

**Next Session: 12:15pm**

Connecting to the Pulse  
of the Planet

Romain Huet



flight

#twitterflight



# Connecting to the Pulse of the Planet

October 22, 2014

#twitterflight



**Romain Huet**

Developer Advocate

@romainhuet

**271M+**  
Active Users

**500M+**  
Tweets per Day

2006

2014



Twitter is global,  
open and real-time



Twitter is the shortest distance  
between @you and your #interests



Twitter is there at  
**world events**





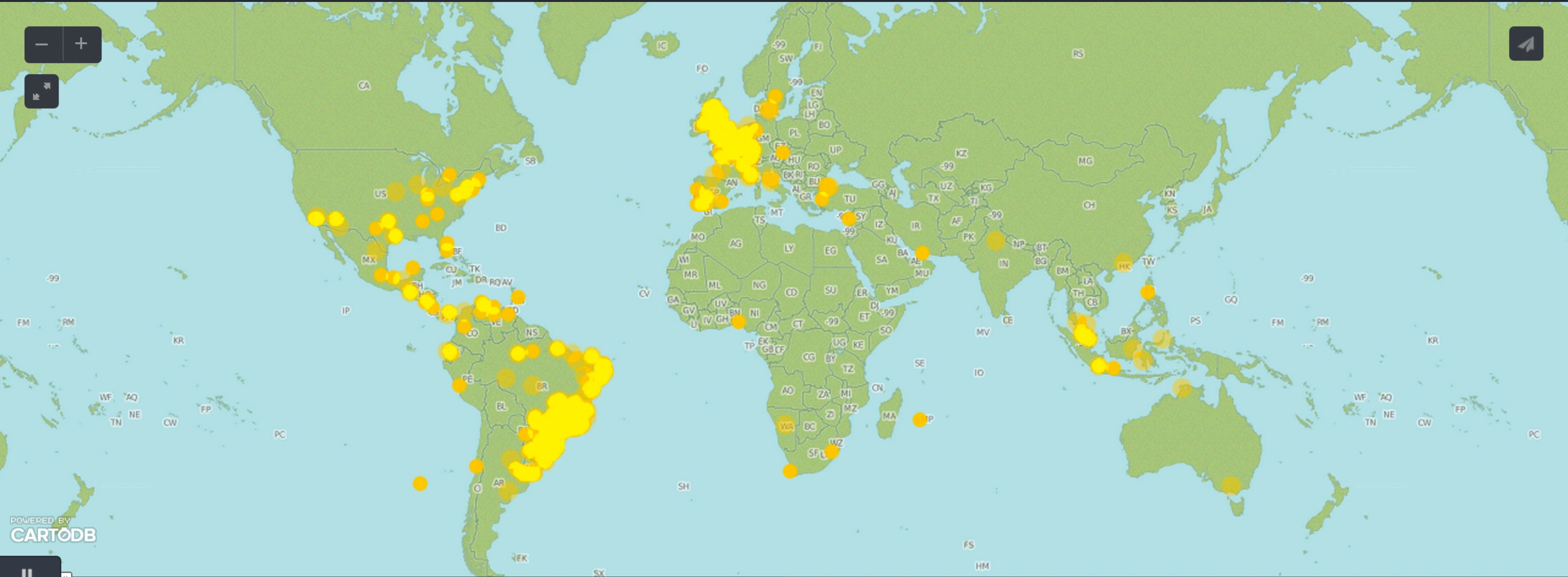
Twitter is the pulse  
of the planet



How can you use Twitter Data  
to power your apps?

# Semi-finals #BRAvGER: How Brazil v Germany played out on Twitter

Geotagged Tweets mentioning key terms around the World Cup game, July 8, 2014, Brazil time



16:55

● Brazil  0 | 0 |  Germany ●

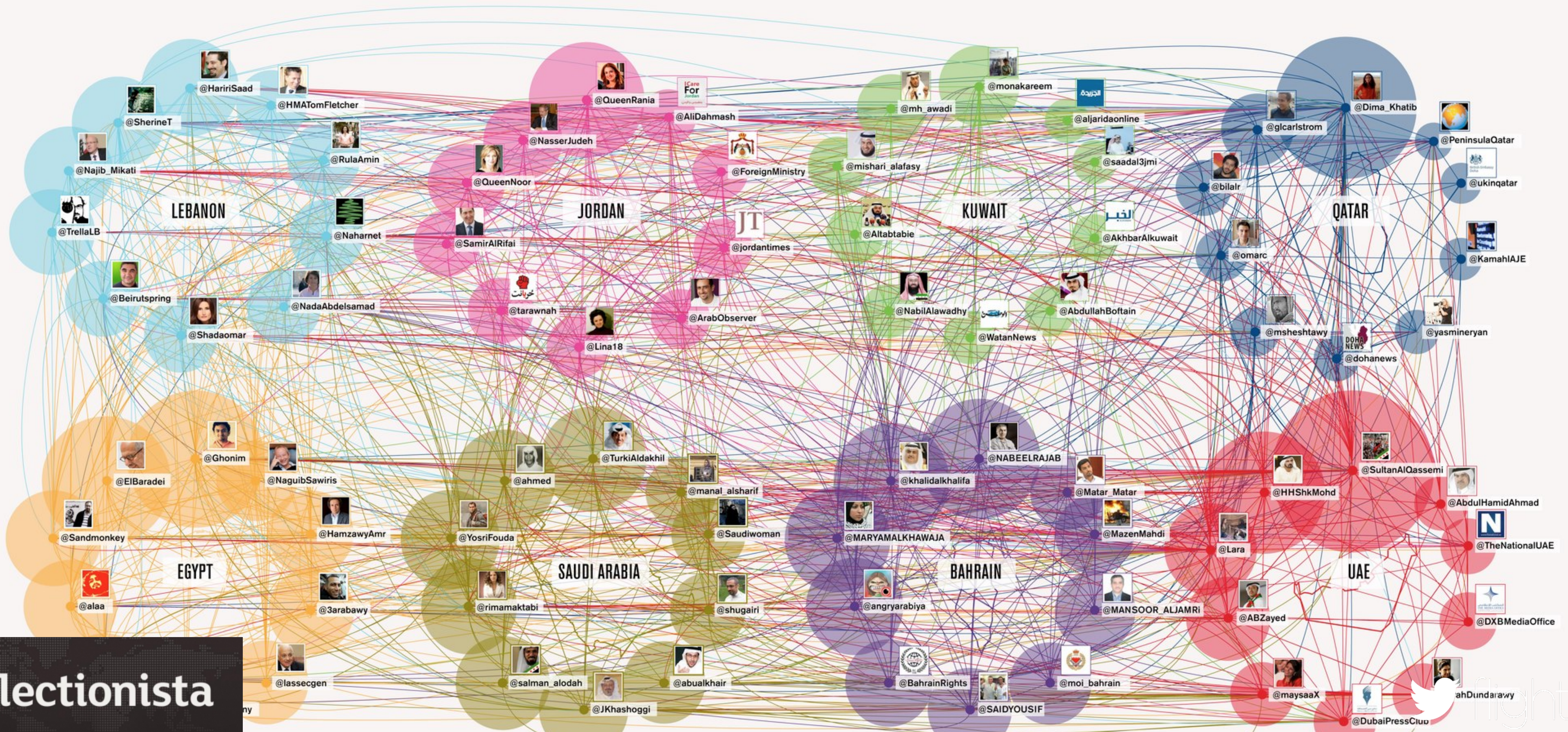


# HOW THE MIDDLE EAST TWEETS: THE MOST CONNECTED

# Portland

Follow us on Twitter: @portlandcomms  
Site: <http://www.portland-communications.com/>  
Powered by @tweetminster

- BAHRAIN
- EGYPT
- JORDAN
- KUWAIT
- LEBANON
- QATAR
- SAUDI ARABIA
- UAE



electionista

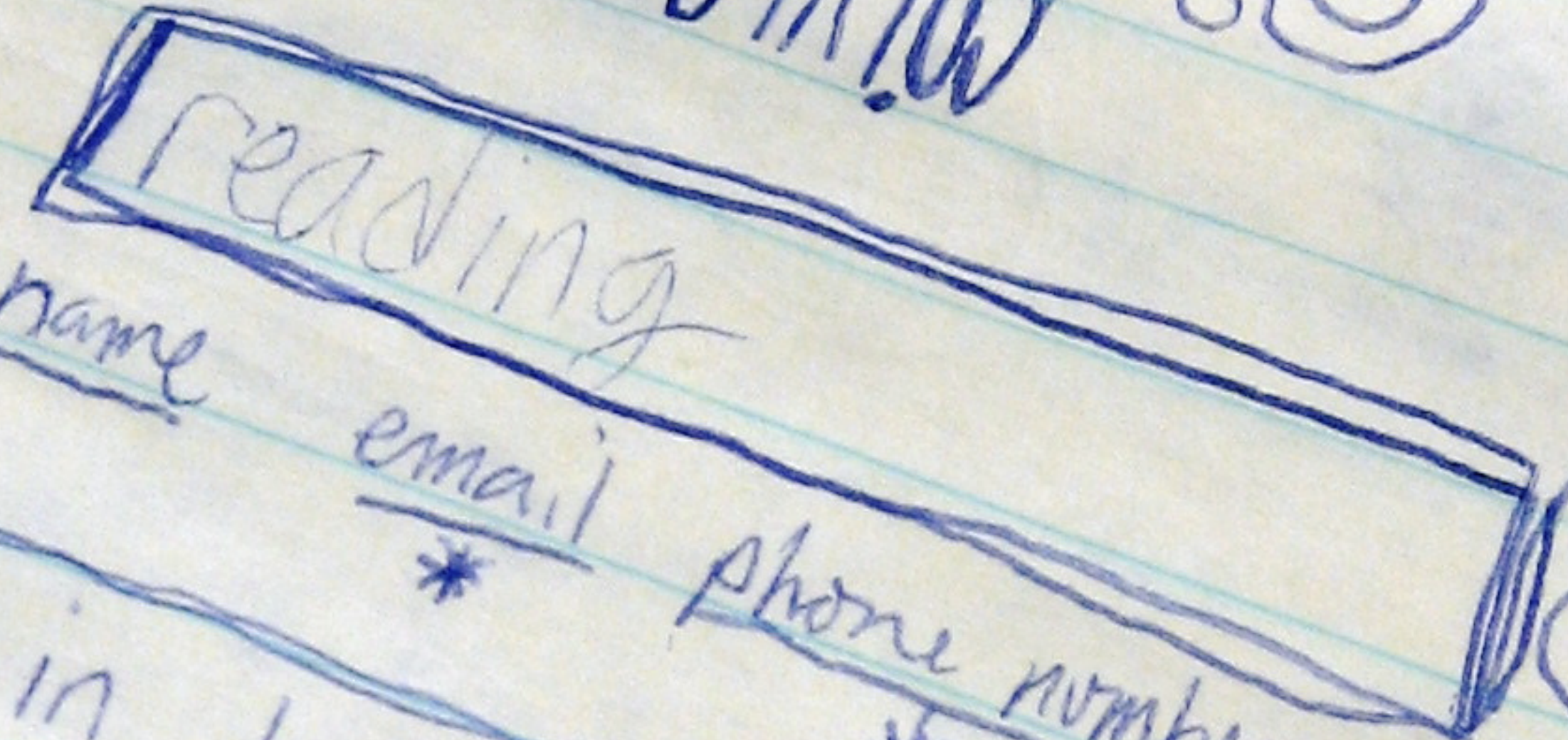
flight



# Data Platform

# STATUS

authentication  
triples



- in bed
- going to park

email looks for jabber ~~presence~~ presence and sets





Twitter APIs deal both with the  
right now and what happened





Streaming APIs allow you to ingest  
what's happening right now

# Streaming APIs let you ingest what's happening right now

```
POST https://stream.twitter.com/1.1/statuses/filter.json
```

Returns Tweets matching **your criteria**.

```
Data follow: 3946301  
track: #twitterflight  
locations: 2.1733,41.3633,2.1933,41.4033  
language: fr
```

# Streaming APIs let you ingest what's happening right now

GET <https://stream.twitter.com/1.1/statuses/sample.json>

Returns **~1%** of all firehose data, statistically relevant.

GET <https://stream.twitter.com/1.1/statuses/firehose.json>

Returns **all** firehose data, requires business deal.





# Anatomy of a Tweet

```
{
  "created_at": "Wed Oct 02 15:53:43 +0000 2013",
  "id": 385432417126739968,
  "id_str": "385432417126739968",
  "text": "Learn more about the powerful #Linux container engine @docker in this video intro with @solomonstre -
  http://t.co/QJLdA1762Y @TwitterOSS",
  "source": "web",
  "truncated": false,
  "in_reply_to_status_id": null,
  "in_reply_to_status_id_str": null,
  "in_reply_to_user_id": null,
  "in_reply_to_user_id_str": null,
  "in_reply_to_screen_name": null,
  "user": {
    "id": 1665823832,
    "id_str": "1665823832",
    "name": "Twitter University",
    "screen_name": "university",
    "location": "",
    "description": "Building a culture of learning for @TwitterEng and the world at large",
    "url": "http://t.co/c8IHJLE1Y8",
    "entities": {
      "url": {
        "urls": [{
          "url": "http://t.co/c8IHJLE1Y8",
          "expanded_url": "http://youtube.com/twitteruniversity",
          "display_url": "youtube.com/twitterunivers\u2026",
          "indices": [0, 22]
```



# 143,199

Tweets per second

## 5,700

TPS on average

A FILM BY HAYAO MIYAZAKI



REST APIs let you perform actions  
and access what happened

# REST APIs let you perform actions and access what happened

Tweets & Media	GET	<code>statuses/lookup</code>
	POST	<code>statuses/update</code>
	POST	<code>media/upload</code>
Timelines	GET	<code>statuses/home_timeline</code>
	GET	<code>statuses/user_timeline</code>
	GET	<code>statuses/mentions_timeline</code>
Search	GET	<code>tweets/search</code>
Following	GET	<code>friends/list</code>

*And many more: Users, Favorites, Direct Messages, Followers, Lists, Trends, Geo, Spam Reporting...*

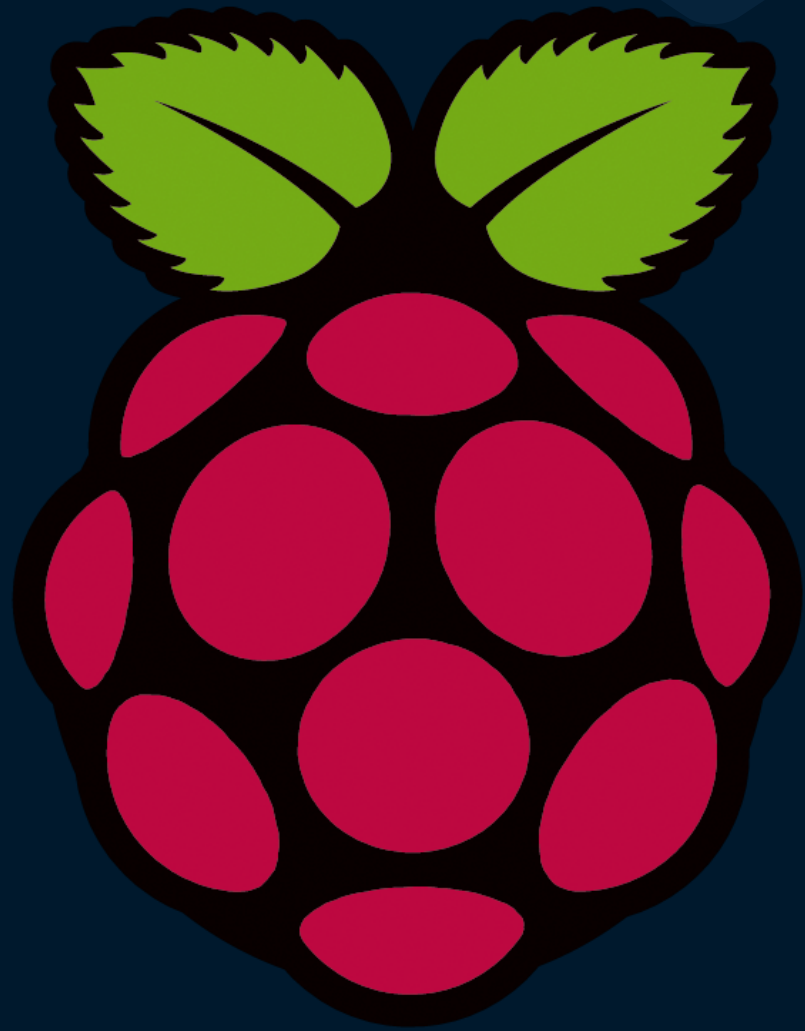




flight

**Demo**

#twitterflight



**Romain Huet**

@romainhuet



Selfie from stage with the great people at [@APIDaysBerlin](#),  
tweeted by a Raspberry Pi! [#apidays](#)

3:59 AM - 6 May 2014

52 RETWEETS 35 FAVORITES

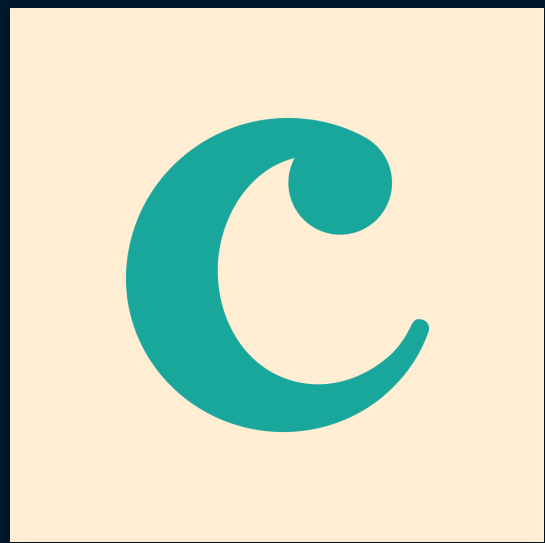


# Mobile Platform



faboric

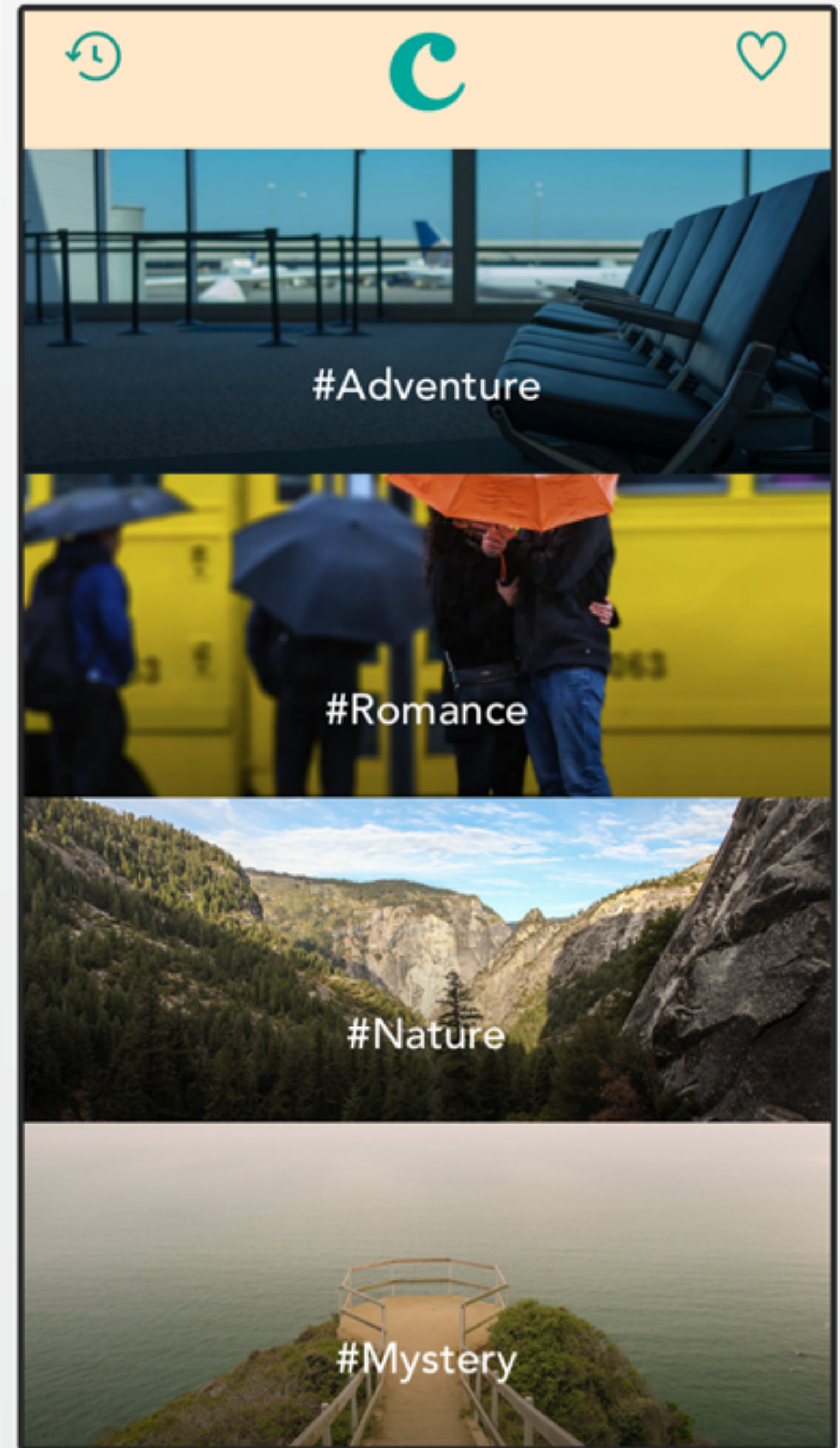




# Cannonball

Fabric Sample App

Open source for iOS & Android:  
[github.com/twitterdev](https://github.com/twitterdev)





NodeCopt





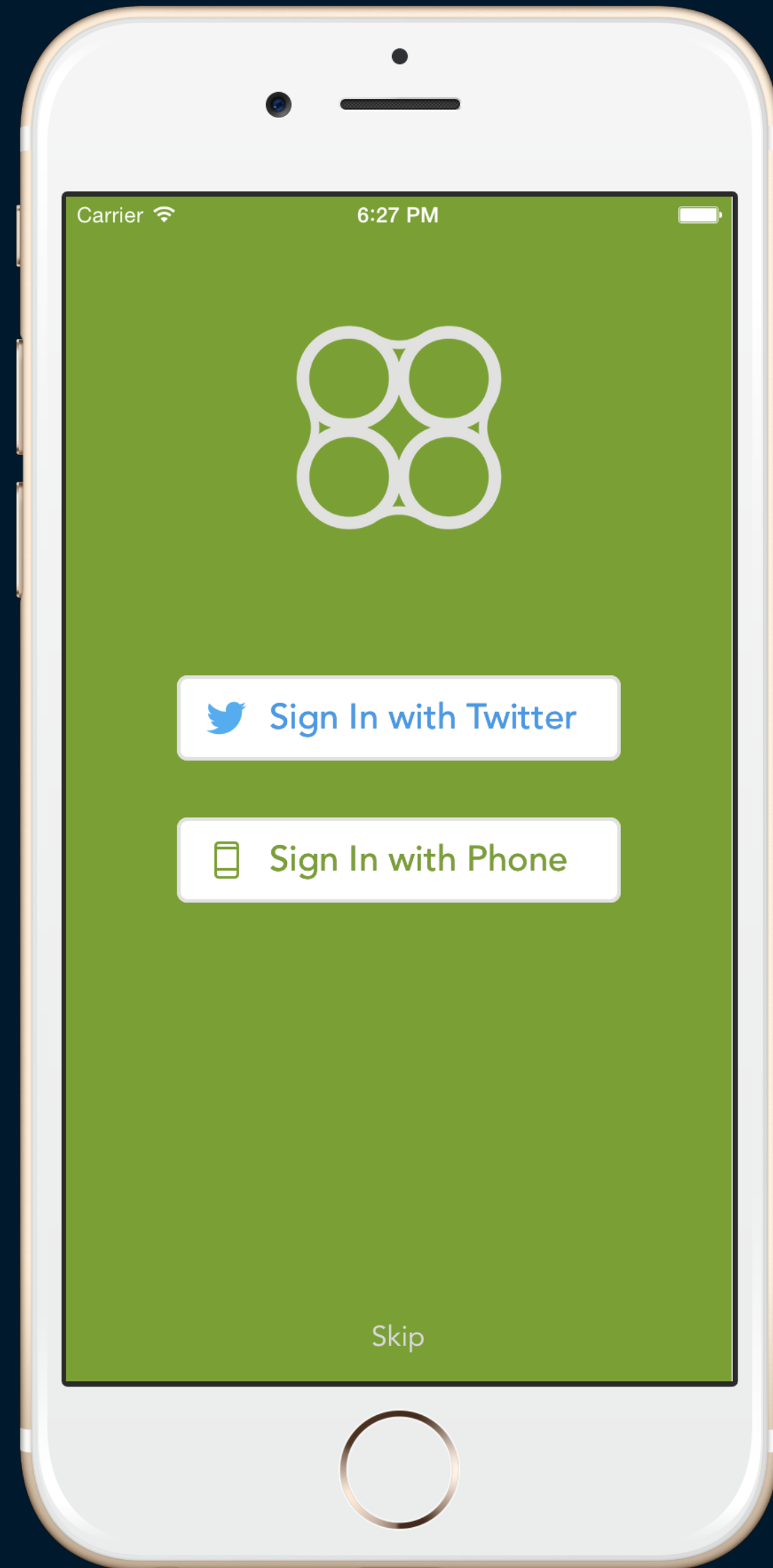




**Demo**

#twitterflight

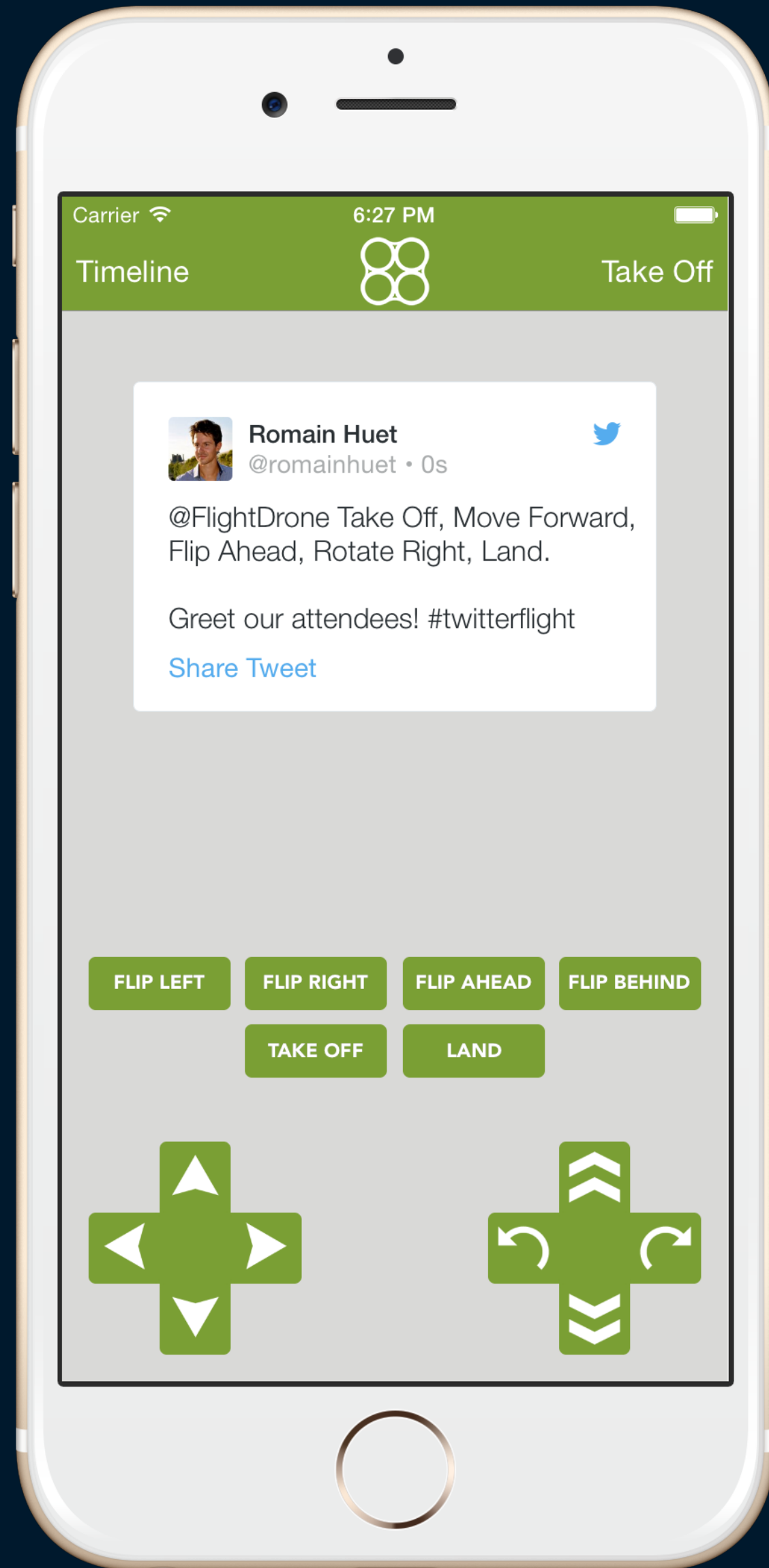
# Sign In with Twitter and Digits



```
@IBAction func signInWithTwitter(sender: UIButton) {
    Twitter.sharedInstance().loginWithCompletion { (session:
TWTRSession!, error: NSError!) -> Void in
        if session != nil {
            // Save any useful data from the Twitter session,
            // then move to the next view controller.
        }
    }
}

@IBAction func signInWithPhone(sender: UIButton) {
    Digits.sharedInstance().authenticateWithCompletion { (session:
DGTSession!, error: NSError!) -> Void in
        if session != nil {
            // Save any useful data from the Digits session,
            // then move to the next view controller.
        }
    }
}
```

# Adding a Tweet View to the View Controller

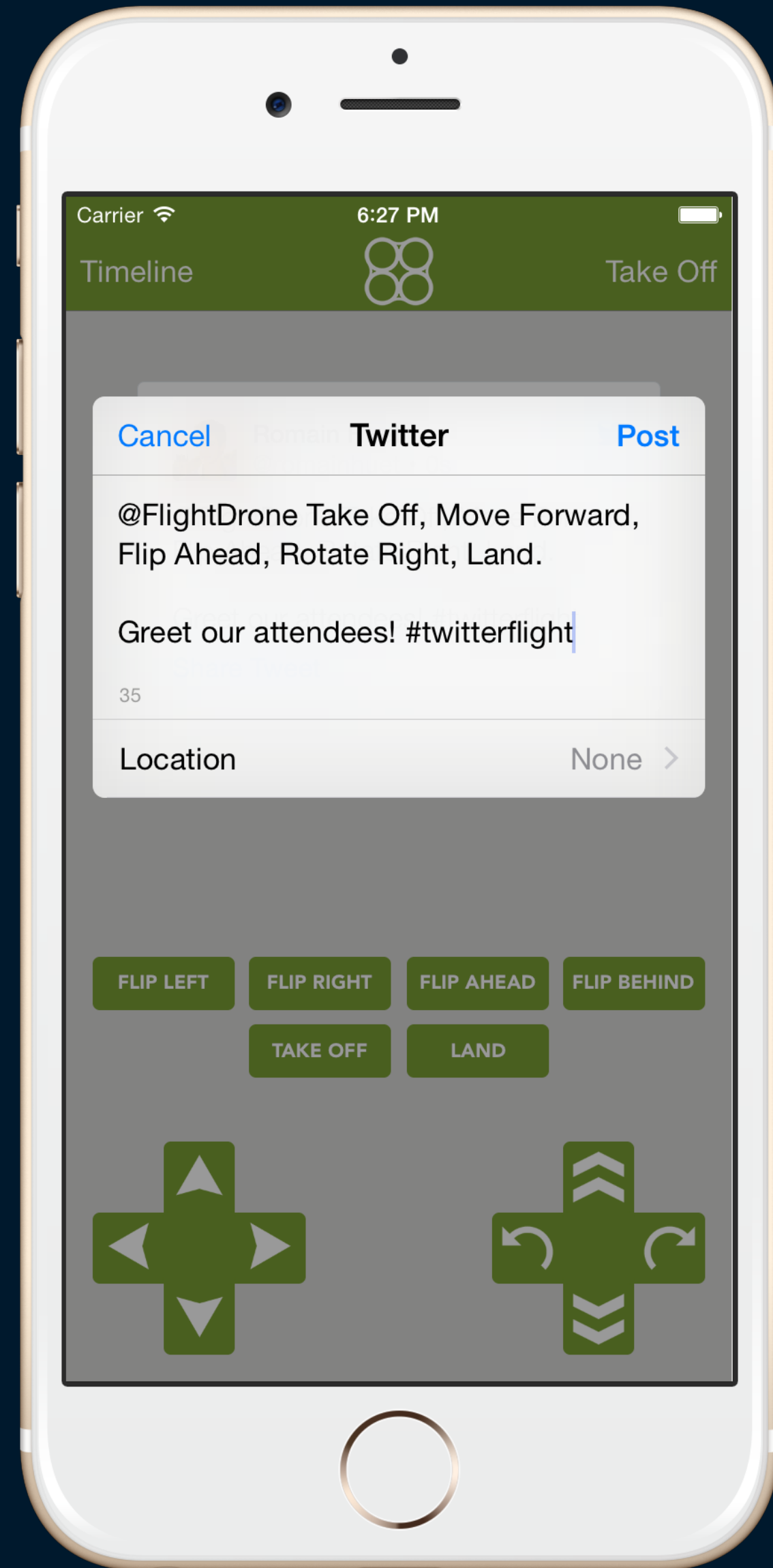


```
override func viewDidLoad() {
    super.viewDidLoad()

    // Request the Twitter API.
    Twitter.sharedInstance().APIClient.loadTweetWithID(tweetId,
completion: { (tweet: TWTRTweet!, error: NSError!) -> Void in
    // Build the Tweet view.
    self.tweetView = TWTRTweetView(tweet: tweet, style: .Regular)

    // Add the Tweet view to the main view.
    self.view.addSubview(self.tweetView)
    })
}
```

# Sharing a Tweet with the Tweet Composer

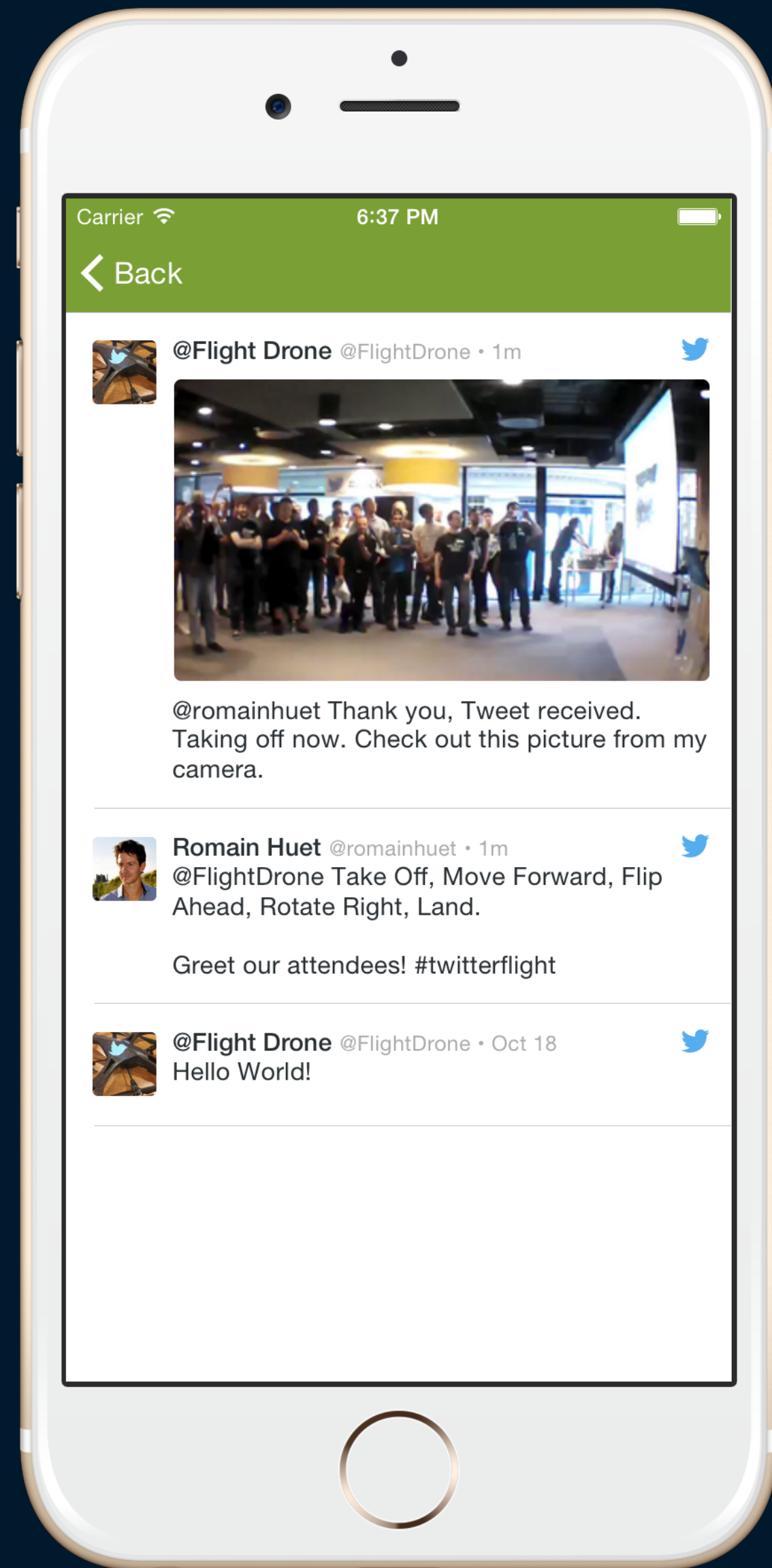


```
@IBAction func postTweet(sender: AnyObject) {
    // Use the TwitterKit to create a Tweet composer.
    let composer = TWTRComposer()

    // Prepare the Tweet text with drone commands.
    composer.setText(droneCommands)

    // Present the composer to the user.
    composer.showWithCompletion({ (result: TWTRComposerResult!) -> Void
in
        if result == .Cancelled {
            println("Tweet composition cancelled")
        } else {
            println("Tweeting commands to the drone!")
        }
    })
}
```

# Displaying Tweets in a Table View



```
override func tableView(tableView: UITableView, cellForRowAtIndexPath
indexPath: NSIndexPath) -> UITableViewCell {
    // Retrieve the Tweet cell.
    let cell =
tableView.dequeueReusableCellWithIdentifier(tweetTableCellReuseIdentifie
r, forIndexPath: indexPath) as TWTRTweetTableViewCell

    // Assign the delegate to control events on Tweets.
    cell.tweetView.delegate = self

    // Retrieve the Tweet model object from the array of loaded Tweets.
    let tweet = tweets[indexPath.row]

    // Configure the cell with the Tweet.
    cell.configureWithTweet(tweet)

    // Return the Tweet cell.
    return cell
}
```



Thank You