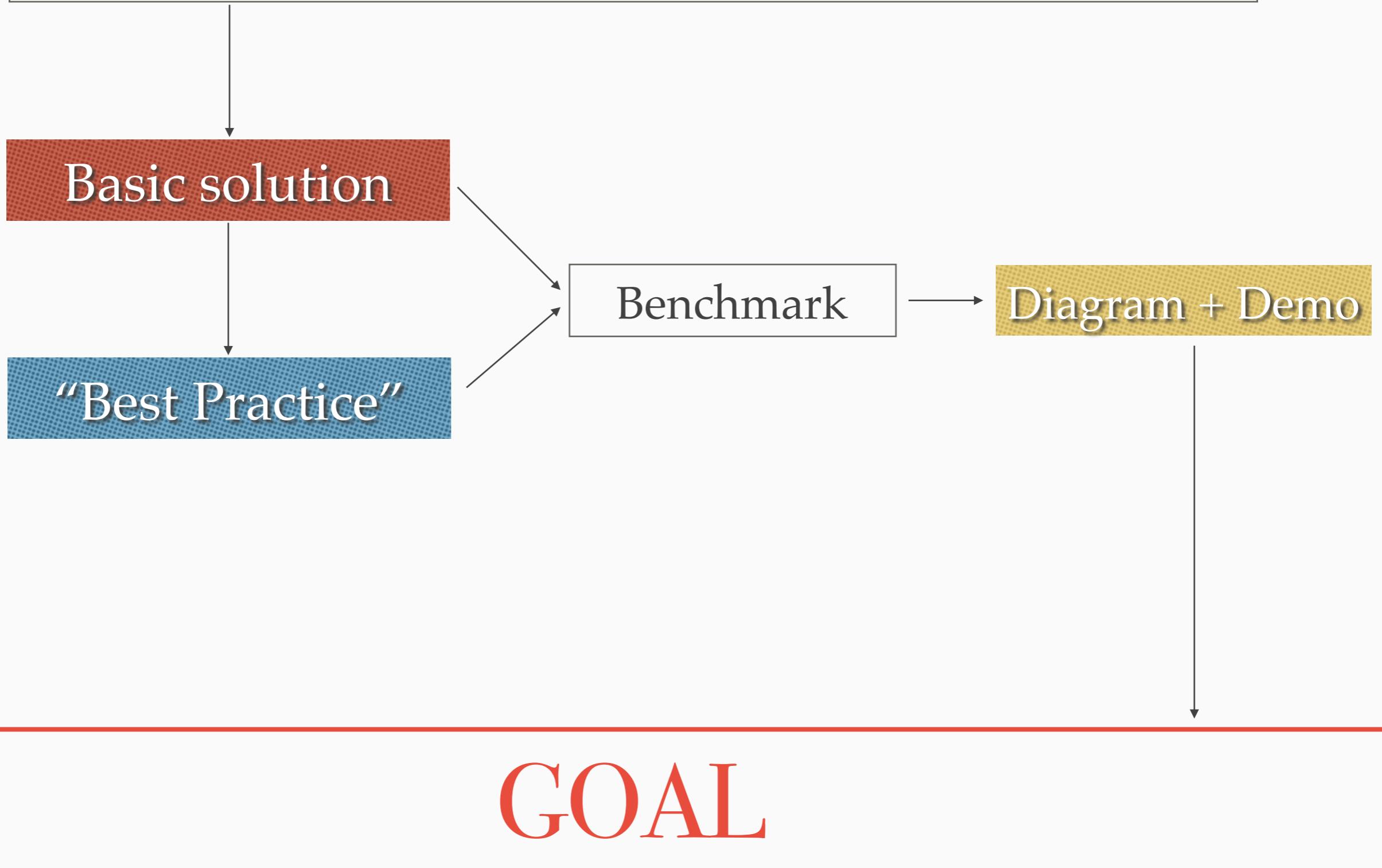

Tek Talk #20



github.com/NghiaTranUIT

nghiatran.xyz

Common problem I encountered while developing feels app



Carrier

10:53 PM



Suggested users for you

[View all >](#)

SUGGESTED USERS

cannibalecore
Ahetze, France

[+ Follow](#)



velvet.alien
London

[+ Follow](#)



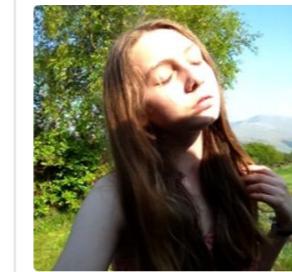
cannibalecore
Ahetze, France

[+ Follow](#)



velvet.alien
London

[+ Follow](#)



theraggedpriest ✅

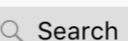
22m



EXPLORE POSTS



[View all >](#)



Search

10:53 PM

Carrier

10:53 PM



Search

10:53 PM

Carrier

10:53 PM



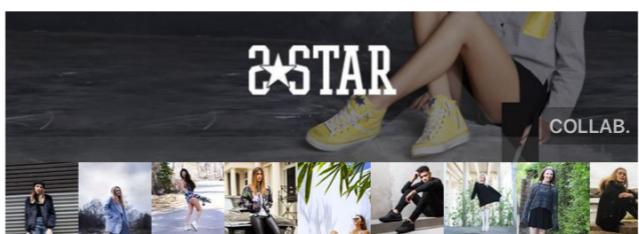
10:53 PM

Discover

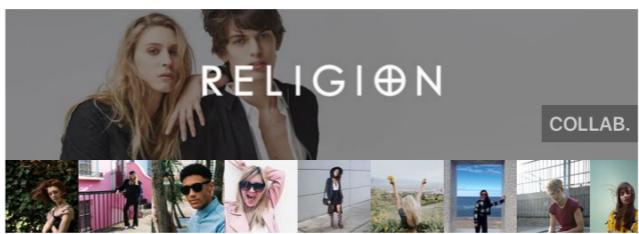


10:53 PM

nghiatran



Unique and bold, Italian footwear brand of a generation 2STAR is looking for global brand ambass... >
May - June 2016



Fearless brand Religion is looking for music loving
ambassadors to rock out at a festival! >
May - June 2016



Nghĩa Trần

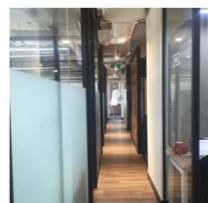
Do anything at Feels...

[Edit Profile](#)

61
Posts

251
Followers

1012
Following



Feed

Explore

Collab

Profile

Common Problem for Beginner

1. *Download image async*
2. *Prevent swapping image in UITableView*
3. *Stop on-going image if it's out of screen*
4. *Caching (memory + disk)*
5. *Preheating*
6. *Global Image Downloader queue*

Bad Practice

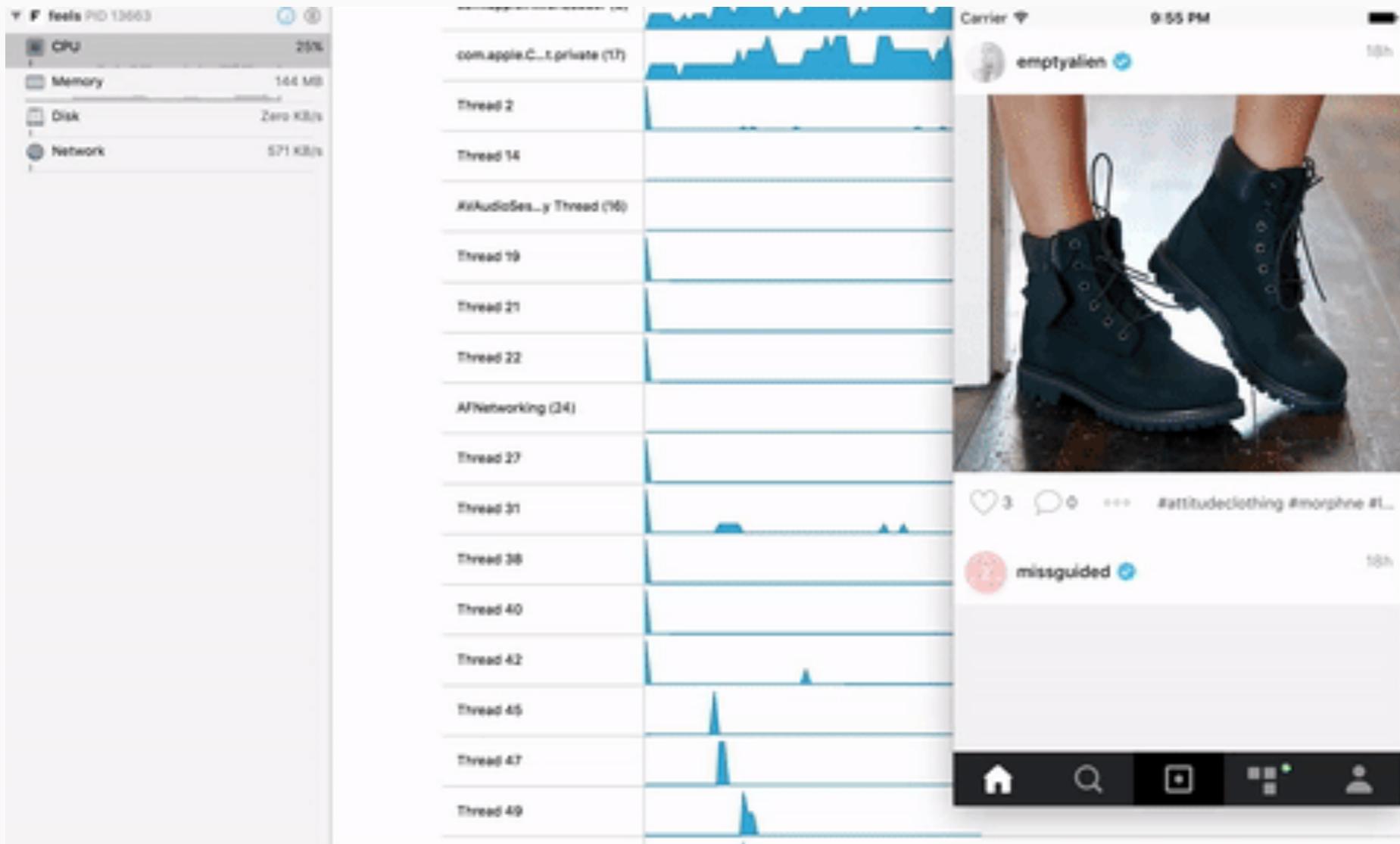
```
FeInsHomeFeedCell *cell = [self.tableView dequeueReusableCellWithIdentifier:[FeInsHomeFeedCell idCell] forIndexPath:indexPath];
PFObject *photoObj = obj;

// Configure
cell.timeFormatter = self.timeIntervalFormatter;
cell.delegate = self;
[cell configureCellWithObj:photoObj forDisplayingCell:NO];

// Download Image
NSString *path = photoObj[kFe_API_Item_image];
NSURL *URL = [NSURL URLWithString:path];

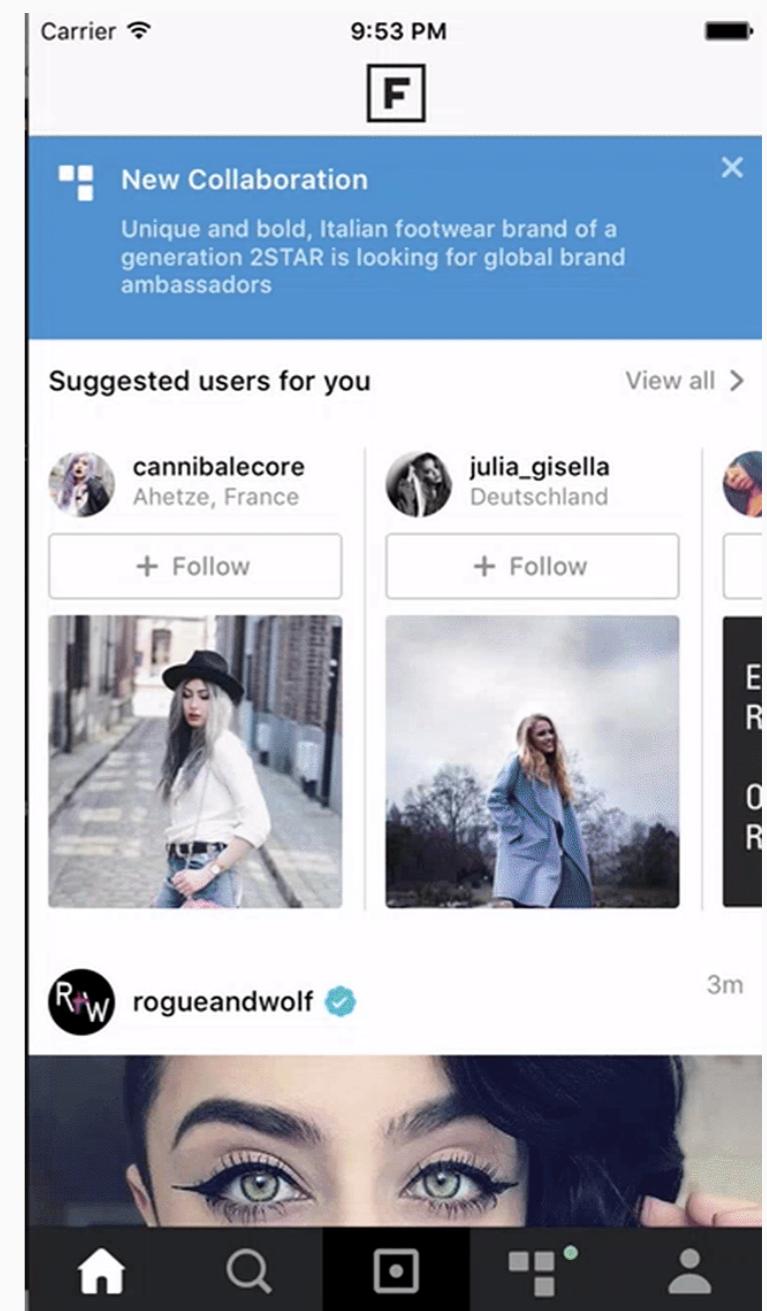
dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_BACKGROUND, 0), ^{
    // Download here
    NSData *data = [NSData dataWithContentsOfURL:URL];
    UIImage *image = [UIImage imageWithData:data];
    dispatch_async(dispatch_get_main_queue(), ^{
        cell.photoImageView.image = image;
    });
});
```

1. While scrolling faster in long table view
2. No caching
3. No Preheating
4. Swap image while scrolling quickly
5. No Global Image Download Queue



Benchmark

- ❖ Too many thread is created.
- ❖ Memory grows significant
- ❖ ...



AFNetworking

```
-(void) prepareForReuse
{
    [super prepareForReuse];
    self.photoImageView.image = nil;
```

```
- (void) downloadPhoto
{
    NSString *path = self.currentObj[kFe_API_Item_image];
    NSURL *URL = [NSURL URLWithString:path];
    [self.photoImageView setImageWithURL:URL];
}
```

- Vẫn chưa có 1 queue tổng thể để quản lý => phải handle từng cell 1
- AFNetworking chỉ cache ở tàng memory và cache ở disk dưới dạng NSURLRequest policy => Hạn chế là phải dùng 1 số thằng trung gian để check xem hình đã cache ở disk chưa

“Best” Practice

- ❖ FeGlobalImageManger
- ❖ FeDownloadOperation : NSOperation (async)
- ❖ FeCompletionOperation : NSOperation (as dependency)
- ❖ Have ability to handle InternetChangedState, TabbarChangedState, cancel all operation if need
- ❖ Integrate with AFNetworking / Alamofire and AlamofireImage
- ❖ or Integrate with DFImageManager

<https://www.draw.io/i/KZTyODI>

Benchmark

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

100% Sat 10:04

F feels > iPhone 6 Finished running feels on iPhone 6 6

By File By Type

PROJECT TARGETS

feels F feels

General Capabilities Resource Tags Info Build Settings Build Phases Build Rules

Identity

Bundle Identifier com.feelsfashionltd.feels
Version 2.6
Build 650200
Team FEELS FASHION LTD

Deployment Info

Deployment Target 8.0
Devices iPhone
Main Interface
Device Orientation Portrait
 Upside Down
 Landscape Left
 Landscape Right
Status Bar Style Default
 Hide status bar
 Requires full screen

App Icons and Launch Images

App Icons Source AppIcon
Launch Images Source LaunchImage
Launch Screen File

Embedded Binaries

Add embedded binaries here

Filter

Icons: Finder, Mail, Safari, Visual Studio Code, Sketch, Xcode, GitHub, iTerm, Google Chrome, iMessage, App Store, System Preferences, Activity Monitor, Quick Look, Trash

Bonus

AWS S3 + IMGIX is the best choice



- ❖ Auto configure with your AWS bucket.
- ❖ Resize and crop your image on-demand.
- ❖ Face detection.

Best IMGIX recipe

<http://imgix.feels.com/user/x7vetgylbr/23e9b073-3fec-42c4-8f6c-ee57f4109241.jpg>

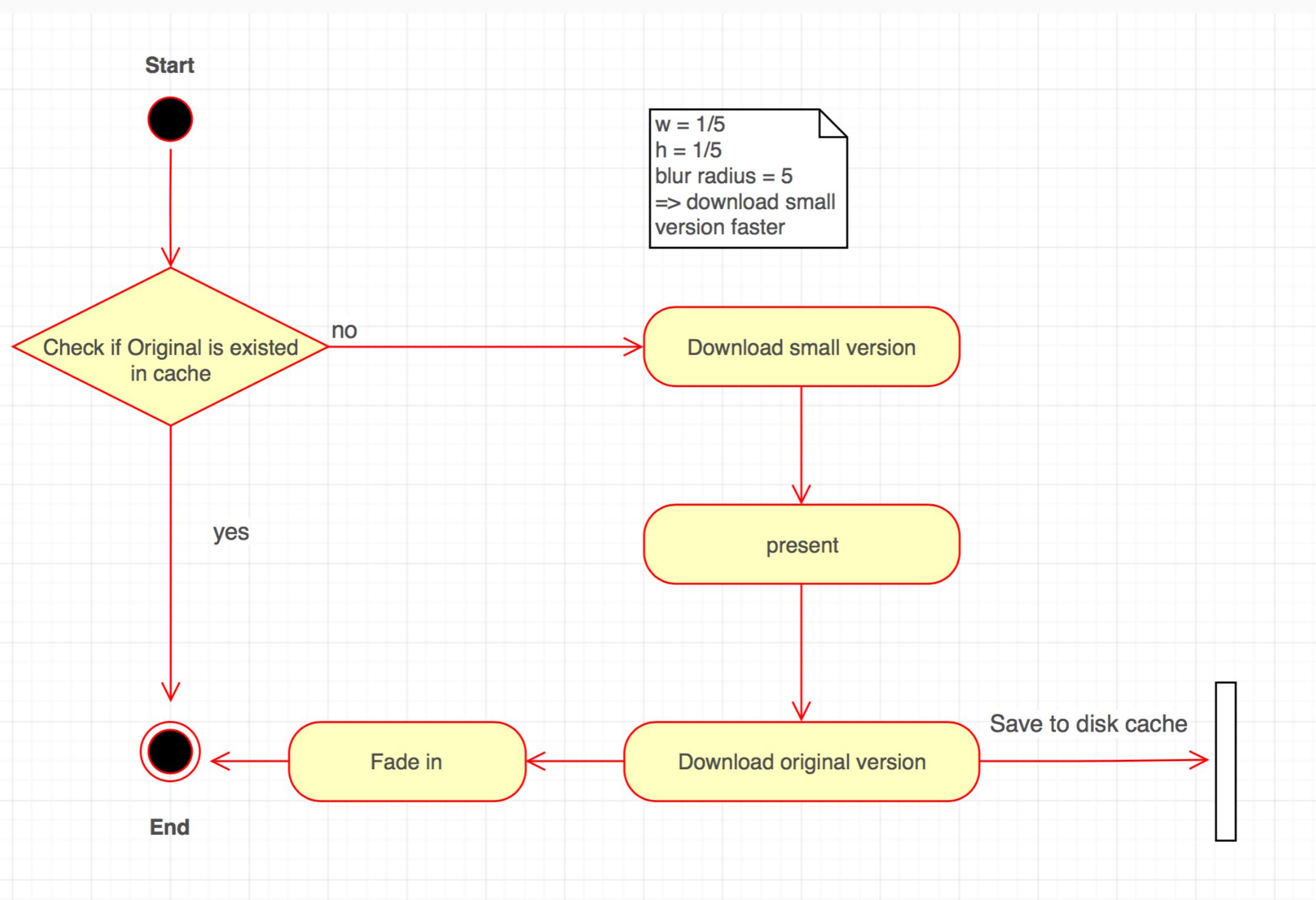
w + h + dpr + fit=crop + crop=faces,top + fm=jpg + q=50

imgix.feels.com/user/x7vetgylbr/23e9b073-3fec-42c4-8f6c-ee57f4109241.jpg?w=375&h=375&dpr=2&fit=crop&crop=faces,top&fm=jpg&q=50

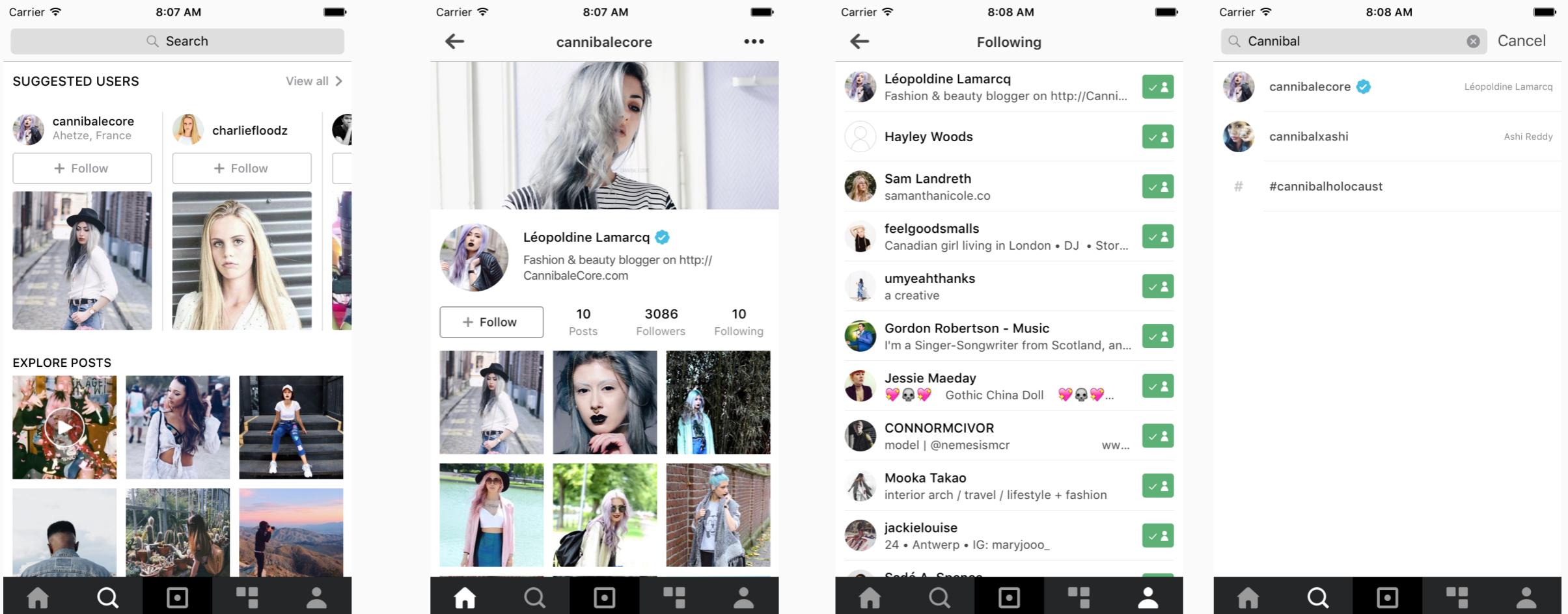
Cool Effect



Diagram



Problem



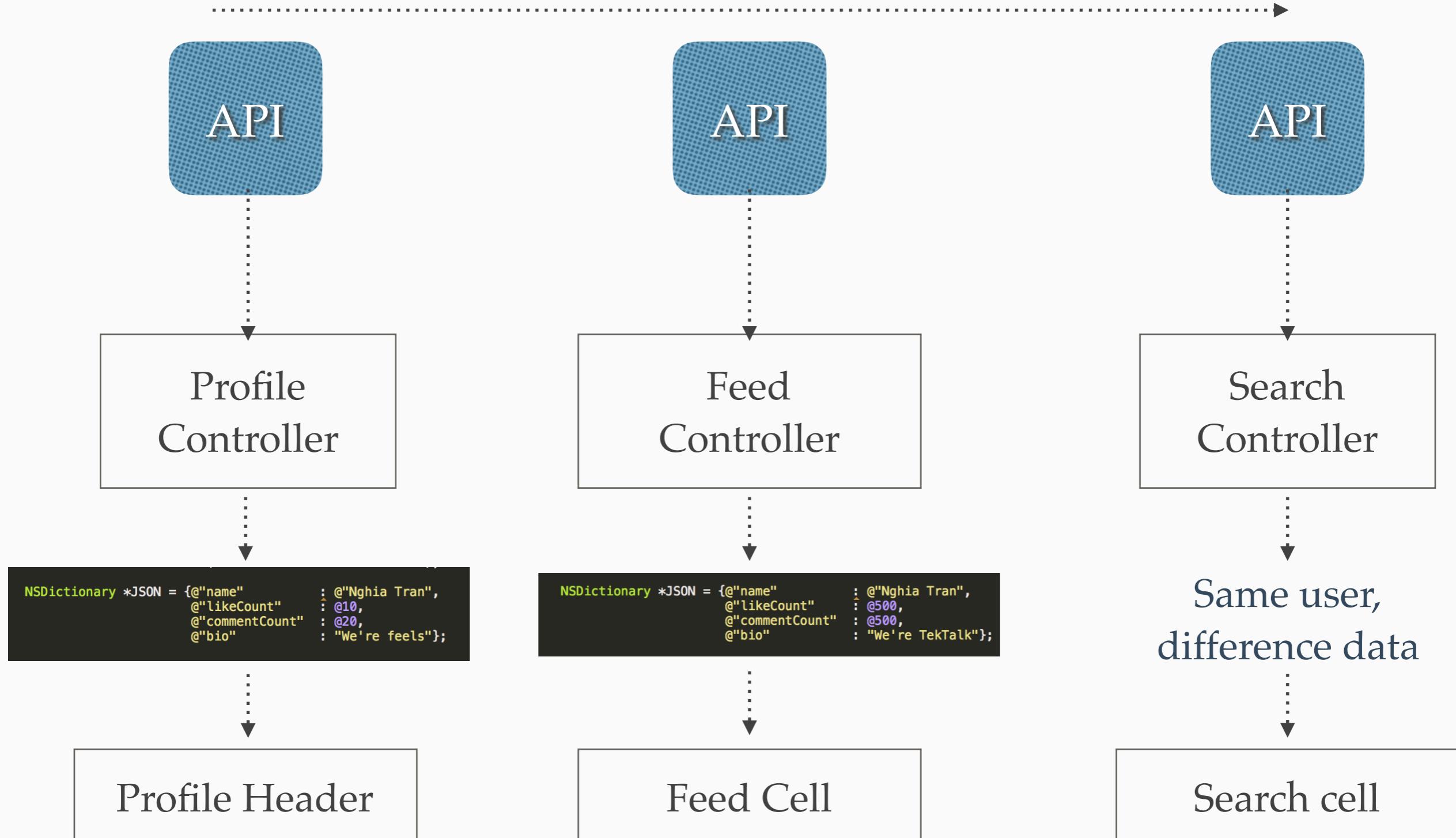
We have many screen - same user data

So what happen if **username** was changed? How can we make all consistent???

What happen if we have bunch of new data?

Such as **FollowerCount**, **Following Count**, **isVerified**, **bio**, ...?

Timeline



Show data

```
354
355 -(void) configureData
356 {
357     // Get data
358     NSNumber *followerCount = self.currentUserObj[kFe_API_User_followerCount];
359     NSNumber *followingCount = self.currentUserObj[kFe_API_User_followingCount];
360     NSNumber *postCount = self.currentUserObj[kFe_API_User_postsCount];
361     NSNumber *isVerify = self.currentUserObj[kFe_API_User_isVerified];
362     NSString *bio = self.currentUserObj[kFe_API_User_bio];
363     NSString *username = self.currentUserObj[kFe_API_User_username];
364
365     self.followerCountLbl.text = [followerCount stringValue];
366     self.followingCountLbl.text = [followingCount stringValue];
367     self.postCountLbl.text = [postCount stringValue];
368     self.verifiedImage.hidden = isVerify && isVerify.boolValue ? NO : YES;
369     [self updateBioWithText:bio];
370     self.usernameLbl.text = username;
371 }
372 }
```

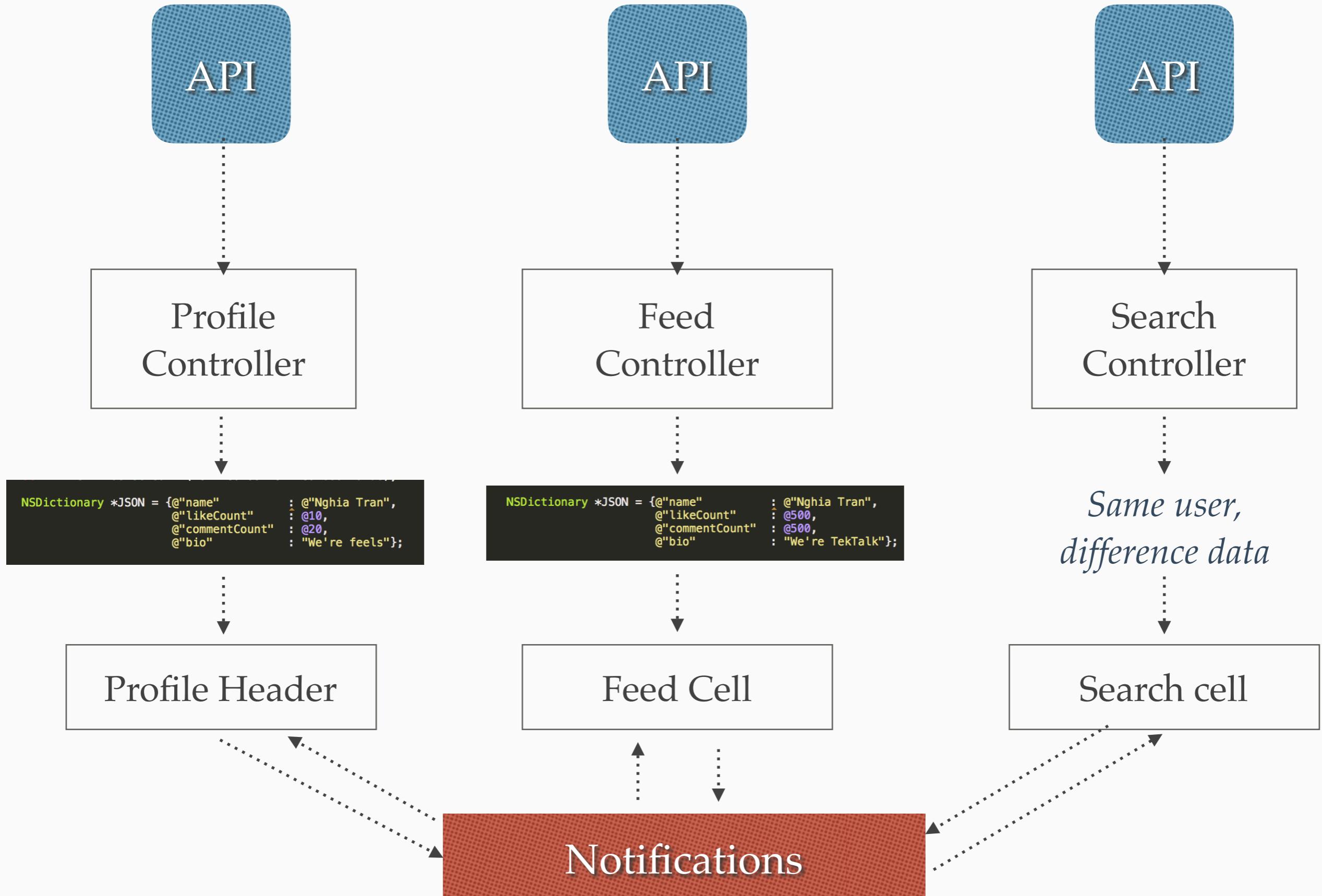
Profile header

```
-(void) configreDataForCell
{
    // Data
    NSNumber *likeCount = self.currentObj[kFe_API_Item_likesCount];
    NSNumber *commentCount = self.currentObj[kFe_API_Item_commentsCount];
    NSNumber *isLiked = self.currentObj[kFe_API_Item_IsLikeByCurrentUser];

    // Show
    [self.likeBtn setTitle:[likeCount stringValue] forState:UIControlStateNormal];
    [self.commentBtn setTitle:[commentCount stringValue] forState:UIControlStateNormal];
    self.likeBtn.selected = (isLiked && isLiked.boolValue);
    ...
}
```

Feed cell

Bad Practice

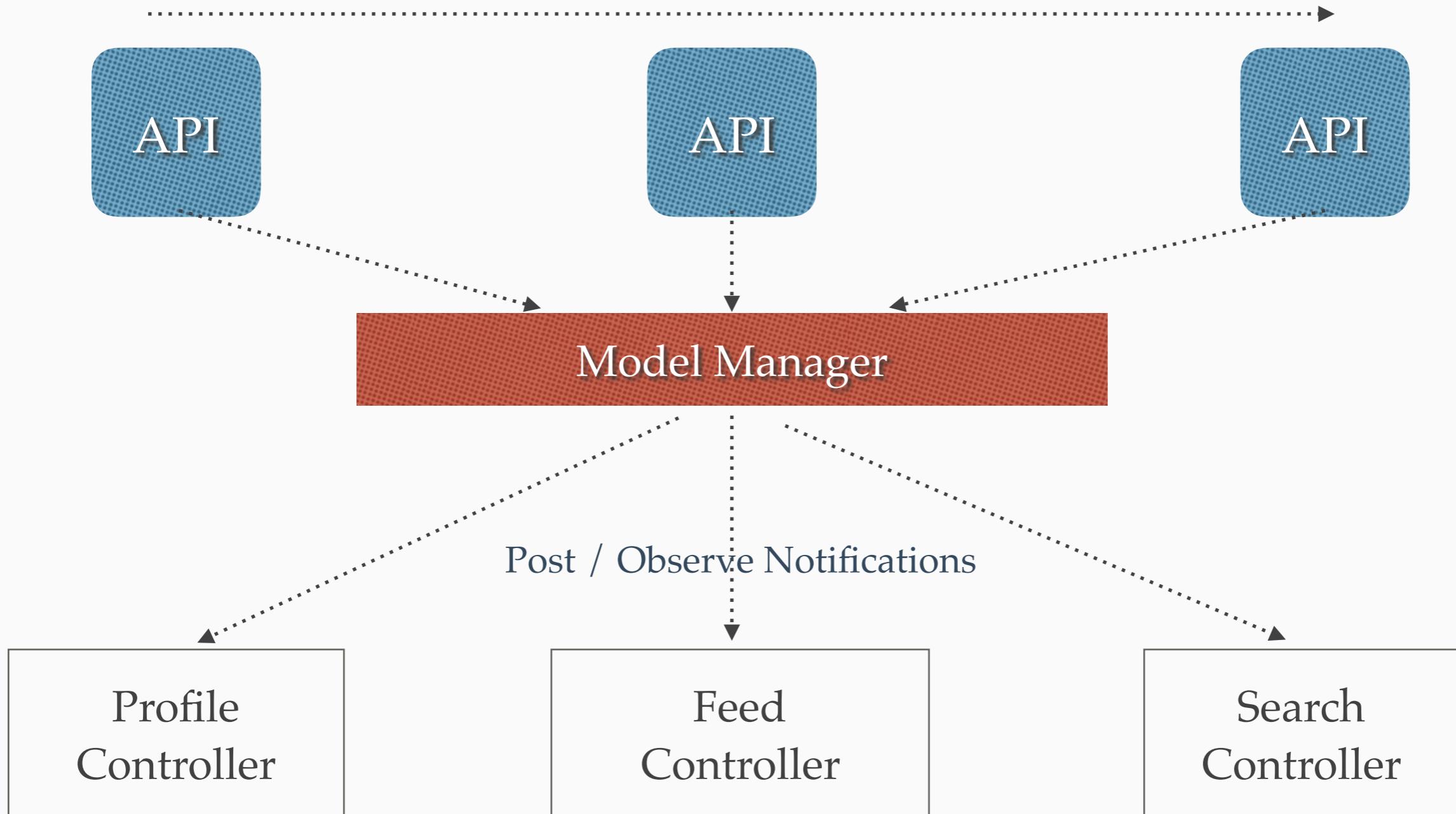


Benchmark + Demo

- ❖ Data user isn't consistently
- ❖ Too many duplicated code for posting / Observer notifications
- ❖ Call reload table view cell too many
- ❖ It's nightmare if we have 10+ screens, 10+ new data
- ❖ Render cells is too expensive

Practice

Timeline



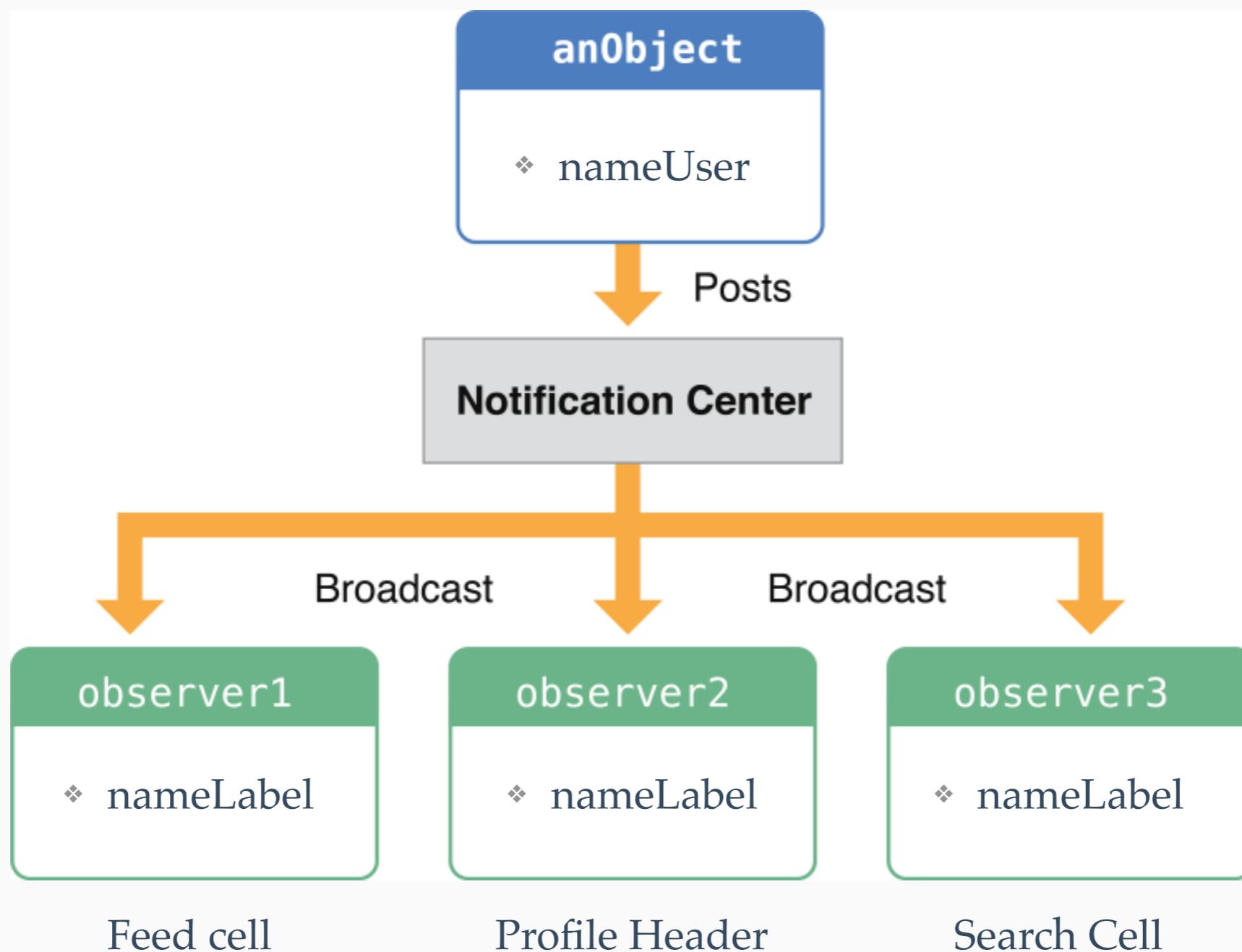
Post notification to previous screen for updating new data

Benchmark + Demo

“Best” Practice

FBKVO + DataModel

What is KVO



Sample code

```
__weak typeof(self) weakSelf = self;
[PFCloud feels_callFunctionInBackgroundWithAPIType:FeAPIManagerAPIType_V2_Profile_Mention withParameters:param block:^(id _Nullable object, NSError *_Nullable error) {

    FeUserCache *userCache = [[FeDataCache shareCache] attributesForUser:weakSelf.currentUserObj];
    [userCache updateUserCacheWithUser:weakSelf.currentUserObj];
```

Get data -> Update Model data

```
// Save
self.followerCount = newFollower;
self.followingCount = newFollowing;

self.bio = bio;
self.location = location;

// Username
self.username = username;

// Verify
self.isVerified = isVerify;

// Video count
self.videoCount = videoCount;

// Music count
self.musicCount = musicCount;

// Look
self.lookCount = lookCount;

// Post
self.postCount = postCount;

// First + last
self.firstName = firstName;
self.lastName = lastName;
```

FeUserData

FBKVO

```
// Cache
FeUserCache *userCache = [[FeDataCache shareCache] attributesForUser:self.currentUserObj];
```

```
// Username
[self.KVO observe:userCache keyPath:@"nameUser" options:NSKeyValueObservingOptionInitial | NSKeyValueObservingOptionNew block:^(id observer, id object, NSDictionary *change) {
```

```
// Is verified icon
[self.KVO observe:userCache keyPath:@"isVerified" options:NSKeyValueObservingOptionInitial | NSKeyValueObservingOptionNew block:^(id observer, id object, NSDictionary *change) {
```

```
// Bio
[self.KVO observe:userCache keyPath:@"bio" options:NSKeyValueObservingOptionInitial | NSKeyValueObservingOptionNew block:^(id observer, id object, NSDictionary *change) {
    typeof(self) strongSelf = weakSelf;

    // Run on main thread
    runOnMainThreadIfNeed:^{
        id value = change[NSKeyValueChangeNewKey];
        NSString *bioStr = (NSString *) value;
        // Update
        [strongSelf updateBioWithText:bioStr];
    });
}];
```

Diagram

<https://www.draw.io/i/3ZtTLmb>

Benchmark + Demo

- ❖ No more Notification code
- ❖ No need to reload TableView/CollectionView
- ❖ All is updated automatically -> Just need to update model -> FBKVO will do the rest