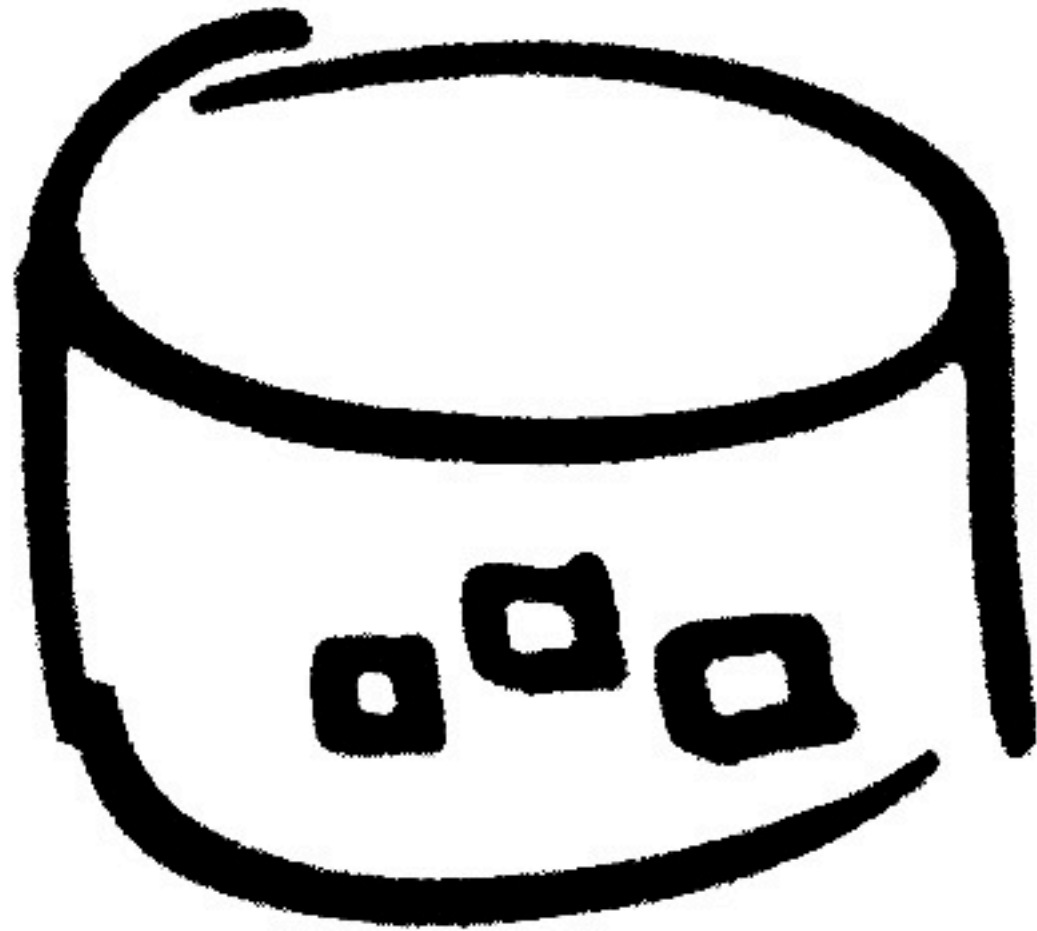


Craig Federighi



Senior Vice President
Software Engineering



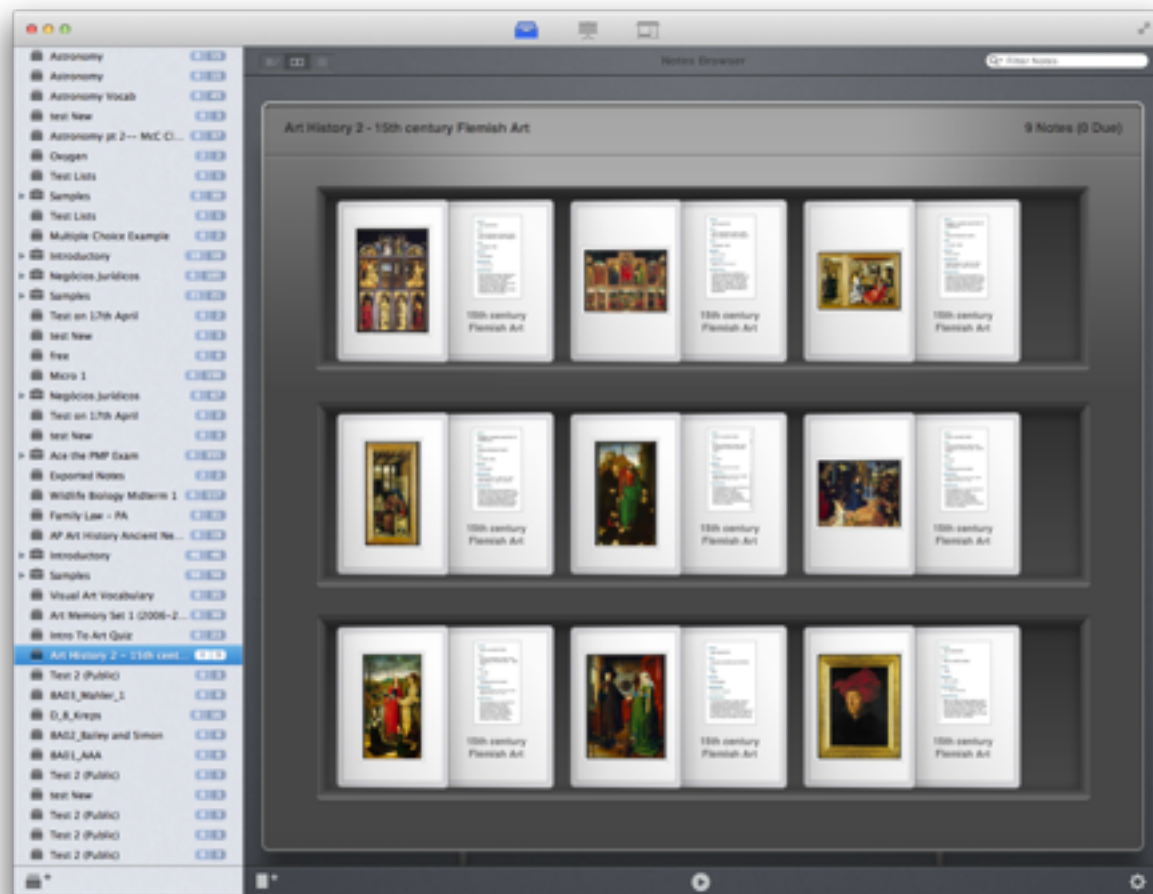


NSManagedObjectContext



NSManagedObjectContexts

Mental about Sync



Mental Case

Why is there no cloud sync?

It's ridiculous that I have to connect to the
same Wi-Fi network

What year is this?



The Mental Blog

SOFTWARE WITH INTELLECT

May 16, 2012

4 notes

Under the Sheets with iCloud and Core Data: The Basics

There can be no doubt that iCloud is an important part of Apple's future. It's now less than a year old, but it has been a mixed start. The service itself seems up to the enormous data traffic it has to push, but some aspects of client-side integration are still very rough around the edges.

You don't have to look far in Apple's developer forums to know that Core Data syncing is one of those rough spots. An appeal on Twitter, and extensive Googling, led me to just one shipping app that currently offers Core Data syncing over iCloud: Time Butler. I know of no other shipping app that is using it, and certainly no apps from well-known, established developers.

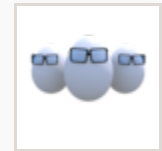
(All eyes are currently on Bare Bones, who have vowed to offer iCloud syncing in their shoebox app Yojimbo before MobileMe syncing is turned off in a few months. Time will tell if they achieve that goal.)

Mental Case

I've spent the last 3-4 months integrating Core Data syncing over iCloud into Mental Case for Mac. It is not in the App Store, but you can find it at mentalfaculty.tumblr.com with a limited audience.

To say it has been a challenge would be an understatement — it has probably

About the author



Drew McCormack
(@drewmccormack) is
founder of The Mental Faculty,
developer of Mental Case

	Search
----------------------	--------

More...

Archive
RSS

The Pre-Ubiquitous Era

To stop worrying about it will require worrying about it a lot at first.

Ken Arnold

Cultured Code

Bare Bones

Black Pixel

Synchronization Architectures

PHP

Java

Rails

Django

Node.js

Google App Engine*

...

Smart Server

Complete Web App

Complete Web App

Pros

Central truth

Cross platform

No lock in

Cons

Expensive

Duplicated effort

Parse

Azure Mobile Services

Dropbox Datastore

Wasabi Sync

Simperium

Helios

RESTful Store

Structured Store

Structured Store

Pros

Central truth

Simple server

No duplicated effort

Cons

Expensive

Vendor lock in

Core Data integration?

iCloud

Dropbox

OmniPresence

Google Drive

Amazon Cloud Drive

Microsoft SkyDrive

BitTorrent Sync

File Store

File Syncing Service

File Syncing Service

Pros

Free

No lock in

No server code

Cons

No central truth

Risk of divergence

No well established framework

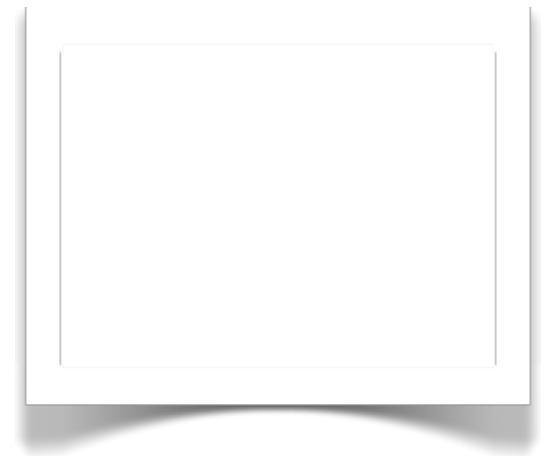
Syncing Databases via Files



Database



Changes
Deltas
Transactions



```
{  
  "entity": "Movie",  
  "globalid": "Movie3c075471aad0",  
  "type": "insert",  
  "properties": {  
    "title": "Twister",  
    "cast": ["Actor4543937534", "Actor9845976497"]  
  }  
}
```



iCloud—Core Data Sync

Problems with iCloud—Core Data

Buggy

Black box

Difficult to test

Yearly release schedule

Lock in



TICDS

(Tim Isted Core Data Sync)

TICDS Background

Open Source (MIT)

First developed by Tim Isted

Maintained by No Thirst Software

Support for Dropbox and iCloud

Shipping in Mental Case and MoneyWell

Not All Roses

Creator is no longer active

Original design/code around 5 years old

Uploads copy of store for each device

Can be slow

Worst: No guarantee stores stay in sync

Where do we go from here?

Core Plot



Introducing Ensembles

Core Data Ensembles

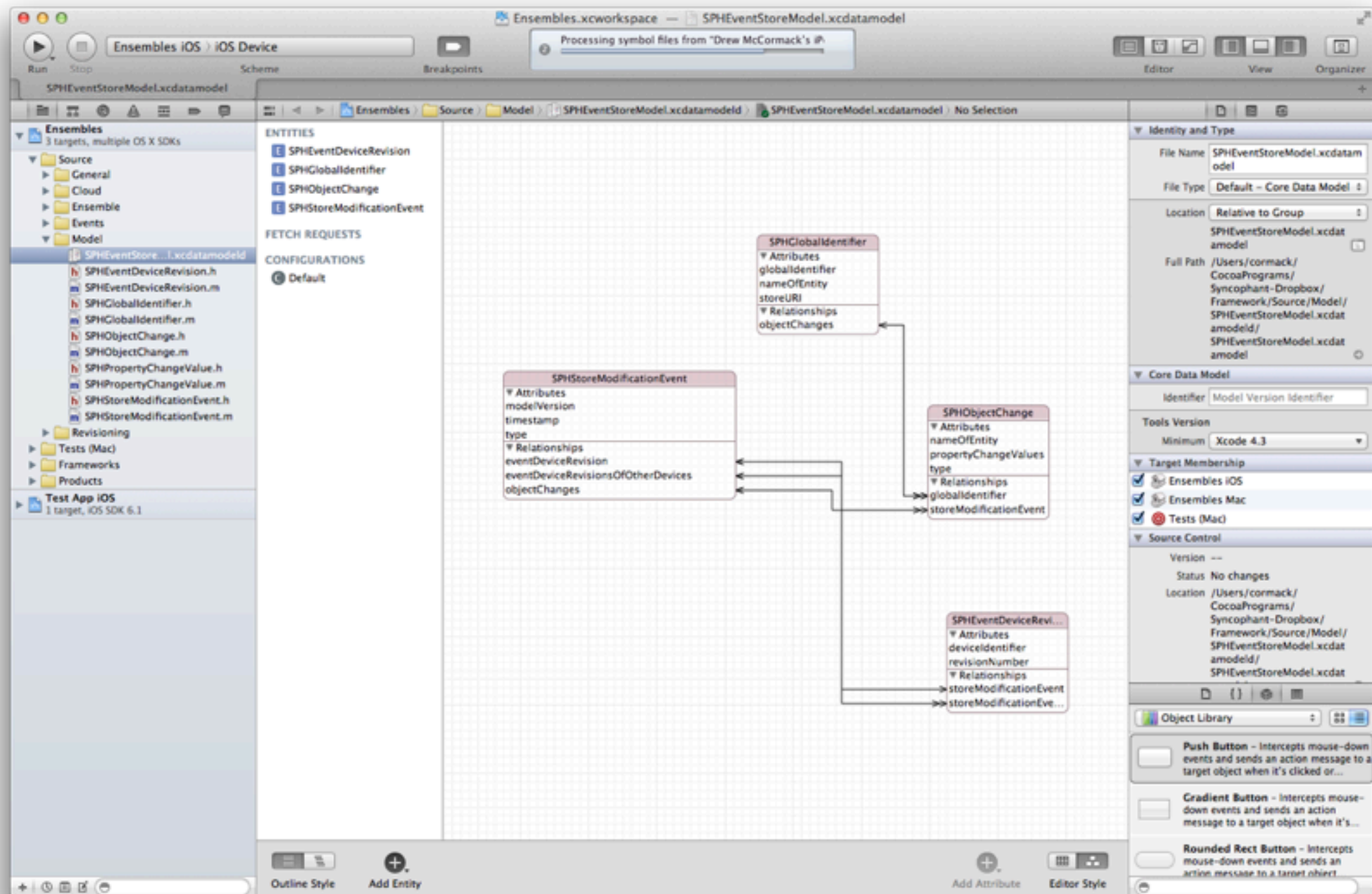
Open source framework

Commercial Licenses?

Syncs Core Data SQLite stores

Mac and iOS

New Concept: *Persistent Store Ensemble*



Non-Invasive



Backend Agnostic



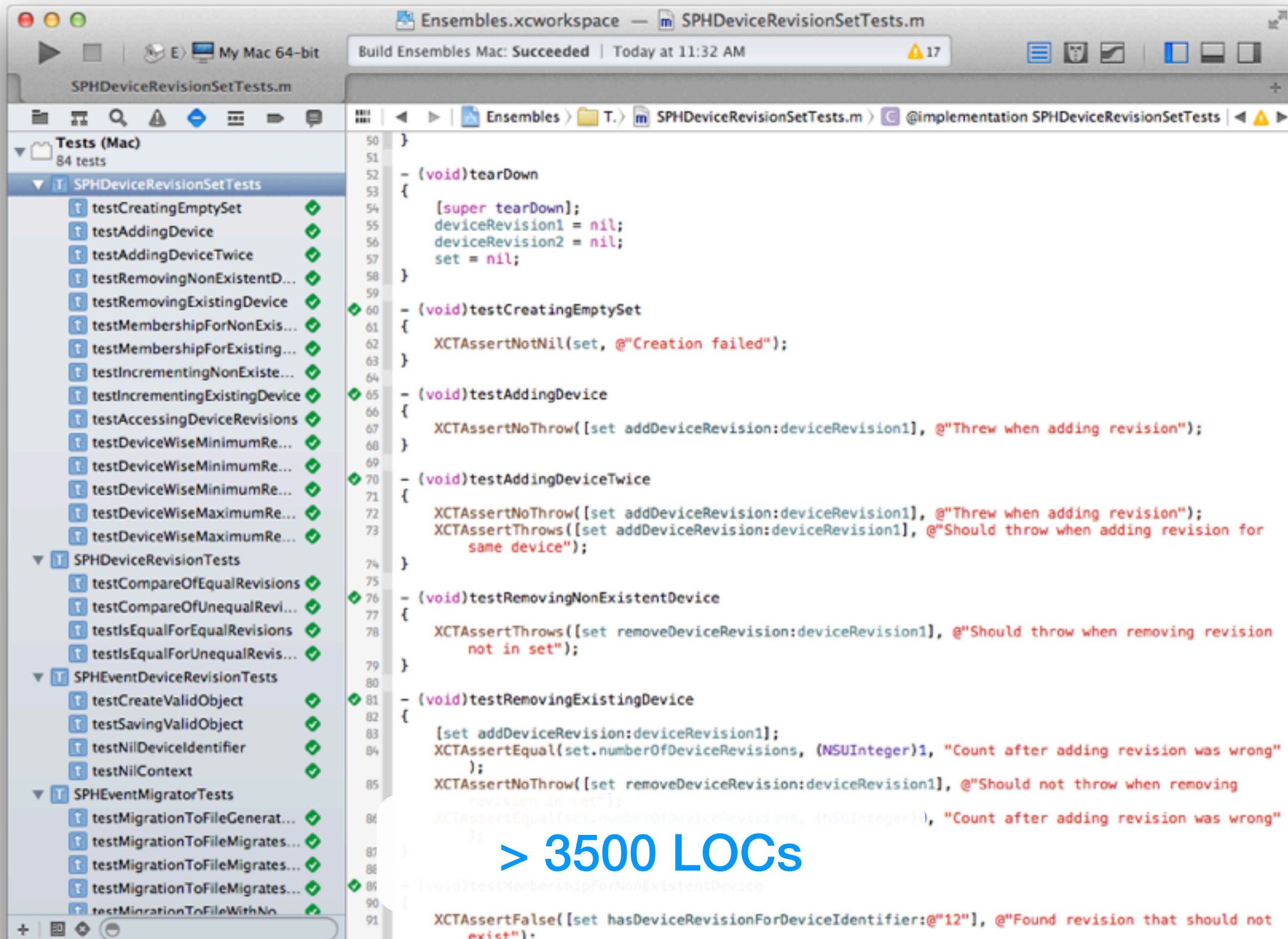
Testable

```
@protocol CDECloudFileSystem <NSObject>
```

- (void)fileExistsAtPath:(NSString *)path
completion:(void(^)(BOOL exists, BOOL isDir, NSError *error))b;
- (void)contentsOfDirectoryAtPath:(NSString *)path
completion:(void(^)(NSArray *contents, NSError *error))b;
- (void)createDirectoryAtPath:(NSString *)path
completion:(SPHCompletionBlock)block;
- (void)uploadLocalFile:(NSString *)fromPath
toPath:(NSString *)toPath
completion:(CDECompletionBlock)block;

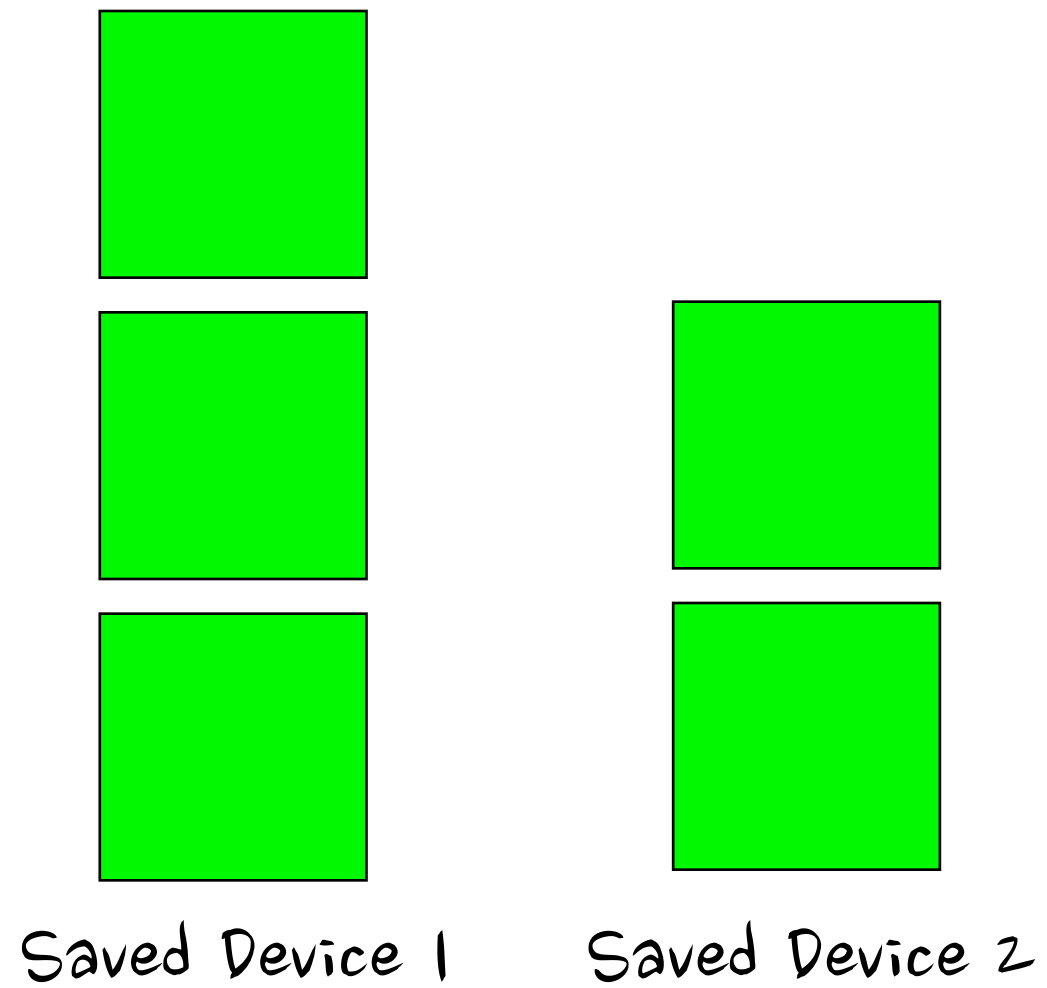
```
...
```

```
@end
```



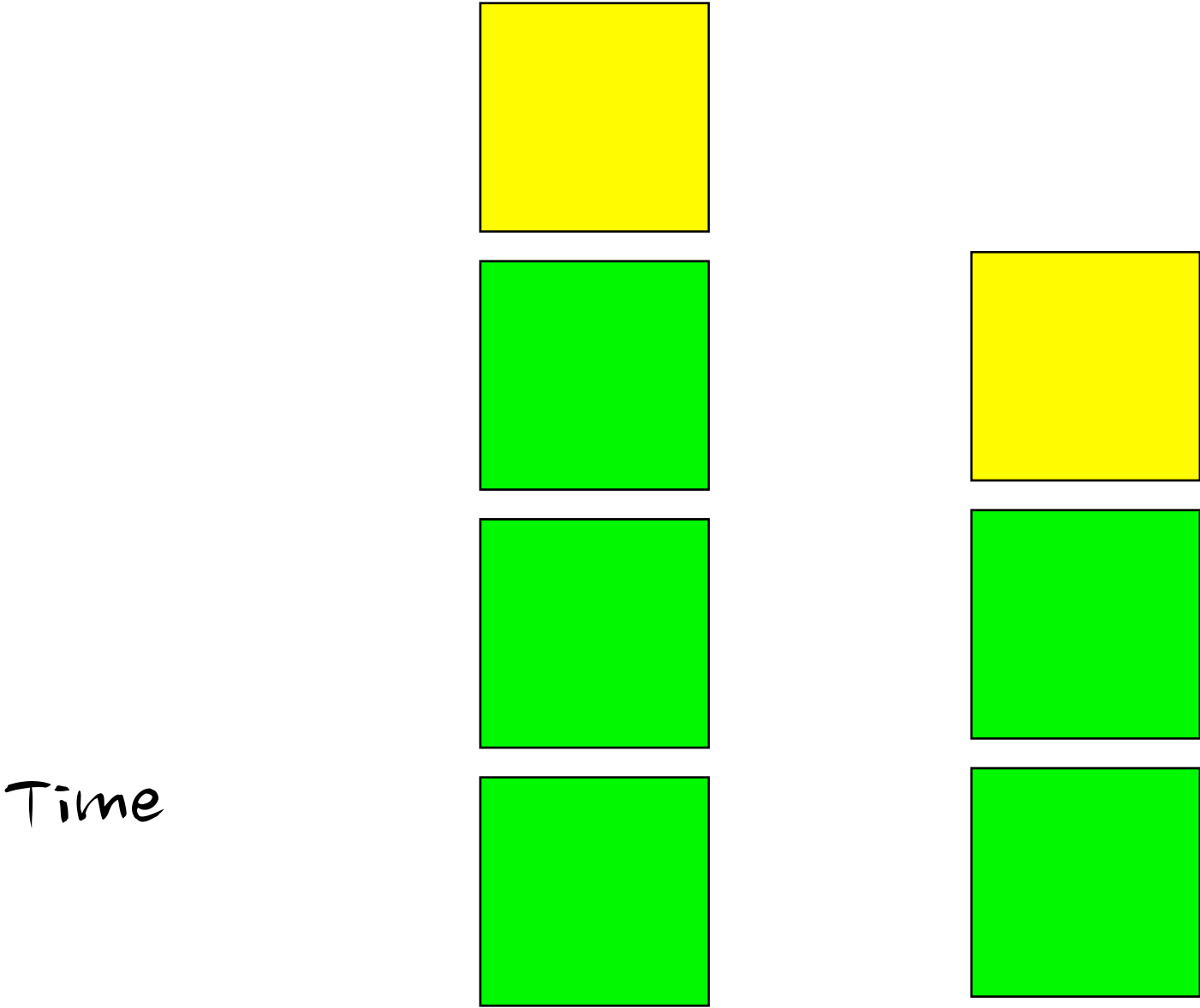
Tested

Time



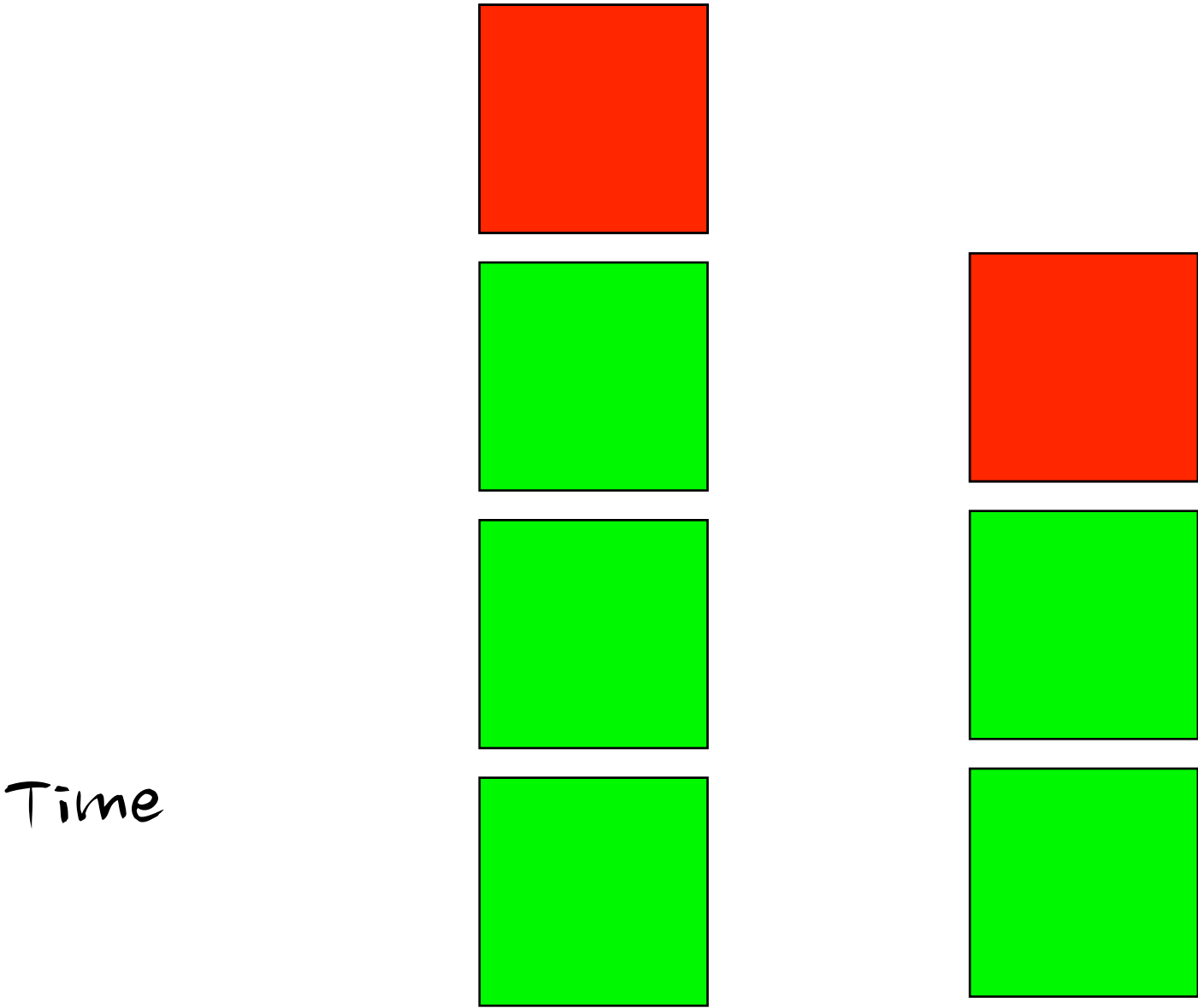
Merging (Centralized)

Concurrent Changes



Merging (Centralized)

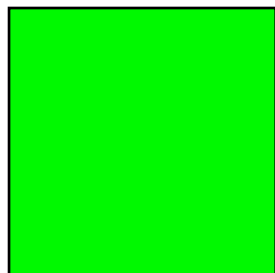
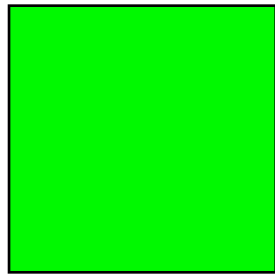
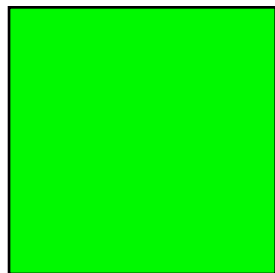
Concurrent Changes



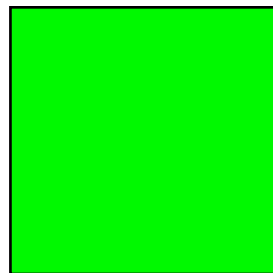
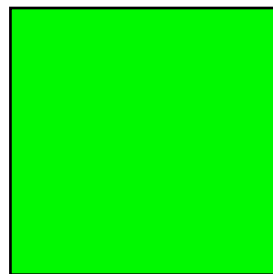
Merging (Centralized)

On Device 1

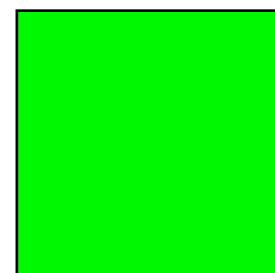
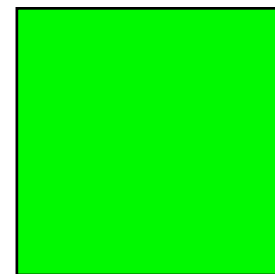
On Device 2



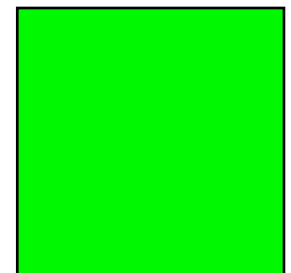
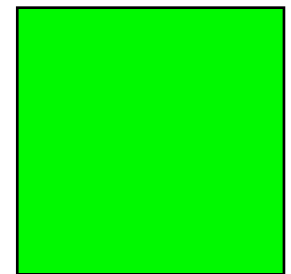
Saved Device 1



Saved Device 2



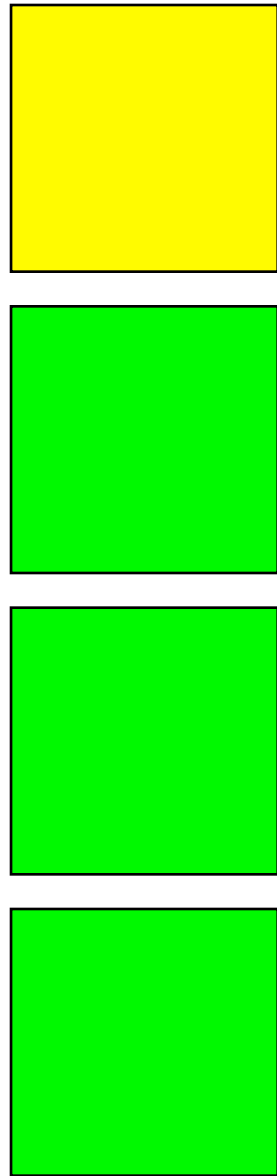
Saved Device 1



Saved Device 2

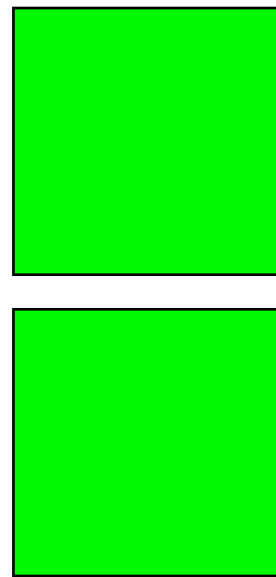
Merging (Decentralized)

On Device 1

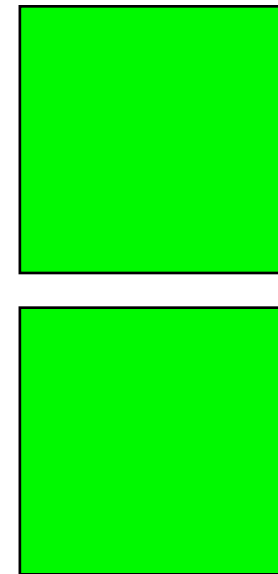


Saved Device 1

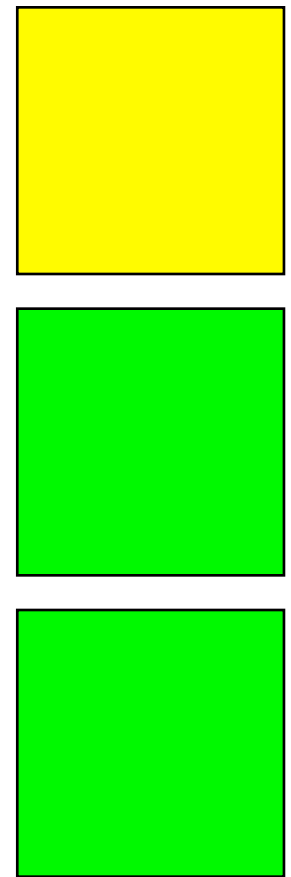
On Device 2



Saved Device 2



Saved Device 1

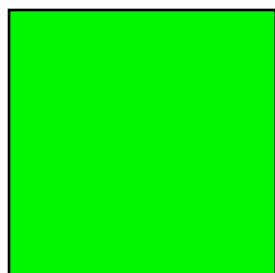
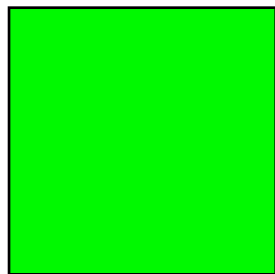
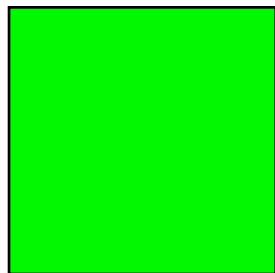
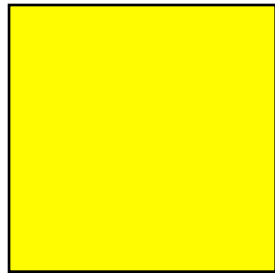


Saved Device 2

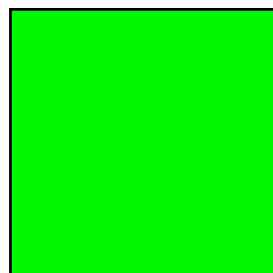
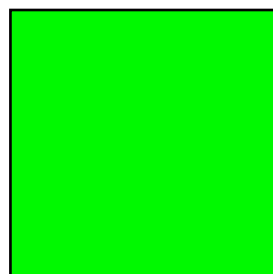
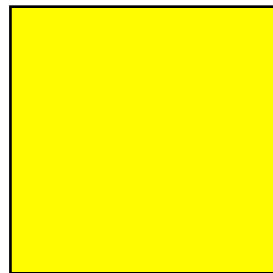
Merging (Decentralized)

On Device 1

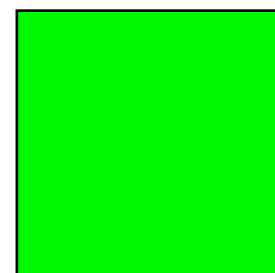
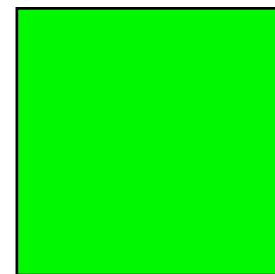
On Device 2



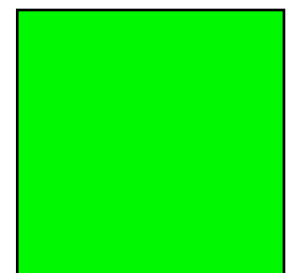
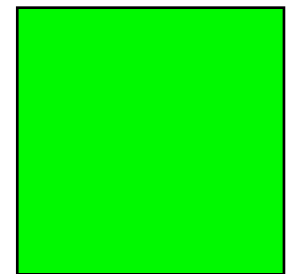
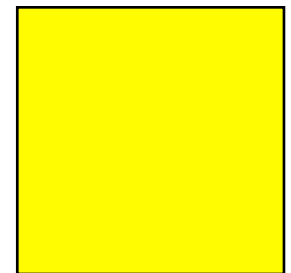
Saved Device 1



Saved Device 2



Saved Device 1

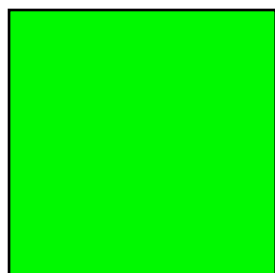
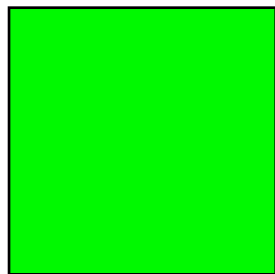
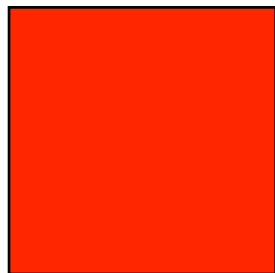
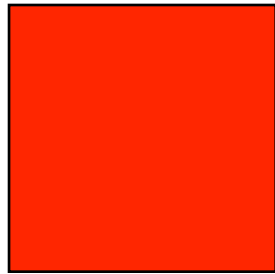


Saved Device 2

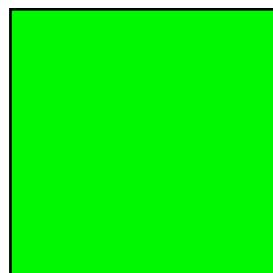
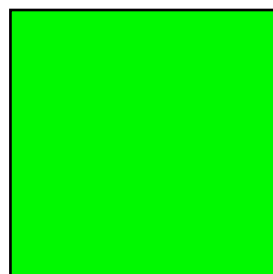
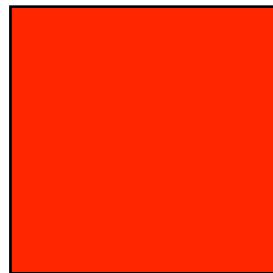
Merging (Decentralized)

On Device 1

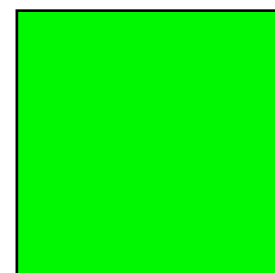
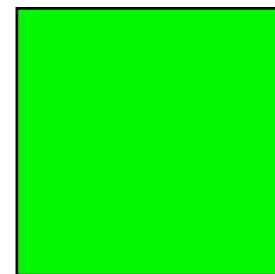
On Device 2



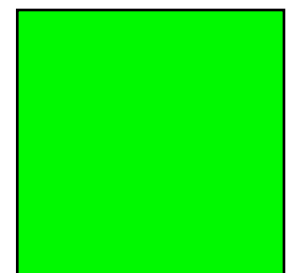
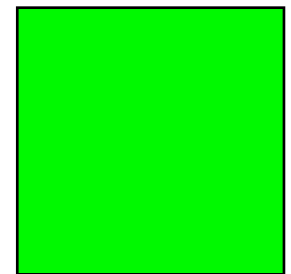
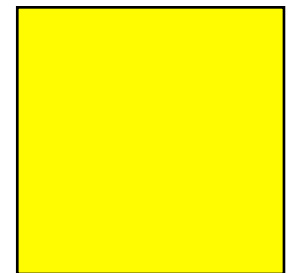
Saved Device 1



Saved Device 2



Saved Device 1

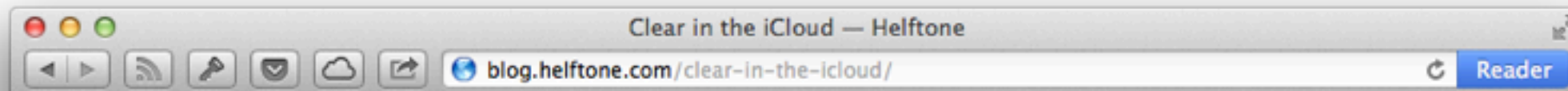


Saved Device 2

Merging (Decentralized)

Milen Dzhumerov (@milend)

blog.helftone.com/clear-in-the-icloud



helftone



Clear in the iCloud

30 May 2013
by Milen

There has been a lot of [talk](#) about iCloud + CoreData (referred to as iCCD hereafter) over the [past few months](#) and I think it is a good time for me to share our journey in getting iCloud integrated in Clear. If you do not want to read through all the details:

```
cloudFileSystem =  
    [[CDEICloudFileSystem alloc] initWithUbiquityContainerIdentifier:@"..."];  
ensemble =  
    [[CDEPersistentStoreEnsemble alloc] initWithEnsembleIdentifier:@"MainStore"  
        persistentStorePath:storePath  
        managedObjectModel:model  
        cloudFileSystem:cloudFileSystem];  
ensemble.delegate = self;
```

Easy...as possible

```
- (void)applicationDidBecomeActive:(UIApplication *)application
{
    if (!ensemble.isLeeched) {
        [ensemble leechPersistentStoreWithCompletion:NULL];
    }
    else {
        [ensemble mergeWithCompletion:NULL];
    }
}
```

Easy...as possible

```
- (void)persistentStoreEnsemble:(CDEPersistentStoreEnsemble *)ensemble
    didSaveMergeChangesWithNotification:(NSNotification *)notif
{
    [managedObjectContext performBlock:^(
        [managedObjectContext mergeChangesFromContextDidSaveNotification:notif];
    )];
}

- (NSArray *)persistentStoreEnsemble:(CDEPersistentStoreEnsemble *)ensemble
    globalIdentifiersForManagedObjects:(NSArray *)objects
{
    return [objects valueForKeyPath:@"uniqueIdentifier"];
}
```

Easy...as possible

Current Status

6000 lines of (dense) framework code

3500 lines of unit tests

First full sync a week ago

Where's the code?





github.com/drewmccormack/ensembles



Take This...

It's still early days for decentralized database sync

TICDS is currently best Core Data offering

Core Data Ensembles is next generation

Twitter: @drewmccormack

Attributions

Photo by Tim Morgan - <http://flic.kr/p/7FrDi>

Photo by sflaw - <http://flic.kr/p/kFToD>

Photo by mattwalker69 - <http://flic.kr/p/eSc17j>

Photo by nishantcm - <http://flic.kr/p/dEeWpH>

Photo by OliBac - <http://flic.kr/p/6wJHQC>

[Photo by Official U.S. Navy Imagery - http://flic.kr/p/9GAmKp](http://flic.kr/p/9GAmKp)