

DÉVELOPPEMENT SOUS IOS

PREMIÈRE APPLICATION



SWIFT / OBJECTIVE-C



TI.A04

Introduction

- Installation de XCode

- Environnement iOS

- Licences

- Distribution

Environnement de développement

- Découverte de XCode

Objective-C / Swift

- Les classes

- Frameworks

- Création d'une interface graphique

- Connexion à un capteur externe via BLE

INTRODUCTION

ETAPES

- Ordinateur Apple
- Crédit d'un Apple ID
- Enregistrement compte développeur
- Chargement et installation de XCode



CRÉATION D'UN APPLE ID (I)

- Web (à préférer)
- iTunes (nécessite la saisie d'une carte pour créer un compte)
- Indispensable pour l'utilisation des services et appareils Apple



The screenshot shows the "Identifiant Apple" creation page. At the top, there's a navigation bar with links for Mac, iPad, iPhone, Watch, TV, Music, Assistance, and search functions. Below the navigation is a header with the text "Identifiant Apple" and links for "Connectez-vous", "Créer votre identifiant Apple", and "Q&A". A dark banner below the header contains the text "Identifiant Apple ou mot de passeoublié ?". The main content area features the heading "Votre compte pour tout l'univers Apple." followed by the text "Un seul identifiant Apple et un seul mot de passe vous donnent accès à tous les services Apple." and a link "En savoir plus sur l'identifiant Apple". Below this are icons for various Apple services: iCloud, App Store, iTunes Store, iMessage, FaceTime, App Store, iTunes Store, and iBooks. At the bottom of the page is a "Créer votre identifiant Apple" button.



The screenshot shows the iTunes Store creation page. At the top, it says "Bienvenue dans l'iTunes Store". Below that is a large image of a computer screen displaying the iTunes interface. To the right, text explains that an Apple ID allows users to download the latest music, video, and more. It includes a "Continuer" button and a note about clicking it to create the Apple ID. At the bottom are "Annuler" and "Continuer" buttons.

iTunes

CRÉATION D'UN APPLE ID (2)

Web

iTunes

CRÉATION D'UN APPLE ID (2)

Conditions générales et Engagement de confidentialité d'Apple

Conditions générales

Vous devez accepter ces Conditions générales afin de pouvoir utiliser ce service.
Si le pays de vos données de facturation n'est pas la France, cliquez ici.

Informations importantes sur les téléchargements de l'iTunes Store
Les téléchargements effectués depuis l'iTunes Store ou iTunes (iTunes, iPhone, iPod, iPad, Mac ou PC) utilisant une technologie en temps réel nommée FairPlay.

- Les téléchargements effectués depuis l'iTunes Store utilisant la technologie FairPlay peuvent être utilisés sur un nombre maximum de cinq ordinateurs, cinq iPhone ou iPod, cinq iPad ou iPhone/iPad combinés. Les téléchargements effectués depuis l'iTunes Store peuvent être utilisés uniquement sous votre propre iPhone/iPad jusqu'à la version 2.0 ou plus, ou avec une version préliminaire à la version 3.2 ou plus des logiciels effectués via FairPlay.
- Tous les téléchargements audio FairPlay par l'intermédiaire d'un lecteur disc sont convertis en téléchargements audio FairPlay par défaut au moment de leur achèvement.
- Les CDs audio peuvent être utilisés de la même manière que la plupart des CDs audio achetés dans le commerce. Les vidéos FairPlay sauvegardées sur un disque ne peuvent être lues que par l'intermédiaire d'un iPhone/iPad.
- Les téléchargements effectués depuis iTunes Plus ou iTunes Radio ont la meilleure qualité de son vis à vis des autres marques de FairPlay. Les téléchargements iTunes Plus peuvent être sauvegardés, copiés ou gravis dans la mesure raisonnablement nécessaire pour une utilisation à des fins privées et non-commerciales.
- Tous les fichiers musicaux et de divertissement que vous achetez sur iTunes, Apple TV, iPod, Mac ou iPhone/iPad. Vous pouvez télécharger un fichier en téléchargement dans les 30 jours suivant l'achat, sans aucun délai de 30 jours. Ainsi, les téléchargements pour streaming et visionnage sur iTunes ou 48 heures pour le visionnage autant de fois que vous le souhaitez une fois que vous avez payé.
- Certains contenus peuvent être téléchargés et écoutés une fois que vous avez acheté l'appareil. Si vous achetez un appareil avec Contenu déjà chargé, certains contenus peuvent être utilisés immédiatement sans avoir à l'acheter à nouveau.

Je lu et j'accepte les Conditions générales.

[Version imprimable](#)

[Retour](#) [Annuler](#) [Accepter](#)

Web

iTunes

CRÉATION D'UN APPLE ID (2)

My Apple ID

Create an Apple ID.

My Apple ID gives you complete access to your Apple experience, including the App Store, iTunes Store, iBooks Store, iCloud, and more. Plus it's free.

Information about services available to you.

Realize Apple Privacy Policy

Create a security question.

It's a great idea to set up a security question that helps verify your identity if you forget your password.

Security Question:

Name:

Select your location.

Choose where you'd like to receive shipping notifications and promotional offers.

Location:

Emergency contacts.

Please enter your emergency contact information.

First Name:

Middle Name:

Last Name:

Preferred language.

Please enter your preferred language.

Language:

Country/Region:

Please enter your country or region.

Country:

City:

State/Province:

Zip/Postal Code:

Preferred currency.

Please enter your preferred currency.

Currency:

Consent to Terms and Conditions.

By clicking "Create Apple ID", you agree to the [Terms of Service](#) and [Apple Privacy Policy](#).

I accept the terms and conditions.

Please enter the characters present in the image below.

I have read and agree to the [Terms of Service](#) and [Apple Privacy Policy](#).

I have read and agree to the [Terms of Service](#) and [Apple Privacy Policy](#).

Web

iTunes

Conditions générales et Engagement de confidentialité d'Apple

Saisissez les informations de votre identifiant Apple

Vous devez accepter des Conditions générales afin de pouvoir utiliser ce service.
Si le pays de vos données de facturation n'est pas la France, cliquez ici.

Informations importantes sur les téléchargements de l'iTunes Store.

• Les téléchargements effectués à partir du iTunes Store, à l'exception de l'application iPad Photo, utilisent une technologie nommée

E-mail:

Mot de passe: **Confirmer le mot de passe:**

Sécurité

Ces questions nous permettront de vérifier votre identité et de protéger vos données personnelles.

Choisissez... **Choisissez...** **Choisissez...**

Adresse e-mail de sauvegarde.

Si vous oubliez vos réponses, nous pouvons utiliser cette autre adresse e-mail pour vous envoyer des instructions sur la réinitialisation de vos données personnelles.

Veuillez saisir votre date de naissance.

Jour: **Mois:** **Année:**

Aimeriez-vous recevoir ce qui suit par e-mail ?

Nouvelles offres dans l'iTunes Store.

Nouvelles offres spéciales et informations sur les produits et services d'Apple.

CRÉATION D'UN APPLE ID (2)

My Apple ID

Create an Apple ID.

Your Apple ID gives you access to a variety of Apple services, including the App Store, Apple Books, iTunes, iCloud, and more. You'll also receive important information about services unless you opt out at any time.

[Read the Apple Customer Privacy Policy](#)

Create an Apple ID.

Please enter Apple ID and password. Enter your Apple ID and password. Apple ID will be used to verify account activity. Please enter the email address used to verify information when you sign up. Please enter a password that's easy to remember but hard for others to guess.

Apple ID

Password

Verify Password

Create a security question.

Select a security question to protect your account. This question will help us verify your identity if you forget your password.

Security Question

Answer

Select your birth date.

Confirming your birth date provides extra security by verifying your identity.

Enter your name.

Please enter your first name.

First Name

Middle Name

Last Name

Enter your primary address.

Please enter your mailing address.

Primary Address

City/Mail Code

Address Line 1

Address Line 2

Neighborhood

State/Province

Zip/Postal Code

Preferred Language

Preferred Language

Contact Preferences

With an Apple ID, you can automatically add the email addresses you've previously used to register with other Apple services to receive messages about the official language of your choice or account. Read the [Apple ID Contact Preferences Policy](#).

I'd like to receive promotional emails from Apple.

Add my email to my account.

Screen Name

Show my First and Last Name (Name Only)

With an Apple ID, you can automatically add the email addresses you've previously used to register with other Apple services to receive messages about the official language of your choice or account. Read the [Apple ID Contact Preferences Policy](#).

Please enter the characters you see in the image below.

I have read and agree to the [Terms of Service](#) and [Apple Customer Privacy Policy](#).

Conditions générales et Engagement de confidentialité d'Apple

Connexion sécurisée

Vous devez accepter ces Conditions générales afin de pouvoir utiliser ce service.
Si le pays de vos données de facturation n'est pas la France, cliquez ici.

Informations importantes sur les téléchargements de l'iTunes Store

Ce téléchargement est en cours depuis iTunes Store à Modulus, à l'aide d'une Mac Mini. Un iPod ou iPhone utilisent une technologie anti-copie électronique FairPlay.

Saisissez les informations de votre Identifiant Apple

Connexion sécurisée

- Les téléchargements effectués par l'App Store sont toujours réalisés via un iPhone ou iPad effectués via l'App Store ou
- Chaque téléchargement auquel vous participez au moins une fois
- Les CDs audio gravés peuvent être sauvegardées sur un disque dur
- Les téléchargements peuvent être sauvegardés
- Tous les téléchargements effectués à l'aide de l'App Store dans les 24 dernières heures pourront être visionnés
- Certains contenus peuvent contenir des éléments autres que

E-mail

pierre-emmanuel.aguerre@wan...

Mot de passe

Confirmer le mot de passe

Sécurité

Ces données personnelles

Choisissez

Choisissez

Choisissez

Indiquez un mode de paiement.

Connexion sécurisée

Vous ne serez facturé qu'après avoir effectué des achats.

Si le pays de votre adresse de facturation n'est pas la France, cliquez ici.

Choisissez un mode de paiement

Numéro de carte

numéro

Expiration

1 / 2013

Cryptogramme

0000 0000 0000 0000

Pour utiliser un code ou un cadeau, saisissez le code correspondant ici.

Saisissez le code

Adresse de facturation

Civilité :

Prénom

Nom de famille

Rue

Appartement, Atelier, bâtiment

Code postal

Ville

France

Téléphone

Apple utilise des méthodes de chiffrement standardisées pour protéger la confidentialité de vos informations personnelles.

Retour

Retour

Annuler

Créer un identifiant Apple

Web

iTunes

CRÉATION D'UN APPLE ID (2)

My Apple ID

Create an Apple ID.

Your Apple ID gives you easy access to a variety of Apple services, including the iTunes Store, Apple Online Store, iChat, and more. We will not share your information with anyone else unless you authorize us to do so.

[Read the Apple Customer Privacy Policy ↗](#)

Create an Apple ID.

Please use Apple ID and password information unique to your Apple ID. This will help us verify your account address for you later. Please enter the email address you've selected below and click Create Account.

Apple ID:

Password:

Confirm Password:

Create a security question.

Select a security question to protect your account. This question will help verify your identity should you forget your password.

Security Question:

Answer:

Select your birth date.

Confirming your birth date provides us with information to help us verify your account and prevent fraud.

Enter your name.

Please enter your full name.

First Name:

Middle Name:

Last Name:

Please provide a shipping address.

Please enter your mailing address.

Country/Region:

Address Line 1:

Address Line 2:

City/City:

State/Province:

Zip/Postal Code:

Preferred Language.

Preferred Language:

Contact Preferences.

With your Apple ID, we'll automatically allow you to receive promotional emails from Apple. Please enter the email address of your choice to receive these emails. Learn more about Apple's Privacy Policy.

Email:

Additional Information.

Read the Apple Customer Privacy Policy ↗

Please type the characters you see in the box.

I have read and agree to the [Terms of Service Policy](#).

My Apple ID

Create an Apple ID.

Your Apple ID gives you easy access to a variety of Apple services, including the iTunes Store, Apple Online Store, iChat, and more. We will not share your information with anyone else unless you authorize us to do so.

[Read the Apple Customer Privacy Policy ↗](#)

Verify your email to complete your Apple ID setup.

Please complete the verification process to update your Apple ID. Check your inbox for a verification email with instructions on how to activate your account.

Web

iTunes

CRÉATION D'UN APPLE ID (3)

CRÉATION D'UN APPLE ID (3)

Chèr(e) Pierre-Emmanuel Aguera,

Vous avez saisi [REDACTED] comme adresse électronique de contact pour votre identifiant Apple ID. Pour terminer le processus, nous devons vérifier qu'il s'agit bien de votre adresse électronique. Cliquez simplement sur le lien ci-dessous et ouvrez une session à l'aide de votre Apple ID et de votre mot de passe.

[Vérifiez maintenant >](#)

Pourquoi ce courrier électronique vous a-t-il été envoyé ?

L'envoi de ce courrier électronique s'applique lorsqu'une personne ajoute ou modifie une adresse électronique de contact pour un compte Apple ID. Si cela ne vous concerne pas, ne vous inquiétez pas. Personne ne peut utiliser votre adresse électronique comme adresse de contact pour un Apple ID sans votre vérification.

Pour plus d'informations, consultez la rubrique [Questions et réponses](#).

Merci,
L'assistance à la clientèle Apple

TM et copyright © 2013 Apple Inc. Apple Sales International, Hollyhill Industrial Estate, Cork, Ireland.
Numéro d'immatriculation de la société : 15719. Numéro de TVA : IE6554690W.
[Tous droits réservés](#) / [Tenez-moi informé](#) / [Engagement de confidentialité](#) / [Mon Apple ID](#)

Si vous ne souhaitez plus recevoir d'e-mails de la part d'Apple ou si votre adresse e-mail a changé, veuillez [cliquer ici](#).

CRÉATION D'UN APPLE ID (3)

Chèr(e) Pierre-Emmanuel Aguera,

Vous avez saisi [REDACTED] votre identifiant Apple ID. Pour terminer la vérification, il s'agit bien de votre adresse électronique et ouvrez une session à l'aide de votre identifiant Apple.

[Vérifiez maintenant >](#)

Pourquoi ce courrier électronique ?
L'envoi de ce courrier électronique modifie une adresse électronique que vous concerne pas, ne vous inquiétez pas. Utilisez cette adresse électronique comme adresse de contact.

Pour plus d'informations, consultez le site [apple.com](#).

Merci,
L'assistance à la clientèle Apple

Mon identifiant Apple

Vérifiez votre adresse électronique.
Veuillez vérifier l'adresse électronique associée à votre identifiant Apple.

Connectez-vous pour vérifier votre adresse électronique.
Pour vérifier l'adresse électronique [REDACTED], veuillez vous connecter avec votre identifiant Apple et votre mot de passe.

Identifiant Apple

[Vous avez oublié votre identifiant Apple ?](#)

Mot de passe

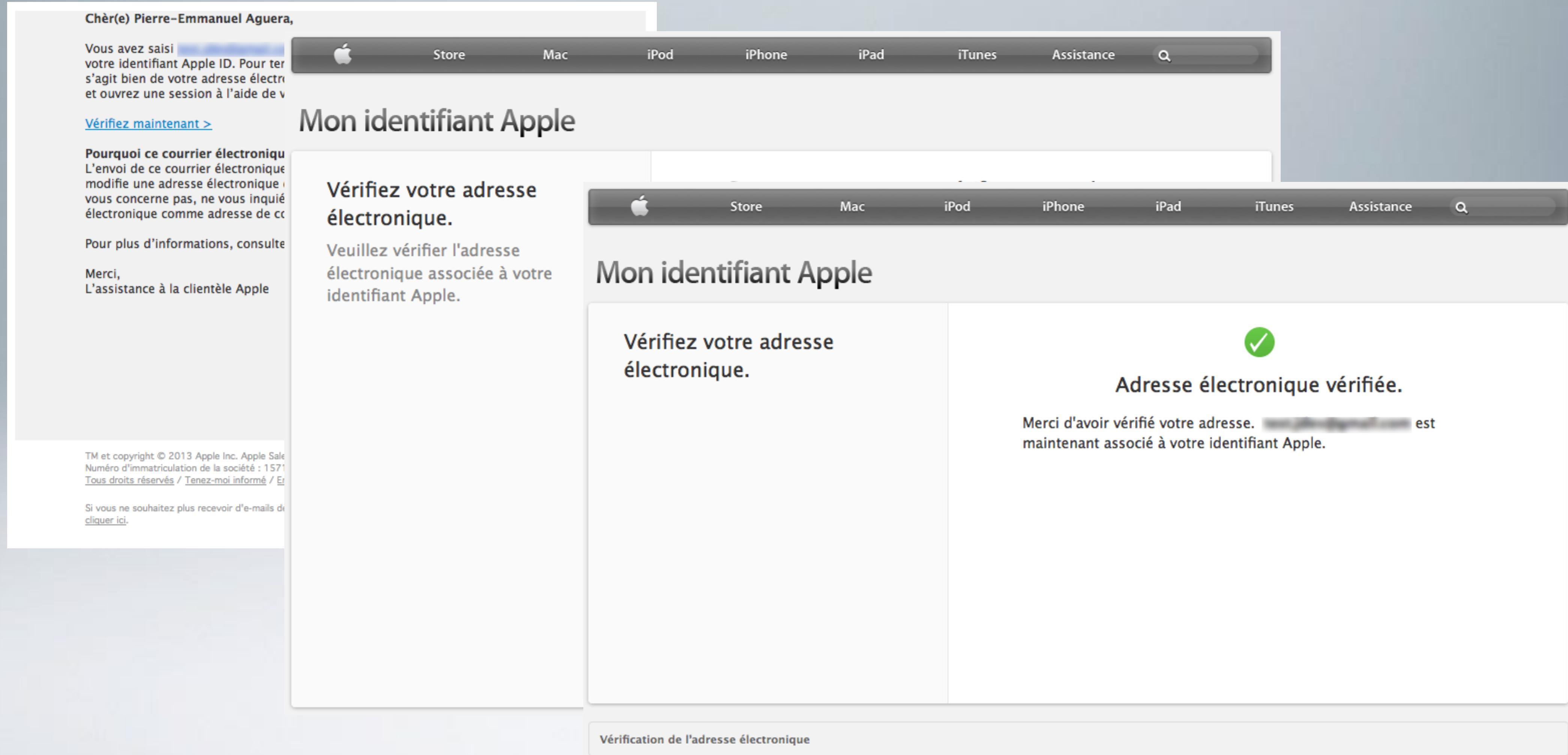
[Vous avez oublié votre mot de passe ?](#)

[Vérifier l'adresse](#)

TM et copyright © 2013 Apple Inc. Apple Sales and Support
Numéro d'immatriculation de la société : 1571
[Tous droits réservés / Tenez-moi informé / Envoyer un e-mail](#)

Si vous ne souhaitez plus recevoir d'e-mails de l'assistance à la clientèle Apple, [cliquer ici](#).

CRÉATION D'UN APPLE ID (3)



The image shows two screenshots of the Apple ID verification process. The top screenshot shows an email from Apple with a subject line "Chèr(e) Pierre-Emmanuel Aguera," and a message body stating "Vous avez saisi [REDACTED] votre identifiant Apple ID. Pour ter s'agit bien de votre adresse électronique et ouvrez une session à l'aide de v". Below the email is a link "[Vérifiez maintenant >](#)". The bottom screenshot shows a web page titled "Mon identifiant Apple" with the heading "Vérifiez votre adresse électronique." It contains the text "Veuillez vérifier l'adresse électronique associée à votre identifiant Apple." To the right of this text is a green checkmark icon and the text "Adresse électronique vérifiée." Below these sections is a message: "Merci d'avoir vérifié votre adresse. [REDACTED] est maintenant associé à votre identifiant Apple." At the bottom of the page is a link "[Vérification de l'adresse électronique](#)". Both screenshots feature a standard Apple navigation bar at the top.

Chèr(e) Pierre-Emmanuel Aguera,

Vous avez saisi [REDACTED] votre identifiant Apple ID. Pour ter s'agit bien de votre adresse électronique et ouvrez une session à l'aide de v

[Vérifiez maintenant >](#)

Pourquoi ce courrier électronique
L'envoi de ce courrier électronique modifie une adresse électronique qui vous concerne pas, ne vous inquiétez pas. Utilisez cette adresse électronique comme adresse de contact.

Pour plus d'informations, consultez le site Web de l'assistance à la clientèle Apple.

Merci,
L'assistance à la clientèle Apple

TM et copyright © 2013 Apple Inc. Apple Sales and Support. Numéro d'immatriculation de la société : 1571. Tous droits réservés / Tenez-moi informé / Envoyer un e-mail à [REDACTED]

Si vous ne souhaitez plus recevoir d'e-mails de l'assistance à la clientèle Apple, cliquer ici.

Vérifiez votre adresse électronique.

Mon identifiant Apple

Vérifiez votre adresse électronique.

✓

Adresse électronique vérifiée.

Merci d'avoir vérifié votre adresse. [REDACTED] est maintenant associé à votre identifiant Apple.

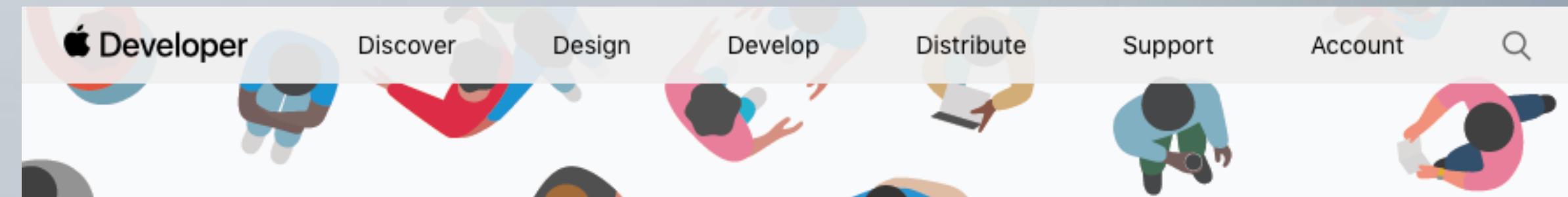
Vérification de l'adresse électronique

CREATION D'UN COMPTE DÉVELOPPEUR

(I)

CREATION D'UN COMPTE DÉVELOPPEUR (I)

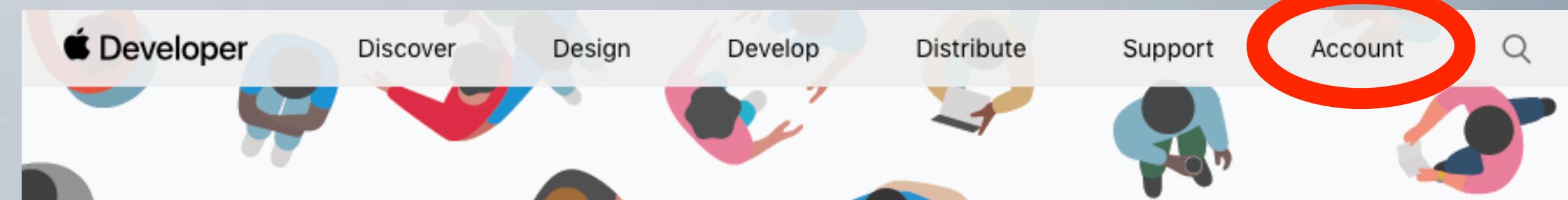
- <https://developer.apple.com>



CREATION D'UN COMPTE DÉVELOPPEUR

(I)

- <https://developer.apple.com>

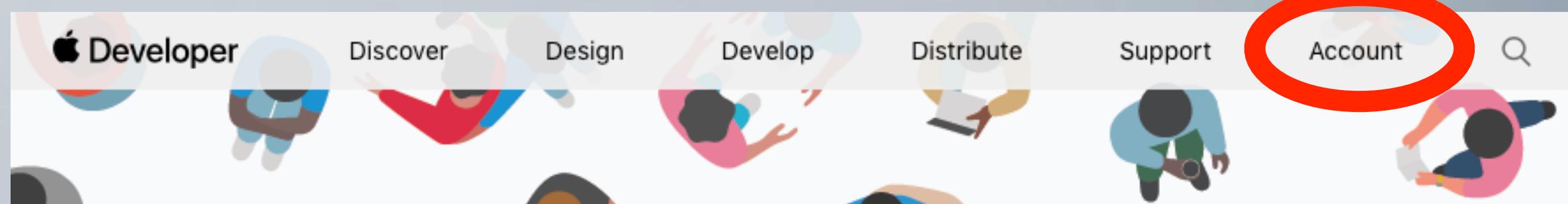


- Account

CREATION D'UN COMPTE DÉVELOPPEUR

(I)

- <https://developer.apple.com>



- Account

- Utilisation de l'AppleID



Apple Developer

Apple ID

Password

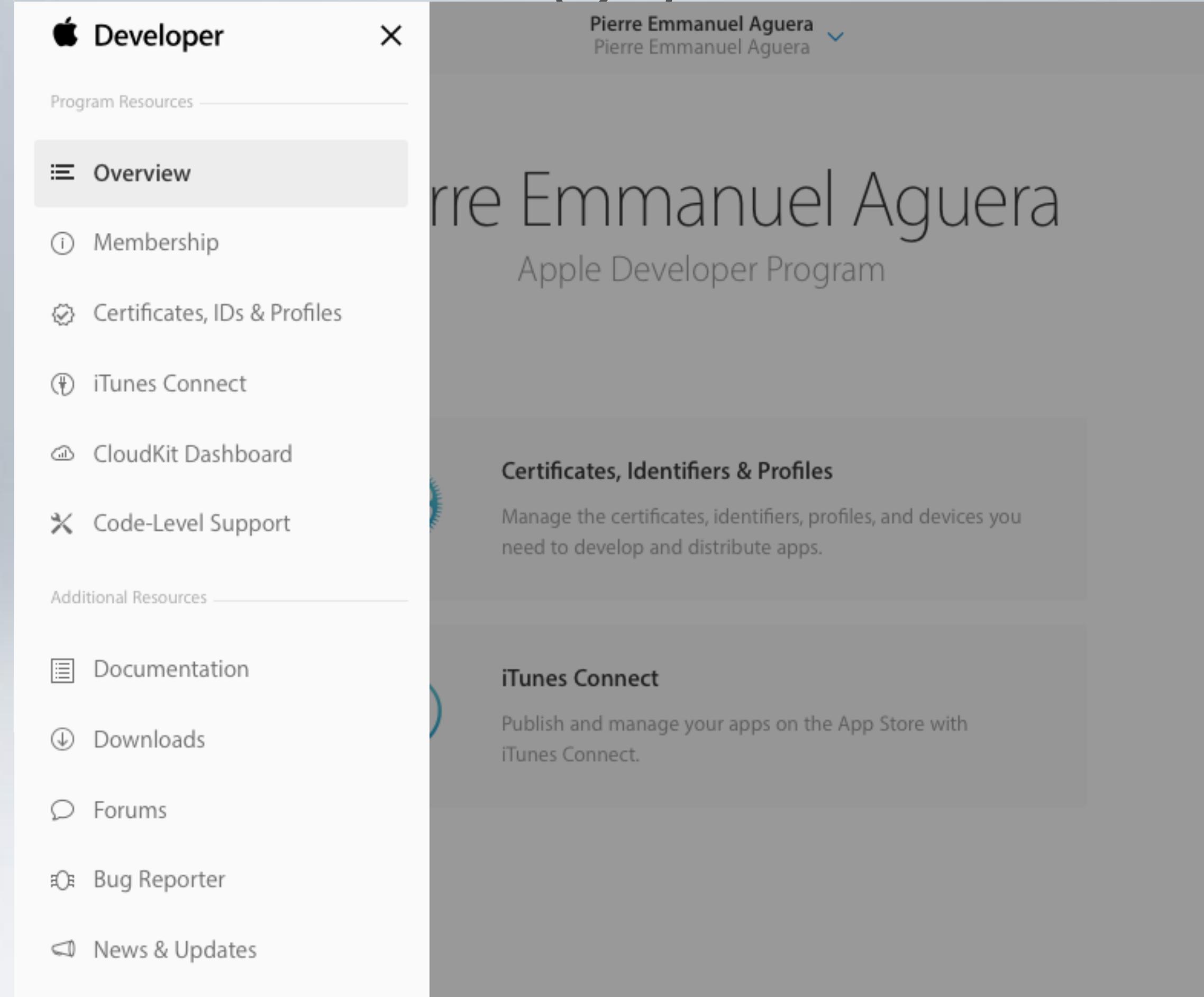
Create Apple ID

Sign In

Forgot ID or Password?

CREATION D'UN COMPTE DÉVELOPPEUR

(2)

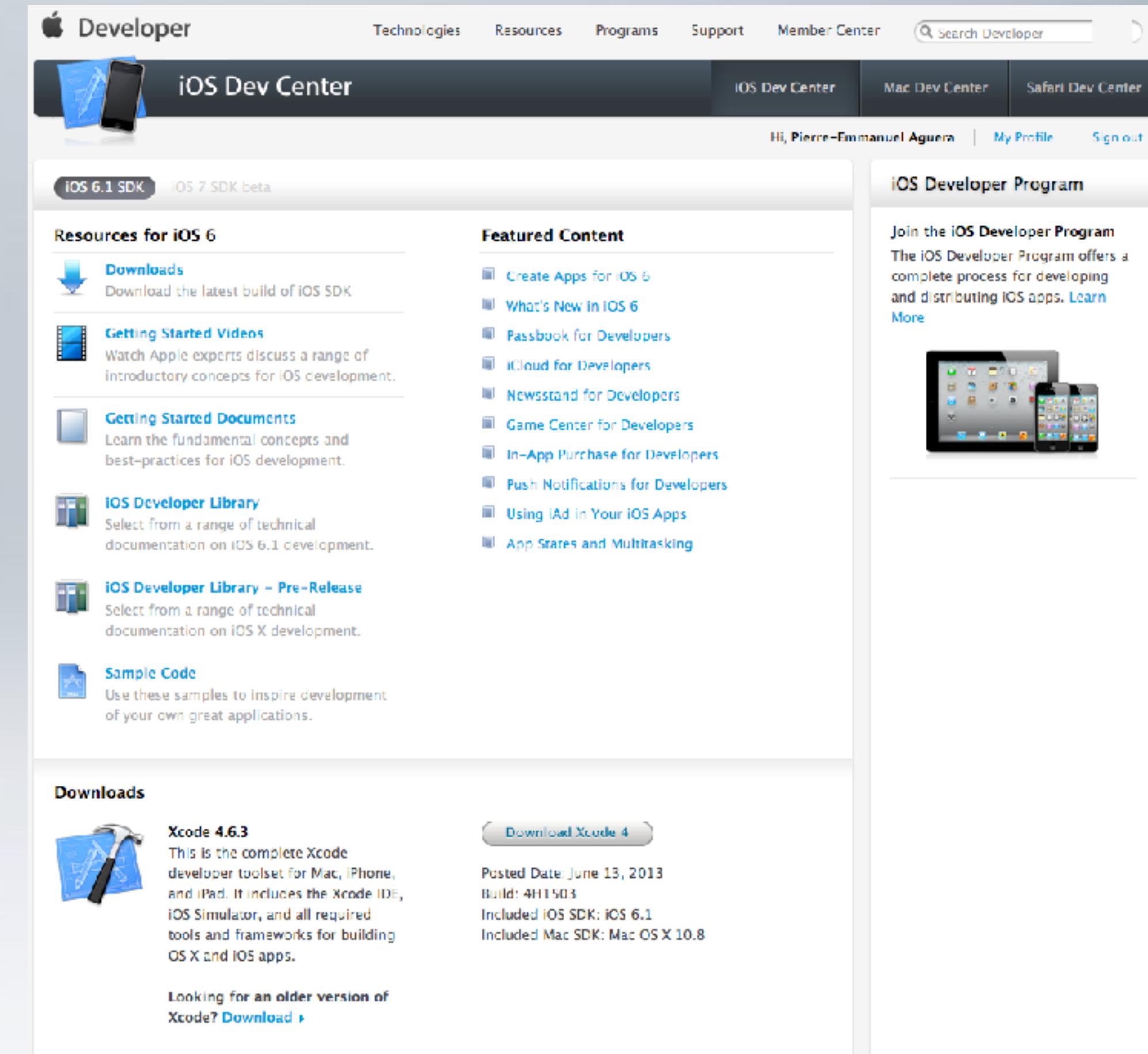


The screenshot shows the Apple Developer Program interface. On the left, a sidebar menu lists several options: Overview (selected), Membership, Certificates, IDs & Profiles (highlighted with a blue box), iTunes Connect, CloudKit Dashboard, and Code-Level Support. Below this is an 'Additional Resources' section with Documentation, Downloads, Forums, Bug Reporter, and News & Updates. On the right, the main content area is titled 'Pierre Emmanuel Aguera' and 'Apple Developer Program'. It features a large image of a person, the name 'Pierre Emmanuel Aguera', and a 'Certificates, Identifiers & Profiles' section with a sub-description: 'Manage the certificates, identifiers, profiles, and devices you need to develop and distribute apps.' Below this is an 'iTunes Connect' section with a sub-description: 'Publish and manage your apps on the App Store with iTunes Connect.'

INSTALLATION DE XCODE

INSTALLATION DE XCODE

Download XCode



The screenshot shows the Apple Developer website's iOS Dev Center. The top navigation bar includes links for Technologies, Resources, Programs, Support, Member Center, and a search bar. The main content area features sections for 'Resources for iOS 6' (Downloads, Getting Started Videos, Getting Started Documents, iOS Developer Library, iOS Developer Library – Pre-Release, Sample Code) and 'Featured Content' (Create Apps for iOS 6, What's New in iOS 6, Passbook for Developers, iCloud for Developers, Newsstand for Developers, Game Center for Developers, In-App Purchase for Developers, Push Notifications for Developers, Using iAd in Your iOS Apps, App Stares and Multitasking). A sidebar on the right is titled 'iOS Developer Program' with a sub-section for 'Join the iOS Developer Program'. The central part of the page displays the 'Downloads' section, which lists 'Xcode 4.6.3' as the latest version, posted on June 13, 2013. It includes details about the build number (4H1503), included SDKs (iOS 6.1, Mac OS X 10.8), and a download button.

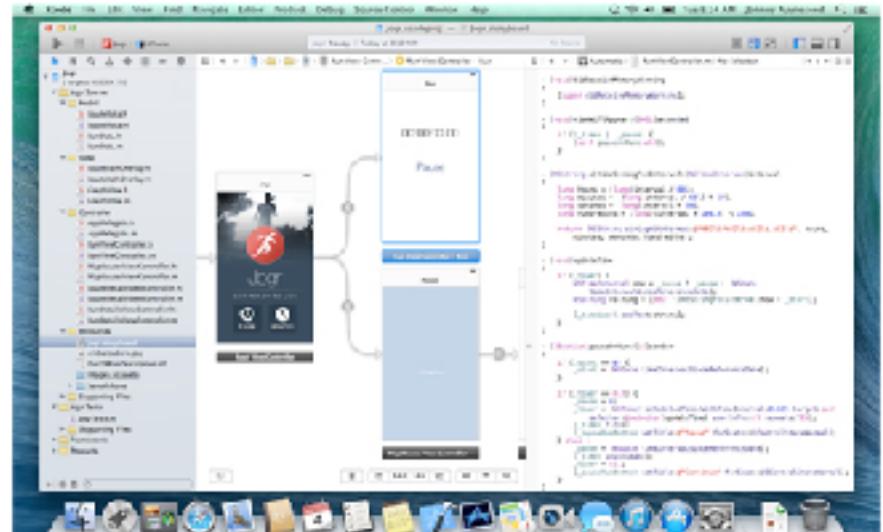
INSTALLATION DE XCODE

- Download XCode
- Renvoi vers Mac App Store

Xcode

Xcode is Apple's powerful integrated development environment for creating apps for Mac, iPhone, and iPad. Xcode includes the Instruments analysis tool, iOS Simulator, and the latest SDKs for iOS and OS X.

The Xcode interface seamlessly integrates code editing, UI design with Interface Builder, testing, and debugging, all within a single window. The embedded Apple LLVM compiler underlines coding mistakes as you type, and is even smart enough to fix the problems for you automatically. [Learn more](#) ▾



Download Xcode 4 for free.

Xcode 4.6.3
Xcode in the Mac App Store has been repackaged, and is now distributed as a stand-alone application. This replaces the Install Xcode package, and adds support for delta updates. Xcode includes a new "Downloads" preference pane to install optional components such as command line tools, and previous iOS Simulators.

Looking for additional developer tools? [View downloads](#) ▾

View in Mac App Store

What's New in Xcode

Xcode User Guide

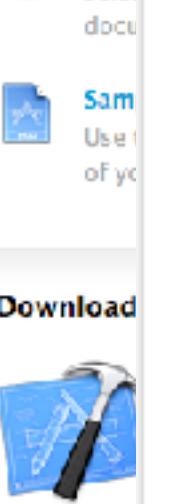
Xcode Release Notes

Download Xcode 5 Developer Preview.

Xcode 5 Developer Preview
This pre-release version of Xcode includes the OS X 10.9 SDK and iOS 7 SDK beta. iOS Developer Program or Mac Developer Program membership is required. Not a member? [Join now](#) ▾

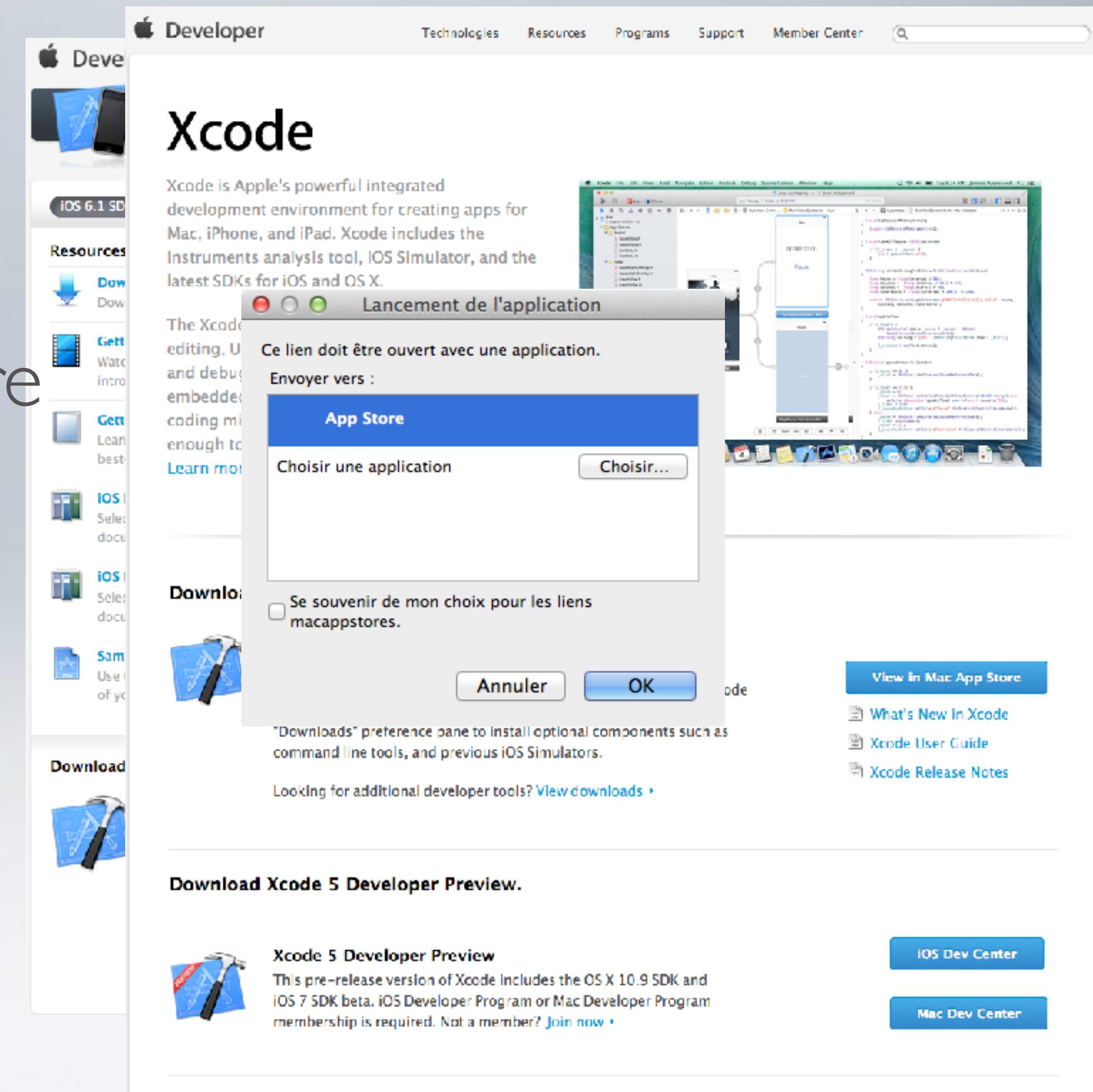
iOS Dev Center

Mac Dev Center



INSTALLATION DE XCODE

- Download XCode
- Renvoi vers Mac App Store
- Ouvrir avec l'App Store



INSTALLATION DE XCODE

Download XCode

• Renvoi vers Mac App Store

Ouvrir avec l'App Store

• Cliquer sur Gratuit



INSTALLATION DE XCODE

- Download XCode

- Renvoi vers Mac App Store

- Ouvrir avec l'App Store

- Cliquer sur Gratuit

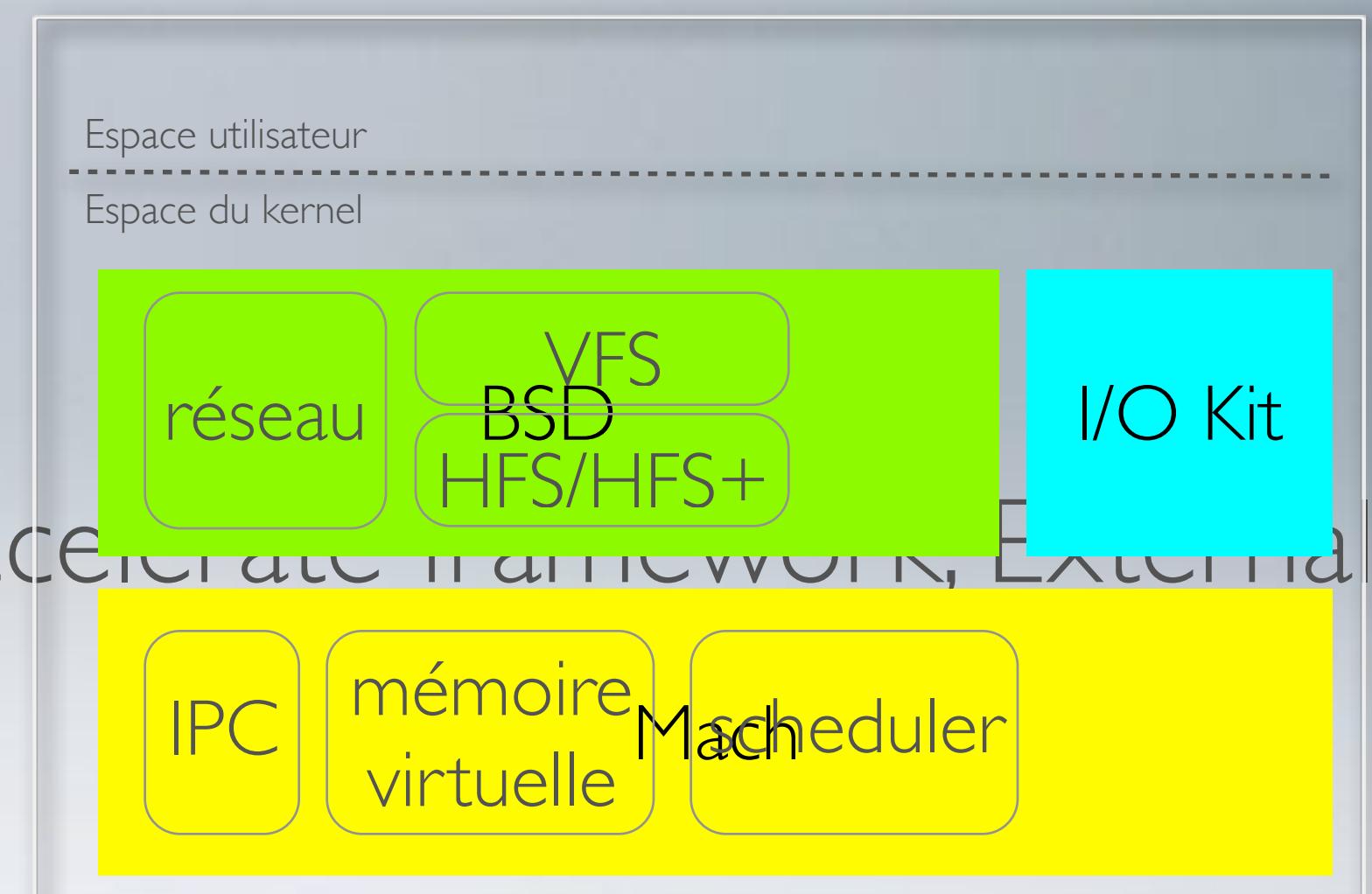
- Et installer l'app



CARACTÉRISTIQUES LOGICIELLES

Kernel

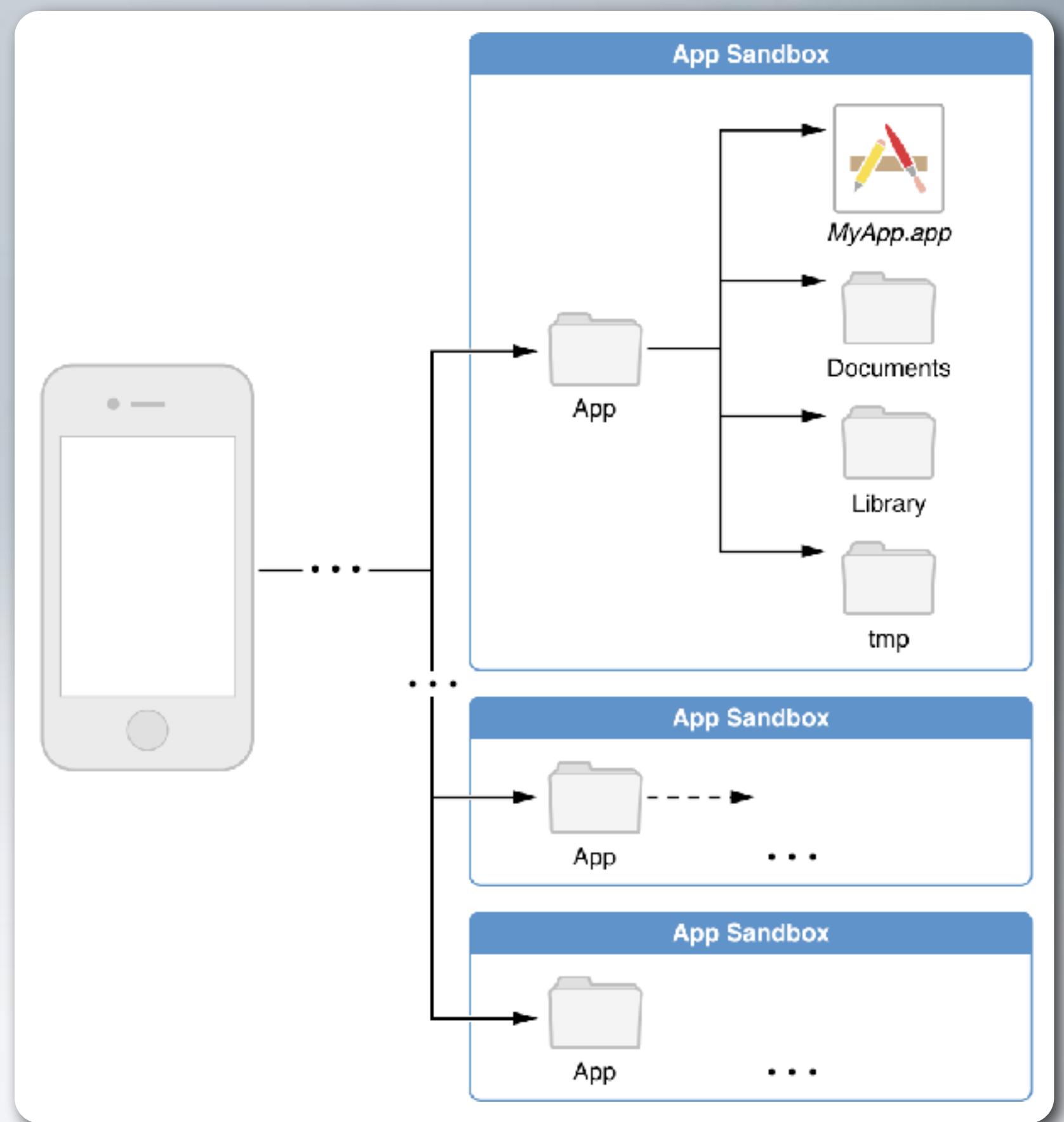
- iOS dérivé de Mac OS X
- Coeur Darwin
- Kernel XNU (X is Not UNIX)
- API spécifiques (SpringBoard, Cocoa Touch, Accelerate Framework, External Accessory Framework, ...)



APPLICATIONS

Limitations

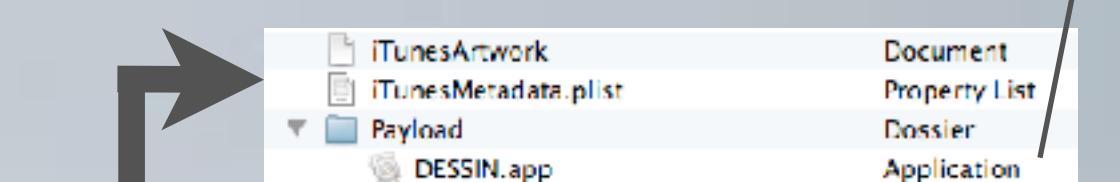
- Mémoire physique limitée
 - pas de swap
 - processus de libération de la mémoire
- Gestion de l'énergie :
 - systèmes de veille
 - multitâche
- Performances matérielles limitées



CARACTÉRISTIQUES LOGICIELLES

Bundles, applications et frameworks

- Bundle : structure hiérarchique standardisée contenant du code et les ressources utilisées par ce code
- Application : packagée dans un bundle contenant l'exécutable et ressources
- Framework : bundle contenant une ou plusieurs bibliothèques et fichiers



Mallette dessin 1.3.ipa iOS App

0345-10 Cubitus1.jpg	Image JPEG
0404-SCENE-NB.jpg	Image JPEG
Blank.jpg	Image JPEG
CEact002 NB.jpg	Image JPEG
CEact003 NB.jpg	Image JPEG
CDeric01 COLOR.jpg	Image JPEG
CDeric02 COLOR a.jpg	Image JPEG
cub4-1.jpg	Image JPEG
Village-coul.jpg	Image JPEG
stickers.txt	Format texte
DESSIN	Fichier Unix
_CodeSignature	Dossier
English.lproj	Dossier
es.lproj	Dossier
French.lproj	Dossier
German.lproj	Dossier
it.lproj	Dossier
ja.lproj	Dossier
pt.lproj	Dossier
ru.lproj	Dossier
SC_Info	Dossier
zh-Hans.lproj	Dossier
bddressin.bdtouch	Document
CreditViewController.nib	Document
dessinPack001.sqlite	Document
dessinPack002.sqlite	Document
dessinPack003.sqlite	Document
dessinPack004.sqlite	Document
dessinPack005.sqlite	Document
dessinPack006.sqlite	Document
dessinPack007.sqlite	Document
dessinPack008.sqlite	Document
dessinPack009.sqlite	Document
dessinPack010.sqlite	Document
MainWindow.nib	Document
PageBoxViewController.nib	Document
PkgInfo	Document
Presentation.nib	Document
ToolBoxViewController.nib	Document
Web_iPad.nib	Document
Web_iPod.nib	Document

CodeResources/	Symbolic link to Code Signature/CodeResources plist
Headers/	Symbolic link to Miscellaneous .h files provided by this
framework	
Resources/	.nib files (GUI), .lproj files, or other files required by
framework	
Versions/	Subdirectory to allow versioning
A/	Letter directories denoting version of this framework
Current/	Symbolic link to preferred framework version
Framework -name	Symbolic link to framework binary, in preferred version

LICENCES DE DÉVELOPPEMENT

Programmes

[https://developer.apple.com/
support/compare-memberships/](https://developer.apple.com/support/compare-memberships/)

Benefits and Resources

	Sign in with Apple ID	Individual	Organization	Enterprise Program
Xcode Developer Tools	•	•	•	•
Xcode Beta Releases	•	•	•	•
Developer Forums	•	•	•	•
Bug Reporter	•	•	•	•
Test on Device	•	•	•	•
Beta OS Releases		•	•	•
Advanced App Capabilities		•	•	•
App Store Distribution		•	•	
In-house App Distribution				•
Safari Extensions	•	•		
Developer ID	•	•	•	
Technical Support Incidents	•	•	•	
Team Management		•	•	
TestFlight Beta Testing	•	•		
App Analytics	•	•		
Cost	Free	99 USD*	99 USD*	299 USD**
Requirement	13+	18+	DUNS Number	DUNS Number

Introduction

RÔLE

<https://developer.apple.com/support/roles/>

Accept Legal Agreements	•		
Renew Membership	•		
Create Developer ID Certificates	•		
Invite Members and Assign Roles (organization only)	•	•	
Create Provisioning Profiles	•	•	○
Approve Certificate Signing Requests	•	•	
Create and Remove Development Certificates	•	•	•
Add and Disable UDIDs	•	•	○
Register and Configure App IDs	•	•	○
Delete App IDs	•	•	
Create an iOS Distribution Certificate and Distribution Provisioning Profiles	•	•	
Create Certificates for Apple Push Notification service and Pass Type IDs	•	•	
Create Mac App Distribution and Mac Installer Distribution Certificates	•	•	
Purchase and Submit Technical Support Incidents	•	•	•
Post in Developer Forums	•	•	•
Download Beta Software	•	•	•
Download Provisioning Profiles	•	•	•
Submit Certificate Signing Requests	•	•	•
Create Safari Extension Certificates	•	•	•
Submit Safari Extensions to the Safari Extensions Gallery	•	•	•

○ Access requires Xcode 7 or later.

DISTRIBUTION D'APPLICATION

Modes de distribution

- 4 façons de distribuer les applications

▶ AppStore

▶ B2B

▶ Ad Hoc

▶ In House

LANCEMENT



Welcome to Xcode

Version 8.3.3 (8E3004b)

-  **Get started with a playground**
Explore new ideas quickly and easily.
-  **Create a new Xcode project**
Create an app for iPhone, iPad, Mac, Apple Watch or Apple TV.
-  **Check out an existing project**
Start working on something from an SCM repository.

Show this window when Xcode launches

[Open another project...](#)

 Arduino_Servo ~/Downloads/Arduino_Servo_Starter-Swift3
 TestBeeWi ~/Documents/Devel/Apple/ios
 Questionnaire ~/Documents/Devel/Apple/ios
 MyPlayground ~/Documents/Devel/Apple/ios
 LocaStims ~/Documents/Devel/Apple/ios
 LocaStims ...uments/Devel/Apple/ios/LocaStims.Master
 LocaStims ...uments/Devel/Apple/ios/LocaStims.Master
 Screen&Touch Time .../ios/Screen&Touch Time/ScreenTouchTime
 Stabilo ~/Documents/Devel/Apple/ios

CRÉATION

Choose a template for your new project:

iOS	watchOS	tvOS	macOS	Cross-platform	Filter
Application					
1					
Single View Application	Game	Master-Detail Application	Page-Based Application	Tabbed Application	
Sticker Pack Application	iMessage Application				
Framework & Library					
Cocoa Touch Framework	Cocoa Touch Static Library	Metal Library			

Cancel Previous Next

Choose options for your new project:

Product Name:	JDEV-ObjC
Team:	Centre de recherche en Neurosciences d...
Organization Name:	Pierre Emmanuel Aguera
Organization Identifier:	cnl.fr
Bundle Identifier:	cnl.fr.JDEV-ObjC
Language:	Objective-C
Devices:	Universal
<input type="checkbox"/> Use Core Data <input type="checkbox"/> Include Unit Tests <input type="checkbox"/> Include UI Tests	

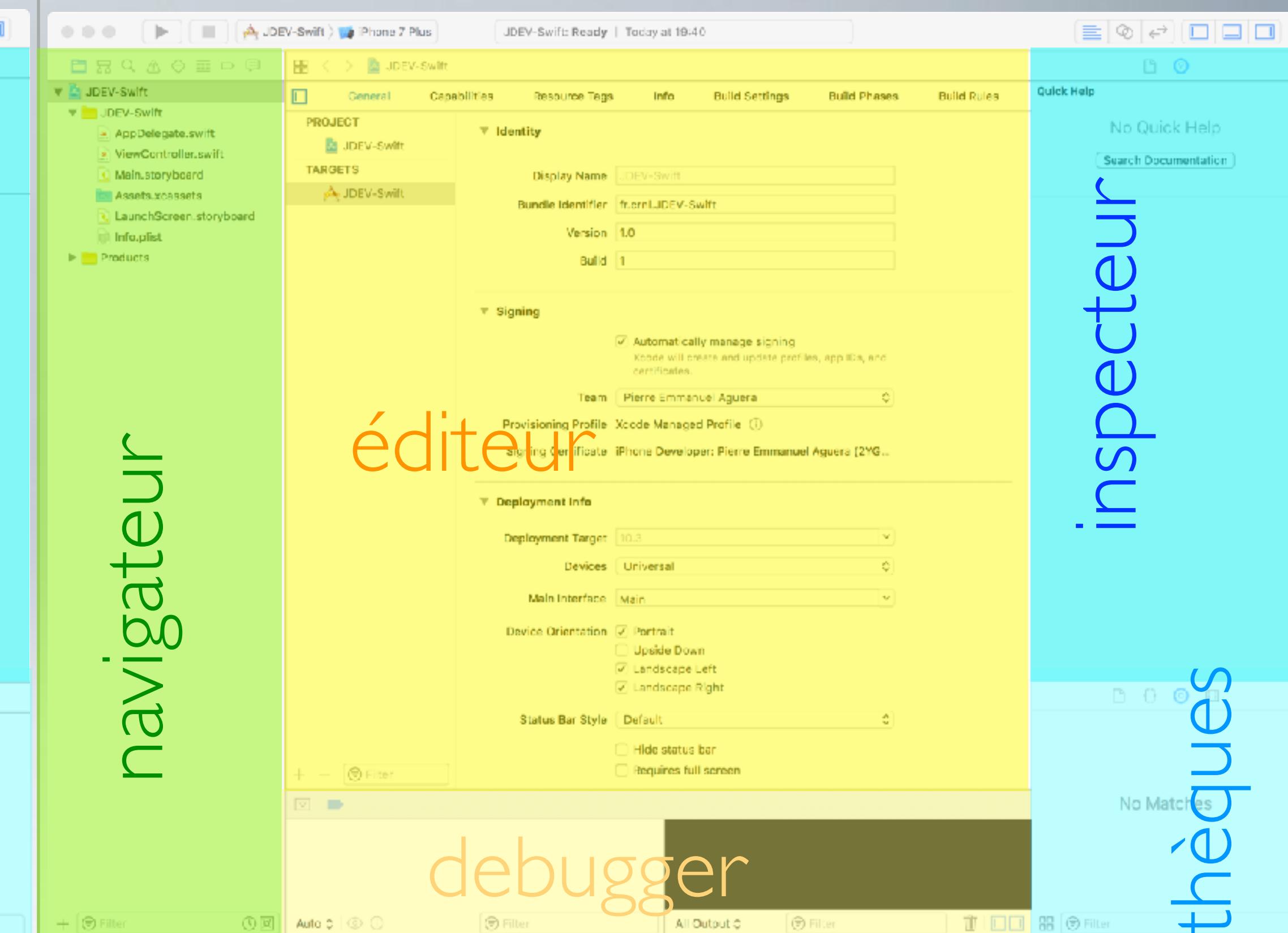
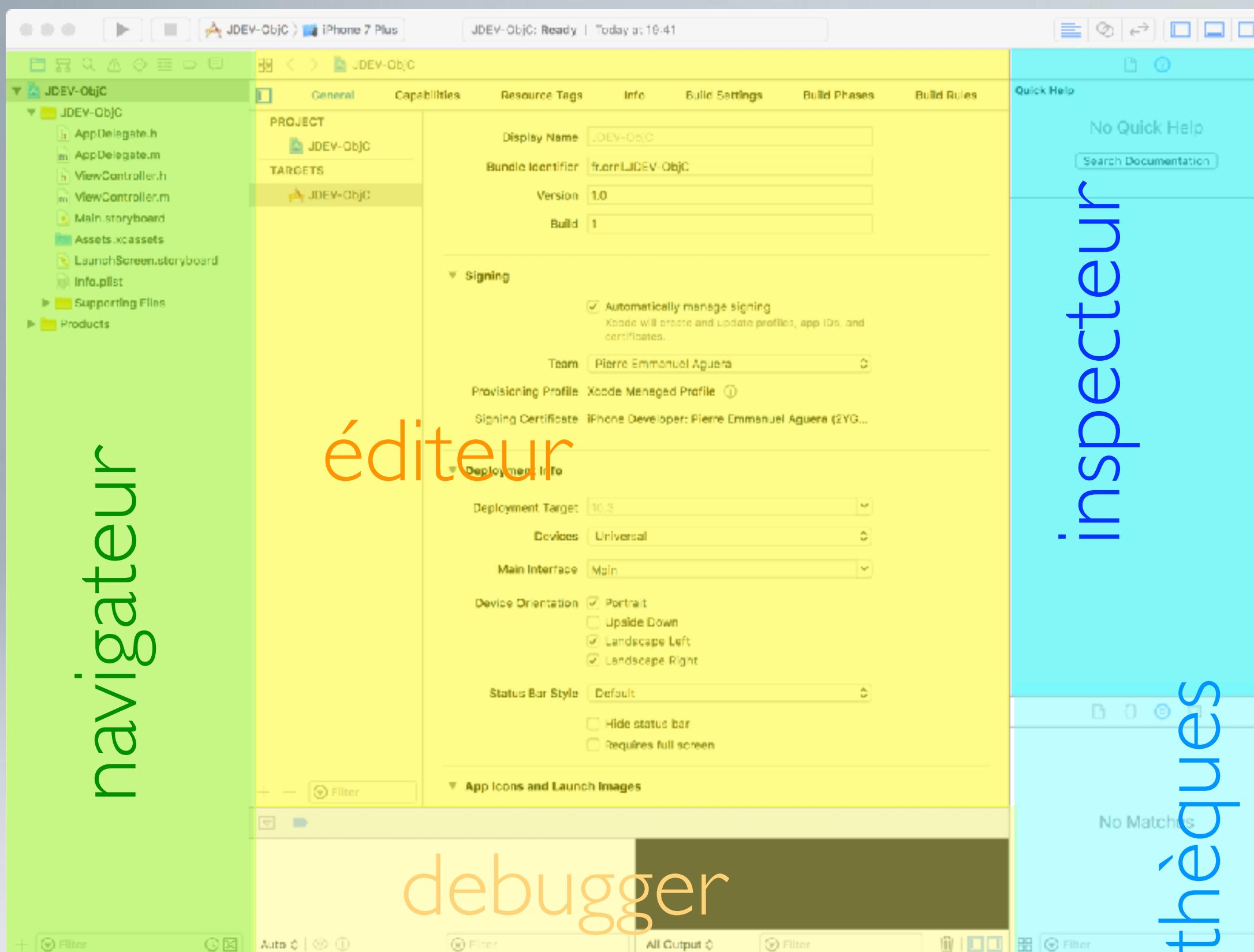
Cancel Previous Next

Choose options for your new project:

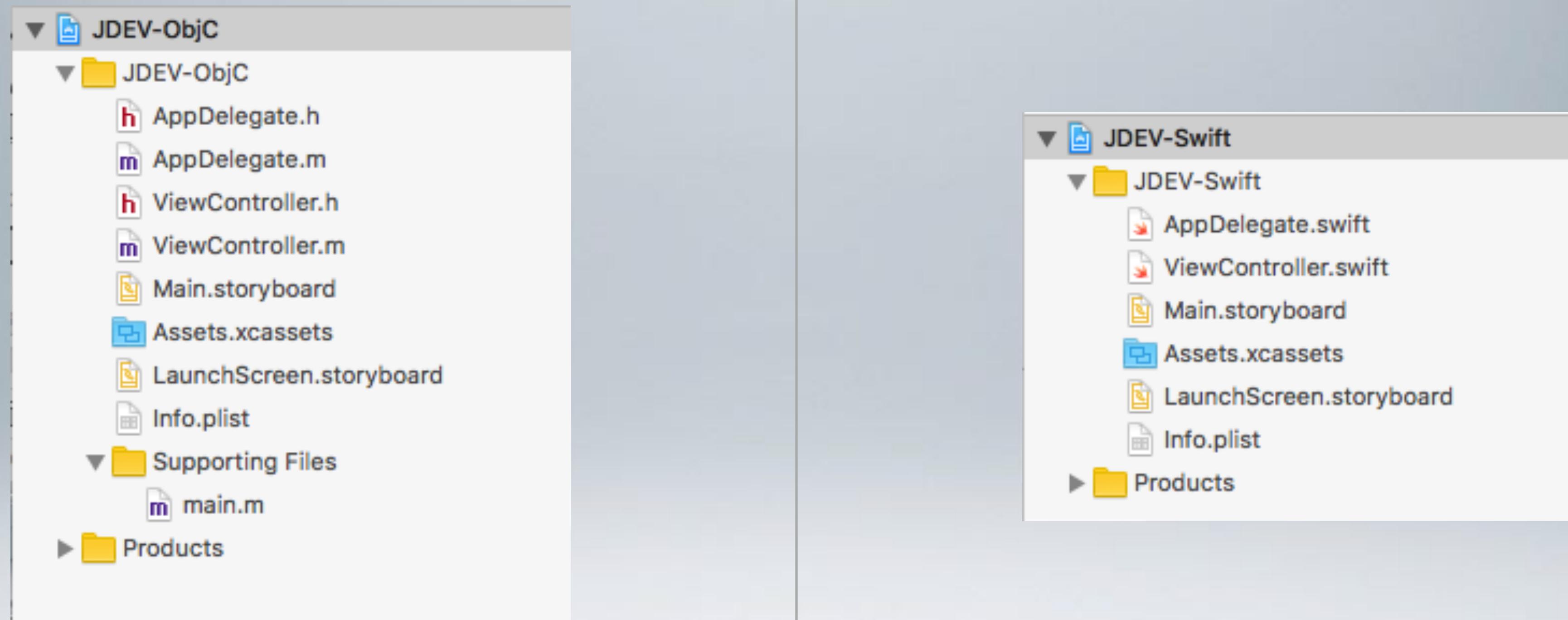
Product Name:	JDEV-Swift
Team:	Centre de recherche en Neurosciences d...
Organization Name:	Pierre Emmanuel Aguera
Organization Identifier:	cnl.fr
Bundle Identifier:	cnl.fr.JDEV-Swift
Language:	Swift
Devices:	Universal
<input type="checkbox"/> Use Core Data <input type="checkbox"/> Include Unit Tests <input type="checkbox"/> Include UI Tests	

Cancel Previous Next

CRÉATION PROJET



DESCRIPTION





1983

basé sur C et Smalltalk

orienté objet



v3.1

première version 2014
version 4.0 en développement

Objective-C, C#, CLU,
Haskell, Python, Racket,
Ruby, Rust et langage D

ELÉMENTS DE SYNTAXE

- Surcouche du C
- Objets Objective-C sont référencés par des pointeurs
- Certaines classes pour objets mutables ou non (NSMutableString, NSString, NSMutableArray, NSArray, ...)
 - Types de base: String, Int, Float, Double, Bool
 - 2 modes pour les variables: immuable (let) ou modifiable (var)
- Déclaration de méthode:
 - (int)maMethode:(int)n;
- appel de méthode:
 - [monObjet methode:param];
- Langage fonctionnel et objet
- Pas de séparateur d'instruction (comme ; en C)
- Déclaration de méthode:
 - func maMethode(n:Int)->Int
 - func maMethode()
- Appel de méthode:
 - monObjet.maMethode(param)



LES CLASSES

Création:
(alloc) init

Destruction:
dealloc

Propriétés:
getter et setter
automatiquement générés
(setter = setProperty, getter =
property)

Variable d'instance
accède par préfixe _

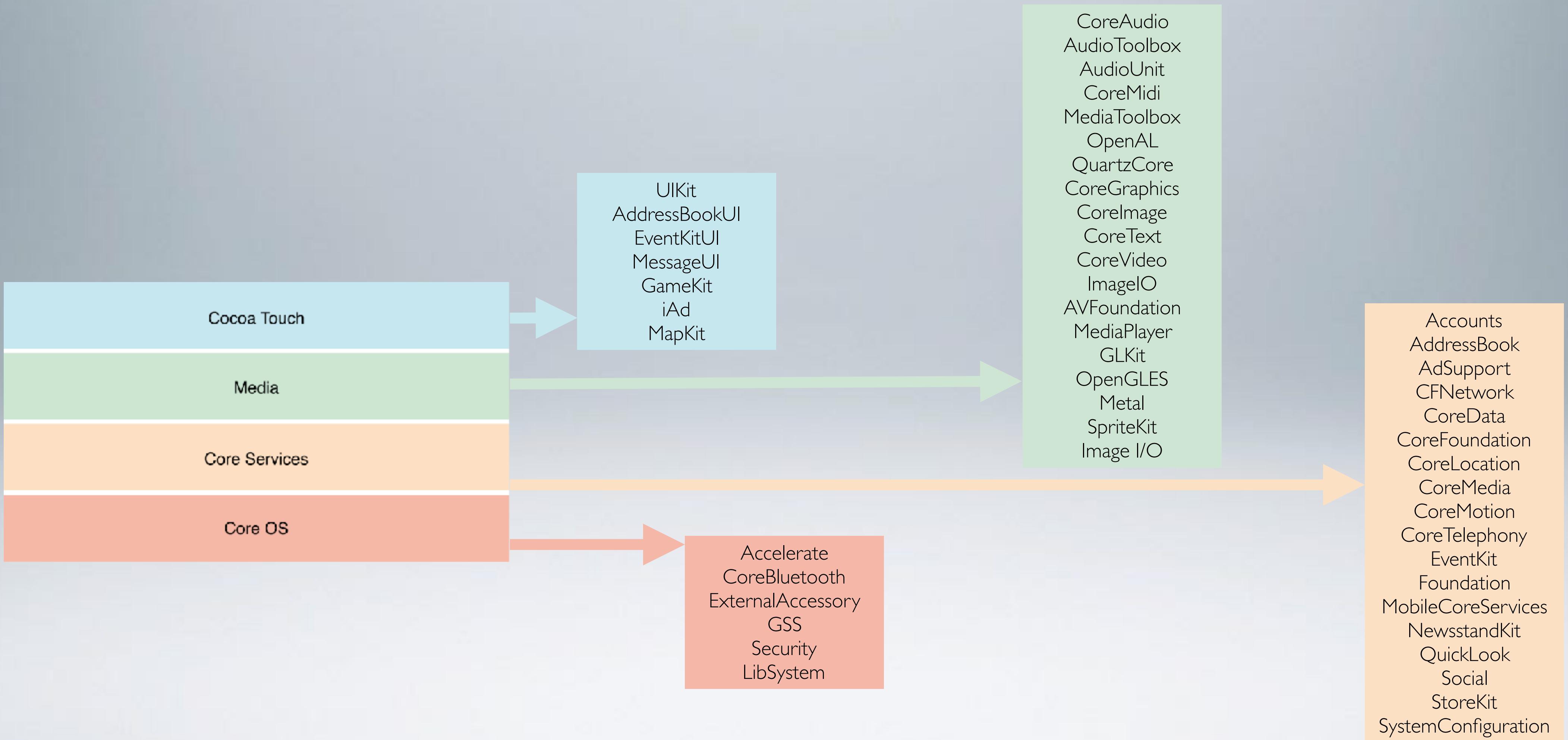
Création:
init

Destruction:
deinit

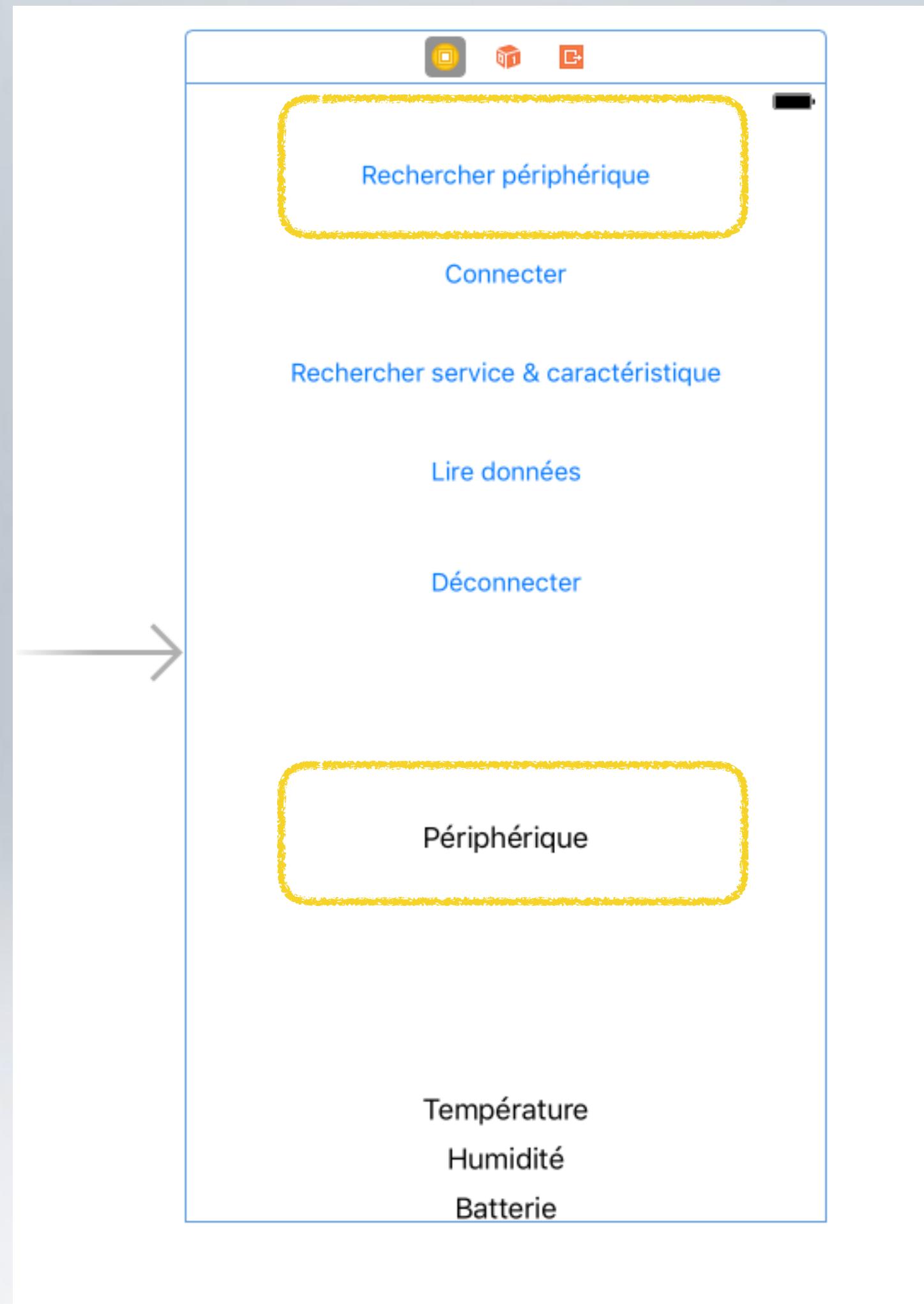
Propriétés:
accès direct



FRAMEWORKS



Interface



Code



```
//  
// ViewController.m  
// JDEV-ObjC  
  
// Created by Pierre Emmanuel Aguera on 23/06/2017.  
// Copyright © 2017 Pierre Emmanuel Aguera. All rights reserved.  
  
#import "ViewController.h"  
  
@interface ViewController ()  
  
@end  
  
@implementation ViewController  
  
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a  
nib.  
}  
  
- (void)didReceiveMemoryWarning {  
    [super didReceiveMemoryWarning];  
    // Dispose of any resources that can be recreated.  
}  
  
@end
```

```
//  
// ViewController.swift  
// JDEV-Swift  
  
// Created by Pierre Emmanuel Aguera on 23/06/2017.  
// Copyright © 2017 Pierre Emmanuel Aguera. All rights reserved.  
  
import UIKit  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a  
nib.  
    }  
  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
  
}
```



Code



```
// ViewController.m
// JDEV-ObjC
//
// Created by Pierre Emmanuel Aguera on 23/06/2017.
// Copyright © 2017 Pierre Emmanuel Aguera. All rights reserved.
//

#import "ViewController.h"

#define NO_PERIPH @"Pas de périphérique"

@interface ViewController : UIViewController
@property IBOutlet UIButton *searchBtn;
@property IBOutlet UILabel *periphLab;
@property NSString *periphName;

@end

@implementation ViewController

- (id)initWithCoder:(NSCoder *)aDecoder {
    self = [super initWithCoder:aDecoder];
    _periphName = NO_PERIPH;
    return self;
}

- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

```
// ViewController.swift
// JDEV-Swift
//
// Created by Pierre Emmanuel Aguera on 23/06/2017.
// Copyright © 2017 Pierre Emmanuel Aguera. All rights reserved.
//

import UIKit

class ViewController: UIViewController {
    @IBOutlet var searchBtn: UIButton?
    @IBOutlet var periphLab: UILabel?
    private var periphName: String

    private let NO_PERIPH = "Pas de périphérique"

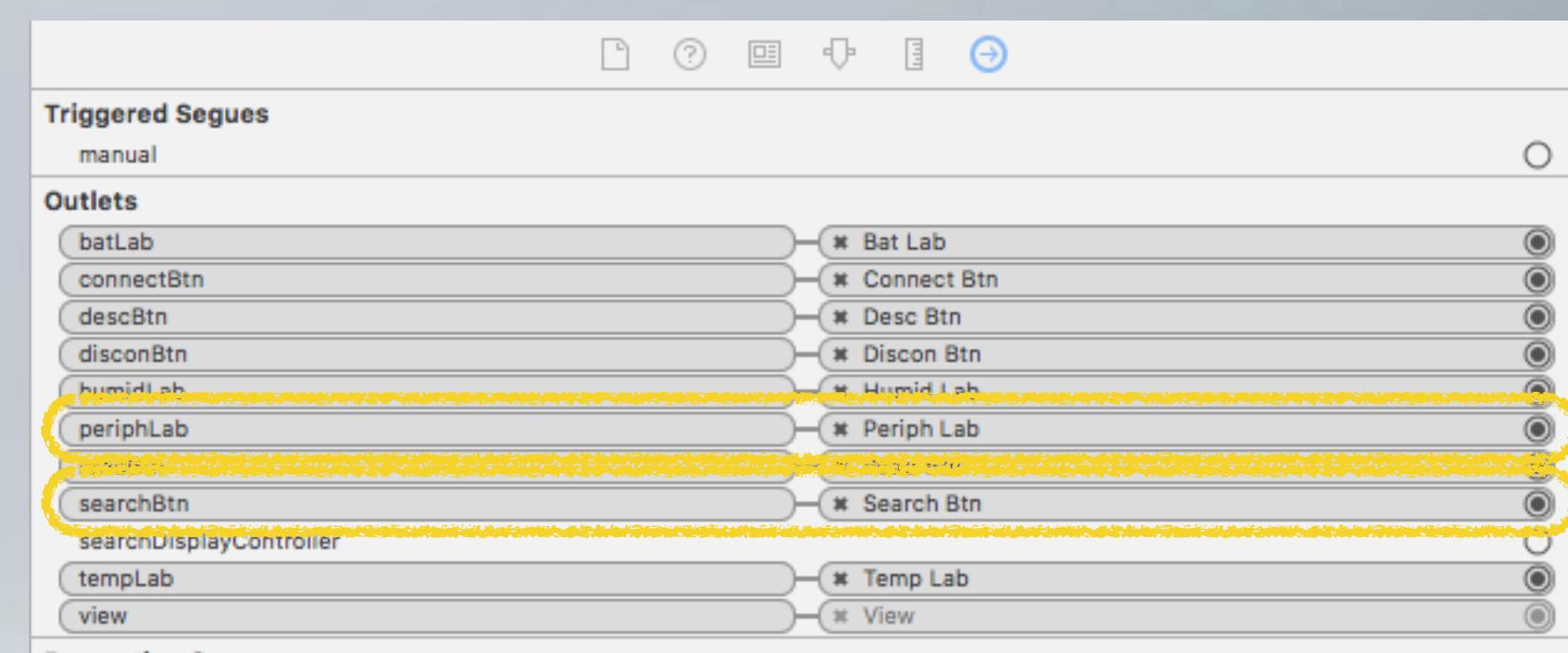
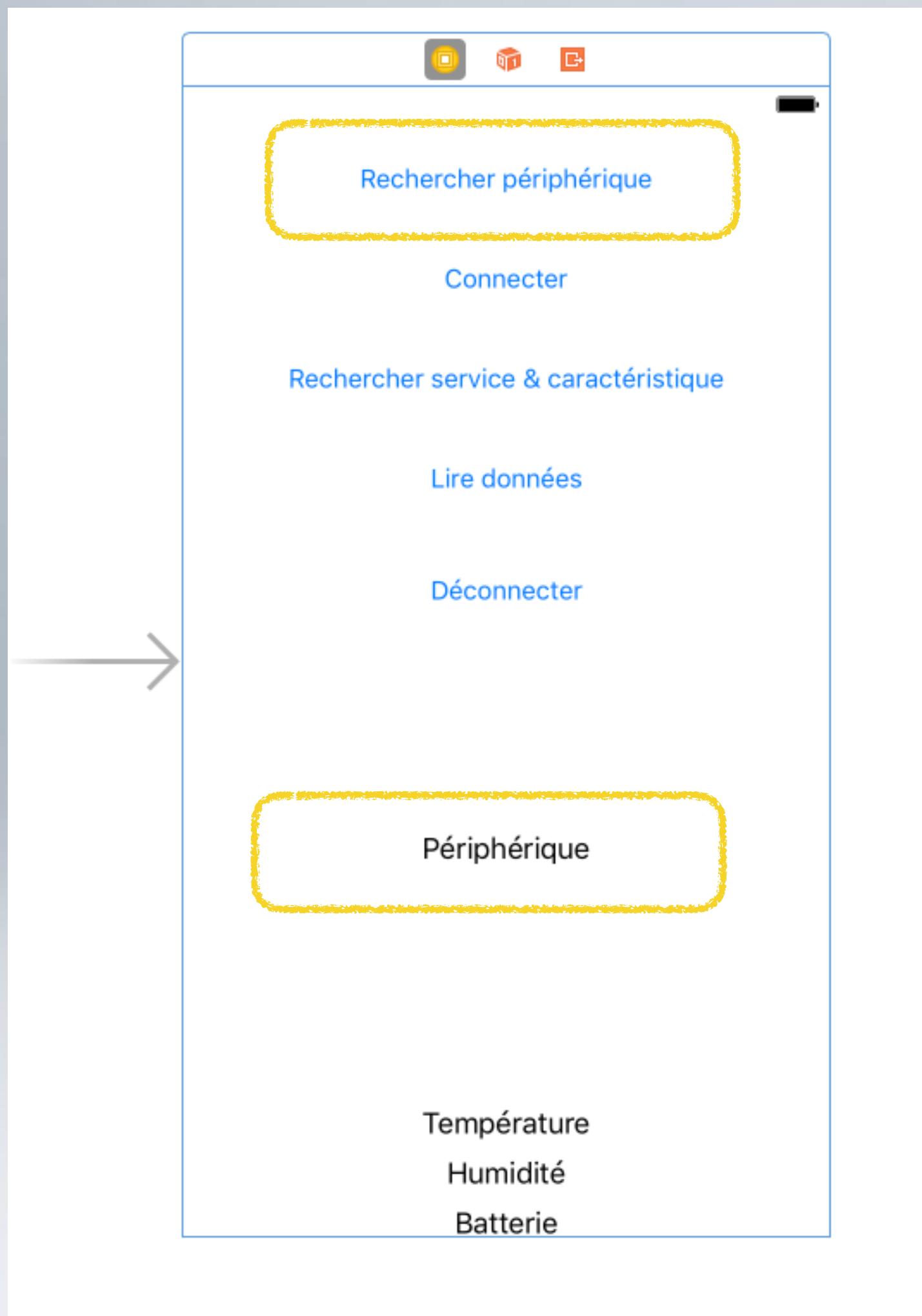
    required init?(coder aDecoder: NSCoder) {
        periphName = NO_PERIPH;
        super.init(coder: aDecoder)
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```



Interface



Code



```
typedef enum {UndefinedState} AppState_t;
```

```
@property AppState_t AppState;
```

```
_AppState = UndefinedState;
```

```
- (void)updateInterface {
    switch (_AppState) {
        case UndefinedState:
            [_searchBtn setEnabled:YES];
            _periphName = @"Pas de périphérique";
            [_periphLab setText:_periphName];
            [_periphLab setTextColor:[UIColor darkGrayColor]];
            break;
    }
}
```

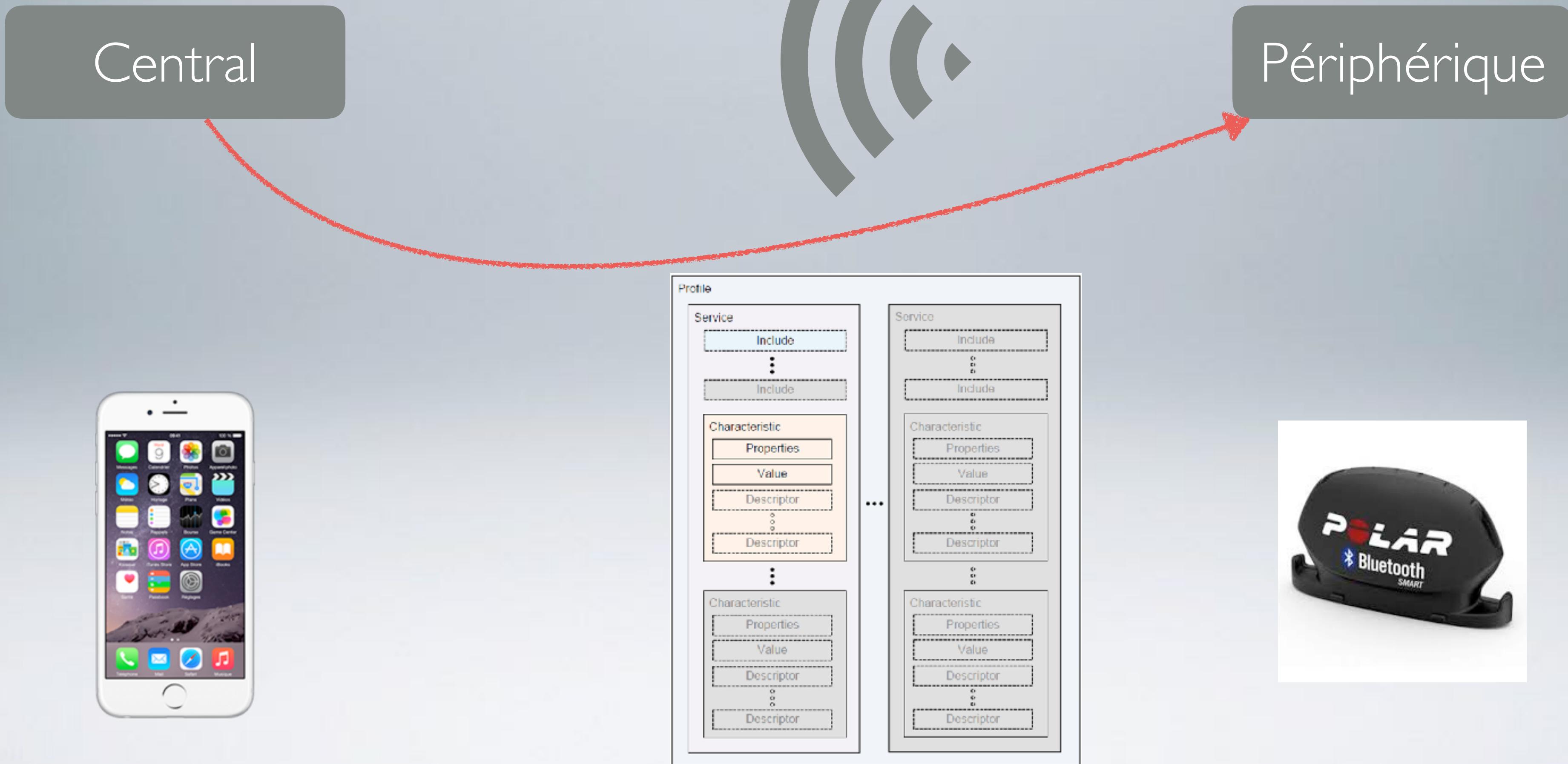
```
enum AppState {case undefinedState}
```

```
private var AppState: AppState
```

```
AppState = .undefinedState;
```

```
func updateInterface() {
    switch AppState {
        case .undefinedState,
            searchBtn?.isEnabled = true
            periphName = "Pas de périphérique"
            periphLab?.text = periphName
            periphLab?.textColor = UIColor.darkGray
    }
}
```

<https://www.bluetooth.com/specifications/gatt/services>



Code



Controller.h

```
@import CoreBluetooth;  
  
@interface ViewController : UIViewController <CBCentralManagerDelegate>
```

Controller.m

```
@property CBCentralManager *centralManager;  
  
_centralManager = [[CBCentralManager alloc] initWithDelegate:self  
queue:nil];
```

```
#pragma mark - CBCentralManagerDelegate  
- (void)centralManager:(CBCentralManager *)central  
didConnectPeripheral:(CBPeripheral *)peripheral {  
    NSLog(@"Central manager did connect to peripheral");  
}  
  
- (void)centralManager:(CBCentralManager *)central  
didFailToConnectPeripheral:(CBPeripheral *)peripheral error:(NSError  
*)error {  
    NSLog(@"Central manager failed to connect to peripheral");  
}  
  
- (void)centralManager:(CBCentralManager *)central  
didDisconnectPeripheral:(CBPeripheral *)peripheral error:(NSError  
*)error {  
    NSLog(@"Central manager did disconnect from peripheral");  
}  
  
- (void)centralManager:(CBCentralManager *)central  
didDiscoverPeripheral:(CBPeripheral *)peripheral advertisementData:  
(NSDictionary *)advertisementData RSSI:(NSNumber *)RSSI {  
}  
  
// method called whenever the device state changes.  
- (void)centralManagerDidUpdateState:(CBCentralManager *)central {
```

```
1 import CoreBluetooth  
  
4 class ViewController: UIViewController, CBCentralManagerDelegate {
```

```
2     private var centralManager: CBCentralManager?
```

```
3     centralManager = CBCentralManager.init(delegate:self, queue:nil)
```

```
// MARK: - CBCentralManagerDelegate  
func centralManager(_ central: CBCentralManager, didConnect  
peripheral: CBPeripheral) {  
    print("Central manager did connect to peripheral")  
}  
  
func centralManager(_ central: CBCentralManager,  
didDisconnectPeripheral peripheral: CBPeripheral, error: Error?) {  
    print("Central manager did disconnect from peripheral")  
}  
  
func centralManager(_ central: CBCentralManager, didDiscover  
peripheral: CBPeripheral, advertisementData: [String : Any], rssi RSSI:  
NSNumber) {  
}  
  
func centralManagerDidUpdateState(_ central: CBCentralManager) {  
}
```



CBPeripheralDelegate

Language

Swift

Objective-C

SDKs

iOS 5.0+

macOS 10.7+

tvOS 9.0+

On This Page

[Symbols](#)

[Relationships](#)

The delegate of a [CBPeripheral](#) object must adopt the CBPeripheralDelegate protocol. The delegate uses this protocol's methods to monitor the discovery, exploration, and interaction of a remote peripheral's services and properties. There are no required methods in this protocol.

Symbols

Discovering Services

`func peripheral(CBPeripheral, didDiscoverServices: Error?)`

Invoked when you discover the peripheral's available services.

`func peripheral(CBPeripheral, didDiscoverIncludedServicesFor: CBSERVICE, error: Error?)`

Invoked when you discover the included services of a specified service.

Discovering Characteristics and Characteristic Descriptors

`func peripheral(CBPeripheral, didDiscoverCharacteristicsFor: CBSERVICE, error: Error?)`

Invoked when you discover the characteristics of a specified service.

`func peripheral(CBPeripheral, didDiscoverDescriptorsFor: CBCharacteristic, error: Error?)`

Invoked when you discover the descriptors of a specified characteristic.

Retrieving Characteristic and Characteristic Descriptor Values

`func peripheral(CBPeripheral, didUpdateValueFor: CBCharacteristic, error: Error?)`

Invoked when you retrieve a specified characteristic's value, or when the peripheral device notifies your app that the characteristic's value has changed.

`func peripheral(CBPeripheral, didUpdateValueFor: CBDSCRIPTOR, error: Error?)`

Invoked when you retrieve a specified characteristic descriptor's value.

Writing Characteristic and Characteristic Descriptor Values

`func peripheral(CBPeripheral, didWriteValueFor: CBCharacteristic, error: Error?)`

Invoked when you write data to a characteristic's value.

`func peripheral(CBPeripheral, didWriteValueFor: CBDSCRIPTOR, error: Error?)`

Invoked when you write data to a characteristic descriptor's value.

Managing Notifications for a Characteristic's Value

`func peripheral(CBPeripheral, didUpdateNotificationStateFor: CBCharacteristic, error: Error?)`

Invoked when the peripheral receives a request to start or stop providing notifications for a specified characteristic's value.

Retrieving a Peripheral's Received Signal Strength Indicator (RSSI) Data

`func peripheralDidUpdateRSSI(CBPeripheral, error: Error?)`

Invoked when you retrieve the value of the peripheral's current RSSI while it is connected to the central manager.

`func peripheral(CBPeripheral, didReadRSSI: NSNumber, error: Error?)`

Invoked after you call `readRSSI()` to retrieve the value of the peripheral's current RSSI while it is connected to the central manager.

Monitoring Changes to a Peripheral's Name or Services

`func peripheralDidUpdateName(CBPeripheral)`

Invoked when a peripheral's name changes.

`func peripheral(CBPeripheral, didModifyServices: [CBSERVICE])`

Invoked when a peripheral's services have changed.

Code



```
typedef enum {UndefinedState, InitializedState} appState_t;
```

```
- (void)updateInterface {
    switch (_appState) {
        case UndefinedState:
        case InitializedState:
            [_searchBtn setEnabled:YES];
            _periphName = @"Pas de périphérique";
            [_periphLab setText:_periphName];
            [_periphLab setTextColor:[UIColor darkGrayColor]];
            break;
    }
}
```

```
switch ([central state]) {
    case CBManagerStatePoweredOn:
        NSLog(@"CoreBluetooth BLE hardware is powered on and ready");
        _appState = InitializedState;
        break;
    case CBManagerStatePoweredOff:
        NSLog(@"CoreBluetooth BLE hardware is powered off");
        _appState = UndefinedState;
        break;
    case CBManagerStateUnauthorized:
        NSLog(@"CoreBluetooth BLE state is unauthorized");
        _appState = UndefinedState;
        break;
    case CBManagerStateUnknown:
        NSLog(@"CoreBluetooth BLE state is unknown");
        _appState = UndefinedState;
        break;
    case CBManagerStateUnsupported:
        NSLog(@"CoreBluetooth BLE hardware is unsupported on this platform");
        _appState = UndefinedState;
        break;
    case CBManagerStateResetting:
        NSLog(@"CoreBluetooth BLE hardware is resetting");
        _appState = UndefinedState;
        break;
}
// Update interface
[self updateInterface];
```

2

```
enum AppState {case undefinedState, initializedState}
```

```
func updateInterface() {
    switch AppState {
        case .undefinedState,
             .initializedState:
            searchBtn?.isEnabled = true
            periphName = "Pas de périphérique"
            periphLab?.text = periphName
            periphLab?.textColor = UIColor.darkGray
    }
}
```

3

```
switch central.state {
    case .poweredOn:
        print("CoreBluetooth BLE hardware is powered on and ready")
        AppState = .initializedState
    case .poweredOff:
        print("CoreBluetooth BLE hardware is powered off")
        AppState = .undefinedState
    case .unauthorized:
        print("CoreBluetooth BLE state is unauthorized")
        AppState = .undefinedState
    case .unknown:
        print("CoreBluetooth BLE state is unknown")
        AppState = .undefinedState
    case .unsupported:
        print("CoreBluetooth BLE hardware is unsupported on this platform")
        AppState = .undefinedState
    case .resetting:
        print("CoreBluetooth BLE hardware is resetting")
        AppState = .undefinedState
}
// Update interface
updateInterface()
```

[ObjC]



```
typedef enum {UndefinedState, InitializedState, SearchingState}  
appState_t;
```

2

```
enum AppState {case undefinedState, initializedState, searchingState}
```

```
case SearchingState:  
[_searchBtn setEnabled:YES];  
[_connectBtn setEnabled:NO];  
[_descBtn setEnabled:NO];  
[_readBtn setEnabled:NO];  
[_disconBtn setEnabled:NO];  
  
[_periphLab setText:_periphName];  
[_periphLab setTextColor:[UIColor darkGrayColor]];  
break;
```

3

```
case .searchingState:  
    searchBtn?.isEnabled = true  
    periphLab?.text = periphName  
    periphLab?.textColor = UIColor.darkGray
```

```
#pragma mark - IBActions  
- (IBAction)searchPeripheral:(id)sender {  
    // Doesn't advertise for service  
    [_centralManager scanForPeripheralsWithServices:nil options:nil];  
  
    _AppState = SearchingState;  
    // Update interface  
    [self updateInterface];  
}
```

```
// MARK: - Actions  
@IBAction func searchPeripheral(sender: UIButton) {  
    // Doesn't advertise for service  
    centralManager?.scanForPeripherals(withServices: nil, options:nil)  
  
    AppState = .searchingState  
    // Update interface  
    updateInterface()  
}
```

Interface



The screenshot shows a mobile application interface for managing peripherals. At the top, there are standard iOS icons for back, home, and recent apps. Below them is a battery icon. A yellow circle highlights the first item in a list:

- Rechercher périphérique
- Connecter
- Rechercher service & caractéristique
- Lire données
- Déconnecter
- Périphérique
- Température
- Humidité
- Batterie

A large grey arrow points from the bottom left towards the 'Périphérique' item.

The diagram is a UML statechart fragment showing 'Received Actions' for a state labeled 'Périphérique'. The actions and their corresponding events are:

- connectToPeripheral: * Connect Btn Touch Up Inside
- connectToPeripheralWithSender: ○
- disconnectFromPeripheral: * Discon Btn Touch Up Inside
- disconnectFromPeripheralWithSender: ○
- readData: * Read Btn Touch Up Inside
- readDataWithSender: ○
- readDescFromPeripheral: * Desc Btn Touch Up Inside
- readDescFromPeripheralWithSender: ○
- searchPeripheral: * Search Btn Touch Up Inside
- searchPeripheralWithSender: ○

The action 'searchPeripheral:' is highlighted with a yellow circle.

```
@property CBPeripheral *thermPeripheral;
```

```
_thermPeripheral = nil;
```

```
NSString *localName = [advertisementData
objectForKey:CBAdvertisementDataLocalNameKey];
if ([localName length] > 0) {
    NSLog(@"Found the device: %@", localName);
    if ([localName compare:@"BeeWi SmartClim"] == NSOrderedSame) {
        _periphName = localName;

        // End scanning
        [_centralManager stopScan];

        // Retain peripheral
        _thermPeripheral = peripheral;

        _appState = PeriphFoundState;

        // Update interface
        [self updateInterface];
    }
}
```

2 private var thermPeripheral: CBPeripheral?

3 thermPeripheral = nil;

```
let localName: String? =
advertisementData[CBAdvertisementDataLocalNameKey] as! String?
if (localName != nil) && (strlen(localName)) > 0 {
    print("Found the device: \(String(describing: localName))")
    if (localName?.compare("BeeWi SmartClim") == .orderedSame) {
        periphName = localName!

        // End scanning
        centralManager?.stopScan()

        // Retain peripheral
        thermPeripheral = peripheral

        AppState = .periphFoundState

        // Update interface
        updateInterface()
    }
}
```

```
typedef enum {UndefinedState, InitializedState, SearchingState,  
PeriphFoundState} AppState_t;
```

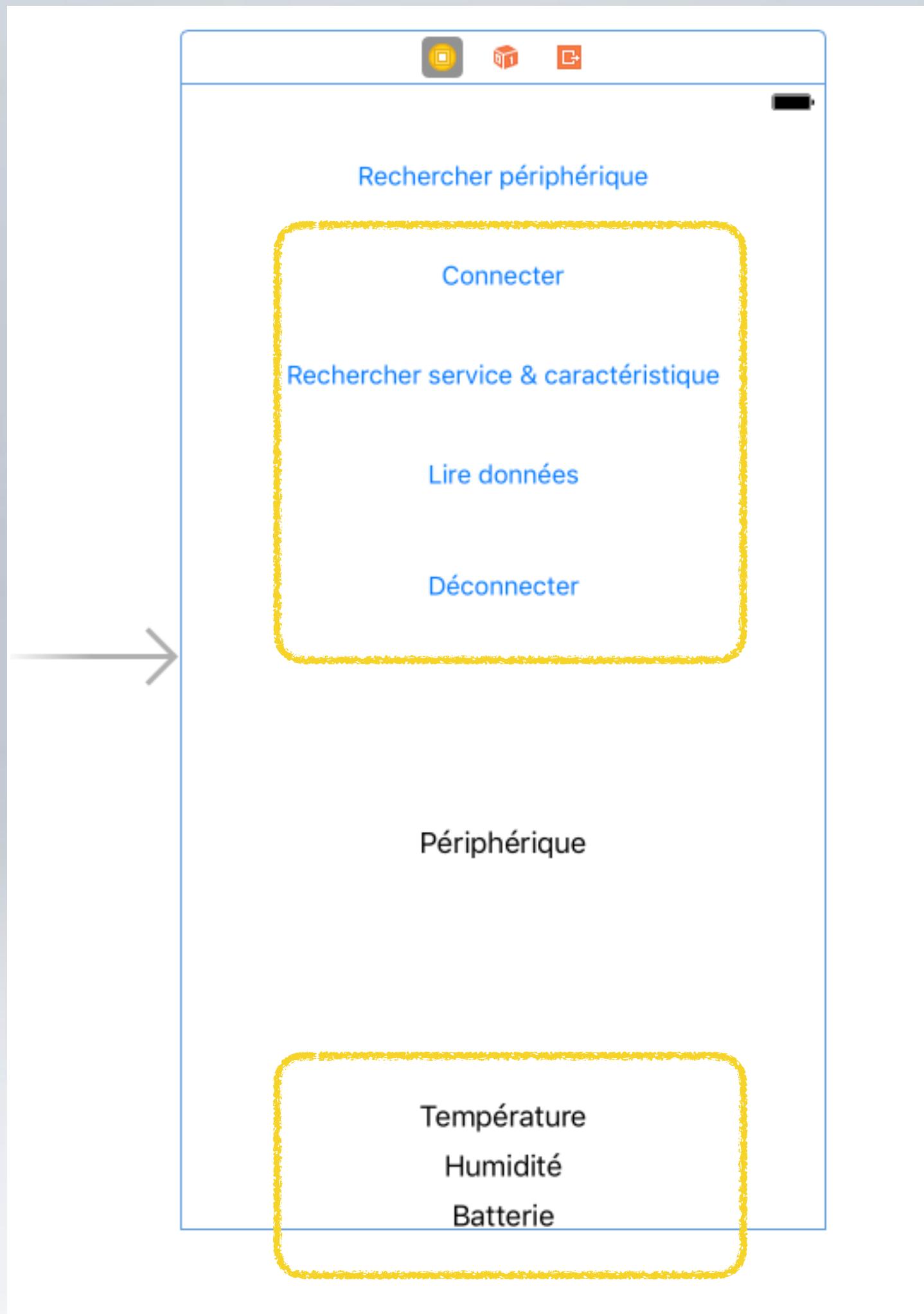
```
enum AppState {case undefinedState, initializedState, searchingState,  
periphFoundState}
```

```
case PeriphFoundState:  
    [_searchBtn setEnabled:YES];  
    [_periphLab setText:_periphName];  
    [_periphLab setTextColor:[UIColor redColor]];  
    break;
```

2

```
case .periphFoundState:  
    searchBtn?.isEnabled = true  
    periphLab?.text = periphName  
    periphLab?.textColor = UIColor.red
```

Interface



Code



```
@property IBOutlet UIButton *connectBtn;
@property IBOutlet UIButton *descBtn;
@property IBOutlet UIButton *readBtn;
@property IBOutlet UIButton *disconBtn;

@property IBOutlet UILabel *tempLab;
@property IBOutlet UILabel *humidLab;
@property IBOutlet UILabel *batLab;
```

```
case UndefinedState:
case InitializedState:
[_searchBtn setEnabled:YES];
[_connectBtn setEnabled:NO];
[_descBtn setEnabled:NO];
[_readBtn setEnabled:NO];
[_disconBtn setEnabled:NO];

_periphName = @"Pas de périphérique";
[_periphLab setText:_periphName];
[_periphLab setTextColor:[UIColor darkGrayColor]];
break;
case SearchingState:
[_searchBtn setEnabled:YES];
[_connectBtn setEnabled:NO];
[_descBtn setEnabled:NO];
[_readBtn setEnabled:NO];
[_disconBtn setEnabled:NO];

[_periphLab setText:_periphName];
[_periphLab setTextColor:[UIColor darkGrayColor]];
break;
case PeriphFoundState:
[_searchBtn setEnabled:YES];
[_connectBtn setEnabled:YES];
[_descBtn setEnabled:NO];
[_readBtn setEnabled:NO];
[_disconBtn setEnabled:NO];

[_periphLab setText:_periphName];
[_periphLab setTextColor:[UIColor redColor]];
break;
```

```
@IBOutlet var connectBtn: UIButton?
@IBOutlet var descBtn: UIButton?
@IBOutlet var readBtn: UIButton?
@IBOutlet var disconBtn: UIButton?

@IBOutlet var tempLab: UILabel?
@IBOutlet var humidLab: UILabel?
@IBOutlet var batLab: UILabel?
```

```
case .undefinedState,
     .initializedState:
    searchBtn?.isEnabled = true
    connectBtn?.isEnabled = false
    descBtn?.isEnabled = false
    readBtn?.isEnabled = false
    disconBtn?.isEnabled = false

    periphName = "Pas de périphérique"
    periphLab?.text = periphName
    periphLab?.textColor = UIColor.darkGray
case .searchingState:
    searchBtn?.isEnabled = true
    connectBtn?.isEnabled = false
    descBtn?.isEnabled = false
    readBtn?.isEnabled = false
    disconBtn?.isEnabled = false

    periphLab?.text = periphName
    periphLab?.textColor = UIColor.darkGray
case .periphFoundState:
    searchBtn?.isEnabled = true
    connectBtn?.isEnabled = true
    descBtn?.isEnabled = false
    readBtn?.isEnabled = false
    disconBtn?.isEnabled = false

    periphLab?.text = periphName
    periphLab?.textColor = UIColor.red
```

2

[ObjC]



```
@property float temperature;  
@property int humidity;  
@property int battery;
```

2

```
private var temperature: Float  
private var humidity: Int  
private var battery: Int
```

```
_temperature = 0.0;  
_humidity = 0;  
_battery = 0;
```

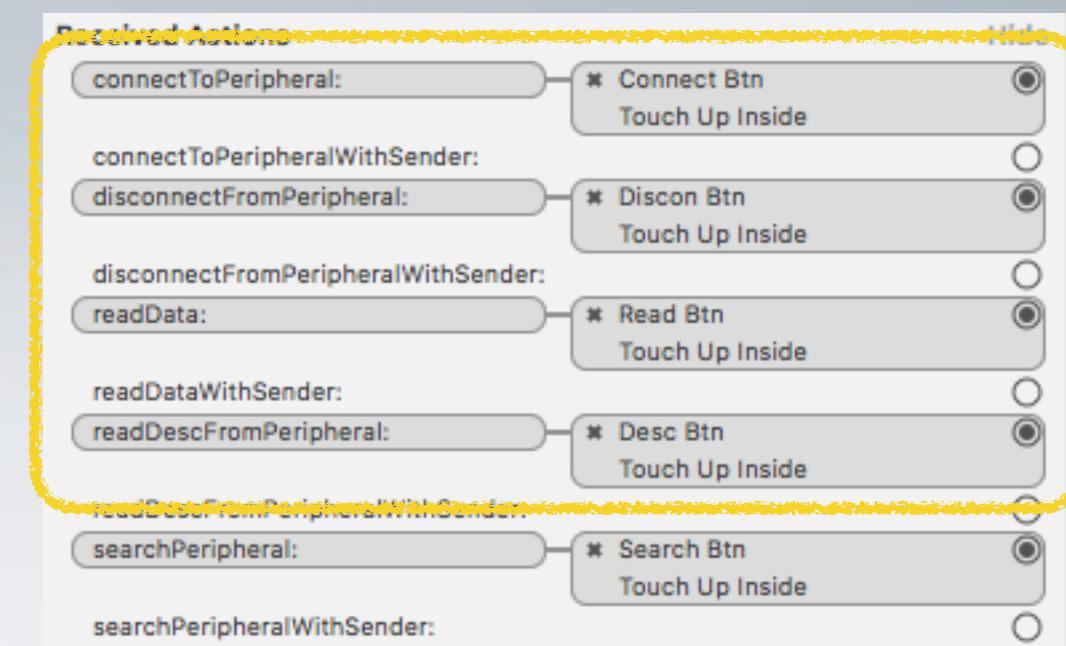
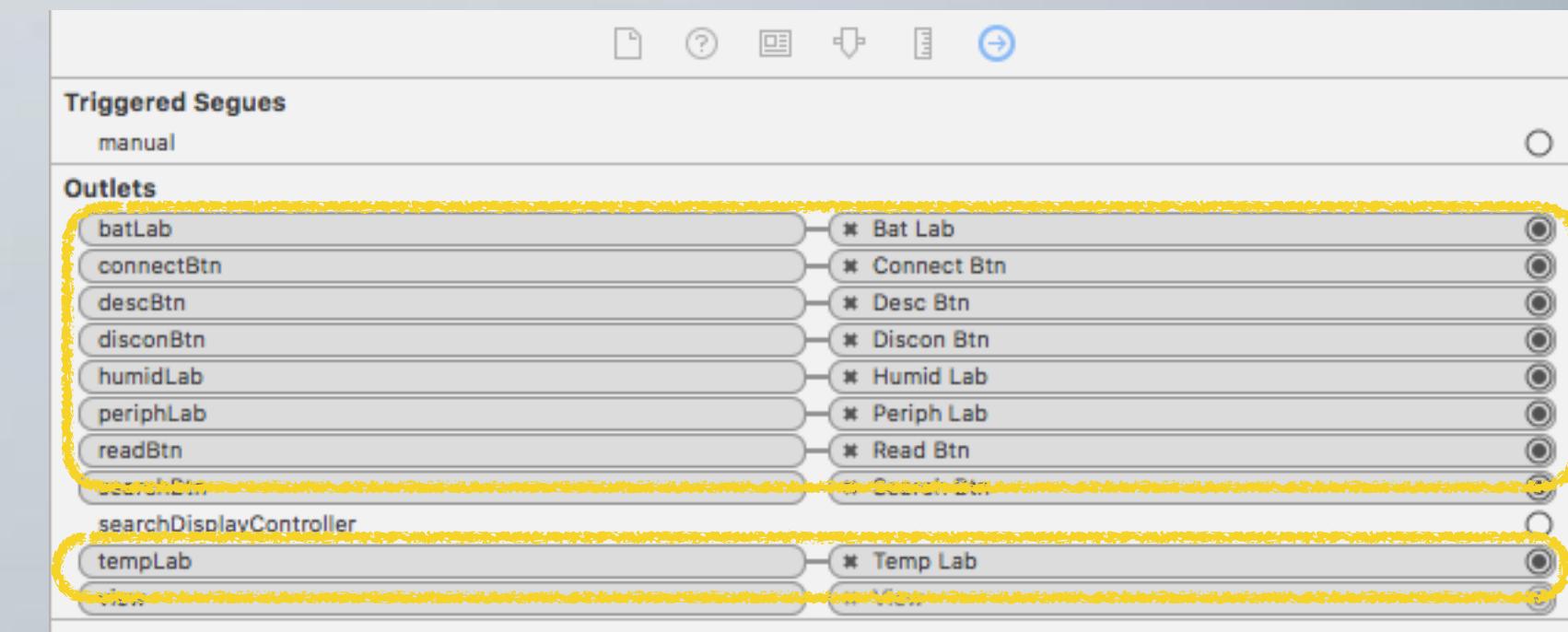
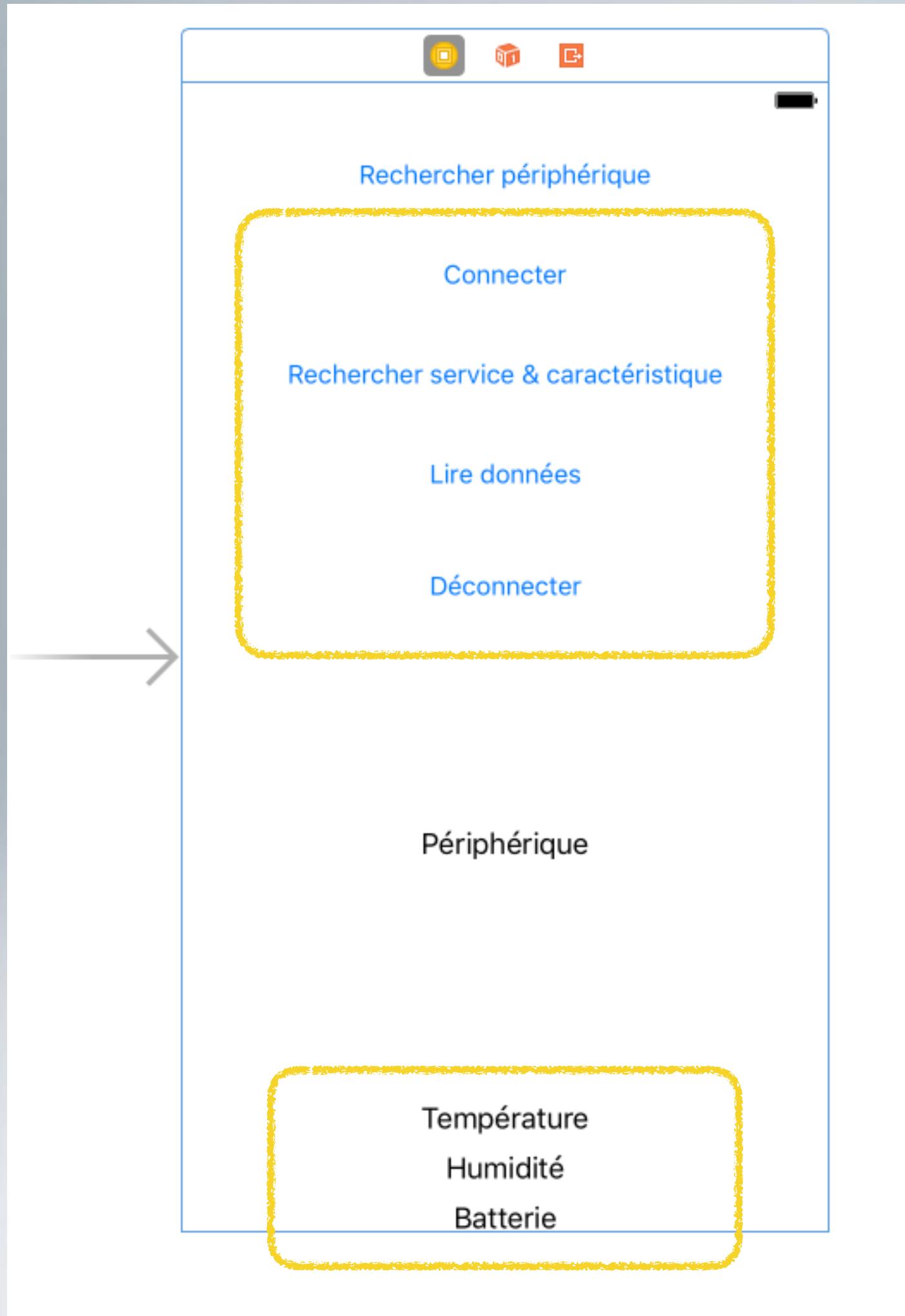
3

```
temperature = 0.0  
humidity = 0  
battery = 0
```

```
- (IBAction)connectToPeripheral:(id)sender {  
    // Connect  
}  
  
- (IBAction)readDescFromPeripheral:(id)sender {  
    // Read service and characteristic description  
}  
  
- (IBAction)readData:(id)sender {  
}  
  
- (IBAction)disconnectFromPeripheral:(id)sender {  
    // Disconnect  
}
```

```
@IBAction func connectToPeripheral(sender: UIButton) {  
    // Connect  
}  
  
@IBAction func readDescFromPeripheral(sender: UIButton) {  
    // Read service and characteristic description  
}  
  
@IBAction func readData(sender: UIButton) {  
}  
  
@IBAction func disconnectFromPeripheral(sender: UIButton) {  
    // Disconnect  
}
```

Interface



Code



Controller.h

```
@interface ViewController : UIViewController <CBCentralManagerDelegate,  
CBPeripheralDelegate>
```

Controller.m

```
#pragma mark - CBPeripheralDelegate  
  
// CBPeripheralDelegate - Invoked when you discover the peripheral's  
available services.  
- (void)peripheral:(CBPeripheral *)peripheral didDiscoverServices:  
(NSError *)error {  
  
// Invoked when you discover the characteristics of a specified  
service.  
- (void)peripheral:(CBPeripheral *)peripheral  
didDiscoverCharacteristicsForService:(CBService *)service error:  
(NSError *)error {  
  
// Invoked when you retrieve a specified characteristic's value, or  
when the peripheral device notifies your app that the characteristic's  
value has changed.  
- (void)peripheral:(CBPeripheral *)peripheral  
didUpdateValueForCharacteristic:(CBCharacteristic *)characteristic  
error:(NSError *)error {  
}
```

```
- (IBAction)connectToPeripheral:(id)sender {  
// Connect  
_thermPeripheral.delegate = self;  
  
[_centralManager connectPeripheral:_thermPeripheral options:nil];  
}
```

```
2 class ViewController: UIViewController, CBCentralManagerDelegate,  
CBPeripheralDelegate {
```

```
// MARK: - CBPeripheralDelegate  
  
// Invoked when you discover the peripheral's available services.  
func peripheral(_ peripheral: CBPeripheral, didDiscoverServices  
error: Error?) {  
}  
  
// Invoked when you discover the characteristics of a specified  
service.  
func peripheral(_ peripheral: CBPeripheral,  
didDiscoverCharacteristicsFor service: CBService, error: Error?) {  
}  
  
// Invoked when you retrieve a specified characteristic's value, or  
when the peripheral device notifies your app that the characteristic's  
value has changed.  
func peripheral(_ peripheral: CBPeripheral, didUpdateValueFor  
characteristic: CBCharacteristic, error: Error?) {  
}
```

```
@IBAction func connectToPeripheral(sender: UIButton) {  
// Connect  
thermPeripheral?.delegate = self  
  
centralManager?.connect(thermPeripheral!, options:nil)  
}
```

[ObjC]





CBPeripheralDelegate

Language

Swift

Objective-C

SDKs

iOS 5.0+

macOS 10.7+

tvOS 9.0+

On This Page

[Symbols](#)

[Relationships](#)

The delegate of a [CBPeripheral](#) object must adopt the CBPeripheralDelegate protocol. The delegate uses this protocol's methods to monitor the discovery, exploration, and interaction of a remote peripheral's services and properties. There are no required methods in this protocol.

Symbols

Discovering Services

`func peripheral(CBPeripheral, didDiscoverServices: Error?)`

Invoked when you discover the peripheral's available services.

`func peripheral(CBPeripheral, didDiscoverIncludedServicesFor: CBSERVICE, error: Error?)`

Invoked when you discover the included services of a specified service.

Discovering Characteristics and Characteristic Descriptors

`func peripheral(CBPeripheral, didDiscoverCharacteristicsFor: CBSERVICE, error: Error?)`

Invoked when you discover the characteristics of a specified service.

`func peripheral(CBPeripheral, didDiscoverDescriptorsFor: CBCharacteristic, error: Error?)`

Invoked when you discover the descriptors of a specified characteristic.

Retrieving Characteristic and Characteristic Descriptor Values

`func peripheral(CBPeripheral, didUpdateValueFor: CBCharacteristic, error: Error?)`

Invoked when you retrieve a specified characteristic's value, or when the peripheral device notifies your app that the characteristic's value has changed.

`func peripheral(CBPeripheral, didUpdateValueFor: CBDSCRIPTOR, error: Error?)`

Invoked when you retrieve a specified characteristic descriptor's value.

Writing Characteristic and Characteristic Descriptor Values

`func peripheral(CBPeripheral, didWriteValueFor: CBCharacteristic, error: Error?)`

Invoked when you write data to a characteristic's value.

`func peripheral(CBPeripheral, didWriteValueFor: CBDSCRIPTOR, error: Error?)`

Invoked when you write data to a characteristic descriptor's value.

Managing Notifications for a Characteristic's Value

`func peripheral(CBPeripheral, didUpdateNotificationStateFor: CBCharacteristic, error: Error?)`

Invoked when the peripheral receives a request to start or stop providing notifications for a specified characteristic's value.

Retrieving a Peripheral's Received Signal Strength Indicator (RSSI) Data

`func peripheralDidUpdateRSSI(CBPeripheral, error: Error?)`

Invoked when you retrieve the value of the peripheral's current RSSI while it is connected to the central manager.

`func peripheral(CBPeripheral, didReadRSSI: NSNumber, error: Error?)`

Invoked after you call `readRSSI()` to retrieve the value of the peripheral's current RSSI while it is connected to the central manager.

Monitoring Changes to a Peripheral's Name or Services

`func peripheralDidUpdateName(CBPeripheral)`

Invoked when a peripheral's name changes.

`func peripheral(CBPeripheral, didModifyServices: [CBSERVICE])`

Invoked when a peripheral's services have changed.

```
typedef enum {UndefinedState=0, InitializedState, SearchingState,
PeriphFoundState, ConnectedState} appState_t;
```

```
enum AppState {case undefinedState, initializedState, searchingState,
periphFoundState, connectedState}
```

```
case ConnectedState:
[_searchBtn setEnabled:NO];
[_connectBtn setEnabled:NO];
[_descBtn setEnabled:YES];
[_readBtn setEnabled:NO];
[_disconBtn setEnabled:YES];

[_periphLab setText:_periphName];
[_periphLab setTextColor:[UIColor greenColor]];
break;
```

3

```
case .connectedState:
searchBtn?.isEnabled = false
connectBtn?.isEnabled = false
descBtn?.isEnabled = true
readBtn?.isEnabled = false
disconBtn?.isEnabled = true

periphLab?.text = periphName
periphLab?.textColor = UIColor.green
```

```
_appState = ConnectedState;
// Update interface
[self updateInterface];
```

2

```
appState = .connectedState
// Update interface
updateInterface()
```

```
#define BEEWI_SERVICE_1_String @"A8B3FA04-4834-4051-89D0-3DE95CDDD318"
```

2 |

```
private let BEEWI_SERVICE_1_String = "A8B3FA04-4834-4051-89D0-3DE95CDDD318"
```

```
typedef enum {UndefinedState=0, InitializedState, SearchingState,  
PeriphFoundState, ConnectedState, GettingDescState} appState_t;
```

1 |

```
enum AppState {case undefinedState, initializedState, searchingState,  
periphFoundState, connectedState, gettingDescState}
```

```
@property NSArray *services;
```

2 |

```
private var services: [CBUUID]=[]
```

```
_services = @[[[UUID UUIDWithString:BEEWI_SERVICE_1_String]]];
```

2 |

```
services.append(CBUUID(string:BEEWI_SERVICE_1_String))
```

```
case GettingDescState:  
[_searchBtn setEnabled:NO];  
[_connectBtn setEnabled:NO];  
[_descBtn setEnabled:NO];  
[_readBtn setEnabled:NO];  
[_disconBtn setEnabled:YES];  
  
[_periphLab setText:_periphName];  
[_periphLab setTextColor:[UIColor greenColor]];  
break;
```

3 |

```
case .gettingDescState:  
    searchBtn?.isEnabled = false  
    connectBtn?.isEnabled = false  
    descBtn?.isEnabled = false  
    readBtn?.isEnabled = false  
    disconBtn?.isEnabled = true  
  
    periphLab?.text = periphName  
    periphLab?.textColor = UIColor.green
```

```
// Read service and characteristic description  
[_thermPeripheral discoverServices:_services];  
  
_appState = GettingDescState;  
// Update interface  
[self updateInterface];
```

2 |

```
// Read service and characteristic description  
thermPeripheral?.discoverServices(services)  
  
appState = .gettingDescState  
// Update interface  
updateInterface()
```

Code



```
for (CBService *service in peripheral.services) {  
    NSLog(@"Discovered service %@", service);  
    NSLog(@"Discovering characteristics for service %@", service);  
    [peripheral discoverCharacteristics:nil forService:service];  
}
```

```
for service in peripheral.services! {  
    print("Discovered service \(service)")  
    print("Discovering characteristics for service %@", service)  
    peripheral.discoverCharacteristics(nil, for: service)  
}
```



Code



```
#define BEEWI_DATA_Characteristic @"A8B3FB43-4834-4051-89D0-3DE95CDDD318"
```

```
typedef enum {UndefinedState=0, InitializedState, SearchingState,  
PeriphFoundState, ConnectedState, GettingDescState, DescFoundState}  
appState_t;
```

```
@property CBCharacteristic *dataChar;
```

```
_dataChar = nil;
```

```
case DescFoundState:  
[_searchBtn setEnabled:NO];  
[_connectBtn setEnabled:NO];  
[_descBtn setEnabled:NO];  
[_readBtn setEnabled:YES];  
[_disconBtn setEnabled:YES];  
  
[_periphLab setText:_periphName];  
[_periphLab setTextColor:[UIColor greenColor]];
```

```
for (CBCharacteristic *characteristic in service.characteristics) {  
    NSLog(@"Discovered characteristic %@", characteristic);  
    if ([[characteristic UUID] UUIDString]  
        compare:BEEWI_DATA_Characteristic] == NSOrderedSame) {  
        _dataChar = characteristic;  
  
        _appState = DescFoundState;  
  
        // Update GUI  
        [self updateInterface];  
    }  
}
```

```
private let BEEWI_DATA_Characteristic = "A8B3FB43-4834-4051-89D0-3DE95CDDD318"
```

4 enum AppState {case undefinedState, initializedState, searchingState, periphFoundState, connectedState, gettingDescState, descFoundState}

2 private var dataChar: CBCharacteristic?

3 dataChar = nil;

5 case .descFoundState:
 searchBtn?.isEnabled = false
 connectBtn?.isEnabled = false
 descBtn?.isEnabled = false
 readBtn?.isEnabled = true
 disconBtn?.isEnabled = true

 periphLab?.text = periphName
 periphLab?.textColor = UIColor.green

6 for characteristic in service.characteristics! {
 print("Discovered characteristic %@", characteristic)
 if characteristic.uuid.uuidString.compare(BEEWI_DATA_Characteristic)
 == .orderedSame {
 dataChar = characteristic

 gameState = .descFoundState

 // Update GUI
 updateInterface()
 }
 }

```
_appState = PeriphFoundState;  
// Update interface  
[self updateInterface];
```

2

```
appState = .periphFoundState  
// Update interface  
updateInterface()
```

```
// Disconnect  
[_centralManager cancelPeripheralConnection:_thermPeripheral];  
_dataChar = nil;
```

```
// Disconnect  
centralManager?.cancelPeripheralConnection(thermPeripheral!)  
dataChar = nil
```

Code



```
typedef enum {UndefinedState=0, InitializedState, SearchingState,  
PeriphFoundState, ConnectedState, GettingDescState, DescFoundState,  
ReadingState} appState_t;
```

```
if ((thermPeripheral != nil) && (dataChar != nil)) {  
    [_tempLab setText:[NSString stringWithFormat:@"Température %.1f °C",  
    temperature]];  
    [_humidLab setText:[NSString stringWithFormat:@"Humidité %d %%",  
    humidity]];  
    [_batLab setText:[NSString stringWithFormat:@"Batterie %d %%",  
    battery]];  
} else {  
    [_tempLab setText:@"Température ---."];  
    [_humidLab setText:@"Humidité ---."];  
    [_batLab setText:@"Batterie ---."];  
}
```

```
case ReadingState:  
    [_searchBtn setEnabled:NO];  
    [_connectBtn setEnabled:NO];  
    [_readBtn setEnabled:YES];  
    [_disconBtn setEnabled:YES];  
  
    [_periphLab setText:_periphName];  
    [_periphLab setTextColor:[UIColor greenColor]];  
    break;
```

```
if ((thermPeripheral != nil) && (dataChar != nil)) {  
    _appState = ReadingState;  
    // Update interface  
    [self updateInterface];  
    [_thermPeripheral readValueForCharacteristic:_dataChar];  
}
```

```
enum AppState {case undefinedState, initializedState,  
searchingState, periphFoundState, connectedState, gettingDescState,  
descFoundState, readingState}
```

```
if ((thermPeripheral != nil) && (dataChar != nil)) {  
    tempLab?.text = String(format:"Température %.1f °C", temperature)  
    humidLab?.text = String(format:"Humidité %ld %%", humidity)  
    batLab?.text = String(format:"Batterie %ld %%", battery)  
} else {  
    tempLab?.text = "Température ---."  
    humidLab?.text = "Humidité ---."  
    batLab?.text = "Batterie ---."  
}
```

```
case .readingState:  
    searchBtn?.isEnabled = false  
    connectBtn?.isEnabled = false  
    descBtn?.isEnabled = false  
    readBtn?.isEnabled = true  
    disconBtn?.isEnabled = true  
  
    periphLab?.text = periphName  
    periphLab?.textColor = UIColor.green
```

```
if ((thermPeripheral != nil) && (dataChar != nil)) {  
    AppState = .readingState  
    // Update interface  
    updateInterface()  
    thermPeripheral?.readValue(for: dataChar!)  
}
```

[ObjC]



Code



```
if ([characteristic UUID] == [_dataChar UUID]) {
    NSData *data = characteristic.value;
    if ([data length] != 10) {
        NSLog(@"bad data length %d (should be 20)", (int)[data length]);
    } else {
        /*# the temperature consists of 3 bytes
        # Positive value: byte 1 & 2 present the tenfold of the temperature
        # Negative value: byte 2 - byte 3 present the tenfold of the
        temperature */
        int temperature_buf[3];
        char *ptr_data_buf = (char *)[data bytes];
        temperature_buf[0] = ((int)ptr_data_buf[0]) & 0x000000ff;
        temperature_buf[1] = ((int)ptr_data_buf[1]) & 0x000000ff;
        temperature_buf[2] = ((int)ptr_data_buf[2]) & 0x000000ff;
        int temperatureInt = (temperature_buf[2]*255)+temperature_buf[1];
        _temperature = (float)temperatureInt / 10.0;

        char humidity_buf[1];
        humidity_buf[0] = ptr_data_buf[4];
        _humidity = (char)*humidity_buf;

        char battery_buf[1];
        battery_buf[0] = ptr_data_buf[9];
        _battery = (char)*battery_buf;

        _appState = DescFoundState;

        // Update GUI (labels)
        [self updateInterface];
    }
}
```

```
if characteristic.uuid == dataChar?.uuid {
    let data : Data = characteristic.value!
    if data.count != 10 {
        print("bad data length \(data.count) (should be 20)")
    } else {
        /*# the temperature consists of 3 bytes
        # Positive value: byte 1 & 2 present the tenfold of the
        temperature
        # Negative value: byte 2 - byte 3 present the tenfold of the
        temperature */
        let data_buf = [UInt8](data)
        var temperature_buf = [UInt8](repeating: 0, count: 3)
        temperature_buf[0] = data_buf[0] & 0x000000ff
        temperature_buf[1] = data_buf[1] & 0x000000ff
        temperature_buf[2] = data_buf[2] & 0x000000ff
        let temperatureInt : Int =
(Int)(temperature_buf[2])*255+Int(temperature_buf[1])
        temperature = Float(temperatureInt) / 10.0

        let humidity_buf: UInt8 = data_buf[4]
        humidity = Int(humidity_buf)

        let battery_buf : UInt8 = data_buf[9]
        battery = Int(battery_buf)

        AppState = .descFoundState

        // Update GUI (labels)
        updateInterface()
    }
}
```



RESSOURCES

ation Apple:

[e.com/documentation/](#)

[ntent/documentation/Swift/Conceptual/](#)

[ming_Language/](#)

oks:

[Language \(Swift 4 Edition\)](#)

[rogramming-language/id1002622538?mt=11](#)

C EDX:

OS (parties I et II)

[n/specifications/gatt/services](#)