3DPass

The Ledger of unique 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.

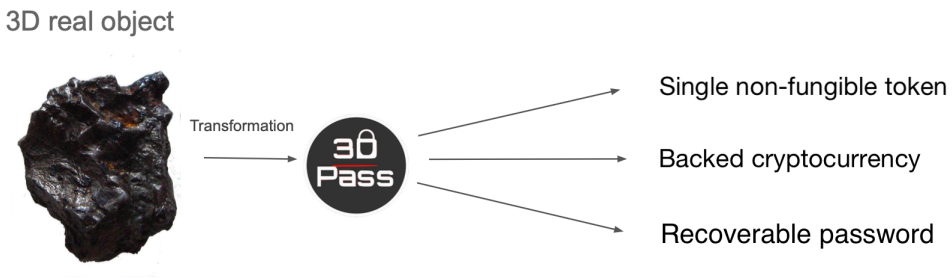
From the initial perspective, the problem statement looks very challenging to solve but there are some opportunities upon further inspection. This includes face recognition technology, many ML projects, NFT, etc. Why is such an emphasis placed shape recognition? It's evident that the shape is the critical 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 objects and jewellery 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. Single non-fungible token;
2. Cryptocurrency, backed by the object (fungible token);
3. Password, recoverable by means of scanning the object.

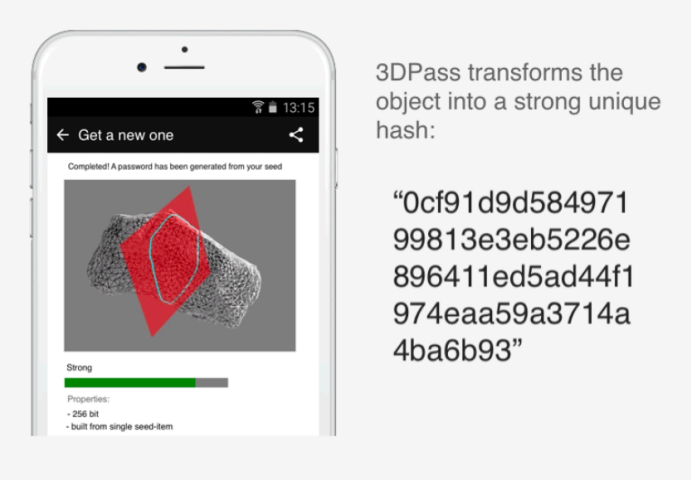


In the foundation of 3DPass lies real world object recognition technology including various research-based algorithms created by either team or community members. The algorithms have been especially adapted to work within the blockchain environment which does not provide any feedback to compare objects to one another while the processing is going on. 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 serves as a significant part of Proof of Scan p2p network consensus used in 3DPass blockchain decentralized platform. This represents 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 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 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 cryptocurrency 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 lots of different potential ways to process 3D object shape in order to solve the same digital transformation issue. And it’s highly likely that some of them would solve a particular problem better than the others. Several different processing algorithms have been developed and tested by the 3DP team (a description of most stable one is presented down bellow). As a community driven project members will add to this with improvements and will create new algorithms in purpose of solving additional challenges with a wider market opportunity.

Candidate-algorithms:

3D Objects

2D Objects (Drawings)

2D Fingerprints

Face Recognition

Voice

Melodies

Radio Signal

Barcodes

QR Codes

This list 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 differ 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

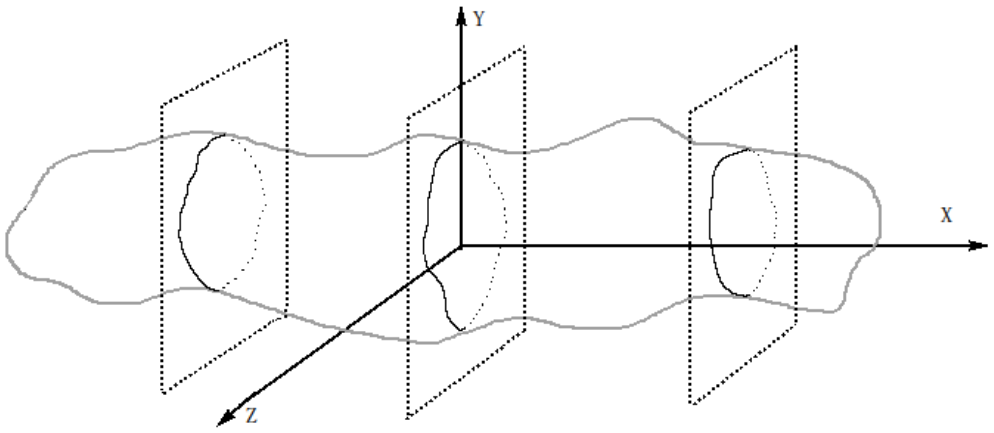
The object orientation has to meet the following conditions:

- 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 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



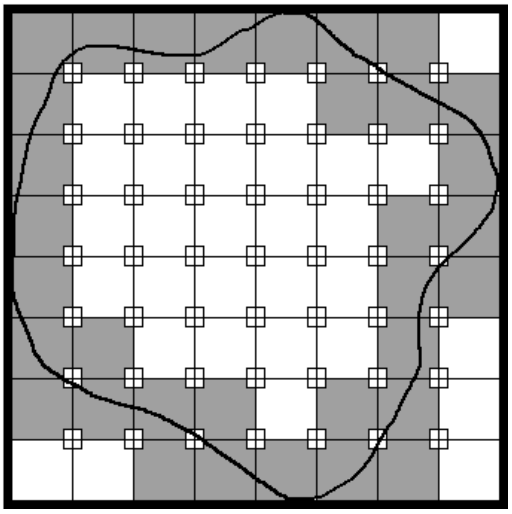
3. For each contour the following steps will be performed:

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)

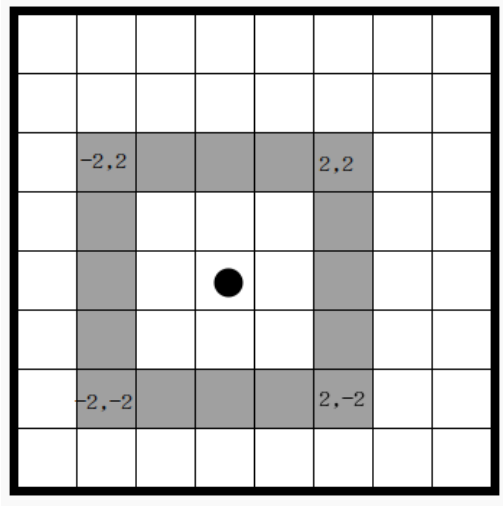
3.3. Find the set E of cells containing our contour. Due to noise neighboring cells the contour passes through are also included. Cells should be added to E if the contour is close to grid vertices less than 10% of cell size, by example (*parameter #3 of the algorithm*).

Drawing 2: Set of cells allowed



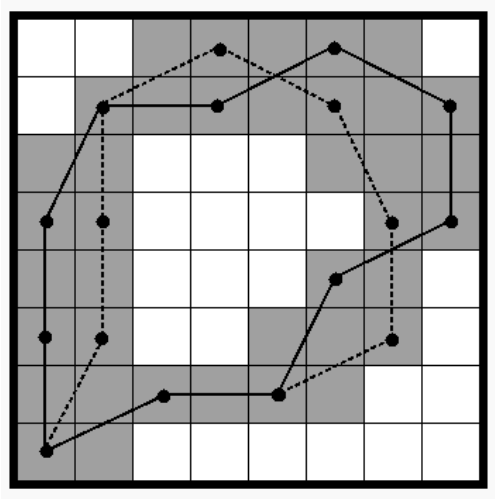
3.4. Generate all the possible polygons with vertices belonging to set E. Neighboring vertices have to be at a distance L, for example L=2 (*parameter #4 of the algorithm*). Vertices must not be close to each other at a distance less than L.

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 countour, 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 concatination 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
```

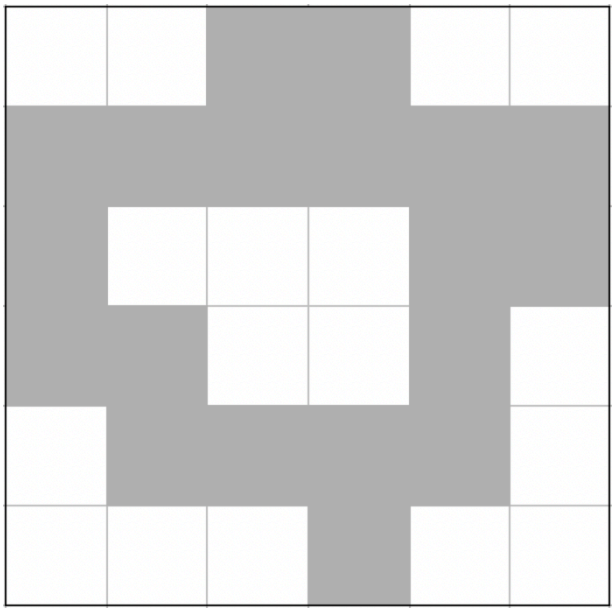
These are two key parameters that need to adjust in order to create the best possible Hash ID depending on 3D scans quality.

Grid scale parameter -g:

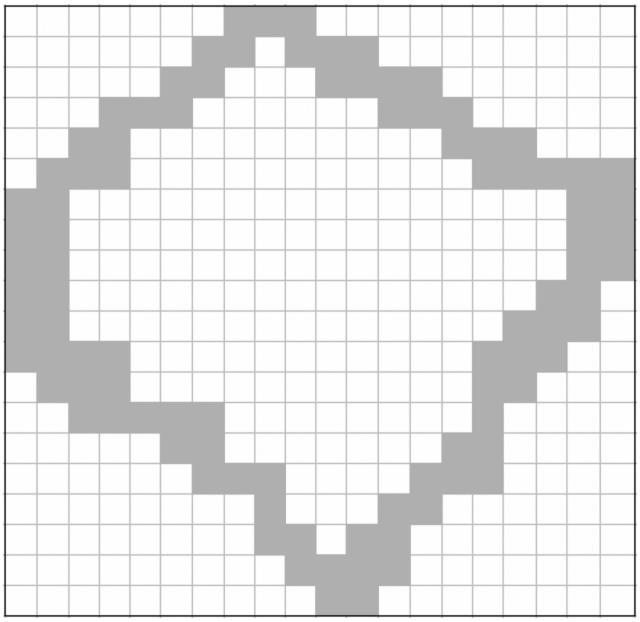
Grid size (-g) is the parameter which is used to adjust the recognition algorithm to the particular 3D scan quality. The higher the scan quality, the higher the number of cells that can be set up for the processing. According to the Grid2d algorithm, by means of increasing number of cells this leads 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:

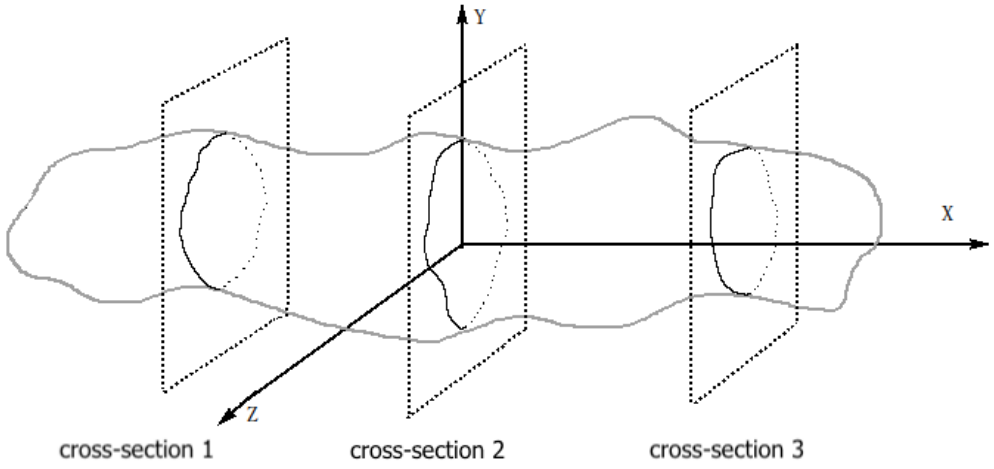


Considerations should be made to set the definition parameter *(-g)* up to the lowest quality of 3D scans that is expect to apply in the future. If the *(-g)* value is set up to be appropriate for HD scanners (*-g=20* or higher) but the scanner becomes unavailable, then the recognition success will not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

Some 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.

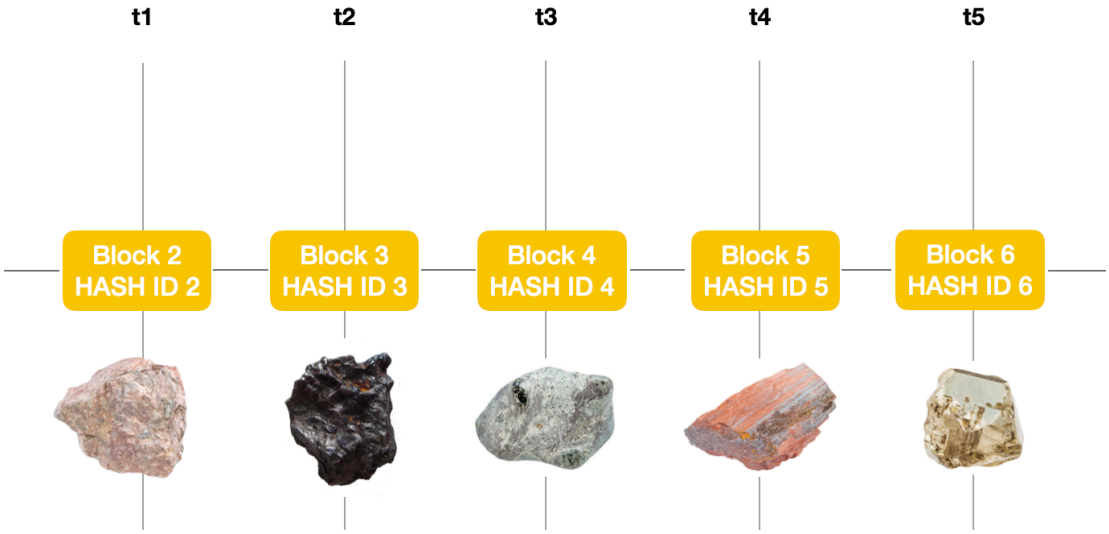
ADDITIONAL PROPERTIES OF 3D OBJECTS

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 3DP network directly.

3DPASS P2P NETWORK

3DPass decentralized network is available to provide the next level of quality to the entire blockchain market, so that anyone could obtain proof that there is only one original object that is published as a digital asset. At a minimum this will make it transparent when dealing with copies. It is called the network of unique things, which can be utilized within smart-contracts and dApps. The network nodes will prevent the duplication of assets.



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. IPFS** is used as a decentralized storage for 3D objects. Integration is provided by the Substrate** framework.

VERIFICATION MECHANISM

3DPass network represents one layer 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 produced by Grid2d which allows to verify the algorithm was really used. This represents a hash taken from a randomly chosen intermediate calculation result of the object processed. Anyone can check it up and verify this while the processing of the object is going on.

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:

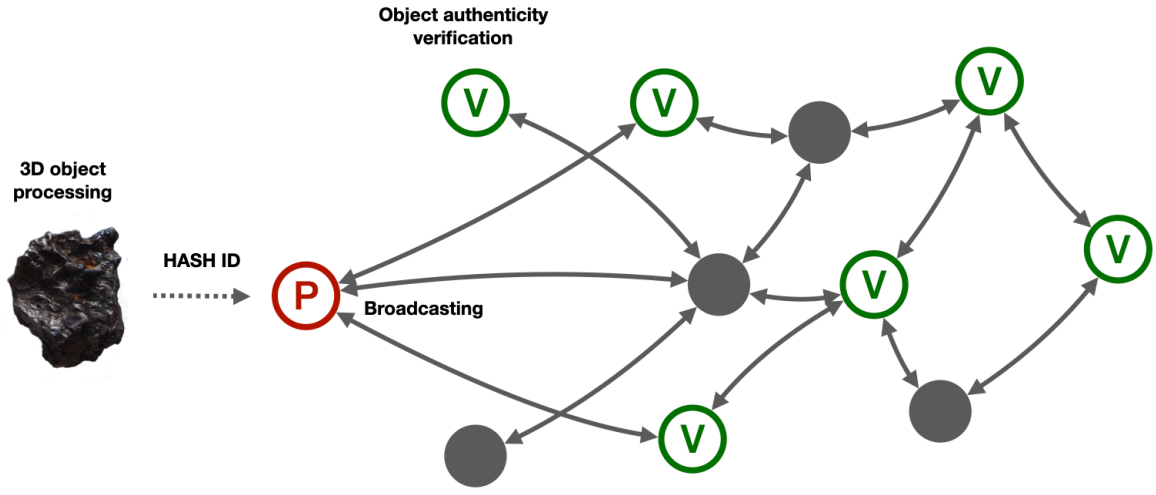
1. Whether or not the Hash ID is corresponding to the shape of the object submitted. Which means the 3D object authenticity check. If the Hash IDs do not match to one another, then the object is considered fake and will be rejected;

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. However, In order to get rewarded from the network there are a few conditions on top (see NEW BLOCK).

Figure 1



P - Publisher (node which has published the object). **V** - Validators (nodes which have verified the object authenticity)

NEW BLOCK

Each new block is going to be built on top of the parent block of the best chain (longest chain rule is actually applied). It is also sealed to the parent block hash with SHA256 cryptographic hash-function.

Block header consists of the following data:

- 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...
```

- The object's Hash ID (Top10 hashes list coming from Grid2d output result, -g 8 -s 66). For more details read 3D OBJECT HASH ID CREATION;

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

- Block hash - the main block identifier. It represents SHA256 hash sealed to the parent block hash with the cryptographic hash function. And besides, the very top hash of the Top10 is used as an additional nonce. Let's say, if the parent block hash is

```
"090ae6b23e2192fa4c2fb40cddad6e8537e2b437c49ff9fb227cf32c4e4085fc"
```

and the very top hash is

"72592f8f6ea67c60ca7d9c7683256c3636a30be464952eb82996bff52ca4415",

then the new block hash would be the hash taken from the seed like this:

"090ae6b23e2192fa4c2fb40cddad6e8537e2b437c49ff9fb227cf32c4e4085fc7
2592f8f6ea67c60ca7d9c7683256c3636a30be464952eb82996bff52ca4415d"

- Difficulty number the block had been proposed with (e.x. 10)

Considering the network needs computing power for object processing. This requires a set of rules and reward with new blocks. The rules in principal are quite similar to bitcoin, however, P3D mining has an object recognition algorithm involved.

Here is the rules:

1. New block target time: 1 block per 60 seconds
2. By default, each 243000 blocks there is a difficulty step up involving "+0" requirement to be added at the front of the new block hash. So, in order to create a new block after "climbing" over the step and getting rewarded, someone has to pick up a unique 3D object, the new block hash would include one more zero at the front like in the examples below:

Block hash example, from genesis block to the block number 243000:
"090ae6b23e2192fa4c2fb40cddad6e8537e2b437c49ff9fb227cf32c4e4085fc";

from the block number 243001 to 486000:
"008cfda2b4811bf2ff1fe3ab92b38e64fc134d98c3dc8764eb8641a477b77a47",

from the block number 3402001 to 3645000:
"00000000000000f2ff1fe3abr6bt8e64fc134d98c3dc8764eb8641aff7b7sa4f", etc.

3. There is a 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).

Note that neither of the rules above touches the objects submitted not in the reason of getting mining rewards. So, miners have to abide to the rules but the other users do not. They might submit any objects they want, of course, being charged P3D for the authenticity validation service provided by miners while they are constructing a new block. So, they have all the authority at the moment to validate either the object just mined and the users' commercial objects.

Once a new block is produced, it is awaiting confirmation to be chosen as a parent block for the next new block. The block reaches maturity after 100 confirmations (100 blocks has been built on top) and gets finalized according to GRANDPA** protocol.

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 different 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 require a higher effort. It would be more efficient to generate

this 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 difficult to find a new unique 3D model for the next block.

BLOCK PRODUCTION MECHANISM

Our block production begins with 3D objects queue. 3D models are being sent to the input of PoScan via RPC in the following format:

```
{
    "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
.....
            ]
}
```

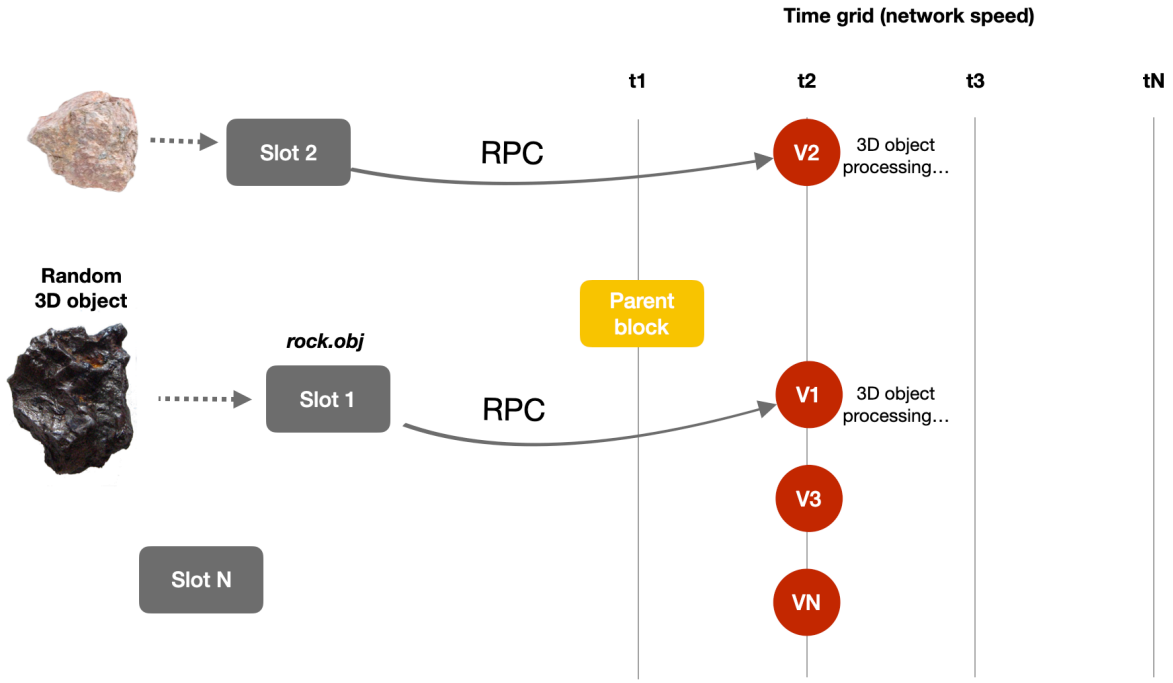
Where one of the parameters is the content of 3D model's file in .obj format, “\n” is being added at the end of each line:

```
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's side, which checks the queue. If there is a 3D model found, the handler would run its shape processing. Miners join the process, doing the same simultaneously. All the nodes, including miners will have chosen the best chain at the time.

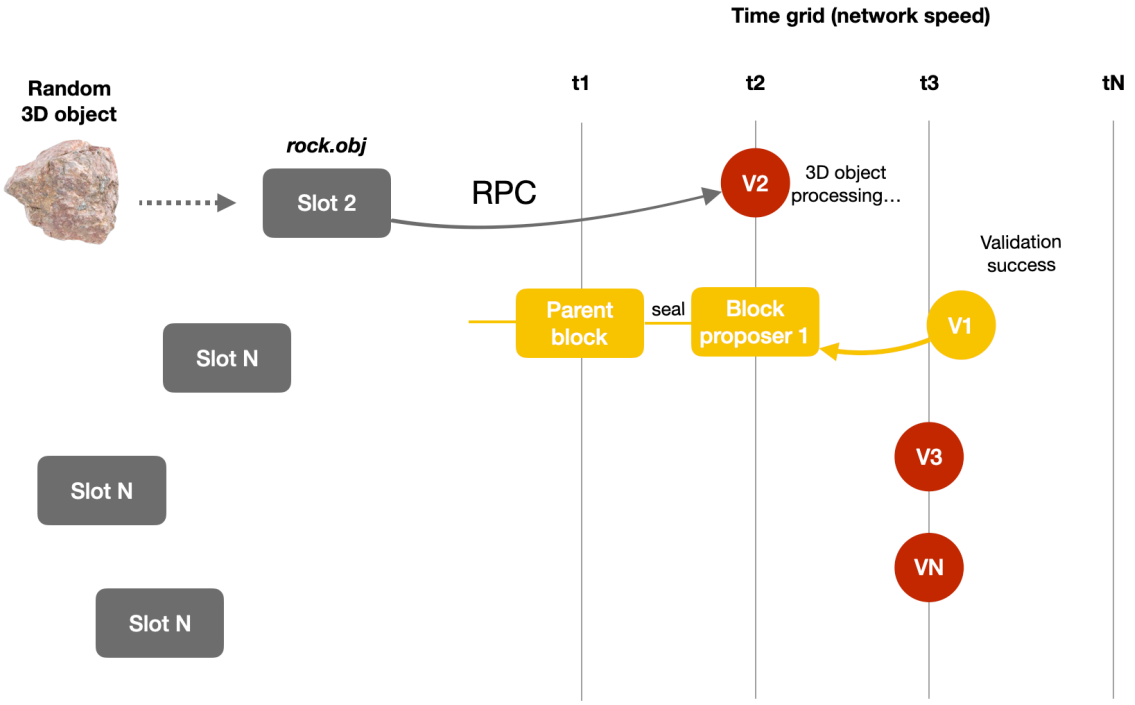
In order to create a new block they will use the parent block (the top block of the best chain). (*Figure 2*)

Figure 2



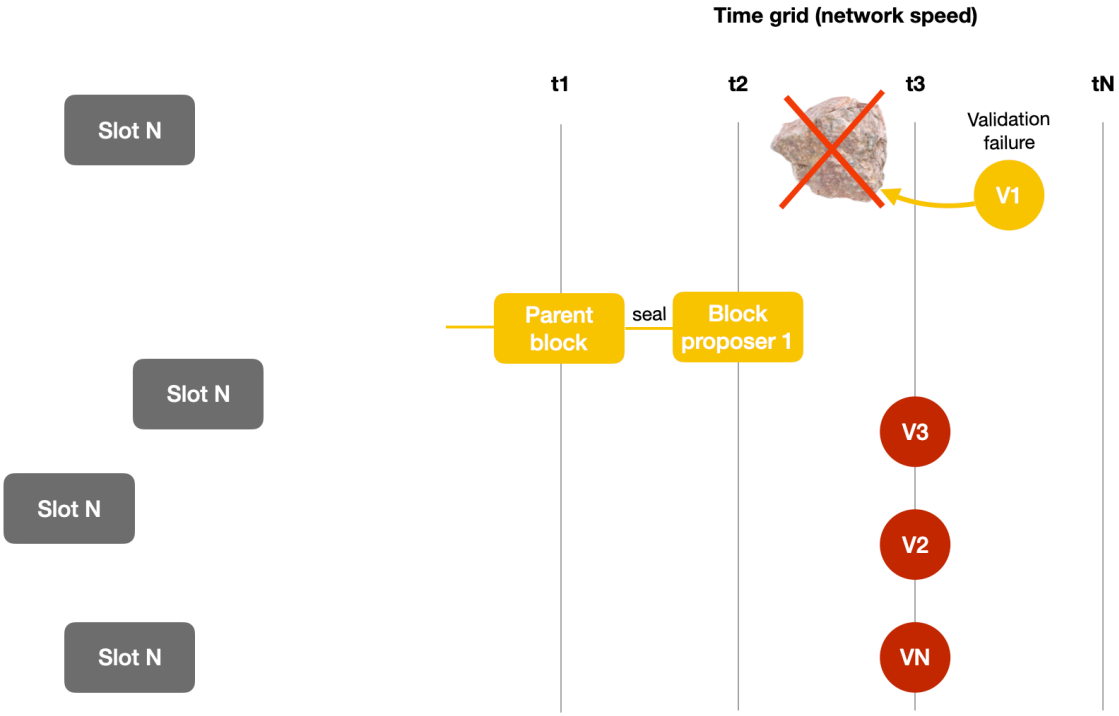
After the HASH ID is calculated, miner tries to create a new block hash. In order to do so, it seals the very top hash of the HASH ID to the parent block hash already chosen at the time. If the resulting hash corresponds to the new block mining rules, it becomes a new block hash, and the new block proposer will be created and broadcasted over the network (*Figure 3*).

Figure 3



If the resulting hash doesn't correspond to the rules, the object will be rejected. Subsequently the node will keep checking the queue for a new one to appear. The block verified participates the Tie Break competition among the miners trying to construct the block on top of the longest chain they believe to be the best one. This logic is quite similar to conventional PoW. (*Figure 4*).

Figure 4

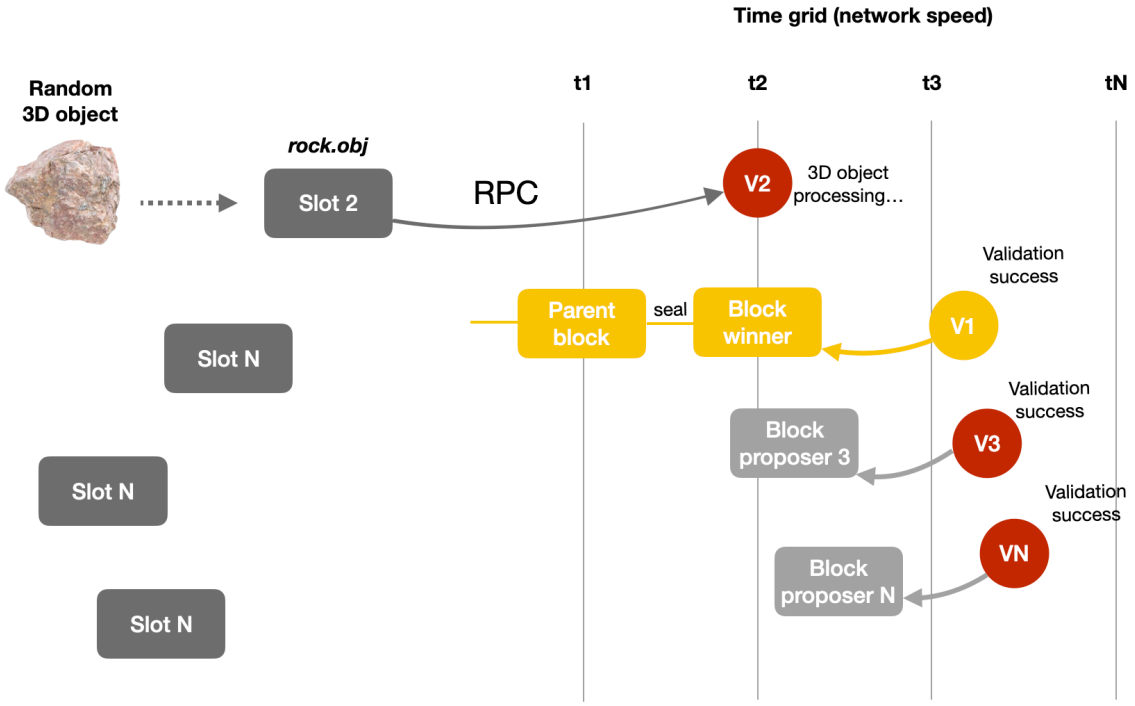


TIE BREAK COMPETITION

The block verified participates the Tie Break competition among the other block Authors (miners), trying to construct the block on top of the longest chain they believe to be the best one. The block proposed earlier wins. This logic is quite similar to conventional PoW.

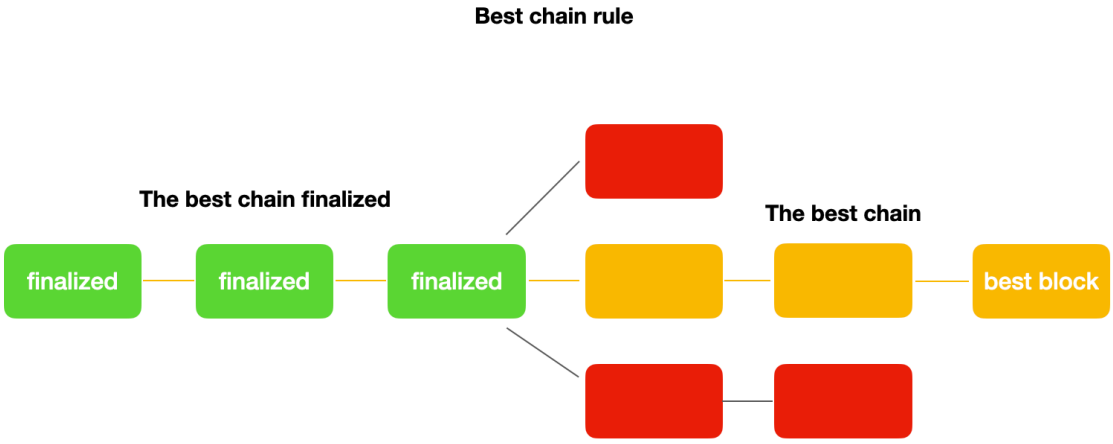
Once created, the new block proposer consists of the following aspects: block header, transactions from transaction pool and some metadata. All the data is signed with the block Author's signature. In order to be able to prove authorship, the miner must have generated the account, its private and public keys and a miner's key (address). The block proposer is being imported by the network Nodes and there is a validation procedure on import that causes the blockchain reconstruction. The best chain has to be chosen. (*Figure 5*)

Figure 5



The protocol contains a chain 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 other nodes. The best chain is selected and must be accepted to be the best chain by all the nodes on the network. The best block should not be further than 3 blocks away from the last block finalized (*Figure 6*). Because of this rule, 51% attack is still possible, however, the attacker must have taken over at least 2/3 of GRANDPA votes in addition (see more Finality gadget: GRANDPA), which makes it much more difficult to arrange.

Figure 6



FINALITY GADGET: GRANDPA

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.

The block Author produced a correct block takes the authority to vote for the best chain finality, as long as they have put 100 000 - 400 000 P3D in a collateral. Thus, Grandpa Authority set consists of the most reliable block authors producing some blocks in the 100 - 8000 most recent blocks looking back, and 2/3 votes of which is needed to make the final decision. After the voting is complete there would be no chance to reselect the best chain.

There is a session of 100 blocks length within which the current validator set is always stable. It can not be changed with new in/outcomers until the session is expired. In case a validator is heading off the validator set, in order to come back the selection threshold is required to pass again.

Selection threshold:

- 100 000 P3D locked + 1 block mined in 100 recent blocks back
- 200 000 P3D locked + 1 block mined in 2000 recent blocks back
- 300 000 P3D locked + 1 block mined in 4000 recent blocks back
- 400 000 P3D locked + 1 block mined in 8000 recent blocks back

Penalties:

There is a list of punishments for Validators that behave against the GRANDPA protocol rules:

- PreVote Equivocation (voting for two different chains simultaneously): 40 000 P3D and get excluded from the validator set
- Not being online/available: 20 000 P3D and get excluded from the validator set
- Not being able to vote for any reason (e.x. Firewall): 20 000 P3D and get excluded from the validator set

Given the fact, that GRANDPA gadget in 3DPass 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.

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:

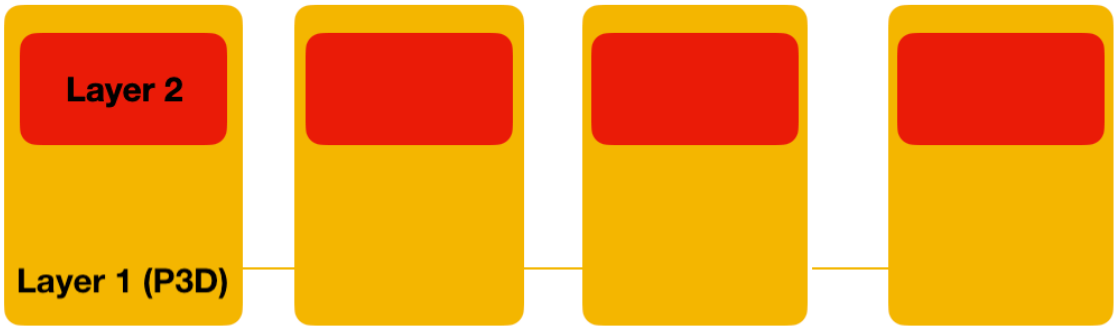
- Issuing NFT single token backed by 3D object
- Private backed currency running where the unit would be a quantum of a 3D object (1 gram, 1 kilogram, 1 meter, etc).
- Decentralized apps development, marketplaces, etc

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

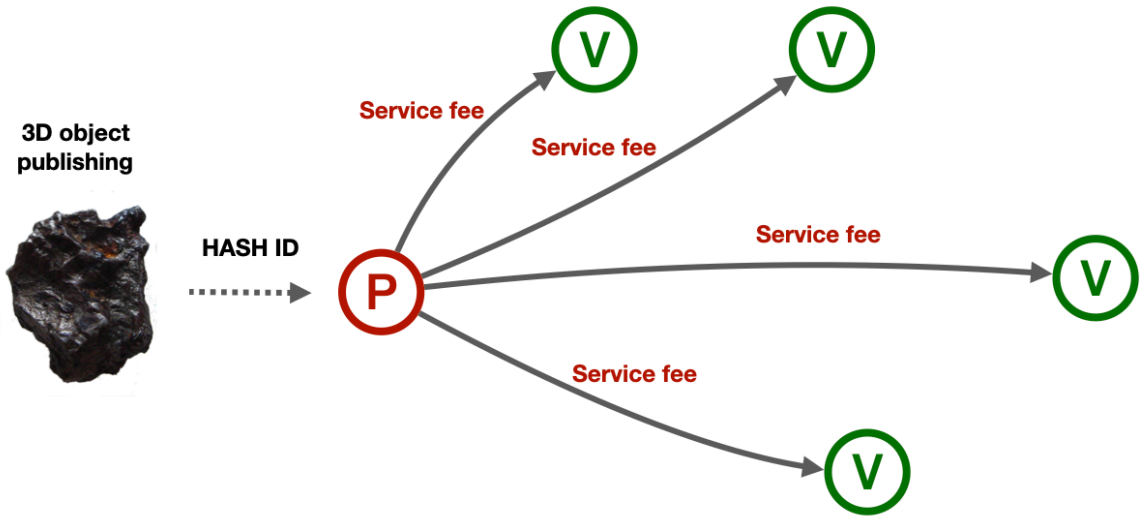
The native 3DPass currency unit is P3D which is required especially for internal needs such as 3D object processing encouragement. The smallest indivisible unit of account in 3DPass is called Crumb. 3DPass coins (P3D) are equal to 1e12 Crumb.

- 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 differ from one block to another.

Figure 8



COMMISSIONS AND REWARDS

Mining rewards

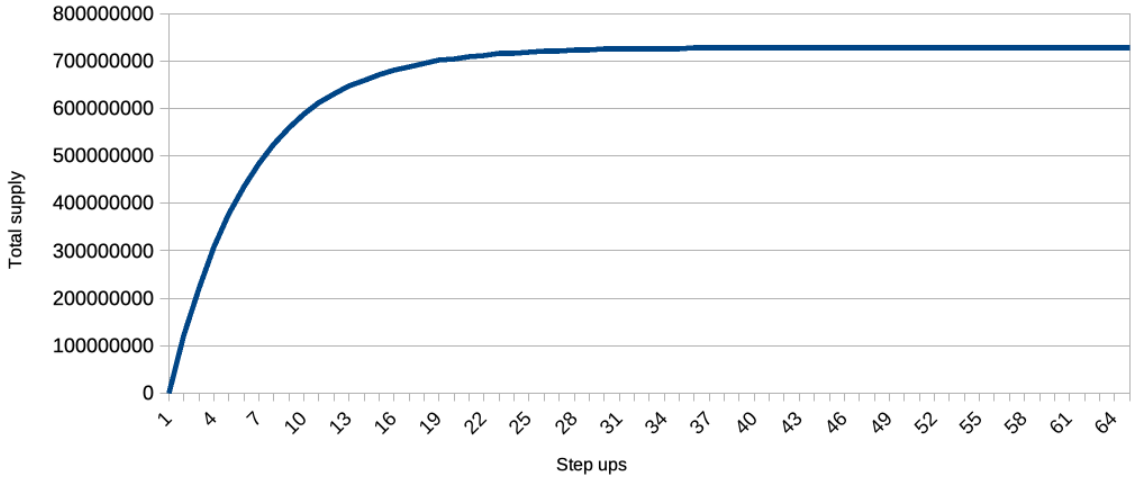
Miners put efforts to find a 3D object shape corresponding to the rules that govern whether or not a new block would be rewarded by the network. There are two rules for the rewards calculation:

1. Starting from the block going after the genesis, mining rewards amount is established as 500 P3D per block,
2. Every 243000 blocks, simultaneously to the difficulty step up occasion, the amount of mining reward will be reduced by dividing by 1.2.

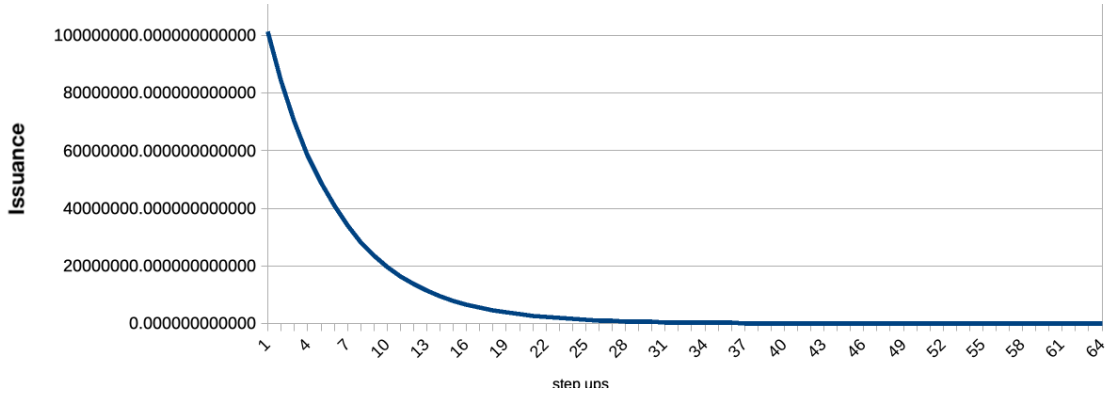
Given the fact that the network speed is 1 block/60 sec, thus each difficulty step of 243000 blocs would approximately take about 169 days.

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

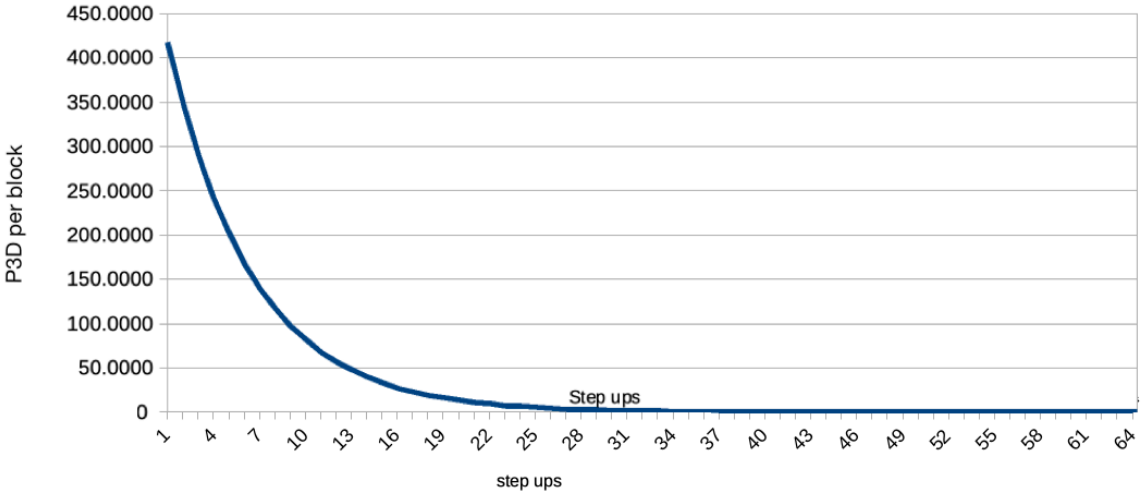
P3D rewards curve



P3D issuance



P3D per block



3D objects validation 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.

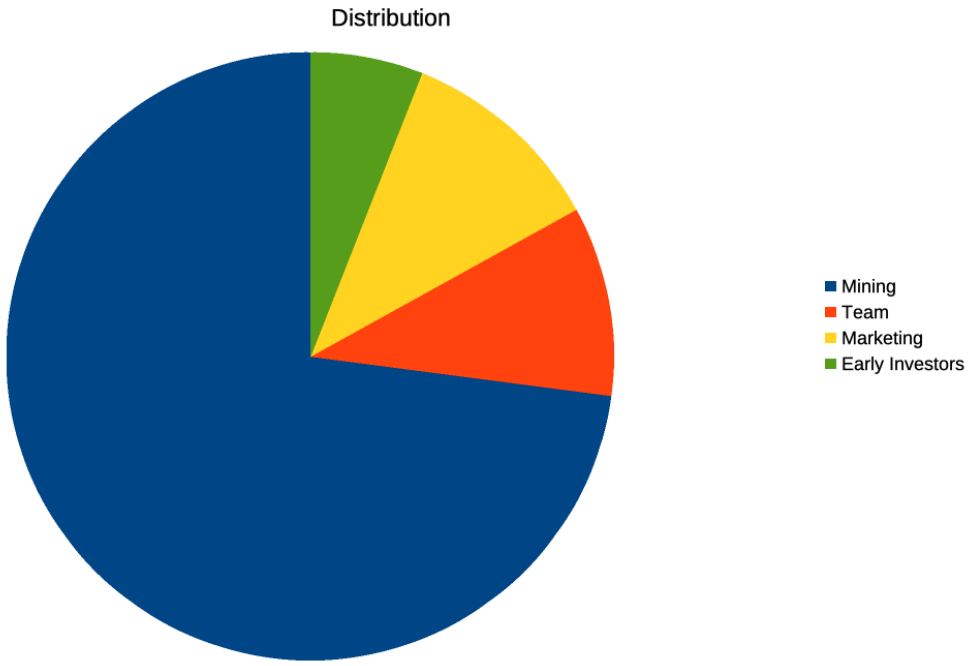
Transaction fee

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)

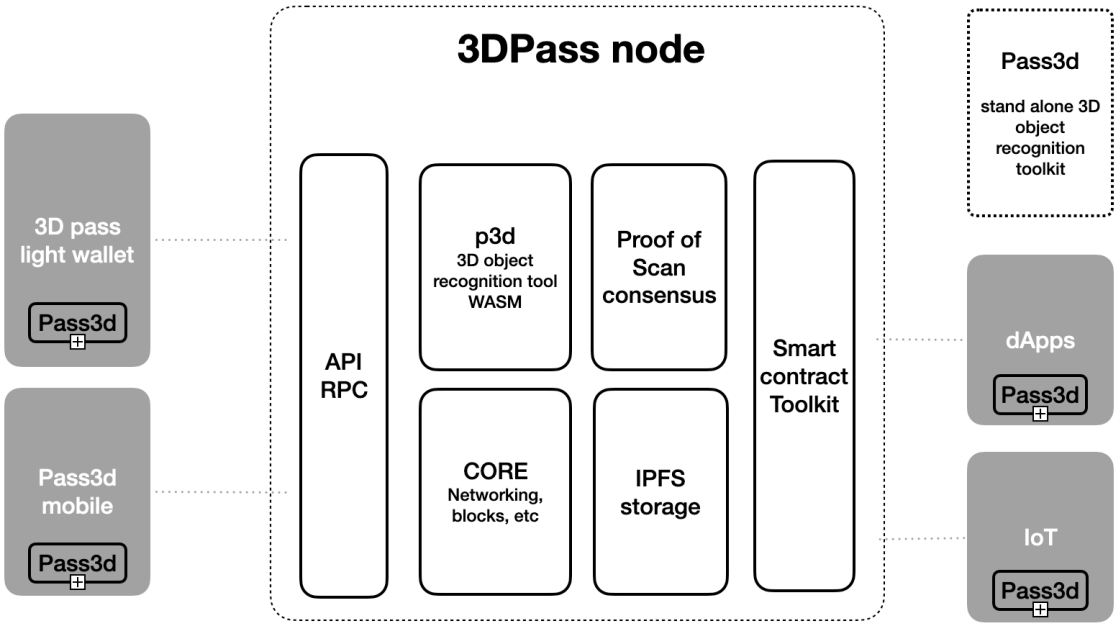


ECO-SYSTEM AND INTEGRATION

Given the fact, that 3DPass is totally open source, meaning anyone can add to development, there is an eco-system scheme (*Figure 9*) representing general functional elements:

- **3DPass NODE** - wallets, dApps, smart-contracts, IoT devices integration using API and RPC. 3DPass node, based on Substrate, has all the network features implemented as pallets: CORE, Pass3d (objects recognition toolkit), IPFS decentralized storage, Smart contracts toolkit, RPC and API.
- **Pass3d and p3d recognition toolkit** - recognition algorithm integration. This toolkit consists of stable recognition algorithms used for digital transformation of objects and 3DPass network consensus. Since, the recognition technology is what starts the digital transformation process of 3D object. The result of processing is HASH ID, it implies every application, integrated into 3DPass eco-system, to have Pass3d toolkit implemented. 3DPass node app and wallets are not an exception either.
- **Proof of Scan consensus** - the logic, using 3D objects recognition, that allows network participants to agree on the state of the blockchain
- **3DPass light web3 wallet** - desktop users and 3D printing labs integration
- **Pass3d mobile client** - smartphone and tablets users integration
- **Smart contracts toolkit** - Substrate based smart contract tools using ink**, a Rust-based embedded domain specific language (eDSL**) for writing WebAssembly** smart contracts. As well, it allows unmodified EVM** code to be executed in the 3DPass blockchain. Some special Substrate** features are designed to closely emulate the functionality of executing contracts on the Ethereum mainnet within the 3DPass network.
- **IPFS storage** - decentralized storage for assets.
- **RPC (remote procedure call)** - the capabilities that allow blockchain users to interact with the network. The NODE provides HTTP and WebSocket RPC servers.

Figure 9



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 one from 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

The testnet will continue to be in operation after the Mainnet launch to provide a platform for experimenting and testing out new features before roll out to the Mainnet. 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 57million which was taken out of the marketing budget.

Rewards

All the activities, covered by the program, will be paid only in P3D Coins. No other currencies accepted. Estimation rate: 1 P3D = 0.00495 USD for all activities requiring resources to get involved. This rate was calculated assuming the amount of the 3DPass founders team's funds of around 500 000 USD invested so far. Team share is 10.1% = 101 000 000 P3D. The founders believe in principles of fair partnership and propose to contribute on the equal conditions. There is also a rewards program for solving issues listed on GitHub in P3D coins.

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

3DPass - The Ledger of unique things
3dpass.org November 1, 2022

Author: PaulS

Acknowledgment: A Mo