

Cameron **Barrie**

 Client Principal at **Bilue**

 cameron@bilue.com.au

 @whalec

Mobile at **Warp** Speed



*If you're not **disrupting**,
you're likely being disrupted*

A **quick** survey



*Worst upgrade ever!!!! Now I can't even log in,
most useless app out of all the banks, really
angry about this can't even transfer money
let alone opening the app. FIX IMMEDIATELY*

– JJ&LJ2013

Tried to move fast



*We don't read the reviews anymore,
the only people who leave reviews
are the **people who complain.***

–Anon friend

*I have spent the last week trying to get up to the bank to order a new card. I looked on the app and it wasn't there, updated today and wow, how convenient I am now able to order another card through the app. **You guys sure make my life easier.** Very pleased with this new update...*

– Serenity-1105

Can move fast



How do we get there?



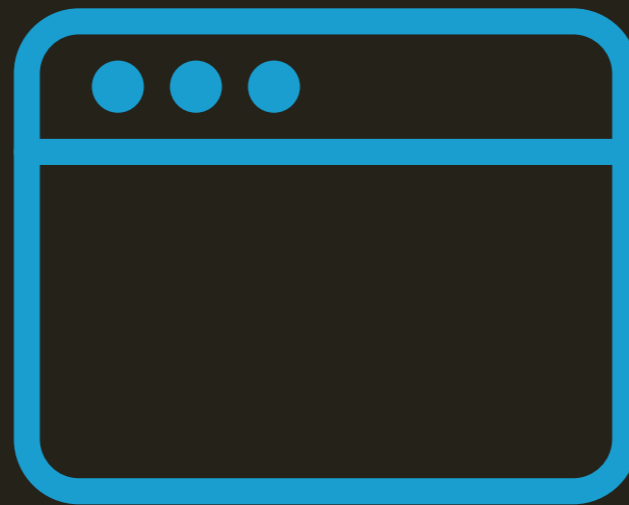
Some **Prerequisites**

- Can you make decisions quickly?
- Are you using source control(I know right)?
- Are you doing Test Driven Development?
- Does the business understand the risk/reward?
- Do you have genuine buy in from the business?

Why is it a challenge?



It's **not** the web



Can't just **refresh**



Walled Garden



5 day **release** cycle



1–2 days for expedited



No update guarantees



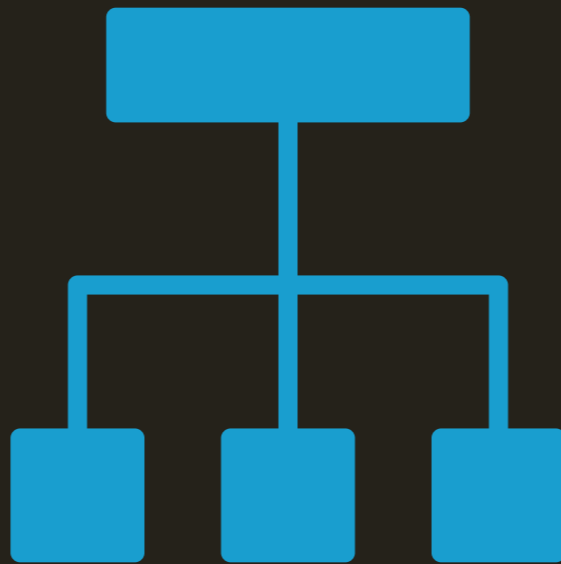
Forced updates are bad



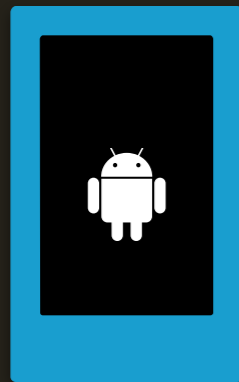
On going **support**



Business Logic



The Challenge

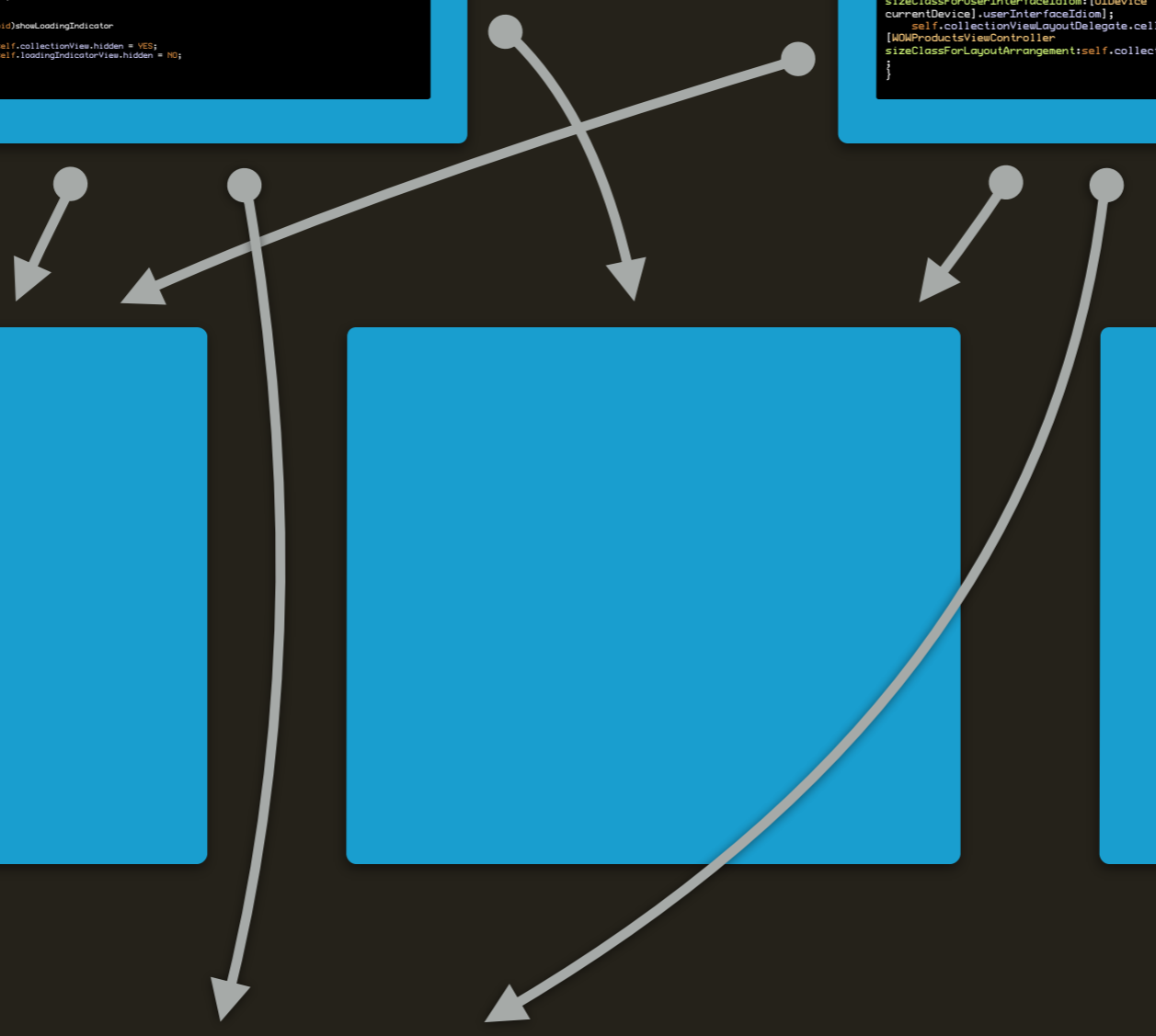
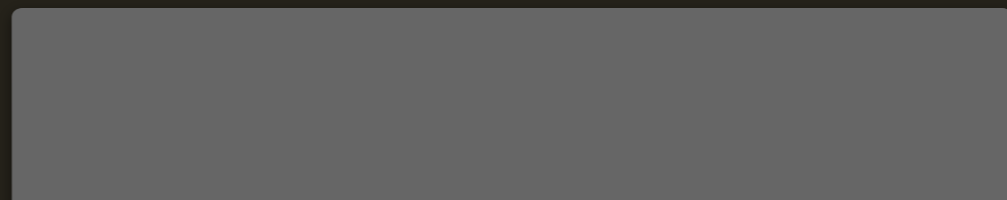
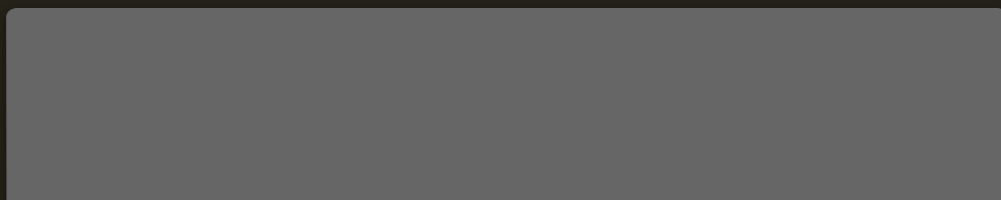
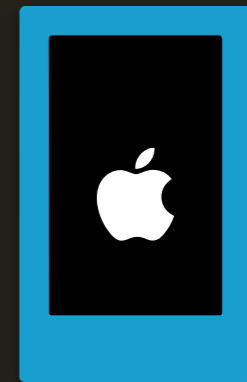


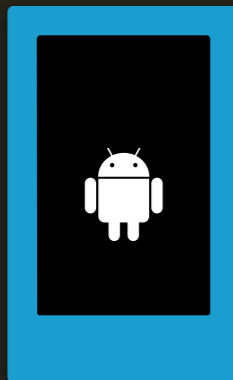
```
- (void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];
    self.barAnimatingScrollViewDelegate.animatableHeaderBar = (UIView <QWAnimatableBar> *)self.navigationController.navigationBar;
}

- (void)fetchData
{
    [self showLoadingIndicator];
    [self.productsDataStore fetchDataWithRequestURL:nil errorHandler:^(NSError *error) {
        [self.dataStore:self.productsDataStore didReceiveError:error];
    }];
}

- (void)showLoadingIndicator
{
    self.collectionView.hidden = YES;
    self.loadingIndicatorView.hidden = NO;
}
```

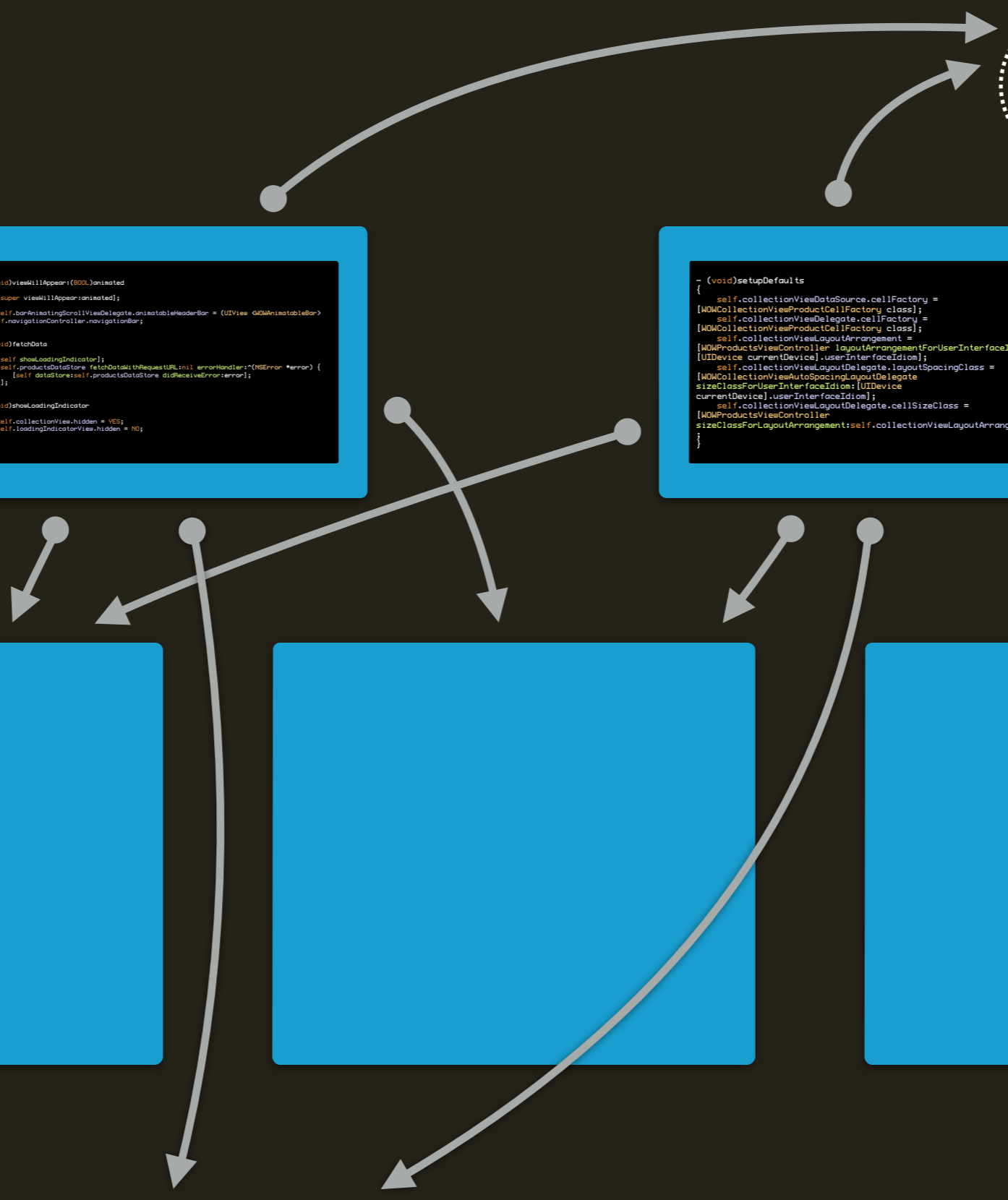
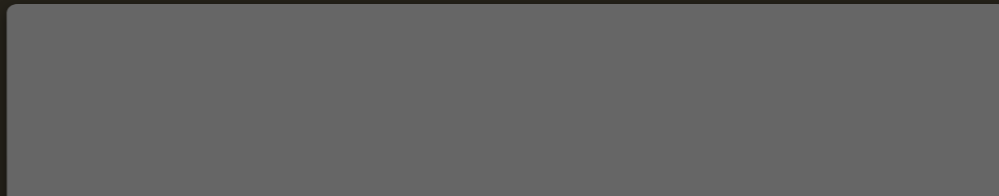
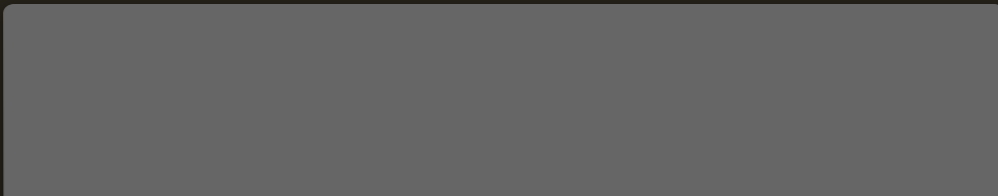
```
- (void)setupDefaults
{
    self.collectionViewDataSource.cellFactory =
    [QWCollectionViewProductCellFactory class];
    self.collectionViewDelegate.cellFactory =
    [QWCollectionViewProductCellFactory class];
    self.collectionViewLayoutArrangement =
    [QWProductsViewController layoutArrangementForUserInterfaceIdiom:
    [UIDevice.currentDevice].userInterfaceIdiom];
    self.collectionViewLayoutDelegate.layoutSpacingClass =
    [QWCollectionViewAutoSpacingLayoutDelegate
    sizeClassForUserInterfaceIdiom:[UIDevice
    currentDevice].userInterfaceIdiom];
    self.collectionViewLayoutDelegate.cellSizeClass =
    [QWProductsViewController
    sizeClassForLayoutArrangement:self.collectionViewLayoutArrangement];
}
```





```
- (void)viewWillAppear:(BOOL)animated  
{  
    [super viewWillAppear:animated];  
    self.barAnimatingScrollViewDelegate.animatableHeaderBar = (UIView <QWAnimatableBar>  
*)self.navigationController.navigationBar;  
}  
  
- (void)fetchData  
{  
    [self showLoadingIndicator];  
    [self.productsDataStore fetchDataWithRequestURL:nil errorHandler:^(NSError *error) {  
        [self.dataStore:self.productsDataStore dismissiveError:error];  
    }];  
}  
  
- (void)showLoadingIndicator  
{  
    self.collectionView.hidden = YES;  
    self.loadingIndicatorView.hidden = NO;  
}
```

```
- (void)setupDefaults  
{  
    self.collectionViewDataSource.cellFactory =  
    [QWCollectionViewProductCellFactory class];  
    self.collectionViewDelegate.cellFactory =  
    [QWCollectionViewProductCellFactory class];  
    self.collectionViewLayoutArrangement =  
    [QWProductsViewController layoutArrangementForUserInterfaceIdiom:  
    [UIDevice.currentDevice].userInterfaceIdiom];  
    self.collectionViewLayoutDelegate.layoutSpacingClass =  
    [QWCollectionViewAutoSpacingLayoutDelegate  
    sizeClassForUserInterfaceIdiom:[UIDevice  
    currentDevice].userInterfaceIdiom];  
    self.collectionViewLayoutDelegate.cellSizeClass =  
    [QWProductsViewController  
    sizeClassForLayoutArrangement:self.collectionViewLayoutArrangement]  
};
```



Recipe for **disaster**



Remove **Business** Logic



Remove **Business** Logic

/post.json

```
{  
  title: "Stop killing unicorns",  
  user_id: 67  
}
```

Remove **Business** Logic

BILPost.m

```
@implementation BILPost
```

```
- (instancetype)initWithDictionary:(NSDictionary *)dictionary {  
    self = [self init];  
    if (self) {  
        self.title = dictionary[@"title"];  
        NSString *urlString = [NSString stringWithFormat:@"%s/users/%s.json",  
                                kAPIBaseURL,  
                                dictionary[@"user_id"]];  
        self.userURL = [NSURL URLWithString:urlString];  
    }  
    return self;  
}
```

Remove **Business** Logic

BILPost.m

```
@implementation BILPost

- (instancetype)initWithDictionary:(NSDictionary *)dictionary {
    self = [self init];
    if (self) {
        self.title = dictionary[@"title"];
        NSString *urlString = [NSString stringWithFormat:@"%s/users/%s.json",
                               kAPIBaseURL,
                               dictionary[@"user_id"]];
        self.userURL = [NSURL URLWithString:urlString];
    }

    return self;
}
```

Craft APIs

/post.json

```
{  
  title: "Stop killing unicorns",  
  user_id: 67  
}
```

Craft APIs

/post.json

```
{  
  title: "Stop killing unicorns",  
  user_id: 67  
}
```

Use hypermedia

/post.json

```
{  
  title: "Stop killing unicorns",  
  user: "http://service.com/api/v2/users/1.json"  
}
```

Use hypermedia

/post.json

```
{  
  title: "Stop killing unicorns",  
  user: "http://service.com/api/v2/users/1.json"  
}
```


Remove **Business** Logic

BILPost.m

```
@implementation BILPost
```

```
- (instancetype)initWithDictionary:(NSDictionary *)dictionary {  
    self = [self init];  
    if (self) {  
        self.title = dictionary[@"title"];  
        NSString *urlString = [NSString stringWithFormat:@"%s/users/%s.json",  
                               kAPIBaseURL,  
                               dictionary[@"user_id"]];  
        self.userURL = [NSURL URLWithString:urlString];  
    }  
    return self;  
}
```

Remove **Business** Logic

BILPost.m

```
@implementation BILPost
```

```
- (instancetype)initWithDictionary:(NSDictionary *)dictionary {  
    self = [self init];  
    if (self) {  
        self.title = dictionary[@"title"];  
        self.userURL = [NSURL URLWithString:dictionary[@"user"]];  
    }  
  
    return self;  
}
```

Use hypermedia

/post.json

```
{  
  title: "Stop killing unicorns",  
  user: "http://service.com/api/v2/users/1.json"  
}
```

Update **anytime**

/post.json

```
{  
  title: "Stop killing unicorns",  
  user: "https://api.v2.users.service.com/cbarrie"  
}
```

Version your APIs



Version APIs

v2/post.json

```
{  
  title: "Stop killing unicorns",  
  user: "https://api.v2.users.service.com/cbarrie"  
}
```

Version APIs

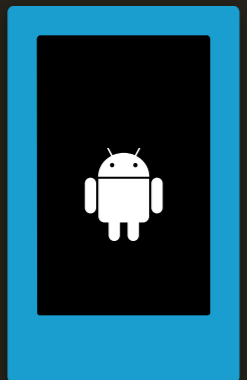
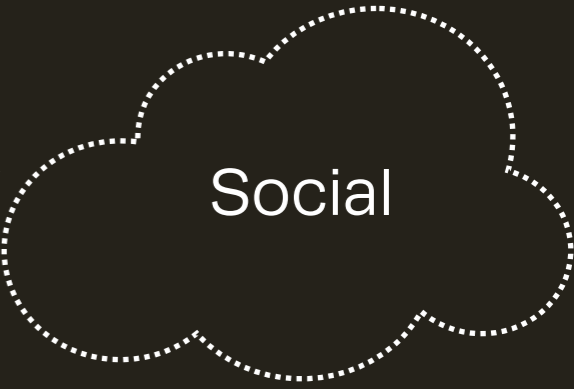
v3/post.json

```
{
  title: "Stop killing unicorns",
  user: {
    name: "Ron Burgundy",
    image: "http://cdn.com/someimage.png",
    href: "https://user.api.service.com/v3/1"
  }
}
```

Version APIs

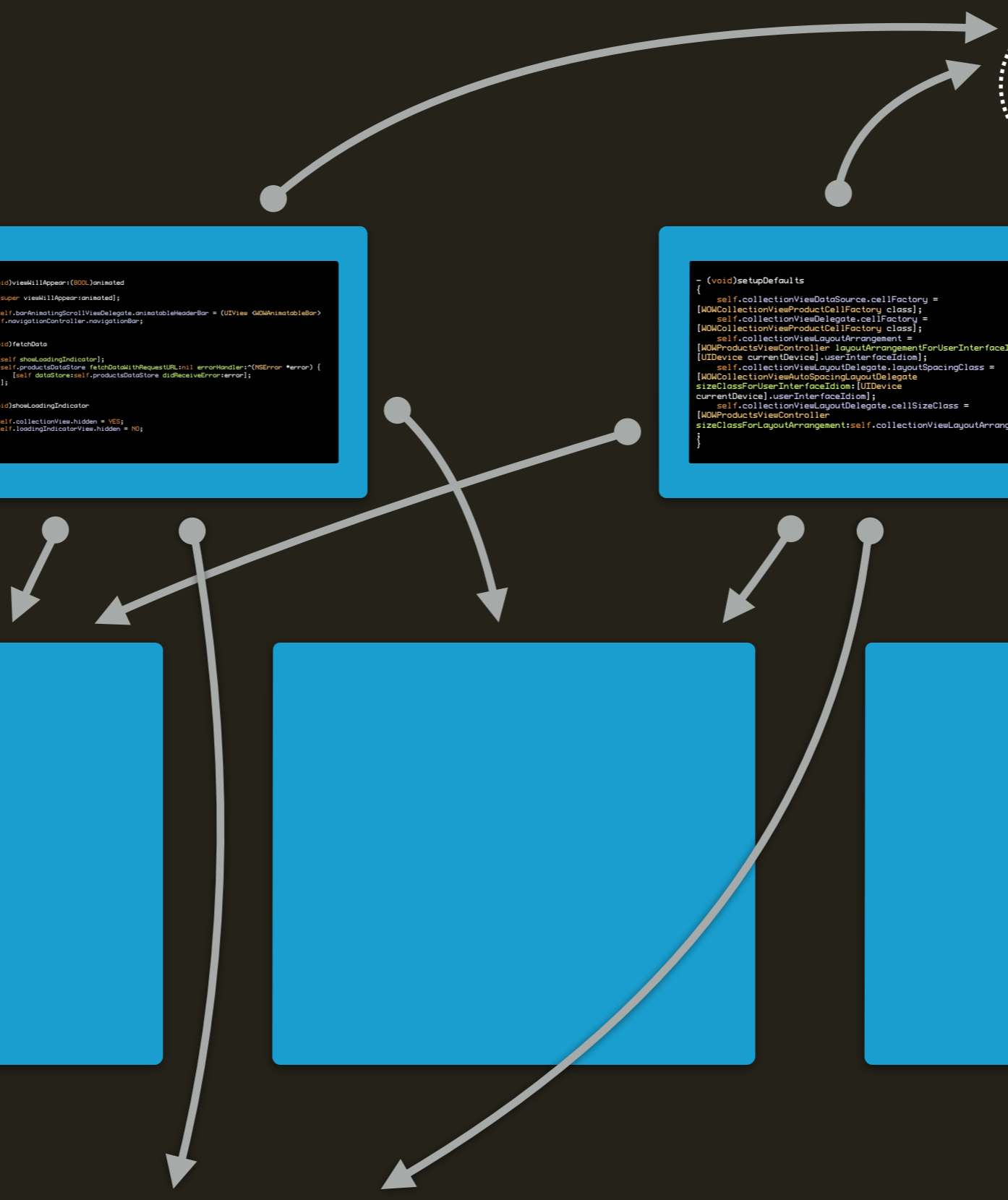
v2/post.json

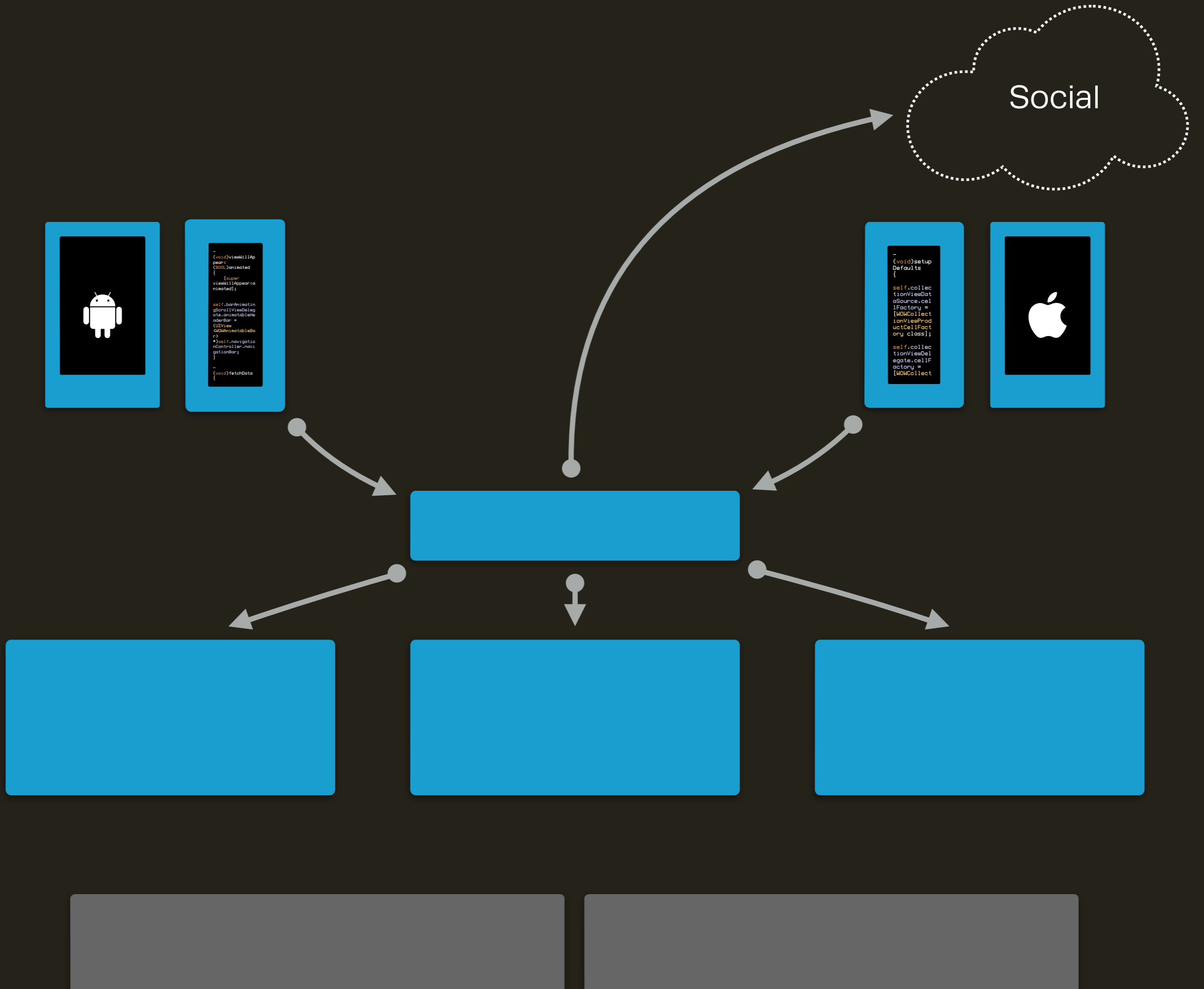
```
{
  title: "Stop killing unicorns",
  user: "https://api.v2.users.service.com/cbarrie",
  userSummary: {
    name: "Brick Tamland",
    image: "http://cdn.com/someimage.png"
  }
}
```

```
- (void)viewWillAppear:(BOOL)animated  
{  
    [super viewWillAppear:animated];  
    self.barAnimatingScrollViewDelegate.animatableHeaderBar = (UIView <QWAnimatableBar>  
*)self.navigationController.navigationBar;  
}  
  
- (void)fetchData  
{  
    [self showLoadingIndicator];  
    [self.productsDataStore fetchDataWithRequestURL:nil errorHandler:^(NSError *error) {  
        [self.dataStore:self.productsDataStore dismissiveError:error];  
    }];  
}  
  
- (void)showLoadingIndicator  
{  
    self.collectionView.hidden = YES;  
    self.loadingIndicatorView.hidden = NO;  
}
```

```
- (void)setupDefaults  
{  
    self.collectionViewDataSource.cellFactory =  
    [QWCollectionViewProductCellFactory class];  
    self.collectionViewDelegate.cellFactory =  
    [QWCollectionViewProductCellFactory class];  
    self.collectionViewLayoutArrangement =  
    [QWProductsViewController layoutArrangementForUserInterfaceIdiom:  
    [UIDevice.currentDevice].userInterfaceIdiom];  
    self.collectionViewLayoutDelegate.layoutSpacingClass =  
    [QWCollectionViewAutoSpacingLayoutDelegate  
    sizeClassForUserInterfaceIdiom:[UIDevice  
    currentDevice].userInterfaceIdiom];  
    self.collectionViewLayoutDelegate.cellSizeClass =  
    [QWProductsViewController  
    sizeClassForLayoutArrangement:self.collectionViewLayoutArrangement]  
};
```





Be **ready** to ship



Feature **flags**



Feature flags

BILGlobalToolbar.m

```
#ifndef SEARCH_INPUT
    self.searchInputView.hidden = YES;
#endif
```

User.xcconfig

```
SEARCH_FEATURES = SEARCH_TOOLBAR SEARCH_INPUT
USER_TOGGLES = $(SEARCH_FEATURES)
```

Feature flags

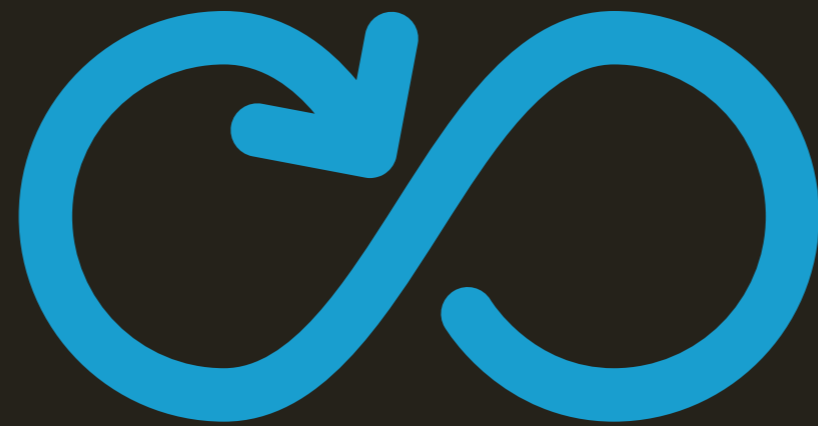
Debug.xcconfig

```
#include "Shared.xcconfig"  
#include "Features.xcconfig"  
#include "User.xcconfig"
```

```
DEBUG_TOGGLES = DEBUG=1
```

```
GCC_PREPROCESSOR_DEFINITIONS = $(inherited) $(DEBUG_TOGGLES)  
                                $(FEATURE_TOGGLES)  
                                $(USER_TOGGLES)
```

Iterative development



Development **pipeline**



Continuous Integration



Continuous Delivery



Continuous Deployment



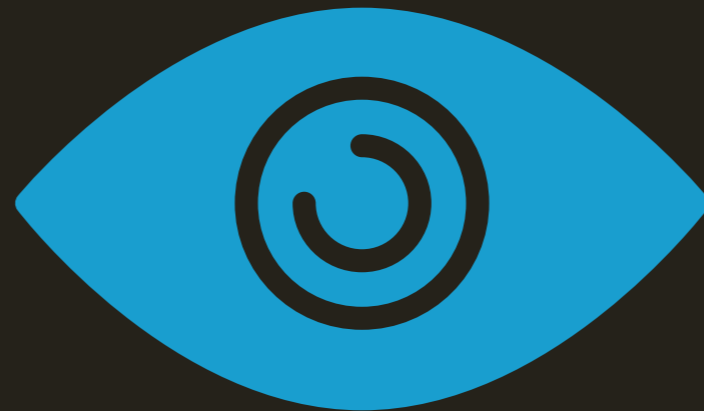
Robust processes



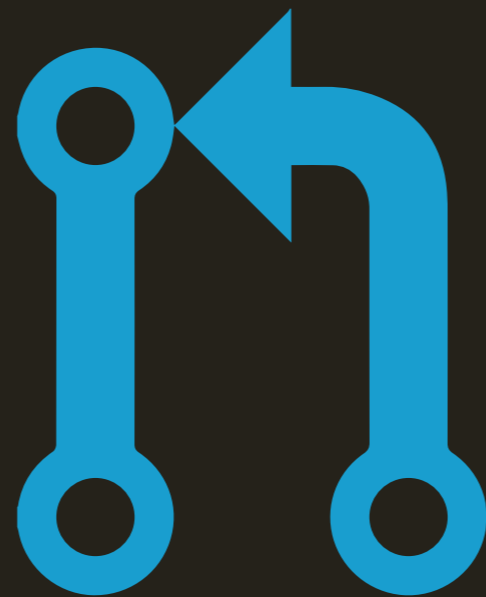
Speedy **collaboration**



Code **previews**



Pull requests



Social code reviews



Don't **stop** the world



Maintain **quality**



Manage Tech Debt



Start thinking for the
business

Product aligned teams

- Business
- Designers
- App Developers
- Backend Developers
- QAs

Start doing what you
can do today

Walled Garden



Remove **Business** Logic



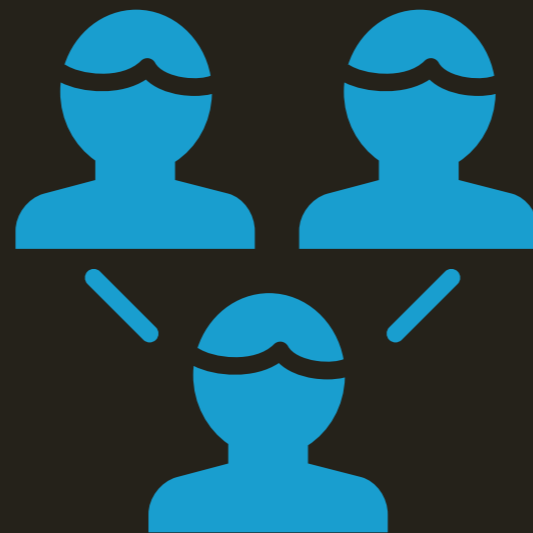
Be **ready** to ship



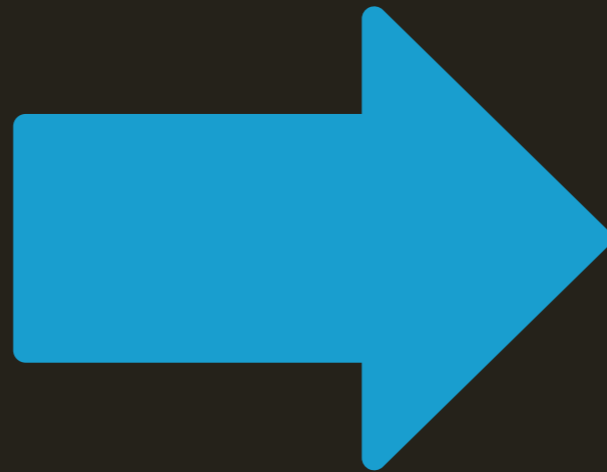
Enable collaboration



Product aligned teams



Keep moving **forward**



Keep **building**



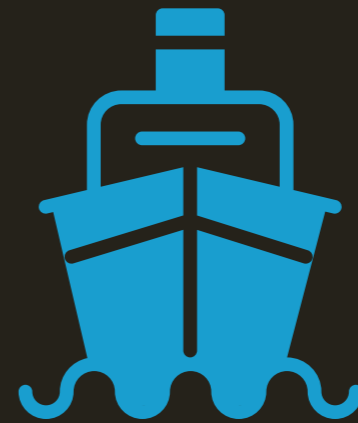
Keep **measuring**



Keep **learning**



Keep **shipping**



Thanks



Cameron **Barrie**

 Client Principal at **Bilue**

 cameron@bilue.com.au

 [@whalec](https://twitter.com/whalec)

Cameron **Barrie**

 Client Principal at **Bilue**

 cameron@bilue.com.au

 @whalec