TrustBeat

trustbeat.eu

Certificate of Timestamp Existence

Compliance report: proof that data existed at a point in time

WHAT THIS REPORT PROVES

1 **The data's hash is included in the Merkle tree**

Recomputing the 3-step RFC 6962 path from the data's hash gives exactly the stored Merkle root.

MATCH

Evidence row 1, Step 1

2 **The Merkle root has a qualified and valid timestamp**

An RFC 3161 timestamp token issued by SK ID Solutions AS - SK TIMESTAMPING UNIT 2026E at 24 Oct 2026 18:00:00 UTC covers exactly this Merkle root. EU DSS validation: signature valid, certificate trusted via the EU Trusted List and not revoked, qualified electronic time stamp.

VALIDATED

Evidence rows 2-7, Step 2

Together, 1 and 2 show that the data with this hash existed no later than 24 Oct 2026 18:00:00 UTC, confirmed by a qualified electronic time stamp.

The data in electronic form is identified only by its SHA-256 hash below (for example a document or file). TrustBeat received only this hash, never the data itself.

HASH OF THE DATA (SHA-256)

```
224abab3f23509f722a28c8a7e2c638a
00a5e76074e2068c76c64150ef408b6e
```

TIMESTAMP	Sat, 24 Oct 2026 18:00:00 UTC
AUTHORITY	SK ID Solutions AS - SK TIMESTAMPING UNIT 2026E
TSA SERIAL	5429318475239939898
CERTIFICATE ID	01M5QH0RKQF1TH8T6TAAYG3N6Y
MERKLE ROOT	aaf64da2d3b024c2eaf75b09202d1382 948f29f450ec1d50c6175550a91ddf2d
MERKLE ALGORITHM	RFC 6962 (rfc6962-sha256)
LONG-TERM STAMPS	None - initial RFC 3161 timestamp only
DESCRIPTION	To_Perasma_V9_EPUB_PDF.zip

Verify online:

https://trustbeat.eu/en/verify?id=01M5QH0RKQF1TH8T6TAAYG3N6Y

Generated 2026-10-24 18:06:41 UTC - trustbeat.eu

EVIDENCE

#	ITEM	EVIDENCE	RESULT
1	**The data's hash is in the timestamped batch**	3-step RFC 6962 path from 224abab3f23509f722a28c8a7e2c638a00a5e76074e2068c76c64150ef408b6e recomputes aaf64da2d3b024c2eaf75b09202d1382948f29f450ec1d50c6175550a91ddf2d, equal to the stored Merkle root	**MATCH** Step 1
2	**The timestamp covers that batch**	Message imprint in the token (SHA-256): aaf64da2d3b024c2eaf75b09202d1382948f29f450ec1d50c6175550a91ddf2d, equal to the Merkle root	**MATCH** Step 2 (inspect)
3	**Timestamp issued by**	SK TIMESTAMPING UNIT 2026E, SK ID Solutions AS (certificate issued by SK TSA CA 2023E, SK ID Solutions AS)	**RECORDED** Step 2 (inspect)
4	**Time of existence**	24 Oct 2026 18:00:00.000 UTC (genTime in the token); token serial 5429318475239939898 (0x4B58D06331245F3A)	**RECORDED** Step 2 (inspect)
5	**Signing certificate at that time**	valid 28 Feb 2026 to 28 Mar 2032; extended key usage: time stamping	**RECORDED** Step 2 (inspect)
6	**EU Trusted List status**	Qualified time-stamp service (QTST), status granted since 8 Jan 2026, SK ID Solutions AS, trusted list EE; covers the time of existence. EU Trusted List checked 24 Oct 2026	**RECORDED** eidas.ec.europa.eu (see Step 2)
7	**Token signature and qualification**	Signature valid, certificate chain trusted via the EU Trusted List and not revoked; a qualified electronic time stamp (QTSA). Validated by TrustBeat with EU DSS 6.2 on 24 Oct 2026 18:00 UTC.	**VALIDATED** Step 2 (independent re-check)

MATCH = recomputed by TrustBeat and equal. RECORDED = read from the timestamp token or the EU Trusted List. VALIDATED / NOT QUALIFIED / FAILED / INCONCLUSIVE = TrustBeat's validation of the token with EU DSS (not a qualified validation service); every result can be re-checked independently with Step 2. Evidence assembled 24 Oct 2026 18:06 UTC from stored data.

MERKLE PROOF - STEP BY STEP

3 proof step(s) from the anchored hash to the Merkle root. RFC 6962: start from SHA-256(0x00 || hash); left = SHA-256(0x01 || sibling || current), right = SHA-256(0x01 || current || sibling).

STEP 0 leaf = SHA-256(0x00 || hash)

right	`224abab3f23509f722a28c8a7e2c638a00a5e76074e2068c76c64150ef408b6e`
result	`23c8d611fe2c8deb0be6fd76bf3d9c27ef00cd68d672fb62e2fbfcad982b1fda`

STEP 1 SHA-256(0x01 || left || right)

left	`c9cac88c800b7110248905e9030773f718b7f3a23b8c4f648ce7c408f15db889`
right	`23c8d611fe2c8deb0be6fd76bf3d9c27ef00cd68d672fb62e2fbfcad982b1fda`
result	`4e103aead02c6409361928dbe3125d624f0463a81dc77cb9136bead1f111f845`

STEP 2 SHA-256(0x01 || left || right)

left	`4e103aead02c6409361928dbe3125d624f0463a81dc77cb9136bead1f111f845`
right	`dfcca52cab2470c9f7b6a47f3ee854c41d186059f9eacf60dc0c245995e32810`
result	`9f6f8c96b64fc66f647595e577941a102150e898896a7908d1b33d7577ff7710`

STEP 3 SHA-256(0x01 || left || right)

left	`9f6f8c96b64fc66f647595e577941a102150e898896a7908d1b33d7577ff7710`
right	`ba88ef177ff5302cdf1bd9a7f104fdaa7f5cc1488964493f0eb6c089f7cc566d`
result	`aaf64da2d3b024c2eaf75b09202d1382948f29f450ec1d50c6175550a91ddf2d` = Merkle root

TIMESTAMP SIGNING CERTIFICATE

SUBJECT	CN=SK TIMESTAMPING UNIT 2026E,organizationIdentifier=NTREE-10747013,O=SK ID Solutions AS,C=EE
ISSUER	CN=SK TSA CA 2023E,organizationIdentifier=NTREE-10747013,O=SK ID Solutions AS,C=EE
SERIAL	`6903e291bc64290793eaa1295b6efb44`
VALID FROM	Sat, 28 Feb 2026 22:00:00 UTC
VALID UNTIL	Sun, 28 Mar 2032 21:59:59 UTC
SIGNATURE ALGORITHM	SHA256withECDSA

RFC 3161 TIMESTAMP TOKEN (DER, base64)

Copy all lines below into a file named token.b64 (used in Verification Instructions, Step 2):


```
MIIHFAYJKoZIhvcNAQcCoIIHBTCCBwECAQMxDTALBglghkgBZQMEAgMwggEH
BgsqhkiG9w0BCRABBKCB9wSB9DCB8QIBAQYGBACPZwEBMC8wCwYJYIZIAWUD
BAIBBCCq9k2i07Akwur3WwkgLROClI8p9FDsHVDGF1VQqR3fLQIIS1jQYzEk
XzoYDzIwMjYxMDI0MTgwMDAwWjADAgEBAghI+6YetV+Ld6BspGowaDEjMCEG
A1UEAwwaU0sgVElNRVNUQU1QSU5HIFVOSVQgMjAyNkUxFzAVBgNVBGEMDk5U
UkVFLTEwNzQ3MDEzMRswGQYDVQQKDBJTSyBJRCBTb2x1dGlvbnMgQVMxCzAJ
BgNVBAYTAkVFoRswGQYIKwYBBQUHAQMEDTALMAkGBwQAgZdeAQGgggOSMIID
jjCCAxOgAwIBAgIQaQPikbxkKQeT6qEpW277RDAKBggqhkjOPQQDAjBdMRgw
FgYDVQQDDA9TSyBUU0EgQ0EgMjAyM0UxFzAVBgNVBGEMDk5UUkVFLTEwNzQ3
MDEzMRswGQYDVQQKDBJTSyBJRCBTb2x1dGlvbnMgQVMxCzAJBgNVBAYTAkVF
MB4XDTI2MDIyODIyMDAwMFoXDTMyMDMyODIxNTk1OVowaDEjMCEGA1UEAwwa
```

U0sgVElNRVNUQU1QSU5HIFVOSVQgMjAyNkUxFzAVBgNVBGEMDk5UUkVFLTEw
NzQ3MDEzMRswGQYDVQQKDBJTSyBJRCBTb2x1dGlvbnMgQVMxCzAJBgNVBAYT
AkVFMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEUob6bmUIA5qAvXRg5HOK
iBiY+N0MxAa6SUt9WJT9XANnOqOWeJ6BE+sqepcuJMEnRrpVRVa0ClaiGXgc
nTuaE6OCAagwggGkMB8GA1UdIwQYMBaAFFoYwDTO17eZ3wiQwAR9GPNzlr3P
MGQGCCsGAQUFBwEBBFgwVjAyBggrBgEFBQcwAoYmaHR0cDovL2Muc2suZWUv
U0tfVFNBX0NBXzIwMjNFLmRlci5jcnQwIAYIKwYBBQUHMAGGFGh0dHA6Ly9h
aWEuc2suZWUvdHNhMIGeBgNVHSAEgZYwgZMwgZAGBgQAj3oBAjCBhTA7Bggr
BgEFBQcCARYvaHR0cHM6Ly93d3cuc2tpZHNvbHV0aW9ucy5ldS9lbi9yZXBv
c2l0b3J5L3RzYS8wRgYIKwYBBQUHAgIwOgw4VFNVIGNlcnRpZmljYXRlIGhh
cyBiZWVuIGlzc3VlZCBhY2NvcmRpbmcgdG8gTkNQKyBwb2xpY3kwFgYDVR0l
AQH/BAwwCgYIKwYBBQUHAwgwMwYDVR0fBCwwKjAooCagJIYiaHR0cDovL2Mu
c2suZWUvc2tfdHNhX2NhXzIwMjNlLmNybDAdBgNVHQ4EFgQULU3mqsNWp2Nl
ZQNGsrN5ofbfRm0wDgYDVR0PAQH/BAQDAgbAMAoGCCqGSM49BAMCA2kAMGYC
MQCDG4x90SPFkCQVlN4rYc/gKurei/1x+2PlQmgMVM464AVfMTVP80vzRAba
JNarQ7MCMQD5ctzTFF4xQdjxneYUD+o7aaKlzeRAgDqcJ6TG4vTA4etxhRgt
FQGD9UoM4Akqf3MxggJKMIICRgIBATBxMF0xGDAWBgNVBAMMD1NLIFRTQSBD
QSAyMDIzRTEXMBUGA1UEYQwOTlRSRUUtMTA3NDcwMTMxGzAZBgNVBAoMElNL
IElEIFNvbHV0aW9ucyBBUzELMAkGA1UEBhMCRUUCEGkD4pG8ZCkHk+qhKVtu
+0QwCwYJYIZIAWUDBAIDoIIBajAaBgkqhkiG9w0BCQMxDQYLKoZIhvcNAQkQ
AQQwHAYJKoZIhvcNAQkFMQ8XDTI2MTAyNDE4MDAwMFowKAYJKoZIhvcNAQk0
MRswGTALBglghkgBZQMEAgOhCgYIKoZIzj0EAwQwTwYJKoZIhvcNAQkEMUIE
QOMz6Ag0WiIyK+1yGidm3IMydMkPOZVGead2Fc4vF+d8MStwpmF7lDKiNQHD
h/rkmOEzdSETGKeEYyFw+t8L7fEwgbIGCyqGSIb3DQEJEAIvMYGiMIGfMIGc
MIGZBCCG2H0kZFOj7c5Zvo6cZ51fwNMNJHy1dmbvtOxZ0xaR6zB1MGGkXzBd
MRgwFgYDVQQDDA9TSyBUU0EgQ0EgMjAyM0UxFzAVBgNVBGEMDk5UUkVFLTEw
NzQ3MDEzMRswGQYDVQQKDBJTSyBJRCBTb2x1dGlvbnMgQVMxCzAJBgNVBAYT
AkVFAhBpA+KRvGQpB5PqoSlbbvtEMAoGCCqGSM49BAMEBEcwRQIhALNqByAS
+7DaJ+eHaQ0G/fKCuqsBmxcpOrX4F18bU4bYAiB74q61uzSgbcIvqcduAq0Y
OFRe3Q2TdgVrMjTQisQ0jg==

VERIFICATION INSTRUCTIONS

Full guide: https://trustbeat.eu/en/manual-verify

Step 1 - Re-derive the Merkle root (Python 3)

This proof uses RFC 6962 (merkle_algorithm = rfc6962-sha256). Hash the data's hash once with a 0x00 prefix to get the leaf, then fold each Merkle Proof Path step in order: SHA-256 of 0x01, then the two values in the order given by the side. The result must equal the Merkle Root. Run the script below with Python 3.

```
import hashlib
H = lambda b: hashlib.sha256(b).digest()
path = [  # (sibling, side) - the Merkle Proof Path rows above
    ('c9cac88c800b7110248905e9030773f718b7f3a23b8c4f648ce7c408f15db889', 'left'),
    ('dfcca52cab2470c9f7b6a47f3ee854c41d186059f9eacf60dc0c245995e32810', 'right'),
    ('ba88ef177ff5302cdf1bd9a7f104fdaa7f5cc1488964493f0eb6c089f7cc566d', 'right'),
]
node = H(b'\x00' + bytes.fromhex('224abab3f23509f722a28c8a7e2c638a00a5e76074e2068c76c64150ef408b6e'))
print('step 0', node.hex())  # compare with the step table
for i, (sibling, side) in enumerate(path, 1):
    s = bytes.fromhex(sibling)
    node = H(b'\x01' + (s + node if side == 'left' else node + s))
    print('step', i, node.hex())
assert node.hex() == 'aaf64da2d3b024c2eaf75b09202d1382948f29f450ec1d50c6175550a91ddf2d', 'MISMATCH'
print('OK - Merkle root matches')
```

Step 2 - Verify the RFC 3161 timestamp token

This step re-checks, independently of TrustBeat, what row 7 reports. Copy all lines of the RFC 3161 Timestamp Token section above into a file named token.b64 (page headers may be copied too; the script below ignores them). Download the certificate of the time-stamp service "SK ID Solutions AS - SK TIMESTAMPING UNIT 2026E" from the EU Trusted List (eidas.ec.europa.eu/efda/tl-browser: find the service, open its service digital identity, download the certificate) and save it as tsa-service-certificate.pem. This is the time-stamp service's own certificate, not a root certificate: the EU Trusted List vouches for the service directly, which is why the verify command uses -partial_chain. Its SHA-256 fingerprint must be 86:D8:7D:24:64:53:A3:ED:CE:59:BE:8E:9C:67:9D:5F:C0:D3:0D:24:7C:B5:76:66:EF:B4:EC:59:D3:16:91:EB (printed in VERIFICATION DATA). Never use a certificate copied from this report or from the token as the trust anchor. The inspect command must show the TSA Serial (openssl prints it in hex: 0x4B58D06331245F3A) and a Message data value equal to the Merkle Root; the verify command must print "Verification: OK".

```
# Decode token.b64 to DER (lines that are not base64, such as page headers, are ignored):
python3 - <<'PY'
import base64, re
b64 = ''.join(re.findall(r'^[A-Za-z0-9+/=]+$', open('token.b64').read(), re.M))
open('tok.der', 'wb').write(base64.b64decode(b64))
PY
# Inspect: Serial number = TSA Serial, Message data = Merkle Root
openssl ts -reply -token_in -in tok.der -text
# The trust anchor's fingerprint must equal the printed SHA-256 fingerprint:
openssl x509 -in tsa-service-certificate.pem -noout -fingerprint -sha256
# Verify the signature and that the token covers the Merkle Root:
ROOT='aaf64da2d3b024c2eaf75b09202d1382948f29f450ec1d50c6175550a91ddf2d'
openssl ts -verify -token_in -in tok.der -digest "$ROOT" -CAfile tsa-service-certificate.pem -partial_chain
```

VERIFICATION DATA

Every value needed to repeat this report's checks without TrustBeat, printed in full. The small grey line under each label says where the value comes from. The signing certificate is printed so you can recognise the genuine one: the trust anchor itself must come from the EU Trusted List or the issuer, never from this report. The data itself is not included: compute its SHA-256 hash yourself and compare it with the anchored hash.

PROOF

Record ID TrustBeat record	01M5QH0RKQF1TH8T6TAAYG3N6Y
Anchored hash (leaf entry) TrustBeat record	224abab3f23509f722a28c8a7e2c638a00a5e76074e2068c76c64150ef408b6e
Hash algorithm TrustBeat record	SHA-256
Merkle algorithm TrustBeat record	rfc6962-sha256
Leaf index TrustBeat record	1
Tree size (leaves in batch) TrustBeat record	7
Batch ID TrustBeat record	01M5QH3KSNTN6MKCAXMH59WNYY
Merkle root TrustBeat record	aaf64da2d3b024c2eaf75b09202d1382948f29f450ec1d50c6175550a91ddf2d
Merkle root, recomputed recomputed by TrustBeat	aaf64da2d3b024c2eaf75b09202d1382948f29f450ec1d50c6175550a91ddf2d
Proof path TrustBeat record	3 step(s): see MERKLE PROOF - STEP BY STEP

TIMESTAMP TOKEN

genTime timestamp token (signed)	2026-10-24T18:00:00Z
Serial number timestamp token (signed)	5429318475239939898 (hex 0x4B58D06331245F3A)
Policy OID timestamp token (signed)	0.4.0.2023.1.1
Message imprint algorithm timestamp token (signed)	SHA-256 (2.16.840.1.101.3.4.2.1)
Message imprint timestamp token (signed)	aaf64da2d3b024c2eaf75b09202d1382948f29f450ec1d50c6175550a91ddf2d
Nonce timestamp token (signed)	0x48fba61eb55f8b77
TSA name field timestamp token (signed)	CN=SK TIMESTAMPING UNIT 2026E,organizationIdentifier=NTREE-10747013,O=SK ID Solutions AS,C=EE
Token size / SHA-256 recomputed by TrustBeat	1816 bytes / 9cc5803f40570aeec68b4d65ebebb4336aeb28396636708a4ff16c289ea31062

SIGNING CERTIFICATE

Subject certificate in the token	CN=SK TIMESTAMPING UNIT 2026E,organizationIdentifier=NTREE-10747013,O=SK ID Solutions AS,C=EE
Issuer certificate in the token	CN=SK TSA CA 2023E,organizationIdentifier=NTREE-10747013,O=SK ID Solutions AS,C=EE

Serial number (hex) certificate in the token	6903e291bc64290793eaa1295b6efb44
Valid from certificate in the token	2026-02-28T22:00:00Z
Valid until certificate in the token	2032-03-28T21:59:59Z
Extended key usage certificate in the token	1.3.6.1.5.5.7.3.8
SHA-256 fingerprint (DER) recomputed by TrustBeat	86d87d246453a3edce59be8e9c679d5fc0d30d247cb57666efb4ec59d31691eb
SHA-256 fingerprint, as openssl prints it recomputed by TrustBeat	86:D8:7D:24:64:53:A3:ED:CE:59:BE:8E:9C:67:9D:5F:C0:D3:0D:24:7C:B5:76:66:EF:B4:EC:59:D3:16:91:EB

EU TRUSTED LIST

Looked up TrustBeat record	2026-10-24T17:54:36.198472Z
Standing at genTime recomputed by TrustBeat	qualified time-stamp service (QTST) granted
Service 1 EU Trusted List (as recorded)	SK Time-Stamping Authority for qualified electronic time stamps (SK-TSA 2026E) / SK ID Solutions AS / list EE
Service 1 type EU Trusted List (as recorded)	http://uri.etsi.org/TrstSvc/Svctype/TSA/QTST
Service 1 status EU Trusted List (as recorded)	http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/granted from 2026-01-08T01:00:00Z
Service 1 list URL EU Trusted List (as recorded)	https://sr.riik.ee/tsl/estonian-tsl.xml

TOKEN VALIDATION (EU DSS)

Validated at TrustBeat record	2026-10-24T18:00:36.273333Z
Validator TrustBeat record	EU DSS 6.2
Indication recomputed by TrustBeat	PASSED
Timestamp qualification recomputed by TrustBeat	QTSA
Errors recomputed by TrustBeat	none
Warnings recomputed by TrustBeat	none
DSS simple report SHA-256 TrustBeat record	ed2514bbeabaa0fc0ebc0fbaf271078c15dff224ab7f366bccb2ea67bdd0fd4b (the report is printed in full below)

Signing certificate (PEM, copied from the token)

SHA-256 fingerprint 86:D8:7D:24:64:53:A3:ED:CE:59:BE:8E:9C:67:9D:5F:C0:D3:0D:24:7C:B5:76:66:EF:B4:EC:59:D3:16:91:EB. Use it to recognise the genuine certificate you obtain from the EU Trusted List; do not use this printed copy as the trust anchor.


```
-----BEGIN CERTIFICATE-----
MIIDjjCCAxOgAwIBAgIQaQPikbxkKQeT6qEpW277RDAKBggqhkjOPQQDAjBdMRgw
FgYDVQQDDA9TSyBUU0EgQ0EgMjAyM0UxFzAVBgNVBGEMDk5UUkVFLTEwNzQ3MDEz
MRswGQYDVQQKDBJTSyBJRCBTb2x1dGlvbnMgQVMxCzAJBgNVBAYTAkVFMB4XDTI2
MDIyODIyMDAwMFoXDTMyMDMyODIxNTk1OVowaDEjMCEGA1UEAwwaU0sgVElNRVNU
QU1QSU5HIFVOSVQgMjAyNkUxFzAVBgNVBGEMDk5UUkVFLTEwNzQ3MDEzMRswGQYD
VQQKDBJTSyBJRCBTb2x1dGlvbnMgQVMxCzAJBgNVBAYTAkVFMFkwEwYHKoZIzj0C
AQYIKoZIzj0DAQcDQgAEUob6bmUIA5qAvXRg5HOKiBiY+N0MxAa6SUt9WJT9XANn
OqOWeJ6BE+sqepcuJMEnRrpVRVa0ClaiGXgcnTuaE6OCAagwggGkMB8GA1UdIwQY
MBaAFFoYwDTO17eZ3wiQwAR9GPNzlr3PMGQGCCsGAQUFBwEBBFgwVjAyBggrBgEF
BQcwAoYmaHR0cDovL2Muc2suZWUvU0tfVFNBX0NBXzIwMjNFLmRlci5jcnQwIAYI
KwYBBQUHMAGGFGh0dHA6Ly9haWEuc2suZWUvdHNhMIGeBgNVHSAEgZYwgZMwgZAG
BgQAj3oBAjCBhTA7BggrBgEFBQcCARYvaHR0cHM6Ly93d3cuc2tpZHNvbHV0aW9u
cy5ldS9lbi9yZXBvc2l0b3J5L3RzYS8wRgYIKwYBBQUHAgIwOgw4VFNVIGNlcnRp
ZmljYXRlIGhhcyBiZWVuIGlzc3VlZCBhY2NvcmRpbmcgdG8gTkNQKyBwb2xpY3kw
FgYDVR0lAQH/BAwwCgYIKwYBBQUHAwgwMwYDVR0fBCwwKjAooCagJIYiaHR0cDov
L2Muc2suZWUvc2tfdHNhX2NhXzIwMjNlLmNybDAdBgNVHQ4EFgQULU3mqsNWp2Nl
ZQNGsrN5ofbfRm0wDgYDVR0PAQH/BAQDAgbAMAoGCCqGSM49BAMCA2kAMGYCMQCD
G4x90SPFkCQVlN4rYc/gKurei/1x+2PlQmgMVM464AVfMTVP80vzRAbaJNarQ7MC
MQD5ctzTFF4xQdjxneYUD+o7aaKlzeRAgDqcJ6TG4vTA4etxhRgtFQGD9UoM4Akq
f3M=
-----END CERTIFICATE-----
```

EU DSS simple validation report (XML), EU DSS 6.2

The report TrustBeat stored when it validated the token; SHA-256 ed2514bbeabaa0fc0ebc0fbaf271078c15dff224ab7f366bccb2ea67bdd0fd4b. Printed for reading: tabs are shown as four spaces and lines longer than 110 characters continue, indented, on the next line.


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<SimpleReport ValidationTime="2026-10-24T18:00:36Z"
    xmlns="http://dss.esig.europa.eu/validation/simple-report">
    <ValidationPolicy>
        <PolicyName>QES AdESQC TL based</PolicyName>
        <PolicyDescription>Validates electronic signatures and indicates whether they are Advanced electronic
            Signatures (AdES), AdES supported by a Qualified Certificate (AdES/QC) or a
        Qualified electronic Signature (QES). All certificates and their related chains supporting the
            signatures are validated against the EU Member State Trusted Lists (this includes
        signer's certificate and certificates used to validate certificate validity status services - CRLs,
            OCSP, and time-stamps).
    </PolicyDescription>
    </ValidationPolicy>
    <DocumentName></DocumentName>
    <ValidSignaturesCount>0</ValidSignaturesCount>
    <SignaturesCount>0</SignaturesCount>
    <Timestamp Id="T-9CC5803F40570AEEC68B4D65EBEBB4336AEB28396636708A4FF16C289EA31062">
        <CertificateChain>
            <Certificate Id="C-86D87D246453A3EDCE59BE8E9C679D5FC0D30D247CB57666EFB4EC59D31691EB"
                trusted="true">
                <QualifiedName>SK TIMESTAMPING UNIT 2026E</QualifiedName>
                <TrustAnchors>
                    <TrustAnchor countryCode="EE">
                        <TSLType>http://uri.etsi.org/TrstSvc/TrustedList/TSLType/EUgeneric</TSLType>
                        <TrustServiceProvider>SK ID Solutions AS</TrustServiceProvider>
                        <TrustServiceProviderRegistrationId>VATEE-100687640</TrustServiceProviderRegistrationI
                            d>
                        <TrustServiceName>SK Time-Stamping Authority for qualified electronic time stamps
                            (SK-TSA 2026E)</TrustServiceName>
                    </TrustAnchor>
                </TrustAnchors>
            </Certificate>
            <Certificate Id="C-F3A6FC612A6912E3ED971F527AEC31EACCDEEB0483604772486D63B602E47DB6">
                <QualifiedName>SK TSA CA 2023E</QualifiedName>
            </Certificate>
            <Certificate Id="C-A272DE102A8038F7E100EACFB6DB1C930D3CB9F51D76B218DE3A4A433BCFD182">
                <QualifiedName>SK ID Solutions ROOT G1E</QualifiedName>
            </Certificate>
        </CertificateChain>
        <Indication>PASSED</Indication>
        <ProductionTime>2026-10-24T18:00:00Z</ProductionTime>
        <ProducedBy>SK TIMESTAMPING UNIT 2026E</ProducedBy>
        <TimestampLevel description="Qualified timestamp">QTSA</TimestampLevel>
        <TimestampScope Id="D-1C54B05A003D206EEC6466ECBE5B3335A7D756B7DC04659A1F94DA510A5DF3CF" name="Digest
            document" scope="DIGEST">Digest of the document content</TimestampScope>
    </Timestamp>
</SimpleReport>
```