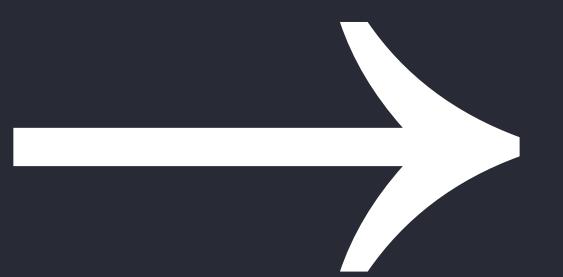


# What's new in iOS 9

Cocoaheads Paris, jan. 2017



@adumi, Trainline



MONTH

DAY

YEAR

HOUR

MIN

JUN

03

2015

AM  
PM

01:21

01:21

DESTINATION TIME

MONTH

DAY

YEAR

HOUR

MIN

OCT

26

1985

AM  
PM

01:22

01:22

PRESENT TIME

MONTH

DAY

YEAR

HOUR

MIN

OCT

26

1985

AM  
PM

01:20

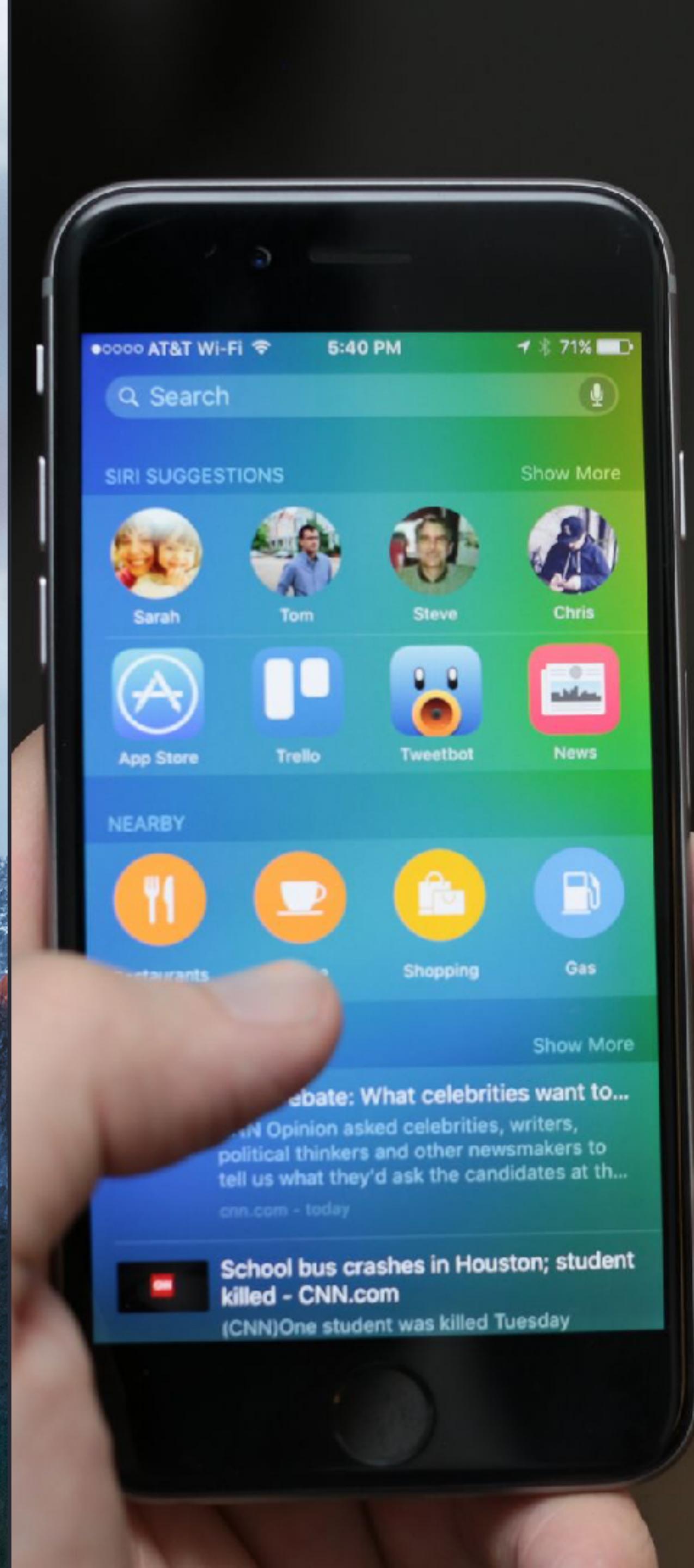
01:20

LAST TIME DEPARTED



The epicenter  
of change.

WWDC15

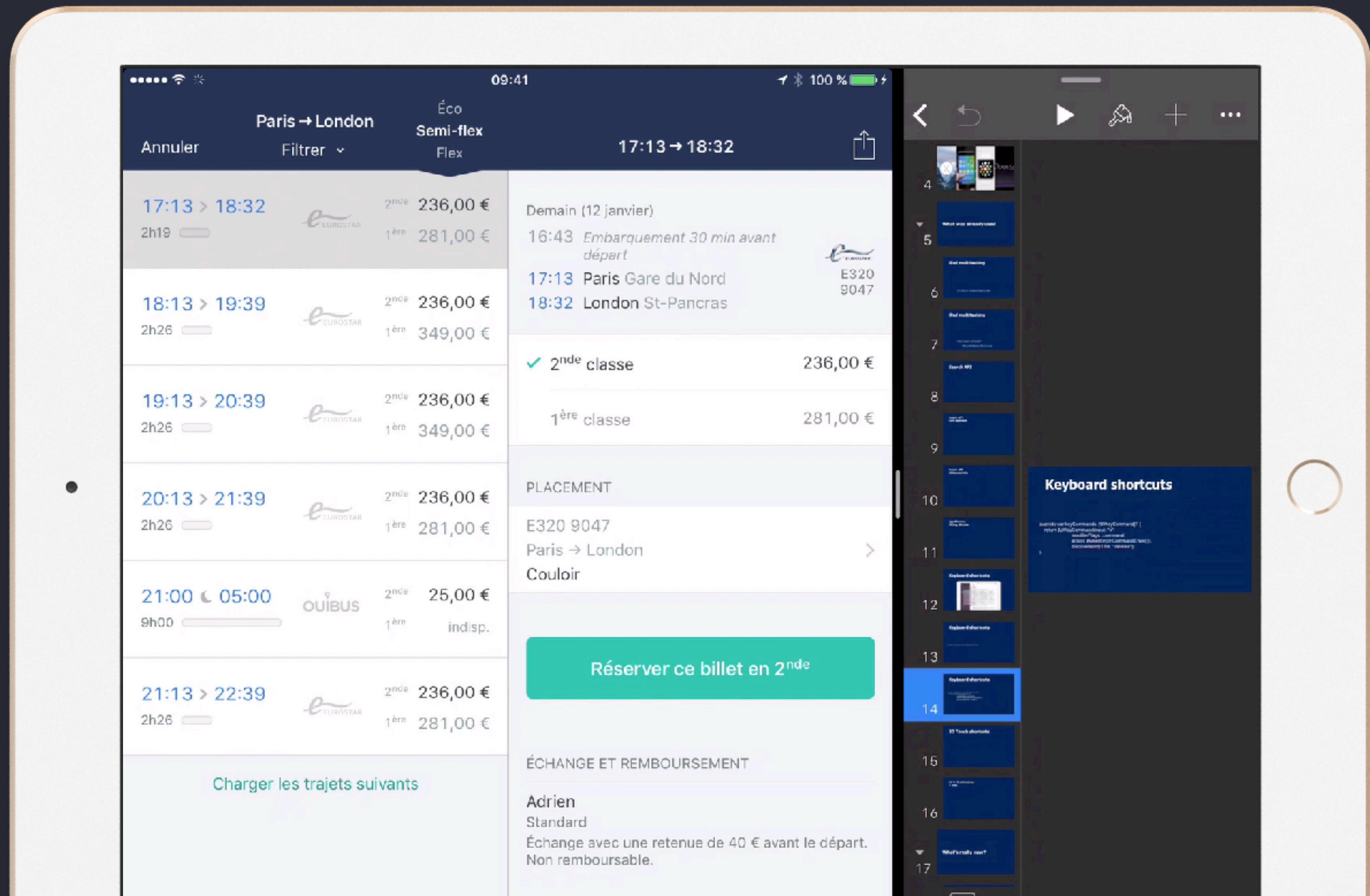




# What is already used

(or could be)

# iPad Multitasking / Split screen & slide over



# iPad Multitasking / Split screen & slide over

Out of the box ✨

(if you support size classes correctly)

# iPad Multitasking / Split screen & slide over

Opt-out (info.plist)

```
<key>UIRequiresFullScreen</key>
<true/>
```

# Search API



Add content to spotlight

The screenshot shows the iOS Spotlight search interface. At the top, a search bar contains the query "carpentras". Below the search bar, there are several cards representing search results:

- CALENDRIER**: A card for a train trip from Paris to Carpentras on Saturday, March 18, 2017, departing from Gare Paris Montparnasse at 07:36 and arriving at 12:36.
- TRAINLINE EU**: Two cards for train tickets from Paris to Carpentras, one for a round-trip and another for a frequent route.
- WIKIPÉDIA**: A card for the Wikipedia page about Carpentras, featuring a thumbnail image of the town.
- SITES SUGGÉRÉS**: A card for the official website of the city of Carpentras.

The status bar at the top of the phone screen indicates signal strength, battery level, and the time (23:20).

# Search API



## NSUserActivity

For current activity indexation (can be public)

```
var activity = NSUserActivity(activityType: "com.ct.capitainetrain")
activity.title = "Paris → Carpentras"
activity.userInfo = ["id": "www.trainline.fr/results/paris/carpentras"]
activity.isEligibleForSearch = true
```

# Search API

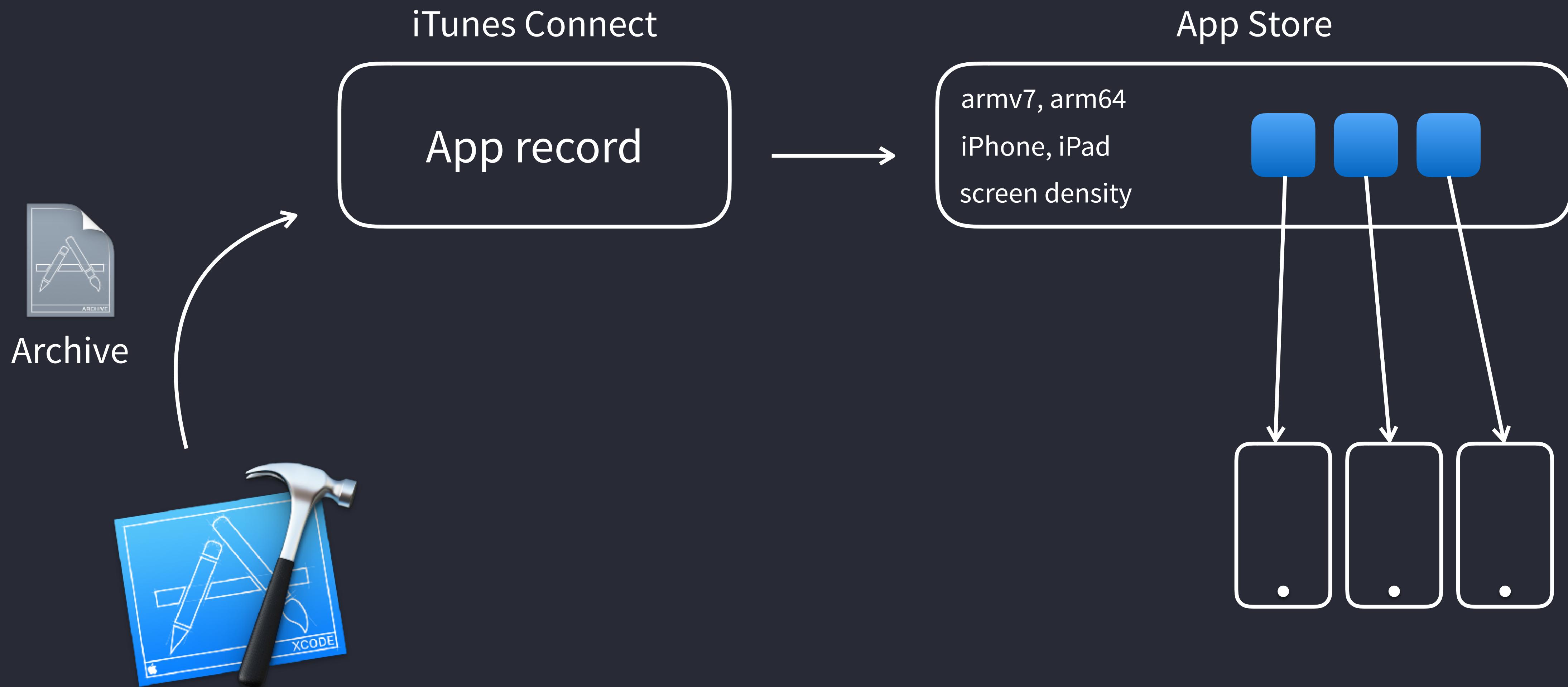


## CoreSpotlight

For massive and/or private, indexation

```
let attributeSet = CSSearchableItemAttributeSet(  
    itemContentType: kUTTypeData)  
attributeSet.title = "Private searchable item"  
  
let item = CSSearchableItem(uniqueIdentifier: "AE4F-C8C24D",  
    domainIdentifier: "domain",  
    attributeSet: attributeSet)  
  
CSearchBar.default().indexSearchableItems([item],  
    completionHandler: nil)
```

# App thinning / Bitcode & App Slicing



# App thinning / Bitcode & App Slicing

~20-50% size gain

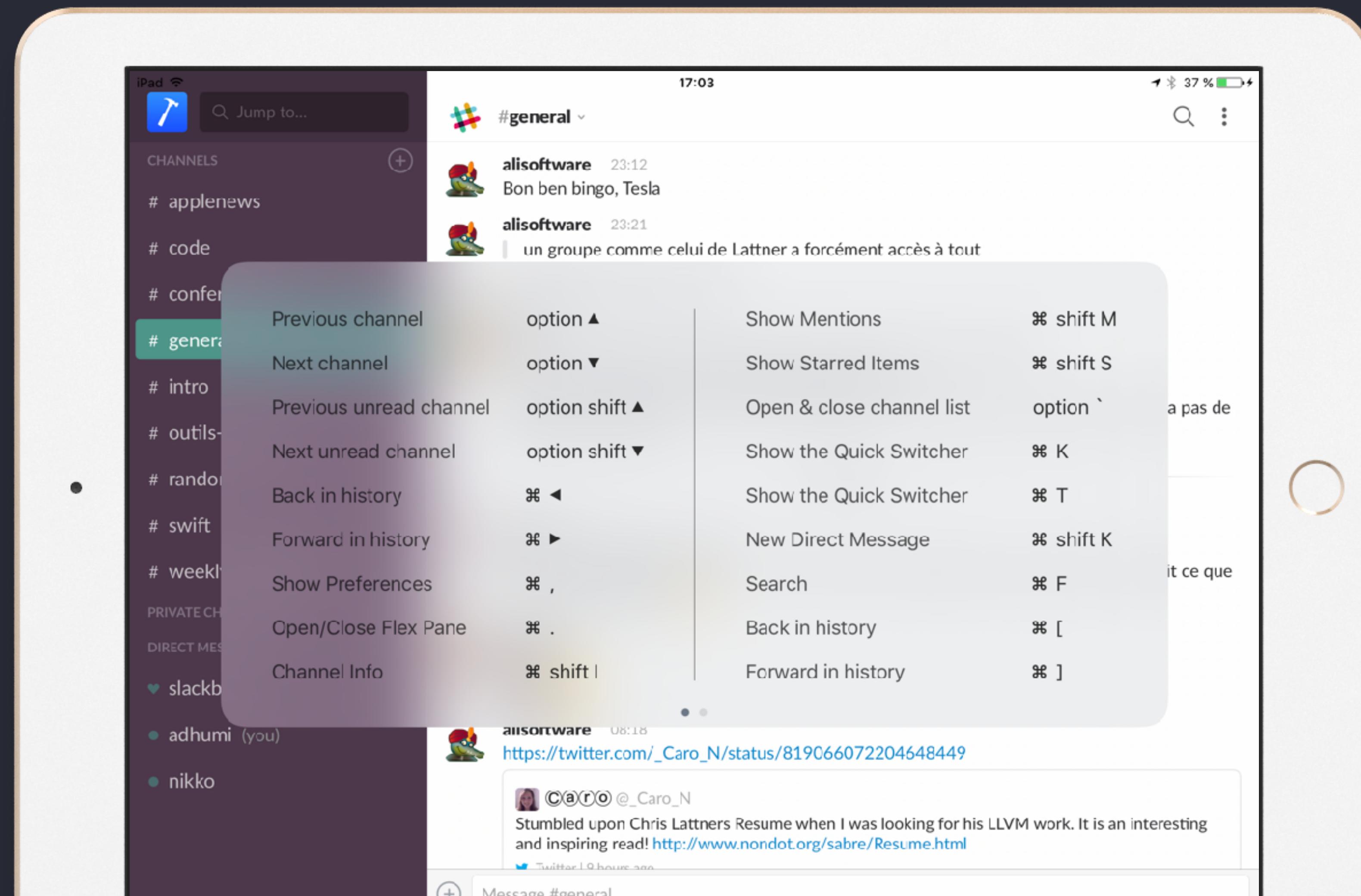
Default on iOS (optional)

Mandatory on watchOS and tvOS



Dependencies need to support Bitcode

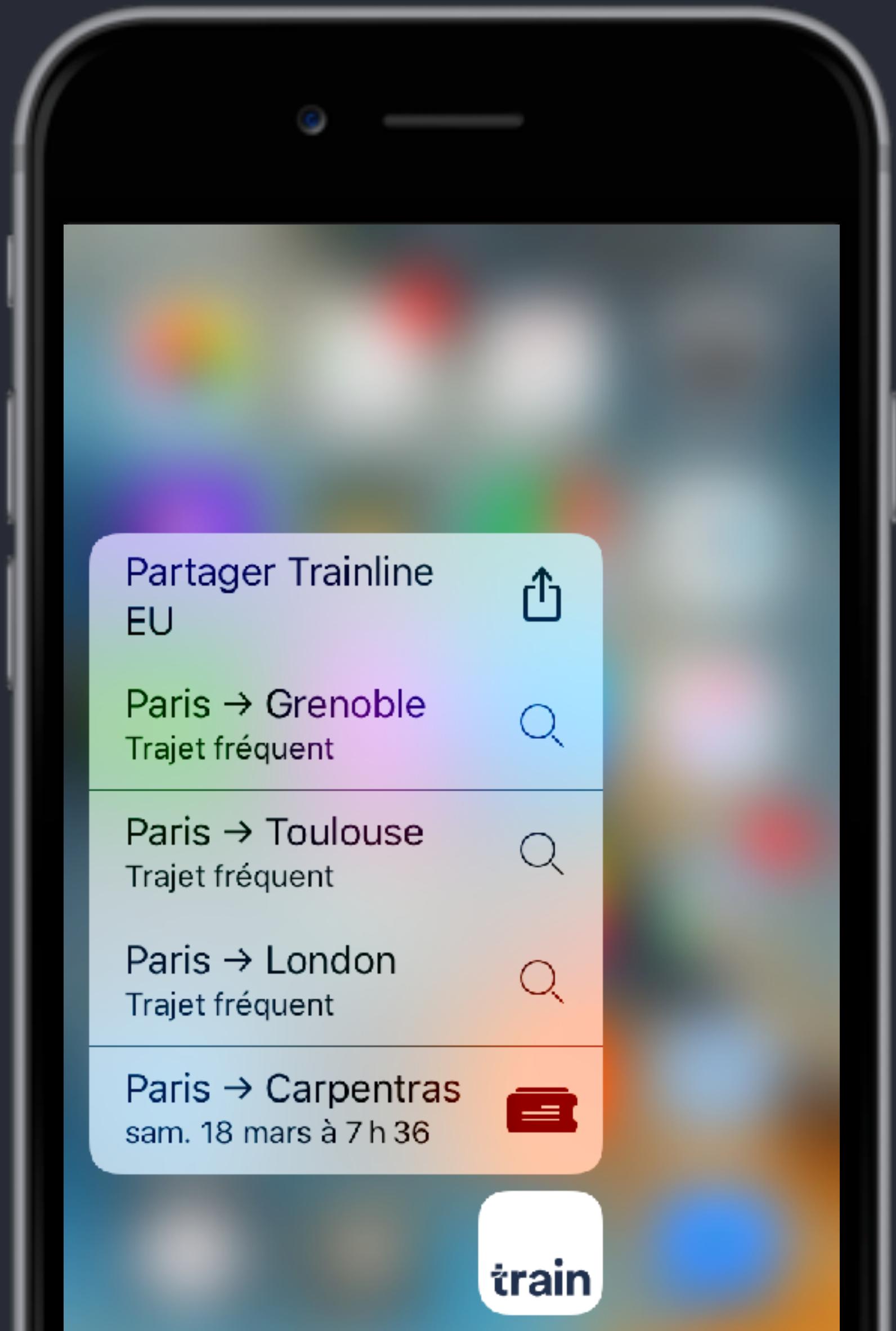
# Keyboard shortcuts



# Keyboard shortcuts

```
override var keyCommands: [UIKeyCommand]? {  
    return [UIKeyCommand(input: "\r",  
                         modifierFlags: .command,  
                         action: #selector(onCommandEnter(_:)),  
                         discoverabilityTitle: "Validate")]  
}
```

# 3D Touch / Quick Actions



# 3D Touch / Quick Actions

Static    Compile time

|                                     |            |   |
|-------------------------------------|------------|---|
| ▀ UIApplicationShortcutItems        | Array      | (3 items)                                   |
| ▀ Item 0                            | Dictionary | (5 items)                                   |
| UIApplicationShortcutItemIconFile   | String     | StarTemplate                                |
| UIApplicationShortcutItemSubtitle   | String     | shortcutSubtitleFavorites                   |
| UIApplicationShortcutItemTitle      | String     | shortcutTitleFavorites                      |
| UIApplicationShortcutItemType       | String     | \$(PRODUCT_BUNDLE_IDENTIFIER).OpenFavorites |
| ▀ UIApplicationShortcutItemUserInfo | Dictionary | (1 item)                                    |
| version                             | String     | 3   |

Dynamic    Runtime

```
let shortcut = UIApplicationShortcutItem(type: "fr.adhumi.app.message",
                                         localizedTitle: "Send message")
UIApplication.shared.shortcutItems = [shortcut]
```

# 3D Touch / Peek & Pop

```
override func viewDidLoad() {
    self.registerForPreviewing(with: self, sourceView: self.view)
}

func previewingContext(_ previewingContext: UIPreviewControllerPreviewing,
                      viewControllerForLocation location: CGPoint)
    -> UIViewController? {
    return MyController()
}

func previewingContext(_ previewingContext: UIPreviewControllerPreviewing,
                      commit vc: UIViewController) {
    self.navigationController?.pushViewController(vc, animated: true)
}
```

# What's really new

and can be adopted

~~UIAlertView~~

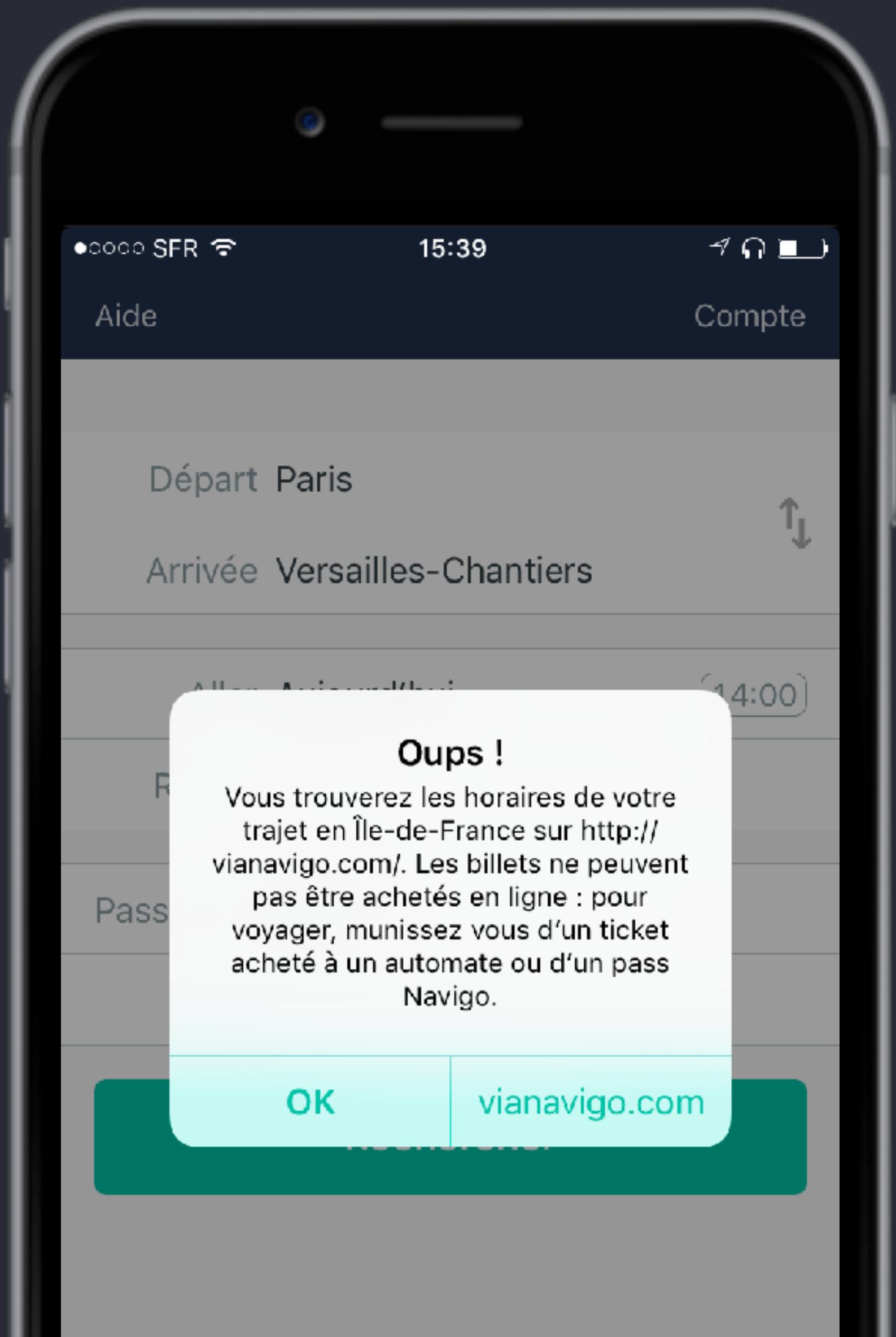
~~UIActionSheet~~



**NEW**

UIAlertController

# UIAlertController



# UIAlertController

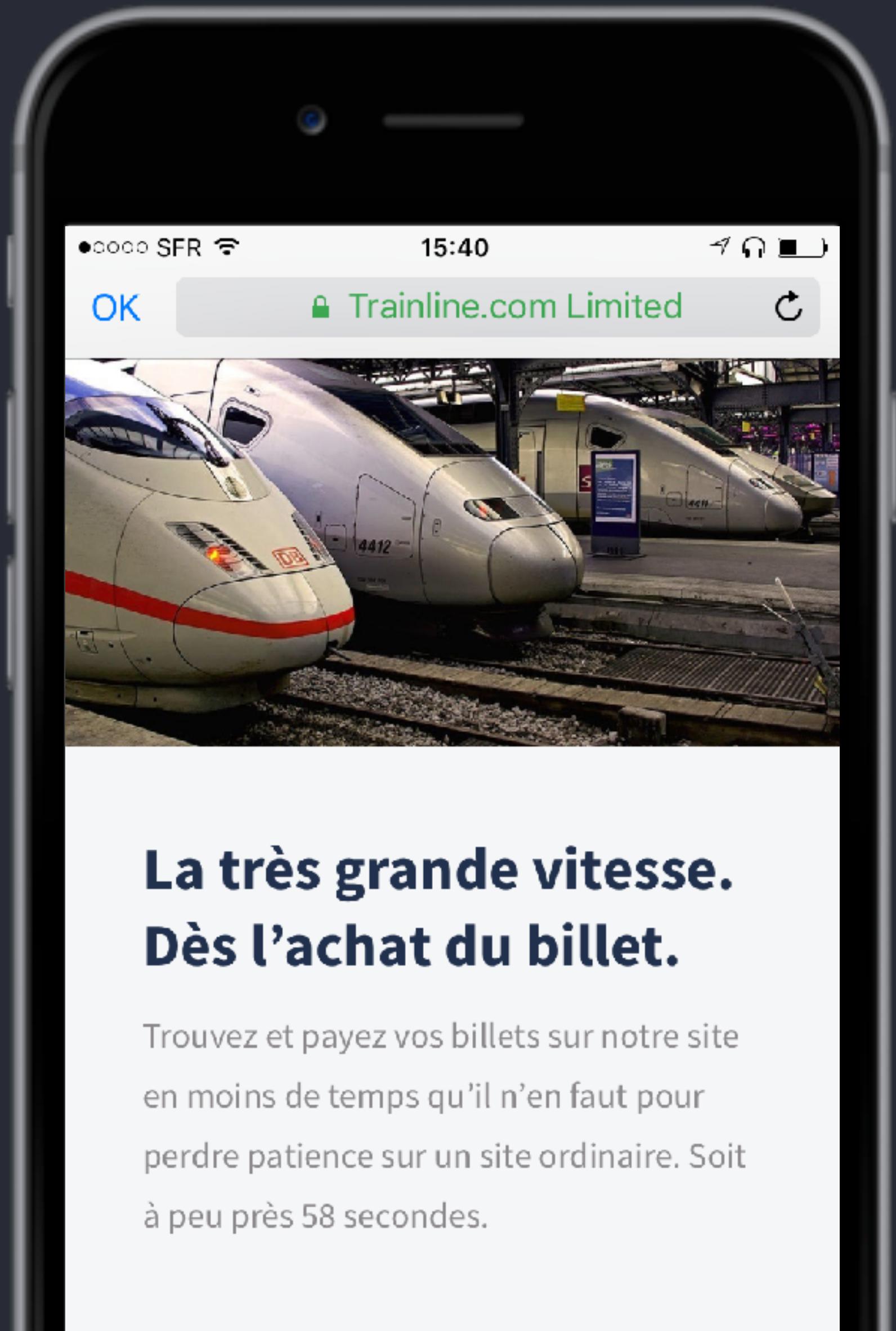
```
let alert = UIAlertController(title: "Uh oh!",  
                            message: "An error occurred.",  
                            preferredStyle: .alert)  
  
let resolve = UIAlertAction(title: "Resolve",  
                           style: .destructive) { (action) in  
    // Do stuff  
}  
alert.addAction(resolve)  
  
let cancel = UIAlertAction(title: "Cancel",  
                           style: .cancel) { (action) in  
    // Do stuff  
}  
alert.addAction(cancel)  
  
self.present(alert, animated: true, completion: nil)
```

~~UIWebView~~  
~~WKWebView~~



**NEW**  
**SFSafariViewController**

# SFSafariViewController



# SFSafariViewController

```
let url = URL(string: "https://www.trainline.fr")  
let safari = SFSafariViewController(url: url!)  
self.navigationController?.pushViewController(safari, animated: true)
```

~~AddressBook.framework~~



**NEW**  
Contacts.framework

# Contacts.framework

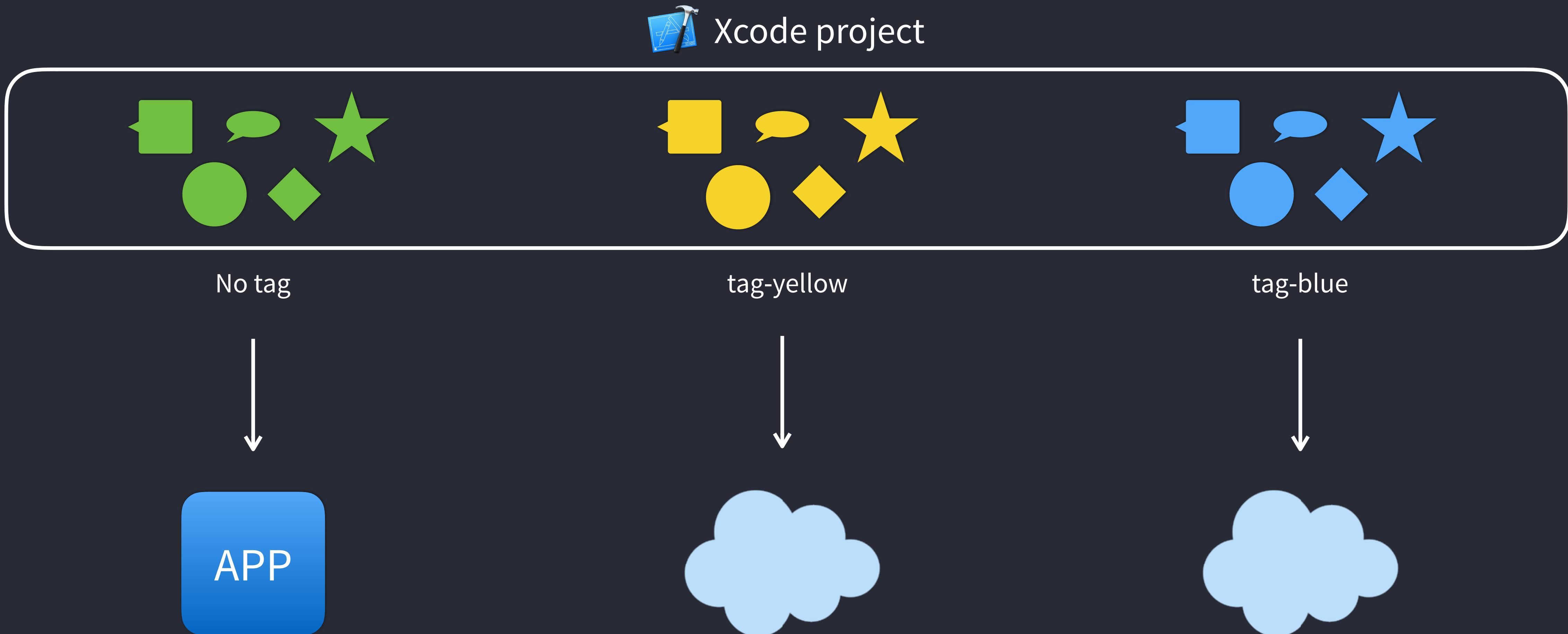
```
let contacts = CNContactPickerController()  
contacts.delegate = self  
self.present(contacts, animated: true, completion: nil)
```

```
func contactPicker(_ picker: CNContactPickerController,  
                   didSelect contact: CNContact) {  
    print(contact.givenName)  
    print(contact.emailAddresses)  
}
```

```
let firstName: String = ABRecordCopyValue(contact,  
                                         kABPersonFirstNameProperty).takeRetainedValue()
```



# App thinning / On Demand Resources



# App thinning / On Demand Resources

```
let tags: Set<String> = ["tag-yellow, tag-blue"]
let request = NSBundleResourceRequest(tags: tags)

request.beginAccessingResources { (error) in
    if let error = error {
        print(error)
    } else {
        print("Resources available")
    }
}
```

```
NSLayoutConstraint(item: view1,  
                    attribute: .centerX,  
                    relatedBy: .equal,  
                   toItem: view2,  
                  attribute: .leading,  
                 multiplier: 1,  
                constant: 0)
```



**NEW**

```
view1.centerXAnchor.constraint(equalTo: view2.leadingAnchor)
```

# Auto Layout

New syntax

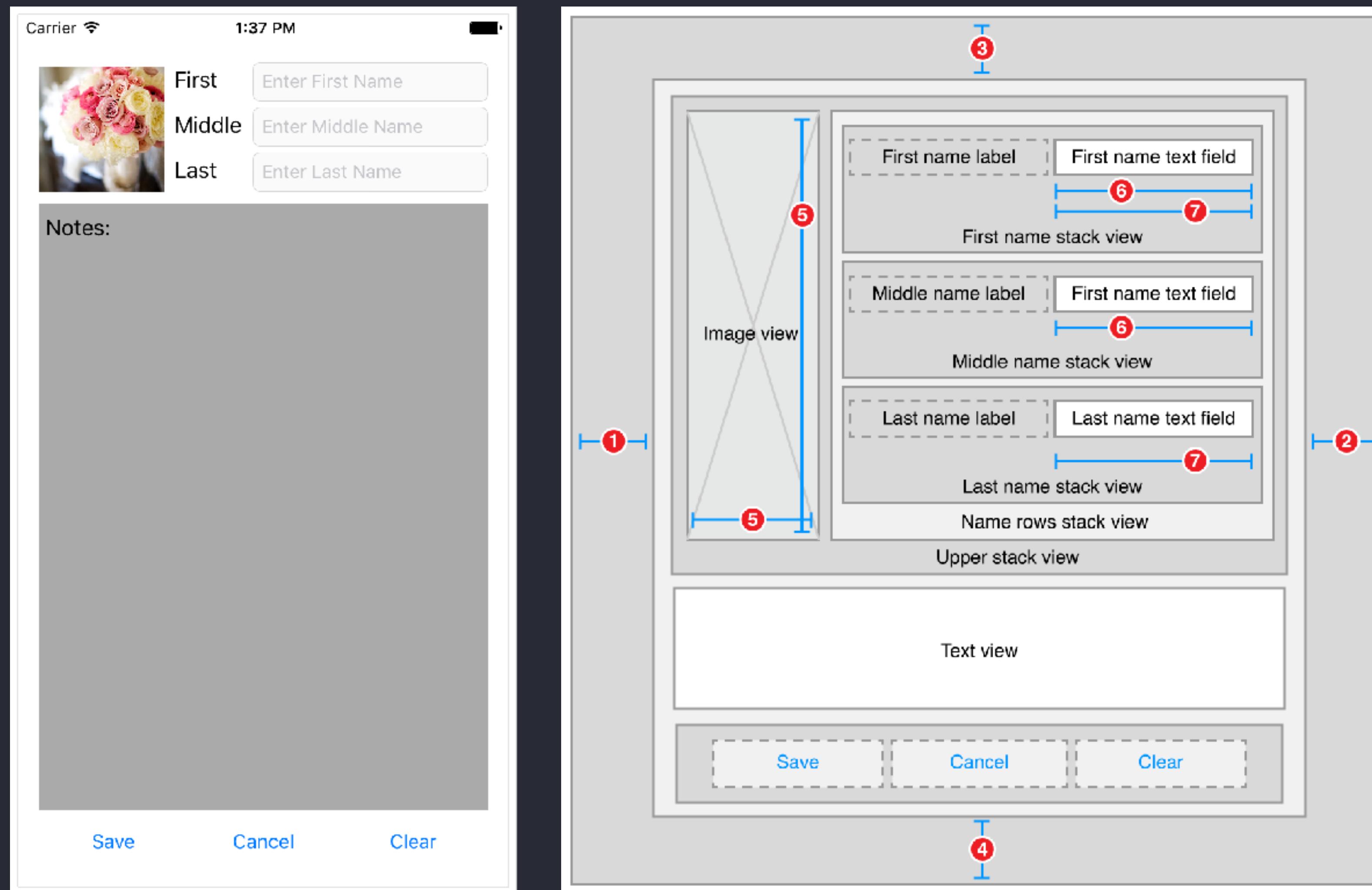
Anchors

UILayoutGuide  
(👉 empty views)

# Auto Layout

## UIStackView

# Auto Layout



# Auto Layout

```
let stackView = UIStackView()
stackView.axis = .vertical
stackView.distribution = .equalSpacing
stackView.alignment = .center
stackView.spacing = 8

stackView.addArrangedSubview(logo)
stackView.addArrangedSubview(title)
```

Split screen

Search APIs

Bitcode, App Slicing

Keyboard shortcuts

3D Touch  
(peek/pop & quick actions)

Auto Layout

UIAlertController

SFSafariViewController

On Demand Resources

Contacts.framework

# Thanks! ❤



@adumi, Trainline



[trainline.fr/jobs](http://trainline.fr/jobs)