

# Coding Exercise

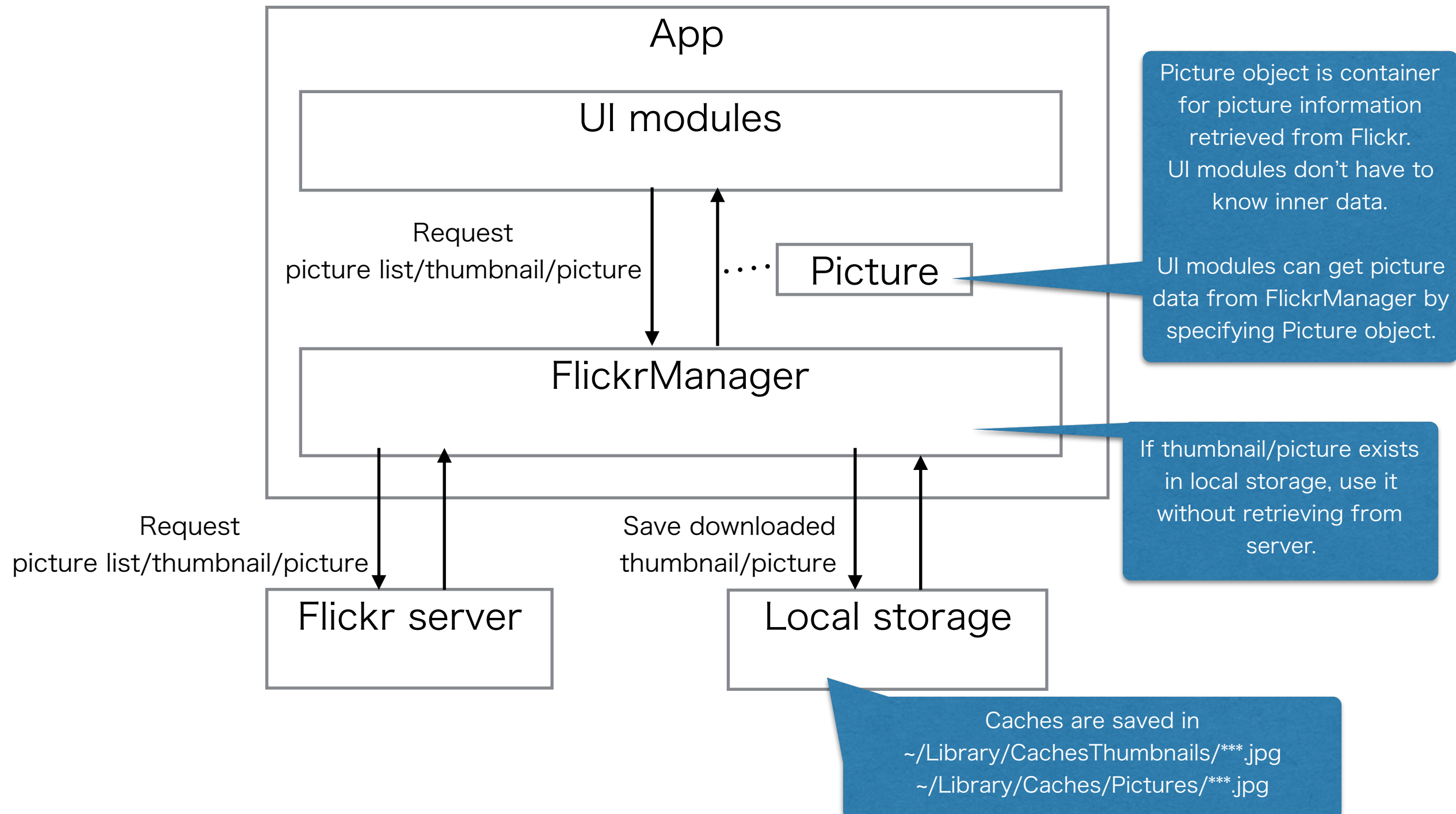
Software architecture

2016/6/6

Hiromitsu Oka

[oka.hirow.pc@gmail.com](mailto:oka.hirow.pc@gmail.com)

# Modules



# Interface

## FlickrManager

```
@interface FlickrManager : NSObject

// Singleton
+ (instancetype)sharedInstance;

// Retrieve picture list from Flickr.
// Specified count of pictures will be retrieved.
// If retrieving succeeded, completion will return pictureList as array of Picture object.
// If retrieving failed, completion will return error and pictureList will be nil.
// completion will be called in UI thread.
- (void)retrievePictureListWithType:(PictureListType)type count:(NSUInteger)count completion:(void(^)(NSArray* pictureList,
NSError* error))completion;

// Retrieve thumbnail image from Flickr.
// If cached image exists, it will be returned without retrieving from Flickr.
// If retrieving succeeded, completion will return UIImage object.
// If retrieving failed, completion will return error and image will be nil.
// completion will be called in UI thread.
- (void)retrieveImageOfPicture:(Picture*)picture forThumbnail:(BOOL)forThumbnail completion:(void(^)(UIImage* image, NSError*
error))completion;

// Return UIImage object if cache exists.
// If not exists, return nil.
// This method must be called in UI thread.
- (UIImage*)getCacheImageOfPicture:(Picture*)picture forThumbnail:(BOOL)forThumbnail;

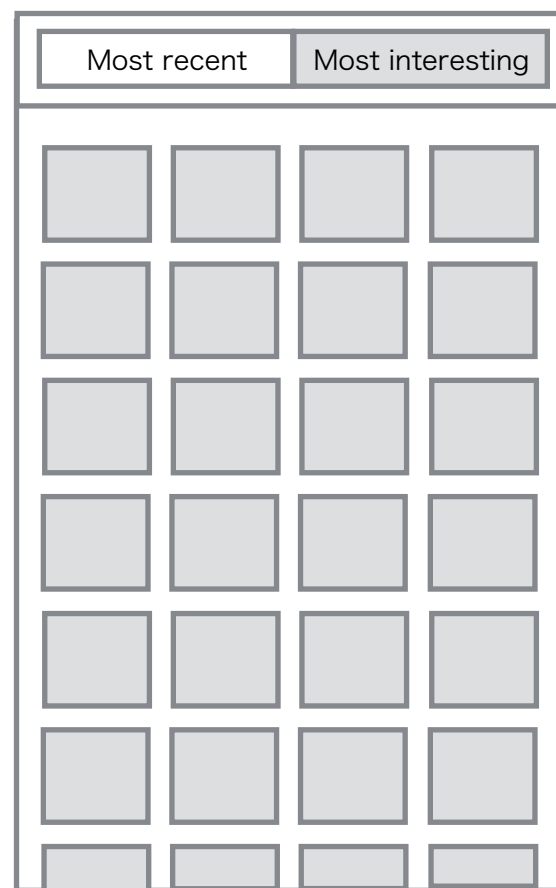
// Delete all cached thumbnails and pictures in ~/Library/Cache/
- (void)deleteAllCacheFile;

@end
```

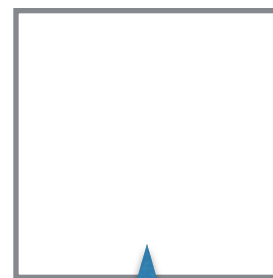
These interfaces use only Picture objects and UIImage objects.  
Not depend on UI structure.

# UI structures

## ThumbnailListVC : UICollectionViewController

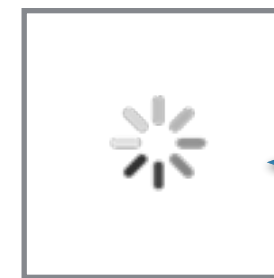


ThumbnailCell



This cell has 3 layers.

LoadingCell



Displayed while retrieving picture list.



Loading view:  
Displayed while retrieving thumbnail .



Error view:  
Displayed if retrieving failed.

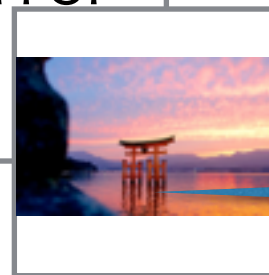
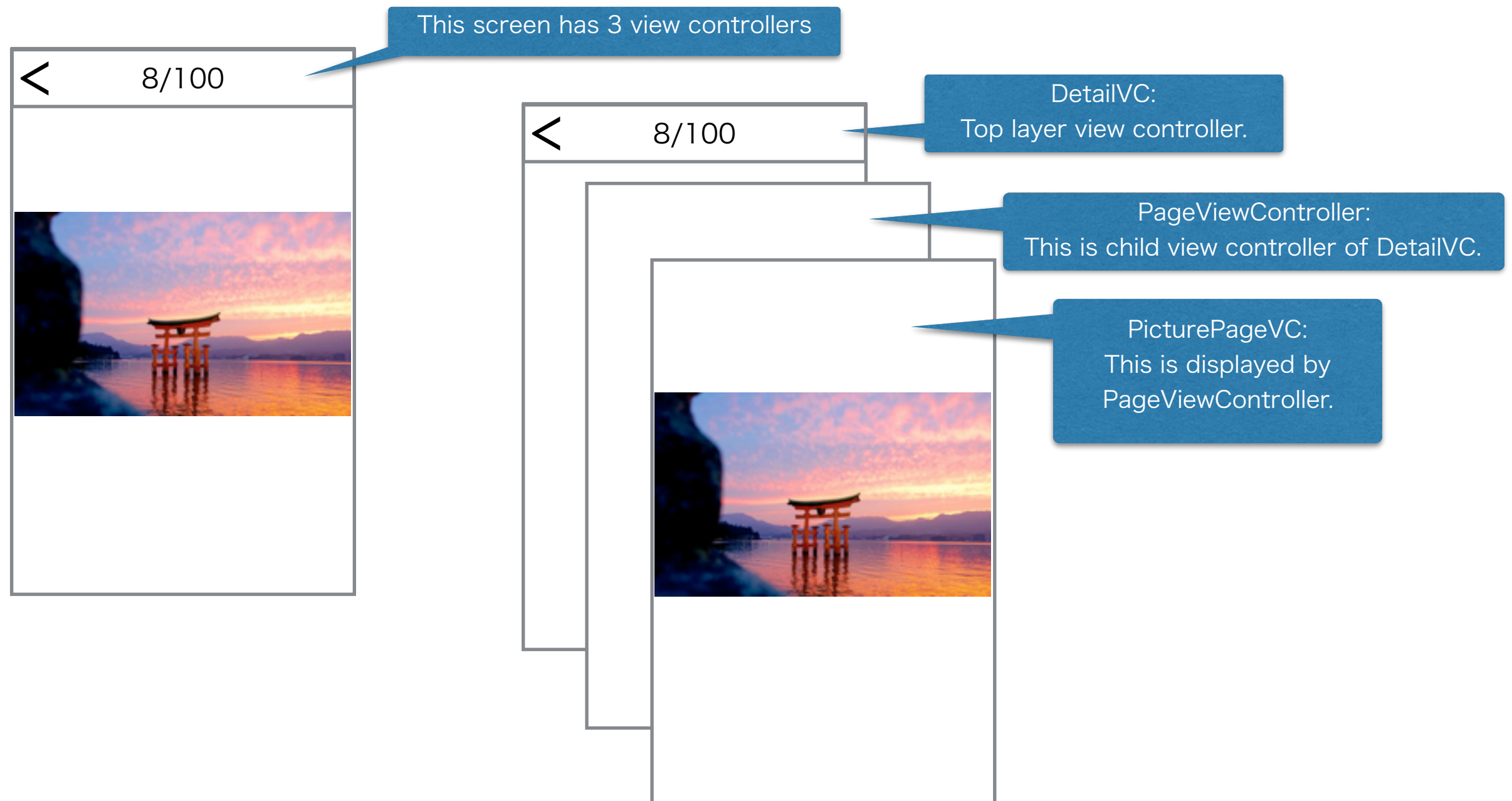


Image view:  
Displayed if retrieving succeeded.

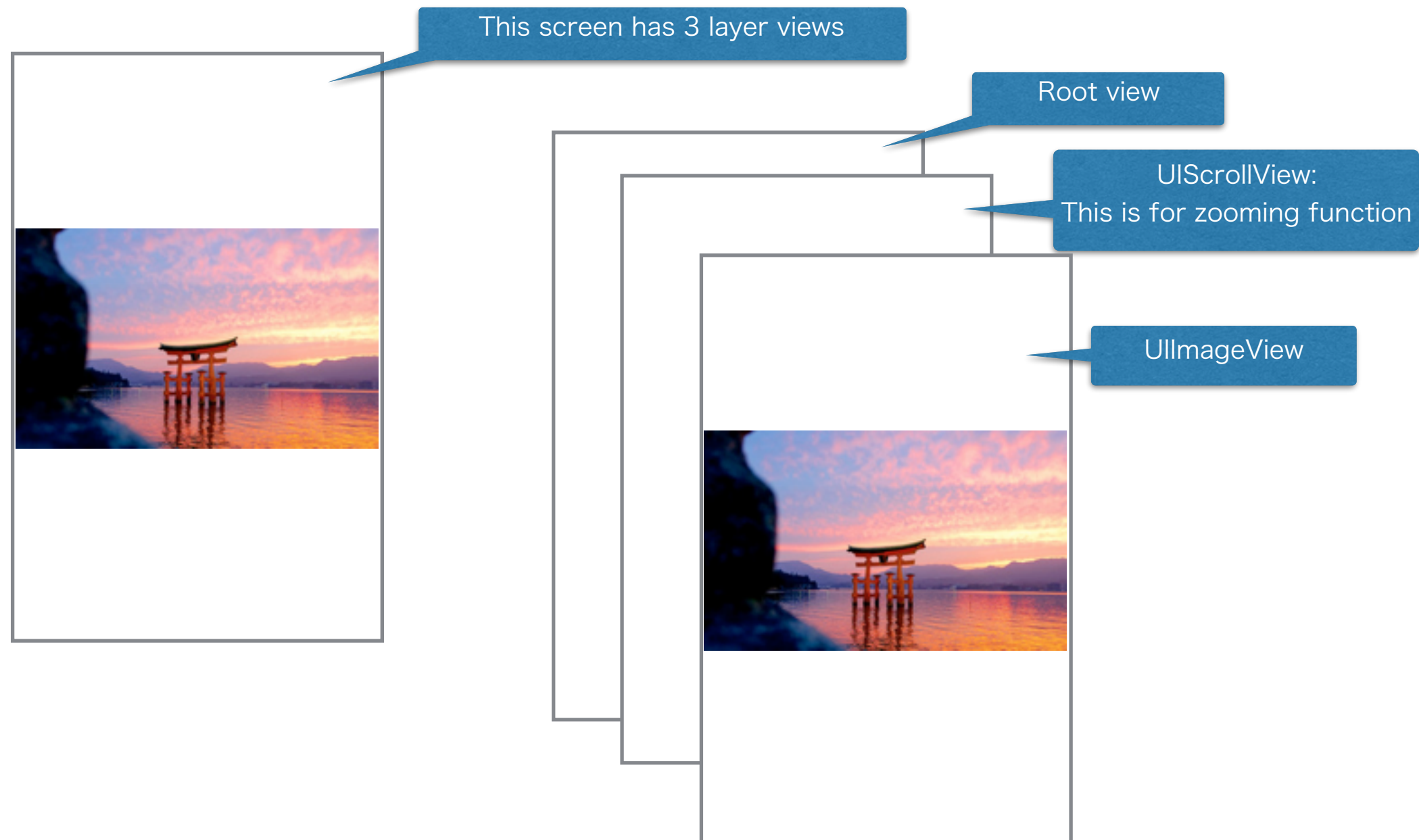
# UI structures

## DetailVC : UIViewController



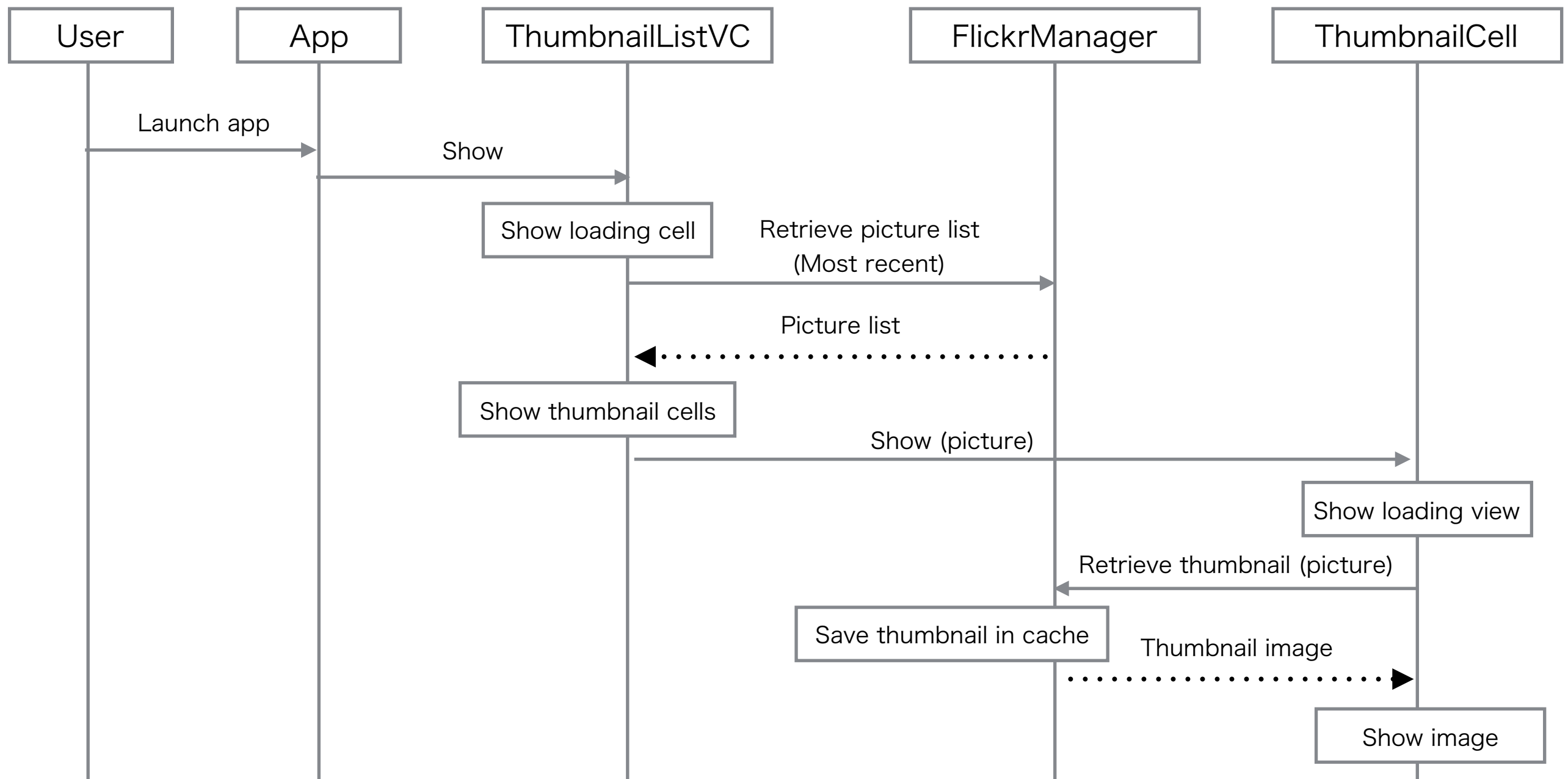
# UI structures

PicturePageVC : **UIViewController**



# Main sequence

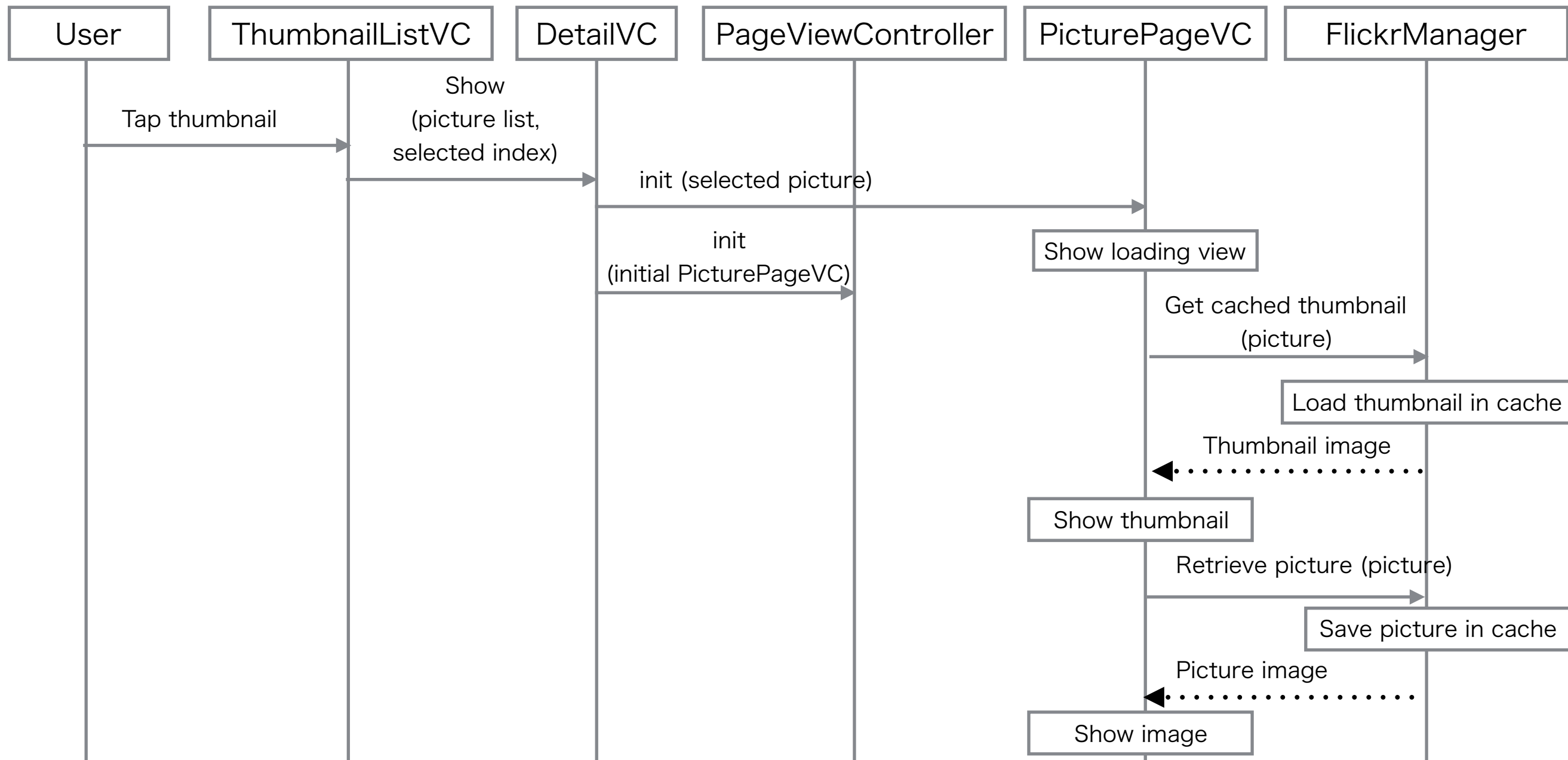
- Selects picture and see it. (1 / 2)





# Main sequence

- Selects picture and see it. (2/2)





# Performance measure

- In Thumbnail list screen, when user scrolls list, many cells will request to get thumbnail at same time.
- It will cause not smooth scroll.
- So I implemented almost code about displaying thumbnail will be executed in background thread.
- In background thread:
  - Check whether cache file exists.
  - Network request for Flickr by AFNetwork module.
  - Save retrieved image into cache.
  - Load image data from cache file.
- In UI thread:
  - Convert image data into UIImage object.
  - Show error dialog.
  - Update imageView.
- Point:
  - We can execute file operations in background thread.
  - We cannot handle UIImage in UI thread.
  - So I split to creating UIImage code into 2 step.
    - Create NSData by file path. (In background thread)
    - Create UIImage by NSData. (In UI thread)

# TODOs

- Consider to retry retrieving picture if retrieving was failed.
  - Now, if retrieving is failed, user cannot see the picture ever.
- Consider how to see more pictures in Flickr.
  - Now, app shows only 100 pictures.
  - Because picture list is updated often in Flickr, so app cannot get actual next picture list by current Flickr's apis.
  - We need api to retrieve next picture list by specifying like a picture ID as index to get actual next picture list.
- Consider cache remove timing.
  - Now, app remove all caches when starting retrieving picture list.
  - Because picture list is updated often in Flickr, so caches of previous picture list will not be displayed next timing.
  - But if we implement above "see more picture" function, old cache may be used.
- Consider thumbnail retrieving timing.
  - Now, app starts retrieving thumbnail when cell will be displayed, so loading cell will be displayed at first.
  - Maybe app should retrieve not yet displayed thumbnails in background.
- Consider double tap zoom in detail screen.
  - Now, app only supports pinch zoom. (for my working hours)
- Consider thumbnail cell's size.
  - Now, thumbnail cell's size is 100\*100, so space between cells is large in some device.
  - App should change cell's size according to screen resolution.
- Consider ATS settings.
  - Now, ATS settings is "Allow arbitrary loads = YES" to allow http load for Flickr api.
  - But app should limit ATS settings only for Flickr api.