Model-View-Controller for iPhone OS

Essential Design Pattern for Flexible Software

Ken Kocienda
Principal Engineer, iPhone Software

Design Patterns

Facade

Strategy

State

Mediator

Delegate

Command

Memento

Composite

Observer

Factory Method

Object-Oriented

Visitor

Prototype

Adapter

Flyweight

Model-View-Controller (MVC)

Iterator

Interpreter

Object/Relational Mapping

Decorator

Builder

Facade

Strategy

State

Mediator

Delegate

Command

Memento

Composite

Observer

Factory Method

Object-Oriented

Design Patterns

Visitor

Prototype

Adapter

Flyweight

Model-View-Controller (MVC)

Iterator

Interpreter

Decorator

Object/Relational Mapping

Builder

Model-View-Controller (MVC)

Why Should You Care?

Great Apps

Development for iPhone

Small screen = simpler app organization

Development for iPad

Bigger screen = more complex app organization

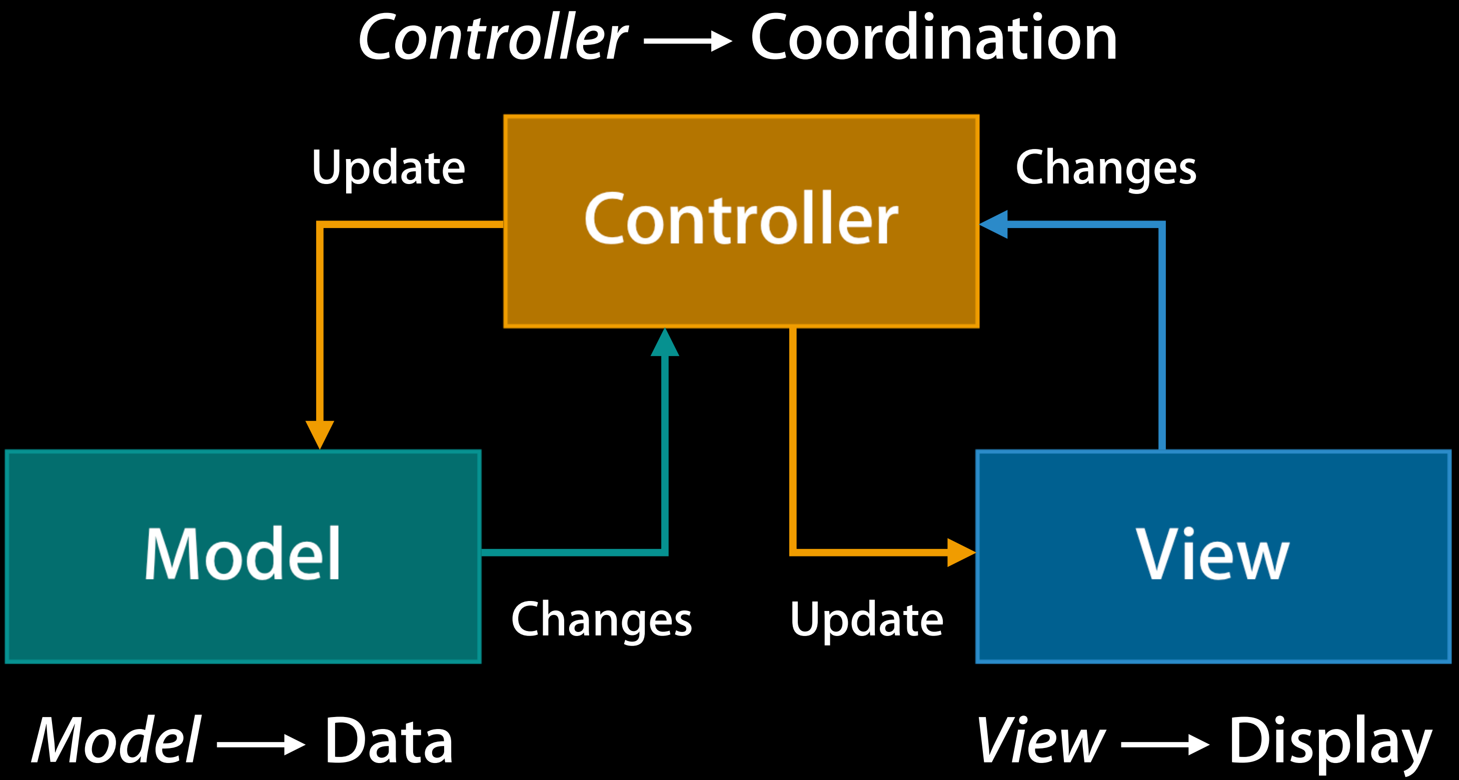
Development for Both

Serving two masters at the same time

MVC Can Help

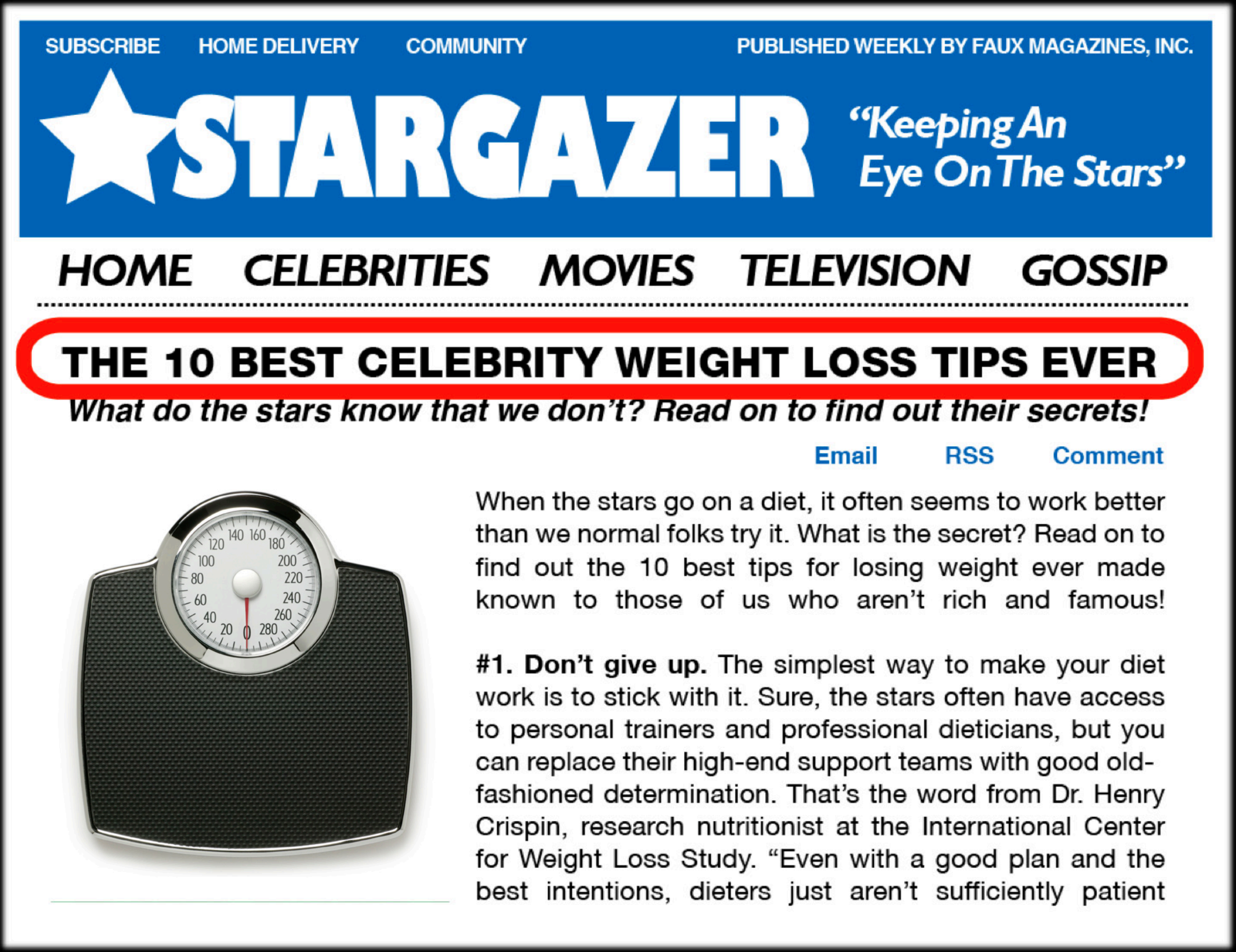
Model-View-Controller

There will be a test on Friday



The “Real World”

The "Real World"



SUBSCRIBE HOME DELIVERY COMMUNITY PUBLISHED WEEKLY BY FAUX MAGAZINES, INC.

STARGAZER "Keeping An Eye On The Stars"

HOME CELEBRITIES MOVIES TELEVISION GOSSIP

THE 10 BEST CELEBRITY WEIGHT LOSS TIPS EVER

What do the stars know that we don't? Read on to find out their secrets!

Email RSS Comment

When the stars go on a diet, it often seems to work better than we normal folks try it. What is the secret? Read on to find out the 10 best tips for losing weight ever made known to those of us who aren't rich and famous!

#1. Don't give up. The simplest way to make your diet work is to stick with it. Sure, the stars often have access to personal trainers and professional dieticians, but you can replace their high-end support teams with good old-fashioned determination. That's the word from Dr. Henry Crispin, research nutritionist at the International Center for Weight Loss Study. "Even with a good plan and the best intentions, dieters just aren't sufficiently patient

The 10 Best MVC Tips Ever

Flexible and Easy to Change

Great Apps

#1. Learn MVC for iPhone OS

#1. Learn MVC for iPhone OS

Common conventions

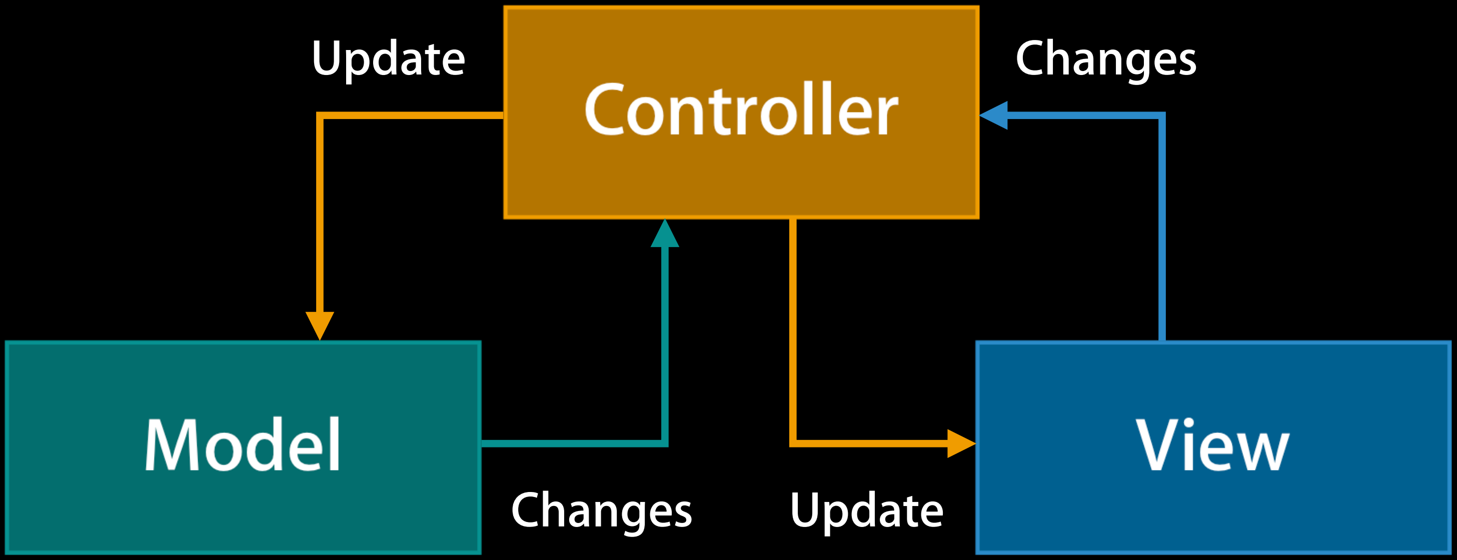
Built up from other design patterns

Connections Between Objects

Model-View-Controller

Connections between objects

Controller ⟶ Coordination

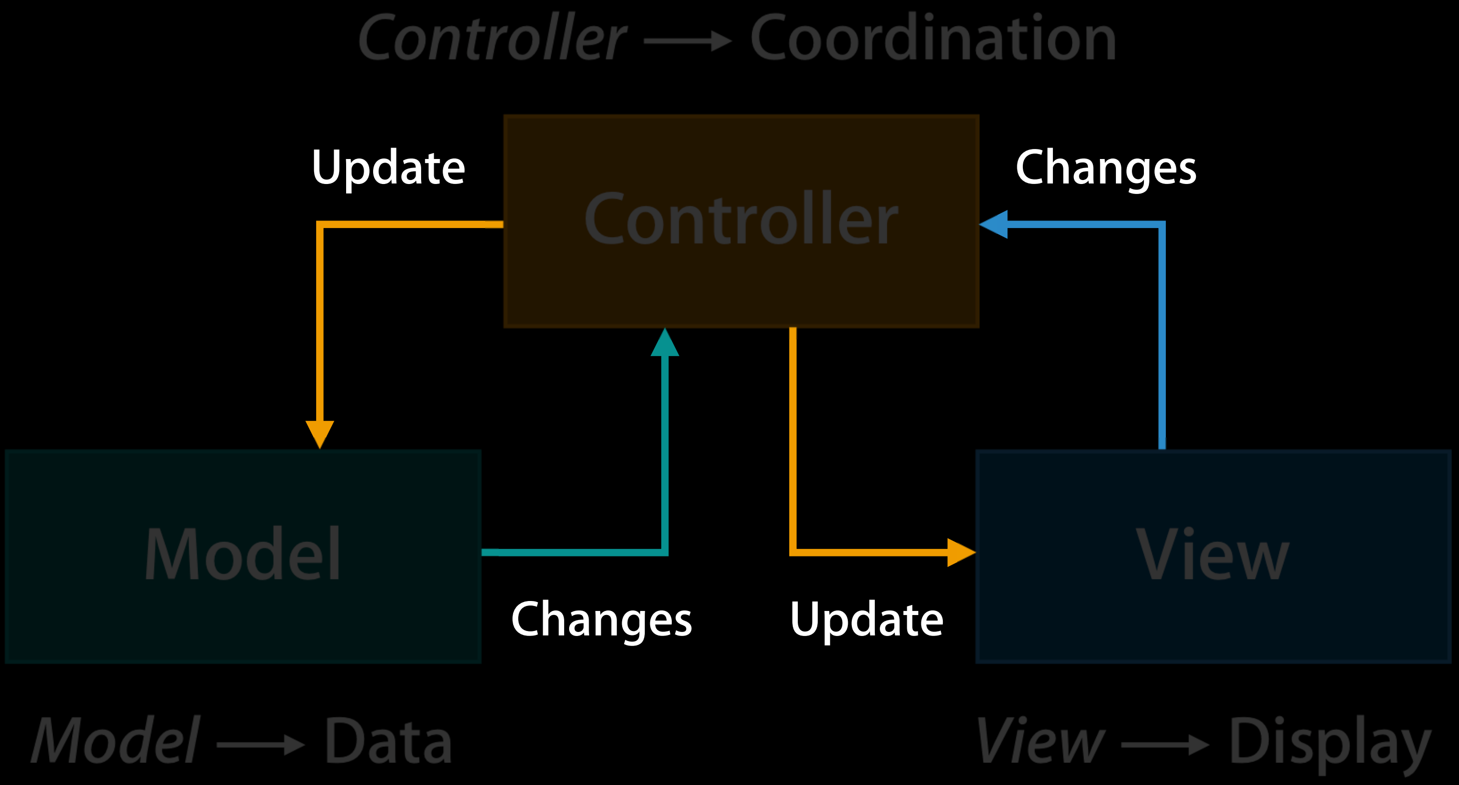


Model ⟶ Data

View ⟶ Display

Model-View-Controller

Connections between objects



Model-View-Controller

Built on lower-level design patterns

Target-Action

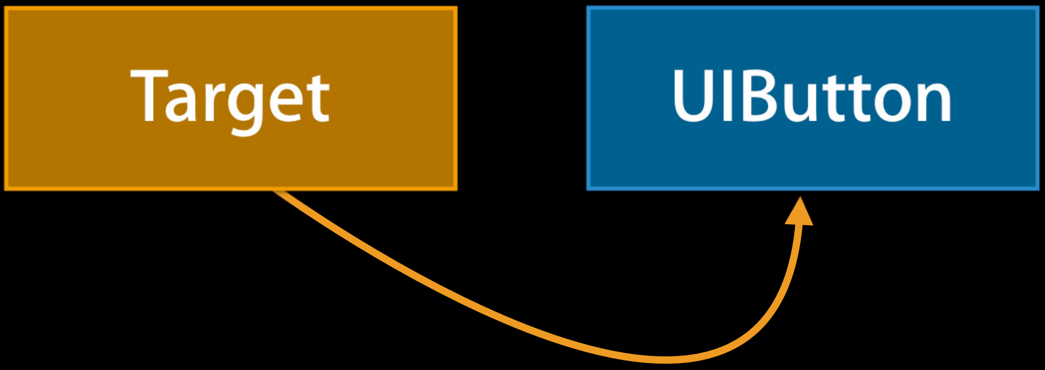
Notification

Delegation

Target-Action

Reusing controls without subclassing

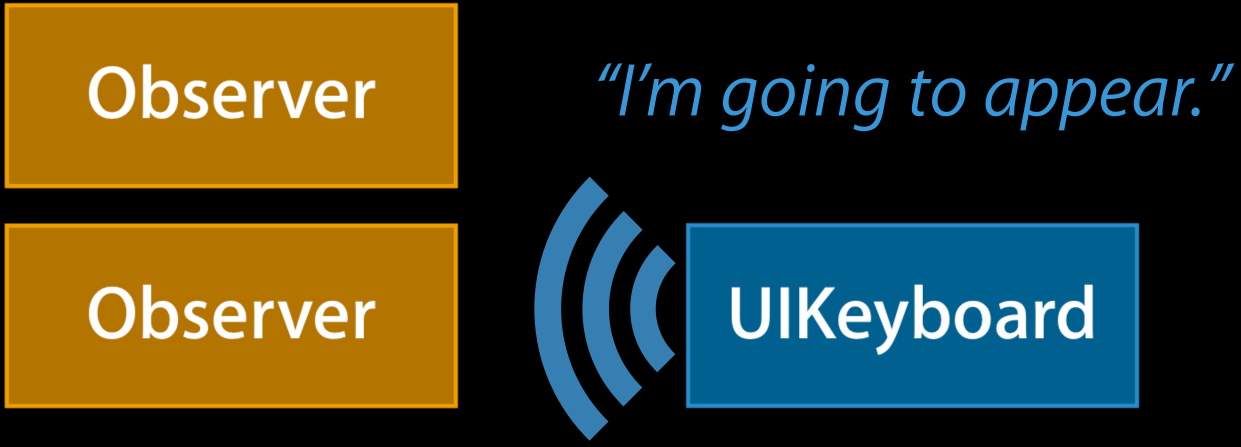
*“When you’re tapped,
call this method on me”*



```
-setTarget:(id)target action:(SEL)action...
```

Notification

Broadcast channels for important news

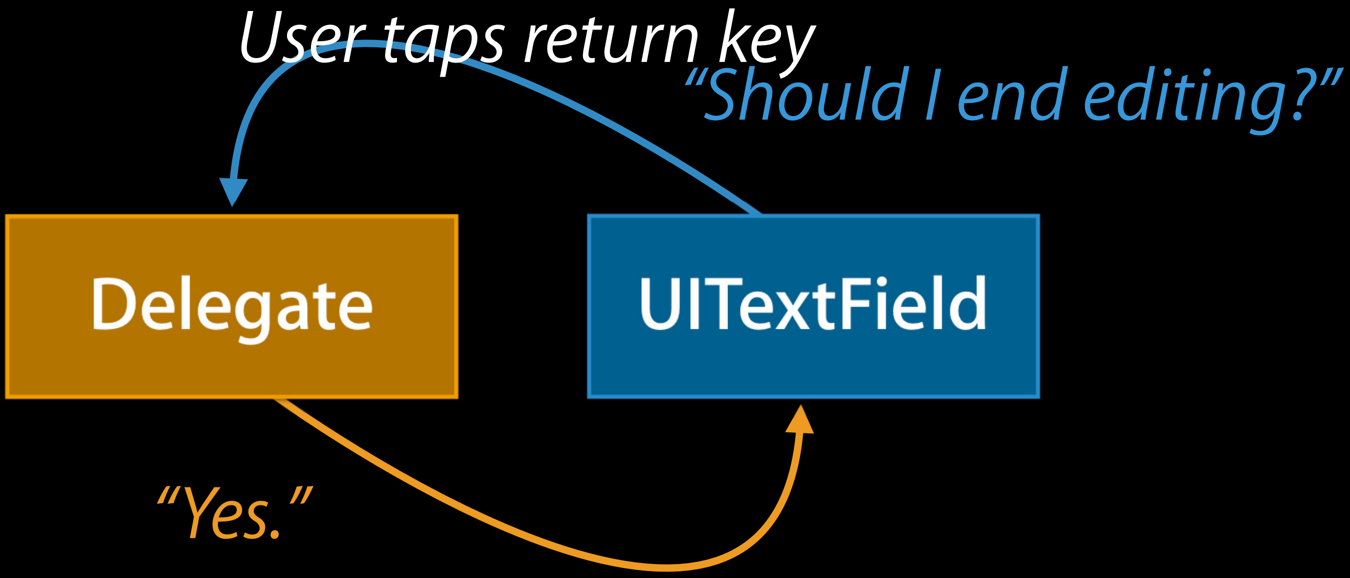


`NSNotificationCenter`

Delegation

Delegation

Reuse without subclassing



`UITextFieldDelegate`

Delegation in UIKit

Many classes have a delegate

UITextField

UIApplication

UIScrollView

UITableView

UIWebView

Delegation in UIKit

Many classes have a delegate

```
will/did/should
```

Delegation in UIKit

UIApplicationDelegate

“I’m about to resign active.”

```
- (void)applicationWillResignActive:
```

Delegation in UIKit

UIScrollViewDelegate

“I zoomed.”

```
- (void)scrollViewDidZoom:
```

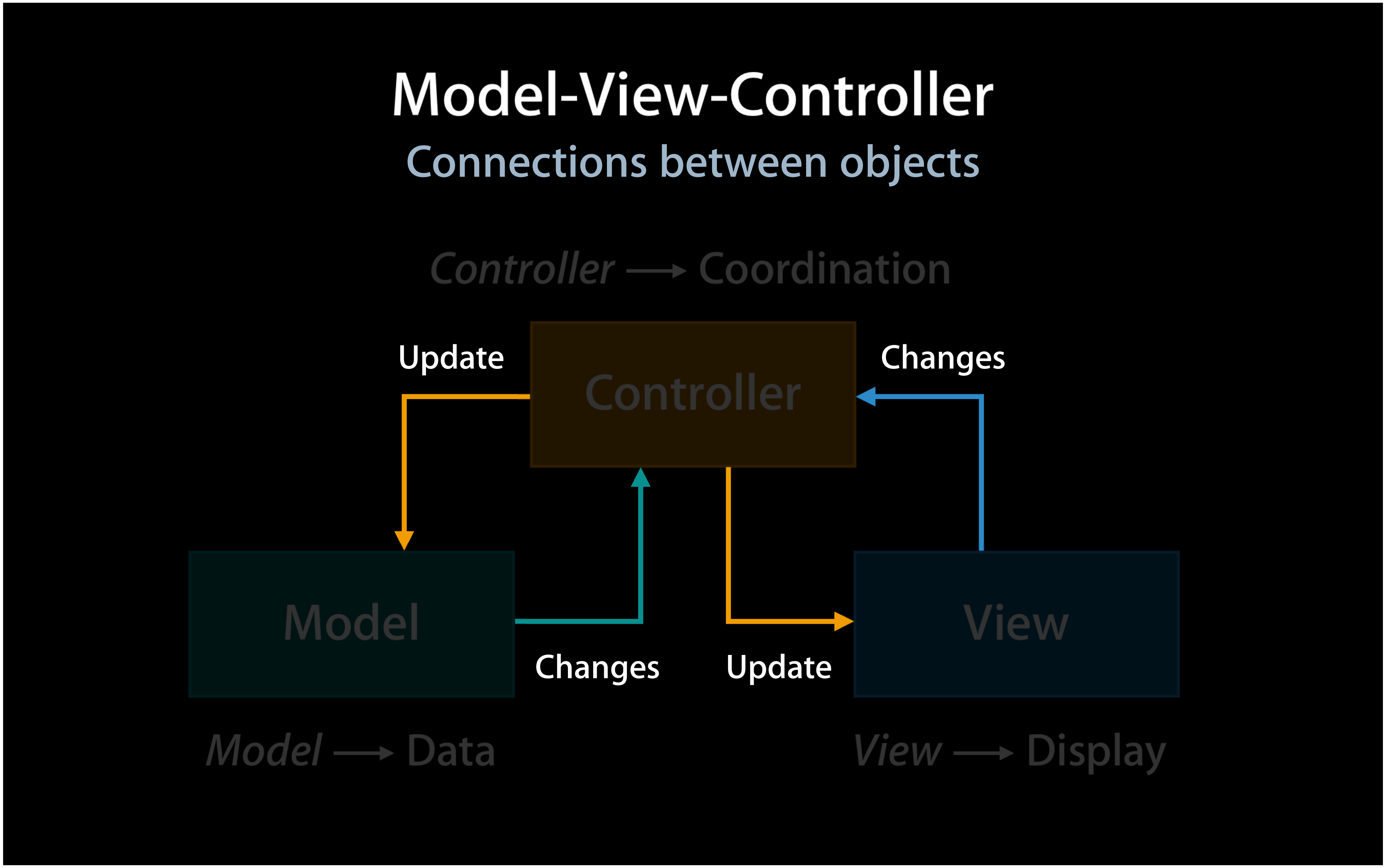
Delegation in UIKit

UITextFieldDelegate

"Should I clear my contents?"

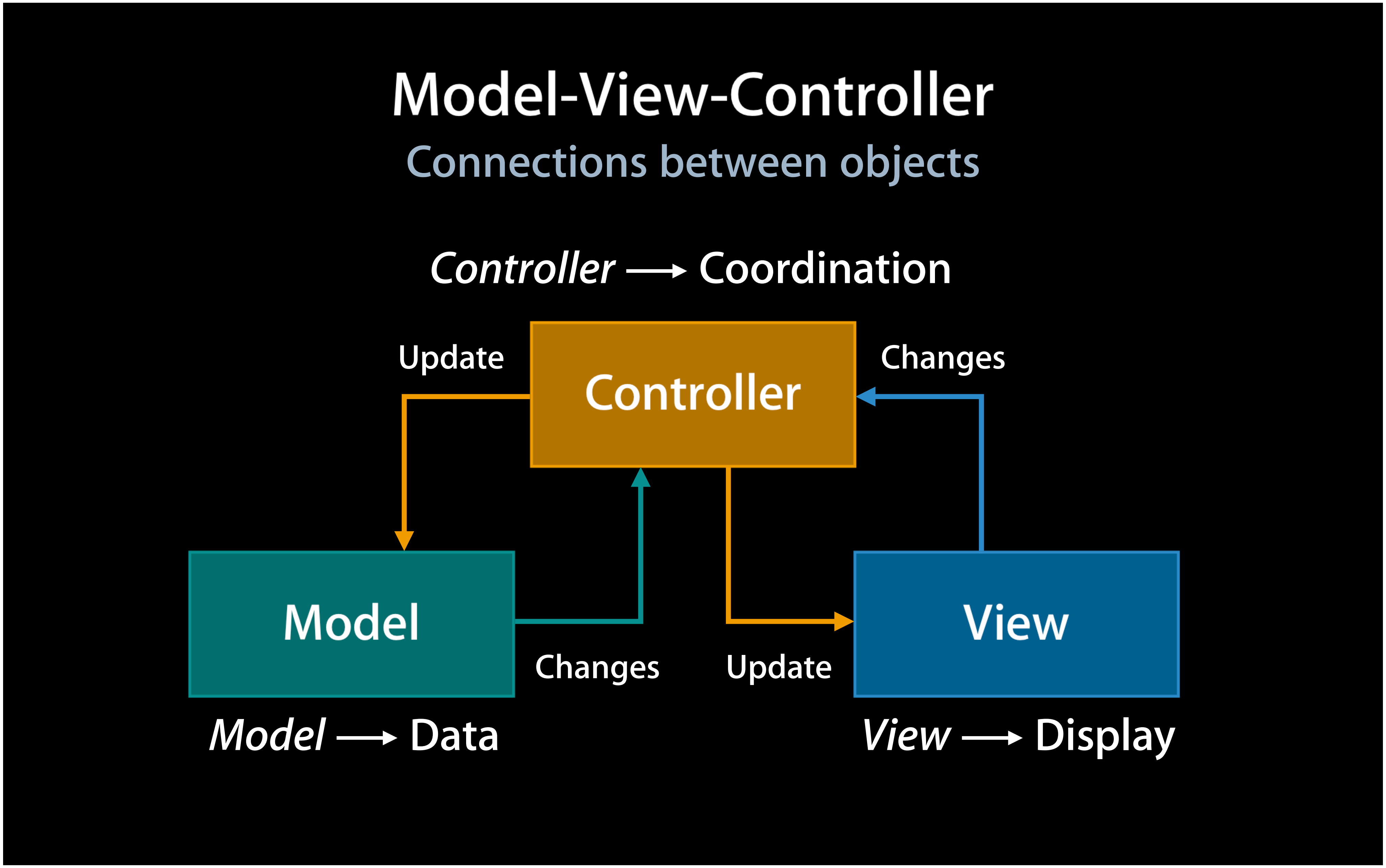
```
- (BOOL)textFieldShouldClear:
```

Flexible and Easy to Change



Model-View-Controller

Connections between objects



Model-View-Controller

Connections between objects

Controller ⟶ Coordination

Model ⟶ Data

View ⟶ Display

#1. Learn MVC for iPhone OS

Common conventions

Built up from other design patterns

#2. Use MVC to Divide Work

#2. Use MVC to Divide Work

Implement a big idea

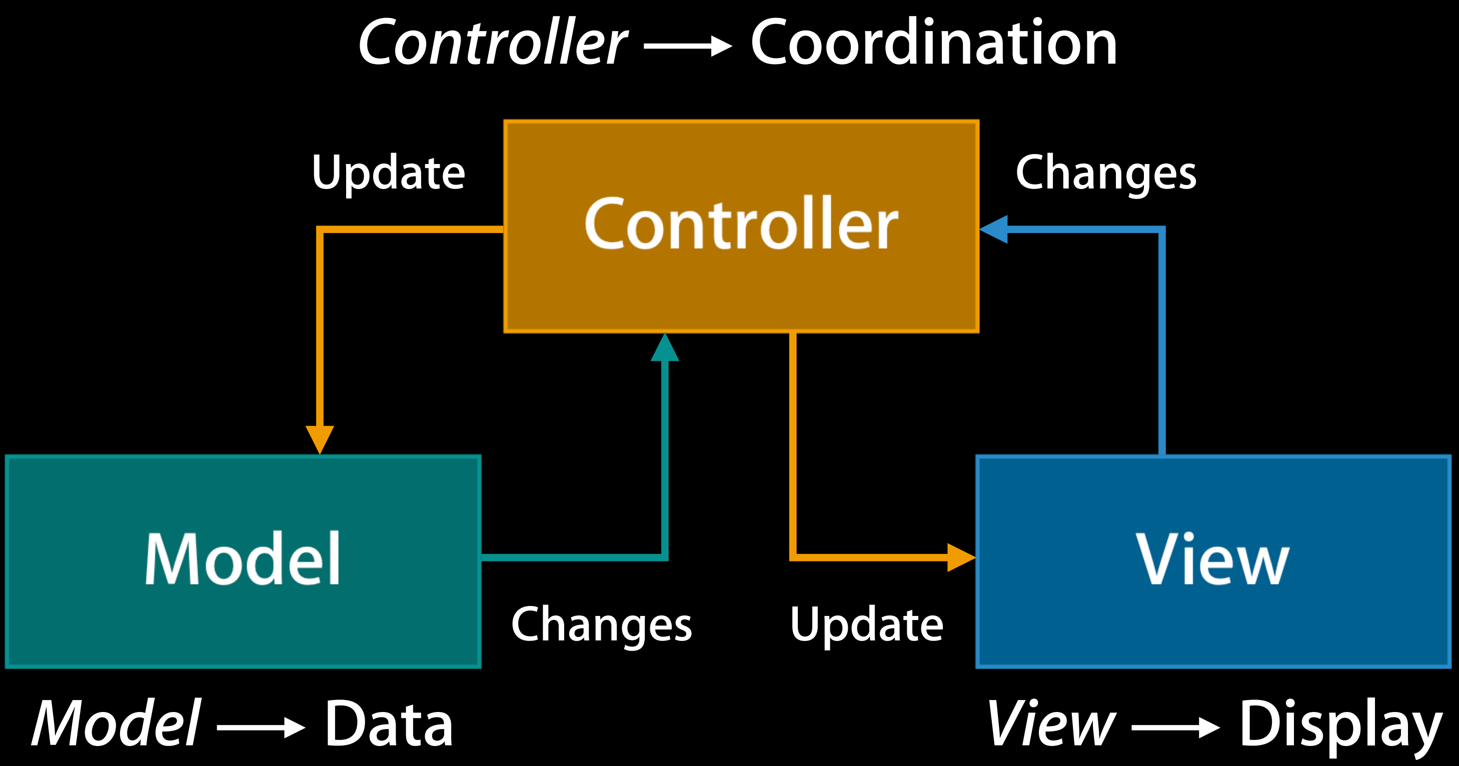
Make manageable pieces

How Does MVC Help?

Useful Buckets

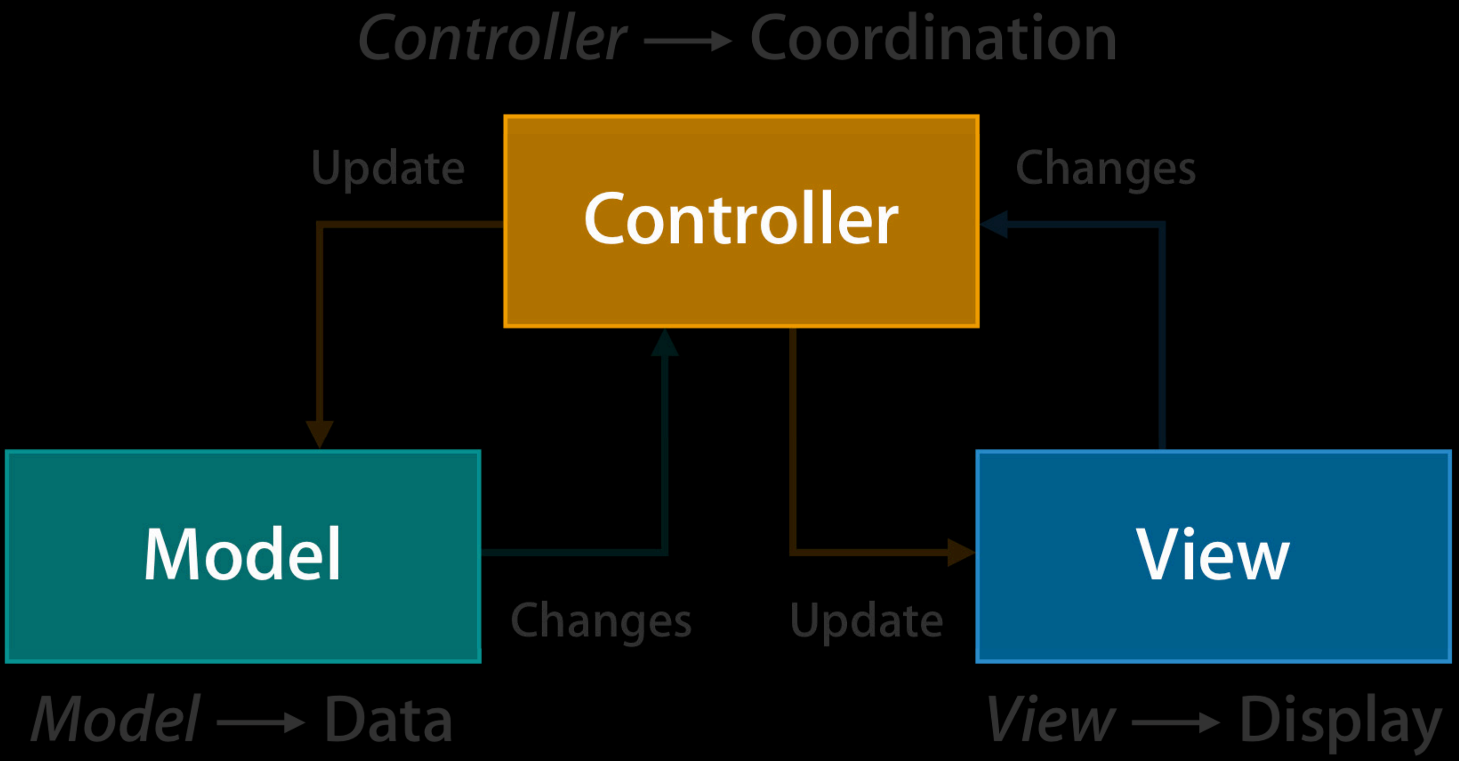
Model-View-Controller

Dividing work into objects



Model-View-Controller

Dividing work into objects



Model

Data/Algorithms/Networking

View

Display/Event Capture/Visual Appeal

Controller

Coordination/Delegation/Odd Jobs

Codebreaker

Model

- Ciphertext
- Plaintext
- Cryptography

The focus of app-specific work

Codebreaker

View

- List view
- Message view
- Add/Delete buttons

UIKit is a huge time-saver

Codebreaker

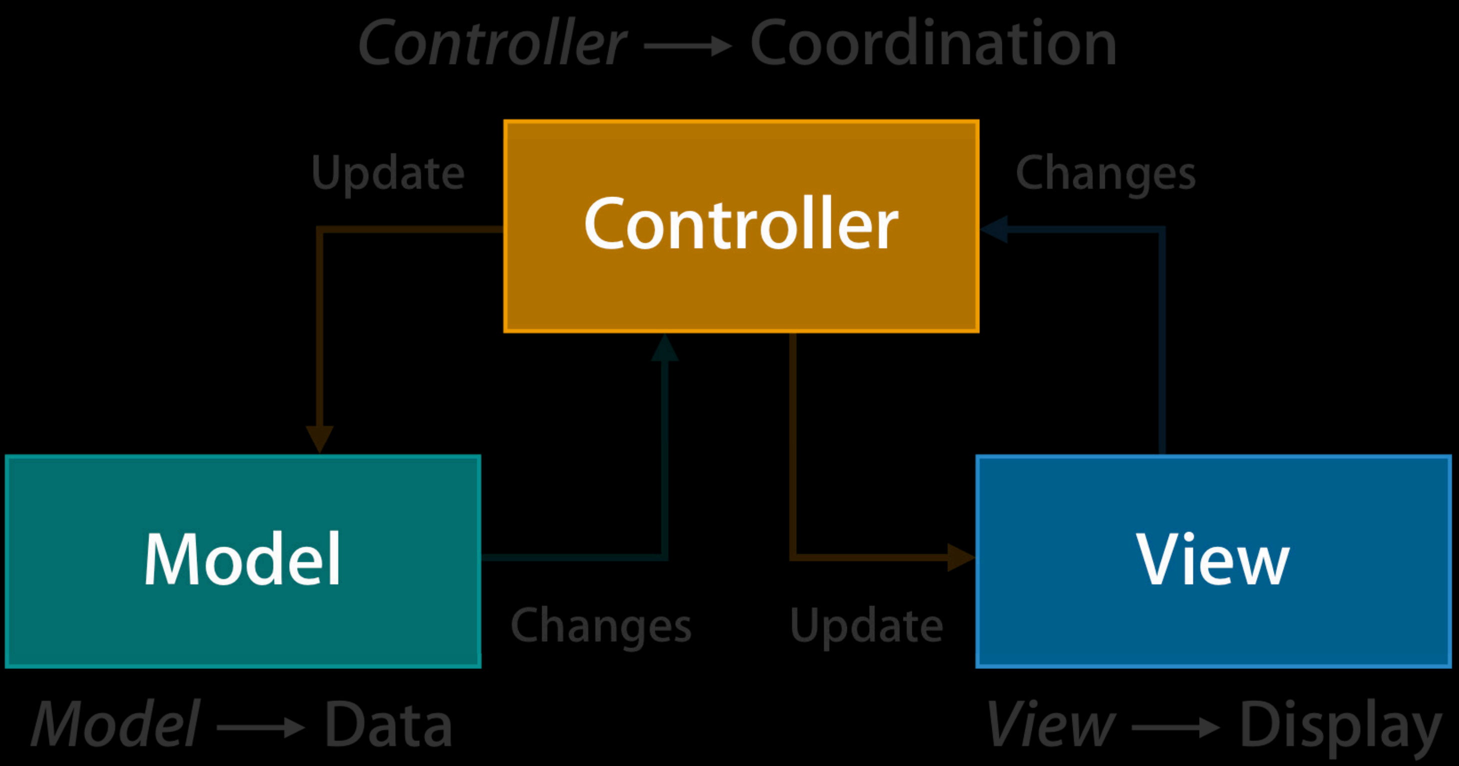
Controller

- Startup/Shutdown
- Navigation/Transitions
- Mediating between model and view

Match controllers to the right job

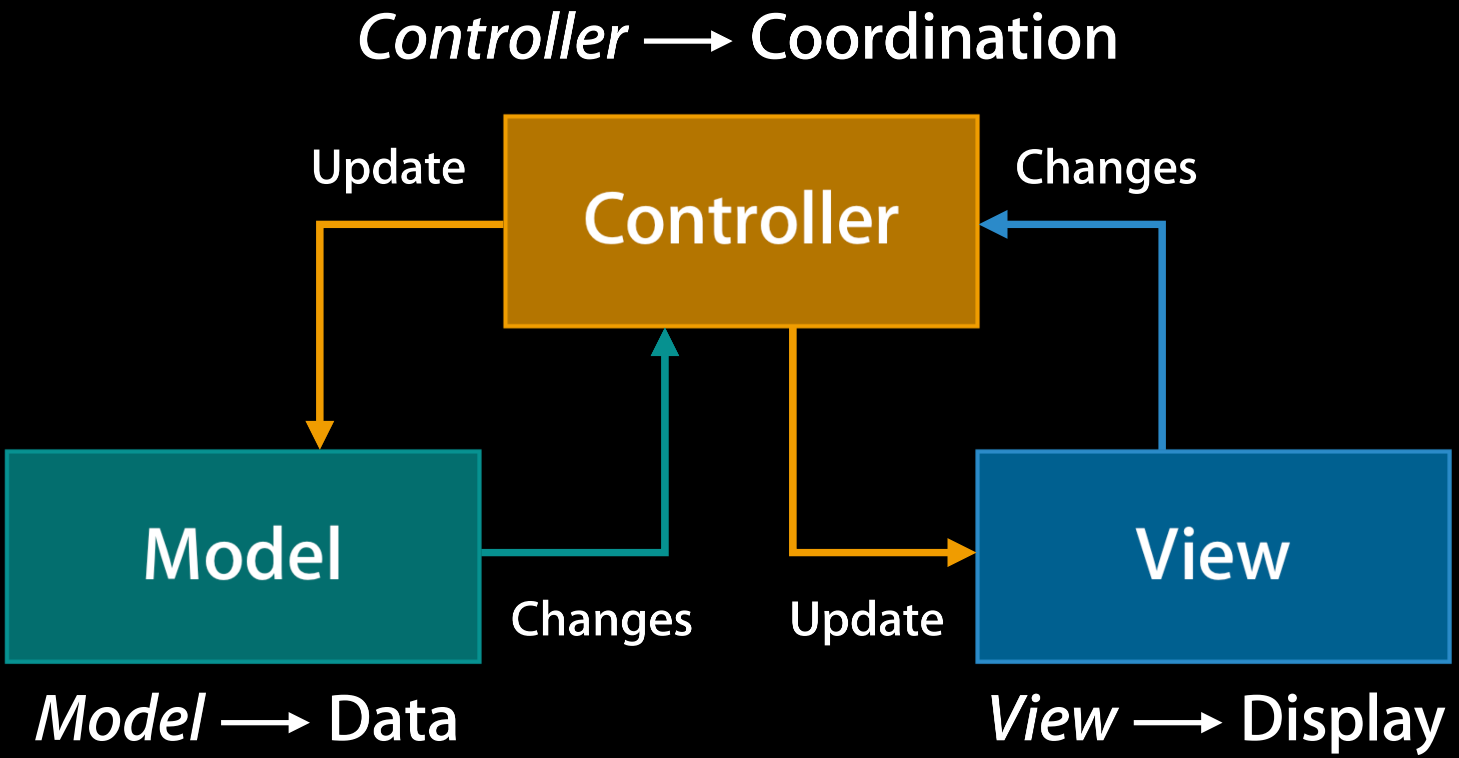
Model-View-Controller

Dividing work into objects



Model-View-Controller

Dividing work into objects



#2. Use MVC to Divide Work

Implement a big idea

Make manageable pieces

#3. Don’t Fight the Framework

#3. Don’t Fight the Framework

Color inside the lines

Make the framework work for you

Don’t Fight the Framework

Three examples

- ✗ Don’t misuse framework classes
  - Don’t remove views from UIViewControllers
- ✗ Don’t re-implement framework classes
  - If you need a split view, use UISplitViewController
- ✗ Don’t make trivial UIKit subclasses
  - Use delegates and notifications

Make the Framework Work for You

Codebreaker

Model

- Ciphertext
- Plaintext
  - ✓ **NSString**
- Cryptography
  - ✓ **Security Framework**

Codebreaker

View

- List view
  - ✓ **UITableView**
- Message view
  - ✓ **UIViewController**
- Add/Delete buttons
  - ✓ **UIControl/UIButton**

Codebreaker

Controller

- Startup/Shutdown
  - ✓ **AppController**
- Navigation/Transitions
  - ✓ **UINavigationController**
- Mediating between model and view
  - ✓ **Custom controller**
    UIKit delegate

#3. Don’t Fight the Framework

Color inside the lines

Make the framework work for you

#4. Don’t Abuse Views

#4. Don’t Abuse Views

Views don’t own data

Data display and event capture

If You Remember One Thing…

Views Don’t Own Data

Views Display Data, No?

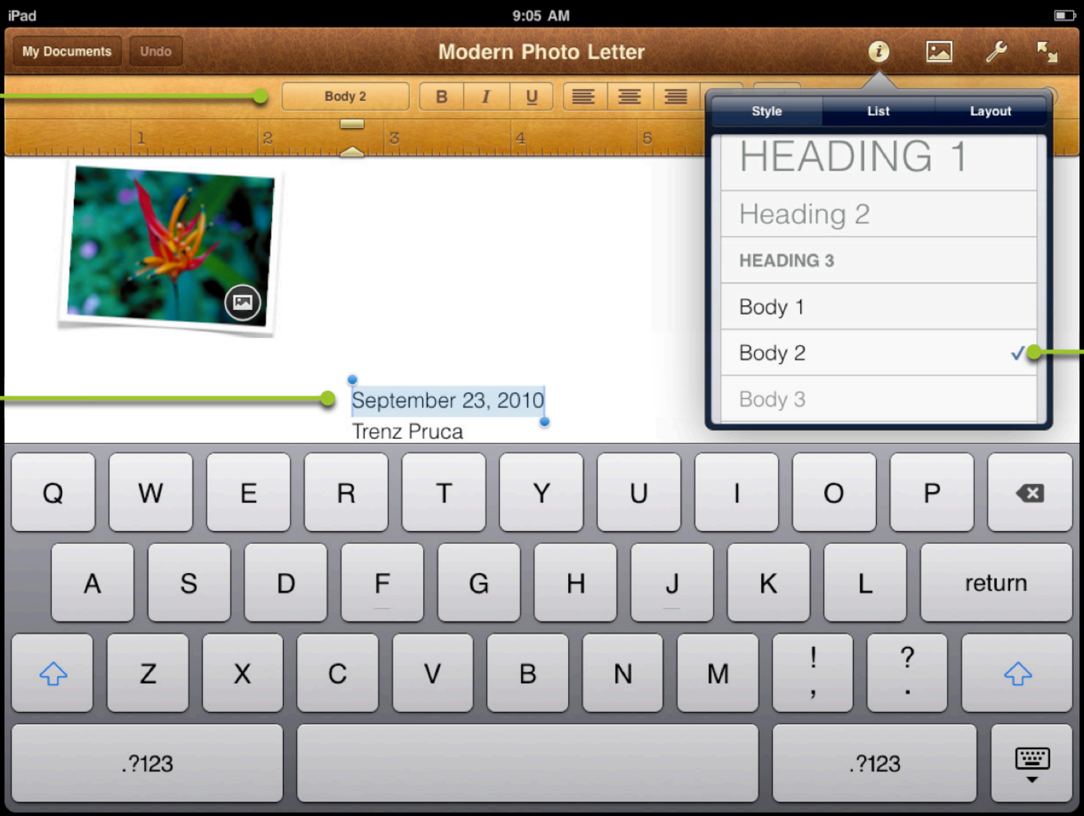
Views Don’t Own Data

The case against

- ✗ Slippery slope
  - First a little data, then data-change methods...
- ✗ Locks you into a view implementation
  - Change to a different view? Copy data?
- ✗ Tight coupling between data and display
  - Where do model and controllers fit in?

Vary Data Display

Inspectors



Model Data

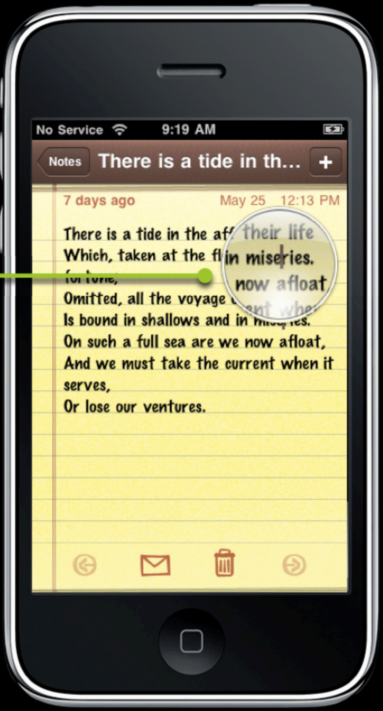
Model Data

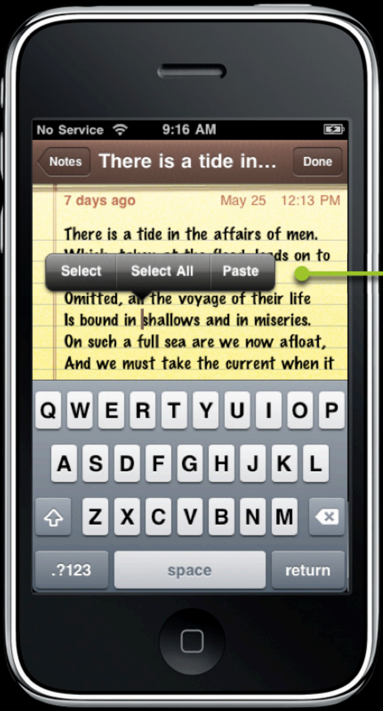
Model Data

Vary Behavior

UIGestureRecognizer

Read-only behavior

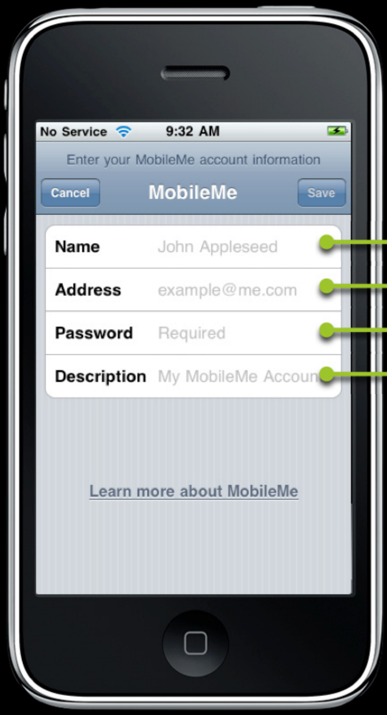

No Service
9:19 AM
Notes
There is a tide in th...
7 days ago
May 25 12:13 PM
There is a tide in the affairs of men.
Which, taken at the flood, leads on to fortune;
Omitted, all the voyage of their life
Is bound in shallows and in miseries.
On such a full sea are we now afloat,
And we must take the current when it serves,
Or lose our ventures.


No Service
9:16 AM
Notes
There is a tide in...
Done
7 days ago
May 25 12:13 PM
Select
Select All
Paste
There is a tide in the affairs of men.
Omitted, all the voyage of their life
Is bound in shallows and in miseries.
On such a full sea are we now afloat,
And we must take the current when it
space
return
.?123

Editing behavior

Promote Reusability

Simple, but effective


No Service 9:32 AM

Enter your MobileMe account information

Name	John Appleseed
Address	example@me.com
Password	Required
Description	My MobileMe Accoun

Subclass for name?
Subclass for address?
Subclass for password?
Is there an echo in here?

#4. Don’t Abuse Views

Views don’t own data

Data display and event capture

#5. Plan for iPhone and iPad

#5. Plan for iPhone and iPad

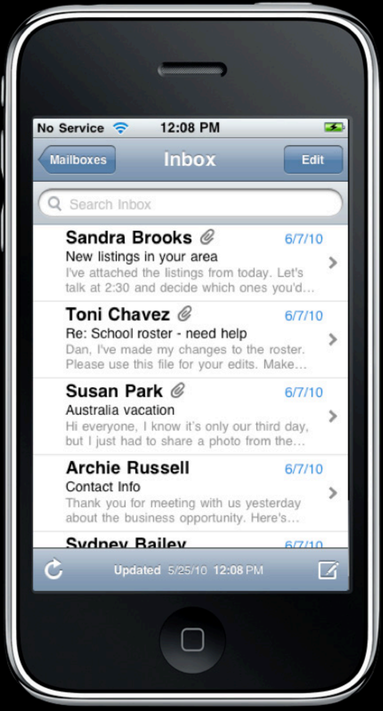
Divide code into modules

Higher-level design than MVC

Mail on iPhone

Mail on iPhone

Shipped in 1.0



Sandra Brooks 6/7/10
New listings in your area
I've attached the listings from today. Let's talk at 2:30 and decide which ones you'd…

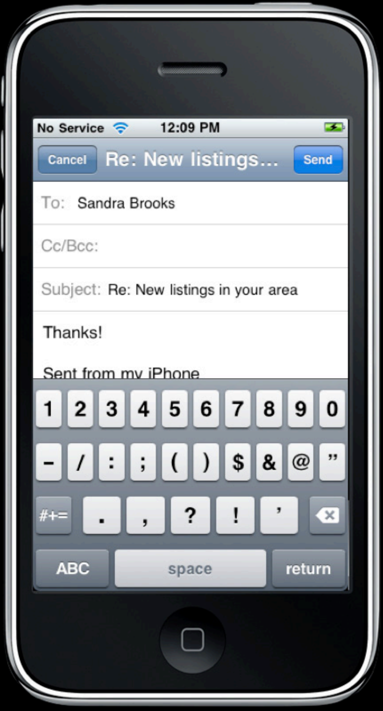
Toni Chavez 6/7/10
Re: School roster - need help
Dan, I've made my changes to the roster. Please use this file for your edits. Make…

Susan Park 6/7/10
Australia vacation
Hi everyone, I know it's only our third day, but I just had to share a photo from the…

Archie Russell 6/7/10
Contact Info
Thank you for meeting with us yesterday about the business opportunity. Here's…

Sydney Bailey 6/7/10

Updated 5/25/10 12:08 PM



To: Sandra Brooks

Cc/Bcc:

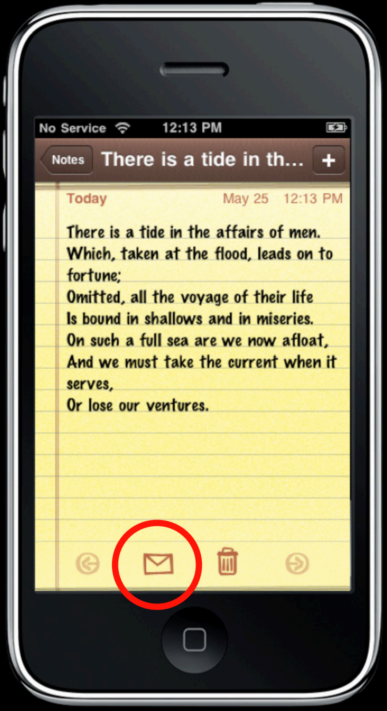
Subject: Re: New listings in your area

Thanks!

Sent from my iPhone

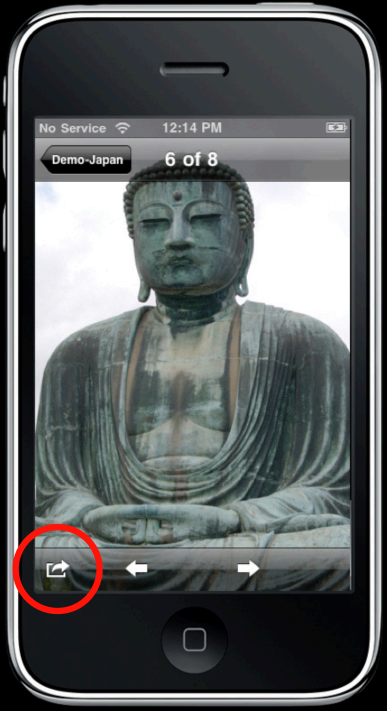
Mail as a System Feature

Other applications can send mail



There is a tide in the affairs of men.
Which, taken at the flood, leads on to fortune;
Omitted, all the voyage of their life
Is bound in shallows and in miseries.
On such a full sea are we now afloat,
And we must take the current when it serves,
Or lose our ventures.

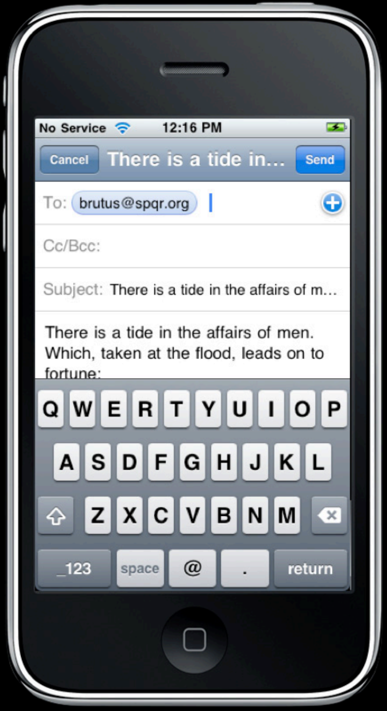
Notes



Photos

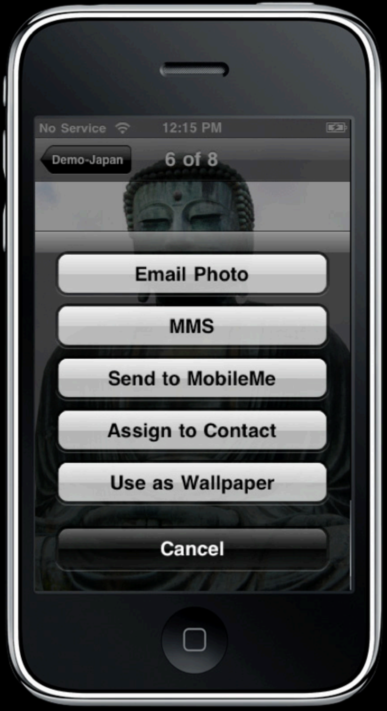
Mail as a System Feature

Other applications can send mail



There is a tide in the affairs of men.
Which, taken at the flood, leads on to
fortune;

Notes

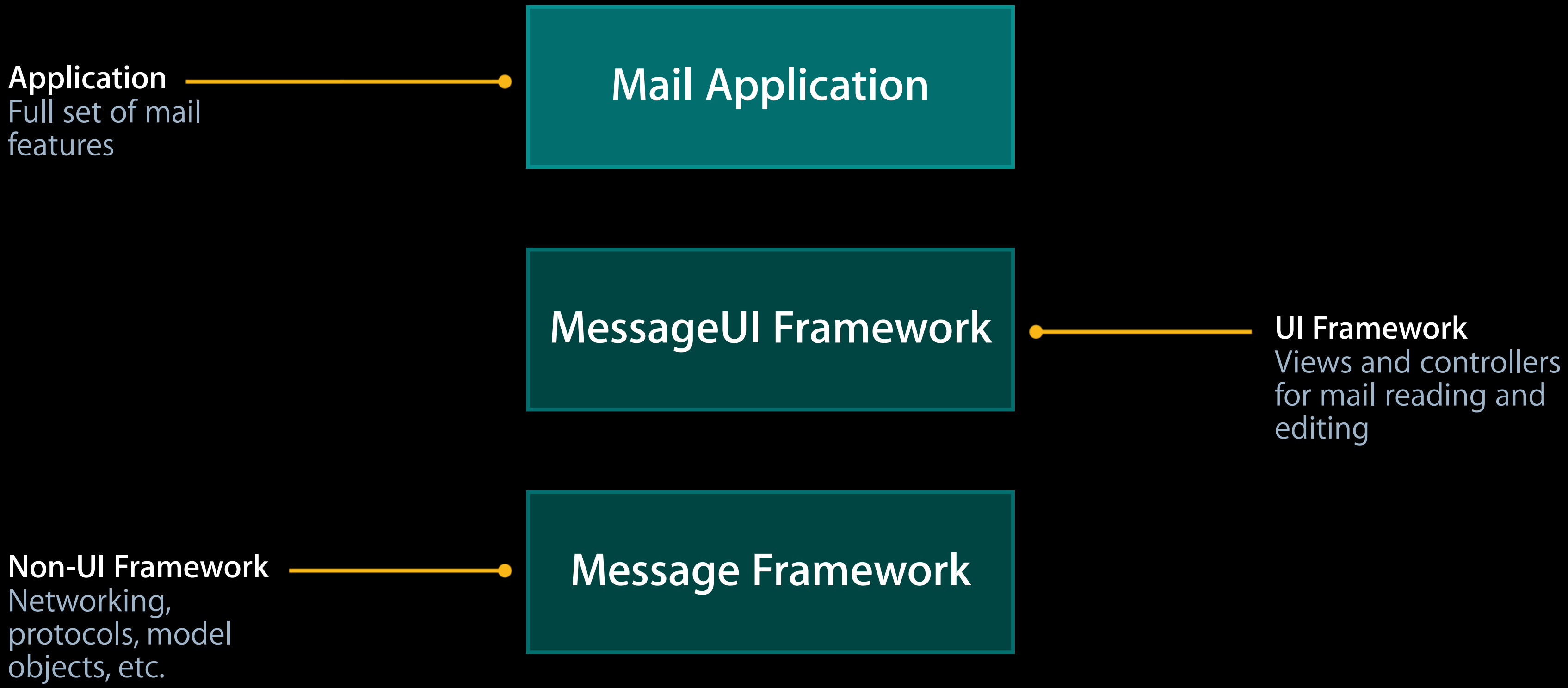


Photos

Factor Out Common Pieces

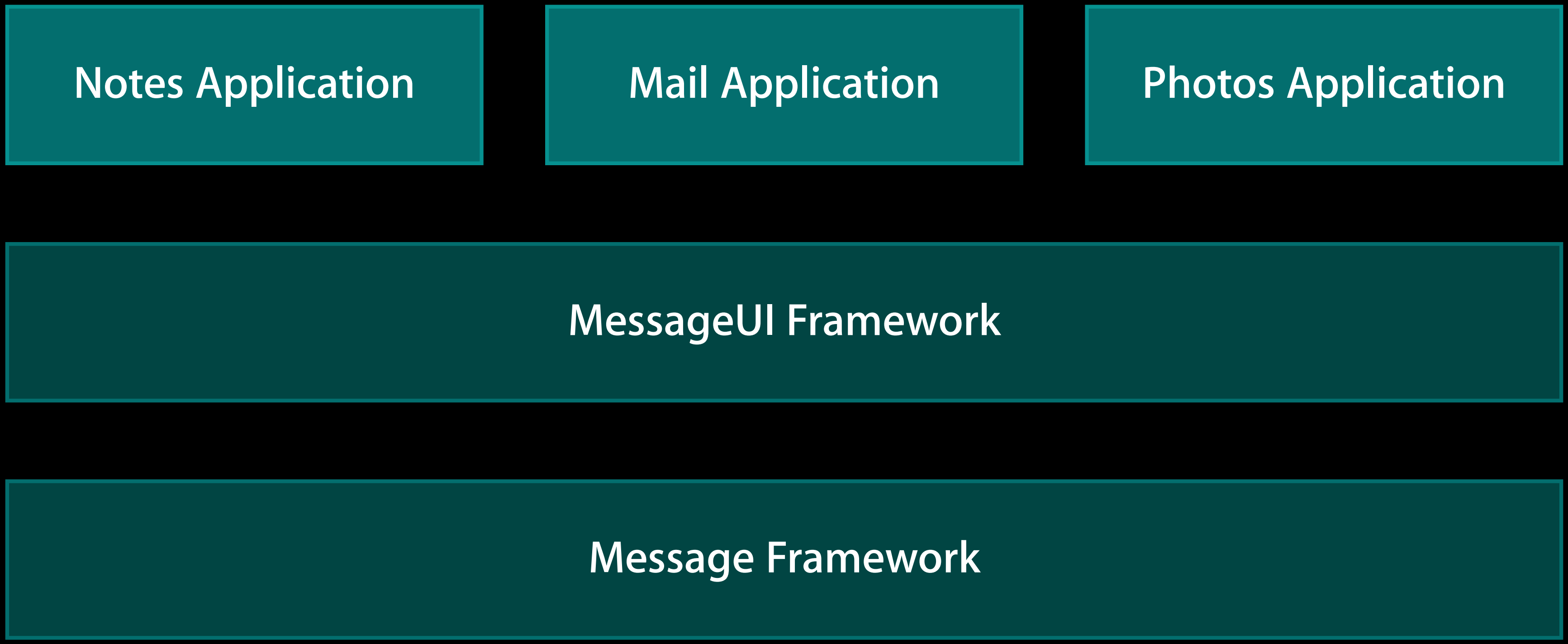
Factor Out Common Pieces

Three-part arrangement


Application
Full set of mail features
Mail Application
MessageUI Framework
UI Framework
Views and controllers for mail reading and editing
Non-UI Framework
Networking, protocols, model objects, etc.
Message Framework

Factor Out Common Pieces

Support additional applications



Hello iPad

iPad
9:41 AM
Wednesday
27
Calendar
Contacts
Notes
Maps
Videos
YouTube
iTunes
App Store
Settings
Safari
Mail
Photos
iPod

Hello iPad

Make great new versions of existing programs

There is a tide in the affairs of
Which, taken at the flood, lead
Omitted, all the voyage of their
Is bound in shallows and in mise
On such a full sea are we now a
And we must take the current
Or lose our ventures.

To: Sandra Brooks

Cc/Bcc, From: iphonedemo102@me.com

Subject: Re: New listings in your area

Thanks!

Sent from my iPad

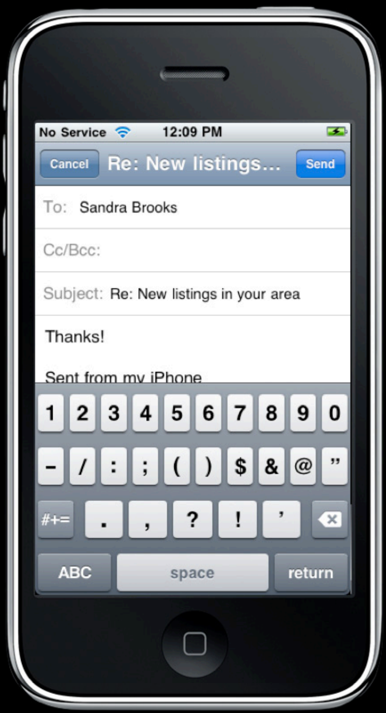
On Jun 7, 2010, at 9:41 AM, Sandra Brooks <erosenthal@apple.com> wrote:

I've attached the listings from today. Let's talk at 2:30 and decide which ones you'd like to see.

- sandra

Real World Code Reuse

Compose View is the same



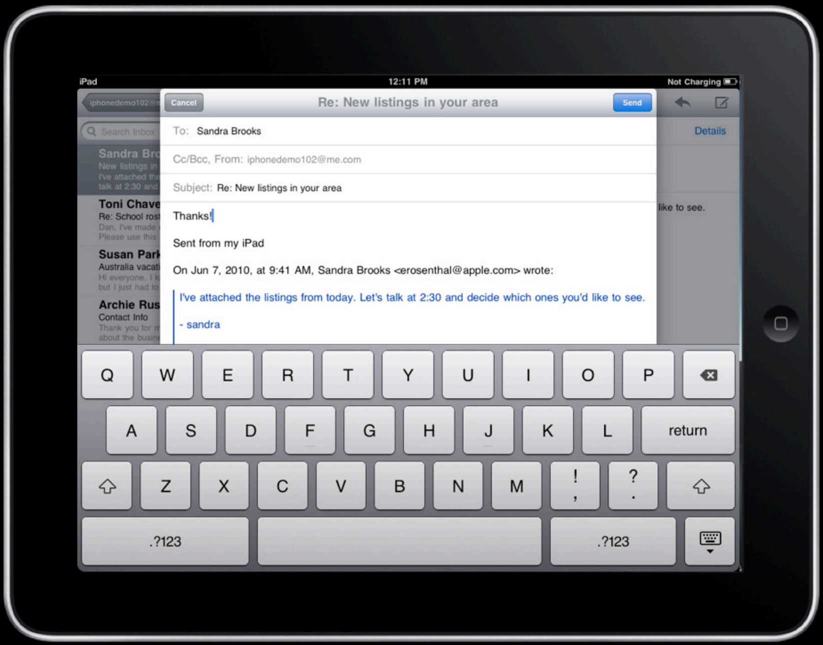
To: Sandra Brooks

Cc/Bcc:

Subject: Re: New listings in your area

Thanks!

Sent from my iPhone



To: Sandra Brooks

Cc/Bcc, From: iphonedemo102@me.com

Subject: Re: New listings in your area

Thanks!

Sent from my iPad

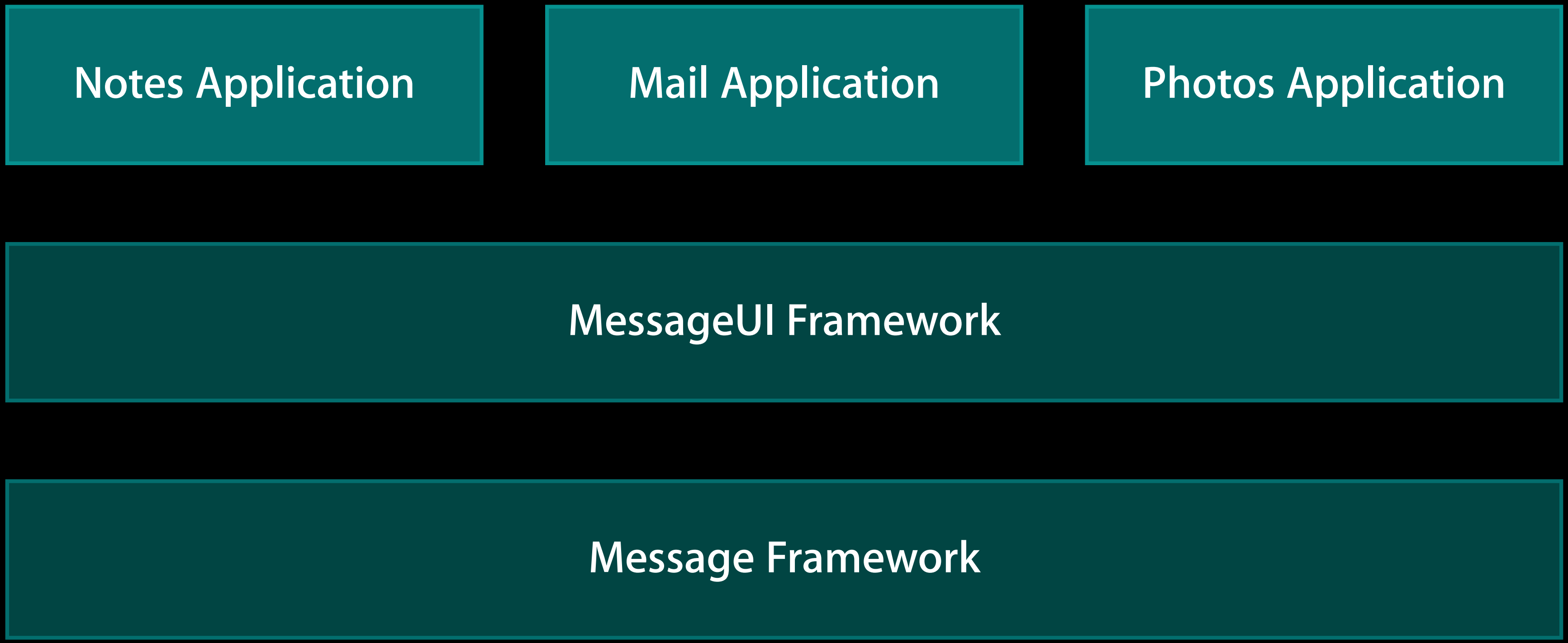
On Jun 7, 2010, at 9:41 AM, Sandra Brooks <erosenthal@apple.com> wrote:

I've attached the listings from today. Let's talk at 2:30 and decide which ones you'd like to see.

- sandra

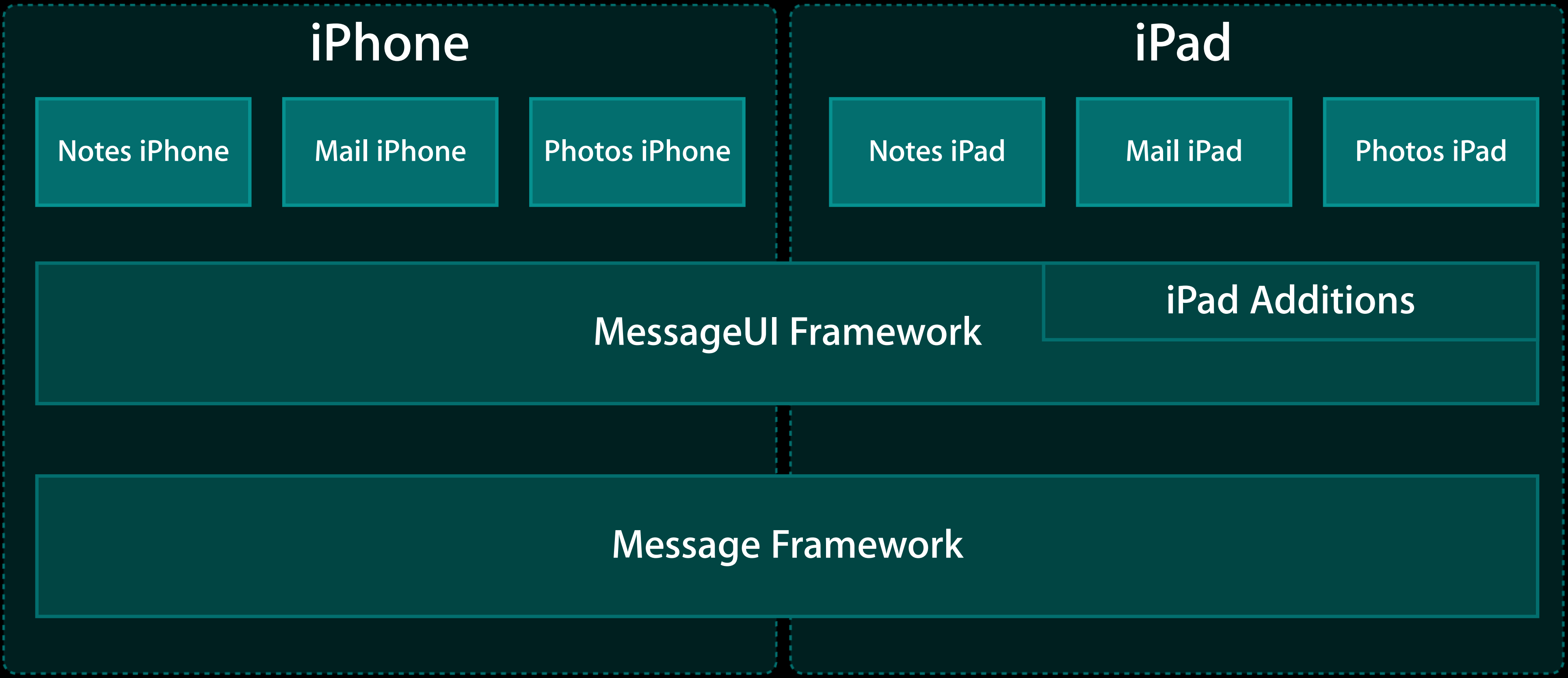
Hello iPad

Make great new versions of existing programs



iPhone/iPad Architecture

Support both devices



Real World?

No Third-Party Frameworks!

Static Library

Simple Code Sharing

#5. Plan for iPhone and iPad

Divide code into modules

Higher-level design than MVC

#6. Strive for Loose Coupling

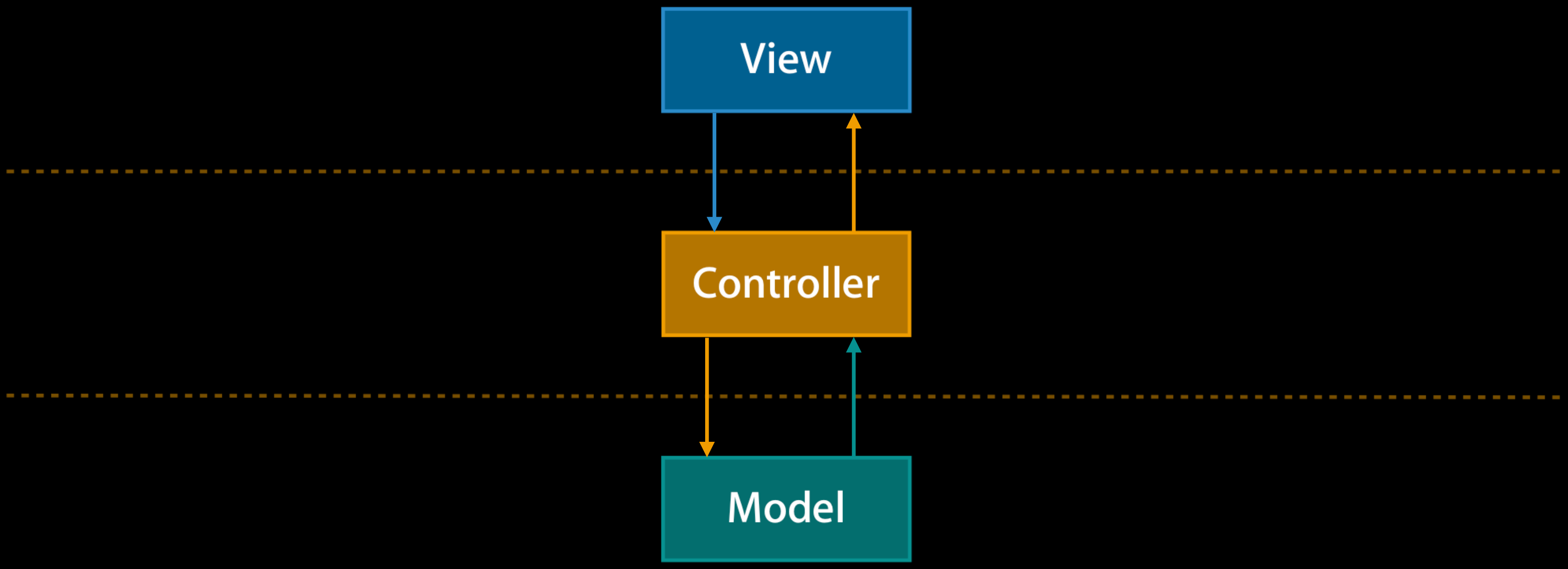
#6. Strive for Loose Coupling

Goal is flexibility

Minimizing mutual dependencies

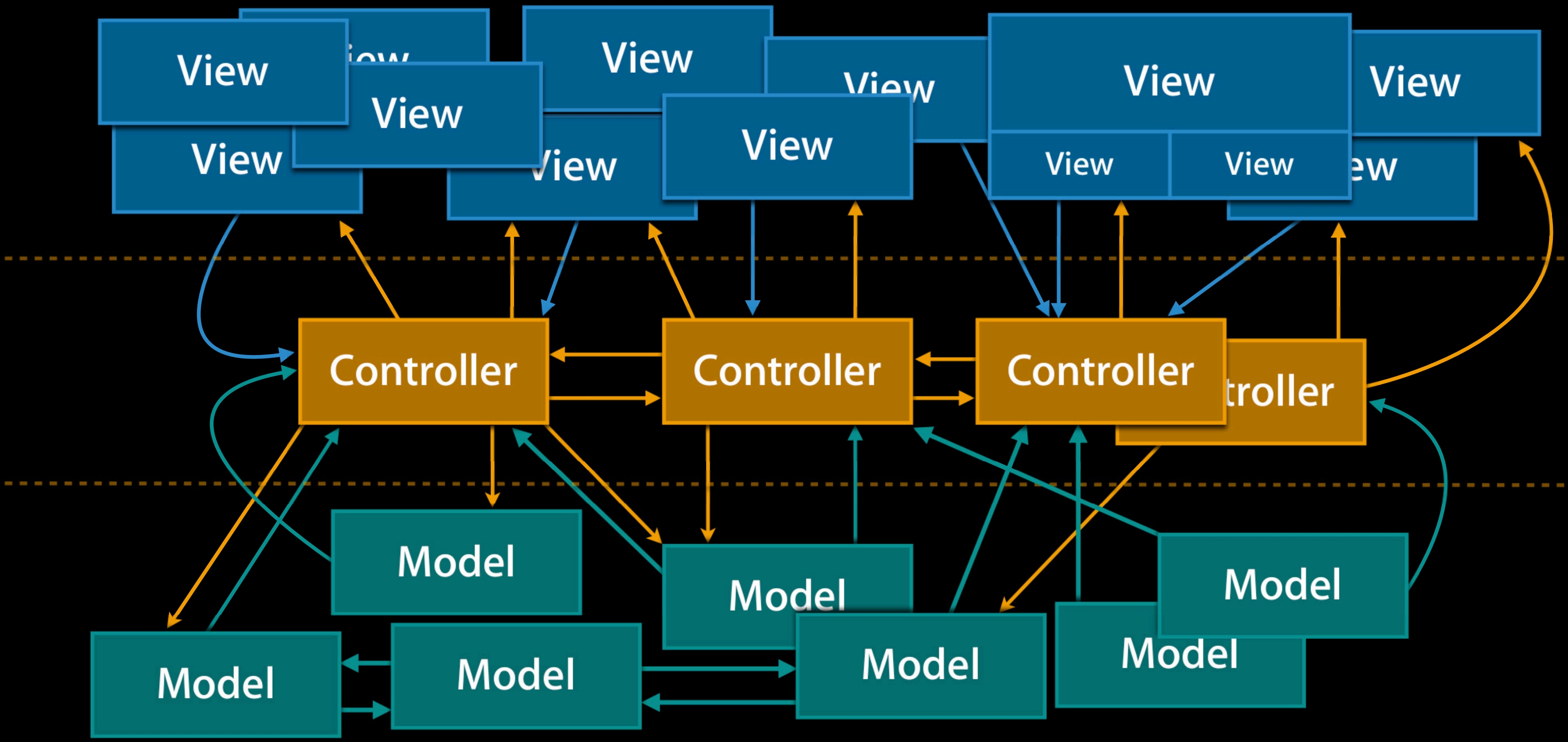
MVC Architecture

In concept



MVC Architecture

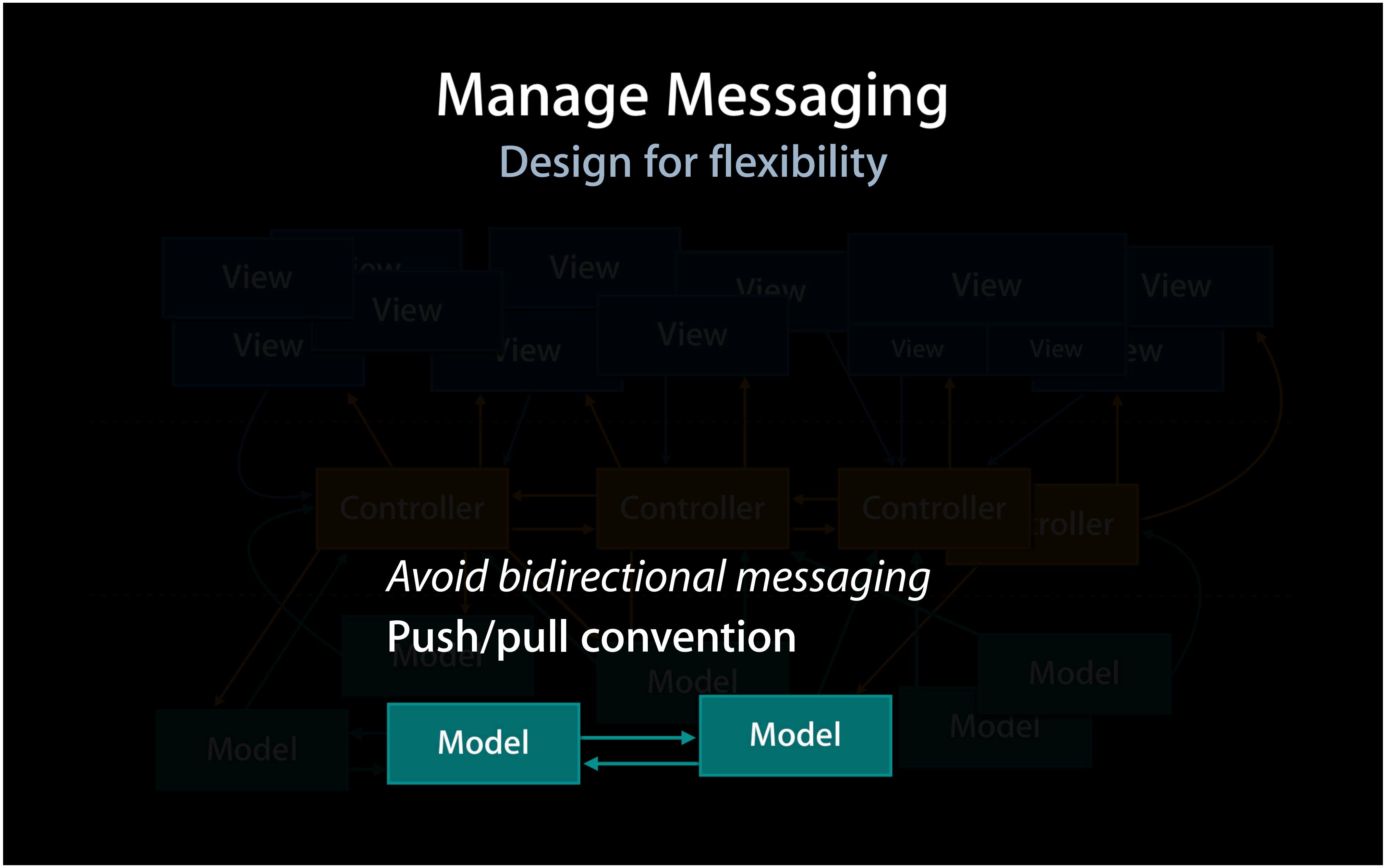
In the real world



Strive for Loose Coupling

Design for flexibility

- ✗ Don’t skip MVC layers when messaging
  - Use controllers to coordinate messages
- ✗ Don’t mix MVC roles in one object
  - Avoid gathering too much work into one place
- ✗ Don’t declare model data in your view classes
  - Is there an echo in here?



Manage Messaging

Design for flexibility

Avoid bidirectional messaging

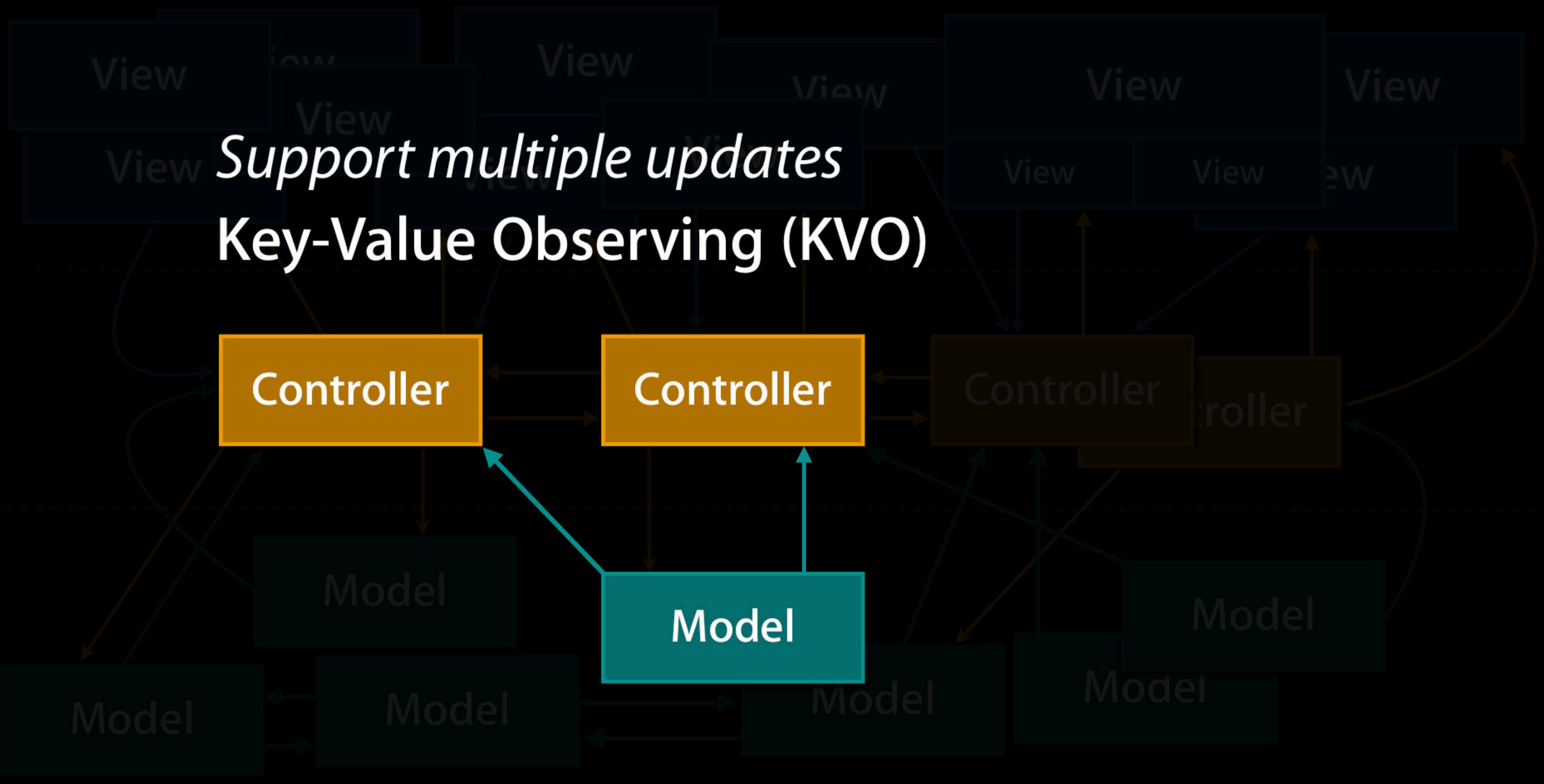
Push/pull convention

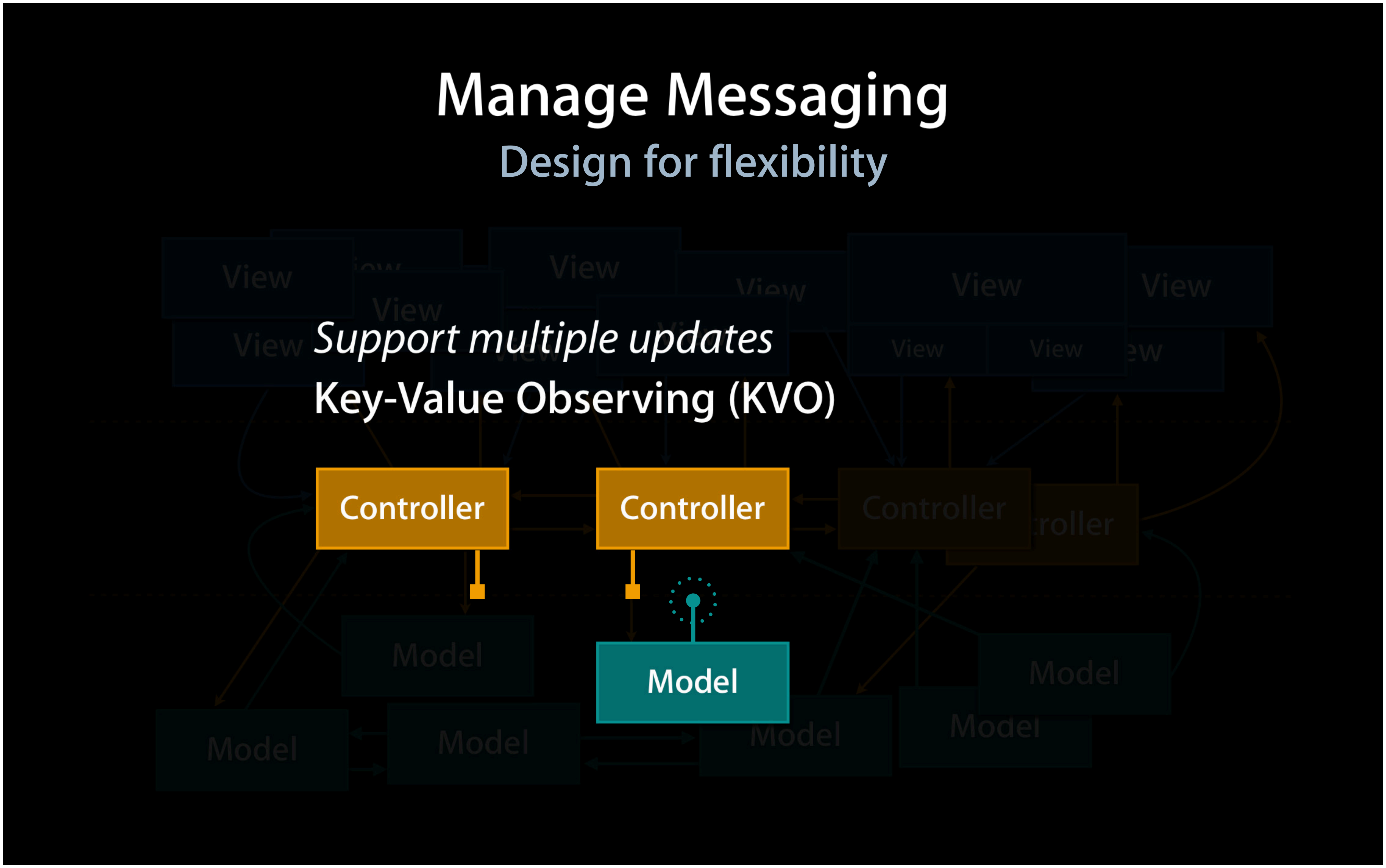
Manage Messaging

Design for flexibility

Support multiple updates

Key-Value Observing (KVO)





Manage Messaging

Design for flexibility

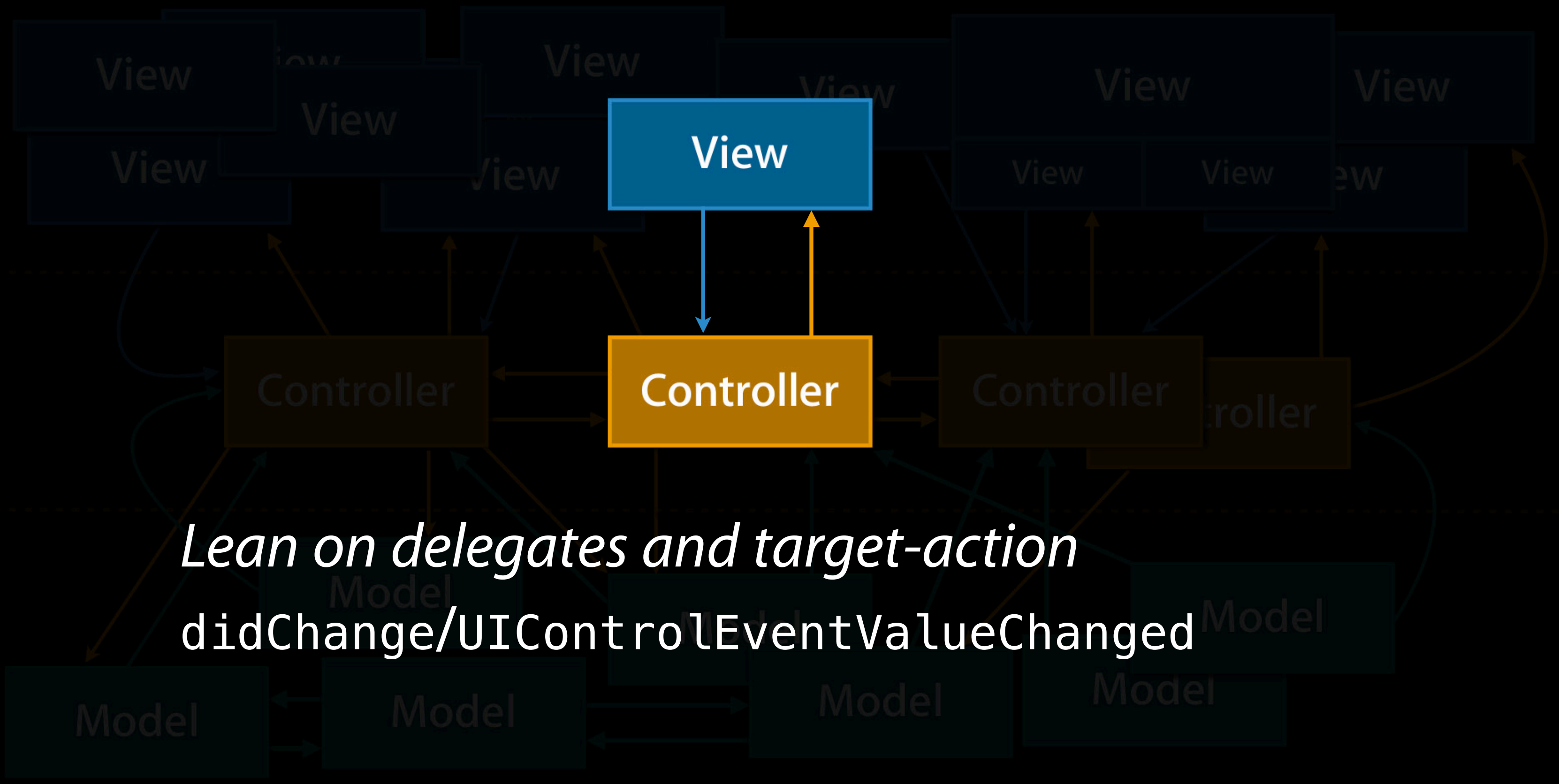
Support multiple updates

Key-Value Observing (KVO)

Controller
Controller
Model

Manage Messaging

Design for flexibility

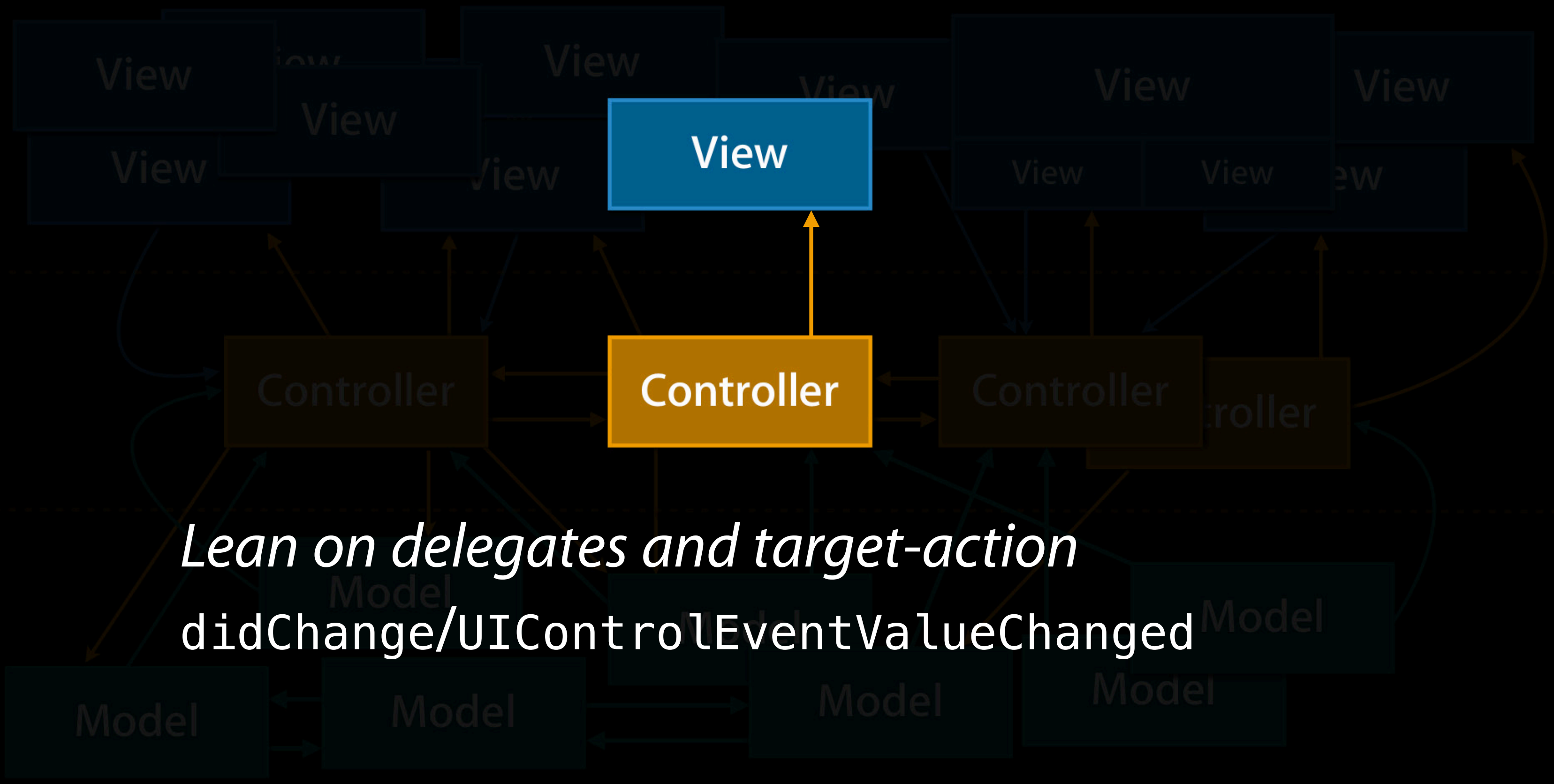


Lean on delegates and target-action

`didChange/UIControlEventValueChanged`

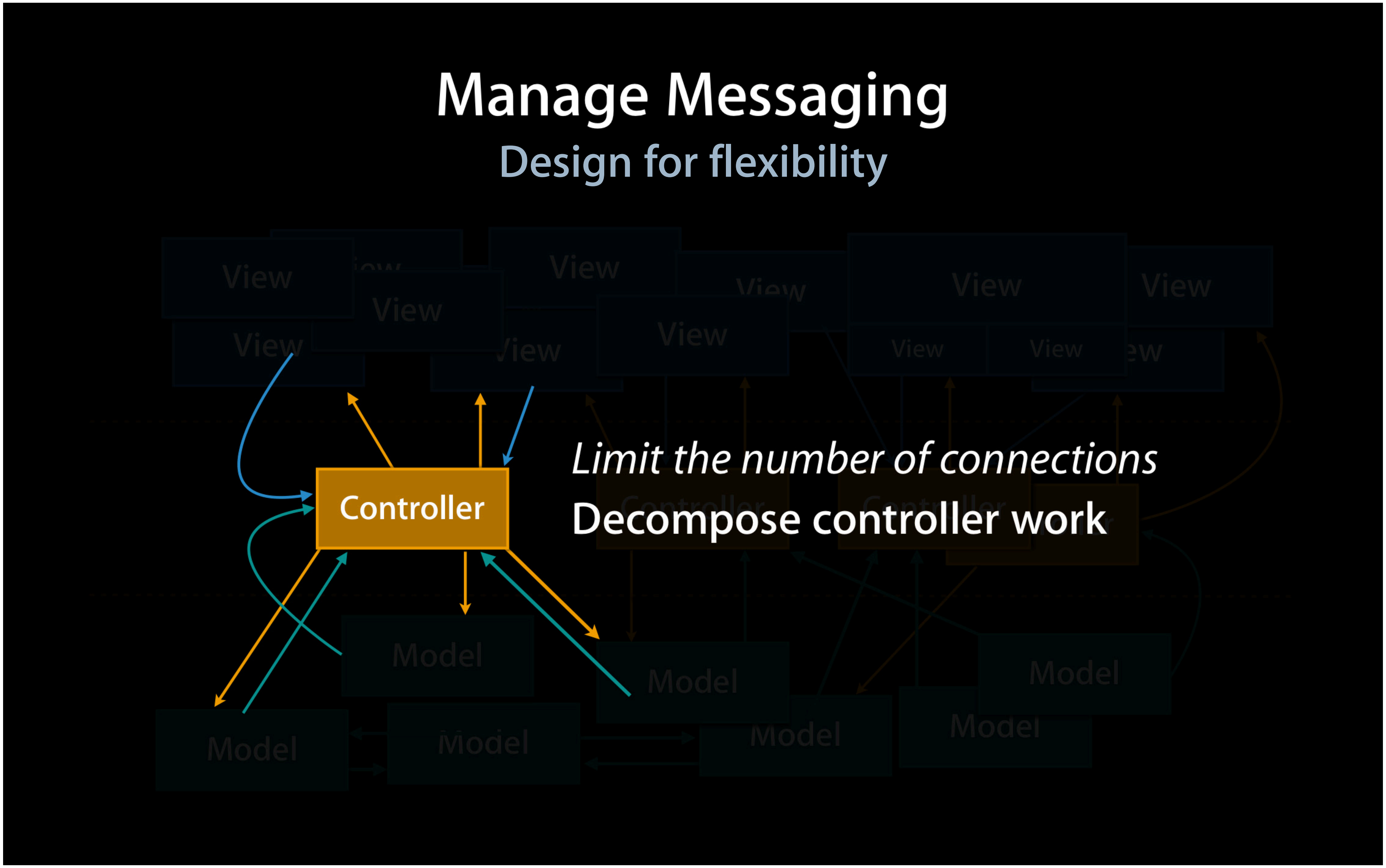
Manage Messaging

Design for flexibility



Lean on delegates and target-action

`didChange/UIControlEventValueChanged`



Manage Messaging

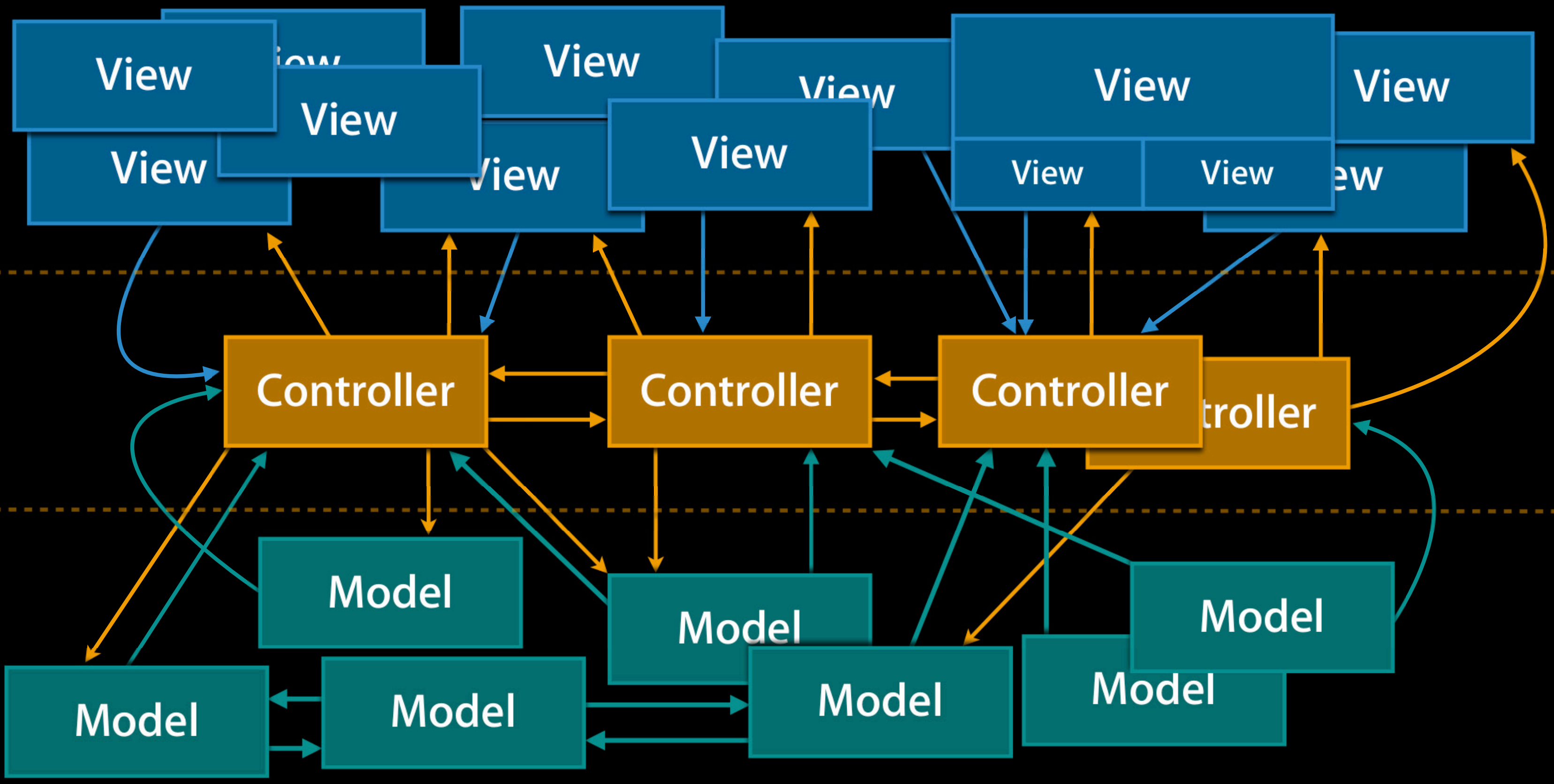
Design for flexibility

Limit the number of connections

Decompose controller work

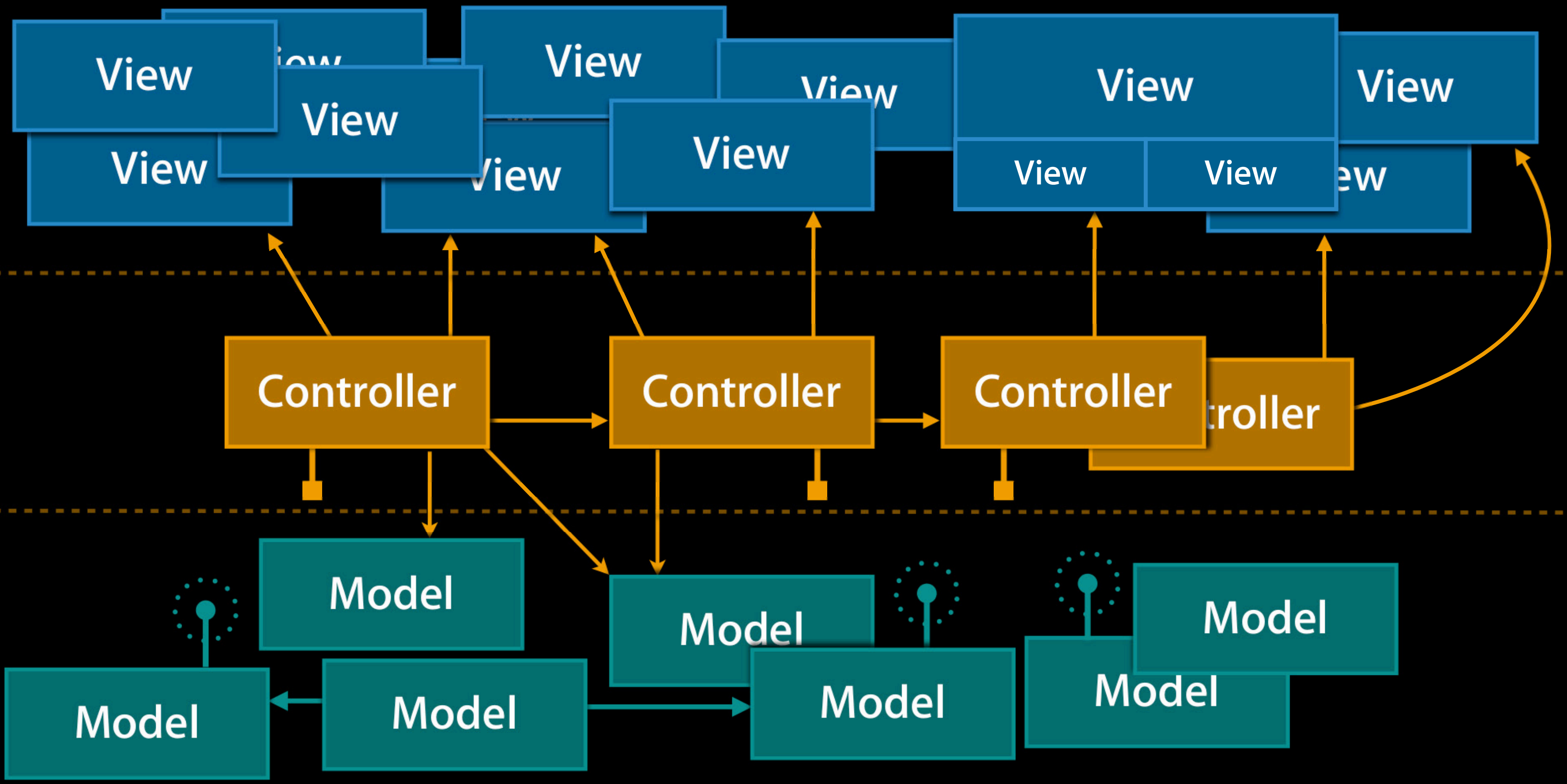
MVC Architecture

In the real world



MVC Architecture

In the real world



#6. Strive for Loose Coupling

Goal is flexibility

Minimizing mutual dependencies

#7. Choose the Right Data Model

#7. Choose the Right Data Model

iPhone OS gives you many options

Finding the right fit

Academic Purity?

A relvar R [table] is in *sixth normal form*...if and only if it satisfies no nontrivial join dependencies at all — where, as before, a join dependency is trivial if and only if at least one of the projections (possibly U_projections) involved is taken over the set of all attributes of the relvar [table] concerned.

Date, C.J. (2006). The relational database dictionary: a comprehensive glossary of relational terms and concepts, with illustrative examples. O'Reilly Series Pocket references. O'Reilly Media, Inc.. p. 90. ISBN 9780596527983.

http://en.wikipedia.org/wiki/Sixth_normal_form

Sixth Normal Form?

Objects

$\text{Runtime} \longrightarrow \text{Saved} \longrightarrow \text{Runtime}$

Data Model Concerns

Serialization

Transactions

Undo

Ease Of Use

Scale

Faults

Object/Relational Mapping

Partial Graphs

Speed

Versioning

Modeling Tools

I/O Churn

Memory Use

SQL

Constraints

Legacy Data

Interdependencies

Model Options

Model Options

- ✓ Property Lists
- ✓ Server/Cloud
- ✓ Archives
- ✓ SQLite
- ✓ Custom Files
- ✓ CoreData

Defaults/Preferences

Wrong tool for the job

Settings Panel test

Property Lists

Simple to use

Strings, numbers, arrays, dictionaries, etc.

Archives

Simple to use

```
– initWithCoder:  – encodeWithCoder:
```

Custom Files

Legacy Code and Data

Create NSObject-based graph

Server/Cloud

✓

High-score list

NSURL loading classes. Server is up to you.

SQLite ✓

Familiar with SQL

Object/Relational Mapping

CoreData

Wealth of features

Investment

Strongly Consider CoreData

Strongly Consider CoreData

Wealth of features

Modeling tools

Simple saving/Restoring

Queries

Undo

Partial graphs

Model Options

- ✓ Property Lists
- ✓ Server/Cloud
- ✓ Archives
- ✓ SQLite
- ✓ Custom Files
- ✓ CoreData

#7. Choose the Right Data Model

iPhone OS gives you many options

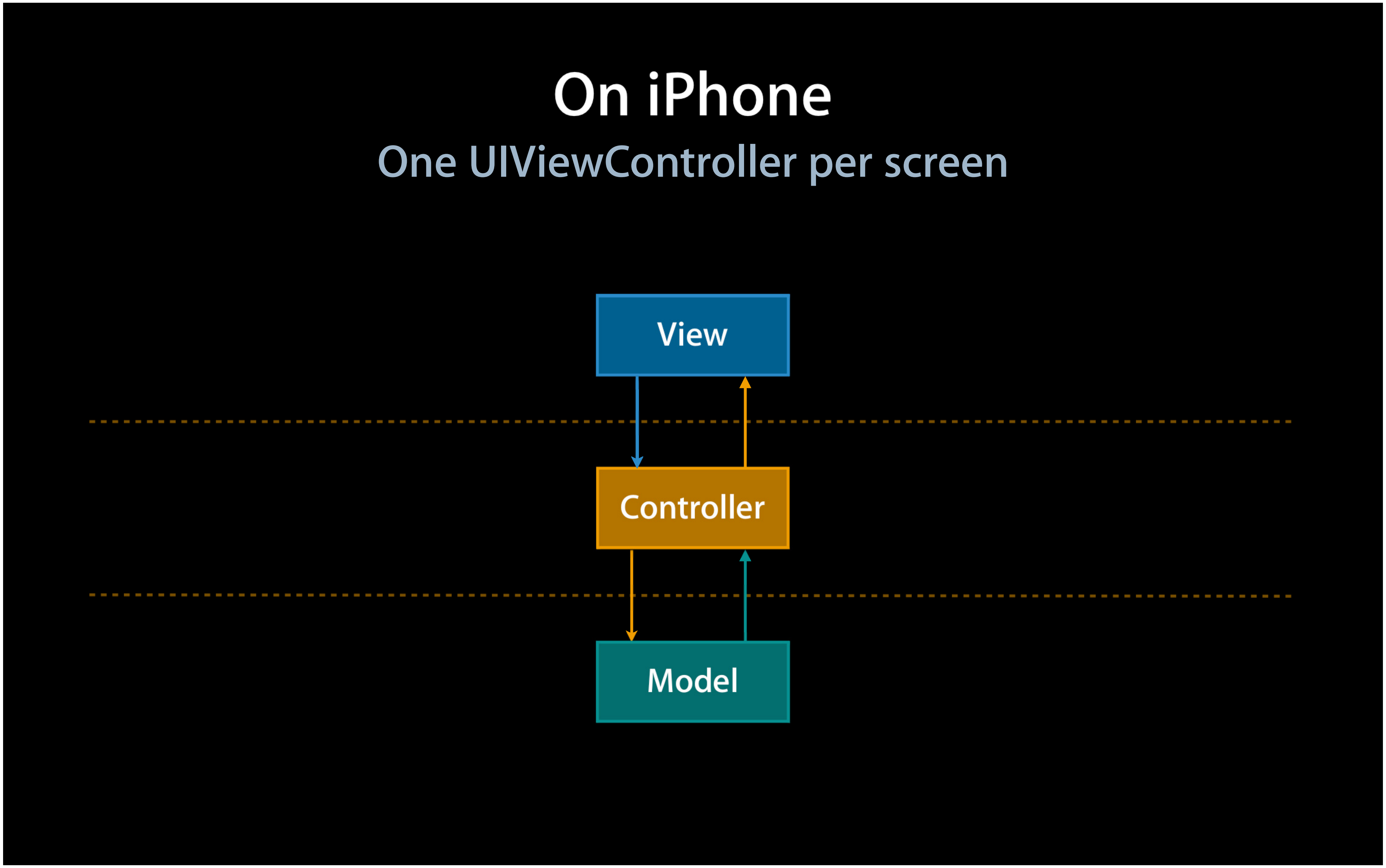
Finding the right fit

#8. Decompose Controller Work

#8. Decompose Controller Work

The right number of controllers

Special iPhone OS controllers



On iPhone

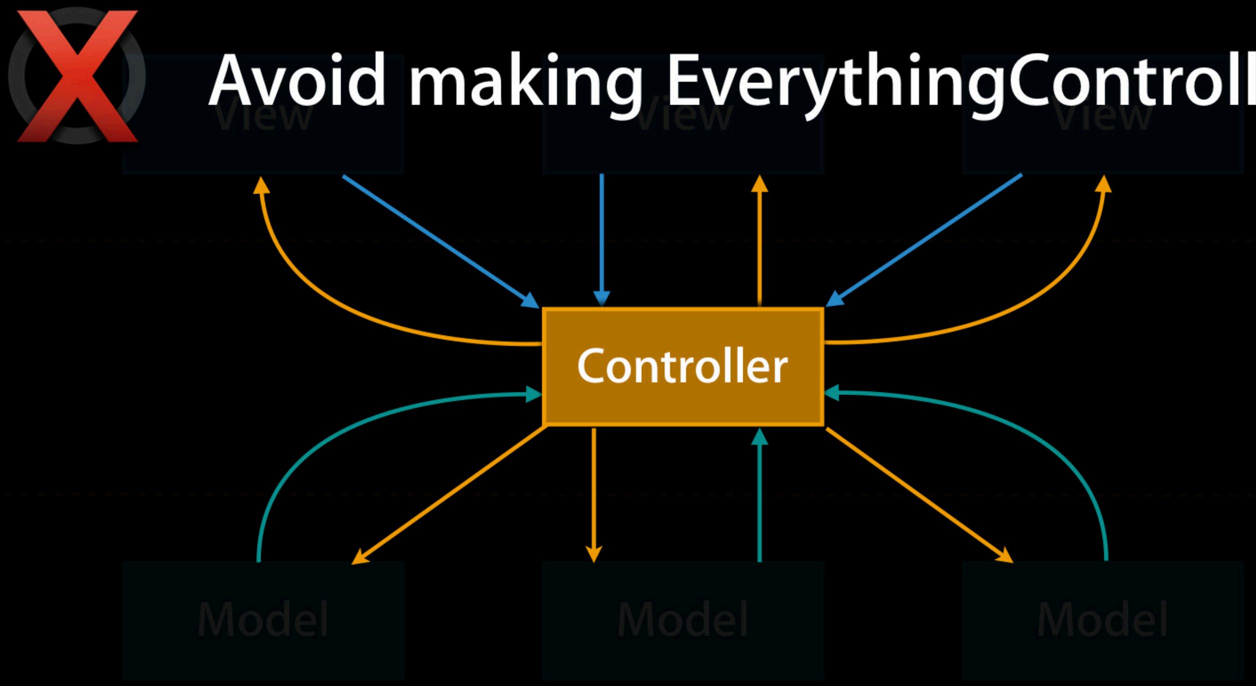
One UIViewController per screen

$$\begin{tikzcd}
\boxed{\text{View}} \arrow[d, shift right] \\
\boxed{\text{Controller}} \arrow[u, shift right] \arrow[d, shift right] \\
\boxed{\text{Model}} \arrow[u, shift right]
\end{tikzcd}$$

On iPad

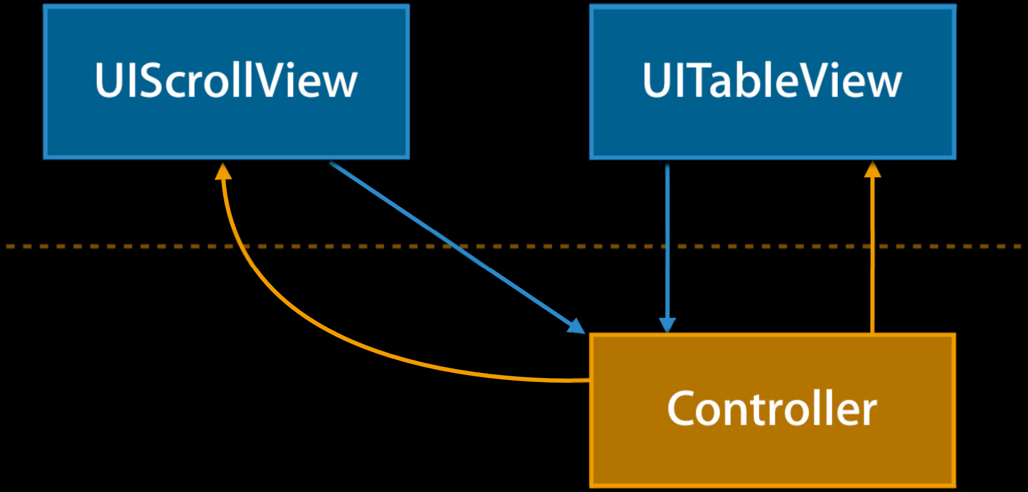
One UIViewController per screen?

✗ Avoid making EverythingControllers



On iPad

One UIViewController per screen?



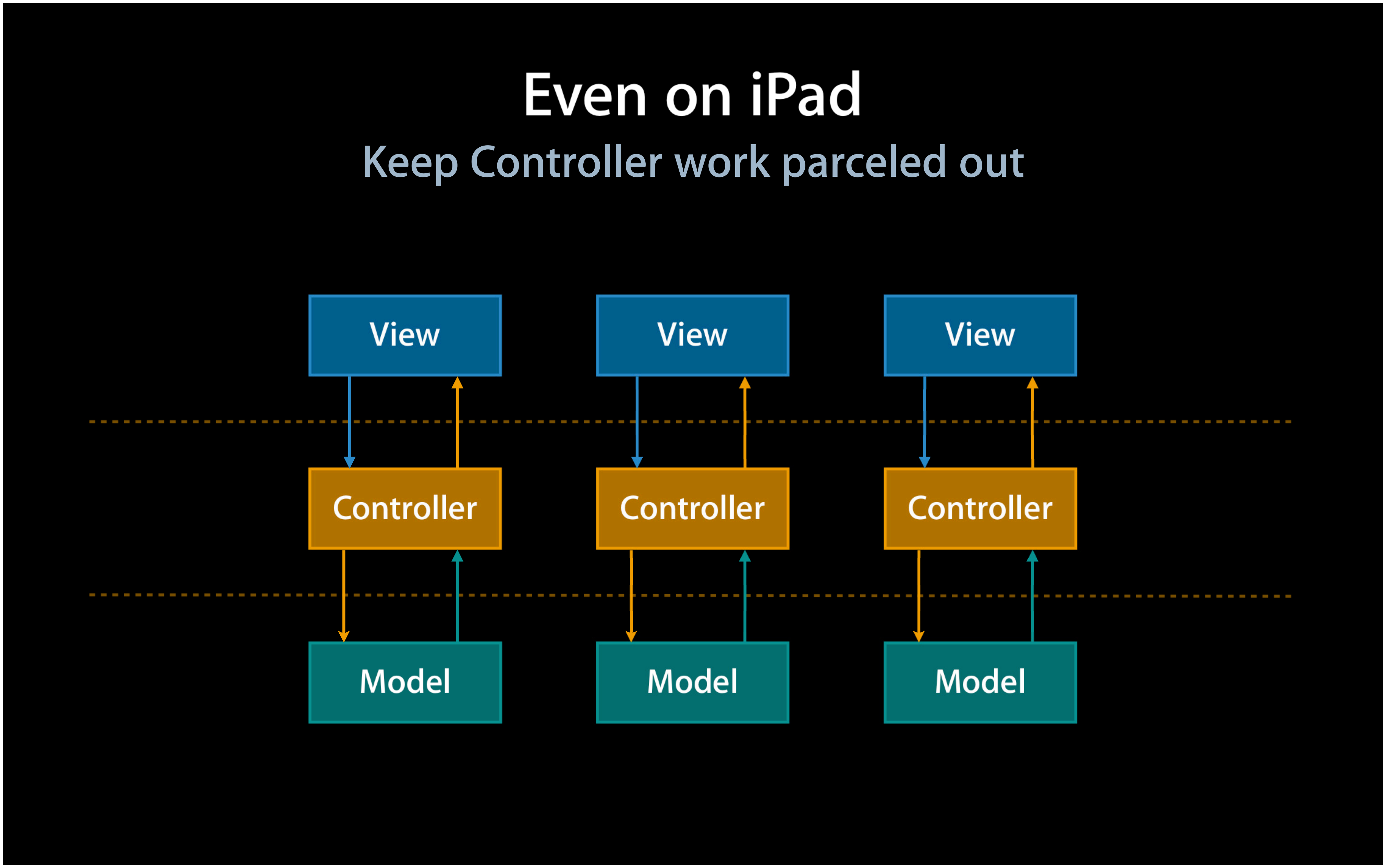
```
- (void)scrollViewDidScroll:(UIScrollView *)scrollView
{
    if ([scrollView isKindOfClass:[UITableView class]]) {
        ...
    } else {
        // "regular" scroll view
}
```

On iPad

One UIViewController per screen?

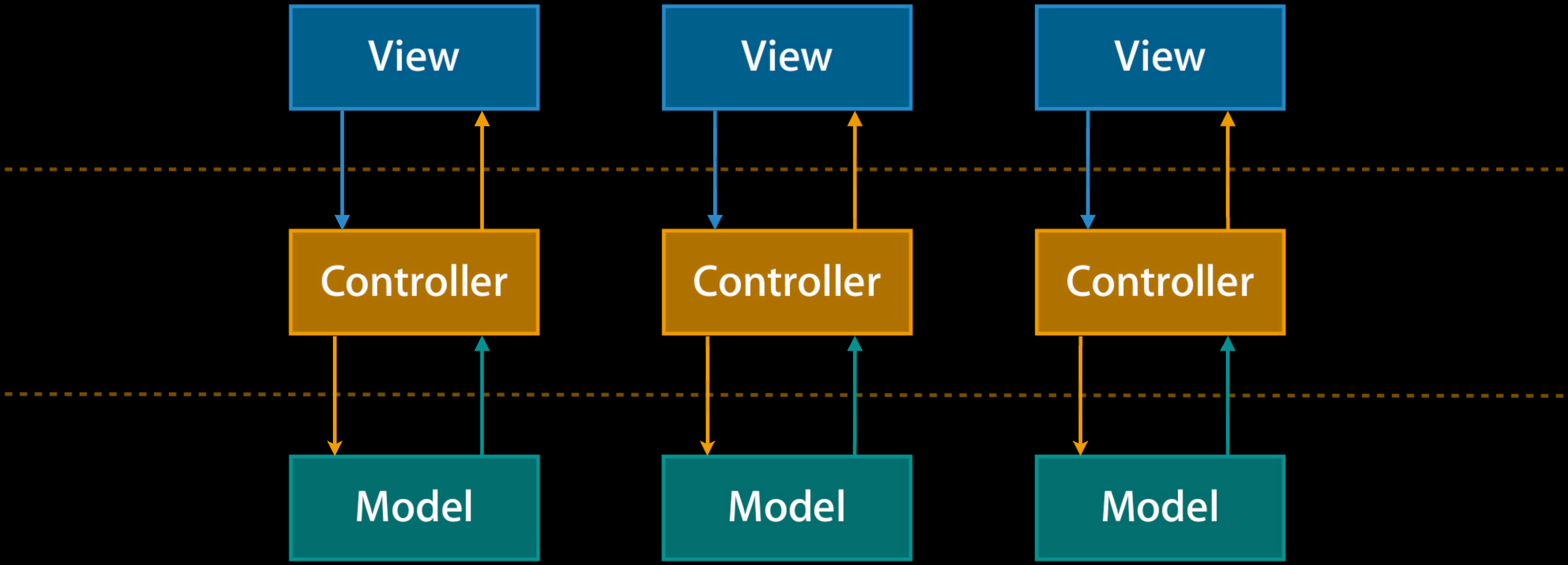
✗ Class checks in delegate methods

```
- (void)scrollViewDidScroll:(UIScrollView *)scrollView
{
    if ([scrollView isKindOfClass:[UITableView class]]) {
        ...
    } else {
        // “regular” scroll view
}
```



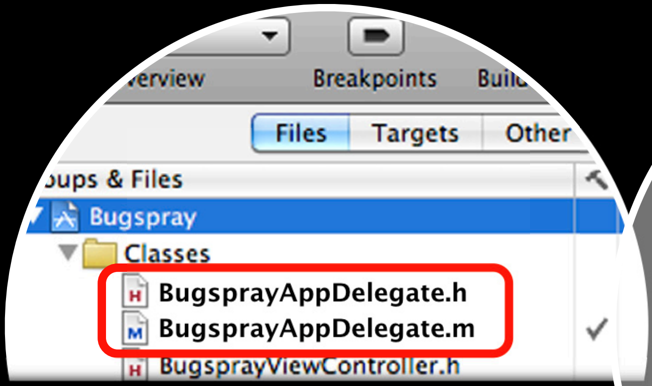
Even on iPad

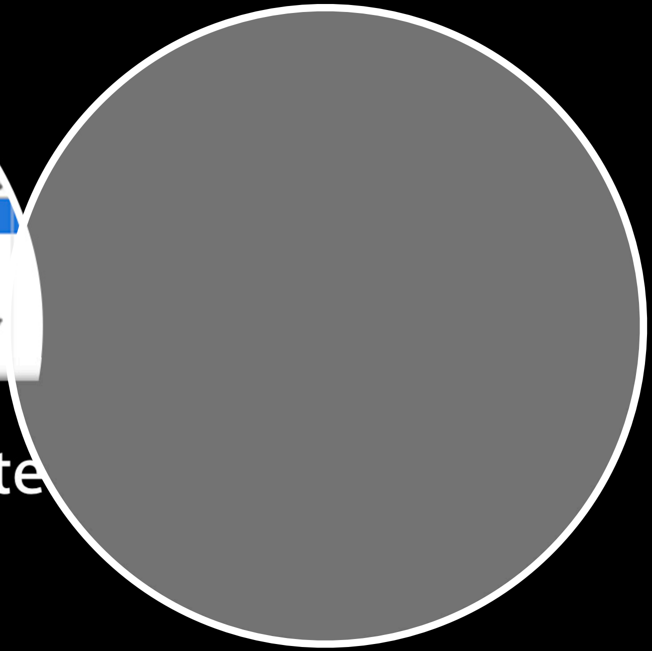
Keep Controller work parceled out



Controllers Play Other Roles

Application Controller



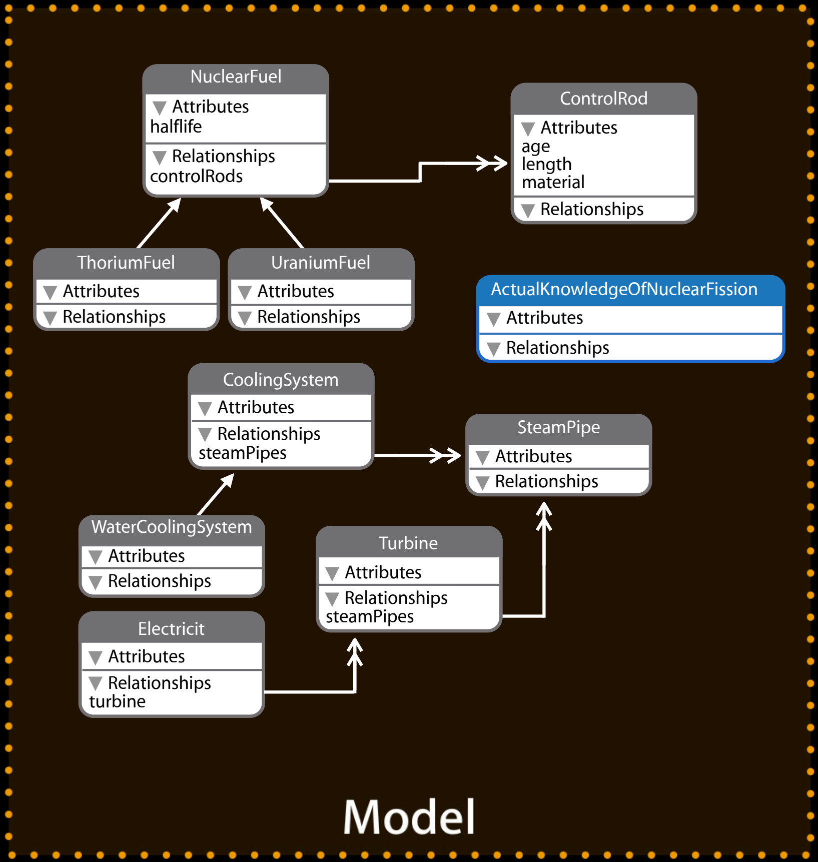


UIApplication delegate

Startup/Shutdown

Root object

Turning over control



CoreData

NSManagedObjectContext

- Saves model to storage
- Write fetches to retrieve objects
- Creating/deleting objects

#8. Decompose Controller Work

The right number of controllers

Special iPhone OS controllers

#9. Take Charge of Your Object Graph

#9. Take Charge of Your Object Graph

Ownership

Lifecycle

Rules

Object Creates Another?

Responsible for Releasing It

Children Don’t Outlive Their Parents

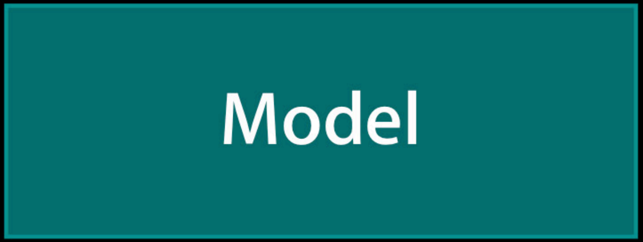
Factory Objects Transfer Ownership

TMTOWTDI

There’s more than one way to do it

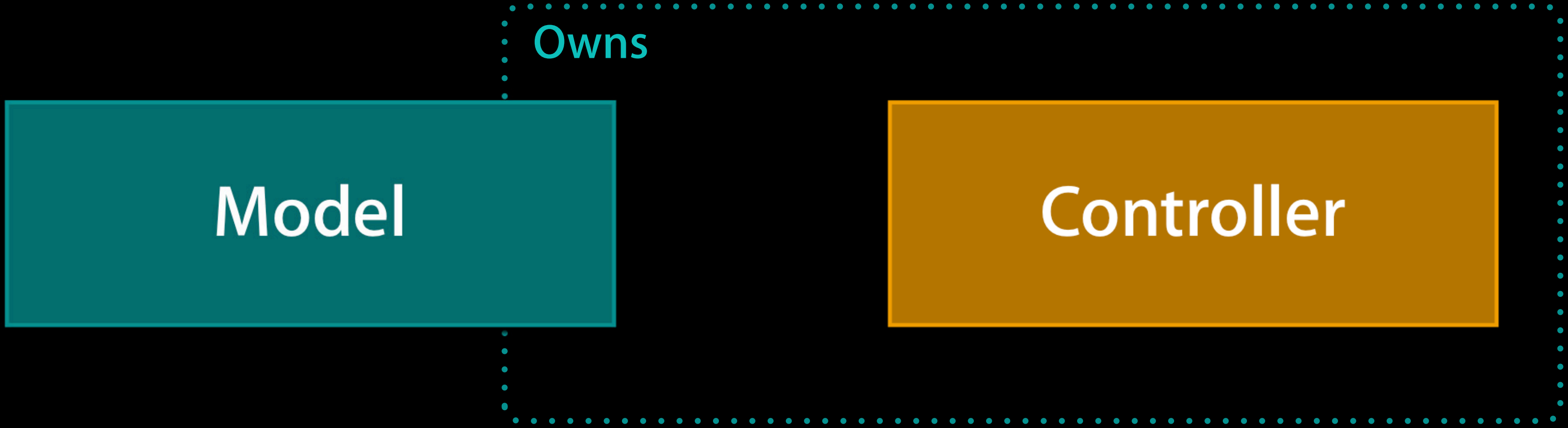
Sample Rule

Models create Controllers



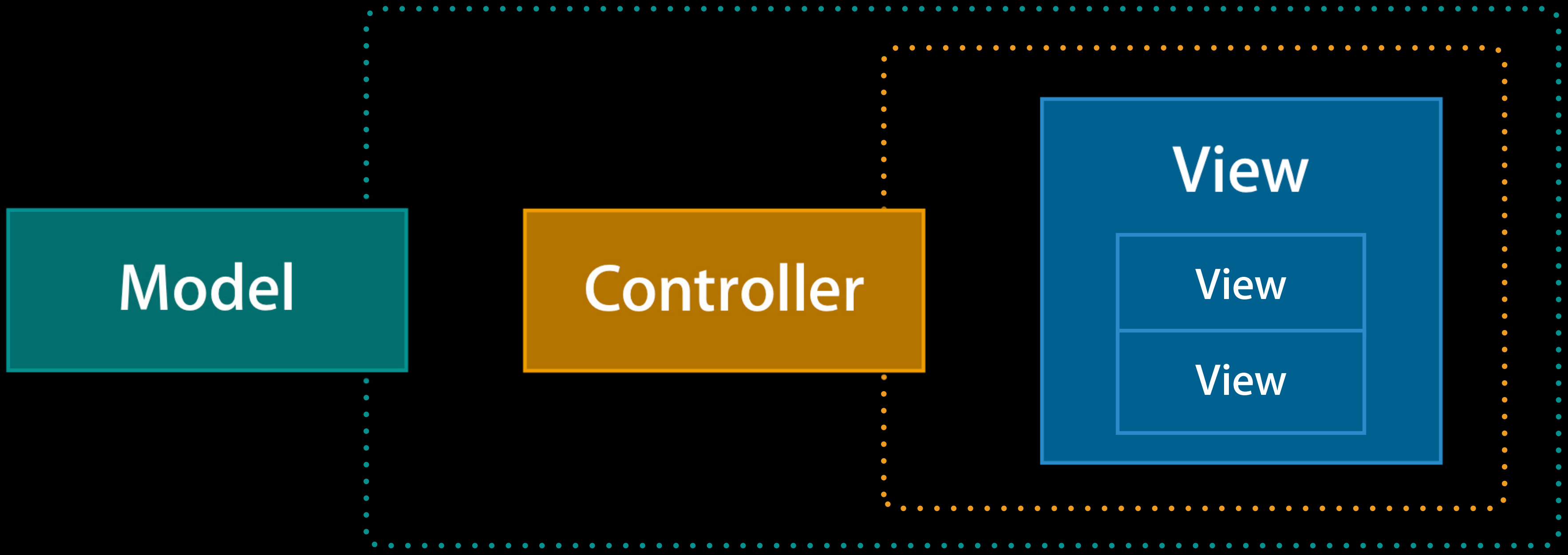
Sample Rule

Models create Controllers



Sample Rule Set

Expanding out

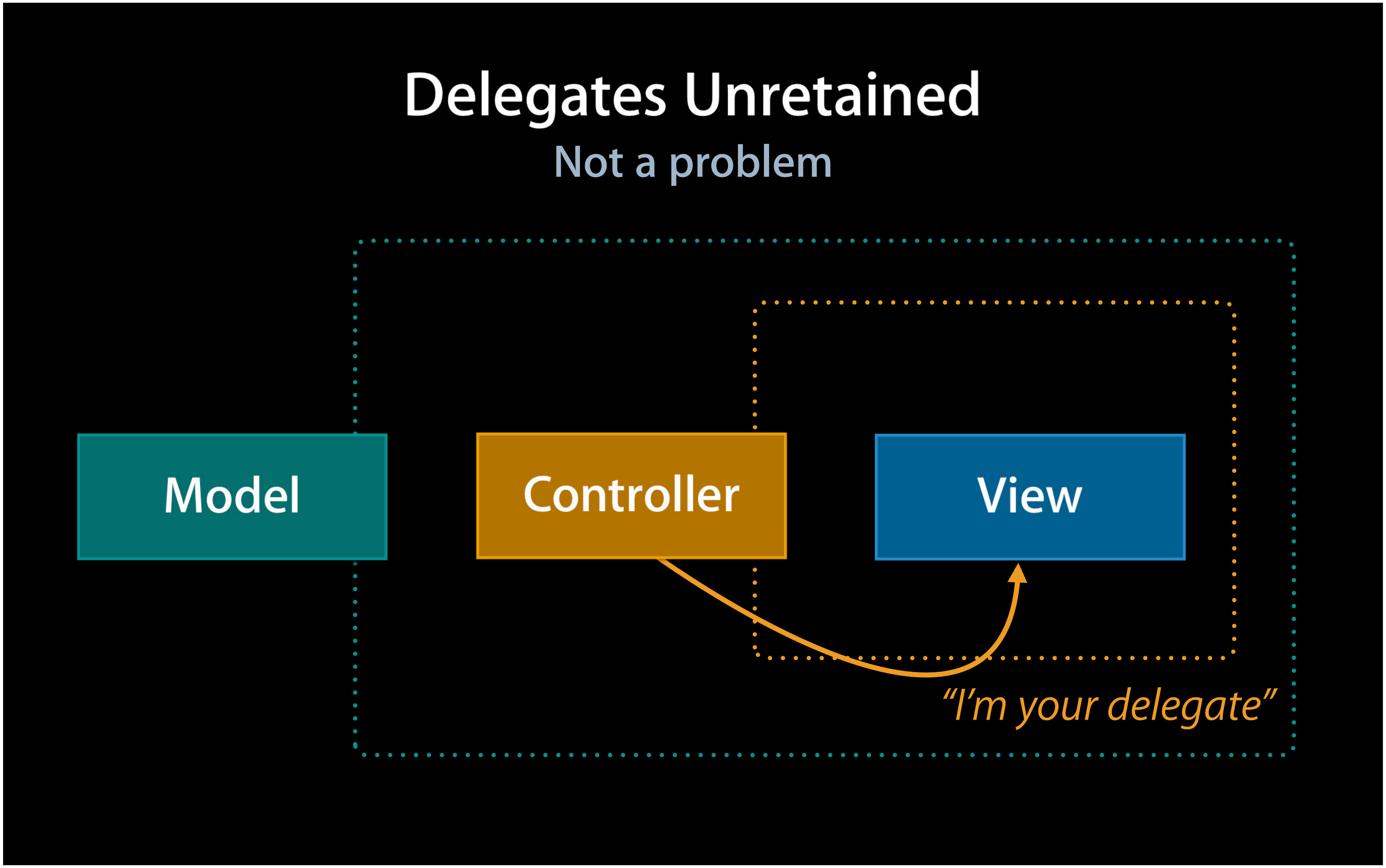


Models never own views

Views never own models or controllers

Delegates Unretained?

Owner in Charge



Delegates Unretained

Not a problem

Got Nibs?

Don’t fight the framework

Got Nibs?

UIViewControllers own their views

Got Nibs?

Split views own master/detail

Got Nibs?

Views own subviews

Got Nibs?

UITextViews *do not* own their text

Rules

#9. Take Charge of Your Object Graph

Ownership

Lifecycle

#10. Coordinate State Changes

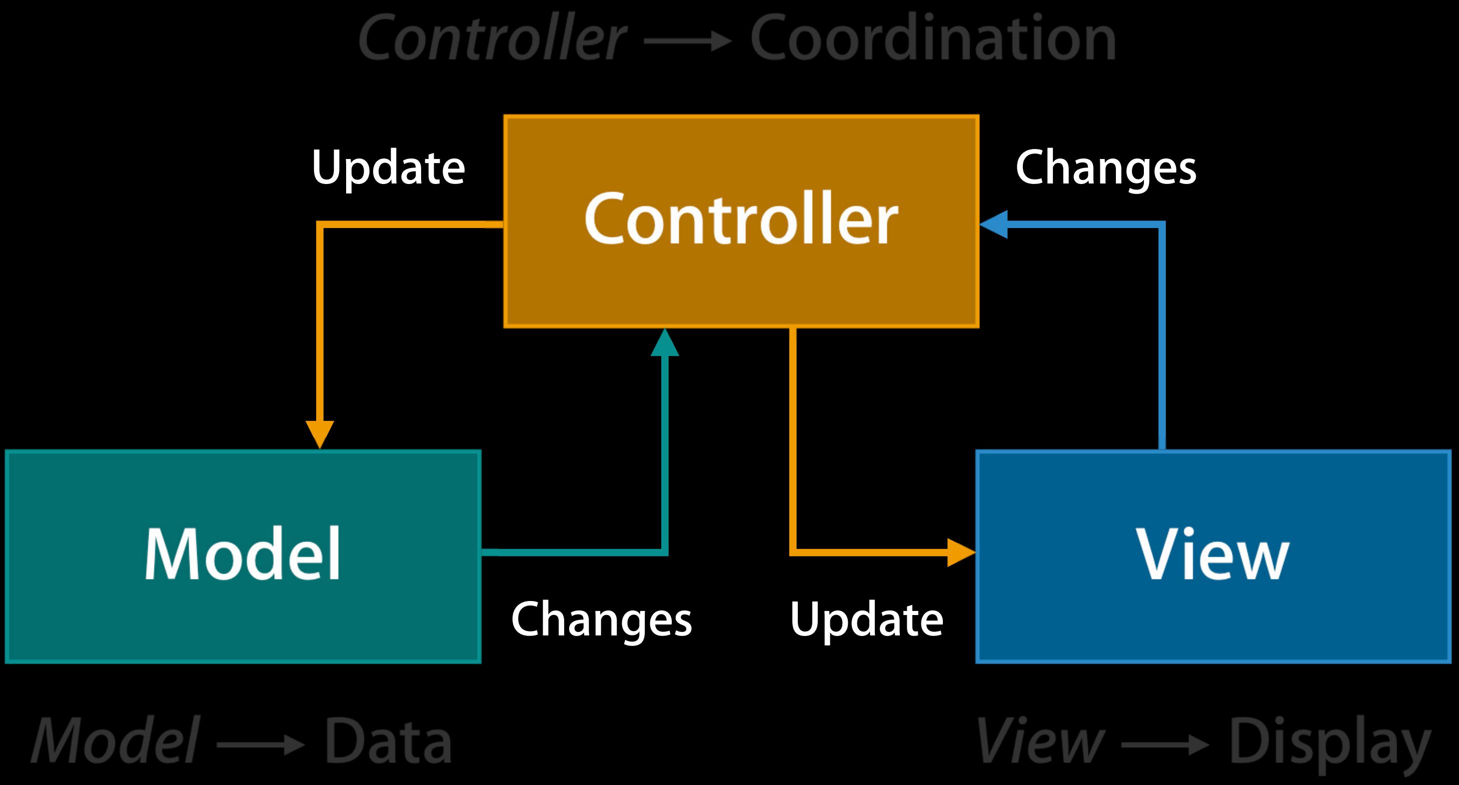
#10. Coordinate State Changes

Updating model after user actions

Updating views after model changes

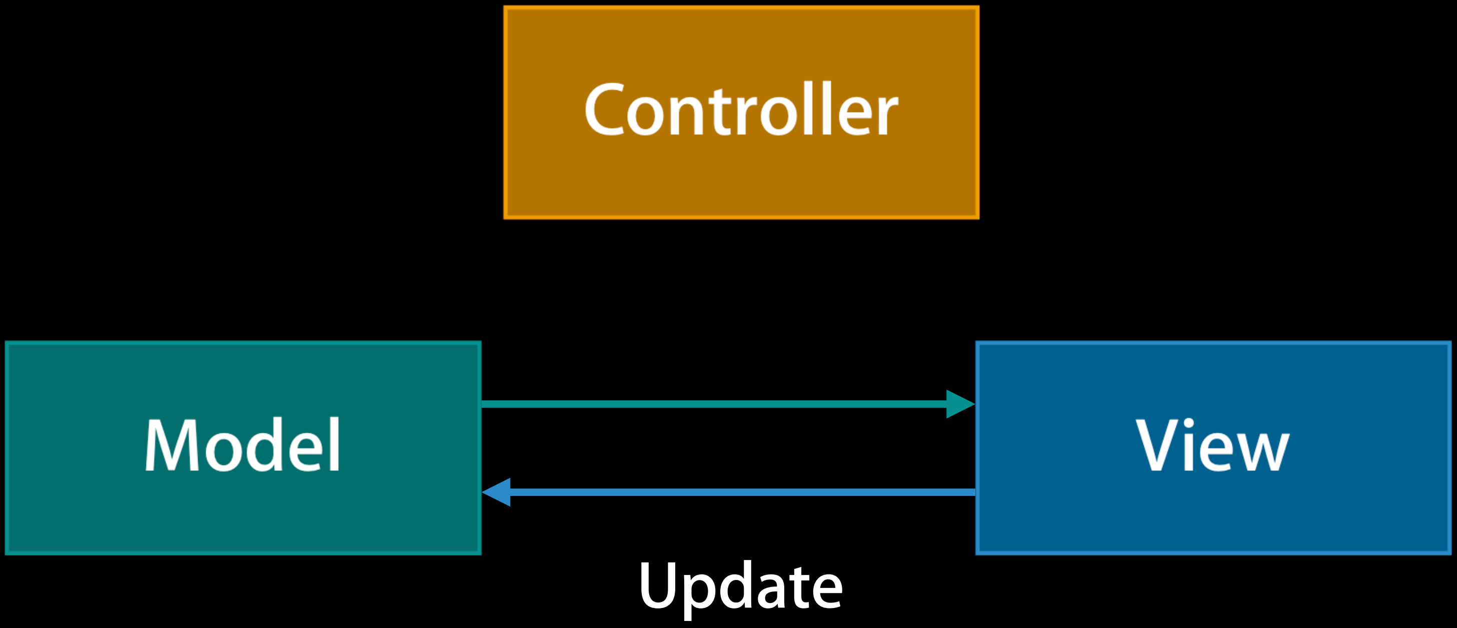
Model-View-Controller

Handling updates

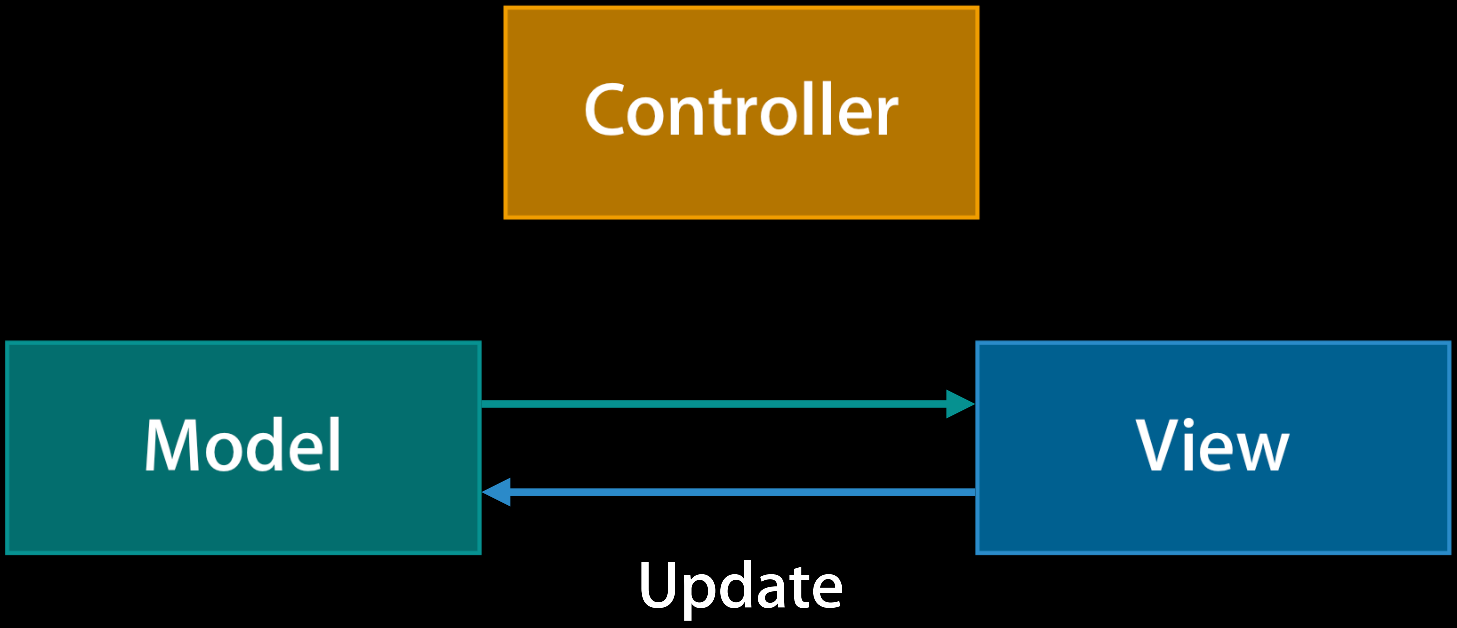


Handling Changes

The wrong way



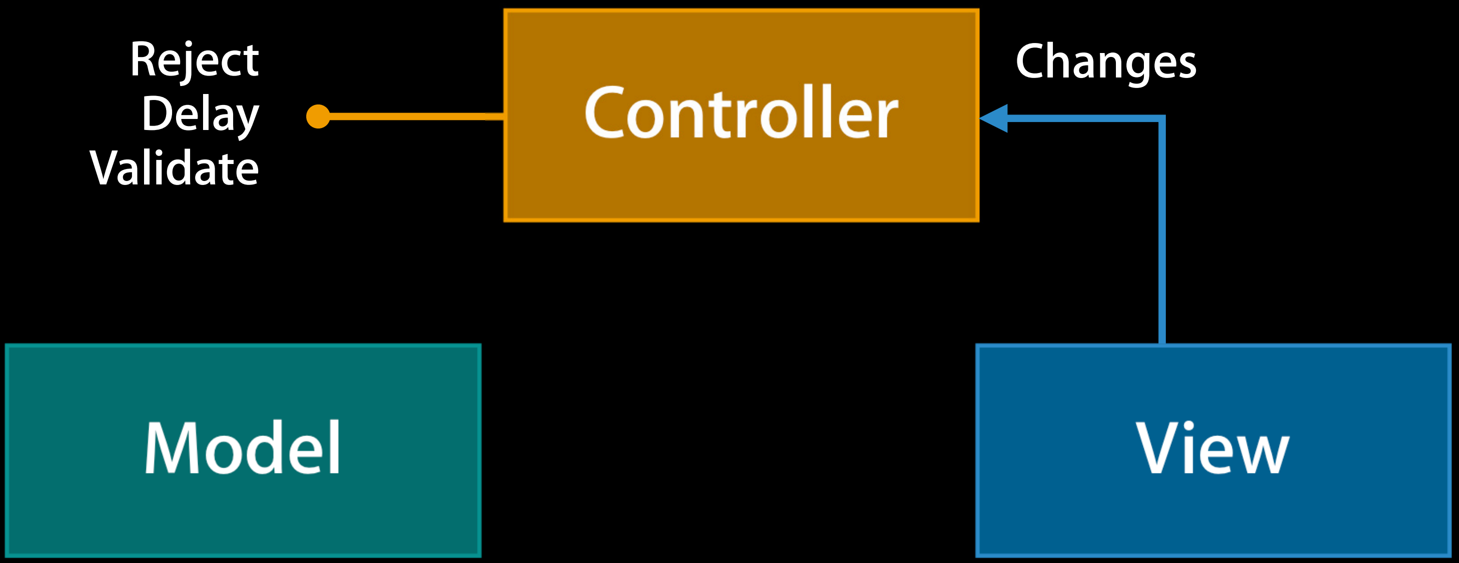
Don’t Cut Out the Controller ✗



Why?

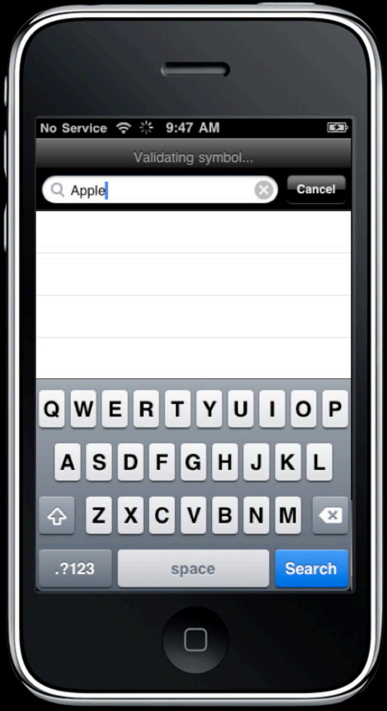
Handling Changes

The right way



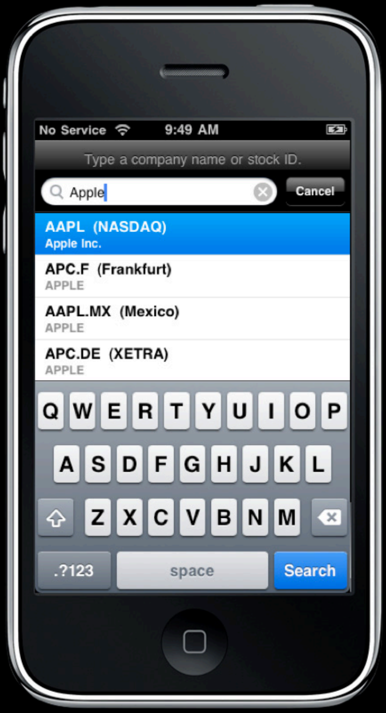
Stocks Application

Network access



Stocks Application

Multiple choice


No Service 9:49 AM

Type a company name or stock ID.

Apple

Cancel

AAPL (NASDAQ)
Apple Inc.

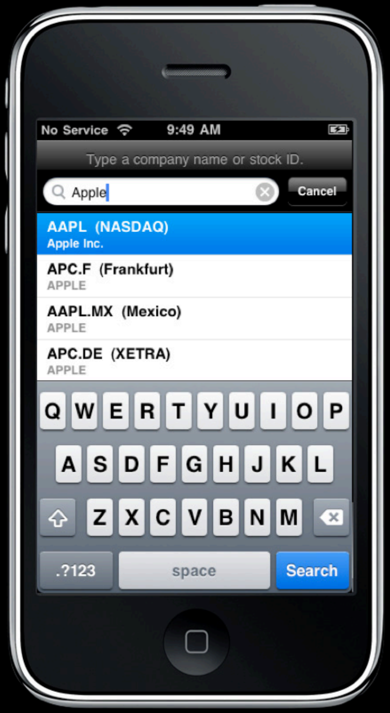
APC.F (Frankfurt)
APPLE

AAPL.MX (Mexico)
APPLE

APC.DE (XETRA)
APPLE

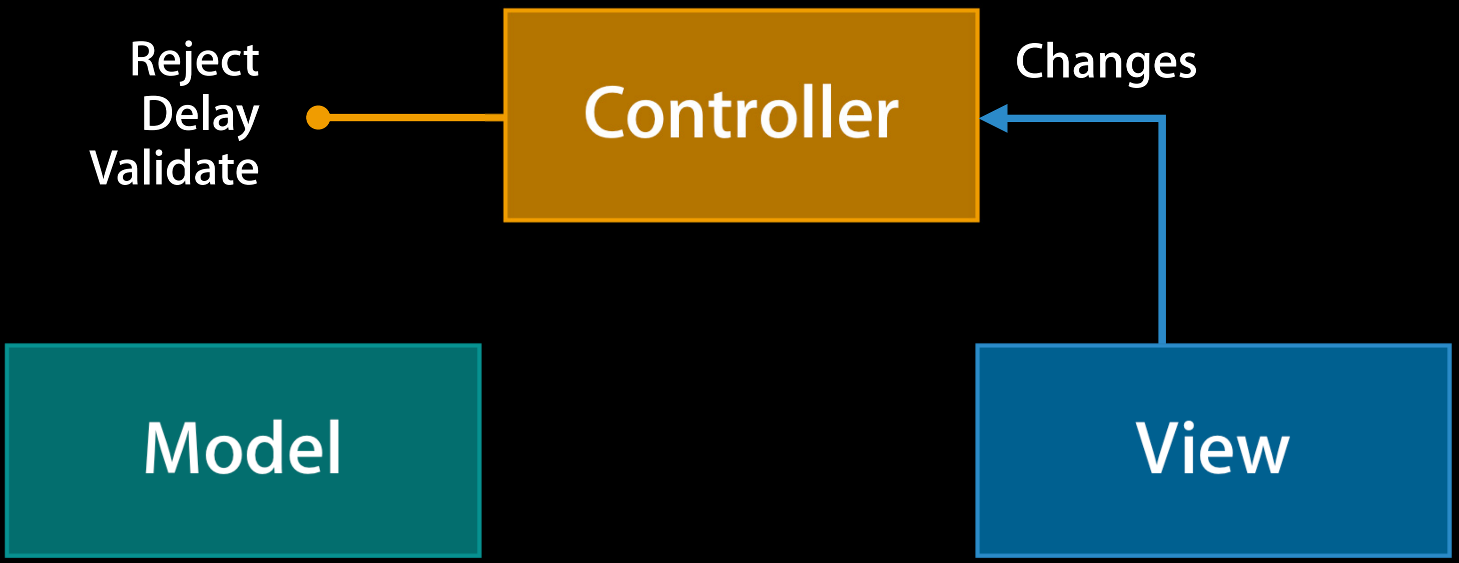
Stocks Application

Commit



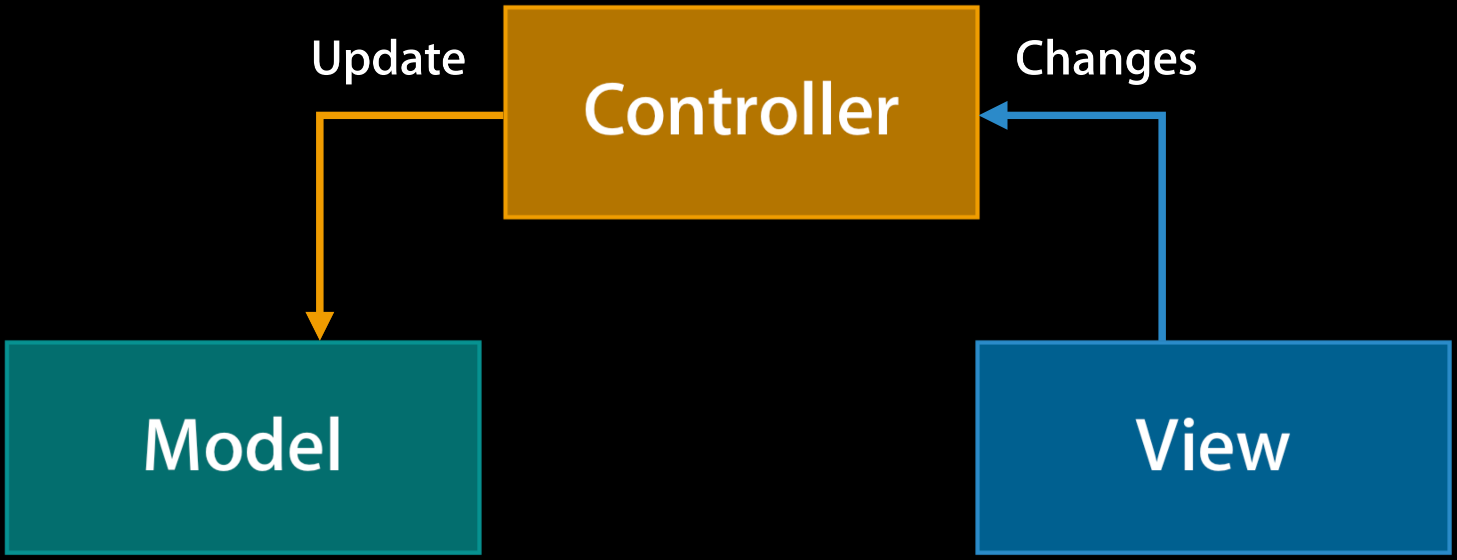
Handling Changes

The right way



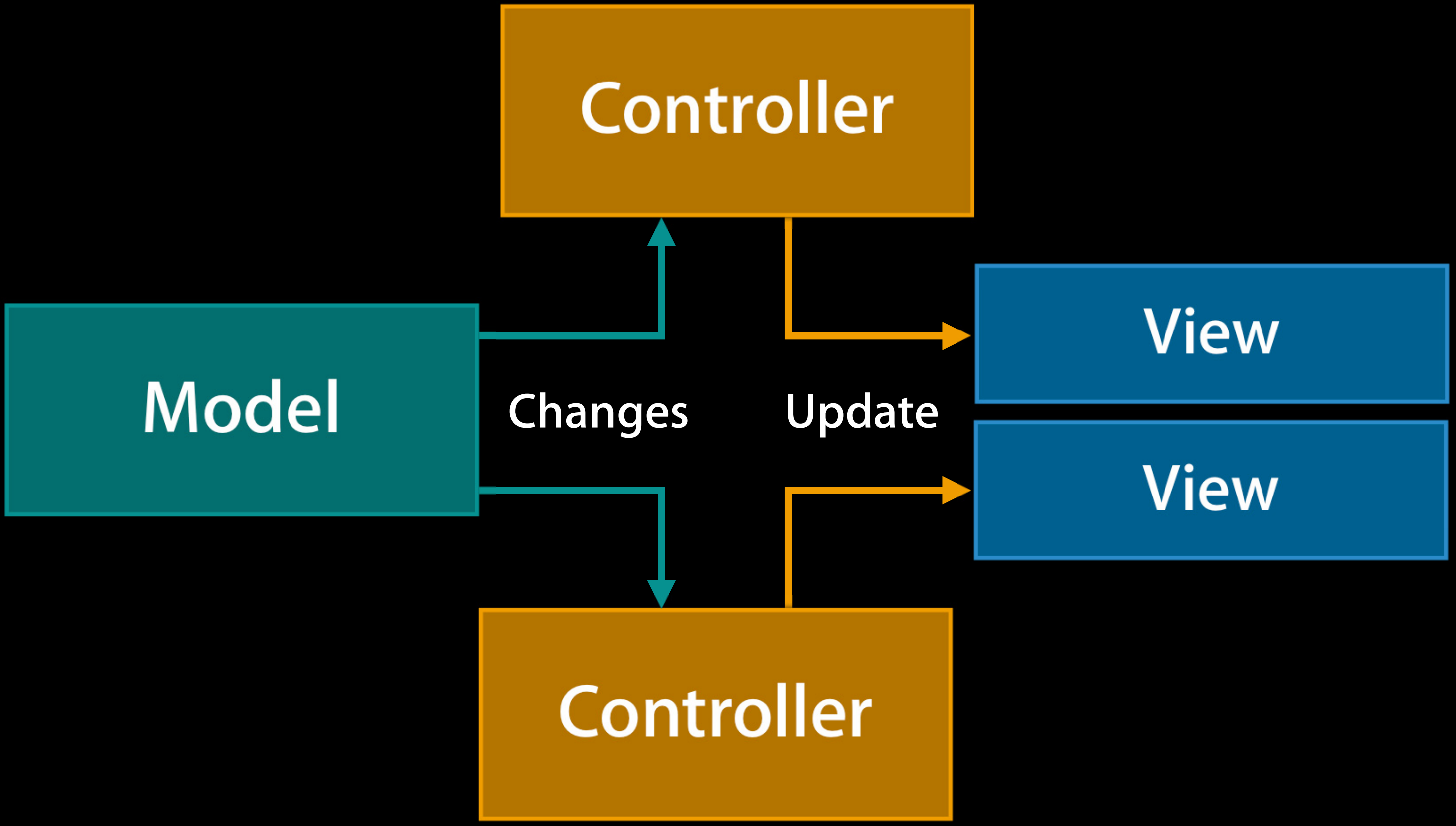
Handling Changes

The right way



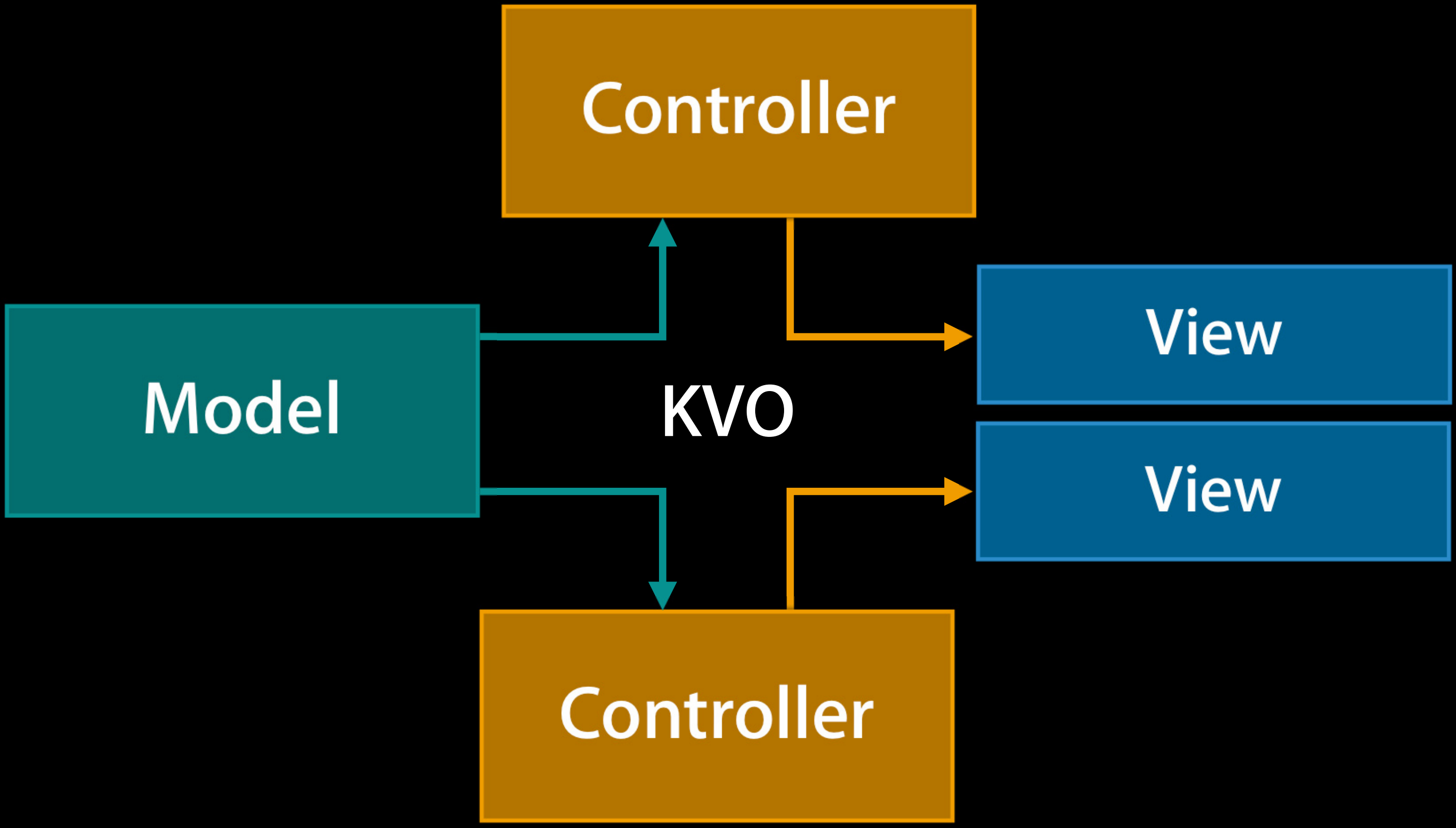
Handling Changes

Updating multiple controllers



Handling Changes

Updating multiple controllers



Key-Value Coding (KVC)

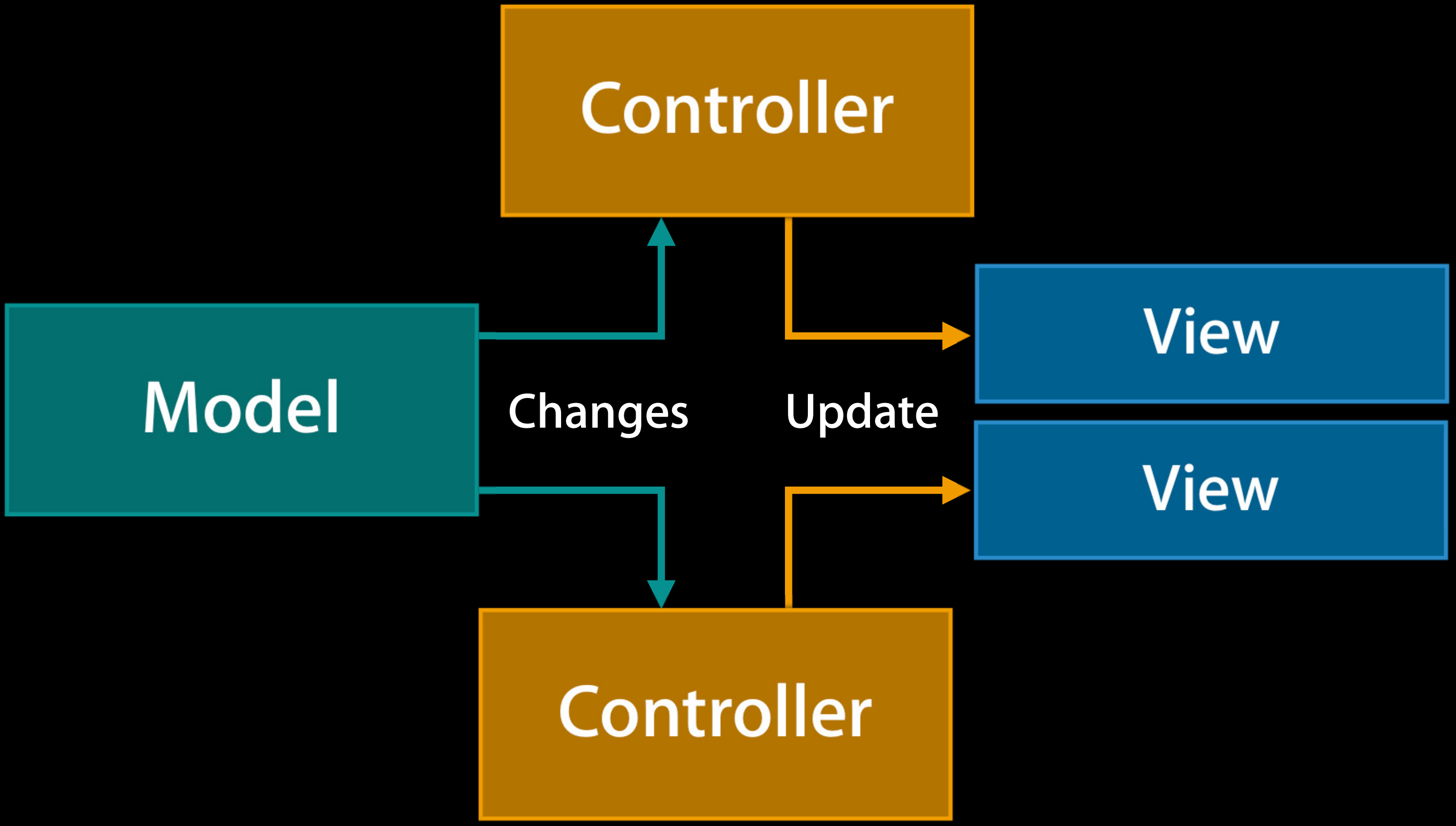
Key-Value Observing (KVO)

```
-getFoo: → -valueForKey:@"foo"

-setFoo: → -setValue:forKey:@"foo"
```

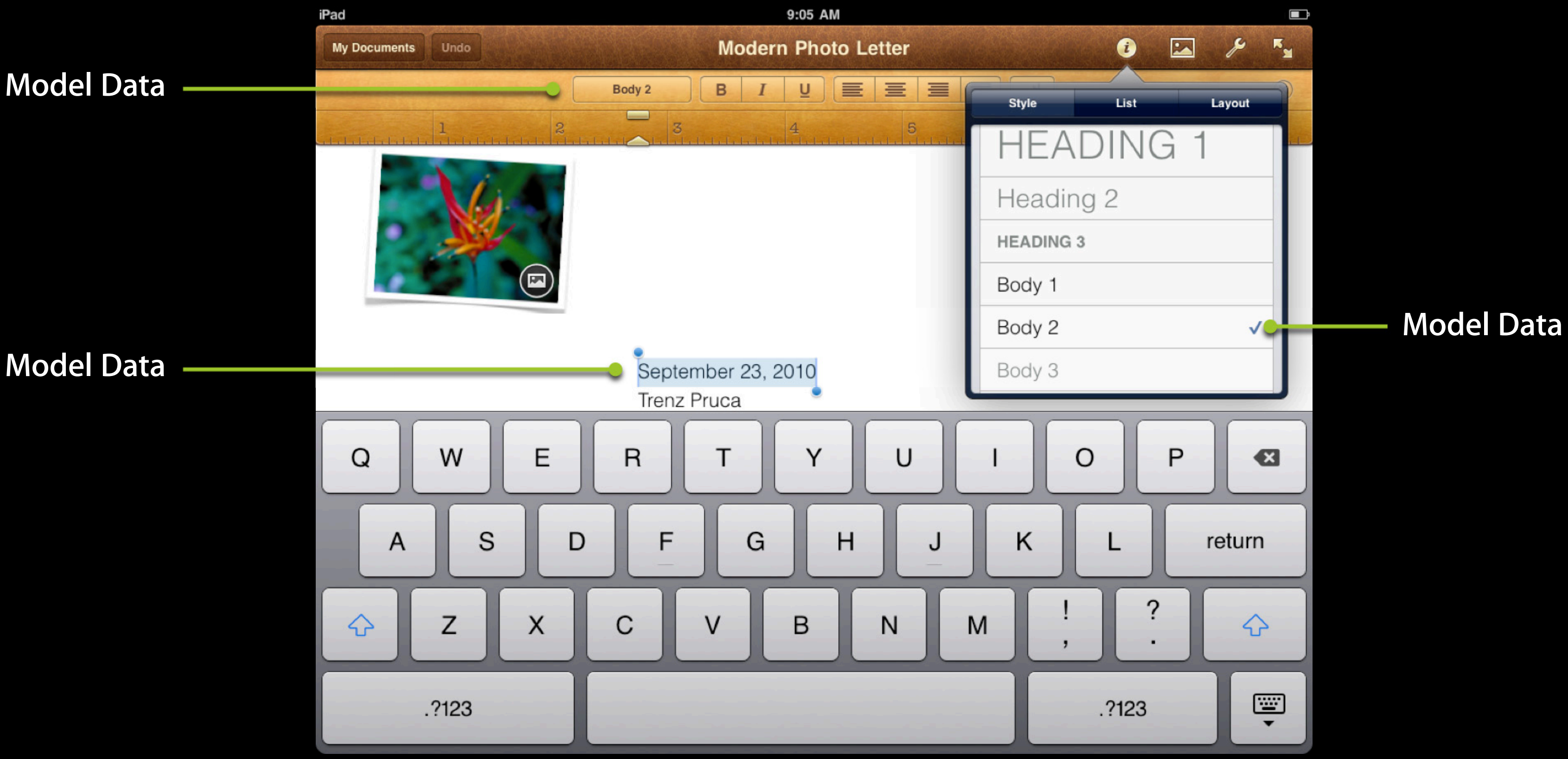
Handling Changes

Updating multiple controllers



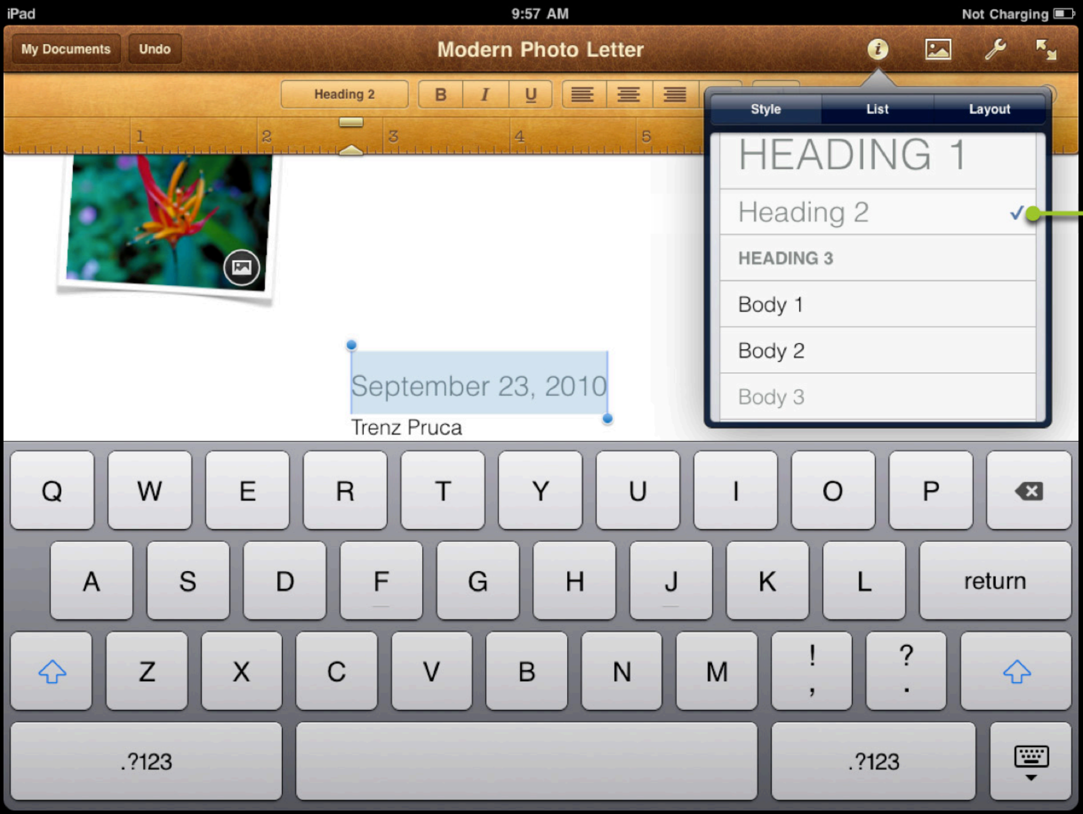
Complex View Updates

Inspectors on iPad



Complex View Updates

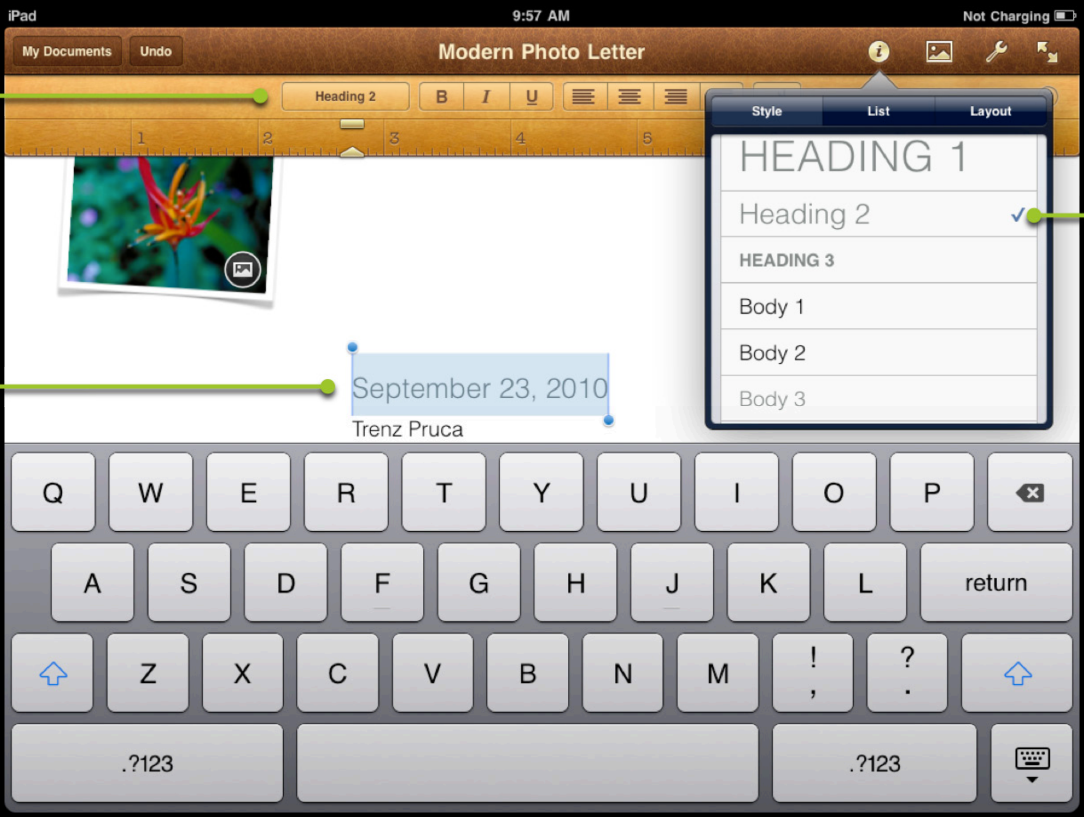
Pages Style Inspector



Change

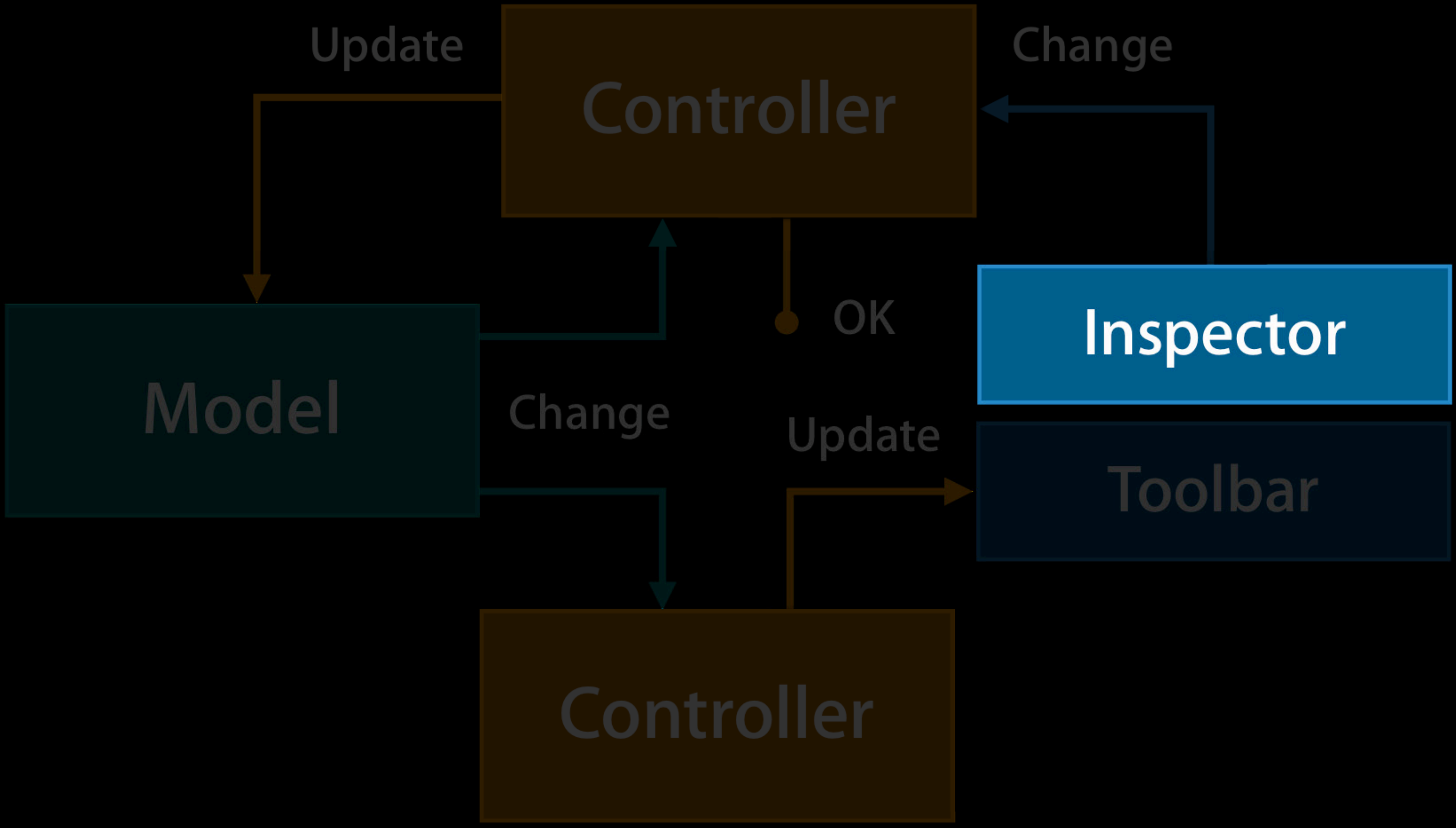
Complex View Updates

Pages Style Inspector



Handling Changes

Updating multiple controllers



MVC Is the Way to Go

#10. Coordinate State Changes

Updating model after user actions

Updating views after model changes

The 10 Best MVC Tips Ever

The 10 Best MVC Tips Ever

#1. Learn MVC for iPhone OS

#2. Use MVC to divide work

#3. Don’t fight the framework

#4. Don’t abuse views

#5. Plan for iPhone and iPad

The 10 Best MVC Tips Ever

#6. Strive for loose coupling

#7. Choose the right data model

#8. Decompose controller work

#9. Take charge of your object graph

#10. Coordinate state changes

Flexible and Easy to Change

Great Apps

Related Sessions

What's New in Cocoa Touch	Mission Tuesday 9:00AM
iPad and iPhone User Interface Design	Mission Tuesday 10:15AM
Designing Apps With Interface Builder	Mission Wednesday 2:00PM
Mastering Core Data	Russian Hill Wednesday 2:00PM
Simplifying Touch Event Handling with Gesture Recognizers	Pacific Heights Wednesday 3:15PM
Advanced Gesture Recognition	Pacific Heights Wednesday 4:30PM
Understanding Foundation	Russian Hill Thursday 9:00AM

Related Sessions

Model-View-Controller for iPhone OS (Repeat)	Russian Hill Thursday 2:00PM
Performance Optimization on iPhone OS	Presidio Thursday 2:00PM
API Design for Cocoa and Cocoa Touch	Marina Thursday 4:30PM
Optimizing Core Data Performance on iPhone OS	Presidio Thursday 4:30PM
iPad and iPhone User Interface Design (Repeat)	Pacific Heights Friday 10:15AM
What's New in Cocoa Touch (Repeat)	Marina Friday 11:30AM

Related Sessions

Model-View-Controller for iPhone OS (Repeat)	Russian Hill Thursday 2:00PM

WWDC10