Pocket Network: A decentralized relay network built on Ethereum for enabling blockchain transactions over network protocols

Luis Correa, Pabel Nuñez, Michael O'Rourke
July 25, 2017

Abstract

Pocket Network is an Ethereum-based decentralized relay network that allows individuals to send any blockchain based token over any network protocol (HTTPS, TCP/IP, etc). Pocket is blockchain agnostic - any blockchain based token can be transferred using Pocket Network. Pocket Network must show dynamic scalability; it must adapt to short surges in popular demand. Integrating an ERC20 compliant Pocket Token (PKT) allows incentives to align in such a way that the network is decentralized and self-sustaining. The goal is to give developers the ability to leverage and facilitate the transfer of any publicly verifiable blockchain based token without worrying about server side security. This paper presents the specifications for such a relay network.

Contents

1 Introduction

The history of being able to purchase, receive and use blockchain based tokens has been a difficult and bumpy one. Though much progress has been made, there are still only a few on-ramps (exchanges, word of mouth) into owning some sort of token [1]. Not only that, but those who seek to purchase tokens from exchanges are a self-selecting group - generally technical and early adopters.

For a blockchain's invaluable traits to be leveraged, its tokens need to be in the hands of hundreds of millions of people. To realize this future, mobile applications must be able to interact with smart contracts and transfer tokens with ease.

Pocket Network aims to increase the surface area of adoption by making it easier for mobile developers to create peer to peer applications using blockchain based tokens. The number of ways that people can receive tokens are increased through the number of applications built using the Pocket Network.

Current forms of sending transactions over network protocols rely on centralized companies vulnerable to many forms of attack and overhead costs. This includes focusing a huge amount of resources on security [2]. This is something that most indie developers cannot afford as centralized companies managing keys for customers are honeypots for hackers [3, 4]. Companies that are able to provide services or applications for mobile devices are limited to:

- Simple wallet functionality (Jaxx [5], Blockchain.info [6])
- Limited to a single token transfer (Infura.io [7])
- In hyper competitive markets (Status.im [8], Toshi [9])

With the exception of Status.im, users of the above applications depend on centralized REST API's. With Pocket Network, all nodes in the network can act as an endpoint for any blockchain based token. With a dynamic minting structure, we expect a market of Relay Nodes to spin up support for any token based on demand (similar to AWS microservices).

There are very few Decentralized Applications (DApps) and peer to peer applications accessible through mobile devices. Partly due to the difficulty of running nodes client side; Each time a user opens an app, they need to sync up the current blockchain completely if running a light node. If the client is not running a light node, then it needs to depend on a centralized infrastructure vulnerable to attack and with high costs to indie developers and small teams. With Pocket Network, as long as a Relay Node in the Pocket Network supports your chosen token, the transaction can be relayed using a simple JSON format.

Pocket Network is a decentralized relay network of REST API endpoints that are intended to give any developer the ability to create a peer to peer mobile application of their own while utilizing any blockchain they wish. The only utility that Pocket Network wishes to offer is to enable a decentralized and scalable relay network for the transfer of any other blockchain token from one user to another. Developers can leverage Pocket Network to create peer to peer applications. They only have to worry about signing and sending the transaction over their chosen protocol. Developers will not have to build out centralized infrastructure to protect a user's tokens because security is relegated to the client.

2 Specification

Similar to networks offering decentralized storage [10], Relay Nodes in the Pocket Network offer resources in the form of REST API endpoints to assist developers in creating peer to peer applications. To register as a Relay Node you must access the API Directory contract (Section 3.1). These endpoints receive signed transactions from client applications and proceed to relay them to their corresponding network.

While most networks whose nodes offer a service generally receive a fee, the Pocket Network offers a solution that will require zero fees for developers wishing to use the network and Relay Nodes' services. The Pocket Network will implement a staking and minting mechanism between developers and Relay Nodes. Oracles will serve as a source of truth for external transactions being relayed. Oracles will not be needed for Ethereum based relays.

2.1 Staking

Developers, Relay Nodes and Oracles wishing to use the Pocket Network will need to acquire PKT and stake them using Pocket Network contracts. Interested parties will be able to stake PKT on behalf of business or project wishing to build on the Pocket Network. Staking contracts will have a minimum amount of time (TBD) before any party can withdraw.

2.1.1 Developer Staking

Developer staking plays a key role in regulating the inflation for the Pocket Network. Developers will have the most at stake compared to Relay Nodes and Oracles. As a product starts generating more transactions the developer must stake more PKT to handle them.

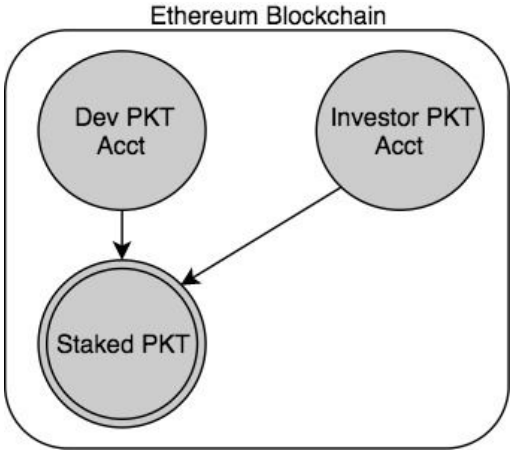

Figure 1: Setup of staked funds. Grayed out circles represent user accounts on the Ethereum Blockchain. Grayed out double circle represents a smart contract that holds staked PKT.

It is important to point out that this enables a unique proposition for developers starting new peer to peer businesses and applications. Part of the cost of starting a tokenized businesses is purchasing and staking PKT. As a result, if the

business fails developers still own their tokens and are not used up in fees. This drastically lowers the amount of risk any entrepreneur needs to take when there is a safety net should they fail.

2.1.2 Throttling

Staked PKT will act as a throttle for the amount of transactions that the developer's application, Relay Node and Oracle can handle. As the network grows along with the peer to peer transaction count, more PKT must be staked to allow for a greater number of transactions. Client applications will either be paying one another or wagering blockchain based tokens in some fashion.

It is important to note that the transactions sent can be from any blockchain. Pocket Network is blockchain agnostic. The client application must have the correct wallet functionality, and the Relay Node Endpoint must be running a node for the given blockchain to be able to relay transactions.

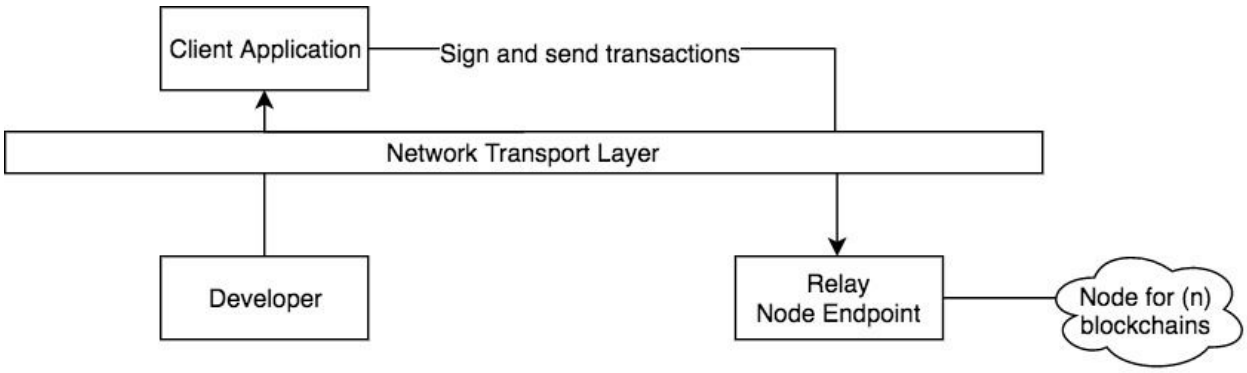

Figure 2: How client applications sign send transactions to Relays. The Relay Node Endpoint must be running a node for the given blockchain of transaction it receives. Transactions will be sent securely over the Network Transport Layer (HTTPS likely). This is a true peer to peer model - the developer has no contact with the Relay Node Endpoint.

Every time a transaction is sent from a client, a Relay Node Contract will create a Relay Contract. If successful in creating the Relay Contract, the Relay Node will send the transaction to the given blockchain. The Relay Node will keep track of a `count` of allowed transactions. Should the maximum amount of transactions be reached the Relay Contract will error out.

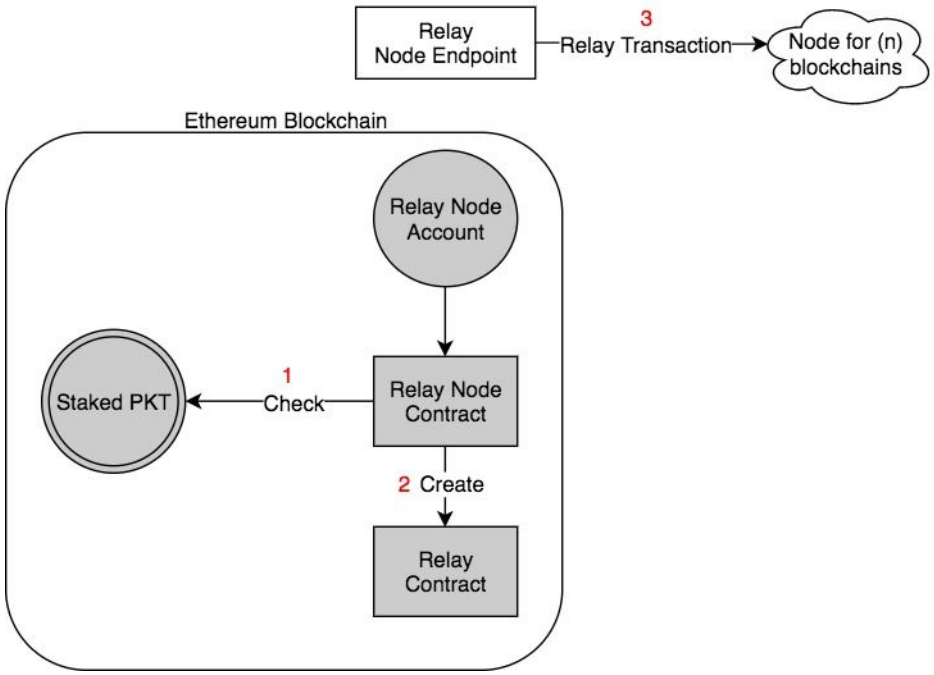

Figure 3: How Relay Contracts get created:

1. When the Relay Node Endpoint Receives a transaction, it will first check the staked PKT of the developer to ensure that the limit is not being reached
2. Create the Relay Contract
3. Once the relay contract is created the transaction will be relayed to given blockchain

There will always be a static base amount of transactions that the minimal amount of staked PKT a Relay Contract (Section 2.3) will allow. The amount of transactions that a given amount of staked PKT will throttle depends on the following:

- Composite log price of PKT at exchanges over the last month
- Amount of PKT in circulation
- Rate of Relay contracts created

2.1.3 Oracles

Oracles are the entities that check the outside world for some sort of verifiable state [11, 12] and act as a source of truth for the external transactions in the Pocket Network. Oracles will receive mint for any transaction they assist in relaying.

When a Relay Contract gets created, it will query the Oracle directory for Oracles available to verify the transaction. The Relay contract will have the relevant information that an Oracle needs to verify. Once enlisted, the Oracles will query the external blockchain until the transaction has cleared. Once cleared on the separate blockchain each Oracle will confirms, and the Relay Contract is destroyed.

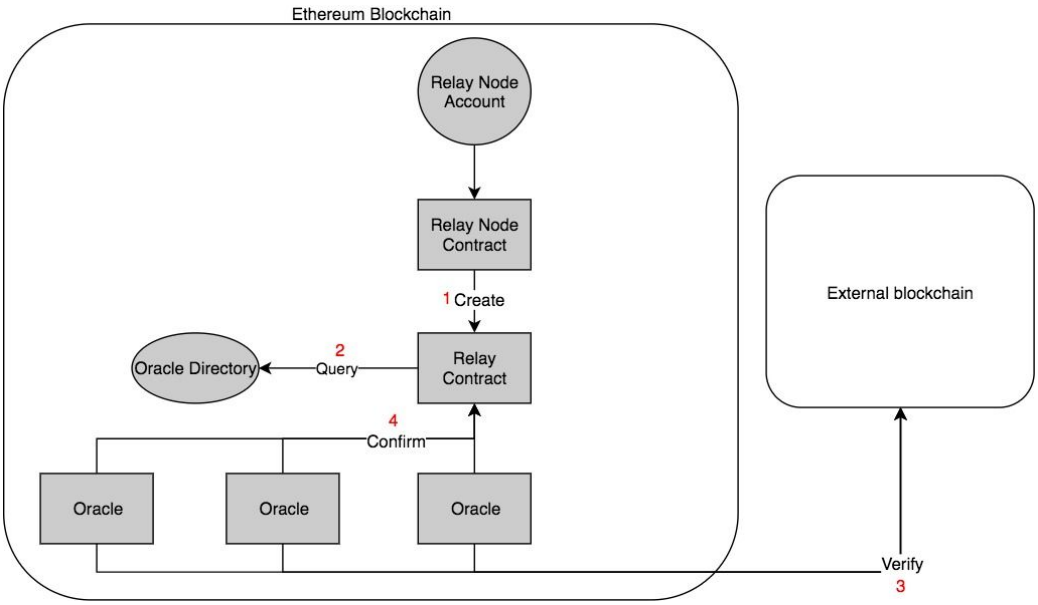

Figure 4: How Oracles are chosen and confirm external transactions:

1. Relay Contract created
2. Relay Contract Queries Oracle directory for available Oracles
3. Oracles verify in external blockchain whether transaction cleared
4. Oracles confirm in the Relay Contract that transaction cleared

Oracles must also stake PKT to verify transactions outside of the network. Limits to the amount of PKT minted will be throttled based on how much is staked.

There is one exception that requires no Oracles - if a transaction is on the Ethereum blockchain there is no need for an Oracle. In this case, the Relay Contract will have a timer based on block time with the relevant information to check if a transaction cleared.

2.1.4 Minting

Relay Nodes and Oracles will mint new PKT as a result of relayed transactions being verified.

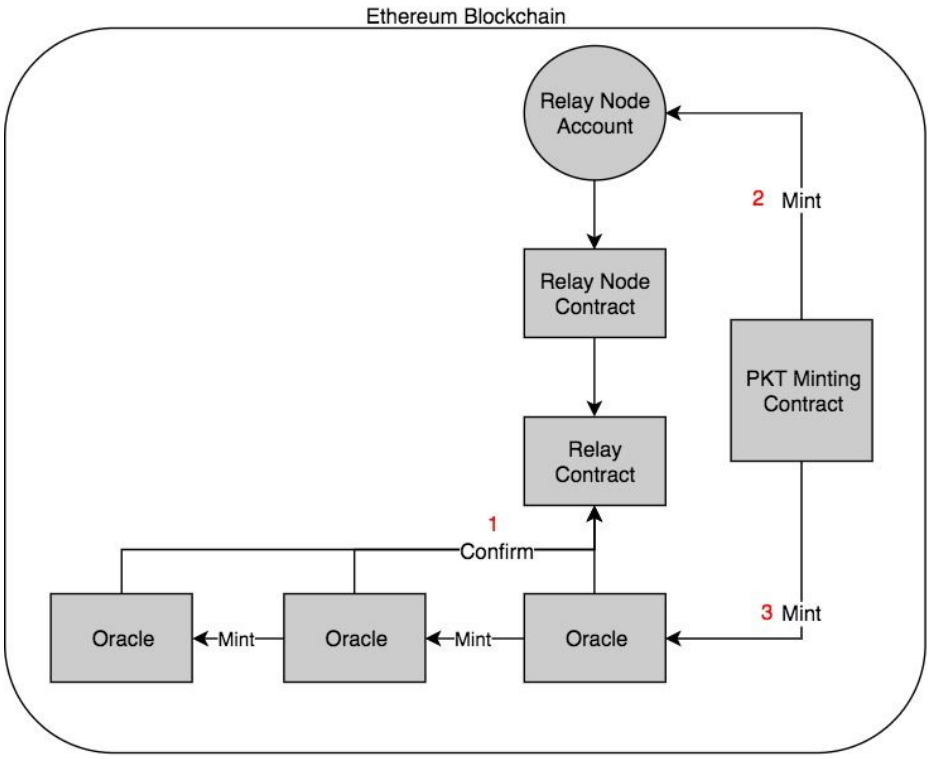

Figure 5:

1. Oracles confirm the external transaction
2. Pocket Minting Contract mints to Relay Node Account
3. Pocket Minting Contract mints to Oracles involved in transaction

PKT will continue to be minted throughout the life of the network. Similar to bitcoin, the number of PKT minted will decrease over time as the network matures. Inflation rate will be determined at a later date.

2.2 Relay Nodes

A Relay Node within the Pocket Network is any account that agrees to relay transactions and mint PKT for successful relays. Relay creation is throttled based on the amount of PKT the Relay Node has staked. The node can relay any information that is publicly verifiable. The Node lists their endpoint in the API Directory Contract (Section 3.1) so a developer can use their services. Relay Nodes create the Relay Contracts that determine when PKT will be minted.

2.3 Developers

Developers use the services of Relay Nodes to facilitate transactions by finding a list of endpoints using the Directory Contract (Section 3) through their client applications. To interact with the Pocket Network, the developer client application must create and sign the peer to peer transaction sent between clients.

A developer could be using any token (Bitcoin, Ethereum, Zcash, custom token, etc) for the client transaction, but must own PKT and stake to facilitate the transaction for their chosen token. The client app also creates and signs a transaction in the given token.

In production, there will be a series of state channels (Section 5.1) that developers and Relay Nodes open up with each other. This will enable drastically lower congestion to the Ethereum network.

2.4 Relay Contracts

The Relay contract acts as the the throttle determining whether a transaction should be relayed. When (x) of (n) oracles designate a transaction has been successfully relayed, new PKT gets minted for the Node. Relay contracts will have timeout and self-destruct methods to ensure that no mint is in limbo.

2.5 Transaction Example

Figure 6 represents a high level sequence of how a transaction gets relayed within the network. A few steps are omitted for simplicity.

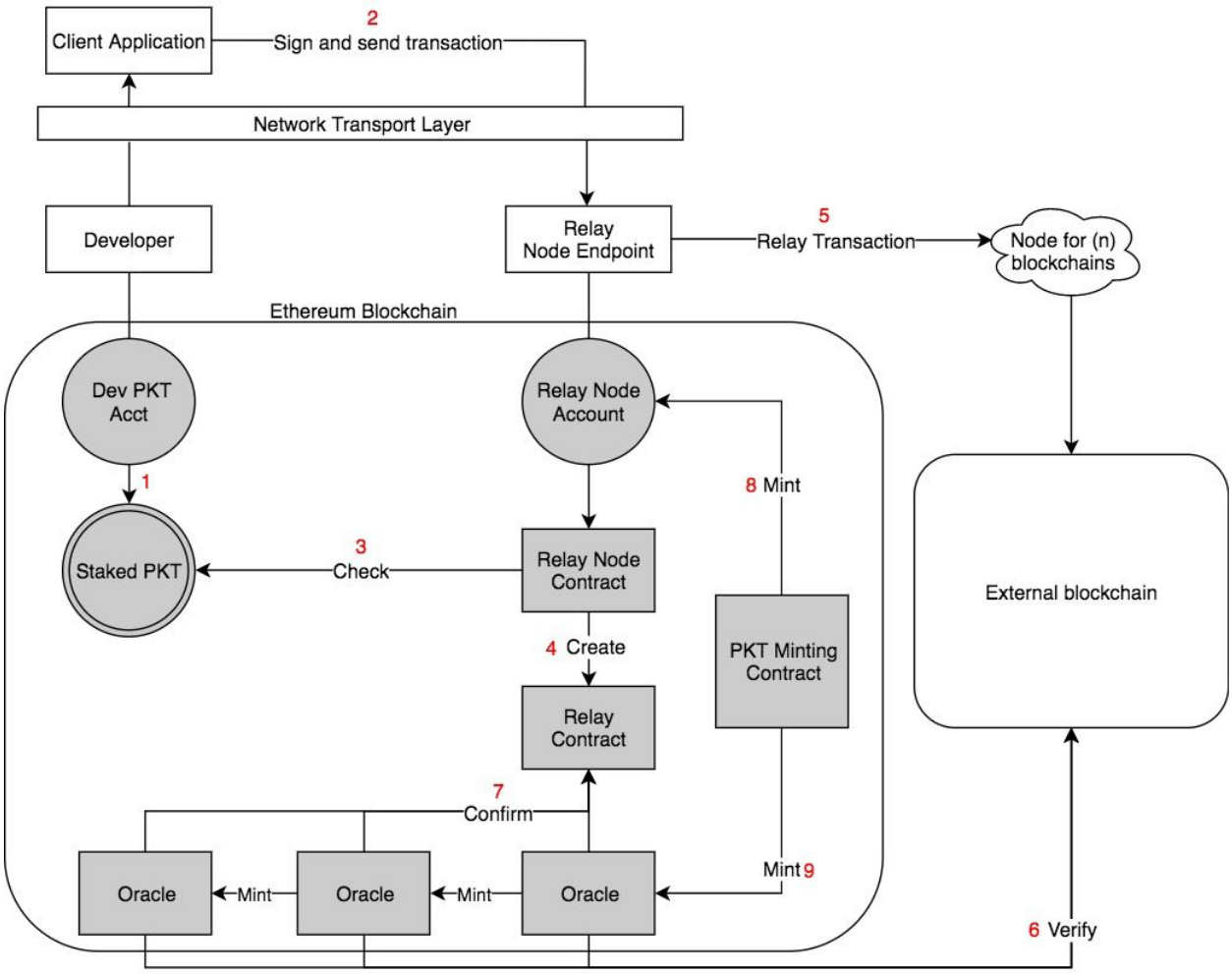

Figure 6 - Grayed out circles are user accounts. Grayed out boxes are contract accounts. An explanation of the flow goes as follows:

1) Developer stakes PKT to allow transactions to occur
2) Client application signs and sends a transaction to Relay Node Endpoint
3) Relay Node checks staked PKT to ensure a transaction is possible
4) Relay Node contract creates Relay contract
5) Relay Node sends the transaction to external blockchain
6) Oracles check publicly verifiable blockchain until X amount of confirmations have passed
7) When transaction is confirmed, confirm with Relay contract
8) Relay Node receives mint

9) Oracles receive mint

This functionality opens up a wealth of possibilities, particularly for mobile developers. They do not need any web3 specific browser to access or harness the power of DApps. Developers can:

- Use any token that is already on the exchanges available to them.
- Do not have to worry about running a light node for any blockchain asset client side.
- DApp just needs to sign transactions and act as a wallet for the end user.

Pocket Network allows a mobile developer to focus on their product and business - whether it's a game that rewards tokens or a peer to peer ridesharing marketplace. Pocket Network facilitates the transfer of tokens for the developer's customers. There is no centralized entity that acts as a honeypot for malicious actors. All security is delegated to the client. Pocket Network allows developers to build applications that put blockchain based assets in the hands of people who need it - without the overhead of centralized security or learning a new programming language.

3 Directory Contracts

Pocket Network will have two directory contracts deployed in the network to allow for easy upgradability. This enables continuous integration of code.

3.1 API Directory Contract

Using a decentralized network of REST API endpoints come with tradeoffs. One tradeoff is guaranteeing that any given endpoint has as close to 100% uptime as possible. Where centralized services have the advantage of spinning up multiple servers on services like AWS to ensure redundancy, users of the Pocket Network do not have that luxury.

To get around this problem, Pocket Network will be using an API Directory Contract. The official address will be on the Pocket Network website. Any Node can publicly record their endpoints with important pieces of data such as region and fee included. Any developer using the Pocket Network will ping the directory.pkt.network URL and will receive a list of filtered endpoints read directly from the Directory Contract.

This ensures a dynamic list of endpoints that will always be available and ensure as close to 100% uptime as possible. The registry will be on the blockchain, and will be verifiable by anyone in the community. We understand this involves a small amount of trust in the Pocket Network team for hosting the directory, but by having the Directory Contract publicly auditable on the blockchain, the community can ensure there is no foul play or favoritism on our part.

3.2 Oracle Directory Contract

Similar to the API Directory Contract, Relay Nodes must query the Oracle Directory to enlist Oracles for a given transaction. It can be implied that if an Oracle has more at stake they are more trustworthy within the network.

4 Node Factory Contract

Users who decide to be Relay Nodes will not be expected to create and deploy their own Relay Node Contracts. To ensure consistency throughout the entire network, there will be a Node Factory Contract that any account that has registered on the API Directory Contract can access.

There can only be one Relay Node Contract per address. Accounts requesting for a new Node contract will be cross-checked with the Directory contract. If a Relay Node decides to stop relaying transactions, their Relay Node Contract will be destroyed and will then need to request a new one. Relay Node contracts can create an unlimited amount of Relay Contracts. When each Relay contract gets created, the current state between the Relay Node account and Developer will be locked and a state channel will be opened with each developer utilizing the Relay Node.

The Node Factory will act as an important data source for the Pocket Network. Publicly accessible data points will include the number of current live Relay Nodes, distribution of Relay Node types running different blockchains and average fee amounts.

5 Protocol Token

PKT connects developers, Relay Nodes and Oracles in a new, dynamic marketplace. Pocket Network is a cryptoeconomic protocol [13] meant to align financial incentives for rational economic actors within the network to relay transactions. The primary use of the PKT will be for developers, Relay Nodes and Oracles to stake PKT to allow the minting of new PKT. Governance (Section 5.2) and voting power will be decentralized.

5.1 State Channels and scalability

At scale, thousands of transactions a second will be sent using the Pocket Network. It is unrealistic for Relay Contracts to be created for each individual transaction. Gas costs would be unsustainable and the Ethereum network would inundated with transactions. State Channels allow developers and Relay Nodes to take every single Relay Contract creation off chain [14, 15]. The relationship between developers and a Relay Nodes is ideal for a state channel implementation due to the potential for many thousands of state updates in a short time frame.

Opening up a state channel connection connecting developers to Relay Nodes allows for several advantages:

- No gas cost per transaction
- Unlimited scalability
- Hourly uptime fees as opposed to per transaction
- Privacy for developers and their businesses

Our initial testnet release will not have a state channel implementation. Part of the development plan is to have state channels fully functional before Pocket Network is released to production.

5.2 Governance

Pocket Network will function as a liquid democracy and have decentralized governance. At first, protocol upgrades will be voted on using a simple multisig wallet. Actors within the network with staked PKT will have increased voting power with diminishing returns as staked amount grows. The Pocket Network team will look into integrating into networks such as Aragon.

5.3 Burning Tokens

As the network grows, more PKT will be staked by larger actors within the network. Staking contracts will have an automatic burn method should there be dishonest actors within the network. When there is more at stake, there is more to lose. This is meant to keep all actors honest within the network.

6 References

1. Michael O'Rourke. Onramps into crypto. http://www.throwingdarts.co/blog-1/2017/6/4/onramps-into-crypto. Accessed 2017-06-04.
2. Jeff Meyerson. Software Engineering Daily: Coinbase Security with Philip Martin. https://overcast.fm/+E6UDlDV4o. Accessed 2017-07-14.
3. Cody Brown. How to lose $8k worth of bitcoin in 15 minutes with Verizon and Coinbase.com. https://medium.com/@CodyBrown/how-to-lose-8k-worth-of-bitcoin-in-15-minutes-with-verizon-and-coinbase-com-ba75fb8d0bac. Accessed 2017-05-31.
4. A Timeline: ShapeShift Hacking Incident. https://info.shapeshift.io/blog/2016/04/19/timeline-shapeshift-hacking-incident. Accessed 2017-06-14.
5. Jaxx. https://jaxx.io/. Accessed 2017-06-14.
6. Blockchain.info. https://blockchain.info/. Accessed 2017-06-14.
7. Infura. https://infura.io/. Accessed 2017-06-14.
8. Status.im. http://status.im/. Accessed 2017-06-14.
9. Toshi. http://www.toshi.org/. Accessed 2017-06-14.
10. Vitalik Buterin. Secret Sharing and Erasure Coding: A Guide For the Aspiring Dropbox Decentralizer. https://blog.ethereum.org/2014/08/16/secret-sharing-erasure-coding-guide-aspiring-dropbox-decentralizer/. Accessed 2017-07-08.
11. Vitalik Buterin. Ethereum and Oracles. https://blog.ethereum.org/2014/07/22/ethereum-and-oracles/. Accessed 2017-06-14.
12. Thomas Bertani. Understanding Oracles. https://blog.oraclize.it/understanding-oracles-99055c9c9f7b. Accessed 2017-06-14.
13. Matthew Goldenberg. Vlad Zamfir-Creating a Cryptoeconomic Protocol from Scratch. https://blog.verity.site/vlad-zamfir-creating-a-cryptoeconomic-protocol-from-scratch-1404b61764bf. Accessed 2017-07-08.
14. Jeff Coleman. State Channels. http://www.jeffcoleman.ca/state-channels/. Accessed 2017-06-14.
15. Ameen Soleimani. State Channels - Ethereum is open for business. https://www.youtube.com/watch?v=MEL50CVOcH4. Accessed 2017-06-14.