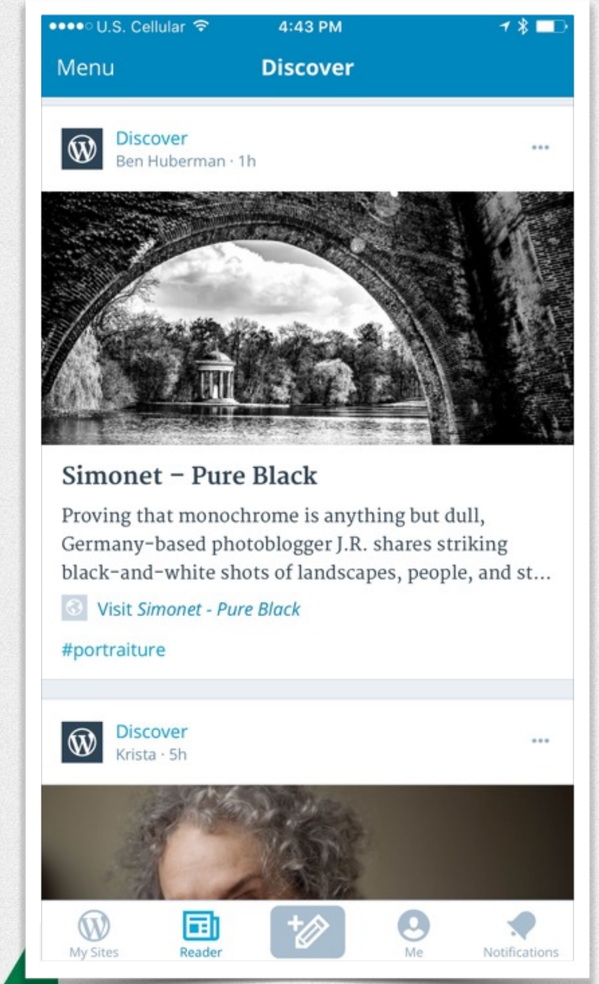
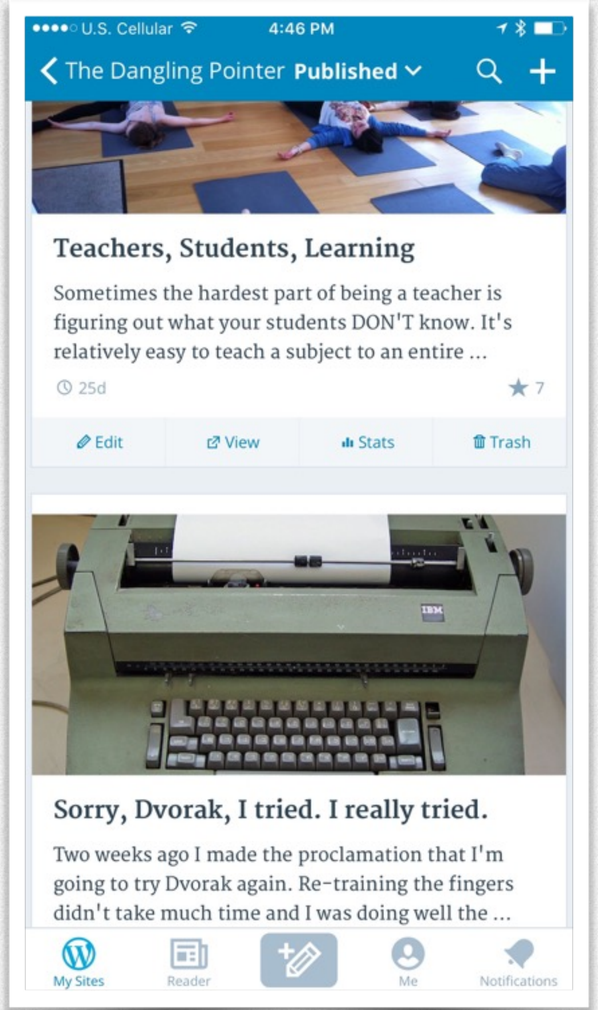
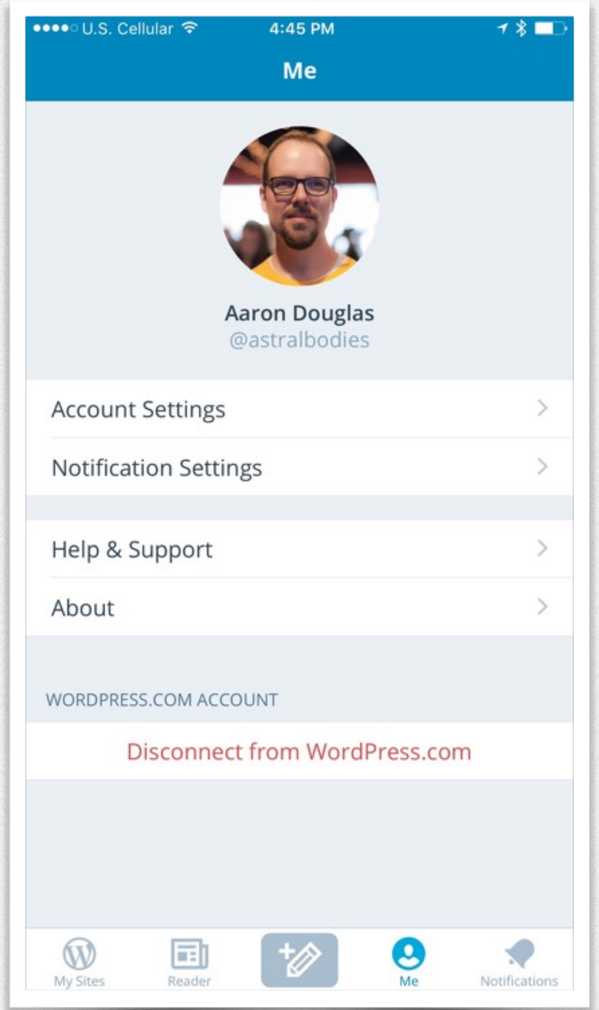
308: Core Data to Online Service Synchronization

Overview

WordPress for iOS







Overview

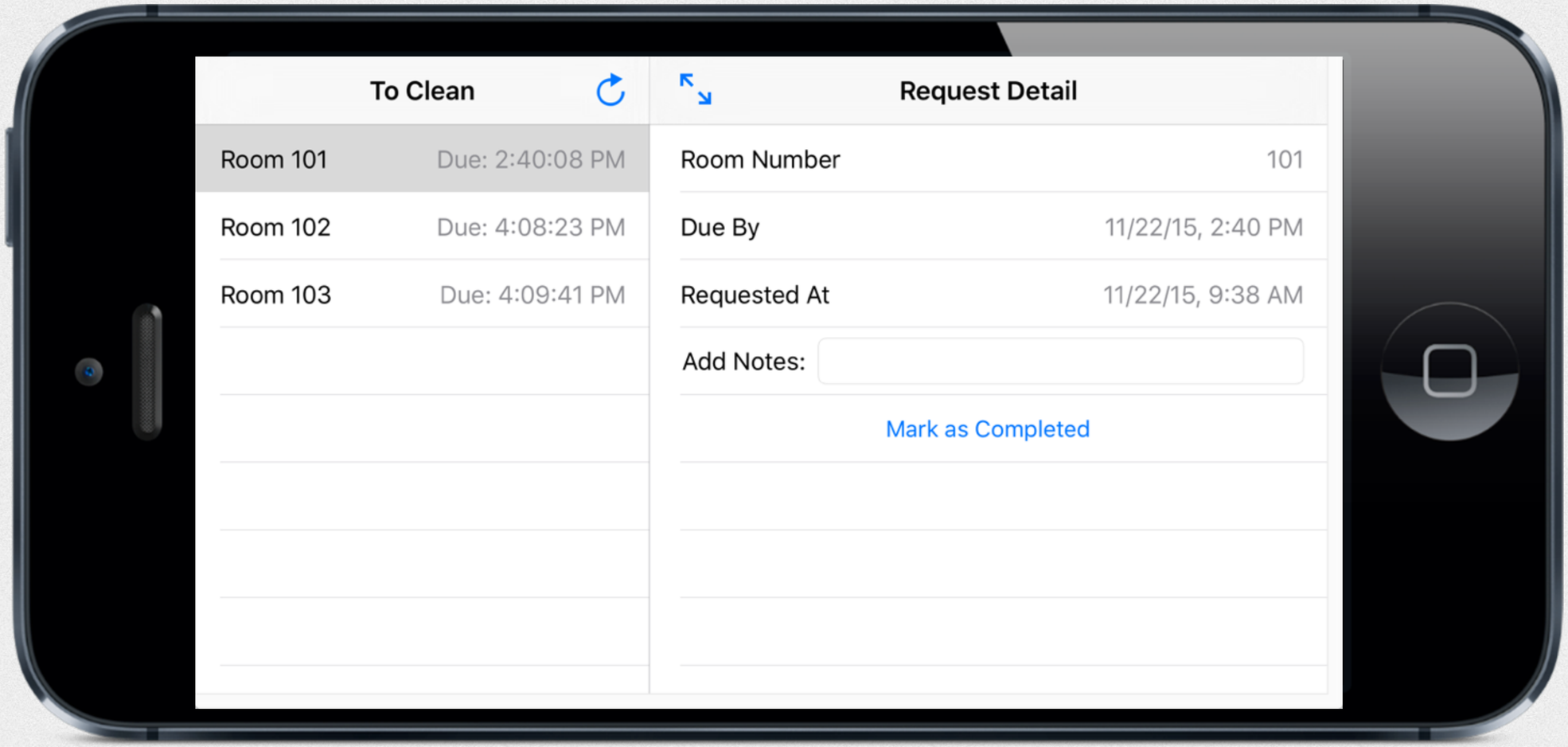
- Clean Rooms app
- Service Architecture
- Data Models
- Synchronization
- Demo
- Lab

CouchDB

- Open Source
- Document-oriented
- NoSQL
- JSON data structure
- HTTP RESTful API

308: Core Data to Online Service Synchronization

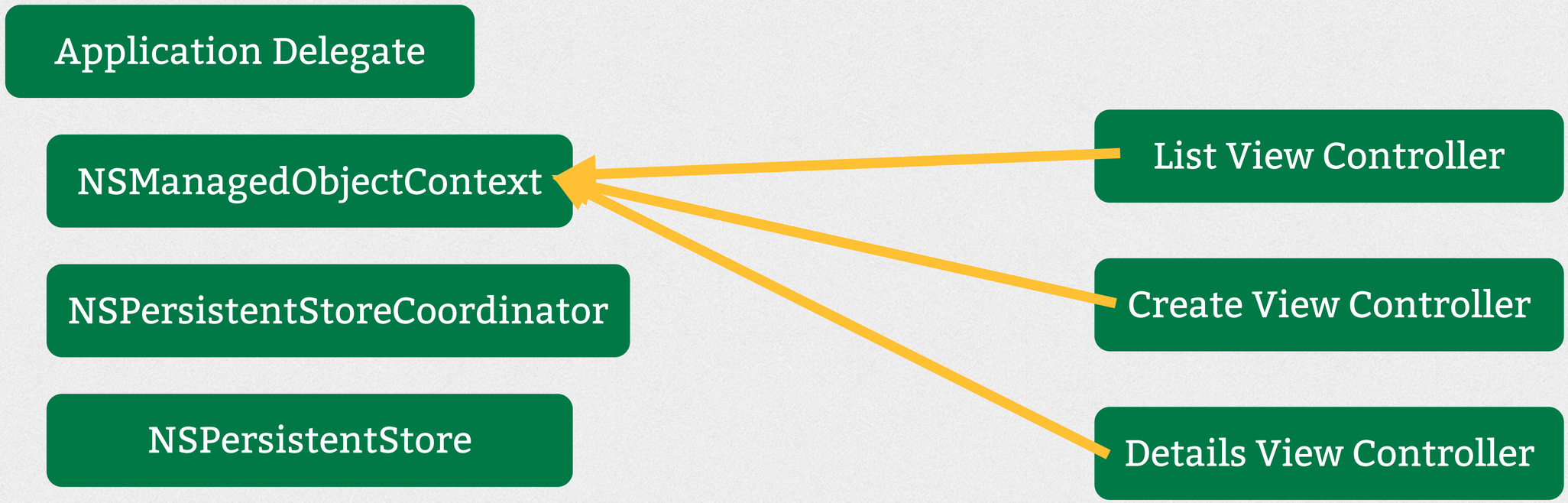
Clean Rooms


To Clean
Room 101
Due: 2:40:08 PM
Room 102
Due: 4:08:23 PM
Room 103
Due: 4:09:41 PM
Request Detail
Room Number
101
Due By
11/22/15, 2:40 PM
Requested At
11/22/15, 9:38 AM
Add Notes:
Mark as Completed

308: Core Data to Online Service Synchronization

Service Architecture

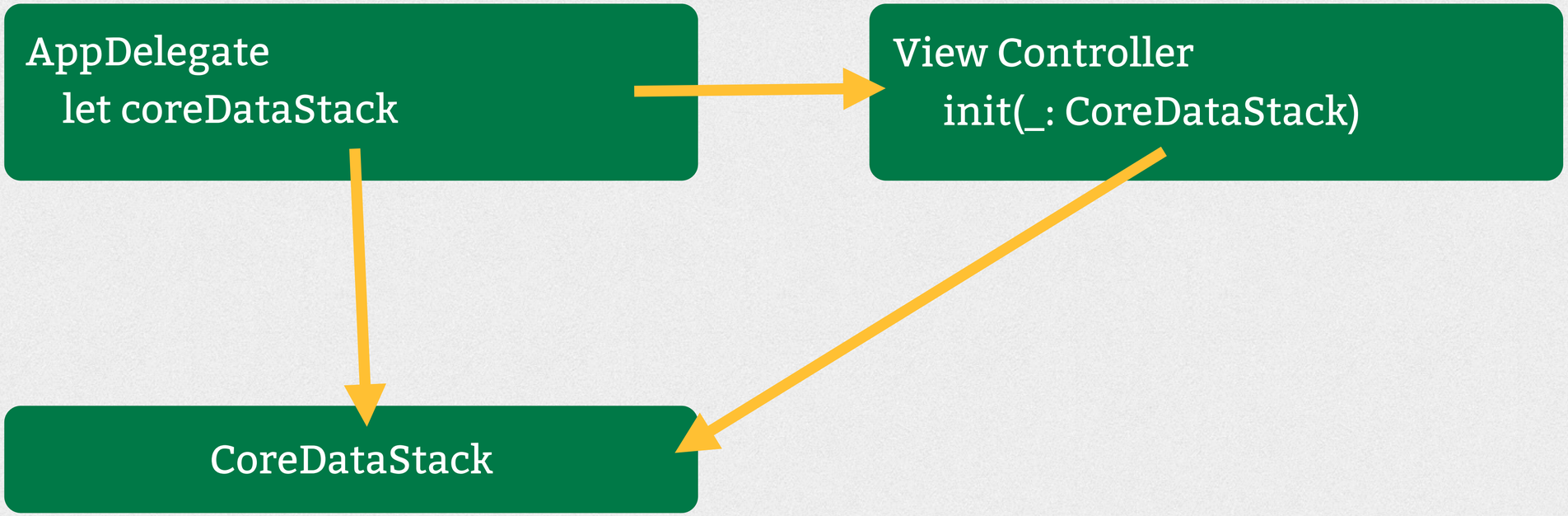
Everything in App Delegate



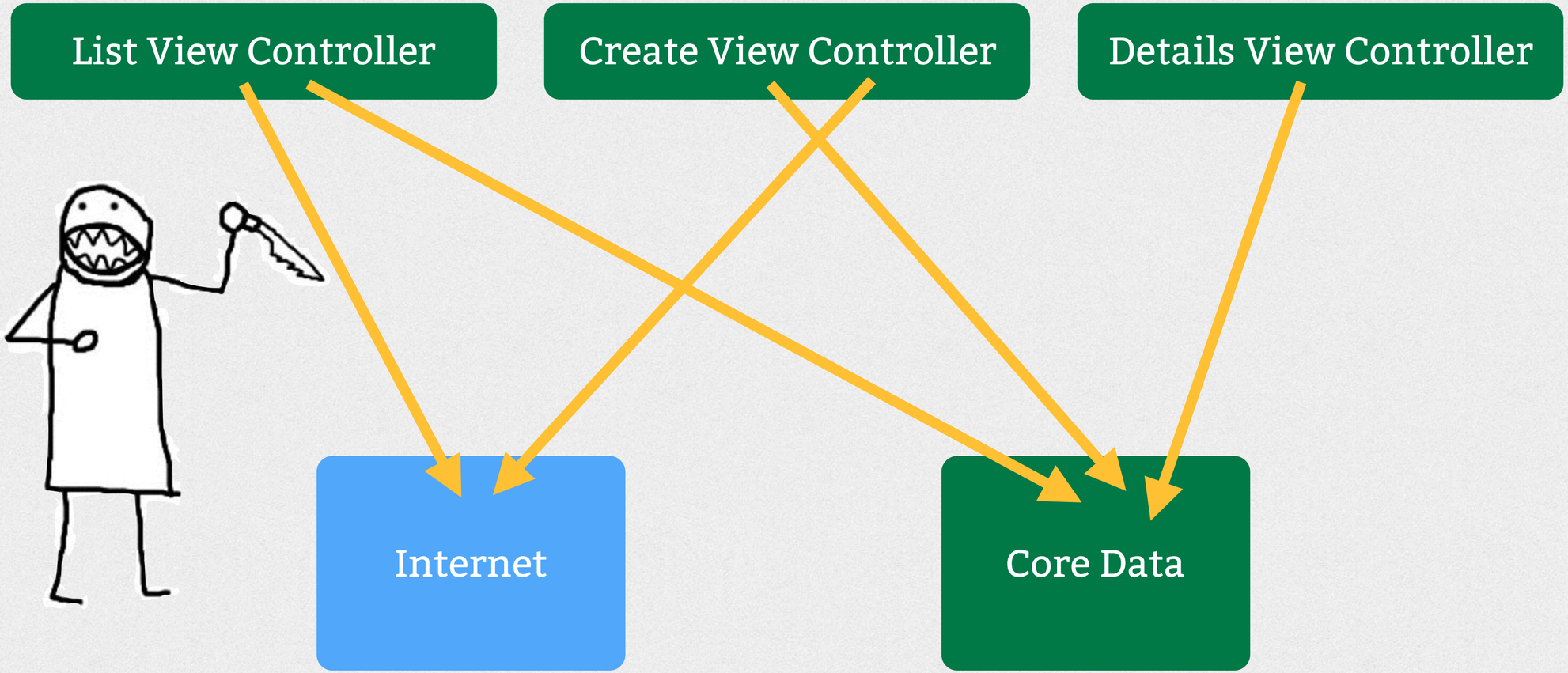
Looks a Little Like This

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

  let appDelegate = UIApplication
    .sharedApplication()
    .delegate as! AppDelegate
  managedObjectContext = appDelegate.managedObjectContext
}
```

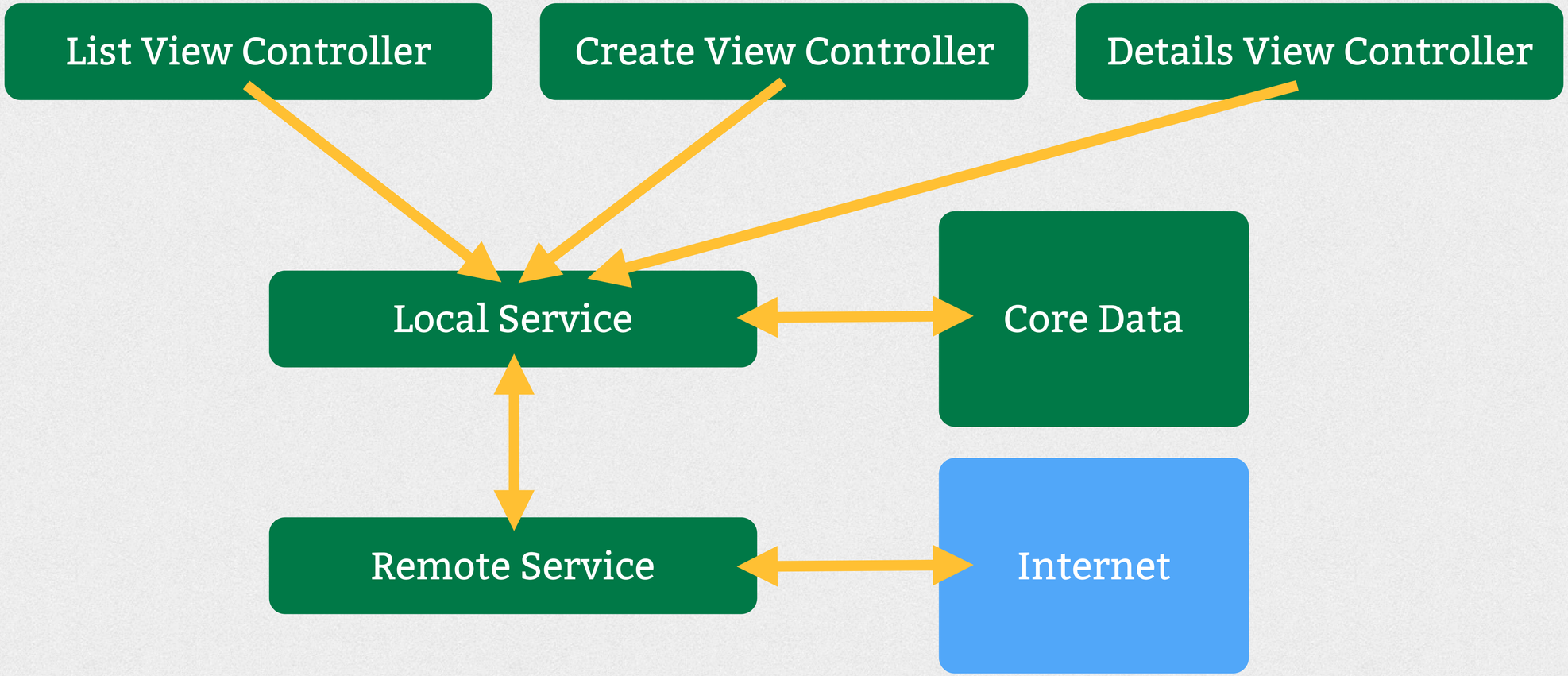
CoreDataStack



Complex View Controllers



Solution


List View Controller
Create View Controller
Details View Controller
Local Service
Core Data
Remote Service
Internet

Service Layers

- Business logic
- Dependencies in init()
- Defines a contract
- Looser coupling
- Unit testability

- Internal API

Local Service

- Persistence to disk or memory
- CRUD operations
- Validation
- Wraps remote calls

Remote Service

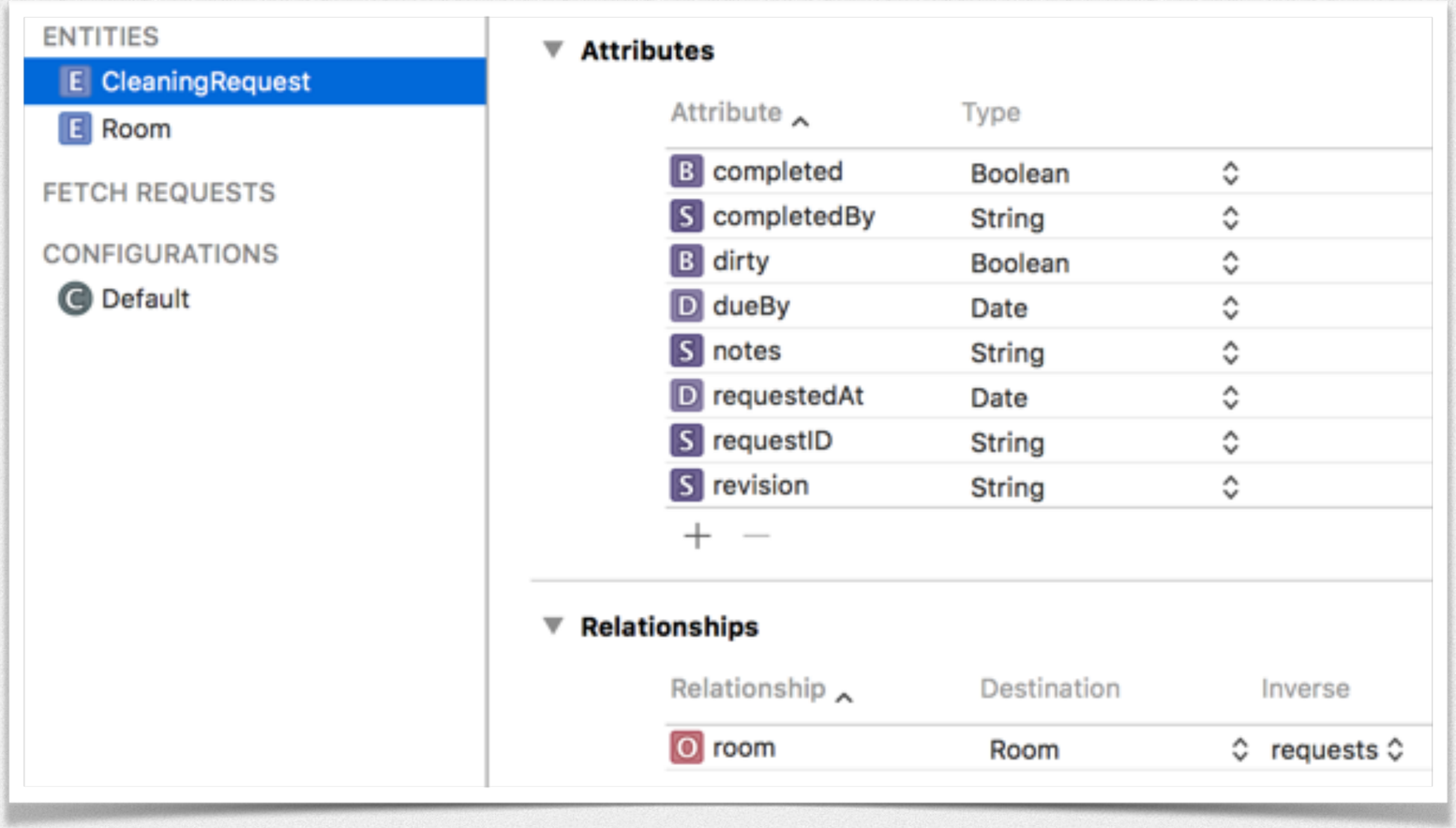
- Networking calls
- De/Serialization
- Translation of data
  - Dates
  - Data format/Localization
- Error conditions
- OHHTTPStubs for testing

Let’s Look at some Code

308: Core Data to Online Service Synchronization

Data Models

Core Data Model



Remote Data Changes


```
{
   "_id": "4da74a0abb4896376622c6eeed0018f4",
   "_rev": "1-108ce4f3e2f178655c67170e987406b8",
   "roomID": "b52353e4f1f3dc2cc38f1afd560005fd",
   "completed": false,
   "completedBy": null,
   "requestedAt": 1448206702.8197300434,
   "dueBy": 1448206702.8197300434,
   "notes": "Testing 123",
}
```

Remote Data Changes

```
{
   "_id": "4da74a0abb4896376622c6eeed0018f4",
   "_rev": "1-108ce4f3e2f178655c67170e987406b8",
   "roomID":
   "complete
   "complete
   "requeste
   "~~dueBy~~":
   "notes":
}
```

```
{
   "_id": "4da74a0abb4896376622c6eeed0018f4",
   "_rev": "1-108ce4f3e2f178655c67170e987406b8",
   "roomID": "b52353e4f1f3dc2cc38f1afd560005fd",
   "completed": false,
   "completedBy": null,
   "requestedAt": 1448206702.8197300434,
   "dueDate": 1448206702.8197300434,
   "notes": "Testing 123",
}
```

Remote Data Changes


```
{
   "_id": "4da74a0abb4896
   "_rev": "1-108ce4f3e2f
   "roomID": "b52353e4f1f
   "completed": false,
   "completedBy": null,
   "requestedA
   "dueBy": 14
   "notes": "T
}
```

```
{
   "_id": "4da74a0abb4896376622c6eeed0018f4",
   "_rev": "1-108ce4f3e2f178655c67170e987406b8",
   "roomID": "b52353e4f1f3dc2cc38f1afd560005fd",
```

```
{
   "_id": "4da74a0abb4896376622c6eeed0018f4",
   "_rev": "1-108ce4f3e2f178655c67170e987406b8",
   "roomID": "b52353e4f1f3dc2cc38f1afd560005fd",
   "completed": "partially",
   "completedBy": null,
   "requestedAt": 1448206702.8197300434,
   "dueDate": 1448206702.8197300434,
   "notes": "Testing 123",
}
```

NSManagedObjects

```
86    }
87
88    func testCoreDataMO() {
89      let room = Room()
90      room.roomNumber = "100"
91    }
92
```

caught "NSInvalidArgumentException", "-[CleanRooms.Room setRoomNumber:]: unrecognized selector

Managed Objects are not lightweight

Separate the Models

Managed Objects	Remote Data
Room	RemoteRoom
CleaningRequest	RemoteCleaningRequest
:NSManagedObject	No Root Class

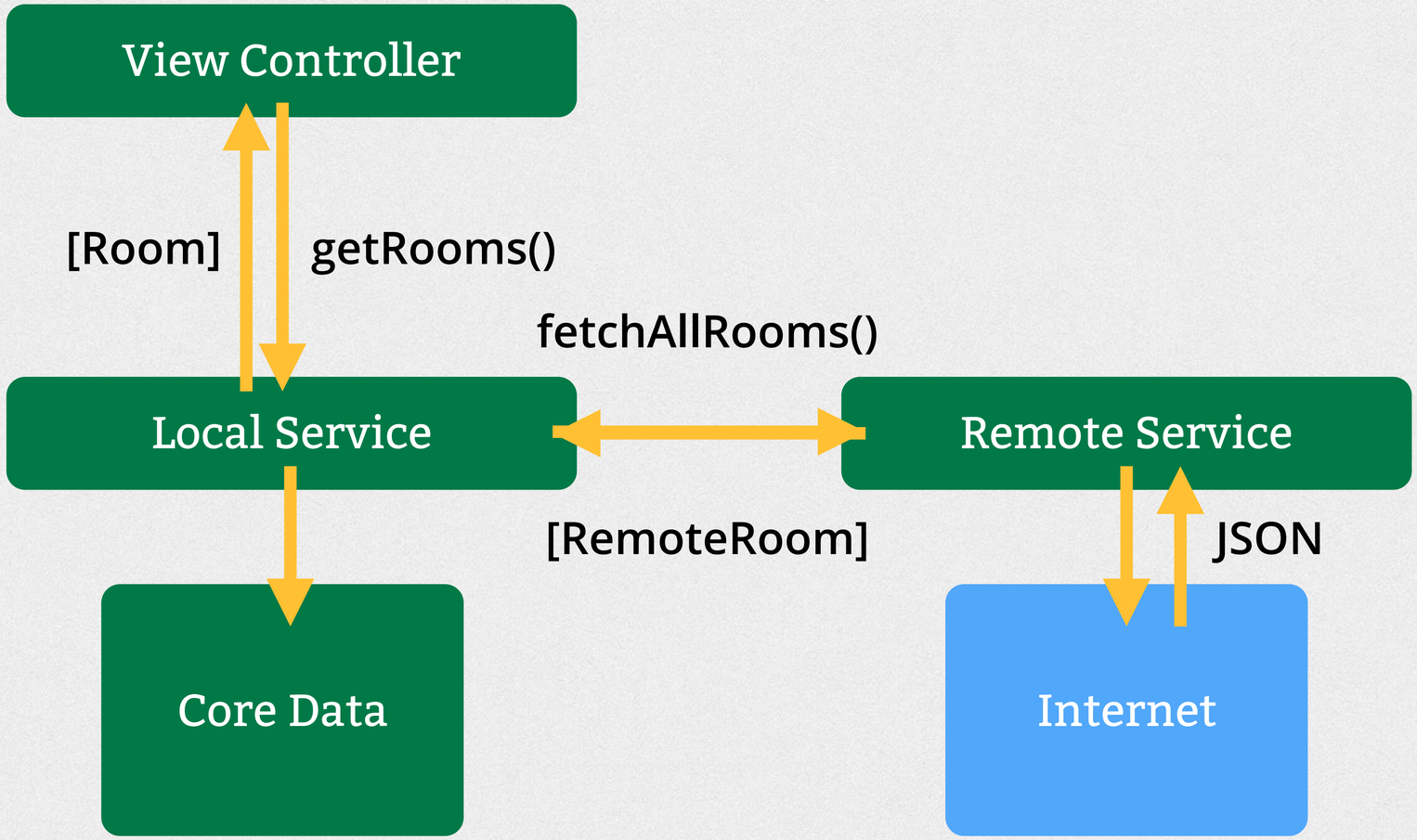
```swift
public extension CleaningRequest {

  @NSManaged var completed: NSNumber?
  @NSManaged var completedBy: String?
  @NSManaged var dirty: NSNumber?
  @NSManaged var dueBy: NSDate?
  @NSManaged var requestedAt: NSDate?
  @NSManaged var requestID: String?
  @NSManaged var revision: String?
  @NSManaged var notes: String?
  @NSManaged var room: Room?

}
```

Core Data

```swift
public struct RemoteCleaningRequest {
  public var requestID: String
  public var revision: String
  public var roomID: String
  public var requestedAt: NSDate?
  public var dueBy: NSDate?
  public var completed: Bool?
  public var completedBy: String?
  public var notes: String?
```

Remote

Separation of Data Models



JSON & Remote Models

```
public struct RemoteCleaningRequest {
  // properties

  public init(requestID: String, revision: String,
    roomID: String, requestedAt: NSDate?,
    dueBy: NSDate?, completed: Bool?,
    completedBy: String?, notes: String?) {
   // ...
  }

  public init(jsonData: [String: AnyObject]) {
   // ...
  }

  public func jsonDict() -> [String: AnyObject] {
   // ...
  }
}
```

Show me the Models

308: Core Data to Online Service Synchronization

Synchronization

raywenderlich.com

Basic Steps of Synchronization

Now delete everything
local not remote

Here's what I changed since X

Here's what's changed

The "Cloud"

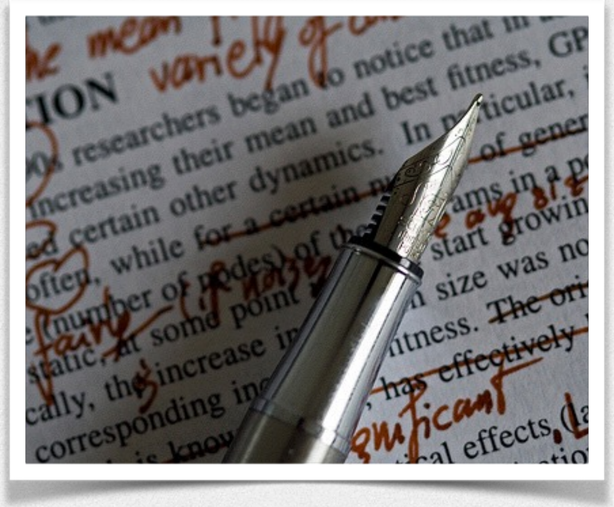
Basic Synchronization

- Upload any local changes (dirty data) if any
- Fetch all data since last successful sync
- Delete all local items not in remote

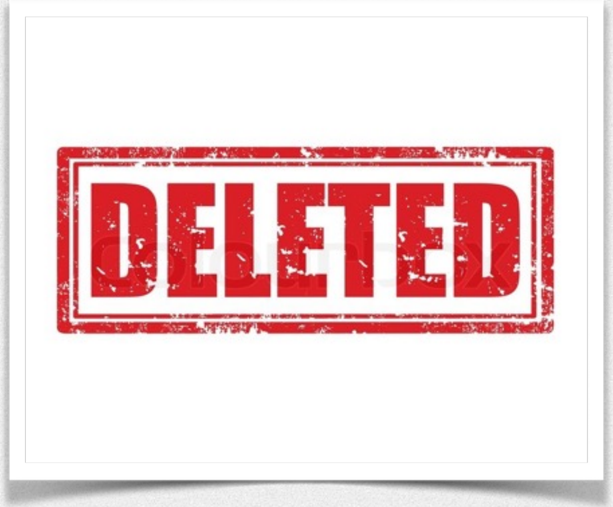
- Conflicts
  - local changes overwrite remote
  - remote data overwrites local

Sync Helpers

isDirty: Bool



revision: String

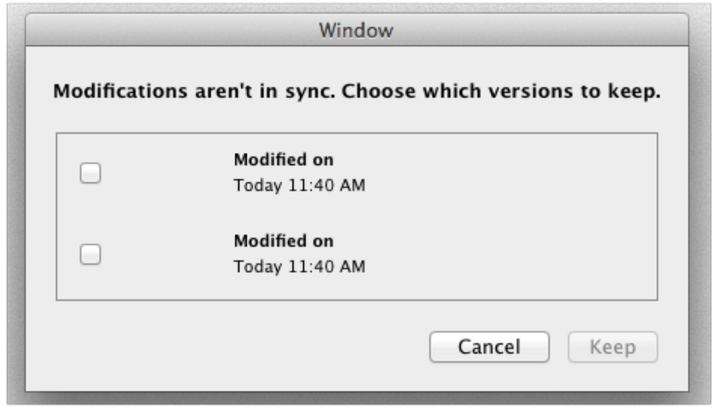


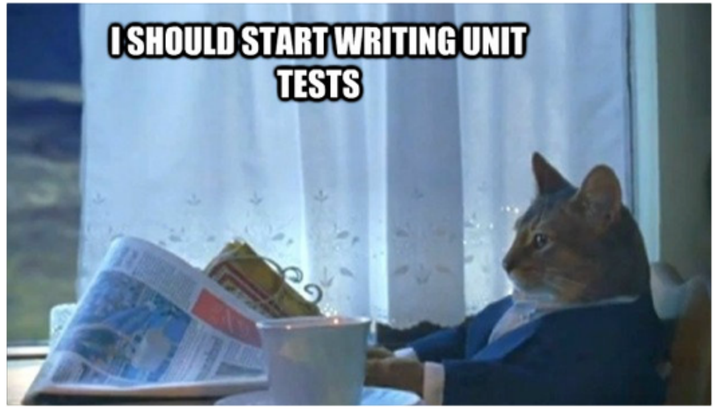
isDeleted: Bool

Things can go wrong

- Conflicts
- Missing related entities (local or remote)
- Connection dies

Establish Rules & Test





Local Services

RoomService

managedObjectContext

getAllRooms() -> [Room]
getRoomByID(_: String) -> Room?

CleaningRequestService

managedObjectContext

getAllCleaningRequests() -> [CleaningRequest]
getAllDirtyCleaningRequests() -> [CleaningRequest]
getCleaningRequestByID(_: String) -> CleaningRequest?
markCleaningRequestAsCompleted(_: String) -> CleaningRequest?

Remote Services

RoomServiceRemote

urlSession

fetchAllRooms(_: ([RemoteRoom]) -> Void)

CleaningRequestServiceRemote

urlSession

fetchAllRequests(_: ([RemoteCleaningRequest]) -> Void)
updateRemoteCleaningRequests(_: ([RemoteCleaningRequest]) -> Void)
updateRemoteCleaningRequest(_: RemoteCleaningRequest, completion: (revision: String) -> Void)

Synchronization Service

SynchronizationService

roomService
roomServiceRemote
cleaningRequestService
cleaningRequestServiceRemote

CleaningRequestService

RoomService

CleaningRequestServiceRemote

RoomServiceRemote

Demo
Lets Synchronize!

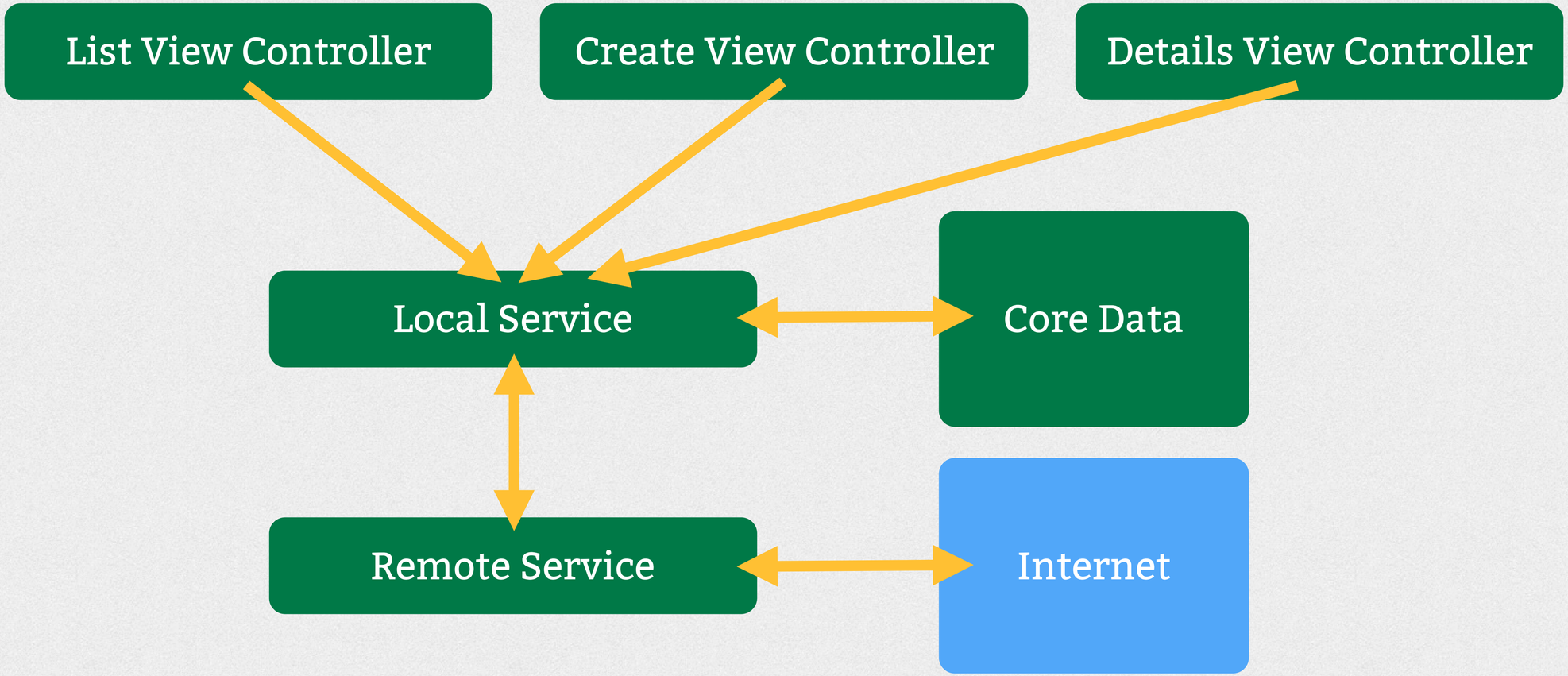
308: Core Data to Online Service Synchronization

Conclusion

What You Learned

- **Part 1**: Service Architecture
- **Part 2**: Data Models
- **Demo**: Synchronization
- **Lab**: Expanding the Model

Solution


List View Controller
Create View Controller
Details View Controller
Local Service
Core Data
Remote Service
Internet

Where To Go From Here?

- Service Layers - http://martinfowler.com/eaaCatalog/serviceLayer.html
- CouchDB - http://couchdb.apache.org
- WordPress for iOS - https://github.com/wordpress-mobile/WordPress-iOS