Design Patterns

Session 102

Jeremy Foo
API OCD ROFLOL

Philosophy

The bigger picture

Hand held Computer

Personal

(Constant) Companion

Relatively Small

Fairly Advanced

Delayed Production Push

Users Expect Quality

Attention to Detail

Do one thing, do it well.

Provide “Of course!” moments

Delight the User

Performance matters

Design Patterns

You will use it more than you think

Rules

Design Patterns

What is it

- Object Orientated Design
- Reusable Patterns
- “Blueprint” to build systems that solve a recurring problem

Design Patterns

How is it good

- Best practices when designing applications
- Understand a pattern, know how to use it everywhere
- Helps in writing readable code
- Decoupling and class isolation
- Design by contract

Design Patterns

How is it good

- Apple uses it religiously
- As well as good third party component developers

Common Design Patterns

So many

Singleton

What is it

- Only 1 instance of the class
- Provides global access

Singleton

Where would you find it

- Any Apple System Managers
    - NSFileManager
    - NSUserDefaults
    - Application Delegate
    - UIApplication

Protocols

What is it

- You know it as @protocol
- Language level feature
- Instances of Adapter Pattern
- Helps in decoupling

Protocols

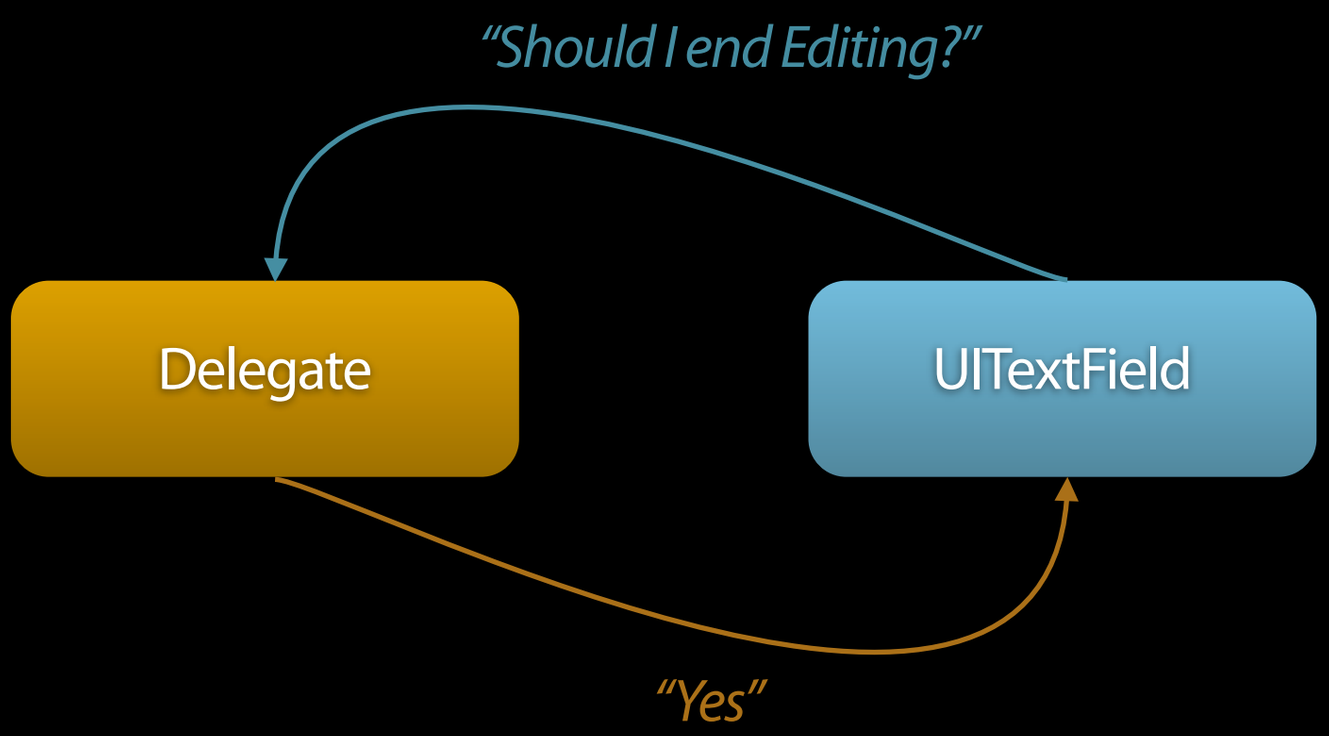
Where would you find it

- Almost everything that supports a delegate
  - UITableView
  - NSCoding
  - Application Delegates
  - ...

Delegate

What is it

- A weak reference to another object (the delegate)
- Host object sends messages when it requires input for a task
- Delegates usually conform to a protocol
- Helps in decoupling



Delegate

Where would you find it

- UITableView
    - UITableViewDataSource
    - UITableViewDelegate
- UIApplication Application Delegate

Observation

What is it

- Request to be notified when something changes
- Listen for state changes

Observation

Where would you find it

- Key-Value observation
- Notifications

Target-Action

What is it

- Specify an object, send it a message
- Use SEL

Target-Action

Where would you find it

- Executing instructions when observing changes
- Delayed execution of messages
- Interface Builder target actions

Composite

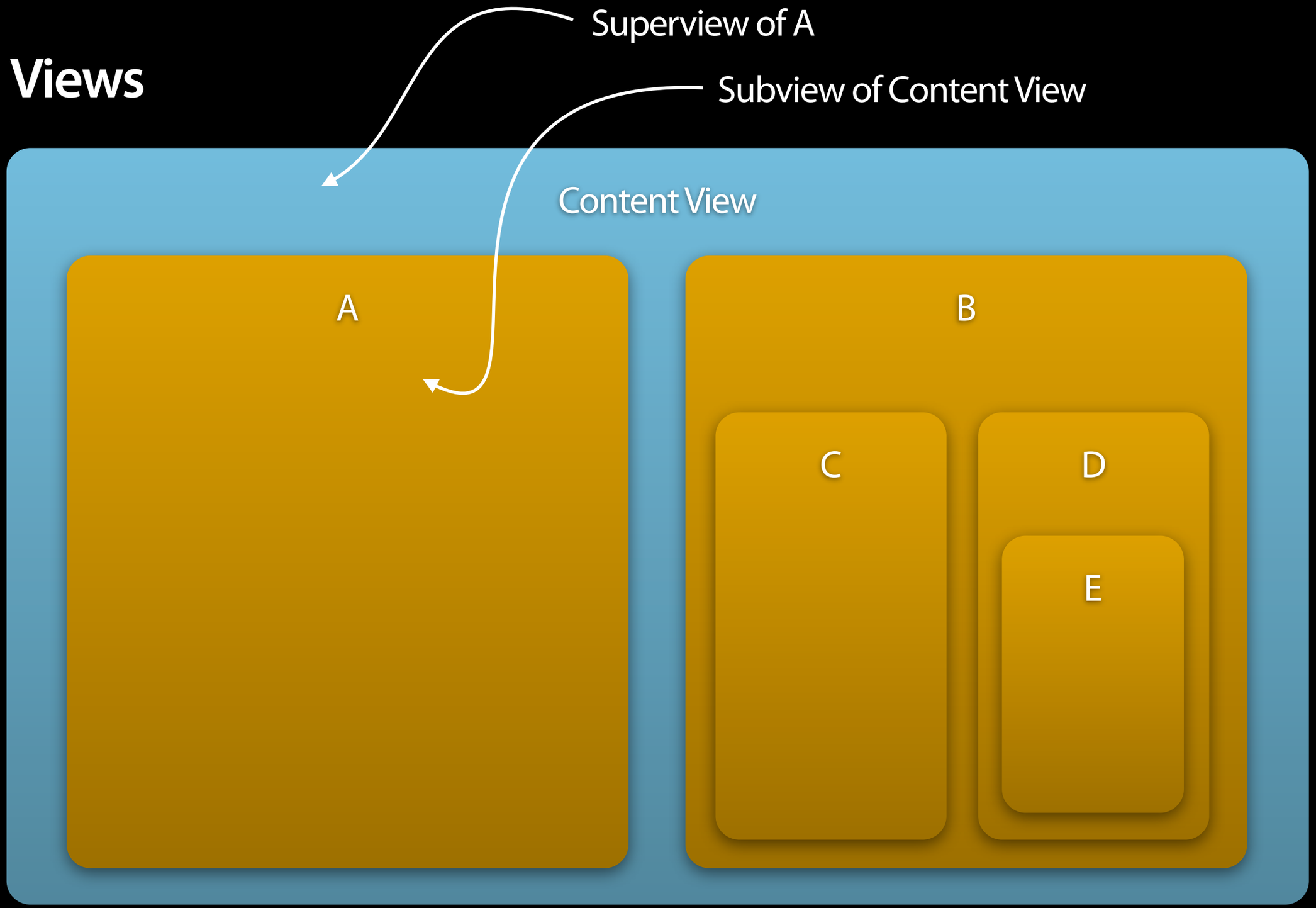
What is it

- Composes related objects into tree structures
- Represent part-whole hierarchies
- Also good alternative to subclassing

Composite

Where would you find it

- Views



Responder Chain

What is it

- A chain of objects that can respond to message
- Pass along message if it doesn’t handle the message

Responder Chain

Where would you find it

- Touch events generated on surface passed along
- Showing UIKeyboard by calling `–becomeFirstResponder`

MVC

One Design Pattern to Bind Them All

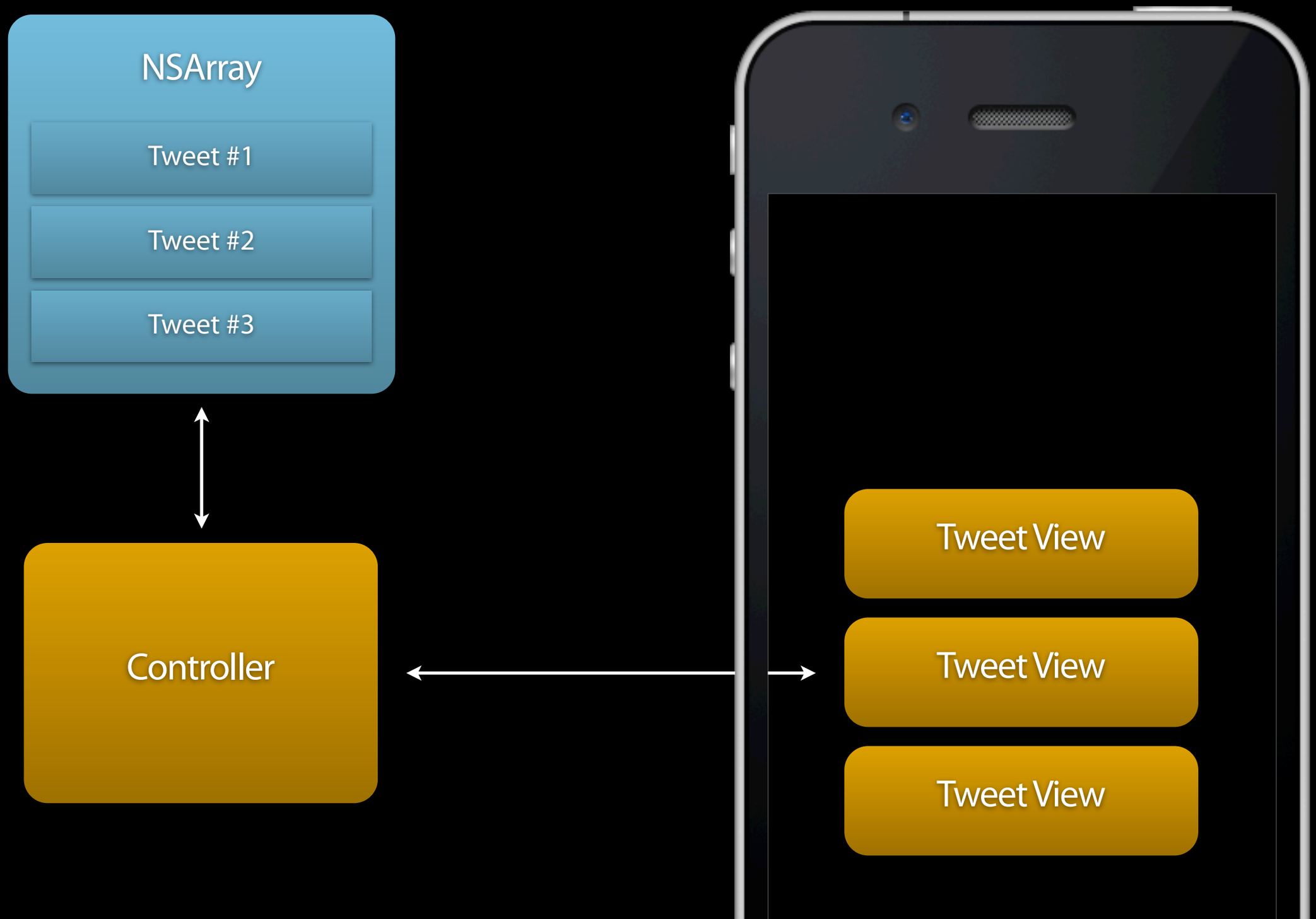
MVC

Overview

- The base pattern that is used for iOS
- Promotes development of decoupled components
- “Areas of responsibility”
    - Data
    - Presentation
    - Coordination

MVC

All together now



MVC

The Model

- The underlying data
- Can be domain specific implementations
- But should primarily deal with data

MVC

The View

- Responsible for how things are shown to the user
- More drawing logic than actual business logic

MVC

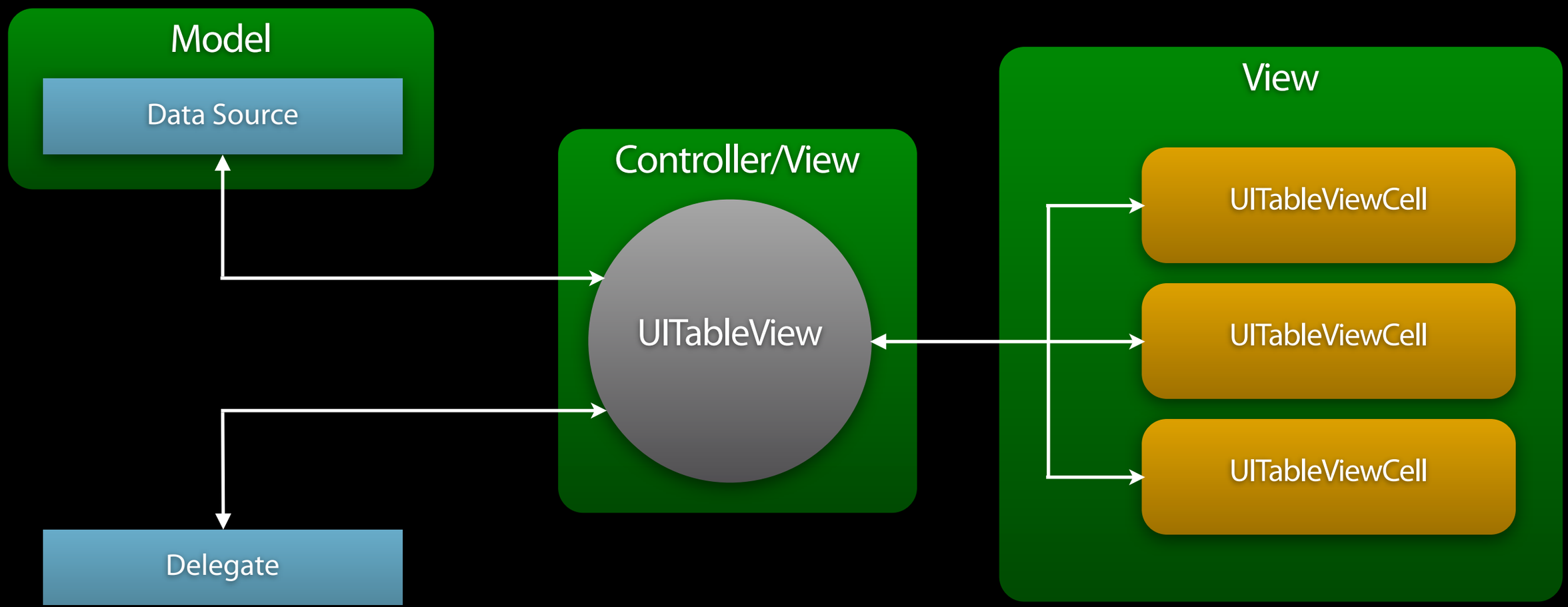
The Controller

- The coordinator of everything
- Takes data and updates the view
- Takes changes in the view and updated the model
- Handles business logic

MVC

UITableView

- Acts like a controller
- Takes data from delegate/datasource and updates view



Project Exercise

If you fail to plan, you plan to fail

Project

Basic Features

- Twitter drafts drawer
- Store drafts for refinement before posting to Twitter
- Track tweets posted
- Edit drafts
- Show personal timeline

Project

Advanced Features (Optional)

- Tweet pictures
- Set location
- Calculate characters remaining
- Create draft from personal timeline
- Cache personal timeline
- Cache retweets
- Store drafts in .json files accessible using iTunes File Sharing

View Controllers

Design guidelines

- One view controller does one thing and does it well
- It should not have knowledge about other view controllers
- Central repository store can use NSUserDefaults

Data Model

Design Guidelines

- Forget about implementation, think about services
  - Figure out what data needs to be stored
  - Figure out what actions are logically linked
- Make sure it does one thing, and one thing well