how developers can integrate the smart contracts into their systems.

Subscriptions

A very compelling feature of Phoenix Payments is that PHNX-IDs permit users to set allowances for different applications. This means that once a user has tied their identity to a resolver, a convenient structure for PHNX transfer between that user and the resolver is established; in particular, it streamlines billing systems.

Subscription Smart Contract

The Phoenix Payments Subscription Smart Contract has been designed to allow businesses to bill a consumer or other business entity a recurring PHNX subscription amount through a number of features:

- Billing Plans - allows users to subscribe to a billing plan. e.g. 1,000 PHNX per month, 2,000 PHNX per month, or 3,000 PHNX per month, and the user can select from platform-generated options.
- Cancellation and Modification - users may cancel, modify, upgrade, or downgrade subscription plans within the smart contract.

- Prorated Billing - calculates the amount of PHNX within a current billing cycle. e.g. if the user gets billed on the first of every month, this function would calculate the amount of PHNX being used for X days / Y days of the month.
- Tiered Pricing - non-linear pricing. e.g. up to 20 users is 1,000 PHNX, up to 100 users is 2,000 PHNX, and up to 1,000 users is 3,000 PHNX
- Quantities - subscribe multiple people within a group to a plan. For example, say you run a hosting company through which customers host sites at a cost of $9.99 per site per month. Most customers host a single site, while some host many. You could create plans for one site ($9.99), two sites ($19.98), and so forth, or subscribe customers to a quantity of the base $9.99 plan.
- Discounts - reduce invoices by a percentage or a flat amount for every invoice, just one invoice, or for a certain length of time. Can also apply to every subscription a customer has or only specific ones.
- Trial Periods - delays payments on active subscriptions using trial periods. e.g. payment remains in a user's PHNX-ID for 20 days, and if they do not cancel the subscription it gets released.
- Taxes - if you need to collect any type of tax on a subscription, such as VAT or sales tax, you can add a percentage in PHNX that can be converted back into the base fiat currency.
- Billing Cycles - recurring billing cycles that end on any calendar year, either monthly, quarterly, or yearly. e.g. 1,000 PHNX is withdrawn from the PHNX-ID balance on the 1st of every month until cancelled.
- Multiple Subscriptions - multiple subscriptions for a single customer by subscribing them to more than one plan or different variations of the same plan. e.g. a single PHNX-ID address can be subscribed to a 1,000 PHNX per month plan and a 5,000 PHNX per month plan.
- Usage Transformation - charge X PHNX for every Y user. e.g. 1,000 PHNX for every 10 users. In this example, users 1-10 all get charged only 1,000 PHNX by the business.
- Authenticate: To finalize any subscription payment, the PHNX 2FA must be performed.

Interest Payments

The market for interest-bearing notes and accounts is enormous, and the market for interest-paying debt products is even larger. However, one of the largest problems facing the lending market is default.

Nearly 40% of all U.S. borrowers are expected to default on their balances within 5 years in the student loan market alone, and between 2008 and 2013, nearly 500 banks failed in the U.S., costing the FDIC $73 Billion to compensate for losses [15].

Phoenix Payments can solve these issues by holding interest from an issuer or borrower in escrow within a PHNX-ID, allowing counterparties to eliminate fraud, prevent default, and validate all terms on-chain.

Interest Payments Smart Contract

Features of the Phoenix Interest Payments Smart Contract include the ability to:

- Create interest rates between 0-100%
- Set the principal amount that the interest is calculated on
- Define Snowflake IDs for both the payer and the payee
- Set the accrual date for the interest payments
- Set the payment schedule for the interest payments
- Set the end date, or term, of the payments
- Remove the set amount of PHNX from the SnowflakeID wallet to an escrow contract according to the payment schedule intervals, to insure there will be no default
- Send Interest by distributing the interest in PHNX from the escrow on the Payment Schedule date
- Send the principal amount on the contract from the PHNX-ID on the End Date
- Use the Phoenix Identity smart contract to confirm receipt and payment of the interest from the payer to the payee
- Create a flag for a disputed interest payment

Coupon Payments

The Phoenix coupon smart contract allows businesses to create coupons with unique characteristics for their services, and for a user to redeem these coupons.

Coupons provide a percent-off or amount-off discount for a particular set of customers and are an important part of e-commerce, POS transactions, invoicing, and orders. An on-chain approach to coupons creates uniqueness, and eliminates fraud and disputes within the marketplace.

Coupon Smart Contract

Features of the Phoenix coupon smart contract include:

- Simple Marketplace:
- One-to-many; seller-to-buyer
- Seller is contract owner (admin)
- Seller can list items with set properties:
- Name
- Description
- Price
- Uid
- Item type
- Purchase listed items at-price by sending a transaction that:
- Calls allow-and-call for the user on Phoenix Identity

- Sets an allowance equal to the price
- Draws the corresponding allowance from the user
- Transfers ownership of the item to the buyer

Gift Cards

Expected to reach over $1 Trillion in value globally, the gift card market is one of the fastest-growing commerce market sectors. However, gift cards also have some of the highest fraud rates in e-commerce, with 9.5% of all fraud attempts in e-commerce occurring through downloadable gift cards.

With PHNX Payments, a business can create gift cards for their services, allowing a user to pay for things in the PHNX ecosystem by using pre-loaded and pre-defined amounts tied to their Snowflake address.

Gift Card Smart Contract

Features of the Payments gift card smart contract include:

- Creation of the gift card by the issuer, including total number of gift cards to be issued, unique codes on the cards, PHNX-ID for business entity, amounts available for each card, and date of expiry.
- Exchange an exact amount of PHNX for a predefined value on the gift card by the purchaser.
- Storage of an electronic record of the gift card amount in a segregated part of the Phoenix user's PHNX-ID (much like an escrow contract), to ensure that users can't use the gift card anywhere where it is not intended to be spent, and it cannot be exchanged for fiat or other cryptocurrency.
- Only the user is allowed to draw down to the exact amount of the gift card, with the gift card being 'burned' at a zero value.
- Users can authenticate the gift card ID and redeem at selected merchants tied to that PHNX-ID.
- Redemption of a gift card can only occur after PHNX Authentication.
- Swapping or transferring of gift cards under conditions specified by the issuer.

Remittance Payments

Remittance payments can be streamlined using PHNX payments and ERC-1484. Traditional services like Western Union, and even upstarts like Transferwise, charge exorbitant fees for remittance. This is because of the many middlemen involved, each taking a slice of the pie as the money is moved in between parties. Blockchain gives us the ability to create a more decentralized remittance ecosystem.

PHNX payments smart contracts can be built to allow developers to build remittance dApps that move PHNX tokens in between Phoenix addresses of users in different countries, with swaps built-in for stablecoin (and potentially fiat) on-ramps and off-ramps. Thanks to ERC-1484 senders guarantee their payment arrives, business providers can guarantee funds are available, and receivers are able to use the received cryptocurrency immediately. As decentralized cryptocurrency markets become more mature, these remittance dApps can be provided with virtually no middleman fees to the end user.

Lottery and One-Time Payments

The PHNX payments lottery smart contract has been designed so that PHNX-IDs can create unique lotteries, where rewards are in the form of PHNX tokens.

The smart contract has been built as a PHNX Escrow contract, a randomizer contract that uses an oracle for creating secure randomized numbers, with the main logic being controlled by an overarching PHNX Lottery contract.

To get over the security risks associated with creating fully-randomized numbers on the blockchain, the contract uses an external oracle, which charges a small amount of ETH everytime it generates a random number.

Invoicing

The e-invoice market is expected to grow to over exponentially in the five years, and by validating, authenticating, and cutting down on transaction fees, blockchain technology has the potential to disrupt this market.

The PHNX Payments Invoicing smart contract has been built to serve this market, by allowing a business to invoice other businesses, or consumers, for a specific amount, on a specified date, and with a set amount of PHNX.

The use of PHNX Tokens In Payments

The movement of PHNX tokens as a payment solution is perhaps the most simple and most common use case in the PHNX ecosystem; however, there are many other applications of the PHNX token within Payments, including:

1. Subscription: When a user wants to pay for a subscription or recurring payment for something, they will use PHNX tokens to pay the service providers, whether on a daily, weekly, monthly, annual, etc. basis
2. Interest: PHNX tokens can be held in escrow for interest payments, helping to prevent fraud
3. Coupon: POS companies and businesses can use the coupon smart contract to give discounts on purchases using PHNX in brick and mortar or e-commerce marketplaces
4. Remittance: ability to send money to anyone around the world by using PHNX
5. Lottery: Smart contract for random number generator where the reward is paid out using PHNX
6. Invoicing: Companies can create invoices and charge payments in PHNX on a certain date, and request a specific amount
7. Gift Cards: Brick and Mortar and e-commerce stores can preload a certain amount of PHNX and tie it to a user's PHNX-ID, which can be used to purchase physical or digital goods

The PHNX token is the driving force behind Payments and is essential to all payments through the PHNX Ecosystem. By encouraging massive business adoption through easy to use toolkits, APIs, and Layer-4 platform creation, the Project PHNX team can make PHNX an accepted form of payment globally, especially in markets with large unbanked populations. This will lead to lower fees, faster economic growth, and higher standards of living for many lower income business owners. The PHNX token will always be at work in the background to ensure the system runs smoothly and securely.

Implications of Phoenix Payments

The PHNX ecosystem, with the addition of PHNX Payments, allows for an all-in-one payment experience, from setting up accounts with Phoenix Identity to completing all types of transactions. Furthermore, by using blockchain technologies such as Chainlink, Uniswap, and atomic swaps, PHNX will power payments not only across ERC20 tokens, but also across chains and traditional payment services.

Consumer Applications

It is envisaged that an increasing number of third-party developers and enterprises will use the Payments protocols to increase the security, flexibility, and interoperability of their consumer product lines. With the above advantages in mind, the consumer will not be the product, as often seen within data-driven systems that rely on user data as a revenue stream, but rather will benefit from increased efficiency, lower fees, greater security, and access to more robust, utility-oriented systems.

The advantage of PHNX Payments in consumer applications is the ease of use for the user, who will not need to be proficient in blockchain to participate in the P2P payment revolution. Payments Payments streamline the payment process in the most efficient way possible for the user. This simplicity combined with effective security will provide a great bridge for consumer adoption.

B2B Applications

More and more startups and enterprises are moving into the blockchain space, testing and leveraging blockchain technology to their advantage. The PHNX protocols provide a shortcut for these companies to integrate blockchain technology, without the need for blockchain developers. However, legacy businesses of all sizes could take advantage of this kind of adoption as it allows them to work smarter, faster, cheaper, and in a secure manner, while also offering the benefit of decentralization, so that one business in particular cannot take advantage of the other.

Payments Partnerships

With the PHNX ecosystem beginning to flourish, certain solutions can help our ecosystem grow and become completely autonomous. When looking at Ethereum and the scalability of its main chain, microtransactions are not always going to be the best fit. Scalability solutions using layer-2 sidechains are going to be extremely beneficial for the ecosystem, allowing for quick and easy payments between individuals and businesses.

Plasma sidechains may prove to be extremely useful for Payments. By tying in this layer-2 sidechain with Payments and Snowflake (identity management protocol), people could potentially be able to pay with virtually any currency, executing the smart contract with PHNX on the backend, while the service provider or user receiving the payment receives the currency of their liking.

(Note: The Phoenix community has not yet made any official decisions regarding what scaling systems to incorporate into the Phoenix ecosystem)

Protocols like Uniswap are also beneficial to the Payments ecosystem. When conducting P2P payments, users will be able to receive payouts in $DAI (an Ethereum based stablecoin) using atomic swaps. An example of this is the Snowmo dApp, which was built using ERC-1484. This type of dApp can be built for any type of service provider, whether e-commerce or a brick and mortar store doing POS (point of sale) payments. This enables payment in PHNX tokens, while the service provider receives $DAI automatically.

Native Mobile QR Codes and NFC

The open-source nature of the Phoenix ecosystem means that smartphone manufacturers and app developers will be able to integrate secure, Payments-based payment systems into mobile platforms. With this integration, QR codes, and NFC will allow fast payments in much the same way as Apple Pay and Android Pay work now.

The open framework will make it easy for apps, such as social media apps, to integrate Payments payments solutions. By linking your Snowflake to your favorite social media account, instant payments between friends will be as simple as Apple Pay and Google Pay.

Meta-Transactions dApp

One of the core functionalities of the Phoenix Identity protocol is the infrastructure it is built upon: ERC-1484. This identity standard for Ethereum is not only DID (digital identity) compliant, but can be fully powered by meta-transactions. This allows a dApp owner to utilise a relayer service thus enabling end user interactions with the Ethereum network in a non-custodial manner. Such interactions may take the form of user onboarding or P2P (peer-to-peer) transactions

This is one of the more powerful technologies for dApp owners to utilize as it breaks down the barriers of adoption by getting users to interact with the ethereum network without requiring them to hold Ether in their wallets.

Risks

The predominant risk facing Payments is Ethereum scaling. This is not such an issue with the Authentication, Storage, and, to some extent, Identity phases of the Phoenix roadmap. Those 3 phases are more reliant on reading off the blockchain and the transactions are infrequent, making users likely to want to make the tradeoff between cost and security of completing actions on-chain. However, Payments as a payment protocol will involve many frequent transactions that may incur unacceptable transaction costs if scaling does not occur. On small transactions, these fees act

There is also a risk associated with the reliance on third-party technologies, such as Plasma or Sharding, in dApps built on top of the PHNX Payments protocols. If these technologies break down or become obsolete, then it would be essential to have a backup solution that can be implemented rapidly.

Storage

The Phoenix Storage Whitepaper is currently in the process of being updated to reflect the changes from the original Hydro whitepaper.

Tokenization

The phoenix Tokenization protocol is a framework of Solidity Smart Contracts tied to the Identity protocol allowing a verified PHNX-ID to create standardized Security Tokens that can be issued, bought, sold, validated, transferred, paid as dividends, and destroyed.

The tokenization protocol comprises four core smart contracts:

- HSTBuyerRegistry.sol — A buyer registry to hold PHNX-IDss of buyers for any security token.
- HSTEscrow.sol — This contracts stores any funds sent by the token creator and the funds that users pay to participate in the issuing. It distributes any refunds, and the corresponding dividends and interests to their receivers.
- HSTServiceRegistry.sol — The Service Registry contract has an array of token addresses, and holds PHNX-IDs of service providers for tokens, this simplifies the creation of an ecosystem of service providers.
- HSTokenRegistry.sol — Keep track of contracts for the issuance of Phoenix Securities

The sum of these contracts allow a fully operational security tokenization suite allowing the ability to;

- **Create Security Token** — unique IDs are attributed to each token, called a Phoenix Security Token or PST.
- **Define PST Rules** — create a dictionary of rules that can be applied to the PST.
- **KYC Approval** — on-chain KYC approval from off-chain KYC provider(s) of token issuer and buyers/sellers for defined ruleset.
- **AML Approval** — on-chain AML approval from off-chain AML provider(s) for token issuer and buyers/sellers for defined ruleset.
- **Limit Owners** — limit ownership percentage, or PHNX amount for any PST.

- **Legal Approval** — on-chain legal approval from off-chain legal providers to prove rightful creation, ownership, and structure of security token.
- **Legal Contracts** — tie the PST to legal contracts and terms/conditions written off-chain via PHNX Storage.
- **Restricted Transfers** — override normal ERC-20 transfer methods to block transfers of PST between wallets if not on a KYC/AML whitelist.
- **Lockup Periods** — set rules to lock token transfers and buy/sells for a period of X time.
- **Admin Function** — an admin or issuer can modify rules, whitelist/blacklist, lock, freeze, or stop token transfers at any time.
- **Participant Functions** — send and receive a token tied to ERC-1484 wallet ID, lockup, freeze, blacklist any PHNX-ID.
- **PST Escrow** — keep PHNX tokens in escrow contract within ERC-1484 of issuer, until the offering is closed, release back to ERC-1484 wallet ID of subscriber from escrow if conditions in legal contract aren't met.
- **Subscription** — use the Payments Subscription task to create a framework for payments and recurring subscriptions to a securitization.
- **Authenticate** — use Phoenix Authentication to authenticate issuance, purchase/sale, transfer.
- **Carried Interest** — calculate carried interest based on the Interest Smart Contract utility function (link when posted).
- **Interest Payout** — payout carried interest/management fee to token issuer on a set schedule to defined wallet IDs on the whitelist.
- **Dividend Payout** — payout dividend from admin pro-rata to Phoenix wallet holders in PHNX.

Authentication

The power of a distributed blockchain allows a vast improvement to authentication over traditional systems. The decentralized nature of the ledger means that a hacker would have to compromise thousands of wallets, at the same time rather than the relatively simple attack on a single point of attack.

Phoenix blockchain-based authentication comes in two parts. Client side and server side.

Client-Side Authentication

Client side Authentication enhances the security offered by traditional 2-factor authentication (2FA) systems such as Google Authenticator. The protocols allow developers to develop a 2FA system that combines the best features of traditional authentication apps and Universal 2nd Factor (U2F) systems, with added benefits.

In traditional Time-based One-Time Password algorithm (TOTP) based authentication systems a shared secret is stored both on the server of the account you want to access, and on the device you are accessing it from. This secret can't be encrypted so if a hacker gains access to that database they have the shared secrets and can bypass 2FA. In Phoenix Authentication this attack vector is removed by use of public key cryptography. The secret (private key) is only stored on the users device so, should a database be breached, the hacker cannot gain access to individual accounts.

There is a cost associated with the setting up of Client-Side authentication, however the system does not make a transaction with every authentication. The only transaction is during the initial creation of a PHNX-ID. After that systems only need to read from the public blockchain, which is a fast and free process, which leaves no footprint.

Client-Side authentication has five parties involved:

1. User - this is the end user that needs to be authenticated to access a system.

2. Application - The platform or database that a User wants to access.

3. PHNX enabled application - Any application interface that can communicate with the PHNX ecosystem as an authentication interface for the User.

4. Application Programming Interface (API) - a background system that allows the PHNX enabled application to interact with the PHNX blockchain.

5. PHNX Blockchain - A digital ledger in which PHNX transactions are made and recorded chronologically and publicly.

Authentication on PHNX is in three stages:

1. Initialization

When setting up, or initializing Client-Side Authentication, Users must generate a randomized cryptographic seed (public/private-key pair) to create a PHNX wallet within a relevant PHNX enabled application. During the seed creation process a PHNX-ID is created that is irreversibly tied to the generated wallet and can act as your identifying passport across the PHNX ecosystem.

2. Authentication Attempt

At the end of the public-key details pass through the relevant API to be recorded on-chain. At this point the user's PHNX-ID is displayed and the User can use this on the relevant Application to forge an authentication link. Once the initial set up is complete and a link is established the User can attempt authentication by asking the Application to generate a message using the Users public-key. The message is sent to the User who then responds by signing with the User's private key within the appropriate PHNX enabled application.

3. Validation

The third and final stage is the validation of the User's signed message by the Application. During this process the application requests, via the API verification that the message was signed by the correct user. If the PHNX Authentication smart contract conforms that the public address of the signed message matches that of the PHNX-ID of the appropriate User then access to the application is granted.

The creation of a PHNX-ID is not tied to any single entity. If it is possible, and encouraged, that other developers and institutions develop the ability to register PHNX-IDs and conduct authentications using their own PHNX enabled applications and APIs. An example might be a cryptocurrency exchange who wishes to benefit from blockchain based authentication by using their own mobile application as a 2FA device. The one proviso is that they must maintain a balance of PHNX tokens to execute smart contract calls.

dApp Store

Traditional app stores like the Apple and Google Play Stores are what we call centralized marketplaces. They are owned and operated by one central (usually private) authority, which at any moment can change the rules or shut it down at the flip of the switch. Over time, these centralized marketplaces retain and gather important data, learn more about you and can even sell it to others without you even realizing it.

A dApp store, or a decentralized application store, on the other hand is built on the blockchain and can be open-sourced and setup to thrive with no central command or ownership. It can be improved, secured and forked by the community. It can have new code proposed by 3rd parties and those same 3rd parties can ensure nothing nefarious is hidden within the code.

The Phoenix dApp Store:

The Phoenix decentralized marketplace for dApps has been built on the core Identity protocol. This marketplace allows a developer to build an application and submit it to the store from anywhere in the world.

You can think of our Identity protocol like this – you have a driver’s license for the United States, which works well for rules and regulations in the United States. You are now on vacation traveling to Europe or a foreign country where a United States driver’s license might not have any use. Now think about this applied to the Ethereum blockchain. With your PHNX-ID you can have identity management and control across the Ethereum network carrying around information you allow to applications and 3rd parties.

The PHNX Token:

The PHNX token plays an important role in the dApp Store as the main transactional token. Developers like any app store, can set their app to be free, or at a cost. Taking the users PHNX and giving it to the developer. Fee structures are yet to be determined as the store itself will get a small cut just like other stores out there.

The On Boarding Process:

When a user arrives for the first time, they will be prompted to follow a few steps to sign up. Like other traditional websites and services, a user needs to follow some steps before they can make a transaction. We took careful consideration as to what was necessary to make on boarding as painless as possible. Since MetaMask is a requirement, cutting edge code

Web3 libraries are used to detect if a user has MetaMask installed – and if not, the platform reacts and guides the user to get it installed.

User Wallets & Transactions:

A user transacts in the store with popular wallet management plugins like MetaMask. Anytime a user needs to transact or stamp data to the chain (buying a dApp, removing a dApp, signing up etc) MetaMask pops open and prompts the user to confirm the transaction. These transactions get displayed in a nice user interface, but are also visible through the users Ethereum wallet public address.

Architecture Scalability:

When it comes to any mainstream application, allowing future growth is extremely important. This comes by optimized modular code, a hosting environment which can handle increased machine power to load balance traffic and planning ahead with versioned releases. The dApp store one started as a prototype and since then it has gone through iterations of development to provide the necessary capacity to scale.

The Code Platform:

The dApp Store uses coding languages used by thousands of developers across the globe – Javascript. ReactJS powers the platform with hand selected web3 and other Ethereum based libraries. This allows us to utilize state of the art integrations like MetaMask, Bitski and other web3 providers.

Open Source Code:

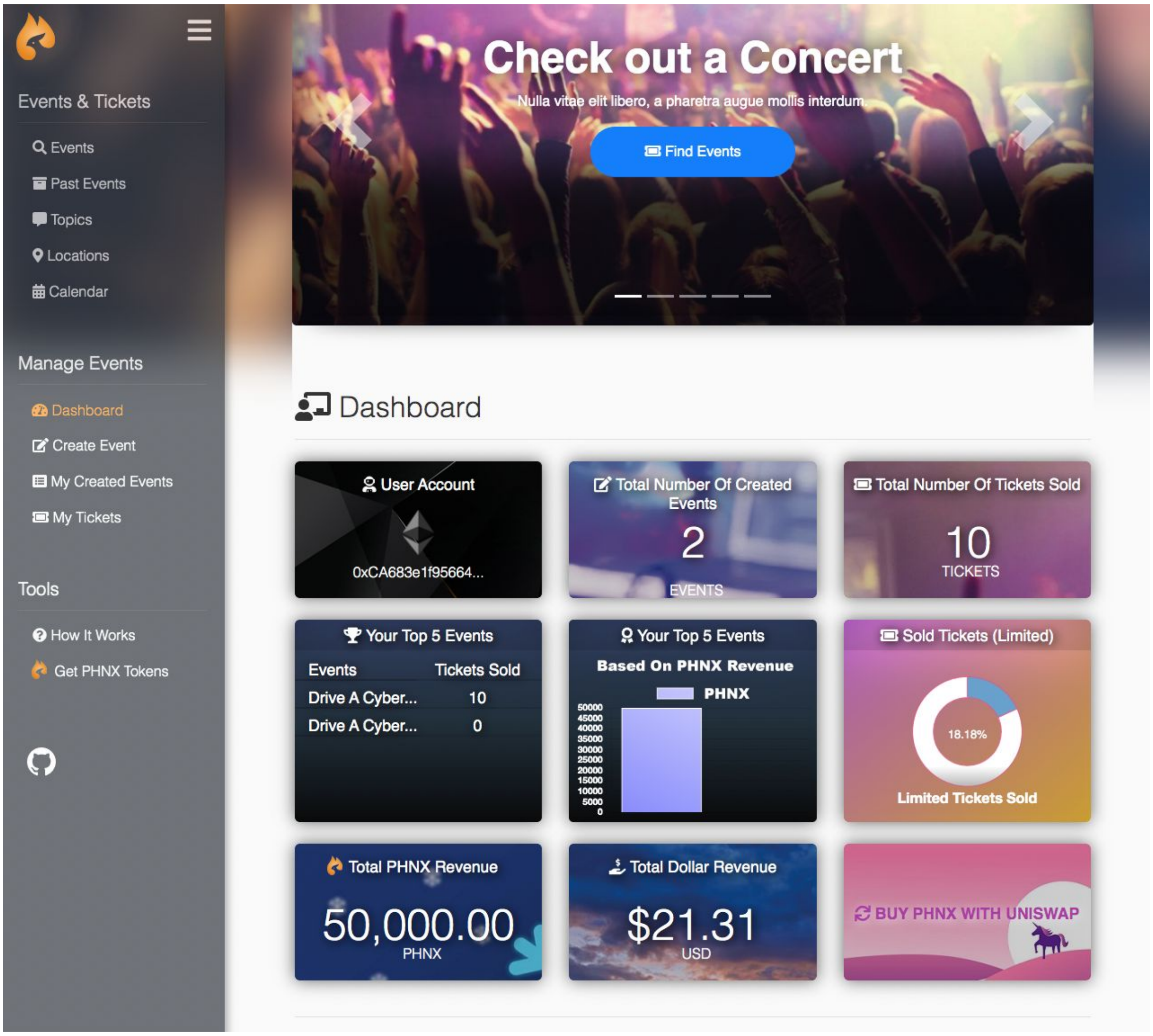
Allowing any developer to publicly see and contribute code is essential to any open-source project and anyone can create a pull request to propose changes through Github. Anyone can fork the repository and create their own version of the store.

Developer Tools:

The dApp Store includes a growing set of developer tools and resources for dev to have all the tools they need to build their first, second or third dApp. Whether you are a seasoned Solidity and ReactJS developer, or if you are new and need to get your feet wet with some tutorials, we are adding more to the scope of material to get you going without having to reinvent the wheel.

Events dApp

The Phoenix Events Marketplace is a decentralized Eventbrite style dApp where you can go to sell or find listings for ticket events.



Events	Tickets Sold
Drive A Cyber...	10
Drive A Cyber...	0

The primary goal of this dApp is to make it as easy as possible to create an event or buy a ticket in PHNX. The powerful PHNX-ID solution allows your tickets to be tied to your ecosystem identity so they can follow you around wherever you go.

Creating events is simple. Standard fields such as events location, time, place and category are all there and as you start typing in the fields, an event preview shows you in real time exactly what your event ticket will look like.

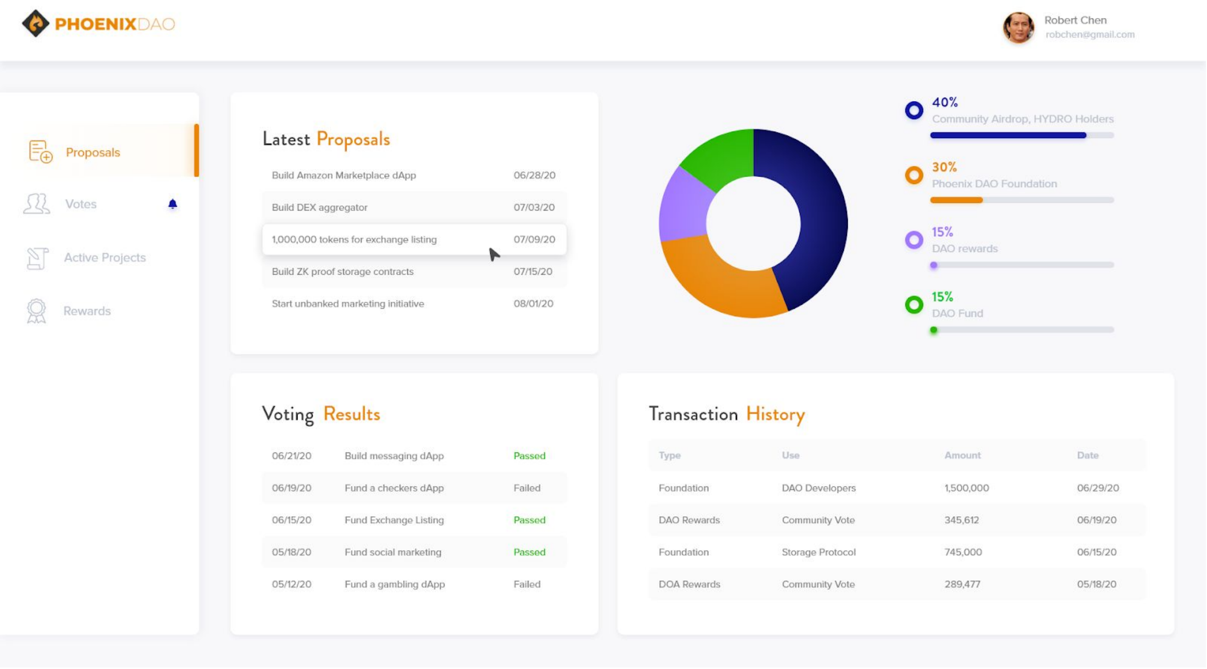
You can choose between the event being FREE, or paid in PHNX. Free events are useful because it can get people familiar with the dApp even if they are not familiar with PHNX.

Another key feature is having a ticket quantity. Maybe you want to go exclusive and only allow 10 tickets or maybe it’s 5,000 tickets to an online event.

Once your event has been created, it’s instantly live in the marketplace. The use of IPFS stores the event image and its data.

The assigned categories powers the front end of the dApp Store – each text box field is a datapoint that allows users to browse what's hot in the marketplace.

The Phoenix DAO



Latest Proposals

Build Amazon Marketplace dApp	06/28/20
Build DEX aggregator	07/03/20
1,000,000 tokens for exchange listing	07/09/20
Build ZK proof storage contracts	07/15/20
Start unbanked marketing initiative	08/01/20

Voting Results

06/21/20	Build messaging dApp	Passed
06/19/20	Fund a checkers dApp	Failed
06/15/20	Fund Exchange Listing	Passed
05/18/20	Fund social marketing	Passed
05/12/20	Fund a gambling dApp	Failed

Transaction History

Type	Use	Amount	Date
Foundation	DAO Developers	1,500,000	06/29/20
DAO Rewards	Community Vote	345,612	06/19/20
Foundation	Storage Protocol	745,000	06/15/20
DOA Rewards	Community Vote	289,477	05/18/20

To achieve full decentralization we can look no further than projects like DAOstack, Colony and Aragon, who have shown that decentralized governance is essential for decentralized networks. Not a foundation that is ultimately controlled by a corporate entity, but one that is governed and controlled by an invested network of community members.

The Phoenix Foundation will set the stage for the project to become a self operating decentralized autonomous organization (DAO).

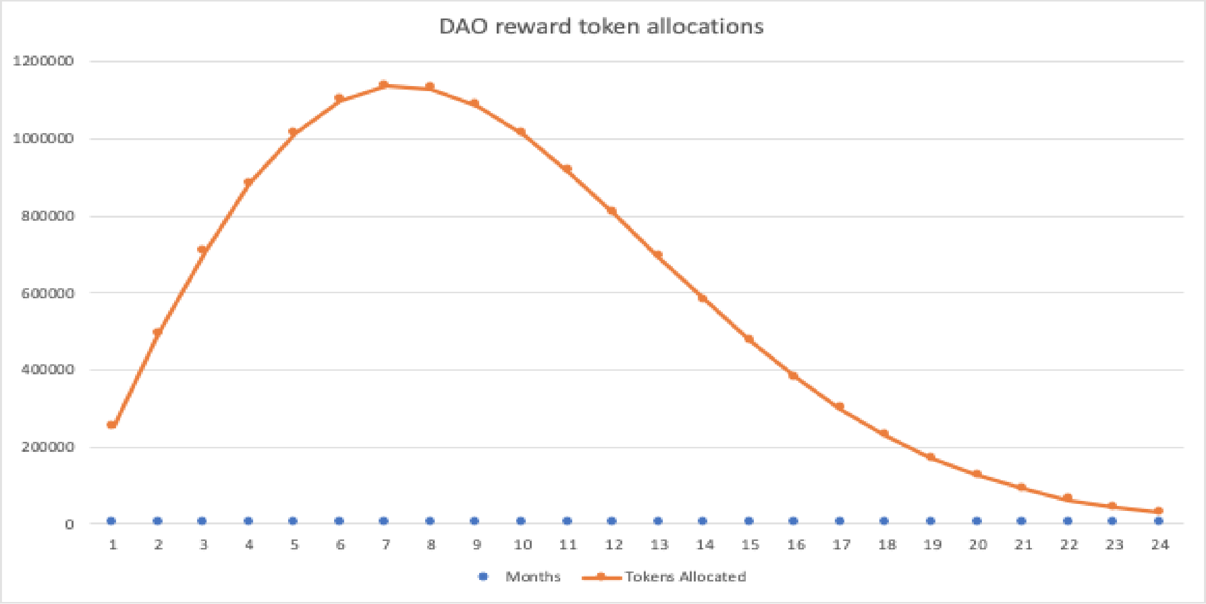
At the core of the Phoenix project is our Identity powered dApp Store and this is where the DAO will be powered. As we move forward with development the transaction structure for dApps will change and a smart contract will take 100 basis points (1%) from every transaction. This will be deducted from the merchant, receiver or resolver dApp owner, and sent into the smart DAO smart contract. As the pool and DAO membership grows then participants who stake PHNX and actively vote for proposals will be able to receive PHNX rewards from the DAO pool

Although the initial rewards will be PHNX the addition of ETH and DAI will take place as the liquidity pool grows.

What is the distribution?

The launch of PHNX saw 30% of the token supply allocated to supporting the DAO ecosystem and rewarding people participating within it.

The below table shows the DAO rewards token distribution over the initial 24 months.:



How does the Dao work?

DAO rewards are allocated based on proposals. In order to be guaranteed a reward, you have to participate in the governance process. A subdomain of the phoenixdao.io website will host the official Phoenix dApp Store.

The distribution of rewards will be based on the number of voters in the campaign and would operate in a similar vein to Ethereum grants from Gitcoin. A basic summary of how the system works is presented below;

- A person wants to see a change in how the project is run and how DAO funds are used. This could be marketing proposals, exchange listings, product development, protocol development or something else.
- That person stakes PHNX tokens and submits a proposal to the community.

- Any community member who is staking PHNX tokens gets a number of votes equivalent to their stake.
- Participation in a proposal/vote gives a 0.3% (30 basis points) return of the tokens you used to vote*.

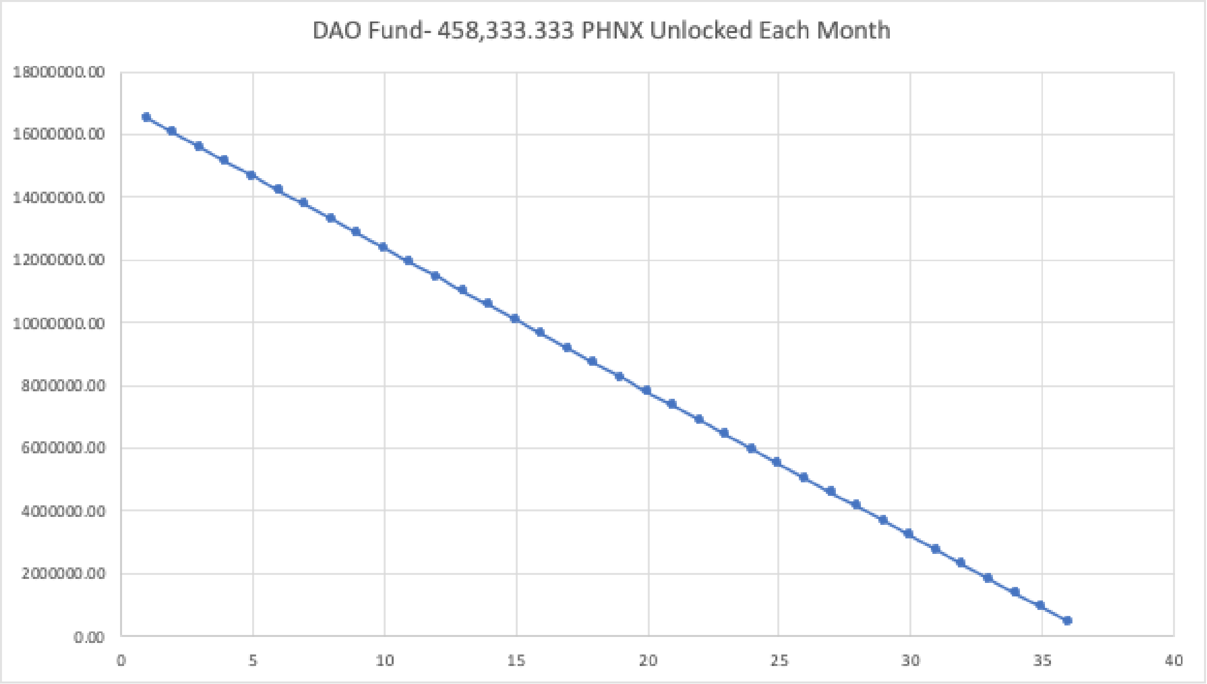
  For example, if you voted with 1,000 PHNX, this means you will receive a 3 PHNX bonus and get 1,003 PHNX in return.

*numbers are an example only and will be confirmed as the foundation and DAO is fully launched.

At its peak, there will be over 1.3M PHNX tokens that are able to be distributed to people participating in the DAO.

How does the reward structure work?

The below table shows token unlocking structure for the initial 36 months of the DAO fund:



If the token allotment is not met over the period of any single month, they are placed back into the contract for release after the 36 month period has come to completion.

To make a proposal, a DAO member will need to hold 10,000 PHNX. This will help reduce spamming of proposals by bad actors. This number can be changed by the community via the DAO voting process.

To vote you will be required to stake a minimum of 1,000 PHNX and you will need to complete KYC through your on-chain digital identity. A KYC portal will be built where users are able to complete KYC using the Phoenix Identity protocol.

Phoenix Foundation Implementation?

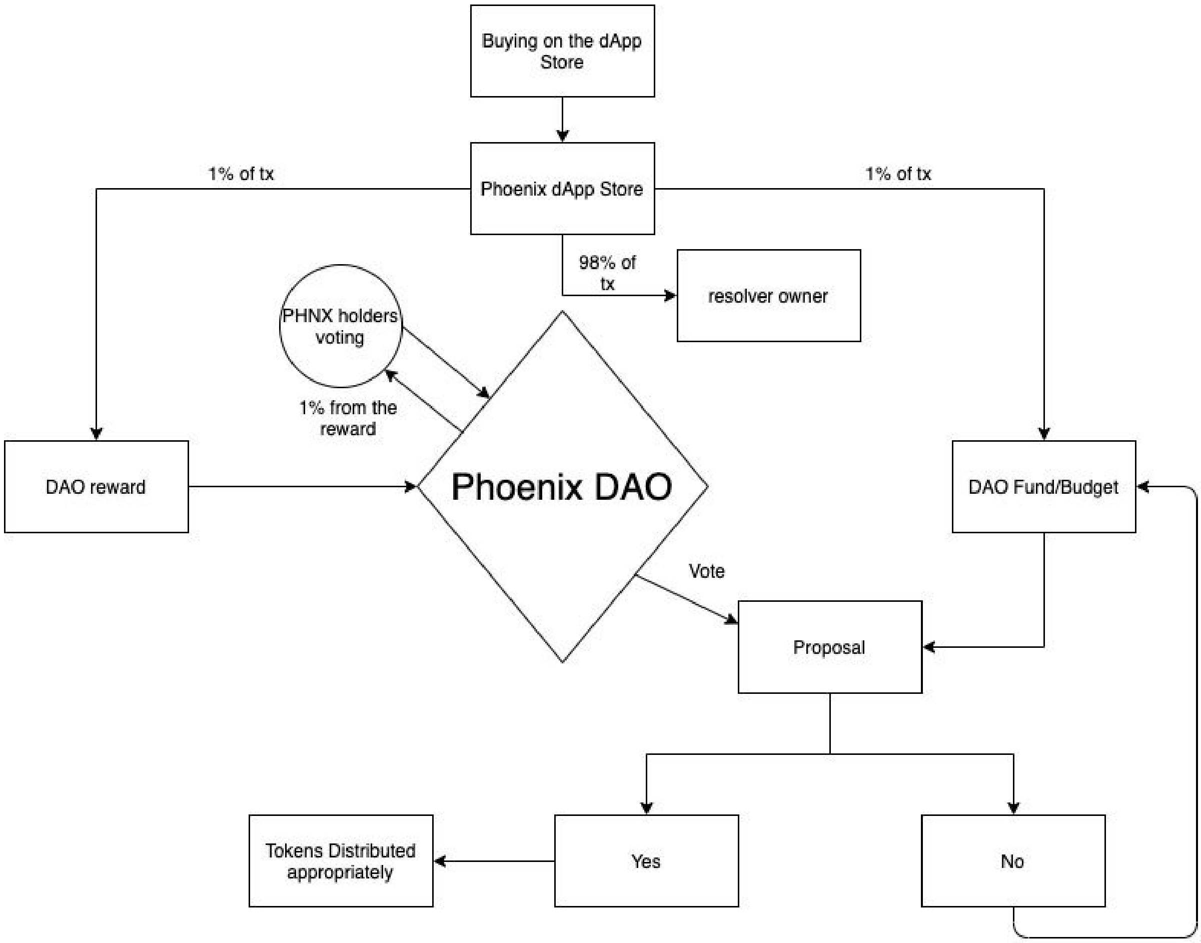
The Phoenix Foundation will be overseeing the creation of the DAO and the relevant smart contracts and structure as follows.

- The creation of the development of a dApp Store smart contract that will implement the DAO fund/rewards pool. This is done by splitting dApp Store fees into two 100 basis points transactions which are sent to both the DAO rewards pool and DAO budget pool.

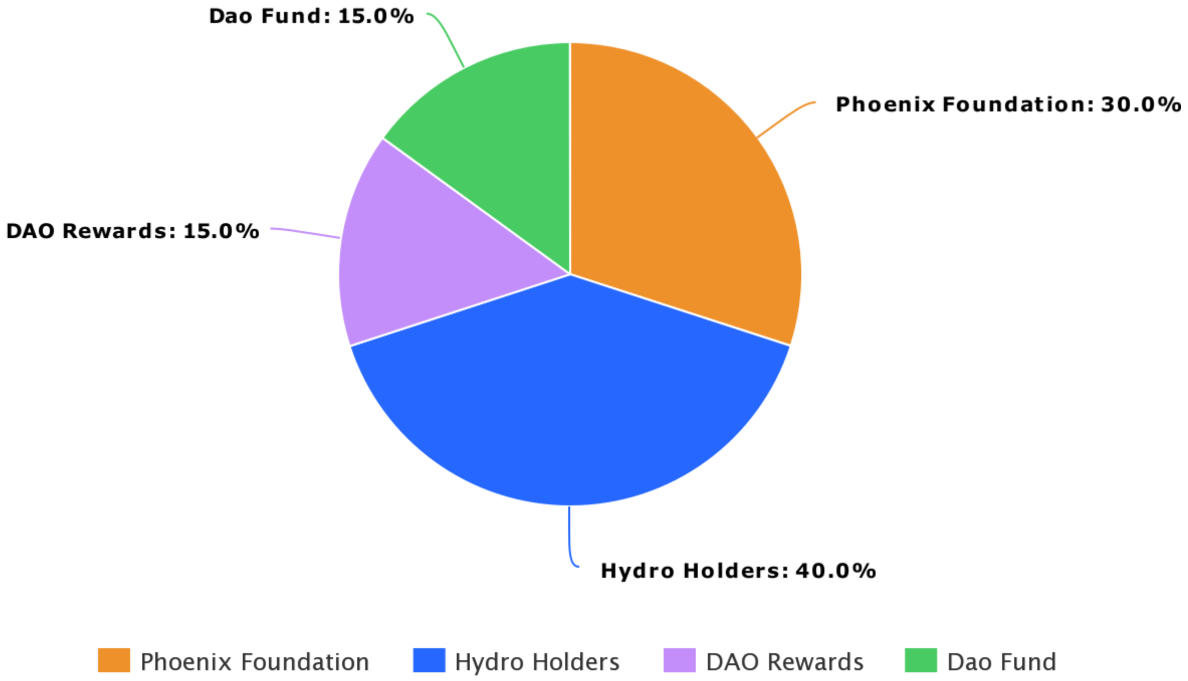
- Creating the DAO using tools provided by Alchemy (DAOstack) or Colony, and building in custom functionalities and analytics on top.

- Development of a smart contract to control the delayed release of tokens into the DAO fund.

- Development of a smart contract that distributes DAO rewards over a set period to people who vote within the DAO.



Initial Token Distribution



<u>Token totals</u>

Foundation	33,000,000
DAO rewards	16,500,000
DAO fund	16,500,000
HYDRO holders	44,000,000
Total Supply	110,000,000

Phoenix DAO Fund

<u>Tokens allocated</u>: 16,500,000

Staking/Voting Rewards

Tokens allocated: 16,500,000 PHNX

Future Foundation/DAO Initiatives

Tokens allocated: 16,500,000 PHNX

Current HYDRO Token Holders

All HYDRO wallet owners who complete KYC on Bittrex will be eligible to receive the airdrop of 1 PHNX per 100 HYDRO tokens.

All tokens that are not collected will be sent to the DAO fund for ecosystem development and future initiatives.

Tokens allocated: 44,000,000

Protocol and Phoenix dApp Store Development

This involves expanding on the current Phoenix protocols to improve contracts relating to authentication, identity, storage, and payments.

Developers will be on-boarded to create various dApps needed by decentralized services.

5 flagship dApps voted on by Foundation Founders to replace current centralized apps; such as the Events dApp created to replace Ticketmaster. The foundation/DAO will use funds from these dApps to continue operations into the future. The foundation/DAO will also take a 2% fee on all dApp sales through a smart contract which is developed, as outlined in the DAO paper.

Future dApp proposals will be accepted and voted on by the Phoenix DAO before being added to the list.

Partnerships will be set up with blockchain developer training groups and blockchain initiatives all around the world to expand the developer ecosystem.

There will be an option on the dApp Store to convert any ERC-20 into PHNX for use on the dApp store (Uniswap/Totle).

Tokens Allocated: 33,000,000

Foundation Funds Breakdown

Foundation funds will be secured in multisig wallets and smart contracts to release tokens on a staggered schedule to prevent market saturation.

Total allocated: 33,000,000

Product Development and Partnerships: 16,500,000

Foundation Development Funds: 16,500,000

- Foundation Core Team: 5,000,000
- Immediate Release: 3,833,333
- Released after 6 months: 3,833,333
- Released after 12 months: 3,833,333

Foundation Core team

5 total core team members:

- Global Communications Director
- Global Blockchain Director
- Global Marketing Director
- Global Partnerships Director
- Global Initiative Director

1,000,000 tokens locked in a smart contract for each core team member, of which 350,000 tokens are released initially, and then subsequent bi-monthly payments of 72,222.22. Foundation core team can vote to remove any other team member with a majority vote and replace them.

The community can also vote in new leadership with a 60% majority vote.

Tokenomics

Use case Overview:

- Protocol layer use cases
- Phoenix dApp Store use cases
- Phoenix DAO use cases
- Hybrid apps use cases

Protocol layer

Phoenix Payments

Token usage:

- As currency for payments, loans, interest in PHNX.

Phoenix Identity

Token usage:

- Organizations requesting a user’s identity information must pay the user in PHNX.
- Allows for dApps to be built on any vertical with an identity layer and enables them to utilize PHNX.

Phoenix Storage

Token usage:

- Fees for file storage paid in PHNX.

Phoenix Tokenization:

Token usage:

- Enables for dividend payments to security token holders in PHNX.
- Interest payments in PHNX.
- Use legal contracts from Phoenix Storage to be approved.
- On-chain approval from Phoenix Identity for KYC/AML users.

Phoenix Authentication

Token usage:

- Micro-transaction of PHNX for authenticating an action.

Phoenix dApp Store

Token usage:

- 2% of all dApp store sales will be sent to the DAO smart contracts for use in expanding the ecosystem through dApp development and future initiatives.

- dApp sales will be priced in PHNX. Option to trade any ERC20 for PHNX instantly.

- Owners of dApps can charge users additional fees in PHNX, ETH, & DAI, and 2% of the sales of these transactions go to the DAO smart contracts.

- All DAO funded flagship dApp sales will be used for ecosystem expansion.

Phoenix DAO

Token usage:

- 1,000 staked PHNX required for voting rights in the DAO.
- 10,000 staked PHNX to create a proposal.
- Voting to include changes to governance structure, leaders, protocol proposals, marketing proposals, listing proposals, funding/rewarding people for participation within the ecosystem.

Events dApp

Token usage:

- Buy, sell, and gift event tickets, which are purchased in PHNX.
- Early access on purchasing tickets can be given to PHNX token holders.

Expected Product Usage

Token usage:

- P2P transfers.
- Discount on fees for using PHNX.
- Staking of PHNX for fee reductions in product subscription models.

Conclusion

Over the past two years the community has been building and developing a robust blockchain ecosystem that has never been given the opportunity to reach its full potential. The creation of the Phoenix DAO finally removes the shackles and gives the world the chance to freely develop and build on the project on their own terms.

PhoenixDAO is a truly decentralized, open-source project that covers identity, payments, tokenization, authentication, and storage, alongside a dApp Store which sits at the heart of the Phoenix ecosystem. A core part of the PhoenixDAO strategy is to actively on-board both individual developers and enterprise level corporations to build using the protocols to power their products. This will enhance decentralization, token utility and the overall ecosystem.

Ecosystem builders and the community will find benefits across the board from this fork as we launch with a true sense of decentralization and community control backed by an extremely passionate, talented team and solid network of partnerships.

References

https://github.com/HydroBlockchain/Hydro-Docs/blob/master/Snowflake/Snowflake_DRAFT.md

https://github.com/HydroBlockchain/Hydro-Docs/blob/master/Raindrop/Hydro_Raindrop_White_Paper_English.pdf

https://github.com/HydroBlockchain/Hydro-Docs/blob/master/Tide/Tide_Whitepaper.md

Change Log

- v1.0 - N/A
- V1.1 - original whitepaper references linked for clarity