Independent Submission
Internet-Draft
Intended status: Experimental
Expires: 11 May 2026

M. E. Jeftovic
easyDNS Technologies Inc.

7 November 2025

# Discovering x402 Resources via DNS TXT Records
## draft-jeftovic-x402-dns-discovery-00

## Abstract

This document defines a DNS-based discovery mechanism for locating x402 payment resources associated with a domain.

Domains publish one or more _x402 TXT records containing URLs where x402-compatible clients can obtain resource manifests and metadata over HTTPS.

The goal is to provide a lightweight, cache-friendly discovery vector that enables automated payment negotiation using the x402 protocol while keeping DNS records static and non-sensitive.

## 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 11 May 2026.

## Copyright Notice

Table of Contents



## 1. Introduction

The x402 protocol reactivates HTTP status code 402 ("Payment Required") as a mechanism for metered and token-based access control.

In this model, clients interact directly with an origin server or a designated facilitator to complete small, machine-to-machine payments before accessing a protected resource.

This draft defines a complementary DNS discovery mechanism for x402.

By publishing a _x402 TXT record as defined in the Domain Name System [RFC1034] [RFC1035], a domain can advertise one or more URLs where clients may retrieve x402 resource manifests and related metadata. This allows agents and applications to locate payment gateways before making application requests, reducing latency and enabling autonomous client behavior.

The TXT layer is a pointer-only facility; all dynamic data (prices, nonces, credentials) are obtained via HTTPS from the discovered URL(s).

## 2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Origin The HTTP host (and optional port) that serves content or APIs protected by x402.

x402 Resource Any endpoint implementing the x402 payment protocol.

Manifest URL An HTTPS URL published via _x402 TXT that, when fetched, returns a structured manifest describing available x402 resources.

Descriptor A short, optional free-text token that semantically identifies the service (e.g., "api", "shop", "tshirt").

## 3. Overview

To enable discovery, an origin publishes one or more TXT records under the underscored node name _x402 at the specific hostname where x402 resources are served:

```
dns _x402.example.com. 300 IN TXT
"v=x4021;url=https://example.com/.well-known/x402"
_x402.shop.example.com. 300 IN TXT
"v=x4021;descriptor=tshirt;url=https://shop.example.com/
x402-discovery"
```

Clients perform a DNS TXT lookup on _x402.<host> using the exact hostname from their intended HTTP request and obtain one or more URLs.

x402-protected resources SHOULD be served over HTTPS. While the x402 protocol operates at the HTTP layer, encrypted transport is strongly recommended for payment-related communications.

Each URL represents an HTTPS endpoint where further x402 metadata can be retrieved using a simple GET request.

The DNS record itself is static and only intended to point to discovery locations.

All dynamic data - such as pricing, currency, or facilitator credentials - are obtained over HTTPS from the URL(s) indicated.

Discovery Scope Each hostname requiring x402 discovery MUST publish its own _x402 TXT record. There is no inheritance from parent domains. Clients MUST query _x402.<exact-request-host> to discover payment resources for that specific hostname.

## 4. Record Format

Each _x402 TXT record MUST contain a single string formatted as semicolon-separated key/value pairs:

abnf record = "v=" version ";" [ descriptor ] "url=" https_url version = "x4021" descriptor = "descriptor=" desc_value ";" desc_value = 1*(%x20-21 / %x23-3A / %x3C-7E) ; printable ASCII excluding quote (0x22) and semicolon (0x3B) https_url = "https://" 1*(VCHAR)

* The optional descriptor provides a short, human-readable identifier for the service (e.g., "api", "shop", "tshirt").
  - Values SHOULD NOT contain semicolons or double quotes; if needed, publishers MAY percent-encode reserved characters per [RFC3986].
  - The descriptor is informational only and MUST NOT be relied upon for authorization or routing decisions.
  - Clients MAY display the descriptor to users to help identify the service.
* Implementations MUST ignore any keys they do not understand.
* Multiple TXT records at the same node are permitted and indicate multiple discovery URLs.

Examples:

dns _x402.api.example.com. 300 IN TXT "v=x4021;descriptor=api;url=https://api.example.com/.well-known/x402" _x402.shop.example.com. 300 IN TXT "v=x4021;descriptor=tshirt;url=https://shop.example.com/payment-info" _x402.example.com. 300 IN TXT "v=x4021;url=https://pay.example.com/x402"

## 5. Client Processing

1. Lookup - The client performs a DNS TXT query for _x402.<host> using the exact hostname from the intended HTTPS origin.

2. Parse - For each TXT record beginning with v=x4021;, extract the url and, if present, the descriptor.

3. Fetch - Issue an HTTPS GET to each discovered URL. Clients MUST validate TLS certificates per [RFC9110] and SHOULD require TLS 1.2 or higher.

4. Interpret - Parse the returned manifest. The manifest format and required fields are defined by the x402 core protocol specification [X402]. Implementations MUST support JSON format for manifests.

5. Negotiate - Proceed with the x402 payment handshake as defined by the x402 protocol, using the information obtained via HTTPS.

Multiple Record Handling If multiple _x402 TXT records exist for a hostname, clients SHOULD attempt to fetch all discovered URLs unless local policy dictates otherwise. The order of TXT records in DNS responses is undefined; clients MUST NOT assume any preference based on record order. Clients MAY use the descriptor field to select among multiple options, or MAY present choices to users.

Caching Clients SHOULD respect DNS TTL values for _x402 records and HTTP cache headers (Cache-Control, Expires) from manifest URLs. When manifest content changes more frequently than DNS records, operators SHOULD use appropriate HTTP cache directives rather than short DNS TTLs.

## 6. Operational Considerations

* The _x402 TXT record is expected to change infrequently and MAY have long TTLs (hours or days).

* Records MUST be published under each subdomain hostname where x402 resources are served (e.g., _x402.api.example.com for resources at api.example.com).

* Operators SHOULD sign their DNS zones with DNSSEC to prevent spoofing.

* Clients SHOULD prefer DNSSEC-validated responses when available.

* Only HTTPS URLs are permitted; clients MUST reject plain HTTP endpoints.

### 6.1. URL Path Conventions (Informative)

While this specification does not mandate any particular URL structure, operators MAY choose to follow the well-known URI convention defined in [RFC8615] by hosting discovery manifests at:

/.well-known/x402

This path aligns with similar discovery mechanisms in other protocols (e.g., Nostr NIP-05, Lightning Address LUD-16) and may aid in automated discovery tools. However, any HTTPS URL specified in the TXT record is valid, and clients MUST NOT assume any particular path structure.

Examples of valid discovery URLs include:

* https://example.com/.well-known/x402 (well-known convention)

* https://api.example.com/x402-manifest (custom path)

* https://cdn.example.com/payment-discovery (CDN-hosted)

* https://us.example.com/x402 (geographic routing)

Note: Use of /.well-known/x402 does not require IANA registration unless it becomes a widely-adopted convention requiring standardization.

## 7. Security Considerations

The TXT discovery mechanism does not convey any sensitive or dynamic payment data.

However, compromise of DNS responses could redirect clients to malicious payment endpoints.

To mitigate this, clients SHOULD verify DNSSEC signatures and servers SHOULD serve discovery manifests over HTTPS with valid certificates per [RFC9110].

Because TXT records are publicly visible, operators MUST NOT embed user-specific, time-sensitive, or confidential data in DNS records.

The optional descriptor is informational only and MUST NOT be relied upon for authorization or routing.

### 7.1. Privacy Considerations

DNS queries for _x402 records may reveal which services a client intends to use. This information may be logged by recursive resolvers and other DNS infrastructure.

Privacy-conscious clients SHOULD consider using DNS-over-HTTPS [RFC8484] or DNS-over-TLS [RFC7858] to prevent disclosure of discovery queries to network observers.

Operators should be aware that publishing _x402 TXT records publicly announces the availability of payment-required resources.

### 7.2. TLS Requirements

Clients MUST validate TLS certificates when fetching manifest URLs. Certificate validation MUST follow [RFC9110] and [RFC9525]. Clients SHOULD require TLS version 1.2 or higher and SHOULD prefer TLS 1.3 when available.

Clients MUST NOT proceed with payment negotiation if certificate validation fails.

## 8. IANA Considerations

IANA is requested to add the following entry to the "Underscored and Globally Scoped DNS Node Names" registry defined in [RFC8552]:

| Node Name | RR Type | Reference |
|---|---|---|
| _x402 | TXT | This document |

Table 1

## 9. References

### 9.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/rfc/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/rfc/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/rfc/rfc3986>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8552] Crocker, D., "Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves", BCP 222, RFC 8552, DOI 10.17487/RFC8552, March 2019, <https://www.rfc-editor.org/rfc/rfc8552>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

[RFC9525] Saint-Andre, P. and R. Salz, "Service Identity in TLS", RFC 9525, DOI 10.17487/RFC9525, November 2023, <https://www.rfc-editor.org/rfc/rfc9525>.

## 9.2. Informative References

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/rfc/rfc7858>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/rfc/rfc8484>.

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <https://www.rfc-editor.org/rfc/rfc8615>.

[X402] Coinbase Developer Platform, "x402 Protocol Specification", 2025, <https://x402.org>.

## Appendix A. Acknowledgments

The author gratefully acknowledges the architects of the original Hypertext Transfer Protocol, including Tim Berners-Lee, Roy Fielding, and the early contributors to the HTTP/1.0 and HTTP/1.1 specifications, whose design of the status code system (1xx-5xx) provided the long-dormant foundation for "Payment Required" (HTTP 402).

Recognition is also extended to Erik Reppel, Coinbase, and the broader x402.org team for their pioneering work in implementing and open-sourcing the modern x402 protocol, and for advancing the idea of agentic micropayments on the web.

Their efforts to operationalize HTTP 402 payments have provided the context and inspiration for this DNS-based discovery mechanism.

## Appendix B. Example Flow

text 1. A client wishes to access https://api.example.com/data 2. It queries DNS for _x402.api.example.com TXT 3. DNS responds with: "v=x4021;descriptor=api;url=https://api.example.com/.well-known/x402" 4. Client performs GET https://api.example.com/.well-known/x402 and receives a JSON manifest enumerating payable API endpoints. 5. Client follows x402 negotiation per core spec, receives 402 Payment Required, completes settlement, and retries successfully.

## Author’s Address

Mark E. Jeftovic
easyDNS Technologies Inc.
Email: markjr@easydns.com