

REST is not enough

Using Push Notifications to better support your mobile clients

Juan Gomez

@_juandg

What this talk is about

- Brief introduction to push notifications
- Why you need to embrace them
- High level details on how to implement them in Python

What I'm not going to cover

- Implementation on the smartphone side.
- Lots of code.
 - Specially production ready code.
- Detailed topics.

**How many hours of your life have
you spent looking at this webpage?**

Delivered 



Delivered On:
Wednesday, 04/09/2014 at 6:37 P.M.

[Request Status Updates »](#)

Left At:
Front Door

[Proof of Delivery](#) 

What time will your package be delivered to your home?
Get **FREE** approximate Delivery Windows on most UPS packages.

Continue

[I am already a UPS My Choice Member](#)

▼ Additional Information

Shipped/Billed On: 04/08/2014
Type: Package
Weight: 1.00 lb

▼ Shipment Progress

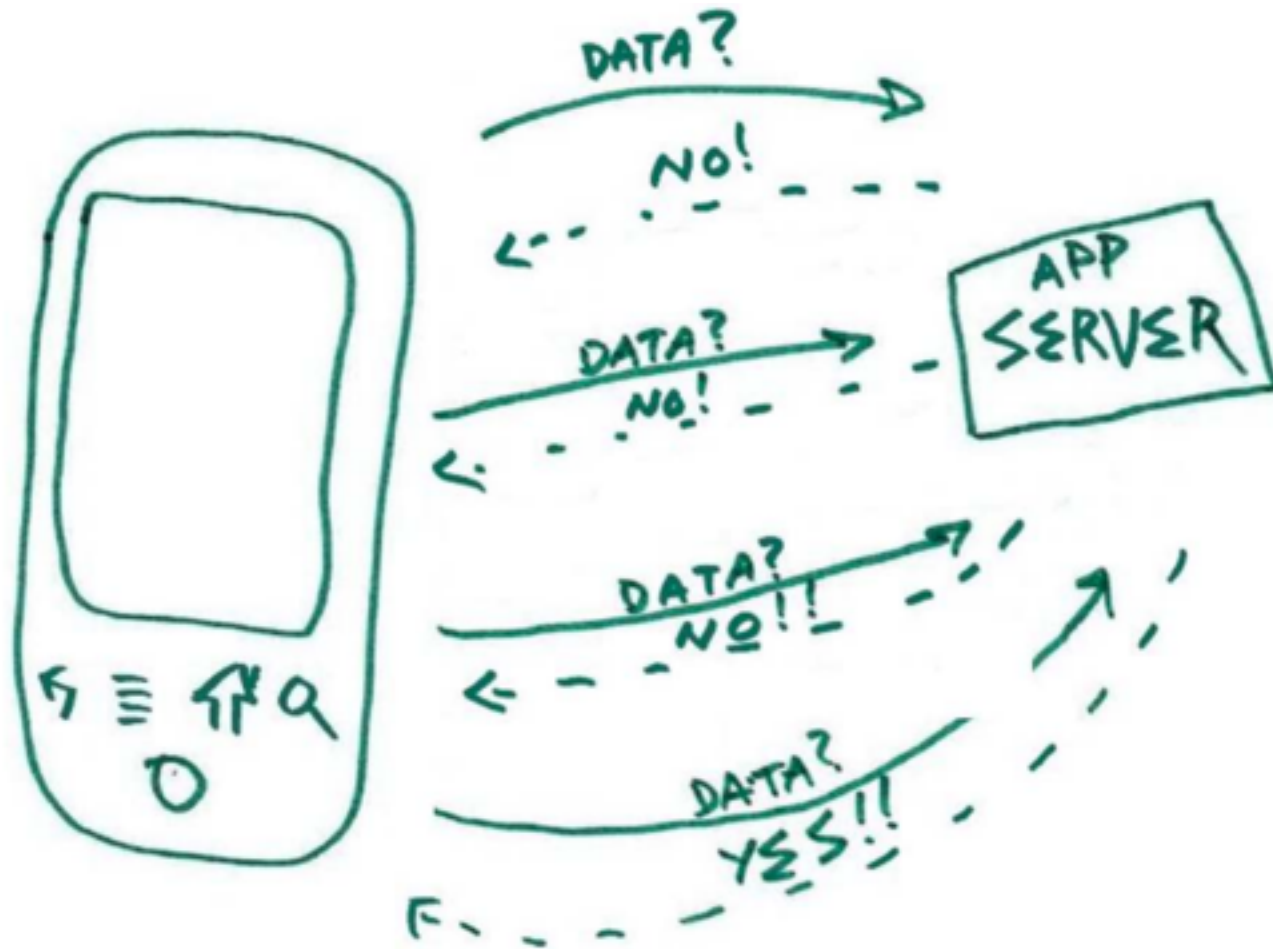
[What's This?](#) 

Location	Date	Local Time	Activity
San Francisco, CA, United States	04/09/2014	6:37 P.M.	Delivered
	04/09/2014	6:43 A.M.	Destination Scan
	04/09/2014	3:00 A.M.	Arrival Scan
Oakland, CA, United States	04/09/2014	2:00 A.M.	Departure Scan
Oakland, CA, United States	04/08/2014	5:19 P.M.	Arrival Scan
Louisville, KY, United States	04/08/2014	3:54 P.M.	Departure Scan
	04/08/2014	12:41 P.M.	Origin Scan
United States	04/08/2014	2:35 A.M.	Order Processed: Ready for UPS



**There has to be
a better way...**

Our current model





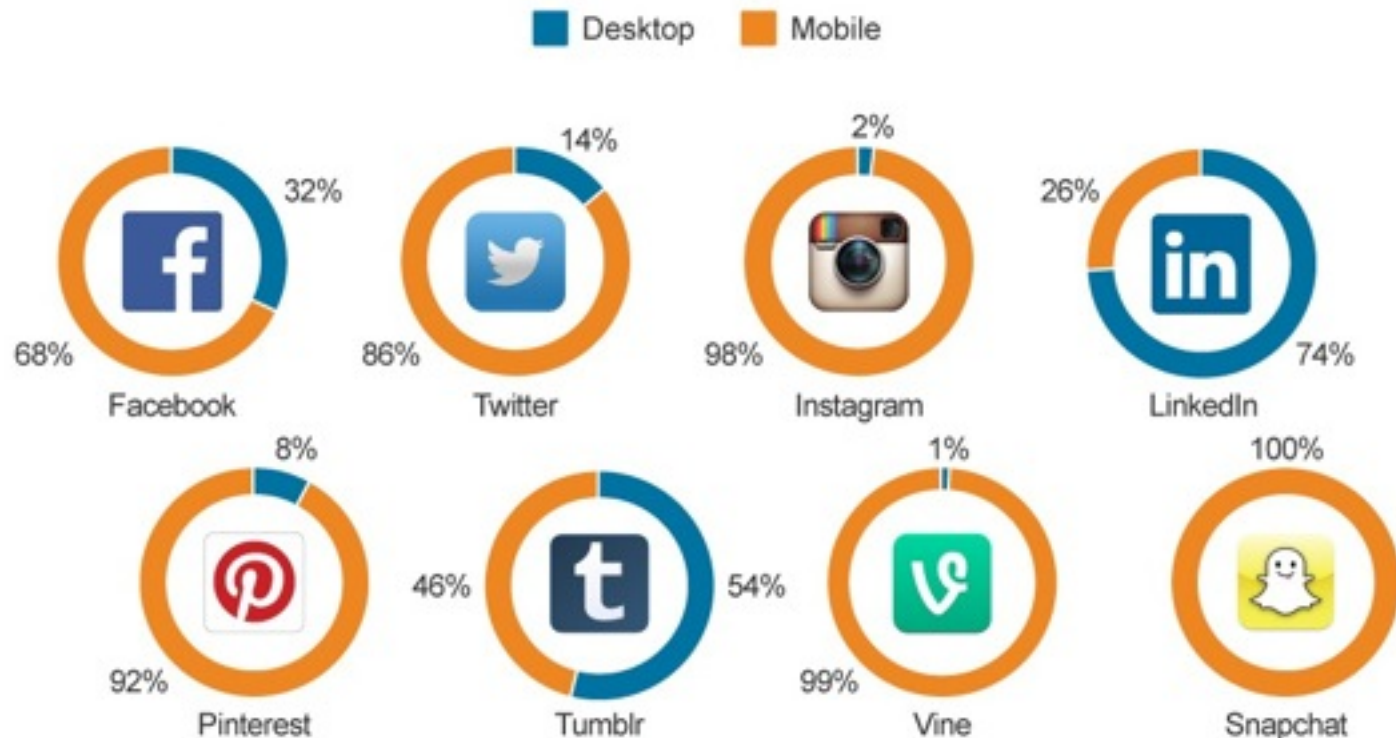
**Smartphone use around the world
isn't growing**

It's exploding

Smartphones are eating the world!

Most Social Networks Are Now Mobile-First

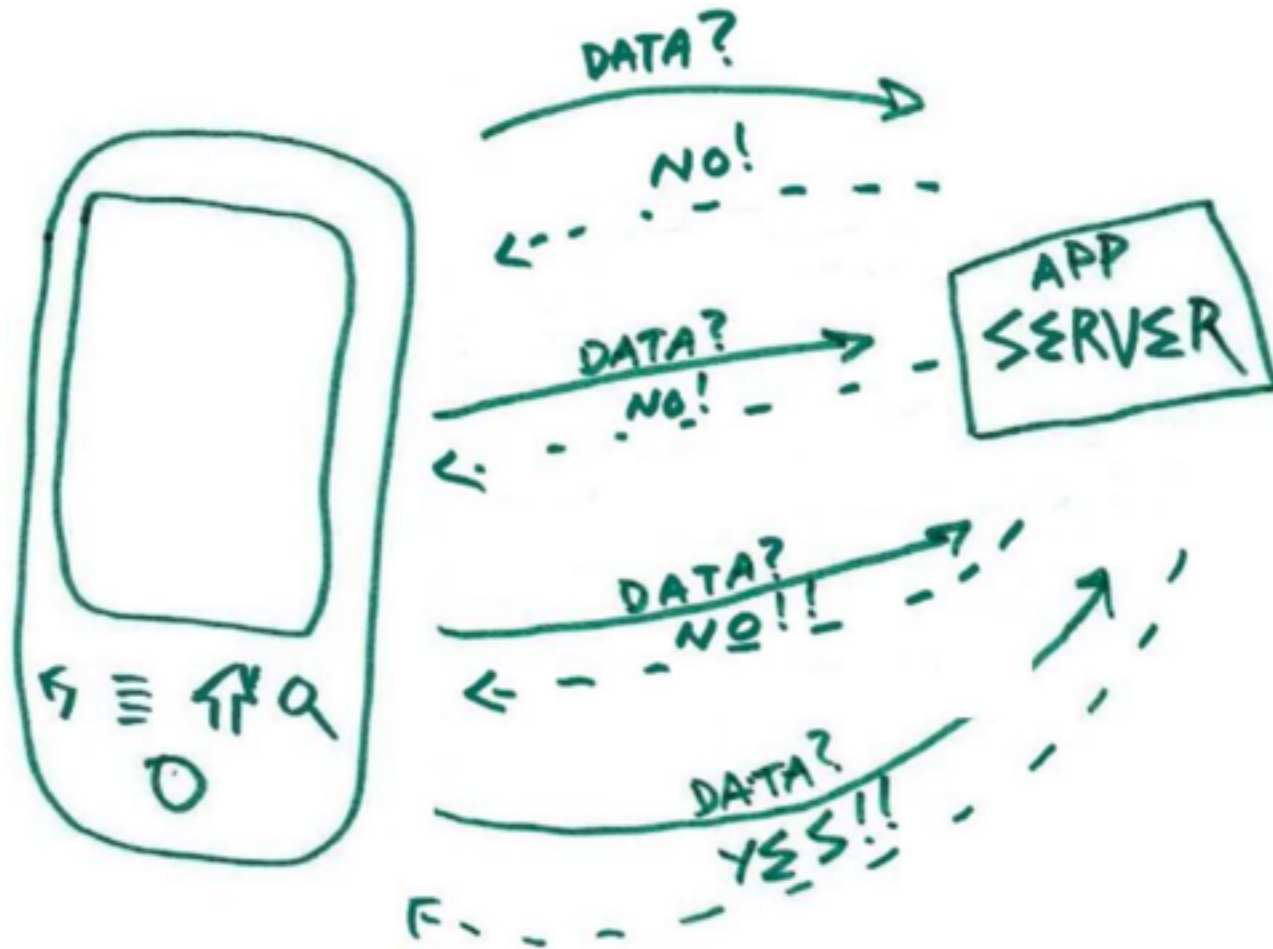
% of time spent on social networks in the United States, by platform*



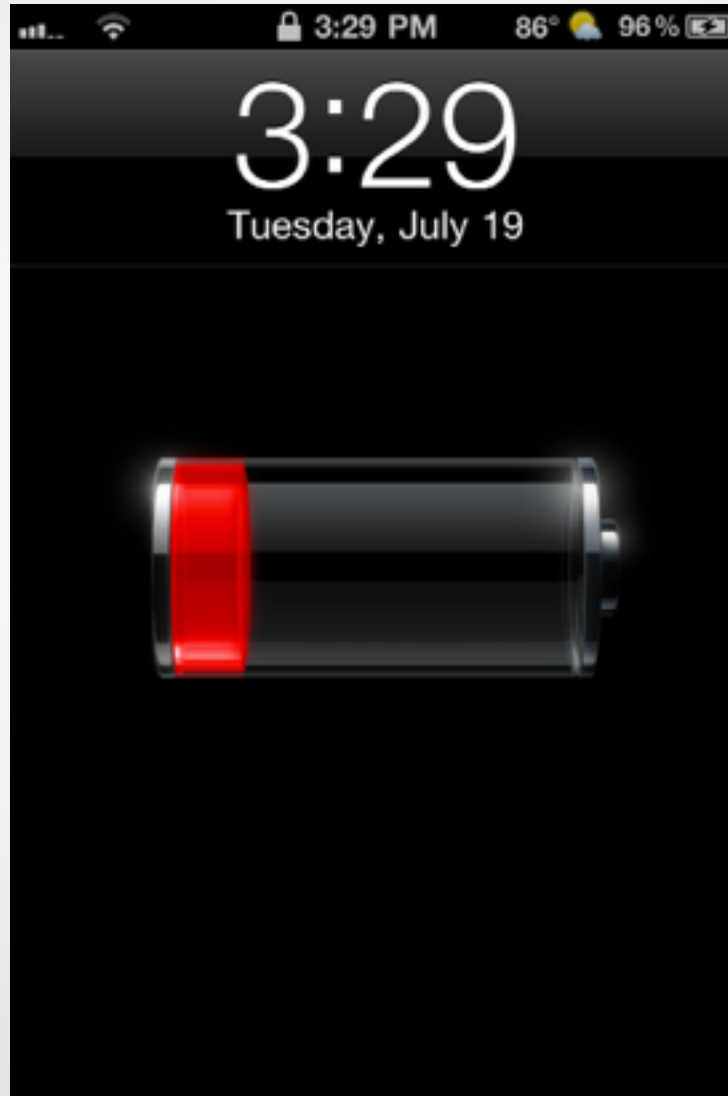
THE WALL STREET JOURNAL * December 2013, Age 18+

Source: comScore © ⓘ © statista

Our current model



Not very efficient



Servers are smart



How do we solve this?



What's a Push Notification?

- Display a short text message
- Play a brief sound
- iOS: Set a number in a badge on the app's icon
- Android: Display a banner on the notification bar

General architecture



Anatomy of a Push Notification

```
{  
  "aps":  
  {  
    "alert":  
    {  
      "action-loc-key": "Open",  
      "body": "Hello, world!"  
    },  
    "badge": 2  
  }  
}
```

High level comparison



Connection	<i>Streaming TCP/IP Socket</i>	<i>HTTP/XMPP</i>
Security	<i>SSL</i>	<i>SSL</i>
Content	<i>Binary</i>	<i>JSON</i>
Max payload	<i>256 bytes</i>	<i>4 kb</i>
Device errors	<i>Asynchronous query</i>	<i>HTTP/XMPP response</i>

Downside of PNs

- Not reliable
 - No guarantee the push will be delivered
- Stale information
 - Expirations

APPLE PUSH NOTIFICATION (APN)

What you need to get started

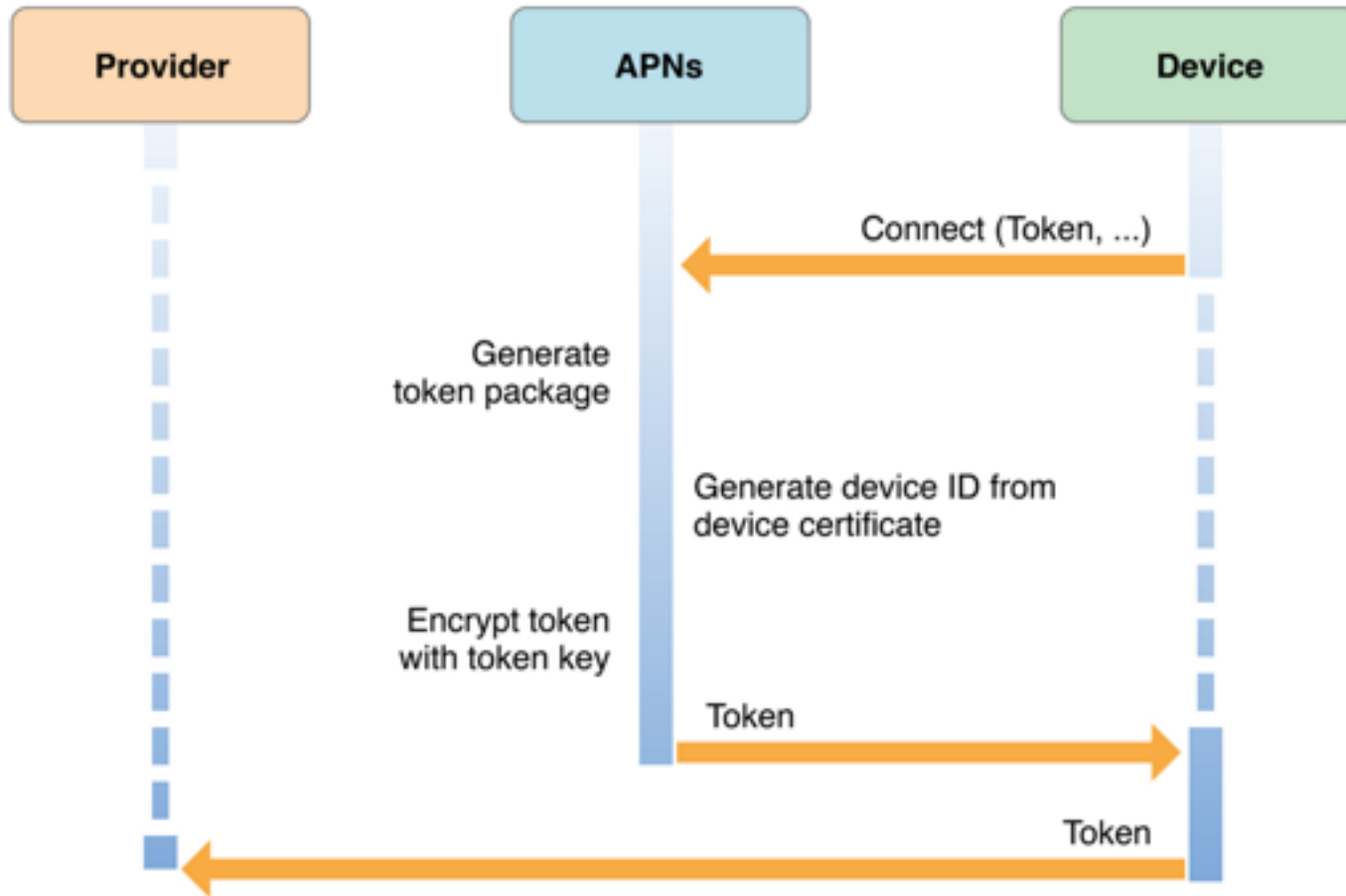
- iOS app
- Provisioning profile
- Certificate
- Device tokens
- Payload (a.k.a your messages)

Obtaining the certificate

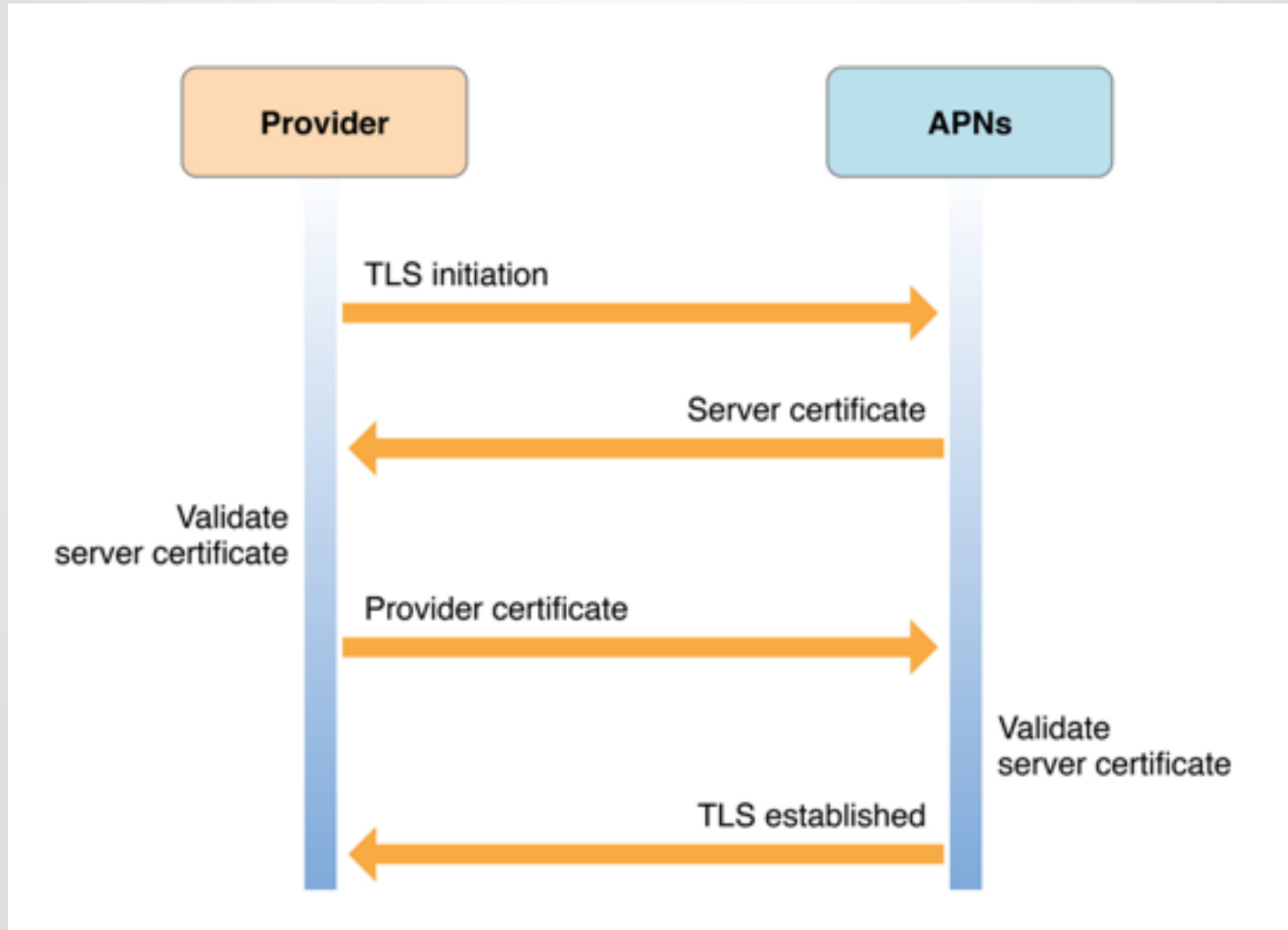
Similar to any other SSL cert

- Generate CSR
- Upload it to Apple
- Link it to your App Id
- Enable Push Notifications
- Generate certificate

Obtaining the device token



Establishing trust with APNs



Best way to do APNs on Python

Use **PyAPNS**

- It's on PyPI
 - `$ pip install apns`
- Source Code on Github
 - <https://github.com/djacobs/PyAPNs>

Send simple Push with PyAPNs

```
from apns import APNs, Frame, Payload
```

```
apns = APNs(use_sandbox=True, cert_file='cert.pem',  
            key_file='key.pem')
```

Send simple Push with PyAPNs

```
token_hex = 'some_token'  
payload = Payload(alert="Hello World!",  
                  sound="default",  
                  badge=1)  
apns.gateway_server.send_notification(token_hex,  
                                     payload)
```

Send multiple PNs with PyAPNs

```
import time
```

```
...
```

```
frame = Frame()
```

```
identifier = 1
```

```
expiry = time.time()+3600
```

```
priority = 10
```

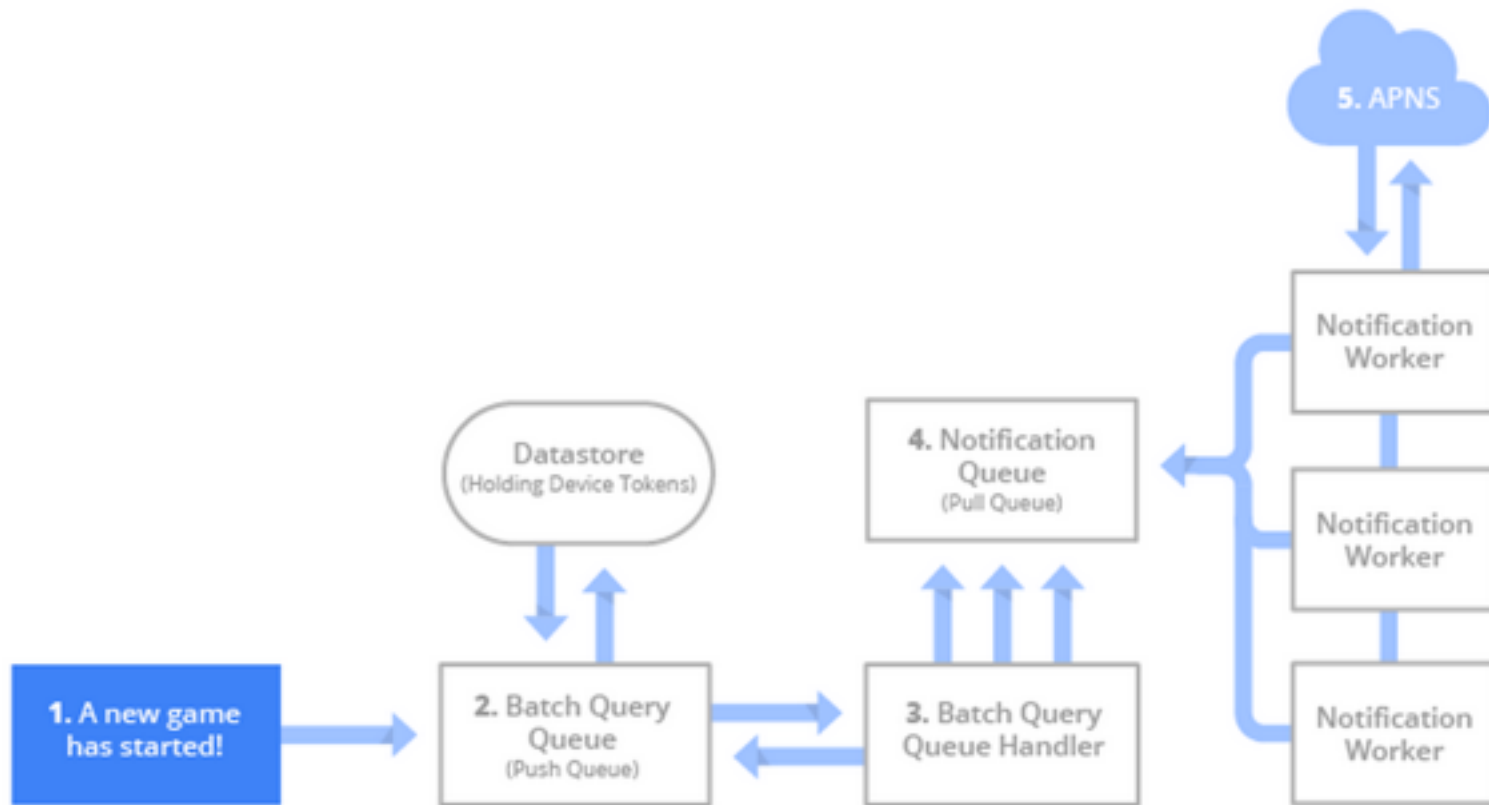
```
frame.add_item('some_token', payload, identifier,  
              expiry, priority)
```

```
frame.add_item(...)
```

```
frame.add_item(...)
```

```
apns.gateway_server.send_notification_multiple(frame)
```

Scaling your APNs implementation



GOOGLE CLOUD MESSAGING (GCM)

What you need to get started

- Android app
- Sender Id
- Application Id (a.k.a Package name)
- Registration Id
- Google User Account
- Sender Auth Token
- Payload (a.k.a your messages)

GCM has two implementations

- GCM HTTP
 - Uses HTTP POST
 - Downstream only
 - Easy to implement
- GCM Cloud Connection Server (CCS)
 - Based on XMPP (Jabber)
 - Two way communication

Good way to do GCM on Python

Use **SleekXMPP**

- It's on PyPI
 - `$ pip install sleekxmpp`
- Source Code on Github
 - <https://github.com/fritzy/SleekXMPP>

Connecting to GCM with SleekXMPP

```
from sleekxmpp import ClientXMPP
xmpp = ClientXMPP('PROJECT_ID@gcm.googleapis.com',
                  'API_KEY')
xmpp.connect(address=('gcm.googleapis.com', 5235),
             use_ssl=True)
```

3RD PARTY ALTERNATIVES

Parse

- Acquired by Facebook a year ago.
- Does much more than just Push Notifications
- Cross-platform SDKs
 - iOS, Android, Windows Phone, etc.
- Uses a very simple REST API for sending PNs

Send simple Push with Parse

```
import json,httplib  
connection = httplib.HTTPSConnection('api.parse.com', 443)  
connection.connect()
```

Send simple Push with Parse

```
connection.request('POST', '/1/push', json.dumps({
    "channels": [ "Giants", "Mets" ],
    "data": {
        "alert": "The Giants won against the Mets 2-3."
    }
}), {
    "X-Parse-Application-Id": "${APPLICATION_ID}",
    "X-Parse-REST-API-Key": "${REST_API_KEY}",
    "Content-Type": "application/json"
})
result = json.loads(connection.getresponse().read())
```

Downsides of Parse

- No Official Python SDK
 - 3rd party “recommended” SDK on Github
- Issues with their Mobile SDK
 - Specifically on Android
- They’re working through them

Urban Airship

- Push is still their main product
 - Moving into mobile marketing and payments.
- Based out of Portland, OR
- Simple REST API
- Cross-platform SDKs
 - iOS, Android, Blackberry, etc.
- Official Python SDK

Send simple Push with UA

```
import urbanairship as ua
airship = ua.Airship('application_key', 'master_secret')
push = airship.create_push()
push.audience = ua.or_(ua.alias('adam'),
                        ua.device_token('some_token'))
push.notification = ua.notification(alert='Hello')
push.device_types = ua.all_
push.send()
```

SUMMARY

Summary

- Mobile is eating the world.
- Push Notification will make your service smarter.
- Using Python for Push Notifications is easy.

Thank You!

Twitter: [@_juandg](https://twitter.com/_juandg)

Email: jgomez@eventbrite.com

Lanyrd: lanyrd.com/profile/juandg/