

# Technical Disclosure Commons

---

Defensive Publications Series

---

August 2021

## Locale-Aware Domain-Connect Specification

Navneet Goel

Aditya Gopalakrishnan

Follow this and additional works at: [https://www.tdcommons.org/dpubs\\_series](https://www.tdcommons.org/dpubs_series)

---

### Recommended Citation

Goel, Navneet and Gopalakrishnan, Aditya, "Locale-Aware Domain-Connect Specification", Technical Disclosure Commons, (August 24, 2021)

[https://www.tdcommons.org/dpubs\\_series/4544](https://www.tdcommons.org/dpubs_series/4544)



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

## **Locale-Aware Domain-Connect Specification**

### **ABSTRACT**

Domain Connect is a standard that enables DNS providers to provide a mechanism for service providers to add or update DNS records on the domain, as follows. The service provider redirects their users (owners of domains) to the DNS provider for authentication and authorization, after which the service provider adds or updates the DNS records on behalf of the user. Although it is possible that the service provider interfaces with the user in a way consistent with the user's locale, the DNS provider might still be incognizant of the user's locale. Furthermore, even recent versions of the Domain Connect specification don't provide ways for a DNS provider to leverage the locale information of the user.

This disclosure describes techniques that preserve the locale/language as the user is passed along from the service provider to the DNS provider. These techniques enable a locale-specific user experience and a load balancing of traffic to the DNS Provider servers based on the user's location.

### **KEYWORDS**

- Domain Connect specification
- Domain name system (DNS)
- DNS provider
- Service provider
- DNS records
- Locale-awareness
- Sync flow
- Async flow

### **BACKGROUND**

An internet domain name such as example.com, makes it easy for users to access websites without having to memorize and enter numeric IP addresses. Users (owners of the domain) - either individuals or businesses - purchase domain names from domain registrars. The

domain name system (DNS) is the phonebook of the internet, mapping domain names to IP addresses by looking up DNS records maintained at name servers. A user can hire service providers to provide various services such as email, documents storage, word processing, advertisement serving, etc. for their website.

A DNS provider (not to be confused with service provider) authoritatively hosts the DNS records, for example, the actual IP address associated with the domain; the nameservers currently used by the domain; the subdomains listed under the domain; etc.

*Domain Connect* is a standard that enables DNS providers to provide a mechanism for service providers to place or update DNS records on the domain, thus freeing the user from having to manually set or update the records. Per the mechanism, the service provider redirects or navigates their user (owner of the domain) to the DNS provider's website for authentication and authorization. The service provider is thereby authorized to add or update the DNS records for the user by interacting with the DNS provider, e.g., using HTTP calls.

Templates [3], which are established contracts between a service provider and a DNS provider, are the primary tool used to add or update DNS records for a given domain. Service providers interact with DNS providers through either the sync flow mechanism or the async flow mechanism to add or update the DNS records using templates.

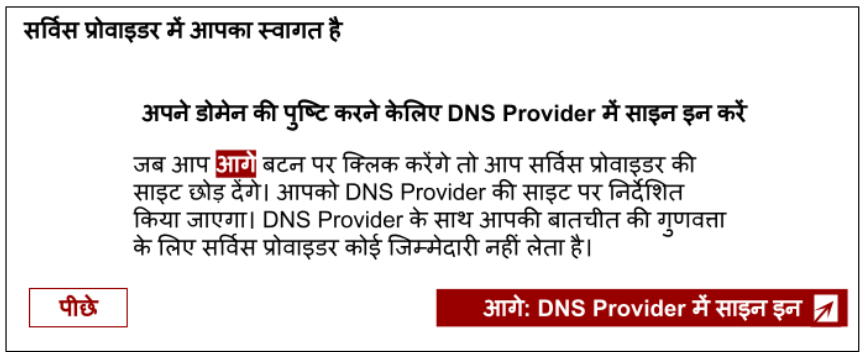
Under the async flow, an OAuth token is retrieved from the DNS provider, and one or more templates are applied on behalf of the user by the service provider. Under the sync flow, no OAuth token is retrieved; rather only a single template is applied at a time. Based on the type of flow (sync or async), different sets of HTTP calls are involved in the process, as described in the Domain Connect specifications [4].

The service provider typically uses some heuristics or rules around the name server records to guess the DNS provider. Additionally, the latest versions of the Domain Connect specification enable the service provider to discover the DNS provider (and the settings to integrate with it) automatically without using any heuristics using a mechanism called DNS provider discovery. Per the DNS provider discovery mechanism, the service provider follows a public TXT record on the *\_domainconnect* subdomain of a given domain to get the address of a settings server. The service provider fetches the settings of the DNS provider for the domain using an HTTP REST API call. The settings have a number of parameters in the form of a JSON object e.g.,

```
{
  "providerId": "xyzdomains.com",
  "providerName": "XYZ Domains",
  "providerDisplayName": "XYZ Domains",
  "urlSyncUX": "https://domainconnect.xyzdomains.com",
  "urlAsyncUX": "https://domainconnect.xyzdomains.com",
  "urlAPI": "https://api.domainconnect.xyzdomains.com",
  "width": 750,
  "height": 750,
  "urlControlPanel": "https://domaincontrolpanel.xyzdomains.com/?domain=%domain%",
  "nameServers": ["ns01.xyzdomainsdns.com", "ns02.xyzdomainsdns.com"]
}
```

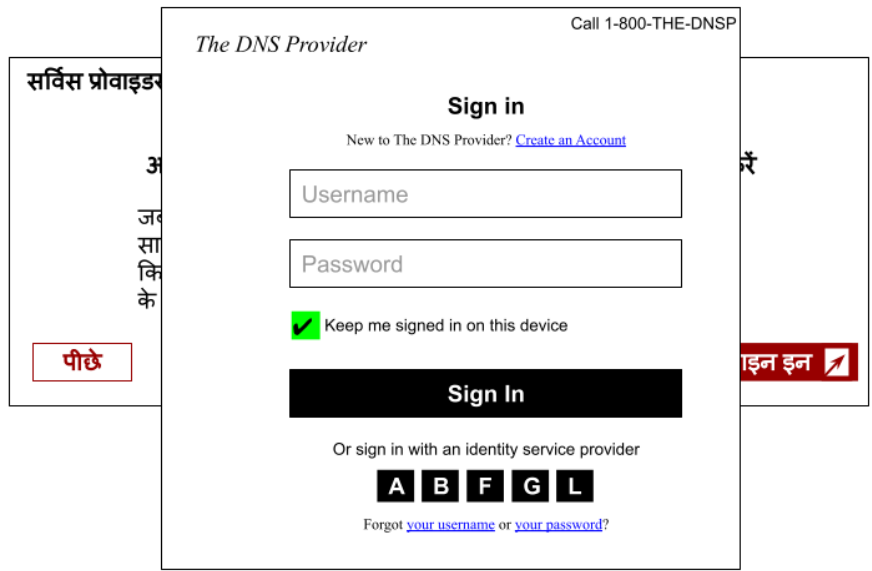
## PROBLEM

The problem is that the entire Domain Connect spec is not locale-aware with respect to both service provider and DNS provider, as depicted from the example below.



**Fig. 1: An example user interface provided by a service provider trying to add services on a particular domain provided by a DNS provider. The service provider is aware of the user’s locale and provides the user interface in the language of the locale (in this example, India:Hindi)**

Fig. 1 illustrates an example user interface provided by a service provider trying to add services on a particular domain provided by a DNS provider. The service provider is aware of the user’s locale and provides the user interface in the language of the locale (in this example, India:Hindi). As explained above, the user is invited to click on a forward button (in red) to be redirected to the website of the DNS provider so that they can sign in to the DNS provider to authenticate and authorize the service provider to add or update the DNS records on behalf of the user.



**Fig. 2: The DNS provider may be incognizant of the user’s locale**

Although the service provider interfaces with the user in a way consistent with the user's locale, the DNS provider can be incognizant of the user's locale. This is illustrated in Fig. 2, where, upon the user clicking the forward button to sign in, the DNS provider returns a sign-in window in English, without accounting for the user's locale. Locale information is lost when the user is handed over from the service provider to the DNS provider. This occurs even when the DNS provider is otherwise known to support the locale of the user and is a competitive domain provider in that locale.

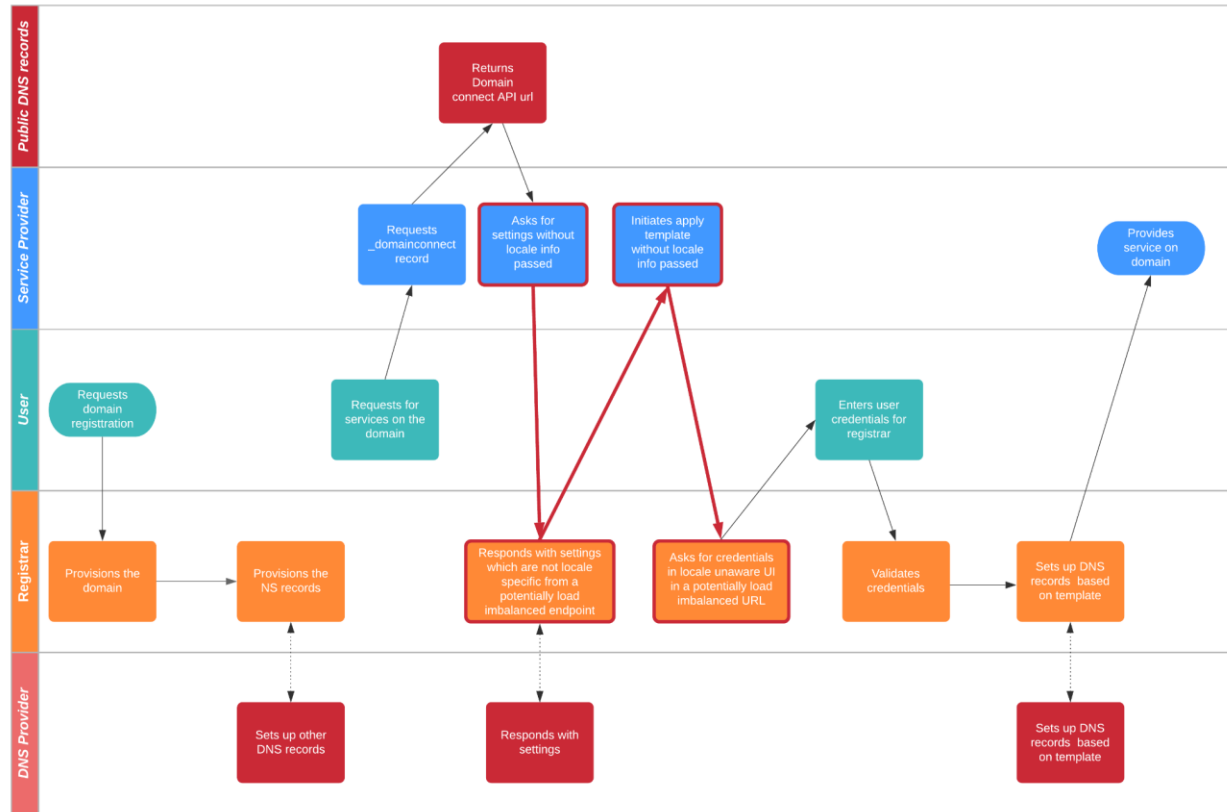
Thus, though the service provider and the DNS provider both offer a user interface in the locale/language of the user, the locale/language information is not passed along from the service provider to the DNS provider. Even though the DNS provider has a locale-specific user interface, the Domain Connect protocol does not currently leverage the locale-specific capabilities of the DNS provider. Indeed, the Domain Connect specification lacks locale-awareness, leading to a lower quality of user experience in this instance. There is no provision to specify the locale of the user in any of the urls involved, including the DNS provider discovery settings call, the sync apply-template url, the authorization url, etc.

The lack of accurate passing of locale information between the service provider and the DNS provider results in a generic (not locale-specific) user experience. It also results in potentially suboptimal traffic routing, e.g., the DNS provider routes the user to a generic (example.com) url which might potentially be load-imbalanced as well instead of routing to a load-balanced url (example.in) This routing of users worldwide in a potentially load-imbalanced manner can lead to bottlenecks.

The above-described issue of locale information getting dropped between the service provider and the DNS provider can manifest in four scenarios based on whether the DNS

Provider supports DNS Provider discovery or not and whether the DNS Provider supports the sync flow or async flow.

Scenario 1: DNS provider supports DNS provider discovery via sync flow



**Fig. 3: Synchronous flow with DNS provider discovery**

In this scenario, illustrated in Fig. 3, the service provider requests and receives settings without passing locale information (thick, red lines). Additionally, the service provider loads the apply template url (which points to the DNS provider UI) which is unaware of the locale of the user and is potentially load imbalanced.

**Example:** A domain of the user is “mydomain.in” which is hosted by the DNS provider “dnsprovider”. The *\_domainconnect* record contains the hostname “api.dnsprovider.com” for

“mydomain.in”. Assume that “dnsprovider” supports both US-English and India-Hindi on the following urls:

US-English	dnsprovider.com
India-Hindi	dnsprovider.in

A Service Provider providing services to “mydomain.in” in India-Hindi locale queries the *\_domainconnect* record to get the settings server url. The Service Provider then proceeds to make a settings call for “mydomain.in” by calling the settings url. The response might look as follows (note that locale information is not passed during the service provider request):

```
{
  "providerId":"dnsproviderid",
  "providerName":"dnsprovider name",
  "providerDisplayName":"dnsprovider display name",
  "urlSyncUX":"http://domainconnect.dnsprovider.com",
  "urlAsyncUX":"http://domainconnect.dnsprovider.com",
  "urlAPI":"http://api.domainconnect.dnsprovider.com",
  "width":750,
  "height":750,
  "urlControlPanel":"http://domaincontrolpanel.dnsprovider.com?domain=mydomain.in"
,
  "nameServers":[
    "ns01.dnsprovider.com",
    "ns02.dnsprovider.com"
  ]
}
```

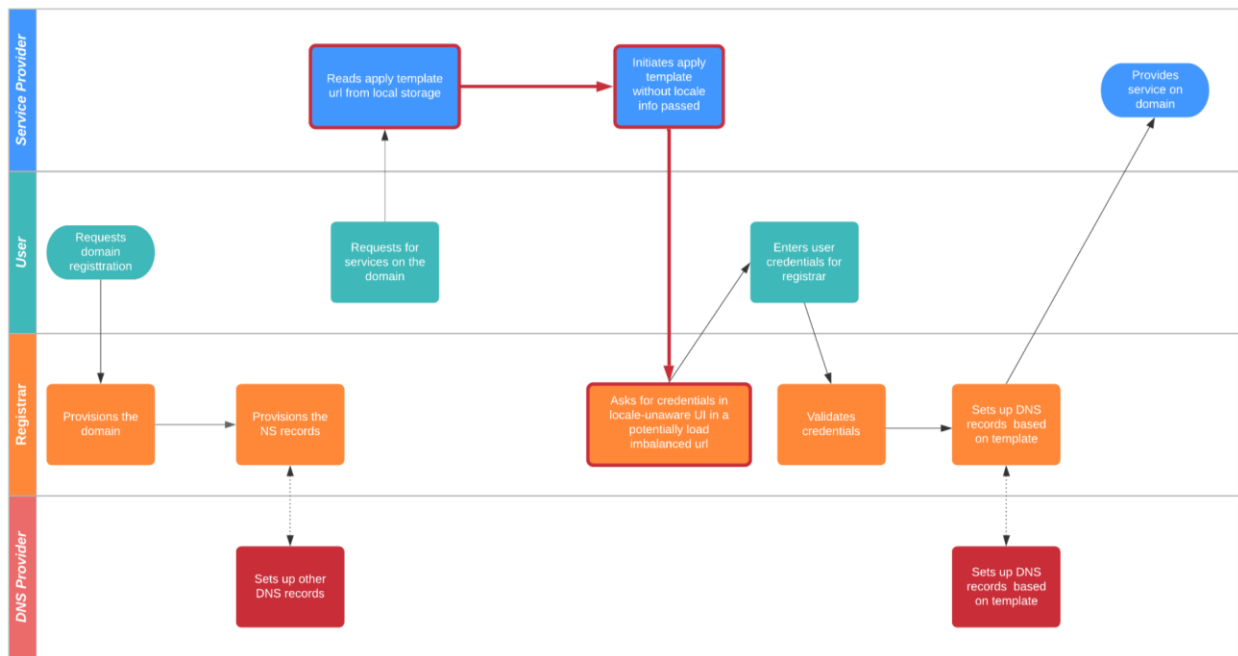
As illustrated in the above response, all the returned urls point to “dnsprovider.com” instead of “dnsprovider.in”. Even though the user sees the India-Hindi locale UI in the service provider website, the DNS provider UI will still be in US-English. Hence, the end-to-end user experience is uneven (or broken). Moreover, the DNS provider UI is loaded from a potentially load-imbalanced url (dnsprovider.com) instead of a load-balanced url (dnsprovider.in).



Scenario 2: DNS provider supports DNS provider discovery via async flow

Due to security concerns [2], this scenario is unsupported as per the specification.

Scenario 3: DNS provider doesn't support DNS provider discovery but supports sync flow



**Fig. 4: DNS provider doesn't support DNS provider discovery but supports sync flow**

In this scenario, illustrated in Fig. 4, the DNS provider establishes the contract with the service provider by sharing the apply-template url (thick, red lines) for the sync flow to the service provider. The apply-template url which is used for applying templates to add or update DNS records (per the latest version of the Domain Connect protocol) is incognizant of the locale.

**Example:** The domain of the user is “mydomain.in”, hosted by the DNS provider “dnsprovider”.

The established apply-template url at the service provider for “mydomain.in” is

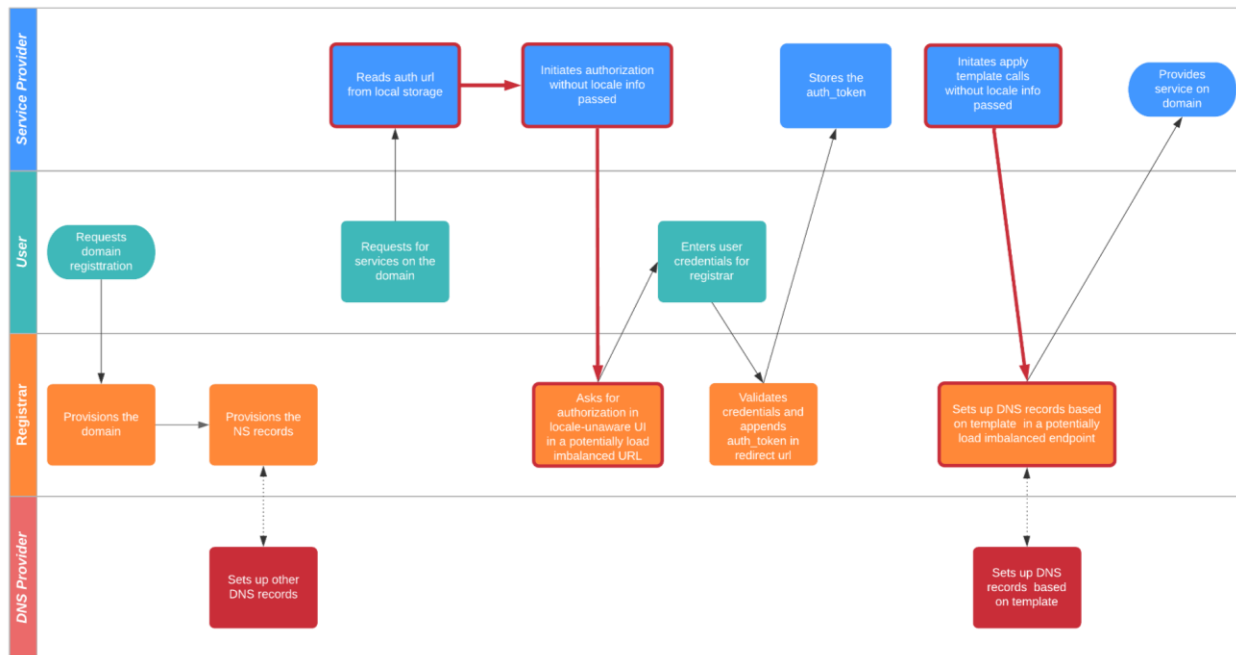
"<http://domainconnect.dnsprovider.com/v2/applytemplate/>". Assume that “dnsprovider” supports

both US-English and India-Hindi on following urls:

US-English	dnsprovider.com
India-Hindi	dnsprovider.in

A Service Provider providing services to “mydomain.in” in India-Hindi locale would apply the template url on "<http://domainconnect.dnsprovider.com/v2/applytemplate/>" without passing any locale information. Even though the user sees the India-Hindi locale UI in the service provider website, the DNS provider UI will still be in US-English. Hence, the end-to-end user experience is uneven (or broken). Moreover, the DNS provider UI is loaded from a potentially load-imbalanced url (dnsprovider.com).

*Scenario 4: DNS provider doesn't support DNS provider discovery but supports async flow*



**Fig. 5: DNS provider doesn't support DNS provider discovery but supports async flow**

In this scenario, illustrated in Fig. 5, the DNS provider establishes the contract with the service provider by sharing the authorization url and few other urls (thick, red lines) for async flow to the service provider. The authorization url which is used for authentication and

authorization in the DNS provider UI (per the latest version of the Domain Connect protocol) is incognizant of the locale of the user.

**Example:** The domain of the user is “mydomain.in”, hosted by the DNS provider “dnsprovider”.

The established authorization url at the service provider for “mydomain.in” is

"<http://domainconnect.dnsprovider.com/v2/authurl/>". Assume that “dnsprovider” supports both

US-English and India-Hindi on the following urls:

US-English	dnsprovider.com
India-Hindi	dnsprovider.in

A Service Provider providing services to “mydomain.in” in India-Hindi locale would load the auth-url on "<http://domainconnect.dnsprovider.com/v2/authurl/>" without passing any locale information. Even though the user sees the India-Hindi locale UI in the service provider website, the DNS provider UI will still be in US-English. Hence, the end-to-end user experience is uneven (or broken). Moreover, the DNS provider UI is loaded from a potentially load-imbalanced url (dnsprovider.com). A similar load imbalancing issue applies to the subsequent token\_url (for fetching/refreshing access token) and to the dns\_api\_url (to apply template asynchronously) as well.

## SOLUTION

This disclosure describes techniques, applicable to the Domain Connect protocol, that preserve the locale/language as the user is passed along from the service provider to the DNS provider. Per the techniques, the login screen of the DNS provider (Fig. 2) is automatically provided in the locale/language detected by the service provider (Fig. 1). Further, having been provided with the user’s locale/language, the techniques enable the DNS provider to route the user’s traffic to servers close to the user’s locale, and enable the DNS provider discovery call to

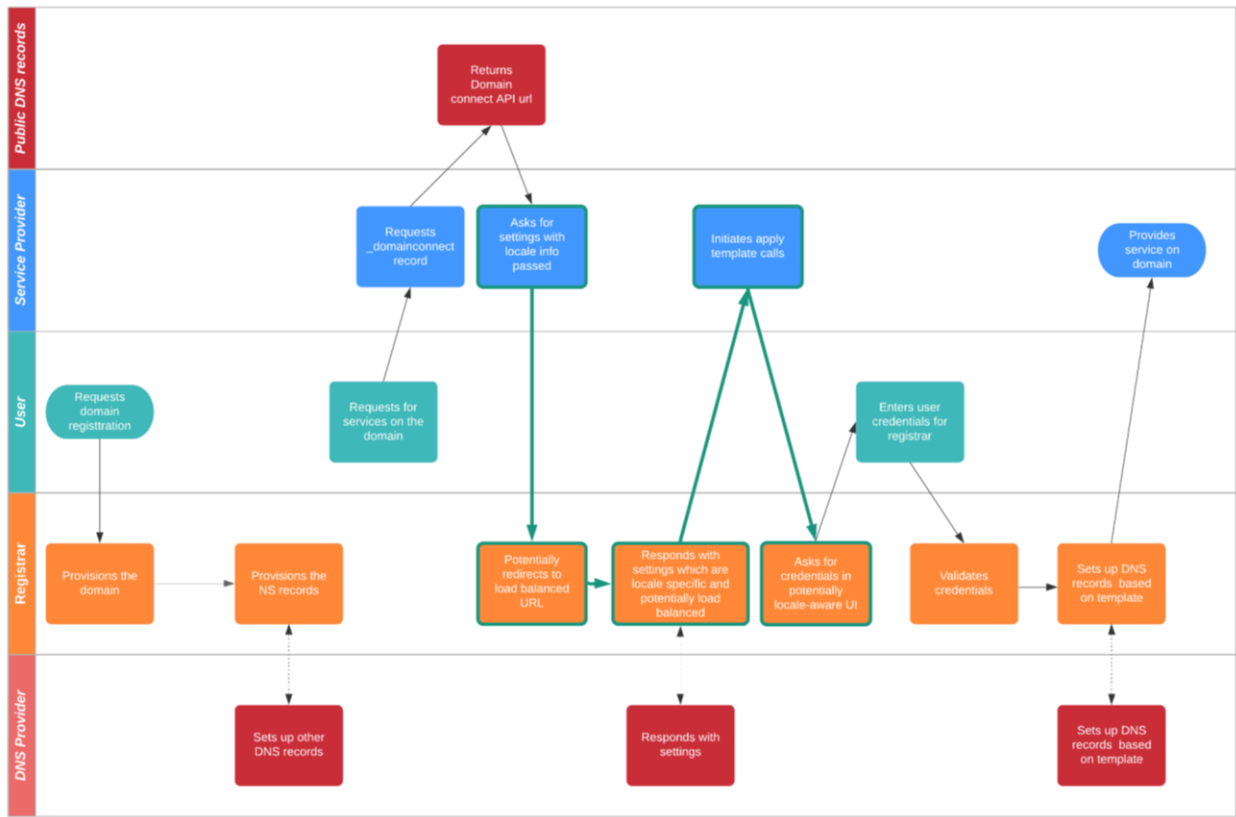
send the user to not generic ([example.com](#)) but to locale-specific ([example.in](#)) urls. In this way, the techniques enable not only a locale-specific experience for the user on the DNS Provider but also achieve load balancing of traffic to the DNS Provider servers.

Per the techniques, the parameters passed by the service provider to the DNS provider during Domain Connect integration are augmented with an extra locale query parameter. The DNS provider can use this extra locale parameter in various urls exposed by the DNS provider as part of the Domain Connect specification, as follows:

- In the settings call the DNS provider can use the extra locale parameter to return a localized and load-balanced version of the DNS provider discovery settings that includes url parameters typically part of the discovered settings e.g., *urlSyncUx*, *urlAsyncUx*, *urlAPI* etc. These urls in the settings call response will be localized and load balanced.
- In case of an established contract of urls between the service provider and the DNS provider, the service provider can pass an extra locale parameter to any of the urls required for the integration, thus enabling the DNS provider to serve a localized UI experience from load-balanced versions of those urls.

Various scenarios (sync/async flow; with/without support for DNS provider discovery settings) for the Domain Connect specification are described below.

Scenario 1: DNS provider supports DNS provider discovery via sync flow



**Fig. 6: In synchronous flow with DNS provider discovery, the DNS provider settings call accepts additional locale parameters and returns settings appropriate to the locale**

Per the techniques, illustrated in Fig. 6, the DNS provider settings url on the DNS provider (the first service-provider-to-DNS-provider call, in HTTPS), which provides the DNS provider settings for a given domain, is augmented to accept an additional locale parameter and the Service provider uses this to pass the detected locale (thick, green lines). If the DNS provider supports the locale included in the query parameters, it returns settings more appropriate to the locale e.g., *urlSyncUx*, *urlAsyncUx*, *urlAPI* etc.

If the DNS provider doesn't support the locale included in the query parameter, it can choose to ignore the query parameter in which case the user experience would be based on a default locale - as it is today. Thus, this technique is backward compatible. The service provider

receives from the DNS provider a JSON object with locale-specific settings, reads it, and directs the user to the urls therein, thereby achieving a locale-specific user experience from load-balanced urls.

**Example:** The domain of the user is “mydomain.in”, hosted by the DNS provider “dnsprovider”.

Assume that the *\_domainconnect* record contains the hostname “api.dnsprovider.com” for “mydomain.in”. Assume that “dnsprovider” supports both US-English and India-Hindi on following urls:

US-English	dnsprovider.com
India-Hindi	dnsprovider.in

A Service Provider providing services to “mydomain.in” in India-Hindi locale queries the *\_domainconnect* record to get the settings server url. The Service Provider then proceeds to make a settings call for “mydomain.in” by calling the settings url. In addition to the domain name the service provider also passes along a locale parameter e.g., India-Hindi. The response is augmented based on the locale parameter passed (including the display name, the optimal servers for the user’s locale, the specific locale parameters for each url, etc.)

```
{ "providerId":"dnsproviderid",
  "providerName":"dnsprovider name",
  "providerDisplayName":"डी एन एस प्रोवाइडर",
  "urlSyncUX":"http://domainconnect.dnsprovider.in?locale=hi_IN",
  "urlAsyncUX":"http://domainconnect.dnsprovider.in?locale=hi_IN",
  "urlAPI":"http://api.domainconnect.dnsprovider.in?locale=hi_IN",
  "width":750,
  "height":750,
  "urlControlPanel":"http://domaincontrolpanel.dnsprovider.in?domain=mydomain.in&locale=hi_IN",
  "nameServers":[
    "ns01.dnsprovider.in",
    "ns02.dnsprovider.in"
  ]
}
```

As illustrated in the above response, all the urls returned point to “dnsprovider.in”, which ensures that the user always visits a url optimal for their locale. Further, the India-Hindi UI in the service provider website survives the transition to the DNS provider UI because the DNS provider ensures that the locale is passed to the specific urls e.g., through the locale parameter (`locale=hi_IN`) in each of the urls. In this manner, the end-to-end user experience is in the detected locale for the user for both the service provider and the DNS provider. Furthermore, load balancing based on the user’s locale is achieved by routing to a server closer to the user’s locale i.e., “dnsprovider.in” as opposed to the generic “dnsprovider.com”.

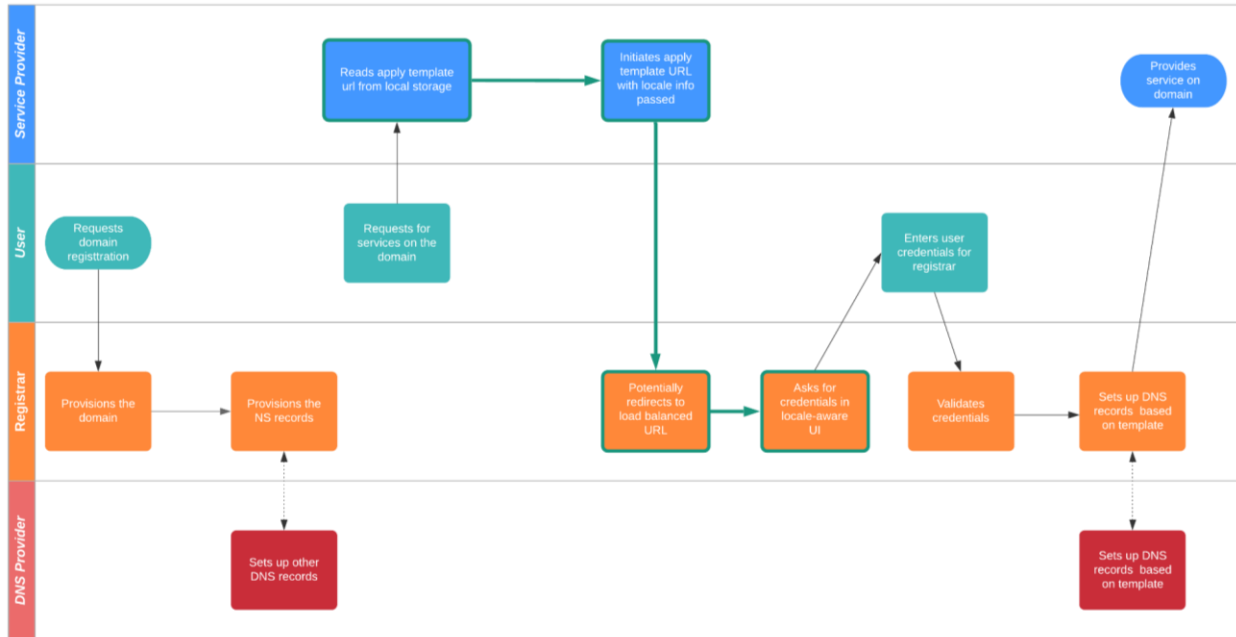
Thus, localization support is achieved on the fly with minimal changes to the specification. The service provider needs to make no change except passing the locale in the settings call. As explained earlier, the DNS provider can enable a localized experience to their users and also route them to a server closer to the user’s locale. As a result, the Domain Connect specification itself becomes locale-aware.

#### *Scenario 2: DNS provider supports DNS provider discovery via async flow*

As mentioned before, due to security concerns [2], this scenario is unsupported as per the specification.

If the DNS provider supports DNS Provider discovery on both the sync and async flows, then the procedure of scenario 1 can be used to preferentially integrate on the sync flow.

*Scenario 3: DNS provider doesn't support DNS provider discovery but supports sync flow*



**Fig. 7: If the DNS provider doesn't support DNS provider discovery but supports sync flow, the service provider initiates an *apply template url* with locale parameter**

As illustrated in Fig. 7, the *apply template url* accepts the locale parameter that the service provider can pass (thick, green lines). The locale parameter can be used by the DNS provider to route the request to an appropriate server and to enable a localized experience.

**Example:** The domain of the user is “mydomain.in”, hosted by the DNS provider “dnsprovider”.

The established apply-template url at service provider side for “mydomain.in” is

"<http://domainconnect.dnsprovider.com/v2/applytemplate/>". Assume that “dnsprovider” supports

both US-English and India-Hindi on the following urls:

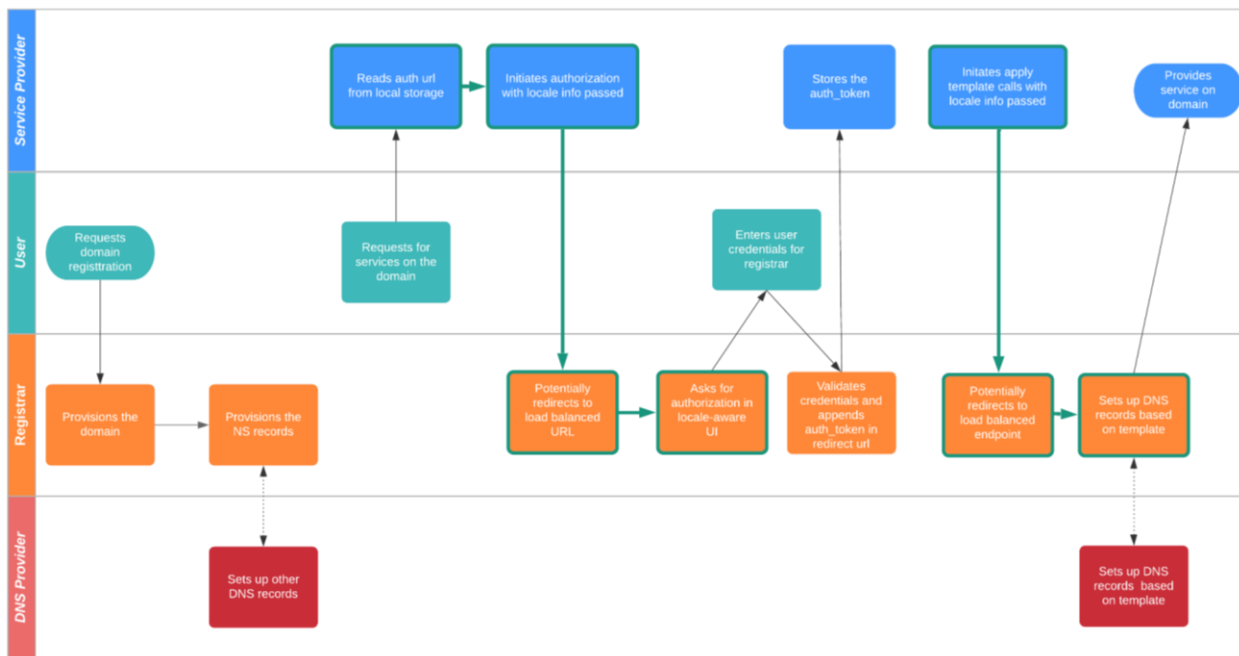
US-English	dnsprovider.com
India-Hindi	dnsprovider.in

A Service Provider providing services to “mydomain.in” in India-Hindi locale applies the template url on "[http://domainconnect.dnsprovider.com/v2/applytemplate?locale=hi\\_IN](http://domainconnect.dnsprovider.com/v2/applytemplate?locale=hi_IN)",



explicitly passing the known locale as a parameter. The DNS Provider redirects first to a server closer to the given locale e.g., to "[http://domainconnect.dnsprovider.in/v2/applytemplate?locale=hi\\_IN](http://domainconnect.dnsprovider.in/v2/applytemplate?locale=hi_IN)". The user sees a locale-aware India-Hindi DNS provider UI after the transition. In this manner, the end-to-end user experience is localized, i.e., in the India-Hindi locale. Furthermore, a url more appropriate to the locale, e.g., “dnsprovider.in” rather than the generic “dnsprovider.com”, is used, leading to optimal load balancing.

*Scenario 4: DNS provider doesn't support DNS provider discovery but supports async flow*



**Fig. 8: the DNS provider doesn't support DNS provider discovery but supports async flow, the authorization url accepts a locale parameter passed along by the service provider**

As illustrated in Fig. 8, the authorization url accepts a locale parameter passed along by the service provider (thick, green lines). The locale parameter can be used by the DNS provider to route the request to a server closer to the user's locale and to enable a localized experience.

The other urls used in async flow, e.g., *token\_url* and *dns\_api\_url*, can also be used to get servers appropriate to the user's locale.

**Example:** The domain of the user is “mydomain.in”, hosted by the DNS provider “dnsprovider”.

The established apply template url at the service provider for “mydomain.in” is

"<http://domainconnect.dnsprovider.com/v2/applytemplate/>". Assume that “dnsprovider” supports

both US-English and India-Hindi on the following urls:

US-English	dnsprovider.com
India-Hindi	dnsprovider.in

A Service Provider providing services to “mydomain.in” in India-Hindi locale navigates to the authorization on "[http://domainconnect.dnsprovider.com/v2/authurl?locale=hi\\_IN](http://domainconnect.dnsprovider.com/v2/authurl?locale=hi_IN)" explicitly passing the known locale as a parameter. The DNS Provider redirects first to a server closer to the given locale e.g., to "[http://domainconnect.dnsprovider.in/v2/authurl?locale=hi\\_IN](http://domainconnect.dnsprovider.in/v2/authurl?locale=hi_IN)". The user sees an India-Hindi locale in the DNS provider UI after the transition. In this manner, the end-to-end user experience is localized i.e., in the India-Hindi locale. Furthermore, a url more suitable for the locale, e.g., “dnsprovider.in” rather than the generic “dnsprovider.com” is used which leads to optimal load balancing. The other urls used in async flow, e.g., *token\_url* and *dns\_api\_url*, can also be thus augmented to get servers appropriate to the user’s locale.

If the DNS provider supports both sync and async flows but doesn’t support DNS Provider Discovery then the procedures of either scenario 3 or scenario 4 can be used, based on the agreement between the DNS provider and the service provider.

Thus, even when DNS provider discovery is not supported, localization support is achieved on the fly for each url involved in the sync or async flow. From the service provider’s perspective, the change is minimal because there is no need for the service provider to maintain a hardcoded mapping of the locale to DNS provider urls. The service provider need make no changes except passing the locale in the calls to the DNS provider. Specifically, the service

providers do not need to change their existing configurations for currently integrated DNS providers. Original urls remain unimpacted. The DNS providers can balance their traffic loads based on locale. With the above-described minimal additions, the Domain Connect specification itself becomes locale-aware.

As mentioned earlier, the techniques are backward compatible. If the DNS provider is unable to support UIs specific to the locale passed by the service provider, it can safely ignore the parameter, so that the flow of the entire spec remains unchanged from its current form.

For DNS providers that don't currently have, but aspire to offer, locale support to their users - the techniques offer a clear pathway to improved services via incremental development. DNS providers who are already locale-aware can use the techniques to deliver an improved response - specifically perform load balancing, render locale-specific UIs. DNS providers can optionally implement the described features in addition to the regular implementation of the Domain Connect specification.

## CONCLUSION

The disclosure describes techniques that preserve the locale/language as the user is passed along from a service provider to a DNS provider. Per the techniques, the login and authorization screen of the DNS provider is in the locale/language detected by the service provider. The techniques enable a locale-specific user experience and an efficient, load-balancing of traffic to the DNS Provider servers.

## REFERENCES

- [1] <https://github.com/Domain-Connect/spec/blob/master/Domain%20Connect%20Spec%20Draft.adoc> accessed on Aug. 9, 2021.
- [2] <https://github.com/Domain-Connect/spec/blob/master/Domain%20Connect%20Spec%20Draft.adoc#dns-provider-discovery> accessed on Aug. 9, 2021.
- [3] <https://github.com/Domain-Connect/Templates> accessed on Aug. 9, 2021.
- [4] <https://github.com/Domain-Connect/spec> accessed on Aug. 9, 2021.
- [5] [https://en.wikipedia.org/wiki/Locale\\_\(computer\\_software\)](https://en.wikipedia.org/wiki/Locale_(computer_software)) accessed on Aug. 9, 2021.
- [6] Goel, Navneet and Gopalakrishnan, Aditya, “Improved Security and Reliability in the DNS Provider Discovery Mechanism of Domain Connect Protocol,” Technical Disclosure Commons, (June 18, 2021) [https://www.tdcommons.org/dpubs\\_series/4390](https://www.tdcommons.org/dpubs_series/4390)