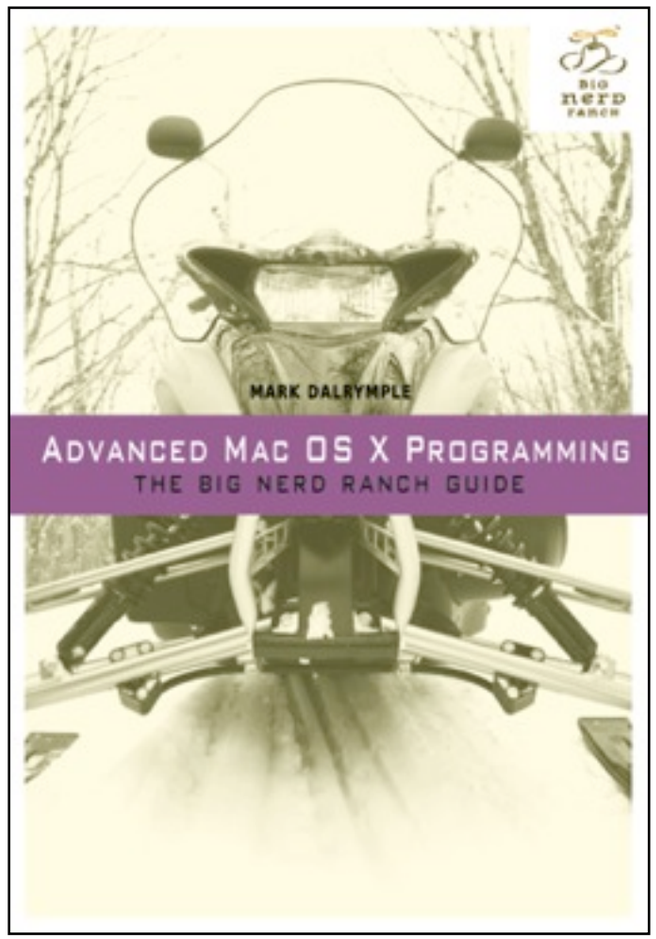
Up next:
The Humble Header

Mark Dalrymple

@borkware

http://borkware.com/cocoaconf



The Humble Header

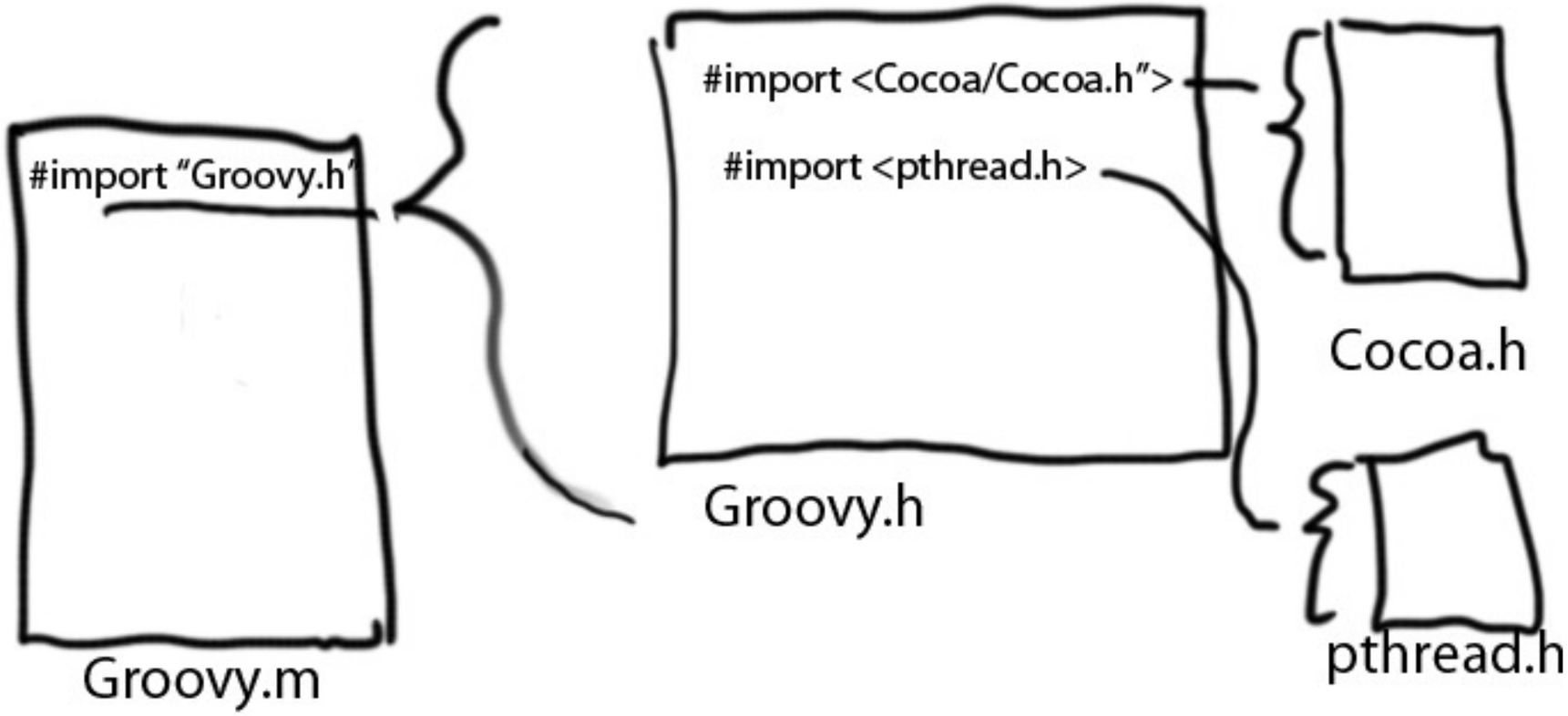
Mark Dalrymple - ~~Pittsburgh~~ ~~CocoaHeads~~

~~Atlanta!~~ CocoaConf

~~Raleigh!~~ DC!

The ‘umble ‘eader

```
#import “Groovy.h”
```



Why Are We Here?

- I'm an old curmudgeon
- One that's been cringing lately

Dude. They're just headers

What's Important?

- From a Popular Intro iPhone Book

```
#define kStateComponent    0
#define kZipComponent      1
```

Indexes for two sections of a UIPickerView

```
#import <UIKit/UIKit.h>

@interface WhateverViewController : UIViewController
@property (weak, nonatomic) IBOutlet UILabel *messageLabel;
...
- (void)updateLabelsFromTouches:(NSSet *)touches;

@end
```

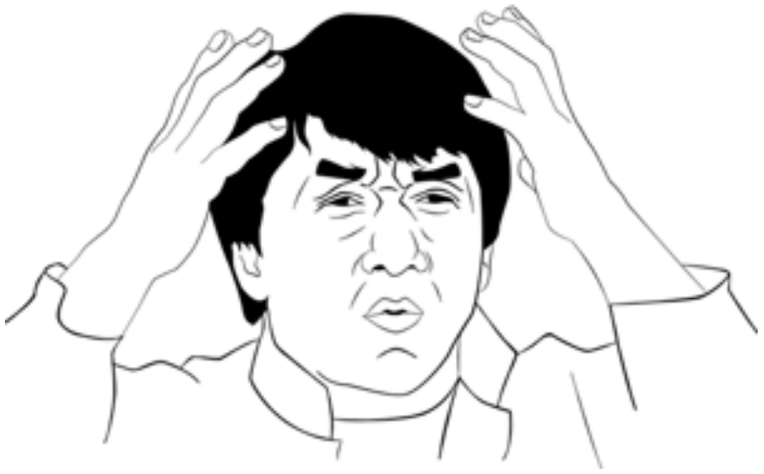
The Pain, The Pain

```
#import <UIKit/UIKit.h>
#import <CoreMotion/CoreMotion.h>

@interface ThingieBouncingBallView : UIView

@property (strong, nonatomic) UIImage *image;
@property (assign, nonatomic) CGPoint currentPoint;
@property (assign, nonatomic) CGPoint previousPoint;
@property (assign, nonatomic) CMAcceleration acceleration;
@property (assign, nonatomic) CGFloat ballXVelocity;
@property (assign, nonatomic) CGFloat ballYVelocity;

- (void)update;

@end // ThingieBouncingBallView
```

The Pain, The Pain

```
#import <UIKit/UIKit.h>
#import <CoreMotion/CoreMotion.h>

@interface ThingieBouncingBallView : UIView

@property (strong, nonatomic) UIImage *image;
@property (assign, nonatomic) CGPoint currentPoint;
@property (assign, nonatomic) CGPoint previousPoint;
@property (assign, nonatomic) CMAcceleration acceleration;
@property (assign, nonatomic) CGFloat ballXVelocity;
@property (assign, nonatomic) CGFloat ballYVelocity;

- (void)update;
@end // ThingieBouncingBallView
```

The Pain, The Pain



```
#import <UIKit/UIKit.h>
#import <CoreMotion/CoreMotion.h>

@interface ThingieBouncingBallView : UIView

@property (strong, nonatomic) UIImage *image;
@property (assign, nonatomic) CGPoint currentPoint;
@property (assign, nonatomic) CGPoint previousPoint;
@property (assign, nonatomic) CMAcceleration acceleration;
@property (assign, nonatomic) CGFloat ballXVelocity;
@property (assign, nonatomic) CGFloat ballYVelocity;

- (void)update;
@end // ThingieBouncingBallView
```

Where did they come from?

- From the early days of C
- Advertise interfaces to (static) libraries

What's the simplest thing we can do...

What are they *for*?

- Communication
  - With the compiler *how many bytes do I deal with?*
  - With other ugly giant bags of mostly water

    *what **is** this and how do I use it?*

Are they still useful?

- Non-interesting question for Fruit programmers
  - But ask Uncle Google for "llvm doug modules"
- Newer languages (Java / C#) look inside of libraries for the interfaces

and not-so-newer, like PL/I

Headers as Documentaton

- I like comments in header more than Header/ JavaDoc

I know, I'm weird

- Sometimes the headers are the only documentation
- Sometimes the best Apple docs are in the headers
- Secret trick - moby.h

My ideal header

```
#import <Foundation/Foundation.h>

@class GRIndoorCyclingDensitometer;

// Groovy is a class about ...
@interface Groovy : NSObject

// User's greeblation threshold, in milli-marklars
@property (copy, nonatomic) NSString *greeblation;

// UI stuff
@property (weak, nonatomic) IBOutlet UILabel *uberDisplayLabel;
- (IBAction) launchNukes: (id) sender;

// The user's marklar density can affect the sub-greeblation threshold.
- (void) setDdensitometer: (GRIndoorCyclingDensitometer *) denselWashington;

@end // Groovy
```

What are they **for**?

What are they **for**?

The public interface for the class

Headers for Unit Tests

GronkTest.h

```
#import <SenTestingKit/SenTestingKit.h>

@interface GronkTest : SenTestCase

@end
```

GronkTest.m

```
#import “GronkTest.h”

@implementation GronkTest

@end
```

More Testing

- Public and private headers for your class
- Public API into GroovyClass.h
  - Testing / Private API into GroovyClass-Private.h
    - Good place for visible class extensions for tests and subclasses

Breaking Cycles

.. Cycling Class.h

```
#import <Foundation/Foundation.h>

#import “GRIndoorCyclingRideSegment.h”

@interface GRIndoorCyclingClass : NSObject
...
- (GRIndoorCyclingRideSegment *) segmentAtIndex: (int) blah;
@end
```

.. Ride Segment.h

```
#import <Foundation/Foundation.h>

#import “GRIndoorCyclingClass.h”

@implementation GRIndoorCyclingRidesegment : NSObject
...
- (GRIndoorCyclingClass *) owningClass;

@end
```

Broken Cycles

.. Cycling Class.h

```
#import <Foundation/Foundation.h>

@class GRIndoorCyclingRideSegment;

@interface GRIndoorCyclingClass : NSObject
...
- (GRIndoorCyclingRideSegment *) segmentAtIndex: (int) blah;
@end
```

.. Ride Segment.h

```
#import <Foundation/Foundation.h>

@class GRIndoorCyclingClass;

@implementation GRIndoorCyclingRidesegment : NSObject
...
- (GRIndoorCyclingClass *) owningClass;

@end
```

Protocols and Headers

```
#import <Foundation/Foundation.h>
#import “SandwichChooserViewController.h”

@interface HappyMealTimeViewController
    : UIViewController <SandwichChooserViewControllerBreadDataSource>
...
@end
```

Requires Header

Protocols and Headers

```
#import <Foundation/Foundation.h>
#import “SandwichChooserViewController.h”

@interface HappyMealTimeViewController
    : UIViewController <SandwichChooserViewControllerBreadDataSource>
...
@end
```

Requires Header

```
#import <Foundation/Foundation.h>

@protocol SandwichChooserViewControllerBreadDataSource;

@interface HappyMealTimeViewController : NSObject {
    id <SandwichChooserViewControllerBreadDataSource> proxyDatasource;
}
...
@end
```

Forward Reference OK

Protocols and Headers

```
#import <Foundation/Foundation.h>
#import “SandwichChooserViewController.h”

@interface HappyMealTimeViewController
    : UIViewController <SandwichChooserViewControllerBreadDataSource>
...
@end
```

Requires Header

Can move into class extension

```
#import <Foundation/Foundation.h>

@protocol SandwichChooserViewControllerBreadDataSource;

@interface HappyMealTimeViewController : NSObject {
    id <SandwichChooserViewControllerBreadDataSource> proxyDatasource;
}
...
@end
```

Forward Reference OK

Class Extensions

```
#import “HappyMealTimeViewController.h”
#import “SandwichChooserViewController.h”

@interface HappyMealTimeViewController ()
    <SandwichChooserViewControllerBreadDataSource> {
    NSInteger _groovyFlonking;
}
...
@end // Extension


@implementation HappyMealTimeViewController
...
    SandwichChooserViewController *sammy = ...;
    sammy.breadDataSource = self;
...
@end
```

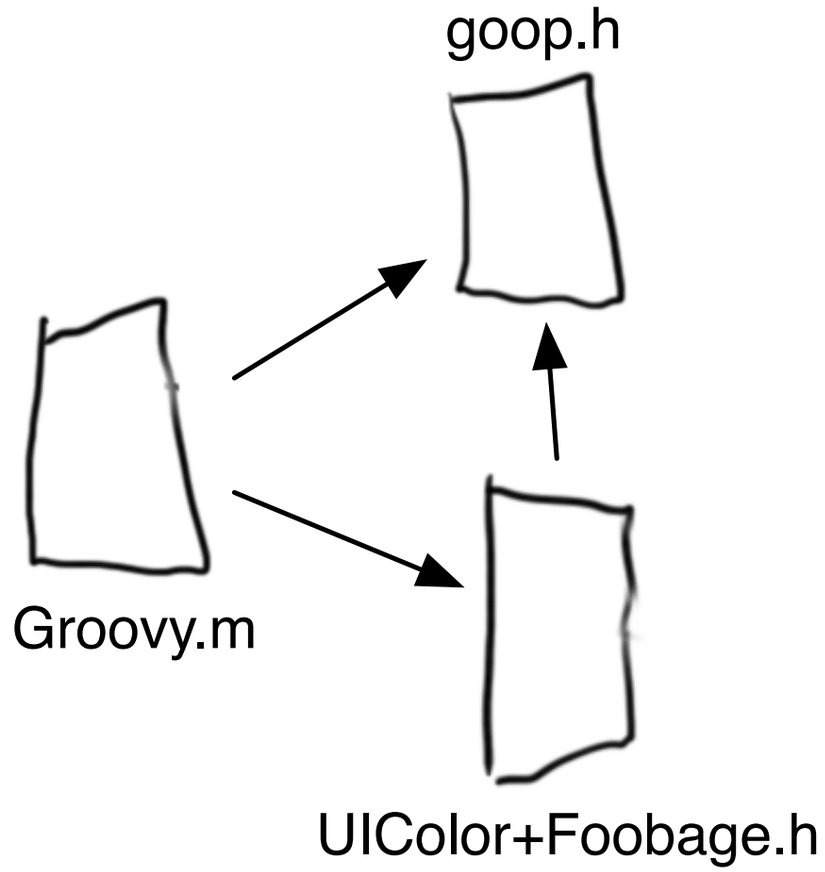
Multiple Inclusion

- #import guarantees one inclusion
- #import is officially deprecated by general gcc
  - Therefore a non-portable directive
  - (not that we really care)
- A file #included multiple times is processed multiple times.

This can be bad

Multiple Inclusion

- Groovy.m includes goop.h
- Groovy.m includes UIColor+Foobage.h, which includes goop.h



Multiple Inclusion

- Groovy.m includes goop.h
- Groovy.m includes UIColor+Foobage.h, which includes goop.h

```
typedef struct goop {
    int blah;
} goop;
```

goop.h

```
...
// From goop.h directly.
typedef struct goop {
    int blah;
} goop;
...
// From goop.h from UIColor+Foobage.h
typedef struct goop {
    int blah;
} goop;
...
```

Groovy.m, post-preprocessing

Yay! Errors!

```
error.m:7:16: error: redefinition of 'goop'
typedef struct goop {
               ^
error.m:3:16: note: previous definition is here
typedef struct goop {
               ^
1 error generated.
```

Include Guards

```
#ifndef GOOP_H_INCLUDED
#define GOOP_H_INCLUDED

typedef struct goop {
    int blah;
} goop;

#endif // GOOP_H_INCLUDED
```

Include Guards

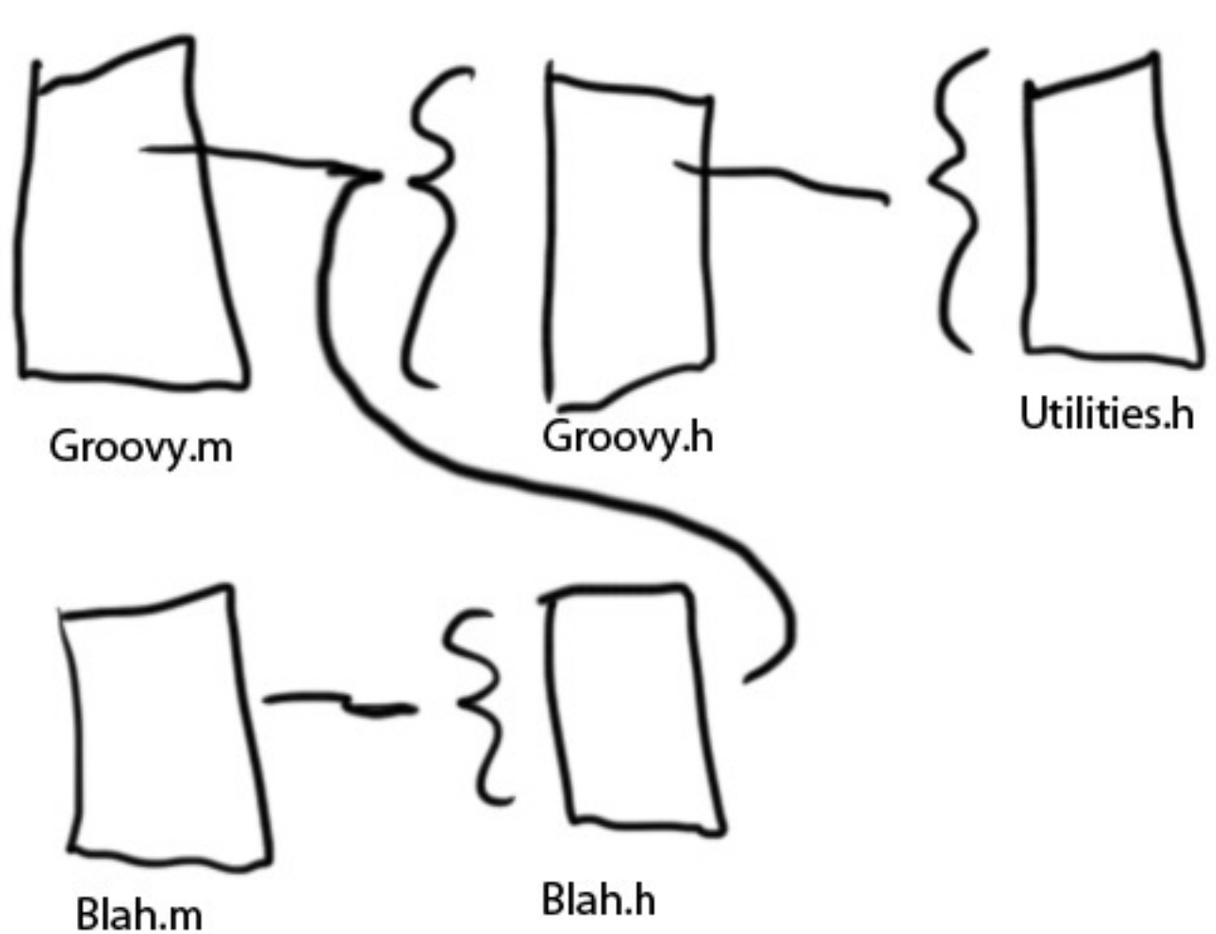
```
#ifndef GOOP_H_INCLUDED
#define GOOP_H_INCLUDED

typedef struct goop {
    int blah;
} goop;
```

```
#endif // GOOP_H_INCLUDED
```

```
// Copyright (c) 2009 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

#ifndef APP_CLIPBOARD_CLIPBOARD_H_
#define APP_CLIPBOARD_CLIPBOARD_H_

#include <map>
#include <string>
#include <vector>
```

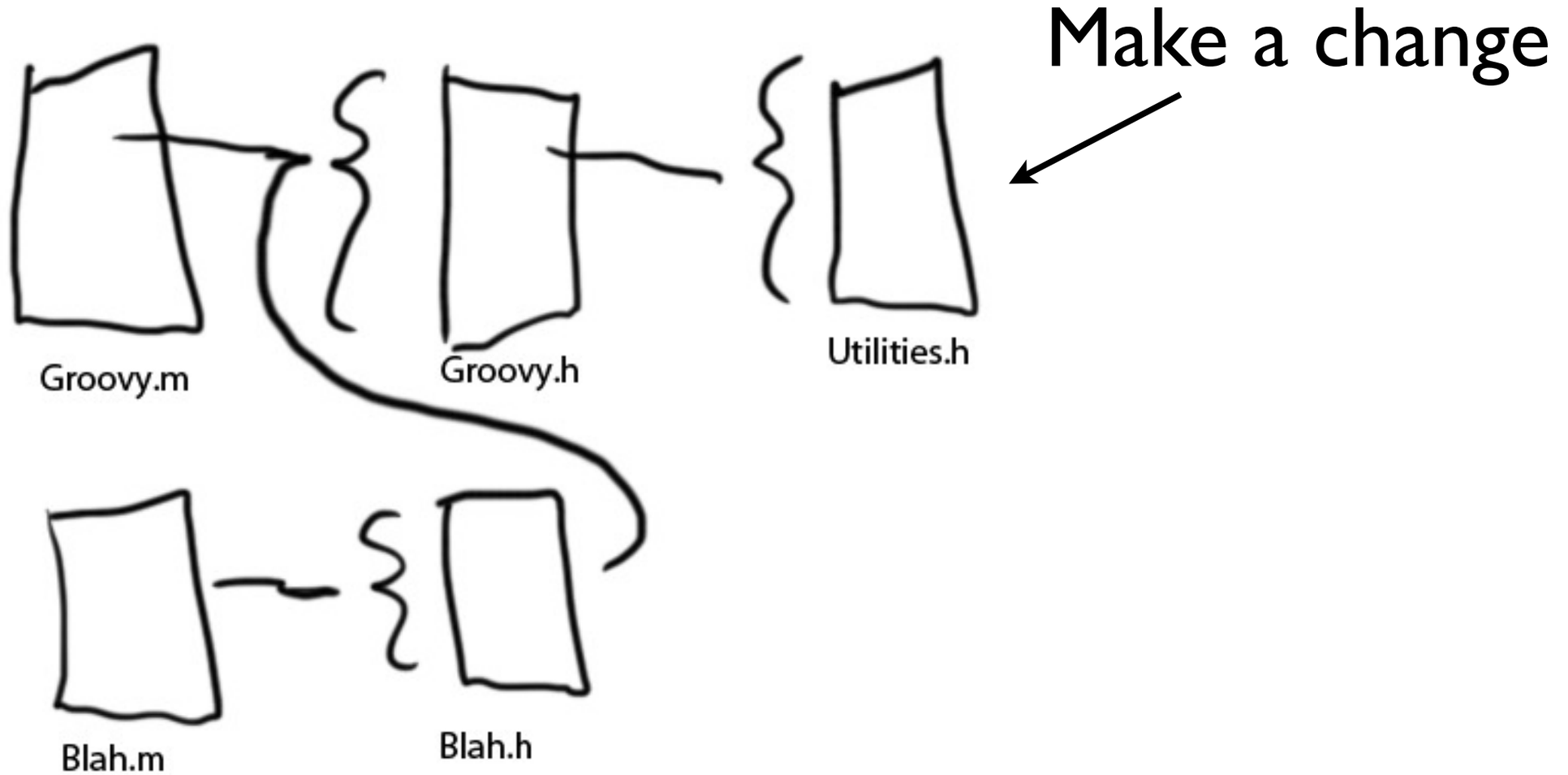
Compile Time Dependencies

- You pay a price for including headers you don't need



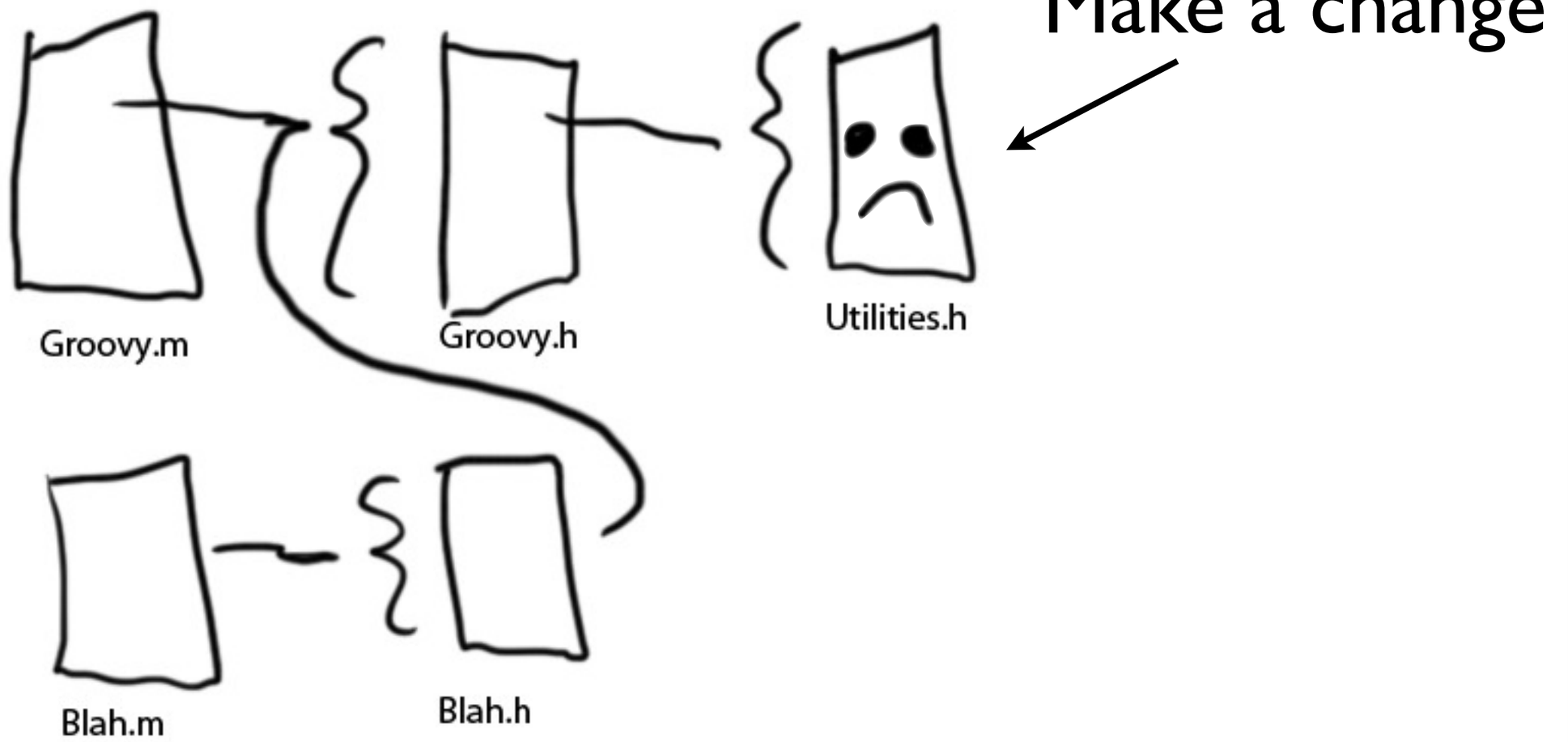
Compile Time Dependencies

- You pay a price for including headers you don't need



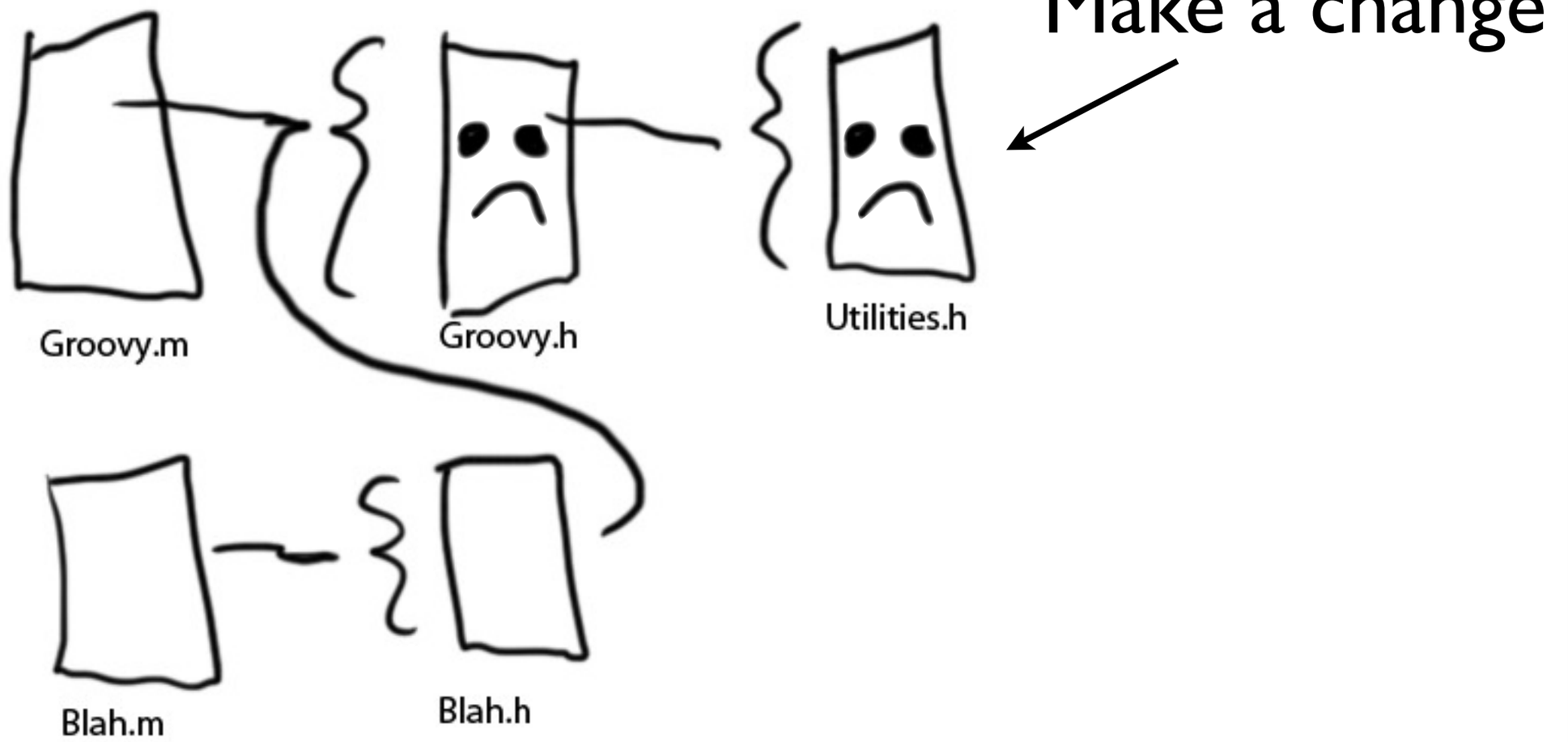
Compile Time Dependencies

- You pay a price for including headers you don't need



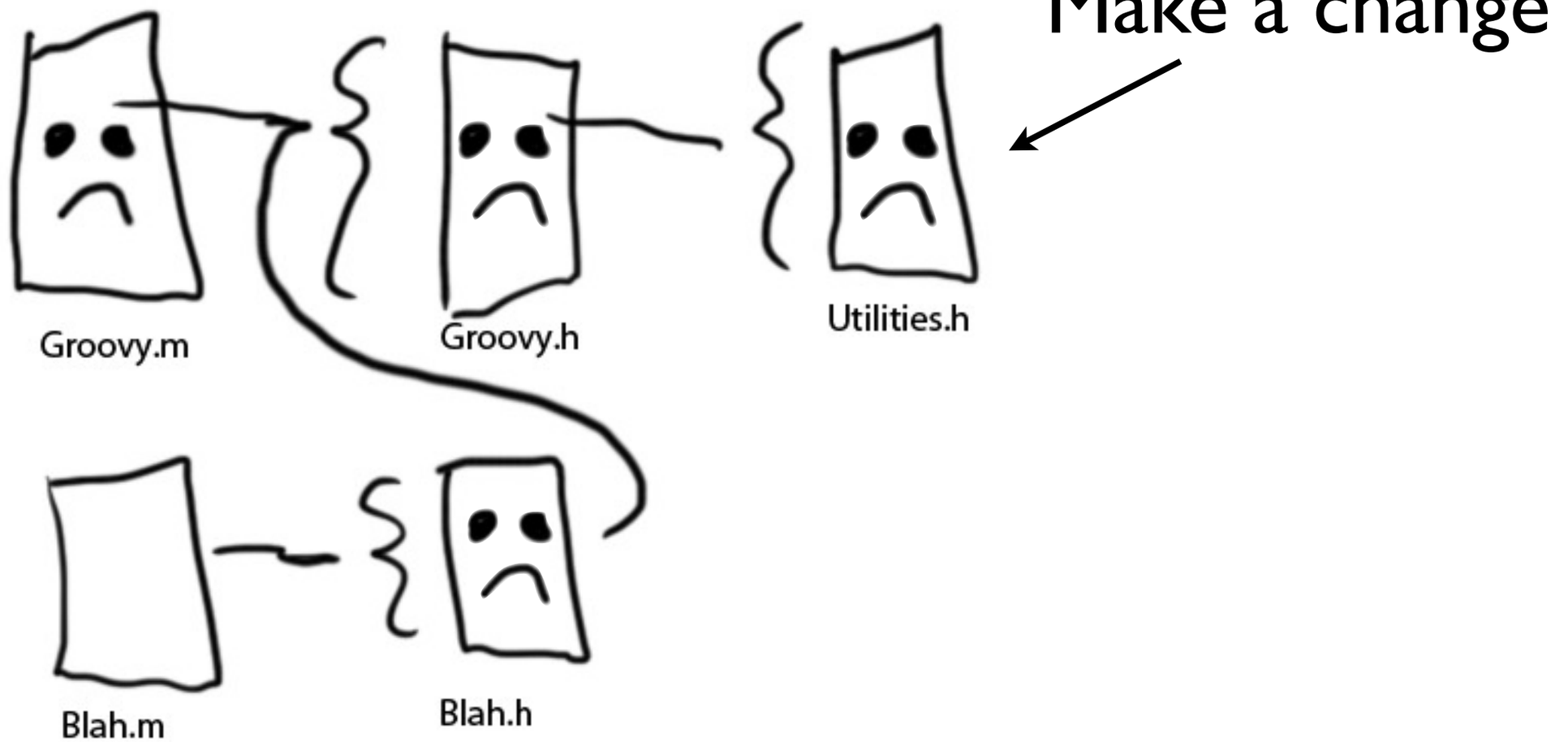
Compile Time Dependencies

- You pay a price for including headers you don't need



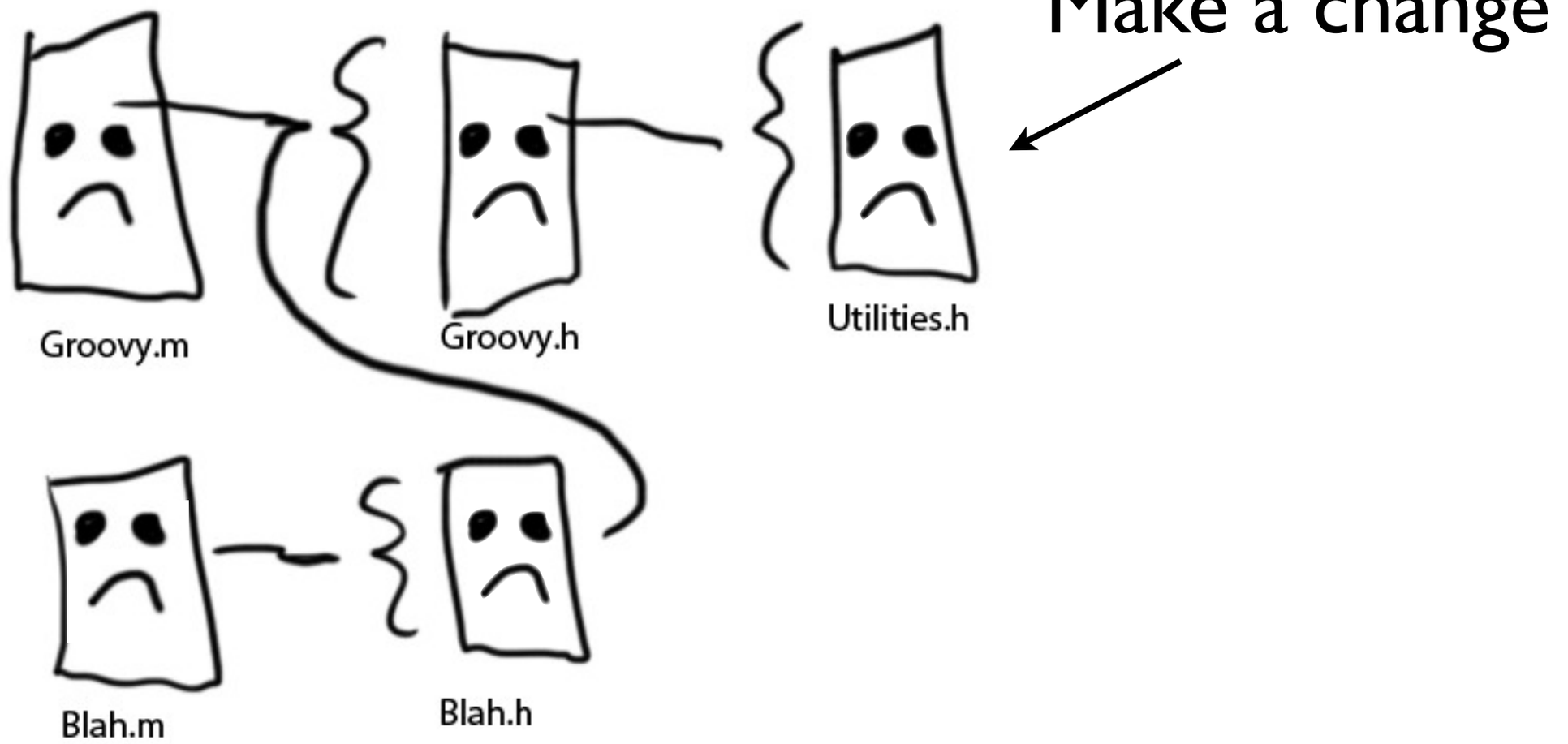
Compile Time Dependencies

- You pay a price for including headers you don't need



Compile Time Dependencies

- You pay a price for including headers you don't need



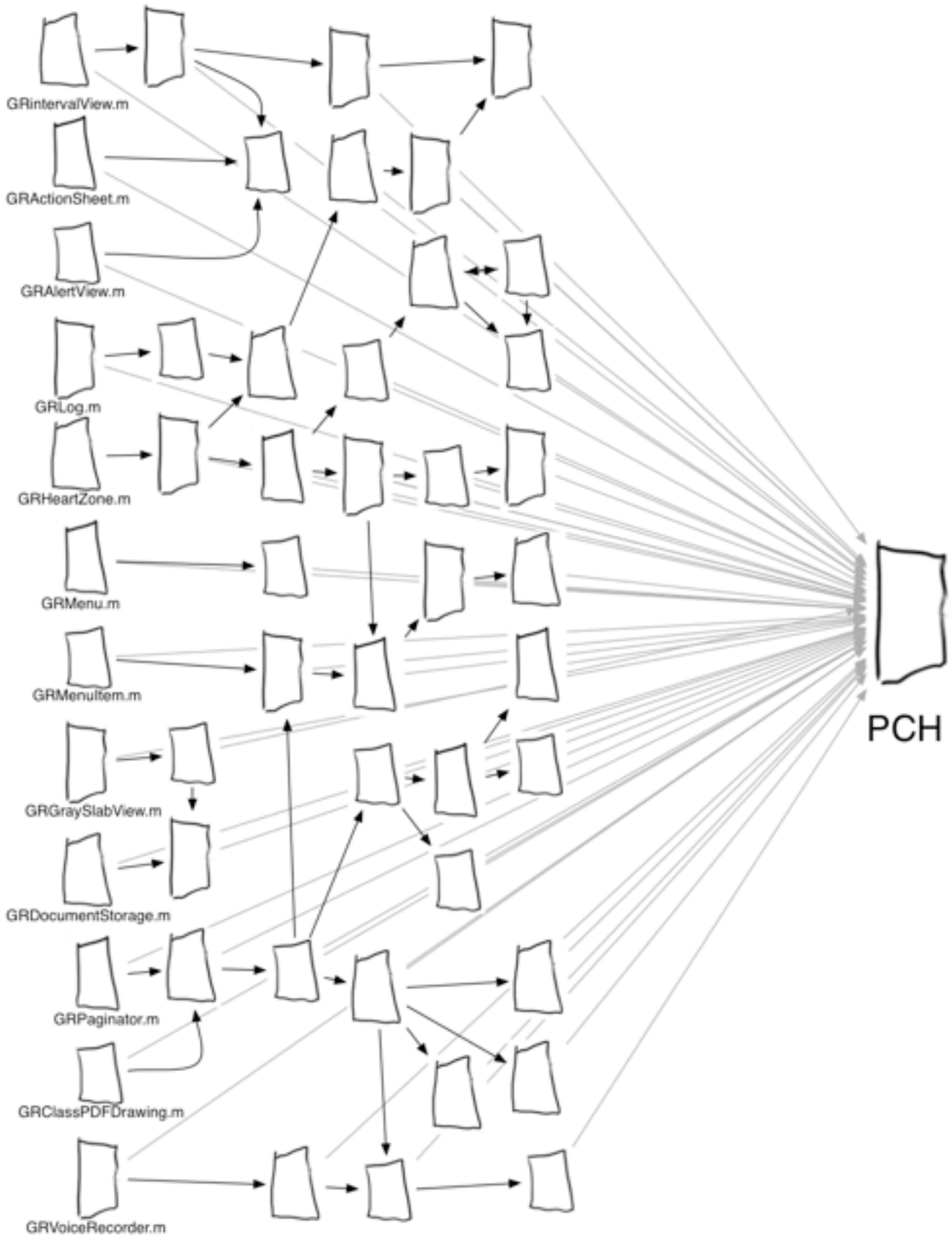
Precompiled Headers

```
//
// Prefix header for all source files of the 'TinyPix' target
// in the 'TinyPix' project
//

#import <Availability.h>

#ifndef __IPHONE_5_0
#warning "This project uses features only available in iOS SDK 5.0 and later."
#endif

#ifdef __OBJC__
    #import <UIKit/UIKit.h>
    #import <Foundation/Foundation.h>
#endif
```

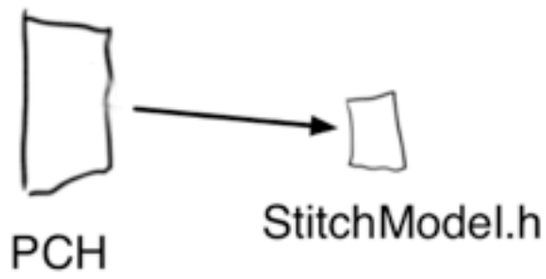
Why is this a problem?

PCH

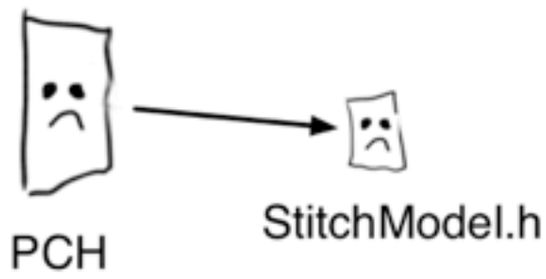
Why is this a problem?



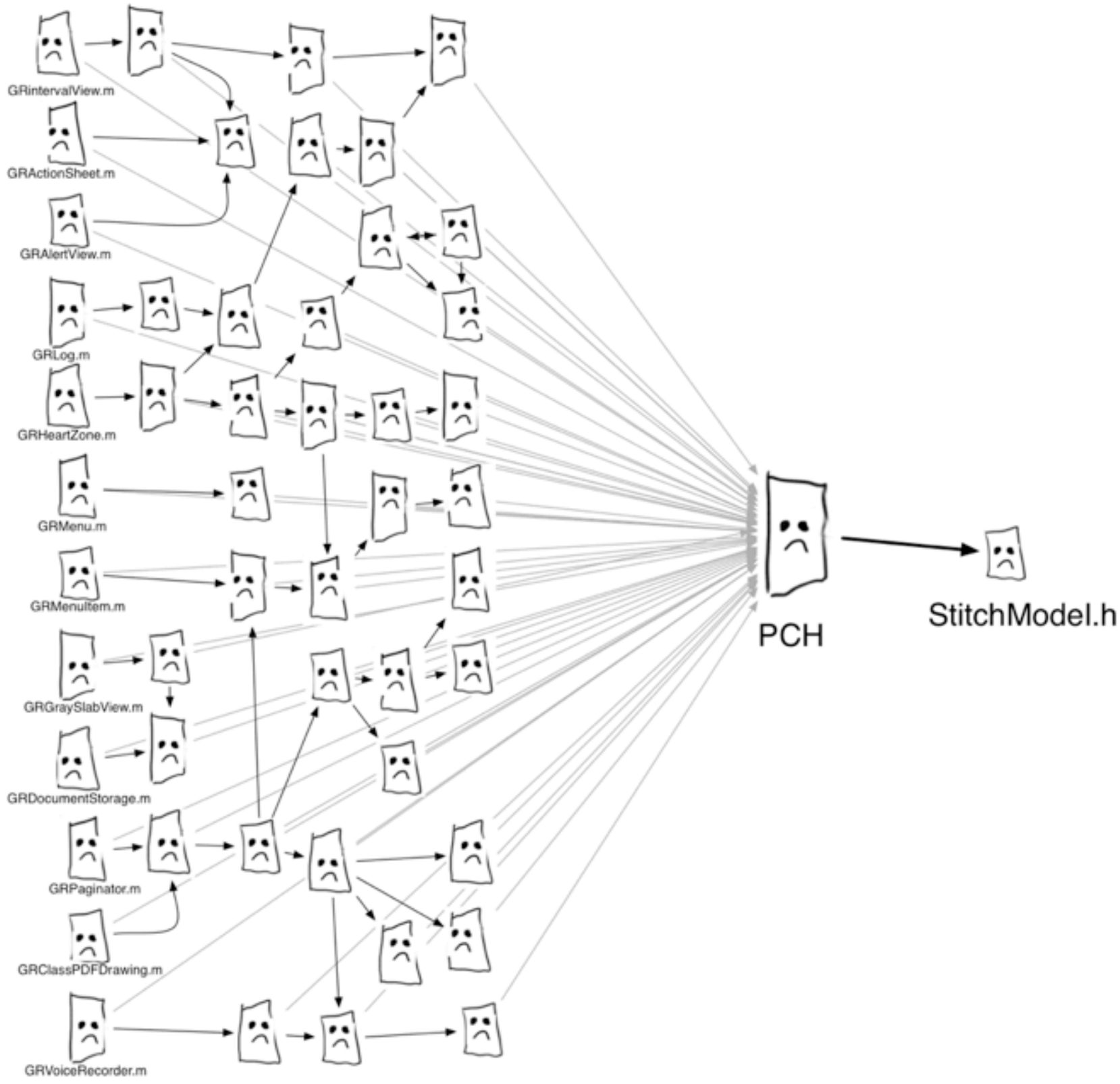
Why is this a problem?



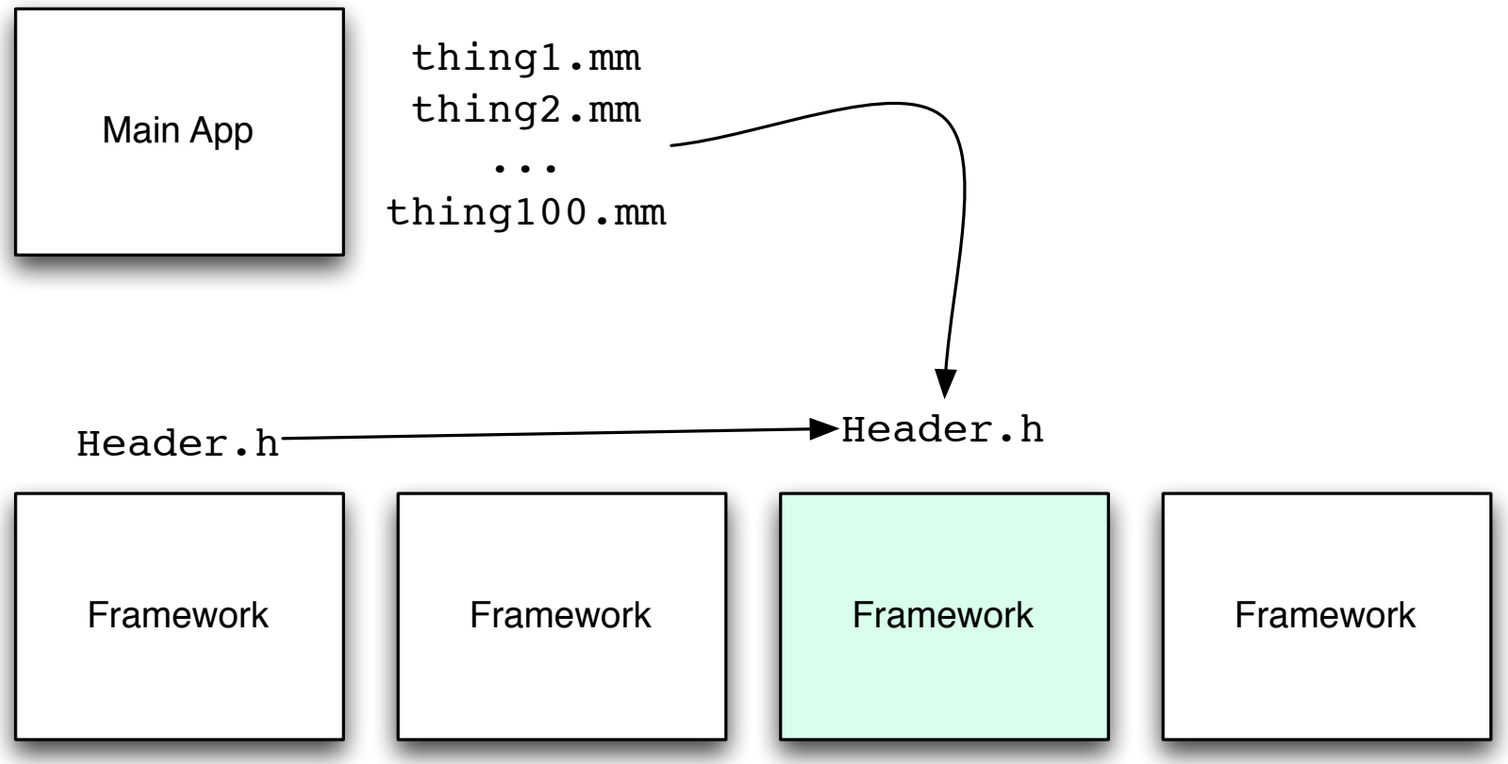
Why is this a problem?



Why is this a problem?



Not Just Academic



Order of #imports

```
// Header for this class first
#import "Groovy.h"

// Any system headers
#import <AVFoundation/AVFoundation.h>
#import <pthread.h>

// Other project headers
#import "GRFroopyViewControllerContollerView.h"
#import "GRMarklar.h"
#import "GRUtilities.h"

@implementaton Groovy
...
```

Groovy.m

Order of #imports

```
// Header for this class first
#import "Groovy.h"

// Any system headers
#import <AVFoundation/AVFoundation.h>
#import <pthread.h>

// Other project headers
#import "GRFroopyViewControllerContollerView.h"
#import "GRMarklar.h"
#import "GRUtilities.h"

@implementaton Groovy
...
```

Groovy.m

Unintentional breakage

```
#import <UIKit/UIKit.h>
@interface Groovy : NSObject
- (void) setPlayer: (AVAudioPlayer *) player;
@end
```

Groovy.h

Unintentional breakage

```
#import <UIKit/UIKit.h>
@interface Groovy : NSObject
- (void) setPlayer: (AVAudioPlayer *) player;
@end
```

Groovy.h

```
#import <AVFoundation/AVFoundation.h>
#import “Groovy.h”
```

Groovy.m

Unintentional breakage

```
#import <UIKit/UIKit.h>
@interface Groovy : NSObject
- (void) setPlayer: (AVAudioPlayer *) player;
@end
```

Groovy.h

```
#import <AVFoundation/AVFoundation.h>
#import “Groovy.h”
```

Groovy.m

```
#import “Groovy.h”
```

Snood.m

```
Compiler says “Dude!  What is this AVAudioPlayer thing?
```

Habits to Break

```
#import <Cocoa/Cocoa.h>

@interface Groovy : NSObject {
@private
  int this;
@protected
  int that;
@public
  int theOther;
}

...
// Private member functions
- (void) updateUI;
- (void) tweakTableView;

// Overrides
- (void) dealloc;
- (void) tableView:numberOfRowsInSection:

// Actually useful stuff
- (void) gronk;
@end
```

Boilerplate! Lame!

```
/* FUNCTION NAME:
 * ARGUMENTS:
 * RETURN:
 * DESCRIPTION:
* SIDE-EFFECTS:
* CAVEATS:
* BY SOMELOSER ON 10/15/90
*/
```

Xcode Header Templates

```
//
//  BIDAppDelegate.h
//  Borkinator
//
//  Created by markd on 10/14/11.
//  Copyright (c) 2011 __MyCompanyName__. All rights reserved.
//
```

#endif

- Headers are for communication
- Keep them simple and minimal
- Put everything else into the implementation