

terms.txt: A Consent and Compensation Protocol for Agentic Web Access

Rajarshi Chowdhury

Abstract—The open web ran on an unwritten bargain: sites admitted crawlers, and search engines sent visitors back. Public measurements show that bargain breaking under AI crawlers and agents. Automated clients now make up most requests, training dominates Cloudflare-classified crawling, and the largest AI platforms fetch thousands of pages for each visitor they return. The web’s common control, `robots.txt`, cannot express identity, purpose, terms, or price, can be circumvented, and newer alternatives are largely proprietary CDN features. We specify `terms.txt`, a `robots.txt`-style file for per-path, per-purpose machine-access terms, plus an origin-enforced exchange using Web Bot Auth signatures, signed intent, delegation tokens, HTTP 402 negotiation, and signed receipts. We define what the exchange can enforce, audit, and leave to contract. A dependency-free implementation adds 0.20 to 0.65 ms per request on one vCPU.

Index Terms—`terms.txt`, `robots.txt`, AI crawlers, web agents, Web Bot Auth, consent, content licensing, Internet economics.

ON September 15, 2026, Cloudflare will change the default terms a new website presents to machines. For every domain onboarded from that day, crawlers in its Training and Agent categories will be blocked on pages carrying advertising unless the owner opts out, and a multi-purpose crawler will be judged by its most restricted purpose [15]. With more than 20 percent of the web behind Cloudflare [1], one vendor’s default will set the opening terms of machine access for a large share of new origins.

That one private company can set these defaults, and that publishers welcomed them, shows how badly existing mechanisms have failed. Those mechanisms are the Robots Exclusion Protocol [8] and the unwritten bargain built around it: sites let crawlers read their pages, operators built indexes, and those indexes sent people back to see ads, click affiliate links, and buy subscriptions.

This article makes three contributions. First, it assembles public measurements from traffic operators and independent researchers to show that the bargain broke, and how quickly. Second, it shows that `robots.txt` cannot express what a site now needs to tell a machine client and is not reliably honored. Third, it specifies `terms.txt`, a `robots.txt`-style file of access terms, and an origin exchange that enforces them. The design composes existing standards work into one request-response flow, runs as origin middleware without a proxy, and measures the cost. It also separates what can

be enforced before delivery, audited afterward, and left to contract. The measurements motivate the work; the protocol is the contribution.

I. THE BARGAIN IN NUMBERS

The economics of the open web were never written into a protocol. They emerged from HTTP, which serves a page to whoever asks; `robots.txt`, which lets a site ask clients not to fetch certain paths; and search engines, which turned crawled pages into referrals. Publishers tolerated crawlers because traffic came back.

Three measurement programs show whether it still does, but they count different things. Imperva, analyzing all requests across its customer base, including API calls, put automated clients at 51 percent of web traffic in 2024 and above 53 percent in 2025 [2]. Cloudflare, counting only HTML page requests behind its network, reported 57.5 percent automated in June 2026 [1]. [3], The figures are not directly comparable, and neither isolates agentic AI traffic. They do point in the same direction.

The metric that captures what comes back is Cloudflare Radar’s crawl-to-refer ratio, published since July 2025: HTML pages a platform’s crawlers request for each HTML page visit it refers to a site [4]. Conventional search is near five to one [3]. From June 19 to 26, 2025, Anthropic’s crawlers requested roughly 70,900 pages per referred visit [4]. Cloudflare later reported Anthropic at 286,000 in January 2025 and 38,000 in July, an 87 percent decline that still left it the most crawl-heavy platform; Perplexity was at 194 and OpenAI near 1,100 [5]. By mid-2026, Radar’s trailing 28-day figure for Anthropic had fallen from about 4,600 in late June to below 2,000 in July as the operator split traffic into purpose-specific agents. OpenAI was in the high hundreds, Perplexity near 190, and Google near five [3]. Even after the largest improvement in the dataset, the most crawl-heavy AI platform still fetched hundreds of times more pages per visitor than search.

Three caveats matter. Ratios are window-specific and cannot be averaged. Referrals from native applications often lack a `Referer` header, which can overstate the imbalance [4]. And the operator publishing the metric also sells the remedy.

Purpose helps explain the ratio because model-training fetches produce no referral by design. Among AI-specific crawlers Cloudflare classifies, training rose from 72 percent of requests in July 2024 to 79 percent in July 2025, while search fell from 26 to 17 percent [5]. In the broader 2026 basket, which adds a mixed-use category, training was 52 percent of classified crawler requests in June 2026, up from 22 percent

R. Chowdhury is an independent researcher (ORCID 0009-0007-7032-2450). Contact: mailtorajarshi@gmail.com. This work was carried out in the author’s personal capacity and does not represent the views of the author’s employer.

Preprint. Submitted to *IEEE Internet Computing*, Special Issue on Future Internet Systems with LLMs and Agents, September 2026. Code and raw results: <https://doi.org/10.5281/zenodo.22647915>.

in spring 2025, while mixed-use crawlers exceeded 36 percent [1]. The snapshots use different denominators but point the same way. Mixed use makes the incentive problem structural: Google’s crawler serves both search and AI products, so a site that wants visibility in the engine supplying roughly 88 percent of referral traffic cannot refuse that crawler’s AI use. Cloudflare estimates this gives Google about twice the information access of leading AI companies [1].

II. WHERE THE CLICKS WENT

In March 2025, Pew Research Center recorded the browsing of 900 U.S. adults and captured 68,879 Google searches. When an AI-generated summary appeared, users clicked a traditional result in 8 percent of visits versus 15 percent without one, clicked a source cited in the summary in 1 percent, and ended the session in 26 percent versus 16 [6]. Pew does not claim causation. It also reconstructed AI-summary exposure by re-running recorded queries rather than capturing what each panelist saw. The crawl still happens. The click often does not.

III. A TEXT FILE FROM 1994

Against this backdrop, most sites still rely on `/robots.txt`. Proposed in 1994 and codified as RFC 9309 in 2022, it tells a conforming crawler which rules to follow but provides no server-side enforcement and is not access control [8]. Its vocabulary is allow or disallow, by named user agent and path prefix. It cannot distinguish training from indexing or a live answer request, state a price, license, or rate, or verify that a client calling itself `GPTBot` really is `GPTBot`. Later files in the same mold, including `humans.txt`, `security.txt`, `ads.txt`, `llms.txt` [20], and `ai.txt` [19], are also declarations with nothing behind them; `ai.txt` says so explicitly. `robots.txt` is a preference signal being asked to do the work of a licensing system, and it fails in four ways.

First, it produces incoherent policy. Longpre and colleagues audited 14,000 domains underlying the C4, RefinedWeb, and Dolma corpora [7]. From April 2023 to April 2024, `robots.txt` restrictions on AI crawlers went from nearly nonexistent to covering more than 5 percent of tokens in those corpora and more than a quarter of tokens from the most actively maintained domains. Terms-of-service restrictions covered 45 percent of C4, and the two channels often contradicted each other. The authors warn that because the file cannot distinguish a commercial training crawler from an archive or research crawler, publishers reacting to the former also block the latter.

Second, the web cannot agree on what to say. Radar’s directive analysis finds `GPTBot` is both the most explicitly disallowed AI crawler and the most explicitly allowed [3], a sign that the format is standing in for a negotiation it cannot support.

Third, it is not reliably honored. In August 2025, Cloudflare documented Perplexity fetching pages from sites that had disallowed its declared crawler, using undeclared user agents and rotating source networks [9]. Whatever one concludes about that dispute, the core weakness is clear: the mechanism

relies on the client truthfully identifying itself, with no way to verify the claim.

Fourth, it is blunt where precision matters. With more than a third of classified crawler requests coming from mixed-use bots [1], a rule meant to block training can also remove a site from discovery. Cloudflare frames the choice for a small site as allowing AI training or losing discoverability [15].

IV. WHAT EXISTS TODAY, AND THE GAP

A replacement is taking shape across four threads. Table 1 compares them.

For identity, Web Bot Auth lets an automated client sign requests with a key published in a well-known directory on its operator’s origin, allowing a server to verify the claimed operator. It builds on RFC 9421 HTTP Message Signatures [13], carries the operator’s HTTPS origin in a `Signature-Agent` dictionary keyed by signature label, and identifies keys by JWK thumbprint. On September 1, 2026, the IETF Web Bot Auth working group adopted it as a Standards Track document [11]. A companion draft defines a Signature Agent Card that advertises identity, purpose, and rate expectations [12]. The draft explicitly excludes authorization, delegation, and an intent vocabulary.

For preferences, the IETF AI Preferences working group is standardizing a vocabulary for how automated systems may use content, plus a way to attach those preferences through `robots.txt` and HTTP headers. The vocabulary is at revision 07, and the attachment draft, revised in August 2026, updates RFC 9309 [10]. The charter excludes enforcement and authentication, so the result is intentionally a richer preference layer, not access control.

For pricing, Cloudflare’s Pay Per Crawl, in private beta since July 1, 2025, lets a site allow, charge, or block each identified AI crawler. Charged crawlers receive HTTP 402 Payment Required, with Cloudflare as merchant of record [14]. On July 1, 2026, Cloudflare said it was beginning to reshape this into Pay Per Use, paying publishers when content appears in an answer rather than per fetch because fetch counts are a poor proxy for value. It calls this an experiment with Ceramic.ai and You.com [16]. Independent gateways use a similar model: Fairfetch returns 402 with a price and usage category and settles over x402, but the agent fetches through Fairfetch’s endpoint rather than the origin [18].

Cloudflare currently ships the most complete proprietary composition. Its July 2026 taxonomy separates Search, Agent, and Training crawlers. A content-use signal, `use=immediate`, `reference`, or `full`, extends the Content Signals convention in `robots.txt` but remains a preference. A Forwarded header carrying `for="openai"` with `use="reference"` conveys transitive trust through intermediaries using RFC 7239. Verified status is no longer default-allow, can be revoked when a bot abuses content-use signals, and is unavailable to bots that reproduce content in full [15].

The open, multi-vendor, standards-track work verifies identity but not terms. The preference work can express intended use but, by charter, does not enforce it. Every mechanism that does enforce access today is tied to a proxy, so a site

TABLE I
MECHANISMS GOVERNING MACHINE ACCESS AND WHAT EACH EXPRESSES AND ENFORCES, AS OF SEPTEMBER 2026.

Mechanism	Identity verified	Purpose	Terms or price	Delegation	Enforced where	Status
robots.txt (RFC 9309)	No	No	No	No	Nowhere, advisory	Standard
llms.txt, ai.txt	No	Partly (ai.txt actions)	No	No	Nowhere, preference	Community proposals
AIPREF vocab and attach	No	Yes (train-ai, search)	No	No	Nowhere, by charter	IETF WG drafts 07 and 05
Content Signals use=	No	Partly (use level)	No	No	Nowhere, preference	Vendor convention
Web Bot Auth	Yes (RFC 9421)	Per bot, via Agent Card	No	No	Origin or proxy	IETF WG document, Sept. 2026
Forwarded transitive trust	Relies on WBA	Per hop	No	Operator only	Proxy	Vendor proposal
Pay Per Crawl (402)	Vendor bot directory	Per bot category	Price per fetch	No	Proxy only	Vendor beta, Pay Per Use experiment announced
Fairfetch gateway (402, x402)	No	Usage category	Price per fetch	No	Gateway only	Open-source product
terms.txt exchange	Yes (Web Bot Auth)	Per request, signed	terms.txt, per path and purpose	Yes, agent-bound, scoped	Any origin	This article

gets it only by routing traffic through a particular company. A publisher on Cloudflare’s free plan gets a purpose-aware allow list and a 402 negotiation surface; a publisher running its own server gets a text file. Meanwhile, the unit of account moved from per crawl to per use within a year, showing that the market has not settled what it is pricing. The rest of this article shows that access terms and compensation need not live in a vendor dashboard.

V. DESIGN: A CONSENT AND COMPENSATION EXCHANGE

We compose the existing pieces into one exchange at the HTTP request boundary. That is the one point where identity, purpose, and terms meet, and it lets an origin enforce policy without renting a proxy. Figure 1 shows the architecture and flow.

Because terms differ by client, the design separates four client classes. A training crawler fetches content to build a model. A search crawler fetches it to build an index that refers traffic. A service-operated agent fetches in real time for a service, such as an answer engine grounding a response. A user-delegated agent fetches for a specific person and should inherit that person’s entitlements, such as a subscription, without revealing that person’s identity to the origin.

In the exchange, an agent holds its operator’s signing key and, when needed, a delegation token from the user’s identity provider and a payment voucher from a settlement service. All are obtained before the request. The agent then sends an ordinary HTTP request with up to six headers. Signature-Agent uses the working-group draft’s dictionary form, sig1="https://bot.example", to name the operator’s origin; its well-known key directory is fetched once and cached under the (URL, key) pair. Signature-Input and Signature carry an RFC 9421 signature over @method,

@authority, @path, the signature-agent member keyed to the label, Access-Intent, and, when present, Access-Delegation and Access-Payment. Validity is limited to five minutes and includes a nonce, a keyid containing the JWK SHA-256 thumbprint, and tag="web-bot-auth". Access-Intent is an RFC 9651 dictionary declaring purpose and use, for example purpose="agent", use="reference". Access-Delegation and Access-Payment carry compact signed tokens. The origin verifies the signature, atomically reserves the nonce before any discovery fetch, parses intent, checks required delegation and scope, evaluates the path-and-purpose terms, and verifies payment when required. It then serves the content with a signed Access-Receipt, returns 402 with price and terms, 401 for identity failures, or 403 for policy refusals. Unsigned requests are handled per path: public paths are served, while protected paths return 403 with an Accept-Signature challenge.

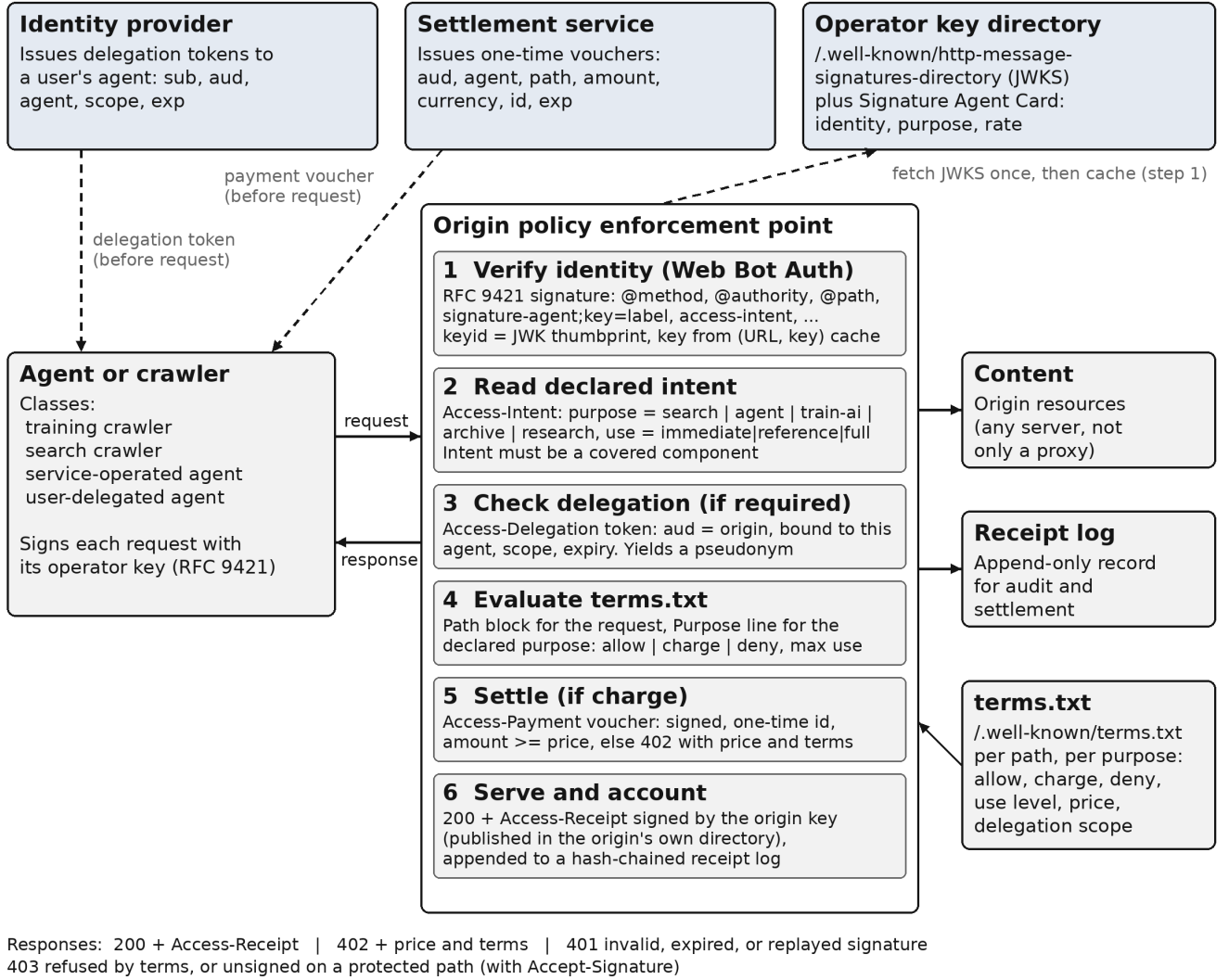


Fig. 1. The consent and compensation exchange at the HTTP boundary. Solid arrows show the request path; dashed arrows show out-of-band setup completed beforehand.

terms.txt is the discoverable half. An origin publishes /.well-known/terms.txt, a robots.txt-style file that any client can fetch before its first request:

```
# /.well-known/terms.txt
Terms-Id: 2026-09-01
Receipt-Keys: https://origin.example/.well-known/http-
  ↳ message-signatures-directory
Payment: voucher https://settle.example/.well-known/
  ↳ settlement

Path: /articles/
Unsigned: allow
Purpose: search    allow    use=reference
Purpose: agent     allow    use=reference  delegation=read:
  ↳ articles
Purpose: train-ai  charge  0.002 USD/request
Purpose: archive   allow    use=full

Path: /premium/
Unsigned: challenge
Purpose: agent     charge  0.01 USD/request  use=reference
  ↳ delegation=read:premium
Purpose: train-ai  deny
```

The header fields name the terms version, the origin's key directory, and accepted payment methods. The receipt-signing key is published in that directory under its thumbprint

so anyone can verify receipts. Each Path block sets the rule for unsigned requests and, for each purpose, an allow, charge, or deny decision with optional use ceiling, price, and delegation scope. The purpose vocabulary, search, agent, train-ai, archive, and research, maps to AIPREF terms and Cloudflare's taxonomy while adding the two categories that the Longpre audit shows are caught in the crossfire. The use levels match Content Signals' three levels, so a site's robots.txt preferences and enforceable terms can use the same language. As with robots.txt, the file itself enforces nothing; enforcement comes from the paired exchange.

Delegation is agent-bound and pseudonymous. A delegation token binds a pairwise pseudonymous subject to a specific operator, audience, named scope such as read:premium, and expiry. A stolen token is therefore useless to another operator, and one entitlement does not unlock another. Because the subject is a pairwise pseudonym from the identity provider, the origin learns that a subscriber's agent is present without learning the subscriber's identity. Vouchers are bound the same

way and include a one-time identifier.

Receipts are the accounting primitive. For every served request, the origin returns an `Access-Receipt` signed with its own key. It contains the operator, any subject pseudonym, declared purpose and use, terms identifier, path, a hash of the request signature, a timestamp, and an identifier. The origin appends each receipt to a hash-chained, append-only log, whose head commits to every receipt issued. This gives both sides a non-repudiable record of what was delivered under which terms, regardless of the unit a settlement model pays by. The protocol does not choose the unit of account; it makes the unit measurable.

For caching and intermediaries, receipted responses are marked private and vary on `Signature-Agent`, `Access-Intent`, and `Access-Delegation`, preventing a shared cache from giving one operator another's receipt. Unsigned responses remain cacheable as they are today. A CDN can forward the request unchanged for the origin to verify or verify it itself and pass a `Forwarded` assertion, as Cloudflare proposes. If an intermediary rewrites a covered component, the signature fails, which is the intended behavior.

VI. WHAT IS ENFORCED, WHAT IS AUDITED, WHAT IS CONTRACTUAL

A signature over a declared purpose proves who made the declaration and that it was not altered; it does not prove the declaration is true. A signed receipt proves content was delivered under stated terms; it does not prove what happened to the bytes afterward. Once content leaves the origin, HTTP cannot govern its use. The design should state that boundary plainly; much product literature does not.

Before delivery, the exchange enforces several facts: the request came from the named operator, is fresh, and is not a replay; the intent was signed rather than inserted by an intermediary; any delegation is valid, in scope, and bound to that operator; the terms permit that purpose and use level on that path; and a charged request carries valid, unspent payment. The prototype refuses requests that fail any of these checks. Its test suite also covers three bypasses an earlier version allowed: a client-supplied mode header, unsigned access to a protected path, and two identical signed requests racing a cold key lookup.

After delivery, the declared purpose is auditable against behavior. An operator that declares `use="reference"` but reproduces content in full, or declares `search` but never refers traffic, leaves evidence in receipts, citations, and traffic. The signed declaration makes that behavior attributable. The remedy is revocation of standing, as Cloudflare now does with `Verified status` [15].

What remains contractual is what a model does with content it lawfully received, including whether it trains on it. A protocol can make terms explicit, attributable, and priced, but it cannot make them self-executing. A header layer does not prevent training, and this design does not assume that it can.

In the threat model, false purpose claims cannot be prevented; they are handled through audit and revocation. Credential theft is limited by validity windows of minutes, key

rotation through the directory, and agent binding of tokens and vouchers, so a stolen token cannot be spent by another operator. Replay is blocked by an atomic check-and-reserve on (operator URL, `keyid`, `nonce`) before any asynchronous discovery, matching the draft's deployment guidance and fixing an ordering bug in the prototype's first version. Rewriting a covered component at a proxy breaks verification. The origin learns only the operator and a pairwise pseudonym, and receipts carry no direct user identifier. This layer does not stop unauthenticated scraping because it cannot force a client to sign. Instead, it makes honest access more capable: only signed requests can receive delegated entitlements, paid content, or receipts, and sites can require signatures on costly paths. Cost-based denial of service is the main systems risk because verification consumes CPU on every request. Mitigations are to check expiry, `nonce`, and key presence before cryptography, rate-limit by operator URL, and exploit the measured fact that refusal costs less than service.

VII. REFERENCE IMPLEMENTATION AND PER-REQUEST COST

We implemented `terms.txt` and the exchange in about 600 lines of dependency-free JavaScript on Node.js 22: a signature-base library with the `terms.txt` parser, an origin enforcement point, a directory resolver, and a harness. The library verifies the working-group draft's Ed25519 test vector E.2.1 and reproduces its thumbprint. The draft's printed signature base omits quotes around the `Signature-Agent` member, but the vector verifies only with them, as RFC 9421 requires. Discovery is bounded by size, key count, and time, refuses redirects, and coalesces concurrent fetches. The harness runs 24 checks each time. In addition to the identity and policy cases above, it verifies that `terms.txt` is served as text and parses to the enforced policy; key lookup is scoped to the (URL, key) pair; bad or cross-agent delegation is rejected; public unsigned requests are served and protected ones challenged; a client-supplied mode header is ignored; two identical signed requests racing a cold key lookup yield exactly one success; and the receipt log's hash chain matches the reported count and head. Benchmark mode is process configuration, with baseline and identity-only servers running as separate processes.

Table 2 reports loopback end-to-end overhead against the passthrough server. One 2.1 GHz Xeon vCPU was shared by the load generator and all server processes, using loopback without TLS. We ran five independent tests at concurrency 1 and five at 32, each with fresh processes and 10,000 measured requests per scenario after 3,000 warm-up requests. Headers were pre-signed, so client signing is excluded, while the load generator's round trip is included. The passthrough control ran first and last in every run. Unqueued latency was identical in both positions, but first-position throughput was about half the final value because of warm-up, so the table uses the final control as baseline. The key directory was fetched once per agent origin per process, taking about 35 ms on loopback. Across ten runs, median Ed25519 verification was 120 microseconds (118 to 122), delegation verification 122, receipt signing 42, and signature-base construction 4.

TABLE II

LOOPBACK END-TO-END OVERHEAD RELATIVE TO THE PASSTHROUGH SERVER: MEDIAN OVER FIVE INDEPENDENT RUNS, WITH RANGES IN BRACKETS. ONE SHARED VCPU, NO TLS.

Scenario	Result	p50 ms at conc. 1	Added ms	Req/s at conc. 32	Header bytes
Passthrough control (final)	200	0.054 [0.052-0.056]	0	15,892 [15,079-16,604]	0
Unsigned, allow path	200, no receipt	0.069 [0.066-0.076]	0.015	8,920 [8,529-9,782]	0
Identity only, three Web Bot Auth headers	200	0.255 [0.246-0.262]	0.201	3,027 [2,856-3,152]	392
Signed search, receipt and log	200 + receipt	0.392 [0.386-0.425]	0.339	2,424 [2,311-2,485]	458
Delegated agent, receipt and log	200 + receipt	0.548 [0.534-0.563]	0.494	1,846 [1,765-1,873]	825
Charged path, no payment	402 + terms	0.426 [0.418-0.431]	0.373	2,227 [2,152-2,254]	825
Charged path, voucher, receipt and log	200 + receipt	0.704 [0.691-0.729]	0.650	1,438 [1,395-1,459]	1,206
Forged signature	401	0.266 [0.260-0.290]	0.212	3,554 [3,390-3,608]	458

Three results stand out. Identity verification is the largest single cost; intent, `terms.txt` evaluation, receipt, and logging add about 0.14 ms beyond it. Refusal is cheaper than service: a 402 adds 0.37 ms versus 0.65 for the paid path, and a forged signature is rejected in 0.21 ms, limiting the leverage of a cost-based attack. Throughput falls from about 15,900 to 1,400–3,000 requests per second because one shared core runs Ed25519 in a single thread. Worker threads or a native backend would raise that ceiling, while 2,400 requests per second per vCPU for the common search path already exceeds most origins’ load. These are marginal loopback costs, not Internet latency; real-traffic deployment remains to be studied. The source, README, MIT license, and ten raw result files are archived at <https://doi.org/10.5281/zenodo.22647915> (release v0.1 of <https://github.com/rch0wdhury/terms-txt>).

VIII. OBJECTIONS

Referrals are the wrong metric. In part, yes, and that strengthens the case. An agent can compare twenty retailers, buy from one, and deliver a conversion without a referral trail. If value no longer flows mainly through clicks, accounting has to move to the one point all parties share: the request. Cloudflare’s shift from per-crawl to per-use pricing reflects the same search for a value-based unit of account.

Pricing access will enclose the open web. The Longpre audit suggests enclosure is already happening, and crudely, because the current instrument cannot distinguish a research crawler from a commercial pipeline [7]. A protocol that lets a site say yes to archives, researchers, and indexing, but no or pay to commercial training, can reduce that overblocking.

Operators will not comply. Signed requests make non-compliance detectable rather than invisible, and only signed requests receive delegated entitlements, paid content, or receipts. The EU AI Act also requires general-purpose model providers to honor machine-readable reservations of rights, which is easier to demonstrate against one standard signal than a patchwork of proprietary ones [17].

IX. WHAT CAN BE DONE NOW, WHAT MUST BE STANDARDIZED, WHAT WOULD CHANGE OUR MIND

Any origin can implement the core exchange today without a proxy: publish `terms.txt`, verify signatures under the

Web Bot Auth draft, cover an intent header with the signature, and issue receipts. Our prototype does all four in about 550 lines, and equivalent nginx, Apache, or Caddy modules are a matter of engineering. Adopting sites immediately gain attributable logs and a negotiation surface, even before operators agree to pay.

Several pieces still need standardization. AIPREF’s vocabulary should bind to a signed request-side intent component; today it exists only as a response-side preference. Web Bot Auth needs a delegation token format and scoping model, which neither charter covers. `terms.txt` needs a grammar and well-known location, and settlement systems need a receipt format they can consume. None is large, and the Web Bot Auth group’s September 2026 adoption of its protocol draft creates a natural place to raise them.

The argument is falsifiable. If signed per-request intent does not reduce mixed-use crawling once deployed, purpose declaration is not the lever. If sites that publish terms and receipts see no change in operator behavior or crawl-to-referral ratios over a year, the incentive does not live at the origin. If receipt logs fail to reconcile with answer-engine citation reports, receipts are not a usable unit of account. A large publisher could test the first two within a year of deployment.

Validation would look like a public, vendor-neutral dataset of crawls, referrals, and receipts with standard definitions. That would let the next version of this argument rely less on measurements from a party that also sells the remedy. The bargain that financed the open web was never written down, and the numbers say it is gone. Its replacement may live in the network or in a vendor dashboard. The pieces needed to put it in the network now exist.

X. A NOTE ON THE DATA

Cloudflare Radar figures come from the public API: `crawl-to-refer` at `radar/bots/crawlers/summary/crawl_refer_ratio` and `purpose` at `radar/ai/bots/summary/crawl_purpose`. Values were retrieved from June through August 2026 and rounded. Table 2 is computed by `aggregate.js` from the archived result files.

REFERENCES

- [1] A. Weiss, Z. Albertson, and E. Lanfear, “Content Independence Day, one year on: building the business model for the agentic Internet,” Cloudflare Blog, Jul. 1, 2026. [Online]. Available: <https://blog.cloudflare.com/agentic-internet-bot-report/>
- [2] Thales, “2026 Bad Bot Report: Bad Bots in the Agentic Age,” Imperva, Apr. 2026. [Online]. Available: <https://www.imperva.com/resources/resource-library/reports/2026-bad-bot-report/>
- [3] Cloudflare, “AI Insights,” Cloudflare Radar. [Online]. Available: <https://radar.cloudflare.com/ai-insights> (accessed Sep. 2026).
- [4] Cloudflare, “The crawl before the fall of referrals: understanding AI’s impact on content providers,” Cloudflare Blog, Jul. 1, 2025. [Online]. Available: <https://blog.cloudflare.com/ai-search-crawl-refer-ratio-on-radar/>
- [5] Cloudflare, “The crawl-to-click gap: Cloudflare data on AI bots, training, and referrals,” Cloudflare Blog, Aug. 29, 2025. [Online]. Available: <https://blog.cloudflare.com/crawlers-click-ai-bots-training/>
- [6] A. Chapekis and A. Lieb, “Google users are less likely to click on links when an AI summary appears in the results,” Pew Research Center, Jul. 22, 2025. [Online]. Available: <https://www.pewresearch.org/short-reads/2025/07/22/google-users-are-less-likely-to-click-on-links-when-an-ai-summary-appears-in-the-results/>
- [7] S. Longpre et al., “Consent in Crisis: The Rapid Decline of the AI Data Commons,” in Proc. NeurIPS 2024, Datasets and Benchmarks Track. arXiv:2407.14933.
- [8] M. Koster, G. Illyes, H. Zeller, and L. Sassman, “Robots Exclusion Protocol,” IETF RFC 9309, Sep. 2022.
- [9] Cloudflare, “Perplexity is using stealth, undeclared crawlers to evade website no-crawl directives,” Cloudflare Blog, Aug. 4, 2025.
- [10] IETF AI Preferences WG, draft-ietf-aipref-vocab-07 (P. Keller and M. Thomson, Eds.) and draft-ietf-aipref-attach-05 (G. Illyes and M. Thomson), Internet-Drafts, Aug. 2026, work in progress. [Online]. Available: <https://datatracker.ietf.org/wg/aipref/documents>
- [11] T. Meunier and S. Major, “HTTP Message Signatures for automated traffic,” draft-ietf-webbotauth-httpsig-protocol-00, IETF Internet-Draft, Sep. 1, 2026, work in progress. [Online]. Available: <https://datatracker.ietf.org/doc/draft-ietf-webbotauth-httpsig-protocol/>
- [12] M. Guerreiro, U. Kirazci, and T. Meunier, “Registry and Signature Agent Card for Web Bot Auth,” draft-meunier-webbotauth-registry-03, IETF Internet-Draft, Jun. 2026, work in progress.
- [13] A. Backman, J. Richer, and M. Sporny, “HTTP Message Signatures,” IETF RFC 9421, Feb. 2024.
- [14] Cloudflare, “Introducing pay per crawl: enabling content owners to charge AI crawlers for access,” Cloudflare Blog, Jul. 1, 2025. [Online]. Available: <https://blog.cloudflare.com/introducing-pay-per-crawl/>
- [15] J.-H. Lee and B. Becker, “Your site, your rules: new AI traffic options for all customers,” Cloudflare Blog, Jul. 1, 2026. [Online]. Available: <https://blog.cloudflare.com/content-independence-day-ai-options/>
- [16] M. Conroy, “Making AI search smarter,” Cloudflare Blog, Jul. 1, 2026. [Online]. Available: <https://blog.cloudflare.com/making-ai-search-smarter/>
- [17] Regulation (EU) 2024/1689 (Artificial Intelligence Act), Art. 53(1)(c), Official Journal of the European Union, Jul. 12, 2024.
- [18] Fairfetch, “fairfetch: the web protocol for the agentic economy,” GitHub repository, 2026. [Online]. Available: <https://github.com/Fairfetch-co/fairfetch>
- [19] W. Song et al., “ai.txt: A Domain-Specific Language for Guiding AI Interactions with the Internet,” arXiv:2505.07834, May 2025.
- [20] J. Howard, “The /llms.txt file,” Answer.AI, Sep. 2024. [Online]. Available: <https://llmstxt.org>