



Avoiding Vendor Lock-in As An ESP Using An Email API Service

ESPs instruct their customers to create various DNS records when onboarding a domain to the ESP platform. These records are ones that are typical for email, such as MX records and records associated with email authentication (i.e., SPF, DKIM, and DMARC records).

In many cases, an ESP does not manage its own delivery infrastructure, but is instead a customer of another backend vendor that has a sending infrastructure, usually cloud-based. This creates a multi-layered customer relationship that looks like this:



Because the backend vendor is responsible for the actual sending and receiving of mail for the ESP's customers, some or all of the DNS records that the ESP's customers create will ultimately point to the backend vendor's infrastructure. How the ESP manages this is key to avoiding what GreenArrow calls "Vendor Lock-in".



An example of Vendor Lock-in using the domain names above might look like this:

Name	Type	Value
m.brand.example	MX	mx.a.backend.example
m.brand.example	MX	mx.b.backend.example
m.brand.example	TXT (SPF)	v=spf1 include:spf.backend.example ~all
esp._domainkey.m.brand.example	TXT (DKIM)	"v=DKIM1; k=rsa; p=MIGfMA0G[...]QIDAQAB"

We see here in the MX and SPF records an example of Vendor Lock-in. The ESP instructs its customer to create records that directly reference its current backend vendor for sending and receiving mail (the `backend.example` hostnames). Let's talk about why this isn't ideal and what can be done to make it better.

The Downsides of Vendor Lock-in

Vendor Lock-in creates the following issues for the ESP and its customers.

It Muddies The Waters For Your Customers

Your customer has entered into a relationship with you, their ESP. When you instruct your customer to create DNS records that don't obviously point to your domain, you run the risk of confusing your customer or, worse, eroding their trust in you. Your customers will have varied levels of technical savvy and know-how, and DNS records that mention a third party may lead to questions, such as "Who is `backend.example` and why are they sending my mail? I thought you were?"

These questions are best avoided, and that means avoiding Vendor Lock-in.

greenarrowemail.com



It Introduces Friction To The Migration Process

Should you, the ESP, ever want to switch to a new backend vendor, such as [GreenArrow's CloudMTA Service](#), Vendor Lock-in will mean that management of the migration process is not solely in your hands.

Migrating to a new platform will require planning on your part. You cannot simply move all your customers to a new platform in one fell swoop; instead, you will have to take into consideration sending volumes, warming up new IPs with the new backend vendor, etc.

Vendor Lock-in will mean additional challenges in the process, because your customers will have to make DNS changes. For your technically savvy customers, you'll be at the mercy of their change management processes, and have to take those into account as part of your migration planning. For those customers who aren't technically savvy, such as small business owners that don't know who manages their DNS or how to update it, their migration could be delayed in ways that cost you money as you continue to support them on a backend platform for much longer than you originally planned.



Avoiding Vendor Lock-in Starts With DNS

It is impossible to onboard a customer without requiring them to configure DNS for their domain(s). The following table lists the domains involved and the DNS record types that are usually created:

Domain	DNS Record Types
RFC5321.MailFrom Domain (dedicated bounce domain)	TXT (SPF), MX
RFC5322.From Domain	DKIM public key, optional MX record, optional DMARC record

In the best-case scenario, the customer will only be asked to configure DNS during their onboarding and never again. To provide this ideal customer experience and avoid Vendor Lock-in, the records created during onboarding should reference records controlled solely by you, the ESP.

Note: Industry best practice for ESP customers is to use a dedicated RFC5321.MailFrom, or Return-Path, domain for sending from the ESP's platform. Doing so ensures that asynchronous bounces, complaints, and other feedback-type messages route back to the ESP platform for ease of suppression list management and such. The RFC5322.From domain, on the other hand, may or may not be one that's dedicated to the ESP sending platform. DKIM signing with a [domain that aligns with the RFC5322.From domain](#) should be done on the ESP platform, but whether or not the ESP platform also hosts inbound mail for this domain will depend on the customer and the ESP. The rest of this document assumes that the ESP platform does **not** host mail for the RFC5322.From domain.

Let's look at the ways to avoid Vendor Lock-in with DNS.



DNS Delegation

So long as your customer is using dedicated domains on your platform, one method for avoiding Vendor Lock-in is to have your customers delegate DNS management for their domains to you. Not all customers will agree to this, but if they do, and if you've got the resources to manage it, it's an effective way to avoid Vendor Lock-in.

Delegating DNS management to you would first require your customer to create NS records and usually a TXT record, like in this example:

Name	Type	Value
m.brand.example	NS	ns1.esp.example
m.brand.example	NS	ns2.esp.example
brand.example	TXT	esp-site-verification=84df32wesdeop9e

Here, ns1.esp.example and ns2.esp.example are hostnames for hypothetical DNS servers that you, the ESP, manage.

The TXT record serves as a "domain verification" record. This record, which contains content that you specify, proves to you that your customer has control over DNS for the domain, or at least the ability to get DNS records created in the brand's apex domain.

Once you've verified that this domain verification record exists, you can then configure DNS for your customer's domain on your servers. For each customer, you would create the necessary records, such as MX, SPF, DKIM public keys, etc., to support their use of their domain (m.brand.example, in this case) on your platform.



In this scenario you would have full control over the DNS records for the customer domain, and you would avoid the downsides of Vendor Lock-in.

It is also worth noting that this domain verification record closes a possible abuse vector. The vector is exposed when one user creates an account with the ESP, sets up a From-header domain, creates the DNS records, and verifies the domain. Later, that user cancels the account with the ESP but does not delete the NS records that delegate the domain to the ESP. Someone else, an attacker rather than the real owner, adds the same From-header domain to their own account with the ESP. The NS records exist, so the domain exists, and without the domain verification record, the attacker can now send abusive email from it.

The idea that a domain verification record will block this abuse vector is predicated on the following assumptions being true:

- The original customer terminating with the ESP means that the ESP “forgets” the domain verification code that was issued the first time the domain was added to the ESP platform, meaning it’s no longer a valid verification code.
- The ESP’s onboarding process requires generation of a new, unique domain verification code when a domain is added to any customer’s account.

If these two assumptions are true, then a leftover record from a former customer is of no use to an imposter, and because the imposter cannot publish a code to the brand’s apex domain, the imposter cannot add the domain to their account and abuse the brand.



The Intermediary CNAME Records

While DNS Delegation can be a clean way to avoid Vendor Lock-in, many domain owners will not agree to directly ceding control of a portion of their namespace to a third party. Thus, a solution with less friction to implement involves the customer creating one or more CNAMEs instead. Let's examine that path now...

Recall the earlier table showing typical DNS records created by an ESP customer:

Name	Type	Value
m.brand.example	MX	mx.a.backend.example
m.brand.example	MX	mx.b.backend.example
m.brand.example	TXT (SPF)	v=spf1 include:spf.backend.example ~all
esp._domainkey.m.brand.example	TXT (DKIM)	"v=DKIM1; k=rsa; p=MIGfMA0G[...]QIDAQAB"

With this configuration, should you wish to migrate to another backend vendor, or even should you wish to rotate a customer's DKIM signing key, your customer will have to update their DNS.



Intermediary CNAME Records avoid both this complexity and Vendor Lock-in. In this scenario, your customer would be instructed to create these CNAME records:

Name	Type	Value
m.brand.example	CNAME	customerID.dns.esp.example
bounce2.m.brand.example	CNAME	bounce2.customerID.dns.esp.example
esp-a._domainkey.m.brand.example	CNAME	esp-a._domainkey.customerID.dns.esp.example
esp-b._domainkey.m.brand.example	CNAME	esp-b._domainkey.customerID.dns.esp.example
esp-c._domainkey.m.brand.example	CNAME	esp-c._domainkey.customerID.dns.esp.example



And then you, the ESP, would create a series of records such as these:

Name	Type	Value
<code>customerID.dns.esp.example</code>	MX	<code>mx.backend.example</code>
<code>customerID.dns.esp.example</code>	MX	<code>mxb.backend.example</code>
<code>customerID.dns.esp.example</code>	TXT (SPF)	<code>v=spf1 include:spf.backend.example ~all</code>
<code>bounce2.customerID.dns.esp.example</code>	TXT	<code>""</code>
<code>esp-a._domainkey.customerID.dns.esp.example</code>	TXT (DKIM)	<code>"v=DKIM1; k=rsa; p=MIGfMA0G[...]QIDAQAB"</code>
<code>esp-b._domainkey.customerID.dns.esp.example</code>	TXT (DKIM)	<code>"v=DKIM1; p="</code>
<code>esp-c._domainkey.customerID.dns.esp.example</code>	TXT (DKIM)	<code>"v=DKIM1; p="</code>

The CNAME records provide the following functionality:

- SPF and MX records for the RFC5321.MailFrom Domain your customer will use on your existing sending platform, which you maintain with the target `customerID.dns.esp.example` records.
- DKIM signing using a d= domain that aligns with your customer's RFC5322.From Domain
- Ease of DKIM key rotation and platform migration should it be necessary

The `bounce2` record is only a placeholder for now, intended for use during a possible future platform migration. We've also added two placeholder DKIM keys (`esp-b` and `esp-c`) to facilitate DKIM key rotation and platform migration, which we'll get to later. Our expectation is that only one DKIM key per sending platform will be in use at any given time.



With this configuration, you as the ESP avoid Vendor Lock-in and its downsides. Not only does your customer never have to touch their DNS again for as long as they are your customer, you now have the ability to migrate them to a new backend vendor and/or rotate their DKIM signing keys independently of all your other customers should the need arise.

DKIM Key Archiving and Wildcard DNS

A third approach to avoiding vendor lock-in is one that relies on Wildcard DNS and what we call DKIM Key Archiving. This approach relies on separate solutions to the two lock-in anchors, namely the DKIM Key and the SPF and MX records associated with the RFC5321.MailFrom domain. While we believe this approach to be inferior to the Intermediary CNAME Records approach because it makes DKIM key rotation more difficult, it is nonetheless one that requires less software development than the other two, and it may be the solution an ESP can implement fastest among those presented here.

DKIM Key Archiving

It is typical for the backend sending platform to generate the DKIM private/public key pair for any signing domain. The public key is presented to the party adding the domain to the backend platform so that it can be published in DNS, and the private key is stored on the backend platform so it can be used in signing outgoing messages. Lock-in is created when the backend platform does not allow exporting the private key.

An alternative to this as a way to avoid vendor lock-in is for the ESP to generate the DKIM key pair at the time of customer onboarding, archive it locally, and upload the pair to the backend sending platform as needed. Most backend platforms allow importing of DKIM private keys.



Generating the key pair is most easily done on a Unix platform using the `openssl` command; the following example shows how to generate a 1024-bit private/public key pair:

Shell

```
openssl genrsa -out dkim_private.pem 1024  
openssl rsa -in dkim_private.pem -pubout -out dkim_public.pem
```

The above two commands will store the private key in the file `dkim_private.pem` and the public key in the file `dkim_public.pem`. The `dkim_public.pem` file contains the value of the `p=` key-value pair in the DKIM public key record.

The method for uploading the private/public key pair to the backend platform will be platform-dependent, and is beyond the scope of this document.



Wildcard DNS for SPF and MX Records

With the wildcard DNS approach, SPF and MX records for all brands will be identical, and this will be accomplished through the use of CNAME records, just as in the Intermediary CNAME Records approach. What's different here is the structure of the records that the ESP creates to accomplish this. These records will look like this:

Name	Type	Value
*.root.dns.esp.example	MX	mx.a.backend.example
*.root.dns.esp.example	MX	mx.b.backend.example
*.root.dns.esp.example	TXT (SPF)	v=spf1 include:spf.backend.example ~all
*.bounce2.dns.esp.example	TXT	""

The brand's DNS records will also look a bit different, as the identifier (`customerID`) will move to the left edge of the DNS record to take advantage of the wildcard:

Name	Type	Value
m.brand.example	CNAME	customerID.root.dns.esp.example
bounce2.m.brand.example	CNAME	customerID.bounce2.dns.esp.example



And finally, the brand will publish a DKIM key record without the use of a CNAME:

Name	Type	Value
esp._domainkey.m.brand.example	TXT (DKIM)	"v=DKIM1; k=rsa; p=MIGfMA0G[...]QIDAQAB"

Having the customer directly publish the DKIM public key in DNS does not create lock-in, because the ESP has archived a copy of the DKIM private key, and can import that key into a different backend sending platform.

Other Methods

The above are the most common and easily implemented methods for avoiding Vendor Lock-in through DNS, and involve minimal effort on the part of the brand; they have to configure DNS once, and then they're done.

More complex customer configurations may require different methods, but the goal for any method implemented must be the same: ensure the brand is only interfacing with you, the ESP, and is only creating DNS records that reference your domain, not the domain of the backend vendor.



How To Migrate While Avoiding Vendor Lock-In

The DNS configurations discussed above put the framework in place to allow the ESP to avoid Vendor Lock-in. The details of the ESP's relationship with its backend vendor are hidden from customers, and the ESP is set up for success in migrating its customer brands to a new backend platform should it choose to do so. Actually accomplishing the migration still presents some challenges for the ESP if it wants to avoid any customer DNS changes, but if done well, the migration can be seamless and essentially invisible to the customers. GreenArrow provides managed services to help our customers achieve this seamless migration.

Let's look at what it takes to do it well.

Necessities for Migration

Migrating a customer from one platform to another will require the following:

- Activating the placeholder RFC5321.MailFrom domain (what we called `bounce2` above)
- Updating the SPF record for the original RFC5321.MailFrom domain (what we called `m.brand.example`)
- Continuity of operations for DKIM signing
- Updates to the MX and TXT records for the RFC5321.MailFrom domain.

Unless all ESP customers are low-volume senders, or the existing backend vendor suddenly terminates operation, migration is likely to be a phased process rather than simply “flipping the switch”. Even with proper email authentication in place from the first message sent from the new platform, it will still be necessary to “warm up” the new platform for each customer while winding down the old platform. This will make the biggest challenge here the management of the MX records for the customer's domain(s).



Activating the Placeholder RFC5321.MailFrom Domain

For ESP customers, the RFC5321.MailFrom domain found in the Return-Path header is the one to which bounces, complaints, unsubscribes, and other feedback-type messages are directed. The left-hand side, or “localpart” of the address usually contains encoded information about the recipient of the message that generated the feedback, and a proper sending platform ingests these feedback messages and updates subscriber records and mailing lists accordingly.

In a perfect world, an ESP customer would use the same RFC5321.MailFrom domain on multiple platforms; however, each platform differs in its methods for encoding and decoding recipient messages, making such a proposition difficult to implement. To achieve this, an ESP would have to deploy a message handling layer in front of its sending platforms, and that layer would have to parse the destination address of the feedback message, determine which platform originated the original message, and then route the feedback message to the correct platform. It’s not impossible to do this, but it is certainly resource-intensive.

Because you’ve had your customer create a placeholder RFC5321.MailFrom domain during the onboarding process, you can avoid this work and simply update a DNS record that you already control. For example, if you’re migrating to the GreenArrow platform, you’d update this record:

Name	Type	Value
bounce2.customerID.dns.esp.example	TXT	" "



To look like this:

Name	Type	Value
bounce2.customerID.dns.esp.example	TXT (SPF)	v=spf1 include:spf.drhinternet.net ~all
bounce2.customerID.dns.esp.example	MX	mx.drhinternet.net

The above two records provide an SPF and MX record for your customer's bounce2 domain, which can be used as the RFC5321.MailFrom domain for mail sent through the new backend platform.

Updating SPF for Existing RFC5321.MailFrom Domain

Although the existing RFC5321.MailFrom domain is not likely to be used on the new backend platform, updating its SPF record during the migration process should be done as a safeguard against any accidental traffic sent through the new platform.

Updating SPF for the existing RFC5321.MailFrom domain while avoiding vendor lock-in is straightforward for the ESP. During the migration period, the SPF record for the original RFC5321.MailFrom domain can include both the legacy vendor and the new one, and then once the migration is done, the legacy vendor can be removed. It is not uncommon for SPF records to have multiple "include" directives, and so long as adding a new include directive doesn't force the SPF record to exceed the ten lookup limit, there should be no problem.



During migration, the SPF record would be configured to look like this:

Name	Type	Value
<code>customerID.dns.esp.example</code>	TXT (SPF)	<code>v=spf1 include:spf.drhinternet.net include:spf.backend.example ~all</code>

The MX records for this domain will be left as is during the migration.

Migrating DKIM for Continuity of Operations

The creation of three DKIM key records during customer onboarding provides ample flexibility during migration to make the process seamless for the customer. There are two primary methods available for ensuring no disruption of DKIM signing during and after the migration.

DKIM Key Rotation Method

Assuming that the ESP is following best practices for rotating keys, such as those published by [M3AAWG](#), at the time migration begins, the customer's DKIM public key records as published by the ESP might look something like this:

Name	Type	Value
<code>esp-a._domainkey.customerID.dns.esp.example</code>	TXT (DKIM)	<code>"v=DKIM1; p="</code>
<code>esp-b._domainkey.customerID.dns.esp.example</code>	TXT (DKIM)	<code>"v=DKIM1; k=rsa; p=MIGfMA0G[...]QIDAQAB"</code>
<code>esp-c._domainkey.customerID.dns.esp.example</code>	TXT (DKIM)	<code>"v=DKIM1; p="</code>



Meanwhile, the customer's CNAME records are still in place:

Name	Type	Value
esp-a._domainkey.m.brand.example	CNAME	esp-a._domainkey.customerID.dns.esp.example
esp-b._domainkey.m.brand.example	CNAME	esp-b._domainkey.customerID.dns.esp.example
esp-c._domainkey.m.brand.example	CNAME	esp-c._domainkey.customerID.dns.esp.example

What we see above is the following:

- The DKIM key with selector `esp-a` has been expired as part of regular key rotation
- The DKIM key with selector `esp-b` is currently in use as the DKIM signing key on the legacy vendor platform
- The DKIM key with selector `esp-c` has not yet been published (or has been expired as part of regular key rotation)

Migrating to the new platform would involve the following steps:

- Configure the new platform to DKIM sign the customer's mail using `esp-c` as the selector and `m.brand.example` as the signing domain
- Publish the new public key at `esp-c._domainkey.customerID.dns.esp.example`
- Start the process of migrating the customer to the new platform
- At least seven and up to thirty days after the migration is finished (per M3AAWG guidelines), expire the key at `esp-b._domainkey.customerID.dns.esp.example`.



DKIM Key Archive and Import Method

If the ESP is generating and archiving DKIM keys for its customers, then the alternative to the DKIM Key Rotation Method for avoiding Vendor Lock-in is to upload the stored keys for the customer to the new platform, assuming that the new platform supports importing DKIM keys. This will allow for the customer's mail traffic to be signed with the same DKIM selector and domain when sending from both platforms.

The process for accomplishing this is straightforward:

- Import the archived private/public key pair into the new platform
- Test to make sure that messages sent from the new platform still pass DKIM validation
- Start the migration of the customer's mail traffic to the new platform



Update DNS for the Legacy RFC5321.MailFrom Domain

The last DNS update for an ESP to make when migrating a customer to a new sending platform while avoiding Vendor Lock-in is the RFC5321.MailFrom Domain used on the legacy sending platform. This record should only be changed after the customer has migrated to the new platform, and that change should take place at least fourteen days after the migration finished, so as to allow for any late-arriving feedback messages to still be properly processed.

The updated record should look like this:

Name	Type	Value
<code>customerID.dns.esp.example</code>	TXT	<code>" "</code>

The previous MX records for the domain will be deleted, and this will make `bounce2.m.brand.example` the new placeholder RFC5321.MailFrom domain for the customer, which will allow for continued avoidance of Vendor lock-in for this customer going forward.

Conclusion

This document provides a blueprint for an ESP to use DNS records to avoid Vendor Lock-in for its sending platform. By following these guidelines, an ESP will avoid customer confusion and will have the flexibility to move customers to a new sending platform with minimal disruption for its customers.