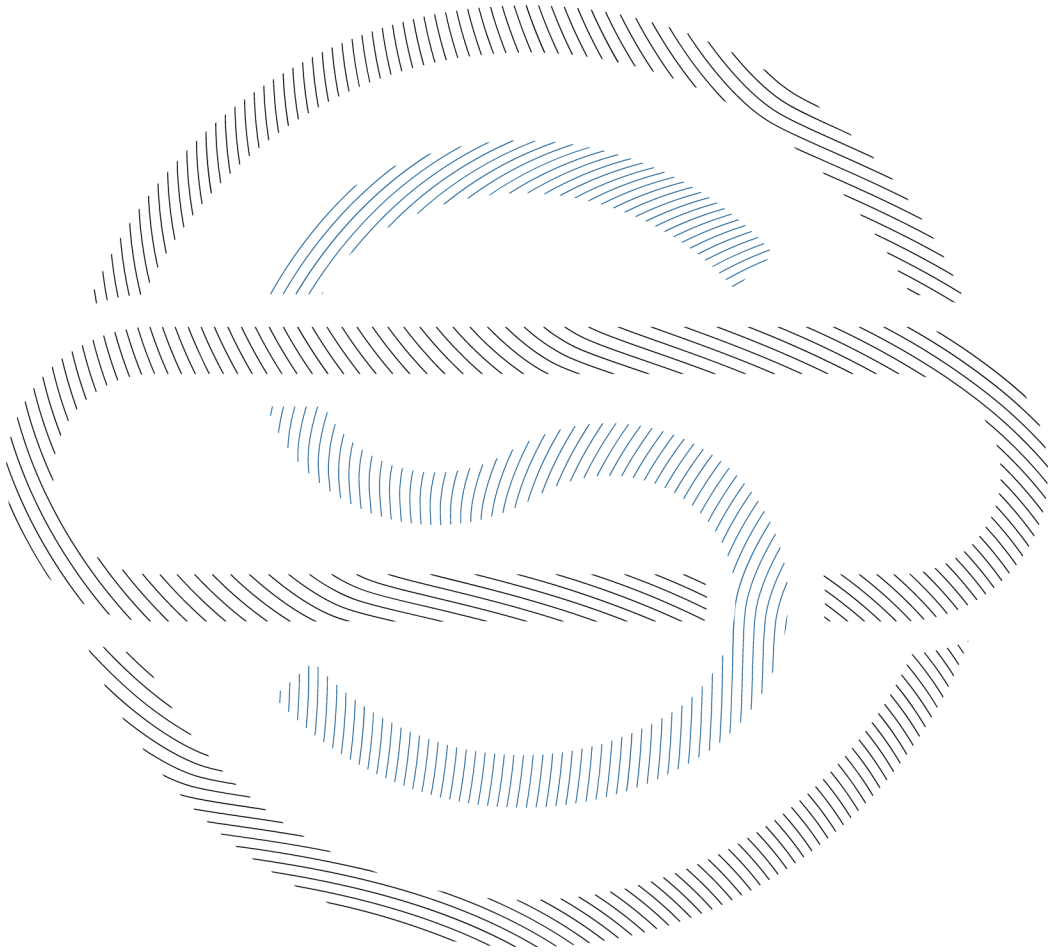
Timestamp Certificate

This timestamp was created with *Bitcoin*



originstamp

Timestamp

Apr-03-2020 00:42:54 UTC

Hash:

9f5099adf04521fe59bcb66745ffb91dbabc5be2720cfd2b1470cf79eed99bdc

Transaction:

575076d71fab626526d69000b4ab8f335fd76d66d355104a314b4580f1d69f44

Private Key:

9dcff52c69312e74b0e1f3a9e06fdc104e30a2df2d89bd565b9c9d8f54f81806

Note:

thibaultlangloisberthelot.com POST N°1 : VISIT ME IN VIRTUAL REALITY (VR). Article written on 03/30/2020 by Thibault LB.

originstamp

Timestamp Certificate

Proof

The proof is necessary for the reproducibility of your document.


```
<?xml version="1.0" encoding="UTF-8"?>
<node type="key" value="9dcff52c69312e74b0e1f3a9e06fdc104e30a2df2d89bd565b9c9d8f54f81806">
 <left type="mesh" value="fc67e37164d1af56649673d55dac8292fb00bede76a4afb5b5e7fab84a9b1795"/>
 <right type="mesh" value="6ecbb72d868fa8525300b5456123812a6d6138578f5a0ecf74d5fe59ffb0acb2">
  <left type="mesh" value="a0de2f748006728240fc098ef4616a5f562ef8c390ed2c23332729727dbd3b80">
   <left type="mesh" value="1581af708a8d3f6e4b5373a8bef5d48a1e64b83d751ff12e956f0493b9d2c27b">
    <left type="mesh" value="a4f21fad16c8ab4bd3331d41d533979aa14271eff64f8bfe8eca50553297c6fa">
     <left type="mesh" value="9eb9e9de3ea7233d811217267fb0205316837dd6525b627249501d77d5b58d02">
      <left type="mesh" value="bc432bebd62c69f1db07dd66fb624720c4d8f6380a57d020bfdd2d2bf352ecfc"/>
      <right type="mesh" value="a35580aa8e6fbf5ae06a7e647c69d4197ade2d9751b9d487c51c833828f111f8">
       <left type="mesh" value="48b2e7b5543853e08b027c37c88931e0146469d84056d56769a6b6b5063059e7"/>
       <right type="mesh" value="f75bab2905270451591c91ebf1a897b6406e31987054abcfccd28161e2f612db">
        <left type="mesh" value="f8196d52daf68105eda81ce43600386fa9db549b24d18825f578085c298efbfe"/>
        <right type="mesh" value="e2f8ac5fdf357949cd7334a3a8d374597ca19e8507e7a8745dba74812903f25a">
         <left type="mesh" value="d7a8fbfcaa2bd34be8df864719486323ecc48b9e6ace906100e08d5791775640"/>
         <right type="mesh" value="591b46f86a54b666c34817b52baac955dead1b5095e776245cc87cc22194ebbd">
          <left type="mesh" value="0ee0b293cc7e68616e789c99778681e4c092a883527c8d6f06fba88c83a6b591">
           <left type="mesh" value="df42b0db9b1d078ef0f6f2c813be48987f86730f81ef6511a8338c6e711da90b">
            <left type="mesh" value="d8a88f96d7a249c05fc223abe82a28c70b1e16181af4f3b6b83e8c34acc6d79a">
             <left type="mesh" value="d5d86fdd462a9642644ad2e54453cf48989b998dfcbb8a52e49521f5cb256f53"/>
             <right type="mesh" value="959b39883fd02caa0743c440a868f6b761eb21c6f5f431d3d47937e714f55808">
              <left type="mesh" value="9f4ba1f27fda058cb603dafbc7d17d2f44a5bab7a87aa39b6305f0f71ae6ec50"/>
              <right type="hash" value="9f5099adf04521fe59bcb66745ffb91dbabc5be2720cfd2b1470cf79eed99bdc"/>
             </right>
            </left>
            <right type="mesh" value="3bc59ca8926feb66f2dfe269b46a5e53821c6b915e4eb743cb54275ff9a6a320"/>
           </left>
           <right type="mesh" value="3028399134bb95a14dcd5b2698358c1499df540f6e5c3c85d74a4c7c8b8ccde5"/>
          </left>
          <right type="mesh" value="e214c0c8666e747a3ffb9c2993703c128fd7fe53e9bfd65ae71ff0e084712f09"/>
         </right>
        </right>
       </right>
      </left>
      <right type="mesh" value="5802ed73e596080dacd089c35d3c421ea7527106cd89585e758711d0d82e033a"/>
     </left>
     <right type="mesh" value="d9b17e89e3d5f933381ad95c6790f0f2b5f91293c28113a7ba461d9b3d57f7e1"/>
    </left>
    <right type="mesh" value="b3acb2275af53dc38f4e305a5e70009cd4802c99e875cd51d6ebae9893a4a000"/>
   </left>
   <right type="mesh" value="9d1a08234c51dc53e87a68af26a296df8c2959806e0be490898732b1f7db0527"/>
  </right>
</node>
```

originstamp

Timestamp Certificate

Verification

OriginStamp is a timestamp service that uses various blockchains like the Bitcoin Blockchain to create tamper-proof timestamps for your data. Instead of backing up your data, OriginStamp embeds a cryptographic fingerprint in the blockchain. It is de facto impossible to deduce the content of your data from your fingerprint. Therefore, the data remains in your company and is not publicly accessible. All you need to do is send this fingerprint to OriginStamp via the interface. The integration of the RESTful API is very simple and convenient and allows all data to be easily tagged with a tamper-proof timestamp. This document shows the procedure and gives instructions for verifying a timestamp created with OriginStamp.

1. Determine the SHA-256 of your original file

There are numerous programs and libraries to calculate the SHA-256 of a file, such as MD5FILE.COM. Simply drag and drop or select your file, to retrieve the SHA-256 of your file.

2. Validate your proof

First, it must be verified that the hash of the original is part of the evidence. In order to check this, the proof can be opened with a conventional editor and its content can be searched for the hash. If the hash cannot be found, either the file was manipulated or the wrong evidence was selected.

3. Determine the private key

The Merkle tree is a tree structure, that allows to organize the seed more efficient than a plain-text seed file. The tree is built from the bottom to the top and follows a defined schema. The value of a node is determined by the aggregated hash of its children.

Left child = a8eb9f308b08397df77443697de4959c156fd4c68c489995163285dbd3eedaef

Right child = ab95adaee8eb02219d556082a7f4fb70d19b8000097848112eb85b1d2fca8f67

Node = SHA-256(a8eb9f308b08397df77443697de4959c156fd4c68c489995163285dbd3eedaefab95adaee8eb02219d556082a7f4fb70d19b8000097848112eb85b1d2fca8f67) = 47e47c96302eeba62ed443dd0c89b3411bbddd2c1ff6bdfb1f833fa11e060b85

This step is performed for all levels of the tree until the hash of the root has been calculated. If the hash of the root is identical as proof, the calculation was successful and the root hash is verified. The top hash corresponds to the private key we embedded in the blockchain through a transaction. For a more detailed explanation of the Merkle tree, we want to refer to Wikipedia.

4. Determine the Bitcoin address

Having determined the private key in the previous step, we can use this private key for a new Bitcoin address. The detailed steps to calculate the uncompressed Bitcoin address can be found here. Lets assume the private key is *4eac8a92f8846ea801669b5834aa733e5345cc5e90875152ea6b9f38c724701e*. The calculated Bitcoin Address is *1CV9tyNSdzrKFC2gtpx3Y5GU9rPWb81R4T*.

5. Check the transactions

Check the transactions. By using any blockexplorer for Bitcoin, you can search for the previously calculated Bitcoin address: *1CV9tyNSdzrKFC2gtpx3Y5GU9rPWb81R4T*. The first transaction for this address testifies to the proof of existence. The timestamp of the file corresponds to the block time of the first transaction.