

Лекция 8.0

Persistence Layer

CoreData or DataCore



База данных ???

И да и нет

| ORM ???

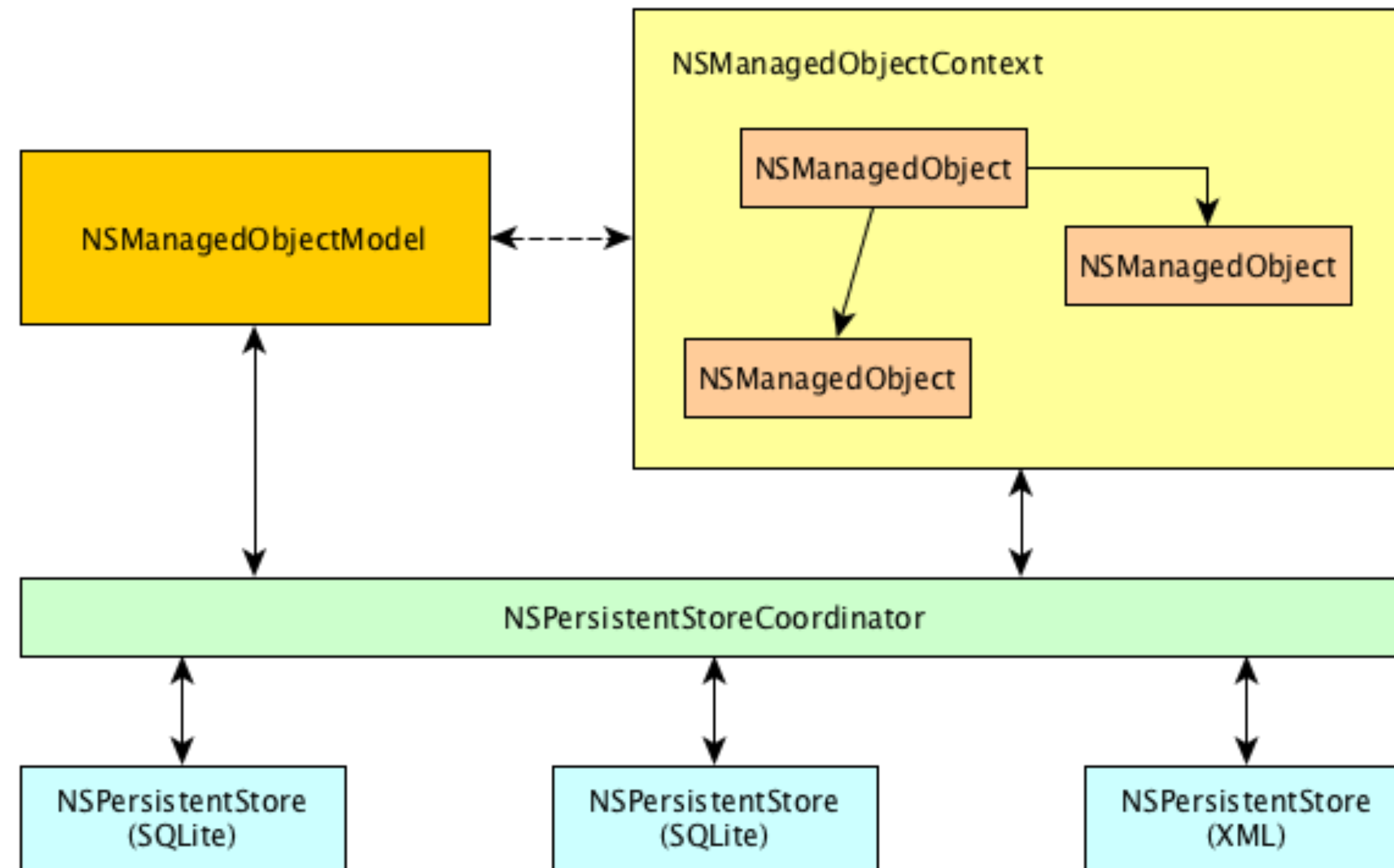
И да и нет

Core Data

Объектная БД с дополнительными штуками

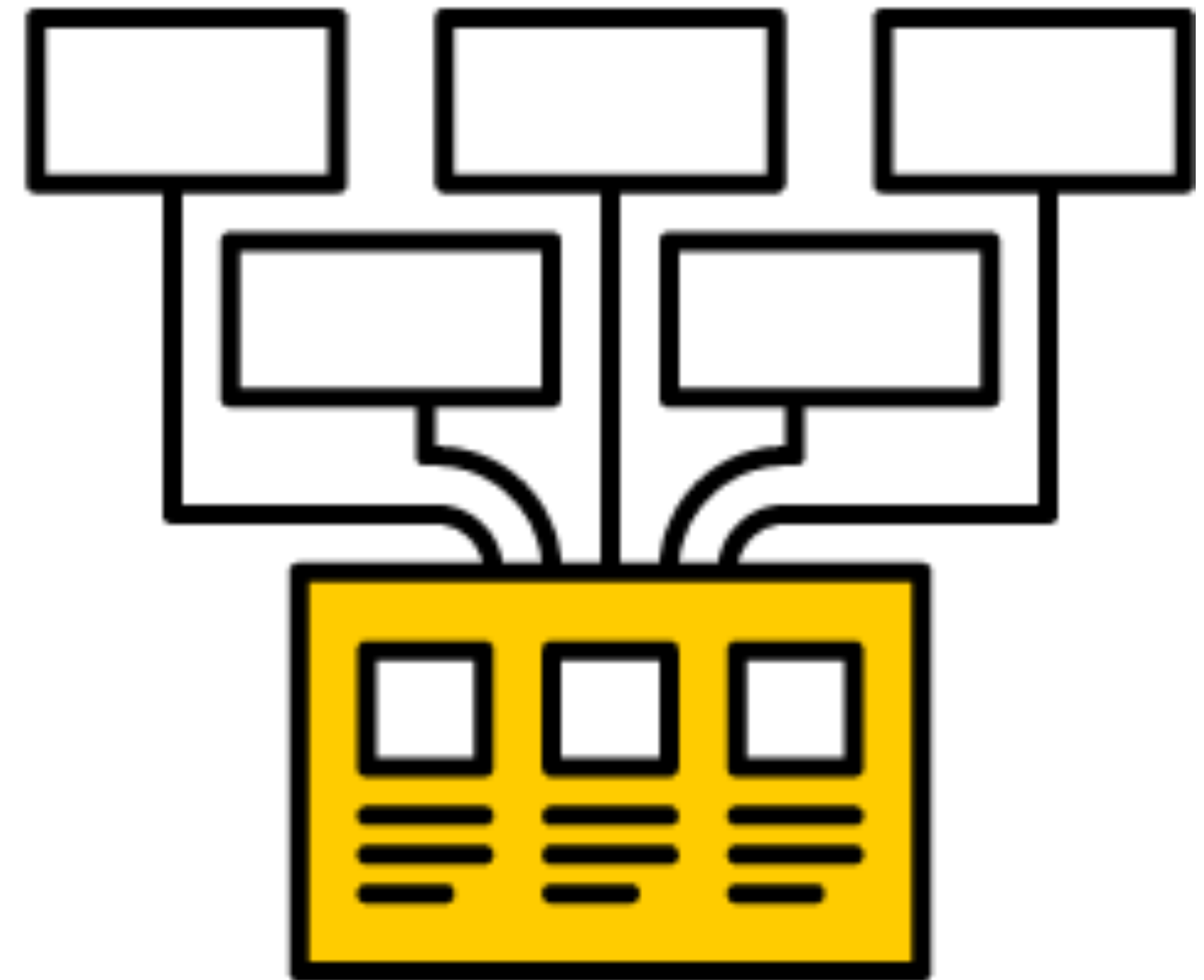


CoreData Stack

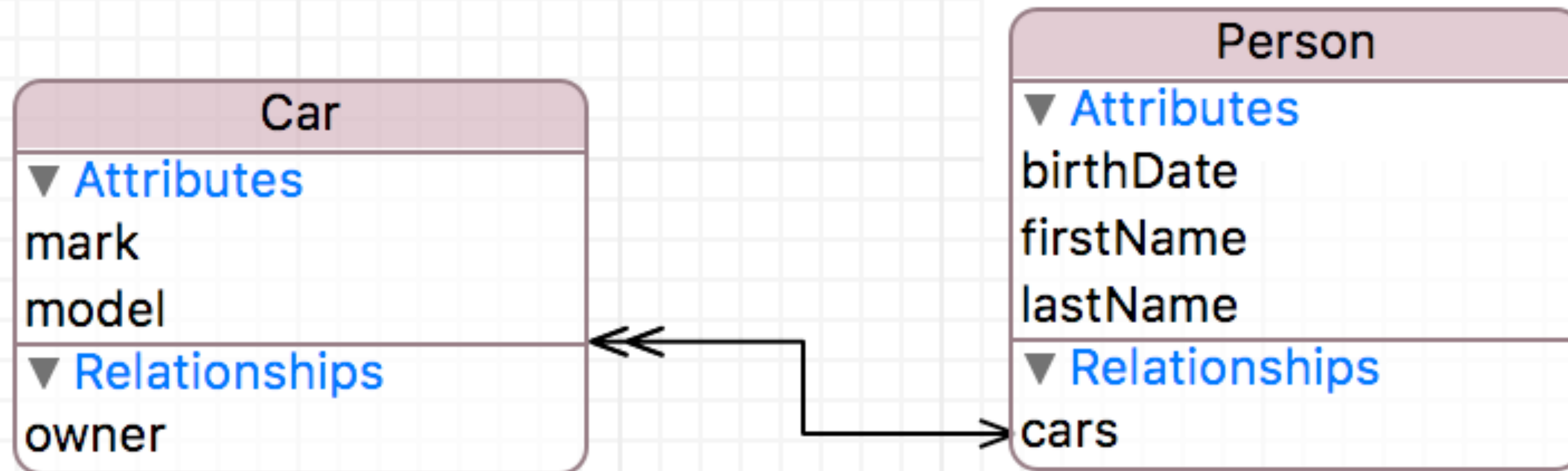


NSManagedObjectModel

› Описывает объектную модель хранимых данных



NSManagedObjectModel



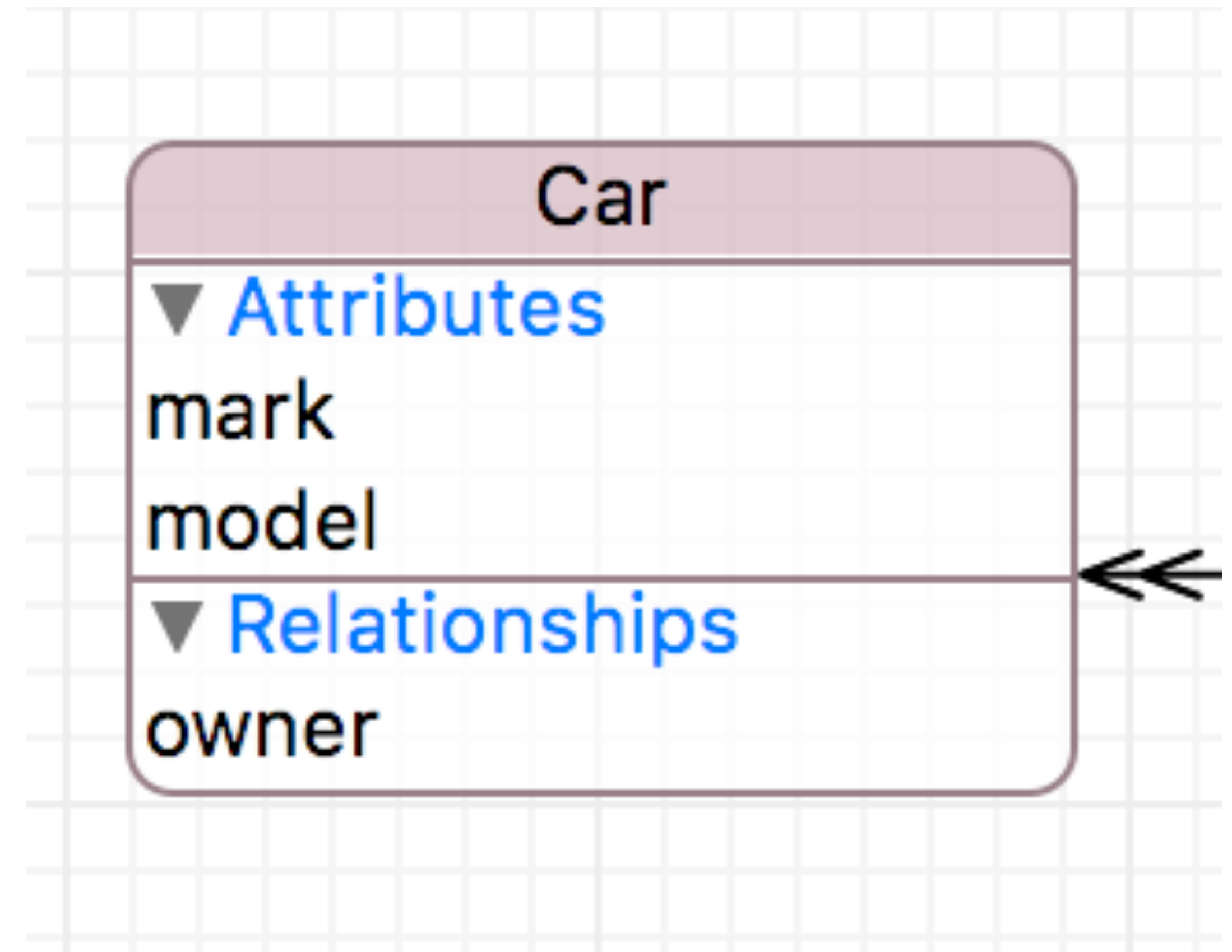
CoreData

Model DEMO

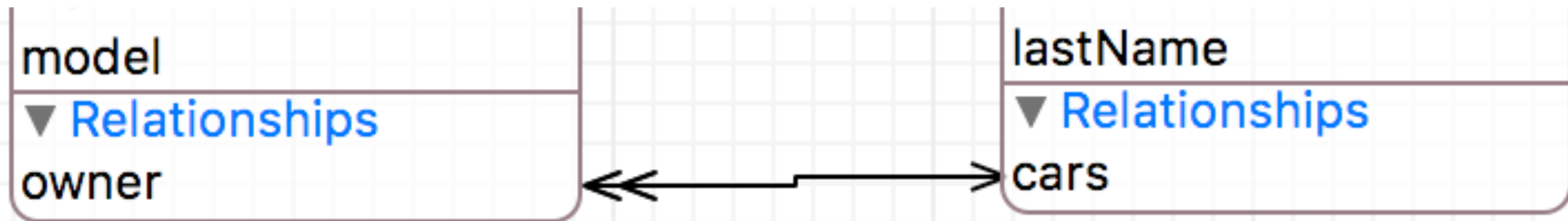


NSManagedObject

- › Отдельный объект (сущность)
- › Хранимые (и transient) свойства
- › Связи
- › ...



NSManagedObject - Relations



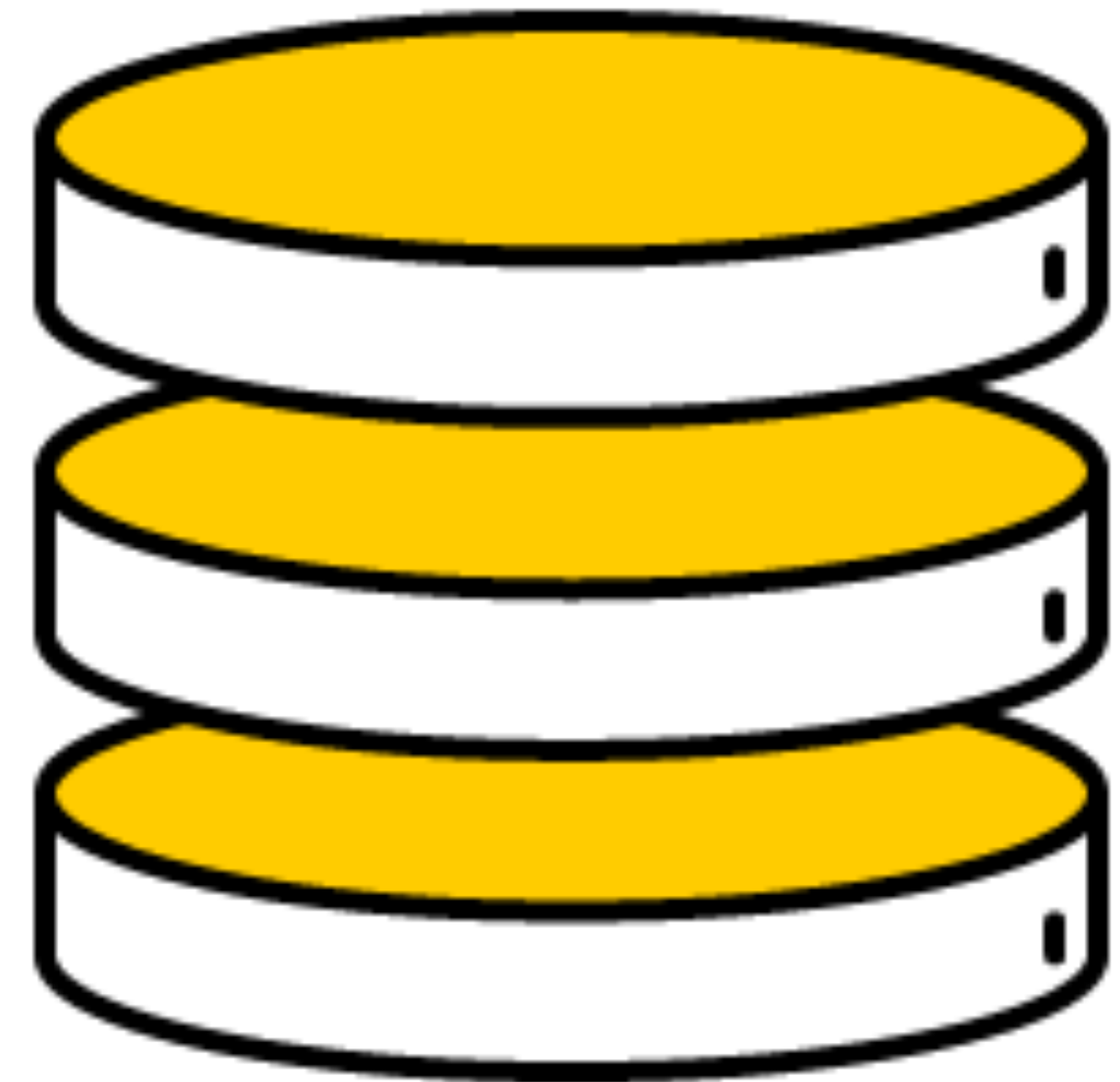
CoreData

Object DEMO



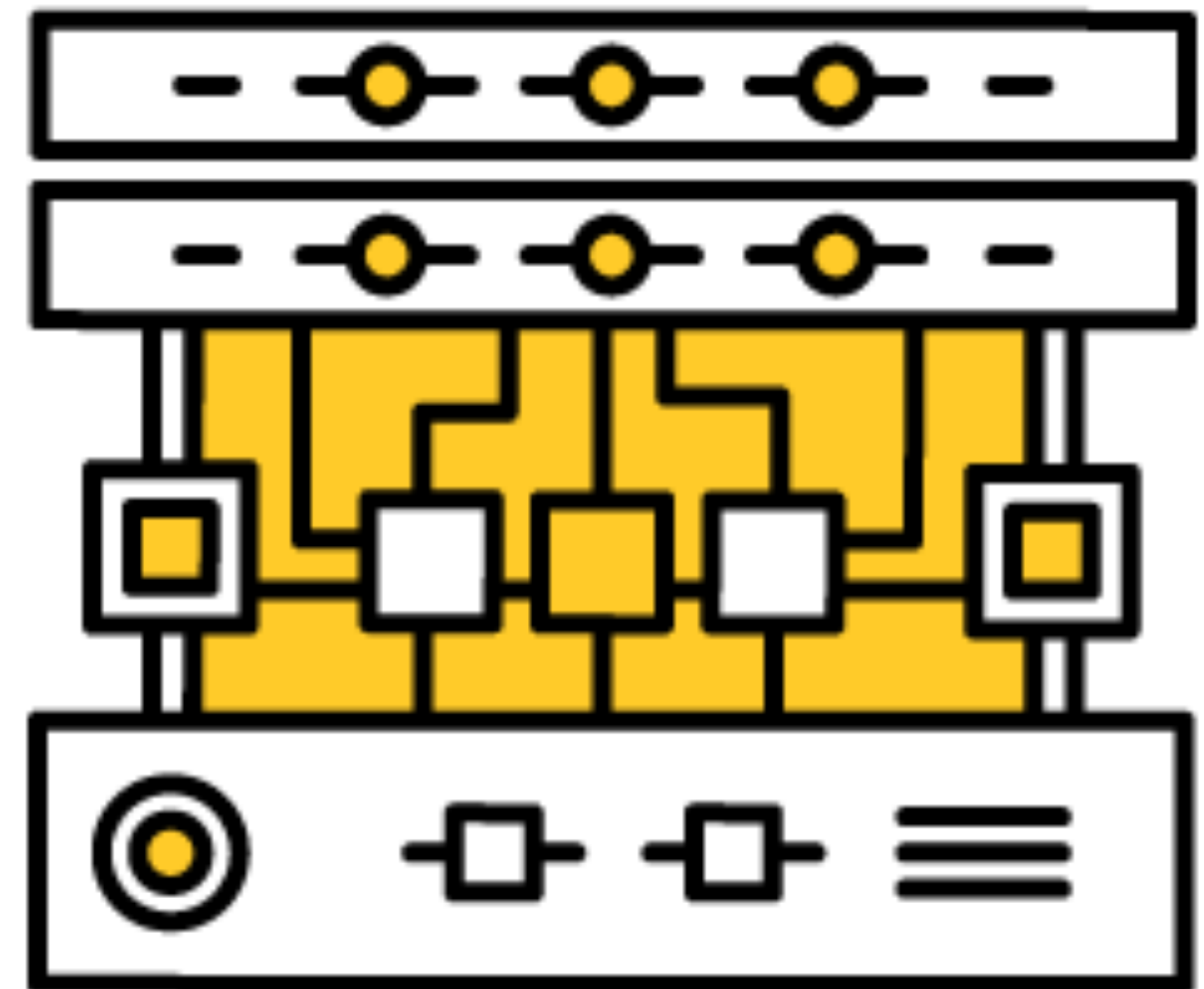
NSPersistenceStore

- › Непосредственно хранит данные
- › Абстракция над хранилищем
- › SQLite, Binary, InMemory, XML



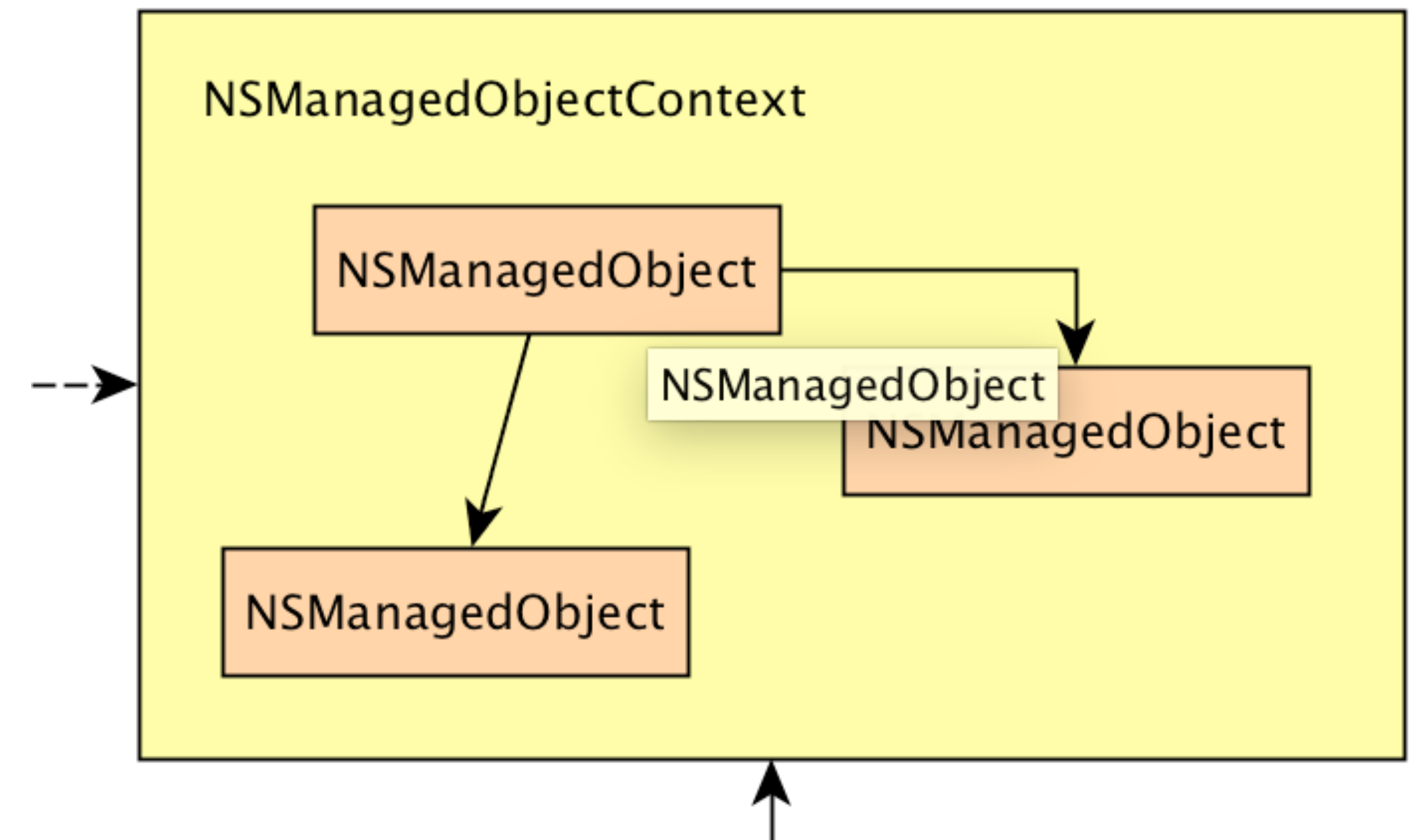
NSPersistentStoreCoordinator

- › Управляет Store'ами
- › Дает интерфейсы доступа к Store'ам для контекстам
- › Участвует в миграциях *

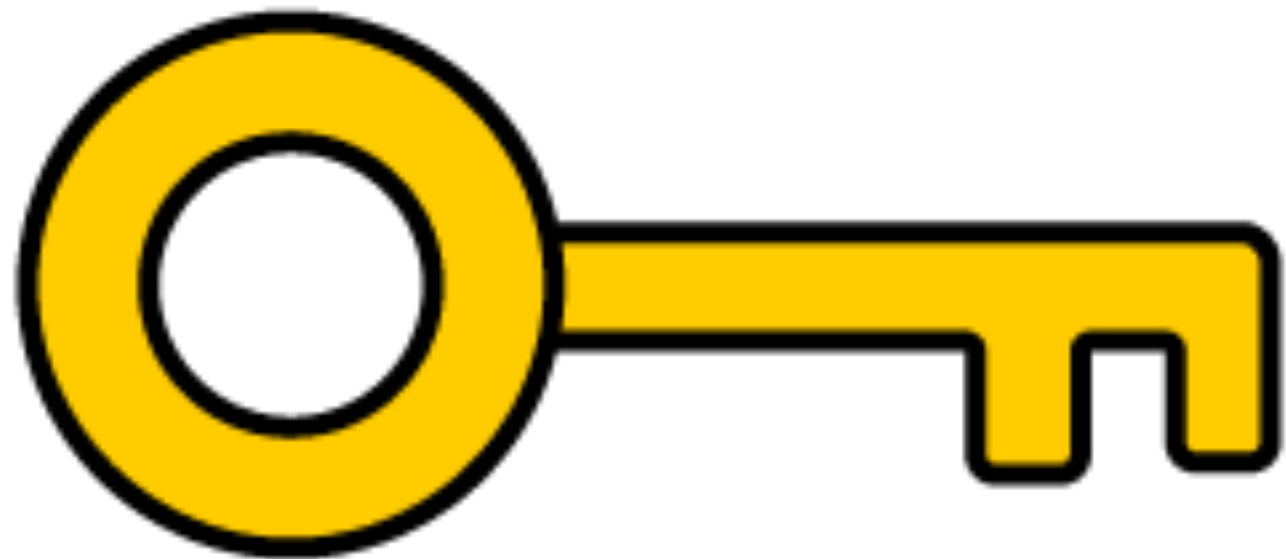


NSManagedObjectContext

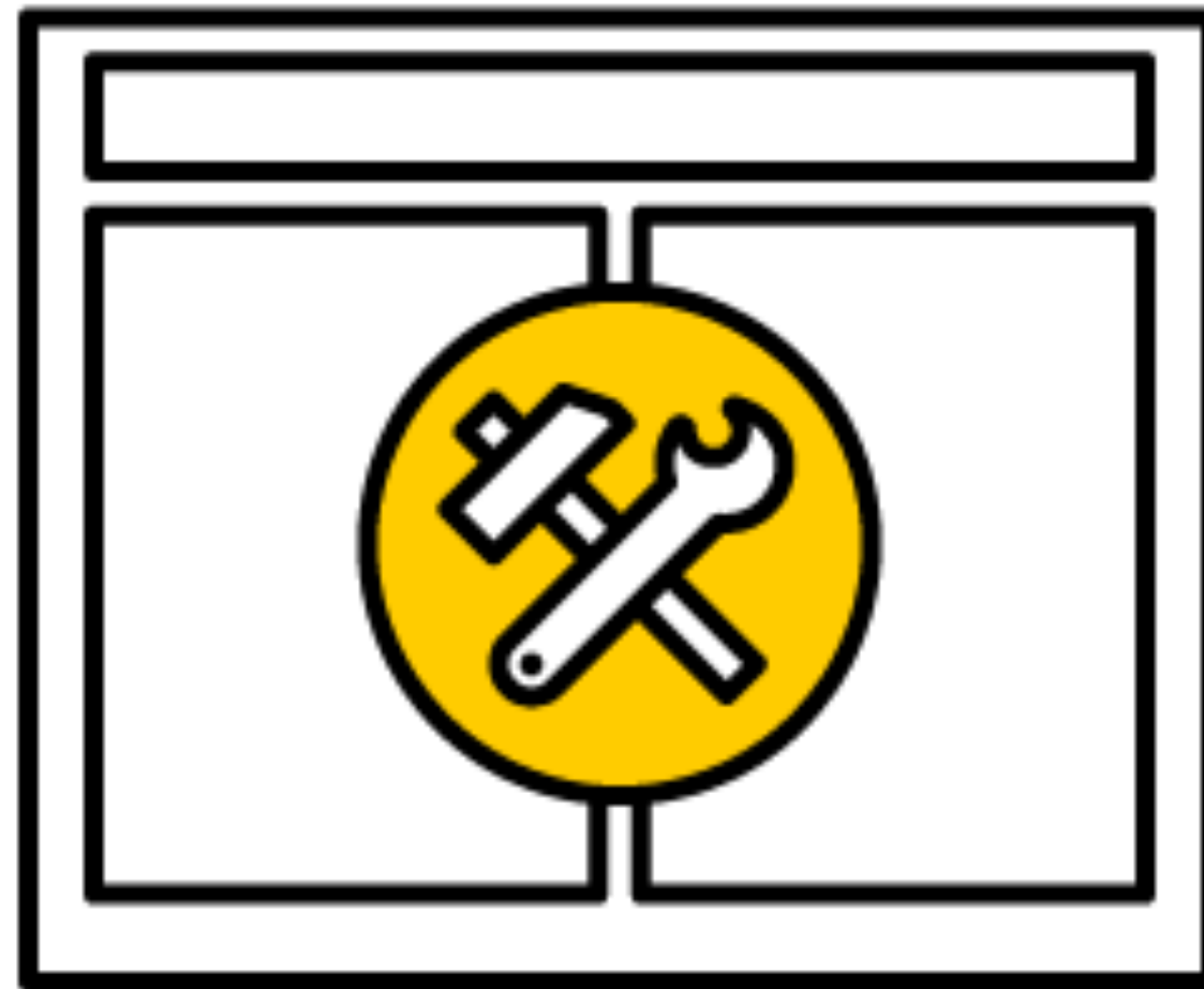
› Хранилище объектов в памяти



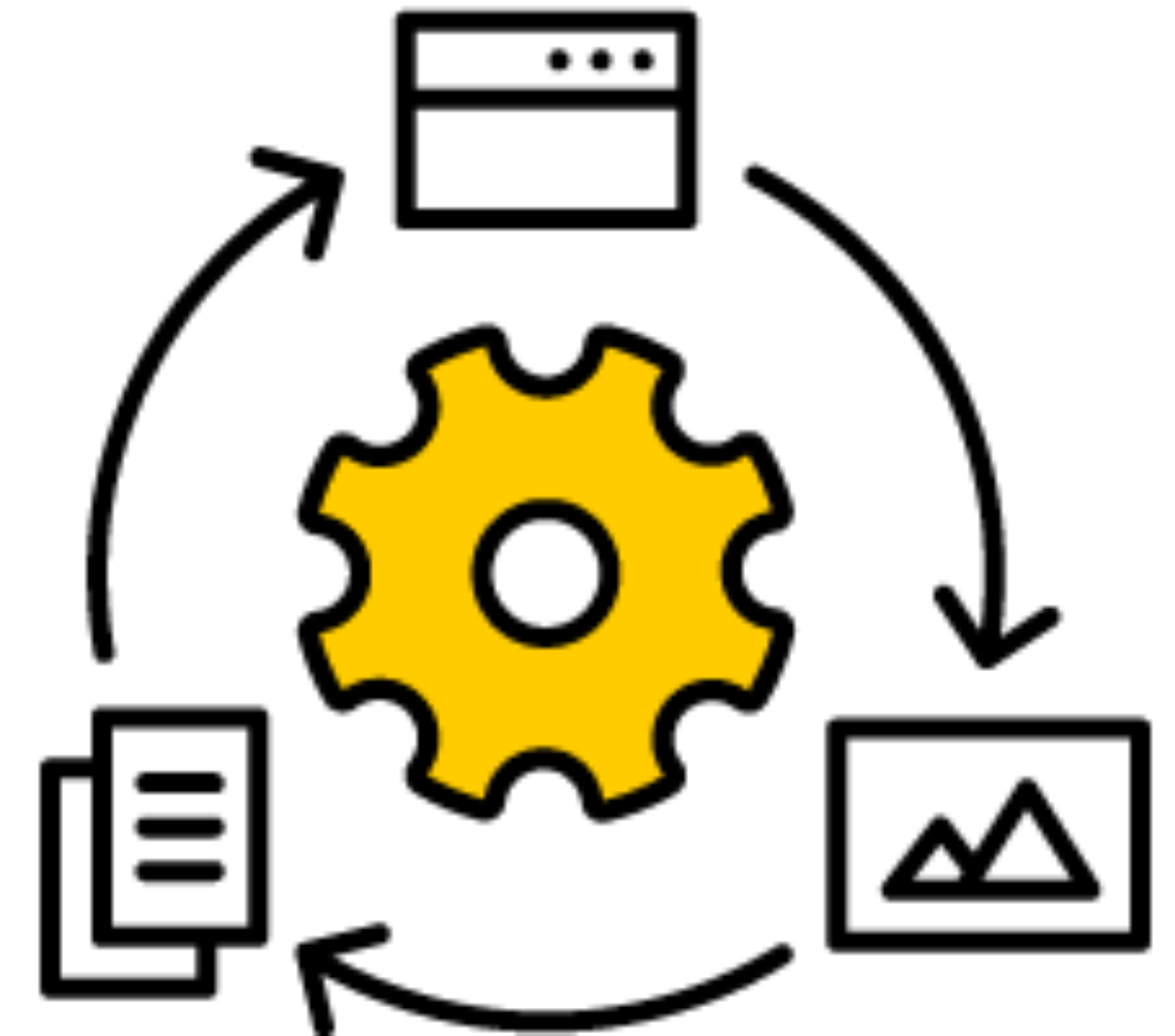
NSManagedObjectContext - Concurrency



Confinement



Main



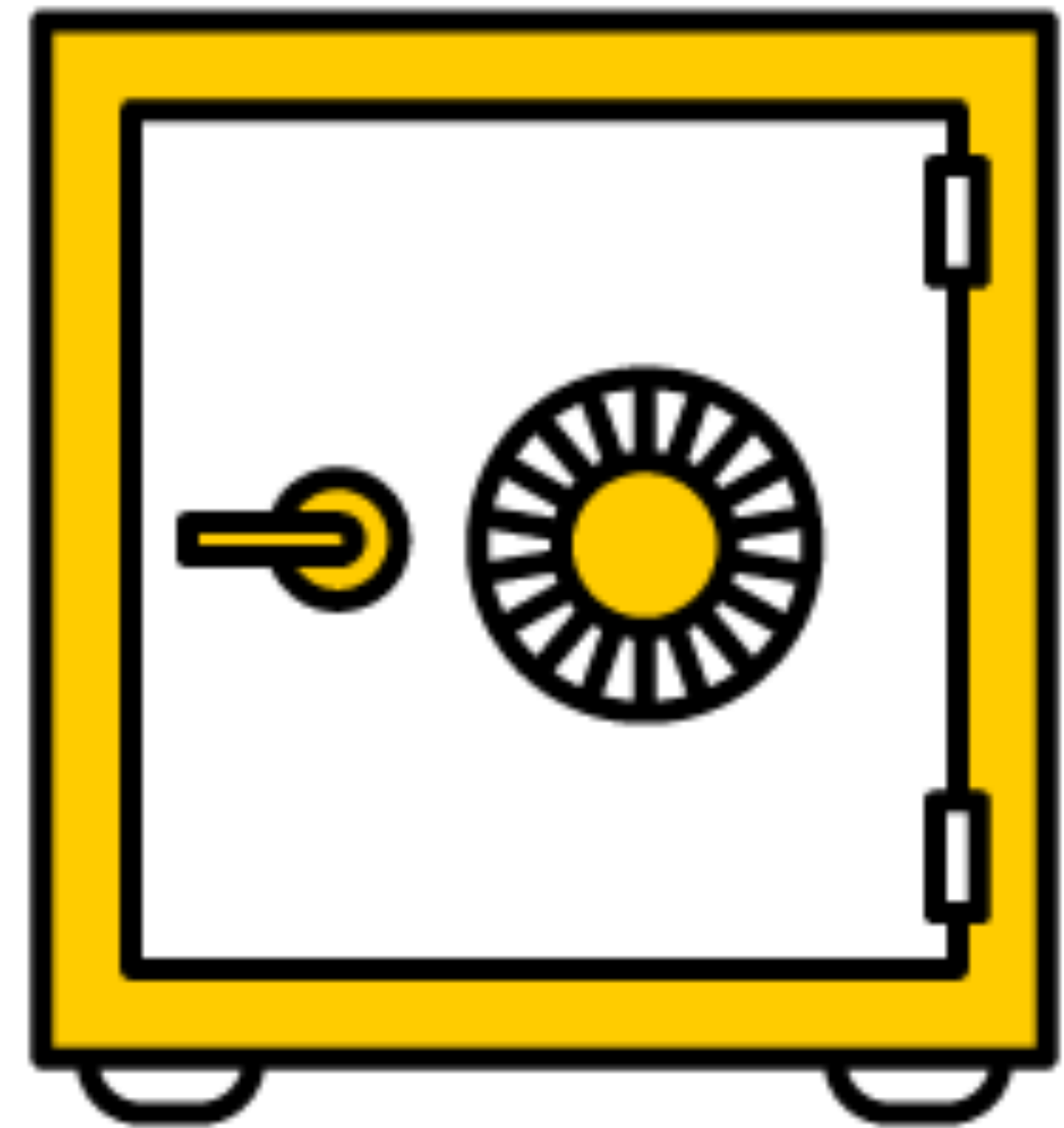
Private

 NSManagedObjectID -
thread safe

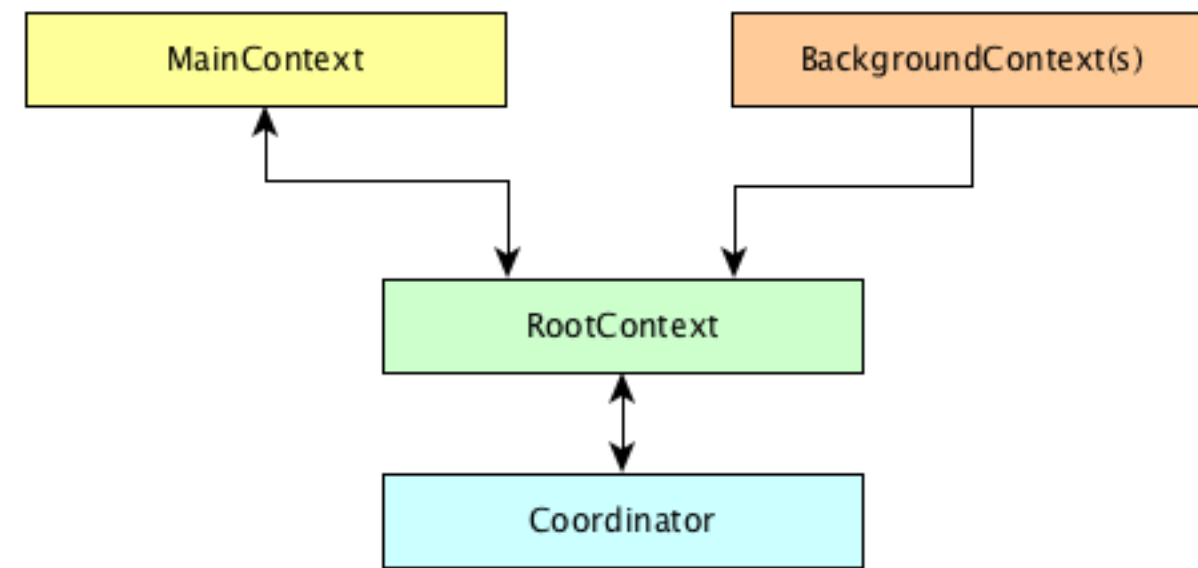
Остальное - нет!

NSPersistentContainer (iOS 10 Only)

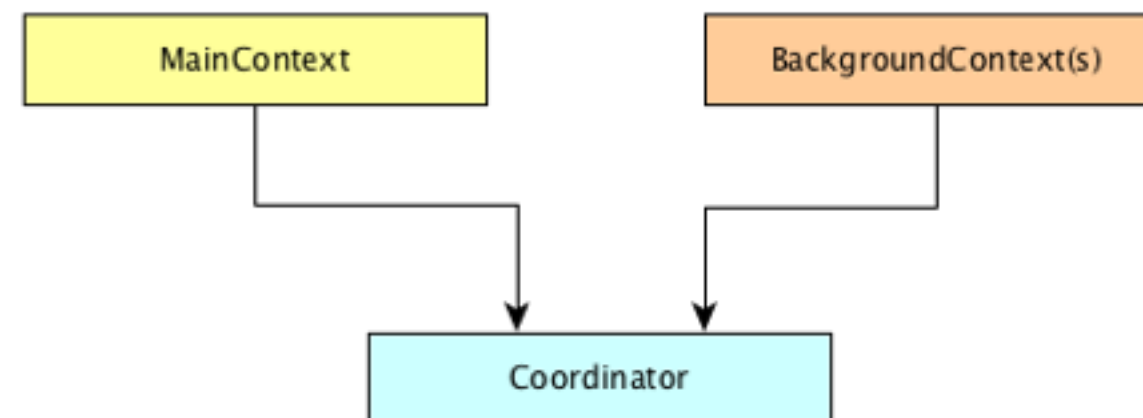
› Собирает в себе все



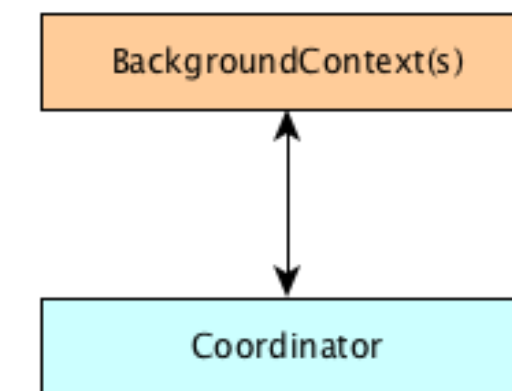
CoreData Stack



Root Context

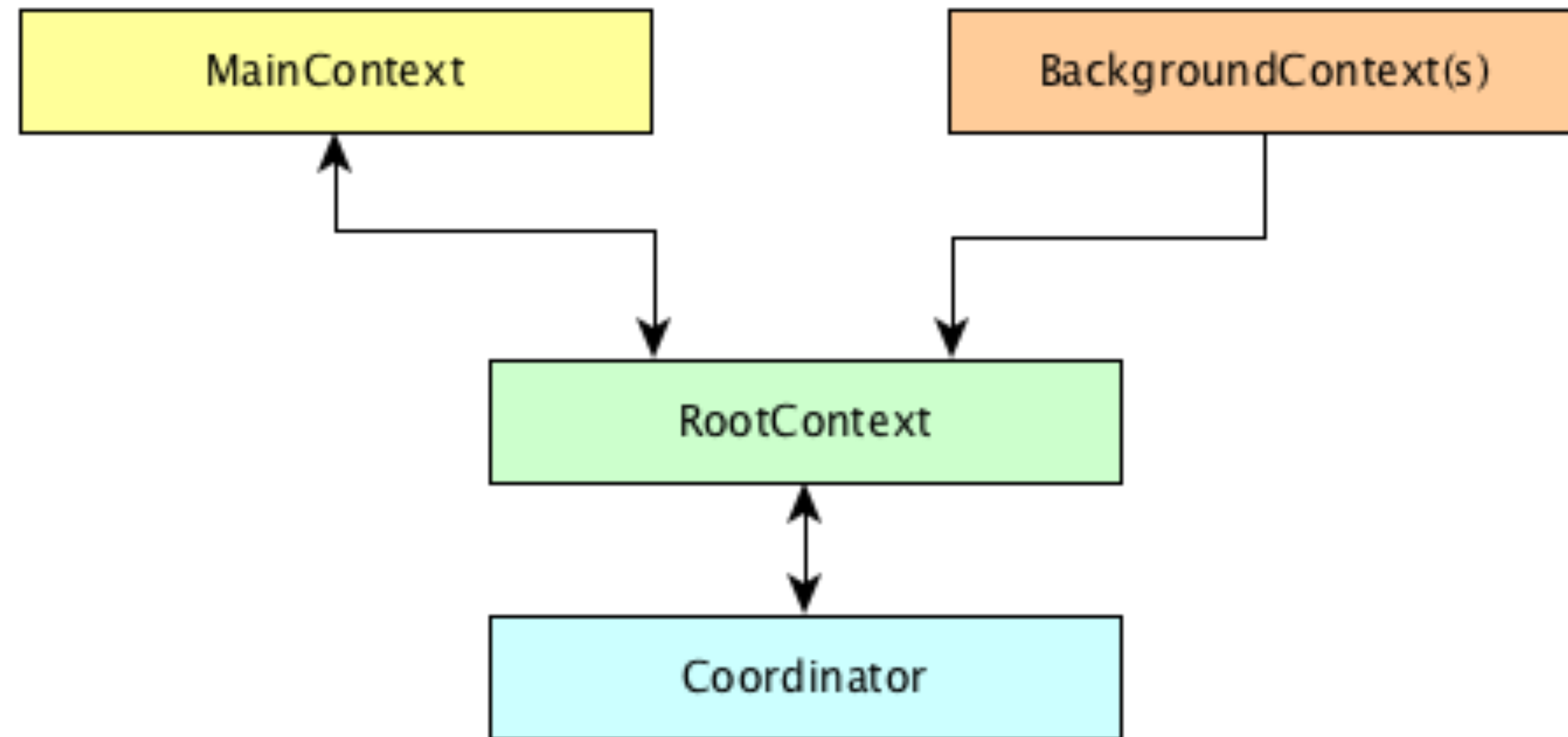


Branch Context

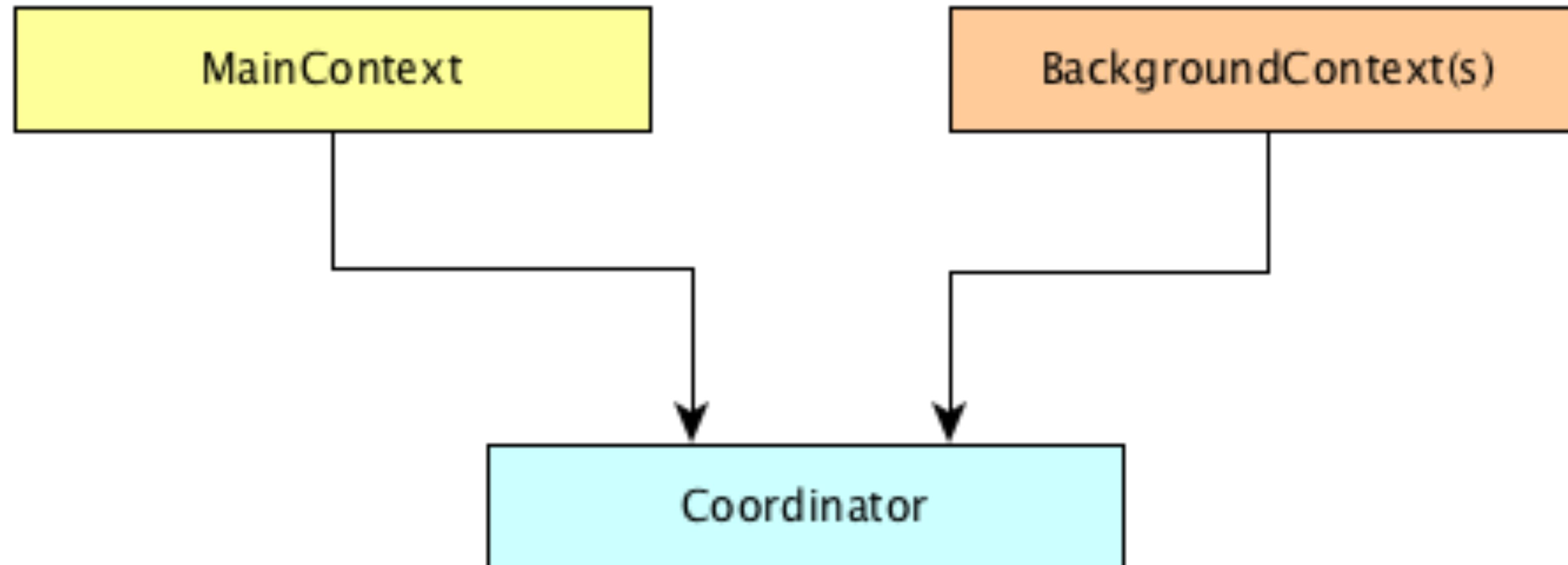


Background Only

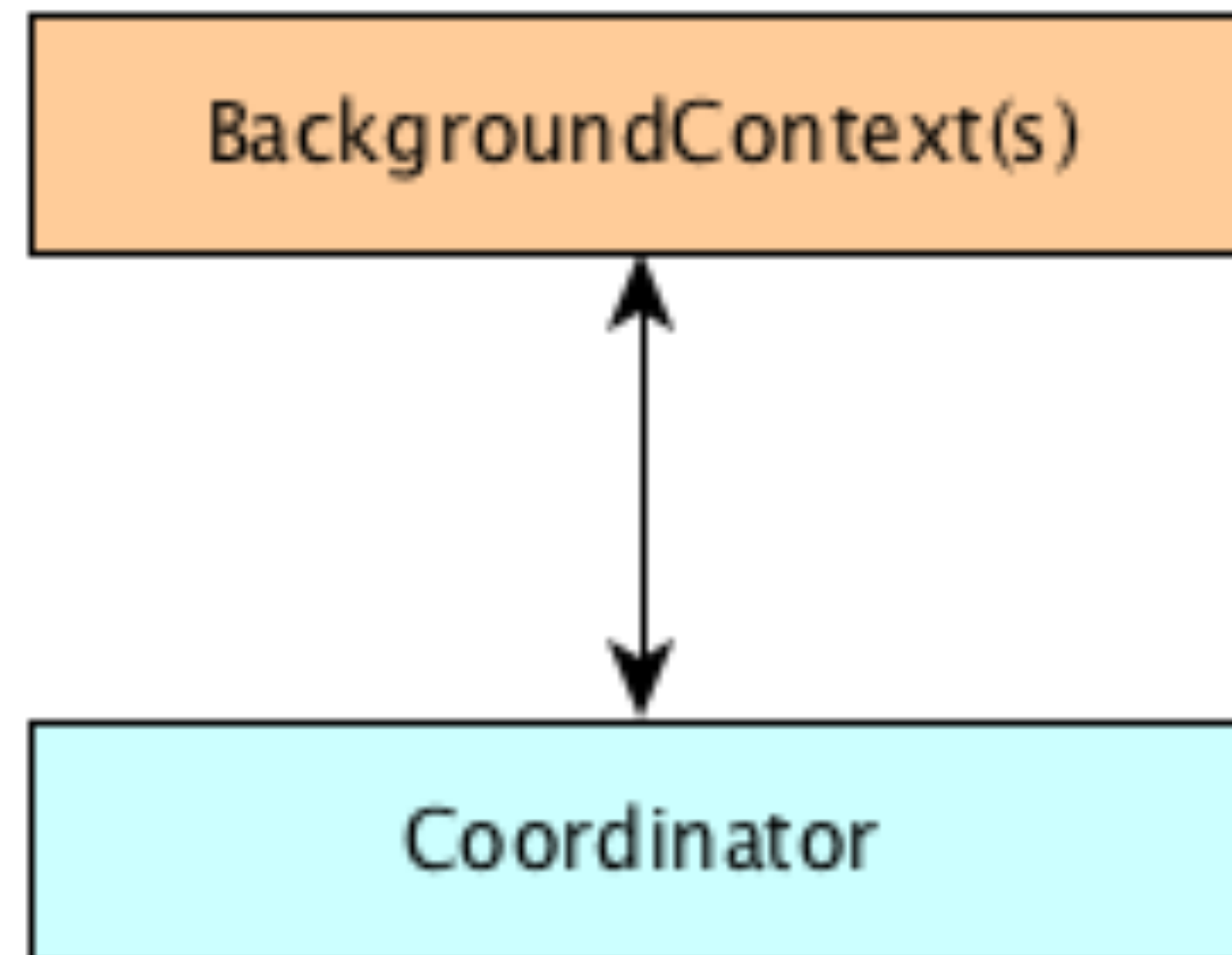
CoreData Stack - RootContext



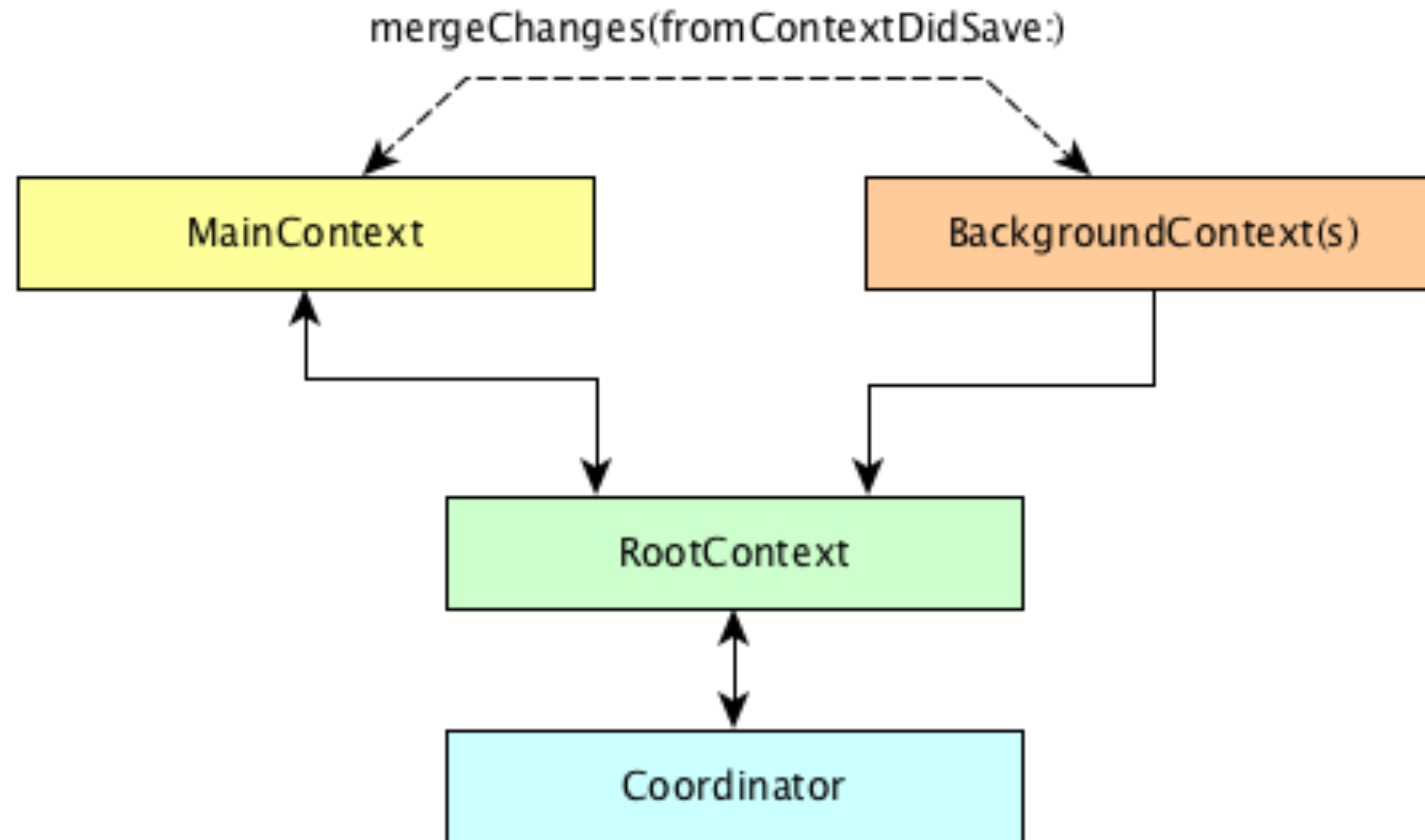
CoreData Stack - Batch Context



CoreData Stack - Background Only



mergeChanges(fromContextDidSave:)

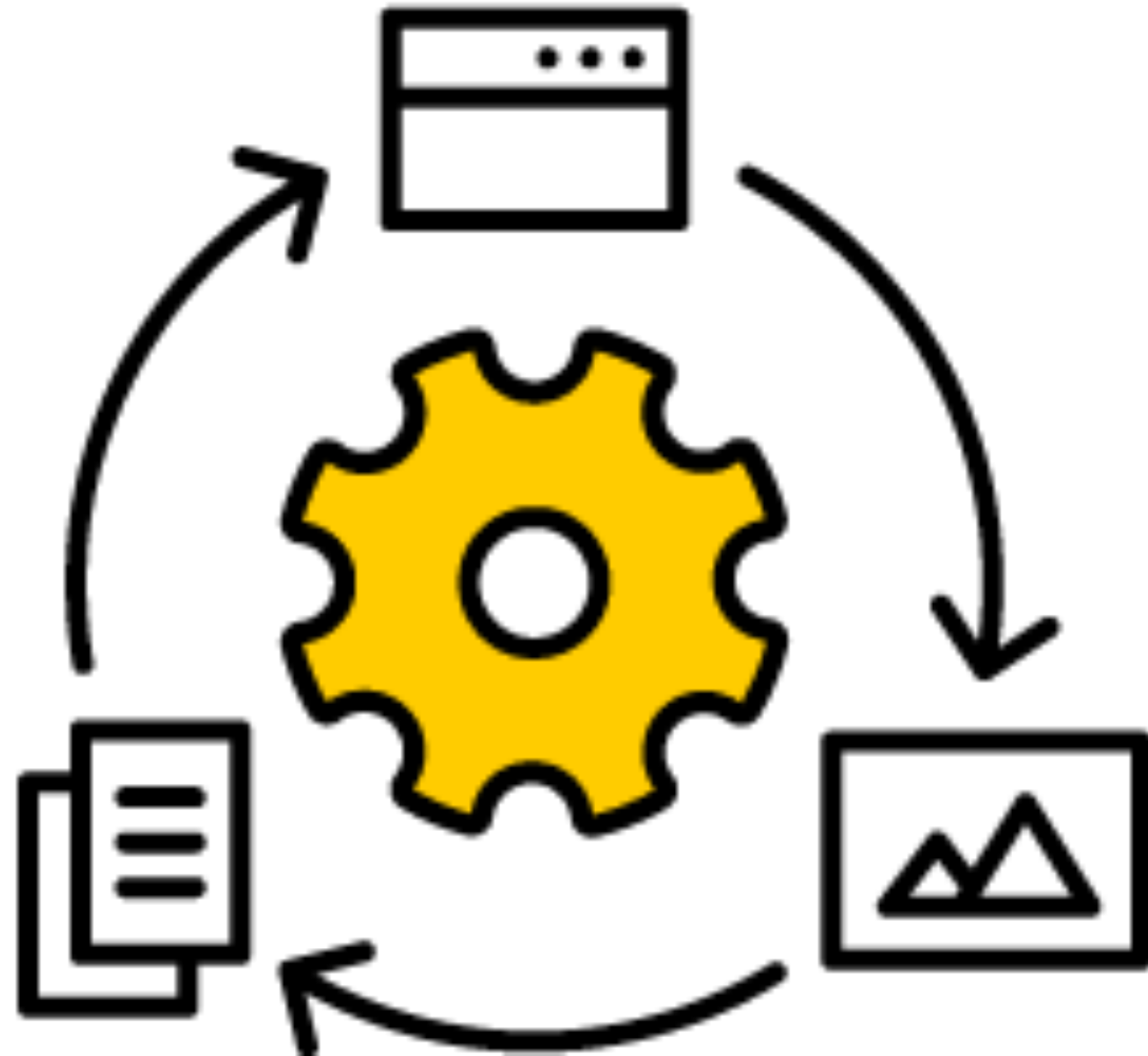


CoreData

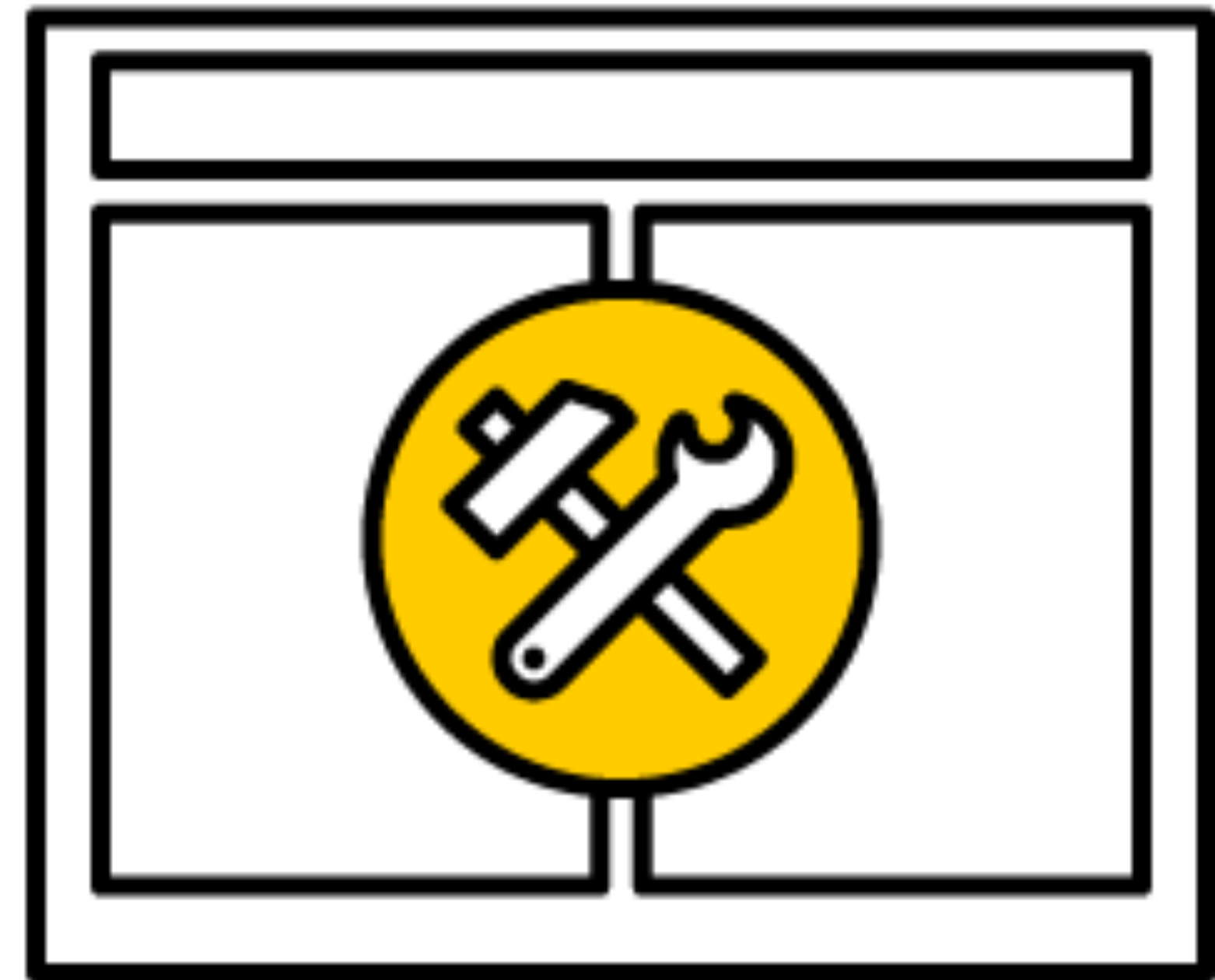
Stack DEMO



CoreData - Queries



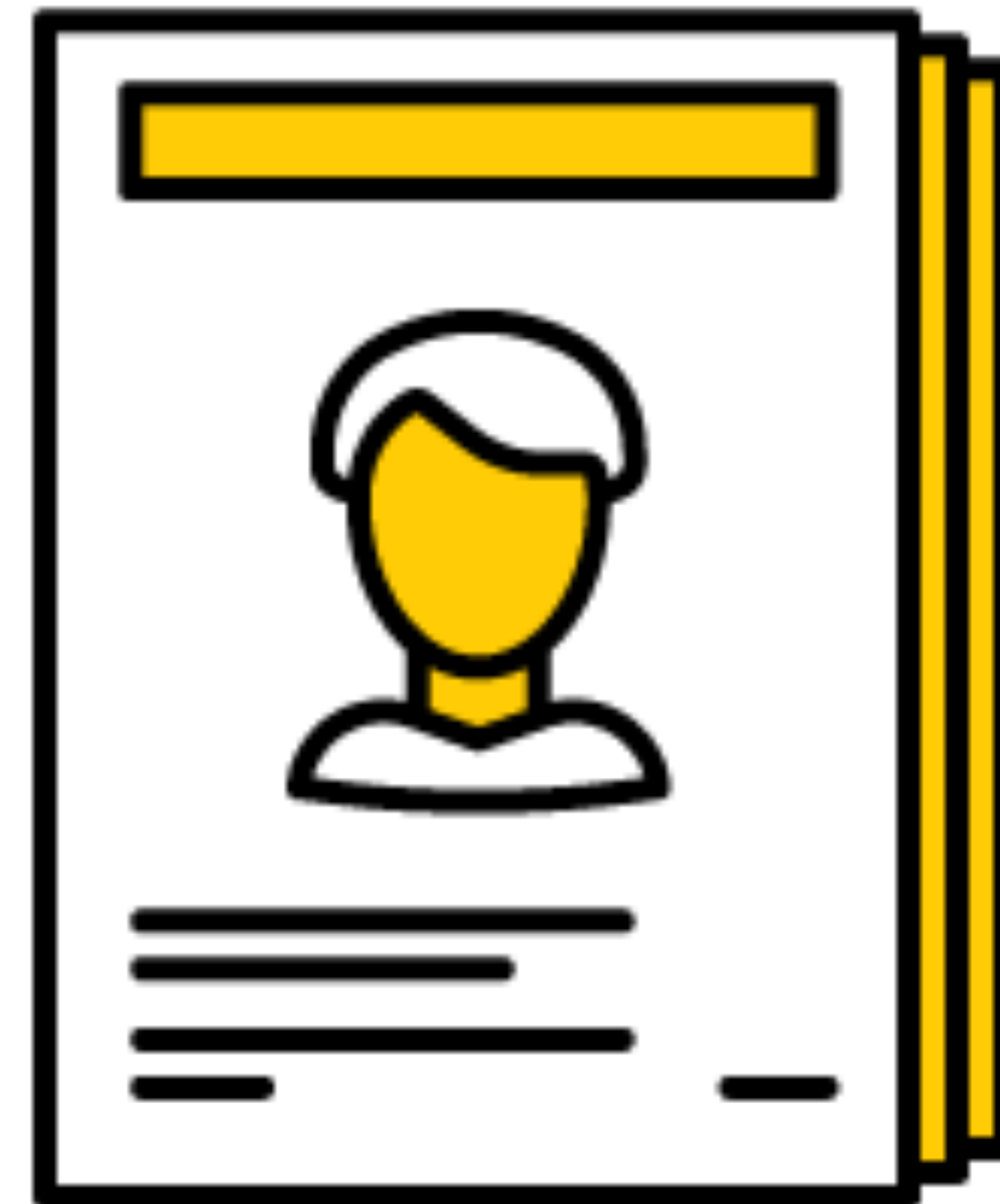
`NSFetchRequest`



`NSFetchResultsController`

NSEntityDescription

- › Описывает сущность
- › Указывает на принадлежность к контексту



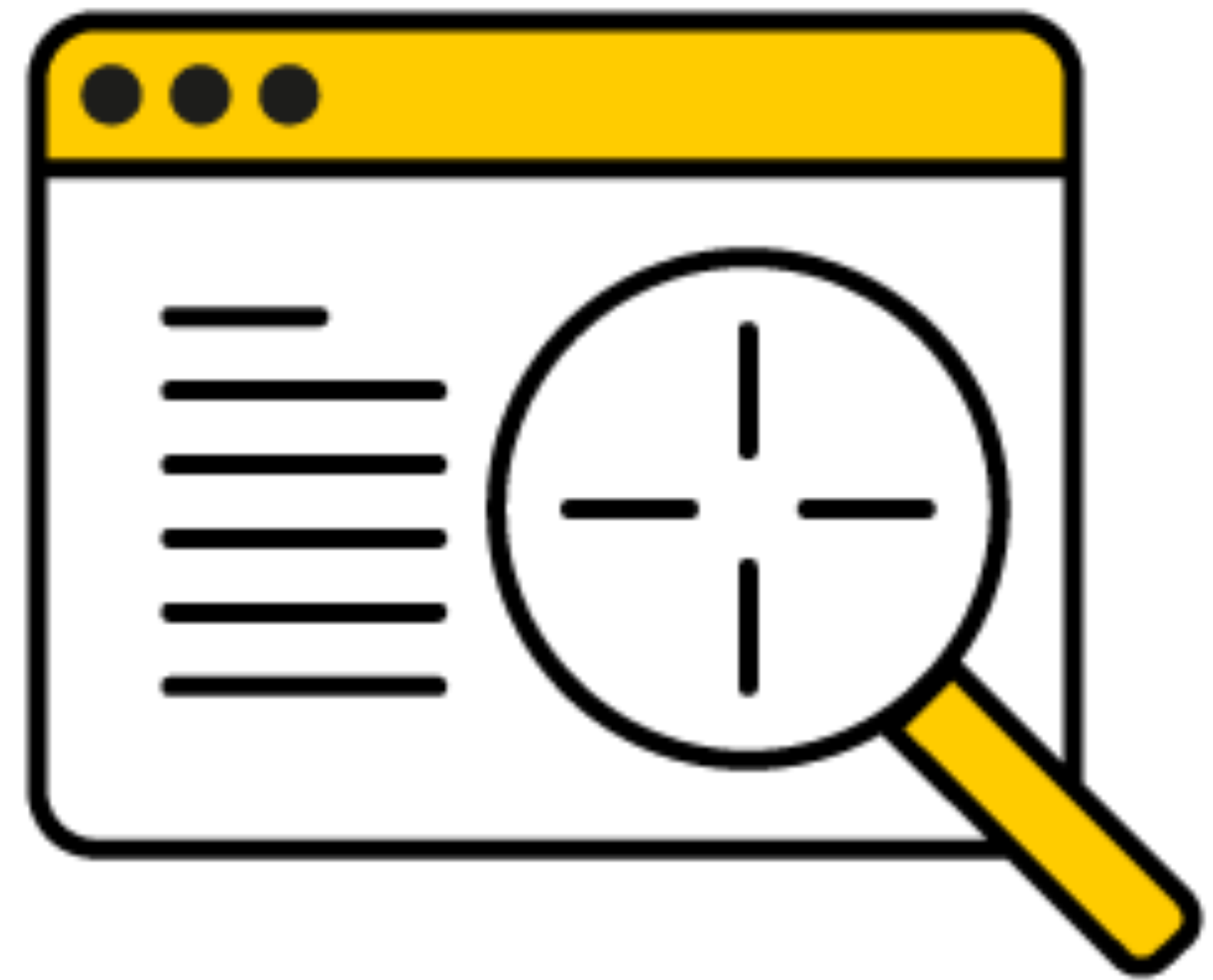
CoreData

Create DEMO



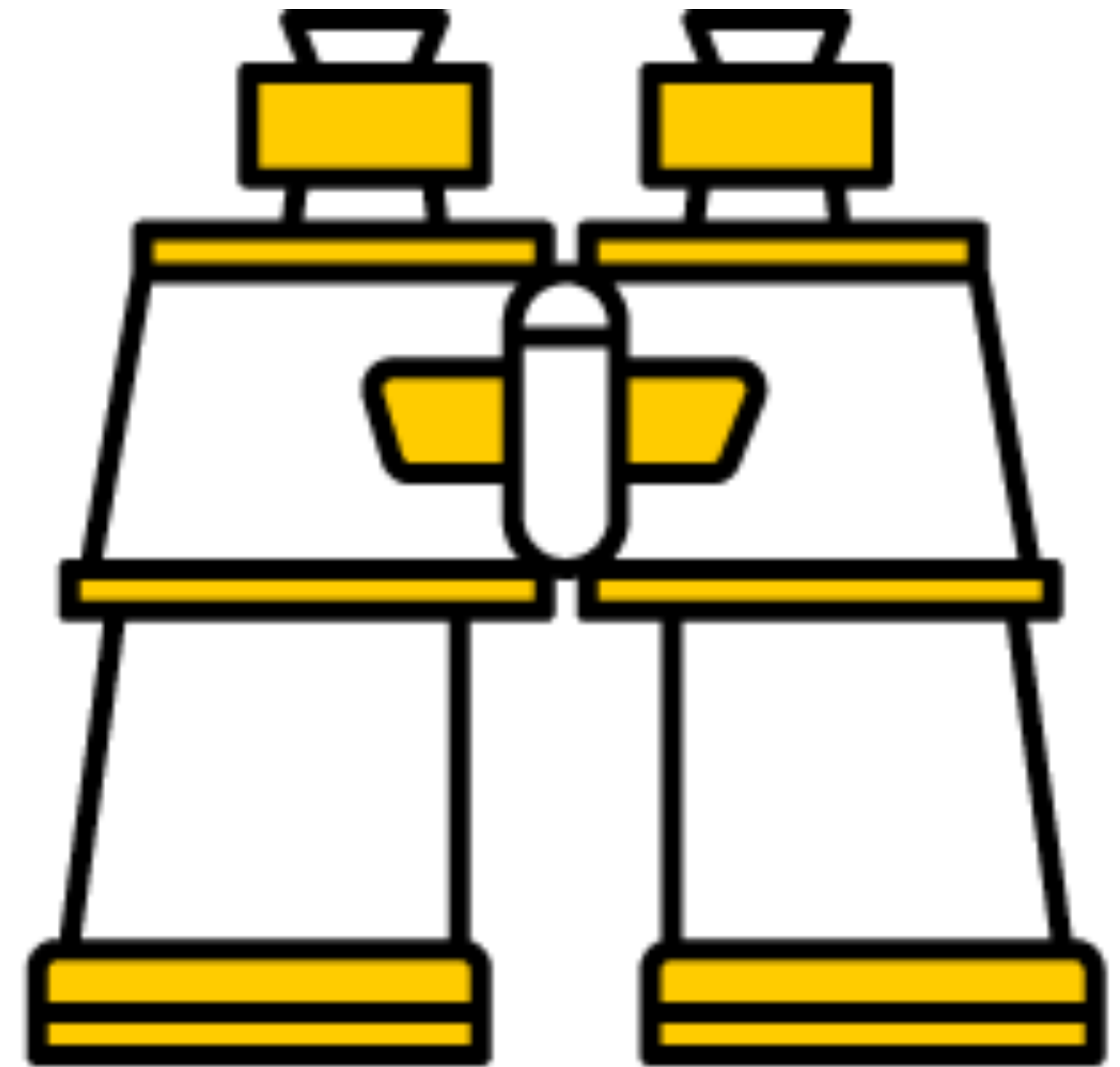
NSURLSession

- › Запрос
- › Предикат поиска
- › Условие сортировки
- › Параметры загрузки



NSPredicate

› Фактически - описание выборки объектов которую вы хотите получить



NSSortDescriptor

› Параметры сортировки



CoreData

Request DEMO



NSFetchResultsController

- › Можно использовать как DataSource для UITableView (например)
- › Использует NSFetchRequest
- › Надо NSManagedObjectContext - main

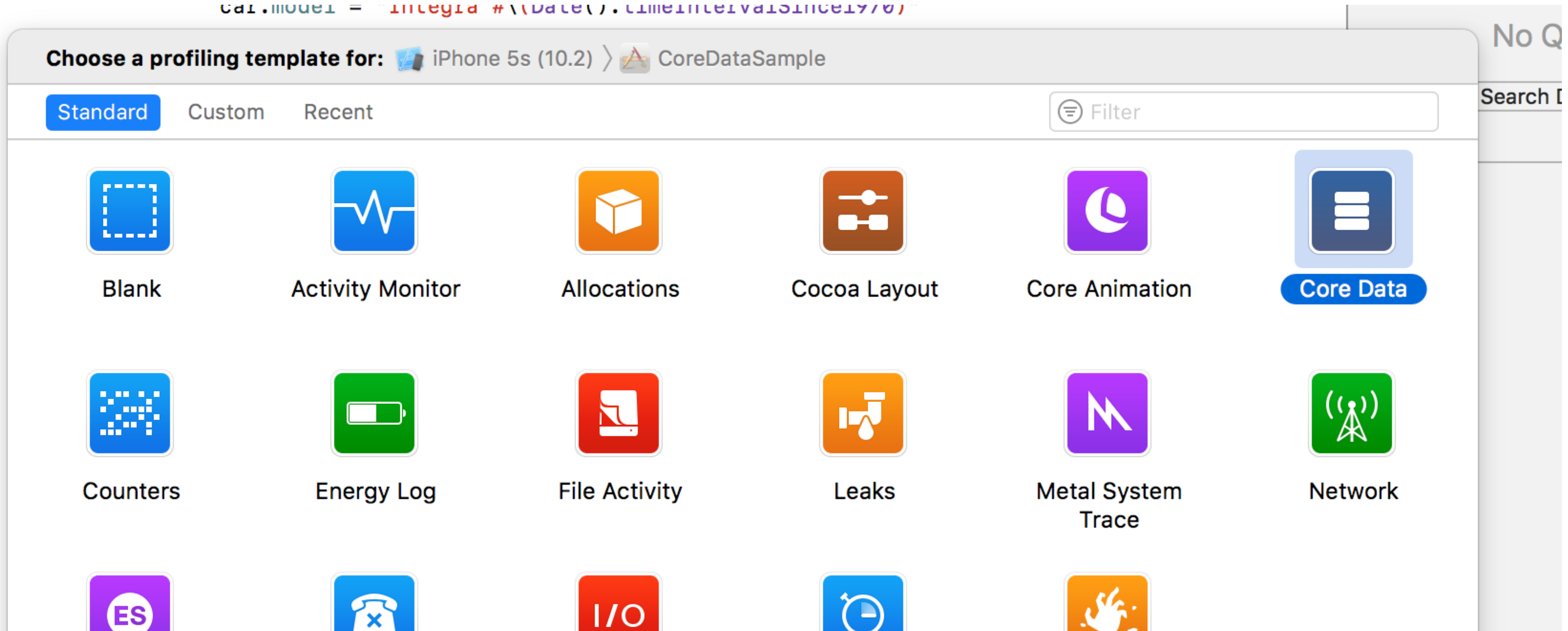


NSManagedObjectModel - Versioning

- › Можно эволюционировать модель через версии
- › Миграции (+Lightweight)

VERSION 01
VERSION 02
VERSION 03
VERSION 04
VERSION 05
VERSION 06

CoreData - Instruments



Вопросы?



Задача



Общие требования

- › Необходимо реализовать локальный кэш на базе CoreData
- › ~~Необходимо продемонстрировать отладку производительности CoreData через Instruments *~~
- › Реализовать работу без использования NSManagedObjectContext с main concurrency type *
- › Необходимо реализовать несколько версий модели и миграции **

Базовая задача

- › Приложение (через операции) должно уметь использовать локальный кэш через Core Data (сохранение заметки и списка, получение конкретной заметки, изменение конкретной заметки)
- › Можно использовать любой способ построения стэка Core Data (но не PersistentContainer)

Усложнение 1 (*)

- › ~~Необходимо поработать с Instruments для профайлинга CoreData~~ на данный момент этот инструмент для iOS не работает (в Xcode 8.x)
- › Реализовать работу со стэком CoreData только на background context'ax

Усложнение 2 (**)

- › Реализовать несколько версий модели и миграции между ними.
- › Миграции не должны быть Lightweight

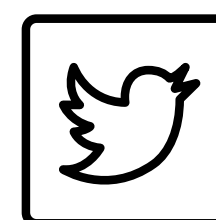
Контакты

Малых Денис

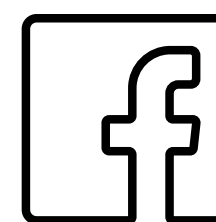
iOS Developer



mrdekk@yandex.ru



@mrdekk



mrdekk

Спасибо за внимание

