





# Hello NSCoder I

Introducción al desarrollo para iOS

@nscoder\_bcn

# NSCoder

- Profesionales y entusiastas del mundo de la manzana
- NSCoder Night
  - Cada jueves 19:00 - Frankfurt de Pelayo
- Twitter: [@nscoder\\_bcn](https://twitter.com/nscoder_bcn)
- [http://groups.google.com/group/nscoder\\_bcn](http://groups.google.com/group/nscoder_bcn)
- <http://nscoder-bcn.tumblr.com/>



# Programa (Mañana)

- Introducción iOS “Hello NSCoder”
- Web Service “What’s up, NSCoder”
- Gestión de Memoria
- Debugging and Memory Leaks
- Testing



# Programa (Tarde)

- Presentación de Omnis Studio
- Introducción a la programación con Omnis Studio
- Programación de una aplicación cliente Omnis con XCode para iPhone/iPad



# Referencias

- Material (github)
  - <https://github.com/NSCoderNightBCN>Hello-NSCoder>
- Apple Developer
  - <http://developer.apple.com/>
- NSCodeCenter
  - <http://www.nscodecenter.com/>
- 85% Cocoa
  - <http://ochentaycincoporcientococoa.tumblr.com/>



# Resultado

- App que nos localiza
- Conecta a un Web Service
- Devuelva eventos a nuestros alrededores
- TableView
- Detail View



[NSCoderBCN event:@”Hello NSCoder” date:28/05/11]





# Introducción iOS

Alfonso @setfilesl  
Andreas @aquarioverde

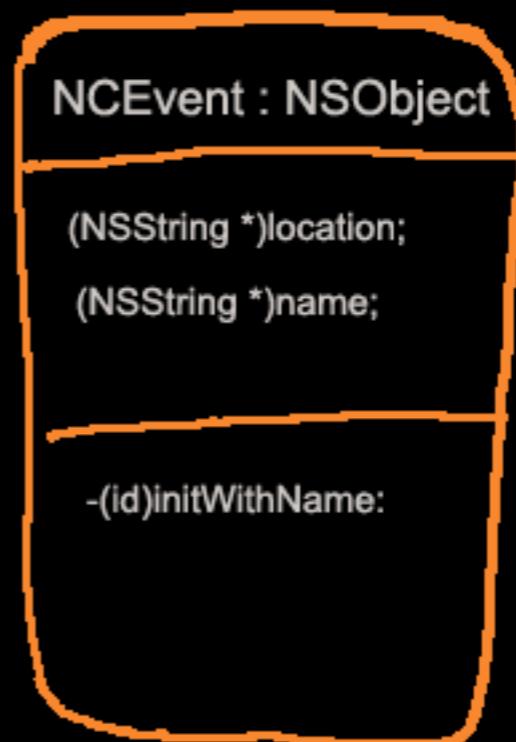
# Requerimientos

- Conocimientos Objective C / iOS
- MacOS X
- iOS SDK
- XCode / Interface Builder



# Objective C

- Class / Instance
- Variable / Properties
- Method / Message
- .h / .m



```
- (id)initWithName:(NSString *)name;  
+ (id)eventFromPList:(NSString *)listPath;
```



# Objective C

- Memory Allocation
  - release / retain / nil
- Message
  - receiver / selector / arguments

```
NCEvent *event = [[NCEvent alloc] initWithNibName:name  
andLocation:location];  
  
[event release];  
event = nil;
```



# Objective C

- Properties
  - property / synthesize
  - nonatomic
  - copy, retain, assign, readonly

```
@property (nonatomic, copy) NSString *name;  
@synthesize name;  
  
self.name = @"Hello NSCoder";  
[self setName:@"Hello NSCoder"];
```



# Objective C

```
@interface NCEvent : NSObject {  
}  
  
@property (nonatomic, copy) NSString *name;  
  
- (id)initWithName:(NSString *)name;
```

```
@end
```

```
@implementation NCEvent  
@synthesize name;
```

```
- (id)initWithName:(NSString *)name {  
    ...  
}
```

```
@end
```

.h

.m



# Objective C

- Inheritance
- Protocol

```
@protocol NCEventDelegate <NSObject>
```

```
@required
```

```
- (void)didFinishUpdatingData;
```

```
@end
```

.h

```
@interface NCController : NSObject <NCEventDelegate> {
```

```
}
```

```
@end
```

.h



# iOS

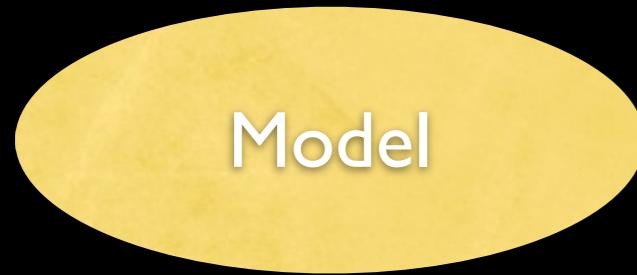
- Frameworks
  - Foundation
  - UIKit
- UIView / UITableView
- UINavigationBar
  - root / push / pop



# iOS - MVC



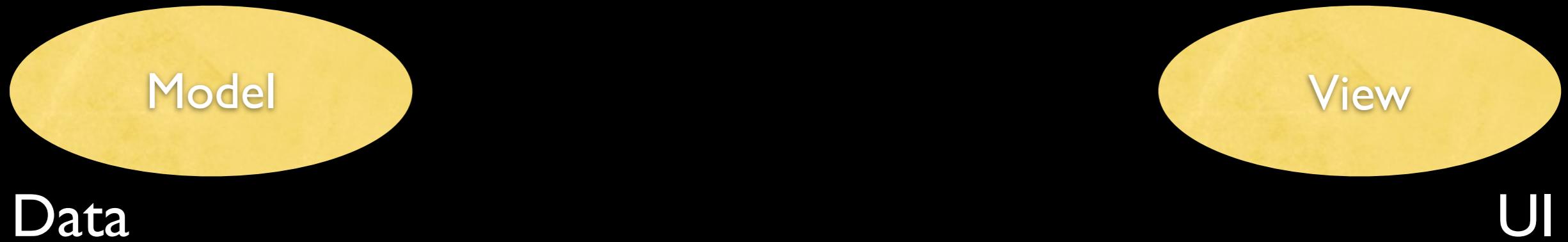
# iOS - MVC



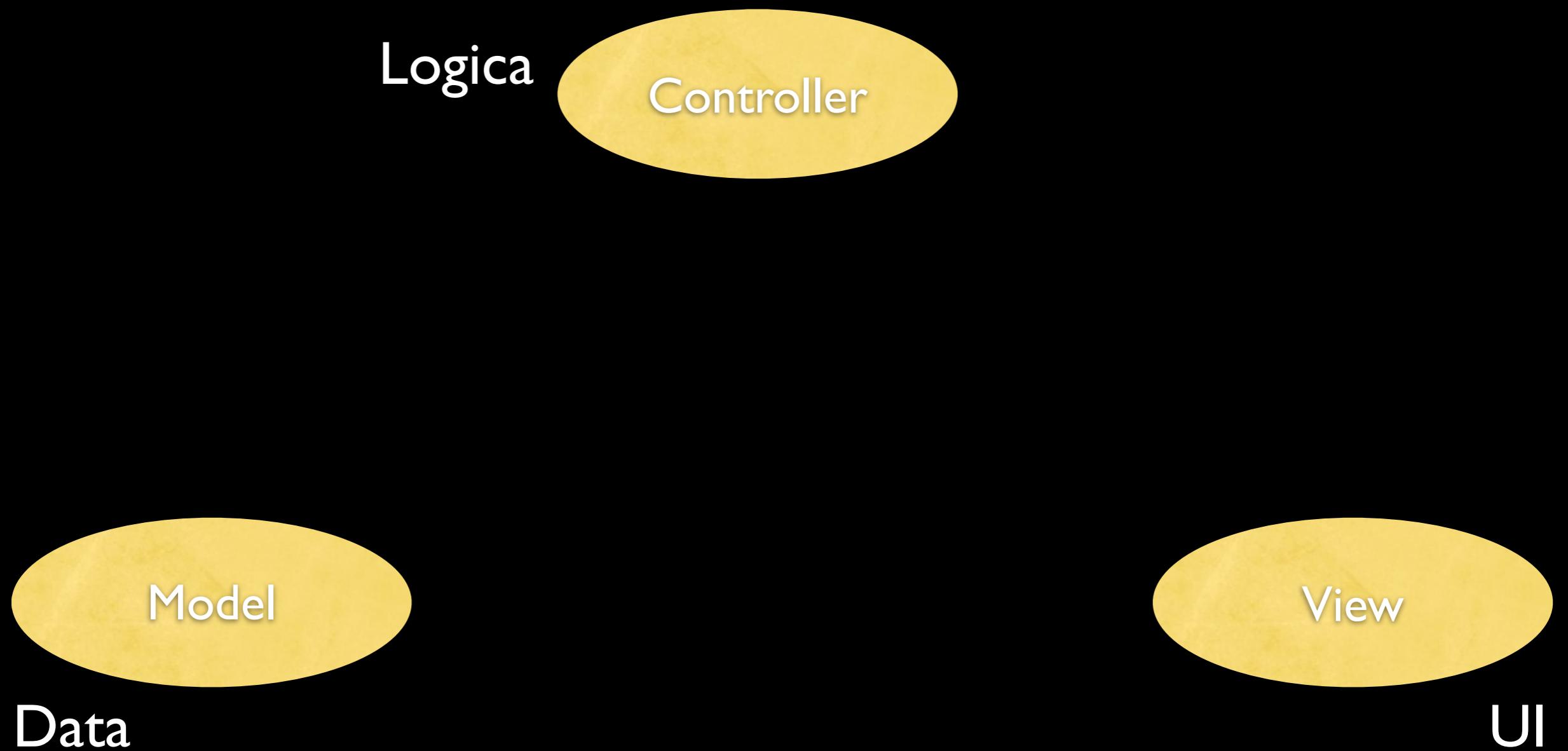
Data



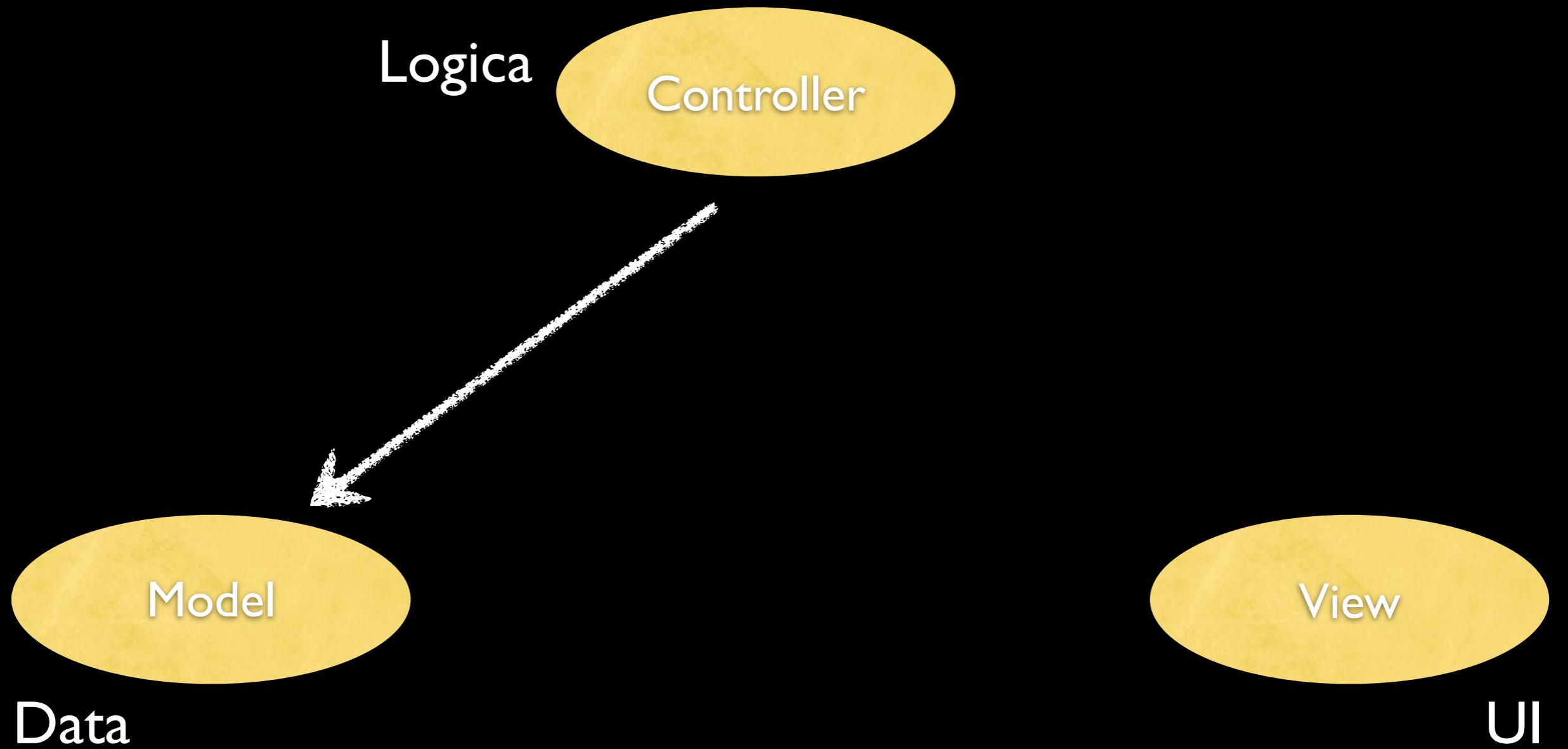
# iOS - MVC



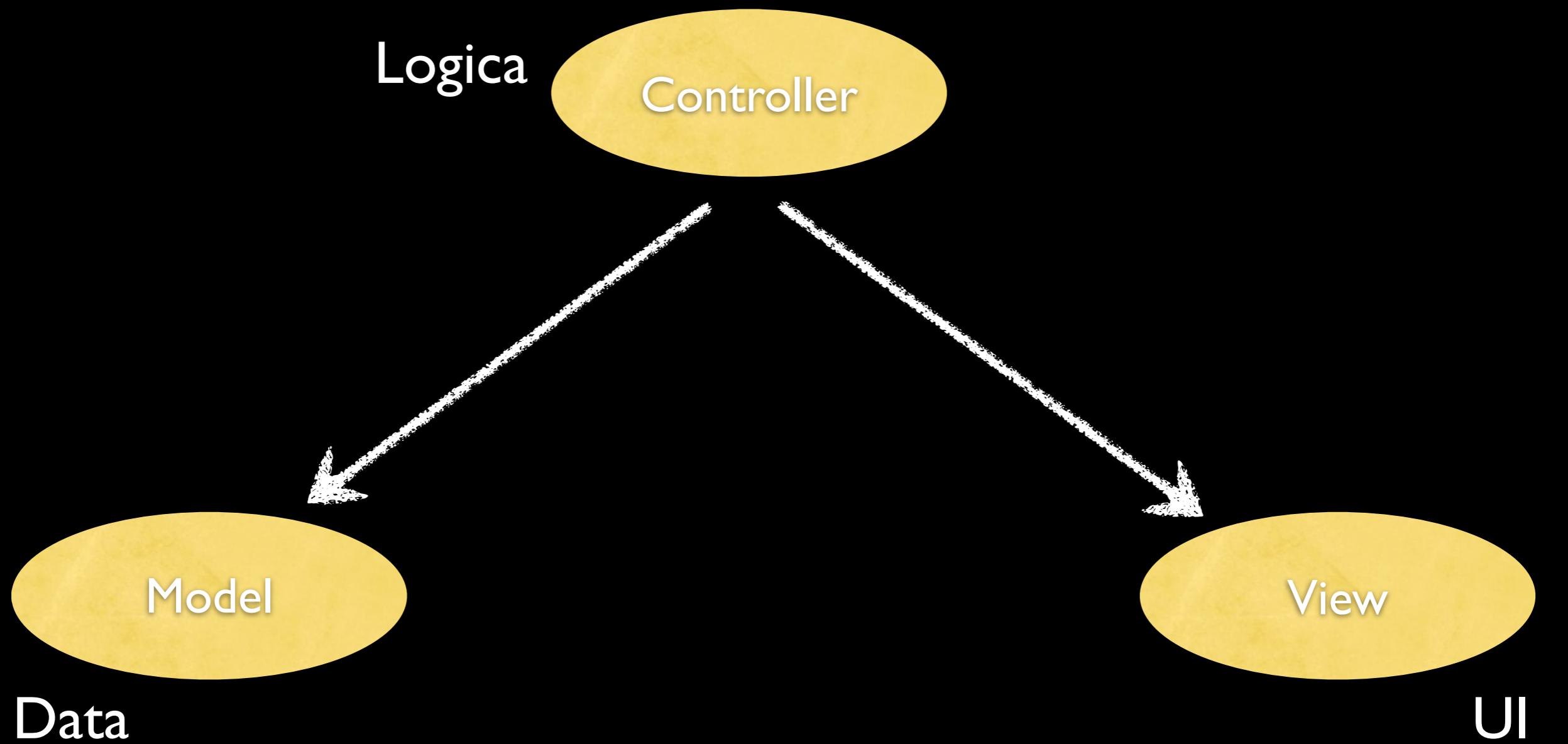
# iOS - MVC



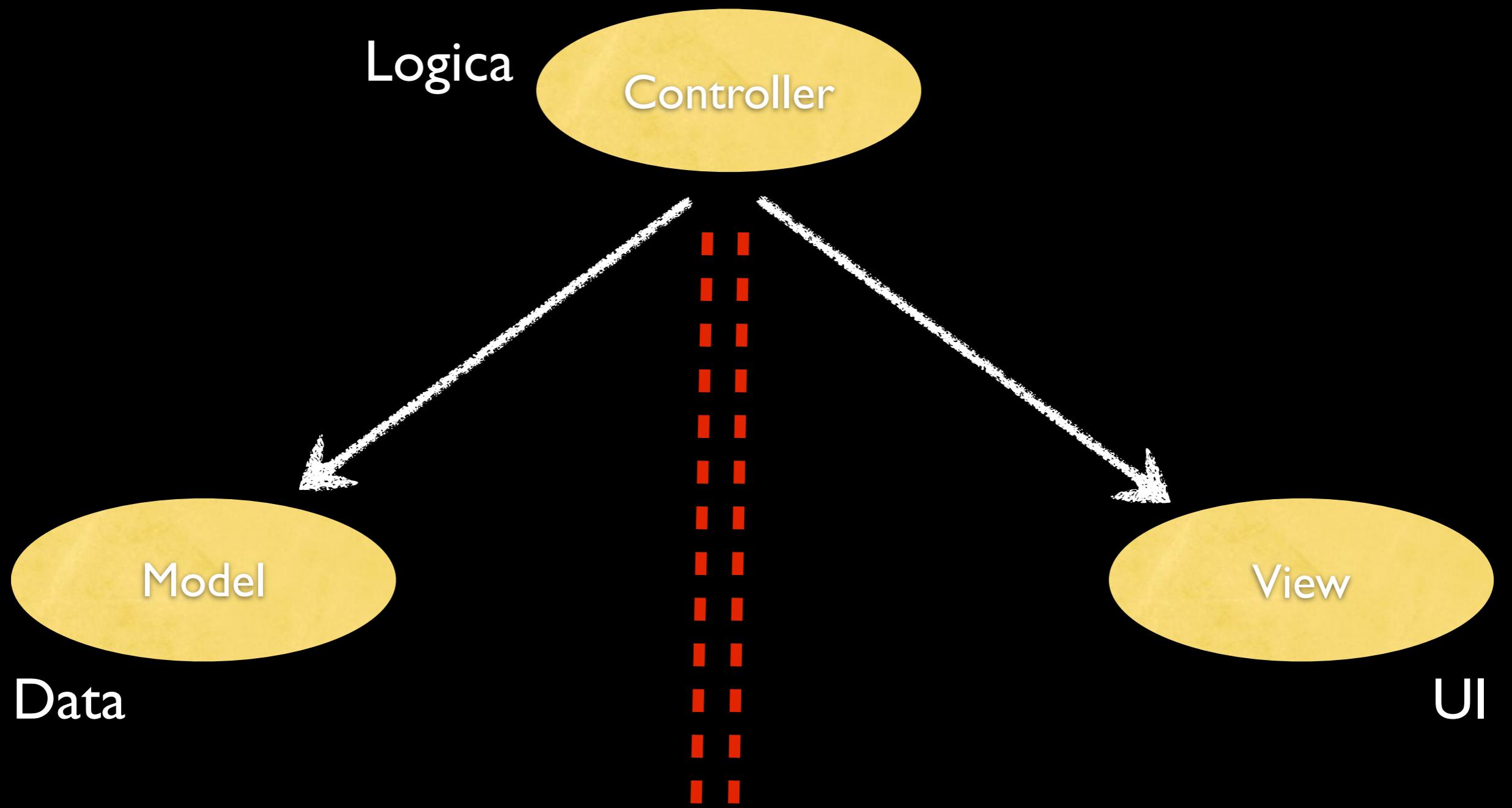
# iOS - MVC



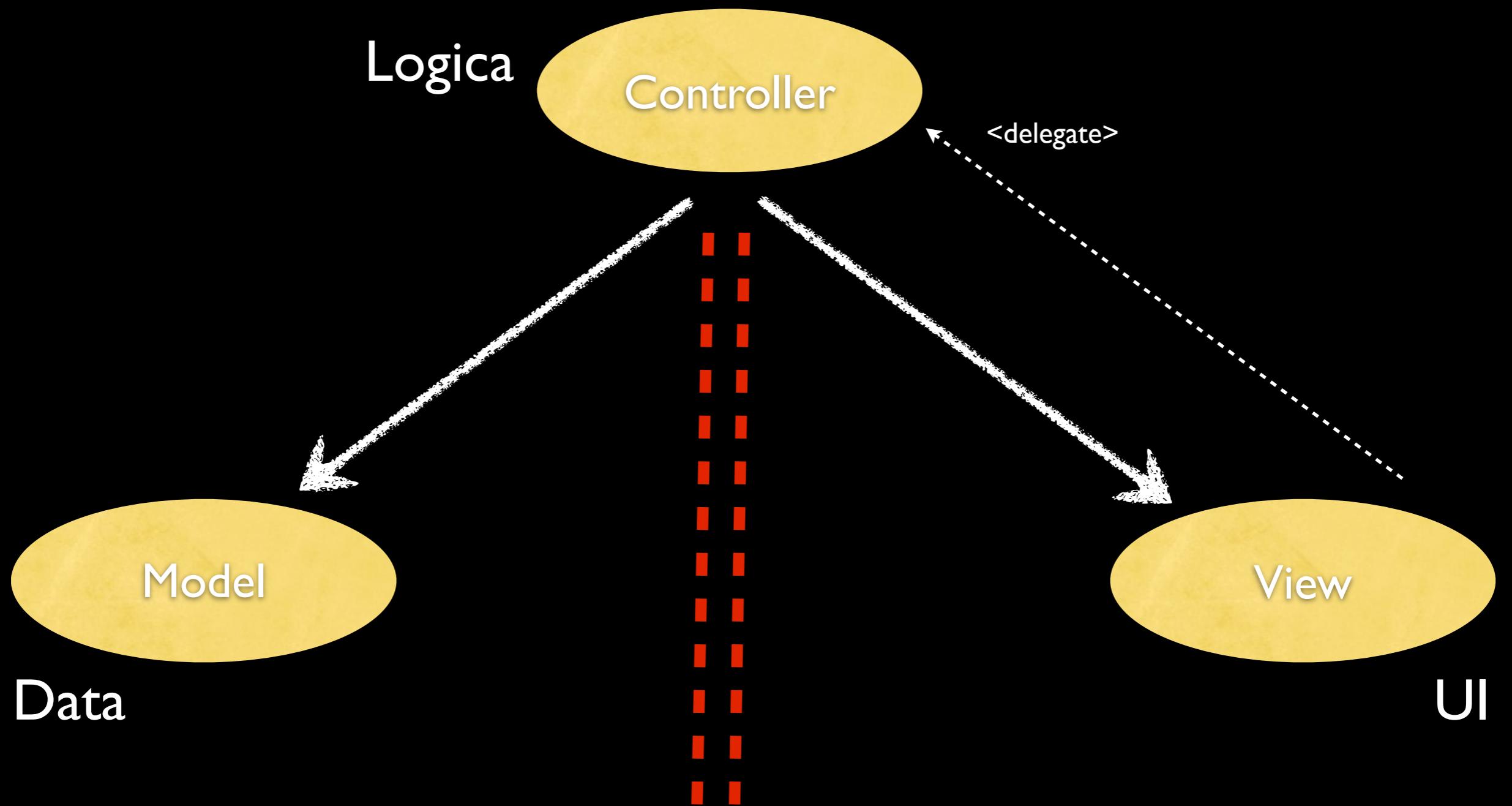
# iOS - MVC



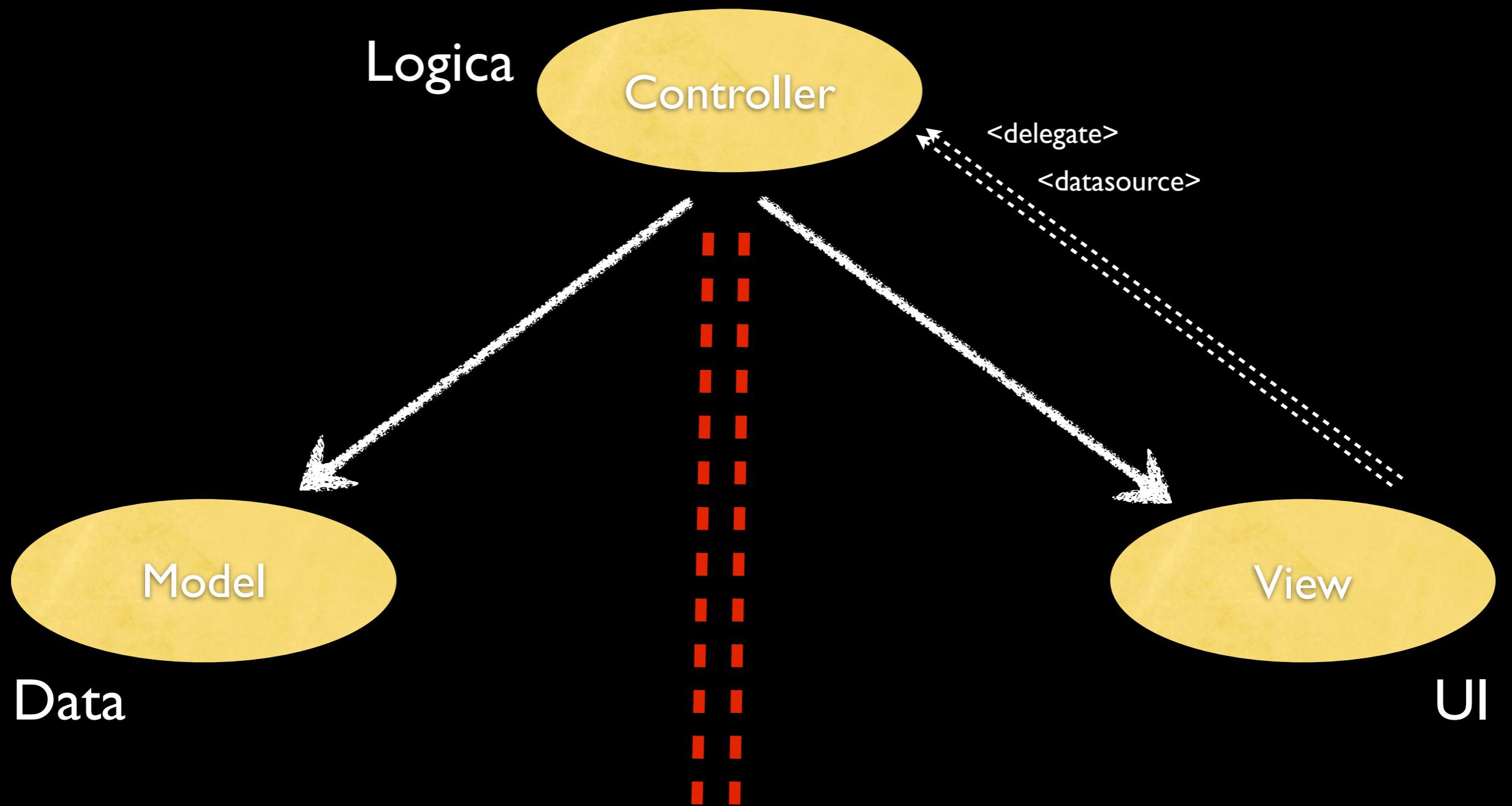
# iOS - MVC



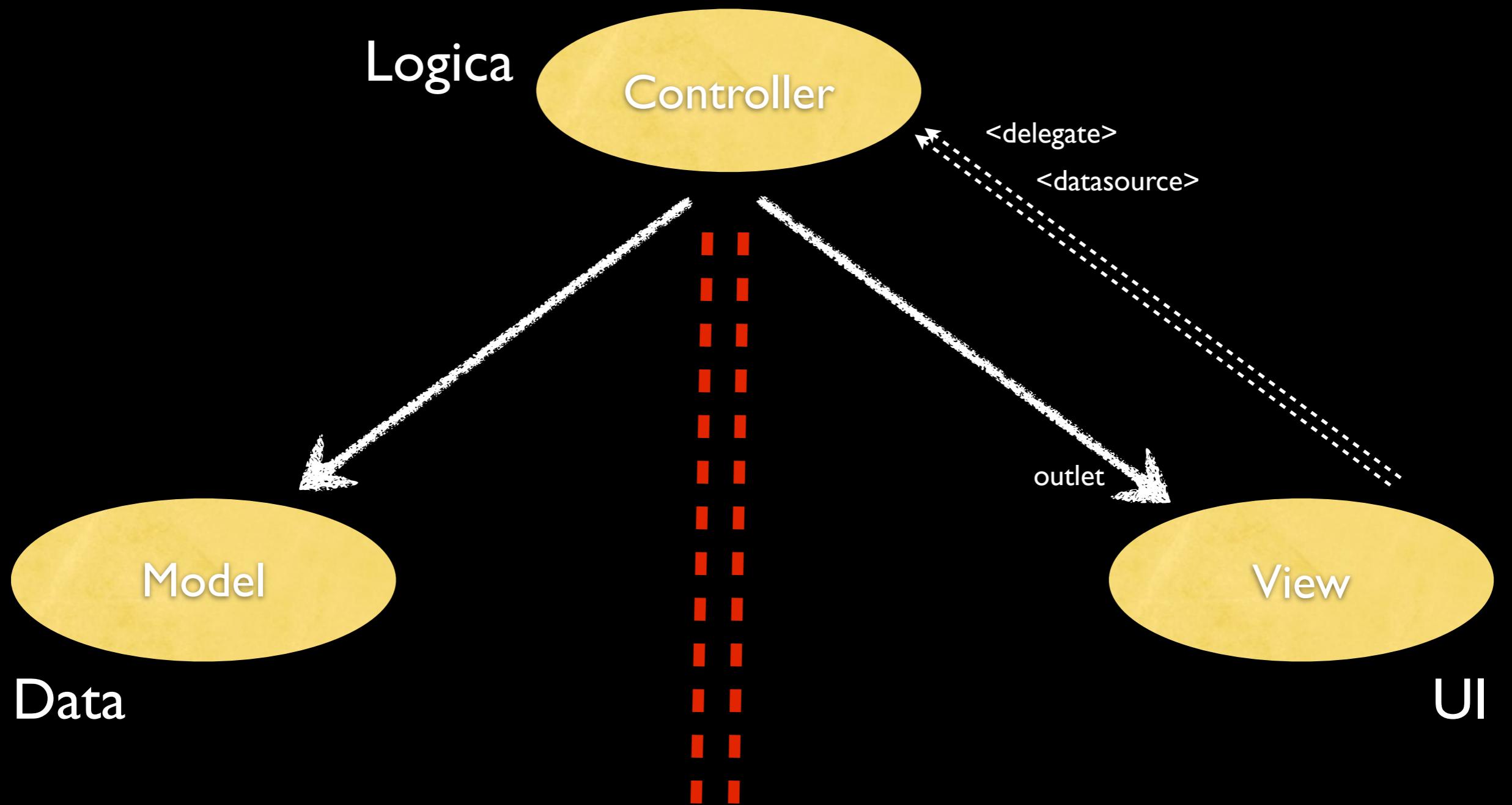
# iOS - MVC



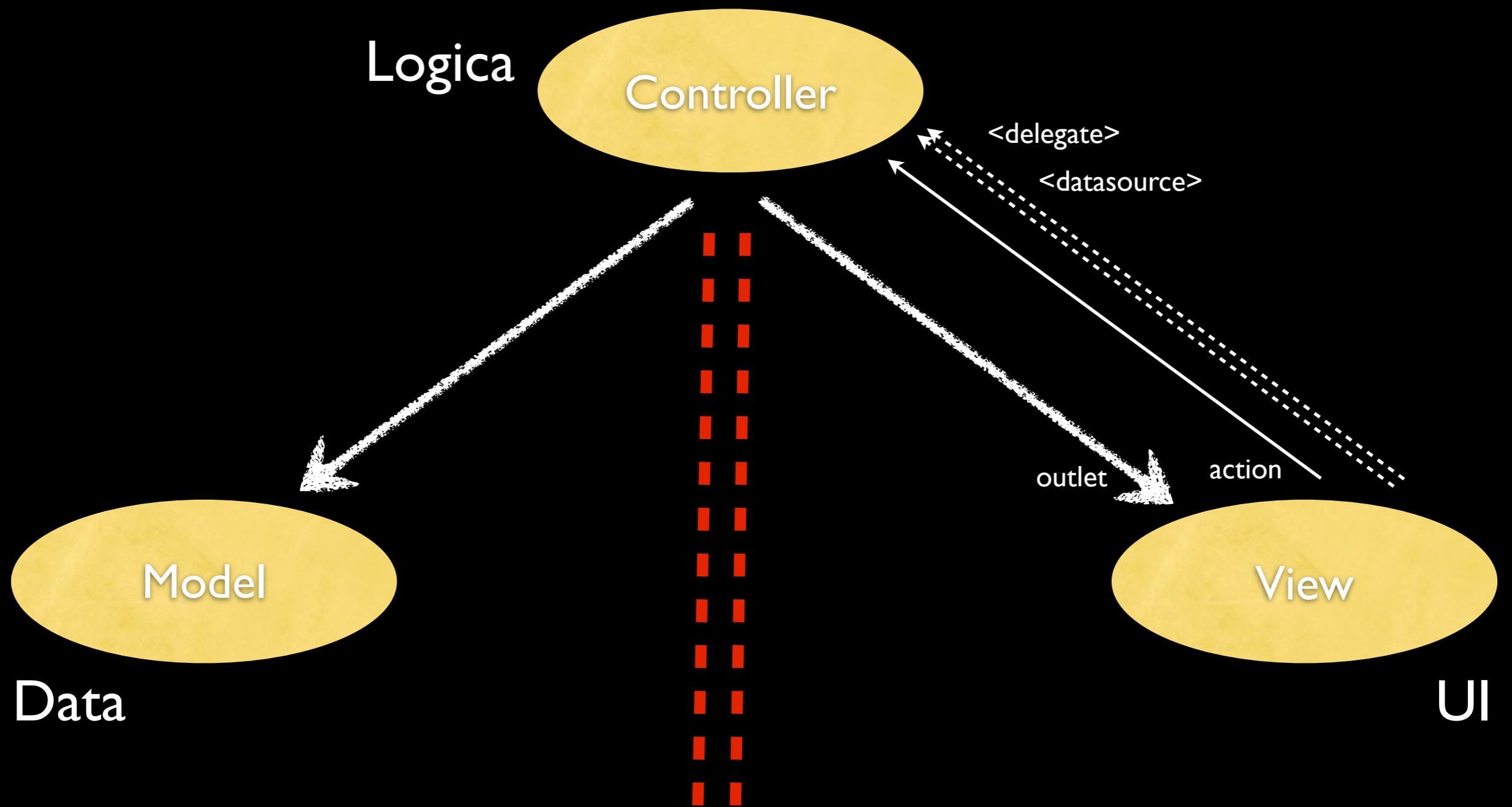
# iOS - MVC



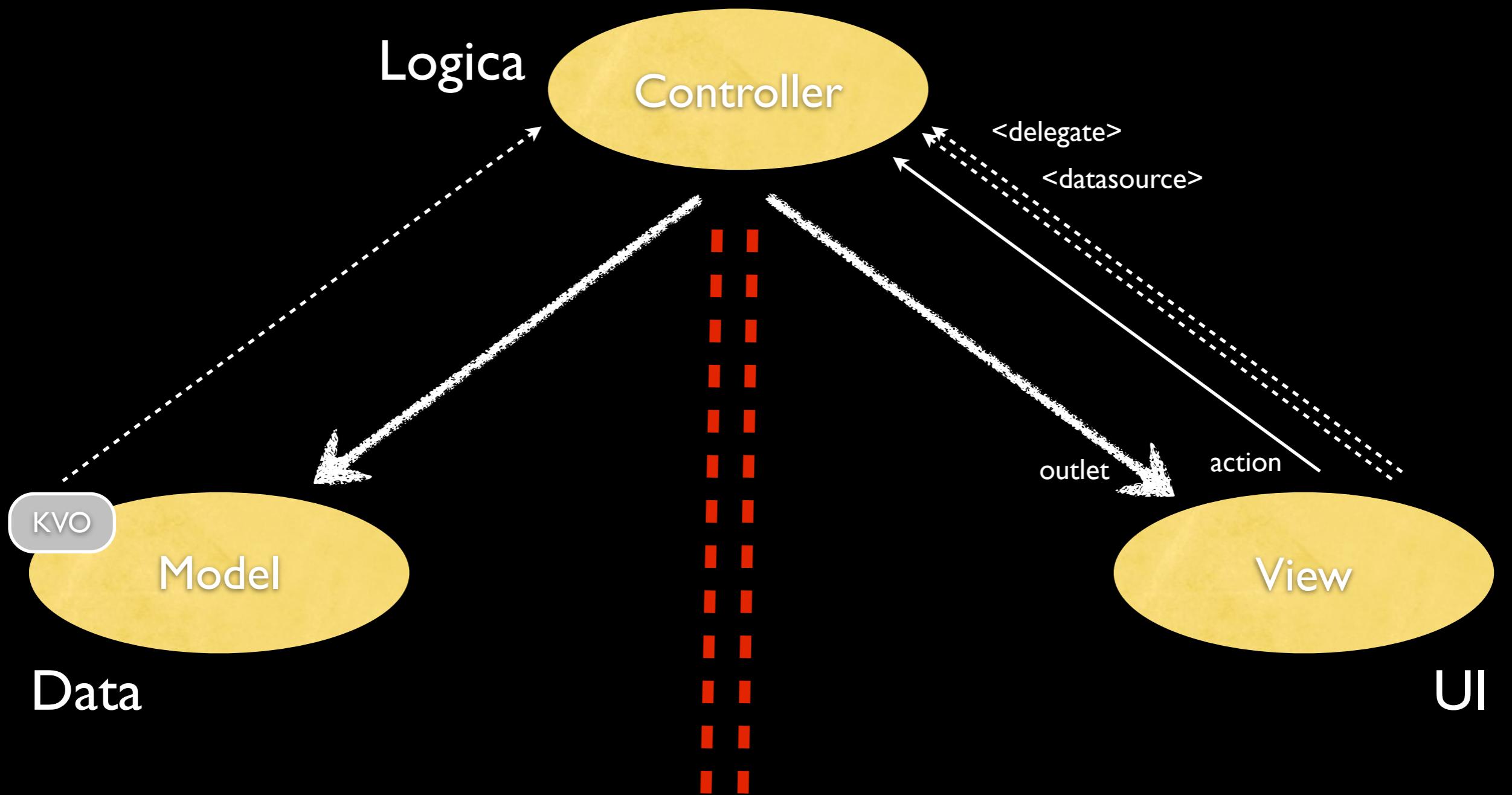
# iOS - MVC



# iOS - MVC



# iOS - MVC



# Pasos

- Crear Proyecto
- Crear DetailView & Controller
- Crear NCEvent objeto
- Add NavigationController
- Crear TableView
- Push/Pop DetailView



# XCode: Crear Proyecto

- IDE for Objective C
- Templates
  - Window-based Application
- com.nscoderbcn.HelloNSCoder
- Include Unit Tests



# DetailsViewController

- Creación de la View de detalles(XIB) y su Controller
- File -> New -> New File
  - UIViewController subclass
  - With XIB
  - Save As: NCDetailsViewController



# XCode Groups

- Creación de grupos para organizar el proyecto
- File -> New -> New Group
  - UI
  - XIB
  - Model



# IB: DetailView



- Herramienta para crear GUI
- XIB: XML file que define nuestra view
- NIB: compilado del XIB
- IBOOutlet / IBAction
  - conectar el Controller con los elementos de la View



# Details View

- NCEventDetailViewController.xib
- añadir y posicionar en UIView
  - UIImageView
    - Image: NSCoder\_Icon.png
  - 3 UILabel
  - UITextView



# Root View Controller

- HelloNSCoderAppDelegate

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:  
NSDictionary *)launchOptions  
{  
  
    NCEventDetailViewController *detailViewController = [[NCEventDetailViewController  
alloc] init];  
  
    self.window.rootViewController = detailViewController;  
    [detailViewController release];  
  
    [self.window makeKeyAndVisible];  
    return YES;  
}
```

.m



# NCEvent

- Creación de la clase con los detalles del evento = Model
- File -> New -> New File
  - Objective C class
  - Subclass of NSObject
  - Save As: NCEvent



# NCEvent

- header
  - properties & init
- implementation
  - synthesize, init, dealloc

```
@interface NCEvent : NSObject {  
}  
  
@property (nonatomic, copy) NSString *name;  
@property (nonatomic, copy) NSString *shortDescription;  
@property (nonatomic, retain) NSDate *startDate;  
@property (nonatomic, copy) NSString *location;  
@property (nonatomic, copy) NSString *fullDescription;  
  
- (id)initWithName:(NSString *)eName startDate:(NSDate *)eStartDate location:(NSString *)eLocation;  
  
@end
```

.h



# Detail View-Controller

- Conectar NCEventDetailViewController a XIB
  - property event
  - property outlets para los elementos del XIB
  - synthesize & dealloc!!!
- Conectar a elementos en XIB

```
@property (nonatomic, retain) NCEvent *event;  
  
@property (nonatomic, retain) IBOutlet UILabel *shortDescriptionLabel;  
@property (nonatomic, retain) IBOutlet UILabel *startDateLabel;  
@property (nonatomic, retain) IBOutlet UILabel *locationLabel;  
@property (nonatomic, retain) IBOutlet UITextView *descriptionView;
```

.h



# Load View Data

- NCEventDetailViewController
- cargar datos el evento al aparecer

```
- (void)viewWillAppear:(BOOL)animated
{
    [self.navigationItem setTitle:event.name];
    self.shortDescriptionLabel.text = event.shortDescription;
    NSDateFormatter *formatter = [[NSDateFormatter alloc] init];
    [formatter setDateFormat:@"yyyy-mm-dd"];
    self.startDateLabel.text = [formatter stringFromDate:event.startDate];
    self.locationLabel.text = event.location;
    self.fullDescriptionTextView.text = event.fullDescription;
}
```



# Crear Evento

- HelloNSCoderAppDelegate
- crear un evento y asignalo al Controller

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    NCEvent *event = [[NCEvent alloc] initWithNibName:@"Hello NSCoder" startDate:[NSDate date] location:@"CINC Barcelona"];
    event.shortDescription = ...
    NCEventDetailViewController *detailViewController = [[NCEventDetailViewController alloc] init];
    detailViewController.event = event;
    [event release];
    ...
}
```

.m



# Navigation Controller

- HelloNSCoderAppDelegate
  - crear Navigation Controller
  - set Detail Controller como root

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:  
NSDictionary *)launchOptions  
{  
    ...  
  
    NCEventDetailViewController *detailViewController = [[NCEventDetailViewController  
alloc] init];  
  
    detailViewController.event = event;  
    [event release];  
  
    UINavigationController *navController = [[UINavigationController alloc]  
initWithRootViewController: detailViewController];  
  
    ...  
}
```

.m



# Navigation Title

- NCEventDetailsViewController
  - asignar un titulo a la Navigation Bar

```
- (void)viewWillAppear:(BOOL)animated
{
    self.navigationItem.title = event.name;
    ...
}
```

.m



# Gracias!

- @nscoder\_bcn
- Alfonso: @setfilesl
- Andreas: @aquarioverde

