IIC TSA — Practice Statement

Publication notice. This document is published by HTTPS CARD — INTERNET IDENTITY CARD LIMITED (England and Wales, company no. 09168431) for the IIC TSA time-stamping service at `https://tsa.internetidentitycard.com/tsa`. IIC TSA is designed to RFC 3161 / RFC 5816. It is **not** a qualified trust service under eIDAS or UK eIDAS, and it is not certified, accredited or recognised by any authority. IIC TSA and VerifBox (verifbox.com) belong to the same company: for VerifBox users, IIC TSA is not an independent third party. Use of the service is subject to the IIC TSA terms of use.

Version 1.0 — approved 10 October 2026; erratum 1 of 11 October 2026 (retention lock deferred; public issuance closed); erratum 2 of 11 October 2026 (section 10: revocation of unexpired TSA certificates at termination made explicit). Owner approval was given by launching the production service on 10 October 2026; the service's policy.json reports status "approved". Records the production ceremony, the production services, the production acceptance (40 of 40 checks) and the VerifBox integration of 10 October 2026, on the simplified design approved by the owner on 9 October 2026. Supersedes draft 9.

Status: version 1.0, approved 10 October 2026. This statement describes how IIC TSA meets the Timestamp Policy v1 and IIC TSA — Technical Specification v1.0; IIC TSA — Risk Register v1.1 assesses its risks. It distinguishes two kinds of statement: **decided** practices, which are binding, and **target** practices, marked as such, which describe the intended design and become binding only when the matching item of section 13 is closed. It follows the structure of ETSI EN 319 421 as a drafting grid only, without claiming conformity, audit or qualification. The English version is authoritative for technical interpretation.

1. Purpose and identification

The Timestamp Policy states what IIC TSA promises; this Practice Statement states how those promises are kept in operation. If the two ever differ, the policy prevails and this statement is corrected. IIC TSA is a trade name of the operator, which runs it within its Internet Identity Card business line and holds PEN 67100; the OIDs used by the TSA are listed in the operator's OID registry.

Same company as VerifBox. VerifBox (verifbox.com), a timestamping website of the same company, is a client of IIC TSA. IIC TSA is therefore not an independent third party for VerifBox's users, and VerifBox discloses this to them before the first proof carrying an IIC TSA token.

Item	Value
Operator	HTTPS CARD - INTERNET IDENTITY CARD LIMITED, company no. 09168431, 124 City Road, London EC1V 2NX, United Kingdom
Service	Internet Identity Card Time Stamp Authority (IIC TSA), address `https://tsa.internetidentitycard.com/tsa`
Policy implemented	IIC TSA Timestamp Policy v1, OID `1.3.6.1.4.1.67100.1.1.1.1` (OID registry of PEN 67100)
Statement version	1.0 (approved 10 October 2026)
Contact	an address on internetidentitycard.com, to be created

2. Architecture

Diagram (interactive version in the source document): issuance path — client → HTTPS → Cloud Run service → Cloud KMS key tsa-2026-a; root IIC TSA Root R1 kept offline, used only for the CRL and to certify a TSA key.

The signer is the only process that can call the signing key, and it does so only while the time guardian holds a valid lease. The root is offline and acts only during ceremonies; it certifies the TSA certificate directly, with no issuing CA. The drawing shows the design approved on 9 October 2026; the items still open are listed in section 13. In production, two Cloud Run services run the same image with the same key and journal (section 9). The time guardian measures over TLS (section 6).

3. Roles and compensating controls

IIC TSA has a **single operator**: the owner, Michael Benaudis, holds every role below. There is no separation of duties and no two-person control; this is stated openly and accepted as a residual risk (section 10). Compensating controls: the Cloud KMS Data Access audit logs record every use of the signing key; each service account holds only the rights it needs; the issuance journal sits under a locked retention policy; and each day's journal batch is time-stamped and anchored by third parties (section 8).

Role	Responsibilities	Held by
TSA Manager	approves this statement and the policy; authorises ceremonies, key rotation and service termination	the owner
Ceremony operator	performs ceremonies alone: root key generation, certification of a TSA key, CRL signing, revocation, destruction	the owner

Role	Responsibilities	Held by
Module Administrator	manages key rings, IAM bindings and audit log settings of Cloud KMS; holds no signer role on the key	the owner
Signing service (software identity, not a person)	calls the signing operation on the current TSA key, under a valid time lease; cannot change keys, IAM or logs	its own service account

Access rights are reviewed at every ceremony and at least once a year.

4. Token signing key

Decided: a Google Cloud KMS key at protection level **SOFTWARE**, not a hardware security module. The signing service calls the Cloud KMS asymmetric-signing API; the private key is generated and kept inside Cloud KMS and is never exposed to the service. The cloud provider operates Cloud KMS: exclusive custody of the key is not claimed, an accepted residual risk (section 10). The key costs about USD 0.06 a month. Moving to HSM protection later is possible if a client requires it; it means a new key and a new certificate.

Aspect	Practice
Key	purpose ASYMMETRIC_SIGN, algorithm EC_SIGN_P256_SHA256 (decided by the policy), protection level SOFTWARE, one key per generation holding a single version (first production generation tsa-2026-a), in a key ring of the production project; region europe-west1
Generation	inside Cloud KMS, done on 10 October 2026 when the production project iic-tsa-prod-67100 was set up (key tsa-2026-a, public key SHA-256 3d11b2af1f9a52d960e79c5aae3553c087a8627e32628a0ab51c0cb83a8a0d02); at the production ceremony of 10 October 2026 (section 5), the TSA certificate was built from the public key read from Cloud KMS, and no certificate request was signed with the key
Export	not possible: Cloud KMS offers no export or wrapping of the private key; no hardware key attestation exists at SOFTWARE level and none is claimed

Aspect	Practice
Access	only the signing service account holds the signer and public-key viewer roles, granted on the key of the current generation only; the Module Administrator holds administration rights but no signer role; no human holds a signer role in production; every use of the key is recorded in the Cloud KMS Data Access audit logs
Authentication	the signing service runs under a workload identity: there is no key file, PIN or credential in code, configuration or environment variables
Signing call	the service computes SHA-256 over the DER-encoded signed attributes and passes it in the request's `digest` field, which Cloud KMS signs as given. The `data` field is never used: the service signs exactly the digest it computed, and a digest placed in that field would be hashed a second time. Before signing, the service checks that the Cloud KMS public key equals the certificate's and that the protection level is SOFTWARE. Request and response integrity checksums (CRC32C of digest and signature) are verified, the signature is normalised to low-S and verified locally before release, and every token is checked by OpenSSL in the acceptance tests (A3)
End of a generation	the key version signs for at most 15 months after the TSA certificate's notBefore date; it is then disabled and scheduled for destruction by the operator (Cloud KMS destroy delay: 30 days in production, 1 day in test), and destroyed within 18 months of the certificate's issuance; dates are recorded and published
Continuity	there is no backup copy of the key outside Cloud KMS; if the key becomes unusable, issuance stops and a new generation is created: a new Cloud KMS key and a new TSA certificate signed by the root

Test keys live in a separate test project (iic-tsa-test-67100), carry distinct names and the distinct test policy 1.3.6.1.4.1.67100.1.1.9.1, and are never served by the production endpoint.

5. PKI and key ceremonies

The hierarchy has two levels, each with one job: the root signs only TSA certificates and its own CRL; the TSA key signs only tokens. There is no issuing CA.

Level	Where the key lives	Algorithm	Validity and profile
Root: "IIC TSA Root R1" (test: "IIC TSA TEST Root R1")	software key, PKCS#8 encrypted with scrypt and AES-256-CBC, used offline on the owner's computer only	ECDSA P-384, signed with SHA-384 (decided)	30 years minus 1 second; basicConstraints critical CA:TRUE, pathlen:0; keyUsage critical keyCertSign, cRLSign
TSA certificate: "IIC TSA tsa-2026-a" (test: "IIC TSA TEST tsa-test-2026-a"), issued directly by the root	Cloud KMS, production project (section 4)	ECDSA P-256, SHA-256 (decided)	135 months minus 1 second; key used for 15 months at most (decided, D2); basicConstraints critical CA:FALSE; keyUsage critical digitalSignature; extendedKeyUsage critical, timeStamping only

Both subjects carry O=HTTPS CARD - INTERNET IDENTITY CARD LIMITED and C=GB, and a subject key identifier computed by RFC 7093 method 1 (leftmost 160 bits of the SHA-256 of the public key). The TSA certificate also carries the authority key identifier and a CRL distribution point, `https://tsa.internetidentitycard.com/tsa/crl` (test: the private run.app address of the test service).

The test hierarchy was created and accepted on 10 October 2026. The production ceremony was performed the same day, offline on the owner's iMac (Docker, Debian 13, OpenSSL 3.5, `--network none`), by the single operator: root IIC TSA Root R1 (P-384), valid 10 October 2026 to 9 October 2056; TSA certificate IIC TSA tsa-2026-a; first root CRL valid until 10 October 2027 17:29 UTC. The encrypted root key is on two USB keys of different brands, verified identical and each decryptable; the passphrase is on paper, kept apart; the working copy was deleted from the computer. Only root.pem, root.crl and the TSA certificate went to Cloud Shell.

Certificate	Test, SHA-256 (10 October 2026)	Production, SHA-256 (10 October 2026)
Root R1	BA:7C:3F:11:CC:23:8D:EC:D7:E1:9E:60:DE:88:6E:10:FA:CA:78:5E:BF:F2:CD:7A:53:2A:12:34:3B:29:DC:76	86:52:E3:D1:3E:72:5F:7C:75:7C:FA:05:3D:64:13:60:AC:A4:43:CF:6C:09:D7:D6:DD:D4:4B:EA:85:1E:D3:B9
TSA certificate	E8:DC:53:E3:36:24:7E:EB:6A:C3:0E:6E:78:6A:1E:D2:16:19:05:F3:7A:E2:E2:C7:63:35:C8:6A:DF:F9:E4:A6	E4:F9:8E:7A:ED:CD:1A:07:D8:1C:A5:9D:1C:19:92:AB:E7:85:7A:2C:A1:CC:8D:F6:9F:72:05:FF:AD:34:FB:E8

Custody of the root key. The root key is a software key, encrypted (PKCS#8, scrypt and AES-256-CBC) under a passphrase of at least 20 characters. The passphrase is written on paper and kept apart from the backups. The whole ceremony folder, encrypted key

included, is copied to two USB drives stored in two different places (done on 10 October 2026: two USB keys of different brands, verified identical and each decryptable; the working copy was then deleted from the computer). The root key is never placed in Cloud Shell, Git, a container image or any cloud service. It is used about once a year, to sign the CRL, and to certify a new TSA key.

Because the root is offline, its CRL is signed during a ceremony: DER-encoded, valid 365 days, re-issued every year at least 30 days before it expires. The first production root CRL, signed at the ceremony of 10 October 2026, is valid until 10 October 2027 17:29 UTC. An emergency revocation ceremony can be held at any time: on a compromise, an emergency CRL revokes the TSA certificate with reason keyCompromise and an invalidity date equal to the trust cut-off. The CRL is published at `/tsa/crl` and `/tsa/root-crl`; VerifBox publishes a copy at `verifbox.com/.well-known/iic-tsa-racine.crl`.

Every ceremony follows the same rules:

1. A fixed script (`ceremony/hierarchy.sh`), rehearsed in full on the test project before the first production run.
2. Offline, on the owner's own computer, with Wi-Fi and network cables off: the script runs in a Docker container built from `ceremony/Dockerfile` (Debian 13, OpenSSL 3.5), started with `--network none`. The image is built before the network is cut. A production ceremony never runs in Cloud Shell.
3. The operator performs the ceremony alone; there is no witness and no video recording.
4. The TSA certificate is checked against the profile of the policy by a tool the operator did not write, before it is published.
5. Any deviation stops the ceremony; it restarts from the beginning after the cause is fixed.
6. A written record lists date, time, each step, the two SHA-256 fingerprints of the root and TSA certificates, and the hashes of the CRLs produced, and is signed by the operator. Only the public files (root certificate, root CRL, TSA certificate) leave the ceremony computer.

The emergency revocation ceremony is rehearsed as well, not only the creation ceremony.

6. Time management

A time guardian inside the Cloud Run service decides whether tokens may be issued; it measures the system clock and never sets it. It works on demand: Cloud Run gives an instance no CPU between requests, so there is no background guardian (`TSA_BACKGROUND_GUARDIAN=0`). The first request after the lease expires triggers a measurement cycle, and the service keeps no instance at rest (minimum instances 0).

1. When the signer finds no valid lease, the guardian queries four sources over TLS: the PTB web clock (`wss://uhr.ptb.de/time`, milliseconds, traceable to UTC(PTB)), the traceable source; Akamai (`https://time.akamai.com/?iso&ms`, milliseconds); Google (`https://www.google.com/generate_204`) and Cloudflare (`https://www.cloudflare.com/cdn-`

`cgi/trace`), whose HTTP Date tick is bracketed between two requests. For each, it computes the offset of the system clock and its uncertainty. A source is usable if it answers within 2.5 seconds with an uncertainty of at most 100 ms.
2. It computes the median offset, the spread between sources and the number of usable sources.
3. It compares the wall clock with a monotonic clock at each cycle: a jump of more than one second is a clock step.
4. It sets the state: SYNC only if at least three sources are usable, PTB among them, the spread is at most 250 ms, the median offset within ±100 ms and no clock step is pending; FAIL otherwise. HOLDOVER is not used in v1. The thresholds are reviewed at day 14 of the measurement campaign.
5. In SYNC, it issues the signer a lease valid for at most 30 seconds, carrying the clock generation identifier. On entering FAIL, it withdraws the current lease at once and issues no other; the signer refuses to sign without a valid lease. No SYNC means no token: the request is rejected with `timeNotAvailable`.

A clock step requires manual readmission by the operator after review of the guardian log; any other FAIL ends when the conditions of SYNC are met again. The guardian logs every cycle: source, offset, uncertainty, state and state changes.

The hosting infrastructure smears leap seconds over 24 hours, while some public sources apply them as a step. During such an event the spread may exceed the threshold and the guardian moves to FAIL: issuance pauses rather than risk an inaccurate `genTime`. This is accepted and documented as a fail-closed behaviour.

Cloud Run blocks outbound UDP: a test on 10 October 2026 got no answer from PTB, NIST, Netnod, Cloudflare, Google NTP or the metadata server. NTP is therefore not used, and NTS (RFC 8915) is no longer a target on Cloud Run. NIST web time is not used, since NIST forbids its use from outside. PTB publishes no terms for automated use of its web clock; the volume is low (one query per measurement cycle).

Measured on the test service on 10 October 2026: offsets of a few milliseconds, PTB about 0–3 ms with an uncertainty of ±8–12 ms. At the production acceptance of 10 October 2026 (18:22 UTC) the guardian was in SYNC with four sources: PTB +0.9 ms ±8 ms, Akamai 0 ms ±4 ms, Google +3.5 ms ±33 ms, Cloudflare +2.6 ms ±44 ms; spread 3.5 ms. A 14-day measurement campaign runs on the test service every 10 minutes since 10 October 2026 and is reviewed at day 14 (share of SYNC cycles, 99th percentile of the median offset, 99th percentile of the spread). By owner decision of 10 October 2026, production opens without waiting for its end, because every token is already gated by a live measurement.

7. Issuance process

Each request goes through the same linear path; any step that fails ends with a rejection and no token. The front and the protocol engine perform steps 1 to 3; the signer, which holds the lease, performs steps 4 to 11 and reads genTime itself.

1. Receive the POST: content type `application/timestamp-query`, body of at most 8,192 bytes.
2. Decode strictly as DER; BER encoding (valid ASN.1, but outside this profile's strict DER requirement), including a default value written out such as a `certReq` FALSE, or trailing data is rejected (`badDataFormat`).
3. Check version, hash algorithm and length, and `reqPolicy` against the policy; reject any extension with `unacceptedExtension`.
4. Check that a valid time lease is held; if none, run a guardian cycle (section 6); without SYNC, reject with `timeNotAvailable`.
5. Draw the serial number: 128 bits from the operating system's cryptographically secure generator, kept positive.
6. Read `genTime`, rounded to the nearest whole UTC second.
7. Build the `TSTInfo` and the signed attributes, including `SigningCertificateV2`.
8. Check that the Cloud KMS public key equals the certificate's and that the protection level is SOFTWARE; have Cloud KMS sign the SHA-256 digest of the DER-encoded signed attributes; verify the CRC32C checksums of digest and signature; put the signature in low-S form, then verify it locally with the TSA certificate's public key.
9. Check that the lease is still valid. If it expired or was withdrawn during signing, or a check of step 8 failed, discard the token and record a "not issued" mark under that serial's name, if it is still free, with the reason lease_expired, sign_failed or check_failed.
10. Record the token: create its journal entry, named after the serial, with a create-only-if-absent condition. If an entry or a mark of that name already exists, it is a collision: discard the token and draw again from step 5 (at most three attempts, then `systemFailure`). If the journal cannot be written, discard the token and answer `systemFailure`.
11. Return the `TimeStampResp` as `application/timestamp-reply`.

The journal entry created at step 10 is the object `entries/<32 hex digits of the serial>` and holds only: serial, status, `gen_time`, key generation, clock state, clock generation, lease identifier, whether a nonce was present, and the SHA-256 of the token's signature. It holds no client data: no document, no file name, no submitted hash value, no nonce value, no IP address. A token is returned only after its entry is written. Entries and marks cannot be modified or deleted during the retention period (section 8), and both are named after the serial alone, so a serial, once used or marked, is never used again.

8. Logs, retention and audit trail

Every record the TSA keeps is append-only and protected against deletion for its whole retention period; none contains a hash value from a request, a nonce value or a client address.

Record	Content	Storage	Retention
Issuance journal	one object per serial (section 7): an entry, or a "not issued" mark	journal bucket of the production project (europe-west1), created only if absent, with a 12-year retention policy (bucket created 10 October 2026); the irreversible lock of that policy is deferred by owner decision of 11 October 2026 until after the first weeks of operation, and its date will be recorded here; test: 1 day, not locked	12 years
Daily journal batch	all entries and marks of the previous UTC day in `batches/journal-YYYY-MM-DD.jsonl` (sorted, canonical JSON lines), built at 00:20 UTC by a Cloud Run job started by Cloud Scheduler; its SHA-256 is time-stamped by DigiCert's public RFC 3161 server (`http://timestamp.digicert.com`, policy 2.16.840.1.114412.7.1, used without contract, no accuracy declared in its tokens; owner decision of 9 October 2026), by FreeTSA (`https://freetsa.org/tsr`) only as a fallback, never by IIC TSA itself, and anchored in Bitcoin through OpenTimestamps	journal bucket; file and proofs copied, with their SHA-256, to the archive bucket (europe-west4, ARCHIVE class), plus the public GitHub repository	12 years
Time guardian log	state changes and every readmission; per-cycle measurements	Cloud Logging of the TSA project; export to a locked bucket is a target (Decision to confirm)	12 years for state changes; 1 year for measurements (target)

Record	Content	Storage	Retention
Cloud KMS audit logs	Data Access logs of each signing call (identity, key version, time) unless its entry holds the signed-attributes digest (logging limits below); administrative actions	Cloud Logging of the TSA project; export to a locked bucket is a target (Decision to confirm)	12 years (target)
CRL archive	each root CRL issued, yearly or emergency	ceremony folder on the two USB backups; locked bucket and the public GitHub repository	12 years
Ceremony records	signed written record, the two SHA-256 certificate fingerprints, CRL hashes	paper original kept by the owner; scanned into a locked bucket	12 years

The daily batch digest is also anchored in Bitcoin through OpenTimestamps. This is independent archival integrity evidence for the journal only; it plays no part in validating a timestamp token, which rests on the TSA signature, its certificate and the revocation evidence; after a key compromise, it dates the journal evidence described below. A failure of DigiCert, FreeTSA or OpenTimestamps never blocks token issuance.

Journal evidence. On request, the operator provides the journal entries of a given day to a relying party. With the daily batch digest and its third-party timestamp or Bitcoin anchor, they show whether a token was issued by the service (Technical Specification, section 9.5).

Archive continuity. Each daily batch file and its proofs are copied to an archive bucket in another region (europe-west4, ARCHIVE storage class), under a 12-year retention policy (lock deferred by owner decision of 11 October 2026, see section 8); public items are also on GitHub. The SHA-256 of every archived object is recorded when it is written. Each quarter, a sample of each record type is restored from the archival copy and checked against those digests, and the journal also against the daily batch digests; the result is recorded. The signing key is not backed up: losing it means a new generation, not a restoration.

Logging limits. Cloud Run request logs, which contain client IP addresses, are excluded from the log buckets of the TSA projects, and request bodies are never logged; the client address is held in memory for at most one minute, for rate limiting only. Aggregate security and availability telemetry (request counts, rejection codes, latency) may be kept, provided it contains no request body, hash value, nonce value or client IP address. If the platform audit log of a signing call records the digest of the signed attributes, those entries are excluded before any log bucket of the TSA projects stores them, the default bucket included, since an exclusion drops a whole entry and a stored entry cannot be edited; signing calls are then traced through the issuance journal rather than through the Cloud

KMS audit logs. What the cloud provider keeps outside the TSA projects is not under the operator's control and is not covered here. What the request and audit logs stored in the TSA projects actually record is checked before opening (section 13). The issuance journal and the records of client requests contain no personal data; administrative and security audit records, including the Cloud KMS audit logs, may contain the identities of authorised personnel and are protected and retained under the operator's privacy and retention rules.

9. Security controls

The TSA is isolated from the operator's other services, including VerifBox: its own projects, identities and keys, so that a fault in the website cannot reach the signing key. VerifBox reaches IIC TSA only as a client, through the private service below, and sends only a SHA-256 hash: never the file, never its user's IP address.

Control	Practice
Projects	two separate Google Cloud projects in region europe-west1 (Belgium), TSA production (iic-tsa-prod-67100) and TSA test (iic-tsa-test-67100), none shared with VerifBox or any website; the archive bucket is in europe-west4; the root is offline and in no cloud project
Production services	two Cloud Run services built from the same image, using the same Cloud KMS key and the same journal: iic-tsa, public at https://tsa.internetidentitycard.com/tsa through Cloud Run domain mapping, limited to 60 requests a minute per address; and iic-tsa-verifbox, private, callable through IAM by VerifBox's service account only, limited to 600 requests a minute per address. Both keep no instance at rest (minimum instances 0). Part of the design approved on 9 October 2026, confirmed by the owner on 10 October 2026 and deployed the same day in iic-tsa-prod-67100: iic-tsa answers at https://iic-tsa-cnnddoxp6q-ew.a.run.app/tsa; the domain mapping tsa.internetidentitycard.com was created (CNAME to ghs.googlehosted.com in Cloud DNS) and its Google-managed certificate was being provisioned at the time of writing; iic-tsa-verifbox is called by VerifBox's service account; the daily batch job runs at 00:20 UTC (DigiCert, OpenTimestamps, archive copy).
Identities	each process runs under its own service account with only the rights it needs: signer and public-key viewer on the TSA key only for the service, object creator on the journal bucket; no long-lived key files exist

Control	Practice
Secrets	none are needed by the signing path; any other secret lives in a secret store with rotation and an access log, never in code, configuration files, environment variables or Git
Alerts	any change to IAM bindings, keys, log settings or retention policies of the TSA projects triggers an alert to the TSA Manager
Network	inbound only through HTTPS, to the public service on tsa.internetidentitycard.com and to the private VerifBox service, each with rate limiting; endpoints: POST /tsa; GET /tsa/certificate, /tsa/chain (root), /tsa/crl, /tsa/root-crl, /tsa/policy.json, /tsa/status, /tsa/health. The service connects only to the four time sources over TLS (HTTPS and WSS, port 443), Cloud KMS and Cloud Storage; the daily batch runs as a separate Cloud Run job that connects only to DigiCert, FreeTSA, the OpenTimestamps calendars and Cloud Storage
Software	dependencies pinned to exact versions; every release built from a tagged commit, reviewed by the operator and tested against the acceptance suite before deployment; there is no second reviewer (section 3)
Vulnerabilities	security advisories for the runtime and dependencies are reviewed weekly; a critical fix is deployed or the service is moved to FAIL

Reports of security issues are received at a security address on internetidentitycard.com, published in that domain's security.txt (to be created).

10. Incidents, compromise and termination

In any doubt about time, key or software integrity, the TSA stops issuing first and investigates second.

Event	Immediate action
Time guardian in FAIL	issuance stops automatically and resumes when SYNC returns; after a clock step, the operator reviews the guardian log before readmission
Cloud KMS unavailable	issuance stops (`systemFailure`); no fallback key is used
Hosting region unavailable	issuance stops; v1 has no failover to another region, by design, since a second region would need a second key generation

Event	Immediate action
Divergence with an external TSA beyond the sum of both accuracies	treated as an incident: issuance stops until the cause is explained
Critical software vulnerability	fix deployed, or service moved to FAIL until it is
Suspected compromise of the signing key	the compromise procedure below
Security breach with a significant impact on the service	issuance stops; the ICO is notified within 24 hours of becoming aware of the breach (UK eIDAS Article 19); affected relying parties, VerifBox included, are informed; a key compromise follows the procedure below, emergency CRL included

Compromise procedure:

1. Stop issuance: move the service to FAIL and disable the key version.
2. Determine, from the issuance journal and the Cloud KMS audit logs, the last instant at which the key is known to have been used only by the signing service.
3. Hold the emergency revocation ceremony: revoke the TSA certificate with reason keyCompromise and that instant as its invalidity date, and publish the emergency CRL on the service and on GitHub; VerifBox updates its copy.
4. Publish a notice stating the trust-cutoff instant. Tokens issued before it remain within the generation's declared validation scope, subject to the evidence and validation rules applicable to each token; the cutoff is a published bound, not a proof that every earlier token was legitimately issued. Tokens whose signature digest appears in a journal batch timestamped or anchored before the cutoff are shown to have been issued by the service.
5. Notify the ICO within 24 hours of becoming aware of the compromise and inform relying parties, VerifBox included.
6. Create a new key generation only after the cause has been identified and fixed: a new Cloud KMS key and a new TSA certificate signed by the root.

Termination of service: users are notified on the website at least three months in advance; issuance stops on the announced date; every TSA certificate not yet expired is revoked in a final root CRL, issued with a validity reaching the end of the longest remaining certificate validity, so that no token can be validated as current after the service has ended; the signing key is destroyed after that final root CRL is published; the journal stays in its locked bucket for the rest of its 12-year retention; the certificates, final root CRL, policy, this statement, the daily journal digests and the journal itself are published on GitHub and deposited with Software Heritage, whose archive does not depend on the operator's accounts, so that they can stay available for the rest of the 12-year retention period even if the operator no longer exists.

Accepted residual risks. These risks remain once their levers are in place and are accepted (Risk Register, section 10): a root compromised during a ceremony; access on the cloud provider's side, since exclusive custody of the Cloud KMS key is not claimed; a TSA key protected in software, not by a hardware security module; a root key held in

software on the owner's computer and on two USB backups; no issuance during a FAIL, an outage or a rotation; a single operator, with no separation of duties (section 3); loss of the cloud account, after which public evidence survives on GitHub and with Software Heritage; a day missing from the archives; and third-party integrations that verify badly, which fail on their own side.

11. Publication and archiving

Everything a verifier needs is published in two places that do not depend on each other: the service itself and the operator's public GitHub repository. Target: the repository is also deposited with Software Heritage, which preserves public source repositories independently of the operator's accounts, so that publication never rests on a single account. It complements, and does not replace, the locked storage and the two USB backups.

Item	On the service	On GitHub
Timestamp Policy (text; JSON summary)	`/tsa/policy.json` (JSON summary only)	yes, each version kept
This Practice Statement	not served	yes, each version kept
Current TSA certificate	`/tsa/certificate`	yes
Root certificate	`/tsa/chain`	yes
Current root CRL	`/tsa/crl`, `/tsa/root-crl`; copy at `verifbox.com/.well-known/iic-tsa-racine.crl`	daily archive
Daily journal digests and their external timestamps	not served	yes
Notices (rotation, revocation, incidents, termination)	`/tsa/status` and the website	yes

The DNS zone of internetidentitycard.com is DNSSEC-signed: a validating resolver returned it as authenticated on 4 October 2026, and it already publishes the fingerprint of the VerifBox document-signing keys. After the production ceremony (target), the fingerprints of the root and TSA certificates are added to that zone as TXT records, so that the trust anchor can be checked without trusting either the service or GitHub alone.

12. Testing before opening

The production endpoint opens only when every criterion below has passed on the test project and again after the production ceremony; results, including the time measurements (sources, offsets, thresholds, method and conclusion) on which the declared accuracy rests, are recorded and kept with the ceremony record. The criteria are defined in the Technical Specification, section 14; this table says how each is shown. The run after the production ceremony uses the production endpoint while it is closed to the

public; its tokens are journaled and kept as acceptance evidence, and none is issued to a client.

Status on 10 October 2026 (go-live): acceptance on the test project iic-tsa-test-67100 passed 43 of 43 checks, with measured offsets of a few milliseconds (section 6). The production infrastructure was created the same day: project iic-tsa-prod-67100, Cloud KMS key tsa-2026-a, journal and archive buckets with 12-year retention (lock pending owner approval). The production ceremony was performed (section 5) and the production acceptance passed at 18:22 UTC: 40 checks passed, 0 failed (`tests/staging-acceptance.sh`, TARGET=prod), guardian in SYNC with four sources. The first production token, serial `0xD22448FD5AD33BD7A393BB6024B28126`, was issued at 18:02:46 UTC and verified with OpenSSL. Production opened without waiting for the end of the 14-day measurement campaign (owner decision of 10 October 2026). VerifBox integration live at about 18:52 UTC: the site publishes the root and CRL at `/.well-known/iic-tsa.json` and `/.well-known/iic-tsa-racine.crl`, its engine pins the root, each proof carries the token over the file fingerprint, verified independently with OpenSSL (Verification: OK) and the public Python verifier. The second verifier written independently of the encoder (A3) is still open.

Id	Criterion	How it is shown
A1	SHA-256, SHA-384 and SHA-512 accepted; SHA-1 and truncated hashes rejected with the right failure code	automated request suite
A2	nonce copied; foreign `reqPolicy` rejected; any extension, critical or not, rejected. **A2b:** with `certReq` TRUE, only the TSA certificate is in `SignedData.certificates`, never the root; with `certReq` absent, `certificates` is absent	automated request suite
A3	tokens verified offline by `openssl ts -verify`, the operator's own verifier and a second verifier written independently of the encoder; OpenSSL run both at `genTime` (`-attime`) and at the current date	three verifiers, same token set
A4	ESSCertIDv2 present, single signature, EKU critical and exclusive	independent certificate and token inspection

Id	Criterion	How it is shown
A5	issuance stops within 5 seconds of the guardian's switch to FAIL; no issued token carries a `genTime` read after the switch, so no `genTime` falls more than 0.5 s inside a FAIL interval. A5b: a lease that expires or is withdrawn during signing yields no token	fault injection on the time guardian
A6	no serial reused after a crash or under concurrent requests. A6b: every issued token has its journal entry, with matching fields	crash and load tests against the journal
A7	the Cloud KMS key at protection level SOFTWARE, with the public key of the TSA certificate, checked before signing; no other identity can sign; TLS and document keys refused as token signers	key and protection-level check, IAM test and negative tokens
A8	published policy, chain and CRLs match the fields of issued tokens; the DNS fingerprints too, on the production run only	cross-check script
A9	BER, trailing bytes, a `certReq` written out as FALSE, version 0, empty hash and JSON bodies rejected	negative vectors
A10	no public claim beyond the statements allowed by the policy; the words qualified, eIDAS, FIPS and government appear only in disclaimers	review of every public page

Both ceremonies, creation and emergency revocation, are rehearsed in full on the test project before the production ceremony.

13. Decisions still open

Each item below is described elsewhere in this statement as a **target**, never as in place. Decided and binding: ECDSA P-256 with SHA-256 for the first generation, the simplified design approved by the owner on 9 October 2026 (a two-level hierarchy with no issuing CA, a Cloud KMS key at protection level SOFTWARE, a single operator, an on-demand time guardian, daily third-party anchoring of the journal, two production Cloud Run services sharing the key and the journal), time measured over TLS against four sources including

PTB (10 October 2026), opening of production without waiting for the 14-day measurement campaign (owner decision of 10 October 2026), region europe-west1 with an archive copy in europe-west4, no failover between regions in v1, and decisions D1 (the signature digest kept in the journal) and D2 (a leaf certificate valid for up to 135 months, its key used for 15 months at most and destroyed within 18 months).

- ☑ Outbound UDP from Cloud Run: tested on 10 October 2026 and blocked; time is now measured over TLS (section 6) and NTS is no longer a target on Cloud Run
- ☐ Time thresholds and declared accuracy: 14-day measurement campaign on the test service, every 10 minutes since 10 October 2026, reviewed at day 14 (share of SYNC cycles, 99th percentile of the median offset, 99th percentile of the spread); production does not wait for it
- ☑ Acceptance: passed on the test project (43 of 43) and in production after the ceremony (40 of 40, 18:22 UTC) on 10 October 2026 (section 12)
- ☑ Production ceremony: performed on 10 October 2026, SHA-256 fingerprints of the production root and TSA certificates recorded (section 5)
- ☐ Retention lock of the production journal and archive buckets (irreversible): deferred by owner decision of 11 October 2026 until after the first weeks of operation
- ☐ Production budget alert
- ☐ HTTPS certificate for tsa.internetidentitycard.com (Google-managed, being provisioned on 10 October 2026)
- ☐ Decision to confirm: export of the time guardian log and the Cloud KMS audit logs to a locked bucket, and their retention
- ☐ What the request and audit logs stored in the TSA projects record for a request and for a signing call: no client address, request body or signed-attributes digest may remain in them
- ☐ The second, independent verifier for criterion A3
- ☐ French versions of the policy and of this statement
- ☐ Service domain (`tsa.internetidentitycard.com`: domain mapping and CNAME created on 10 October 2026, HTTPS certificate still being provisioned), contact and security addresses on internetidentitycard.com, and its security.txt
- ☐ Registrar lock and automatic renewal of internetidentitycard.com; billing alerts (production budget alert still to be set) and a second payment method for the cloud account
- ☐ Internal service agreement with VerifBox, a client of the IIC TSA operated by the same company: access conditions, continuity if either business is transferred, and the disclosure VerifBox makes to its users that the TSA is not an independent third party (the integration itself is live since 10 October 2026)
- ☐ Deposit of the public repository with Software Heritage, as a preservation copy independent of the operator's accounts

- ☐ Approval record, at approval: version, effective date, approver, SHA-256 of the approved documents (the policy, this statement, the Technical Specification and the Risk Register) and of the policy in JSON, policy OID and publication date

Revision history

Only the latest draft is current; earlier copies are withdrawn. Draft revisions before approval change neither the policy OID nor its version.

Draft	Date	Change
1.0, erratum 2	2026-10-11	Section 10, termination: made explicit that every TSA certificate not yet expired is revoked in the final root CRL, whose validity reaches the end of the longest remaining certificate validity (clarification found during the self-assessment against ETSI EN 319 421, requirement OVR-7.14-02; no change to the service)
1.0, erratum 1	2026-10-11	Sections 8 and 13: the retention lock is deferred by owner decision, not yet applied; section 11: public issuance on the public endpoint closed on 10 October 2026 (available to VerifBox only, public access on request), recorded as an operational notice on /tsa/status

Draft	Date	Change
1.0	2026-10-10	Approved: owner approval given by launching the production service (policy.json status "approved"). Production ceremony recorded (offline on the owner's iMac, single operator; root R1 and tsa-2026-a fingerprints replace the placeholders; first root CRL to 10 October 2027 17:29 UTC; two USB backups verified, working copy deleted; only public files to Cloud Shell); production services iic-tsa and iic-tsa-verifbox deployed in iic-tsa-prod-67100, domain mapping created, daily batch job at 00:20 UTC; production acceptance 40 of 40 checks at 18:22 UTC with the guardian measurements; first production token at 18:02:46 UTC; VerifBox integration live; open items updated (retention lock, budget alert, HTTPS certificate, second verifier, day-14 review)

Draft	Date	Change
9	2026-10-10	Facts of 10 October 2026. Changed from the older design: the four NTP sources over UDP (PTB, NIST, Netnod, Cloudflare) are replaced by four sources over TLS (PTB web clock, the traceable one; Akamai; Google and Cloudflare via the HTTP Date tick), since Cloud Run blocks outbound UDP; usability, SYNC rule and logged fields updated; NIST web time excluded; NTS no longer a target on Cloud Run; no background guardian (TSA_BACKGROUND_GUARDIAN =0); network row and diagram updated. Two production services recorded as part of the approved design, no longer a Decision to confirm. Added: test project and test fingerprints, acceptance 43/43, measured offsets; production project iic-tsa-prod-67100, key tsa-2026-a and its public-key hash, buckets not yet locked, domain verified with Google; production fingerprints left as placeholders until the ceremony, pending the USB keys; owner decision to open production without waiting for the 14-day measurement campaign; DigiCert tokens declare no accuracy

Draft	Date	Change
8	2026-10-09	Revised to the simplified design approved by the owner on 9 October 2026; title without version number. Two-level hierarchy, no issuing CA; Cloud KMS key at SOFTWARE level, HSM and key attestation (A7) removed; single operator with compensating controls, Key Officers and Witness removed; offline ceremony on the owner's computer in a Docker container without network, root key encrypted in software with two USB backups, hardware tokens, safes and video removed; yearly root CRL and emergency CRL; on-demand time guardian with four named sources, NTS and staging checks as open items; journal fields, daily DigiCert, FreeTSA and OpenTimestamps anchoring, archive in europe-west4; two production services as a Decision to confirm; ICO notification within 24 hours; disclosure that VerifBox belongs to the same company
7	2026-10-08	certReq absent or TRUE accepted, and a FALSE written out rejected as non-DER (step 2, A2b, A9); one journal object per serial, entry or "not issued" mark, so that a marked serial is never drawn again; log exclusions and checks stated for the logs stored in the TSA projects, what the cloud provider keeps elsewhere being outside the operator's control

Draft	Date	Change
6	2026-10-08	Any request extension rejected, as RFC 3161 requires; lease withdrawn on entering FAIL, automatic return to SYNC except after a clock step, and a source traceable to UTC required among the usable ones; steps 4 to 11, genTime included, performed by the signer; one key per generation; daily batches also anchored in Bitcoin; a SHA-256 recorded for every archived object; retention raised to 12 years; two-person control stated as procedural; the cloud provider's access and the accepted residual risks stated; the archiver's network access; termination deposit with Software Heritage; log entries holding the signed-attributes digest excluded before storage, as the policy requires; criteria A2b, A3, A5, A7, A8 and A10 aligned with the specification; open items: platform logs, root and issuing CA validity, substitutes, registrar lock and billing alerts, D1 and D2 recorded as decided

Draft	Date	Change
5	2026-10-07	Technical Specification v1.0 and Risk Register v1.0 cited under their IIC TSA titles, replacing the open item to retitle them; issuance order corrected: genTime read before the token is built, and the journal entry written after signing and before the response; genTime rounded to the nearest second; low-S signatures; root CRL; data-field rationale made exact; criteria A5b and A6b. Proposed for review: D1, the signature digest of each token kept in the journal as evidence after a compromise; D2, a leaf certificate valid for up to 135 months whose key signs for at most 15 months, with root and issuing CA targets of 30 and 20 years
4	2026-10-07	Personal-data distinction aligned with the policy; Software Heritage stated as complementing, not replacing, the locked storage and the safe; draft number and date shown in the identification table
3	2026-10-07	Third-party TSA for the journal cross-check; time measurements kept with the ceremony record; Software Heritage deposit and approval record added to the open items
2	2026-10-07	The TSA is placed in the Internet Identity Card business line and renamed IIC TSA; policy and test OIDs assigned. Copies titled "VerifBox TSA" are withdrawn; no change of substance
1	2026-10-04	Initial draft, including the corrections of the first reviews

© 2026 Https Card — Internet Identity Card Ltd, 124 City Road, London EC1V 2NX, United Kingdom. UK Company No. 09168431. Governed by the laws of England and Wales.