

Zap

Connecting smart contracts to the off-chain world

Whitepaper v2.1 10.10.2017

Archived copy · cryp2nova.com

Contents

1 Executive Summary
- 1.1 Problem Overview
- 1.2 Mission Statement
- 1.3 Core Objectives

2 Primary Components
- 2.1 Token Mechanism
- 2.2 Key Actors
- 2.3 Economic Incentives

3 Potential Applications
- 3.1 Finance
- 3.2 Insurance
- 3.3 Real Estate
- 3.4 Shipping
- 3.5 Dapps

4 Roadmap
- 4.1 Finances and Activities
- 4.2 Development Roadmap

5 Zap FAQ

6 Zap Architecture
- 6.1 Oracle-bonding mechanism
- 6.2 Encrypted peer-to-peer(p2p) data feeds
- 6.3 On-off Chain Data Proxy

7 Legal Considerations
- 7.1 About Synapse Foundation
- 7.2 Tokens

CRYP2NOVA

Archived copy · cryp2nova.com

1. Executive Summary

1.1 Problem Overview

Blockchain technology empowers the creation of smart contracts capable of governing any arbitrary interaction, and the decentralized application layer built atop is poised to disrupt nearly every arena of human behavior. But for smart contracts to operate based on real world events (and provide the maximum value), they must first receive real world data. For example, an insurance contract must know whether a given reported event meets specified criteria to be “covered” and then other facts in order to anticipate the level of “damages,” and a decentralized energy grid must know the use patterns (i.e., volume and time) for different customers (e.g., residential versus commercial).

Smart contracts can run algorithmic calculations as well as store and retrieve data, but because every node runs every calculation, it's simply not feasible to make arbitrary requests for data. Oracles resolve this issue by providing the results of any query to any contract. A primary driver of the value of the blockchain is its decentralization; however, the value-add from a decentralized application is greatly reduced (if it isn’t negated altogether) if there’s only a single data source. True trust requires choice; the absence of choices for data extinguishes trust and replaces it with the compulsion of “take it or leave it.” The challenge of devising a scalable method for curating oracles remains.

Prediction market platforms like Gnosis and Augur have proposed to solve the oracle problem by distributing consensus on event outcomes across their userbase. Though this may work well for certain cases, it is slow (requiring days or even weeks before event resolution), unwieldy (requiring the mobilization of thousands of users who are poorly incentivized by a relative pittance), and potentially compromised due to the highly centralized distribution of tokens (in Gnosis's case, 97.75% of all tokens are controlled by just three addresses).

Aeternity has proposed an alternative blockchain which, they claim, would allow any user to become an oracle. However, this capability is limited to yes/no questions and allows users to contradict one another in proportion to their deposited funds. The danger of enabling well funded users to decide the truth presents a severe credibility flaw and is hardly an ideal scenario.

Oraclize and ChainLink provide services for linking existing APIs to the blockchain, but are limited in that each requires data to pass through a single aggregator. They are acting as oracles, resting on their own reputation and the resulting potential for profit loss. Zap's system includes this model in its set of economic incentives, but only as one of several techniques. Ultimately Oraclize and ChainLink could be individual oracles on the Zap platform.

Though the demand for data by the blockchain is new, the demand for data is not. Consider the popular legend that following the famous Battle of Waterloo in 1815 (at which British and other continental armies defeated the French army under Napoleon), Nathan Rothschild was able to make profitable stock purchases by learning and trading on the news of Napoleon’s defeat before official word of the news

Archived copy · cryp2nova.com

reached England and competing traders. The exact method by which Rothschild got his information first is still a subject of intense historical dispute, but nonetheless the account illustrates the value of data. We recognize that value; we also recognize the value to our customers in having the best data, and the best choices to source and receive data. We intend to empower smart contracts to utilize any available information.

1.2 Mission Statement

Zap's objective is to be disruptive, driving change in a wide range of global industries, including finance, insurance, real estate, and shipping. Zap will also find applications in dynamic new distributed application protocols, providing new monetization opportunities for individuals and emerging economies. Zap is well-positioned to be the premier provider of data for smart contracts, and stands to potentially monetize any device linked to the Internet of Things (IoT).

1.3 Core Objectives

Build a Robust, Source Agnostic Oracle Network
Zap is bringing together the existing wealth of global data with the diverse capabilities of distributed applications by ensuring the secure creation of oracles.

Incentivize Oracle Creation and Curation
Zap is building a global, decentralized data marketplace and populating it with unique incentivization tools, empowering anyone to begin monetizing their data.

Fuel the Next Generation of Embedded Dapps
Zap is supplying a much-needed fundamental piece of the Ethereum ecosystem and the Web 3.0 paradigm, enabling developers to construct Dapps that simply could not function without it.

CRYP2NOVA

Archived copy · cryp2nova.com

2. Primary Components

2.1 Token Mechanism

The token distributed at the token launch is known as the Zap Token, or ZAP. This is the only time that these tokens can be created, and therefore the total supply of ZAP is fixed. In order to create an oracle or make queries for data, both providers and subscribers must bond their ZAP, locking it up in an individual oracle. They will then gain control over an oracle-specific integer value known as "dots".

Dots can be used to query their oracle or destroyed to release ZAP from their respective oracle. One Dot is equal to one query to its respective oracle, and is non-exchangeable and indivisible. A Dot is not a token, and is only an integer value. The amount of ZAP necessary to bond to produce one Dot is determined by a price/supply curve delineated by the data provider during oracle creation, and is designed to introduce several dynamic economic mechanisms for incentivizing oracle curation by creators, subscribers, and speculators alike.

Dynamic economic elements will necessarily attract speculators. This intentional design element will act as a refinement mechanism, incentivizing the discovery and publicity of potentially useful data, utilizing many of the same economic incentives as prediction markets.

ZAP bonding occurs via the oracle smart contract. Prior to bonding their ZAP tokens to an oracle, users will be able to see exactly how many dots they will receive as a result. Users can choose to produce as many dots as they are able to afford based on the predetermined price curve. There is no limit for how many times ZAP may be bonded to produce dots.

Zap seeks to not only create useful software, but also a community of those interested in sharing their data. To do this, we need to create a model that lowers the barrier to entry for repeat users. Users holding dots are not only incentivized to bring others to participate in the same data feed, but also to bring competing oracles into the marketplace as well, since the necessary bonding of ZAP will reduce liquidity, driving up the underlying value of all oracles on the network. Data providers are even incentivized to bring similar data providers onto the network, since oracles are more secure (and thus more profitable for the providers) when constructed from bundled data feeds from multiple providers.

2.2 Key Actors

Oracles

To create an oracle, a data provider selects:

- An ethereum account which acts as the provider identifier
- An IPFS key pair for data routing
- One or more data feeds
- Bonding variables that govern the derivative of the dot price/dot supply curve
- An arbitrary bond of Zap that will produce the oracle's initial dot supply

Archived copy · cryp2nova.com

Subscribers
To query an oracle, a subscriber:

- Bonds Zap to the desired oracle, redeeming a quantity of dots
- Chooses conditions upon which they will receive data
- Provides an IPFS key pair for encrypted peer-to-peer communications

2.3 Economic Incentives

We could have designed the Zap protocol such that a subscriber simply makes a payment directly to the provider. Instead, we designed a baroque and complicated smart contract middle-man. There are several reasons for this:

Escrow
The simplest function of the oracle contract is to act an escrow between provider and subscriber, holding both dots and data and releasing both simultaneously.

Speculation
A more elaborate function of the oracle contract is to provide a mechanism for a third type of actor to insert themselves within the ecosystem. A **speculator** is defined as any actor who may have no interest in using, receiving or transferring particular data, but whose interest instead lies in projecting or predicting which oracles (and their related data flows) may become more useful or in demand in the future, and maximizing their current supply of Zap. Rather than searching for data they need, the speculator will be rewarded based on his ability to locate and bond to oracles that are likely to become useful in the future, destroying dots to release Zap when the ratio is favorable compared to their entry point.

This introduction of speculation transforms the Zap Oracle Marketplace into a prediction market for data, and as such, a whole new class of behavior is likely to arise governing the search for useful, true data.

Speculation, however, introduces other security concerns. The door is now open for market participants to have the opportunity to profit without providing any data. This risk for exit scams is mediated by the fact that each oracle is tied to a unique ethereum address. Speculators and subscribers (and indeed, all participants) bond to a relatively new oracle at their own risk.

In a mature ecosystem, we would expect to find professional data providers with thousands of oracles tied to a single ethereum address, the history of which is stored on the blockchain. This acts as a sort of trustless (i.e., non-trust-based) reputation. If a data provider has been running a profitable data business for years, they stand to lose far more from running an exit scam than they would stand to gain. This trustless reputation even opens up possibilities for another revenue stream, as long standing accounts might be able to monetize their reputation by acting as a guarantor or co-signer on new data providers, renting out their reputation to give new actors a leg up. The risk remains, however, since acting as a guarantor for a bad actor may eventually reflect poorly on the guarantor or co-signer's own reputation.

Archived copy · cryp2nova.com

Bonding

Zap is introducing the economic mechanism of bonding curves for the first time into the smart contract ecosystem. No economic device like them have been released into the marketplace so far, though they have been inspired in part by Simon de la Rouviere's writings on curation markets.

The development team spent significant time evaluating possible parameters to incentivize oracle creation and curation, but decided that because nothing like the bonding mechanism has been seen before, we ought to leave the variables up to the oracle creators themselves. We expect the Zap ecosystem to become a testing ground for economic incentivization, and over time, users will gravitate toward bonding variables that incentivize both truth and profit. We look forward to watching this experimentation, and are sure that users will discover crypto-economic mechanisms beyond what we initially contemplated.

Archived copy · cryp2nova.com

3. Applications

On the Zap oracle marketplace, anyone from an individual to large corporations can create or access oracles on their own terms. Decentralized application developers will be able to create a new generation of Dapps capable of integrating “real-world” data. What follows is a short outline of how existing markets can make use of Zap marketplace.

3.1 Finance

For years, leading minds have been saying that ‘data is oil’. Google and Facebook produce valuable data through their everyday operations, and sell that data to those who are able to turn that data into profit. Traders in various markets are constantly looking for alternative data. This data, often called Alpha data, is capable of producing a profit; and is thusly highly coveted by traders. There are tens of thousands of hedge funds that are filled with quantitative and algorithmic traders (Quants) who use data in order to make their trades. Quants extrapolate metadata in order to analyze trends. With oraclized smart contracts, quants will have another tool that can be used to execute their trades automatically. Through the Zap Store, traders can access numerous data feeds, as well as request specific data that is not available through a bounty.

3.2 Insurance

We believe the insurance industry will benefit greatly from smart contracts. The ZAP store’s data marketplace will provide insurance companies with an opportunity to provide self actuating insurance, that automatically pays customers eligible for a payout. With smart contracts, insurance payouts can be predetermined, streamlining the process for insurers and consumers alike. In this scenario, doctors in a decentralized insurance network would use their private key to sign the smart contract, releasing the funds in order to pay for their service. This is an example of humans acting as oracles.

3.3 Real Estate

Commercial real estate agencies, and their clients, would stand to gain a great deal from real-time heatmaps of foot traffic which can serve as a proxy (or indicator) of shopper volume and consequently indicate the higher likelihood of retail sales. In the decentralized Zap marketplace, anyone could go on the streets with a clicker counter and track foot traffic.

Apart from heatmaps, real estate transactions can be made much simpler through the use of smart contracts. A smart contract can be linked to a data feed that will show a change in ownership of a specific property. Once this change in ownership is recognized, the oracle will automatically sign the transaction, and release the payment to the former owner of said property. This principle can be used for rentals as well.

3.4 Shipping

Shipping is one of the largest industries on the planet and given the demand for transport of goods to meet demand, generally increasing demand for certain imported goods and commodities (and in some cases, their scarcity) and increasing globalization, the shipping industry is one of the industries most

Archived copy · cryp2nova.com

immune to obsolescence. The magnitude of commerce is enormous. Billions of packages are shipped around the world each year, and yet millions of those packages “disappear” due to factors including inconsistent tracking methods or controls, carelessness, or even outright fraud or theft. Smart contracts can be used to track shipments, inventory and indeed, any economic activity characterized by a process (e.g., construction, assemblage, even large-scale financial reporting). By empowering smart contracts to track the movement of goods and allow for verification of shipment, customers can gain the comfort of real-time knowledge of delivery, while financial institutions extending letters of credit to secure transactions pending performance may be able to reduce the risk to their capital of nonperformance or other loss of product. Shipping companies can create an escrow smart contract that the customer can deposit into, and are only paid upon delivery of the package.

Furthermore, the shipping companies can use Zap as an alternate source of income by publishing their data as an oracle and allowing others to access the data. The data that generated by shipping companies is immense, and is already used to determine market trends.

3.5 Dapps

Decentralized applications are protocols that leverage smart contracts and blockchain technology in order to create ‘unstoppable apps’ that don’t require middlemen such as Apple’s App Store or Google’s Play Store in order to operate.

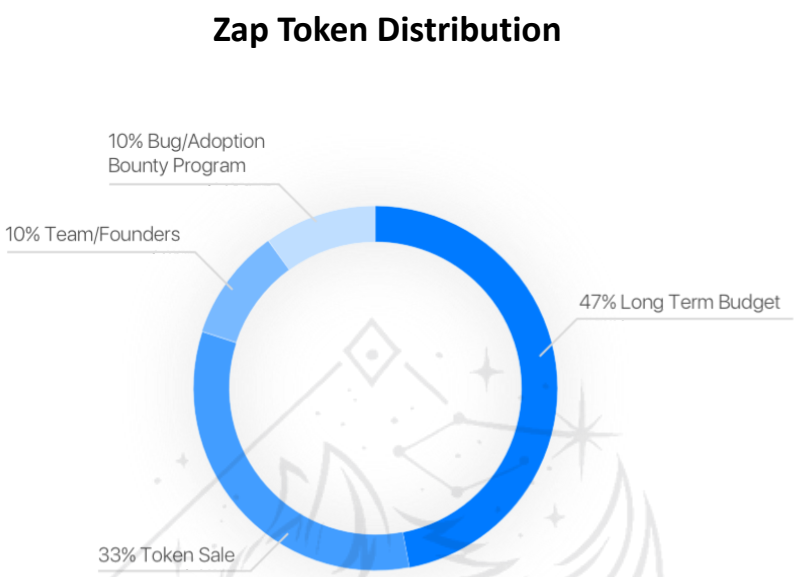
One of the Dapps that is outlined in the Ethereum whitepaper is crop insurance. A farmer could insure their crop through a Dapp that is connected to an external data feed. The Ethereum whitepaper states that: “If a farmer in Iowa purchases a derivative that pays out inversely based on the precipitation in Iowa, then if there is a drought, the farmer will automatically receive money and if there is enough rain the farmer will be happy because their crops would do well”. While this is an ideal use for a decentralized application, there is currently no way to build this Dapp. It requires an oraclized data feed to function, in this case a feed of the amount of precipitation in Iowa.

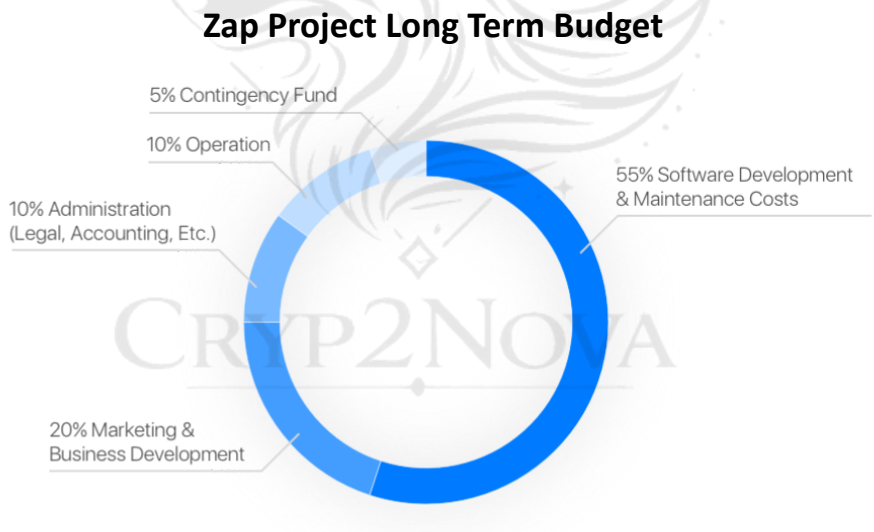
Currently the only Dapps that are available are those that operate only based on information on the blockchain. These Dapps, while valuable in their own right, have not even begun to scratch the surface of what Ethereum, smart contracts, and Dapps are capable of. In order to make Dapps and smart contracts capable of the feats that are outlined in the Ethereum whitepaper, there needs to be a method of making off-chain information usable by smart contracts. The Zap oracle marketplace is the solution.

Archived copy · cryp2nova.com

4. Roadmap

4.1 Finances and Activities

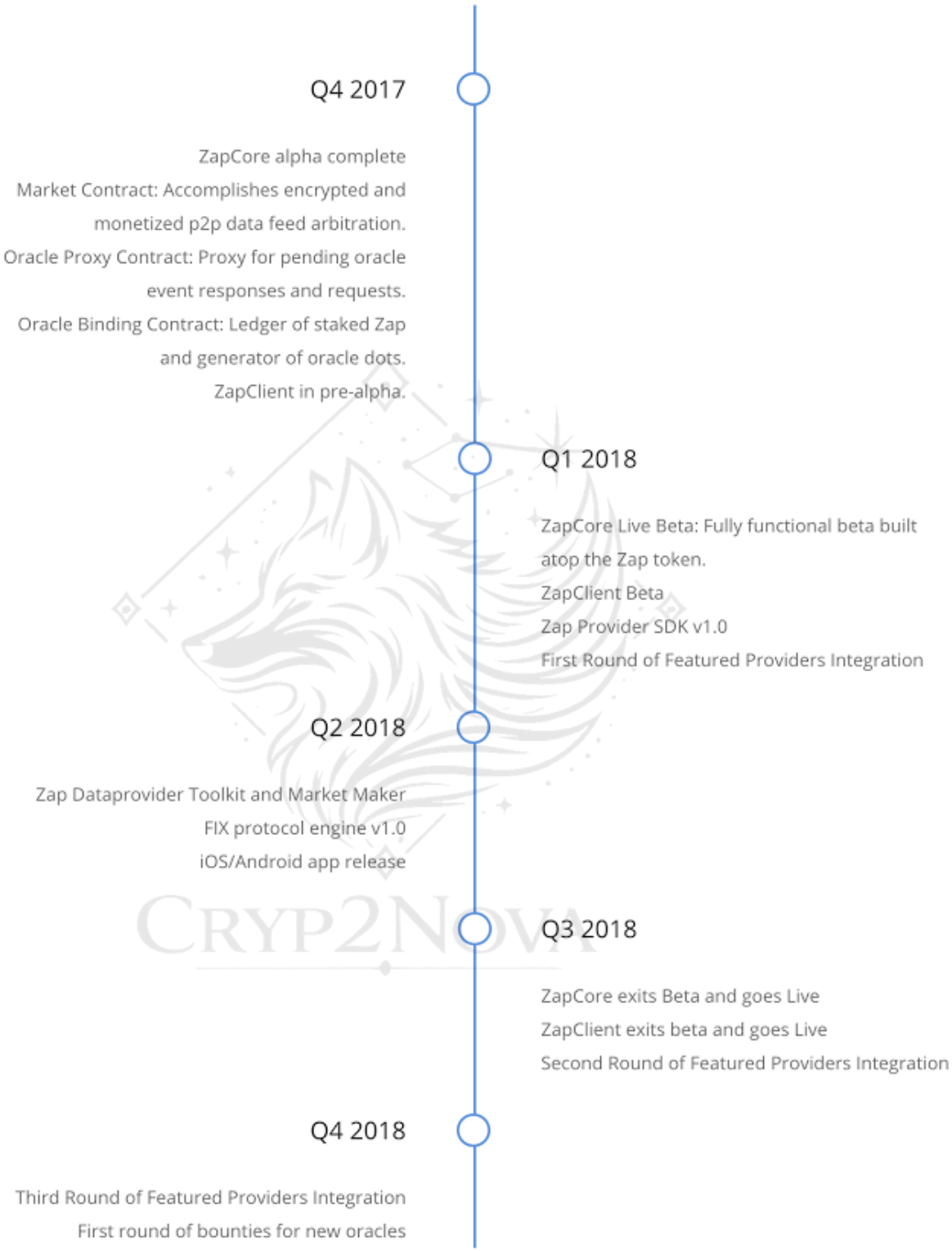




Our public beta web and mobile applications are already functioning on the Ethereum testnet and available on github. We raised over two million dollars in pre-sale, and have over 18,000 data providers ready for implementation on the Ethereum mainnet. Based on community feedback, we've abstracted our oracle creation protocol, expanding our possible range of economic incentivization mechanisms, leaving all variables up to the end user. We are well on our way to launching the first truly free market for data and oracles. We expect to continue to develop this global network.

Archived copy · cryp2nova.com

4.2 Development Roadmap



Archived copy · cryp2nova.com

5. ZAP FAQs

What is Zap?

Zap is a token based on Ethereum technology. It is the only mechanism for bonding to oracles in the Zap ecosystem and producing dots.

What do Zaps represent?

The Zap is a tokenized digital asset which conveys a legal right enabling the tokenholder to access a new blockchain based data market. The tokenholder's rights are solely of the nature of a customer purchasing a good or service, and of a licensee, and those rights (and all other terms and conditions regarding the tokenholder's relationship with, and rights regarding, the tokens) are detailed in the Network Access License Agreement which is posted on our website and which you are strongly encouraged to read. All purchases of Zap tokens are subject to the Network Access License Agreement. All holders of Zap tokens will be deemed to have entered into, accepted and agreed to all terms and conditions of the Network Access License Agreement by their purchase and acceptance of Zap tokens. All holders of Zap tokens take possession of their tokens subject to such Network Access License Agreement.

What can I do with the Zap tokens?

Every holder of Zap tokens has the full range of rights to use the tokens to access and monetize data feeds within our network. The Network Access License Agreement, available on our website, sets forth these rights.

Can I get a refund of my Zap token if I don't use the network?

No. Zap tokens are utility tokens and a consumer product. A limited number of Zap tokens will be offered for sale. As such, your possession of tokens, even if you choose not to use them for any reason, restricts the ability of other interested consumers to use the tokens and participate in our network.

How much of the company do I own with each token?

The Zap tokens carry no ownership right or stake in the Zap Foundation, the legal entity which is creating and selling the tokens and overseeing the development, management and operation of the network and marketplace, or in any other entity. Zap tokens are utility tokens and a consumer product. You have rights as a consumer and participant in the network and marketplace.

Can the Zap token rise in value?

The tokens allow for access to and participation in the network and marketplace, within which all holders (or participants) may create, market and monetize data feeds. The value of those feeds, and thus of a

Archived copy · cryp2nova.com

participant's own commercial activity, may increase as a participant realizes greater success (for whatever reason) in monetizing data. Please remember that Zap tokens are utility tokens intended to be used. The tokens are not in any way analogous to securities, investment contracts or comparable ownership interests. They are not designed for (and we neither endorse nor encourage) speculation. There is no representation, promise, suggestion, inference or implication that Zap tokens have or will ever hold a particular value beyond the utility of the token to access and participate in our network, and the consequent ability of any holder of tokens to use our network for commercial or personal use. Zap tokens give no rights in any company and do not represent any ownership right, creditor right or any right to participate in any distributions, dividends, income streams, profit shares or any other type of monetary or equity interest in any company, entity or project. Zap tokenholders also have no right to participate in the governance or management of any company. Zap tokens are distributed as a functional product. Synapse Foundation management, or its designees, have the exclusive power to operate, manage and direct the operations of the network and marketplace, without any input from tokenholders.

What amount is being raised? What's the cap of tokens? Will there be a follow-on offering?

We intend to sell to the public up 520 million tokens. We do not plan at this time on having a follow-on or other subsequent offering of these or any other tokens or other participation rights in our network. This means the number of Zap tokens, and public access to the network and marketplace, will be strictly limited.

What crypto-currencies are accepted in the crowdsale?

ETH will be the only crypto-currency or digital asset accepted for the purchase of tokens in the crowdsale. You will be required to have an Ethereum wallet pointed at the token/crowdsale address to participate in the crowdsale. ZAPs are Ethereum derived tokens. If you hold BTC or some other crypto-currency it can be exchanged for ETH and used to participate in the crowdsale.

When will the Crowdsale happen? The crowdsale is scheduled to begin October 21st at 4pm UTC.

What rights to Zap holders have?
Purchasers of Zap tokens have the rights as consumers and holders of a license right to access the network and marketplace, as set forth in the Network Access License Agreement. Tokenholders have no right to participate in or otherwise have any say in the management of the Foundation, the network or the marketplace.

Are ZAP tokens transferable?

Zap tokens may be immediately bonded to produce an oracle. Zap tokens are not designed with restrictions on transfer. Zap tokens are designed and intended to be used in the network and marketplace.

Archived copy · cryp2nova.com

6. Zap Architecture

At its core, Zap is centered on a provider-subscriber model administered by the **ZapMarket** contract. The ZapMarket contract acts as a registry for data bids and asks, allowing peers to enter into real-time, encrypted p2p communications via IPFS Publish/Subscribe API and for data providers to monetize their streams via payment channels for subscriptions and token payments for one-time purchases.

A ***provider*** is comprised of

- Realtime datafeed
- Ethereum address registered with ZapMarket contract
- IPFS keypair public key with ZapMarket contract
- Data encryption public key registered with ZapMarket contract
- Quantity of ZAP bonded to the ZapMarket contract

A ***subscriber*** is comprised of

- Ethereum address registered with ZapMarket contract
- IPFS public key registered with ZapMarket contract
- Data encryption public key registered with ZapMarket contract
- Quantity of ZAP bonded under an existing provider in the ZapMarket contract

Web3 and IPFS apis are integrated into a common client, shared by both providers and subscribers, and capable of interfacing with each of Zap's fundamental subsystems:

Oracle-bonding mechanism, where access tokens (dots) for a given provider are generated by bonding ZAP tokens to this registered provider

Encrypted peer-to-peer(p2p) data feeds, monetized via the ZapMarket contract

On-off chain oracle data proxy via the ZapDataProxy contract

Merkling of oracle event request and hash anchoring for validation of historical data, accomplished as a service provided by the Zap network

6.1 Oracle-bonding mechanism

Provider registration, subscription payment, and oracle event payments are all accomplished via Zap's Oracle-bonding mechanism. This process can be largely masked on an application layer, providing most subscribers with a traditional cryptocurrency payment flow and providers with a registration bond which can be redeemed.

CRYP2NOVA

Archived copy · cryp2nova.com

Provider registration*
Would-be data providers must register with the Zap market contract to become available to subscribers. Interactions with subscribers are arbitrated here, by way of provider-specific access tokens (dots) generated by bonding ZAP under a given provider's ledger in the market contract.

To become listed, the provider customizes a *supply curve* of ZAP/dots-produced (as discussed in earlier sections) and places a bond of ZAP, thus generating an amount of access tokens corresponding to ZAP/dot ratio at the supply curve's origin.

These dots can be produced by would-be subscribers via a subscriber bond of ZAP to the provider. Provider registration*

Subscription and oracle event payments
To subscribe to a data feed, the subscriber must bond an amount of ZAP to the provider. This can represent a payment to the provider (provider is able to burn a dot to redeem a ZAP while maintaining ZAP/dot point on supply curve) or an increase in total ZAP bonded (moving the ZAP/dot point on supply curve)

*The process by which a would-be data provider registers a listing on the Zap platform is discussed in the next section.
**dots are non-transferrable.

6.2 Encrypted peer-to-peer(p2p) data feeds

The Zap client and Zap market contract facilitate socket connections between any peers wishing to share data. What is accomplished are bi-directional data channels relying on a Publish/Subscribe model between IPFS peers.

Providers and Subscribers swap public keys via the market contract and establish a private IPFS pubsub room through which data, encrypted via subscriber's public key, is pushed by provider. Tokens to be bonded are held in escrow until termination of the channel.

What follows is a breakdown of provider-subscriber flow as handled by provider and subscriber via the Zap client and Ethereum contract.

Archived copy · cryp2nova.com

Provider

```
 create new market contract instance
 set network provider
 set ethereum private key

 create provider instance
  group = "NYCBarometers"
  subscription_rate=200
    const provider = new SynapseProvider(group,
subscription_rate);

  for existing provider
set ECDH private key from config
load subscriptions from config
listen for subscription events
for each subscription
 publish data

  for new provider
generate ECDH keypair
save to config file

TRANSACT with market contract, returns bool
   Market->registerSynapseProvider(bytes32   group,
uint256 public_key, uint256 syn_rate)
on success
 listen for subscription events
 for each subscription
 publish data

  Provider is now listening for new subscription
and subscription terminations
```

Subscriber

```
 create new market contract instance
 set network provider
 set ethereum private key

 create subscriber instance
  provider index
  provider group
   amount of syn to pay ( defines subscription
period)

 if subscription config file exists
  EDCH private key from config file
  load subscriptions from config file
  subscribe to pubsub channel

 if new subscription
  group = "NYCBarometers"
  newSubscription(group)

  CALL to market contract, returns provider index
Market->requestSynapseProvider(group)
  on success:
CALL to market contract, returns provder address
 Market->getProviderAddress(group, index)
  CALL to market contract, returns provider public
key
Market->getProviderPublic(gourp, provider_index)

  generate random pubsub channel id
  encrypt channel id with provider's public key
  create nonce

  TRANSACT with market contract
initSynapseDataFeed(
  group,
  providers_address,
  public_key,
  euuid_hex,
  nonce_hex,
  amount
 )

on success
 create new subscription
 market contract emitts SynapseDataPurchase event
```

Archived copy · cryp2nova.com

Provider	**Subscriber**
receive SynapseDataPurchase event containing subscriber's public key, encyrpted pubsub channel decrypt pubsub channel encrypt data to be sent over channel via subscriber's public key push data over pubsub channel	Awaiting provider

Provider and Subscriber are now in encrypted communication in a hidden channel over IPFS

6.3 On-off Chain Data Proxy

Users who wish to leverage providers' data channels for input into their smart contracts can import the ZapDataProxy contract, which proxies data supplied by a provider to any custom callback function, releasing a user-supplied bond of ZAP to the provider, upon delivery of requisite data. (Note: compatible feeds must be time-series objects of key value pairs.)

To interface with the ZapDataProxy, an event containing the desired key names of provider feed, and the time or key value of time-series data for the provider to wait/listen for must be fired from the contracts constructor.

```
event OracleEventRequest( address providerAddress, uint zapStake, string keyNames, string trigger)
```

where

`providerAddress` is ethereum address of the provider

`zapStake` is ZAP to be bonded

`keyNames` is comma separated list of data feed keys mapping to desired value

`trigger` is either

unix timestamp at which provider is to deliver values at keynames prefaced by `'t': t1507851944`

Comma separated list of values corresponding to keyNames parameter: `23,12,54`

and a callback function, `_callback` with which to supply the providers return data, with a function signature capable of digesting the desired data.

ZapDataProxy compliant user contract

```
import ZapDataProxy

contract CustomUserContract{
 function CustomUserContract(address providerAddress, uint zapStake, string keyNames, string trigger){
  OracleEventRequest(address providerAddress, zapStake, keyNames, trigger)
 }

 function _callback( bytes32 response){
//custom business logic
 }
}
```

Archived copy · cryp2nova.com

7. Legal Considerations

About Synapse Foundation

The Synapse Foundation is an Isle of Man non-profit organization with offices in Zug, Switzerland. The Synapse Foundation was formed to build, promote, and oversee the Zap project and engage in activities in furtherance of those objectives. The organization is committed to community engagement and sponsoring user participation within a world of decentralized data. The Zap token and associated network and marketplace all use trade secrets and intellectual property either created, owned or licensed by the Synapse Foundation.

Legal Disclaimers

Nothing stated in the Zap technical white paper is to be construed as financial, taxation, investment, legal or other advice. This white paper does not constitute an offer or invitation in any place which, or to any person to whom, it would not be lawful to make such an offer or invitation. ZAP tokens and related authentication tokens generated by the platform are not being offered or sold to residents of the United States, Hong Kong, or the People's Republic of China. If you are uncertain about whether participation in any token distribution event is appropriate for you, you should seek the advice of your own legal, tax or other qualified professional.

This Zap technical white paper is for information purposes only. The Synapse Foundation does not represent, warrant or guarantee the accuracy of or the conclusions reached in this white paper, and this white paper is provided "as is". The Synapse Foundation does not make and expressly disclaims all representations and warranties, express, implied, statutory or otherwise, whatsoever, including, but not limited to: (i) warranties of merchantability, fitness for a particular purpose, suitability, usage, title or noninfringement; (ii) that the contents of this white paper are free from error; and (iii) that such contents will not infringe third-party rights.

The Synapse Foundation, its affiliates and their respective officers, directors, owners, partners, consultants, contractors, attorneys, agents and employees shall have no liability for damages of any kind arising out of the use, reference to, or reliance on this white paper or any of the content contained herein, even if advised of the possibility of such damages. ZAP tokens and any related authentication tokens are functional utility tokens within the Synapse platform and their purchase for any other use is not recommended or endorsed. ZAP and any related authentication tokens are non-refundable. ZAP tokens and any related authentication tokens convey, possess and have attached no rights beyond the Network Access License and the **Network End User License Agreement** that governs the use of each token.

Holders of the Tokens or platform tokens shall have no rights in equity, title or interest in any dividends or distributions of income, gain or profits from the Company or its affiliates; shall not benefit from, share in, receive or otherwise participate in any capital appreciation in or of the Company or its affiliates; shall

Archived copy · cryp2nova.com

not benefit from, share in, receive or otherwise participate in any distribution of any assets upon a liquidation or dissolution of the Company or its affiliates; shall have no rights, title or interest as either a shareholder, noteholder, bondholder or creditor of the Company or its affiliates; shall have no rights, title or interest to participate in any other transaction of the Company or its affiliates, except as ordinary commercial transactions as a retail customer; shall have no future rights, benefits or privileges in the Company or its affiliates, including any rights in the future development of the Company or its affiliates or any rights to any intellectual property or trade secrets of the Company or its affiliates, except the License granted under this Agreement; shall have no rights to refund or sell back an unused token or anything else which could be received upon conversion, exercise, purchase or redemption of such tokens, or any part thereof; and no rights to any refunds, credits, exchanges or other compensation, or to any new license or new Token, if any token's conversion or exchange or "mining" right should expire or be terminated.

Tokens

ZAP tokens and any related authentication tokens are not interests or participation rights in the Synapse Foundation or any affiliate, and holders of the ZAP tokens and any related authentication tokens shall have no rights to vote as to the affairs or management in said Foundation or any affiliate, no rights to share or participate in the revenues, capital gains or any distributions or dividends of any entity, nor any rights as a lender, creditor or guarantor. Holders shall have only the rights of the network access license represented by the ZAP tokens and any related authentication tokens.

ZAP tokens are sold as a functional good and all proceeds received by the Synapse Foundation may be spent freely by Synapse Foundation absent any conditions. ZAP tokens are intended for experts in dealing with cryptographic tokens and blockchain-based software systems. In no event will the Synapse Foundation or its affiliates be liable to any person or entity for any damages, losses, liabilities, costs or expenses of any kind, whether direct or indirect, consequential, compensatory, incidental, actual, exemplary, punitive or special for the use of, reference to, or reliance on this white paper or any of the content contained herein, including, without limitation, any loss of business, revenues, profits, data, use, goodwill or other intangible loss.

CRYP2NOVA

Archived copy · cryp2nova.com