UPC

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

# Project technical report

Document versions

| **Version:** | **Modifications:** | **Made by:** |
|---|---|---|
| V1.0 | Creación del documento | UPC |
| V2.0 | Definición técnica | UPC |

# Index

1. Project requirements and context....4
2. Technical definition of the problem....5
3. Certifydoc analysis methodology....5
4. Certifydoc certification process....7
4.1. Timestamp Protocol....8
4.2. European Regulation: eIDAS....9
5. Blockchain Legalization Engine Analysis....11
5.1. Certification of Blockchain transactions via web application....11
5.2. Blockchain transaction certification via API....14
5.3. Verification process....15
6. File certification....17
7. Conclusions....22

# 1. Project requirements and context

This project consists of the analysis and validation of Certifydoc's Blockchain Legalization Engine (BLE), a solution designed to notarize data stored in the blockchain with legal relevance. The BLE platform acts as an intermediary between blockchain data and qualified EU trust service providers.

During the course of the project, the analysis and validation of the certification of Certifydoc files has also been carried out. The service is accessible on-demand and is offered in Software as a Service mode, being able to interact with the system through a web form or through a REST API. In turn, the results of the notarization can be available via email or through the REST API.

## Project requirements

The project requires the execution of several phases to guarantee its technical and commercial viability:

- **Testing and experimentation:** Validate the technical proposal through concept tests, both technological and with end users, to verify the functionality and usability of the service.
- **Technical and process advice:** Consulting to improve processes and technological aspects based on the results of initial tests.
- **Security and performance evaluation:** Specific tests to ensure the integrity and protection of data, as well as the stability and performance of the system in real use scenarios.
- **Feasibility report:** Development of a detailed report that concludes the testing findings and validates the project proposal.

## Project context

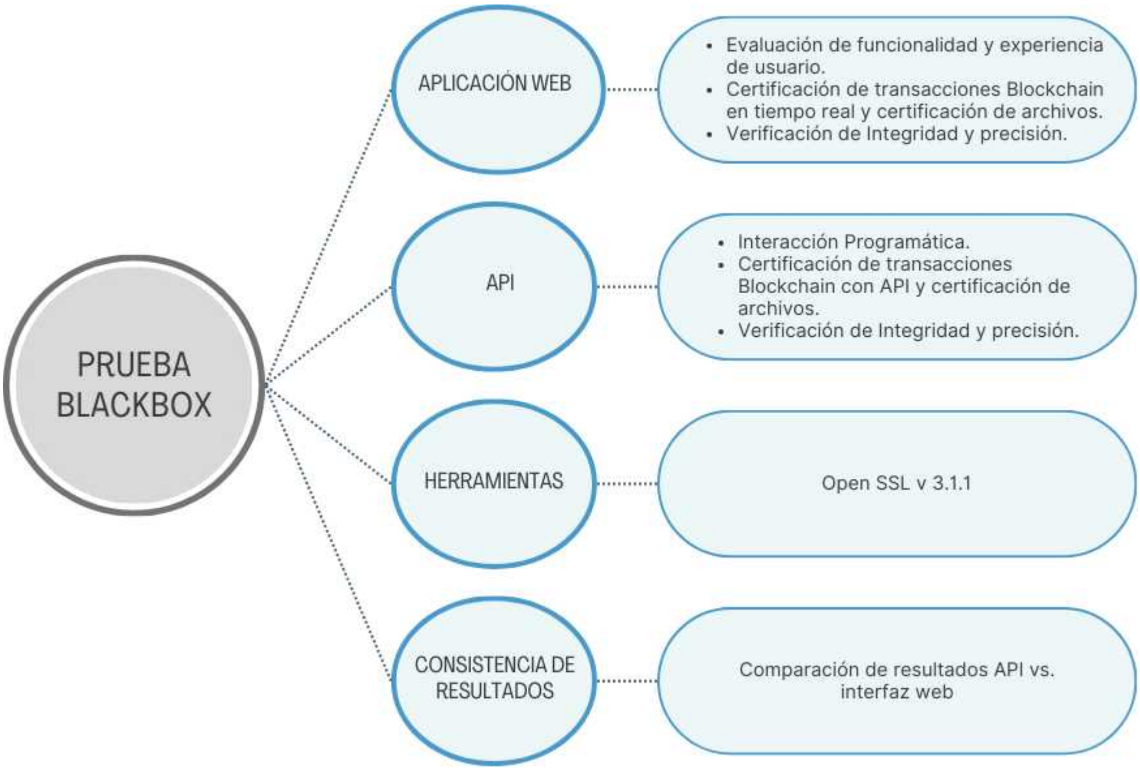
This project is part of the Services Proposal of the Digital Innovation Hubs Support Program, seeking to explore and expand the possibilities of notarization of documents with legal relevance to blockchain technologies. The BLE solution not only proposes a technological innovation but also aligns with the current needs for digitization and document security in the European Union, offering a practical and scalable approach that can be adopted in various sectors and applications.

## 2. Technical definition of the problem

Advice and evaluation on Certifydoc certification services;  Blockchain Legalization Engine platform that offers Blockchain transactions certification services by creating timestamps with the hash of the transaction and its subsequent certification through a European certifier, and file certification platform. The analysis has been carried out using a Black Box methodology.

## 3. Certifydoc analysis methodology

The analysis of Certifydoc's Blockchain Legalization Engine (BLE) and file certification has been carried out using a black box approach, evaluating the application from an external perspective without accessing the source code or the internal structure of the system.


PRUEBA BLACKBOX
APLICACIÓN WEB
• Evaluación de funcionalidad y experiencia de usuario.
• Certificación de transacciones Blockchain en tiempo real y certificación de archivos.
• Verificación de Integridad y precisión.
API
• Interacción Programática.
• Certificación de transacciones Blockchain con API y certificación de archivos.
• Verificación de Integridad y precisión.
HERRAMIENTAS
Open SSL v 3.1.1
CONSISTENCIA DE RESULTADOS
Comparación de resultados API vs. interfaz web

This analysis was performed from two different interfaces to provide a comprehensive evaluation:

1. **From the Certifydoc web application:**
   a. Access as a typical user: The Certifydoc web application was accessed as a typical user to evaluate both the general functionality and user experience. This included navigating the interface, managing accounts, and performing basic functions.
   b. Blockchain transaction and file certification: Transaction certification testing was performed through the user interface to observe the

certification process in real time and evaluate the usability and accessibility of the system. As well as file certification tests.

   c. Integrity and accuracy verification: Information provided during the certification process was checked for integrity and accuracy, ensuring that displayed data and results were consistent and reliable.

2. **Using the Certifydoc API**:
   a. Programmatic interaction: The API provided by Certifydoc was used to interact with the platform programmatically, which allowed the flexibility and integration capacity of the service to be evaluated.
   b. Certification of blockchain transactions and files: Certifications of blockchain transactions were executed using the various functions available in the API, with the objective of testing the robustness and efficiency of these programmatic interfaces. As well as file certification tests.
   c. Integrity and accuracy verification: Information provided during the certification process was checked for integrity and accuracy, ensuring that displayed data and results were consistent and reliable.
   d. Consistency of results: The consistency of the results obtained through the API compared to those generated through the web interface was verified, ensuring that both platforms provided equivalent results.

**Verification tools:**

To verify the validity of the certifications, OpenSSL (version 3.1.1) was used, an open source cryptographic library that facilitates the implementation of security protocols. This software made it possible to confirm the authenticity and security of the certifications generated by BLE, ensuring that the system complies with current security standards.

## 4. Certifydoc certification process

Certifydoc uses advanced technology to certify both blockchain transactions and digital files, ensuring the authenticity and integrity of data by creating verifiable timestamps.

### Common technical process

For all types of certification, Certifydoc uses secure cryptography techniques to ensure that timestamps and certifications are irrefutable and legally valid. Additionally, it uses the RFC 3161 standard to ensure global compatibility and recognition of certifications. Each certification issued by Certifydoc can be independently verified using tools like OpenSSL, providing transparency and reliability.

### Advantages of the process

- Security: The use of SHA256 hashing and the RFC 3161 protocol ensures robust security against tampering and forgery.
- Flexibility: Ability to handle both individual transactions and multiple files, offering solutions adapted to various needs.
- Verifiability: Each certification can be independently verified, offering a high level of transparency and trust for users.

Below is a detail of how each type of certification is carried out:

Type A) Certification of a single file. This process involves:

- **Hash calculation**: The user uploads the file to the Certifydoc platform, which automatically calculates the SHA256 hash of the file.
- **Timestamp request**: As with blockchain transactions, a timestamp query is generated for the file hash.
- **Timestamp response**: An authorized certifier issues a timestamp response that verifies the integrity and authenticity of the file at the specified time.

Type B) Multiple file certifications with and without strong source encryption (up to 17 MB). This process involves:

- **Creation of the .zip file**: The user selects several files, which Certifydoc compresses into a single .zip file.
- **.zip hash**: The SHA256 hash of the entire .zip file is calculated.
- **Timestamp process**: The same timestamp request and response process is followed, certifying the set of files as a single entity.

Type C) Unlimited file hash certifications. This process involves:

- **Obtain the SHA256 Hash of files, partitions, disks, memory areas with a third-party tool**. Typically only expert users use this type of process, optimized to create certified backup copies of device memories for judicial investigations by computer forensics.
- **Timestamp process**: The same timestamp request and response process is followed, certifying the set of files as a single entity.

Type D) Certification of blockchain transactions in which Certifydoc certifies the hash of the blockchain transaction. This process involves:

- **Receiving the hash**: The user provides the hash of the blockchain transaction they want to certify.
- **Timestamp generation**: Certifydoc generates a timestamp query (.tsq) for the given hash.
- **Certification by authorized entity**: The hash is sent to a qualified time certification service, which returns a timestamp response (.tsr) certifying the exact time of the transaction.

The real objective of the project is the analysis of the blockchain transaction certification process (Type D), additionally the file certification analysis has been added (Types A, B and C).

## 4.1. Timestamp Protocol

The Timestamp Protocol used in Certifydoc is based on the RFC 3161 standard, known as "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)." This protocol is fundamental to the issuance of digital timestamps, and is widely recognized and used in the industry to ensure the integrity and authenticity of electronic documents.

**Timestamp protocol operation**

- **Definition and purpose**: The time stamping service allows agencies to supply electronic documents to be stamped. A timestamp is an electronic signature performed by a Time Stamping Authority (TSA), demonstrating that the data has existed and has not been altered since a specific time, based on a reliable source of time.

- **Technical process**: The Legalization Blockchain Engine generates a Timestamp Query (.tsq) that will subsequently be certified by a European certifier, which will generate a Timestamp Response (.tsr).

    - **Timestamp query**: It is a request sent to a time certification service to obtain a timestamp associated with a specific data or event. This request generally includes the cryptographic hash of a document or transaction, and asks for a timestamp that certifies the time of occurrence of that data.

- **Timestamp response**: It is the response from the time certification service that includes the generated timestamp, a digital timestamp that verifies when the data specified in the query was certified.

**RFC 3161 Technical Specifications**

- **Message format**: RFC 3161 specifies how request and response messages should be structured, including required and optional fields.
- **Cryptographic algorithms**: Details recommended algorithms for signing and verifying timestamps, along with security considerations to preserve the integrity and authenticity of issued timestamps.

**Integration and application**

- **OpenSSL TS (Timestamping)**: This is a subset of OpenSSL specifically designed to generate and verify timestamps in accordance with the RFC 3161 standard. It offers both a command line interface and a programmatic API, facilitating integration into a variety of applications and systems. OpenSSL TS supports multiple input and output formats, making it extremely versatile and suitable for numerous industrial applications.

This standard provides a solid foundation for the implementation of interoperable and secure timestamping services, and is crucial for the efficient and reliable operation of Certifydoc's Blockchain Legalization Engine.

## 4.2. European Regulation: eIDAS

Throughout our analysis of the Blockchain Legalization Engine and its ability to deliver an efficient and secure user experience, it is crucial to consider the regulatory framework in which these technologies operate. Regulation (EU) No. 910/2014, better known as eIDAS, is the legal framework that supports the certification and temporary stamping services offered by platforms such as Certifydoc.

**Strong eIDAS legal framework**

The eIDAS Regulation establishes a robust legal framework for electronic transactions within the European Union's Digital Single Market, specifically addressing electronic identification and trust services such as electronic timestamping. This framework not only promotes the use of secure and trusted electronic identities for cross-border transactions, but also ensures the interoperability of electronic timestamps across the EU.

**Impact and Benefits of eIDAS**

- **Improved interoperability:** eIDAS ensures that a country's electronic identification systems are recognized by all other Member States, thus promoting efficiency and security in cross-border transactions.

- **Transaction Security:** This regulation raises the level of security for business transactions, minimizing the administrative burden and reducing costs, resulting in more efficient business processes and an increase in profits.
- **Trust and Legal Acceptance:** Trust services that comply with eIDAS can be used as evidence in judicial proceedings, which is crucial for the legal acceptance and validity of digital transactions and certifications.

### Relevance of eIDAS for Certifydoc

- **Security and authenticity standards:** eIDAS defines the security standards that services such as the Blockchain Legalization Engine must meet to guarantee the authenticity and integrity of digital certifications. This is especially pertinent given that our analyses have confirmed the robustness and reliability of the Certifydoc platform in issuing verified timestamps.
- **Interoperability in the EU:** The regulation makes it easier for time stamps issued by Certifydoc to be recognized and accepted in all EU Member States, which is vital for cross-border operations and for companies operating on a European scale.
- **Legal framework for trust services:** By establishing a legal framework for time stamping services, eIDAS helps ensure that these services maintain their validity and reliability, strengthening trust in the solutions provided by Certifydoc to its users.

### Impact of eIDAS on blockchain technology adoption

The inclusion of requirements for timestamping services under eIDAS is a significant step towards the integration of blockchain technologies into the European digital ecosystem. By ensuring that documents and transactions are secure and legally validated, eIDAS not only supports technology adoption but also promotes trust and security on digital platforms across Europe.

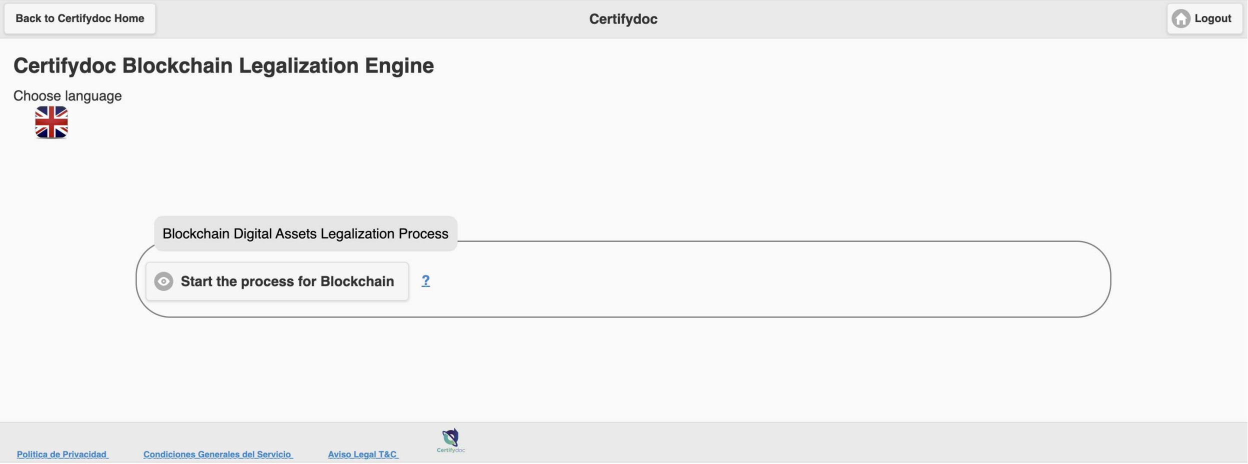
The framework provided by eIDAS is essential for the operation and expansion of services such as those offered by Certifydoc. Furthermore, it establishes the necessary legal and technical foundations for a broader adoption of blockchain-based certification solutions, ensuring that these technologies are harmoniously integrated into the legal and business framework of the European Union.

# 5. Blockchain Legalization Engine Analysis

This section details the operation of the Blockchain Legalization Engine, describing the protocols and procedures used for timestamp generation and certification, as well as the interaction methods through the web application and the API.

## 5.1. Certification of Blockchain transactions via web application

Certifydoc's Blockchain Certification Engine has a web application (https://www.certifydoc.eu/blockchain-legalization-engine/) from which clients can generate certificates using a graphical interface.



Back to Certifydoc Home

Certifydoc

Logout

Certifydoc Blockchain Legalization Engine

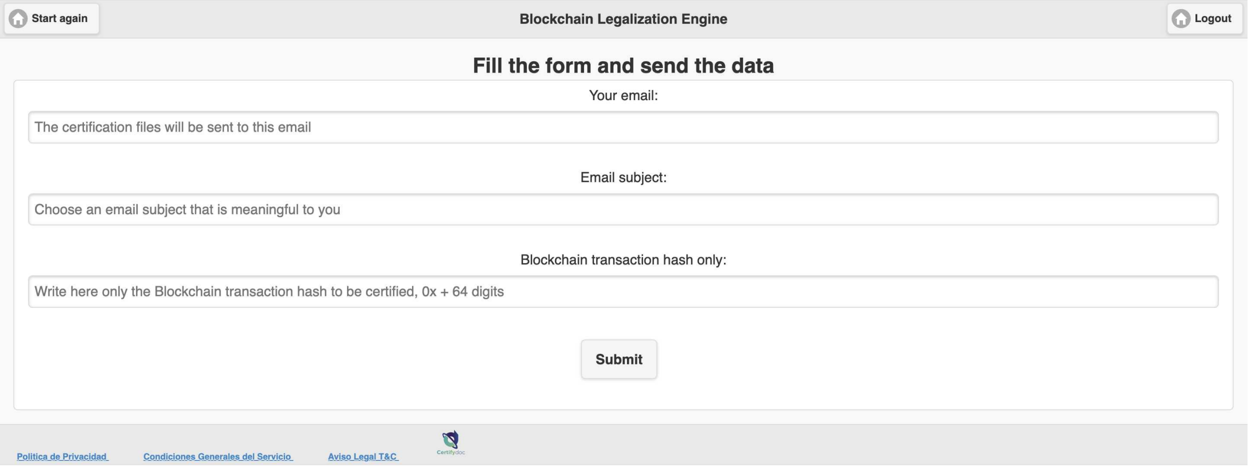
Choose language

Blockchain Digital Assets Legalization Process

Start the process for Blockchain ?

Politica de Privacidad Condiciones Generales del Servicio Aviso Legal T&C

There is a single process to generate Blockchain transaction certificates:



Start again

Blockchain Legalization Engine

Logout

Fill the form and send the data

Your email:

The certification files will be sent to this email

Email subject:

Choose an email subject that is meaningful to you

Blockchain transaction hash only:

Write here only the Blockchain transaction hash to be certified, 0x + 64 digits

Submit

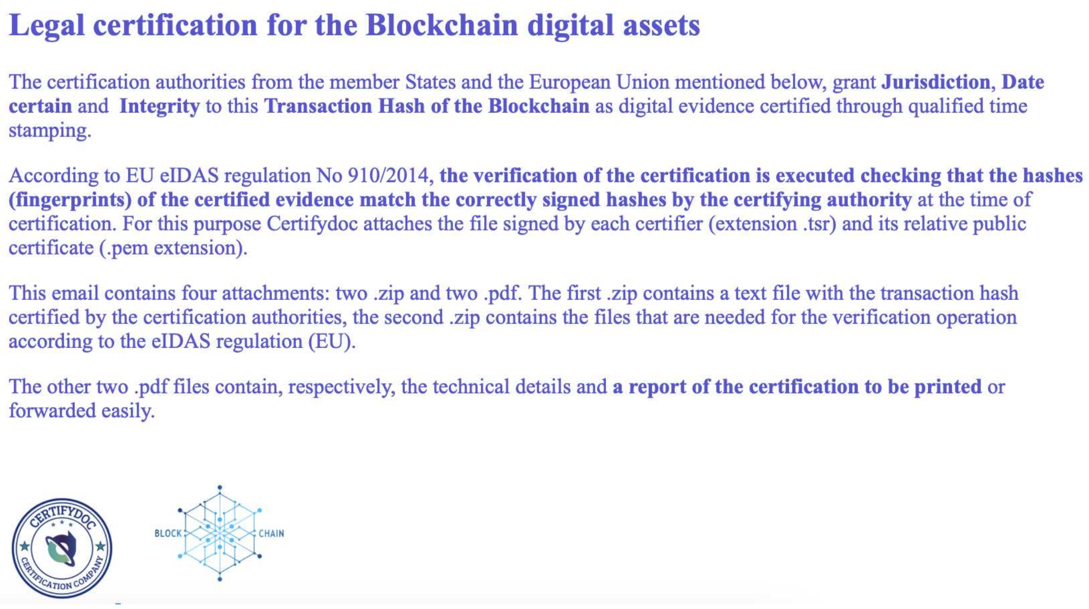
Politica de Privacidad Condiciones Generales del Servicio Aviso Legal T&C

In this menu the user must complete the following information:

1. **Email:** Email to which the certification will arrive.
2. **Email subject:** Subject of the email with the certification.
3. **Blockchain transaction hash:** The hexadecimal hash of the Blockchain transaction to be certified.

Once the form is submitted, the user consumes a certification.

The user will receive an email with information and a series of documents related to the certification:



## Legal certification for the Blockchain digital assets

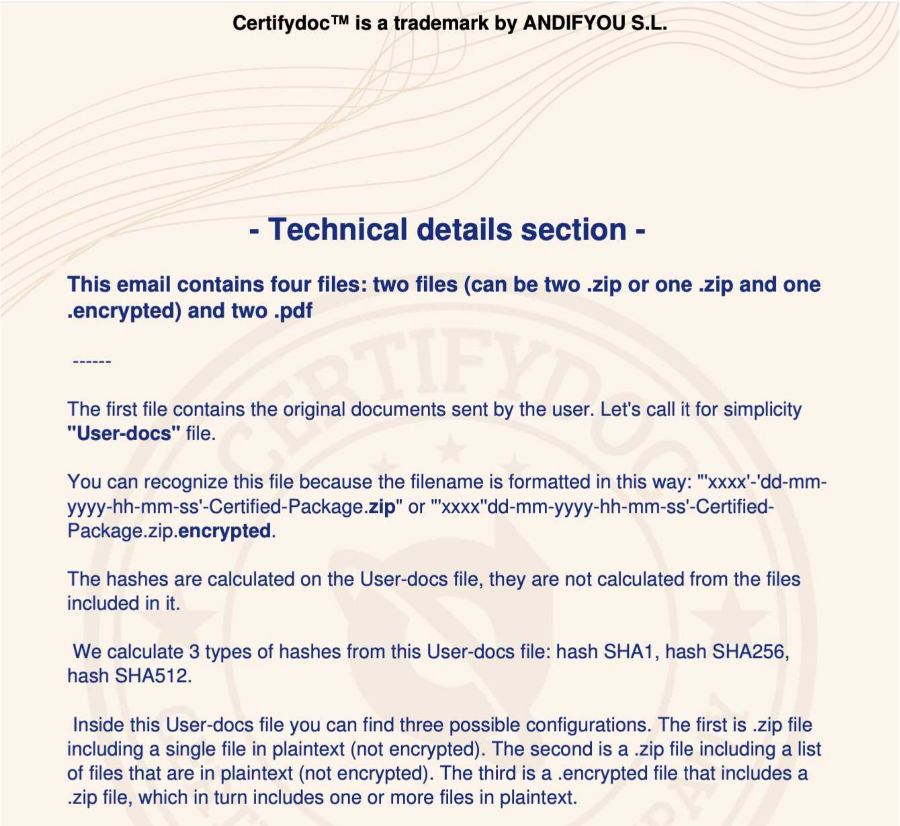
The certification authorities from the member States and the European Union mentioned below, grant **Jurisdiction**, **Date certain** and **Integrity** to this **Transaction Hash of the Blockchain** as digital evidence certified through qualified time stamping.

According to EU eIDAS regulation No 910/2014, **the verification of the certification is executed checking that the hashes (fingerprints) of the certified evidence match the correctly signed hashes by the certifying authority** at the time of certification. For this purpose Certifydoc attaches the file signed by each certifier (extension .tsr) and its relative public certificate (.pem extension).

This email contains four attachments: two .zip and two .pdf. The first .zip contains a text file with the transaction hash certified by the certification authorities, the second .zip contains the files that are needed for the verification operation according to the eIDAS regulation (EU).

The other two .pdf files contain, respectively, the technical details and **a report of the certification to be printed** or forwarded easily.

1. **Technical details:** PDF file that explains the technical details of the other files in the email.



**Certifydoc™ is a trademark by ANDIFYOU S.L.**

## - Technical details section -

**This email contains four files: two files (can be two .zip or one .zip and one .encrypted) and two .pdf**

------

The first file contains the original documents sent by the user. Let's call it for simplicity **"User-docs"** file.

You can recognize this file because the filename is formatted in this way: "'xxxx'-'dd-mm-yyyy-hh-mm-ss'-Certified-Package.**zip**" or "'xxxx''dd-mm-yyyy-hh-mm-ss'-Certified-Package.zip.**encrypted**.

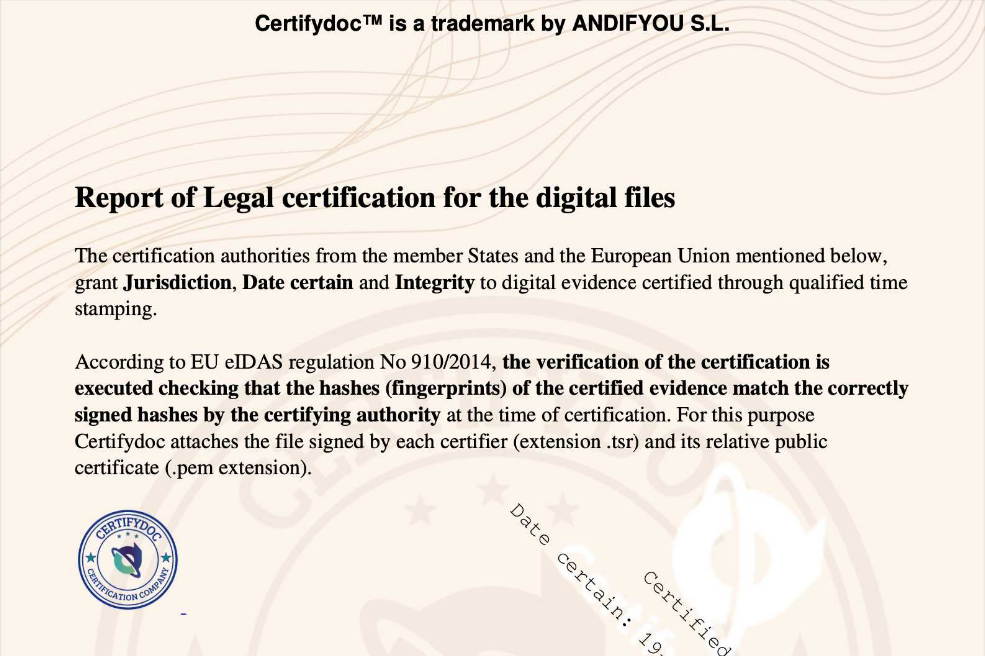
The hashes are calculated on the User-docs file, they are not calculated from the files included in it.

We calculate 3 types of hashes from this User-docs file: hash SHA1, hash SHA256, hash SHA512.

Inside this User-docs file you can find three possible configurations. The first is .zip file including a single file in plaintext (not encrypted). The second is a .zip file including a list of files that are in plaintext (not encrypted). The third is a .encrypted file that includes a .zip file, which in turn includes one or more files in plaintext.

2. **Certification report:** PDF file is an explanation of the certification. It is important to note that this file only represents an explanatory report of the

certification. In the case of a forensic expert, the legal value of the certificate consists of the .tsr response files and the .pem public certificate of the certifying authority.



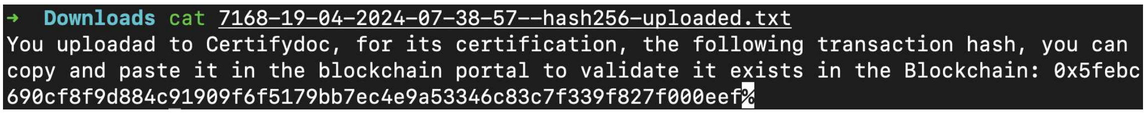
**Certifydoc™ is a trademark by ANDIFYOU S.L.**

**Report of Legal certification for the digital files**

The certification authorities from the member States and the European Union mentioned below, grant **Jurisdiction**, **Date certain** and **Integrity** to digital evidence certified through qualified time stamping.

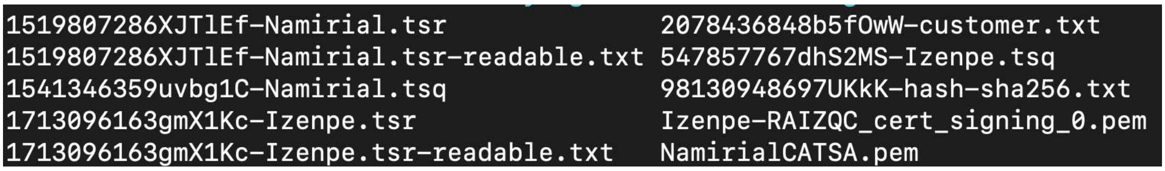
According to EU eIDAS regulation No 910/2014, **the verification of the certification is executed checking that the hashes (fingerprints) of the certified evidence match the correctly signed hashes by the certifying authority** at the time of certification. For this purpose Certifydoc attaches the file signed by each certifier (extension .tsr) and its relative public certificate (.pem extension).

3. **User data:** ZIP file containing a .txt with the hash of the certified Blockchain transaction.



```
➜  Downloads cat 7168-19-04-2024-07-38-57--hash256-uploaded.txt
You uploadad to Certifydoc, for its certification, the following transaction hash, you can
copy and paste it in the blockchain portal to validate it exists in the Blockchain: 0x5febc
690cf8f9d884c91909f6f5179bb7ec4e9a53346c83c7f339f827f000eef%
```

4. **Certification:** ZIP file that contains the certification and the files necessary to verify it.



```
1519807286XJTlEf-Namirial.tsr                2078436848b5fOwW-customer.txt
1519807286XJTlEf-Namirial.tsr-readable.txt   547857767dhS2MS-Izenpe.tsq
1541346359uvbg1C-Namirial.tsq                98130948697UKkK-hash-sha256.txt
1713096163gmX1Kc-Izenpe.tsr                  Izenpe-RAIZQC_cert_signing_0.pem
1713096163gmX1Kc-Izenpe.tsr-readable.txt     NamirialCATSA.pem
```

## 5.2. Blockchain transaction certification via API

Certifydoc's Blockchain Certification Engine has an API that allows us to generate certifications programmatically. The API has 4 different possible processes:

1. **Email response:** The API will send us an email with the files. This process has the same result as doing the certification via the web.

   https://www.certifydoc.eu/wp-json/certifydocapi/v1/sdbx-notarization-api/

   ?submitethereum=submitethereum

   &hashethereum=0x9c820dc62b09fcb517ff22d4888d06c8cc858d59cbefff8830865814a852e1a3

   &response_type=email_response

   &emailsubjectethereum=Test API sandbox Blockchain to email En 230508

   &emailethereum=email@dominio.es

   &language=en

2. **API response complete:** We receive the same files as in the email but as a response from the API. The files are base64 encoded.

   https://www.certifydoc.eu/wp-json/certifydocapi/v1/sdbx-notarization-api/

   ?submitethereum=submitethereum

   &hashethereum=0x9c820dc62b09fcb517ff22d4888d06c8cc858d59cbefff8830865814a852e1a3

   &response_type=api_response

   &api_response_options=complete

   &language=en

3. **API response reduced:** We receive the response through the API. It sends us the Timestamp Response (.tsr), the CA certificate (.pem) and the certification report (.pdf). The files are base64 encoded.

   https://www.certifydoc.eu/wp-json/certifydocapi/v1/sdbx-notarization-api/

   ?submitethereum=submitethereum

   &hashethereum=0xa85c41d8ffd5212a7537b4556eb8c748b723fbb51a56f67b7961e830087adb6b

   &response_type=api_response

   &api_response_options=reduced

&language=en

4. **API response minimized:** We receive the response through the API. It sends us the Timestamp Response (.tsr) and the CA certificate (.pem). The files are base64 encoded.

   https://www.certifydoc.eu/wp-json/certifydocapi/v1/sdbx-notarization-api/

   ?submitethereum=submitethereum

   &hashethereum=0xea586e91e71a09e9fb27fa1cfbfdeea582269f2bc30da37d8ed75b6facc6bea7

   &response_type=api_response

   &api_response_options=minimized

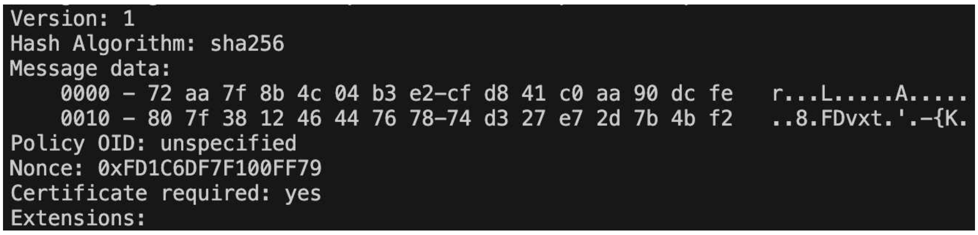
   &language=en

### 5.3. Verification process

We can verify the information contained in the different files:

1. The **Timestam Query** (.tsq) contains useful information such as the hash to be certified. You can verify its data by executing the command:

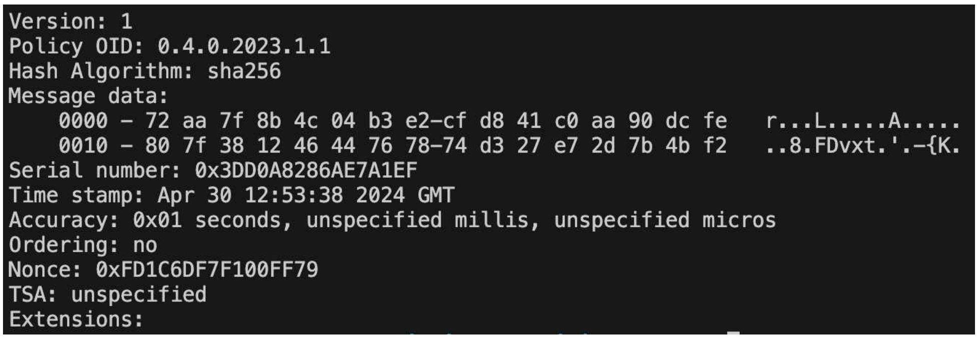
```
openssl ts -query -in <tsq_filename>.tsq -text
```



```
Version: 1
Hash Algorithm: sha256
Message data:
    0000 - 72 aa 7f 8b 4c 04 b3 e2-cf d8 41 c0 aa 90 dc fe   r...L.....A.....
    0010 - 80 7f 38 12 46 44 76 78-74 d3 27 e7 2d 7b 4b f2   ..8.FDvxt.'.-{K.
Policy OID: unspecified
Nonce: 0xFD1C6DF7F100FF79
Certificate required: yes
Extensions:
```

2. The **Timestamp Response** (.tsr) contains useful information such as the certificate hash and the timestamp of the certificate. You can verify its data by running the command:

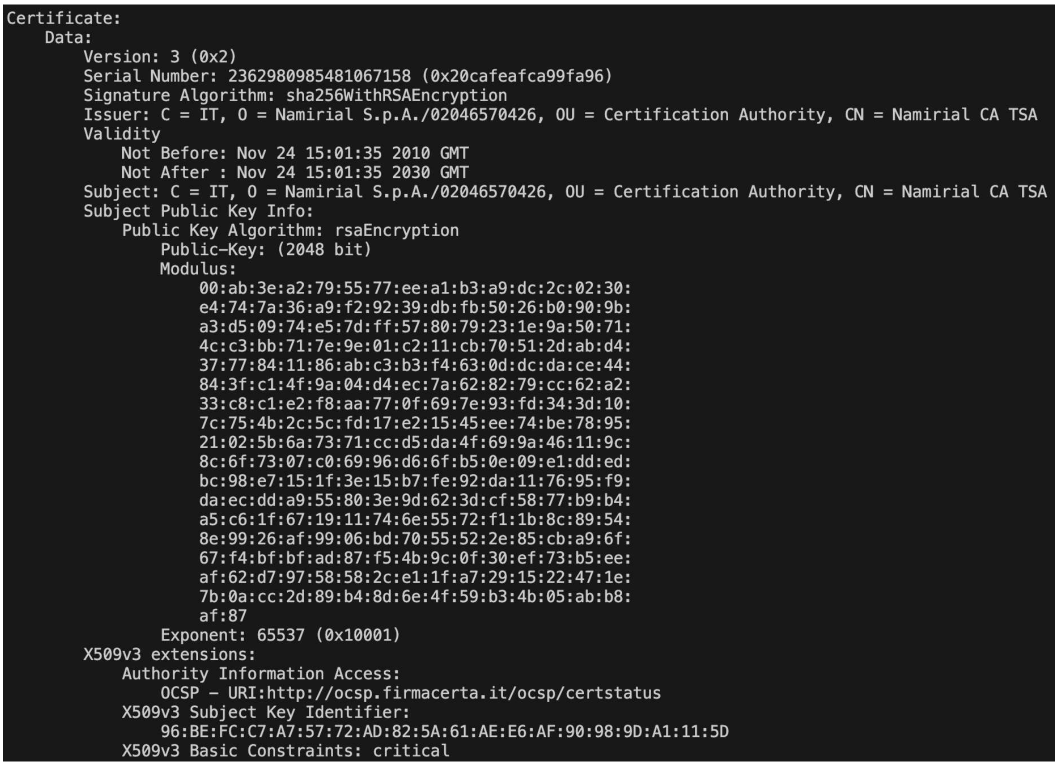
```
openssl ts -reply -in <tsr_filename>.tsr -text
```



```
Version: 1
Policy OID: 0.4.0.2023.1.1
Hash Algorithm: sha256
Message data:
    0000 - 72 aa 7f 8b 4c 04 b3 e2-cf d8 41 c0 aa 90 dc fe   r...L.....A.....
    0010 - 80 7f 38 12 46 44 76 78-74 d3 27 e7 2d 7b 4b f2   ..8.FDvxt.'.-{K.
Serial number: 0x3DD0A8286AE7A1EF
Time stamp: Apr 30 12:53:38 2024 GMT
Accuracy: 0x01 seconds, unspecified millis, unspecified micros
Ordering: no
Nonce: 0xFD1C6DF7F100FF79
TSA: unspecified
Extensions:
```

3. The **CA certificate** (.pem) contains useful information such as the CN of the certifier. You can verify your data by executing the command:

```
openssl x509 -in <pem_filename>.pem -text -noout
```



```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2362980985481067158 (0x20cafeafca99fa96)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = IT, O = Namirial S.p.A./02046570426, OU = Certification Authority, CN = Namirial CA TSA
        Validity
            Not Before: Nov 24 15:01:35 2010 GMT
            Not After : Nov 24 15:01:35 2030 GMT
        Subject: C = IT, O = Namirial S.p.A./02046570426, OU = Certification Authority, CN = Namirial CA TSA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ab:3e:a2:79:55:77:ee:a1:b3:a9:dc:2c:02:30:
                    e4:74:7a:36:a9:f2:92:39:db:fb:50:26:b0:90:9b:
                    a3:d5:09:74:e5:7d:ff:57:80:79:23:1e:9a:50:71:
                    4c:c3:bb:71:7e:9e:01:c2:11:cb:70:51:2d:ab:d4:
                    37:77:84:11:86:ab:c3:b3:f4:63:0d:dc:da:ce:44:
                    84:3f:c1:4f:9a:04:d4:ec:7a:62:82:79:cc:62:a2:
                    33:c8:c1:e2:f8:aa:77:0f:69:7e:93:fd:34:3d:10:
                    7c:75:4b:2c:5c:fd:17:e2:15:45:ee:74:be:78:95:
                    21:02:5b:6a:73:71:cc:d5:da:4f:69:9a:46:11:9c:
                    8c:6f:73:07:c0:69:96:d6:6f:b5:0e:09:e1:dd:ed:
                    bc:98:e7:15:1f:3e:15:b7:fe:92:da:11:76:95:f9:
                    da:ec:dd:a9:55:80:3e:9d:62:3d:cf:58:77:b9:b4:
                    a5:c6:1f:67:19:11:74:6e:55:72:f1:1b:8c:89:54:
                    8e:99:26:af:99:06:bd:70:55:52:2e:85:cb:a9:6f:
                    67:f4:bf:bf:ad:87:f5:4b:9c:0f:30:ef:73:b5:ee:
                    af:62:d7:97:58:58:2c:e1:1f:a7:29:15:22:47:1e:
                    7b:0a:cc:2d:89:b4:8d:6e:4f:59:b3:4b:05:ab:b8:
                    af:87
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Authority Information Access:
                OCSP - URI:http://ocsp.firmacerta.it/ocsp/certstatus
            X509v3 Subject Key Identifier:
                96:BE:FC:C7:A7:57:72:AD:82:5A:61:AE:E6:AF:90:98:9D:A1:11:5D
            X509v3 Basic Constraints: critical
```

Once we have verified that the files have the expected content, we can verify the certificate by executing the following command:

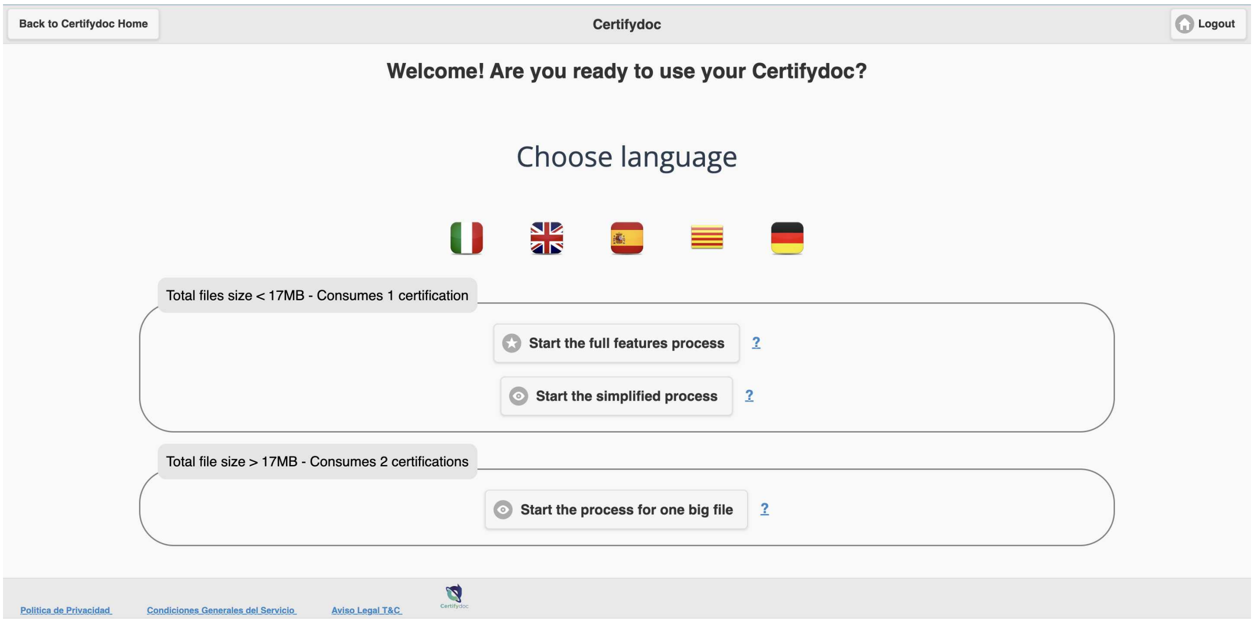
```
openssl ts -verify -in <tsr_filename>.tsr -CAfile <pem_filename>.pem -digest $(cat
<trasaction_to_verify>.txt)
```

## 6. File certification

This section details the operation for file certification, describing the protocols and procedures used for timestamp generation and certification, as well as the interaction methods through the web application and the API.

### 6.1. File certification via web application

Certifydoc file certification has a web application ([https://www.certifydoc.eu/certification/](https://www.certifydoc.eu/certification/)) from which clients can generate certificates using a graphical interface.


Back to Certifydoc Home
Certifydoc
Logout

Welcome! Are you ready to use your Certifydoc?

Choose language

Total files size < 17MB - Consumes 1 certification

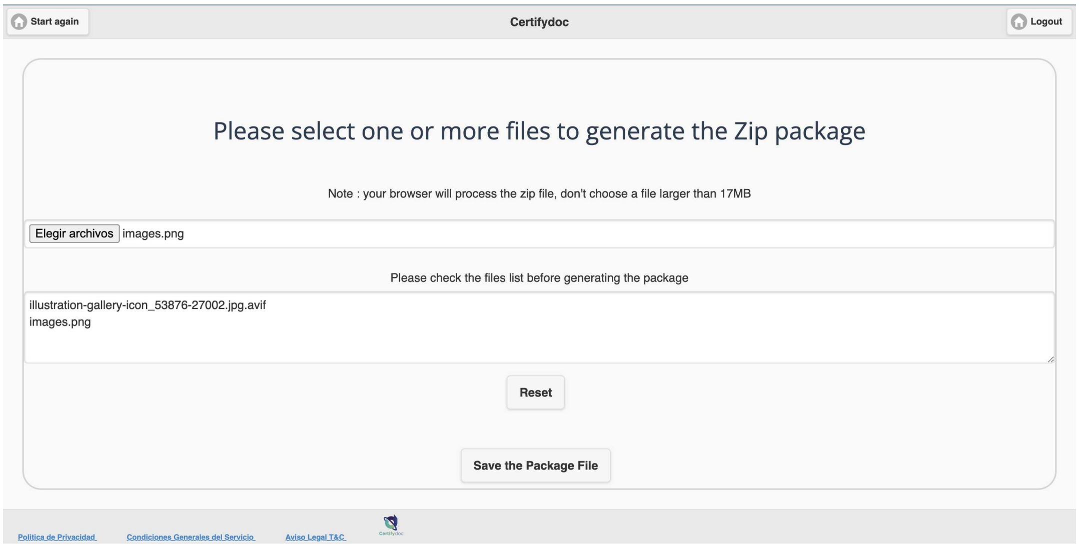
Start the full features process
?
Start the simplified process
?

Total file size > 17MB - Consumes 2 certifications

Start the process for one big file
?
Politica de Privacidad
Condiciones Generales del Servicio
Aviso Legal T&C
Certifydoc

The web application has 3 different processes:

1. **Complete process:** You can upload multiple files, which will be saved inside a ZIP. You can also encrypt the content. It has a limitation of 17MB.



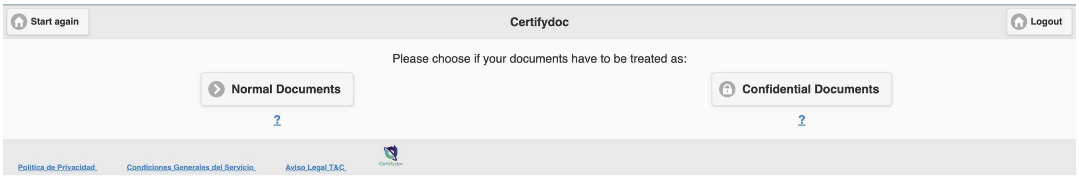
Please select one or more files to generate the Zip package

Note : your browser will process the zip file, don't choose a file larger than 17MB

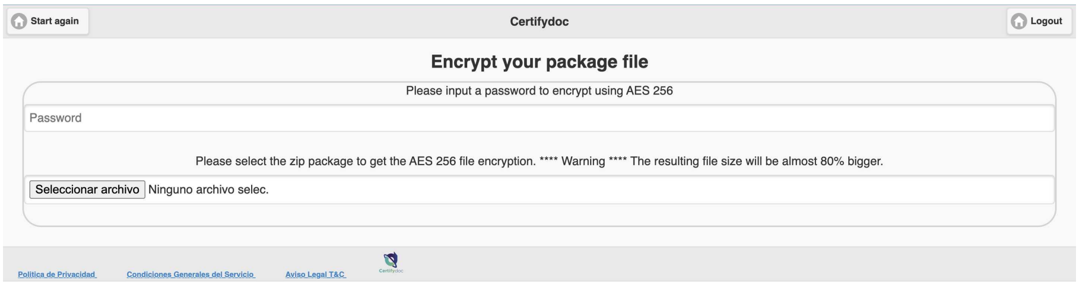
images.png

Please check the files list before generating the package

illustration-gallery-icon_53876-27002.jpg.avif
images.png



Please choose if your documents have to be treated as:



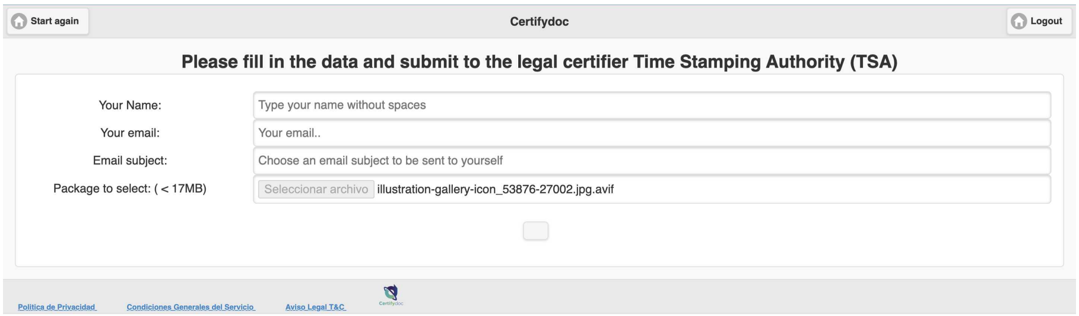
Encrypt your package file

Please input a password to encrypt using AES 256

Password

Please select the zip package to get the AES 256 file encryption. **** Warning **** The resulting file size will be almost 80% bigger.

Ninguno archivo selec.



Please fill in the data and submit to the legal certifier Time Stamping Authority (TSA)

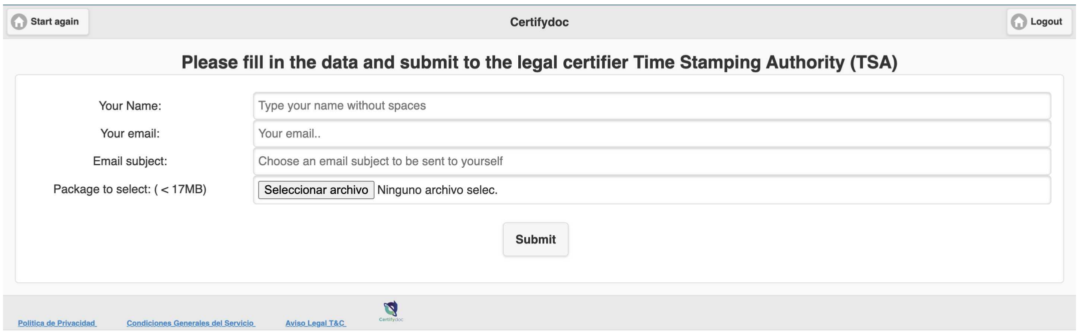
Your Name: Type your name without spaces

Your email: Your email..

Email subject: Choose an email subject to be sent to yourself

Package to select: ( < 17MB) illustration-gallery-icon_53876-27002.jpg.avif

2. **Simplified process:** You can only upload a single file. It has a limitation of 17MB.



Please fill in the data and submit to the legal certifier Time Stamping Authority (TSA)

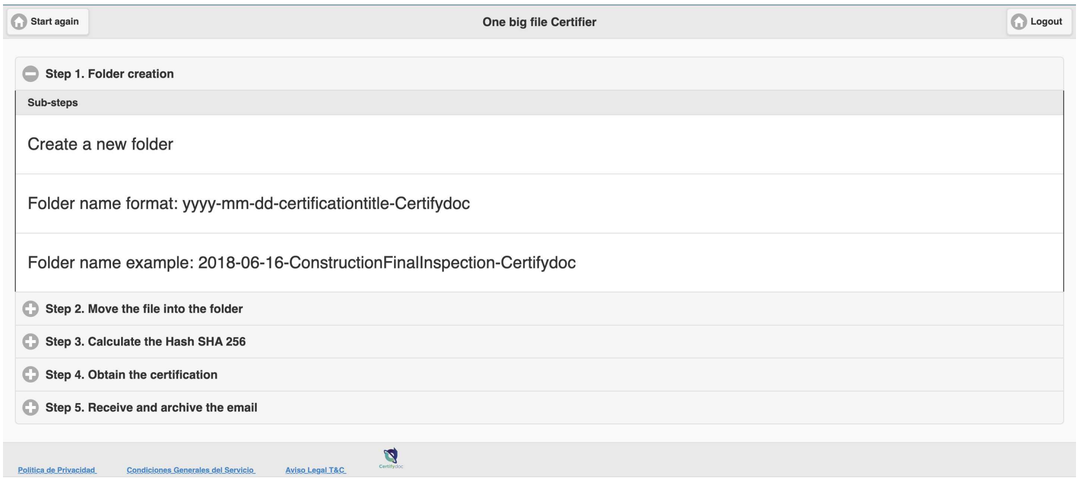
Your Name: Type your name without spaces

Your email: Your email..

Email subject: Choose an email subject to be sent to yourself

Package to select: ( < 17MB) Ninguno archivo selec.

3. **Process for a single large file:** You can upload the sha256 hash of a single file or file system portion without size limitation. This file can be a ZIP or an entire hard drive, so for practical purposes you can certify multiple files.



One big file Certifier

Step 1. Folder creation

Sub-steps

Create a new folder

Folder name format: yyyy-mm-dd-certificationtitle-Certifydoc

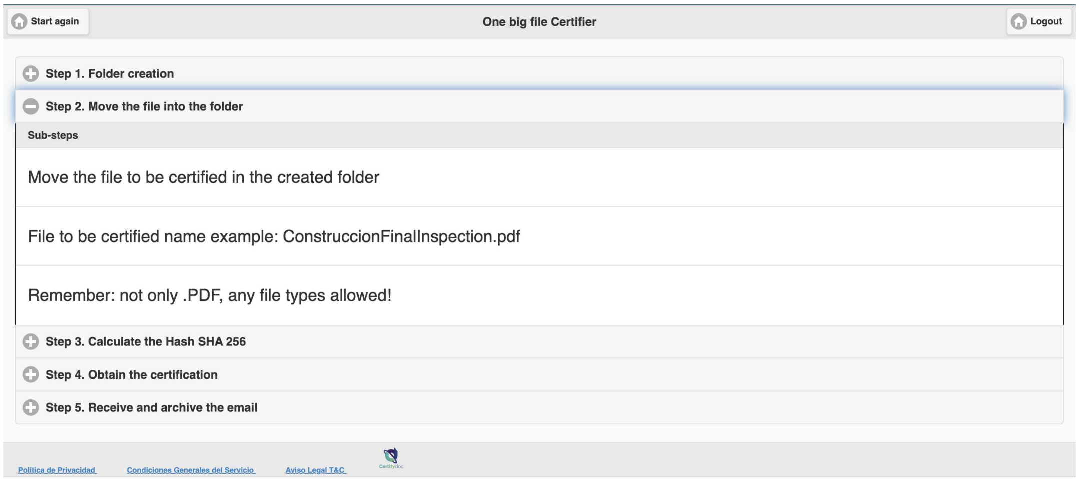
Folder name example: 2018-06-16-ConstructionFinalInspection-Certifydoc

Step 2. Move the file into the folder

Step 3. Calculate the Hash SHA 256

Step 4. Obtain the certification

Step 5. Receive and archive the email



One big file Certifier

Step 1. Folder creation

Step 2. Move the file into the folder

Sub-steps

Move the file to be certified in the created folder

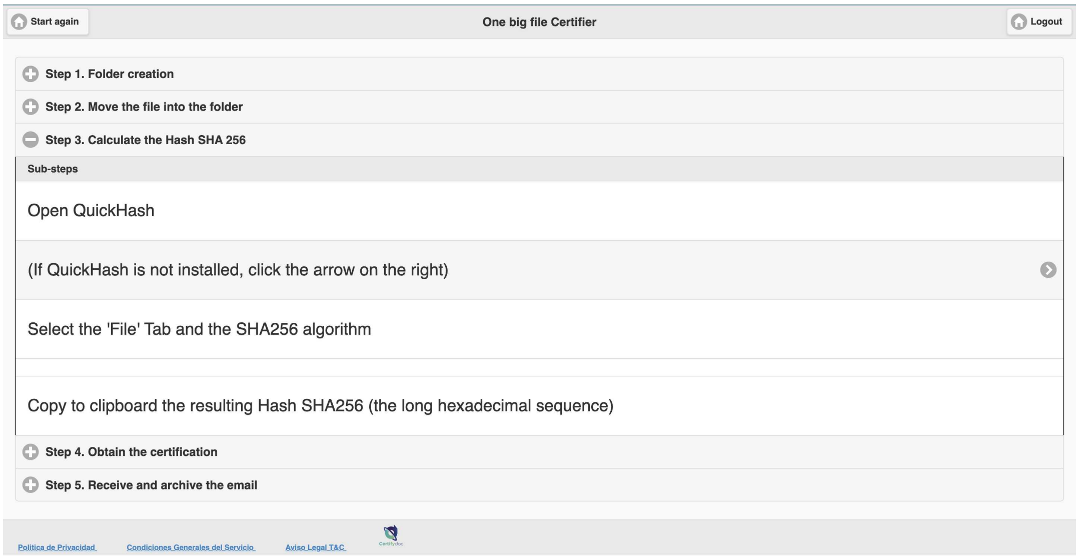
File to be certified name example: ConstruccionFinalInspection.pdf

Remember: not only .PDF, any file types allowed!

Step 3. Calculate the Hash SHA 256

Step 4. Obtain the certification

Step 5. Receive and archive the email



One big file Certifier

Step 1. Folder creation

Step 2. Move the file into the folder

Step 3. Calculate the Hash SHA 256

Sub-steps

Open QuickHash

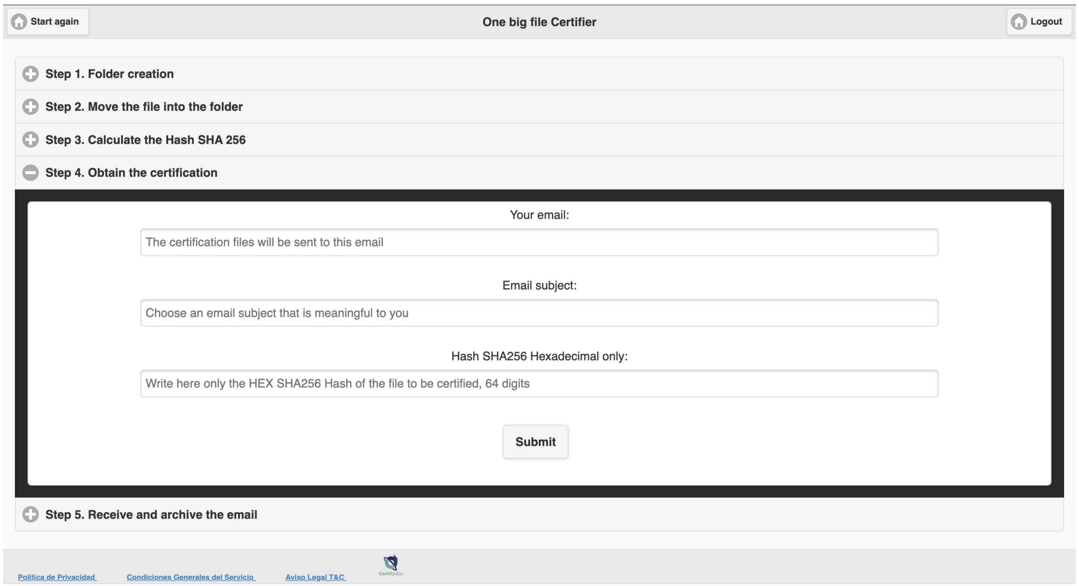
(If QuickHash is not installed, click the arrow on the right)

Select the 'File' Tab and the SHA256 algorithm

Copy to clipboard the resulting Hash SHA256 (the long hexadecimal sequence)

Step 4. Obtain the certification

Step 5. Receive and archive the email



Step 1. Folder creation

Step 2. Move the file into the folder

Step 3. Calculate the Hash SHA 256

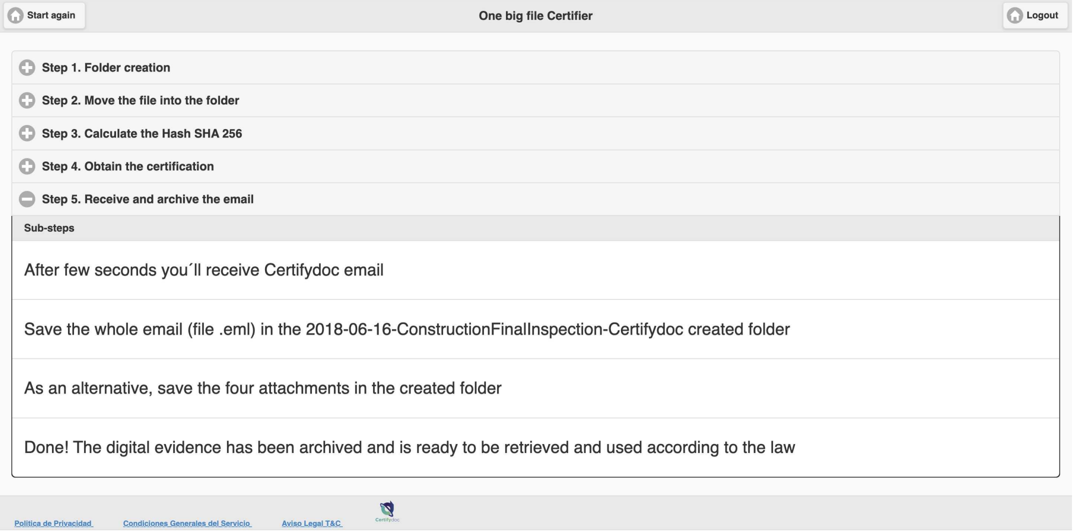
Step 4. Obtain the certification

Your email: The certification files will be sent to this email

Email subject: Choose an email subject that is meaningful to you

Hash SHA256 Hexadecimal only: Write here only the HEX SHA256 Hash of the file to be certified, 64 digits

Step 5. Receive and archive the email



Step 1. Folder creation

Step 2. Move the file into the folder

Step 3. Calculate the Hash SHA 256

Step 4. Obtain the certification

Step 5. Receive and archive the email

Sub-steps

After few seconds you´ll receive Certifydoc email

Save the whole email (file .eml) in the 2018-06-16-ConstructionFinalInspection-Certifydoc created folder

As an alternative, save the four attachments in the created folder

Done! The digital evidence has been archived and is ready to be retrieved and used according to the law

### 6.2. File certification via API

We can also certify files using the API. There are 4 possible processes (they are the same as in Blockchain Legalization Engine):

1. **Email response:** The API will send us an email with the files. This process has the same result as doing the certification via the web.
2. **API response complete:** We receive the same files as in the email but as a response from the API. The files are base64 encoded.
3. **API response reduced:** We receive the response through the API. It sends us the Timestamp Response (.tsr), the CA certificate (.pem) and the certification report (.pdf). The files are base64 encoded.
4. **API response minimized:** We receive the response through the API. It sends us the Timestamp Response (.tsr) and the CA certificate (.pem). The files are base64 encoded.

### 6.3. Verify file certification

When verifying the file certificate, what we are verifying is if the Timestamp Response (.tsr) contains the SHA256 hash of the file to be verified and if it has been signed by a certifying authority. We have two possibilities:

1. If we have only certified one file, the Timestamp Response contains SHA256 hash of the file.

   ```
   openssl ts -verify -in <tsr_filename>.tsr -CAfile <pem_filename>.pem -data
   <file_to_verify>
   ```

2. If we have certified several files, the Timestamp Response contains the SHA256 hash of the ZIP of the files.

```
penssl ts -verify -in <tsr_filename>.tsr -CAfile <pem_filename>.pem -data <zip_to_verify>.zip
```

## 7. Conclusions

After an exhaustive analysis that involved repeated use of both the Web Application and the Certifydoc Blockchain Certification Engine API and file certification using a Black Box testing approach, consistent and satisfactory results have been obtained in all certifications carried out.

Through multiple interactions with the platform, all issued certifications have been verified without finding any significant drawbacks. This verification process involved the creation of timestamps with the hash of Ethereum transactions and all transactions from other blockchains with the same hexadecimal format, in addition to their subsequent certification using the tool provided by Certifydoc. As well as the same verification process for the file certification.

Both the Certifydoc Web Application and API have been observed to offer a smooth and efficient user experience. The user interface of the Web Application is intuitive and easy to navigate, facilitating the transaction certification process. Likewise, the API has proven to be robust and reliable, allowing programmatic interaction with the platform effectively.

During the certification verification process using OpenSSL, the authenticity and validity of all certifications issued by Certifydoc has been confirmed. The OpenSSL tool has successfully validated the generated timestamps, supporting the integrity of the time certification process implemented by the platform.

In summary, the results obtained indicate that the Certifydoc Blockchain Certification Engine (BLM) tool as well as the file certification process works correctly and fulfil its purpose of certifying Blockchain transactions and files reliably and accurately.

These findings are encouraging and support the effectiveness of the platform as a viable solution for time and integrity certification in the context of Ethereum transactions and transactions on other blockchains that feature the same hexadecimal format. Moreover the file certification process has been verified to function correctly.