

High Performance Core Data

HighPerformanceCoreData.com

MatthewMorey.com | @xzolian

ChaiOne

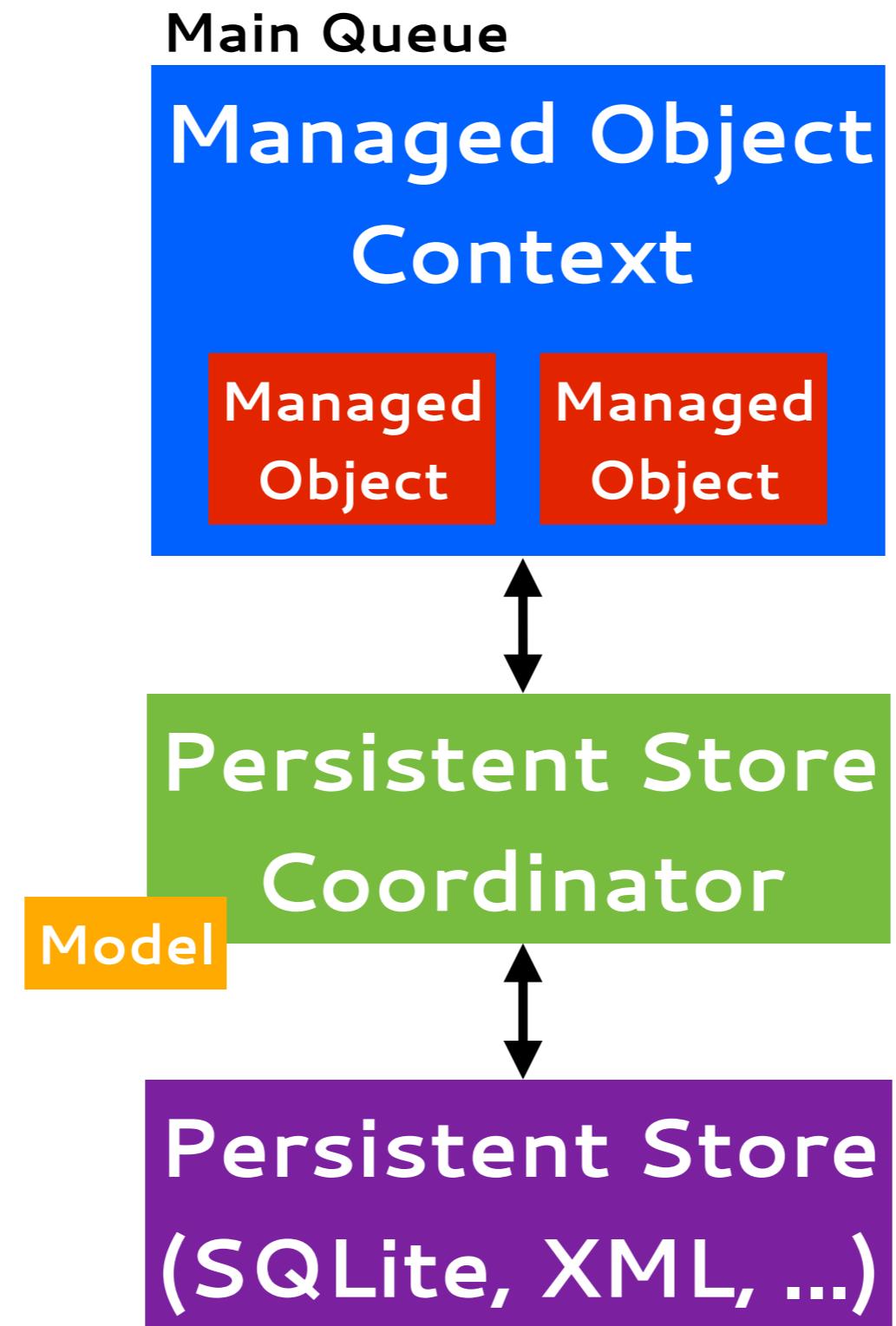




BuoyExplorer.com



Agenda



Memory

Less

More

Speed

Slow

Faster

Memory



Speed

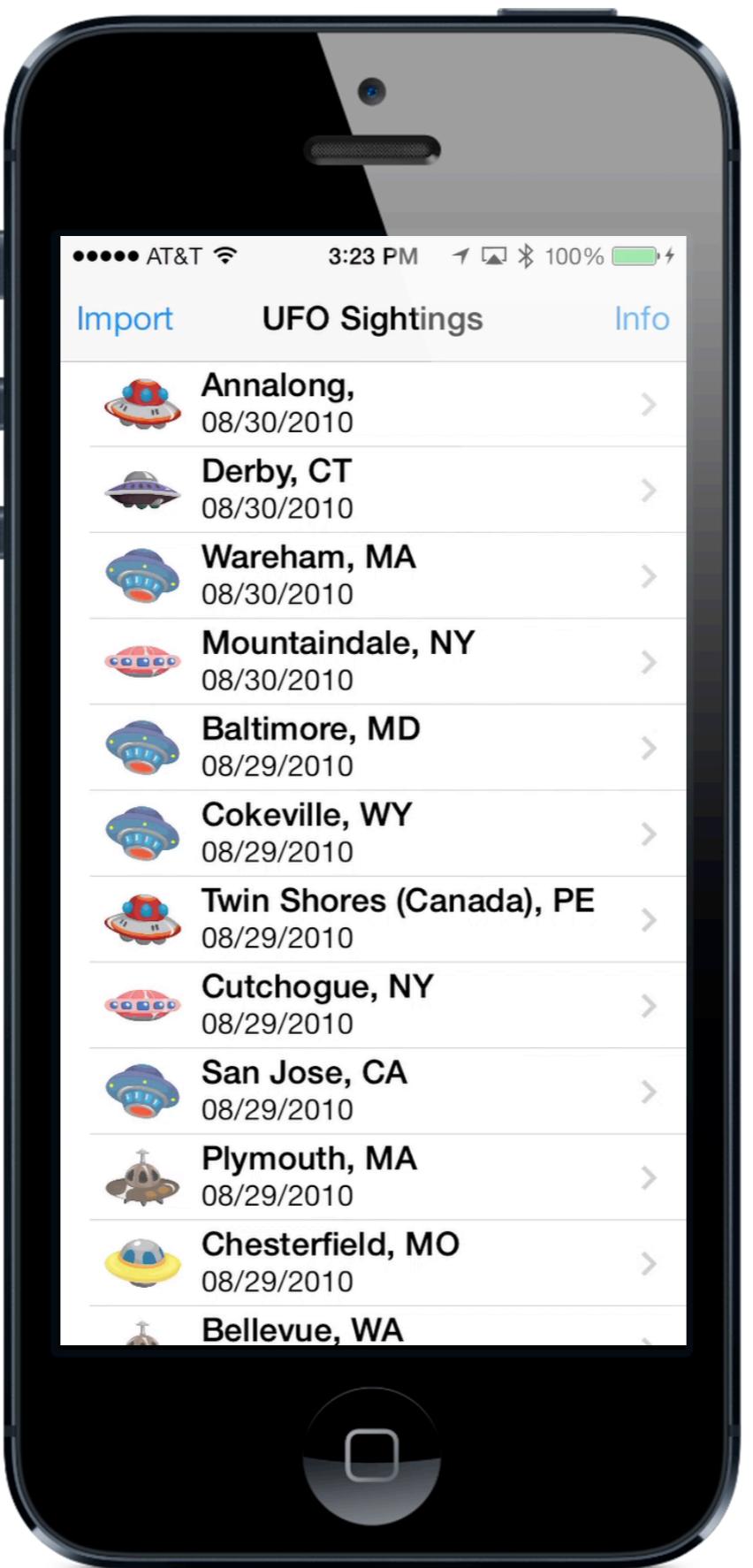


Memory



Speed





Tools

```
#define TICK NSDate *startTime = [NSDate date]
```

```
#define TOCK NSLog(@"Elapsed Time: %f",  
-[startTime timeIntervalSinceNow])
```

```
#define TICK NSDate *startTime = [NSDate date]

#define TOCK NSLog(@"Elapsed Time: %f",
-[startTime timeIntervalSinceNow])

. . .

TICK;
[self methodYouWantToMeasure];
TOCK;
```

-com.apple.CoreData.SQLDebug [1,2,3]

```
39
40 @implementation MDMUFOViewController
41
42 - (id)initWithStyle:(UITableViewStyle)style {
43
44     self = [super initWithStyle:style];
45     if (self) {
46         // Custom initialization
47     }
48     return self;
49 }
```

-- VTSUAL | TNF --

◀ ▶ ⏪ ⏵ ⏹ ⏸

© 2013 Pearson Education, Inc.

```
2013-11-02 10:12:43.284 MDMHPCoreData[74586:70b] CoreData: annotation: Connecting to sqlite database file at "/Users/xzolian/Library/Application Support/iPhone Simulator/7.0.3/Applications/D833E75D-7E6F-46DC-B394-B488C2FA5410/Documents/UFO.sqlite"
2013-11-02 10:12:43.286 MDMHPCoreData[74586:70b] CoreData: sql: pragma journal_mode=wal
2013-11-02 10:12:43.286 MDMHPCoreData[74586:70b] CoreData: sql: pragma cache_size=200
2013-11-02 10:12:43.287 MDMHPCoreData[74586:70b] CoreData: sql: SELECT Z_VERSION, Z_UUID, Z_PLIST FROM Z_METADATA
2013-11-02 10:12:43.297 MDMHPCoreData[74586:70b] CoreData: sql: SELECT 0, t0.Z_PK FROM ZUFOSIGHTING t0 ORDER BY t0.ZSIGHTED DESC
2013-11-02 10:12:49.322 MDMHPCoreData[74586:70b] CoreData: annotation: sql connection fetch time: 6.0256s
2013-11-02 10:12:49.323 MDMHPCoreData[74586:70b] CoreData: annotation: total fetch execution time: 6.0263s for 61392 rows.
2013-11-02 10:12:49.337 MDMHPCoreData[74586:70b] CoreData: sql: SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZDESC, t0.ZDURATION, t0.ZGUID, t0.ZLOCATION, t0.ZREPORTED, t0.ZSHAPE, t0.ZSIGHTED FROM ZUFOSIGHTING t0 WHERE t0.Z_PK IN (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) ORDER BY t0.ZSIGHTED DESC LIMIT 20
2013-11-02 10:12:49.338 MDMHPCoreData[74586:70b] CoreData: annotation: sql connection fetch time: 0.0009s
2013-11-02 10:12:49.339 MDMHPCoreData[74586:70b] CoreData: annotation: total fetch execution time: 0.0014s for 20 rows.
2013-11-02 10:13:58.993 MDMHPCoreData[74586:70b] CoreData: sql: SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZDESC, t0.ZDURATION, t0.ZGUID, t0.ZLOCATION, t0.ZREPORTED, t0.ZSHAPE, t0.ZSIGHTED FROM ZUFOSIGHTING t0 WHERE t0.Z_PK IN (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) ORDER BY t0.ZSIGHTED DESC LIMIT 20
2013-11-02 10:13:58.994 MDMHPCoreData[74586:70b] CoreData: annotation: sql connection fetch time: 0.0009s
2013-11-02 10:13:58.995 MDMHPCoreData[74586:70b] CoreData: annotation: total fetch execution time: 0.0013s for 20 rows.
2013-11-02 10:15:59.439 MDMHPCoreData[74586:70b] CoreData: sql: SELECT t0.ZSHAPE, COUNT( t0.ZSHAPE) FROM ZUFOSIGHTING t0 GROUP BY t0.ZSHAPE
2013-11-02 10:16:05.738 MDMHPCoreData[74586:70b] CoreData: annotation: sql connection fetch time: 6.2996s
2013-11-02 10:16:05.739 MDMHPCoreData[74586:70b] CoreData: annotation: total fetch execution time: 6.3003s for 31 rows.
2013-11-02 10:16:05.740 MDMHPCoreData[74586:70b] Elapsed Time: 6.308689
```

All Output

```
39
40 @implementation MDMUFOViewController
41
42 - (id)initWithStyle:(UITableViewStyle)style {
43
44     self = [super initWithStyle:style];
45     if (self) {
46         // Custom initialization
47     }
48     return self;
49 }
```

-- VTSUAL LTNE --

MDMHPCoreData

```
39
40 @implementation MDMUFOViewController
41
42 - (id)initWithStyle:(UITableViewStyle)style {
43
44     self = [super initWithStyle:style];
45     if (self) {
46         // Custom initialization
47     }
48     return self;
49 }
```

```
/Users/xzolian/xcode/projects/MDMHPCoreData/2/MDMHPCoreData/MDMHPCoreData/
-- VTSUAI | TNF --
MDMHPCoreData
2013-11-02 10:12:43.284 MDMHPCoreData[74586:70b] CoreData: annotation: Connecting to sqlite database file at "/Users/xzolian/Library/Application Support/iPhone Simulator/7.0.3/Applications/D833E75D-7E6F-46DC-B394-B488C2FA5410/Documents/UFO.sqlite"
2013-11-02 10:12:43.286 MDMHPCoreData[74586:70b] CoreData: sql: pragma journal_mode=wal
2013-11-02 10:12:43.286 MDMHPCoreData[74586:70b] CoreData: sql: pragma cache_size=200
2013-11-02 10:12:43.287 MDMHPCoreData[74586:70b] CoreData: sql: SELECT Z_VERSION, Z_UUID, Z_PLIST FROM Z_METADATA
2013-11-02 10:12:43.297 MDMHPCoreData[74586:70b] CoreData: sql: SELECT 0, t0.Z_PK FROM ZUFOSIGHTING t0 ORDER BY t0.ZSIGHTED DESC
2013-11-02 10:12:49.322 MDMHPCoreData[74586:70b] CoreData: annotation: sql connection fetch time: 6.0256s
2013-11-02 10:12:49.323 MDMHPCoreData[74586:70b] CoreData: annotation: total fetch execution time: 6.0263s for 61392 rows.
2013-11-02 10:12:49.337 MDMHPCoreData[74586:70b] CoreData: sql: SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZDESC, t0.ZDURATION, t0.ZGUID, t0.ZLOCATION, t0.ZREPORTED, t0.ZSHAPE, t0.ZSIGHTED FROM ZUFOSIGHTING t0 WHERE t0.Z_PK IN (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) ORDER BY t0.ZSIGHTED DESC LIMIT 20
2013-11-02 10:12:49.338 MDMHPCoreData[74586:70b] CoreData: annotation: sql connection fetch time: 0.0009s
2013-11-02 10:12:49.339 MDMHPCoreData[74586:70b] CoreData: annotation: total fetch execution time: 0.0014s for 20 rows.
2013-11-02 10:13:58.993 MDMHPCoreData[74586:70b] CoreData: sql: SELECT 0, t0.Z_PK, t0.Z_OPT, t0.ZDESC, t0.ZDURATION, t0.ZGUID, t0.ZLOCATION, t0.ZREPORTED, t0.ZSHAPE, t0.ZSIGHTED FROM ZUFOSIGHTING t0 WHERE t0.Z_PK IN (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) ORDER BY t0.ZSIGHTED DESC LIMIT 20
2013-11-02 10:13:58.994 MDMHPCoreData[74586:70b] CoreData: annotation: sql connection fetch time: 0.0009s
2013-11-02 10:13:58.995 MDMHPCoreData[74586:70b] CoreData: annotation: total fetch execution time: 0.0013s for 20 rows.
2013-11-02 10:15:59.439 MDMHPCoreData[74586:70b] CoreData: sql: SELECT t0.ZSHAPE, COUNT( t0.ZSHAPE) FROM ZUFOSIGHTING t0 GROUP BY t0.ZSHAPE
2013-11-02 10:16:05.738 MDMHPCoreData[74586:70b] CoreData: annotation: sql connection fetch time: 6.2996s
2013-11-02 10:16:05.739 MDMHPCoreData[74586:70b] CoreData: annotation: total fetch execution time: 6.3003s for 31 rows.
2013-11-02 10:16:05.740 MDMHPCoreData[74586:70b] Elapsed Time: 6.308689
```

-com.apple.CoreData.SQLDebug [1,2,3]

-com.apple.CoreData.SyntaxColoredLogging 1

-com.apple.CoreData.MigrationDebug 1

-com.apple.CoreData.SQLiteDebugSynchronous [0,1,2]

-com.apple.CoreData.SQLiteIntegrityCheck [1]

-com.apple.CoreData.ThreadingDebug [1,2,3]



\$ sqlite3

```
$ sqlite3 UFO.sqlite

sqlite> select * from sqlite_master;

table|ZUFOSIGHTING|ZUFOSIGHTING|3|CREATE TABLE
          ZUFOSIGHTING ( Z_PK...VARCHAR )

table|Z_PRIMARYKEY|Z_PRIMARYKEY|4|CREATE TABLE
          Z_PRIMARYKEY (Z_ENT...INTEGER)

table|Z_METADATA|Z_METADATA|5|CREATE TABLE
          Z_METADATA (Z_VERSION...BLOB)
```

```
Documents — bash — 83x28

sqlite> SELECT t0.ZSHAPE, COUNT( t0.ZSHAPE )
      FROM ZUFOSIGHTING t0 GROUP BY t0.ZSHAPE;

changed|1
changing|1546
chevron|760
cigar|1782
circle|5271
cone|265
...
teardrop|595
triangle|6082
unknown|4490
```



Pony Debugger

Screenshot of the Chrome DevTools Dev Storage panel showing IndexedDB data.

The left sidebar lists databases:

- Frames
- Web SQL
- IndexedDB
 - Twitter Test MOC - com.s...
 - Tweet
 - User
 - Local Storage
 - Session Storage
 - Cookies
 - Application Cache

The "User" object under the "Twitter Test MOC" database is selected in the sidebar.

The main area displays the contents of the "User" object:

#	Key (Ke...)	Value
150	"x-core..."	► User
151	"x-core..."	▼ User <ul style="list-style-type: none">class: "PDUser"name: "Greg Marsh"profilePictureURL: "http://a0.twimg.com/profile_images/323435245/remoteID: 22956380screenName: "gmarsh17"tweets: _NSFaultingMutableSet<Tweet> <count: 1><ul style="list-style-type: none">▼ 0: PDTweet<ul style="list-style-type: none">class: "PDTweet"remoteID: 239132921074692100retrievalDate: 2012-08-24 23:16:45 +0000text: "@jenfromkidcity @RoughHouseFood @NoRACupcakeCo Amazing► user: Usercount: 1
152	"x-core..."	► User
153	"x-core..."	► User
154	"x-core..."	► User
155	"x-core..."	► User
156	"x-core..."	► User



SQLite **Professional**

UFO.sqlite — Edited

Execute Schema Data Query Structure Export

Filter

Tables

- ZUFOSIGHTING 61392 Rows
- Z_METADATA 1 Row
- Z_PRIMARYKEY 1 Row

Views

ZSIGHTED	ZDESC	ZDURATION	ZGUID	ZLOCATION	ZSHA
5200	Slow mo...	4-1/2minutes	21ec9972-2ff6-401...	Peru, IN	egg
4800	2 lights a...	10 seconds	c206a34b-0d59-48...	Plano, TX	light
5600	seven ob...	15 minutes	237ad81b-851b-43...	Liberty Hill, TX	light
5200	Large Bo...	35 seconds	67bd7dff-89db-469...	Houston, TX	formation
4800	Cluster o...	5 seconds	5d00c036-b9de-46...	Michigan City, IN	triangle
200	STRANGE...		46e183a5-fc0e-4b9...	Crown Point, IN	disk
6400	3 u.f.os...	2 minutes	9cb61158-6b97-4b...	Wabash, IN	light
200	4 bright...	7 seconds	42fb9469-95a0-4de...	Plover, WI	light
400	Stationar...	100 minutes	f8c28c8f-ad13-424...	Albuquerque, NM	unknown
8400	i was dri...	10 minutes	145b7e27-1c2c-4d...	Salt Lake City, UT	unknown
3200	don&apo...	5 min	9ef1e7f7-6318-4fd...	Winamac, IN	light
9600	egg-sha...	three minutes	04798678-cfc7-434...	Acworth, GA	sphere
3200	Object w...	1hr(still occurring)	997678f2-cef5-405...	Costa Mesa, CA	light
4400	Just unde...	ongoing	3a6c737b-10d3-4b...	Oshkosh, WI	fireball
6800	A STRAN...		8530ce0e-606a-45d...	Egypt (I can not tell y...	light
56800	Strange...	about 10 sec	125af3ec-08f3-4f06...	Skarysko-Kamienna (...	triangle
6400	You do r...	all my life	48e02263-ab36-4c...	Yakima, WA	light
4400	Bright cir...	2 minutes	d67cd667-90bd-45...	Hollywood, FL	sphere
000	AN ORA...	15 SECONDS	f3fbe1d1-1eea-44ca...	Wahpeton, ND	light
2800	2 Object...	5 minutes	d8a8ebea-27a5-446...	St. Louis Park, MN	formation
8000	Two brig...	a few minutes	ed02c7ca-bd55-40...	Bronxville, NY	light
800	Triangle...	2-3 minutes	7a415881-27e8-42...	Springfield, MO	triangle
30800	Large (2...	0:02 (Min.)	45c6d396-90f7-41d...	Van Nuys (Los Angel...	triangle
92000	Triangula...	1 hour	41ba31d0-01a1-4b...	Alston, MI	triangle
122800	9/17/19...	4-5 seconds	3d472f27-eac7-4c4...	Long Beach (Termina...	other
4800	Bright fla...	2 sseconds	f0044beb-dd9e-401...	San Diego, CA	flash
96800	I knew w...	over several days	063fc682-8504-490...	Dayton, OH	other
1600	White ov...	15 seconds	23fbcc30-d165-49f...	Tampa, FL	oval
400	((NUFOR...		626f4df2-d8ad-4f8f...	Soldatna, AK	disk
0000	Chevron...	3-4 seconds	fad1d542-a63b-4b7...	Las Vegas, NV	chevron
5600	Repeated...	10-15 mts	29457f31-7aa4-4fc...	South Bend, IN	rectangle
84400	i saw a s...	visiting	37f5096c-bda7-422...	Two Egg Mariana, FL	circle
8800	having ju...	2 seconds	1b3c3eb5-e331-4e7...	Seattle (Belmont &a...	other



Edit

61392 records. 0.491 seconds.

Filter

UFO.sqlite — Edited

Execute

Filter

Tables

- ZUFOSIGHTING
61392 Rows
- Z_METADATA
1 Row
- Z_PRIMARYKEY
1 Row

Views

1 SELECT t0.ZSHAPE, COUNT(t0.ZSHAPE) FROM ZUFOSIGHTING t0 GROUP BY t0.ZSHAPE

Star Filter Refresh

ZSHAPE	COUNT(t0.ZSHAPE)
fireball	3452
flare	1
flash	990
formation	1788
hexagon	1
light	12202
other	4593
oval	2869
pyramid	1
rectangle	966
round	2
sphere	3637
teardrop	595
triangle	6082
	1100

+ Settings

31 records. 0.139 seconds.

Schema Data Query Structure

Export



Core Data Editor Version 5

Untitled — Edited



Configuration CSV Import Generate Code

ENTITIES	Object ID	Desc	Duration	Guid	Location	Reported	Shape	Sighted
E UFO Sighting 61392	p1	Siting of a...	5 minutes	c49497c8...	Wimberley...	4/8/04, 1...	triangle	2/25/90, 1...
	p2	3:00am h...	5minutes	f3006465...	Knox, IN	12/23/05...	triangle	12/18/05,...
	p3	I'm...	20 sec	0f72719e...	St-Louis d...	11/26/01...	light	11/18/01,...
	p4	White fire...	45 seconds	4f1a8ec9...	Cerrillos, TN	10/10/06...	changing	10/1/06, 1...
	p5	Likely exp...		d41a6a72...	Lake Wale...	2/14/05,...		1/24/05, 1...
	p6	Possible U...	1-2 Min	04168c50...	Fayettevill...	1/14/09,...	light	9/30/08, 1...
	p7	We were a...	2 to 3 min.	65df1341...	Camden P...	8/12/02,...	unknown	8/10/02, 1...
	p8	Strange n...	About 7...	79fe5386...	Jesup, GA	5/18/10,...	unknown	5/17/10, 1...
	p9	I observed...	10 minutes	5de42d76...	Westchest...	1/16/00,...	triangle	1/11/00, 1...
	p10	Stationary...	over an hour	334c1250...	State Coll...	5/9/10, 1...	flash	5/9/10, 12...
	p11	Multicolor...	2 or more...	9a858a1a...	Allentown...	10/9/06,...	sphere	10/9/06, 1...
	p12	My girlfrie...	10 seconds	4f0debd7...	Basom, NY	7/7/05, 1...	fireball	3/12/05, 1...
	p13	Pink anom...	10 minutes	54d58bbc...	Rolla, MO	10/28/01...	light	10/28/01,...
	p14	TheFull*M...	5-6 seconds	6d0ff5e3...	Portland (...	6/19/00,...	unknown	6/16/00, 1...

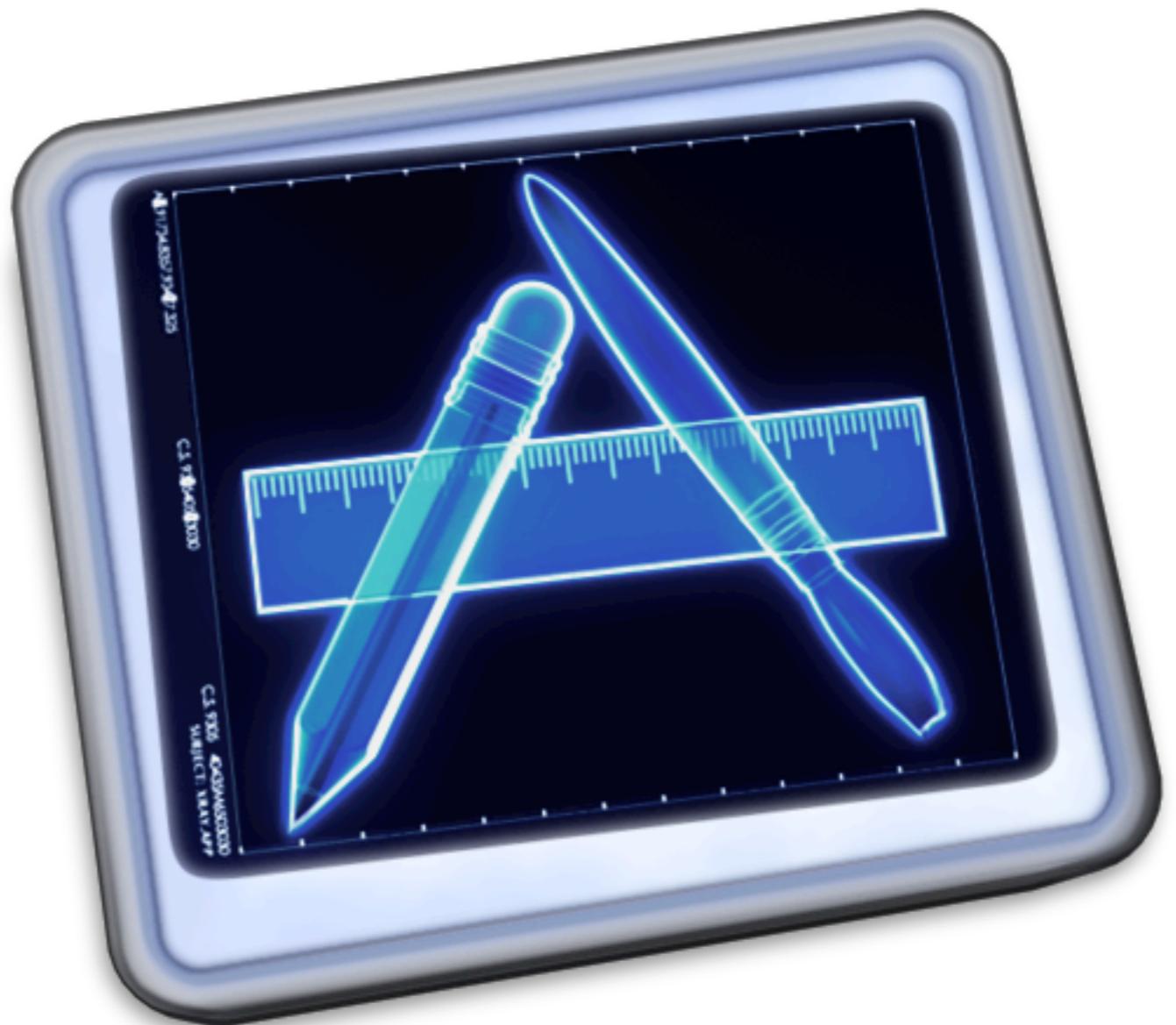
+ -

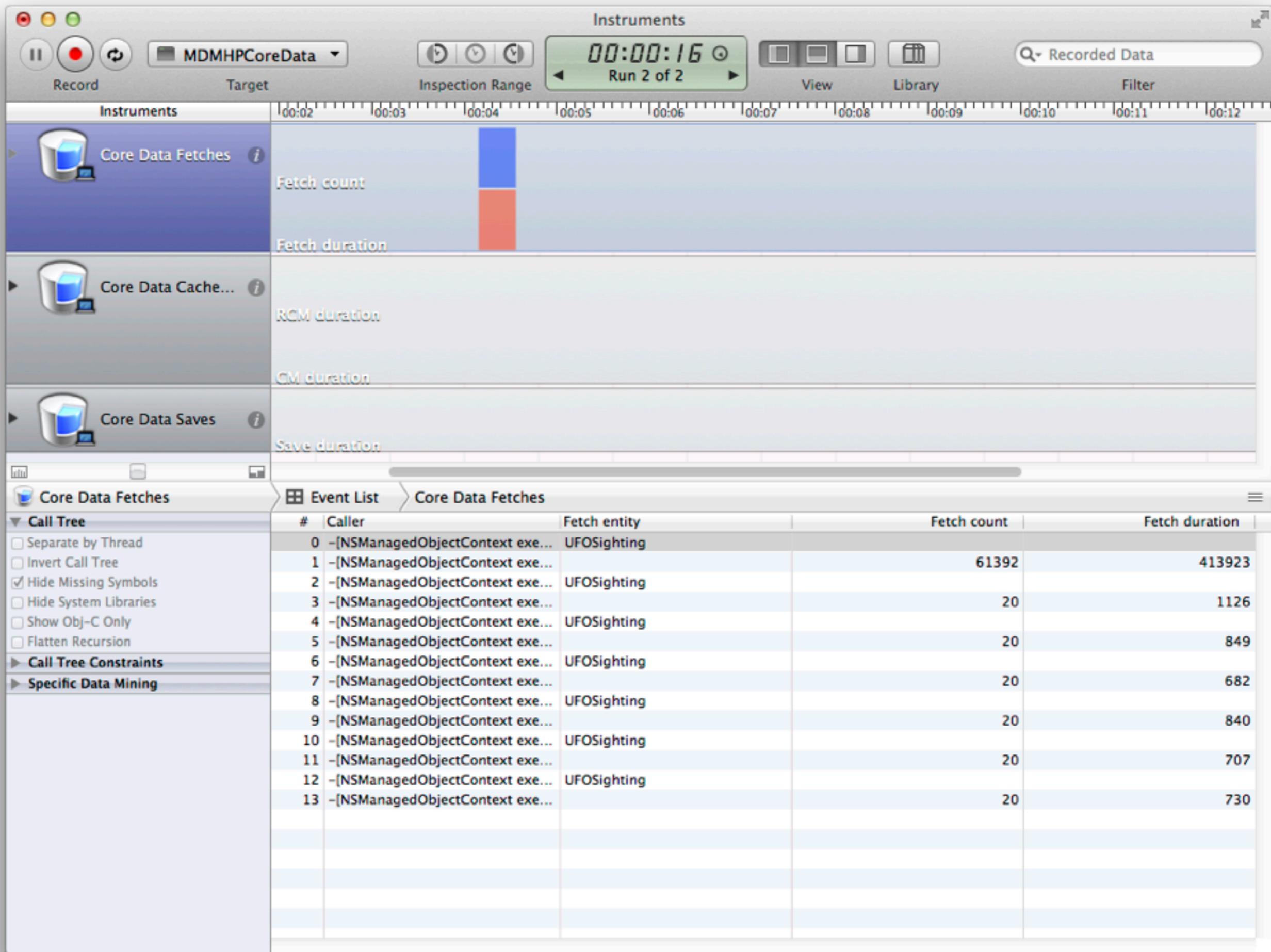
Q

RELATIONSHIPS

No Validation Errors

No Selection





Tools

- Macros
- Launch Arguments
- SQL
- Instruments
- Measure, Measure, Measure

Fetching

MDMHPCoreData.xcodeproj — MDMUFOListViewController.m

iPhone Retina (4-inch) | Finished running MDMHPCoreData on iPhone Retina (4-inch) | No Issues

MDMHPCoreData > MDMHPCoreData > controller > MDMUFOListViewController.m > -setupTableDataSource

```
75 - (void)setupTableDataSource {  
76  
77     NSAssert(self.managedObjectContext, @"Forgot to set managed  
78     NSFetchedResultsController *fetchedResultsController = [[NSFetchedResultsController alloc] init];  
79     [fetchedResultsController setFetchRequest:[NSFetchRequest fetchRequestWithEntityName:@"UFO"]];  
80     [fetchedResultsController setSortDescriptors:@[[NSSortDescriptor sortDescriptorWithKey:@"date" ascending:NO]]];  
81     self.dataSource = [[MDMFetchedResultsTableDataSource alloc] initWithFetchedResultsController:fetchedResultsController];  
82     __typeof(self) __weak weakSelf = self;  
83     self.dataSource.configureCellBlock = ^(MDMUFOSightingCell * _Nonnull cell, MDMUFOSighting *sighting) {  
84         [weakSelf configureCell:cell withUFOSighting:sighting];  
85     };  
86     self.tableView.dataSource = self.dataSource;  
87     [self.tableView reloadData];  
88 }  
89  
90 - (void)configureCell:(MDMUFOSightingCell *)cell withUFOSighting:(MDMUFOSighting *)sighting {  
91     cell.textLabel.text = sighting.title;  
92     cell.detailTextLabel.text = [NSString stringWithFormat:@"%@ (%.2f, %.2f)",  
93         sighting.location.latitude, sighting.location.longitude, sighting.date];  
94 }
```

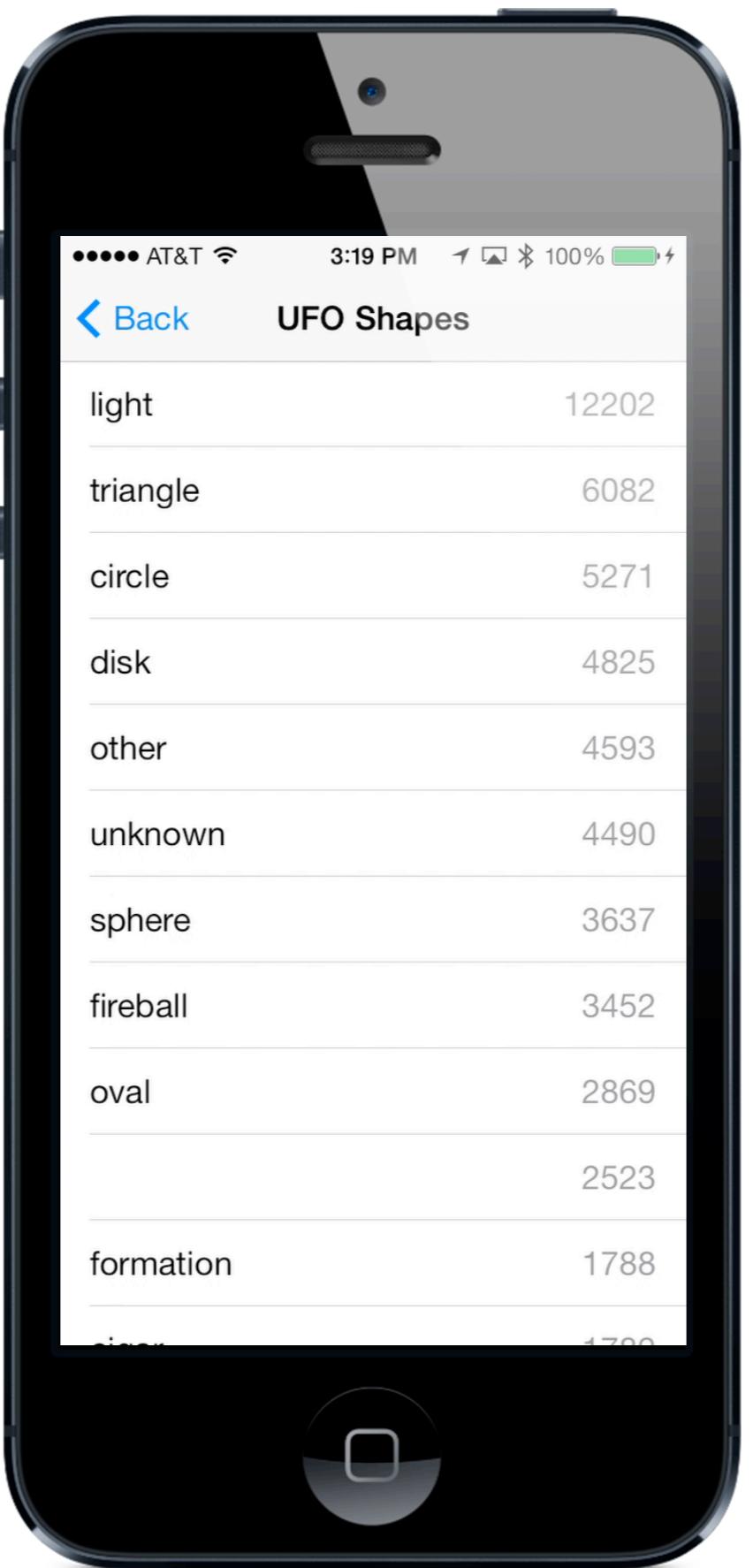
No Debug Session

/Users/xzolian/xcode/projects/MDMHPCoreData/2/MDMHPCoreData/

-- TNSERT --

```
NSFetchRequest *fetchRequest =  
    [NSFetchRequest fetchRequestWithEntityName:@"UFOsighting"];  
  
[fetchRequest setFetchBatchSize:20];
```

```
74  
75 - (void)setupTableDataSource {  
76  
77     NSAssert(self.managedObjectContext, @"Forgot to set managed  
78     NSFetchRequest *fetchRequest = [NSFetchRequest fetchRequestWithEntityName:  
79     [fetchRequest setSortDescriptors:@[[NSSortDescriptor sortDescriptorWithKey:  
80         @"name" ascending:YES]]];  
81     }  
82  
83     NSFetchedResultsController *fetchedResultsController = [[NSFetchedResultsController alloc]  
84     initWithFetchRequest:fetchRequest  
85     managedObjectContext:self.managedObjectContext  
86     sectionNameKeyPath:@"name"  
87     cacheName:@"fetchedResults"];  
88     self.dataSource = [[MDMFetchedResultsTableDataSource alloc]  
89     initWithStyle:UITableViewStylePlain];  
90     __typeof(self) __weak weakSelf = self;  
91     self.dataSource.configureCellBlock = ^(MDMUFOSightingCell *cell)  
92     [weakSelf configureCell:cell withUFOsighting:sighting];  
93 };  
94  
95     self.tableView.dataSource = self.dataSource;  
96     [self.tableView reloadData];  
97 }  
98  
99
```



NSManagedObject

```
NSFetchRequest *fetchRequest =
[NSFetchRequest fetchRequestWithEntityName:@"UFOSighting"];

NSArray *UFOSightingsArray =
[self.managedObjectContext executeFetchRequest:fetchRequest
                                      error:NULL];

NSMutableDictionary *uniqueShapesDictionary =
NSMutableDictionary dictionary;

for (UFOSighting *sighting in UFOSightingsArray) {
    ...
    // Count unique shape items
    ...
}
```

NSDictionaryResultType

```
NSFetchRequest *fetchRequest =  
[NSFetchRequest fetchRequestWithEntityName:@"UFOsighting"];  
  
[fetchRequest setResultType:NSDictionaryResultType];  
[fetchRequest setPropertiesToFetch:@"shape"];
```

NSExpression

```
NSFetchRequest *fetchRequest =
[NSFetchRequest fetchRequestWithEntityName:@"UFOsighting"];

NSEntityDescription *entityDescription =
[[NSEntityDescription alloc] init];

[entityDescription setName:@"count"];
[entityDescription setExpression:
[NSEntityDescription expressionForFunction:@"count:" arguments:
@[[NSEntityDescription expressionForKeyPath:@"shape"]]]];

[fetchRequest setPropertiesToFetch:@[@"shape", entityDescription]];
[fetchRequest setPropertiesToGroupBy:@[@"shape"]];
[fetchRequest setResultType:NSDictionaryResultType];
```

Reader
PDF

iOS Developer Library
Developer
Search

NSExpression Class Reference

Table of Contents
Jump To...

Function	Parameter	Returns	Availability
<code>average:</code>	An <code>NSArray</code> object containing <code>NSExpression</code> objects representing numbers	An <code>NSNumber</code> object (the average of values in the array)	OS X v10.4 and later
<code>sum:</code>	An <code>NSArray</code> object containing <code>NSExpression</code> objects representing numbers	An <code>NSNumber</code> object (the sum of values in the array)	OS X v10.4 and later
<code>count:</code>	An <code>NSArray</code> object containing <code>NSExpression</code> objects representing numbers	An <code>NSNumber</code> object (the number of elements in the array)	OS X v10.4 and later
<code>min:</code>	An <code>NSArray</code> object containing <code>NSExpression</code> objects representing numbers	An <code>NSNumber</code> object (the minimum of the values in the array)	OS X v10.4 and later
<code>max:</code>	An <code>NSArray</code> object containing <code>NSExpression</code> objects representing numbers	An <code>NSNumber</code> object (the maximum of the values in the array)	OS X v10.4 and later
<code>median:</code>	An <code>NSArray</code> object containing <code>NSExpression</code> objects representing numbers	An <code>NSNumber</code> object (the median of the values in the array)	OS X v10.5 and later
<code>mode:</code>	An <code>NSArray</code> object containing <code>NSExpression</code> objects representing numbers	An <code>NSArray</code> object (the mode of the values in the array)	OS X v10.5 and later
	An <code>NSArray</code> object		

Revision History
Predicate Programming Guide

Prefetch Required Relationships

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

Fetching

- Don't fetch more than you need
- Set a batch size
- Let SQLite do the calculations
- Prefetch required relationships

Predicates

Light Comparisons First

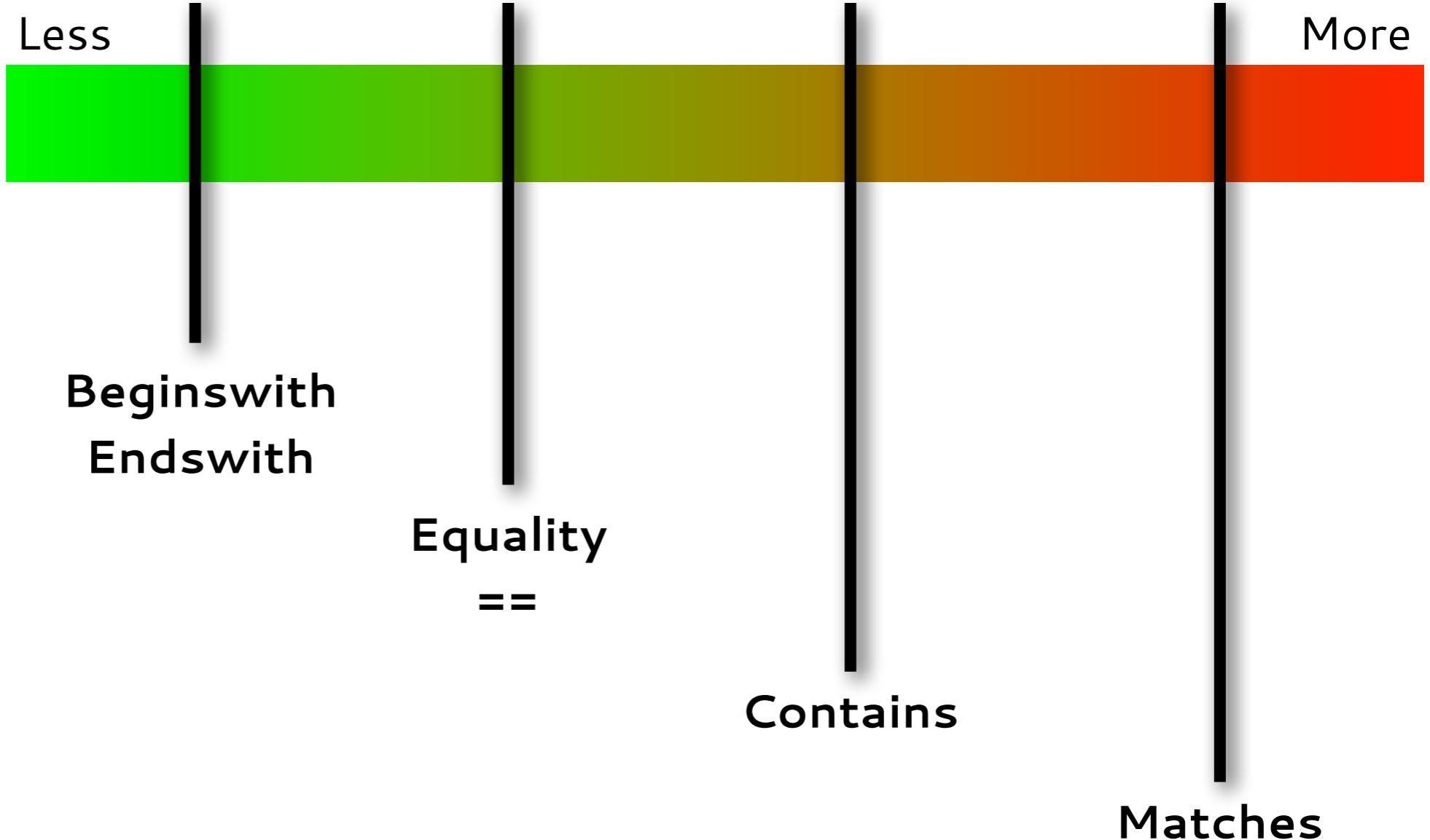
```
[fetchRequest setPredicate:  
    [NSPredicate predicateWithFormat:  
        @"shape == %@ AND duration > %i", @"sphere", 30]];
```

```
[fetchRequest setPredicate:  
    [NSPredicate predicateWithFormat:  
        @"shape == %@", @sphere, 30]]];
```

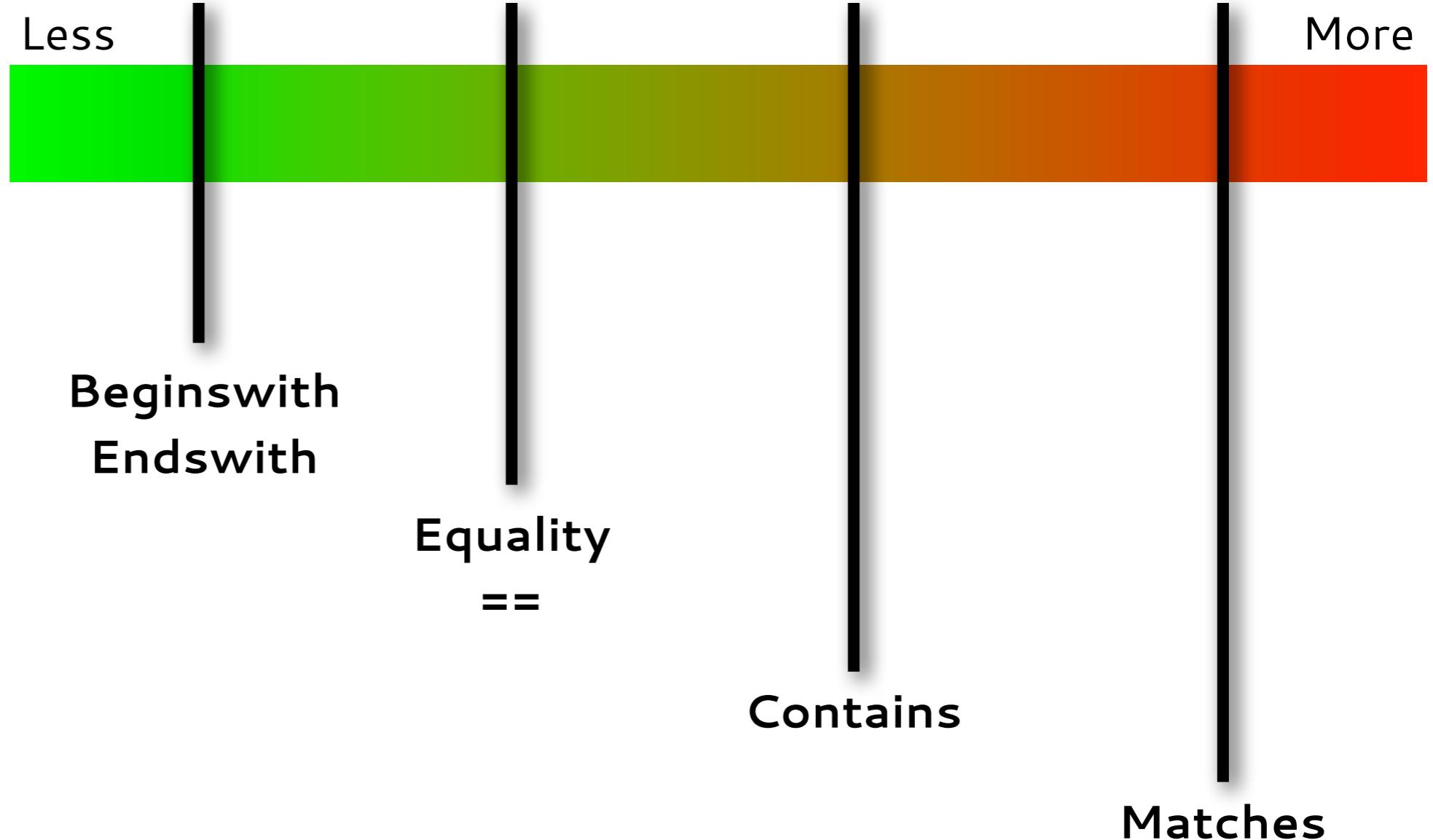
```
[fetchRequest setPredicate:  
    [NSPredicate predicateWithFormat:  
        @"duration > %i AND shape == %@", 30, @sphere]]];
```

Be Cautions with Strings

Predicate Costs



Predicate Costs



[cd] cost you even more

Predicates

- Do light (numerical) comparisons first
- Beginswith/Endswith instead of Contains/Matches
- Don't use [cd]

Searching

Canonicalize String Properties

Canonicalize String Properties

Description	canonicalizedDesc
Green Mën	green men
BEAM	beam
Próbë	probe
ABDUCTION	abduction

```
- (void)setDesc:(NSString *)desc {
    if (_desc != desc) {
        _desc = desc;
        [self updateCanonicalizedDesc:desc];
    }
}

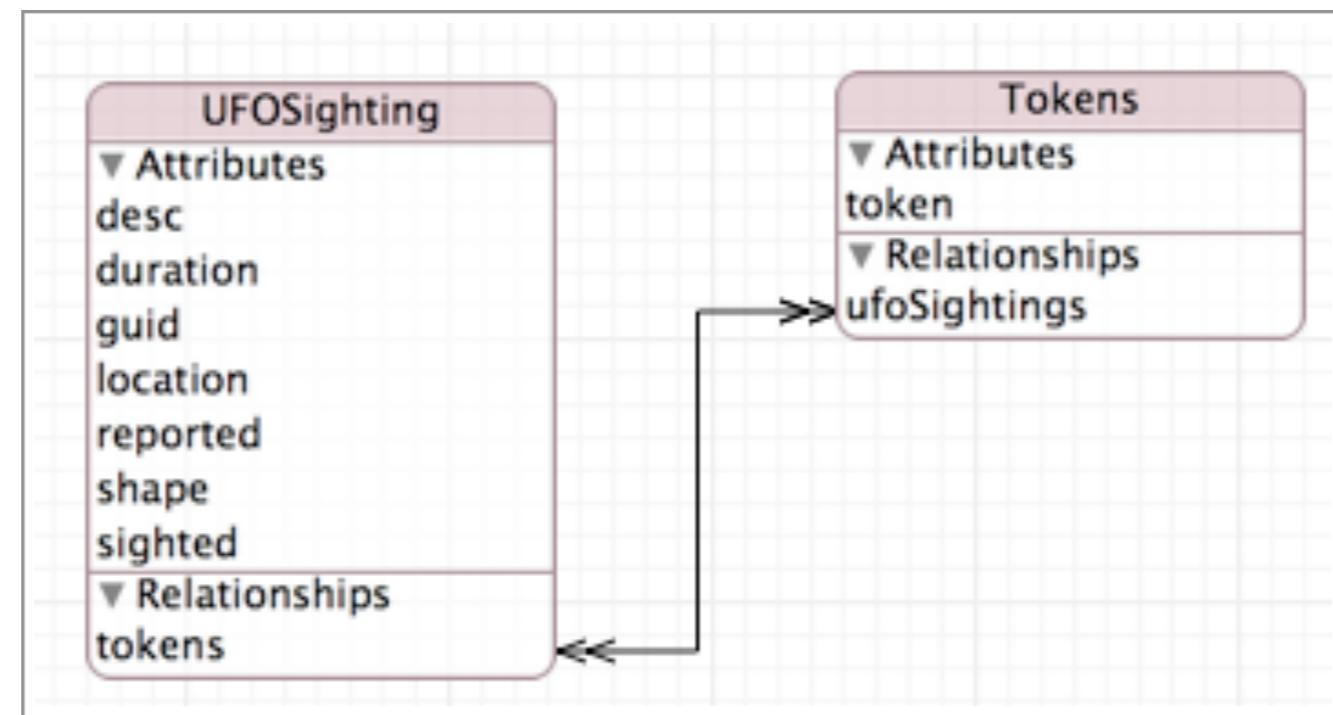
- (void)updateCanonicalizedDesc:(NSString *)desc {
    NSMutableString *result = [NSMutableString stringWithString:desc];
    CFStringNormalize((CFMutableStringRef)result,
                      kCFStringNormalizationFormD);
    CFStringFold((CFMutableStringRef)result,
                 kCFCompareCaseInsensitive |
                 kCFCompareDiacriticInsensitive |
                 kCFCompareWidthInsensitive, NULL);

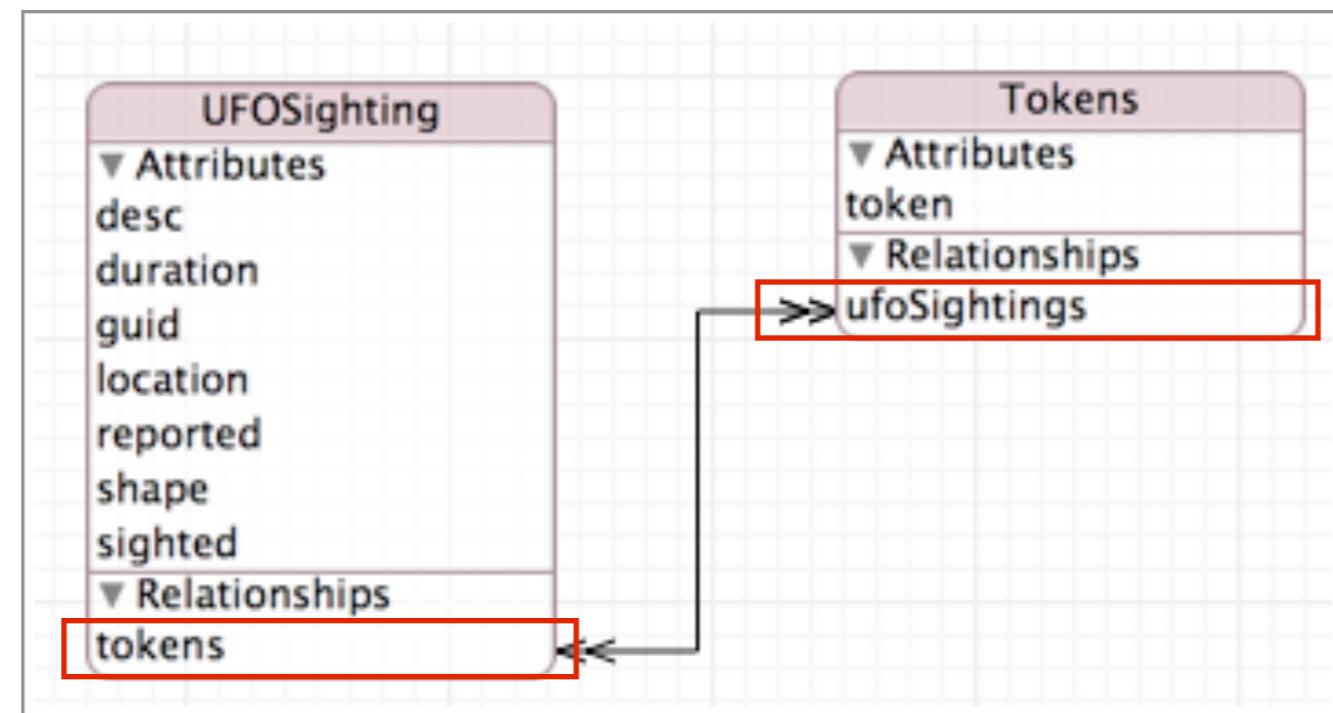
    self.canonicalizedDesc = [NSString stringWithString:result];
}
```

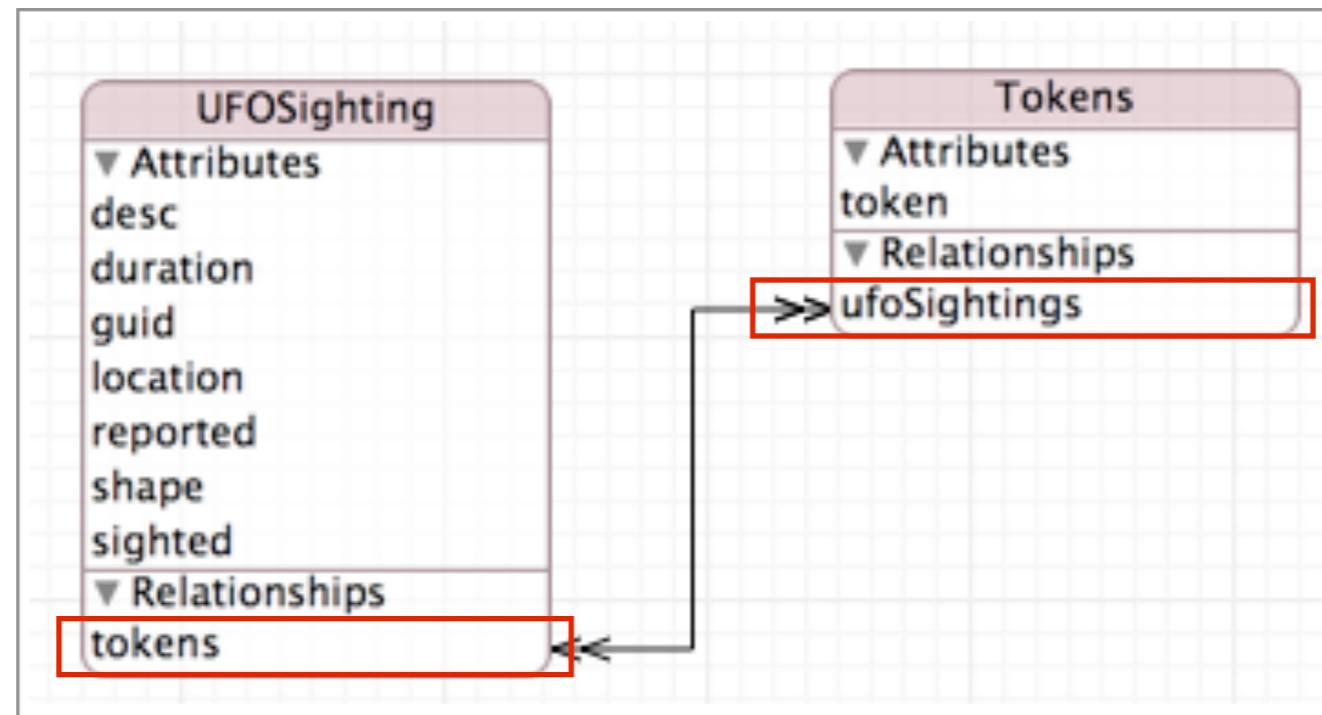
```
- (void)setDesc:(NSString *)desc {
    if (_desc != desc) {
        _desc = desc;
        [self updateCanonicalizedDesc:desc];
    }
}

- (void)updateCanonicalizedDesc:(NSString *)desc {
    NSMutableString *result = [NSMutableString stringWithString:desc];
    CFStringNormalize((CFMutableStringRef)result,
                      kCFStringNormalizationFormD);
    CFStringFold((CFMutableStringRef)result,
                 kCFCompareCaseInsensitive |
                 kCFCompareDiacriticInsensitive |
                 kCFCompareWidthInsensitive, NULL);
    self.canonicalizedDesc = [NSString stringWithString:result];
}
```

Canonicalized Tokens

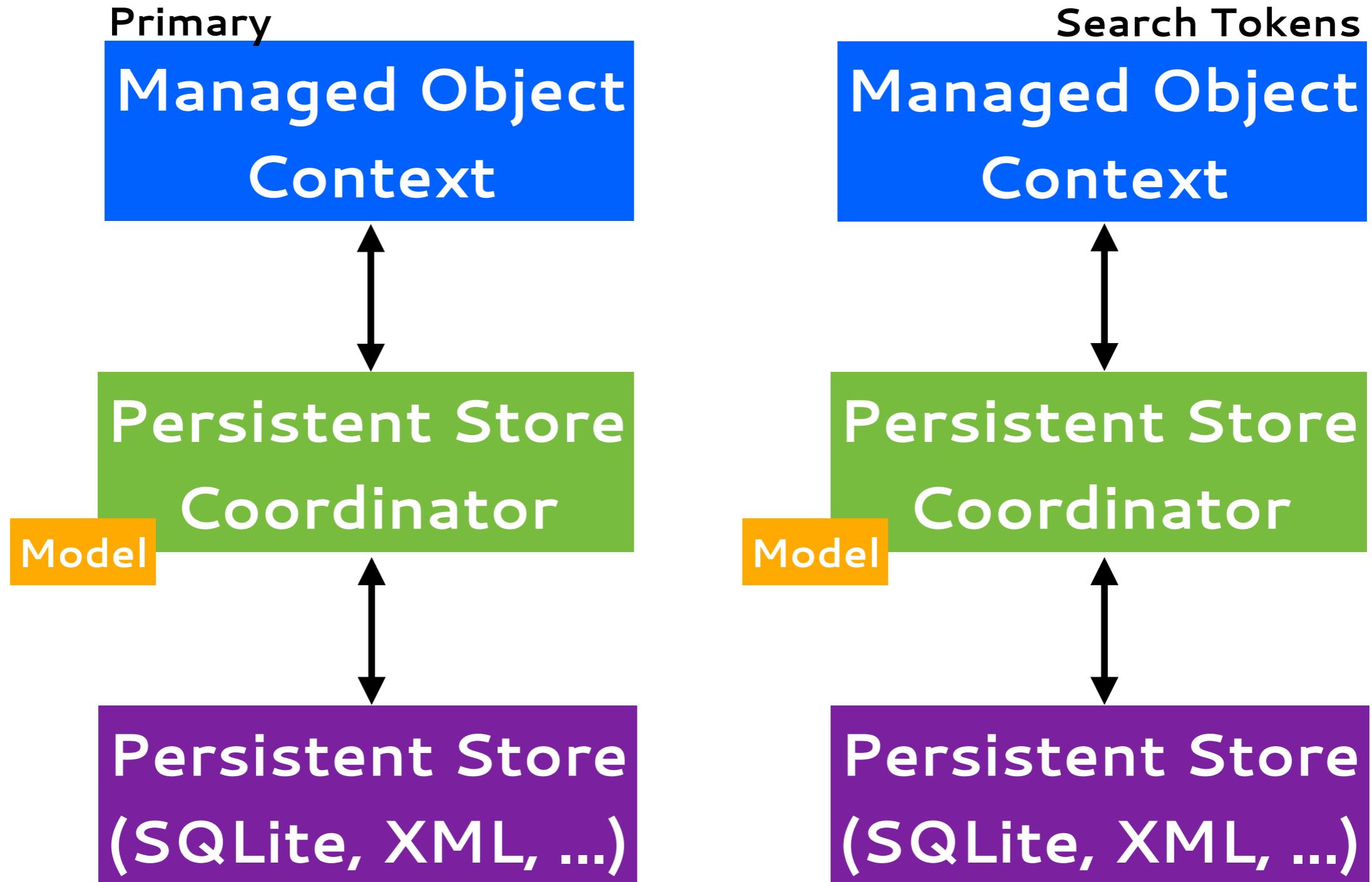


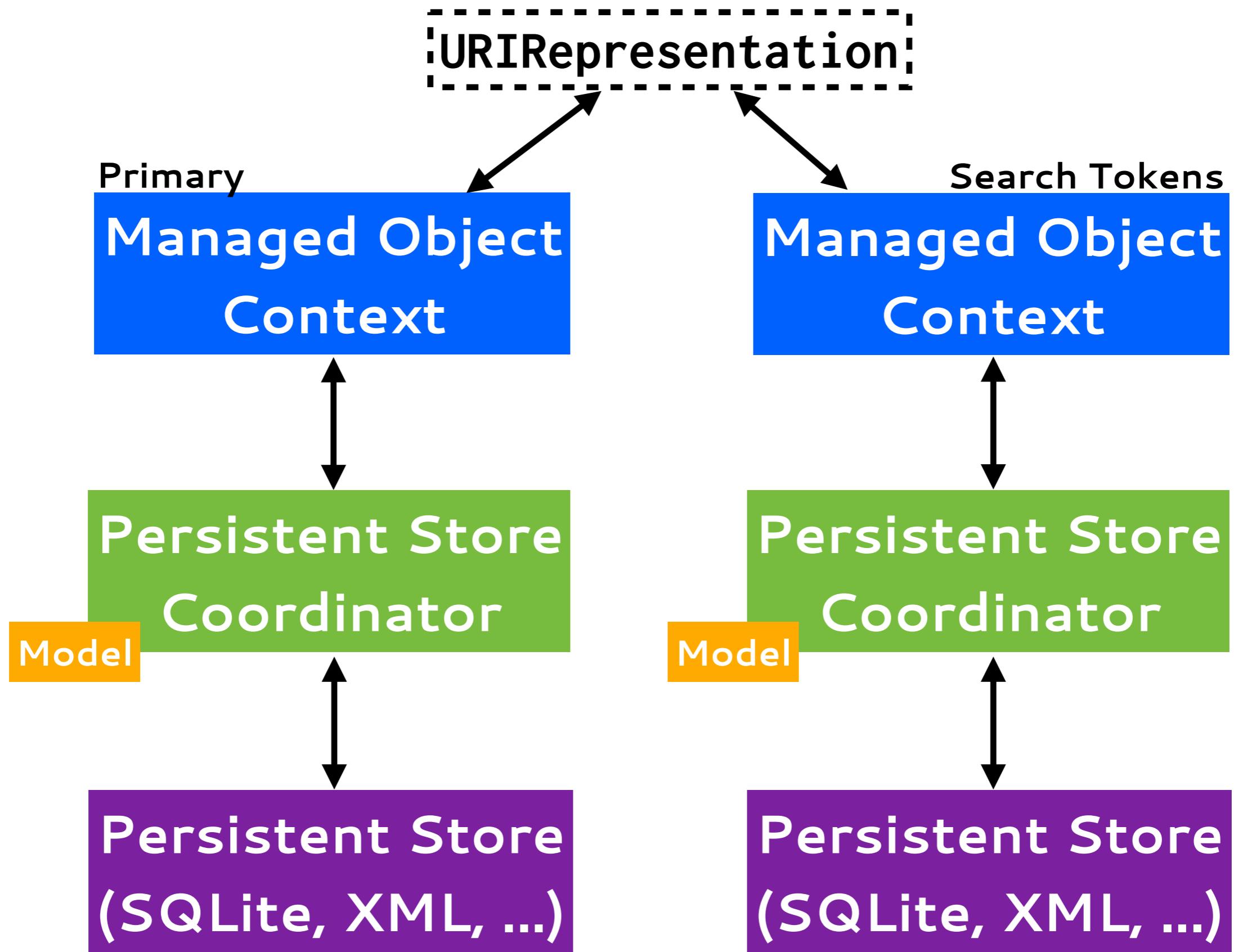




componentsSeparatedByCharactersInSet

Separate Stack





Hash of Tokens

Searching

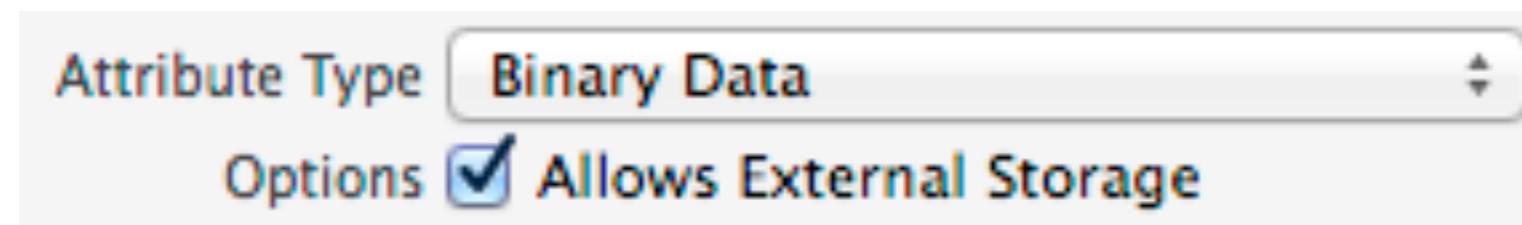
- Canonicalize text properties
- Use tokens
- Separate persistence stack
- Hash of tokens

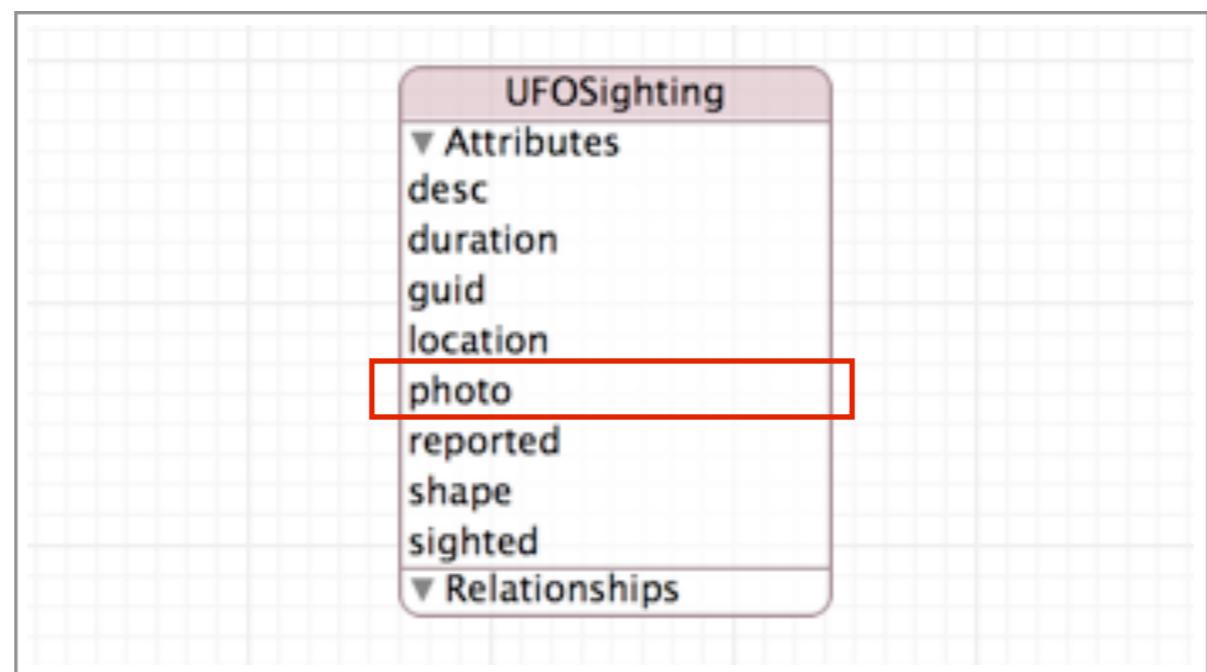
Data Model

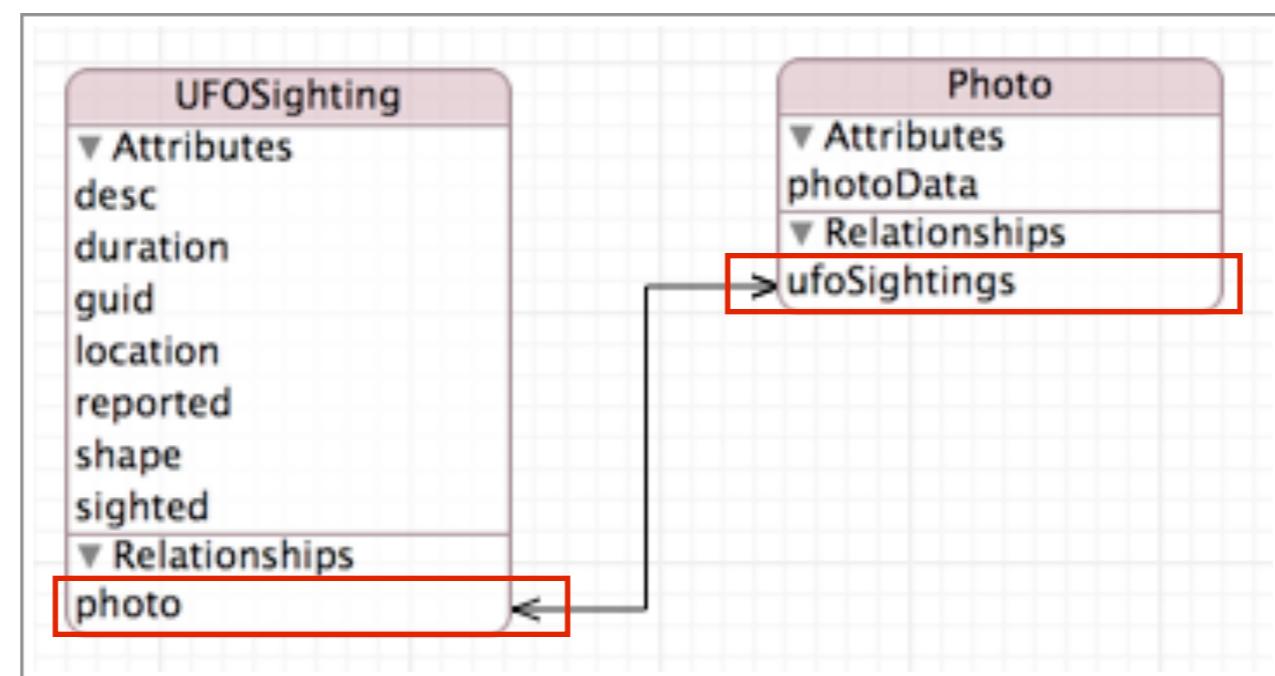
Don't Overnormalize

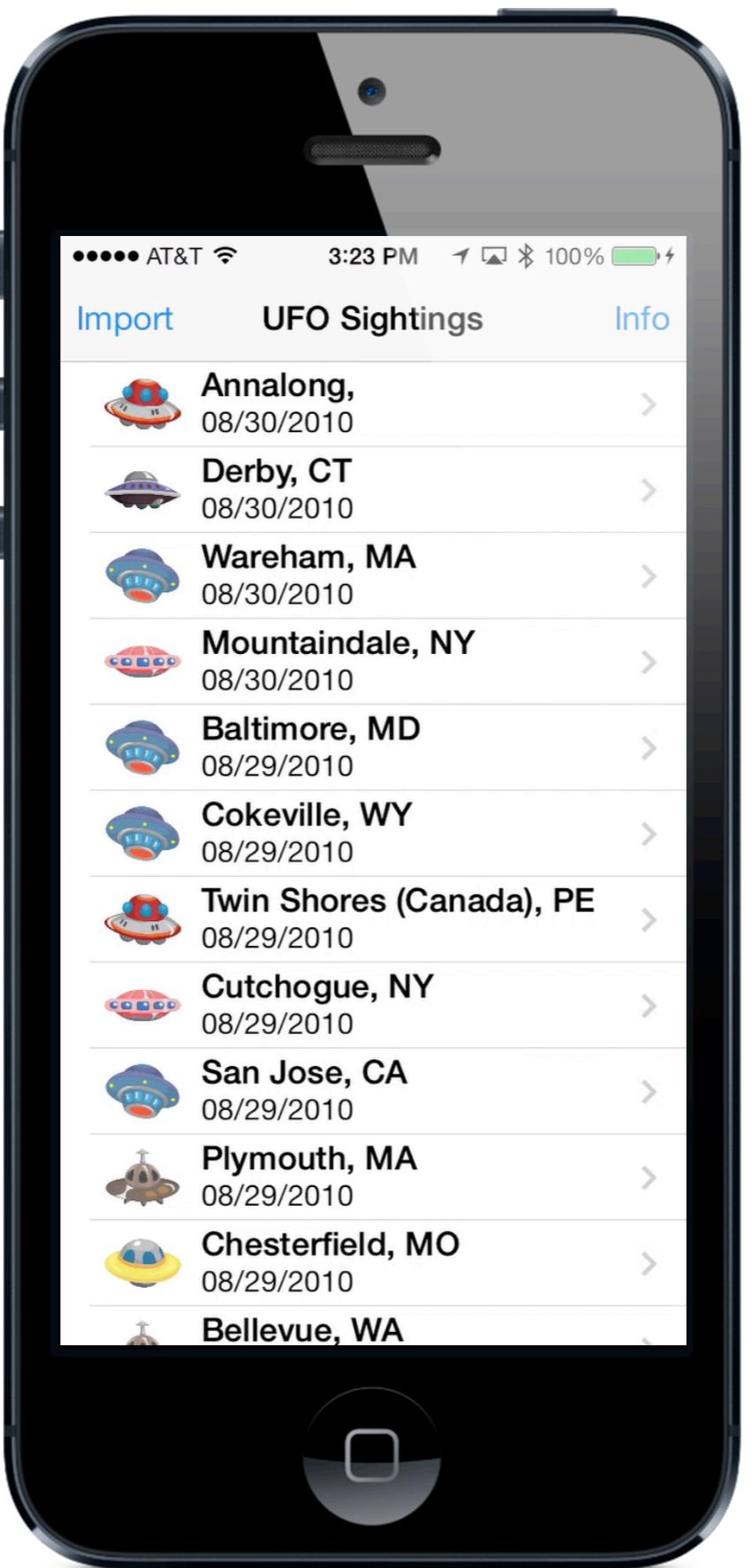
- Duplication isn't always bad**
- Let UI drive data model**

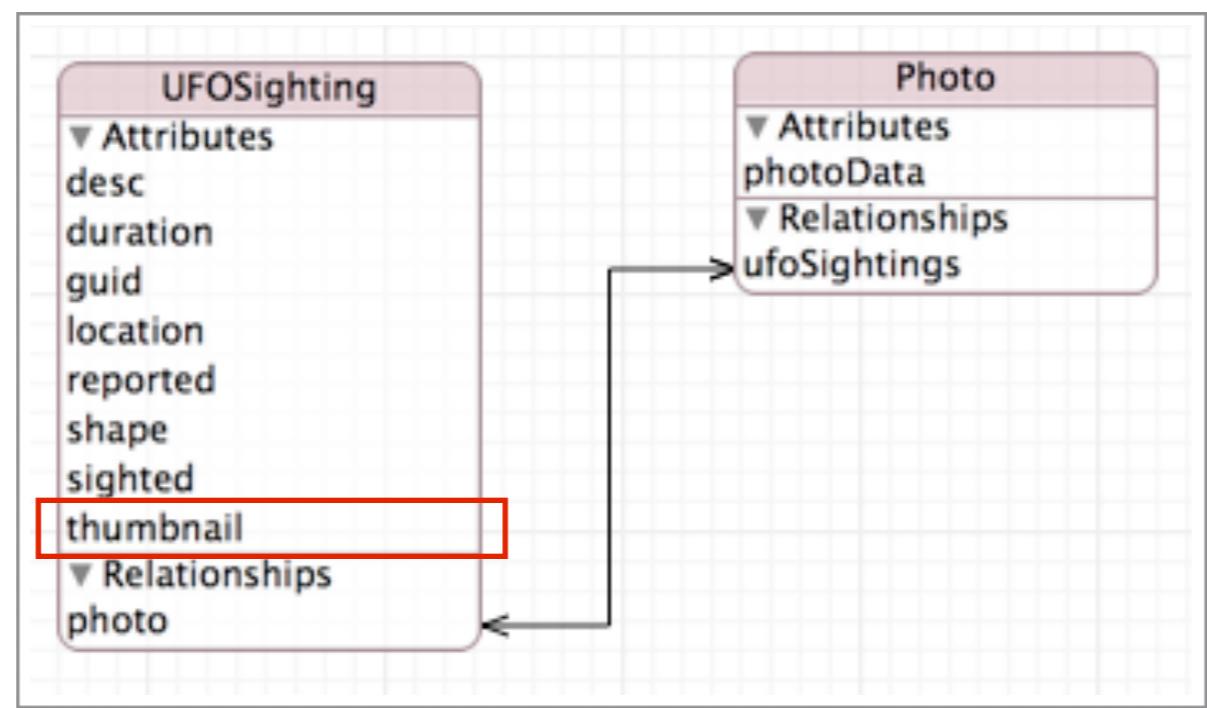
Use External Storage











Data Model

- Don't overnormalize
- Use external storage for large attributes
- Large blobs as separate entity
- Less data takes less time to fetch

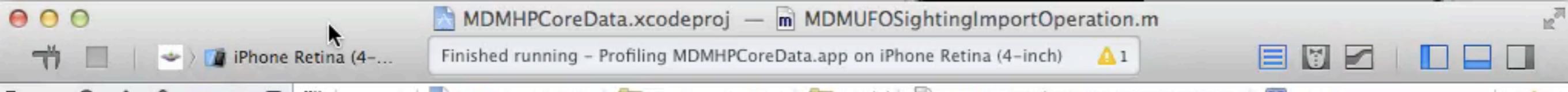
App Launch

```
dispatch_queue_t queue =
    dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
dispatch_async(queue, ^{
    // Create persistent store coordinator with model
    // Add persistent store to store coordinator
    // Create managed object context
    // Context is fully initialized, notify view controllers
    dispatch_sync(dispatch_get_main_queue(), ^{
        [self persistentStackInitialized];
    });
});
```

“Lots of Things” Problem

```
- (void)applicationDidEnterBackground:(UIApplication *)application {
    UIApplication *app = [UIApplication sharedApplication];
    __block UIBackgroundTaskIdentifier backgroundTask = 0;
    backgroundTask = [app beginBackgroundTaskWithExpirationHandler:^{
        [app endBackgroundTask:backgroundTask];
        backgroundTask = UIBackgroundTaskInvalid;
    }];
    dispatch_async(
        dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
        // Delete or update a bunch of NSManagedObjects
        // If success
        [app endBackgroundTask:backgroundTask];
        backgroundTask = UIBackgroundTaskInvalid;
        // If failure
        [app endBackgroundTask:backgroundTask];
        backgroundTask = UIBackgroundTaskInvalid;
    });
}
```

Importing Data



```
52
53 - (void)main {
54
55 // 1) WORST: Main queue context with naive find-or-create alg
56 //           Will block UI and take a long time
57
58
59     self.managedObjectContext = self.persistenceStack.managedObject
60     self.managedObjectContext.undoManager = nil;
61     [self.managedObjectContext performBlockAndWait:^{
62         [self import];
63     }];
64
65 // 2) BAD: Private queue context with naive find-or-create alg
66 //           Will not block UI, but will take a long time
67
68
69 //     self.managedObjectContext = [self.persistenceStack newF
70 //     self.managedObjectContext.undoManager = nil;
71 //     [self.managedObjectContext performBlockAndWait:^{
```

/Users/xzolian/xcode/projects/MDMHPCoreData/2/MDMHPCoreData/MDMHPCoreData/model/

No Selection

2013-11-10 16:25:39.012 MDMHPCoreData[15098:1303] Progress: 0.968546
2013-11-10 16:25:39.046 MDMHPCoreData[15098:1303] Progress: 0.978531
2013-11-10 16:25:39.085 MDMHPCoreData[15098:1303] Progress: 0.988516
2013-11-10 16:25:39.123 MDMHPCoreData[15098:1303] Progress: 0.998501
2013-11-10 16:25:39.129 MDMHPCoreData[15098:1303] Progress: 1.000000

All Output

Main Queue vs Private Queue

```
self.managedObjectContext = self.persistenceStack.managedObjectContext;
self.managedObjectContext.undoManager = nil;
[self.managedObjectContext performBlockAndWait:^{
    [self import];
}];
```

Main Queue vs Private Queue

```
self.managedObjectContext = self.persistenceStack.managedObjectContext;
self.managedObjectContext.undoManager = nil;
[self.managedObjectContext performBlockAndWait:^{
    [self import];
}];
```

```
NSManagedObjectContext *privateManagedObjectContext =
    [[NSManagedObjectContext alloc]
        initWithConcurrencyType:NSPrivateQueueConcurrencyType];
[privateManagedObjectContext setPersistentStoreCoordinator:
    self.managedObjectContext.persistentStoreCoordinator];
privateManagedObjectContext.undoManager = nil;

[privateManagedObjectContext performBlockAndWait:^{
    [self import];
}];
```

```

58  //
59  //    self.managedObjectContext = self.persistenceStack.managedObjectContext;
60  //    self.managedObjectContext.undoManager = nil;
61  //    [self.managedObjectContext performBlockAndWait:^{
62  //        [self import];
63  //    }];
64
65
66 // 2) BAD: Private queue context with naive find-or-create algorithm
67 //        Will not block UI, but will take a long time
68 //
69 self.managedObjectContext = [self.persistenceStack newPrivateQueueContext];
70 self.managedObjectContext.undoManager = nil;
71 [self.managedObjectContext performBlockAndWait:^{
72     [self import];
73 }];
74
75
76 // 3) GOOD: Private queue context with batch importing algorithm
77 //        Will not block UI, but will have high memory usage

```

/Users/xzolian/xcode/projects/MDMHPCoreData/2/MDMHPCoreData/MDMHPCoreData/model/

No Selection

	2013-11-10 16:25:39.012 MDMHPCoreData[15098:1303] Progress: 0.968546
	2013-11-10 16:25:39.046 MDMHPCoreData[15098:1303] Progress: 0.978531
	2013-11-10 16:25:39.085 MDMHPCoreData[15098:1303] Progress: 0.988516
	2013-11-10 16:25:39.123 MDMHPCoreData[15098:1303] Progress: 0.998501
	2013-11-10 16:25:39.129 MDMHPCoreData[15098:1303] Progress: 1.000000

All Output

Typical Import Algorithm

- 1) JSON to NSDictionary**
- 2) Find existing NSManagedObject**
- 3) Optionally, create new NSManagedObject**
- 4) Set attributes of new/updated NSManagedObject**

Typical Import Algorithm

- 1) JSON to NSDictionary**
- 2) Find existing NSManagedObject**
- 3) Optionally, create new NSManagedObject**
- 4) Set attributes of new/updated NSManagedObject**

Inefficient Find-Or-Create Algorithm

```
+ (instancetype)findOrCreateWithIdentifier:(id)identifier
    inContext:(NSManagedObjectContext *)context {
    NSFetchedResultsController *fetchedResultsController = [NSFetchedResultsController
        fetchedResultsControllerWithEntityName:@"UFOsighting"];
    fetchedResultsController.predicate = [NSPredicate predicateWithFormat:@"%K = %@", @"GUID", identifier];
    fetchedResultsController.fetchLimit = 1;
    id object = [[context executeFetchRequest:fetchedResultsController error:NULL]
        lastObject];
    if (object == nil) {
        object = [UFOsighting insertNewObjectIntoContext:context];
    }
    return object;
}
```

Inefficient Find-Or-Create Algorithm

```
+ (instancetype)findOrCreateWithIdentifier:(id)identifier
                                    inContext:(NSManagedObjectContext *)context {
    NSFetchedResultsController *fetchedResultsController = [NSFetchedResultsController
        fetchedResultsControllerWithEntityName:@"UFOsighting"];
    fetchedResultsController.predicate = [NSPredicate predicateWithFormat:@"%K = %@", @"GUID", identifier];
    fetchedResultsController.fetchLimit = 1;
    id object = [[context executeFetchRequest:fetchedResultsController error:&error] lastObject];
    if (object == nil) {
        object = [UFOsighting insertNewObjectIntoContext:context];
    }
    return object;
}
```

New Data

1

2

3

4

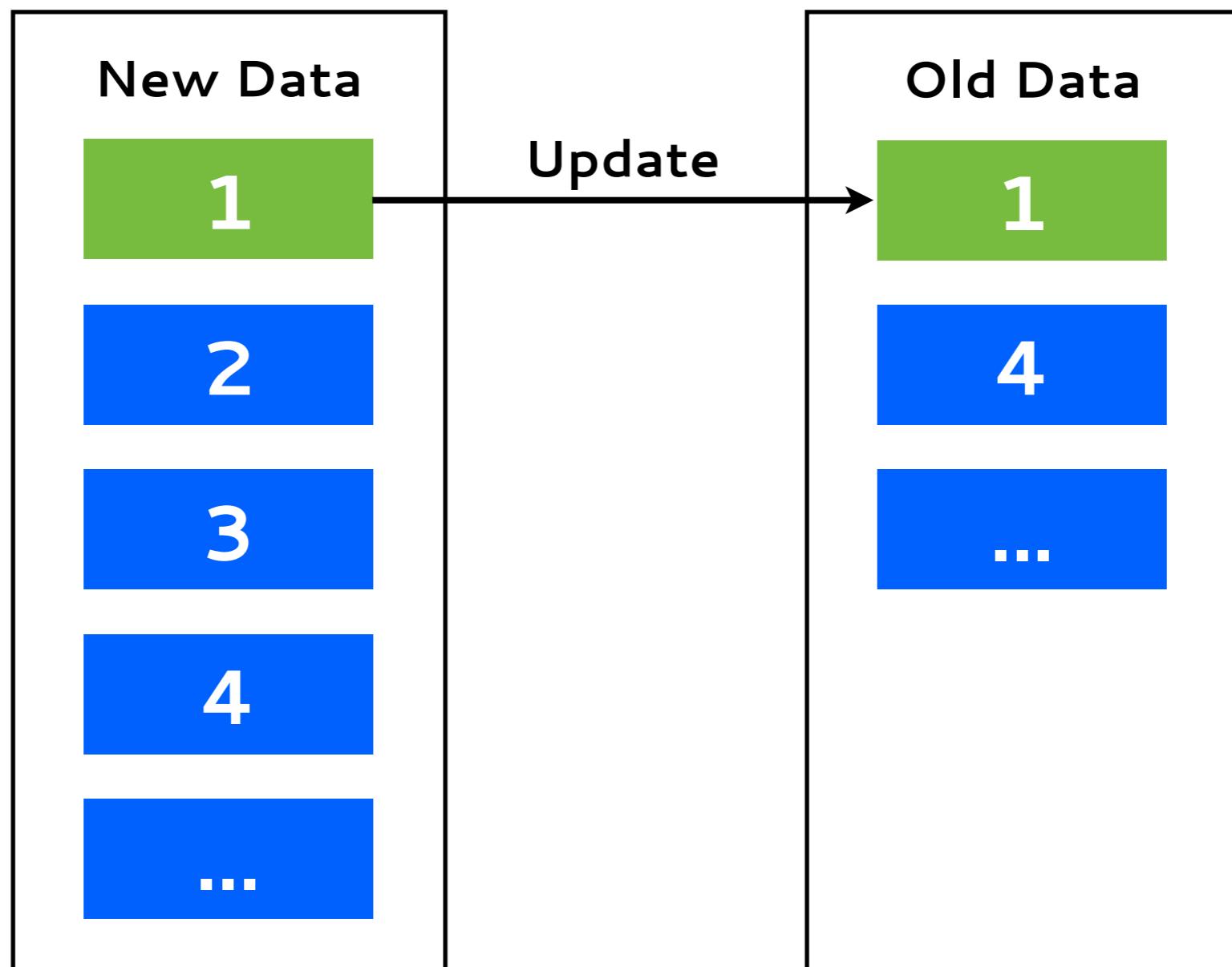
...

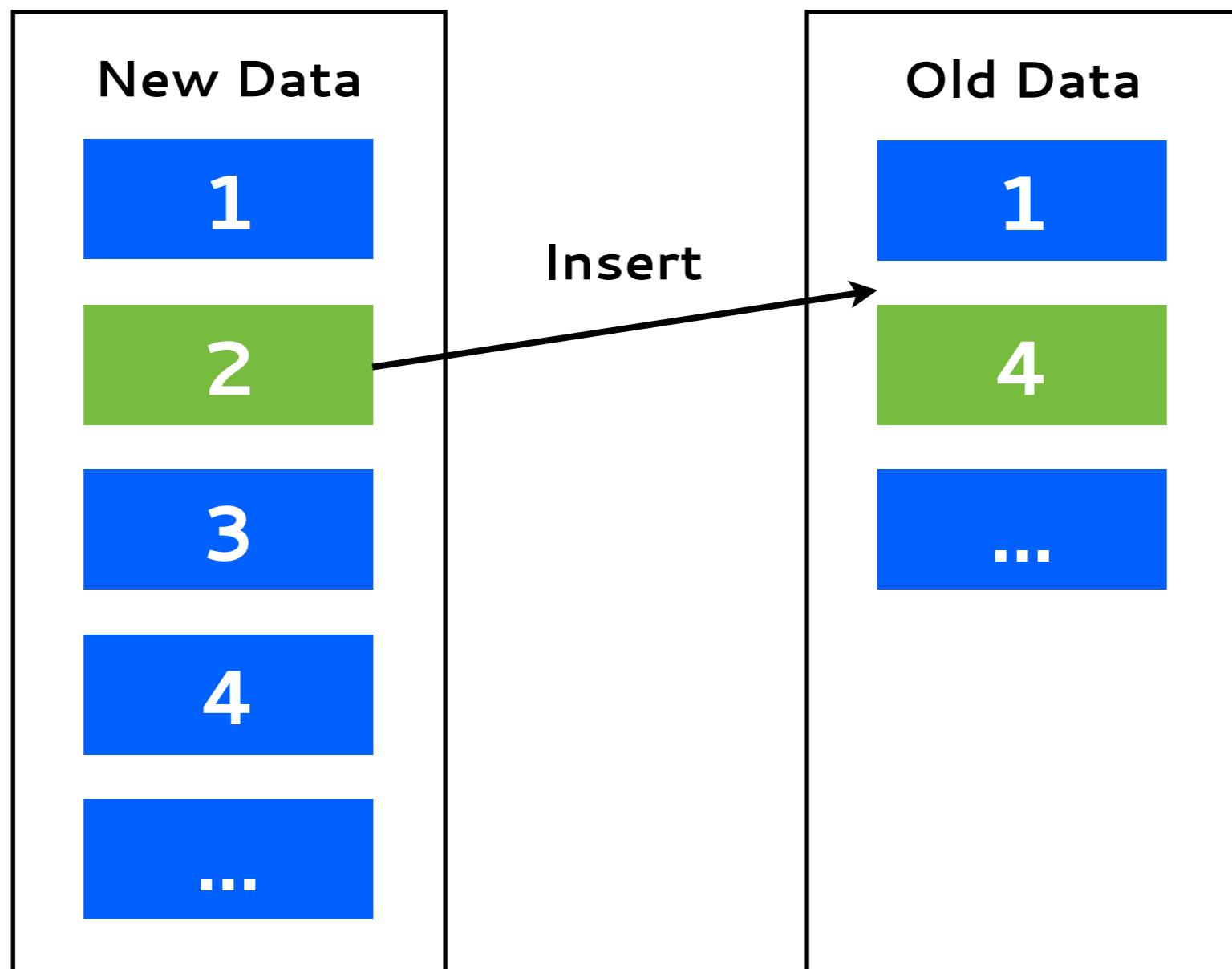
Old Data

1

4

...





New Data

1

2

3

4

...

Old Data

1

2

4

...

New Data

1

2

3

4

...

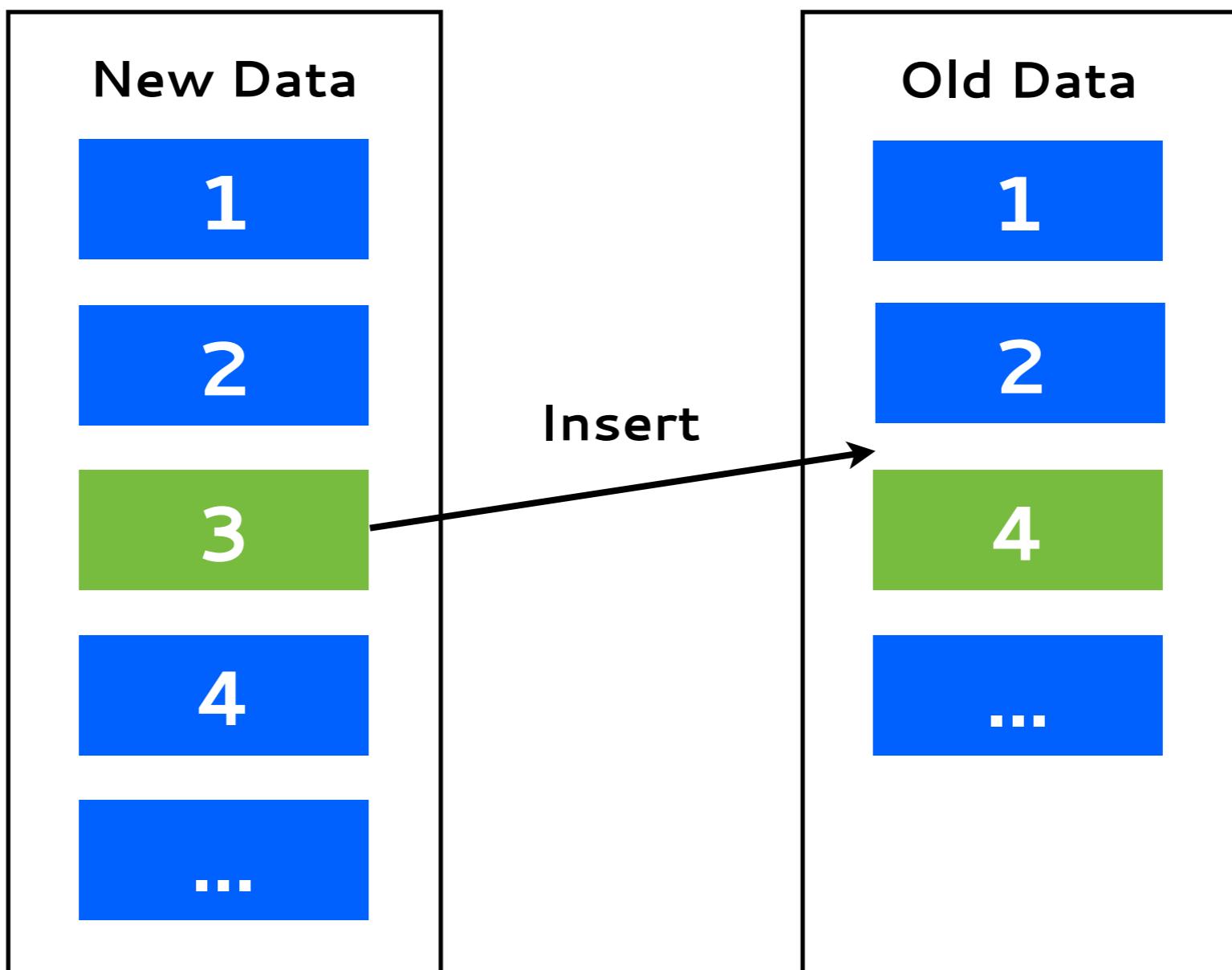
Old Data

1

2

4

...



New Data

1

2

3

4

...

Old Data

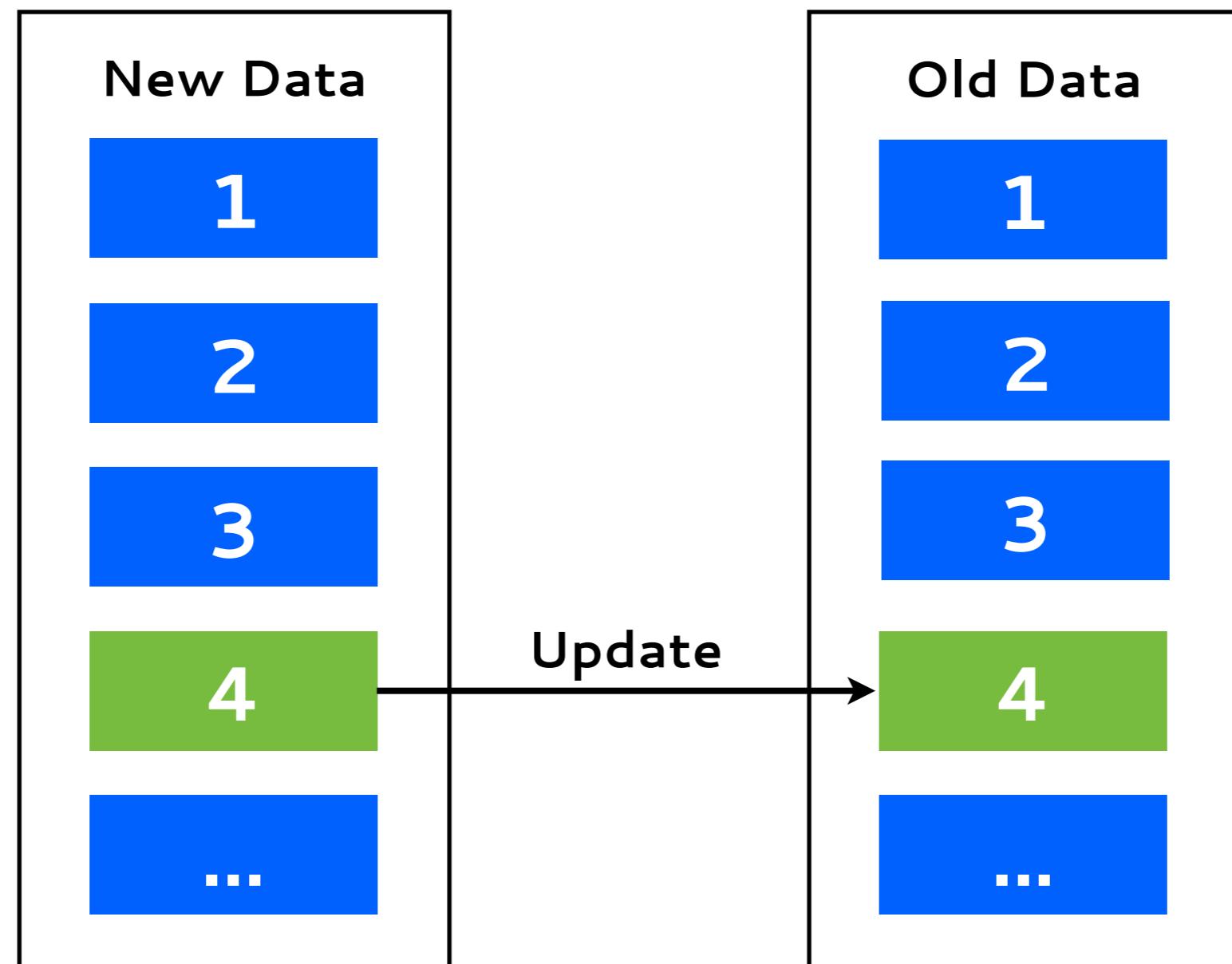
1

2

3

4

...



~~Inefficient~~ Find-Or-Create Algorithm

```
NSEnumerator *jsonEnumerator = [sortedArray objectEnumerator];
NSEnumerator *fetchResultsEnumerator = [fetchResults objectEnumerator];
NSDictionary *UFOSightingDictionary = [jsonEnumerator nextObject];
UFOSighting *UFOSightingManagedObject = [fetchResultsEnumerator nextObject];

while (UFOSightingDictionary) {

    // Check if managed object already exist

    // Not an update, create new managed object

    // Set new attributes

    if (isUpdate) {
        UFOSightingDictionary = [jsonEnumerator nextObject];
        UFOSightingManagedObject = [fetchResultsEnumerator nextObject];
    } else {
        UFOSightingDictionary = [jsonEnumerator nextObject];
    }
}
```

Inefficient Find-Or-Create Algorithm

```
NSEnumerator *jsonEnumerator = [sortedArray objectEnumerator];
NSEnumerator *fetchResultsEnumerator = [fetchResults objectEnumerator];
NSDictionary *UFOSightingDictionary = [jsonEnumerator nextObject];
UFOSighting *UFOSightingManagedObject = [fetchResultsEnumerator nextObject];

while (UFOSightingDictionary) {

    // Check if managed object already exist

    // Not an update, create new managed object

    // Set new attributes

    if (isUpdate) {
        UFOSightingDictionary = [jsonEnumerator nextObject];
        UFOSightingManagedObject = [fetchResultsEnumerator nextObject];
    } else {
        UFOSightingDictionary = [jsonEnumerator nextObject];
    }
}
```

Inefficient Find-Or-Create Algorithm

```
NSEnumerator *jsonEnumerator = [sortedArray objectEnumerator];
NSEnumerator *fetchResultsEnumerator = [fetchResults objectEnumerator];
NSDictionary *UFOSightingDictionary = [jsonEnumerator nextObject];
UFOSighting *UFOSightingManagedObject = [fetchResultsEnumerator nextObject];

while (UFOSightingDictionary) {

    // Check if managed object already exist

    // Not an update, create new managed object

    // Set new attributes

    if (isUpdate) {
        UFOSightingDictionary = [jsonEnumerator nextObject];
        UFOSightingManagedObject = [fetchResultsEnumerator nextObject];
    } else {
        UFOSightingDictionary = [jsonEnumerator nextObject];
    }
}
```

Inefficient Find-Or-Create Algorithm

```
NSEnumerator *jsonEnumerator = [sortedArray objectEnumerator];
NSEnumerator *fetchResultsEnumerator = [fetchResults objectEnumerator];
NSDictionary *UFOSightingDictionary = [jsonEnumerator nextObject];
UFOSighting *UFOSightingManagedObject = [fetchResultsEnumerator nextObject];

while (UFOSightingDictionary) {

    // Check if managed object already exist

    // Not an update, create new managed object

    // Set new attributes

    if (isUpdate) {
        UFOSightingDictionary = [jsonEnumerator nextObject];
        UFOSightingManagedObject = [fetchResultsEnumerator nextObject];
    } else {
        UFOSightingDictionary = [jsonEnumerator nextObject];
    }
}
```

```
66 // 2) BAD: Private queue context with naive find-or-create algorithm
67 // Will not block UI, but will take a long time
68 //
69 // self.managedObjectContext = [self.persistenceStack newPrivateQueueContext];
70 // self.managedObjectContext.undoManager = nil;
71 // [self.managedObjectContext performBlockAndWait:^{
72 //     [self import];
73 // }];
74 //
75 //
76 // 3) GOOD: Private queue context with batch importing algorithm
77 // Will not block UI, but will have high memory usage
78 //
79 self.managedObjectContext = [self.persistenceStack newPrivateQueueContext];
80 self.managedObjectContext.undoManager = nil;
81 [self.managedObjectContext performBlockAndWait:^{
82     [self batchImport];
83 }];
84 //
85 //
```

/Users/xzolian/xcode/projects/MDMHPCoreData/2/MDMHPCoreData/MDMHPCoreData/model/

No Selection

	2013-11-10 16:25:39.012 MDMHPCoreData[15098:1303] Progress: 0.968546 2013-11-10 16:25:39.046 MDMHPCoreData[15098:1303] Progress: 0.978531 2013-11-10 16:25:39.085 MDMHPCoreData[15098:1303] Progress: 0.988516 2013-11-10 16:25:39.123 MDMHPCoreData[15098:1303] Progress: 0.998501 2013-11-10 16:25:39.129 MDMHPCoreData[15098:1303] Progress: 1.000000
--	--

All Output

```
NSUInteger totalUFOSightings = [UFOSightingsArray count];
NSInteger totalBatches = totalUFOSightings / MDM_BATCH_SIZE_IMPORT;

// Create array with just the unique keys
NSArray *jsonGUIDArray = [sortedArray valueForKey:UFO_KEY_JSON_GUID];

for (NSInteger batchCounter = 0; batchCounter <= totalBatches; batchCounter++) {

    // Create batch range based on batch size
    NSRange range = NSMakeRange(batchCounter*MDM_BATCH_SIZE_IMPORT, MDM_BATCH_SIZE_IMPORT);
    NSArray *jsonBatchGUIDArray = [jsonGUIDArray subarrayWithRange:range];

    // Grab sorted persisted managed objects, based on the batch
    NSFetchedResultsController *fetchRequest = [[NSFetchedResultsController alloc] initWithEntityName:@"UFOSighting"];
    NSPredicate *fetchPredicate = [NSPredicate predicateWithFormat:@"%@ IN %@", @"GUID", jsonBatchGUIDArray];
    // ...

    while (UFOSightingDictionary) {
        // ...
        // Efficient find-or-create algorithm
        // ...
    }
}
```

```
NSUInteger totalUFOSightings = [UFOSightingsArray count];
NSInteger totalBatches = totalUFOSightings / MDM_BATCH_SIZE_IMPORT;

// Create array with just the unique keys
NSArray *jsonGUIDArray = [sortedArray valueForKey:UFO_KEY_JSON_GUID];

for (NSInteger batchCounter = 0; batchCounter <= totalBatches; batchCounter++) {

    // Create batch range based on batch size
    NSRange range = NSMakeRange(batchCounter*MDM_BATCH_SIZE_IMPORT, MDM_BATCH_SIZE_IMPORT);
    NSArray *jsonBatchGUIDArray = [jsonGUIDArray subarrayWithRange:range];

    // Grab sorted persisted managed objects, based on the batch
    NSFetchedResultsController *fetchRequest = [[NSFetchedResultsController alloc] initWithEntityName:@"UFOSighting"];
    NSPredicate *fetchPredicate = [NSPredicate predicateWithFormat:@"%@ IN %@", @"GUID", jsonBatchGUIDArray];
    // ...

    while (UFOSightingDictionary) {
        // ...
        // Efficient find-or-create algorithm
        // ...
    }
}
```

```
NSUInteger totalUFOSightings = [UFOSightingsArray count];
NSInteger totalBatches = totalUFOSightings / MDM_BATCH_SIZE_IMPORT;

// Create array with just the unique keys
NSArray *jsonGUIDArray = [sortedArray valueForKey:UFO_KEY_JSON_GUID];

for (NSInteger batchCounter = 0; batchCounter <= totalBatches; batchCounter++) {

    // Create batch range based on batch size
    NSMakeRange range = NSMakeRange(batchCounter*MDM_BATCH_SIZE_IMPORT, MDM_BATCH_SIZE_IMPORT);
    NSArray *jsonBatchGUIDArray = [jsonGUIDArray subarrayWithRange:range];

    // Grab sorted persisted managed objects, based on the batch
    NSFetchedResultsController *fetchedResultsController = [[NSFetchedResultsController alloc] initWithEntityName:@"UFOSighting"];
    NSPredicate *fetchPredicate = [NSPredicate predicateWithFormat:@"%@ IN %@", @"GUID", jsonBatchGUIDArray];
    // ...

    while (UFOSightingDictionary) {
        // ...
        // Efficient find-or-create algorithm
        // ...
    }
}
```

Pre-Populated SQLite File in App Bundle

**Download Pre-Populated
SQLite File**

Importing Data

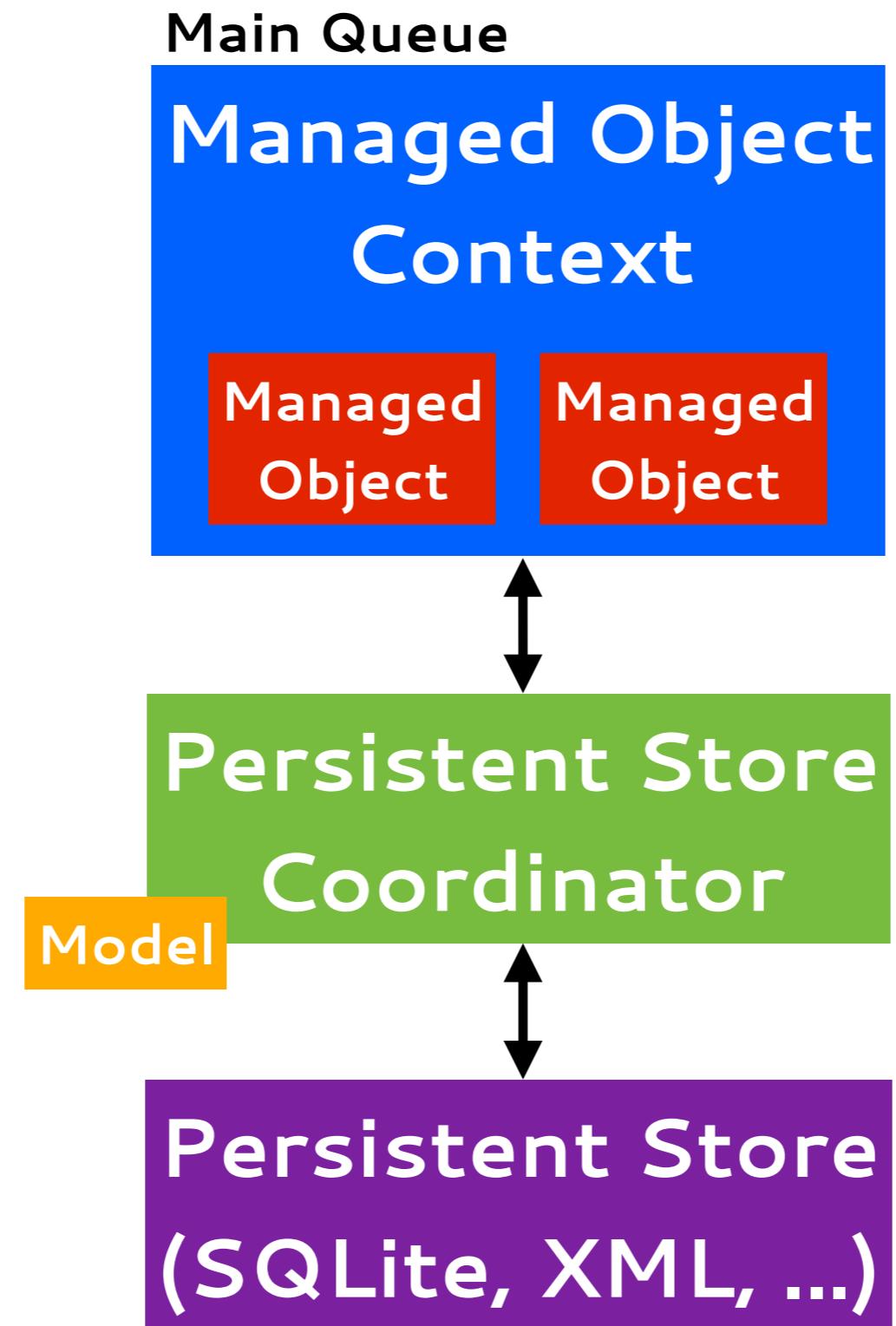
- Do import work on private queues
- Use efficient find-or-create algorithm
- Work in batches to keep memory low
- Use pre-populated SQLite file

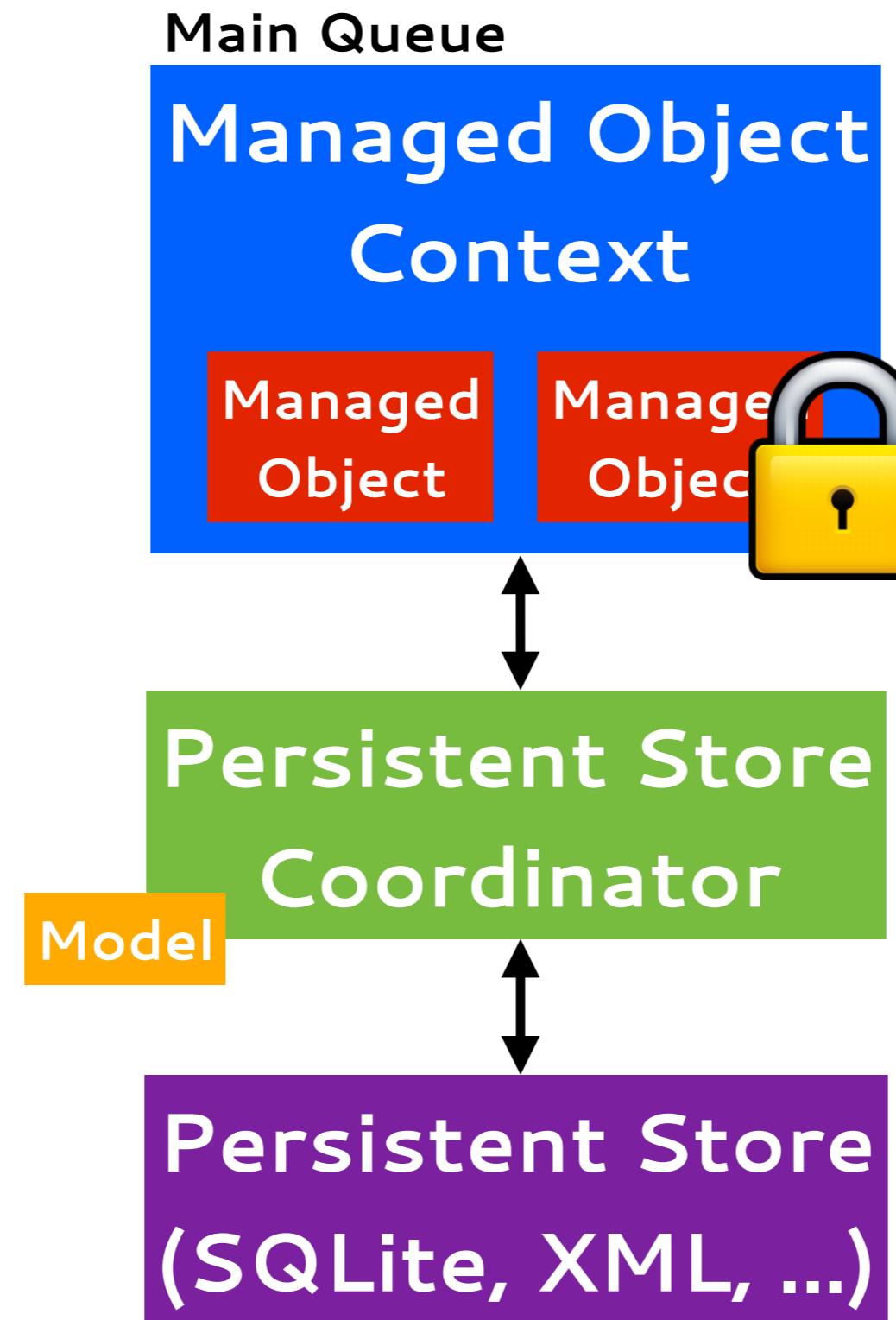
Memory

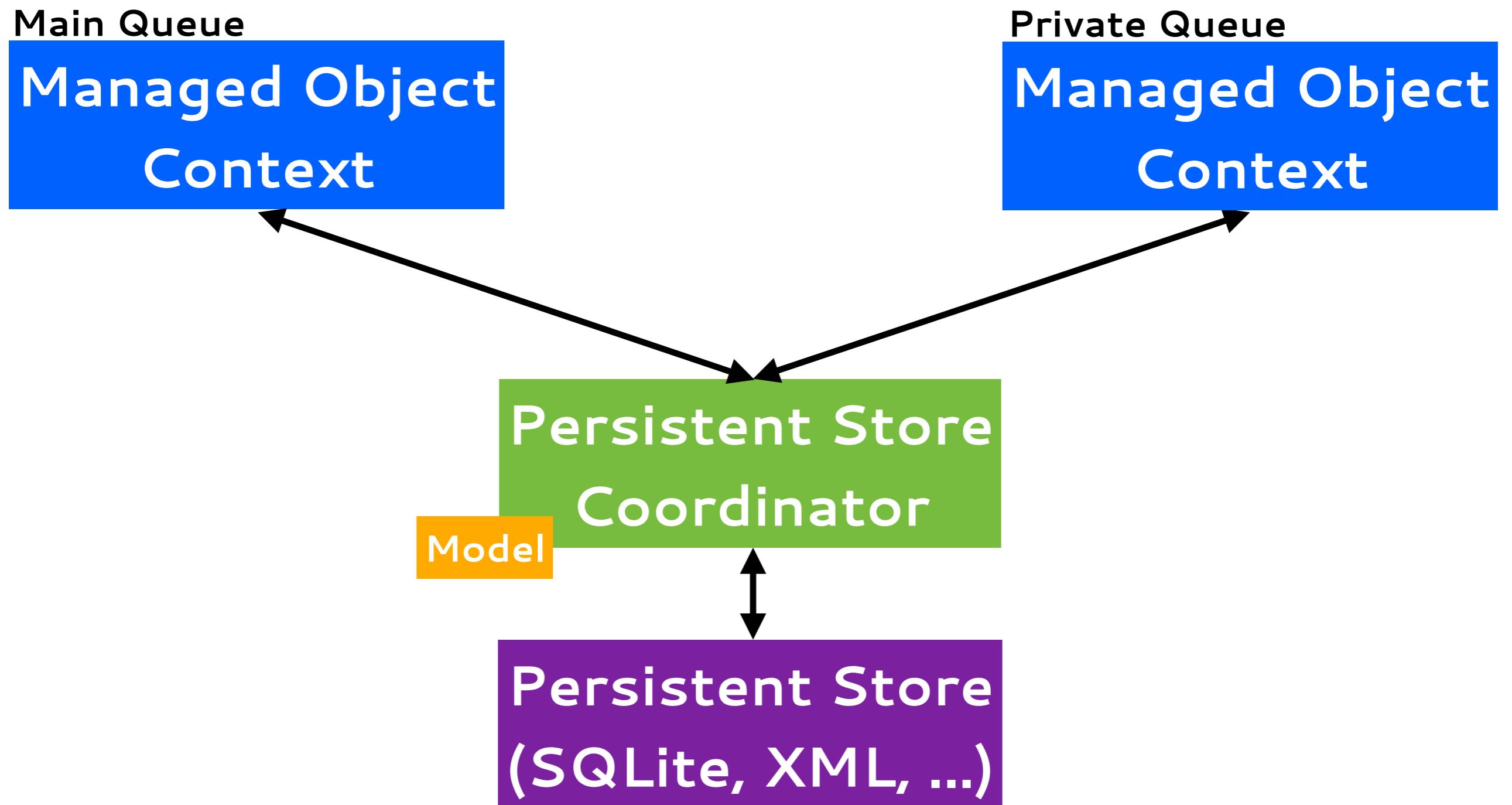
```
[context refreshObject:UFOsightingManagedObject mergeChanges:NO];
```

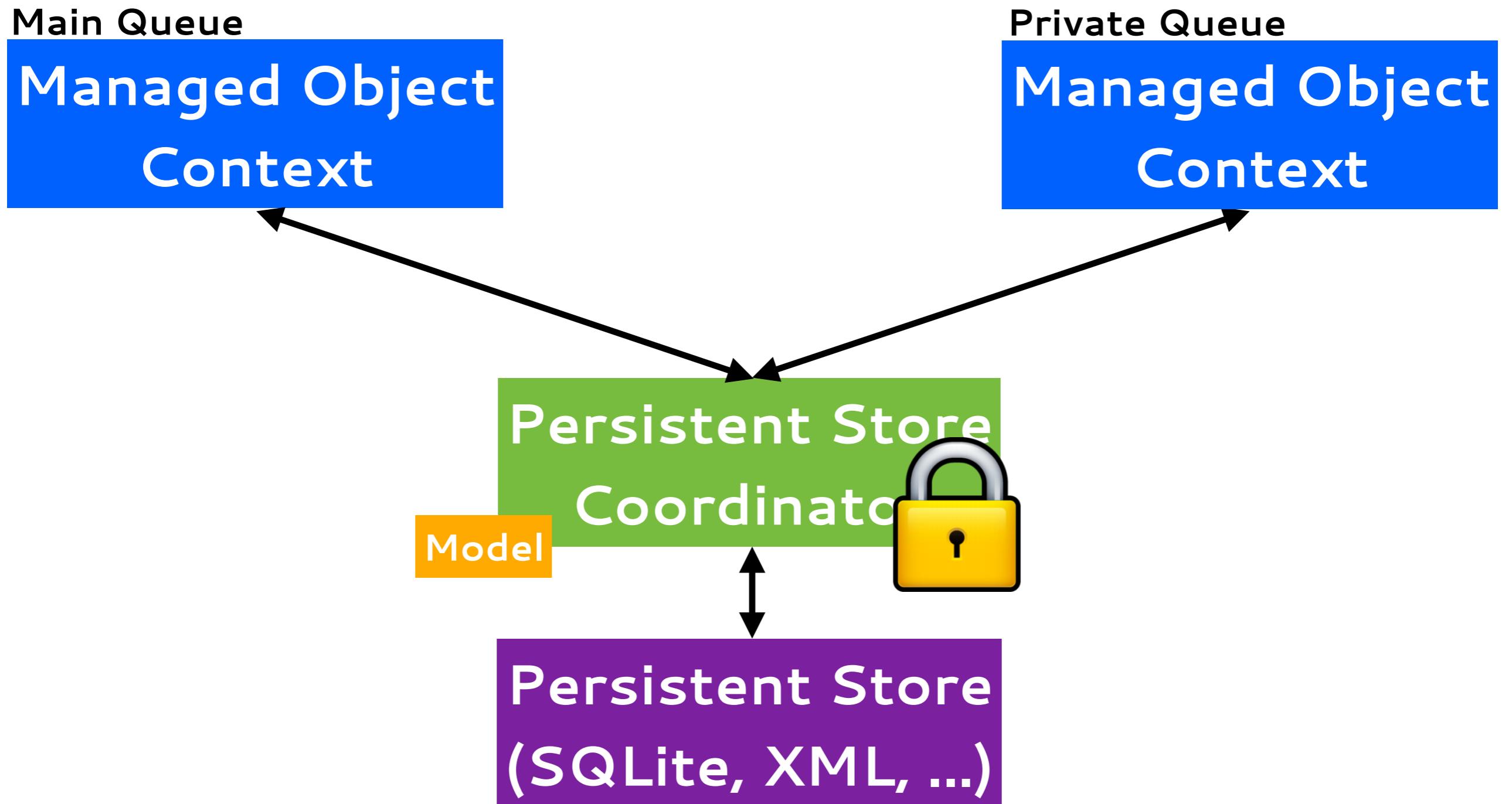
```
[context reset];
```

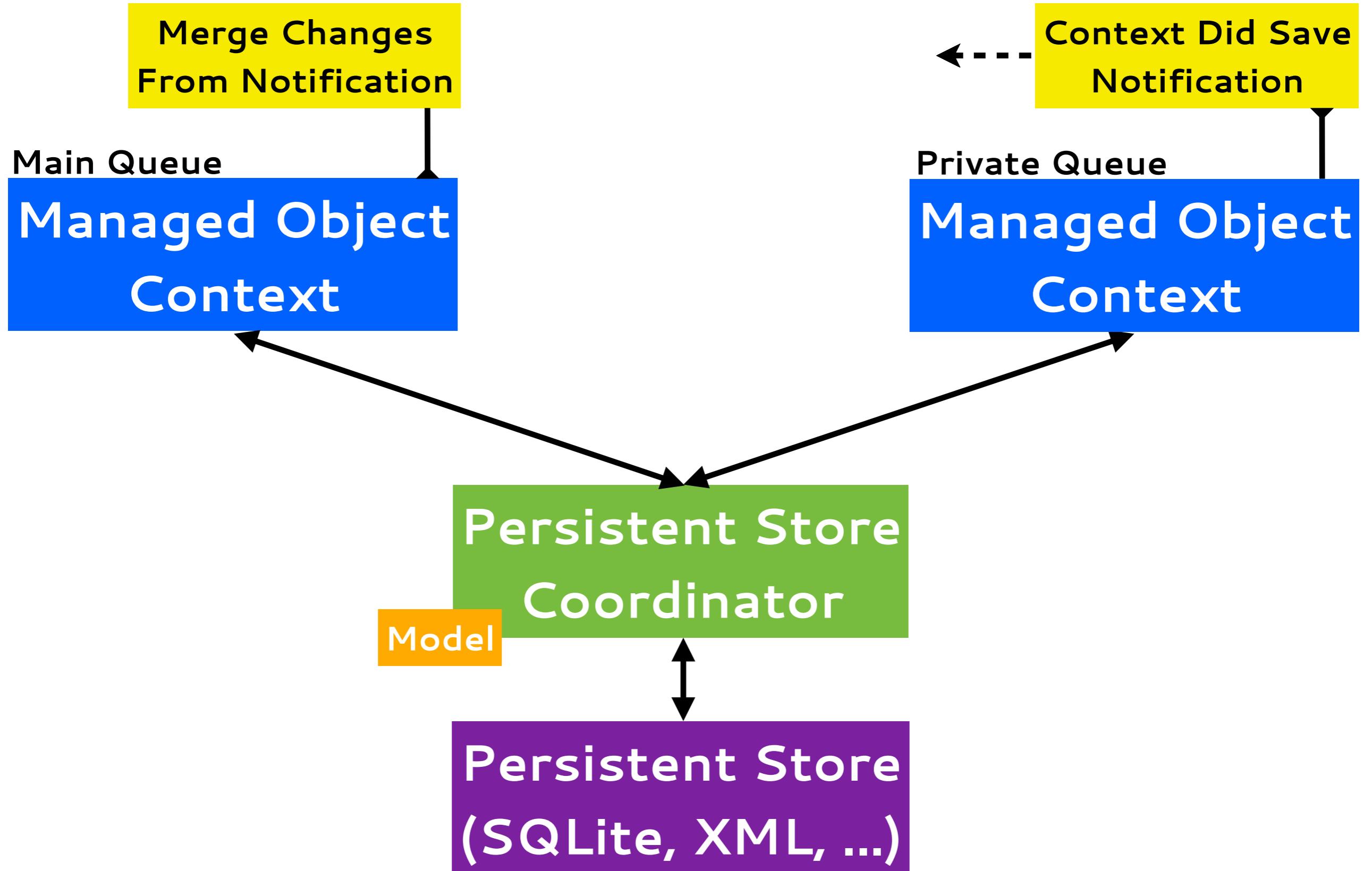
Concurrency Models

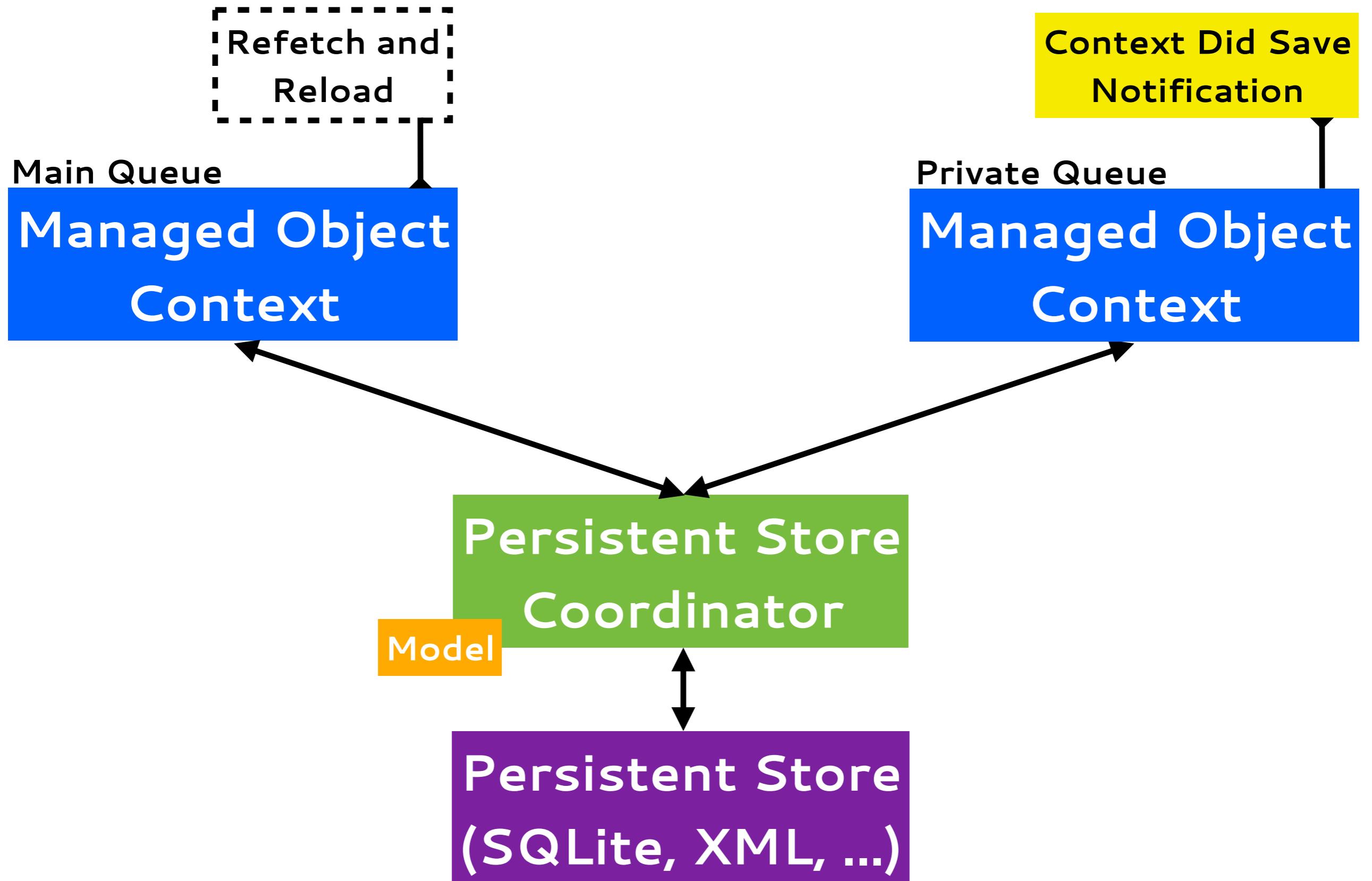


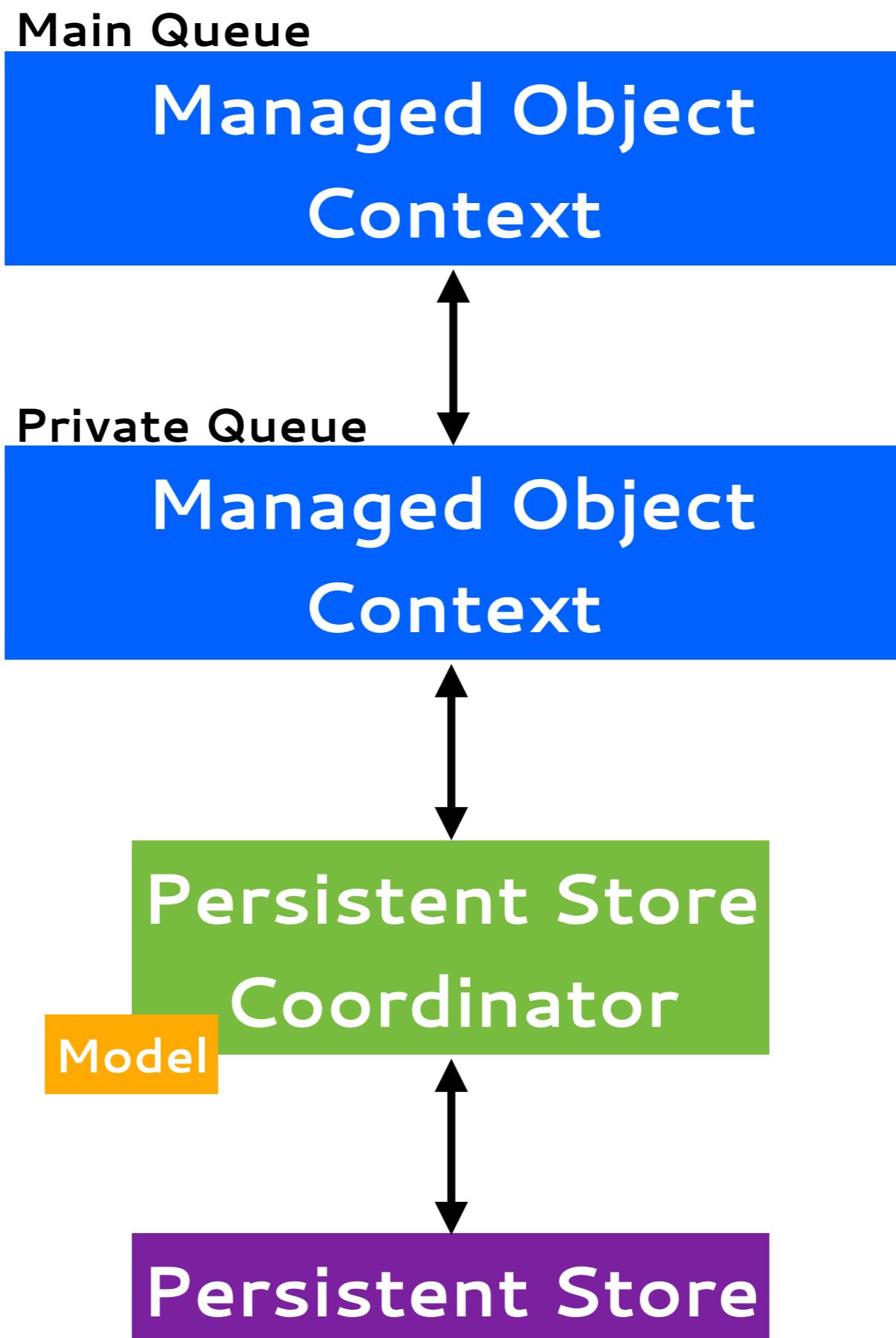


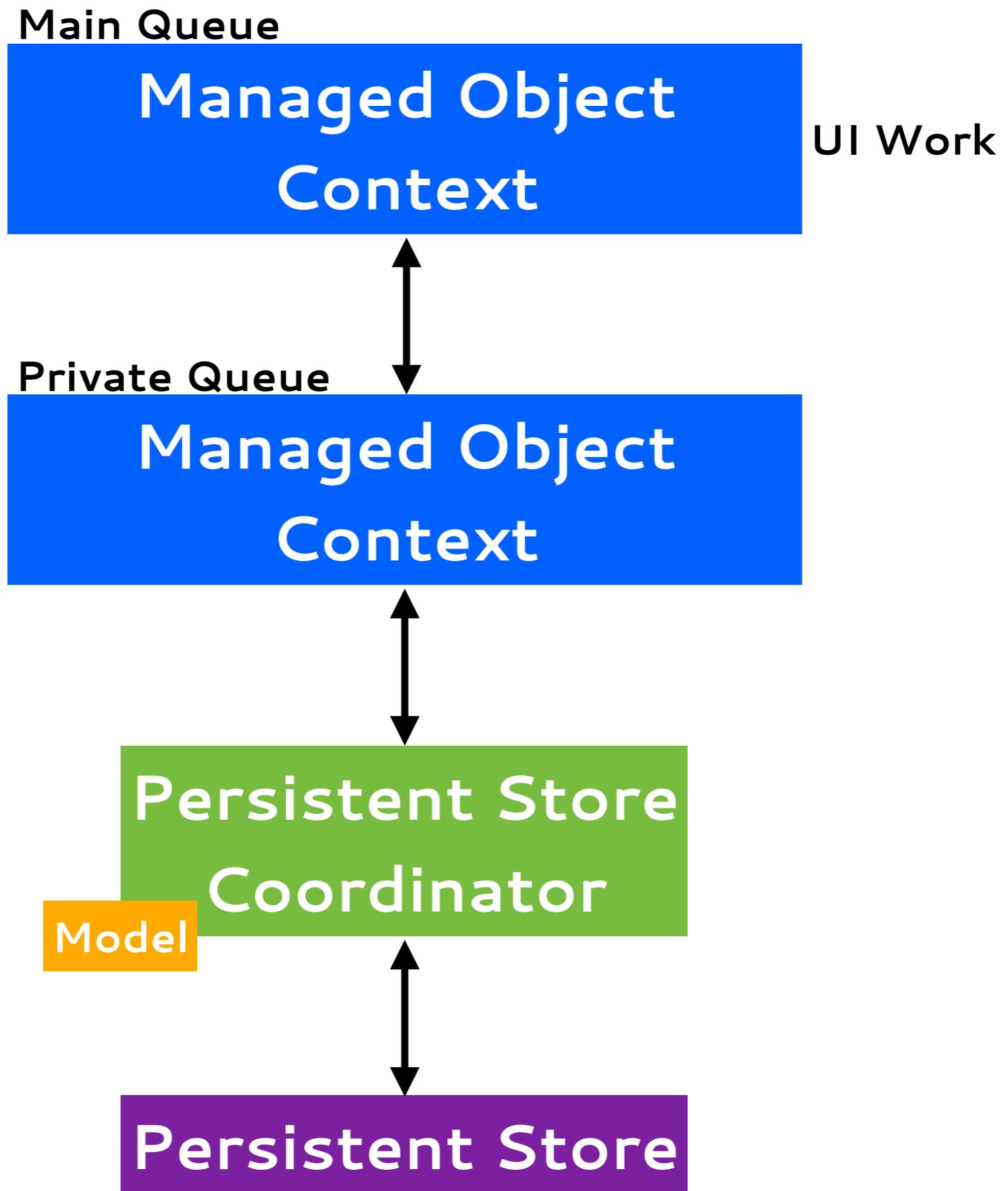


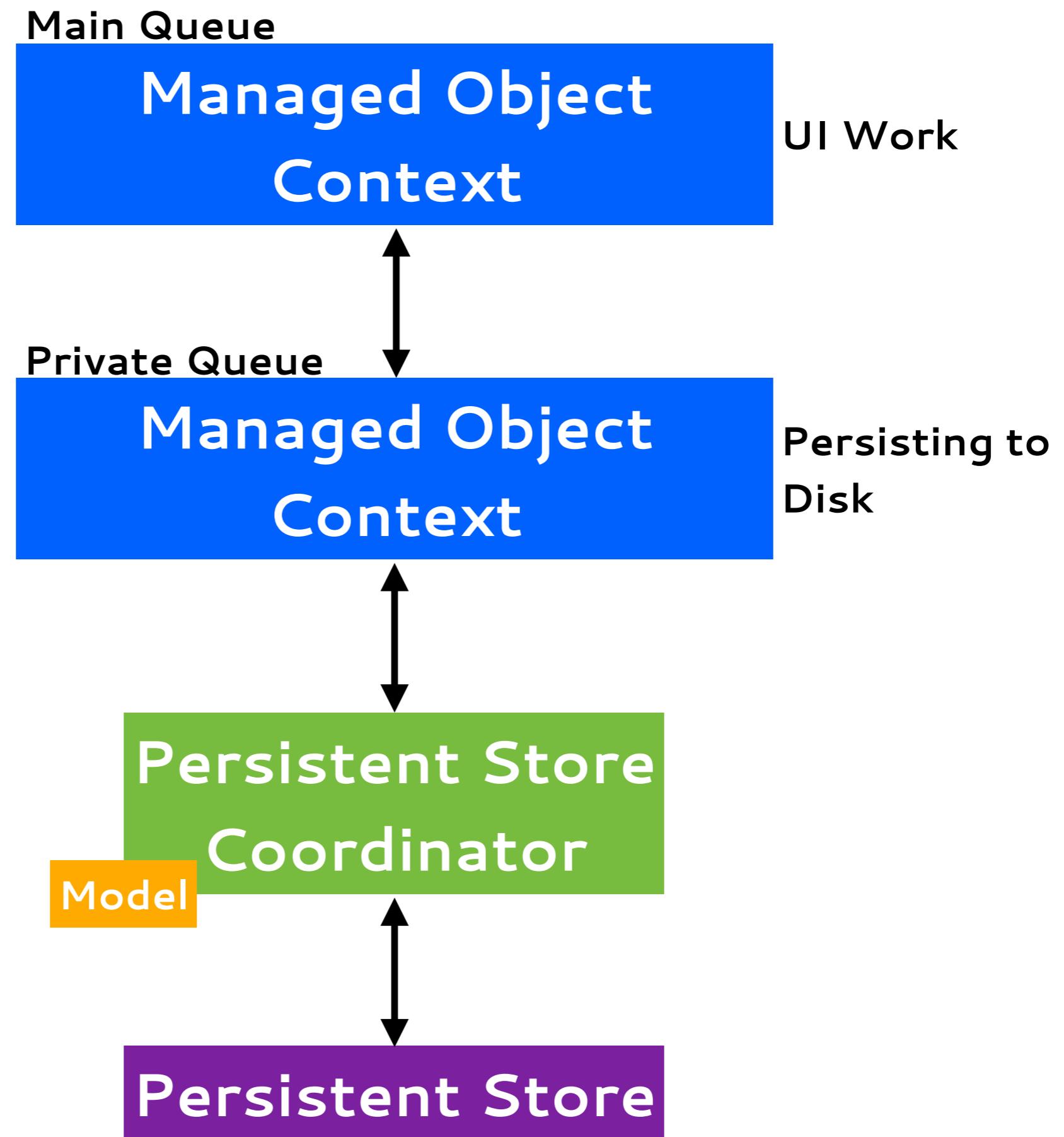


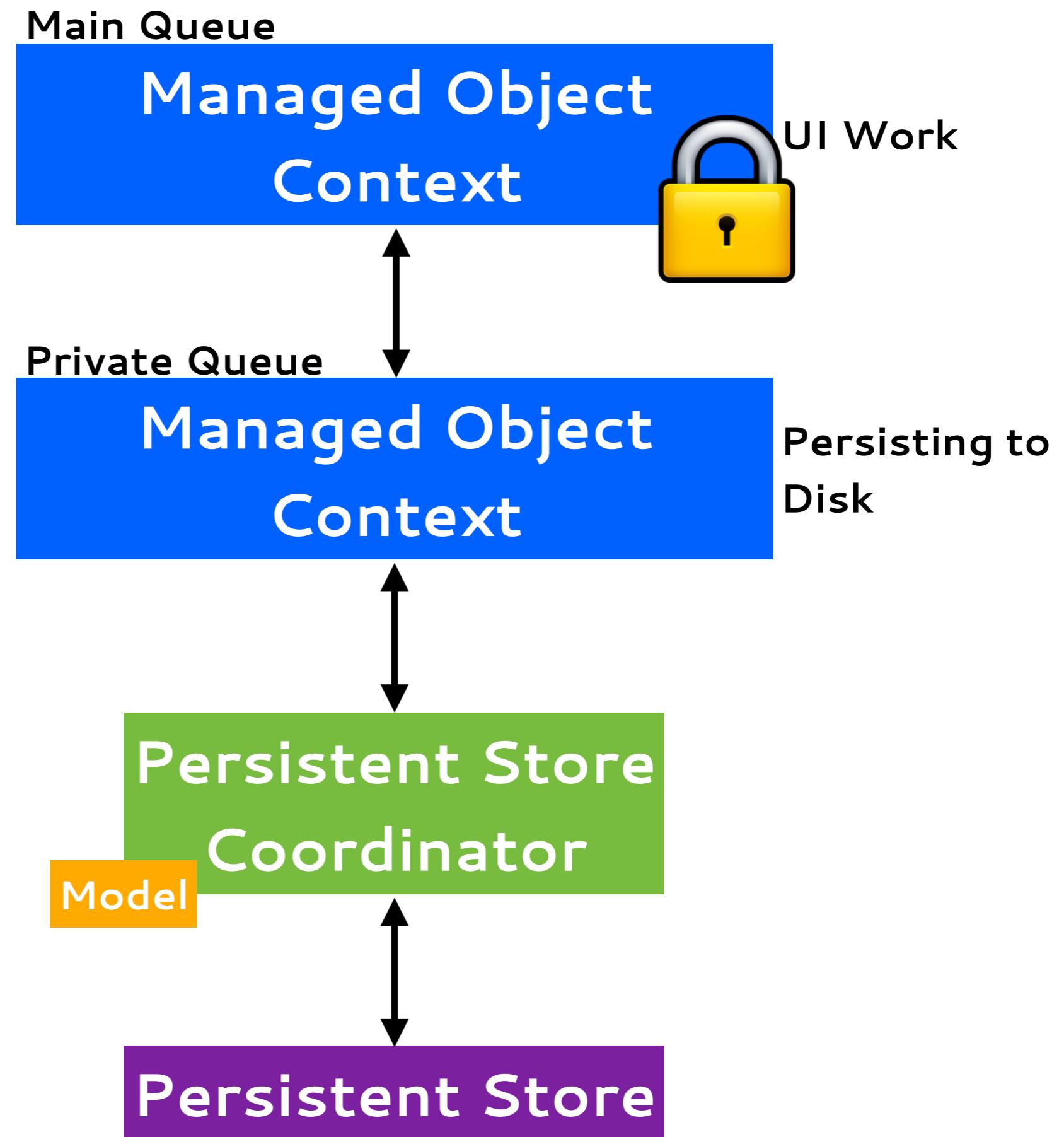


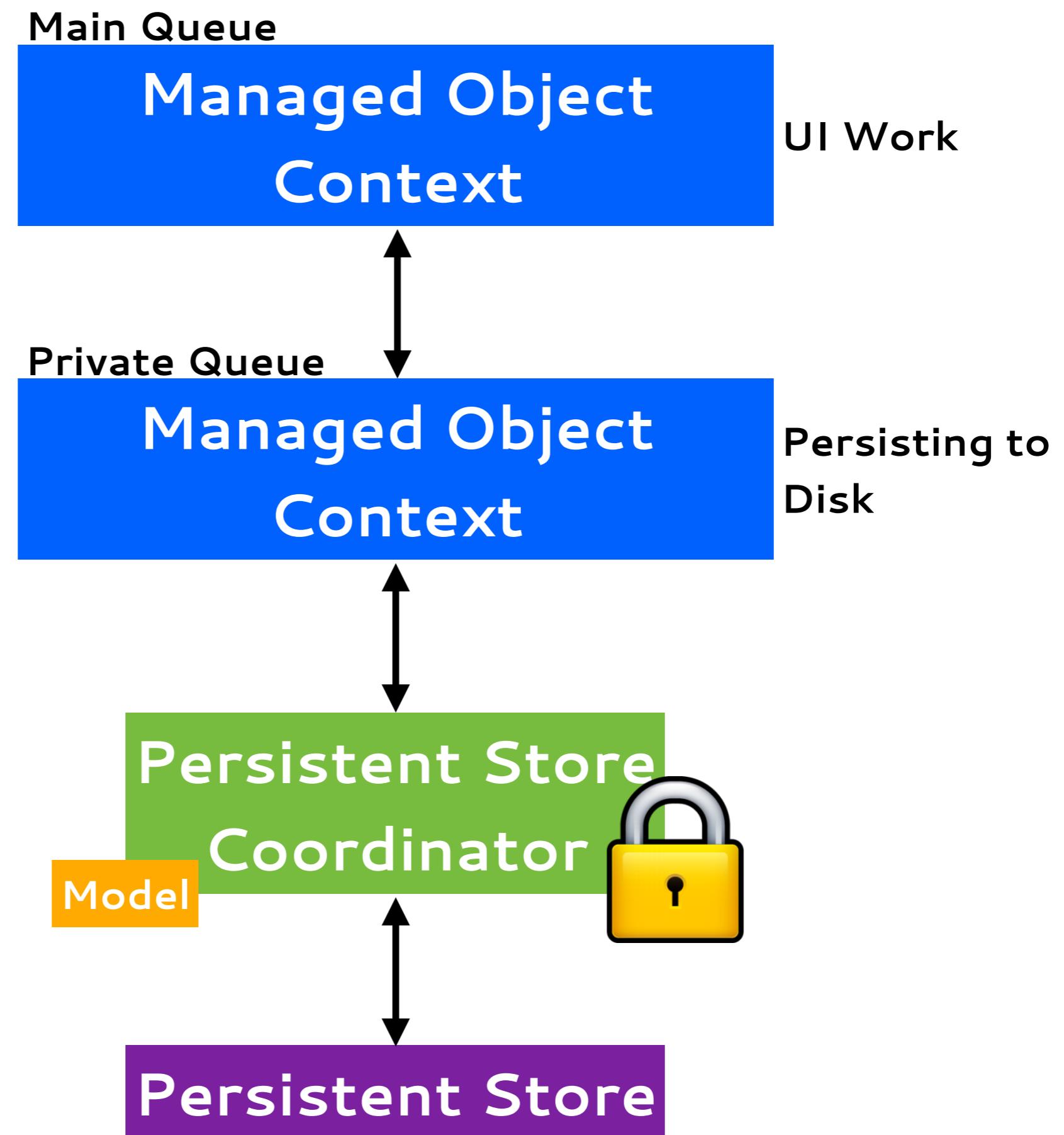


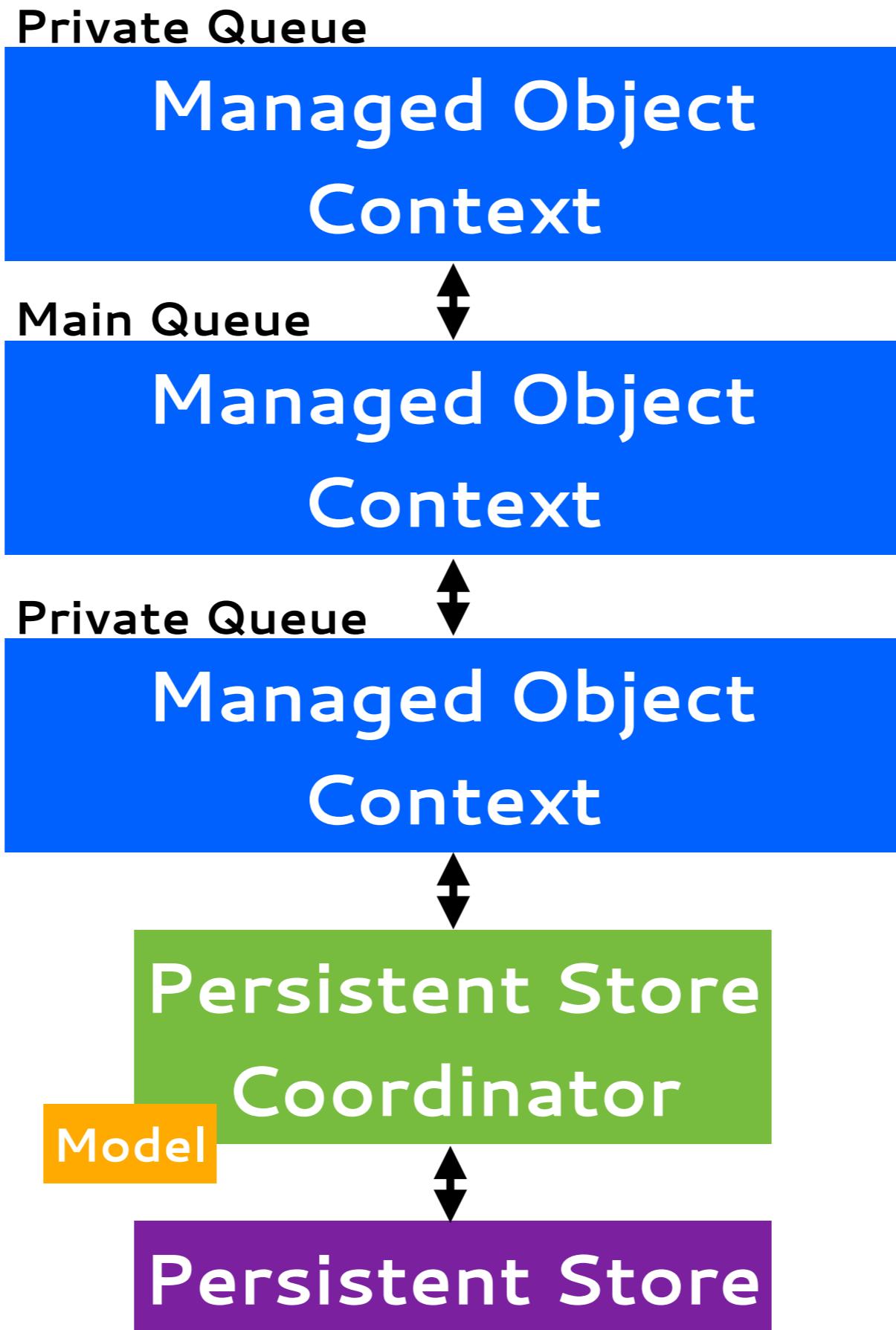


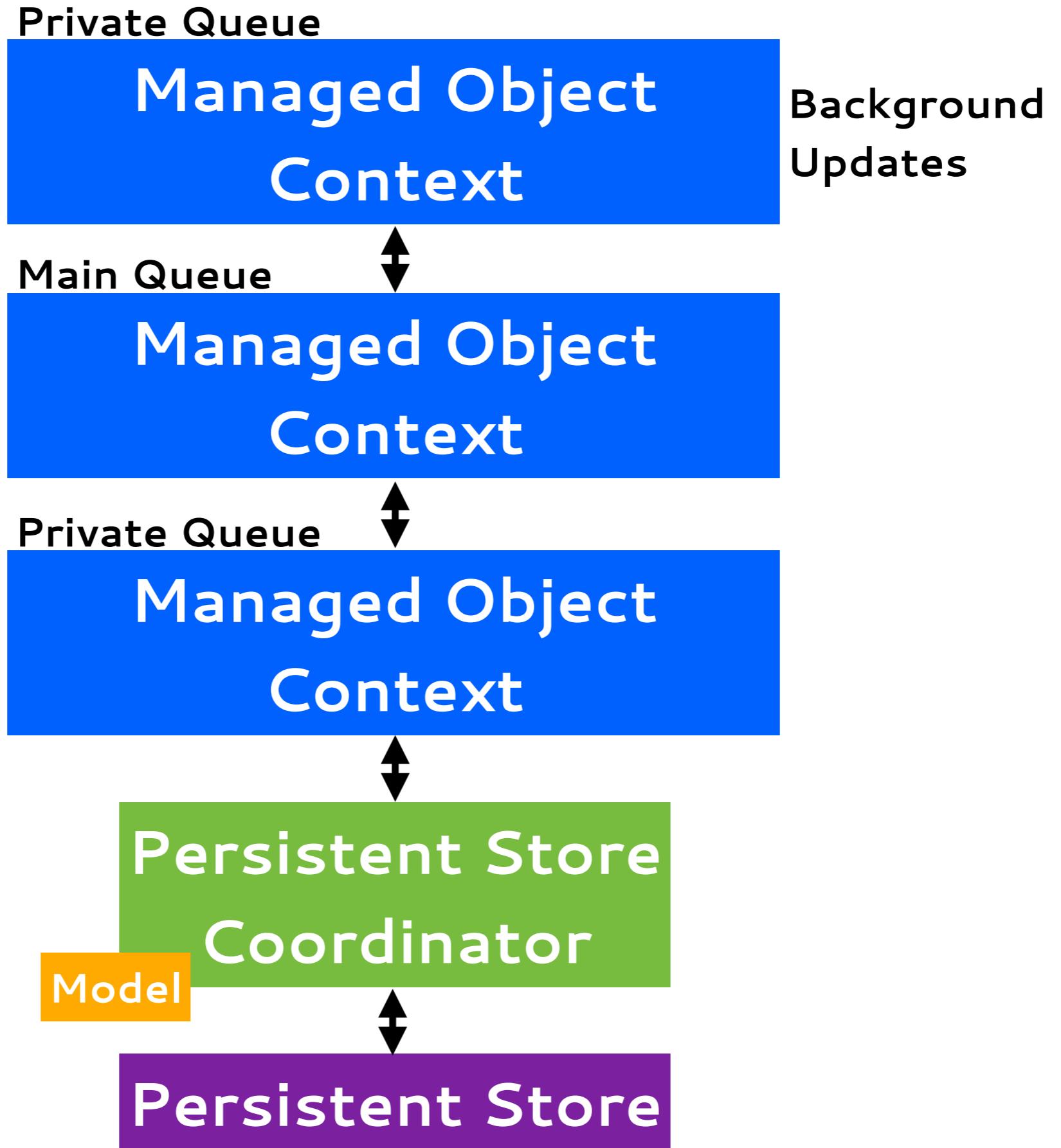


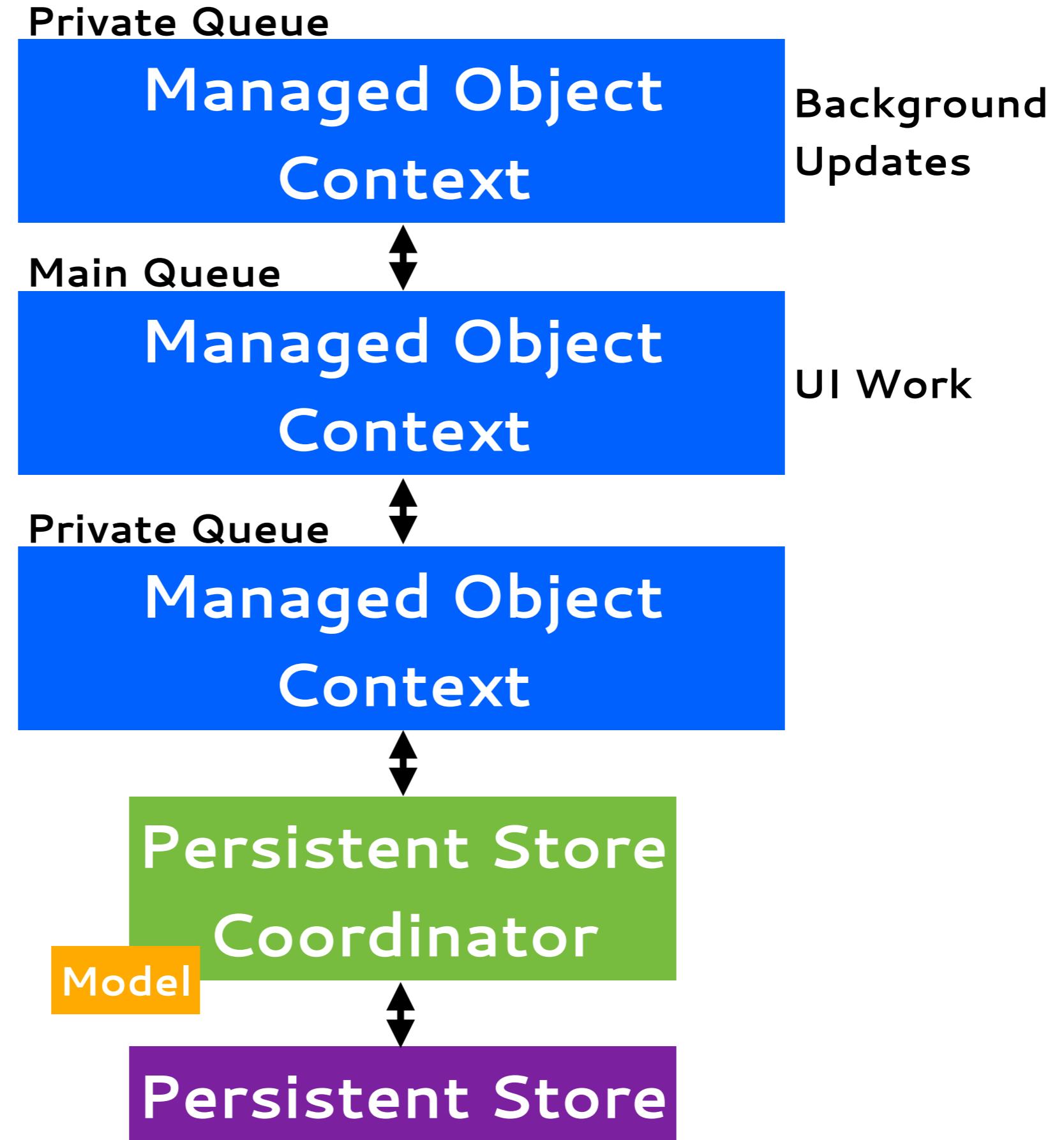


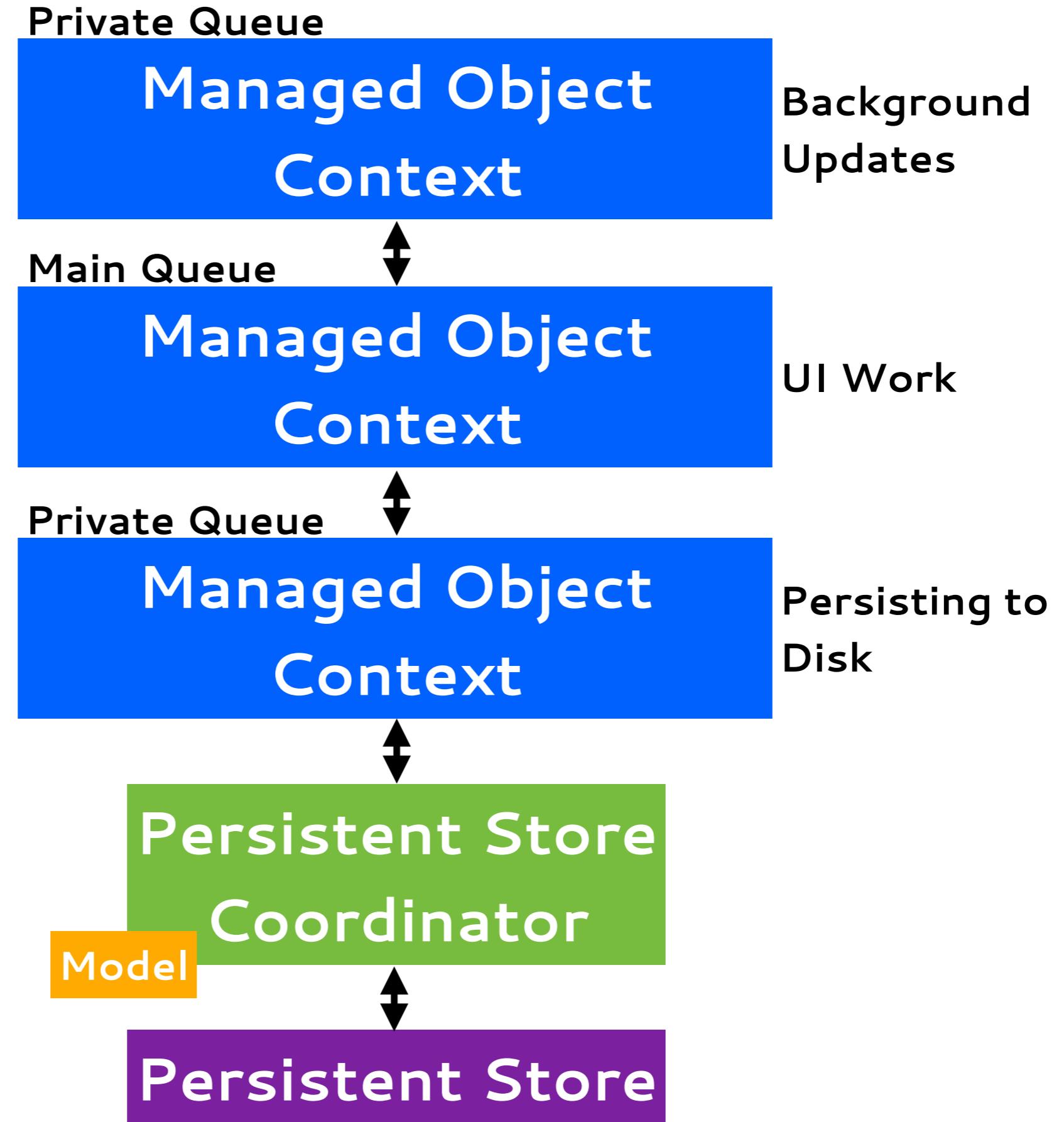


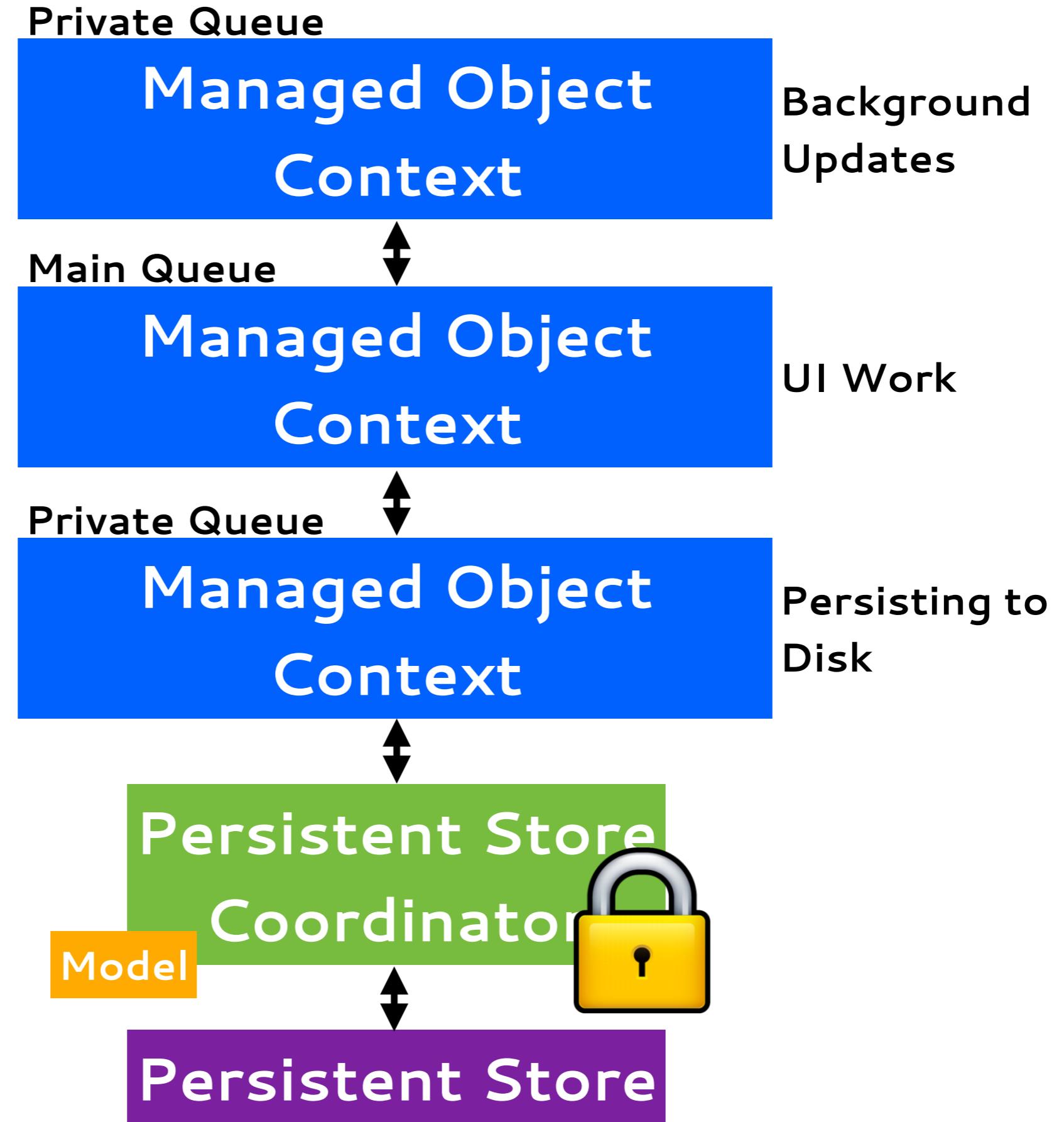












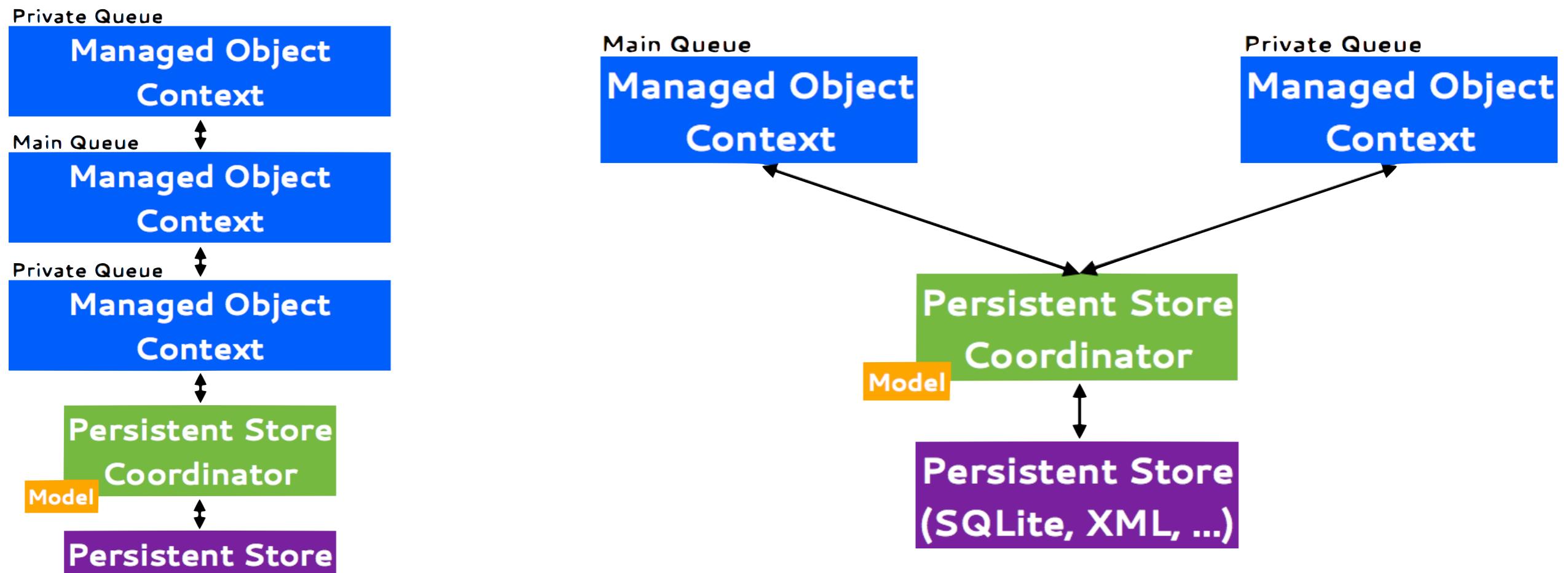
```
__block NSManagedObjectContext *temporaryContext = self.workerObjectContext;
__block NSManagedObjectContext *managedObjectContext = self.managedObjectContext;
__block NSManagedObjectContext *writerObjectContext = self.writerManagedObjectContext;

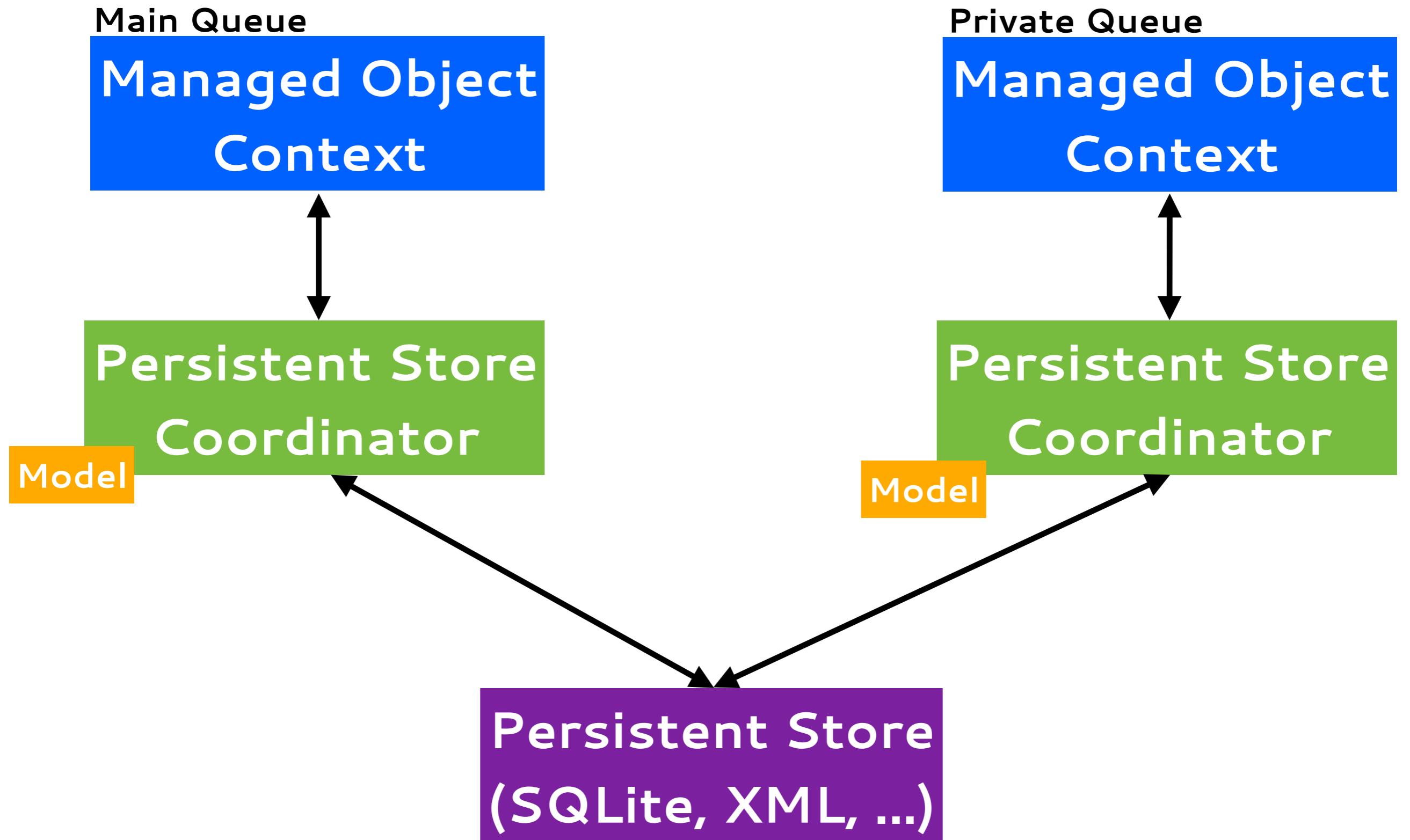
[temporaryContext performBlock:^{
    NSError *error = nil;
    if (![temporaryContext save:&error]) {
        // TODO: Handle error
    }
}

[managedObjectContext performBlock:^{
    NSError *error = nil;
    if (![managedObjectContext save:&error]) {
        // TODO: Handle error
    }
}

[writerObjectContext performBlock:^{
    NSError *error = nil;
    if (![writerObjectContext save:&error]) {
        // TODO: Handle error
    }

    // Success!!!
}]; // writerObjectContext
}]; // managedObjectContext
}]; // temporaryContext
```





Main Queue

Managed Object
Context

Private Queue

Managed Object
Context

Persistent Store
Coordinator

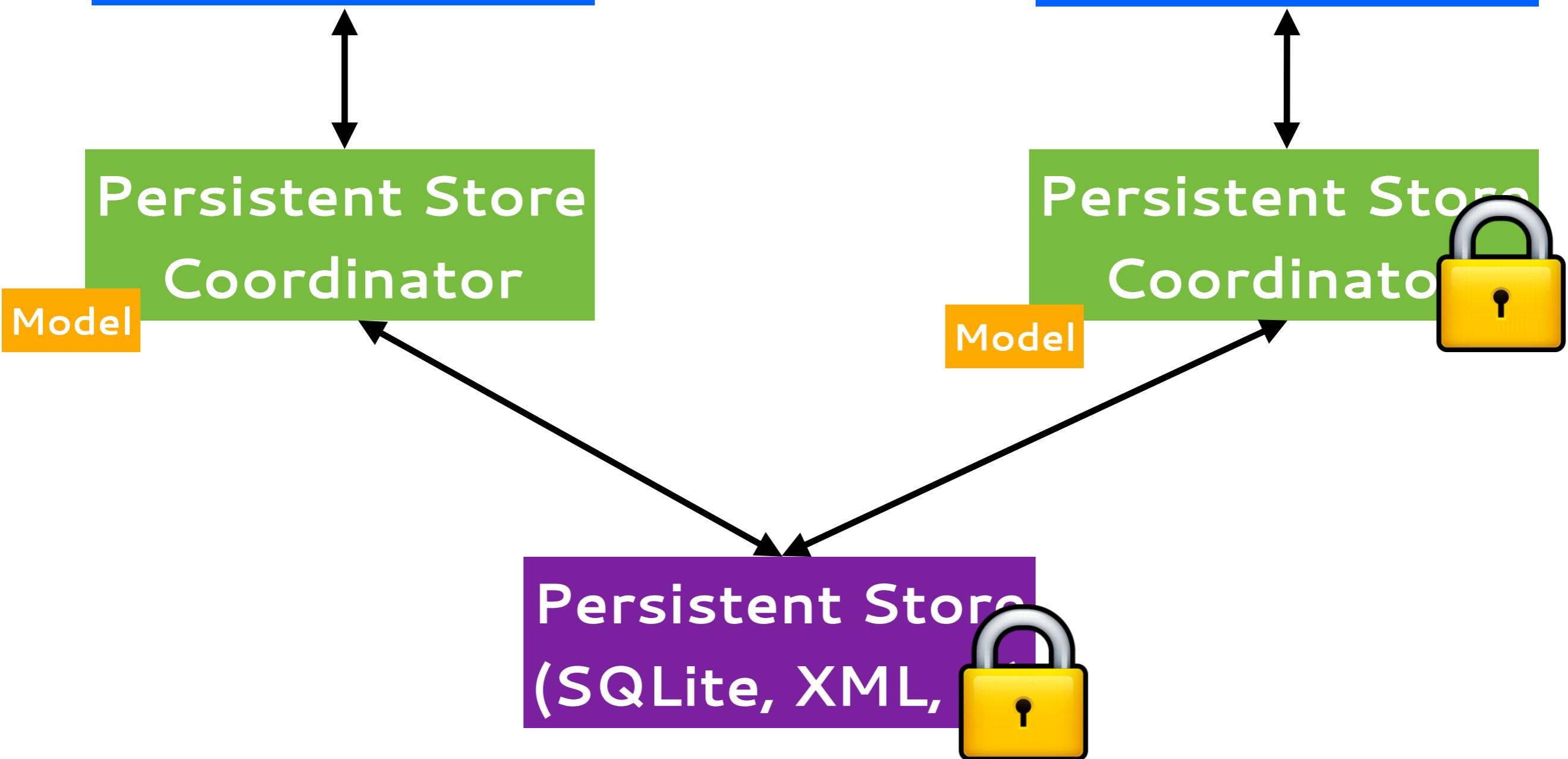
Model

Persistent Store
Coordinator

Model



Persistent Store
(SQLite, XML,



Main Queue

Managed Object
Context

Private Queue

Managed Object
Context

Persistent Store
Coordinator

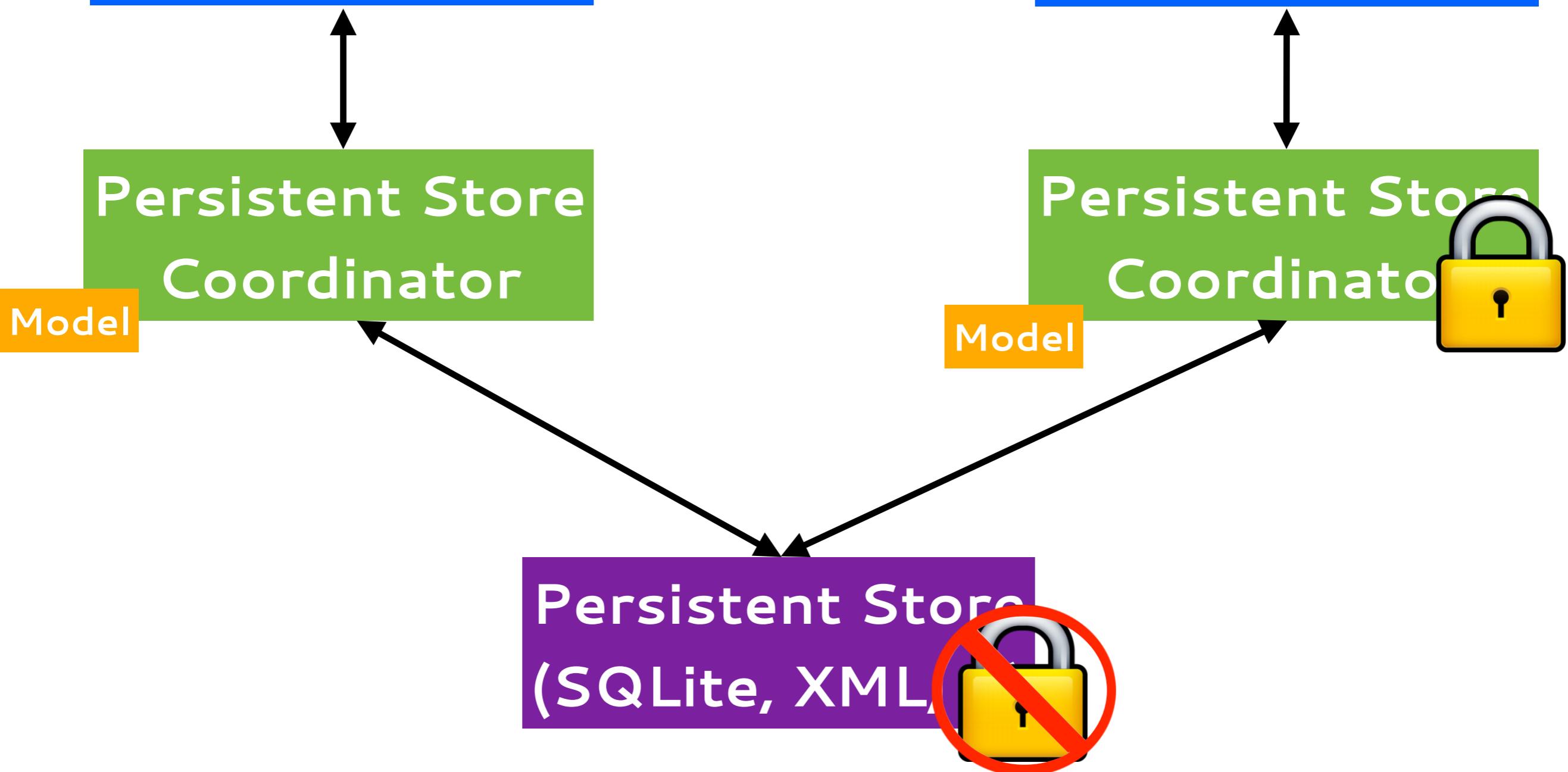
Model

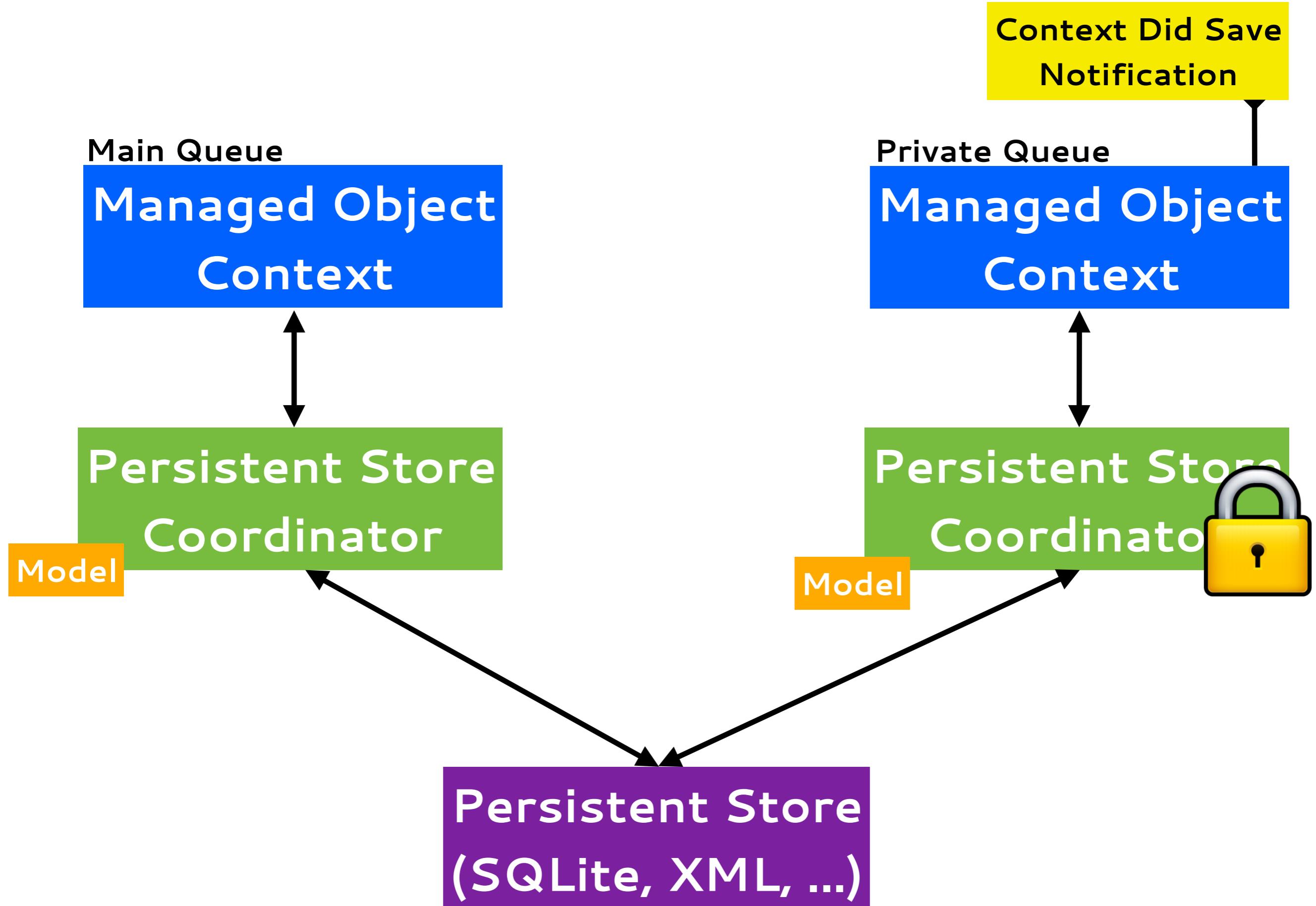
Persistent Store
Coordinator

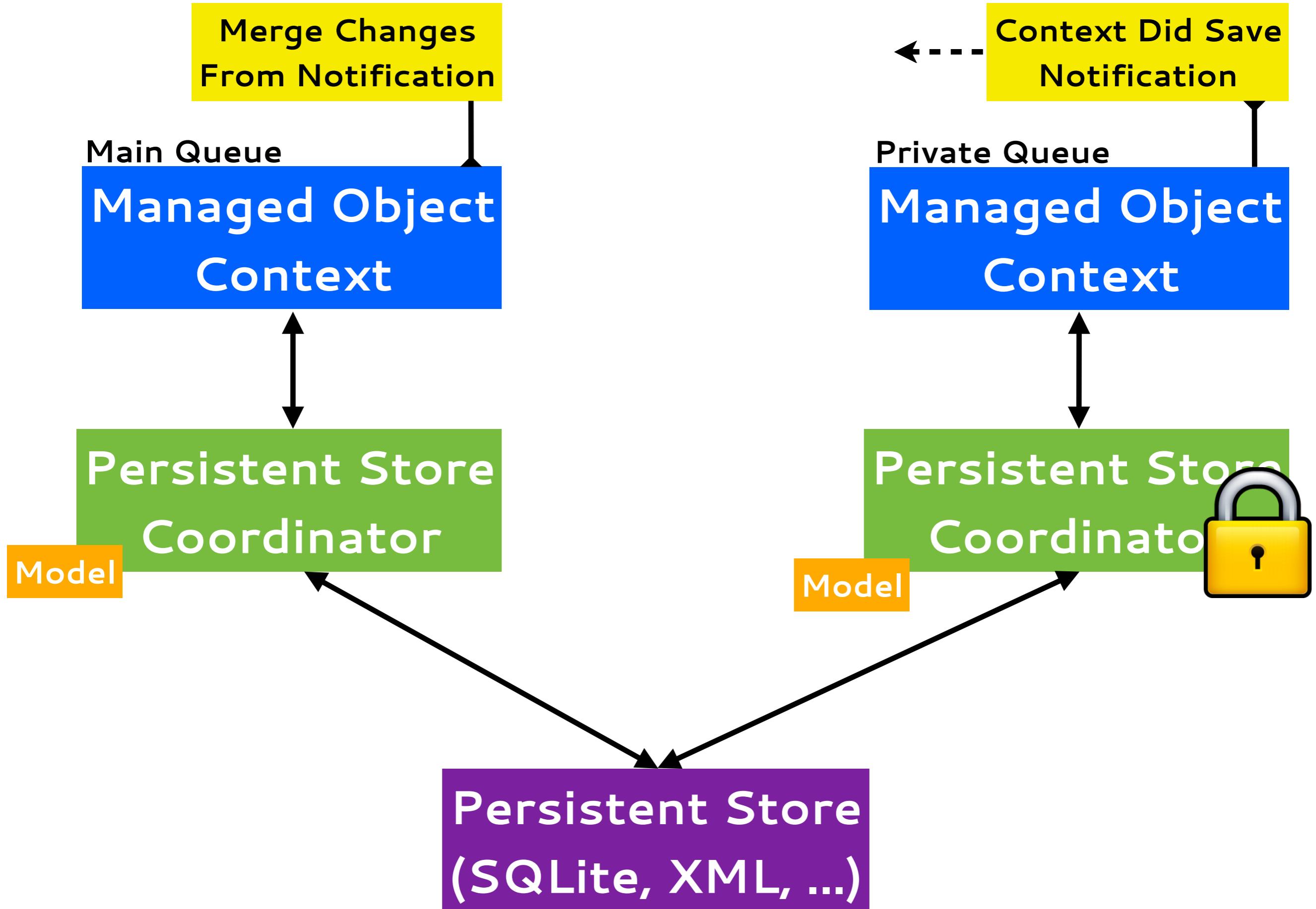
Model

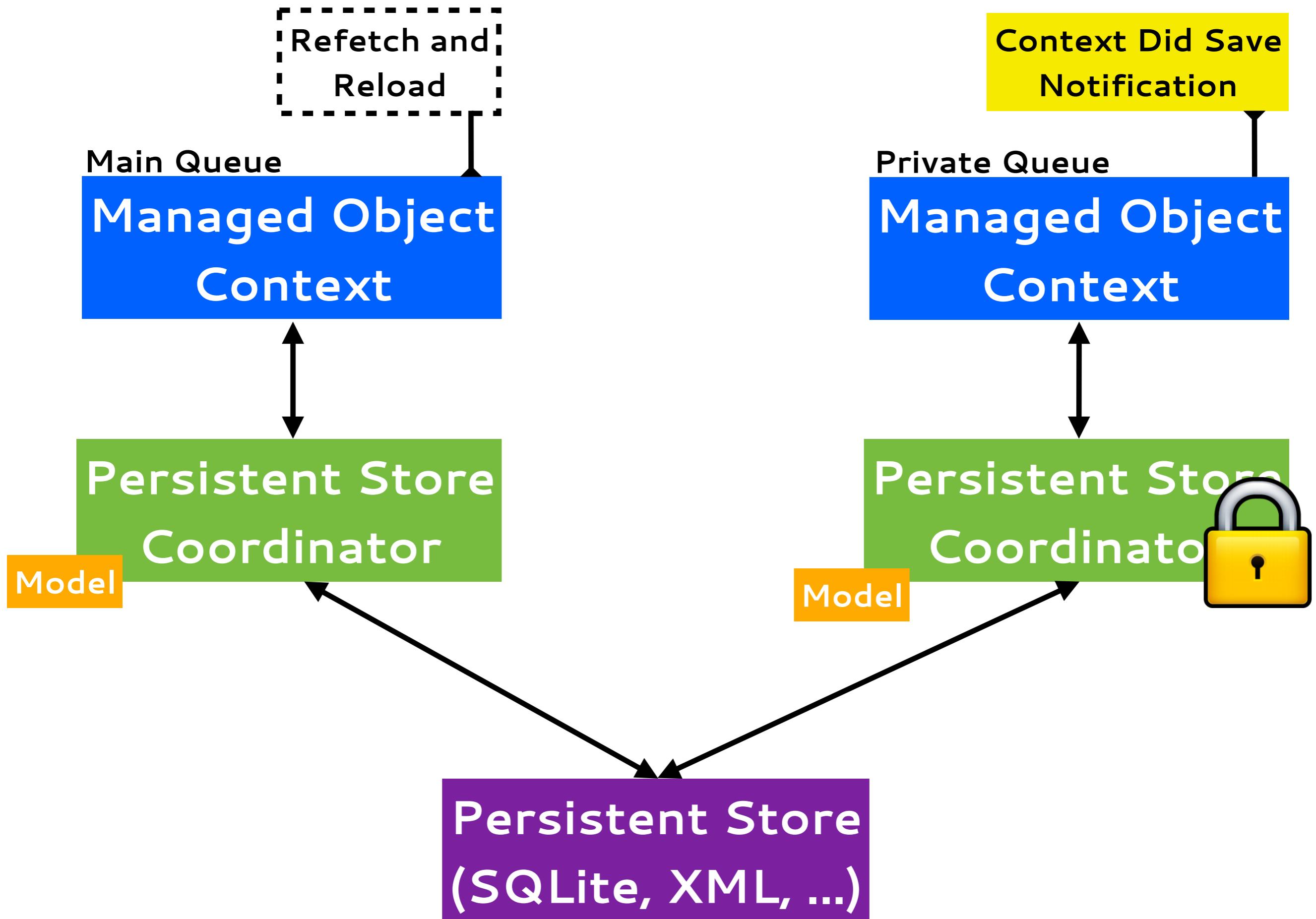


Persistent Store
(SQLite, XML, ...)









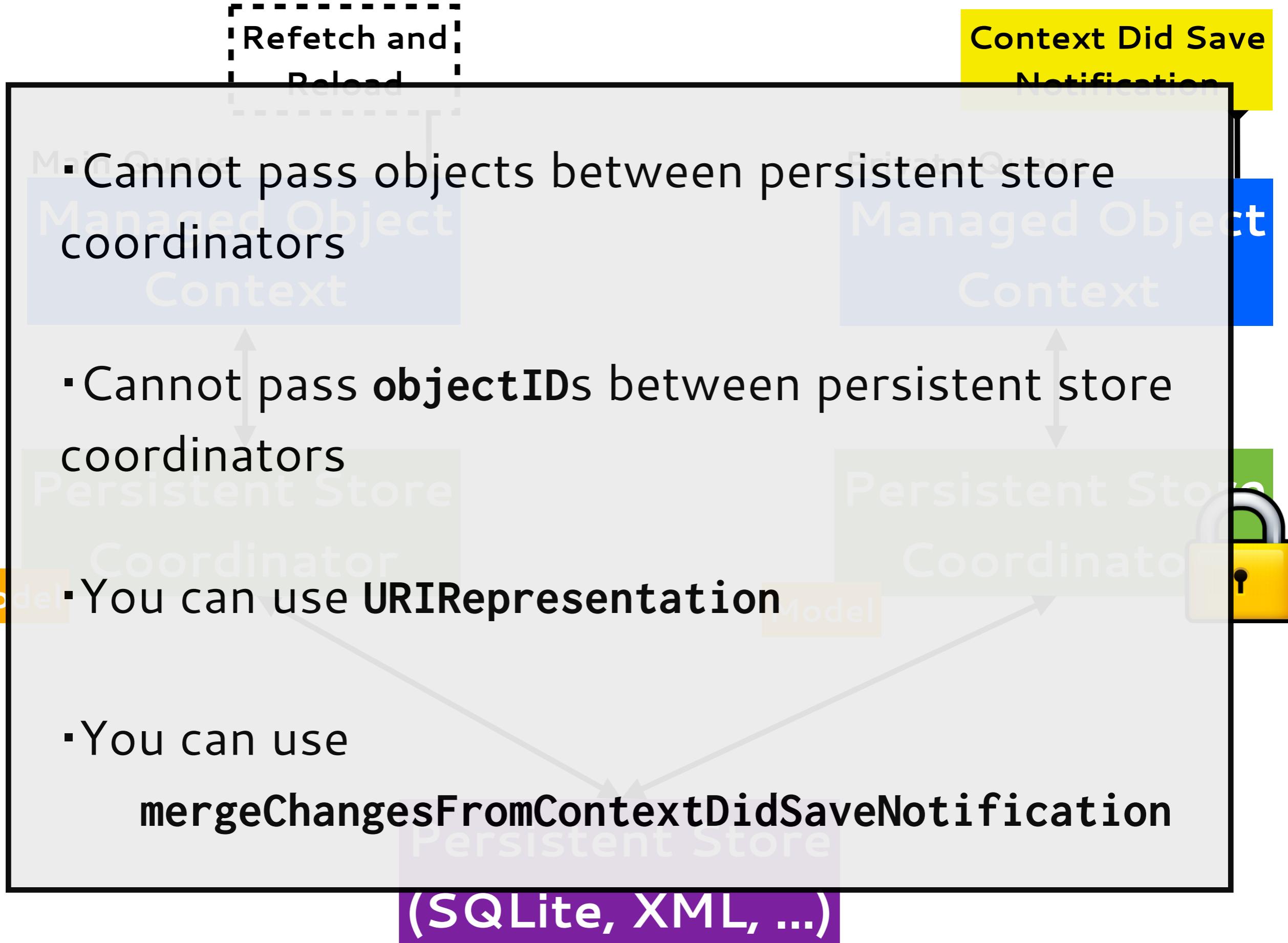
```
116 // View controllers should reload/refresh after import
117 // Takes advantage of SQLite WAL mode (only available
118 //
119 self.managedObjectContext = [self.persistenceStack newPrivateContext];
120 self.managedObjectContext.undoManager = nil;
121 [self.managedObjectContext performBlockAndWait:^{
122     self.persistenceStack.disableMergeNotifications = YES;
123     [self batchImport];
124     self.persistenceStack.disableMergeNotifications = NO;
125 }
126 // Notify everyone that an import operation has been completed
127 [[NSNotificationCenter defaultCenter] postNotificationName:@"objectImported"
128 object:[batchImport objects]];
129 }];
130
131 }
132
133 - (void)saveManagedObjectContext {
134
135     NSError *saveError;
```

/Users/xzolian/xcode/projects/MDMHPCoreData/2/MDMHPCoreData/MDMHPCoreData/model/

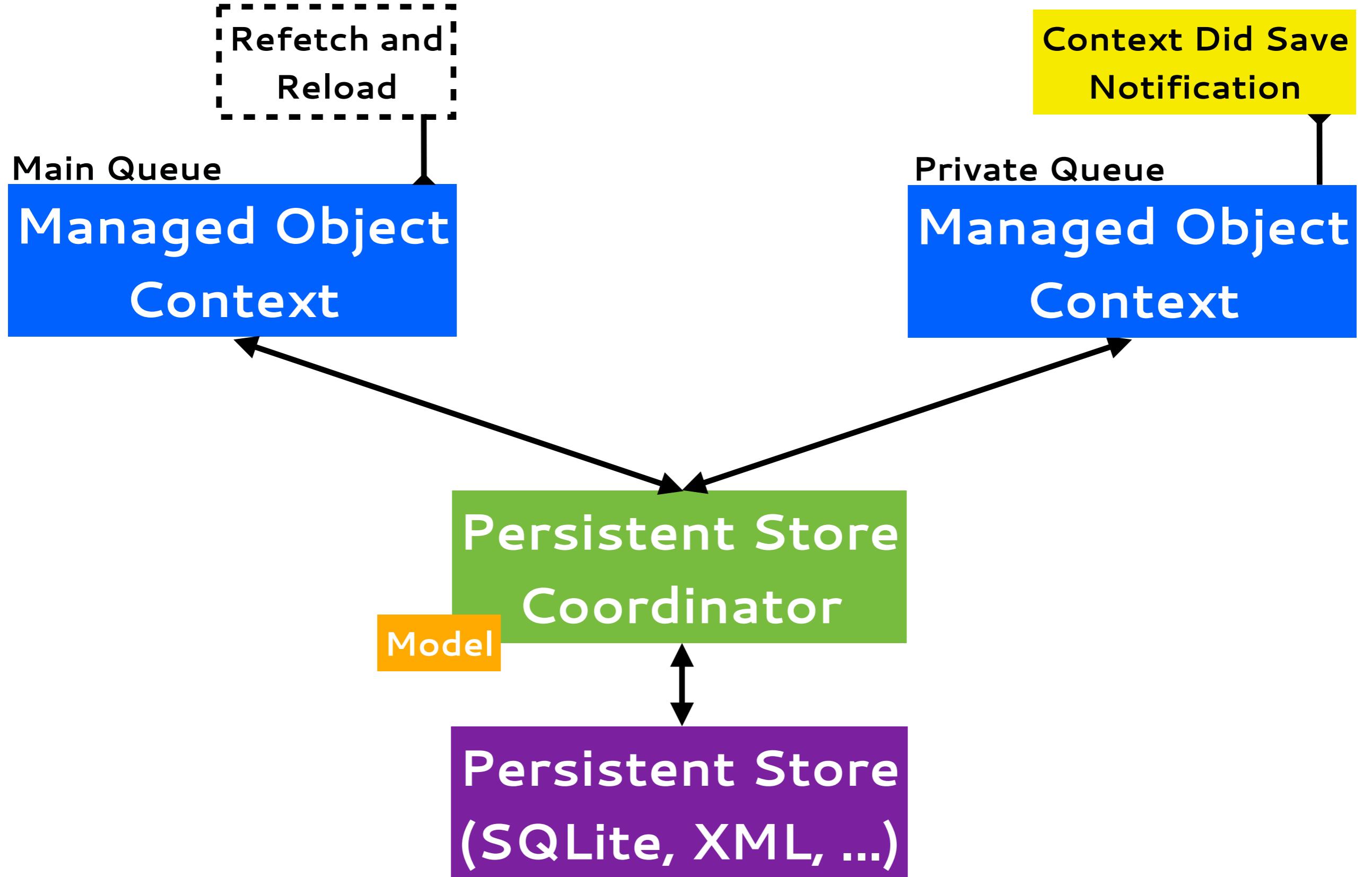
No Selection

2013-11-10 16:25:39.012 MDMHPCoreData[15098:1303] Progress: 0.968546
2013-11-10 16:25:39.046 MDMHPCoreData[15098:1303] Progress: 0.978531
2013-11-10 16:25:39.085 MDMHPCoreData[15098:1303] Progress: 0.988516
2013-11-10 16:25:39.123 MDMHPCoreData[15098:1303] Progress: 0.998501
2013-11-10 16:25:39.129 MDMHPCoreData[15098:1303] Progress: 1.000000

All Output

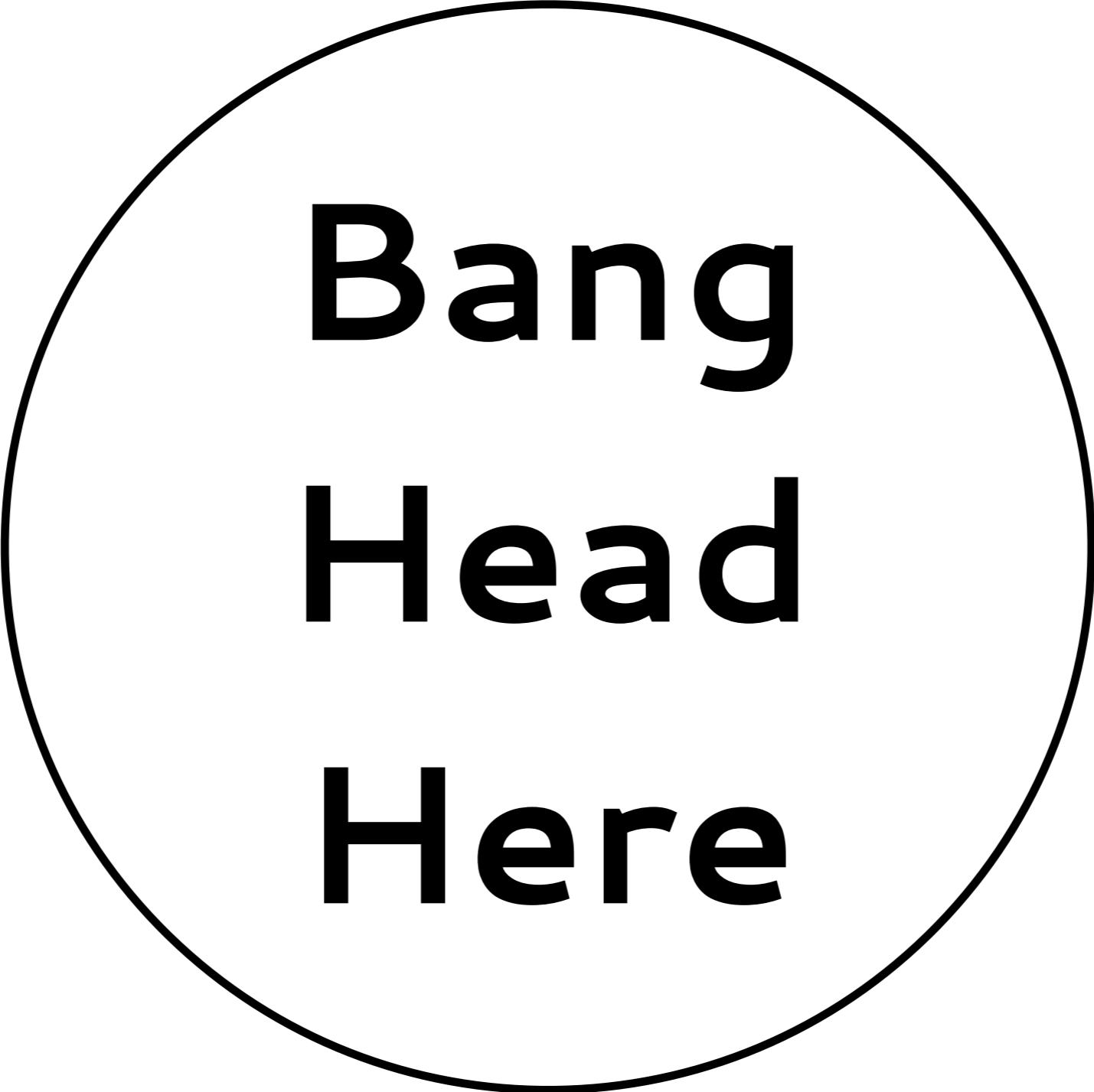


TLDR



Concurrency Models

- Use the simplest model that meets your needs**
- Update main context after large imports have completed**



**Bang
Head
Here**

In general...

- Use the tools, and measure, measure, measure
- Don't load more than you need to
- Don't use [cd]
- Be smart with your data model
- Import on a private queue in batches
- Pick the correct (but simplest) concurrency model

Questions?

High Performance Core Data.com

MatthewMorey.com | @xzolian