Apple Push Notification Service

Apple Push Notification service (APNs) is the centerpiece of the remote notifications feature. It is a robust and highly efficient service for propagating information to iOS (and, indirectly, watchOS), tvOS, and OS X devices. Each device establishes an accredited and encrypted IP connection with APNs and receives notifications over this persistent connection. If a notification for an app arrives when that app is not running, the device alerts the user that the app has data waiting for it.

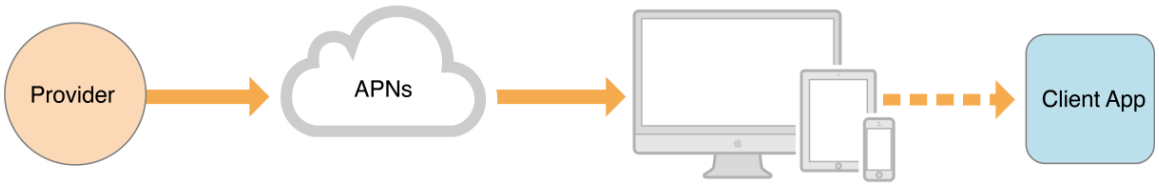
You provide your own server to generate the remote notifications for your users. This server, known as the *provider*, gathers data for your users and decides when a notification needs to be sent. For each notification, the provider generates the notification payload and attaches that payload to an HTTP/2 request, which it then sends to APNs using a persistent and secure channel using the HTTP/2 multiplex protocol. Upon receipt of your request, APNs handles the delivery of your notification payload to your app on the user's device.

For information about the format of the requests that you send to APNs, and the responses and errors you can receive, see APNs Provider API. For information about how to implement notification support in your app, see Registering, Scheduling, and Handling User Notifications.

The Path of a Remote Notification

Apple Push Notification service transports and routes remote notifications for your apps from your provider to each user's device. Figure 3-1 shows the path each notification takes. When your provider determines that a notification is needed, you send the notification and a device token to the APNs servers. The APNs servers handle the routing of that notification to the correct user device, and the operating system handles the deliver of the notification to your client app.

Figure 3-1 Pushing a remote notification from a provider to a client app

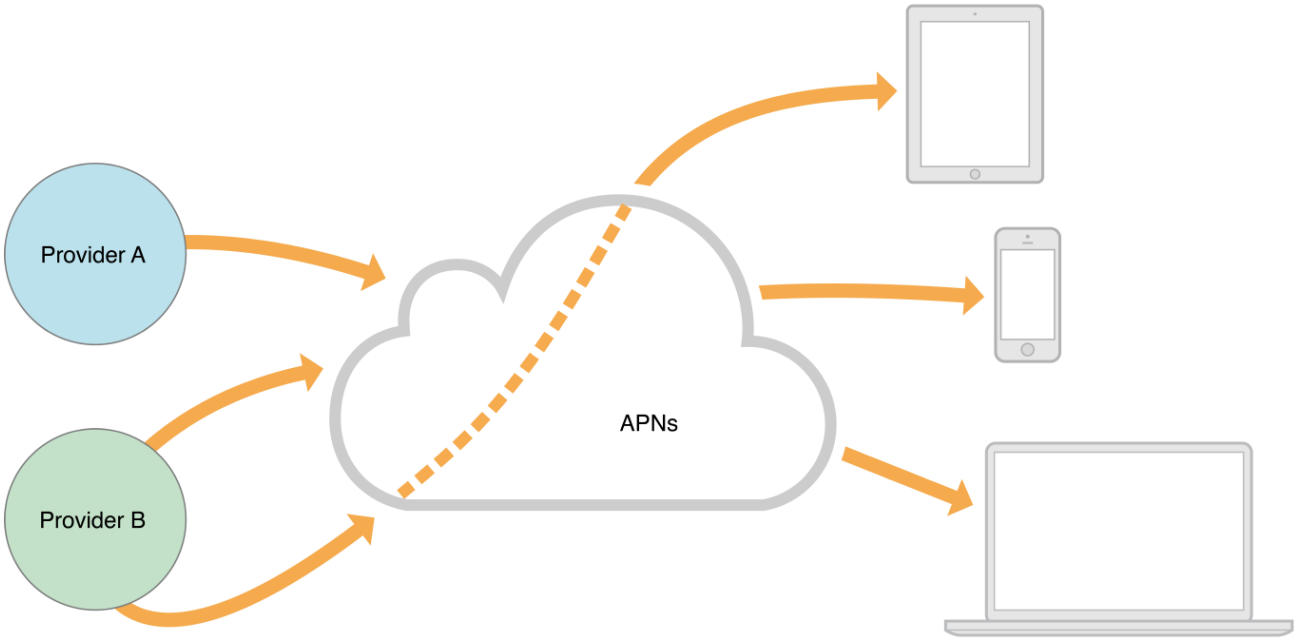


The device token you provide to the server is analogous to a phone number; it contains information that enables APNs to locate the device on which your client app is installed. APNs also uses it to authenticate the routing of a notification. The device token is provided to you by your client app, which receives the token after registering itself with the remote notification service.

The notification payload is a JSON dictionary containing the data you want sent to the device. The payload contains information about how you want to notify the user, such as using an alert, badge or sound. It can also contain custom data that you define.

Figure 3-2 shows a more realistic depiction of the virtual network APNs makes possible among providers and devices. The device-facing and provider-facing sides of APNs both have multiple points of connection; on the provider-facing side, these are called gateways. There are typically multiple providers, each making one or more persistent and secure connections with APNs through these gateways. And these providers are sending notifications through APNs to many devices on which their client apps are installed.

Figure 3-2 Pushing remote notifications from multiple providers to multiple devices



For information about getting the device token, see Token Generation and Dispersal. For information about the notification payload, see The Remote Notification Payload.

Quality of Service

Apple Push Notification service includes a default Quality of Service (QoS) component that performs a store-and-forward function. If APNs attempts to deliver a notification but the device is offline, the notification is stored for a limited period of time, and delivered to the device when it becomes available. Only one recent notification for a particular app is stored. If multiple notifications are sent while the device is offline, the new notification causes the prior notification to be discarded. This behavior of keeping only the newest notification is referred to as *coalescing* notifications.

If the device remains offline for a long time, any notifications that were being stored for it are discarded.

Security Architecture

To ensure secure communication, APNs regulates the entry points between providers and devices using two different levels of trust: connection trust and token trust.

Connection trust establishes certainty that APNs is connected to an authorized provider for whom Apple has agreed to deliver notifications. APNs also uses connection trust with the device to ensure the legitimacy of that device. Connection trust with the device is handled automatically by APNs but you must take steps to ensure connection trust exists between your provider and APNs.

Token trust ensures that notifications are routed only between legitimate start and end points. Token trust involves the use of a device token, which is an opaque identifier assigned to a specific app on a specific device. Each app instance receives its unique token when it registers with APNs and must share this token with its provider. Thereafter, the token must accompany each notification sent by your provider. Providing the token ensures that the notification is delivered only to the app/device combination for which it is intended.

> **Note:** A device token is not a unique ID that you can use to identify a device. Device tokens can change after updating the operating system on a device. As a result, apps should send their device token

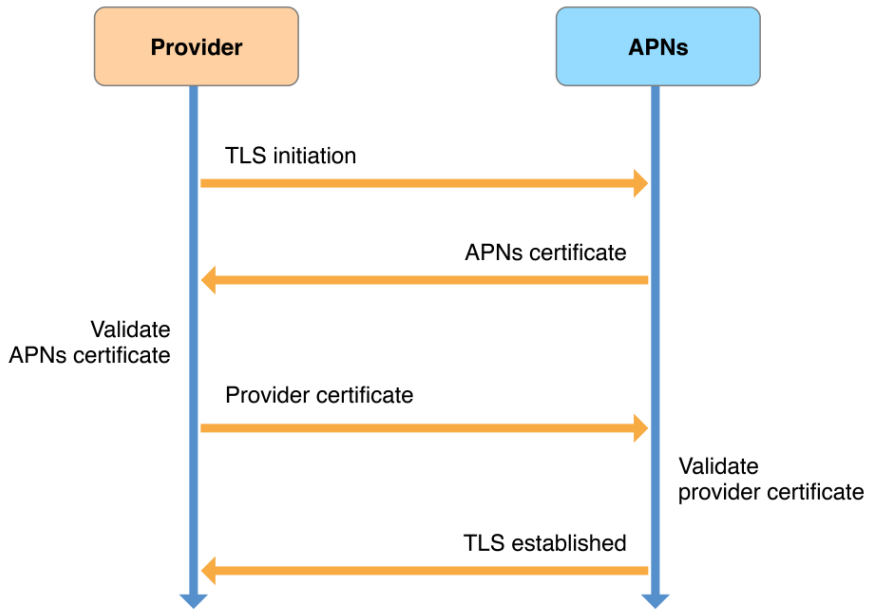
APNs servers also have the necessary certificates, CA certificates, and cryptographic keys (private and public) for validating connections and the identities of providers and devices.

Provider-to-APNs Connection Trust

Each provider must have a unique provider certificate and private cryptographic key, which are used to validate the provider's connection with APNs. The provider certificate (which is provisioned by Apple) identifies the topics supported by the provider. (A topic is the bundle ID associated with one of your apps.)

Your provider establishes connection trust with APNs through TLS peer-to-peer authentication. After the TLS connection is initiated, you get the server certificate from APNs and validate that certificate on your end. Then you send your provider certificate to APNs, which validates that certificate on its end. After this procedure is complete, a secure TLS connection is established; APNs is now satisfied that the connection has been made by a legitimate provider.

Figure 3-3 Establishing connection trust between a provider and APNs



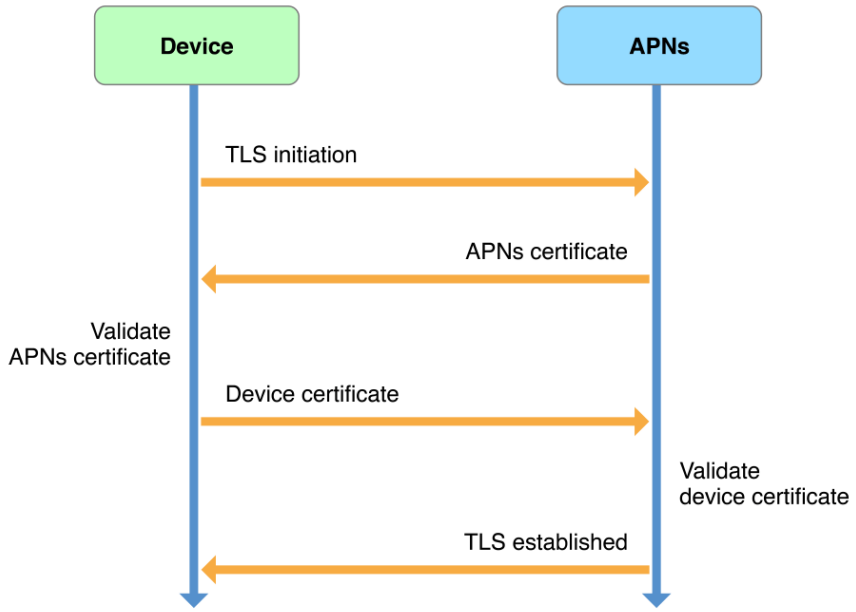
The HTTP/2 provider connection is valid for delivery to one specific app, identified by the topic (bundle ID) specified in the certificate. Depending on how you configure and provision your APNs SSL certificate, the connection can also be valid for delivery of remote notifications to any Apple Watch complications or backgrounded VoIP services associated with that primary app. See APNs Provider API for details. APNs also maintains a certificate revocation list; if a provider's certificate is on this list, APNs can revoke provider trust (that is, refuse the connection).

APNs-to-Device Connection Trust

Each device has a device certificate and private cryptographic key, which are used to validate the device's connection with APNs. The device obtains its certificate and key at device activation time and stores them in the keychain.

You do not have to do anything to establish connection trust between APNs and a device. APNs establishes the identity of a connecting device through TLS peer-to-peer authentication. In the course of this procedure, the device initiates a TLS connection with APNs, which returns its server certificate. The device validates this certificate and then sends its device certificate to APNs, which validates that certificate.

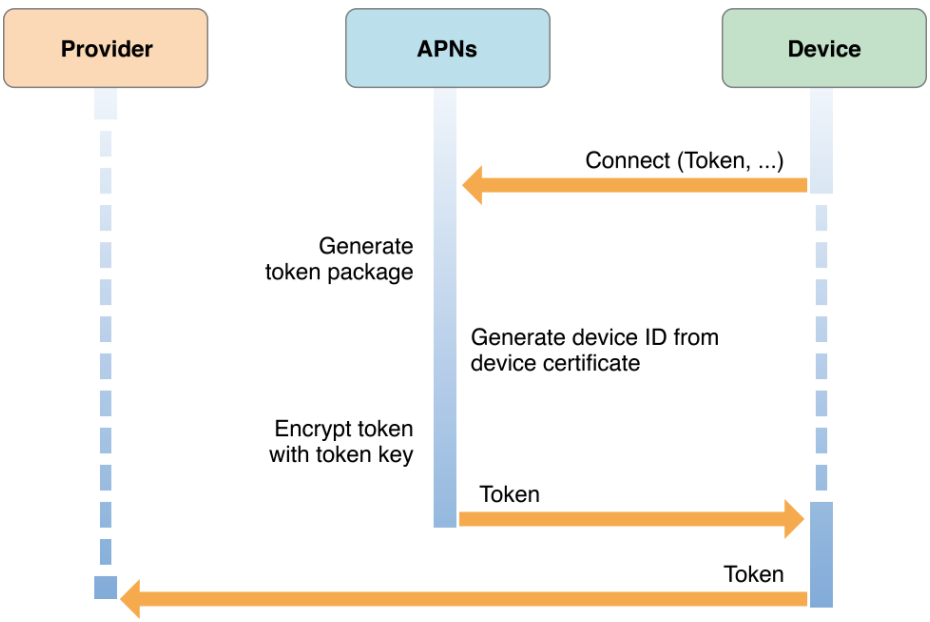
Figure 3-4 Establishing connection trust between a device and APNs



Token Generation and Dispersal

An app must register with the system to receive remote notifications, as described in Registering for Remote Notifications. Upon receiving a registration request, the system forwards the request to APNs, which generates a unique device token, for the app, using information contained in the device's certificate. It then encrypts the token using a token key and returns it to the device, as shown in Figure 3-5. The system delivers the device token to your app as an `NSData` object. Upon receiving this token, your app must forward it to your provider in either binary or hexadecimal format. Your provider cannot send notifications to the device without this token.

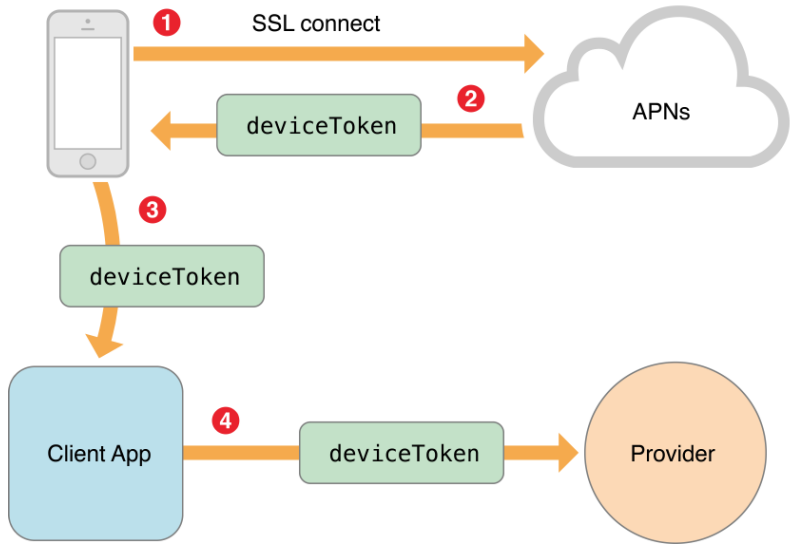
Figure 3-5 Managing the device token



Important: APNs device tokens are of variable length. Do not hardcode their size.

Figure 3-6 illustrates the token generation and dispersal sequence, but in addition shows the path of the token as it is returned by APNs and subsequently forwarded to your custom provider.

Figure 3-6 Sharing the device token



Token Trust (Notification)

Every notification that your provider sends to APNs must be accompanied by the device token associated of the device for which the notification is intended. APNs decrypts the token using its token key to ensure the validity of the notification source—that is, your provider. APNs uses the device ID contained in the device token to determine the identity of the target device. It then sends the notification to that device, as shown in Figure 3-7.

Figure 3-7 Identifying a device using the device token