from web to mobile

native mobile development vs. web development

Who am I ?

- Hy-tsoung Chang
- R&D and Software Engineer, 2004.
  - into everything : curiosity, technical
  - Internet Startups Internet : Exalead, Criteo..., lastly at Dashlane
  - Women in IT and Software Development

Why this talk ?

- If you want to launch a new product
  - en 2000 : web
  - en 2015 : web and mobile.

- For an IT professionnal, what are the differences between web et mobile development?

from web to mobile

web vs native mobile development

Web vs Native Mobile

- About :
  - Technologies
  - Projet management
  - Product positioning

Agenda

- Web/Mobile comparison on 4 subjects :
  - Programming
  - Releasing
  - Conception, UI & UX Design
  - Marketing & Business Strategies

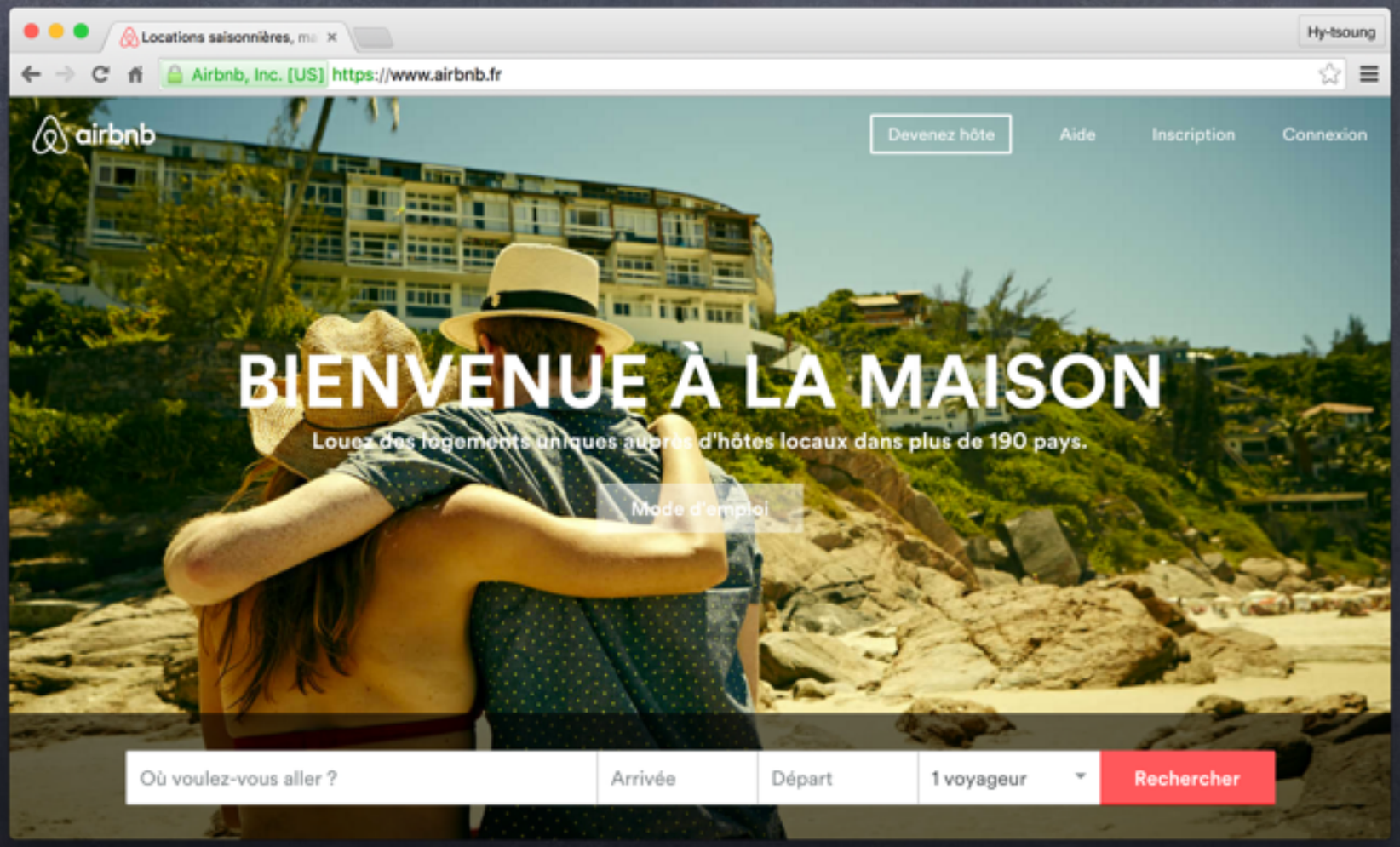
some definitions

- Mobile development :
  - native
  - iOS or Android (*)

(*) Technologies such as ReactNative, PhoneGap ou other hybrids : not directly mentionned... Nor Windows Phone.

some definitions

- Web developement:
  - websites or web applications

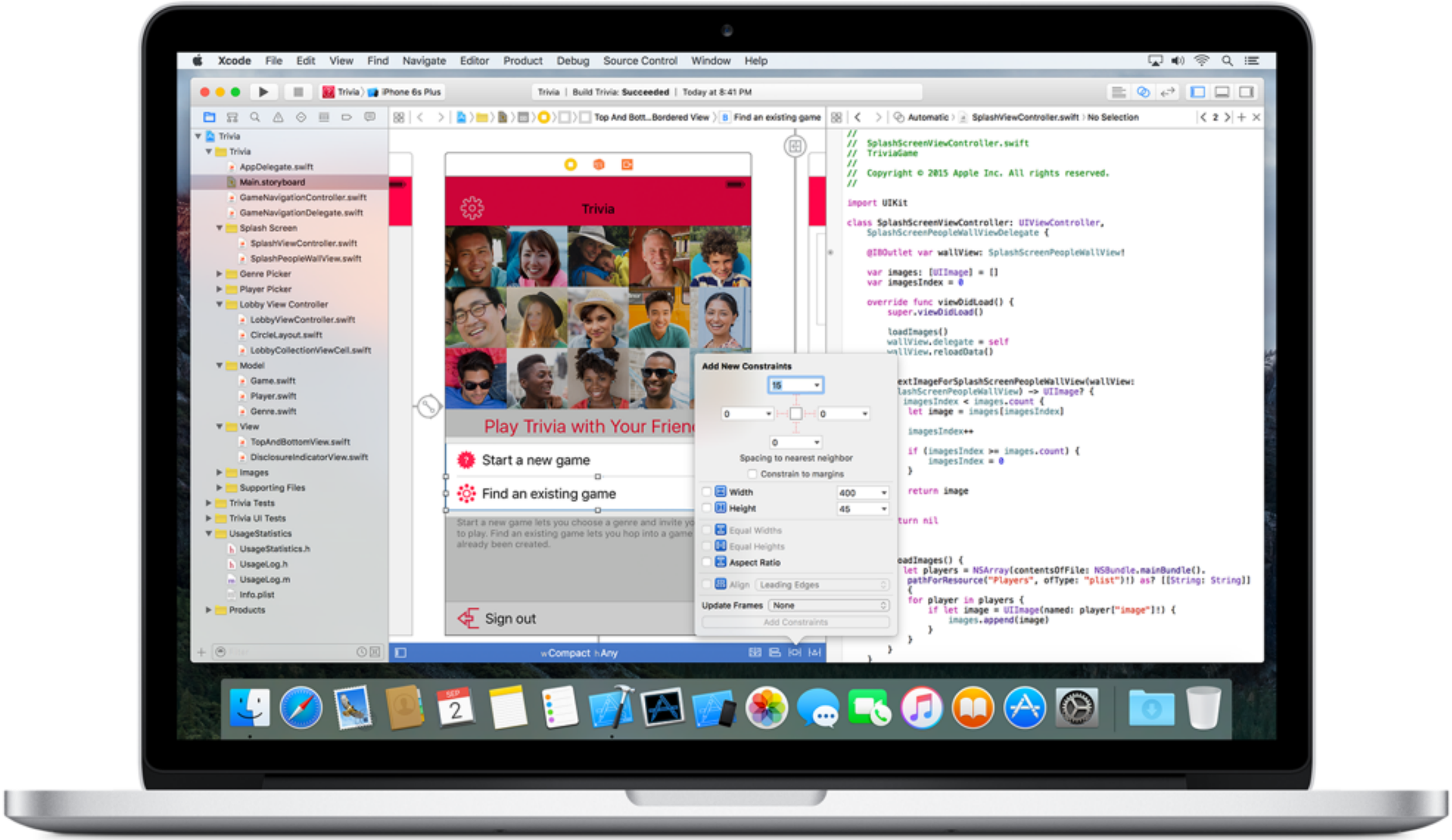


Programming

Tools

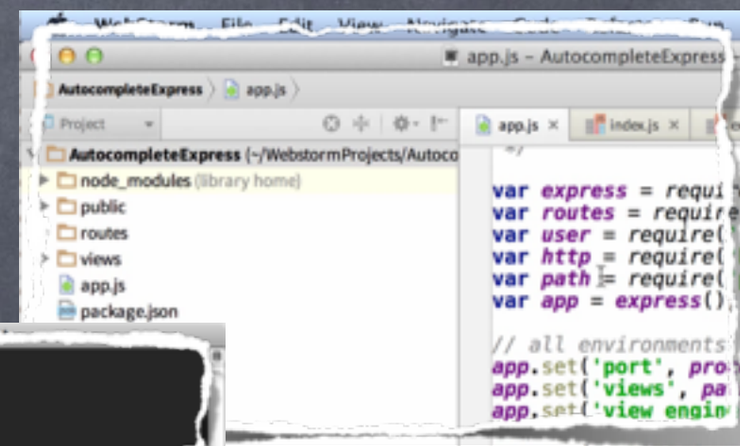
- Mobile
  - dedicated IDE : Xcode ou AndroidStudio.
- Web
  - Advanced text editors

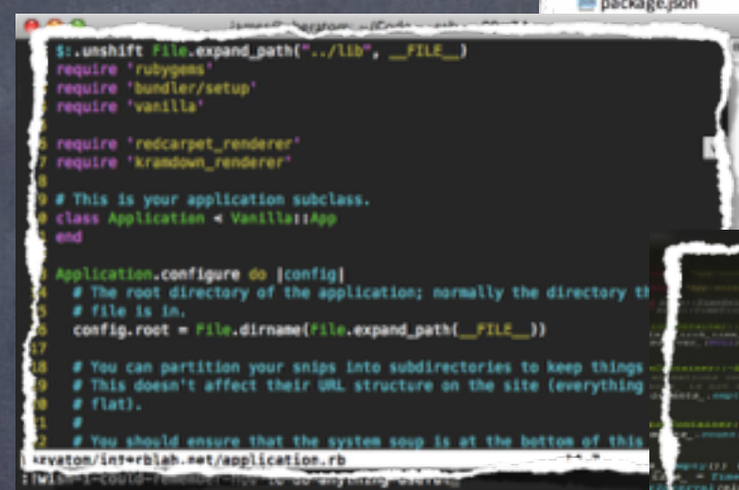
iOS : Xcode (Mac)

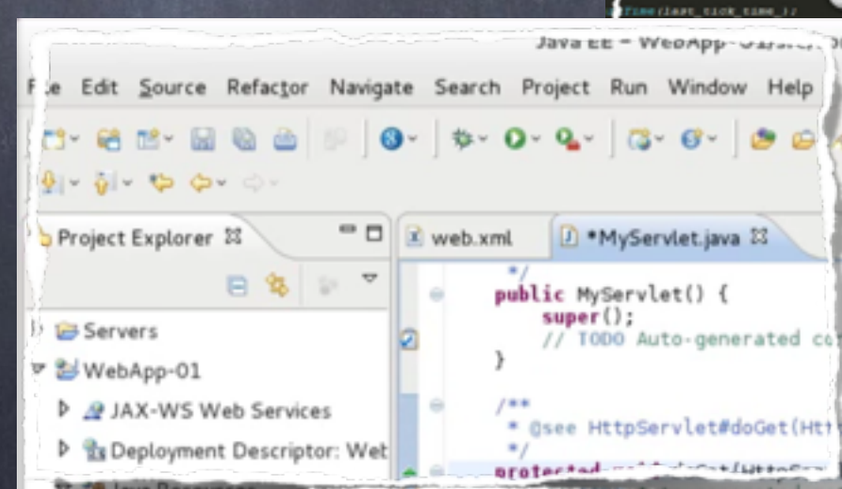


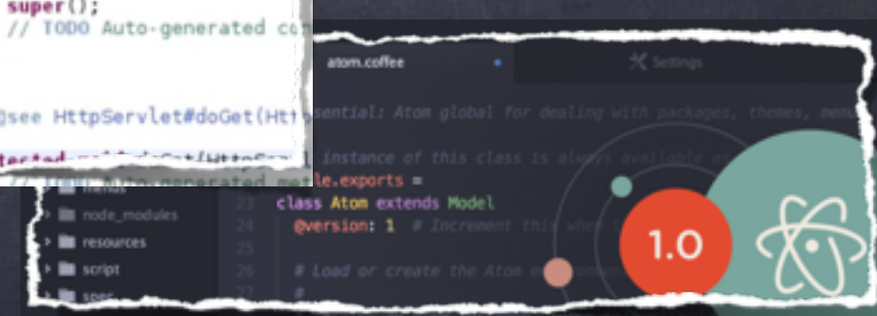
Web Development

- Large diversity
- Sublime, Atom, VI, Eclipse, WebStorm (JetBrains), ... etc.









Fondamental items to build a screen

- Mobile
  - iOS: objective-c/swift + storyboard
  - Android: java + .xml
- Web
  - html + css + javascript

Language details

- Mobile :
  - Object-oriented, official SDKs
  - Design Patterns for structuring
- Web :
  - Html, Javascript, Css, Php...
  - Scripting, no compilation, untyped.

Ex: Hello World

- code a view on mobile (iOS, objective-c)

```
@interface HelloWorld : UIView
@property UILabel *myLabel;
@end

@implementation
- (id)initWithFrame:(CGRect)frame {
   self = [super initWithFrame:frame];
   if (self) {
      _myLabel.text = @«Hello World!»;
      [self addSubview:_myLabel];
   }
   return self;
}
@end
```

Ex: Hello World

- code a web page (html)

```
<html>
  <body>
    <h>Hello World!</h>
  </body>
</html>
```

Frameworks and Ecosystems

- Actors on Mobile : Google and Apple.
  - 1 major OS release / year
- Actors on Web: developers, web community
  - New frameworks every year

Costs ?

	iOS	Android	Web
License	99 $	25 $	
Hosting			servers
Others...	% Apple, devices	% Google, devices	

develop then test ?

develop on mobile

PrintPhoto | Build PrintPhoto: **Succeeded** | Today at 16:36

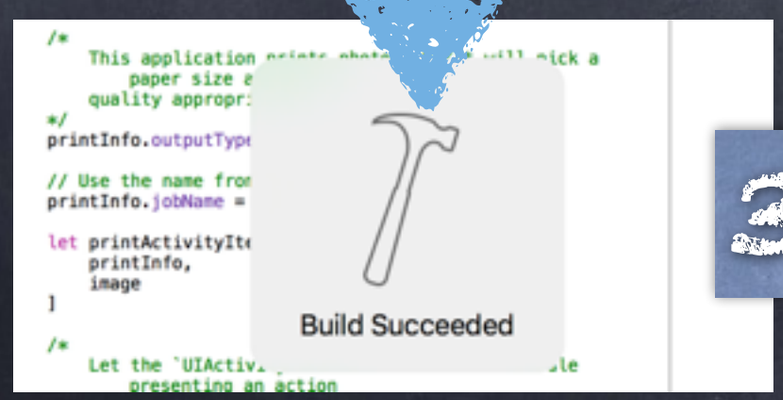
Pr...to 〉 CustomAssetPrintPageRenderer.swift 〉 CustomAssetPrintPageRenderer

- PrintPhoto
  - README.md
  - PrintPhoto
    - AppDelegate.swift
    - Standard Printing
      - StandardAssetPrintViewController.swift
    - Custom Asset Print Rendering
      - CustomAssetPrintViewController.swift
      - CustomAssetPrintPageRenderer.swift
    - Main.storyboard
    - Assets.xcassets
    - LaunchScreen.storyboard
    - Info.plist
  - Products

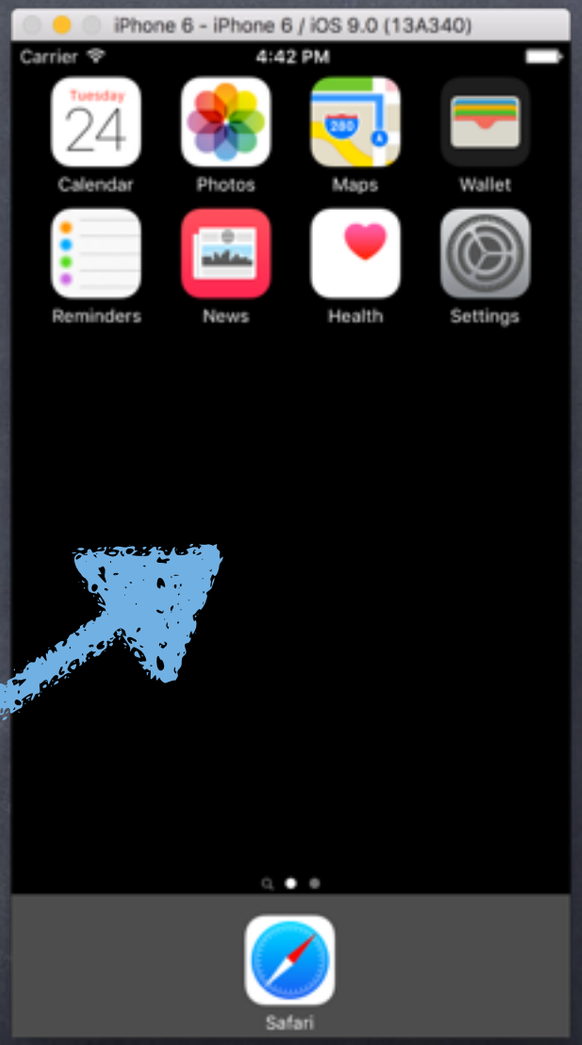
```swift
    override func drawPageAtIndex(pageIndex: Int, inRect printableRect: CGRect) {
        /*
            When `drawPageAtIndex(_:inRect:)` is invoked, `paperRect` reflects the
            size of the paper we are printing on and `printableRect` reflects the
            rectangle describing the imageable area of the page, that is the portion
            of the page that the printer can mark without clipping.
        */
        let paperSize = paperRect.size
        let imageableAreaSize = printableRect.size

        /*
            If `paperRect` and `printableRect` are the same size, the sheet is
            borderless and we will use the fill algorithm. Otherwise we will uniformly
            scale the image to fit the imageable area as close as is possible without
            clipping.
        */
        let fillsSheet = paperSize == imageableAreaSize

        let imageSize = image.size

        let destinationRect: CGRect
        if fillsSheet {
            destinationRect = CGRect(origin: .zero, size: paperSize)
        }
        else {
            destinationRect = printableRect
        }

        /*
            Calculate the ratios of the destination rectangle width and height to
            the image width and height.
        */
        let widthScale = destinationRect.width / imageSize.width
        let heightScale = destinationRect.height / imageSize.height

        // Scale the image to have some padding within the page.
```

1

2

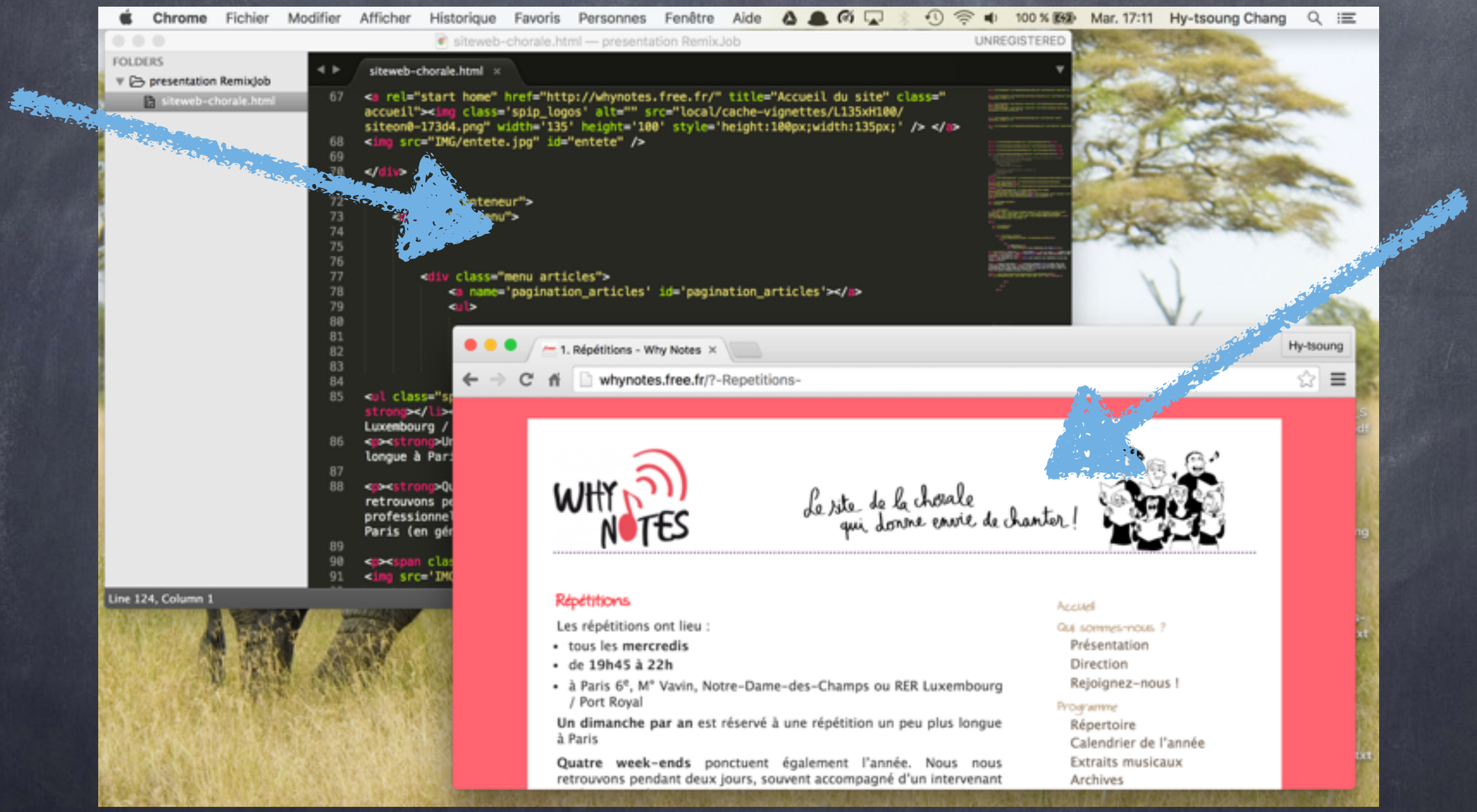


3



a few minutes

develop on web



a few seconds

Fragmentation on « target platforms »

	iOS	Android	Web
target platforms specificities	iOS 7.0+ 20 models	Models, Constructors, Updates	Standards, Browsers, Updates
Need specific code ?	No	No...	Yes
Difficulty	+	++	++

Release

New version, ready to deploy !

WEB deploy

Ready

↓

Staged rollout

↓

Full deploy

↓

Rollback

a few min /
a few hours

Android deploy

Ready

↓

Staged rollout

↓

Full deploy

↓

« Rollback » ?

a few hours /
a few days

iOS deploy

* « Expedite review » is possible

on iOS => finger crossed for iOS !

Agenda

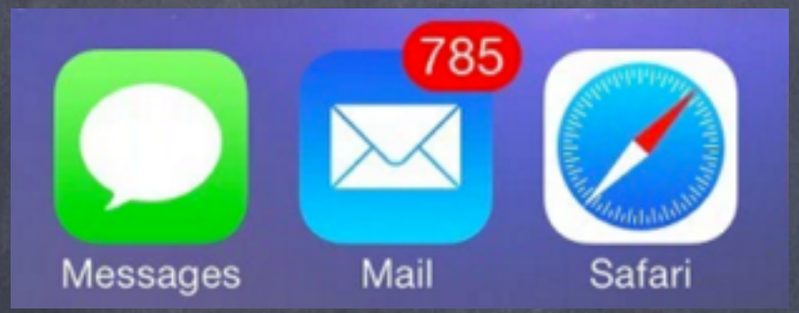
- Web/Mobile comparison on 4 subjects :
  - ~~Programming~~
  - ~~Releasing~~
  - Conception, UI & UX Design
  - Marketing & Business Strategies

Product Specs

User with a good connection or not ?

Improve user reach

- Mobile



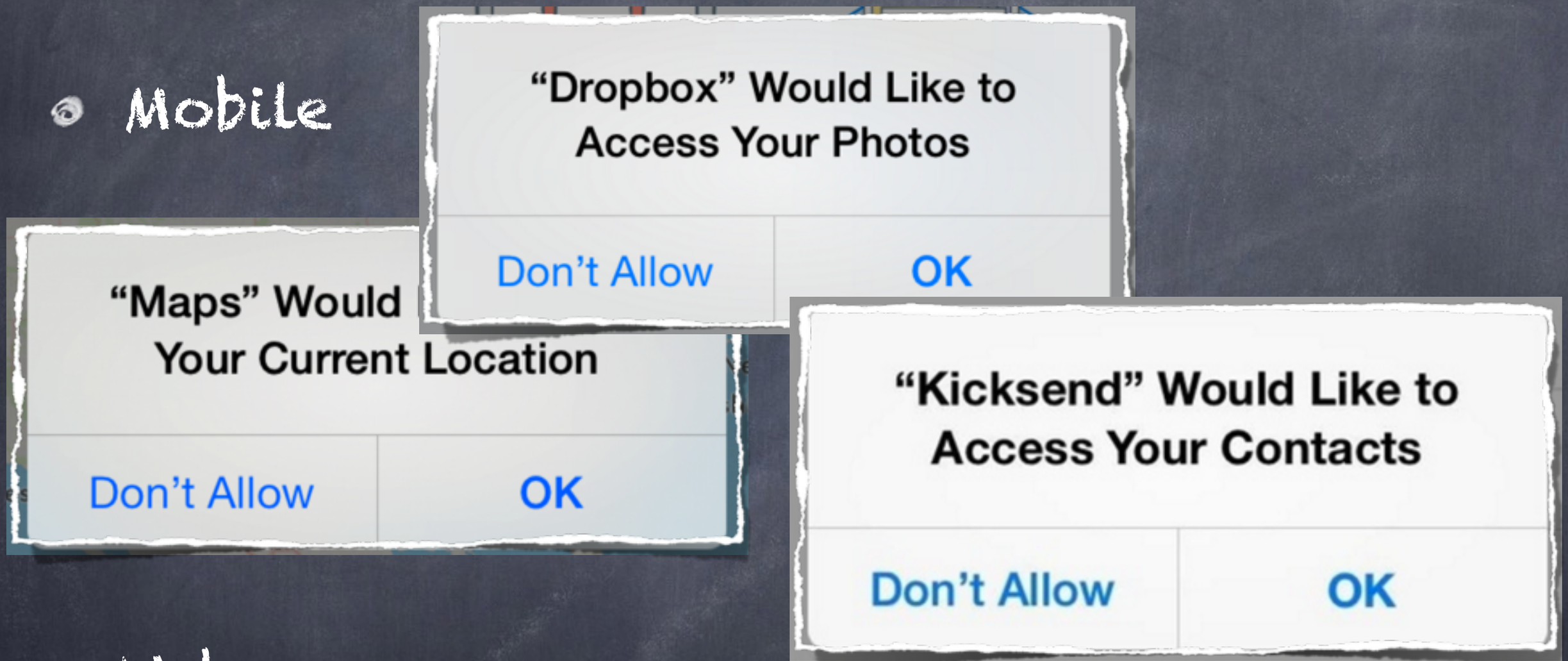
- Web
  - User has to go back to the website

Multiple Interactions

- Mobile
- Web !

Access to personal data

- Mobile

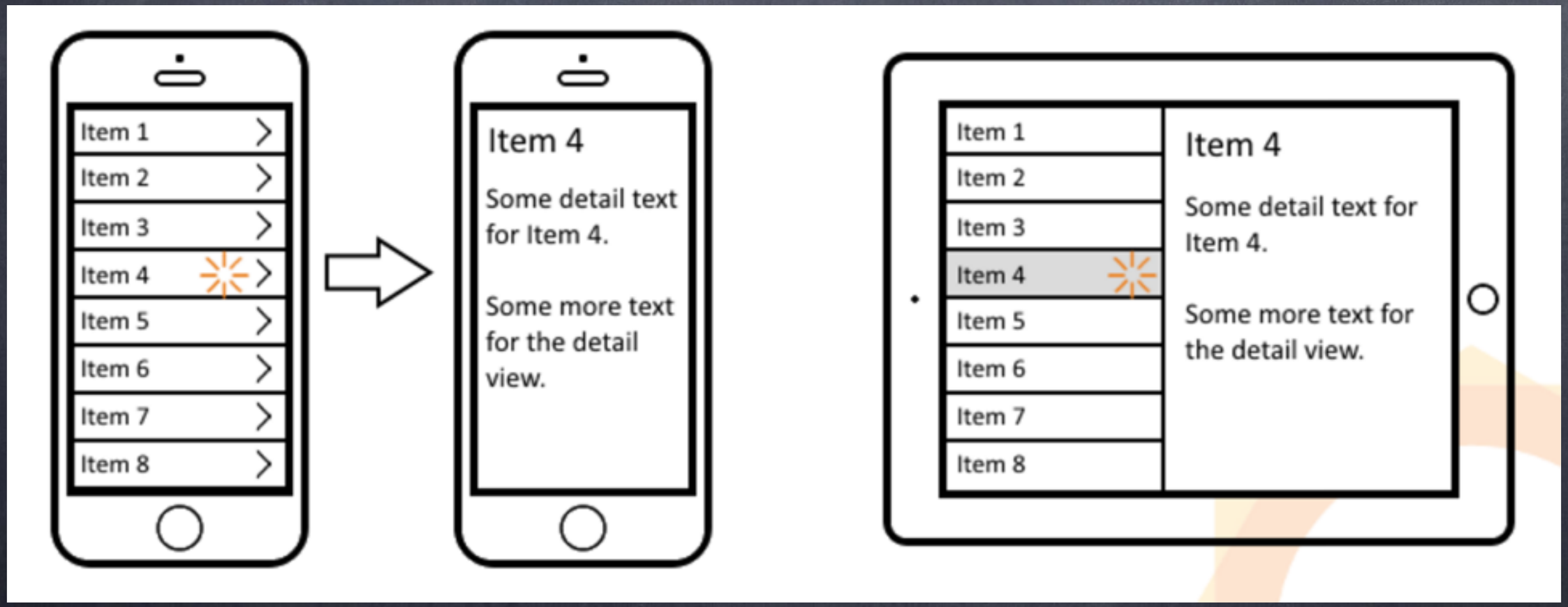


- Web
  - A bit of geolocalisation

UI & UX Design

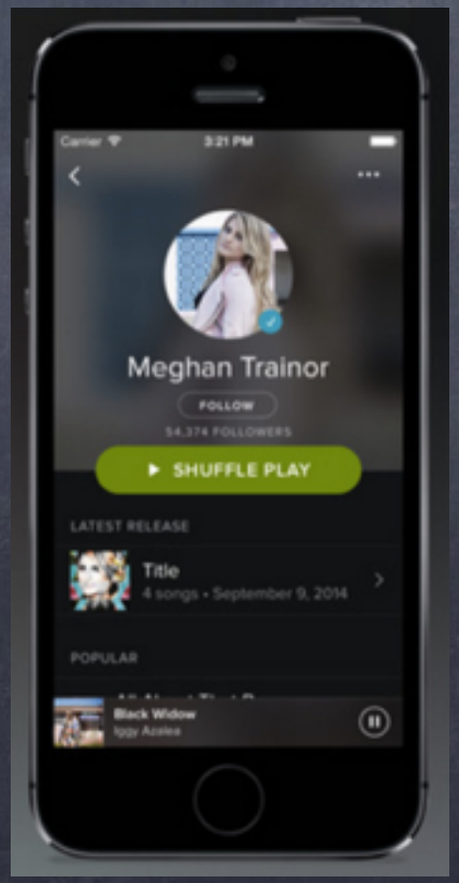
Mobile - Spatial Layout

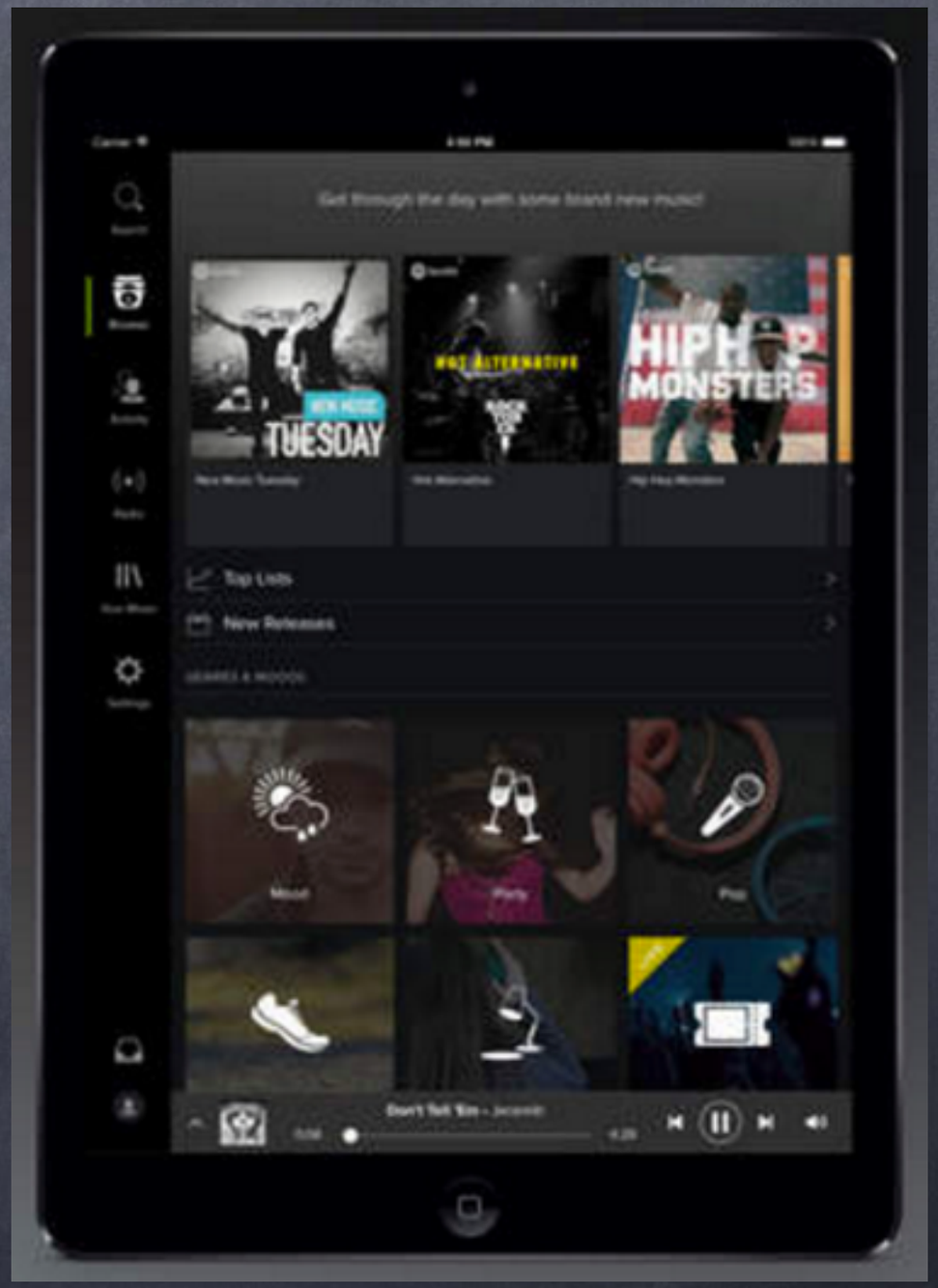
- Challenge on smartphones



Mobile UI

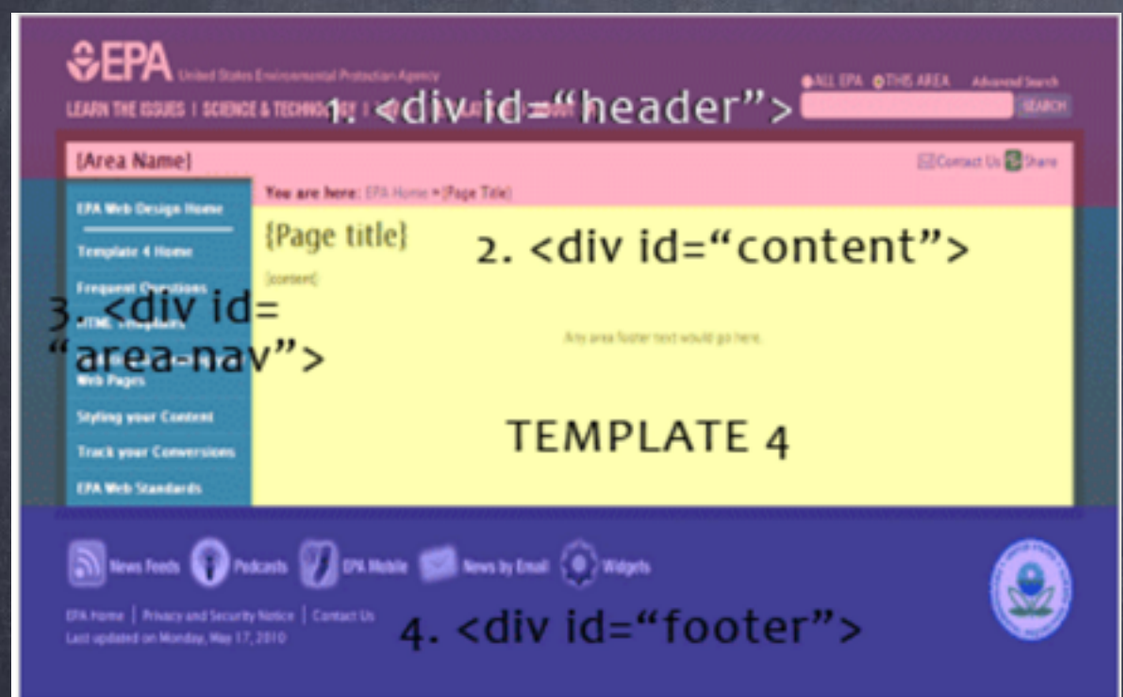
- Mobile : size, rotation, resolution ?
- APIs and SDKs





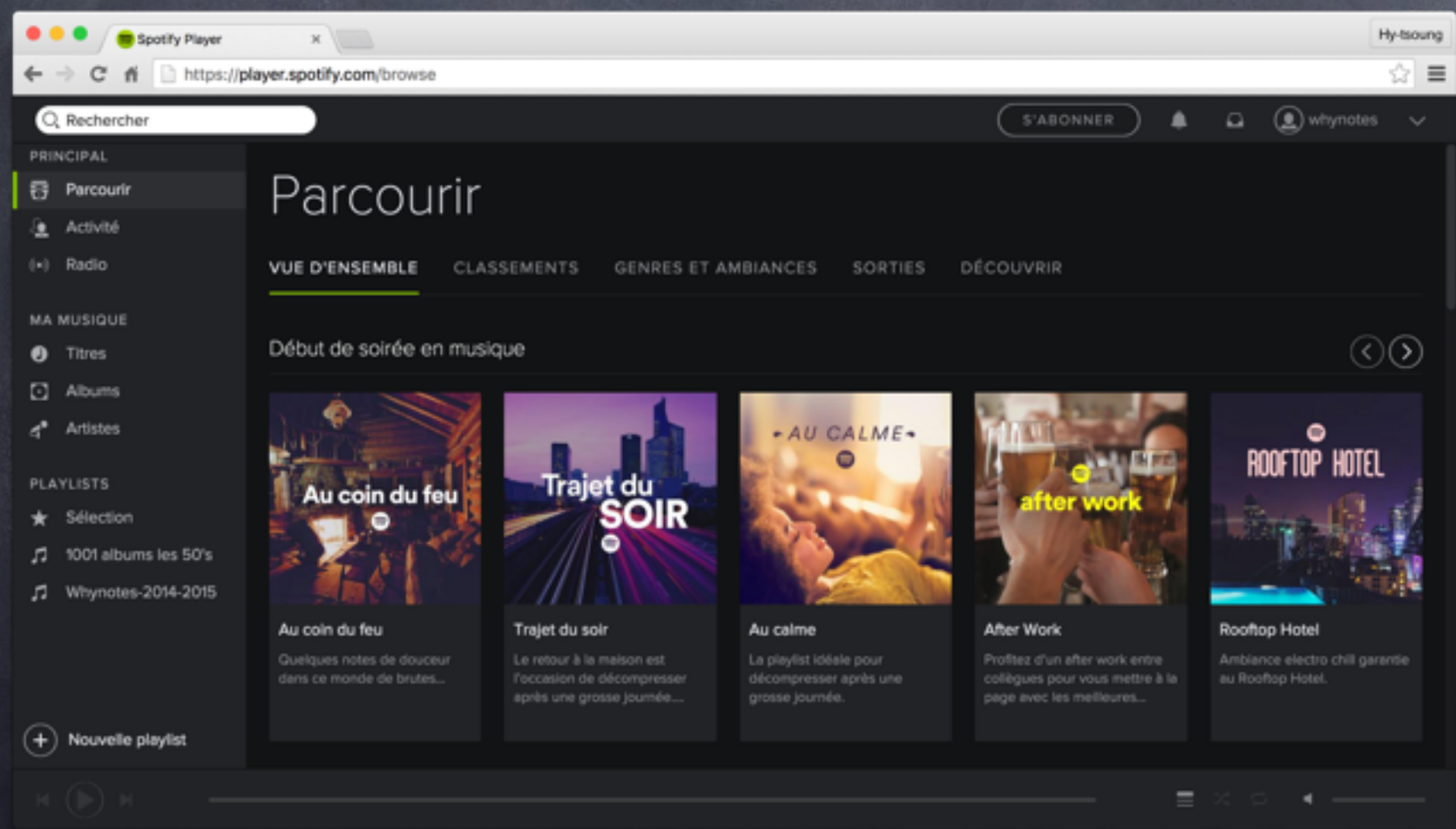
Web - Spatial Layout

- pretty standard layouts, mature



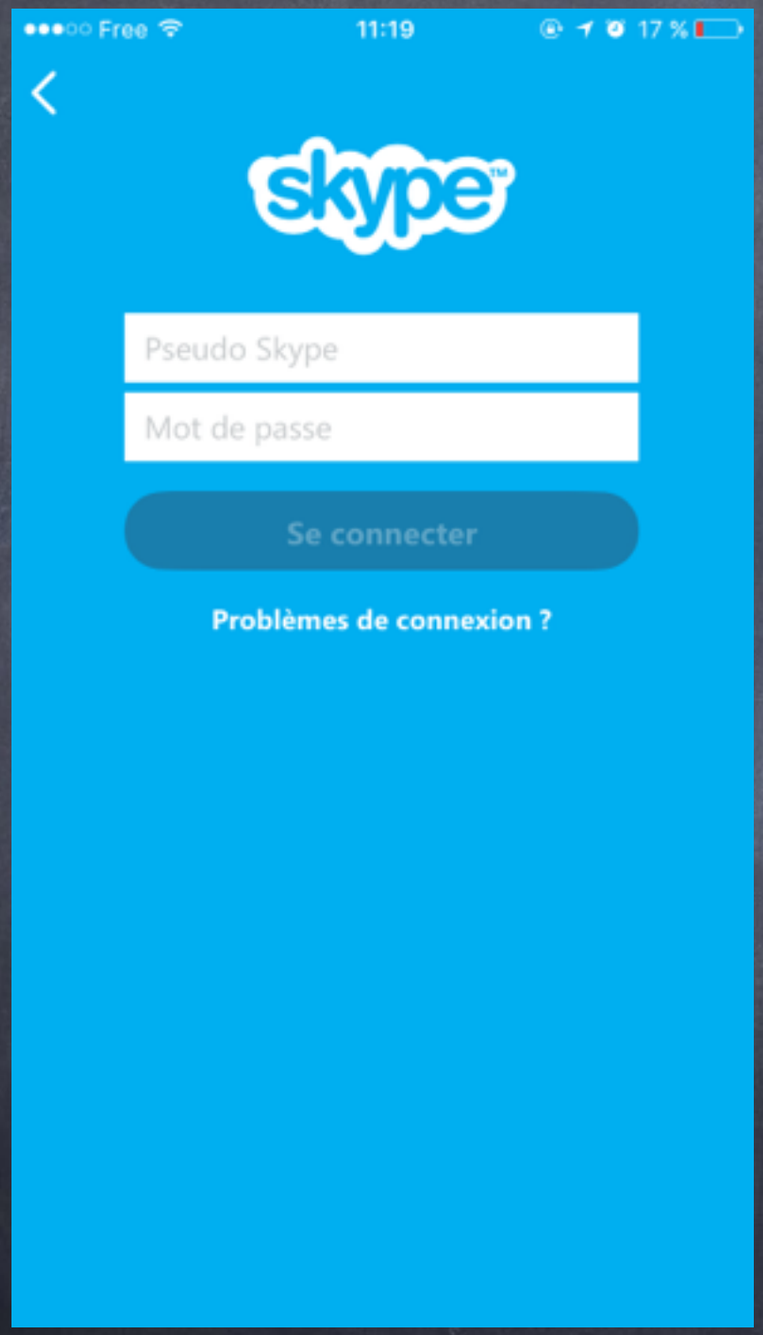
Web UI

- Web : for small screens ? Nice-to-have



- « Responsive UI »...

Ex : keyboard management on mobile



Design and Visual Identity

- Mobile :
  - Guidelines !
  - vs specific Branding ?
- Web :
  - Ergonomy.
  - Independent websites bring novelty

Product Marketing

Separate positioning on Web and Mobile

- For a given product or brand...
  - ex: Expectations ?
  - ex: Interactivity ?
  - ex: Institutional content ? Practical informations ?

Challenges on Launch

- Mobile
  - App must be downloaded
  - App must be opened
- Web
  - Website must be reachable via search
  - Web reputation

Metrics

- Mobile
  - Reviews et ratings on the store
  - Relationships with Google or Apple to obtain the « Editor's Choice »
- Web
  - SEP Optimization

Conclusion

And for tomorrow ?

Development costs

- Mobile Product from a Web Product:
  - start from scratch
- Mobile market is thriving
  - Few trainings vs lots of evolutions

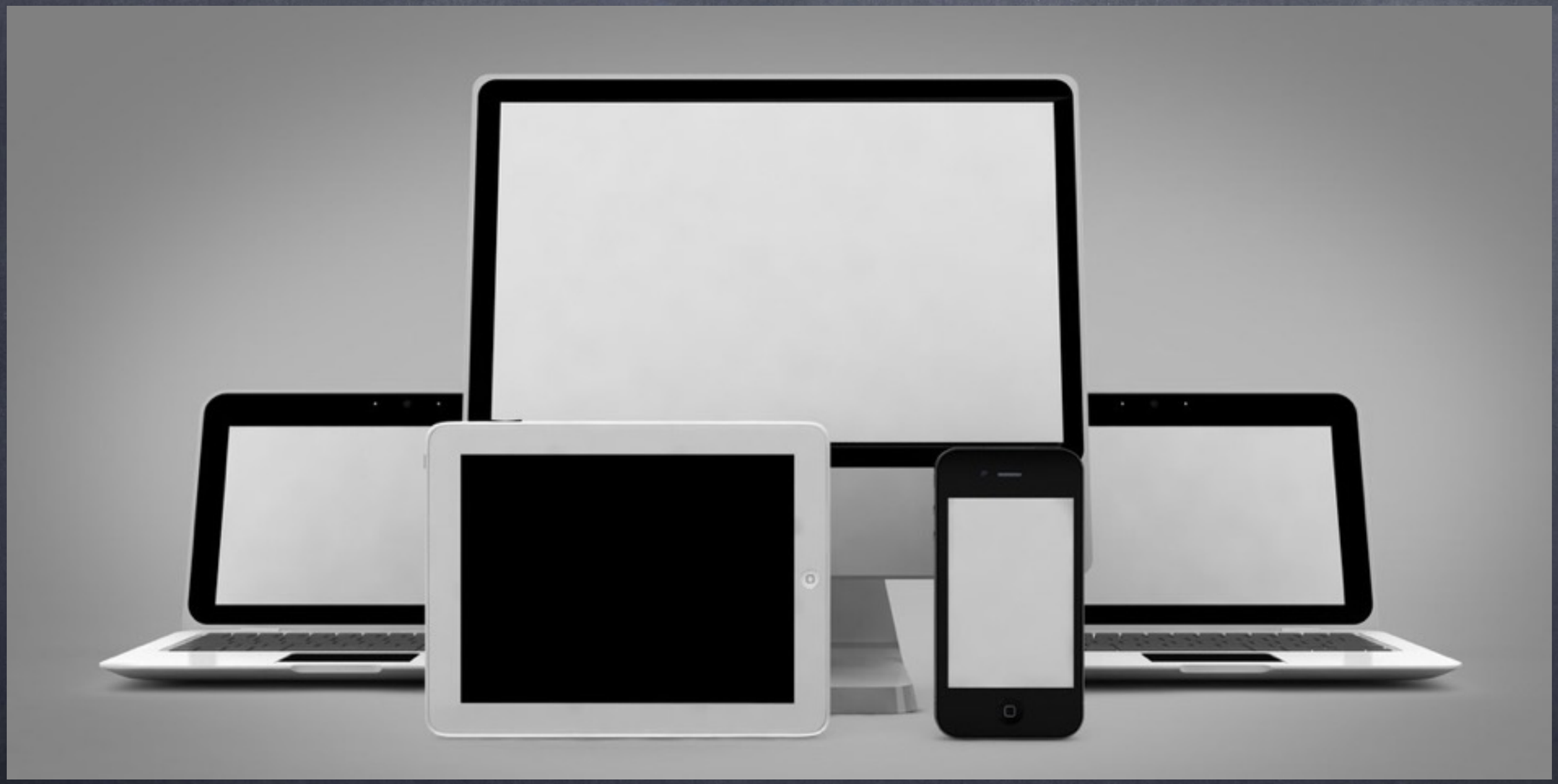
Other initiatives hybrids web/mobile

- PhoneGap... :
  - javascript encapsulated in webview
- ReactNative :
  - javascript compiled into native code

Web is adapting, and counter-attacks

- ReactNative <- Facebook (web)
- Responsive UI
- Offline Web Application (Google Docs)
- Javascript : more mature and structured
- Html 5.

Thanks



Hytsoung Chang

Annexes

Android : AndroidStudio

MessagingService > Application > src > main > res > layout > fragment_message_me.xml

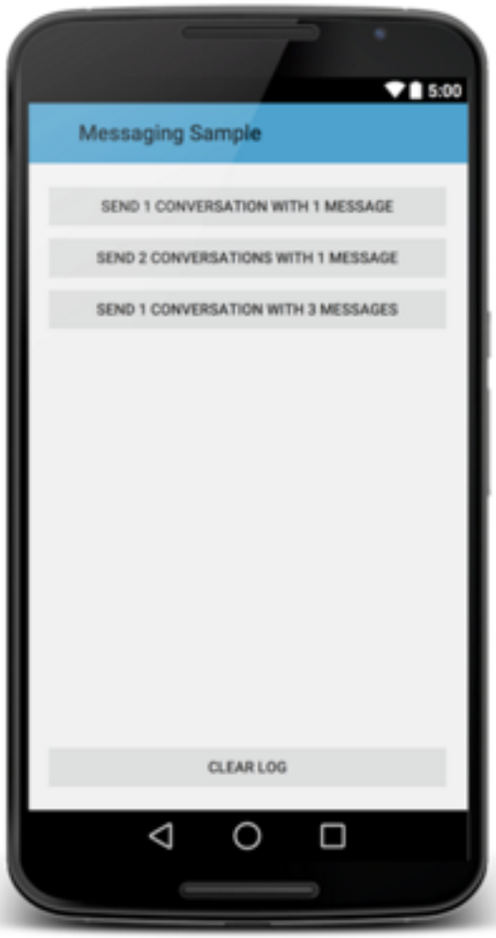
Application
- manifests
- java
  - com.example.android.messagingservice
    - Conversations
    - MainActivity
    - MessageLogger
    - MessageReadReceiver
    - MessageReplyReceiver
    - MessagingFragment
    - MessagingService
- res
  - drawable
  - layout
    - activity_main.xml
    - fragment_message_me.xml (2)
  - values
  - xml
- Gradle Scripts

fragment_message_me.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp">

    <Button
        android:id="@+id/send_1_conversation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Send 1 conversation with 1 message"/>

    <Button
        android:id="@+id/send_2_conversations"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Send 2 conversations with 1 message"/>

    <Button
        android:id="@+id/send_1_conversation_3_messages"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Send 1 conversation with 3 messages"/>

    <TextView
        android:id="@+id/data_port"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:scrollbars="vertical"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/clear"
        android:text="Clear Log"/>

</LinearLayout>
```

Design | Text

Preview: Nexus 6 · AppTheme · 21



TODO · 6: Android · Terminal · Event Log · Gradle Console · Memory Monitor

Beta-tests

- Mobile
  - Android : Beta Channel (Google)
  - iOS : TestFlight (Apple), HockeyApp (3rd party)
- Web
  - Beta test hébergé sous contrôle interne complet
  - A faire soi-même

Hello World

- version mobile Android

```
public class HelloWorld extends Activity {

   @Override public void onCreate(Bundle savedState) {
      super.onCreate(savedState);
      RelativeLayout layout = new RelativeLayout(this);
      TextView text = new TextView(this);
      text.setText(«Hello World!»);
      relativeLayout.addView(text);
      setContentView(layout)
   }
}
```

SDK et frameworks

- Mobile:
  - SDK définis par Apple ou Google Android. Ils sont la référence.
- Web:
  - De nombreux frameworks open-source. Adoption par la communauté au fur et à mesure

Outils WYSIWYG

- Mobile
  - iOS : storyboard puis drag&drop vers code logique
  - Android : similaire
- Web
  - pas d'outil WYSIWYG « approuvé »

Comparaison avec une application desktop

- Application Desktop <-> Application mobile
  - Le mobile est une application classique avec des spécificités en plus
    - stockage sur machine locale, programme local
- Application Desktop <-> Application web
  - le web a commencé à faire des applications très récemment
  - fonctionnalités locales mais stockage online (?)

Mise en production

- Mobile
  - iOS : soumission de la version à l'AppStore et attente de la validation => ~2 semaines - 1 mois.
  - Android : upload et publication sur PlayStore => accessible en quelques heures - 1 journée.
- Web
  - déployer le code sur le(s) serveur(s) => quasi immédiat
- ...A prendre en compte pour le lancement d'un nouveau produit ou une mise à jour !

Mise en prod de correctifs

- Mobile: devoir envoyer une nouvelle version et passer par la soumission, validation, release publique.
- Web: déployer le correctif, fin.

Quels problèmes de scalabilité ?

- Mobile :
  - le problème se pose moins, car consommation ressources matérielles essentiellement sur devices, donc calibrage en amont.
- Web :
  - si trop de trafic : ajouter des nouveaux serveurs et re-architecturer la prochaine version.

Utilisateur avec une connection réseau ou non ?

- Mobile : prévoir l'expérience offline, et/ou un réseau lent.
  - Ex: téléchargement au préalable et/ou en arrière plan.
  - Application avec utilité offline
- Web : n'a pas de sens...
  - Même si Google Docs offline

Examples de controverses

- Mobile :
  - Menu principal rétractables à gauche
  - Fenêtres qui se « placent » les unes sur les autres.
- Web :
  - Barre de menu toujours accessible
  - Pages web sur un même template en général.

Interactions supplémentaires sur le mobile

- Interagir autrement avec l'utilisateur :
  - Pinch, swipe, 3D touch, secouer l'appareil, le retourner, faire vibrer, sonner, etc.
- Web: (souris)

Accès à des données personnelles de l'utilisateur

- Mobile : transferts de données perso
  - Carnet d'adresses, photos, localisation GPS.
- Web : outils en émergence
  - proposition d'UX améliorée via géolocalisation.

Indicateurs clés

- En général :
  - Nombre d'utilisateurs uniques récurrents (par semaine, par mois), nombre d'inscrits.
- Mobile :
  - Nombre de téléchargements
- Web :
  - trafic

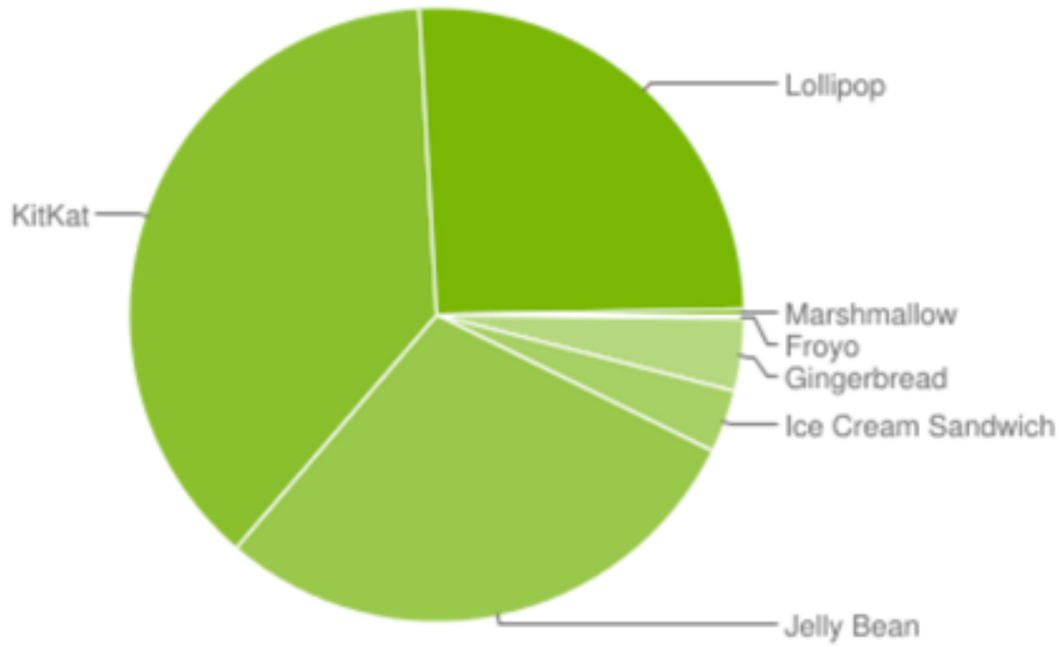
Web vs Mobile

- Défi actuel : Création d'un nouvel usage à mi-chemin entre une application personnelle et l'univers Internet

This section provides data about the relative number of devices running a given version of the Android platform.

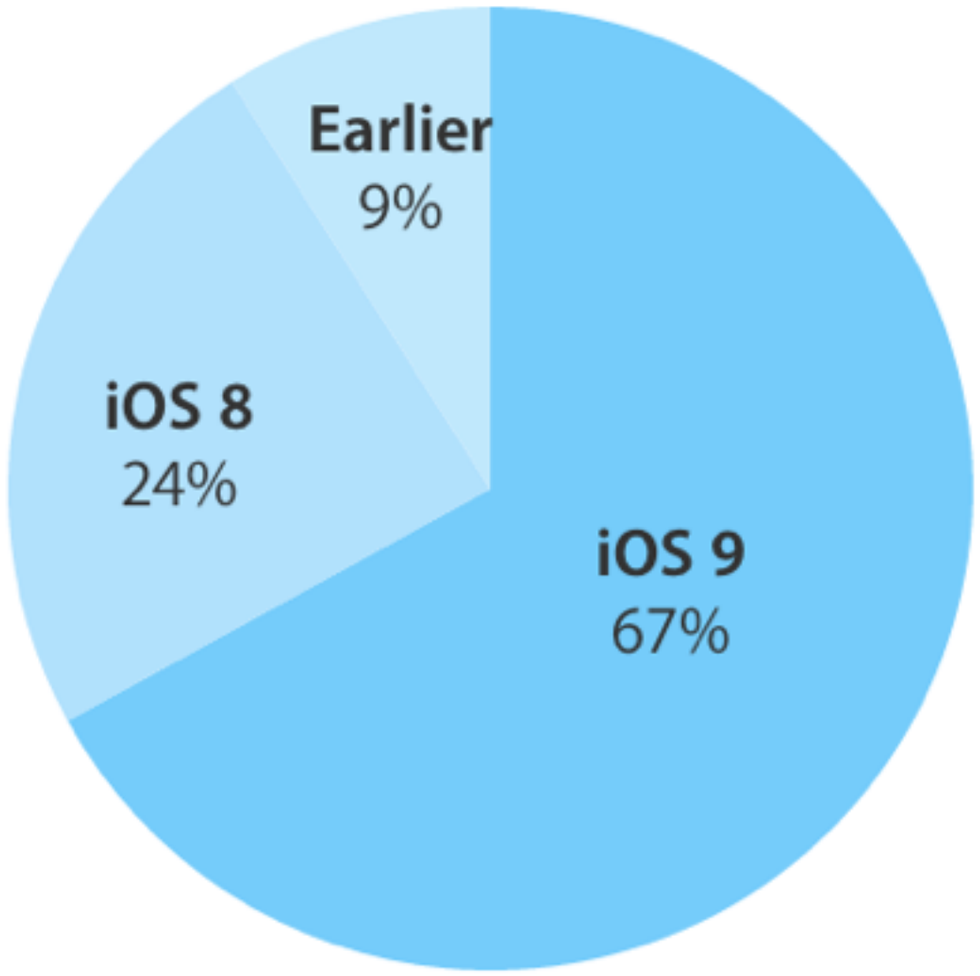
For information about how to target your application to devices based on platform version, read Supporting Different Platform Versions.

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	3.3%
4.1.x	Jelly Bean	16	11.0%
4.2.x		17	13.9%
4.3		18	4.1%
4.4	KitKat	19	37.8%
5.0	Lollipop	21	15.5%
5.1		22	10.1%
6.0	Marshmallow	23	0.3%



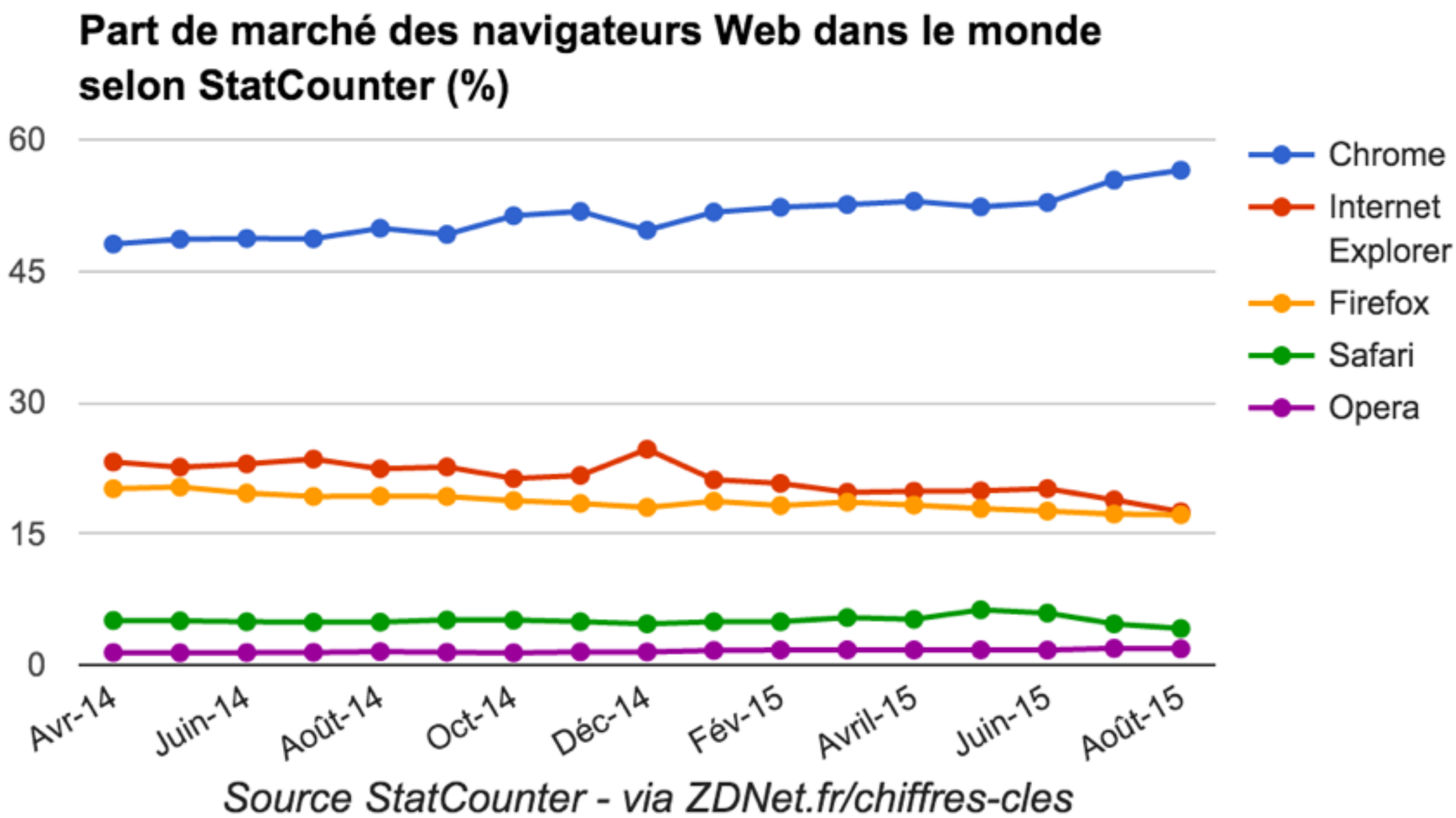
Data collected during a 7-day period ending on November 2, 2015.

67% of devices are using iOS 9.

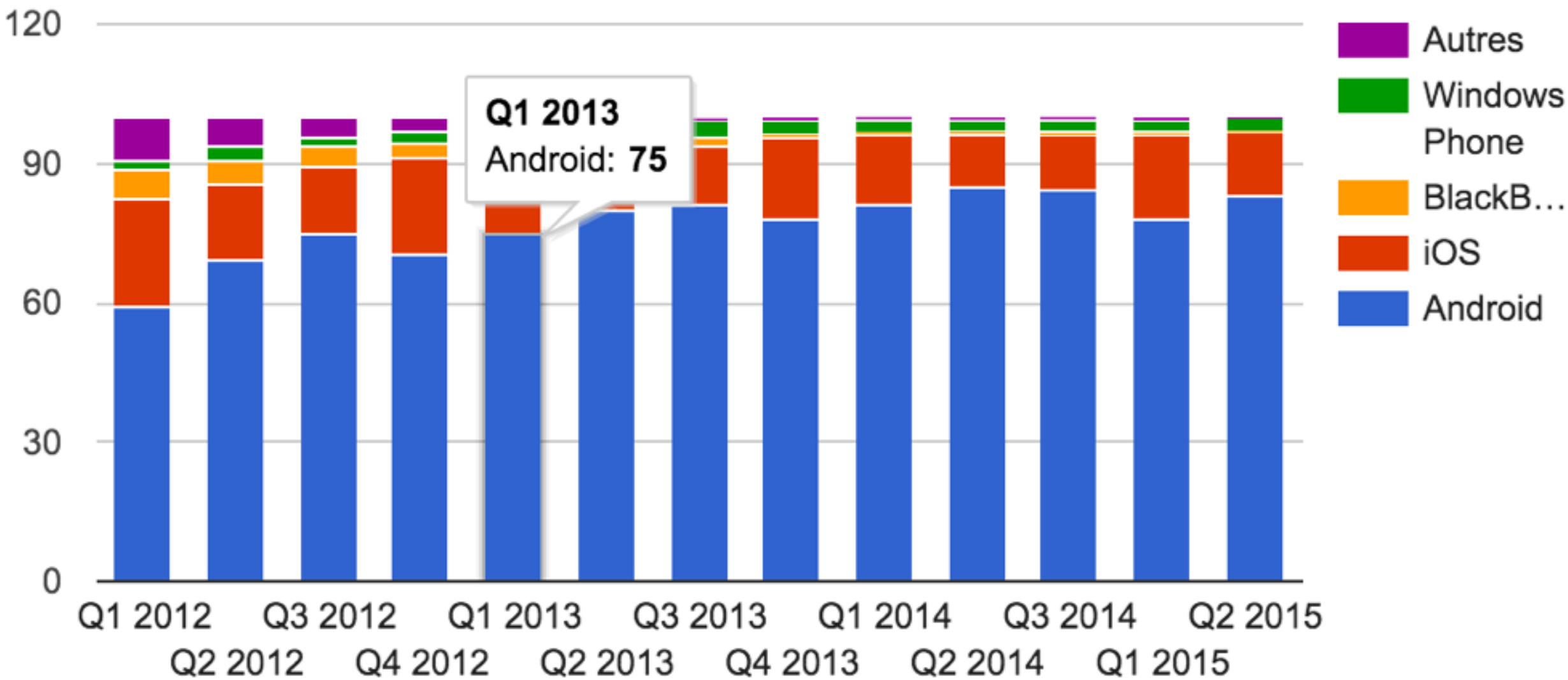


As measured by the App Store on November 16, 2015.

Navigateurs dans le monde et en Europe (2015)



Parts de marché mondiales des OS de smartphones de 2012 à 2015 (%)



Source IDC - via ZDNet.fr/chiffres-cles

Monitorer la version ?

- Par ordre d'importance :
  - Sur mobile
    - Crash reports
    - Reviews utilisateurs
    - Logs (s'il y en a)
  - Sur web
    - Exception logs, Logs (s'il y en a)

UI sur différentes tailles d'écran

- Web : « Responsive UI »...

Mobile : Bataille d'influence

- Acteurs économiques et technologiques avec un poids importants :
  - Android <- Google (!)
  - Apple
- Et des développeurs mobiles sous la dépendance de Google/Apple.

Monétisation

- Mobile :
  - prix bas ou achats intégrés
- Web :
  - Offres/Souscriptions premium
  - Financement via publicités