

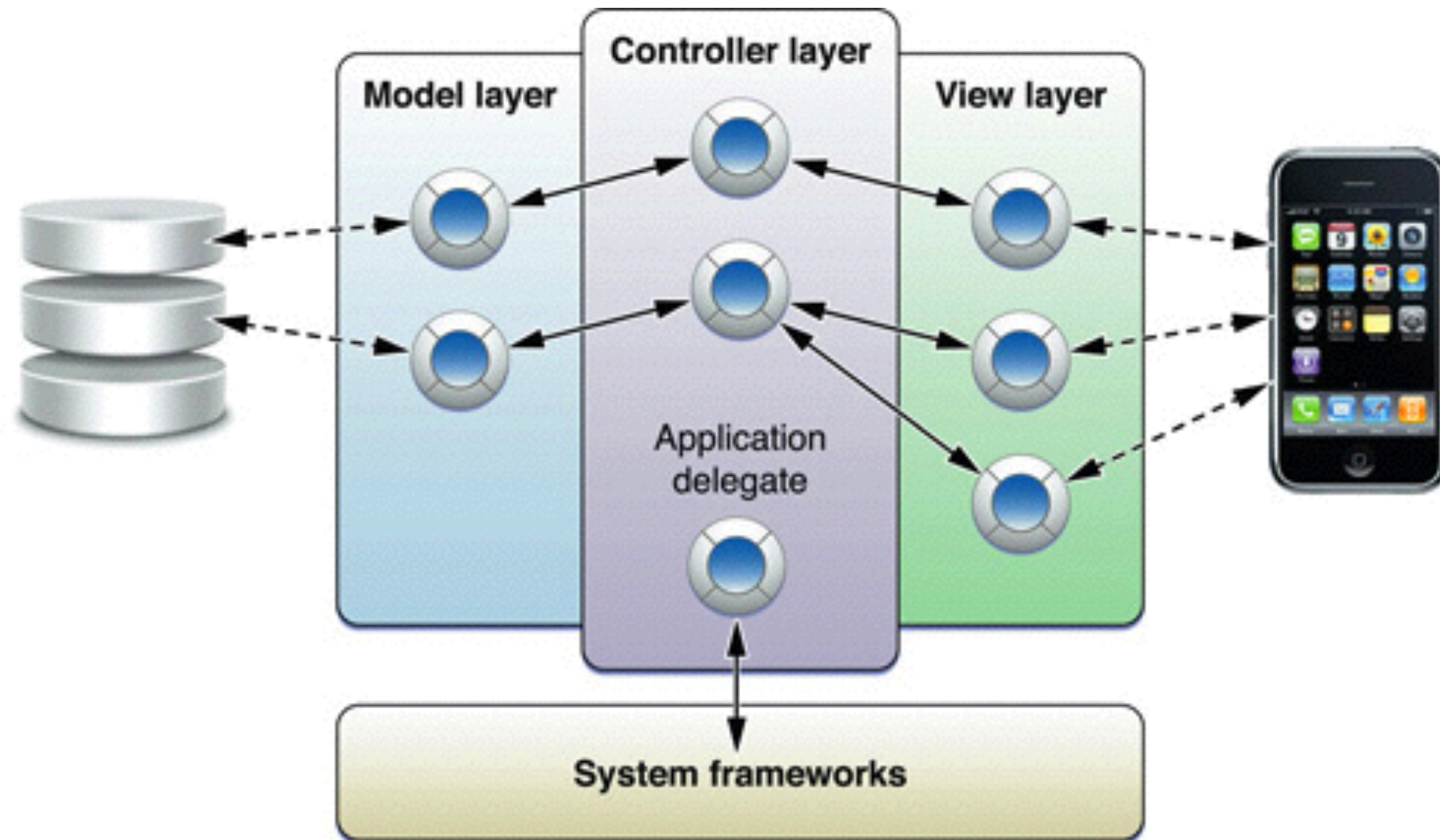
Cocoa

Jean-Baptiste.Yunes@univ-paris-diderot.fr

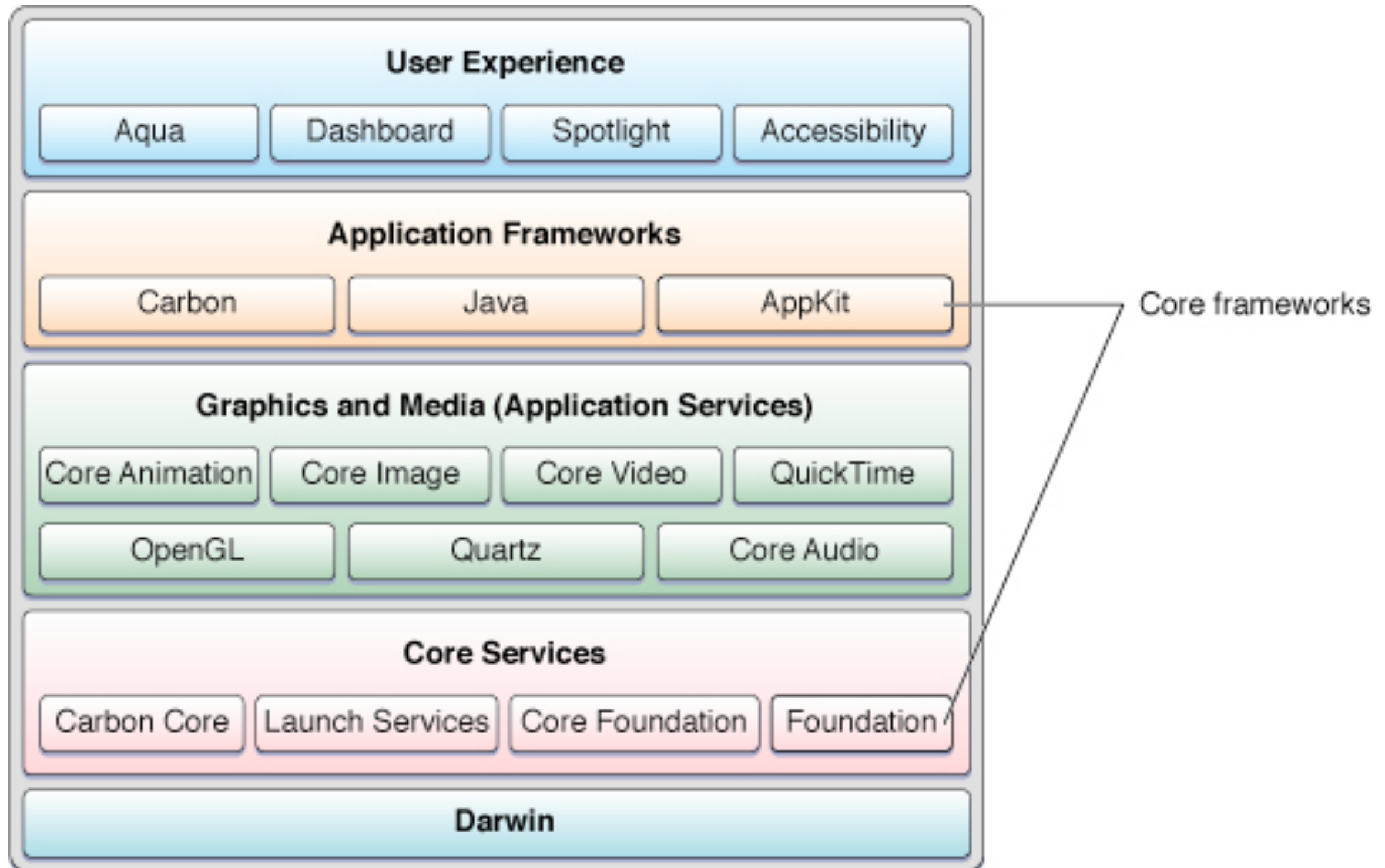
v 0.9

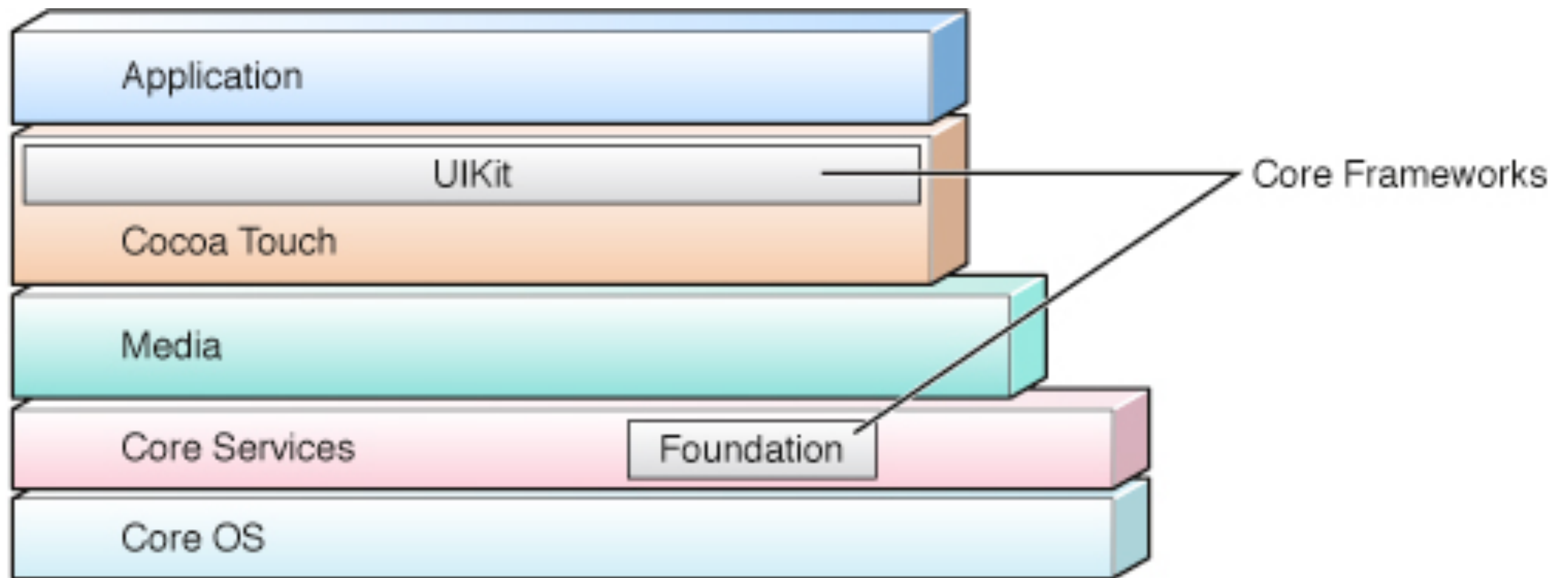


- Le modèle de Cocoa

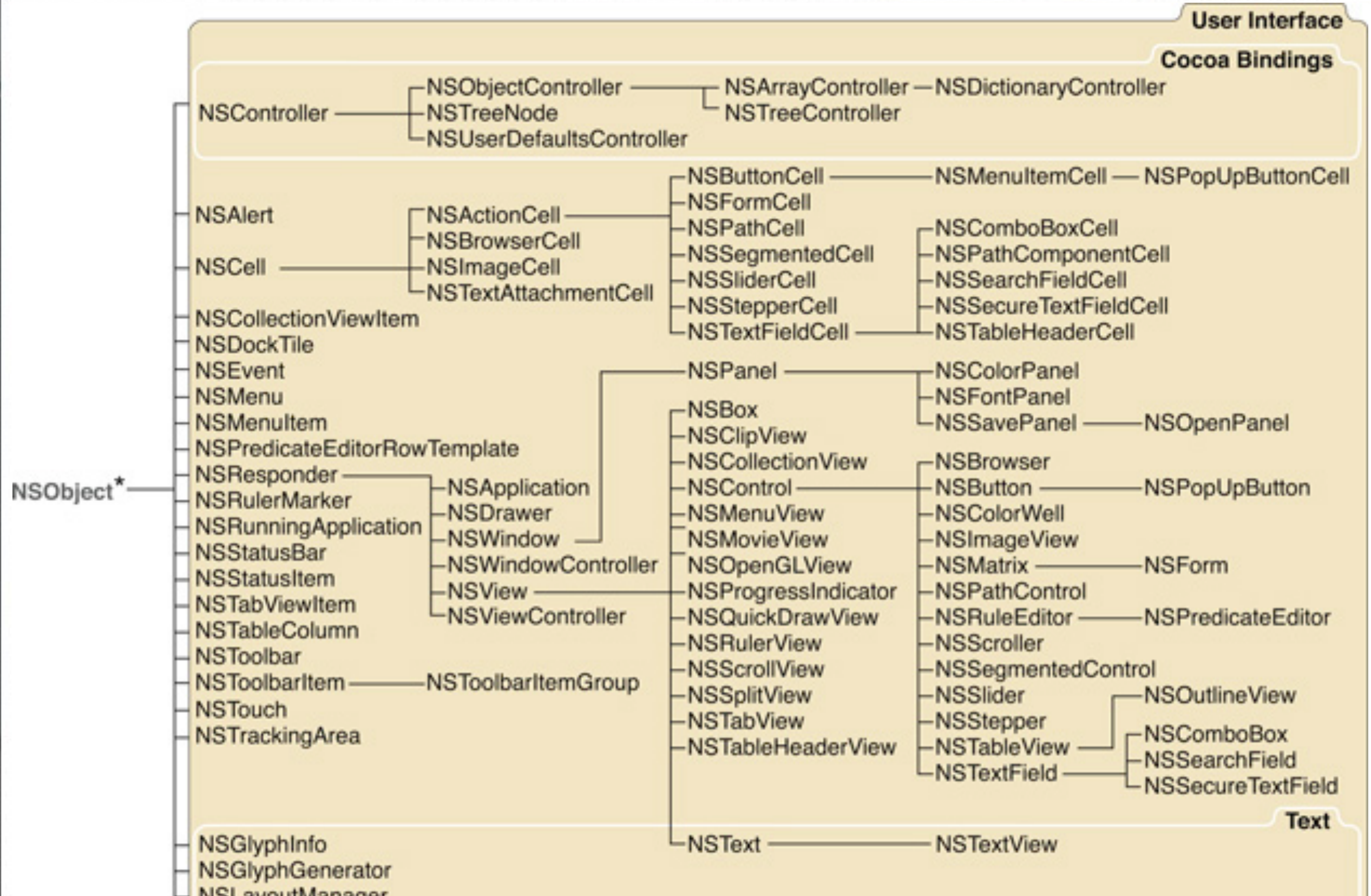


- deux Cocoa (pour le prix d'un) !
 - pour OSX
 - Foundation+AppKit
 - pour iOS
 - Foundation+UIKit
- Appkit : NS... (ex : NSButton)
- UIKit : UI... (ex : UIButton)

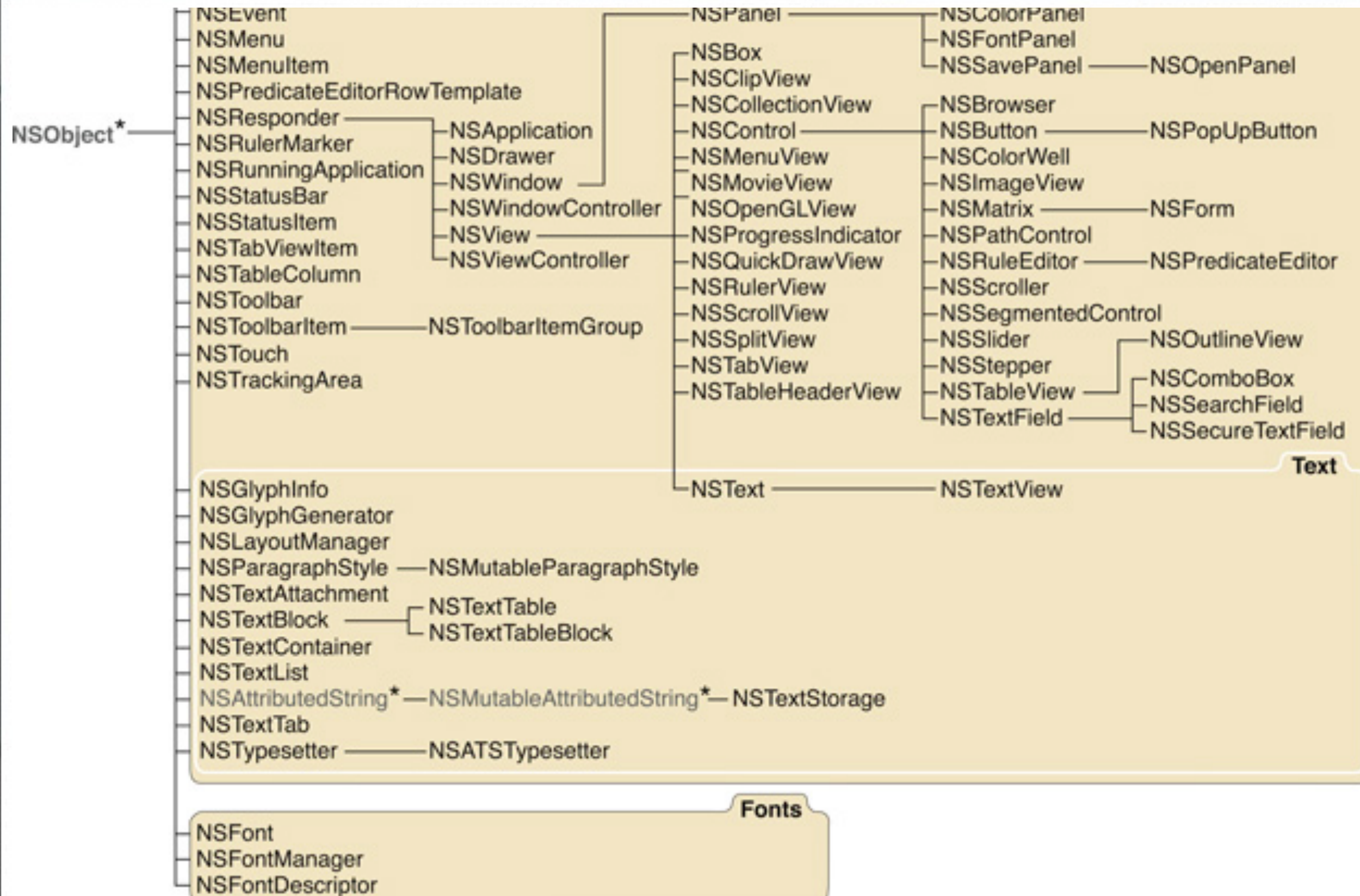




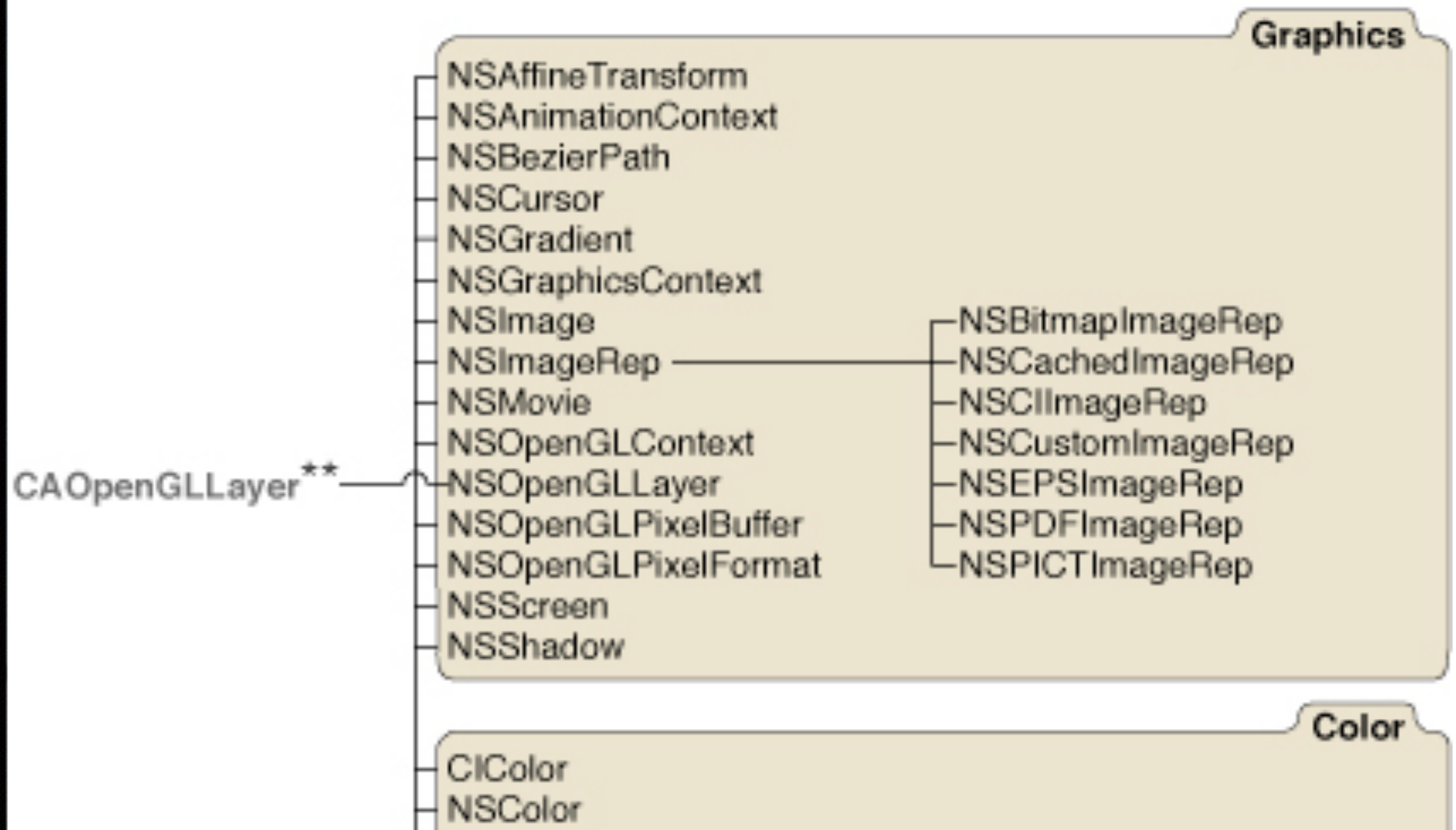
- pour simplifier donc :
- Cocoa c'est Foundation + l'interface graphique
- disons que ce sont les premières choses que l'on voit de Cocoa

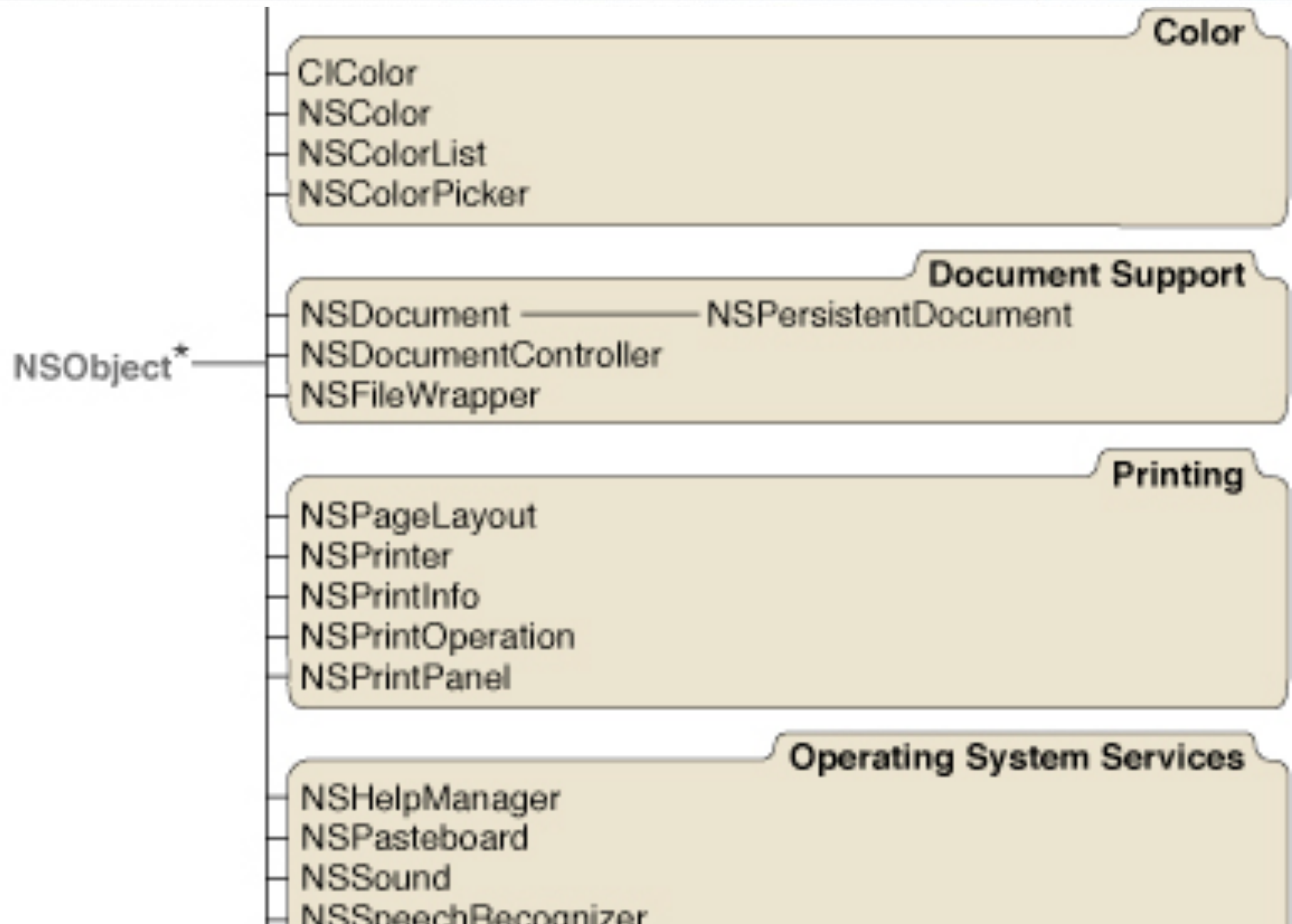


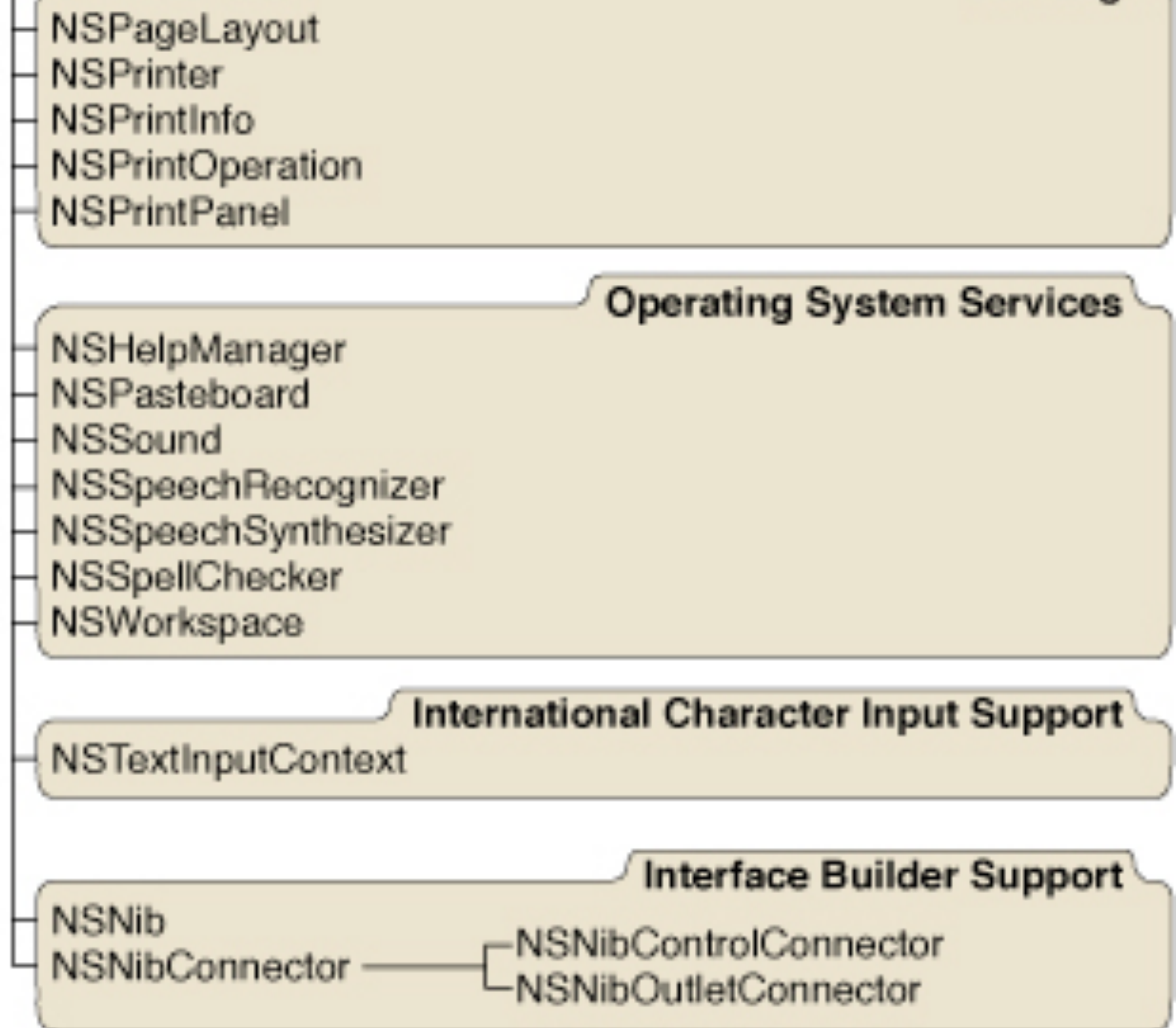
Introduction

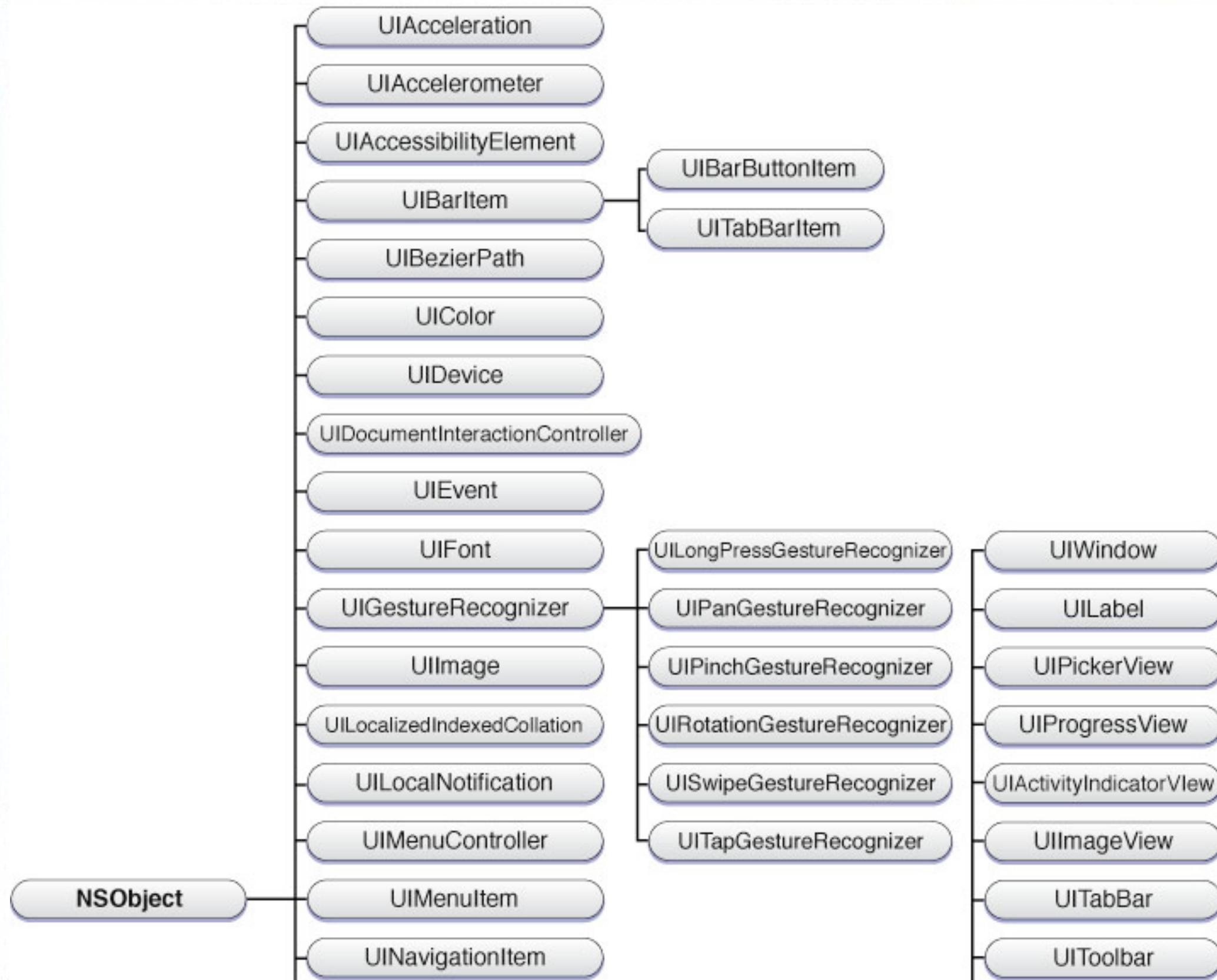


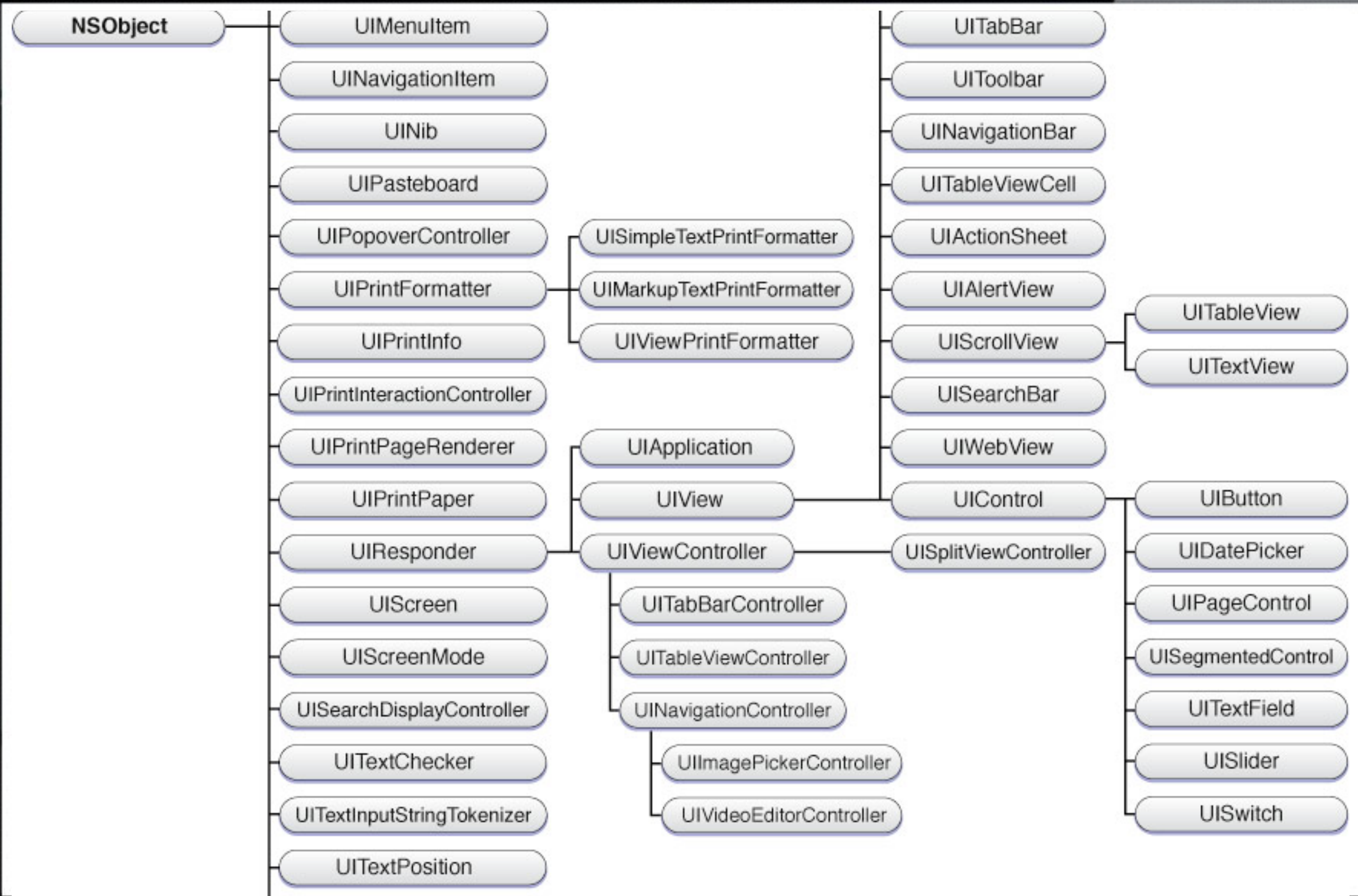
Objective-C Application Kit Continued



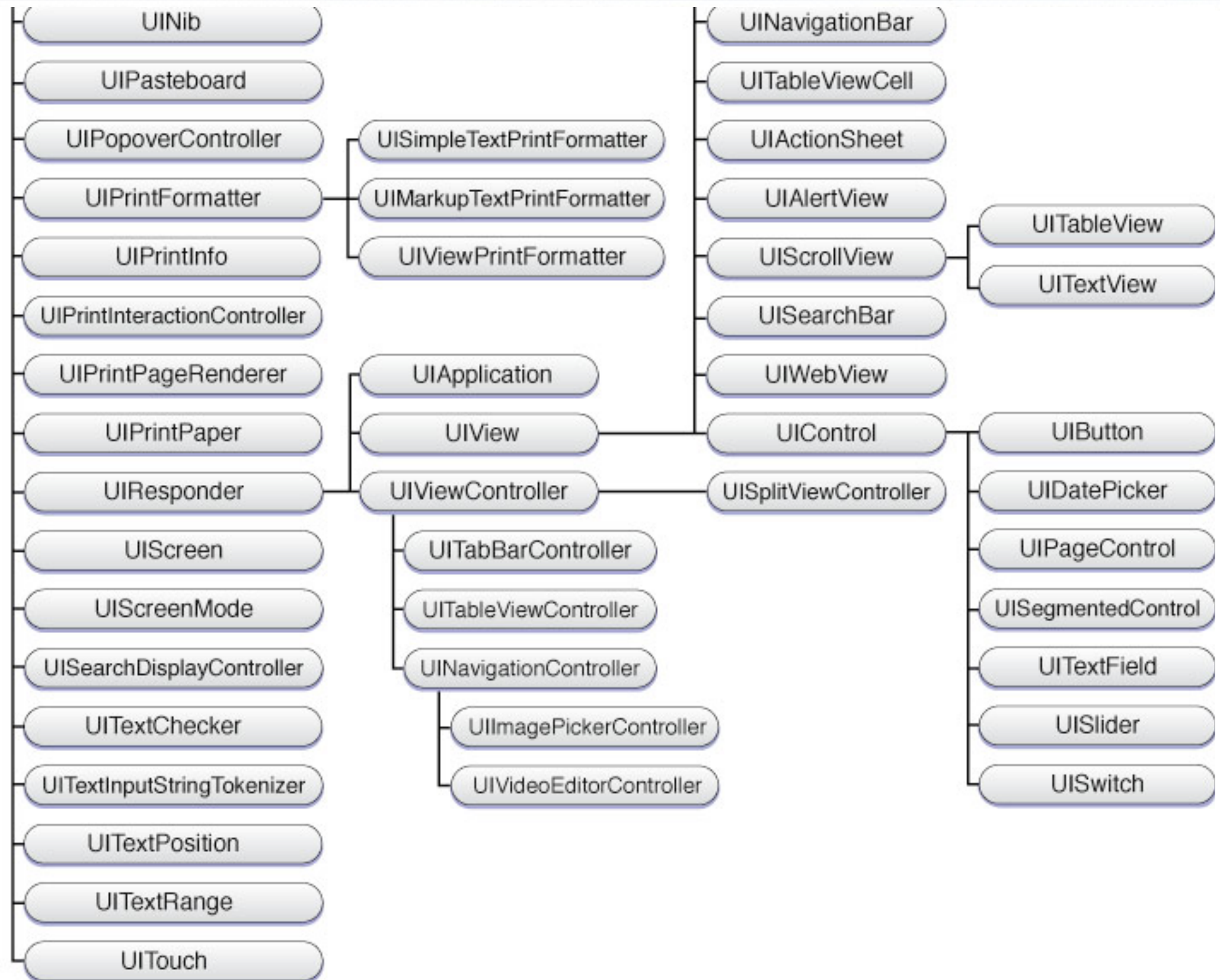








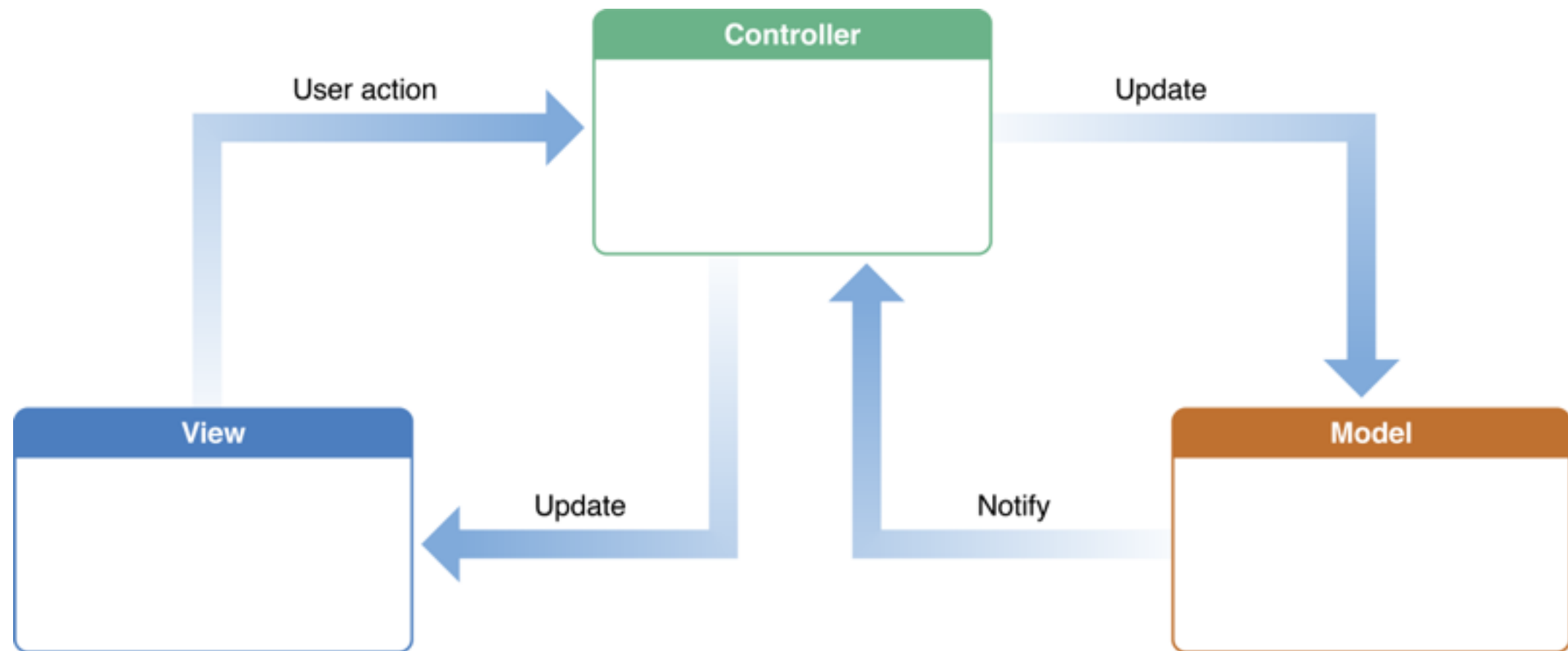
Introduction



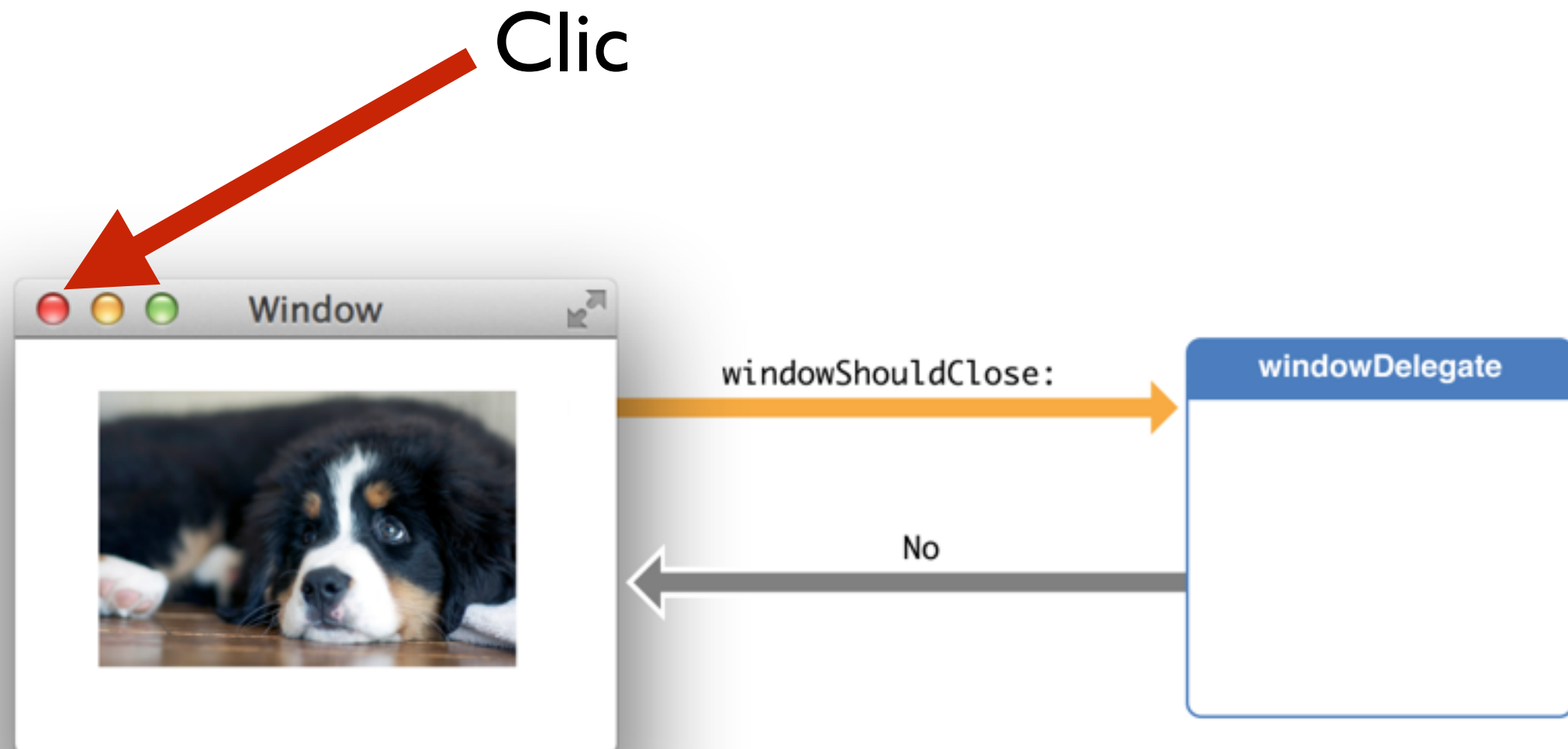
La conception d'applications

- La livraison d'une application conforme à l'ergonomie (au sens général) requiert des soins particuliers
- On ne traitera pas ici de toutes ses questions qui sont importantes
- La conception (au sens de la programmation, ce qui nous intéresse ici) nécessite la maîtrise de différents patterns employés par Cocoa

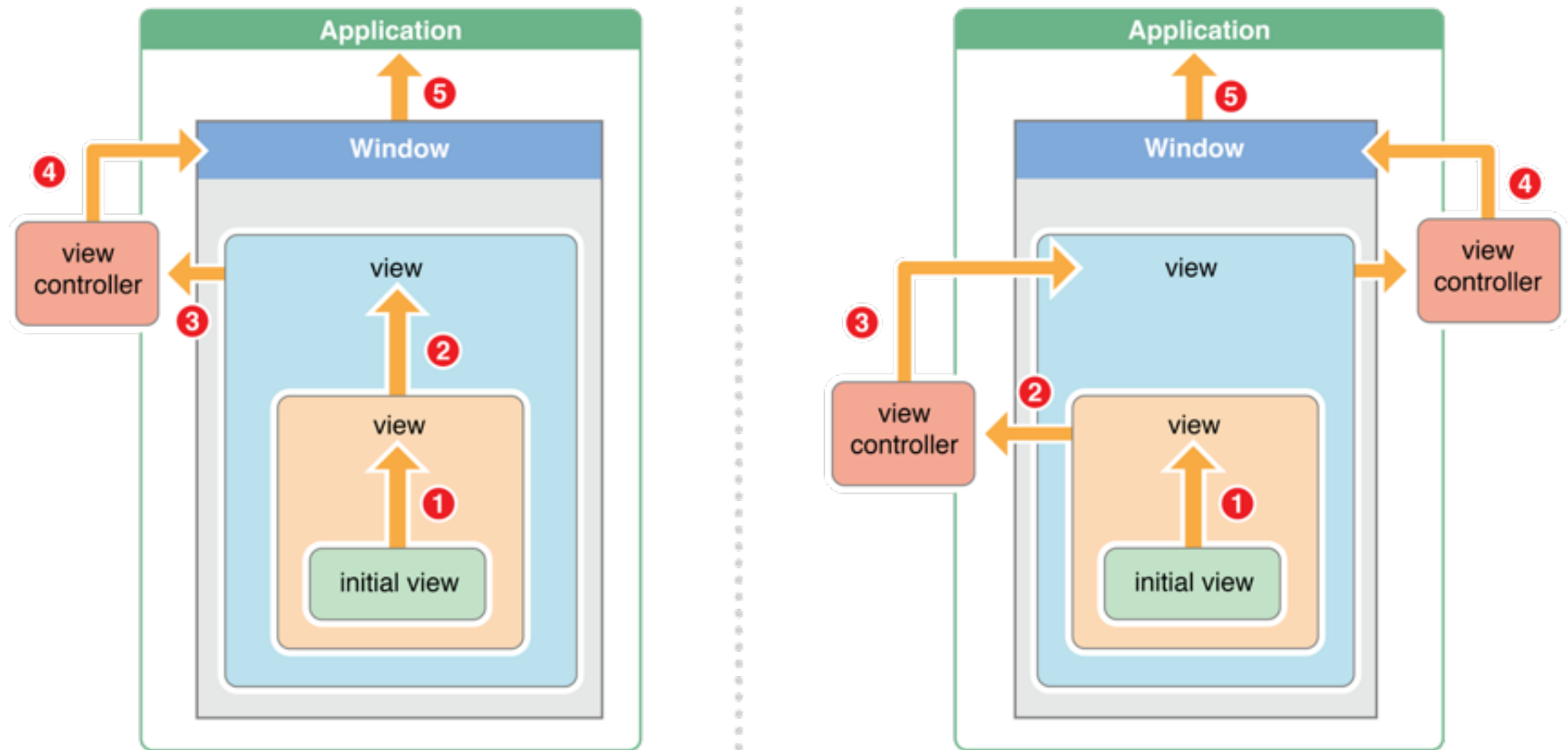
- Le MVC de Cocoa est le pattern fondamental principal



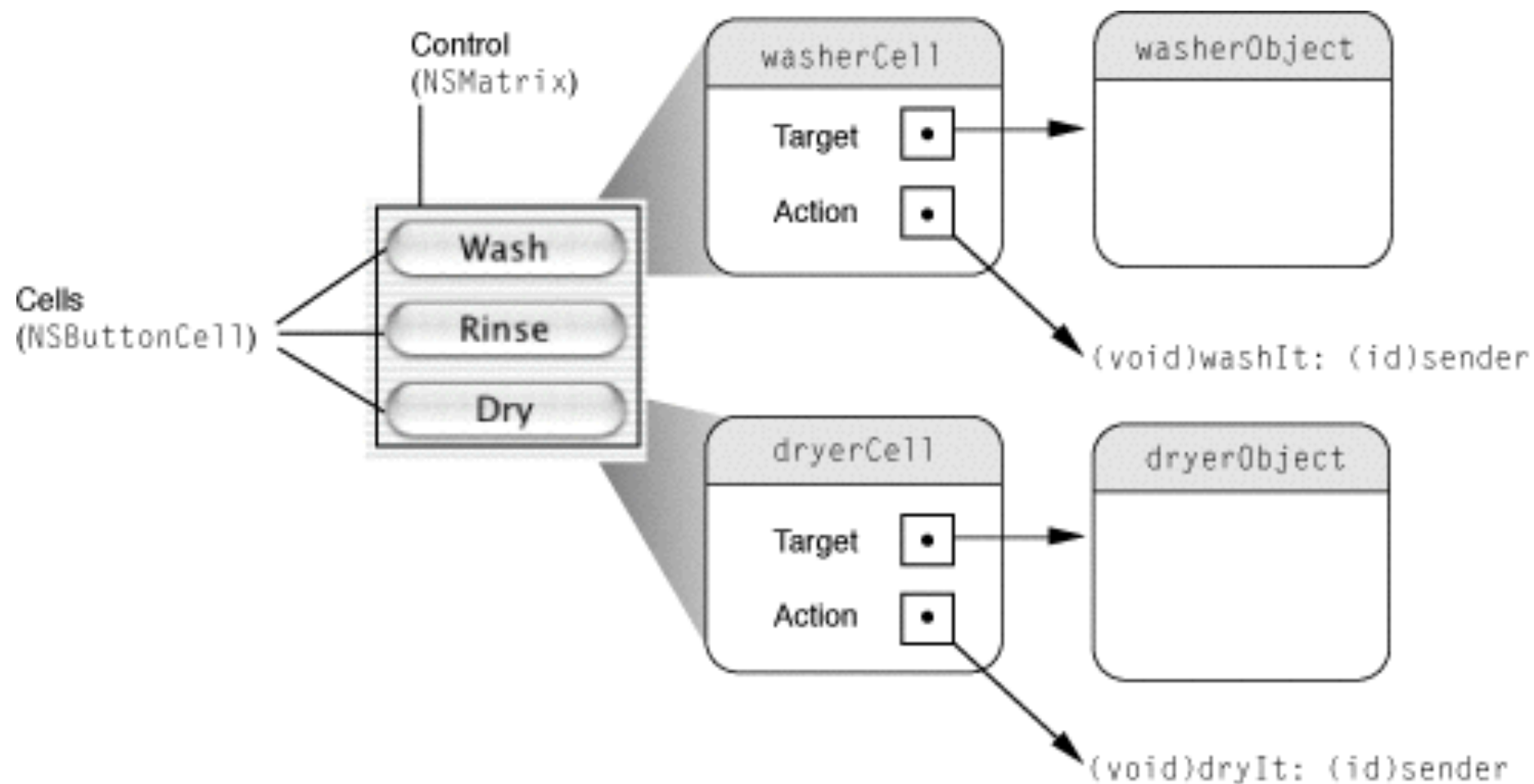
- La délégation qui est préférée à l'héritage



- Le pattern **responder chain** (chaîne de responsabilité) qui contrôle la distribution des événements vers les contrôles dans Cocoa

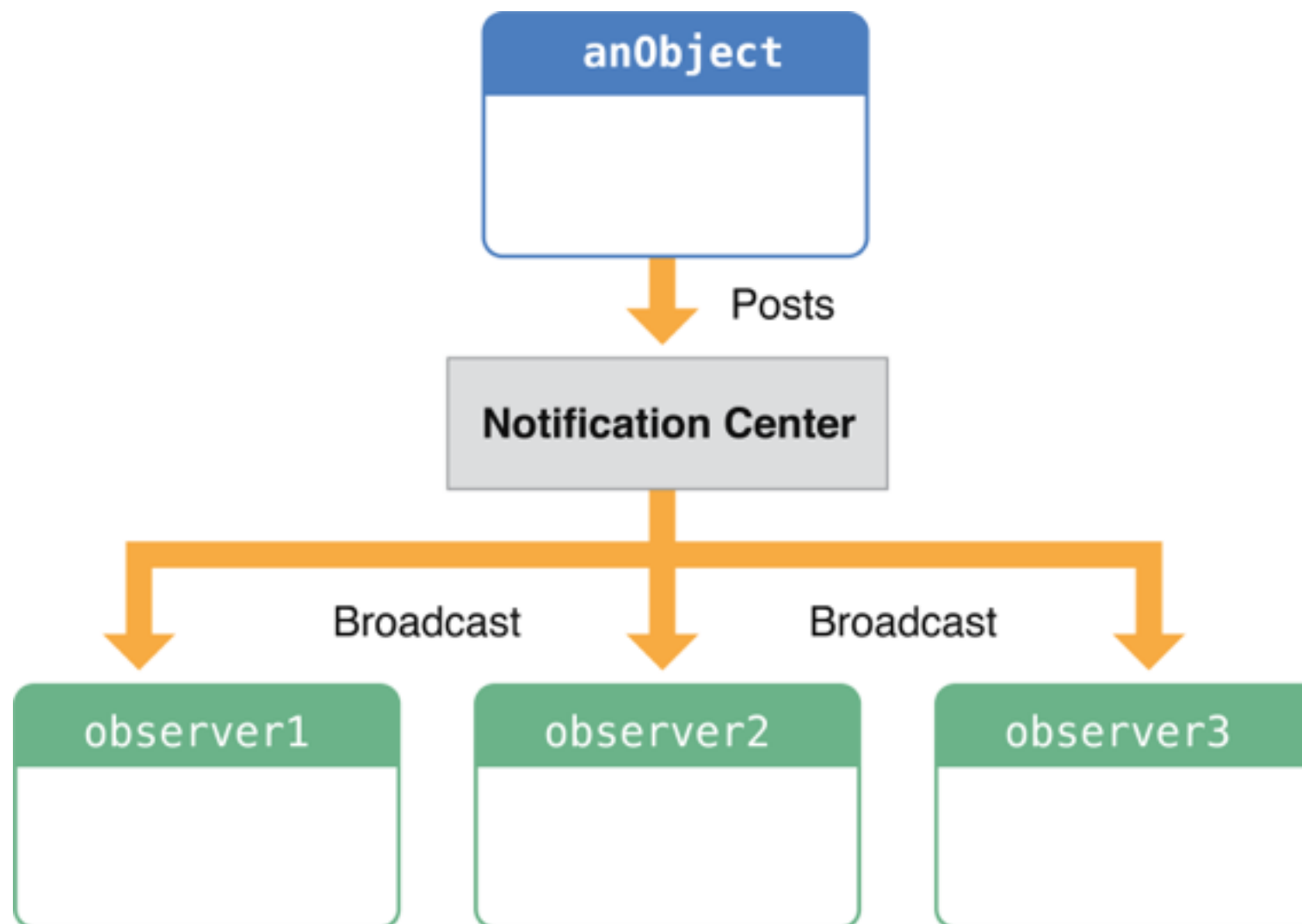


- Le pattern cible-action (**target-action**) qui établit un lien entre un contrôle (ex.: bouton) qui génère une action à destination d'un objet qui recevra l'action (la cible)

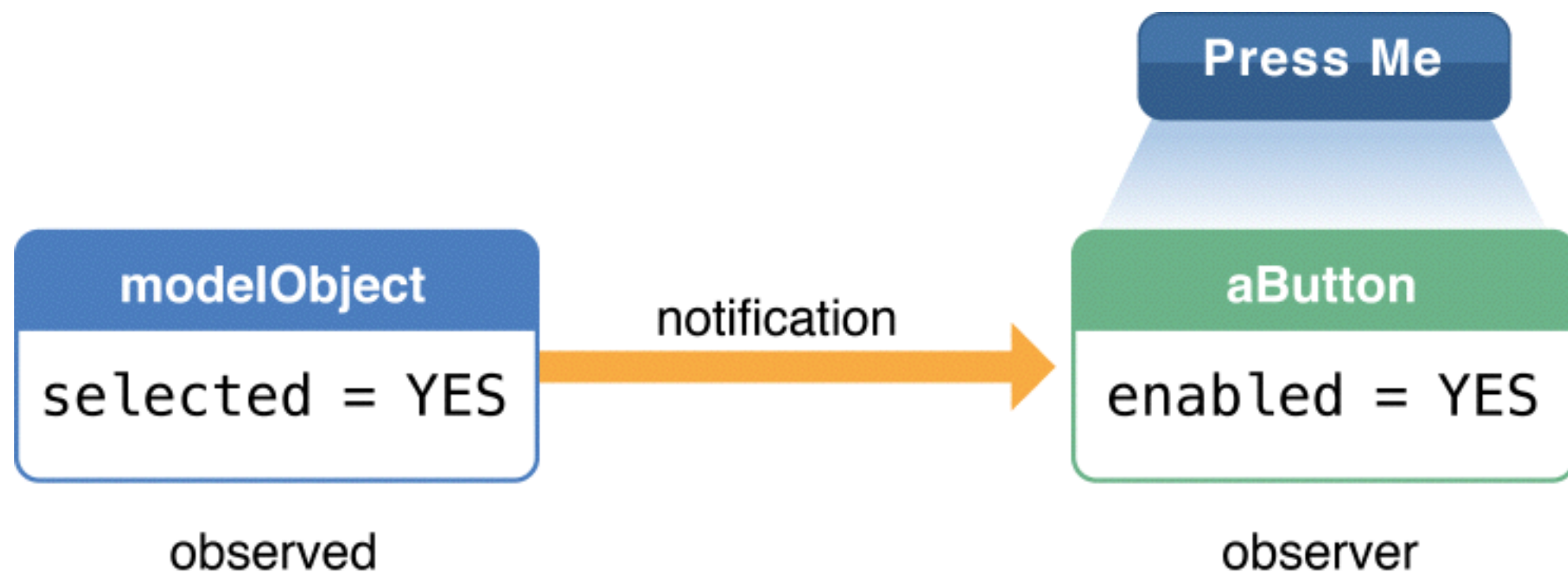


- Les blocs (en Swift les clôtures) qui encapsulent un environnement et un code dont l'exécution est différée

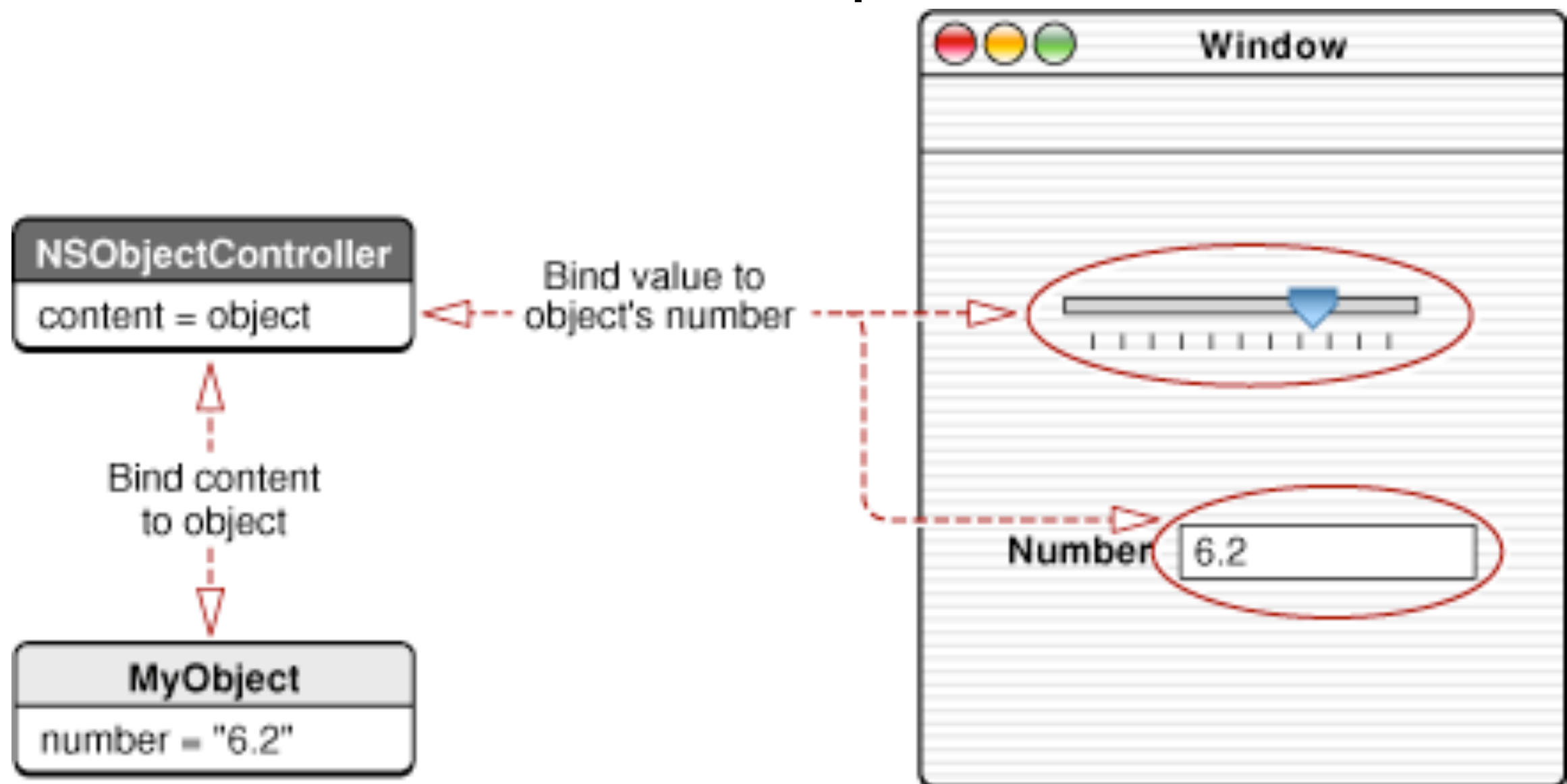
- Les notifications qui constituent un système de distribution de messages auxquels les objets peuvent s'abonner pour les recevoir.



- Le pattern **KVO Key-Value Observing** (observer) qui permet d'être notifié des modifications des propriétés (attributs) des objets

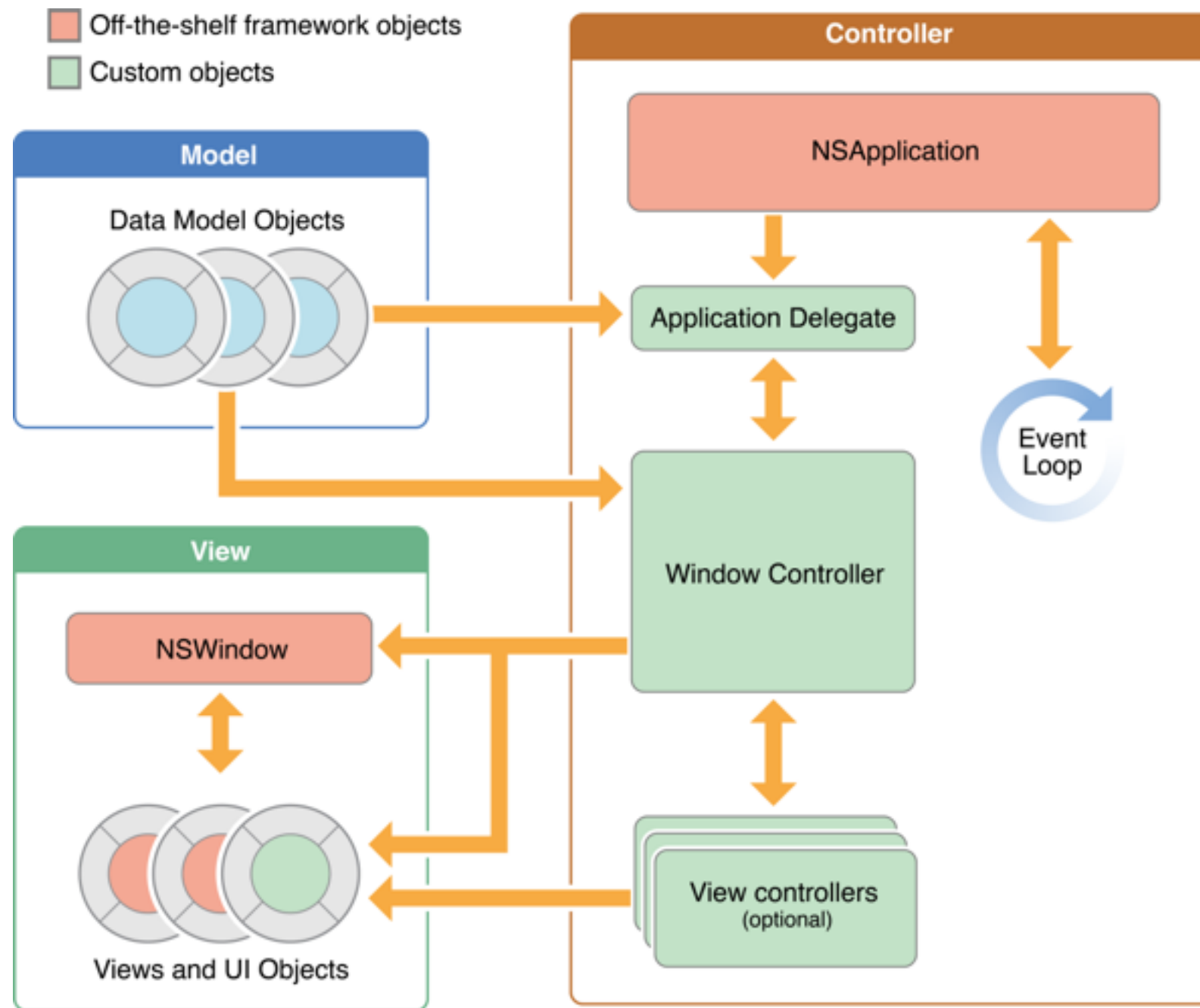


- Les **bindings** qui permettent de lier une vue à une propriété de sorte que la modification de l'un soit directement répercutée sur l'autre

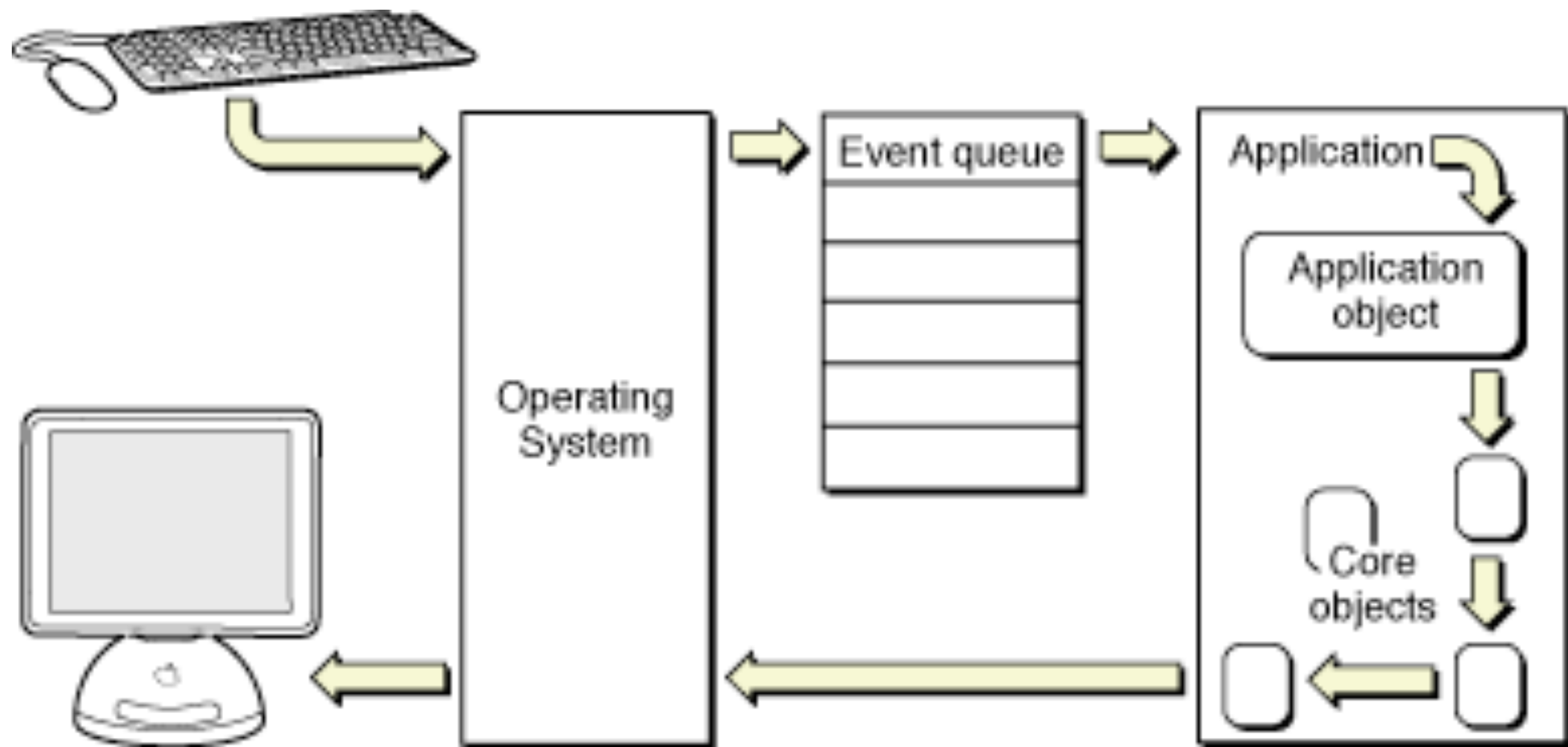


- On considère 3 types d'applications :
- l'utilitaire mono-fenêtré : le calendrier
- la bibliothèque mono-fenêtrée : le gestionnaire de photos ou de musique qui masque à l'utilisateur les fichiers manipulés
- l'application multi-fenêtrée : l'éditeur de texte

- L'anatomie d'une application Cocoa contient généralement les éléments suivants :



- L'objet `NSApplication` (requis) qui :
 - gère les interactions entre l'application et le système
 - lance la boucle d'événements
- Cet objet est utilisé nativement, les personnalisations étant prises en charge par son délégué



- L'objet délégué de l'application (conforme au protocole `NSApplicationDelegate`) dont la fonction est de :
 - créer les objet nécessaires au fonctionnement de l'application
 - prendre en charge les transitions entre états de l'application (lancement, terminaison, iconification, etc)

- Le contrôleur de fenêtre dont le rôle est de prendre en charge la gestion d'une fenêtre et de se coordonner en accord avec le système
- On peut sous-classer ce type

- La fenêtre à qui peut être associée des fenêtres secondaires (en général modales) et qui présente un menu, une barre de titre, etc.

- Le contrôleur de vue qui coordonne la gestion d'une hiérarchie de vues lorsque le contenu de la fenêtre est trop complexe pour que la logique soit intégralement contenue dans le contrôleur de fenêtre

- La vue qui représente une région rectangulaire et qui est responsable de son contenu et qui reçoit les événements survenues dans cette région
- Les vues peuvent se recouvrir pour former une hiérarchie; les plus hautes masquant tout ou partie de celles qui sont en dessous

- Les contrôles qui sont des sous-classes de vue standardisées offrant des fonctionnalités standard : boutons, champs, vue tabulaires, etc

- Les applications multi-fenêtrées utilisent deux concepts supplémentaires situés entre le délégué d'application et le contrôleur de fenêtres :
 - le contrôleur de document
 - le document
- Dans ce modèle d'application, une fenêtre représente un document

