App Frameworks

#WWDC15

What's New in Core Data

Session 220

Rishi Verma Core Data Engineer
Scott Perry Core Data Engineer

What Is Core Data?

To persist or not to persist

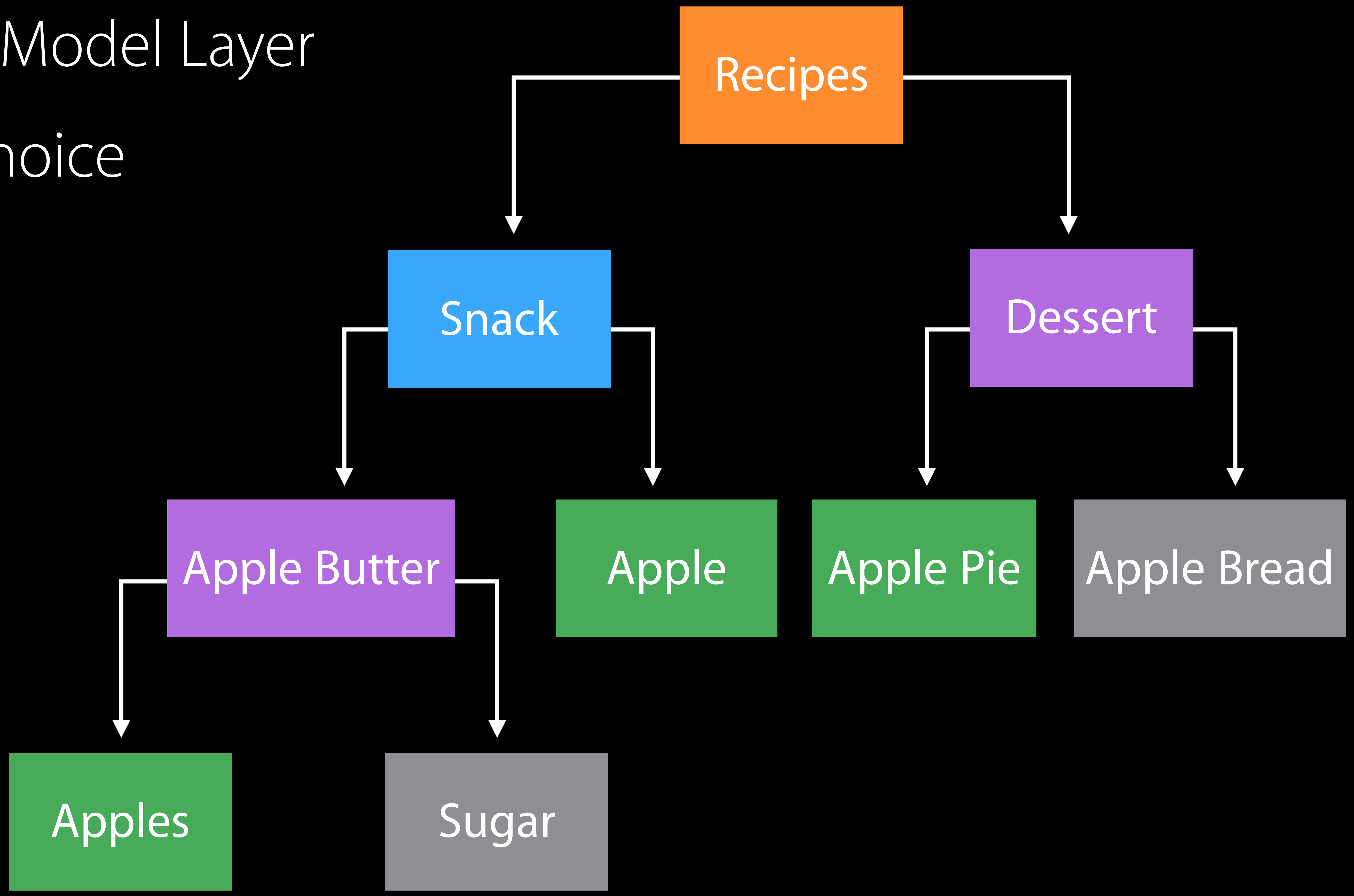
Rishi Verma Core Data Engineer

Object Graph Management

Manage my graph with Core Data

Bridge your data simply into a Cocoa Model Layer

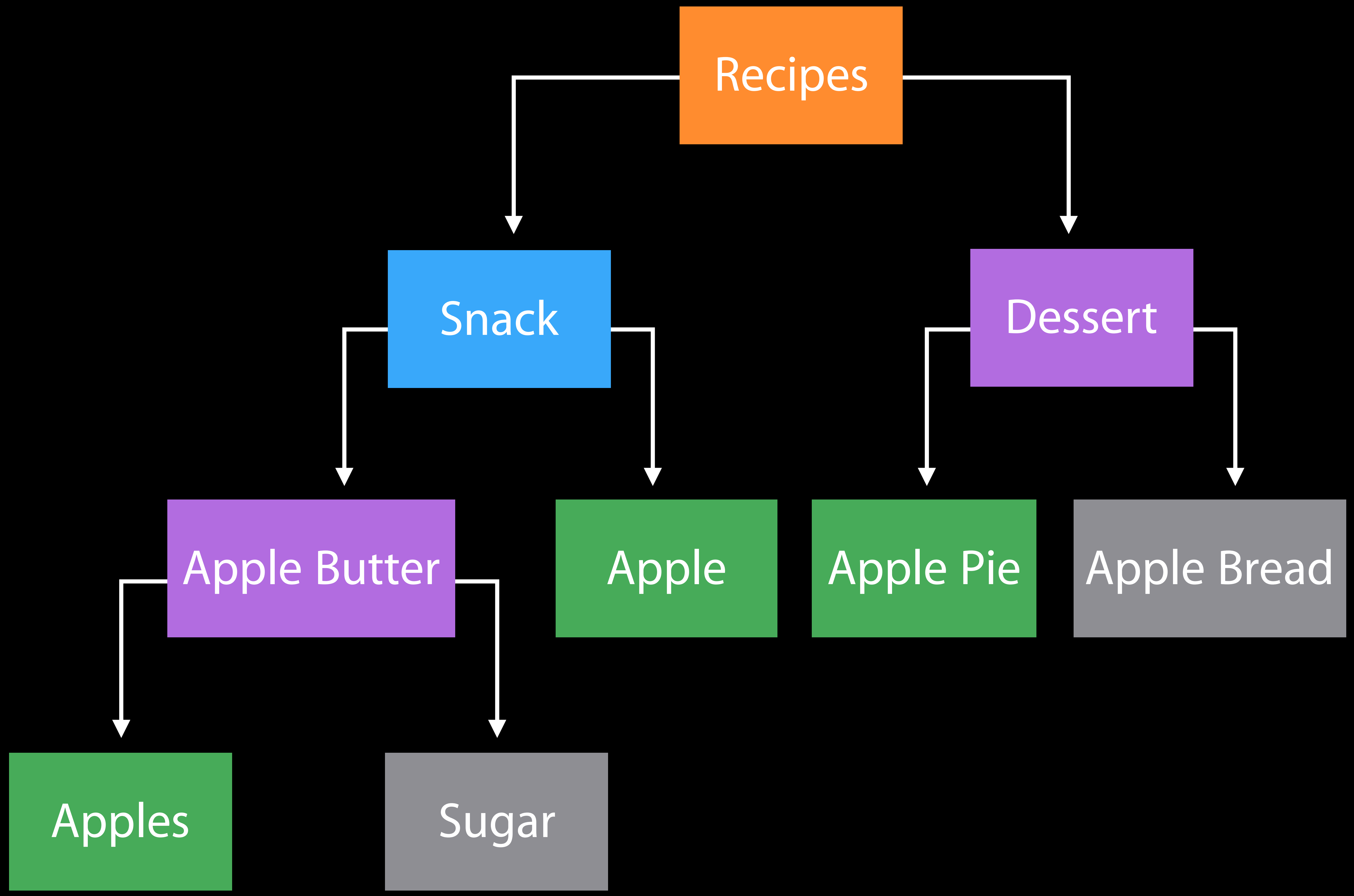
Persist data in the back end of your choice



Automatic Graph Management

Relationships can be complicated…

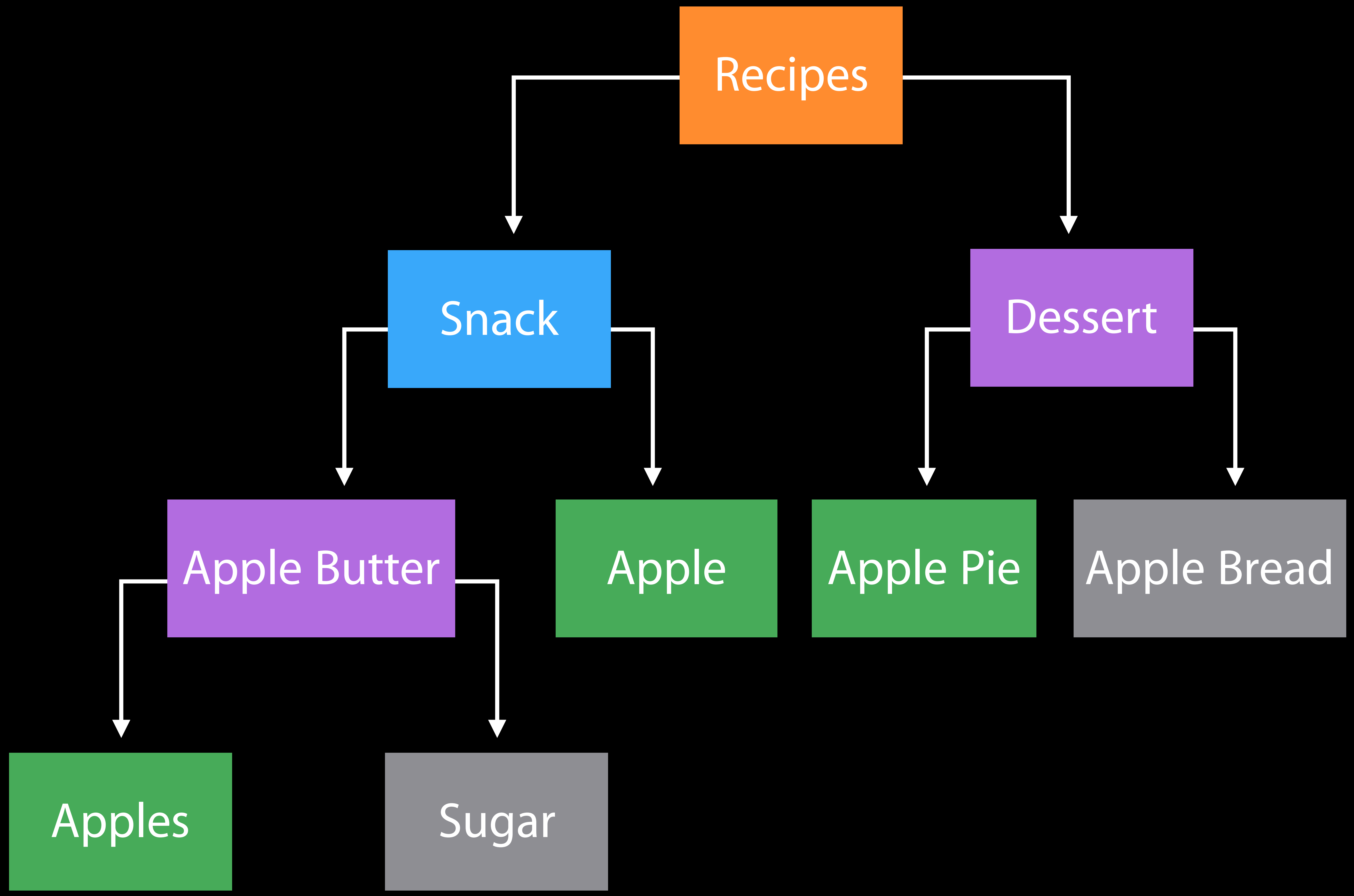
Automatic relationship maintenance



NSFetchRequest

Finding a needle in a haystack

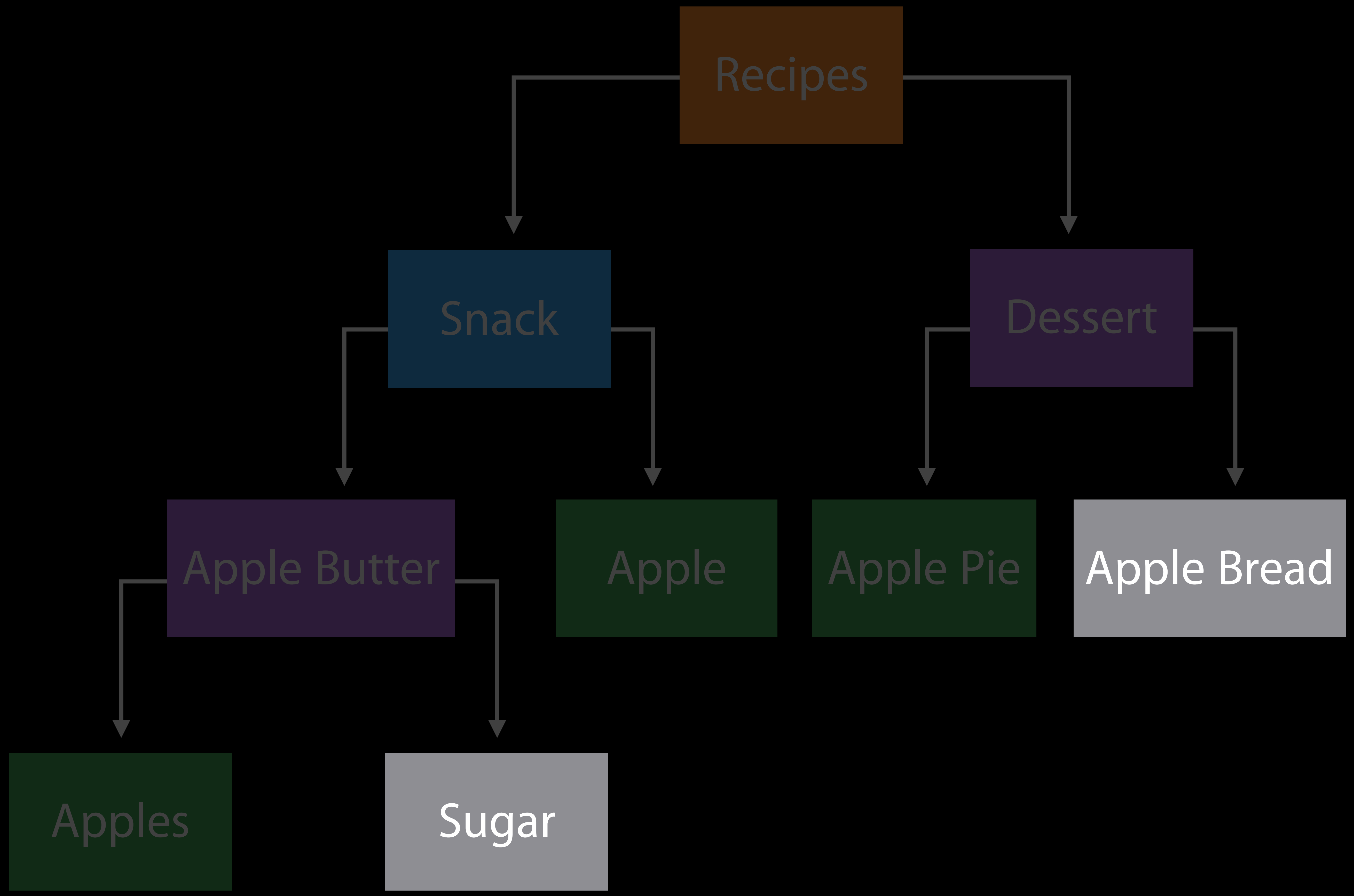
Find the data you need



NSFetchRequest

Finding a needle in a haystack

Find the data you need

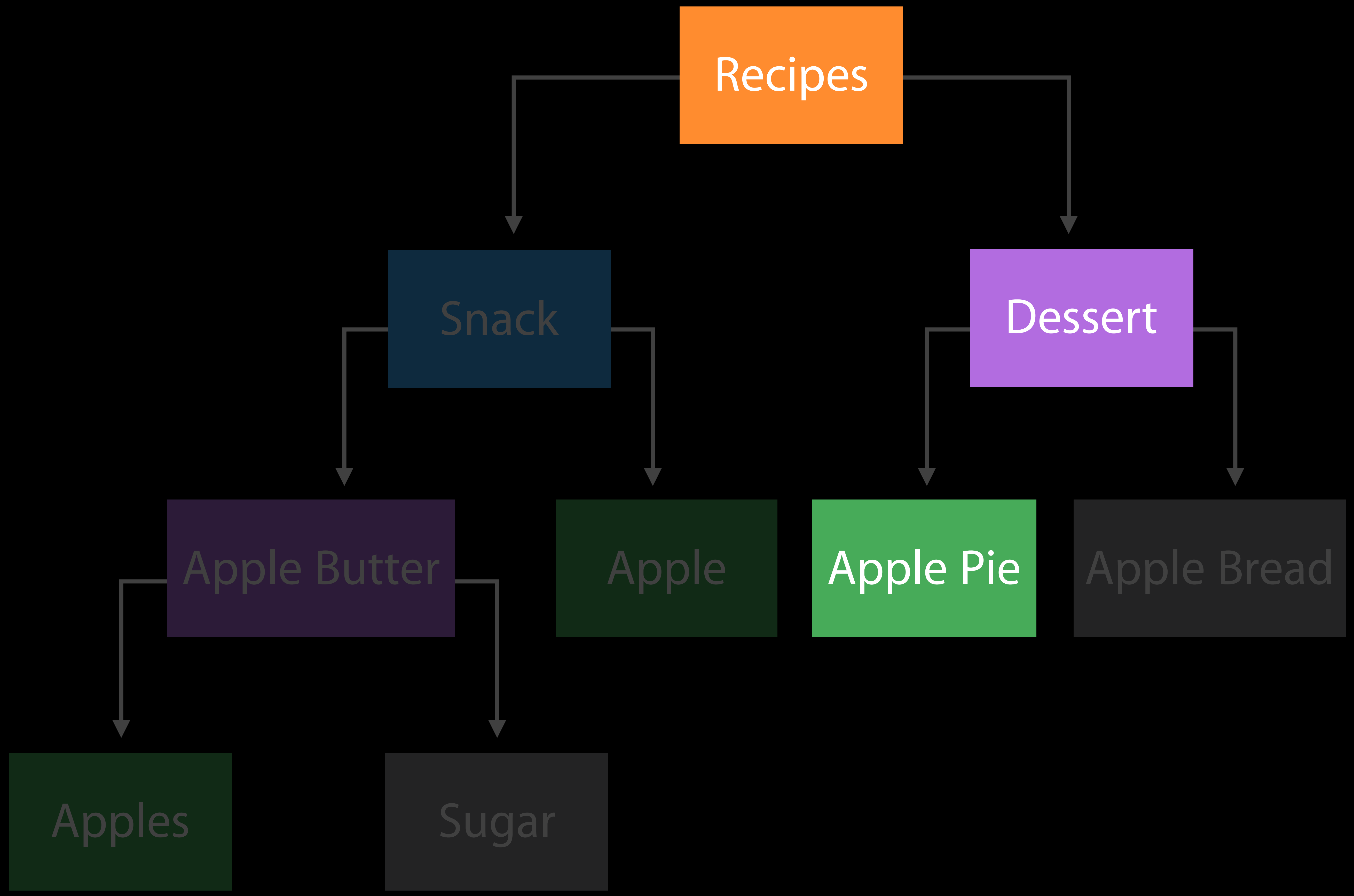


NSFetchRequest

Finding a needle in a haystack

Find the data you need

Batching



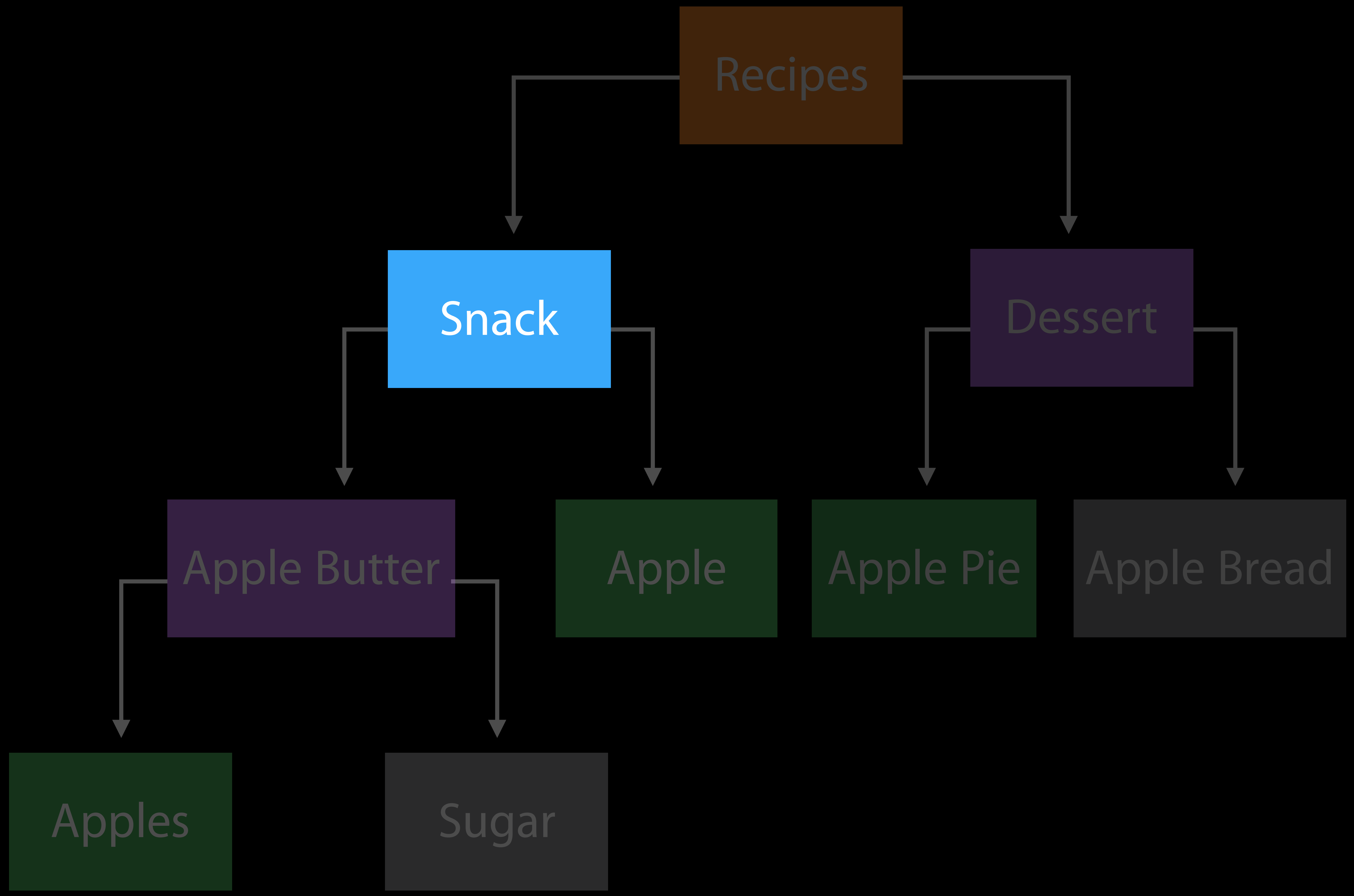
NSFetchRequest

Finding a needle in a haystack

Find the data you need

Batching

Relationship prefetching



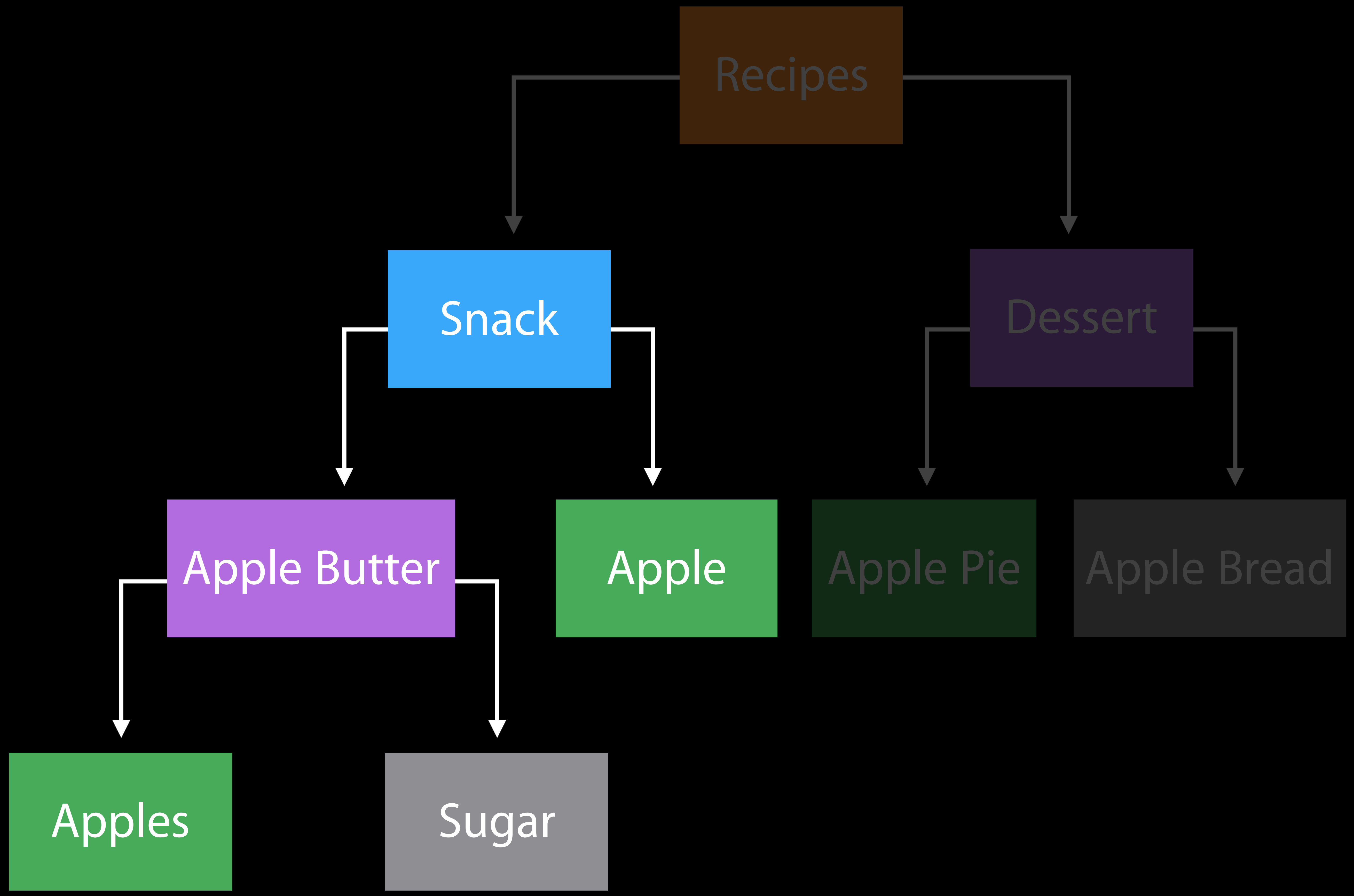
NSFetchRequest

Finding a needle in a haystack

Find the data you need

Batching

Relationship prefetching



NSFetchRequest

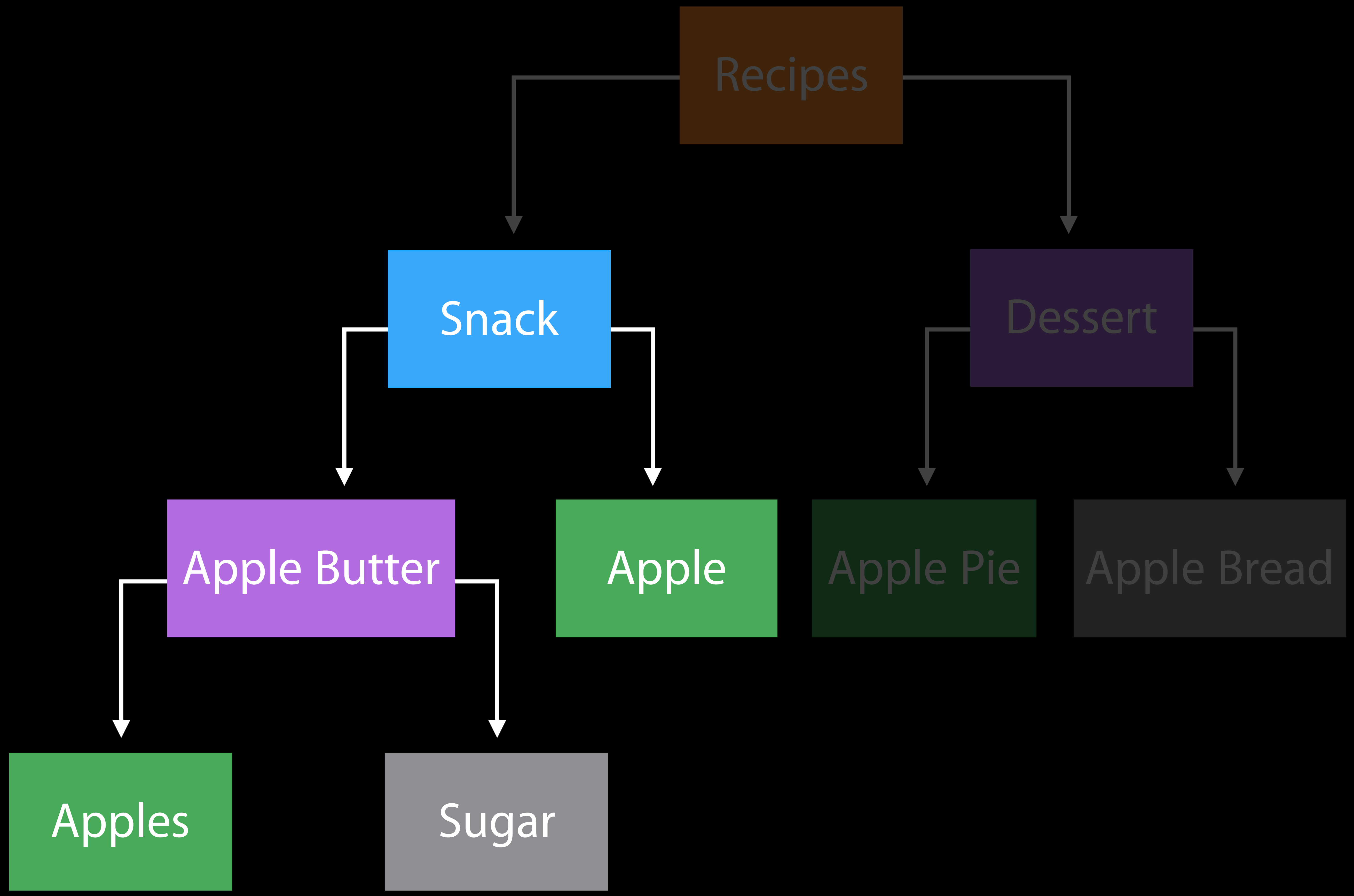
Finding a needle in a haystack

Find the data you need

Batching

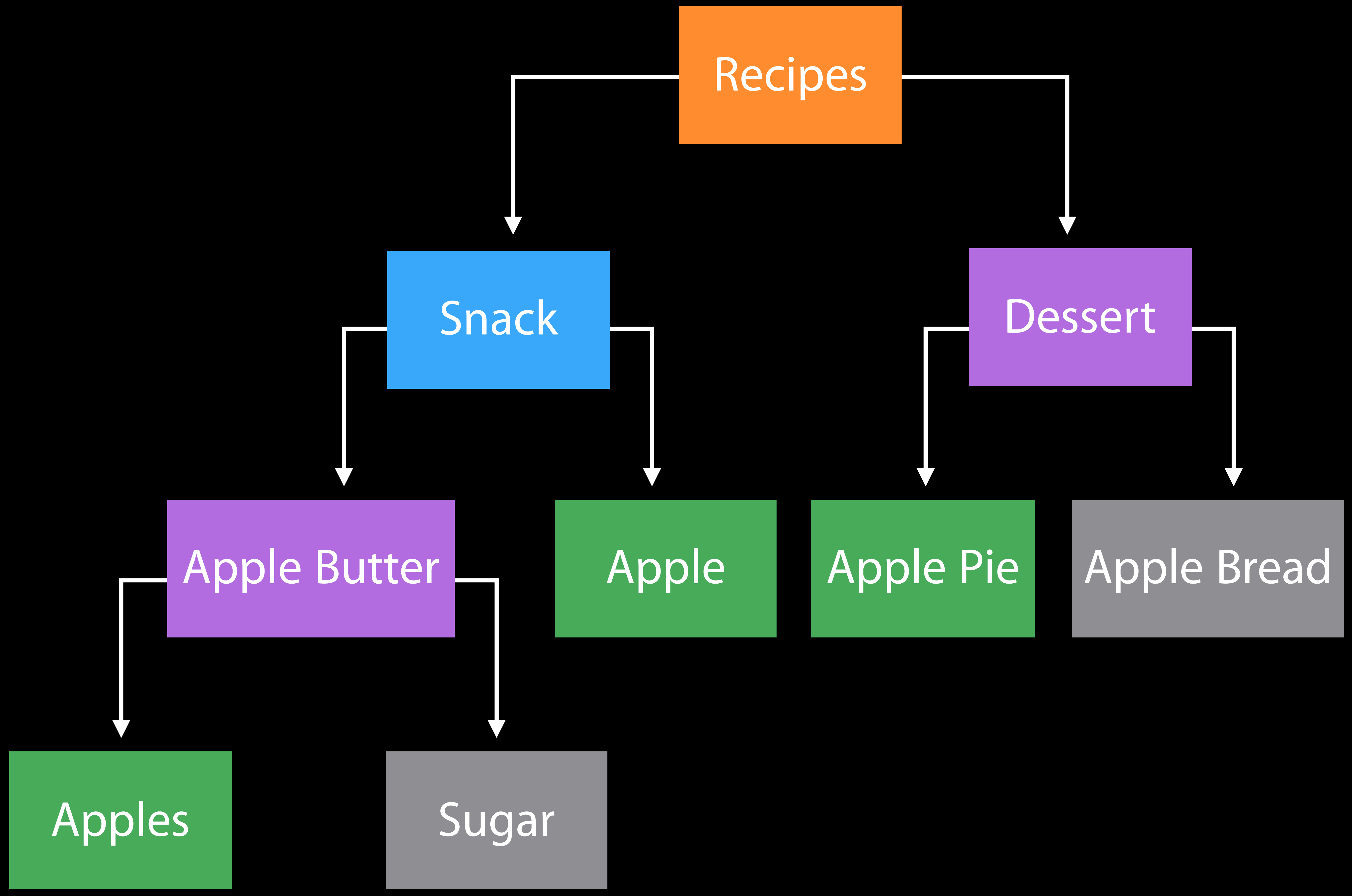
Relationship prefetching

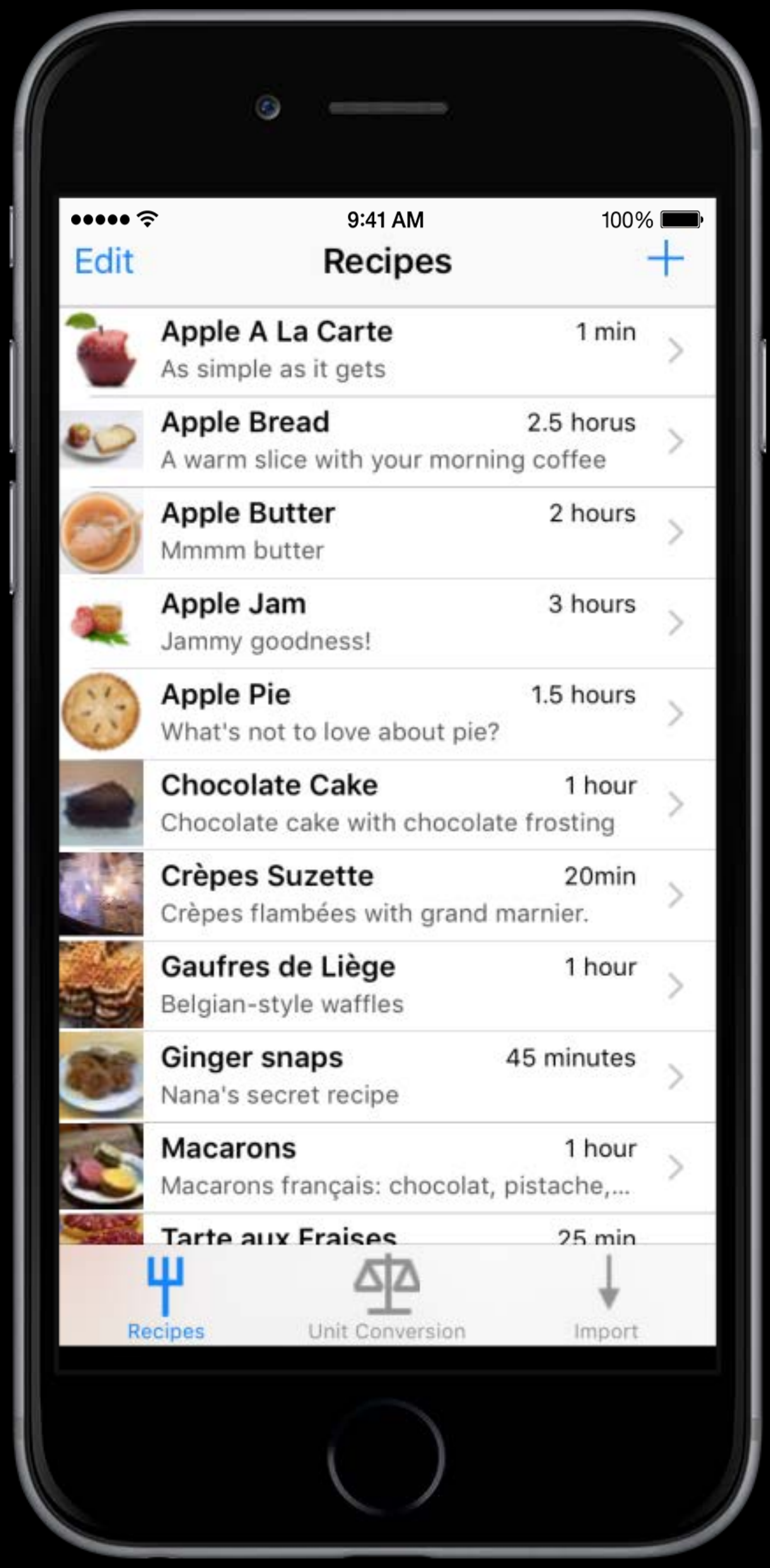
Then tie this with your UI and…



View and Controller Support

My UI brings all the updates to the users

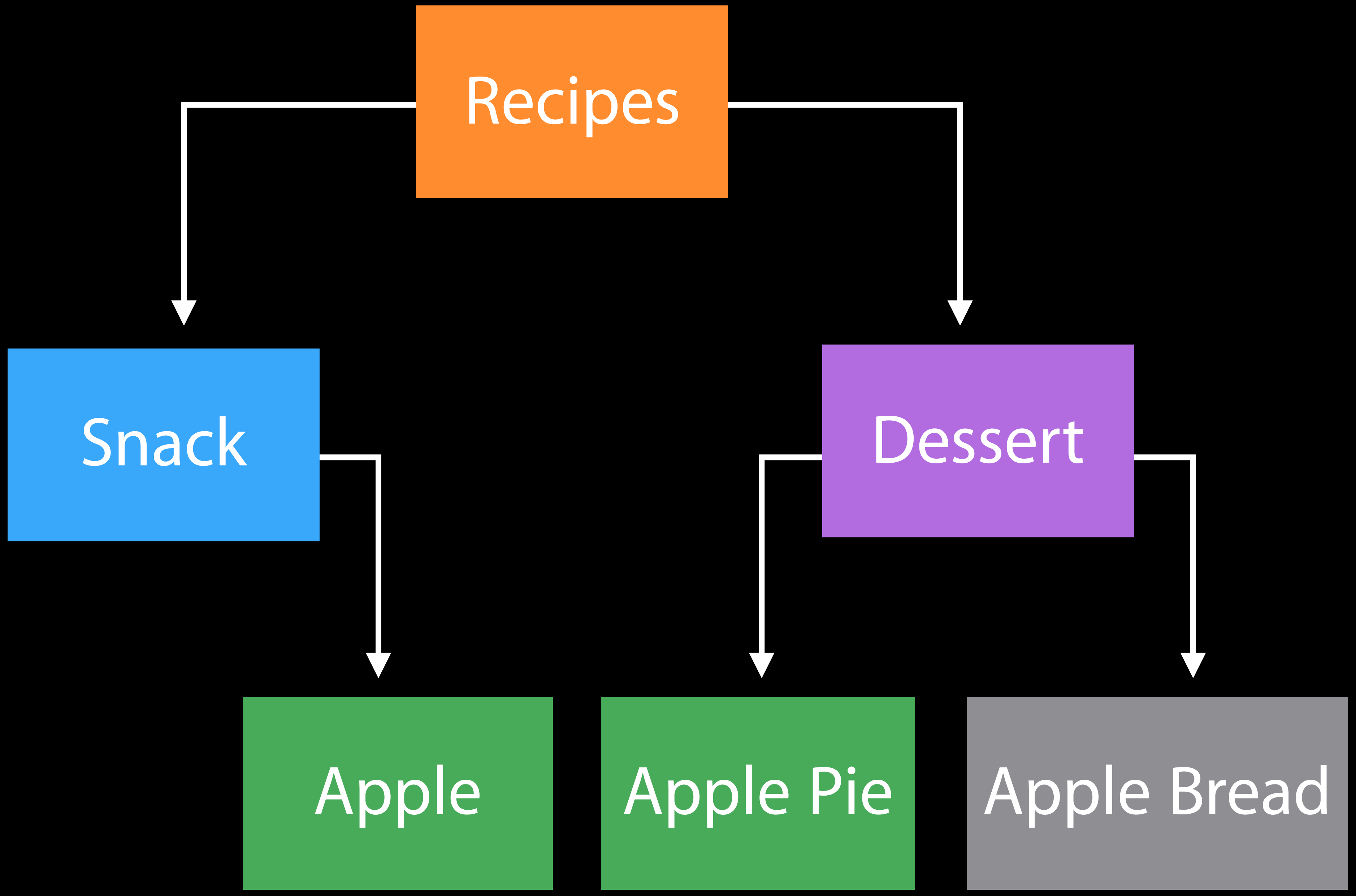


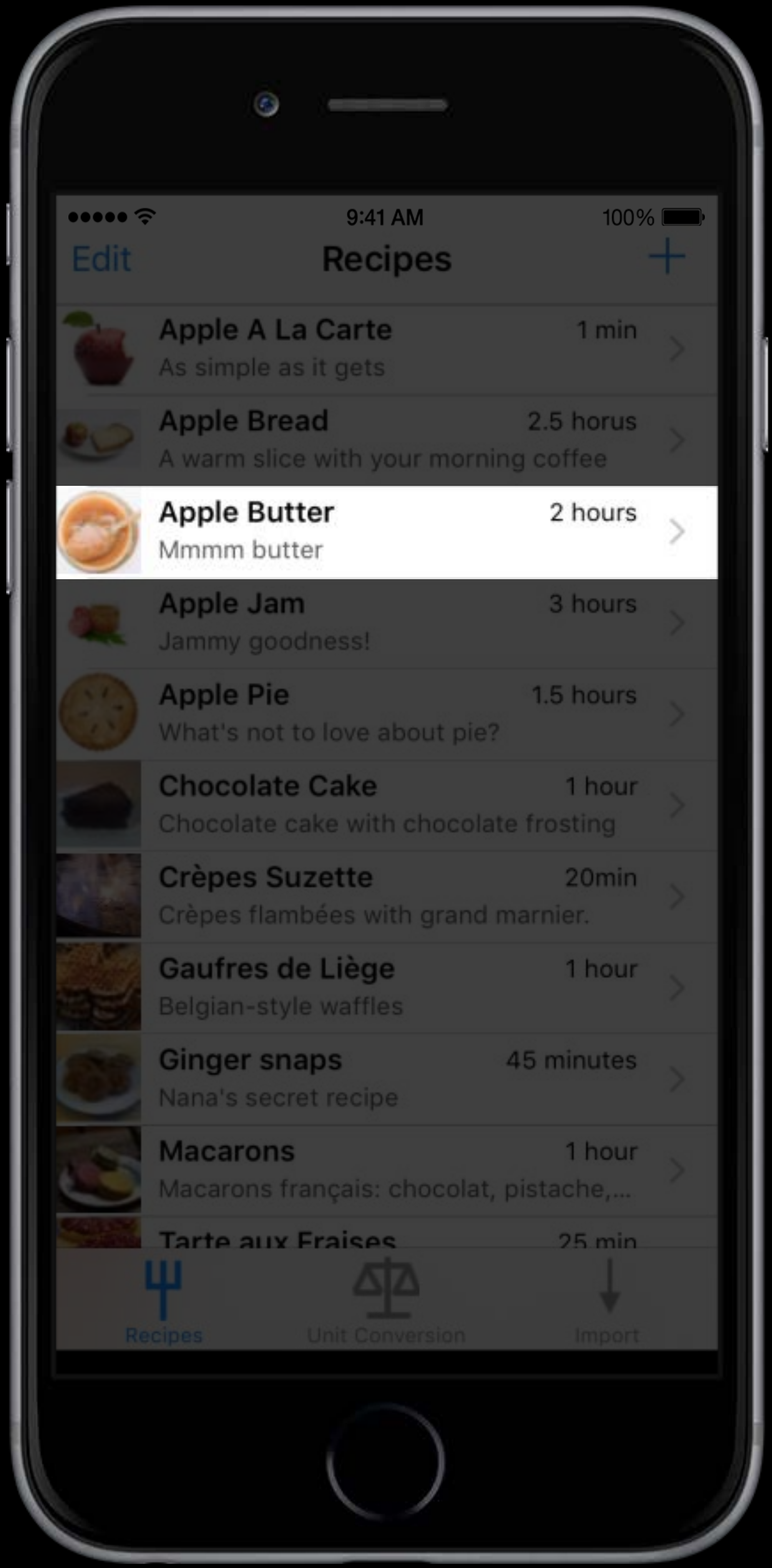


Apple A La Carte As simple as it gets	1 min
Apple Bread A warm slice with your morning coffee	2.5 horus
Apple Butter Mmmm butter	2 hours
Apple Jam Jammy goodness!	3 hours
Apple Pie What's not to love about pie?	1.5 hours
Chocolate Cake Chocolate cake with chocolate frosting	1 hour
Crèpes Suzette Crèpes flambées with grand marnier.	20min
Gaufres de Liège Belgian-style waffles	1 hour
Ginger snaps Nana's secret recipe	45 minutes
Macarons Macarons français: chocolat, pistache,...	1 hour
Tarte aux Fraises	25 min

View and Controller Support

My UI brings all the updates to the users





Apple A La Carte 1 min
As simple as it gets

Apple Bread 2.5 horus
A warm slice with your morning coffee

Apple Butter 2 hours
Mmmm butter

Apple Jam 3 hours
Jammy goodness!

Apple Pie 1.5 hours
What's not to love about pie?

Chocolate Cake 1 hour
Chocolate cake with chocolate frosting

Crèpes Suzette 20min
Crèpes flambées with grand marnier.

Gaufres de Liège 1 hour
Belgian-style waffles

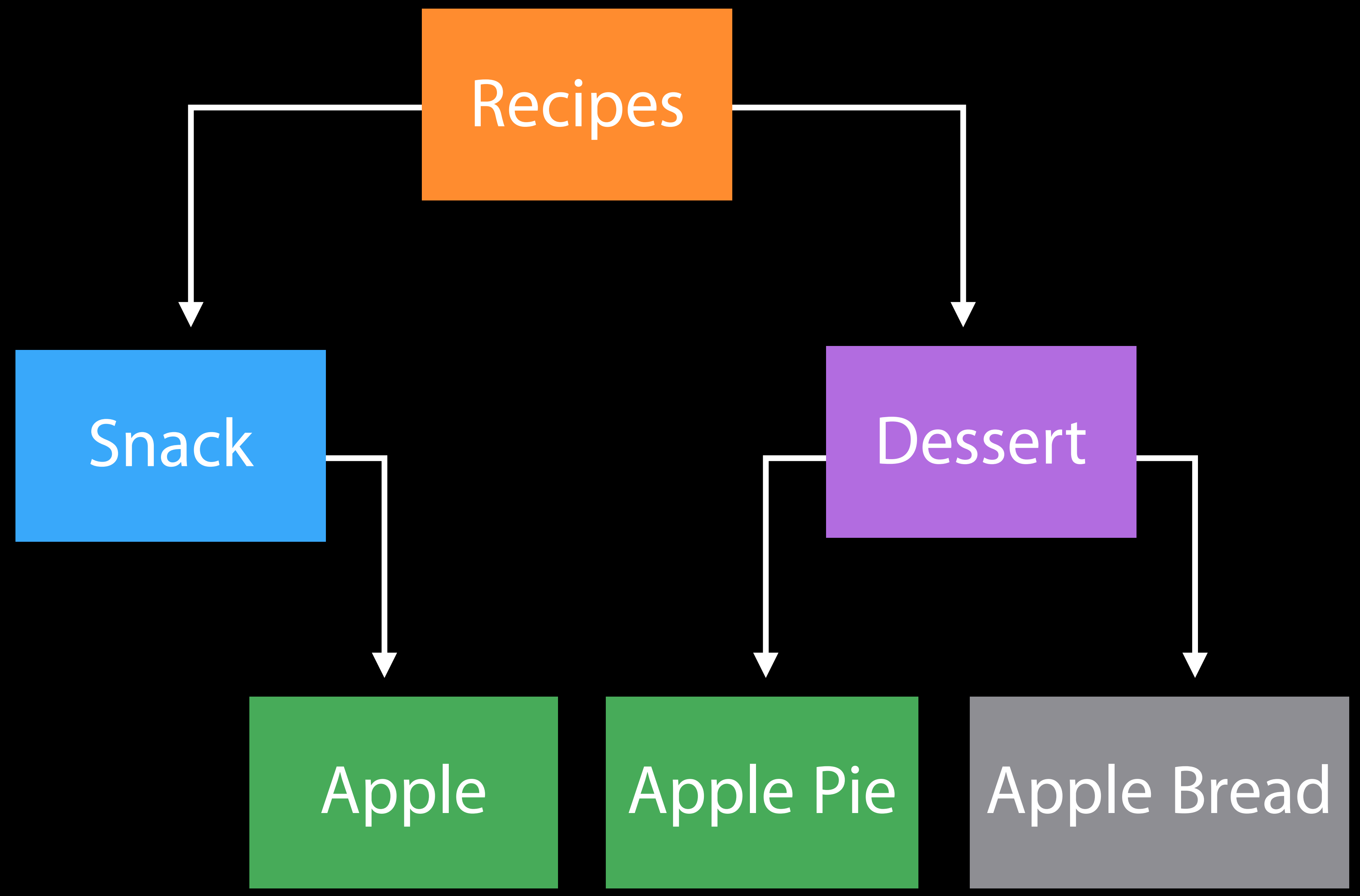
Ginger snaps 45 minutes
Nana's secret recipe

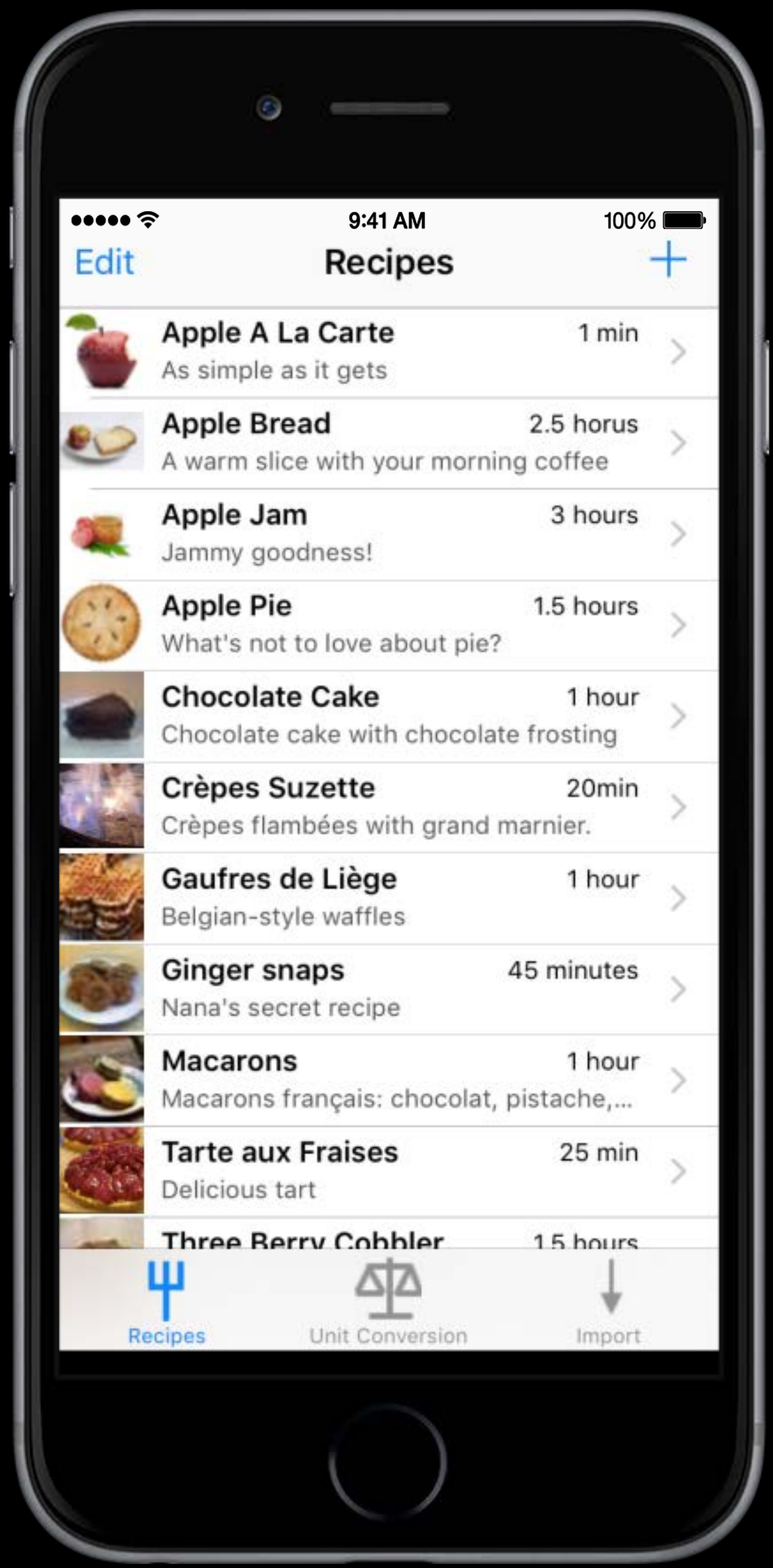
Macarons 1 hour
Macarons français: chocolat, pistache,...

Tarte aux Fraises 25 min

View and Controller Support

My UI brings all the updates to the users





Apple A La Carte 1 min
As simple as it gets

Apple Bread 2.5 horus
A warm slice with your morning coffee

Apple Jam 3 hours
Jammy goodness!

Apple Pie 1.5 hours
What's not to love about pie?

Chocolate Cake 1 hour
Chocolate cake with chocolate frosting

Crèpes Suzette 20min
Crèpes flambées with grand marnier.

Gaufres de Liège 1 hour
Belgian-style waffles

Ginger snaps 45 minutes
Nana's secret recipe

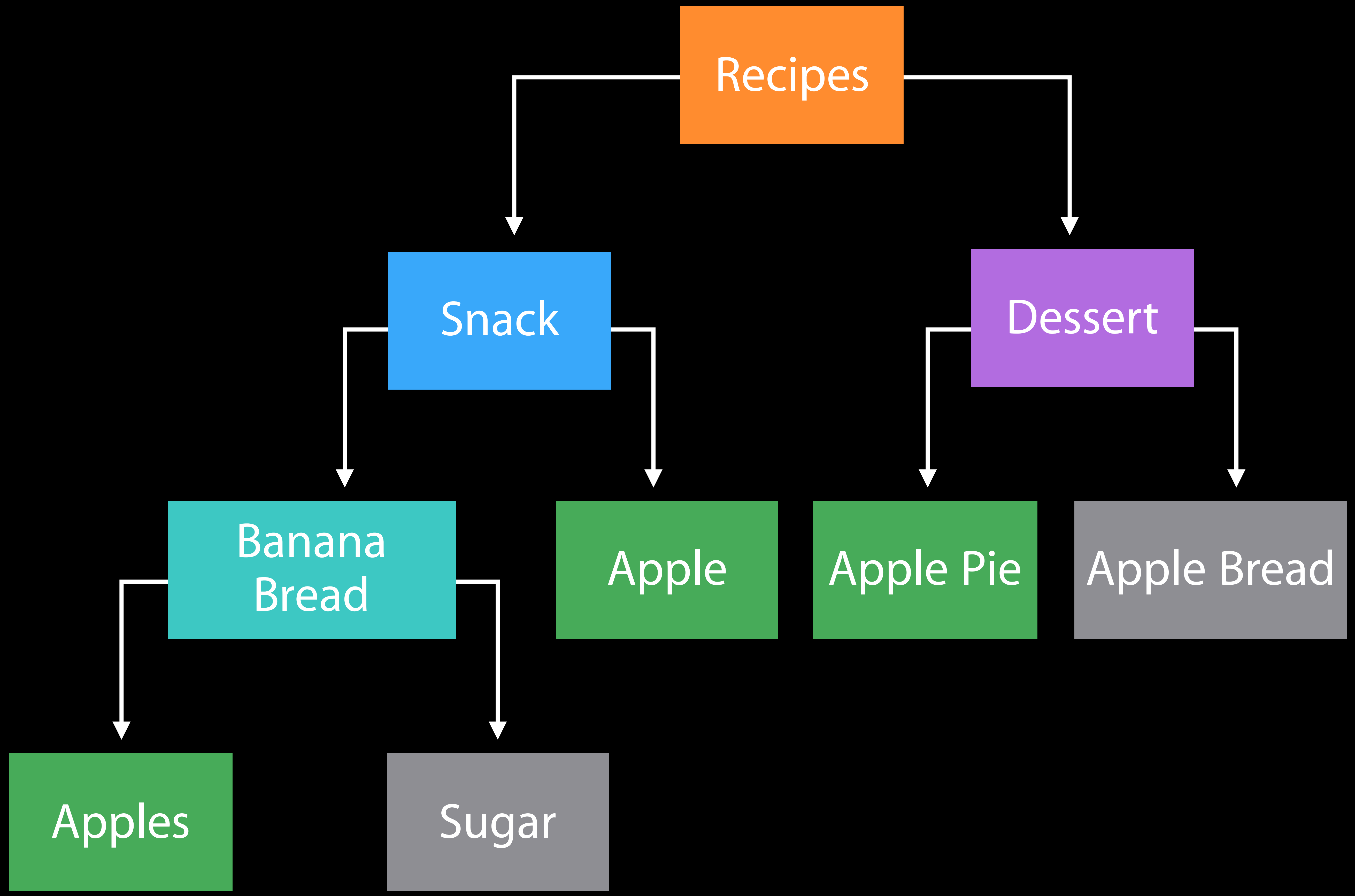
Macarons 1 hour
Macarons français: chocolat, pistache,...

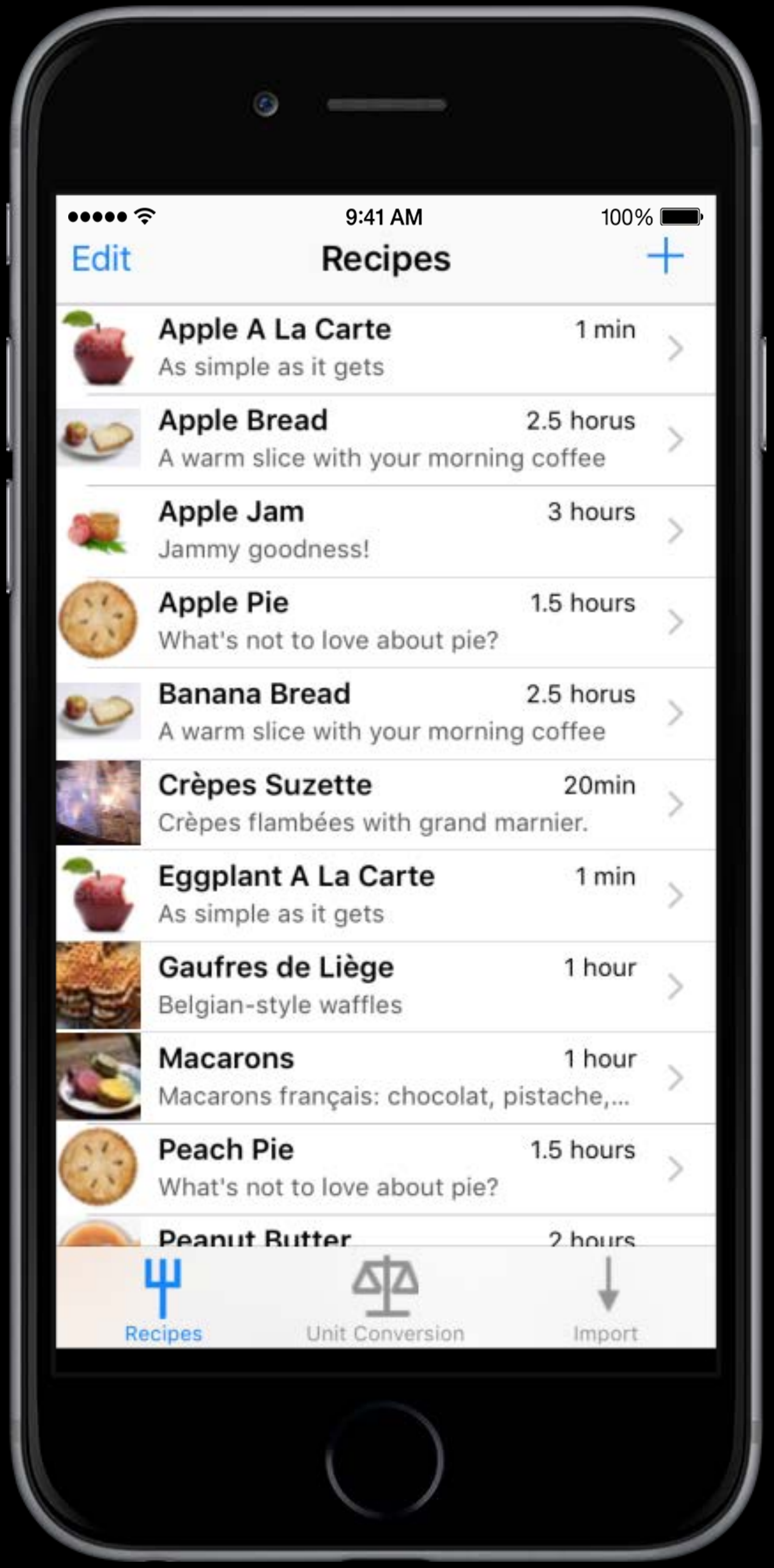
Tarte aux Fraises 25 min
Delicious tart

Three Berry Cobbler 1.5 hours

View and Controller Support

My UI brings all the updates to the users





Apple A La Carte 1 min
As simple as it gets

Apple Bread 2.5 horus
A warm slice with your morning coffee

Apple Jam 3 hours
Jammy goodness!

Apple Pie 1.5 hours
What's not to love about pie?

Banana Bread 2.5 horus
A warm slice with your morning coffee

Crèpes Suzette 20min
Crèpes flambées with grand marnier.

Eggplant A La Carte 1 min
As simple as it gets

Gaufres de Liège 1 hour
Belgian-style waffles

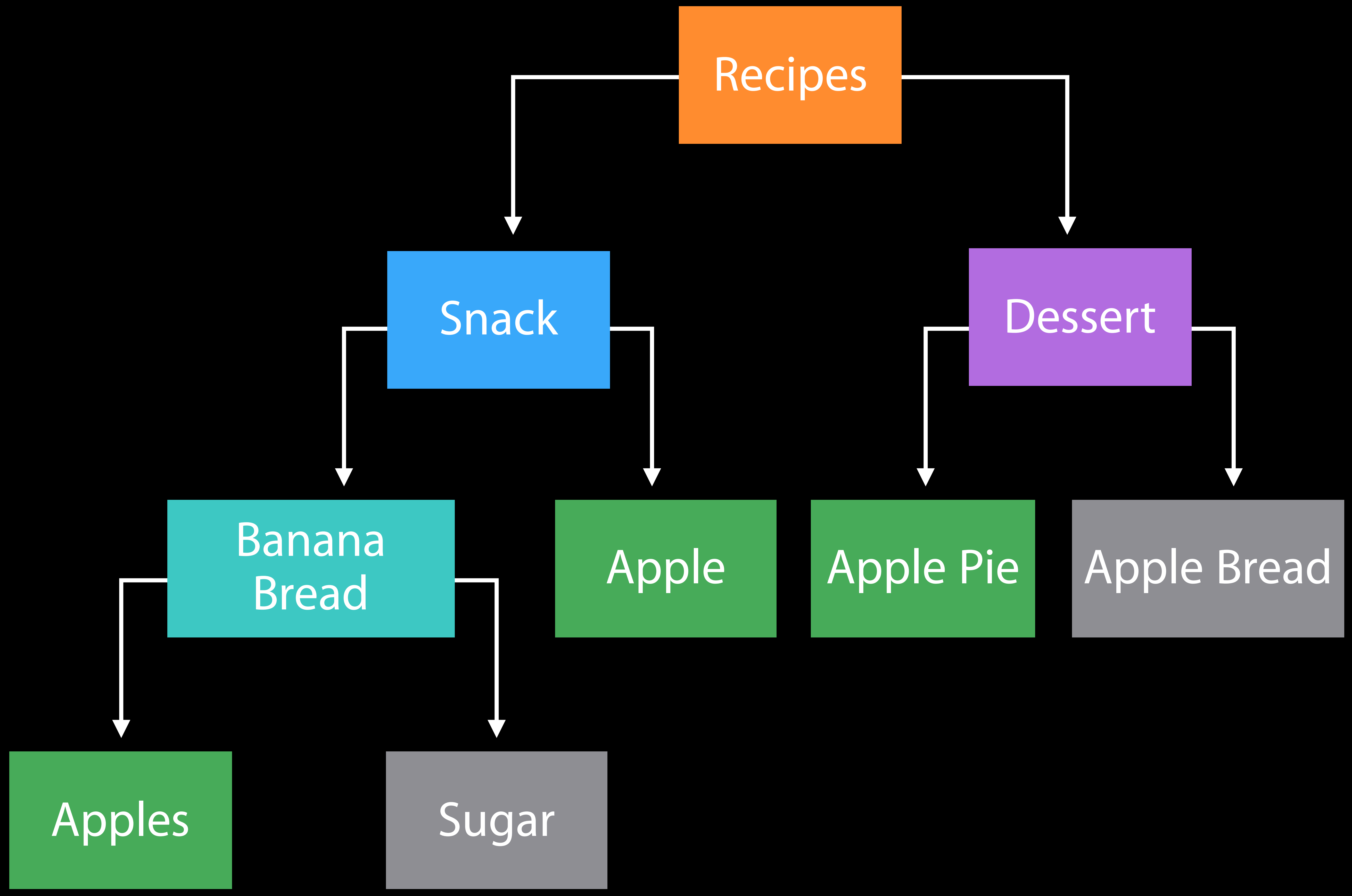
Macarons 1 hour
Macarons français: chocolat, pistache,...

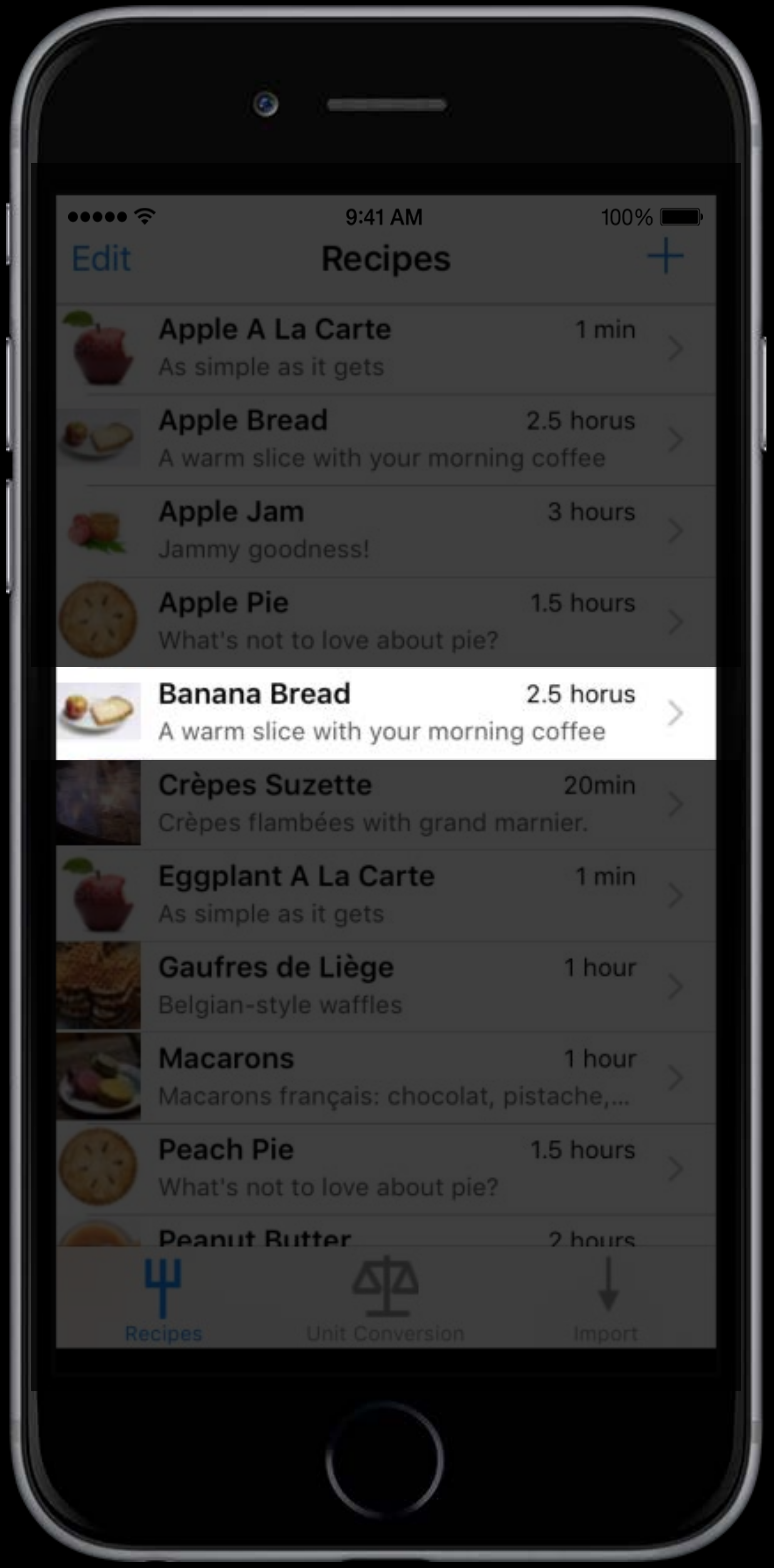
Peach Pie 1.5 hours
What's not to love about pie?

Peanut Butter 2 hours

View and Controller Support

My UI brings all the updates to the users





Apple A La Carte	1 min
As simple as it gets	
Apple Bread	2.5 horus
A warm slice with your morning coffee	
Apple Jam	3 hours
Jammy goodness!	
Apple Pie	1.5 hours
What's not to love about pie?	
Banana Bread	2.5 horus
A warm slice with your morning coffee	
Crèpes Suzette	20min
Crèpes flambées with grand marnier.	
Eggplant A La Carte	1 min
As simple as it gets	
Gaufres de Liège	1 hour
Belgian-style waffles	
Macarons	1 hour
Macarons français: chocolat, pistache,...	
Peach Pie	1.5 hours
What's not to love about pie?	
Peanut Butter	2 hours

Multi-Writer Conflict Handling

On your mark, set that merge policy, and done

CoreData versions all objects

Several types of merge policies available

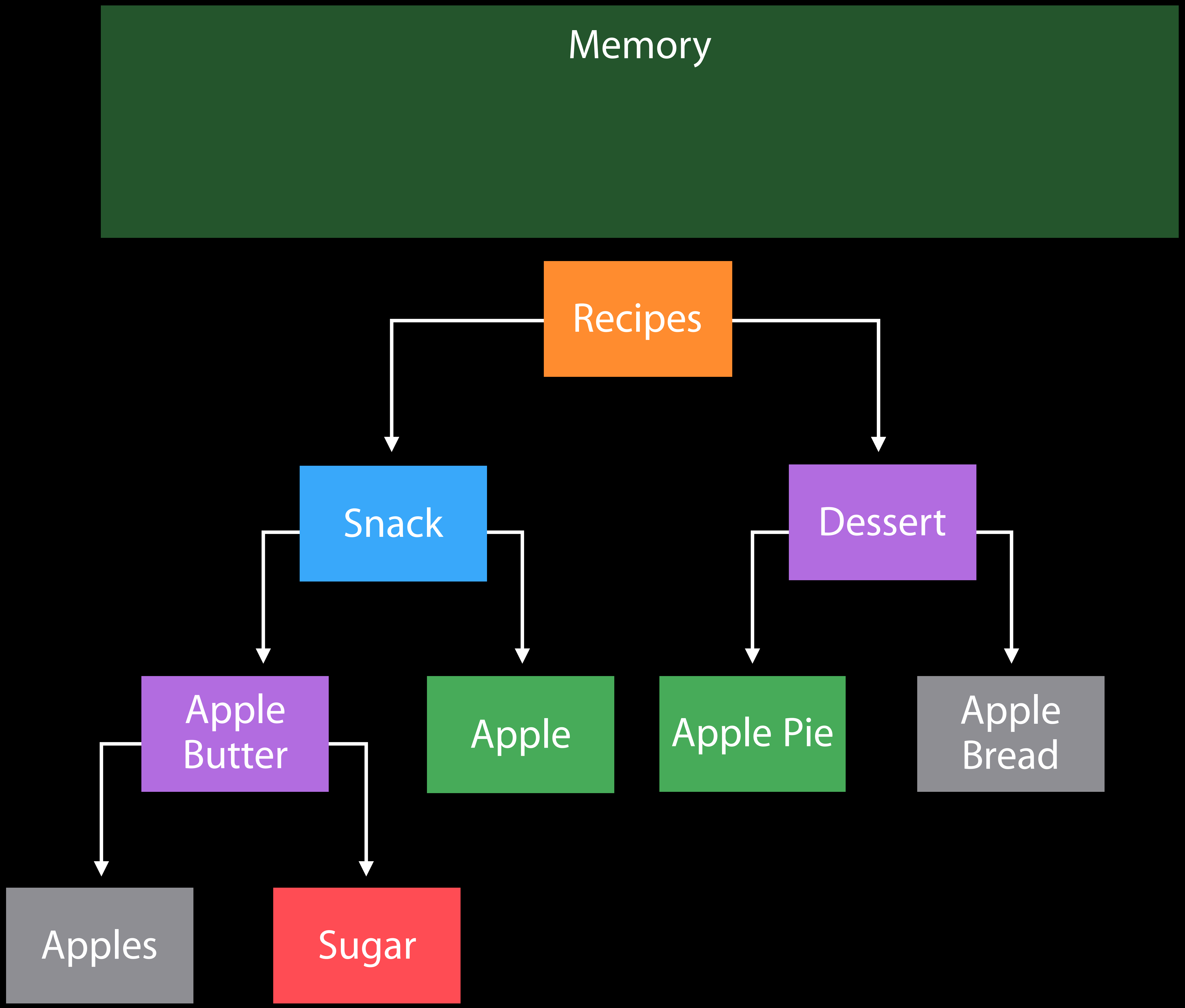
- Defaults to error
- Persistent store vs. in-memory

Memory Efficiencies

APIs with benefits

Excellent memory scalability

Aggressive lazy loading

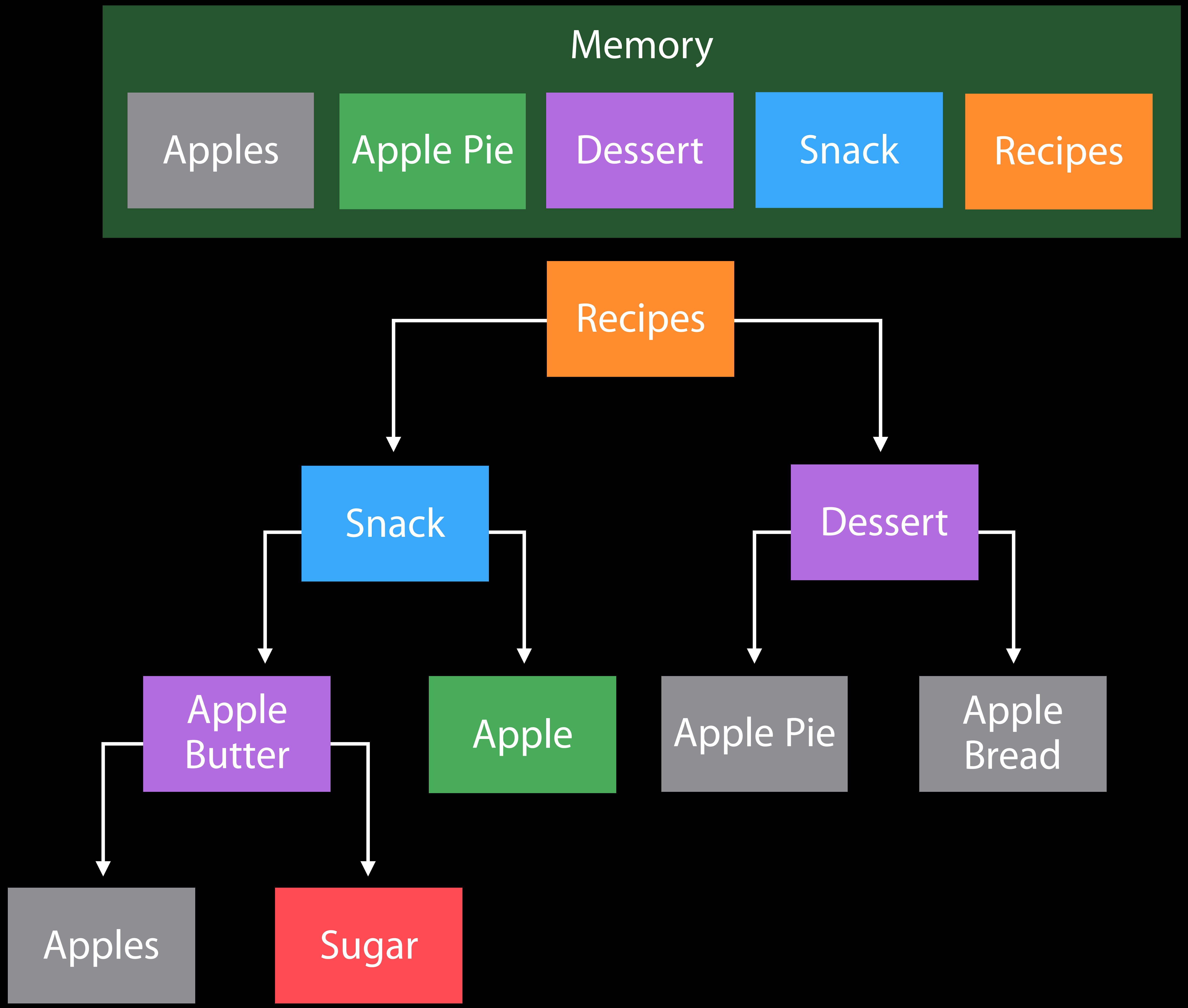


Memory Efficiencies

APIs with benefits

Excellent memory scalability

Aggressive lazy loading

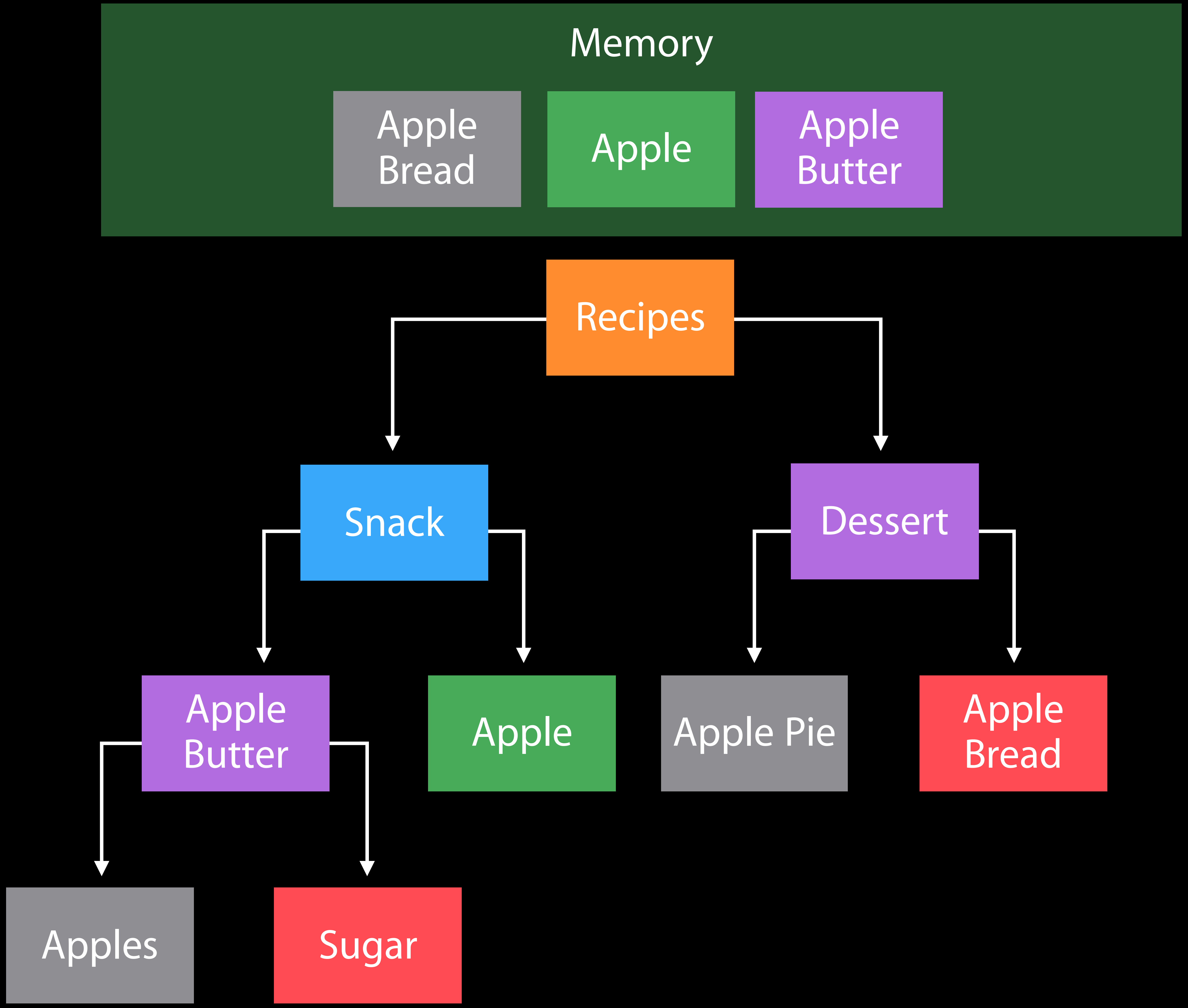


Memory Efficiencies

APIs with benefits

Excellent memory scalability

Aggressive lazy loading



Smaller Footprint

Less is more

50%–70%

Less code

400,000

API Enhancements

hasPersistentChangedValues

NEW

NSManagedObject

```
var hasPersistentChangedValues: Bool { get }
```

No false positives setting a value to itself

Skips transient properties

objectIDsForRelationshipNamed

NEW

NSManagedObject

```
func objectIDsForRelationshipNamed(key: String) -> [NSManagedObjectID]
```

Reads cache or fetches the objectIDs

Doesn’t materialize entire relationship

Useful working with large, many-to-many relationships

objectIDsForRelationshipNamed

NEW

Code example

objectIDsForRelationshipNamed

NEW

Code example

```
let relations = person.objectIDsForRelationshipNamed("family")
```

objectIDsForRelationshipNamed

NEW

Code example

```
let relations = person.objectIDsForRelationshipNamed("family")


let fetchFamily = NSFetchRequest(entityName: “Person")
fetchFamily.fetchBatchSize = 100
fetchFamily.predicate = NSPredicate(format: "self IN %@", relations)
```

objectIDsForRelationshipNamed

NEW

Code example

```
let relations = person.objectIDsForRelationshipNamed("family")


let fetchFamily = NSFetchRequest(entityName: “Person")
fetchFamily.fetchBatchSize = 100
fetchFamily.predicate = NSPredicate(format: "self IN %@", relations)

let batchedRelations = managedObjectContext.executeFetchRequest(fetchFamily)
```

objectIDsForRelationshipNamed

NEW

Code example

```
let relations = person.objectIDsForRelationshipNamed("family")


let fetchFamily = NSFetchRequest(entityName: “Person")
fetchFamily.fetchBatchSize = 100
fetchFamily.predicate = NSPredicate(format: "self IN %@", relations)

let batchedRelations = managedObjectContext.executeFetchRequest(fetchFamily)

for relative in batchedRelations {
   // work with relations 100 rows at a time
}
```

refreshAllObjects

NEW

NSManagedObjectContext

```
func refreshAllObjects()
```

Affects all registered objects in a context

Preserves unsaved changes

Managed Object references remain valid

Useful for breaking retain cycles

mergeChangesFromRemoteContextSave

NEW

NSManagedObjectContext

```
class func mergeChangesFromRemoteContextSave(changeNotificationData:
[NSObject : AnyObject], intoContexts contexts: [NSManagedObjectContext])
```

Better for changes from different coordinators

Fetches latest row data

Handles ordering with nested contexts

No Love for Exceptions

This is not the data you are looking for

Why is Core Data unable to fulfill a fault?

Managed objects are implicit futures

- Cocoa place holders for a row of data
- Often lazily loaded
- Part of a larger graph

Data deleted out from underneath this reference

shouldDeleteInaccessibleFaults

NSManagedObjectContext

```
var shouldDeleteInaccessibleFaults: Bool
```

- Defaults to YES
- Does not effect APIs with error parameters

Bad faults marked deleted

Missing data treated as NULL/nil/0

NSPersistentStoreCoordinator API

It’s my file and I’ll do what I want to

Truncating and copying databases

Don’t bypass the API layers

- NSFileManager and POSIX are bad for databases
- Will corrupt your files if open connections exist

Deleting a file with open locks ends badly…very badly

destroyPersistentStoreAtURL

NSPersistentStoreCoordinator

NEW

```
func destroyPersistentStoreAtURL(url: NSURL, withType storeType: String,
options: [NSObject : AnyObject]?) throws
```

Honors locking protocols

Handles details reconfiguring emptied files

- Journal mode, page size, etc.
- Need to pass same options as addToPersistentStore
- Accidentally switching journal modes can deadlock

replacePersistentStoreAtURL

NEW

NSPersistentStoreCoordinator

```
func replacePersistentStoreAtURL(destinationURL: NSURL, destinationOptions:
[NSObject : AnyObject]?, withPersistentStoreFromURL sourceURL: NSURL,
sourceOptions: [NSObject : AnyObject]?, storeType: String) throws
```

Same pattern as destroyPersistentStoreAtURL

If destination doesn’t exist, this does a copy

Unique Constraints

I got 99 problems and they are all duplicates…

Find or Create Pattern

Unique constraints

```
managedObjectContext.performBlock {
  let createRequest = NSFetchRequest(entityName: "Recipe")
  createRequest.resultType = ManagedObjectIDResultType
  let predicate =
    NSPredicate(format: "source = %@ AND externalID = %@", source,externalID)

  let results = self.managedObjectContext.executeFetchRequest(createRequest)
  if (results.count) {
      //update it!
  } else {
      //create it!
  }
}
```

One of a Kind

Unique constraints

Unique attributes across all instances of an entity

- Email addresses
- Part numbers
- UPC
- ISBN
- Unique key/value pairs

Best Practices

Unique constraints

Best for values unmodified after object creation

Sub-entities may extend constraints

- Parent (UUID)
- Sub-entity (UUID, EMAIL)

Recovery uses merge policies

Demo

How to utilize unique constraints

Deleting Multiple Objects

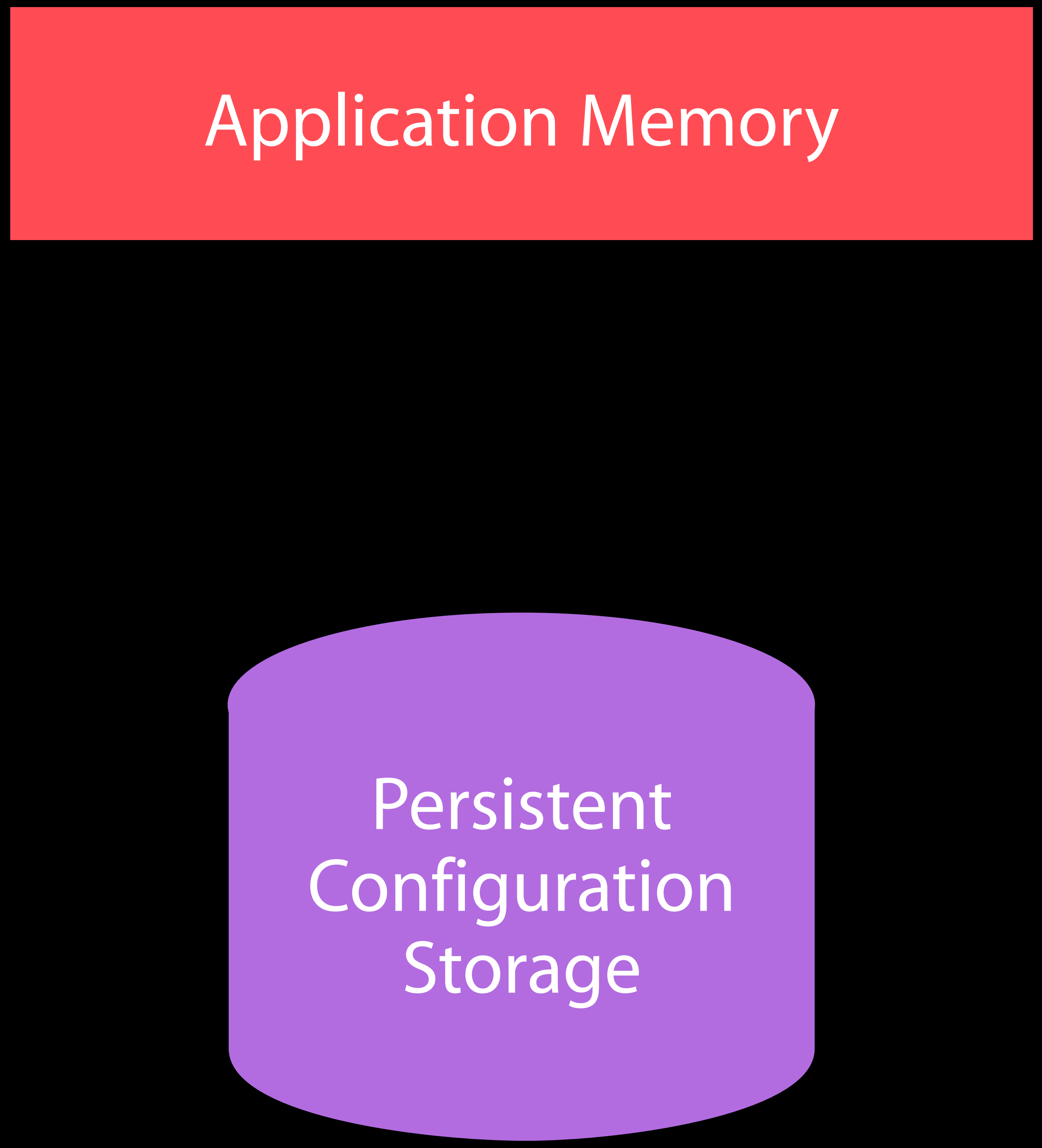
Take one down, pass it around…

Scott Perry Code Generator

Object Deletion

The problem

Today, deleting objects requires

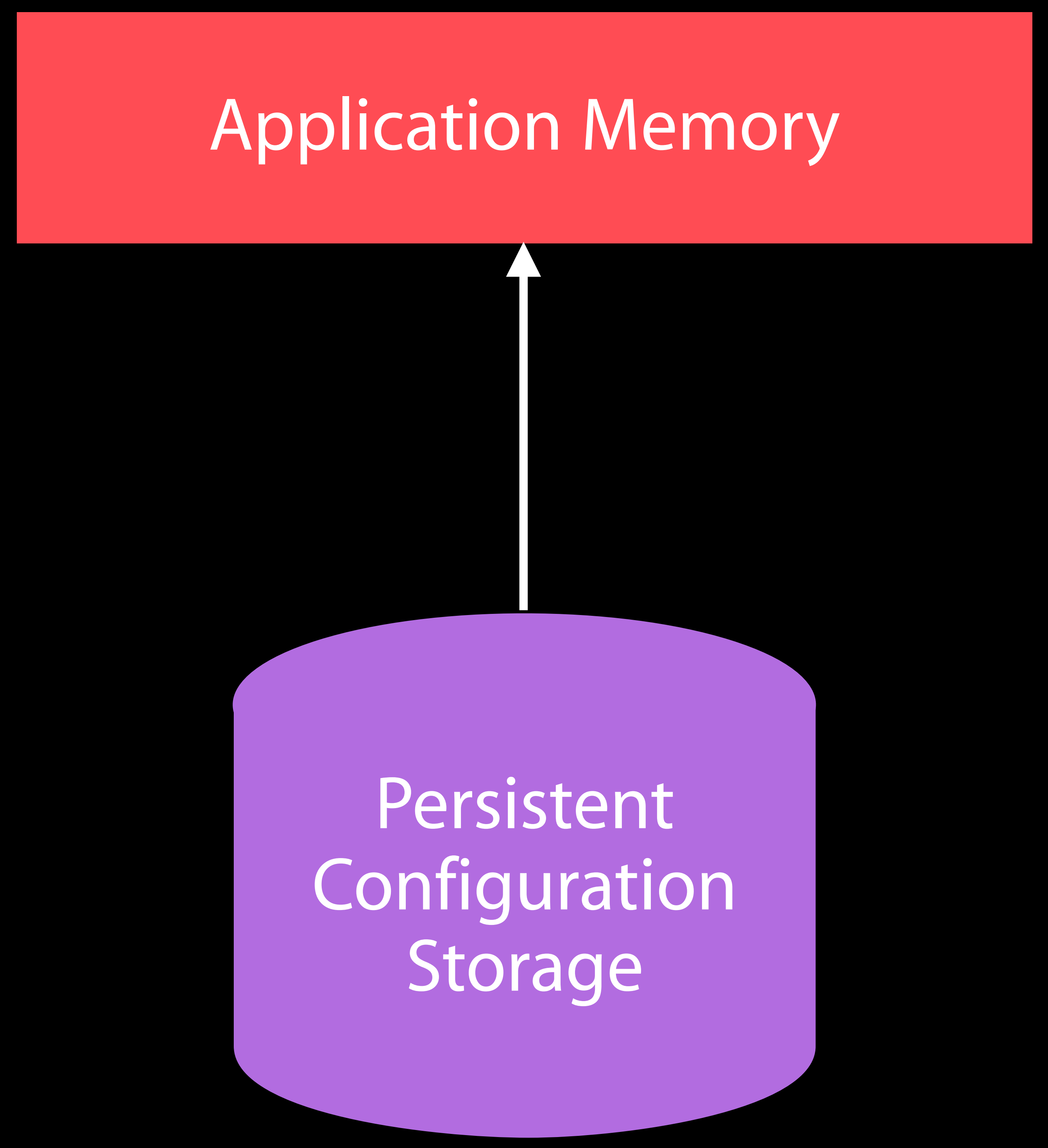


Object Deletion

The problem

Today, deleting objects requires

- Fetching some objects

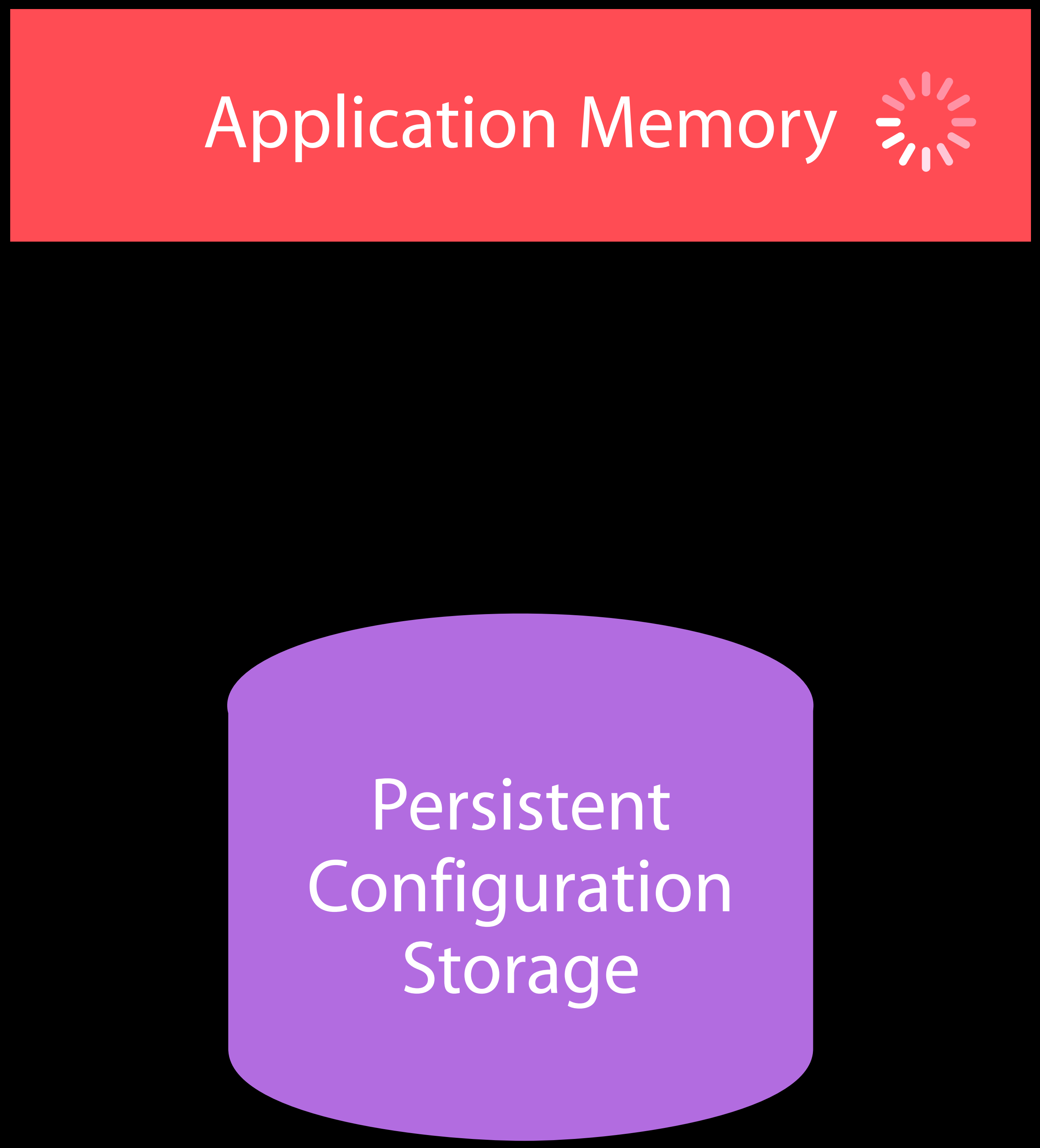


Object Deletion

The problem

Today, deleting objects requires

- Fetching some objects
- Marking each object for deletion

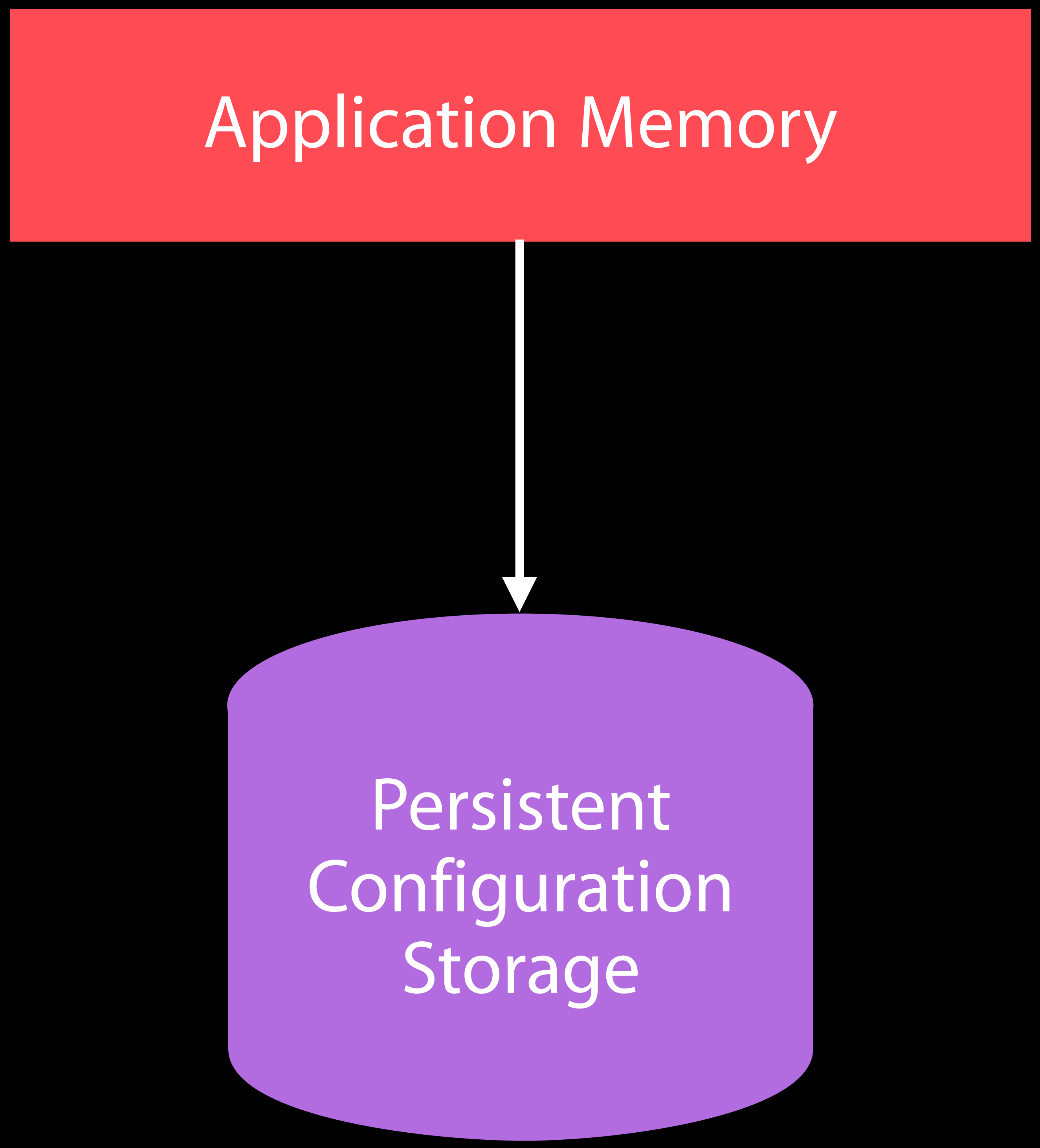


Object Deletion

The problem

Today, deleting objects requires

- Fetching some objects
- Marking each object for deletion
- Saving the changes

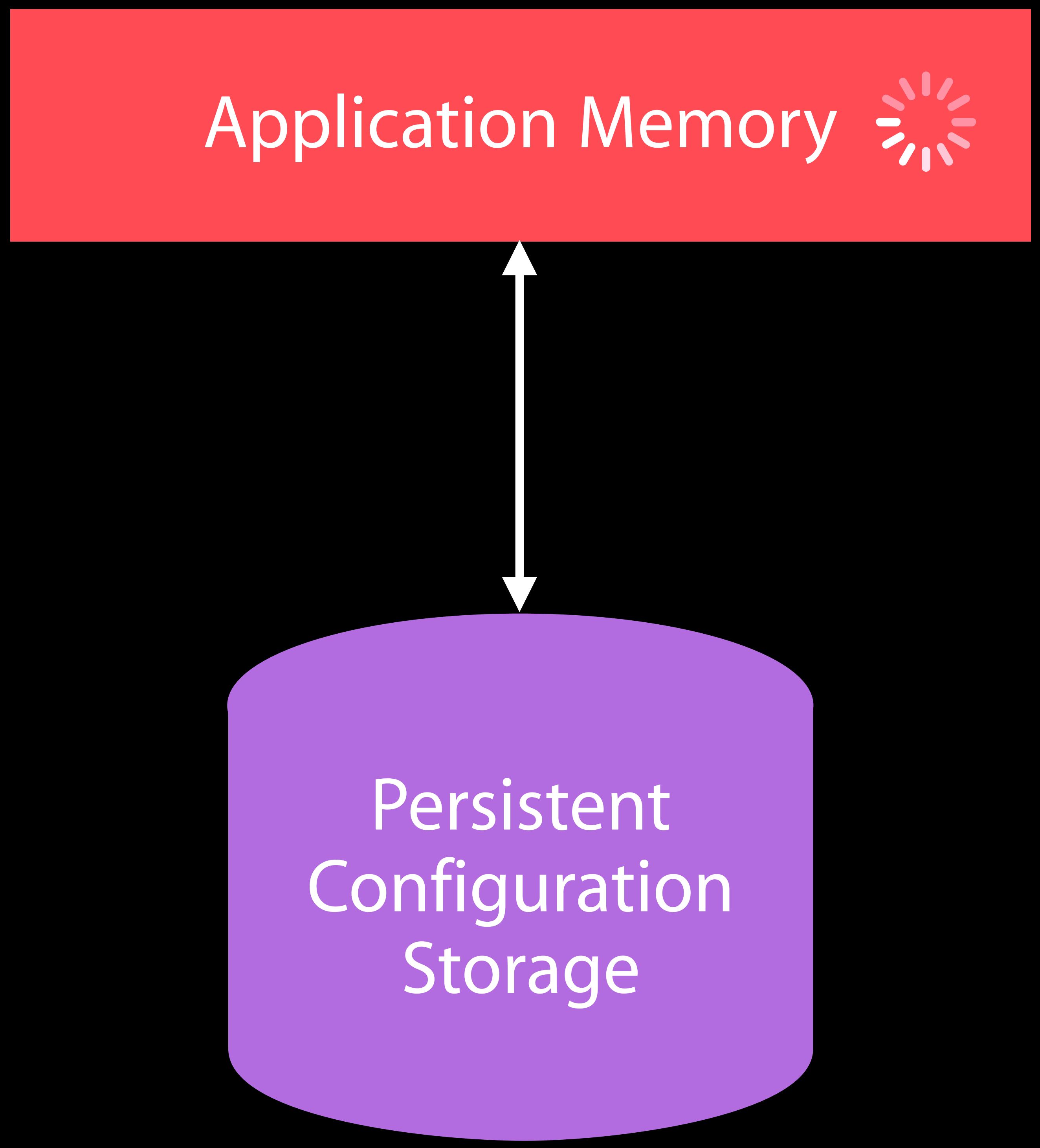


Object Deletion

The problem

Today, deleting objects requires

- Fetching some objects
- Marking each object for deletion
- Saving the changes
- Repeat



Object Deletion

The problem

You shouldn’t have to load objects into memory to delete them

NSBatchDeleteRequest

NEW

The solution

Very similar to NSBatchUpdateRequest

- Acts directly on the Persistent Store

NSBatchDeleteRequest

The solution

NEW

Very similar to NSBatchUpdateRequest

- Acts directly on the Persistent Store

Instances of NSBatchDeleteRequest wrap an instance of NSFetchRequest

- One entity
- One or more stores
- Supports predicates as well as sort descriptors and offset/limit

NSBatchDeleteResult

The solution

NEW

Success/failure

Count of objects deleted

Object IDs of objects deleted

Batch Deletions

Limitations

Changes are not reflected in the context

Not all validation rules are enforced

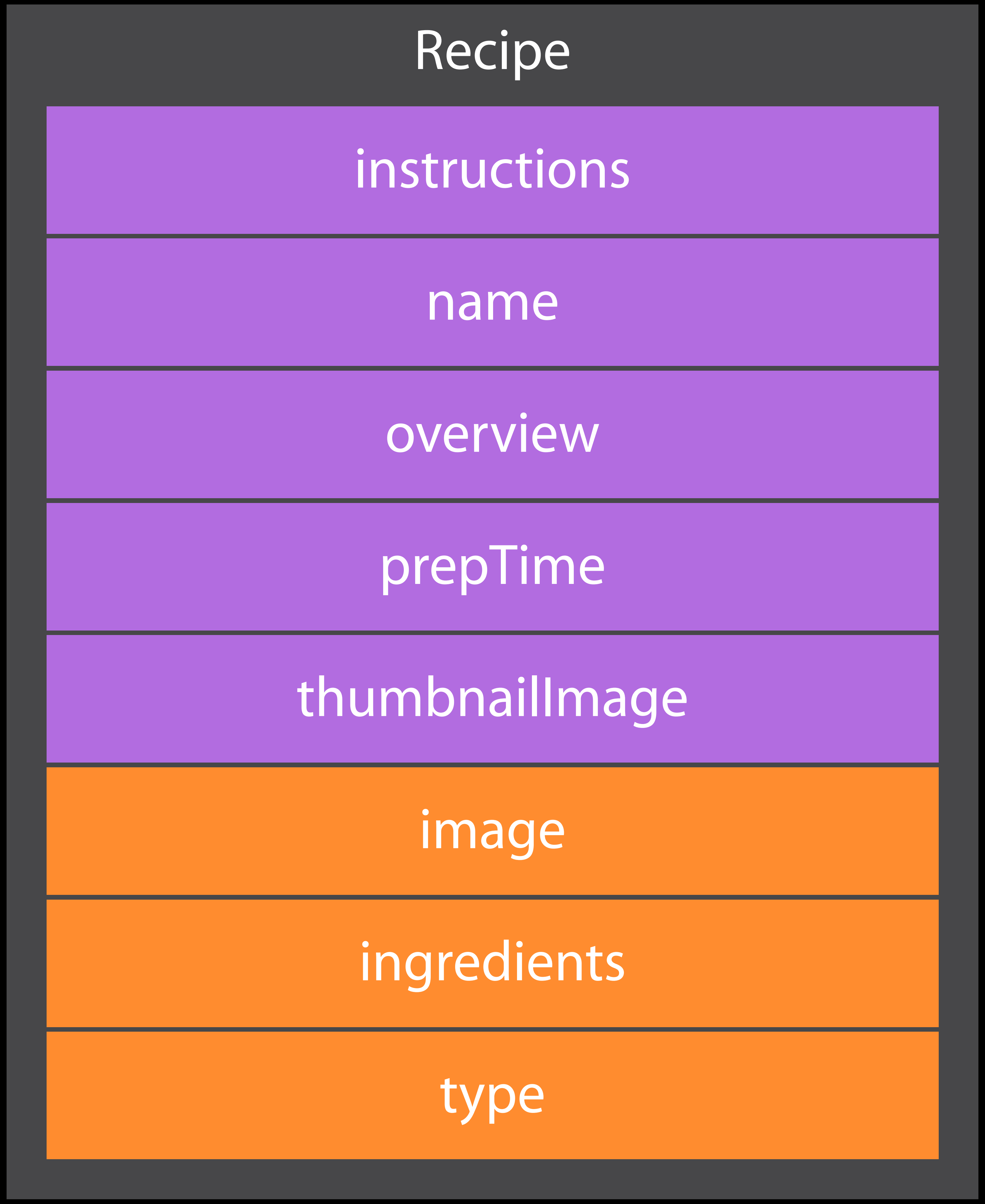
No object notifications

Demo

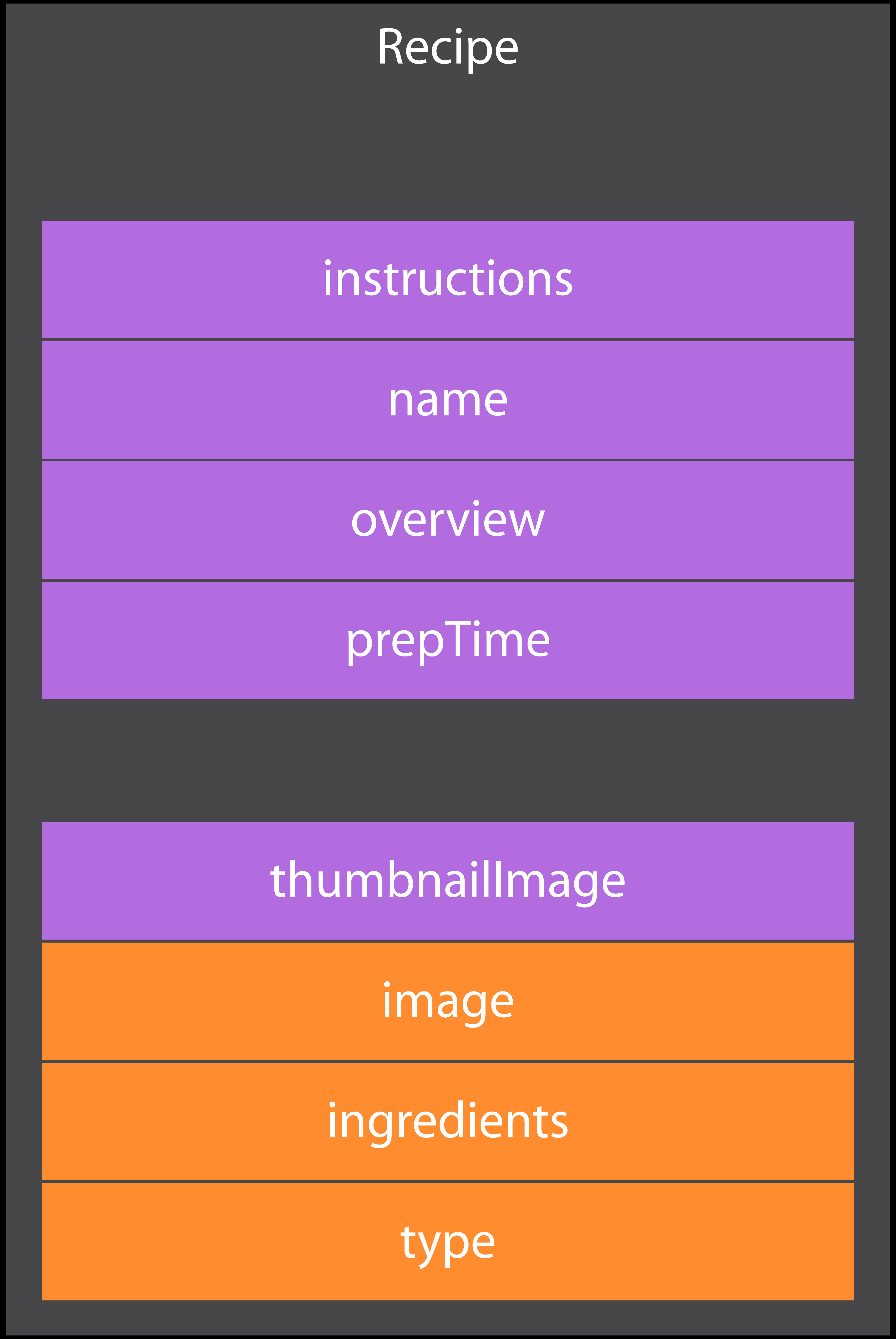
NSBatchDeleteRequest

Model Versioning

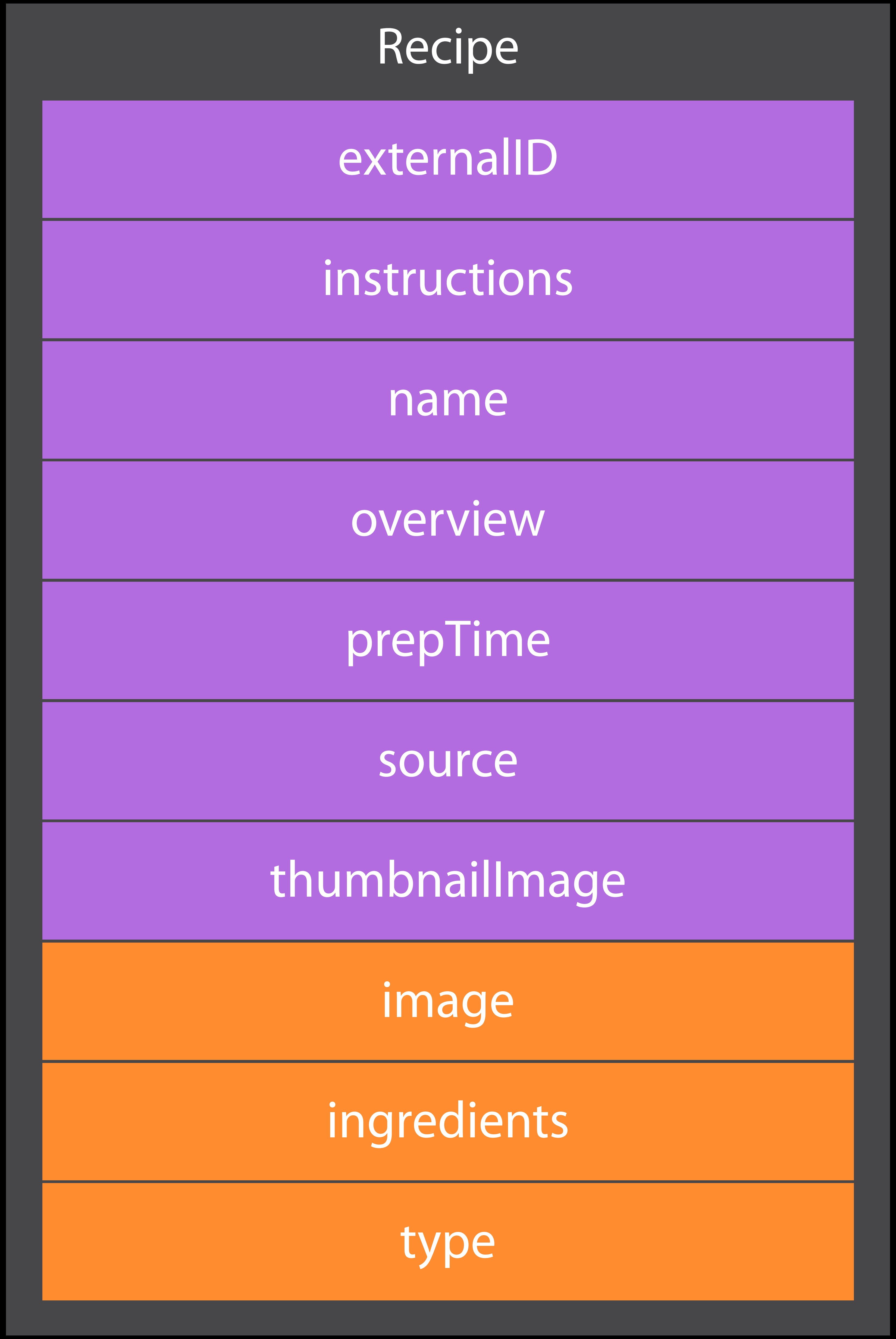
Models Change



Models Change



Models Change



Models Change

…But migrations stay the same

```
Error Domain=NSCocoaErrorDomain Code=134130 "Persistent store migration failed, missing source managed
object model." UserInfo=0x1054a2380 {
    URL=file:///private/var/mobile/Containers/Data/Application/6CD803A7-91EC…
    metadata={
        NSPersistenceFrameworkVersion = 619;
        NSStoreModelVersionHashesVersion = 3;
        NSStoreModelVersionIdentifiers =     (
            ""
        );
        NSStoreType = SQLite;
        NSStoreUUID = "EF65B546-1D30-48A4-9090-E274F4DF7822";
        "_NSAutoVacuumLevel" = 2;
        NSStoreModelVersionHashes =     {
            Recipe = <81b7e3b1 450cf990 6f1c8f36 89786a0b f61715cb afd9016b …
            …
        };
    },
    reason=Can't find model for source store
}
```

Models Change

…But migrations stay the same

```
Error Domain=NSCocoaErrorDomain Code=134130 "Persistent store migration failed, missing source managed
object model." UserInfo=0x1054a2380 {
    URL=file:///private/var/mobile/Containers/Data/Application/6CD803A7-91EC…
    metadata={
        NSPersistenceFrameworkVersion = 619;
        NSStoreModelVersionHashesVersion = 3;
        NSStoreModelVersionIdentifiers =     (
            ""
        );
        NSStoreType = SQLite;
        NSStoreUUID = "EF65B546-1D30-48A4-9090-E274F4DF7822";
        "_NSAutoVacuumLevel" = 2;
        NSStoreModelVersionHashes =     {
            Recipe = <81b7e3b1 450cf990 6f1c8f36 89786a0b f61715cb afd9016b …
            …
        };
    },
    reason=Can't find model for source store
}
```

Models Change

The problem

Iterating models is cumbersome

Forgetting to deploy model versions is dangerous

Models Change

The problem

Iterating models is cumbersome

Forgetting to deploy model versions is dangerous

Automatic lightweight migrations should “Just Work™”

Model Caching

The solution

NEW

NSManagedObjectModel copied to store

Automatically updates existing stores

Lightweight migrations fetch the model from the store

Model Caching

Limitations

Only SQLite stores

Cached model is not available to explicit migrations

API Modernization

Generics and Nullability

Better living through more explicit types

`nonnull` (default), `nullable`, and `null_resettable`

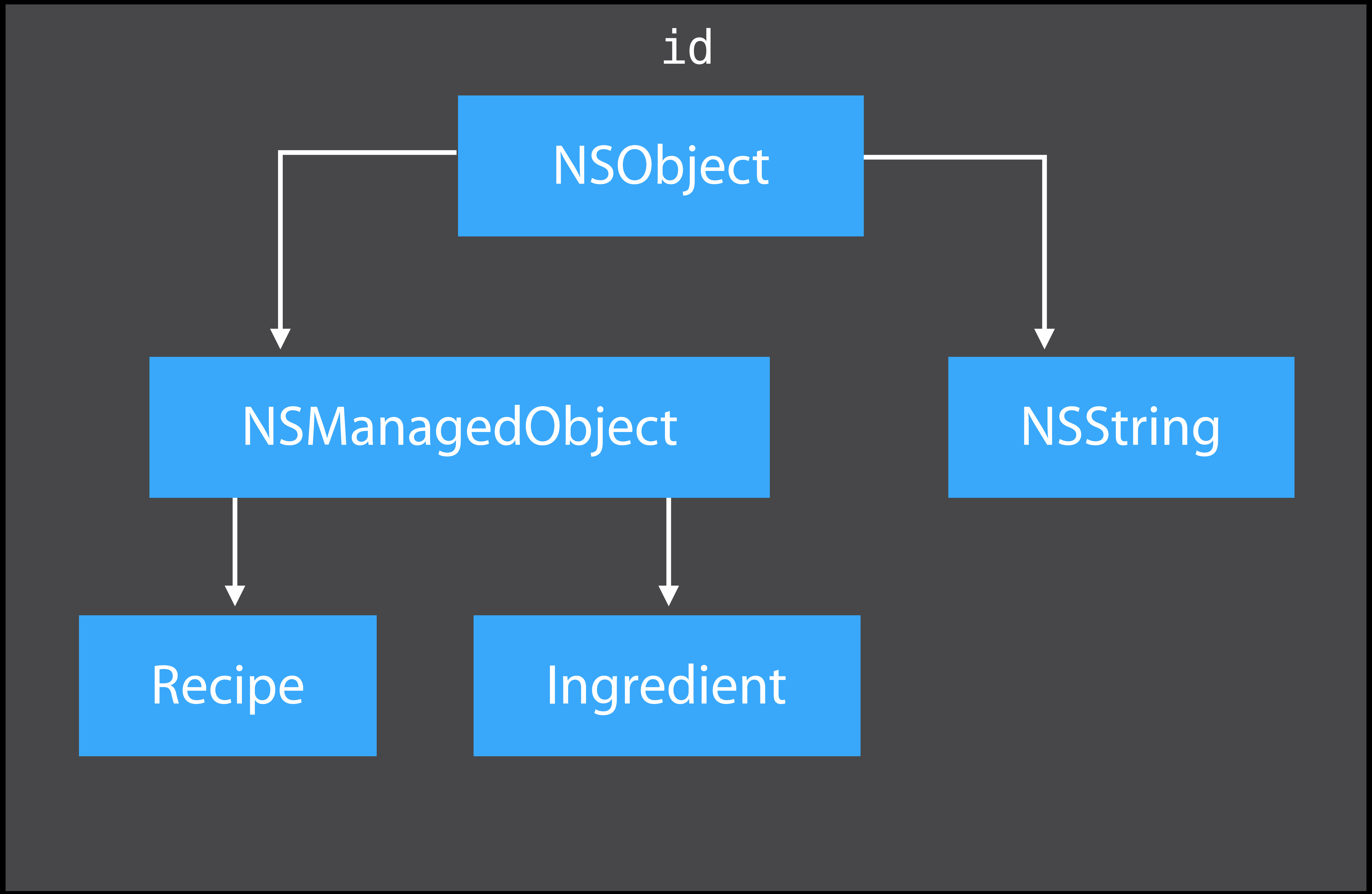
`__kindof` allows for easier casting

Generics and Nullability

Better living through more explicit types

`nonnull` (default), `nullable`, and `null_resettable`

`__kindof` allows for easier casting

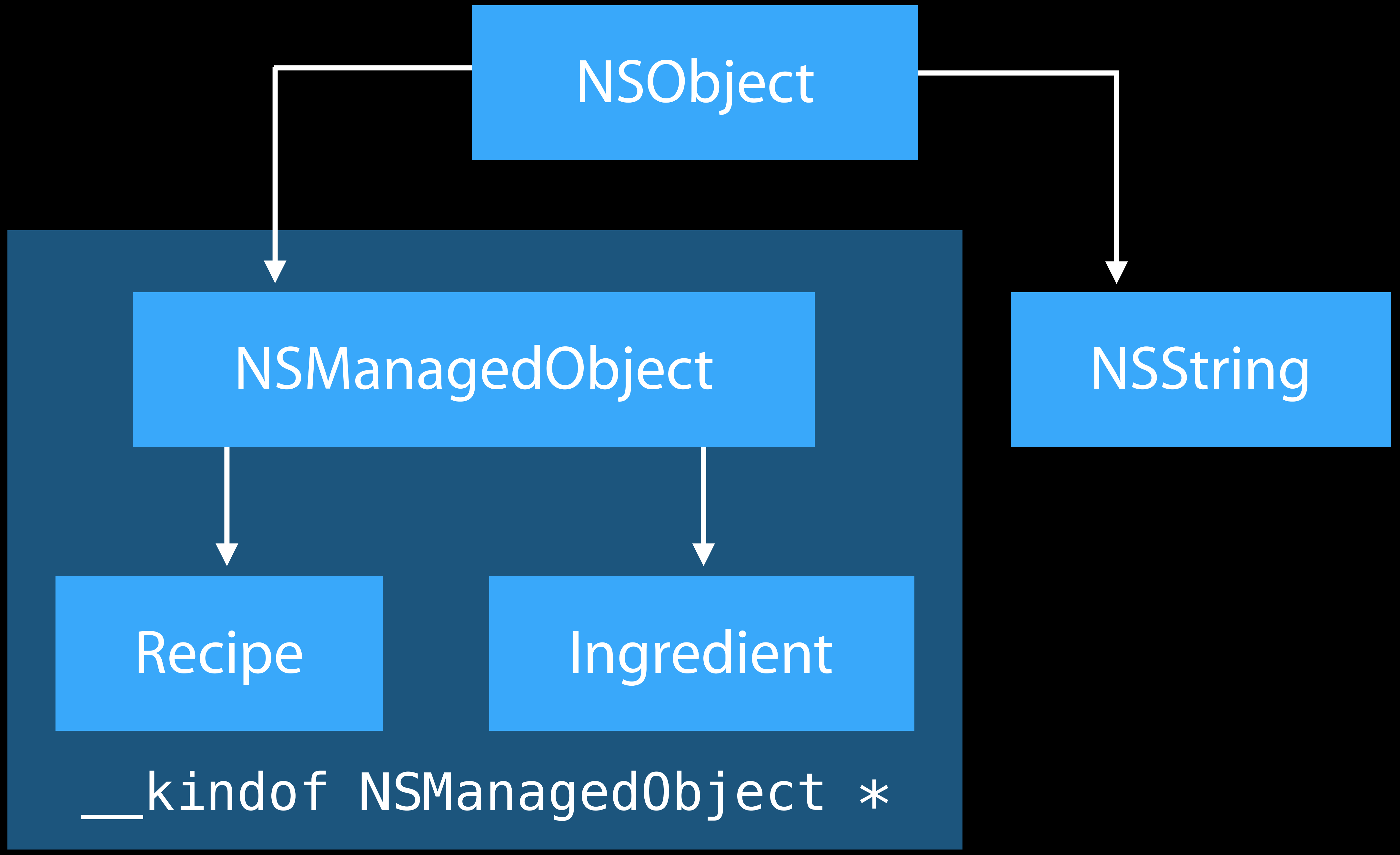


Generics and Nullability

Better living through more explicit types

`nonnull` (default), `nullable`, and `null_resettable`

`__kindof` allows for easier casting



Generics and Nullability

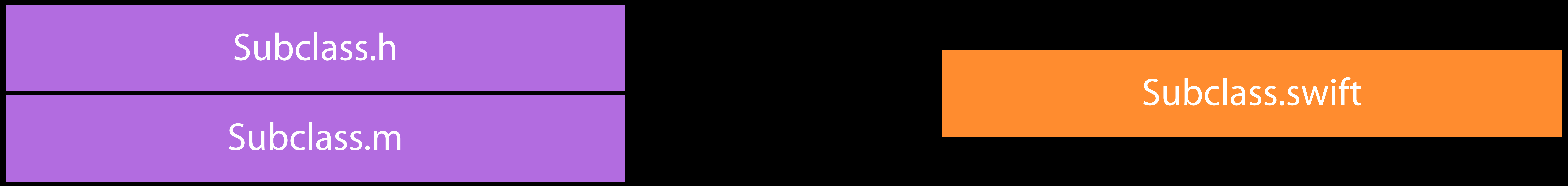
Better living through more explicit types

`nonnull` (default), `nullable`, and `null_resettable`

`__kindof` allows for easier casting

Generated subclasses use generics for to-many relationships

Generated Subclasses



Generated Subclasses

Subclass.h

Subclass+NSManagedProperties.h

Subclass.m

Subclass.swift

Subclass+NSManagedProperties.swift

Generated Subclasses

Subclass+NSManagedProperties.h

Subclass+NSManagedProperties.swift

Generated Subclasses

Subclass+NSManagedProperties.h

```
#import "Recipe.h"

NS_ASSUME_NONNULL_BEGIN

@interface Recipe (CoreDataProperties)

@property (nullable, nonatomic, retain) id thumbnailImage;
@property (nullable, nonatomic, retain) NSString *source;
@property (nullable, nonatomic, retain) NSString *instructions;
@property (nullable, nonatomic, retain) NSString *prepTime;
@property (nullable, nonatomic, retain) NSString *overview;
@property (nullable, nonatomic, retain) NSString *externalID;
@property (nullable, nonatomic, retain) NSString *name;
@property (nullable, nonatomic, retain) NSSet<Ingredient *> *ingredients;
@property (nullable, nonatomic, retain) NSManagedObject *image;
@property (nullable, nonatomic, retain) NSManagedObject *type;

@end

@interface Recipe (CoreDataGeneratedAccessors)

- (void)addIngredientsObject:(Ingredient *)value;
- (void)removeIngredientsObject:(Ingredient *)value;
- (void)addIngredients:(NSSet<Ingredient *> *)values;
- (void)removeIngredients:(NSSet<Ingredient *> *)values;

@end

NS_ASSUME_NONNULL_END
```

Subclass+NSManagedProperties.swift

```
import Foundation
import CoreData

extension Recipe {

    @NSManaged var thumbnailImage: NSObject?
    @NSManaged var source: String?
    @NSManaged var instructions: String?
    @NSManaged var prepTime: String?
    @NSManaged var overview: String?
    @NSManaged var externalID: String?
    @NSManaged var name: String?
    @NSManaged var ingredients: NSSet?
    @NSManaged var image: NSManagedObject?
    @NSManaged var type: NSManagedObject?

}
```

Concurrency

Confinement is dead, long live queues

Concurrency

Confinement is dead, long live queues

`ConfinementConcurrencyType` is deprecated

Concurrency

Confinement is dead, long live queues

`ConfinementConcurrencyType` is deprecated

`init()` has been deprecated

Concurrency

Confinement is dead, long live queues

`ConfinementConcurrencyType` is deprecated

`init()` has been deprecated

`init(concurrencyType:)` is the designated initializer

- Use `PrivateQueueConcurrencyType` or `MainQueueConcurrencyType`

Concurrency

Confinement is dead, long live queues

`ConfinementConcurrencyType` is deprecated

`init()` has been deprecated

`init(concurrencyType:)` is the designated initializer

- Use `PrivateQueueConcurrencyType` or `MainQueueConcurrencyType`

NSManagedObjectContext Documentation	developer.apple.com
What's New in Core Data on iOS	WWDC11

Core Data Performance

Apps Improve

Models get more complex

Stores get larger

Queries get more interesting

Apps Improve

Models get more complex

Stores get larger

Queries get more interesting

Apps stay fast!

Slow Can Be Surprising

Scale differs between development and production

The simulator is faster than the device

Slow Can Be Surprising

Scale differs between development and production

The simulator is faster than the device

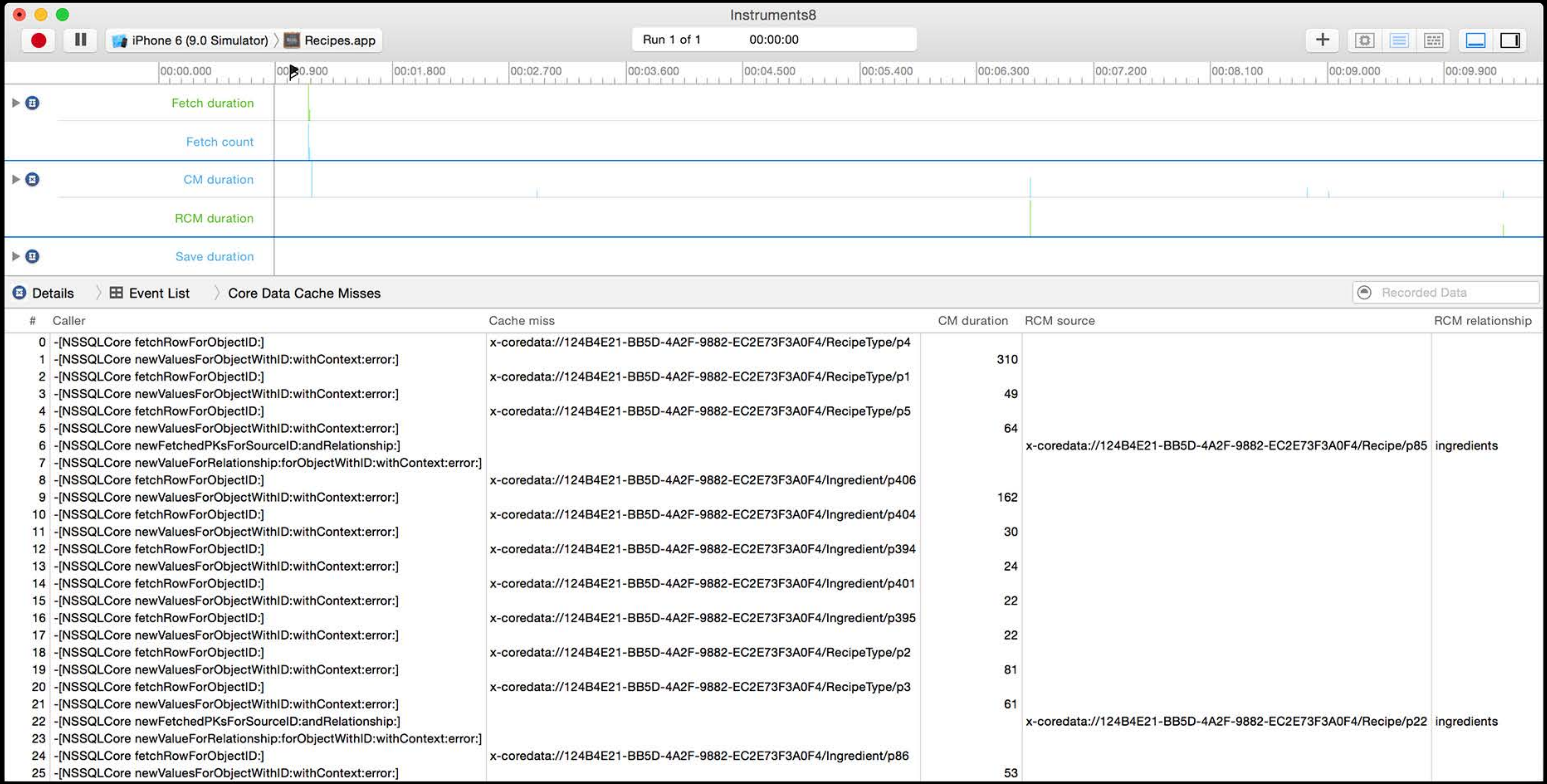
Users use devices in production

Find Problems Before They Find You

Predicting the future with tools

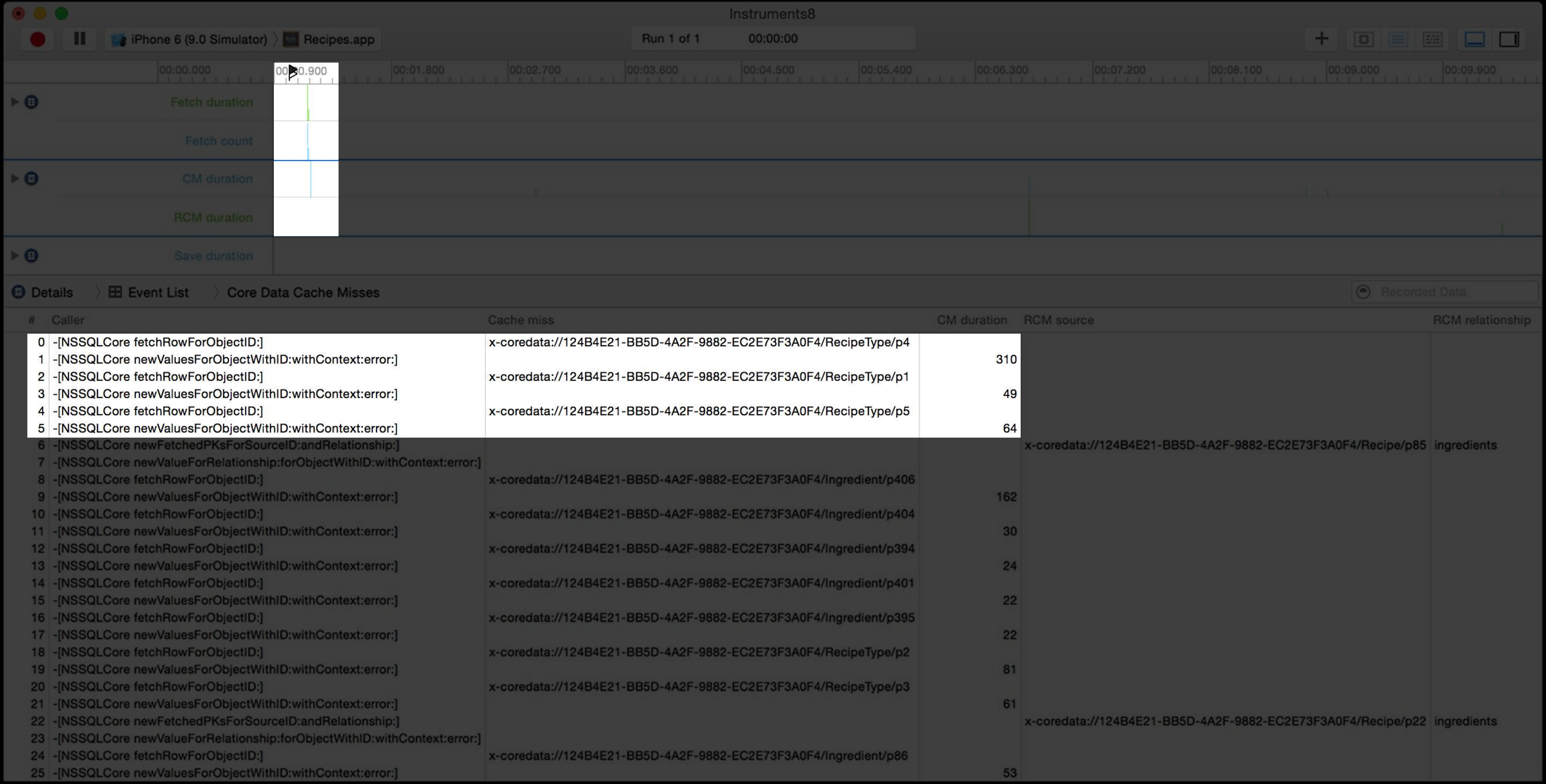
Relationship Faults

Relationship Faults



#	Caller	Cache miss	CM duration	RCM source	RCM relationship
0	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p4			
1	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		310		
2	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p1			
3	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		49		
4	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p5			
5	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		64		
6	-[NSSQLCore newFetchedPKsForSourceID:andRelationship:]			x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Recipe/p85	ingredients
7	-[NSSQLCore newValueForRelationship:forObjectWithID:withContext:error:]				
8	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p406			
9	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		162		
10	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p404			
11	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		30		
12	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p394			
13	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		24		
14	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p401			
15	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		22		
16	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p395			
17	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		22		
18	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p2			
19	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		81		
20	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p3			
21	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		61		
22	-[NSSQLCore newFetchedPKsForSourceID:andRelationship:]			x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Recipe/p22	ingredients
23	-[NSSQLCore newValueForRelationship:forObjectWithID:withContext:error:]				
24	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p86			
25	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		53		

Relationship Faults



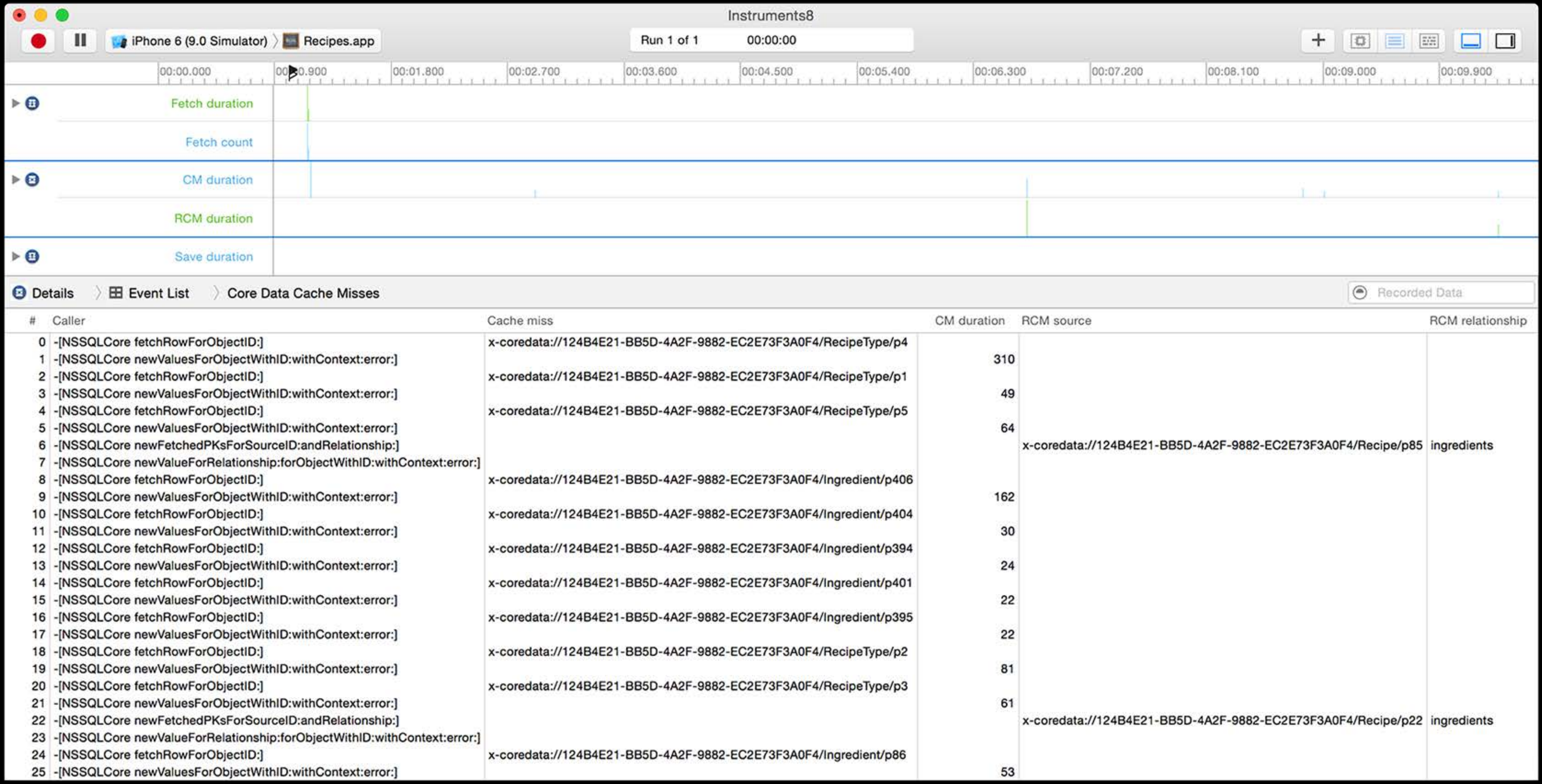
#	Caller	Cache miss	CM duration	RCM source	RCM relationship
0	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p4			
1	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		310		
2	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p1			
3	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		49		
4	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p5			
5	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		64		
6	-[NSSQLCore newFetchedPKsForSourceID:andRelationship:]			x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Recipe/p85	ingredients
7	-[NSSQLCore newValueForRelationship:forObjectWithID:withContext:error:]				
8	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p406			
9	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		162		
10	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p404			
11	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		30		
12	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p394			
13	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		24		
14	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p401			
15	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		22		
16	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p395			
17	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		22		
18	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p2			
19	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		81		
20	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p3			
21	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		61		
22	-[NSSQLCore newFetchedPKsForSourceID:andRelationship:]			x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Recipe/p22	ingredients
23	-[NSSQLCore newValueForRelationship:forObjectWithID:withContext:error:]				
24	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p86			
25	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		53		

Relationship Faults

Prefetch the objects you're going to use

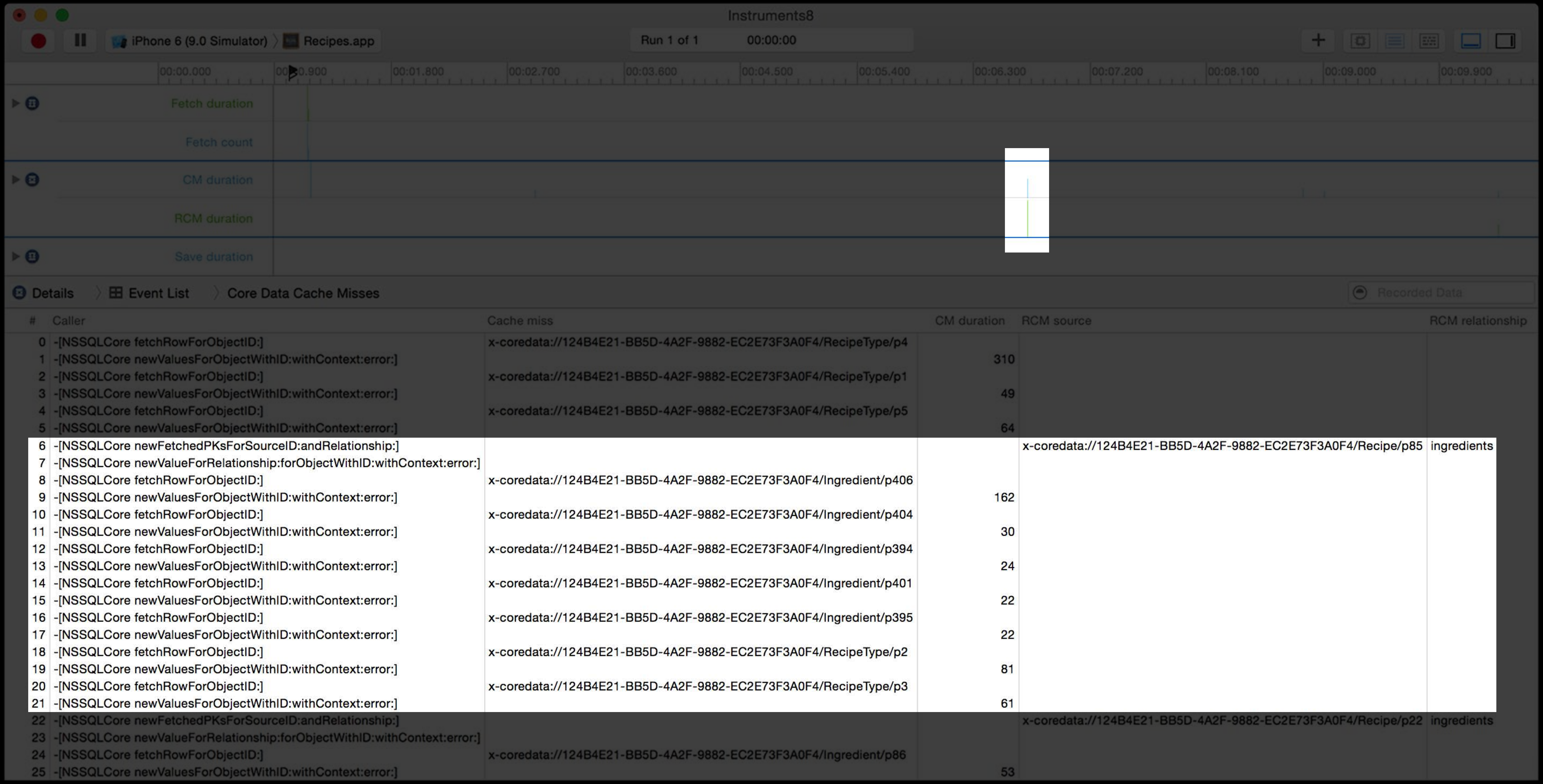
```
var recipeRequest = NSFetchRequest(entityName:"Recipe")

let sortDescriptor = NSSortDescriptor(key:"name", ascending: true)
recipeRequest.sortDescriptors = [sortDestcriptor]

recipeRequest.relationshipKeyPathsForPrefetching = ["type"]

context.executeFetchRequest(recipeRequest)
```

Relationship Faults



#	Caller	Cache miss	CM duration	RCM source	RCM relationship
0	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p4			
1	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		310		
2	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p1			
3	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		49		
4	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p5			
5	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		64		
6	-[NSSQLCore newFetchedPKsForSourceID:andRelationship:]			x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Recipe/p85	ingredients
7	-[NSSQLCore newValueForRelationship:forObjectWithID:withContext:error:]				
8	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p406			
9	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		162		
10	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p404			
11	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		30		
12	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p394			
13	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		24		
14	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p401			
15	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		22		
16	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p395			
17	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		22		
18	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p2			
19	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		81		
20	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p3			
21	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		61		
22	-[NSSQLCore newFetchedPKsForSourceID:andRelationship:]			x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Recipe/p22	ingredients
23	-[NSSQLCore newValueForRelationship:forObjectWithID:withContext:error:]				
24	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p86			
25	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		53		

Relationship Faults



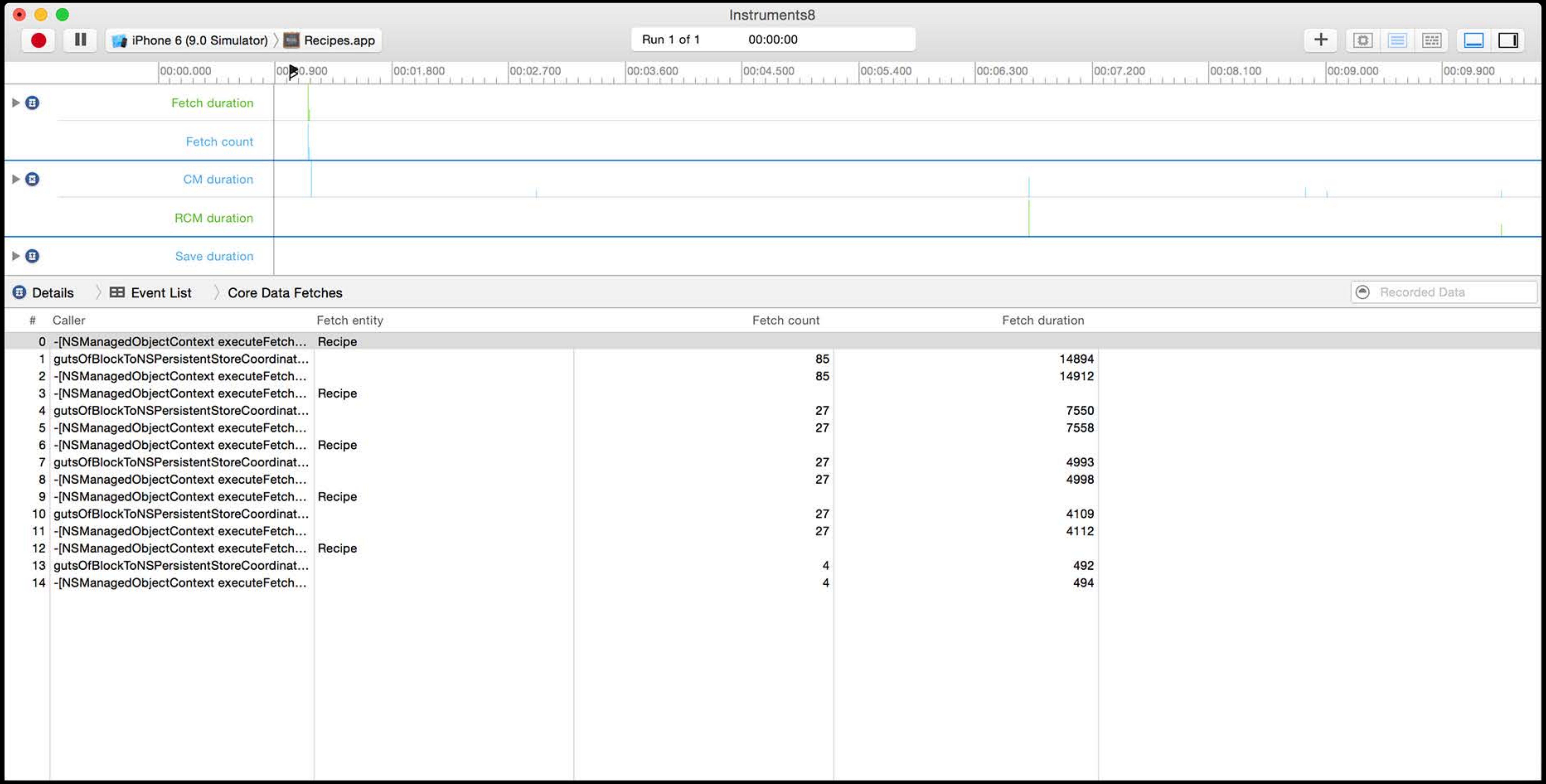
#	Caller	Cache miss	CM duration	RCM source	RCM relationship
0	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p4			
1	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		310		
2	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p1			
3	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		49		
4	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p5			
5	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		64		
6	-[NSSQLCore newFetchedPKsForSourceID:andRelationship:]			x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Recipe/p85	ingredients
7	-[NSSQLCore newValueForRelationship:forObjectWithID:withContext:error:]				
8	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p406			
9	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		162		
10	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p404			
11	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		30		
12	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p394			
13	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		24		
14	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p401			
15	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		22		
16	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p395			
17	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		22		
18	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p2			
19	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		81		
20	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/RecipeType/p3			
21	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		61		
22	-[NSSQLCore newFetchedPKsForSourceID:andRelationship:]			x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Recipe/p22	ingredients
23	-[NSSQLCore newValueForRelationship:forObjectWithID:withContext:error:]				
24	-[NSSQLCore fetchRowForObjectID:]	x-coredata://124B4E21-BB5D-4A2F-9882-EC2E73F3A0F4/Ingredient/p86			
25	-[NSSQLCore newValuesForObjectWithID:withContext:error:]		53		

Relationship Faults

Prefetch the objects you’re going to use

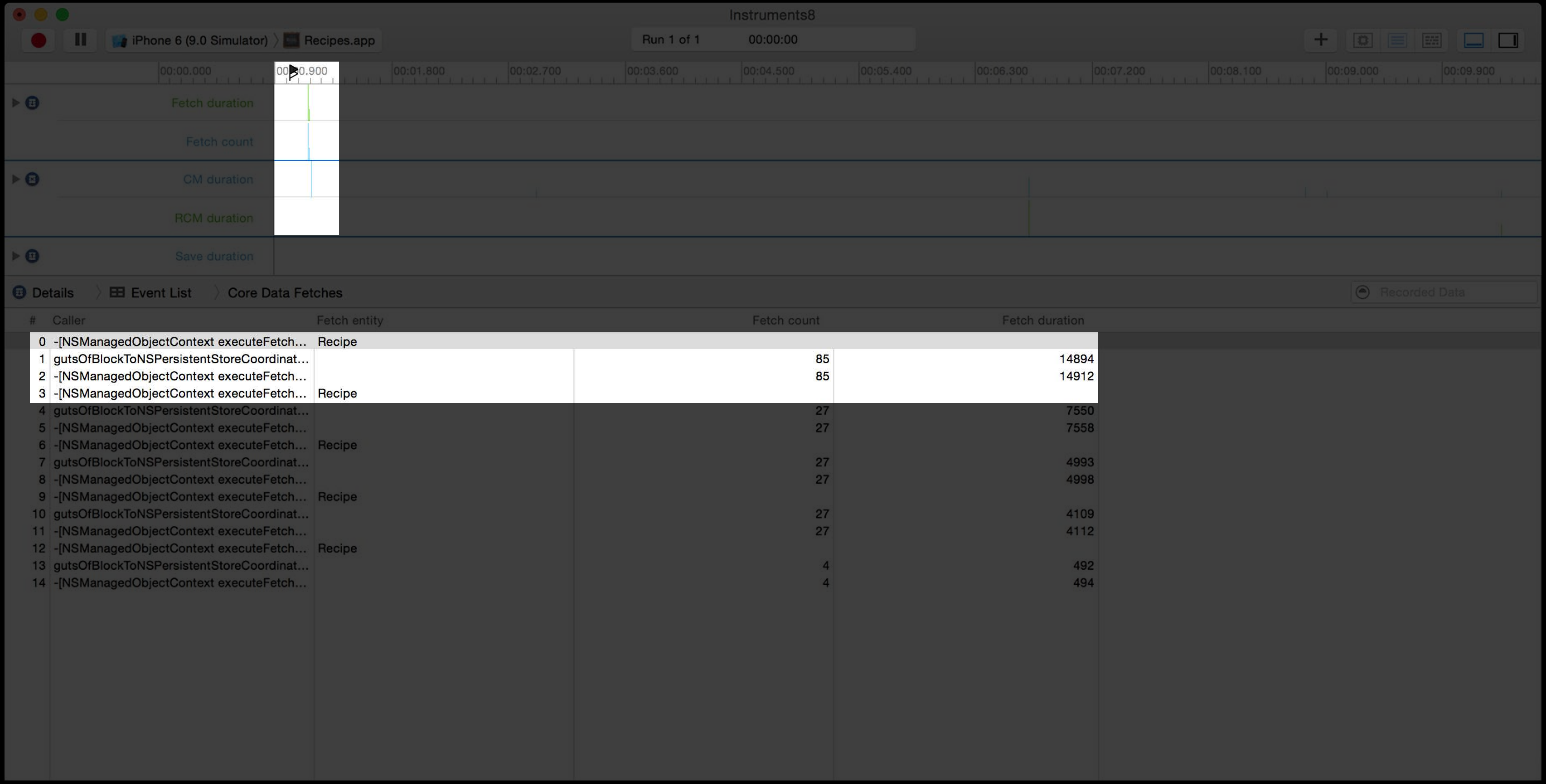
```
var ingredientRequest = NSFetchRequest(entityName:"Ingredient")

ingredientRequest.predicate = NSPredicate(format:"recipe = %@",
argumentArray:[recipe])

context.executeFetchRequest(ingredientRequest)
```

Large Fetches



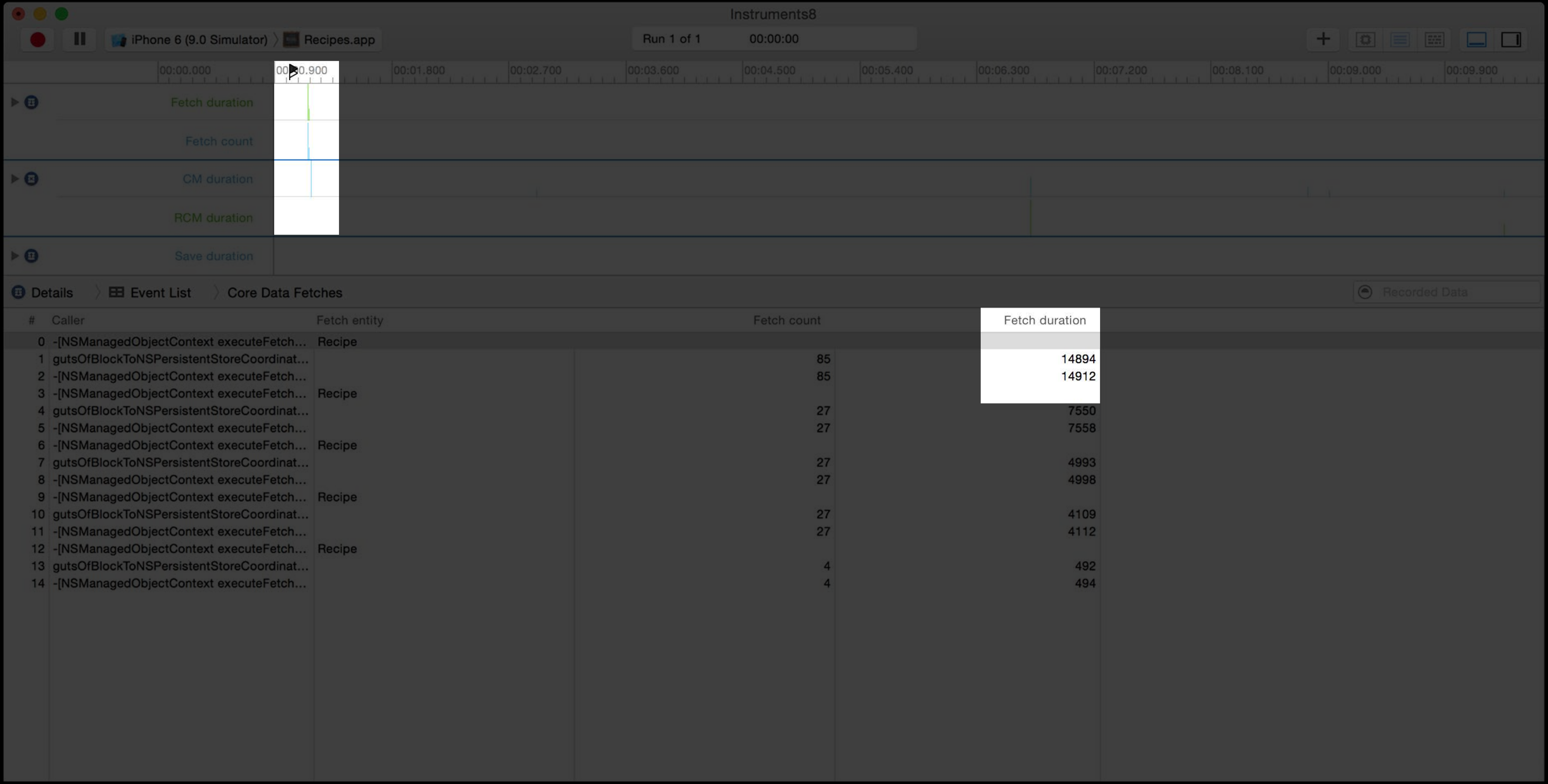
#	Caller	Fetch entity	Fetch count	Fetch duration
0	-[NSManagedObjectContext executeFetch...	Recipe		
1	gutsOfBlockToNSPersistentStoreCoordinat...		85	14894
2	-[NSManagedObjectContext executeFetch...		85	14912
3	-[NSManagedObjectContext executeFetch...	Recipe		
4	gutsOfBlockToNSPersistentStoreCoordinat...		27	7550
5	-[NSManagedObjectContext executeFetch...		27	7558
6	-[NSManagedObjectContext executeFetch...	Recipe		
7	gutsOfBlockToNSPersistentStoreCoordinat...		27	4993
8	-[NSManagedObjectContext executeFetch...		27	4998
9	-[NSManagedObjectContext executeFetch...	Recipe		
10	gutsOfBlockToNSPersistentStoreCoordinat...		27	4109
11	-[NSManagedObjectContext executeFetch...		27	4112
12	-[NSManagedObjectContext executeFetch...	Recipe		
13	gutsOfBlockToNSPersistentStoreCoordinat...		4	492
14	-[NSManagedObjectContext executeFetch...		4	494

Large Fetches



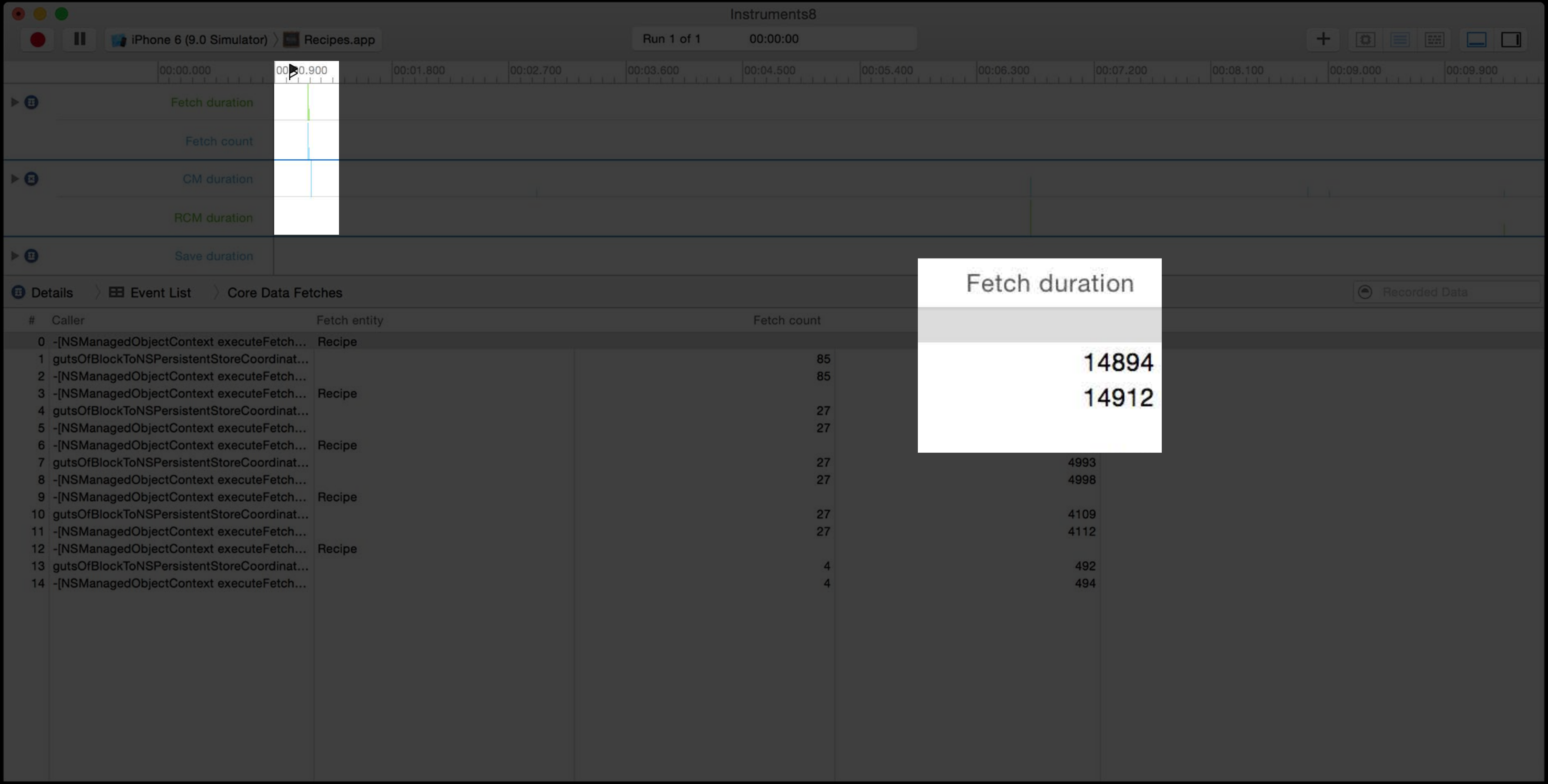
#	Caller	Fetch entity	Fetch count	Fetch duration
0	-[NSManagedObjectContext executeFetch...	Recipe		
1	gutsOfBlockToNSPersistentStoreCoordinat...		85	14894
2	-[NSManagedObjectContext executeFetch...		85	14912
3	-[NSManagedObjectContext executeFetch...	Recipe		
4	gutsOfBlockToNSPersistentStoreCoordinat...		27	7550
5	-[NSManagedObjectContext executeFetch...		27	7558
6	-[NSManagedObjectContext executeFetch...	Recipe		
7	gutsOfBlockToNSPersistentStoreCoordinat...		27	4993
8	-[NSManagedObjectContext executeFetch...		27	4998
9	-[NSManagedObjectContext executeFetch...	Recipe		
10	gutsOfBlockToNSPersistentStoreCoordinat...		27	4109
11	-[NSManagedObjectContext executeFetch...		27	4112
12	-[NSManagedObjectContext executeFetch...	Recipe		
13	gutsOfBlockToNSPersistentStoreCoordinat...		4	492
14	-[NSManagedObjectContext executeFetch...		4	494

Large Fetches



#	Caller	Fetch entity	Fetch count	Fetch duration
0	-[NSManagedObjectContext executeFetch...	Recipe		
1	gutsOfBlockToNSPersistentStoreCoordinat...		85	14894
2	-[NSManagedObjectContext executeFetch...		85	14912
3	-[NSManagedObjectContext executeFetch...	Recipe		
4	gutsOfBlockToNSPersistentStoreCoordinat...		27	7550
5	-[NSManagedObjectContext executeFetch...		27	7558
6	-[NSManagedObjectContext executeFetch...	Recipe		
7	gutsOfBlockToNSPersistentStoreCoordinat...		27	4993
8	-[NSManagedObjectContext executeFetch...		27	4998
9	-[NSManagedObjectContext executeFetch...	Recipe		
10	gutsOfBlockToNSPersistentStoreCoordinat...		27	4109
11	-[NSManagedObjectContext executeFetch...		27	4112
12	-[NSManagedObjectContext executeFetch...	Recipe		
13	gutsOfBlockToNSPersistentStoreCoordinat...		4	492
14	-[NSManagedObjectContext executeFetch...		4	494

Large Fetches



Fetch duration

14894
14912

#	Caller	Fetch entity	Fetch count	Fetch duration
0	-[NSManagedObjectContext executeFetch...	Recipe		
1	gutsOfBlockToNSPersistentStoreCoordinat...		85	14894
2	-[NSManagedObjectContext executeFetch...		85	14912
3	-[NSManagedObjectContext executeFetch...	Recipe		
4	gutsOfBlockToNSPersistentStoreCoordinat...		27	[illegible]
5	-[NSManagedObjectContext executeFetch...		27	[illegible]
6	-[NSManagedObjectContext executeFetch...	Recipe		
7	gutsOfBlockToNSPersistentStoreCoordinat...		27	4993
8	-[NSManagedObjectContext executeFetch...		27	4998
9	-[NSManagedObjectContext executeFetch...	Recipe		
10	gutsOfBlockToNSPersistentStoreCoordinat...		27	4109
11	-[NSManagedObjectContext executeFetch...		27	4112
12	-[NSManagedObjectContext executeFetch...	Recipe		
13	gutsOfBlockToNSPersistentStoreCoordinat...		4	492
14	-[NSManagedObjectContext executeFetch...		4	494

Large Fetches

Take advantage of batching

```
var recipeRequest = NSFetchRequest(entityName:"Recipe")

let sortDescriptor = NSSortDescriptor(key:"name", ascending: true)
recipeRequest.sortDescriptors = [sortDestcriptor]

recipeRequest.fetchBatchSize = 30

context.executeFetchRequest(recipeRequest)
```

Complex Fetches

```
-com.apple.CoreData.SQLDebug 1
```

Larger time/count ratio

Complex Fetches

`-com.apple.CoreData.SQLDebug 1`

Larger time/count ratio

```
CoreData: sql: SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZEXTERNALID, t0.ZINSTRUCTIONS,
t0.ZNAME, t0.ZOVERVIEW, t0.ZPREPTIME, t0.ZSOURCE, t0.ZTHUMBNAILIMAGE,
t0.ZIMAGE, t0.ZTYPE FROM ZRECIPE t0 WHERE  NOT ( t0.Z_PK IN (SELECT
n1_t0.Z_PK FROM ZRECIPE n1_t0 GROUP BY  n1_t0.ZSOURCE,  n1_t0.ZEXTERNALID ))
CoreData: annotation: sql connection fetch time: 0.0766s
CoreData: annotation: total fetch execution time: 0.0786s for 85 rows.
```

Complex Fetches

```
-com.apple.CoreData.SQLDebug 1
```

Larger time/count ratio

```
CoreData: annotation: Connecting to sqlite database file at "/Users/numist/…
…
CoreData: sql: SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZEXTERNALID, t0.ZINSTRUCTIONS,
t0.ZNAME, t0.ZOVERVIEW, t0.ZPREPTIME, t0.ZSOURCE, t0.ZTHUMBNAILIMAGE,
t0.ZIMAGE, t0.ZTYPE FROM ZRECIPE t0 WHERE  NOT ( t0.Z_PK IN (SELECT
n1_t0.Z_PK FROM ZRECIPE n1_t0 GROUP BY  n1_t0.ZSOURCE,  n1_t0.ZEXTERNALID ))
CoreData: annotation: sql connection fetch time: 0.0766s
CoreData: annotation: total fetch execution time: 0.0786s for 85 rows.
```

Complex Fetches

EXPLAIN QUERY PLAN

```
$ sqlite3 "/Users/numist/…/Recipes.sqlite"
sqlite>
```

Complex Fetches

EXPLAIN QUERY PLAN

```
$ sqlite3 "/Users/numist/…/Recipes.sqlite"
sqlite> EXPLAIN QUERY PLAN SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZEXTERNALID,
t0.ZINSTRUCTIONS, t0.ZNAME, t0.ZOVERVIEW, t0.ZPREPTIME, t0.ZSOURCE,
t0.ZTHUMBNAILIMAGE, t0.ZIMAGE, t0.ZTYPE FROM ZRECIPE t0 WHERE  NOT ( t0.Z_PK
IN (SELECT n1_t0.Z_PK FROM ZRECIPE n1_t0 GROUP BY n1_t0.ZSOURCE,
n1_t0.ZEXTERNALID ));
```

Complex Fetches

EXPLAIN QUERY PLAN

```
sqlite> EXPLAIN QUERY PLAN SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZEXTERNALID,
t0.ZINSTRUCTIONS, t0.ZNAME, t0.ZOVERVIEW, t0.ZPREPTIME, t0.ZSOURCE,
t0.ZTHUMBNAILIMAGE, t0.ZIMAGE, t0.ZTYPE FROM ZRECIPE t0 WHERE  NOT ( t0.Z_PK
IN (SELECT n1_t0.Z_PK FROM ZRECIPE n1_t0 GROUP BY n1_t0.ZSOURCE,
n1_t0.ZEXTERNALID ));
sele  order          from  deta
----  -------------  ----  ----
0     0              0     SCAN TABLE ZRECIPE AS t0
0     0              0     EXECUTE LIST SUBQUERY 1
1     0              0     SCAN TABLE ZRECIPE AS n1_t0
1     0              0     USE TEMP B-TREE FOR GROUP BY
```

Complex Fetches

EXPLAIN QUERY PLAN

```
sqlite> EXPLAIN QUERY PLAN SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZEXTERNALID,
t0.ZINSTRUCTIONS, t0.ZNAME, t0.ZOVERVIEW, t0.ZPREPTIME, t0.ZSOURCE,
t0.ZTHUMBNAILIMAGE, t0.ZIMAGE, t0.ZTYPE FROM ZRECIPE t0 WHERE  NOT ( t0.Z_PK
IN (SELECT n1_t0.Z_PK FROM ZRECIPE n1_t0 GROUP BY n1_t0.ZSOURCE,
n1_t0.ZEXTERNALID ));
sele  order          from  deta
----  -------------  ----  ----
0     0              0     SCAN TABLE ZRECIPE AS t0
0     0              0     EXECUTE LIST SUBQUERY 1
1     0              0     SCAN TABLE ZRECIPE AS n1_t0
1     0              0     USE TEMP B-TREE FOR GROUP BY
```

Complex Fetches

EXPLAIN QUERY PLAN

```
sqlite> EXPLAIN QUERY PLAN SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZEXTERNALID,
t0.ZINSTRUCTIONS, t0.ZNAME, t0.ZOVERVIEW, t0.ZPREPTIME, t0.ZSOURCE,
t0.ZTHUMBNAILIMAGE, t0.ZIMAGE, t0.ZTYPE FROM ZRECIPE t0 WHERE  NOT ( t0.Z_PK
IN (SELECT n1_t0.Z_PK FROM ZRECIPE n1_t0 GROUP BY n1_t0.ZSOURCE,
n1_t0.ZEXTERNALID ));
sele  order              from  deta
----  -------------      ----  ----
0     0                  0     SCAN TABLE ZRECIPE AS t0
0     0                  0     EXECUTE LIST SUBQUERY 1
1     0                  0     SCAN TABLE ZRECIPE AS n1_t0
1     0                  0     USE TEMP B-TREE FOR GROUP BY
```

Complex Fetches

EXPLAIN QUERY PLAN

```
sqlite> EXPLAIN QUERY PLAN SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZEXTERNALID,
t0.ZINSTRUCTIONS, t0.ZNAME, t0.ZOVERVIEW, t0.ZPREPTIME, t0.ZSOURCE,
t0.ZTHUMBNAILIMAGE, t0.ZIMAGE, t0.ZTYPE FROM ZRECIPE t0 WHERE  NOT ( t0.Z_PK
IN (SELECT n1_t0.Z_PK FROM ZRECIPE n1_t0 GROUP BY n1_t0.ZSOURCE,
n1_t0.ZEXTERNALID ));
sele  order          from  deta
----  -------------  ----  ----
0     0              0     SCAN TABLE ZRECIPE AS t0
0     0              0     EXECUTE LIST SUBQUERY 1
1     0              0     SCAN TABLE ZRECIPE AS n1_t0
1     0              0     USE TEMP B-TREE FOR GROUP BY
```

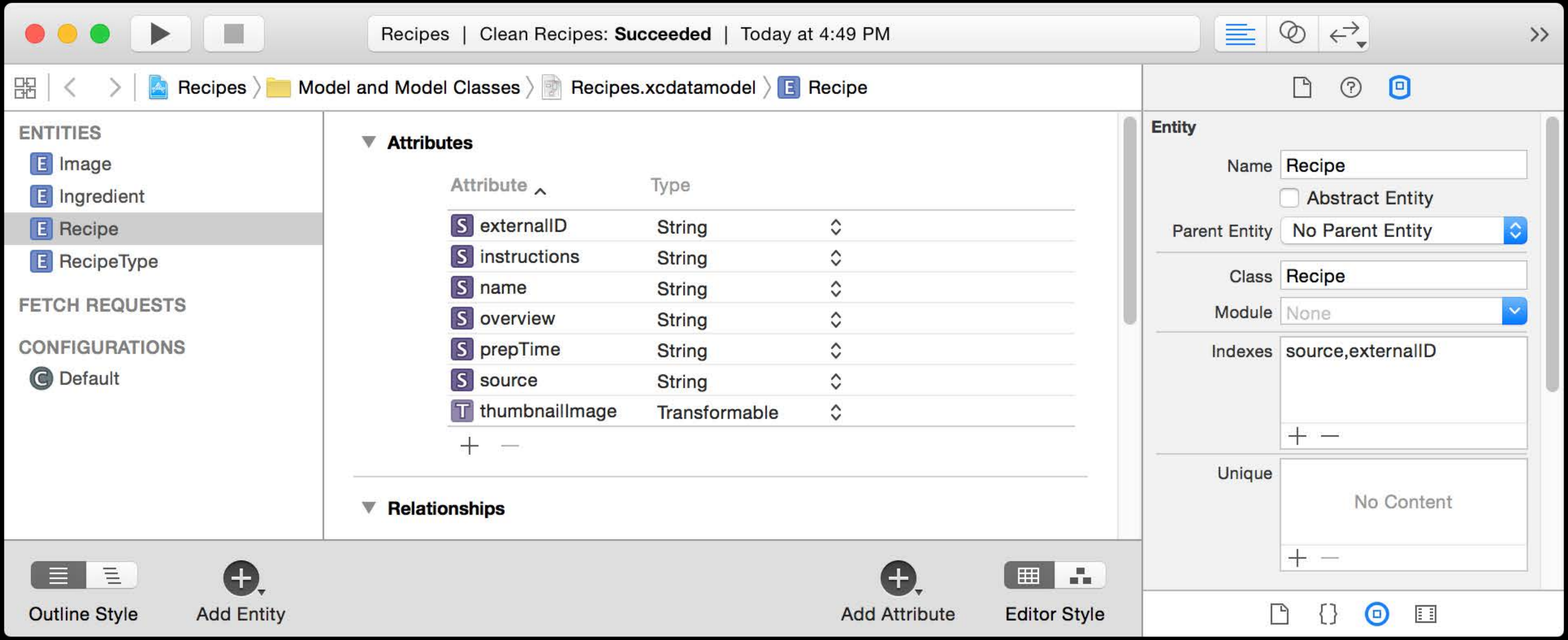
Complex Fetches

EXPLAIN QUERY PLAN

```
sqlite> EXPLAIN QUERY PLAN SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZEXTERNALID,
t0.ZINSTRUCTIONS, t0.ZNAME, t0.ZOVERVIEW, t0.ZPREPTIME, t0.ZSOURCE,
t0.ZTHUMBNAILIMAGE, t0.ZIMAGE, t0.ZTYPE FROM ZRECIPE t0 WHERE  NOT ( t0.Z_PK
IN (SELECT n1_t0.Z_PK FROM ZRECIPE n1_t0 GROUP BY n1_t0.ZSOURCE,
n1_t0.ZEXTERNALID ));
sele  order          from  deta
----  -------------  ----  ----
0     0              0     SCAN TABLE ZRECIPE AS t0
0     0              0     EXECUTE LIST SUBQUERY 1
1     0              0     SCAN TABLE ZRECIPE AS n1_t0
1     0              0     USE TEMP B-TREE FOR GROUP BY
```

Complex Fetches

Large fetches benefit from indexes



Recipes | Clean Recipes: **Succeeded** | Today at 4:49 PM

Recipes 〉 Model and Model Classes 〉 Recipes.xcdatamodel 〉 Recipe

ENTITIES
- Image
- Ingredient
- Recipe
- RecipeType

FETCH REQUESTS

CONFIGURATIONS
- Default

Attributes

Attribute	Type
externalID	String
instructions	String
name	String
overview	String
prepTime	String
source	String
thumbnailImage	Transformable

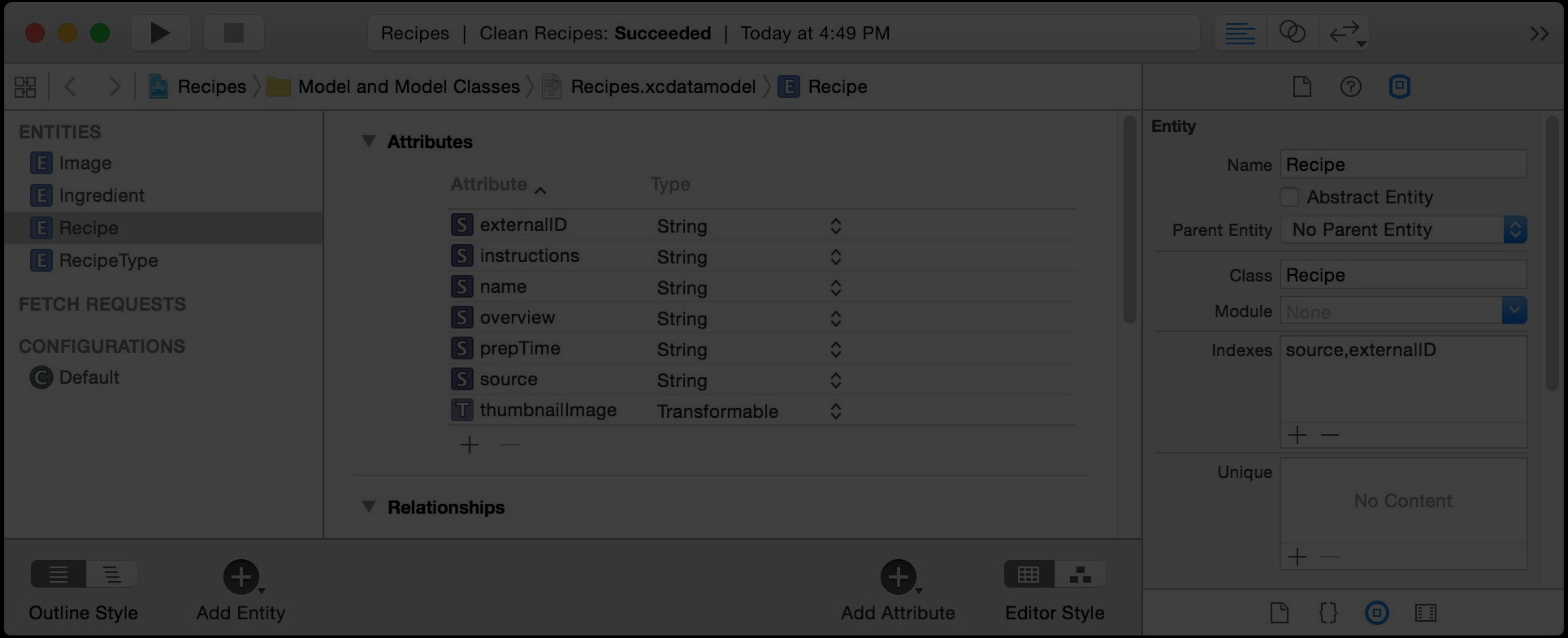
Relationships

Entity

Name	Recipe
	Abstract Entity
Parent Entity	No Parent Entity
Class	Recipe
Module	None
Indexes	source,externalID
Unique	No Content

Complex Fetches

Large fetches benefit from indexes



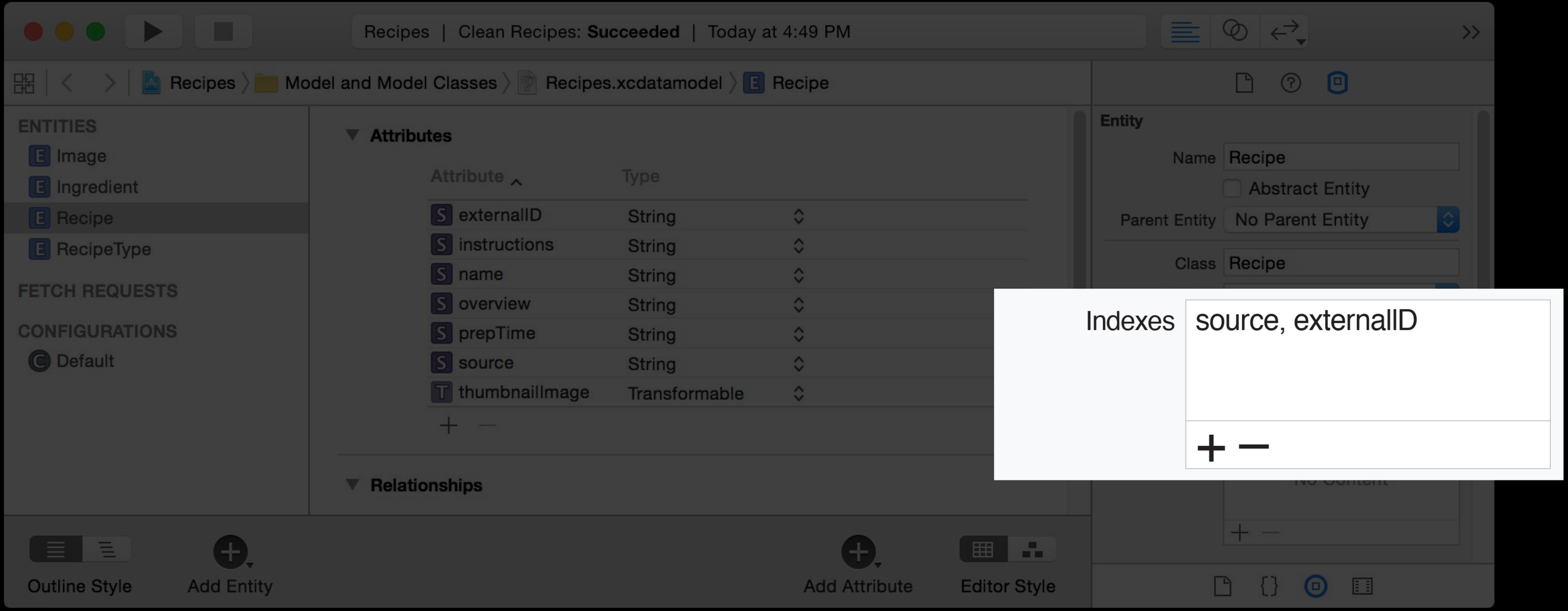
Attribute	Type
externalID	String
instructions	String
name	String
overview	String
prepTime	String
source	String
thumbnailImage	Transformable

Entity

Name	Recipe
	Abstract Entity
Parent Entity	No Parent Entity
Class	Recipe
Module	None
Indexes	source,externalID
Unique	No Content

Complex Fetches

Large fetches benefit from indexes



Recipes | Clean Recipes: **Succeeded** | Today at 4:49 PM

Recipes 〉 Model and Model Classes 〉 Recipes.xcdatamodel 〉 Recipe

Attributes

Attribute	Type
externalID	String
instructions	String
name	String
overview	String
prepTime	String
source	String
thumbnailImage	Transformable

Entity

Name: Recipe

Abstract Entity

Parent Entity: No Parent Entity

Class: Recipe

Indexes: source, externalID

Complex Fetches

Verify a better plan

```
sqlite> EXPLAIN QUERY PLAN SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZEXTERNALID,
t0.ZINSTRUCTIONS, t0.ZNAME, t0.ZOVERVIEW, t0.ZPREPTIME, t0.ZSOURCE,
t0.ZTHUMBNAILIMAGE, t0.ZIMAGE, t0.ZTYPE FROM ZRECIPE t0 WHERE  NOT ( t0.Z_PK
IN (SELECT n1_t0.Z_PK FROM ZRECIPE n1_t0 GROUP BY n1_t0.ZSOURCE,
n1_t0.ZEXTERNALID ));
sele  order          from  deta
----  -------------  ----  ----
0     0              0     SCAN TABLE ZRECIPE AS t0
0     0              0     EXECUTE LIST SUBQUERY 1
1     0              0     SCAN TABLE ZRECIPE AS n1_t0 USING COVERING INDEX
                           ZRECIPE_ZSOURCE_ZEXTERNALID
```

Complex Fetches

Irreducible complexity

```
sqlite> EXPLAIN QUERY PLAN SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZEXTERNALID,
t0.ZINSTRUCTIONS, t0.ZNAME, t0.ZOVERVIEW, t0.ZPREPTIME, t0.ZSOURCE,
t0.ZTHUMBNAILIMAGE, t0.ZIMAGE, t0.ZTYPE FROM ZRECIPE t0 WHERE  NOT ( t0.Z_PK
IN (SELECT n1_t0.Z_PK FROM ZRECIPE n1_t0 GROUP BY n1_t0.ZSOURCE,
n1_t0.ZEXTERNALID ));
sele  order          from  deta
----  -------------  ----  ----
0     0              0     SCAN TABLE ZRECIPE AS t0
0     0              0     EXECUTE LIST SUBQUERY 1
1     0              0     SCAN TABLE ZRECIPE AS n1_t0 USING COVERING INDEX
                           ZRECIPE_ZSOURCE_ZEXTERNALID
```

Complex Fetches

Irreducible complexity

Get off the main thread

- Private queue context
- NSAsynchronousFetchRequest

Look for Problem Patterns

Relationship faults

- Lots of small queries slow down your app

Large fetches

- Make Core Data do the work

Complex fetches

- Add indices and try more powerful predicates
- Avoid blocking UI threads

http://bugreport.apple.com

Bugs
- Sample app bonus

Feature requests

Enhancement ideas

Performance issues
- Sample store bonus

Documentation improvements

More Information

Developer Portal
developer.apple.com

Documentation and Sample Code
developer.apple.com/library

Developer Forums
developer.apple.com/forums

Developer Technical Support
developer.apple.com/support/technical

Related Labs

Core Data Lab	Frameworks Lab C	Thursday 3:30PM
Core Data Lab	Frameworks Lab E	Friday 10:00AM

WWDC15