Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 18 September 2026

M. Dierken
paywalls.net

17 March 2026

Conditional Access for HTTP
draft-dierken-conditional-access-http-00

Abstract

This document introduces conditional access to resources based on pricing constraints signaled via request and response headers. An origin states the conditions for serving a resource, and a client states the constraints under which it will accept it.

This specification applies this conditional access approach using standard HTTP request and response semantics while staying interoperable with intermediaries. It introduces a Pricing response header field that communicates current pricing terms and an If-Price-LTE request header field that allows a client to declare a maximum acceptable price.

When a request is served under such conditions, the response includes a Response-Id header identifying the served response context. This identifier can be used for reconciliation, auditing, and downstream usage reporting.

This specification defines access condition signaling only. Accounting and usage reporting may be implemented using a companion protocol such as usage-log.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 18 September 2026.

Table of Contents

1. Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF).

2. Introduction

HTTP provides mechanisms for conditional requests and access control, including authentication, authorization, rate limiting, and representation-state preconditions. These mechanisms allow servers to express conditions based on identity, resource state, or server capacity. However, HTTP does not currently provide a standardized mechanism for expressing economic or policy conditions under which a resource may be served. This specification introduces such a mechanism while remaining compatible with existing HTTP conditional request semantics.

Many web resources today are accessed under conditions such as:

* usage pricing
* quota limits
* licensing terms
* contractual access policies

Historically, these conditions have often been implemented using JavaScript or proprietary APIs layered above HTTP. This approach works well for browsers but creates interoperability problems for automated clients and intermediaries that interact directly with HTTP semantics.

Automation systems, APIs, and AI agents increasingly retrieve web content without executing JavaScript. In such environments, access conditions implemented through client-side code cannot be reliably expressed or enforced.

This document defines a protocol mechanism that allows an origin server to explicitly communicate access conditions using HTTP metadata. Clients can then declare the conditions under which they are willing to receive the resource.

This mechanism restores capabilities historically implemented through JavaScript paywalls and tracking pixels into explicit HTTP protocol semantics suitable for automated clients and intermediaries.

Access conditions may be dynamic and may vary across time, identity, or other policy factors. Because the Pricing field explicitly communicates the minimum acceptable price, clients typically have little incentive to probe for lower prices. However, clients may issue repeated requests to observe changes in pricing across time, identities, or other request characteristics. This specification

therefore allows servers to publish stability metadata such as valid_until and next_floor while still enabling efficient price discovery.

3. Representation Lifecycle

Conditional access introduces a lifecycle for served representations:

1. Access conditions are communicated.
2. The client declares acceptable conditions.
3. The origin serves the representation if conditions are satisfied.
4. The served response receives a durable identifier.

The identifier allows the served response to be referenced later for purposes such as:

* auditing and reconciliation
* operational diagnostics
* downstream usage reporting

A separate protocol may be used to report downstream usage, which is particularly applicable to cached representations.

4. Terminology

Origin The server hosting the resource.

Client A software agent requesting the resource.

Access Conditions Constraints under which the origin is willing to serve a representation.

Response Context The specific instance of a response served by the origin.

5. Protocol Overview

This specification introduces two HTTP header fields:

* Pricing
* If-Price-LTE

It also defines the Response-Id response header used to identify served responses.

A typical interaction proceeds as follows:

1. A client requests a resource.

2. The origin communicates current access conditions.

3. The client declares acceptable conditions.

4. The origin serves the resource if conditions are satisfied.

6. Price Discovery

A client may request a resource without declaring an acceptable price.

Example:

```
GET /resource HTTP/1.1
Host: example.com
Authorization: Bearer agt_XYZ
```

If the resource requires payment or another access condition, the origin may respond:

```
HTTP/1.1 402 Payment Required
Pricing: floor=0.02, valid_until=@1743595200, next_floor=0.05, effective=@1743552000,
currency="USD", unit="request"
```

The Pricing field communicates a price quote for accessing the resource. A quote may include the current minimum acceptable price (floor) as well as additional metadata describing the stability or evolution of that price.

6.1. Price Quote Metadata

Servers MAY include additional metadata describing the stability and evolution of pricing conditions. These fields allow clients to make informed decisions and to understand when pricing is expected to remain stable or change.

floor The minimum acceptable price at the time the response is generated.

valid_until Timestamp indicating how long the current price quote is expected to remain valid.

next_floor Indicates the minimum acceptable price that will apply after the next scheduled price change.

effective Timestamp indicating when next_floor becomes active.

Together, these members describe the lifecycle of a price quote. The current quote applies when the response is generated. When next_floor and effective are present, they describe the next scheduled pricing change. When valid_until is present, it indicates how long the current quote is expected to remain stable.

Servers SHOULD ensure that valid_until intervals are long enough to provide useful stability information and reduce unnecessary repeated requests.

Example:

```
HTTP/1.1 402 Payment Required
Pricing: floor=0.02, valid_until=@1743595200, next_floor=0.05, effective=@1743552000,
currency="USD", unit="request"
```

This response indicates that the current floor price is 0.02. After the effective time, the minimum acceptable price will be at least 0.05.

7. Conditional Access Request

A client may include a constraint declaring the maximum acceptable price.

Example:

```
GET /resource HTTP/1.1
Host: example.com
Authorization: Bearer agt_XYZ
If-Price-LTE: 0.03; currency=USD; unit=request
```

If the constraint is satisfied, the origin may serve the response.

```
HTTP/1.1 200 OK
Pricing: applied=0.02, currency="USD", unit="request"
Response-Id: resp_82fd
```

If the constraint is not satisfied:

```
HTTP/1.1 402 Payment Required
Pricing: floor=0.05, currency="USD", unit="request"
```

When price quote metadata such as valid_until is present, clients should treat the quote as stable within that interval and can use that metadata to avoid unnecessary repeated requests made only to observe whether pricing has changed.

8. Pricing Header

Pricing communicates pricing conditions and related quote metadata for the resource.

The field value is an RFC 8941 Structured Field Dictionary.

Example:

Pricing: floor=0.02, currency="USD", unit="request"

Possible members include:

floor minimum acceptable price

valid_until Timestamp indicating how long the current price quote is expected to remain valid.

next_floor Indicates the minimum acceptable price that will apply after the next scheduled price change.

effective Timestamp indicating when next_floor becomes active.

applied price applied to a served response

currency ISO-4217 currency

unit normalization unit

The applied member MUST appear when the resource is served under conditional access.

The applied member MUST NOT appear in responses that do not serve the resource.

9. Price Quote Stability

Because the Pricing field explicitly communicates the minimum acceptable price (floor), clients typically have little incentive to probe for lower prices. However, clients may issue repeated requests to observe changes in pricing across time, identities, or other request characteristics. Servers concerned about such behavior may publish stability metadata such as valid_until or next_floor.

10. Response Identifier

Response-Id identifies the origin-served response context.

Example:

Response-Id: resp_7Y2f

The identifier is opaque to clients.

A Response-Id value is assigned by the origin and MUST uniquely identify the served response context.

Servers SHOULD include a Response-Id when serving a response under conditional access.

The identifier can be used for:

* auditing
* reconciliation
* downstream usage reporting

11. Pricing Units

This specification defines two units:

request
cpm

cpm represents price per 1000 requests.

Servers MUST normalize units when evaluating If-Price-LTE.

12. Response Semantics

200 resource served

402 price constraint not satisfied

403 access disallowed

429 rate limited

13. Authentication

Conditional access based on economic or policy conditions typically requires identifying the client.

Authentication mechanisms may include:

* OAuth bearer tokens
* mutual TLS
* other HTTP authentication schemes

The authentication method is outside the scope of this specification.

14. Interaction with Caching

Served responses may be cached according to normal HTTP semantics.

Cached representations may later be used without contacting the origin.

A Response-Id allows the served response context to be referenced by downstream usage reporting mechanisms when such reporting is implemented.

Pricing metadata such as valid_until, floor, or next_floor describes the server’s view of pricing conditions at the time the response was generated. These fields do not modify HTTP caching semantics. Intermediaries and caches MUST continue to follow standard HTTP caching rules when determining response freshness.

15. Relationship to Usage Reporting

Conditional access defines the terms under which a representation may be served.

In deployments where cached representations are reused without contacting the origin, accounting may require reporting downstream usage.

The pricing terms defined by this specification apply only to conditional access to a representation. They do not define or constrain any separate conditions that may apply to downstream usage of a served or cached representation. Any such usage policies are outside the scope of this specification.

Such reporting can be implemented using a companion protocol such as usage-log.

Usage reporting mechanisms are outside the scope of this specification.

16. Security Considerations

Origins should authenticate clients before serving resources under conditional access.

17. Privacy Considerations

Conditional access mechanisms operate at the client or operator level and do not require identifying individual end users.

Deployments should avoid including user-level identifiers in access metadata unless required for the application.

18. IANA Considerations

This document requests registration of the following HTTP header fields:

* Pricing
* If-Price-LTE
* Response-Id