HASHFLIX

A SETTLEMENT PROTOCOL FOR PAY-PER-VIEW CONTENT MARKETS

Non-custodial campaign escrow on Solana, view measurement committed on-chain at the moment of observation, and payouts computed by a published schema that anyone can recompute and anyone can challenge.

REVISION	ISSUED	STATUS	AUTHOR
DRAFT V0.9	AUGUST 2026	IN DEVELOPMENT	SU.ENGINEERING

This document specifies a protocol under development. No programs are deployed, no campaign has settled, and the figures in the worked examples are illustrative. The protocol has no token, and nothing in this document is a commitment to issue one.

ABSTRACT

Pay-per-view creator marketing has become a functioning market: brands deposit budgets, open pools of creators publish short-form content, and payouts are computed from platform view counts at negotiated CPM rates. Every existing implementation of this market is operated by a single custodial intermediary that holds the funds, reads the measurements, applies undisclosed acceptance rules, and reports an unauditable total. We present Hashflix, a protocol that restructures this market around three mechanisms: (i) non-custodial campaign escrow on Solana, under which no operator can access, redirect, or withhold funds; (ii) committed measurement, under which every view-count reading taken from a platform API is signed and anchored on-chain at observation time, making the measurement record append-only and publicly replayable; and (iii) reproducible settlement, under which payouts are computed by a versioned, published scoring function over the committed record, disputable through a bonded challenge game and backstopped by bonded, attested creator identities. We give the protocol specification, a settlement function designed to inherit the content platforms' own fraud remediation, an economic analysis deriving stake requirements under which view fraud has negative expected value, and a threat analysis. We deliberately do not claim trustless verification of view authenticity: we show that no protocol can make that claim, because the trust bottleneck is the data source rather than the data pipeline, and we define precisely which trust assumptions Hashflix removes, which it retains, and what each retained assumption costs an adversary to abuse.

01

INTRODUCTION

Advertising infrastructure and creator-economy infrastructure solved opposite halves of the same problem. Advertising built measurement: third-party ad servers, viewability standards, accredited measurement vendors, and audit trails, all layered over a payment system that still runs on invoices and net-terms trust. The creator economy built payments in the opposite direction: platforms such as Whop's Content Rewards, ClipAffiliates, and Noise move money from brands to thousands of small creators at per-view rates, but perform measurement, fraud filtering, and settlement inside a single company with no external audit surface.

The result is a market with real and growing volume settling entirely on operator trust. Whop reports over 100 million clipping plays per day across its system and has tracked several million dollars in creator payouts [1]. ClipAffiliates, one of the few operators to publish fraud data, reports that roughly one in three views ever submitted to its platform originated from accounts subsequently banned for view manipulation [2]. Brands in this market cannot verify what they were charged for; creators cannot contest rejections with evidence, because the evidence is private; and both sides extend unsecured credit to the operator between deposit and payout.

This paper specifies Hashflix, a settlement protocol for pay-per-view content markets. The protocol makes a deliberately bounded claim. It does not verify that views are authentic, and Section 3.3 argues no protocol can. It instead guarantees four properties:

1. **Fund safety.** Campaign budgets are held by an on-chain program. The operator has no withdrawal path. (Section 4.2)
2. **Measurement integrity over time.** Every reading is committed at observation time. The record is append-only; retroactive revision is publicly detectable. (Section 4.4)
3. **Settlement reproducibility.** The function from committed measurements to payouts is published and deterministic. Any party can recompute any payout. (Sections 4.5, 4.6)
4. **Accountable adjudication.** Payouts are disputable through a bonded challenge game with objective resolution, and adverse automated decisions carry a binding human-review path. (Section 4.7)

We further specify an identity and staking layer (Section 4.8) under which the protocol's fraud economics hold against sybil strategies, an economic analysis deriving the stake sizes that make fraud unprofitable (Section 5), a threat analysis (Section 6), and the compliance posture that constrains several design choices (Section 7).

Contributions. To our knowledge, Hashflix is the first specified protocol for pay-per-view content settlement with non-custodial funds; the first to formalize settlement on platform-audited view counts rather than instantaneous readings (the surviving-count rule, Section 4.5); and the first in its category to state a complete trust model, including the assumptions it cannot remove.

02

BACKGROUND AND RELATED WORK

2.1 Pay-per-view content markets

The market Hashflix targets emerged commercially between 2024 and 2026. Operators run one of two models. In the *marketplace* model (Whop Content Rewards [1], ClipAffiliates [2], Contentrewards, clipping.net), brands post campaigns with a CPM rate and budget, and an open pool of creators submits short-form posts on TikTok, Instagram Reels, and YouTube Shorts; the operator reads view counts through platform APIs under creator-granted authorization and pays out per thousand views after an internal review period, typically 72 hours. In the *managed* model (Noise [3], and agency operators in the crypto vertical), the operator additionally controls content templates and creator rosters. Published rates range from $0.20 to $6 per thousand views for general campaigns, with crypto and finance campaigns paying $4 to $9 per thousand [4], a premium driven by paid-channel advertising restrictions on those verticals across major platforms.

In every operating implementation, the operator custodies brand deposits, applies non-public fraud filters, and computes payouts privately. Fees range from 5.9% to 9%, charged variously to the brand or deducted from creator payouts.

2.2 Attention-market protocols

A parallel crypto-native category, sometimes labeled InfoFi, rewards creators from project budgets according to platform-scored *mindshare* rather than per-view delivery: Kaito distributes rewards by AI-scored share of voice on X [5]; Cookie DAO operated a comparable product until X's enforcement action against engagement farming ended it [6]. These systems demonstrate demand for programmatic creator incentives but differ from Hashflix in unit of account (mindshare score versus delivered views), scoring transparency (proprietary models versus a published schema), and settlement (allocations at operator discretion versus program-executed payouts). The Cookie DAO shutdown is treated in Section 6 as a live instance of platform-policy risk.

2.3 Oracles for web data

Bringing web-served facts on-chain has two research lineages. Trusted-hardware and notary designs (Town Crier [7], TLSNotary [8]) interpose an attestor in a TLS session; proof-based designs (DECO [9], and deployed systems such as Reclaim Protocol [10]) allow a client to prove statements about TLS responses with selective disclosure. These systems prove *provenance*: that a server said X at time t. They cannot strengthen the *accuracy* of X. Section 3.3 builds on this distinction, which is decisive for view counts: a proof over an inflated count is a valid proof of an inflated count. Hashflix therefore uses provenance machinery (signed, committed readings; optionally TLS proofs, Section 10) while locating fraud resistance in economics and in the settlement function rather than in the transport.

2.4 On-chain attestation and identity

The Ethereum Attestation Service established attestation registries as a composable primitive [11]; the Solana Attestation Service (SAS) [12] provides the equivalent on Hashflix's settlement chain, with commercial identity issuers (including Sumsub [13]) writing reusable KYC attestations bound to self-custodied wallets. Hashflix consumes SAS credentials as a verifier and is never an issuer. Sybil resistance through identity is bounded: document-based verification deduplicates within an issuer network but does not guarantee one-person-one-credential [14], and proof-of-personhood systems make stronger uniqueness claims at the cost of specialized enrollment [15]. Section 5.1 sizes stakes under the weaker, realistic assumption.

2.5 Optimistic dispute mechanisms

Hashflix's challenge game follows the optimistic pattern of TrueBit [16], optimistic rollup fraud proofs, and UMA's optimistic oracle [17]: results stand unless challenged within a window; challenges are bonded; the party found wrong funds the party found right. Hashflix's instantiation is simpler than general computation disputes because resolution is objective: a challenged payout is recomputed from the committed measurement record and the revealed scoring parameters in force at decision time, so the dispute reduces to deterministic re-execution over public inputs.

03

SYSTEM MODEL

3.1 Actors

- **Brand** *B*: funds a campaign; sets rate, caps, rules; may strike non-compliant submissions.
- **Creator** *C*: publishes posts, grants read authorization for their own posts' statistics, submits post references, posts a stake, receives payouts.
- **Operator / Attester** *A*: the entity holding platform developer credentials; runs measurement polling, signs readings, commits roots, executes scoring, operates the human-review desk. At v1 this is su.engineering (Section 8).
- **Escrow program** *E*: the on-chain program holding campaign funds and stakes, executing settlement and dispute outcomes.
- **Identity issuer** *I*: a third-party verification provider writing SAS attestations; independent of *A*.
- **Challenger** *X*: any party disputing a scored result by posting a bond.
- **Platform** *P*: the content platform (TikTok, YouTube, Instagram) serving view counts through its developer API. Not a protocol participant; the protocol's principal external dependency.

3.2 Trust assumptions

We state the trust model as obligations the protocol does and does not discharge.

Removed.

- *B* and *C* need not trust *A* with funds: *E* holds all balances; *A* has no withdrawal instruction. (T1)
- *B* and *C* need not trust *A*'s history: readings are committed at observation time; the record is append-only. (T2)
- *B* and *C* need not trust *A*'s arithmetic: settlement is a published deterministic function of committed inputs. (T3)

- *A* need not be trusted for identity issuance: *I* issues; *A* verifies. (T4)

Retained.

- All parties trust *P*'s view counts as the measurement source, net of the mitigation in Section 4.5. (R1)
- All parties trust *A* to read *P* honestly at each observation instant. This is bounded by T2 and T3 (a dishonest reading is permanently committed and disputable) and by the challenge game's economics, but a reading that is wrong at commitment time and consistent with *P*'s subsequent behavior may go undetected. (R2)
- Scoring threshold values are trusted blind during their period of force and verified in arrears through commit-and-reveal. (R3)
- Contested adverse decisions terminate in human judgment. (R4)

3.3 Why view authenticity cannot be verified, by anyone

A view count is a claim by *P* about events (impressions) that only *P* observed, filtered by fraud systems only *P* operates. Every party outside *P*, including any protocol, receives the output of that pipeline. Cryptographic transport (Section 2.3) can prove the output's provenance and integrity in transit; it has no purchase on the pipeline's accuracy. It follows that:

1. A "trustless view verification" protocol is impossible under any proof system, because the object to be verified is defined by a trusted party's internal computation.
2. The strongest achievable properties are provenance (what *P* said), permanence (what was said cannot be unsaid quietly), reproducibility (what follows from what was said is checkable), and economic deterrence (what it costs to game *P* exceeds what gaming pays).

Hashflix is a construction of exactly these four properties. The settlement function (Section 4.5) additionally *co-opts P's own fraud remediation*, which operates with information and budget unavailable to any external party, as the protocol's first fraud filter.

3.4 Design constraints

Two constraints shape the protocol against more decentralized alternatives.

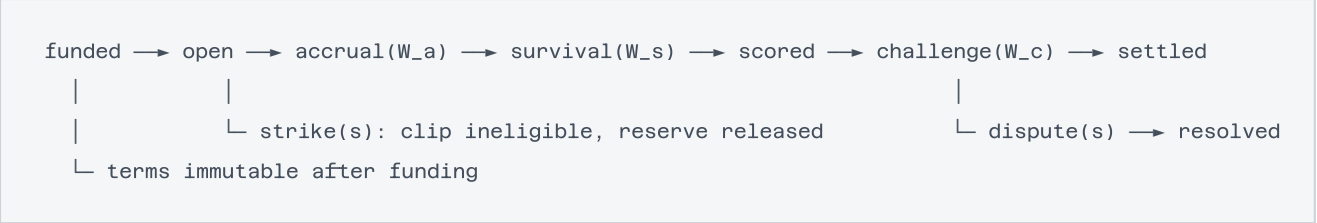
Consented API access only. Platform terms prohibit automated collection of user data outside authorized developer access [18]. A design in which independent nodes fetch public post pages is more trust-minimized on paper and violates the terms of the platforms the protocol depends on; it is also operationally fragile against anti-scraping systems. Hashflix reads statistics exclusively through platform developer APIs under authorization each creator grants for their own posts. Consequence: measurement is necessarily performed by holders of registered developer credentials, i.e., named legal entities. R2 is therefore irreducible at the protocol layer; Section 8 discusses widening the attester set contractually.

Non-custodial by construction. Under MiCA, custody and administration of crypto-assets on behalf of clients, and their transfer, are licensed activities [19]. *E* is designed so that no natural or legal person, including *A*, controls client funds: deposits bind irrevocably to campaign terms, payouts execute against attested inputs, and residuals return to *B* by program logic. Section 7 details the perimeter.

PROTOCOL SPECIFICATION

Notation. A campaign is $K = (B, r, m, cap, R, W_a, W_s, T_0)$ with rate r in USDC per thousand views, budget m, per-clip cap cap, rules document hash R, accrual window W_a, survival window W_s, and funding time T_0. A submission is $s = (C, P, id\_post, t_s, h_p)$ with perceptual hash h_p. A reading is $\rho = (id\_post, P, v, e, t)$ with view count v, engagement vector e (likes, comments, shares, saves where exposed), and timestamp t.

4.1 Campaign lifecycle



1. **Funding.** B transfers m USDC to a campaign account of E; terms K are recorded and immutable. The protocol fee $\phi \cdot m$ (Section 5.3) transfers to the treasury at funding; the reward pool is $(1-\phi) \cdot m$.
2. **Open.** Creators submit posts. Submissions failing deduplication (4.3) or eligibility are rejected at entry. B may strike any submission for violation of R during open and accrual phases; strikes are recorded on-chain with a reason code, release the clip's budget reservation, and are disputable like any adverse decision (4.7).
3. **Accrual.** For clip c, views accrue during $[t_s(c), t_s(c)+W_a]$. Readings are taken and committed throughout (4.4).
4. **Survival.** Readings continue during $[t_s(c)+W_a, t_s(c)+W_a+W_s]$. No new views count; the phase exists to observe platform corrections. A post must remain live and readable through survival; a post that becomes unreadable settles at zero.
5. **Scoring and challenge.** A executes the scoring function (4.6) over the committed record and posts per-clip results; a challenge window W_c opens (4.7).
6. **Settlement.** E pays unchallenged and dispute-resolved amounts to creator wallets and returns the pool residual to B.

4.2 Escrow program

E is a Solana program. Per campaign it derives a program-controlled account holding the USDC reward pool, and per creator a stake account (4.8). Its instruction set, abbreviated:

- `init_campaign(K)` funds and freezes terms; transfers fee.
- `submit(s, auth_proof)` registers a submission after deduplication and identity checks.
- `strike(id_s, reason)` (signer: B) marks ineligibility.
- `commit_root(epoch, root, sig_A)` anchors a Merkle root of the epoch's readings (4.4).
- `post_scores(campaign, results_root, sig_A)` opens the challenge window.
- `challenge(id_s, bond)` / `resolve(id_s, outcome, evidence)` run the dispute game (4.7).
- `settle(campaign)` executes payouts and residual return after W_c and open disputes close.
- `slash(C, evidence)` forfeits stake on resolved fraud findings.

A signs measurements and scores; it holds no instruction that moves pool funds to itself or to arbitrary parties. The settle path is constrained to (creator wallets registered at submission, *B*'s funding wallet).

4.3 Submission and deduplication

At submission the clip's media is fetched once and perceptually hashed [20]; h_p is checked against the global corpus of prior submissions across all campaigns. Matches within a distance threshold are rejected, removing resubmission and cross-campaign recycling at entry. h_p is stored; raw media is not.

4.4 Measurement and commitment

A polls *P*'s API for each active clip on a fixed schedule (launch default: every 6 hours) under the creator's OAuth grant. Readings are batched per epoch; each batch forms a Merkle tree over leaves $H(\rho_i)$; `commit_root` anchors the root with *A*'s signature. Full batches are published to a public mirror (repository plus permanent storage) so that any party can (i) verify inclusion of any reading against the on-chain root and (ii) replay scoring end-to-end. Two properties follow: the record is append-only (a revised reading cannot replace a committed one without a visible new commitment), and *A*'s claims are non-repudiable (every committed root is signed).

Platform APIs return point-in-time snapshots, not history [18]; the committed series *is* the history, which is why commitment at observation time, rather than publication at settlement time, is load-bearing.

4.5 Settlement function: the surviving count

For clip c with committed series $\{v(t)\}$, define the accrual value $v_a(c) = \max\{ v(t) : t \le t_s(c)+W_a \}$ and the settlement value

$$\hat{v}(c) = \min\{ v(t) : t_s(c)+W_a \le t \le t_s(c)+W_a+W_s \},$$

capped at $v_a(c)$. The gross payout before scoring is $g(c) = \min(r \cdot \hat{v}(c)/1000, \text{cap})$.

Rationale. Content platforms audit their own counts: corrections remove views their fraud systems attribute to manipulation [21, 22]. An organic count is non-decreasing, so its minimum over the survival phase equals its value at the phase's start and the rule is neutral for honest creators. A manipulated count is corrected downward with positive probability during survival, and the minimum captures the correction automatically. The rule therefore inherits, at zero marginal cost, the fraud detection of the only party with session-level data. The price is latency: $W_a + W_s$ (launch default 7 + 7 days) versus the 72-hour norm. Window lengths are protocol parameters to be recalibrated per platform from observed correction latencies (Section 9).

4.6 Scoring

Scoring maps the committed series and account context to a multiplier $\sigma(c) \in \{0, 1\}$ (binary at v1; graduated multipliers are future work) via a published, versioned schema S_v with committed parameters θ_v. Feature classes:

- **Curve shape.** Deviation of $\{v(t)\}$ from organic short-form accrual profiles (front-loaded growth, smooth decay); step discontinuities and post-decay linear segments flag purchased views.
- **Engagement consistency.** Ratios of likes, comments, shares, saves to views against per-vertical, per-view-bracket bands; comment length and repetition statistics.
- **Cost-asymmetric signals.** Shares and saves per view are weighted above raw views: their unit forgery cost is materially higher.

- **Account priors.** Account age, posting cadence, and the share of the creator's prior clips that settled cleanly.
- **Cross-account structure.** Correlation of posting times, curve shapes, and commenter sets across submissions within and across campaigns.

Transparency with lagged parameters. S_v (features, functional form, combination rule) is public. The parameter vector θ_v is committed as $H(\theta_v \| salt)$ on-chain when brought into force and revealed when retired, so any historical decision is verifiable against the parameters then in force, while live parameters cannot be tuned against. The schema's weight toward cost-asymmetric signals is the deliberate hedge against the residual risk of parameter leakage: the defense is priced in forgery cost, not only in secrecy.

Adverse decisions. $\sigma(c)=0$ and strikes are adverse automated decisions with real effect on C. Each carries the dispute path of 4.7 and, independently, a binding human review on request, recorded with the reviewer's rationale (Section 7).

4.7 Disputes

Any party may challenge a posted result (a payout, a zero-score, or a strike) within W_c by bonding b_x. Resolution is objective: the disputed clip is re-scored by deterministic re-execution of $S_v(\theta_v)$ over the committed record; for strikes, review is against the campaign's hashed rules R. If the challenge is upheld, the challenger receives the loser's bond share and the corrected result executes; if rejected, the challenger's bond funds the prevailing side. Because inputs are public and the function is deterministic, disputes cannot devolve into oracle games; the human-review path (R4) exists for rule-interpretation cases (strikes) rather than arithmetic ones. Section 5.2 sizes b_x against griefing.

4.8 Identity and staking

Identity. Before first cumulative withdrawal above threshold τ, C completes verification with issuer I, which writes an SAS attestation binding a verified-identity credential to C's payout wallet [12, 13]. E gates withdrawals above τ on a live attestation. A consumes the attestation and never receives underlying documents. The credential also discharges AML screening obligations attached to payouts (Section 7). The uniqueness guarantee is deduplication within I's network, not proof of personhood; the economic analysis assumes an attacker can obtain k small distinct credentials (Section 5.1).

Stake. C bonds $\beta(tier)$ USDC scaled to their reputation tier's per-clip cap. Earnings vest linearly across the survival window rather than releasing at scoring. Resolved fraud findings slash the bond and unvested earnings. Reputation (cleanly settled volume, account age in protocol, challenge history) is bound to the attested identity, is non-transferable, and gates access to higher caps and higher-rate campaigns.

05

ECONOMIC ANALYSIS

5.1 Fraud expected value and bond sizing

Consider an attacker submitting a manipulated clip targeting gross payout $G \le cap$. Let p be the probability the fraud is caught by any layer (platform correction during survival, schema, challenge) before vesting completes, β the bond at the attacker's tier, and c_f the attacker's direct cost of manufacturing the fraudulent views. The attempt's expected value is

$$\mathbf{EV = (1-p)\cdot G - p\cdot(\beta + \lambda G) - c_f},$$

with λ the vested-forfeiture fraction ($\lambda \to 1$ as detection latency shortens relative to vesting). Requiring EV < 0 for all $G \le cap$ under conservative $\lambda = 0$ gives the bond rule

$\beta > cap \cdot (1-p)/p$.

At $p = 0.25$, $\beta = 3 \cdot cap$; at $p = 0.4$, $\beta = 1.5 \cdot cap$. The published one-in-three fraud share of an operator running only private filters [2] suggests p well above 0.25 is achievable once platform corrections (4.5) and adversarial challenges (4.7) stack on schema detection. Identity multiplicity: an attacker holding k credentials runs k independent instances of the same negative-EV game while paying k verification and bonding costs; multiplicity scales exposure, not profitability. What multiplicity does enable is distributing volume below per-identity visibility thresholds, which is why cross-account structural features (4.6) are part of the schema rather than an afterthought, and why caps bind per identity per campaign.

Purchased-view market prices place c_f low in absolute terms but not against these bonds: the binding constraint on the attacker is β at risk, not media cost.

5.2 Challenge economics

A challenge costs b_x and returns a share $\gamma \cdot \beta$ (against creators) or a fixed bounty from the treasury (against A's scoring, funded from the fee). Honest challenges are profitable whenever the challenger's recomputation shows a deviation, since resolution is deterministic re-execution: the challenger's evidentiary cost is compute, not persuasion. Griefing (challenging correct results to delay settlement) costs the griefer b_x per attempt with zero success probability against deterministic resolution, and delays only the challenged clip, not the campaign. b_x is set above resolution gas and processing cost and below the minimum economically meaningful payout deviation, so that every true deviation is worth challenging and no false challenge is.

5.3 Fee model

The protocol charges $\phi = 5\%$ of the brand deposit at funding, and nothing on payouts. Comparables: 7% brand-side (ContentRewards), approximately 6% all-in (Whop general stack), 9% deducted from creator payouts (ClipAffiliates) [1, 2]. Charging the funding side keeps the payout leg whole, which is both the creator-aligned position and the operationally simpler one: the settlement instruction pays exactly the scored amount. Fee revenue funds operations, the review desk, the challenge bounty pool, and open-source development. The protocol has no token, and this document is not a commitment to issue one.

06

THREAT ANALYSIS

We enumerate adversaries against the protocol's guarantees; each row names the binding countermeasure, not every applicable one.

ADVERSARY AND STRATEGY	GUARANTEE ATTACKED	BINDING COUNTERMEASURE
Creator buys views (panel or bot traffic)	Payout correctness	Surviving-count rule inherits platform correction (4.5); curve and engagement features (4.6); bond makes residual EV negative (5.1)

ADVERSARY AND STRATEGY	GUARANTEE ATTACKED	BINDING COUNTERMEASURE
Creator rents aged accounts with organic history, bots the clip	Payout correctness	Hardest single case. Survival window plus cost-asymmetric signals; per-identity caps; slashing on resolution. Detection probability here bounds honest *p* estimates and hence bond sizing
Coordinated ring spreads volume across many small identities	Sybil resistance	Verification and bond cost per identity (5.1); cross-account structural features (4.6); per-identity caps
Clip recycling across campaigns	Budget integrity	Perceptual-hash deduplication at entry (4.3)
Creator deletes or privates post after scoring pressure	Measurement continuity	Unreadable-in-survival settles at zero (4.1)
Engagement-pod farming (real humans, coordinated)	Payout correctness	Commenter-set overlap and timing correlation features; cost asymmetry of sustained coordinated saves/shares
Wash conversions (future on-chain settlement tier)	Attribution integrity	Funding-graph exclusion, holding periods, per-wallet caps (Section 10)
Operator misreports a reading	Measurement integrity (R2)	Signed commitment at observation time (non-repudiation); any party holding a platform-consistent series can challenge; operator cannot profit via funds (T1)
Operator revises history	T2	Append-only committed record; revision requires a new visible commitment
Operator scores dishonestly	T3	Deterministic public schema; challenge with treasury bounty (5.2)
Operator absconds	T1	No withdrawal path exists to abscond with
Brand strikes clips to avoid paying	Creator protection	Strikes are on-chain, reasoned, disputable, human-reviewable (4.6, 4.7)
Fake brand and colluding creators drain incentives	Treasury	The protocol subsidizes nothing; a self-dealing campaign moves the brand's own money minus the fee
Platform policy reclassifies paid clipping	Protocol viability	Not mitigable at the protocol layer. Managed by compliance posture (disclosure as payout condition, consented APIs only) and by multi-platform scope; precedent: Cookie DAO's product shutdown after X policy enforcement [6]
Platform revokes API access	Measurement availability	Not mitigable without terms violations; multi-platform scope reduces concentration; open risk (Section 9)
Identity issuer failure or compromise	T4, sybil floor	Issuer plurality is supported by SAS's multi-issuer registry; attestations are revocable and re-issuable

Two entries deserve emphasis because they are the ones a diligent reader will probe. The aged-account attack is the strongest known strategy against any per-view market, including incumbents; Hashflix's position is not that it detects every instance but that survival, caps, and slashing bound its profitability, and that the committed record makes every undetected instance a permanent, re-examinable artifact rather than a closed case. The platform-policy risk is existential for the category, not specific to this protocol, and the protocol's design (consent-only measurement, enforced disclosure) is chosen to sit on the survivable side of any plausible enforcement line.

07

COMPLIANCE POSTURE

Compliance constraints are treated as protocol design inputs, not legal boilerplate, because three of them dictated architecture.

Asset regulation (MiCA). Custody or administration of crypto-assets for clients, and transfer services, are licensed CASP activities in the EEA [19]. *E* is constructed so that no party performs them: deposits bind to immutable terms, no operator withdrawal instruction exists, payouts execute programmatically to registered wallets, residuals return to the funder. The protocol layer carries no fiat rails; introducing them, pooling balances, or adding operator payout discretion would re-open the analysis, and is therefore excluded from the roadmap at the protocol layer.

Payout obligations. Payouts to verified identities above threshold carry AML screening and, where applicable, transfer-reporting duties. The identity layer (4.8) is positioned at the withdrawal threshold to discharge these once, through the issuing provider's screening, rather than per campaign.

Data protection (GDPR). The operator is controller for creator OAuth tokens, the measurement series, and payout metadata; platform-side analytics processing may additionally constitute joint controllership under platform terms [18]. Scoring and strikes are automated decisions with significant effect; Art. 22 requires a human-review path, which the protocol implements as a binding, recorded step (4.6, 4.7) rather than a support policy. Cross-account structural analysis is profiling and is disclosed as such in the protocol's transparency documentation.

Advertising disclosure. EU consumer-protection law attaches disclosure duties to any consideration, and enforcement against undisclosed influencer content is active and tightening across member states [23]. Hashflix enforces disclosure mechanically: eligibility requires the platform's own commercial-content designation on the post, verified at measurement time. A post that drops the designation becomes ineligible from that reading forward.

08

GOVERNANCE AND PROGRESSIVE DECENTRALIZATION

v1. su.engineering, as an incorporated entity, operates *A*: it holds platform developer agreements, the identity-vendor relationship, data controllership, and the review desk. The protocol's programs, schema, and tooling are open source; external contribution is open from launch. This concentration is disclosed rather than obscured: given Section 3.4, *some* named entity must hold these positions at any point in the protocol's life.

v2. With the open market (any brand may list; any attested creator may participate), governance and treasury transfer to a foundation: fee rate ϕ, stake schedule $\beta(tier)$, windows (W_a, W_s, W_c), challenge bond b_x, supported platforms, treasury allocation, grants, and final-tier dispute appeals. The foundation employs staff and contracts protocol maintenance to su.engineering under an arm's-length services agreement, with the stated objective of widening both the contributor set and, contractually, the attester set over time (multiple credentialed operators cross-committing readings is compatible with Section 3.4; a permissionless attester set is not).

The boundary, stated once. Platform developer agreements, the identity-vendor contract, data controllership, and Art. 22 review require a named, staffed, liable legal entity. A staffed foundation qualifies; a token vote does not. Ownership, parameters, treasury, and operator selection decentralize; assumption R2 is reassigned, audited, and

multiplied, never eliminated. The protocol publishes this boundary so that its decentralization claims remain checkable against it.

The protocol launches without a token. Governance design at v2 does not presuppose one.

09

LIMITATIONS

1. **R1 is irreducible.** Settlement quality is bounded by platform count quality; the surviving-count rule narrows, and cannot close, the gap between counted and genuine views.
2. **Single attester at v1.** R2 is held by one entity, mitigated by commitment, determinism, and challenges, but not removed. Multi-attester operation is contractual future work, not a launch property.
3. **Latency.** $W_a + W_s$ defers payment relative to the 72-hour market norm. The protocol trades speed for settled-count quality and says so; creators price the trade through the rates brands must offer.
4. **Window calibration is unmeasured.** The 7+7 day defaults encode an assumption about platform correction latency that must be replaced with measured distributions per platform before the parameters are considered final.
5. **Threshold secrecy is temporary by design and imperfect in principle.** Lagged reveal narrows adaptive tuning; it does not eliminate inference from decisions. The cost-asymmetric feature weighting is the standing hedge.
6. **Coverage.** Document-based identity excludes creators whose states do not interoperate with the issuer network, in tension with the payout layer's global reach. Issuer plurality mitigates; it does not resolve.
7. **API concentration.** Measurement availability is a platform decision. There is no compliant fallback, and the protocol declines to pretend otherwise.

10

FUTURE WORK

Settlement on on-chain outcomes. For advertisers whose conversion event is itself on-chain (swaps, mints, deposits), settlement can move from views to attributed actions: per-clip referral tagging carried into the transaction, funding-graph exclusion of self-dealing wallets, per-wallet caps, and net-of-unwind accounting. Unlike view counts, these events are publicly verifiable and expensive to forge, making them the one settlement basis in this market that admits a genuinely trust-minimized construction. This is the protocol's strongest reason to live on its settlement chain, and its most consequential planned extension.

TLS-proof measurement fallback. For creators declining OAuth, client-generated TLS proofs over their own analytics [9, 10] can substitute for API polling at reduced cadence, with the accuracy caveats of Section 3.3 unchanged.

Graduated scoring. Replacing binary σ with calibrated multipliers once post-launch data supports calibration.

Attester plurality. Contracted second attesters with independent credentials cross-committing readings, converting R2 from single-party to any-honest-of-n within the consent constraint.

Reputation portability. Standardized attestation of settlement history so creator reputation is legible to third parties beyond the protocol.

11

CONCLUSION

Pay-per-view content marketing reached real scale by rebuilding media buying without measurement infrastructure, and its participants currently pay for that absence in operator trust: unauditable measurement, undisclosed fraud filtering, and custodial settlement. Hashflix specifies the missing settlement layer under a deliberately honest trust model. Funds are safe by construction rather than by reputation; measurement is permanent and replayable rather than private; settlement is a published function rather than an assertion; and the one assumption no design in this category can remove, trust in the platform's own numbers, is stated, priced, and turned partially to the protocol's advantage by settling only on counts that survive the platform's own audits. We believe markets adopt verifiability when it is cheaper than trust; Hashflix prices it at 5%, below the cost of trust currently on offer.

REFERENCES

[1] Whop. Content Rewards: platform statistics and creator payout data. https://whop.com/content-rewards, accessed August 2026. [2] ClipAffiliates. Platform fraud-rate disclosure and campaign case data. https://clipaffiliates.com, accessed August 2026. [3] Noise Inc. Pay-per-view creator marketing for brands. https://getnoise.com, accessed August 2026. [4] Rate data aggregated from published campaign terms across clipping platforms and agencies, 2025–2026; su.engineering market research notes, available on request. [5] Kaito AI. Yaps and mindshare-based creator rewards. https://kaito.ai, accessed August 2026. [6] Cookie DAO. Snaps product discontinuation notice following X platform policy enforcement, 2026. [7] F. Zhang, E. Cecchetti, K. Croman, A. Juels, E. Shi. Town Crier: An Authenticated Data Feed for Smart Contracts. ACM CCS, 2016. [8] TLSNotary. Protocol documentation. https://tlsnotary.org. [9] F. Zhang, D. Maram, H. Malvai, S. Goldfeder, A. Juels. DECO: Liberating Web Data Using Decentralized Oracles for TLS. ACM CCS, 2020. [10] Reclaim Protocol. zkTLS provider documentation. https://reclaimprotocol.org. [11] Ethereum Attestation Service. https://attest.org. [12] Solana Foundation and Solana Identity Group. Solana Attestation Service. https://attest.solana.com, 2025. [13] Sumsub. Reusable digital identity with Solana Attestation Service integration, 2025. https://sumsub.com. [14] J. R. Douceur. The Sybil Attack. IPTPS, 2002. [15] Worldcoin. A New Identity and Financial Network (whitepaper), 2023. [16] J. Teutsch, C. Reitwießner. A Scalable Verification Solution for Blockchains (TrueBit), 2017. [17] UMA Project. Optimistic Oracle documentation. https://uma.xyz. [18] TikTok. Developer Terms of Service and Display API documentation. https://developers.tiktok.com. Analogous terms: YouTube Data API, Instagram Graph API. [19] Regulation (EU) 2023/1114 on markets in crypto-assets (MiCA), Titles IV–V. [20] C. Zauner. Implementation and Benchmarking of Perceptual Image Hash Functions. 2010. [21] YouTube Help. How video views are counted; view audits and count freezing. [22] TikTok. Community Guidelines: Integrity and Authenticity; removal of inauthentic engagement. [23] European Commission. Digital Fairness: consumer protection in influencer marketing; member-state enforcement sweeps 2024–2026.

Hashflix Protocol is developed in the open by su.engineering. Specification, programs, and schema at hashflix.com.