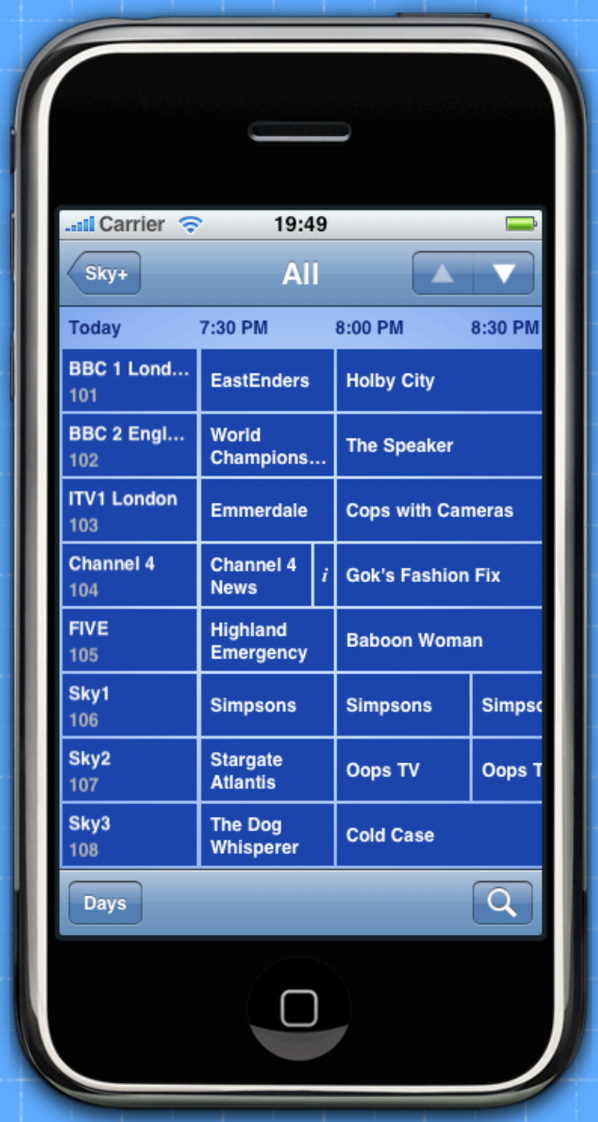
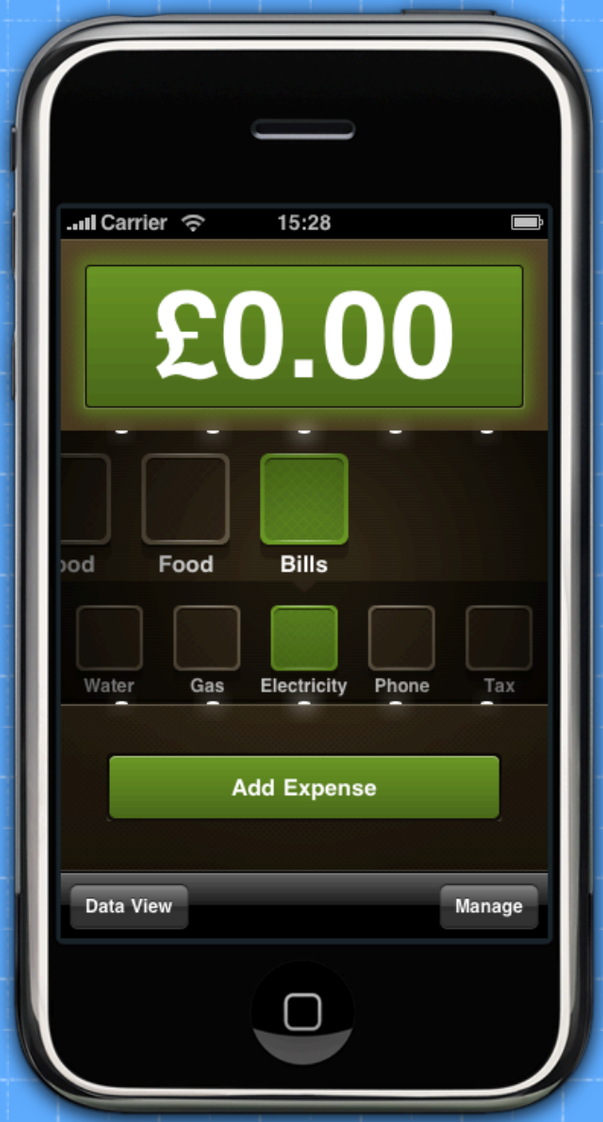
# Introduction to Core Data

*Daniel Tull*

Sky+

Little Spender





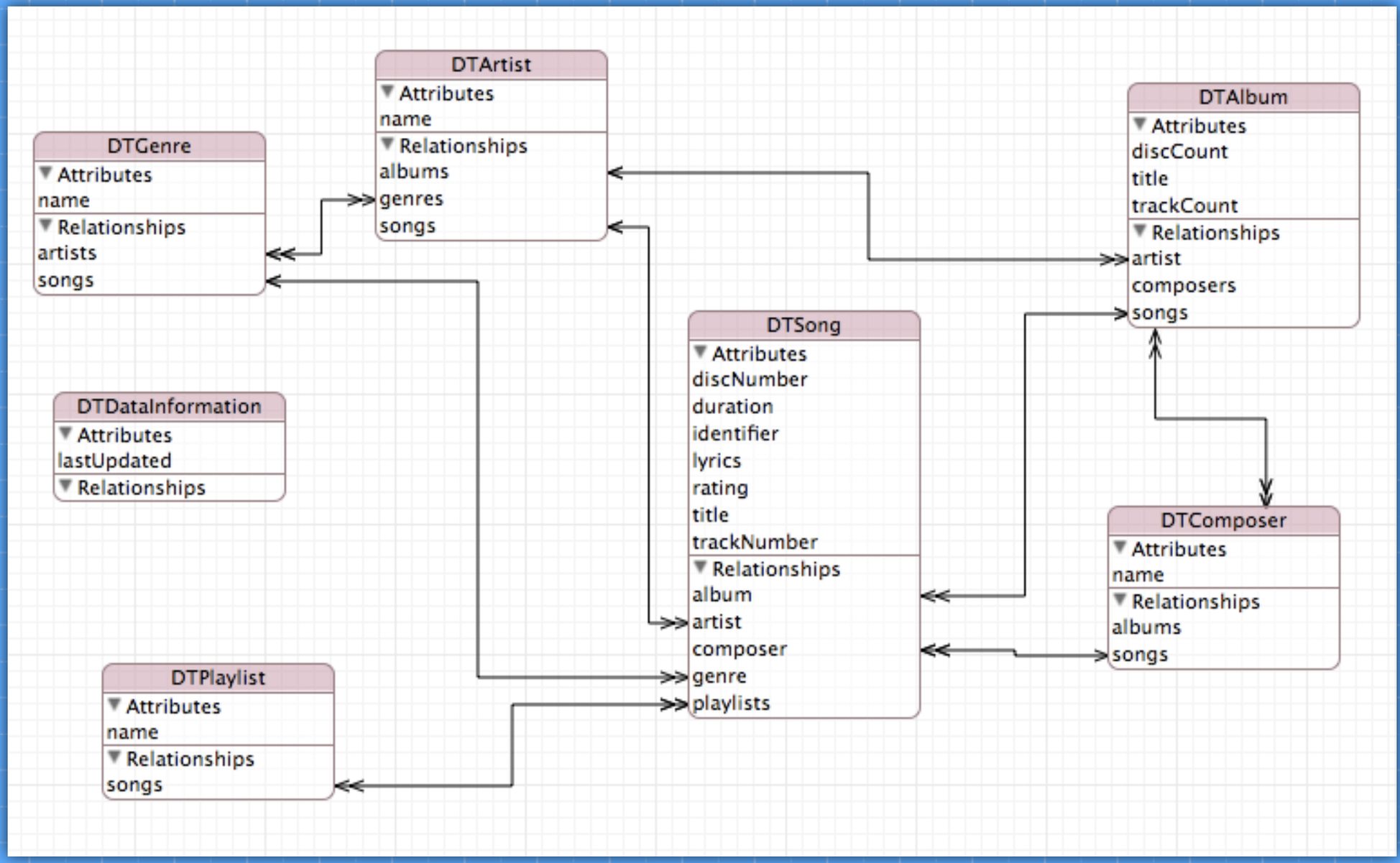
Weather Maps

Fourgy

# Core Data Stack

NSManagedObjectModel

# Managed Object Model



# Attributes

NSNumber

NSString

BOOL

NSDate

NSData

Transformable

**Transformable**

You put in code to transform the object or value into a compatible type.

# Relationships

Properties that reference other objects

Can be to-one or to-many (NSSet - unordered)

Can (and should) have inverses, which are worked out by Core Data when you set the other property

Delete rule ensures propagation

# Relationships

**DTGenre**

*Attributes*

name

*Relationships*

artists

songs

**DTArtist**

*Attributes*

name

*Relationships*

albums

genres

songs

**DTAlbum**

*Attributes*

name

discCount

trackCount

*Relationships*

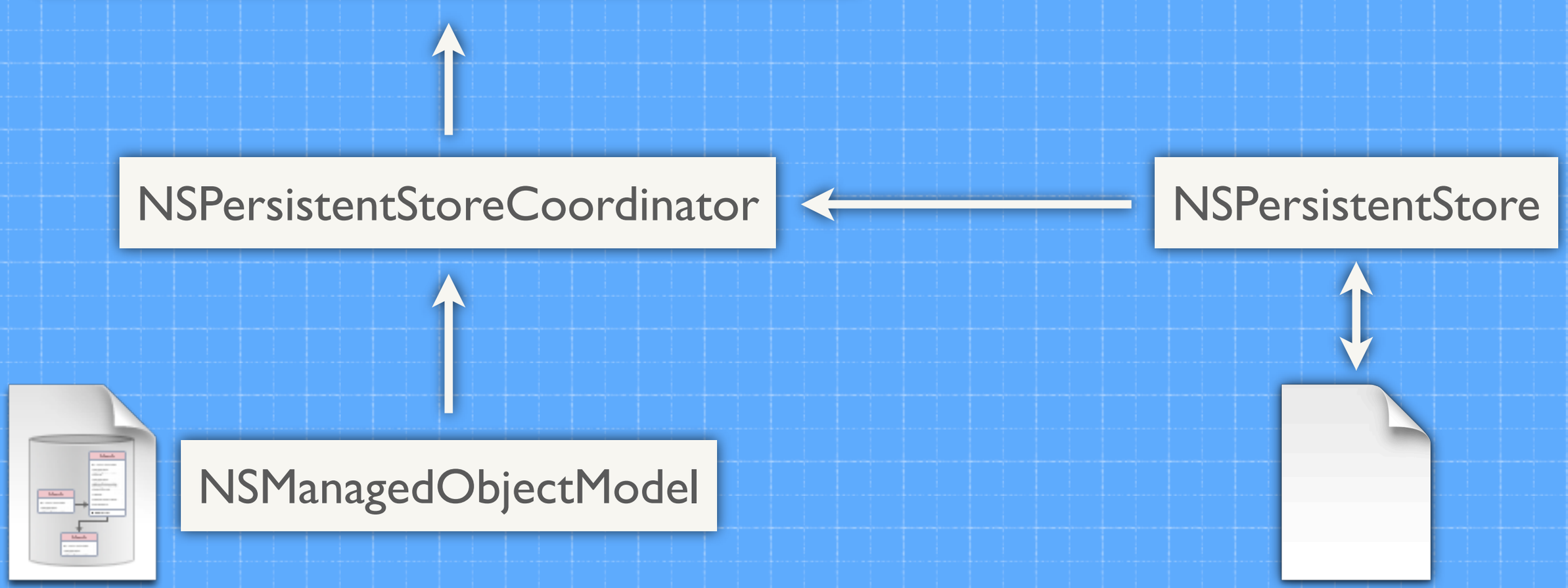
artist

composers

songs

# Core Data Stack

NSManagedObjectContext



# Create and Save Objects

```
DTSong *song = [NSEntityDescription
                    insertNewObjectForEntityForName:@“DTSong”
                            inManagedObjectContext:managedObjectContext];

DTArtist *artist = [NSEntityDescription
                    insertNewObjectForEntityForName:@“DTArtist”
                            inManagedObjectContext:managedObjectContext];

artist.name = @“Moby”;

song.title = @“Porcelain”;

song.artist = artist;

   ...

NSError *error = nil;
[managedObjectContext save:&error];
```

# Create and Save Objects

NSManagedObjectContext

DTArtist

DTSong

NSPersistentStoreCoordinator

NSPersistentStore

NSManagedObjectModel

# Fetching Objects

All songs by “La Roux” sorted by title.

| | |
|---|---|
| Entity | `DTSong` |
| Predicate | `@“artist.name == %@”, @“La Roux”` |
| Sorting | `@“title”` |

# Fetching Objects

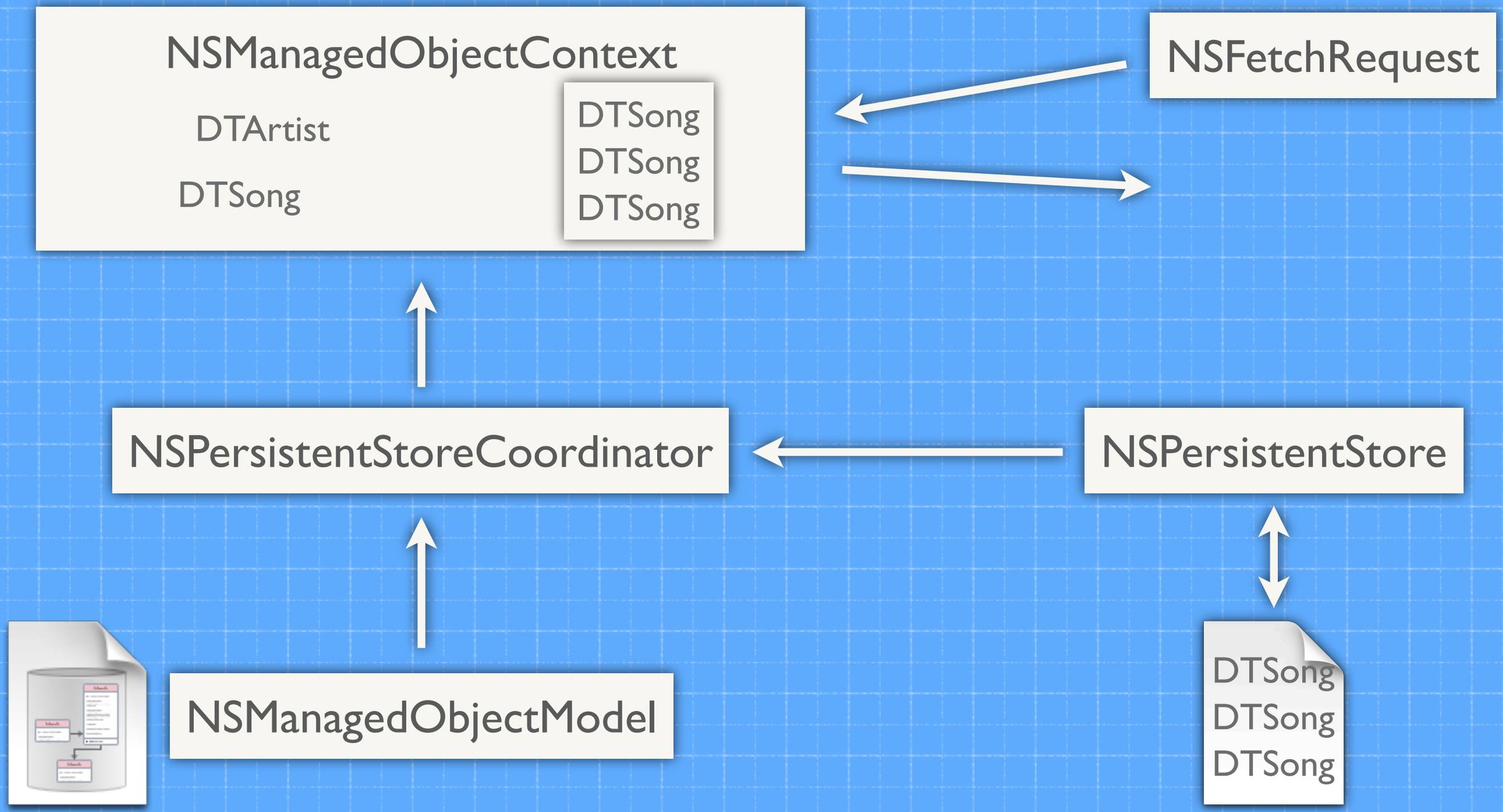
Fetch all playlists which contain more than an hour of music.

| | |
|---|---|
| Entity | `DTPlaylist` |
| Predicate | `@“@sum.songs.duration > 3600”` |
| Sorting | `@“name”` |

# Fetching Objects

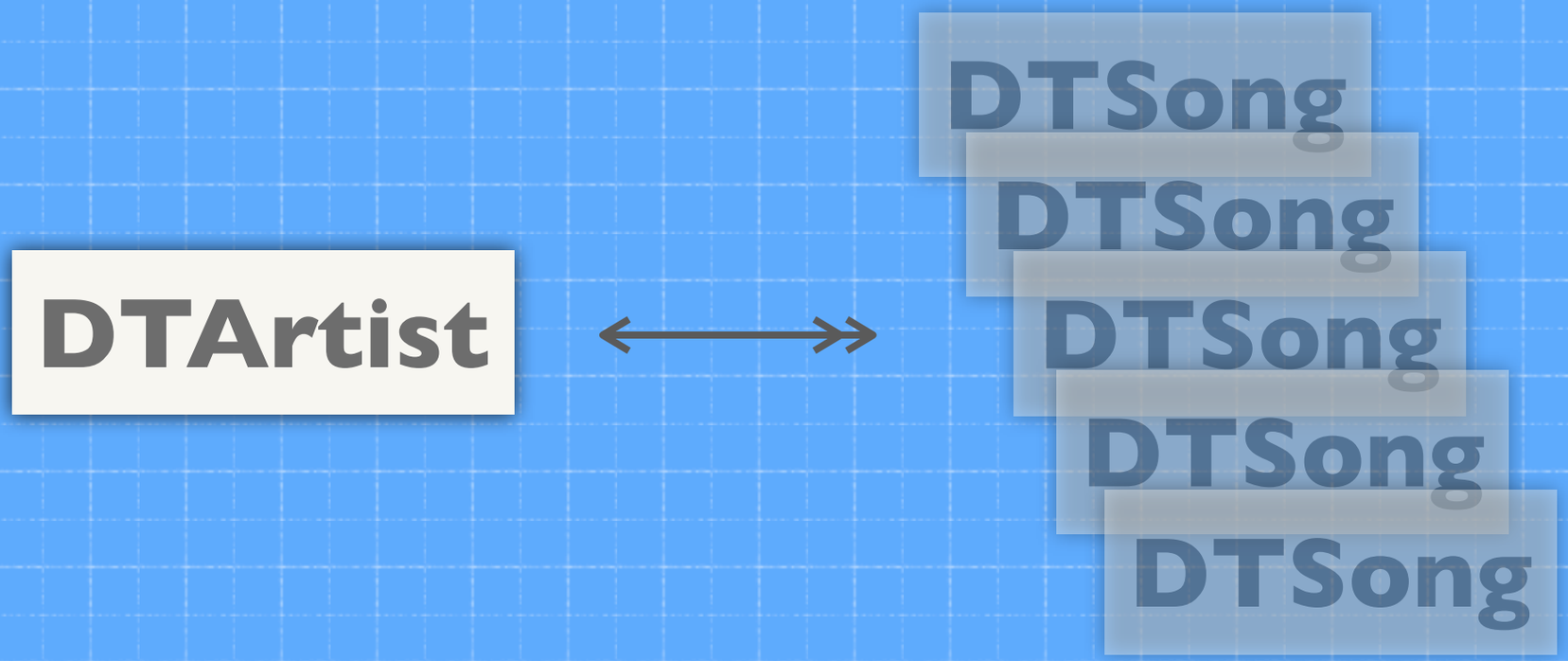
```
NSEntityDescription *entity = [NSEntityDescription
                                    entityForName:@“DTSong”
                                inManagedObjectContext:managedObjectContext];

NSPredicate *predicate = [NSPredicate
                        predicateWithFormat:@“artist.name == %@”, @“La Roux”];

NSSortDescriptor *sortDescriptor = [[NSSortDescriptor alloc]
                                            initWithKey:@“title” ascending:YES];

NSArray *descriptors = [NSArray arrayWithObject:sortDescriptor];

NSFetchRequest *request = [[NSFetchRequest alloc] init];
[request setEntity:entity];
[request setPredicate:predicate];
[request setSortDescriptors:sortDescriptors];

NSArray *fetchResult = [managedObjectContext executeFetchRequest:request
                                                            error:&error];
```

# Fetching Objects



# Performance: Faults

Faults are placeholders for related objects, which mean you don't use memory on unwanted objects.



# Performance: Batching

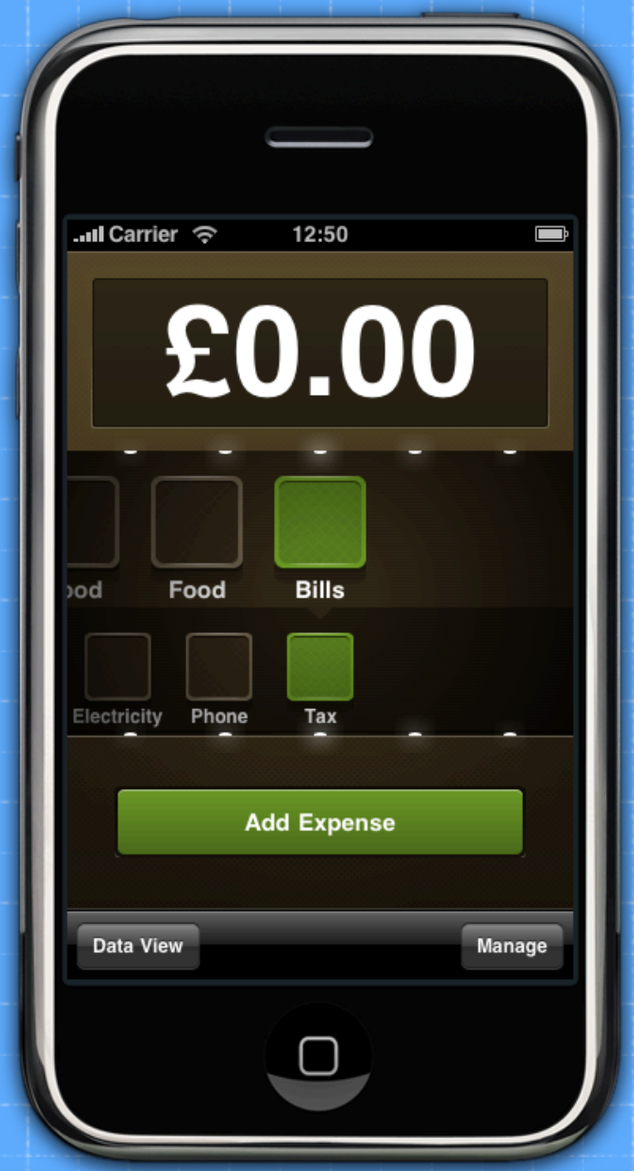
Batching allows you to bring only a subset of objects into memory.

# Demo

# DTMusicModel

*hg.danieltull.co.uk/dtmusicmodel*

# DTGridView




# DTSwapView

*hg.danieltull.co.uk/dtkit*

# Thank You

*dt@danieltull.co.uk*

*hg.danieltull.co.uk*