



Components


Ari Grant


Our Journey


- Layout of a feed story
- Code for a feed story's header
- Components
- Building blocks
- Data pipeline
- Optimizations

NEWS FEED




News Feed




Tom McTommer 

Just now · 

It's cold.

 Like |  Comment |  Share




News Feed



Tom McTommer 🐻 feeling cold ⌵

1 min · 🌐

It's cold.

 Like |  Comment |  Share


News Feed






Tom McTommer 🧑‍🦲 feeling cold at **North Pole** ⌵

Just now · Rotterdam, Netherlands · 🌐

It's cold.



North Pole
★★★★★
Landmark 🔖

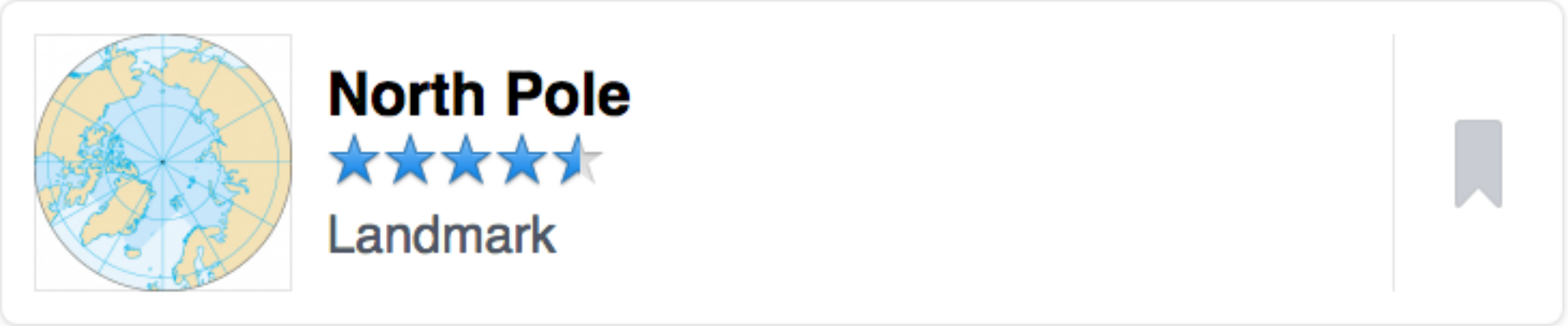
 Like |  Comment |  Share



News Feed

Ira Tnarg was tagged in a post. ▼

 **Tom McTommer** 🥶 feeling cold with **Ira Tnarg** at **North Pole**
Just now · Rotterdam, Netherlands · 🌐

It's cold.

 **North Pole**
★★★★★
Landmark 🔖

 Like |  Comment |  Share

News Feed



Tom McTommer 🥶 feeling cold with **Ira Tnarg** at **North Pole** ▼

Just now · Rotterdam, Netherlands · 🌐

It's cold.



North Pole
★★★★★
Landmark



1 Comment

 Like |  Comment |  Share

News Feed



Ira Tnarg shared **Tom McTommer's status update.** ▼

Just now · 

This post gave me chills.



Tom McTommer 🥶 feeling cold with **Ira Tnarg** at **North Pole**

6 mins · Rotterdam, Netherlands · 

It's cold.

 Like |  Comment |  Share

News Feed



Tom McTommer

September 10 at 6:08am · 🌐

What a silly bear... — 😜 feeling silly.



👍 Like

💬 Comment

➦ Share

News Feed

 **Tom McTommer** changed his profile picture. 
September 10 at 6:09am · 



 Like |  Comment |  Share

News Feed

2014



 Learned to speak French with a Polar Bear

[Show Top Posts](#)

News Feed



Tom McTommer

July 31 · San Francisco, CA · 🌐



Learned to speak French with a Polar Bear

July 31 with **Maria Greene** at **San Francisco Zoo**

Favorite word — polar bear. Method — at the zoo.

This is the story!



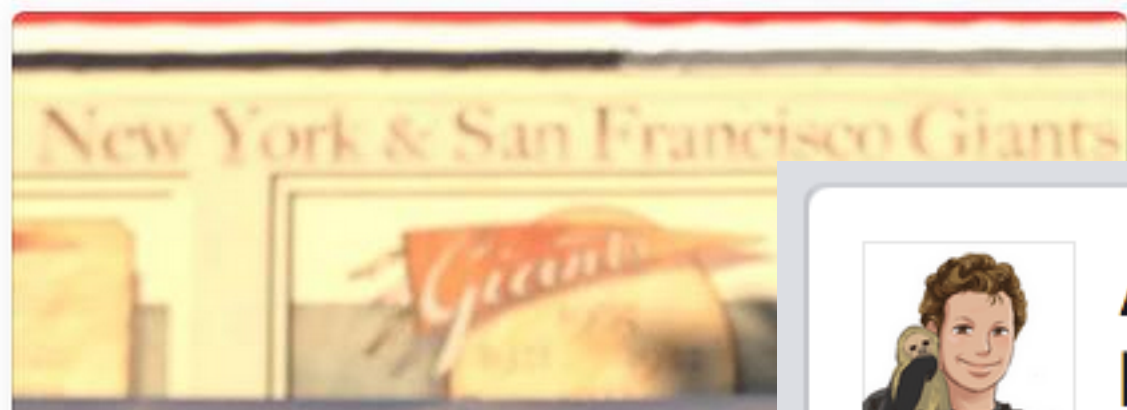
News Feed

Ayme Alvarez is now friends with Bonnie Roy and 2 other people.



Ayme Alvarez and Bonnie Roy are now friends.

December 12, 2012



Bonnie Roy

UC San Francisco (...)
4 mutual friends



Like



Ayme Alvarez and T...
Havelick are now frie...

December 10, 2012



Ari Grant li...
by Jennifer

13 mins ·

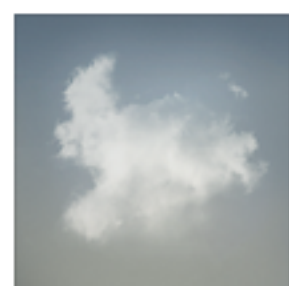
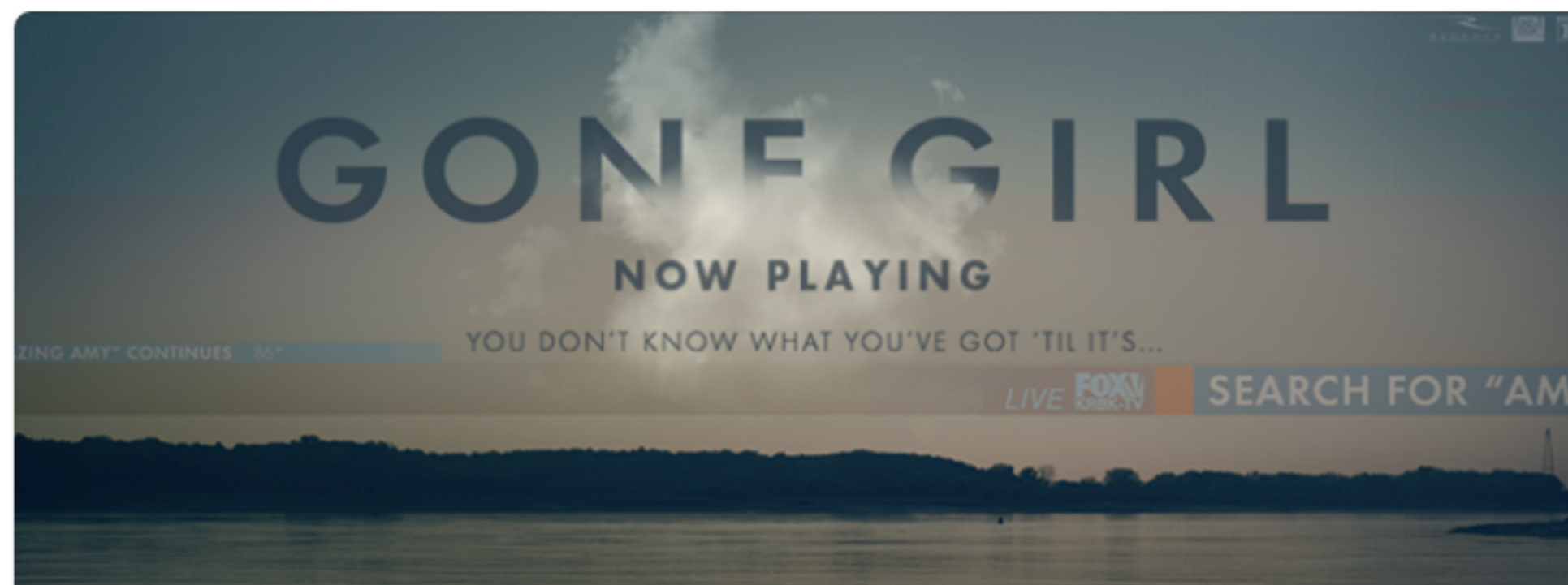


Ari Grant

October 4 at 11:26pm · San Francisco, CA ·

Twisted, intense, and stressful.

It was highly engaging and unraveled in ways I would have never expected. I left the theater feeling uneasy yet somehow ready for more. — 🎬 watching [Gone Girl](#) at Westfield San Francisco Centre.



Gone Girl ✓

Movie

★★★★★ · 312,114 likes

22 Likes · 4 Comments



Like

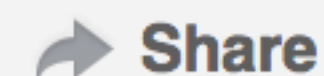


Share

San Francisco, CA ·

the continental for a bit. —
ark from San Francisco

International Airpo...



Share



Bewakoofiyaan

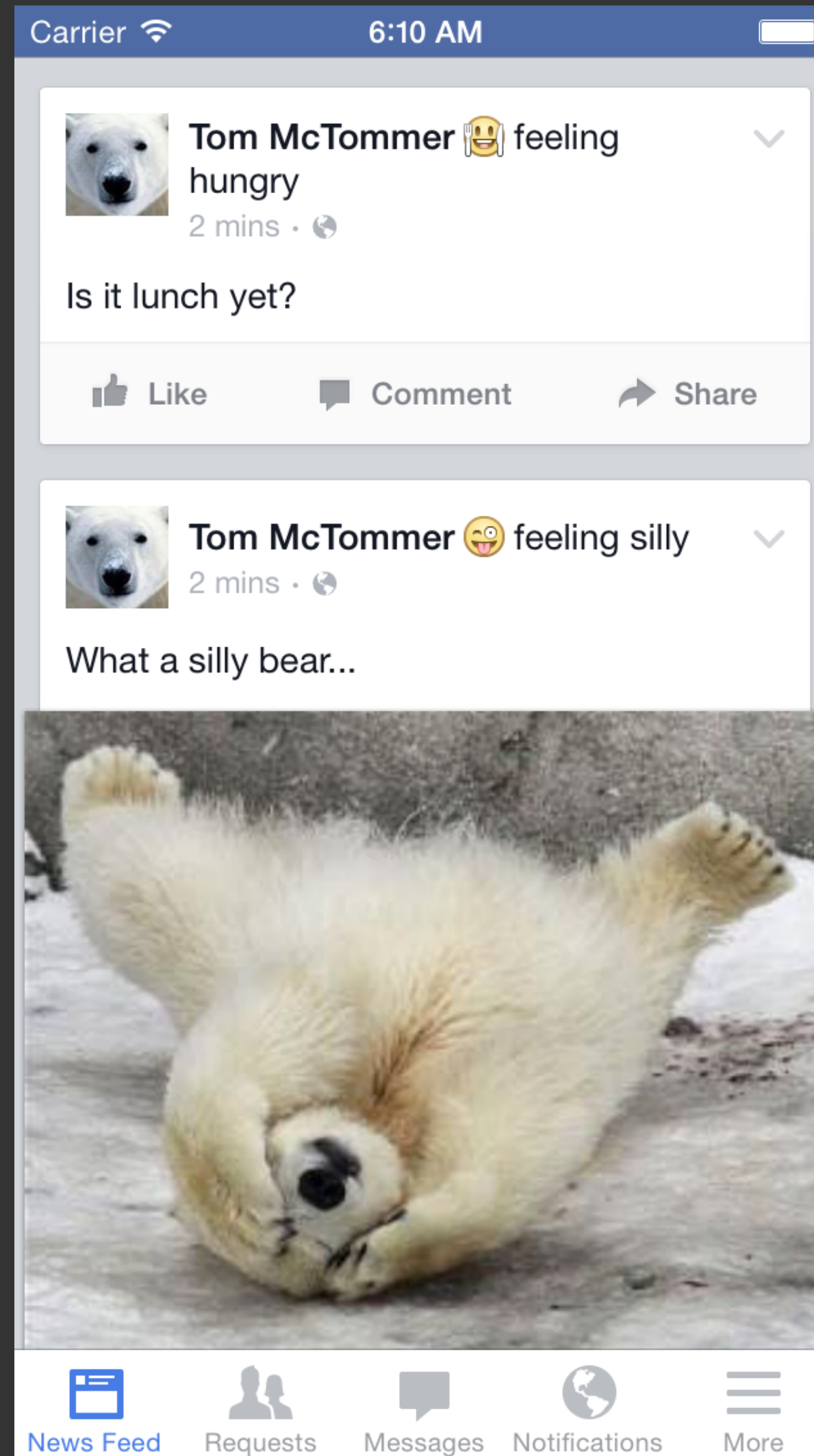
Movie

928,585 likes

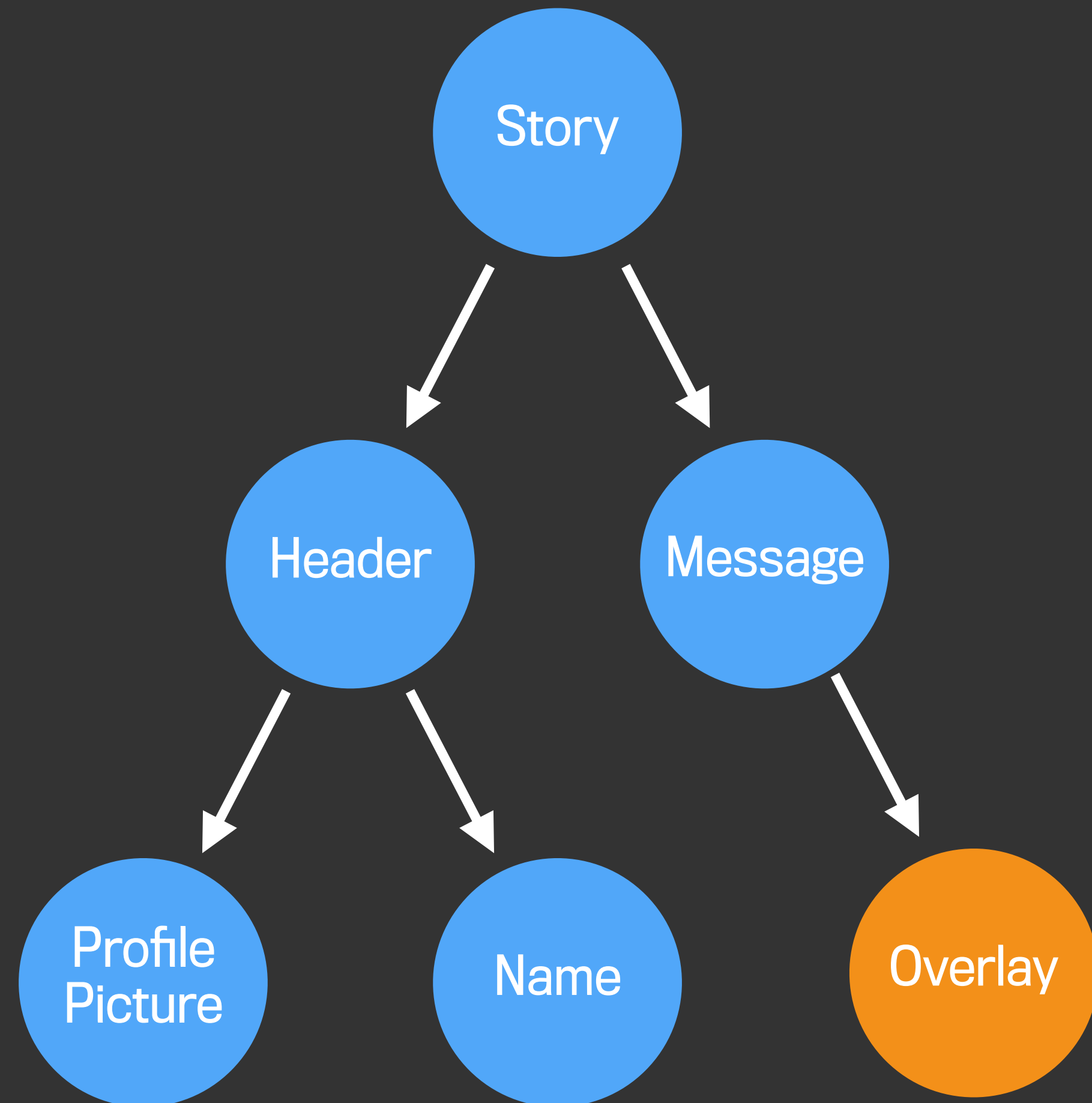


News Feed

A list of stories



iOS' View APIs



```
-(CGSize)sizeThatFits:(CGSize)size  
{  
    ... // Math  
}
```

```
-(void)layoutSubviews  
{  
    ... // Math  
}
```

Main (UI) Thread Only!

The Reality of a Complex UI

On iOS

- **60 fps** is desired
- Text layout can take **many ms**
- Configuring **many views is slow**
- Heterogeneous data
- View sizing and **layout can be slow**
- Views can only layout on the UI thread
- Recycling code is manually written
- You have to juggle text- and UI-threads
- Views are mutable and always changing
- Layout code is lots of math
- **17 ms** per frame
- Use a **background thread**
- **Recycle** views
- Don't use "row-based" view-types
- Do sizing and layout **in the background**
- Oh...
- Oh.....
- Oh.....
- Oh.....
- Oh.....

Goals

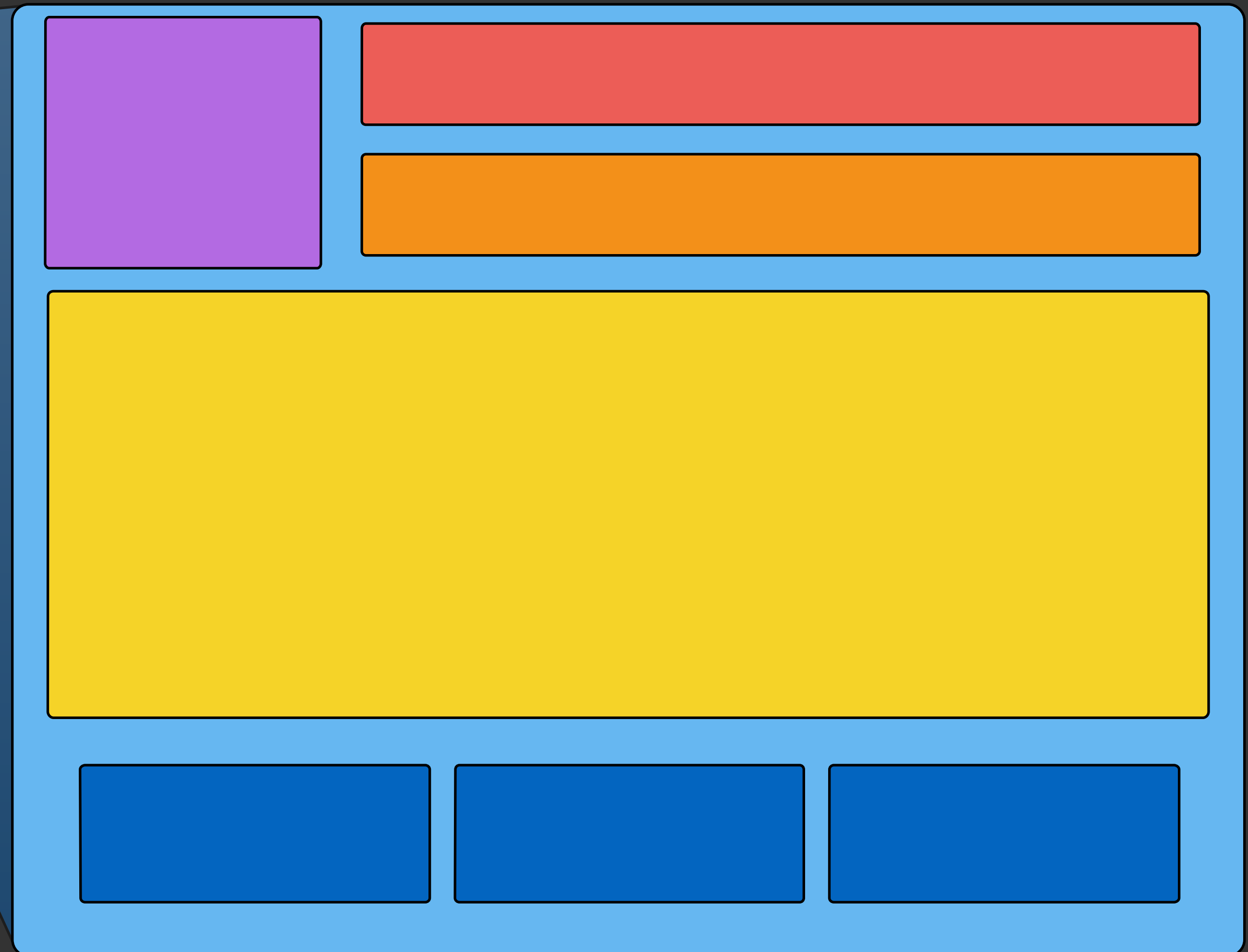
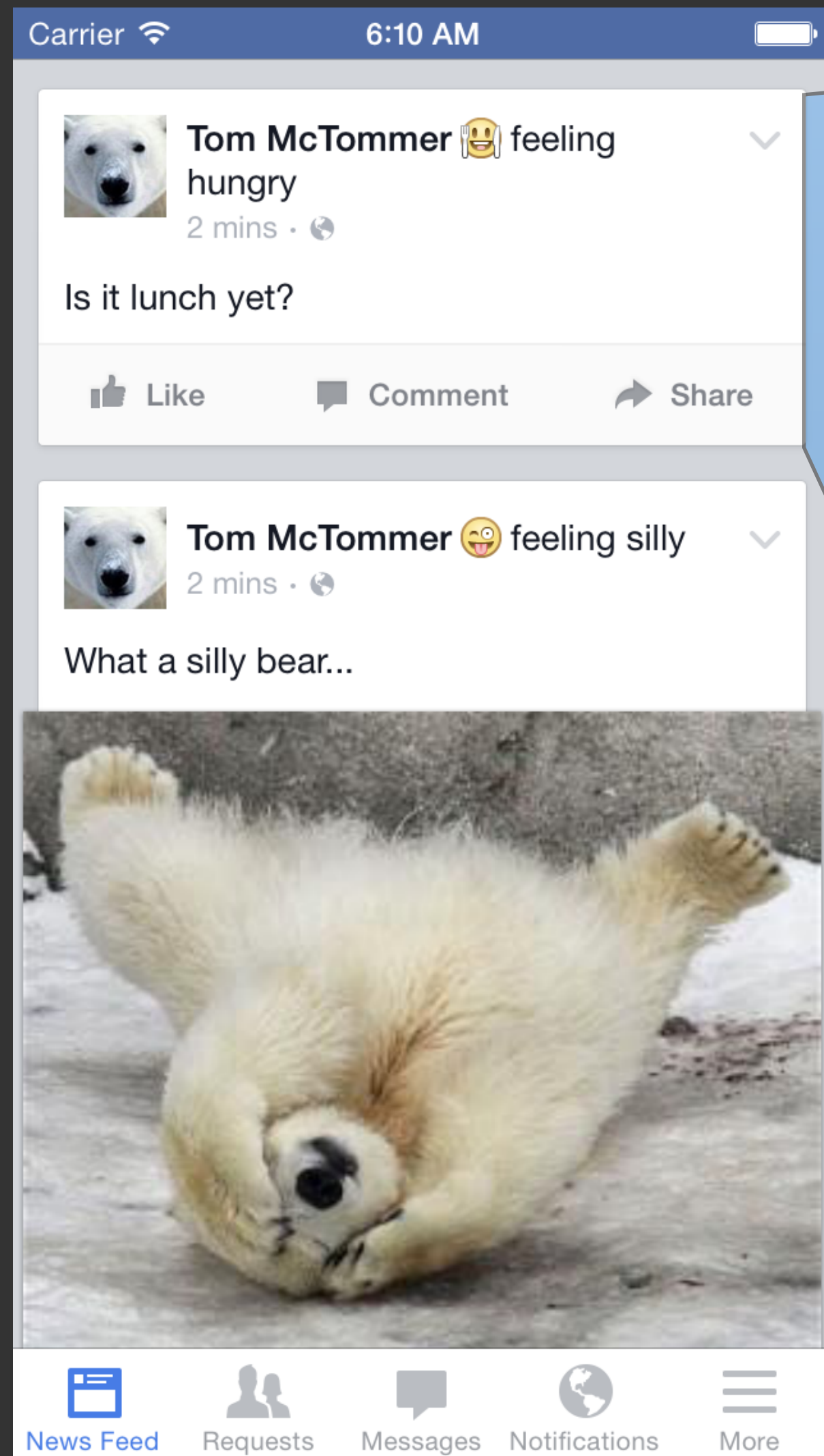
Move complexity to “behind the scenes”

- Easy to layout and recycle views
- One-way data flow when state changes
- Encourage **composition** and enforce **immutability**
- Make the worst errors impossible
- Allow asynchronous computation **invisibly**

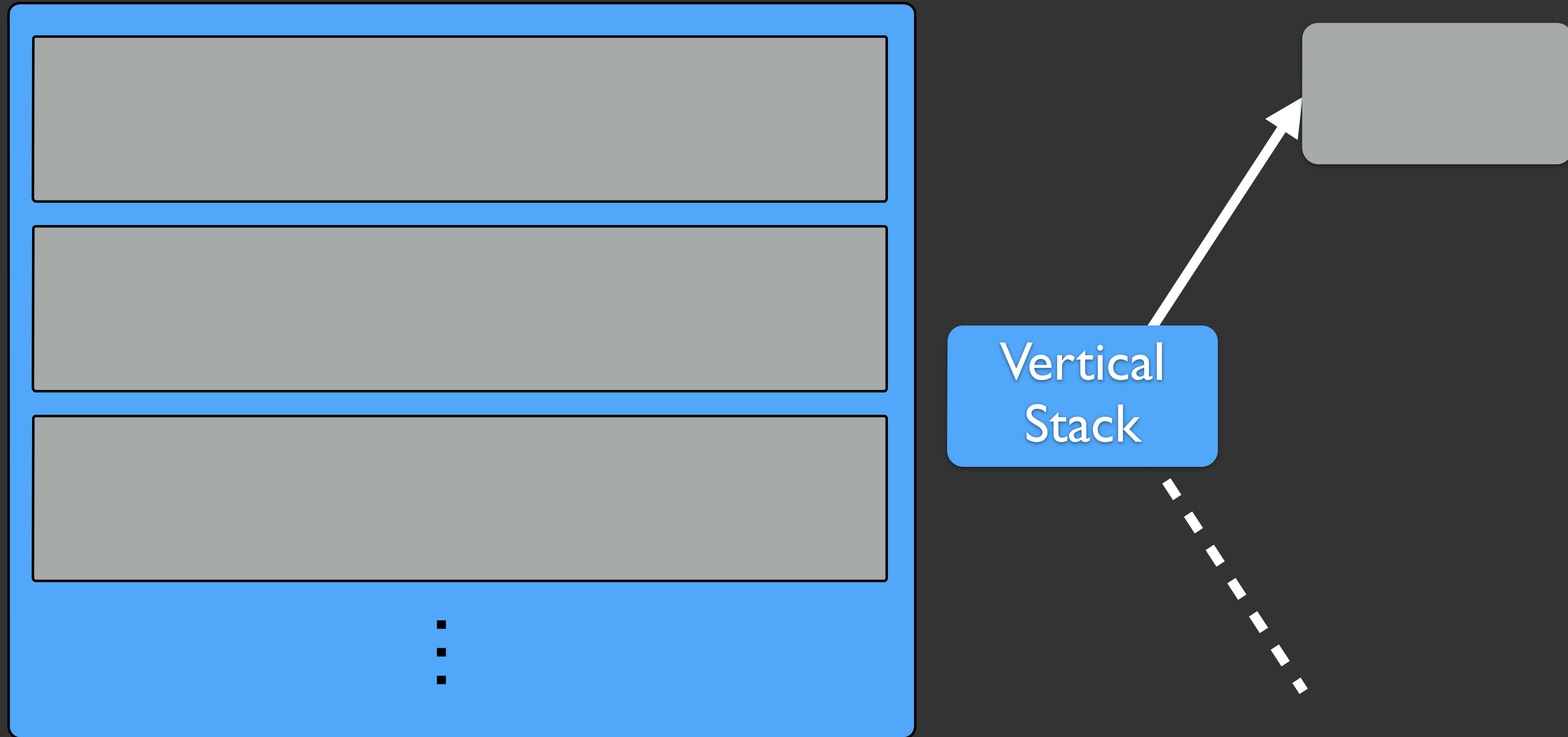


BUILDING A FEED STORY

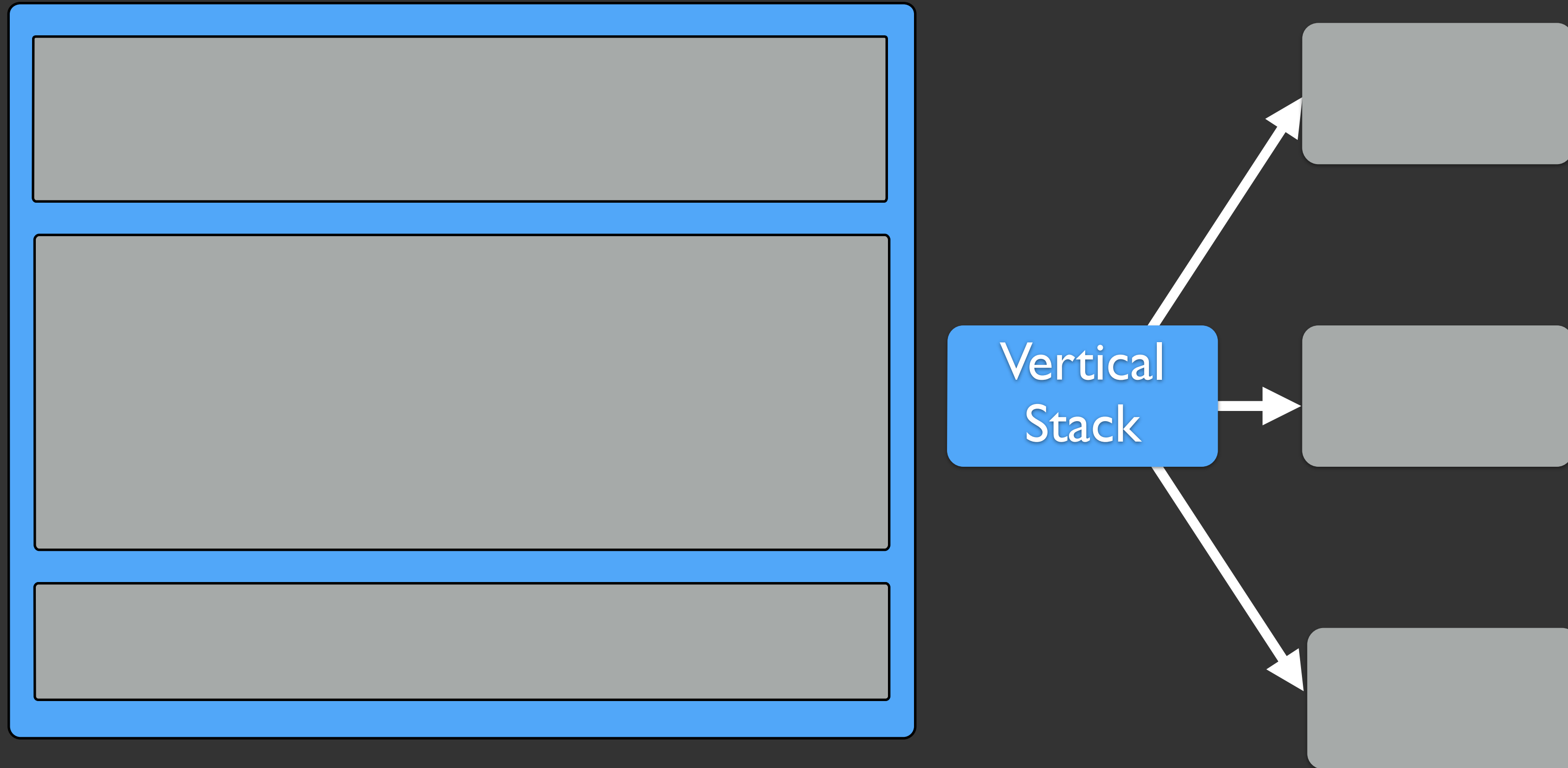
Layout of a Feed Story



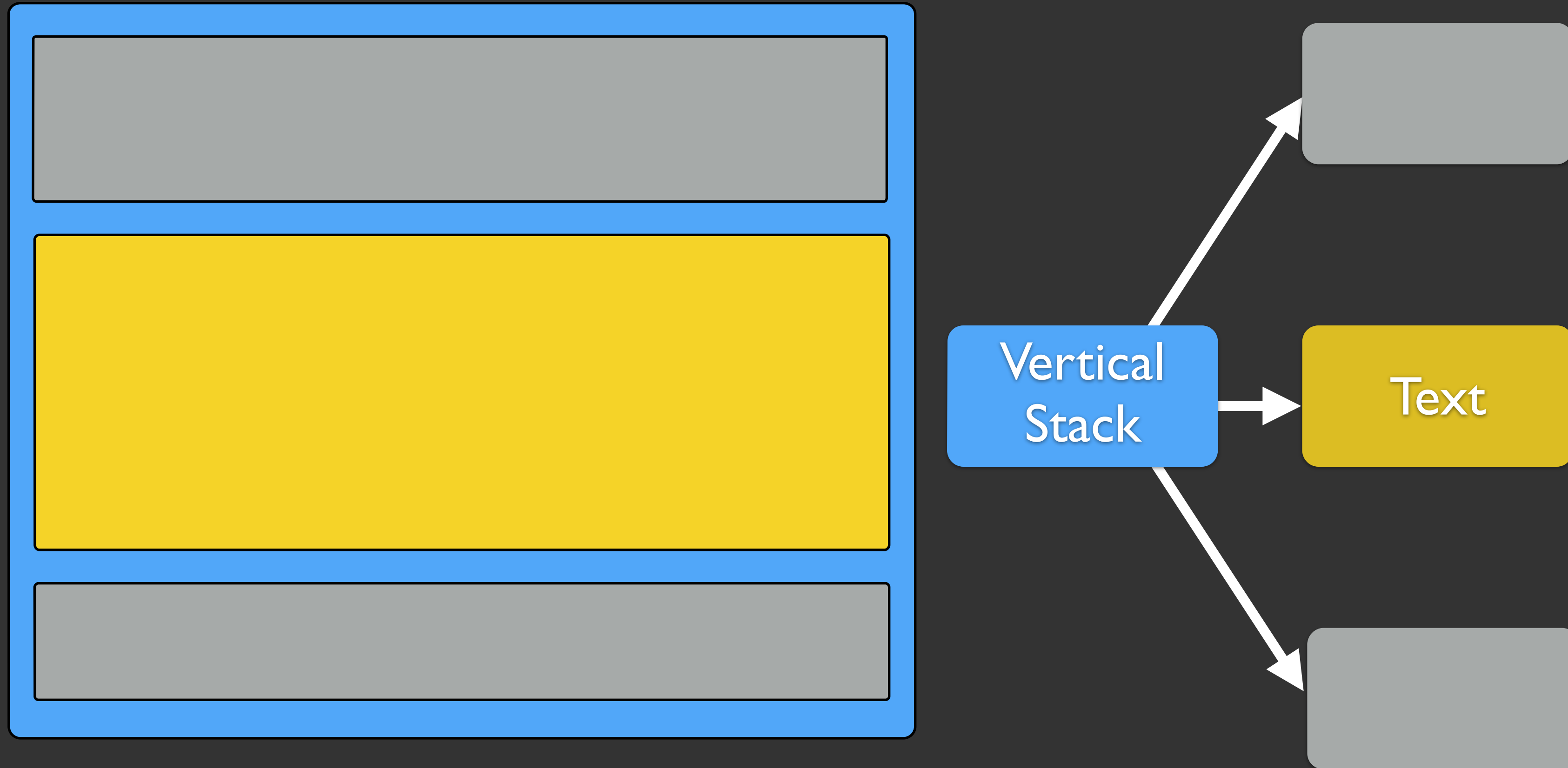
Layout of a Feed Story



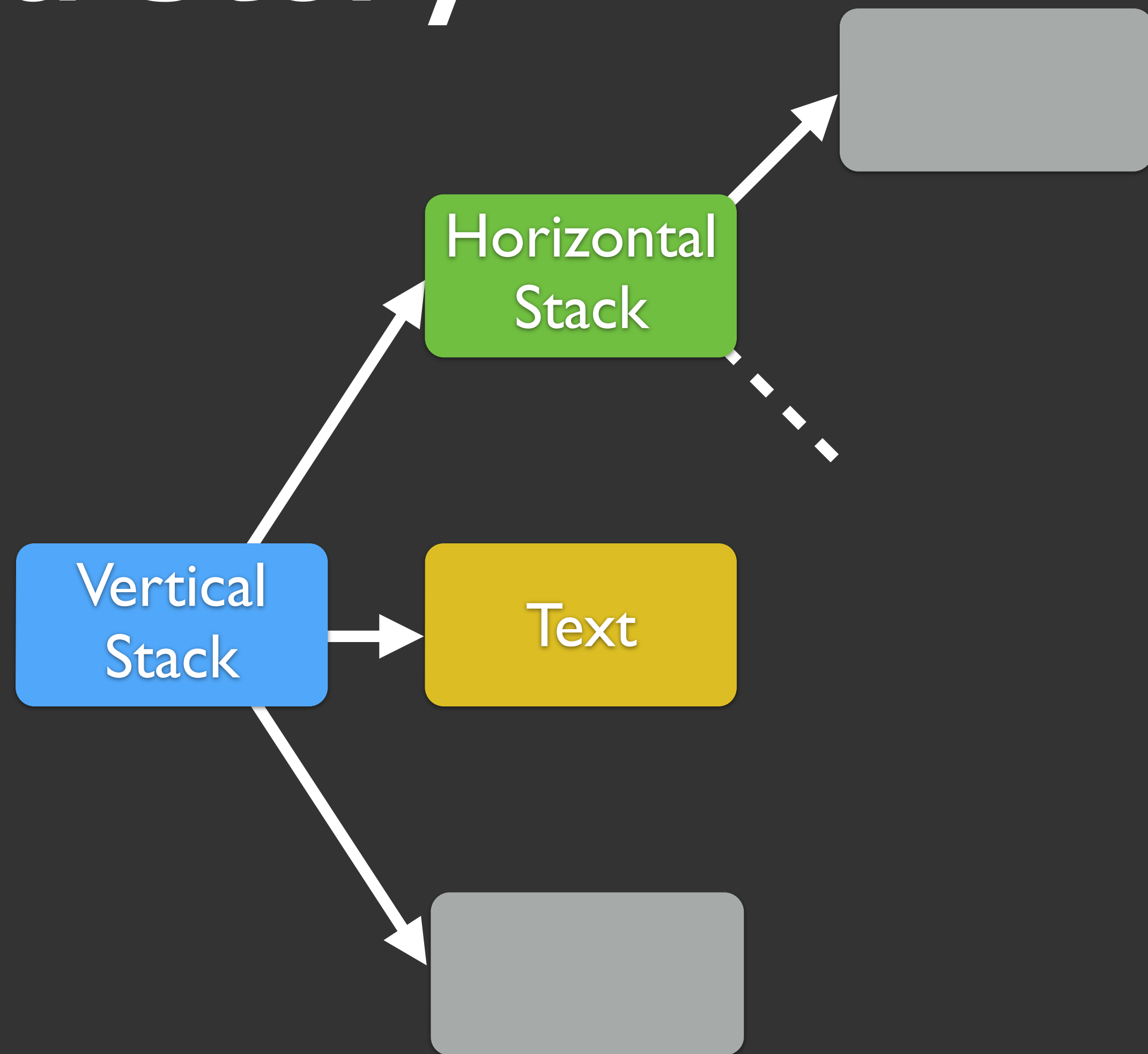
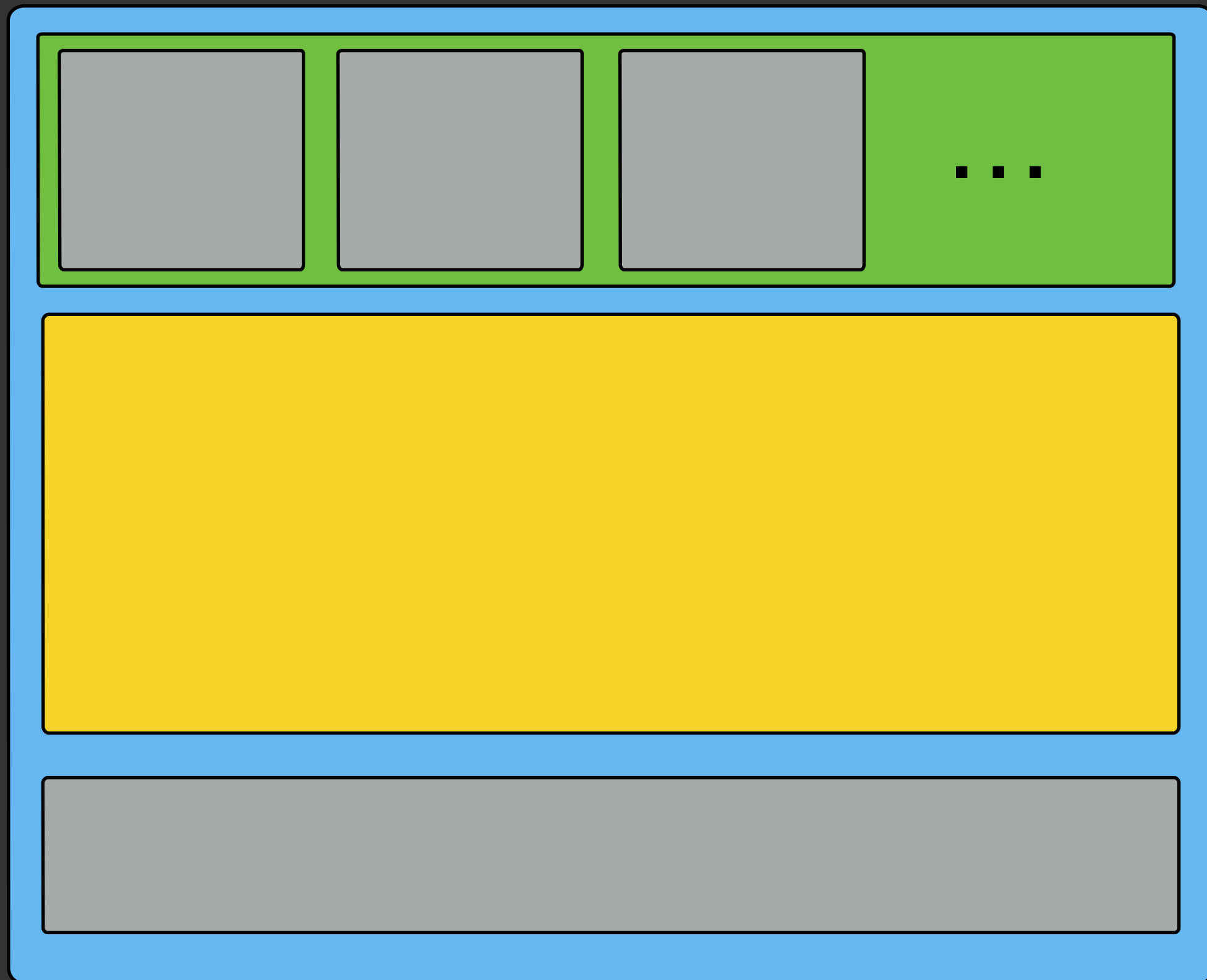
Layout of a Feed Story



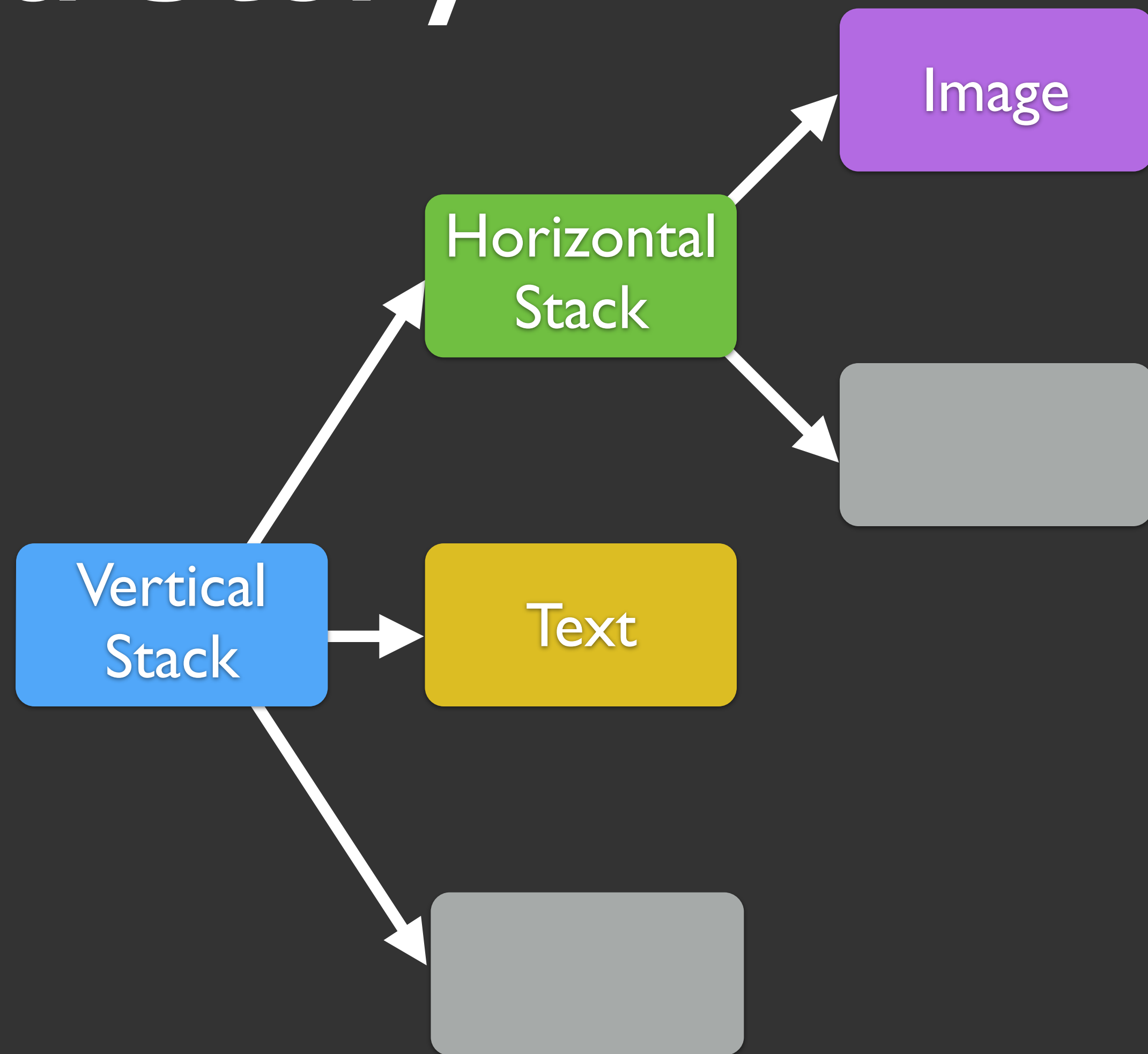
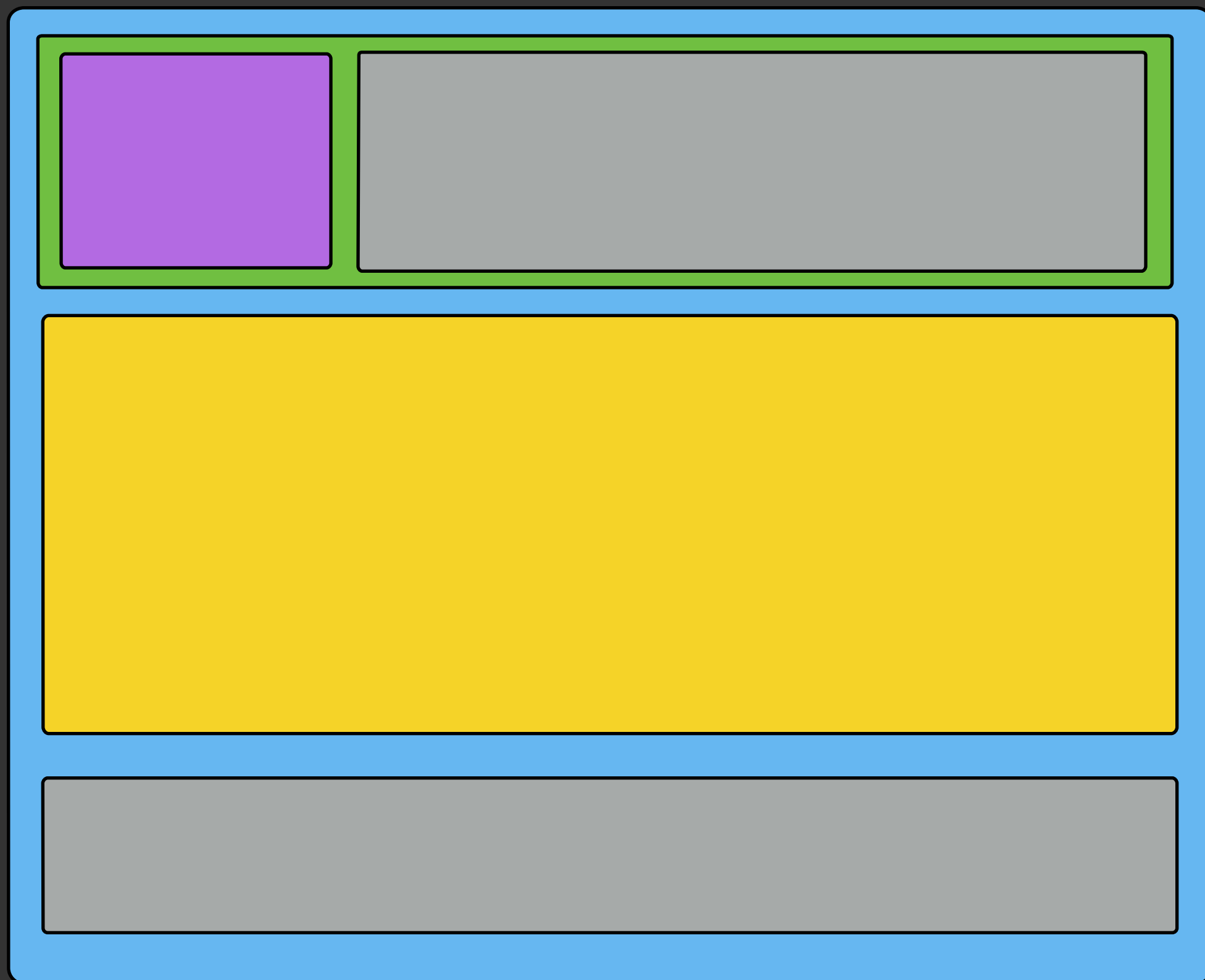
Layout of a Feed Story



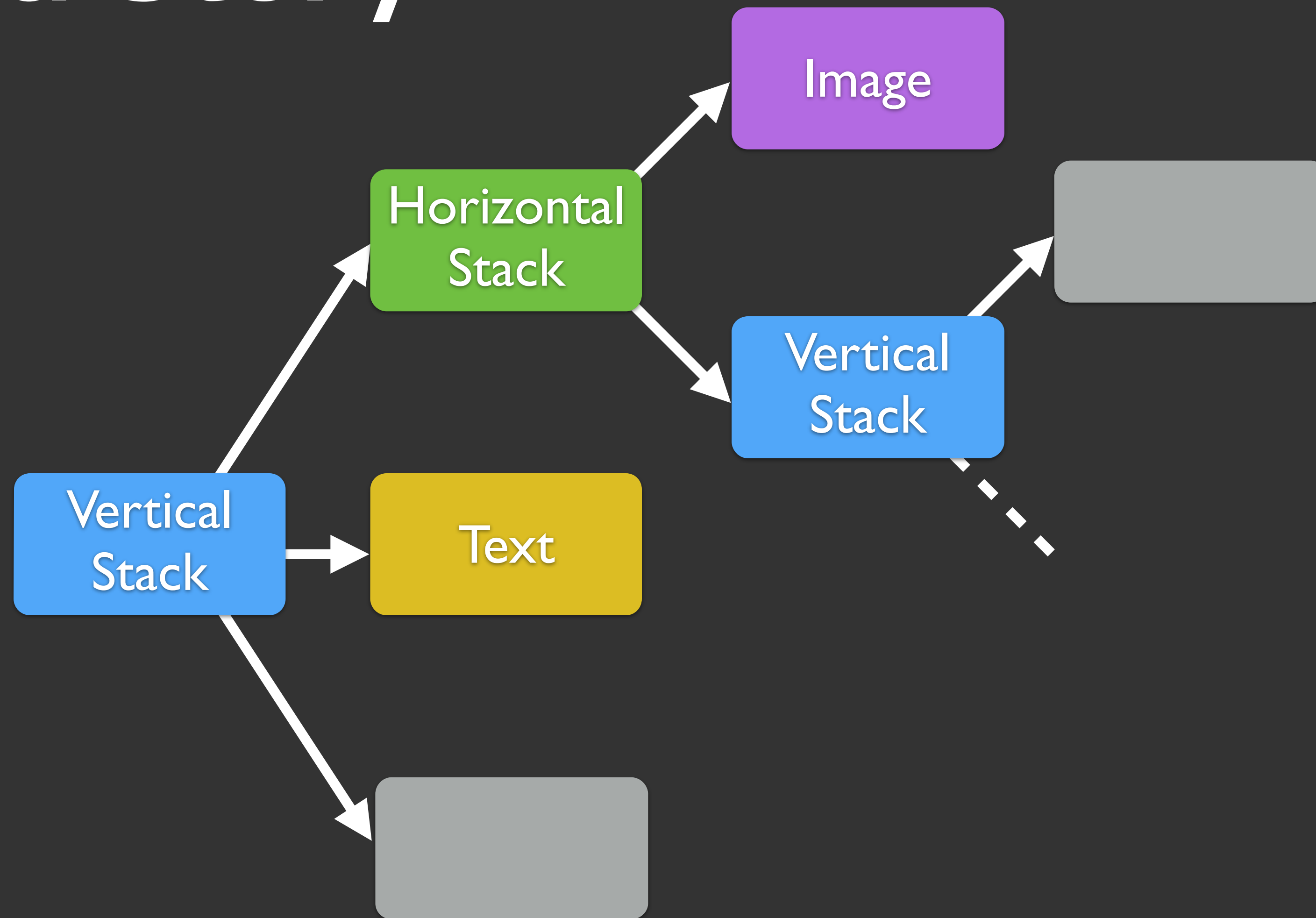
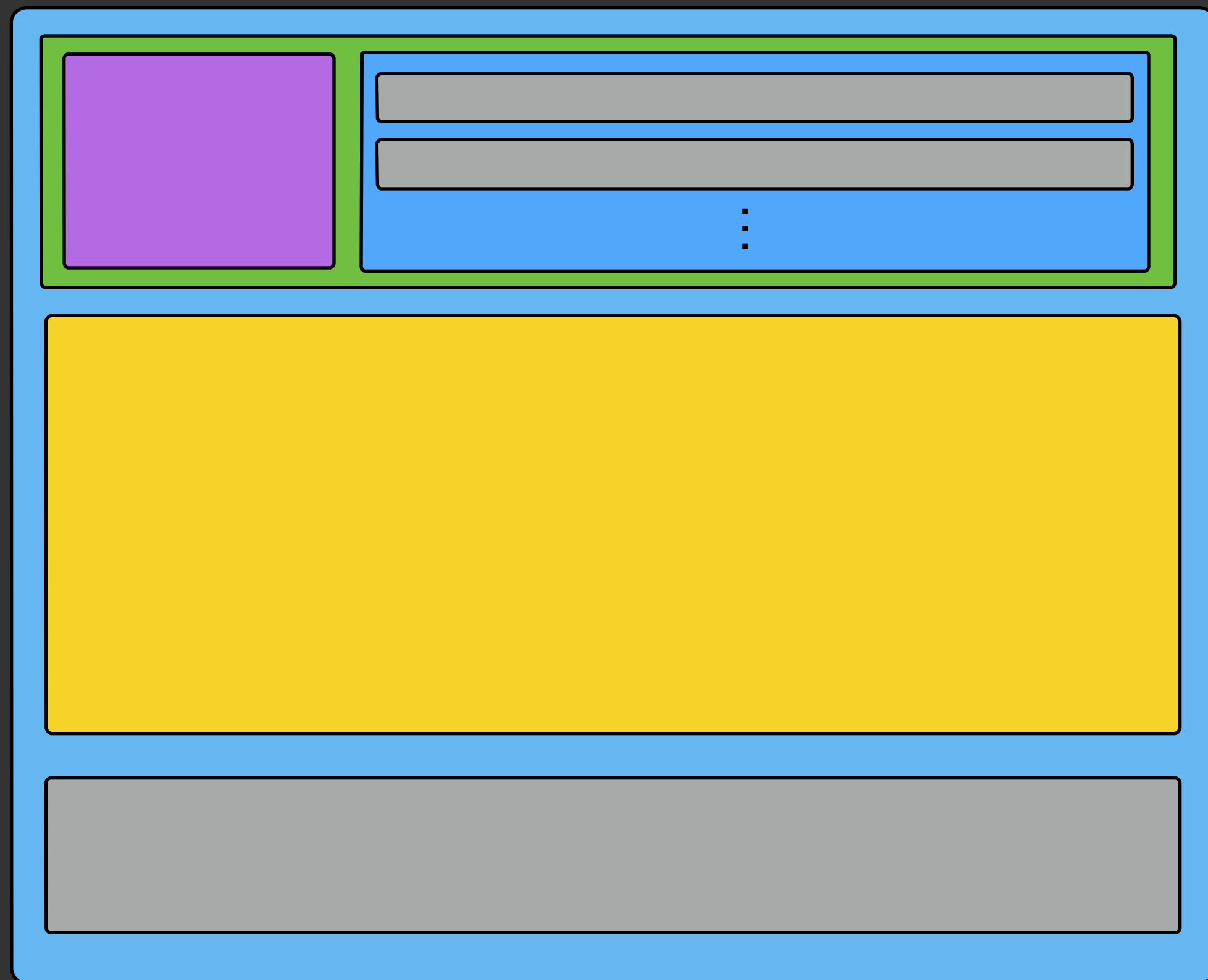
Layout of a Feed Story



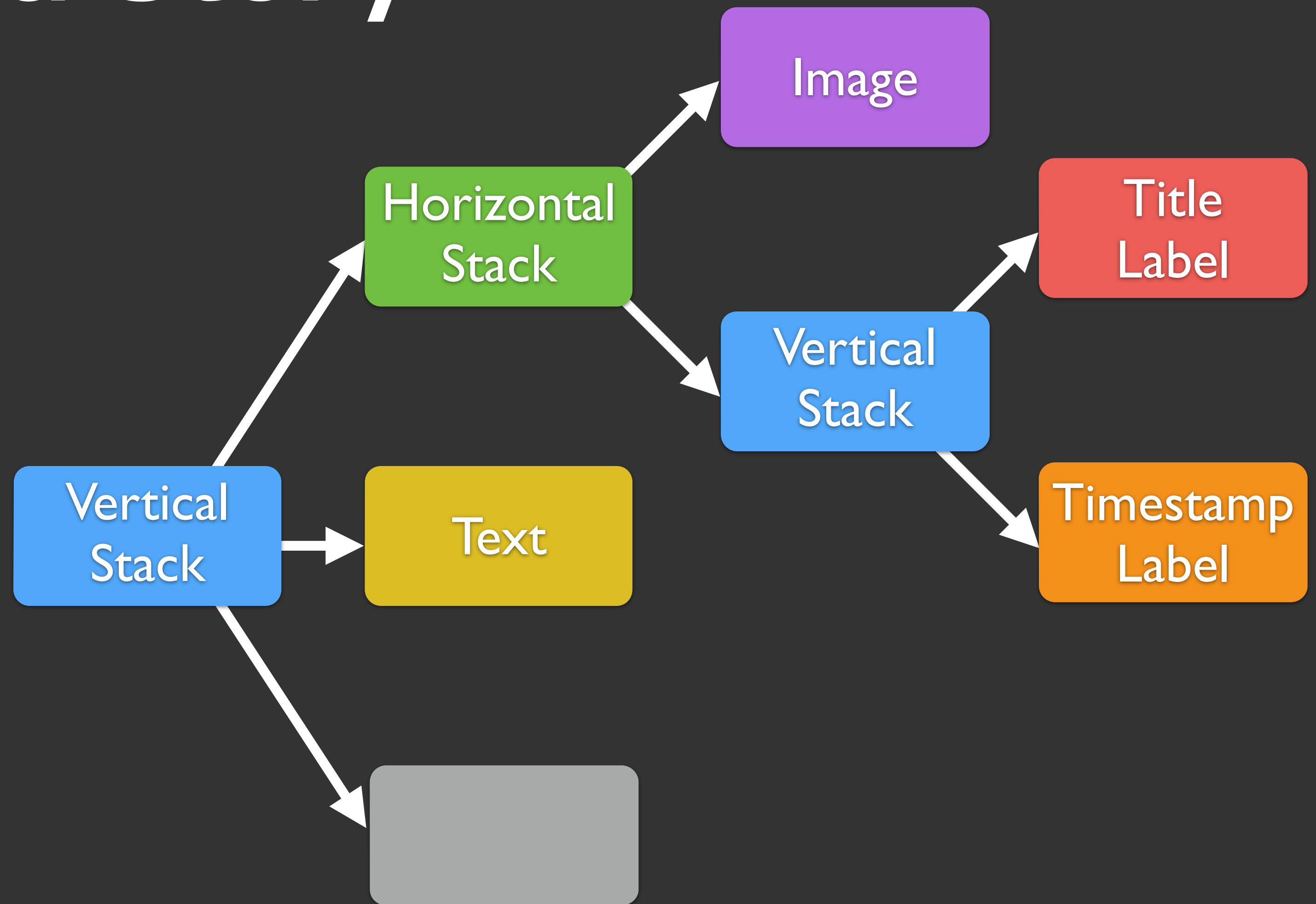
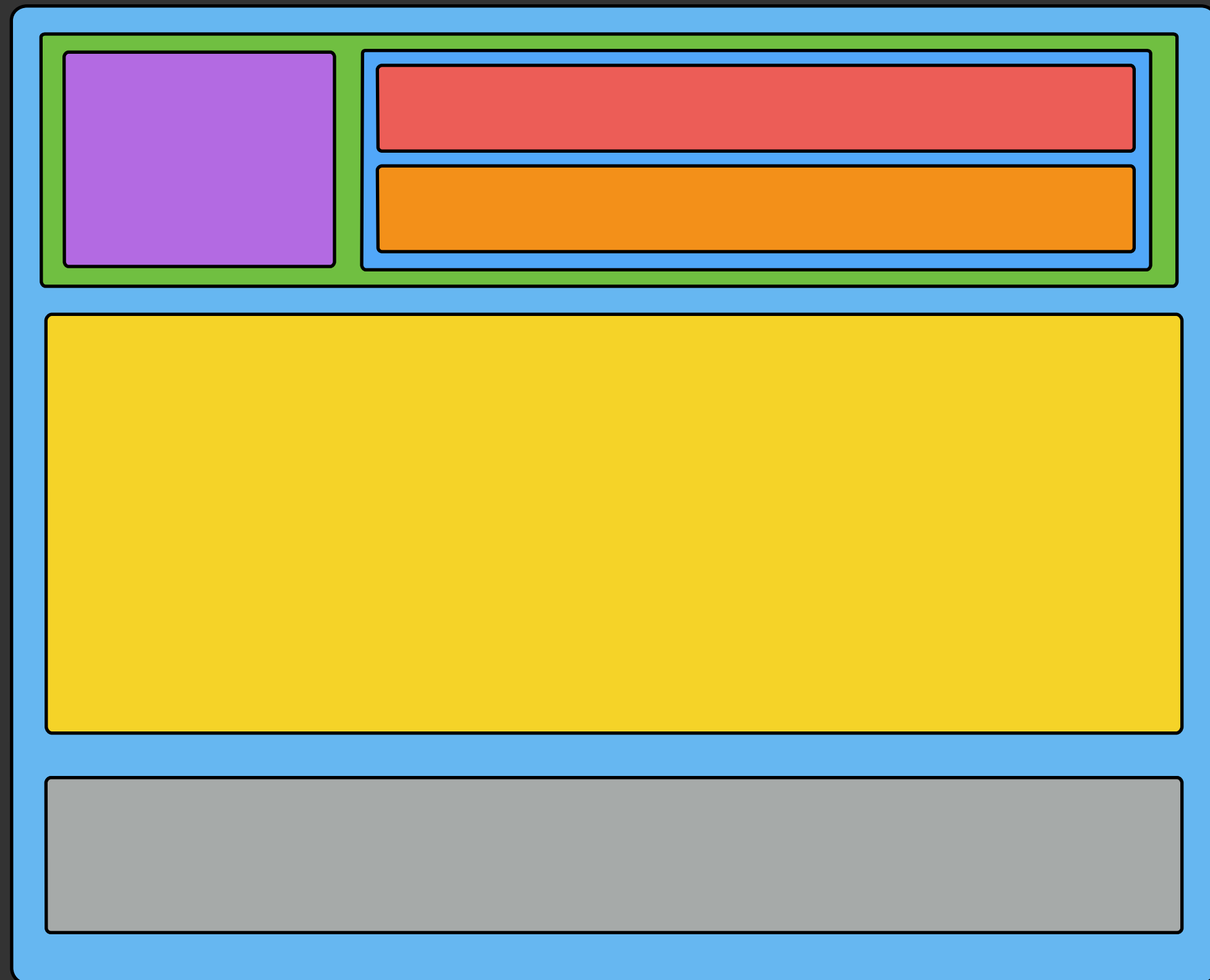
Layout of a Feed Story



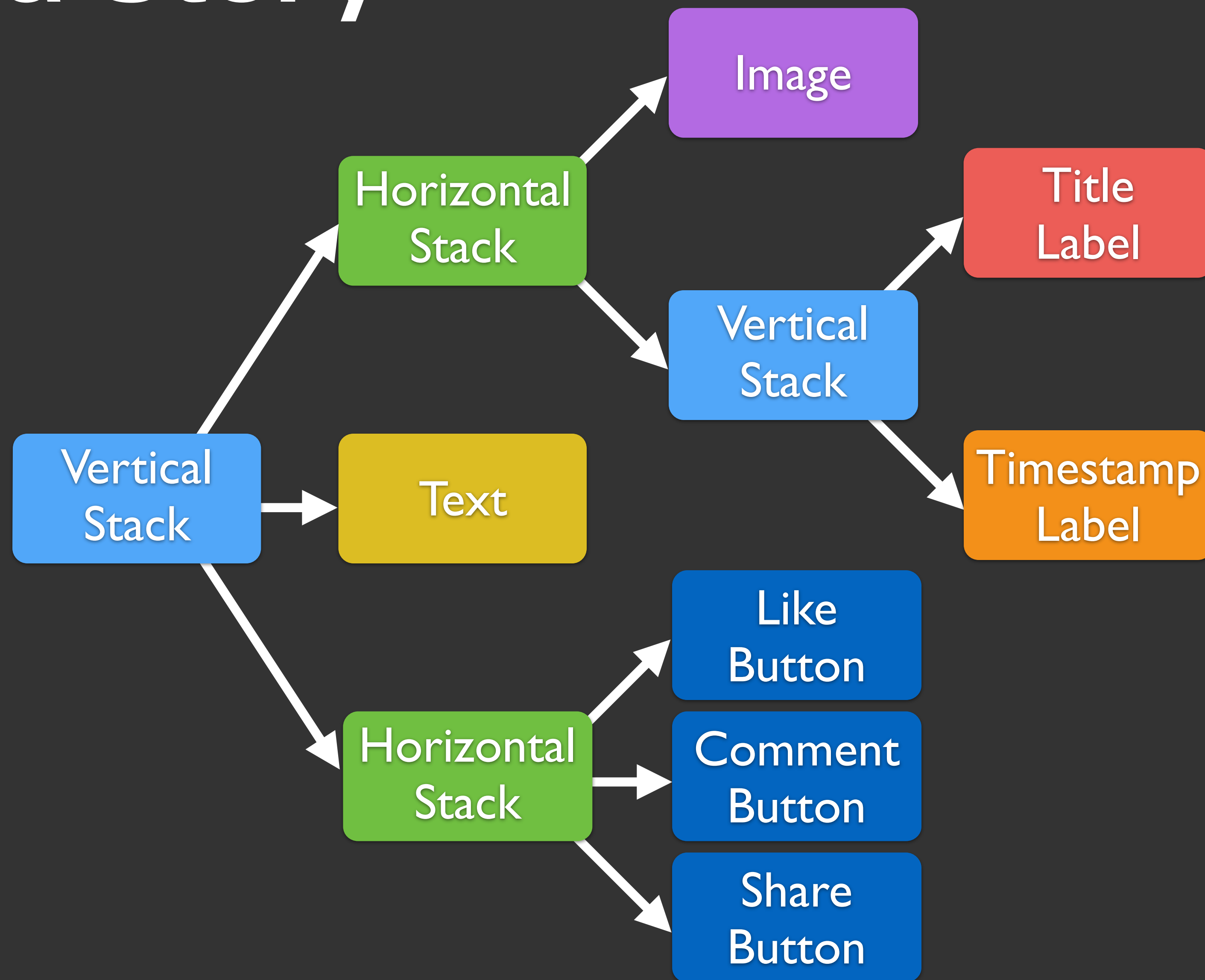
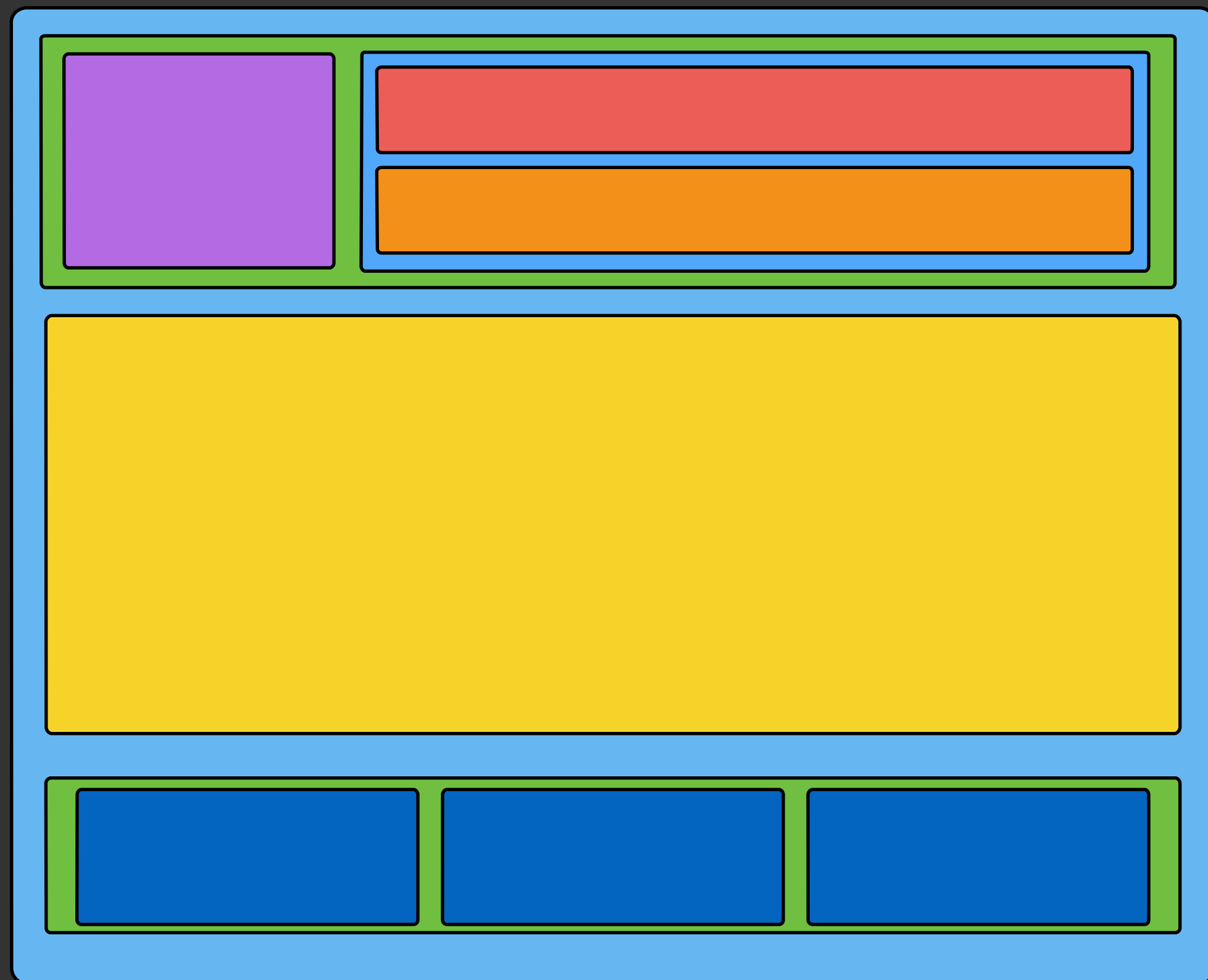
Layout of a Feed Story



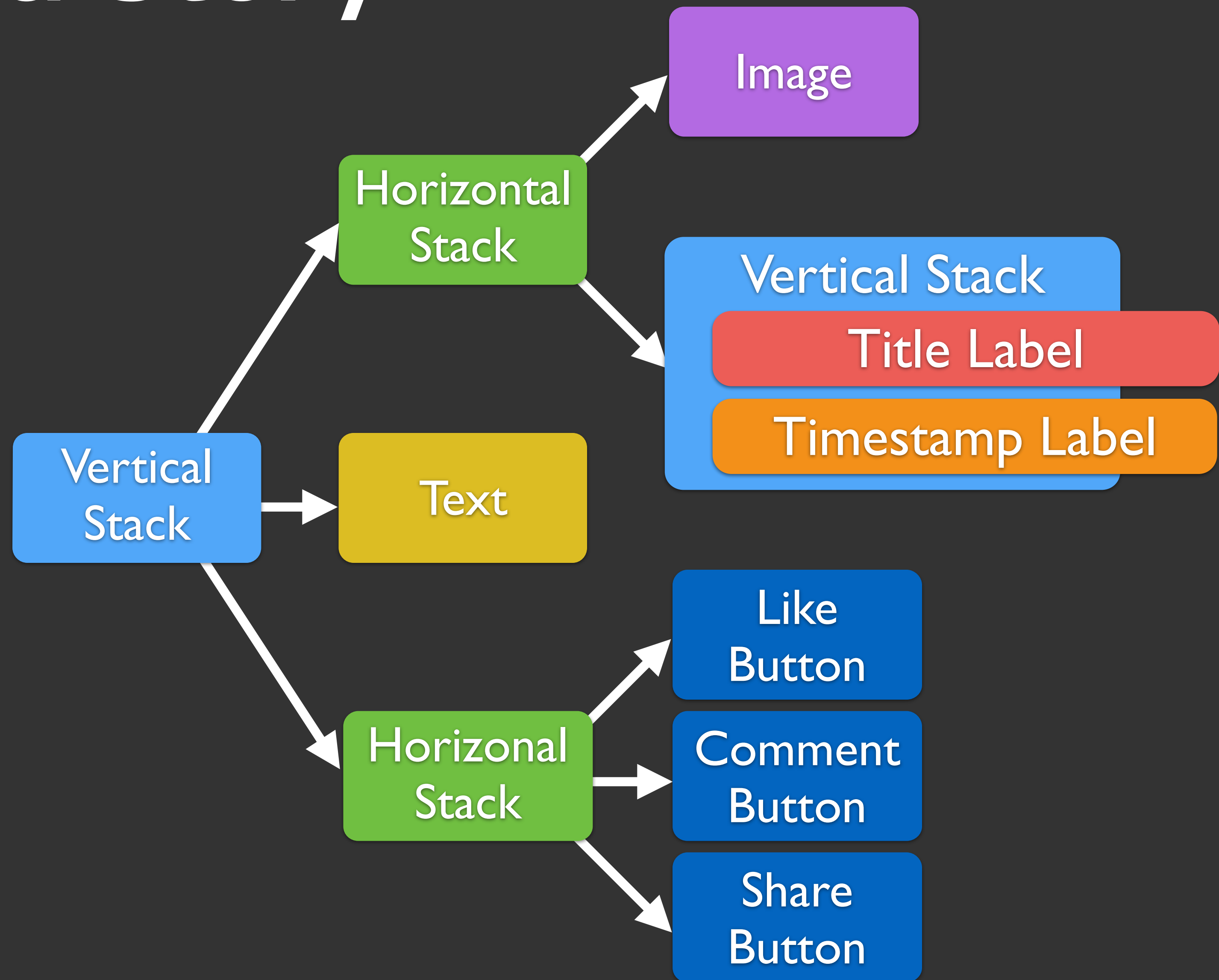
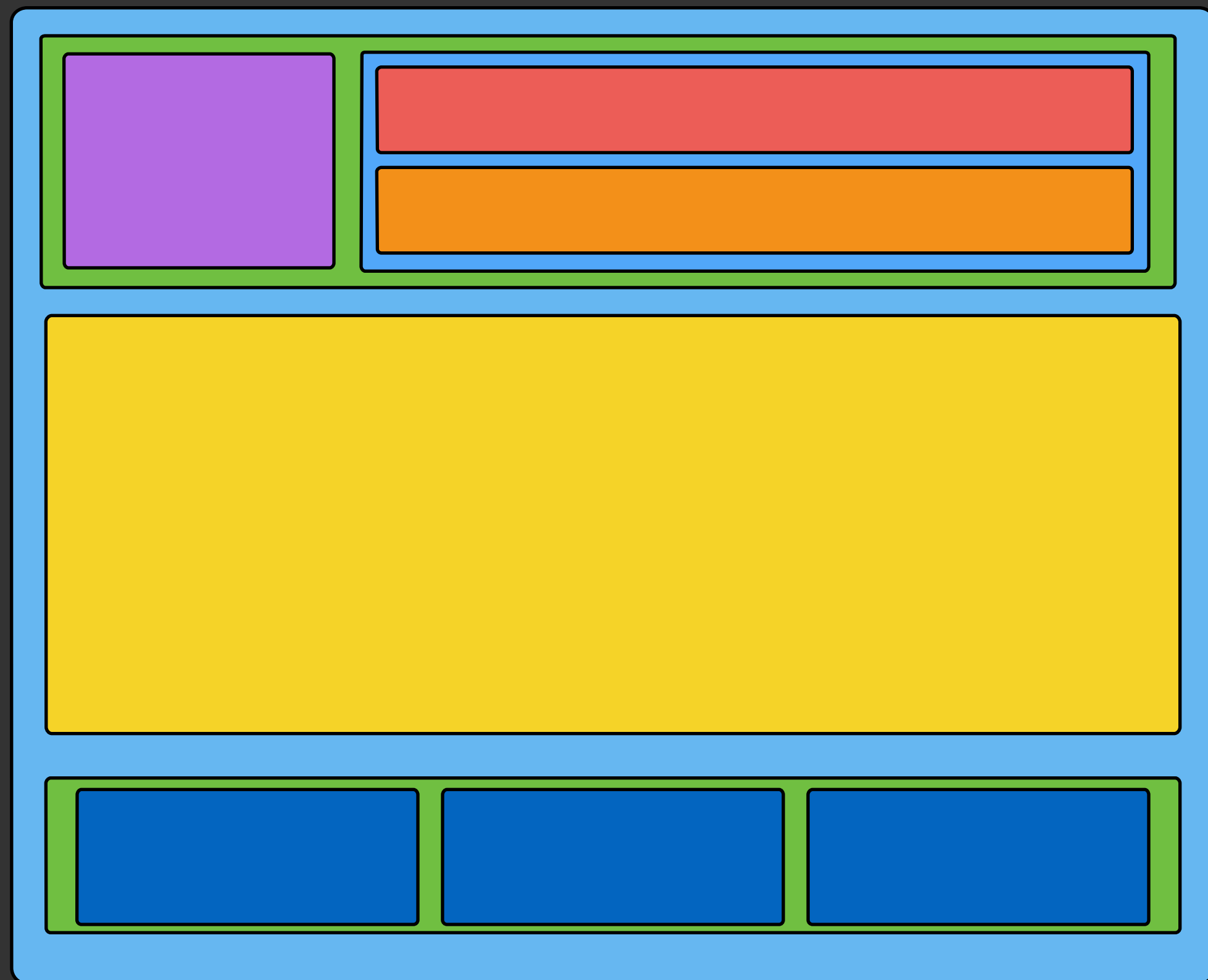
Layout of a Feed Story



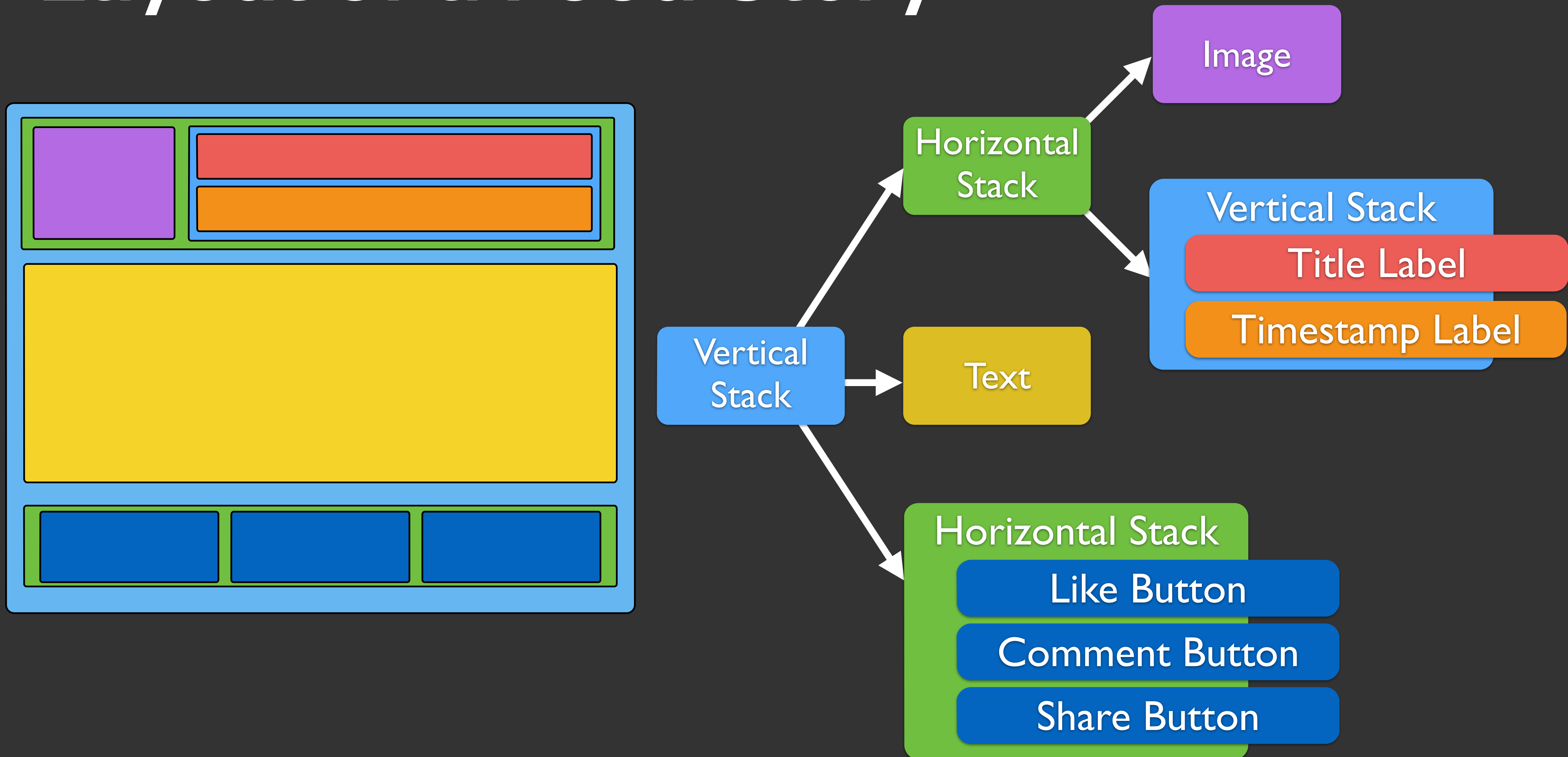
Layout of a Feed Story



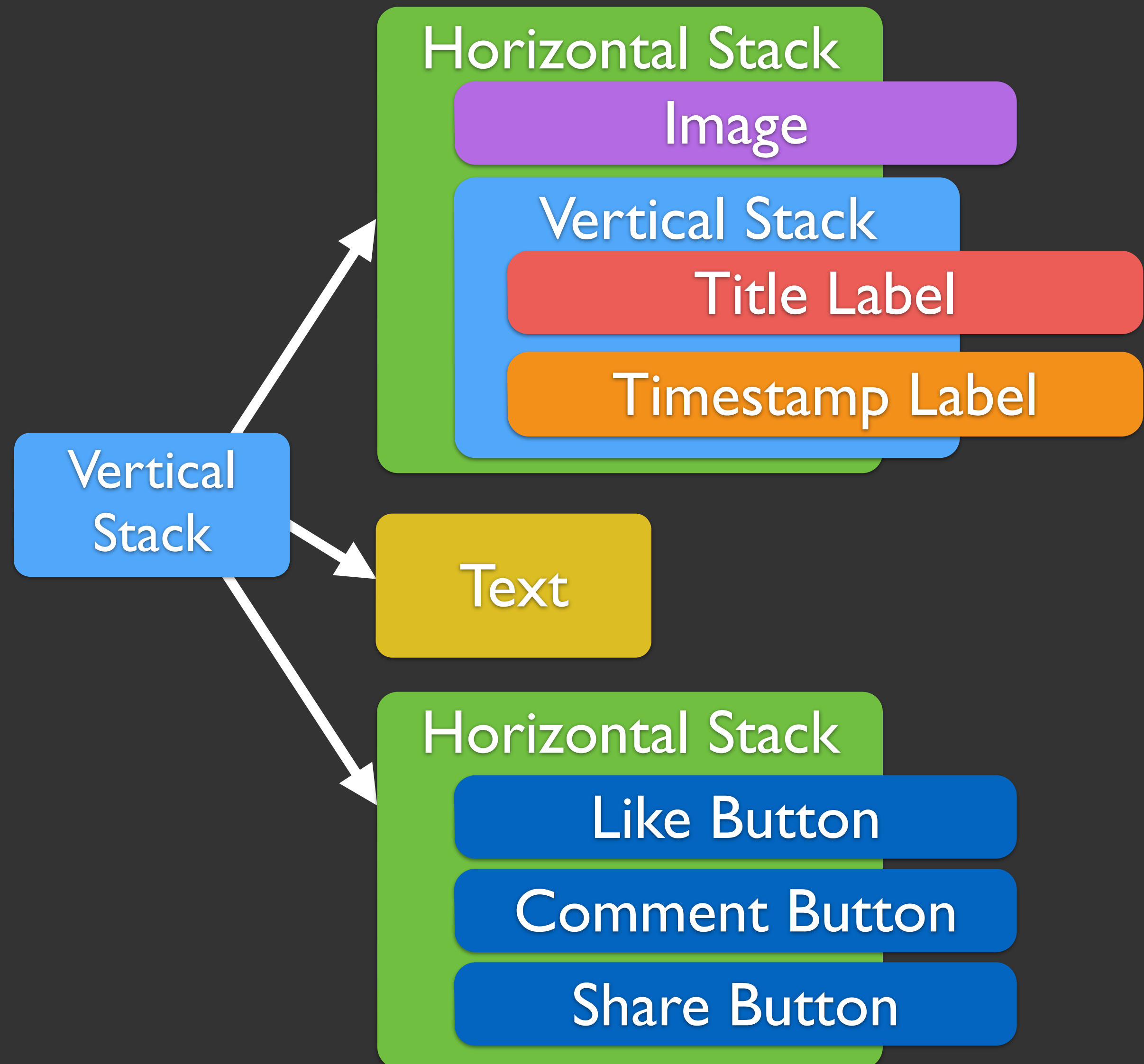
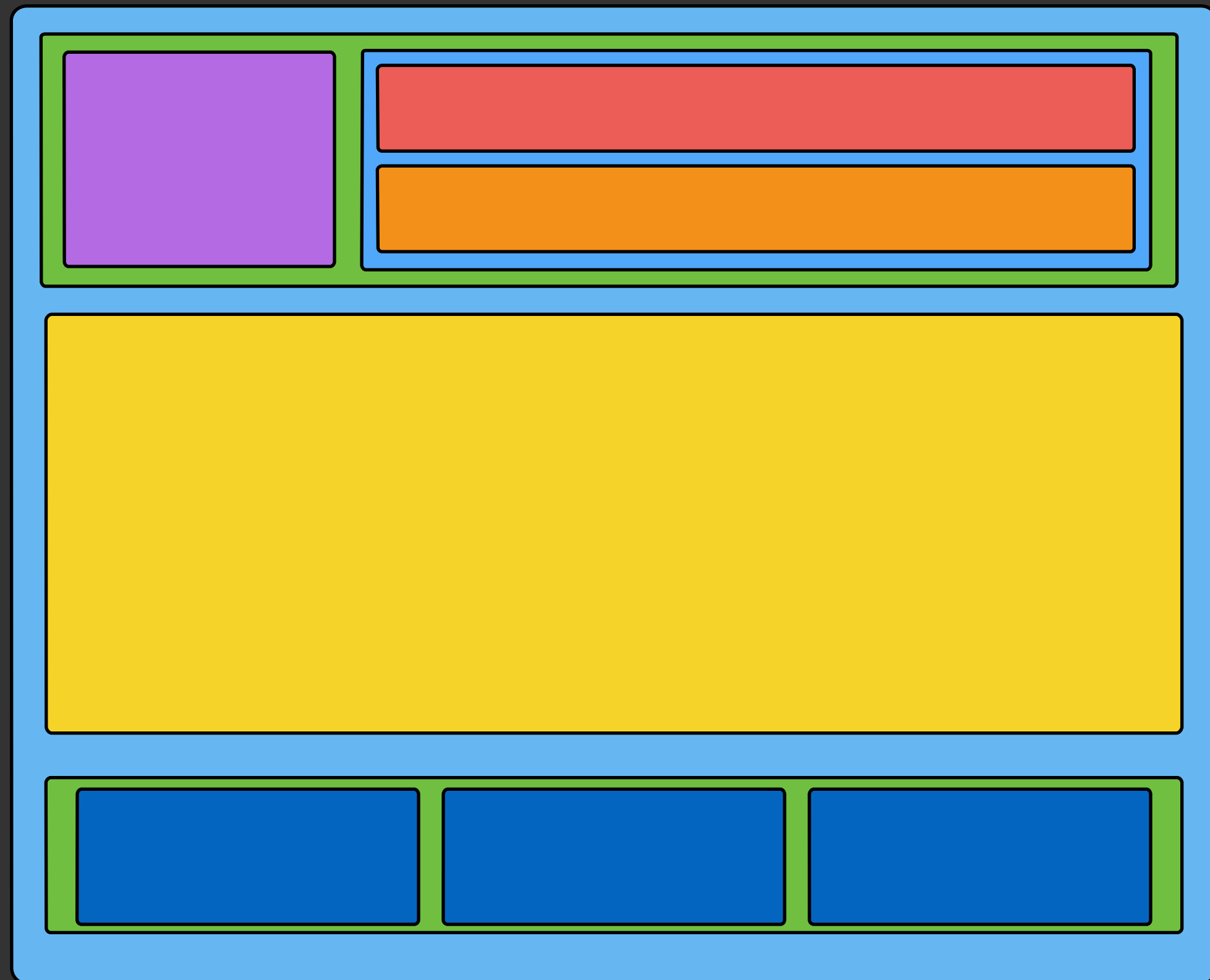
Layout of a Feed Story



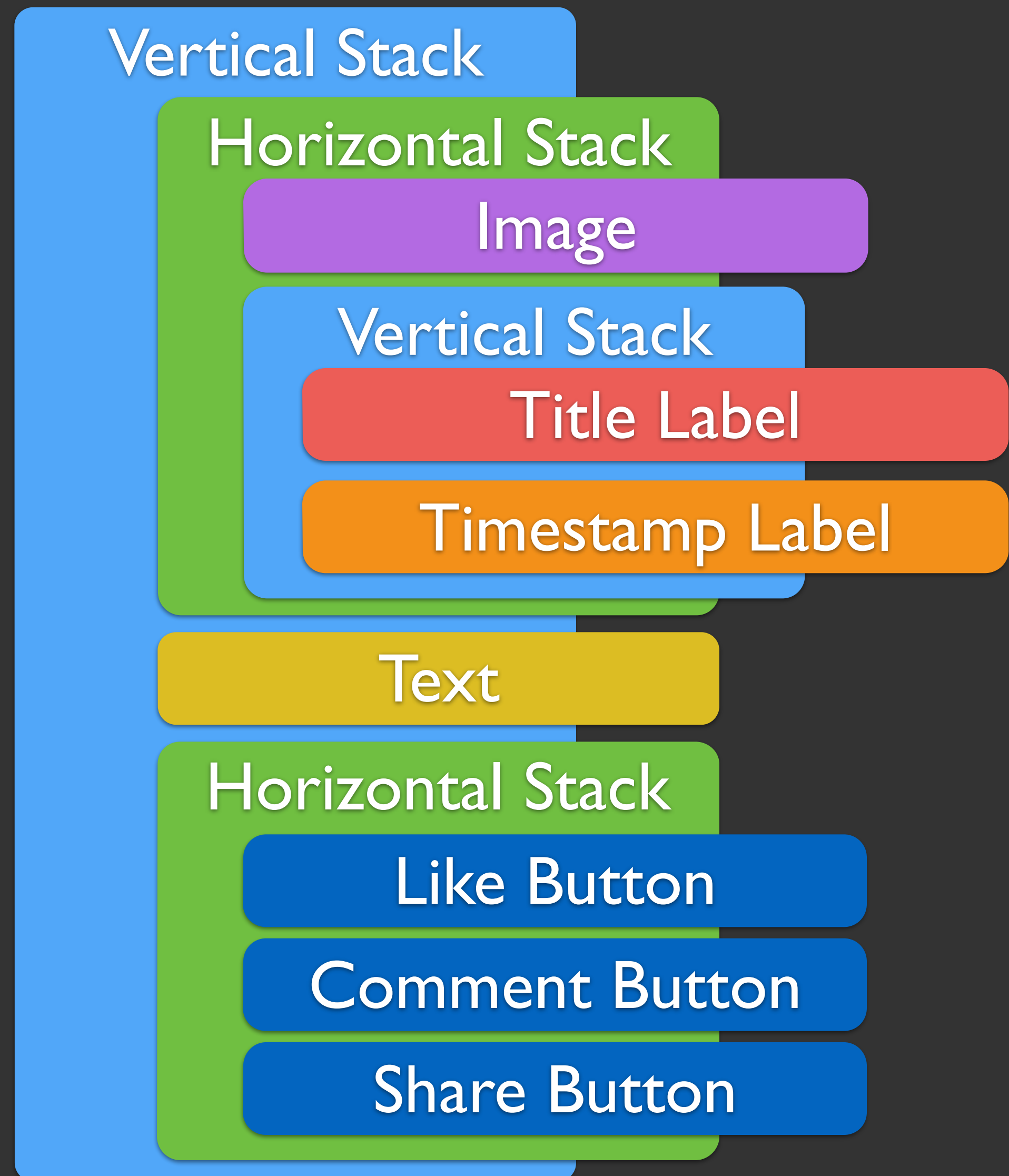
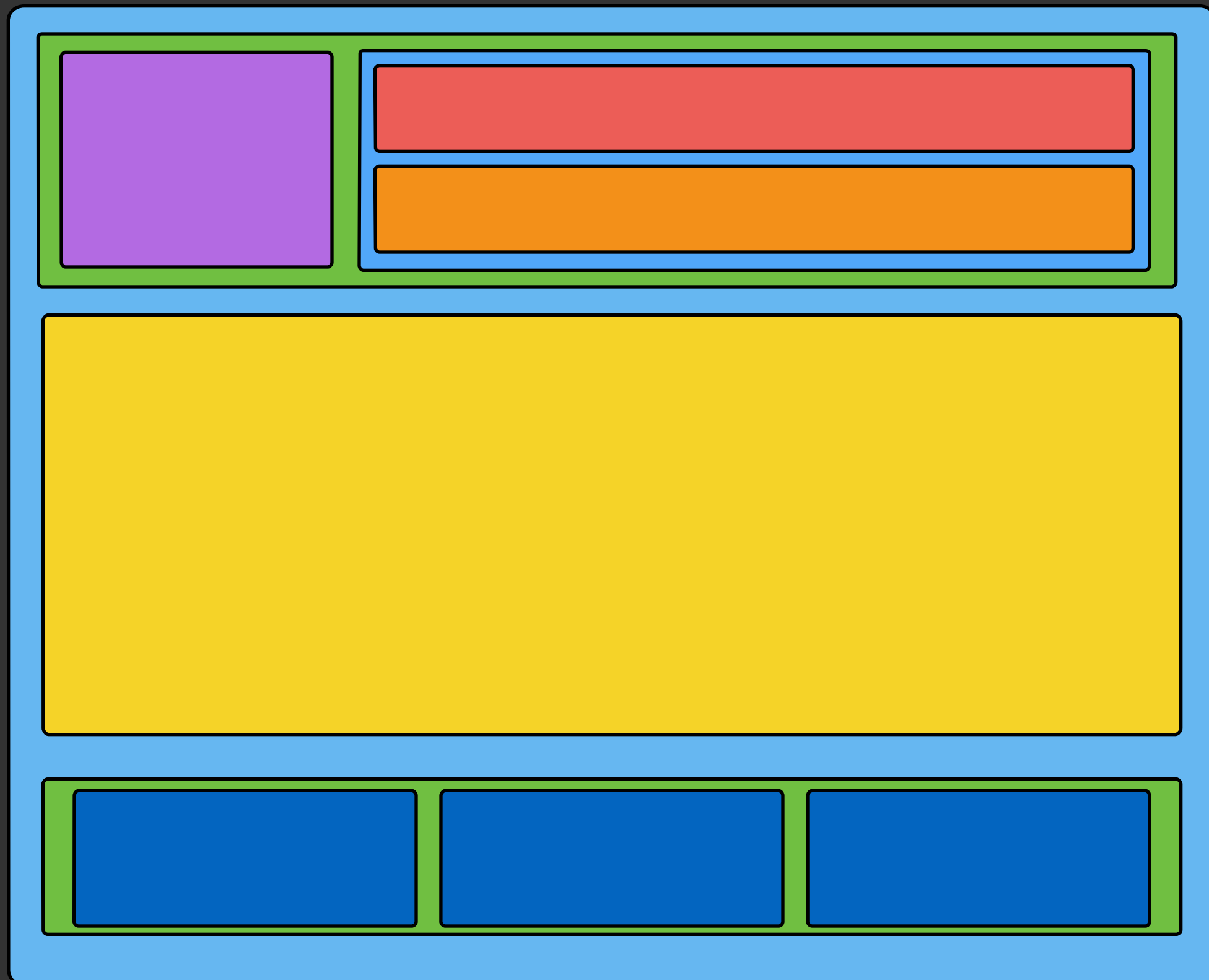
Layout of a Feed Story



Layout of a Feed Story



Layout of a Feed Story



Feed Story Header

```
<Stack direction="horizontal" spacing=10>  
  <Image contents={image} />  
  <Stack direction="vertical" spacing=12>  
    <Label text={title} />  
    <Label text={timestamp} />  
  </Stack>  
</Stack>
```

Horizontal Stack

Image

Vertical Stack

Title Label

Timestamp Label

Feed Story Header

```
[ CPStackLayoutComponent  
  newWithStyle:{  
    ...  
  }  
  children:{  
    ...  
  } ]
```

Horizontal Stack

Image

Vertical Stack

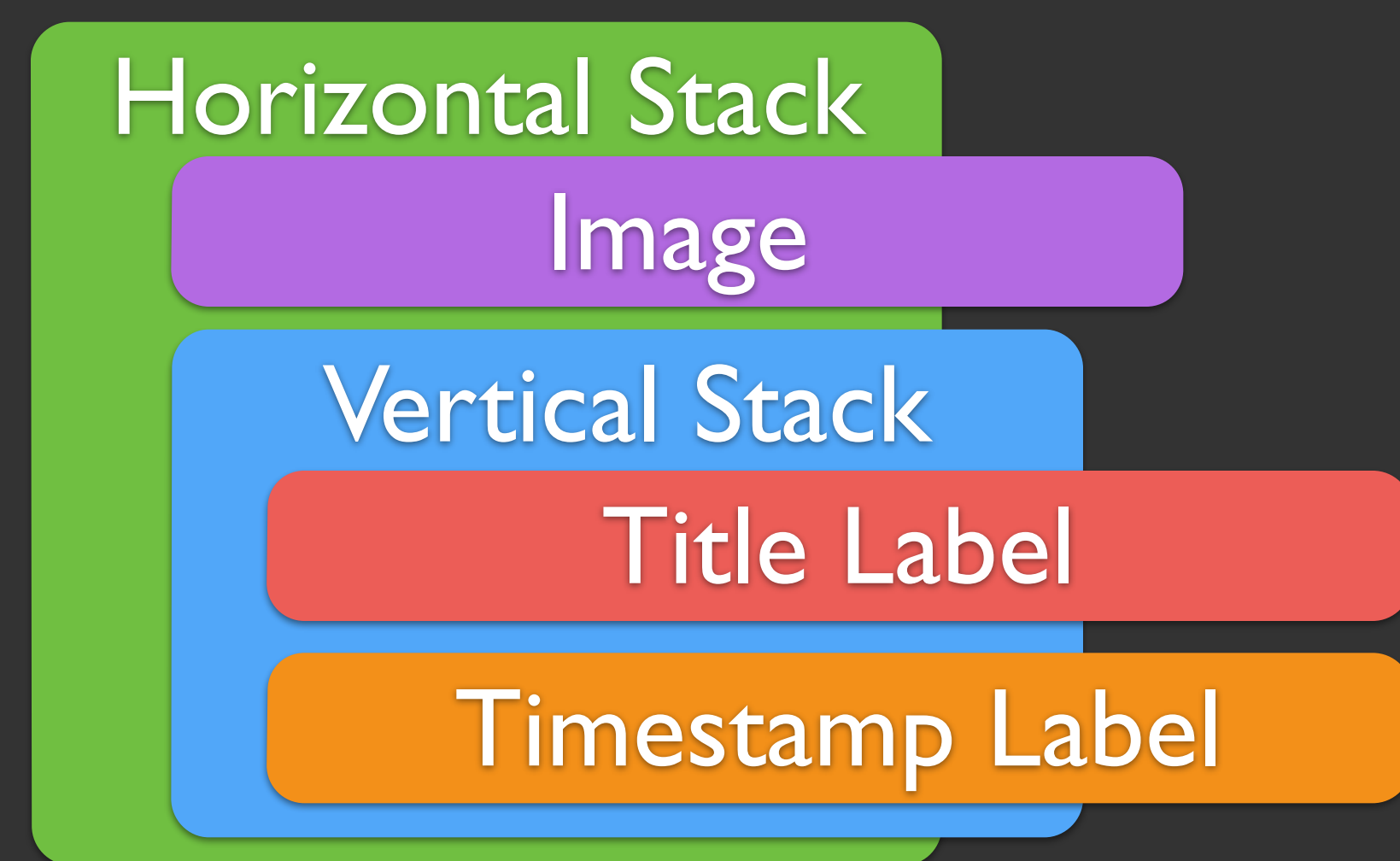
Title Label

Timestamp Label

Feed Story Header

```
[CPStackLayoutComponent  
  newWithStyle:{  
    ...  
  }  
  children:{  
    ...  
  }]  
  children:@[  
    ...  
  ] ?
```

```
children:{  
  {[Foo new], .bar = 7},  
  {[Baz new]}  
}  
children:@[  
  [StackLayoutChild newWithComponent:[Foo new] bar:7],  
  [StackLayoutChild newWithComponent:[Baz new]]  
]
```



Feed Story Header

```
[ CPStackLayoutComponent  
  newWithStyle:{  
    ...  
  }  
  children:{  
    ...  
  } ]
```

Horizontal Stack

Image

Vertical Stack

Title Label

Timestamp Label

Feed Story Header

```
[CPStackLayoutComponent  
  newWithStyle:{  
    .direction = CPStackLayoutDirectionHorizontal,  
    .spacing = 10  
  }  
  children:{  
    ...  
  }  
]
```

Horizontal Stack

Image

Vertical Stack

Title Label

Timestamp Label

Feed Story Header

```
[CPStackLayoutComponent  
  newWithStyle:{  
    .direction = CPStackLayoutDirectionHorizontal,  
    .spacing = 10  
  }  
  children:{  
    { [CPImageComponent newWithImage:image] },  
    ...  
  }  
]
```

Horizontal Stack

Image

Vertical Stack

Title Label

Timestamp Label

Feed Story Header

```
[CPStackLayoutComponent
  newWithStyle:{
    .direction = CPStackLayoutDirectionHorizontal,
    .spacing = 10
  }
  children:{
    { [CPImageComponent newWithImage:image] },
    { [CPStackLayoutComponent
      newWithStyle:{
        ...
      }
      children:{
        ...
      }
    ] }
  ]
}
```

Horizontal Stack

Image

Vertical Stack

Title Label

Timestamp Label

Feed Story Header

```
[CPStackLayoutComponent
  newWithStyle:{
    .direction = CPStackLayoutDirectionHorizontal,
    .spacing = 10
  }
  children:{
    { [CPImageComponent newWithImage:image] },
    { [CPStackLayoutComponent
      newWithStyle:{
        .direction = CPStackLayoutDirectionVertical,
        .spacing = 12
      }
      children:{
        ...
      }
    ] }
  ]
}
```

Horizontal Stack

Image

Vertical Stack

Title Label

Timestamp Label

Feed Story Header

```
[CPStackLayoutComponent
  newWithStyle:{
    .direction = CPStackLayoutDirectionHorizontal,
    .spacing = 10
  }
  children:{
    {[CPImageComponent newWithImage:image]},
    {[CPStackLayoutComponent
      newWithStyle:{
        .direction = CPStackLayoutDirectionVertical,
        .spacing = 12
      }
      children:{
        {[CPLabelComponent newWithText:title]},
        ...
      ]}]
  ]]
```

Horizontal Stack

Image

Vertical Stack

Title Label

Timestamp Label

Feed Story Header

```
[CPStackLayoutComponent
newWithStyle:{
  .direction = CPStackLayoutDirectionHorizontal,
  .spacing = 10
}
children:{
  {[CPImageComponent newWithImage:image]},
  {[CPStackLayoutComponent
newWithStyle:{
  .direction = CPStackLayoutDirectionVertical,
  .spacing = 12
}
children:{
  {[CPLabelComponent newWithText:title]},
  {[CPLabelComponent newWithText:timestamp]}
}]}
}]
```

Horizontal Stack

Image

Vertical Stack

Title Label

Timestamp Label

Feed Story Header

```
[CPStackLayoutComponent
  newWithStyle:{
    .direction = CPStackLayoutDirectionHorizontal,
    .spacing = 10
  }
  children:{
    {[CPImageComponent newWithImage:image]},
    {[CPStackLayoutComponent
      newWithStyle:{
        .direction = CPStackLayoutDirectionVertical,
        .spacing = 12
      }
      children:{
        {[CPLabelComponent newWithText:title]},
        {[CPLabelComponent newWithText:timestamp]}
      ]}]
  ]}]
```

Feed Story Header

```
[CPStackLayoutComponent
  newWithStyle:{
    .direction = CPStackLayoutDirectionHorizontal,
    .spacing = 10
  }
  children:{
    {[CPImageComponent newWithImage:image]},
    {[CPStackLayoutComponent
      newWithStyle:{
        .direction = CPStackLayoutDirectionVertical,
        .spacing = 12
      }
      children:{
        {[CPLabelComponent newWithText:title]},
        {[CPLabelComponent newWithText:timestamp]}
      ]}]
  ]}]
```

Feed Story Header

```
return [super newWithComponent:
  [CPStackLayoutComponent
    newWithStyle:{
      .direction = CPStackLayoutDirectionHorizontal,
      .spacing = 10
    }
    children:{
      {[CPIImageComponent newWithImage:image]},
      {[CPStackLayoutComponent
        newWithStyle:{
          .direction = CPStackLayoutDirectionVertical,
          .spacing = 12
        }
        children:{
          {[CPLabelComponent newWithText:title]},
          {[CPLabelComponent newWithText:timestamp]}
        }
      ]}]
    ]
];
```

Feed Story Header

```
+ (instancetype)newWithImage:(UIImage *)image
                        title:(NSString *)string
                        timestamp:(NSString *)timestamp
{
    return [super newWithComponent:
            [CPStackLayoutComponent
             newWithStyle:{
                 .direction = CPStackLayoutDirectionHorizontal,
                 .spacing = 10
             }
            children:{
                {[CPIImageComponent newWithImage:image]},
                {[CPStackLayoutComponent
                 newWithStyle:{
                     .direction = CPStackLayoutDirectionVertical,
                     .spacing = 12
                 }
                 children:{
                     {[CPLabelComponent newWithText:title]},
                     {[CPLabelComponent newWithText:timestamp]}
                 }
                ]}]
            }];
}
```

Feed Story Header

```
+ (instancetype)newWithImage:(UIImage *)image
                        title:(NSString *)string
                        timestamp:(NSString *)timestamp
{
    CPComponent *component = storyHeader(image, title, timestamp);
    return [super newWithComponent:component];
}

static CPComponent *storyHeader(UIImage *image,
                                NSString *title,
                                NSString *timestamp)
{
    return ...;
}
```

Feed Story Header

```
@interface StoryHeaderComponent : CPCCompositeComponent
+ (instancetype)newWithImage:(UIImage *)image
                        title:(NSString *)string
                        timestamp:(NSString *)timestamp;
@end

@implementation StoryHeaderComponent

+ (instancetype)newWithImage:(UIImage *)image
                        title:(NSString *)string
                        timestamp:(NSString *)timestamp
{
    CPCComponent *component = storyHeader(image, title, timestamp);
    return [super newWithComponent:component];
}

@end
```




COMPONENTS

Wrapping a View

```
[CPCOMPONENT
  newWithView: { [UIView class],
    {
      { @selector(setBackgroundColor:), [UIColor orangeColor] },
      { @selector(setAlpha:), @0.5 },
    }
  }
  size: {}
]

[CPCOMPONENT
  newWithView: { [UIImageView class],
    {
      { @selector(setImage:), [UIImage imageNamed:@"toaster.png"] },
    }
  }
  size: {}
]
```

Sizing A Component

```
[CPComponent
  newWithView:{ [UIView class],
    {
      { @selector(setBackgroundColor:), [UIColor orangeColor] },
      { @selector(setAlpha:), @0.5 },
    }
  }
  size:{ 100, 200 }
]

{ 100, Percent(50) }

{ Percent(100), 200 }

{ Percent(100), Percent(50) }

{ Auto(), Auto() } "Autonomous"

{}
```

```
{ width, height }

{
  .minWidth = 50,
  .maxWidth = 250,
  .minHeight = 0,
  .maxHeight = Percent(100),
}
```

CPCComponent

The Root Class

- Configure a(n optional) view
- Can take an “explicit” size
- Privately produces the sized view

- Immutable
- Everything is configured in `+newWith...`
- Components are like pure functions
- Internal methods for layout, view configuration, and receiving events

CPCCompositeComponent

A Class Built for Composition

- Wraps one component via +newWithComponent:
- Forwards sizing and configuration

```
@interface StoryHeaderComponent : CPCCompositeComponent
+ (instancetype)newWithImage:(UIImage *)image
                        title:(NSString *)string;
@end

@implementation StoryHeaderComponent

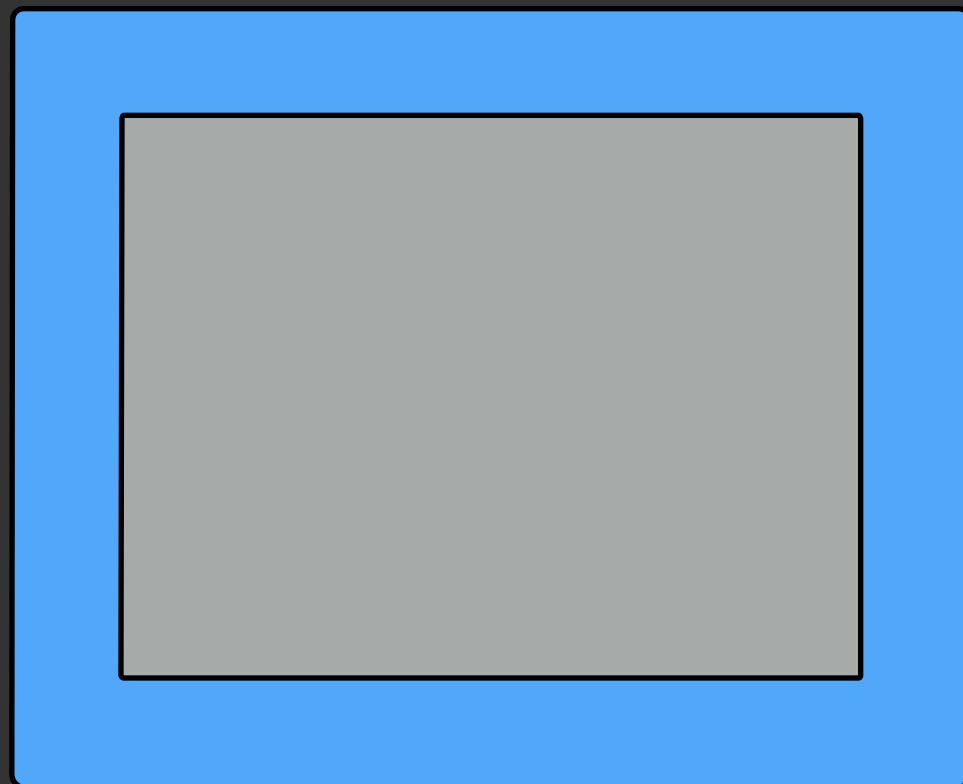
+ (instancetype)newWithImage:(UIImage *)image
                        title:(NSString *)string
{
    return [super newWithComponent:storyHeader(image, title)];
}

@end
```

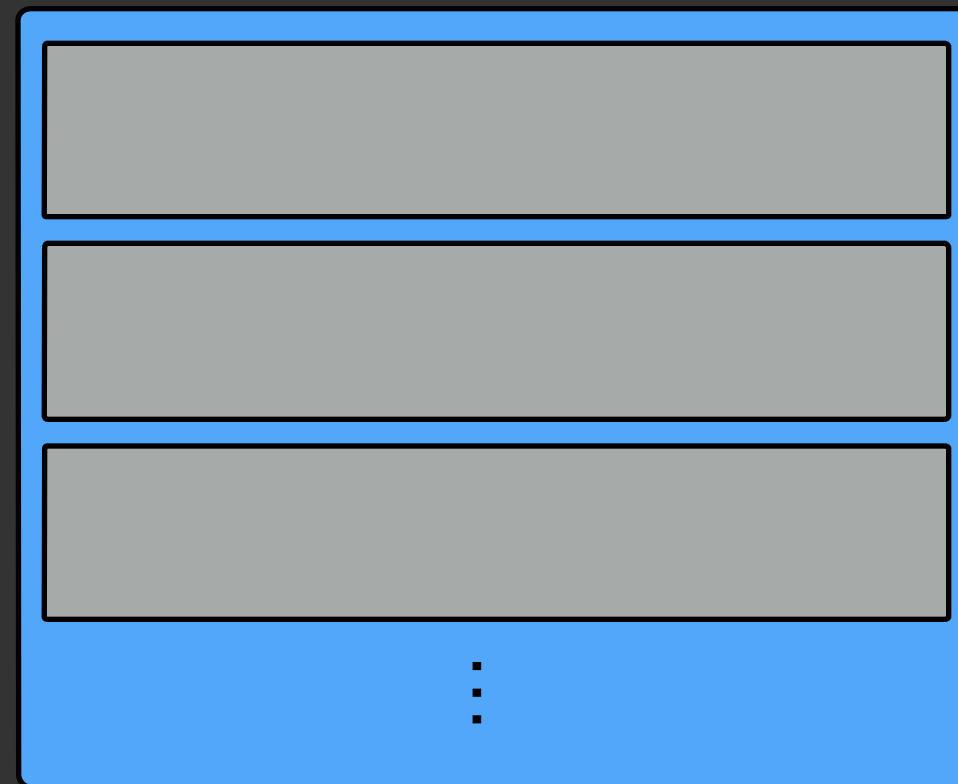
Building Blocks

Subclass `CPCCompositeComponent` and wrap a “primitive”

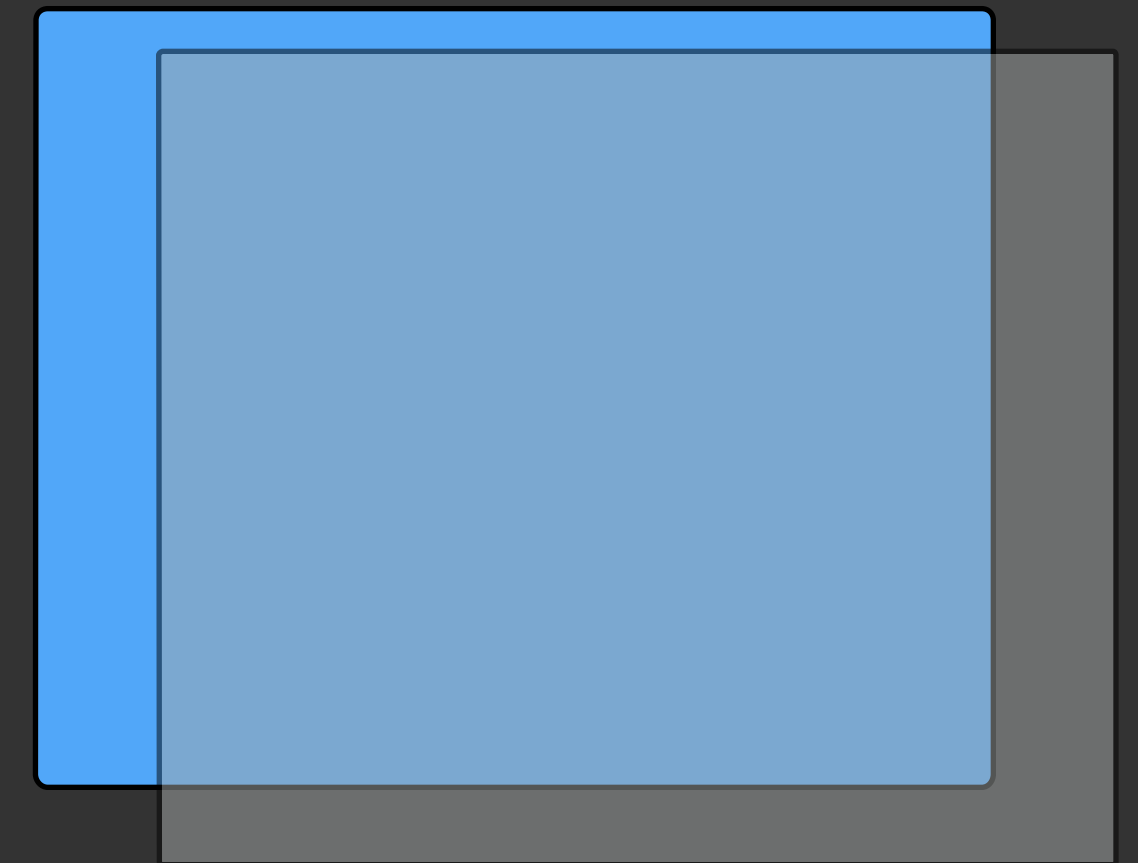
`CPInsetComponent`



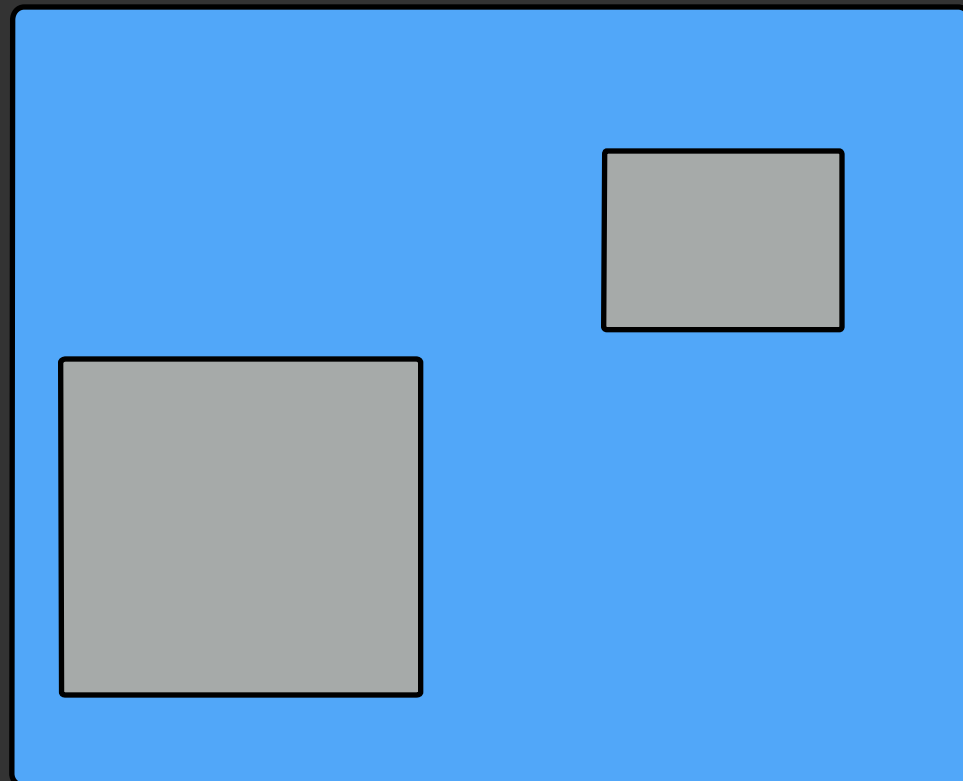
`CPStackLayoutComponent`



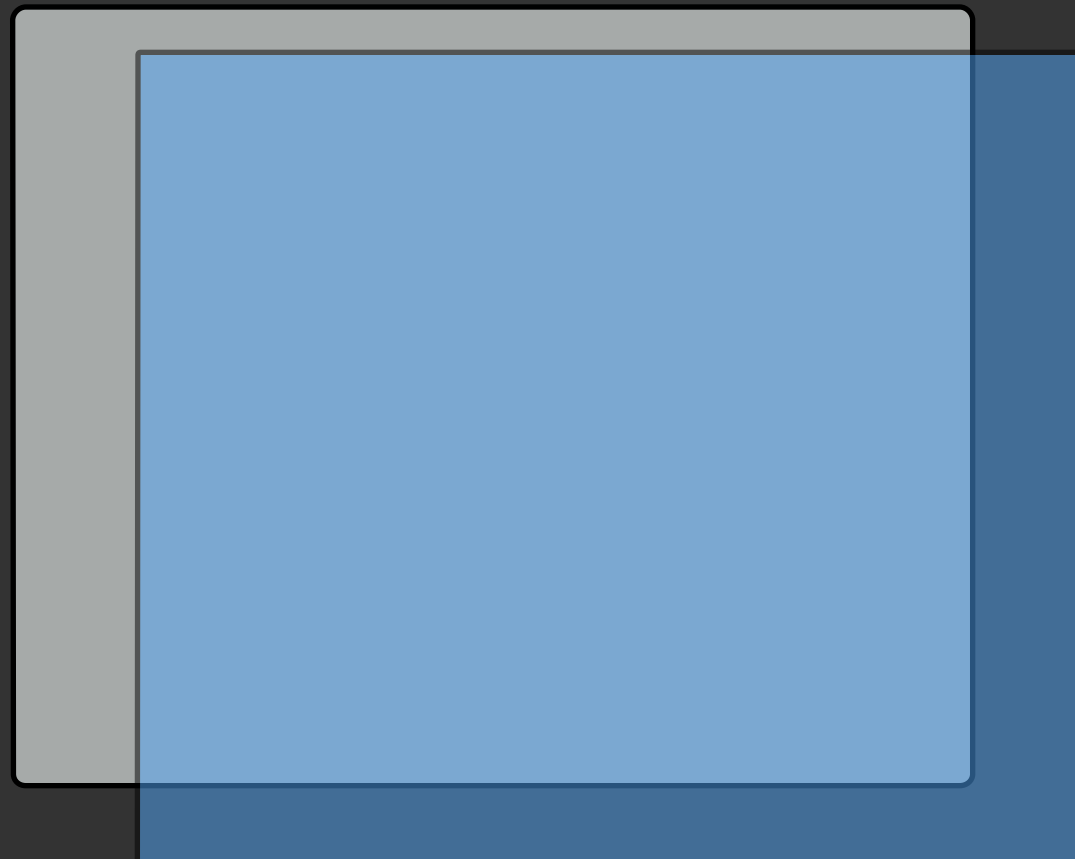
`CPOverlayComponent`



`CPStaticLayoutComponent`



`CPBackgroundComponent`

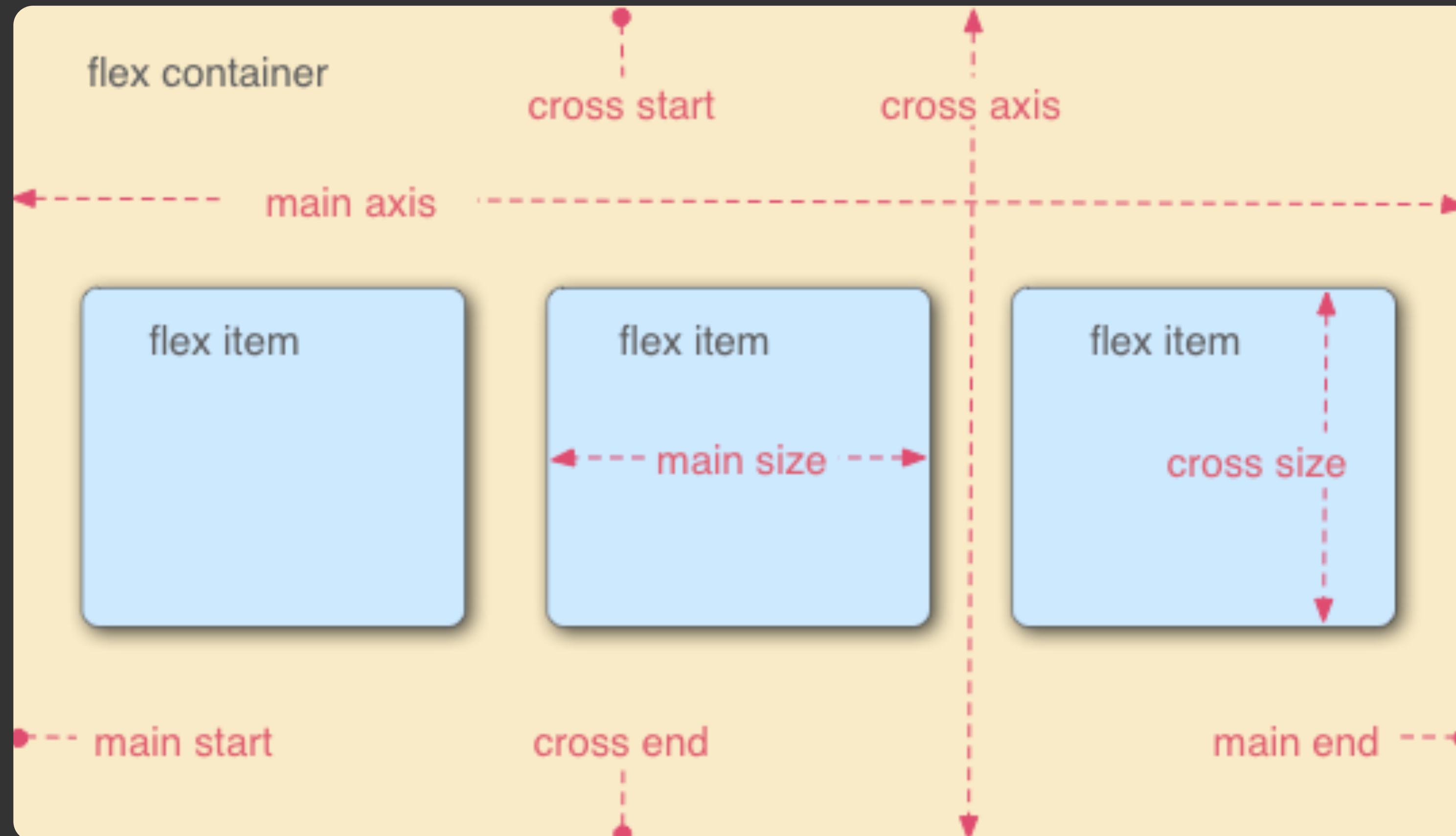


`CPTextComponent`
`CPImageComponent`
`CPButtonComponent`
`CPScrollComponent`

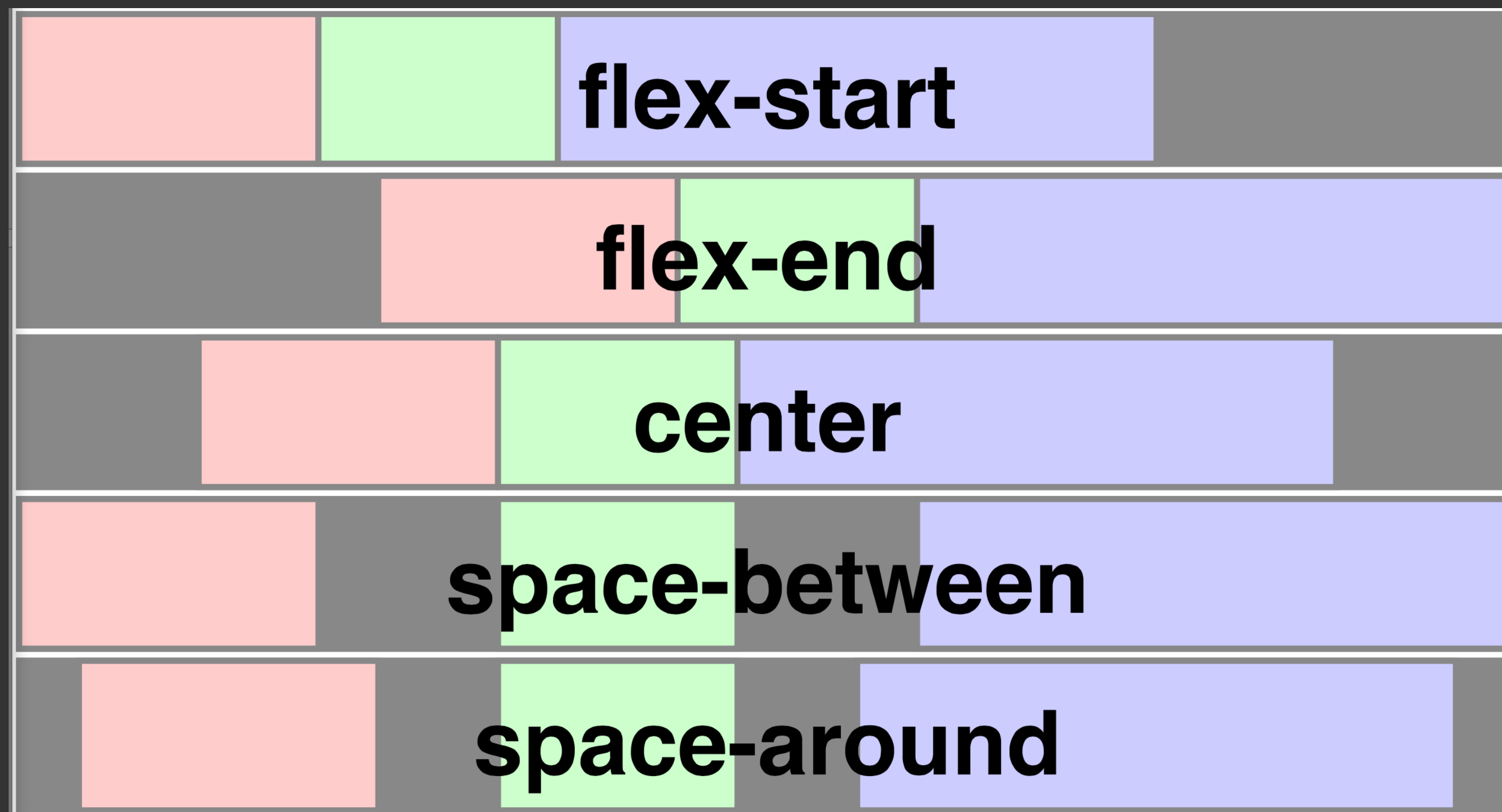
⋮



Flexbox



Flexbox



Creating Primitives

Rarer than you think

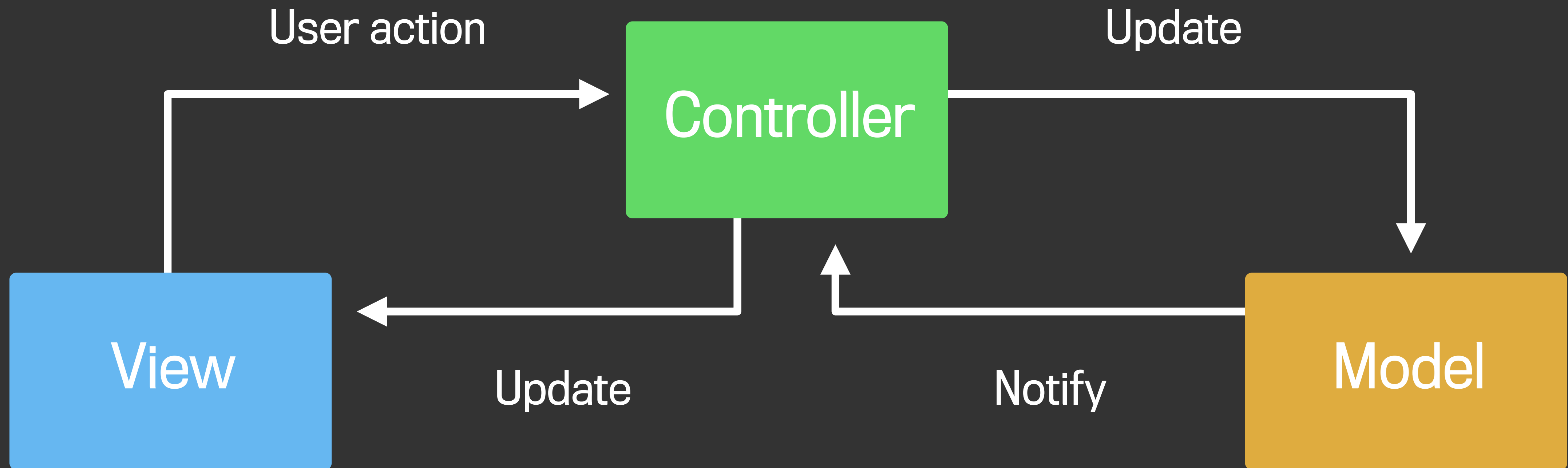
- Subclass `CPCCompositeComponent`
 - Wrap existing components
 - Wrap custom `UIViews`
- Subclass `CPCComponent`
 - Layout: implement `-layoutThatFits:`
 - Configuration: override `-mountInContext:`



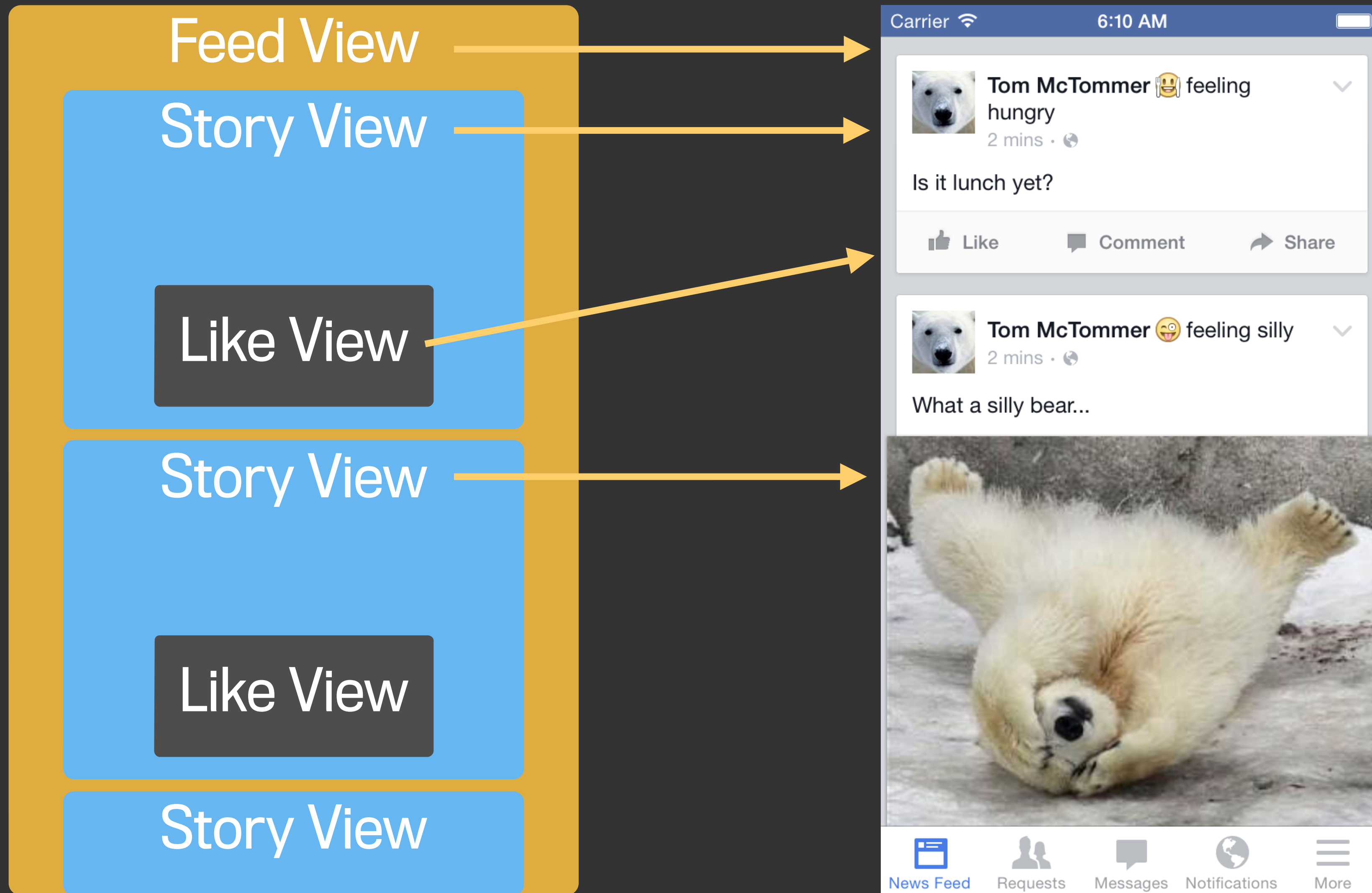
DATA FLOW

MVC

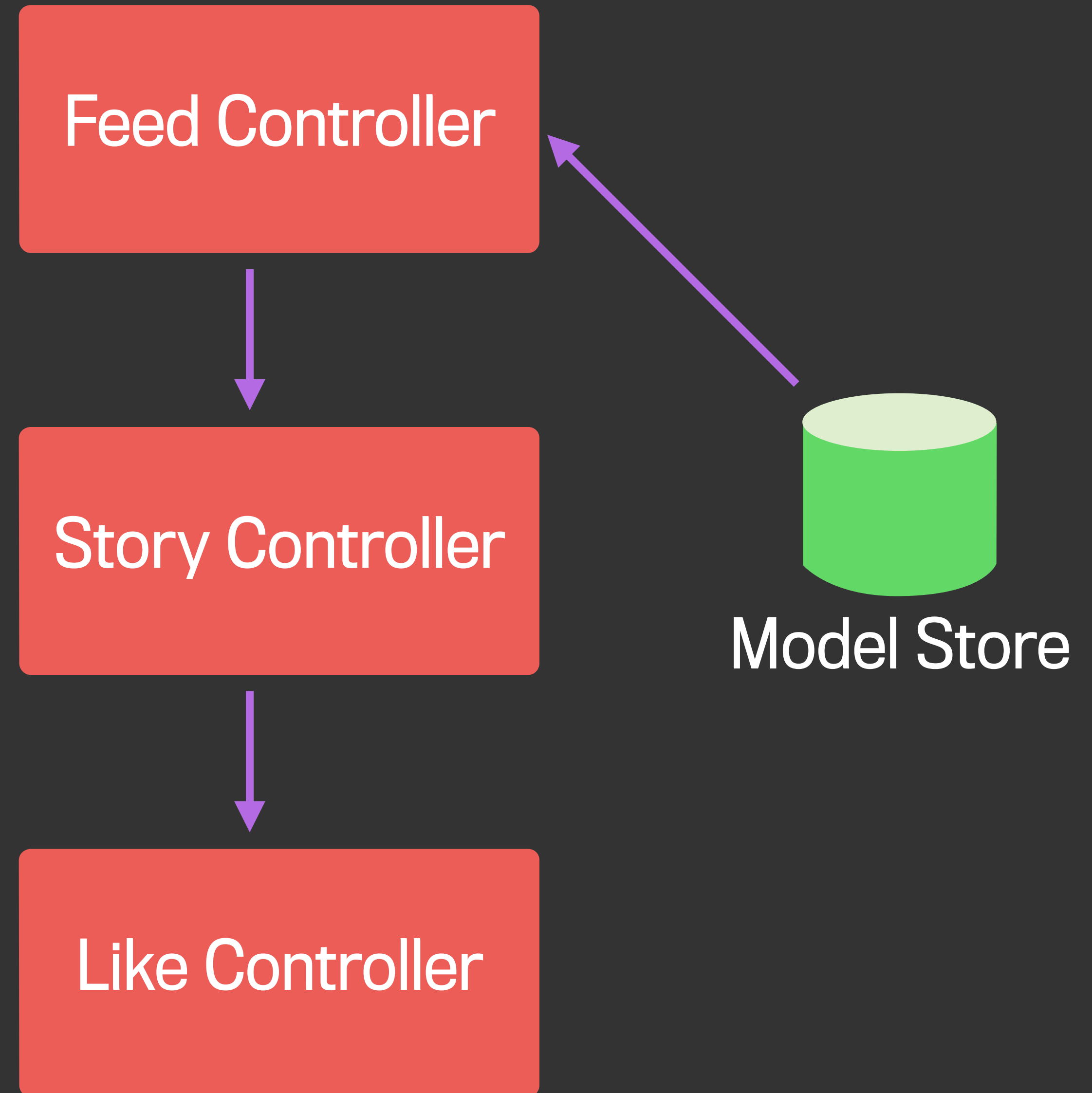
MVC: Model-View-Controller



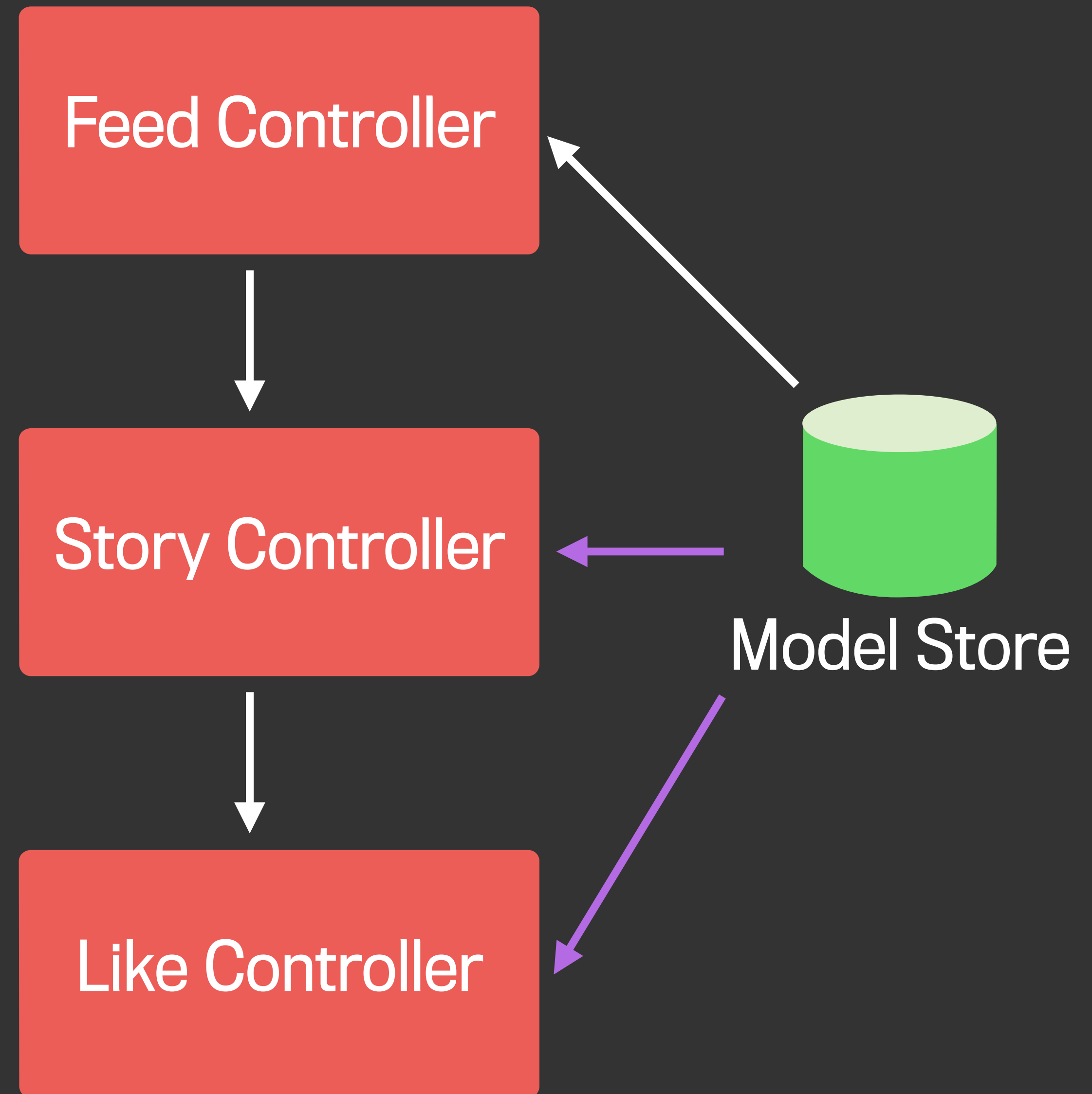
MVC - Views



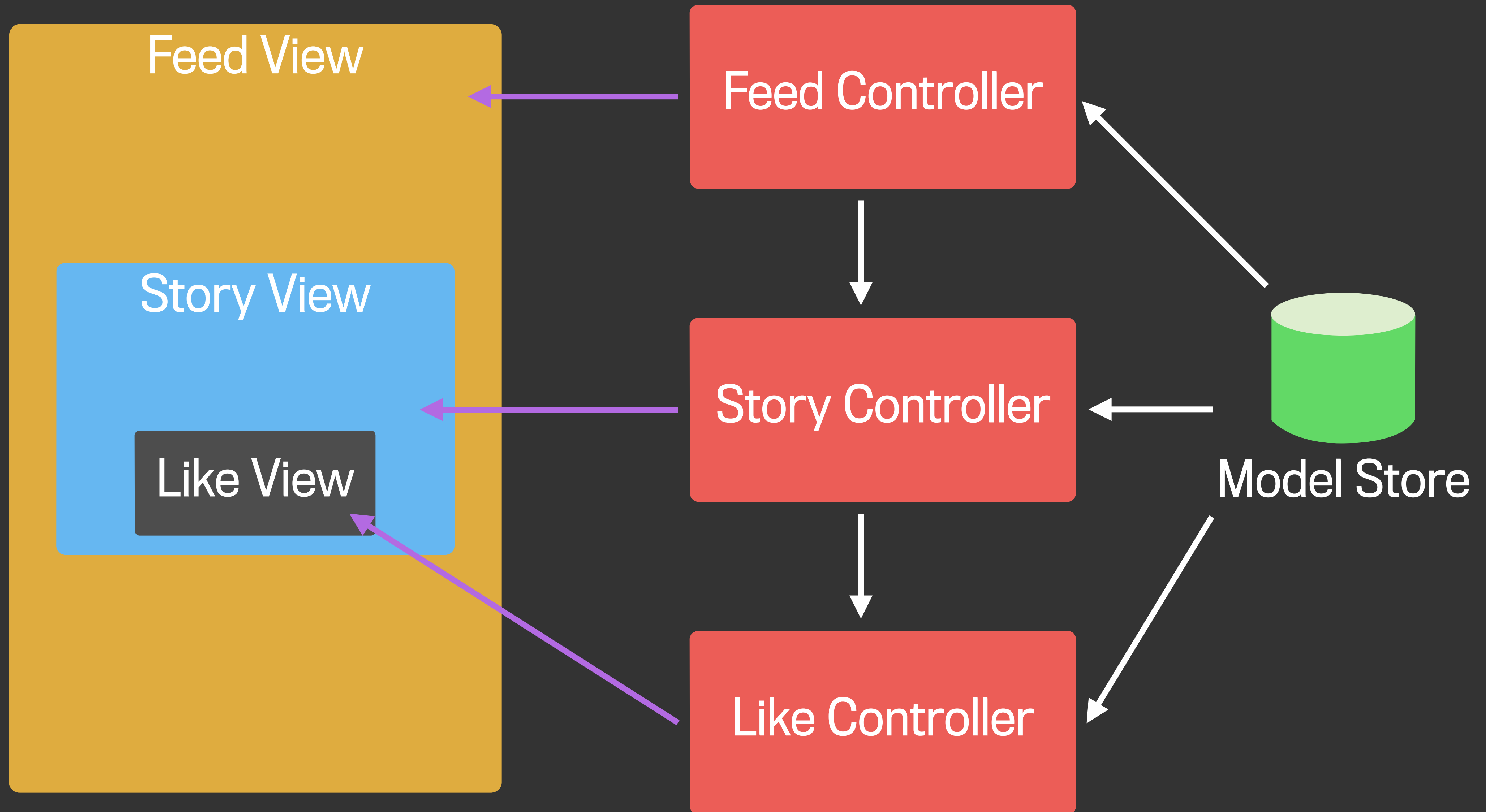
MVC Data and Event Flow



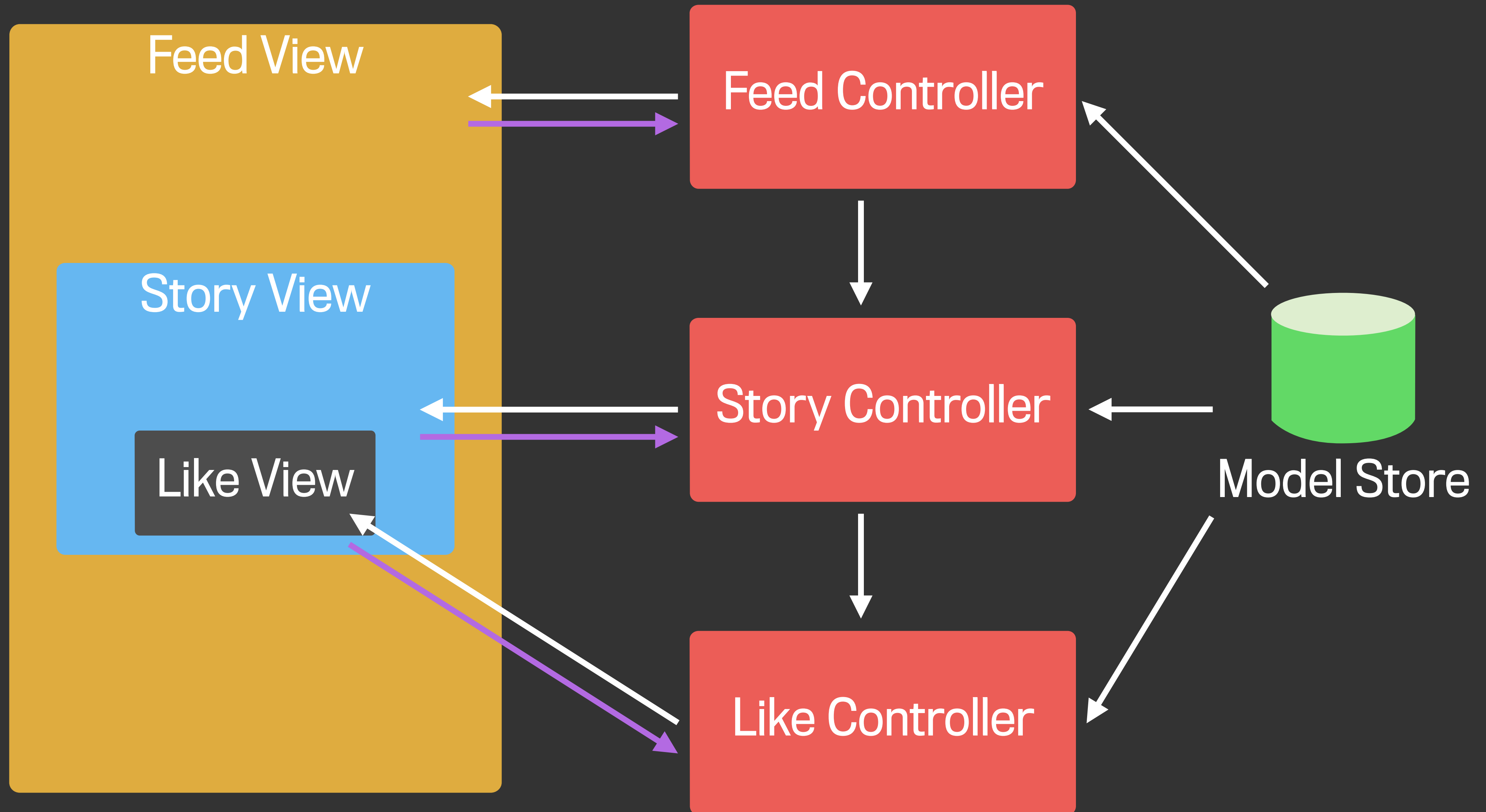
MVC Data and Event Flow



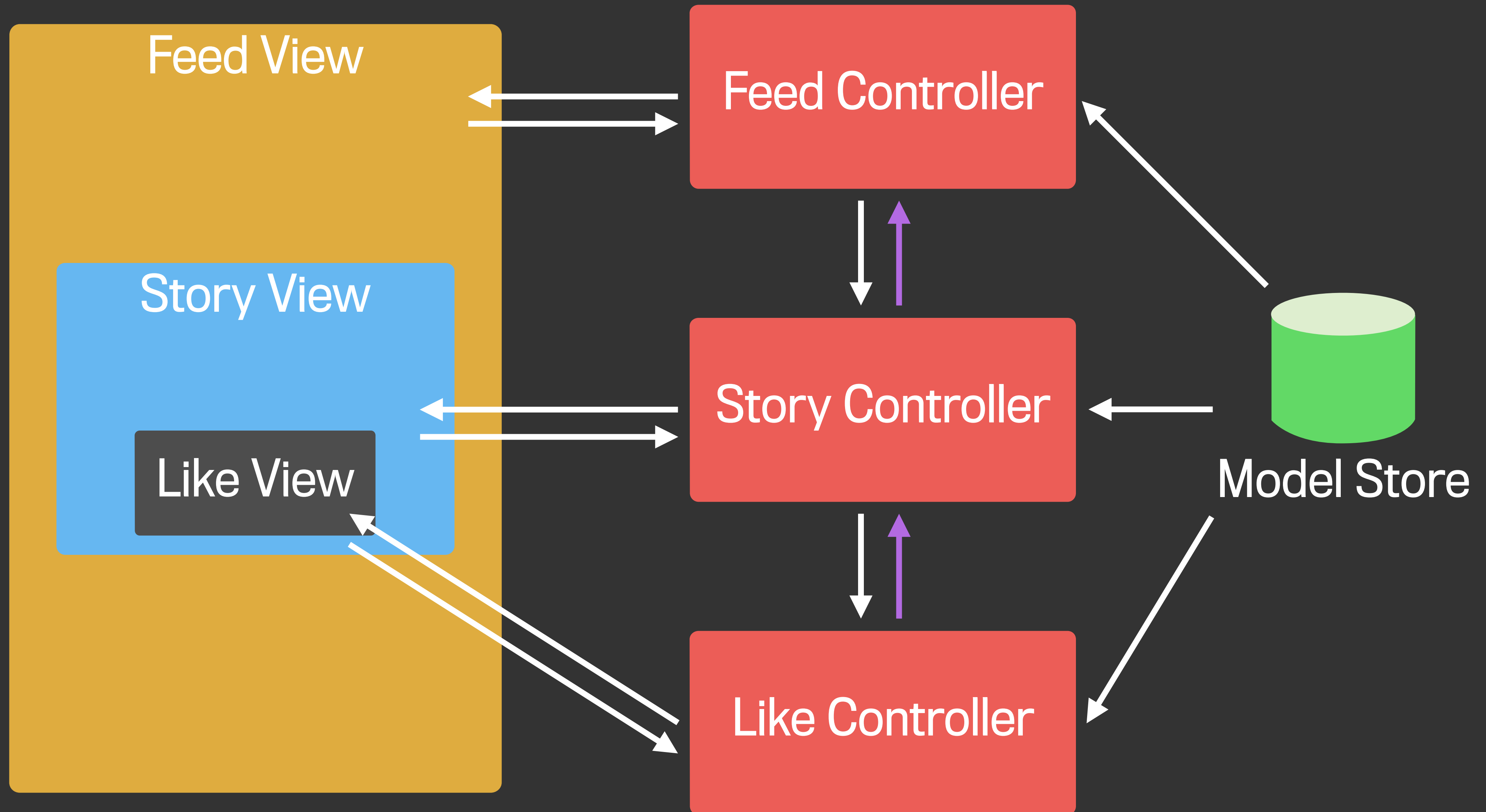
MVC Data and Event Flow



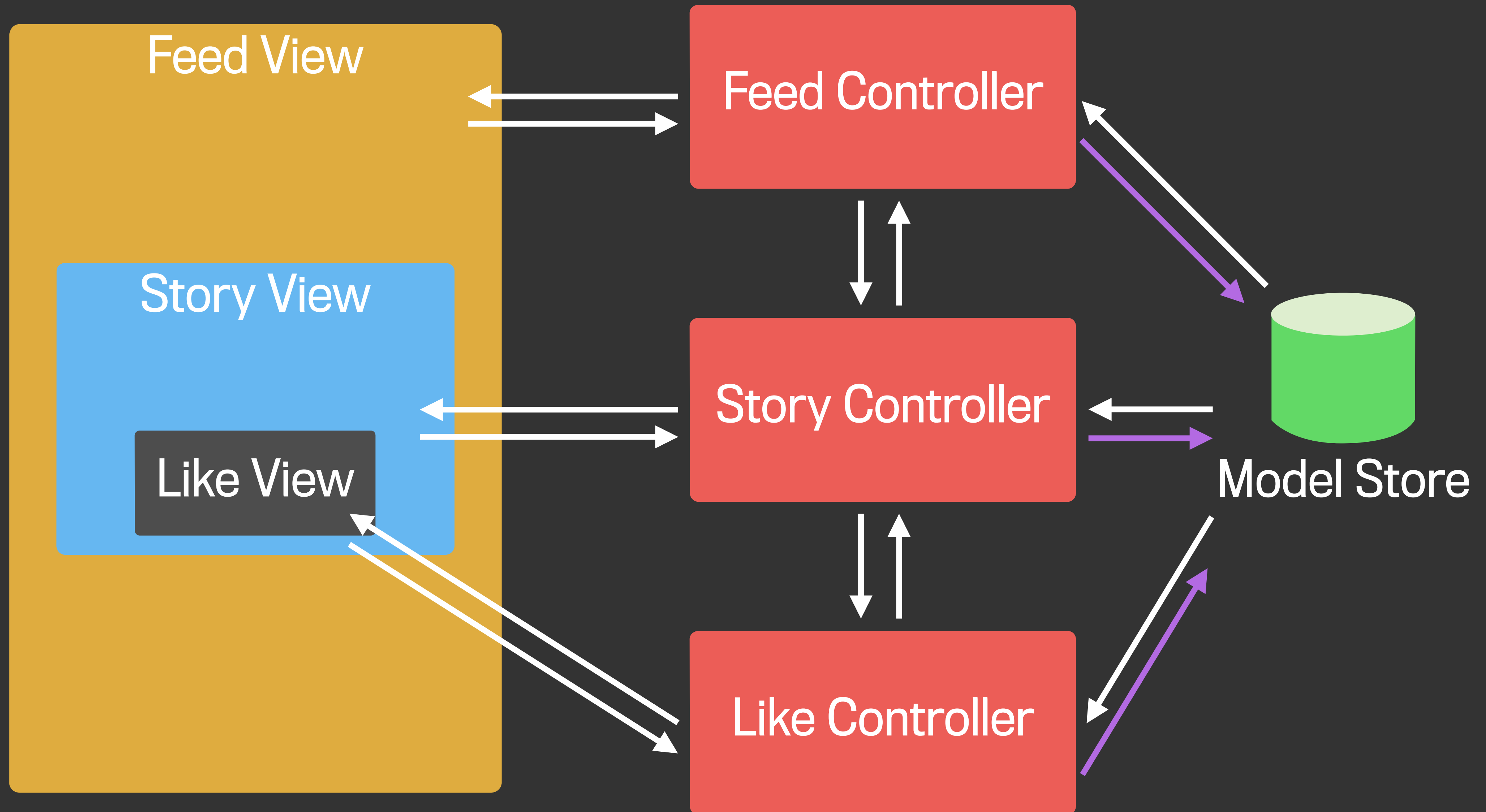
MVC Data and Event Flow



MVC Data and Event Flow



MVC Data and Event Flow



 **H. Wade Minter**
25 minutes ago

I'm the DJ, he's the rapper. — at **Eagle Fun Fest.**

1 Like 1 Comment

 Like  Comment

 **Brian Dewey**
27 minutes ago in Seattle

Judging from Facebook, all of my friends are traveling someplace cool this weekend. Jealous...

2 Comments

 **Hilary Hahn**
28 minutes ago

On April 28, many artists gathered together to honor Beate Gordon, former Asia Society Director of Performing Arts. One of those was "In 27 Pieces" composer Somei Satoh. You can watch a full video of the concert

MUTABILITY

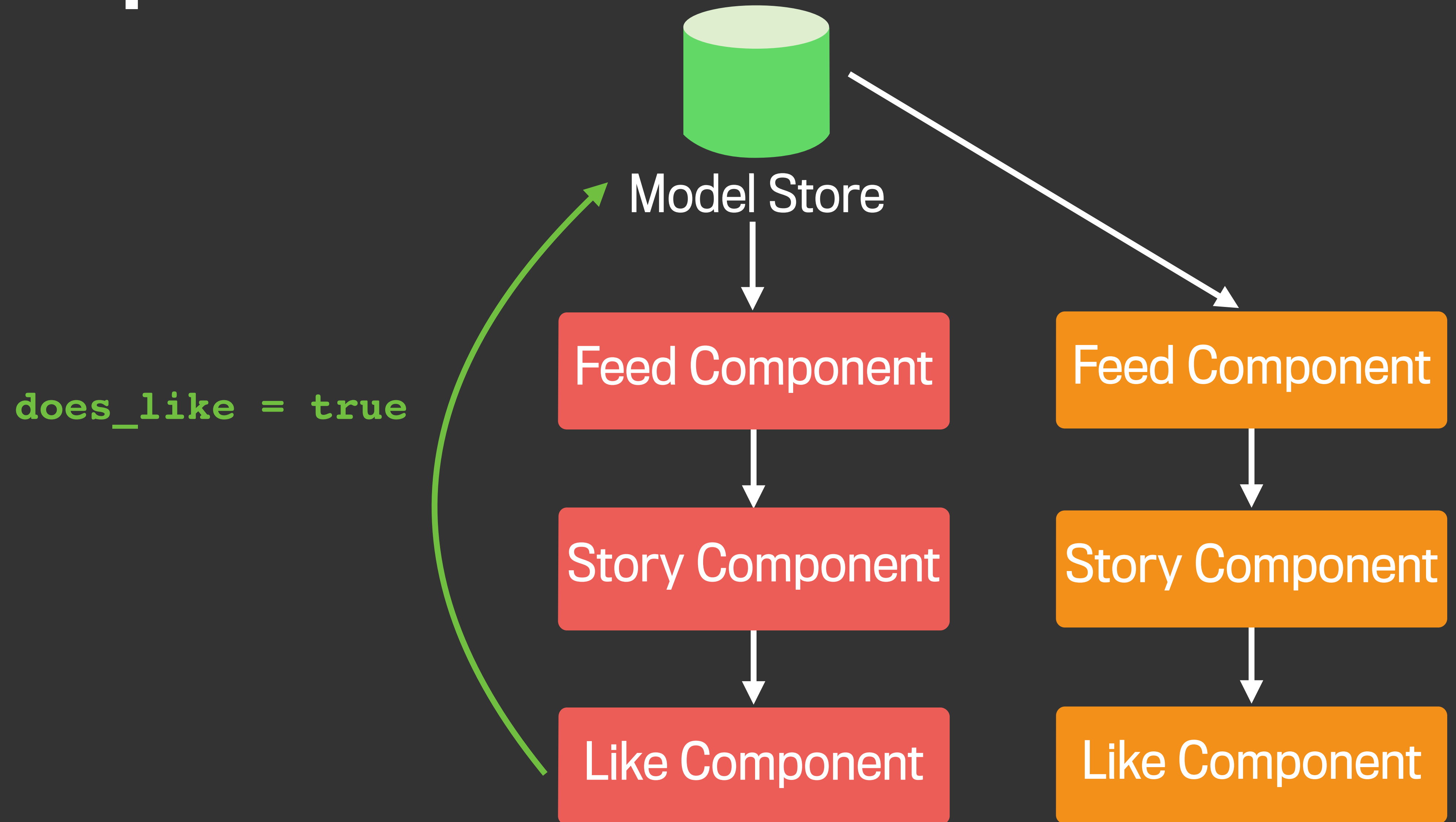


Immutable

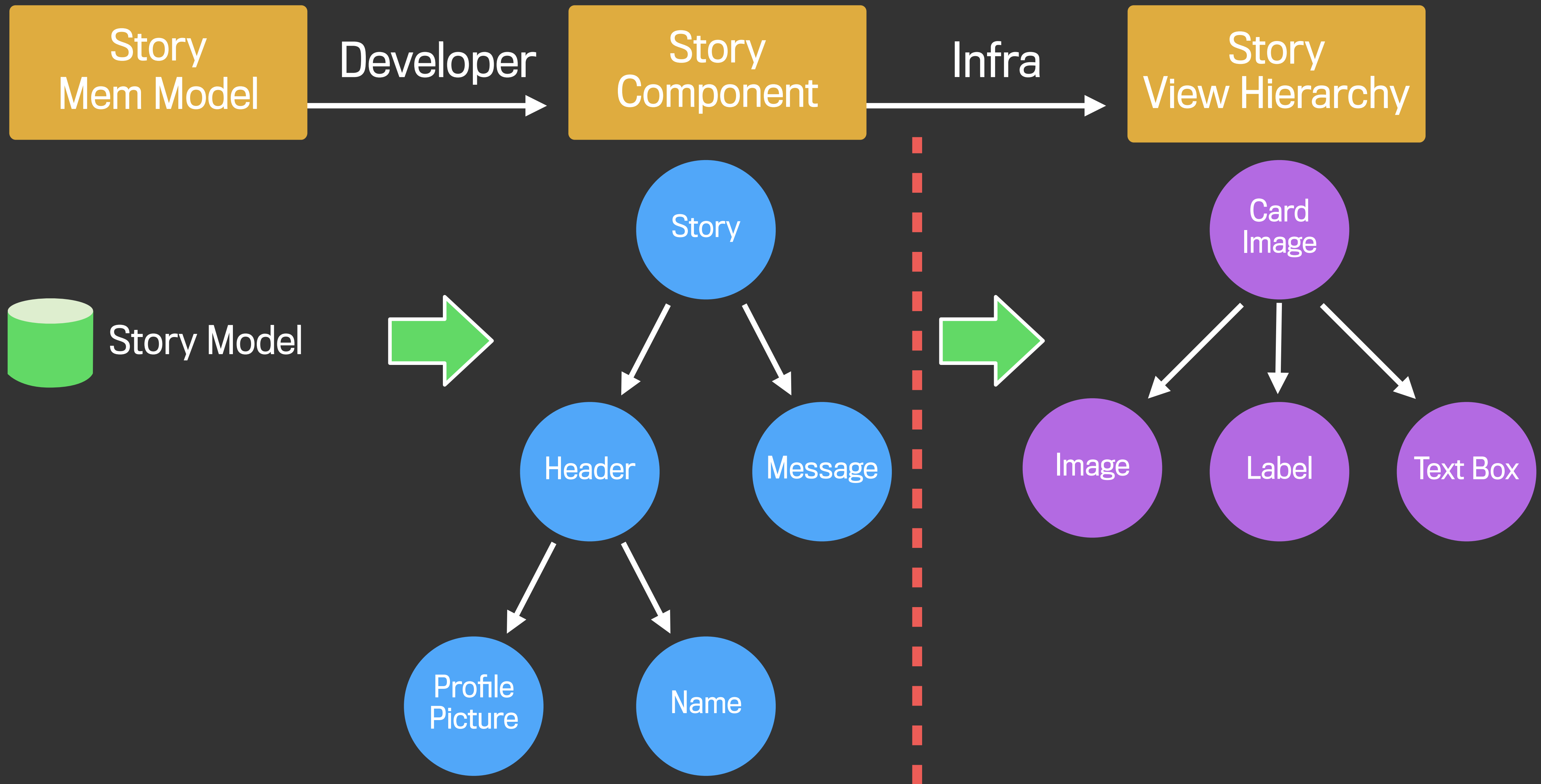
```
- (void)userTappedLike
{
    [FBAPI sendLikeRequestForStory:self.story];
    self.story.doesLike = YES;
    self.story.likeCount += 1;
    self.likeButton.selected = YES;
    [self.likeButton performAnimation:FBLikeAnimation()];
    self.likeCountLabelText = [self likeCountText];
    if (self.likeCountLabel.hidden) {
        self.likeCountLabel.hidden = NO;
    }
    [self.view setNeedsLayout];
}

- (void)userTappedLike
{
    [FBMutator applyLikeMutation:self.story];
}
```

Components Data and Event Flow



Rendering Pipeline





ONE WAY

CPU MUCH?

Optimizations

Keeping the CPU under control

- Allocate and size components off the UI thread
- Recycle views
- Stack-allocate objects

- Minimize configuration by differentiating attributes that:
 - can be smashed over
 - need to be “unconfigured”
 - have a shortcut if the attribute has already been applied

View Recycling

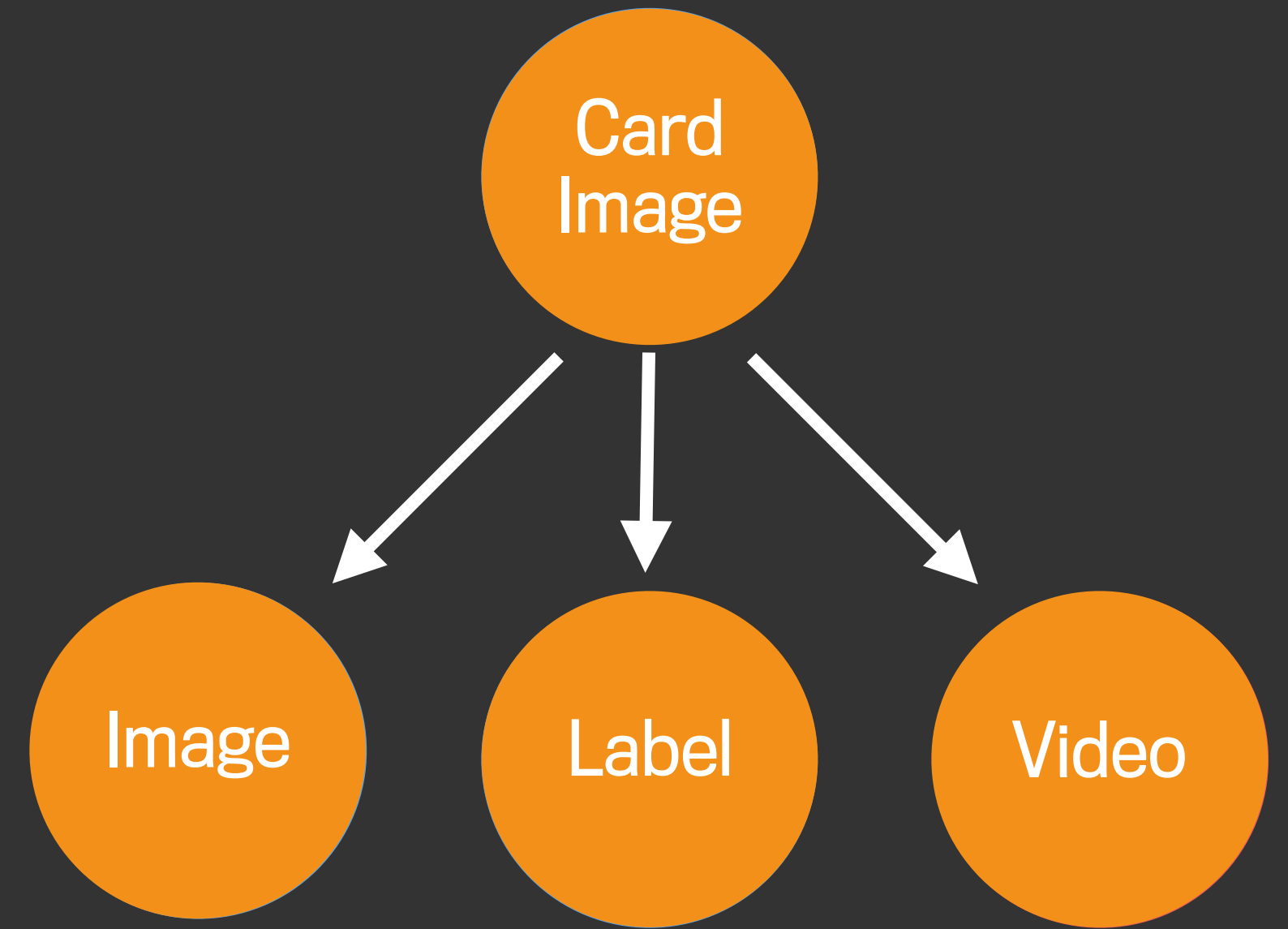
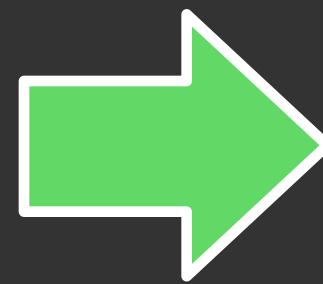
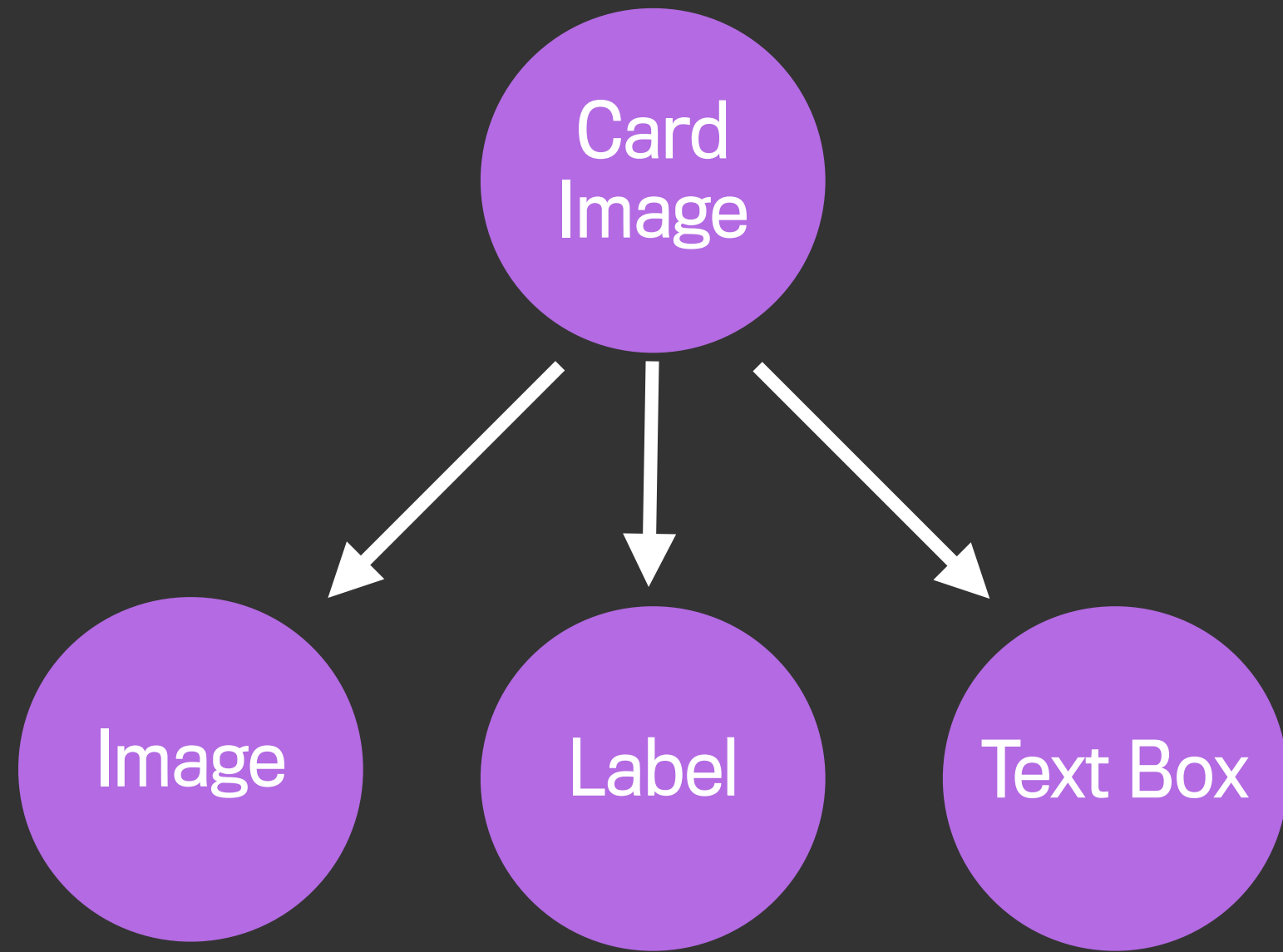


Image Label at the top.

The best text-box the world has ever seen.
Totes.

Image Wicked cool label.

INTERNAL STATE

Immutability and Encapsulation

- Components are immutable (short-lived)
- Components cannot have mutable state
- New state can be handed to new components
- Feels just like they are long lived

Component State

```
+ (instancetype)newWithColor:(UIColor *)color
{
    NSNumber *alpha = self.state(^id {
        return @1.0;
    });

    return [super newWithComponent:
        [FBComponent newWithView:[UIView class], {
            { @selector(setBackgroundColor:), color },
            { @selector(setAlpha:), alpha },
        } size:{}]];
}

- (void)handleDimEvent:(id)sender
{
    [self updateState:^id(NSNumber *currentValue) {
        return @([currentValue floatValue] * 0.8);
    }];
}
```


**PUTTING IT ALL
TOGETHER**

Don't make me...

- Configure views
- Implement view recycling
- Write math for layout and sizing
- Listen for state changes
- Worry about threads and mutability

Do the hard work for me

- “This is the view hierarchy I want.”
- The **system** recycles and configures views.
- “This is the layout I want.”
- The **system** does all the math.
- “This is the state-change I want.”
- The **system** runs the component creation function and provides the new state.

- **Declare** all the things.



BETTER
PROGRAMMING MODEL

70% LESS CODE

TO WRITE NEWS FEED

PERFORMANT

OUT OF THE BOX

Some of the News Feed Story Component

```
[ CPStackLayoutComponent
  newWithView: {}
  style: {
    .direction = CPStackLayoutDirectionVertical,
    .alignItems = CPStackLayoutAlignItemsStretch,
    .spacing = 10,
  }
  children: {
    { [ CPStoryHeaderComponent newWithStory:story ] },
    { [ CPStoryMessageComponent newWithStory:story ] },
    { [ CPStoryLikeCommentComponent newWithStory:story ] }
  } ]
```

THE END

QUESTIONS?