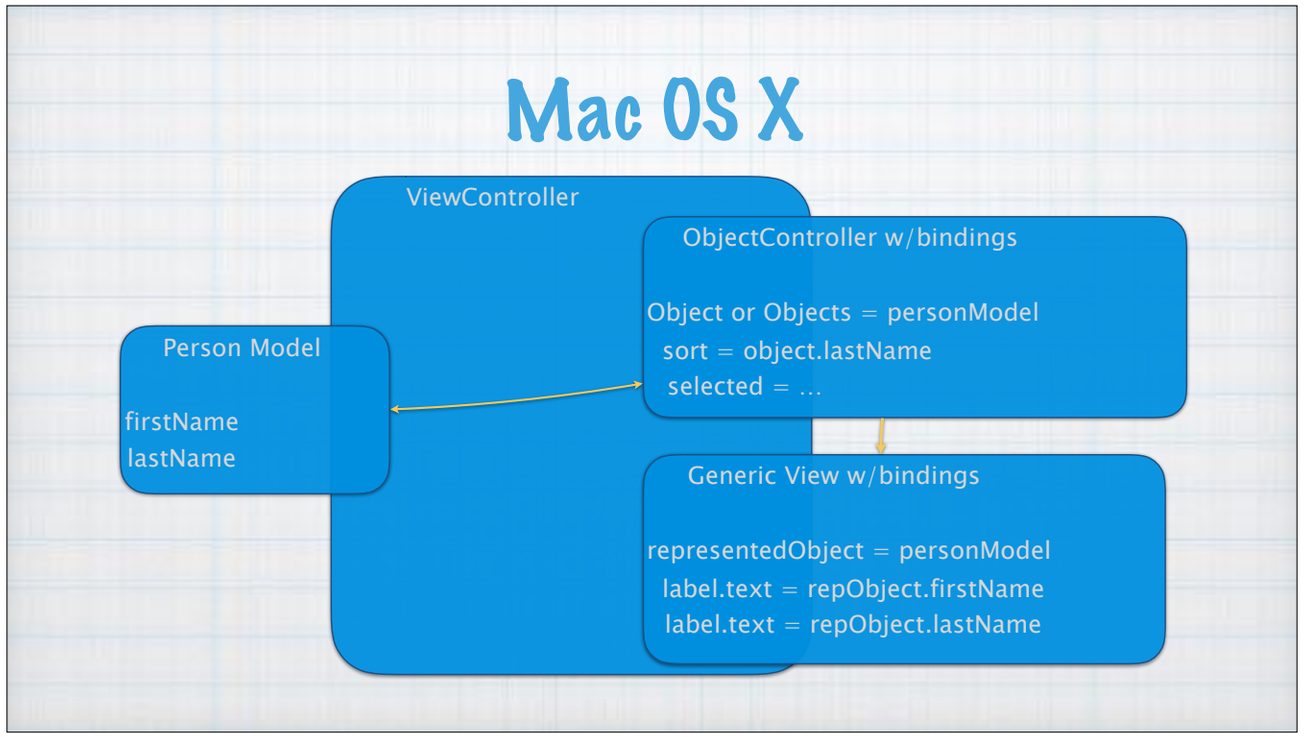
MyVC

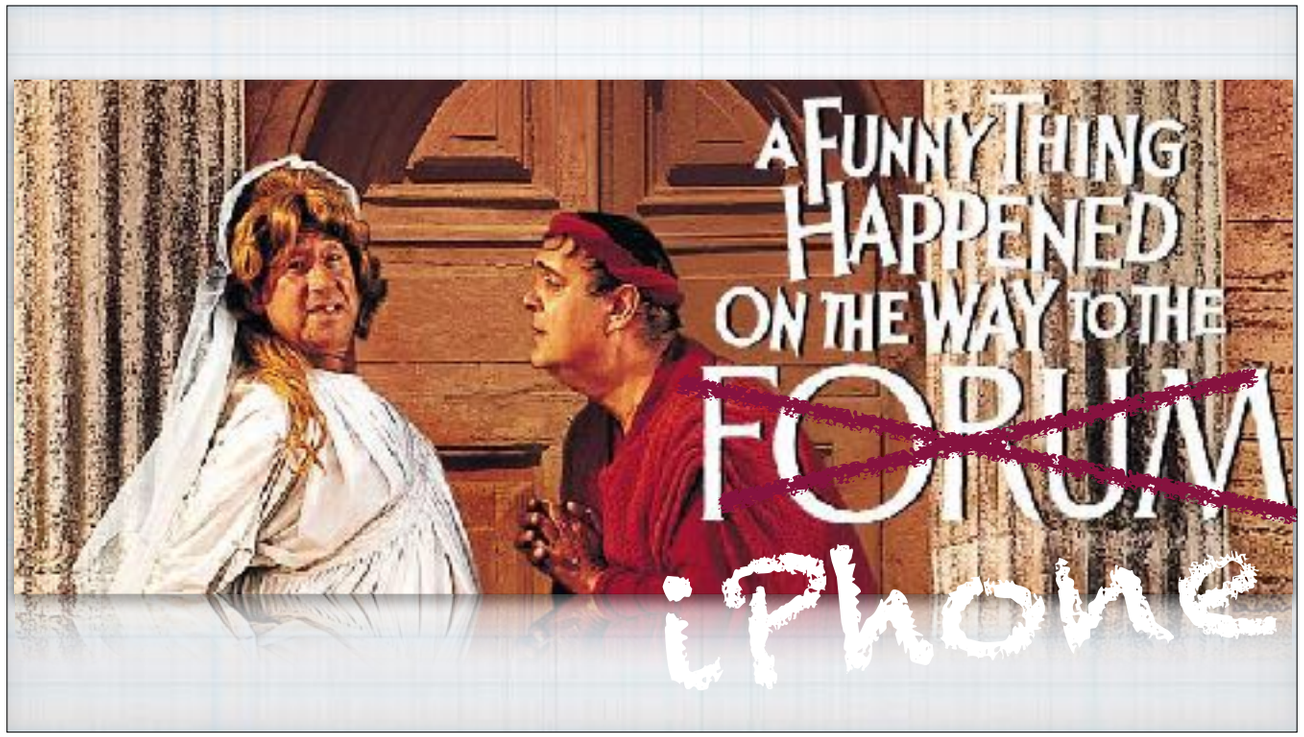
An opinion on (Model->Views) + controllers as a pattern

Demo App Features

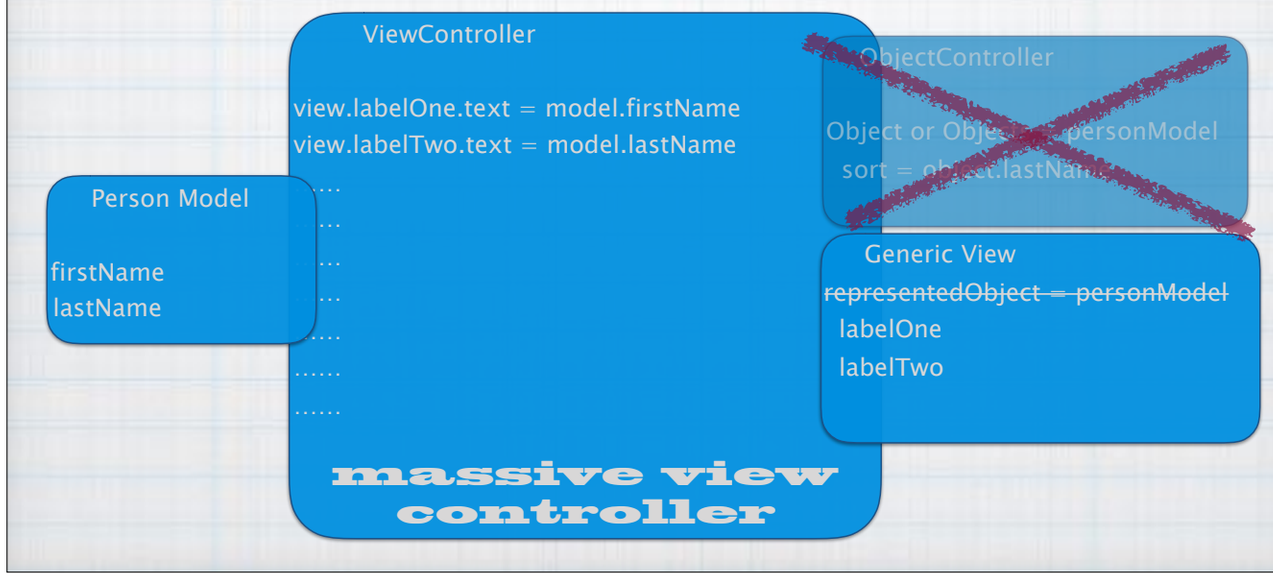
Potential for Multiple views for a single model - image & listing & ?

Potential for Multiple models -

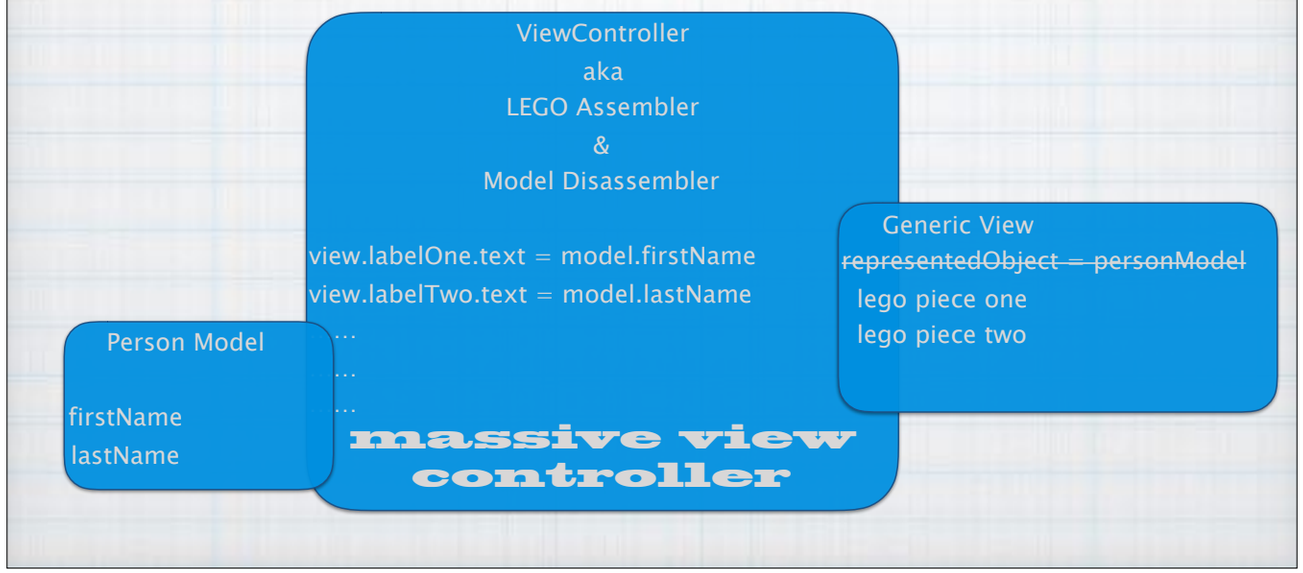

Mac OS X
ViewController
ObjectController w/bindings
Object or Objects = personModel
sort = object.lastName
selected = …
Person Model
firstName
lastName
Generic View w/bindings
representedObject = personModel
label.text = repObject.firstName
label.text = repObject.lastName


A FUNNY THING
HAPPENED
ON THE WAY TO THE
FORUM
iPhone

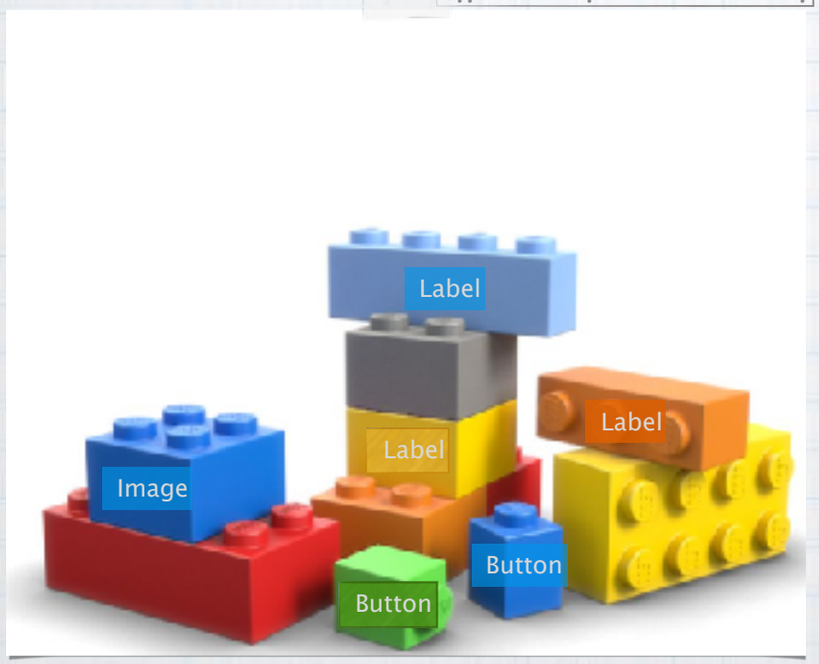
iOS has no Bindings ¯_(ツ)_/¯



iOS UIViews are treated like LEGO



Apple iOS Example Code craftsmanship



LEGO Views

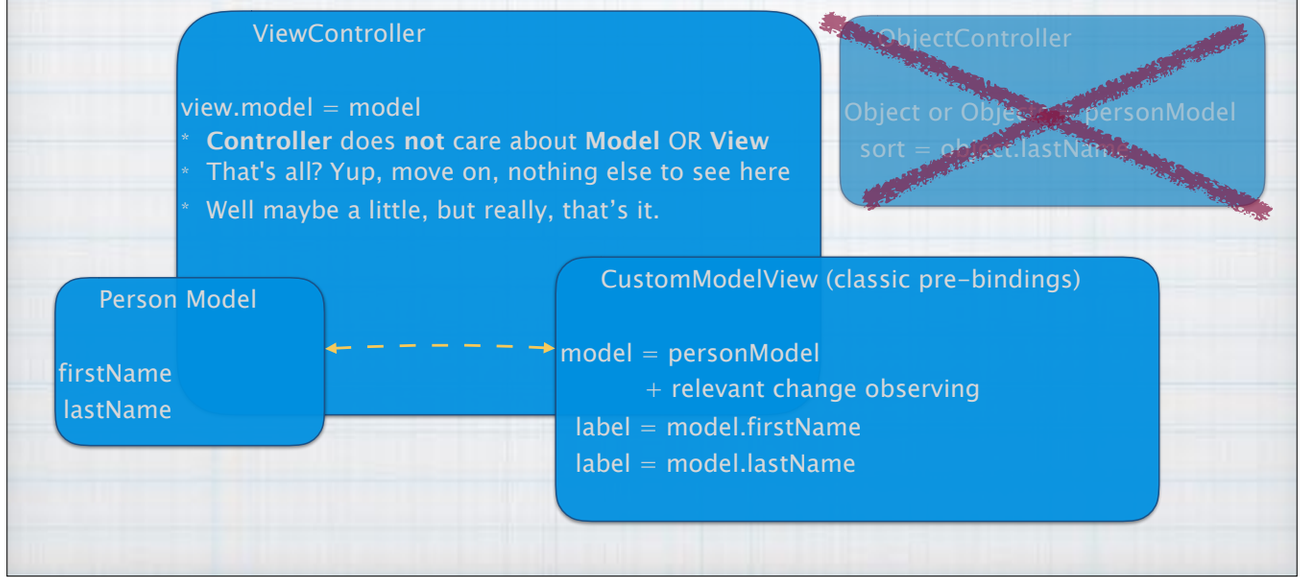
* Constantly assembling the same pieces
* Not very reusable at the app level
* No reusable craftsmanship
* Difficult consistency & theming
* Constantly re-applying the same characteristics throughout app
  font, size, color, …

Apple Says ...

"If you don't use the bindings mechanism, you have to subclass an existing view class to add the ability to observe change notifications posted by a model object."

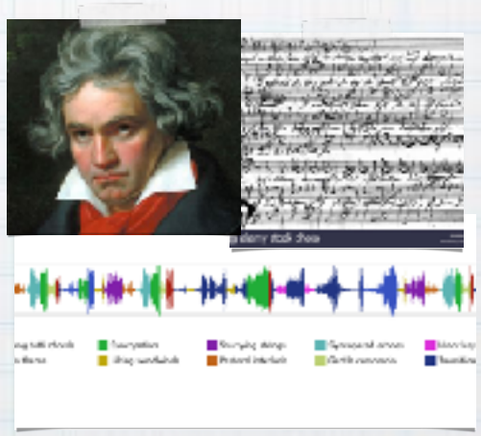
-Apple OS X Model-View-Controller Concept Documentation

iOS should be like OS X w/o bindings



Your app is all about your customModelViews

- It is the ONLY way to perceive the model
- aka how the user “views” the model
- Your Views *should* know about your Models
- CustomViews are more reusable not less
- Your app is more consistent
- Many different types of views of a model
- Views are Compose-able & hierarchical



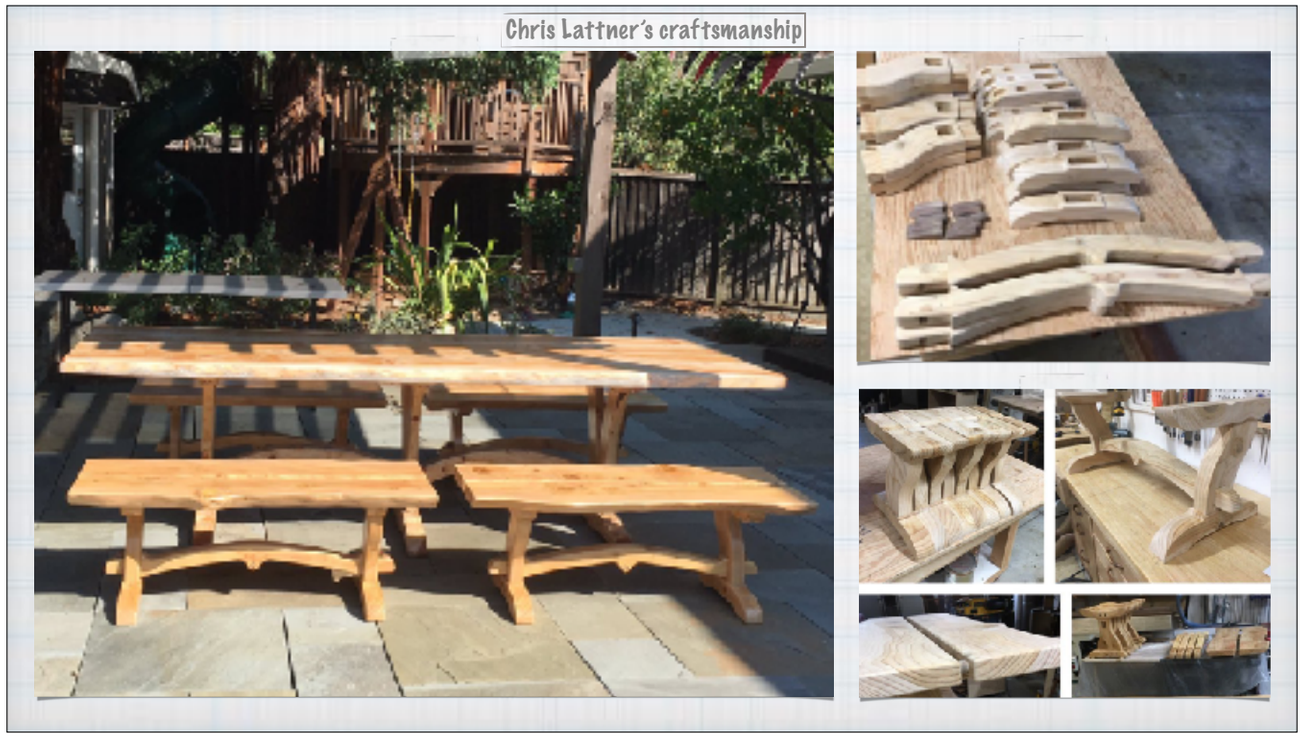
Not to be confused with the MVVM View Model pattern. UIKit Views are not code less templates. They are designed to be subclassed and customized.

Oops We misnamed ViewController

* ViewControllers do not really control views!
* They link models to views and modify models.
* Views do not modify models only the view appearance.
* Views can modify the view! Such as Pan, zoom, scroll…
* View modifying itself should not effect the model.

Confused by 2 Types of Controls?

* 1) There are controls that modify models
  * Stepper, slider modifying a model value
  * TextField modifying a model property
  * These controls act in a ViewController to modify model
* 2) There are controls that modify views
  * Scroll view scrolling a list of contacts does not modify the contacts and is “controlled” by the ScrollView itself. Not a ViewController.
  * Same for Pan and Zoom of view content by a view.
* Whether a control is 1 or 2 depends on it’s use not it’s name.



Custom views are like crafted reusable parts specific to your uses/app(s).
Look at those reusable yet beautiful bench and table parts. The same part is used for both the benches and the table, yet unlike lego, the end result is an easy to assemble table and bench with a beautiful crafted look made from reusable parts.
That is MVC custom views!

Really?

Live Demo of MyVC iOS app

Resources

* https://developer.apple.com/library/content/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html
  The above page is wrong and misleading for iOS!
  It's explanations are based on having bindings. iOS does not.

* Original MVC http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html

* https://github.com/apple/swift-evolution/blob/master/proposals/0161-key-paths.md

* https://developer.apple.com/videos/play/wwdc2017/212/ What's new in foundation.