Attestazione di esistenza per opere digitali: architettura, garanzie e limiti

Whitepaper tecnico, v1.0 Spazio Genesi ETS

Pubblicato il 21 luglio 2026.

Abstract

This document describes, in technical terms, what a digital-existence attestation system built on public infrastructure can and cannot guarantee. The system computes a SHA-256 fingerprint of a digital work — on the primary channels, entirely on the client, so the file itself never leaves the user's device — and binds it to a server-generated timestamp with an HMAC token. Three independent time anchors corroborate the resulting certificate: the HMAC itself, an RFC 3161 timestamp from a Certificate Authority trusted in Adobe's AATL program (embedded in a PAdES B-LT/LTV signature), and a Bitcoin timestamp via OpenTimestamps. None of these mechanisms proves authorship; together they prove that a specific fingerprint existed at a specific instant in time, and that the resulting certificate has not been altered since.

The document is deliberately as precise about what the system does **not** guarantee as about what it does: the signing identity is self-signed (not a qualified eIDAS trust service), declared metadata (title, author, year) are self-declarations, and the HMAC signing key is a single point of trust mitigated — not eliminated — by a documented recovery path and continuous self-testing. A public, machine-readable governance registry (the Genesis Trust Framework) makes every claim in this document independently checkable, including the integrity of this PDF itself, which is attested on the very service it describes.

Sommario

Questo documento descrive, in termini tecnici, cosa un sistema di attestazione di esistenza per opere digitali costruito su infrastruttura pubblica può e non può garantire. Il sistema calcola l'impronta SHA-256 di un'opera digitale — sui canali principali interamente **sul dispositivo dell'utente**, che il file non lascia mai — e la vincola a un timestamp generato dal server tramite un token HMAC. Tre àncore temporali indipendenti corroborano il certificato risultante: l'HMAC stesso, una marca temporale RFC 3161 da un'autorità di certificazione riconosciuta nel programma Adobe AATL (incorporata in una firma PAdES B-LT/LTV), e un'ancora Bitcoin via OpenTimestamps. Nessuno di questi meccanismi prova la paternità dell'opera; insieme provano che una determinata impronta esisteva a un determinato istante, e che il certificato risultante non è stato alterato da allora.

Il documento è deliberatamente preciso tanto su ciò che il sistema **non** garantisce quanto su ciò che garantisce: l'identità del firmatario è self-signed (non un servizio fiduciario qualificato eIDAS), i metadati dichiarati (titolo, autore, anno) sono auto-dichiarazioni, e la chiave di firma HMAC è un punto di fiducia singolo — mitigato, non eliminato, da un meccanismo di recupero e da un'auto-

verifica continua. Un registro di governance pubblico e leggibile da macchina (il Genesis Trust Framework) rende ogni affermazione di questo documento verificabile in modo indipendente, inclusa l'integrità di questo stesso PDF, attestato sul servizio che descrive.

---

1. Scopo e pubblico

Questo documento non è materiale promozionale. È un documento tecnico pubblico rivolto a chi deve valutare con rigore che cosa il sistema di attestazione garantisce — e, in modo altrettanto esplicito, che cosa non garantisce. Si rivolge a prestatori di servizi fiduciari (nel seguito del processo di valutazione per un eventuale sigillo elettronico qualificato), al Consiglio Direttivo e all'assemblea dei soci dell'ente, ad accademie e partner convenzionati che devono spiegare il servizio ai propri iscritti, a revisori esterni indipendenti, e a ricercatori e sviluppatori che vogliono integrare o verificare il sistema in autonomia.

Il principio guida di questo documento — e, più in generale, di tutta la comunicazione pubblica del progetto — è che **le garanzie si dichiarano insieme ai limiti**. Non esiste una sezione "vantaggi" senza una sezione "limiti" di pari dignità: il valore probatorio di un'attestazione dipende tanto da ciò che copre quanto dalla trasparenza su ciò che non copre.

Chi fornisce questo servizio non è indifferente a come lo si valuta. Spazio Genesi è un ente del terzo settore (ETS) senza scopo di lucro: non ha soci di controllo da remunerare né un obiettivo di uscita che renda conveniente esagerare le garanzie o minimizzare i limiti. Questo non sostituisce le prove tecniche descritte nelle sezioni seguenti — nessuna forma giuridica, di per sé, garantisce l'onestà di un sistema — ma è la ragione strutturale per cui le scelte descritte in questo documento (politiche fail-open dichiarate, una sezione limiti di pari peso rispetto alle garanzie, un registro di governance pubblico anziché autodichiarato) sono coerenti con l'assenza di un incentivo commerciale a fare diversamente.

2. Il sistema in breve

Il sistema produce **attestazioni di esistenza** ("proof of existence") per opere digitali di qualsiasi formato: un certificato PDF firmato che dichiara "questa impronta crittografica esisteva a questo istante", con tre meccanismi indipendenti a corroborare la data.

2.1 Flusso end-to-end

1. L'utente seleziona l'opera; il **browser** ne calcola l'impronta SHA-256 con l'API WebCrypto nativa. Il file non viene mai inviato al server, per nessuna operazione — né attestazione né verifica.
2. Il server riceve la sola impronta (più eventuali metadati facoltativi dichiarati dall'utente — titolo, autore, anno, note), genera un timestamp lato server e un token HMAC che vincola impronta e timestamp insieme.
3. Su richiesta esplicita dell'utente, il server genera un certificato PDF con l'impronta, il timestamp, un QR di verifica, ed effettua tre operazioni di ancoraggio (§3): l'apposizione della firma HMAC (già fatto), l'invio a una terza parte per la marca temporale RFC 3161, e la sottomissione a più calendari OpenTimestamps per l'ancoraggio Bitcoin.

4. Il certificato firmato viene archiviato e restituito all'utente.
5. La verifica successiva può avvenire in due modi indipendenti: un ricalcolo dell'impronta in locale (mai un invio del file al server), oppure il solo controllo della validità della firma HMAC — utile per chi possiede il certificato ma non l'opera stessa in quel momento.

2.2 Sei canali, una sola garanzia crittografica

Il sistema è raggiungibile da sei canali distinti — un'interfaccia web, un bot di messaggistica, due server per agenti software (uno locale, uno remoto), un'interfaccia programmatica per sviluppatori, e una pagina pubblica permanente per ogni opera attestata. I meccanismi crittografici che garantiscono l'integrità di un certificato (HMAC, timestamp server-side, rate limiting) sono **identici su tutti i canali**: nessun canale ha un percorso di emissione "più debole" degli altri. Ciò che cambia da canale a canale è esclusivamente **dove transita il file** da attestare — non l'integrità del risultato.

Sul canale web e sui due canali per agenti software il file non transita da alcun server: l'impronta è calcolata sul dispositivo dell'utente — dal browser nel primo caso, dall'agente che esegue codice in locale negli altri due — e solo l'impronta viene trasmessa. Il canale di messaggistica fa eccezione dichiarata: lì il file transita per forza di cose, l'impronta è calcolata al volo e i byte vengono scartati senza essere conservati, e questo è comunicato all'utente **prima** del primo utilizzo, non dopo.

2.3 Full privacy by design

La scelta architetturale più rilevante del canale primario (l'interfaccia web) è che l'impronta si calcola interamente lato client: il server non riceve, non elabora e non può conservare i byte dell'opera. Questo non è un compromesso sulla capacità probatoria del certificato: un'impronta crittografica a 256 bit non è invertibile, quindi il certificato attesta propriamente "questa impronta esisteva a questo istante" indipendentemente da dove l'impronta sia stata calcolata — la garanzia anti-retrodatazione resta intatta perché timestamp e firma restano generati esclusivamente dal server. È lo stesso modello di fiducia adottato da OpenTimestamps stesso, i cui calendari pubblici vedono solo digest, mai i file originali.

3. Le tre àncore temporali indipendenti

Un singolo certificato attestato porta tre prove indipendenti della sua data, ciascuna verificabile senza dover fidarsi delle altre due né del servizio che le ha raccolte.

3.1 HMAC — l'integrità immediata

Al momento dell'emissione, il server calcola un token HMAC-SHA256 che vincola insieme l'impronta dell'opera, il timestamp generato dal server (mai dal client) e — quando presenti — i metadati dichiarati dall'utente, normalizzati in una forma canonica stabile. Nessuna di queste componenti può essere alterata dopo l'emissione senza invalidare la firma: un tentativo di modificare l'impronta dichiarata, il timestamp o i metadati di un certificato già emesso produce un token che non supera più la verifica.

I campi puramente descrittivi — nome del file, dimensione dichiarata, tipo MIME — restano **fuori** dal perimetro vincolato dalla firma: sono metadati di comodo per la leggibilità del certificato, non parte della prova. Questo è un residuo di design dichiarato, non una lacuna scoperta successivamente: l'impronta crittografica, il timestamp e i metadati dichiarati dall'autore sono l'unico contenuto che il certificato prova davvero.

3.2 Marca temporale RFC 3161 — la terza parte fidata

Il certificato PDF è firmato in formato PAdES B-LT con una marca temporale RFC 3161 ottenuta da un'autorità di certificazione presente nel programma Adobe Approved Trust List (AATL), con validazione a lungo termine (LTV: la catena del certificato e le risposte OCSP sono incorporate nel documento). La conseguenza pratica è che un lettore PDF conforme (Adobe Acrobat in primis) mostra la data del certificato come attestata da una terza parte indipendente e riconosciuta — senza dover fidarsi della sola dichiarazione del servizio. Il meccanismo di firma è fail-open per design: se la terza parte (TSA) o il servizio di verifica dello stato del certificato (OCSP) non sono raggiungibili al momento dell'emissione, il certificato viene comunque prodotto, privo della sola marca temporale — l'emissione non si blocca per un disservizio esterno.

3.3 Ancoraggio Bitcoin — la prova decentralizzata

Ogni certificato viene inoltre ancorato alla blockchain Bitcoin tramite il protocollo OpenTimestamps: l'impronta dell'opera (concatenata con un nonce) viene sottomessa in parallelo a più calendari pubblici e gratuiti, ciascuno dei quali costruisce indipendentemente una prova crittografica che, una volta confermata, lega quella impronta a un blocco Bitcoin specifico — quindi a un intervallo di tempo verificabile da chiunque, senza fidarsi né del servizio né dei calendari stessi. La prova è "pendente" al momento dell'emissione e matura entro poche ore, quando la transazione Bitcoin sottostante riceve conferma. La ridondanza su più calendari indipendenti significa che la prova resta valida anche se uno o più calendari smettono di funzionare: ne basta uno a rispondere.

3.4 Indipendenza e degradazione controllata

Le tre àncore non dipendono l'una dall'altra: la caduta di una non invalida le altre due, e ciascuna poggia su un'infrastruttura distinta — i sistemi del servizio, un'autorità di marcatura temporale terza, la rete Bitcoin attraverso calendari pubblici indipendenti. La differenza rilevante non è però solo tecnica: **la prima delle tre è l'unica sotto il controllo di chi gestisce il servizio**, e le altre due esistono precisamente per questo. È il ragionamento sviluppato al §4.3 (M4) e dichiarato come limite al §8.5.

Se la terza parte per la marca temporale o i calendari OpenTimestamps sono temporaneamente irraggiungibili, il sistema **non blocca l'emissione**: preferisce produrre un certificato con una garanzia in meno piuttosto che nessun certificato — una scelta di design esplicita, non un compromesso accidentale. La firma interna resta in ogni caso la garanzia minima, sempre presente e verificabile istantaneamente; il certificato che ne risulta è però distinguibile da uno completo, perché la riga di ancoraggio semplicemente non compare quando la prova non esiste. Il degrado è visibile, non silenzioso.

4. Modello di minaccia

Un sistema che produce prove è credibile solo se dichiara contro quali avversari quelle prove reggono. Questa sezione elenca ciò che il sistema protegge, da chi, e — per ogni minaccia — se esiste una mitigazione tecnica oppure se il rischio residuo è **accettato esplicitamente**. Non ci sono omissioni silenziose: dove una minaccia non è mitigata, è detto.

4.1 Ciò che va protetto

1. **L'integrità del legame impronta ↔ istante ↔ metadati dichiarati.** È il cuore probatorio: tutto il resto è contorno.
2. **La verificabilità nel tempo**, anche indipendentemente dal servizio. Una prova che valesse solo finché il servizio esiste sarebbe una prova debole.
3. **L'assenza dei byte dell'opera sui canali che non ne prevedono il transito** (§2.2). Non è riservatezza nel senso della custodia: è assenza. Ciò che non si riceve non si può perdere, né sequestrare, né registrare per errore.
4. **La continuità dell'archivio** dei certificati emessi e delle prove di ancoraggio.

4.2 Gli avversari considerati

- **F — Falsificatore esterno.** Vuole un certificato che dichiari un'impronta o una data che il server non ha mai osservato. Può chiamare qualunque endpoint pubblico e costruire richieste arbitrarie; non conosce la chiave di firma.
- **C — Titolare di credenziale legittima che ne abusa.** Ha un accesso valido e vuole spingersi oltre ciò per cui gli è stato concesso.
- **S — Il servizio stesso**, o chi ne compromettesse i segreti: l'unico avversario che detiene la chiave di firma. È l'avversario che un fornitore tende a omettere dal proprio modello di minaccia; qui è incluso, ed è trattato al punto M4.
- **A — Attaccante sull'archivio.** Vuole alterare o cancellare certificati già emessi.
- **T — Terze parti** (autorità di marcatura temporale, calendari di ancoraggio, infrastruttura di hosting): irraggiungibili, in errore, o compromesse.
- **O — Osservatore.** Vuole sapere che cosa gli utenti stanno attestando, o ottenerne i contenuti.

4.3 Minacce, mitigazioni e residui

M1 — Far firmare un certificato con impronta inventata o data retrodatata *(F)*. Il token di firma è emesso dal server e vincola insieme impronta, istante e metadati canonici; l'endpoint che produce il certificato non si fida del contenuto ricevuto, ma verifica che ciò che sta per stampare coincida esattamente con ciò che il token autentica. Senza la chiave non è costruibile un token valido per valori diversi. **Residuo: nessuno rispetto a questo avversario.**

M2 — Riusare un token legittimo con valori alterati *(C)*. I campi vincolati dalla firma non sono alterabili: qualunque modifica invalida la verifica. **Residuo accettato**: i campi puramente descrittivi — nome del file, dimensione, tipo — restano fuori dal perimetro firmato, quindi chi possiede un token

legittimo può alterarne la sola visualizzazione. È un residuo dichiarato dall'origine, non una scoperta successiva (§6, §8.4).

M3 — Abusare della propria quota *(C)*. Mitigato da quote, limitazione di frequenza per indirizzo, revoca e allarmi automatici al superamento delle soglie. Ciò che questo avversario **non** può fare è falsificare: una credenziale compromessa emette certificati *veri* entro la propria quota, perché non tocca il meccanismo che garantisce l'integrità. **Residuo accettato**: un abuso "autorizzato" resta tale fino alla revoca, che è un atto umano.

M4 — Retrodatare dall'interno *(S)*. Chi detiene la chiave di firma può firmare qualunque coppia impronta/istante: il servizio, in astratto, potrebbe attestare a sé stesso una data che non ha mai osservato. **Questa è la ragione per cui le tre àncore non sono ridondanza decorativa.** La marca temporale è emessa da una terza parte che il servizio non controlla, e nessuno — gestore incluso — può far comparire un digest in un blocco Bitcoin già minato. Un certificato che porta tutte e tre le àncore **non è retrodatabile nemmeno da chi gestisce il servizio**. Un certificato che porta la sola firma interna — perché le terze parti erano irraggiungibili al momento dell'emissione (§3.4) — lo sarebbe: per questo il degrado è visibile sul certificato stesso, che semplicemente non riporta la riga di ancoraggio quando la prova non esiste. **Residuo dichiarato**, non eliminato: vedi §8.5.

M5 — Perdere la chiave, o ruotarla per errore *(S, involontario)*. La perdita impedirebbe di emettere nuove attestazioni valide; una rotazione errata invaliderebbe la verifica di tutti i certificati già emessi, perché il meccanismo non prevede una doppia chiave. Mitigato dalla recuperabilità dichiarata del segreto (il fatto che esista è pubblico; il meccanismo no, per ragioni di sicurezza) e da **due verifiche automatiche continue e distinte**: una sonda interna che a ogni campionamento di stato rifà un giro completo di firma e verifica, e un canary esterno indipendente che rispedisce periodicamente al servizio un'attestazione nota, firmata una volta sola con la chiave corretta. Il secondo esiste proprio perché il primo, per costruzione, non rileverebbe una rotazione errata: rifarebbe firma e verifica con la chiave *attuale*, e resterebbe coerente con sé stesso. **Residuo**: la chiave resta singola (§8.5).

M6 — Alterare o cancellare l'archivio *(A)*. La mitigazione più importante è architetturale: **l'archivio non è la prova**. La distruzione completa dell'archivio non invalida un certificato già in possesso dell'utente — la firma interna resta verificabile sull'endpoint pubblico, la marca temporale è incorporata nel PDF, l'ancoraggio su blockchain è indipendente da qualunque nostro sistema. L'archivio è un servizio di recupero, non il fondamento della garanzia. È comunque replicato su un backup geograficamente separato, e la sua ripristinabilità è stata verificata con una prova di ripristino reale (§6.2). **Residuo accettato**: chi perdesse il proprio certificato in coincidenza con la perdita di archivio e backup perderebbe il documento, non la data — che resterebbe dimostrabile per via indipendente a chi conservi il file originale e la propria copia della prova di ancoraggio.

M7 — Terze parti irraggiungibili o in errore *(T)*. Mitigato da ridondanza (l'ancoraggio interroga più calendari indipendenti in parallelo, e ne basta uno che risponda) e da una politica **fail-open dichiarata**: se una terza parte non risponde, il certificato viene comunque emesso con una garanzia in meno, invece di non essere emesso affatto. **Residuo accettato e reso visibile**: il certificato che ne

risulta è distinguibile a occhio da uno completo, perché la riga di ancoraggio manca. Se invece l'autorità di marcatura temporale emettesse una marca errata, resterebbero valide le altre due àncore: nessuna delle tre dipende dalle altre.

M8 — Scoprire che cosa è stato attestato *(O)*. Sul canale primario non c'è nulla da intercettare: il file non viene mai inviato, quindi non esiste un punto in cui possa essere osservato o registrato per errore. Gli altri canali dichiarano il proprio modello di transito prima dell'uso, mai dopo. La chiave d'accesso a un certificato archiviato e alla sua pagina pubblica è **l'impronta stessa**, a 256 bit: non è enumerabile, e chi non la conosce non può scoprire che una certa opera sia stata attestata. Il modello di fiducia è dichiarato e simmetrico: **chiunque conosca l'impronta può consultare la pagina pubblica**, il che è la funzione del servizio, non una falla.

M9 — Far credere al pubblico una garanzia superiore a quella reale *(interno, anche in buona fede)*. È l'unica minaccia di questo elenco che non ha un avversario esterno: il rischio che un materiale divulgativo, una pagina o un futuro collaboratore lascino intendere una qualificazione che il servizio non possiede. È formalizzata come rischio nel registro pubblico, con un controllo dedicato che ne è la mitigazione: una dichiarazione esplicita e stabile di ciò che il servizio **non** è, versionata insieme al resto (§8.1, §9). Questo documento è a sua volta parte di quella mitigazione.

4.4 Il metodo alla prova: una falla reale, trovata e corretta

Il modello di minaccia non è un esercizio compilato a posteriori. Nella fase iniziale di sviluppo, un audit di sicurezza esterno condotto a scatola chiusa individuò un difetto grave: l'endpoint che produceva il certificato firmato accettava senza verifica i dati ricevuti dal client, rendendo possibile far firmare un certificato con un'impronta arbitraria e una data retrodatata. La correzione — il token che vincola impronta e istante, e la verifica di coerenza prima della firma — è esattamente il meccanismo descritto al §3.1: non un'aggiunta successiva, ma la risposta diretta a quel difetto, applicata nello stesso ciclo di sviluppo in cui fu segnalato.

Il caso è riportato qui, e non omesso, per una ragione precisa: la credibilità di un modello di minaccia si misura da quello che ha già trovato, non da quanto è completo sulla carta.

5. Fondamenti crittografici

Tutta la garanzia del sistema poggia su una sola primitiva: la funzione di hash crittografica SHA-256. Questa sezione dichiara che cosa esattamente si assume di quella primitiva, quale ne è lo stato di sicurezza pubblico alla data di pubblicazione, e che cosa accadrebbe se quell'assunzione venisse meno.

5.1 SHA-256

SHA-256 appartiene alla famiglia SHA-2, standardizzata dal NIST nella pubblicazione FIPS 180-4 (*Secure Hash Standard*). Produce, da un input di lunghezza arbitraria, un digest di 256 bit; è deterministica (lo stesso input dà sempre lo stesso digest) e non invertibile in pratica. Il NIST raccomanda SHA-256 come minimo per le applicazioni di funzioni hash che richiedono interoperabilità, e la famiglia SHA-2 resta pienamente ammessa nelle proprie linee guida di transizione (SP 800-131A).

5.2 Tre proprietà, e a chi serve ciascuna

Le funzioni di hash crittografiche si valutano su tre proprietà distinte, che nel discorso divulgativo vengono spesso confuse. La distinzione conta, perché **proteggono parti diverse**:

- **Resistenza alla preimmagine** — dato un digest, è computazionalmente infattibile risalire a un input che lo produca. È ciò che rende l'impronta comunicabile senza rivelare l'opera: è la proprietà su cui si fonda la privacy del canale primario (§2.3).
- **Resistenza alla seconda preimmagine** — dato uno specifico input, è infattibile trovarne un *altro*, diverso, con lo stesso digest. Protegge **chi possiede un certificato** contro chiunque altro: nessun terzo può fabbricare un file diverso che risulti coperto dalla vostra attestazione. È la proprietà che rende il certificato significativo per chi deve valutarlo.
- **Resistenza alle collisioni** — è infattibile trovare *due* input qualsiasi con lo stesso digest, potendoli scegliere entrambi. Protegge **i terzi contro chi attesta**: senza di essa, chi attesta potrebbe preparare in anticipo due contenuti con la stessa impronta, attestarne uno e sostenere in seguito di aver attestato l'altro.

Una precisazione che il progetto ritiene doveroso fare, perché la formulazione semplificata circola anche in letteratura divulgativa: **non è vero che a una prova di esistenza serva solo la seconda preimmagine**. Serve anche la resistenza alle collisioni, ma per proteggere una parte diversa da quella che si tende a immaginare — non chi attesta, bensì chi deve credergli. È utile saperlo perché è proprio la resistenza alle collisioni, e non la seconda preimmagine, la proprietà che storicamente è caduta per MD5 e per SHA-1.

Va detto per completezza che un attaccante capace di produrre collisioni su SHA-256 potrebbe preparare un'attestazione *ambigua*, ma non potrebbe comunque **alterare né retrodatare** un'attestazione già emessa: quella è protetta dalla firma e dalle àncore temporali, non dalla sola impronta.

5.3 Stato dell'arte alla data di pubblicazione

Alla data di questo documento (luglio 2026), **non esiste alcuna collisione nota per SHA-256 completo**, che opera su 64 passi. I migliori risultati pubblici riguardano versioni ridotte dell'algoritmo:

- la prima collisione **pratica** su SHA-256 ridotto a 31 passi, ottenuta in circa 1,2 ore su 64 thread (ASIACRYPT 2024);
- attacchi di collisione estesi fino a **37 passi su 64**, presentati a EUROCRYPT 2026 (IACR ePrint 2026/232, febbraio 2026) — il primo avanzamento sul numero di passi attaccabili dopo oltre un decennio.

Trentasette passi su sessantaquattro è un margine ampio, e nessuno di questi risultati si applica all'algoritmo completo. Il progresso, però, è reale e va riconosciuto: è il motivo per cui questa sezione dichiara una data, e non una condizione permanente.

Per la **seconda preimmagine** — la proprietà che protegge chi detiene un certificato — non è noto alcun attacco migliore di quelli generici. L'unico attacco generico contro le costruzioni di tipo Merkle–Damgård, cui SHA-256 appartiene (Kelsey–Schneier, 2005), riduce il lavoro solo per messaggi molto

lunghi: per un messaggio di 2^k blocchi il costo scende all'ordine di 2^(256−k). Per dare un ordine di grandezza concreto: anche per un file da un terabyte (circa 2^34 blocchi) il costo resterebbe dell'ordine di 2^222 operazioni. Non c'è alcun margine di praticabilità, oggi né in alcuno scenario prevedibile.

Neppure un calcolatore quantistico su larga scala cambierebbe questo quadro in modo sostanziale: l'algoritmo di Grover riduce la ricerca di una preimmagine da 2^256 a circa 2^128 operazioni, un livello ancora fuori portata. È la ragione per cui i percorsi di transizione post-quantistica riguardano la crittografia a chiave pubblica e non impongono la sostituzione delle funzioni hash della famiglia SHA-2.

5.4 La lezione di SHA-1: come muore una funzione di hash

SHA-1 non è caduta all'improvviso. La complessità stimata per trovarne una collisione scese progressivamente a partire dal 2005, ben al di sotto della soglia di forza bruta; la prima collisione effettivamente calcolata fu pubblicata nel 2017. Il NIST ne ha poi disposto il ritiro formale (annuncio del dicembre 2022): SHA-1 è già vietata per la generazione di firme digitali, ed è deprecata fino al 31 dicembre 2030 e non ammessa oltre tale data.

La lezione non è che SHA-256 farà la stessa fine — sarebbe una previsione, e questo documento non ne fa. La lezione è **di metodo**: una funzione di hash non si rompe in un giorno, si indebolisce per gradi, pubblicamente, nell'arco di anni. Nel caso di SHA-1 sono trascorsi oltre due decenni fra i primi indebolimenti teorici e il divieto d'uso: un preavviso ampio, e soprattutto pubblico. È esattamente questo che rende possibile dichiarare oggi come ci si comporterebbe.

5.5 Che cosa accadrebbe se SHA-256 si indebolisse

Primo: le prove già emesse resterebbero prove d'epoca. Che un algoritmo si indebolisca nel 2035 non conferisce a nessuno la capacità di aver prodotto, nel 2026, una collisione che nel 2026 non era producibile. Il valore probatorio di un'attestazione va valutato rispetto allo stato dell'arte **alla sua data** — ed è precisamente per questo che la data è a sua volta ancorata a fonti indipendenti dal servizio.

Secondo: una transizione sarebbe una ri-attestazione, non una riscrittura. Un'opera già attestata potrebbe essere nuovamente attestata con l'algoritmo successivo. Il nuovo certificato porterebbe la data nuova; il vecchio continuerebbe a esistere e a valere per la propria. Nessuna prova viene invalidata retroattivamente, e nessuna viene retrodatata.

Terzo, e va detto perché è la parte scomoda: la dipendenza è condivisa. L'ancoraggio su Bitcoin e il protocollo OpenTimestamps si fondano essi stessi su SHA-256, come gran parte dell'infrastruttura mondiale di firma digitale. Un indebolimento pratico di SHA-256 non sarebbe un problema di questo servizio: sarebbe un evento sistemico. Dichiararlo è più utile che rivendicare un'indipendenza fra le àncore che, su questo specifico piano, non esiste — l'indipendenza descritta al §3.4 riguarda i fornitori e i punti di guasto, non la primitiva crittografica sottostante.

Una nota tecnica, in senso favorevole: fra le tre àncore, la firma interna è quella che un attacco alle collisioni intaccherebbe meno. La sicurezza di un codice di autenticazione HMAC non riposa sulla resistenza alle collisioni della funzione hash sottostante, bensì sulle proprietà della sua funzione di

compressione — motivo per cui HMAC-MD5 rimase considerato robusto come autenticatore anche dopo che MD5 era caduta come funzione di hash.

6. Catena di custodia

Ciò che il token HMAC vincola — impronta, timestamp, metadati dichiarati in forma canonica — è distinto da ciò che resta puramente descrittivo (nome file, dimensione, tipo MIME): questa distinzione, già introdotta al §3.1, è la base su cui si fonda ogni garanzia di integrità successiva all'emissione.

6.1 Archiviazione

I certificati emessi vengono archiviati in un deposito con residenza dei dati nell'Unione Europea, e sono recuperabili solo da chi conosce l'impronta dell'opera: la chiave di recupero è l'impronta stessa, a 256 bit, quindi non è possibile enumerare l'archivio né scoprire per tentativi che cosa vi sia stato depositato.

6.2 Backup offsite e prova di ripristino

L'archivio primario è replicato su un backup geograficamente separato, con verifica periodica di completezza. Il valore di un backup non dichiarato è nullo finché non viene provato: una prova di ripristino reale ha prelevato tre campioni rappresentativi (un'opera recente, un'opera del primo periodo di attività del servizio, un'opera con ancoraggio Bitcoin già maturo) esclusivamente dal backup — non dall'archivio primario — e li ha verificati con gli stessi strumenti pubblici disponibili a chiunque: la validità della firma HMAC tramite l'endpoint pubblico di verifica, la validità della prova di ancoraggio Bitcoin contro un blocco reale, e la coerenza del badge pubblico d'archivio. Tutti e tre i campioni sono risultati verificati con esito positivo; il verbale pubblico della prova documenta anche un errore di metodo emerso durante l'esecuzione (uno strumento di estrazione non ancora aggiornato ai metadati facoltativi) e la sua correzione — un esempio, non un'eccezione, del modo in cui questo progetto tratta i propri errori: documentati, non nascosti.

6.3 Garanzie di recupero

Il servizio pubblica, in una pagina dedicata soggetta a revisione periodica, garanzie minime di recupero del certificato differenziate per fascia di utilizzo — un impegno di custodia, non solo una possibilità tecnica. Questo documento non riporta le cifre esatte (soggette a revisione più frequente di quanto sia opportuno per un PDF immutabile): si rimanda alla pagina pubblica delle condizioni di utilizzo per i valori correnti.

7. Il tempo

7.1 Perché il timestamp è solo lato server

Il timestamp che entra nella firma è generato esclusivamente dal server, mai ricevuto dal client: è il meccanismo che rende impossibile a **chi richiede l'attestazione** retrodatarla. Un client potrebbe dichiarare qualunque data desiderasse, ma quella dichiarazione non entrerebbe mai nel token firmato

— solo l'istante osservato dal server viene vincolato dalla firma.

La precisazione «a chi richiede l'attestazione» è necessaria e non è una formalità: rende impossibile la retrodatazione a tutti tranne che a chi detiene la chiave di firma, cioè al servizio stesso. A rendere impraticabile anche quella provvedono le altre due àncore, non questa (§4.3 M4, §8.5).

7.2 Granularità e affidabilità di ciascuna àncora

Le tre àncore temporali offrono garanzie di granularità diverse, e questo documento le dichiara esplicitamente invece di presentarle come equivalenti. Il timestamp del server e la firma HMAC sono immediati e disponibili all'istante dell'emissione. La marca temporale RFC 3161 è anch'essa immediata, incorporata direttamente nella firma del PDF. L'ancoraggio Bitcoin via OpenTimestamps è l'unico dei tre a maturare nel tempo: la prova nasce "pendente" e diventa verificabile in modo completamente indipendente solo dopo la conferma della transazione Bitcoin sottostante, tipicamente entro poche ore.

Precisione e affidabilità, però, sono due cose diverse, e vanno lette in ordine inverso. L'àncora più **precisa** — l'istante al secondo osservato dal server — è anche la meno **indipendente**, perché è il nostro orologio. La marca di terza parte è ugualmente precisa e assai più indipendente. L'ancoraggio su blockchain è il meno preciso dei tre — non individua un istante ma un intervallo, quello del blocco che lo contiene — ed è però il più difficile da contestare, perché non richiede di fidarsi di nessuno, noi inclusi. Un lettore che debba valutare un certificato farebbe bene a considerarle in quest'ordine.

7.3 Cosa significa "prova di esistenza a una data"

Nessuna delle tre àncore prova che l'utente abbia creato l'opera in quel momento, né che ne sia l'autore: tutte e tre provano che una specifica impronta crittografica — quindi, con probabilità praticamente certa, uno specifico contenuto — esisteva già a quella data. È lo stesso concetto tecnico su cui si fonda l'intero protocollo OpenTimestamps ("proof of existence"), qui applicato con tre livelli di corroborazione indipendenti anziché uno solo.

8. Limiti dichiarati

Questa è la sezione più importante del documento. Ogni limite qui elencato è una scelta consapevole o un residuo noto: nessuno è stato scoperto da terzi e ammesso a posteriori. Per ciascuno si dichiara che cosa il sistema non garantisce, perché, che cosa attenua il problema e — dove esiste — che cosa servirebbe per superarlo.

8.1 L'identità del firmatario non è garantita da una terza parte

Il servizio produce **attestazioni di esistenza non qualificate**: firma elettronica avanzata con certificato self-signed, più marca temporale RFC 3161 da un'autorità presente nella trust list di Adobe e ancoraggio su Bitcoin. **Non è un servizio fiduciario qualificato ai sensi di eIDAS.**

La conseguenza è concreta e chiunque può verificarla da sé aprendo un nostro certificato in un lettore PDF conforme: la **data** risulta attestata da una terza parte attendibile, mentre l'**identità** del firmatario resta di validità sconosciuta — il riscontro complessivo del documento non diventa quindi pienamente

verde. Non si tratta di un difetto di configurazione: è precisamente ciò che significa non essere un prestatore qualificato.

Il rischio residuo è altrettanto concreto: un omonimo potrebbe generare un proprio certificato self-signed intestato allo stesso nome. A distinguere un certificato autentico da un'imitazione restano la firma interna, verificabile sull'endpoint pubblico del servizio, e la presenza in archivio.

Il superamento di questo limite richiede un **sigillo elettronico qualificato** presso un prestatore di servizi fiduciari qualificato. È nella roadmap pubblica del progetto, condizionato alla sostenibilità economica dell'organizzazione; **non è associato ad alcuna data**, perché i tempi non dipendono da noi. Sul piano tecnico si tratterebbe della sostituzione del certificato di firma e del fornitore, non di una modifica dell'architettura.

8.2 I metadati dichiarati sono auto-dichiarazioni

Titolo, autore, anno e note sono forniti dall'utente. Il sistema li rende **immutabili** — sono vincolati alla firma, e un'alterazione successiva invalida la verifica — ma **non li verifica**: nessun meccanismo del sistema prova che chi attesta sia l'autore di ciò che attesta.

La distinzione fra immutabilità e veridicità è il punto: il certificato etichetta esplicitamente questi campi come *dichiarati*, con un'avvertenza sulla loro natura. Ciò che il sistema prova è che quei metadati erano esattamente quelli, in quella forma, a quella data. Chi dichiara il falso lascia dunque una dichiarazione falsa, datata e non ripudiabile — che è diverso da nulla, ma non è una prova di paternità.

Superare questo limite richiederebbe un'identità forte legata alla persona. Oggi le uniche identità presenti nel sistema sono indirizzi email verificati tramite provider terzi, e solo per chi richiede credenziali di sviluppo o un abbonamento: il percorso di attestazione ordinario resta anonimo per scelta. Modelli di identità digitale europea sono seguiti con interesse; nessuno è oggetto di impegno in questa versione del documento.

8.3 Il sistema attesta un'impronta, non dei byte

Sul canale primario il server non vede mai il file. Ne consegue che chiunque può attestare un'impronta **senza possedere il file corrispondente** — inclusa un'impronta inventata di sana pianta.

Questo non indebolisce la prova, e vale la pena esporre l'argomento per intero anziché rassicurare:

- un'impronta inventata non è preimmagine di alcun contenuto: nessuno potrà mai esibire un file che la produca, quindi il certificato che ne risulta è inutilizzabile per chiunque, incluso chi l'ha ottenuto;
- attestare l'impronta di un'opera altrui era già possibile a chiunque ne possedesse i byte, anche quando il file transitava dal server; e nella prova di esistenza prevale comunque **la data più antica**;
- la paternità, come detto al §8.2, non era comunque provata prima e non lo è ora.

Ciò che il server ha smesso di verificare è **soltanto** che l'impronta derivi da byte realmente presentati. Dimensione e tipo di file erano già campi descrittivi non vincolati anche in precedenza. Il prezzo di questa scelta è dichiarato apertamente, e così il suo corrispettivo: il file non lascia mai il dispositivo dell'utente.

8.4 Nome, dimensione e tipo del file non sono vincolati dalla firma

Come già osservato al §3.1 e al §6, i campi puramente descrittivi restano fuori dal perimetro firmato: chi possiede un token legittimo può alterarne la visualizzazione sul proprio certificato. Impronta, istante e metadati dichiarati restano invece vincolati. È un residuo di basso impatto, dichiarato dall'origine e mai presentato diversamente.

8.5 La chiave di firma è singola: un residuo di fiducia nel servizio

Chi detiene la chiave di firma interna può firmare qualunque coppia impronta/istante. In astratto, il servizio potrebbe quindi attestare a sé stesso una data che non ha mai osservato. È il limite strutturale più serio dell'architettura, ed è il motivo per cui le altre due àncore esistono.

L'attenuazione non è retorica ma verificabile: la marca temporale è emessa da una terza parte che il servizio non controlla, e nessuno — gestore incluso — può far comparire un digest in un blocco Bitcoin già minato. **Un certificato che porta tutte e tre le àncore non è retrodatabile nemmeno da chi gestisce il servizio.** Un certificato che porta la sola firma interna, perché le terze parti erano irraggiungibili al momento dell'emissione, lo sarebbe: è la ragione per cui il degrado è reso visibile sul certificato stesso e non nascosto (§3.4, §4.3 M7).

Un secondo aspetto va dichiarato: **la chiave non è ruotabile**. Il meccanismo non prevede una doppia chiave, quindi una rotazione invaliderebbe la verifica di tutti i certificati già emessi. Non è una svista, è un vincolo di progetto — e ha una conseguenza operativa esplicita: la custodia del segreto diventa un controllo critico, la sua recuperabilità è dichiarata come esistente (il meccanismo non è pubblico, per ragioni di sicurezza: descriverlo ridurrebbe la sicurezza senza aggiungere nulla di verificabile da terzi), e la sua integrità è sorvegliata da due verifiche automatiche continue e indipendenti (§4.3 M5).

Ciò che supererebbe davvero il limite è un **registro di trasparenza pubblico e append-only**, su cui pubblicare gli impegni in modo che nemmeno il gestore possa riscrivere il proprio passato. È una voce in valutazione, citata qui per completezza: **non è un impegno assunto da questo documento**.

8.6 Non è una conservazione a norma, né un servizio forense

Il sistema applica per ispirazione i principi di acquisizione e conservazione della prova digitale pertinenti al proprio modello — impronta calcolata con algoritmo dichiarato e codice ispezionabile, archivio con tre àncore indipendenti — ma **non offre**, perché il modello è self-service e non forense, un registro di catena di custodia per singola prova consultabile dall'utente, ruoli formalizzati di primo intervento e analisi, né report di validazione degli strumenti nel senso investigativo classico.

La ragione è strutturale, non un ritardo di implementazione: le norme forensi nascono per chi maneggia prove **altrui** all'interno di un procedimento, mentre qui è l'autore ad attestare la propria opera in autonomia. L'applicabilità è per questo dichiarata come **parziale**, mai piena, nei requisiti collegati del registro pubblico.

8.7 La risoluzione dello storico di stato è di trenta minuti

La pagina pubblica di stato del servizio campiona ogni trenta minuti: sotto quella soglia il sistema **non inventa un dato che non possiede**. La rappresentazione aggregata è onesta in entrambe le direzioni — un disservizio di pochi minuti non viene trasformato in un giorno di guasto, né viene fatto sparire nell'arrotondamento.

8.8 Ciò che il sistema non promette affatto

Per chiudere ogni ambiguità, quattro cose che un'attestazione di esistenza **non** è:

- **Non è una protezione contro la copia o l'uso non autorizzato.** Il sistema non impedisce nulla; produce una prova, non un controllo d'accesso.
- **Non è una registrazione del diritto d'autore presso alcun ente.** Il diritto d'autore sorge con la creazione dell'opera, non con una registrazione; il servizio fornisce una prova di data, non un titolo.
- **Non attesta l'originalità dell'opera** né che essa non violi diritti di terzi.
- **Non rileva se un contenuto sia stato generato da un'intelligenza artificiale.** È una domanda diversa, cui rispondono tecnologie di provenienza: se ne parla al §10.

9. Standard e riferimenti

Il sistema si appoggia, in modo esplicito e verificabile, ai seguenti standard e riferimenti pubblici:

- **RFC 3161** — Internet X.509 Public Key Infrastructure Time-Stamp Protocol: il protocollo con cui viene richiesta la marca temporale di terza parte (§3.2).
- **ETSI PAdES** (PDF Advanced Electronic Signatures), profilo B-LT: il formato di firma con validazione a lungo termine usato per il certificato PDF.
- **OpenTimestamps**: il protocollo aperto di ancoraggio su blockchain usato per la terza àncora temporale (§3.3).
- **RFC 9116** (`security.txt`): il formato standard con cui il servizio pubblica il proprio canale di responsible disclosure per la ricerca di sicurezza in buona fede, con un impegno pubblico di safe harbor.
- **GDPR** (Regolamento UE 2016/679): il percorso di attestazione anonimo (il canale primario) non tratta alcun dato personale dell'utente finale; i soli trattamenti esistenti nel sistema riguardano l'identità di chi richiede credenziali per sviluppatori o un abbonamento, documentati pubblicamente con base giuridica e tempi di conservazione.
- **ISO/IEC 27037** (linee guida per identificazione, raccolta e conservazione della prova digitale): applicata **per ispirazione e in modo parziale**, con i limiti dichiarati al §8.6. L'applicabilità è registrata come *parziale*, mai piena, nei requisiti pubblici collegati.
- **W3C Verifiable Credentials**: modello di riferimento seguito con attenzione come possibile evoluzione futura del formato di certificato; nessuna adozione né impegno di roadmap in questa versione del documento.

9.1 Posizionamento rispetto a eIDAS

Il riferimento normativo è il Regolamento (UE) n. 910/2014, come modificato dal Regolamento (UE) 2024/1183 (comunemente "eIDAS 2"). Il posizionamento del servizio è dichiarato in modo esplicito e stabile, ed è il medesimo pubblicato sul registro di trasparenza del progetto.

Che cosa il servizio è. Produce attestazioni di esistenza **non qualificate**: firma elettronica avanzata con certificato self-signed, marca temporale RFC 3161 emessa da un'autorità riconosciuta nella trust list di Adobe, ancoraggio su blockchain Bitcoin.

Che cosa il servizio non è. Non è un prestatore di servizi fiduciari qualificato. Non figura in alcuna lista di fiducia nazionale o europea, non emette firme, sigilli o marche temporali qualificate, e non gode delle presunzioni che il Regolamento riserva ai servizi qualificati — in particolare la presunzione di accuratezza della data e dell'ora e di integrità dei dati associati, che l'articolo 41 riconosce alle **sole** marche temporali qualificate.

Che cosa questo non significa. Il Regolamento stabilisce un principio di non discriminazione: a una firma elettronica (art. 25), a una marca temporale elettronica (art. 41) e a un documento elettronico (art. 46) non possono essere negati gli effetti giuridici e l'ammissibilità come prova in giudizio per il solo motivo della loro forma elettronica o del fatto che non soddisfino i requisiti della corrispondente versione qualificata. La differenza fra qualificato e non qualificato non è dunque fra "vale" e "non vale": è che la sola versione qualificata beneficia di una presunzione stabilita per legge — per la marca temporale, l'articolo 41 la definisce come presunzione di accuratezza della data e dell'ora e di integrità dei dati associati — che sposta l'onere della prova su chi contesta. La versione non qualificata resta pienamente ammissibile come prova, ma senza quell'automatismo specifico: la sua efficacia probatoria va argomentata nel merito, invece di beneficiare di una presunzione di legge. Questo servizio si colloca — dichiaratamente — nel secondo gruppo.

Roadmap. L'evoluzione verso un sigillo elettronico qualificato è nella roadmap pubblica del progetto ed è condizionata alla sostenibilità economica dell'organizzazione. Non è associata ad alcuna data (§8.1).

10. Posizionamento rispetto a C2PA / Content Credentials

La domanda che questo documento riceve più spesso da lettori tecnici è se un'attestazione di esistenza sia in concorrenza con le Content Credentials del C2PA. La risposta è no, e la ragione è che rispondono a due domande diverse.

10.1 Che cosa fa C2PA

La Coalition for Content Provenance and Authenticity (C2PA) è un progetto della Joint Development Foundation, affiliata della Linux Foundation, e pubblica una specifica aperta con licenza esente da royalty. La versione corrente alla data di questo documento è la **2.4, di aprile 2026**.

L'oggetto centrale della specifica è il **Content Credential**: un manifest firmato che raccoglie asserzioni sull'asset — origine, dispositivo o strumento di produzione, modifiche successive, impiego di intelligenza artificiale generativa — legato crittograficamente al contenuto tramite valori di hash. Il

manifest è tipicamente **incorporato nel file**, ma la specifica prevede anche manifest conservati all'esterno.

L'adozione è reale e crescente, sia negli strumenti di editing sia in fotocamere di più produttori, nei generatori di contenuti basati su intelligenza artificiale e in alcune piattaforme di pubblicazione.

10.2 Due modelli, due domande

C2PA risponde a: **da dove viene questo contenuto e che cosa gli è stato fatto**. È una risposta ricca di contesto — una storia, non un fatto isolato.

Un'attestazione di esistenza risponde a: **questo esatto contenuto esisteva già a questa data**. È un solo bit di informazione, privo di contesto, ma non falsificabile e non rimovibile.

Nessuna delle due risposte sostituisce l'altra.

10.3 Il punto di frizione: la durabilità

Un manifest incorporato viaggia con il file e ne condivide la sorte: un salvataggio con uno strumento che non conosce lo standard, una ricompressione al caricamento su una piattaforma, uno screenshot — e le Content Credentials scompaiono, **senza lasciare alcun segnale che siano mai esistite**. Il problema è riconosciuto dalla specifica stessa, che dedica una sezione al confronto fra dati incorporati e conservati all'esterno; la coalizione vi ha risposto con i cosiddetti *soft binding* — filigrane invisibili e impronte percettive — e con un'API di risoluzione che consente di ritrovare il manifest a partire dal contenuto anche quando i metadati sono stati rimossi.

L'impronta crittografica ha esattamente il problema opposto. Non porta con sé alcun contesto: nulla, dentro il file, dice che è stato attestato — occorre conservare il certificato, o conoscere l'impronta. In compenso **non può essere rimossa**, perché non si trova nel file: è una proprietà matematica del file. Finché il file esiste identico, l'impronta si ricalcola.

La simmetria vale in entrambe le direzioni, e per onestà va detta anche nella direzione sfavorevole: i *soft binding* attenuano la fragilità di C2PA, ma introducono una dipendenza da un servizio di risoluzione remoto e da algoritmi di filigranatura la cui robustezza è oggetto di ricerca attiva; l'impronta esatta non richiede alcun servizio per essere ricalcolata, ma è per costruzione fragile a **qualunque** modifica del file, anche di un solo bit — un ritaglio, una ricompressione, una conversione di formato producono un'impronta diversa, e l'attestazione precedente non li copre.

10.4 Perché sono complementari

Un'opera generata con strumenti di intelligenza artificiale e dotata di manifest C2PA può essere **anche** attestata: il manifest racconta come è nata, l'attestazione fissa che quella esatta versione esisteva a quella data — utile, per esempio, se il manifest venisse in seguito rimosso, o per stabilire una precedenza rispetto a un'opera concorrente.

Un'opera priva di qualunque provenienza incorporata — perché rimossa, mai apposta, o prodotta con strumenti che non supportano lo standard — non ha alcuna Content Credential da esibire. Lì l'attestazione di esistenza è l'unica prova disponibile.

Il contesto normativo europeo rende questa complementarità più rilevante, non meno: gli obblighi di trasparenza sui contenuti generati o manipolati da sistemi di intelligenza artificiale introdotti dal regolamento europeo sull'intelligenza artificiale, applicabili dal 2 agosto 2026, spingono l'adozione di marcature di provenienza leggibili da macchina. Quanto più la provenienza viaggia dentro i metadati, tanto più diventa utile disporre anche di una prova di data che dai metadati **non dipende**.

10.5 Nessuna adozione di C2PA in questa versione

È una decisione esplicita, non un'omissione, e ha tre ragioni distinte:

1. **Indipendenza dal formato.** Il sistema attesta l'impronta di qualunque tipo di file; C2PA copre un insieme definito di formati media.
2. **Incompatibilità con il modello di privacy.** Apporre un manifest significa **modificare il file dell'utente**: operazione impossibile in un'architettura in cui il file non lascia mai il dispositivo e il server non ne vede i byte. Non è una questione di priorità di sviluppo, è un'incompatibilità strutturale con la garanzia principale del servizio.
3. **Lo stesso nodo dell'identità.** Firmare manifest richiederebbe un'identità di firma riconosciuta come attendibile dagli strumenti di verifica — esattamente il limite dichiarato al §8.1.

Va però osservato che le due tecnologie sono già oggi **componibili dall'utente senza alcuna integrazione**: attestare un file che contiene già Content Credentials funziona senza modifiche, e l'impronta calcolata copre anche il manifest incorporato. L'attestazione, in quel caso, congela anche la provenienza dichiarata — che diventa così a sua volta datata e non alterabile.

11. Trasparenza operativa

Le affermazioni di questo documento che riguardano il sistema — comprese quelle sui suoi limiti — non chiedono di essere credute: sono riscontrabili in un registro di governance pubblico e leggibile da macchina (controlli, rischi, decisioni, evidenze), oltre che nel codice sorgente dei componenti pubblicati con licenza libera. Le poche affermazioni che riguardano invece lo stato della ricerca crittografica o standard di terzi (§5, §10) citano la fonte pubblica corrispondente e la data in cui è stata consultata.

Il registro alimenta un punteggio di maturità pubblico, calcolabile da chiunque a partire dagli stessi dati versionati nel repository — non un punteggio di marketing, ma una media di dieci indicatori indipendenti (trasparenza, integrità, tracciabilità, documentazione, automazione, audit, conservazione, riproducibilità, privacy, governance), ciascuno derivato da fatti raccolti automaticamente piuttosto che da autodichiarazioni. Il punteggio è consultabile in tempo reale sul Trust Center pubblico del progetto, insieme al suo storico: ogni rilevazione periodica è committata e conservata nel repository pubblico, quindi un calo resterebbe visibile — e la sua eventuale rimozione lo sarebbe altrettanto.

Le evidenze che sostengono i controlli attivi sono raccolte in gran parte in modo automatico da un collettore che interroga direttamente i sistemi in produzione, non da dichiarazioni periodiche di chi gestisce il servizio. Il registro prevede inoltre una **revisione esterna annuale**, affidata a una terza parte indipendente senza legami con l'ente né con gli enti formativi convenzionati (Radixia srl, individuata e confermata nel luglio 2026); alla data di questo documento il revisore è stato designato e

il piano operativo della prima revisione è pubblicato, ma la revisione non è ancora stata svolta. Il suo esito sarà un verbale pubblico, come già avvenuto per la prova di ripristino (§6.2) e per la prima revisione interna periodica.

Il servizio pubblica un canale dichiarato di responsible disclosure (§9) e una pagina di stato pubblica che riporta, con risoluzione onesta, sia i disservizi sia i rallentamenti che **non** hanno causato alcun disservizio ma che restano visibili nel dettaglio della giornata: un episodio minore non viene né esagerato né fatto sparire nell'aggregazione dello storico (§8.7).

12. Storia delle revisioni · Licenza · Come verificare questo documento

Versione corrente: v1.0 — pubblicato il 21 luglio 2026. Il file pubblicato `whitepaper-v1.0.pdf` non viene più modificato dopo l'attestazione della sua impronta (§ sotto): eventuali correzioni successive alla pubblicazione danno luogo a una nuova versione (`whitepaper-v1.1.pdf` e successive), mai a una sostituzione silenziosa del file esistente. Le versioni precedenti restano pubblicate.

Come verificare questo documento: l'impronta SHA-256 di questo PDF è stata a sua volta attestata sul servizio che questo documento descrive. L'impronta e il link alla pagina pubblica di verifica sono pubblicati sulla pagina di accompagnamento di questo documento (`trust.spaziogenesi.org/whitepaper.html`) — non qui, per una ragione tecnica precisa: un PDF non può contenere l'impronta di sé stesso, perché aggiungerla ne cambierebbe il contenuto e quindi l'impronta stessa. Chiunque può scaricare questo PDF, ricalcolarne l'impronta in autonomia (con qualunque strumento SHA-256, non necessariamente il nostro) e confrontarla con quella pubblicata sulla pagina di accompagnamento — la stessa chiusura circolare già applicata al bundle di evidenze mensile del registro di governance.