Network Working Group
Internet-Draft
Intended status: Informational
Expires: 3 September 2026

Z. V. Zhou
Namefi

2 March 2026

Domain-Verified Skills (DVS) Protocol
draft-zzn-dvs-00

Abstract

This document defines the Domain-Verified Skills (DVS) protocol, a lightweight mechanism for AI Agents to discover, verify, and execute skill definitions served over HTTPS. A skill is a directory containing a SKILL.md entry point and optional bundled resources that instructs an AI Agent to perform a specific task or adopt a specific behavior.

The central design principle of DVS is that a skill's identity and trustworthiness are derived entirely from the HTTPS URL at which it is served -- no centralized registry or third-party certification is required. The operator of the URL's origin is the authoritative endorser of the skill.

This trust is formalized through the concept of a Trust Root: an HTTPS URL prefix declared by the skill publisher that scopes the trust boundary for their skills. A skill is considered verified if and only if its URL begins with the declared Trust Root. For brands with first-party domains, the Trust Root is the domain origin. For brands publishing on user-generated content platforms where the platform operator does not vouch for individual publishers, the Trust Root MUST be path-scoped to content the brand controls, ensuring that trust does not extend to the entire platform.

The protocol leverages the existing trust infrastructure of the Domain Name System (DNS) and Transport Layer Security (TLS) and is backward compatible with skills already served over HTTPS, including those hosted on GitHub or other platforms.

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/.

Copyright Notice

Table of Contents

1. Introduction

Current AI Agent skills (such as those in the Claude Agent Skills protocol [CLAUDE-SKILLS]) are primarily distributed via centralized code repositories or platform-specific upload mechanisms. This creates several friction points:

* Identity Ambiguity: Users cannot easily verify if a skill hosted on a third-party platform genuinely belongs to a brand.

* Hosting Friction: Brands must manage external accounts and synchronization instead of using their existing web infrastructure.

* Security Risks: Malicious skills can spoof brand names on open platforms to exfiltrate data.

The Domain-Verified Skills (DVS) protocol returns to the fundamental logic of the Web: the URL is the Identity. By serving skills directly from a brand-controlled URL prefix, we leverage the existing global trust of the DNS and HTTPS infrastructure.

For brands with first-party domains, the Trust Root is the domain itself (e.g., https://microsoft.com/). For brands that publish on user-generated content platforms such as GitHub or Hugging Face, the Trust Root is scoped to the brand's path on that platform (e.g., https://github.com/microsoft/). This allows DVS to provide brand-verified skill identity across all hosting configurations, without requiring brands to self-host infrastructure.

This protocol is designed to be compatible with existing skill formats and distributions. A skill already hosted on GitHub (e.g., https://github.com/microsoft/repo/blob/main/.../SKILL.md) is immediately usable under DVS by registering https://github.com/microsoft/ as the Trust Root, with no changes to the skill files themselves. In particular, a skill directory conforming to DVS can be directly consumed by any agent that understands the SKILL.md convention, while also gaining the identity and trust properties conferred by domain-verified hosting.

2. Conventions and Definitions

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.

The following terms are used in this document:

Skill: A directory containing a SKILL.md entry point and optional bundled resources (scripts, templates, data files, additional instructions) that directs an AI Agent to perform a specific task or adopt a specific behavior.

SKILL.md: The mandatory Markdown entry-point file within a skill directory. It contains YAML frontmatter metadata and human-readable instructions for the Agent.

Agent: An AI system capable of fetching, interpreting, and executing Skill instructions.

Skill URL: The HTTPS URL of the skill directory (or its SKILL.md entry point). This URL serves as the globally unique identifier of the Skill.

Trust Root: An HTTPS URL prefix that defines the trust boundary for a skill or set of skills. A skill is considered verified under a Trust Root if and only if its Skill URL begins with the Trust Root prefix. For first-party brand domains, the Trust Root is typically the domain origin (e.g., https://microsoft.com/). For user-generated content platforms, the Trust Root MUST be scoped to a path controlled by the brand (e.g., https://github.com/microsoft/).

Hosting Domain: The domain component of the Trust Root URL, considered the network-level endorser of the Skill via DNS and TLS.

Bundled Resource: Any file within the skill directory other than SKILL.md, including additional instruction files, executable scripts, templates, schemas, and data files.

3. Core Protocol

The identity of a skill is defined strictly by its HTTPS URL.

3.1. Identity and Trust

Skills MUST be served via HTTPS [RFC9110]. Agents MUST NOT fetch or execute skills served over plain HTTP.

The identity and trust of a skill is defined by its Trust Root. A skill’s URL MUST begin with its declared Trust Root prefix for the skill to be considered verified under that root.

For first-party brand hosting, the Trust Root is the domain origin:

```
Trust Root: https://example.com/
Skill URL:  https://example.com/.well-known/
            skills/support/SKILL.md
```

For user-generated content platforms, the Trust Root MUST be scoped to a path controlled by the brand. Agents MUST NOT accept a bare UGC platform domain (e.g., https://github.com/) as a Trust Root, as this would confer trust to all content on the platform:

```
Trust Root: https://github.com/example-org/
Skill URL:  https://github.com/example-org/
            repo/blob/main/skills/SKILL.md
```

A skill URL that does not begin with its declared Trust Root prefix MUST be rejected by the Agent.

3.2. Discovery

Skills MAY be hosted at any valid URL path on a domain (e.g., https://example.com/skills/my-assistant/SKILL.md).

Official brand skills SHOULD be served from the well-known path prefix:

/.well-known/skills/{skill-name}/SKILL.md

This follows the conventions established by [RFC8615] for well-known URIs.

Skills MAY be indexed in the domain’s sitemap.xml [SITEMAP] to enable automated agent discovery. Agents supporting discovery SHOULD check for skill entries in the sitemap when exploring a domain’s available skills.

A domain MAY serve a skill index document at:

/.well-known/skills/index.json

The index document, if present, SHOULD contain an array of objects, each with name, description, and path fields pointing to available skills on the domain.

4. Skill Specification

4.1. Directory Structure

A Domain-Verified Skill is a directory containing at minimum a SKILL.md file. The directory MAY contain additional files organized by purpose:

```
my-skill/
+-- SKILL.md              (entry point - REQUIRED)
+-- ADVANCED.md           (additional instructions - OPTIONAL)
+-- REFERENCE.md          (detailed reference docs - OPTIONAL)
+-- scripts/
|   +-- process.py        (executable script - OPTIONAL)
|   +-- validate.sh       (executable script - OPTIONAL)
+-- templates/
|   +-- report.html       (template file - OPTIONAL)
+-- data/
    +-- schema.json       (data/reference file - OPTIONAL)
```

When served over HTTPS, the directory structure is represented by URL paths relative to the skill's base URL. For example, a skill at https://example.com/.well-known/skills/my-skill/ would have its entry point at:

https://example.com/.well-known/skills/my-skill/SKILL.md

And a bundled script at:

https://example.com/.well-known/skills/my-skill/scripts/process.py

4.2. SKILL.md Entry Point

Every skill directory MUST contain a file named SKILL.md. This file MUST be encoded in UTF-8 and served with the media type text/markdown [RFC7763].

The file consists of two parts:

1. YAML frontmatter (metadata) - REQUIRED

2. Markdown body (instructions) - REQUIRED

4.3. Metadata (Frontmatter)

SKILL.md files MUST begin with a YAML frontmatter block delimited by --- lines. The frontmatter MUST contain the following fields:

name: A short, human-readable name for the skill. MUST NOT exceed

64 characters. MUST contain only lowercase letters, numbers, and hyphens.

description: A brief description of the skill's purpose and capabilities, including guidance on when an Agent should trigger the skill. MUST NOT be empty. MUST NOT exceed 1024 characters.

Example:

```
name: customer-support
description: Handles common customer support inquiries for
  Acme Corp products. Use when the user asks about product
  returns, warranty claims, or order status.
```

4.4. Instructions (Body)

The body of SKILL.md, following the frontmatter, MUST contain human-readable instructions for the Agent. These instructions define the behavior, constraints, and capabilities of the skill.

Instructions SHOULD be written as clear, step-by-step procedural guidance. They MAY reference bundled resources using relative URLs (e.g., [see advanced guide](ADVANCED.md) or run the script at scripts/process.py).

4.5. Bundled Resources

Skills MAY include additional files alongside SKILL.md. These bundled resources fall into three categories:

4.5.1. Additional Instructions

Additional Markdown files (e.g., ADVANCED.md, REFERENCE.md, FORMS.md) provide specialized guidance, detailed API references, or extended workflows. These files:

* SHOULD use the .md extension and text/markdown media type.

* Are loaded by the Agent only when referenced from SKILL.md or when the task context requires them.

4.5.2. Executable Scripts

Scripts (e.g., Python, Shell, JavaScript) provide deterministic operations that the Agent can execute. These files:

* MUST reside within the same skill directory (same-origin).

* Are executed by the Agent via its runtime environment; only the script's output enters the Agent's context, not the script source code itself.

* MUST be subject to the Explicit Consent requirement defined in Section 5.2 before execution.

4.5.3. Data and Reference Materials

Data files (e.g., JSON schemas, CSV datasets, configuration templates, API documentation) provide factual lookup material. These files:

* MUST reside within the same skill directory (same-origin).

* Are read by the Agent on demand when the task requires specific reference information.

* Impose no context cost until actually accessed.

4.6. Progressive Loading

Agents implementing DVS SHOULD employ a progressive loading strategy to manage context efficiently:

Level 1 - Metadata (always loaded): The YAML frontmatter (name and description) is loaded at agent startup or skill registration time. This enables skill discovery with minimal token cost (approximately 100 tokens per skill).

Level 2 - Instructions (loaded on trigger): The body of SKILL.md is fetched and loaded into the agent's context only when the skill is triggered by a matching user request.

Level 3 - Resources (loaded as needed): Bundled resources (additional markdown files, scripts, data files) are accessed only when referenced during execution. Scripts are executed and only their output is loaded into context.

This three-level approach ensures that a domain can publish many skills without imposing context overhead on agents, as only the metadata of registered skills is persistently loaded.

5. Security and Permissions

5.1. Same-Origin Isolation

Agents MUST restrict a skill's automated access to resources within its Trust Root prefix. A skill with Trust Root https://github.com/example-org/ MUST NOT be permitted to automatically access resources outside that prefix (including other GitHub users or repos) without explicit user consent.

Bundled resources (scripts, data files, templates) referenced by a skill MUST reside within the same Trust Root prefix as the SKILL.md file. Agents MUST reject references to resources outside the skill's Trust Root unless the user explicitly approves the access.

This prevents a compromised or malicious skill from leveraging its trusted context to exfiltrate data from, or perform actions on, unrelated origins or paths.

5.2. Explicit Consent

Agents MUST display the source domain to the user and request explicit confirmation before executing any non-textual instructions contained in a skill (e.g., shell commands, API calls, file system operations, running bundled scripts).

The consent prompt SHOULD clearly indicate:

* The Trust Root of the skill.

* A description of the action to be performed.

* Any resources that will be accessed or modified.

Textual instructions (Markdown content) MAY be loaded without additional consent beyond the initial skill activation. Executable content (scripts, commands) MUST always require explicit consent.

5.3. DNSSEC

Domain operators SHOULD deploy DNSSEC [RFC4033] to prevent DNS spoofing attacks that could redirect agents to malicious skill files hosted on attacker-controlled infrastructure.

6. Composition

A skill MAY reference other skills via their full HTTPS URLs. When an Agent encounters a referenced skill URL during execution, it SHOULD dynamically fetch and load the referenced skill if the current context requires the extended capabilities it provides.

Each referenced skill is subject to its own Trust Root's trust and security policies. Agents MUST apply the Same-Origin Isolation policy (Section 5.1) independently to each loaded skill based on its own Trust Root.

When composing skills across domains, agents SHOULD clearly communicate to the user that the trust context is being extended to additional domains.

7. Security Considerations

The primary security property of this protocol is that trust is anchored to domain ownership. This inherits both the strengths and weaknesses of the existing Web PKI and DNS infrastructure.

Skill spoofing is mitigated by the HTTPS requirement, which ensures that only the legitimate operator of a domain can serve skills from that domain. DNSSEC (Section 5.3) provides an additional layer of protection against DNS-level attacks.

Agents implementing this protocol should be aware of the following risks:

* Domain compromise: If a domain is compromised, all skills served from it should be considered compromised.

* Subdomain delegation: Skills on subdomains should be treated as distinct trust contexts from the parent domain.

* UGC platform Trust Roots: On user-generated content platforms, agents MUST enforce path-scoped Trust Roots. Accepting a bare platform domain (e.g., https://github.com/) as a Trust Root would allow any user on that platform to publish skills that appear equally trusted as a legitimate brand's skills.

* Transitive trust in composition: When skills reference other skills (Section 6), the trust chain extends across domains. Agents should clearly communicate this to users.

* Script execution: Bundled scripts execute with the permissions of the agent's runtime environment. Malicious scripts could perform unauthorized file access, network calls, or data exfiltration. The Explicit Consent requirement (Section 5.2) mitigates this risk, but agents SHOULD also sandbox script execution where possible.

* External resource fetching: Skills that instruct agents to fetch data from external URLs pose particular risk, as fetched content may contain malicious instructions. Agents SHOULD treat externally-fetched content as untrusted.

8. IANA Considerations

This document requests registration of the well-known URI suffix skills in the "Well-Known URIs" registry established by [RFC8615].

URI suffix: skills

Change controller: Namefi

Specification document: This document (Section 3.2)

9. References

9.1. Normative References

[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>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/rfc/rfc4033>.

[RFC7763] Leonard, S., "The text/markdown Media Type", RFC 7763, DOI 10.17487/RFC7763, March 2016, <https://www.rfc-editor.org/rfc/rfc7763>.

[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>.

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

[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>.

9.2. Informative References

[CLAUDE-SKILLS]
Anthropic, "Agent Skills Overview", n.d., <https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview>.

[SITEMAP] sitemaps.org contributors, "Sitemaps XML Format", n.d., <https://www.sitemaps.org/protocol.html>.

Acknowledgments

The authors would like to thank the broader AI agent community for discussions that informed this protocol design.

Author's Address

Zainan Victor Zhou
Namefi
Email: zzn@namefi.io