Domain Connect

Pawel Kowalik

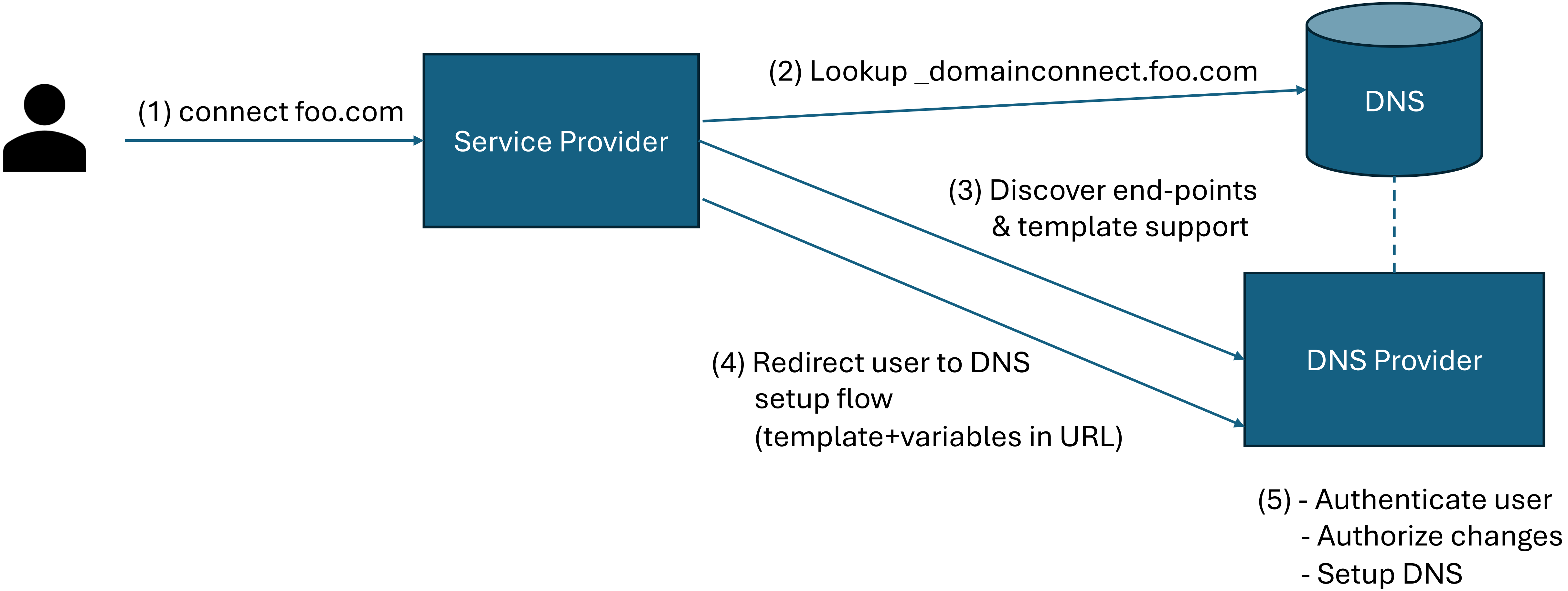
5 Oct 2025 – DCONN WG

draft-kowalik-domainconnect

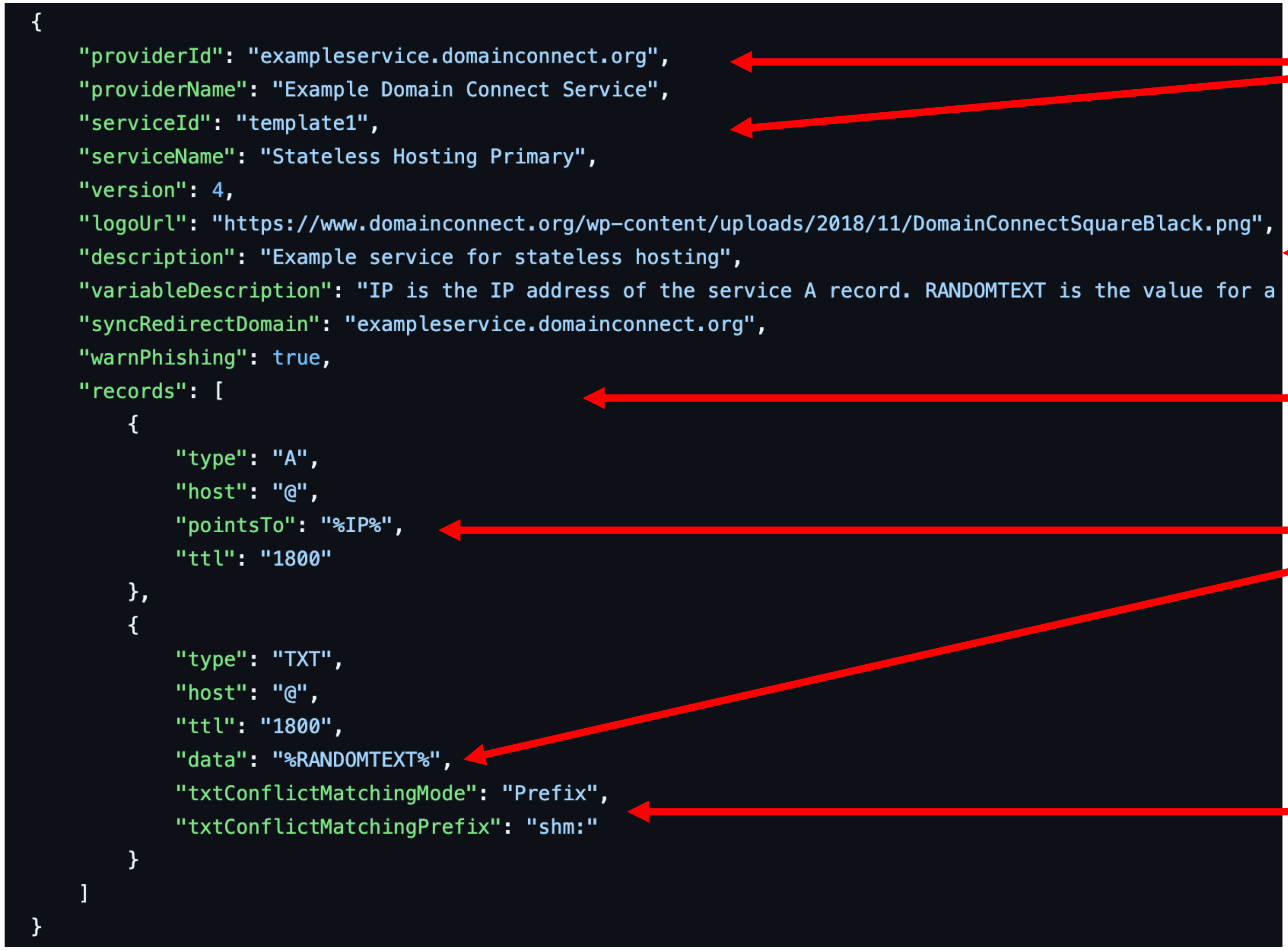
What it is

- An application-level protocol to allow service providers the ability to setup appropriate zone records to integrate with their products
- Makes DNS setup to integrate 3rd party (mainly SaaS) products way less complicated for end users
- Templates allowing both static values and variables
- Provide a customer experience that is simple, integrated and seamless
- Operational gains for both service and DNS providers

How it works


(1) connect foo.com
Service Provider
(2) Lookup _domainconnect.foo.com
DNS
(3) Discover end-points
& template support
DNS Provider
(4) Redirect user to DNS
setup flow
(template+variables in URL)
(5) - Authenticate user
- Authorize changes
- Setup DNS

Example template



```
{
    "providerId": "exampleservice.domainconnect.org",
    "providerName": "Example Domain Connect Service",
    "serviceId": "template1",
    "serviceName": "Stateless Hosting Primary",
    "version": 4,
    "logoUrl": "https://www.domainconnect.org/wp-content/uploads/2018/11/DomainConnectSquareBlack.png",
    "description": "Example service for stateless hosting",
    "variableDescription": "IP is the IP address of the service A record. RANDOMTEXT is the value for a
    "syncRedirectDomain": "exampleservice.domainconnect.org",
    "warnPhishing": true,
    "records": [
        {
            "type": "A",
            "host": "@",
            "pointsTo": "%IP%",
            "ttl": "1800"
        },
        {
            "type": "TXT",
            "host": "@",
            "ttl": "1800",
            "data": "%RANDOMTEXT%",
            "txtConflictMatchingMode": "Prefix",
            "txtConflictMatchingPrefix": "shm:"
        }
    ]
}
```

Identification

Metadata

Resource records

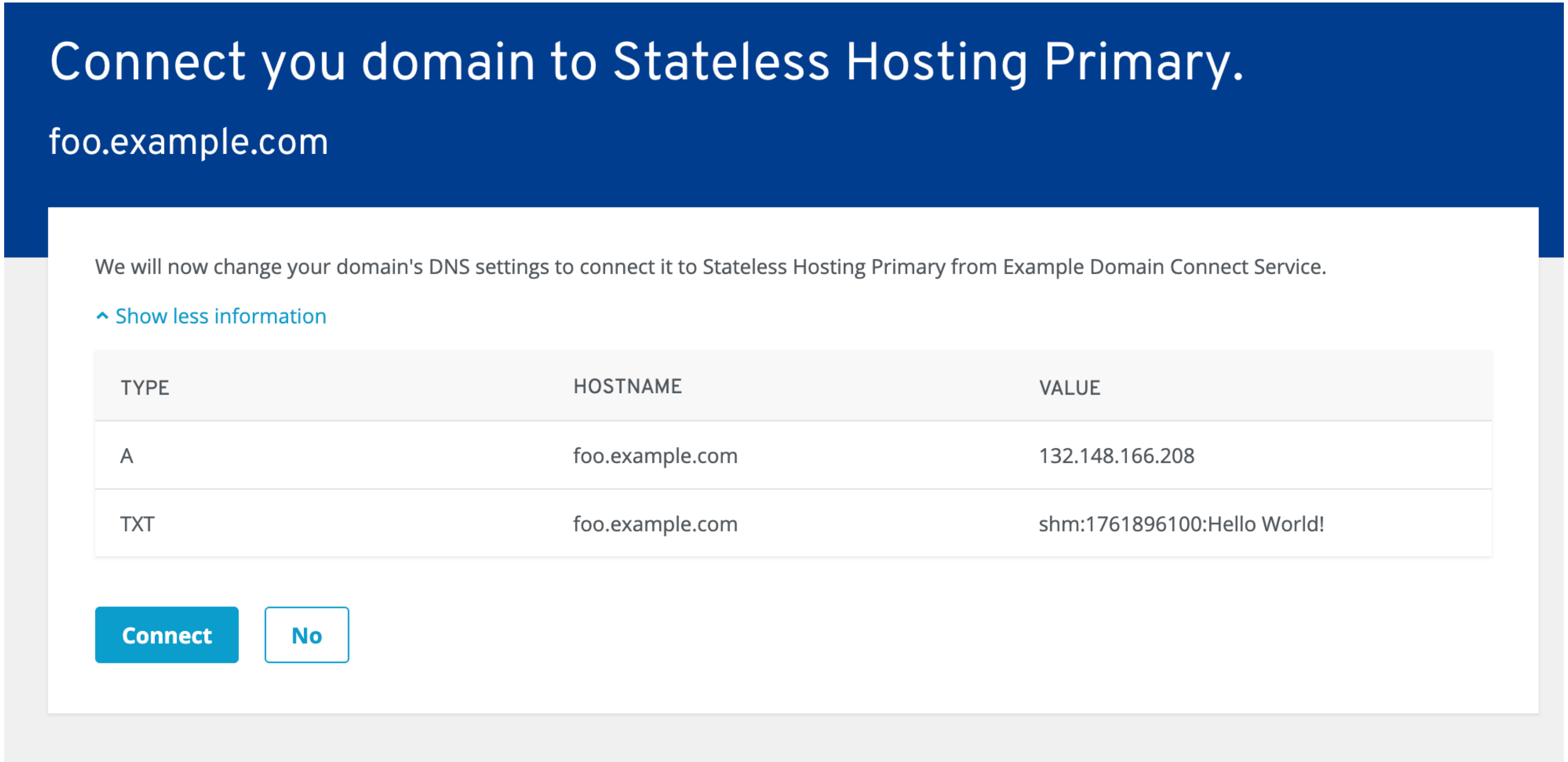
Template variables

Conflict resolution controls

Template application (synchronous)

```
HTTP/1.1 301 Found
Location:
https://domainconnect.dnsprovider.example/sync/v2/domainTemplates/providers/exampleservice.domainconnect.org/services/template1/apply
?domain=example.com&host=foo
&RANDOMTEXT=shm%3A1761896100%3AHello%20World%21
&IP=132.148.166.208
```

User consent (not defined in the protocol)



Connect you domain to Stateless Hosting Primary.

foo.example.com

We will now change your domain's DNS settings to connect it to Stateless Hosting Primary from Example Domain Connect Service.

TYPE	HOSTNAME	VALUE
A	foo.example.com	132.148.166.208
TXT	foo.example.com	shm:1761896100:Hello World!

Protocol features summary

- Templates that define the API contract between the providers
- 2 flows:
  - One-off synchronous flow
  - Recurrent asynchronous flow using on OAuth
- Conflict resolution on RR level
- Merging of SPF TXT records from different services
- Support for ephemeral records (like ownership verification)
- URL protected with signature

Implementation status

- DNS Providers
  - ~20 providers, incl. GoDaddy, IONOS, Cloudflare, Squarespace Domains (former Google), Wordpress.com or Plesk
  - 35% of the .com zone (May’24)
- Service Providers
  - 300 templates from over 120 providers, incl. O365, Google Workspace, Apple Cloud+, Weebly, Squarespace…

Issues reported so far

Security

Broader trust than warranted

- **Raised by:** Paul Hoffman
- The protocol assumes the user trusts the originating application with all their DNS records, not just the specific ones the application is expected to change.

This has been clarified in -01:

the scope of changes is constrained by the template and vetted by DNS operator when onboarding the template

Registrant understanding as attack vector

- **Raised by:** Paul Hoffman
- The protocol's security relies on the assumption that a registrant will understand the template's technical language and human-readable text when granting consent. Both are potential attack vectors.

This has been clarified in -01:

The scope of the changes is presented by DNS provider and controlled by them. Text added in "Trust Model" section.

It is rather not right to mandate UX of DNS provider, but some text can be added as further security consideration.

User warnings controllable by attacker

- **Raised by:** Paul Hoffman
- Warnings appear to be influenced or controlled by the (potentially malicious) application.

This has been clarified in -01:

> Some of the user-facing UI elements are indeed defined by the Service Provider (name, service name and logo). This is however vetted and onboarded by the DNS provider. If dynamic values are allowed it has to be same way allowed by the template definition.

Core security model & user consent

- **Raised by:** Paul Wouters (in Charter discussion)
- **Summary:**
  - "very nervous" about the security implications of the OAuth flow, and tokens flowing around with access to the zone
  - "unskilled registrants" must grant access to a SaaS party
  - template scoping "hard to follow" and is unsure a user can make a proper choice.

Most of it overlaps with the previous points.

In OAuth case the tokens are scoped to single template and resource records included in this template. Only subdomains are open-ended (none, single, multiple or any).

Plausible attacks exist despite request signing

- **Raised by:** Paul Hoffman
- Statement that plausible attacks exist that do not require forgery, even with request signing. No further details.

This issue is not addressed as details to the potential issue were not provided.

Once available it may be evaluated whether protocol changes are needed or risk included in the security considerations.

False sense of security

- **Raised by:** Paul Hoffman
- By being "somewhat safer" than manual editing, the protocol may lead to decreased user attentiveness, making them more vulnerable to attacks.

This is more a sociotechnical issue and a typical trade-off between usability and security.

My take: people who don’t understand DNS would be deceived the same way by copy-paste instruction (see: ClearFix attack)

Issues reported so far

Specification ambiguity, imprecision, and inconsistencies

Core misunderstanding due to poor documentation

- **Raised by:** Paul Hoffman
- acknowledged this was a "bad misunderstanding" *caused by* the draft's poor wording. The protocol's core security model—that the DNS provider vets all requests—is not clearly documented.

This has been addressed in -01, but no further review or feedback happened to confirm

HTTP Directorate Review (1)

- **Raised by:** Darrel Miller
- Use of hard-coded paths (e.g., /v2/…) instead of discoverable mechanisms
- Evaluate Linkset RFC 9264

Protocol has built in discoverability for main end-points.
Further changes would break the protocol for little value.

HTTP Directorate Review (2)

- **Raised by:** Darrel Miller
- Ambiguous variable substitution. Unclear handling of whitespace, data types, and input sources (query vs. JSON).

addressed in -01. Section 10.9.3.

HTTP Directorate Review (3)

- **Raised by:** Darrel Miller
- Inconsistent variable syntax. Use of both %var% and {var}.

{var} notation is used wherever protocol URI templates are defined. This has been reviewed in -01.

%var% substitution is used in all dynamic elements of the protocol, like templates or URLs from the discovery document.

This was done to avoid confusion of the two and injection attacks against bad implementation.

HTTP Directorate Review (4)

- **Raised by:** Darrel Miller
- Unclear redirect_uri communication and the relation to syncRedirectDomain

Revised in -01

HTTP Directorate Review (5)

- **Raised by:** Darrel Miller
- Potential parameter name collision. No namespace for template variables, risking collision with protocol parameters.

Clarified (event more) in -01 which variable names are built-in.

Change would be breaking for all the existing templates.

So far, no source of confusion from existing implementations.

HTTP Directorate Review (6)

- **Raised by:** Darrel Miller
- Unnecessary OAuth2 re-definition. Redefining OAuth2 parameters instead of deferring to RFC 6749.

This was defined to have a standalone specification. Asynchronous flow is using OAuth2, but is not 100% the same, due to additional parameters needed to describe scope (in the time as RAR was not yet even conceptualised).

HTTP Directorate Review (7)

- **Raised by:** Darrel Miller
- Ambiguous asynchronous apply. Unclear precedence for parameters via query string vs. JSON body.

Not addressed yet.

Needs input from implementers which precedence is better or whether query string option can be entirely removed.

HTTP Directorate Review (8)

- **Raised by:** Darrel Miller
- Not using HTTP Problem Details (RFC 9457)
- Registering a media type for templates

Not addressed.

Both changes may potentially be breaking for a client not expecting “new” media type.

Is it worth doing for the “pureness”?

Idea: use those only if the client defines media type it in “Accept” header.

Conflicting/overlapping record writes

- **Raised by:** Arnt Gulbrandsen
- The draft does not specify how to handle scenarios where multiple services attempt to write exactly same record content (e.g., two services adding same CAA record)

This is in fact covered by the standard protocol features (multiInstance and essential template settings).

Question: shall best practice be part of the core specification? Or maybe an appendix, same as examples right now?

Ambiguity and Flawed Logic in Section 11.5 (DS Record Updates)

- **Raised by:** Peter Thomassen
- Section 11.5 is confusing, stating DS updates require the "DNS Provider" to be the "registrar." If true, the operation would be internal and Domain Connect unnecessary.

Addressed in -01

Ambiguity regarding UI design

- **Raised by:** Paul Hoffman
- The draft gave impression of mandating specific browser UI flows, which is inappropriate for a protocol doc.

Addressed in -01. Flows are described using sequence diagrams without mandating any specific UI.

IANA registration for _domainconnect record

- **Raised by:** Marco Davids
- The _domainconnect TXT record used for discovery was not registered with IANA as per RFC 8552, a necessary step for standardization.

Addressed in -01

Protocol versioning

- **Raised by:** Peter Thomassen
- clarification on how protocol versioning is handled to ensure backward compatibility

Open. Shall it be defined in the core in very detail or may be delegated to “future 2.0” version, when it appears?

Ambiguity of underscore hosts

- **Raised by:** Paul Hoffman
- clarification on how to handle applying templates to "underscore hosts”, meaning using for example “_dmarc” in host parameter.

Open. My personal take is that it shall be forbidden, as no legit use-case seems to need it and it may lead to confusion.

Shall the same apply to template variables rendering host names?

Formal issues

Copyright notice

- **Raised by:** Mohamed Boucadair
- Domain connect specification on Website/Github contains standard MIT license copyright notice from GoDaddy.

This is fixed now with update issued by GoDaddy on 21.10.2025 to Creative Commons CC0 1.0 Universal Public Domain Dedication, waiving all copyright and related rights.

Commit 33e3262.

Feature requests

Ambiguity of underscore hosts

- **Raised by:** Sami Kerola
- Proposes adding an “extensions" object to the discovery response to allow providers (like Cloudflare) to advertise non-standard, provider-specific settings (e.g., "proxied": "bool", "flattened": "bool").

Does it fit the charter to define such extensibility?

Summary

- A lot of issues have been addressed in -01, but no really reviews teling if it’s sufficient
- All addressable security remarks are also covered. Is there more to cover?
- Some other open issues need WG input
- WG adoption?

Open Mic

Thank you

Pawel Kowalik
pawel.kowalik@denic.de
https://www.linkedin.com/in/pawelk/