OpenGem

NFT AUDIT

Watch Certificate™

Asset storage & persistence assessment

March 2nd, 2023

Audit performed for:
Watch Certificate | en.watchcertificate.com | @wtchcertificate
by:
Franck Dupont | franck@opengem.com | @franckdpt
Damien Dupont | damien@opengem.com | @dam_dpt
opengem.com

Table of contents

The purpose of Audit

The purpose of the audit is to ascertain whether NFT assets are correctly **decentralized**, **persistent**, and **easy to retrieve**. These three criteria are at the core of GEM grading.

We built Gem scores to make platforms more **transparent,** especially as regards what users can mint and how they own what they mint.

We audit how the NFT is minted, how NFT assets are stored, and evaluate its **alteration** by third parties without the holder's authorization.

We sincerely believe that **ownership** is the key to ensuring the democratization of NFT.

Our specially built multiple score schemes regarding platforms are in the public domain. They are published and can be accessed on our medium here.

To learn more about our vision, kindly read our whitepaper here.

Overview

Project

Collection name: **Watch Certificate**
NFT name: **Watch Certificate**
Website: **watchcertificate.com**
Twitter: **@wtchcertificate**
Description: "***Collection de certificats délivrés par Watch Certificate™.***"

Blockchain: **Polygon**
Smart contract: **0x14255bcDF743C884f614B2b60212F81AC744819a**
Explorer:
https://polygonscan.com/address/0x14255bcDF743C884f614B2b60212F81AC744819a

Audit

Delivery Date: **March 9th, 2023**
Method of audit: **Manual review**
Consultants: **2**

Audit public page:
https://opengem.com/audits/score/watch-certificate/0x14255bcDF743C884f614B2b60212F81AC744819a

Executive Summary

Watch Certificate reaches **100/100 Gem score**.
The collection uses excellent storage methods to ensure data **persistence** and **decentralization.**

All metadata are **on-chain** and **retrievable** as long as the Polygon blockchain exists.

Findings

Id	Title	Recommendations	Severity
1	On-chain text data	-	None
2	Provenance proof accessiblity	-	None
3	On-chain stored images	-	None
4	Documentation	-	None

1. Context

1.1 Utility Token

Due to the strong growth in demand for watches, both new and second hand, certification and tracking of watches has become essential. Watch Certificate™, by adopting NFT technology, offers irrefutable and unalterable proof of their provenance and history.

The Watch Certificate™ NFT allows the watch's history to be tracked in a transparent and accessible manner, ensuring that the ownership and value of the item is protected. Customers can rest assured. Their investment is secure and authenticated by a respected and recognized company in the field.

1.2 Scoring System

According to the context, we defined different levels of data importance to measure the ownership weight.
The scoring system is based on these levels of metadata importance:

First-tier:
- The watch brand [text]
- The watch model [text]
- The watch reference [text]
- The watch year [text]

Second-tier:
- The watch face condition [photo]
- The watch back condition [photo]

Third-tier:
- The watch history & owner [text]

For each tier, we will audit :
- **The storage of the asset**
  Is it decentralized ? How permanent is the storage?

- **The persistence of the on-chain asset reference**
  Is anybody or anything able to alter the asset reference on-chain?
- **The presence & persistence of an on-chain provenance proof**
  As owner, am I able to prove the ownership by hashing my asset locally, even if the company doesn't exist anymore?

We will also audit **how easy is to retrieve** of all these information.

2. The Watch Information (T1)

The main information of the watch are :

- The watch brand [text]
- The watch model [text]
- The watch reference [text]
- The watch year [text]

All these information defined the unicity of the physical item. It's crucial that this is immutable.

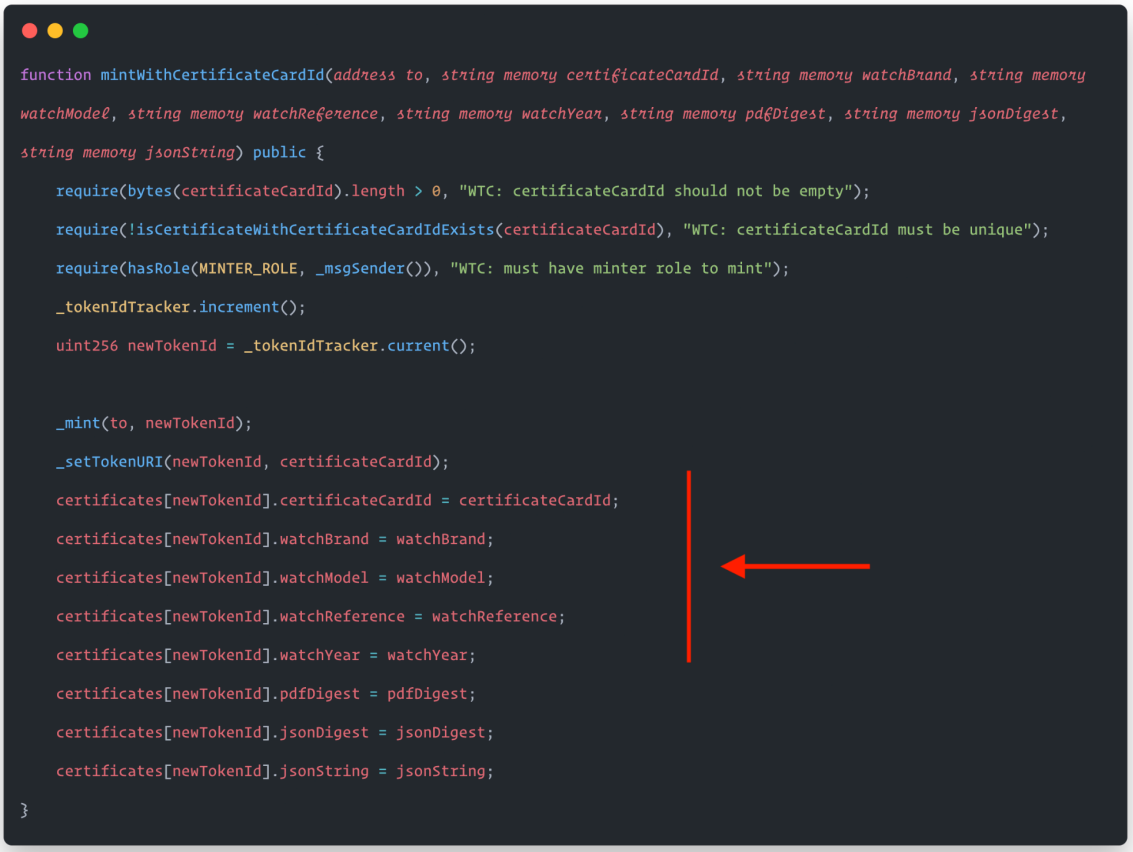
Here is an example :

- Rolex
- Datejust
- 16220
- 2000s

2.1 Storage & Persistence

All these main informations are written on-chain, in the NFT. It is readable publicly by anyone.

The storage process is made during the mint of the token via the function named ***mintWithCertificateCardId****()*.

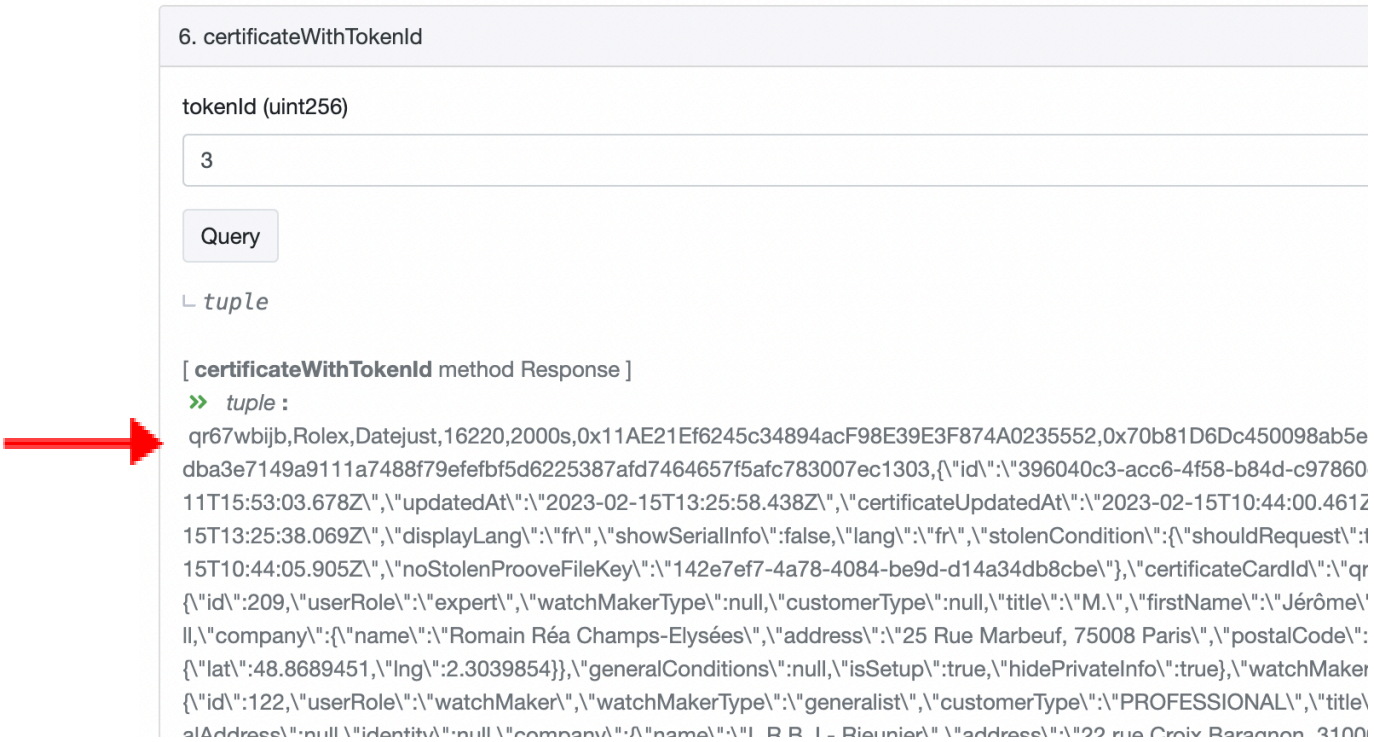


```
function mintWithCertificateCardId(address to, string memory certificateCardId, string memory watchBrand, string memory
watchModel, string memory watchReference, string memory watchYear, string memory pdfDigest, string memory jsonDigest,
string memory jsonString) public {
    require(bytes(certificateCardId).length > 0, "WTC: certificateCardId should not be empty");
    require(!isCertificateWithCertificateCardIdExists(certificateCardId), "WTC: certificateCardId must be unique");
    require(hasRole(MINTER_ROLE, _msgSender()), "WTC: must have minter role to mint");
    _tokenIdTracker.increment();
    uint256 newTokenId = _tokenIdTracker.current();

    _mint(to, newTokenId);
    _setTokenURI(newTokenId, certificateCardId);
    certificates[newTokenId].certificateCardId = certificateCardId;
    certificates[newTokenId].watchBrand = watchBrand;
    certificates[newTokenId].watchModel = watchModel;
    certificates[newTokenId].watchReference = watchReference;
    certificates[newTokenId].watchYear = watchYear;
    certificates[newTokenId].pdfDigest = pdfDigest;
    certificates[newTokenId].jsonDigest = jsonDigest;
    certificates[newTokenId].jsonString = jsonString;
}
```

When minting the token, an unique token ID is generated to which all the T1 information are attached on-chain.

```
function certificateWithTokenId(uint256 tokenId) public view returns (Certificate memory) {
    require(_exists(tokenId), "WTC: tokenId must exist to get certificate");

    return certificates[tokenId];
}
```

These informations are accessible via a public function named ***certificateWithTokenId()*** callable from Polygon Scan for example :



We noticed an ***update()*** function in the contract. After discussion with the team, this is to maintain the watch data up-to-date. But the only data that need to be updated does not includ these T1 information. It keeps immutable the unicity of the token. The update process concerns some T2 and T3 data that need to be up-to-date (condition of the watch, name of the new owner, etc.)

```
function update(uint256 tokenId, string memory pdfDigest, string memory jsonDigest, string memory jsonString) public {
    require(_exists(tokenId), "WTC: tokenId must exist to update certificate");
    //solhint-disable-next-line max-line-length
    require(_isApproved(_msgSender(), tokenId) || hasRole(MINTER_ROLE, _msgSender()),
"WTC: must have minter role or be approved to update");
    certificates[tokenId].pdfDigest = pdfDigest;
    certificates[tokenId].jsonDigest = jsonDigest;
    certificates[tokenId].jsonString = jsonString;
    emit CertificateUpdate(tokenId, certificates[tokenId].certificateCardId, certificates[tokenId].pdfDigest,
certificates[tokenId].jsonDigest, certificates[tokenId].jsonString);
}
```

So, no one can rewrite or delete the T1 information, as long as Polygon exists.

2.2 Provenance Proof

As soon as the original data is written on-chain, a provenance proof is not required. The on-chain T1 data itself is a proof of ownership.

2.3 Conclusion ✅

We have here the best storage methods to ensure ownership.

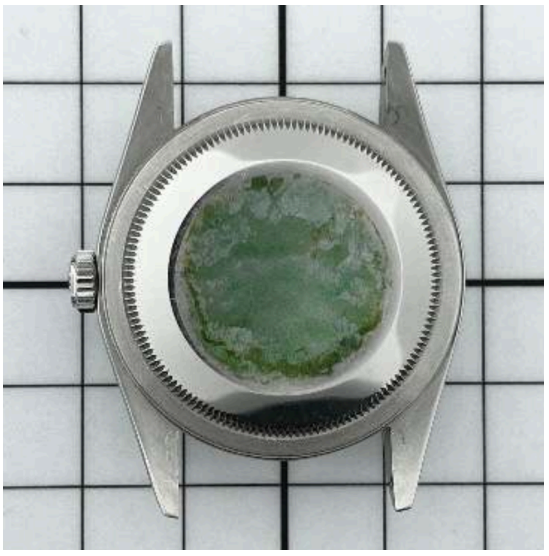
- Anyone can verify and read the watch main information on-chain at anytime.
- Absolutely no one can alter or delete these information, as soon as Polygon blockchain exists.
- Even if Watch Certificate company doesn't exist anymore, all owners will be able to retrieve all these information, as soon as Polygon blockchain exists.

3. The Watch Condition (T2)

To keep track of the watch, Watch Certificate provides a follow-up condition of the watches directly in the on-chain certificate. The idea is to make photos of the watch publicly accessible throughout its entire life.

These data are [image] files.
Here are some examples of watch pictures taken for certification:





3.1 Storage & Persistence

Storing an image in a blockchain can be tricky, expensive, and may not always be a good idea. However, as we are working with watch certifications,

tracking the condition of the watch is crucial. Therefore, it is important to make all watch photos publicly available on-chain in a decentralized way.

There are several practices to achieve this, and based on our recommendations, Watch Certificate has chosen to store the image using the Event Emitting method. This method involves putting the base64 of an image file into an event that is emitted in the header of the next validated block. It drastically reduces the cost of the transaction of an on-chain file upload. Then, the idea is to store the transaction hash that emitted the event inside the NFT.

The upload is done by the event ***ImageData()*** emitted in the ***setFaceImageData() & setBackImageData()*** functions :

```
function setFaceImageData(uint256 tokenId, string calldata imageName, uint256 imagePartIndex, string calldata imageData) public {
    require(_isApproved(_msgSender(), tokenId) || hasRole(MINTER_ROLE, _msgSender()),
"WTC: must have minter role or be approved to set face image data");
    require(_exists(tokenId), "WTC: tokenId must exist to set face image data");
    ImageStorage faceImage = new ImageStorage();

    if (imagePartIndex == 0) {
        // INFO: reinitialize image
        certificates[tokenId].faceImage = faceImage;
    }
    emit ImageData(tokenId, certificates[tokenId].certificateCardId, imageName, imagePartIndex, imageData);
}
```

Once event emitted & image uploaded, the corresponding transaction is set in the NFT via the ***setFaceImageDataTxHash() & setBackImageDataTxHash().***

```
function setFaceImageDataTxHash(uint256 tokenId, string memory txHash) public {
    require(_isApproved(_msgSender(), tokenId) || hasRole(MINTER_ROLE, _msgSender()),
"WTC: must have minter role or be approved to set face image data tx hash");
    require(_exists(tokenId), "WTC: tokenId must exist to set face image data tx hash");
    certificates[tokenId].faceImage.setTxHashImageData(txHash);
}
```

Now any user can consult the logs of the transaction via the ***getFaceImageDataTxHash()*** & ***getBackImageDataTxHash()*** via Polygon Scan, and then retrieve the image file.

```
function getFaceImageDataTxHash(uint256 tokenId, uint256 imagePartIndex) public view returns (string memory) {
    require(_exists(tokenId), "WTC: tokenId must exist to get face image data tx hash");
    Certificate memory certificate = certificates[tokenId];
    ImageStorage faceImage = certificate.faceImage;
    string memory txHashImageData = faceImage.txHashImageData(imagePartIndex);

    return txHashImageData;
}
```

We wrote an article to help users to do it themself here: https://medium.com/art-for-all-nft-platform/how-to-retrieve-on-chain-large-sized-data-df65adef0c2a

3.2 Provenance Proof

As soon as the original data is written on-chain, a provenance proof is not required. The on-chain T2 data itself is a proof of ownership.

3.3 Conclusion ✅

We have here the best storage methods to ensure ownership of image files.

- Anyone can follow the watch condition by consulting the images on-chain at any time.
- Absolutely no one can alter or delete these images as long as the Polygon blockchain exists.
- Even if the Watch Certificate company no longer exists, all owners will still be able to retrieve all these images as long as the Polygon blockchain exists.

4. The Watch History (T3)

There are many other information written in the certificate that let to keep track of the watch condition and its estimated price:

MONTRE
Marque
Modèle
Référence
Année
Diamètre
Poids brut

VALEUR DE MARCHÉ
Estimation
Date de l'estimation

IDENTIFIANT
Watch Certificate™ ID

ÉTAT & CONFORMITÉ
Score d'état
Score de conformité de la montre
Score de conformité du bracelet

LUMENS
Lumens
Lumens
Matière lumens

PROFESSIONNEL
Inspecté par
Date de l'inspection
Société
Adresse de la société
N° d'immatriculation

EXPERT
Validé par
Date de la validation
Société
Adresse de la société
N° d'immatriculation

COMMENTAIRES
Du professionnel
Maintenance
De l'expert

PROPRIÉTAIRE
Email
Nom complet
Date de naissance

BLOCKCHAIN
Type de blockchain
Type de titre
Date de création
Preuve de création
Adresse du contrat
Stockage du titre de propriété
Propriétaire

VÉRIFICATION DU STATUT DE VOL
Résultat
Date de la recherche

4.1 Storage & Persistence

We notice that all these informations are on-chain, except:

- Owner private informations : e-mail / full name / birth date
- Estimated price by the expert
- Serial number of the case and the movement

The on-chain data are in the NFT & publicly readable by using the

These data are updatable by Watch Certificate after an expert certification by using the ***update()*** function:

```
function update(uint256 tokenId, string memory pdfDigest, string memory jsonDigest, string memory jsonString) public {
    require(_exists(tokenId), "WTC: tokenId must exist to update certificate");
    //solhint-disable-next-line max-line-length
    require(_isApproved(_msgSender(), tokenId) || hasRole(MINTER_ROLE, _msgSender()),
"WTC: must have minter role or be approved to update");
    certificates[tokenId].pdfDigest = pdfDigest;
    certificates[tokenId].jsonDigest = jsonDigest;
    certificates[tokenId].jsonString = jsonString;
    emit CertificateUpdate(tokenId, certificates[tokenId].certificateCardId, certificates[tokenId].pdfDigest,
certificates[tokenId].jsonDigest, certificates[tokenId].jsonString);
}
```

This is necessary to update the T3 data in order to keep the condition of the watch, its estimated price, etc., up-to-date.

The updates of each certificate are tracked and can be found in the blockchain history forever. Watch Certificate provides a user-friendly tool to retrieve it from any date.

The three types of data mentioned above that are off-chain are not included in the NFT for understandable reasons. This choice is necessary to respect the privacy of users and to avoid unwanted analysis of price estimation evolution and watch models in the market by using the blockchain. To ensure the consistency of these off-chain data, Watch Certificate chose to use a provenance proof explained below.

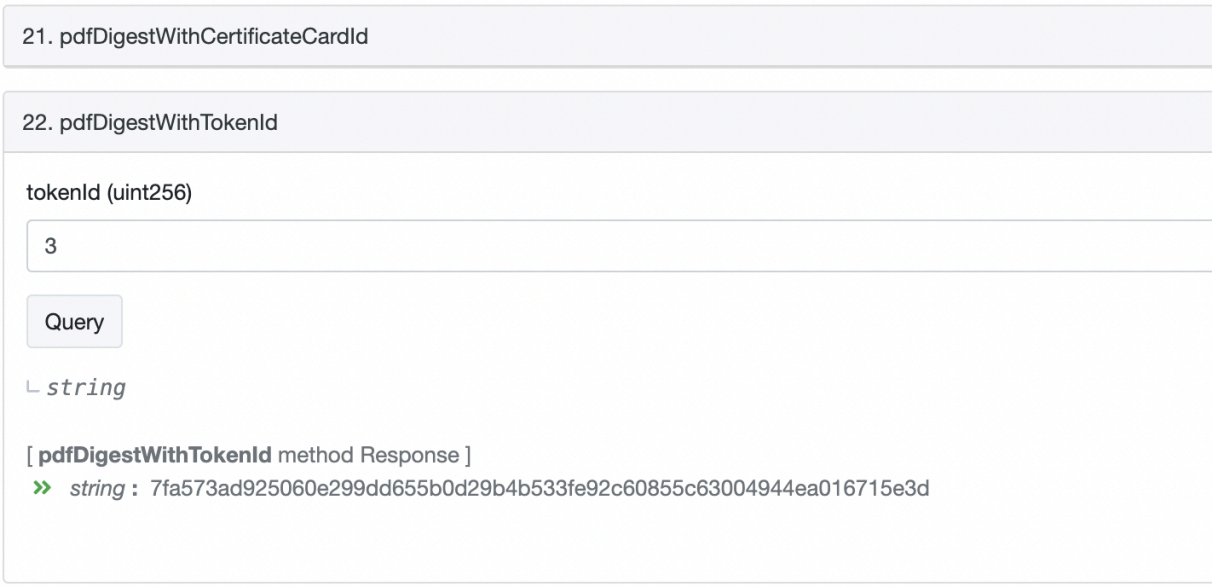
4.2 Provenance Proof

For the T3 on-chain information, a provenance proof is not required once the original data is written on-chain.

However, other sensitive off-chain information can be proven by using SHA256.
Watch Certificate writes on-chain the provenance proof of these data by using ***pdfDigestWithTokenId()*** and ***pdfDigestWithCertificateId().***

```
function pdfDigestWithTokenId(uint256 tokenId) public view returns (string memory) {
    require(_exists(tokenId), "WTC: tokenId must exist to get certificate digest");

    return certificates[tokenId].pdfDigest;
}
```

Anyone can verify the provenance proof of any PDF certificate by hashing it with SHA256 and comparing it with the results of these two functions on Polygon Scan.



Watch Certificate also provides a user-friendly tool to make this verification : https://medium.com/@contact_68746/nft-validator-c702fad20b42

4.3 Conclusion ✅

In conclusion, the T3 data is mostly on-chain, in the NFT itself, meaning that anyone can read it. Even if these data are updatable on-chain, it can be tracked as long as Polygon keeps all the changes in the blockchain. This guarantees that the watch history is stored, readable, and persistent forever. The off-chain T3 data, which includes confidential data, needs to be kept unreadable for business reasons. However, this does not compromise

ownership, as it is verifiable by using the provenance proof stored on the blockchain.

The storage methods used by Watch Certificate ensure the ownership of the watch history:

- The information is readable by anyone on the blockchain
- Only Watch Certificate can certify an update that is stored on-chain forever
- Confidential data is verifiable by using on-chain provenance proof
- All users will be able to retrieve this information even if Watch Certificate company no longer exists, as long as Polygon blockchain exists.

5. Documentation

Watch Certificate helps users retrieve information on every certificate in a user-friendly way. They provide a certificate verification tool at https://nft.watchcertificate.org/. The tool allows users to verify if a PDF certificate is really on the blockchain and signed by Watch Certificate. Additionally, users can track the history of a watch by browsing all the certifications of the specific watch they are looking for. This makes the readability of blockchain data much easier. A tutorial for this tool can be found at https://medium.com/@contact_68746/nft-validator-c702fad20b42.

The Watch Certificate team has also written an article to assist users in verifying certificates directly from a blockchain explorer like Polygon Scan, without using their tool. Even if the Watch Certificate company no longer exists, any user can still verify any certificate without the tool described above. Although this approach is more low-level and less user-friendly, it is a highly decentralized way to verify certificates.

Here is the tutorial :
https://medium.com/@contact_68746/proof-of-integrity-on-polygon-3b4841d647f5

6. Audit Conclusion

After auditing Watch Certificate's smart contract, we can conclude that the company has implemented the most advanced and secure technical standards to ensure the highest level of security and integrity of watch information. It's worth noting and commending their initiative to store images inside the blockchain, which makes the data ultra-persistent. This is a rare feature that sets Watch Certificate apart from others in the industry. Furthermore, the documentation provided is highly appreciated and easily accessible to users who want to verify watch certifications on the blockchain without advanced technical skills.

In summary, every user can be assured that:

- All certifications are accessible, readable, and available to anyone as long as the Polygon blockchain exists, even if Watch Certificate no longer exists.
- Photos of the watches are also available on the blockchain.
- No one, except Watch Certificate, can add or update certifications on the blockchain.
- All certification updates on the blockchain are traceable, persistent, accessible, and readable by anyone.
- Sensitive and confidential data, such as owner information and watch serial numbers, are encrypted and verifiable by anyone but not readable in plain text.
- The verification methods are well-documented by Watch Certificate and can still be used even if the company no longer exists.
- Watch Certificate achieves a **Gem score** of **100/100** and meets all of our criteria for persistence, sovereignty, and decentralization.

How to be audited

Our team is working hard to audit platforms and technologies to make NFT projects more transparent.

Ownership remains the key to ensuring the democratization of NFT.

How to be audited

We are 100% independent.
No ads, no influence for brands.
No one can pay OpenGem to advertise any NFT project.
No one can pay OpenGem to upgrade any Gem score.

We make audits.

✅ **If you want your project to be audited, contact us here.**