Core Data Performance

Optimization and Debugging

Session 211

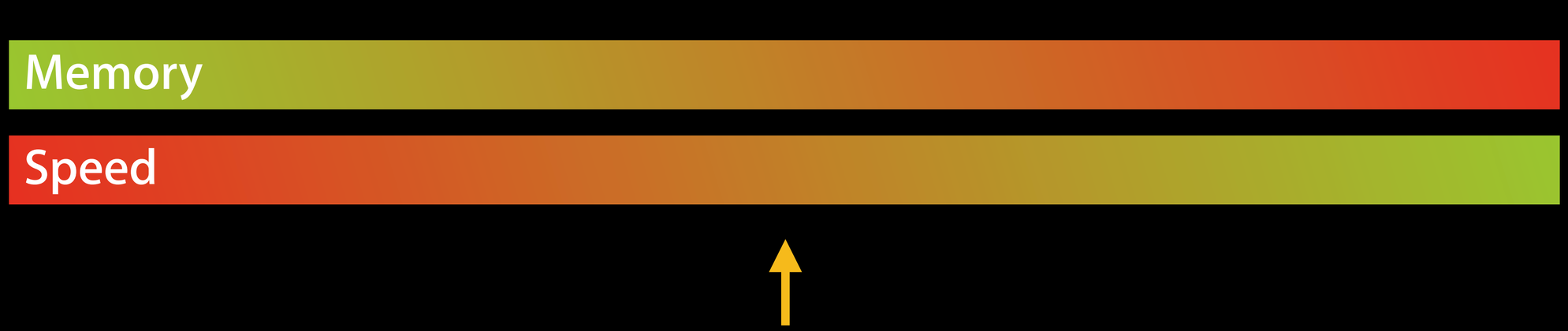
Tim Isted
Software Engineer

Introduction

- Optimization is a balance
  - Minimize memory usage
  - Maximize speed

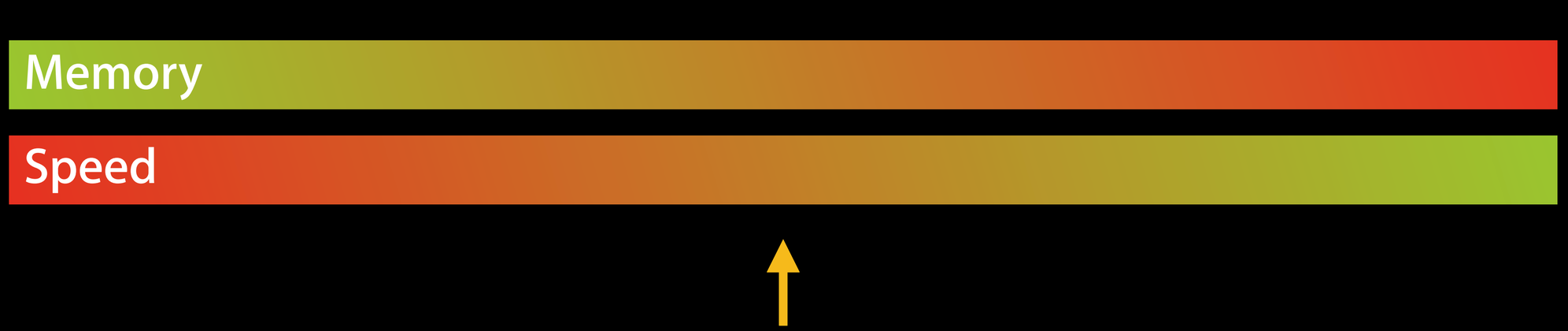
Introduction

- Optimization is a balance
  - Minimize memory usage
  - Maximize speed



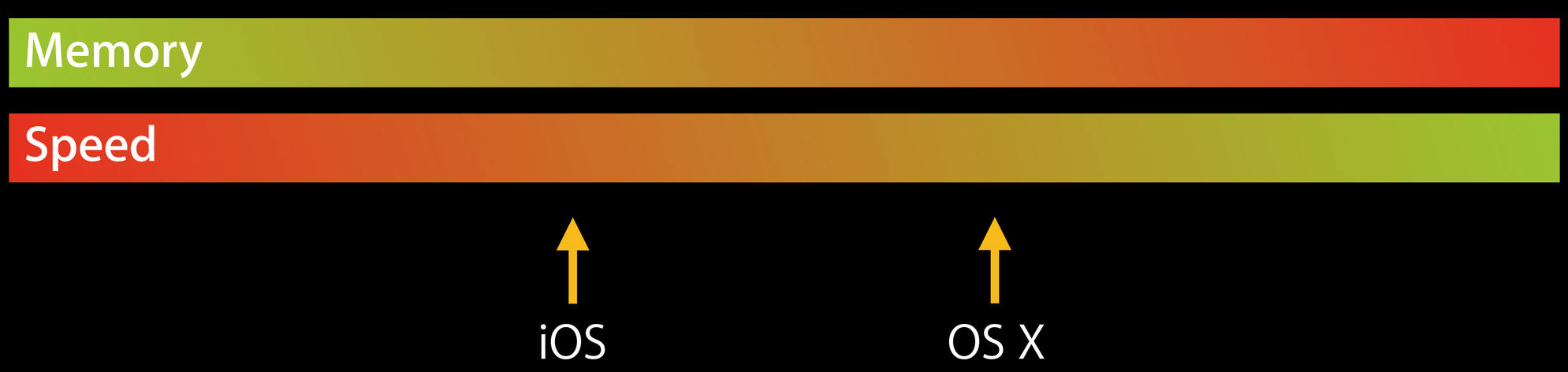
Introduction

- Optimization is a balance
  - Minimize memory usage
  - Maximize speed



Introduction

- Optimization is a balance
  - Minimize memory usage
  - Maximize speed



Performance Pitfalls

- Loading too much
- Firing many faults
- Frequent cache misses
- Expensive queries
- Incurring too many locks

What You Will Learn

- Tools
  - Instruments
  - Debug logging
- Optimizing models, fetches, and predicates
- Choices for concurrency
- Optimizing text searching

Measuring Performance

Measuring Performance

- Instruments
  - What are you looking for?
  - How long should it take?
- Interpreting the results
  - Cache misses?
  - Fetches?
  - Faults firing?
  - Saves?
  - Memory usage?

Measuring Performance

- Instruments
  - What are you looking for?
  - How long should it take?
- Interpreting the results
  - Cache misses?
  - Fetches?
  - Faults firing?
  - Saves?
  - Memory usage?

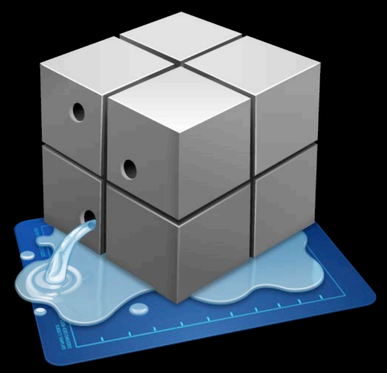
Core Data

Measuring Performance

- Instruments
  - What are you looking for?
  - How long should it take?
- Interpreting the results
  - Cache misses?
  - Fetches?
  - Faults firing?
  - Saves?
  - Memory usage?

Core Data

Time Profiler



Allocations

File Activity

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Tim Isted

MyContacts.xcodeproj — MyContacts.xcdatamodel

MyContacts › iPhone

Finished running MyContacts on iPhone — No Issues

MyContacts › MyContacts › MyContacts.xcdatamodeld › MyContacts.xcdatamodel › Contact

ENTITIES

Contact

FETCH REQUESTS

CONFIGURATIONS

Default

Contact
▼ Attributes
firstName
lastName
photo
▼ Relationships

No Debug Session

Outline Style

Add Entity

Add Attribute

Editor Style

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Tim Isted

MyContacts.xcodeproj

MyContacts › iPhone

Running MyContacts on iPhone No Issues

MyContacts
PID 1334, Running

CPU 0%

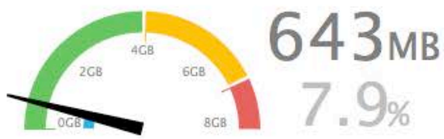
Memory 643.3 MB

Memory Report

Memory

Profile In Instruments

Memory Utilized

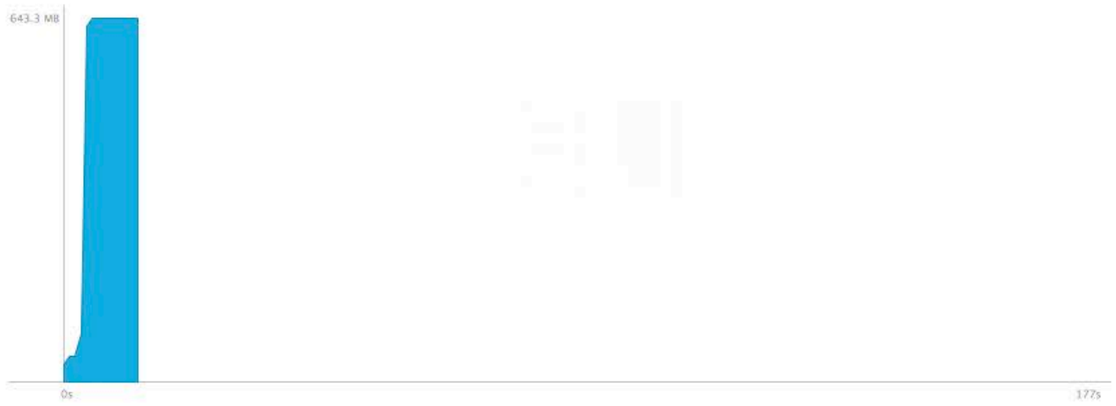


643MB
7.9%

Utilization Comparison

MyContacts
643.3 MB
Other Processes
4.91 GB
Free
2.46 GB

Memory
Duration: 12 sec
High: 643.3 MB
Low: 31.6 MB



MyContacts

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Tim Isted

MyContacts.xcodeproj

MyContacts › iPhone

Running MyContacts on iPhone | No Issues

Memory Report

MyContacts
PID 1334, Running

CPU 0%

Memory 643.3 MB

Memory

Profile In Instruments

Memory Utilized

643 MB
7.9%

0GB 2GB 4GB 6GB 8GB

Utilization Comparison

MyContacts
643.3 MB

Other Processes
4.91 GB

Free
2.46 GB

Memory
Duration: 12 sec
High: 643.3 MB
Low: 31.6 MB

643.3 MB

0s 177s

MyContacts

Instruments File Edit View Instrument Window Help

Tim Isted

MyContacts.xcodeproj — MyContacts.xcdatamodel

MyContacts › iPhone

Instruments

Record | Target: MyContacts | Inspection Range | 00:00:10 Run 1 of 1 | View | Library | Filter: Recorded Data

ENTITIES
Contact

FETCH REQUESTS

CONFIGURATIONS
Default

No Debug Session

Instruments

Core Data Fetches — Fetch count, Fetch duration

Core Data Cache... — RCM duration, ICM duration

Core Data Saves

Core Data Fetches › Event List › Core Data Fetches

Call Tree
- Separate by Thread
- Invert Call Tree
- Hide Missing Symbols
- Hide System Libraries
- Show Obj-C Only
- Flatten Recursion

Call Tree Constraints

Specific Data Mining

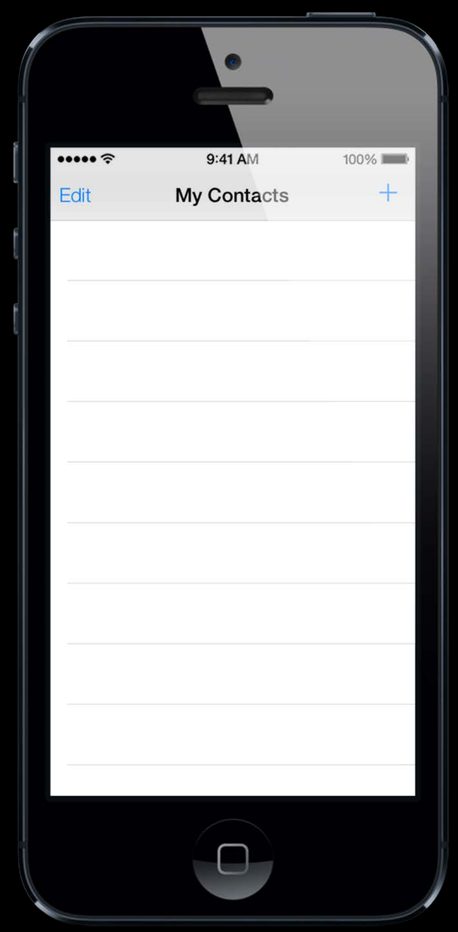
#	Caller	Fetch entity	Fetch count	Fetch duration
0	-[NSManagedObjectContext exe...	Contact		
1	-[NSManagedObjectContext exe...		505	3143116

Outline Style | Add Entity | Add Attribute | Editor Style

Optimizing Fetch Requests

Don’t Fetch More than You Need

- Only 10 or so rows are visible
- Don’t fetch every possible object
- Use a Fetch Batch Size of 20



Don’t Fetch More Than You Need

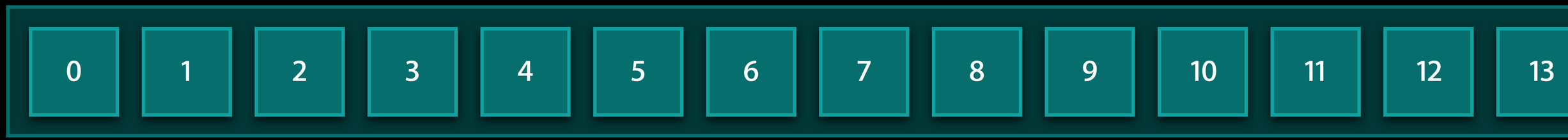
- Set a Fetch Batch Size

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.fetchBatchSize = 20;
```

Don’t Fetch More Than You Need

- Set a Fetch Batch Size

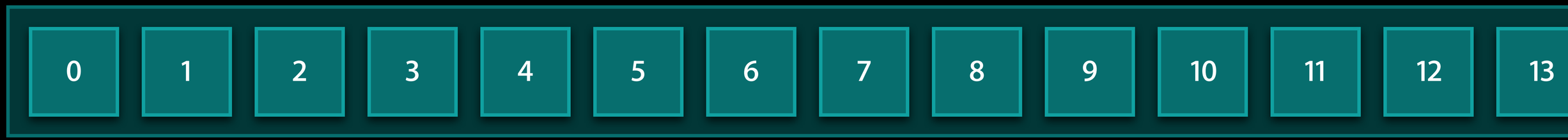
```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.fetchBatchSize = 20;
```



Don’t Fetch More Than You Need

- Set a Fetch Batch Size

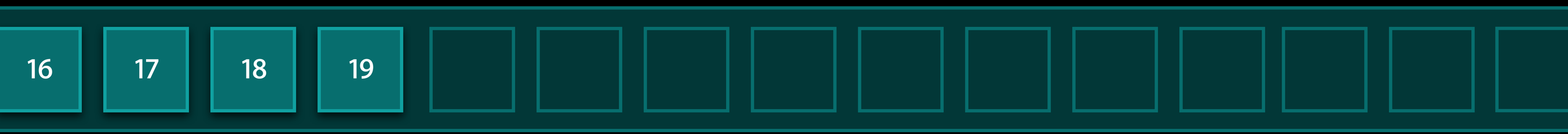
```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.fetchBatchSize = 20;
```



Don’t Fetch More Than You Need

- Set a Fetch Batch Size

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.fetchBatchSize = 20;
```



Don’t Fetch More Than You Need

- Set a Fetch Batch Size

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.fetchBatchSize = 20;
```

Don’t Fetch More Than You Need

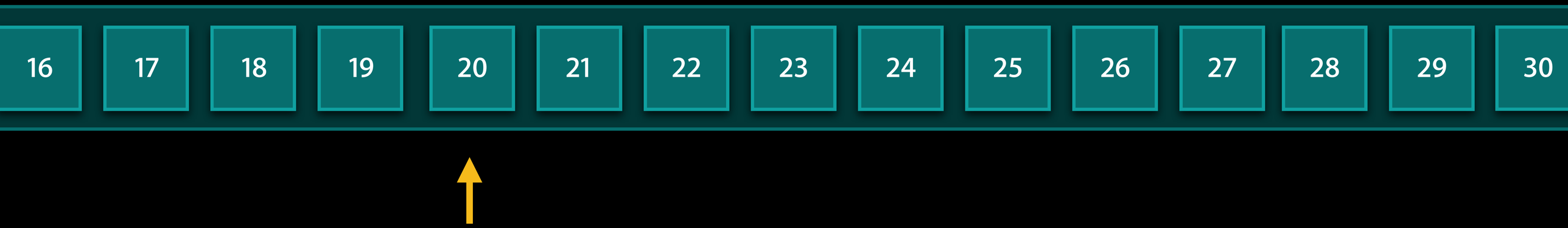
- Set a Fetch Batch Size

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.fetchBatchSize = 20;
```

Don’t Fetch More Than You Need

- Set a Fetch Batch Size

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.fetchBatchSize = 20;
```



MyContacts.xcodeproj — APLMasterViewController.m

MyContacts › iPhone

Finished running MyContacts on iPhone — No Issues

MyContacts › MyContacts › APLMasterViewController.m › -fetchStoredContacts

No Debug Session

```
39
40  - (void)fetchStoredContacts {
41      NSFetchRequest *fetchRequest = [[NSFetchRequest alloc] init]
        ;
42      // Edit the entity name as appropriate.
43      NSEntityDescription *entity = [NSEntityDescription
        entityForName:@"Contact" inManagedObjectContext:self.
        managedObjectContext];
44      [fetchRequest setEntity:entity];
45
46      // Set the batch size to a suitable number.
47      [fetchRequest setFetchBatchSize:0];
48
49      [fetchRequest setSortDescriptors:@[[NSSortDescriptor
        sortDescriptorWithKey:@"lastName" ascending:YES],
        [NSSortDescriptor sortDescriptorWithKey:@"firstName"
        ascending:YES]]];
50
51      NSError *anyError = nil;
52      NSArray *results = [self.managedObjectContext
        executeFetchRequest:fetchRequest error:&anyError];
53      if (!results) {
54          NSLog(@"Error: %@", anyError);
55      }
```

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Tim Isted

MyContacts.xcodeproj

MyContacts › iPhone

Running MyContacts on iPhone | No Issues

Memory Report

MyContacts
PID 3966, Running

CPU 0%

Memory 62.5 MB

Memory

Profile In Instruments

Memory Utilized

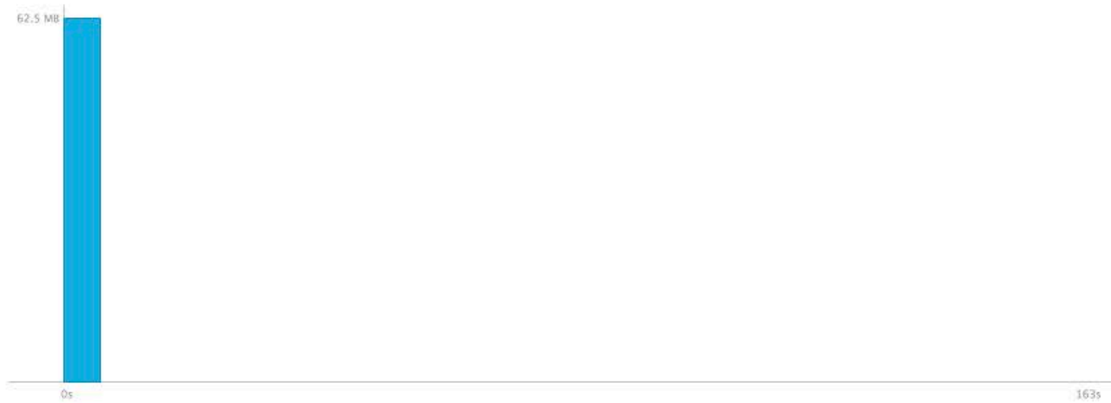
62.5 MB
0.76%

Utilization Comparison

- MyContacts
  62.5 MB
- Other Processes
  5.43 GB
- Free
  2.51 GB

Memory

Duration: 5 sec
High: 62.5 MB
Low: 62.5 MB



MyContacts

Instruments File Edit View Instrument Window Help

Tim Isted

MyContacts.xcodeproj — APLMasterViewController.m

MyContacts › iPhone

No Debug Session

Instruments

Stop | Target: MyContacts | Inspection Range | 00:00:04 Run 1 of 1 | View | Library | Filter: Recorded Data

Instruments

Core Data Fetches

Fetch count

Fetch duration

Core Data Cache...

RCM duration

CM duration

Core Data Saves

Core Data Fetches | Event List | Core Data Fetches

Call Tree

Separate by Thread

Invert Call Tree

Hide Missing Symbols

Hide System Libraries

Show Obj-C Only

Flatten Recursion

Call Tree Constraints

Specific Data Mining

#	Caller	Fetch entity	Fetch count	Fetch duration
0	-[NSManagedObjectContext exe...	Contact		
1	-[NSManagedObjectContext exe...		505	2982
2	-[NSManagedObjectContext exe...	Contact		
3	-[NSManagedObjectContext exe...		20	176772

```
executeFetchRequest:fetchRequest error:&anyError];
 if (!results) {
     NSLog(@"Error: %@", anyError);
 }
```

Instruments File Edit View Instrument Window Help

Tim Isted

MyContacts.xcodeproj — APLMasterViewController.m

MyContacts › iPhone

No Debug Session

Instruments

Record | Target: MyContacts | Inspection Range | 00:00:14 Run 1 of 1 | View | Library | Filter: Recorded Data

Instruments

Core Data Fetches — Fetch count, Fetch duration

Core Data Cache... — RCM duration, ICM duration

Core Data Saves

Core Data Fetches › Event List › Core Data Fetches

Call Tree
- Separate by Thread
- Invert Call Tree
- Hide Missing Symbols
- Hide System Libraries
- Show Obj-C Only
- Flatten Recursion

Call Tree Constraints

Specific Data Mining

#	Caller	Fetch entity	Fetch count	Fetch duration
0	-[NSManagedObjectContext exe...	Contact		
1	-[NSManagedObjectContext exe...		505	2982
2	-[NSManagedObjectContext exe...	Contact		
3	-[NSManagedObjectContext exe...		20	176772
4	-[NSManagedObjectContext exe...	Contact		
5	-[NSManagedObjectContext exe...		20	165388
6	-[NSManagedObjectContext exe...	Contact		
7	-[NSManagedObjectContext exe...		20	163723
8	-[NSManagedObjectContext exe...	Contact		
9	-[NSManagedObjectContext exe...		20	162572
10	-[NSManagedObjectContext exe...	Contact		
11	-[NSManagedObjectContext exe...		20	199429

```
executeFetchRequest:fetchRequest error:&anyError];
 if (!results) {
     NSLog(@"Error: %@", anyError);
 }
```

Optimizing the Data Model

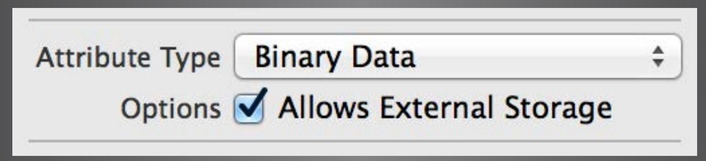
Optimizing the Data Model

Design the model for your app's usage

- Don't overnormalize
- Duplication isn't necessarily a bad thing

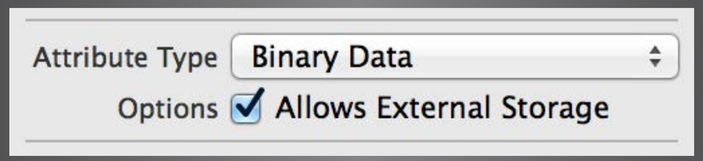
Data and External Files

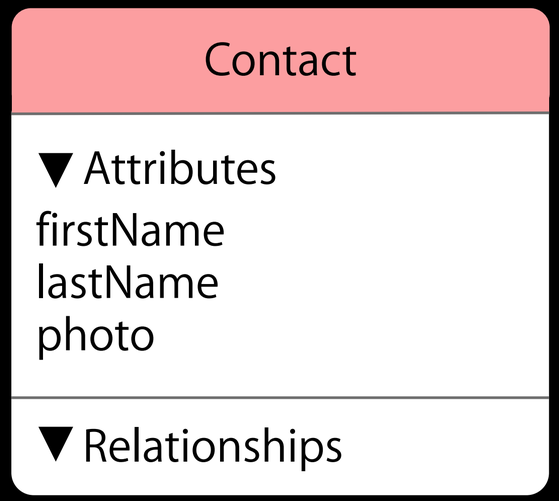
- Use external storage



Data and External Files

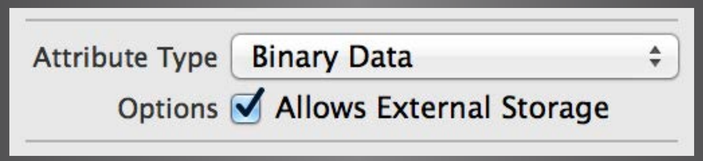
- Use external storage



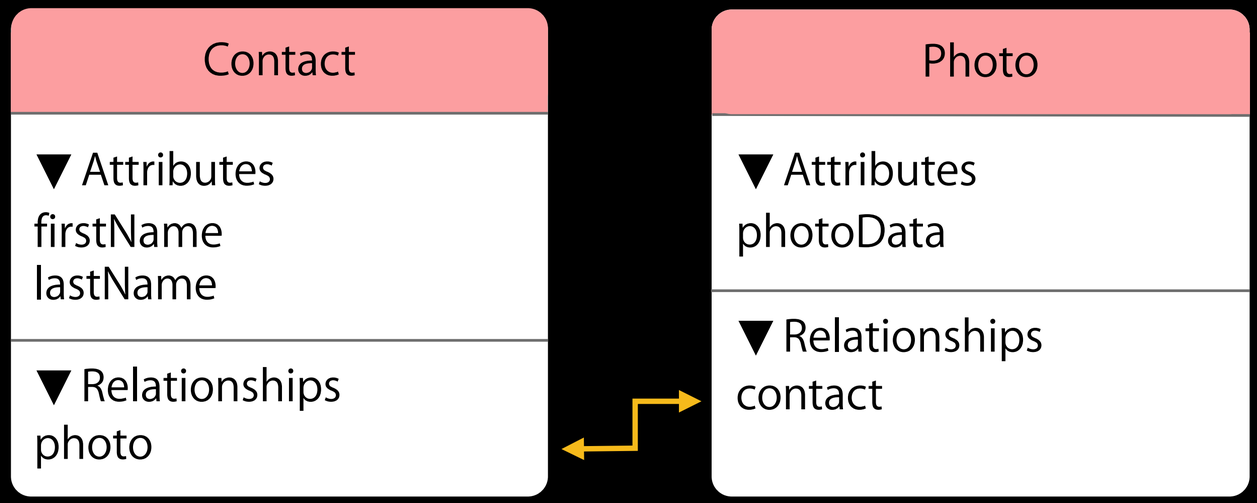


Data and External Files

- Use external storage



- Put binary data in a separate entity



Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Tim Isted

MyContacts.xcodeproj — MyContacts.xcdatamodel

MyContacts › iPhone

Finished running – Profiling MyContacts.app on iPhone — No Issues

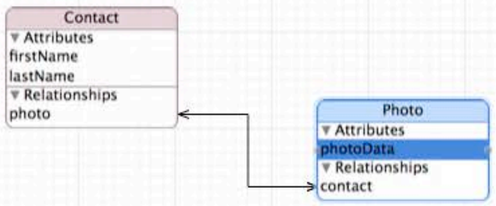
MyContacts › MyContacts › MyContacts.xcdatamodeld › MyContacts.xcdatamodel › Photo › photoData

ENTITIES
- Contact
- Photo

FETCH REQUESTS

CONFIGURATIONS
- Default

No Debug Session



Attribute

Name: photoData

Properties: Transient, Optional, Indexed

Attribute Type: Binary Data

Options: Allows External Storage

Advanced: Index in Spotlight, Store in External Record File

User Info

Key | Value

Versioning

Hash Modifier: Version Hash Modifier

Renaming ID: Renaming Identifier

Outline Style | Add Entity | Add Attribute | Editor Style

Fetching Related Objects

Prefetch relationships if you know you need them

- Set relationship key paths for prefetching:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

[request setRelationshipKeyPathsForPrefetching:@[ @"photo" ]];
```

Optimizing the Model

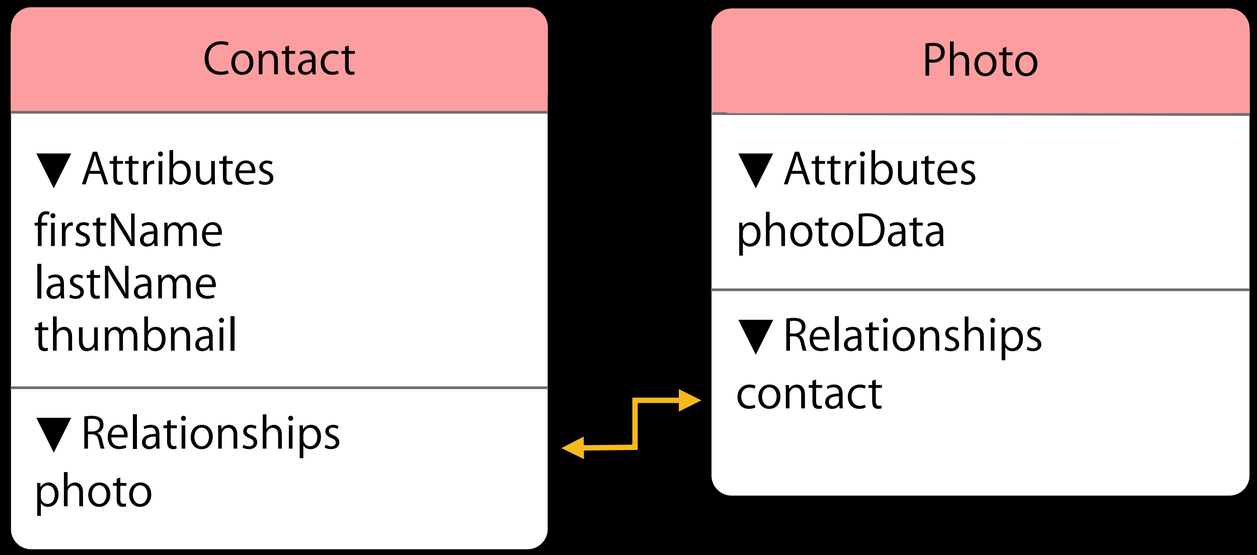
Don’t store more than you need

- Don’t store a 10MB image just to show a tiny thumbnail
- Cache the thumbnail separately
- Less data takes less time to fetch

Optimizing the Model

Don’t store more than you need

- Don’t store a 10MB image just to show a tiny thumbnail
- Cache the thumbnail separately
- Less data takes less time to fetch



Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Tim Isted

MyContacts.xcodeproj — MyContacts.xcdatamodel

MyContacts 〉 iPhone

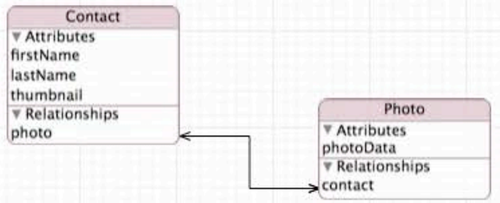
Finished running MyContacts on iPhone — No Issues

MyContacts 〉 MyContacts 〉 MyContacts.xcdatamodeld 〉 MyContacts.xcdatamodel 〉 No Selection

ENTITIES
- Contact
- Photo

FETCH REQUESTS

CONFIGURATIONS
- Default



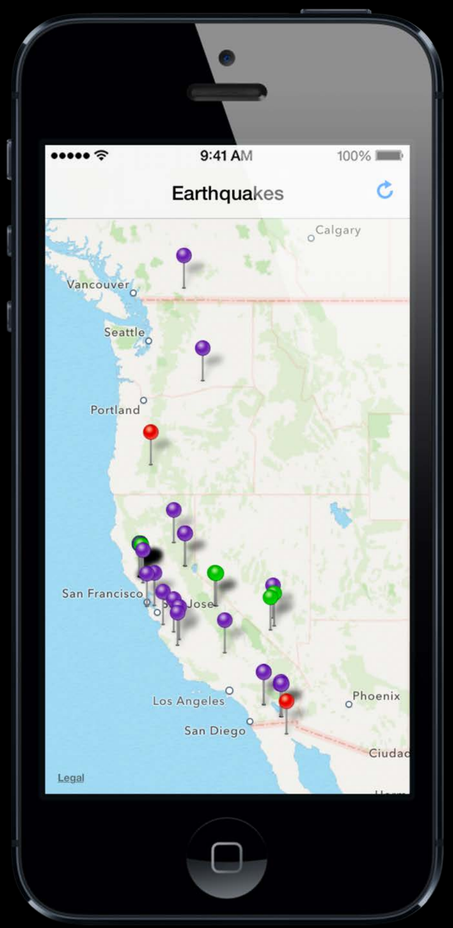
No Debug Session

No Selection

Outline Style — Add Entity — Add Attribute — Editor Style

Performing Background Tasks

Implementing Update or Insert



Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Tim Isted

Earthquakes.xcodeproj — Earthquakes.xcdatamodel

Earthquakes › iPhone

Finished running Earthquakes on iPhone — No Issues

Earthquakes › Earthquakes › Earthquakes.xcdatamodeld › Earthquakes.xcdatamodel › Quake

No Debug Session

ENTITIES

Quake

FETCH REQUESTS

CONFIGURATIONS

Default

Attributes

Attribute	Type
depth	Decimal
detailURL	String
latitude	Decimal
longitude	Decimal
magnitude	Decimal
placeName	String
time	Date
tsunami	Boolean
uuid	String

Relationships

Relationship	Destination	Inverse

Fetched Properties

Fetched Property	Predicate

Outline Style — Add Entity — Add Attribute — Editor Style

Instruments File Edit View Instrument Window Help

Tim Isted

Earthquakes.xcodeproj — Earthquakes.xcdatamodel

Earthquakes › iPhone

ENTITIES

Quake

FETCH REQUES

CONFIGURATIO

Default

No Debug Session

Instruments

Stop

Target: Earthquakes

Inspection Range

00:00:46

Run 1 of 1

View

Library

Filter: Recorded Data

Instruments

00:00 00:10 00:20 00:30 00:40 00:50 01:00 01:10

Core Data Fetches

Fetch count

Fetch duration

Core Data Cache...

RCM duration

CM duration

Core Data Saves

Core Data Fetches

Event List

Core Data Fetches

Call Tree

Separate by Thread

Invert Call Tree

Hide Missing Symbols

Hide System Libraries

Show Obj-C Only

Flatten Recursion

Call Tree Constraints

Specific Data Mining

#	Caller	Fetch entity	Fetch count	Fetch duration
0	-[NSManagedObjectContext exe...	Quake		
1	-[NSManagedObjectContext exe...		3	3698
2	-[NSManagedObjectContext exe...	Quake		
3	-[NSManagedObjectContext exe...		44	2767
4	-[NSManagedObjectContext exe...	Quake		
5	-[NSManagedObjectContext exe...		1	3933
6	-[NSManagedObjectContext exe...	Quake		
7	-[NSManagedObjectContext exe...		1	2642
8	-[NSManagedObjectContext exe...	Quake		
9	-[NSManagedObjectContext exe...		1	2380
10	-[NSManagedObjectContext exe...	Quake		
11	-[NSManagedObjectContext exe...		1	2759
12	-[NSManagedObjectContext exe...	Quake		
13	-[NSManagedObjectContext exe...		1	2399
14	-[NSManagedObjectContext exe...	Quake		
15	-[NSManagedObjectContext exe...		1	2569
16	-[NSManagedObjectContext exe...	Quake		
17	-[NSManagedObjectContext exe...		1	2254
18	-[NSManagedObjectContext exe...	Quake		
19	-[NSManagedObjectContext exe...		1	3861

Outline Style

Add Entity

Add Attribute

Editor Style

Instruments File Edit View Instrument Window Help

Tim Isted

Earthquakes.xcodeproj — Earthquakes.xcdatamodel

Earthquakes › iPhone

ENTITIES

Quake

FETCH REQUES

CONFIGURATI

Default

No Debug Session

Instruments

Record | Target: Earthquakes | Inspection Range | 00:00:56 Run 1 of 1 | View | Library | Filter: Recorded Data

Instruments

00:00 00:10 00:20 00:30 00:40 00:50 01:00 01:10

Core Data Fetches

Fetch count

Fetch duration

Core Data Cache...

RCM duration

CM duration

Core Data Saves

Core Data Fetches › Event List › Core Data Fetches

Call Tree

Separate by Thread

Invert Call Tree

Hide Missing Symbols

Hide System Libraries

Show Obj-C Only

Flatten Recursion

Call Tree Constraints

Specific Data Mining

#	Caller	Fetch entity	Fetch count	Fetch duration
261	-[NSManagedObjectContext exe...		1	2618
262	-[NSManagedObjectContext exe...	Quake		
263	-[NSManagedObjectContext exe...		1	2429
264	-[NSManagedObjectContext exe...	Quake		
265	-[NSManagedObjectContext exe...		1	2276
266	-[NSManagedObjectContext exe...	Quake		
267	-[NSManagedObjectContext exe...		1	2318
268	-[NSManagedObjectContext exe...	Quake		
269	-[NSManagedObjectContext exe...		1	2321
270	-[NSManagedObjectContext exe...	Quake		
271	-[NSManagedObjectContext exe...		1	2185
272	-[NSManagedObjectContext exe...	Quake		
273	-[NSManagedObjectContext exe...		1	2230
274	-[NSManagedObjectContext exe...	Quake		
275	-[NSManagedObjectContext exe...		1	2255
276	-[NSManagedObjectContext exe...	Quake		
277	-[NSManagedObjectContext exe...		1	2289
278	-[NSManagedObjectContext exe...	Quake		
279	-[NSManagedObjectContext exe...		1	2297
280	-[NSManagedObjectContext exe...	Quake		

Outline Style | Add Entity | Add Attribute | Editor Style

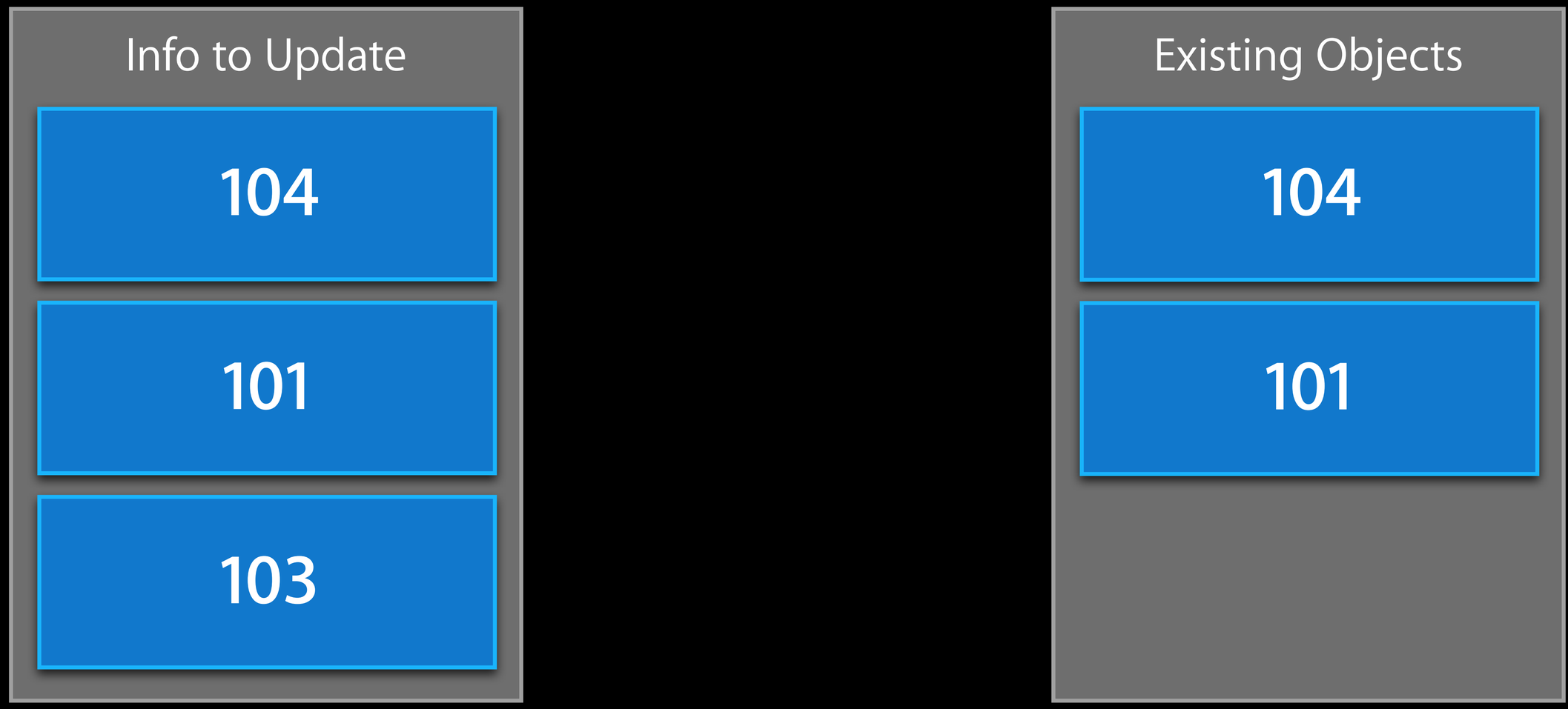
Implementing Update or Insert

- Sort your input objects by ID
- Execute one, sorted fetch request for matching IDs
- Iterate through both input and existing objects collections
  - If IDs match, it's an update
  - If not, it's an insert

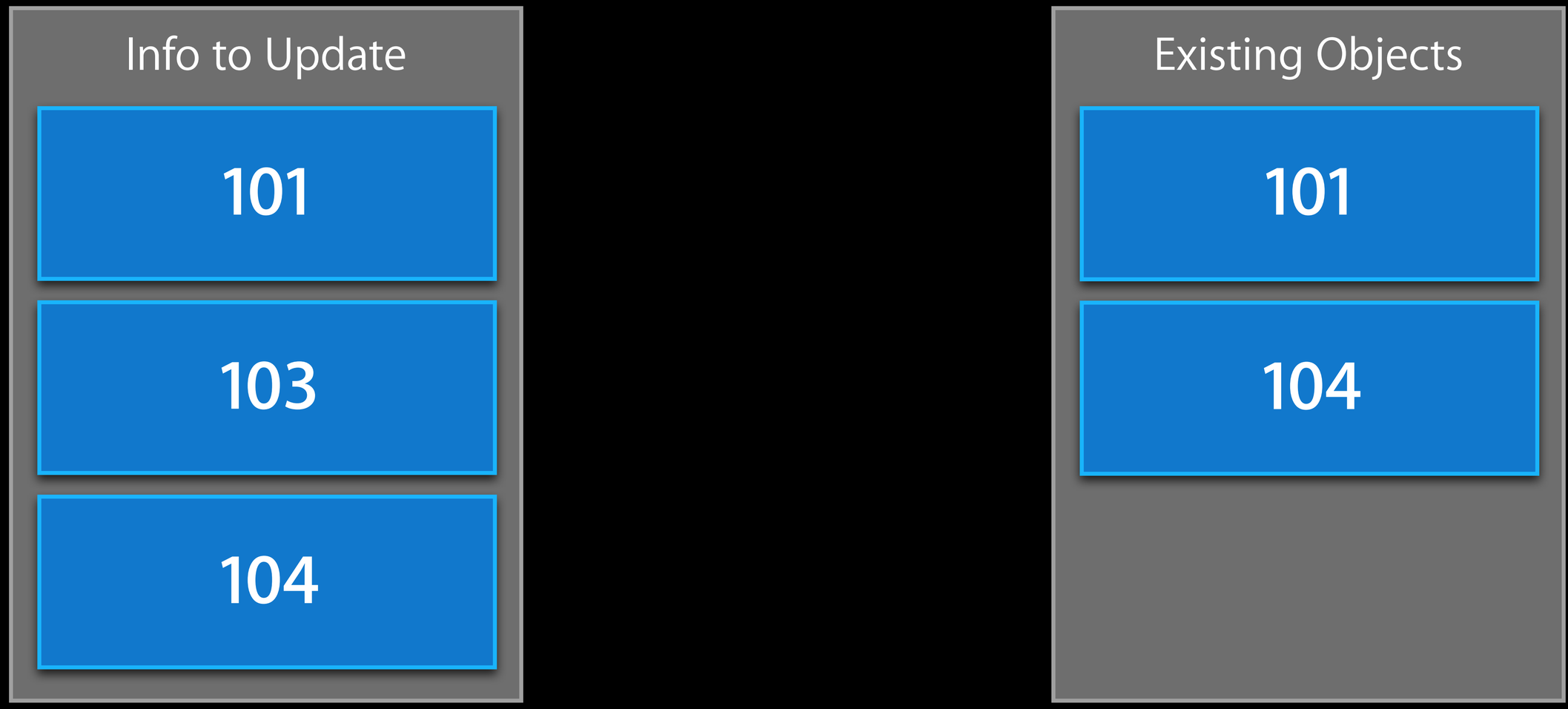
Implementing Update or Insert

Implementing Update or Insert

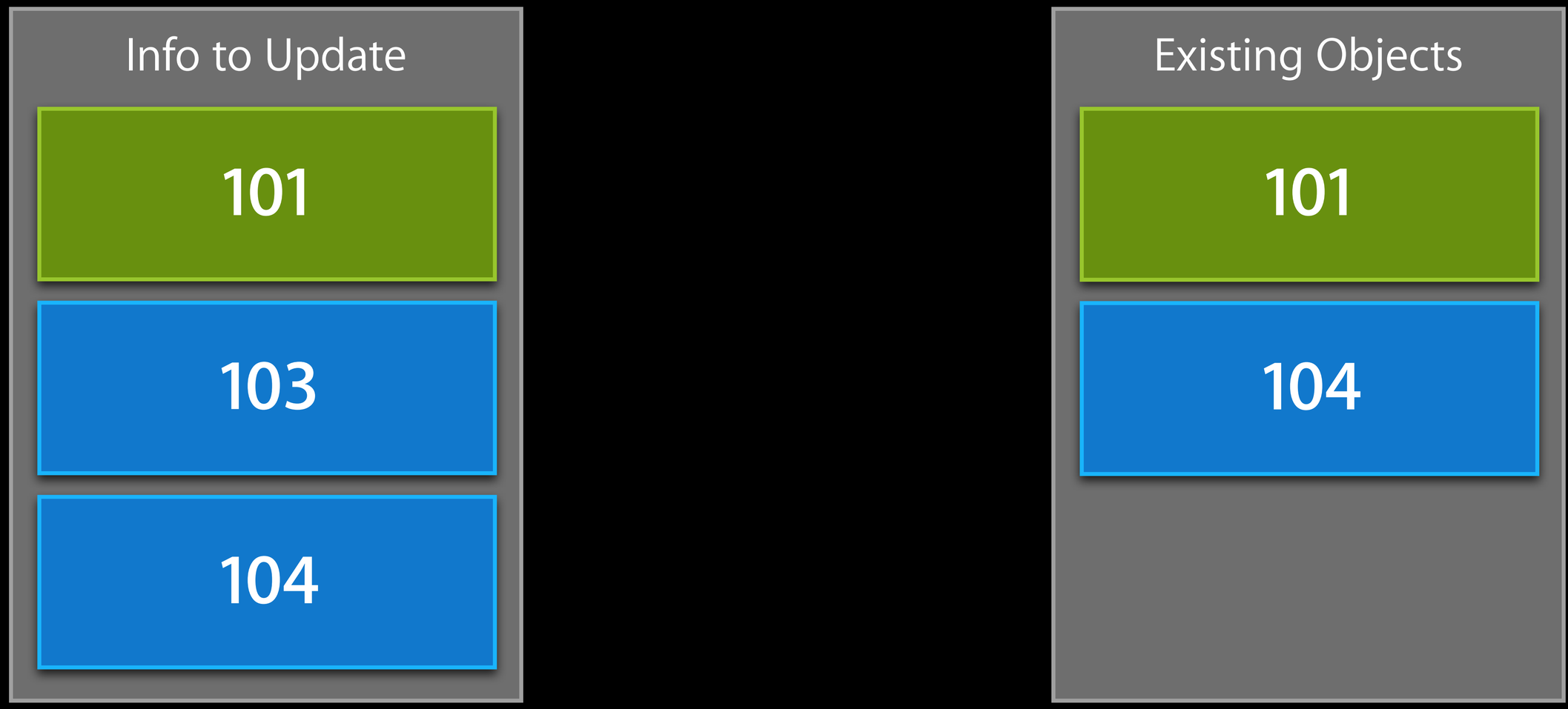
Implementing Update or Insert



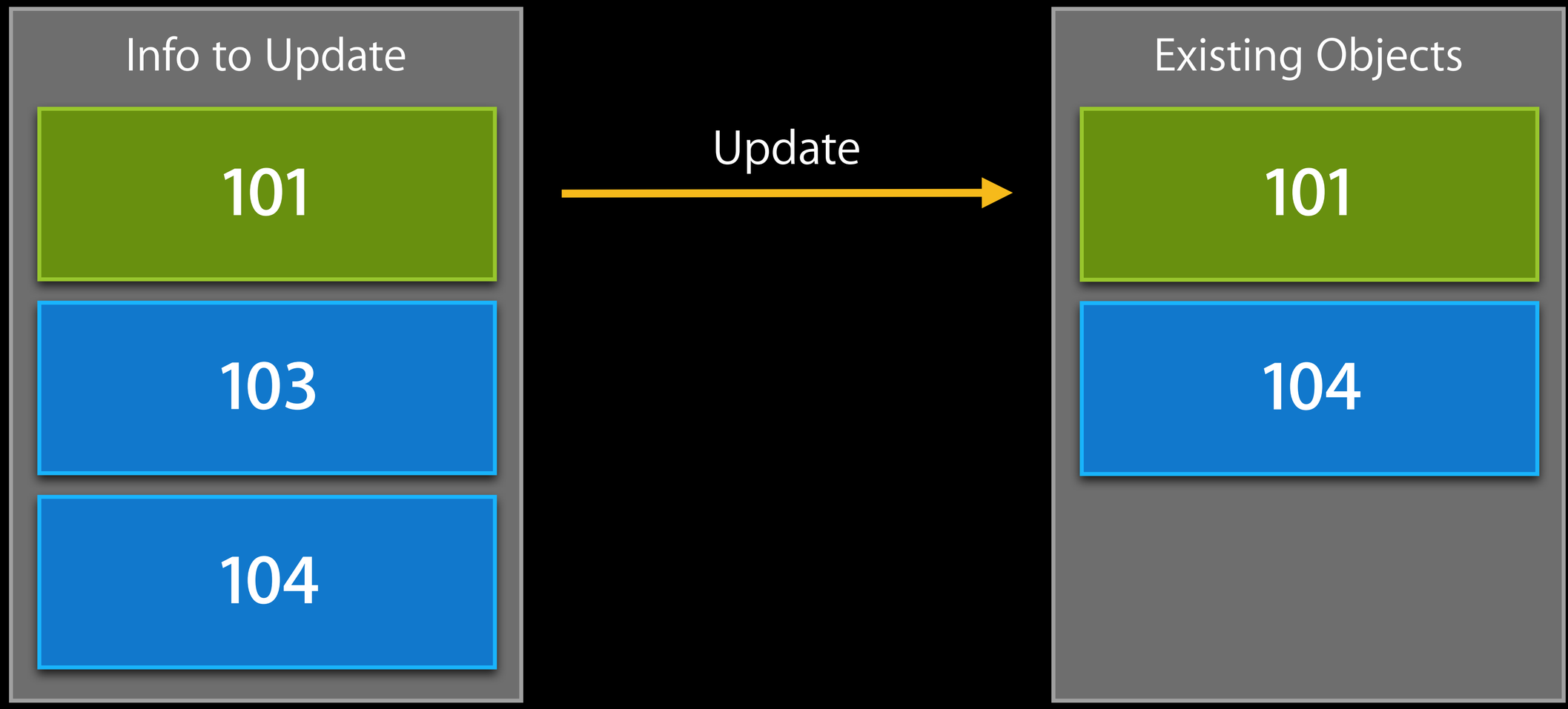
Implementing Update or Insert



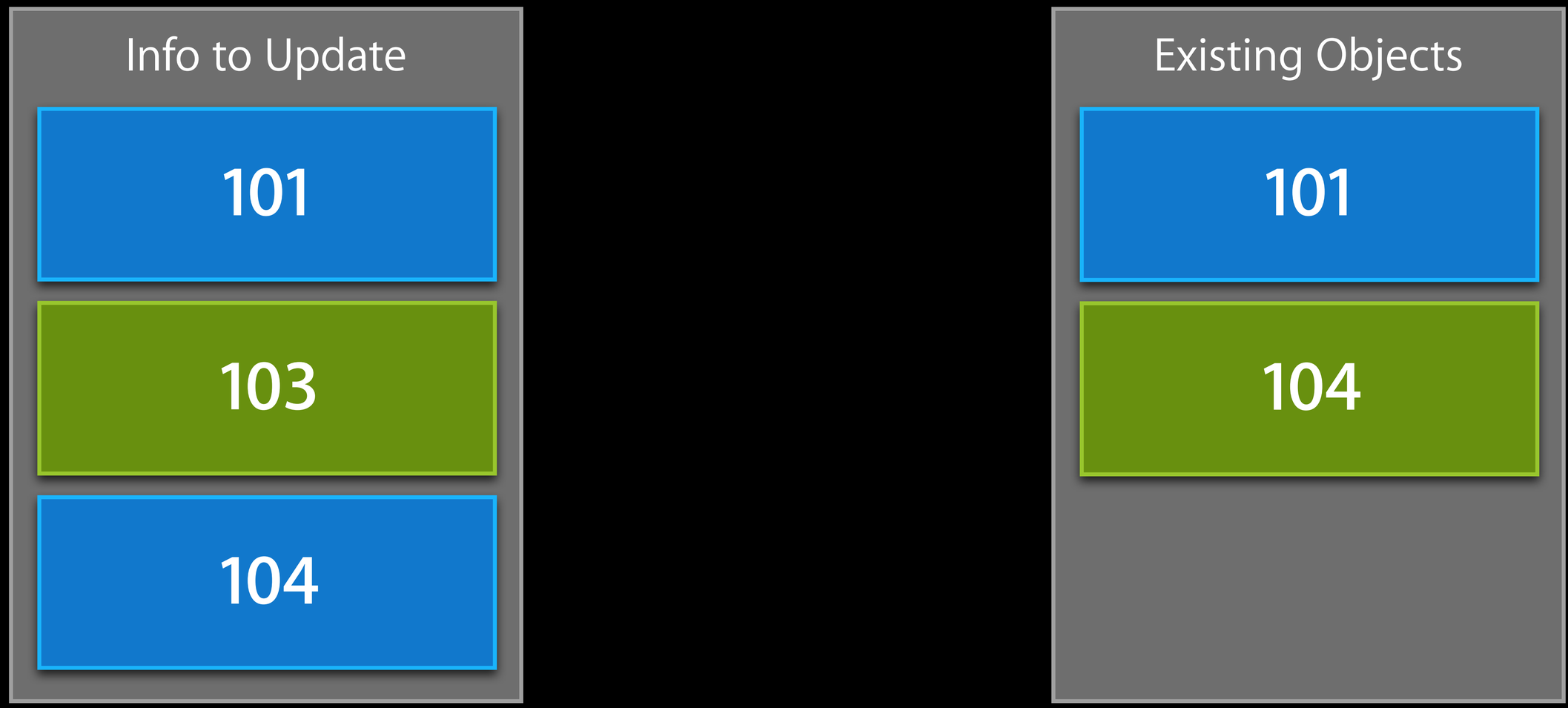
Implementing Update or Insert



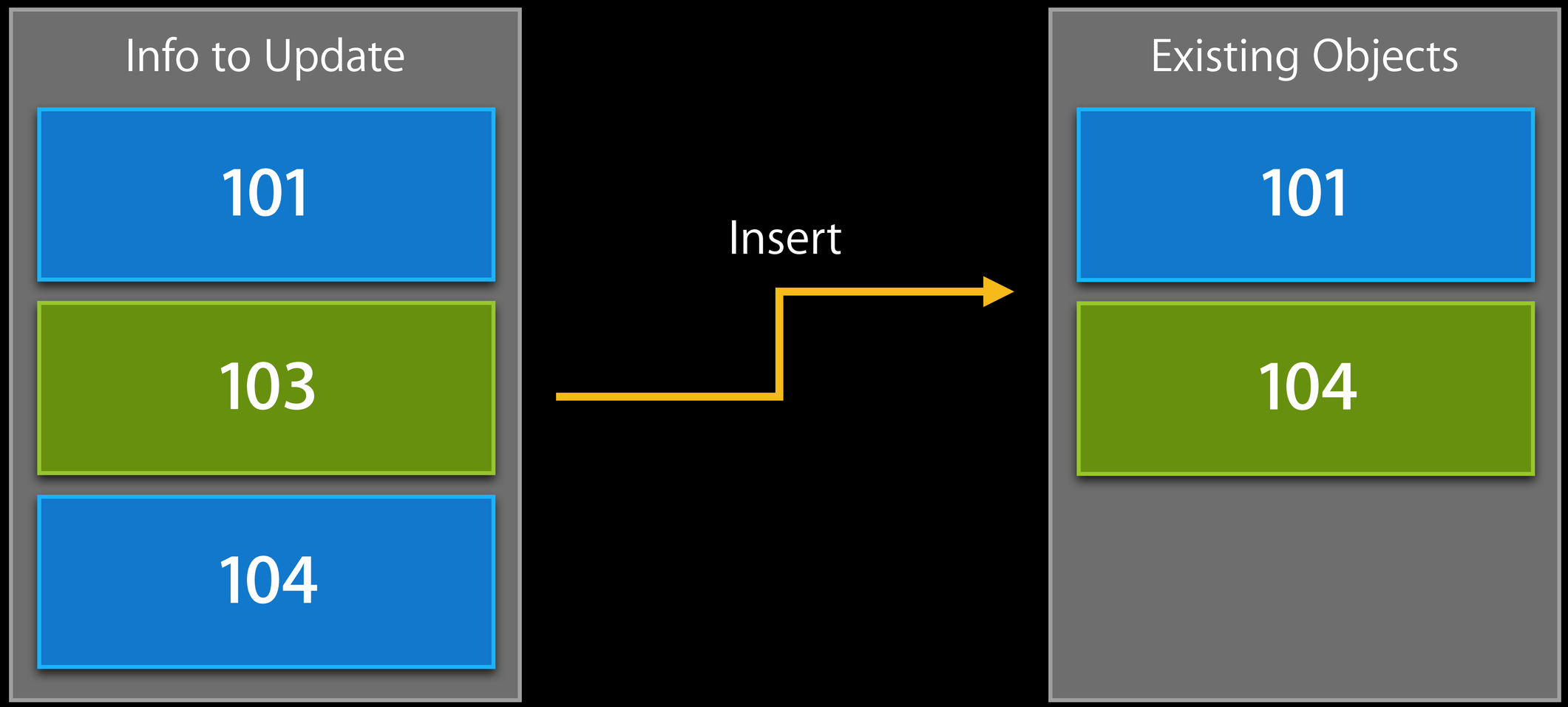
Implementing Update or Insert



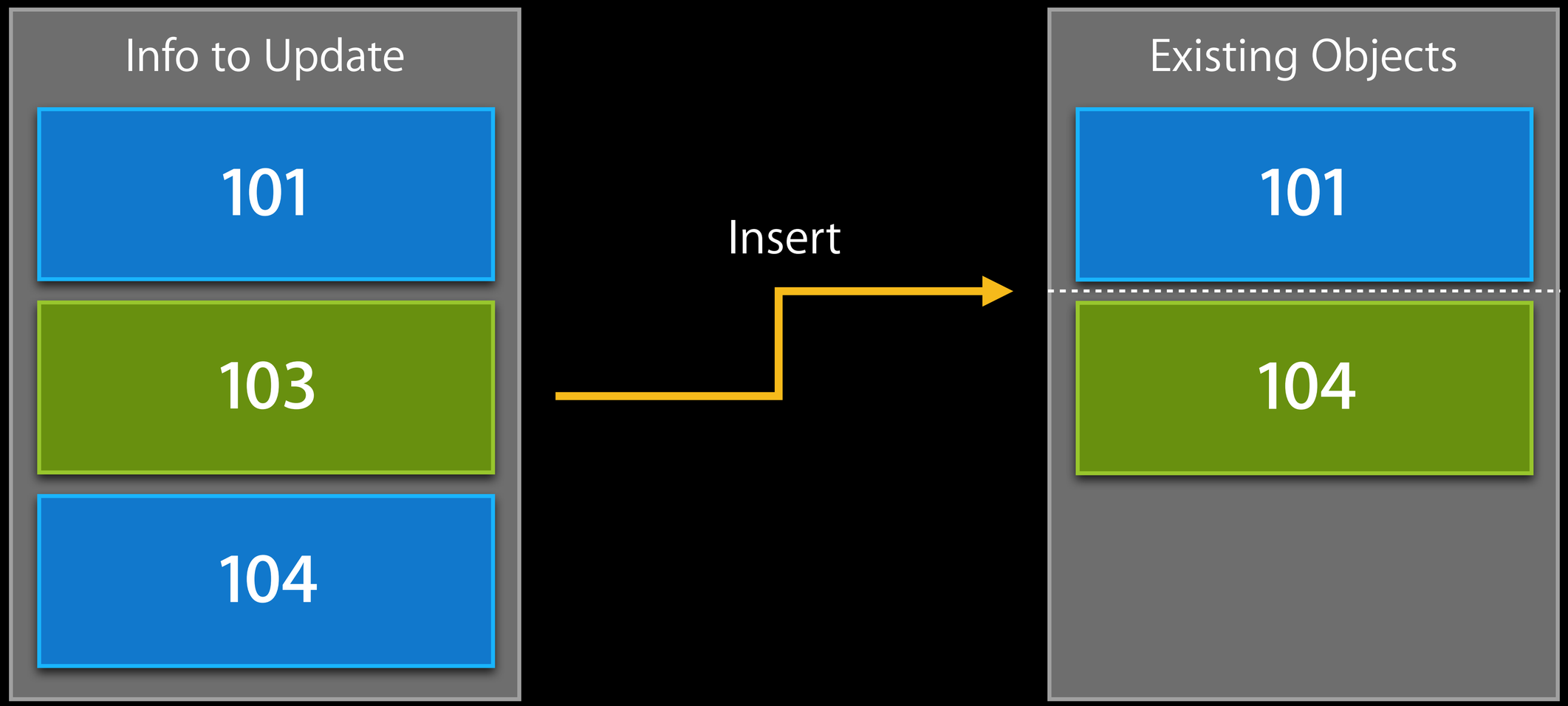
Implementing Update or Insert



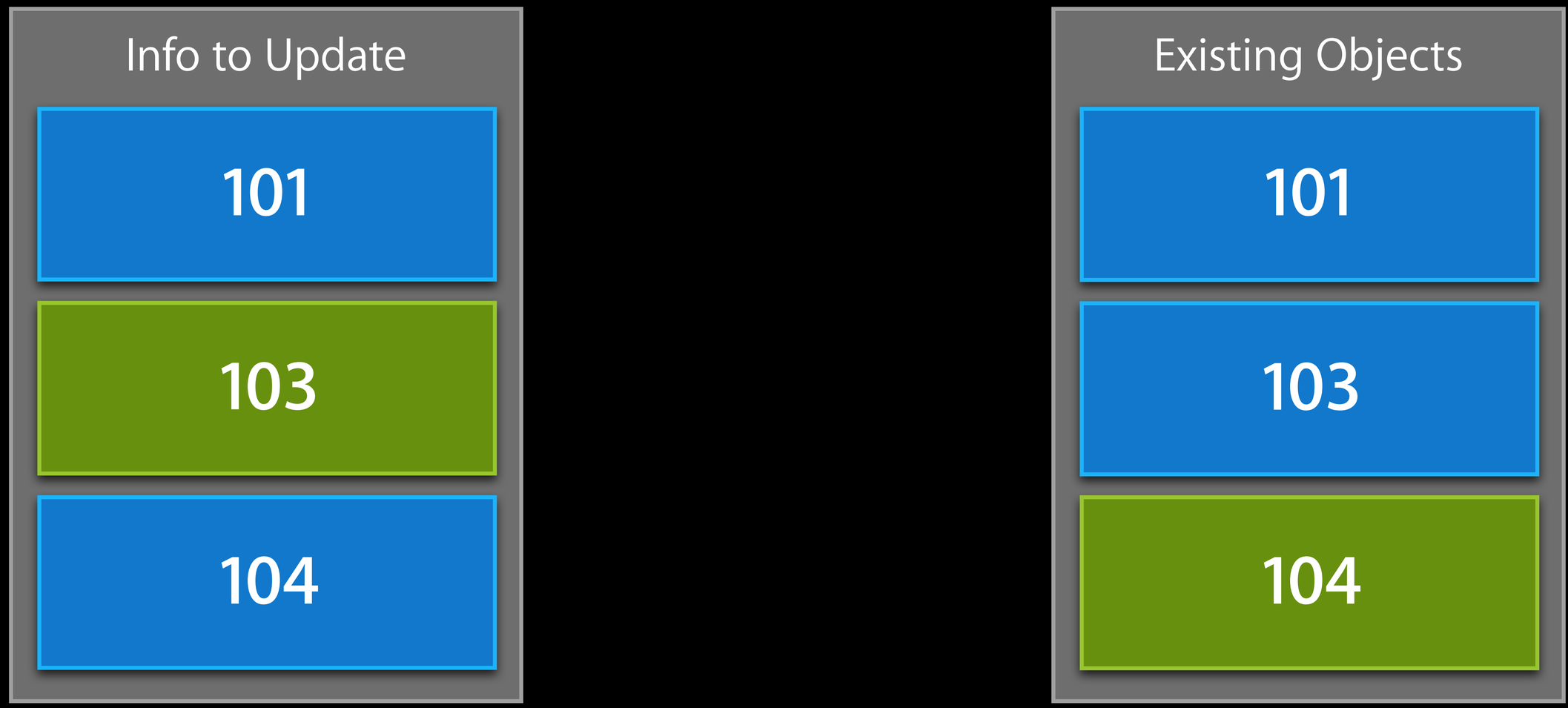
Implementing Update or Insert



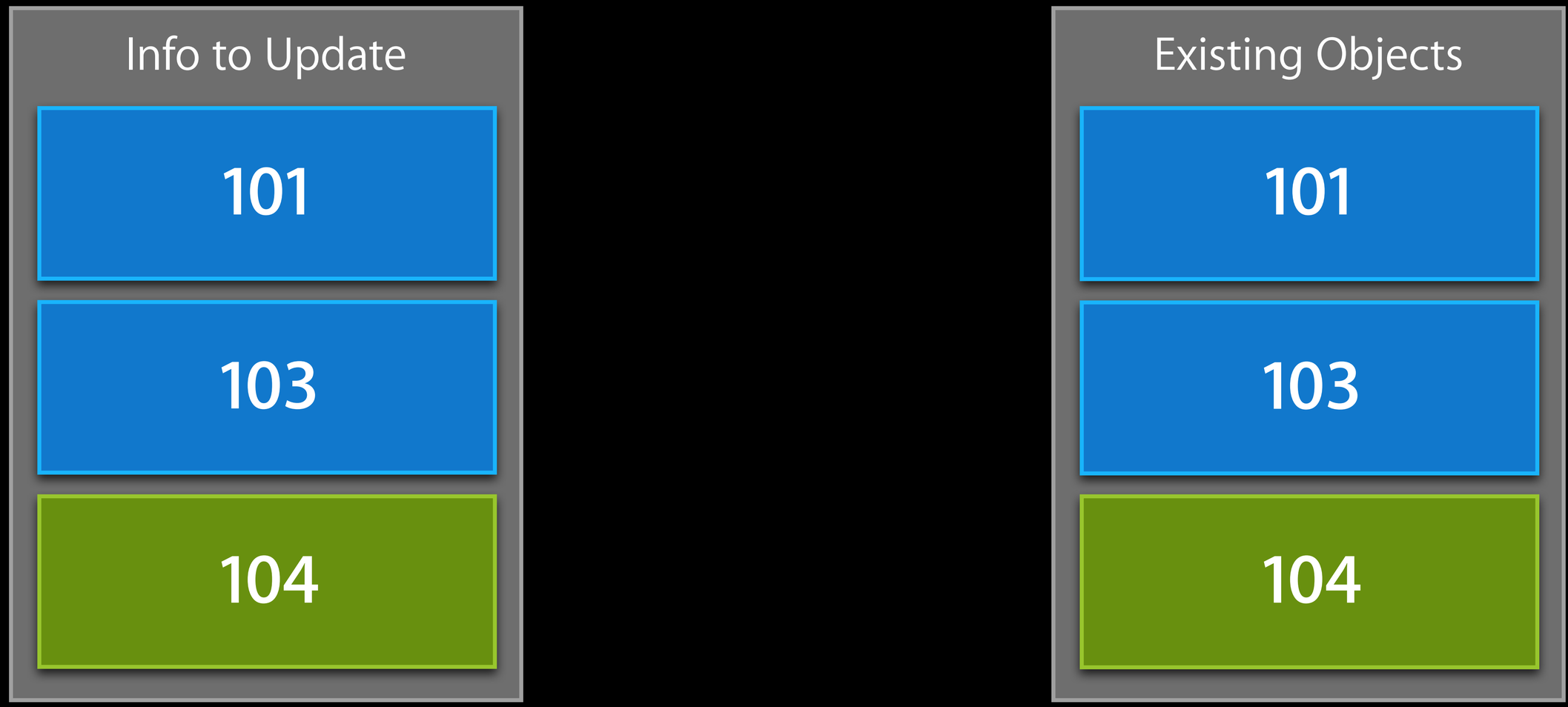
Implementing Update or Insert



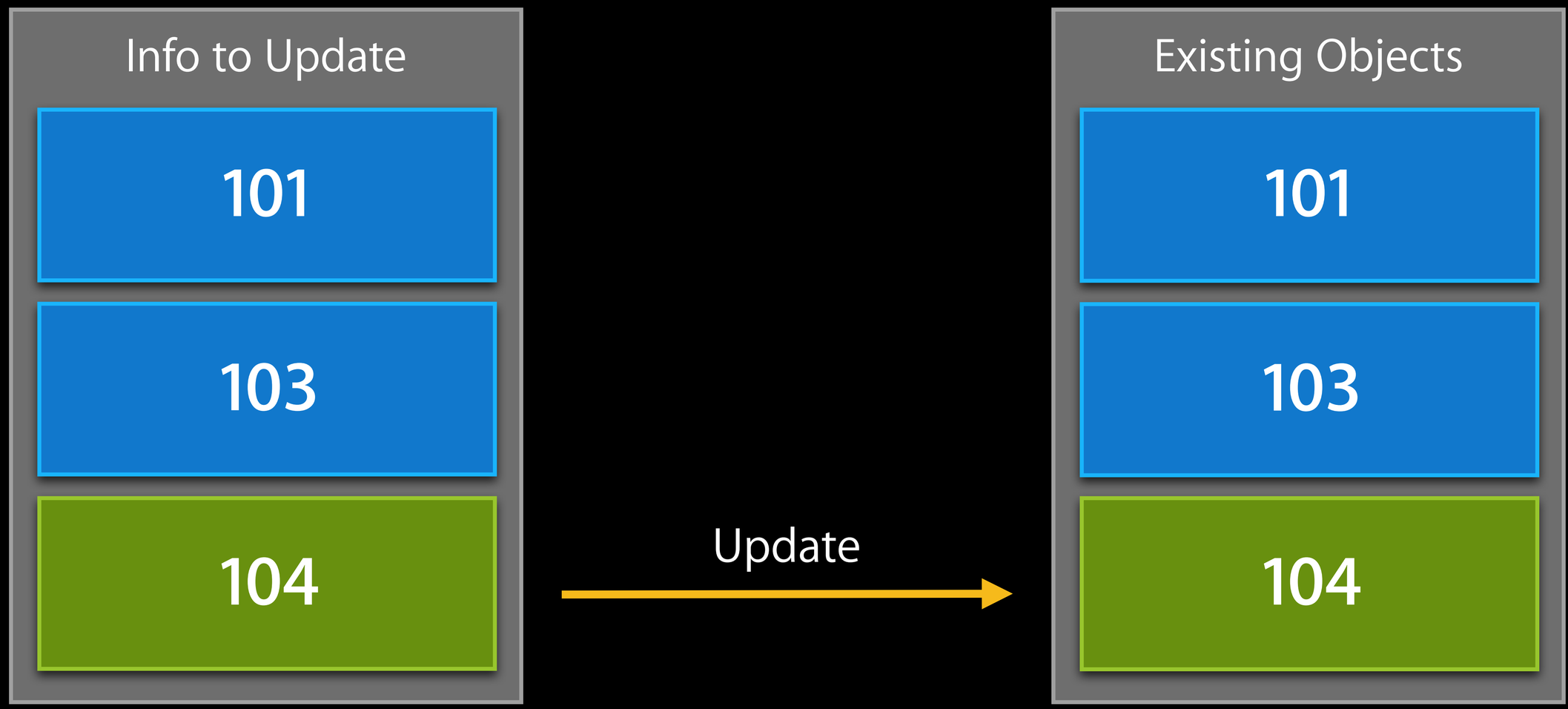
Implementing Update or Insert



Implementing Update or Insert



Implementing Update or Insert



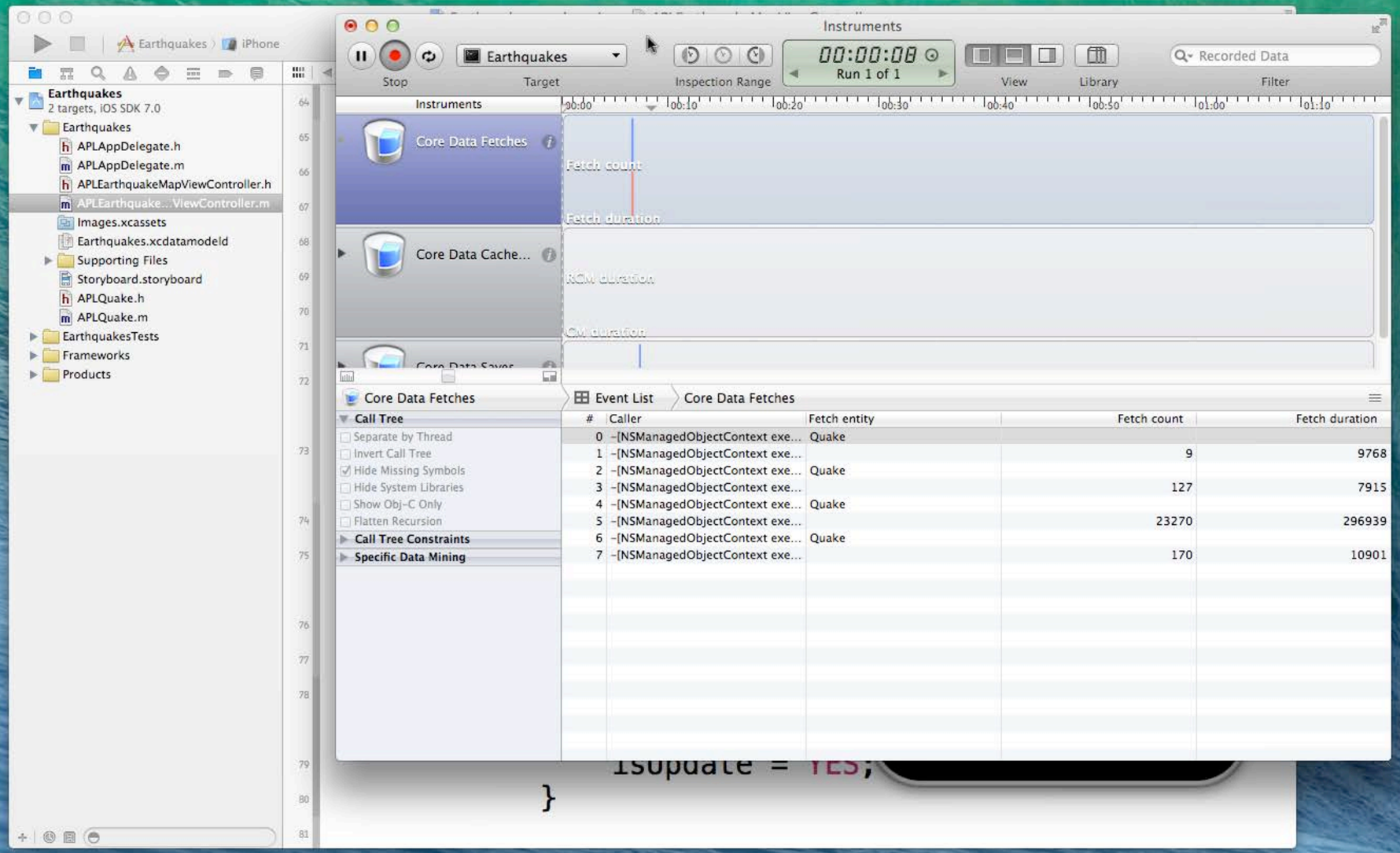
Earthquakes.xcodeproj — APLEarthquakeMapViewController.m

Earthquakes › iPhone

Finished running - Profiling Earthquakes.app on iPhone — No Issues

Earthquakes › Earthquakes › APLEarthquakeMapViewController.m › -refreshAction:


```
        }

        NSEnumerator *jsonQuakeEnumerator =
            [results objectEnumerator];

        NSEnumerator *matchingQuakeEnumerator =
            [matchingQuakes objectEnumerator];

        NSDictionary *jsonQuake = [jsonQuakeEnumerator
            nextObject];
        APLQuake *quake = [matchingQuakeEnumerator
            nextObject];
        while (jsonQuake) {
            NSDictionary *properties =
                jsonQuake[@"properties"];

            BOOL isUpdate = NO;
            if ([properties[@"code"] isEqualToString:
                [quake uuid]]) {
                isUpdate = YES;
            }
```


Earthquakes
iPhone
Earthquakes
2 targets, iOS SDK 7.0
Earthquakes
APLAppDelegate.h
APLAppDelegate.m
APLEarthquakeMapViewController.h
APLEarthquake...ViewController.m
Images.xcassets
Earthquakes.xcdatamodeld
Supporting Files
Storyboard.storyboard
APLQuake.h
APLQuake.m
EarthquakesTests
Frameworks
Products
Instruments
Stop
Earthquakes
Target
Inspection Range
00:00:08
Run 1 of 1
View
Library
Recorded Data
Filter
Core Data Fetches
Fetch count
Fetch duration
Core Data Cache...
Core Data Fetches
Event List
Core Data Fetches
Call Tree
Separate by Thread
Invert Call Tree
Hide Missing Symbols
Hide System Libraries
Show Obj-C Only
Flatten Recursion
Call Tree Constraints
Specific Data Mining
Caller Fetch entity Fetch count Fetch duration
0 -[NSManagedObjectContext exe... Quake
1 -[NSManagedObjectContext exe... 9 9768
2 -[NSManagedObjectContext exe... Quake
3 -[NSManagedObjectContext exe... 127 7915
4 -[NSManagedObjectContext exe... Quake
5 -[NSManagedObjectContext exe... 23270 296939
6 -[NSManagedObjectContext exe... Quake
7 -[NSManagedObjectContext exe... 170 10901
isUpdate = YES;
}

Implementing Update or Insert

- Work in batches
- Experiment to find optimal batch size
- Test on all devices you support

Minimizing Memory Usage

Refaulting and Resetting

- Turn a single managed object back into a fault:

```
[context refreshObject:object mergeChanges:YES];
```

Refaulting and Resetting

- Turn a single managed object back into a fault:

```
[context refreshObject:object mergeChanges:NO];
```

- Reset an entire context, clearing all its managed objects:

```
[context reset];
```

Refaulting and Resetting

- Turn a single managed object back into a fault:

```
[context refreshObject:object mergeChanges:NO];
```

- Reset an entire context, clearing all its managed objects:

```
[context reset];
```

Any existing references to managed objects will be invalid

Earthquakes.xcodeproj — APLEarthquakeMapViewController.m

Earthquakes › iPhone

Finished running - Profiling Earthquakes.app on iPhone — No Issues

Earthquakes › Earthquakes › APLEarthquakeMapViewController.m › -refreshAction:

No Debug Session

```
#define kBatchSize 500
        NSUInteger count = [results count];
        NSUInteger numBatches = count / kBatchSize;
        numBatches += count % kBatchSize > 0 ? 1 : 0;

        for (int batchNumber = 0; batchNumber < numBatches;
            batchNumber++) {
            NSArray *subResults = [results subarrayWithRange:
                NSMakeRange(batchNumber * kBatchSize, MIN
                (kBatchSize, count - batchNumber * kBatchSize))];

            NSFetchRequest *matchingQuakeRequest =
                [NSFetchRequest
                fetchRequestWithEntityName:@"Quake"];
            matchingQuakeRequest.predicate = [NSPredicate
                predicateWithFormat:@"uuid in %@", [subResults
                valueForKeyPath:@"properties.code"]];
            matchingQuakeRequest.sortDescriptors = @[
                [NSSortDescriptor sortDescriptorWithKey:@"uuid"
                ascending:YES]];

            NSArray *matchingQuakes = [context
                executeFetchRequest:matchingQuakeRequest error:&
                anyError];
```

Instruments File Edit View Instrument Window Help

Tim Isted

Earthquakes.xcodeproj — APLEarthquakeMapViewController.m

Earthquakes › iPhone

No Debug Session

Instruments

Stop | Target: Earthquakes | Inspection Range | 00:00:08 Run 1 of 1 | View | Library | Filter: Recorded Data

Instruments

Core Data Fetches — Fetch count, Fetch duration

Core Data Cache... — RCM duration, ICM duration

Core Data Saves

Core Data Fetches | Event List | Core Data Fetches

Call Tree

- Separate by Thread
- Invert Call Tree
- Hide Missing Symbols
- Hide System Libraries
- Show Obj-C Only
- Flatten Recursion

Call Tree Constraints

Specific Data Mining

#	Caller	Fetch entity	Fetch count	Fetch duration
0	-[NSManagedObjectContext exe...	Quake		
1	-[NSManagedObjectContext exe...		43	3908
2	-[NSManagedObjectContext exe...	Quake		
3	-[NSManagedObjectContext exe...		500	14641
4	-[NSManagedObjectContext exe...	Quake		
5	-[NSManagedObjectContext exe...		500	13613
6	-[NSManagedObjectContext exe...	Quake		
7	-[NSManagedObjectContext exe...		500	12984
8	-[NSManagedObjectContext exe...	Quake		
9	-[NSManagedObjectContext exe...		500	13959
10	-[NSManagedObjectContext exe...	Quake		
11	-[NSManagedObjectContext exe...		500	13905
12	-[NSManagedObjectContext exe...	Quake		
13	-[NSManagedObjectContext exe...		500	12750
14	-[NSManagedObjectContext exe...	Quake		
15	-[NSManagedObjectContext exe...		500	12920
16	-[NSManagedObjectContext exe...	Quake		
17	-[NSManagedObjectContext exe...		500	14100
18	-[NSManagedObjectContext exe...	Quake		
19	-[NSManagedObjectContext exe...		500	13462

```
#defin
NSArray *matchingQuakes = [c
    executeFetchRequest:matchingQuakeRequest error:&
    anyError];
```

Fetch Only What You Need

Fetch Dictionaries

Consider returning dictionaries instead of managed objects

- Configure the fetch request to return dictionaries:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Quake"];
[request setResultType:NSDictionaryResultType];
```

Fetch Dictionaries

Consider returning dictionaries instead of managed objects

- Configure the fetch request to return dictionaries:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Quake"];
[request setResultType:NSDictionaryResultType];
```

- Just fetch the values you need:

```
[request setPropertiesToFetch:@[ @"magnitude" ]];
```

Use Aggregate Operations

Use SQLite to perform your calculations

- Configure the fetch request to return dictionaries:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Quake"];
[request setResultType:NSDictionaryResultType];
```

- Use an expression description:

```
NSExpressionDescription *ed = [[NSExpressionDescription alloc] init];
ed.name = @"minimum";
ed.expression = [NSExpression expressionForFunction:@"min:"
              arguments:@[ [NSExpression expressionForKeyPath:@"magnitude"]]];
```

Use Aggregate Operations

Use SQLite to perform your calculations

- Configure the fetch request to return dictionaries:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Quake"];
[request setResultType:NSDictionaryResultType];
```

- Use an expression description:

```
NSExpressionDescription *ed = [[NSExpressionDescription alloc] init];
ed.name = @"minimum";
ed.expression = [NSExpression expressionForFunction:@"min:"
              arguments:@[ [NSExpression expressionForKeyPath:@"magnitude"]]];
```

- Set the properties to fetch:

```
[request setPropertiesToFetch:@[ ed ]];
```

Group Results

Use SQLite to group your results automatically

Group Results

Use SQLite to group your results automatically

```
NSFetchRequest *request =
                [NSFetchRequest fetchRequestWithEntityName:@"Quake"];
[request setResultType:NSDictionaryResultType];
```

- Use an expression description:

```
NSExpressionDescription *ed = [[NSExpressionDescription alloc] init];
ed.name = @"count";
ed.expression = [NSExpression expressionForFunction:@"count:"
            arguments:@[ [NSExpression expressionForKeyPath:@"magnitude"]]];
```

- Set the properties to fetch and group by:

```
[request setPropertiesToFetch:@[ @"magnitude", ed ]];
[request setPropertiesToGroupBy:@[ @"magnitude" ]];
```

See What's Going On

Using SQL Logging

See what Core Data is doing behind the scenes

- Pass argument on launch:
  - `-com.apple.CoreData.SQLDebug 1`
- Use value of 1, 2, or 3
- See raw SQL queries
- Get exact timings
- Note: SQLite schema is private and subject to change

```
28      NSFetchRequest *request = [NSFetchRequest
            fetchRequestWithEntityName:@"Quake"];
29      request.resultType = NSDictionaryResultType;
30
31      NSExpressionDescription *ed = [[NSExpressionDescription alloc] init];
32      ed.name = @"minimum";
33      ed.expression = [NSExpression expressionForFunction:@"min:" arguments:@[
            [NSExpression expressionForKeyPath:@"magnitude"]]];
34
35      request.propertiesToFetch = @[ ed ];
36      request.predicate = [NSPredicate predicateWithFormat:@"time > %@",
            [NSDate dateWithTimeIntervalSinceNow:-60*60*24*7]];
37
38      NSError *anyError = nil;
39      NSArray *results = [self.managedObjectContext executeFetchRequest:request
            error:&anyError];
40      if (!results || [results count] < 1) {
41          NSLog(@"Error fetching: %@", anyError);
42          return;
43      }
44
45      self.minimumCell.detailTextLabel.text = results[0][@"minimum"];
46
```

Earthquakes.xcodeproj — APLMagnitudesViewController.m

Earthquakes › iPhone | Running Earthquakes on iPhone | 6

Earthquakes › Earthquakes › APLMagnitudesViewController.m › -prepareForSegue:sender:

```
28      NSFetchRequest *request = [NSFetchRe
            fetchRequestWithEntityName:@"Qua
29      request.resultType = NSDictionaryRes
30
31      NSExpressionDescription *ed = [[NSEx                    loc] init];
32      ed.name = @"minimum";
33      ed.expression = [NSExpression expres                    " arguments:@[
            [NSExpression expressionForKeyPa
34
35      request.propertiesToFetch = @[ ed ];
36      request.predicate = [NSPredicate pre                    e > %@",
            [NSDate dateWithTimeIntervalSince
37
38      NSError *anyError = nil;
39      NSArray *results = [self.managedObje                    Request:request
            error:&anyError];
40      if (!results || [results count] < 1)
41          NSLog(@"Error fetching: %@", anyE
42          return;
43      }
44
45      self.minimumCell.detailTextLabel.text = results[0][@"minimum"];
```

Earthquakes

Earthquakes.xcodeproj — APLMagnitudesViewController.m

Earthquakes › iPhone | Running Earthquakes on iPhone | 6

Earthquakes › Earthquakes › APLMagnitudesViewController.m › -prepareForSegue:sender:

```
28      NSFetchRequest *request = [NSFetchRequest
            fetchRequestWithEntityName:@"Quake"];
29      request.resultType = NSDictionaryResultType;
30
31      NSExpressionDescription *ed = [[NSExpressionDescription alloc] init];
32      ed.name = @"minimum";
33      ed.expression = [NSExpression expressionForFunction:@"min:" arguments:@[
            [NSExpression expressionForKeyPath:@"magnitude"]]];
34
35      request.propertiesToFetch = @[ ed ];
36      request.predicate = [NSPredicate predicateWithFormat:@"time > %@",
            [NSDate dateWithTimeIntervalSinceNow:-60*60*24*7]];
```

Earthquakes

```
2013-06-09 20:43:28.382 Earthquakes[5082:907] CoreData: annotation: Connecting to sqlite database file at "/Users/timisted/
Library/Application Support/iPhone Simulator/7.0/Applications/C244682A-39B5-426A-99CC-7FDEF1662024/Documents/
Earthquakes.sqlite"
2013-06-09 20:43:28.383 Earthquakes[5082:907] CoreData: sql: pragma cache_size=200
2013-06-09 20:43:28.384 Earthquakes[5082:907] CoreData: sql: SELECT Z_VERSION, Z_UUID, Z_PLIST FROM Z_METADATA
2013-06-09 20:43:28.393 Earthquakes[5082:907] CoreData: sql: SELECT min( t0.ZMAGNITUDE) FROM ZQUAKE t0 WHERE  t0.ZTIME > ?
2013-06-09 20:43:28.398 Earthquakes[5082:907] CoreData: annotation: sql connection fetch time: 0.0054s
2013-06-09 20:43:28.398 Earthquakes[5082:907] CoreData: annotation: total fetch execution time: 0.0059s for 1 rows.
2013-06-09 20:43:28.400 Earthquakes[5082:907] CoreData: sql: SELECT max( t0.ZMAGNITUDE) FROM ZQUAKE t0 WHERE  t0.ZTIME > ?
2013-06-09 20:43:28.405 Earthquakes[5082:907] CoreData: annotation: sql connection fetch time: 0.0050s
2013-06-09 20:43:28.406 Earthquakes[5082:907] CoreData: annotation: total fetch execution time: 0.0054s for 1 rows.
2013-06-09 20:43:28.406 Earthquakes[5082:907] CoreData: sql: SELECT avg( t0.ZMAGNITUDE) FROM ZQUAKE t0 WHERE  t0.ZTIME > ?
2013-06-09 20:43:28.412 Earthquakes[5082:907] CoreData: annotation: sql connection fetch time: 0.0050s
2013-06-09 20:43:28.412 Earthquakes[5082:907] CoreData: annotation: total fetch execution time: 0.0055s for 1 rows.
```

All Output

Earthquakes.xcodeproj — APLMagnitudesViewController.m

Earthquakes › iPhone | Running Earthquakes on iPhone | 6

Earthquakes › Earthquakes › APLMagnitudesViewController.m › -prepareForSegue:sender:

```
28      NSFetchRequest *request = [NSFetchRequest
            fetchRequestWithEntityName:@"Quake"];
29      request.resultType = NSDictionaryResultType;
30
31      NSExpressionDescription *ed = [[NSExpressionDescription alloc] init];
32      ed.name = @"minimum";
33      ed.expression = [NSExpression expressionForFunction:@"min:" arguments:@[
            [NSExpression expressionForKeyPath:@"magnitude"]]];
34
35      request.propertiesToFetch = @[ ed ];
36      request.predicate = [NSPredicate predicateWithFormat:@"time > %@",
            [NSDate dateWithTimeIntervalSinceNow:-60*60*24*7]];
```

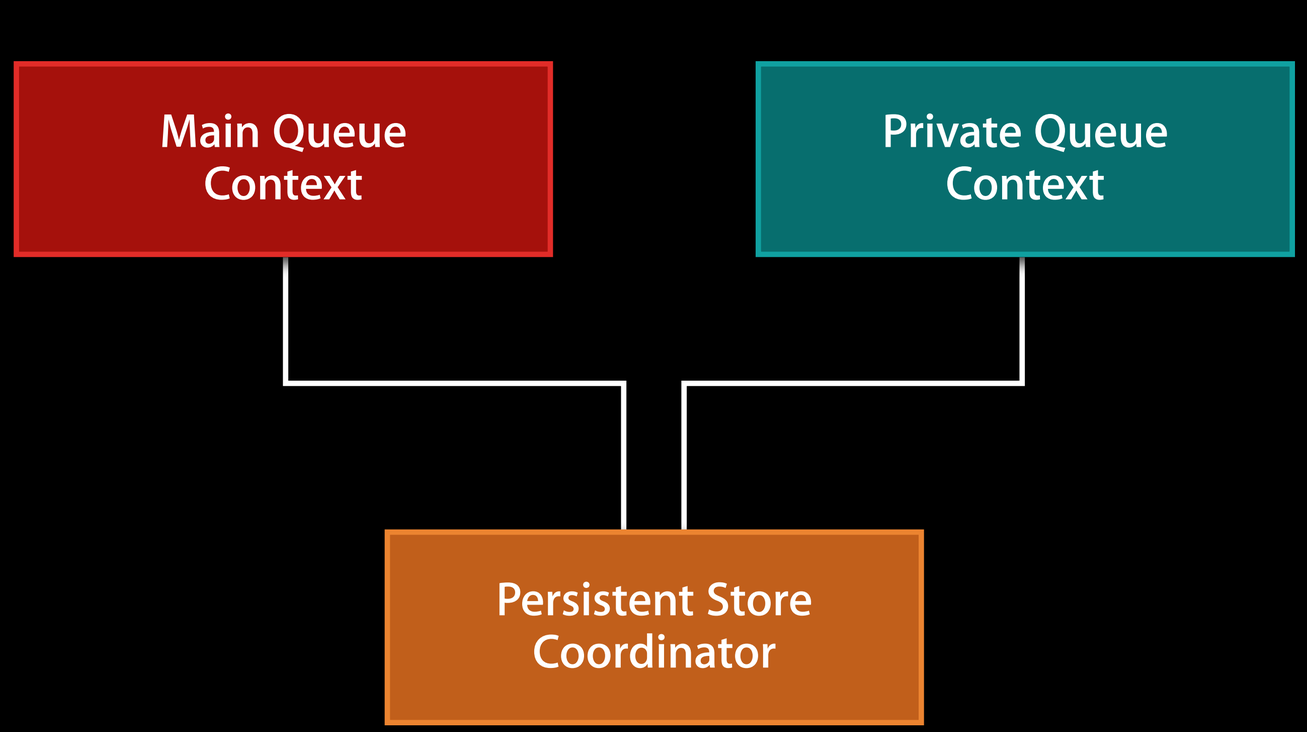
Earthquakes

```
2013-06-09 20:43:28.384 Earthquakes[5082:907] CoreData: sql: SELECT Z_VERSION, Z_UUID, Z_PLIST FROM Z_METADATA
2013-06-09 20:43:28.393 Earthquakes[5082:907] CoreData: sql: SELECT min( t0.ZMAGNITUDE) FROM ZQUAKE t0 WHERE  t0.ZTIME > ?
2013-06-09 20:43:28.398 Earthquakes[5082:907] CoreData: annotation: sql connection fetch time: 0.0054s
2013-06-09 20:43:28.398 Earthquakes[5082:907] CoreData: annotation: total fetch execution time: 0.0059s for 1 rows.
2013-06-09 20:43:28.400 Earthquakes[5082:907] CoreData: sql: SELECT max( t0.ZMAGNITUDE) FROM ZQUAKE t0 WHERE  t0.ZTIME > ?
2013-06-09 20:43:28.405 Earthquakes[5082:907] CoreData: annotation: sql connection fetch time: 0.0050s
2013-06-09 20:43:28.406 Earthquakes[5082:907] CoreData: annotation: total fetch execution time: 0.0054s for 1 rows.
2013-06-09 20:43:28.406 Earthquakes[5082:907] CoreData: sql: SELECT avg( t0.ZMAGNITUDE) FROM ZQUAKE t0 WHERE  t0.ZTIME > ?
2013-06-09 20:43:28.412 Earthquakes[5082:907] CoreData: annotation: sql connection fetch time: 0.0050s
2013-06-09 20:43:28.412 Earthquakes[5082:907] CoreData: annotation: total fetch execution time: 0.0055s for 1 rows.
2013-06-09 20:43:37.932 Earthquakes[5082:907] CoreData: sql: SELECT t0.ZMAGNITUDEFLOOR, COUNT( t0.ZMAGNITUDEFLOOR) FROM
ZQUAKE t0 GROUP BY  t0.ZMAGNITUDEFLOOR ORDER BY t0.ZMAGNITUDE
2013-06-09 20:43:37.955 Earthquakes[5082:907] CoreData: annotation: sql connection fetch time: 0.0222s
2013-06-09 20:43:37.955 Earthquakes[5082:907] CoreData: annotation: total fetch execution time: 0.0227s for 10 rows.
```

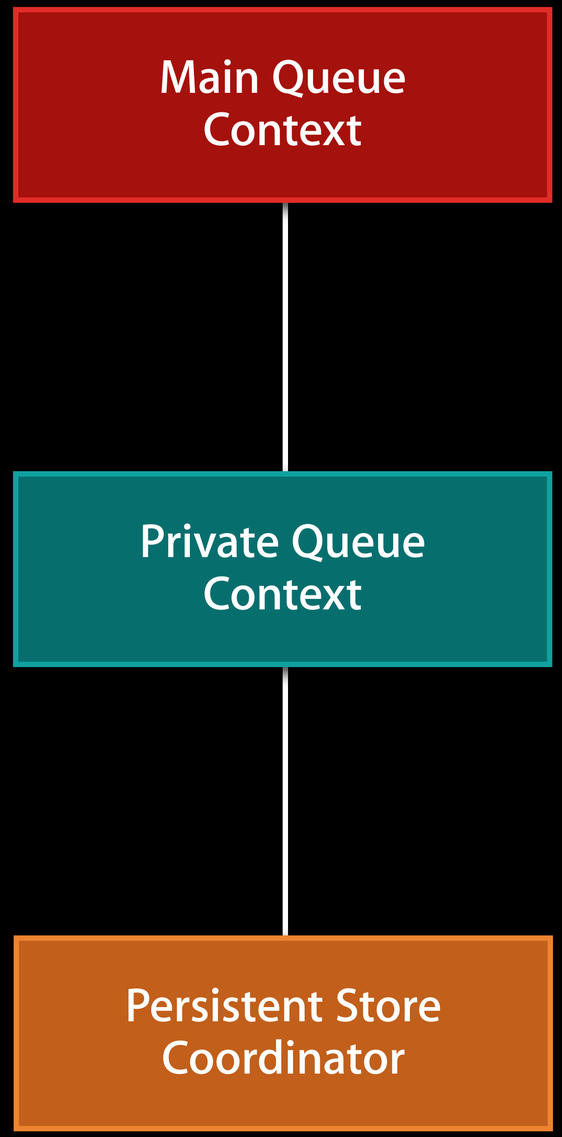
All Output

Concurrency Models

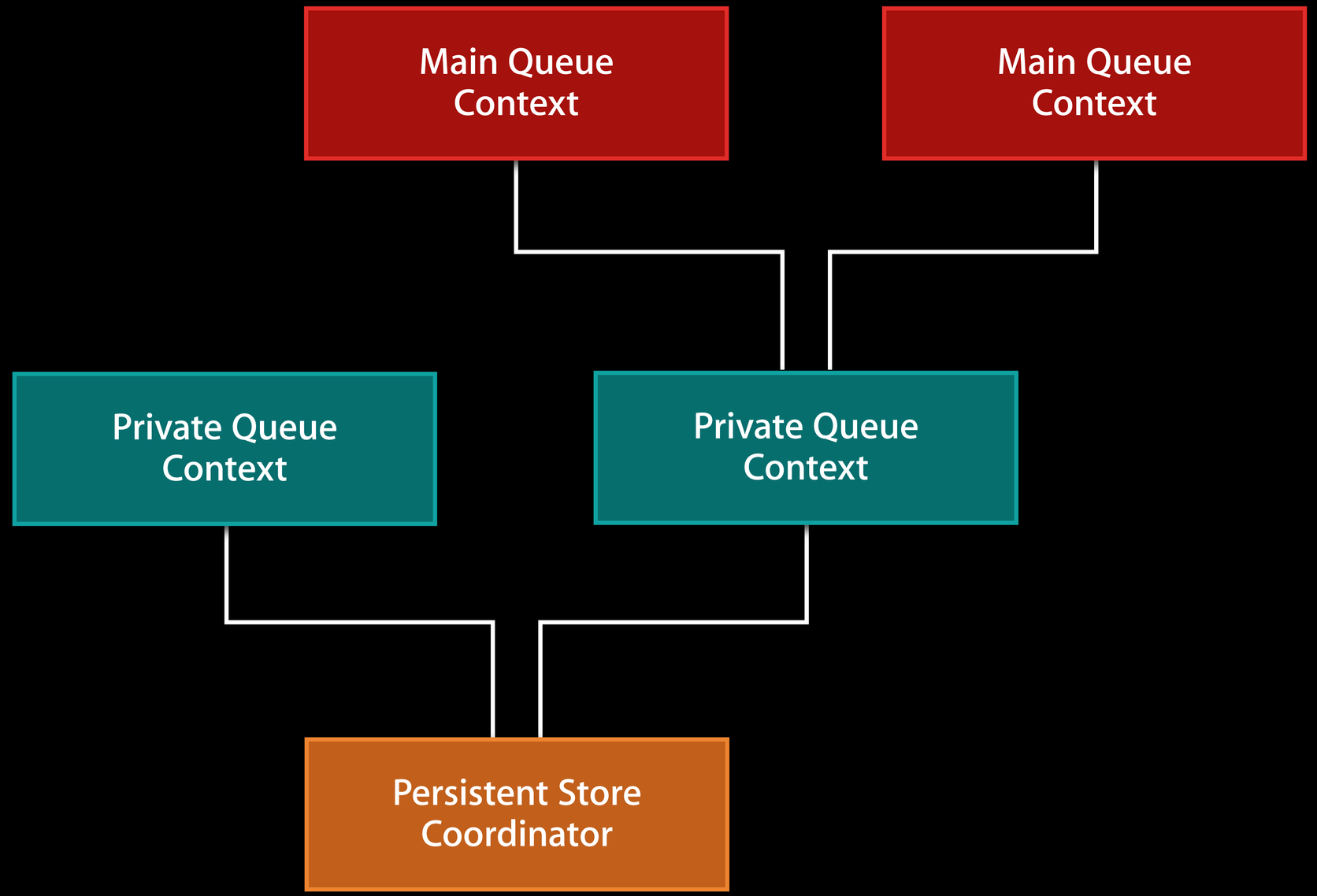
Concurrency with Core Data



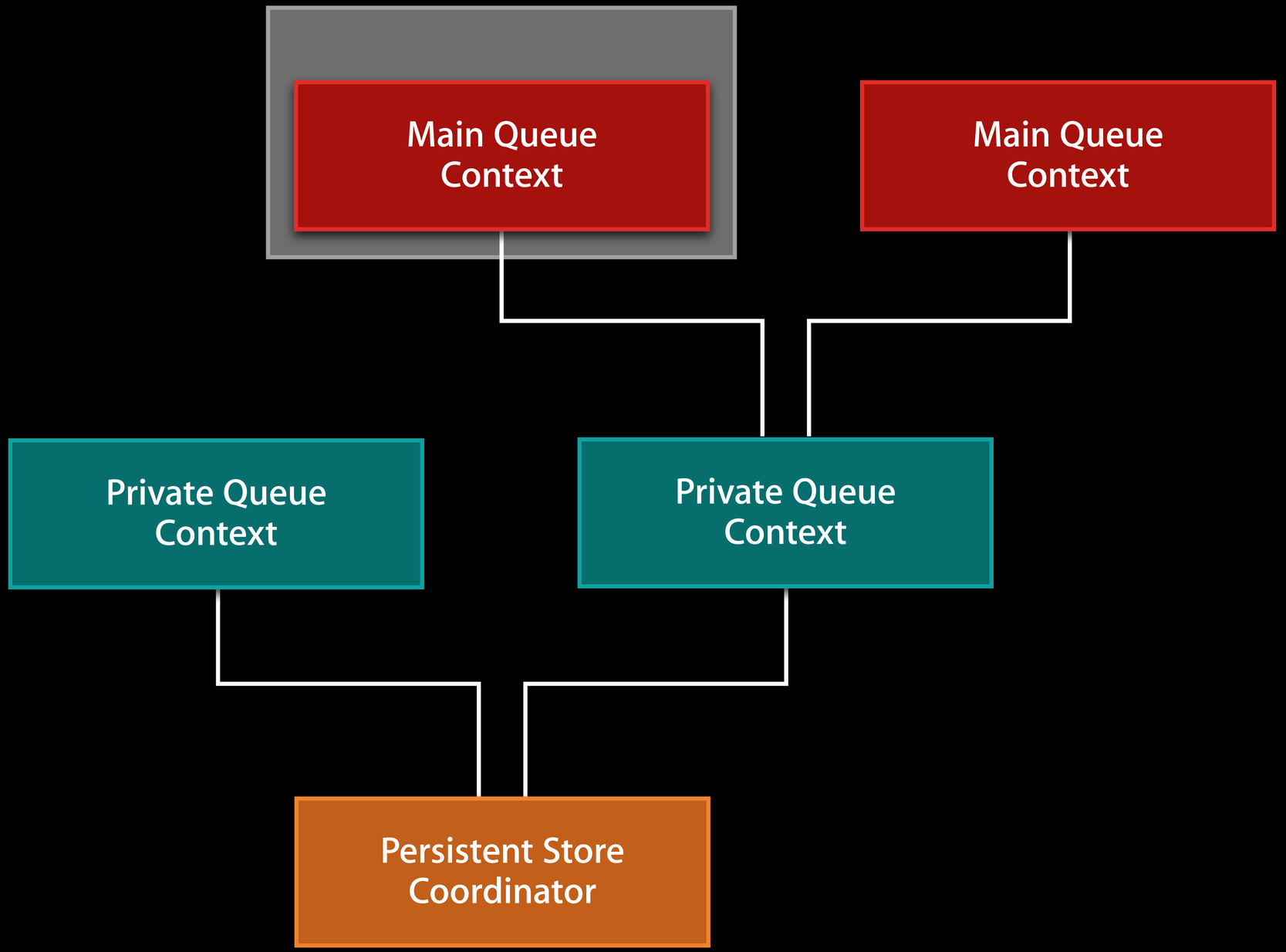
Concurrency with Core Data



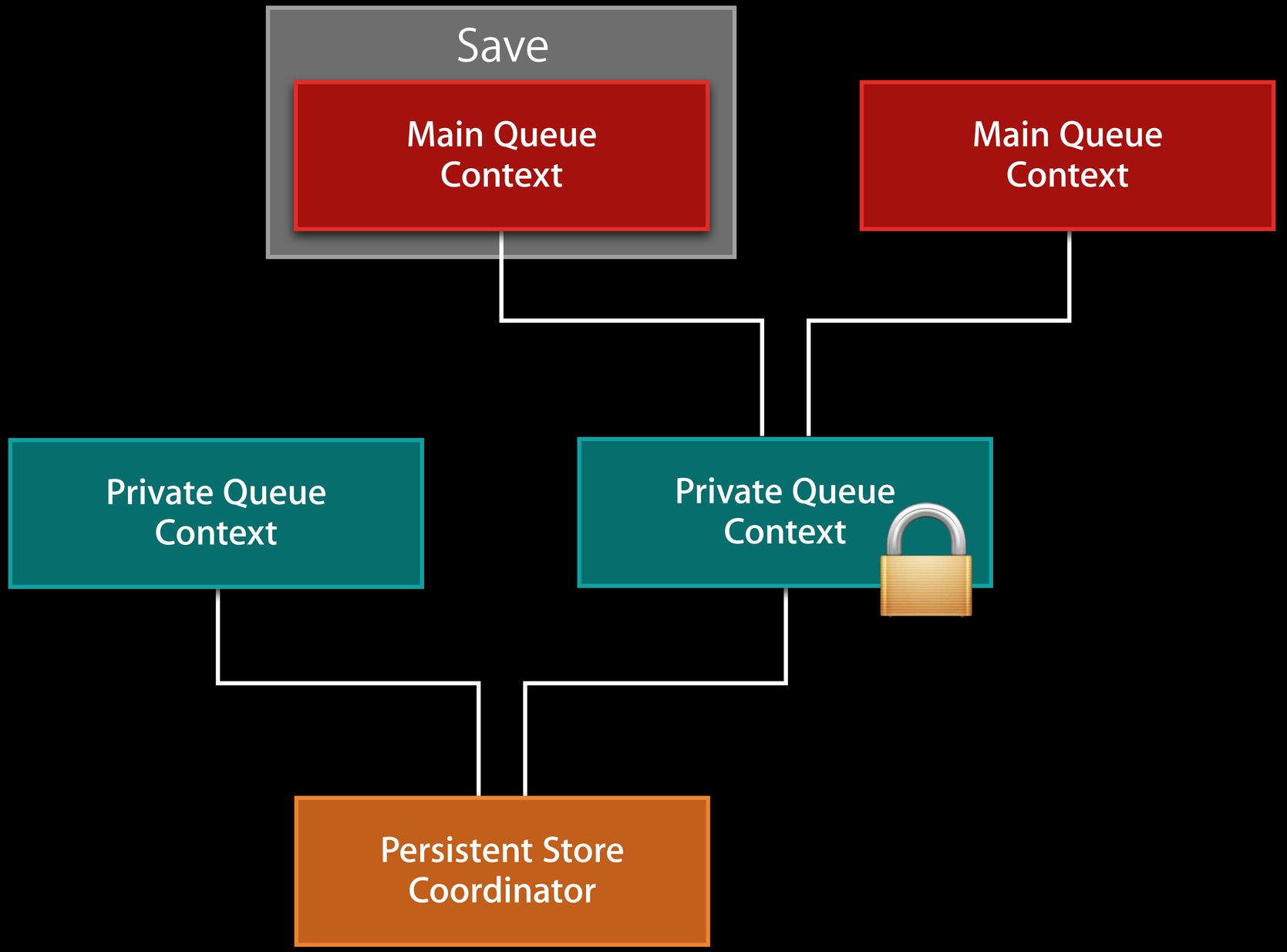
Concurrency with Core Data



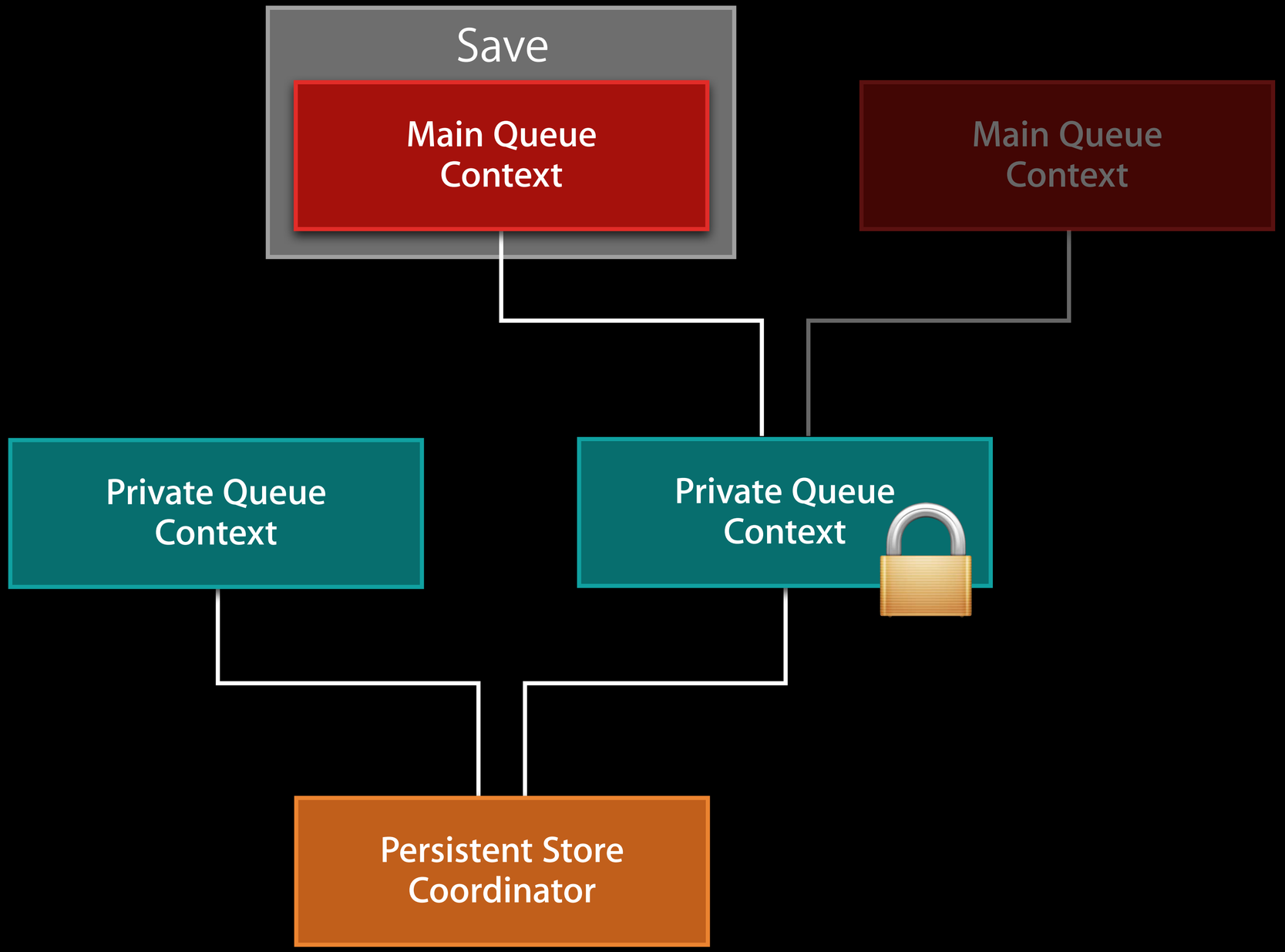
Concurrency with Core Data



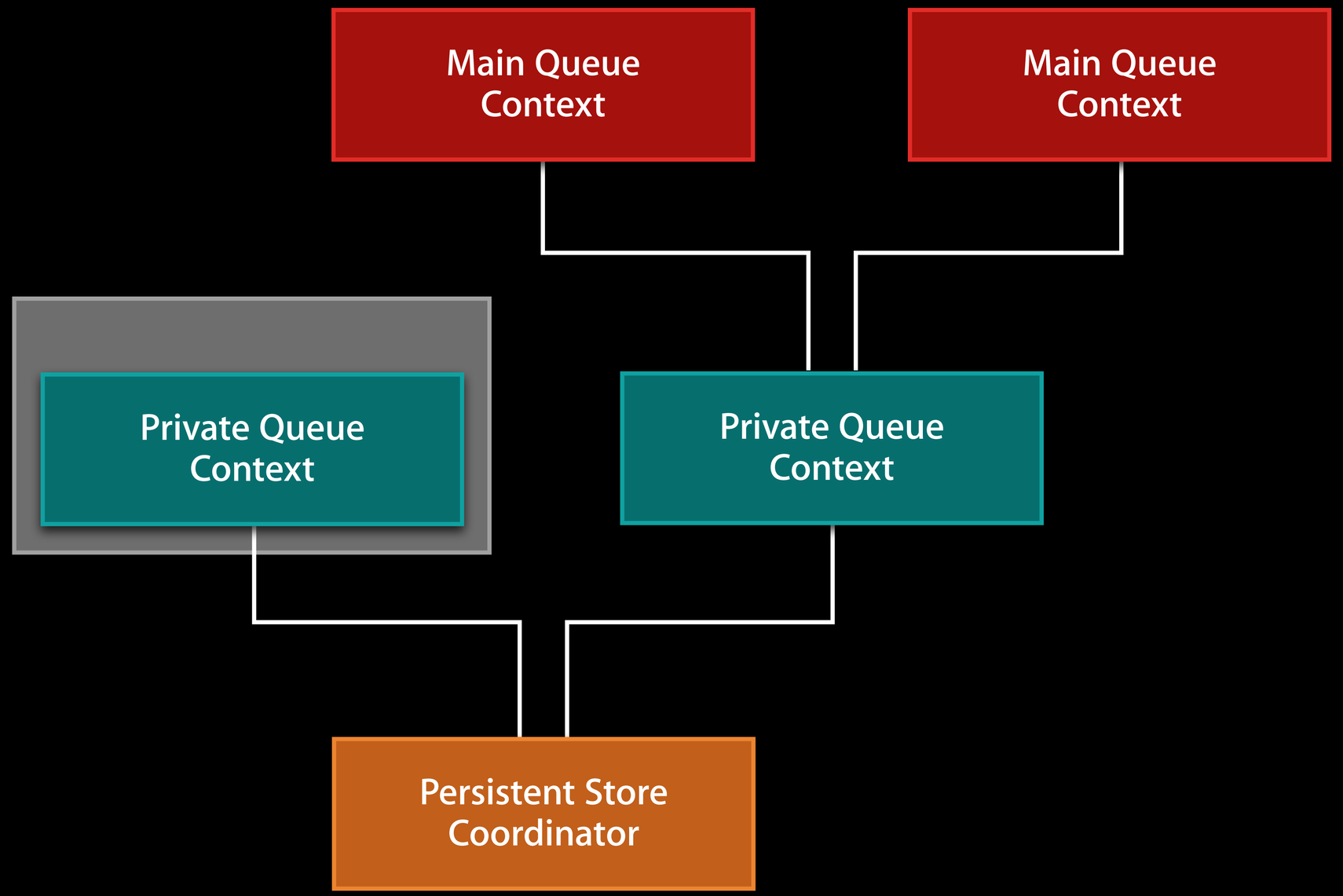
Concurrency with Core Data



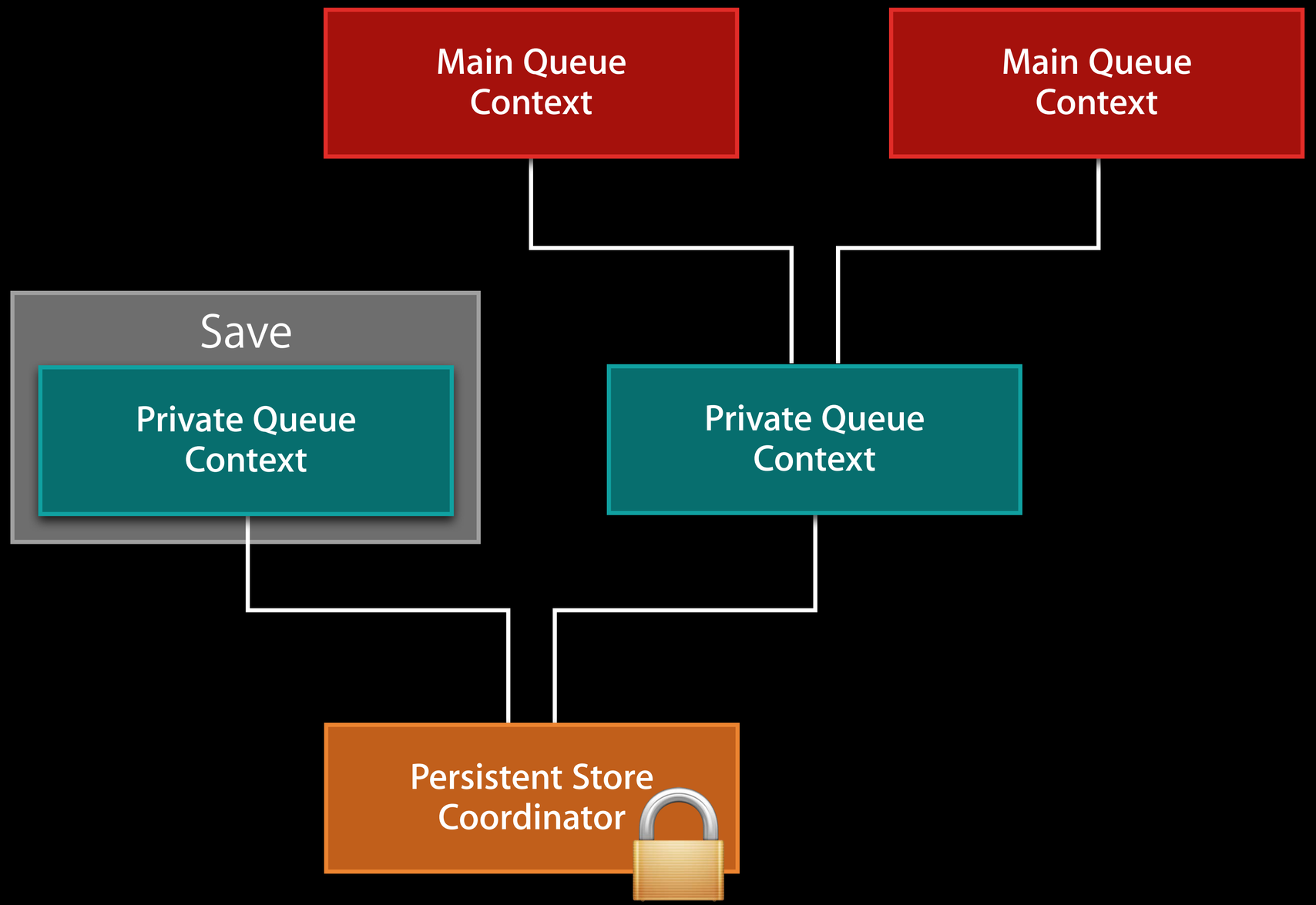
Concurrency with Core Data



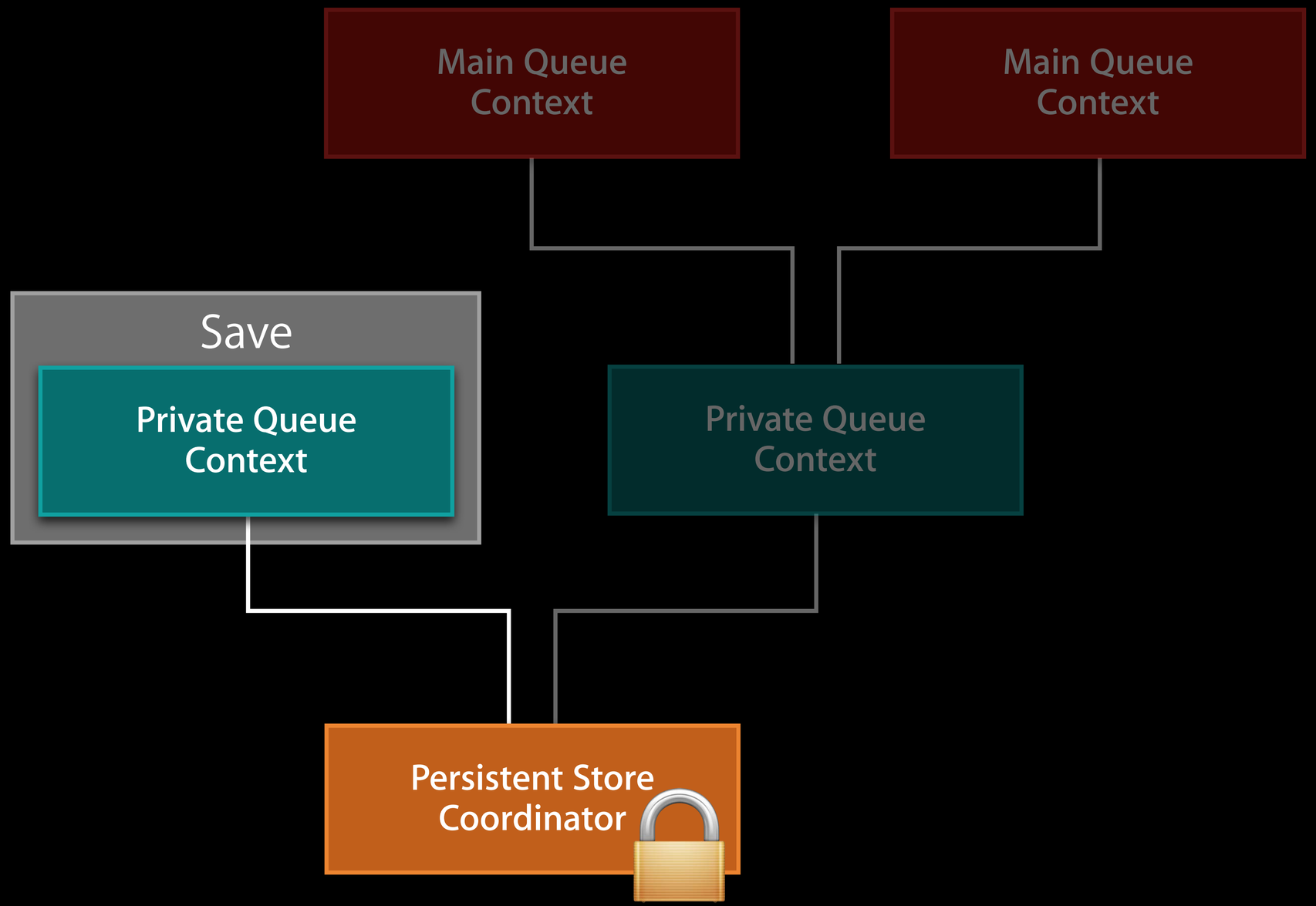
Concurrency with Core Data



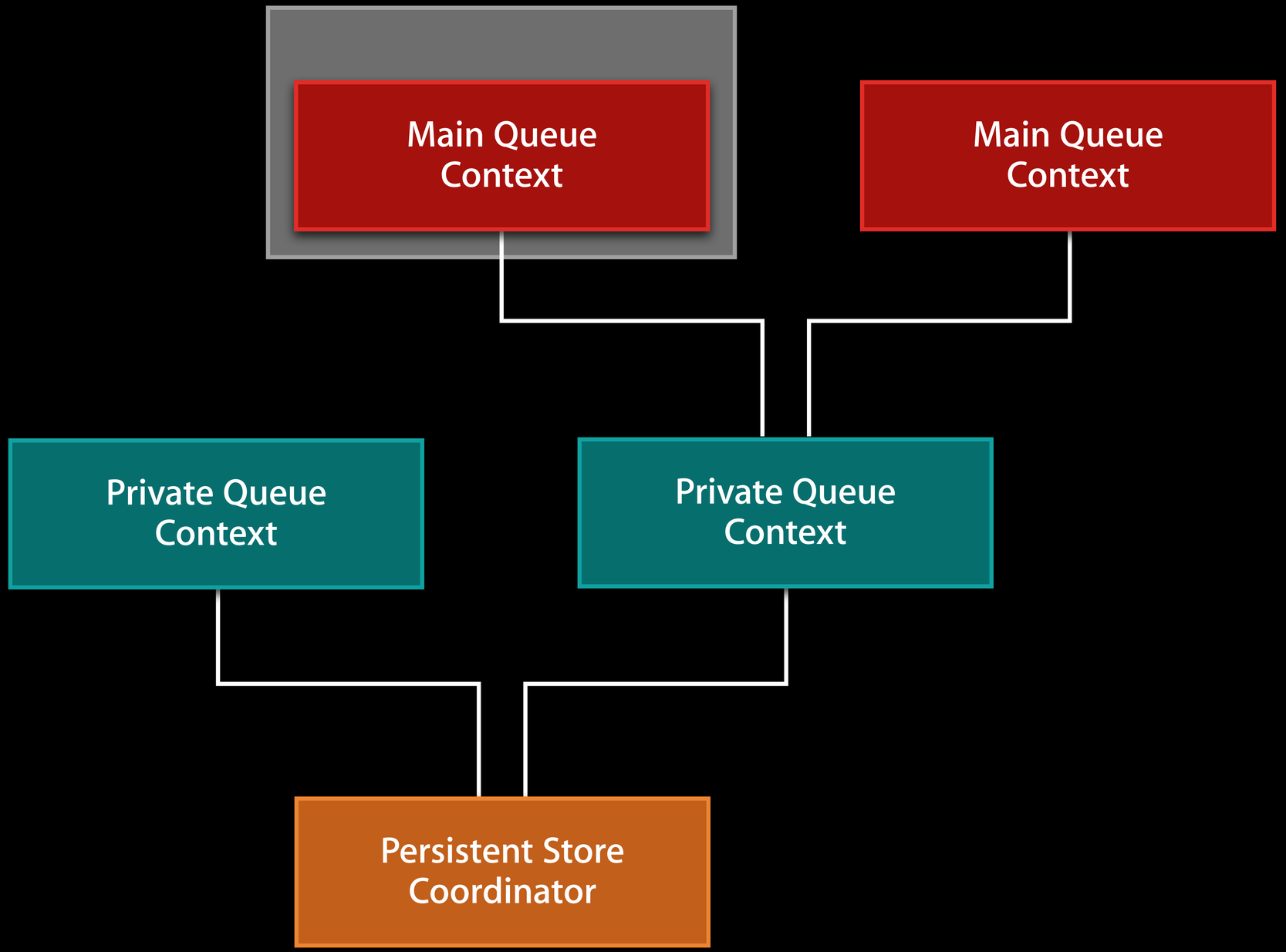
Concurrency with Core Data



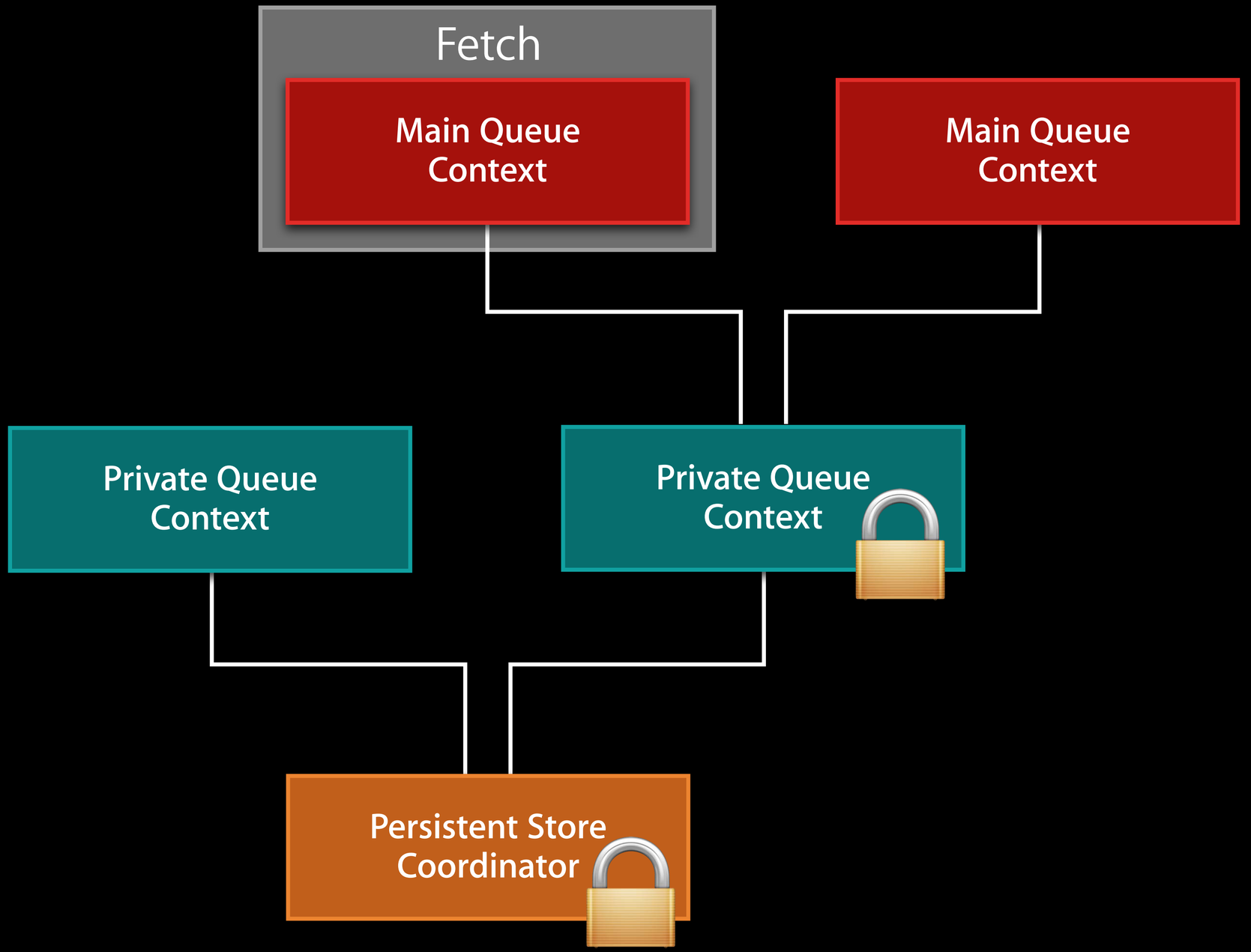
Concurrency with Core Data



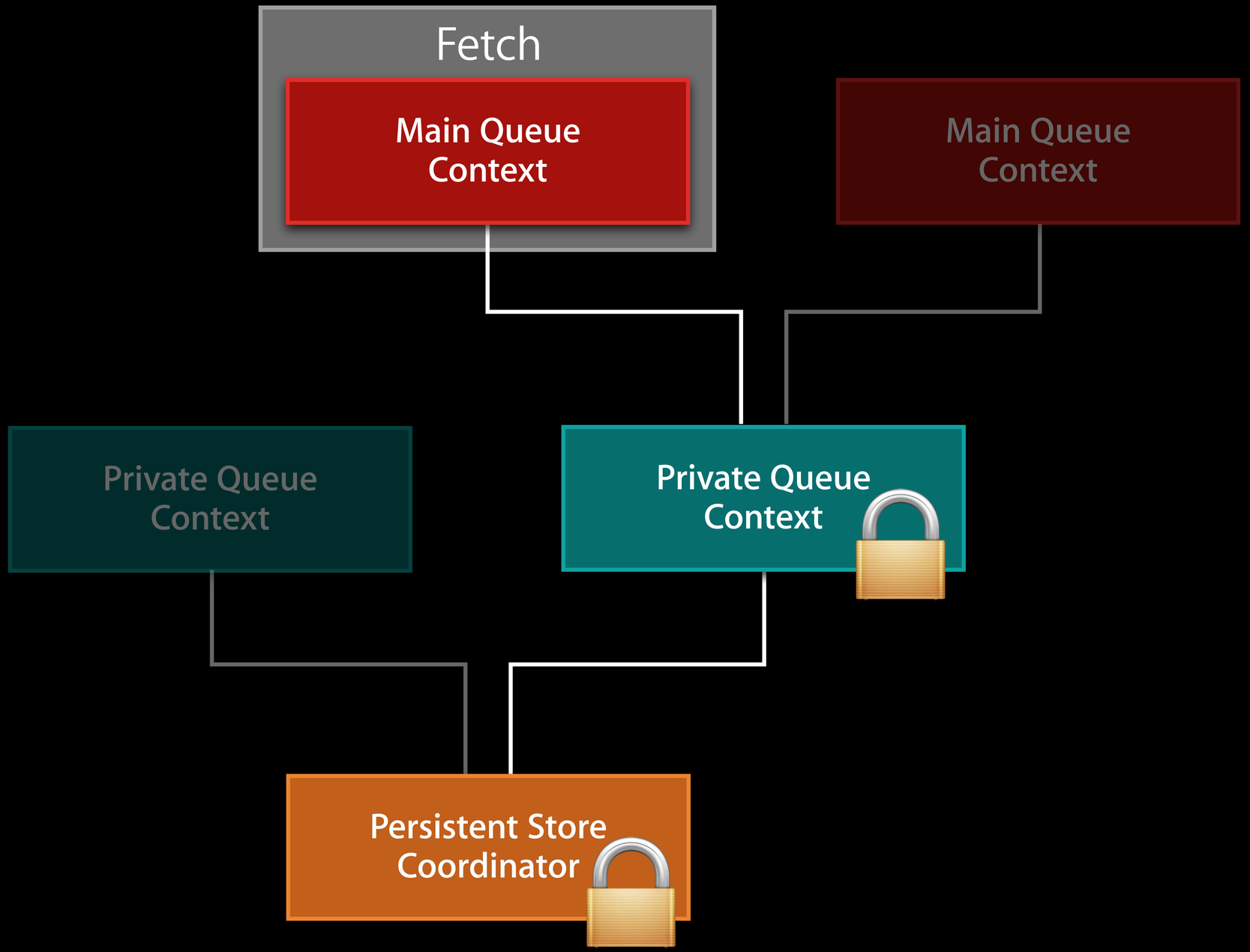
Concurrency with Core Data



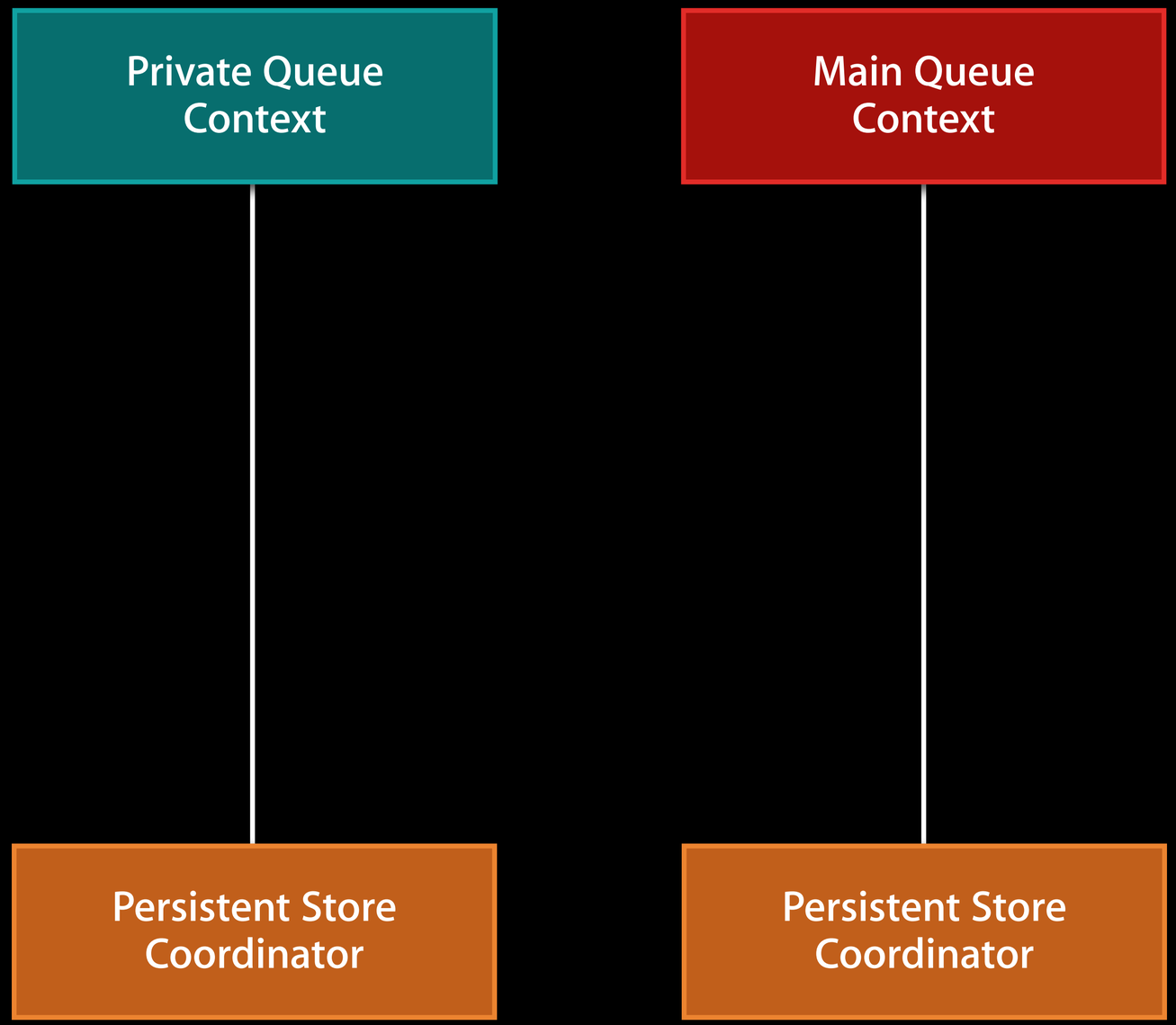
Concurrency with Core Data



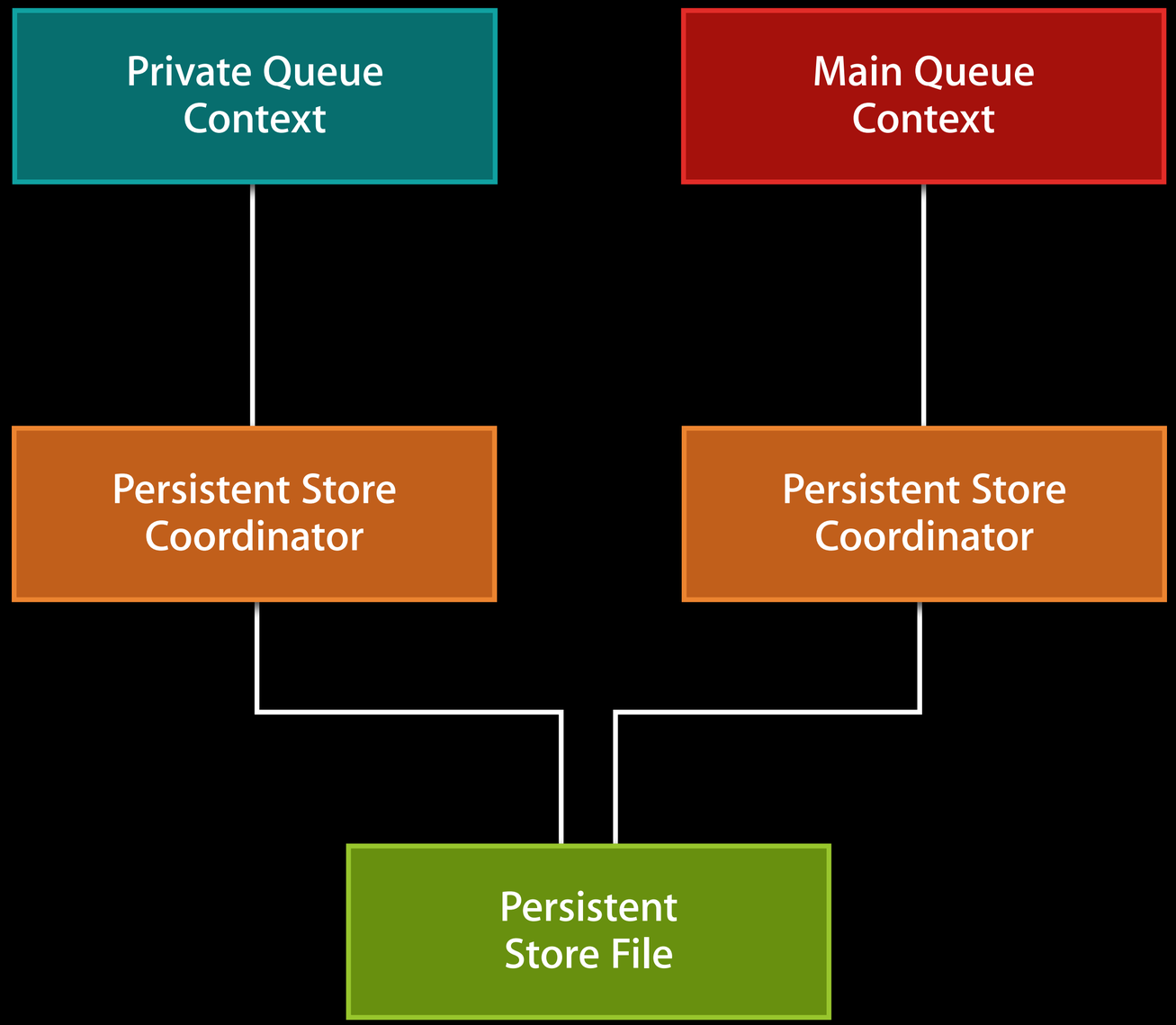
Concurrency with Core Data



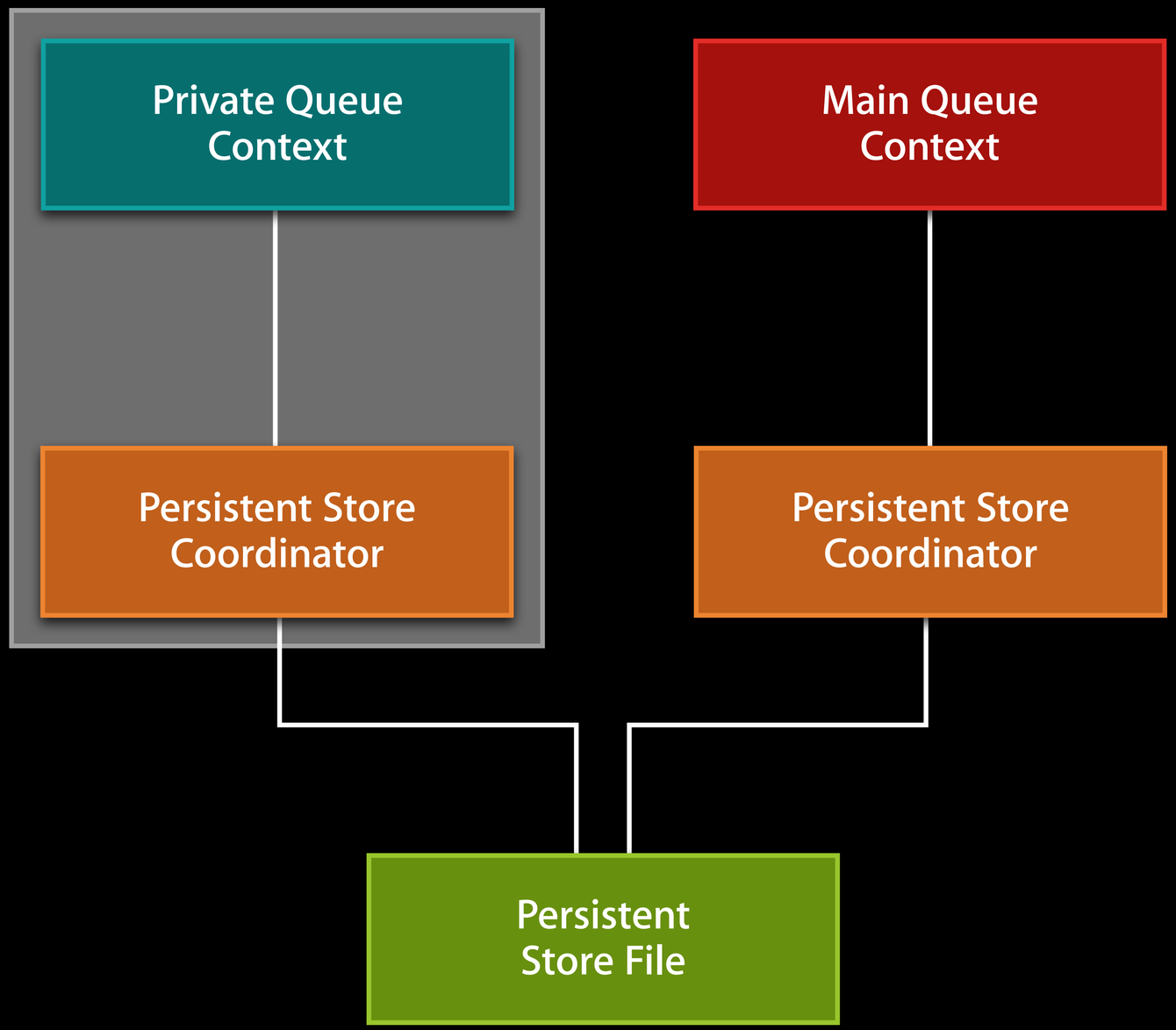
Concurrency with Core Data



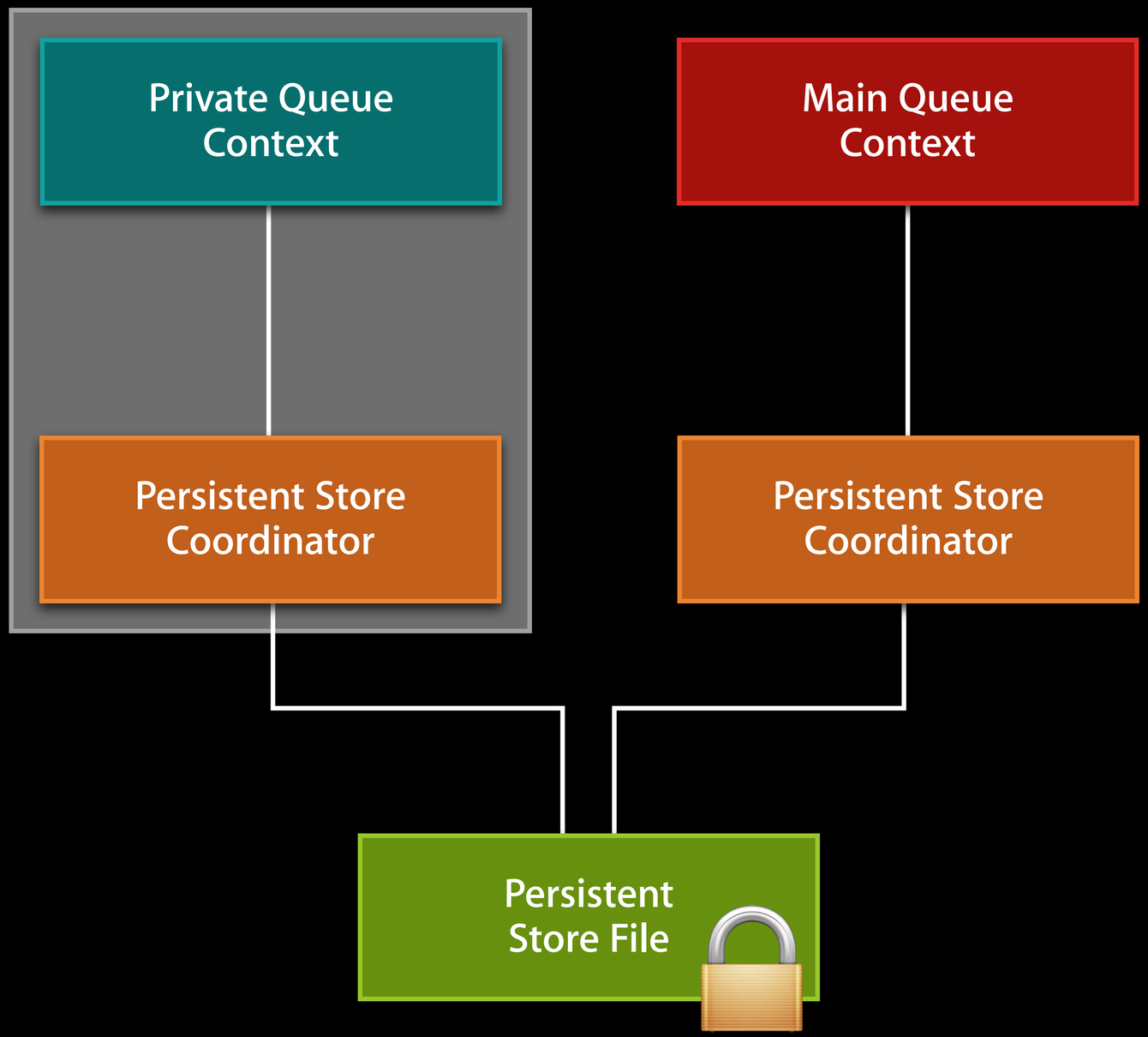
Concurrency with Core Data



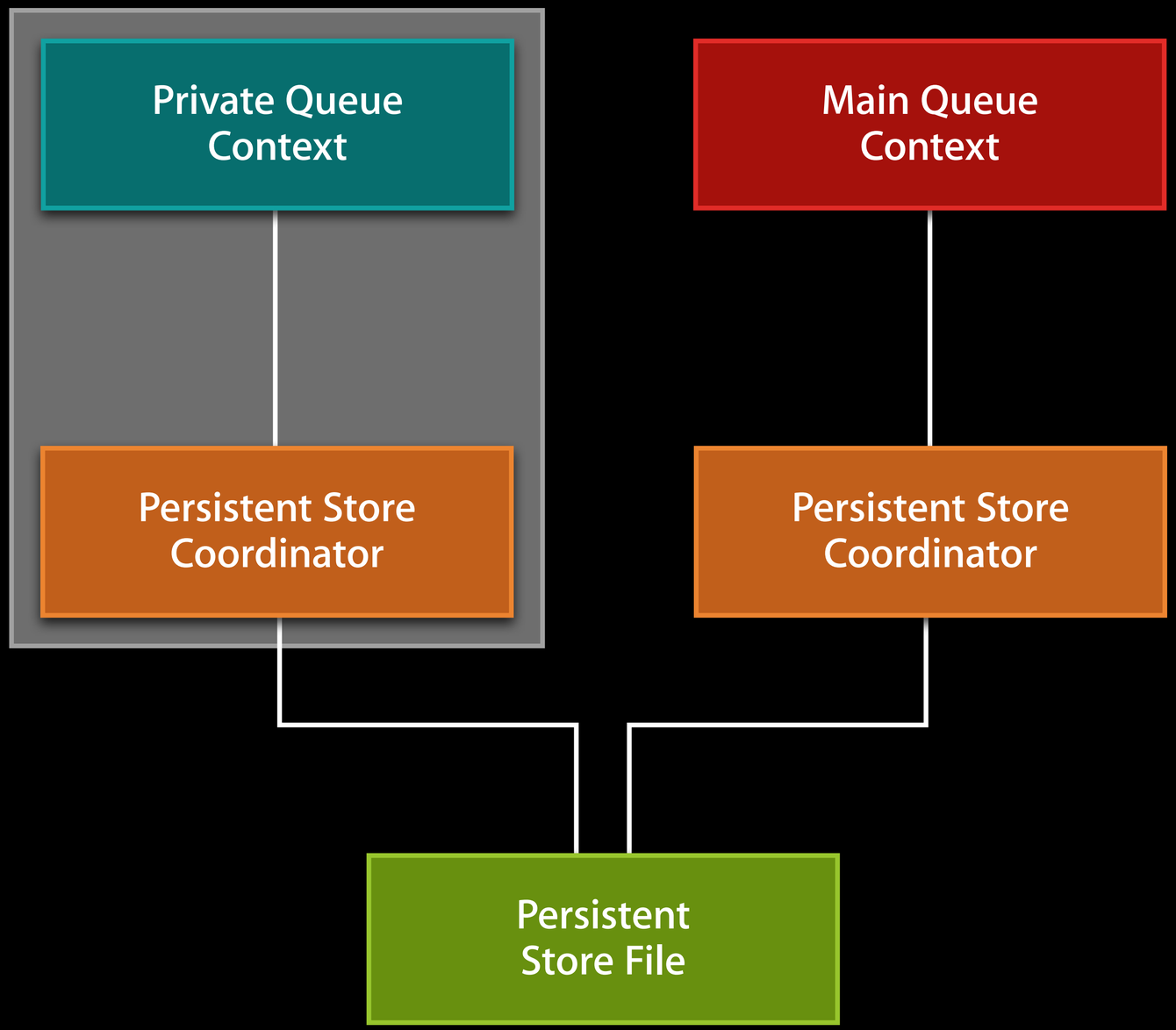
Concurrency with Core Data



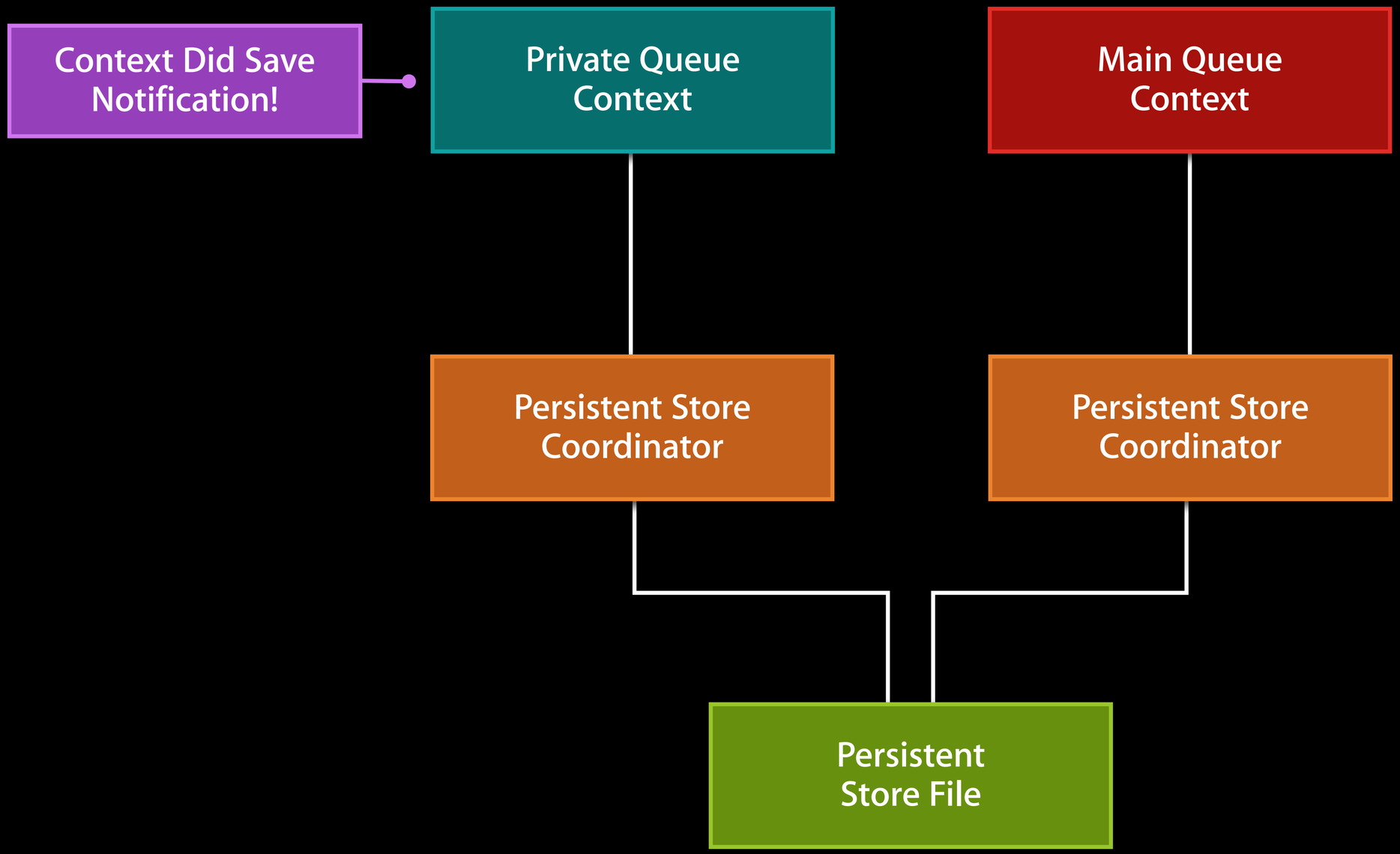
Concurrency with Core Data



Concurrency with Core Data



Concurrency with Core Data



Concurrency with Core Data

`-mergeChangesFromContextDidSaveNotification:`

Context Did Save Notification!

Private Queue Context

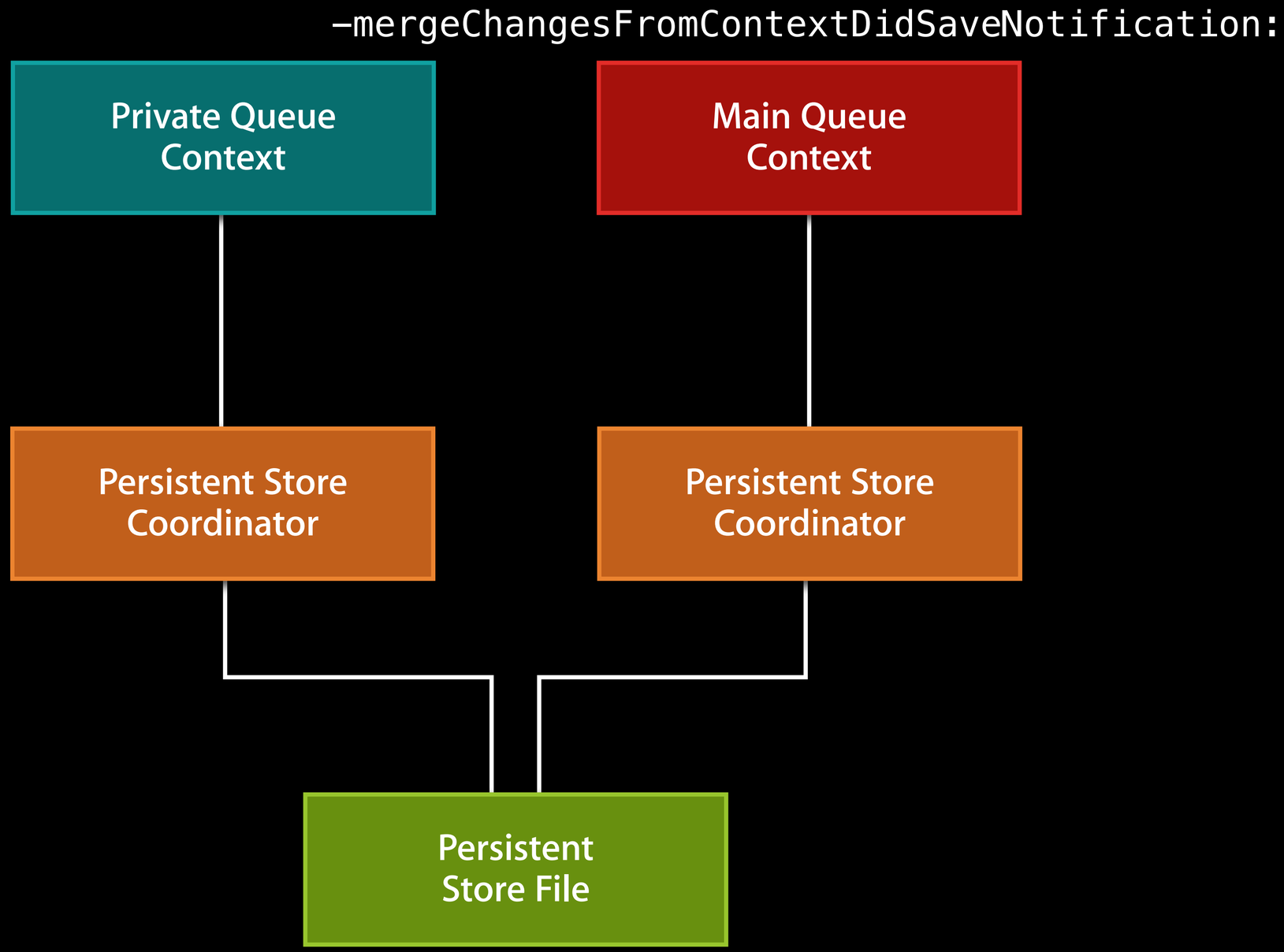
Main Queue Context

Persistent Store Coordinator

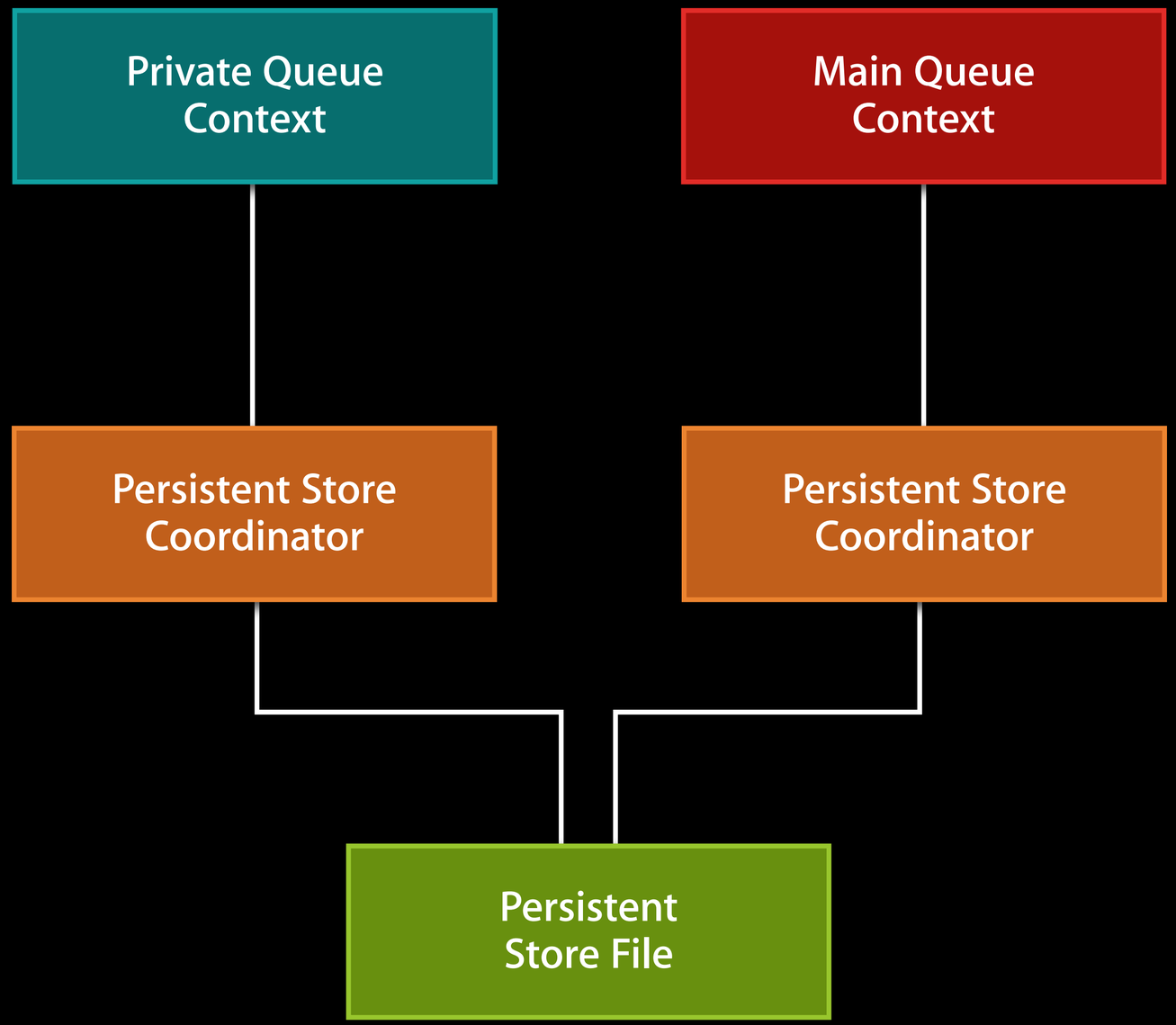
Persistent Store Coordinator

Persistent Store File

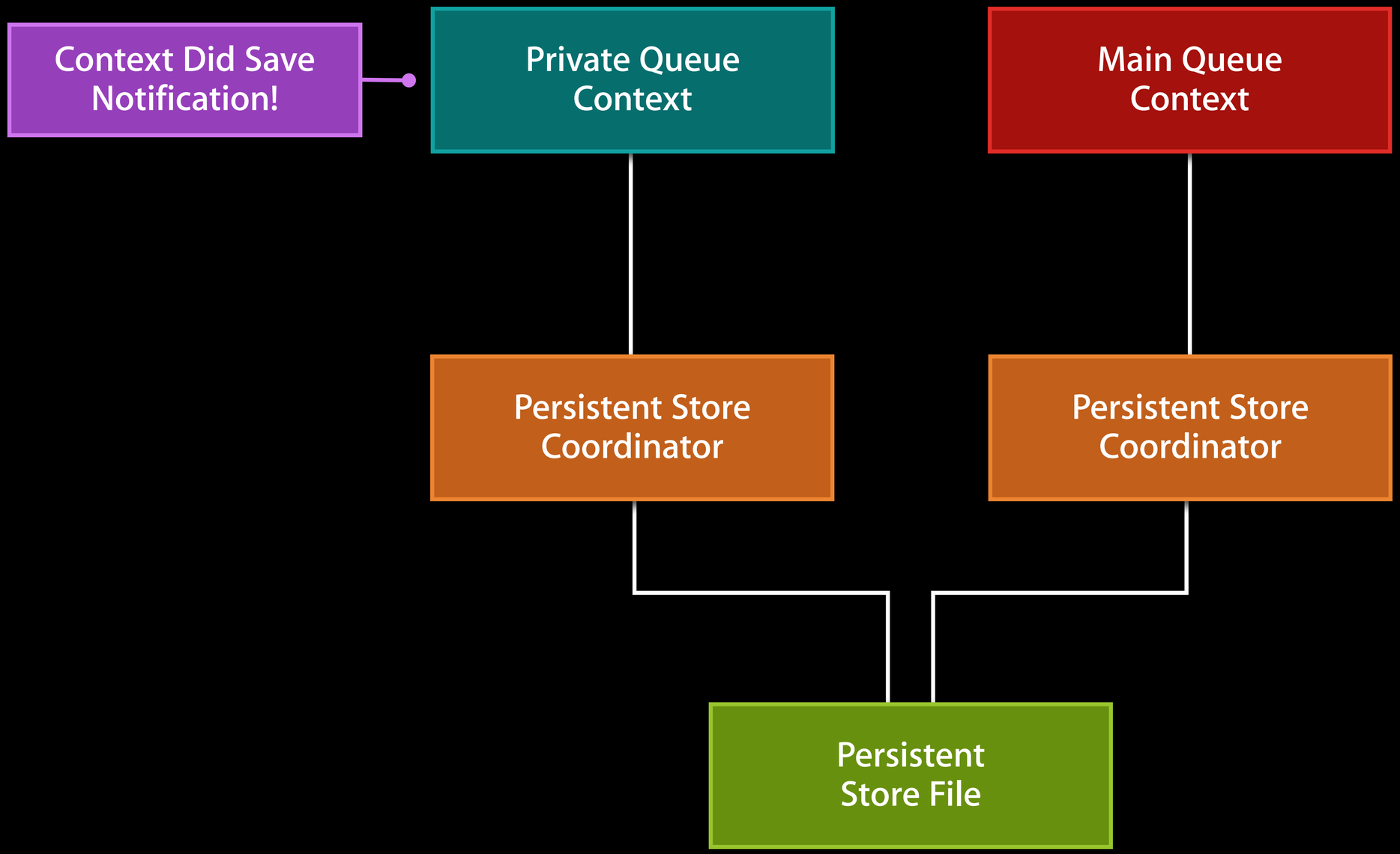
Concurrency with Core Data



Concurrency with Core Data



Concurrency with Core Data



Concurrency with Core Data

Refetch and Reload Data

Context Did Save Notification!

Private Queue Context

Main Queue Context

Persistent Store Coordinator

Persistent Store Coordinator

Persistent Store File

SQLite Write-Ahead Logging

- Supports multiple concurrent reads and one concurrent write
- Enabled by default on iOS 7 and OS X 10.9

SQLite Write-Ahead Logging

- Supports multiple concurrent reads and one concurrent write
- Enabled by default on iOS 7 and OS X 10.9
- Available in iOS 4+ and OS X 10.7+
  - Set options dictionary when adding a persistent store:
    - `@{ NSSQLitePragmasOption: @"journal_mode = WAL" }`

Efficient Text Queries

Optimizing Predicates

Optimizing Predicates

- Set a Predicate:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.predicate = [NSPredicate
    predicateWithFormat:@"firstName == %@ AND age > %i", @"John", 40];
```

Optimizing Predicates

- Set a Predicate:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.predicate = [NSPredicate
    predicateWithFormat:@"firstName == %@ AND age > %i", @"John", 40];
```

Optimizing Predicates

- Set a Predicate:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.predicate = [NSPredicate
    predicateWithFormat:@"firstName == %@ AND age > %i", @"John", 40];
```

Text comparison
is expensive

Optimizing Predicates

- Set a Predicate:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.predicate = [NSPredicate
predicateWithFormat:@"firstName == %@ AND age > %i", @"John", 40];
```

Text comparison is expensive

Numeric comparison is cheap

Optimizing Predicates

- Set a Predicate:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.predicate = [NSPredicate
        predicateWithFormat:@"age > %i && firstName == %@", 40, @"John"];
```

Optimizing Predicates

- Set a Predicate:

```
NSFetchRequest *request =
                    [NSFetchRequest fetchRequestWithEntityName:@"Contact"];

request.predicate = [NSPredicate
predicateWithFormat:@"age > %i && firstName == %@", 40, @"John"];
```

Put numeric comparison first

Predicate Costs

- In increasing cost:
- [cd] increases cost even more

Predicate Costs

- In increasing cost:
- [cd] increases cost even more

Predicate Costs

- In increasing cost:
- [cd] increases cost even more

Use Canonicalized Searches

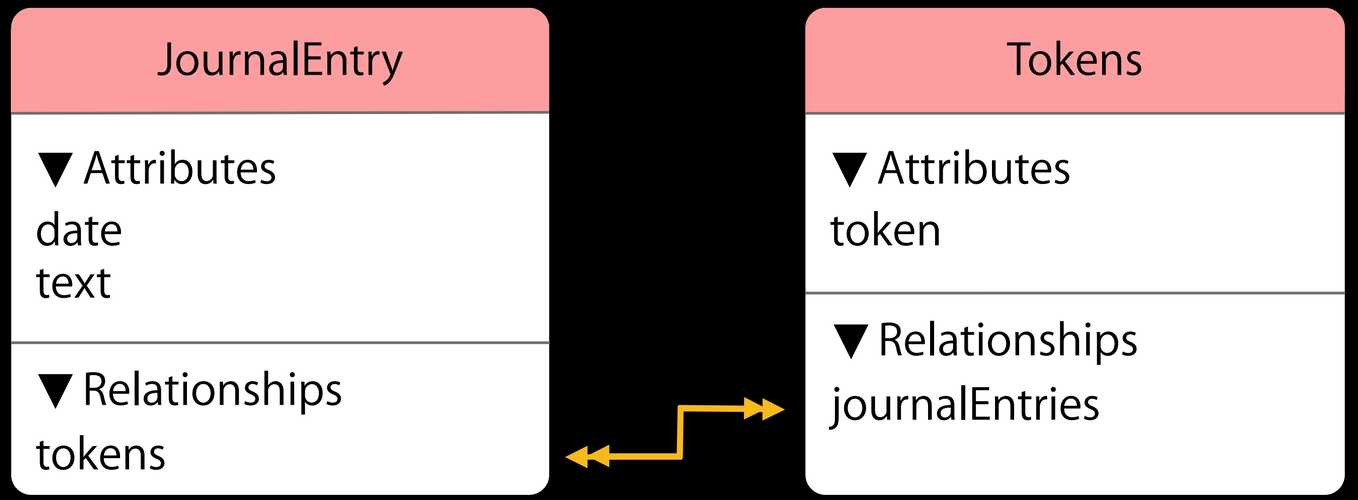
- Maintain a canonicalized text property
  - Set in custom accessor, whenever actual text is set
- Use a [n] query, and pass in the canonicalized query term

Canonicalized Tokens

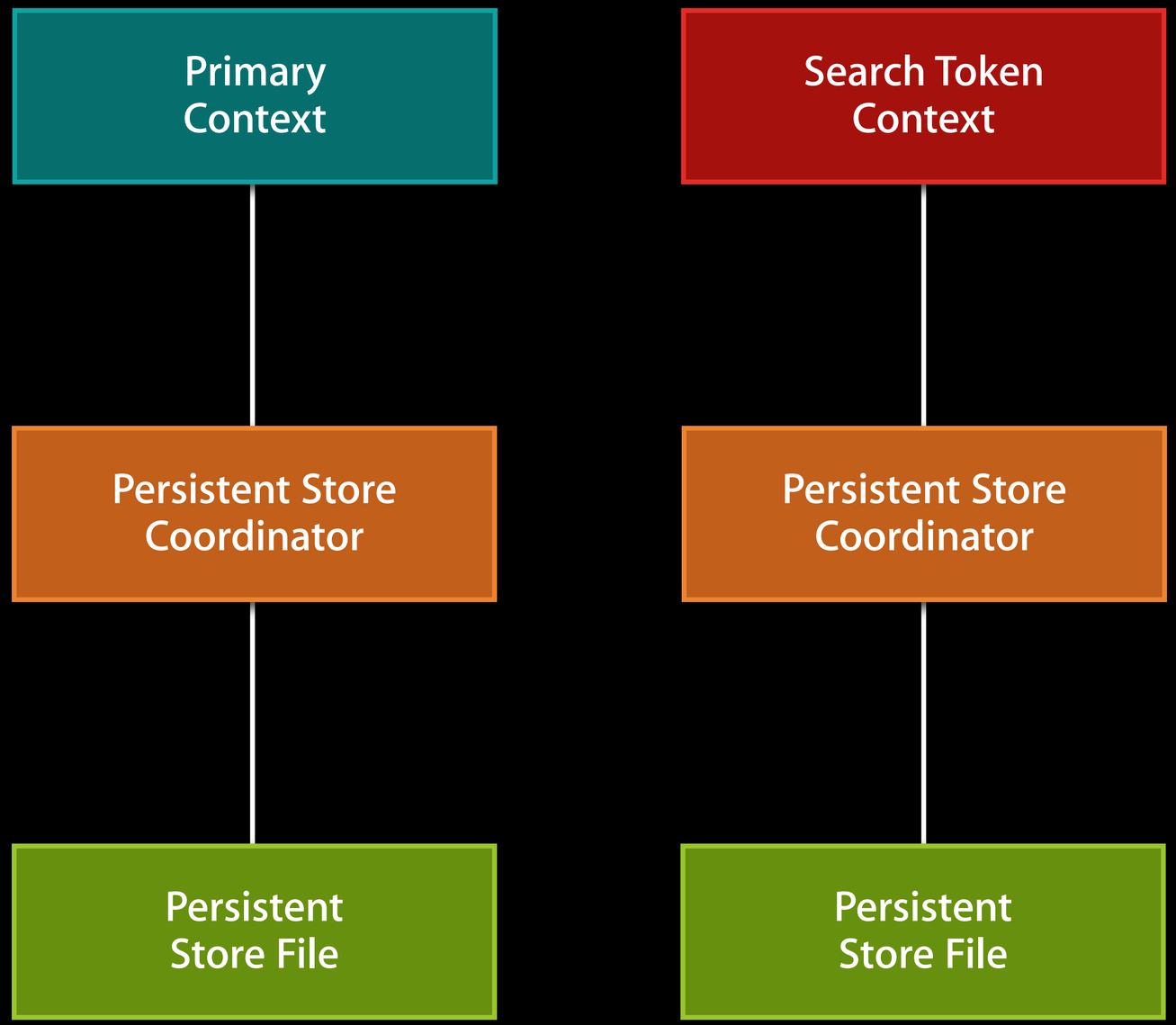
- Maintain separate entity for tokens
- Extract tokens from a string

  `–componentsSeparatedByCharactersInSet:`

  - Consider whitespace, symbols, punctuation

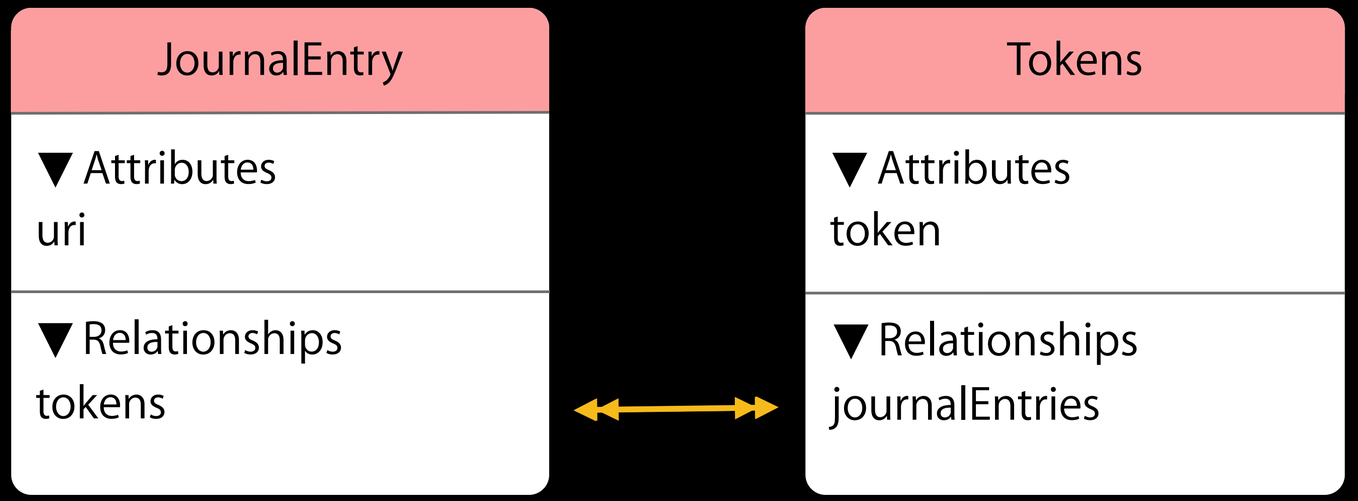


Use a Completely Separate Stack



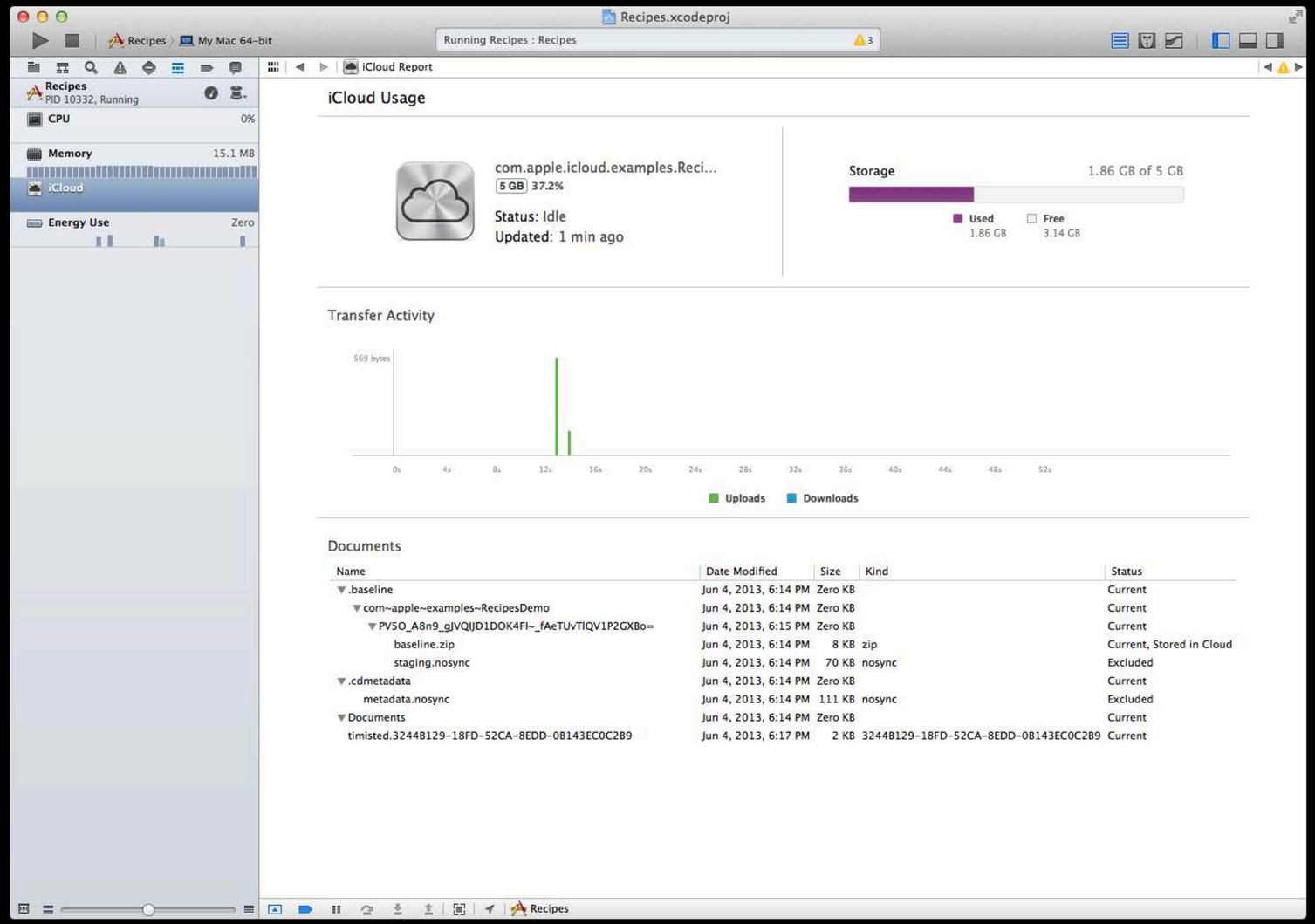
Use a Completely Separate Stack

- Run a separate Core Data stack just for the tokens
- Use URI representation to refer to your destination objects



Debugging Core Data with iCloud

New Debug Gauges in Xcode



What's New in Core Data

Pacific Heights
Wednesday 9:00AM

Using Ubiquity Logging

See what Core Data and iCloud are doing behind the scenes

- Pass argument on launch:

  `-com.apple.CoreData.Ubiquity.LogLevel 3`

- Use value of 1, 2, or 3

What's New in Core Data

Pacific Heights
Wednesday 9:00AM

Recap

Don’t Work Too Hard

- Measure everything first
- Leverage SQLite as much as possible
- Measure again
- Balance memory vs speed
- Optimize predicates, fetches, saves
- Measure again

Labs

Core Data Lab	Services Lab B Wednesday 3:15-6:00PM
Core Data Lab	Frameworks Lab A Thursday 2:00-4:15PM
Core Data Lab	Services Lab A Friday 9:00-11:15AM
Instruments and Performance Lab	Tools Lab B Thursday 2:00-4:15PM

More Information

Dave DeLong
Frameworks Evangelist
delong@apple.com

Documentation
Developer Library
http://developer.apple.com/

Apple Developer Forums
http://devforums.apple.com

WWDC2013