HOLogram

Human Obfuscation Layer

L’AI vede un riflesso, non l’originale

Whitepaper

Versione 1.0 (Marzo 2026) - BaffiSan

info@hologramprotocol.org

www.hologramprotocol.org

Abstract

L’evoluzione dell’intelligenza artificiale ha reso possibile identificare e profilare gli esseri umani non attraverso ciò che dichiarano, ma attraverso il loro comportamento digitale: la velocità con cui digitano, le micro-pause tra una pressione e l’altra, le traiettorie del puntatore, la dinamica dello scroll, il modo in cui interagiscono con pulsanti, link e contenuti.

Queste caratteristiche, note come biometria comportamentale, costituiscono oggi un identificatore unico, persistente e difficilmente modificabile.

La profilazione comportamentale avviene in modo silenzioso e costante, grazie a semplici script integrati nelle pagine web e a modelli di AI in grado di ricostruire pattern personali con elevata precisione.
L’utente non ha strumenti per difendersi: i browser moderni proteggono dal fingerprinting del dispositivo, ma non proteggono dal fingerprinting del comportamento umano.

HOLogram propone un nuovo paradigma di tutela: un Human Obfuscation Layer, uno strato locale che offusca, normalizza e riduce la granularità dei segnali comportamentali, preservando la naturalezza dell'interazione ma impedendo l’identificazione e la profilazione avanzata da parte dei sistemi di AI.

Il protocollo HOLogram introduce un approccio human-centric, trasparente e conforme ai principi del diritto digitale, operando completamente in locale e senza raccogliere o trasmettere dati dell’utente. HOLogram si presenta come standard aperto e come base per un ecosistema di strumenti e implementazioni decentralizzate, sostenuto da un modello di reputazione non finanziario (HOLoToken) basato sulla logica *Proof-of-Contribution*.

L’obiettivo è offrire all’essere umano un nuovo diritto digitale:

il diritto a non essere riconosciuto dal proprio comportamento.

Indice

Abstract 1

1. Introduzione 1

2. Il Problema 2

3. Obiettivi di HOLogram 4

3.1 Proteggere l'identità comportamentale dell'utente 4

3.2 Offuscare senza alterare l'esperienza d'uso 4

3.3 Introdurre un modello di protezione locale, indipendente e trasparente 4

3.4 Contrastare la profilazione comportamentale AI-driven 5

3.5 Creare uno standard aperto per la privacy comportamentale 5

3.6 Favorire un ecosistema decentralizzato e meritocratico 5

3.7 Stabilire un nuovo diritto digitale: la privacy comportamentale 6

4. Principi Fondativi 7

4.1 Human-Centric Privacy 7

4.2 Local-Only & Zero-Retention 7

4.3 Minimizzazione della Riconducibilità 7

4.4 Offuscamento Non Manipolativo 8

4.5 Naturalità dell'Interazione 8

4.6 Adaptive Obfuscation 8

4.7 Trasparenza e Controllo dell'Utente 8

4.8 Neutralità Tecnologica 9

4.9 Apertura del Protocollo 9

4.10 Merito come motore evolutivo 10

4.11 Conservare la Libertà Comportamentale dell'Individuo 10

5. Architettura del Protocollo HOLogram 11

5.1 ShadowDOM Input Vault 11

5.2 Pointer & Scroll Normalizer 11

5.3 Persona Mixer (session-consistent) 12

5.4 Behavioral Differential Privacy Engine *(Componente tecnica di offuscamento)* 12

5.5 Behavioral Privacy Budget (Componente di gestione e controllo) 13

5.6 Exposure HUD (Human-Visible Transparency Layer) 13

5.7 Optional HID Cloak (livello OS) 14

5.8 Sintesi del flusso operativo 14

5.9 Filosofia architetturale 15

6. Funzionamento Operativo del Sistema 16
6.1 Acquisizione dei segnali (Input Capture Layer) 16
6.2 Normalizzazione del segnale (Signal Shaping Layer) 16
6.3 Personalizzazione sintetica (Persona Mixer Layer) 17
6.4 Protezione differenziale (Behavioral Differential Privacy Engine) 17
6.5 Gestione dell'esposizione (Behavioral Privacy Budget Layer) 17
6.6 Emissione verso il browser o l'applicazione (Output Projection Layer) 18
6.7 Integrazione con sistemi esterni 18
6.8 Trasparenza verso l'utente (Exposure HUD) 18
6.9 Ciclo di vita di una sessione 19
6.10 Sintesi 19
7. Sicurezza, Limitazioni e Modelli di Minaccia 20
7.1 Obiettivi di sicurezza 20
7.2 Perimetro di fiducia (Trust Boundary) 20
7.3 Modello di minaccia (avversari e capacità) 20
7.4 Strategie difensive del protocollo 21
7.5 Rilevabilità e anti-rilevazione 22
7.6 Interazione con sistemi anti-bot 22
7.7 Correlazione cross-site 23
7.8 Attacchi di adattamento (adversarial adaptation) 23
7.9 Limitazioni note 23
7.10 Scelte di sicurezza e impatto su UX 23
7.11 Test, validazione e audit 24
7.12 Roadmap difensiva 24
7.13 Sintesi 24
8. Compliance & Aspetti Legali 25
8.1 Assenza di trattamento dati 25
8.2 Conformità ai principi generali di tutela 25
8.3 Neutralità e assenza di manipolazione 25
8.4 Assenza totale di profilazione 26
8.5 Trasparenza verso l'utente 26
8.6 Compatibilità con i contesti ad alta tutela 26
8.7 Limitazioni giuridiche e implementative 27
8.8 Ruolo del protocollo nella tutela dei diritti digitali 27
8.9 Riferimenti normativi — nota non vincolante 27

8.10 Sintesi operativa 28
9. Ecosistema & Standardizzazione di HOLogram 29
9.1 Obiettivo dell'ecosistema 29
9.2 Ruoli nell'ecosistema 29
9.3 Artefatti ufficiali 29
9.4 Processo RFC (proposte di cambiamento) 30
9.5 Interoperabilità & Conformità 30
9.6 Varianti & Fork 31
9.7 Integrazione con HOLoToken (HRL/HCS) 31
9.8 Percorso di Standardizzazione 31
9.9 Integrazioni di riferimento (adozione pratica) 32
9.10 Trasparenza, Audit & Sicurezza 32
9.11 Governance leggera (meritocratica e decentralizzata) 32
9.12 Tabella di marcia (indicativa) 32
9.13 Sintesi 33
10. Visione a Lungo Termine 34
10.1 Proteggere la libertà comportamentale come diritto digitale 34
10.2 HOLogram come strato nativo del Web 34
10.3 Evoluzione verso standard aperti (HLP – HOLogram Layer Protocol) 35
10.4 Ecosistema di varianti, fork e innovazione distribuita 35
10.5 Adattamento ai nuovi modelli di AI comportamentale 35
10.6 Decentralizzazione attraverso HOLoToken (HRL/HCS) 36
10.7 Integrazione trasversale nei futuri ambienti digitali 36
10.8 Missione finale 36
10.9 Conclusione 37
11. HOLoToken — HRL & HCS (Proof-of-Contribution, non finanziario) 38
11.1 Scopo e principi 38
11.2 HRL — HOLogram Reputation Layer (il "sistema") 38
11.3 HCS — HOLogram Contribution Score (la "metrica") 39
11.4 Classi di contributo (catalogo base) 39
11.5 Meccanismo di calcolo (Proof-of-Contribution) 39
11.6 Badge, ruoli, permessi (esempi) 40
11.7 API minime (esportazione & integrazione) 41
11.8 Governance del sistema reputazionale 41
11.9 Sicurezza & anti-abuso 41

11.10 Privacy & conformità 41
11.11 Limiti e scopi 42
11.12 Roadmap reputazionale (indicativa) 42
11.13 Esempio d'uso (end-to-end) 42
12. Hash / Notarizzazione di HOLogram 43
12.1 Obiettivo della notarizzazione 43
12.2 Generazione dell'hash (SHA-256) 43
12.3 Pubblicazione dell'hash in repository pubblici 44
12.4 Pubblicazione su IPFS (InterPlanetary File System) 44
12.5 Timestamping distribuito (opzionale ma consigliato) 44
12.6 Versionamento e archivio storico 44
12.7 Paternità sotto pseudonimo 45
12.8 Verifica da parte della community 45
12.9 Sintesi 46
A.1 Struttura dei Moduli del Layer 47
A.2 Esempio di Ciclo di Elaborazione (pseudocodice) 47
A.3 Input Vault — Specifiche Minime 48
A.4 Normalizzazione — Modello Operativo 48
A.5 Persona Mixer — Parametri Sintetici 48
A.6 Differential Privacy Engine — Linee Guida 49
A.7 Behavioral Privacy Budget — Logica Operativa 49
A.8 Output Projector — Garanzia di Coerenza 52
A.9 Glossario Tecnico 52
A.10 Schema Riassuntivo del Flusso 52

1. Introduzione

L'interazione umana con il web ha sempre avuto due facce: ciò che l'utente intende comunicare (il contenuto) e ciò che il suo corpo involontariamente rivela (il comportamento). Nell'era dell'Intelligenza Artificiale, questa seconda faccia è diventata centrale: i sistemi osservano come scriviamo, come muoviamo il puntatore, come scorriamo le pagine, come esitazioni e micro-pause si ripetono nel tempo. Questi pattern costituiscono una biometria comportamentale: una firma unica, stabile, difficilmente modificabile, capace di ricondurre l'interazione a una persona anche in assenza di cookie, login o fingerprinting tradizionale.

Questa trasformazione ha creato un vuoto di tutela. Le soluzioni note—VPN, navigazione privata, difese anti-fingerprinting del dispositivo—proteggono l'ambiente, non l'essere umano. Possono celare l'indirizzo IP, omogeneizzare le API del browser, limitare la raccolta di attributi hardware; non schermano però la dinamica con cui una mano si muove o una mente si ferma a pensare prima di premere Invio. Il risultato è che l'utente resta esposto: determinabile, correlabile, talvolta profilabile ben oltre la sua consapevolezza.

HOLogram nasce per colmare esattamente questo vuoto. È un Human Obfuscation Layer: uno strato locale, trasparente e controllabile, che offusca, normalizza e riduce la granularità dei segnali comportamentali prima che raggiungano gli script delle pagine o i modelli di AI che li analizzano. L'obiettivo non è "ingannare" il web, ma riequilibrare la relazione tra persona e piattaforma: permettere di interagire in modo naturale senza cedere, come costo implicito, la propria identità comportamentale.

Il protocollo si fonda su alcuni principi semplici:

- Human-centric privacy: la protezione riguarda l'essere umano, non solo il dispositivo;
- Local-only: nessun dato lascia il terminale; nessuna telemetria, nessuna conservazione;
- Trasparenza e controllo: l'utente vede e decide il livello di offuscamento applicato;
- Neutralità: HOLogram non manipola decisioni; riduce solo la fedeltà dei segnali esposti;
- Apertura: è uno standard; le implementazioni possono essere molteplici e indipendenti.

La promessa è semplice: preservare la naturalezza dell'interazione, ma far sì che chi osserva riceva una versione meno identificante dei segnali originali. In altre parole, un ologramma del comportamento: visibile, utilizzabile, funzionale, ma non idoneo a ricostruire con precisione l'identità di chi lo genera.

HOLogram è, inoltre, pensato come infrastruttura di ecosistema. Il documento definisce i componenti minimi (vault dell'input, normalizzatore di puntatore e scroll, mixer di persona coerente per sessione, budget di esposizione comportamentale) e lascia spazio a varianti e fork che ne estendano o specializzino il comportamento in contesti diversi (browser, sistemi operativi, ambienti embedded). Per favorire una crescita meritocratica e decentralizzata, il progetto prevede un sistema non finanziario di riconoscimento dei contributi—HOLoToken—basato su *proof-of-contribution*, utile a dare visibilità e voce a chi costruisce.

Questo whitepaper non presenta un prodotto, ma una regola del gioco: un protocollo minimale per **restituire all'essere umano il diritto a non essere riconosciuto dal proprio comportamento**.

Il resto—codice, ottimizzazioni, porting, integrazioni—appartiene alla community.

2. Il Problema

L'identità digitale di una persona non è composta solo dai dati che fornisce intenzionalmente—nome, indirizzo e-mail, preferenze—ma anche da una serie di segnali involontari che emergono durante l'interazione con un dispositivo. Ogni utente possiede un modo unico di digitare, muovere il puntatore, scorrere una pagina, selezionare un elemento grafico. Questi pattern, inizialmente considerati rumore di fondo, sono oggi riconosciuti come biometria comportamentale: una firma individuale precisa e difficilmente alterabile.

La progressiva diffusione di modelli di Intelligenza Artificiale addestrati su eventi ad alta frequenza ha trasformato questi segnali in uno strumento di riconoscimento estremamente efficace. Bastano pochi millisecondi di differenza tra una pressione dei tasti e l'altra, minuscole variazioni nelle curvature del mouse, micro-pause nel prendere decisioni, per ricostruire un profilo identificativo stabile. A differenza dei cookie—cancellabili—o degli indirizzi IP—mascherabili—la biometria comportamentale è intrinseca alla persona: compare ogni volta che usa un computer.

Il problema non è solo tecnico, ma strutturale. Questi segnali:

- vengono raccolti in modo silenzioso, tramite semplici script di pagina;
- non richiedono consenso esplicito né interazione dell'utente;
- si prestano a essere combinati e correlati tra siti diversi;
- permettono l'identificazione anche in modalità privata, con VPN o attraverso reti anonime;
- vengono utilizzati per classificare stati emotivi, propensione all'errore, indecisione, vulnerabilità.

L'essere umano, senza rendersene conto, rivela al sistema informazioni che non può controllare: il ritmo delle sue mani, la velocità con cui reagisce, la modalità con cui esplora uno schermo. Una piattaforma che osserva questi segnali può distinguere la persona da milioni di utenti apparentemente anonimi, ricostruire la sua presenza nel tempo e associare comportamenti a identità reali anche senza dati espliciti.

A questa vulnerabilità si somma un secondo problema: l'asimmetria di potere. L'utente non ha strumenti nativi per proteggersi. Non esistono browser che schermino il comportamento umano. Le soluzioni disponibili oggi—VPN, modalità anonima, anti-tracking, blocco di terze parti—proteggono il dispositivo, non il corpo digitale. Anche i sistemi più robusti, progettati per difendere l'anonimato a livello di rete o di fingerprinting hardware, non intervengono sulla dimensione comportamentale.

Il risultato è che l'utente è "anonimo" solo in apparenza: la sua identità comportamentale rimane esposta, costante, riconoscibile. Chi osserva può determinare se un accesso proviene dalla stessa persona che ha visitato un altro sito, stabilire abitudini, ritmi circadiani digitali, capacità cognitive, o persino valutare condizioni momentanee come stanchezza, stress o esitazione.

La mancanza di uno strumento per interporre un livello di protezione tra l'essere umano e l'AI che lo osserva costituisce una lacuna critica nell'architettura della privacy moderna. La biometria comportamentale, ormai utilizzata in strumenti antifrode, sistemi di sicurezza, piattaforme commerciali e reti pubblicitarie, rappresenta una forma di osservazione costante che il cittadino non può controllare né ridurre.

Il problema è quindi semplice nella sua definizione ed enorme nelle sue conseguenze:

ogni utente porta con sé un'impronta comportamentale che non può evitare di rivelare, e il web moderno è progettato per coglierla e trasformarla in informazione strutturata.

HOLogram nasce per affrontare precisamente questa vulnerabilità: restituire all'essere umano la possibilità di interagire con l'ambiente digitale senza esporre la propria firma comportamentale. È una proposta per riequilibrare un rapporto oggi sbilanciato, riducendo la superficie di osservazione non volontaria e ripristinando un margine di autodeterminazione nell'interazione quotidiana con il web.

3. Obiettivi di HOLogram

HOLogram nasce con una finalità chiara: restituire all'essere umano il controllo sulla propria identità comportamentale nell'ambiente digitale.

In un contesto dove modelli di AI sono in grado di dedurre informazioni personali osservando *come* l'utente interagisce, e non solo *ciò che comunica*, si rende necessario un nuovo livello di protezione, pensato per difendere la parte più inconsapevole e fragile della presenza online: il comportamento.

Gli obiettivi fondamentali del protocollo possono essere sintetizzati come segue.

3.1 Proteggere l'identità comportamentale dell'utente

HOLogram mira a impedire che i micro-segnali prodotti naturalmente dall'utente (digitazione, movimenti del puntatore, tempi di reazione) vengano utilizzati per:

- identificarlo in modo univoco;
- tracciare la sua presenza su piattaforme differenti;
- ricostruire profili psicologici o modelli di comportamento;
- dedurre caratteristiche sensibili (stress, indecisione, vulnerabilità).

L'obiettivo non è eliminare l'interazione umana, ma impedire che questa venga sfruttata come impronta digitale comportamentale.

3.2 Offuscare senza alterare l'esperienza d'uso

La protezione non deve compromettere l'interazione.

HOLogram si propone di:

- conservare la naturalezza del gesto umano;
- preservare la funzionalità completa delle applicazioni web;
- ridurre la granularità dei dati esposti senza generare movimenti "artificiali";
- evitare che l'offuscamento stesso diventi un segnale identificante.

L'obiettivo è uno strato discreto, invisibile e armonico.

3.3 Introdurre un modello di protezione locale, indipendente e trasparente

HOLogram opera esclusivamente in locale, senza trasmettere dati, senza telemetria e senza dipendere da servizi esterni.

L'obiettivo è garantire:

- autonomia dell'utente;

- assenza di dipendenze da terze parti;
- nessuna raccolta dei segnali comportamentali;
- piena comprensibilità del processo tramite un HUD minimale.

La protezione appartiene all'utente — non al browser, non al sito, non a un fornitore.

3.4 Contrastare la profilazione comportamentale AI-driven

I modelli moderni sono in grado di ricostruire identità, pattern cognitivi e tendenze decisionali analizzando come l'utente interagisce.

HOLogram si propone di:

- ridurre la quantità di "informazione comportamentale utile" esposta;
- impedire correlazioni cross-site;
- abbattere la precisione dei classificatori comportamentali;
- riequilibrare il rapporto tra utente e modelli di osservazione.

L'obiettivo è ridurre l'efficacia della sorveglianza invisibile, non aggirarla con artifici ingannevoli.

3.5 Creare uno standard aperto per la privacy comportamentale

HOLogram non è un prodotto, ma un protocollo.

L'obiettivo è definire un insieme di regole minime, chiare e implementabili da:

- Browser;
- Estensioni;
- sistemi operativi;
- applicazioni;
- comunità open-source;
- contesti ad alta sensibilità (giornalismo, attivismo, PA).

Uno standard aperto aumenta la superficie di protezione e neutralizza la dipendenza da singoli attori.

3.6 Favorire un ecosistema decentralizzato e meritocratico

HOLogram è pensato per evolvere attraverso contributi distribuiti, non tramite un'autorità centrale.Per questo introduce un sistema di riconoscimento non finanziario — HOLoToken (Proof-of-Contribution) — che permette di:

- valorizzare chi partecipa;

- tracciare contributi tecnici in modo trasparente;
- creare ruoli e responsabilità nella comunità;
- supportare un modello aperto, evolutivo e resiliente.

L'obiettivo è che HOLogram cresca per contributo collettivo, come è accaduto nelle grandi infrastrutture open del passato.

3.7 Stabilire un nuovo diritto digitale: la privacy comportamentale

Il fine ultimo è riconoscere e difendere quello che oggi non ha nome ma che incide profondamente sulla libertà personale:

il diritto a non essere riconosciuti dal proprio comportamento online.

HOLogram vuole introdurre un paradigma che sposti l'attenzione dalla protezione del dispositivo alla protezione dell'essere umano, aprendo la strada a un nuovo livello di dignità digitale.

4. Principi Fondativi

HOLogram si basa su una serie di principi che definiscono la sua identità, la sua etica e il suo funzionamento. Questi principi sono la struttura portante del protocollo: guidano il design, limitano ciò che il Layer può fare e stabiliscono ciò che non può — e non deve — mai fare. Sono il punto di contatto tra architettura tecnica, tutela dell'utente e rispetto della libertà digitale.

4.1 Human-Centric Privacy

Il protocollo nasce per proteggere l'essere umano, non il dispositivo che utilizza né la rete attraverso cui comunica.

La biometria comportamentale è una caratteristica personale che emerge involontariamente: difenderla significa difendere la persona.

HOLogram introduce uno strato che opera direttamente dove il comportamento si manifesta, impedendo che venga trasformato in informazione identificante.

4.2 Local-Only & Zero-Retention

Ogni operazione avviene sul dispositivo dell'utente.

HOLogram non invia dati a server esterni, non sincronizza, non memorizza, non conserva cronologie comportamentali, non registra log.

Il Layer è:

- un filtro;
- non un raccoglitore;
- un buffer;
- non un archivio.

Quando il segnale offuscato è stato trasmesso al browser, il Layer non mantiene traccia del dato originale.

4.3 Minimizzazione della Riconducibilità

La protezione non si basa su blocchi drastici, ma su riduzione progressiva della fedeltà dei segnali comportamentali.

Il principio è semplice:

meno granularità = meno possibilità di riconoscimento.

Ogni movimento, ogni pressione, ogni scroll è trasformato mantenendo la funzionalità ma riducendo l'unicità.

4.4 Offuscamento Non Manipolativo

HOLogram non altera l'intenzione dell'utente, non modifica l'input semantico, non crea automatismi finalizzati a influenzare decisioni.

Il Layer deve:

- Proteggere;
- Mascherare;
- Normalizzare.

ma mai guidare l'utente verso azioni diverse da quelle previste.

Questo principio separa nettamente il protocollo da tecniche ingannevoli o manipolative.

4.5 Naturalità dell'Interazione

La protezione deve integrarsi in modo fluido.

HOLogram non può introdurre movimenti artificiali, ritardi innaturali, o pattern che rendano l'utente sospetto ai sistemi antifrode o anti-bot.

L'output deve essere:

- umano;
- realistico;
- coerente;
- compatibile con qualsiasi sito moderno.

Non un comportamento casuale, ma un comportamento non identificante.

4.6 Adaptive Obfuscation

La quantità di offuscamento deve variare in base al contesto, al livello di esposizione e alle preferenze dell'utente.

HOLogram non applica una protezione fissa, calibra il livello utilizzando il Behavioral Privacy Budget, incrementando l'offuscamento quando necessario.

Il protocollo è dinamico, non statico.

4.7 Trasparenza e Controllo dell'Utente

Ogni operazione del Layer deve essere comprensibile e controllabile.

L'utente può visualizzare:

- il livello attuale di protezione;

- il consumo del privacy budget;
- le componenti attive del Layer;
- eventuali cambiamenti automatici del livello di offuscamento.

Il controllo non è totale — in alcuni casi il Layer deve prevenire scelte pericolose — ma è sempre esplicito e consapevole.

4.8 Neutralità Tecnologica

Il protocollo non è legato a:

- un browser specifico;
- un linguaggio di programmazione;
- un sistema operativo.

Deve poter essere implementato:

- come estensione;
- come modulo OS;
- come componente embedded;
- come libreria di terze parti.

La neutralità garantisce la longevità del protocollo.

4.9 Apertura del Protocollo

HOLogram è uno standard aperto, non un software proprietario.

Chiunque può:

- leggere le specifiche;
- implementarle;
- proporre estensioni;
- creare fork;
- contribuire al Layer;
- sviluppare varianti.

La protezione comportamentale deve essere un bene comune.

4.10 Merito come motore evolutivo

Per favorire un ecosistema libero e resiliente, HOLogram integra un sistema di riconoscimento dei contributi (HOLoToken), non finanziario e basato su *proof-of-contribution*.

Il valore non deriva dal denaro, ma dal miglioramento del protocollo.

Ogni contributore lascia una traccia del proprio impegno; il protocollo cresce attraverso un modello decentralizzato, aperto e meritocratico.

4.11 Conservare la Libertà Comportamentale dell'Individuo

Il fine ultimo è proteggere una dimensione non ancora normata ma fondamentale della dignità digitale: la libertà del proprio comportamento.

Nessun sistema dovrebbe poter dedurre identità, vulnerabilità o intenzioni osservando come un individuo si muove o digita.

HOLogram nasce per affermare questo principio.

5. Architettura del Protocollo HOLogram

L'architettura di HOLogram è costruita come uno strato intermedio, locale e trasparente, che si interpone tra i segnali comportamentali generati dall'utente e i sistemi digitali che li osservano. Ogni componente ha la funzione di ridurre la fedeltà del comportamento reale, impedire l'identificazione attraverso pattern biometrici e preservare allo stesso tempo la naturalezza dell'interazione.

Il protocollo si compone di moduli indipendenti, combinabili ed estendibili, progettati secondo tre principi essenziali:

1. offuscare senza degradare l'esperienza;
2. proteggere senza manipolare;
3. operare senza raccolta o trasmissione di dati.

5.1 ShadowDOM Input Vault

La digitazione rappresenta uno dei canali principali di identificazione comportamentale: il ritmo, i tempi di pressione, le pause e le correzioni sono elementi unici e difficili da imitare.

Lo ShadowDOM Input Vault intercetta gli eventi nativi di input (keydown, keyup, keypress), li isola in un contesto protetto e produce una sequenza normalizzata di eventi (input, change) priva della granularità temporale che caratterizza la biometria dell'utente.

In sintesi, il Vault:

- preserva l'intento semantico;
- elimina i micro-tempi legati al ritmo reale;
- impedisce agli script di pagina di accedere ai pattern biometrici di digitazione.

Il risultato è un testo corretto e trasmesso come sempre ma non associabile al modo unico in cui l'utente lo ha digitato.

5.2 Pointer & Scroll Normalizer

Anche i movimenti del puntatore e dello scroll sono sorgenti decisive di riconoscimento: traiettorie, accelerazioni, tremolii, curvature e tempi di reazione rivelano caratteristiche motorie individuali.

Il Pointer & Scroll Normalizer interviene applicando tre trasformazioni:

1. Campionamento uniforme degli eventi ad alta frequenza;
2. Quantizzazione delle coordinate, dei delta di movimento e dell'intensità dello scroll;
3. Smoothing controllato per eliminare irregolarità uniche senza introdurre movimenti artificiali riconducibili a un "bot".

Il risultato conserva la naturalità del gesto umano ma riduce la possibilità di estrarre una firma biometrica.

5.3 Persona Mixer (session-consistent)

L'offuscamento puro può generare un comportamento ripetitivo e riconoscibile nel tempo. Per evitare che "l'offuscamento stesso" diventi impronta, il protocollo introduce il Persona Mixer.

A ogni sessione viene associata una persona sintetica, definita da un insieme di parametri variabili:

- velocità apparente del puntatore;
- fluidità dello smoothing;
- modello di digitazione normalizzata;
- stile di scroll con micro-pause e variabilità minima.

La persona:

- rimane coerente durante la sessione;
- cambia a ogni nuova sessione;
- evita correlazioni longitudinali e fingerprinting dell'offuscamento.

Per evitare che l'offuscamento stesso diventi una nuova impronta, i parametri delle persone sintetiche sono estratti entro range plausibili e parzialmente sovrapposti, con vincoli di variabilità tali da evitare estremi riconoscibili e, al contempo, prevenire regolarità eccessive.
Le classi parametriche (ad es. velocità apparente, smoothing, micro-pause) sono ampiezze controllate: sufficientemente diverse per ridurre la correlazione tra sessioni, ma non così divergenti da risultare fingerprintabili come "stile HOLogram".

5.4 Behavioral Differential Privacy Engine *(Componente tecnica di offuscamento)*

Il *Behavioral Differential Privacy Engine* è ispirato ai principi della privacy differenziale e li adatta a flussi di eventi in tempo reale, con l'obiettivo di ridurre la sensibilità dell'output a micro-variazioni dell'input. Non si tratta dell'applicazione classica su dataset statici, bensì di un approccio operazionale che impiega rumore calibrato e regole di stabilizzazione per limitare l'estrazione di firme biometriche dai segnali comportamentali.

La sua funzione è garantire che piccole variazioni nel comportamento reale dell'utente non producano differenze significative nell'output osservato dai siti.

Opera attraverso:

- Aggiunta di rumore controllato (es. jitter temporale minimo, variazione micro-statistica dei delta);
- Riduzione della sensibilità dei segnali comportamentali;
- Stabilizzazione dei pattern per impedire correlazioni su larga scala;
- Regole deterministicamente variabili per evitare fingerprinting dell'offuscamento stesso.

Il DP Engine rappresenta la base matematica e algoritmica del protocollo, garantendo una protezione formalizzabile e riproducibile.

5.5 Behavioral Privacy Budget (Componente di gestione e controllo)

Il Behavioral Privacy Budget è il livello di supervisione che decide quanto comportamento reale può essere esposto e quando aumentare automaticamente la protezione.

Si tratta di un sistema dinamico che:

- monitora la quantità di informazione comportamentale rivelata;
- valuta la complessità degli eventi esposti;
- applica soglie progressive di protezione;
- intensifica l'offuscamento quando il budget si avvicina al limite;
- permette all'utente di configurare livelli di protezione (basso, medio, alto).

Il Budget agisce "sopra" il DP Engine: il DP Engine regola il *come*, il Budget regola il *quanto*.

Esempio di comportamento dinamico:

- Sessione appena iniziata → offuscamento leggero;
- Interazione intensa e ripetitiva → aumento automatico dello smoothing;
- Rischio di correlazione cross-site → riduzione della frequenza degli eventi;
- Budget quasi esaurito → modalità di protezione massima.

Il *Behavioral Privacy Budget* si rigenera in modo controllato durante la sessione (idle/pause prolungate, eventi a bassa intensità) e si azzera al reset (nuova sessione o cambio dominio), consentendo una protezione dinamica e non punitiva. Il consumo del budget è funzione di frequenza, complessità geometrica (ampiezza, variazione angolare, traiettoria), ed entropia comportamentale del segnale, così da riflettere la quantità di informazione potenzialmente esposta.

5.6 Exposure HUD (Human-Visible Transparency Layer)

HOLogram integra un piccolo layer di trasparenza: l'Exposure HUD, un'interfaccia minimale che mostra:

- livello corrente di protezione;
- stato del Behavioral Privacy Budget;
- attività dei moduli;
- eventuali escalation automatiche dell'offuscamento.

L'obiettivo non è fornire un pannello complesso ma rendere il Layer comprensibile e mantenere il controllo nelle mani dell'utente.

5.7 Optional HID Cloak (livello OS)

Il modulo opzionale HID Cloak estende la protezione oltre il browser, agendo sugli eventi hardware (tastiera, mouse, trackpad) prima che raggiungano qualsiasi applicazione.

Serve in contesti ad alta sensibilità:

- giornalismo investigativo;
- attivismo;
- contesti aziendali avanzati;
- sistemi embedded.

Non è essenziale ma rappresenta un'evoluzione naturale del Layer.

L'integrazione OS-level può richiedere API di accessibilità o componenti a privilegi elevati e potrebbe comportare impatti di compatibilità con driver o policy di sicurezza della piattaforma. Questo modulo è pertanto opzionale e destinato a ambienti controllati (alta sicurezza) con validazione dedicata.

5.8 Sintesi del flusso operativo

Input umano

↓

ShadowDOM Input Vault

↓

Pointer/Scroll Normalizer

↓

Persona Mixer

↓

Behavioral Differential Privacy Engine

↓

Behavioral Privacy Budget

↓

Output offuscato verso browser/siti

Il sistema non altera l'intenzione né modifica il contenuto, trasforma solo la forma comportamentale, proiettando nel web un "ologramma" dell'interazione reale.

5.9 Filosofia architetturale

L'architettura di HOLogram non è monolitica: è un protocollo aperto, pensato per essere:

- Modulare;
- Estensibile;
- Verificabile;
- Replicabile;
- implementabile in molte varianti e fork.

L'unico vincolo sono i Principi Fondativi: protezione, naturalità, località e non identificabilità.

6. Funzionamento Operativo del Sistema

Il funzionamento operativo di HOLogram può essere descritto come un processo di trasformazione dei segnali comportamentali.

L'obiettivo è ridurre la riconducibilità dei pattern e, allo stesso tempo, preservare la naturalezza e la funzionalità dell'interazione umana.

Il sistema opera in cinque fasi principali: acquisizione, normalizzazione, personalizzazione, protezione ed emissione verso l'applicazione finale.

6.1 Acquisizione dei segnali (Input Capture Layer)

Ogni interazione digitale genera una sequenza di eventi: pressione dei tasti, movimenti del puntatore, scroll, click, selezioni, micro-pausa decisionali.

HOLogram intercetta questi eventi prima che raggiungano gli script di pagina:

- eventi di tastiera (keydown, keyup, keypress);
- eventi del puntatore (mousemove, pointermove, mousedown, mouseup);
- eventi di scrolling (wheel, scroll);
- eventi di interazione GUI (focus, blur, hover).

L'obiettivo non è bloccare, ma intercettare e filtrare.

Gli eventi raw sono quindi trasferiti ai moduli successivi, senza mai essere memorizzati né trasmessi.

6.2 Normalizzazione del segnale (Signal Shaping Layer)

La seconda fase riguarda la trasformazione dei segnali in forma meno dettagliata.

Il sistema applica:

- riduzione della frequenza (downsampling);
- quantizzazione delle coordinate;
- smoothing dei movimenti;
- omogeneizzazione dei tempi;
- standardizzazione delle traiettorie minori;
- eliminazione delle micro-variazioni uniche.

Questi processi convertono i segnali comportamentali reali in una forma compatibile, funzionale, ma non identificante.

Il contenuto dell'azione rimane invariato (digitazione, click, selezione), ma la forma con cui si manifesta è stata resa anonima.

6.3 Personalizzazione sintetica (Persona Mixer Layer)

I segnali normalizzati sono poi sottoposti al Persona Mixer, che introduce una variabilità genuinamente umana ma sganciata dall'identità reale dell'utente.

Il Mixer:

- seleziona una "persona sintetica" all'avvio della sessione;
- assegna parametri di comportamento coerenti (velocità, flusso, micro-pause);
- applica variazioni leggere e continue;
- evita qualsiasi regolarità artificiale riconducibile all'offuscamento stesso.

Questo strato produce una "firma comportamentale fittizia" che non può essere attribuita a un individuo reale, né replicata tra sessioni diverse.

6.4 Protezione differenziale (Behavioral Differential Privacy Engine)

Il segnale sintetizzato viene quindi elaborato dal Differential Privacy Engine, che applica logiche ispirate alla privacy differenziale per garantire che ogni variazione minima nel comportamento reale non modifichi sostanzialmente l'output esposto.

In pratica:

- piccole variazioni della digitazione non producono variazioni distinguibili;
- il rumore aggiunto è calibrato per essere invisibile all'utente ma efficace per i sistemi di profilazione;
- il segnale risultante ha proprietà di non-sensibilità → una modifica del comportamento reale non altera significativamente l'esito finale.

Il DP Engine opera come ultima barriera matematica tra l'utente reale e il web.

6.5 Gestione dell'esposizione (Behavioral Privacy Budget Layer)

Il comportamento offuscato è poi valutato dal Privacy Budget, che ha il compito di monitorare quanta "informazione comportamentale" il sito ha già ricevuto.

Se il budget è basso, il sistema:

- aumenta il livello di protezione;
- riduce la frequenza degli eventi;
- intensifica la quantizzazione;
- semplifica le traiettorie;
- limita ulteriori segnali complessi.

Il budget funge da meccanismo dinamico di autodifesa → più una pagina tenta di osservare, più HOLogram riduce ciò che è visibile.

6.6 Emissione verso il browser o l'applicazione (Output Projection Layer)

Una volta elaborato, il segnale finale viene proiettato verso il browser.

Ciò che l'applicazione vede è:

- un movimento umano;
- una digitazione coerente;
- uno scroll naturale;

ma non la vera dinamica comportamentale dell'utente.

Il sito riceve un *ologramma comportamentale* → una proiezione funzionale ma non riconducibile alla persona reale.

6.7 Integrazione con sistemi esterni

L'output di HOLogram è pienamente compatibile con:

- browser moderni;
- framework web;
- sistemi antifrode;
- logiche UI standard;
- applicazioni web complesse.

L'offuscamento avviene nel livello invisibile tra input utente e pagina → ciò che arriva alla piattaforma è perfettamente conforme alle aspettative funzionali.

6.8 Trasparenza verso l'utente (Exposure HUD)

Durante tutto il processo, l'utente può:

- visualizzare il livello di protezione;
- monitorare il budget comportamentale;
- osservare il comportamento del Layer;
- comprendere eventuali escalation automatiche;
- interrompere o aumentare la protezione.

Il protocollo non opera nell'ombra: offusca il comportamento, non la consapevolezza dell'utente.

6.9 Ciclo di vita di una sessione

Una sessione tipica segue questa sequenza:

1. Avvio → generazione della persona sintetica.
2. Input → eventi raw catturati dal Vault.
3. Trasformazione → normalizzazione dei segnali.
4. Caratterizzazione → applicazione del Mixer.
5. Protezione → DP Engine + Budget.
6. Output → eventi compatibili inviati ai siti.
7. Chiusura → eliminazione di ogni stato, reset totale.

Al termine della sessione, nessuna informazione comportamentale persiste.

6.10 Sintesi

HOLogram non modifica ciò che l'utente vuole fare ma come il comportamento viene esposto.

L'interazione rimane reale, la funzionalità intatta, la libertà preservata, ma la parte più vulnerabile dell'essere umano — la biometria comportamentale — non viene più ceduta al web.

7. Sicurezza, Limitazioni e Modelli di Minaccia

La sicurezza di HOLogram non si misura nella sola robustezza algoritmica, ma nella capacità del sistema di ridurre la correlabilità comportamentale in scenari reali, anche in presenza di avversari dotati di visibilità e risorse. Questo capitolo definisce obiettivi di sicurezza, perimetro di fiducia, modello di minaccia, strategie di difesa, limitazioni note e linee di mitigazione.

7.1 Obiettivi di sicurezza

1. Riduzione della riconducibilità: diminuire la capacità di legare una sessione a un individuo sulla base dei pattern di digitazione, puntatore e scroll;
2. Invarianza funzionale: preservare la piena usabilità delle pagine e la naturalezza dell'interazione (nessuna rottura di flussi UI);
3. Non-tracciabilità dell'offuscamento: evitare che i pattern prodotti dal Layer diventino essi stessi un'impronta riconoscibile nel tempo;
4. Assenza di telemetria: impedire ogni forma di tracciamento interno o esterno al Layer (local-only, zero-retention);
5. Resistenza al rilevamento: rendere difficile per uno script lato pagina discriminare con certezza la presenza del Layer.

7.2 Perimetro di fiducia (Trust Boundary)

- Trusted:
  - Il dispositivo dell'utente come ambiente di esecuzione locale;
  - Il codice di HOLogram (estensione/proxy/driver), verificato dall'utente;
  - La configurazione del Layer (policy, livelli, HUD).
- Untrusted:
  - Le pagine e gli script caricati dal web;
  - Infrastrutture di tracciamento comportamentale;
  - Terze parti (CDN, advertising, analytics);
  - Eventuali osservatori di rete che inferiscono pattern temporali di input/traffic shaping.

7.3 Modello di minaccia (avversari e capacità)

1. Fingerprinter lato pagina:
   - Inietta listener ad alta frequenza (keydown, mousemove, wheel) per estrarre feature temporali/geometriche;

- Applica classificatori su micro-pattern per re-identificare l'utente;
- Tenta di rilevare l'offuscamento (es. precisione temporale ridotta, quantizzazione coordinate).

2. Piattaforma con correlazione cross-site:
   - Condivide feature tra più domini (o usa terze parti comuni) per costruire identità persistenti;
   - Combina segnali comportamentali con fingerprint di device/rete.
3. Anti-bot/anti-abuse aggressivo:
   - Classificatori orientati a distinguere "umano reale" vs "automazione mascherata";
   - Penalizza jitter "non umano" o schemi troppo regolari.
4. Osservatore di rete *(fuori scopo, ma rilevante)*:
   - Non accede agli eventi di input, ma può inferire pattern temporali dai tempi di richiesta/risposta;
   - Non è obiettivo primario di HOLogram, che agisce a livello input/DOM.
5. Avversario mirato (alta risorsa):
   - Combina fingerprint classico, segnali comportamentali residui, contesto di login, telemetria del browser;
   - Esegue test multi-passo per rilevare la presenza del Layer.

7.4 Strategie difensive del protocollo

- Separazione e riduzione delle feature:
  - *ShadowDOM Input Vault*: sottrae i micro-tempi di digitazione; la pagina vede contenuto, non ritmo;
  - *Pointer & Scroll Normalizer*: downsampling, quantizzazione, smoothing controllato.
- Variazione coerente (session-consistent):
  - *Persona Mixer*: introduce variabilità "umana plausibile" coerente nella singola sessione ma diversa tra sessioni, prevenendo correlazioni longitudinali e fingerprint dell'offuscamento.
- Protezione formale e controllo dinamico:
  - *Behavioral Differential Privacy Engine*: riduce la sensibilità dell'output a piccole variazioni dell'input (rumore minimo, regole stabilizzanti);
  - *Behavioral Privacy Budget*: controlla la quantità complessiva di informazione rivelata; intensifica la protezione quando l'esposizione cresce.

- Trasparenza verso l'utente:
    - *Exposure HUD*: consente di comprendere e, se necessario, aumentare la protezione.

7.5 Rilevabilità e anti-rilevazione

Rischio: uno script tenta di dedurre la presenza di HOLogram misurando precisione temporale, granularità coordinate, regolarità eccessiva o differenze tra keydown/keyup attesi e input/change ricevuti.

Contromisure:

1. Coerenza semantica: fornire eventi che rispettano la "shape" delle API; nessun errore di semantica/event bubbling;
2. Stocasticità entro range umani: parametri del Mixer scelti entro intervalli di plausibilità (velocità, micro-pause, curvature);
3. Variabilità policy-driven: pattern di quantizzazione e smoothing non costanti tra sessioni e siti (entro limiti naturali);
4. Mascheramento soft della precisione: la riduzione della risoluzione temporale/coordinate avviene con jitter sottosoglia percezione.

Pur agendo sul perimetro input/DOM, il Layer non elimina tutti i canali laterali: pattern temporali di rete (sequenze richiesta/risposta) possono, in alcuni casi, rivelare correlazioni indirette. Questi vettori sono fuori scopo primario e richiedono strumenti complementari (p.es. reti anonime, traffic shaping) quando necessari.

7.6 Interazione con sistemi anti-bot

Rischio: normalizzazione e smoothing eccessivi potrebbero sembrare "robotici".

Mitigazione:

- *Soglie conservative* di smoothing e quantizzazione;
- *Persona Mixer* per introdurre micro-irregolarità "umane";
- *Profilo per sito*: modalità "compatibilità" su domini noti per anti-bot aggressivi;
- *Fallback graduale*: se la pagina segnala problemi, ridurre l'intensità dell'offuscamento per mantenere la funzionalità.
- In contesti ad alta sensibilità temporale (p.es. gaming, grafica interattiva, trading in tempo reale), il Layer può ridurre automaticamente l'offuscamento (profilo *compatibilità*) per preservare la reattività percepita, mantenendo comunque una protezione minima contro la profilazione comportamentale.

7.7 Correlazione cross-site

Rischio: anche dopo l'offuscamento, pattern consistenti potrebbero correlare sessioni su domini differenti.
Mitigazione:

- *Persona per sessione e per eTLD+1*: resetting del Mixer per dominio/sessione;
- *Budget per sito*: esposizione limitata e indipendente tra domini;
- *Variabilità dei parametri*: ampie classi di persone sintetiche, senza "modello universale".

7.8 Attacchi di adattamento (adversarial adaptation)

Rischio: modelli che imparano a "invertire" l'offuscamento (es., riconoscere la quantizzazione).

Mitigazione:

- *Policy evolutive*: aggiornare periodicamente parametri e funzioni di smoothing/quantizzazione;
- *Set casuali controllati*: multi-strategie selezionate al volo entro famiglie equivalenti in termini di usabilità;
- *Test di regressione comportamentale*: benchmark interni che misurano la diminuzione di accuratezza dei classificatori noti (senza incorporare dataset personali).

7.9 Limitazioni note

1. Non anonimizza il traffico di rete: HOLogram opera sull'input; per anonimato di rete servono altri strumenti (Tor/VPN);
2. Non impedisce il fingerprinting non comportamentale: UA, canvas, font, WebGL richiedono difese di browser privacy-first;
3. Compatibilità variabile: alcune web-app con interazioni altamente temporizzate (es. gaming o editor grafici complessi) possono richiedere profili "compatibilità" o whitelist;
4. Persistenza di segnali residuali: anche con offuscamento, tracce statistiche minori possono rimanere (obiettivo del protocollo: renderle insufficienti a correlazione affidabile);
5. Threat model locale: non protegge da malware sul dispositivo, keylogger a livello OS o osservatori con accesso fisico.

7.10 Scelte di sicurezza e impatto su UX

- Downsampling/Quantizzazione: lieve impatto su micro-reattività; non percepibile nella maggioranza dei flussi UI;
- Smoothing: bilanciato per evitare "inerzia" del puntatore;
- Persona Mixer: variazioni minime per preservare la naturalezza;

- Budget dinamico: escalation graduale, con preferenza per la continuità d'uso rispetto a blocchi bruschi.

7.11 Test, validazione e audit

- Test di compatibilità: suite cross-browser su set di applicazioni reali (form, editor, SPA, canvas);
- Test di efficacia: confronto pre/post HOLogram su classificatori comportamentali noti (senza dataset personali), misurando il calo di accuratezza;
- Audit di codice: revisione indipendente e verifiable build per estensioni/driver;
- Misure di trasparenza: changelog pubblico delle policy, versione del DP Engine, report delle regressioni.

La suite di test include varianti cross-browser che tengono conto di differenze nella frequenza e sequenza degli eventi (PointerEvents vs MouseEvents, gestione dello scroll e del throttling nativo), assicurando la coerenza semantica dell'output e la plausibilità umana a parità di contesto.

7.12 Roadmap difensiva

- v0: parametri conservativi, profilo unico;
- v1: profili per sito, HUD avanzato, budget per dominio;
- v2: famiglie di funzioni di smoothing/quantizzazione con selezione casuale controllata;
- v3: testing automatico contro suite di classificatori comportamentali aggiornati; policy adattive per contesti ad alto rischio.

7.13 Sintesi

HOLogram non promette invisibilità assoluta; promette riduzione sistematica della riconducibilità e ri-equilibrio del rapporto tra persona e piattaforme osservanti. Le sue difese sono progettate per resistere ai tracciatori comportamentali senza spezzare l'esperienza d'uso, assumendo che gli avversari si adattino e che il protocollo debba evolvere di conseguenza. La sicurezza di HOLogram è, per progettazione, un processo continuo e trasparente.

8. Compliance & Aspetti Legali

HOLogram è progettato come un meccanismo di protezione dell'identità comportamentale e non come un sistema di raccolta, analisi o conservazione di dati. La sua natura è puramente trasformativa: intercetta i segnali comportamentali generati dall'utente, li normalizza localmente e li inoltra senza memorizzarli né trasmetterli. Per questo, il protocollo non configura un trattamento di dati personali e non introduce nuovi rischi per la privacy: al contrario, nasce per ridurli.

Di seguito sono descritti gli aspetti di conformità che guidano la progettazione del Layer.

8.1 Assenza di trattamento dati

HOLogram non raccoglie, non conserva, non archivia e non trasmette alcuna informazione relativa all'utente.

Nello specifico, il sistema:

- non salva dati biometrici o comportamentali;
- non genera log persistenti;
- non registra alcun identificatore;
- non utilizza telemetria;
- non invia dati a server remoti o terze parti.

Ogni segnale viene elaborato solo in memoria volatile, trasformato e immediatamente rilasciato verso il browser, senza lasciare traccia.

8.2 Conformità ai principi generali di tutela

Sebbene HOLogram non gestisca dati personali, la sua architettura rispetta volontariamente i principi più elevati di tutela e sicurezza:

• Minimizzazione: riduce la quantità di informazioni comportamentali esposte ai siti;

• Limitazione della finalità: la funzione del Layer è una sola: proteggere, non raccogliere;

• Integrità e riservatezza: nessun dato esce dal dispositivo; nessuna copia è conservata;

• Privacy by design: il protocollo nasce per prevenire il trattamento anziché gestirlo.

8.3 Neutralità e assenza di manipolazione

HOLogram non altera il comportamento intenzionale dell'utente, non introduce bias, non modifica decisioni, non suggerisce alternative e non esercita alcuna forma di influenza.

Il Layer si limita a:

- filtrare i segnali comportamentali;
- ridurne la granularità;
- impedirne l'uso per identificazione;
- proiettare un'interazione funzionale ma non riconducibile.

Il protocollo non manipola e non valuta l'utente.

8.4 Assenza totale di profilazione

HOLogram non crea profili, non misura preferenze, non deduce caratteristiche emotive o cognitive, non costruisce cluster e non effettua inferenze.

L'unico elemento che il protocollo genera — HOLoToken — è:

- non personale;
- non comportamentale;
- non trasferibile;
- non economico;
- non legato all'utente finale.

HOLoToken è una misura interna e non finanziaria del contributo tecnico dei partecipanti allo sviluppo del protocollo stesso. Non ha alcuna relazione con l'identità o il comportamento dell'utente protetto dal Layer.

8.5 Trasparenza verso l'utente

HOLogram non opera nell'ombra: include un livello di trasparenza (Exposure HUD) che consente all'utente di:

- visualizzare in tempo reale il livello di offuscamento;
- comprendere la funzione attiva del Layer;
- regolare il comportamento del sistema;
- aumentare o ridurre la protezione secondo necessità.

L'utente resta sempre in controllo del protocollo che lo protegge.

8.6 Compatibilità con i contesti ad alta tutela

Il protocollo può essere utilizzato in scenari sensibili:

- Giornalismo;

- Attivismo;
- navigazione privata;
- contesti aziendali avanzati;
- protezione dei minori;
- ambienti esposti a sorveglianza comportamentale.

Grazie alla sua natura locale e priva di trattamento, HOLogram non introduce rischi aggiuntivi in alcuno di questi contesti.

8.7 Limitazioni giuridiche e implementative

Il protocollo definisce lo standard tecnico, non la sua implementazione specifica.

Eventuali implementazioni di terze parti potrebbero:

- introdurre funzionalità aggiuntive;
- raccogliere o sincronizzare dati;
- collegare HOLogram a servizi esterni.

In tali casi, la responsabilità della conformità ricade sull'implementazione, non sul protocollo originario.

Il Whitepaper rimane neutrale e indipendente.

8.8 Ruolo del protocollo nella tutela dei diritti digitali

HOLogram tutela una dimensione fondamentale e spesso trascurata: la libertà di non essere riconosciuti dal proprio comportamento. Il protocollo non limita funzionalità, non riduce diritti e non introduce obblighi. Offre uno strumento che permette all'utente di controllare ciò che, nel web contemporaneo, è normalmente invisibile e incontrollabile: l'esposizione della propria biometria comportamentale.

8.9 Riferimenti normativi — nota non vincolante

Pur non configurando un sistema di trattamento dati personali, HOLogram è progettato per aderire ai principi generali di tutela espressi nelle principali normative europee sui diritti digitali (es. AI Act, GDPR, ePrivacy). Tale allineamento è ottenuto tramite elaborazione esclusivamente locale, assenza di conservazione, riduzione dell'esposizione comportamentale e neutralità funzionale. Il protocollo non interpreta né sostituisce tali normative: opera in un perimetro tecnico che ne rispecchia i principi fondamentali senza generare trattamenti o basi giuridiche.

8.10 Sintesi operativa

HOLogram è conforme perché:

- opera interamente in locale;
- non memorizza dati;
- non effettua trattamento;
- non genera profili;
- non classifica l'utente;
- non modifica intenzioni;
- non introduce telemetrie;
- non produce rischi.

È un meccanismo di tutela, non un sistema di analisi.

9. Ecosistema & Standardizzazione di HOLogram

HOLogram non è un'applicazione monolitica: è un protocollo. Per essere utile, deve vivere in un ecosistema fatto di implementazioni multiple, fork, varianti, librerie e integrazioni, tutte allineate ai Principi Fondativi.

Questa sezione definisce chi fa cosa, quali artefatti rilasciare, come cooperare e come convergere verso uno standard aperto.

9.1 Obiettivo dell'ecosistema

- Favorire adozione rapida su più piattaforme (browser, OS, ambienti embedded);
- Garantire coerenza minima tra varianti (stessa semantica del protocollo);
- Sostenere innovazione distribuita (fork/varianti specializzate);
- Conservare interoperabilità e verificabilità (reference tests e policy pubbliche);
- Avanzare verso la standardizzazione senza sacrificare l'apertura.

9.2 Ruoli nell'ecosistema

- Autore del protocollo (BaffiSan): custodisce la visione, mantiene il whitepaper e la semantica del protocollo;
- Maintainer di reference implementation: curano la *prima* implementazione libera, minimale, verificabile;
- Implementatori indipendenti: producono estensioni, moduli OS, SDK, plugin, porting, propongono varianti;
- Verificatori (QA & security): gestiscono test di compatibilità, validano profili, eseguono audit del codice;
- Community contributor: sviluppano feature, fix, documentazione, traduzioni, benchmark, RFC;
- Integrator (browser, OS, vendor): adottano HOLogram nativamente o come opzione "privacy layer".

9.3 Artefatti ufficiali

1. Whitepaper HOLogram v1.x:
    - sorgente semantico del protocollo (questo documento).
2. Reference Implementation (RI):
    - estensione browser minimale + test suite;
    - codice pulito, commentato, riproducibile (build verificabile).

3. Specifica Tecnica (HLP – HOLogram Layer Protocol):
    - definizioni di interfaccia, eventi, parametri, semantica dei moduli (Input Vault, Normalizer, Persona Mixer, DP Engine, Privacy Budget, HUD).
4. Test di Conformità (HCT – HOLogram Compliance Tests):
    - set di prove riproducibili che convalidano la conformità di una build alla semantica del protocollo.
5. Reputation Policy (HRL/HCS):
    - regole pubbliche, versionate, per l'assegnazione HOLoToken (non finanziario).
6. Security/Threat Model Notes:
    - minacce, regressioni, contromisure, livelli di protezione consigliati.
7. Ecosystem Registry:
    - elenco pubblico di implementazioni, versioni, stato conformità, badge.

9.4 Processo RFC (proposte di cambiamento)

- Le modifiche al protocollo passano tramite RFC pubbliche (testo breve + motivazione + impatto);
- Ogni RFC indica: *semantica modificata*, *moduli impattati*, *compromessi di UX/sicurezza*, *piano di migrazione*;
- Quorum tecnico: approvazione richiede revisione di maintainer RI + almeno un verificatore sicurezza + consenso della community (discussione aperta);
- Le RFC approvate diventano HLP vX.Y; le implementazioni segnalano compatibilità (HLP vX.Y-compat).

9.5 Interoperabilità & Conformità

- Ogni implementazione dichiara:
    - Versione HLP supportata (es. HLP 1.2);
    - Modulo coperti (Input Vault/Normalizer/Persona/DP/Privacy Budget/HUD);
    - Limitazioni note (modalità compatibilità per siti complessi).
- Il set HCT include:
    - test funzionali (es. coerenza eventi, shape API);
    - test di "human plausibility" (parametri entro range naturali);
    - test di sicurezza (assenza di *leakage* banali, anti-rilevazione basilare);

- test di non-regressione rispetto a versioni precedenti.

- Le build che superano HCT ottengono badge “HOLogram-Compliant (HLP X.Y)”.

9.6 Varianti & Fork

Le varianti sono implementazioni con scopo specifico (mobile-first, high-security, enterprise, embedded).

I fork sono deviazioni più ampie, sperimentali o di lungo periodo.

Linee guida:

- Compatibilità semantica: le varianti dovrebbero restare conformi a una versione HLP (o dichiarare dove divergono);
- Trasparenza: documentare parametri, profili, differenze rispetto a RI;
- Rientro nel trunk: se una variante dimostra benefici chiari e generalizzabili, proporre una RFC per standardizzarne gli aspetti portabili.

9.7 Integrazione con HOLoToken (HRL/HCS)

- Ogni repo/implementazione pubblica gli eventi di contributo (PR, patch, test, docs, review).
- L’HRL raccoglie e attesta contributi, applica policy versionate e calcola HCS (metrica reputazionale);
- HOLoToken è il *nome* del sistema reputazionale (non finanziario); abilita badge e ruoli nel progetto (es. Reviewer, Maintainer, Docs Steward);
- Il punteggio non ha valore economico, non è trasferibile; è solo merito tecnico, visibile e verificabile.

9.8 Percorso di Standardizzazione

1. Fase 0 — Comunità
    - Pubblicazione Whitepaper; RI minimale; HCT iniziale; raccolta feedback.
2. Fase 1 — Draft Tecnico (HLP 1.x)
    - Specifiche stabili per moduli base; suite test maturata; adozione in più implementazioni.
3. Fase 2 — Gruppo di Lavoro
    - Costituzione di un gruppo aperto (es. Working Group informale) con maintainer, ricercatori, integrator.
4. Fase 3 — Proposta a organismi di standardizzazione

- Presentazione di HLP come proposta di standard aperto presso sedi idonee (comunità e fondazioni che promuovono privacy e interoperabilità).

5. Fase 4 — Stabilizzazione & Governance
    - Regole chiare di evoluzione semantica, versioning, test, compatibilità retroattiva, migrazioni.

La standardizzazione non deve bloccare l'innovazione: il *trunk* di HOLogram resta libero; le norme aiutano solo a mettere d'accordo implementatori diversi.

9.9 Integrazioni di riferimento (adozione pratica)

- Browser: estensioni verificate, o modalità nativa "HOLogram Mode" per profili sensibili;
- OS: modulo HID Cloak (opzionale) per proteggere app native; "per-app policy" integrata;
- DevTool: librerie per testare la "percezione umana" dell'output offuscato (evitare segnali da bot);
- PA/Enterprise: policy consigliate per contesti regolati (profili conservativi, whitelist etica, audit interno);
- Formazione & ricerca: dataset sintetici (non personali) per valutare resilienza di classificatori comportamentali all'azione del Layer.

9.10 Trasparenza, Audit & Sicurezza

- Changelog pubblico di parametri, funzioni di smoothing/quantizzazione e regole DP;
- Verifiable build per RI ed estensioni; riproducibilità delle release;
- Audit periodici (indipendenti) su codice e policy; traccia delle mitigazioni;
- Security-response definita (tempi, canali, embargo, ringraziamenti reputazionali).

9.11 Governance leggera (meritocratica e decentralizzata)

- Processo RFC come fonte di verità per l'evoluzione;
- Ruoli derivati da HCS (HOLoToken) + elezioni tecniche per responsabilità critiche;
- Nessuna autorità centrale proprietaria; la semantica resta nel Whitepaper; l'attuazione è della community;
- Compatibilità e pluralità: implementazioni diverse, stesso linguaggio di protezione.

9.12 Tabella di marcia (indicativa)

- T0 — Rilascio *Whitepaper v1.0* + RI minima + HCT v0 + HRL/HCS policy v0;

- T0+3 mesi — Porting cross-browser + aggiunta profili per sito + HUD avanzato + HCT v1;
- T0+6 mesi — Gruppo di lavoro informale, proposta HLP 1.1, CI di conformità pubblica, audit di sicurezza 1;
- T0+9 mesi — Implementazioni OS/HID opzionali, estensioni enterprise, HCT v2 con benchmark comportamentali;
- T0+12 mesi — Draft di standard aperto (HLP 1.2), pacchetto “Ecosystem Registry”, audit di sicurezza 2, roadmap 24 mesi;

9.13 Sintesi

L’ecosistema HOLogram mette al centro l’utente e la comunità. Il protocollo definisce un linguaggio comune; le implementazioni lo rendono vivo; i test lo mantengono coerente; la reputazione (HOLoToken) riconosce chi costruisce. La standardizzazione è il naturale approdo di un percorso che resta, fin dall’inizio, aperto, meritocratico, verificabile.

10. Visione a Lungo Termine

La nascita di HOLogram rappresenta l'introduzione di un nuovo livello dell'esperienza digitale: un livello dedicato non all'infrastruttura, non alle reti, non ai dispositivi, ma all'essere umano come soggetto da proteggere. Se i primi protocolli di Internet sono nati per connettere macchine, e quelli successivi per collegare identità digitali, HOLogram inaugura una terza era: la protezione della presenza comportamentale dell'individuo.

La visione a lungo termine non si limita all'implementazione tecnica attuale. Il protocollo è concepito per evolvere, ramificarsi, adattarsi all'ambiente digitale in trasformazione, e soprattutto per resistere. Ciò che segue definisce le traiettorie evolutive fondamentali che guidano il futuro di HOLogram.

10.1 Proteggere la libertà comportamentale come diritto digitale

Nel web contemporaneo, la parte più vulnerabile dell'utente non è ciò che afferma, ma ciò che fa. Il ritmo della digitazione, le esitazioni, i micromovimenti del puntatore, le abitudini d'interazione: tutti questi elementi possono essere osservati, interpretati, correlati.

La visione di HOLogram è trasformare quella che oggi è una vulnerabilità strutturale in un diritto riconosciuto:

- il diritto a non essere identificati dal proprio comportamento;
- il diritto a mantenere il controllo dei propri segnali involontari;
- il diritto a interagire senza essere analizzati oltre misura.

HOLogram non è un prodotto: è una dichiarazione di autonomia comportamentale dell'individuo nell'era dell'AI.

10.2 HOLogram come strato nativo del Web

Nella prospettiva evolutiva, HOLogram può diventare uno strato intrinseco del web, analogamente a:

- TLS per la sicurezza;
- CSP per il controllo dei contenuti;
- SameSite per la protezione dei cookie;
- sandboxing per gli script.

L'obiettivo finale non è estendere un browser o fornire una libreria, ma far sì che i browser, in futuro, implementino HOLogram come componente di sistema, con:

- normalizzazione comportamentale integrata;
- protezione nativa dei segnali;
- controlli granulari gestiti dall'utente;
- modalità avanzate per contesti ad alta tutela.

Un futuro in cui “HOLogram Mode” sia una funzionalità standard, al pari della navigazione privata.

10.3 Evoluzione verso standard aperti (HLP – HOLogram Layer Protocol)

La longevità del protocollo non dipende da una singola implementazione, ma dalla sua capacità di diventare un linguaggio comune.

Il futuro vede HOLogram evolvere in:

- uno standard aperto;
- un protocollo versionato;
- una semantica condivisa tra implementazioni indipendenti;
- un riferimento per browser, OS e soluzioni di privacy-by-design.

Questo percorso non richiede autorità centrale: richiede chiarezza, ordine, test condivisi e un processo RFC meritocratico.

10.4 Ecosistema di varianti, fork e innovazione distribuita

Come per tutti i protocolli realmente aperti, la forza di HOLogram risiede nella sua pluralità. Nel lungo periodo, l’ecosistema tenderà a svilupparsi in:

- varianti ottimizzate (mobile, enterprise, sicurezza avanzata);
- fork sperimentali con approcci innovativi all’offuscamento;
- implementazioni OS-level integrate nella gestione degli eventi hardware;
- tool complementari per test, audit, benchmarking;
- librerie per piattaforme emergenti (AR/VR, edge computing, IoT).

Il ruolo del Whitepaper è mantenere un punto di riferimento che assicuri la coerenza concettuale, non uniformità totale.

10.5 Adattamento ai nuovi modelli di AI comportamentale

I sistemi di intelligenza artificiale saranno sempre più sofisticati nella lettura dei segnali comportamentali.

Per questo HOLogram deve essere:

- Adattivo;
- Iterativo;
- capace di evolversi in risposta ai nuovi attacchi;
- aperto a contributi scientifici;

- in grado di modificare regole DP e smoothing con politiche versionate.

La visione è un protocollo che cresce insieme ai modelli che deve mitigare — mai statico, mai concluso.

10.6 Decentralizzazione attraverso HOLoToken (HRL/HCS)

Il futuro del protocollo non dipende da un'entità centrale ma dalla comunità che lo costruisce.

HOLoToken — inteso come sistema reputazionale non finanziario — permette di:

- attribuire merito a chi contribuisce;
- strutturare ruoli e responsabilità;
- stabilire processi decisionali decentralizzati;
- mantenere una governance leggera ma solida.

Nel lungo periodo, il sistema reputazionale può rendere HOLogram:

- indipendente da singoli individui;
- resistente ai conflitti interni;
- sostenuto da un ecosistema di manutentori verificabili;
- in grado di portare avanti la sua missione anche senza la presenza del suo originale ideatore.

10.7 Integrazione trasversale nei futuri ambienti digitali

La protezione comportamentale sarà rilevante non solo nel web tradizionale, ma in:

- interfacce vocali;
- sistemi AR/VR immersivi;
- ambienti di lavoro remoti;
- applicazioni assistite da AI;
- dispositivi IoT con input sensibili;
- sistemi di identity-less authentication.

La visione a lungo termine è un protocollo che vive al di sopra dei dispositivi, al di sopra delle interfacce, al di sopra delle piattaforme.

Ovunque ci sia un comportamento umano misurabile, HOLogram può costituire una forma di tutela.

10.8 Missione finale

La missione di HOLogram è semplice ma radicale:

proteggere l'essere umano dal diventare un oggetto misurabile.

Un web in cui:

- la persona può interagire senza essere scomposta in micro-pattern;
- i modelli non possono estrarre identità invisibili;
- la biometria comportamentale non è più destinata a diventare un'impronta involontaria;
- la libertà di agire non implica essere osservati oltre la propria intenzione.

La visione ultima è un ecosistema digitale in cui la protezione comportamentale è un diritto naturale, e HOLogram è il protocollo fondamentale che permette di esercitarlo.

10.9 Conclusione

HOLogram non è solo un Layer tecnico. È un cambio di paradigma: dall'osservazione del comportamento alla libertà del comportamento. La sua ambizione non è sostituire il web, ma bilanciarlo, riportando l'individuo al centro del rapporto tra esseri umani e intelligenze artificiali.

È un protocollo progettato per durare, per evolvere e per essere adottato senza barriere.

Un protocollo che può crescere grazie alla comunità, e che può rimanere vivo anche oltre chi lo ha concepito.

11. HOLoToken — HRL & HCS (Proof-of-Contribution, non finanziario)

11.1 Scopo e principi

HOLoToken è il meccanismo di riconoscimento meritocratico dell'ecosistema HOLogram.

Nasce per:

- valorizzare i contributi tecnici e non tecnici;
- incentivare qualità, manutenzione e sicurezza del protocollo;
- decentralizzare l'evoluzione tramite regole chiare, verificabili e pubbliche;
- evitare ogni natura economica o speculativa.

Non è una valuta, non è trasferibile, non è scambiabile, non dà diritto a rendimenti né rappresenta asset. È un indicatore di merito, registrato e verificabile.

11.2 HRL — HOLogram Reputation Layer (il "sistema")

HRL è lo strato reputazionale del protocollo, responsabile di raccogliere, verificare, calcolare, registrare ed esporre la reputazione nell'ecosistema, secondo una logica *Proof-of-Contribution*.

Responsabilità dell'HRL:

1. Raccolta eventi di contributo:

   commit, patch, test, fix di sicurezza, documentazione, traduzioni, triage, benchmark ripetibili, revisioni tecniche, RFC;

2. Verifica & attestazione:

   integrazioni CI, firme dei maintainer, code-owners, quorum di review, tracciabilità dei revert;

3. Policy di calcolo:

   pesi, bonus/malus, decadimento temporale, anti-gaming (pubblici e versionati);

4. Calcolo & ricalcolo:

   batch o near-real-time, con log di audit e motivazioni;

5. Registry reputazionale:

   ledger pubblico (non finanziario) con stato, storico, metadati e versioni delle policy;

6. Esposizione via API:

   endpoint minimi per consultazione punteggi, badge e attestazioni;

7. Portabilità:

   esportazione dei profili reputazionali; badge/medaglie firmate; feed per terze parti.

Garanzie:

1. Trasparenza: policy pubbliche, versionamento, changelog;
2. Imparzialità: criteri deterministici, quorum di revisione, possibilità di *appeal*;
3. Privacy-by-design: nessuna raccolta di dati personali oltre lo stretto necessario alla verifica dei contributi.

11.3 HCS — HOLogram Contribution Score (la "metrica")

HCS è il valore che misura il merito di ciascun contributore.

Può essere:

- scalare (HCS totale);
- vettoriale (dimensioni: *security*, *core-code*, *docs*, *i18n*, *QA/bench*, *research*, ...).

L'HCS abilita:

- badge e ruoli (Bronze/Silver/Gold; Maintainer; Security reviewer; Docs steward, ...);
- permessi (es. voto tecnico su RFC di sicurezza sopra soglia *HCS_security*);
- priorità (es. code-review queue; accesso a canali tecnici).

Il layer HRL "fa funzionare" il sistema; l'HCS "racconta" il merito.

11.4 Classi di contributo (catalogo base)

- Code Patch (S/M/L; core vs plugin);
- Security Fix (con test, CVE o advisory);
- Test Suite (coverage, stabilità, flake-rate);
- Documentation (architettura, API, guide, esempi);
- Internationalization (i18n) (traduzioni con QA);
- Benchmark & QA (ripetibili, con setup descritto);
- Triage & Repro (issue riprodotta, riduzione casi);
- RFC & Review (proposte e revisioni approvate);
- Research Notes (analisi, threat model, mitigazioni).

(Il catalogo è estendibile via RFC.)

11.5 Meccanismo di calcolo (Proof-of-Contribution)

Esempio sobrio; i valori sono segnaposto e vanno calibrati dalla community.

11.5.1 Punteggio base per evento

- Code Patch (M): +15
- Security Fix (M): +40 *(+10 se CVE/advisory)*
- Test Suite (complete): +10
- Documentation (S/M): +5 / +8
- i18n (M, con QA): +8
- RFC approvata: +18
- Review tecnica approvata (con quorum): +12
- Triage con riproduzione: +10
- Benchmark ripetibile: +18

11.5.2 Bonus/Malus & vincoli

- Review quorum: +0→+20 (in base al numero/ruolo dei reviewer)
- Revert: −x% del guadagno collegato all'evento ripristinato
- Decay: −y% dopo 12 mesi (mantiene "frescura" del profilo)
- Anti-gaming:
    - tetto giornaliero per micro-eventi ripetitivi
    - penalità per frammentazione artificiale di patch
    - rifiuto di contributi non sostanziali (*spam/low-value*)

11.5.3 Vettorialità

Oltre all'HCS_totale, si calcolano componenti per categoria:

HCS = <security, core, docs, i18n, QA, research, ...> → consente permessi granulari (es. voto su RFC di sicurezza con *HCS_security* ≥ soglia).

11.5.4 Trasparenza & ricorso

- ogni variazione HCS include motivo, policy version, hash evento;
- canale di appeal con esito tracciato nel ledger reputazionale.

11.6 Badge, ruoli, permessi (esempi)

- Contributor Bronze: HCS_tot ≥ 200
- Contributor Silver: HCS_tot ≥ 600
- Contributor Gold: HCS_tot ≥ 1200
- Security Reviewer: HCS_security ≥ 300 + quorum nomina
- Maintainer Core: HCS_core ≥ 800 + election tecnica

- Docs Steward: HCS_docs ≥ 250 + milestone docs

(Soglie e ruoli sono parametrici e deliberati via RFC.)

11.7 API minime (esportazione & integrazione)

Endpoint di sola lettura (indicativi):

```
GET /holo/api/v1/profile/{id}        # HCS totale + vettori + badge correnti
GET /holo/api/v1/history/{id}        # variazioni HCS con motivazioni
GET /holo/api/v1/leaderboard?dim=security&n=100
GET /holo/api/v1/badge/{id}          # attestazione firmata del badge
GET /holo/api/v1/policy/version      # versione corrente delle policy HRL
```

Consistenza forte non richiesta: è sufficiente eventual consistency (ricalcoli periodici).

11.8 Governance del sistema reputazionale

- Policy pubbliche e versionate (semantica: *policy vX.Y*);
- Processo RFC per proporre modifiche (nuovi pesi, nuove classi);
- Quorum tecnico per approvazioni critiche (es. sicurezza);
- Audit: log firmati delle decisioni e dei ricalcoli;
- Compatibilità retroattiva: migrazioni con regole documentate;
- Nessun potere discrezionale singolo: le policy prevalgono sulle persone.

11.9 Sicurezza & anti-abuso

- Sybil-resistance soft: legame a identità di contributo verificabili (firma commit, storico repo, review incrociate);
- Detezione anomalie: pattern di contributi artificiali, burst sospetti, ripetitività non motivata;
- Revert-aware: ricalcolo HCS al ripristino;
- Rate-limit & caps: limiti per categoria e periodo;
- Appeal process: evita penalità permanenti per errori onesti.

11.10 Privacy & conformità

- HRL tratta solo eventi di contributo e metadati minimi necessari alla verifica;
- Nessuna *profilazione personale*; nessun dato sensibile; nessuna telemetria comportamentale utente;

- Ledger reputazionale non contiene informazioni private; solo attestazioni tecniche;
- L'HCS non è utilizzabile come fattore d'identità reale, ma solo come merito tecnico nell'ecosistema.

11.11 Limiti e scopi

- HOLoToken non è denaro, non è un investimento, non è trasferibile o convertibile;
- È un indicatore reputazionale: misura e certifica l'apporto al protocollo;
- Il suo scopo è abilitare ruoli, permessi, priorità e riconoscimenti interni, non creare valore economico.

11.12 Roadmap reputazionale (indicativa)

- v0 — Policy minime, HCS scalare, badge base;
- v1 — HCS vettoriale, API pubbliche, attestazioni firmate;
- v2 — Integrazioni esterne (browser, repo multipli), quorum avanzati, modelli anti-gaming evoluti;
- v3 — Portabilità inter-ecosistemi (badge interoperabili), federation policy.

11.13 Esempio d'uso (end-to-end)

1. Eva propone una patch di sicurezza con test → *CI ok*, due maintainer approvano.
2. HRL registra l'evento, applica policy v1.2: +40 (fix) +10 (test) +16 (quorum 2 senior) ⇒ +66 su *HCS_security* e *HCS_totale*.
3. Il ledger reputazionale aggiorna i badge; *HCS_security* di Eva supera 300 → diventa Security Reviewer.
4. La sua nuova qualifica consente di esprimere voto tecnico su RFC di sicurezza.

12. Hash / Notarizzazione di HOLogram

L'integrità e la paternità del Whitepaper HOLogram non dipendono da un'autorità centrale, ma da un insieme di prove crittografiche distribuite che garantiscono che il documento pubblicato sia autentico, integro e attribuibile allo pseudonimo dell'autore al momento della pubblicazione.

Questo capitolo definisce le procedure consigliate per garantire una notarizzazione distribuita e resistente alla manomissione.

12.1 Obiettivo della notarizzazione

La notarizzazione serve a:

- dimostrare la data di pubblicazione del Whitepaper;
- garantire l'immutabilità del contenuto;
- consentire alla comunità di verificare l'autenticità del documento;
- preservare la paternità del protocollo in modo decentralizzato;
- permettere un'evoluzione del Whitepaper mantenendo la storia verificabile.

Non serve una registrazione tradizionale: la prova deve essere aperta, pubblica e verificabile da chiunque.

12.2 Generazione dell'hash (SHA-256)

Ogni versione del Whitepaper deve essere distribuita insieme al suo hash crittografico:

SHA-256(HOLogram.pdf) = <hash>

L'hash:

- identifica in modo univoco la versione del documento;
- permette di verificare che non sia stato modificato;
- costituisce la base della notarizzazione distribuita.

Procedura consigliata

1. Creare la versione finale del PDF.
2. Generare l'hash tramite tool standard:

   sha256sum HOLogram.pdf
3. Pubblicare l'hash insieme al documento.

12.3 Pubblicazione dell'hash in repository pubblici

Per garantire massima trasparenza, l'hash deve essere pubblicato in più luoghi:

- Repository GitHub ufficiale (commit immutabile);
- File "HASHES.txt" contenente tutte le versioni precedenti;
- Read-only gist come copia ridondante;
- README con riferimento all'hash della versione attuale.

Ogni commit del repository funge da attestazione temporale *non alterabile*.

12.4 Pubblicazione su IPFS (InterPlanetary File System)

Per garantire l'immutabilità distribuita del documento, è consigliato pubblicarlo su IPFS.

L'upload restituisce un CID (Content Identifier):

ipfs://<CID>

Il CID incorpora implicitamente l'hash contenuto del file.

Se il PDF cambia, cambia anche il CID → quindi i contenuti non possono essere manipolati.

12.5 Timestamping distribuito (opzionale ma consigliato)

Per una certificazione temporale più robusta, è possibile utilizzare servizi di timestamping decentralizzato.

In questa sezione *non nominiamo piattaforme specifiche*: il protocollo indica solo la logica.

Il meccanismo:

- prende l'hash SHA-256;
- lo include in un registro distribuito con timestamp verificabile;
- permette a chiunque di dimostrare che l'hash era pubblico a partire da quella data.

Questo passo è opzionale, ma utile per consolidare ulteriormente la prova di paternità.

12.6 Versionamento e archivio storico

Ogni evoluzione del Whitepaper deve essere:

- numerata (v1.0, v1.1, v2.0, ...);
- accompagnata da un nuovo hash;
- aggiunta al file HASHES.txt;
- conservata nel repository come tag immutabile;

- pubblicata su IPFS per preservarne lo storico.

Questo crea una catena verificabile di versioni:

v1.0 → hash1 → CID1

v1.1 → hash2 → CID2

v2.0 → hash3 → CID3

La comunità può così verificare l'evoluzione del documento anche a distanza di anni.

12.7 Paternità sotto pseudonimo

Lo pseudonimo BaffiSan è attestato tramite:

- l'hash del documento;
- l'associazione pubblica dell'hash a GitHub/ArXiv/IPFS;
- la firma digitale del commit (se utilizzata);
- la persistenza delle pubblicazioni nel tempo.

La paternità non dipende da identità anagrafiche, ma dalla continuità crittografica:

chi controlla lo pseudonimo è colui che può pubblicare versioni firmate dell'hash.

Come per i protocolli open più rilevanti, questo garantisce:

- anonimato dell'autore;
- verificabilità della sua continuità;
- protezione contro appropriazioni indebite;
- indipendenza dall'identità reale.

12.8 Verifica da parte della community

Chiunque può verificare un documento HOLogram:

1. scarica il PDF da qualsiasi fonte;
2. calcola l'hash locale;
3. confronta l'hash con quello pubblicato nel repository/IPFS;
4. verifica la presenza dell'hash nella cronologia dei commit;
5. verifica che l'hash corrispondente appaia nei timestamp pubblici.

Se tutti i passaggi coincidono, il documento è autentico.

12.9 Sintesi

La notarizzazione distribuita di HOLogram:

- non richiede autorità centrali;
- non necessita di registrazioni ufficiali;
- si basa esclusivamente su hash, pubblicazione pubblica e copie distribuite;
- protegge la paternità del documento;
- preserva l'integrità del protocollo nel tempo;
- garantisce trasparenza e verificabilità aperta.

È un sistema semplice, verificabile e coerente con la filosofia del protocollo:

proteggere senza controllare, attestare senza identificare.

✔ APPENDICE TECNICA — HOLogram

Questa Appendice raccoglie definizioni operative, pseudocodice concettuale, schemi dei moduli e un glossario minimo del protocollo HOLogram. Non costituisce un'implementazione, ma un riferimento tecnico che permette a sviluppatori, ricercatori e integratori di comprendere il funzionamento del Layer e di verificarne la coerenza nelle diverse varianti.

A.1 Struttura dei Moduli del Layer

Il protocollo HOLogram è composto da sei moduli principali, descritti qui in forma più operativa:

```
+-------------------------------------------------+
|            HOLogram Layer                       |
+-------------------------------------------------+
| 1. Input Capture (Vault)                        |
| 2. Signal Normalizer                            |
| 3. Persona Mixer                                |
| 4. DP Engine (Diff. Privacy)                    |
| 5. Privacy Budget Controller                    |
| 6. Output Projector                             |
+-------------------------------------------------+
```

Ogni implementazione deve rispettare la funzione semantica dei moduli, non necessariamente la loro forma o il loro ordine interno (eccetto Vault → DP Engine → Output, che è obbligatorio).

A.2 Esempio di Ciclo di Elaborazione (pseudocodice)

Il seguente pseudocodice illustra il flusso logico generale:

```
function HOLogramProcess(event):

raw = Capture(event)

shaped = Normalize(raw)

personaAdjusted = ApplyPersona(shaped, activePersona)

dpProtected = ApplyDP(personaAdjusted, dpParameters)

budgetState = UpdateBudget(dpProtected)

if budgetState.thresholdReached:

dpProtected = IncreaseProtection(dpProtected)
```

Emit(dpProtected)

Questo pseudocodice non è vincolante, ma chiarisce i passaggi formali richiesti dal protocollo.

A.3 Input Vault — Specifiche Minime

A.3.1 Eventi minimi da intercettare

- Tastiera: keydown, keyup, keypress
- Puntatore: mousemove, pointermove, mousedown, mouseup
- Scroll: wheel, scroll
- Interazioni UI: focus, blur, input

A.3.2 Requisiti semantici

- Nessun evento nativo ad alta risoluzione deve raggiungere la pagina.
- Il contenuto dell'evento (testo, coordinate, stato) deve essere preservato.
- I tempi e le microvariazioni devono essere rimossi o sostituiti.

A.4 Normalizzazione — Modello Operativo

A.4.1 Downsampling

Riduzione della frequenza degli eventi:

targetFrequency = 60 Hz

accumulator += 1

if accumulator % (nativeFreq / targetFrequency) == 0:

  emit event

A.4.2 Quantizzazione

Riduzione della precisione delle coordinate:

quantizedX = round(raw.x / qStep) * qStep

quantizedY = round(raw.y / qStep) * qStep

A.4.3 Smoothing

Applicazione di una funzione di livellamento:

smoothed = prev + alpha * (quantized - prev)

A.5 Persona Mixer — Parametri Sintetici

Una "persona" è definita da un insieme di variabili:

Persona:

pointerJitterRange

scrollPauseProbability

typingLatencyOffset

smoothingAggressiveness

curvatureSoftening

L'implementazione deve:

- generare una persona diversa per ogni sessione;
- mantenerla coerente nel tempo della sessione;
- evitare valori estremi o non plausibili.

A.6 Differential Privacy Engine — Linee Guida

HOLogram non applica la privacy differenziale classica su dataset statici:
il DP Engine adotta principi ispirati alla DP per flussi di eventi (stream), impiegando rumore minimo calibrato, riduzione di sensibilità e stabilizzazione per limitare l'emergere di micro-pattern identificanti senza degradare la UX.

A.6.1 Rumore calibrato (bozza)

dpEvent.value = event.value + jitter_minimo_calibrato(ε)

con ε selezionato entro range umani e policy variabili per sessione/sito, seguito da una stabilizzazione dolce per evitare segnali "meccanici".

A.6.2 Stabilizzazione

dpEvent = Smooth(dpEvent, stabilityFactor)

A.7 Behavioral Privacy Budget — Logica Operativa

A.7.1 Costruzione del Budget

Ogni evento ha un "peso comportamentale":

typeWeight:

keypress: 3

mousemove: 1

scroll: 2

click: 1

Il budget totale viene consumato così:

budget -= typeWeight[event.type] * intensity(event)

A.7.2 Escalation automatica

Quando il budget scende sotto una soglia:

if budget < threshold_low:

increase smoothing

increase quantization

reduce frequency

A.7.3 Rigenerazione del Budget

Il *Behavioral Privacy Budget* non è un valore statico: segue un modello di esaurimento controllato e rigenerazione modulare.

La rigenerazione avviene secondo tre modalità, definite per rendere il comportamento del Layer prevedibile ma non triviale da sfruttare:

1. Rigenerazione a soglia temporale (timer-based):

   Dopo un intervallo di tempo continuo (es. 60–120 secondi), una piccola quota di budget viene ripristinata.

   Questo previene il consumo totale nei siti che generano input naturali continui (es. lettura lunga con scroll moderati).

2. Rigenerazione a eventi "neutri":

   Eventi a bassa intensità (hover senza click, pause prolungate, idle) possono contribuire alla rigenerazione, riflettendo il fatto che l'utente *non sta esponendo nuova informazione comportamentale significativa*.

3. Rigenerazione completa al reset di sessione:

   Una nuova sessione utente (chiusura e riapertura del tab, cambio dominio o tempo di inattività esteso) ripristina il budget al massimo.

Il modello è parametricamente definito ed evita che il Layer rimanga in "protezione massima" per troppo tempo, preservando naturalezza e fluidità.

A.7.4 Calcolo dell'intensità degli eventi (Event Intensity Model)

La diminuzione del budget è proporzionale alla quantità di informazione comportamentale potenzialmente rivelata.

Ogni evento ha un costo calcolato come:

costo = w_type × f_freq × g_geom × h_entropy

Dove:

- w_type = peso base dell'evento

  - keypress = 3
  - mousemove = 1
  - scroll = 2
  - click = 1

- f_freq = fattore di frequenza

  Aumenta il costo se l'evento si ripete troppo rapidamente (segnale di alta granularità).

- g_geom = fattore geometrico

  Basato su:

  - ampiezza del movimento;
  - variazione angolare;
  - complessità della traiettoria;
  - deviazioni significative → maggiore esposizione.

- h_entropy = fattore di entropia comportamentale

  Aumenta se l'evento contiene micro-variazioni complesse e difficili da replicare, perché queste sono molto informative per un classificatore.

Esempio informale:

→ un singolo click ha costo minimo;

→ 30 mousemove complessi, con micro-curvature irregolari, consumano molto di più;

→ un pattern di digitazione intensa consuma rapidamente un blocco di budget.

A.7.5 Escalation e Decrescita graduale dell'Esposizione

Quando il budget scende sotto soglie predefinite:

- Soglia 1 (attenzione)

  aumento leggero dello smoothing, riduzione moderata della frequenza degli eventi.

- Soglia 2 (protezione attiva)

  aumento significativo di quantizzazione e smoothing.

- Soglia 3 (protezione massima)

  il Layer riduce drasticamente esposizioni ad alta frequenza e applica varianti più forti della persona sintetica.

Quando il budget viene rigenerato, l'offuscamento diminuisce gradualmente per garantire transizioni fluide.

A.8 Output Projector — Garanzia di Coerenza

L'Output Projector deve garantire:

- compatibilità con DOM standard,
- continuità dell'esperienza utente,
- assenza di artefatti non umani,
- timing entro range naturali.

A.9 Glossario Tecnico

Biometria comportamentale:

Insieme delle caratteristiche involontarie che rendono riconoscibile il comportamento umano nell'interazione digitale.

Offuscamento:

Trasformazione dei segnali con perdita controllata di granularità.

Persona sintetica:

Profilo comportamentale plausibile generato dal Mixer per sostituire la firma reale.

DP Engine:

Motore che minimizza la sensibilità dell'output rispetto a microvariazioni dell'input.

Budget comportamentale:

Valore numerico che rappresenta la quantità rimanente di informazione comportamentale "sicura" da esporre.

A.10 Schema Riassuntivo del Flusso

Utente
↓
Input Vault
↓
Normalizer
↓
Persona Mixer
↓
DP Engine
↓
Privacy Budget
↓
Output Projector → Browser/Sito