

Application Flow and Model View Controller

Introduction to iPhone Development
IAP 2009 ❄

Yesterday

Objective-C Crash Course.

we saw:

Classes

Interfaces

Properties

Memory Management

Remaining Question

init methods

Today

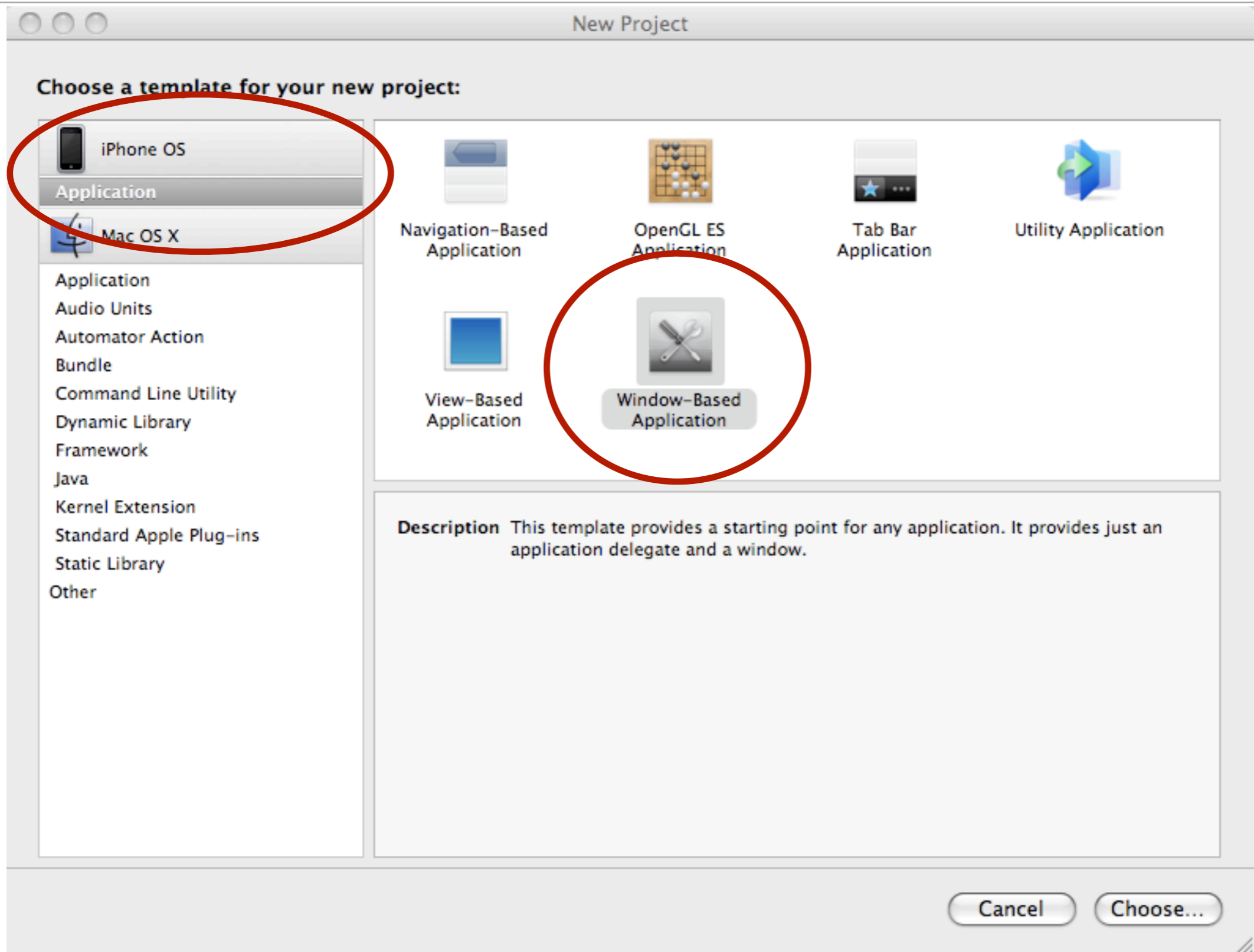
Application “Flow”

MVC

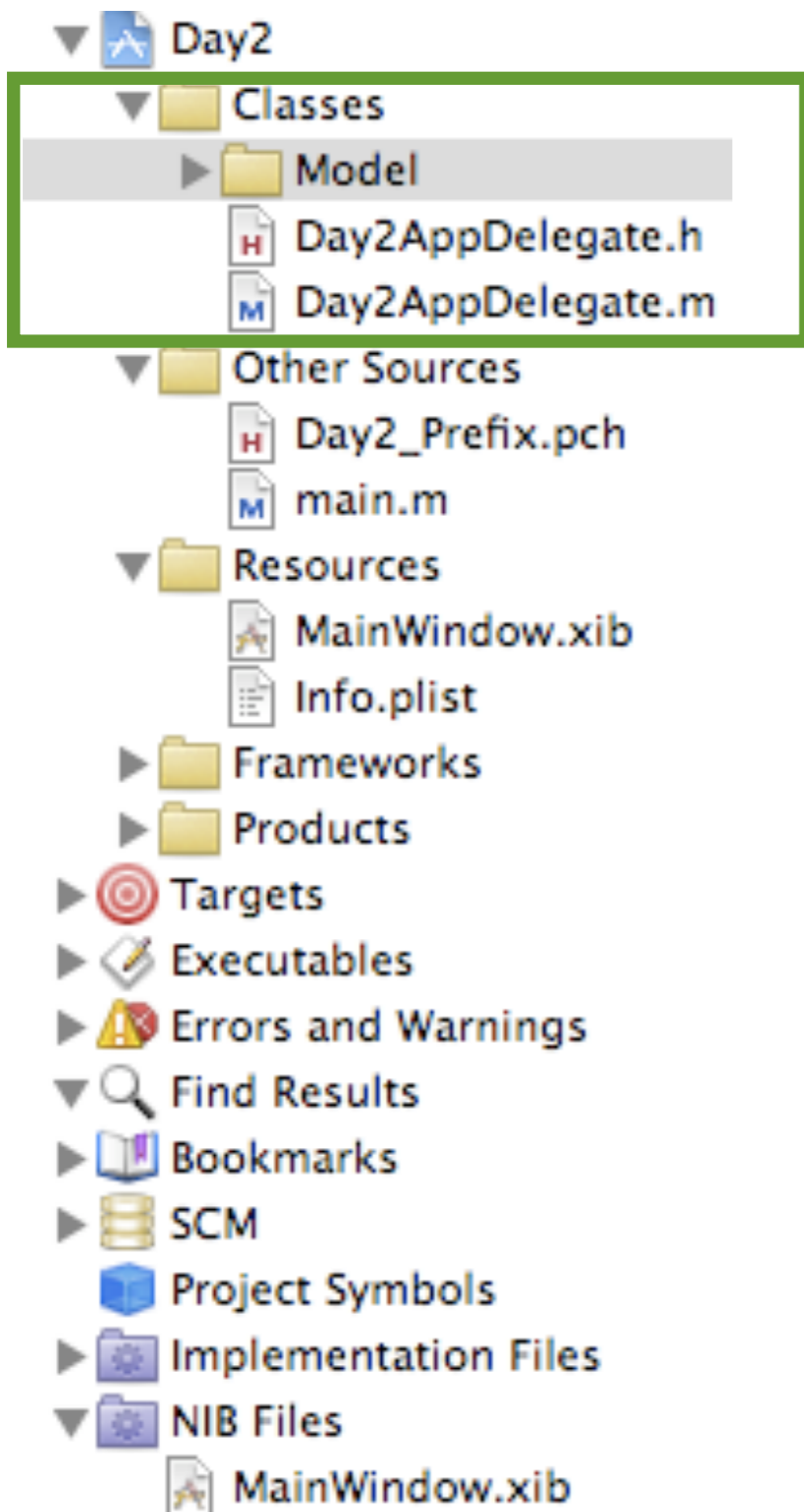
Interface Containers

Changed format: ditch the “lab” half --
we’ll code together during the lecture.

So...



iPhone Application Anatomy



Your Code

main() routine

Resources

NIB Files

Code Breakdown

Application Delegate

Everything Else

Resources

NIB Files (Interface Files)

+ *info.plist*

Application Flow

UIApplication



Main NIB Initialized

AppDelegate

```
- (void)applicationDidFinishLaunching:(UIApplication *)application  
{
```

Initialize Your Root Controller & Interface

```
[window makeKeyAndVisible];  
}
```

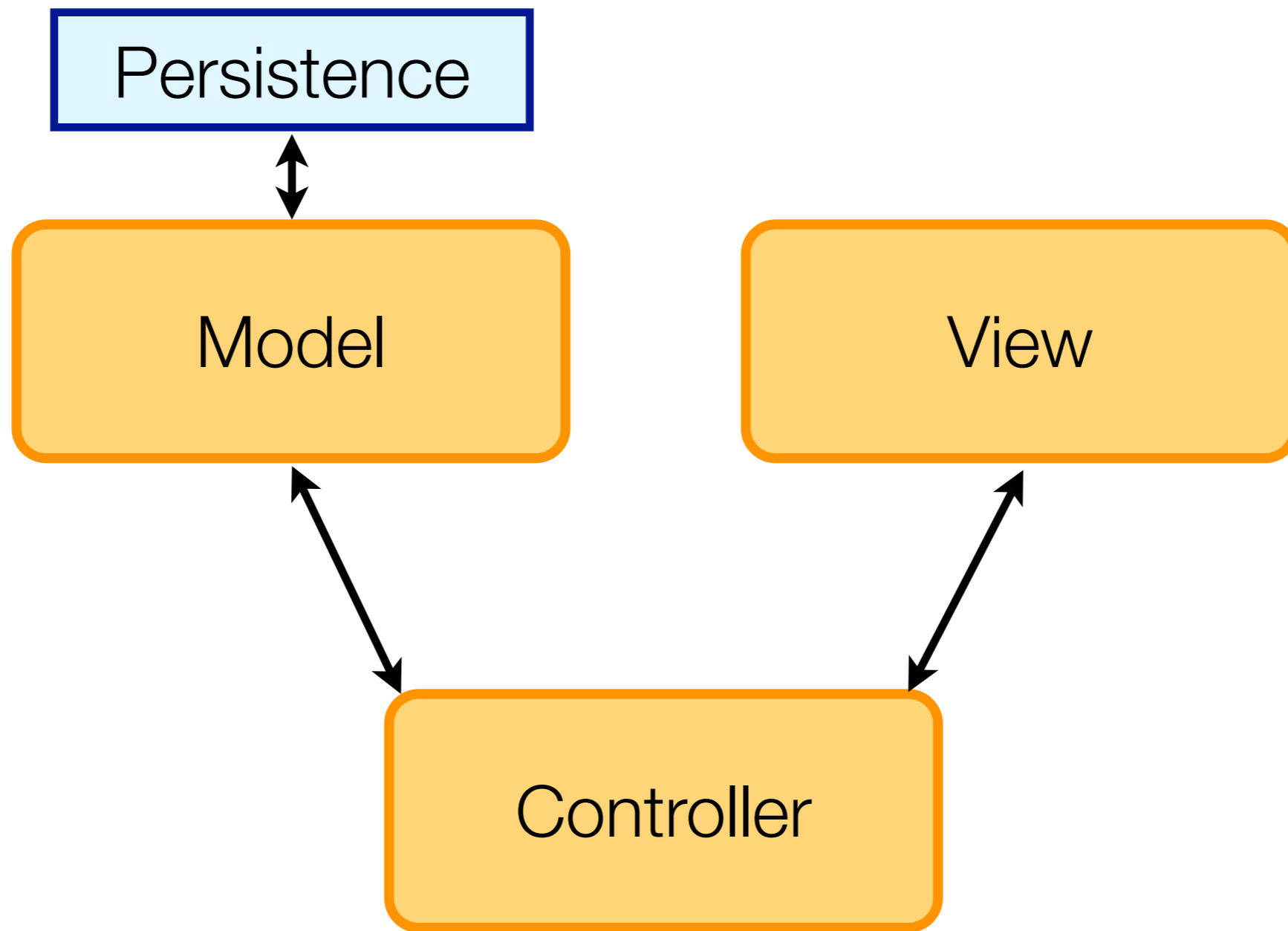
After which point it is..

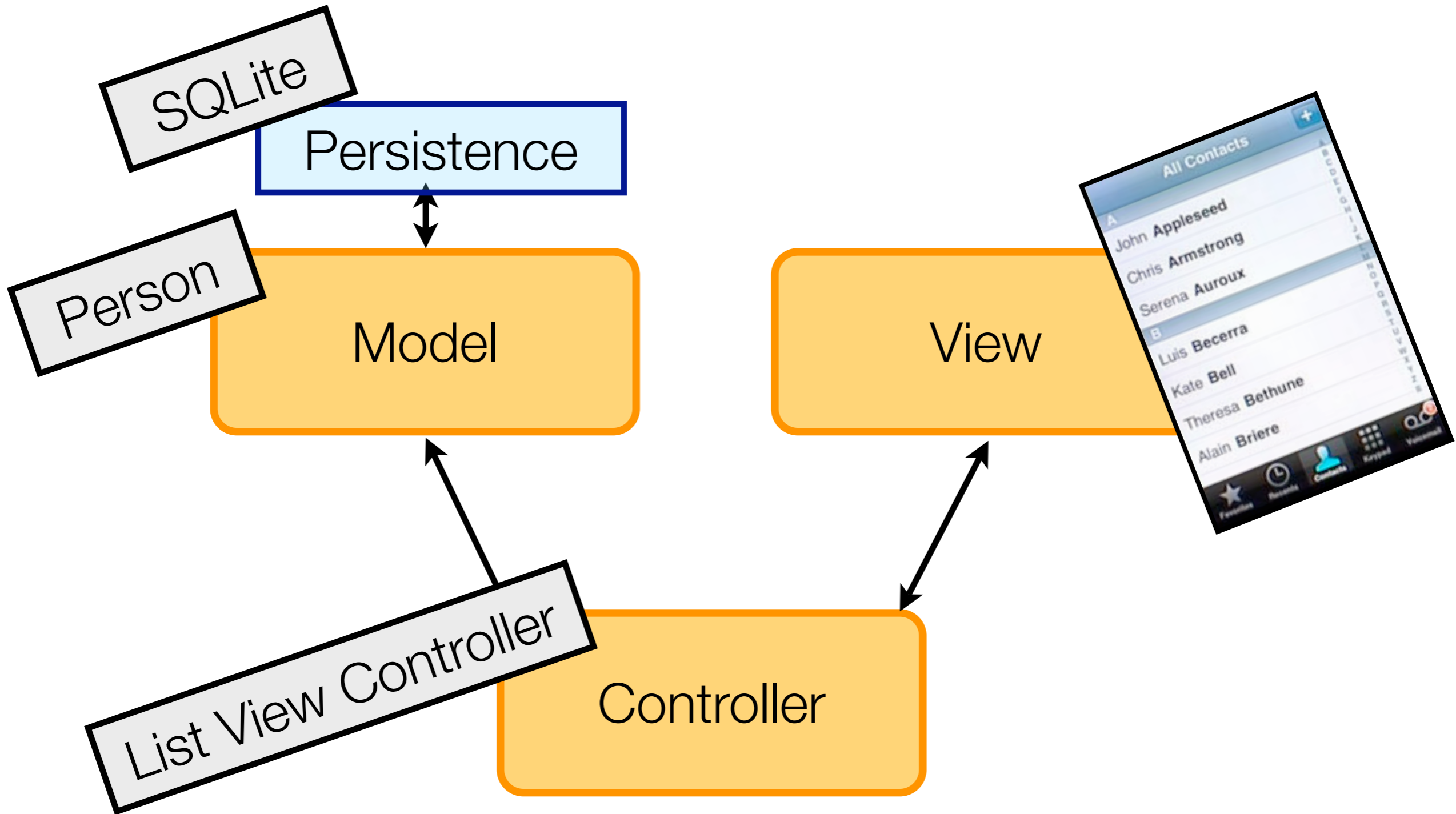




So Application Design and
UI Design are intimately paired.

Information-Driven Applications





A Simple Controller and View

UIViewController

UIView

notes:

why we're not starting out with IB

placeholder view controller

loadView -- called automagically
set breakpoint to show

Information-Driven Applications

Information Driven + UI Driven

Controller + View

ViewController

*And then the models just
help you encapsulate data*

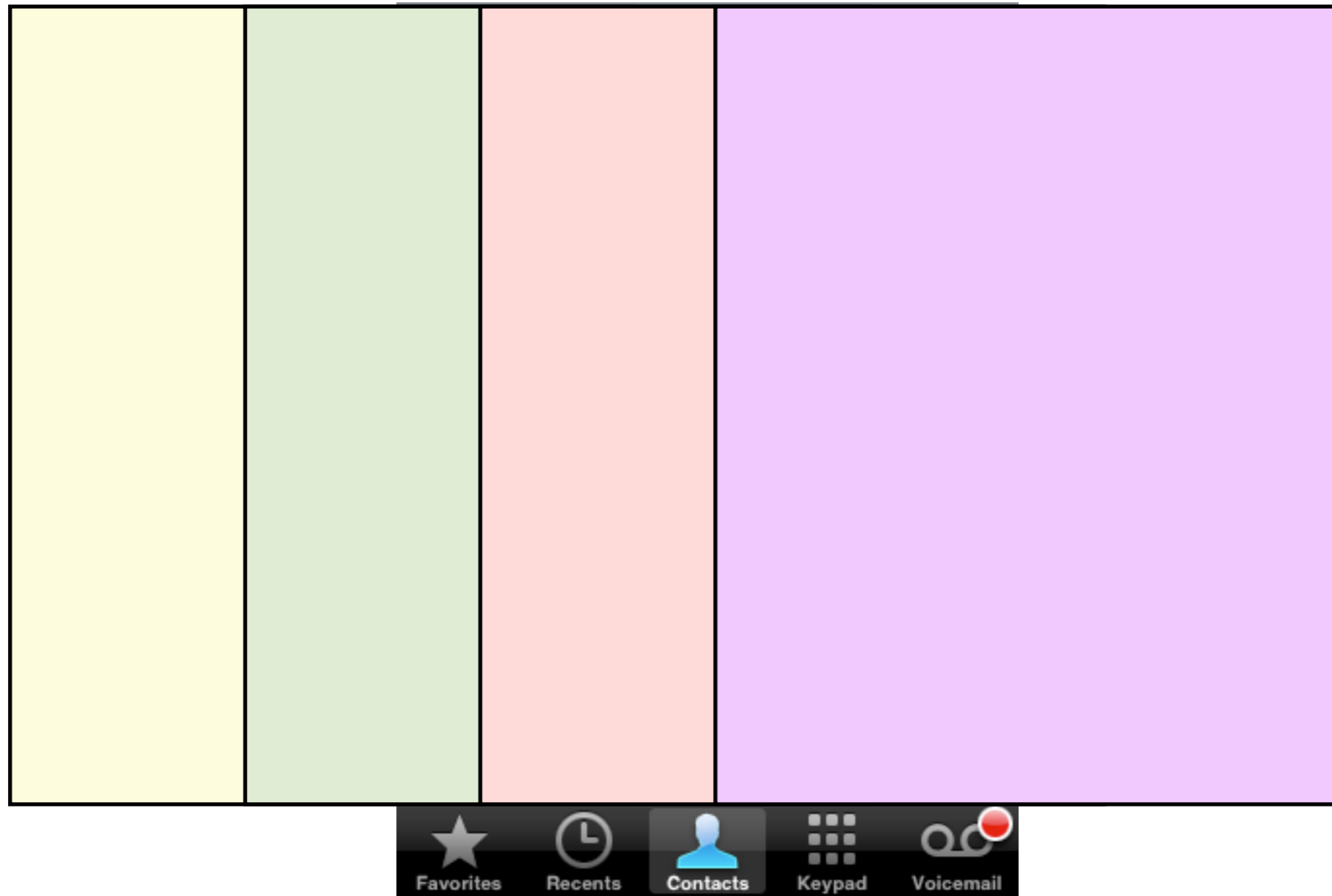
UI Containers

Containers for Navigating through Views

Containers for Navigating through Data

<http://www.youtube.com/watch?v=0QYoESMe5Zo>

Multiplex Between Views



UITabBarController

Push Views on a Stack



Lists of Items

Lists of Views

UITabBarController

remember:
NSArray -- why the nil?

Setting the title

title -> icon, other things

Stacks of Views

UINavigationController

remember:
What would have happened
if a reference to view was in loadView?

Lists of Views

UITableViewController

remember:

Use bare-bones; where did the title go??
Show what happens when release text in cell.

The Shell of Our Application



Tomorrow:

Interface Builder

Detail Views

Creating/Adding/Editing