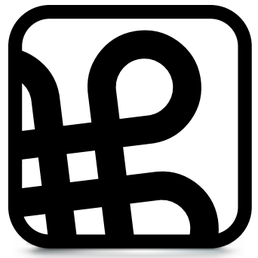


Macoun’11

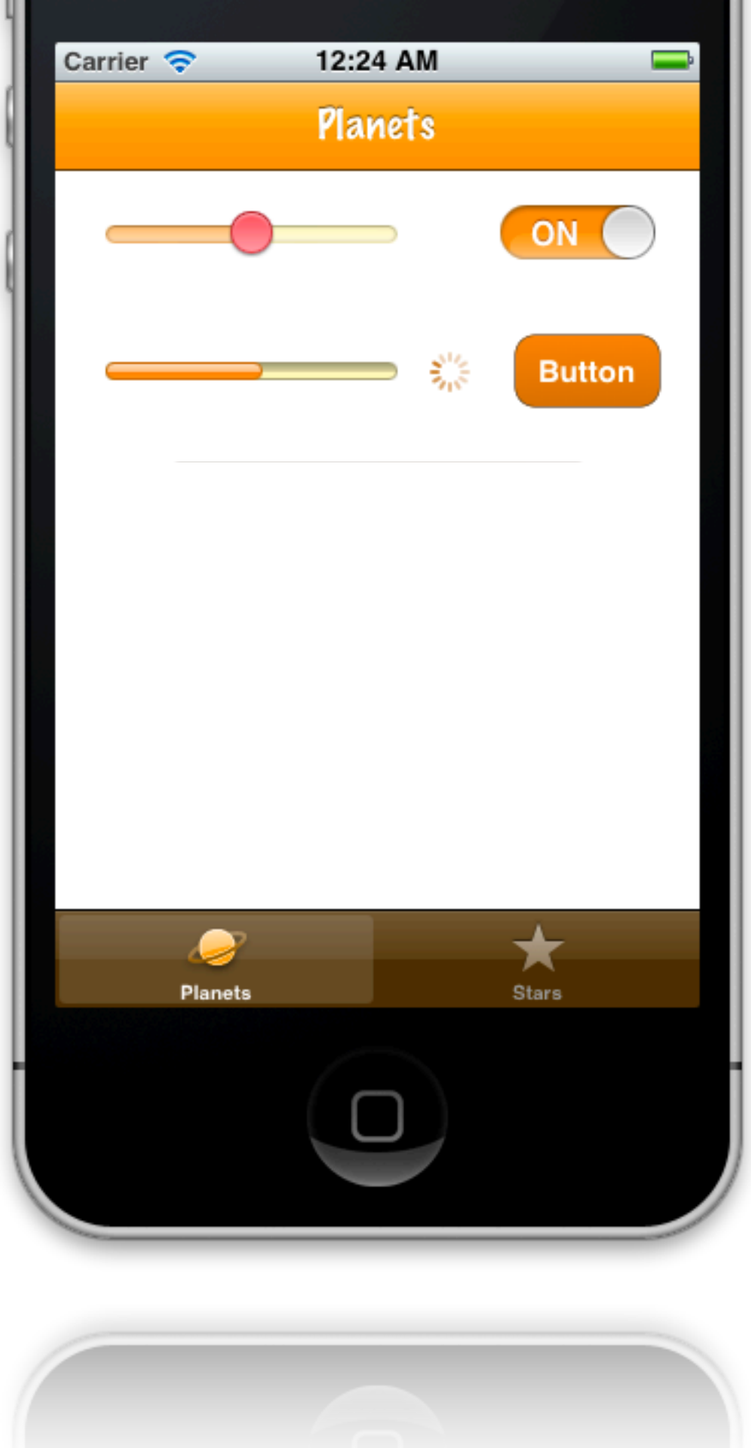
Appearance Customization in iOS 5

Ortwin Gentz
@futuretap

Neues in iOS 5

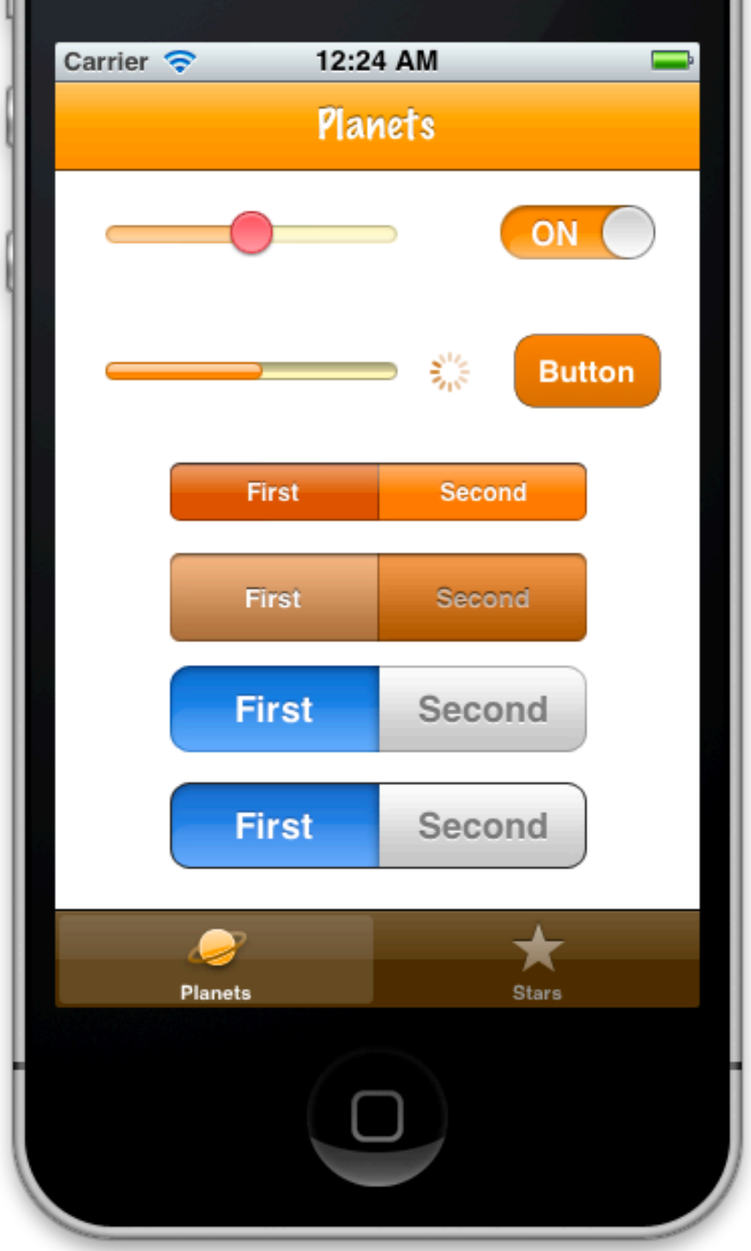
tintColor

- UISlider
- UISwitch
- UIActivityIndicator
- UIProgressView
- UIButton
- UITabBar



tintColor

- UISlider
- UISwitch
- UIActivityIndicator
- UIProgressView
- UIButton
- UITabBar



Background Images

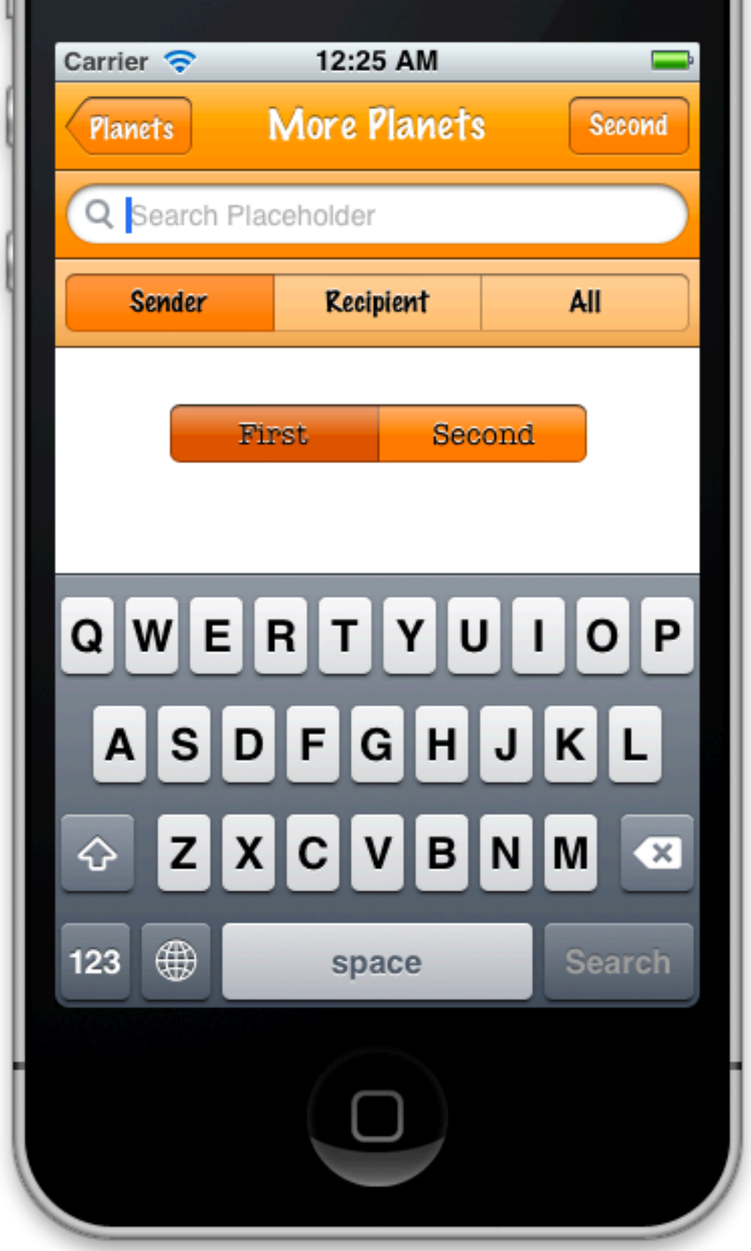
- UI*Bar
- UIBarButtonItem
- UISlider
- UIProgressView
- UISegmentedControl

Background Images

- UI*Bar
- UIBarButtonItem
- UISlider
- UIProgressView
- UISegmentedControl

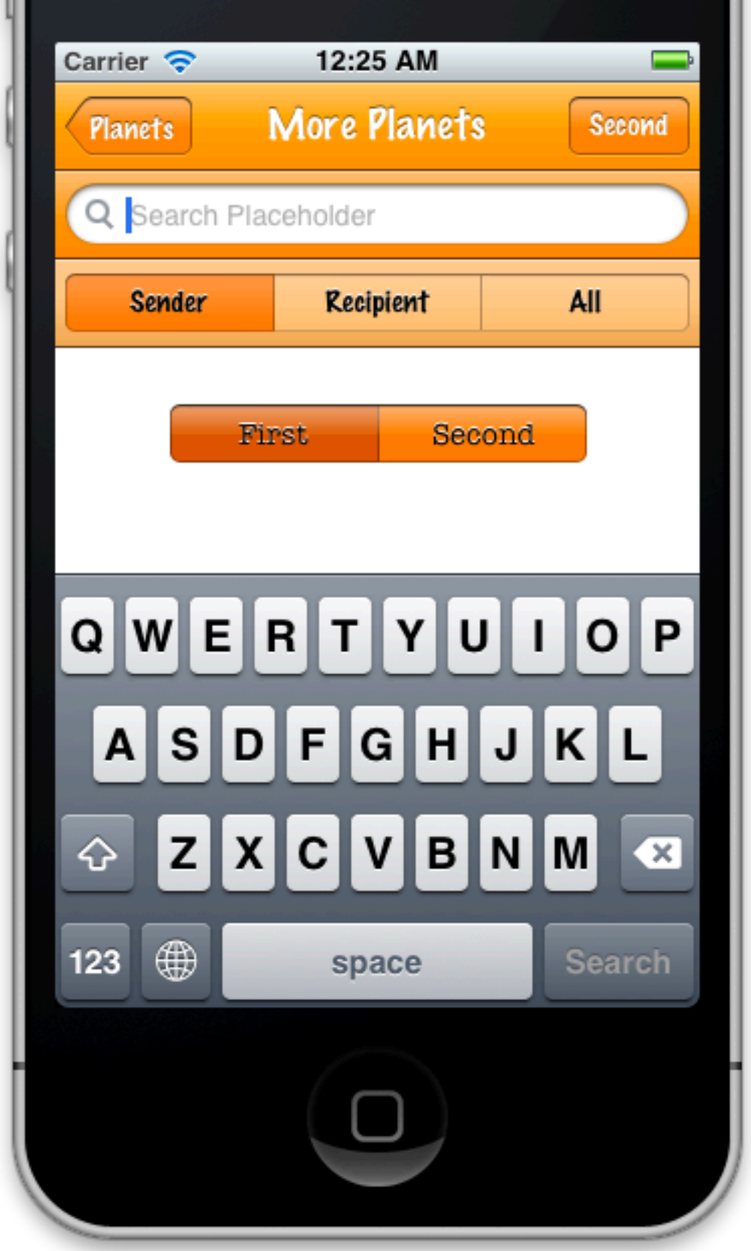
Text-Stile

- UINavigationBar (title)
- UIBarButtonItem
- UISearchBar
- UISegmentedControl



Text-Stile

- Font
- Text Color
- Shadow Color
- Shadow Offset



UIBarButtons nebeneinander

		tintColor	backgroundImage	textAttributes
Views	**UIActivityIndicatorView**			
	UIProgressView			
Controls	**UIButton**		✔	✔
	UISegmentedControl	✔		
	UISlider			
	UISwitch			
Bars	**UITabBar**			
	UIToolbar	✔		
	UIBarButtonItem (generell)	✔	mit customView	
	UIBarButtonItem (back)			
	UINavigationBar	✔		mit customView
	UISearchBar	✔		

		tintColor	backgroundImage	textAttributes
Views	**UIActivityIndicatorView**	neu		
	UIProgressView	neu	neu	
Controls	**UIButton**	neu	✔	✔
	UISegmentedControl	✔	neu	neu
	UISlider	neu	neu	
	UISwitch	neu		
Bars	**UITabBar**	neu	neu	
	UIToolbar	✔	neu	
	UIBarButtonItem (generell)	✔	neu (vorher m. customView)	neu
	UIBarButtonItem (back)	neu	neu	
	UINavigationBar	✔	neu	neu (vorher m. customView)
	UISearchBar	✔	neu	neu (für scopeBar)

Jedes Element einzeln anpassen?

UIAppearance Proxy!

Einer für alle

```
UISlider *slider = [[UISlider alloc] initWithFrame:…];
slider.minimumTrackTintColor = [UIColor redColor];
```

Einer für alle

```
UISlider *slider = [[UISlider alloc] initWithFrame:…];
slider.minimumTrackTintColor = [UIColor redColor];
```

```
UISlider *slider = [UISlider appearance];
slider.minimumTrackTintColor = [UIColor redColor];
```

Etwas spezifischer bitte

```
UISlider *slider = [UISlider appearance];
slider.minimumTrackTintColor = [UIColor redColor];
```

Etwas spezifischer bitte

```
UISlider *slider = [UISlider appearance];
slider.minimumTrackTintColor = [UIColor redColor];
```

```
UISlider *slider = [UISlider appearanceWhenContainedIn:
   Class <UIAppearanceContainer>)…, nil];
slider.minimumTrackTintColor = [UIColor redColor];
```

Etwas spezifischer bitte

```
UISlider *slider = [UISlider appearance];
slider.minimumTrackTintColor = [UIColor redColor];
```

```
UISlider *slider = [UISlider appearanceWhenContainedIn:
   Class <UIAppearanceContainer>)…, nil];
slider.minimumTrackTintColor = [UIColor redColor];
```

Etwas spezifischer bitte

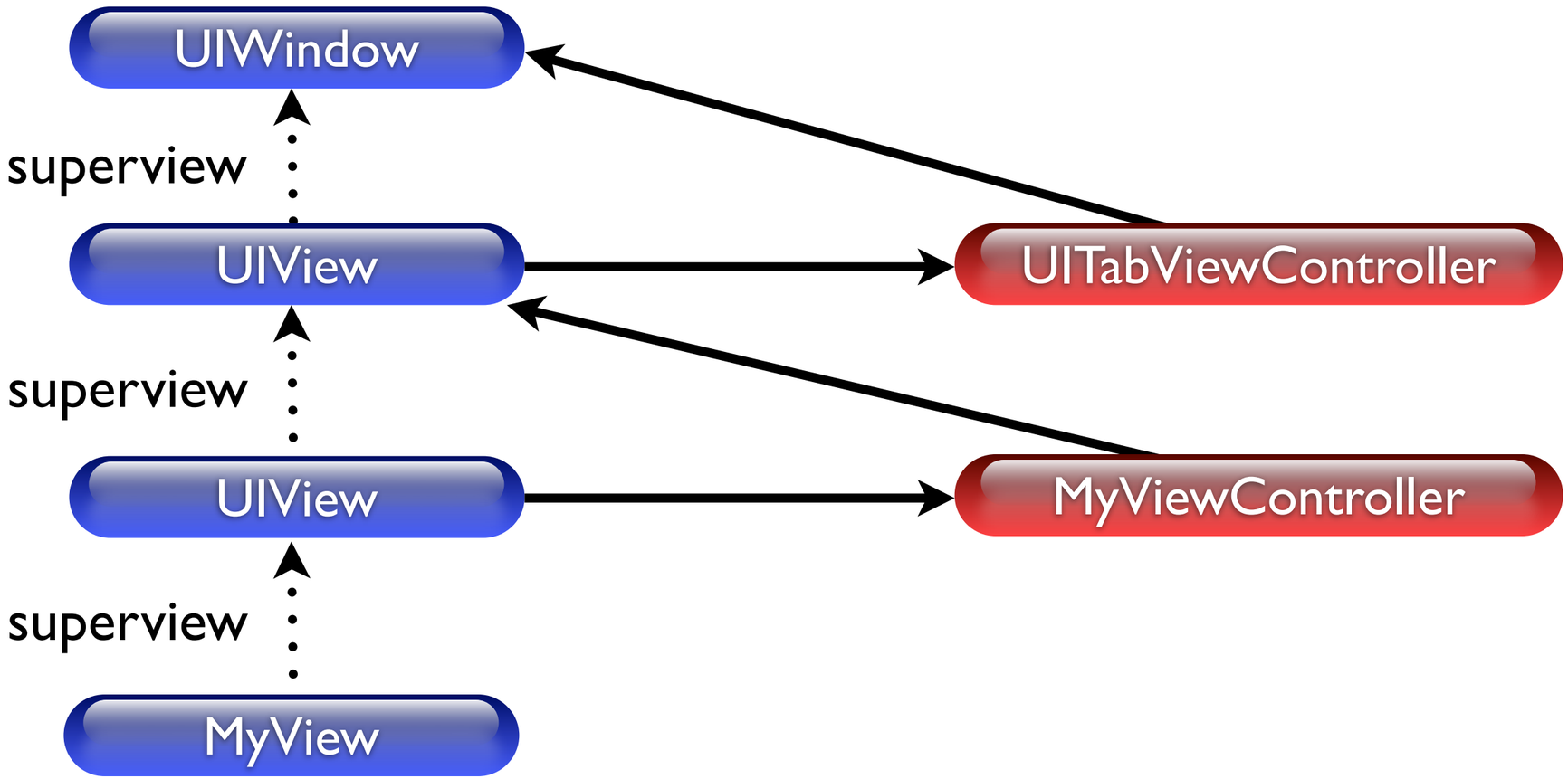
```
UISlider *slider = [UISlider appearance];
slider.minimumTrackTintColor = [UIColor redColor];
```

```
UISlider *slider = [UISlider appearanceWhenContainedIn:
   [MyViewController class], nil];
slider.minimumTrackTintColor = [UIColor redColor];
```

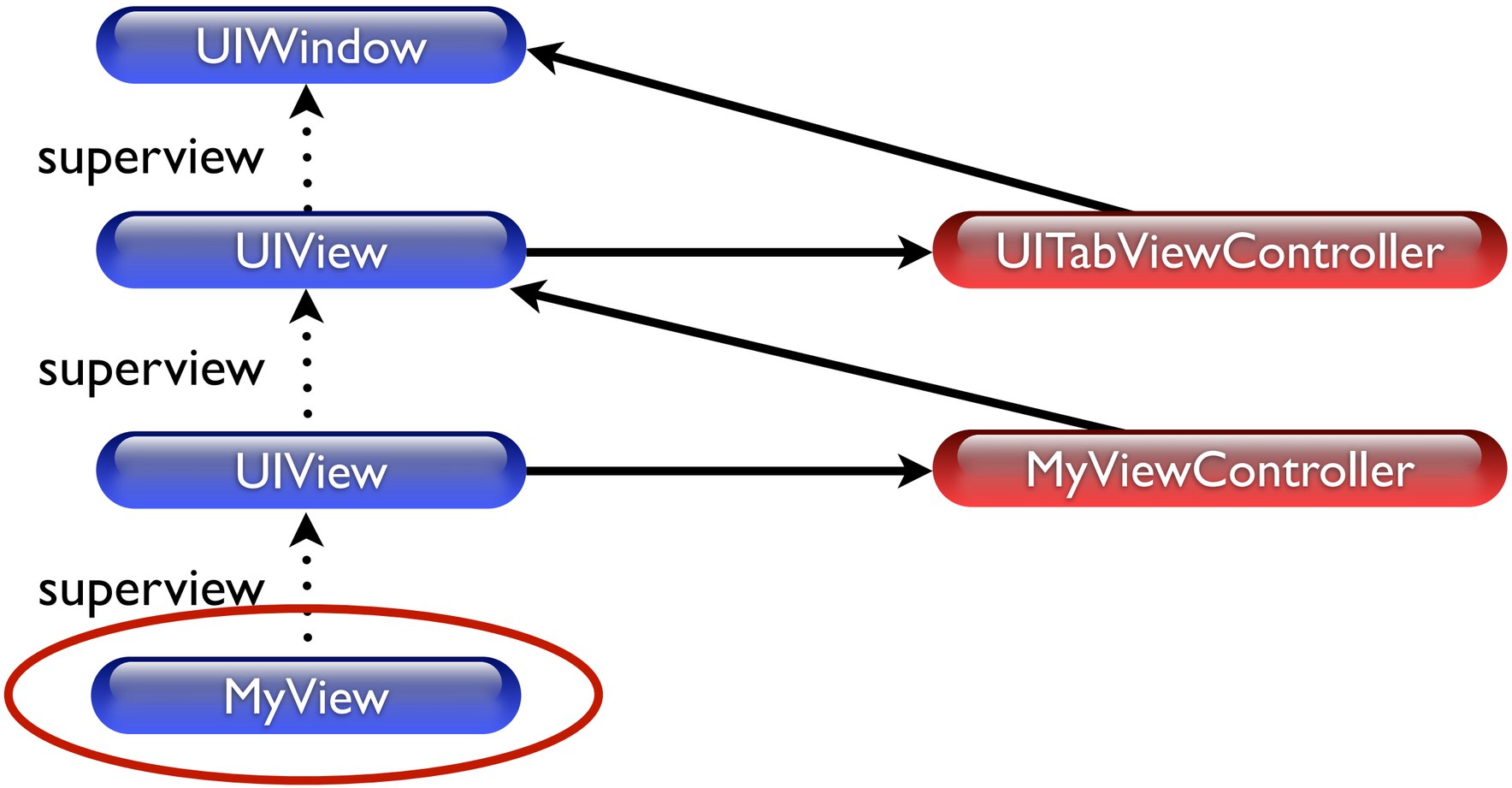
UIAppearanceContainer

- Views
- ViewController
- PopoverController

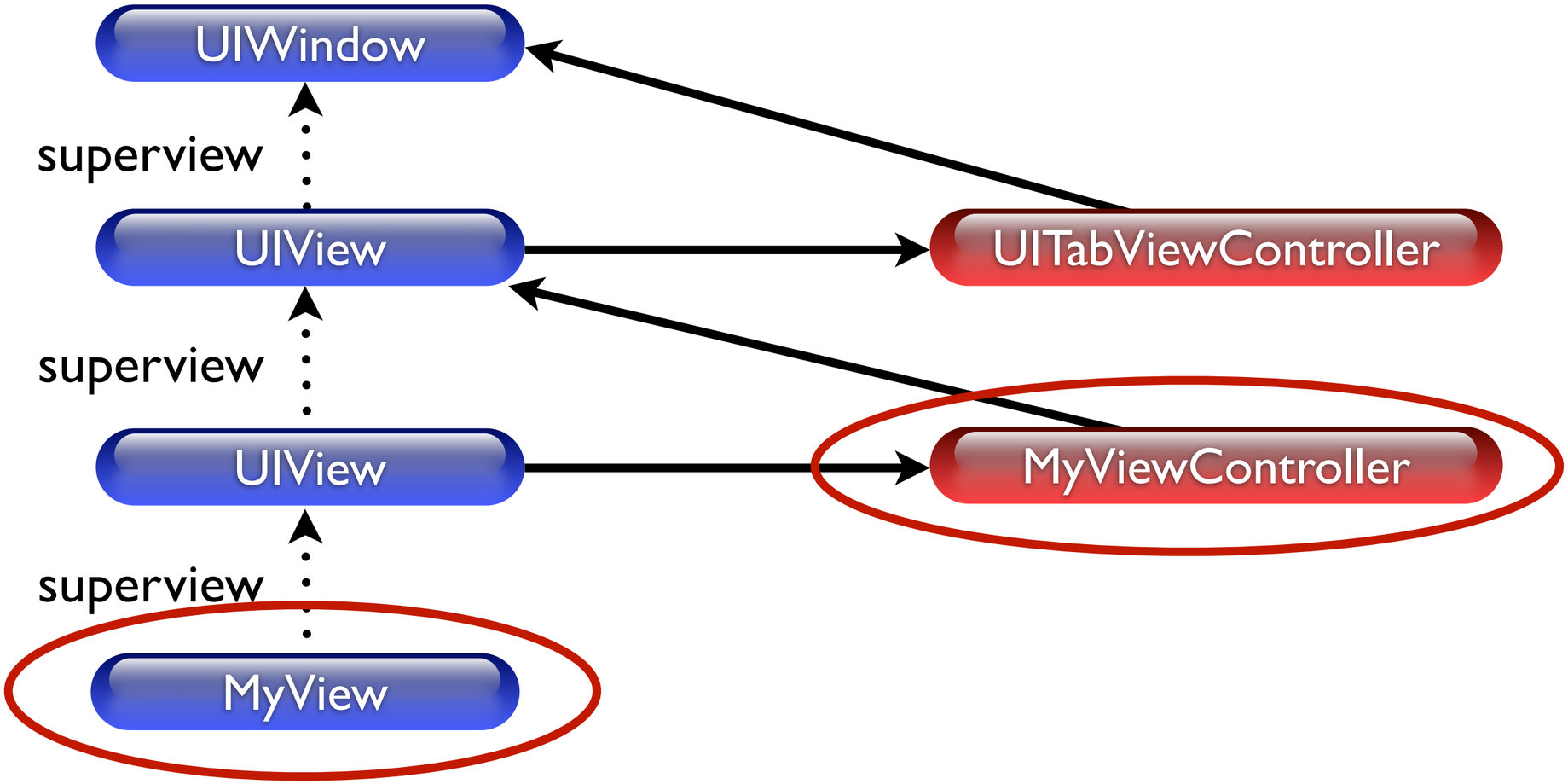
Containment



Containment



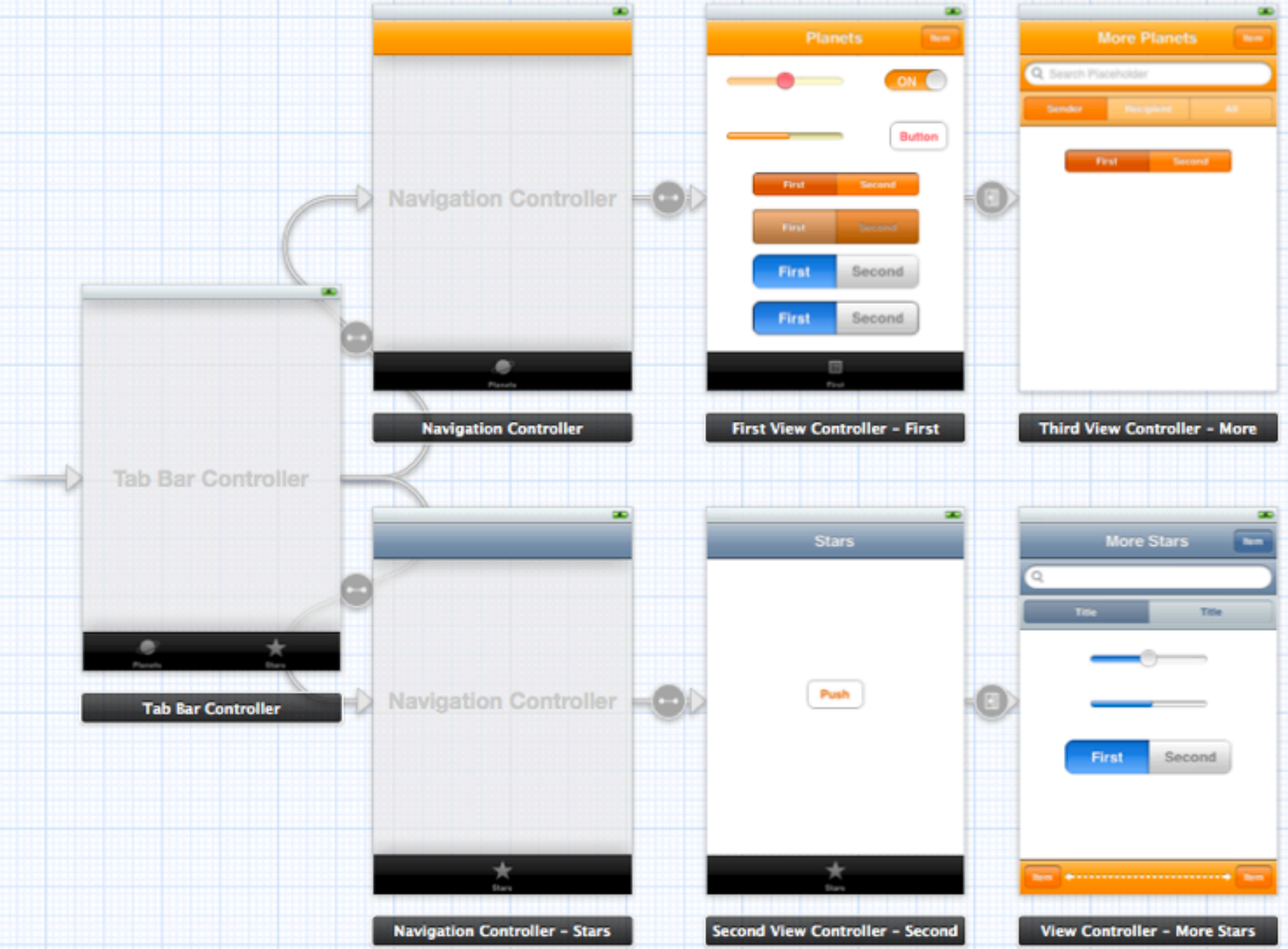
Containment

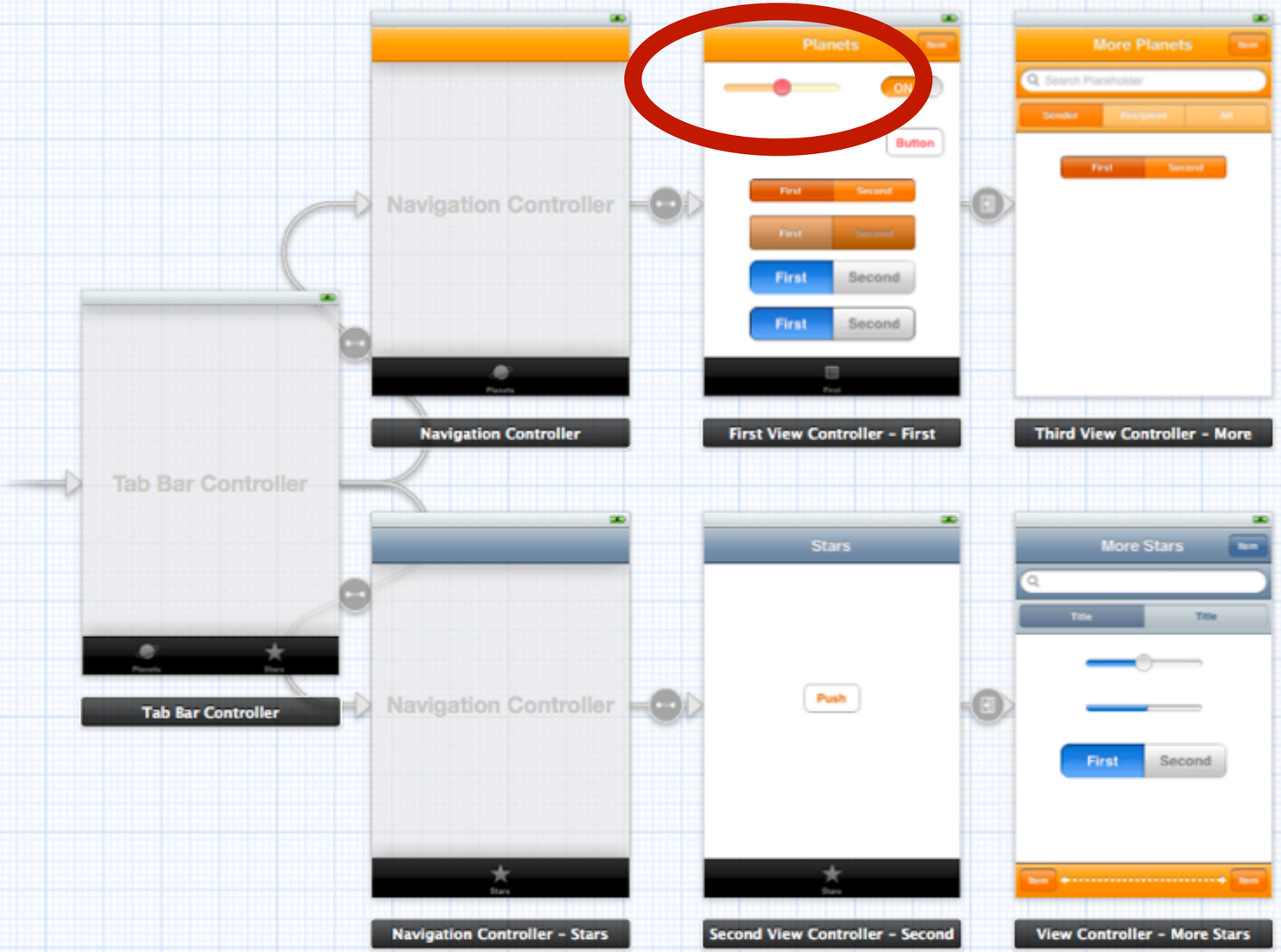


Containment

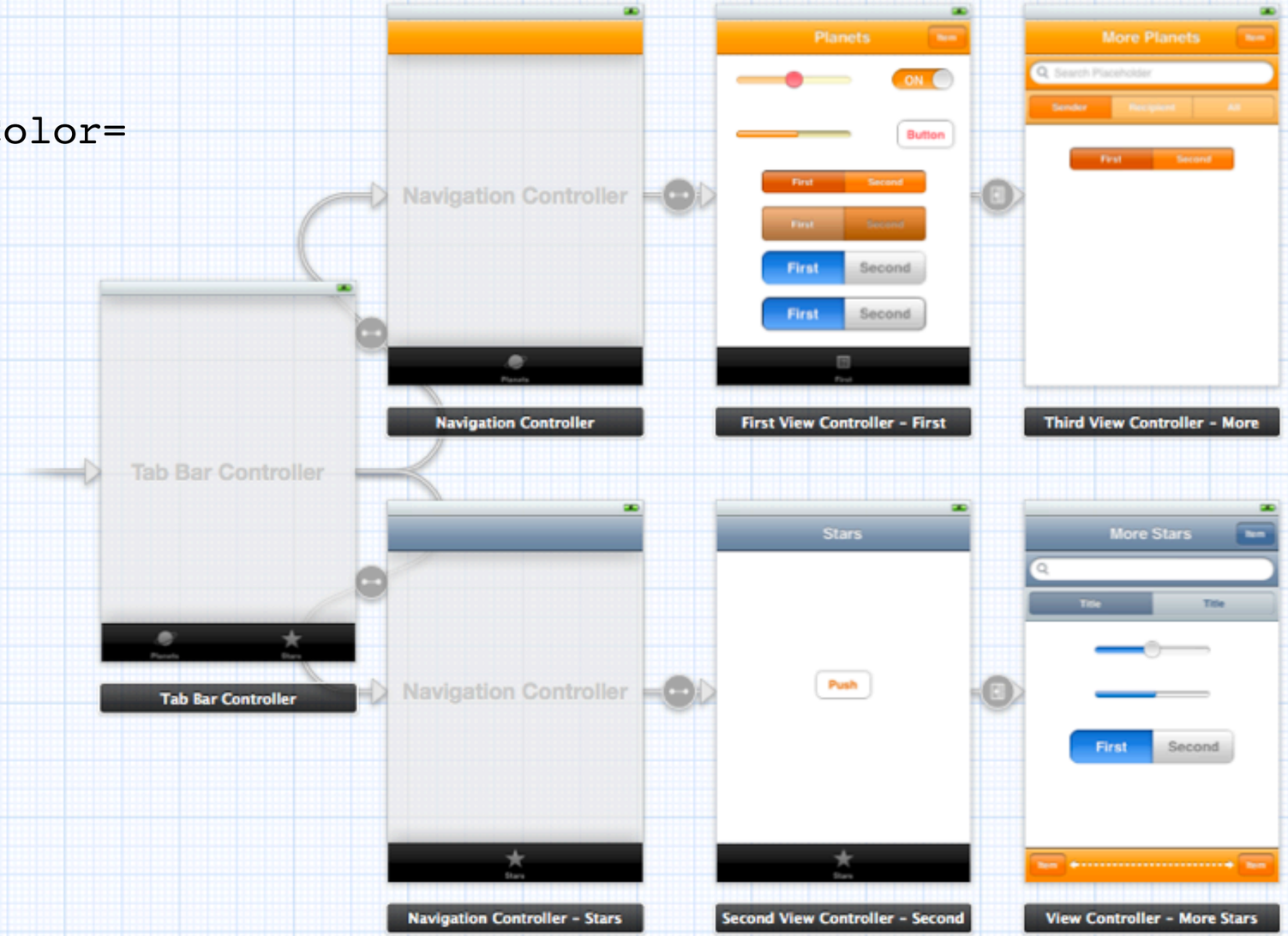
```
UISlider *slider = [UISlider appearanceWhenContainedIn:
    [MyView class], [MyViewController class], nil];
slider.minimumTrackTintColor = [UIColor redColor];
```

Beispiele

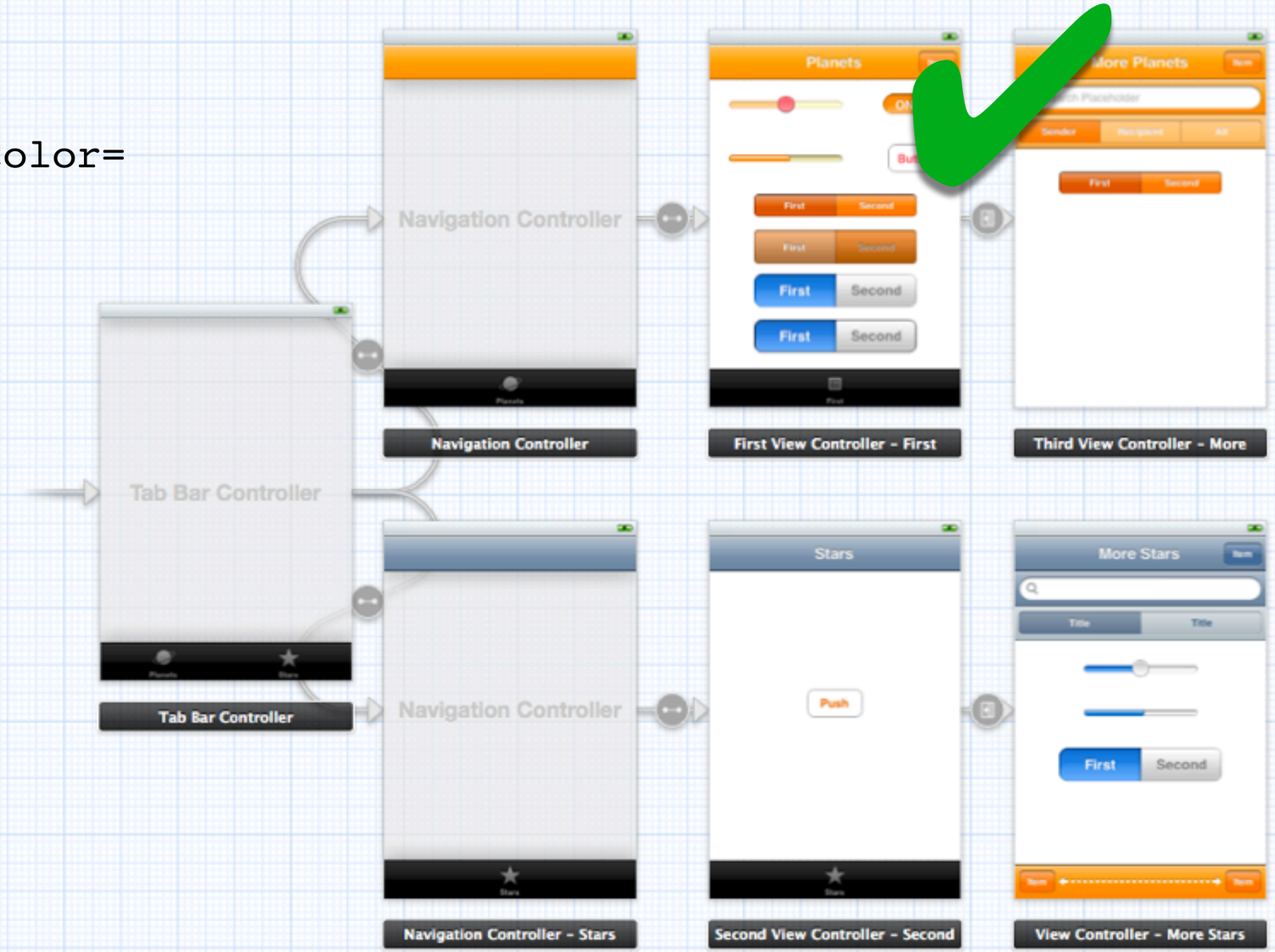

Navigation Controller
Planets
ON
Button
First
Second
First
Second
First
Second
First
Second
More Planets
Search Placeholder
Sender
Recipient
All
First
Second
Navigation Controller
First View Controller - First
Third View Controller - More
Tab Bar Controller
Tab Bar Controller
Navigation Controller
Stars
Push
More Stars
Title
Title
First
Second
Navigation Controller - Stars
Second View Controller - Second
View Controller - More Stars


Navigation Controller
Planets
Button
First
Second
First
Second
First
Second
First
Second
More Planets
Search Placeholder
Sender
Recipient
All
First
Second
Tab Bar Controller
Navigation Controller
First View Controller - First
Third View Controller - More
Tab Bar Controller
Navigation Controller
Stars
Push
More Stars
Title
Title
First
Second
Navigation Controller - Stars
Second View Controller - Second
View Controller - More Stars

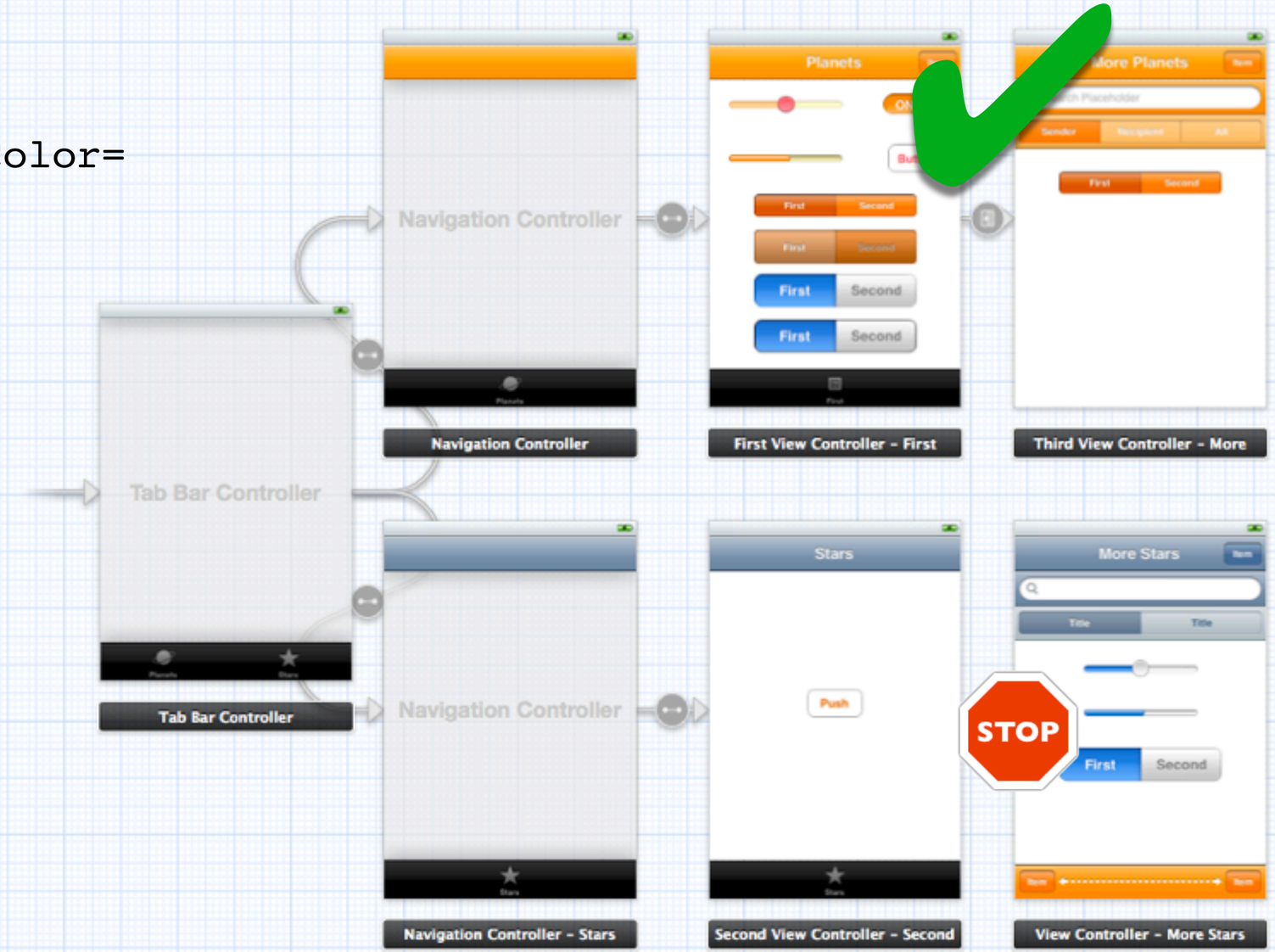
```
[UISlider appearance].tintColor=
```



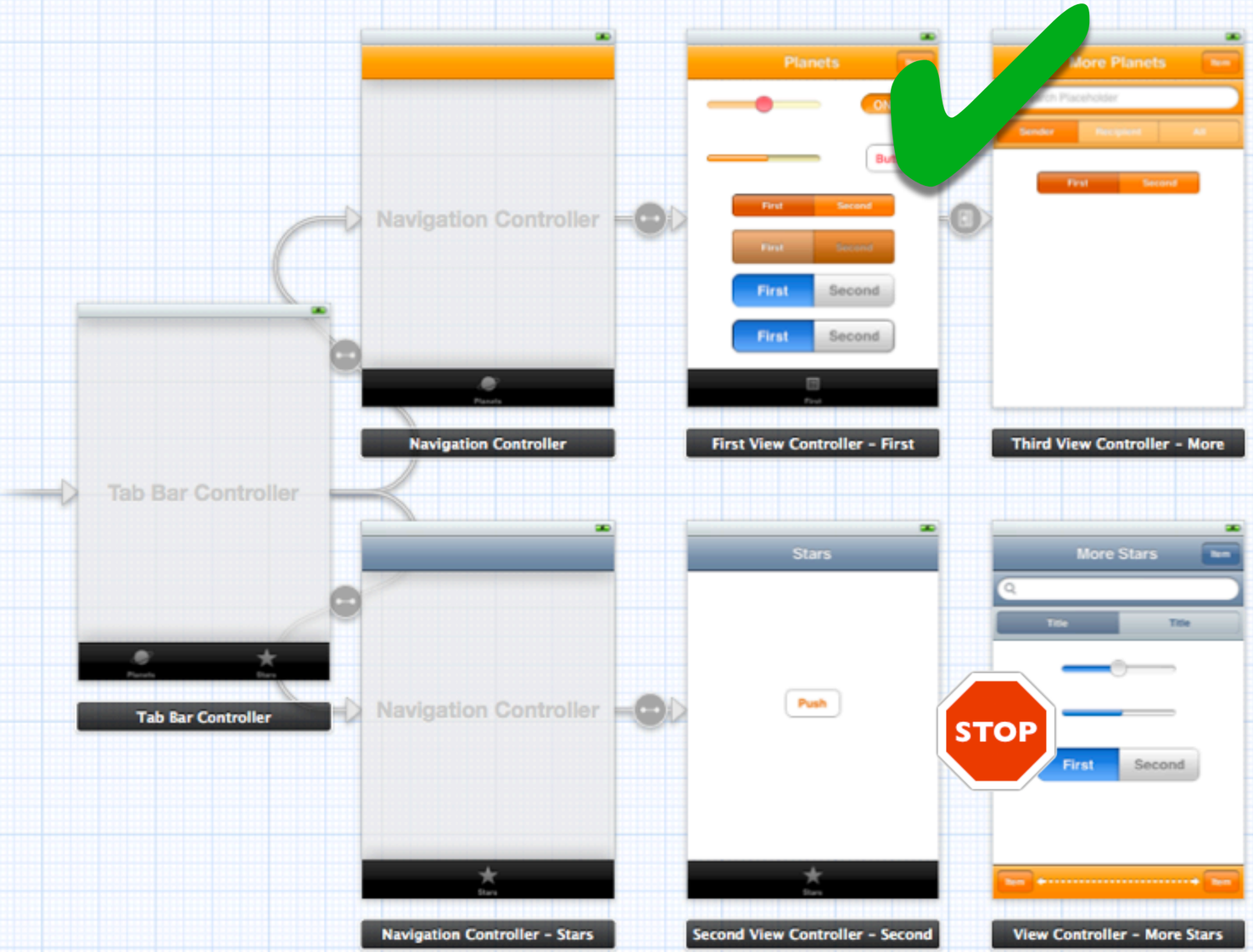
```
[UISlider appearance].tintColor=
```



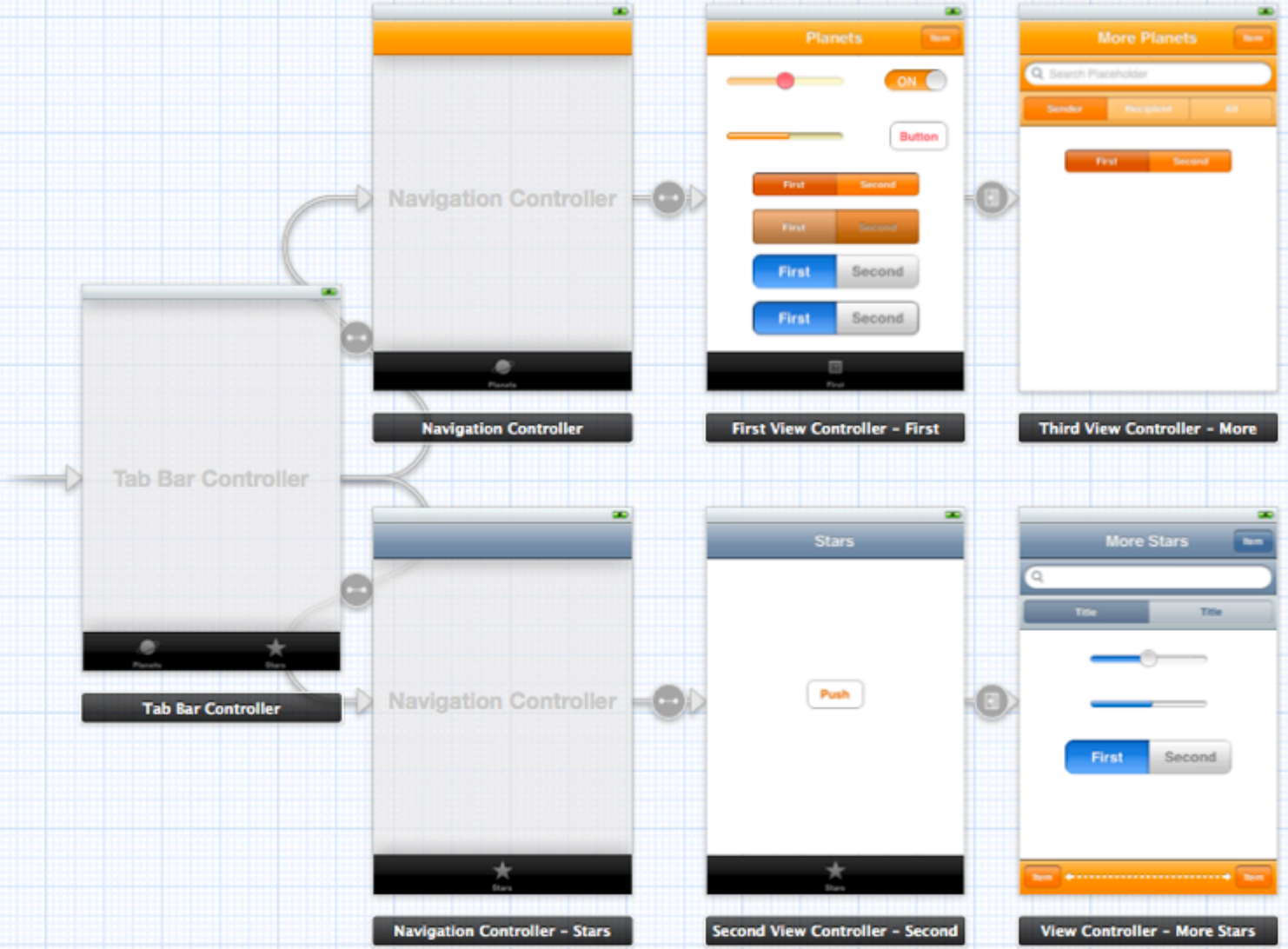
```
[UISlider appearance].tintColor=
```

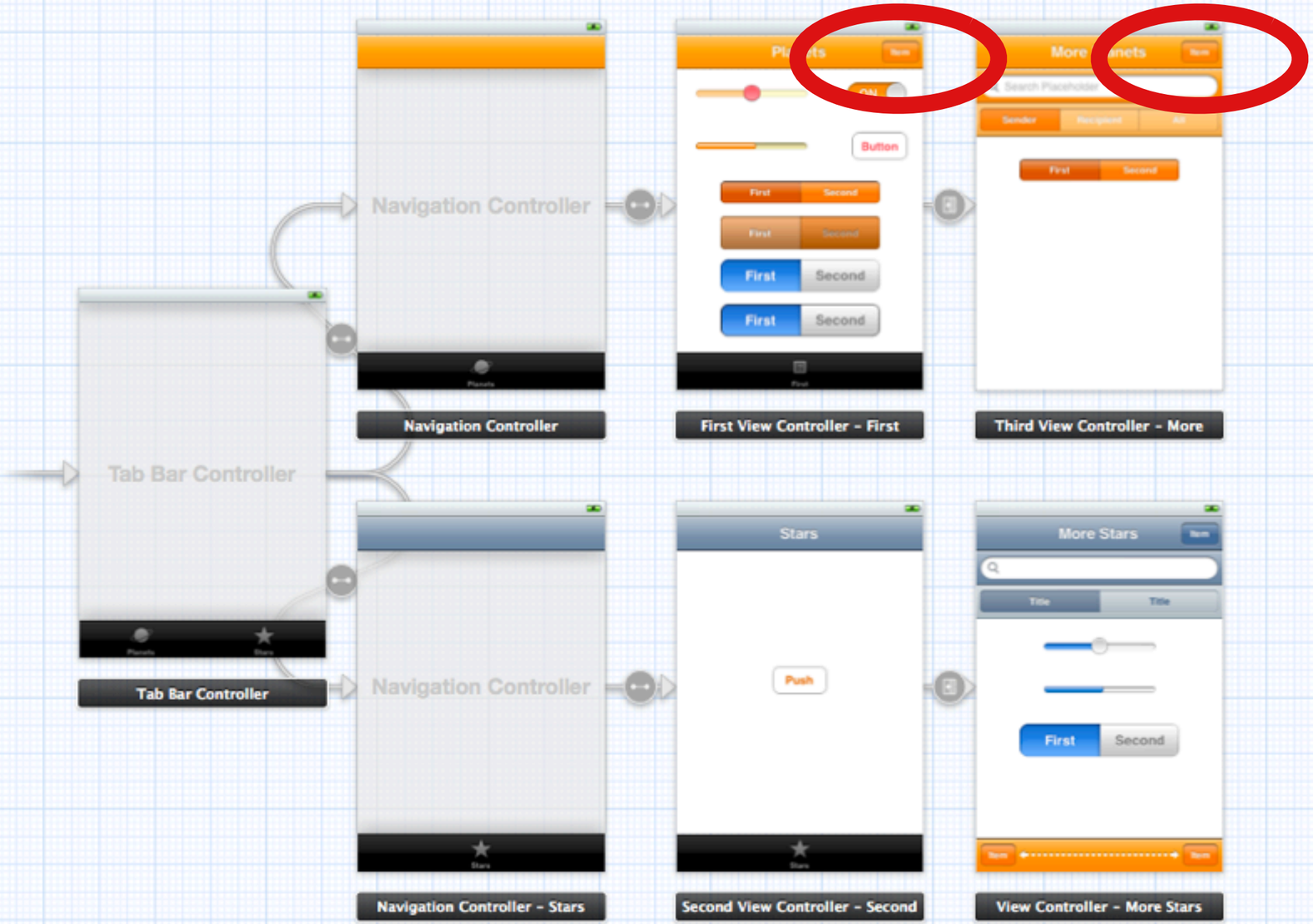


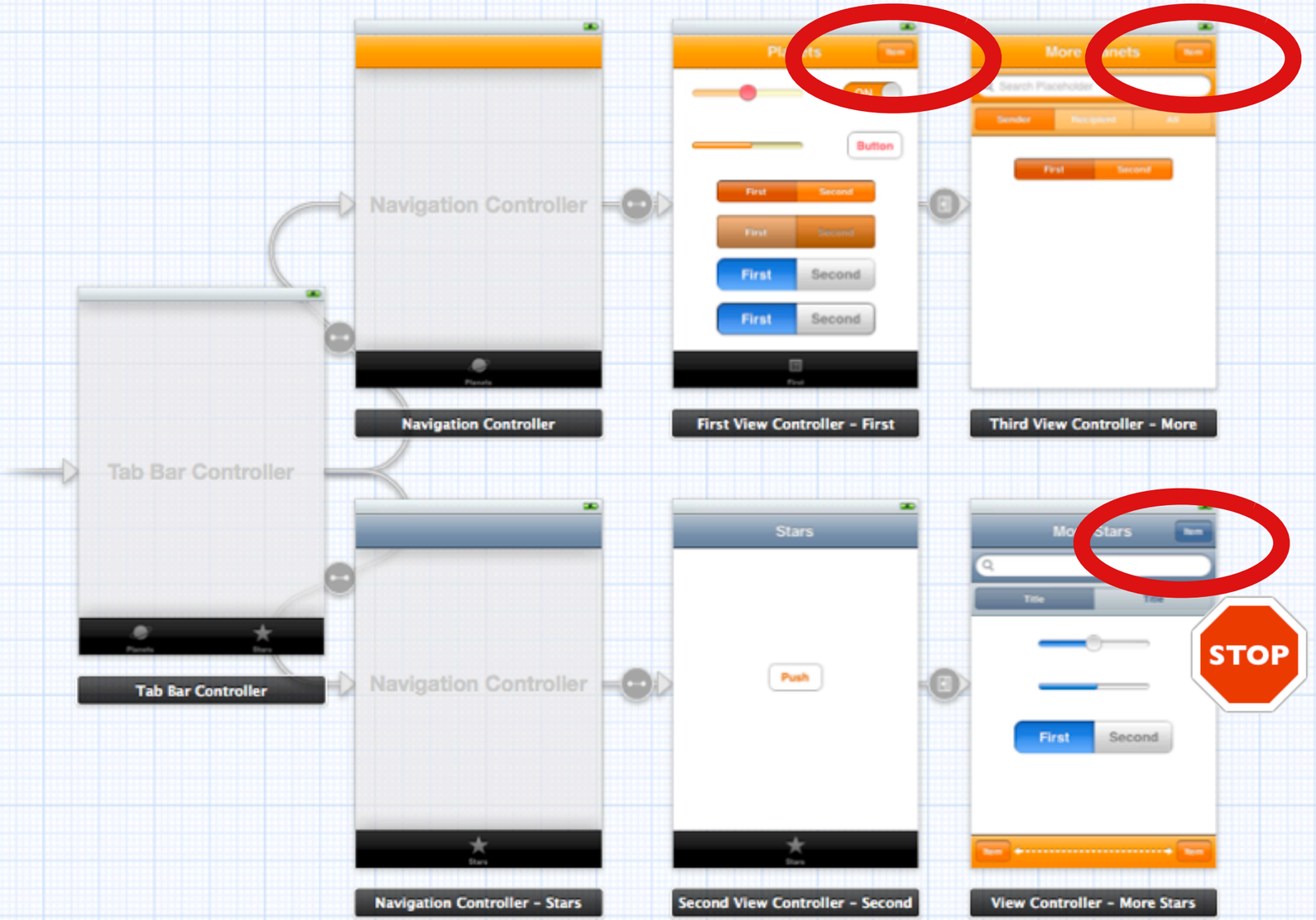
```
[UISlider
appearanceWhenContainedIn:
[FirstViewController class],
nil].tintColor=
```



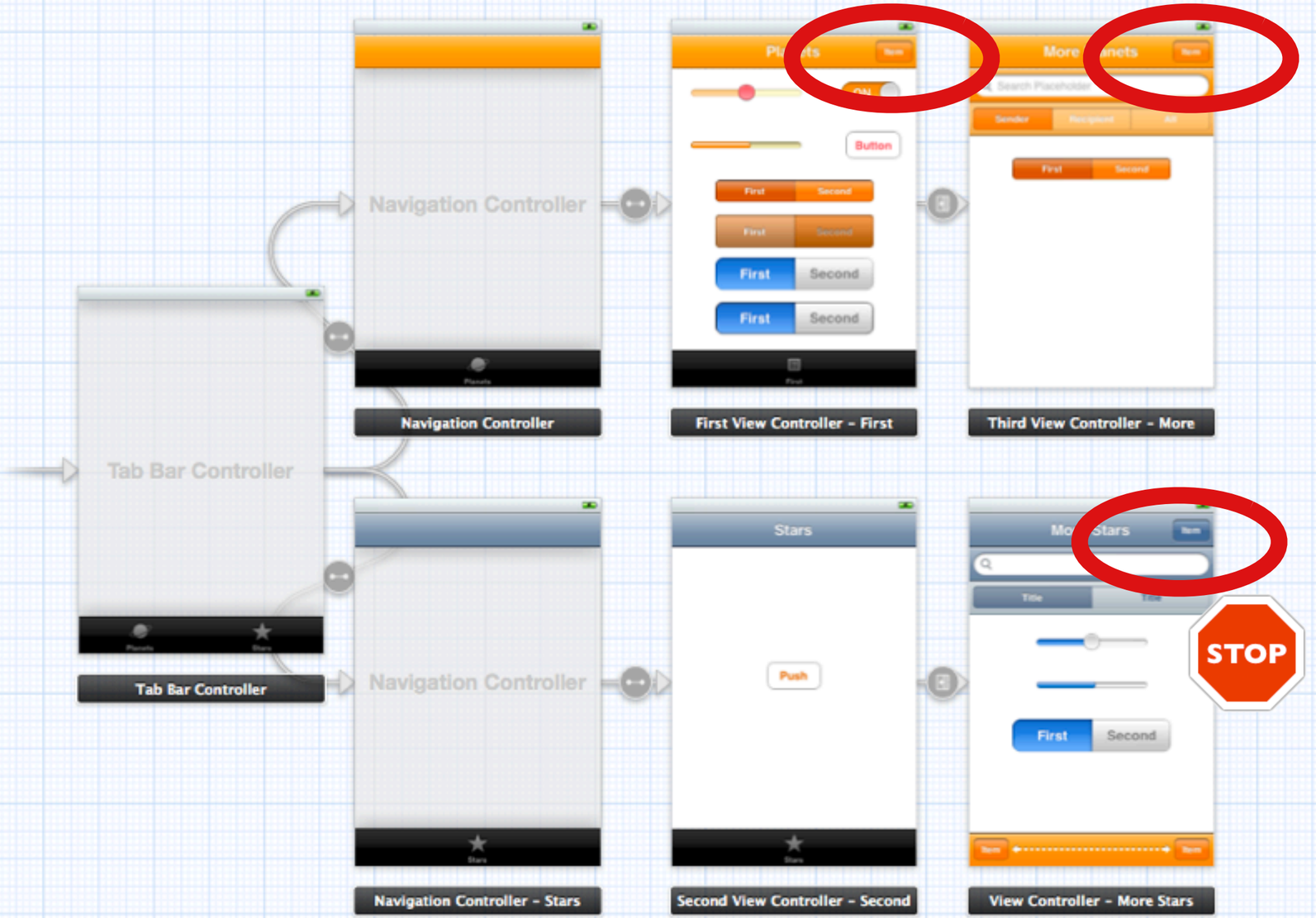
```
[UISlider
appearanceWhenContainedIn:
[FirstViewController class],
nil].tintColor=
```


Navigation Controller
Planets
ON
Button
First
Second
First
Second
First
Second
First
Second
More Planets
Search Placeholder
Sender
Recipient
All
First
Second
Navigation Controller
First View Controller - First
Third View Controller - More
Tab Bar Controller
Tab Bar Controller
Navigation Controller
Stars
Push
More Stars
Title
Title
First
Second
Navigation Controller - Stars
Second View Controller - Second
View Controller - More Stars

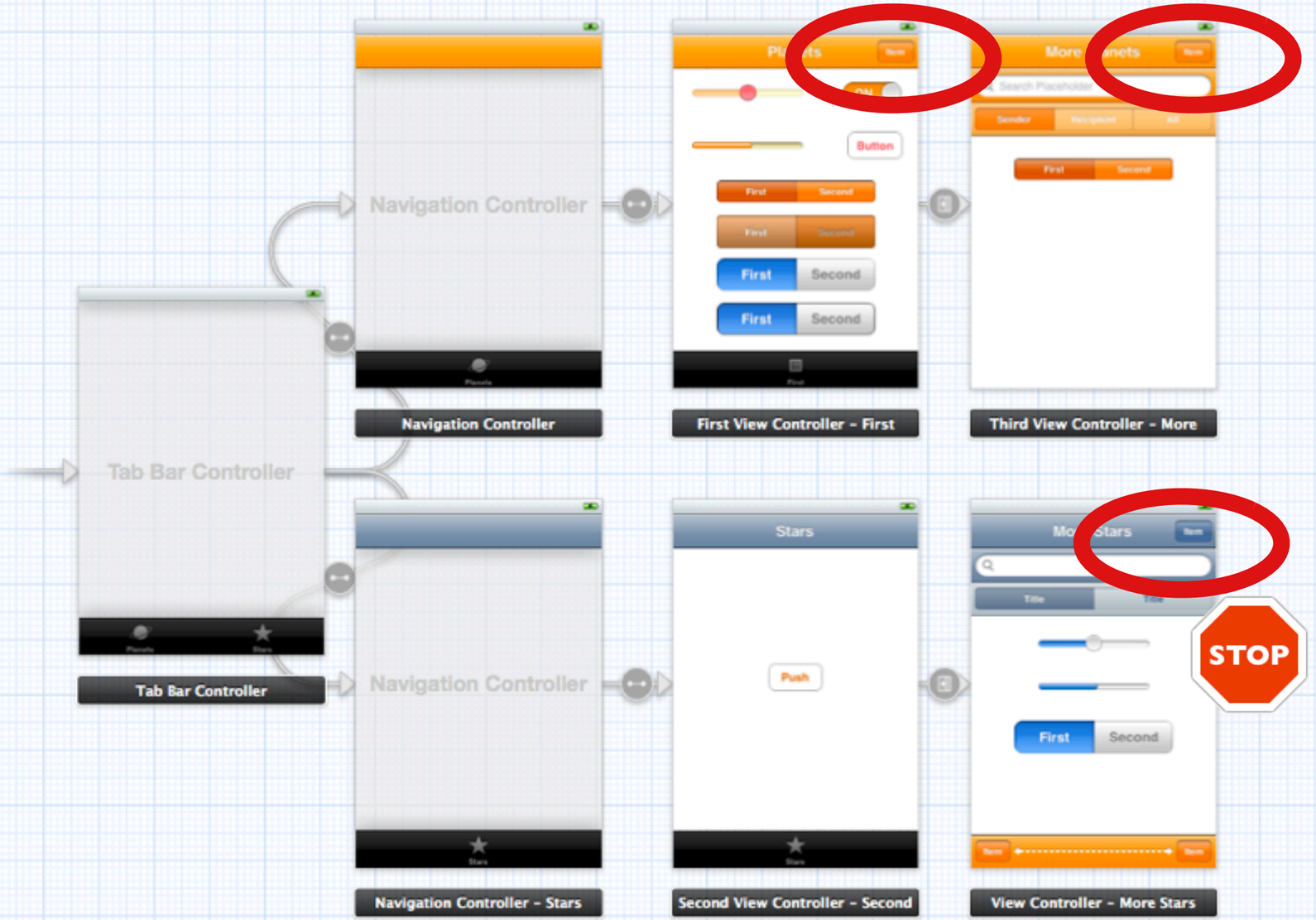

Navigation Controller
Navigation Controller
First
Second
Button
First View Controller - First
More
Search Placeholder
Sender
Recipient
All
First
Second
Third View Controller - More
Tab Bar Controller
Tab Bar Controller
Navigation Controller
Navigation Controller - Stars
Stars
Push
Second View Controller - Second
More Stars
Title
Title
First
Second
View Controller - More Stars


Navigation Controller
Tab Bar Controller
Navigation Controller
First View Controller - First
Third View Controller - More
Stars
Navigation Controller
Tab Bar Controller
Navigation Controller - Stars
Second View Controller - Second
View Controller - More Stars
Push
First
Second
STOP

```
[UIBarButtonItem
appearanceWhenContainedIn:
```

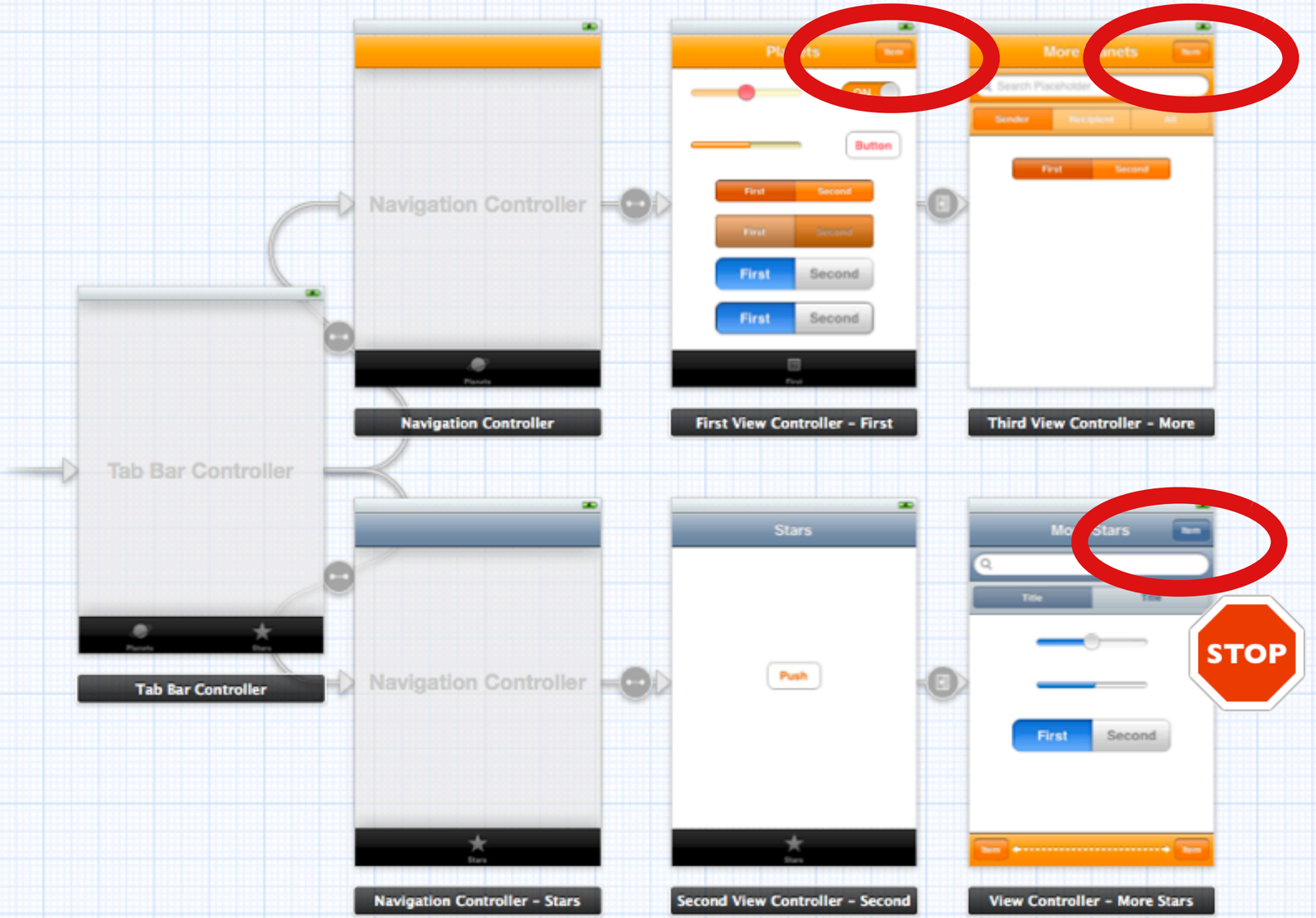


```
[UIBarButtonItem
appearanceWhenContainedIn:
[FirstViewController class],
nil]
```

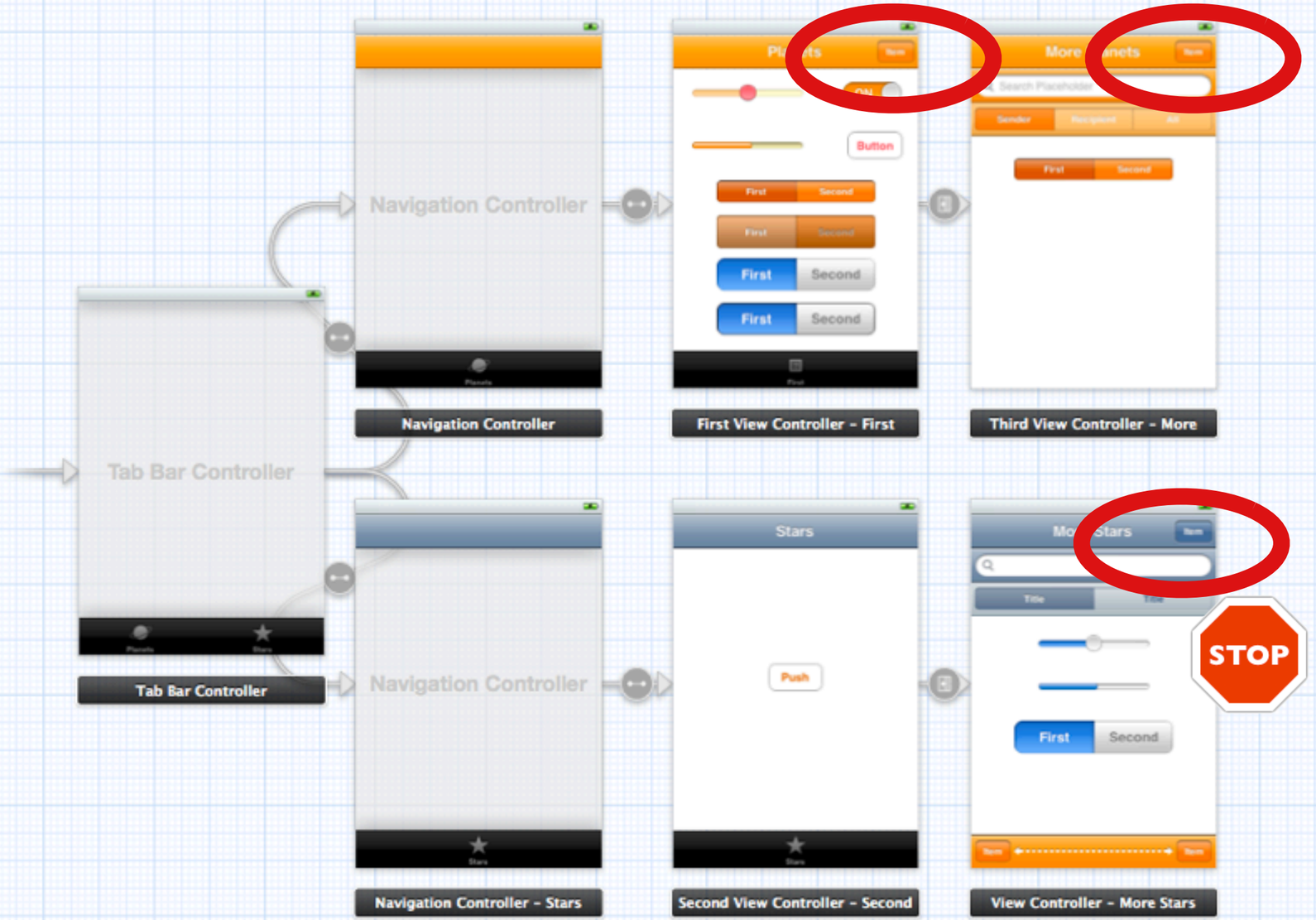


```
[UIBarButtonItem
appearanceWhenContainedIn:
[FirstViewController class],
nil]
```

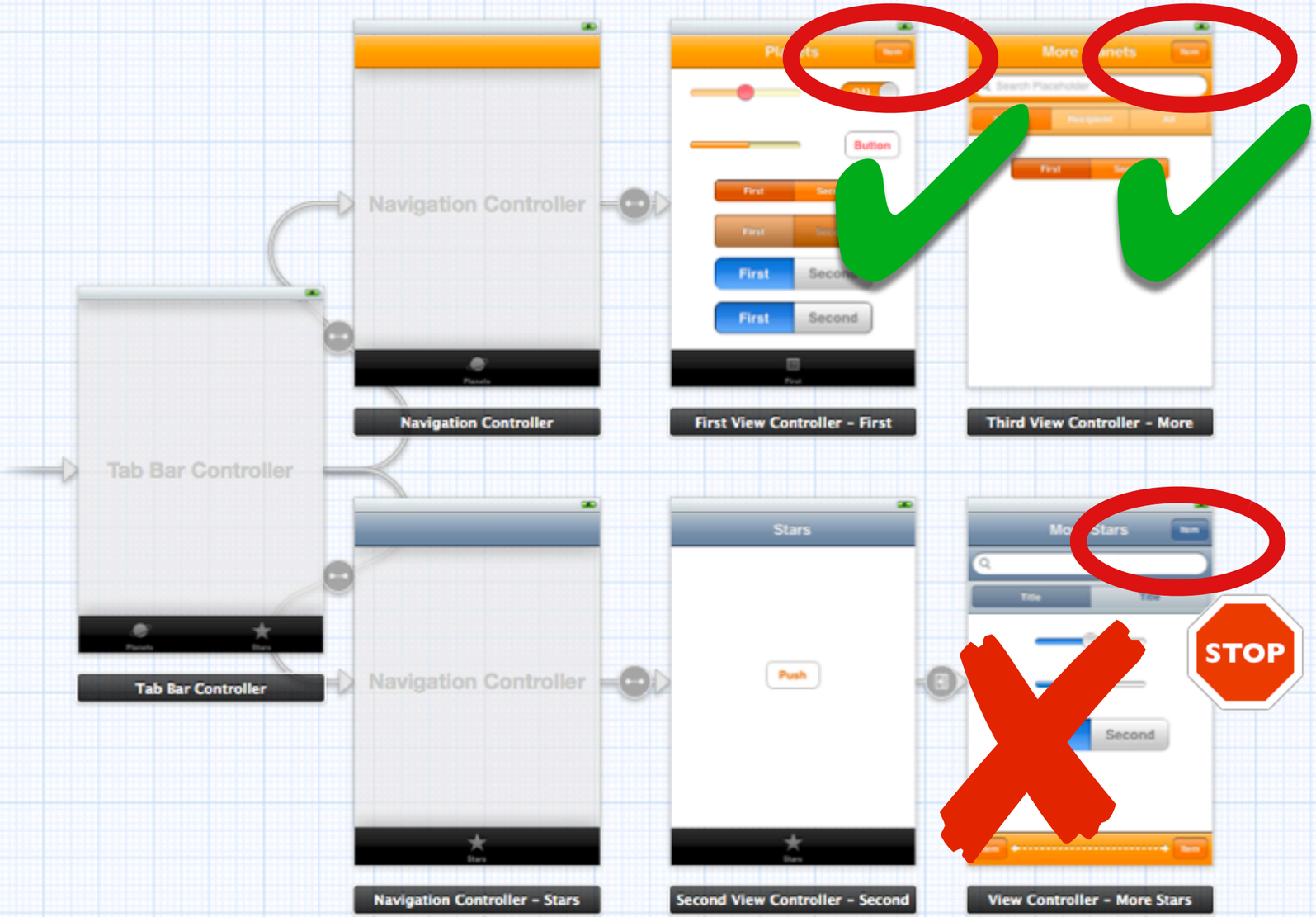
```
[UIBarButtonItem
appearanceWhenContainedIn:
```



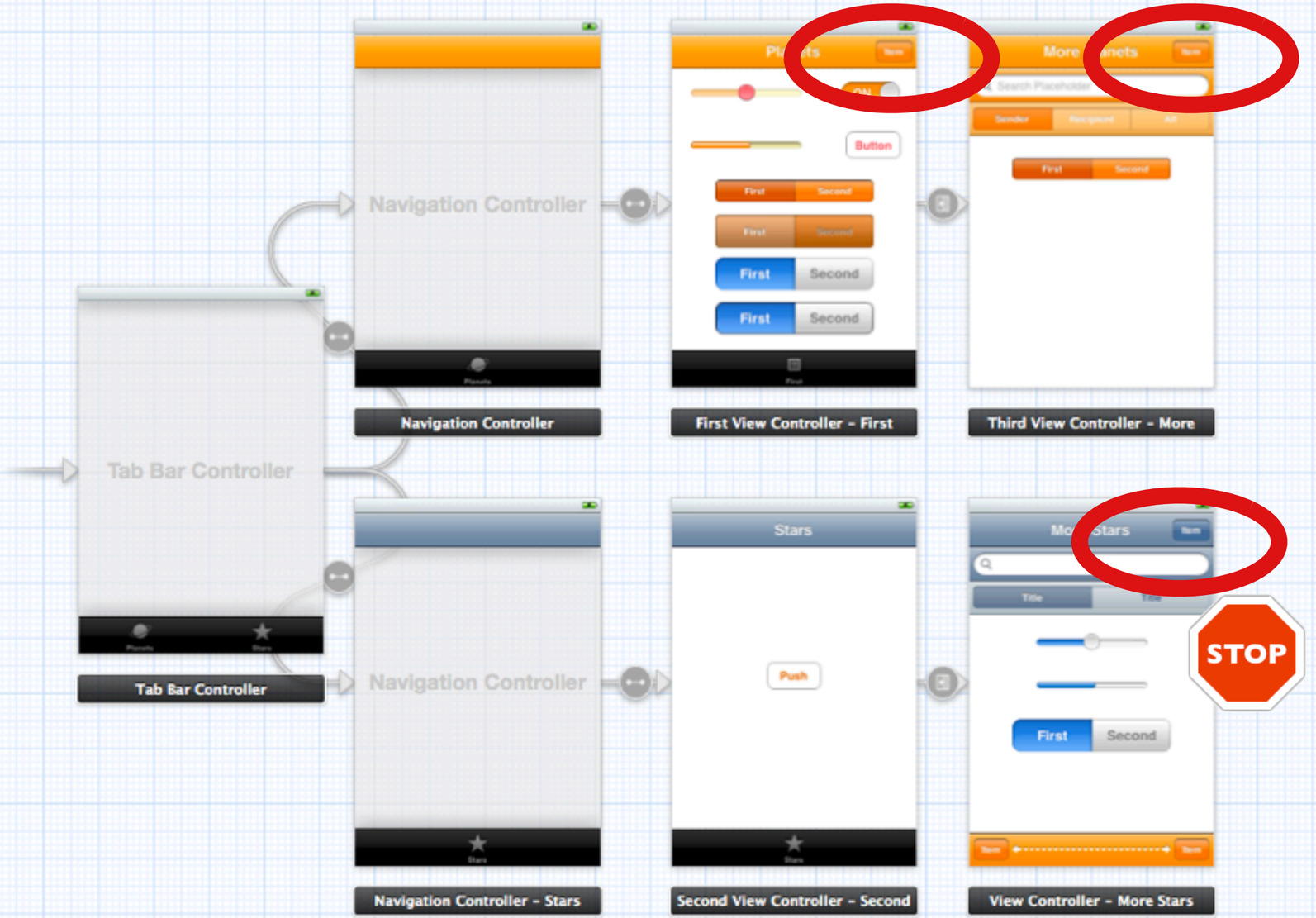
```
[UIBarButtonItem
appearanceWhenContainedIn:
[UINavigationController
class], nil]
```



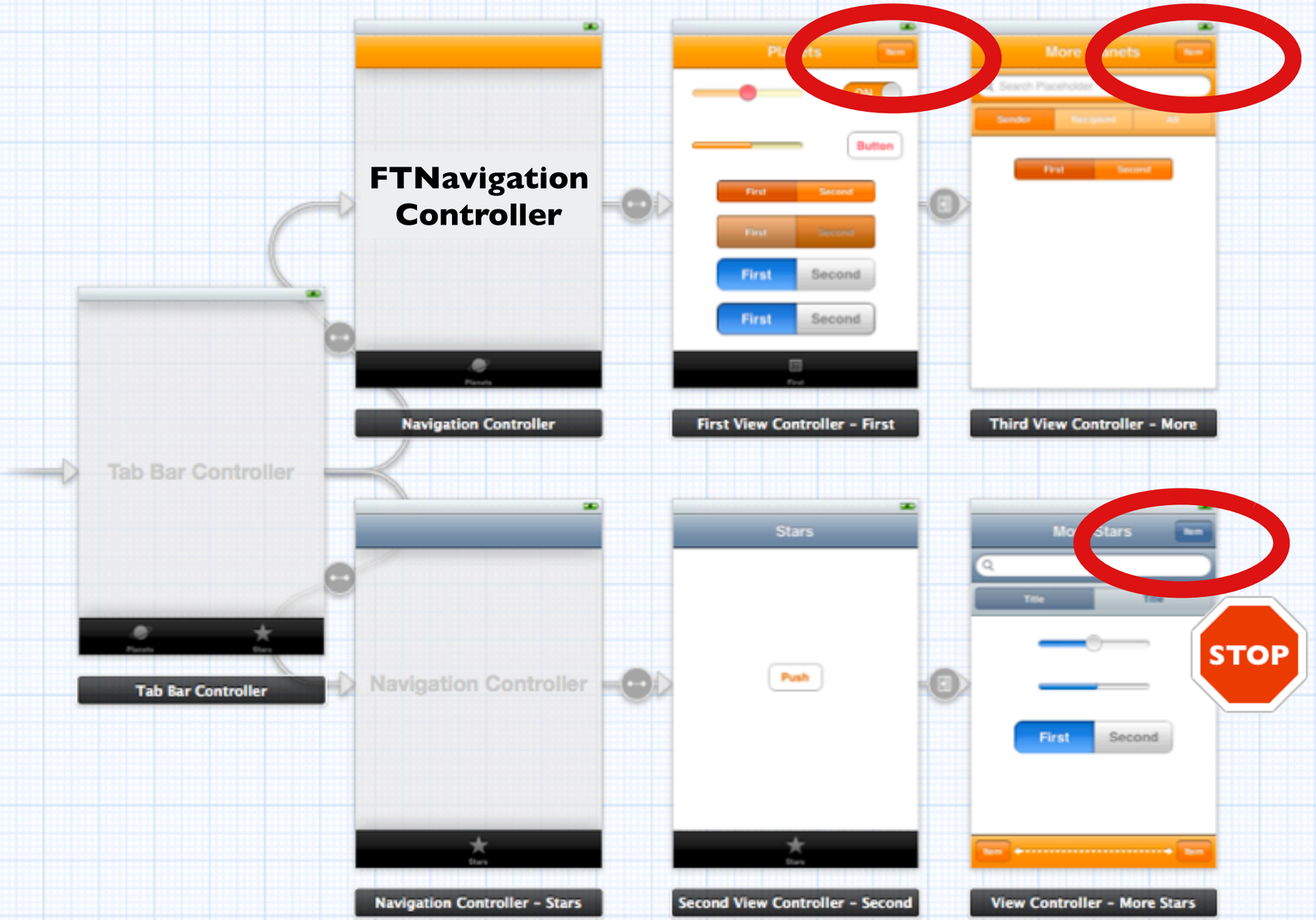
```
[UIBarButtonItem
appearanceWhenContainedIn:
[UINavigationController
class], nil]
```



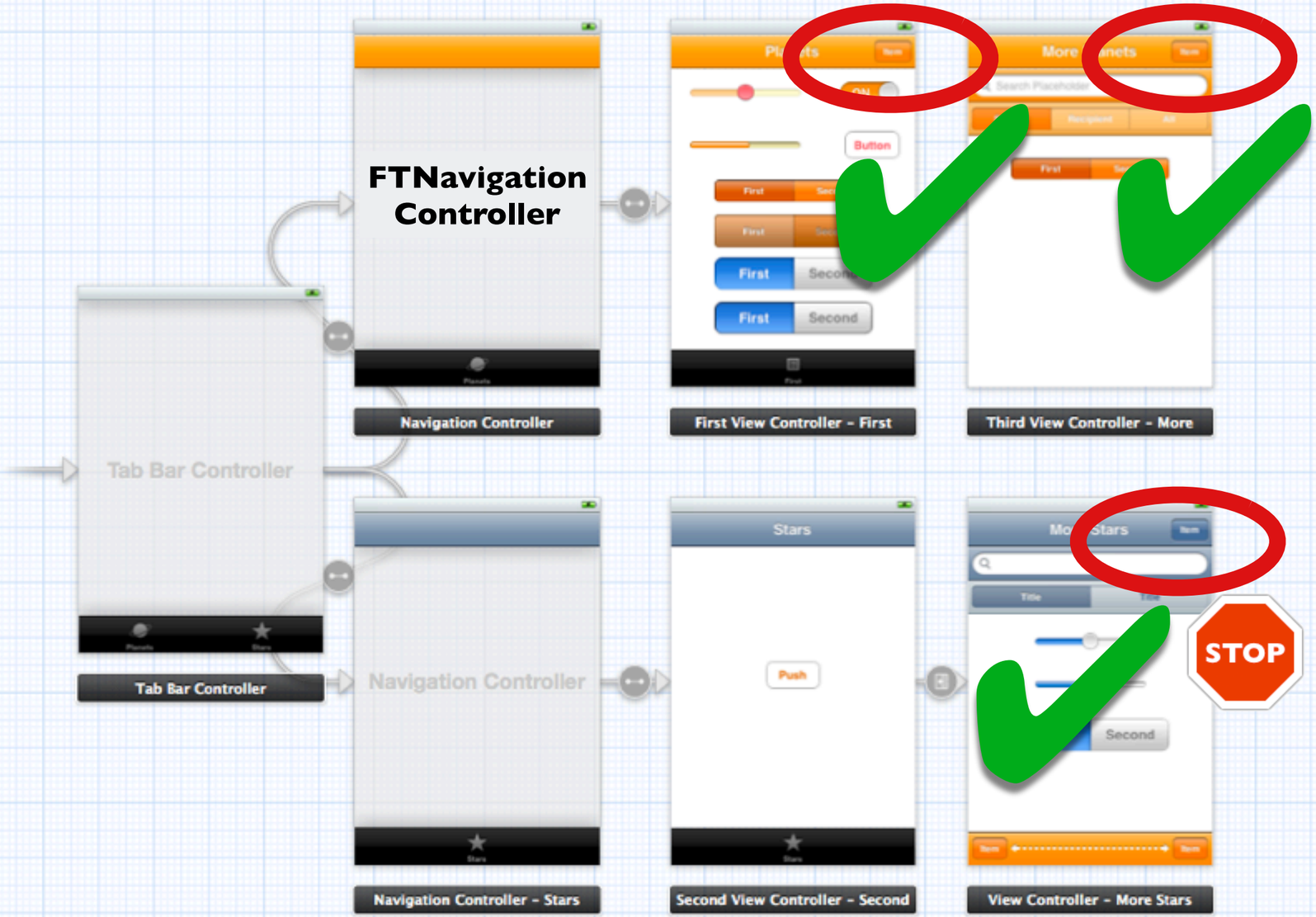
```
[UIBarButtonItem
appearanceWhenContainedIn:
```

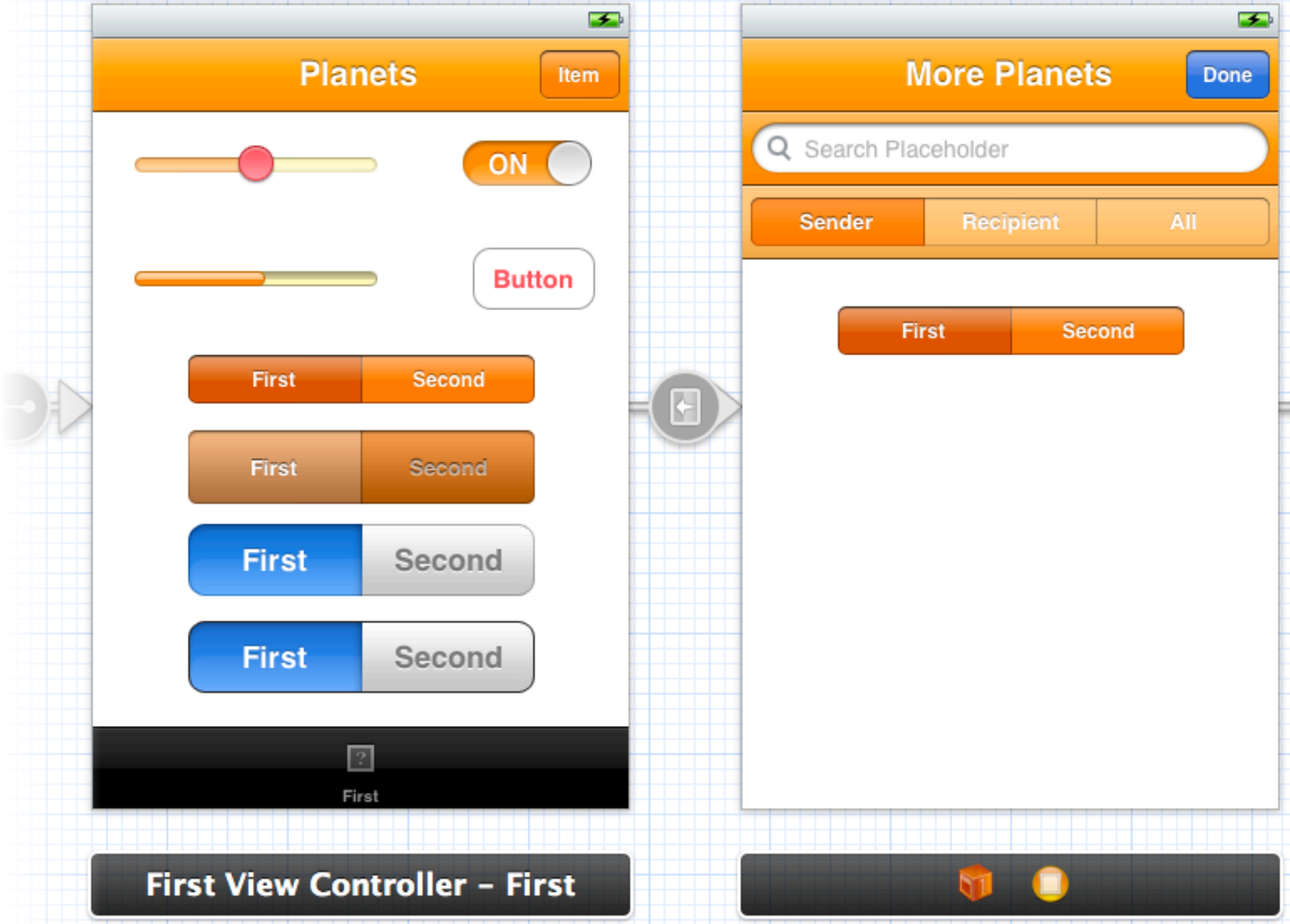


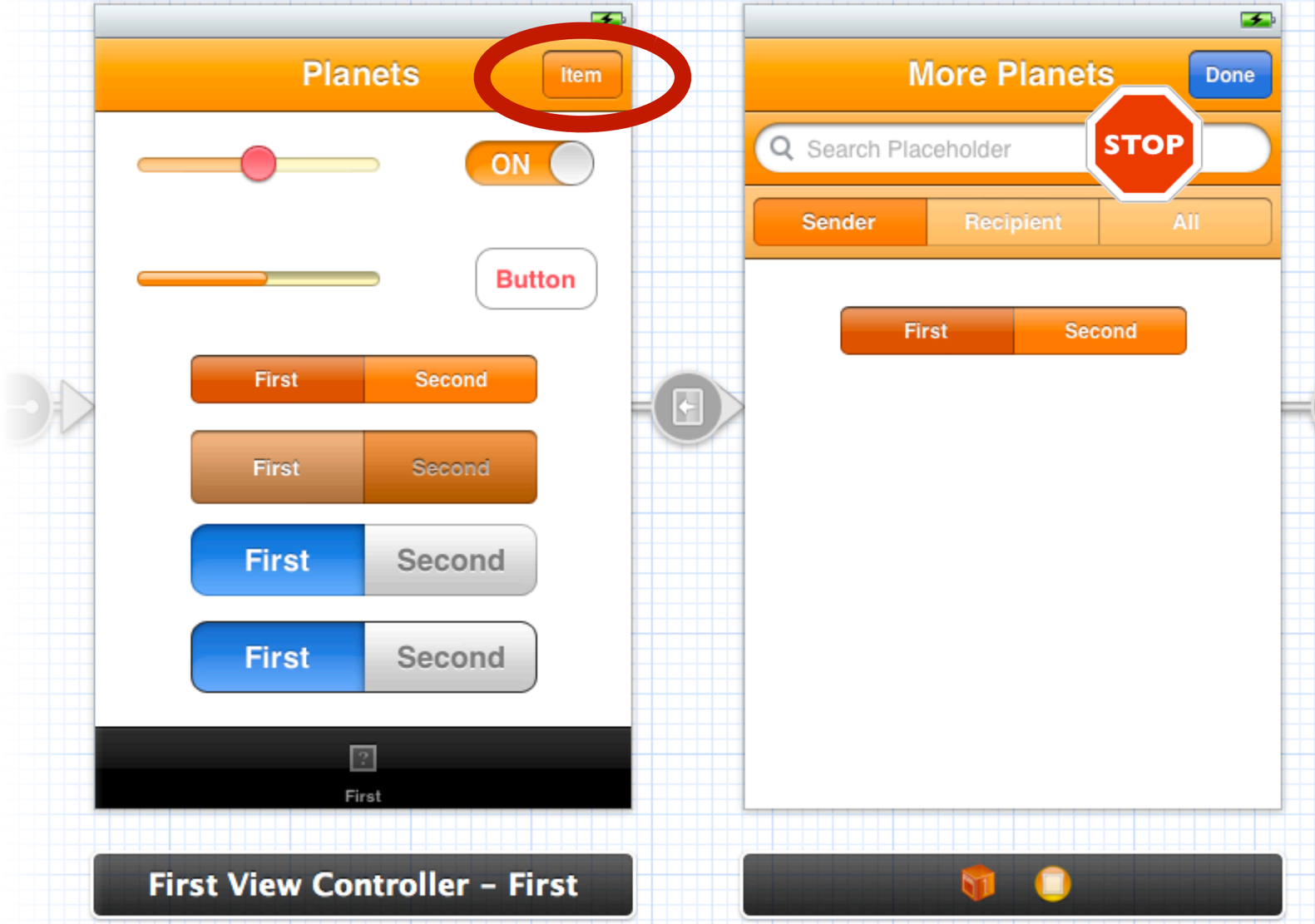
```
[UIBarButtonItem
appearanceWhenContainedIn:
[FTNavigationController
class], nil]
```



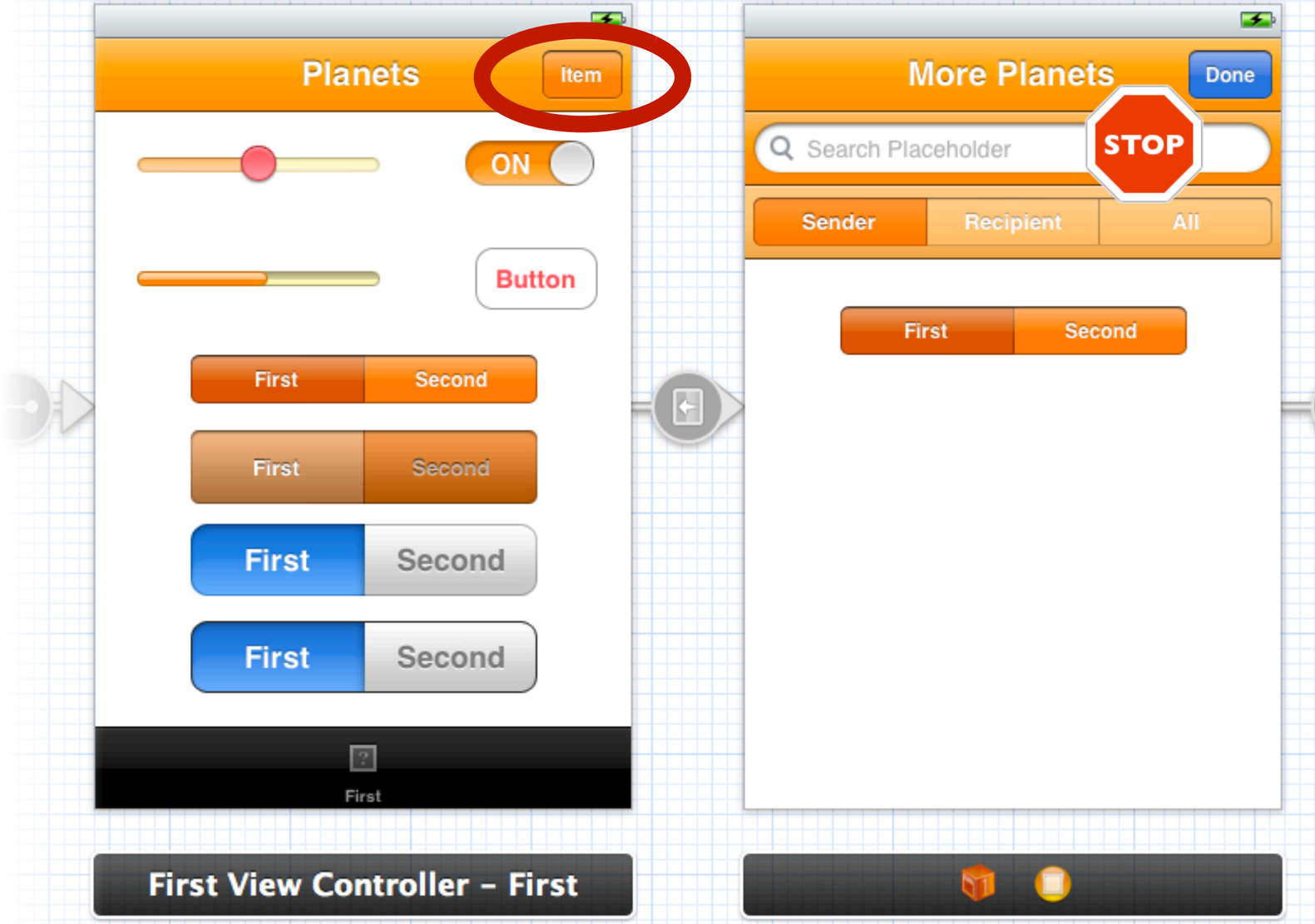
```
[UIBarButtonItem
appearanceWhenContainedIn:
[FTNavigationController
class], nil]
```



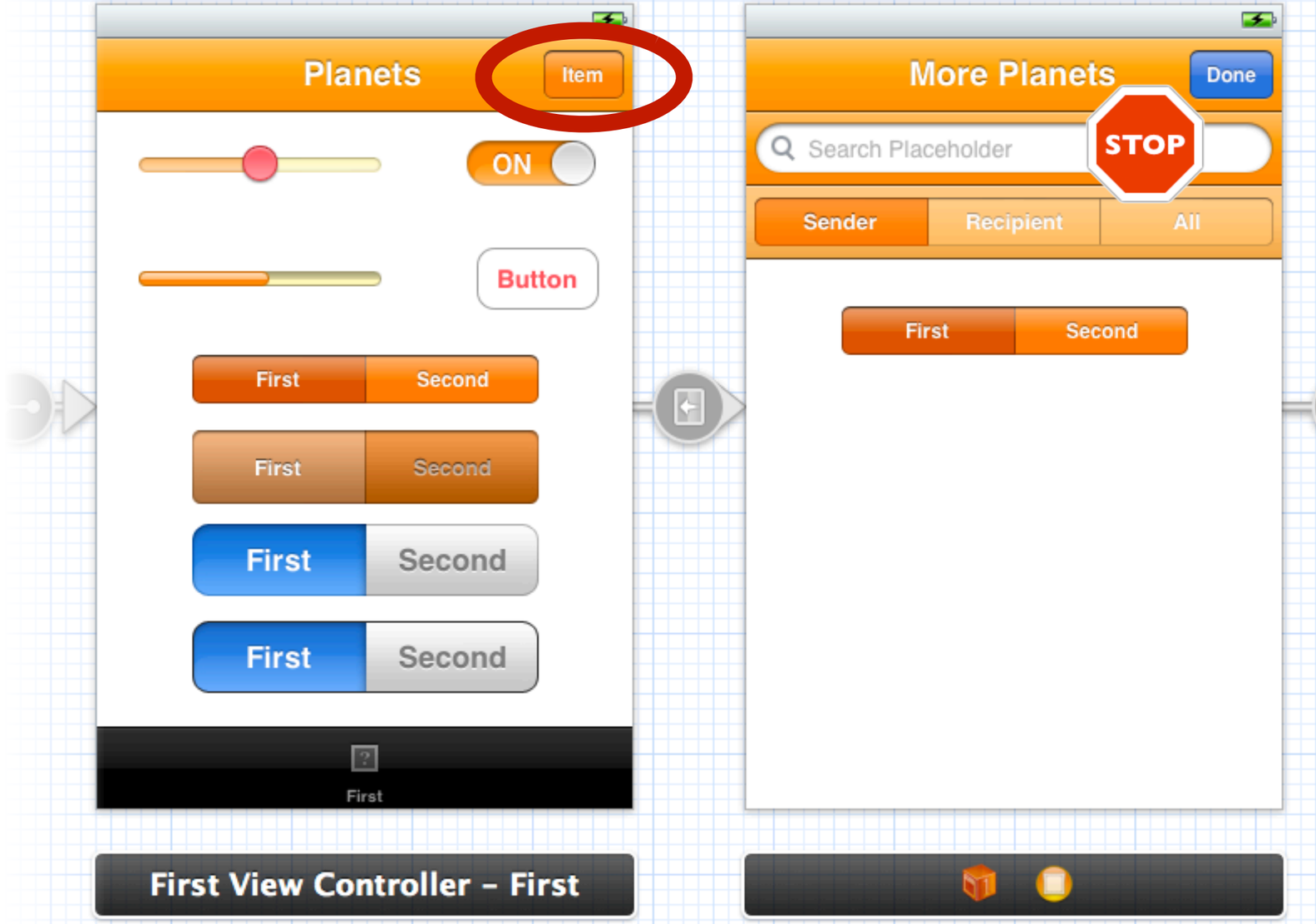

Planets
Item
ON
Button
First
Second
First
Second
First
Second
First
Second
First
More Planets
Done
Search Placeholder
Sender
Recipient
All
First
Second
First View Controller – First


Planets
Item
ON
Button
First
Second
First
Second
First
Second
First
Second
First
More Planets
Done
Search Placeholder
STOP
Sender
Recipient
All
First
Second
First View Controller – First

```
[UIBarButtonItem
appearanceWhenContainedIn:
```

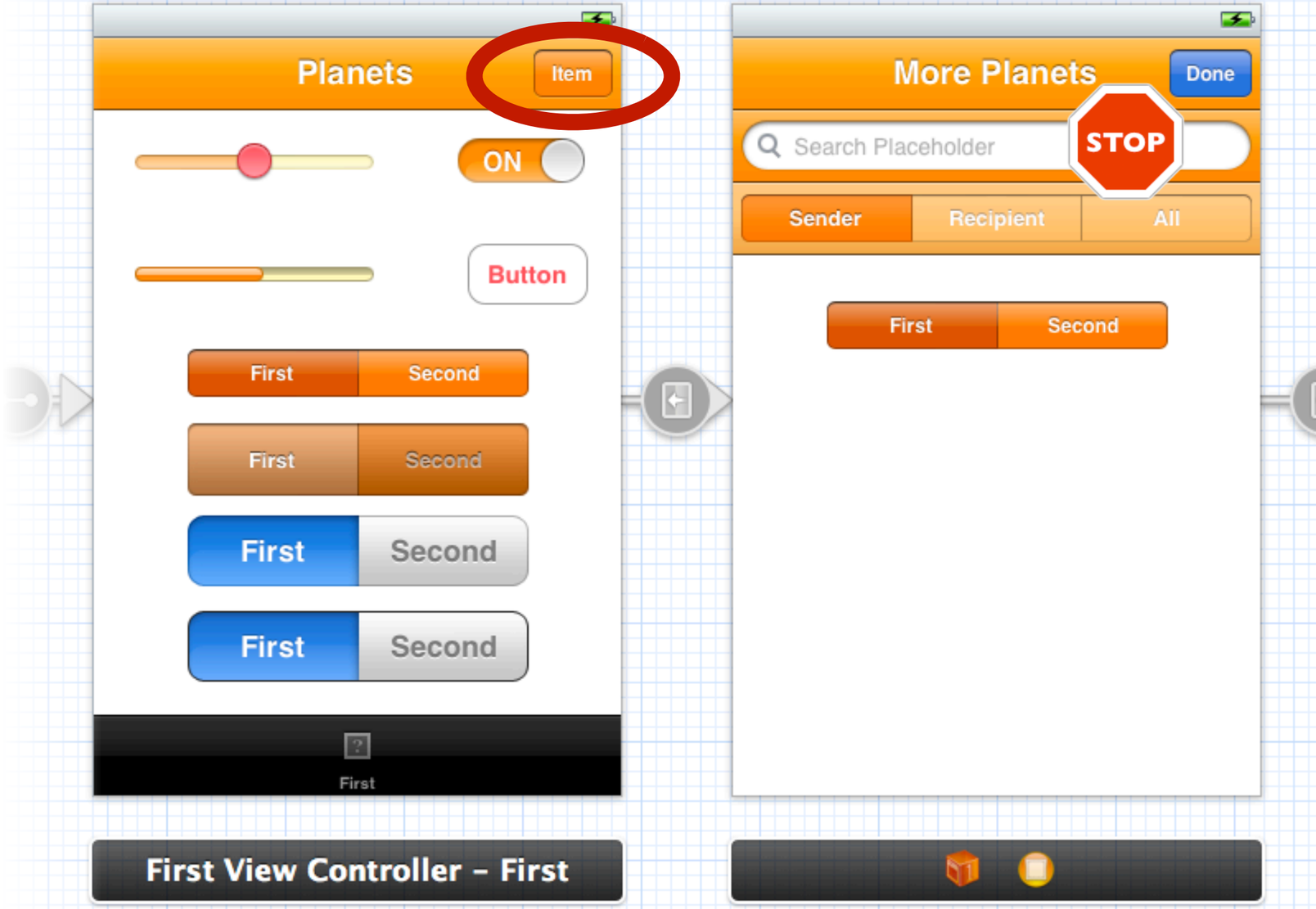


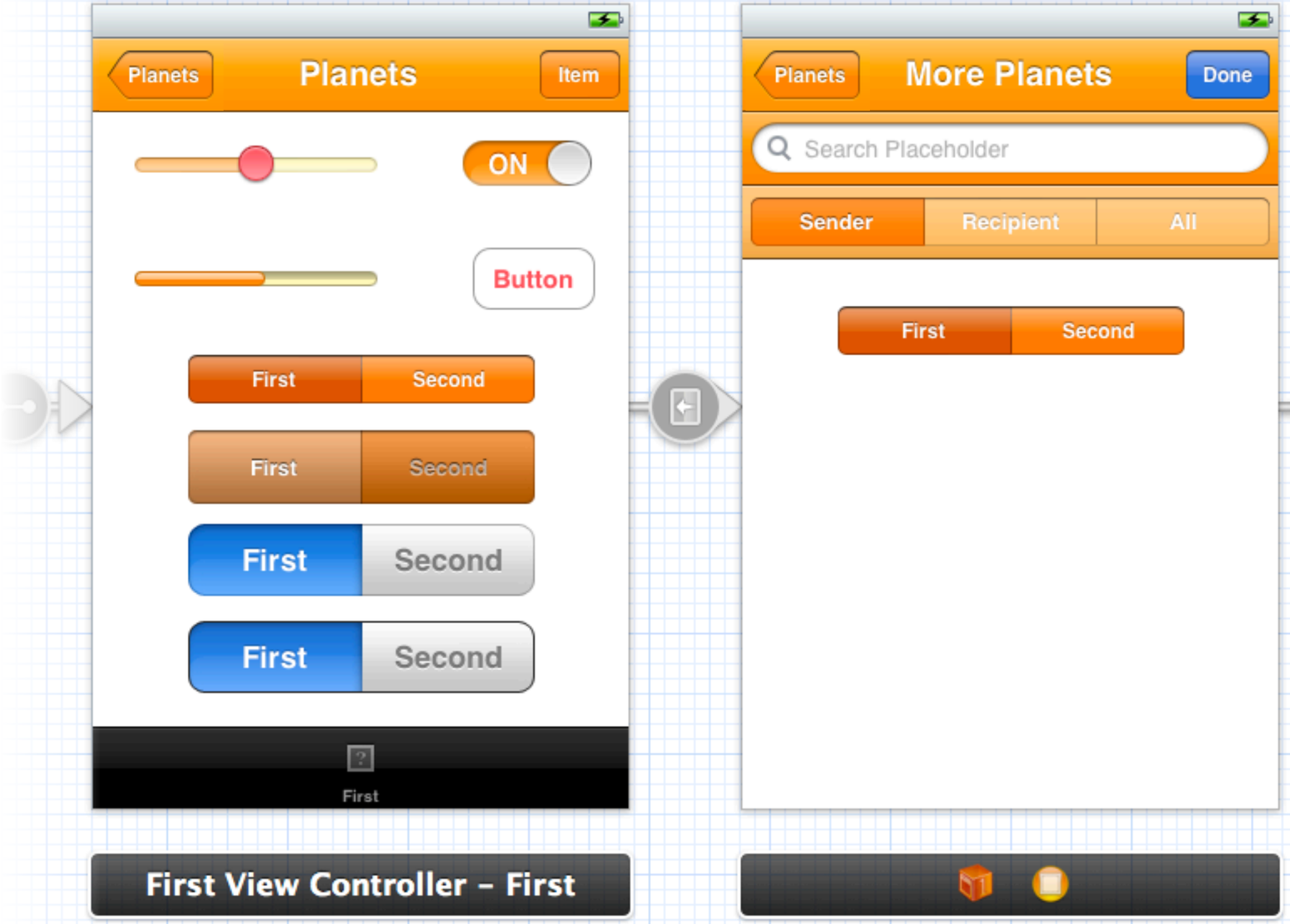
```
[UIBarBu      tem
appearanc     enContainedIn:
```



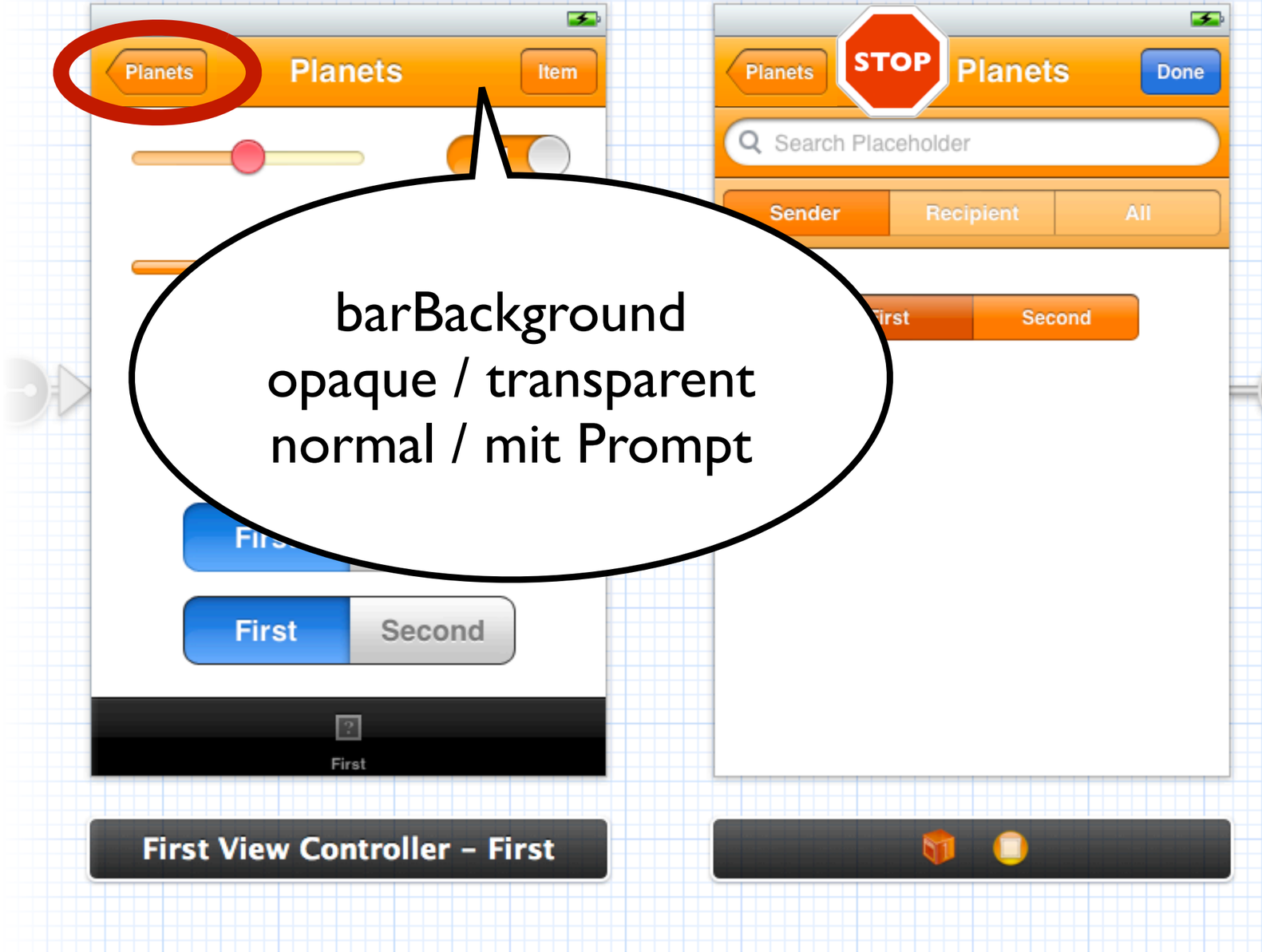
```
[UIBarBu      tem
appearanc     enContainedIn:
```

```
firstViewController.
navigationItem.
rightBarButtonItem…
```

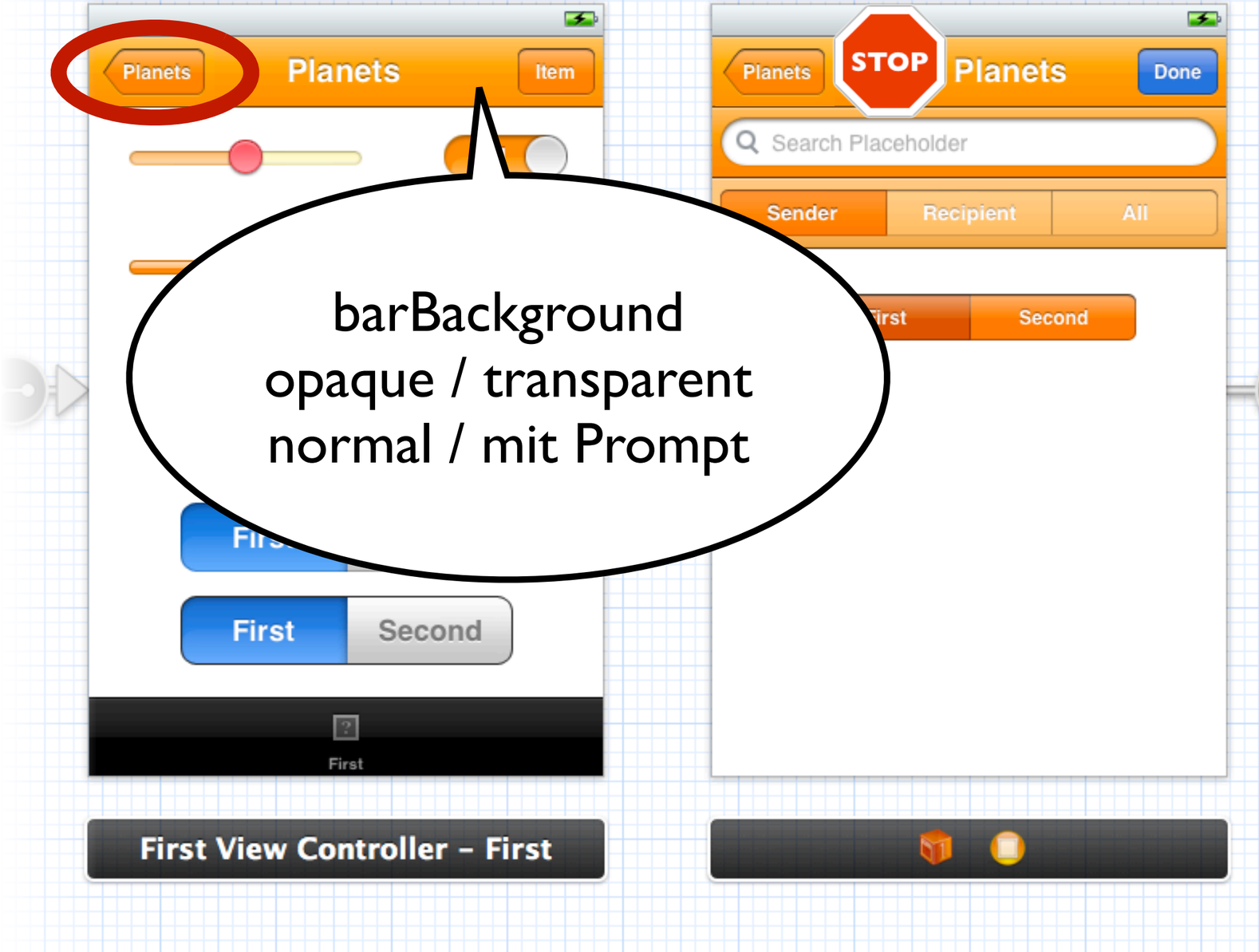



Planets
Planets
Item
ON
Button
First
Second
First
Second
First
Second
First
Second
First
Planets
More Planets
Done
Search Placeholder
Sender
Recipient
All
First
Second
First View Controller – First

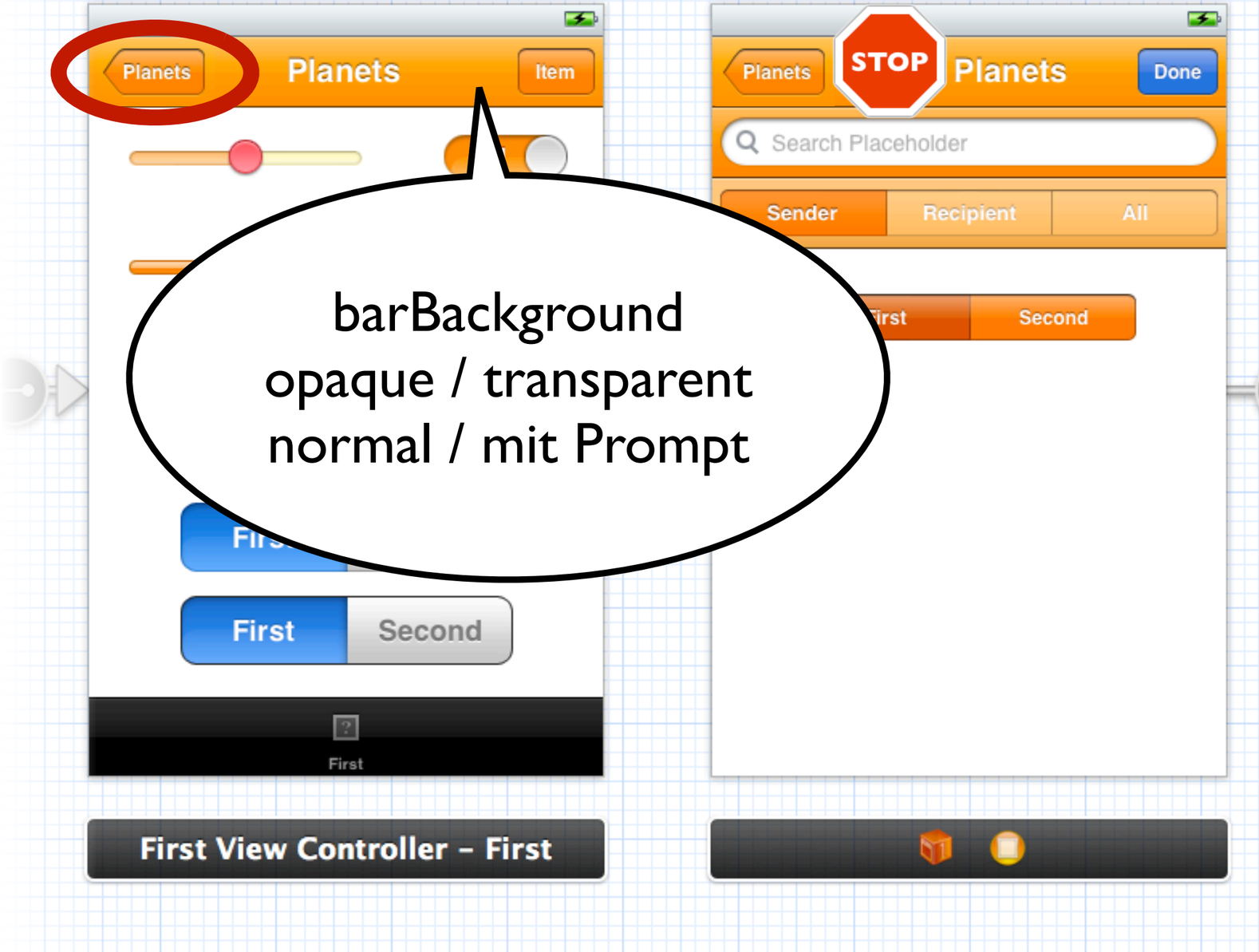

Planets
Planets
Item
ON
Button
First
Second
First
Second
First
Second
First
Second
First
Planets
STOP
Planets
Done
Search Placeholder
Sender
Recipient
All
First
Second
First View Controller – First


Planets
Planets
Item
barBackground
opaque / transparent
normal / mit Prompt
First
Second
First
First View Controller – First
Planets
STOP
Planets
Done
Search Placeholder
Sender
Recipient
All
Second

✗ Appearance Container

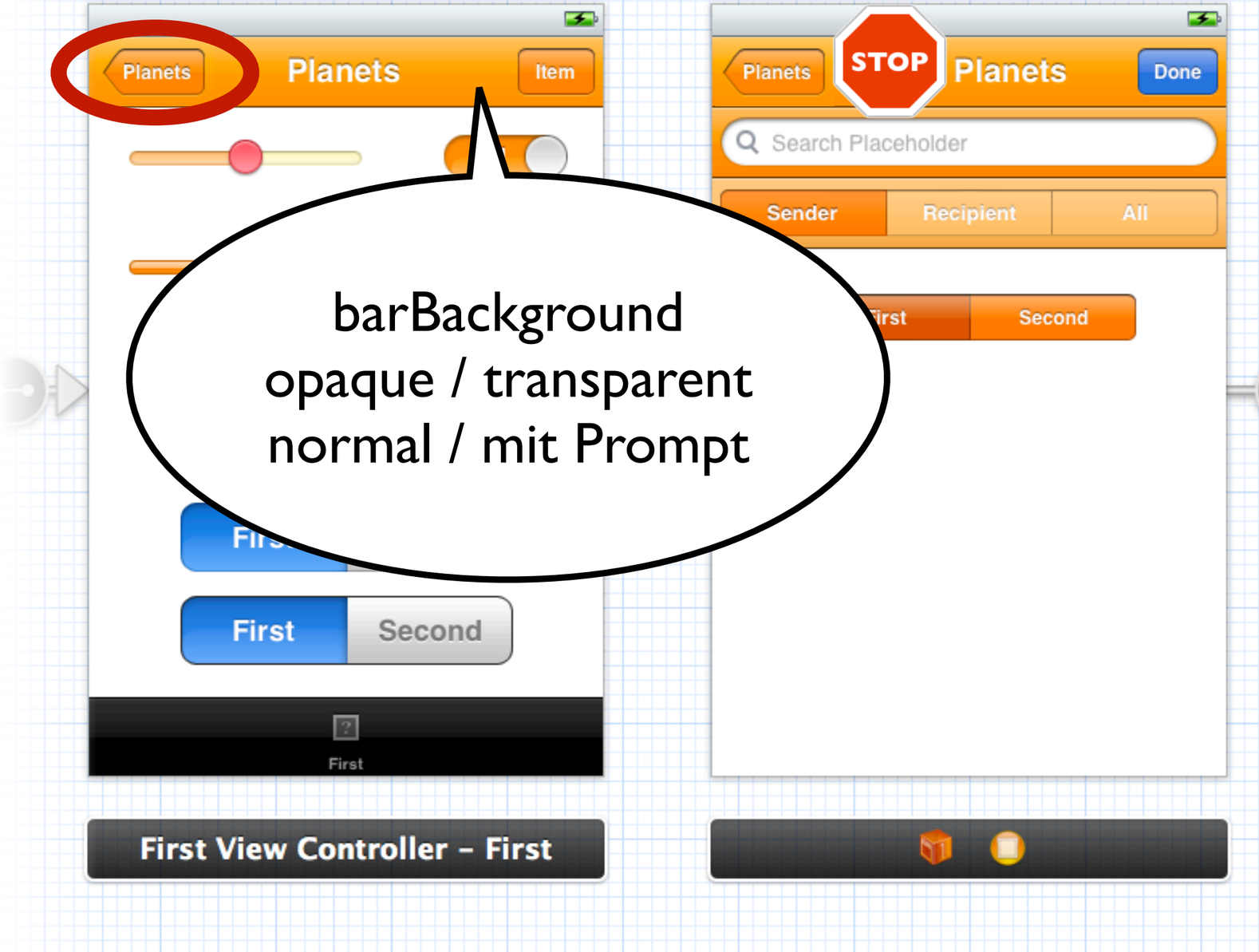


- ✗ Appearance Container
- ✗ Direkte Adressierung



- ✗ Appearance Container
- ✗ Direkte Adressierung

?



Appearance Wunschzettel

- Navigation Bars: mit/ohne Prompt, transparent und opaque
- Erweiterung Containment-Konzept durch universelle Selektoren
- UITableView: sectionIndex, grouped Style
- UIScrollView indicator
- UIPopover
- MKMapView Callouts
- UIActionSheet und UIAlertView
- UIMenuItem

Aber: Kein Vergleich zu vorher!


3G
09:41
Wohin?
Essen & Trinken
Favoriten
Kategorien
Verlauf


09:41
Wohin?
Essen & Trinken
Verlauf
Background Image


09:41
Wohin?
Essen & Trinken
Verlauf
Background Image

3G
09:41
Essen & Trinken
A
Afrikanisch
Asiatisch
Europäisch
Lateinamerikanisch
Nordamerikanisch
Essen
Trinken
Eigene Einträge
Favoriten
Kategorien
Verlauf


09:41
Wohin?
Essen & Trinken
Verlauf
09:41
sen & Trinken
A
kanisch
siatisch
Verlauf
Background Image
Transparente Buttons Unterschiedliche Backgrounds


09:41
Wohin?
Essen & Trinken
Verlauf
Background Image
09:41
sen & Trinken
A
kanisch
siatisch
Verlauf
Transparente Buttons Unterschiedliche Backgrounds
Telekom.de 3G 15:49 66 %
Cafés
Details
1,2 km
Cafe an der Uni
19 ratings
Telefon 089 28986600
Homepage www.cadu.de
Adresse Ludwigstraße 24
80539 München
Deutschland


09:41
Wohin?
Essen & Trinken
Verlauf
09:41
sen & Trinken
A
kanisch
siatisch
Verlauf
1,2 km
19 ratings
Telefon 089 28986600
Homepage www.cadu.de
Adresse Ludwigstraße 24
80539 München
Deutschland
Background Image
Transparente Buttons Unterschiedliche Backgrounds
Custom TableView Background


Background Image
09:41
Wohin?
Essen & Trinken
Verlauf
Transparente Buttons Unterschiedliche Backgrounds
09:41
sen & Trinken
A
kanisch
siatisch
Verlauf
Custom TableView Background
1,2 km
19 ratings
Telefon 089 28986600
Homepage www.cadu.de
Adresse Ludwigstraße 24
80539 München
Deutschland
Netzbetreiber
18:54
Cancel
New Contact
Done
edit
First
Last
Restaurant Monsun
work 06966429933
mobile Phone
home Email
ringtone Default
text tone Default


09:41
Wohin?
Essen & Trinken
Verlauf
Background Image
09:41
sen & Trinken
A
kanisch
siatisch
Verlauf
Transparente Buttons Unterschiedliche Backgrounds
Custom TableView Background
1,2 km
19 ratings
Telefon 089 28986600
Homepage www.cadu.de
Adresse Ludwigstraße 24
80539 München
Deutschland
Netzbetreiber
18:54
Cancel
New Contact
Done
edit
First
Last
Monsun
text tone
Custom System View Controller


09:41
Wohin?
Essen & Trinken
Verlauf
Background Image
Custom TableView Background
1,2 km
Custom System View Controller
Netzbetreiber
18:54
Cancel
New Contact
Done
edit
First
Last
text tone
iOS 3-5
1370
LoC


09:41
Wohin?
Essen & Trinken
Verlauf
Background Image
Custom TableView Background
iOS 3-5
1370
LoC
Netzbetreiber
18:54
Cancel
New Contact
Done
edit
First
Last
Monsun
iOS 5
770
LoC

Vor iOS 5

- Nachbauen von Controls (z.B. Segmented Control)
- Patchen von System-Views (Navigation Bars)
- Projekt appearance44 von Tobias Klonk, @tonklon

Patchen von System-Views

```
UITableView

-(void)layoutSubviews
```

Patchen von System-Views

```
UITableView

-(void)layoutSubviews
```

```
UITableView(Custom)

-(void)layoutSubviews {
  // do your stuff

  // call original implementation
}
```

Patchen von System-Views

```
UITableView

-(void)layoutSubviews
```

```
UITableView(Custom)

-(void)layoutSubviews {
  // do your stuff

  [super layoutSubviews];
}
```

Patchen von System-Views

```
UITableView

-(void)layoutSubviews
```

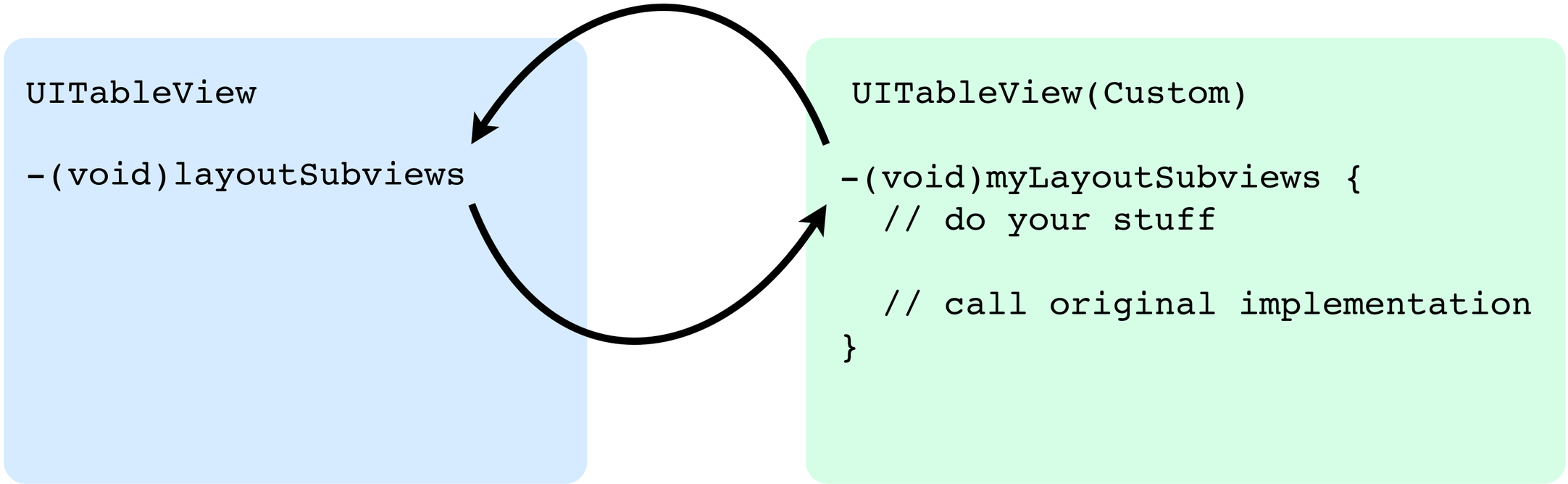
```
UITableView(Custom)

-(void)layoutSubviews {
  // do your stuff

  [super layoutSubviews];
}
```

Method Swizzling

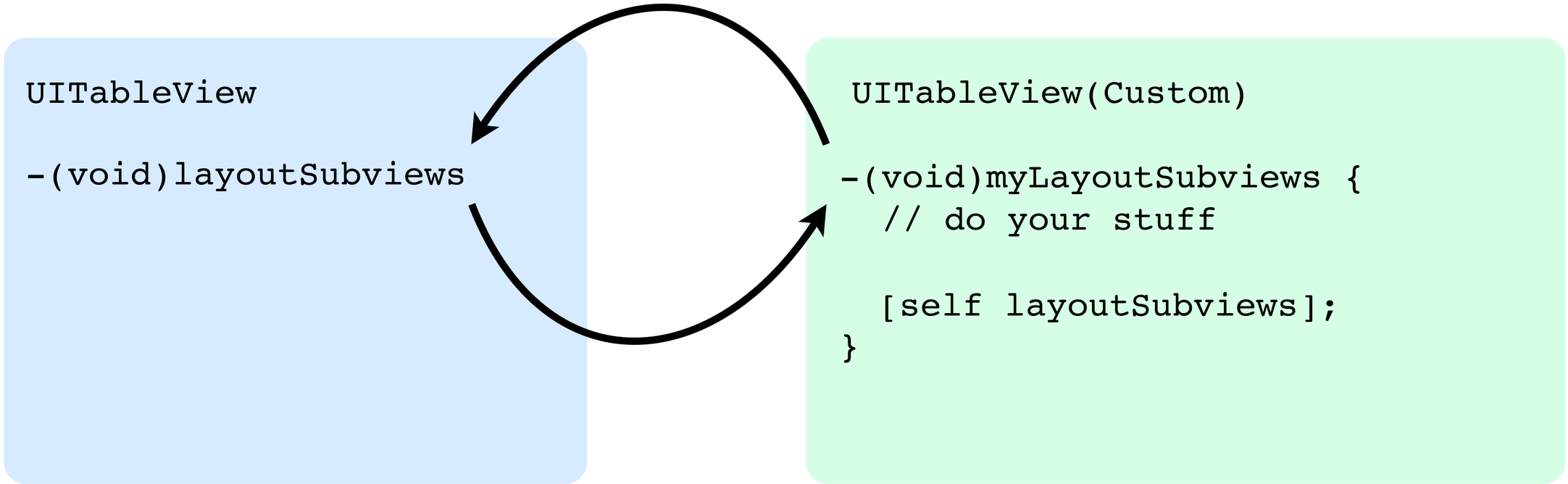
```
UITableView

-(void)layoutSubviews
```

```
UITableView(Custom)

-(void)myLayoutSubviews {
  // do your stuff

  // call original implementation
}
```

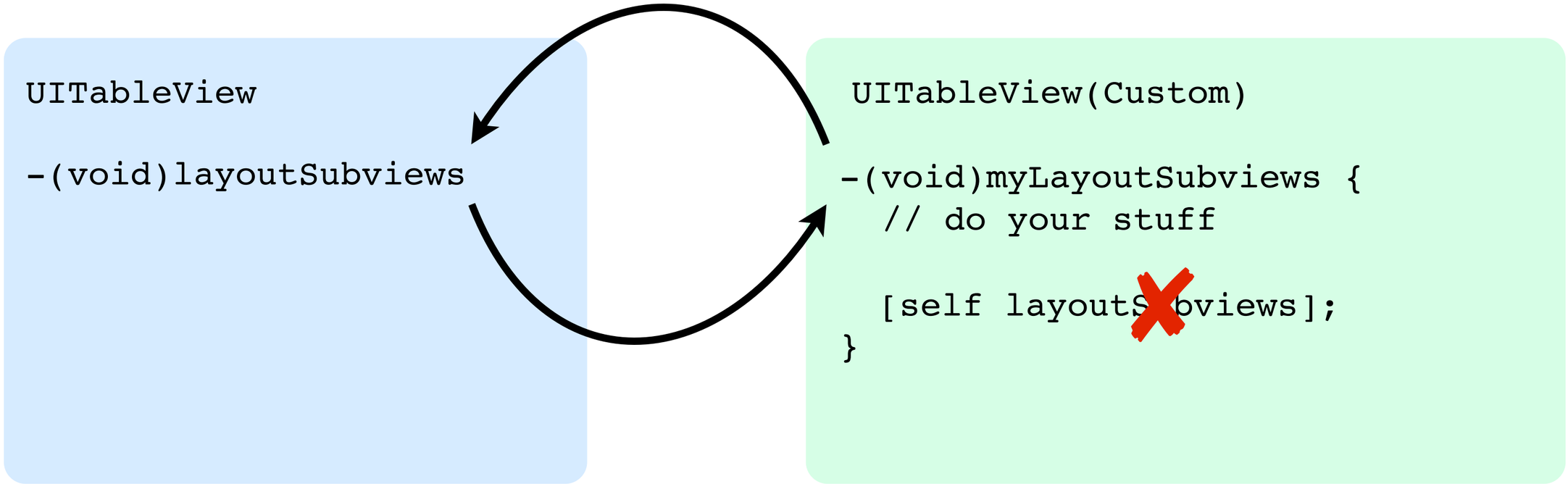
Method Swizzling



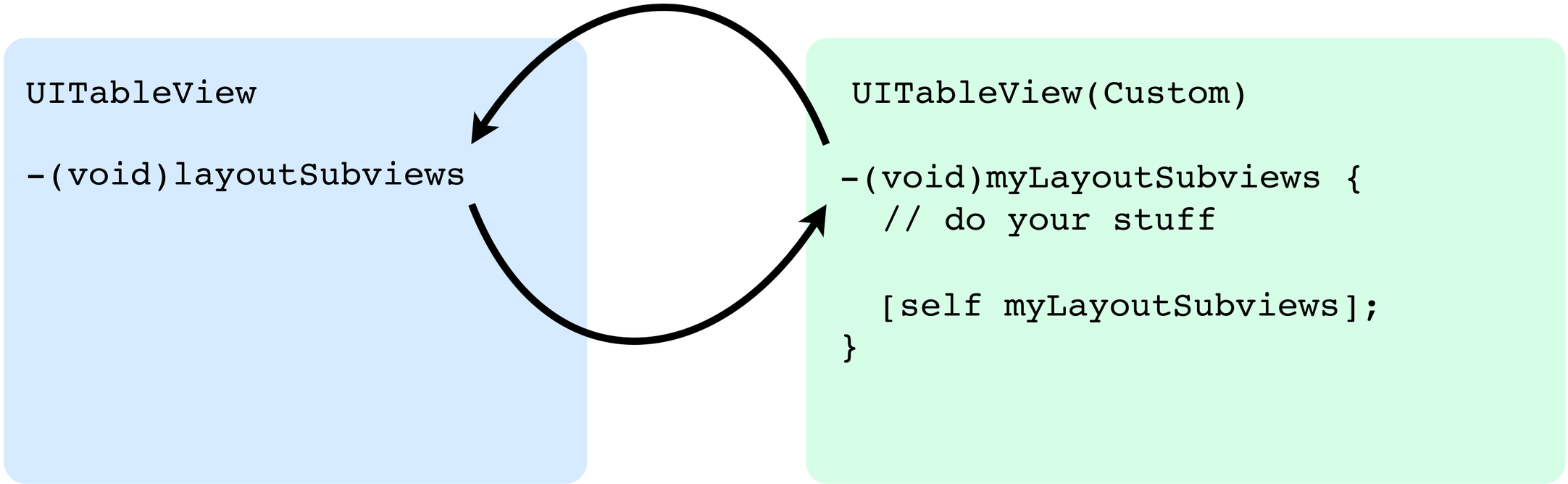
Method Swizzling



Method Swizzling



Method Swizzling



```
// implementation by Mike Ash
// http://cocoadev.com/index.pl?MethodSwizzling (down to bottom!)

void Swizzle(Class c, SEL orig, SEL new) {

  Method origMethod = class_getInstanceMethod(c, orig);
  Method newMethod = class_getInstanceMethod(c, new);

  if(class_addMethod(c, orig, method_getImplementation(newMethod),
method_getTypeEncoding(newMethod))) {
    class_replaceMethod(c, new, method_getImplementation(origMethod),
method_getTypeEncoding(origMethod));
  } else {
    method_exchangeImplementations(origMethod, newMethod);
  }
}
```

UITableView

Custom Background

UITableView

Custom Background

tableHeaderView

UITableView

Custom Background

```
+ [UIColor colorWithPatternImage:]
```

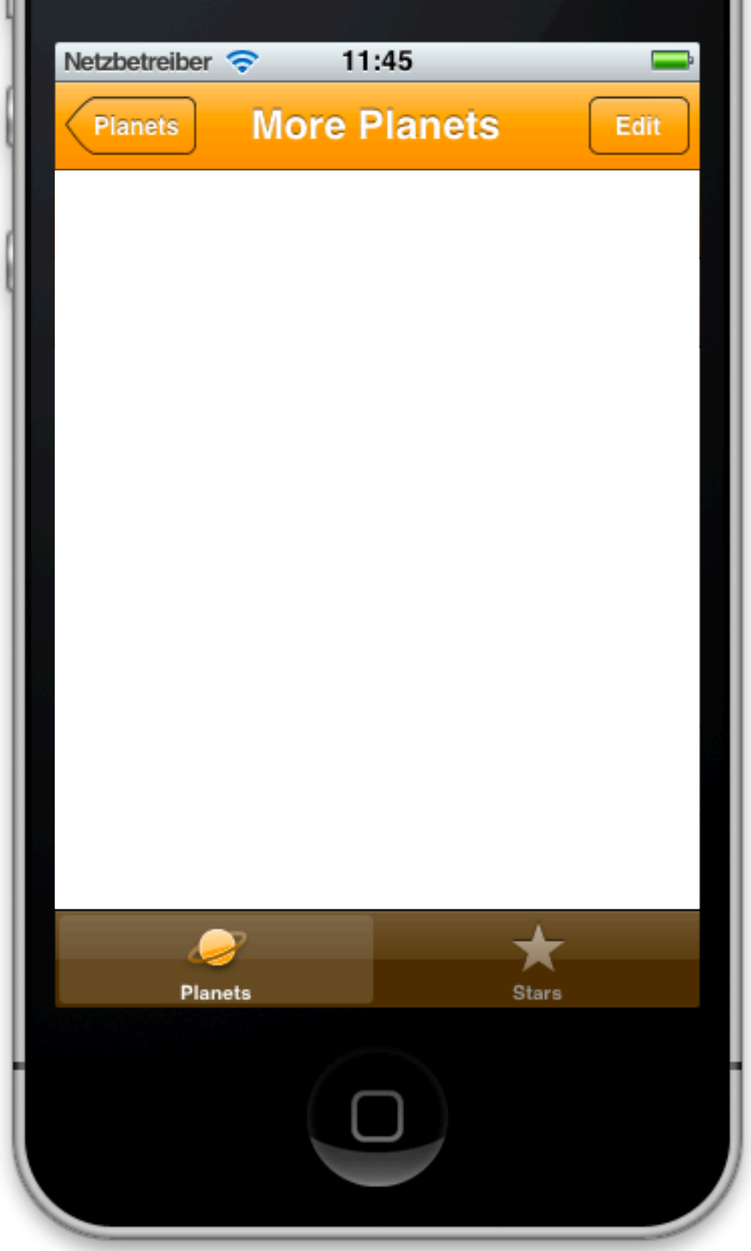
tableHeaderView

```
Swizzle([UITableView class], @selector(layoutSubviews), @selector
(myLayoutSubviews));

@implementation UITableView (FTCustom)
- (void)myLayoutSubviews {
    if (self.style == UITableViewStyleGrouped) {
        UIColor *bgColor = [UIColor groupedTableViewBackgroundColor];
        self.backgroundColor = bgColor;
        self.tableHeaderView.backgroundColor = bgColor;
        self.tableFooterView.backgroundColor = bgColor;
    }
    [self myLayoutSubviews]; // call the original implementation
}
@end
```

```
Swizzle([UITableView class], @selector(layoutSubviews), @selector
(myLayoutSubviews));

@implementation UITableView (FTCustom)
- (void)myLayoutSubviews {
    if (self.style == UITableViewStyleGrouped) {
        UIColor *bgColor = [UIColor groupedTableViewBackgroundColor];
        self.backgroundColor = bgColor;
        self.tableHeaderView.backgroundColor = bgColor;
        self.tableFooterView.backgroundColor = bgColor;
    }
    [self myLayoutSubviews]; // call the original implementation
}
@end
```

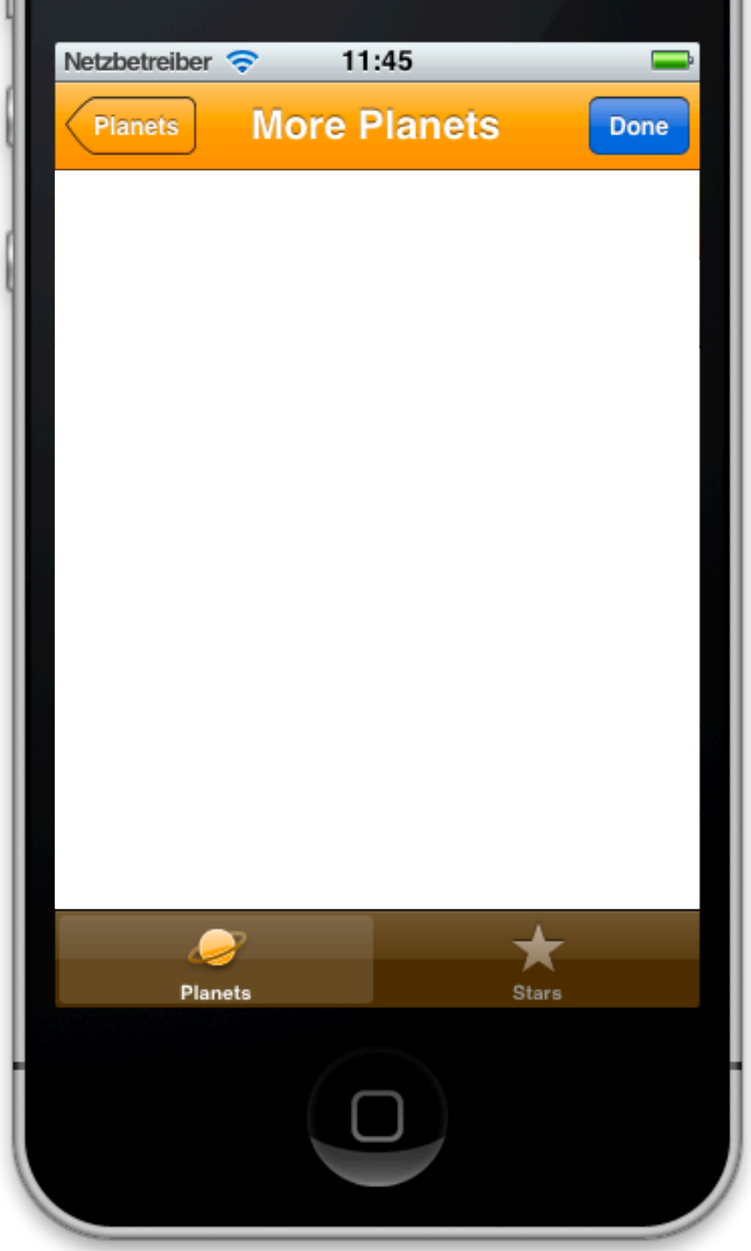
UIBarButtonItem

Custom Background Images für Normal/Done



UIBarButtonItem

Custom Background Images für Normal/Done



```
@implementation UIBarButtonItem (FTCustom)

- (void)mySetStyle:(UIBarButtonItemStyle)newStyle {
   UIImage *backgroundImage = (newStyle == UIBarButtonItemStyleDone) ?
     doneImage : normalImage;

   [self setBackgroundImage:backgroundImage
                   forState:UIControlStateNormal
                 barMetrics:UIBarMetricsDefault];
   // optionally set the highlighted state, too

   [self mySetStyle:newStyle]; // original implementation
}
```

```
@implementation UIBarButtonItem (FTCustom)

- (void)mySetStyle:(UIBarButtonItemStyle)newStyle {
    UIImage *backgroundImage = (newStyle == UIBarButtonItemStyleDone) ?
      doneImage : normalImage;

    [self setBackgroundImage:backgroundImage
                    forState:UIControlStateNormal
                  barMetrics:UIBarMetricsDefault];
    // optionally set the highlighted state, too

    [self mySetStyle:newStyle]; // original implementation
}
```

```
- (id)initWithBarButtonSystemItem_My:(UIBarButtonSystemItem)
systemItem target:(id)target action:(SEL)action {
    self = [self initWithBarButtonSystemItem_My:systemItem
      target:target action:action]; // original implementation

    if (UIBarButtonSystemItemDone == systemItem ||
      UIBarButtonSystemItemSave == systemItem) {
        self.style = UIBarButtonItemStyleDone; // call -mySetStyle:
    }
    return self;
}
```

```
- (id)initWithBarButtonSystemItem_My:(UIBarButtonSystemItem)
systemItem target:(id)target action:(SEL)action {
    self = [self initWithBarButtonSystemItem_My:systemItem
      target:target action:action]; // original implementation

    if (UIBarButtonSystemItemDone == systemItem ||
      UIBarButtonSystemItemSave == systemItem) {
        self.style = UIBarButtonItemStyleDone; // call -mySetStyle:
    }
    return self;
}

- (void)myAwakeFromNib {
    [self myAwakeFromNib]; // original implementation
    self.style = self.style; // call -mySetStyle:
}
```

Demo-Projekt

- Bald auf GitHub
- Follow @futuretap

Resumé

- tolle neue Möglichkeiten
- große Erleichterung für Standard Views und -Controls
- Einschränkungen bei Navigation Bars
- gut testen, teilweise überraschende Bugs
- viele Views sind noch nicht customizable
- Method Swizzling immer noch nötig

Fragen?

Ortwin Gentz
gentz@futuretap.com
@futuretap

Vielen Dank