

# Push Notifications

Setup, Remote, Silent, Local

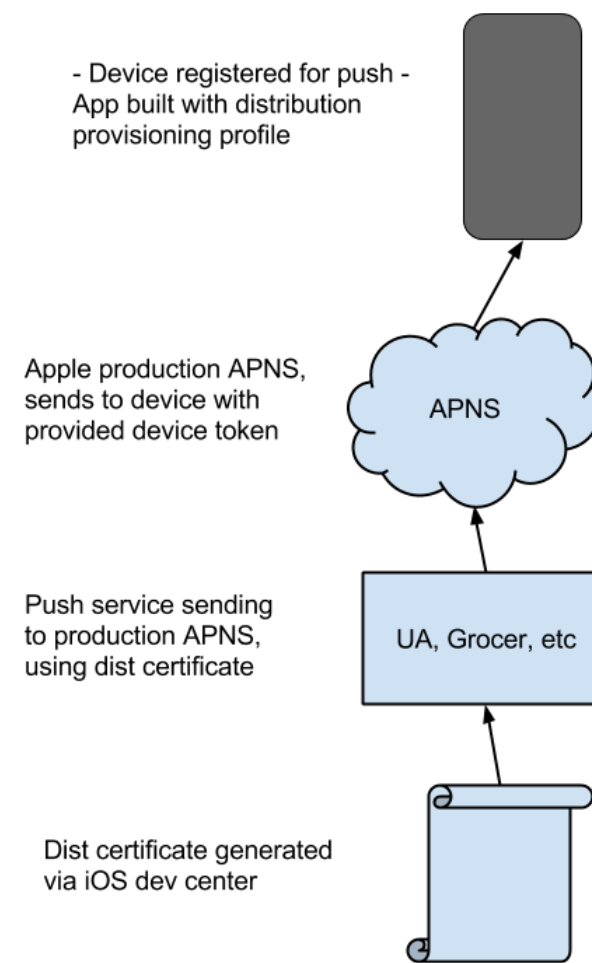
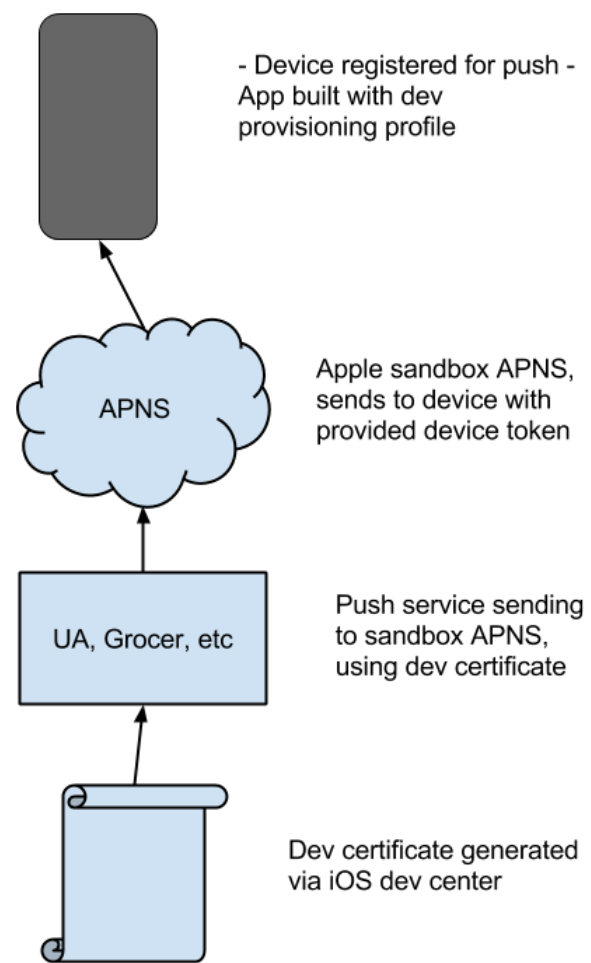
# Receiving Notifications

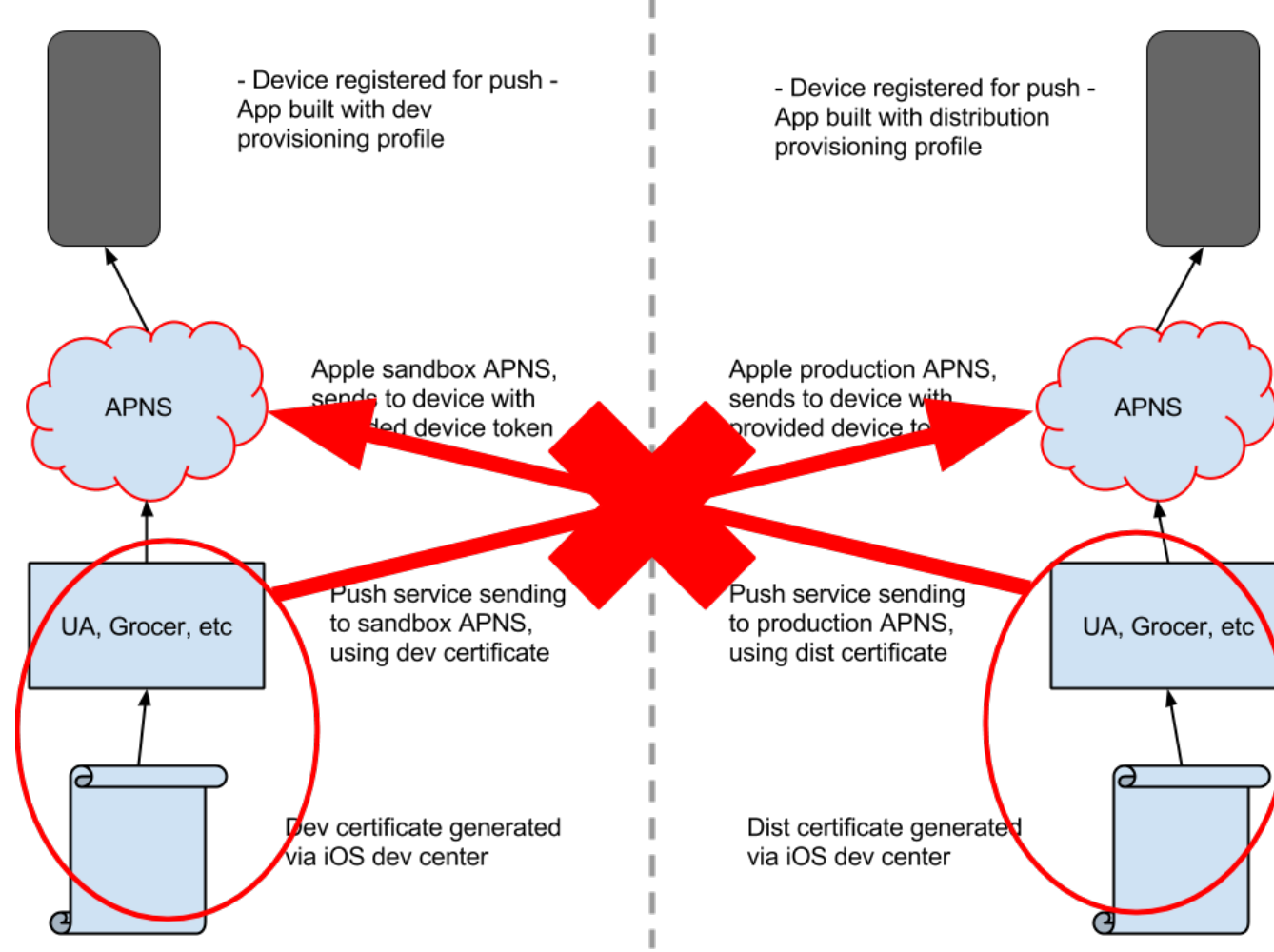
- Certificates through iOS Developer Centre
- Server or third-party service setup
- Register to receive
- Handling payload
- Extras

# Apple's Blessing

- Register app to use remote notifications — iOS Dev centre and project settings
- Generate certificates for the server — iOS Dev centre. Similar to process for provisioning certificates
- Development / Distribution — not interchangeable

Opinion: Use the same RSA key for the entire project.





Device token for development/sandbox will not work on production/distribution.

There are generally no errors thrown from Apple when one token is used for another server — this is why it's a common headache.

# Sender Service

- Popular choices include Urban Airship, Push.io, Parse
- Rails: Grocer, PHP: PHP-APNS, Python: apns, etc.
- Requires: certificates in previous steps.

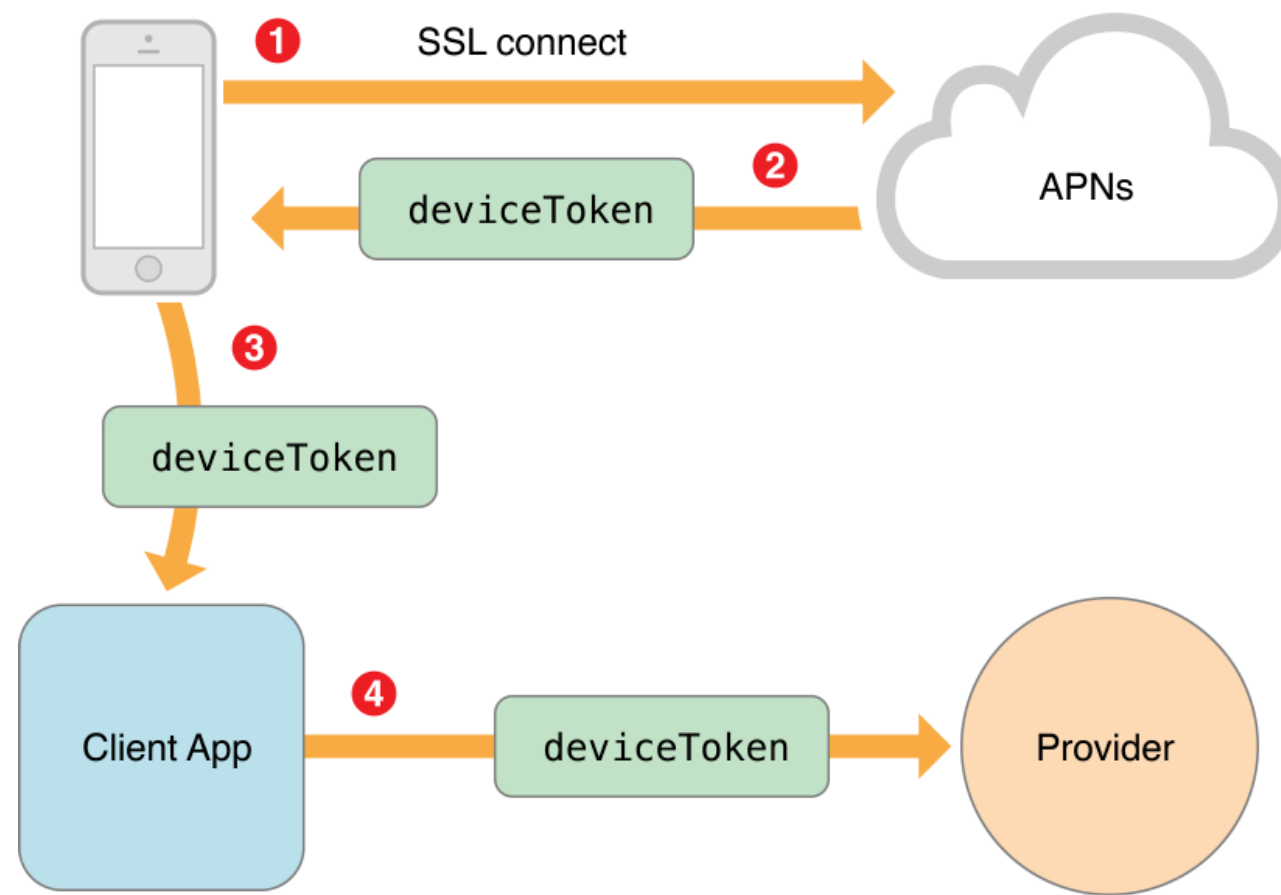
Build & show example of rails using grocer and using UA.

Make a brief note about the kinds of errors that can be received from APNS. apple feedback servers [feedback.\[sandbox\].push.apple.com](https://feedback.[sandbox].push.apple.com)

# Register to Receive

- On the device only — no simulator support
- iOS 7 background modes: remote notifications
- `[[UIApplication sharedApplication] registerForRemoteNotificationTypes:...]`
- Send device token to service within `application:didRegisterForRemoteNotificationWithDeviceToken:`
- Service uses device token to target device  
*eg. {device\_token, payload} => APNS*

- Note difference for dist/dev APNS again. Really important.
- `registerForRemoteNotificationTypes:` contacts APNS
  - can fail
- token is NSData, though a common method to get the string representation is through description and removing spaces/brackets... don't do that.





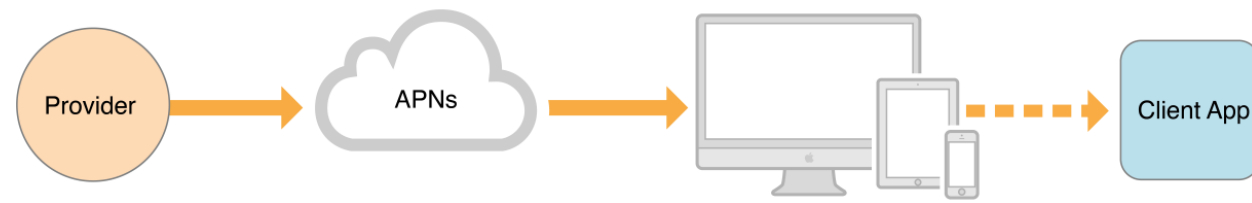
# Handle Payload

- Typical minimal payload:

```
{
  aps: {
    alert: "Hello iOS"
  }
}
```

- `alert` determines the text for alert banner or UIAlertView. Considered silent if no alert is present
- Customize freely, though must contain `aps`
- Limited to a total of 256 bytes

Uses are fairly limitless



# Handle Payload

- Three possible entry points for remote notification handling:  
`application:didFinishLaunchingWithOptions:`  
`application:didReceiveRemoteNotification:`  
`application:didReceiveRemoteNotification:fetchCompletionHandler:`
- Use supplied `UIApplication` and state to determine how user entered app
- Use the final signature if using the remote-notifications background mode only

# Handle Payload

- App badges are handled by Apple automatically
- Apple looks for `badge` key within `aps`

# Troubleshooting

- I never got the push notification!  
*Check to make sure you're sending and expecting to receive on the same path (development, or production).*
- It took forever to get the push notification!  
*Are you on dev? Dev servers are generally not load balanced or very fast. Depends on service being used.*
- I tried to send 3 billion notifications and Apple got mad at me!  
*Try batching them. Apple only wants to keep the connection open for multiple pushes for so long. Experiment with different amounts.*

# Silent Push

- Does not include `alert` key in APS dictionary
- Before iOS 7, only when app was open already
- Push a request to update the app's data *right now* instead of waiting for a background fetch
- Immediately push a purchase to a user's device
- Don't disrupt a user who is in the app already

# Local Notifications

- No APNS involved
- Create and schedule:

```
n = [UILocalNotification new];  
n.fireDate = someDate;  
[[UIApplication sharedApplication]  
scheduleLocalNotification:n];
```
- Cancel

```
[n cancel];  
[[UIApplication sharedApplication]  
cancelAllLocalNotifications]
```
- Calendar reminders, 'come back to the app', etc.

# Code

- iOS <https://github.com/pivotal-rebecca/ios-push-demo-ios>
- Rails <https://github.com/pivotal-rebecca/ios-push-demo-rails>
- UA Tutorial: <http://docs.urbanairship.com/build/ios.html>
- Push.io tutorial: <https://docs.push.io/iOS/Overview>