3DPass

The Ledger of Things

3dpass.org

INTRODUCTION

Since the beginning of blockchain technology, hundreds of coins appeared on the market providing issuing tokens and smart contracts in hope of wide spread implementation. The majority of tokens and coins currently issued are backed by digital assets which are not coupled with real goods and services such as fiat currencies, shares and offerings. This presents an application problem, in the real world the majority of transactions are related to physical objects or services that refer to them. As we dive deeper it becomes apparent that the digital transformation of real objects presents a key to get access to many p2p deals all over the globe. The key for the implementation is to transform real objects into digital assets in order for anyone at any given time point to provide the required evidence of the object's authenticity.

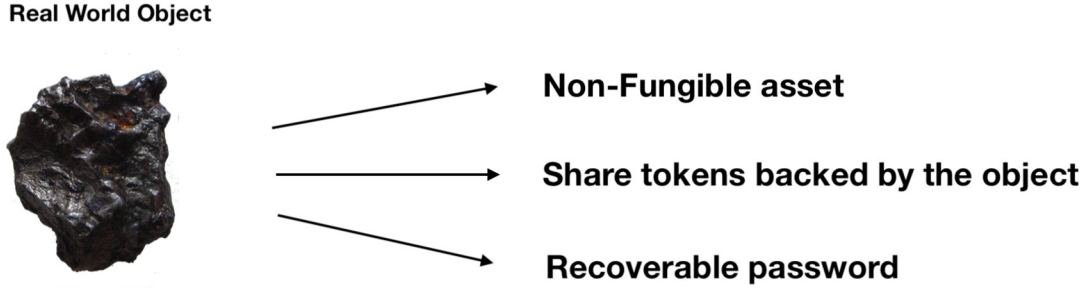
From the initial perspective, the problem statement looks very challenging to solve but there are some opportunities upon further inspection. This includes face recognition technology, many ML projects, NFT, etc. Why is such an emphasis placed shape recognition? It's evident that the shape is the critical property of any 3D object. Once the shape is identified you add additional properties such as weight, clarity, density, owner's biometric data, etc. This can be used to reliably identify the entire object by several aspects.

Let us consider if we could reduce the variety of objects to one of the most common which are solid objects that have structural rigidity. In this regard the transformation issue is simplified and provides a solution for deals to be made such as but not limited to precious stones, vehicles, appliances, real estate, art and jewelry items on the blockchain. Additionally as an added benefit, the method of generating and recovering of passwords based on real objects is unlocked. Solving the disconnect between physical and digital items via this approach enables the digitization of real objects as well as turning the blockchain "real".

ABOUT

3DPass is an open source p2p platform which enables the transformation of real or virtual objects into a trustless sustainable digital identity called HASH ID and to leverage it as a digital asset, such as:

1. 3DPRC-2** single Non-Fungible token;
2. 3DPRC-2** share tokens backed by the object properties;
3. Password, which is recoverable by means of getting the object scanned.

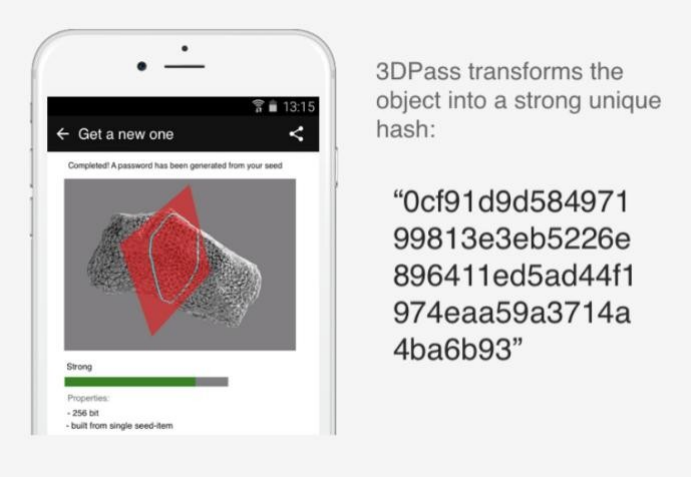


In the foundation of 3DPass lies real world object recognition technology including various research-based algorithms created by either team or community members. The algorithms have been especially adapted to work within the blockchain environment which does not provide any feedback to compare objects to one another while on the process of recognition. This is caused by the absence of a trusted backend, since the blockchain architecture implies that equal nodes that are owned by users who do not trust each other. So that any object, event or transaction must be verified by a math calculations performed off-line. The first algorithm called Grid2d is designed to recognize 3D objects shape. It was suggested by the 3DPass team member Michael Co in 2020 as a response to the real world objects digital transformation challenge.

The implementation of those algorithms is *pass3d recognition toolkit*** which plays a significant role in Proof of Scan p2p network consensus used in The Ledger of Things blockchain decentralized platform. This represents kind of a different concept of a Layer 1 network having assets tethered to the smart-contracts and dApps logic. You may refer to this as the network of unique things, which allows for the utilization within smart-contracts and deals. The network nodes will prevent the duplication of assets.

The first features are mostly provided by the recognition algorithm developed and published by the 3DPass team. Although, one of the most critical aspects of the project is to encourage the community to develop additional features and add them into the 3DPass platform. (see RECOGNITION ALGORITHMS)

HASH ID



Pass3d recognition toolkit captures distinctive properties of the object shape and produces a sustainable digital identity named HASH ID. This will remain stable irrespective of how many different 3D scans of the same object have been processed. It enables the establishment of a one-to-one correspondence between the object and its digital asset. Conventional NFTs do not provide this due to just relying on a signature of a file (e.x. If I changed the file with just one dot, the signature would have changed completely creating a copy of the asset easily and without any control). By means of leveraging objects recognition, this prevents the asset from being copied.

The recognition algorithm Grid2d is flexible enough to adjust the definition level of processing to the 3D scanning resolution or accuracy of 3D model. It allows to define the

asset property rights border and to distinguish whether or not the particular object is considered genuine or fake.

HASH ID provides excellent level of privacy containing "0 knowledge" of the object it was produced from. The object's seed data is protected by cryptographic standard SHA-2.
A key aspect for distinguishing 3D objects is the uniqueness of the shape as well as the scanning quality and app settings.

TRANSFORMATION

In order to get transformed, a real 3D object is scanned by a professional 3D scanner or smartphone camera and then processed by the recognition toolkit. However, this does not exclude digital objects created in a virtual environment. The key aspected in both cases is the shape which is the primary property.

In order to recognize, all that is required is to scan and process the object again. If the object has an original shape, the hashes are going to be matched (read more RECOGNITION ALGORITHMS).

3Dpass platform provides three tokenization options:

1. The first one is creating a single non-fungible token or a registry of assets so that 1 3D object = 1 NFT (not one file = 1 NFT);
2. The second one is minting share tokens backed by the objects. And the unit of that kind of currency would be a quantum (for example, 1 gram);
3. The third one is using real world objects as passwords, recoverable by means of scanning;
4. You can also create your own L2 chains of limited supply asset for gaming or Metaverse (see 3DPASS P2P NETWORK).

HASH ID MULTI-OBJECT OPTIONS

3DPass allows to create a Hash ID not just from one item but from several which could be either a few real objects or some properties of one like weight, clarity, density, owner’s biometric data, etc. Examples of such combinations could look as follows:

- Hash 1 = object shape + weight + clarity
- Hash 2 = 1st object shape + 2nd object shape + owners iris scan
- Hash 3 = object shape + weight + owner’s fingerprint

Property values represent an additional seed data the Hash ID would be created from. And this is nothing but multi-factor authentication** where 3D shape is the main factor but the other properties are the additional ones.

In the example above:

- The object shape and its weight are “something that you have” factors
- The owner’s biometric data is “something that you are” factor

It works similar to multi signature with several keys involved. The lack any of those means losing ability to recover the Hash ID entirely.

RECOGNITION ALGORITHMS

There are numerous approaches to processing the shape of 3D objects to address digital transformation challenges. It is probable that certain methods will outperform others for specific issues. The 3DPass team has developed and tested several processing algorithms, with the most stable one described below. As a community-driven project, members will continue to enhance these algorithms and devise new ones to tackle additional challenges and explore broader market opportunities.

Candidate-algorithms:

- 3D Objects
- 2D Objects (Drawings)
- 2D Fingerprints
- Face Recognition
- Voice
- Melodies
- Radio Signal
- Barcodes
- QR Codes

This list is not exclusive and can be extended to anything that is recognizable by means of machine processing.

GRID2D ALGORITHM

Algorithm name: Grid2d, 3D object recognition, Author: Michael Co

First of all, the hash calculated has to be reproducible i.e stable for different scans of the same object accounting for any noise from scanning. There is no feedback to compare and provide a single 100% working hash from each 3D scan automatically. So a sorted short list of hashes is required as a result to give users the opportunity of picking up the most stable one. The hash ID will be considered stable if the hash value present on the top of the list is consistent every time a new scan of the same object is processed. The algorithm logic is flexible enough to adjust the definition level of processing to the 3D scanning resolution and accuracy which might differentiate depending on the scanning device.

The simplest way to get some unique characteristics from surface of the object is to cut it into N slices and process each of them separately. The problem, therefore, now turns into

2D. By means of combining results from N slices, it becomes possible to calculate the final hash.

Prerequisites:

- The object must be simply connected (i.e one piece)
- The object must not have the regular form
- The colors and textures are ignored

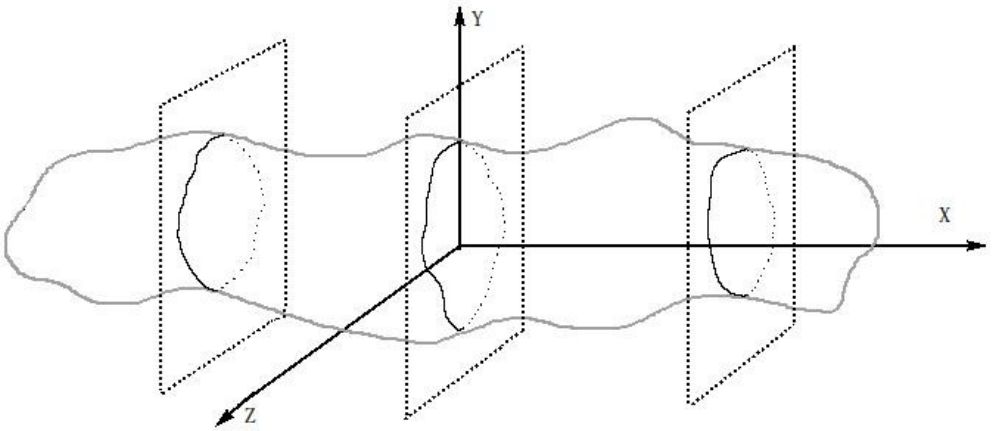
The object orientation has to meet the following conditions:

- The center of coordinates is at the center of mass of the object
- Cartesian coordinates coincide with main inertia vectors

In order to get it unambiguously the main inertia components have to be different, let's say, by 10% (*parameter #1 of algorithm*). Objects like sphere or cylinder should be rejected. So, the axis X coincides with the inertia vector corresponded to maximum inertia component, next axis Y and last axis Z corresponded to minimum inertia component.

2. Let's cut the object by N planes uniformly spaced alongside the OX axis. Then either crosses of planes and object's surface produce N contours.

Drawing 1: Cutting the object with 3 planes



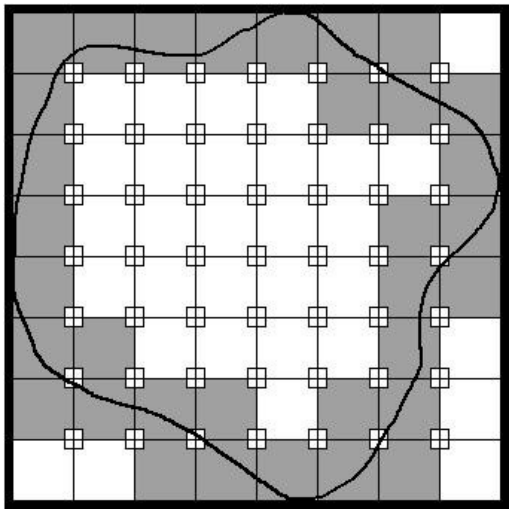
3. For each contour the following steps will be performed:

3.1. Scale the contour to fit a square having sides equal to maximum size of contour alongside X or Y coordinates.

3.2. Select the number of cells in the square MxM (*parameter #2 of the algorithm*). Let us assume M=8 (like a chess board)

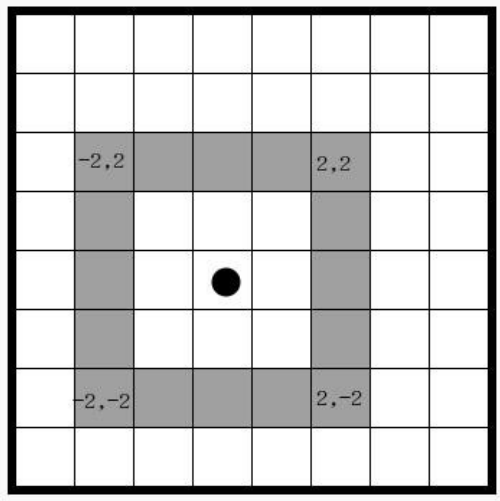
3.3. Find the set E of cells containing our contour. Due to noise neighboring cells the contour passes through are also included. Cells should be added to E if the contour is close to grid vertices less than 10% of cell size, by example (*parameter #3 of the algorithm*).

Drawing 2: Set of cells allowed



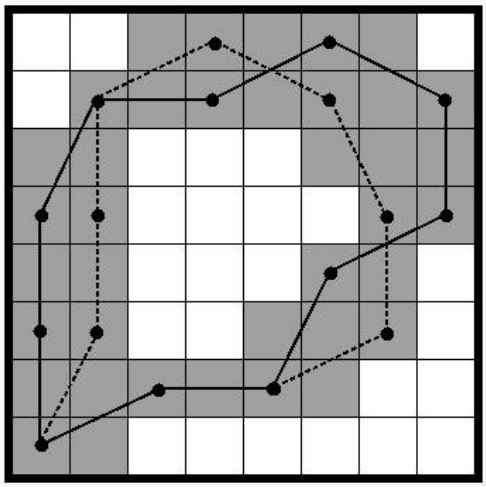
3.4. Generate all the possible polygons with vertices belonging to set E. Neighboring vertices have to be at a distance L, for example L=2 (*parameter #4 of the algorithm*). Vertices must not be close to each other at a distance less than L.

Drawing 3: Building a polygon



Down below are exposed two allowed polygons of the object represented at Drawing 1

3.5. Calculate root mean square deviation (RMSD)** from the polygon and the contour and do it for each polygon. In order to get this done we approximate the contour with splines and evaluate Q points uniformly spaced in approximating curve (*parameter #5 of the algorithm*). Also we need to evaluate Q points at the polygon. Spline approximation can be evaluated by *splprep*** function from Python *scipy* package.



The RMSD is:

$$RMSD=\frac{1}{Q}\sum_{i=0}^{Q}(P_i-R_i)^2$$

Where Pi=(xi, yi) — point at the contour, Ri — point at polygon.

3.6. Sort all the generated polygons by RMSD and take T best of them (*parameter #6 of the algorithm*).

4. Perform cartesian product of all the best polygons from each slice. The product of two polygons means concatenation of their points (coordinates of vertices).

For example, if:

$$P_1=((x_{11},y_{11}),(x_{12},y_{12}),...(x_{1k},y_{1k}))$$
$$P_2=((x_{21},y_{21}),(x_{22},y_{22}),...(x_{2k},y_{2k}))$$

We get:

$$P_1\circ P_2=(x_{11},y_{11},x_{12},y_{12},...x_{1k},y_{1k},x_{21},y_{21},x_{22},y_{22},...x_{2k},y_{1k})$$

So, we've got the input data for sha256. The last step is trivial.

The output of Grid2d algorithm is a list of hashes sorted by its affinity to object's surface.

3D OBJECT HASH ID CREATION

The input to our recognition toolkit called "*pass3d"* is a 3D model of the object (formats *stl* or *obj are required*). The output is a top10 hashes list inherent to the object shape.

```
USAGE:

    pass3d --algo <algo> --grid <grid> --infile <infile> --sect <sect>

OPTIONS:

    -a, --algo <algo>          3d hash algorithm Algorithm. Supported
algorithms: Grid2d
    -g, --grid <grid>        Number of cells in Grid2d algorithm

    -i, --infile <infile>     The path to the file to read

    -s, --sect <sect>        Number of sections in Grid2d algorithm
```

Success:

The object shape is considered to be recognized if there is at least one hash-value matched among two different processing results. In order to produce a HASH ID, two or more different 3D scans have to be processed of the same object and to be compared to the top10 results. The same parameters should be set up every time. It's recommended to use the same equipment, as well.

For example:

The first 3D scan processing…

```
~/Desktop/3dpass$ ./pass3d -i pir1.obj -a grid2d -g 8 -s 68

Select top 10 hashes

"9bccac20a0586638cc74a2ff295c987d470794f24f008b02ce02643d0281f03f"

"11c41b6b30b191a2d61ae803d48cc42e83f9fdaac730665b24e3272672133efd"

"6f37f712139012d1c118cadea3a44b0535fa6b4b1272b1da49af3eb6498011f6"

"4453ed1aa4dabe394a0cedd79f8edb0940fb43a5558fbfa89ce56dad3fc8876c"

"aa4019c8c160da9d2af69edc19589aabd925bc696966b967f92b71947f75f8f0"

"090ae6b23e2192fa4c2fb40cddad6e8537e2b437c49ff9fb227cf32c4e4085fc"

"dd227121b91adcb5beabb0be9412613ebdfde8c5660301eb17583fa644b8793d"

"880cfda2b4811bf2ff1fe3ab92b38e64fc134d98c3dc8764eb8641a477b77a47"

"15cc9ef656a14c9ffde999512d11bd81cd5eaedaa81139a61847d470ea01043b"

"543e1c3929ea810f4e8c7cfc27f0b60df21a9374089f2278617dae327e32b034"
```

The second 3D-scan processing…

```
~/Desktop/3dpass$ ./pass3d -i pir2.obj -a grid2d -g 8 -s 68
Select top 10 hashes
"72592f8f6ea67c60ca7d9c7683256c3636a30be464952eb82996bff52ca4415d"
"3720e731b9aa04b08d83de34a796cbc389fce2c62365c68206c5610106db053d"
"a65008cdc77f72b47eda70e7c2eb57f93e4fffde5a5356549ac7dbf5d422dffa"
"5930d4a4a98ddff21997daaa8410b151f85dcdb7bfe6b0fb1a05af0e99c276fc"
"6846a36abb6dc50df6845627e6553ede8337e7350254ae8d02b7b7a696c79192"
"b20cf89afb10f14795afe517c82d7f6185da840e6035c48b488792e2df61846d"
"aa4019c8c160da9d2af69edc19589aabd925bc696966b967f92b71947f75f8f0"
"deb83d22570bfc07b8881618dc34a6624616521475bac17798b7348cf6684fd1"
"dd227121b91adcb5beabb0be9412613ebdfde8c5660301eb17583fa644b8793d"
"543e1c3929ea810f4e8c7cfc27f0b60df21a9374089f2278617dae327e32b034"
```

In those two processing results 3 of the top10 hash-values matched. Therefore, the object is recognized.

If 3 or 4 different 3D scans of the object were processed, the most stable one Hash ID existing among the top10 of any 3D scan of this object could be picked. The more scans that are processed the more likely the best stable Hash ID will be picked. It could take 3 or 4 scans processed to choose, nonetheless in some cases choices could be limited due to only having one hash matched.

Failure:

The object shape would not considered to be recognized if there was no one match in the top10 results.

Parameters adjustment:

```
-g, --grid <grid>        Number of cells in Grid2d(#2parameter)
-s, --sect <sect>        Number of cross-sections in Grid2d
```

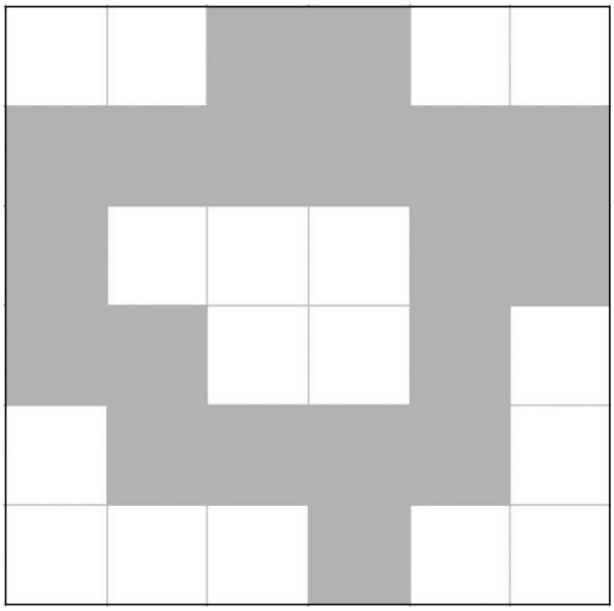
These are two key parameters that need to adjust in order to create the best possible Hash ID depending on 3D scans quality.

Grid scale parameter -g:

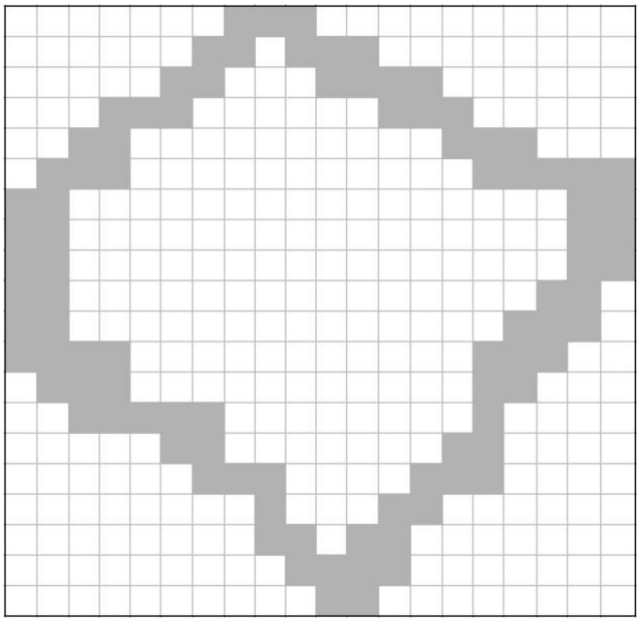
Grid size (-g) is the parameter which is used to adjust the recognition algorithm to the particular 3D scan quality. The higher the scan quality, the higher the number of cells that can be set up for the processing. According to the Grid2d algorithm, by means of increasing number of cells this leads to following a 3D scan cross-section contour more closely to the actual curve. This signifies an increase in the precision for recognizing the object shape. Additionally this reduces the possibilities of errors in the future. It's all about the balance between accuracy of the shape recognition and the ability to get the stable Hash ID.

Low definition scanners, especially smartphone apps, increase the likelihood of errors between two random scans taken from the same object. High definition and professional scanners might have a variability of 3 µm. So, it is recommended to perform several 3D scans made by the same equipment and to set the number of cells as high as feasible, provided a successful recognition result is achieved. This will be the best set up and might require some attempts to adjust the optimal (*-g*) parameter's value according to the scan quality.

Parameter *-g=6 (6x6 grid)* example:



Parameter *-g=20 (20x20 grid)* the same object example:

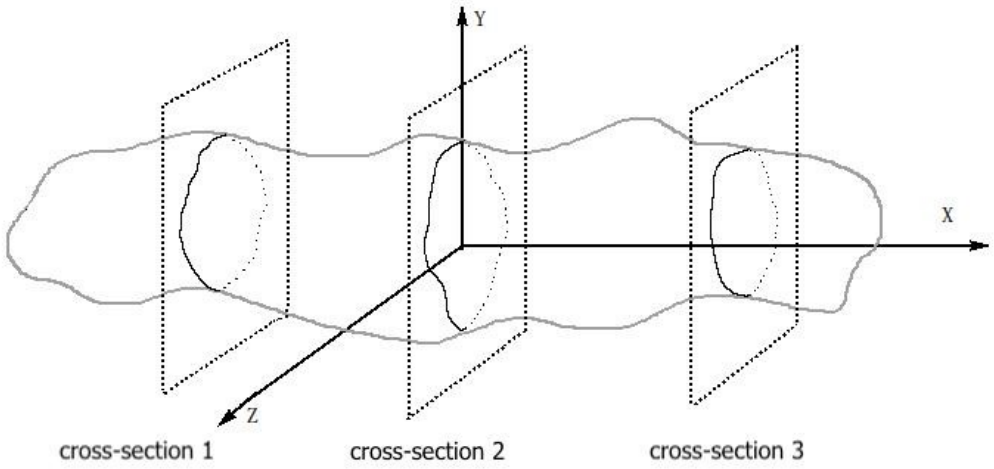


Considerations should be made to set the definition parameter *(-g)* up to the lowest quality of 3D scans that is expect to apply in the future. If the *(-g)* value is set up to be appropriate for HD scanners (*-g=20* or higher) but the scanner becomes unavailable, then the recognition success will not be met with a subsequent low quality scan which is recommended as *-g=6.*

Number of cross-sections parameter:

The more cross-sections that are set, the higher the hash strength that is achieved. Each cross-section represents a unique contour, this corresponds to the unique seed data the future hash would be created from. The capturing of more unique distinctions from the object shape, will provide a greater hash strength. As an example, if the set up of just one cross-section (*-s=1*), only one contour can be leverage of the object which represents a small data set. This will not be enough to describe the entire object shape. This would be equivalent to describing the shape of a whole apple from just one slice. Therefore in order to achieve the recognition of the entire shape instead of using just a few slices it is recommended to set up at least 100 cross-sections (*-s=100*).

Parameter *-s=3* example:



General recommendations:

1. The same set of parameters should be used for the same object while processing. Otherwise, this could lead to not succeeding with the recognition;
2. It is recommended to set up the grid parameter *(-g)* value according to the lowest scan definition that is expected to be used in the future. Such values as *-g=6* or *-g=7 (6x6 and 7x7 grid)* would be recommended for smartphones and tablets;
3. The number of cross-sections should be at least 100 (*-s=100*) in terms of leveraging the entire object shape instead of just a few slices.

The rest of parameters should be determined by means of experimentation and tests.

RESEARCH AND DEVELOPMENT

Since the conception of the 3DPass project in 2019, the development of several prototypes of different recognition algorithm proposal have been investigated. As a result “Grid2d” algorithm was determined to be stable and flexible enough to accommodate the different scan qualities. Testing with a range of 3D scanning devices from smartphone apps to professional HD scanners has revealed a set of parameters recommended to reach the maximum efficiency at recognition. These have been set up as default and continue to be optimized through continuous experimentation.

3D OBJECT’S DATA PRIVACY

3DPass never collects or transmits any 3D object data processed by the recognition toolkit unless the user decides to store it on the blockchain in open. The recognition algorithm uses RAM only, running on user’s local device without any Internet connection involved. It is assumed that users are responsible for the security of their own devices that are running 3Dpass. For example, if anyone had a 3D scan of any private physical asset, saved on their device, which is confidential such as the seed of the password or a rare diamond. The user is responsible for preventing the device from becoming compromised by a third party. This could lead to the third party gaining access to the

confidential 3D scan file, so that the secret information is disclosed. 3DPass will not be held liable and the principal of self custody is required from users.

The user should also consider the capacity and purpose for which 3DPass is used. In case of public objects such as real estates, vehicles, etc., the assumption should be taken that these shapes are disclosed and transparent. Nevertheless the user is able to scan any item before an exchange for authentication or eventually through marketplaces.

HASH ID STRENGTH

Since Hash ID is based on 3D object shape, its strength depends on how predictable the object shape is. If an object has a simple shape such as a ball, cube or some predictable and well known shape, then all of those seed-objects will provide weak strength Hash IDs. In most cases 3DPass is able to recognize them and reject them before the generating process is even started.

Should the object be a piece of natural rock or randomly deformed and hardened clay which on its surface has different elevations, depths, elongation, then this will provide a strong Hash ID.

If the object has a predictable shape it is recommended to use MULTI-OBJECT OPTIONS as 2 factor authentication to get more uniqueness to it. For example, the ball has the most predictable shape of all the shapes in the world. Should this be coupled with biometric data, this combination of “ball + biometric data” would provide a strong and non predictable HASH ID. This could also apply by leveraging different properties of the object like weight, clarity, density, etc.

3D object shape examples:

Good strength

Weak strength

Excellent Strength

As mentioned previously the Hash ID strength is of course depended on the 3D scanner resolution leveraged for taking scans from the object. If 3D scanning is performed by a High Definition scanner, the 3D model will closely follow the shape of the object surface to a higher precision. This quality allows 3DPass to recognize many minute details that might differentiate the object from others. Thus the higher the quality of scan that is performed, the higher the potential Hash ID strength will be.

ADDITIONAL PROPERTIES OF 3D OBJECTS

Besides the 3D shape, any real world object has additional properties that might help to reliably authenticate it by several sights. The most critical are measurable ones, such as: weight, density, clarity, hardness, coloring, owner's biometric data, etc. All of those properties can be revealed by lab measurements and scanning. Lab measurement equipment might represent smart devices (IoT) connected to 3DPass network directly.

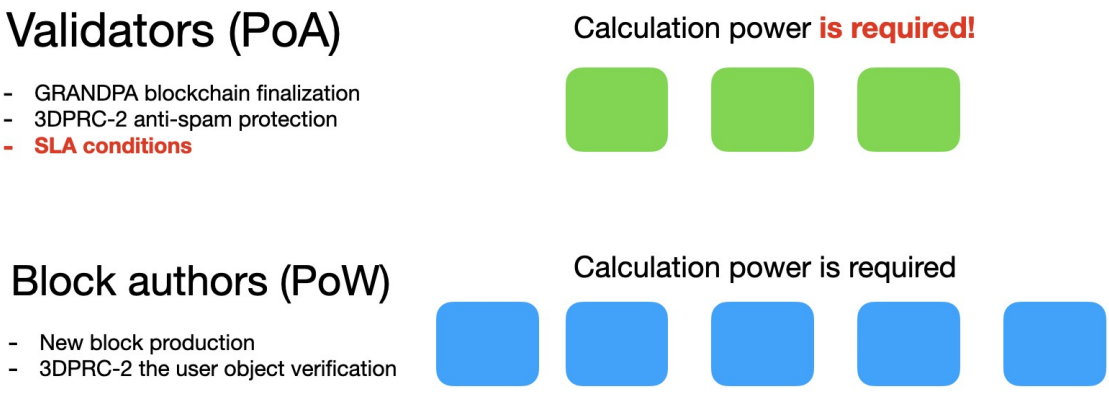
THE LEDGER OF THINGS P2P NETWORK

3DPass decentralized network is to provide the next level of quality to the entire blockchain market, delivering its instruments and tools for the tokenization of objects. The Nodes are equipped with recognition toolkit and will prevent the duplication of assets, even if its file was slightly changed (ex. with one dot, pixel or one byte). At minimum, it allows to establish and track 1:1 correspondence between the object and its digital asset. At max we will see endless of useful smart contracts and dApps operating within the eco-system and exploiting the HASH ID feature as an idea of authentication of real world objects (or digital objects) all across the Internet.

It is essential to not only have the opportunity of issuing non-fungible tokens but to also reliably be sure that the digital assets are corresponding with the physical asset. Irrespective of whether it pertains to the physical object or the digital 3D model due to each one having a set of distinctive properties that can be recognized. In order to solve this issue, computing power is required within the network for 3D object processing as well as spam protection.

Components:

- "Proof of Scan" consensus protocol is a hybrid *PoW (Proof of Work)* + *PoA (Proof of Authority)*, where PoW aspect is based on recognition technology and ASIC-resistant, as well. PoA part is ensured by GRANDPA deterministic blockchain finality protocol leveraged by the most reliable nodes on the network called Validators operating under SLA conditions (selection mechanism is similar to Proof of Stake).
- 3DPRC-2 tokenization standard is part of the network consensus, which is responsible for handling the user objects and its transformation into share-tokens (either, fungible or non-fungible).
- IPFS** is used as a decentralized storage
- Integration is provided by the Substrate framework
- Forkless upgrade feature, Smart-contracts, embedded DEX (Decentralized Exchange) and more than 30 useful modules operating.

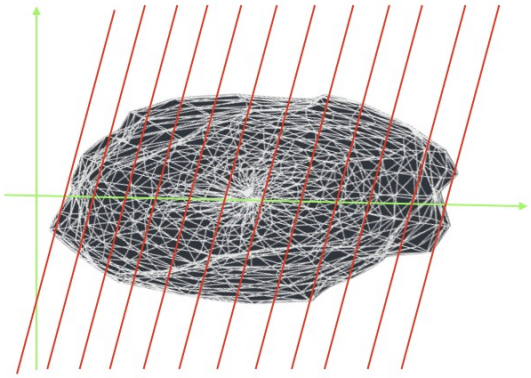


THE OBJECT VERIFICATION MECHANISM

3DPass network represents two layers of equal nodes leveraging Grid2d recognition algorithm, mentioned above. The very first purpose of the network is to check up the authenticity and uniqueness of the objects submitted by users.

For example (*Figure 1*), a user has just submitted a 3D object, hoping to construct a new block. The following aspects would be required to be added into the queue, such as:

1. 3D model of the object (.stl or .obj standards are supported**)
2. Hash ID of the object produced by Grid2d recognition algorithm
3. An additional security token (*rotation_bytes*) either being accepted by Grid2d. The token represents *8 bytes,* by which an angle of rotation *(μ = x°)* of the object is defined before getting crossed.



Now, every Node in the network being synchronized, can import the object and run the 3D shape recognition process in order to accept or decline the object. The nodes check up the following aspects:

1. Whether or not the Hash ID is corresponding to the shape of the object submitted. Which means the 3D object authenticity check. If the Hash IDs do not match to one another, then the object is considered fake and will be rejected;
2. Whether or not the same-shaped object is already existing on the network. If it is this implies that there is a duplicate and this will lead to rejection;
3. Whether or not the security tokens matched (*rotation_bytes*). If matched, that means the actual recognition algorithm was properly leveraged by the user who submitted the object. It provides assurance that the Hash ID was not copied and pasted from somewhere or just accidentally generated.

The object, which passed the verification check will be accepted (see more 3DPRC-2 tokenization standard** for the user objects and NEW BLOCK for the objects involved in mining).

NEW BLOCK

New block is going to be built on top of the Best block currently chosen for the Best chain (the longest chain rule is operating). Thus, Best block becomes the parent block while construction. The block is sealed to its parent block hash with SHA256** cryptographic hash-function, resulting as *double_hash*, which is a subject for difficulty to affect in order maintain the network target block time.

Block header consists of the following data:

- *Pre-hash* – main block identifier, which represents SHA256 hash sealed to the parent block hash with the cryptographic hash function.

```
"090ae6b23e2192fa4c2fb40cddad6e8537e2b437c49ff9fb227cf32c4e4085f"
```

- *Nonce* – a 3D model of the object in the content of *obj*** file format *(100 kb limit)*;

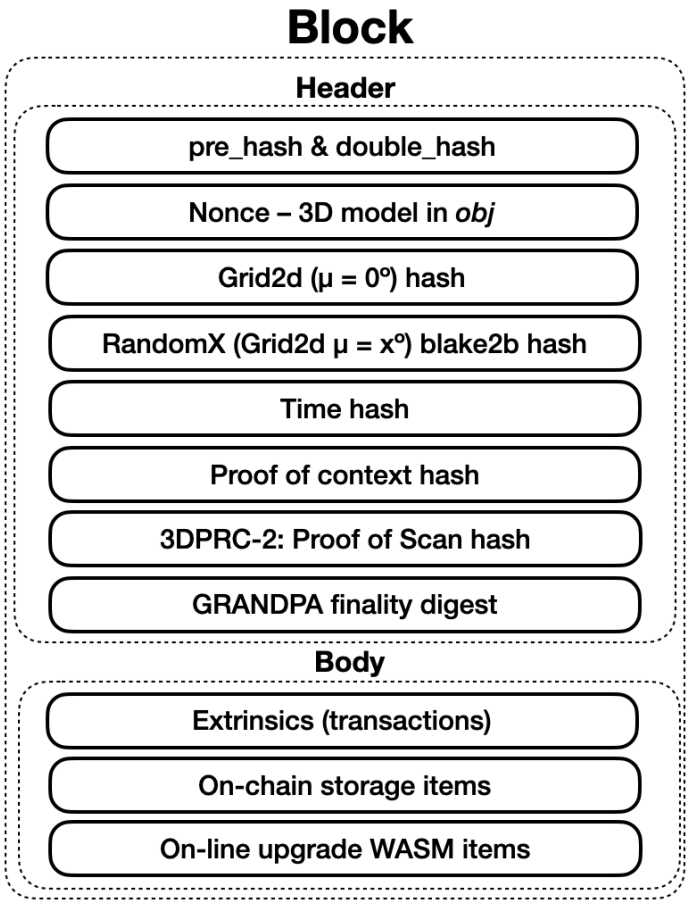
```
vn 0.283063 -5.353324 3.218996
v 18.720348 -73.204567 28.592705
vn 0.485004 -4.463090 4.332598
v 18.970524 -72.797684 29.112286
vn 0.214328 -4.602965 4.216415
v 18.241102 -72.822502 29.146963
vn 0.596465 -5.149169 3.457356
v 19.314495 -73.075134 28.739380
vn -0.907998 -5.773625 2.259512
v 16.425535 -73.428177 27.811790
vn -0.962188 -5.416298 2.968105
v 16.679434 -73.179283 28.457644
vn -1.245864 -5.276230 3.089694
v 16.022053 -73.088127 28.386955
vn -0.492690 -5.834898 2.220737
v 17.292595 -73.480713 27.954941
vn -0.463700 -5.458229 3.000435
v 17.404690 -73.217438 28.568790...
```

- *Grid2d (μ = 0°) hash* – 3D model Hash ID created with no rotation *(μ = 0°)*

```
~/Desktop/3dpass$ ./pass3d -i pir2.obj -a grid2d -g 8 -s 66
Select top 10 hashes
 "72592f8f6ea67c60ca7d9c7683256c3636a30be464952eb82996bff52ca4415d"
"3720e731b9aa04b08d83de34a796cbc389fce2c62365c68206c5610106db053d"
"a65008cdc77f72b47eda70e7c2eb57f93e4fffde5a5356549ac7dbf5d422dffa"
"5930d4a4a98ddff21997daaa8410b151f85dcdb7bfe6b0fb1a05af0e99c276fc"
"6846a36abb6dc50df6845627e6553ede8337e7350254ae8d02b7b7a696c79192"
"b20cf89afb10f14795afe517c82d7f6185da840e6035c48b488792e2df61846d"
"aa4019c8c160da9d2af69edc19589aabd925bc696966b967f92b71947f75f8f0"
"deb83d22570bfc07b8881618dc34a6624616521475bac17798b7348cf6684fd1"
"dd227121b91adcb5beabb0be9412613ebdfde8c5660301eb17583fa644b8793d"
"543e1c3929ea810f4e8c7cfc27f0b60df21a9374089f2278617dae327e32b034"
```

- *RandomX (Grid2d μ = x°) hash* – 3D model Hash ID created with some certain angle of rotation *(μ = x°)* and then processed by the *RandomX*** hashing algorithm on top. The result is presented as a *blake2b*** hash
- *Time hash* – representation of the system time new block was created at

- *Proof of Context hash* – output of the *hist_steps* extension (see more Proof of Work task)
- *Proof of Scan hash* – the user object verification proof in accordance with 3DPRC-2** tokenization standard
- GRANDPA** finality digest



Due to the fact that computing power is required for the object processing, this will require a set of rules and rewards about new blocks. The rules in principal are quite similar to bitcoin, however, P3D mining has an object recognition algorithm involved.

The rules are:

1. New block target time: 1 block per 60 seconds
2. There is a dynamic difficulty adjustment rule, which serves to maintain the network velocity around the block target time (1 block per 60 seconds) in average. Because of the network hashrate volatility, the mining issue has to be made dynamic with respect to difficulty by making it harder or easier to solve in order to ensure that a certain amount of blocks are produced for the aggregation period (1 hour).
3. Grid2d algorithm is part of Runtime logic, which is upgradable online by Open Governance call (see more BLOCK EXECUTION and FORK-LESS UPGRADE).
4. Difficulty adjustment mechanism is part of the Runtime logic, which is upgradable online by Open Governance call (see more BLOCK EXECUTION and FORK-LESS UPGRADE). Current implementation of the Difficulty module is https://github.com/3Dpass/3DP/tree/main/pallets/difficulty

Note, that neither of the rules above touches the user objects verification. So, miners have to obey the rules but the other users do not. They might submit any objects they want, of course, being charged with the verification fee in P3D in accordance to 3DPRC-2** tokenization standard rules.

Once a new block is produced, it is awaiting a confirmation to be chosen as a parent block for the next new block. New block gets finalized in accordance with GRANDPA** protocol by means of the PoA validator set operating, which is another part of the consensus.

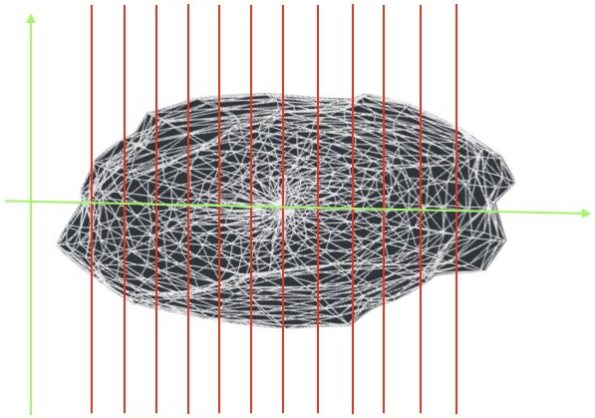
POW (1 NODE = 1 VOTE), ASIC-resistant

Advanced version of the PoW component offers some unique features for distributed mining and GPU/FPGA resistance, such as:

1. Difficulty for the straightforward way of calculations *(m)* < Difficulty for the reverse one *(m^n),*
2. It is guaranteed having 3DPass recognition toolkit implemented into the miners' *(= block constructors in 3DPass)* software to ensure the object verification service delivery,
3. *Grid2d* output is getting processed by the *RandomX*** hashing function to equalize the mining velocity of CPU and GPU to one another
4. Leveraging full blockchain db is required, because of the "proof of context" extension ensuring the dynamic usage of memory in the mining loop, which helps 3DPass resist against FPGA devices sensitive to memory load speed.

Proof of Work task

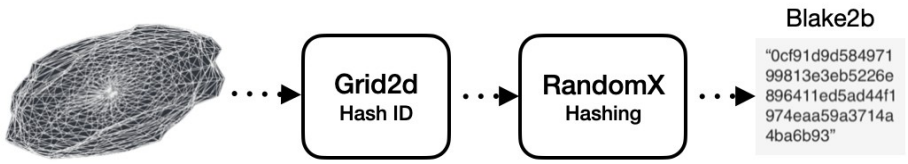
In order to prove 3D object exists at the beginning of the mining loop the bock author first must get 3D object scanned with no rotation angle *(μ = 0°)* resulting as Grid2d output 1 (top 10 hashes). The second component is the *time_hash*, which must timestamp onto the block header and impact the *pre_hash*, as well.



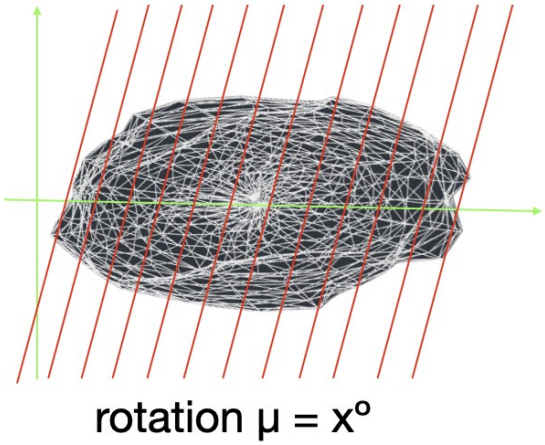
rotation μ = 0°

That is ensured by the *rotation_bytes* (8 bytes, by which the target rotation angle is defined) calculated out of a combination of both the *pre_hash* and the *topest Grid2d zero-angled* hash *(μ = 0°) processed* by the *RandomX** hashing*. So that any modification made on either of them will inevitably impact on the rest. Changing the *pre_hash* will affect the *double_hash* the difficulty limit is applied to.

Getting the Grid2d output RandomX*ed*:



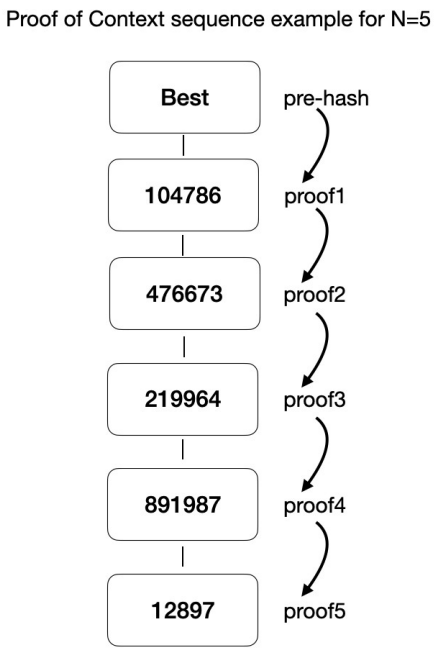
It wasn't until having the object scanned with zero angle that the target rotation angle *(μ = x°)* could be calculated. The same as usual the block author would be challenged with picking up some 3D object shape and get it scanned with the target rotation angle *(μ = x°)* until the *pre_hash* makes *double_hash* meet current *difficulty* number.



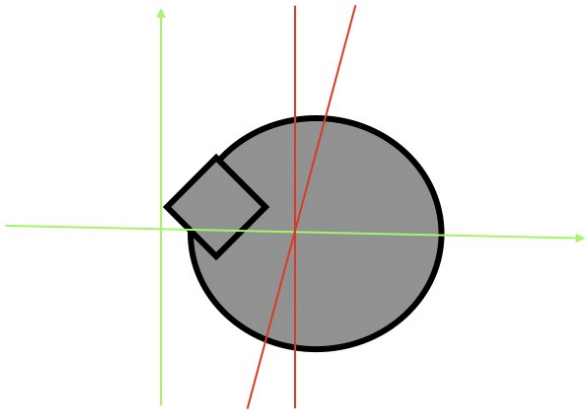
The entire cycle of the object processing would look like this:

Random 3D model in *obj* → Grid2d (3D model *μ = 0°)* output → Random X *(Grid2d μ = 0°)* blake2b hash → rotation_bytes *x = 4 bytes* out of *pre_hash + 4 bytes* out of *RandomX (Grid2d μ = 0°) blake2b* hash → Grid2d (3D model *μ = x°*) output → ***Random X* (Grid2d *μ = x°) blake2b* hash**

However, there will be an additional "proof of context" task to solve, having the access to the full blockchain db is required for. Depending on the pre_hash calculated, there is going to be a directed sequence of N pseudo randomly chosen blocks to pick up *(ex. 104786 → 476673 → 219964 → 891987 → 12897)* and prove its availability for the mining loop. Every mining loop will require another one different sequence to prove.



There is a rule on new block import, which requires: *Grid2d output 1 ≠ Grid2d output 2*. Therefore, the object shape must be complex enough to entirely avoid the Grid2d recognition error that is most likely to face with, especially, when it comes to scanning some regular-shaped objects (ex. a sphere).



Thus, reverse method of calculations (getting 3D object recovered directly out of the *pre_hash* previously found) becomes highly inefficient, comparing to the straightforward one, because of the *rotation_bytes* being kept a secret. Whereas, the straightforward way reveals the secret in the process of calculations. The straightforward method difficulty (m) is related to the reverse difficulty as $m < m^{n.}$

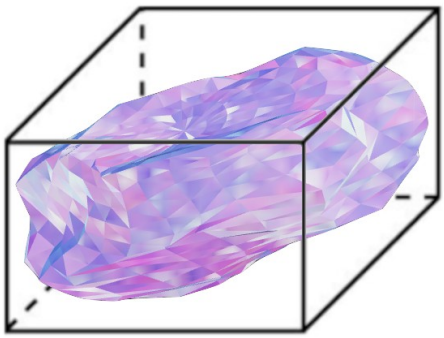
3D OBJECT SHAPE CONSISTENCY CHECK

In order to ensure 3D objects submitted as *nonce* meet the Grid2d recognition algorithm input requirements, there is a bunch of checks on the object shape being applied on the block import by both the *Runtime* and *Native* code (see BLOCK EXECUTION), including but not limited to:

- Vertex {} points to an invalid halfedge

- Halfedge {} pointed to by vertex {} does not start in that vertex, but instead in {}
- Vertex {} does not point to a halfedge
- Halfedge {} points to an invalid twin halfedge {}
- Halfedge twin pointed to by halfedge {} does not point back to halfedge
- Invalid orientation: The halfedge {} and its twin halfedge {} points to the same vertex {}
- Halfedge {} does not point to a twin halfedge
- Halfedge {} points to an invalid vertex {}
- Halfedge {} does not point to a vertex
- Halfedge {} points to an invalid face {}
- Halfedge {} points to a face but not a next halfedge
- Halfedge {} points to an invalid next halfedge {}
- Halfedge {} points to a next halfedge but not a face
- Halfedge next pointed to by halfedge {} does not point back to halfedge
- Length of edge {} is too small
- Face {} points to an invalid halfedge {}
- Halfedge pointed to by face {} does not point to back to face
- Face {} does not point to a halfedge
- Area of face {} is too small ({})
- Vertex {} and Vertex {} is connected one way, but not the other way
- Vertex {} and Vertex {} is connected by multiple edges
- volume > 0.1 * bound_volume (see the figure down below)

bound_volume



These checks will ensure 3D object's surface is simply connected and has sufficient volume to impact on the center of inertia of mass.

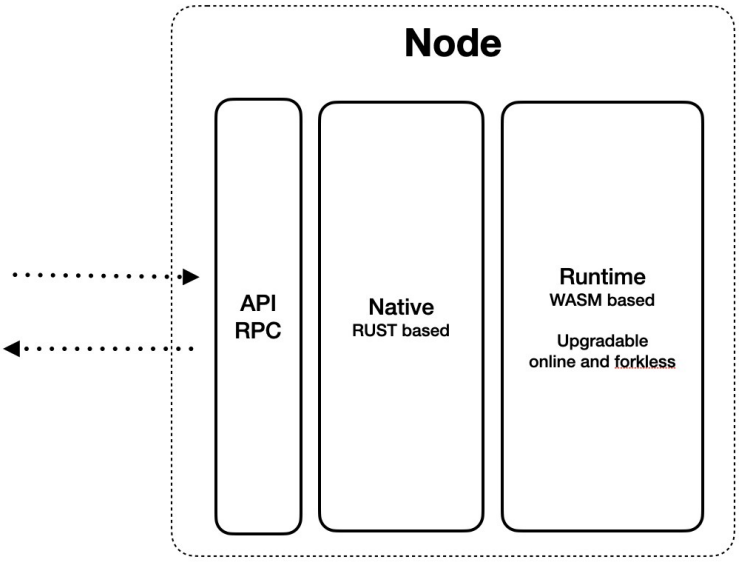
RANDOMNESS

Randomness is one of the most important parts of the consensus. It is leveraged by miners picking up random 3D objects of unique shapes with the target of being as unpredictable as possible. There are two alternative ways for miners to get a 3D object's model. The first one is to get a real world object scanned by a 3D scanner, which would be unpredictable enough but would still require some higher effort. It would be more efficient to generate it on a computer. However, computers are bad at random numbers and would tend to create quite similar-shaped 3D models. Of course, the same or very similar-shaped object will be rejected by the recognition algorithm (depending on the set of parameters applied, *Grid2d* will recognize the object with certain error). Statistically it would accumulate additional difficulty to generate a new unique shape, which have yet not

existed on the blockchain. The more blocks are mined, the more difficulty to find a new unique 3D model for the next block.

BLOCK EXECUTION

3DPass Node logic can be considered as kind of a two-piece design. The first one is the *Native* RUST code and the second one is a *WASM (WebAssembly**)* based Runtime, which is upgradable online. Blocks are being imported through the Native component and then executed by the Runtime logic.



FORK-LESS UPGRADE

By means of using WASM in Substrate** (the framework powering 3DPass) the chain is given the ability to upgrade its runtime logic without hard forking. Hard forking is a standard method of upgrading a blockchain that is slow, inefficient, and error prone due to the levels of offline coordination required, and thus, the propensity to bundle many upgrades into one large-scale event. By deploying WASM on-chain and having nodes auto-enact the new logic at a certain block height, upgrades can be small, isolated, and very specific.

As a result of storing the Runtime as part of the state, the Runtime code itself becomes state sensitive and calls to Runtime can change the Runtime code itself. Therefore the 3DPass Host needs to always make sure it provides the Runtime corresponding to the state in which the entry point has been called.
The Runtime upgrades can be deployed in accordance with the Open Governance procedures, such as Referendums, Technical Committee vote and Council vote.

BLOCK PRODUCTION MECHANISM

New block production begins with the queue handling random 3D models, which are being sent to the input of PoScan via the Node RPC API with *push_mining_object method*:

```
{
    "jsonrpc":"2.0",
    "id":1",
    "method":"push_mining_object",
```

```
      "params": [
          1,
          "o\n
  v 0.05508197844028473 0.7671535015106201 -0.14178061485290527\n
v   0.05349433422088623 0.764365017414093 -0.10946107655763626\n
v   0.04743874818086624 0.7608485817909241 -0.07884219288825989\n
  .....
              ]
  }
```

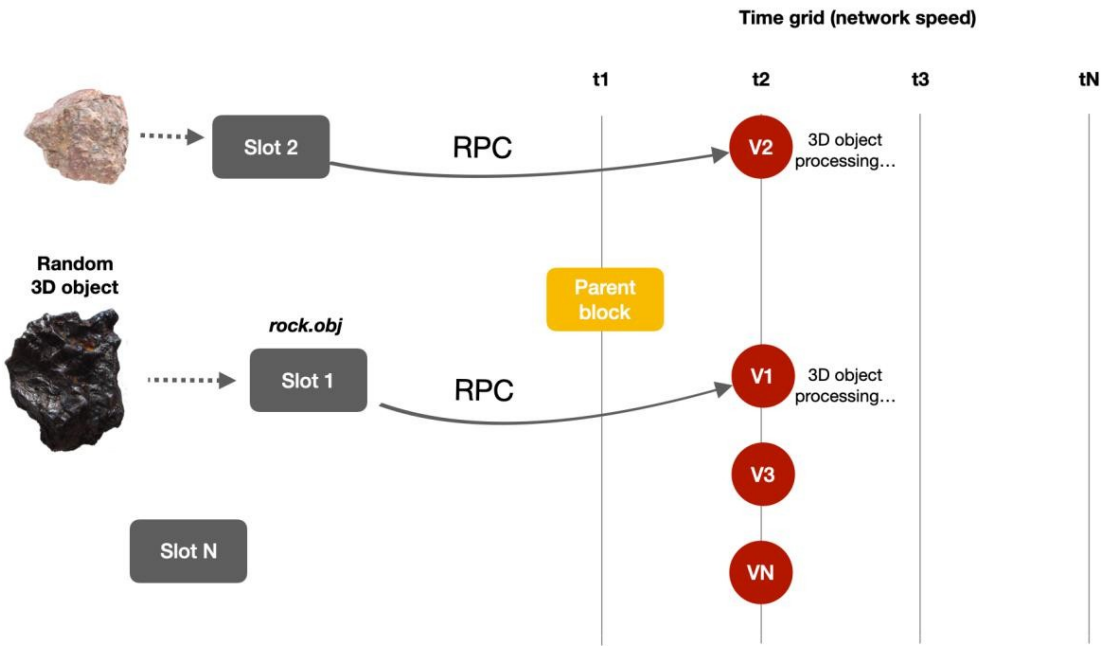
Where one of the parameters is the content of 3D model's file in *obj* format, “\n” is being added at the tail of each line:

```
  v 0.05508197844028473 0.7671535015106201 -0.14178061485290527\n
v   0.05349433422088623 0.764365017414093 -0.10946107655763626\n
v   0.04743874818086624 0.7608485817909241 -0.07884219288825989\n
```

There is a handler on the Node side, which checks the queue and gets 3D objects processed consequentially slot by slot with grid2d recognition algorithm generating HASH ID for each as an output. The Node has selected Best chain by the time. So, miners will always pick up current Best block (the topest block in Best chain) as the parent to construct new one on top. They will join the competition simultaneously, handling 3D objects form the queue in hope to find the one, which makes grid2d produce correct HASH ID used as *nonce*.

In order to create a new block they will use the parent block (the top block of the best chain). (*Figure 2*)

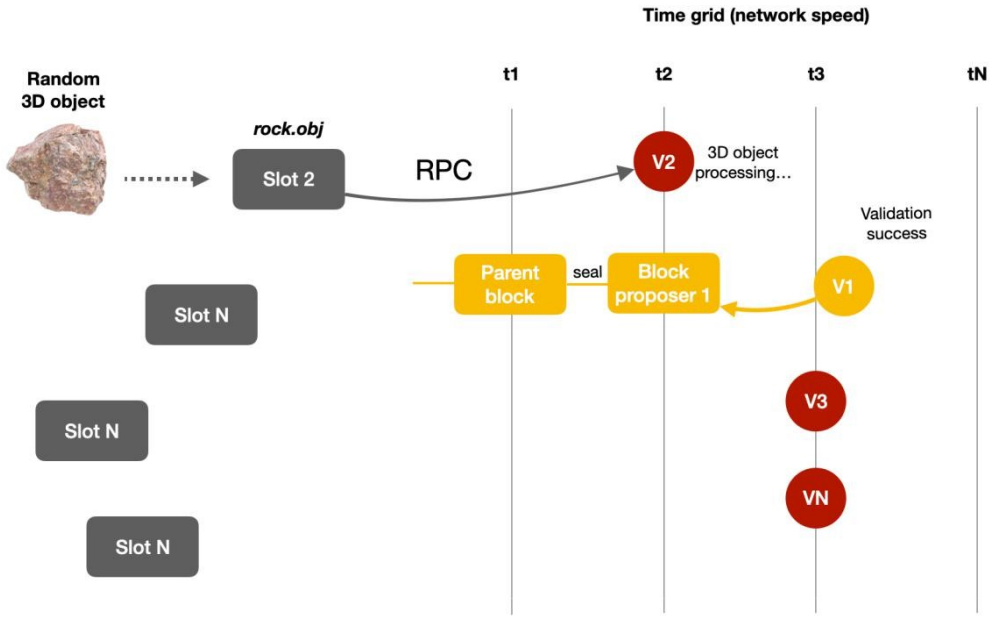
Figure 2



Block author will try sealing the object HASH ID on top of the parent block hash in the blockchain Merkle tree, following the PoW task exactly. If all required proofs are gathered

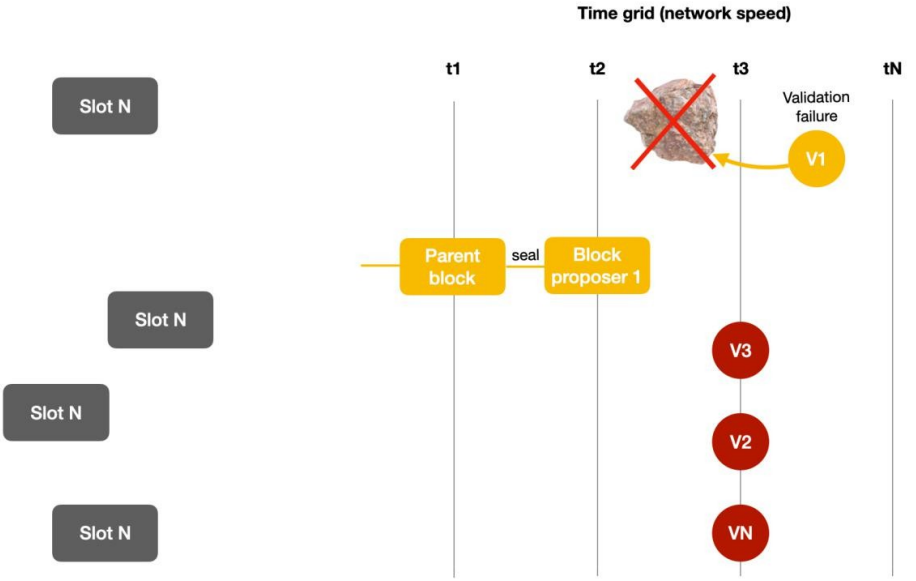
and meet the rules, the new block proposer will be created → executed → signed with the block author signature and broadcasted out through the Node peers. Block execution means the execution of of all the runtime calls/extrinsics and logic of its upgradable modules operating within The Ledger of Things, including but not limited to: transactions, smart-contracts, 3DPRC-2 tokenizaton logic, etc. (*Figure 3*).

Figure 3



There is a block verification procedure implemented on the block import of each Node (peer), which will repeat the PoW task independently and check on the proofs required. Having at least one check not passed will result with the block rejection by the peer, due to the invalid block header. Every runtime module operating will check their proofs during the block execution in accordance of its logic. The block will also be rejected by runtime if the proofs are not delivered or incorrect.(*Figure 4*).

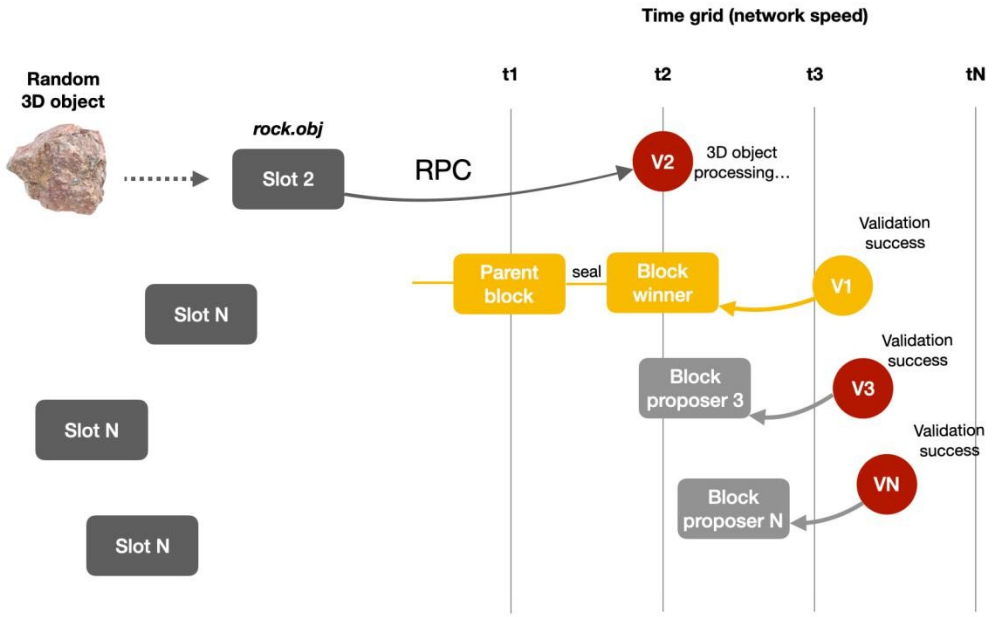
Figure 4



TIE BREAK COMPETITION

The block proposer, if verified, participates in Tie Break competition against the other block authors attempting to construct the block on top of the longest chain, which they believe to be Best chain. This logic is quite commonly used in relation to PoW approach. The block proposed earlier wins. The block time is determined by the time_hash being part of block header (follow the block structure) required for the PoW task to solve. (*Figure 5*)

Figure 5



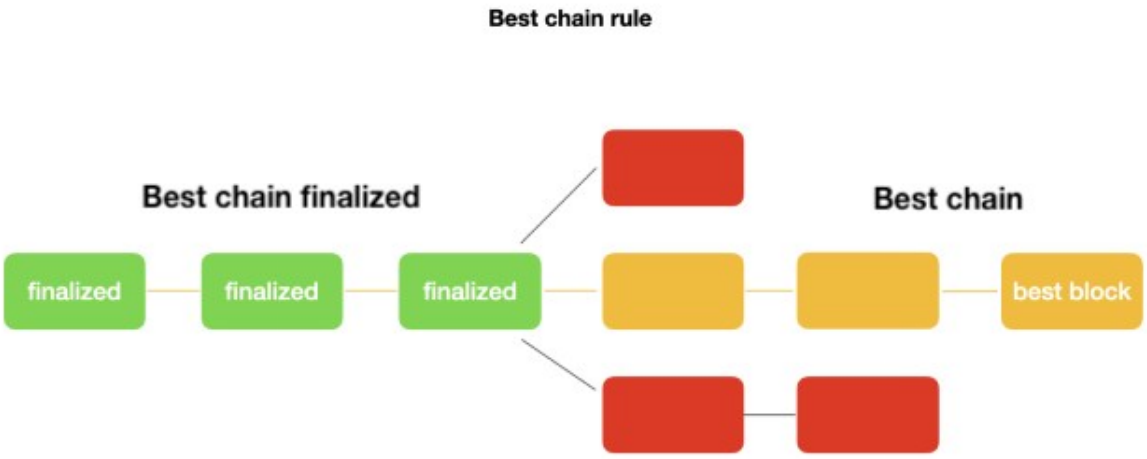
FORK CHOICE RULE

The protocol contains a scoring rule for a given chain (the Best chain rule). Each honest Node will propagate the chain with the highest score it knows about to all the other nodes through its peers. Best chain is selected and must be accepted by all the nodes on the network. Current Best chain rule is the longest chain wins.

The blockchain will stick with the rules of probabilistic finality and might be reorganized during the block authors competition, until Best chain is determined by the second layer of authorities (Validators) leveraging GRANDPA** deterministic finality protocol. Best chain, once finalized, can never be reorganized. Block authors are not allowed to construct new block outside of the finalized chain, because of the GRANDPA proofs being part of block header.

Validators will be monitoring over the blockchain being constructed by the block authors and chasing new blocks to finalize. They will vote upon Best chain and, once consensus is reached, Best chain reaches finality with new blocks include

Figure 6



BLOCK FINALITY GADGET: GRANDPA (POA) PROOF OF AUTHORITY

3DPass network implicates a provable finality protocol called GRANDPA** which guarantees blocks to reach finality in opposite to probabilistic finality (e.x. Nakamoto protocol which first was applied in Bitcoin). Pease refer to the GRANDPA paper to read full description of the protocol.

GRANDPA (PoA) protocol is implemented into The Ledger of Things as a number of traits, such as:

- *validatorSet* pallet is to select and control upon the set of authorities (Validators),
- *Session* pallet is to ensure the rotation in the set,
- *imOnline* pallet is to check is everyone online in the set,
- Grandpa pallet is runtime part of GRANDPA PoA consensus protocol
- *Offences* pallet is to mitigate a risk of the equivocation attack

There is a set of the most reliable nodes on the network called Validator set, which members are allowed to take the authority to vote for Best chain finality, as long as they meet the SLA conditions.

There is a session of 120 blocks length, within which the current validator set is always stable. It can not be changed with new in/out comers until the session is expired. If any validator left the set, in order to come back the selection threshold is required to pass again. Thee session is part of runtime (*Session module*), which is upgradable online and forkless.

Initial SLA conditions are presented as follows, however, a change can be made by the Open Governance vote:

Standard Hardware

The transaction weights in PoScan were benchmarked on standard hardware. It is recommended that Node holders stick to at least the standard in order to ensure the Node capability of processing new blocks on time (block target time = 60 sec). The

following are not minimum requirements, however downgrading hardware below the standard might lead to some issues with the Node performance.

- CPU: Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz
- Storage: A NVMe solid state drive. Should be reasonably sized to deal with blockchain growth. Starting around 80GB–160GB will be okay for the first six months of Realis, but will need to be re-evaluated every six months.
- Memory: 64GB.

There is a strong recommendation for Validators to use a dedicated server to run the Node on. Not only will it require to be capable of handling blocks on time (target time: 60 sec), but also to get the user objects processed in accordance to 3DPRC-2 tokenization standard operating within The Ledger of Things. Unlike many other projects, it is REQUIRED for Validators in 3DPass to provide additional computing power leveraged for the user object processing in accordance to 3DPRC-2 tokenization standard.

Terms and Conditions

- Online: 24/7, ping 333 ms from all over the globe to participate in GRANDPA voting rounds
- Firewall: No restrictions
- Internet traffic: No limits
- Performance at importing new blocks: up to 10 sec per block
- Performance at handling the user objects: up to 20 sec per object
- On-chain identity level of confidence: Reasonable**

Selection threshold

There is a threshold for new validator to pass, which includes some certain amount of P3D locked up for the collateral as well as to prove the set up fee transaction paid to Treasury. The set up fee is required to pay just once. However, the collateral needs to remain locked up all the way through the node operating period. If the lock out period expires, the node is being moved out the validator set. In order to get back the threshold is required to pass again.

Collateral

- 400 000 P3D locked up on the owner's address
- Minimum lock period is 43200 blocks (~ 1 month)

Setup fee

- 10 000 P3D one-time payment to Treasury account: d1EjCsWUVnKTG3dysQC2MWDfZKngtiwV2ZLegWRfFMbUR5d6c

Rewards

- Block finalization: 50% of total block rewards (the rest 50% goes to the block author)
- The user objects processing: 50% of the object verification fee (paid by user)

Penalties

- Equivocation attack (Voting for two different chains simultaneously. It mostly occurs to the validators using the same keyset on different servers and thus causing a threat for the network to get split): 40 000 P3D and getting out of the validator set as well as from the session
- Not being online/available: 20 000 P3D and getting out of the validator set

- Not being able to participate in GRANDPA voting rounds for any reason (Firewall, incorrect keys set up, etc.): 20 000 P3D and getting out of the validator set
- Not being able to get the user object processed in 20 sec: 100% of the user object validation service fee and getting out of the validator set
- Losing Reasonable level of confidence: getting out of the validator set

Locking up funds
The `validatorSet` module provides the `lock` method for Validators to put funds on the collateral:

```
lock (
       amount,
       until,
       period
)
```

- `amount` - the amount in Crumbs (min indivisible units, 1 P3D = 1e12 Crumb) to lock
- `until` - the block number until which funds are going to stay locked. Minimum lock time is 43200 blocks ahead the current best block.
- `period` - auto re-lock option (period): a period in blocks your funds are going to be re-locked over and over automatically. For example, if chosen 45000 blocks, when expired the funds are going to be re-locked for the next 45000 blocks over and over again, until you exit the loop manually. Skip this option if you want to control your locks manually.

Joining the validator set
Either, the validator can join the set by themselves calling out the `validatorSet` module API with the following method or through the Open Governance vote:

```
addValidatorSelf (
             validaortid
)
```

- `validatorid` - validator's account to join

Rejoining the validator set
There is a "comeback window" validator can rejoin after getting ruled out without a necessity to pay the set up fee.

- Ban period: 3 hours, since heading off the validator set
- Comeback window: 1 week, since the ban period got expired

```
rejoinValidaotor (
             validaortid
)
```

- `validatorid` - validator's account to rejoin

Legitimate exit
There is a legitimate way for Validators to exit without getting penalties, which is to wait until the lock period has expired and call out the "unlock funds".

In case the auto re-lock option is being used, placing new lock is required to exit the loop. Just set up a new lock without auto re-lock option (min lock period is + 43200 blocks ahead the current best block) and wait, till it has expired. And then call out the `validatorSet` module with the `unlock` method.

```
unlock (
      amount
)
```

Given the fact, that GRANDPA** gadget in The Ledger of Things is applied on top of PoScan chain already built by the time and providing itself conventional probabilistic finality, the finalization stall situation is a serious incident but might not be fatal. It does not affect the block production as well as the chain moving forward. In case the current validator set is occurred to be incapable to vote, the next validator set takes their turn after the session is expired and keeps up with the best block, voting for all the chain non-finalized yet.

ADDRESSES AND KEYS

Master account in 3DPass is defined by a *mnemonic seed phrase* which can be utilized for the construction of various types of keys. Whatever the key type was used, there is always a possibility to validate the standard account by its public key.

For example, you have got your Master account defined by the mnemonic seed phrase like this one debris minor crater swear crane whale clever into now tone grid proud), and at least two different types of keys can be generated from out out of, such as:

- *sr25519* (Schnorrkel) - standard accounts designed to control funds with
- *ed25519* (Edwards) - used for GRANDPA** finalization

The addresses, however, are derived from the public key with the mainnet *Network ID: 71,* so that a *sr25519* account data would look like this:

```
./target/release/poscan-consensus key generate --scheme Sr25519

Secret phrase: debris minor crater swear crane whale clever into now tone
grid proud
Network ID: 71
Secret seed:
0x3026a7ee1b5014b72287681c68e55b7eca44d11fcfb86254f1efec21845abf9a
Public key (hex):
0x0ed64e59d2d9c1c828a41a0f3cac53d92f99bc99df795e11b804c5ebb2c96b10
Account ID:
0x0ed64e59d2d9c1c828a41a0f3cac53d92f99bc99df795e11b804c5ebb2c96b10 Public
key (SS58): d1CbAtpwbLAtLA6FXnDdR8FzwjqU9kKMYVWUzucaRvWzShEek
SS58 Address: d1CbAtpwbLAtLA6FXnDdR8FzwjqU9kKMYVWUzucaRvWzShEek
```

Let's inspect the same seed phrase or private key to derive an *ed25519 key for* GRANDPA*:*

```
./target/release/poscan-consensus key inspect --scheme Ed25519 "debris
minor crater swear crane whale clever into now tone grid proud"
```

```
Secret phrase: debris minor crater swear crane whale clever into now tone
grid proud Network ID: 71 Secret seed:
0x3026a7ee1b5014b72287681c68e55b7eca44d11fcfb86254f1efec21845abf9a Public
key (hex):
0x6bfcaf1780f0c9f11ee93366c7cf0f24b0a77675966589c82398236da32024d3
Account ID:
0x6bfcaf1780f0c9f11ee93366c7cf0f24b0a77675966589c82398236da32024d3 Public
key (SS58): d1EhJkMSTJQDgnkyet8kaB4QppPzYBX1XJWETKF8C2fv1ruxS SS58
Address: d1EhJkMSTJQDgnkyet8kaB4QppPzYBX1XJWETKF8C2fv1ruxS
```

ImOnline key is to be derived from GRANDPA seed (--suri):

```
./target/release/poscan-consensus key insert --scheme Sr25519 --base-path
~/3dp-chain/ --chain mainnetSpecRaw.json --key-type imon --suri
0x3026a7ee1b5014b72287681c68e55b7eca44d11fcfb86254f1efec21845abf9a
```

THE USER OBJECT TOKENIZATION AS PART OF THE CONSENSUS

There is 3DPRC-2** tokenizaton standard operating on the Ledger of Things. 3DPRC-2 (3Dpass Request for Comments), proposed by PaulS in September 2023, is a standard p2p protocol for the tokenization of objects operating within “The Ledger of Things” decentralized blockchain platform.

Every tokenized object acquires its unique on-chain identity HASH ID. By means of utilization of recognition algorithms implemented, all the assets approved by the network, will be protected from being copied to the extent of the error of the algorithm precision. Please refer the 3DPRC-2** paper for more detail.

3DPRC-2 is implemented as the following components:

1. Advanced version of “Proof of Scan”

The protocol is weaved into “The Ledger of Things” PoW component in a way to tackle the user objects authentication along with the ones being mined. The protocol ensures for users to get a complete service always resulting as either the object acceptance (the asset is allowed to be created) or its rejection (copy is found on the db). The network is responsible for the user object authentication as much as for any block on the blockchain irrespective to the actual dollar value attached to;

2. “0 knowledge proof”

Every judgement provided by miners about the object authenticity is protected by a secret knowledge of its HASH ID** being unavailable for them, until they get the object processed. Every proof is being verified by the majority of the network to make a final decision on whether to accept or reject the block containing the judgement;

3. PoScan module, poscanAssets module (storage and API)

The *PoScan* pallet as well as the *poscanAssets* pallet are both integrated into the network runtime providing the access to the network decentralized storage by means of the object tokenization API, which allows for:

- the user object authentication and its protection from being copied to the extent for the recognition algorithm precision;
- non-fungible digital asset creation;
- property rights definition and its transfers;
- backed cryptocurrency issuance (the tokenization of the object share)

The Object categories and recognition algorithms

This is apparent, that it does not make any sense to compare objects by its HASH ID, provided they got processed with different recognition algorithms/parameters. However, HASH IDs need to be compared in order to guarantee for users the absence of copies on the blockchain data base.

By means of categorization of the object types, we are setting up some “standard” algorithms (presets) to be available for use within each category. And every preset defines the level of precision, at which the object is going to be recognized.

Initial list of categories is presented as follows:

- 3D objects
- Algo: *grid2d_v3a -s 12 -g 8* (learn more GRID2D ALGO)
- 2D drawings
- Music
- Biometrics
- Radio signals
- Movements
- Texts

The categories and presets are being moderated by the Asset committee** vote. The committee members are being assigned by the Council**.

3DPRC-2 allows to endow the object with its additional properties, which will be utilized for the object tokenization. The object properties are categorized as follows:

Relative:

1. *Non-fungible* (propIdx: 0) - allows for the object to be tokenized as a non-fungible asset. MaxSupply=1 rule will be applied to the token created for this property

2. *Share* (propIdx: 1) - allows for the tokenization of the object share (%), the MaxSupply value format of which is restricted to *10^x*. So, the *MaxSupply=10^x* rule will be applied to the token created for this property.

Absolute:

1. *Weigh*t (propIdx: 2) - allows for the tokenization of the object weight, the *MaxSupply* value is limited to *2^128* (max value in the system).

2. Any other measurable object property, which could be presented as an absolute value (ex. Length, Square, Volume, Density, Clarity, Pressure, Time, Speed, Frequency, Bandwidth, Amount of symbols and so forth)

The list of the object properties available on mainnet is being managed by the Council** vote.

The Object authentication protocol

The protocol represents a sequence of actions performing by validators and miners actively participating in the network consensus at the time an object is being submitted by user.

1. Submitting an object

While placing an order, 3dpass user provides the following:

- an object to tokenize: rock.obj (in this example we are going to take 3D model in .obj** format)

- the object HASH ID (top10 hashes from the Grid2d** recognition algorithm output; preset: *-a grid2d_v3a -s 12 -g 8 -d 10*; pass3d** recognition toolkit is being used as the implementation of Grid2d):

```
admin@admin pass3d % ./target/release/pass3d -s 12 -g 8 -a grid2d_v3a -
d 10 -i rock.obj
"7fbe267b208d0ab9c34fa184d4593d0ae39d5bb56852ecc633cb1f6d9eb5aae0"
"460c81f58510e211d162d458459f81575e630817c1b87bc3c0d2c3eec3ae68b7"
"8047c7a4d2a75ce466c9510edb37ac1e98826ef383ab5d4c860b5010e42faa6e"
"5001723ef6d85862733d25765f0700e381cc3a1f6d6b7dbb4efb8aeccc4615d4"
"b8efd617a07099edcdbf2596d10787a6e16a4c59a5e36e46adedc1dd2424b8c9"
"10c8d4984cf840845a7d7785871c70790bf1c55e11a3da54a41d3f21be5a7ea7"
"833a2fc1f26ab9eb9bcc5b4e668f5d6bb91d6f87fb0798f99ce6bb4730dfc218"
"abdedd28e03147d94e0d054d0bf24781bac01fdd81401cd3fb226b5a6e1a5371"
"97c538a99641202385572df245d9cfb300f895a6c94e7caf074023a5ead1c62c"
"db2453f53270c9003a63b4d643d9bcc991ad7630d70a0d1511781f74a7a00fee"
```

- The list of the object additional properties, which will be used as the options and MaxSupply limits on the object tokenizaton
- Authentication fee P3D/Byte: Let's take 100 P3D as an example. This fee will be distributed among the miners and validators taking part in the process (70 miners /30 validators). The fee will be charged for each confirmation ordered (1 confirmation = 1 block). The fee can be set up by Council** vote.
- Storage fee: P3D/Byte (regular network storage fee)
- Number of confirmations: let's take 6, for example. The more confirmations the user orders, the more reliable result He gets, especially, when it comes to the network potentially being attacked at the time. This is unlikely to happen, however, there is always a possibility for every blockchain network to get through this sort of experience. It is expected that the user is able to follow the gap between Best block and the Block finalized (*normally the gap = 2 blocks)* to estimate the network state, before submitting the order.

There is a simple copy check on the object submit extrinsic (transaction). If there is a copycat on the blockchain discovered, the transaction will fail (learn more about pass3d output to know how objects are supposed to be compared by its HASH IDs)

2. Estimation/Anti-SPAM protection

Once having the object submitted and fees paid, Validators (the most reliable nodes of 3Dpass network) from the current GRANDPA validator set start estimating how long would it take for the object to get processed by the network in average. They will try to process the object and, if successful, provide the time in milliseconds at which they managed to get it done like this:

```
Validator 1 - 128
```

```
Validator 2 - 36
Validator 3 - 32 …
```

There is a limit of 3 blocks for the estimation phase to complete.

The validators will have to import the block containing the object and get it processed once again at the next step when miners have added their judgements. Assuming the block target time in 3dpass set up as 60 sec/block, there is a time frame limit of 10 seconds every validator must have finished processing within.

In disregards to the reason why a given validator didn't manage the estimation in time (10 sec), its vote won't be taken. On top of that, every “weird” estimation result (vote) will be statistically ruled out and the average processing time – calculated.

For, example:

```
Validator 1 - 1128 - (ruled out)
Validator 2 - 36 +
Validator 3 - 32 +
Validator 4 - 0.1- (ruled out)
Validator 5 - 24 +
Validator 6 - 18 + …
```

The threshold for the object to pass is established at “2/3 +1” out of the actual number of validators in the set. This is the same threshold ratio as the one being utilized for the GRANDPA finalization. The main requirement for this procedure to work correctly would be having “2/3+1” validators providing true data on the object processing.

30% of the fee paid by the user will distributed among the validators-estimators in equal. Implying, there is about 400 validators in the set, 30 P3D/400= 0.075 P3D each validator gets.

It is assumed, that most of the actual validators are motivated enough to keep the network healthy, otherwise the network as a whole would be considered unsafe. Although, the validators are not providing any substantial proof of their work, which could not be taken for 100% truth by miners taking its turn at the next step.

This is a snapshot example of the object Estimated from the storage (short version, the data is presented partially. See more POSCAN API):

```
[
    [
      4
    ]
    {
      state: {
        Estimated: 2,693
      }
      obj:
0x6f200a7620302e313520302e383520302e320a7620302e313520302e383620....
      category: {
```

```
        Objects3D: Grid2dLow
      }
      hashes: [

0x7fbe267b208d0ab9c34fa184d4593d0ae39d5bb56852ecc633cb1f6d9eb5aae0

0x460c81f58510e211d162d458459f81575e630817c1b87bc3c0d2c3eec3ae68b7

0x8047c7a4d2a75ce466c9510edb37ac1e98826ef383ab5d4c860b5010e42faa6e

0x5001723ef6d85862733d25765f0700e381cc3a1f6d6b7dbb4efb8aeccc4615d4

0xb8efd617a07099edcdbf2596d10787a6e16a4c59a5e36e46adedc1dd2424b8c9

0x10c8d4984cf840845a7d7785871c70790bf1c55e11a3da54a41d3f21be5a7ea7

0x833a2fc1f26ab9eb9bcc5b4e668f5d6bb91d6f87fb0798f99ce6bb4730dfc218

0xabdedd28e03147d94e0d054d0bf24781bac01fdd81401cd3fb226b5a6e1a5371

0x97c538a99641202385572df245d9cfb300f895a6c94e7caf074023a5ead1c62c

0xdb2453f53270c9003a63b4d643d9bcc991ad7630d70a0d1511781f74a7a00fee
      ]
      whenCreated: 2,690
      whenApproved:
      owner: d1f5KGsoZ3xzB6Ecmv92DPizD1x5eToNHM1CPfSC1Xu4nzecN
      estimators: [
        [
          d1ePg4fK97U913xAnZzZ9dUf1W1XUA4bYVC2Zre8K9vjSixnc
          36
        ]
        [
          d1asoD7V6hdff4ExvtjRbdVT398Rg8fKEruYsKFS5P9mkT4fy
          32
        ]
      ]
      estOutliers: []
      approvers: []

      numApprovals: 6
      estRewards: 70,000,000,000,000,000
      authorRewards: 30,000,000,000,000,000
      prop: [
        {
          "propIdx": 0,
          "maxValue": 1
        }
        {
          "propIdx": 1,
          "maxValue": 100000
        }
      ]
    }
  ]
]
```

The object becomes *NotApproved*, if it doesn't pass the estimation within 3 blocks timeframe limit.

3. Gathering confirmations/approvals

Once having the order status at *Estimated*, the actual block author, who found new block, is getting privileged to provide the judgement "*Approved"* about the object. There is no option for miners to provide a negative judgement available.

Any honest block author is expected to do the processing job on the user object. The proof of work is required at this step to be attached to the block. If the proof turns out to be incorrect or fake, the entire block will be rejected by the majority of the network.

There is an additional copy check on the block import for every full 3dpass node. If there is a copycat of the user object on the blockchain discovered, the block will be rejected, either. In order to get a proof the block author should get the object processed with the same preset as the user actually have, with the only exception for the *top11* hashes are to be claimed from *pass3d*:

```
admin@admin pass3d % ./target/release/pass3d -s 12 -g 8 -a grid2d_v3a -d
11 -i rock.obj
"7fbe267b208d0ab9c34fa184d4593d0ae39d5bb56852ecc633cb1f6d9eb5aae0"
"460c81f58510e211d162d458459f81575e630817c1b87bc3c0d2c3eec3ae68b7"
"8047c7a4d2a75ce466c9510edb37ac1e98826ef383ab5d4c860b5010e42faa6e"
"5001723ef6d85862733d25765f0700e381cc3a1f6d6b7dbb4efb8aeccc4615d4"
"b8efd617a07099edcdbf2596d10787a6e16a4c59a5e36e46adedc1dd2424b8c9"
"10c8d4984cf840845a7d7785871c70790bf1c55e11a3da54a41d3f21be5a7ea7"
"833a2fc1f26ab9eb9bcc5b4e668f5d6bb91d6f87fb0798f99ce6bb4730dfc218"
"abdedd28e03147d94e0d054d0bf24781bac01fdd81401cd3fb226b5a6e1a5371"
"97c538a99641202385572df245d9cfb300f895a6c94e7caf074023a5ead1c62c"
"db2453f53270c9003a63b4d643d9bcc991ad7630d70a0d1511781f74a7a00fee"
"ab198306dcee347bd84a942ffc9ab06c8458865fee03c5d1b20ab45b807dc1c5" - the
11th hash (proof of the first miner's work)
```

Miner knows nothing about this 11th hash, until having the object processed. This secret, called "0 knowledge proof", will be leveraged by the network majority while importing the block.

It is totally up to the block author whether or not to trust this collective estimation result coming from validator set. So, there is always a risk for him to loose his block rewards. And there is a benefit to get rewarded on top by user, if the judgement is accepted.

It is assumed, that every miner will rely on himself and make an independent decision about both aspects the object authenticity and the processing time. If any of those fail, the judgement should never be provided.

If the block author decides to skip, the block will not contain the judgement, meaning, there is no confirmation will be gained for the object.

If the block was accepted by the network majority, then *"+1"* confirmation the object gets. Now the next block author takes its turn, and the proof of work will be the *12th hash* from the *grid2d* output:

```
admin@admin pass3d % ./target/release/pass3d -s 12 -g 8 -a grid2d_v3a -d
12 -i rock.obj
"7fbe267b208d0ab9c34fa184d4593d0ae39d5bb56852ecc633cb1f6d9eb5aae0"
"460c81f58510e211d162d458459f81575e630817c1b87bc3c0d2c3eec3ae68b7"
"8047c7a4d2a75ce466c9510edb37ac1e98826ef383ab5d4c860b5010e42faa6e"
"5001723ef6d85862733d25765f0700e381cc3a1f6d6b7dbb4efb8aeccc4615d4"
"b8efd617a07099edcdbf2596d10787a6e16a4c59a5e36e46adedc1dd2424b8c9"
"10c8d4984cf840845a7d7785871c70790bf1c55e11a3da54a41d3f21be5a7ea7"
"833a2fc1f26ab9eb9bcc5b4e668f5d6bb91d6f87fb0798f99ce6bb4730dfc218"
"abdedd28e03147d94e0d054d0bf24781bac01fdd81401cd3fb226b5a6e1a5371"
"97c538a99641202385572df245d9cfb300f895a6c94e7caf074023a5ead1c62c"
"db2453f53270c9003a63b4d643d9bcc991ad7630d70a0d1511781f74a7a00fee"
"ab198306dcee347bd84a942ffc9ab06c8458865fee03c5d1b20ab45b807dc1c5"
"82fd9527291ba30cdddeb5786bac083f8092987e379c9433d81a2eb1bb8c447a" - the
12th hash (proof of the second miner's work)
```

Again, the miner knows nothing about this 12th hash, until having the object processed. If there is no hash below the top11 available, null will be taken for proof.

This procedure repeats itself, up until the block at which required number of confirmations is reached. In this particular example the number is *6*, so the final *6th* judgement will contain the HASH ID expanded to the *top16* hashes:

```
admin@admin pass3d % ./target/release/pass3d -s 12 -g 8 -a grid2d_v3a -d
16 -i miner/rock.obj
"7fbe267b208d0ab9c34fa184d4593d0ae39d5bb56852ecc633cb1f6d9eb5aae0"
"460c81f58510e211d162d458459f81575e630817c1b87bc3c0d2c3eec3ae68b7"
"8047c7a4d2a75ce466c9510edb37ac1e98826ef383ab5d4c860b5010e42faa6e"
"5001723ef6d85862733d25765f0700e381cc3a1f6d6b7dbb4efb8aeccc4615d4"
"b8efd617a07099edcdbf2596d10787a6e16a4c59a5e36e46adedc1dd2424b8c9"
"10c8d4984cf840845a7d7785871c70790bf1c55e11a3da54a41d3f21be5a7ea7"
"833a2fc1f26ab9eb9bcc5b4e668f5d6bb91d6f87fb0798f99ce6bb4730dfc218"
"abdedd28e03147d94e0d054d0bf24781bac01fdd81401cd3fb226b5a6e1a5371"
"97c538a99641202385572df245d9cfb300f895a6c94e7caf074023a5ead1c62c"
"db2453f53270c9003a63b4d643d9bcc991ad7630d70a0d1511781f74a7a00fee"
"ab198306dcee347bd84a942ffc9ab06c8458865fee03c5d1b20ab45b807dc1c5" - 11th
hash (proof of the first miner work)
"82fd9527291ba30cdddeb5786bac083f8092987e379c9433d81a2eb1bb8c447a" - 12th
hash (proof of the second miner work)
"10eac3abc33a75b16c1ca33aaf97db92542a452453265bf5c088dc33be750137" - 13th
hash (proof of the third miner work)
"fdbf00e59e8b4d7ae44950751694ebeb2bb6ac969e166d4500d9bfbd886d7523" - 14th
hash (proof of the forth miner work)
"2405d8048e0fead22be4aa4394104136e15c0ca578299068c2a9dbf3c7c68b59" - 15th
hash (proof of the fifth miner work)
"905f5f8032ff6c956422ac0560431820e2bfa47bb0cf58368fd49dbc1e23e48c" - 16th
hash (proof of the sixth miner work)
```

Once the last confirmation ordered by user is approved by the majority of the network, the object becomes Approved and available for further operations with the asset.

The object becomes NotApproved 5 blocks after last confirmation or the block, at which it was Estimated, if there is still no new confirmation available.

Every block author provided the network with correct judgement is getting rewarded with its share, which is equal to 70%*fee/n, where n is the number of confirmations ordered by user. In this example *n=6* and *70% fee = 70 P3D* So, every honest miner gets *11.666666666666 P3D*.

This is a snapshot example of the object approved from the storage (short version, the data is presented partially. See more POSCAN API):

```
[
    [
      4
    ]
    {
      state: {
        Approved: 2,699
      }
      obj:
0x6f200a7620302e313520302e383520302e320a7620302e313520302e383620....
      category: {
        Objects3D: Grid2dLow
      }
      hashes: [

0x7fbe267b208d0ab9c34fa184d4593d0ae39d5bb56852ecc633cb1f6d9eb5aae0

0x460c81f58510e211d162d458459f81575e630817c1b87bc3c0d2c3eec3ae68b7

0x8047c7a4d2a75ce466c9510edb37ac1e98826ef383ab5d4c860b5010e42faa6e

0x5001723ef6d85862733d25765f0700e381cc3a1f6d6b7dbb4efb8aeccc4615d4

0xb8efd617a07099edcdbf2596d10787a6e16a4c59a5e36e46adedc1dd2424b8c9

0x10c8d4984cf840845a7d7785871c70790bf1c55e11a3da54a41d3f21be5a7ea7

0x833a2fc1f26ab9eb9bcc5b4e668f5d6bb91d6f87fb0798f99ce6bb4730dfc218

0xabdedd28e03147d94e0d054d0bf24781bac01fdd81401cd3fb226b5a6e1a5371

0x97c538a99641202385572df245d9cfb300f895a6c94e7caf074023a5ead1c62c

0xdb2453f53270c9003a63b4d643d9bcc991ad7630d70a0d1511781f74a7a00fee
      ]
      whenCreated: 2,690
      whenApproved: 2,699
      owner: d1f5KGsoZ3xzB6Ecmv92DPizD1x5eToNHM1CPfSC1Xu4nzecN
      estimators: [
        [
          d1ePg4fK97U913xAnZzZ9dUf1W1XUA4bYVC2Zre8K9vjSixnc
          36
        ]
        [
          d1asoD7V6hdff4ExvtjRbdVT398Rg8fKEruYsKFS5P9mkT4fy
          32
        ]
      ]
```

```
      estOutliers: []
      approvers: [
        {
          accountId: d1jygGfK97U913xAnZzZ9dUf1W1XUA4bYVC2Zre8KTrgnjs6
          when: 2,695
          proof:
0xab198306dcee347bd84a942ffc9ab06c8458865fee03c5d1b20ab45b807dc1c5
        }
        {
          accountId: d19ouGfK97U9edj69nZzZ9dUf1W1XUA4bYVC2Zre8K9Jgclj
          when: 2,696
          proof:
0x82fd9527291ba30cdddeb5786bac083f8092987e379c9433d81a2eb1bb8c447a
        }
      ]
      numApprovals: 6
      estRewards: 70,000,000,000,000,000
      authorRewards: 30,000,000,000,000,000
      prop: [
        {
          "propIdx": 0,
          "maxValue": 1
        }
        {
          "propIdx": 1,
          "maxValue": 100000
        }
      ]
    }
  ]
]
```

New algorithm requirements

This protocol is proposed to be a standard for the tokenization of objects corresponding to the categories specified (see more “THE OBJECT CATEGORIES AND RECOGNITION ALGORITHMS”)

- Every recognition algorithm to add must provide *“0 knowledge proof”* about the object, so that the network can verify and agree upon its authenticity. 3D object authentication procedure utilizing the *grid2d* recognition algorithm can be used as a reference.
- Every recognition algorithm must be open source and free from license restrictions, which could prevent its distribution for free.
- Every recognition algorithm must be able to get objects processed within the timeframe limit of 10 sec, while running on average full 3dpass node.
- Every recognition algorithm must rely on public data, stored on 3dpass blockchain (objects, HASH IDs, zero knowledge proof).

Property rights definition

One of the most important challenges related to the tokenization of objects all across the entire blockchain world is how could the property rights be initially defined. Simply

speaking, how could we verify the actual owner of the object/asset at the time the object is being submitted by someone claiming they have all the rights required for.

3DPRC-2 proposes a trustworthy procedure for the property rights definition and its correction after any potential changes, which might had happened as an outcome of any argument or adjudication, any other legal action leading to the rights correction outside of the blockchain storage, which needs to be taken by “The Ledger of Things” for direct order and, finally, got executed.

Let’s classify the property rights into the following aspects:

1. Common public assets
Intangible or tangible assets the property rights has expired for (50 years after the author’s death has actually lasted by the time the asset appears on “The Ledger of Things”). For example, it could be a piece of classic music or Venus de Milo statue. Those assets are most commonly priceless and obtained by the humanity as a whole, representing the legacy of human civilization. Everyone is allowed for its replications commercial use (ex, anyone can listen to and make a copy of any piece of classic music without any license fee requirements/restrictions). Although, the original object properties can be identified by the recognition algorithms (its shape, weight, melody, voice, etc.)

3DPRC-2 will always allow for anyone to issue crypto currencies (fungible and non-fungible) backed by any common public asset stored on “The Ledger of Things”, due to the fact that there is no restrictions could be applied.

2. Private assets
Intangible or tangible assets, the property rights of which are established and still valid by the time the asset appears on “The Ledger of Things”. The asset is obtained by either a person/group of persons or an organization. In terms of the world wide intellectual property legislation, the property rights have been valid since the first reliable publication made by its author. Tangible assets are usually published on the government registries (real state, vehicles, etc).

3DPRC-2 will always guarantee for anyone, who is the actual object owner, to establish their property rights towards the asset stored on “The Ledger of Things”, provided the ownership has been verified by means of special property rights extension(option) of The Object authentication protocol (see more THE OBJECT AUTHENTICATION PROTOCOL).

The extended option of the protocol will use a list of trustworthy external resourced, which data can be considered reliable enough for making decisions about the on-chain private assets, to verify the asset owner.

For example, https://www.wipo.int/ - WIPO (World Intellectual Property Organization)**
The list of resources can be moderated by the Asset committee** vote.

The ownership verification protocol:

1. By means of leveraging pass3d** recognition toolkit, the object owner must have created the multi-object HASH ID, the seed of which would be presented as a

combination of the following components: A. The object tokenized (ex. 3D model in *.obj* format) B. The owner biometric data (optional)

2. By means of using 3dpass wallet** signature, the object owner must have composed a message in the format as follows:

```
-- Start message --
object: 0x6f200a7620302e313520302e383520302e320a7620302e313520302e383620....
category: Objects3D: Grid2dLow hashes:
7fbe267b208d0ab9c34fa184d4593d0ae39d5bb56852ecc633cb1f6d9eb5aae0
460c81f58510e211d162d458459f81575e630817c1b87bc3c0d2c3eec3ae68b7
8047c7a4d2a75ce466c9510edb37ac1e98826ef383ab5d4c860b5010e42faa6e
5001723ef6d85862733d25765f0700e381cc3a1f6d6b7dbb4efb8aeccc4615d4
b8efd617a07099edcdbf2596d10787a6e16a4c59a5e36e46adedc1dd2424b8c9
10c8d4984cf840845a7d7785871c70790bf1c55e11a3da54a41d3f21be5a7ea7
833a2fc1f26ab9eb9bcc5b4e668f5d6bb91d6f87fb0798f99ce6bb4730dfc218
abdedd28e03147d94e0d054d0bf24781bac01fdd81401cd3fb226b5a6e1a5371
97c538a99641202385572df245d9cfb300f895a6c94e7caf074023a5ead1c62c
db2453f53270c9003a63b4d643d9bcc991ad7630d70a0d1511781f74a7a00fee
-- End message --

-- Start P3D wallet signature --
0xfa020a57c5ef765a65610d5985b7bb391f95abf79adb9eaa598e751d5f9ea02b00673d879f82
6f347565473db512ce472711c6f7742bf546df3723ec610f928c
-- End P3D wallet signature --

-- Start public key --
d1ELrwRkSw5bXwg7NVSbpWMdjdc7oBF32ESx2cZpCUfqXZwhe
-- End public key --
```

The message must be signed with the owner P3D account (in the example above *d1ELrwRkSw5bXwg7NVSbpWMdjdc7oBF32ESx2cZpCUfqXZwhe*).

3. The object owner must have published the message on the db of any reputable resource from the actual list available on the network storage (for example https://www.wipo.int/ - WIPO). The message must be accessible via the open public API (as an additional properties, for example), so that the validators can verify the data from all around the globe automatically.
4. Once having all these 3 steps above done, the object owner submits the authentication order (see more THE OBJECT AUTHENTICATION PROTOCOL) with the option ‘ownership.proof’. A public link/reference to the object on the external data base must be provided as the proof.
5. The object authentication protocol will be executed as it is described in the THE OBJECT AUTHENTICATION PROTOCOL chapter, but with the couple additional rules. The rules are: A. The network nodes will try to verify the message via the API (external worker (oracle) will be used on the node side). The judgement “Approved” will be accepted if the message will be verified on block import, otherwise the block is going to be rejected; B. If any copycat of the object has been discovered, the actual owner/owners will be applied.

3. Unpublished assets

Intangible or tangible assets, the property rights of which are NOT established by the time the asset appears on “The Ledger of Things”. For example, it could have been something like new 3D statue or new pop music track, anything unpublished yet.

The combination of such aspects as the timestamp, copy protection, data change protection make “The Ledger of Things” one of the most reliable decentralized data bases to make the first publication for any new object created. It would be enough for its owner to get the object through the authentication procedure and claim they have all the rights required.

Backed currency issuance
Backed asset creation (*Share token* or *Non-Fungible*):

This method, provided by the *poscanAssets* module, allows for the object owner to create a backed currency (a token backed by the object property). Only one of the object properties is allowed to be tokenized, as long as the object is Approved at the authentication stage (see more *poscan.putObject*).

By means of dealing with the object properties, it is possible to turn the object into either Fungible or Non-fungible asset, depending on the purpose of its tokenization. For example, the tokenization of the object Share (as well as such properties as Weight, Square, Volume, Length, etc) will always stand for its collective ownership or ICO (Initial Coin Offering). These properties will always be tokenized as Fungible assets, the MaxSupply of which is limited to the property value attached to the object. For example, if the object weight is 1000 gram, then the token MaxSupply=1000 limit will be set up for the token created (you won't be able to issue more than 1000 minimum indivisible units). While transferring tokens, the object share is being transferred.

There is a specific property named *Non-Fungible*, which is leveraged to get the object tokenized as a Non-fungible asset. If chosen, the *MaxSupply = 1* limit will be applied to the token created. Whereas 1 is the minimum indivisible unit of The Ledger of Things. By means of transferring this unit, the ownership of the entire object is being transferred.

```
create(
    id,
    admin,
    minBalance,
    objDetails
)
```

- id: Compact<*u32*> - the index id for the asset
- admin: AccountId - the admin P3D account
- minBalance: *u128* - min balance in tokens to keep any account alive (accounts going below min balance are going to be removed)
- objDetails:
  - objIdx: *u32* - the object index id on the poScan module storage
  - propIdx: *u32* - the property index id on the poScan module storage
  - maxSupply: *u128* - MaxSupply limit in tokens, which is going to be apply to the asset. Must not exceed the object property MaxValue (see more *poscan.putObject*).

The poscanAsset logic gets the asset bounded to the user object and thus guarantees to have the token MaxSupply complied with the limit set up by the property value.

Setting up the asset metadata:

This method allows to set up metadata for the asset actually created.

```
setMetadata(
         id,
         name,
         symbol,
         decimals
)
```

- id: *Compact<u32>* - the index id of the asset, actually existing on the poscanAssets module storage, you are about to set up metadata for
- name: *Bytes* - the asset name (ex. "My very expensive diamond shares")
- symbol: *Bytes* - the asset symbol (ex. XYZ - 1000 XYZ)
- decimals: *u8* - the number of decimals applied to the asset (ex. decimals: 4 will set up min indivisible unit at 0.0001 XYZ)

Tokens minting:

Once having the asset created and set up its metadata, it is possible to use this method to mint some tokens. *MaxSupply* limit the asset has been created with cannot be exceeded.

```
mint(
   id,
   amount
)
```

- id: Compact<*u32*> - the index id of the asset, actually existing on the poscanAssets module storage, you are about to mint tokens for
- amount: Compact<*u128*> - amount to mint

Token transfers:

This method allows to transfer minted tokens from one account to another. Some P3D is required to cover the Ledger of Things transaction fee.

```
transfer(
        id,
        target,
        amount
)
```

- id: Compact<*u32*> - the index id of the asset, actually created on the poscanAssets module storage
- target: AccountId - P3D account to receive the transfer
- amount: Compact<*u128*> - amount to transfer

PoScan API:

- Explore the actual API methods available on 3DPass WIKI: https://github.com/3Dpass/3DP/wiki/3DPRC%E2%80%902-PoScan-API

- PoScan module implementation: https://github.com/3Dpass/3DP/tree/test/pallets/poscan
- poscanAssets module implementation: https://github.com/3Dpass/3DP/tree/main/pallets/poscan-assets

SMART CONTRACTS AND PRIVATE CHAINS

Any object transformed and verified might be applied as a subject for the smart-contract running on 3Dpass network. A variety of ways have been considered to utilize this option such as:

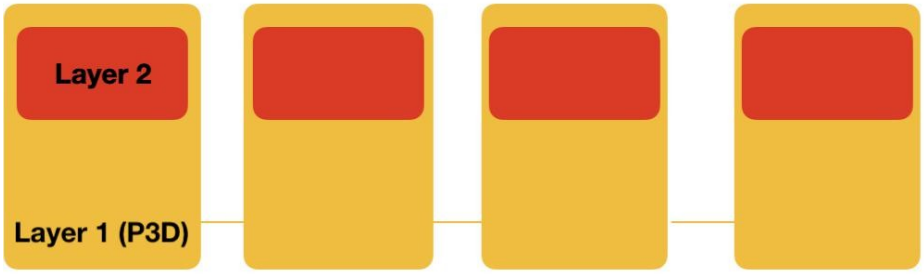
- Issuing 3DPRC-2 NFT single token backed by 3D object
- Private backed share token, the unit of which would be a quantum of a 3D object (gram, 1 kilogram, 1 meter, etc).
- Decentralized apps development, marketplaces, etc

This functionality has been implemented according to Substrate Smart Contracts Toolkits**. Please see the document in the reference section for more details.

Proof of Scan consensus allows the creation of your own rules to produce chains of limited supply assets for gaming, metaverse, augmented as well as virtual reality. All that is required is to utilize the same verification mechanism.

Conditions can be proposed for your objects submitted, which will put some limitation to the total supply. For example, you can set up a rule of only 3D models accepted which would have up to 9 peaks on the surface and a HASH ID that contains at least 10 prime numbers. Utilize 3DPass new block rules as a template.

Figure 7



P3D UNIT

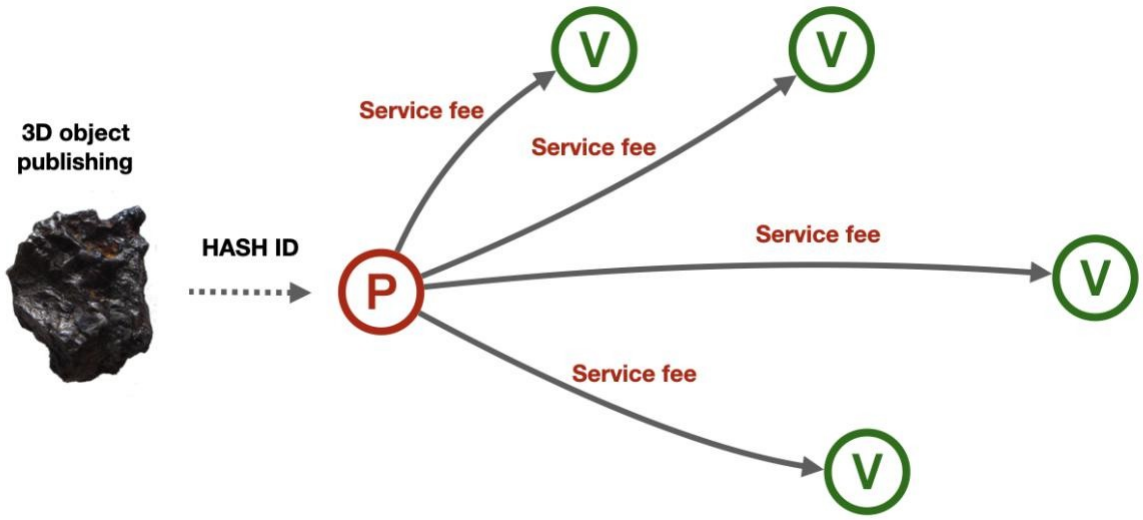
The native utility currency unit is P3D which is required especially for internal needs such as 3D object processing encouragement. The smallest indivisible unit of account in 3DPass is called Crumb. 3DPass coins (P3D) are equal to 1e12 Crumb:

- Crumb: 0.000000000001
- MicroP3D (uP3D): 0.000001000000
- MilliP3D (mP3D): 0.001000000000
- P3D: 1.000000000000

ECONOMICS

The economy model is straightforward, there are always some validators on demand providing 3D object authentication for customers such as miners and regular users with a with a fee. Miners will publish 3D objects in hope of getting block rewards from the network if those are meeting the specified conditions. Regular users do the same for their own commercial reasons (ex. buying&selling those objects in digital space). Irrespective of the purpose for the object publication, Validators charge their service fee which might differentiate from one block to another.

Figure 8



COMMISSIONS AND REWARDS

New block rewards

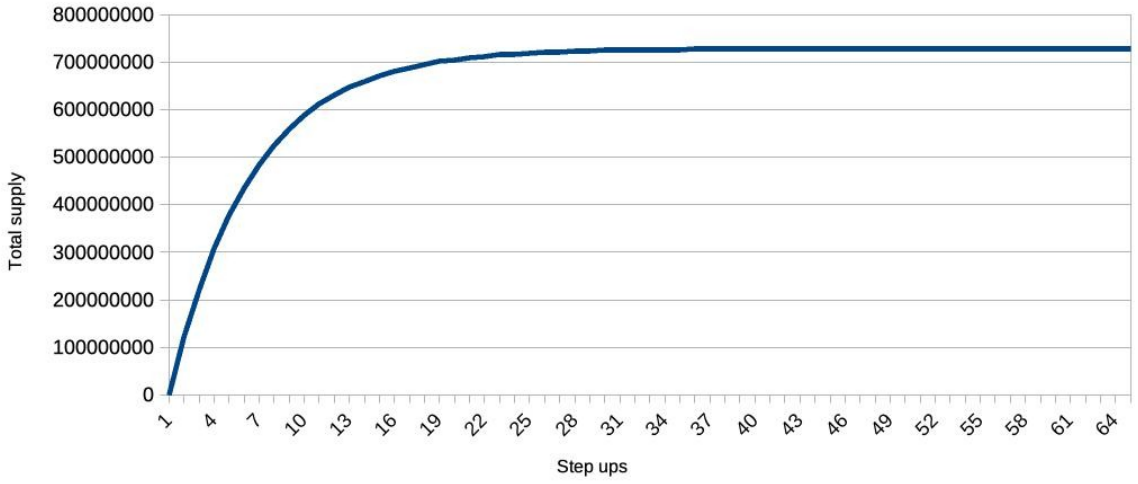
Both Miners and Validators put their efforts on finding a 3D object shape corresponding to the rules that govern whether or not a new block author would be rewarded by the network. Block rewards will be split 50/50 *(the ratio is managed by Open Governance vote and could be changed)* between the Block author and Validators (the validators share is to be distributed among all the Validators in the validator set in equal). These are the rules for new block rewards calculation:

1. Starting from the block going after the genesis, mining rewards amount is established as 500 P3D per block,
2. There is a step up event happening every 243000 blocks, at which current rewards will be reduced dividing by 1.2.

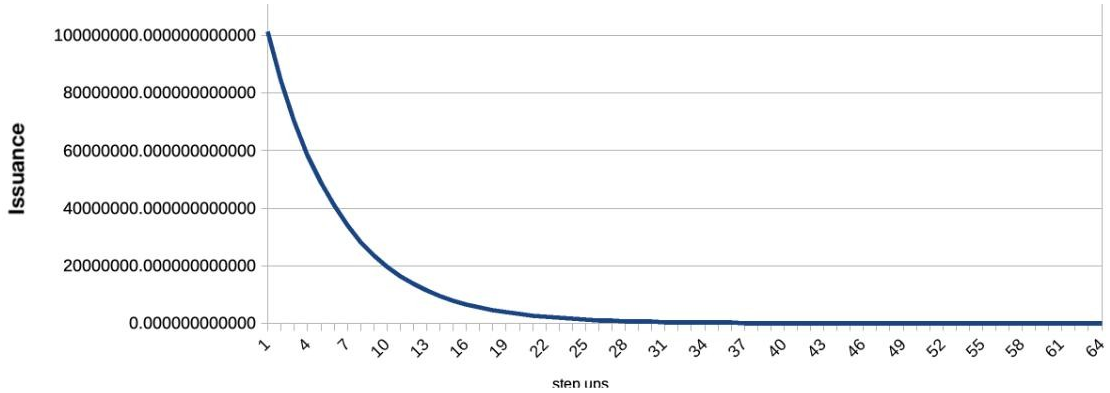
Given the network velocity at 1 block/60 sec, reaching next step throughout the period of 243000 blocs would approximately take about 169 days.

step up N	blocks/step	P3D per block	emission	total supply
				0
1	243000	500.0000	121500000.000000000000	121500000
2	243000	416.6667	101250000.000000000000	222750000
3	243000	347.2222	84375000.000000000000	307125000
4	243000	289.3519	70312500.000000000000	377437500
5	243000	241.1265	58593750.000000000000	436031250
6	243000	200.9388	48828125.000000000000	484859375
7	243000	167.4490	40690104.166666700000	525549479.166667
8	243000	139.5408	33908420.138888900000	559457899.305556
9	243000	116.2840	28257016.782407400000	587714916.087963
10	243000	96.9033	23547513.985339500000	611262430.073303
11	243000	80.7528	19622928.321116300000	630885358.394419
12	243000	67.2940	16352440.267596900000	647237798.662016
13	243000	56.0783	13627033.556330700000	660864832.218346
14	243000	46.7319	11355861.296942300000	672220693.515289
15	243000	38.9433	9463217.747451900000	681683911.26274
16	243000	32.4527	7886014.789543250000	689569926.052284
17	243000	27.0439	6571678.991286040000	696141605.04357
18	243000	22.5366	5476399.159405040000	701618004.202975
19	243000	18.7805	4563665.966170870000	706181670.169146
20	243000	15.6504	3803054.971809050000	709984725.140955
21	243000	13.0420	3169212.476507540000	713153937.617462
22	243000	10.8684	2641010.397089620000	715794948.014552
23	243000	9.0570	2200841.997574680000	717995790.012126
24	243000	7.5475	1834034.997978900000	719829825.010105
25	243000	6.2896	1528362.498315750000	721358187.508421
26	243000	5.2413	1273635.415263130000	722631822.923684
27	243000	4.3677	1061362.846052610000	723693185.769737
28	243000	3.6398	884469.038377172000	724577654.808114
29	243000	3.0332	737057.531980977000	725314712.340095
30	243000	2.5276	614214.609984147000	725928926.950079
31	243000	2.1064	511845.508320123000	726440772.458399
32	243000	1.7553	426537.923600102000	726867310.382
33	243000	1.4628	355448.269666752000	727222758.651666
34	243000	1.2190	296206.891388960000	727518965.543055
35	243000	1.0158	246839.076157467000	727765804.619213
36	243000	0.8465	205699.230131222000	727971503.849344
37	243000	0.7054	171416.025109352000	728142919.874453
38	243000	0.5878	142846.687591127000	728285766.562045
39	243000	0.4899	119038.906325939000	728404805.46837
40	243000	0.4082	99199.088604949000	728504004.556975
41	243000	0.3402	82665.907170790800	728586670.464146
42	243000	0.2835	68888.255975659000	728655558.720122
43	243000	0.2362	57406.879979715900	728712965.600101
44	243000	0.1969	47839.066649763200	728760804.666751
45	243000	0.1641	39865.888874802700	728800670.555626
46	243000	0.1367	33221.574062335600	728833892.129688
47	243000	0.1139	27684.645051946300	728861576.77474
48	243000	0.0949	23070.537543288600	728884647.312284
49	243000	0.0791	19225.447952740500	728903872.760236
50	243000	0.0659	16021.206627283700	728919893.966864
51	243000	0.0549	13351.005522736500	728933244.972386
52	243000	0.0458	11125.837935613700	728944370.810322
53	243000	0.0382	9271.531613011430	728953642.341935
54	243000	0.0318	7726.276344176190	728961368.618279
55	243000	0.0265	6438.563620146830	728967807.181899
56	243000	0.0221	5365.469683455690	728973172.651583
57	243000	0.0184	4471.224736213070	728977643.876319
58	243000	0.0153	3726.020613510890	728981369.896932
59	243000	0.0128	3105.017177925750	728984474.91411
60	243000	0.0106	2587.514314938120	728987062.428425
61	243000	0.0089	2156.261929115100	728989218.690354
62	243000	0.0074	1796.884940929250	728991015.575295
63	243000	0.0062	1497.404117441040	728992512.979413
64	243000	0.0051	1247.836764534200	728993760.816177

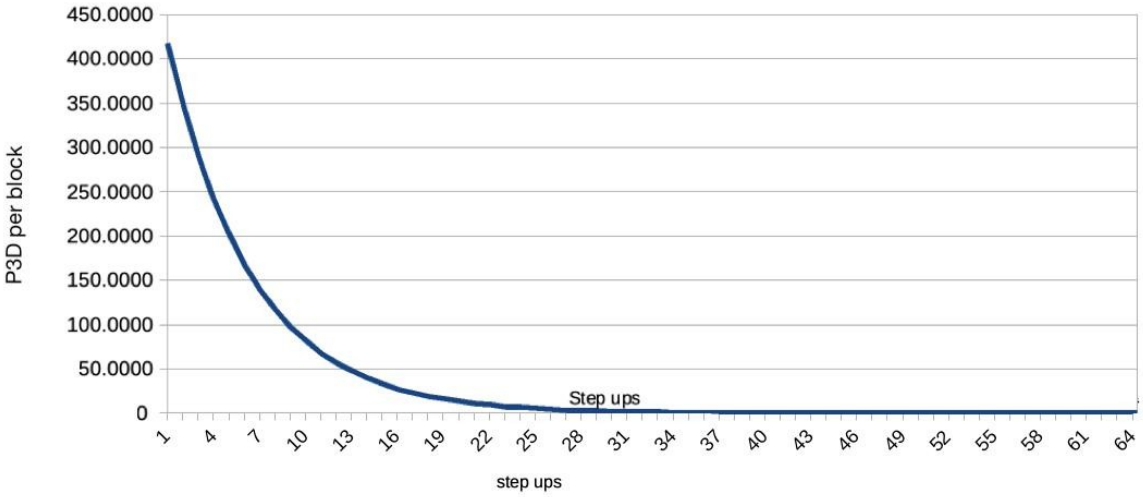
P3D rewards curve



P3D issuance



P3D per block



The objects verification service fee

Regardless of whether the object was published by a miner or any other user, it's always up to Validators to decide how much P3D is appropriate fee for the service in particular moment. It always comes to computing power, which is required for 3D object shape recognition to be applied. This requires costs, energy, gear, etc. Miners and other users will propose a certain fee for each 3D object. The measure of validation fee is P3D/byte, where the byte is 1 byte of the 3D object data to be processed. The validation fee divides among the Validators that performed the processing and completed the additional security token check (see more VERIFICATION MECHANISM). The minimum commission amount is 1 Crumb/byte.

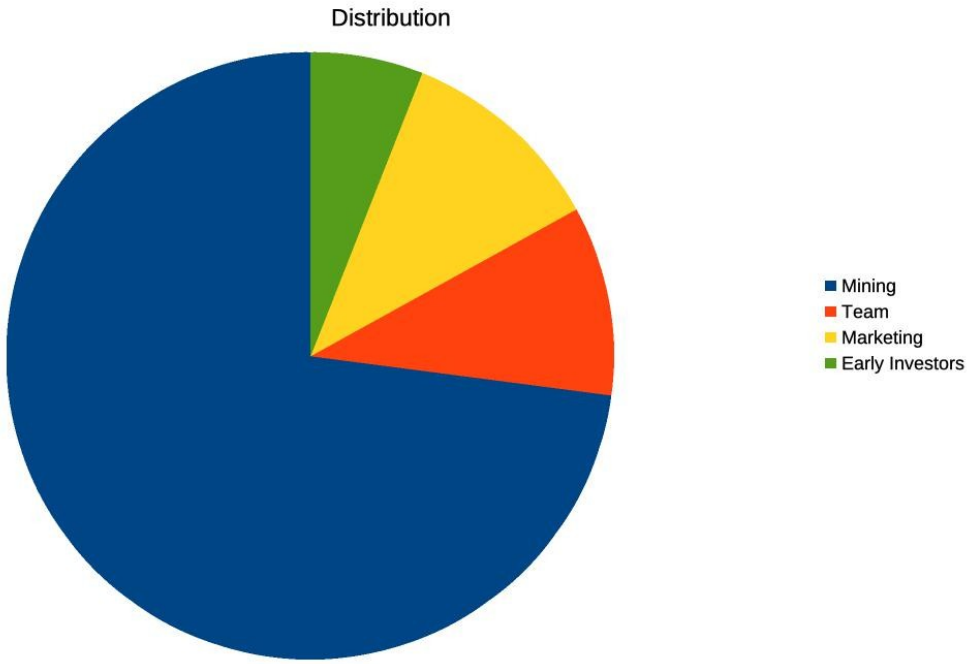
Transaction fee

3DPass leverages a transaction fee mechanism as conventional as the most blockchain networks do. Any transaction might be attached to a new block by one of the validators accepting or rejecting the fee amount set up by a sender. Minimum commission amount is 1 Crumb/byte, where the byte is 1 byte of the transaction weight.

The “Crumb” is 0.000000000001 P3D (smallest indivisible unit)

TOTAL SUPPLY AND SHARES DISTRIBUTION

1. Total supply: 1 000 000 000 P3D
2. Mining rewards: 729 000 000 P3D (72.9 % of total supply)
3. Team share: 101 000 000 P3D (10.1% of total supply), issued in genesis block
4. Early Investors: 60 000 000 P3D (6% of total supply), issued in genesis block
5. Marketing budget: 110 000 000 P3D (11% of total supply)



Detail:

Mining – 729 000 000 P3D (72.9 % of total supply, including the testnet mining rewards ~ 5.7%)

The team share – 101 000 000 P3D (10.1% of total supply), issued in genesis block. Vesting schedule: 30 000 000 P3D locked until the block #905700; 30 000 000 P3D locked until the block #1431300; 30 000 000 P3D locked until the block #1956900:

d1CNDotJXNPvnSA5EQXpSbkUyXBVmaggkARY7kcgXim4BqeBJ
d1E8Bh1ZoTjnSfRnnQCKgteV1ipd9yMvK3dQnD3gAHLv3notP
d1GejSwamhiKSGP9sbChYq4GJjWSrpA7v3PdfNijdwL8aGYCB
d1GZ8GxP3KzKJGRYmp9HMwxurnSKx3ACcqeZqLY5kpbLEyjzE

d1F9pWAgHjcADhxrg9DeuiE4KJaNnFBzxCHJvsJUysygrWFnQ
d1GjsUT5uKxmzrBZo2ed7Q5Woc8gi3g6mRenK8nXw9qhA9SSx
d1GA9xWx3WgpQHp8LHCXHbYoZdvjY3NHhU6gR2fsdVCiC4TdF
d1HBZPs7ZMs5wa3PWk6RorjXKXHx85XkuD9JjN3nyEofW5MBp
d1Hun336VEgwk5tYFs6TusUESTyKMrLyHh3Yk3d2LMdZqaeDp

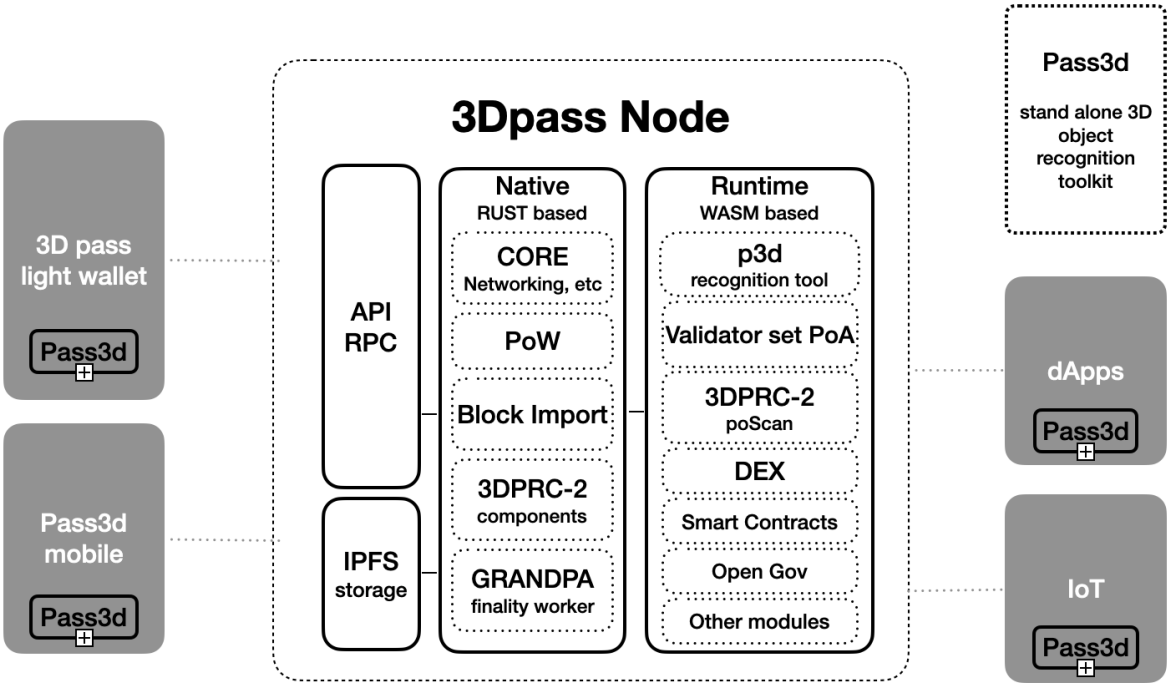
Marketing budget – 110 000 000 P3D (11% of total supply), issued in genesis block. The budget is being spent through the Treasury account controlled by the Open Governance d1ESJKwsk6zP8tBNJABUnf8mtKcqo1U2UVG7iEZ7uytGbWKAL

Early investors budget: 60 000 000 P3D (6% of total supply), issued in genesis block and then turned into the *Contribution budge*t. The budget is transferred to and being spent through the Treasury account d1EjCsWUVnKTG3dysQC2MWDfZKngtiwV2ZLegWRfFMbUR5d6c controlled by the Governance. Initial address the budget was issued with: d1EVSxVDFMMDa79NzV2EvW66PpdD1uLW9aQXjhWZefUfp8Mhf

INTEGRATION

- NODE – the Node leverages Substrate framework and implemented as two-piece design Rust native part and Runtime part (WASM-based), which is upgradable online and allows for multiple useful modules to operate (see more forkless upgrade).
- RPC (remote procedure call) - the capabilities that allow blockchain users to interact with the network. The NODE provides HTTP and WebSocket RPC servers.
- CORE Networking – the libp2p is used as as networking stack for the Nodes to communicate with each other.
- 3Dpass light wallet – desktop users and 3D printing labs integration
- Pass3d mobile – smartphone and tablets users integration

Figure 9



RUNTIME WASM BASED UPGRADABLE MODULES OPERATING

Here is the list of functional modules operating on mainnet, which might be leveraged for dApps creation:

1. **Authorship module** – traces block authors
2. **Difficulty module** – manages difficulty adjustments
3. **Rewards module** – mining PoW rewards implementation
4. **MiningPool module** – decentralized mining pool
5. **Validator set module** – manages PoA GRANDPA** validators rewards and logic
6. **Session module** – ensures the rotation of the PoA GRANDPA** Validators
7. **Grandpa module** – maps GRANDPA** off-chain voter with the Runtime
8. **Offenses module** – handles GRANDPA** equivocation reports (hard fork attempts)
9. **ImOnline module** – ensures that GRANDPA** validators have been online during the session
10. **PoScan module** – handles the user objects verification in 3DPRC-2** standard
11. **PoscanAssets module** – allows for issuing of either 3DPRC-2** the object share tokens or regular fungible assets
12. **AssetsConversion module** – on-chain DEX, decentralized exchange based on Uniswap v2 rules
13. **PoscanPoolAssets module** – DEX Liquidity Pools (LP) tokens management
14. **AtomicSwap module** – allows to atomically swap any asset on-chain/cross-chain for P3D
15. **PoscanAtomicSwap module** – allows to atomically swap any asset on-chain/cross-chain for any on-chain asset on The Ledger of Things, except for P3D
16. **Council module** – manages the Council routine and the election of its members
17. **Technical Committee module** – manages the Technical Committee routine
18. **Democracy module** – handles referendums proposals made by Council
19. **Referenda module** – handles referendums proposals made by community members
20. **ConvictionVoting module** – ensures the network majority vote
21. **phragmenElection module** – ensures the Council vote
22. **Bounties module** – manages Treasury spending proposals
23. **ChildBounties module** – allows to split a bounty into several child-bounties and track them separatelly
24. **Treasury module** – manages funding of spending proposals
25. **White list module** – allows to whitelist any call to be available for a referendum
26. **Smart Contracts module** – smart contracts based on ink language
27. **IPFS file storage** – decentralized file storage
28. **Timestamp module** – allows to timestamp on any on-chain data
29. **System module** – ensures the block size control, etc
30. **Transaction storage module** – transaction pool
31. **MultiSig module** – allows for leveraging multi-signature accounts
32. **Transaction payment module** – manages transaction fee
33. **Vesting module** – allows for vested transfers to operate
34. **Substrate module** – system substrate framework module
35. **Scored pool module** – allows to create a group of members elected
36. **Scheduler module** – allows to schedule any system call in the future
37. **Preimage module** – allows to create and manage system calls
38. **Identity module** – allows to assign on-chain identity to any account

Every module operating provides its own RPC API to interact with.

OPEN GOVERNANCE SYSTEM

3DPass leverages a sophisticated self governance mechanism, which allows it to evolve gracefully overtime at the ultimate behest of its assembled stakeholders. The stated goal is to ensure that the majority of the stake can always command the network. 3DPass decentralized governance system is comprised of three main components:

1. **Council** – An approval-voted, elected executive "government" to manage parameters, admin and spending proposals.
2. **Technical Committee** – A technocratic committee to manage the fork-less online upgrade timelines.
3. **Referenda** – A general voting system for everything else which rewarded long-term stakeholders with increased influence.

In order to do that, the network brings together various novel mechanisms, including an amorphous state-transition function stored on-chain and defined in a platform-neutral intermediate language (i.e. WebAssembly) and several on-chain voting mechanisms such as referenda with adaptive super-majority thresholds and batch approval voting. All changes to the protocol must be agreed upon by stake-weighted referenda.

There is a Treasury pot controlled by the Council, which adds to transparency, especially, when it comes to the budget spendings and grants. The funds held in the Treasury can be spent by making a spending proposal that, if approved by the Council, will enter a waiting period before distribution.

The basement of the governance mechanism was inherited from Substrate, Polkadot and Kusama. Follow the actual documentation to learn about the governance in 3DPass: https://3dpass.org/governance

Democracy

The Ledger of Things has implemented `Democracy` pallet using `Referenda` trait as a voting system. To make any changes to the network, the idea is to compose active token holders and the council together to administrate a network upgrade decision. No matter whether the proposal is proposed by the public (token holders) or the council, it finally will have to go through a referendum to let all holders, weighted by stake, make the decision.

Referenda

Referenda are simple, inclusive, stake-based voting schemes. Each referendum has a specific proposal associated with it that takes the form of a privileged function call in the runtime (that includes the most powerful call: set_code, which can switch out the entire code of the runtime, achieving what would otherwise require a "hard fork").

Referenda are discrete events, have a fixed period where voting happens, and then are tallied and the function call is made if the vote is approved. Referenda are always binary; your only options in voting are "aye", "nay", or abstaining entirely.

Referenda can be started in one of several ways:

- Publicly submitted proposals;
- Proposals submitted by the council, either through a majority or unanimously;
- Proposals submitted as part of the enactment of a prior referendum;

- Emergency proposals submitted by the Technical Committee and approved by the Council

All referenda have an enactment delay associated with them. This is the period between the referendum ending and, assuming the proposal was approved, the changes being enacted. Referenda is considered baked if it is closed and tallied. Again, assuming the proposal was approved, it would be scheduled for enactment. Referenda is considered unbaked if it is pending an outcome, i.e. being voted on.

For the first two ways that a referendum is launched, this is a fixed time of 28 days. For the third type, it can be set as desired. Emergency proposals deal with major problems with the network that need to be "fast-tracked". These will have a shorter enactment time.

Proposing a Referendum:

Public Referenda
Anyone can propose a referendum by depositing the minimum amount of tokens for a certain period (number of blocks). If someone agrees with the proposal, they may deposit the same amount of tokens to support it – this action is called endorsing. The proposal with the highest amount of bonded support will be selected to be a referendum in the next voting cycle.

Note that this may be different from the absolute number of endorsements; for instance, three accounts bonding 20 P3D each would "outweigh" ten accounts bonding a single P3D each.

The bonded tokens will be released once the proposal is tabled (that is, brought to a vote). There can be a maximum of 100 public proposals in the proposal queue.

Council Referenda
Unanimous Council – When all members of the council agree on a proposal, it can be moved to a referendum. This referendum will have a negative turnout bias (that is, the smaller the amount of stake voting, the smaller the amount necessary for it to pass – see Adaptive Quorum Biasing).

Majority Council – When agreement from only a simple majority of council members occurs, the referendum can also be voted upon, but it will be majority-carries (51% wins).
There can only be one active referendum at any given time, except when there is also an emergency referendum in progress.

Voting Timetable
Every 28 days, a new referendum will come up for a vote, assuming there is at least one proposal in one of the queues. There is a queue for Council-approved proposals and a queue for publicly submitted proposals. The referendum to be voted upon alternates between the top proposal in the two queues.

The "top" proposal is determined by the amount of stake bonded behind it. If the given queue whose turn it is to create a referendum that has no proposals (is empty), and proposals are waiting in the other queue, the top proposal in the other queue will become a referendum. Multiple referenda cannot be voted upon in the same period, excluding emergency referenda. An emergency referendum occurring at the same time as a regular referendum (either public- or council-proposed) is the only time that multiple referenda will be able to be voted on at once.

Voting on a referendum

Every 28 days, a new referendum will come up for a vote, assuming there is at least one proposal in one of the queues. There is a queue for Council-approved proposals and a queue for publicly submitted proposals. The referendum to be voted upon alternates between the top proposal in the two queues.

In order to vote, a voter generally must lock their tokens up for at least the enactment delay period beyond the end of the referendum. This is in order to ensure that some minimal economic buy-in to the result is needed and to dissuade vote selling.

Example:

- Peter: Votes No with 10 P3D for a 128 week lock period => 10 x 6 = 60 Votes
- Logan: Votes Yes with 20 P3D for a 4 week lock period => 20 x 1 = 20 Votes
- Kevin: Votes Yes with 15 P3D for 8 week lock period => 15 x 2 = 30 Votes

Even though combined both Logan and Kevin vote with more P3D than Peter, the lock period for both of them is less than Peter, leading to their voting power counting as less.

Tallying

Depending on which entity proposed the proposal and whether all council members voted yes, there are three different scenarios. We can use the following table for reference.

Entity	Metric
Public	Positive Turnout Bias (Super-Majority Approve)
Council (Complete agreement)	Negative Turnout Bias (Super-Majority Against)
Council (Majority agreement) 32	Simple Majority soon

We are going to need, as well, both the following information and one of the formulas listed below to calculate the voting result. For example, let's use the public proposal as an example, so the Super-Majority Approve formula will be applied. There is no strict quorum, but the super-majority required increases with lower turnout.

- approve – the number of aye votes
- against – the number of nay votes

- turnout – the total number of voting tokens (does not include conviction)
- electorate – the total number of tokens issued in the network

Super-Majority Approve
A positive turnout bias, whereby a heavy super-majority of aye votes is required to carry at low turnouts, but as turnout increases towards 100%, it becomes a simple majority-carries as below.

$$\frac{against}{\sqrt{turnout}} < \frac{approve}{\sqrt{electorate}}$$

Super-Majority Against
A negative turnout bias, whereby a heavy super-majority of nay votes is required to reject at low turnouts, but as turnout increases towards 100%, it becomes a simple majority-carries as below.

$$\frac{against}{\sqrt{electorate}} < \frac{approve}{\sqrt{turnout}}$$

Simple-Majority
Majority-carries, a simple comparison of votes; if there are more aye votes than nay, then the proposal is carried, no matter how much stake votes on the proposal.

$$approve > against$$

Example:

- John: 500 P3D
- Peter: 100 P3D
- Lilly: 150 P3D
- JJ: 150 P3D
- Ken: 600 P3D

John: Votes Yes for a 4 week lock period => 500 x 1 = 500 Votes
Peter: Votes Yes for a 4 week lock period => 100 x 1 = 100 Votes
JJ: Votes No for a 16 week lock period => 150 x 3 = 450 Votes

-
- approve = 600
- against = 450
- turnout = 750
- electorate = 1500

Thus:

$$\frac{450}{\sqrt{750}} < \frac{600}{\sqrt{1500}}$$

$$16.432 < 15.492$$

Since the above example is a public referendum, Super-Majority Approve would be used to calculate the result. Super-Majority Approve requires more aye votes to pass the referendum when turnout is low, therefore, based on the above result, the referendum will be rejected. In addition, only the winning voter's tokens are locked. If the voters on the losing side of the referendum believe that the outcome will have negative effects, their tokens are transferrable so they will not be locked into the decision. Moreover, winning proposals are autonomously enacted only after some enactment period.

Voluntary Locking

The Ledger of Things utilizes an idea called Voluntary Locking that allows token holders to increase their voting power by declaring how long they are willing to lock up their tokens, hence, the number of votes for each token holder will be calculated by the following formula:

```
votes = tokens * conviction_multiplier
```

The conviction multiplier increases the vote multiplier by one every time the number of lock periods double.

Lock Periods	Vote Multiplier	Length in Days
0	0.1	0
1	1	28
2	2	56
4	3	112
8	4	224
16	5	448
32	6	896

The maximum number of "doublings" of the lock period is set to 6 (and thus 32 lock periods in total), and one lock period equals 28 days. Only doublings are allowed; you cannot lock for, say, 24 periods and increase your conviction by 5.5.

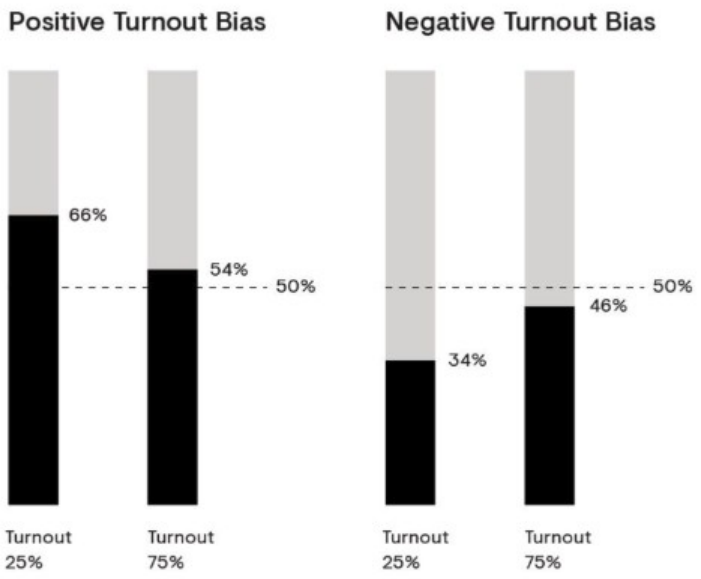
While a token is locked, you can still use it for voting and staking; you are only prohibited from transferring these tokens to another account.

Votes are still "counted" at the same time (at the end of the voting period), no matter for how long the tokens are locked.

Adaptive Quorum Biasing

3Dpass leverages a concept, "Adaptive Quorum Biasing" (first introduced by Polkadot), which functions as a lever that the council can use to alter the effective super-majority required to

make it easier or more difficult for a proposal to pass in the case that there is no clear majority of voting power backing it or against it.



Let's use the image above as an example.

If a publicly submitted referendum only has a 25% turnout, the tally of "aye" votes has to reach 66% for it to pass since we applied Positive Turnout Bias.

In contrast, when it has a 75% turnout, the tally of "aye" votes has to reach 54%, which means that the super-majority required decreases as the turnout increases.

When the council proposes a new proposal through unanimous consent, the referendum would be put to a vote using Negative Turnout Bias. In this case, it is easier to pass this proposal with low turnout and requires a super-majority to reject. As more token holders participate in voting, the bias approaches a plain majority carries.

Referring to the above image, when a referendum only has 25% turnout, the tally of "aye" votes has to reach 34% for it to pass.

In short, when the turnout rate is low, a super-majority is required to reject the proposal, which means a lower threshold of "aye" votes have to be reached, but as turnout increases towards 100%, it becomes a simple majority.

All three tallying mechanisms – majority carries, super-majority approve, and super-majority against – equate to a simple majority-carries system at 100% turnout.

Council
The Council is an on-chain entity embracing several actors, whereas each one represents an on-chain account. Current 3Dpass Council consists of 13 accounts elected by 3Dpass community.

Council responsibilities are:

- Taking control over Treasury spendings
- Proposing sensible referenda
- Cancelling dangerous or malicious referenda
- Electing the Technical Committee
- Approving new recognition algorithms for Proof of Scan
- Dealing with Validators

The referendum proposed by Council
For a referendum to be proposed by the Council, a strict majority of members must be in favor, with no member exercising a veto. Vetoes may be exercised only once by a member for any single proposal; if, after a cool-down period, the proposal is resubmitted, they may not veto it a second time.

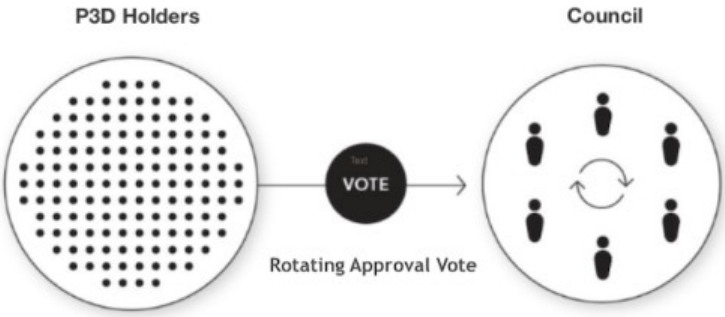
Council motions which pass with a 3/5 (60%) super-majority – but without reaching unanimous support – will move to a public referendum under a neutral, majority-carries voting scheme. In the case that all members of the council vote in favor of a motion, the vote is considered unanimous and becomes a referendum with negative adaptive quorum biasing.

A referendum proposal cancelation
A proposal could be canceled, as long as the Technical Committee would unanimously agree to do so, or if Root origin (e.g. sudo) triggered this functionality.

Additionally, the two-thirds (2/3) of majority of the Council can cancel a referendum. This may function as a last-resort if there is an issue found late in the referendum proposal, such as a bug in the code of the runtime that the proposal would institute.

If the cancellation is controversial enough that the Council cannot exceed the two-thirds of majority, then it will be left to the stakeholders en masse to determine the fate of the proposal.

Council elections



All stakeholders are free to signal their approval of any of the registered candidates. Council elections mechanism operates under Phragmén** voting rules introduced by Edvard Phragmén in the 1890s, which is one of the most common methods leveraging in Nominated Proof of Stake (nPoS). Council terms last for one week.

At the end of each term, Phragmén election algorithm runs and the result will choose the new councillors based on the vote configurations of all voters. The election also chooses a set number of runners up which is currently 20 that will remain in the queue with their votes intact. The algorithm is implemented as Phragmén Election Module.

As opposed to a "first-past-the-post" electoral system, where voters can only vote for a single candidate from a list, a Phragmén election is a more expressive way to include each voters' views. Token holders can treat it as a way to support as many candidates as they want. The election algorithm will find a fair subset of the candidates that most closely matches the expressed indications of the electorate as a whole.

Let's have a look at the example below.
Round 1

Token Holders	**.**	**Candidates**	**.**	**.**	**.**
.	A	B	C	D	E
Peter	.	X	X	X	X
Alice	.	X	.	.	
Bob	.	.	X	X	X
Kelvin	X	.	X	.	..
Total	2	1	3	2	2

The above example shows that candidate C wins the election in round 1, while candidates A, B, D & E keep remaining on the candidates' list for the next round.

Token Holders	.	Candidates	.	.	.
.	A	B	C	D	
Peter	X	X	.	.	
Alice	X	X	.	.	
Bob	X	X	X	X	
Kelvin	X	X	.	.	
Total	4	4	1	1	

For the top-N (say 4 in this example) runners-up, they can remain and their votes persist until the next election. After round 2, even though candidates A & B get the same number of votes in this round, candidate A gets elected because after adding the older unused approvals, it is higher than B.

Prime member
The council, being an instantiation of Substrate's Collective pallet, implements what's called a prime member whose vote acts as the default for other members that fail to vote before the timeout. The prime member is getting chosen in accordance to Borda** count

The purpose of having a prime member of the council is to ensure a quorum, even when several members abstain from a vote. Council members might be tempted to vote a "soft rejection" or a "soft approval" by not voting and letting the others vote. With the existence of a prime member, it forces councillors to be explicit in their votes or have their vote counted for whatever is voted on by the prime.

Technical Committee
The Technical Committee (TC) is composed of the teams that have successfully implemented or specified The Ledger of Things runtime. Teams are added or removed from the TC via a combination of simple majority vote of the Council + Referenda vote.

The purpose of the TC is to safeguard against malicious referenda, implement bug fixes, reverse faulty runtime updates, or add new but battle-tested features. The TC has the power to fast-track proposals by using the Democracy pallet, and is the only origin that is able to trigger the fast-tracking functionality. We can think of the TC as a "unique origin" that cannot generate proposals, but are able to fast track existing proposals.

Fast-tracked referenda are the only type of referenda that can be active alongside another active referendum. Thus, with fast-tracked referenda it is possible to have two active referendums at the same time. Voting on one does not prevent a user from voting on the other.

ECO-SYSTEM
The Ledger of Things eco-system provides a revolutionary toolkit for the tokenization of objects and its transformation into Real World Assets (RWA) presented as either share-tokens or non-fungible assets. Once transformed, there is a variety of options available for the asset owner on how to proceed with their plan, including but not limited to: transfers, trading, ICO, dApp creation/integration, smart-contracts, cross-chain teleport, etc.

- Node – The implementation of The Ledger of Things Node
- 3DPRC-2 – revolutionary tokenization standard proposed by 3Dpass
- DEX – embedded DEX module to trade share-tokens
- Marketplace – to swap non-fungible backed assets
- dApps – smart-contracts module and runtime development
- Cross-chain bridge – to teleport the assets over to the external blockchains
- Stablecoins – on-chain or bridged (over the cross-chain bridge)
- Launchpads – to welcome start-ups into The Ledger of Things eco-system

USE CASE – TOKENIZATION OF LARGE PRECIOUS STONES

Current safe havens for investments include gold and silver. These are not enough for a growing world population and preservation of its assets. Large precious stones lack liquidity, a buy and sell spread could be as much as x5 higher.
3DPass will create a trustworthy environment for the tokenization of large precious stones. It will add to liquidity and will reduce volatility.

3DPass recognition technology is able to identify precious stone shape with an accuracy of up to ~ 3 μm which is sufficient to authenticate the whole stone by several signs: the shape, the weight, the clarity (for diamonds), etc. Even if the stones were cut, it is possible to differentiate from one another because of the manual cutting technology. Modern HD 3D scanners are able to distinguish those differences.

GOODS

Unfortunately many users have suffered from fraud since the internet was invented, especially in marketplaces. There is no widespread technology with the ability to proof whether it relates to a real object for sale or just a picture grabbed from the internet. 3DPass will provide the proof of authenticity for each real life object tokenized on the platform. Ideally It will add to trust and will reduce fraud activities.

For example, a marketplace could develop their own 3D scanning smartphone application with the 3Dpass recognition toolkit integrated. The application could for example capture the users current time, current location, etc. Then, using 3DPass platform, the marketplace could have created their own data base of 3D scans of goods. It would be required for the seller to scan the object for sale until it is published on the market. It would approximately take ~ 3 min to scan by smartphone camera and then to record a short video. The marketplace could potentially recognize whether it is a real object for sale or a fake one by means of comparing Hash IDs from user and from the 3DPass decentralized data base.

REAL ESTATE

Real estate object shape is one of the crucial properties of any object. It might be captured by scanning or 3D modeling and recognized by means of 3DPass processing. Additional properties would be the object's location, post address, passport id etc. This will make the real estate object turn into a 100% identifiable digital asset within the digital space. An additional layer would be a coin representing 1 square meter of the property.

OBJECTS AS PASSWORDS

Since encryption was first applied people have not invented a better solution for the storage of passwords than a piece of paper. This of course carries risks such as loosing or damaging the piece of paper. 3Dpass will provide an alternative way of creating passwords and the secure recovery by means of scanning 3D objects. This will provide an additional layer of safety and will protect the seed-data from deforming as long as a suitable object is chosen.

Check out some of advantages:

- Resistant data carriers might be chosen for a seed such as a piece of rock. This will not be damaged even with several months of exposure to solar radiation, water, electromagnetic radiation, temperature (-100C +500C), etc. A piece of paper or flash memory drive would be completely damaged in those circumstances.
- Mistake proofing of the human aspect such as typos, being unable to read the letters or symbols etc. This is a very common issue for passwords and scanning could eliminate the need for this similar to how easy and consistent it is to scan a QR code.
- A higher effort is required to scan a real 3D object compared to just taking a photo of which provides another layer of security compared to an easier accessible foto from a compromised system

TESTNET REWARDS CONTRIBUTION PROGRAM

As part of the initial testnet before genesis of the mainnet a 1 to 1 exchange ratio was implemented for swapping 3DPt to P3D. The total quantity mined before mainnet genesis is 57 MP3D which was taken out of the marketing budget: https://3dpass.org/testnet-rewards

On the October 25th 2022 theTestnet rewards had been distributed. All 3DPts mined before the block #106390 were transferred to the same accounts on the mainnet. Here we have published the distribution script, which separated the funds mined before and after the block #106390 and performed its distribution. Distribution script: https://github.com/3Dpass/3DP/commit/03de3da9106c06b260359a587982925fd5ba3586

The testnet period has taken into account in the current implementation of the rewards curve for the mainnet: https://github.com/3Dpass/3DP/blob/1a557fbbefc22a7d04793609936951699b7223e0/runtime/src/lib.rs#L431

```
//------------------- rewards
const TESTNET_LAST_BLOCK: u32 = 106390;
const REWARDS_STEP: usize = 243000;
const MAX_REWARS_IDX: usize = 90;
const REWARDS: [u128; MAX_REWARS_IDX] = [
    50_000_000 * MILLICENTS, 41_666_670 * MILLICENTS, 34_722_220 * MILLICENTS,
    28_935_190 * MILLICENTS, 24_112_650 * MILLICENTS, 20_093_880 * MILLICENTS,
    16_744_900 * MILLICENTS, 13_954_080 * MILLICENTS, 11_628_400 * MILLICENTS,
    09_690_330 * MILLICENTS, 08_075_280 * MILLICENTS, 06_729_400 * MILLICENTS,
    05_607_830 * MILLICENTS, 04_673_190 * MILLICENTS, 03_894_330 * MILLICENTS,
    03_245_270 * MILLICENTS, 02_704_390 * MILLICENTS, 02_253_660 * MILLICENTS,
    01_878_050 * MILLICENTS, 01_565_040 * MILLICENTS, 01_304_200 * MILLICENTS,
    01_086_840 * MILLICENTS, 00_905_700 * MILLICENTS, 00_754_750 * MILLICENTS,
    00_628_960 * MILLICENTS, 00_524_130 * MILLICENTS, 00_436_770 * MILLICENTS,
    00_363_980 * MILLICENTS, 00_303_320 * MILLICENTS, 00_252_760 * MILLICENTS,
    00_210_640 * MILLICENTS, 00_175_530 * MILLICENTS, 00_146_280 * MILLICENTS,
    00_121_900 * MILLICENTS, 00_101_580 * MILLICENTS, 00_084_650 * MILLICENTS,
    00_070_540 * MILLICENTS, 00_058_780 * MILLICENTS, 00_048_990 * MILLICENTS,
```

```
    00_040_820 * MILLICENTS, 00_034_020 * MILLICENTS, 00_028_350 * MILLICENTS,
    00_023_620 * MILLICENTS, 00_019_690 * MILLICENTS, 00_016_410 * MILLICENTS,
    00_013_670 * MILLICENTS, 00_011_390 * MILLICENTS, 00_009_490 * MILLICENTS,
    00_007_910 * MILLICENTS, 00_006_590 * MILLICENTS, 00_005_490 * MILLICENTS,
    00_004_580 * MILLICENTS, 00_003_820 * MILLICENTS, 00_003_180 * MILLICENTS,
    00_002_650 * MILLICENTS, 00_002_210 * MILLICENTS, 00_001_840 * MILLICENTS,
    00_001_530 * MILLICENTS, 00_001_280 * MILLICENTS, 00_001_060 * MILLICENTS,
    00_000_890 * MILLICENTS, 00_000_740 * MILLICENTS, 00_000_620 * MILLICENTS,
    00_000_510 * MILLICENTS, 00_000_430 * MILLICENTS, 00_000_360 * MILLICENTS,
    00_000_300 * MILLICENTS, 00_000_250 * MILLICENTS, 00_000_210 * MILLICENTS,
    00_000_170 * MILLICENTS, 00_000_140 * MILLICENTS, 00_000_120 * MILLICENTS,
    00_000_100 * MILLICENTS, 00_000_080 * MILLICENTS, 00_000_070 * MILLICENTS,
    00_000_060 * MILLICENTS, 00_000_050 * MILLICENTS, 00_000_040 * MILLICENTS,
    00_000_030 * MILLICENTS, 00_000_030 * MILLICENTS, 00_000_020 * MILLICENTS,
    00_000_020 * MILLICENTS, 00_000_020 * MILLICENTS, 00_000_010 * MILLICENTS,
    00_000_010 * MILLICENTS, 00_000_010 * MILLICENTS, 00_000_010 * MILLICENTS,
    00_000_010 * MILLICENTS, 00_000_010 * MILLICENTS, 00_000_000 * MILLICENTS,
];
```

GRANT PROGRAM

There is a grant program established by the Council, which is to incentivize community contribution efforts directed on development of the The Ledger of Things eco-system. https://3dpass.org/coin#distribution-contribution

RESPONSIBILITY DISCLAIMER

3DPass is an open source free p2p software distributed according to MIT License** which is non commercial and community supported. Any business or person might use it for own commercial purposes though. In that case 3DPass's software might get a part of commercial products but those businesses will not have anything to do with 3DPass in general. 3DPass disclaims responsibility for any commercial projects based on it. 3DP is a platform's unit of account that's necessary only to provide the ability of tokenizing 3D objects and of decentralization that keeps digital assets secure.

CONCLUSIONS

The 3DPass project proposed a fully decentralized platform, which is capable of solving the global issue of digital transformation of 3D objects (digital and real ones). Any 3D solid object might be transformed into a digital asset by means of 3DPass recognition technology. Each asset all over the platform might be verified by either its owner or independent majority of validators providing the object authenticity check. Each asset might represent a single non-fungible token or a digital currency backed by the object and the currency unit could be a quantum (1 gram, 1 kilogram, 1 meter, etc). Each asset might be exchanged directly to the other asset or digital currency, be used within smart-contracts, be used by IoT smart-devices. As a bonus, users get the ability to leverage 3D objects as recoverable passwords.

REFERENCE (**)

1. pass3d recognition toolkit https://github.com/3Dpass/pass3d
2. SHA-2 https://en.wikipedia.org/wiki/SHA-2
3. RMSD https://en.wikipedia.org/wiki/Root-mean-square_deviation

4. splprep https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.splprep.html
5. Substrate https://substrate.io/
6. obj format https://en.wikipedia.org/wiki/Wavefront_.obj_file
7. stl format https://en.wikipedia.org/wiki/STL_(file_format)
8. ink https://paritytech.github.io/ink-docs/
9. Rust https://www.rust-lang.org/
10. eDSL https://wiki.haskell.org/Embedded_domain_specific_language
11. WebAssembly https://webassembly.org/
12. EVM https://ethereum.org/en/developers/docs/evm/
13. IPFS https://ipfs.io/
14. GRANDPA https://polkadot.network/blog/polkadot-consensus-part-2-grandpa/
15. Substrate Smart Contracts Toolkits https://docs.substrate.io/v3/runtime/smart-contracts/
16. MIT License https://en.wikipedia.org/wiki/MIT_License
17. Blake2b hash https://en.wikipedia.org/wiki/BLAKE_(hash_function)#BLAKE2b_algorithm
18. RandomX hashing: https://github.com/tevador/RandomX
19. 3DPRC-2 standard: https://github.com/3Dpass/whitepaper/blob/main/3DPRC-2.md
20. PoScan module: https://github.com/3Dpass/3DP/tree/test/pallets/poscan
21. PoScanAssets module: https://github.com/3Dpass/3DP/tree/main/pallets/poscan-assets
22. PoScan API: https://github.com/3Dpass/3DP/wiki/3DPRC%E2%80%902-PoScan-API
23. Phragmén's voting rules: https://en.wikipedia.org/wiki/Phragmen%27s_voting_rules
24. Borda count: https://en.wikipedia.org/wiki/Borda_count

3DPass – The Ledger of Things
3dpass.org June 1, 2024
Author: PaulS
Author of Grid2d recognition algorithm: Michael Co
Co-authorship on 3D object shape validation: 神炫
Acknowledgment on PoW task component: MPM
Acknowledgment: A Mo