

ASYNC VIEWCONTROLLERS

Tom Adriaenssen

@inferis :: <http://inferis.org>

CONTROLLERS

WIFIAW



ASWING

ANATOMY OF ASYNC DATA

1. start slow call
2. show *hey, keep yer pants on* indicator
3. wait for call to finish
4. remove *pants* indicator
5. show your actual data

**SOME
HISTORY**



VANHECKE PIETER ▾

0321203

...

Q apple

✖ Cancel

APPLE INC. Share
AAPL Usa >

GOLDEN APPLE OIL & GAS INC Share
GAPJ Usa >

APPLE INC. Share
APC Frankfurt >

APPLE RUSH COMPANY INC Share
APRU Usa >

APPLE GREEN HOLDIN COM USD0.0001 Share
BLES Usa >

APPLE 7.7STN1014Z Certificate
32E4Z Euronext Amsterdam >

ING SPRINTER SHORT (APPL F) Turbo

Q W E R T Y U I O P

A S D F G H J K L

↑ Z X C V B N M ⌂

123



space

Search

VANHECKE PIETER ▾

0321203

...

apple

✖ Cancel

APPLE INC. Share >
AAPL Usa

GOLDEN APPLE OIL & GAS INC Share >
GAPJ Usa

APPLE INC. Share >
APC Frankfurt

APPLE RUSH COMPANY INC Share >
APRU Usa

APPLE GREEN HOLDIN COM USD0.0001 Share >
BLES Usa

APPLE 7.7STN1014Z Certificate >
32E4Z Euronext Amsterdam

ING SPRINTER SHORT (APPLE) Turbo

Q W E R T Y U I O P

A S D F G H J K L

⬆ Z X C V B N M ⬅

123



space

Search

**MASSIVE
VIEW
CONTROLLER**

MASSIVE VIEW CONTROLLER

IPHONE - 1.0

// Lines of code, including whitespace

- * Hotspot:
 - * HSVViewController .h: 25, .m: 716
 - * HSBoxController .h: 44, .m: 370
- * Portfolio: .h: 14, .m: 523



Search

Home

VANHECKE PIETER
0321203

https://platformaccept.bolero.be/v1.1/api



apple



APPLE INC.

AAPL

Share
Usa >

GOLDEN APPLE OIL & GAS INC

GAPJ

Share
Usa >

APPLE INC.

APC

Share
Frankfurt >

APPLE RUSH COMPANY INC

APRU

Share
Usa >

APPLE GREEN HOLDIN COM USD0.0001

BLES

Share
Usa >

Pending orders (4)

Period: 1 year ▾



Analysis & Insights (66)

Show all results

> Latest news >

Q

W

E

R

T

Y

U

I

O

P



A

S

D

F

G

H

J

K

L

return



Z

X

C

V

B

N

M

!

?



123



123



ACCEPT (Debug) -S:7->7- 09:41 100% https://platformaccept.bolero.be/v1/api

Search Home VANHECKE PIETER 0321203

apple

ending orders (4)

Period: 1 year

APPLE INC. AAPL Share Usa >

GOLDEN APPLE OIL & GAS INC GAPJ Share Usa >

APPLE INC. APC Share Frankfurt >

APPLE RUSH COMPANY INC APRU Share Usa >

APPLE GREEN HOLDIN COM USDO.0001 BLES Share Usa >

Analysis & Insights (66) Show all results > Latest news >

Q W E R T Y U I O P X return

A S D F G H J K L

Z X C V B N M ! ? .

123

123

The screenshot displays a mobile application interface for financial or market data. At the top, a navigation bar includes the time (09:41), battery status (100%), and a URL (https://platformaccept.bolero.be/v1/api). Below the bar, the word "apple" is typed into a search field, and a dropdown menu lists five companies: "APPLE INC. AAPL Share Usa >", "GOLDEN APPLE OIL & GAS INC GAPJ Share Usa >", "APPLE INC. APC Share Frankfurt >", "APPLE RUSH COMPANY INC APRU Share Usa >", and "APPLE GREEN HOLDIN COM USDO.0001 BLES Share Usa >". To the right of the search results is a line chart showing price movement from July to March, with a notable upward trend and a circled point in November. Below the chart, a blue bar contains the text "Analysis & Insights (66)" and "Show all results". Further down, a "Latest news" section is visible. A virtual keyboard is overlaid on the screen, with letters Q through P on the first row and A through L on the second row. The third row contains punctuation marks (!, ?, .) and a return key. The fourth row contains symbols for uppercase (Z, X, C, V, B, N, M), punctuation (!, ?), and a backspace key. The fifth row contains numeric keys (123) and a keyboard icon.

TECHNICAL DEBT

**BREAK
THINGS
UP**

INTRODUCING ASYNCVIEWCONTROLLER



```
@interface AsyncViewController : UIViewController  
  
@property (nonatomic, strong, readonly)  
    UIView<AsyncView> *asyncView;  
@property (nonatomic, strong, readonly)  
    UIView<AsyncStatusView> *statusView;  
  
- (void)performAsyncDataRequest;  
  
@end
```

Y U NO SWIFT?

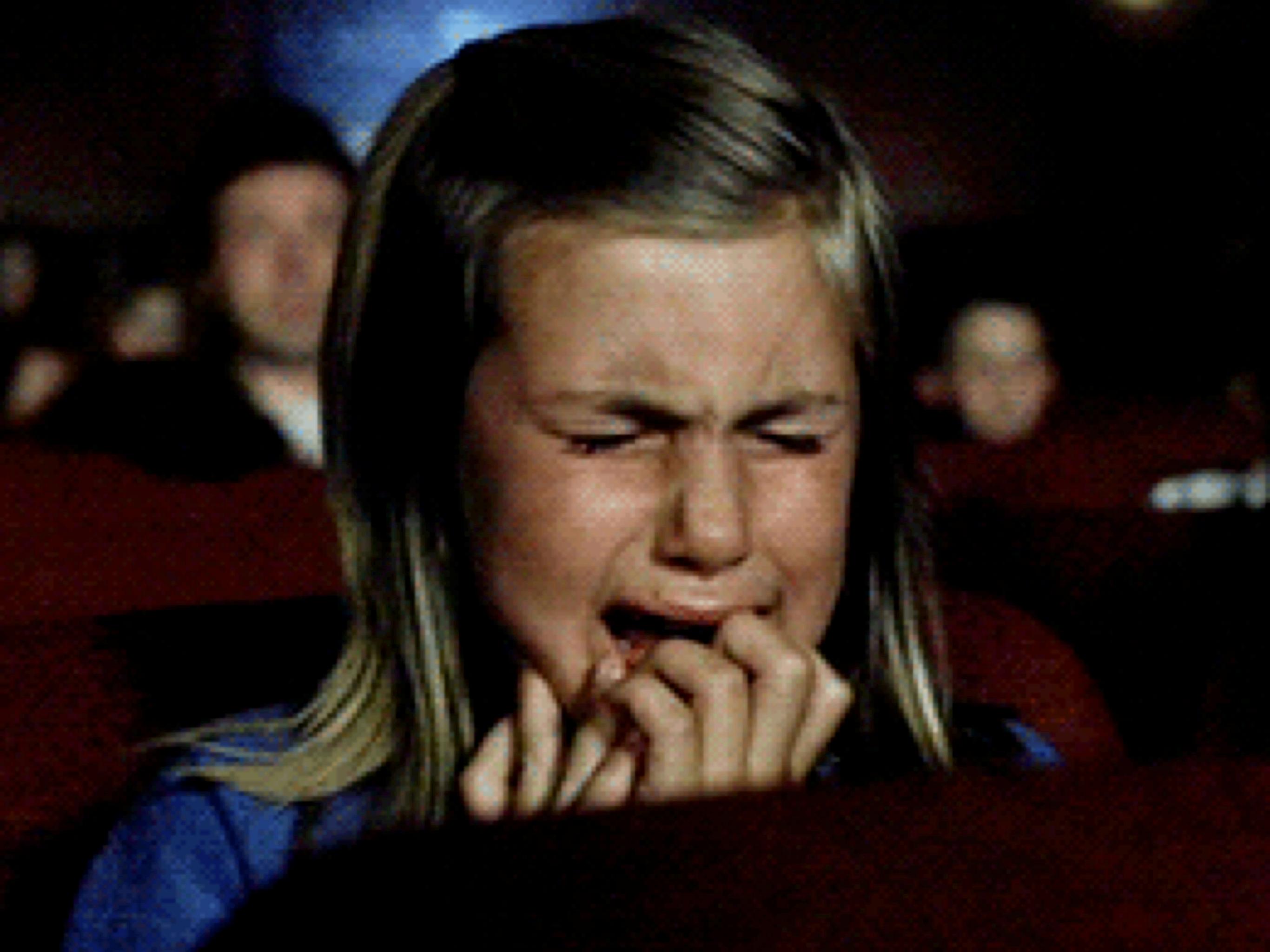


```
@property (nonatomic, strong) UIView<IIAsyncView> *asyncView;
```

UIView<IIAsyncView> *

*@Inferis None yet. You can
define a generic parameter <T:
BaseClass where T: Protocol>,
but that isn't representable as a
type yet.*

– @jckarter



rdar://20990743

A person is sitting at a desk, facing a computer monitor. They are wearing a dark long-sleeved shirt. On the desk, there is a black keyboard and a black mouse. The background is a light-colored wall.

rdar://20990743

```
@interface AsyncViewController : UIViewController  
  
@property (nonatomic, strong, readonly)  
    UIView<AsyncView> *asyncView;  
@property (nonatomic, strong, readonly)  
    UIView<AsyncStatusView> *statusView;  
  
- (void)performAsyncDataRequest;  
  
@end
```

SUBCLASS OF
UIViewController

```
@property (nonatomic, strong, readonly)  
UIView<AsyncView> *asyncView;
```

```
@property (nonatomic, strong, readonly)  
UIView<AsyncStatusView> *statusView;
```

- `performAsyncDataRequest`

ASYNC DATA FETCH

&

STATUSVIEW + ASYNCVIEW

&

VIEW CONTROLLER CONTAINMENT

SRP

A MORE PURE

W W W W G

EASE OF USE

FLEXIBLE

TESTABLE



```
@protocol IIAsyncView <NSObject>

@required
@property (nonatomic, strong, readonly) id<AsyncData> asyncData;

- (void)asyncDataApplyValueAnimated:(BOOL)animated;

@optional
// ...

@end
```

```
@protocol IIAsyncData <NSObject>
```

```
@required
```

```
@property (nonatomic, assign, readonly) BOOL loading;
```

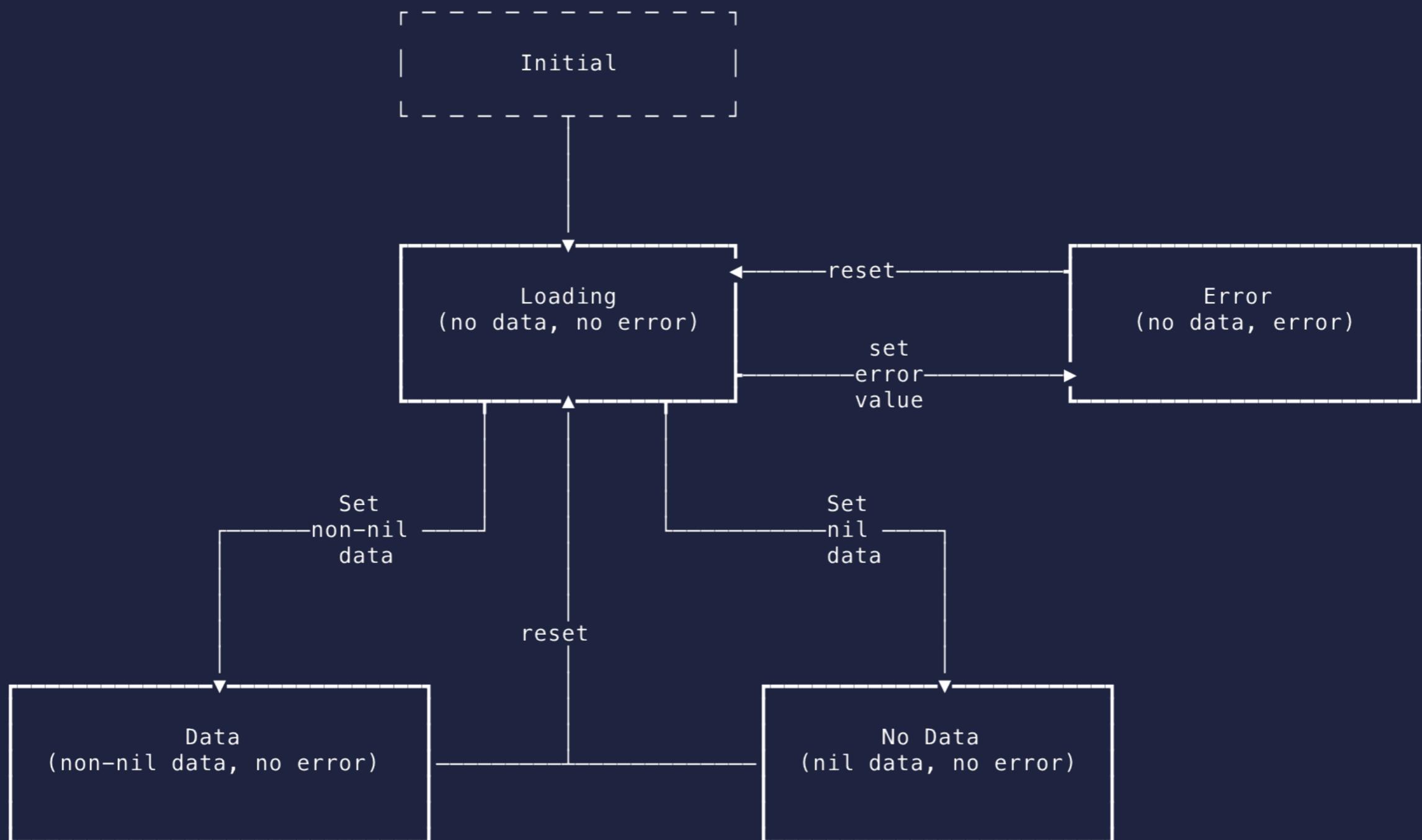
```
@property (nonatomic, strong) NSError *error;
```

```
@property (nonatomic, strong) id value;
```

```
- (void)reset;
```

```
@end
```

```
typedef NS_ENUM(NSUInteger, AsyncStatusViewState) {  
    IIAsyncStatusLoading,  
    IIAsyncStatusError,  
    IIAsyncStatusNoData,  
    IIAsyncStatusData  
};
```



```
- (AsyncStatusViewState)determineState
{
    // determine the new state
    if (![self isAsyncViewLoaded] || [self.asyncView.asyncData isLoading]) {
        return AsyncStatusLoading;
    }
    else if (self.asyncView.asyncData.error) {
        return AsyncStatusError;
    }
    else if (self.asyncView.asyncData.value) {
        return AsyncStatusData;
    }
    else {
        return AsyncStatusNoData;
    }
}
```

```
@protocol IIAsyncData <NSObject>

@required
@property (nonatomic, assign, readonly) BOOL loading;
@property (nonatomic, strong) NSError *error;
@property (nonatomic, strong) id value;
@property (nonatomic, weak) id<IIAsyncDataDelegate> asyncDataDelegate;

- (void)reset;
- (void)invalidateState;

@end
```

```
@protocol IIAsyncDataDelegate <NSObject>

@required
- (void)asyncData:(id<IIAsyncData>)data
    didInvalidateStateForced:(BOOL)forced;
- (void)reloadAsyncData;

@end
```

```
@protocol AsyncView <NSObject>
```

```
@required
```

```
// ...
```

```
@optional
```

- (id)asyncNoDataMessage;
- (BOOL)asyncCanReload;

```
@end
```

```
@protocol AsyncStatusView <NSObject>

@required
@property (nonatomic, strong) UIView<IIAsyncView> *asyncView;

- (void)transitionToLoadingStateAnimated:(BOOL)animated;
- (void)transitionToErrorState:(NSError*)error animated:(BOOL)animated;
- (void)transitionToNoDataStateAnimated:(BOOL)animated;
- (void)transitionToDataStateAnimated:(BOOL)animated;

@end
```

```
@interface AsyncData : NSObject <AsyncData>
```

```
@interface AsyncView : UIView <AsyncView>
```

```
@interface AsyncStatusView : NSObject <AsyncStatusView>
```

```
@interface AsyncViewController : UIViewController
```

```
@interface IIAsyncViewController : UIViewController  
- (void)reloadAsyncData;  
  
- (UIView<IIAsyncStatusView>*)loadStatusView;  
  
- (void)willTransitionToNewStateAnimated:(BOOL)animated;  
- (void)didTransitionToLoadingStateAnimated:(BOOL)animated;  
- (void)didTransitionToErrorStateAnimated:(BOOL)animated;  
- (void)didTransitionToDataStateAnimated:(BOOL)animated;  
- (void)didTransitionToNoDataStateAnimated:(BOOL)animated;  
  
@end
```

EXAMPLE

```
@interface SuchViewController ()  
  
@property (nonatomic, strong) LaggyBackend *backend;  
  
@end  
  
@implementation SuchViewController  
  
- (void)performAsyncDataRequest  
{  
    [self.backend thatReallySlowCall:^(id result) {  
        self.asyncView.asyncData.value = result;  
    } error:^(NSError* error) {  
        self.asyncView.asyncData.error = error;  
    }];  
}  
  
@end
```

```
@interface SuchView ()  
  
@property (nonatomic, strong) IBOutlet UILabel *label;  
  
@end  
  
@implementation SuchView  
  
- (void)asyncDataApplyValueAnimated:(BOOL)animated  
{  
    self.label.text = [NSString stringWithFormat:  
                      @"Such %@", self.asyncData.value];  
}  
  
- (BOOL)asyncCanReload  
{  
    return self.asyncData.error != nil;  
}
```

**UNDER THE
HOOD**

```
// In AsyncViewController

- (void)viewDidLoad
{
    [self wrapWithStatusView];

    // call regular viewDidLoad
    [super viewDidLoad];

    // force an update of the state to loading
    [self updateState:YES];

    [self performAsyncDataRequest];
}
```

```
- (void)wrapWithStatusView
{
    // don't wrap if there's a status view already
    if (self.statusView) return;

    // get async view. This is just self.view when first wrapping.
    UIView<IIAsyncView> *asyncView = (UIView<IIAsyncView>*)self.view;

    // now create the status view
    UIView<IIAsyncStatusView> *statusView = [self loadStatusView];
    statusView.frame = asyncView.frame;
    statusView.asyncView = asyncView;
    asyncView.asyncData.asyncDataDelegate = self;

    // make main view the wrapping view
    [super setView:statusView];
}
```

STOP

*hammer
TIME!*

```
@interface UIViewController : NSObject <UILayoutSupport>
```

```
// These objects may be used as layout items  
// in the NSLayoutConstraint API
```

```
@property(nonatomic,readonly,strong)
```

```
    id<UILayoutSupport> topLayoutGuide;
```

```
@property(nonatomic,readonly,strong)
```

```
    id<UILayoutSupport> bottomLayoutGuide;
```

```
@end
```

Reaction



1. find constraints related to layout guides before wrapping
2. wrap views
3. first layout pass: recreate constraints

```
- (NSArray*)findLayoutGuideConstraints:(UIView*)view
{
    NSMutableArray *constraints = [NSMutableArray new];

    UIViewController *viewController = [self findViewDelegate:view];
    if (!viewController) return;

    for (NSLayoutConstraint *constraint in view.constraints) {
        if (constraint.class != [NSLayoutConstraint class]) continue;

        if (constraint.firstItem == viewController.topLayoutGuide ||
            constraint.secondItem == viewController.topLayoutGuide) {
            [constraints addObject:@[constraint, @YES, viewController.topLayoutGuide]];
        }
        else if (constraint.firstItem == delegate.bottomLayoutGuide ||
                  constraint.secondItem == delegate.bottomLayoutGuide) {
            [constraints addObject:@[constraint, @NO, viewController.bottomLayoutGuide]];
        }
    }

    return [constraints copy];
}
```

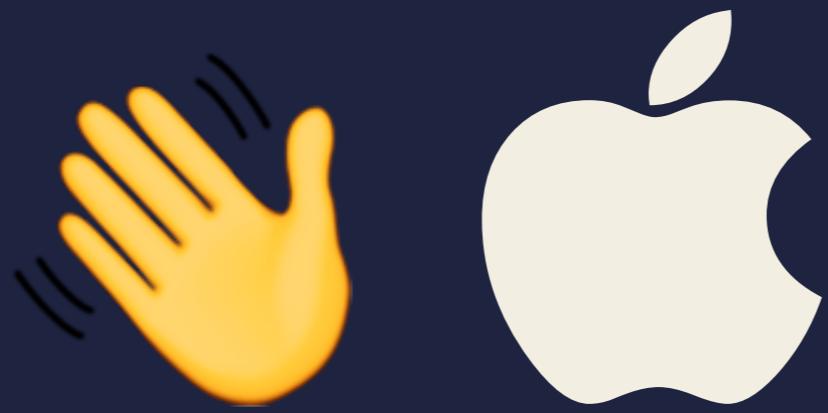
- (**UIViewController***)findViewDelegate:(**UIView***)view

```
- (UIViewController*)findViewDelegate:(UIView*)view
{
    const char *iVarName = [[NSString stringWithFormat:@"_%@", @"view"] UTF8String];

    Ivar iVar = class_getInstanceVariable(view.class, iVarName);

    UIViewController *delegate = nil;
    if (iVar) {
        delegate = object_getIvar(view, iVar);
        if ([delegate isKindOfClass:[UIViewController class]]) {
            return delegate;
        }
    }

    return nil;
}
```



```
- (void)reapplyLayoutGuideConstraints:(NSArray*)constraintInfos
{
    UIViewController *viewController = [self findViewDelegate:self];
    if (!viewController) return;

    for (NSArray *constraintInfo in constraintInfos) {
        NSLayoutConstraint *oldConstraint = constraintInfo[0];
        BOOL topGuide = [constraintInfo[1] boolValue];

        // replace first and/or second item with new guides
        id firstItem = oldConstraint.firstItem == constraintInfo[2]
            ? (topGuide ? viewController.topLayoutGuide : viewController.bottomLayoutGuide)
            : oldConstraint.firstItem;
        id secondItem = oldConstraint.secondItem == constraintInfo[2]
            ? (topGuide ? viewController.topLayoutGuide : viewController.bottomLayoutGuide)
            : oldConstraint.secondItem;

        // Create new constraint
        NSLayoutConstraint *constraint =
            [NSLayoutConstraint constraintWithItem:firstItem
                attribute:oldConstraint.firstAttribute
                relatedBy:oldConstraint.relation
                toItem:secondItem
                attribute:oldConstraint.secondAttribute
                multiplier:oldConstraint.multiplier
                constant:oldConstraint.constant];
        [self addConstraint:constraint];
    }
}
```

A dense grid of binary code (0s and 1s) arranged to spell out the word "DATA". The letters are formed by clusters of 1s against a background of 0s. The "D" is on the left, "A" is in the center, "T" is on the right, and "A" is below it. The binary digits are packed closely together, creating a high-resolution digital pattern.

DETAILS

MASSIVE VIEW CONTROLLER A REPRISE

MASSIVE VIEW CONTROLLER

IPHONE - 1.0

// Lines of code, including whitespace

- * Portfolio: .h: 14, .m: 523
- * Hotspot:
 - * HSVViewController .h: 25, .m: 716
 - * HSBoxController .h: 44, .m: 370

MASSIVE VIEW CONTROLLER

IPHONE - 1.2

// Lines of code, including whitespace

- * Portfolio: .h: 16, .m: 373
- * Hotspot:
 - * HSVViewController: .h: 28, .m: 539
 - * HSBoxController: 41.h, .m: 108

MASSIVE VIEW CONTROLLER

IPHONE - 1.0 VS 1.2

- * Portfolio: .h: 114%, .m: 71%
- * Hotspot:
 - * HSVViewController: .h: 112%.h, .m: 75%
 - * HSBoxController: .h: 93%.h, .m: 29%

-＼(ツ)／-

TAKEAWAY

1. embrace View Controller Containment
2. reduce dependencies
3. avoid boilerplate
4. DRY
5. testable

OPEN SOURCE

Reference implementation:

<http://github.com/inferis/IIASyncViewController>

THANKS!

Get in touch!

@inferis

<http://blog.inferis.org>