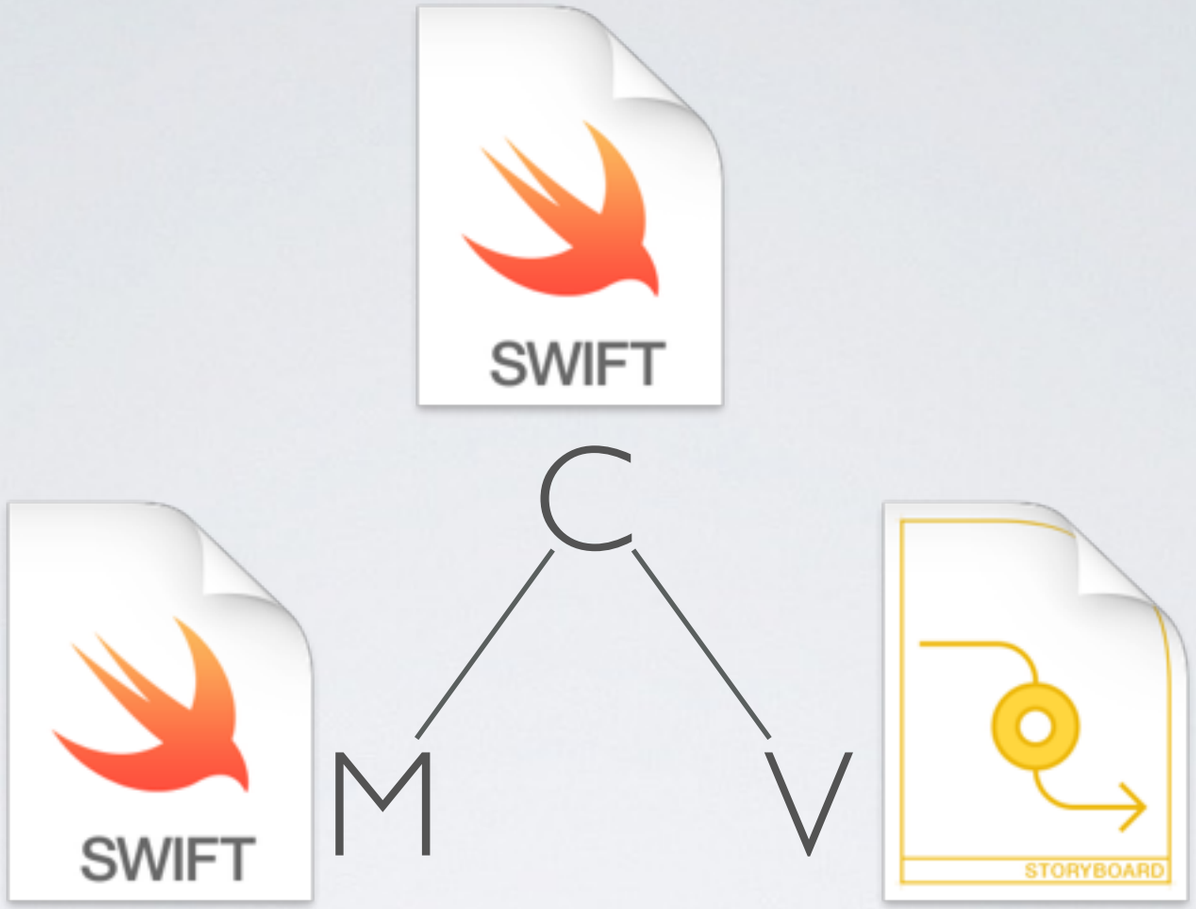


MODEL-VIEW-CONTROLLER AND TABLEVIEWS

Nick Chen
Fall 2015
talentspark.io

PREVIOUSLY

GitHubDemo 〉 iPhone 6 | Running GitHubDemo on iPhone 6

GitHubDemo 〉 GitHubDemo 〉 ViewController.swift 〉 No Selection

```swift
//
//  ViewController.swift
//  GitHubDemo
//
//  Created by Nick Chen on 7/28/15.
//  Copyright © 2015 TalentSpark. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var hello: UILabel!
    @IBOutlet weak var button: UIButton!
    @IBOutlet weak var textField: UITextField!


    @IBAction func buttonClicked(sender: UIButton) {
        hello.text = "Hello, Clicker!"
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from
            a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


}
```

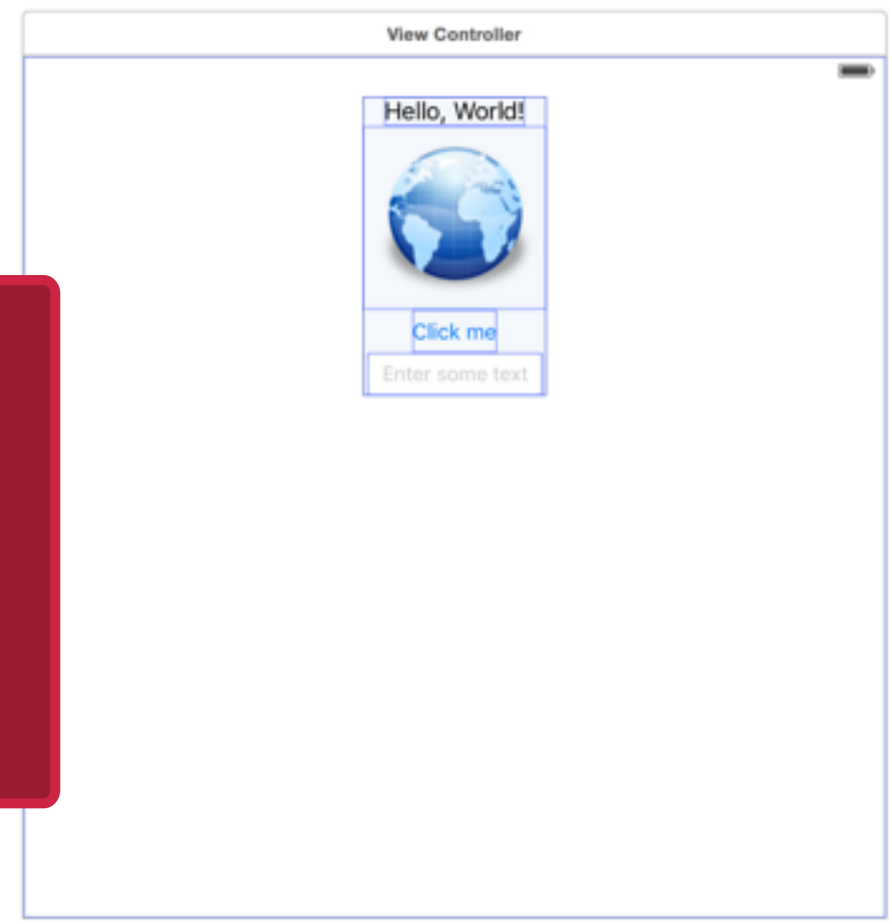
Manual 〉 GitHubDemo 〉 GitHubDemo 〉 Main.storyboard 〉 Main.storyboard (Base)

wAny hAny

GitHubDemo

PREVIOUSLY

```
//
//  ViewController.swift
//  GitHubDemo
//
//  Created by Nick Chen on 7/28/15.
//  Copyright © 2015 TalentSpark. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var hello: UILabel!
    @IBOutlet weak var button: UIButton!
    @IBOutlet weak var textField: UITextField!


    @IBAction func buttonClicked(sender: UIButton) {
        hello.text = "Hello, Clicker!"
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the
            a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recre
    }


}
```

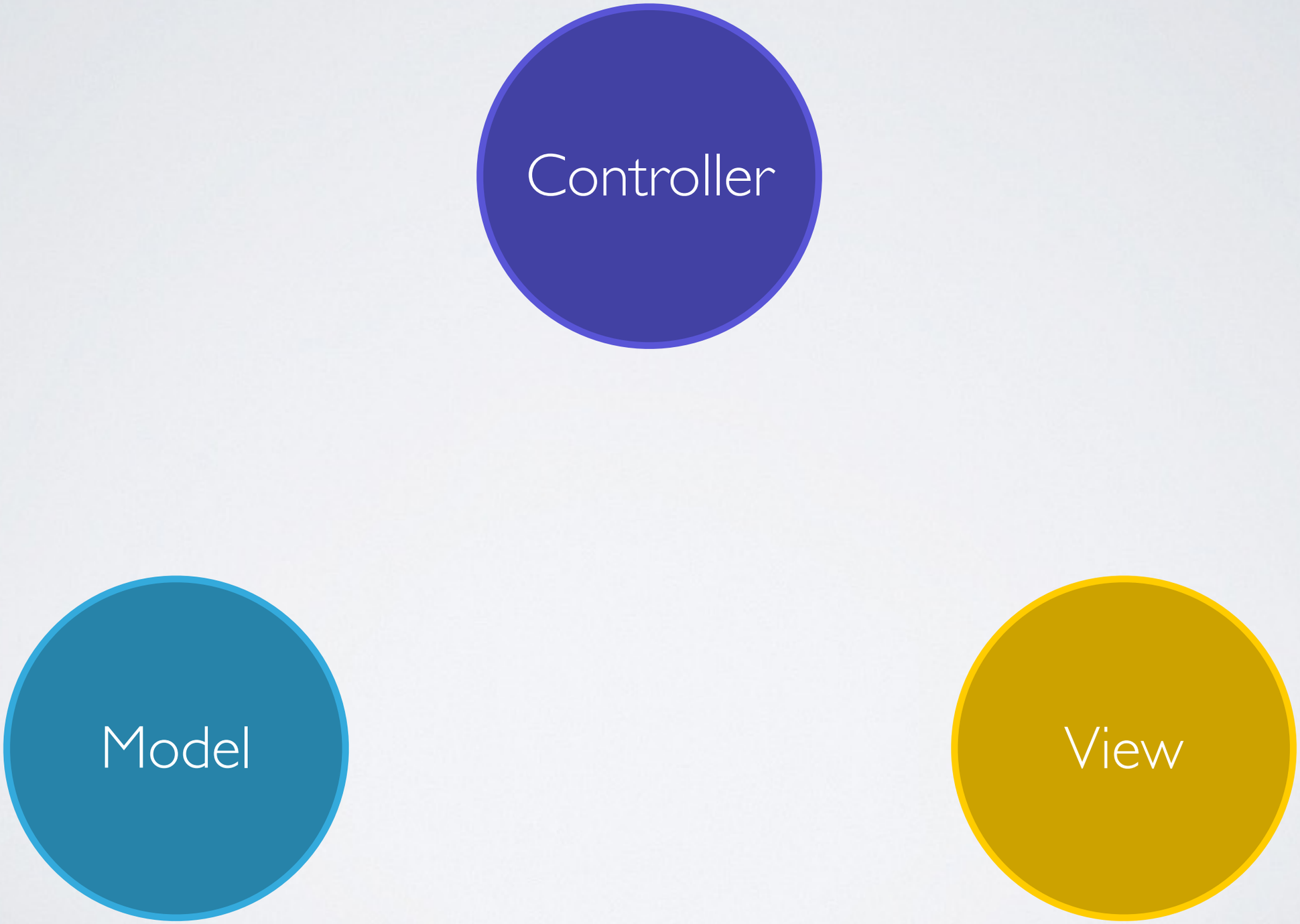


Everything inside one class 😩

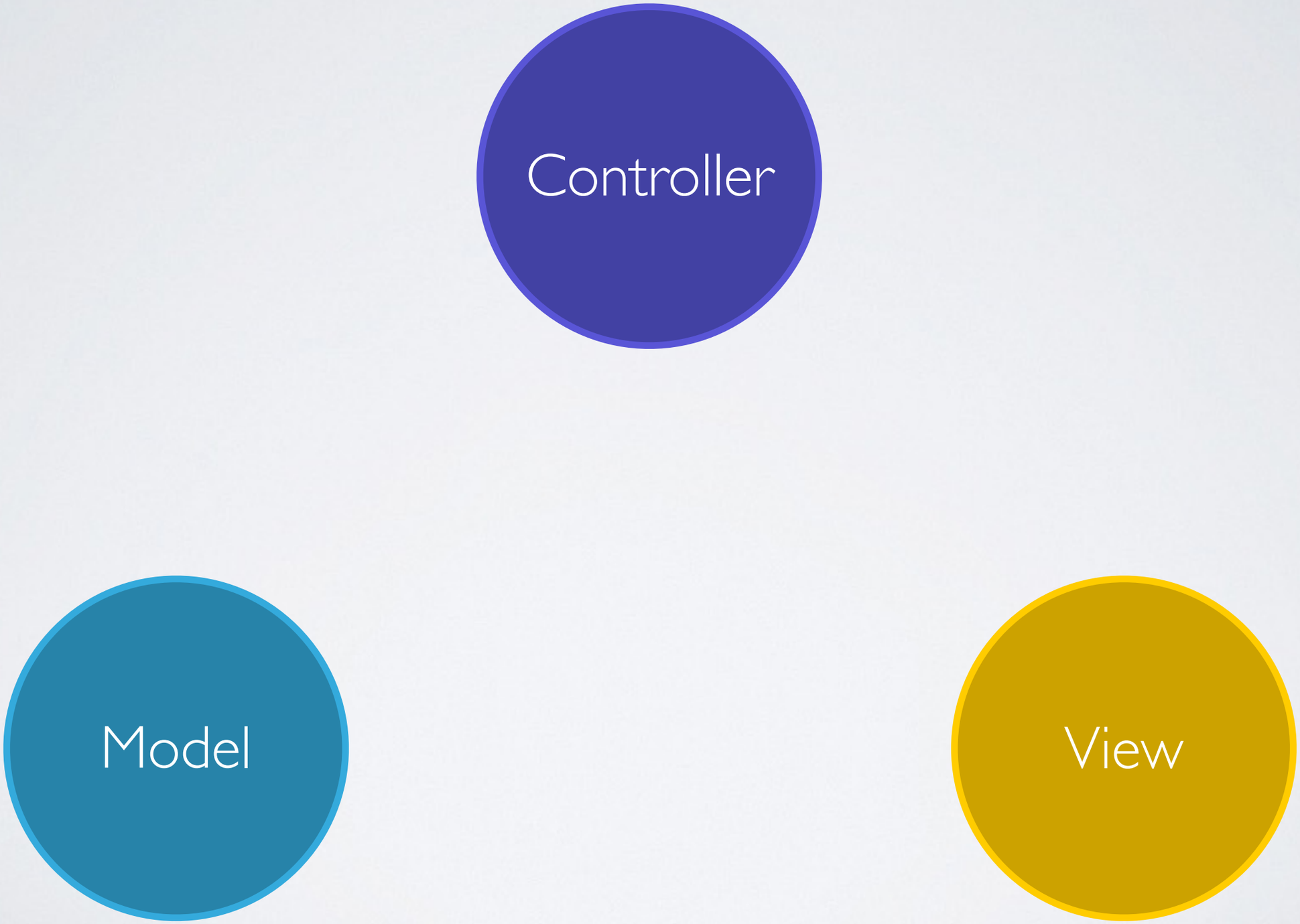
SPAGHETTI CODE

SPAGHETTI CODE

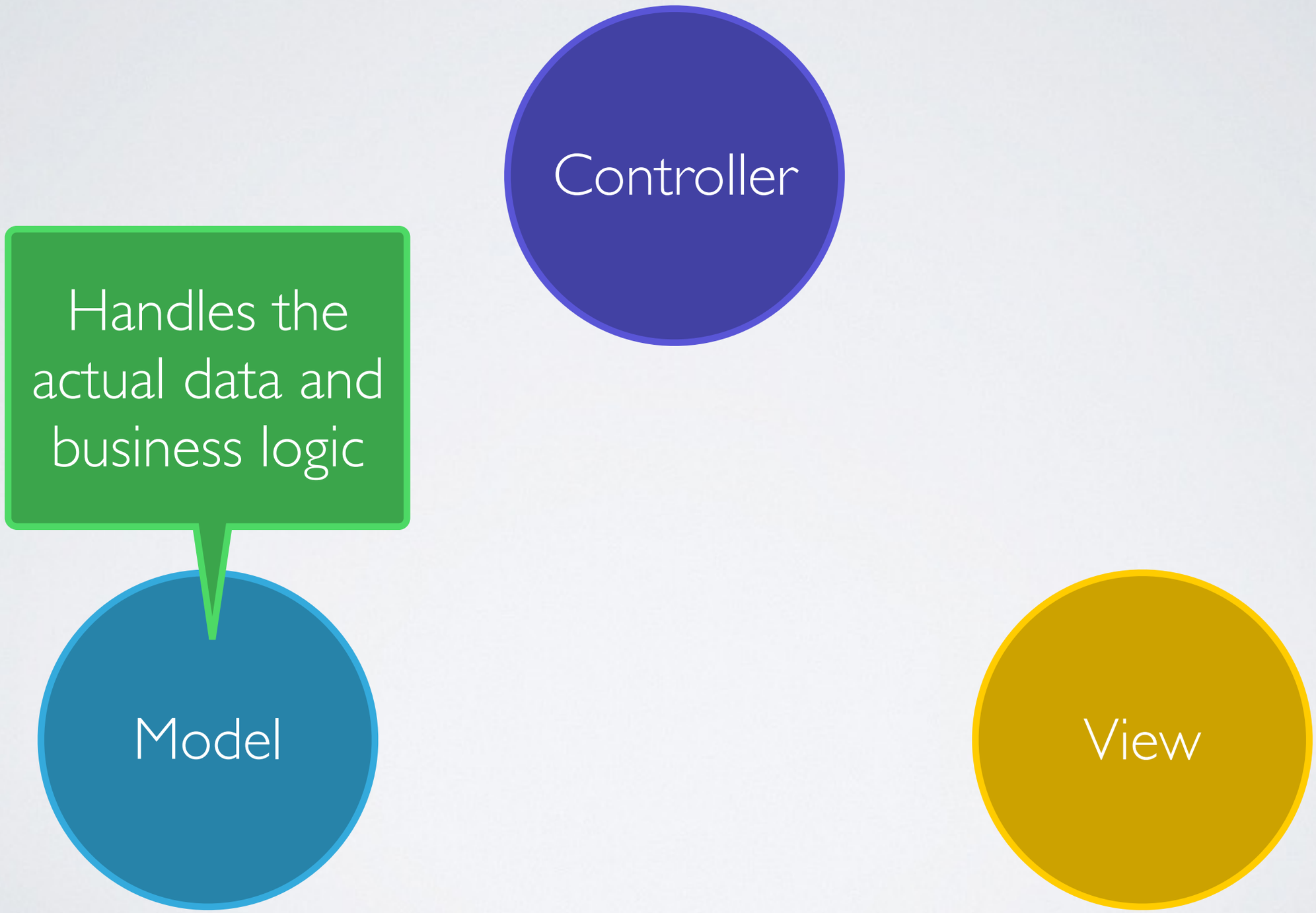
SPAGHETTI CODE



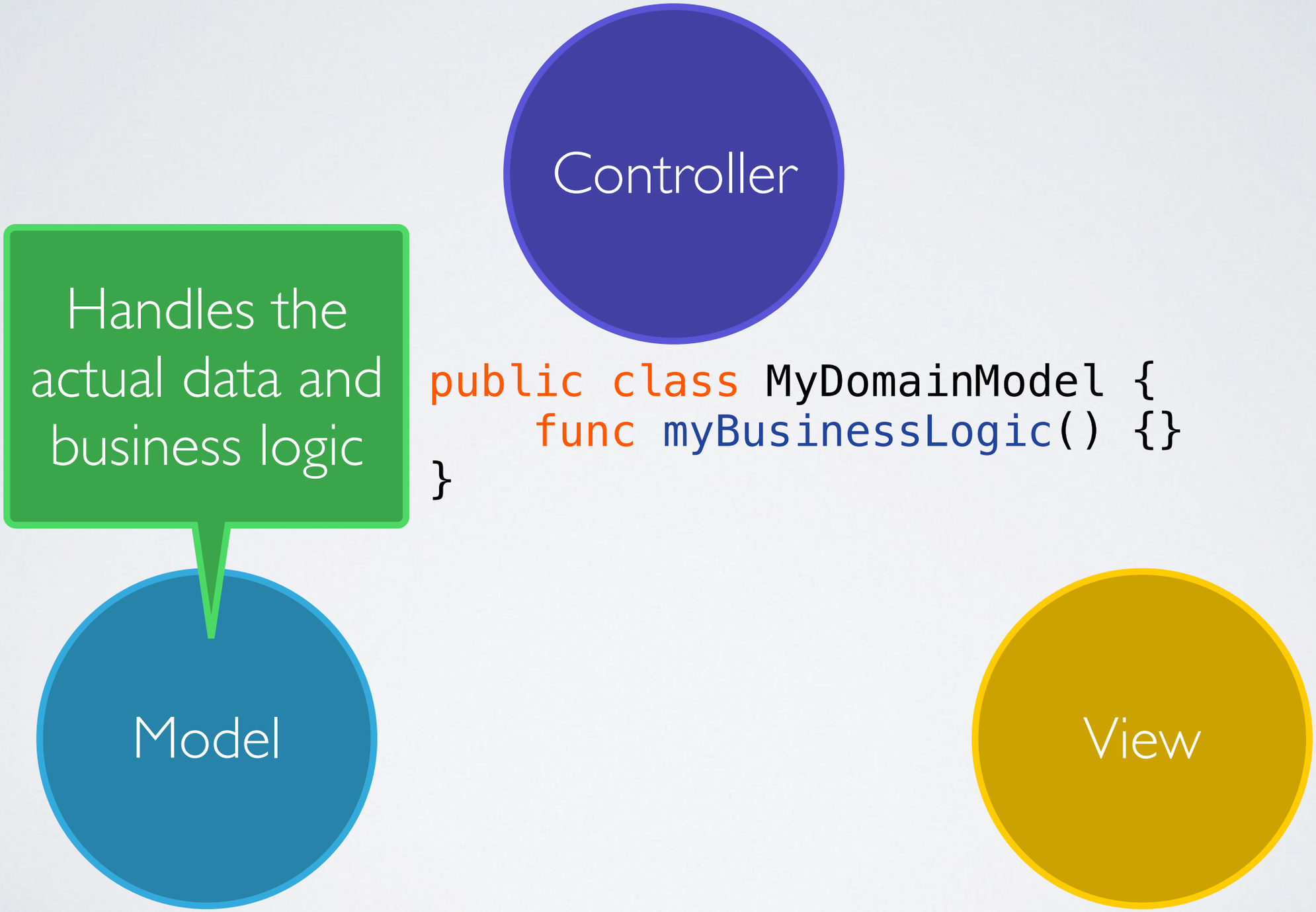
NON-SPAGHETTI CODE



NON-SPAGHETTI CODE

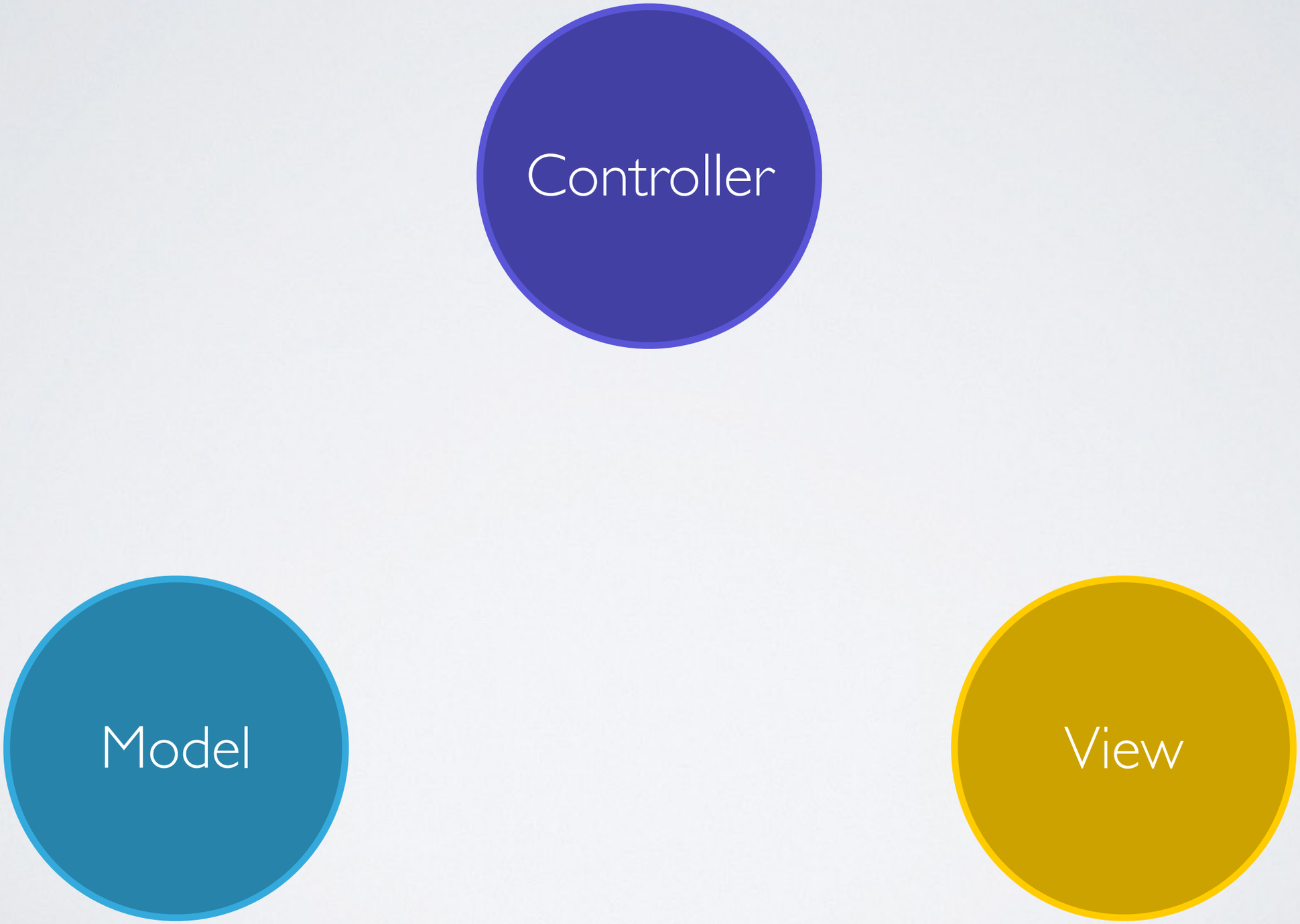


NON-SPAGHETTI CODE

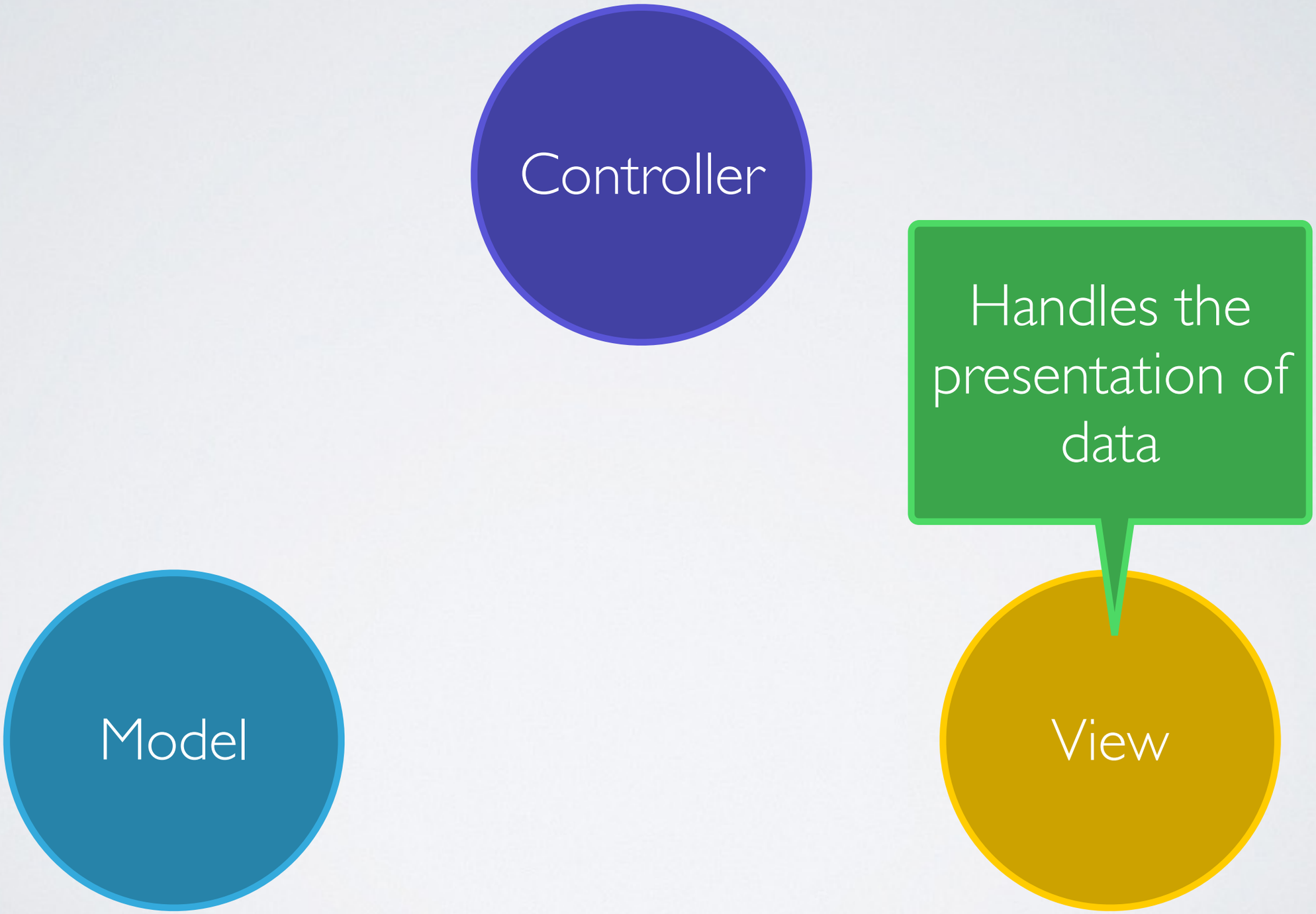



```
public class MyDomainModel {
    func myBusinessLogic() {}
}
```

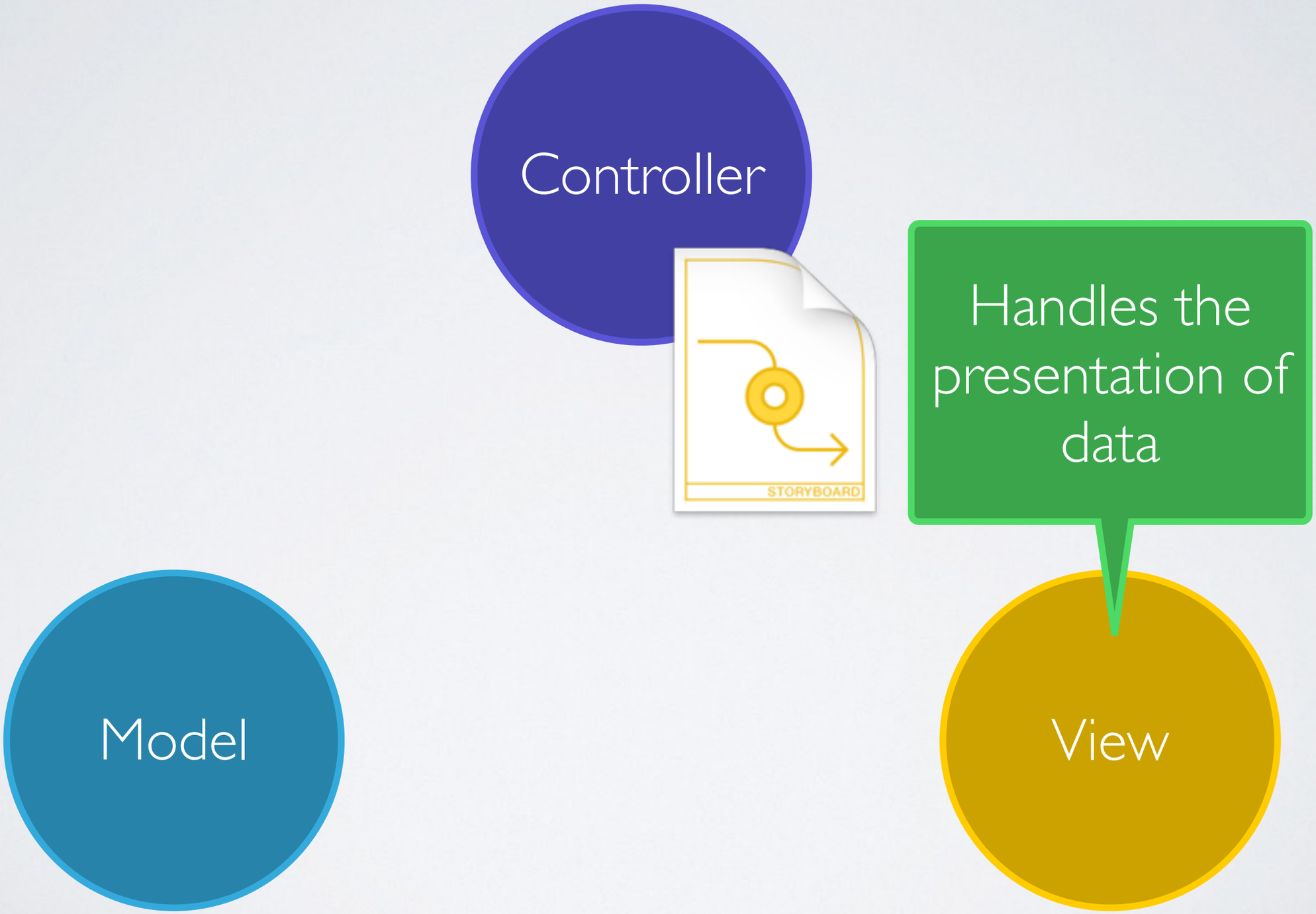
NON-SPAGHETTI CODE



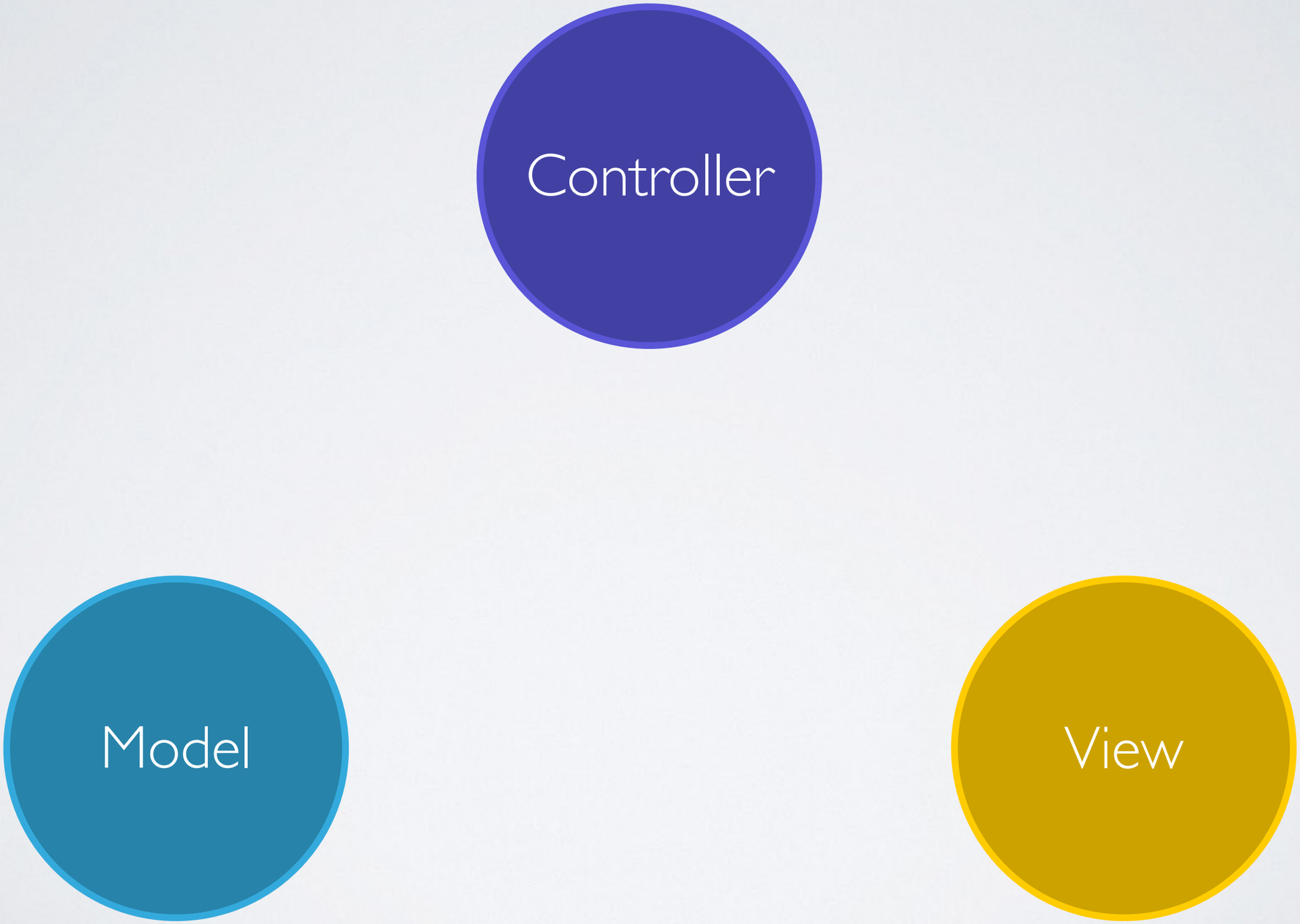
NON-SPAGHETTI CODE



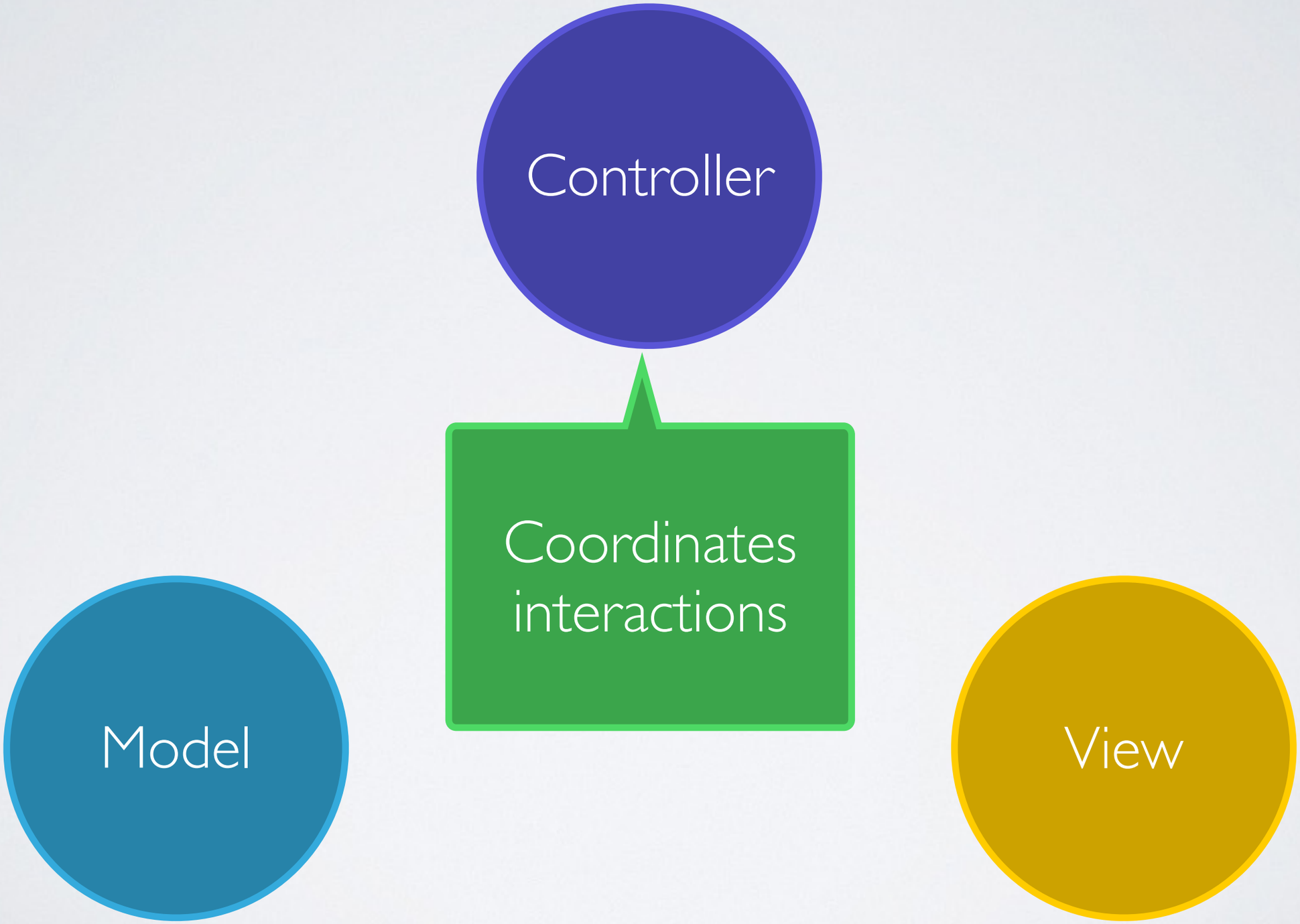
NON-SPAGHETTI CODE



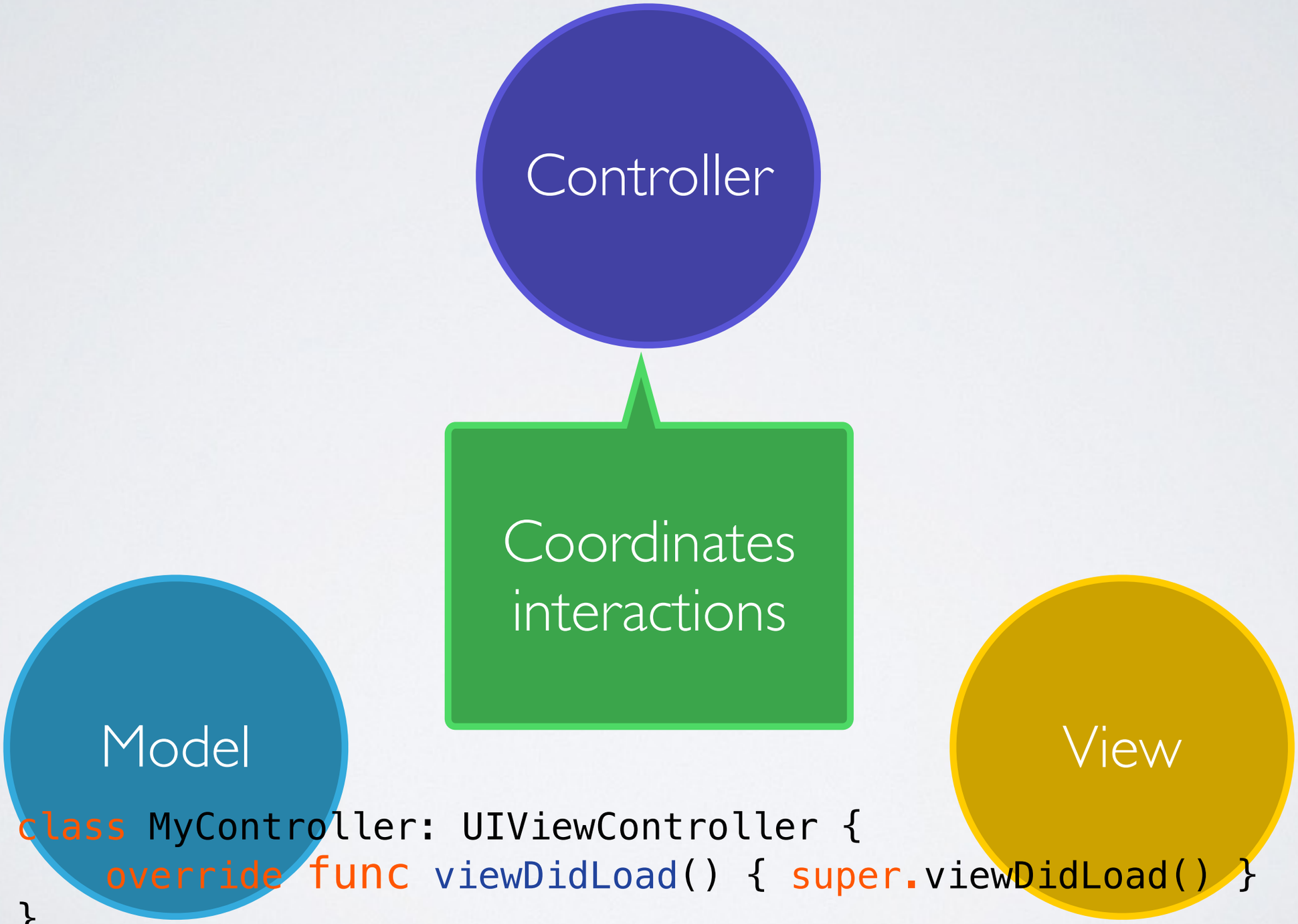
NON-SPAGHETTI CODE



NON-SPAGHETTI CODE

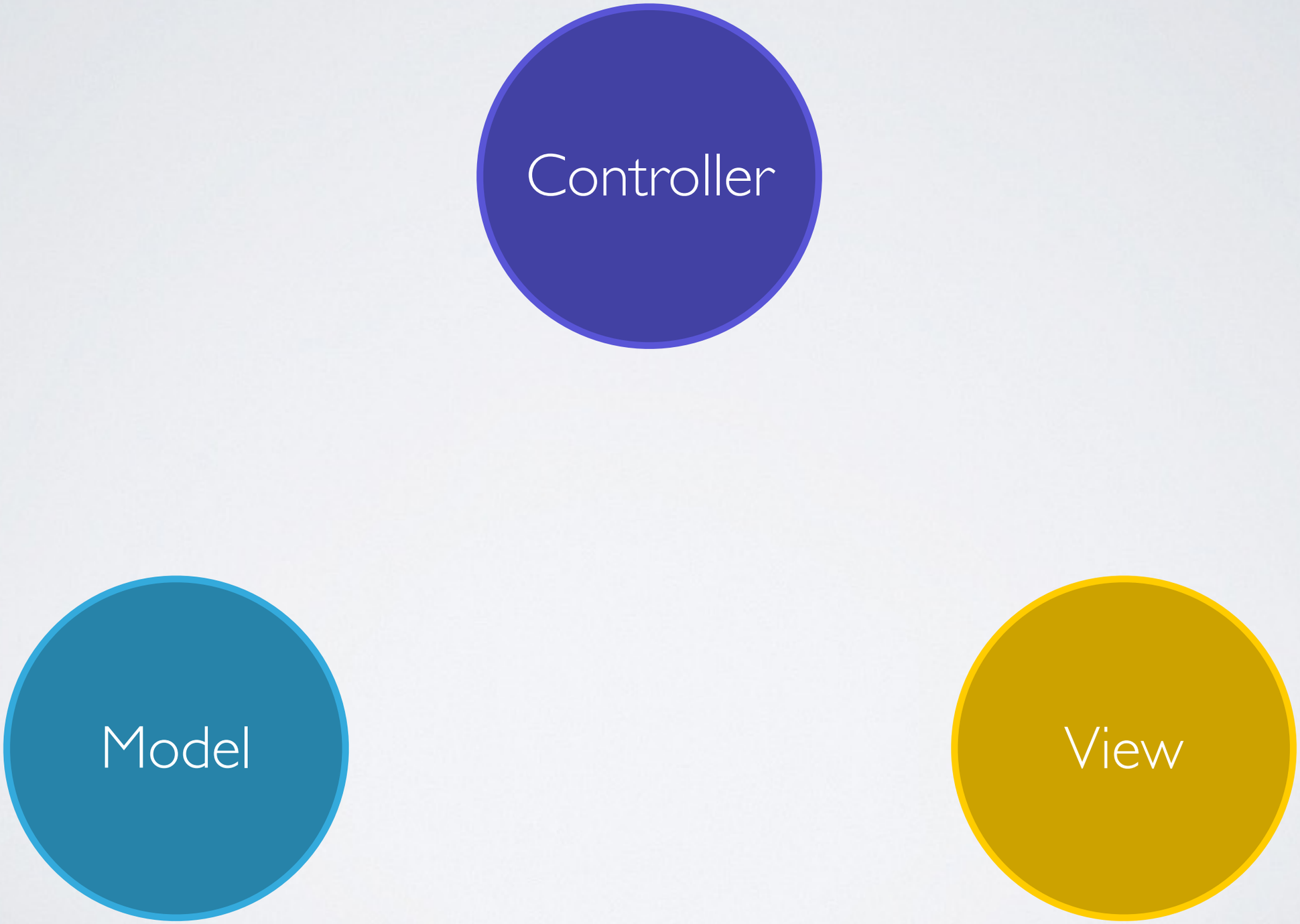


NON-SPAGHETTI CODE

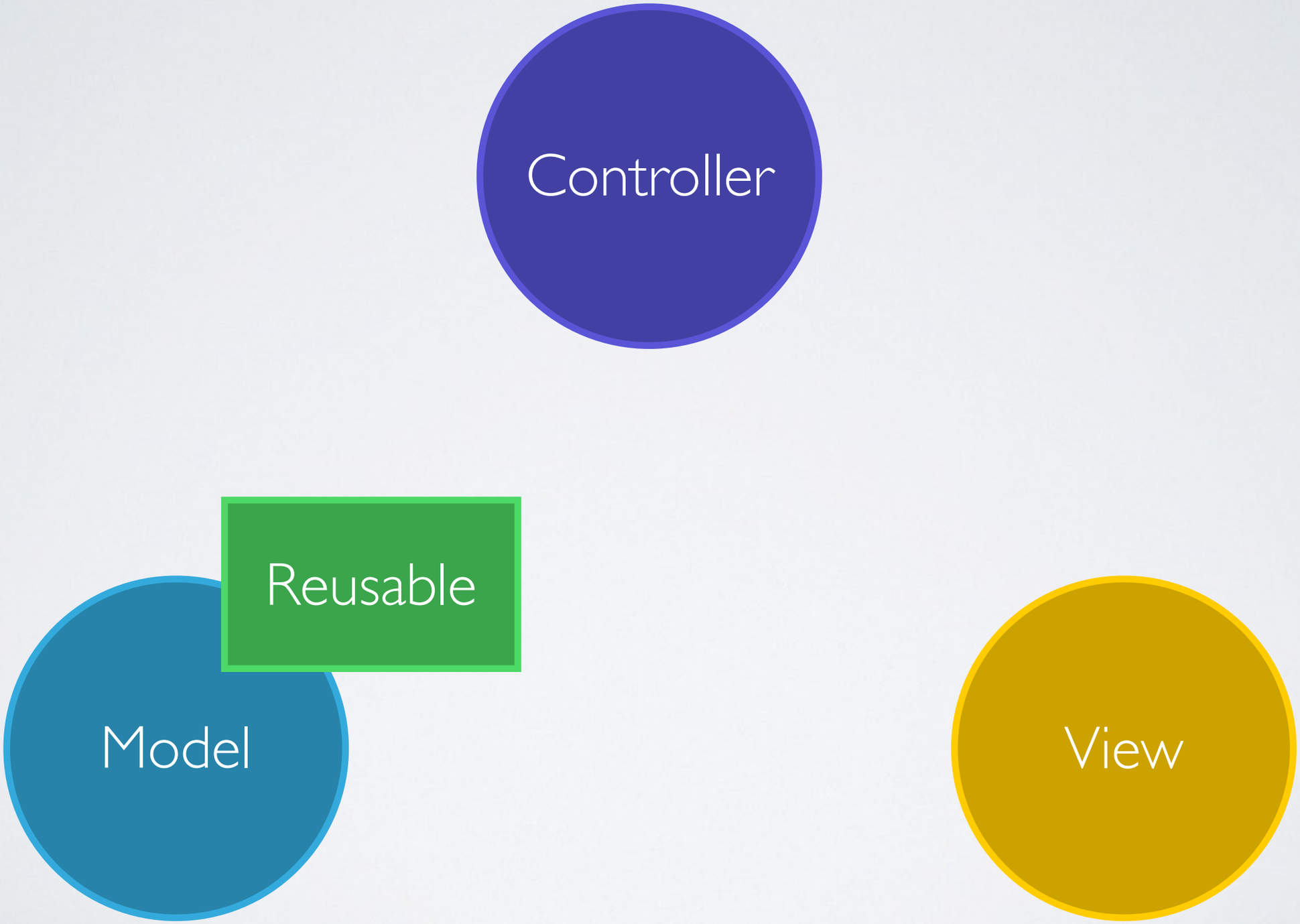



```
class MyController: UIViewController {
    override func viewDidLoad() { super.viewDidLoad() }
}
```

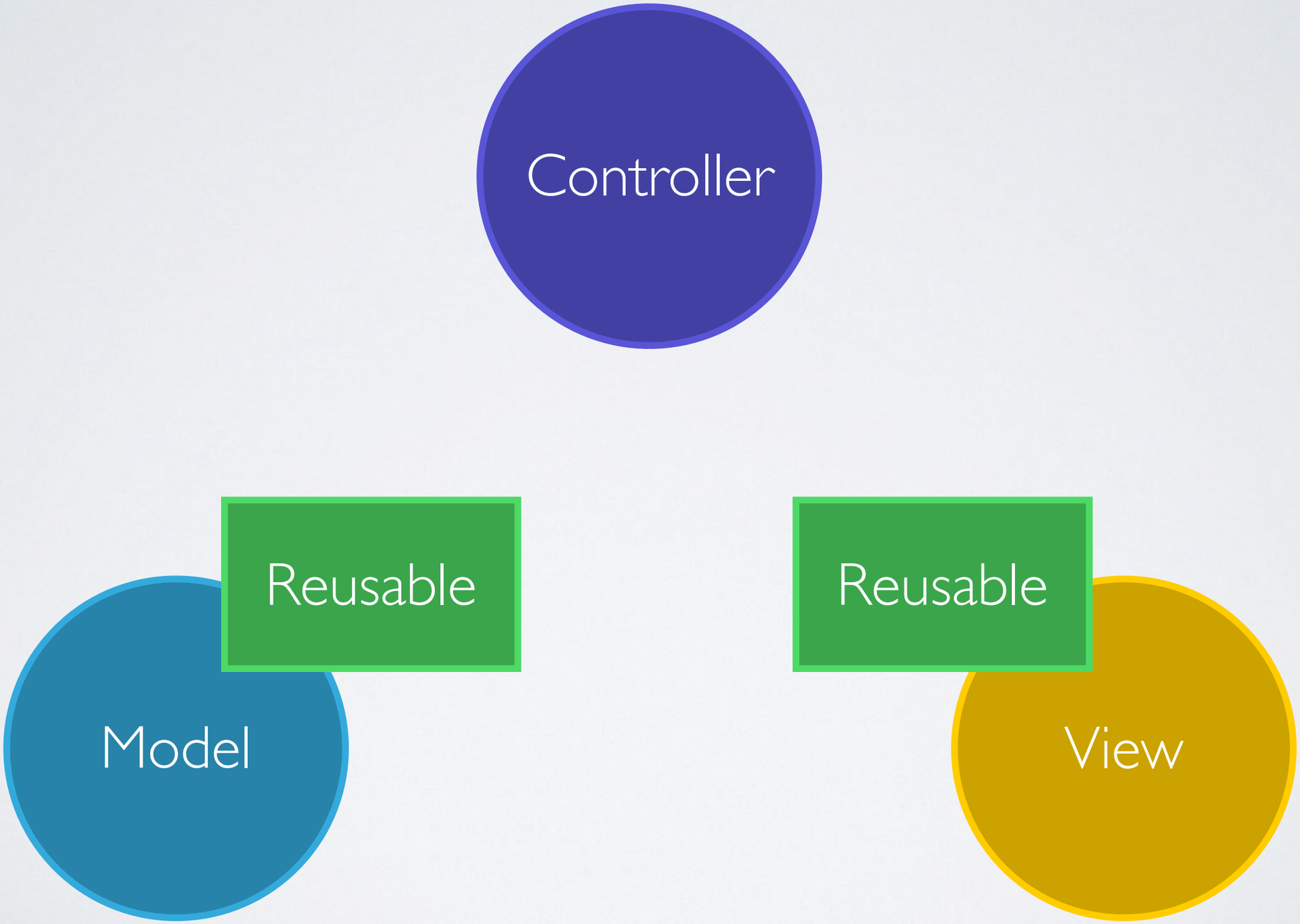
NON-SPAGHETTI CODE



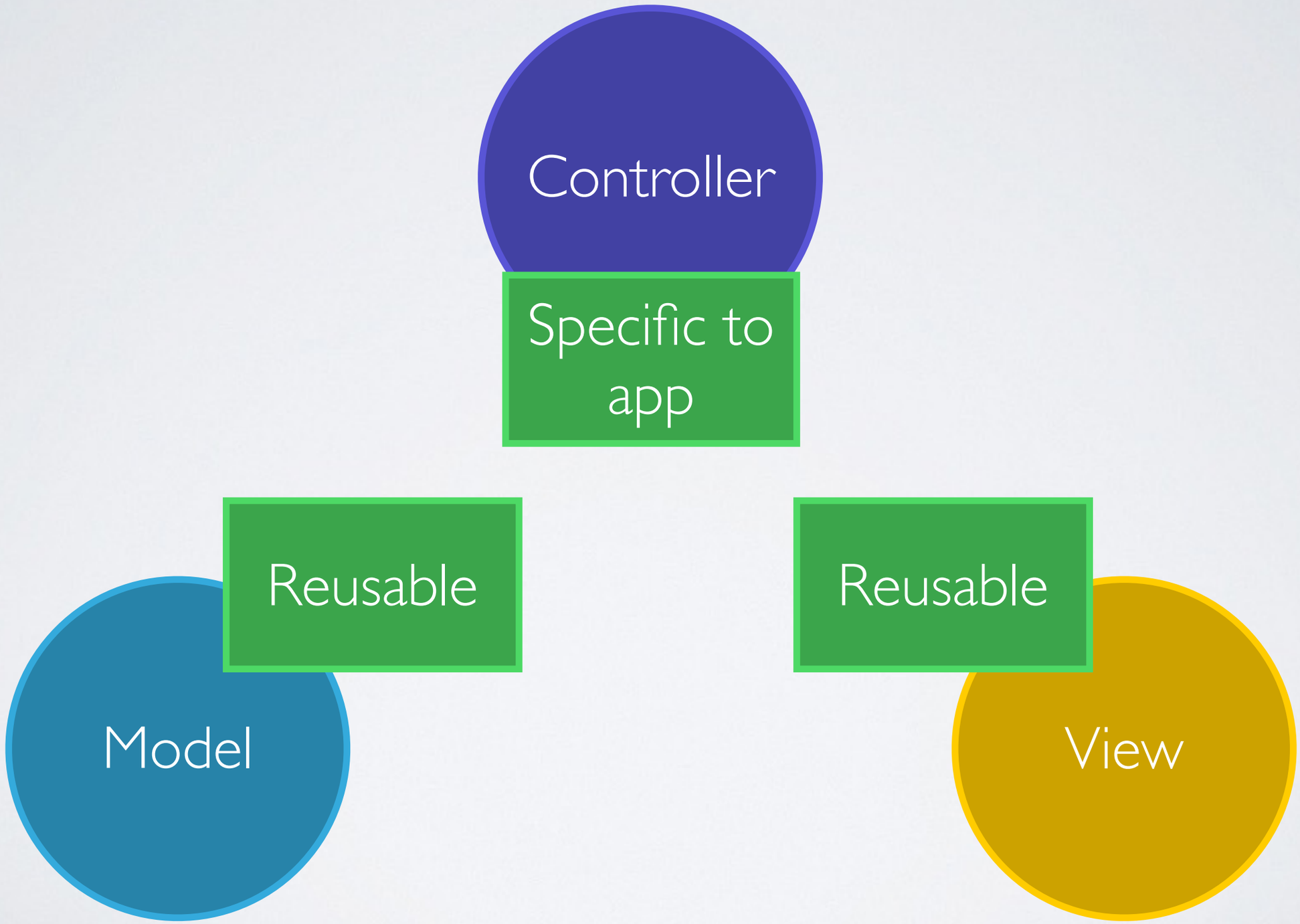
NON-SPAGHETTI CODE



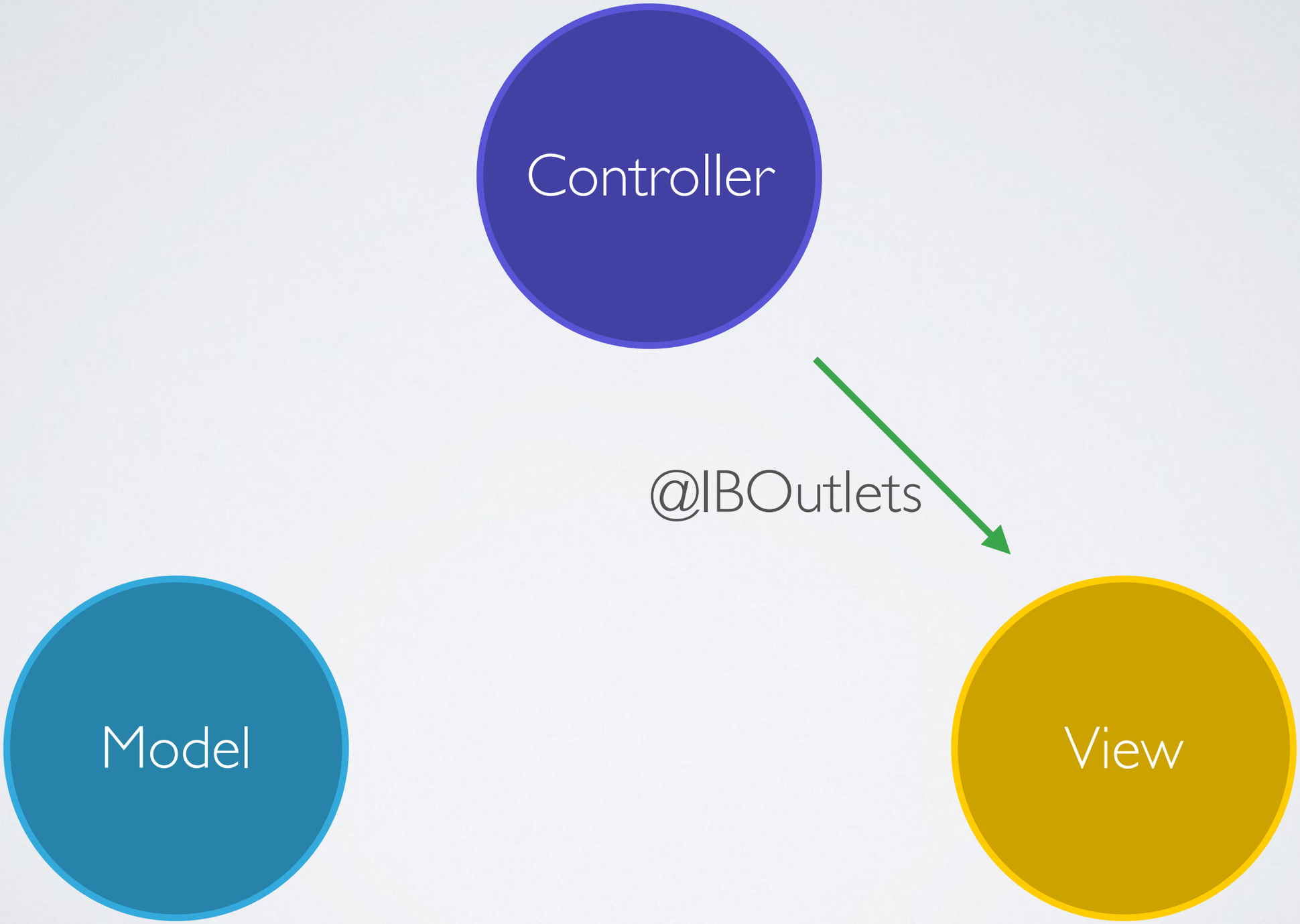
NON-SPAGHETTI CODE



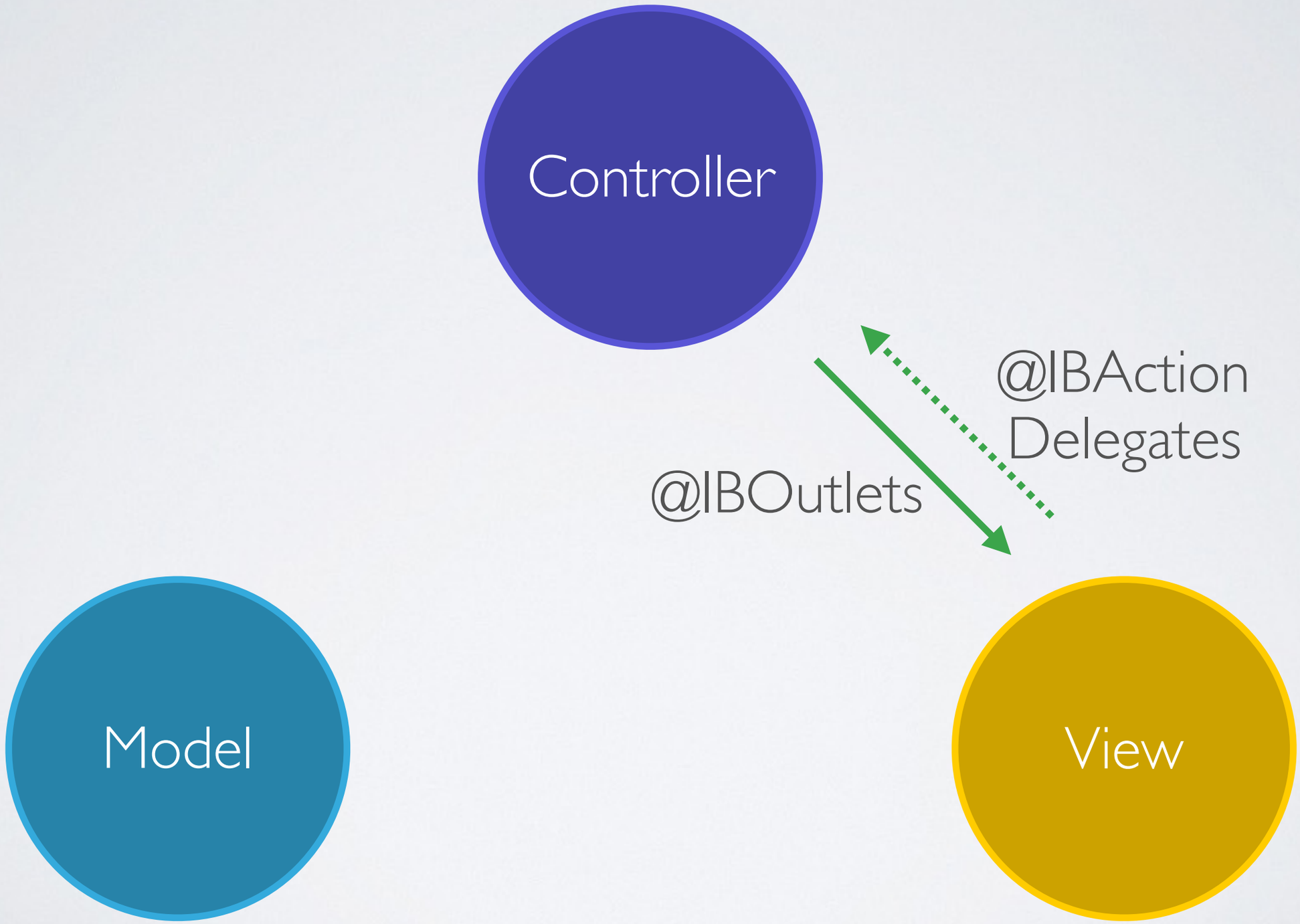
NON-SPAGHETTI CODE



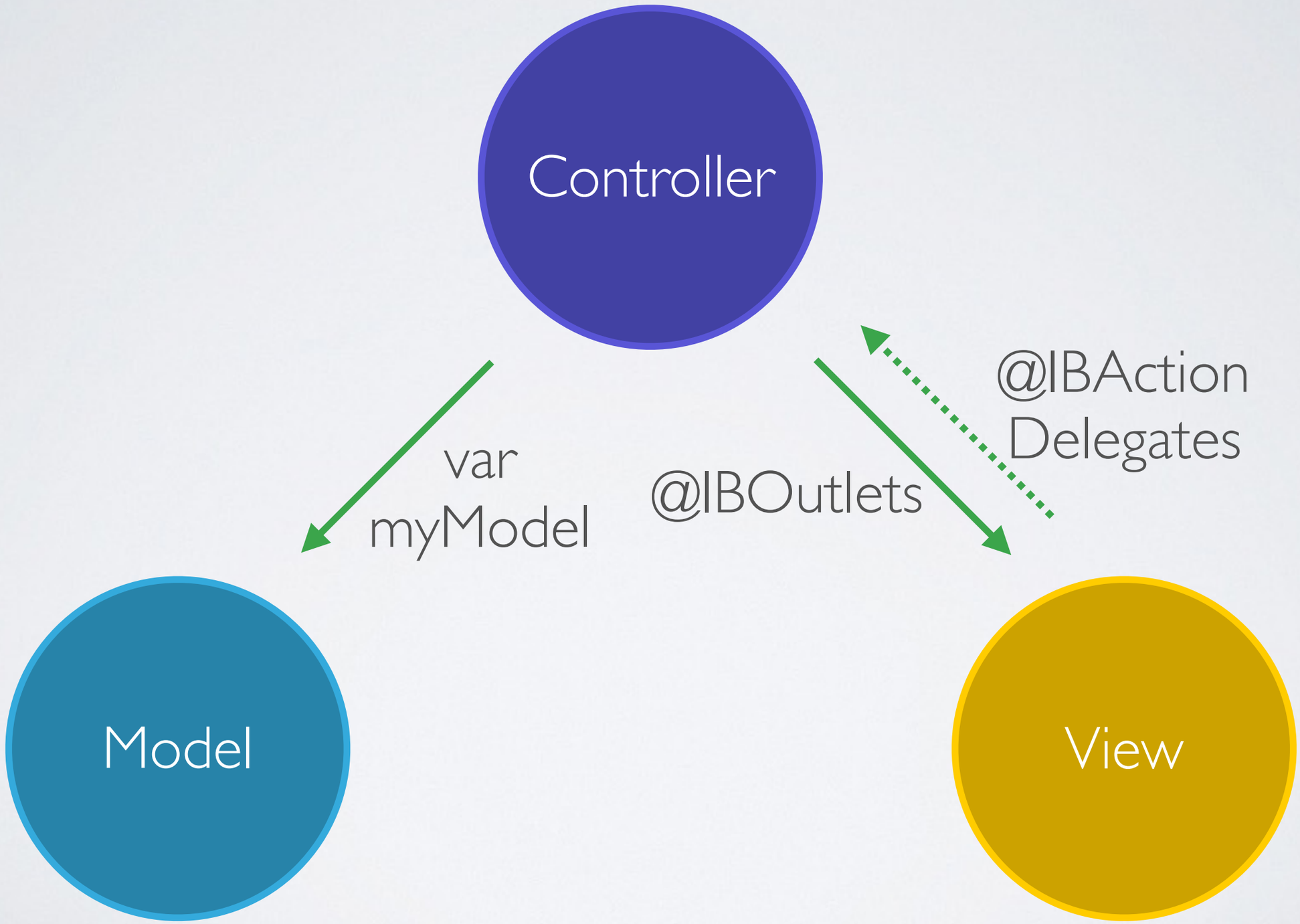
NON-SPAGHETTI CODE



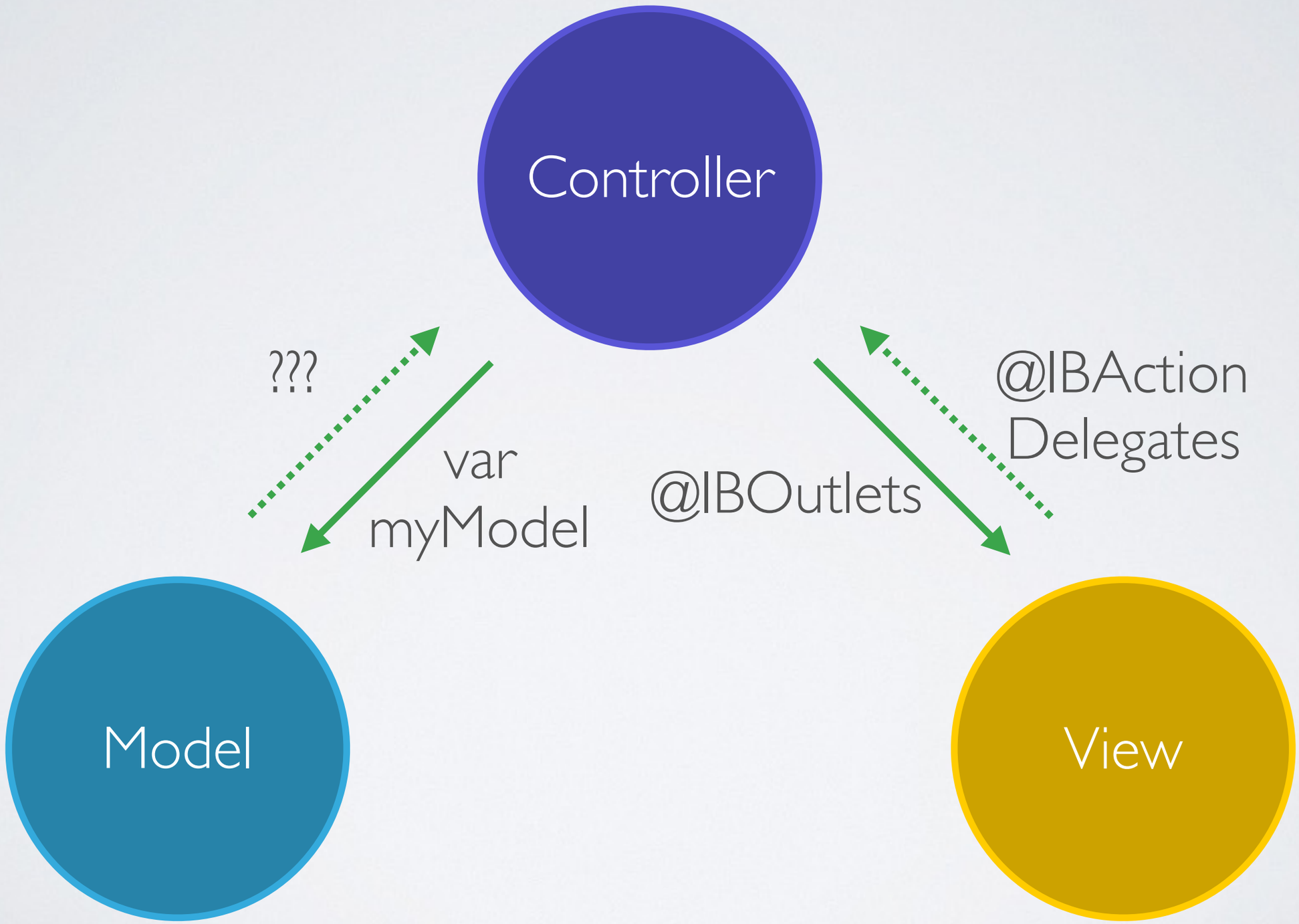
NON-SPAGHETTI CODE



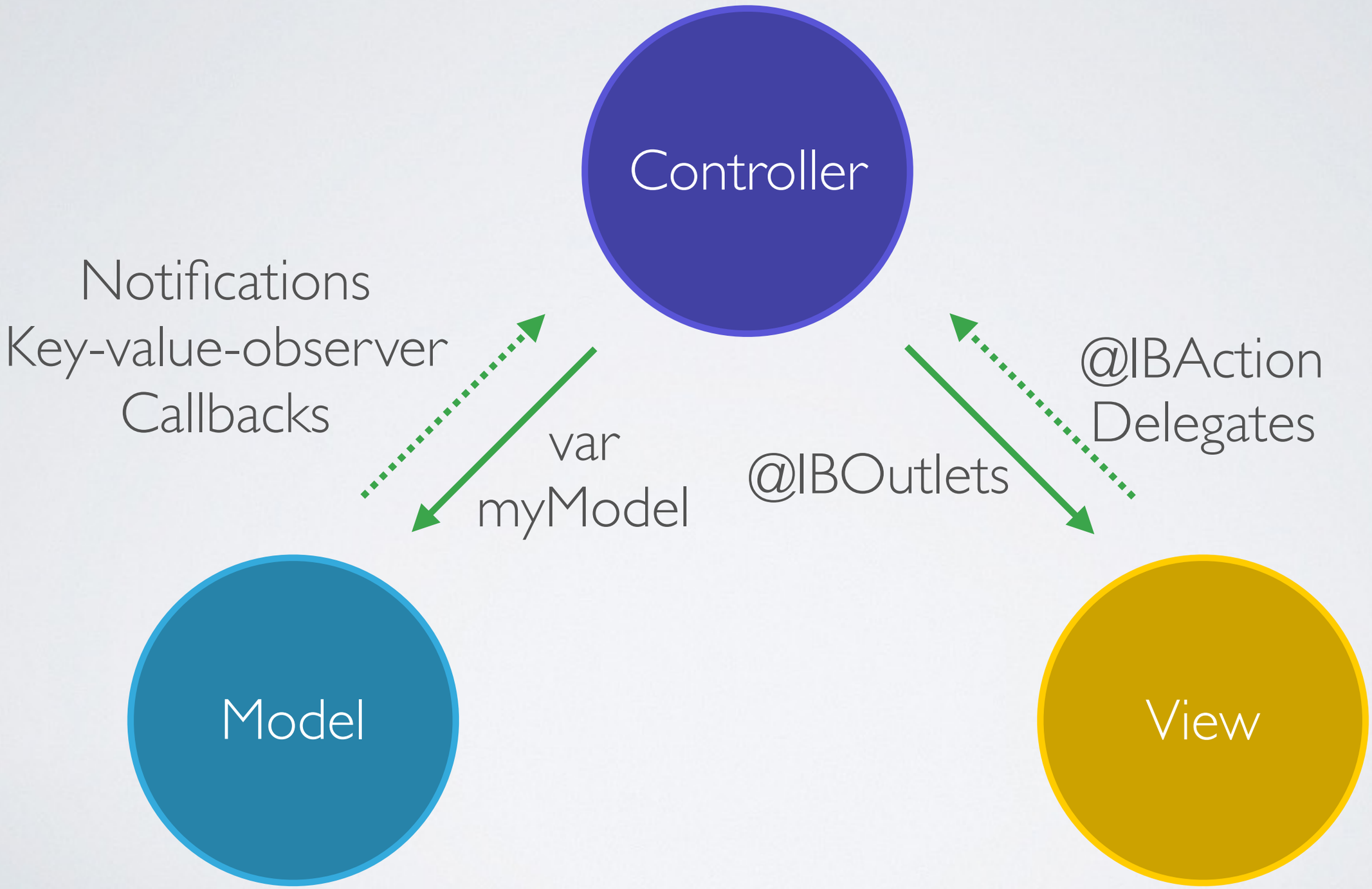
NON-SPAGHETTI CODE



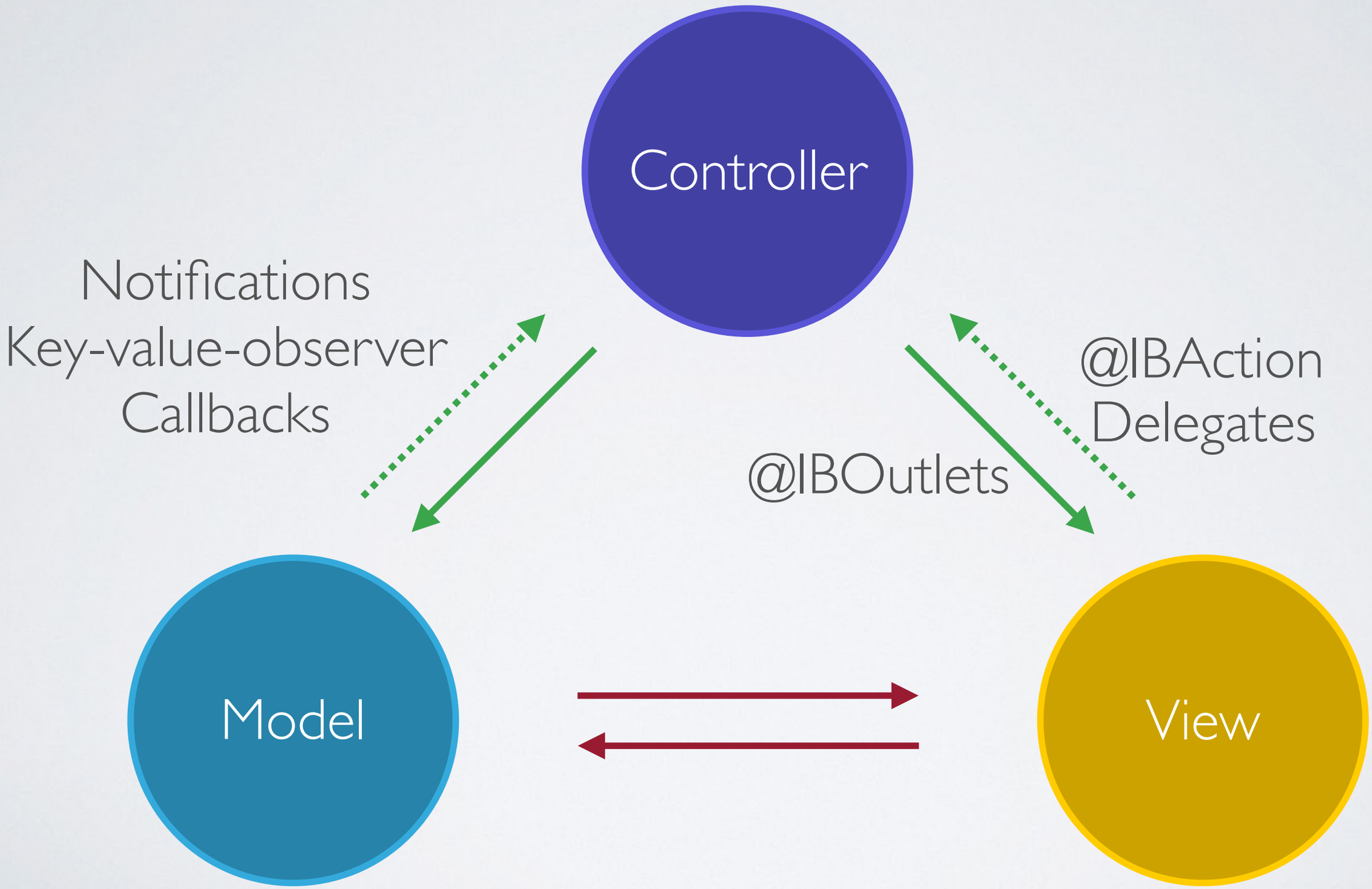
NON-SPAGHETTI CODE



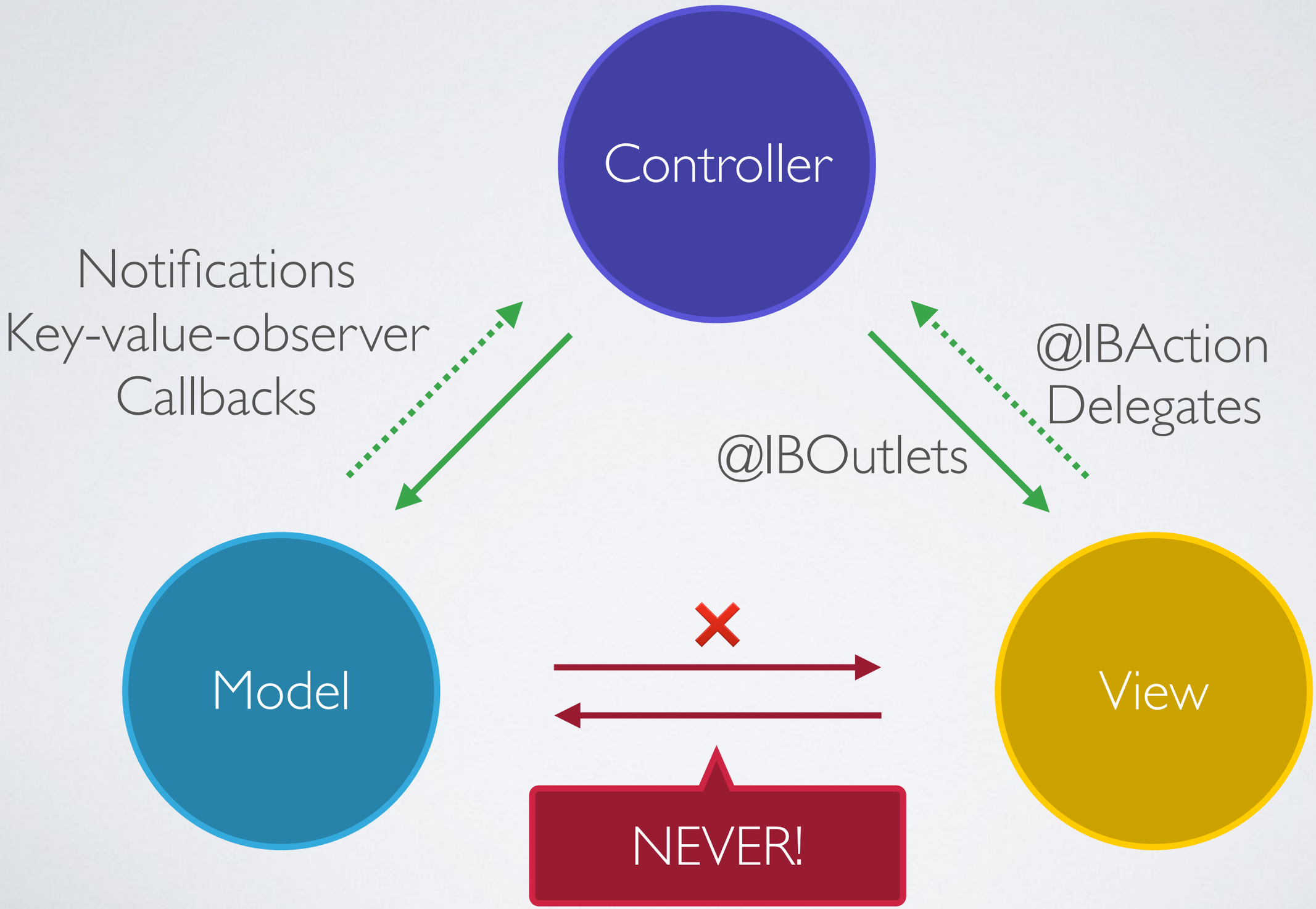
NON-SPAGHETTI CODE



NON-SPAGHETTI CODE

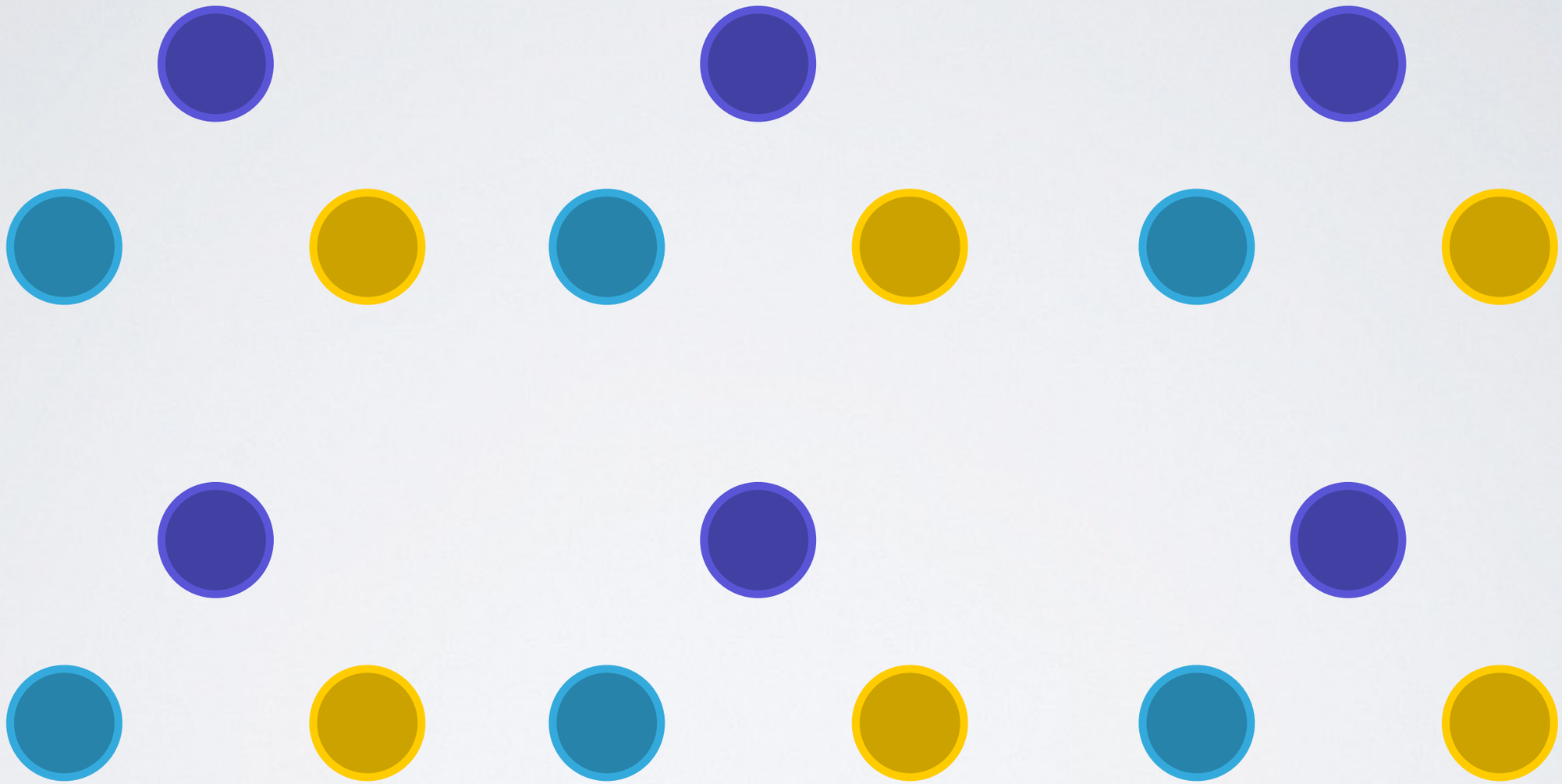


NON-SPAGHETTI CODE

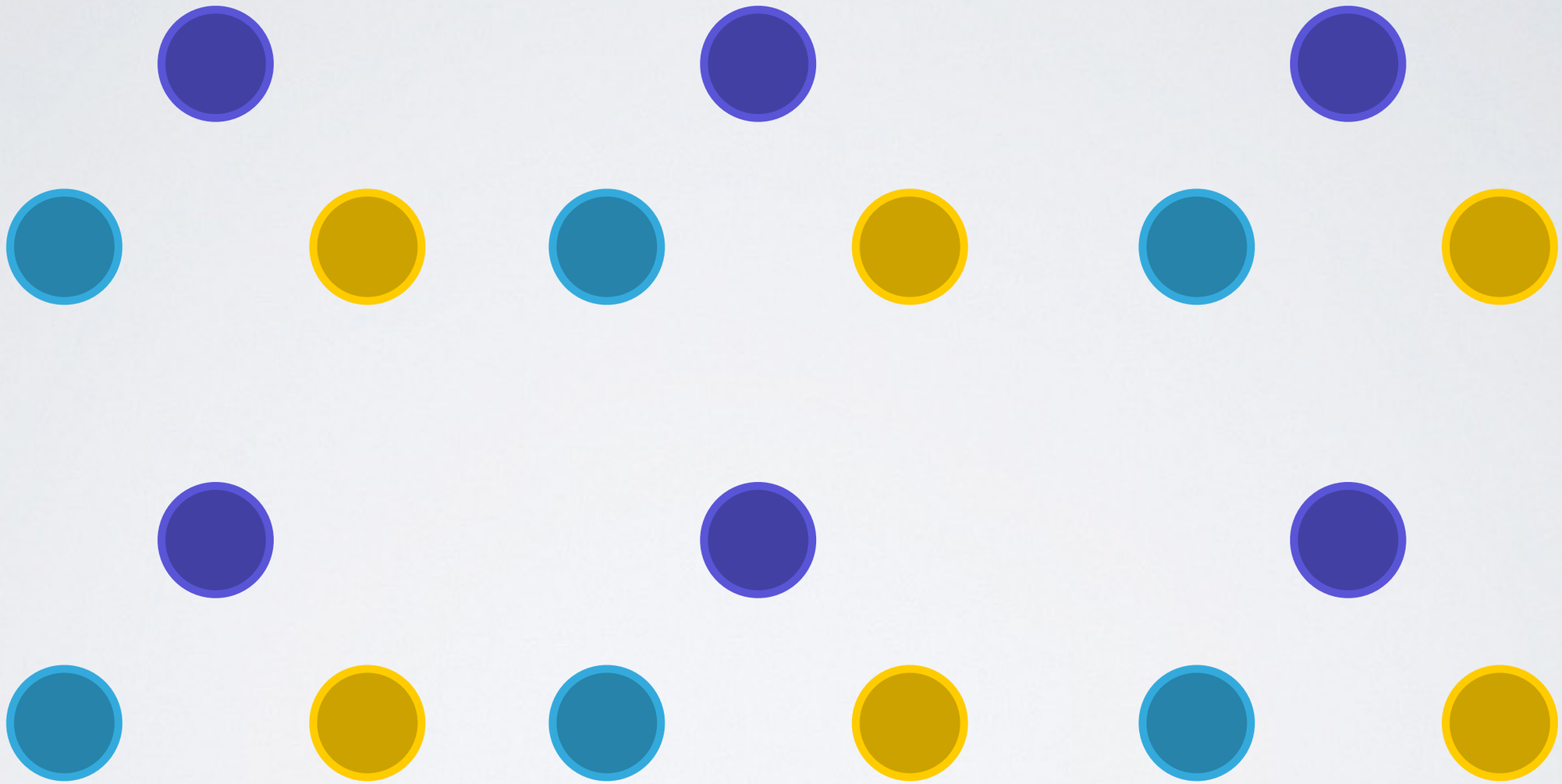


MULTIPLE MVCS

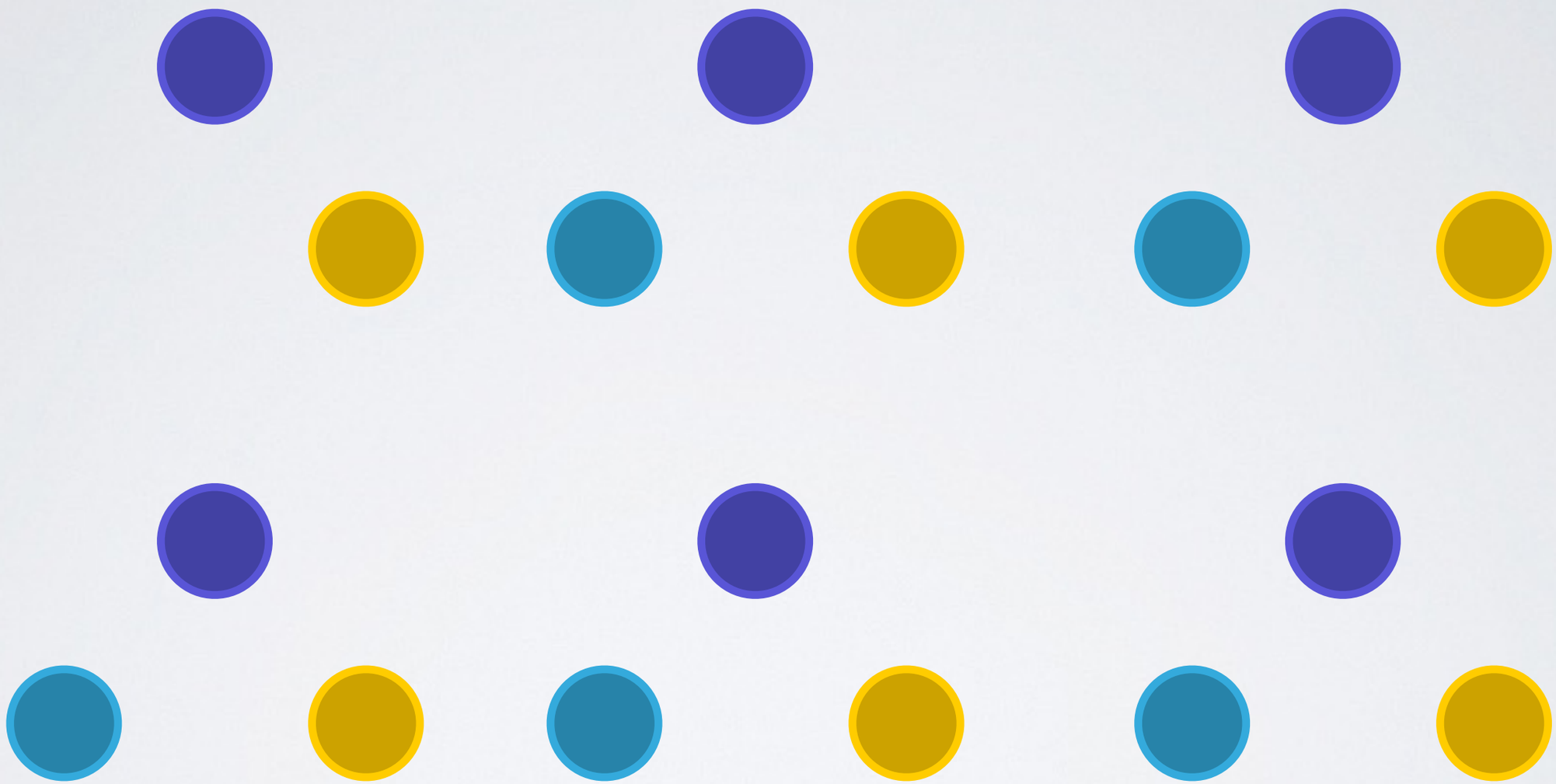
MULTIPLE MVCS



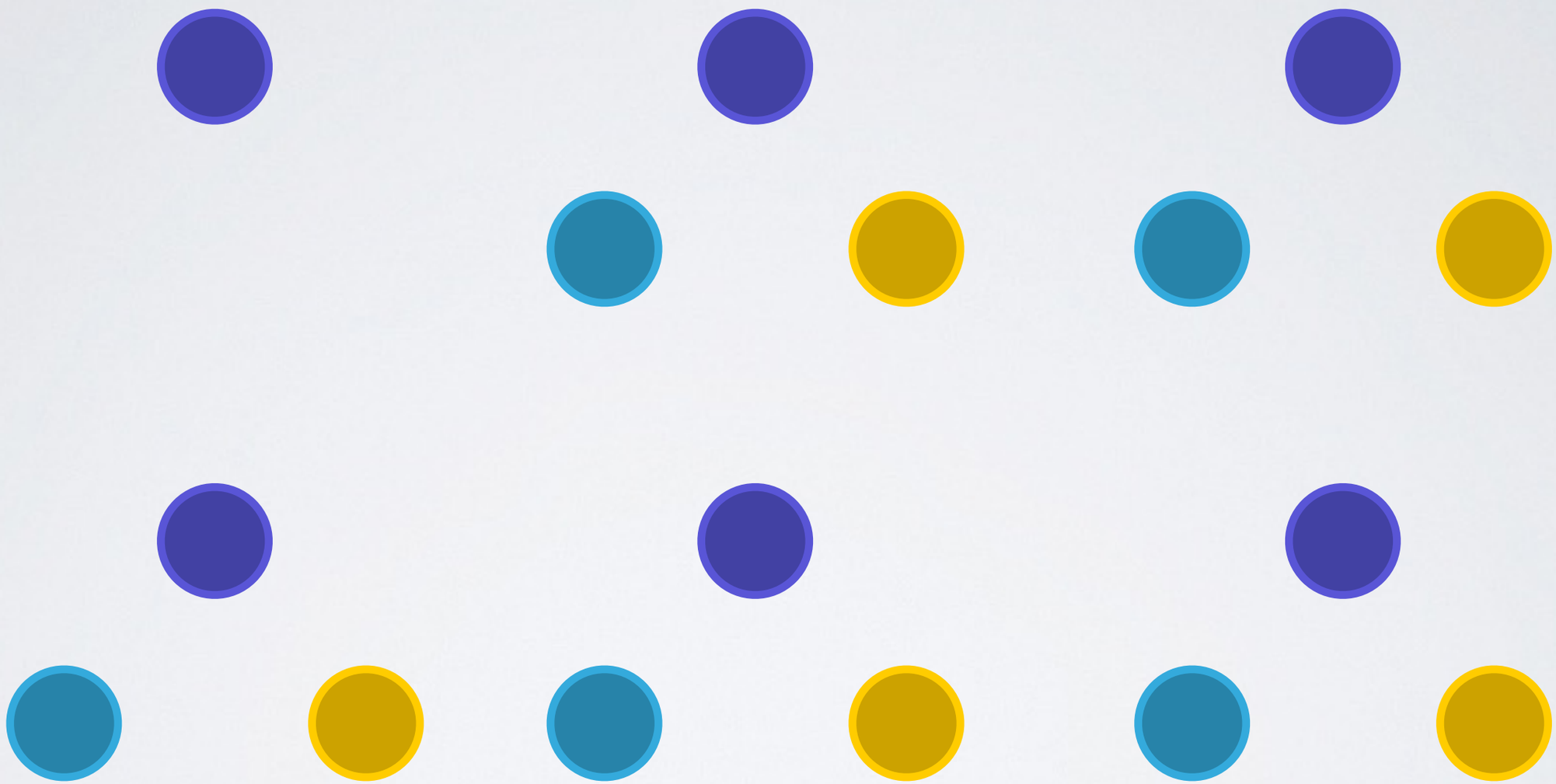
MULTIPLE MVCS



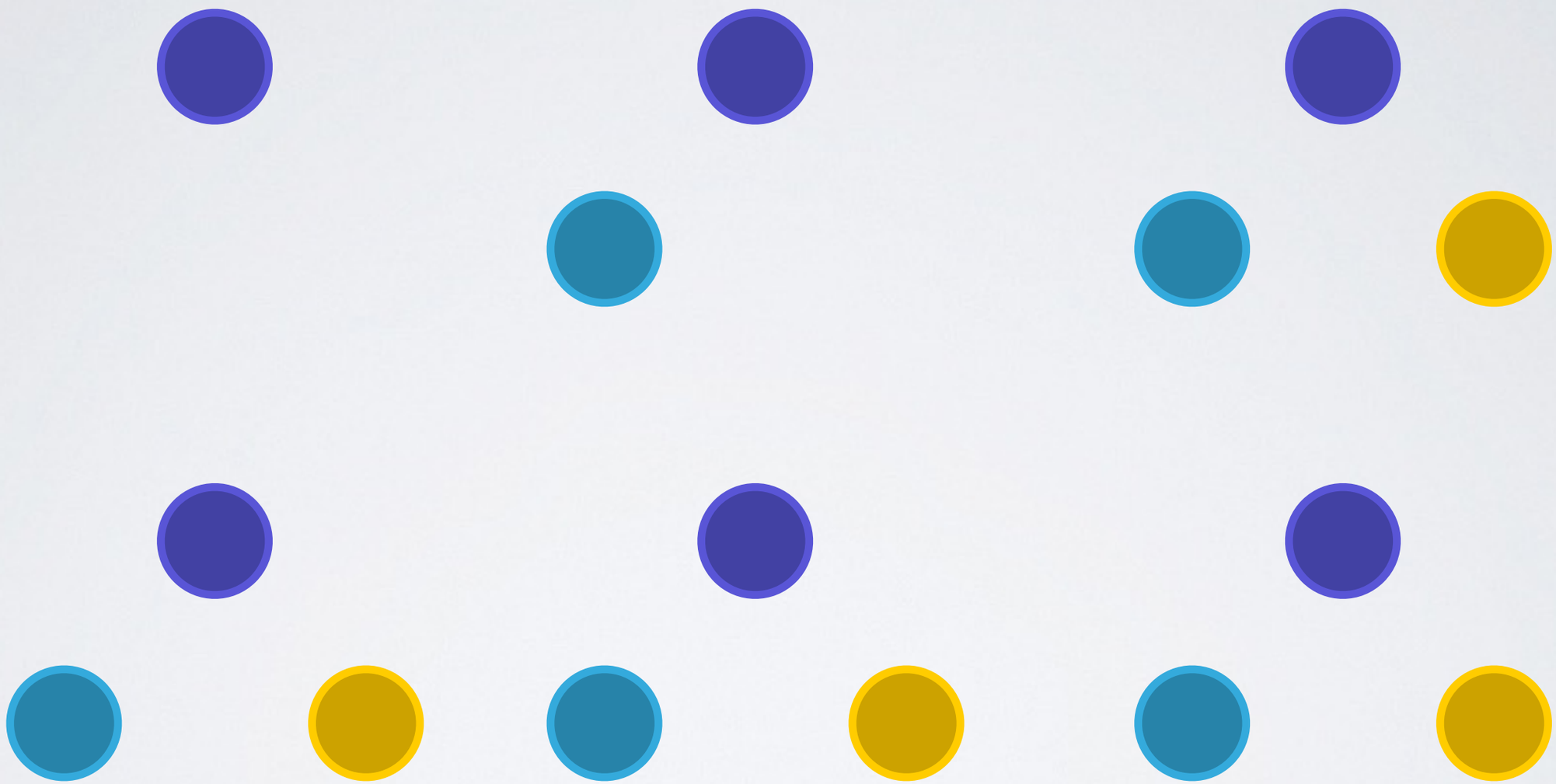
MULTIPLE MVCS



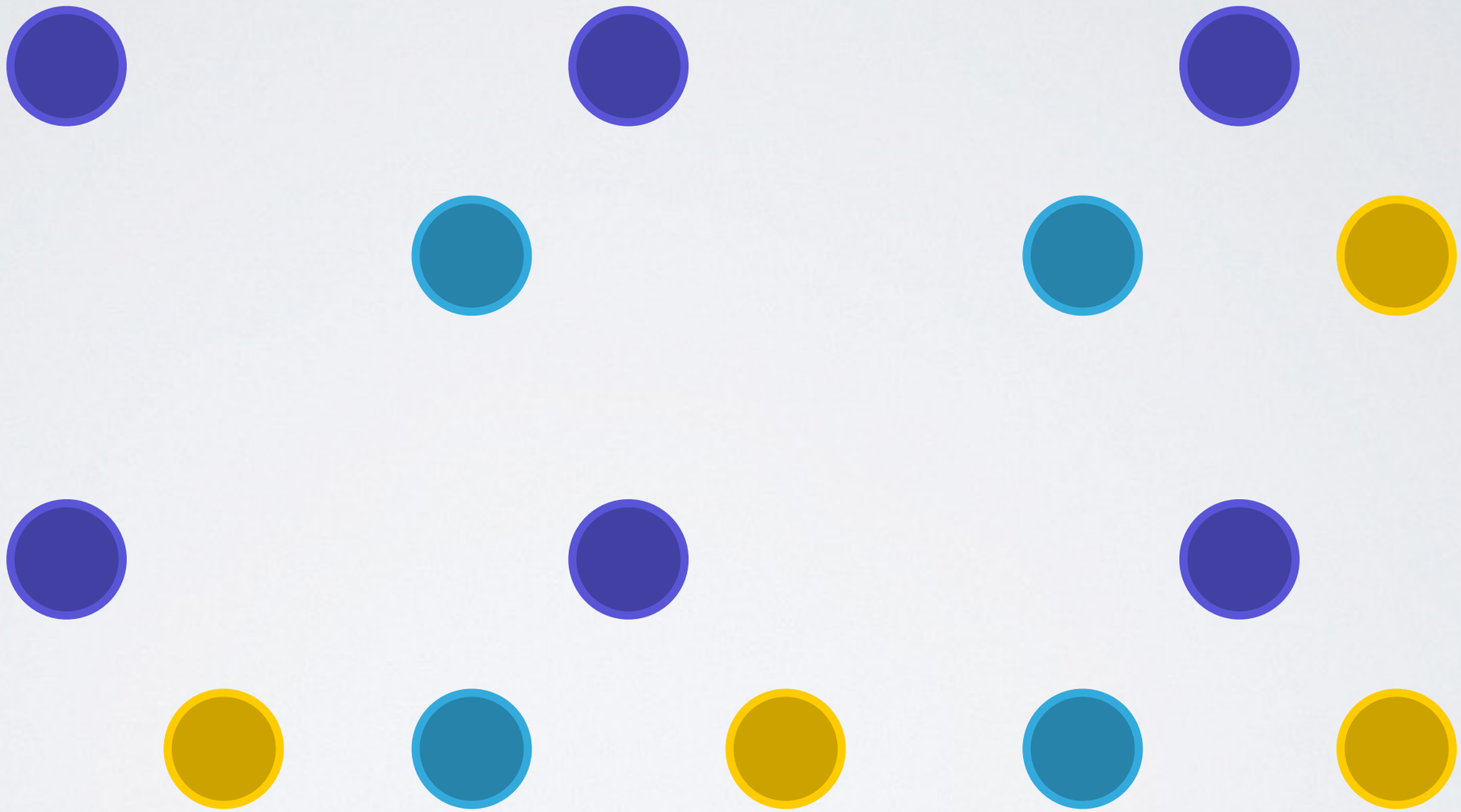
MULTIPLE MVCS



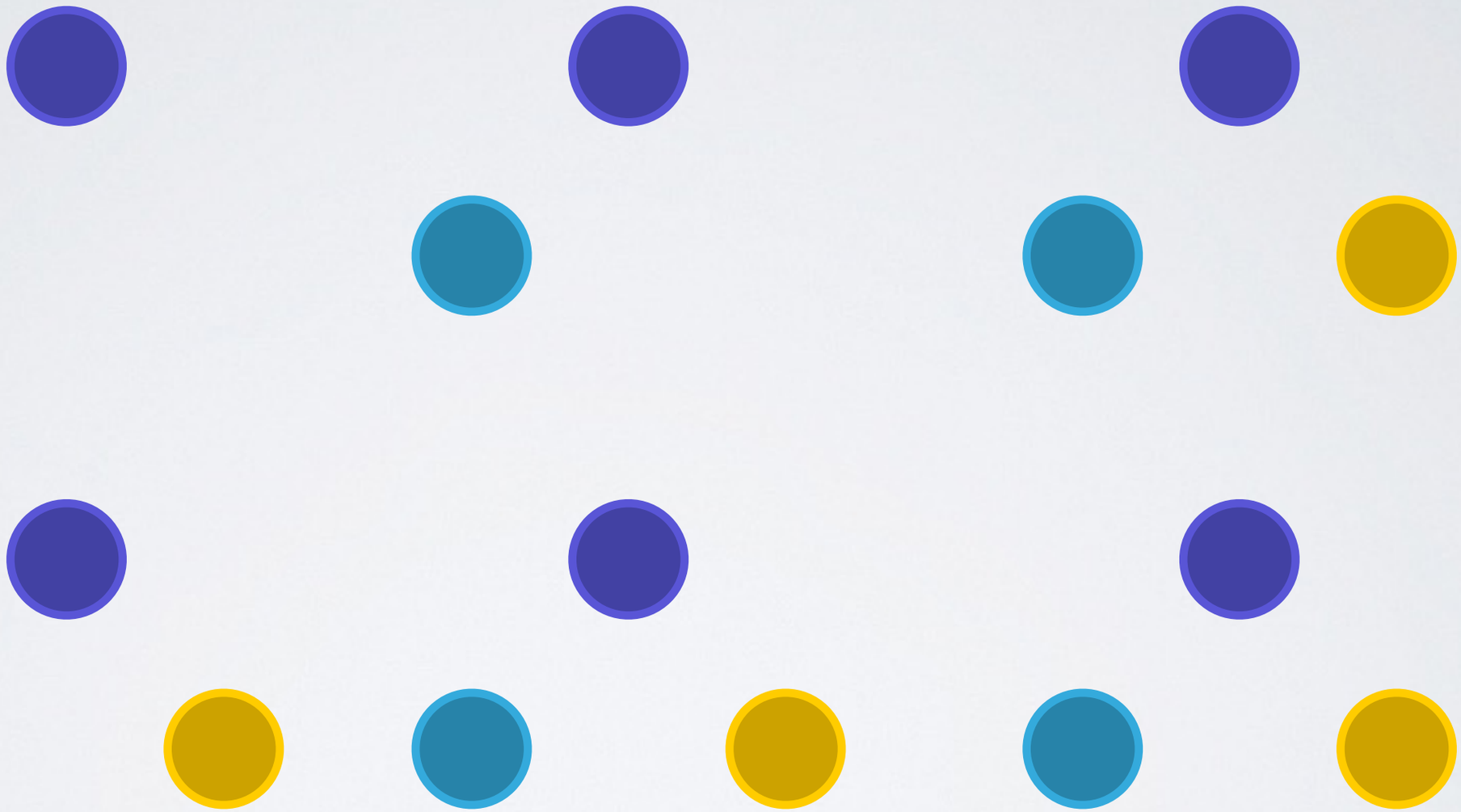
MULTIPLE MVCS



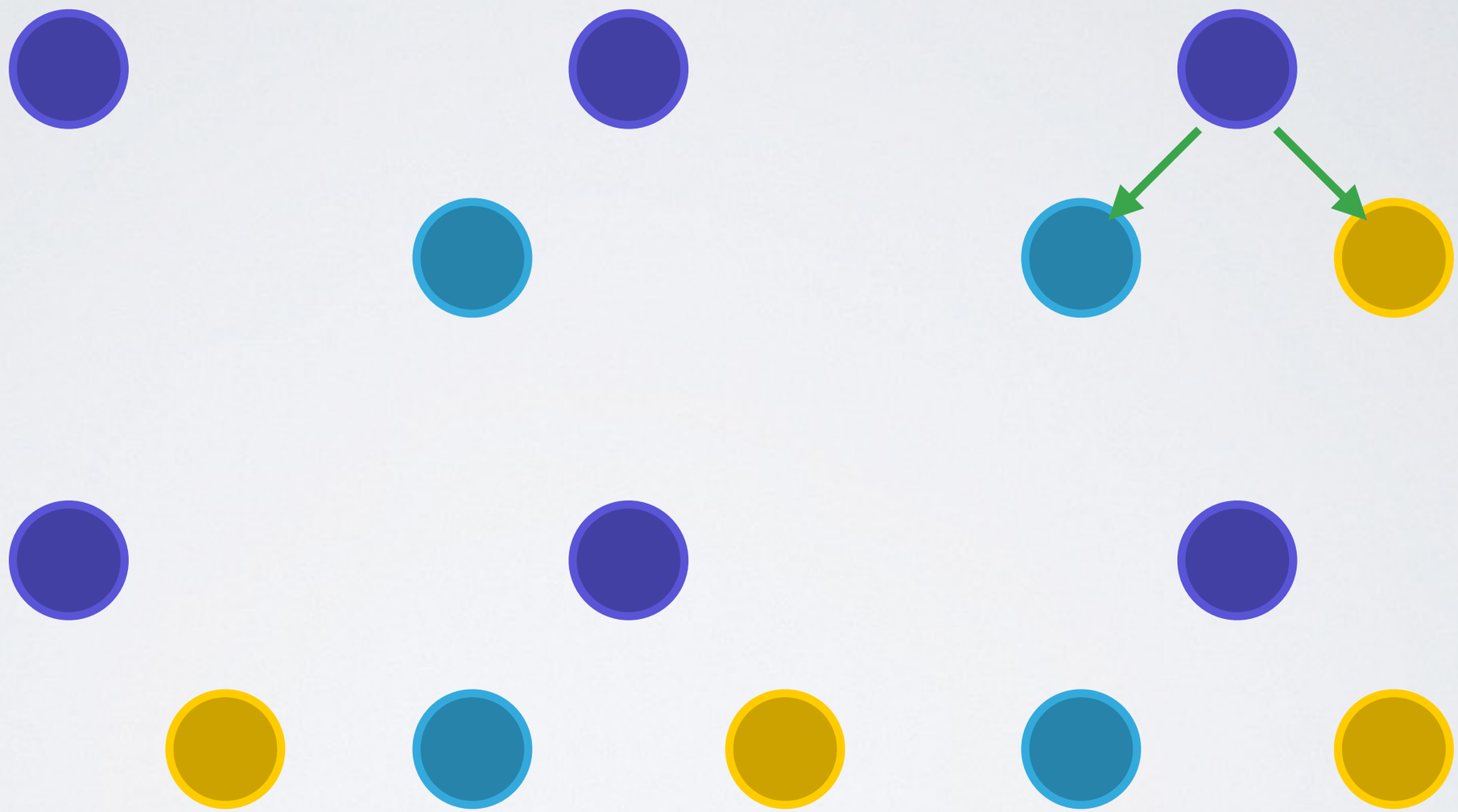
MULTIPLE MVCS



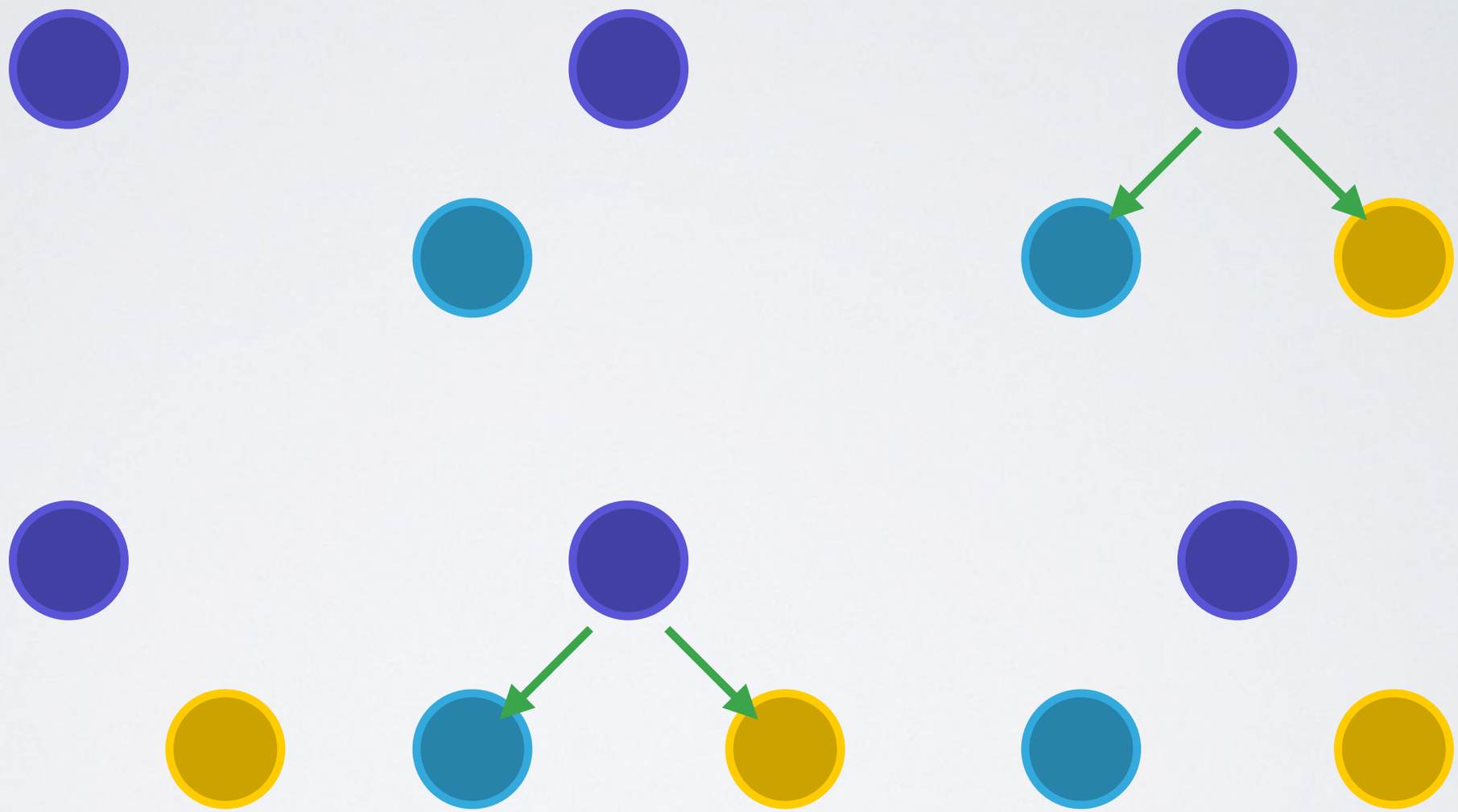
MULTIPLE MVCS



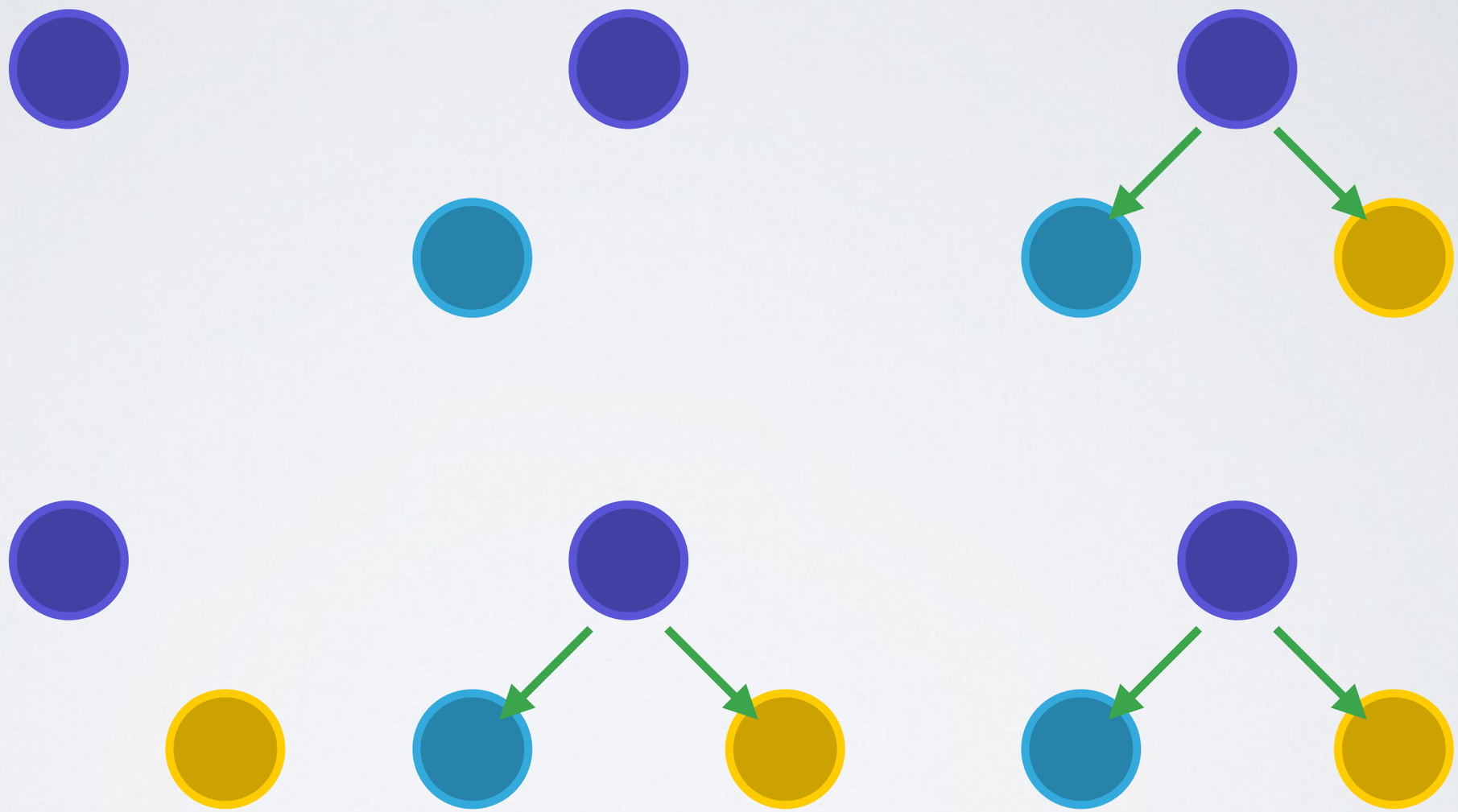
MULTIPLE MVCS



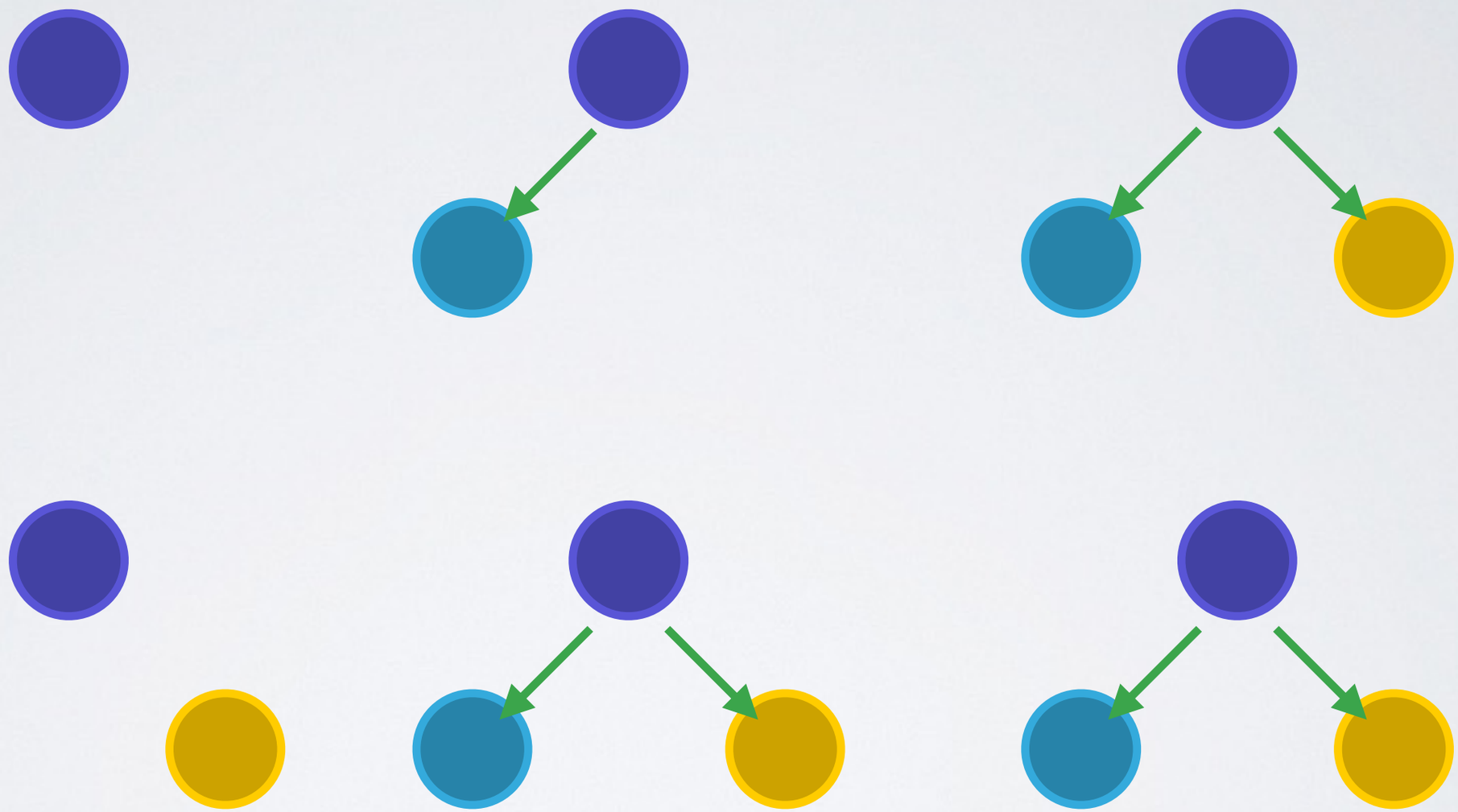
MULTIPLE MVCS



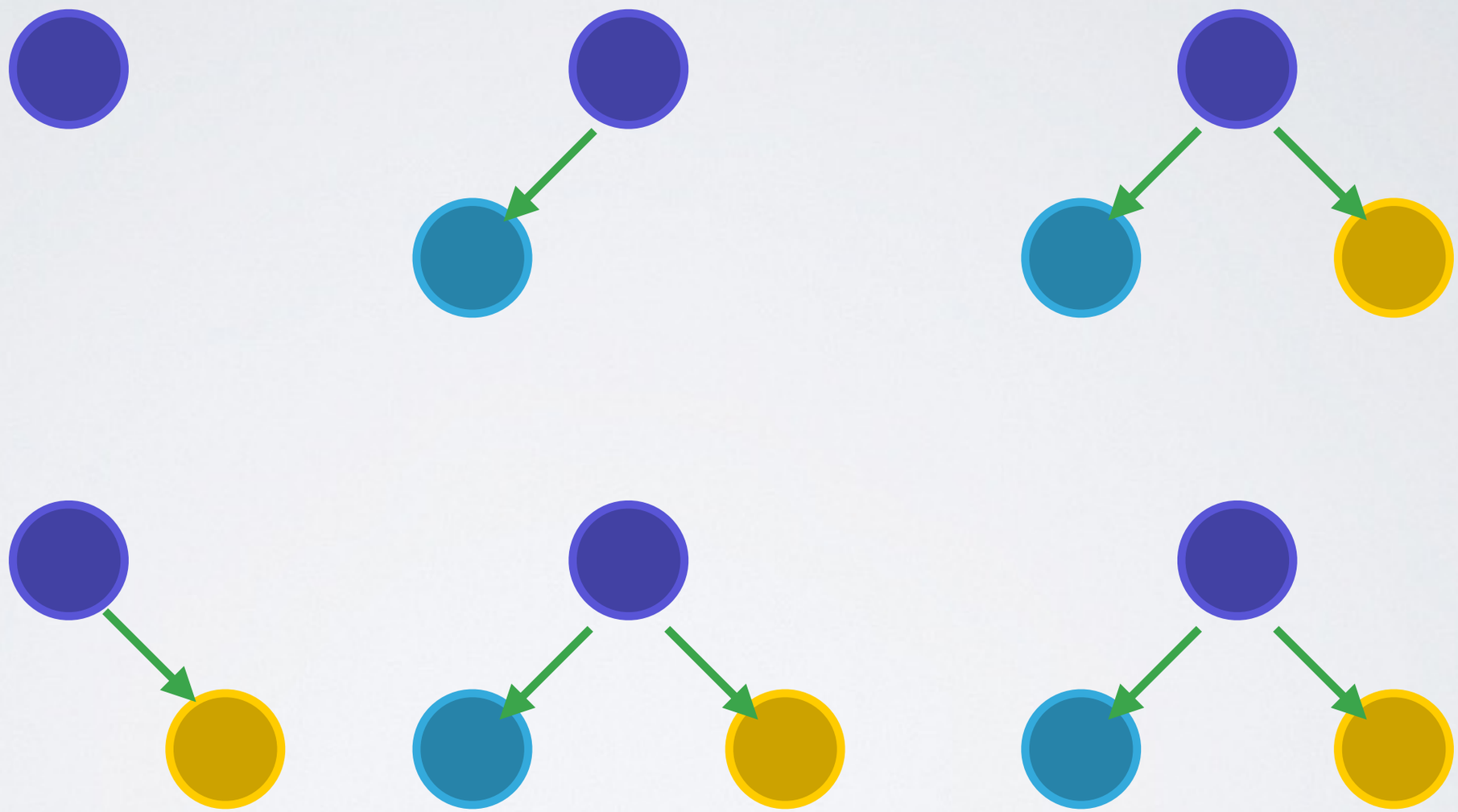
MULTIPLE MVCS



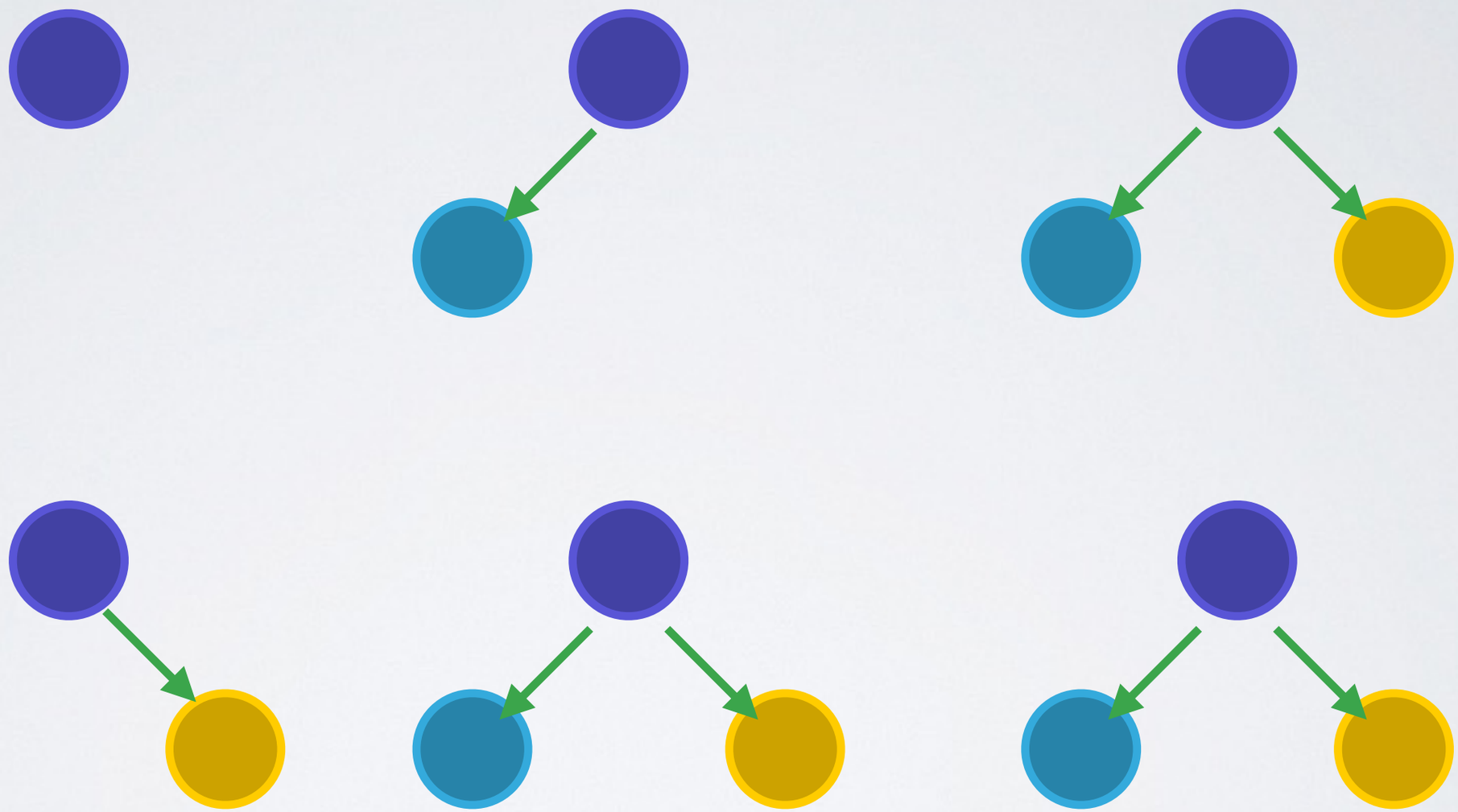
MULTIPLE MVCS



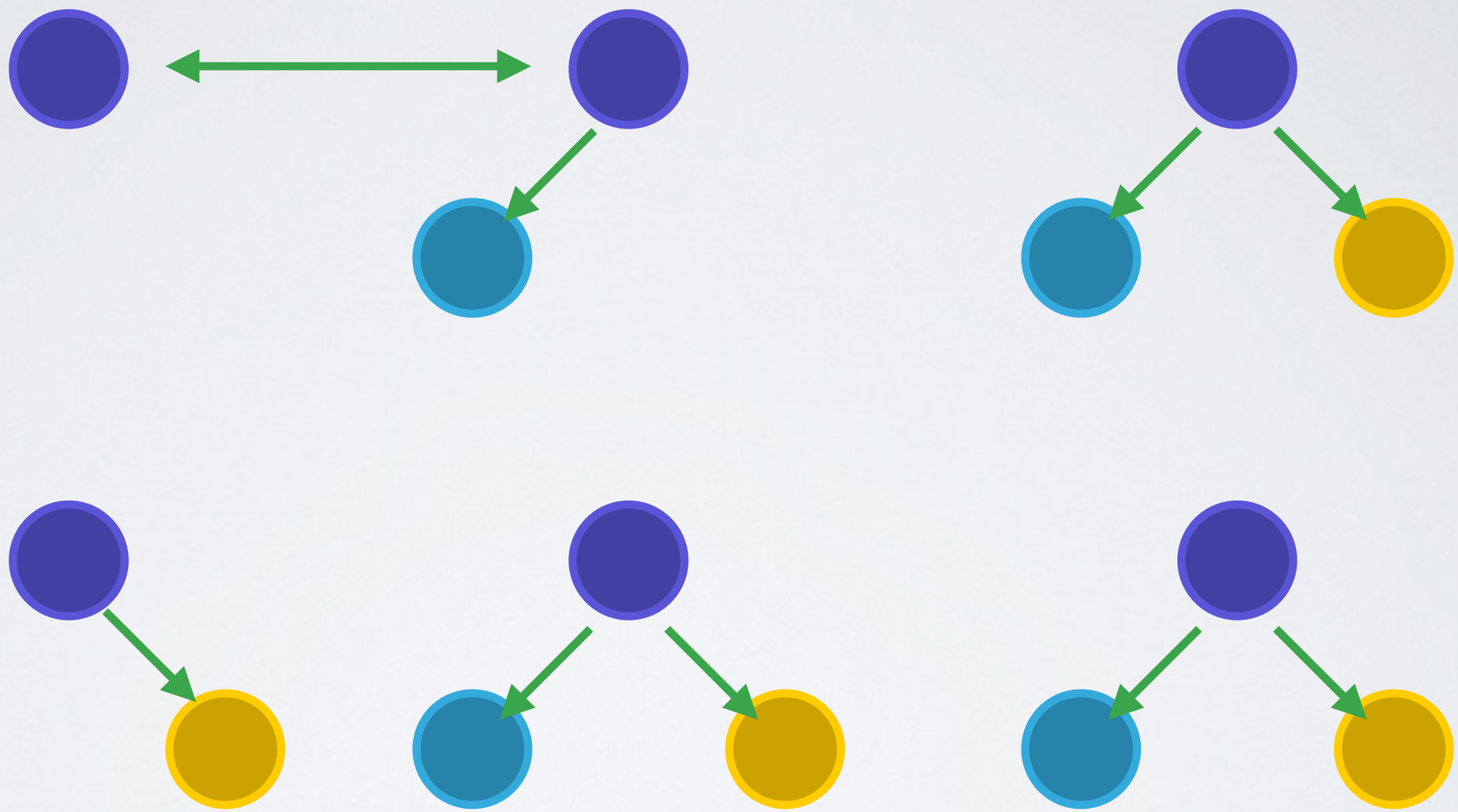
MULTIPLE MVCS



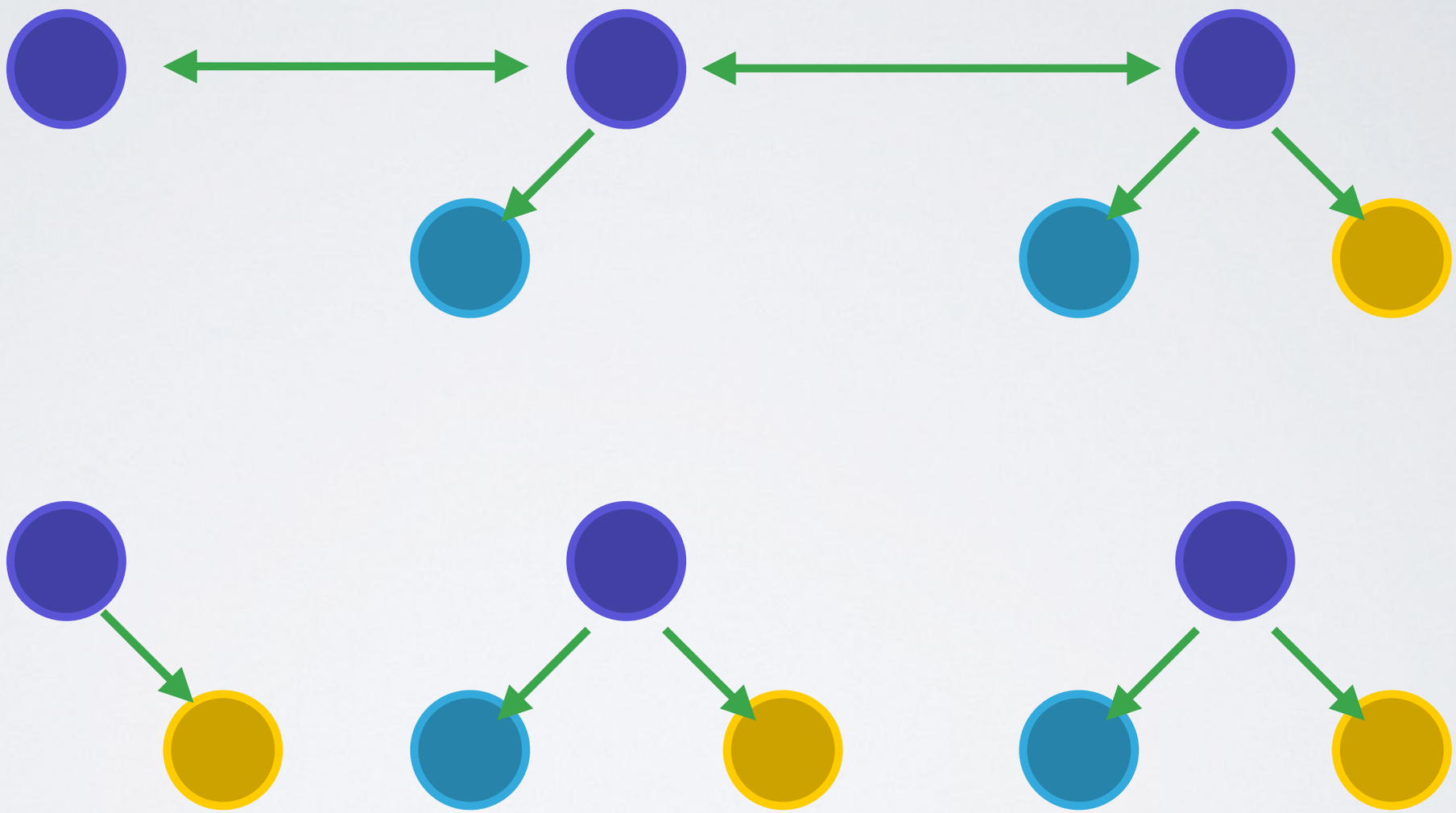
MULTIPLE MVCS



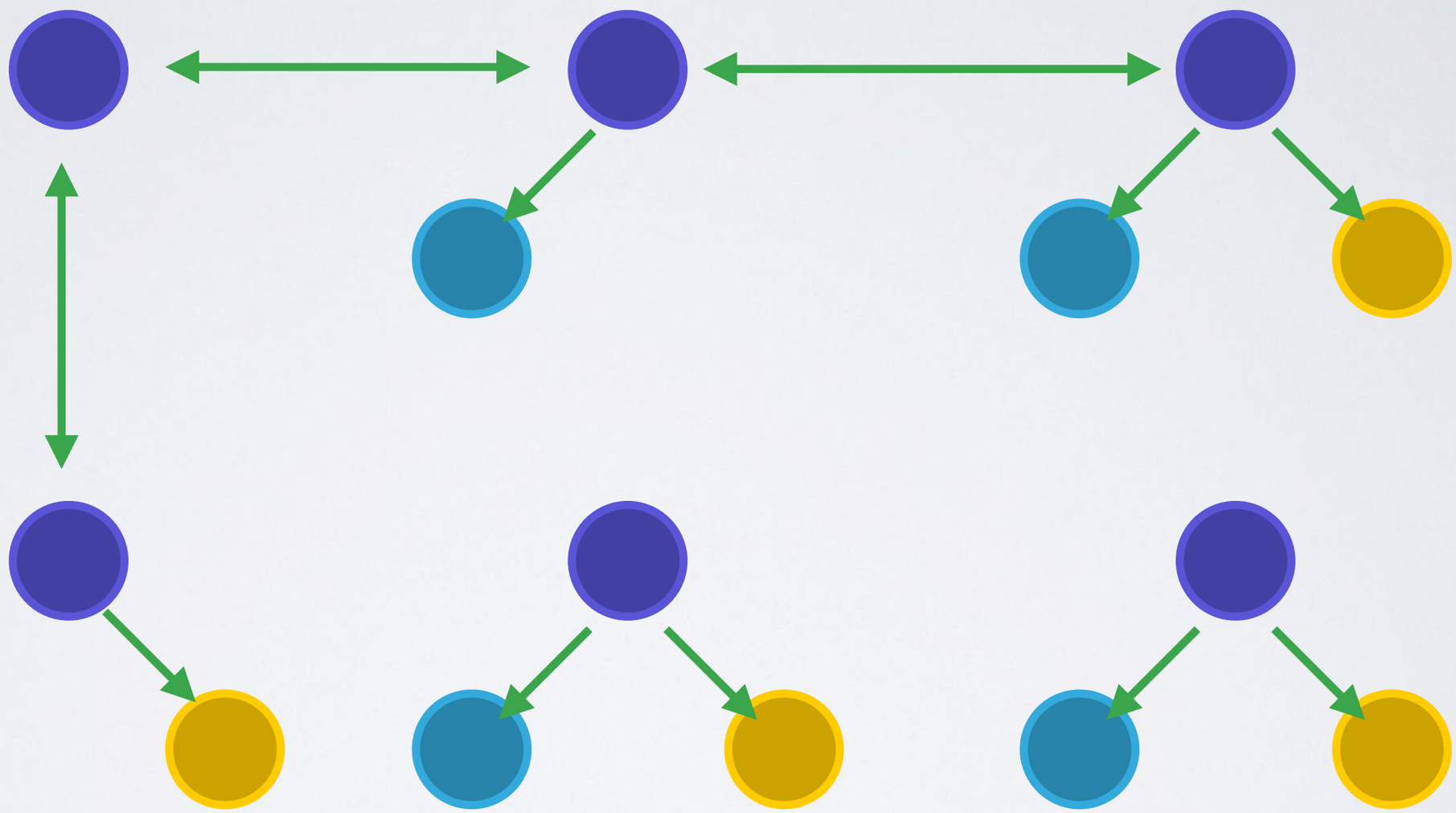
MULTIPLE MVCS



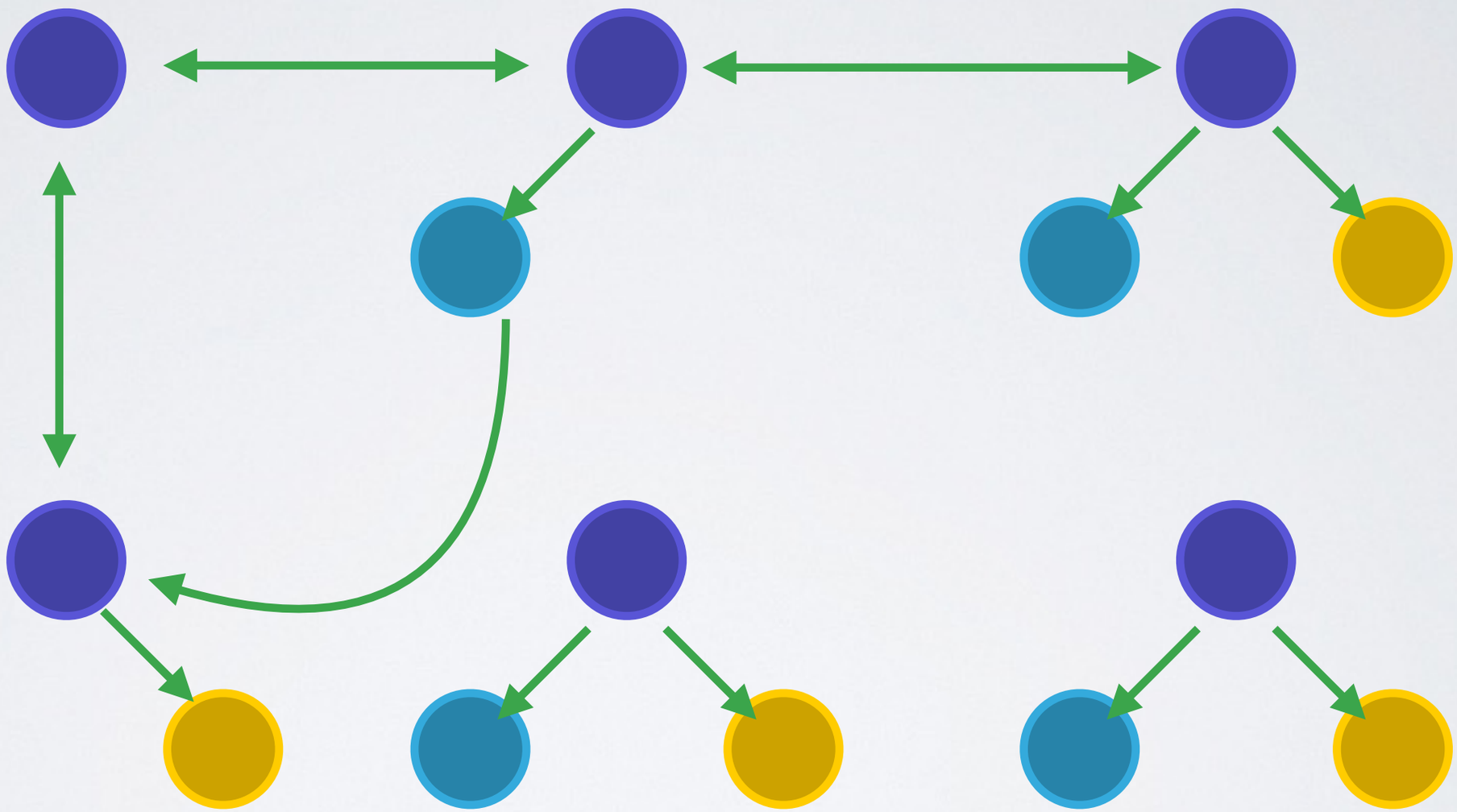
MULTIPLE MVCS



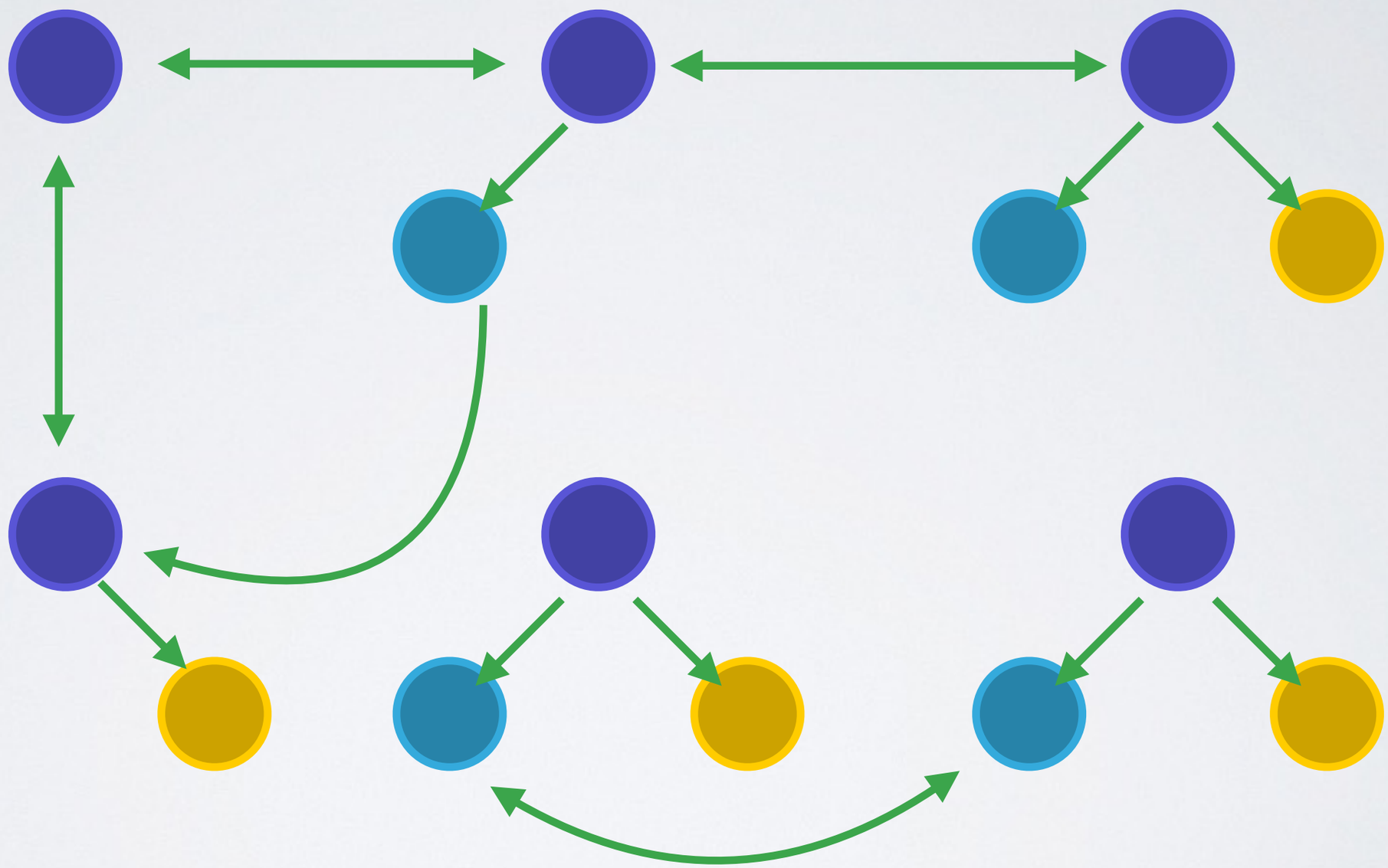
MULTIPLE MVCS



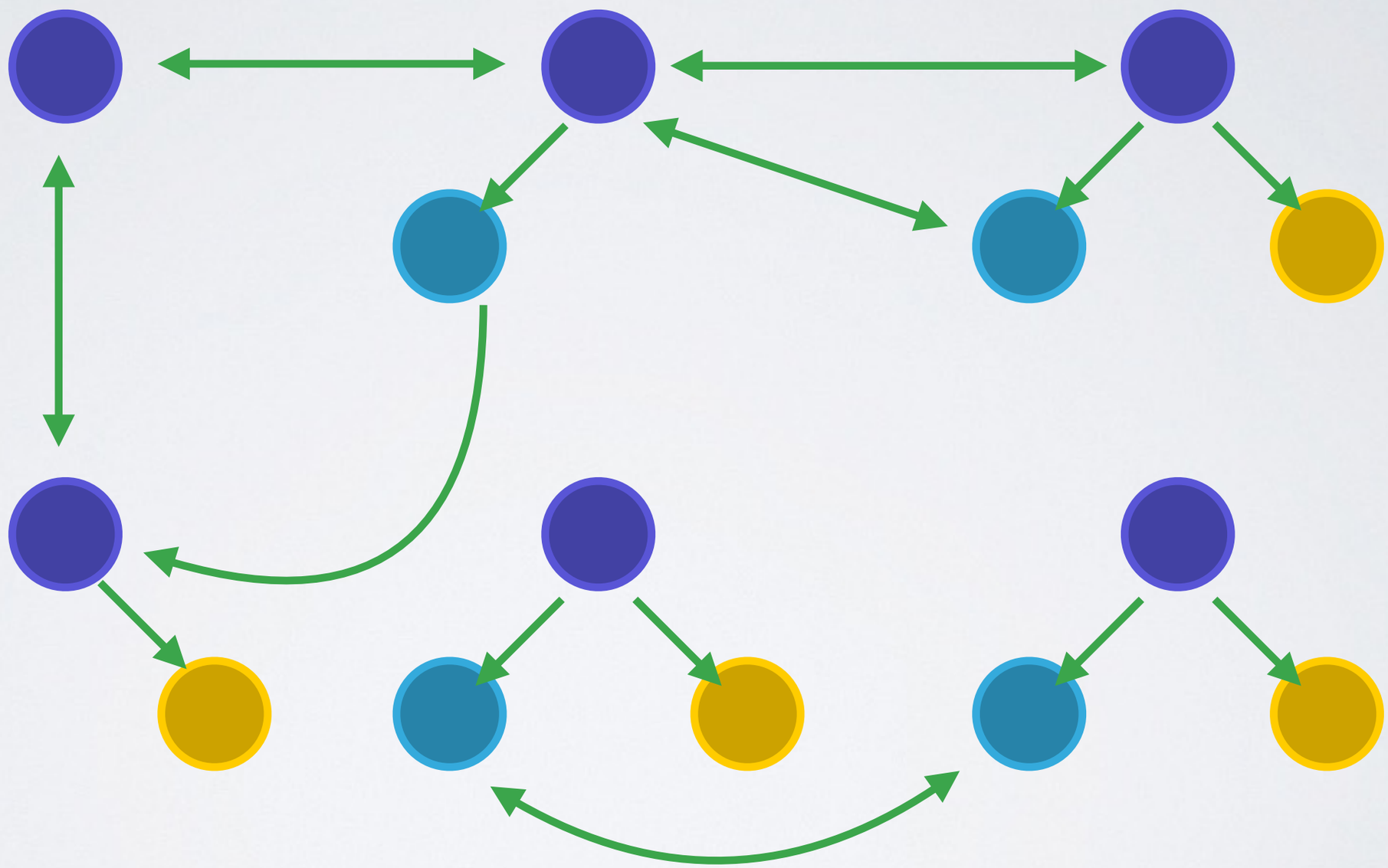
MULTIPLE MVCS



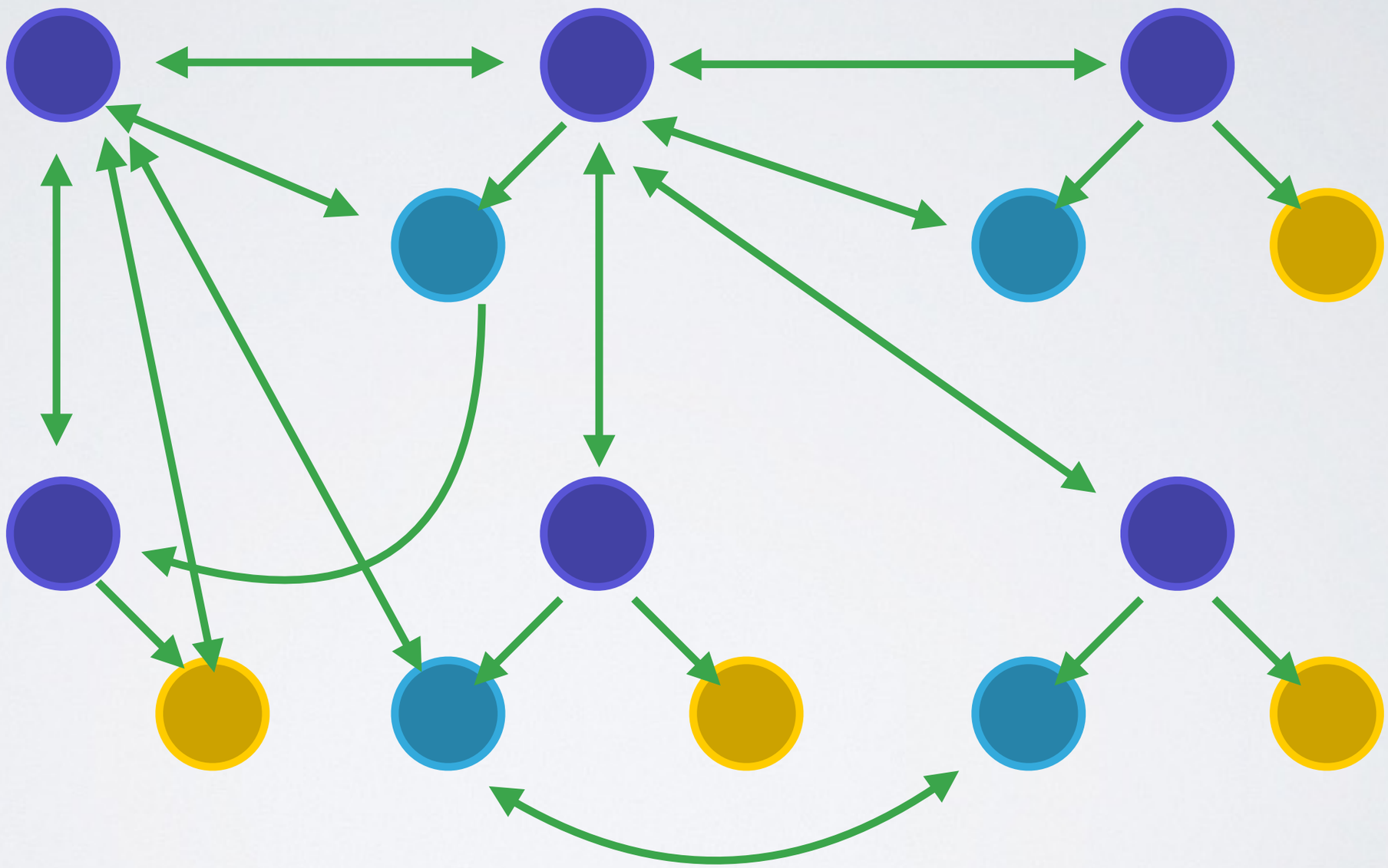
MULTIPLE MVCS



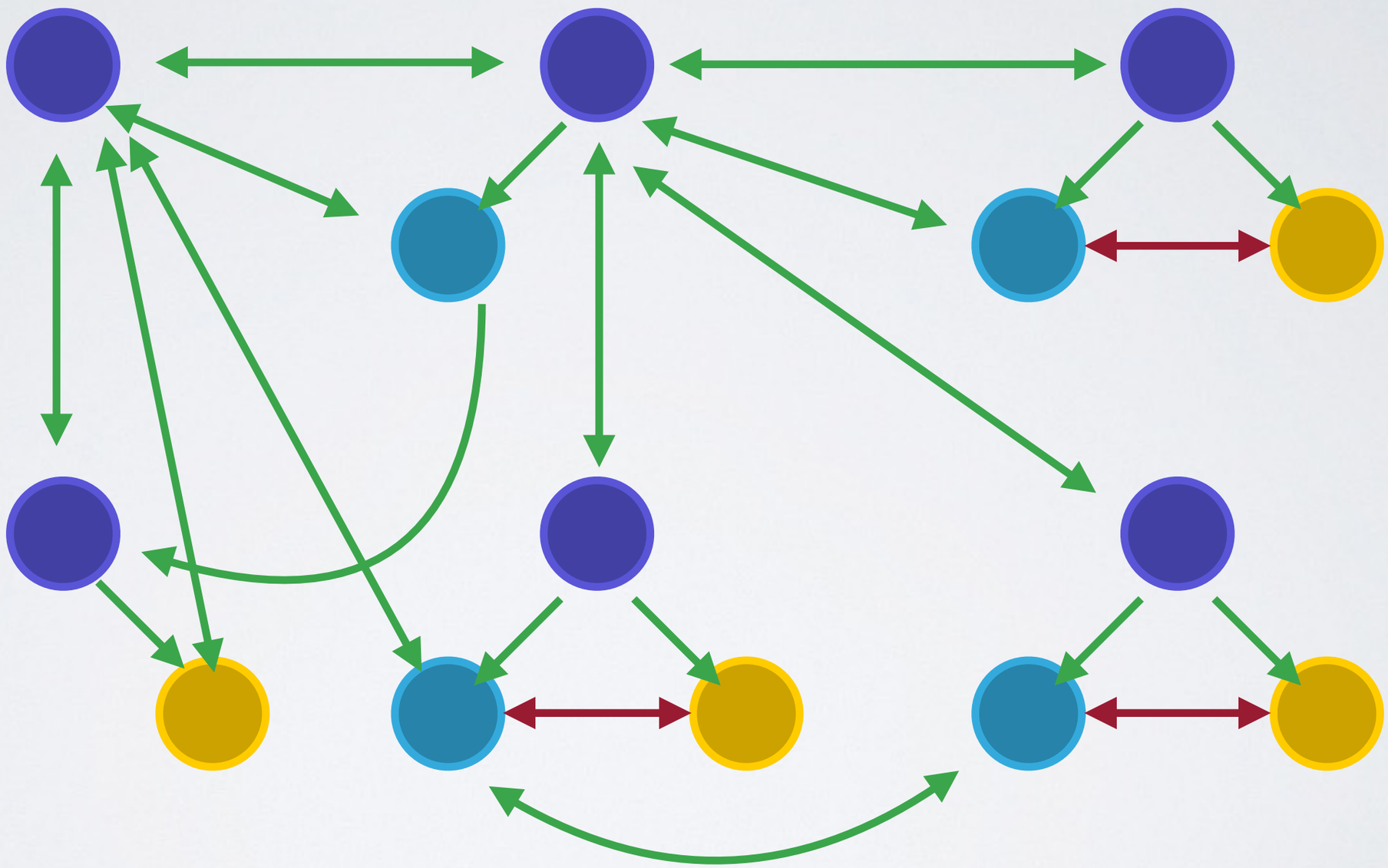
MULTIPLE MVCS



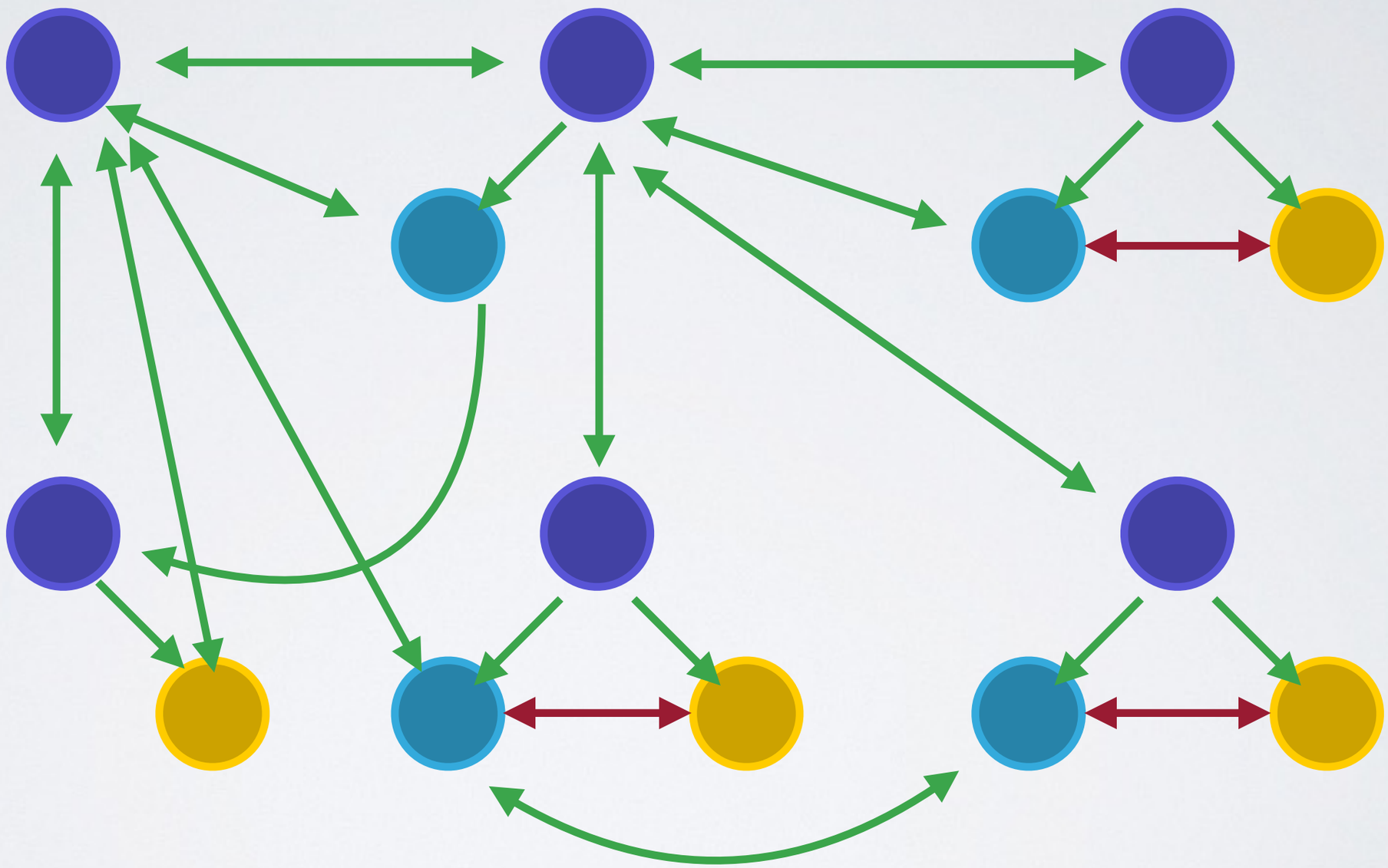
MULTIPLE MVCS



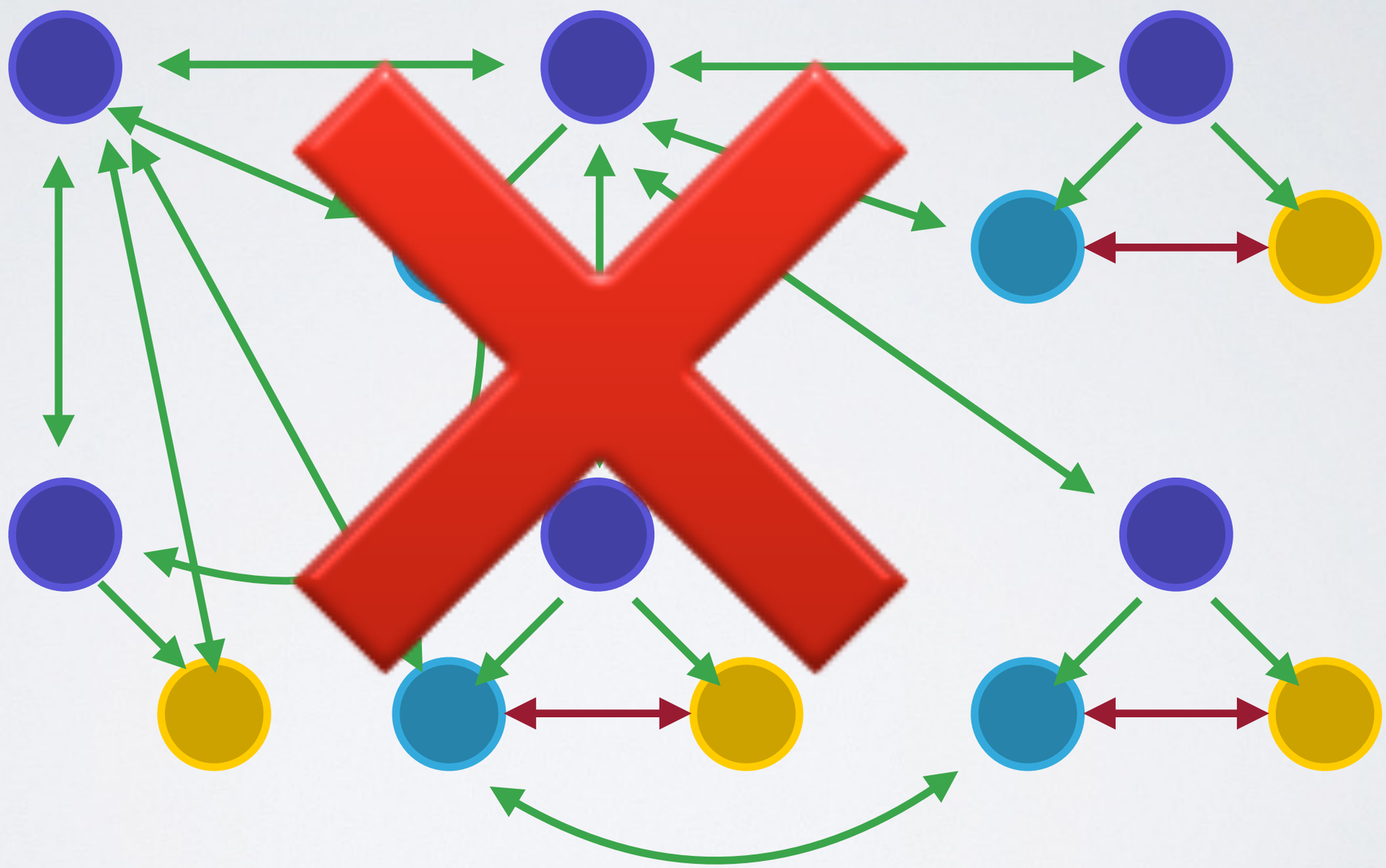
MULTIPLE MVCS



MULTIPLE MVCS FAILING



MULTIPLE MVCS FAILING



MVC (DETOUR)

martinfowler.com/eaaDev/uiArchs.html

GUI Architectures

There have been many different ways to organize the code for a rich client system. Here I discuss a selection of those that I feel have been the most influential and introduce how they relate to the patterns.

18 July 2006

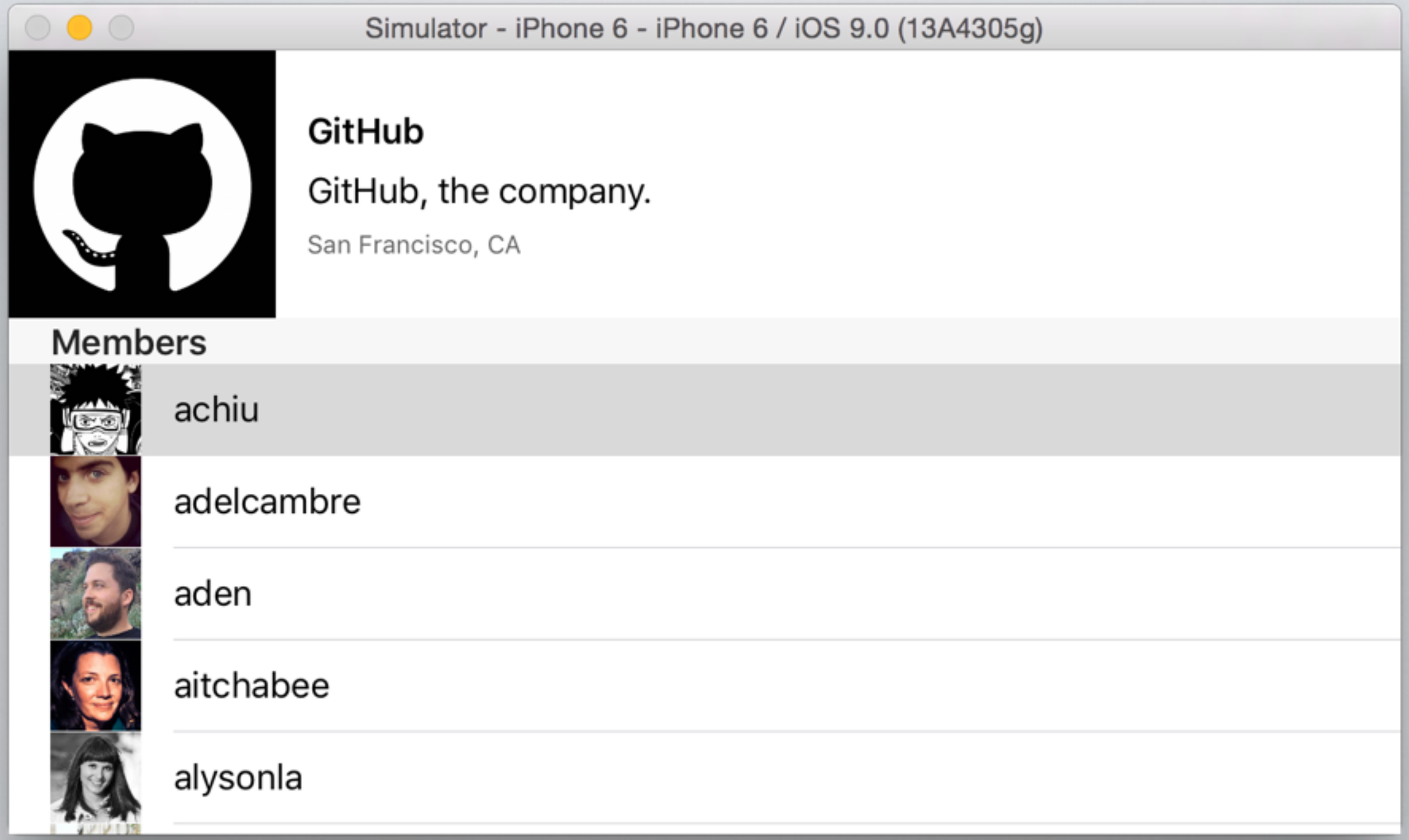
Martin Fowler

Contents

Forms and Controls
Model View Controller
VisualWorks Application Model
Model-View-Presenter (MVP)
Humble View

This is part of the Further Enterprise Application Architecture development writing that I was doing in the mid 2000's. Sadly too

LET'S BUILD THIS USING MVC



VIEW

- Let's prototype the design in Storyboards first.
- Get a better feel for table views and auto-layouts

TABLE VIEW (STATIC)

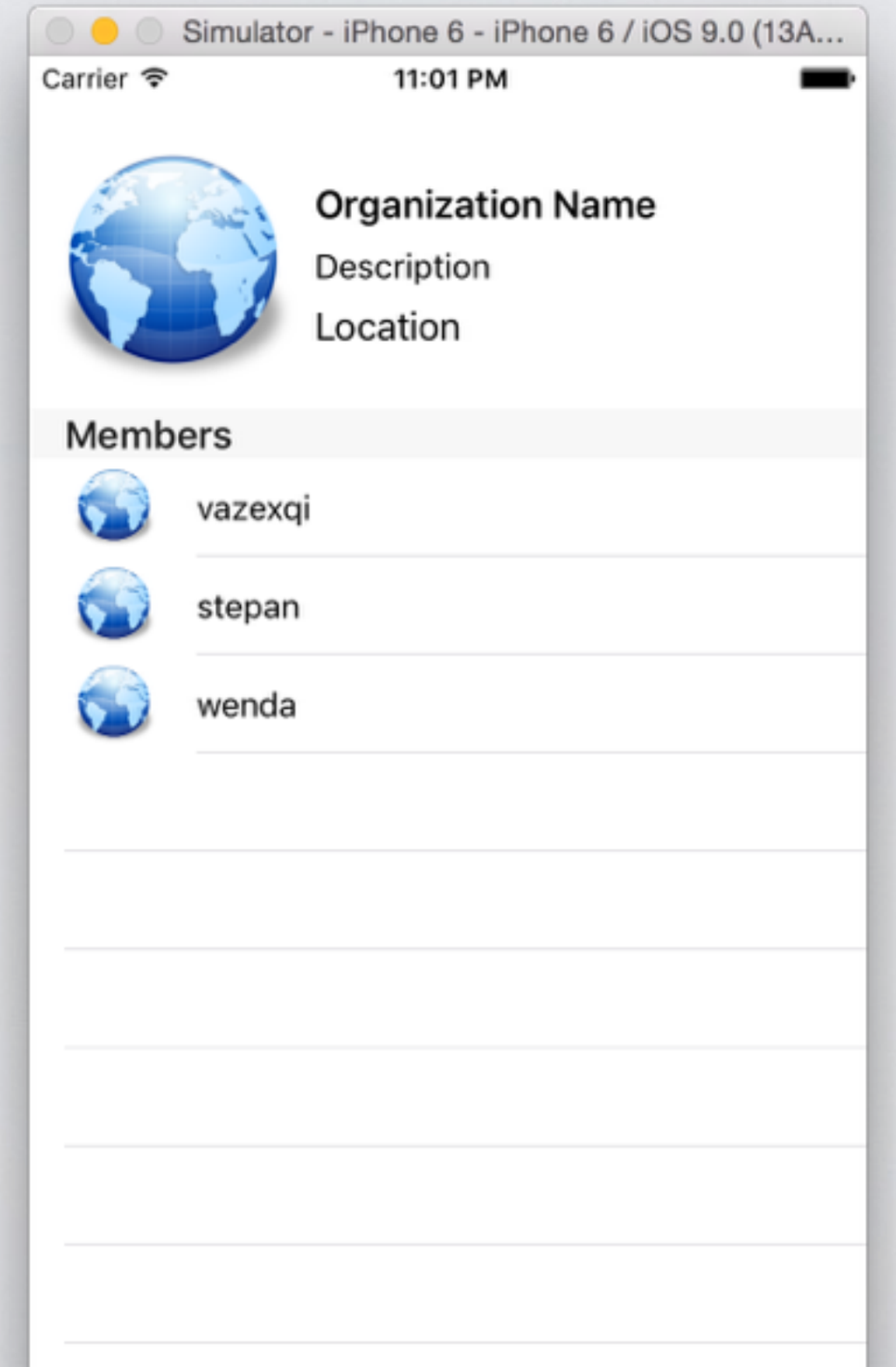


TABLE VIEW (STATIC)

TABLE VIEW (STATIC)

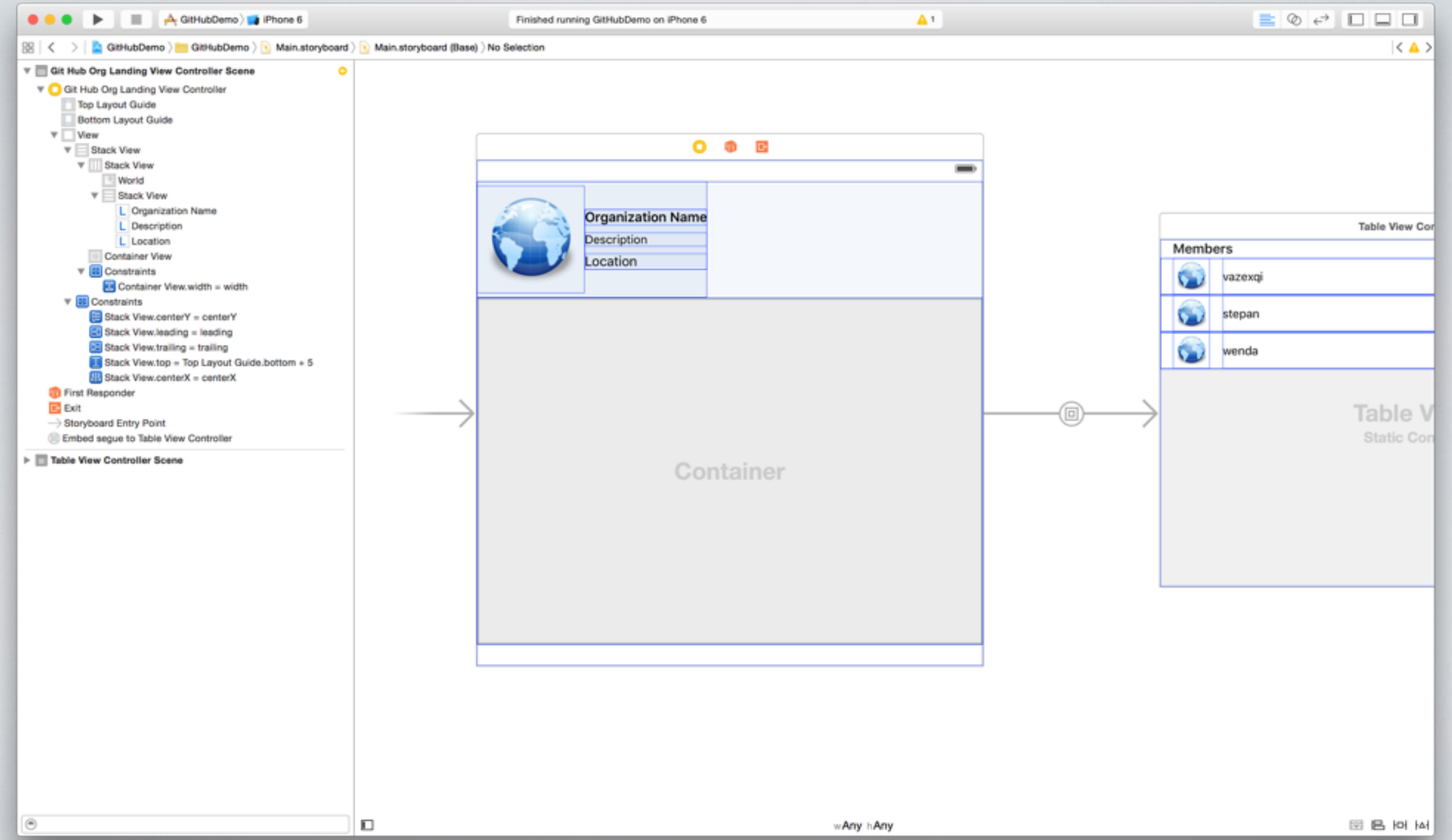


TABLE VIEW (STATIC)

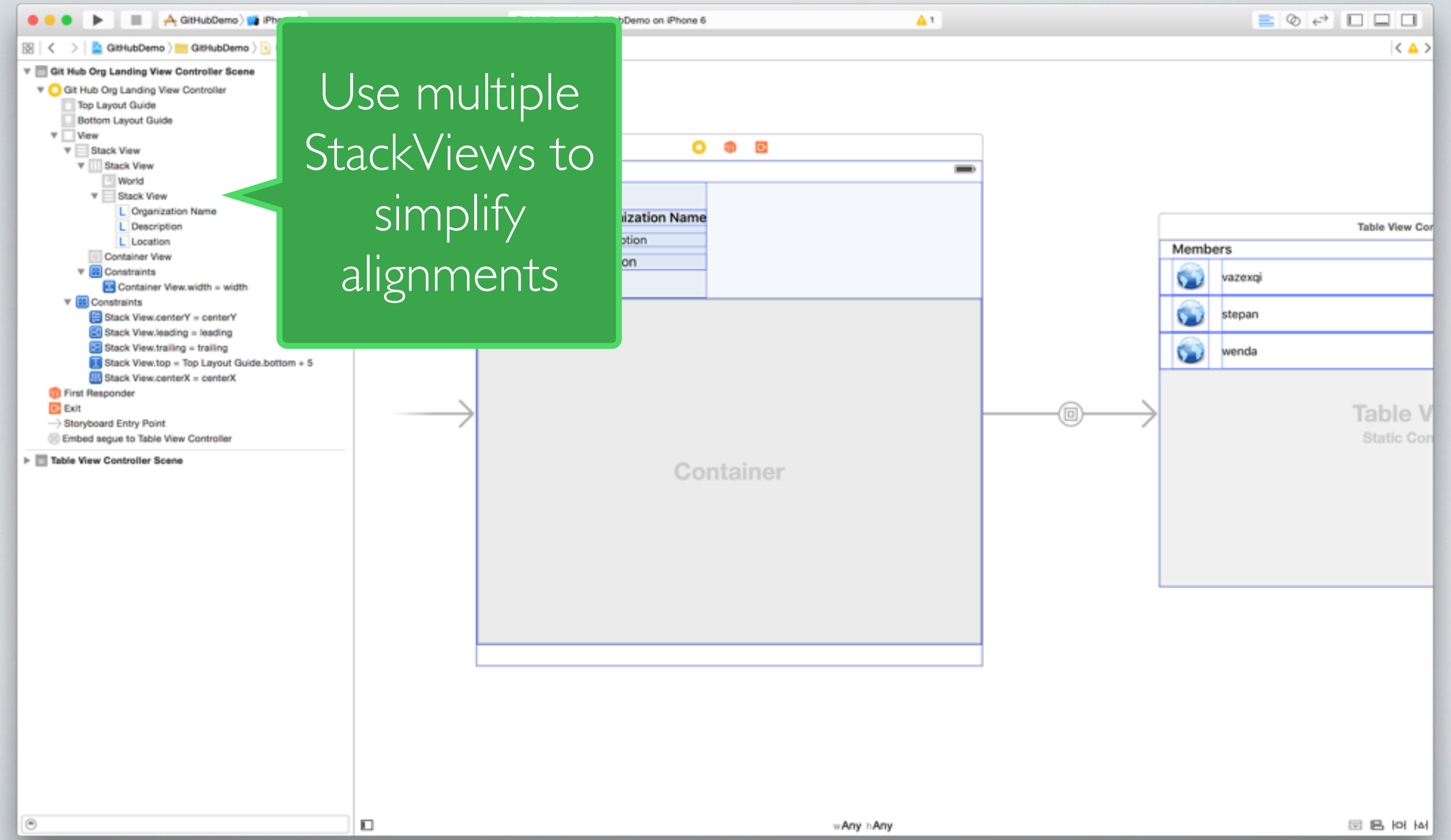


TABLE VIEW (STATIC)

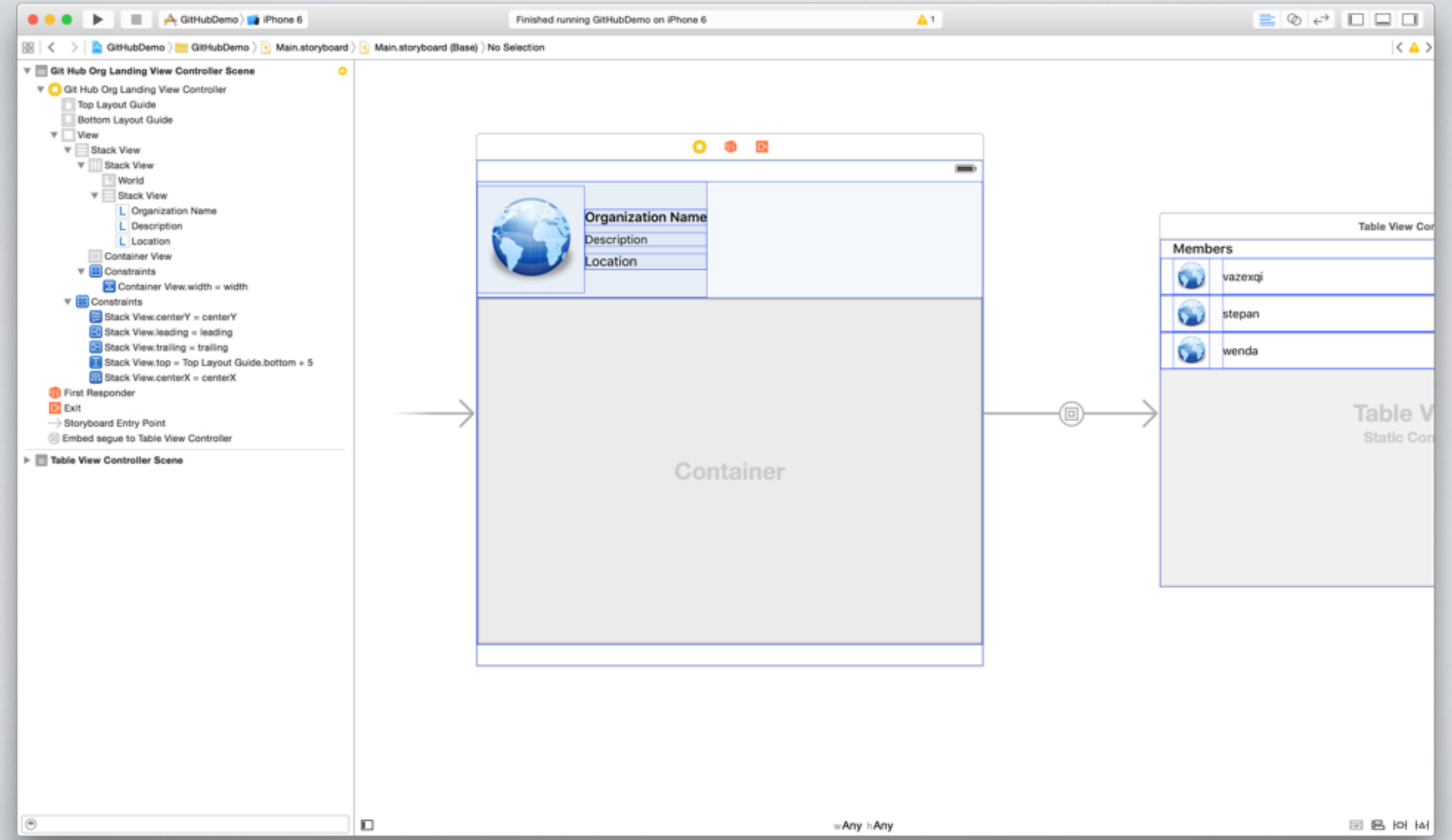


TABLE VIEW (STATIC)

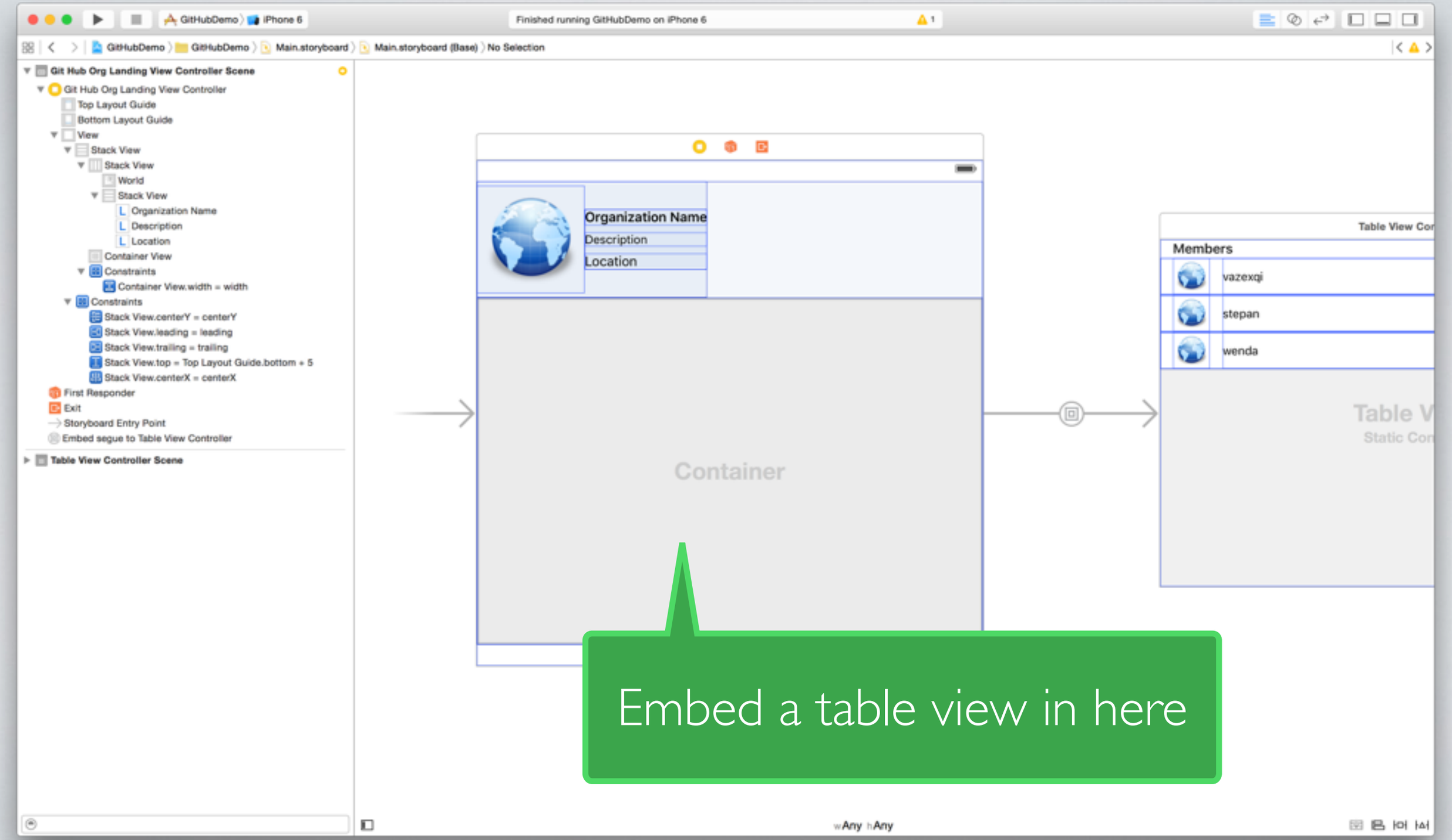


TABLE VIEW (STATIC)

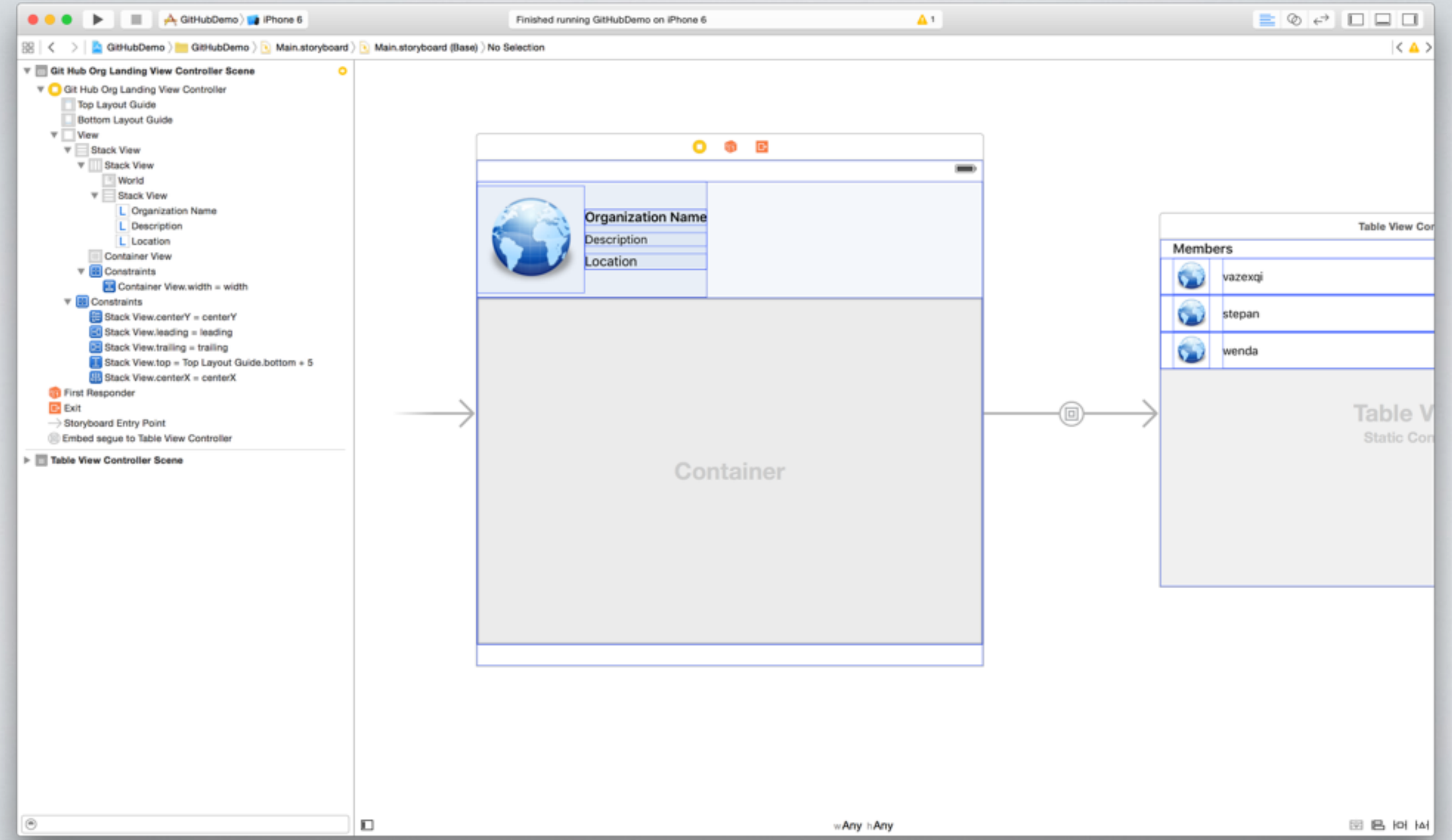
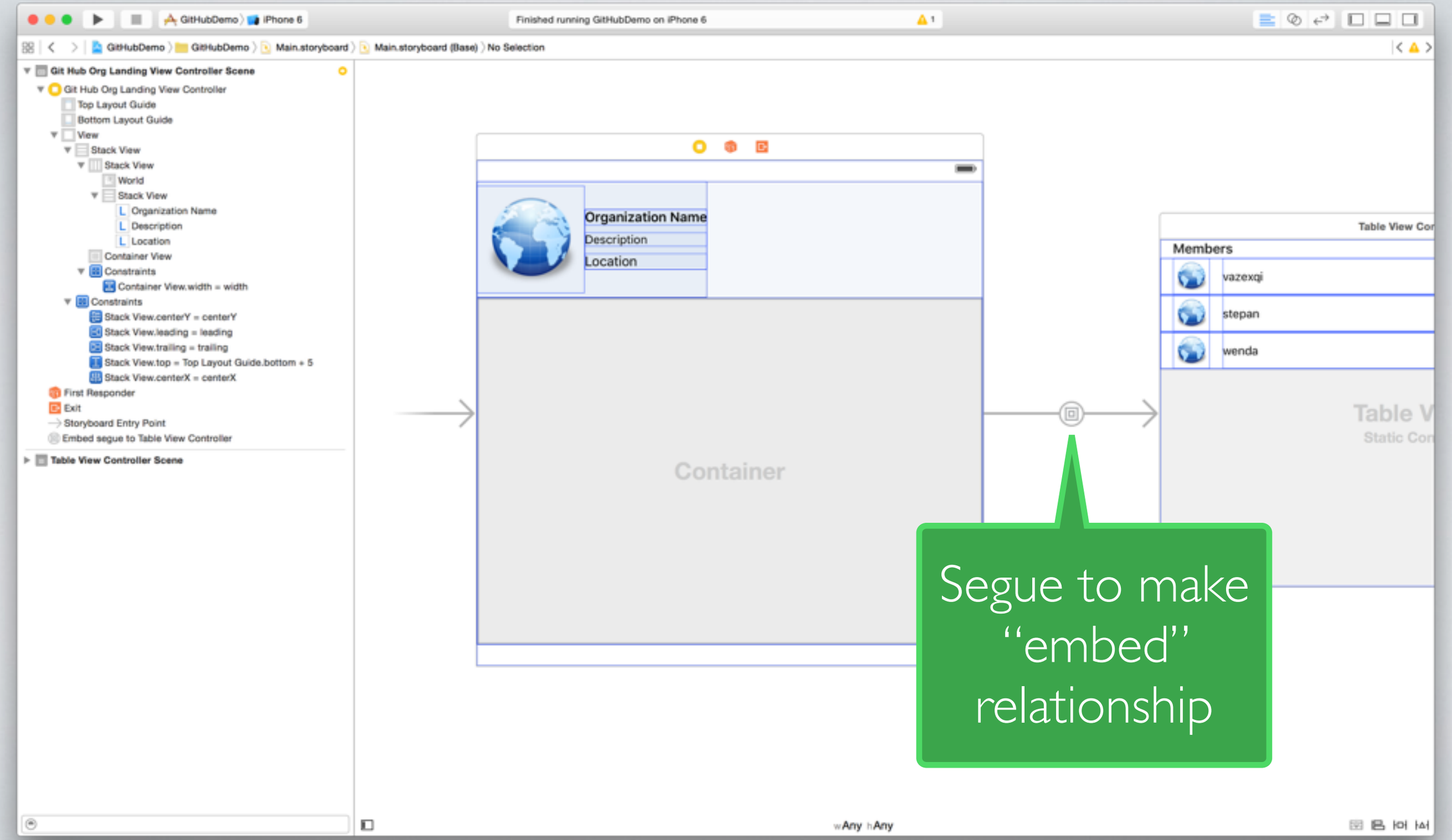


TABLE VIEW (STATIC)



HANDS-ON

MODEL

We need some way to get the data from GitHub

GITHUB API

developer.github.com/v3/orgs/#get-an-organizati

Get an organization

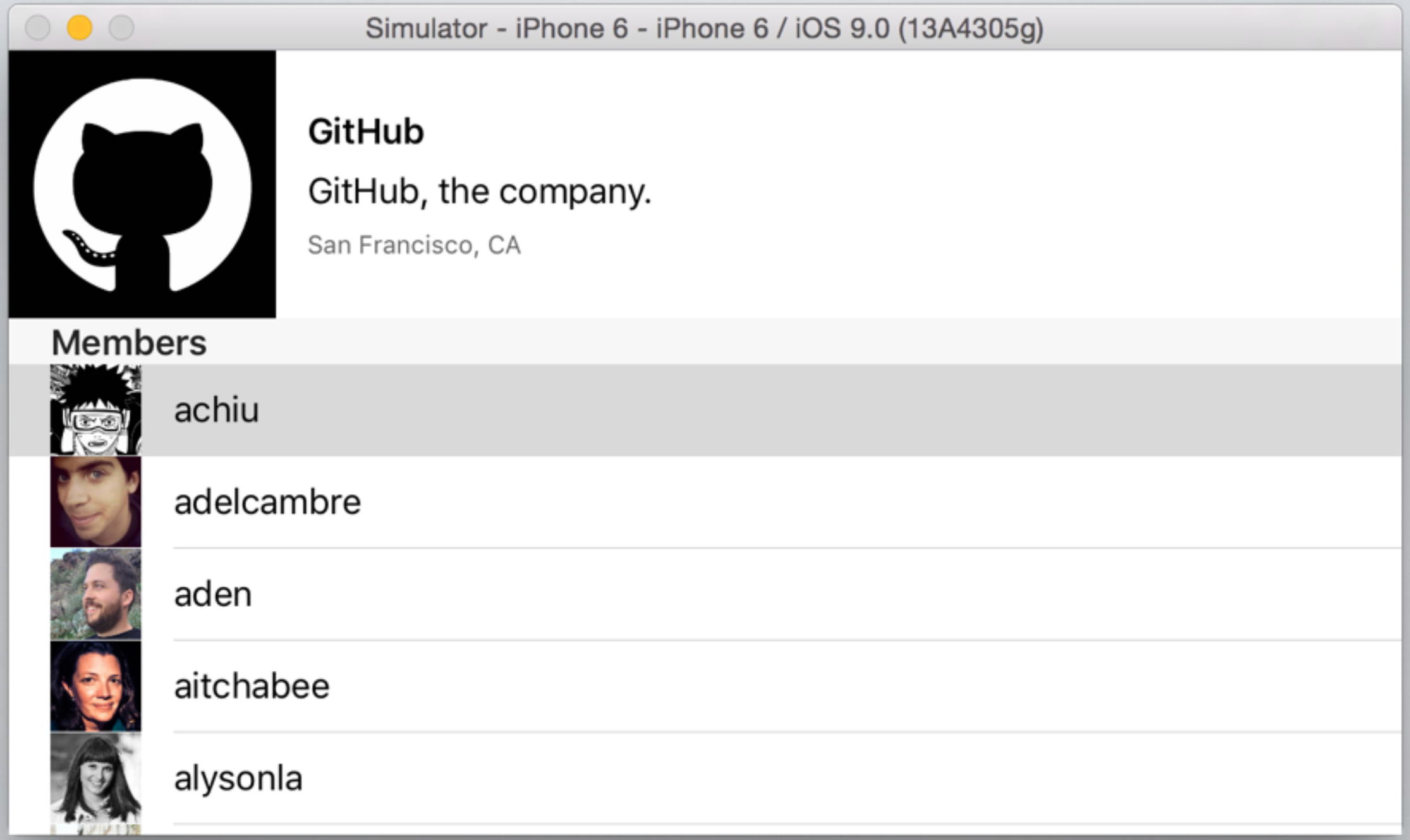
```
GET /orgs/:org
```

Response

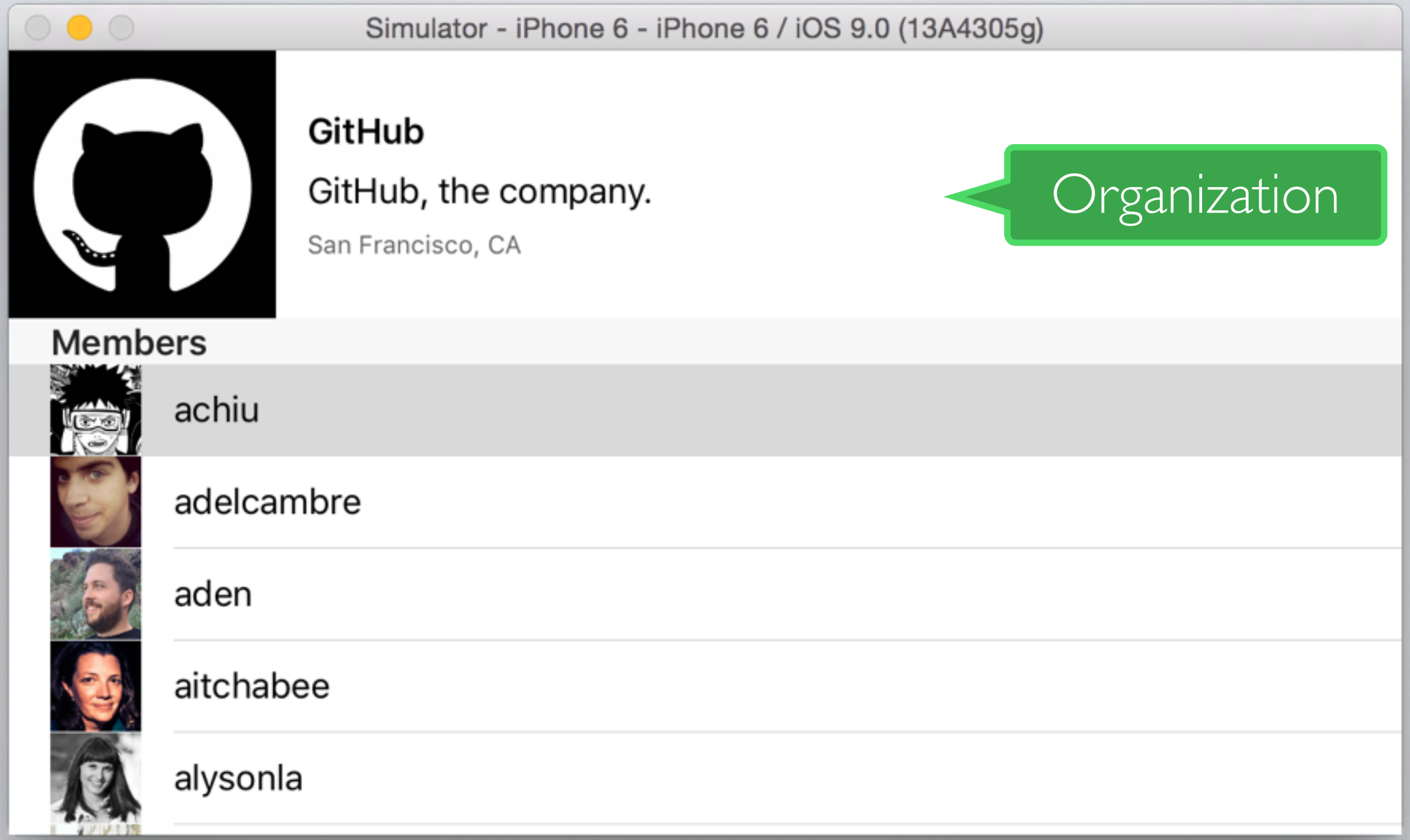
```
Status: 200 OK
X-RateLimit-Limit: 5000
X-RateLimit-Remaining: 4999
```


```
{
  "login": "github",
  "id": 1,
  "url": "https://api.github.com/orgs/github",
  "avatar_url": "https://github.com/images/error/octocat_happy.gif",
  "description": "A great organization",
  "name": "github",
  "company": "GitHub",
  "blog": "https://github.com/blog",
  "location": "San Francisco",
  "email": "octocat@github.com",
```

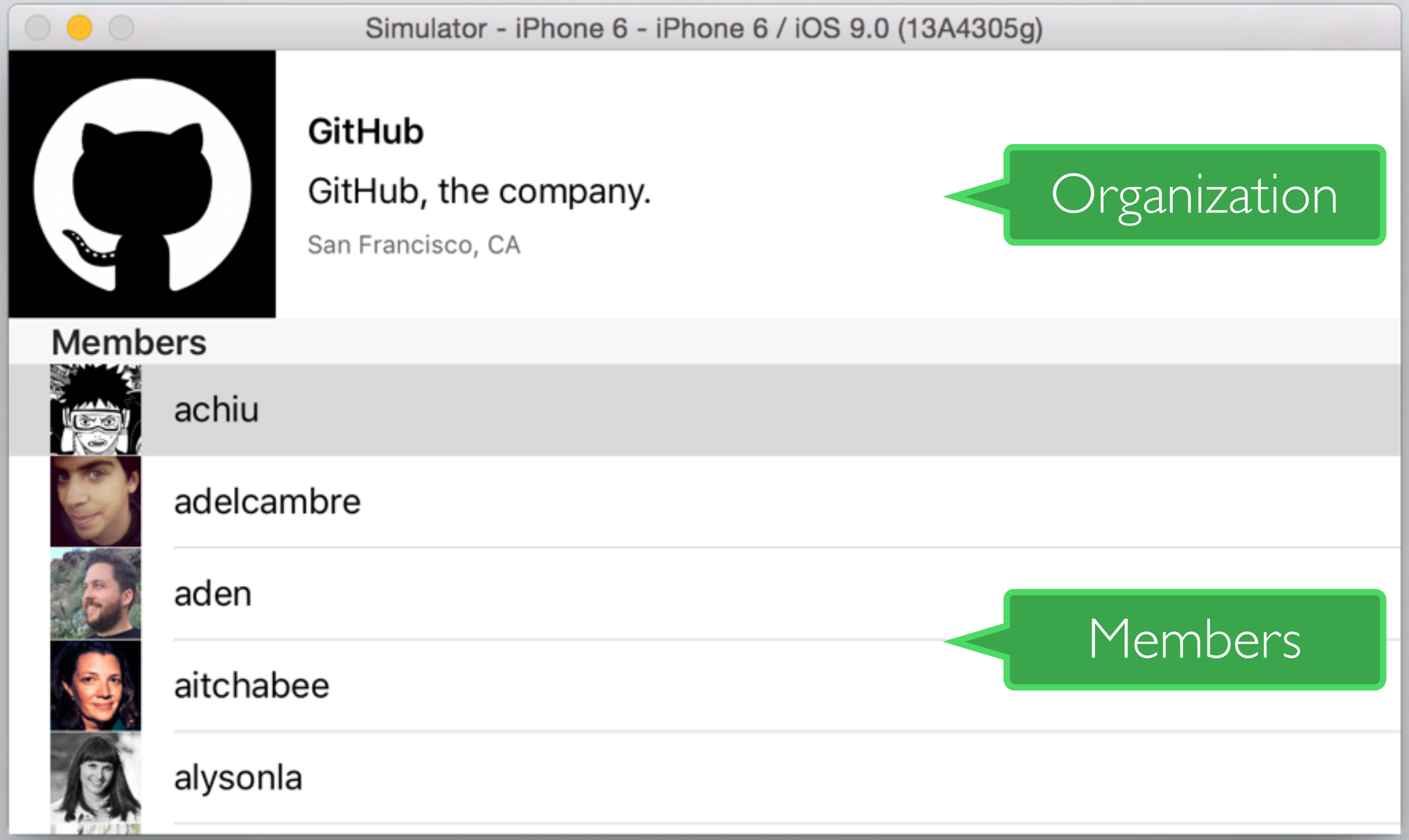
GITHUB API


Simulator - iPhone 6 - iPhone 6 / iOS 9.0 (13A4305g)
GitHub
GitHub, the company.
San Francisco, CA
Members
achiu
adelcambre
aden
aitchabee
alysonla

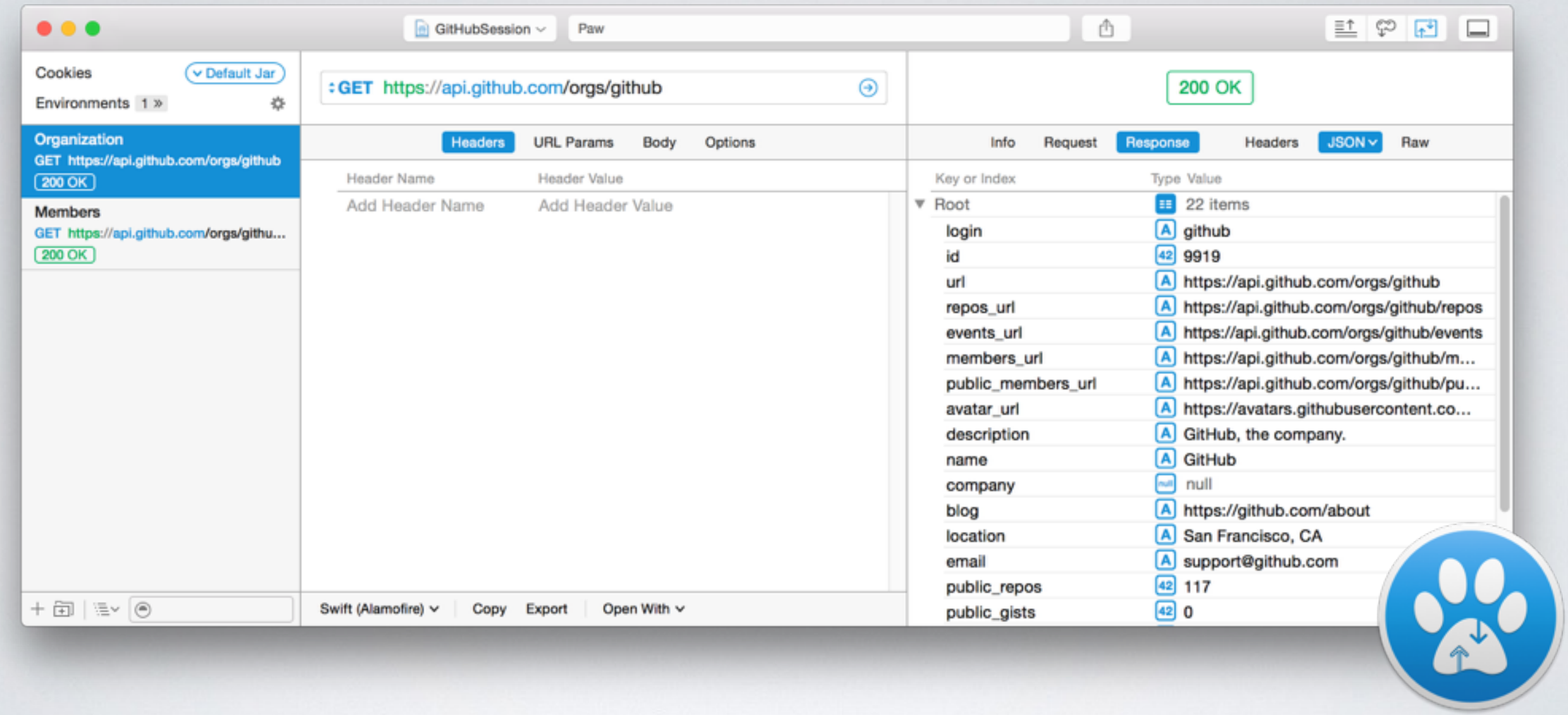
GITHUB API


Simulator - iPhone 6 - iPhone 6 / iOS 9.0 (13A4305g)
GitHub
GitHub, the company.
San Francisco, CA
Organization
Members
achiu
adelcambre
aden
aitchabee
alysonla

GITHUB API


Simulator - iPhone 6 - iPhone 6 / iOS 9.0 (13A4305g)
GitHub
GitHub, the company.
San Francisco, CA
Organization
Members
achiu
adelcambre
aden
aitchabee
alysonla
Members

GITHUB API


GitHubSession
Paw
Cookies
Default Jar
Environments 1
Organization
GET https://api.github.com/orgs/github
200 OK
Members
GET https://api.github.com/orgs/githu...
200 OK
GET https://api.github.com/orgs/github
200 OK
Headers
URL Params
Body
Options
Header Name
Header Value
Add Header Name
Add Header Value
Info
Request
Response
Headers
JSON
Raw
Key or Index
Type Value
Root
22 items
login
github
id
9919
url
https://api.github.com/orgs/github
repos_url
https://api.github.com/orgs/github/repos
events_url
https://api.github.com/orgs/github/events
members_url
https://api.github.com/orgs/github/m...
public_members_url
https://api.github.com/orgs/github/pu...
avatar_url
https://avatars.githubusercontent.co...
description
GitHub, the company.
name
GitHub
company
null
blog
https://github.com/about
location
San Francisco, CA
email
support@github.com
public_repos
117
public_gists
0
Swift (Alamofire)
Copy
Export
Open With

GITHUB API

GitHubSession | Paw

Cookies | Default Jar
Environments 1

Organization
GET https://api.github.com/orgs/github
200 OK

Members
GET https://api.github.com/orgs/githu...
200 OK

GET https://api.github.com/orgs/github

Organization

200 OK

Info | Request | Response | Headers | JSON | Raw

Key or Index	Type	Value
Root		22 items
login	A	github
id	42	9919
url	A	https://api.github.com/orgs/github
repos_url	A	https://api.github.com/orgs/github/repos
events_url	A	https://api.github.com/orgs/github/events
members_url	A	https://api.github.com/orgs/github/m...
public_members_url	A	https://api.github.com/orgs/github/pu...
avatar_url	A	https://avatars.githubusercontent.co...
description	A	GitHub, the company.
name	A	GitHub
company	null	null
blog	A	https://github.com/about
location	A	San Francisco, CA
email	A	support@github.com
public_repos	42	117
public_gists	42	0

Swift (Alamofire) | Copy | Export | Open With

GITHUB API

GitHubSession | Paw

Cookies | Default Jar
Environments 1

Organization
GET https://api.github.com/orgs/github
200 OK

Members
GET https://api.github.com/orgs/githu...
200 OK

GET https://api.github.com/orgs/github/members

Headers | URL Params | Body | Options

Timeout: No timeout second(s)

Redirections: Follow Redirects
Redirect Authorization
Redirect Original Method

Cookies: Automatically Send Cookies
Store Received Cookies

Authorization: Set HTTP Basic Auth
Set OAuth 1 | Set OAuth 2
Set Amazon S3 Signature

Swift (Alamofire) | Copy | Export | Open With

200 OK

Info | Request | Response | Headers | JSON | Raw

Key or Index	Type	Value
▾ Index 0	17 items	
login	A	achiu
id	42	24772
avatar_url	A	https://avatars.githubusercontent.co...
gravatar_id	A	
url	A	https://api.github.com/users/achiu
html_url	A	https://github.com/achiu
followers_url	A	https://api.github.com/users/achiu/fol...
following_url	A	https://api.github.com/users/achiu/fol...
gists_url	A	https://api.github.com/users/achiu/gi...
starred_url	A	https://api.github.com/users/achiu/st...
subscriptions_url	A	https://api.github.com/users/achiu/su...
organizations_url	A	https://api.github.com/users/achiu/orgs
repos_url	A	https://api.github.com/users/achiu/repos
events_url	A	https://api.github.com/users/a
received_events_url	A	https://api.github.com/user
type	A	User
site_admin	01	true

GITHUB API

GitHubSession
Paw

Cookies
Default Jar
Environments 1

GET https://api.github.com/orgs/github/members

200 OK

Organization
GET https://api.github.com/orgs/github
200 OK

Members
GET https://api.github.com/orgs/githu...
200 OK

Headers URL Params Body Options

Info Request Response Headers JSON Raw

Members

Cookies
Automatically Send Cookies
Store Received Cookies

Authorization
Set HTTP Basic Auth
Set OAuth 1
Set OAuth 2
Set Amazon S3 Signature

Swift (Alamofire) Copy Export Open With

Key or Index	Type	Value
index 0		17 items
login	A	achiu
id	42	24772
avatar_url	A	https://avatars.githubusercontent.co...
gravatar_id	A	
url	A	https://api.github.com/users/achiu
html_url	A	https://github.com/achiu
followers_url	A	https://api.github.com/users/achiu/fol...
following_url	A	https://api.github.com/users/achiu/fol...
gists_url	A	https://api.github.com/users/achiu/gi...
starred_url	A	https://api.github.com/users/achiu/st...
subscriptions_url	A	https://api.github.com/users/achiu/su...
organizations_url	A	https://api.github.com/users/achiu/orgs
repos_url	A	https://api.github.com/users/achiu/repos
events_url	A	https://api.github.com/users/a
received_events_url	A	https://api.github.com/user
type	A	User
site_admin	01	true

SWIFTYJSON

1. mvim -vf Podfile (Vim)

```
1 pod 'Alamofire', :git => 'https://github.com/Alamofire/Alamofire.git', :branch =>
  'swift-2.0'
2 pod 'SwiftyJSON', :git => 'https://github.com/SwiftyJSON/SwiftyJSON.git', :branch =>
  'xcode7'
3 use_frameworks!
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
NORMAL  master  Podfile                    unix < utf-8 < ruby  100%  ln  3:15
```

SWIFTYJSON

1. vazexqi@daigo: ~/Development/TalentSpark/GitHubDemo (zsh)

```
vazexqi@daigo:~/Development/TalentSpark/GitHubDemo|➜  pod install        ± master ✓ 9:52:35
```

MODEL

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 Member

```
public class GitHubOrganization {
    static let DEFAULT_ORGANIZATION_NAME = "github"

    struct Organization {
        let avatarURL: String
        let organizationName: String
        let description: String
        let location: String
    }

    struct Member {
        let login: String
        let avatarUrl: String
    }

    public init() {}

    func fetchOrganizationDetails(callback: (Organization) -> ()) {…}

    func fetchMembersDetails(callback: ([Member]) -> ()) {…}
```

MODEL

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 Member

```swift
public class GitHubOrganization {
    static let DEFAULT_ORGANIZATION_NAME = "github"

    struct Organization {
        let avatarURL: String
        let organizationName: String
        let description: String
        let location: String
    }

    struct Member {
        let login: String
        let avatarUrl: String
    }

    public init() {}

    func fetchOrganizationDetails(callback: (Organization) -> ()) {…}

    func fetchMembersDetails(callback: ([Member]) -> ()) {…}
```

MODEL

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 Member

```
public class GitHubOrganization {
    static let DEFAULT_ORGANIZATION_NAME = "github"

    struct Organization {
        let avatarURL: String
        let organizationName: String
        let description: String
        let location: String
    }

    struct Member {
        let login: String
        let avatarUrl: String
    }

    public init() {}

    func fetchOrganizationDetails(callback: (Organization) -> ()) {…}

    func fetchMembersDetails(callback: ([Member]) -> ()) {…}
```

Load the icon, organization name, description and location

MODEL

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 Member

```
public class GitHubOrganization {
    static let DEFAULT_ORGANIZATION_NAME = "github"

    struct Organization {
        let avatarURL: String
        let organizationName: String
        let description: String
        let location: String
    }

    struct Member {
        let login: String
        let avatarUrl: String
    }

    public init() {}

    func fetchOrganizationDetails(callback: (Organization) -> ()) {…}

    func fetchMembersDetails(callback: ([Member]) -> ()) {…}
```

MODEL

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 Member

```swift
public class GitHubOrganization {
    static let DEFAULT_ORGANIZATION_NAME = "github"

    struct Organization {
        let avatarURL: String
        let organizationName: String
        let description: String
        let location: String
    }

    struct Member {
        let login: String
        let avatarUrl: String
    }

    public init() {}

    func fetchOrganizationDetails(callback: (Organization) -> ()) {…}

    func fetchMembersDetails(callback: ([Member]) -> ()) {…}
```

Just need the icon and username

MODEL

GitHubOrganization.swift

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 GitHubOrganization

```
26      }
27
28      func fetchOrganizationDetails(callback: (Organization) -> Void) {
29          // Fetch organization details
30          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME)
31              .responseJSON { _, _, jsonObj in
32                  self.populateOrganizationInfoWith(jsonObj.value!, callback:
                        callback)
33          }
34      }
35
36      func populateOrganizationInfoWith(data: AnyObject?, callback: (Organization) ->
            Void) {
37          let json = JSON(data!)
38          callback(Organization(avatarURL: json["avatar_url"].stringValue,
                organizationName: json["name"].stringValue, description:
                json["description"].stringValue, location:
                json["location"].stringValue))
39      }
40
41      func fetchMembersDetails(callback: ([Member]) -> Void) {
42          // Fetch organization members
43          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME + "/members")
```

MODEL

GitHubOrganization.swift

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 GitHubOrganization

```
26      }
27
28      func fetchOrganizationDetails(callback: (Organization) -> Vo
29          // Fetch organization details
30          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME)
31              .responseJSON { _, _, jsonObj in
32                  self.populateOrganizationInfoWith(jsonObj.value!
                        callback)
33          }
34      }
35
36      func populateOrganizationInfoWith(data: AnyObject?, callback: (Organization) ->
            Void) {
37          let json = JSON(data!)
38          callback(Organization(avatarURL: json["avatar_url"].stringValue,
                organizationName: json["name"].stringValue, description:
                json["description"].stringValue, location:
                json["location"].stringValue))
39      }
40
41      func fetchMembersDetails(callback: ([Member]) -> Void) {
42          // Fetch organization members
43          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME + "/members")
```

We've seen how to use Alamofire before

MODEL

GitHubOrganization.swift

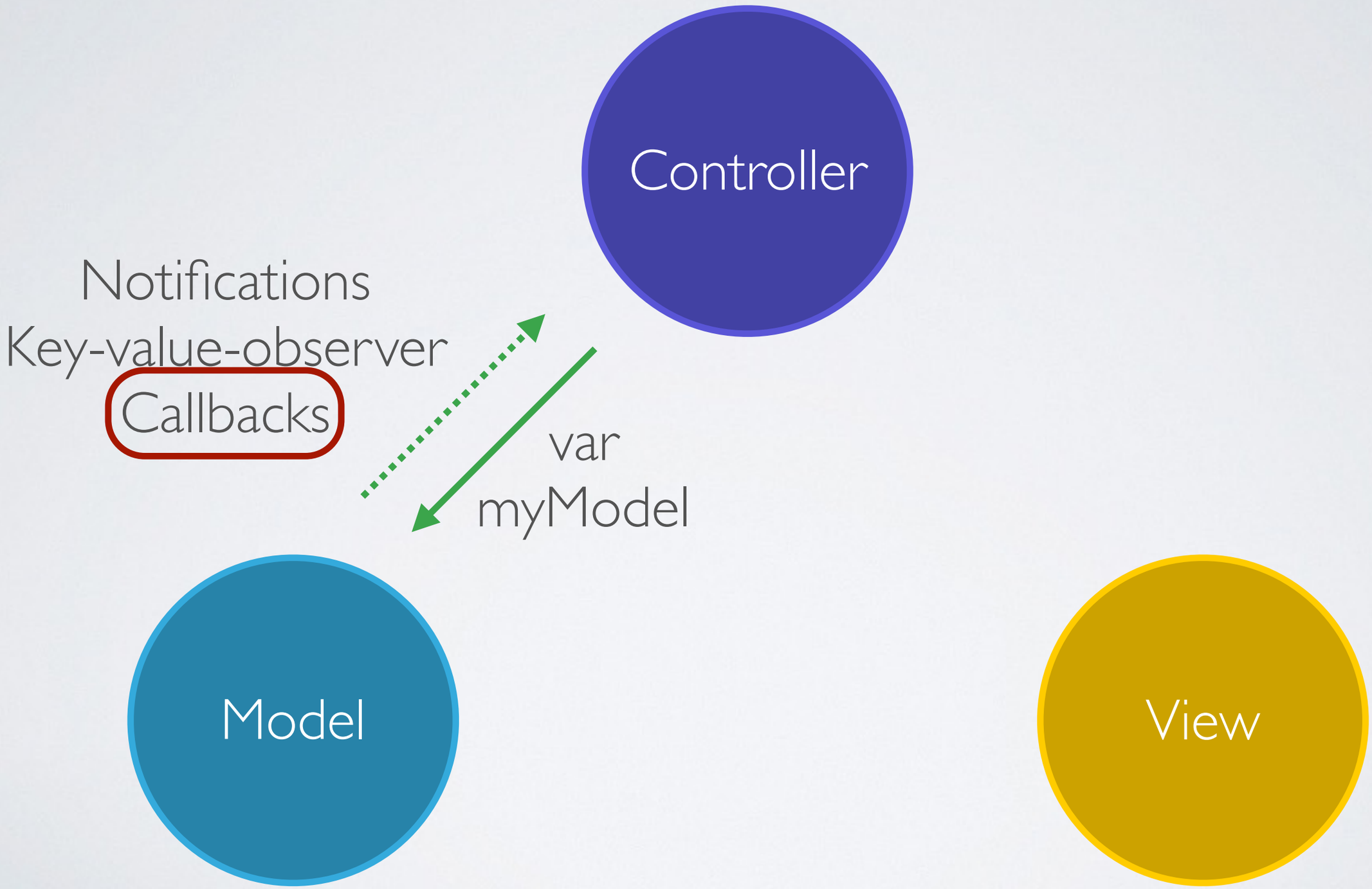
GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 GitHubOrganization

```
26      }
27
28      func fetchOrganizationDetails(callback: (Organization) -> Vo
29          // Fetch organization details
30          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME)
31              .responseJSON { _, _, jsonObj in
32                  self.populateOrganizationInfoWith(jsonObj.value!
                        callback)
33          }
34      }
35
36      func populateOrganizationInfoWith(data: AnyObject?, callback: (Organization) ->
            Void) {
37          let json = JSON(data!)
38          callback(Organization(avatarURL: json["avatar_url"].stringValue,
                    son["name"].stringValue, description:
                    .stringValue, location:
                    ringValue))
39      }
40
41      func                llback: ([Member]) -> Void) {
42          // Fetch organization members
43          Alamofire.request(.GET, "https://api.github.com/orgs/" +
                GitHubOrganization.DEFAULT_ORGANIZATION_NAME + "/members")
```

We've seen how to use Alamofire before

This is SwiftyJSON

NON-SPAGHETTI CODE



MODEL

GitHubOrganization.swift

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 GitHubOrganization

```
    }

    func fetchOrganizationDetails(callback: (Organization) -> Void) {
        // Fetch organization details
        Alamofire.request(.GET, "https://api.github.com/orgs/" +
            GitHubOrganization.DEFAULT_ORGANIZATION_NAME)
            .responseJSON { _, _, jsonObj in
                self.populateOrganizationInfoWith(jsonObj.value!, callback:
                    callback)
        }
    }

    func populateOrganizationInfoWith(data: AnyObject?, callback: (Organization) ->
        Void) {
        let json = JSON(data!)
        callback(Organization(avatarURL: json["avatar_url"].stringValue,
            organizationName: json["name"].stringValue, description:
            json["description"].stringValue, location:
            json["location"].stringValue))
    }

    func fetchMembersDetails(callback: ([Member]) -> Void) {
        // Fetch organization members
        Alamofire.request(.GET, "https://api.github.com/orgs/" +
            GitHubOrganization.DEFAULT_ORGANIZATION_NAME + "/members")
```

MODEL

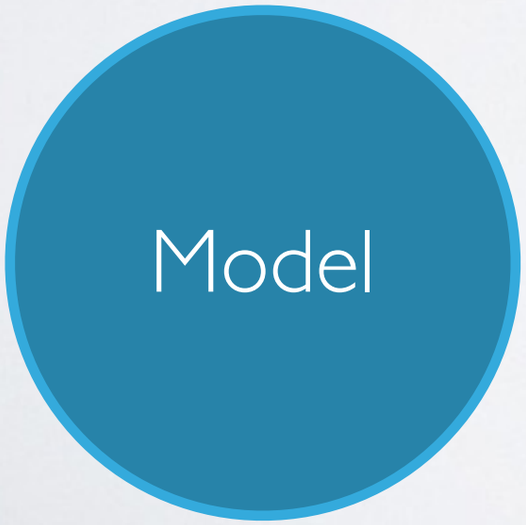
GitHubOrganization.swift

GitHubDemo 〉 GitHubDemo 〉 GitHubOrganization.swift 〉 GitHubOrganization

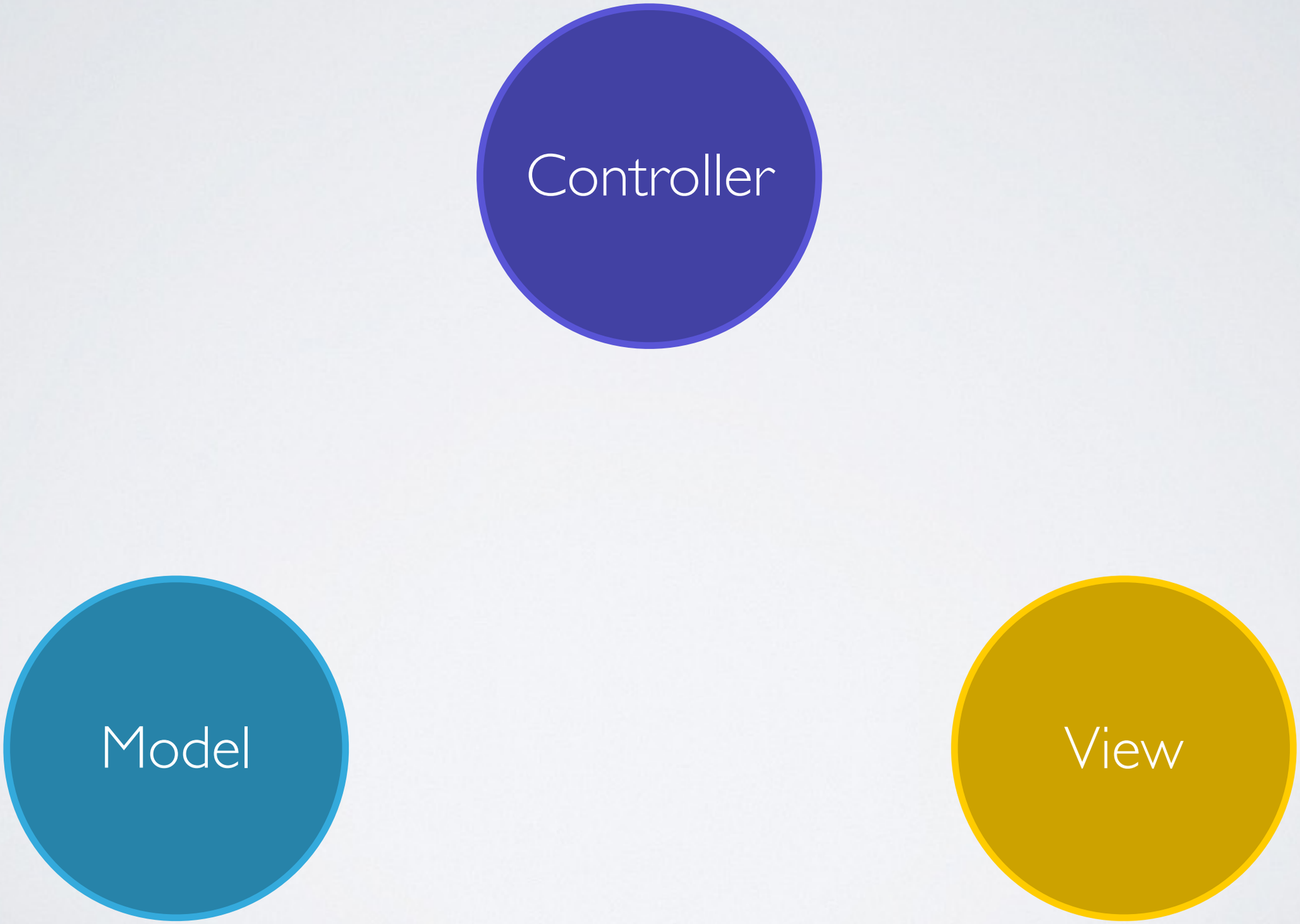
```
    }

    func fetchOrganizationDetails(callback: (Organization) -> Void) {
        // Fetch organization details
        Alamofire.request(.GET, "https://api.github.com/orgs/" +
            GitHubOrganization.DEFAULT_ORGANIZATION_NAME)
            .responseJSON { _, _, jsonObj in
                self.populateOrganizationInfoWith(jsonObj.value!, callback:
                    callback)
        }
    }

    func populateOrganizationInfoWith(data: AnyObject?, callback: (Organization) ->
        Void) {
        let json = JSON(data!)
        callback(Organization(avatarURL: json["avatar_url"].stringValue,
            organizationName: json["name"].str
            json["description"].stringValue, l
            json["location"].stringValue))
    }

    func fetchMembersDetails(callback: ([Membe
        // Fetch organization members
        Alamofire.request(.GET, "https://api.g
            GitHubOrganization.DEFAULT_ORGANIZ
```

How we communicate with our controller

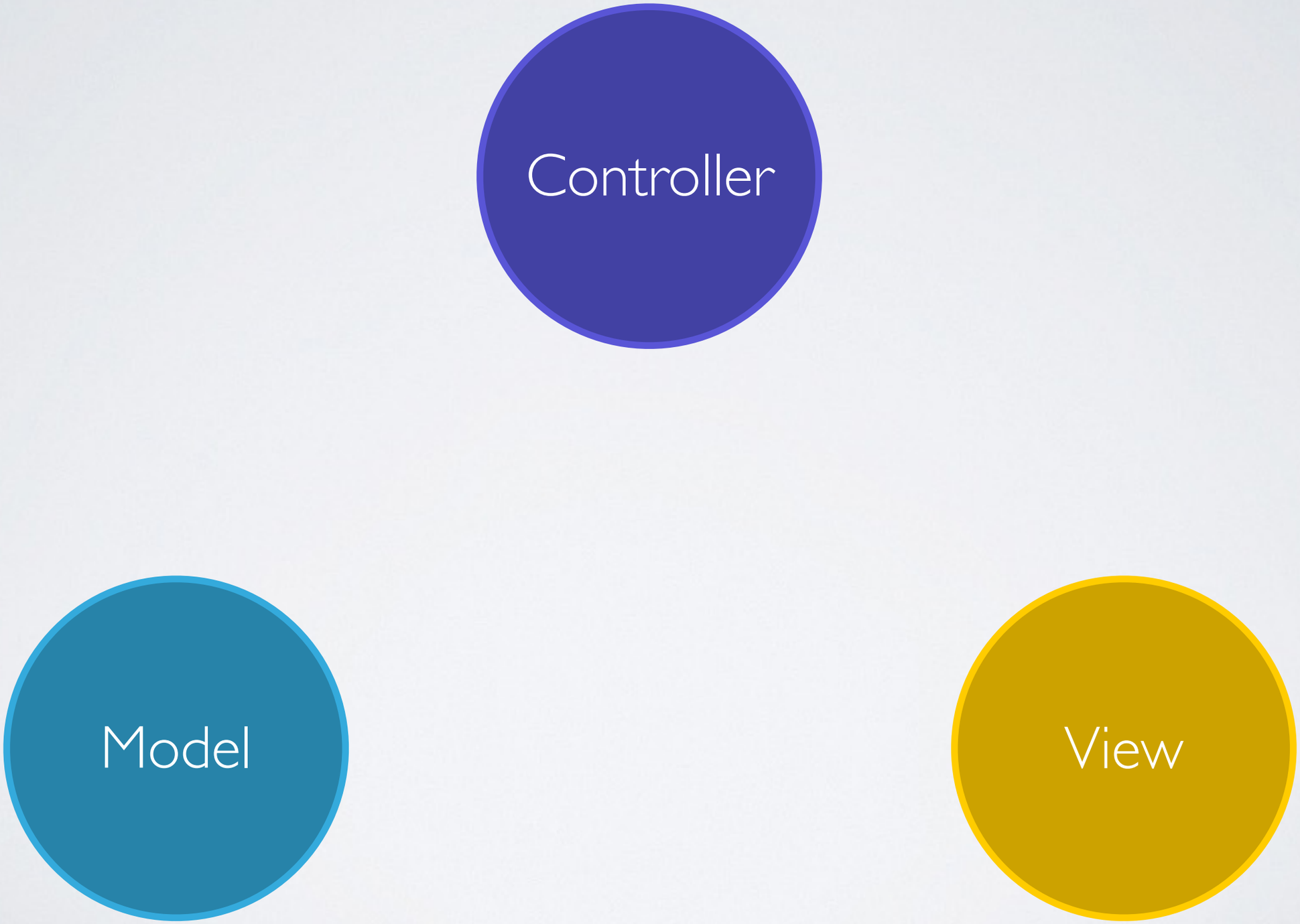
CONTROLLER



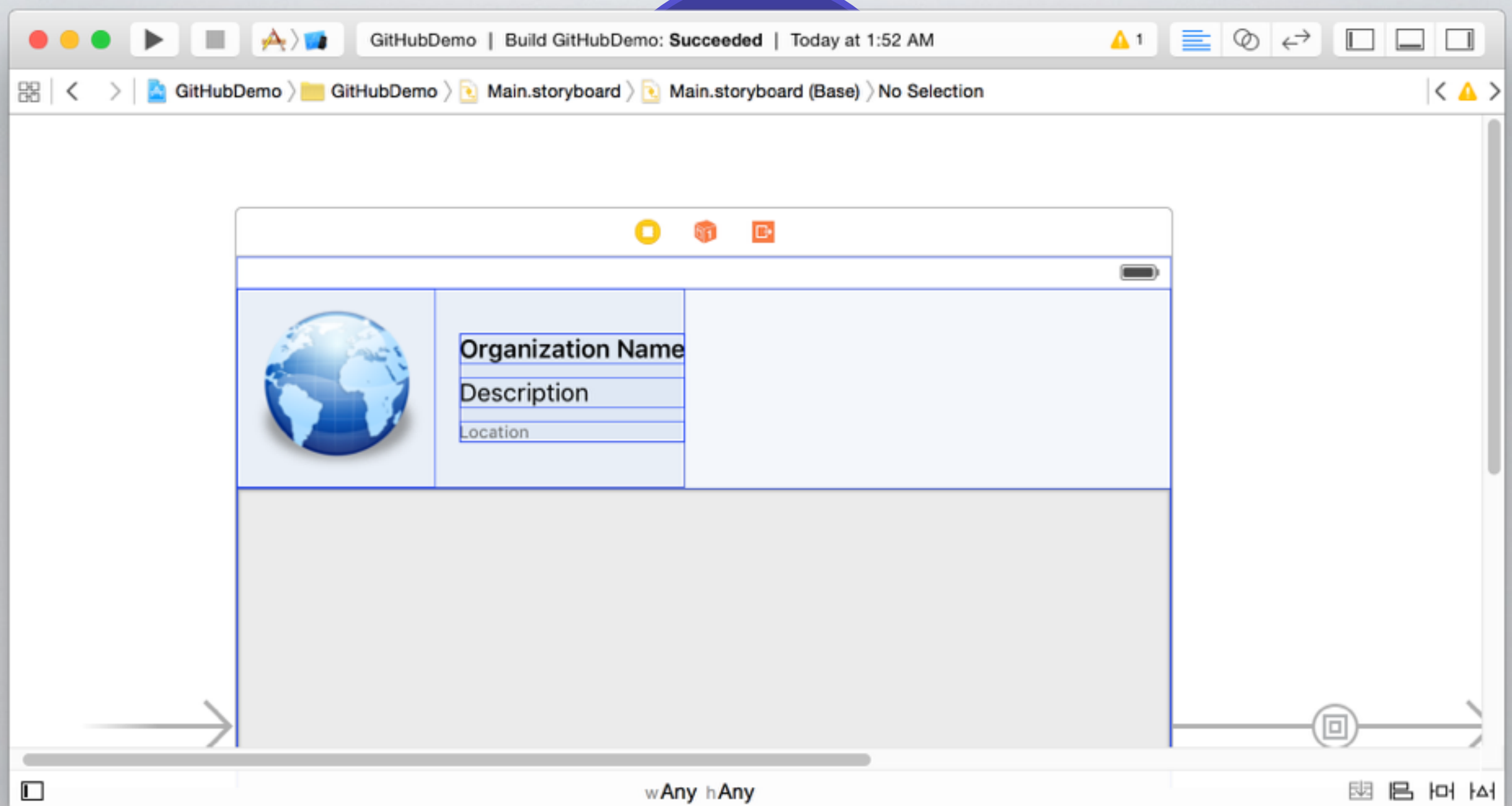
CONTROLLER



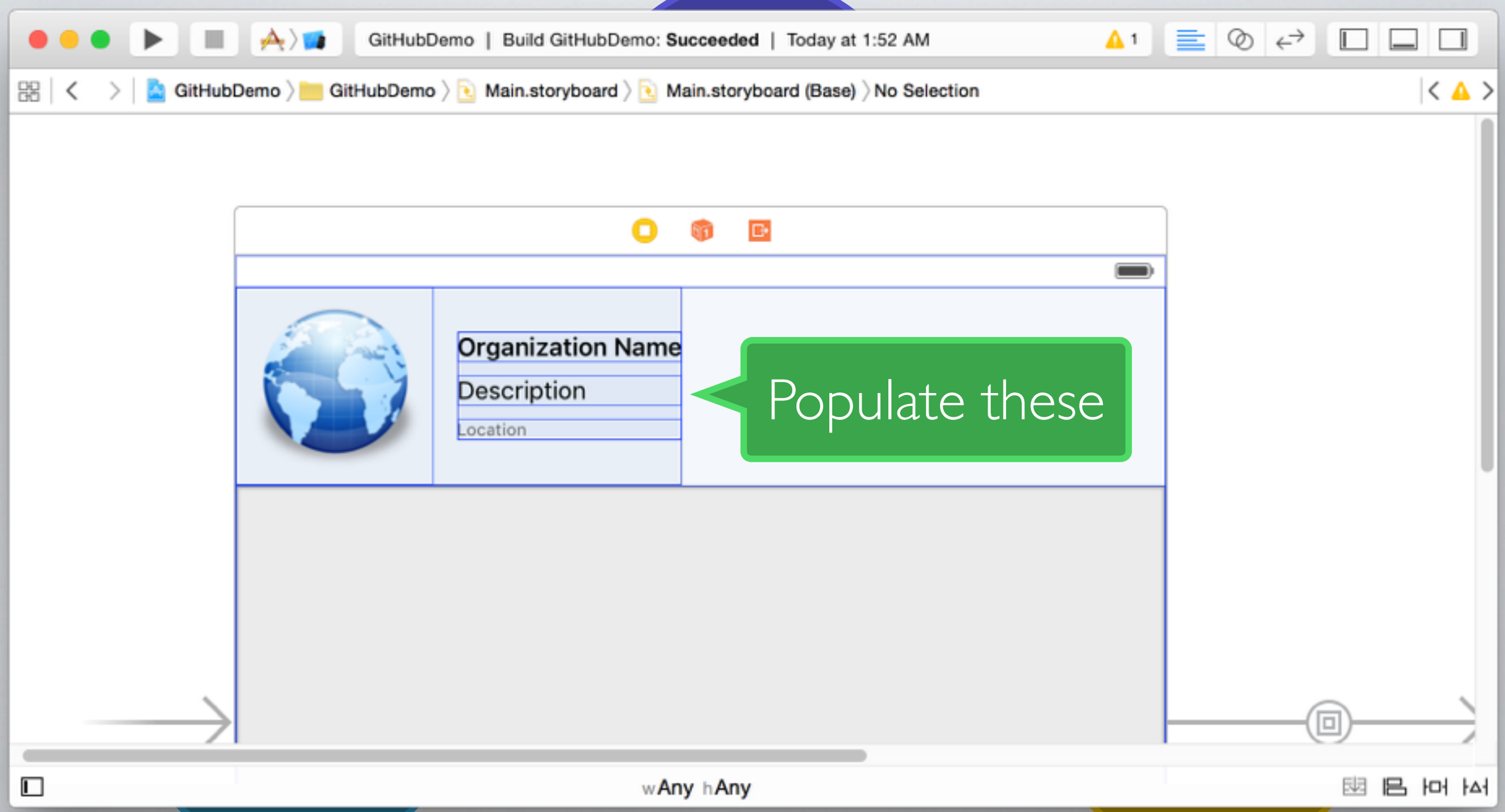
CONTROLLER #1



CONTROLLER #1



CONTROLLER #1



CONTROLLER #1

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubLandingViewController.swift 〉 No Selection

```
    @IBOutlet weak var avatar: UIImageView!
    @IBOutlet weak var organizationLabel: UILabel!
    @IBOutlet weak var descriptionLabel: UILabel!
    @IBOutlet weak var locationLabel: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()
        GitHubOrganization().fetchOrganizationDetails { (org: GitHubOrganization.
            Organization) -> () in
            self.organizationLabel.text = org.organizationName
            self.descriptionLabel.text = org.description
            self.locationLabel.text = org.location

            if let url = NSURL(string: org.avatarURL) {
                if let data = NSData(contentsOfURL: url){
                    self.avatar.contentMode = UIViewContentMode.ScaleAspectFit
                    self.avatar.image = UIImage(data: data)
                }
            }
        }
    }
```

CONTROLLER #1

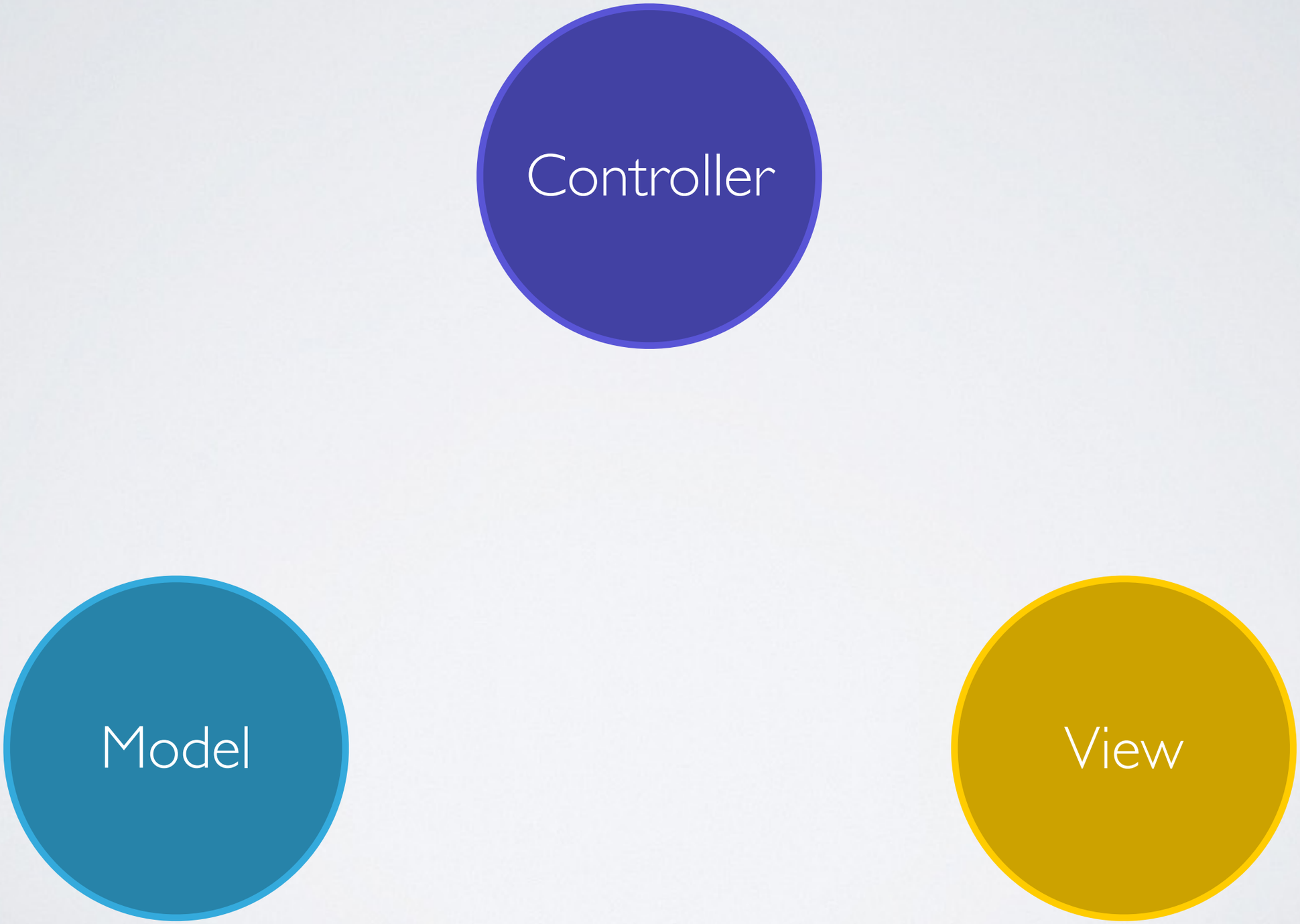
GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubLandingViewController.swift 〉 No Selection

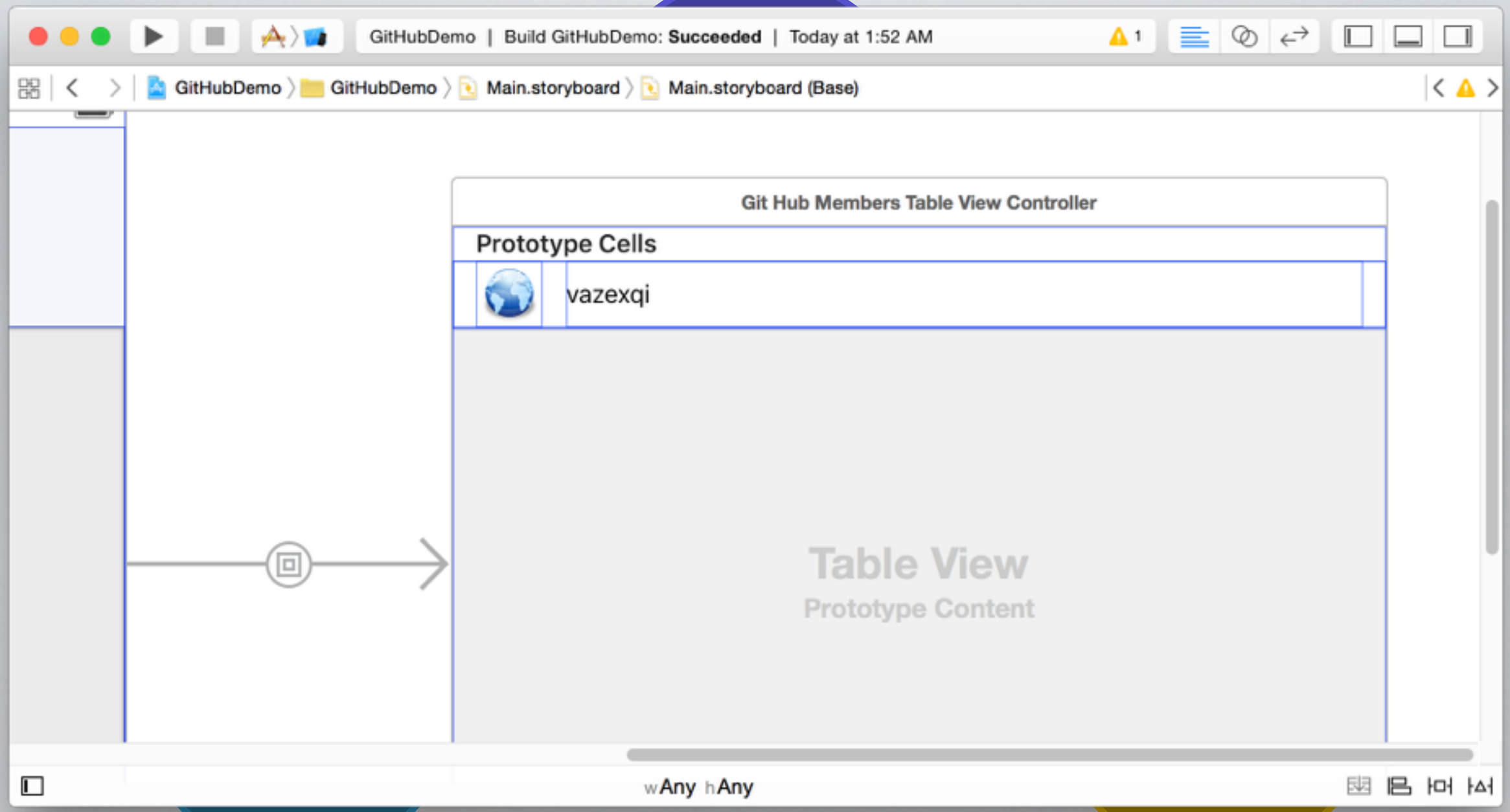
```
@IBOutlet weak var avatar: UIImageView!
@IBOutlet weak var organizationLabel: UILabel!
@IBOutlet weak var descriptionLabel: UILabel!
@IBOutlet weak var locationLabel: UILabel!

override func viewDidLoad() {
    super.viewDidLoad()
    GitHubOrganization().fetchOrganizationDetails { (org: GitHubOrganization.
        Organization) -> () in
        self.organizationLabel.text = org.organizationName
        self.descriptionLabel.text = org.description
        self.locationLabel.text = org.location

        if let url = NSURL(string: org.avatarURL) {
            if let data = NSData(contentsOfURL: url){
                self.avatar.contentMode = UIViewContentMode.Sca
                self.avatar.image = UIImage(data: data)
            }
        }
    }
}
```

This simplistic technique has some UI artifacts

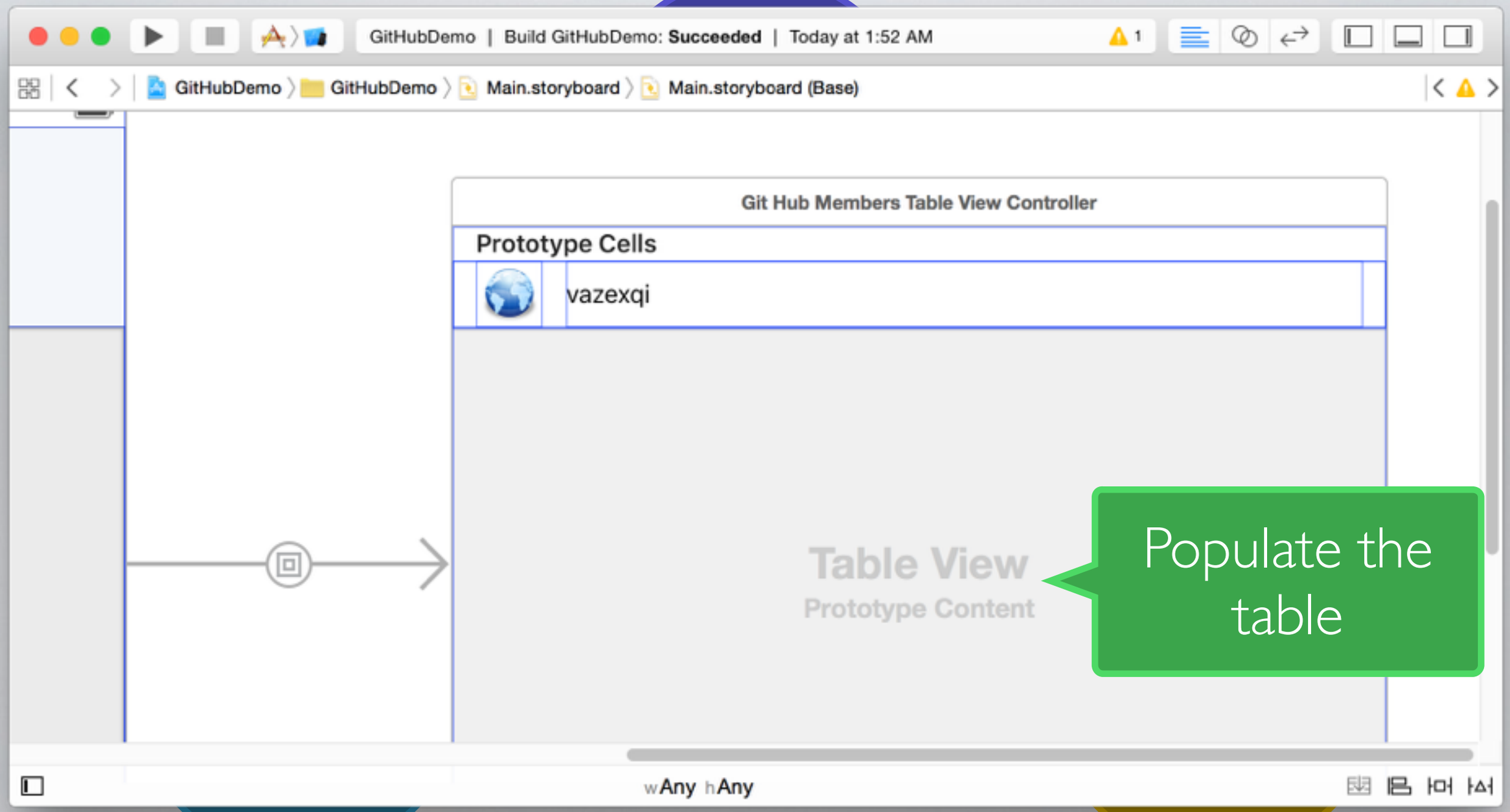
CONTROLLER #2



CONTROLLER #2



CONTROLLER #2



TABLEVIEWCONTROLLER

UITableViewControlle

tableviewcontroller

UITableViewController Class Reference

- UITableViewController
- tabBarController:didSele…
- tabBarController:shouldS…
- tabBarController(_:didSe…
- tabBarController(_:shoul…
- tabBarController:willBeg…
- tabBarController(_:willB…
- tabBarController:didEndC…
- tabBarController:willEnd…
- tabBarController(_:didEn…

Initializing the UITableViewController Object

- initWithStyle:

Getting the Table View

- tableView

Configuring the Table Behavior

- clearsSelectionOnViewWillAppear

Refreshing the Table View

- refreshControl

Inherits From

- NSObject
  - UIResponder
    - UIViewController
      - UITableViewController
        - CABTMIDICentralViewC…

Conforms To

- NSCoding
- NSObject
- UIAppearanceContainer
- UIContentContainer
- UITableViewDataSource
- UITableViewDelegate
- UITraitEnvironment

Import Statement

OBJECTIVE-C

```
@import UIKit;
```

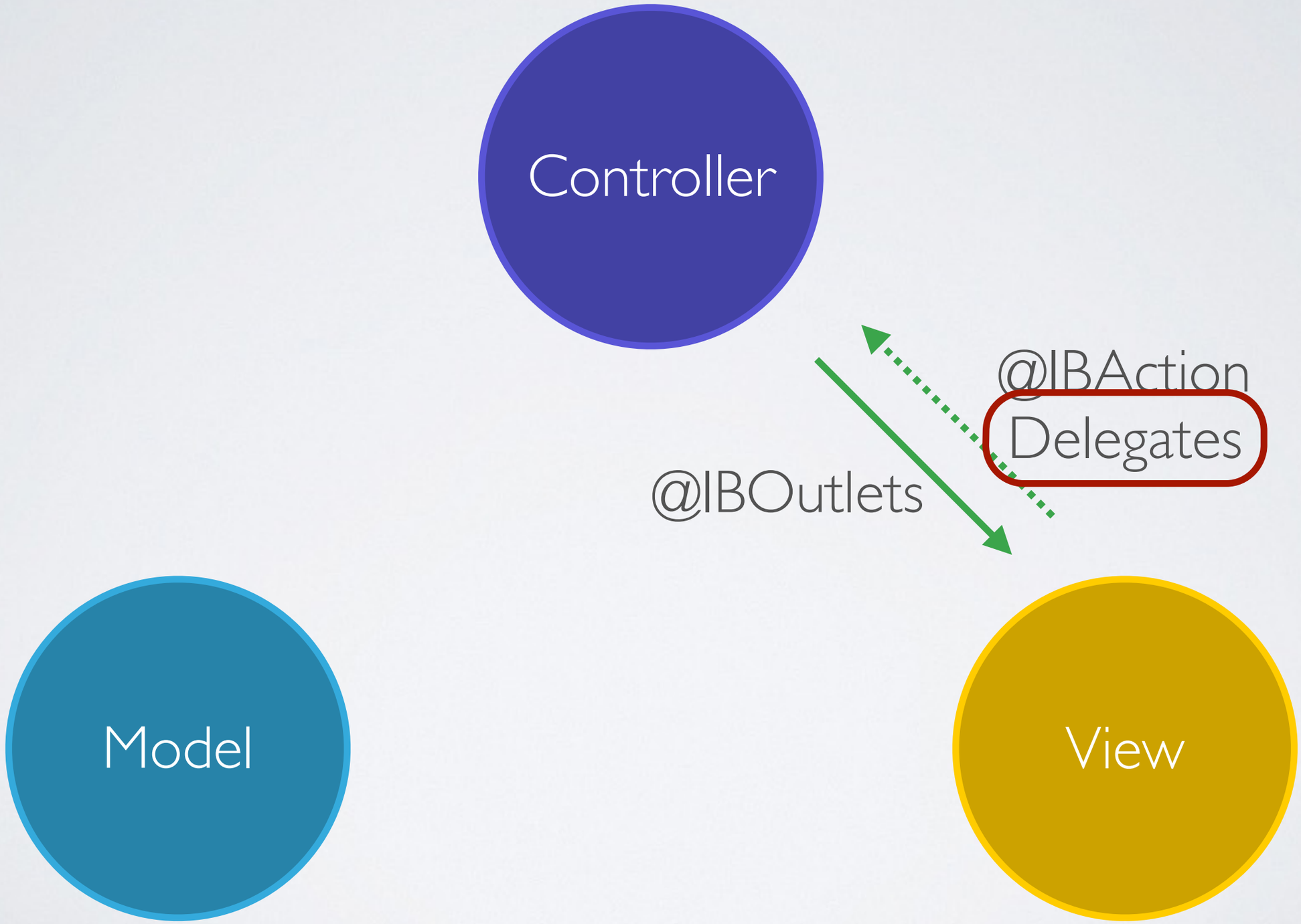
Availability

Available in iOS 2.0 and later

The `UITableViewController` class creates a controller object that manages a table view. It implements the following behavior:

- If a nib file is specified via the `initWithNibName:bundle:` method (which is declared by the superclass `UIViewController`), `UITableViewController` loads the table view archived in the nib file. Otherwise, it creates an unconfigured

NON-SPAGHETTI CODE



UITABLEVIEWDATASOURCE

UITableViewDataSou

uitableviewdatasource

UITableViewDataS... › - tableView:titleFor...

UITableViewDataSource
Google 5
Stack Overflow 5

Configuring a Table View
- tableView:cellForRowAtInd...
- numberOfSectionsInTableVi...
- tableView:numberOfRowsInS...
- sectionIndexTitlesForTabl...
- tableView:sectionForSecti...
- tableView:titleForHeaderI...
- tableView:titleForFooterI...

Inserting or Deleting Table Rows
- tableView:commitEditingSt...
- tableView:canEditRowAtInd...

The `UITableViewDataSource` protocol is adopted by an object that mediates the application's data model for a `UITableView` object. The data source provides the table-view object with the information it needs to construct and modify a table view.

As a representative of the data model, the data source supplies minimal information about the table view's appearance. The table-view object's delegate—an object adopting the `UITableViewDelegate` protocol—provides that information.

The required methods of the protocol provide the cells to be displayed by the table-view as well as inform the `UITableView` object about the number of sections and the number of rows in each section. The data source may implement optional methods to configure various aspects of the table view and to insert, delete, and reorder rows.

NOTE

To enable the swipe-to-delete feature of table views (wherein a user swipes horizontally across a row to display a Delete button), you must implement the `tableView:commitEditingStyle:forRowAtIndexPath:` method.

UITABLEVIEWDATASOURCE

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 No Selection

```
// MARK: - Table view data source

override func numberOfSectionsInTableView(tableView: UITableView) -> Int {…}

override func tableView(tableView: UITableView, titleForHeaderInSection
    section: Int) -> String? {…}

override func tableView(tableView: UITableView, numberOfRowsInSection section:
    Int) -> Int {…}

override func tableView(tableView: UITableView, cellForRowAtIndexPath
    indexPath: NSIndexPath) -> UITableViewCell {…}
}
```

UITABLEVIEWDATASOURCE

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 numberOfSectionsInTableView(_:)

```
// MARK: - Table view data source

override func numberOfSectionsInTableView(tableView: UITableView) -> Int {...}

                                                  w, titleForHeaderInSection

                                                  w, numberOfRowsInSection section:

                                                  w, cellForRowAtIndexPath
                                                  l {...}
}
```

Declaration	`override func numberOfSectionsInTableView(tableView: UITableView) -> Int`
Description	Asks the data source to return the number of sections in the table view. An object representing the table view requesting this information.
Parameters	`tableView` An object representing the table view requesting this information.
Returns	The number of sections in tableView.
Availability	iOS (8.1 and later)
Declared In	UIKit
Reference	UITableViewDataSource Protocol Reference

UITABLEVIEWDATASOURCE

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:titleForHeaderInSection:)

```
// MARK: - Table view data source

override func numberOfSectionsInTableView(tableView: UITableView) -> Int {…}

override func tableView(tableView: UITableView, titleForHeaderInSection

                                                                        iew, numberOfRowsInSection section:

                                                                        iew, cellForRowAtIndexPath
                                                                        ell {…}
}
```

Declaration	`override func tableView(tableView: UITableView, titleForHeaderInSection section: Int) -> String?`
Description	Asks the data source for the title of the header of the specified section of the table view. The table-view object asking for the title.
Parameters	`tableView` The table-view object asking for the title. `section` An index number identifying a section of `tableView` .
Returns	A string to use as the title of the section header.
Availability	iOS (8.1 and later)

Related Declarations	`tableView(_:numberOfRowsInSection:)`
	`tableView(_:cellForRowAtIndexPath:)`
	`tableView`
	Show more...

Declared In	UIKit
Reference	UITableViewDataSource Protocol Reference

UITABLEVIEWDATASOURCE

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:numberOfRowsInSection:)

```
// MARK: - Table view data source

override func numberOfSections InTableView(tableView: UITableView) -> Int {…}

override func tableView(tableView: UITableView, titleForHeaderInSection
    section: Int) -> String? {…}

override func tableView(tableView: UITableView, numberOfRowsInSection section:

                                                  ew, cellForRowAtIndexPath
                                                  ell {…}
}
```

Declaration	`override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int`
Description	Tells the data source to return the number of rows in a given section of a table view. The table-view object requesting this information.
Parameters	`tableView` The table-view object requesting this information. `section` An index number identifying a section in `tableView`.
Returns	The number of rows in section.
Availability	iOS (8.1 and later)

Related Declarations	`tableView(_:titleForHeaderInSection:)`
	`tableView(_:cellForRowAtIndexPath:)`
	`tableView`
	Show more...

Declared In	UIKit
Reference	UITableViewDataSource Protocol Reference

UITABLEVIEWDATASOURCE

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:cellForRowAtIndexPath:)

```
// MARK: - Table view data source

override func numberOfSectionsInTableView(tableView: UITableView) -> Int {…}

override func tableView(tableView: UITableView, titleForHeaderInSection
    section: Int) -> String? {…}

override func tableView(tableView: UITableView, numberOfRowsInSection section:
    Int) -> Int {…}

override func tableView(tableView: UITableView, cellForRowAtIndexPath
                                                    Cell {…}
}
```

Declaration	`override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell`
Description	Asks the data source for a cell to insert in a particular location of the table view. A table-view object requesting the cell.
Parameters	`tableView` A table-view object requesting the cell. `indexPath` An index path locating a row in `tableView`.
Returns	An object inheriting from UITableViewCell that the table view can use for the specified row.
Availability	iOS (8.1 and later)

Related Declarations	`tableView(_:titleForHeaderInSection:)`
	`tableView(_:numberOfRowsInSection:)`
	`tableView`
	Show more...

Declared In	UIKit

UITABLEVIEWDATASOURCE

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:titleForHeaderInSection:)

```
    override func tableView(tableView: UITableView, cellForRowAtIndexPath
        indexPath: NSIndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCellWithIdentifier("memberCell",
            forIndexPath: indexPath)

        let member = members[indexPath.row]
        cell.textLabel?.text = member.login

        if let url = NSURL(string: member.avatarUrl) {
            if let data = NSData(contentsOfURL: url){
                cell.imageView?.image = UIImage(data: data)
            }
        }

        return cell
    }
}
```

UITABLEVIEWDATASOURCE

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:cellForRowAtIndexPath:)

```
override func tableView(tableView: UITableView, cellForRowAtIndexPath
    indexPath: NSIndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCellWithIdentifier("memberCell",
        f

    let m
    cell.

    if le
        i

        }
    }

    retur
    }
}
```

Declaration	`func dequeueReusableCellWithIdentifier(identifier: String, forIndexPath indexPath: NSIndexPath) -> UITableViewCell`
Description	Returns a reusable table-view cell object for the specified reuse identifier. A string identifying the cell object to be reused. This parameter must not be `nil`.
Parameters	`identifier` A string identifying the cell object to be reused. This parameter must not be `nil`. `indexPath` The index path specifying the location of the cell. The data source receives this information when it is asked for the cell and should just pass it along. This method uses the index path to perform additional configuration based on the cell's position in the table view.
Returns	A UITableViewCell object with the associated reuse identifier.
Availability	iOS (6.0 and later)
Related Declarations	`dequeueReusableCellWithIdentifier(_:)`
Declared In	UIKit
Reference	UITableView Class Reference

UITABLEVIEWDATASOURCE

GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:titleForHeaderInSection:)

```
    override func tableView(tableView: UITableView, cellForRowAtIndexPath
        indexPath: NSIndexPath) -> UITableViewCell {
        let cell = tableView.dequeueReusableCellWithIdentifier("memberCell",
            forIndexPath: indexPath)

        let member = members[indexPath.row]
        cell.textLabel?.text = member.login

        if let url = NSURL(string: member.avatarUrl) {
            if let data = NSData(contentsOfURL: url){
                cell.imageView?.image = UIImage(data: data)
            }
        }

        return cell
    }
}
```

UITABLEVIEWDATASOURCE

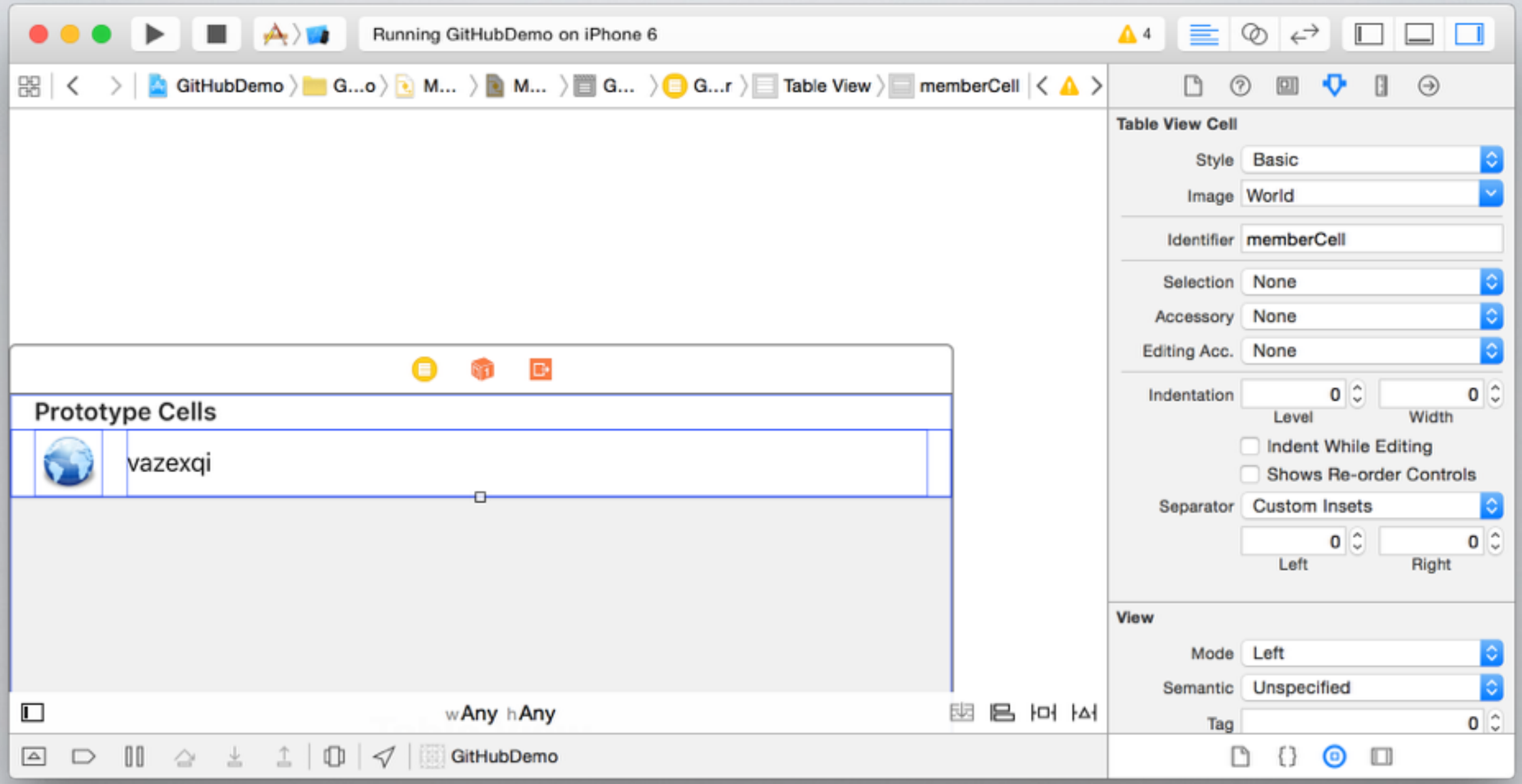
GitHubDemo | Build GitHubDemo: **Succeeded** | Today at 1:52 AM

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:titleForHeaderInSection:)

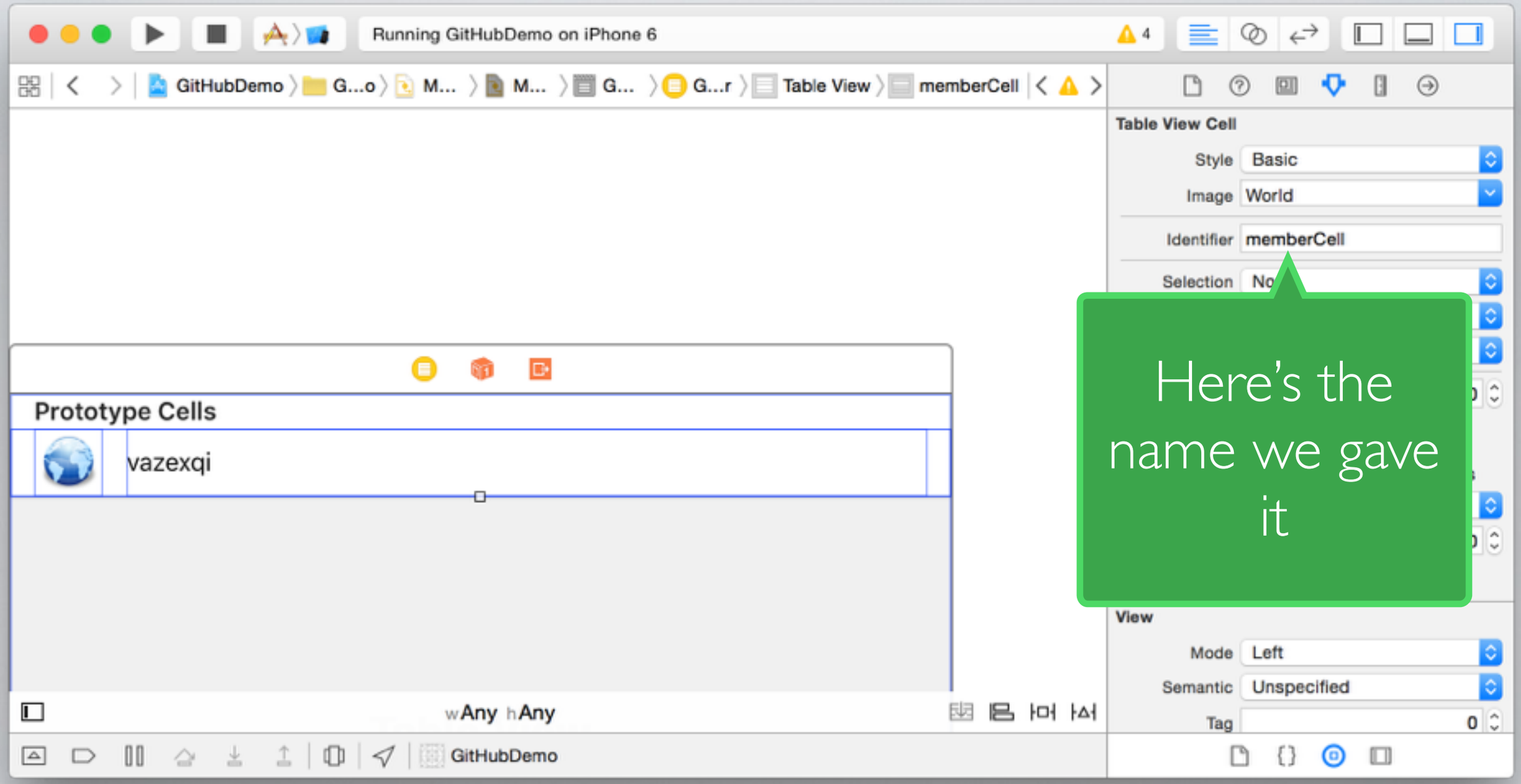
```
override func tableView(tableView: UITableView, cellForRowAtIndexPath
    indexPath: NSIndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCellWithIdentifier("memberCell",
        forIndexPath: indexPath)

    let member = members[indexPath.row]
    cell.textLabel?.text = member.login

    if let url = NSURL(string: member.avatarUrl) {
        if let data = NSData(contentsOfURL: url){
            cell.imageView?.image = UIImage(data: da
        }
    }

    return cell
}
}
```

Where did this name come from?

UITABLEVIEWDATASOURCE



UITABLEVIEWDATASOURCE



Here's the name we gave it

UITABLEVIEWDELEGATE

UITableViewDelegate

uitableviewdelegate

UITableViewDelegate
Google 5
Stack Overflow 5

Configuring Rows for the Table View
- tableView:heightForRowAtI…
- tableView:estimatedHeight…
- tableView:indentationLeve…
- tableView:willDisplayCell…

Managing Accessory Views
- tableView:editActionsForR…
- tableView:accessoryTypeFo…
- tableView:accessoryButton…

Managing Selections
- tableView:willSelectRowAt

UITableViewDelegate Protocol Reference

Availability

Available in iOS 2.0 and later

The delegate of a `UITableView` object must adopt the `UITableViewDelegate` protocol. Optional methods of the protocol allow the delegate to manage selections, configure section headings and footers, help to delete and reorder cells, and perform other actions.

Many methods of the `UITableViewDelegate` protocol take `NSIndexPath` objects as parameters and return values. UIKit declares a category on `NSIndexPath` that enables you to get the represented row index (`row` property) and section index (`section` property), and to construct an index path from a given row index and section index (`indexPathForRow:inSection:` method). Because rows are located within their sections, you usually must evaluate the section index number before you can identify the row by its index number.

UITABLEVIEWDELEGATE

UITableViewDelegate

uitableviewdelegate

UITableViewDelegate
Google 5
Stack Overflow 5

Configuring Rows for the Table View
- tableView:heightForRowAtI…
- tableView:estimatedHeight…
- tableView:indentationLeve…
- tableView:willDisplayCell…

Managing Accessory Views
- tableView:editActionsForR…
- tableView:accessoryTypeFo…
- tableView:accessoryButton…

Managing Selections

UITableViewDelegate Protocol Reference

Availability

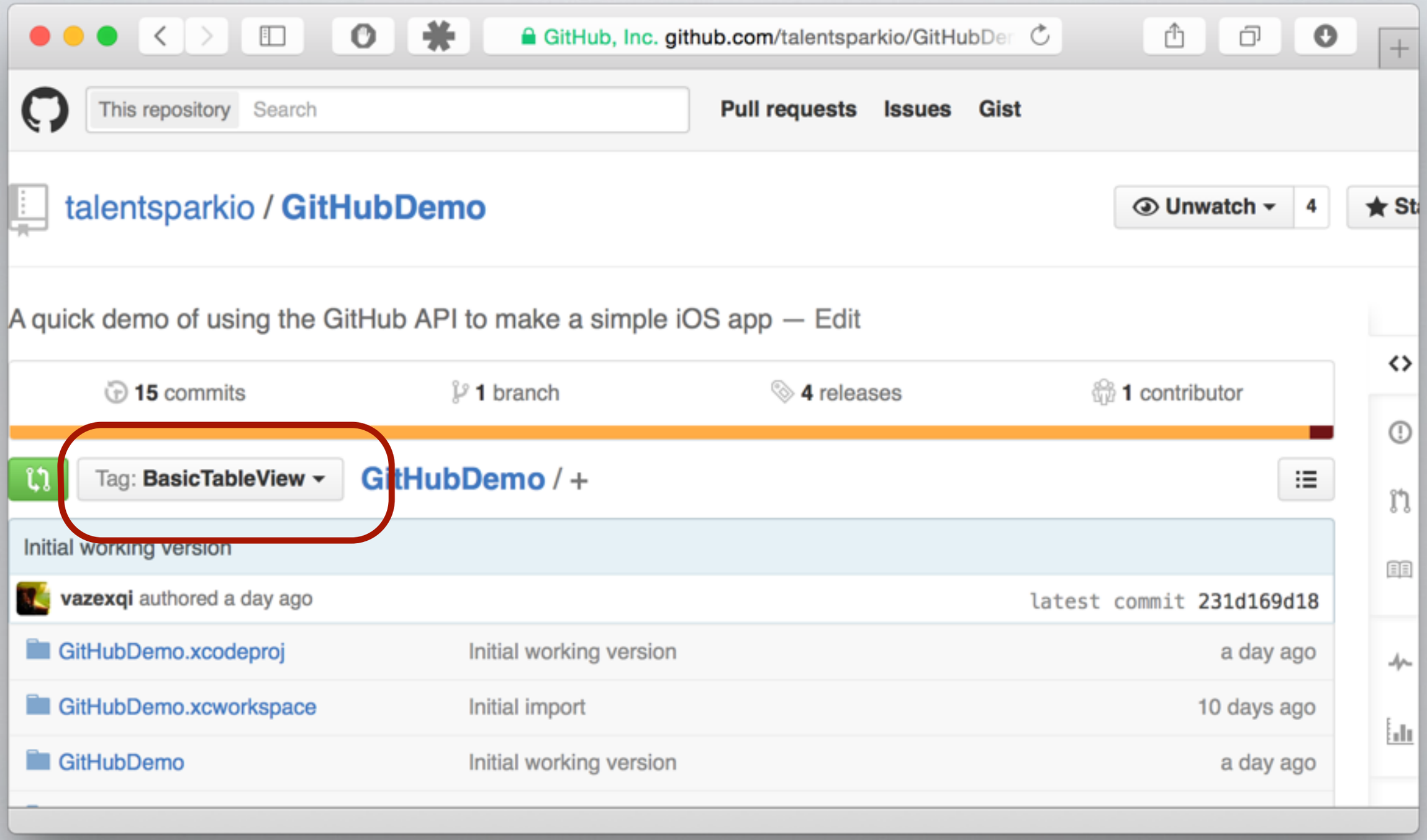
Available in iOS 2.0 and later

The delegate of a `UITableView` object must adopt the `UITableViewDelegate` protocol. Optional methods of the protocol allow the delegate to manage selections, configure section headings and footers, help to delete and reorder cells, and perform other actions.

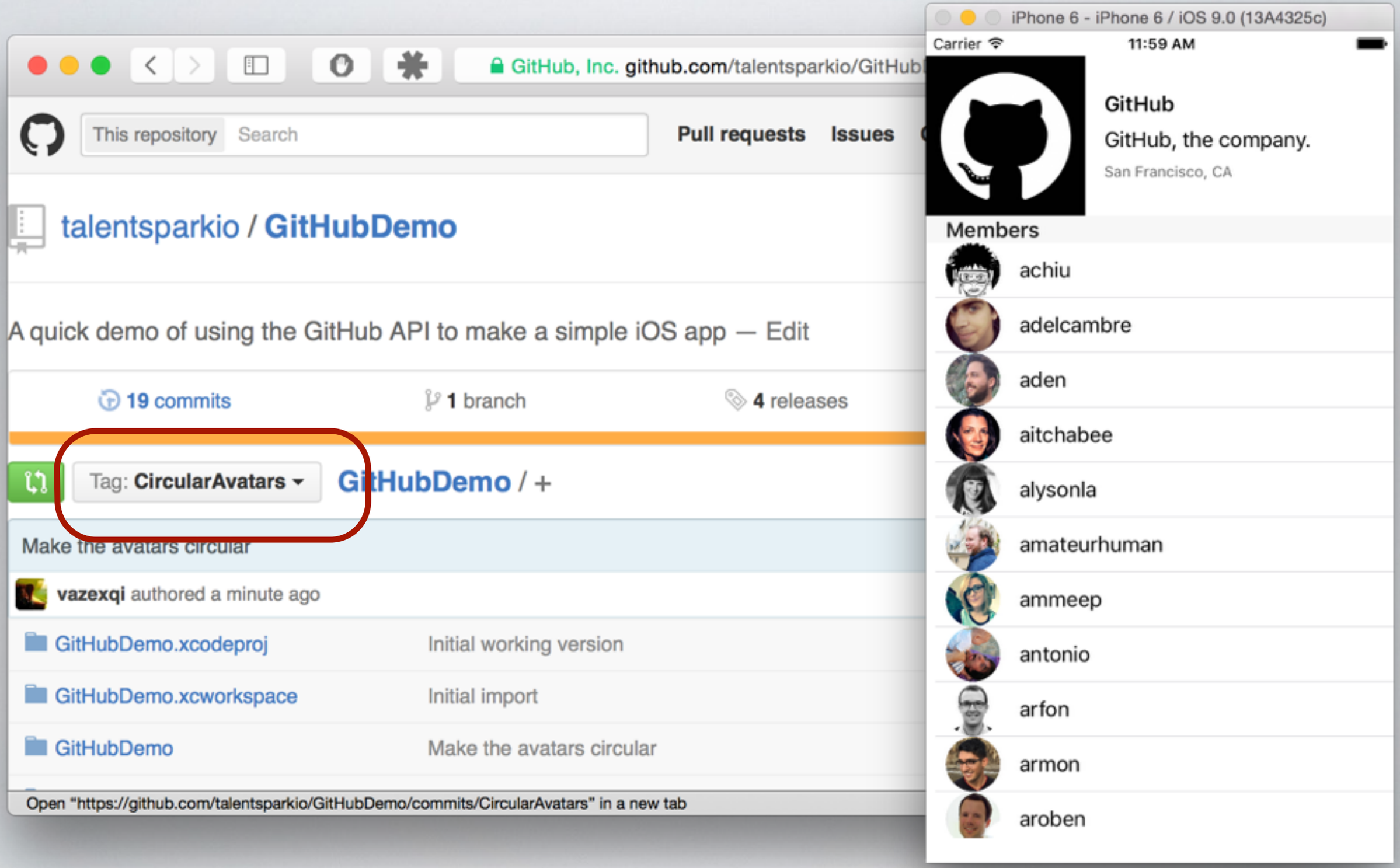
Many methods of the `UITableViewDelegate` protocol take `NSIndexPath` objects as parameters and return values. UIKit declares a category on `NSIndexPath` that enables you to get the represented row index (`row` property) and section index (`section` property), and to construct an index path from a given row index and section index (`indexPathForRow:inSection:` method). Because rows are located within their sections, you usually must evaluate the section index number before you can identify the row by its index number.

Didn't use it this time

CODE


GitHub, Inc. github.com/talentsparkio/GitHubDe
This repository
Search
Pull requests
Issues
Gist
talentsparkio / GitHubDemo
Unwatch
4
A quick demo of using the GitHub API to make a simple iOS app — Edit
15 commits
1 branch
4 releases
1 contributor
Tag: BasicTableView
GitHubDemo / +
Initial working version
vazexqi authored a day ago
latest commit 231d169d18
GitHubDemo.xcodeproj
Initial working version
a day ago
GitHubDemo.xcworkspace
Initial import
10 days ago
GitHubDemo
Initial working version
a day ago

CODE


GitHub, Inc. github.com/talentsparkio/GitHubD
This repository
Search
Pull requests
Issues
talentsparkio / GitHubDemo
A quick demo of using the GitHub API to make a simple iOS app — Edit
19 commits
1 branch
4 releases
Tag: CircularAvatars
GitHubDemo / +
Make the avatars circular
vazexqi authored a minute ago
GitHubDemo.xcodeproj
Initial working version
GitHubDemo.xcworkspace
Initial import
GitHubDemo
Make the avatars circular
Open "https://github.com/talentsparkio/GitHubDemo/commits/CircularAvatars" in a new tab
iPhone 6 - iPhone 6 / iOS 9.0 (13A4325c)
Carrier
11:59 AM
GitHub
GitHub, the company.
San Francisco, CA
Members
achiu
adelcambre
aden
aitchabee
alysonla
amateurhuman
ammeep
antonio
arfon
armon
aroben

BONUS (GCD)

Running GitHubDemo on iPhone 6

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:cellForRowAtIndexPath:)

```
NSIndexPath) -> UITableViewCell {
let cell = tableView.dequeueReusableCellWithIdentifier("memberCell",
    forIndexPath: indexPath)

let member = members[indexPath.row]
cell.textLabel?.text = member.login

if let url = NSURL(string: member.avatarUrl) {
    if let data = NSData(contentsOfURL: url){
        if let avatarSquare = UIImage(data: data) {
            let avatarCircle = UIImage.roundedRectImageFromImage
                (avatarSquare, imageSize: avatarSquare.size, cornerRadius:
                avatarSquare.size.width / 2)
            cell.imageView?.image = avatarCircle
        }
    }
}

return cell
```

GitHubDemo

BONUS (GCD)

Running GitHubDemo on iPhone 6

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:cellForRowAtIndexPath:)

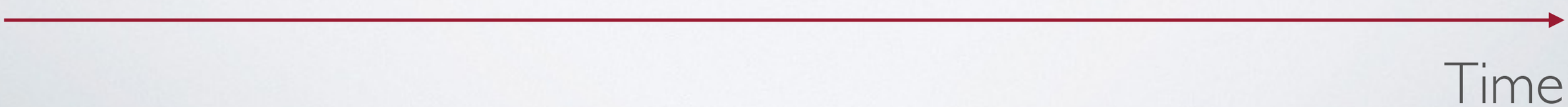
```
NSIndexPath) -> UITableViewCell {
let cell = tableView.dequeueReusableCellWithIdentifier("memberCell",
    forIndexPath: indexPath)

let member = members[indexPath.row]
cell.textLabel?.text = member.login

if let url = NSURL(string: member.avatarUrl) {
    if let data = NSData(contentsOfURL: url){
        if let avatarSquare = UIImage(data: data) {
            let avatarC
                (avatar
                avatarS
            cell.imageV
        }
    }
}

return cell
```

This makes a network call… blocks the UI

GitHubDemo

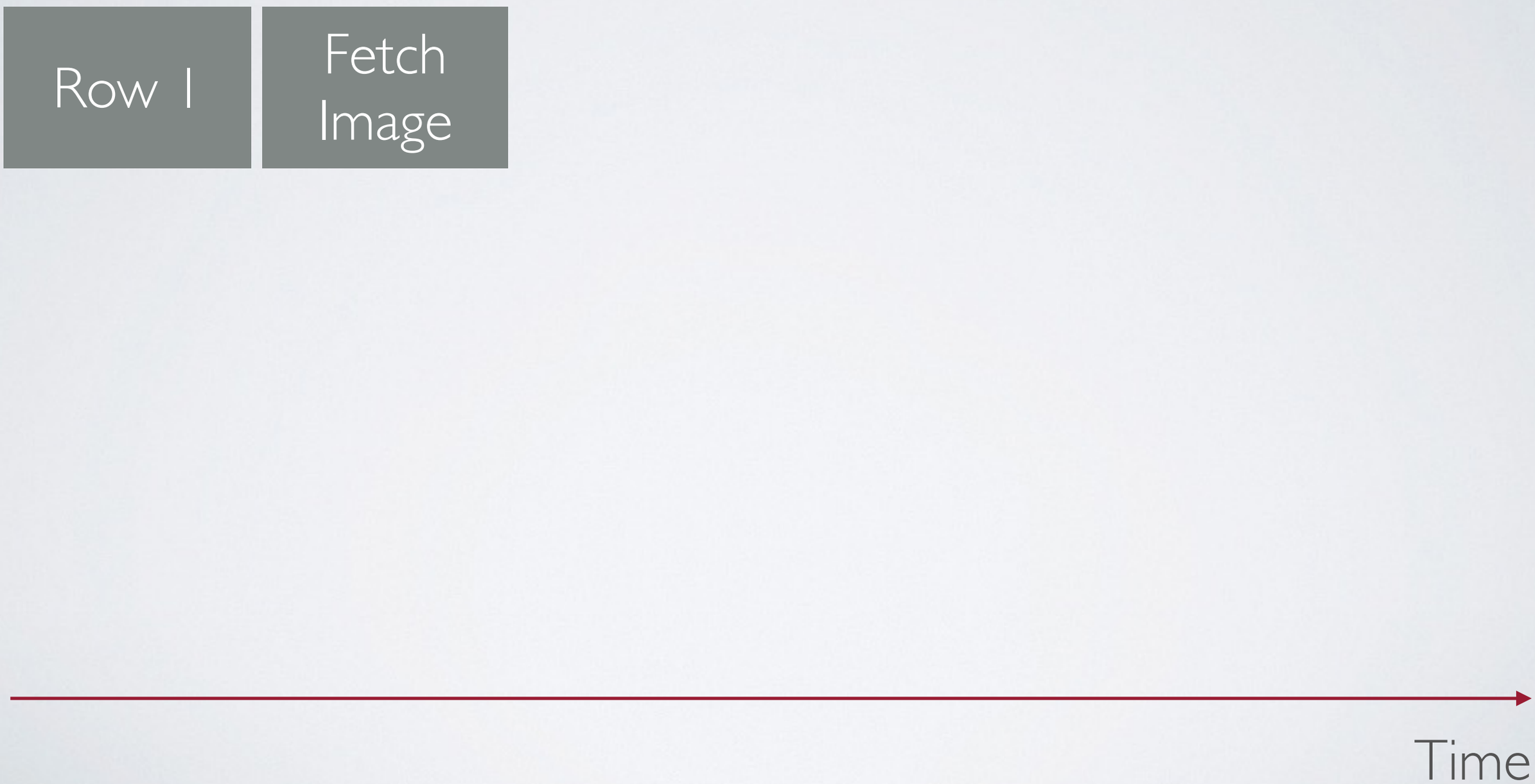
BONUS (GCD)

BONUS (GCD)

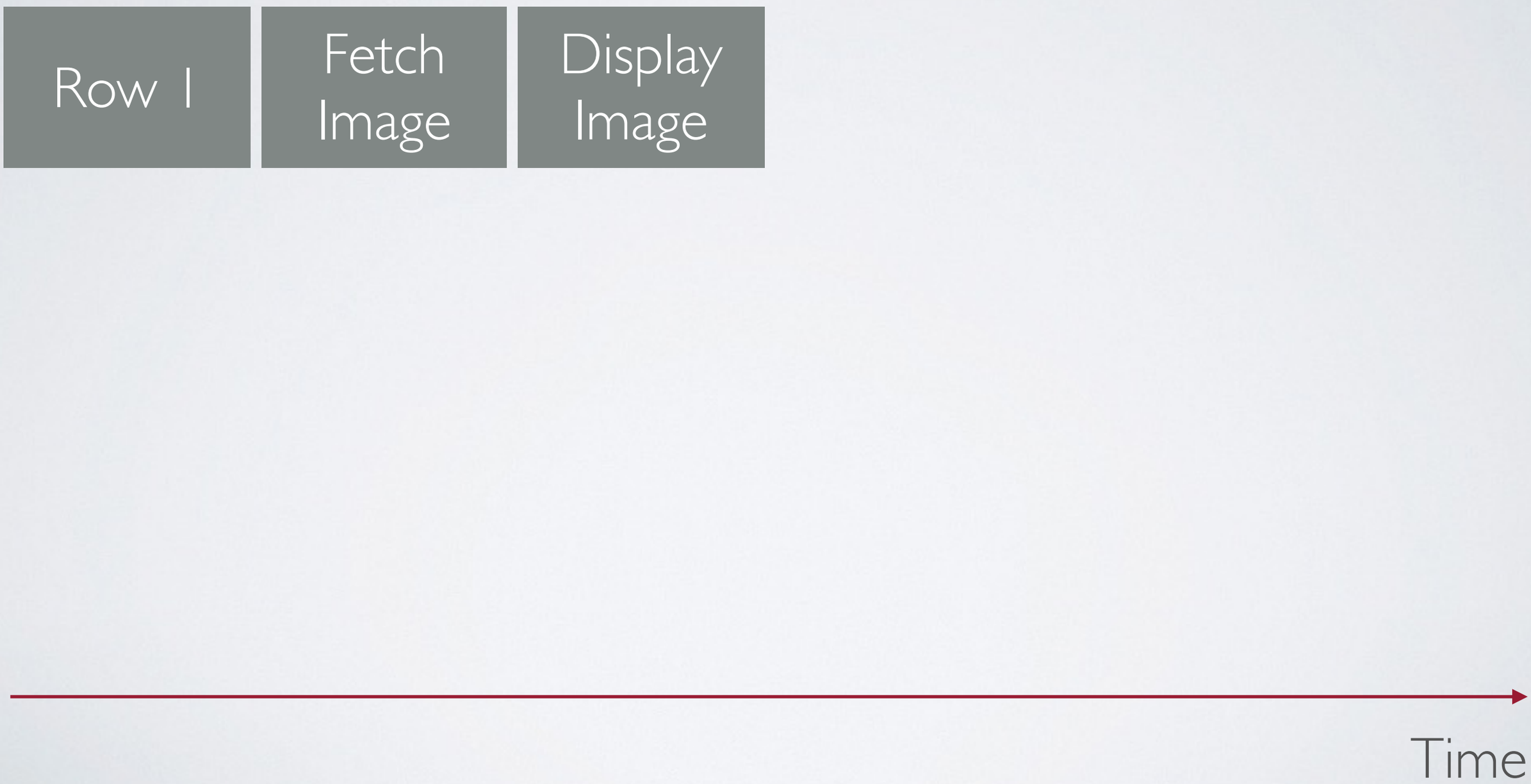


BONUS (GCD)

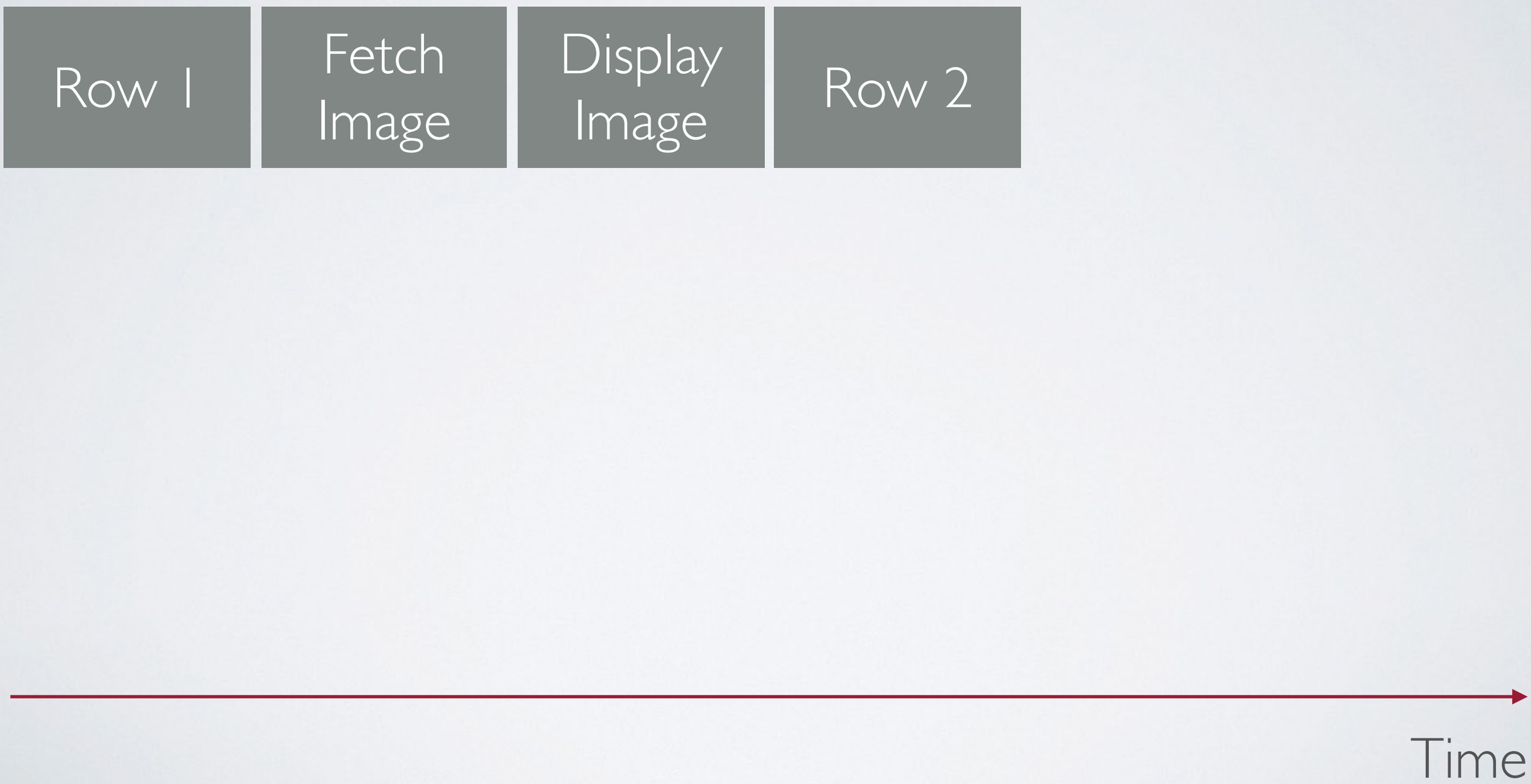
BONUS (GCD)



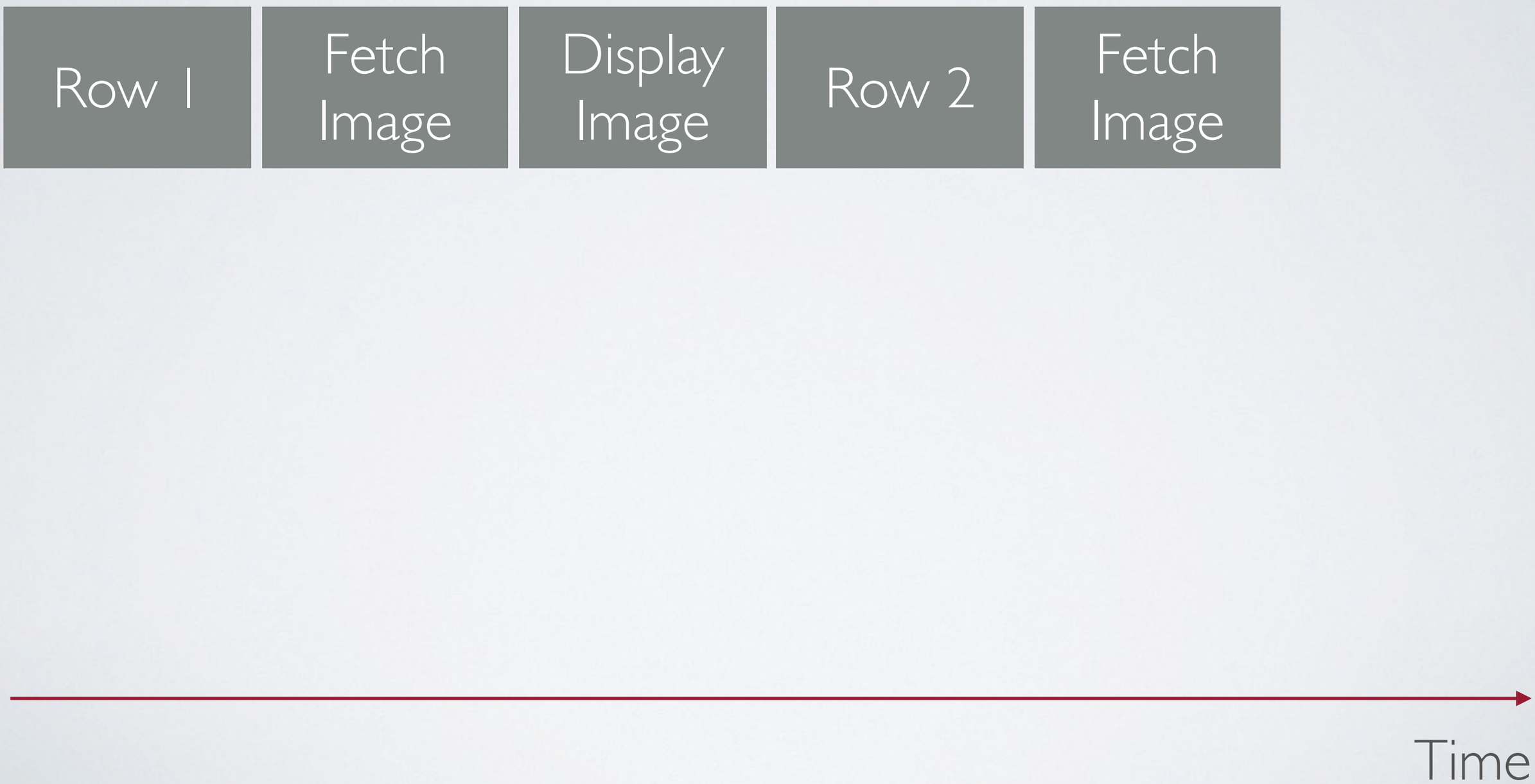
BONUS (GCD)



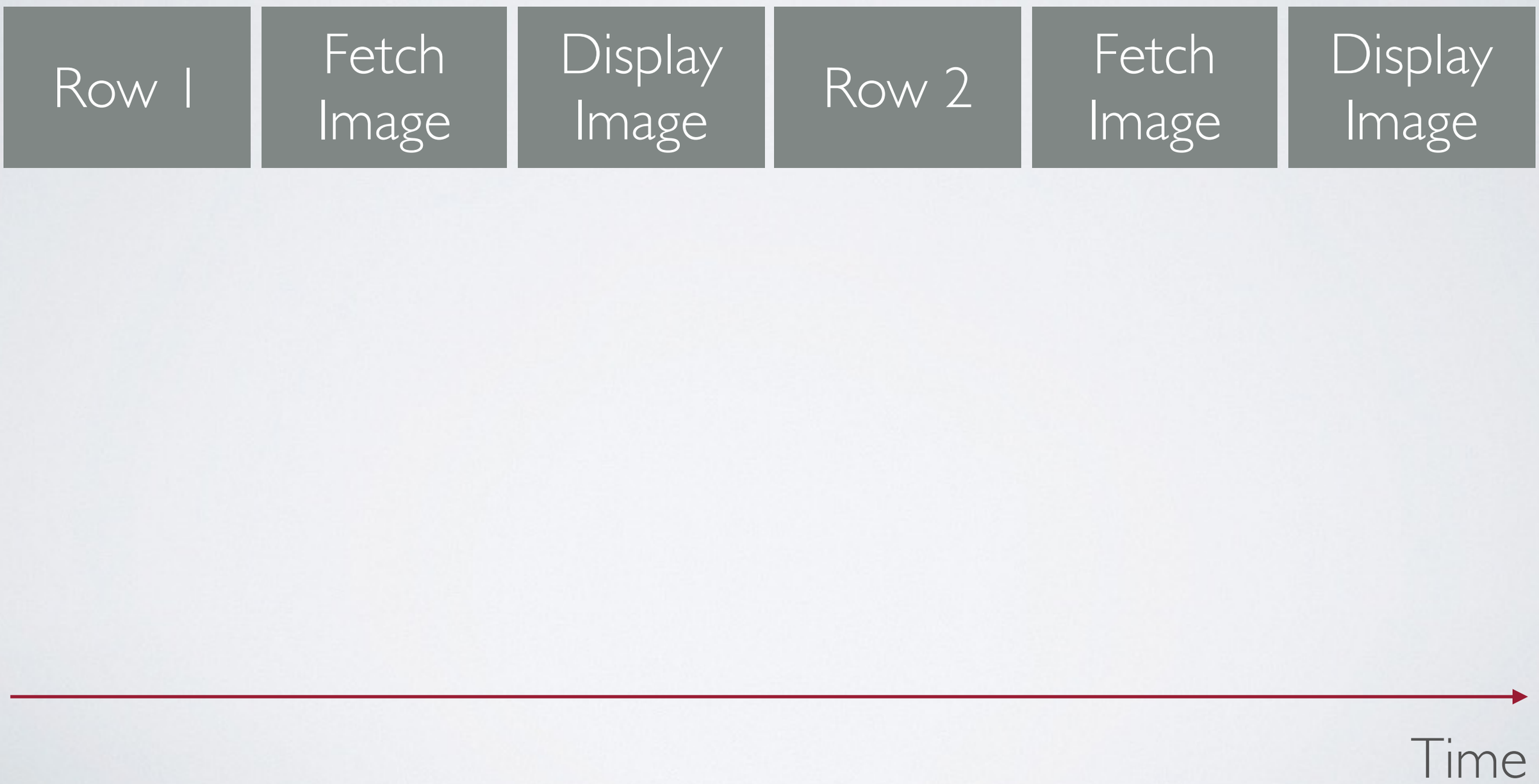
BONUS (GCD)



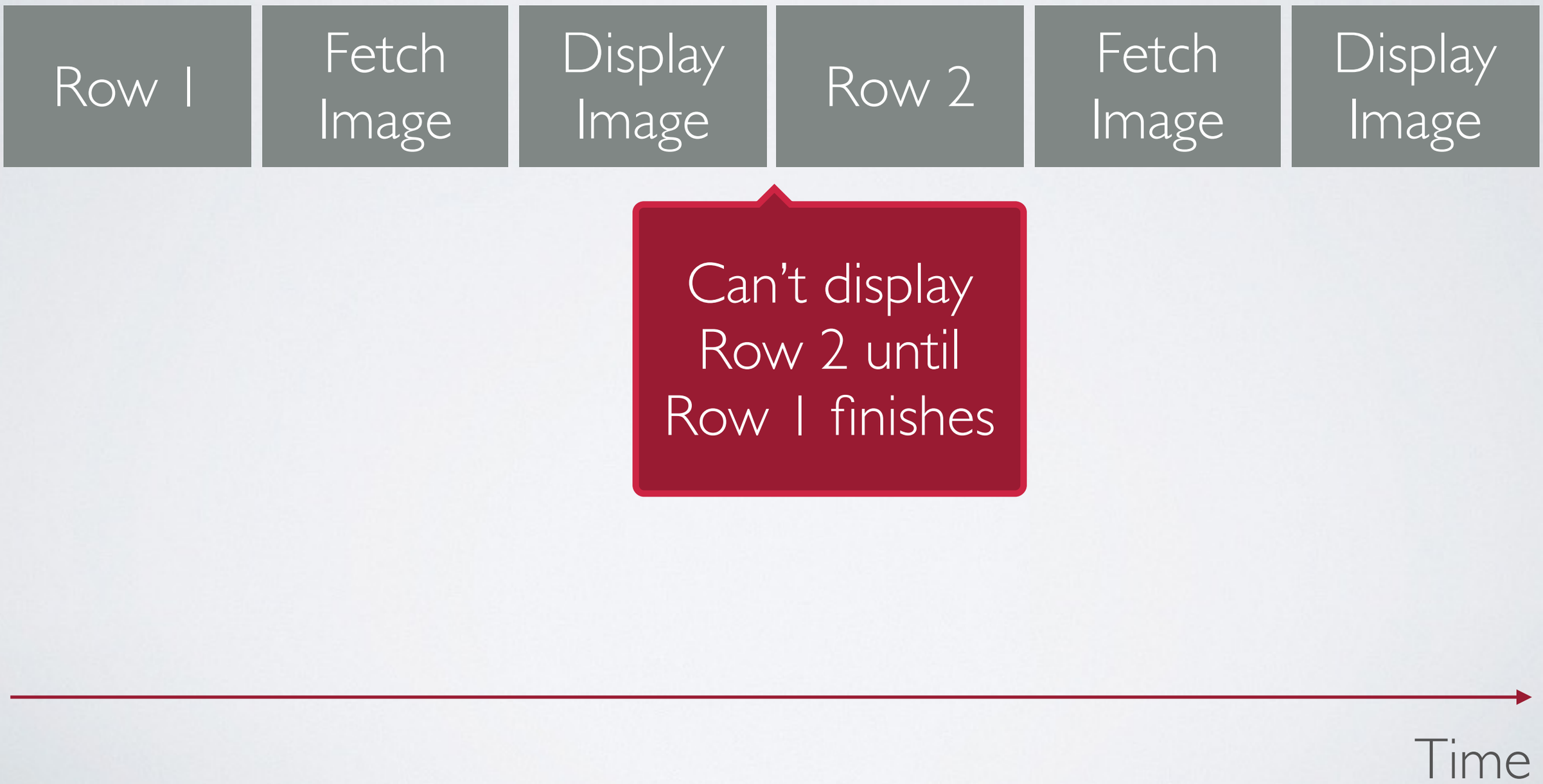
BONUS (GCD)



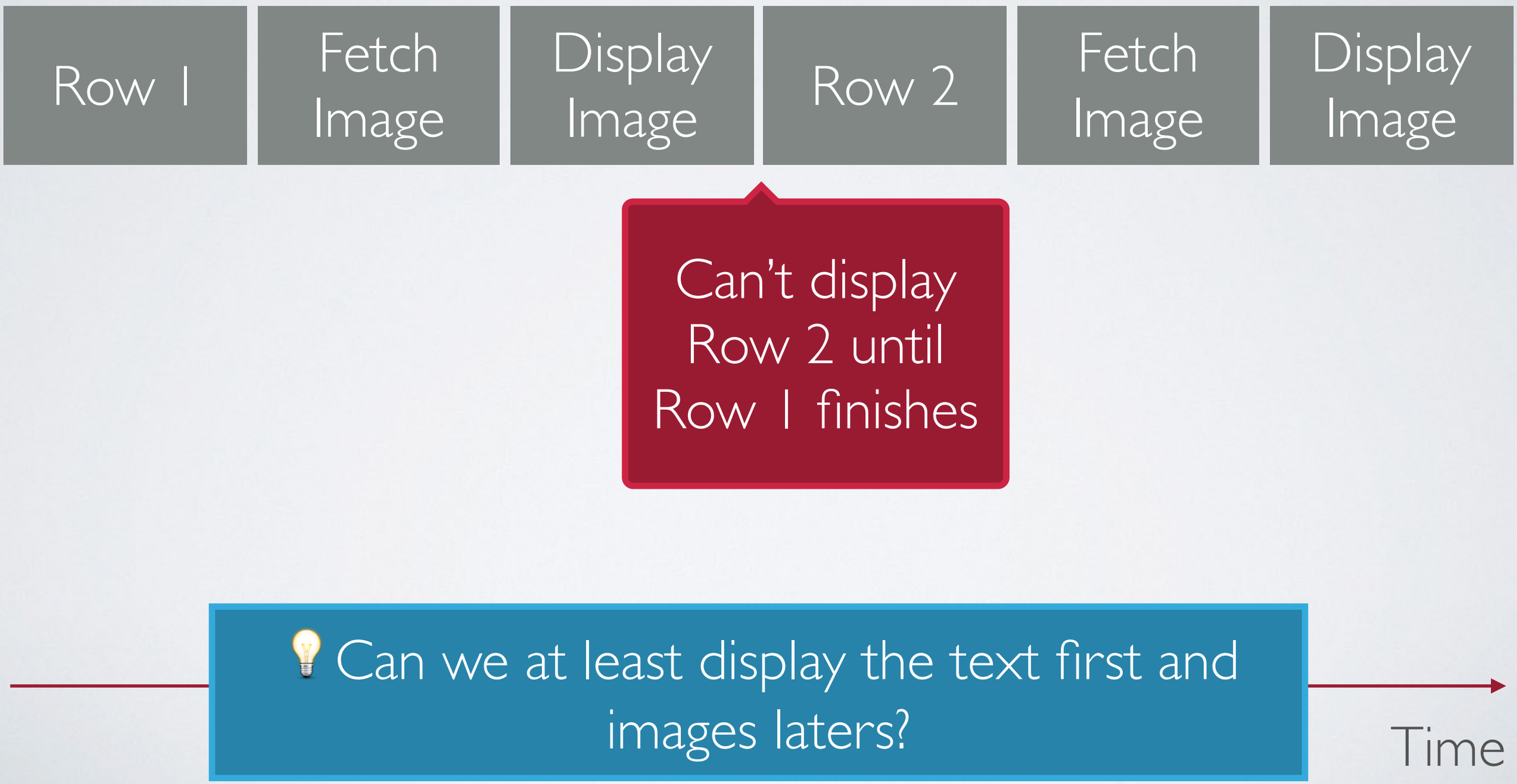
BONUS (GCD)


Row 1
Fetch Image
Display Image
Row 2
Fetch Image
Display Image
Time

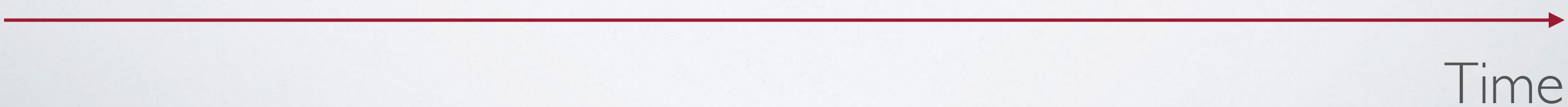
BONUS (GCD)


Row 1
Fetch Image
Display Image
Row 2
Fetch Image
Display Image
Can't display Row 2 until Row 1 finishes
Time

BONUS (GCD)

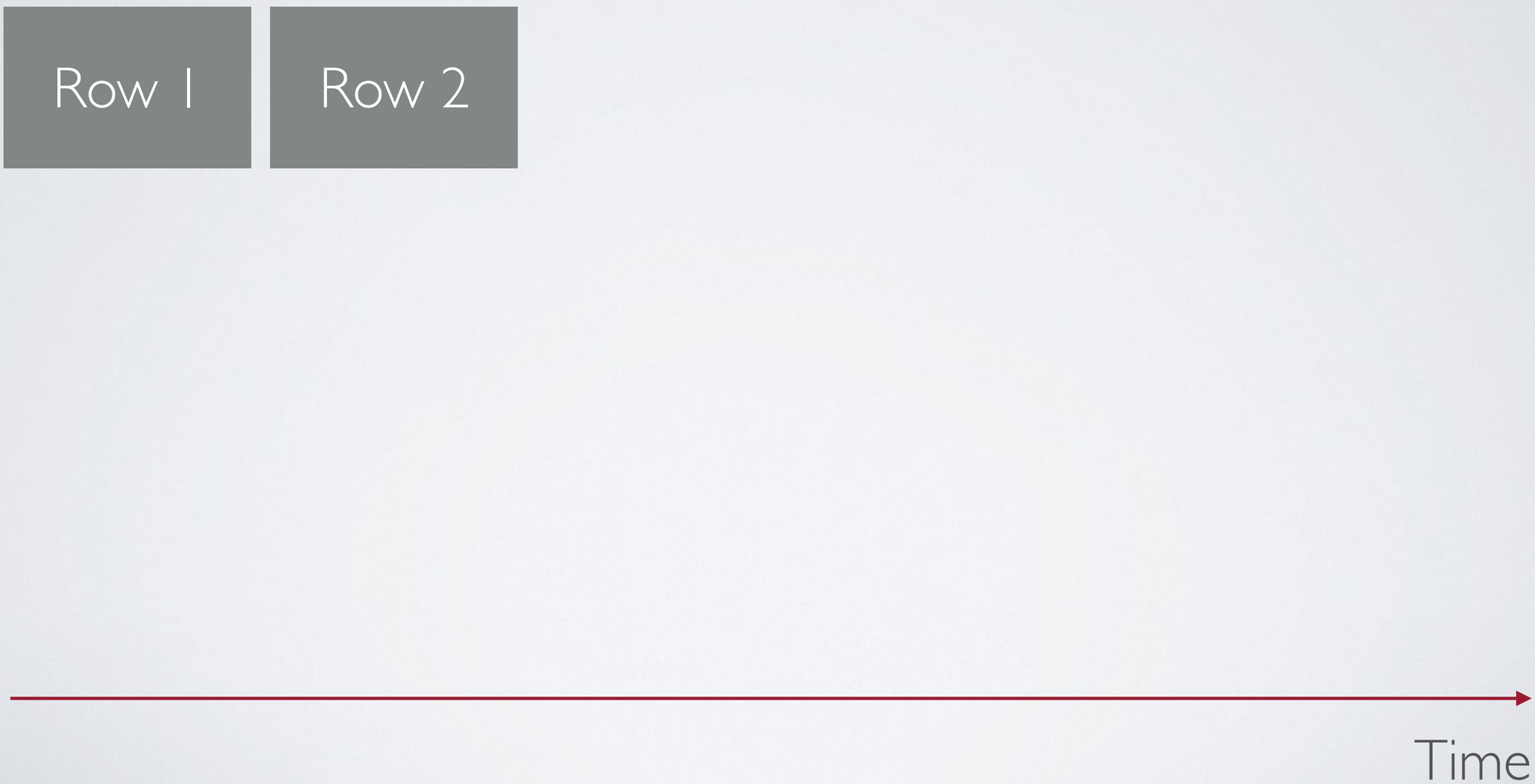

Row 1
Fetch Image
Display Image
Row 2
Fetch Image
Display Image
Can't display Row 2 until Row 1 finishes
Can we at least display the text first and images laters?
Time

BONUS (GCD)

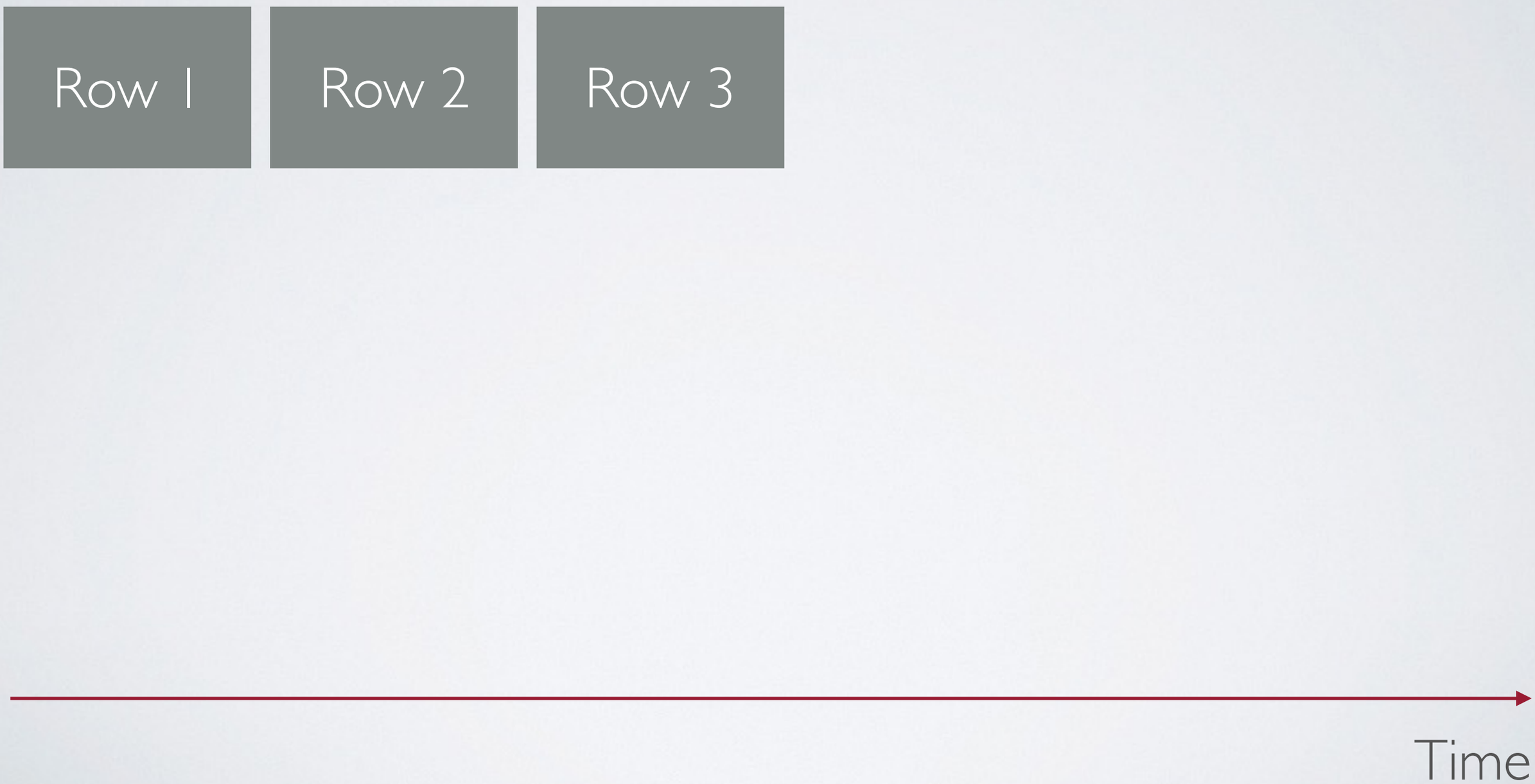


BONUS (GCD)

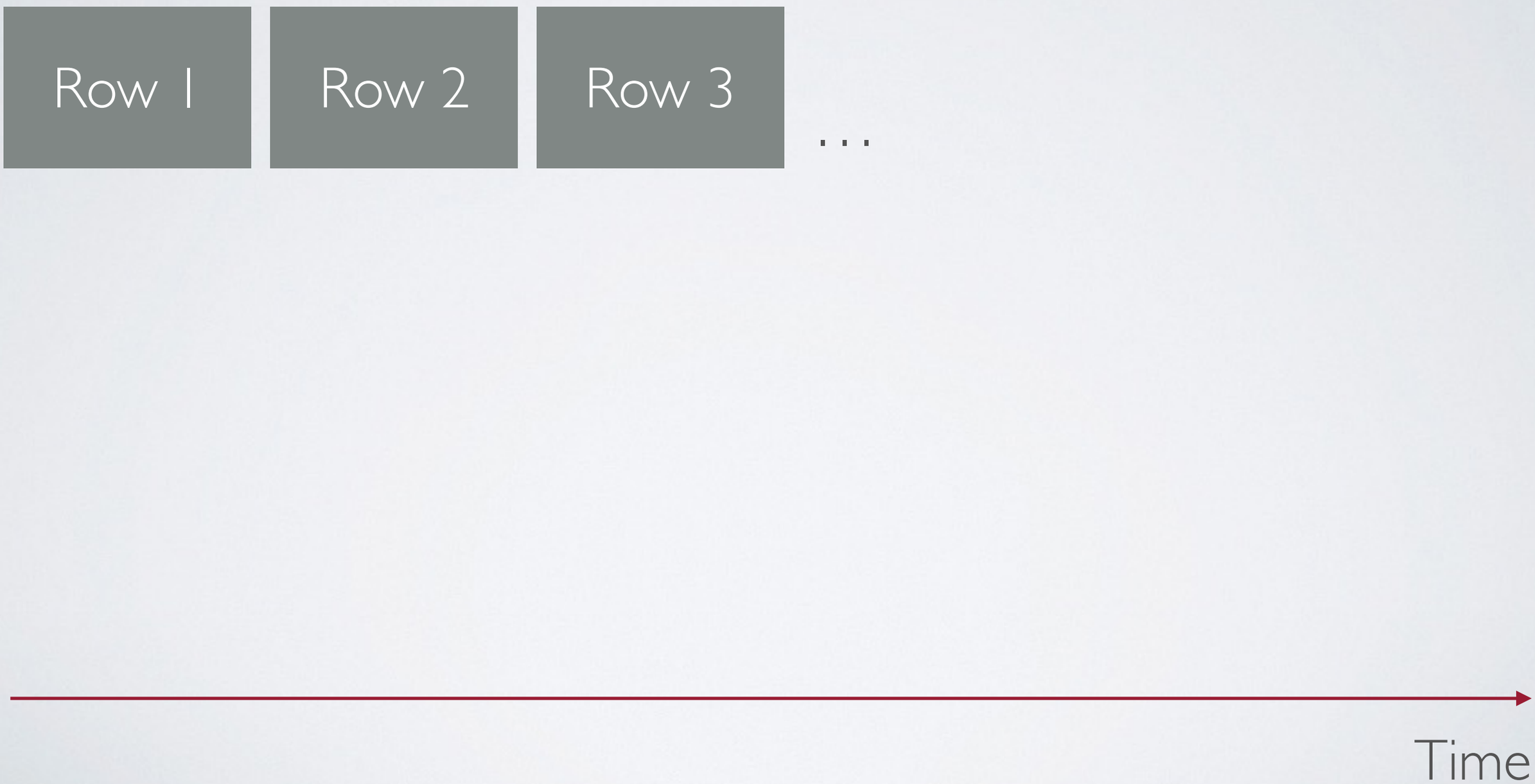
BONUS (GCD)



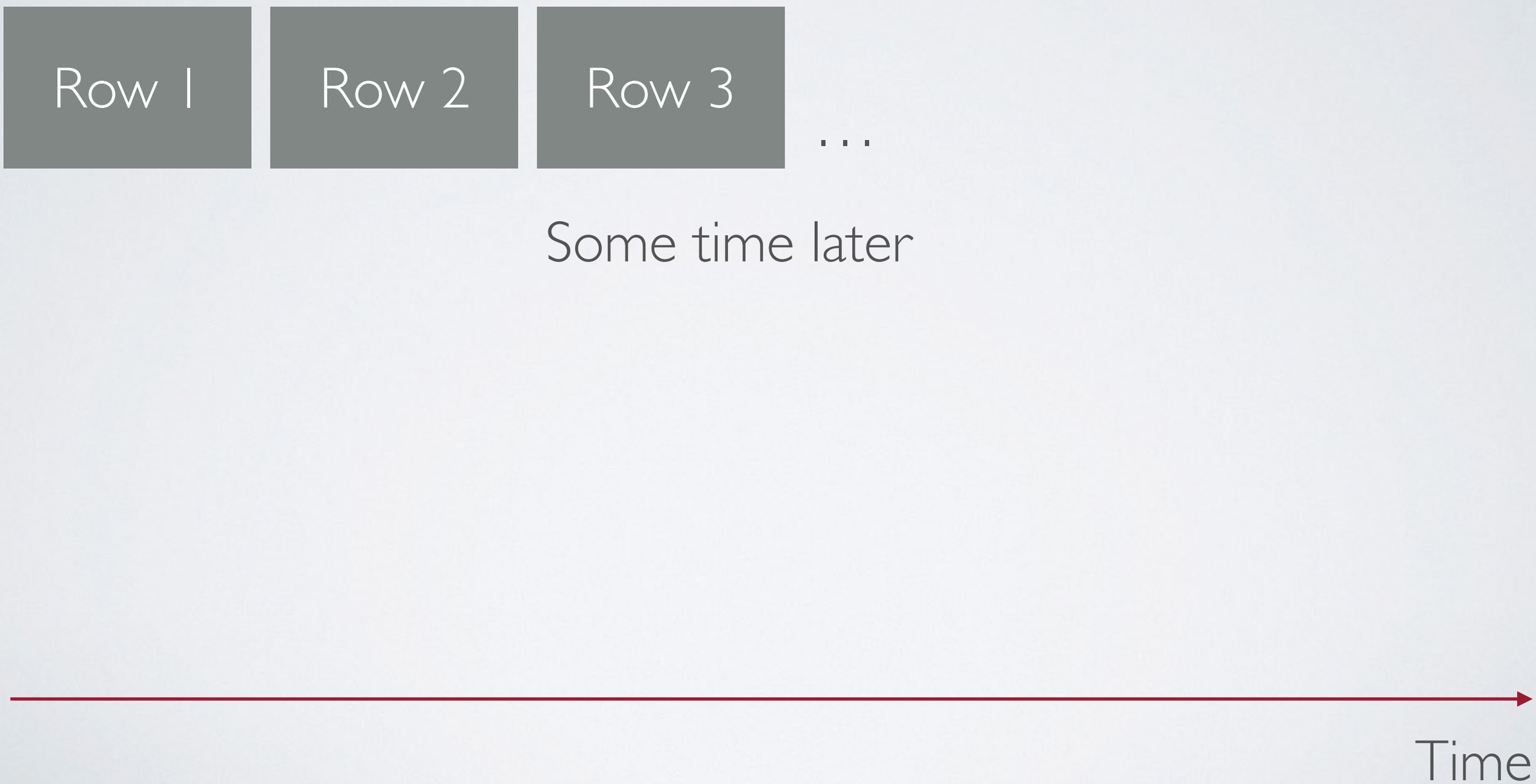
BONUS (GCD)



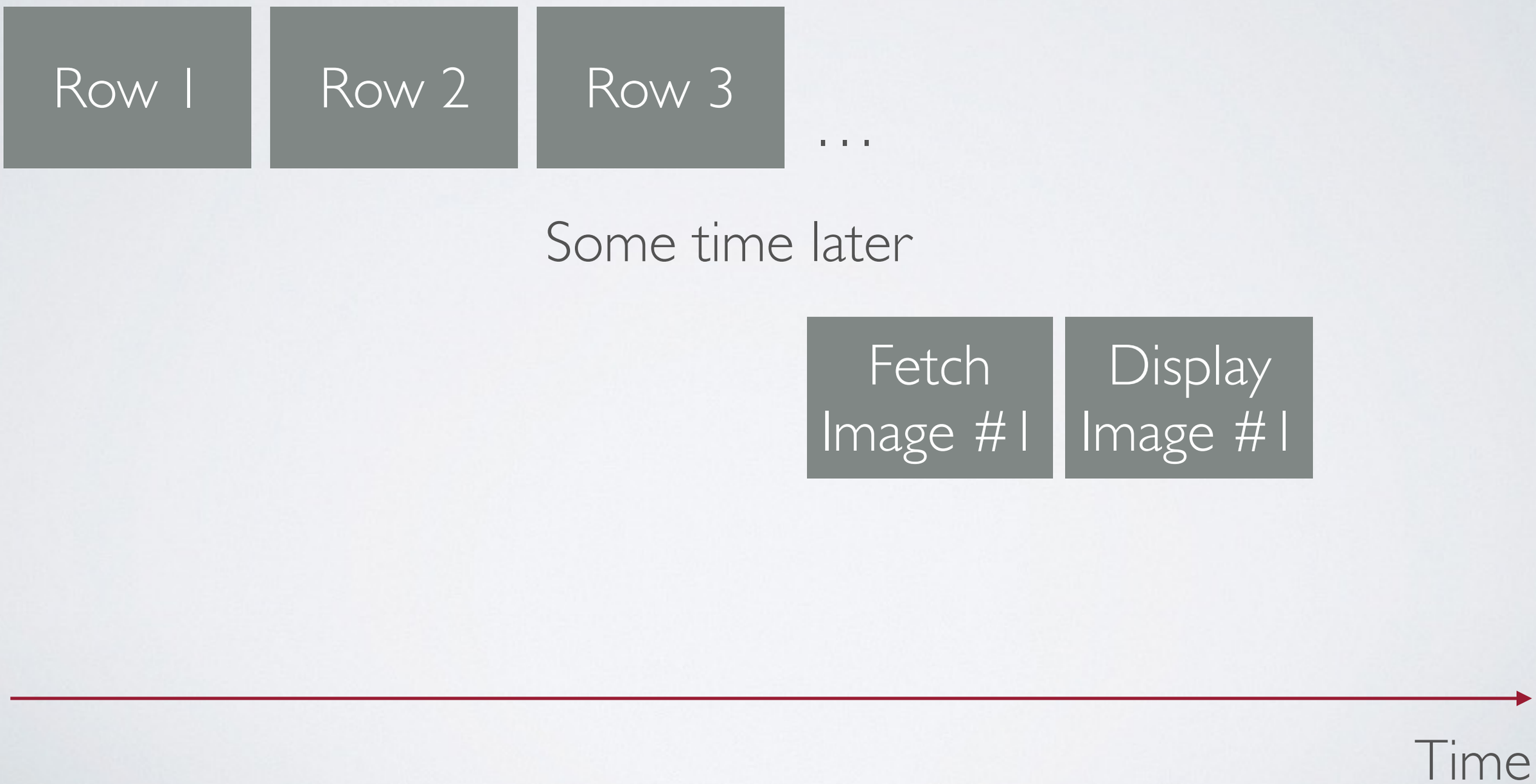
BONUS (GCD)



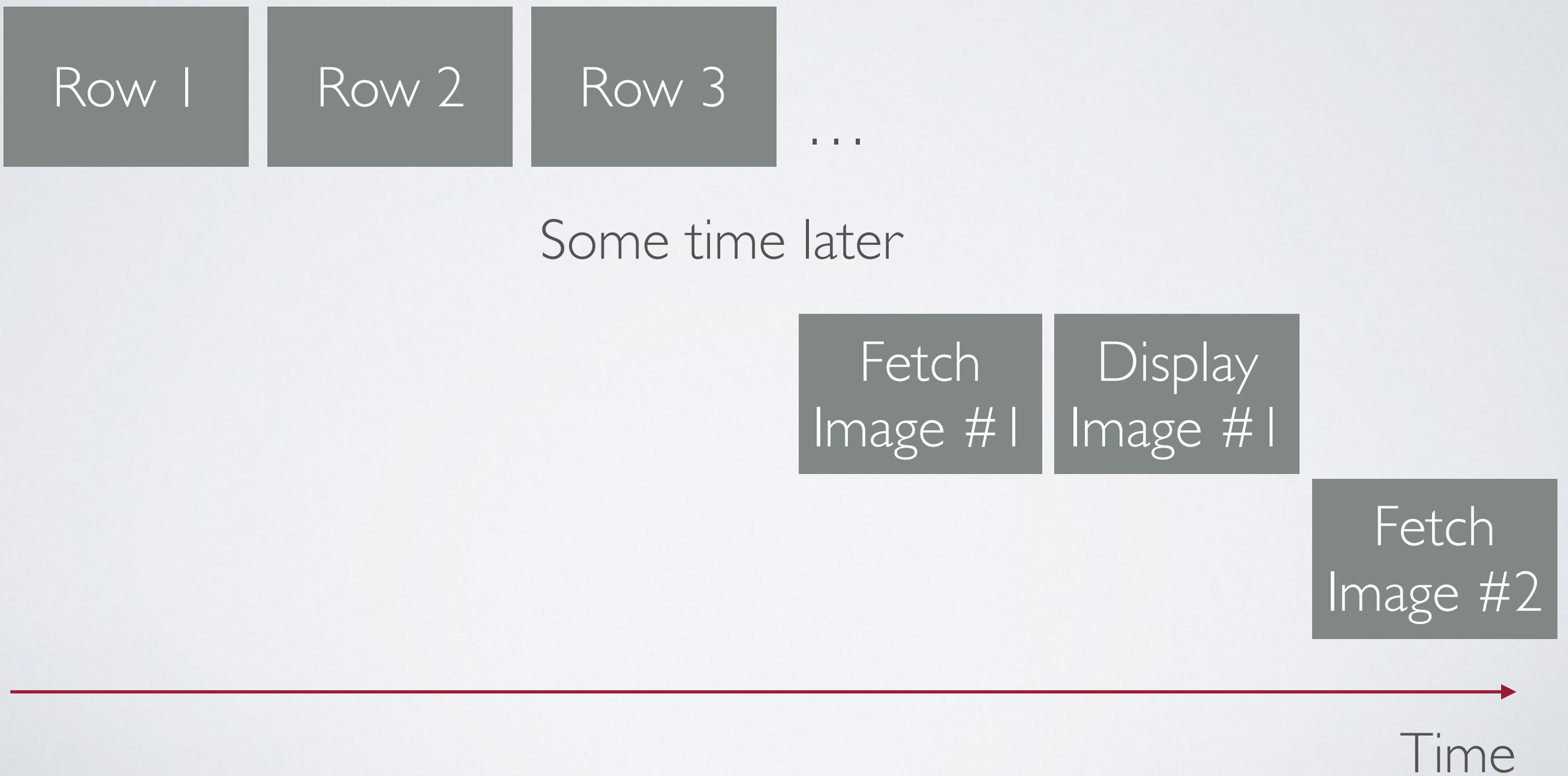
BONUS (GCD)



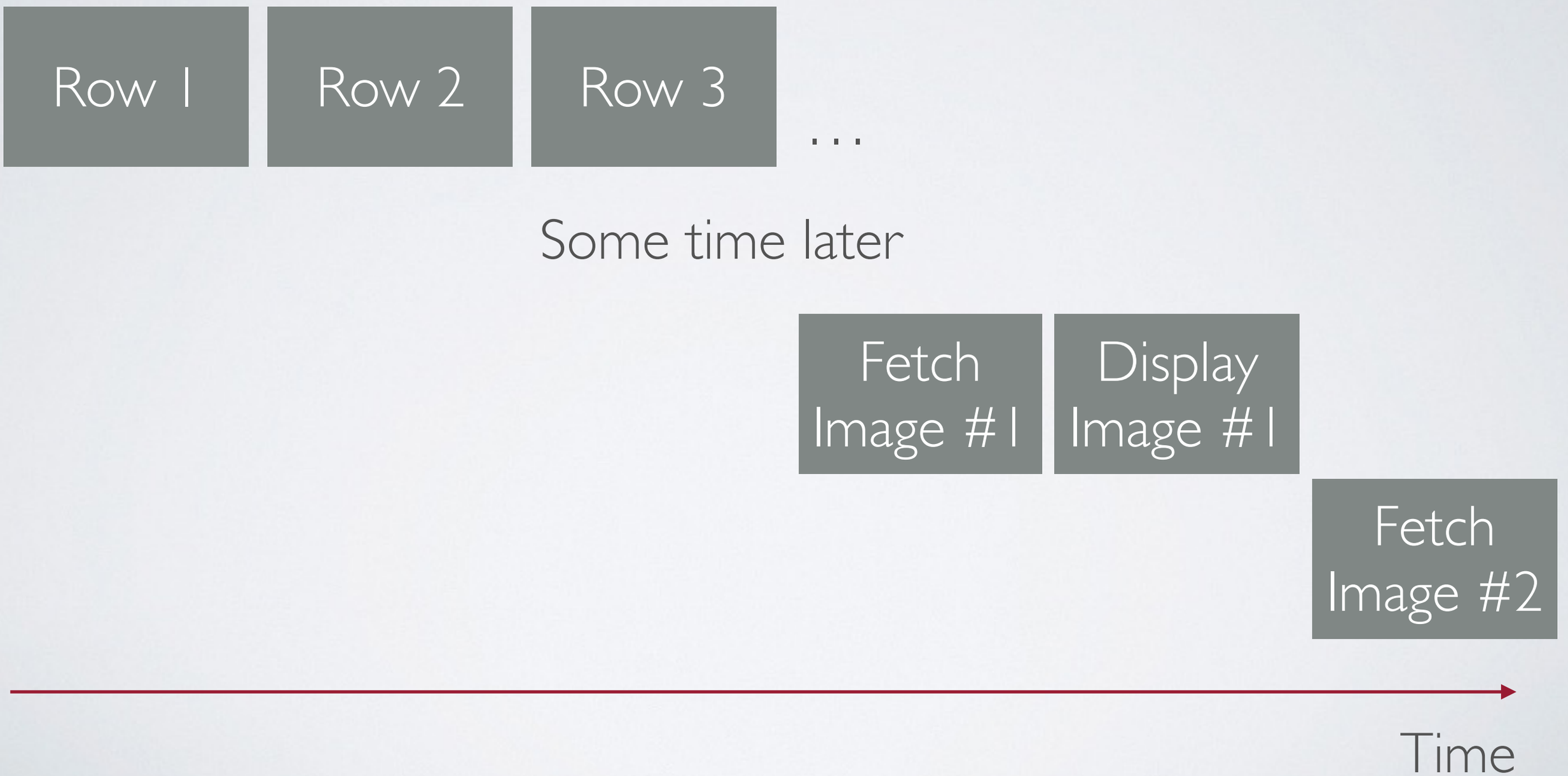
BONUS (GCD)


Row 1
Row 2
Row 3
...
Some time later
Fetch Image #1
Display Image #1
Time

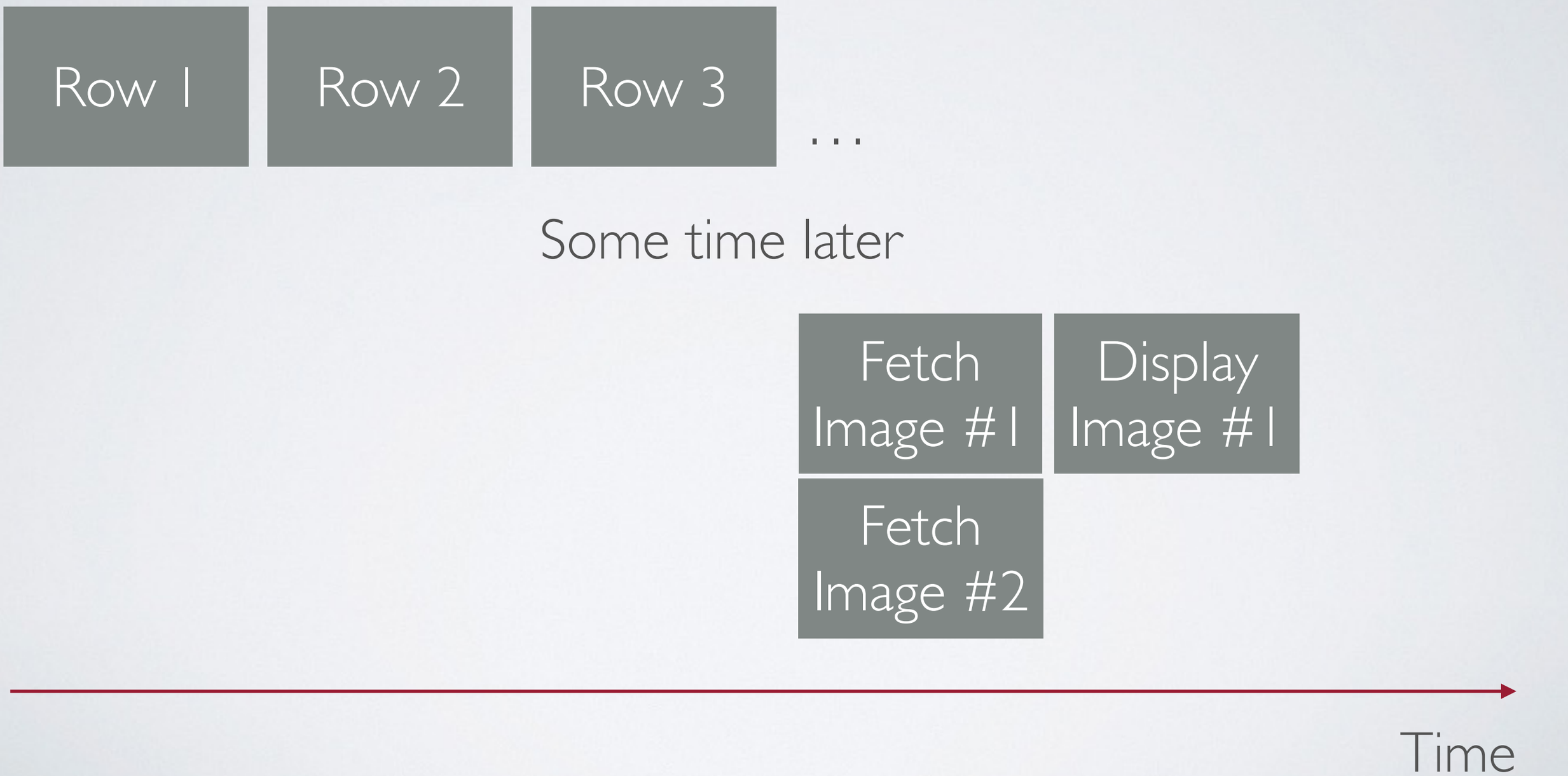
BONUS (GCD)


Row 1
Row 2
Row 3
...
Some time later
Fetch Image #1
Display Image #1
Fetch Image #2
Time

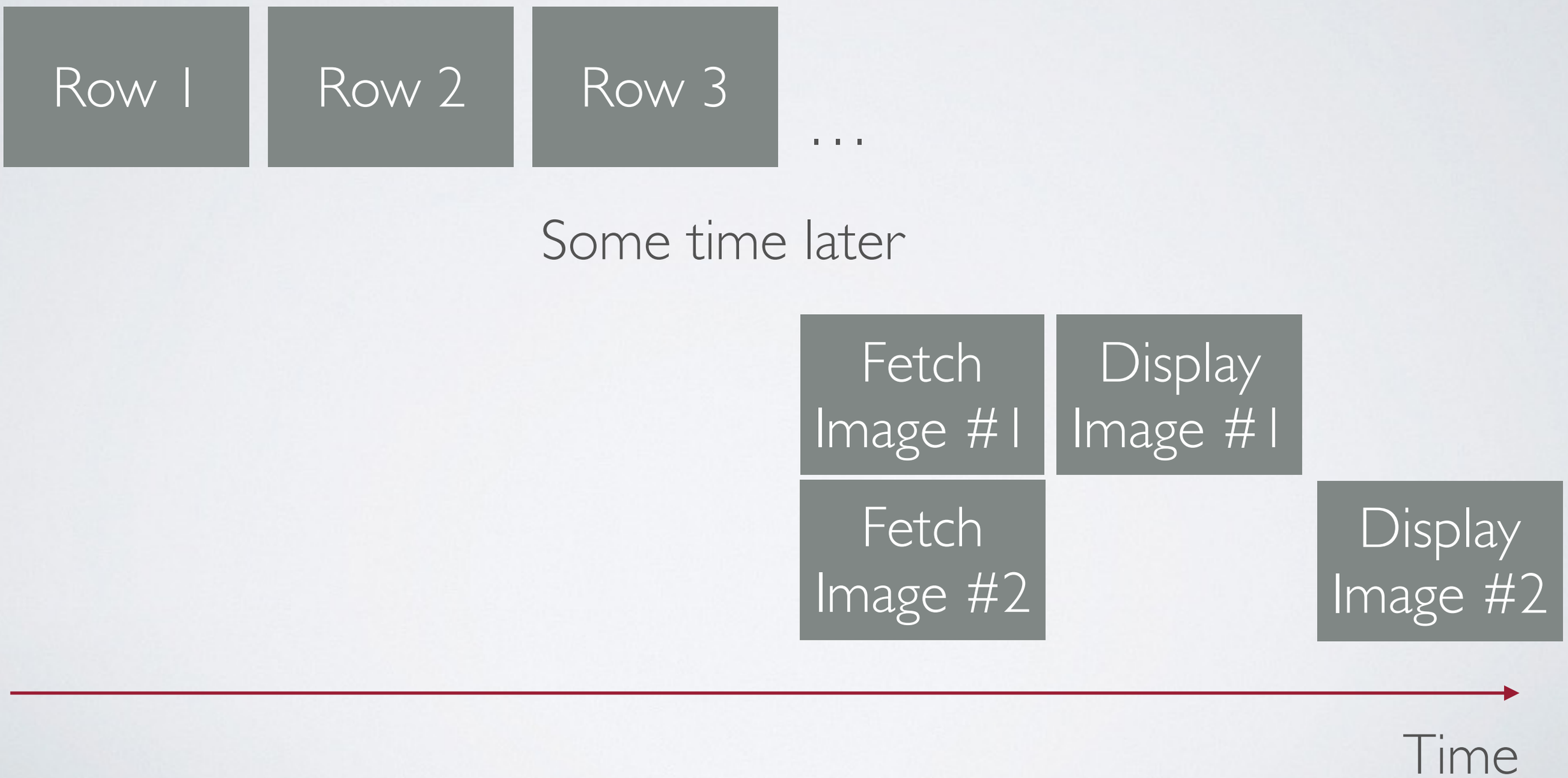
BONUS (GCD)


Row 1
Row 2
Row 3
...
Some time later
Fetch Image #1
Display Image #1
Fetch Image #2
Time

BONUS (GCD)


Row 1
Row 2
Row 3
...
Some time later
Fetch Image #1
Display Image #1
Fetch Image #2
Time

BONUS (GCD)


Row 1
Row 2
Row 3
...
Some time later
Fetch Image #1
Display Image #1
Fetch Image #2
Display Image #2
Time

BONUS (GCD)

Running GitHubDemo on iPhone 6

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:cellForRowAtIndexPath:)

```
let member = members[indexPath.row]
cell.textLabel?.text = member.login

if let url = NSURL(string: member.avatarUrl) {
    dispatch_async(dispatch_get_global_queue(Int(QOS_CLASS_USER_INITIATED.
        rawValue), 0)) { // Dispatch on this thread
        if let data = NSData(contentsOfURL: url){
            if let avatarSquare = UIImage(data: data) {
                dispatch_async(dispatch_get_main_queue()) { // When we
                    finally want to display the image use UI thread
                    let avatarCircle = UIImage.roundedRectImageFromImage
                        (avatarSquare, imageSize: avatarSquare.size,
                        cornerRadius: avatarSquare.size.width / 2)
                    cell.imageView?.image = avatarCircle
                }
            }
        }
    }
}
```

GitHubDemo

BONUS (GCD)

Running GitHubDemo on iPhone 6

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:cellForRowAtIndexPath:)

```
cell.textLabel?.text = member.login

if let url = NSURL(string: member.avatarUrl) {
    dispatch_async(dispatch_get_global_queue(
        rawValue), 0)) { // Dispatch on this t
        if let data = NSData(contentsOfURL: ur
            if let avatarSquare = UIImage(data: data) {
                dispatch_async(dispatch_get_main_queue()) { // When we
                    finally want to display the image use UI thread
                    let avatarCircle = UIImage.roundedRectImageFromImage
                        (avatarSquare, imageSize: avatarSquare.size,
                        cornerRadius: avatarSquare.size.width / 2)
                    cell.imageView?.image = avatarCircle
                }
            }
        }
    }
}
```

A different queue

GitHubDemo

BONUS (GCD)

Running GitHubDemo on iPhone 6

GitHubDemo 〉 GitHubDemo 〉 GitHubMembersTableViewController.swift 〉 tableView(_:cellForRowAtIndexPath:)

```
let member = members[indexPath.row]
cell.textLabel?.text = member.login

if let url = NSURL(string: member.avatarUrl) {
    dispatch_async(dispatch_get_global_queue(
        rawValue), 0)) { // Dispatch on this t
        if let data = NSData(contentsOfURL: ur
            if let avatarSquare = UIImage(data: data)
                dispatch_async(dispatch_get_main_qu
                    finally want to display the image
                    let avatarCircle = UIImage.rounde
                        (avatarSquare, imageSize: avatarSquare.size,
                        cornerRadius: avatarSquare.size.width / 2)
                    cell.imageView?.image = avatarCircle
                }
            }
        }
    }
}
```

A different queue

The main queue

GitHubDemo

BONUS (TESTING TIPS)

- Adroit Annie - does my application work for power users?
- Boring Bob - does my application work for normal users? Write black box and white box tests.
- Clueless Charlie - does my application work for novice users? Do user study.
- Diabolical Dave - does my application withstand attacks from diabolical users? Do thread modeling, fuzz testing, etc.

NEXT TIME

- Revisit auto-layouts to see some other examples
- How to persist data/edit data in table views