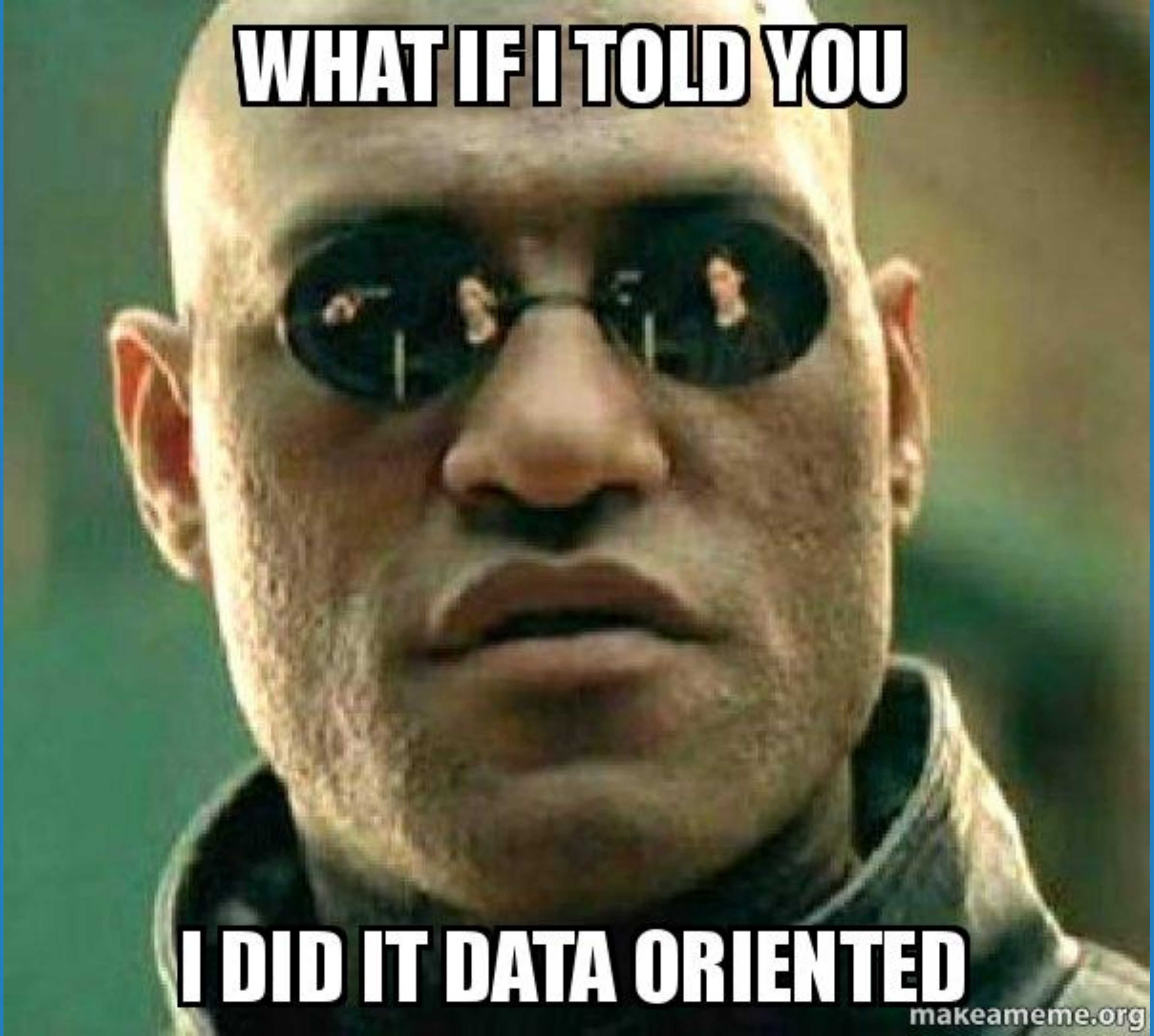
UIKonf App Architecture

&

Data Oriented Design

HOW DOES ONE SIMPLY IMPLEMENT
CONFERENCE APP

```
+------------+   +----------------+
|            |   |                |
| Objects    |   | Data           |
+------------+   +----------------+
| Talk       |   | Title          |
| Speaker    |   | Description    |
| Organizer  |   | StartTime      |
| Volunteer  |   | EndTime        |
| Venue      |   | Name           |
| Break      |   | TwitterHandle  |
| Workshop   |   | PhotoURL       |
| ...        |   | ...            |
|            |   |                |
+------------+   +----------------+
```


WHAT IF I TOLD YOU
I DID IT DATA ORIENTED
makeameme.org

Time slot

```
[
  {
  "t_id": "startDay1",
  "startTime": "18-09-00",
  "endTime": "18-10-00",
  "description": "Check in first day",
  "locations": [
                "Heimathafen Neukölln"
                ]
  },
  ...
```

Organizer

```
...
{
"name": "Maxim Zaks",
"twitter": "@iceX33",
"bio": "Software developer with a history in IDE development,
  Web development and even Enterprise Java development
  (He was young and under bad influence).
  Nowadays working as a game developer (preferably iOS).
  Regular visitor and occasional speaker at conferences.",
"photo": "http://www.uikonf.com/static/images/maxim-zaks.png",
"organizer": true
},
...
```

Speaker

```
...
{
"name": "Graham Lee",
"twitter": "@iwasleeg",
"bio": "Graham Lee works at Facebook, where he helps people make better tests
  so they can help people make better software.
  In the past he worked with some other people,
  and has written books and blogs so he can work
  with people he hasn't met too. His blog is at sicpers.info.",
"photo": "http://www.uikonf.com/static/images/Graham-Lee.png"
},
...
```

Talk

```
...
{
"title": "World Modeling",
"speaker_name": "Mike Lee",
"t_id": "session1Day1",
"t_index": 1
},
...
```

Location

```
...
{
"name": "Heimathafen Neukölln",
"address": "Karl-Marx-Str. 141, 12043 Berlin",
"description": "Conference venue"
},
...
```

Import Data from JSON Array

```
for item in jsonArray {
    let entity = context.createEntity()

    for pair in (item as! NSDictionary) {
        let (key,value) = (pair.key as! String,
                           pair.value as! JsonValue)

        let component = converters[key]!(value)
        entity.set(component)
    }
}
```

What is a context?

It's a managing data structure

```swift
public class Context {
    public func createEntity() -> Entity
    public func destroyEntity(entity : Entity)
    public func entityGroup(matcher : Matcher) -> Group
}
```

What's an entity?

Bag of components

Entity

```
public class Entity {
    public func set(c:Component, overwrite:Bool = false)
    public func get<C:Component>(ct:C.Type) -> C?
    public func has<C:Component>(ct:C.Type) -> Bool
    public func remove<C:Component>(ct:C.Type)
}
```

What's a component

It's just data (value object)

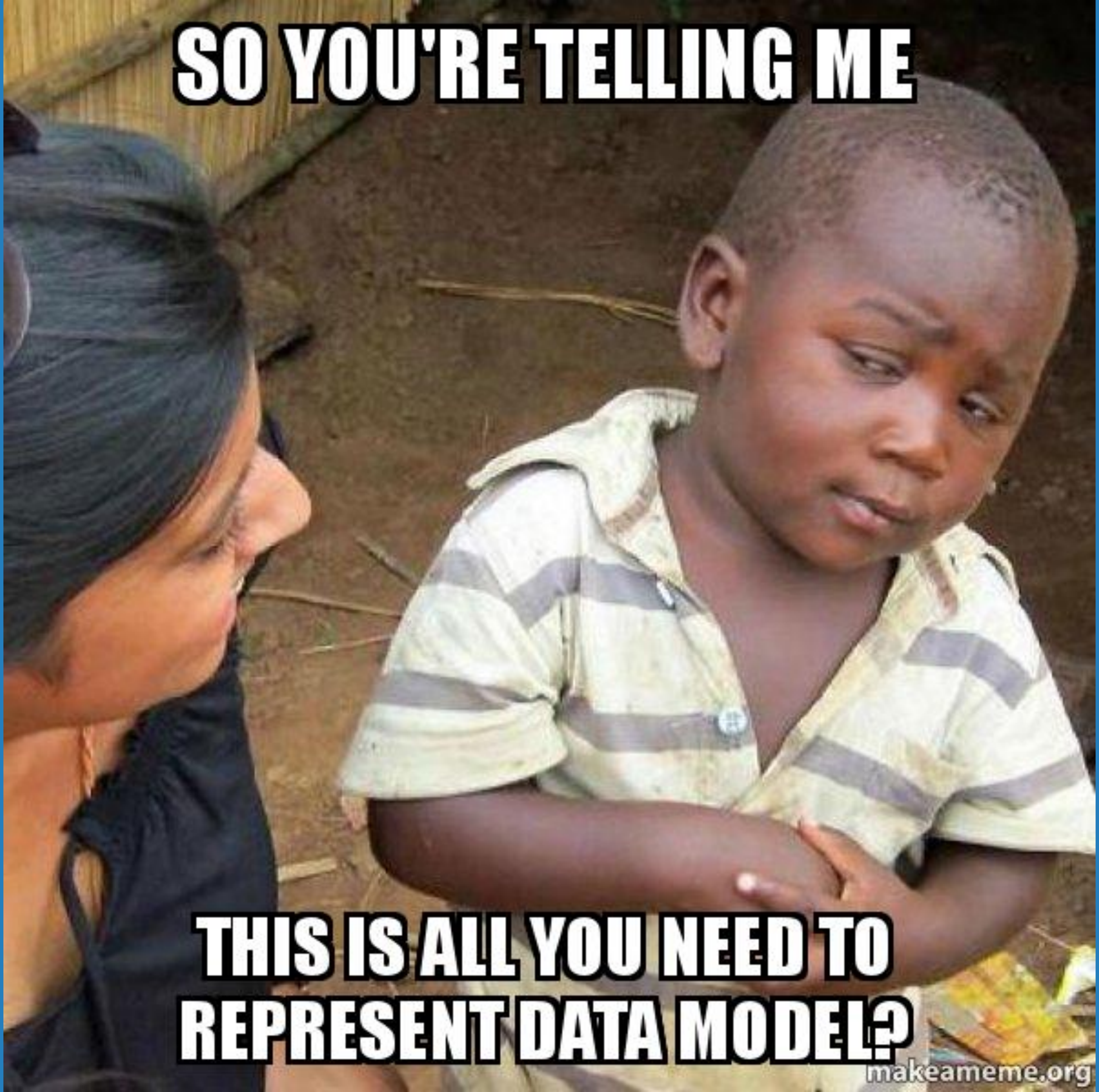
Components

```
struct NameComponent : Component, DebugPrintable {
    let name : String
    var debugDescription: String{
        return "[\(name)]"
    }
}
```

What's a component

It also can be just a flag

```
struct OrganizerComponent : Component {}
```


SO YOU'RE TELLING ME
THIS IS ALL YOU NEED TO
REPRESENT DATA MODEL?
makeameme.org

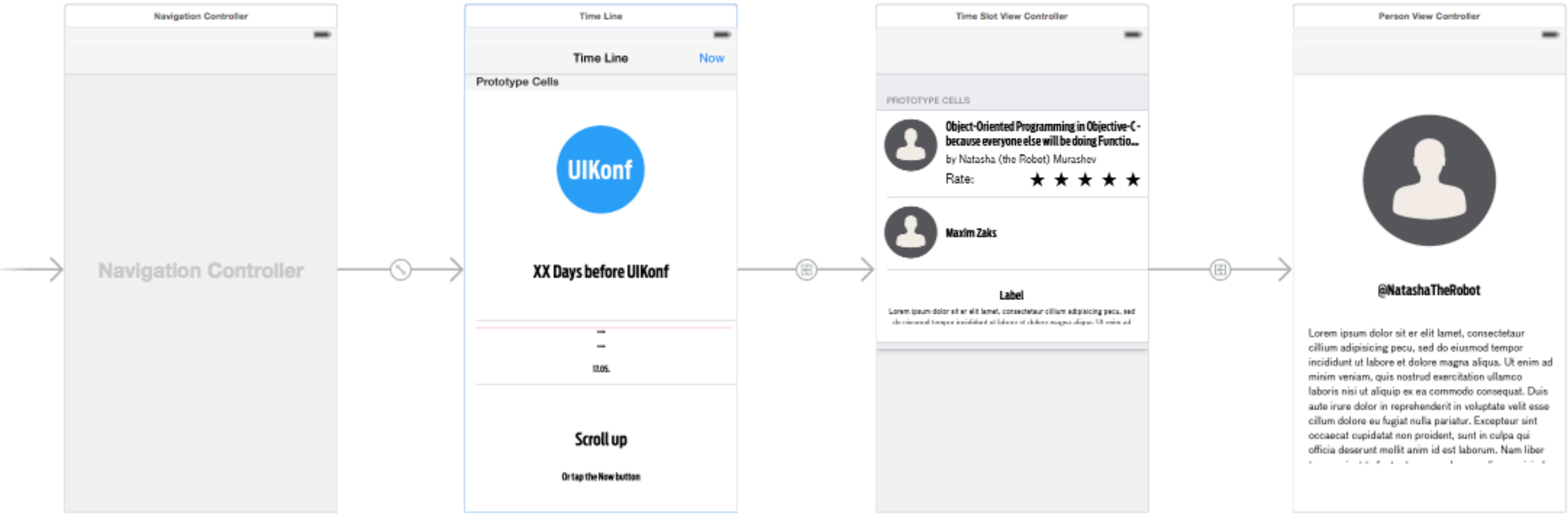
Import Data from JSON Array

```
for item in jsonArray {
    let entity = context.createEntity()

    for pair in (item as! NSDictionary) {
        let (key,value) = (pair.key as! String,
                           pair.value as! JsonValue)

        let component = converters[key]!(value)
        entity.set(component)
    }
}
```

Converters dictionary

```
typealias Converter = (JsonValue) -> Component

let converters : [String : Converter] = [
    "t_id" : {
        TimeSlotIdComponent(id: $0 as! String)
    },
    "t_index" : {
        TimeSlotIndexComponent(index: $0 as! Int)
    },
    ...
```

How does it work with UIKit



Lorem ipsum dolor sit er elit lamet, consectetaur cillium adipisicing pecu, sed

Lorem ipsum dolor sit er elit lamet, consectetaur cillium adipisicing pecu, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Nam liber

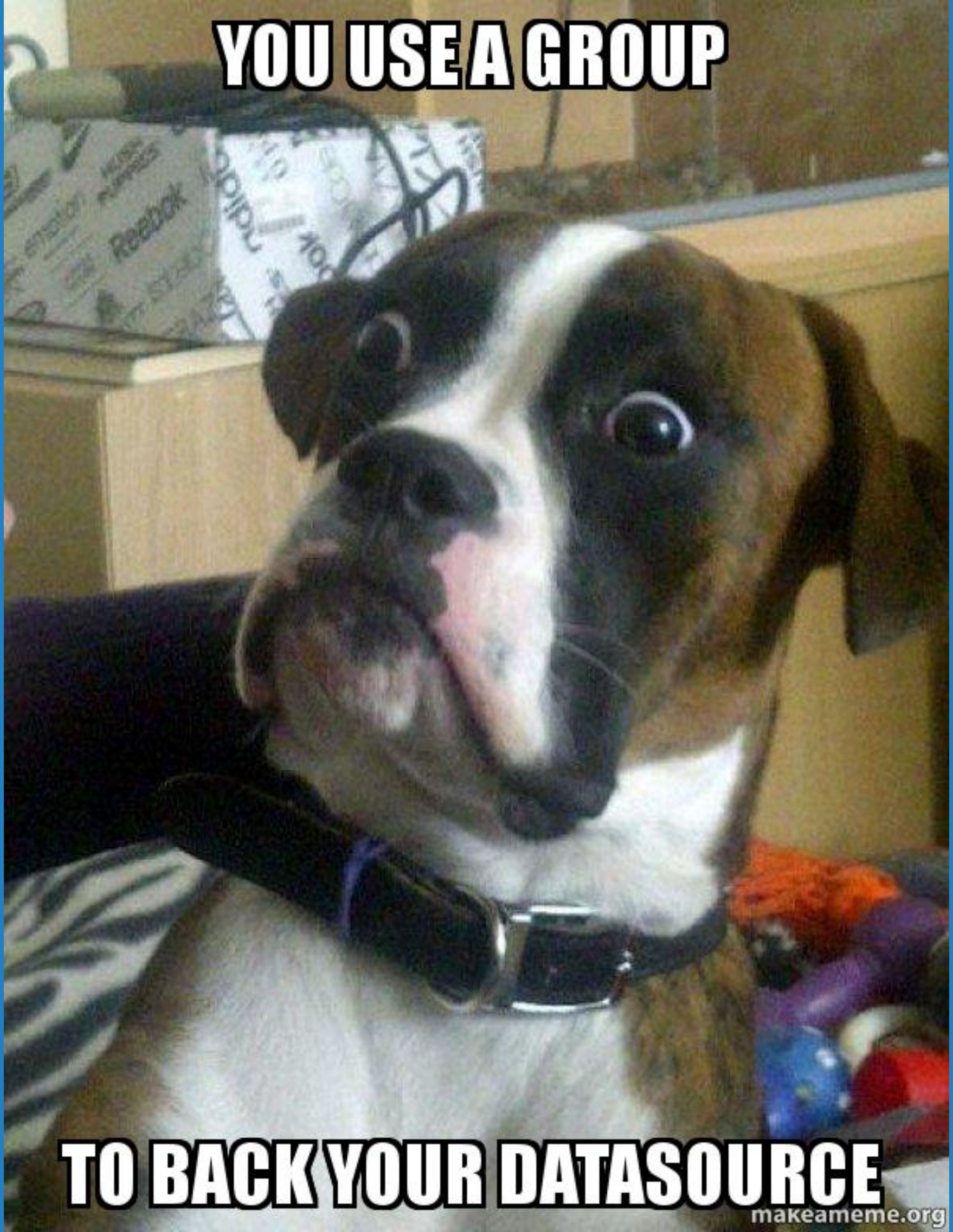
```
override func viewDidLoad() {
    super.viewDidLoad()

    groupOfEvents = context.entityGroup(
        Matcher.Any(StartTimeComponent, EndTimeComponent))

    setNavigationTitleFont()

    groupOfEvents.addObserver(self)

    context.entityGroup(
        Matcher.All(RatingComponent)).addObserver(self)

    readDataIntoContext(context)

    syncData(context)

}
```

What's a group?

Subset of Entites

Entity

```
public class Group : SequenceType {
    public var count : Int
    public var sortedEntities: [Entity]
    public func addObserver(observer : GroupObserver)
    public func removeObserver(observer : GroupObserver)
}
```


YOU USE A GROUP
TO BACK YOUR DATASOURCE
makeameme.org

```
override func viewDidLoad() {
    super.viewDidLoad()

    groupOfEvents = context.entityGroup(
        Matcher.Any(StartTimeComponent, EndTimeComponent))

    setNavigationTitleFont()

    groupOfEvents.addObserver(self)

    context.entityGroup(
        Matcher.All(RatingComponent)).addObserver(self)

    readDataIntoContext(context)

    syncData(context)

}
```

THIS IS LIKE KVO
IN OBJC ... BUT IN SWIFT
makeameme.org

```
extension TimeLineViewController : GroupObserver {

    func entityAdded(entity : Entity) {
        if entity.has(RatingComponent){
            updateSendButton()
        } else {
            reload()
        }
    }

    func entityRemoved(entity : Entity,
        withRemovedComponent removedComponent : Component) {
        if removedComponent is RatingComponent {
            return
        }
        reload()
    }
}
```

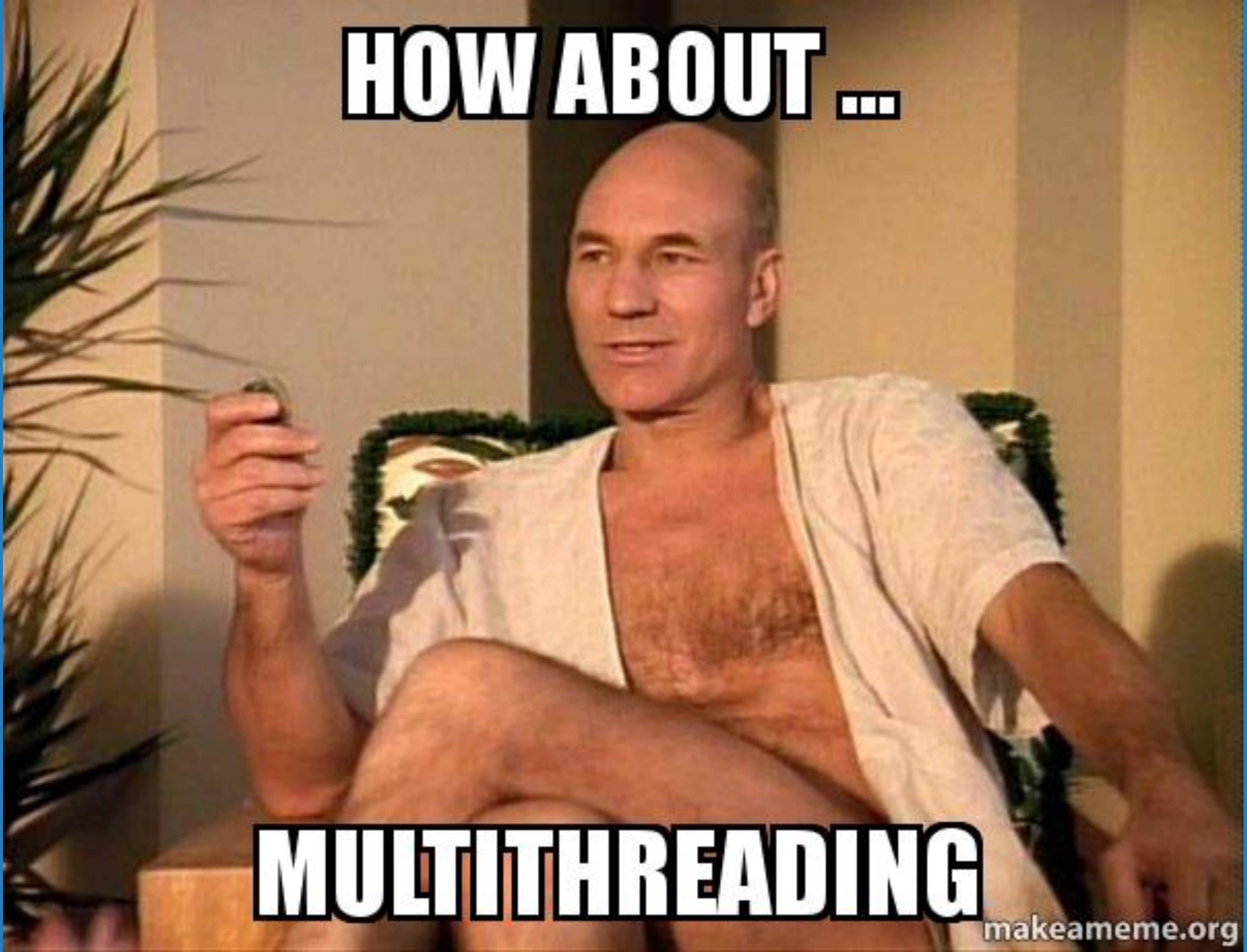
```
private lazy var reload :
    dispatch_block_t = dispatch_debounce_block(0.1) {
    ...
}
```

MAY I COME IN?
NOPE!
makeameme.org

COOL STUFF
CONTINUE WITH UIKIT INTEGRATION
BRILLIANT!
makeameme.org

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

    let sectionName = sectionNameTable[indexPath.section]()!
    let cellIdentifier  = cellIdTable[sectionName]!

    let cell  = tableView.dequeueReusableCellWithIdentifier(
                    cellIdentifier, forIndexPath: indexPath) as! EntityCell

    let cellEntity = cellEntityTable[sectionName]!(indexPath.row)

    cell.updateWithEntity(cellEntity, context: context)

    return cell as! UITableViewCell
}
```

```swift
protocol EntityCell {
    func updateWithEntity(entity : Entity, context : Context)
}

class LocationCell: UITableViewCell, EntityCell {

    @IBOutlet weak var nameLabel: UILabel!
    @IBOutlet weak var descriptionLabel: UITextView!

    func updateWithEntity(entity : Entity, context : Context){

        nameLabel.text = entity.get(NameComponent)!.name
        let descriptionText = entity.get(DescriptionComponent)?.description
        let address = entity.get(AddressComponent)!.address

        descriptionLabel.text = descriptionText != nil ?
                                descriptionText! + "\n" + address : address
    }
}
```


HOW ABOUT ...
MULTITHREADING
makeameme.org

```
struct PhotoComponent : Component, DebugPrintable {
    let url : NSURL
    let image : UIImage
    let loaded : Bool
    var debugDescription: String{
        return "[\(url), loaded: \(loaded)]"
    }
}
```

Data:

```
{
    ...
    "photo": "http://www.uikonf.com/static/images/maxim-zaks.png",
    ...
}
```

Converter:

```
"photo" : {
  PhotoComponent(url: NSURL(string:$0 as! String)!,
                image : UIImage(named:"person-icon")!, loaded: false)
},
func setPhoto() {
    let photoComponent = entity!.get(PhotoComponent)!
    imageView.image = photoComponent.image
    if !photoComponent.loaded {
        ...
    }
}
```

```
var detachedPerson = entity!.detach
cancelLoadingPhoto =
    dispatch_after_cancellable(
        0.5, dispatch_get_global_queue(QOS_CLASS_DEFAULT, 0))
    {
        if  let data = NSData(contentsOfURL: photoComponent.url),
            let image = UIImage(data: data)
            {

            let photoComponent = detachedPerson.get(PhotoComponent)!
            detachedPerson.set(
                PhotoComponent(url: photoComponent.url,
                                image:image, loaded:true),
                overwrite: true
            )
            detachedPerson.sync()
    }
}
```

What's a detached Entity?

It's an Entity implemented as a struct

with a *sync* method

```
var detachedPerson = entity!.detach
cancelLoadingPhoto =
    dispatch_after_cancellable(
        0.5, dispatch_get_global_queue(QOS_CLASS_DEFAULT, 0))
    {
        if  let data = NSData(contentsOfURL: photoComponent.url),
            let image = UIImage(data: data)
            {

            let photoComponent = detachedPerson.get(PhotoComponent)!
            detachedPerson.set(
                PhotoComponent(url: photoComponent.url,
                                image:image, loaded:true),
                overwrite: true
            )
            detachedPerson.sync()
    }
}
```

Recap

- Context is a managing Data Structure
- Entity is a bag of components
- Components are just value types
- Groups are subsets on the components
- You can observe groups -> KVO like behavior
- Use detached entity if you want to go on another queue

Tanks you

Maxim - @iceX33

Links

- UIKonf App on Github
- Blog: Think different about Data Model
- Blog: What is an entity framework
- Book: Data oriented Design (C++ heavy)
- Talk: Data-Oriented Design and C++ (hardcore)