
Otto's Guide to Pocket



Tome I : Claim and Proof lifecycle

First things first: Why are we here ?

So, you are tasked with understanding the utility of Pocket Network?
Maybe you are a poor soul trying to understand the mysteries of life, the universe and everything?
Or maybe someone guided you here to look for answers about the “black magic” that the Pocket utility layer seems to made be off?

Well, maybe you are just curious or bored or both? xD

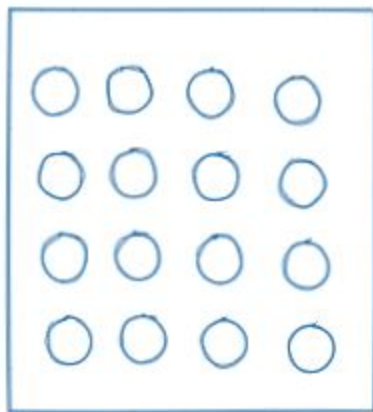
Anyways, do not worry child, I will try to explain this arcane art on simple terms.

Let's give it a quick glance from the top of the mountain.



1

Nodes staked for ethereum



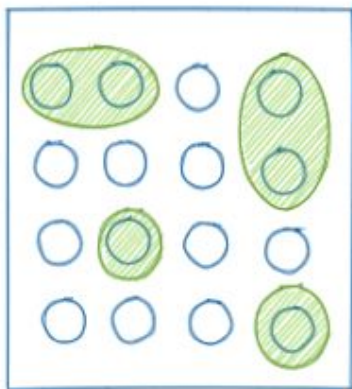
Application staked for ethereum



staked amount 6000

2

Nodes staked for ethereum



Selected nodes to serve in a session at height 5
6 nodes per session

Application staked for ethereum

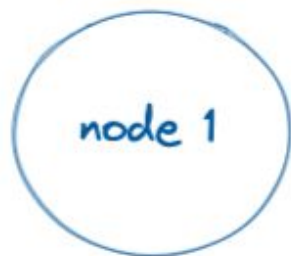


staked amount 6000

3

Node staked for ethereum

Application staked for ethereum

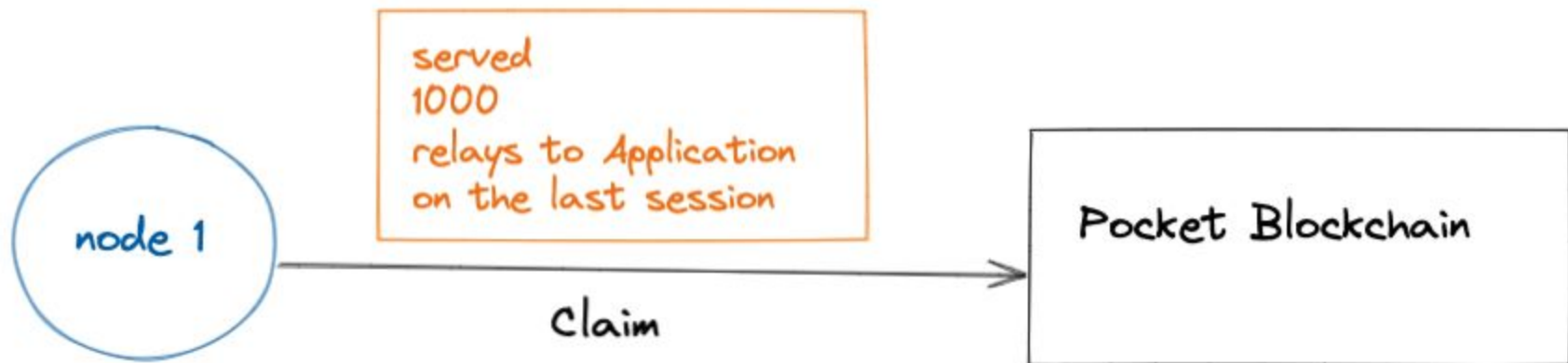


Max relays = 1000

staked amount 6000

4

Sessions end after 4 blocks and a new session starts



5

we enter a security Waiting period of 3 sessions ZzzZZzzz



The proof "asked" by the network of the work i "claim" to have done for the Application

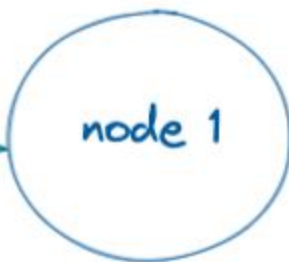
Proof



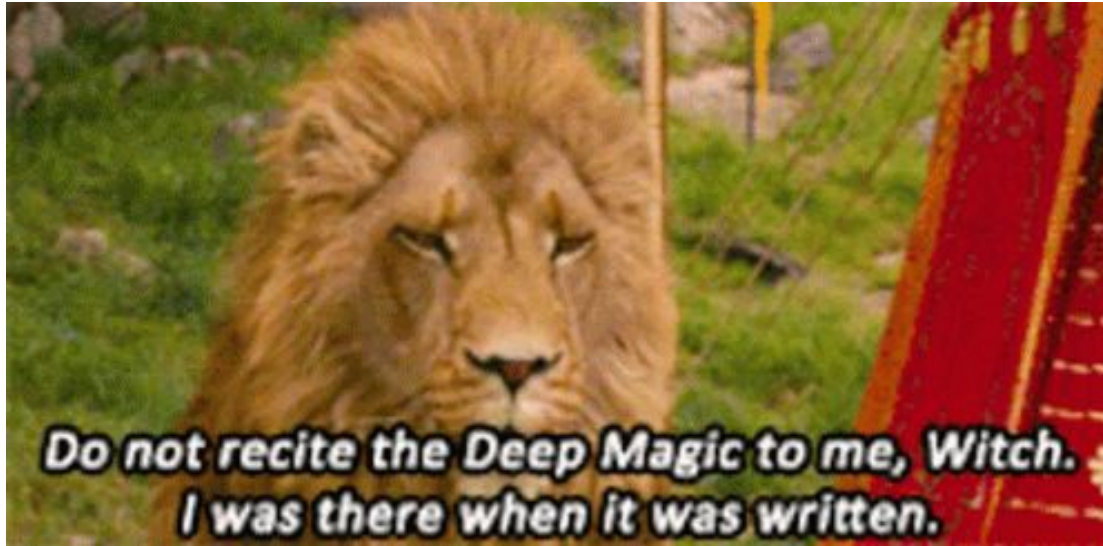


6

Rewards!



The OG's After my "explanation"



Warning!

If you understand everything up until this point, congratulations! You know kung-fu!

The next part might be boring for you as we are gonna get a closer look at the cycle and I might butcher some technicalities for the sake of a simpler explanation.

If you are in for the ride, remember to fasten your seatbelts and bring popcorn as I try my best to simplify the complexities of a system while trying to not look bad in front of the engineers that made this xD

Also please stop me at any time if you have any questions.



So Claim and Proofs, right?

Yes, if you are here I suppose that you are familiar with Pocket Network, if not you may need to rethink your decisions in life... Anyways, to achieve its main goal of incentivizing the access to other blockchains, Pocket rewards **nodes** that allow **applications** to access their blockchain nodes throughout the Pocket Network protocol with its native token called **POKT**.

But before the work done by the nodes of the network is rewarded and new tokens are minted, nodes first need to demonstrate that the work they did in a given session was valid.

So let's see how that flow starts.



Start of the cycle: Sessions

First and foremost we have the session, you can think of it as group of **N** nodes that was pseudo randomly selected to serve an application **A** requests for the Blockchain **B** and the duration of **X** blocks .

There are some conditions that the protocol expect to be met for creating a session that can be rewarded:

- We should have **N** nodes staked and non jailed for blockchain **B** as a minimum to be able to create a session.
- Blockchain **B** should be part of the “**SupportedBlockchains**” on chain parameter

A is the application Address.

N is taken from “**SessionNodeCount**” on chain parameter,

B is the arbitrary ID set for a given chain.

X is the session duration and defined based on the “**BlocksPerSession**” on chain parameter.



No Jimmy, Its PSEUDO RANDOM!



So we got a session, then what?

After we got a valid session the nodes selected need to serve relay request from the application.

The protocol, based on the application **A** staked amount, sets a limit for the amount of relays that each node can service to the application **A** during the session. (**BaseRelaysPerPOKT** * staked amount) / **N**

After the **X** amount of blocks has passed the session ends and a new session is generated for the app with a new set of nodes.

TIP: The node needs to serve at least the “**MinimumNumberOfProofs**” to the application to be able to get a reward for that session.

Session ended, where is my reward? Entering the CLAIMsss

So, after the session ends your node, in the next X blocks based on their address, will submit to the network what we call a **Claim**, the claim sets on chain the amount of relays (work) that a node “claims” to have done for an application **A** on a session.

Without getting into the weeds, the claim also sets into stone the base (root) that we are going to use to submit the proof later to show the world we did the work we claim to have done.







Lets jump into the weeds,... a little.

As a node is servicing, the node is also storing information about the relays that is being served, when the session ends the node seals the evidence and stores it on a merkle tree, after this step we get the merkle root that will be used as the base of our claim.

The importance of this step can be summarized with this screenshot from wikipedia:

Uses [\[edit \]](#)

Hash trees can be used to verify any kind of data stored, handled and transferred in and between computers. They can help ensure that data [blocks](#) received from other peers in a [peer-to-peer network](#) are received undamaged and unaltered, and even to check that the other peers do not lie and send fake blocks.

Now we Proof, right?



Well, not that fast. First your claim needs to be mature and not expired, that means that you need to account for:

- X = the blocks per session defined by the “**BlocksPerSession**” parameter
- The “**ClaimSubmissionWindow**” parameter that defines the amount of sessions that you need to wait before submitting a proof for a claim
- The “**ClaimExpiration**” parameter that defines the max amount of blocks a claim is going to be on the state before expiring

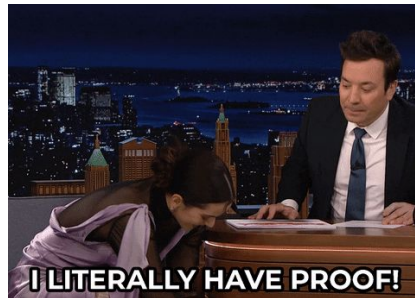
So yeah you have a limited time window to submit a proof.

Window : After the claim is mature (**current value is 3 sessions / 12 blocks**) and before claim expiration (**24 blocks / 6 session**)

Ok we are on the window, what happens now?

So the node will need to submit a proof for that claim that was maturing in the state. So on the next **X** blocks, the node, based on its address, and using a now revealed secret index, will submit a proof tx that contains information about one specific relay done during that session that will be confirmed by the network by using the properties of the merkle trees.

Simplifying a bit (and maybe butchering xD) how our hash trees work, the proof will only be valid if the information submitted (leaf) can be traced to the root that we submitted earlier when we set claim on the state.



And we are done!

If the proof is accepted by the network the node is rewarded tokens based on the “RelaysToTokensMultiplier” on chain parameter and also, after PIP-22, based on its staked amount and the “bucket” or tier it belongs.

- 15k – 29.999k 1X”RelaysToTokensMultiplier”
- 30k – 44.999k 2X”RelaysToTokensMultiplier”
- 45k – 59.999k 3X”RelaysToTokensMultiplier”
- 60k – to the infinity and beyond 4X”RelaysToTokensMultiplier”






Some takeaways and additional information

- Work done by a node is rewarded after this, now not so mysterious, time window.
- The node has the responsibility of keeping its pocket evidence db in good condition as problems like corruption will generate claims/proofs that cannot be redeemed. (Thank you for your free service xD)
- The claims and proof are submitted automatically by the node if running, if the node runner stops the node it may miss its windows to submit claims and proofs. If within the time window and at latest height node will retry submission.
- For reference about the parameters and their current values:
<https://docs.pokt.network/learn/protocol-parameters/>

Just for the record

And like someone said:



SBF  @SBF_FTX · Nov 10

21) NOT ADVICE, OF ANY KIND, IN ANY WAY

I WAS NOT VERY CAREFUL WITH MY WORDS HERE, AND DO NOT MEAN ANY OF THEM IN A TECHNICAL OR LEGAL SENSE; I MAY WELL HAVE NOT DESCRIBED THINGS RIGHT though I'm trying to be transparent. I'M NOT A GOOD DEV AND PROBABLY MISDESCRIBED SOMETHING.

Q&A Time

?????



Questions?!?