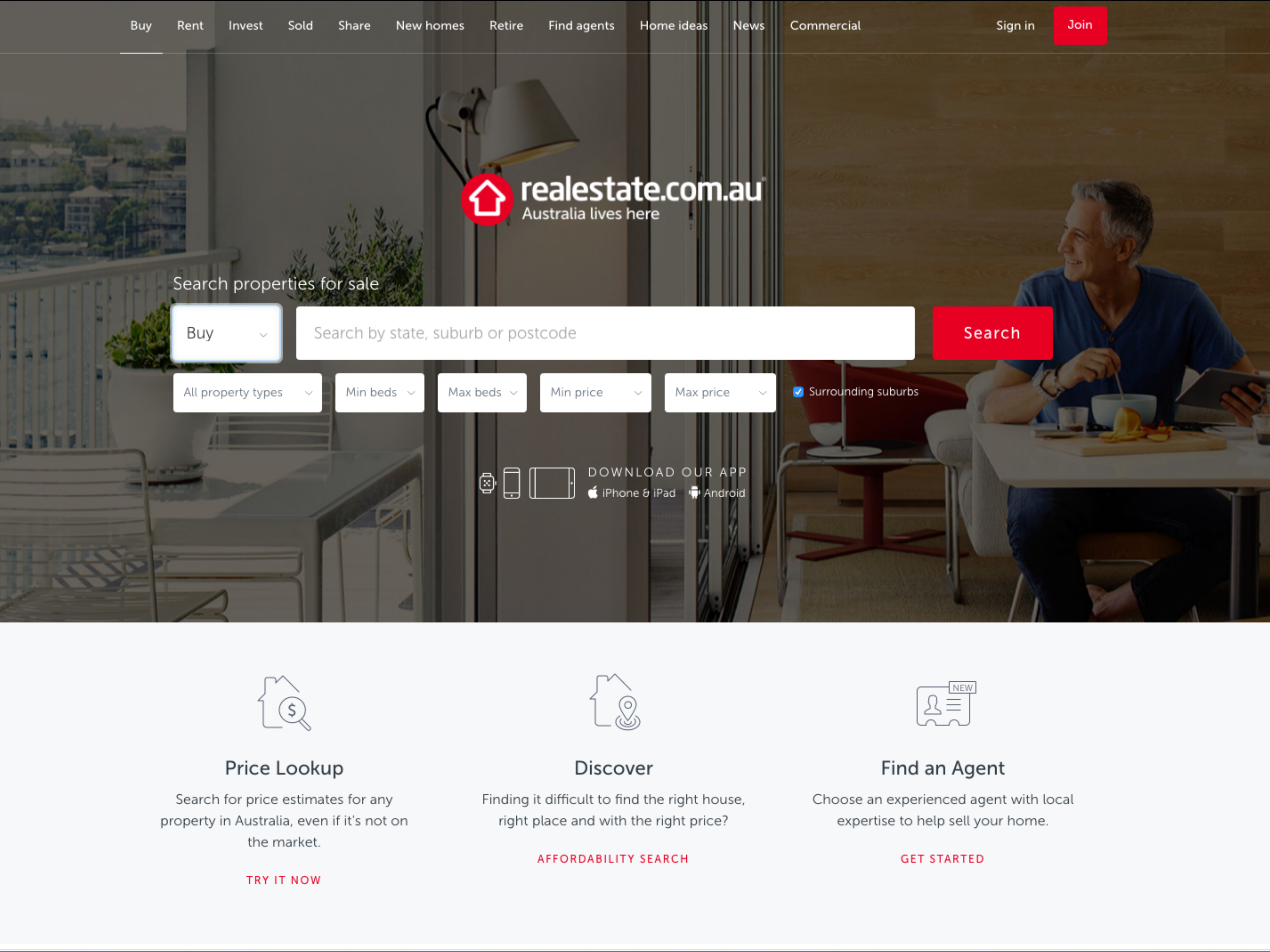
遗留系统的架构重构

@凉粉小刀



realestate.com.au

Australia lives here

Search properties for sale

DOWNLOAD OUR APP

Price Lookup

Search for price estimates for any property in Australia, even if it's not on the market.

TRY IT NOW

Discover

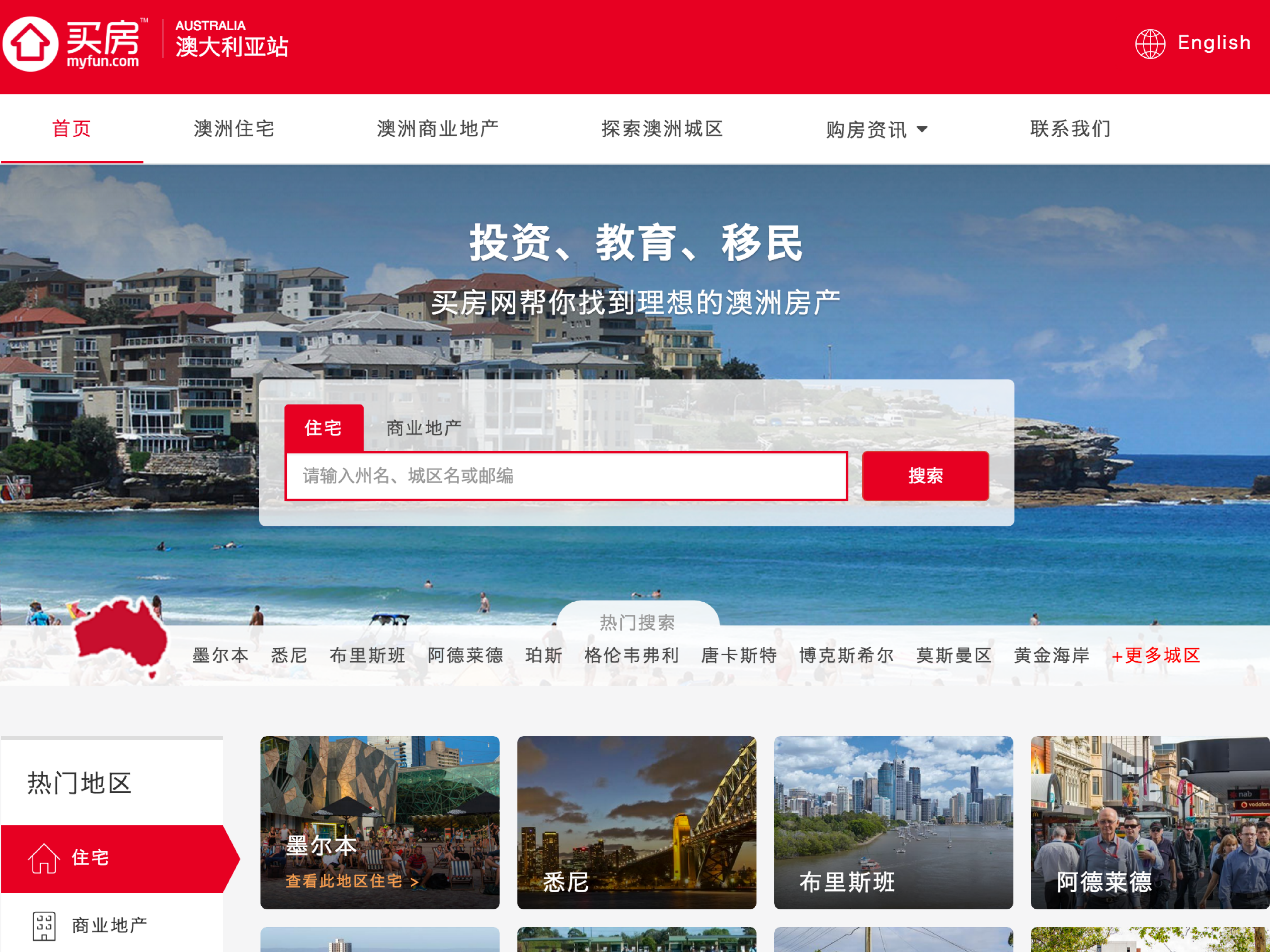
Finding it difficult to find the right house, right place and with the right price?

AFFORDABILITY SEARCH

Find an Agent

Choose an experienced agent with local expertise to help sell your home.

GET STARTED



买房™ myfun.com

AUSTRALIA
澳大利亚站

English

首页　澳洲住宅　澳洲商业地产　探索澳洲城区　购房资讯 ▾　联系我们

投资、教育、移民

买房网帮你找到理想的澳洲房产

住宅　商业地产

请输入州名、城区名或邮编

搜索

热门搜索

墨尔本　悉尼　布里斯班　阿德莱德　珀斯　格伦韦弗利　唐卡斯特　博克斯希尔　莫斯曼区　黄金海岸　+更多城区

热门地区

住宅

商业地产

墨尔本

查看此地区住宅 >

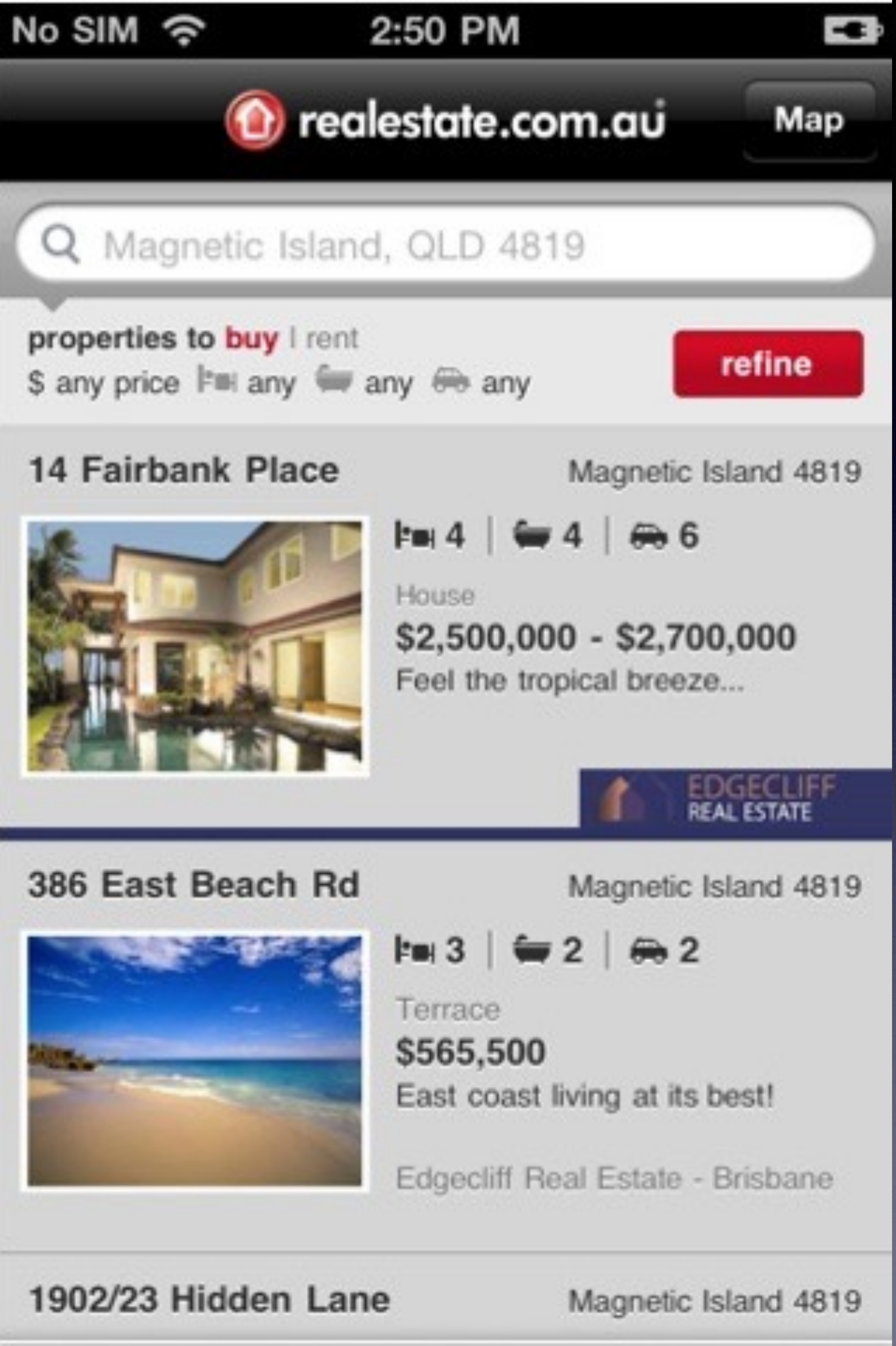
悉尼

布里斯班

阿德莱德

7 Dev in Melbourne

4 Dev in ThoughtWorks, Xi’an



properties to buy | rent

$ any price 🛏 any 🛁 any 🚗 any

14 Fairbank Place Magnetic Island 4819

🛏 4 | 🛁 4 | 🚗 6

House

$2,500,000 - $2,700,000

Feel the tropical breeze...

EDGECLIFF REAL ESTATE

386 East Beach Rd Magnetic Island 4819

🛏 3 | 🛁 2 | 🚗 2

Terrace

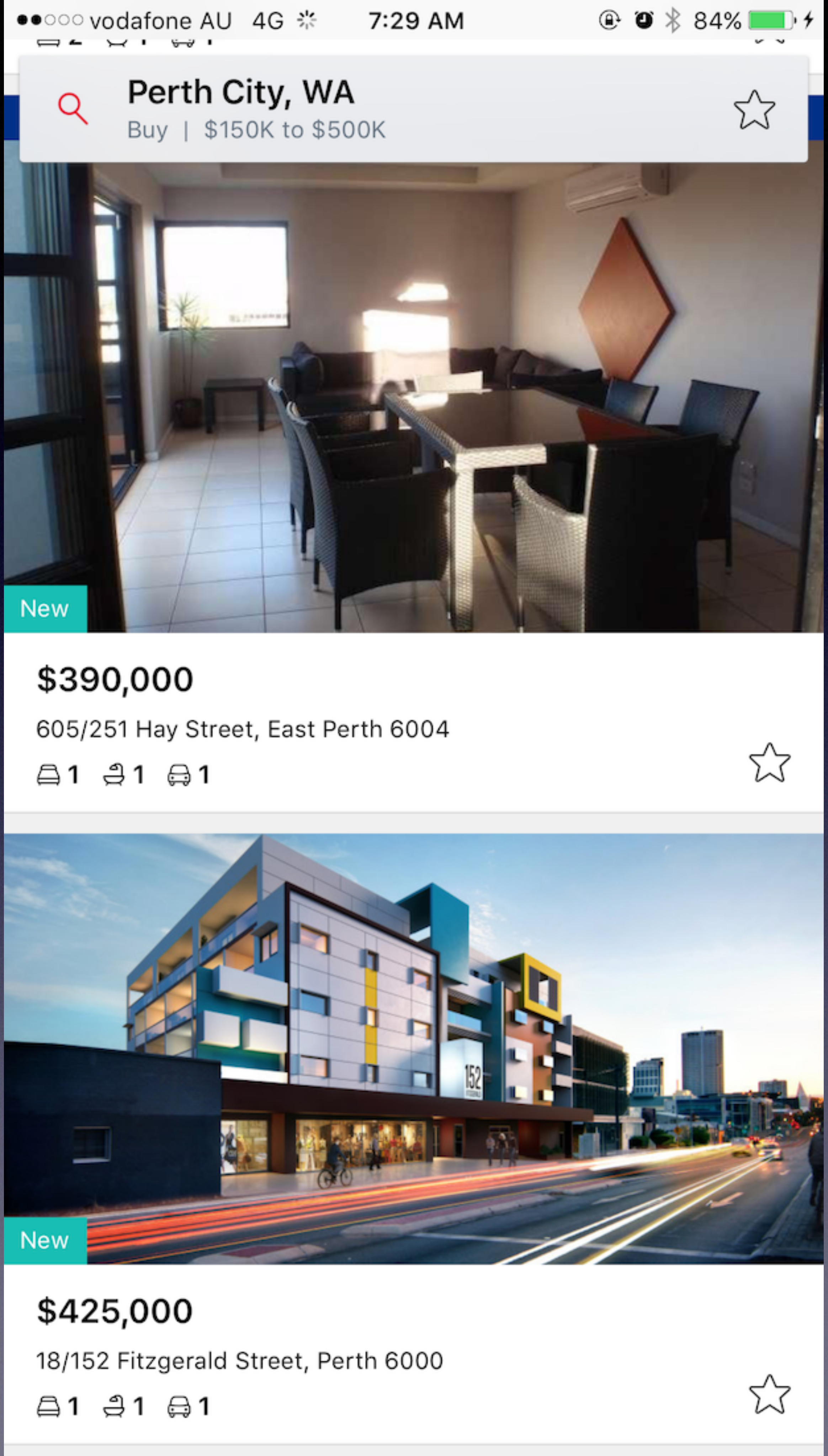
$565,500

East coast living at its best!

Edgecliff Real Estate - Brisbane

1902/23 Hidden Lane Magnetic Island 4819

2010.12.16



●●○○○ vodafone AU 4G 7:29 AM 84%

Perth City, WA

Buy | $150K to $500K ☆

New

$390,000

605/251 Hay Street, East Perth 6004

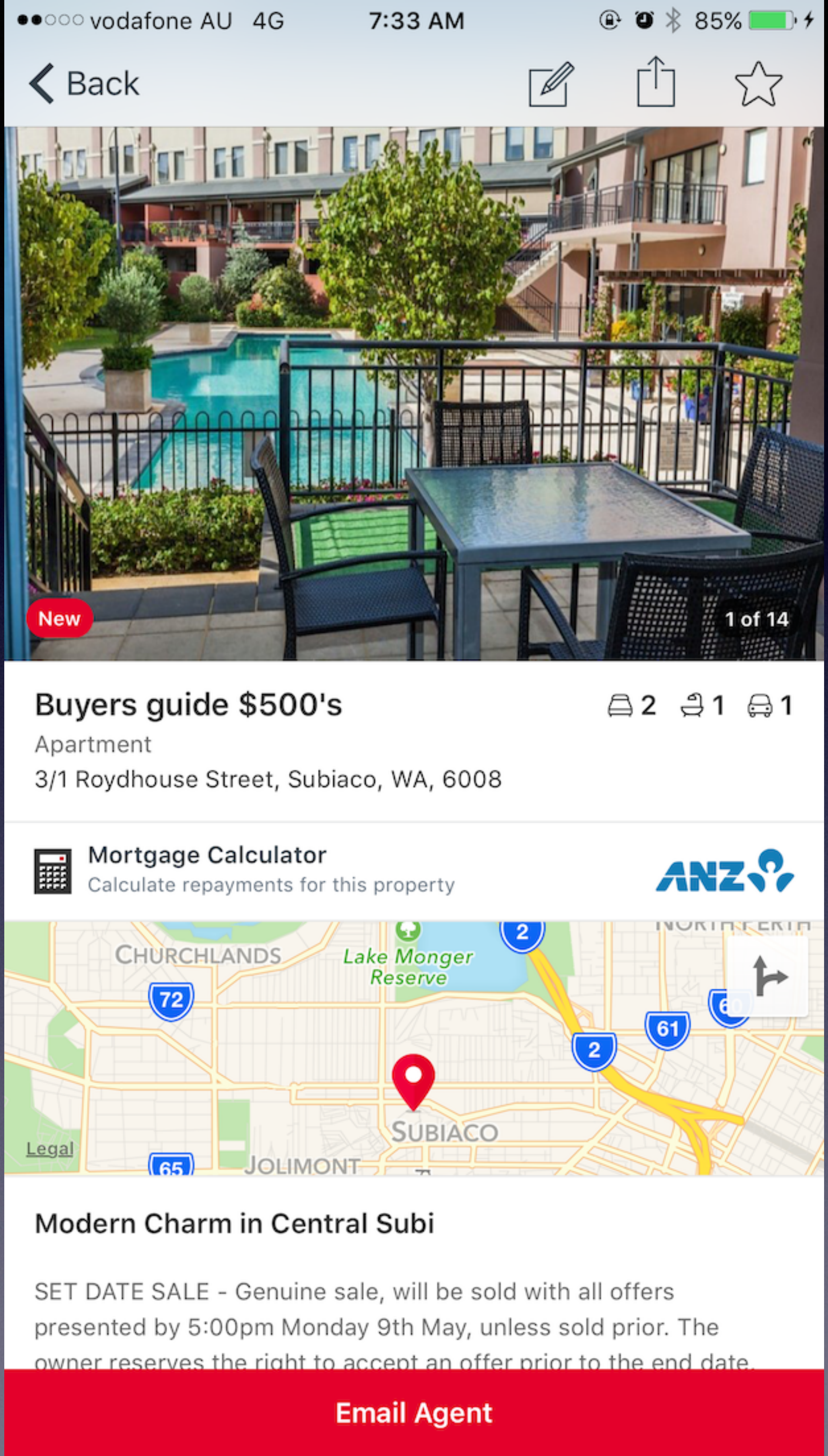
1 1 1 ☆

New

$425,000

18/152 Fitzgerald Street, Perth 6000

1 1 1 ☆



●●○○○ vodafone AU 4G 7:33 AM 85%

< Back ☆

New

1 of 14

Buyers guide $500's 2 1 1

Apartment

3/1 Roydhouse Street, Subiaco, WA, 6008

Mortgage Calculator

Calculate repayments for this property

ANZ

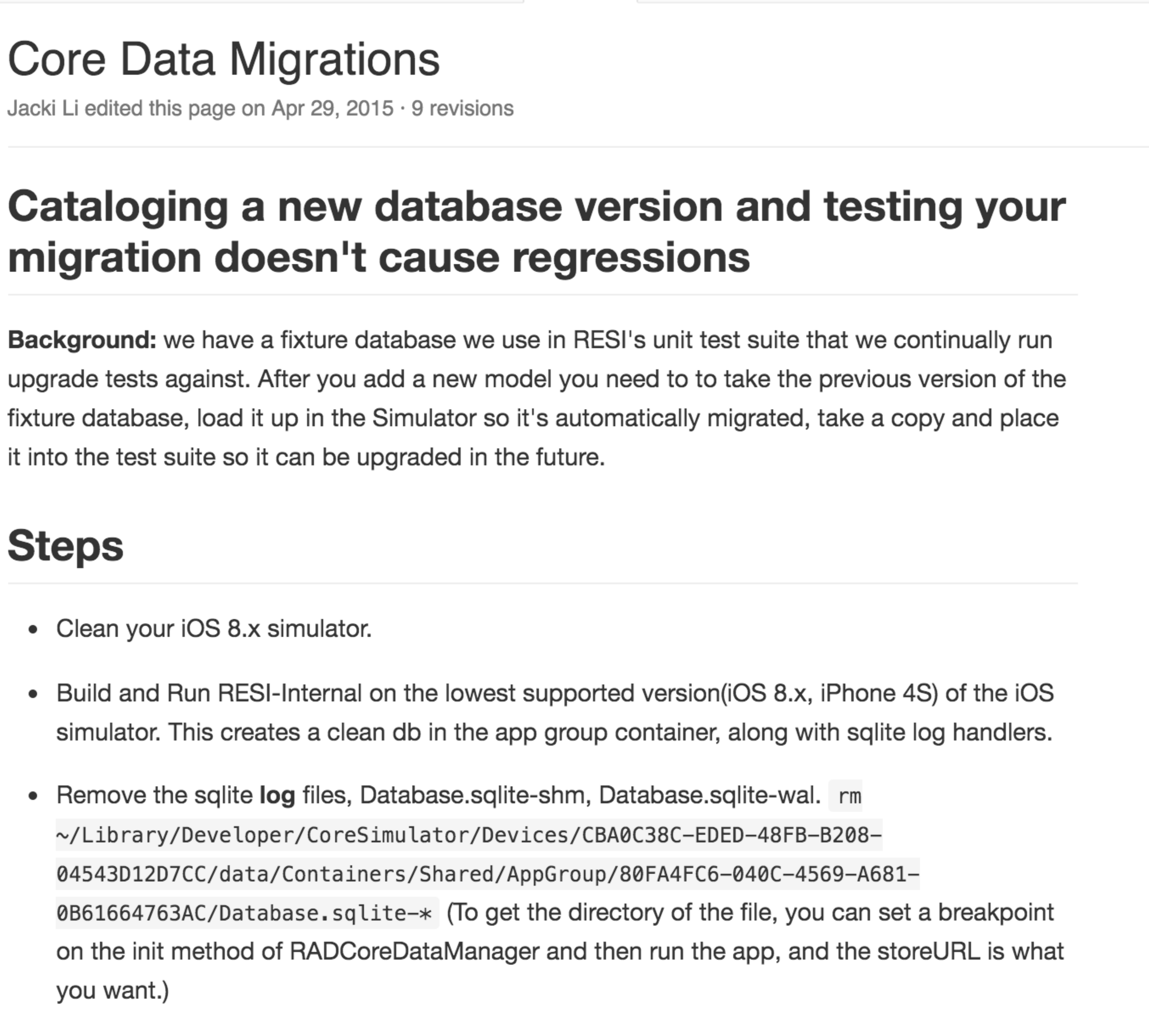
Modern Charm in Central Subi

SET DATE SALE - Genuine sale, will be sold with all offers presented by 5:00pm Monday 9th May, unless sold prior. The owner reserves the right to accept an offer prior to the end date.

Email Agent

真实世界的故事

一切从CoreData开始



Core Data Migrations

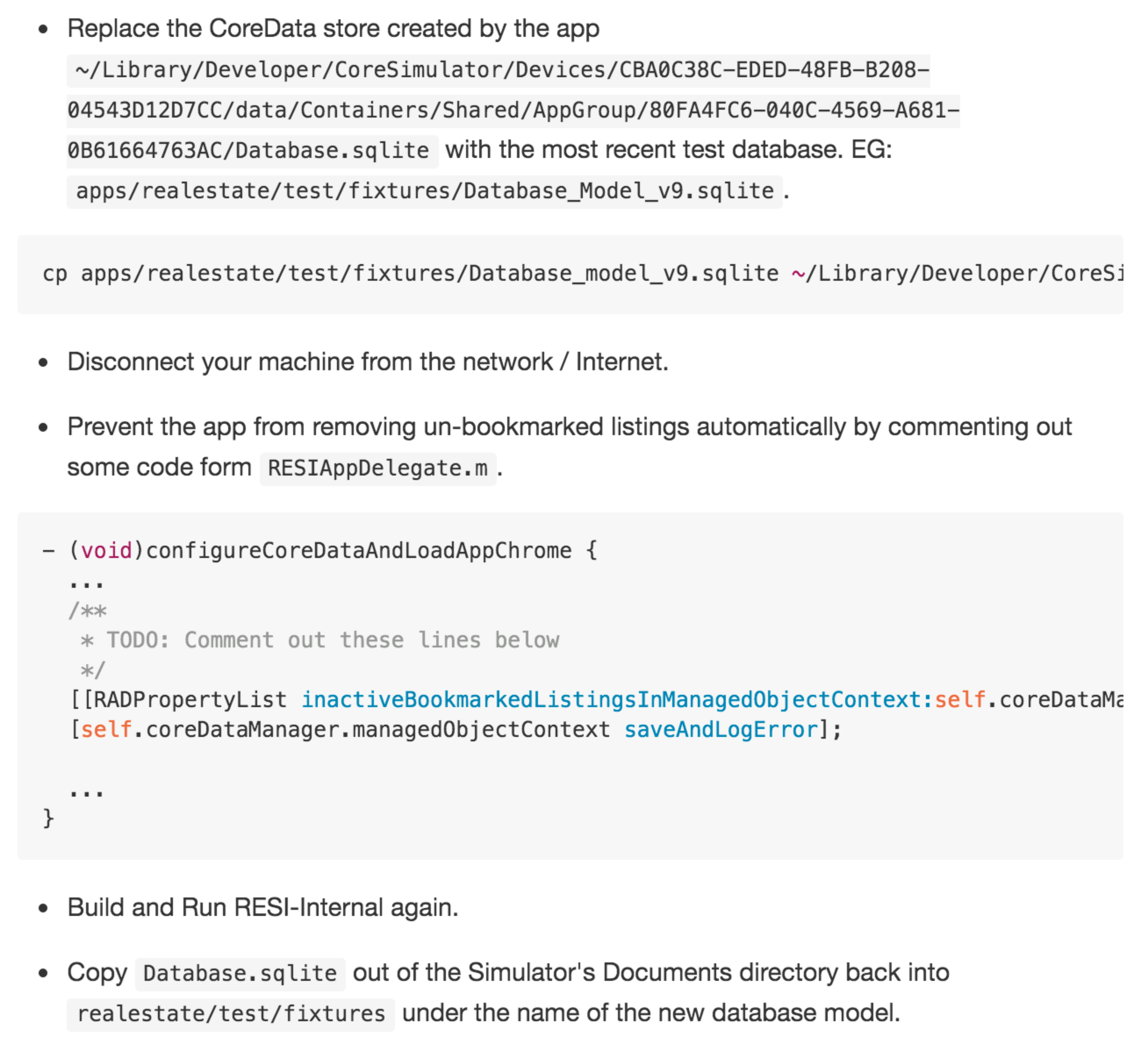
Jacki Li edited this page on Apr 29, 2015 · 9 revisions

Cataloging a new database version and testing your migration doesn't cause regressions

Background: we have a fixture database we use in RESI's unit test suite that we continually run upgrade tests against. After you add a new model you need to to take the previous version of the fixture database, load it up in the Simulator so it's automatically migrated, take a copy and place it into the test suite so it can be upgraded in the future.

Steps

- Clean your iOS 8.x simulator.
- Build and Run RESI-Internal on the lowest supported version(iOS 8.x, iPhone 4S) of the iOS simulator. This creates a clean db in the app group container, along with sqlite log handlers.
- Remove the sqlite **log** files, Database.sqlite-shm, Database.sqlite-wal. `rm ~/Library/Developer/CoreSimulator/Devices/CBA0C38C-EDED-48FB-B208-04543D12D7CC/data/Containers/Shared/AppGroup/80FA4FC6-040C-4569-A681-0B61664763AC/Database.sqlite-*` (To get the directory of the file, you can set a breakpoint on the init method of RADCoreDataManager and then run the app, and the storeURL is what you want.)



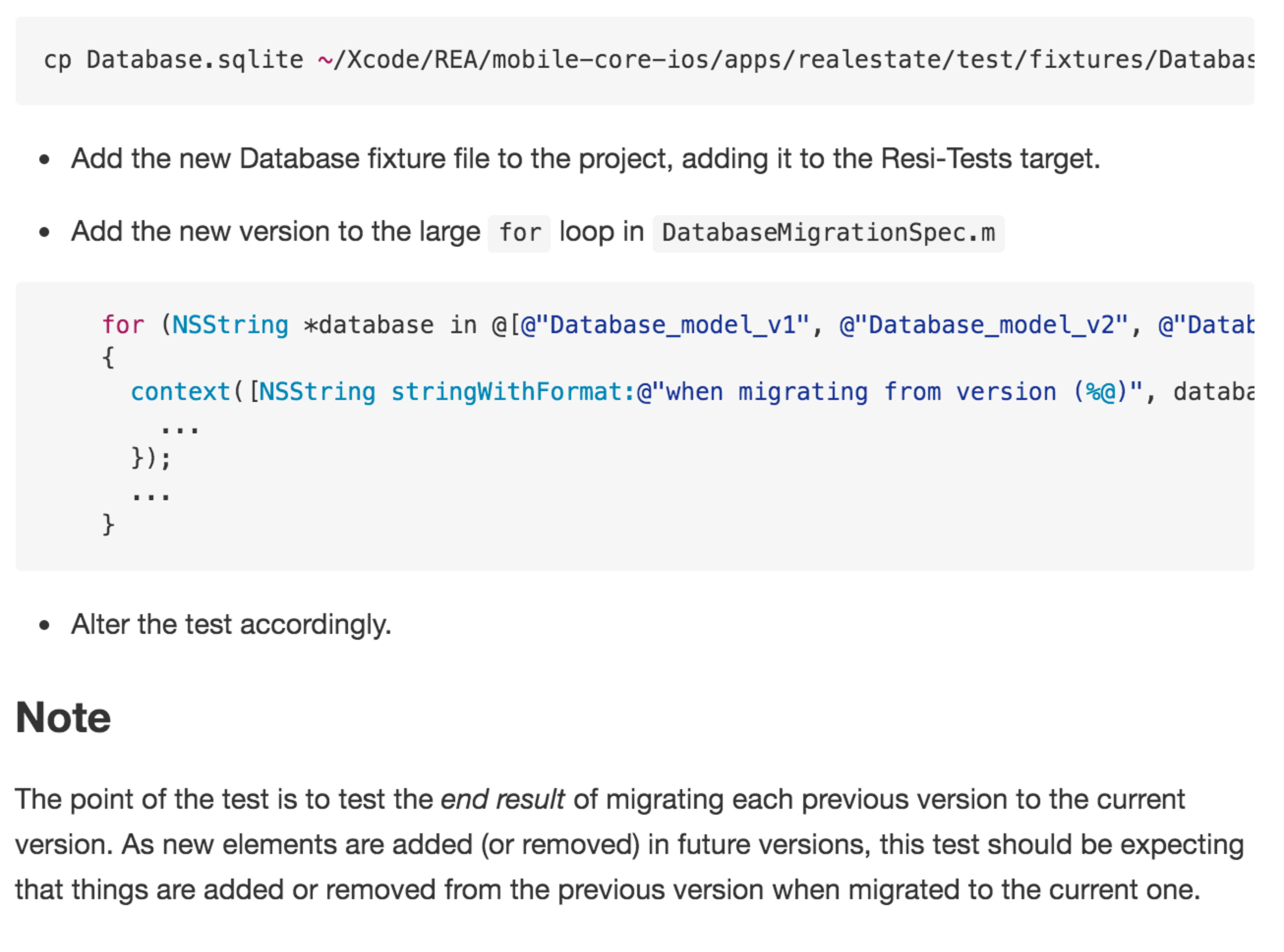
- Replace the CoreData store created by the app `~/Library/Developer/CoreSimulator/Devices/CBA0C38C-EDED-48FB-B208-04543D12D7CC/data/Containers/Shared/AppGroup/80FA4FC6-040C-4569-A681-0B61664763AC/Database.sqlite` with the most recent test database. EG: `apps/realestate/test/fixtures/Database_Model_v9.sqlite`.

```
cp apps/realestate/test/fixtures/Database_model_v9.sqlite ~/Library/Developer/CoreSi
```

- Disconnect your machine from the network / Internet.
- Prevent the app from removing un-bookmarked listings automatically by commenting out some code form `RESIAppDelegate.m`.

```
- (void)configureCoreDataAndLoadAppChrome {
  ...
  /**
   * TODO: Comment out these lines below
   */
  [[RADPropertyList inactiveBookmarkedListingsInManagedObjectContext:self.coreDataMa
  [self.coreDataManager.managedObjectContext saveAndLogError];

  ...
}
```

- Build and Run RESI-Internal again.
- Copy `Database.sqlite` out of the Simulator's Documents directory back into `realestate/test/fixtures` under the name of the new database model.



```
cp Database.sqlite ~/Xcode/REA/mobile-core-ios/apps/realestate/test/fixtures/Databas
```

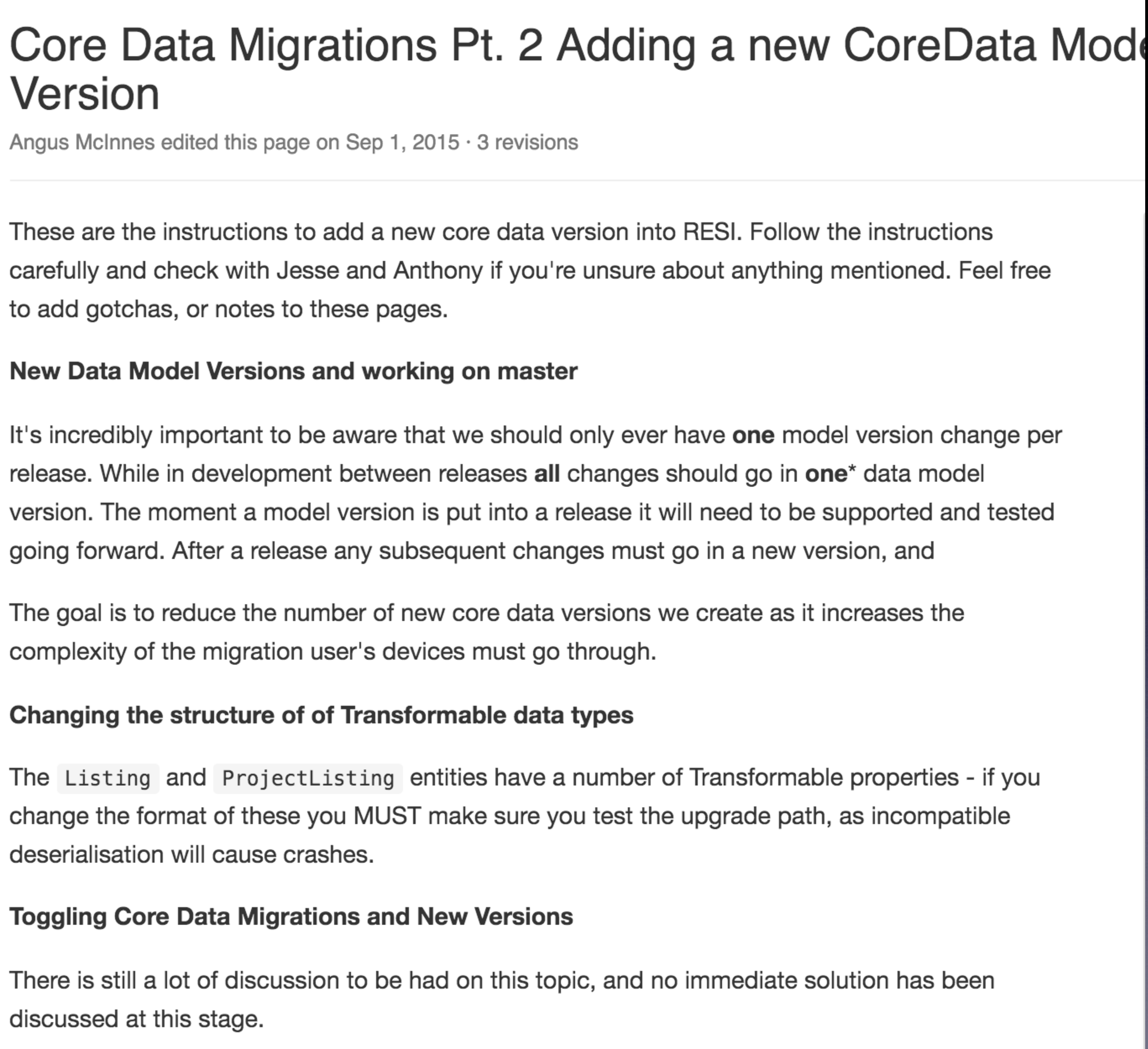
- Add the new Database fixture file to the project, adding it to the Resi-Tests target.
- Add the new version to the large `for` loop in `DatabaseMigrationSpec.m`

```
    for (NSString *database in @[@"Database_model_v1", @"Database_model_v2", @"Datab
    {
      context([NSString stringWithFormat:@"when migrating from version (%@)", databa
        ...
      });
      ...
    }
```

- Alter the test accordingly.

Note

The point of the test is to test the *end result* of migrating each previous version to the current version. As new elements are added (or removed) in future versions, this test should be expecting that things are added or removed from the previous version when migrated to the current one.



Core Data Migrations Pt. 2 Adding a new CoreData Mode Version

Angus McInnes edited this page on Sep 1, 2015 · 3 revisions

These are the instructions to add a new core data version into RESI. Follow the instructions carefully and check with Jesse and Anthony if you're unsure about anything mentioned. Feel free to add gotchas, or notes to these pages.

New Data Model Versions and working on master

It's incredibly important to be aware that we should only ever have **one** model version change per release. While in development between releases **all** changes should go in **one*** data model version. The moment a model version is put into a release it will need to be supported and tested going forward. After a release any subsequent changes must go in a new version, and

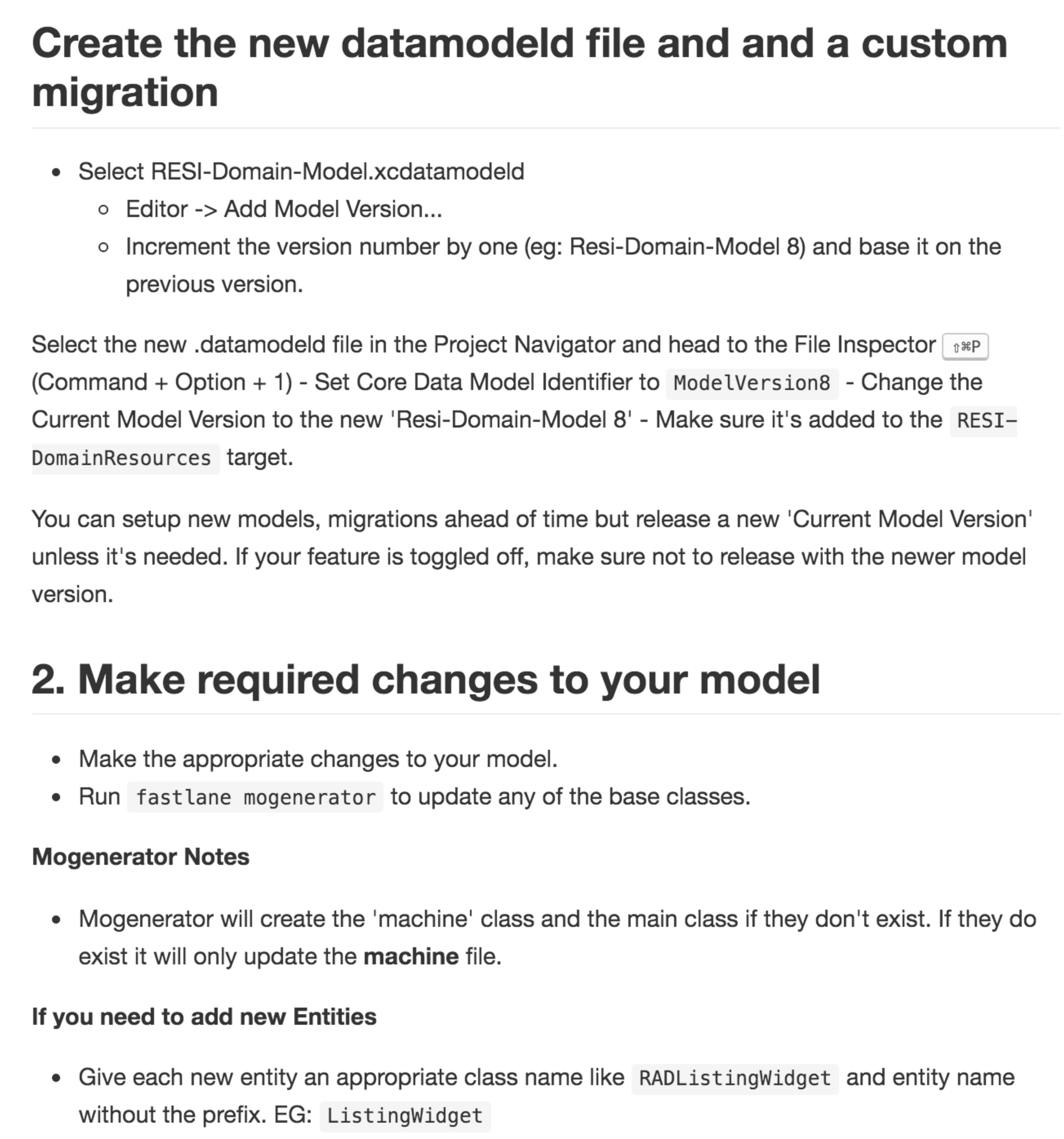
The goal is to reduce the number of new core data versions we create as it increases the complexity of the migration user's devices must go through.

Changing the structure of of Transformable data types

The `Listing` and `ProjectListing` entities have a number of Transformable properties - if you change the format of these you MUST make sure you test the upgrade path, as incompatible deserialisation will cause crashes.

Toggling Core Data Migrations and New Versions

There is still a lot of discussion to be had on this topic, and no immediate solution has been discussed at this stage.



Create the new datamodeld file and and a custom migration

- Select RESI-Domain-Model.xcdatamodeld
    - Editor -> Add Model Version...
    - Increment the version number by one (eg: Resi-Domain-Model 8) and base it on the previous version.

Select the new .datamodeld file in the Project Navigator and head to the File Inspector `⇧⌘P` (Command + Option + 1) - Set Core Data Model Identifier to `ModelVersion8` - Change the Current Model Version to the new 'Resi-Domain-Model 8' - Make sure it's added to the `RESI-DomainResources` target.

You can setup new models, migrations ahead of time but release a new 'Current Model Version' unless it's needed. If your feature is toggled off, make sure not to release with the newer model version.

2. Make required changes to your model

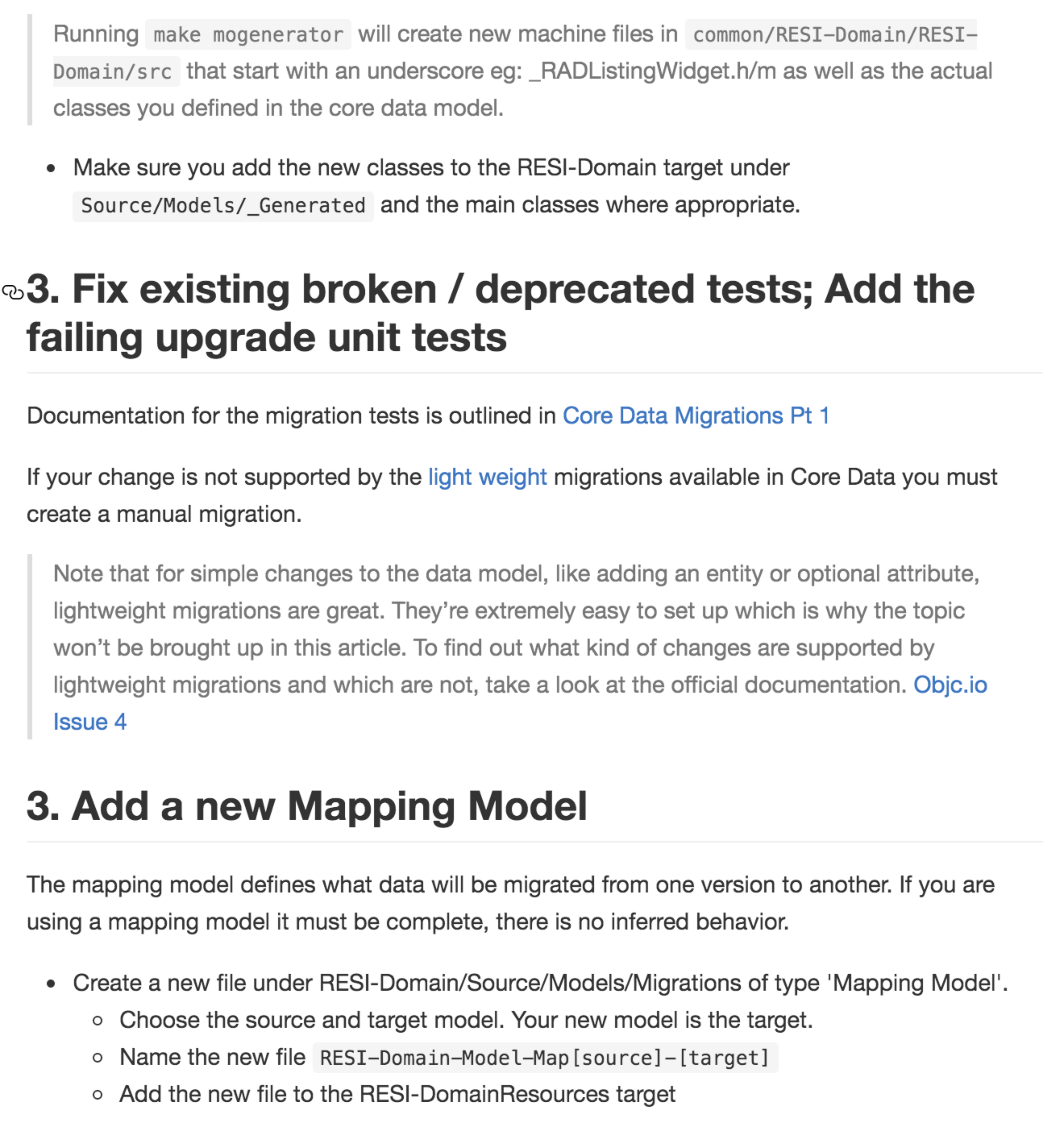
- Make the appropriate changes to your model.
- Run `fastlane mogenerator` to update any of the base classes.

Mogenerator Notes

- Mogenerator will create the 'machine' class and the main class if they don't exist. If they do exist it will only update the **machine** file.

If you need to add new Entities

- Give each new entity an appropriate class name like `RADListingWidget` and entity name without the prefix. EG: `ListingWidget`



> Running `make mogenerator` will create new machine files in `common/RESI-Domain/RESI-Domain/src` that start with an underscore eg: _RADListingWidget.h/m as well as the actual classes you defined in the core data model.

- Make sure you add the new classes to the RESI-Domain target under `Source/Models/_Generated` and the main classes where appropriate.

3. Fix existing broken / deprecated tests; Add the failing upgrade unit tests

Documentation for the migration tests is outlined in Core Data Migrations Pt 1

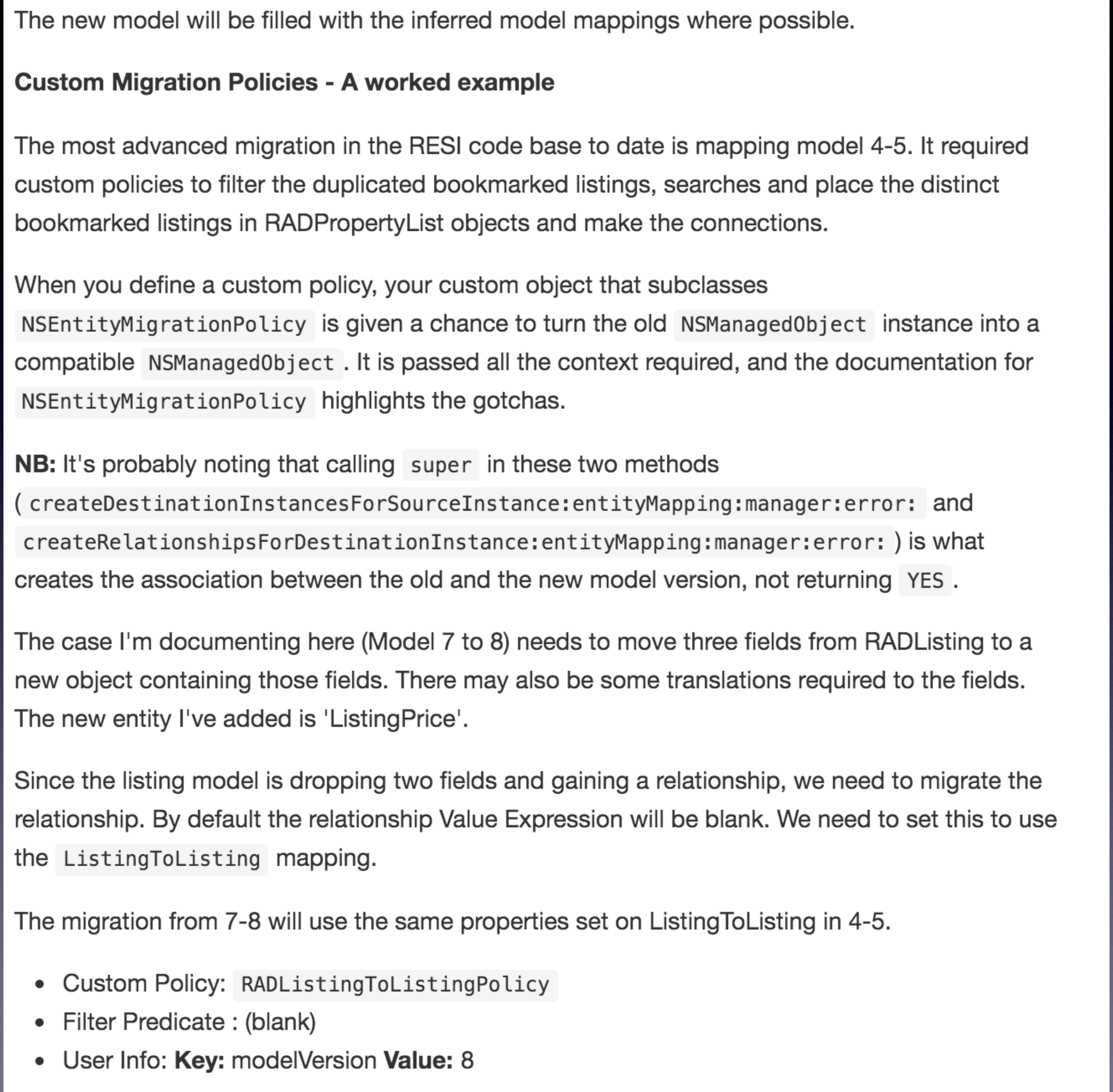
If your change is not supported by the light weight migrations available in Core Data you must create a manual migration.

> Note that for simple changes to the data model, like adding an entity or optional attribute, lightweight migrations are great. They're extremely easy to set up which is why the topic won't be brought up in this article. To find out what kind of changes are supported by lightweight migrations and which are not, take a look at the official documentation. Objc.io Issue 4

3. Add a new Mapping Model

The mapping model defines what data will be migrated from one version to another. If you are using a mapping model it must be complete, there is no inferred behavior.

- Create a new file under RESI-Domain/Source/Models/Migrations of type 'Mapping Model'.
  - Choose the source and target model. Your new model is the target.
  - Name the new file `RESI-Domain-Model-Map[source]-[target]`
  - Add the new file to the RESI-DomainResources target



The new model will be filled with the inferred model mappings where possible.

Custom Migration Policies - A worked example

The most advanced migration in the RESI code base to date is mapping model 4-5. It required custom policies to filter the duplicated bookmarked listings, searches and place the distinct bookmarked listings in RADPropertyList objects and make the connections.

When you define a custom policy, your custom object that subclasses `NSEntityMigrationPolicy` is given a chance to turn the old `NSManagedObject` instance into a compatible `NSManagedObject`. It is passed all the context required, and the documentation for `NSEntityMigrationPolicy` highlights the gotchas.

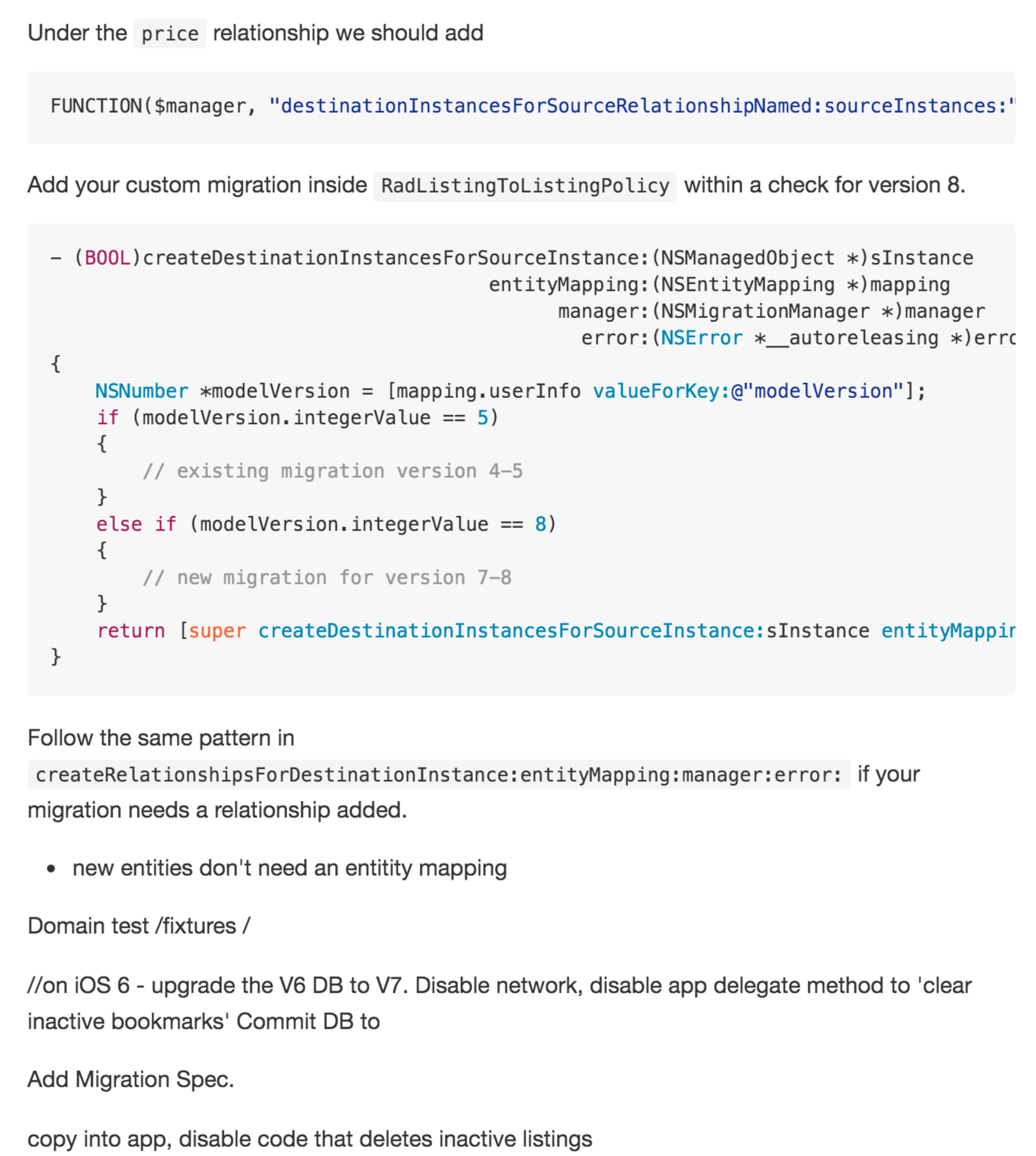
NB: It's probably noting that calling `super` in these two methods (`createDestinationInstancesForSourceInstance:entityMapping:manager:error:` and `createRelationshipsForDestinationInstance:entityMapping:manager:error:`) is what creates the association between the old and the new model version, not returning `YES`.

The case I'm documenting here (Model 7 to 8) needs to move three fields from RADListing to a new object containing those fields. There may also be some translations required to the fields. The new entity I've added is 'ListingPrice'.

Since the listing model is dropping two fields and gaining a relationship, we need to migrate the relationship. By default the relationship Value Expression will be blank. We need to set this to use the `ListingToListing` mapping.

The migration from 7-8 will use the same properties set on ListingToListing in 4-5.

- Custom Policy: `RADListingToListingPolicy`
- Filter Predicate : (blank)
- User Info: **Key:** modelVersion **Value:** 8



Under the `price` relationship we should add

```
FUNCTION($manager, "destinationInstancesForSourceRelationshipNamed:sourceInstances:'
```

Add your custom migration inside `RadListingToListingPolicy` within a check for version 8.

```
- (BOOL)createDestinationInstancesForSourceInstance:(NSManagedObject *)sInstance
                                      entityMapping:(NSEntityMapping *)mapping
                                            manager:(NSMigrationManager *)manager
                                              error:(NSError *__autoreleasing *)errc
{
    NSNumber *modelVersion = [mapping.userInfo valueForKey:@"modelVersion"];
    if (modelVersion.integerValue == 5)
    {
        // existing migration version 4-5
    }
    else if (modelVersion.integerValue == 8)
    {
        // new migration for version 7-8
    }
    return [super createDestinationInstancesForSourceInstance:sInstance entityMappir
}
```

Follow the same pattern in `createRelationshipsForDestinationInstance:entityMapping:manager:error:` if your migration needs a relationship added.

- new entities don't need an entitity mapping

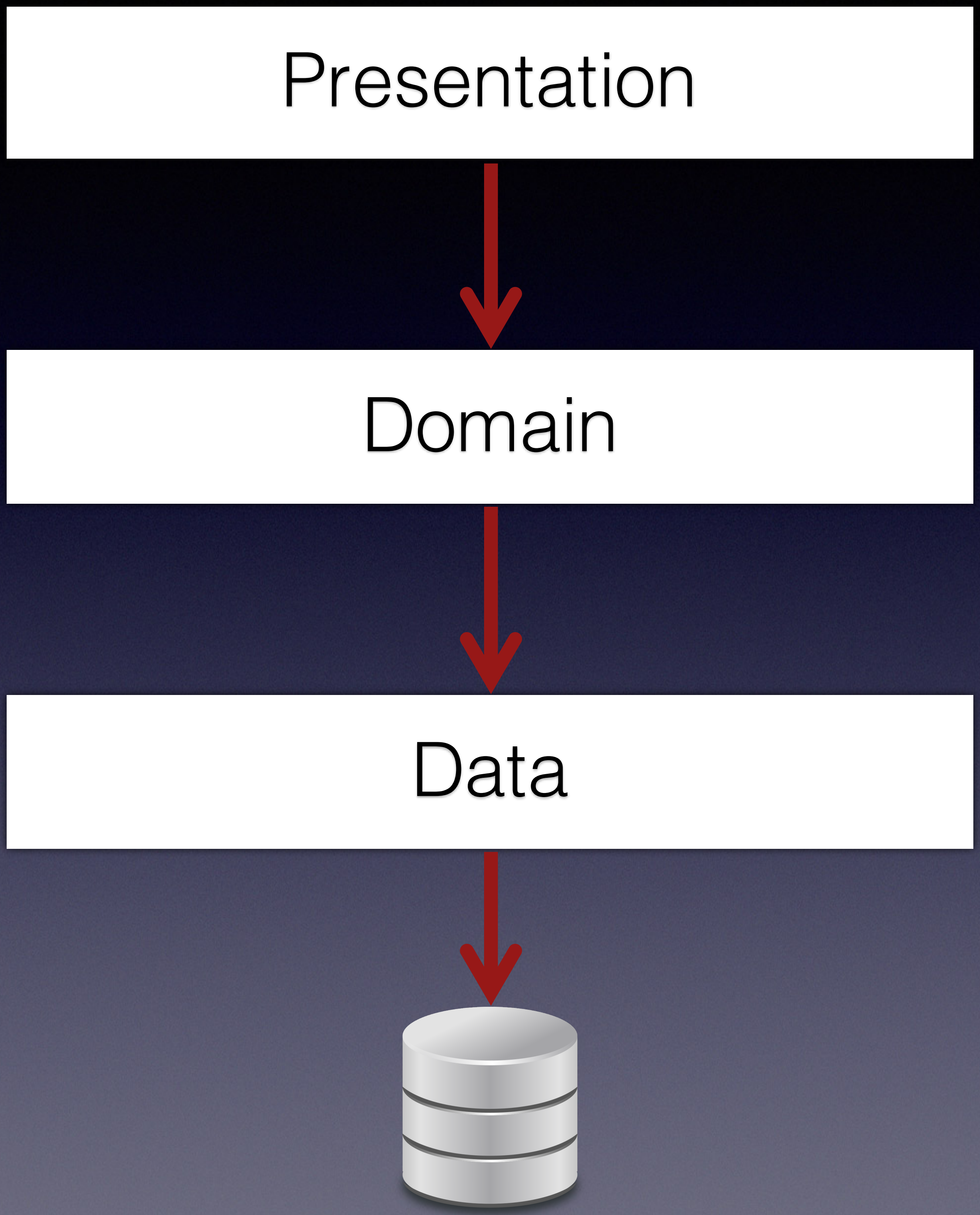
Domain test /fixtures /

//on iOS 6 - upgrade the V6 DB to V7. Disable network, disable app delegate method to 'clear inactive bookmarks' Commit DB to

Add Migration Spec.

copy into app, disable code that deletes inactive listings

正常的三层结构是这样的

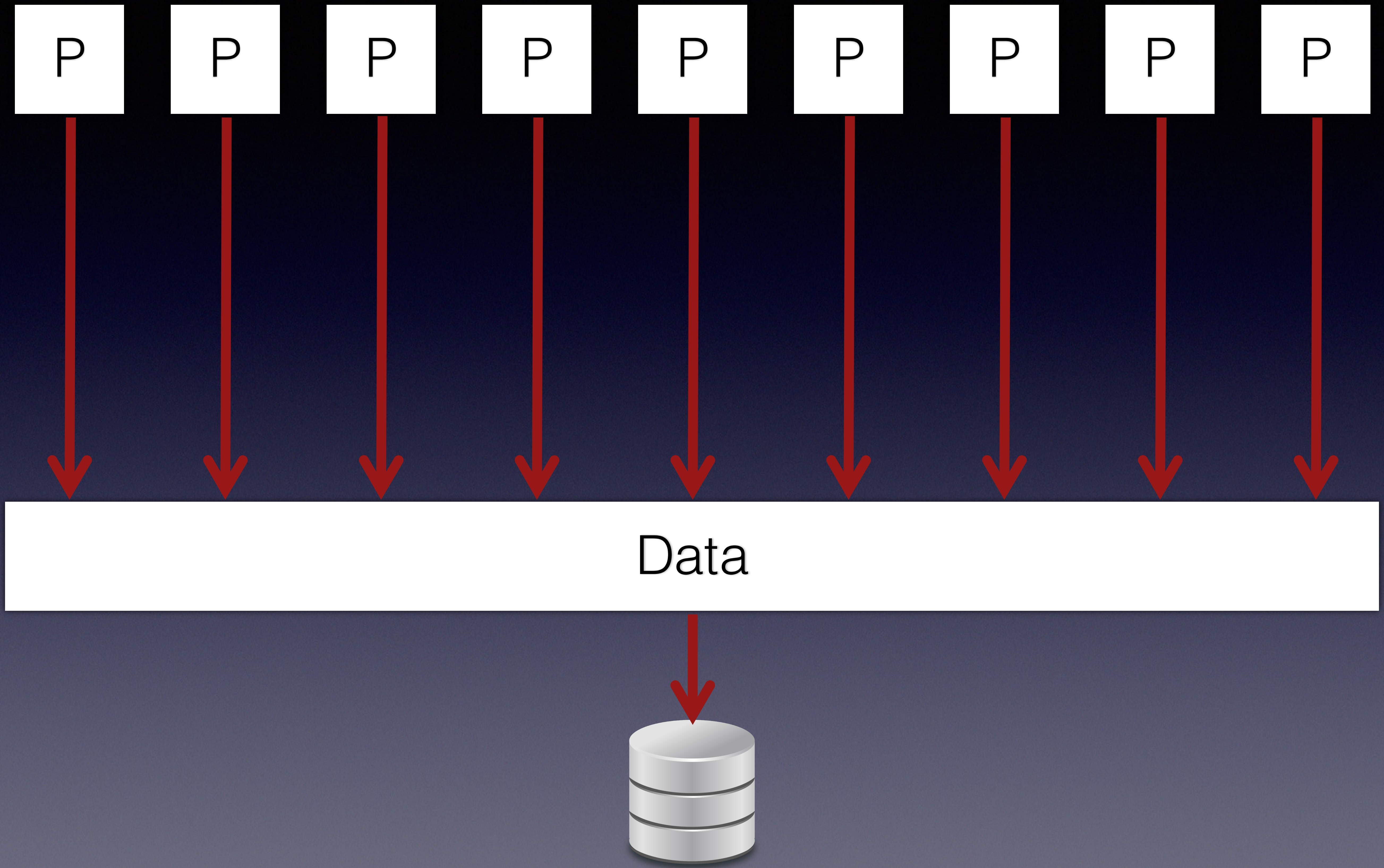


Presentation

Domain

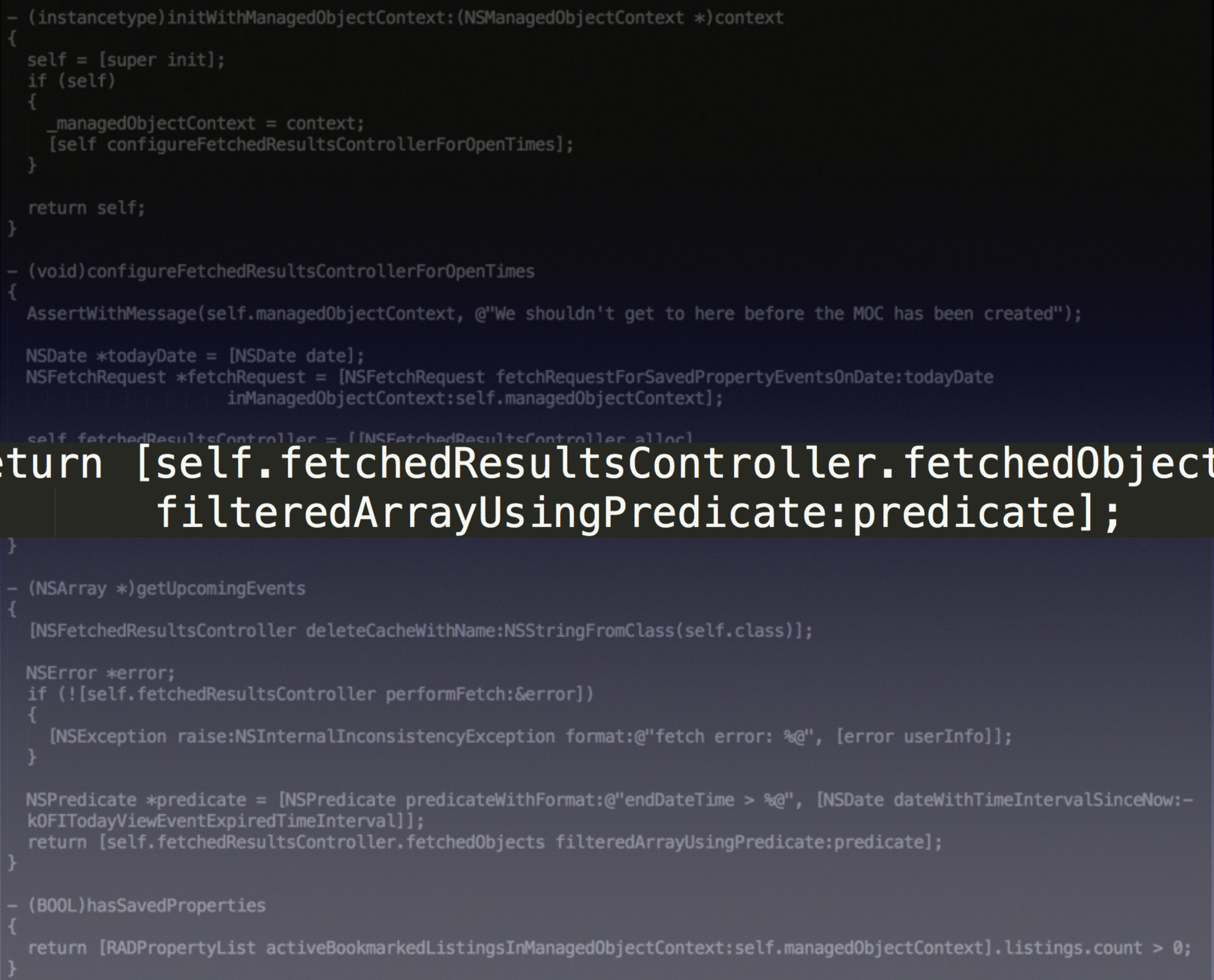
Data

而我们的系统呢？



P
P
P
P
P
P
P
P
P
Data

方式一：ViewController直接读写数据

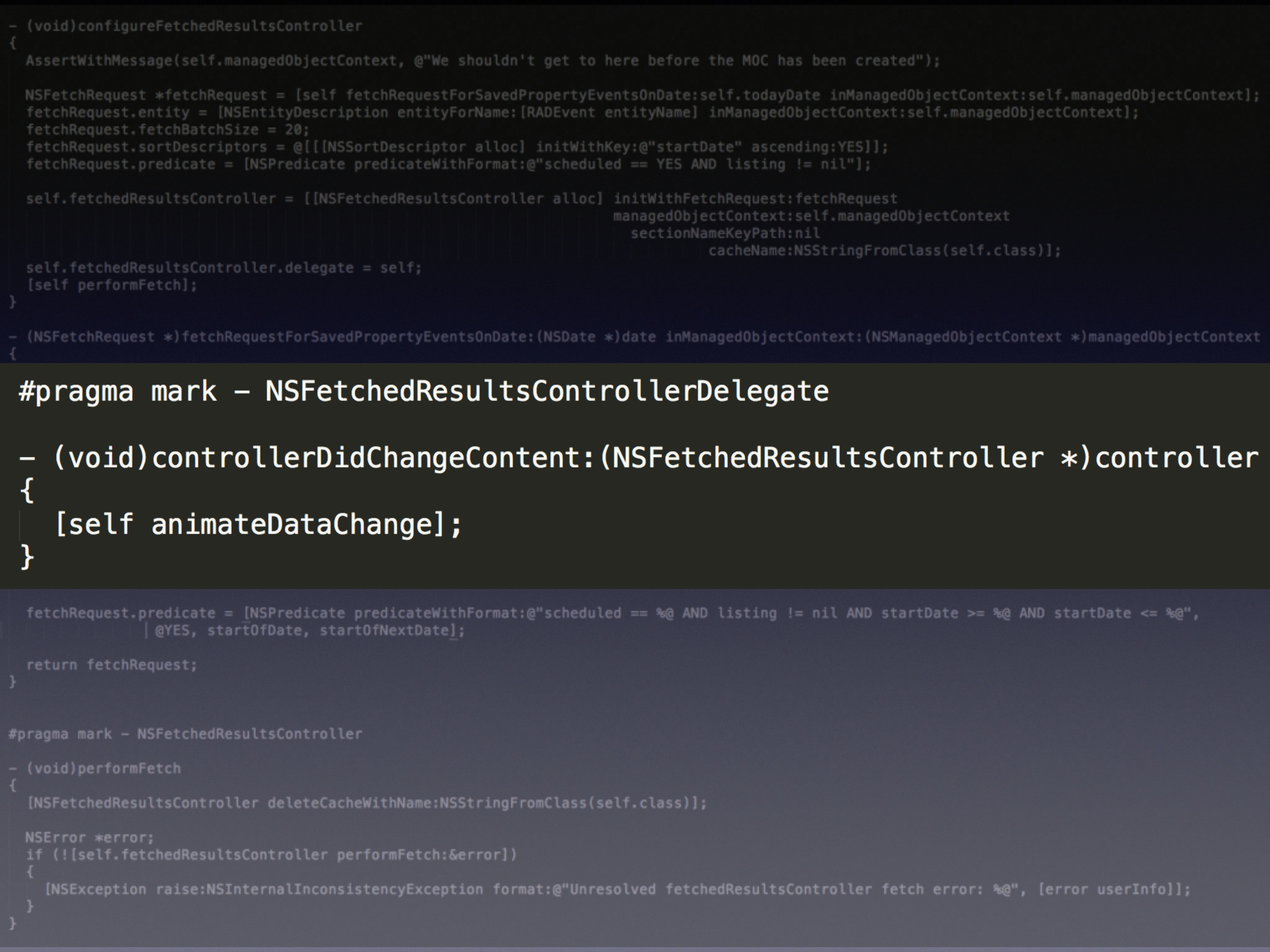


```
- (instancetype)initWithManagedObjectContext:(NSManagedObjectContext *)context
{
  self = [super init];
  if (self)
  {
    _managedObjectContext = context;
    [self configureFetchedResultsControllerForOpenTimes];
  }

  return self;
}

- (void)configureFetchedResultsControllerForOpenTimes
{
  AssertWithMessage(self.managedObjectContext, @"We shouldn't get to here before the MOC has been created");

  NSDate *todayDate = [NSDate date];
  NSFetchRequest *fetchRequest = [NSFetchRequest fetchRequestForSavedPropertyEventsOnDate:todayDate
                     inManagedObjectContext:self.managedObjectContext];

  self.fetchedResultsController = [[NSFetchedResultsController alloc]
```

```
return [self.fetchedResultsController.fetchedObjects
         filteredArrayUsingPredicate:predicate];
```

```
}

- (NSArray *)getUpcomingEvents
{
  [NSFetchedResultsController deleteCacheWithName:NSStringFromClass(self.class)];

  NSError *error;
  if (![self.fetchedResultsController performFetch:&error])
  {
    [NSException raise:NSInternalInconsistencyException format:@"fetch error: %@", [error userInfo]];
  }

  NSPredicate *predicate = [NSPredicate predicateWithFormat:@"endDateTime > %@", [NSDate dateWithTimeIntervalSinceNow:-
  kOFITodayViewEventExpiredTimeInterval]];
  return [self.fetchedResultsController.fetchedObjects filteredArrayUsingPredicate:predicate];
}

- (BOOL)hasSavedProperties
{
  return [RADPropertyList activeBookmarkedListingsInManagedObjectContext:self.managedObjectContext].listings.count > 0;
}
```

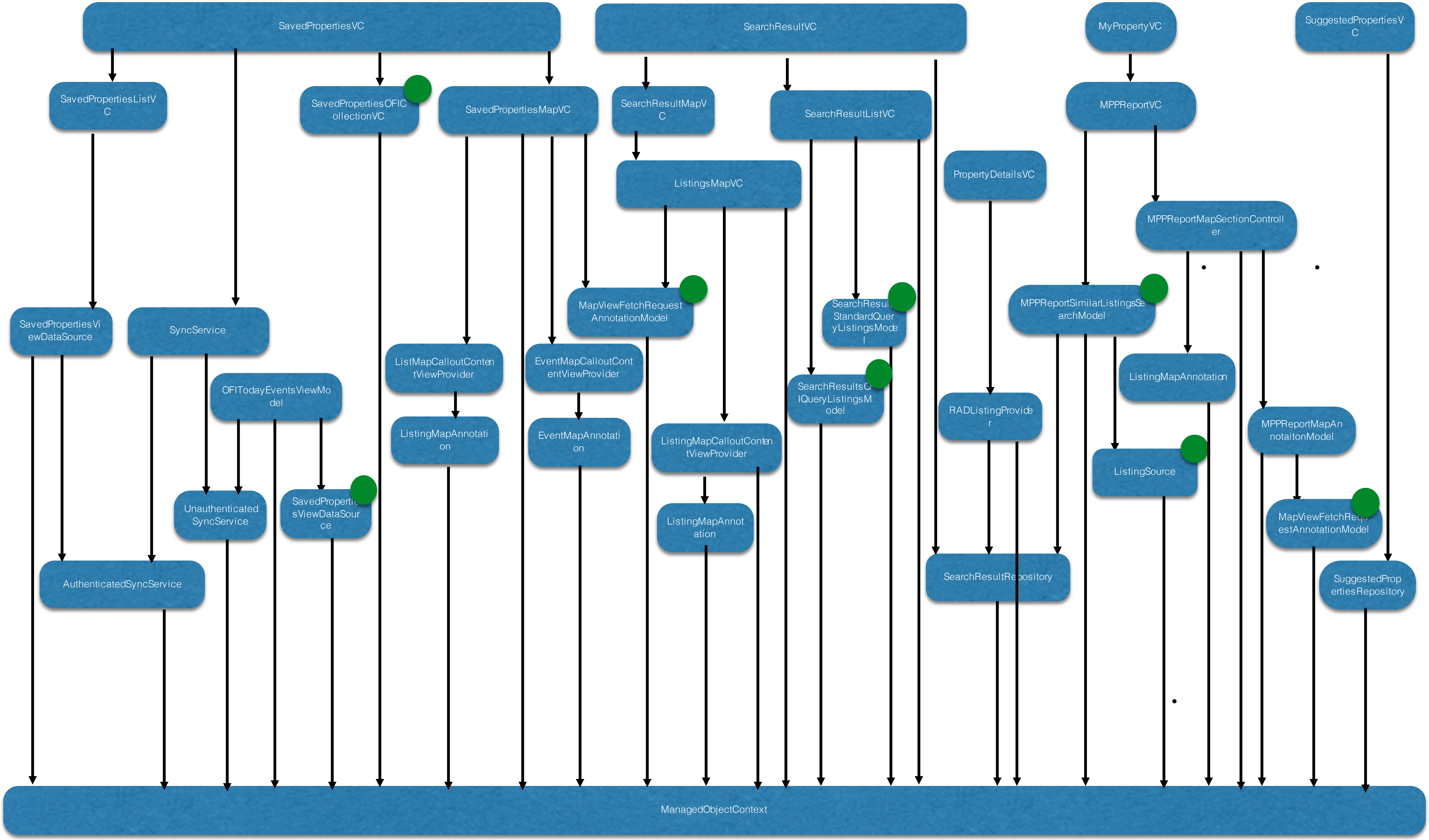
方式二：ViewController作为Delegate存在



```
- (void)configureFetchedResultsController
{
  AssertWithMessage(self.managedObjectContext, @"We shouldn't get to here before the MOC has been created");

  NSFetchRequest *fetchRequest = [self fetchRequestForSavedPropertyEventsOnDate:self.todayDate inManagedObjectContext:self.managedObjectContext];
  fetchRequest.entity = [NSEntityDescription entityForName:[RADEvent entityName] inManagedObjectContext:self.managedObjectContext];
  fetchRequest.fetchBatchSize = 20;
  fetchRequest.sortDescriptors = @[[[NSSortDescriptor alloc] initWithKey:@"startDate" ascending:YES]];
  fetchRequest.predicate = [NSPredicate predicateWithFormat:@"scheduled == YES AND listing != nil"];

  self.fetchedResultsController = [[NSFetchedResultsController alloc] initWithFetchRequest:fetchRequest
                                                                      managedObjectContext:self.managedObjectContext
                                                                        sectionNameKeyPath:nil
                                                                                 cacheName:NSStringFromClass(self.class)];
  self.fetchedResultsController.delegate = self;
  [self performFetch];
}

- (NSFetchRequest *)fetchRequestForSavedPropertyEventsOnDate:(NSDate *)date inManagedObjectContext:(NSManagedObjectContext *)managedObjectContext
{
```

```
#pragma mark - NSFetchedResultsControllerDelegate

- (void)controllerDidChangeContent:(NSFetchedResultsController *)controller
{
  [self animateDataChange];
}
```

```
  fetchRequest.predicate = [NSPredicate predicateWithFormat:@"scheduled == %@ AND listing != nil AND startDate >= %@ AND startDate <= %@",
                            @YES, startOfDate, startOfNextDate];

  return fetchRequest;
}


#pragma mark - NSFetchedResultsController

- (void)performFetch
{
  [NSFetchedResultsController deleteCacheWithName:NSStringFromClass(self.class)];

  NSError *error;
  if (![self.fetchedResultsController performFetch:&error])
  {
    [NSException raise:NSInternalInconsistencyException format:@"Unresolved fetchedResultsController fetch error: %@", [error userInfo]];
  }
}
```



SavedPropertiesVC

SearchResultVC

MyPropertyVC

SuggestedPropertiesVC

SavedPropertiesListVC

SavedPropertiesOFICollectionVC

SavedPropertiesMapVC

SearchResultMapVC

SearchResultListVC

MPPReportVC

ListingsMapVC

PropertyDetailsVC

MPPReportMapSectionController

MapViewFetchRequestAnnotationModel

SearchResul StandardQueryListingsModel

MPPReportSimilarListingsSearchModel

SavedPropertiesViewDataSource

SyncService

ListMapCalloutContentViewProvider

EventMapCalloutContentViewProvider

SearchResultsC lQueryListingsModel

ListingMapAnnotation

OFITodayEventsViewModel

RADListingProvider

MPPReportMapAnnotaitonModel

ListingMapAnnotation

EventMapAnnotation

ListingMapCalloutContentViewProvider

ListingSource

UnauthenticatedSyncService

SavedPropertie sViewDataSource

ListingMapAnnotation

MapViewFetchRequ estAnnotationModel

SearchResultRepository

AuthenticatedSyncService

SuggestedPropertiesRepository

ManagedObjectContext

40+ NSManagedObjects

AGREEMENT

干脆从头再来吧

理想很丰满 现实很骨感

长时间无法交付

范围失去控制

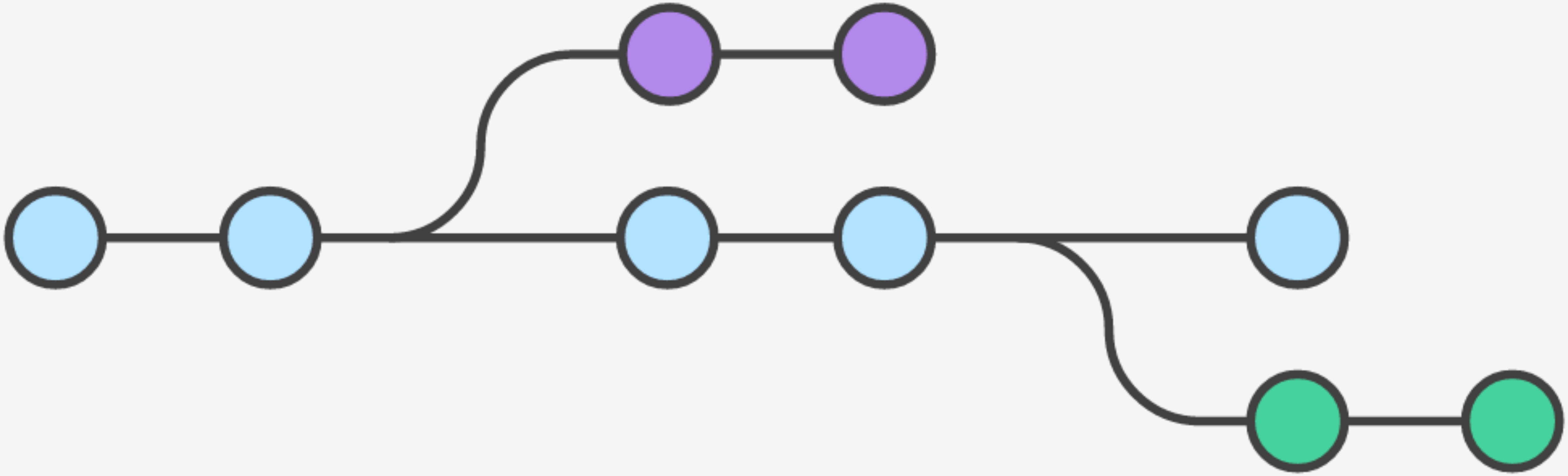
Elsa, do you wanna rewrite the app?

Go away, Anna

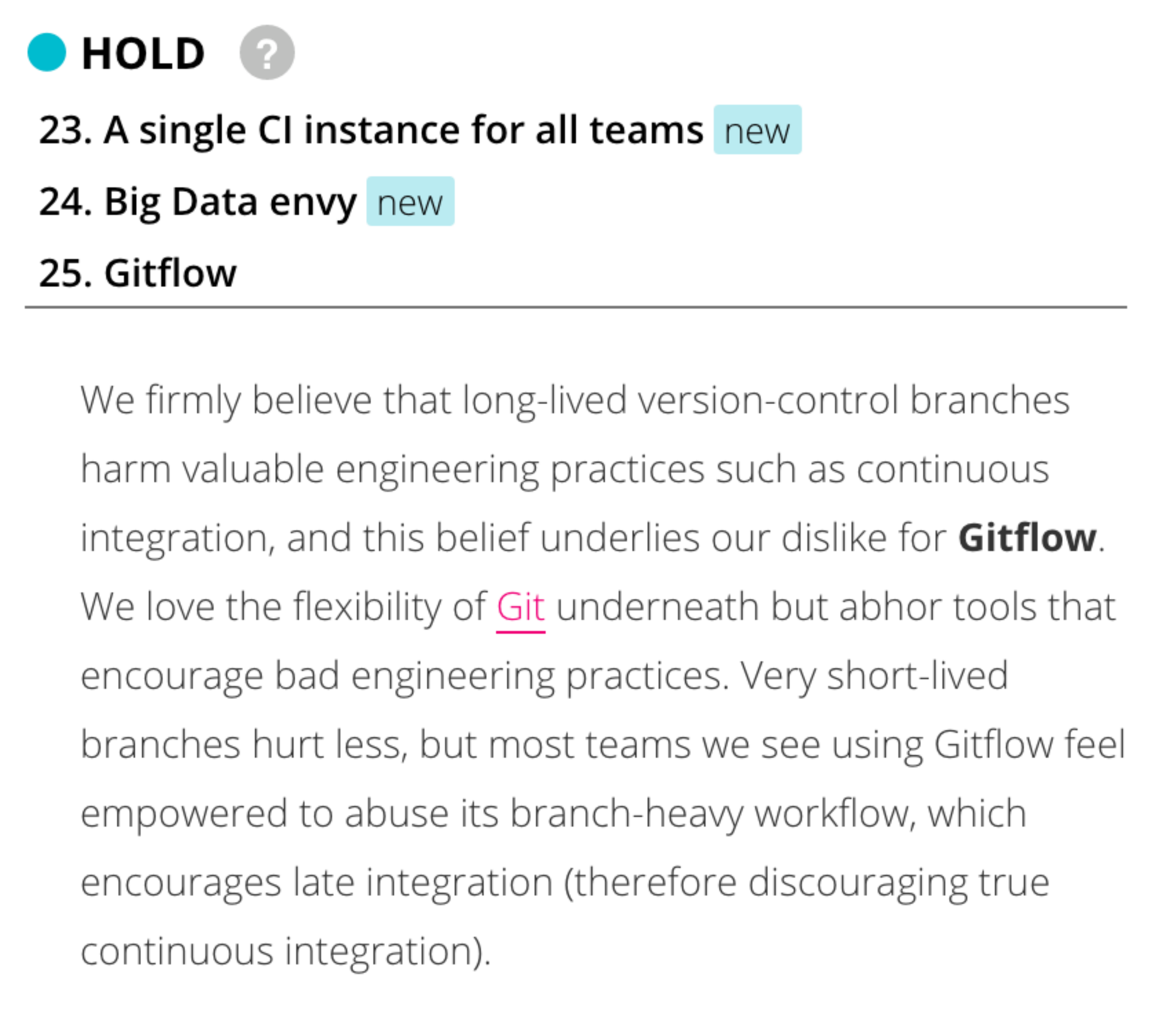
Ok, bye

~~重層~~

特性分支？



BOJIJI
ISE OLUWA N'



HOLD

23. A single CI instance for all teams new

24. Big Data envy new

25. Gitflow

We firmly believe that long-lived version-control branches harm valuable engineering practices such as continuous integration, and this belief underlies our dislike for **Gitflow**. We love the flexibility of Git underneath but abhor tools that encourage bad engineering practices. Very short-lived branches hurt less, but most teams we see using Gitflow feel empowered to abuse its branch-heavy workflow, which encourages late integration (therefore discouraging true continuous integration).

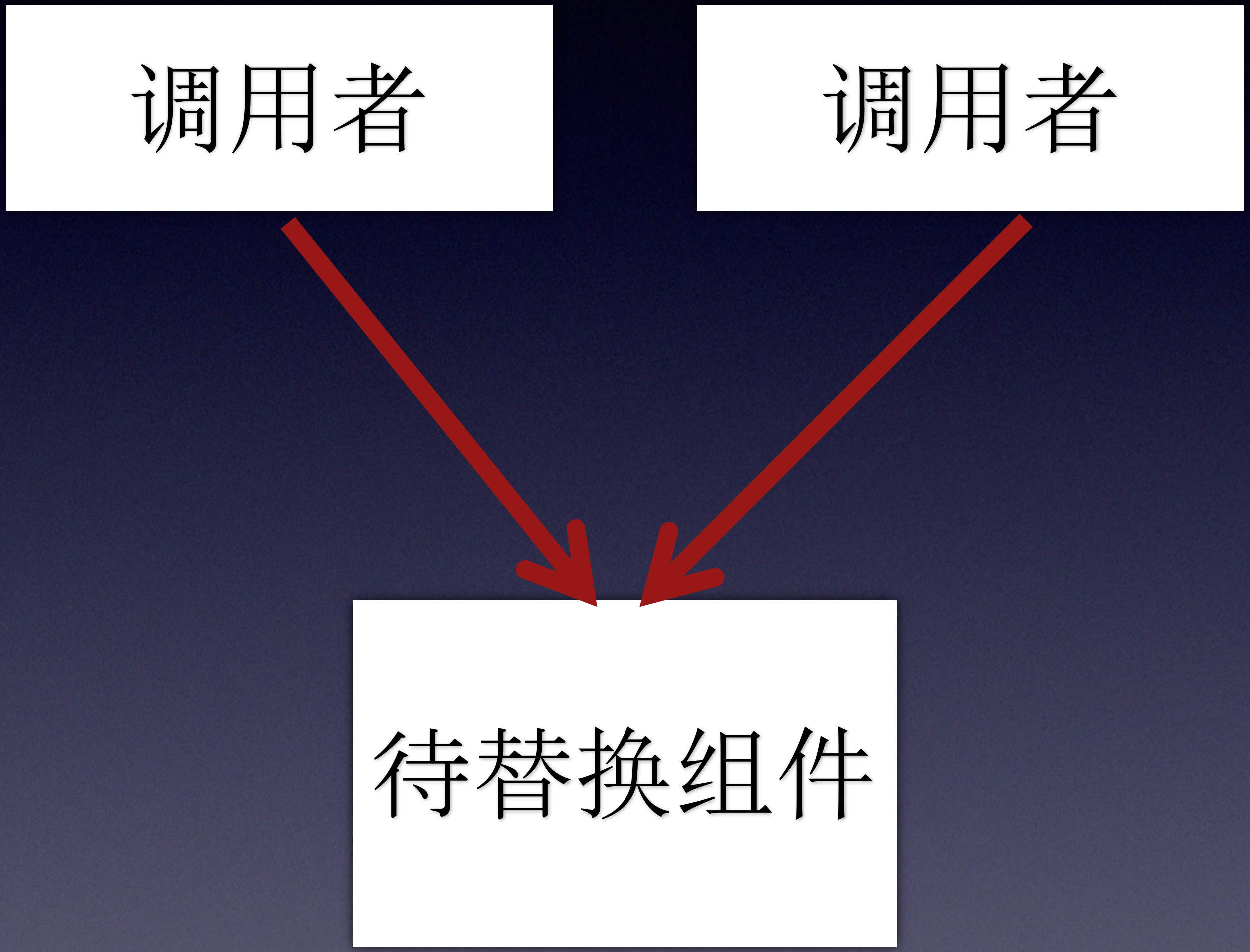
我们想把所有的事情都放到master上做

~~特性分支~~

抽象分支（Branch by abstraction）

Branch by abstraction is a technique for making large-scale changes to a system without disrupting its operations.

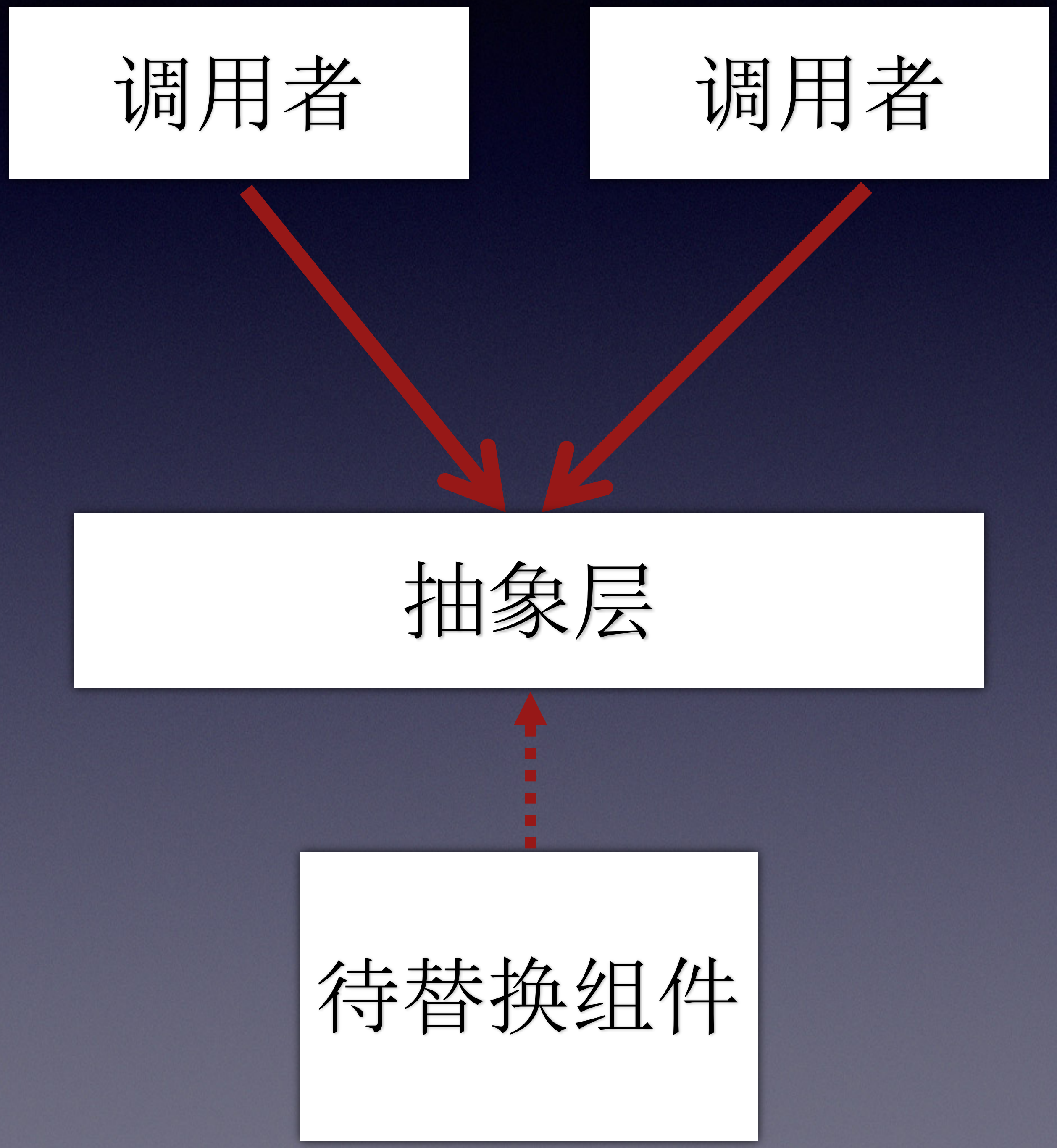
http://martinfowler.com/bliki/BranchByAbstraction.html



调用者

调用者

待替换组件

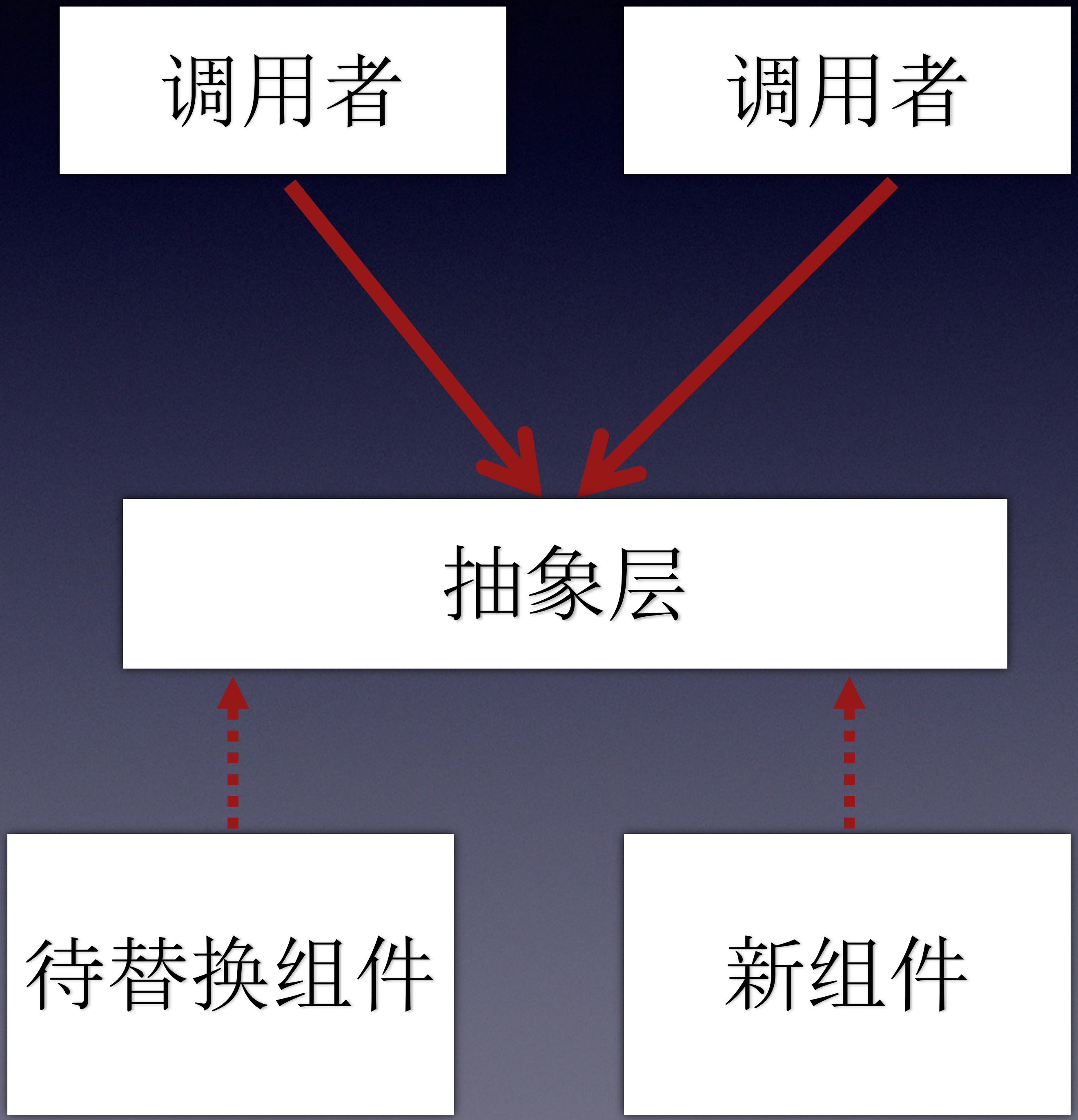


调用者

调用者

抽象层

待替换组件



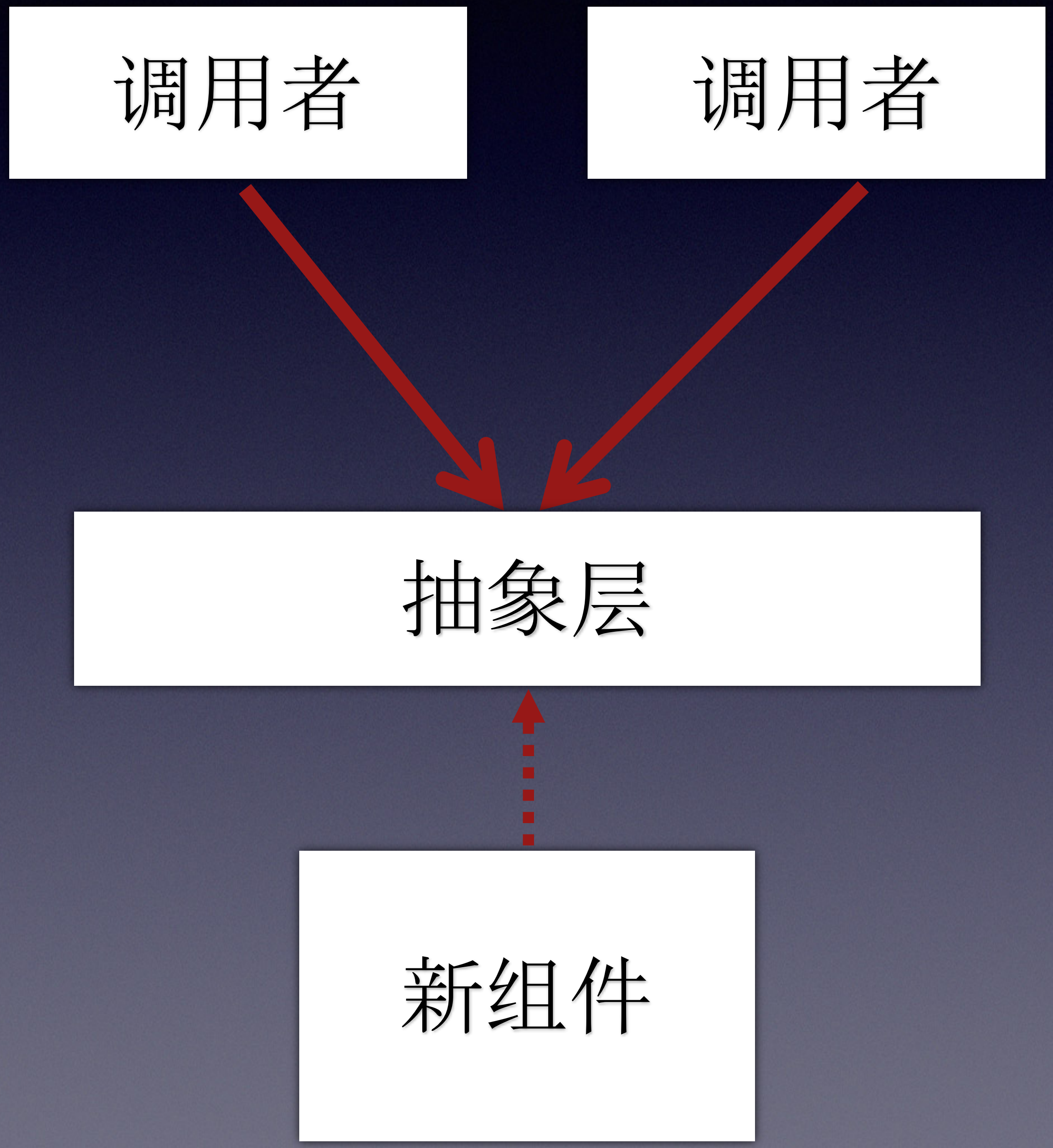
调用者

调用者

抽象层

待替换组件

新组件



调用者

调用者

抽象层

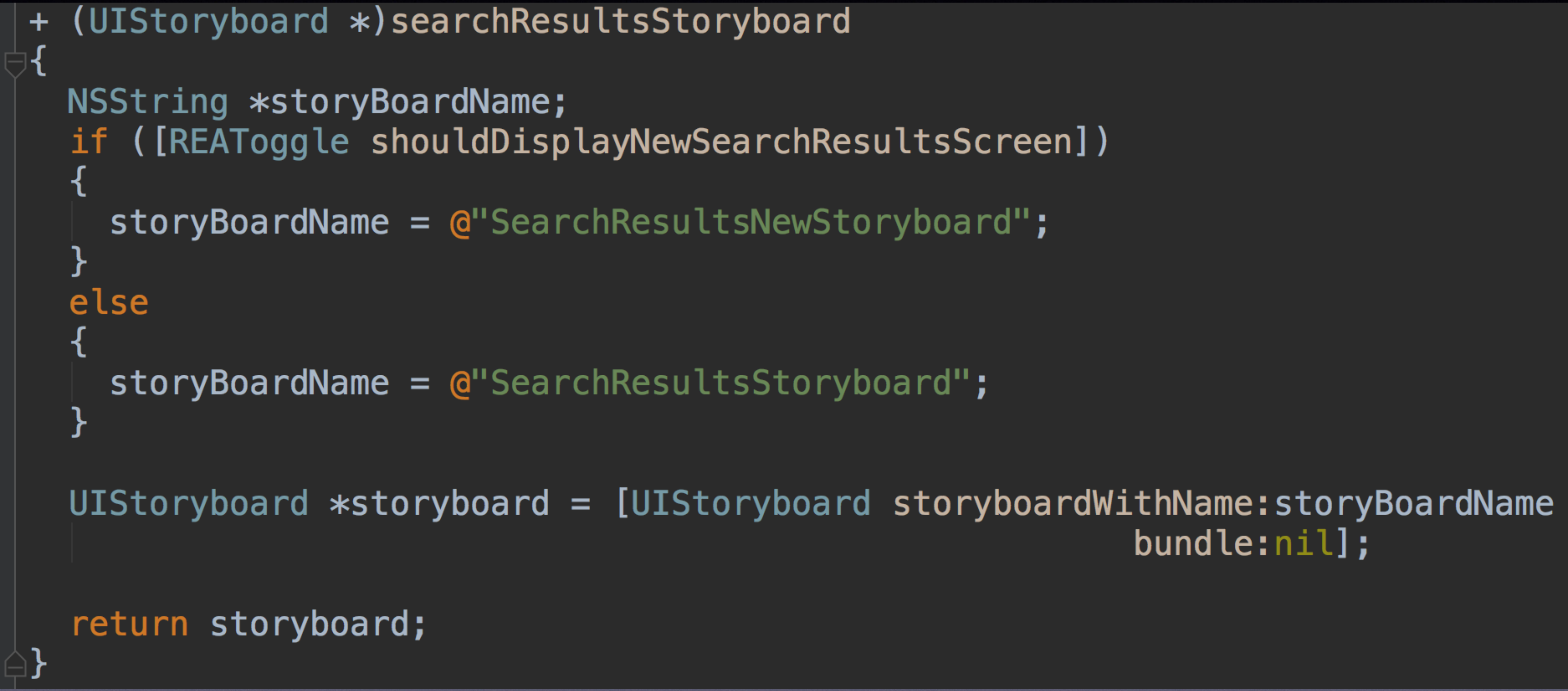
新组件

现有功能不受影响

每次提交均可合并到主干

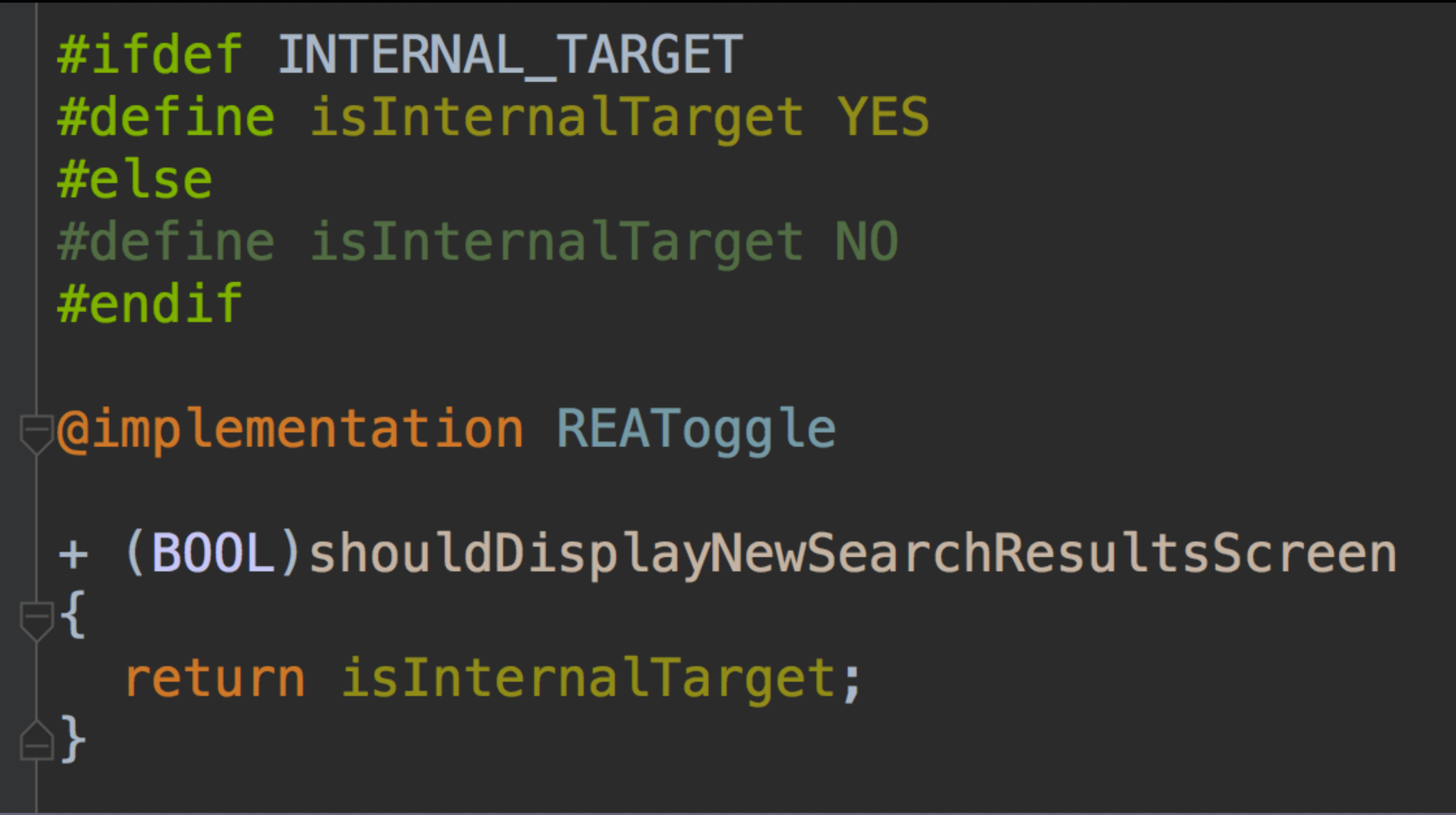
用单元测试和集成测试验证新组件

特性开关



```
+ (UIStoryboard *)searchResultsStoryboard
{
  NSString *storyBoardName;
  if ([REAToggle shouldDisplayNewSearchResultsScreen])
  {
    storyBoardName = @"SearchResultsNewStoryboard";
  }
  else
  {
    storyBoardName = @"SearchResultsStoryboard";
  }

  UIStoryboard *storyboard = [UIStoryboard storyboardWithName:storyBoardName
                                                       bundle:nil];

  return storyboard;
}
```

方案1: GCC_PREPROCESSOR_DEFINITIONS

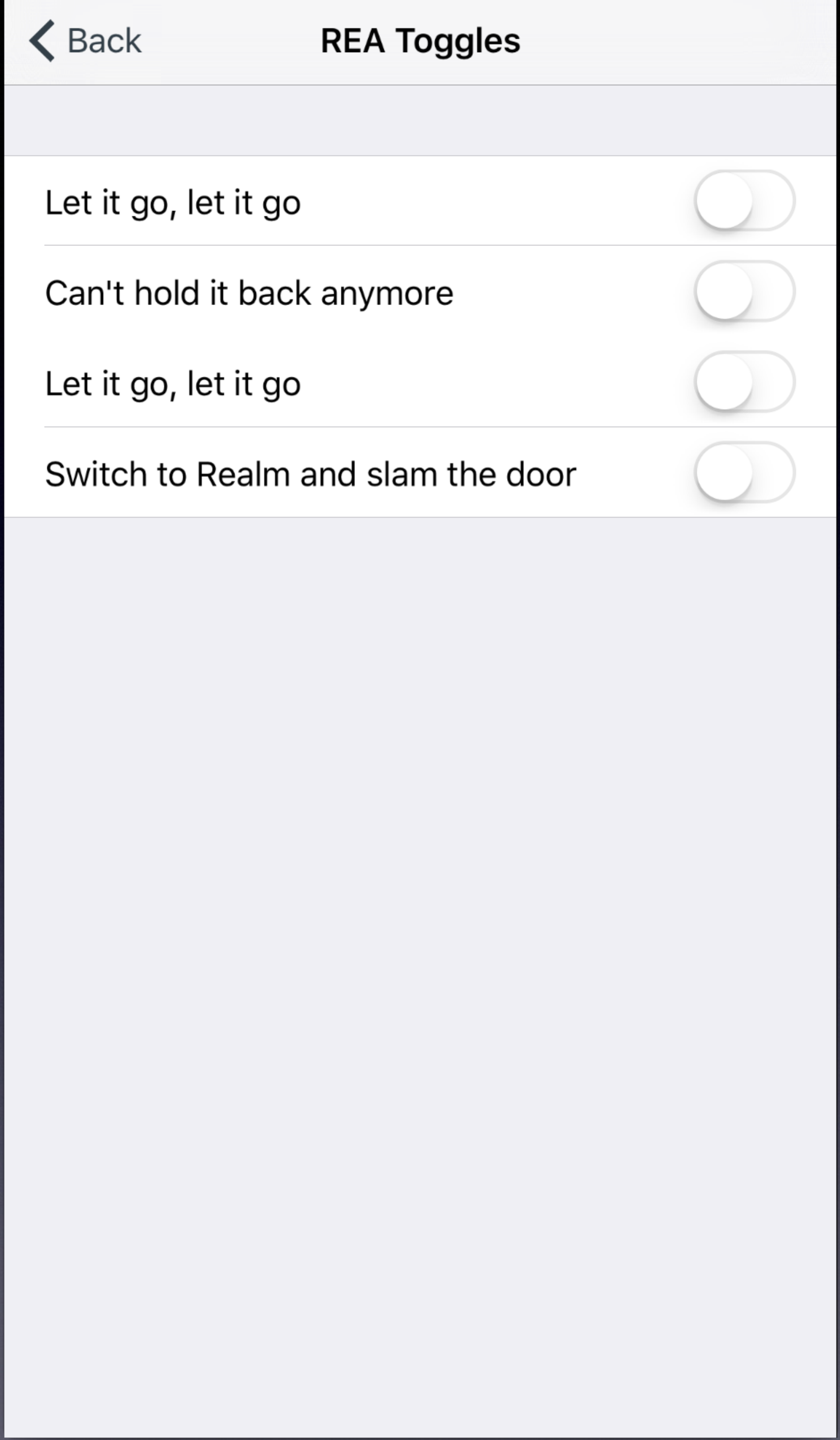


```
#ifdef INTERNAL_TARGET
#define isInternalTarget YES
#else
#define isInternalTarget NO
#endif

@implementation REAToggle

+ (BOOL)shouldDisplayNewSearchResultsScreen
{
  return isInternalTarget;
}
```

所有人都基于主干开发

用户看不到未完成的功能

方案2: NSUserDefaults



REA Toggles

Let it go, let it go ○

Can't hold it back anymore ○

Let it go, let it go ○

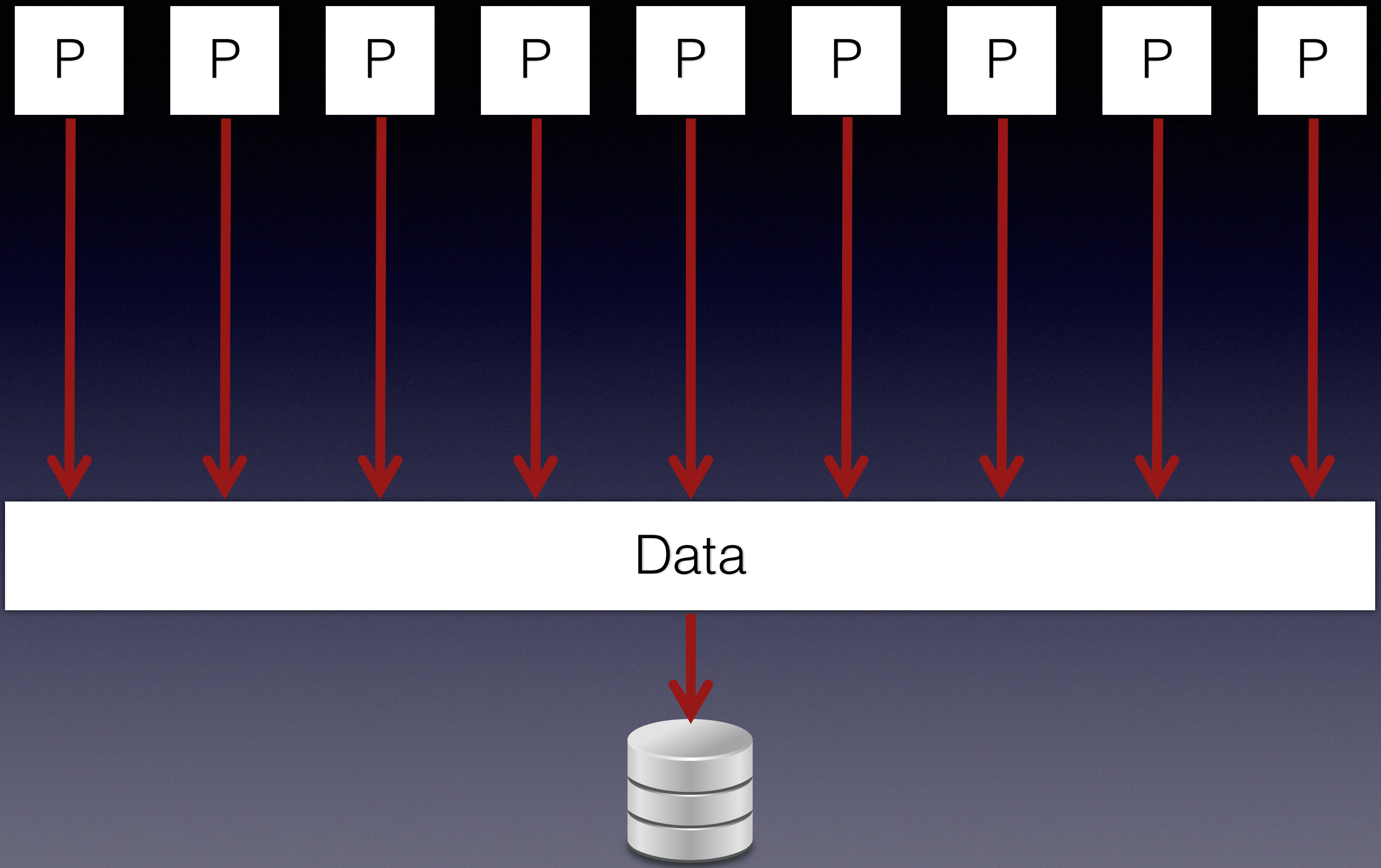
Switch to Realm and slam the door ○

方案3: 调用后台服务

方案4: A/B测试框架

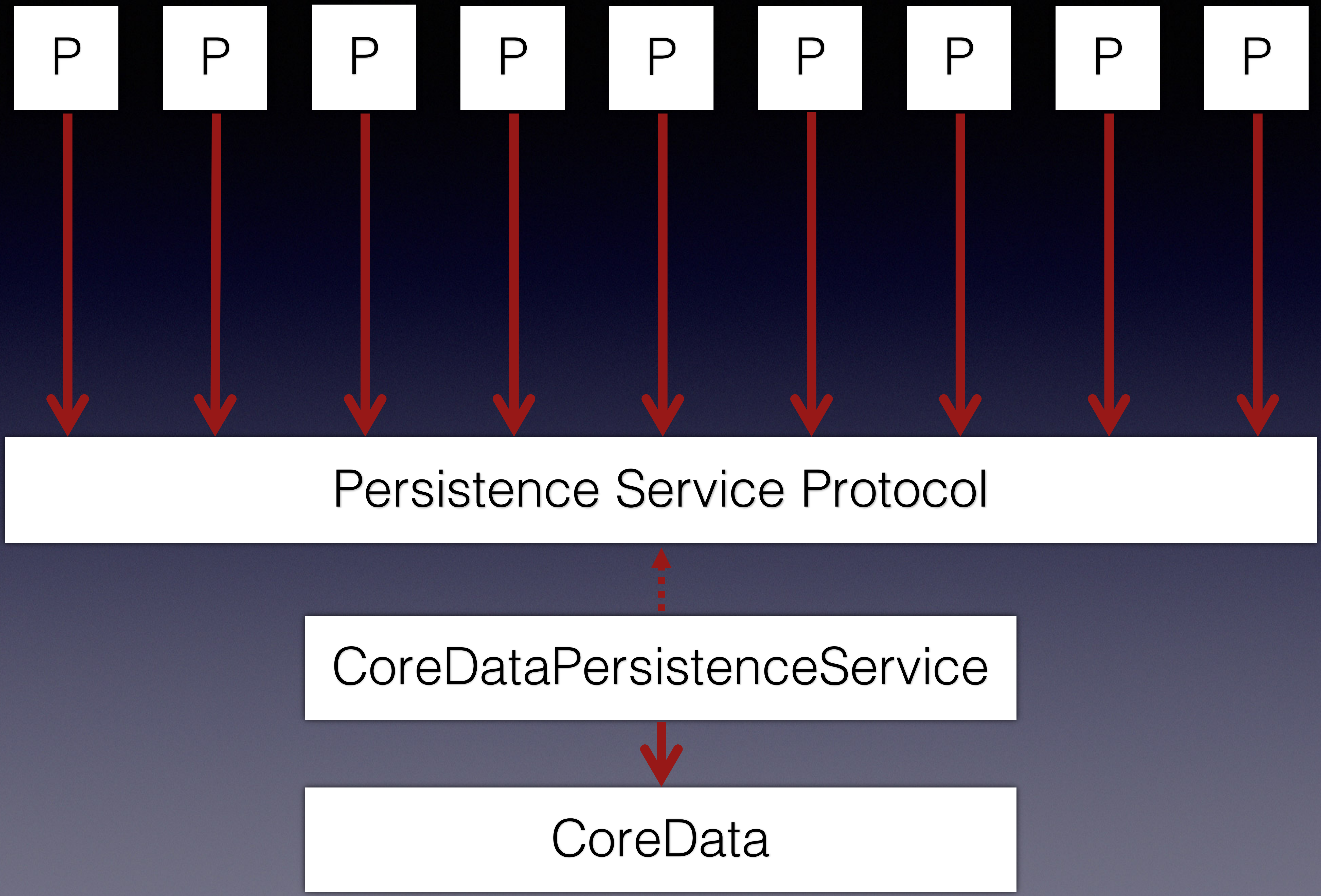
抽象分支 + 特性开关

建立数据访问层



P P P P P P P P P

Data

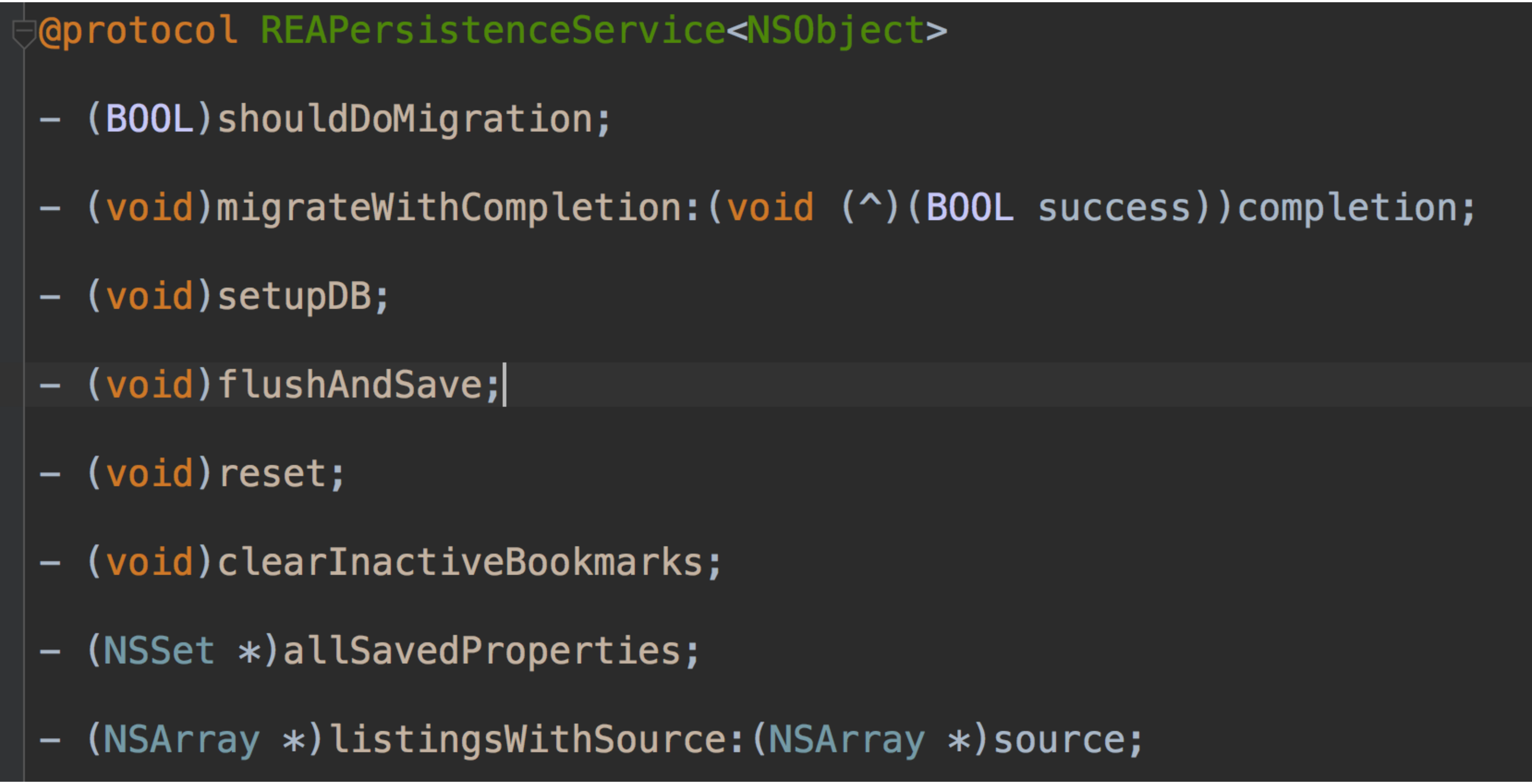


P P P P P P P P P

Persistence Service Protocol

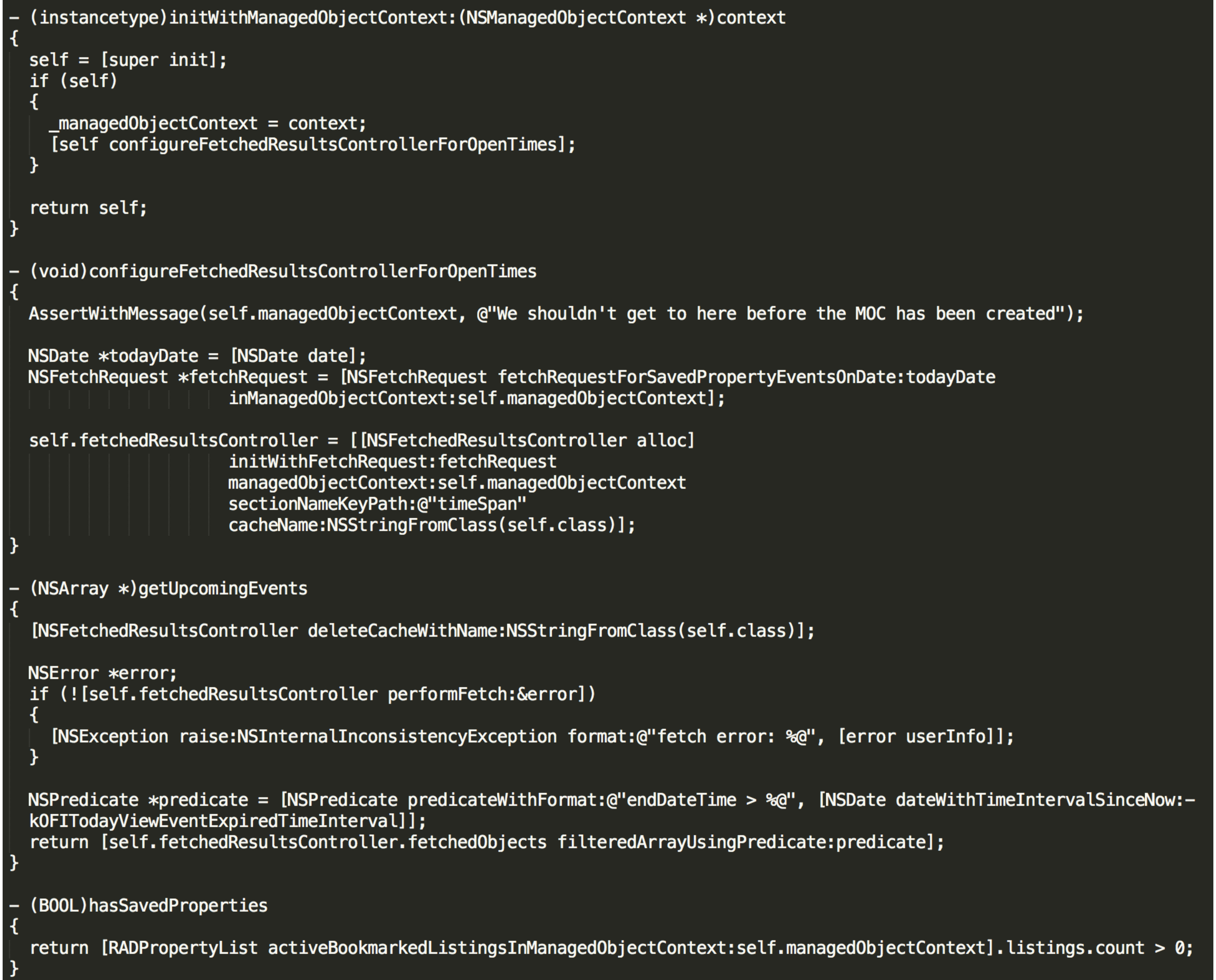
CoreDataPersistenceService

CoreData

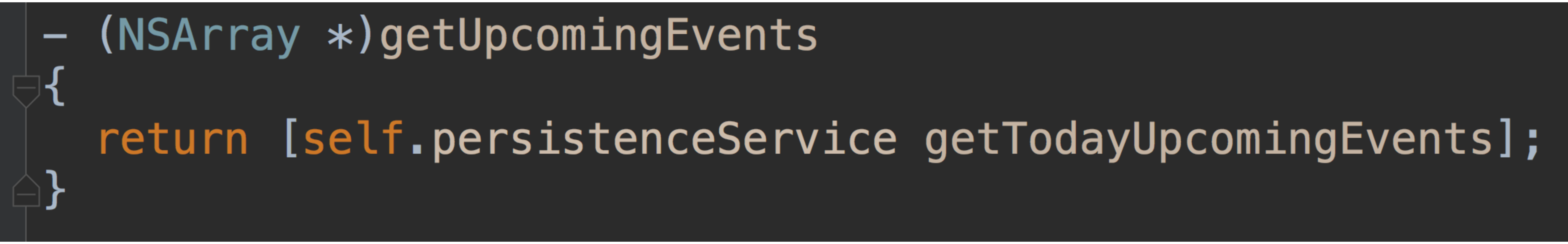


```
@protocol REAPersistenceService<NSObject>

- (BOOL)shouldDoMigration;

- (void)migrateWithCompletion:(void (^)(BOOL success))completion;

- (void)setupDB;

- (void)flushAndSave;

- (void)reset;

- (void)clearInactiveBookmarks;

- (NSSet *)allSavedProperties;

- (NSArray *)listingsWithSource:(NSArray *)source;
```

方式一：ViewController直接读写数据



```
- (instancetype)initWithManagedObjectContext:(NSManagedObjectContext *)context
{
  self = [super init];
  if (self)
  {
    _managedObjectContext = context;
    [self configureFetchedResultsControllerForOpenTimes];
  }

  return self;
}

- (void)configureFetchedResultsControllerForOpenTimes
{
  AssertWithMessage(self.managedObjectContext, @"We shouldn't get to here before the MOC has been created");

  NSDate *todayDate = [NSDate date];
  NSFetchRequest *fetchRequest = [NSFetchRequest fetchRequestForSavedPropertyEventsOnDate:todayDate
                      inManagedObjectContext:self.managedObjectContext];

  self.fetchedResultsController = [[NSFetchedResultsController alloc]
                      initWithFetchRequest:fetchRequest
                      managedObjectContext:self.managedObjectContext
                      sectionNameKeyPath:@"timeSpan"
                      cacheName:NSStringFromClass(self.class)];
}

- (NSArray *)getUpcomingEvents
{
  [NSFetchedResultsController deleteCacheWithName:NSStringFromClass(self.class)];

  NSError *error;
  if (![self.fetchedResultsController performFetch:&error])
  {
    [NSException raise:NSInternalInconsistencyException format:@"fetch error: %@", [error userInfo]];
  }

  NSPredicate *predicate = [NSPredicate predicateWithFormat:@"endDateTime > %@", [NSDate dateWithTimeIntervalSinceNow:-
  kOFITodayViewEventExpiredTimeInterval]];
  return [self.fetchedResultsController.fetchedObjects filteredArrayUsingPredicate:predicate];
}

- (BOOL)hasSavedProperties
{
  return [RADPropertyList activeBookmarkedListingsInManagedObjectContext:self.managedObjectContext].listings.count > 0;
}
```



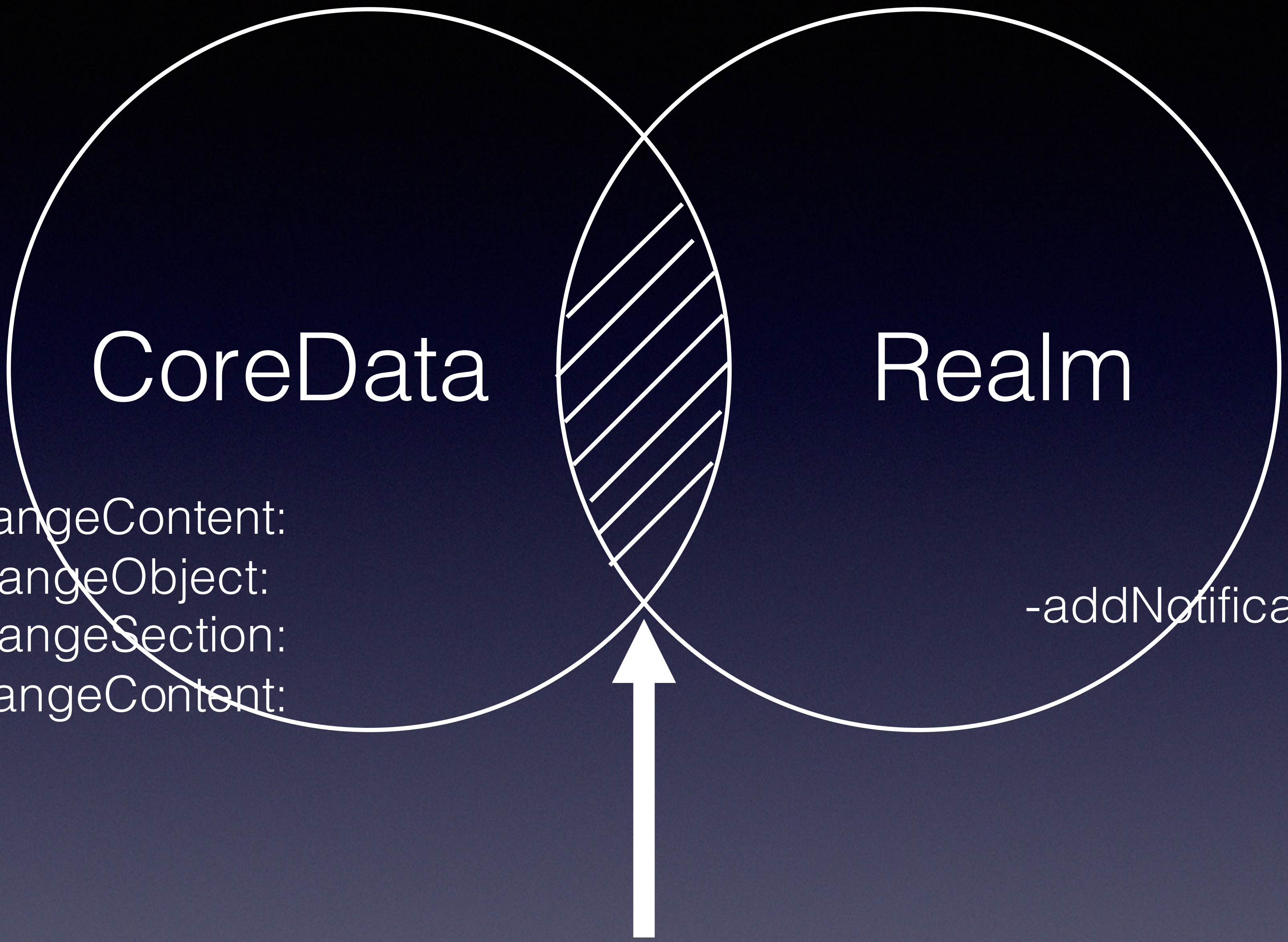
```
- (NSArray *)getUpcomingEvents
{
    return [self.persistenceService getTodayUpcomingEvents];
}
```

方式二：ViewController作为Delegate存在

- controllerWillChangeContent:
- controller:didChangeObject:
- controller:didChangeSection:
- controllerDidChangeContent:

VS

-addNotificationBlock:



CoreData

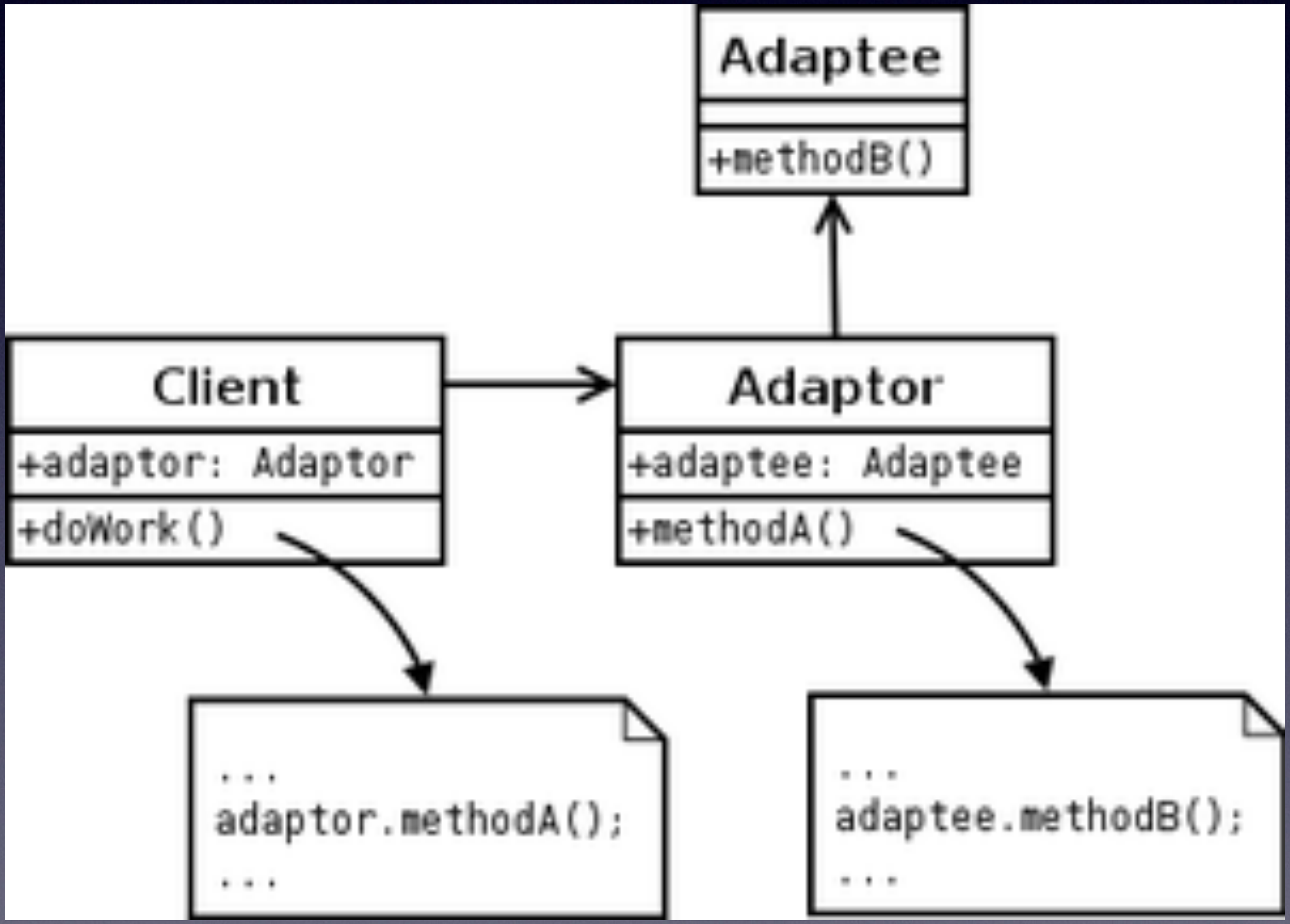
Realm

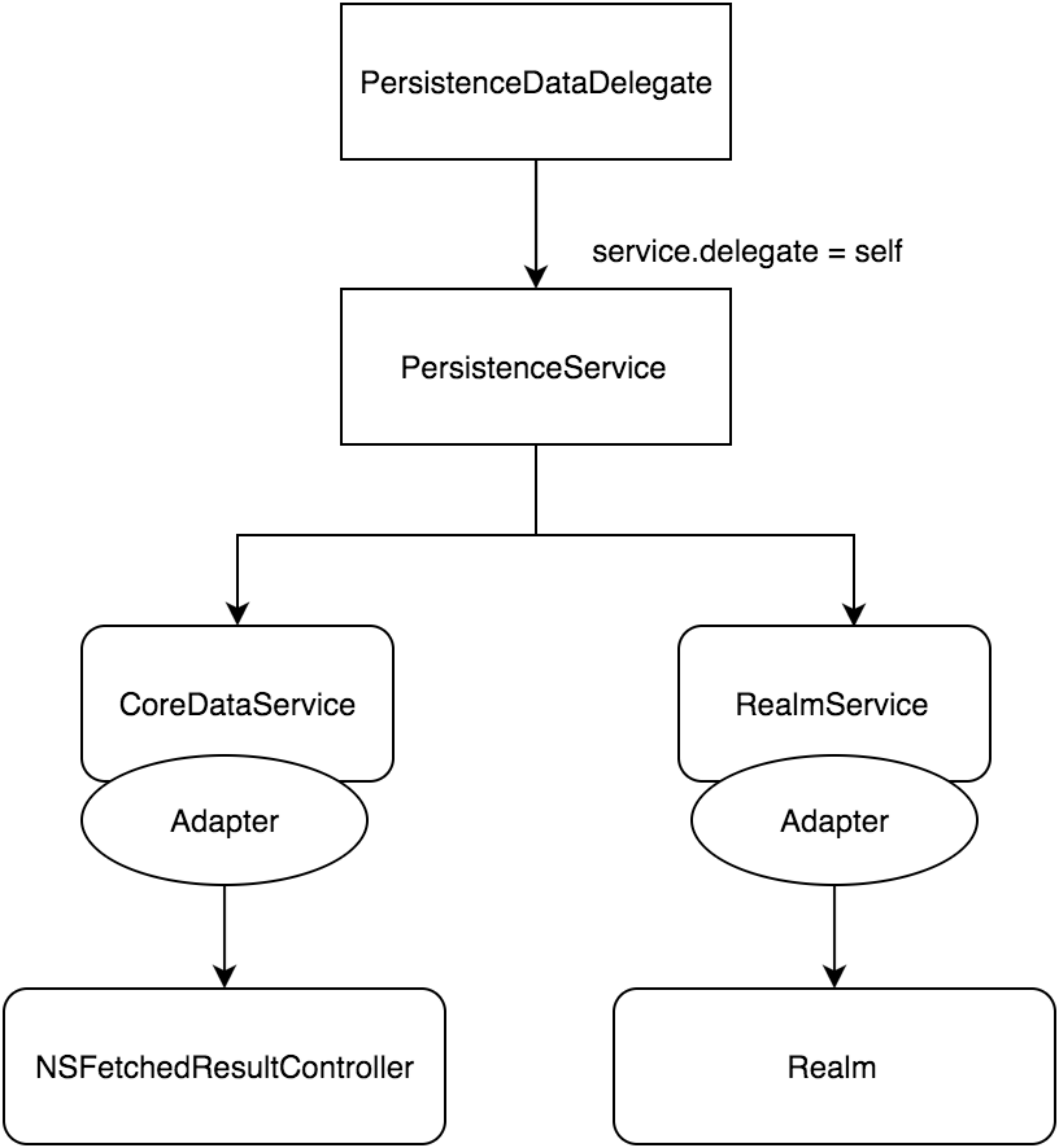
- controllerWillChangeContent:
- controller:didChangeObject:
- controller:didChangeSection:
- controllerDidChangeContent:

-addNotificationBlock:

- contentDidChange:(id)content

Adapter模式

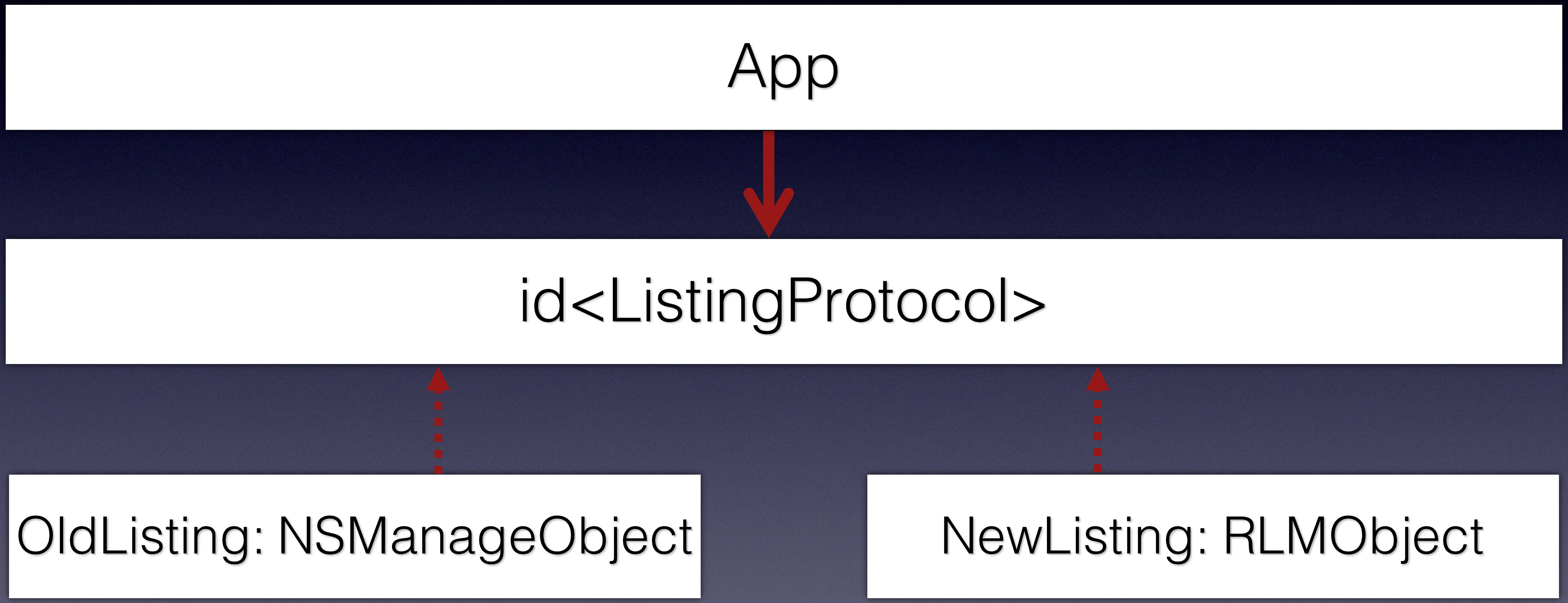

Adaptee
+methodB()
Client
+adaptor: Adaptor
+doWork()
Adaptor
+adaptee: Adaptee
+methodA()
...
adaptor.methodA();
...
...
adaptee.methodB();
...


PersistenceDataDelegate
service.delegate = self
PersistenceService
CoreDataService
Adapter
RealmService
Adapter
NSFetchedResultController
Realm

建立数据访问层

建立数据访问层

为数据对象提取协议


App
id<ListingProtocol>
OldListing: NSManageObject
NewListing: RLMObject

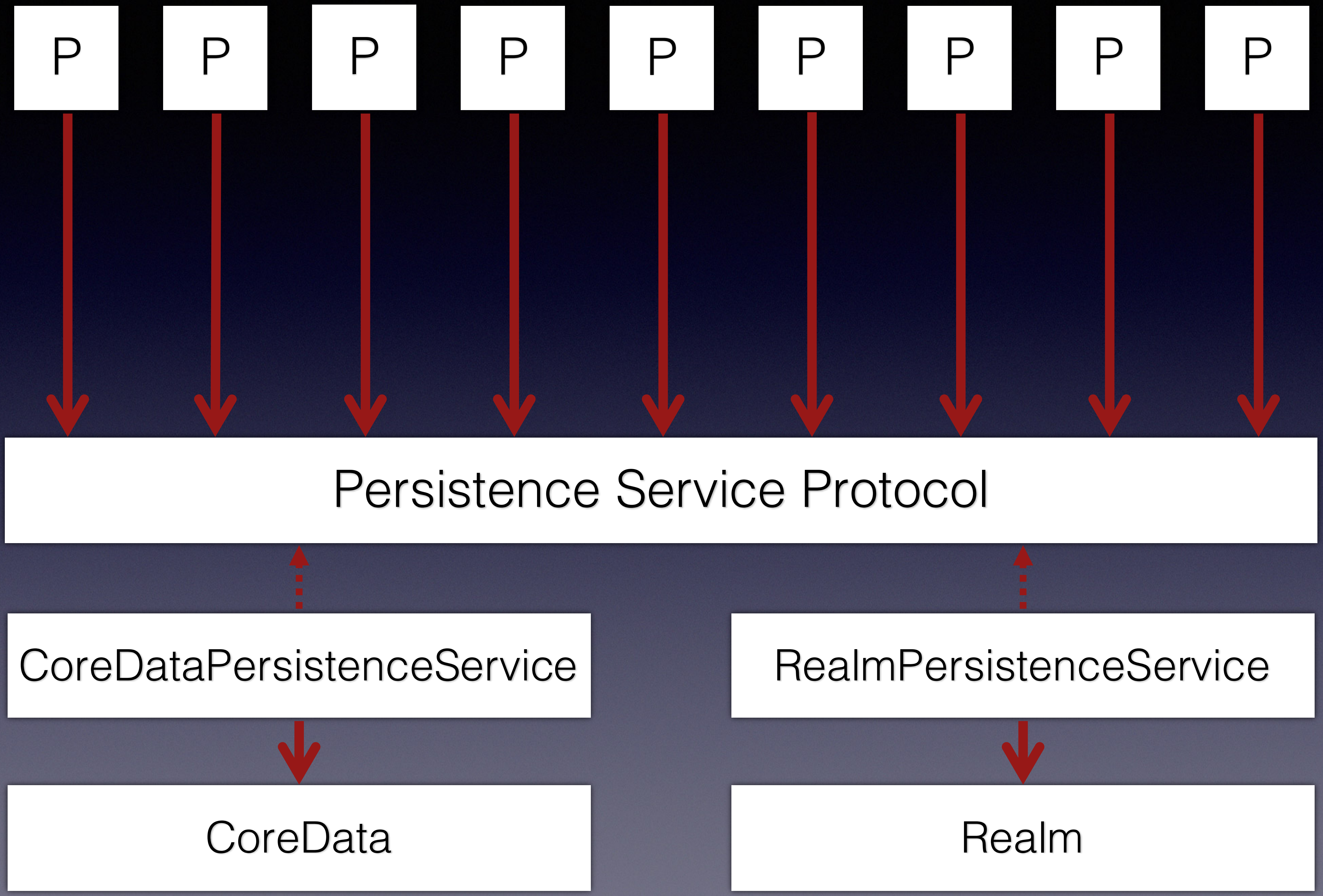
建立数据访问层

为数据对象提取协议

建立数据访问层

为数据对象提取协议

新的持久化服务



P P P P P P P P P

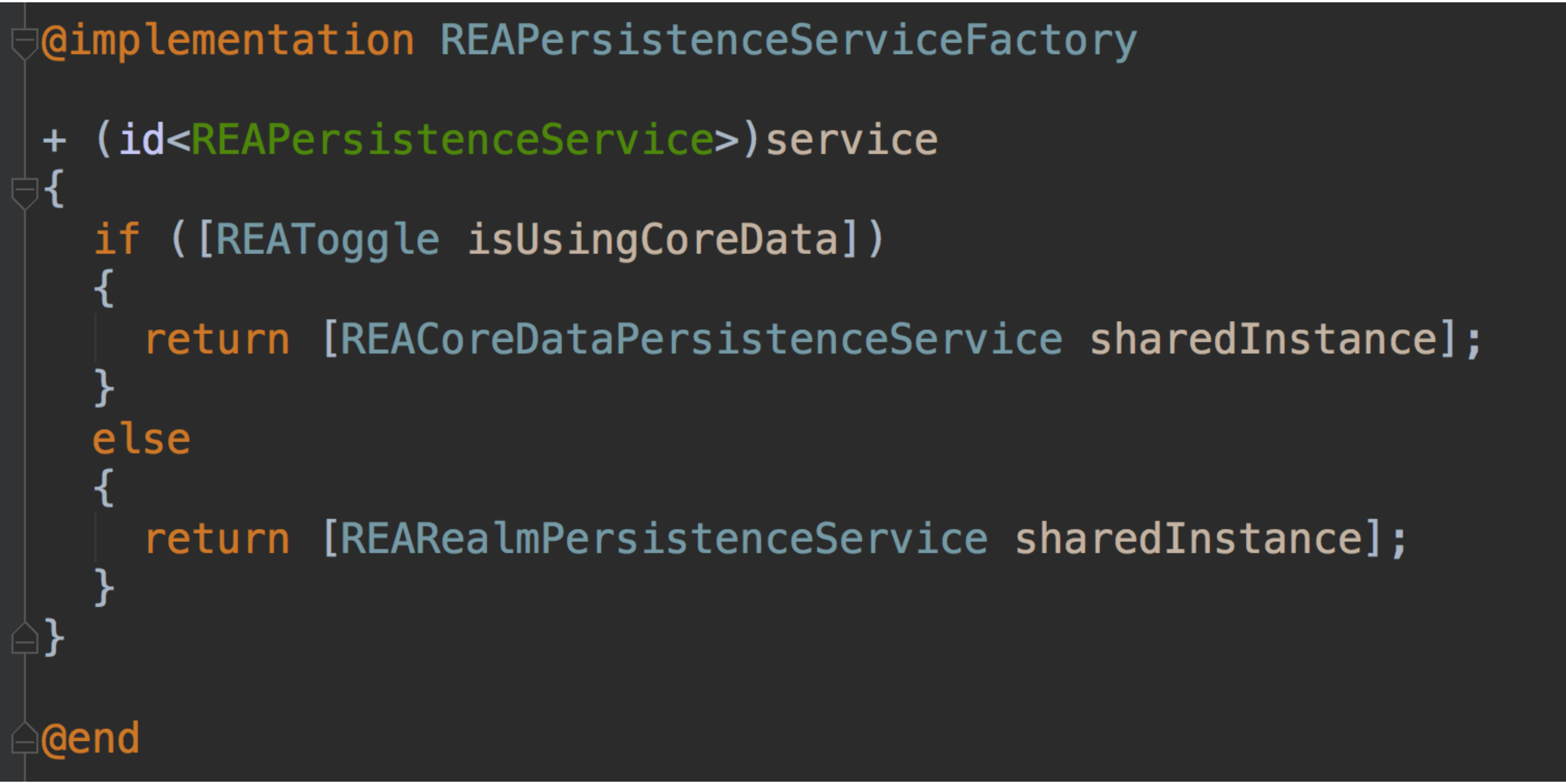
Persistence Service Protocol

CoreDataPersistenceService

RealmPersistenceService

CoreData

Realm



```objectivec
@implementation REAPersistenceServiceFactory

+ (id<REAPersistenceService>)service
{
  if ([REAToggle isUsingCoreData])
  {
    return [REACoreDataPersistenceService sharedInstance];
  }
  else
  {
    return [REARealmPersistenceService sharedInstance];
  }
}

@end
```

当心坑洞

不支持级联删除

不支持多态

难以理解的反向关联

建立数据访问层

为数据对象提取协议

新的持久化服务

建立数据访问层

为数据对象提取协议

新的持久化服务

打开开关，内测

建立数据访问层

为数据对象提取协议

3. New persistence service

打开开关,内测

数据迁移

建立数据访问层

为数据对象提取协议

新的持久化服务

打开开关，内测

数据迁移

后续清理

建立数据访问层

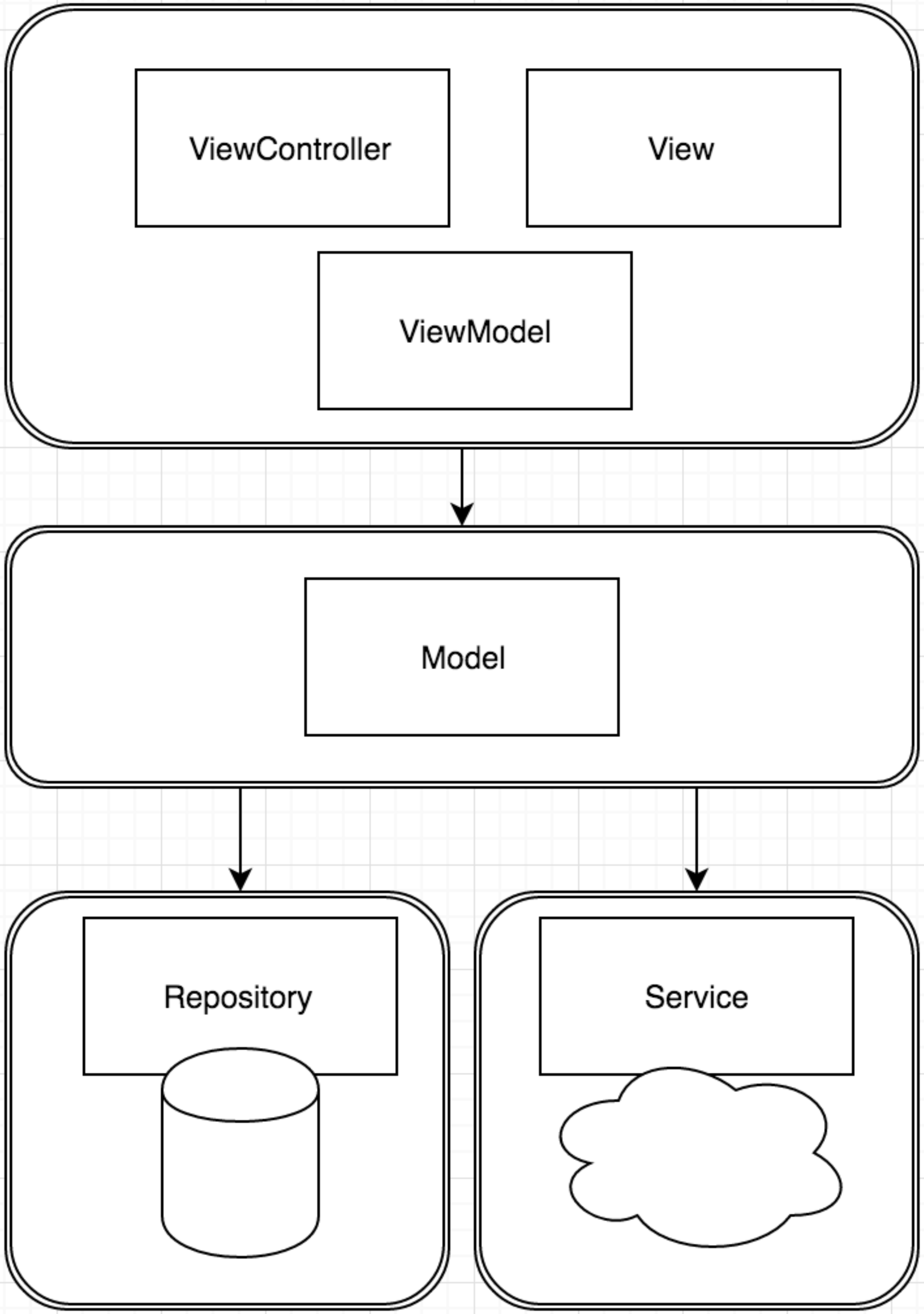
为数据对象提取协议

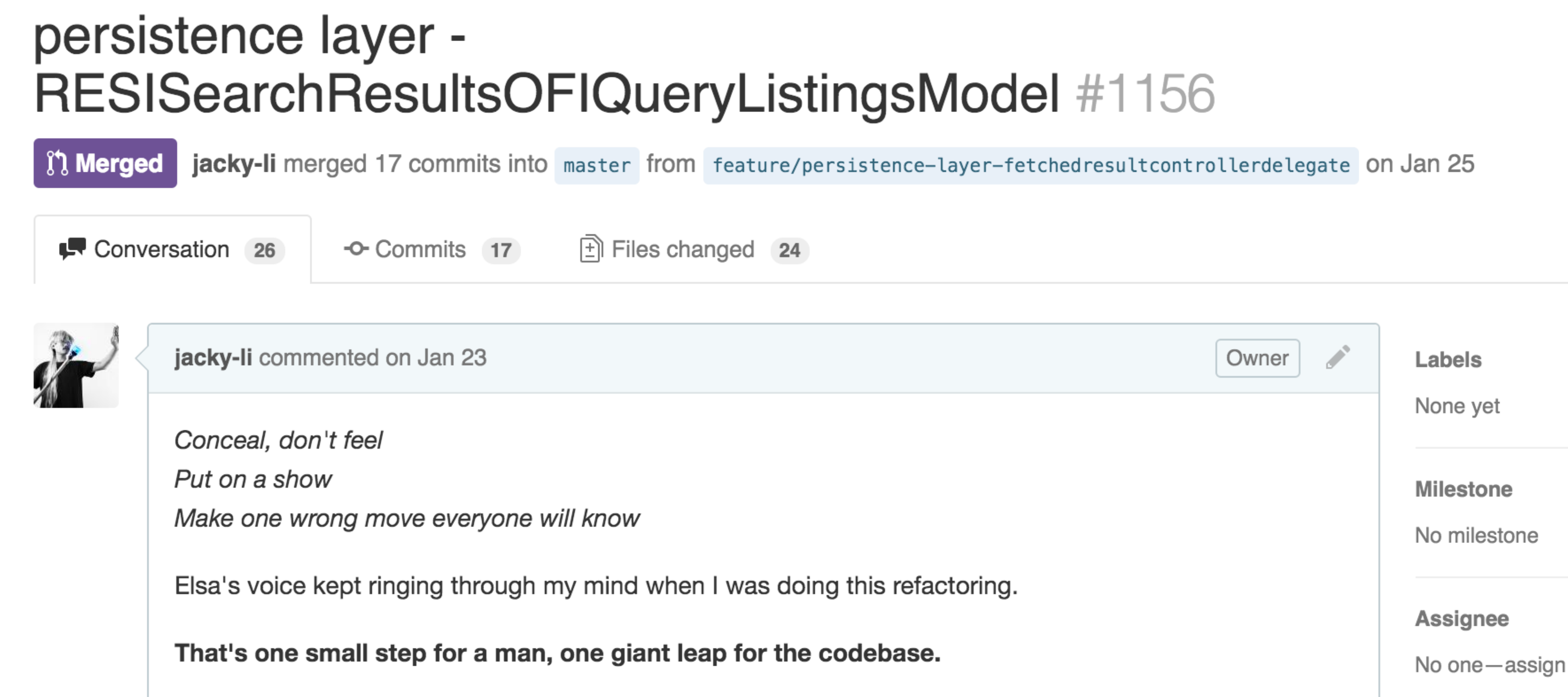
新的持久化服务

打开开关,内测

数据迁移

后续清理


ViewController
View
ViewModel
Model
Repository
Service



persistence layer - RESISearchResultsOFIQueryListingsModel #1156

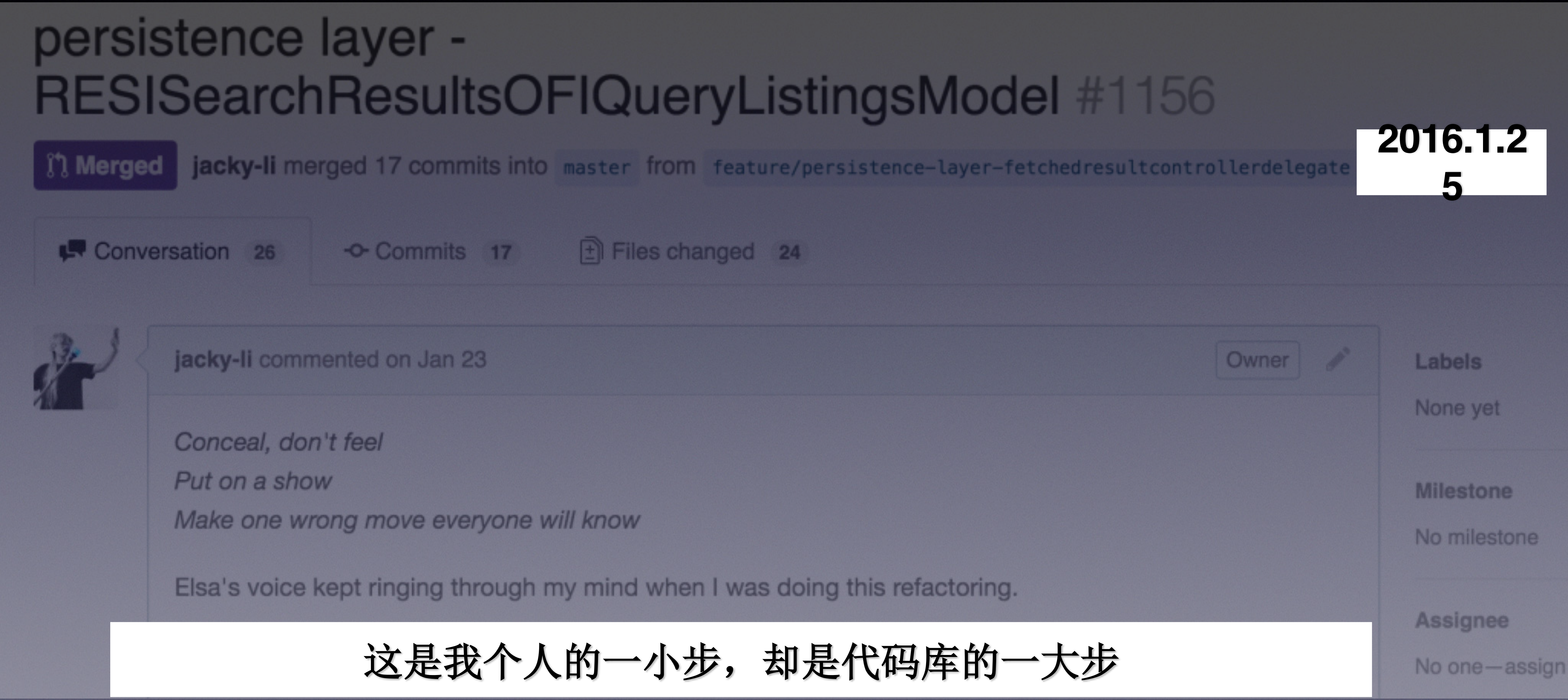
Merged jacky-li merged 17 commits into `master` from `feature/persistence-layer-fetchedresultcontrollerdelegate` on Jan 25

jacky-li commented on Jan 23

Conceal, don't feel
Put on a show
Make one wrong move everyone will know

Elsa's voice kept ringing through my mind when I was doing this refactoring.

That's one small step for a man, one giant leap for the codebase.



persistence layer - RESISearchResultsOFIQueryListingsModel #1156

jacky-li merged 17 commits into `master` from `feature/persistence-layer-fetchedresultcontrollerdelegate`

jacky-li commented on Jan 23

Conceal, don't feel
Put on a show
Make one wrong move everyone will know

Elsa's voice kept ringing through my mind when I was doing this refactoring.

Labels

None yet

Milestone

No milestone

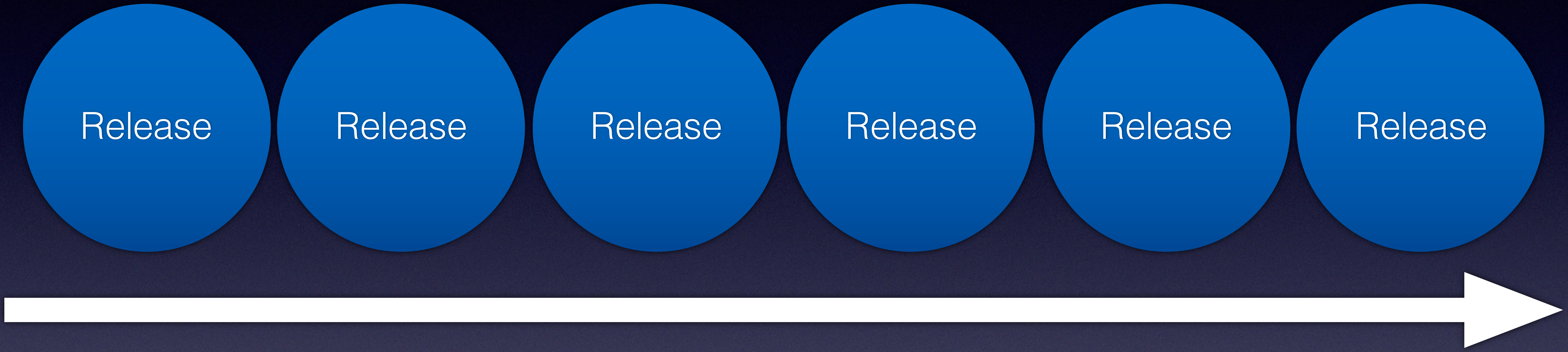
Assignee

No one—assign

2016.1.25

这是我个人的一小步，却是代码库的一大步

合并了100+ Pull Requests



Release

Release

Release

Release

Release

Release

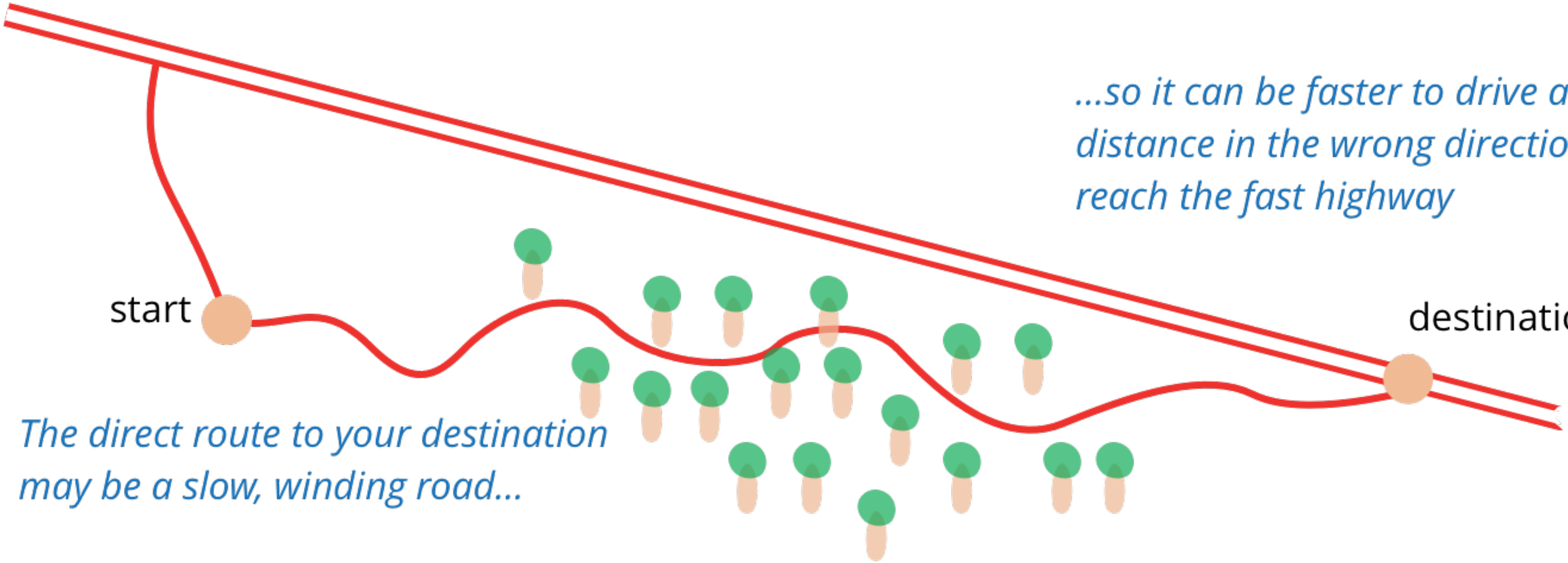
fears that once controlled me can’t get to me at all

AUTO
DIM

要勇敢

KYUSS
Parliament
adidas

Only
Losers
Drop
Litter
CANTERBURY
CITY COUNCIL



...so it can be faster to drive a short distance in the wrong direction to reach the fast highway

The direct route to your destination may be a slow, winding road...

http://martinfowler.com/articles/preparatory-refactoring-example/

也要谨慎

For the First Time in Forever
By Kristen Bell and Idina Menzel
www.youtube.com/OttoDadYT

make one wrong move everyone will know

正确的重构手法+大量的测试覆盖

抽象分支+特性开关

设计会过时，但是设计原则不会

CoreData never bother me anyway

Thank you!