

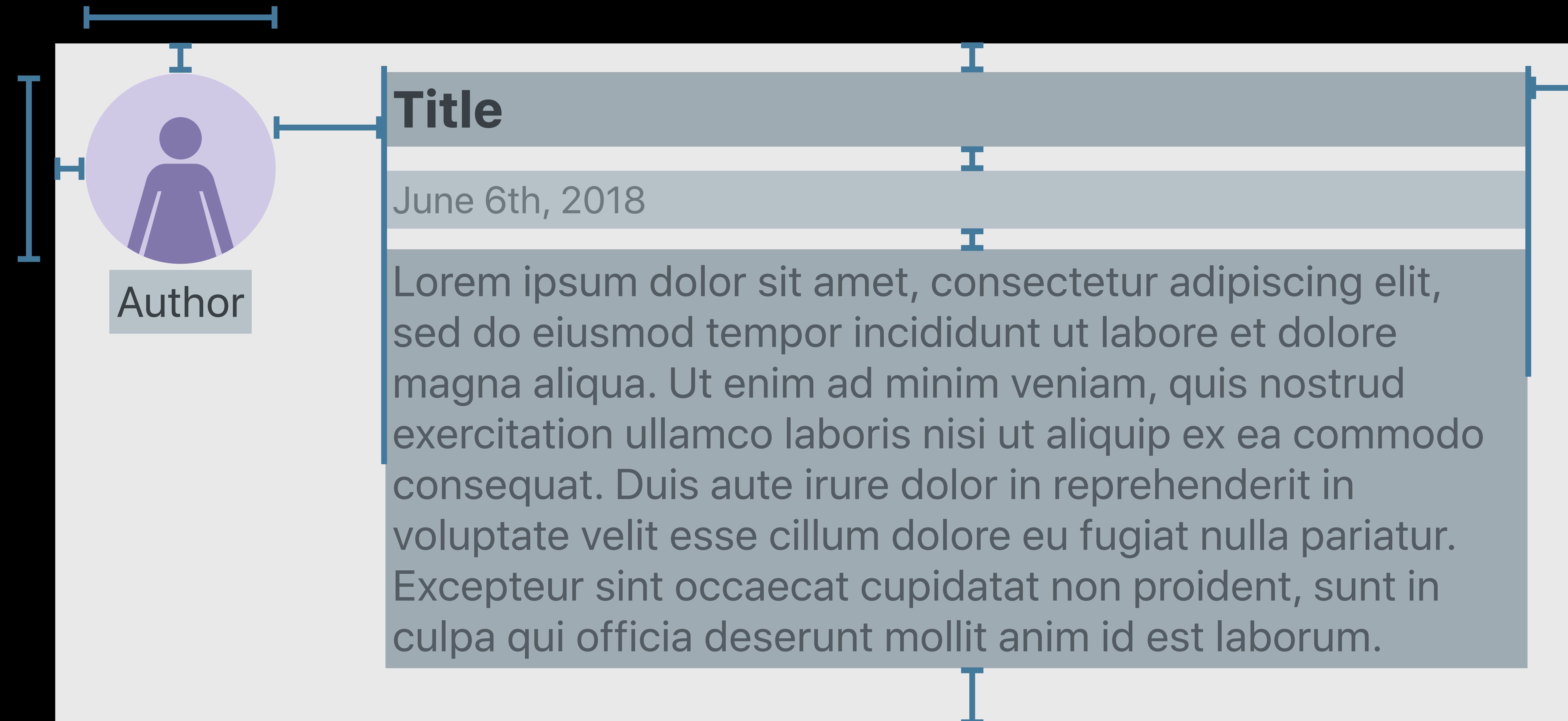
#WWDC18

High Performance Auto Layout

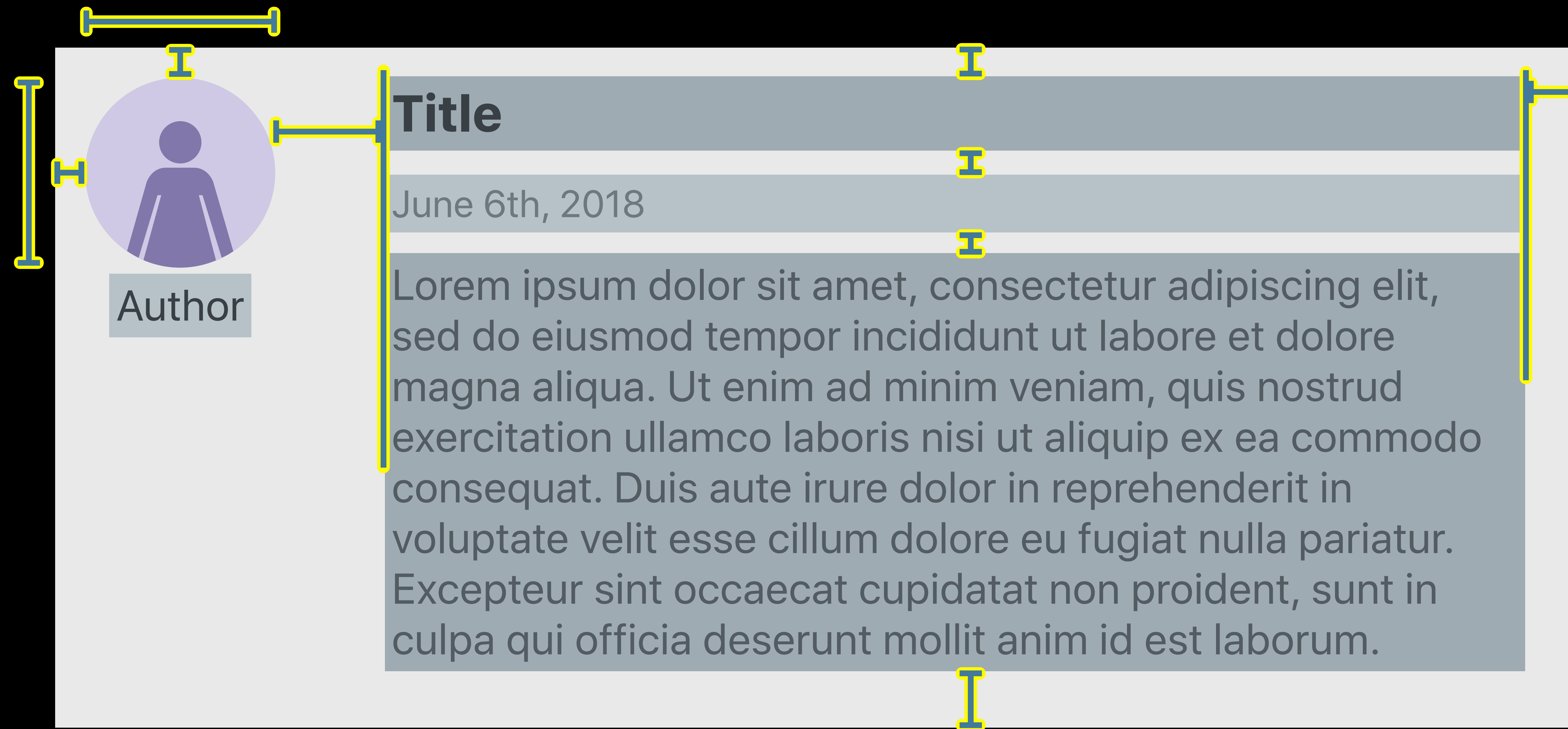
Ken Ferry, iOS System Experience

Kasia Wawer, iOS Keyboards

Auto Layout



Auto Layout



iOS 12 improvements

Internals and intuition

Building efficient layouts

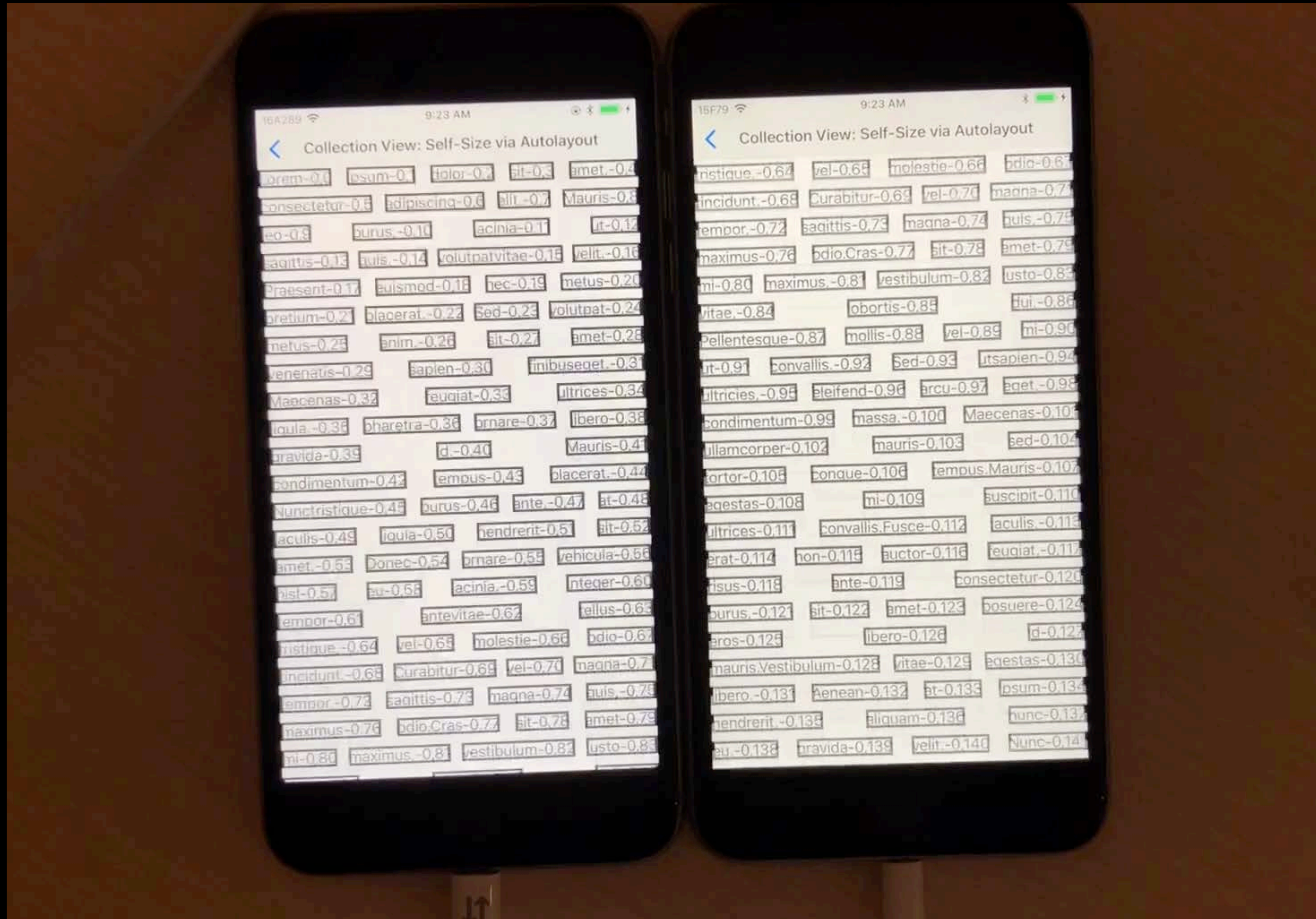
iOS 12 improvements

Internals and intuition

Building efficient layouts

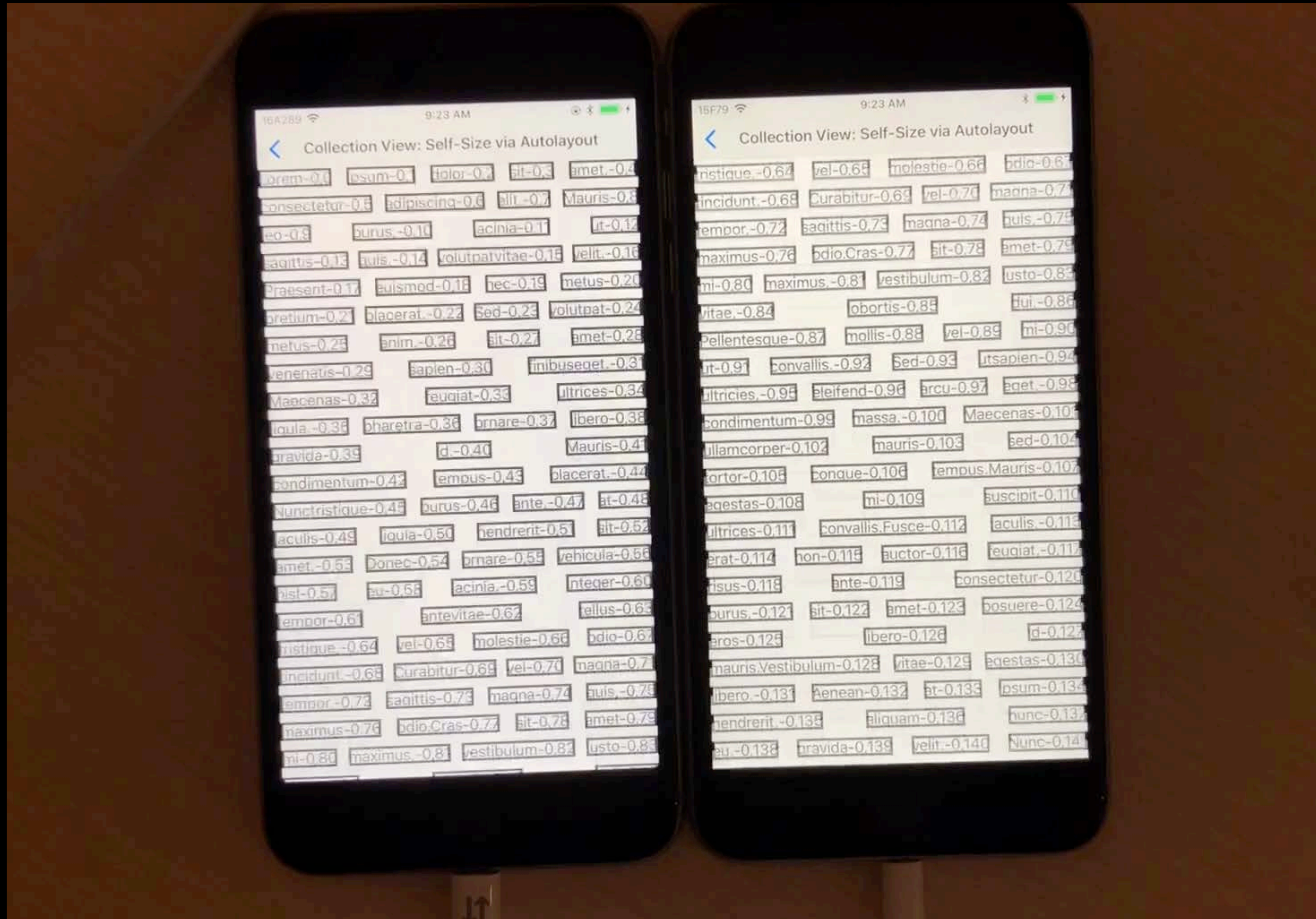
Performance Improvements in iOS 12

You get it for free



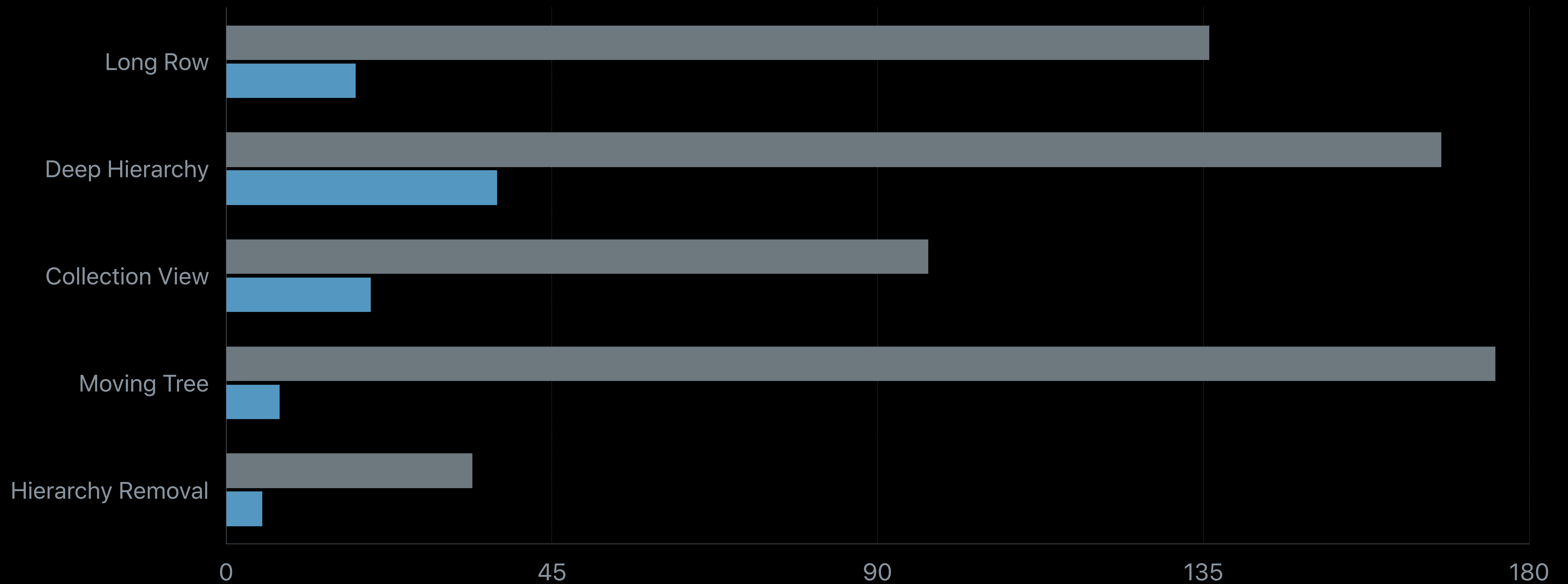
Performance Improvements in iOS 12

You get it for free



Performance Improvements in iOS 12

Shorter bars are better



iOS 12 improvements

Internals and intuition

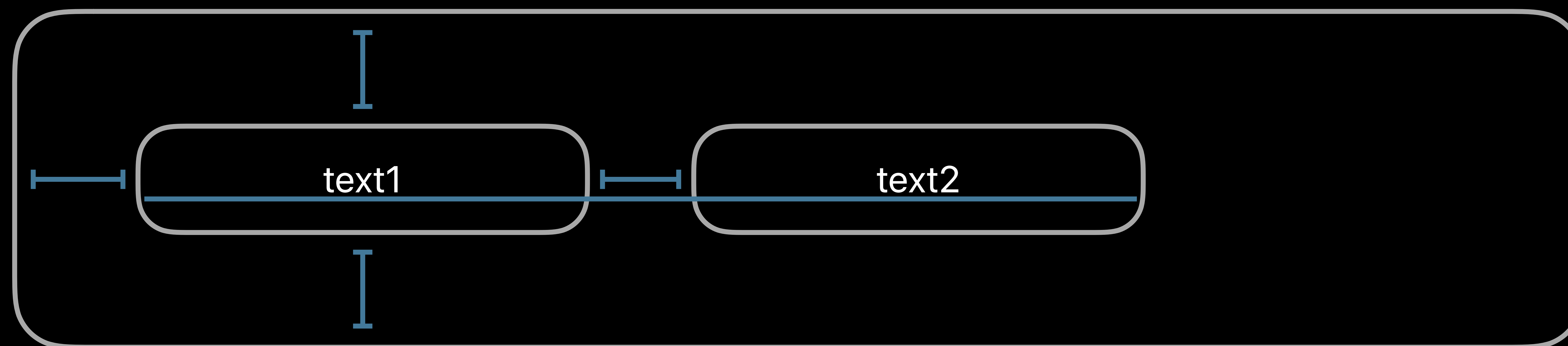
Building efficient layouts

iOS 12 improvements

Internals and intuition

Building efficient layouts

Simple Layout



// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline],
                                                metrics: nil,
                                                views: views)
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)

    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```


// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline],
                                                metrics: nil,
                                                views: views)

    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)

    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```


// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
```

```
    NSLayoutConstraint.deactivate(myConstraints)
```

```
    myConstraints.removeAll()
```

```
    let views = ["text1":text1, "text2":text2]
```

```
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",  
                                                options: [.alignAllFirstBaseline],  
                                                metrics: nil,  
                                                views: views)
```

```
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",  
                                                options: [],  
                                                metrics: nil,  
                                                views: views)
```

```
    NSLayoutConstraint.activate(myConstraints)
```

```
    super.updateConstraints()
```

```
}
```


// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
```

```
    NSLayoutConstraint.deactivate(myConstraints)
```

```
    myConstraints.removeAll()
```

```
    let views = ["text1":text1, "text2":text2]
```

```
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",  
                                                options: [.alignAllFirstBaseline],  
                                                metrics: nil,  
                                                views: views)
```

```
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",  
                                                options: [],  
                                                metrics: nil,  
                                                views: views)
```

```
    NSLayoutConstraint.activate(myConstraints)
```

```
    super.updateConstraints()
```

```
}
```


// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline],
                                                metrics: nil,
                                                views: views)
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)
    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```

// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline],
                                                metrics: nil,
                                                views: views)
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)

    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```


// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline],
                                                metrics: nil,
                                                views: views)

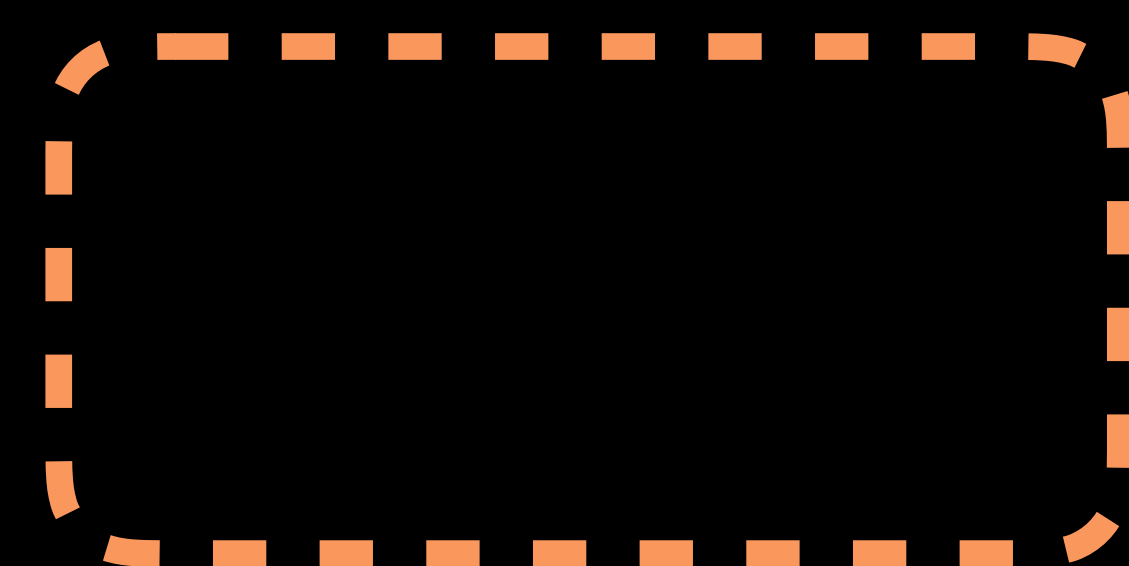
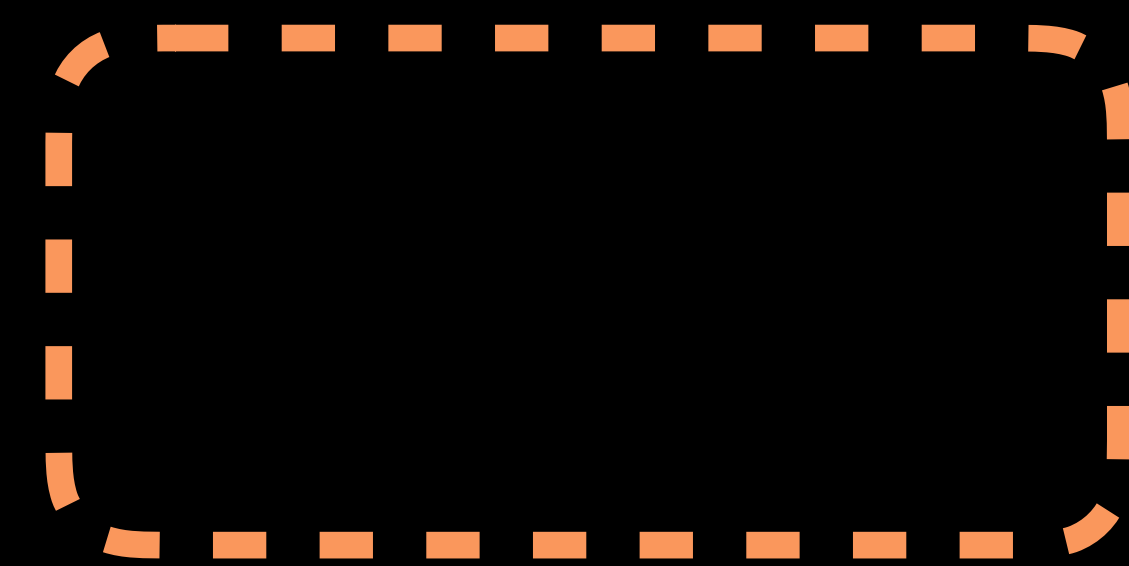
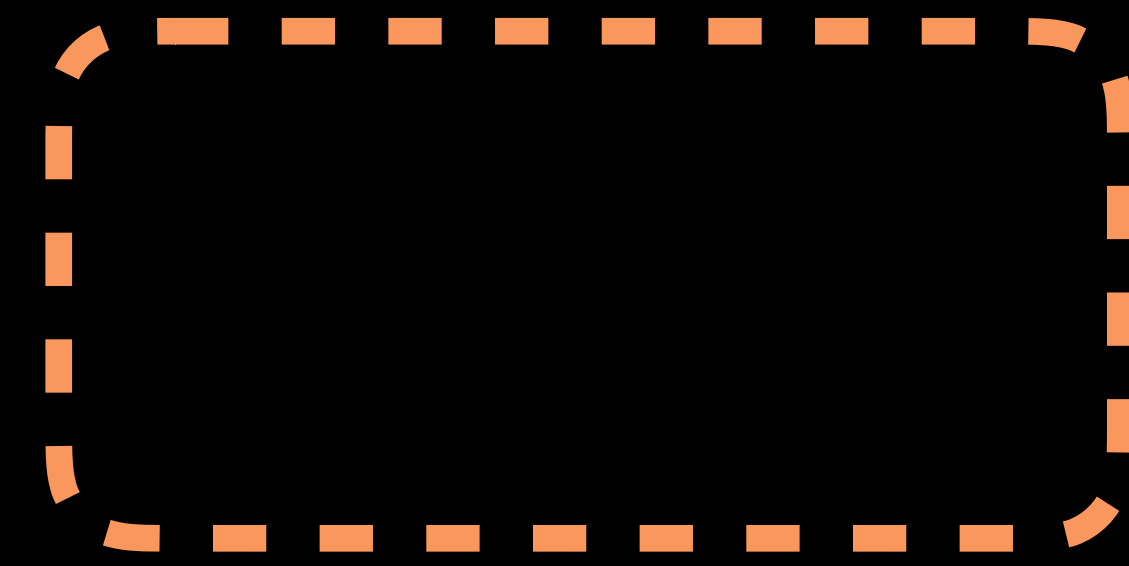
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)

    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```

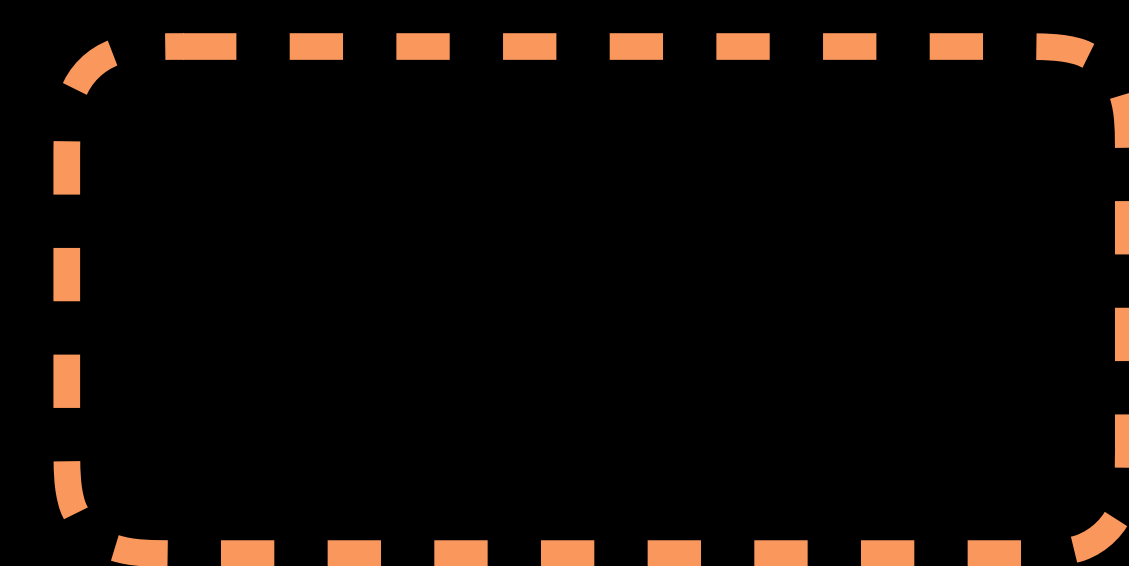
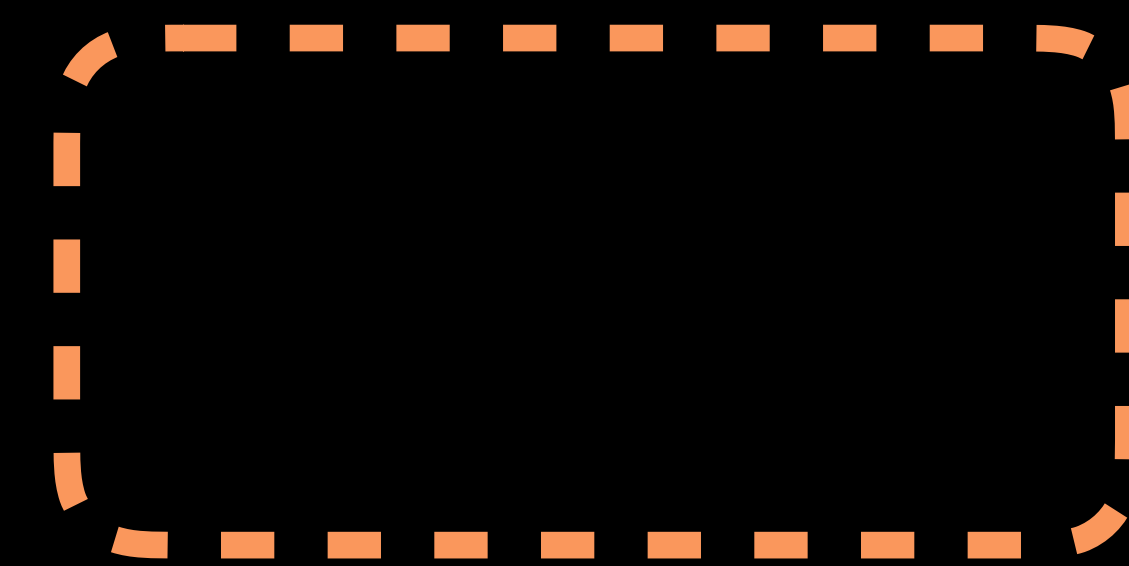
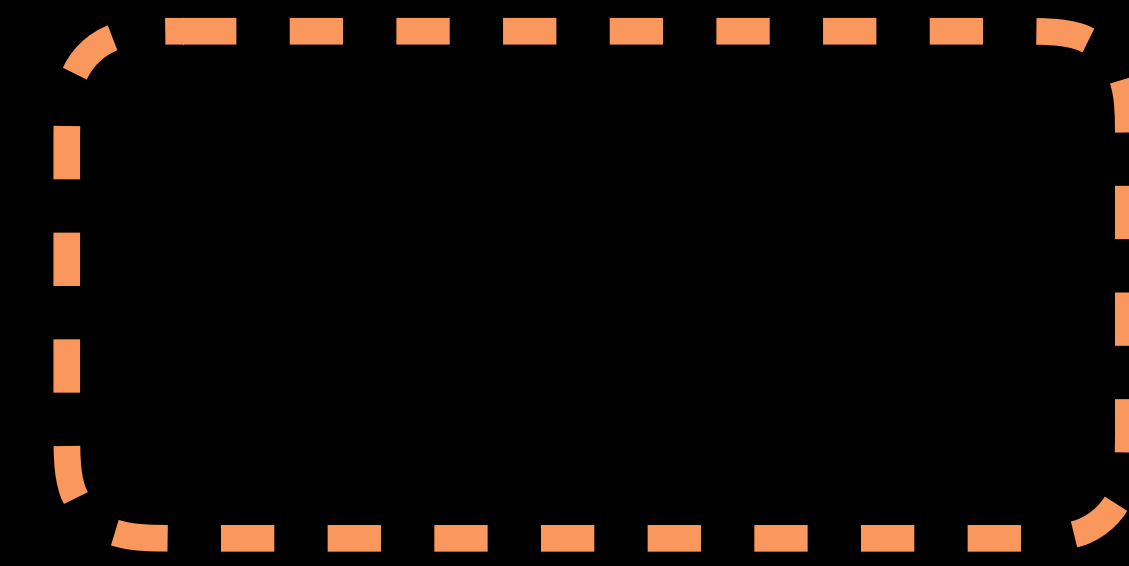
The Render Loop

What is updateConstraints?

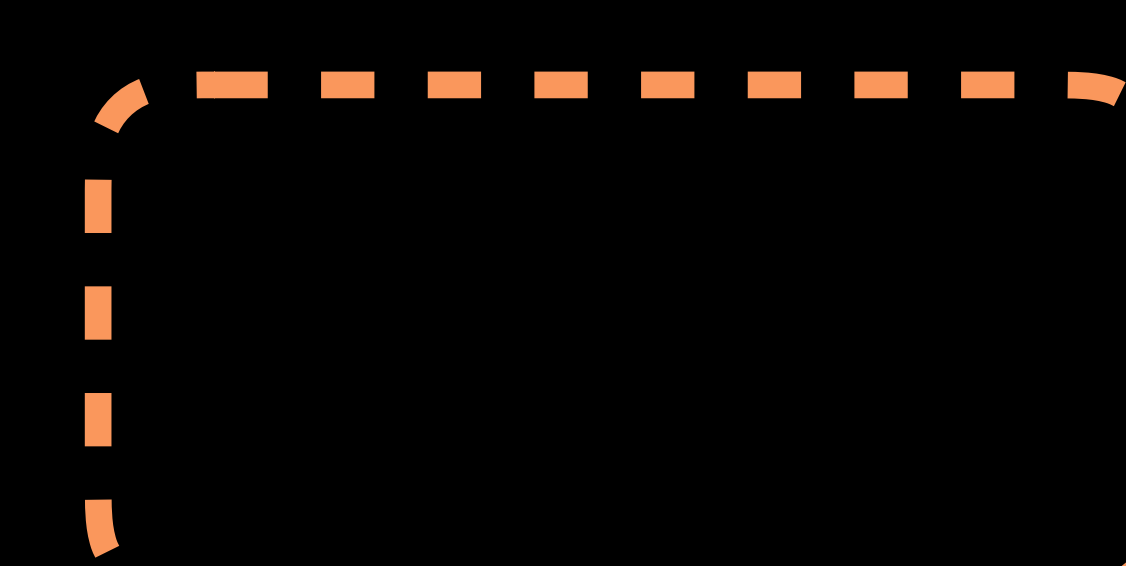
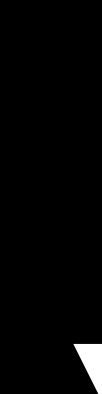
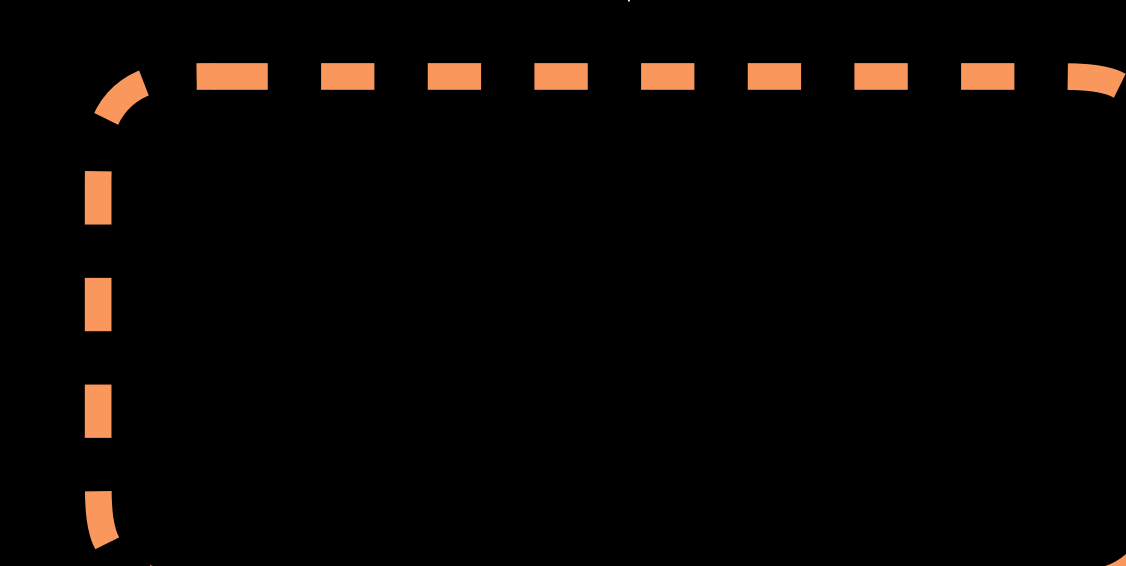
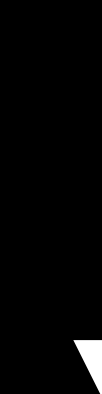
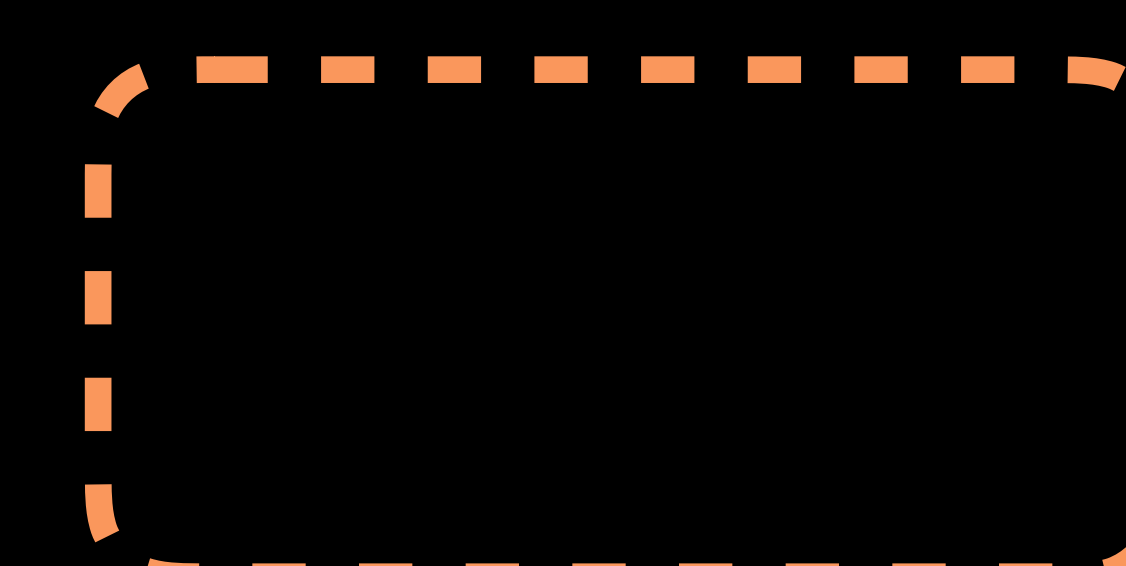
Update Constraints



Layout



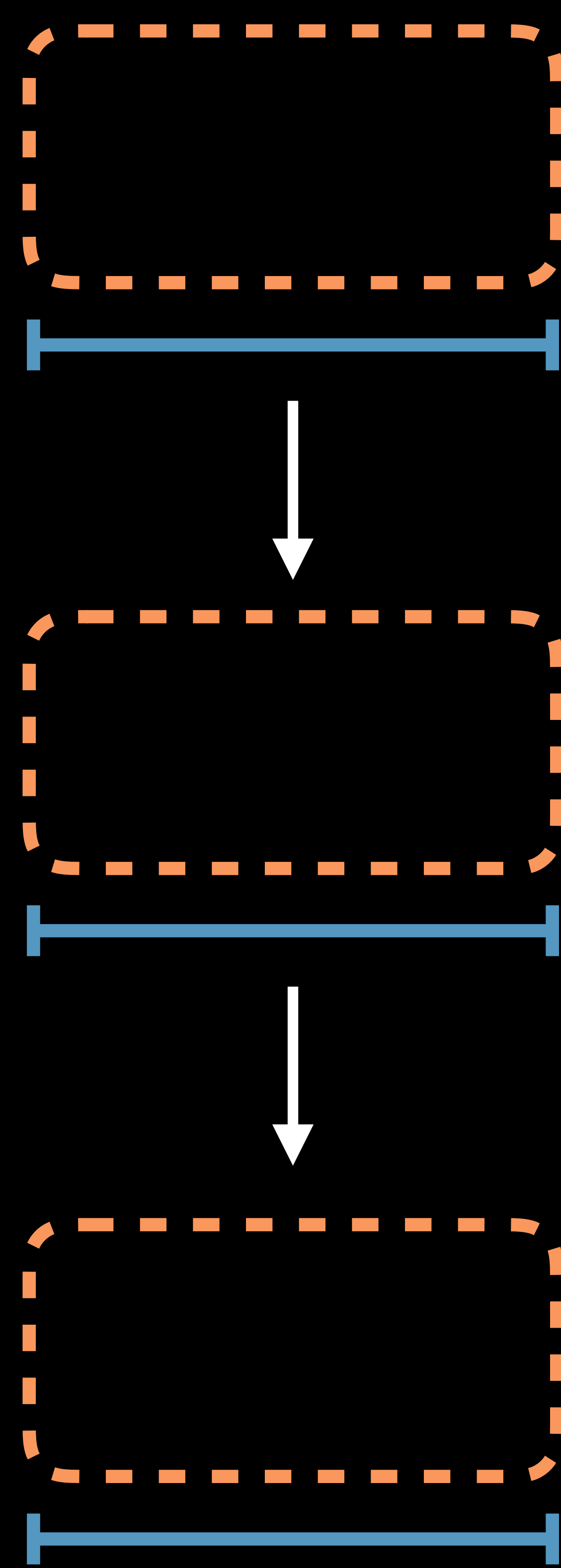
Display



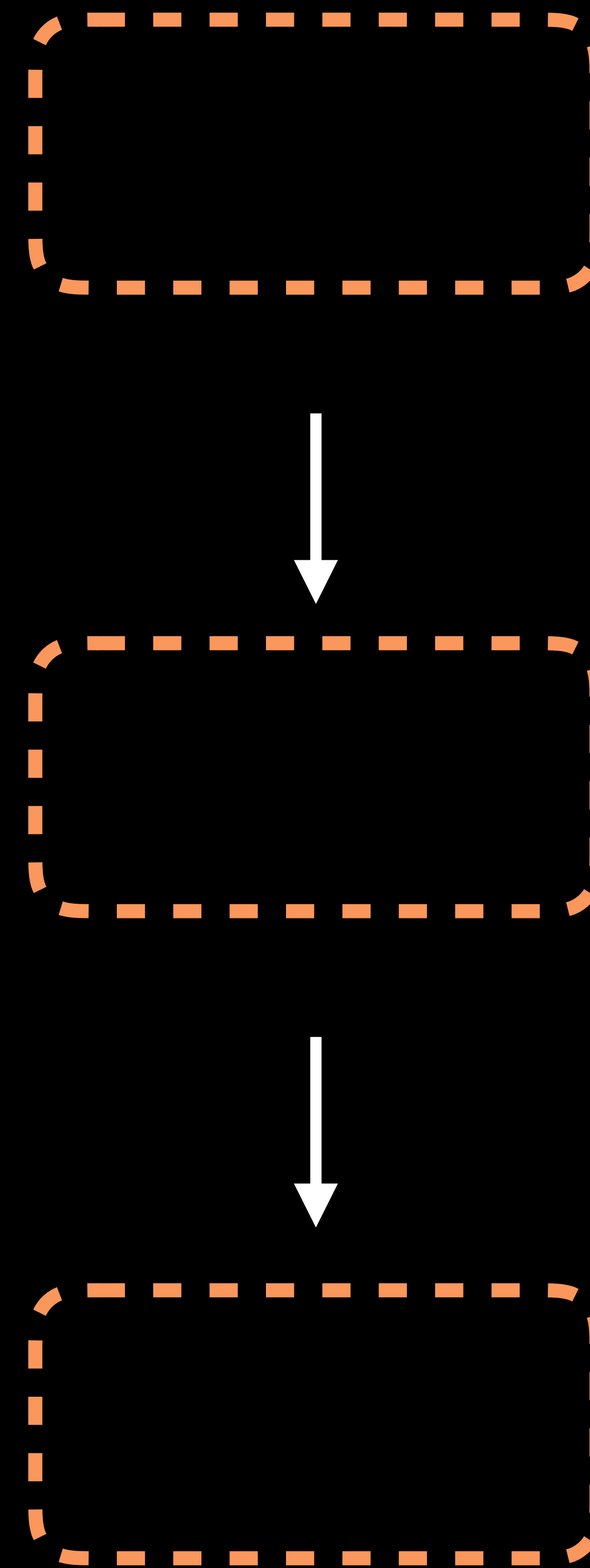
The Render Loop

What is updateConstraints?

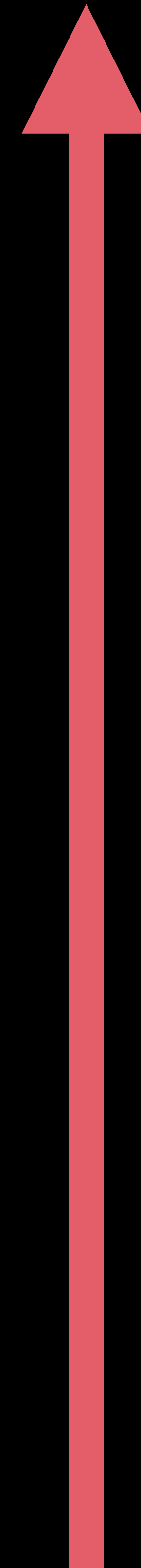
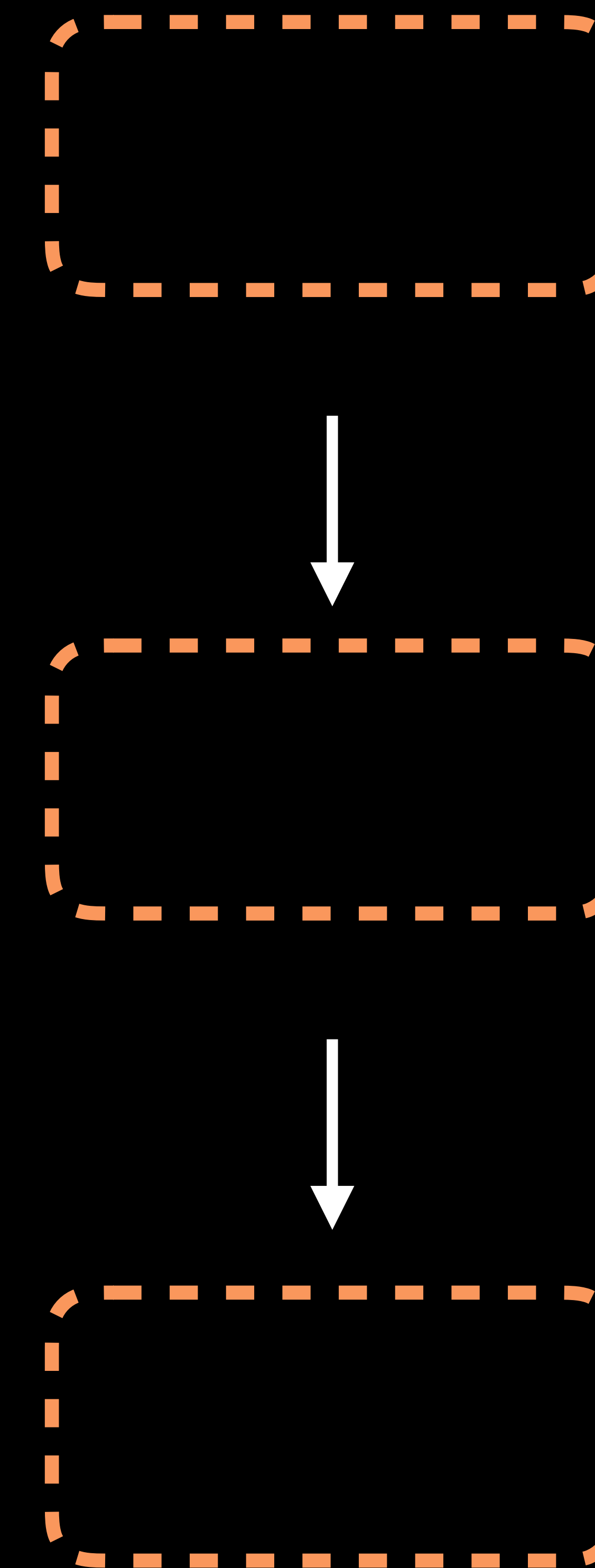
Update Constraints



Layout



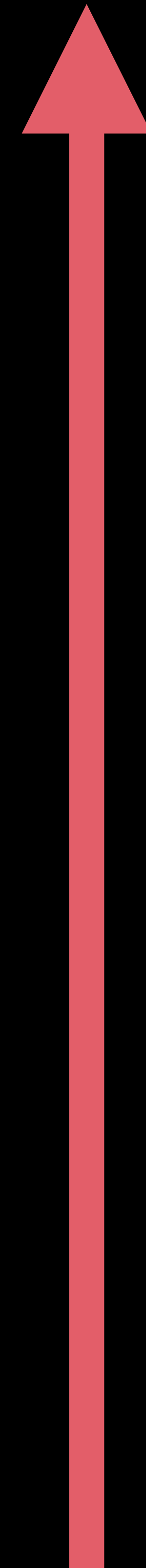
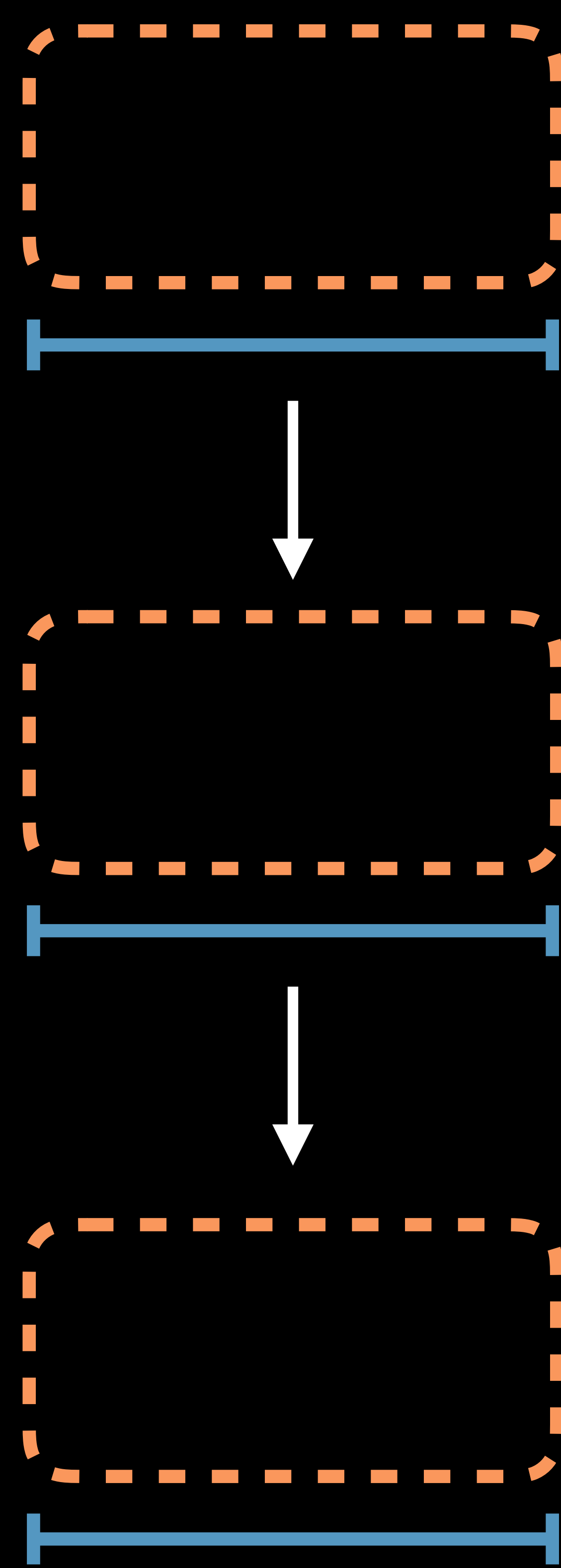
Display



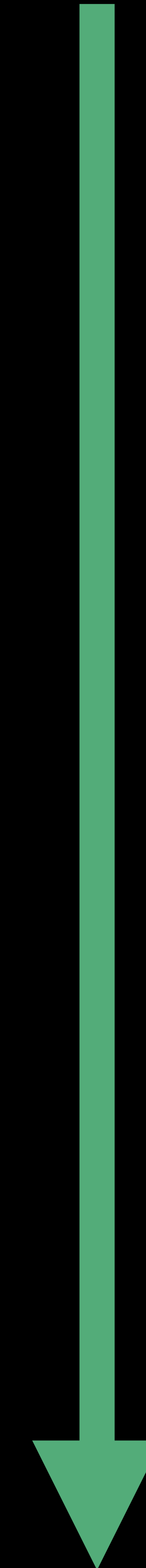
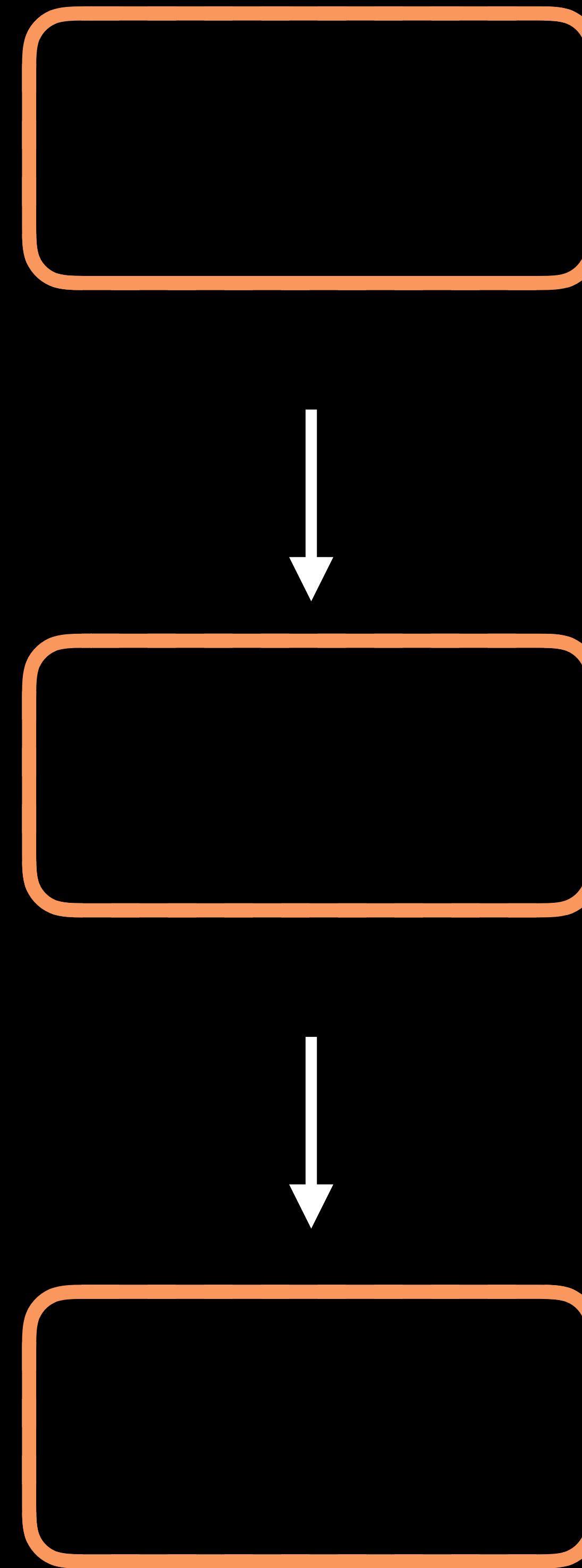
The Render Loop

What is updateConstraints?

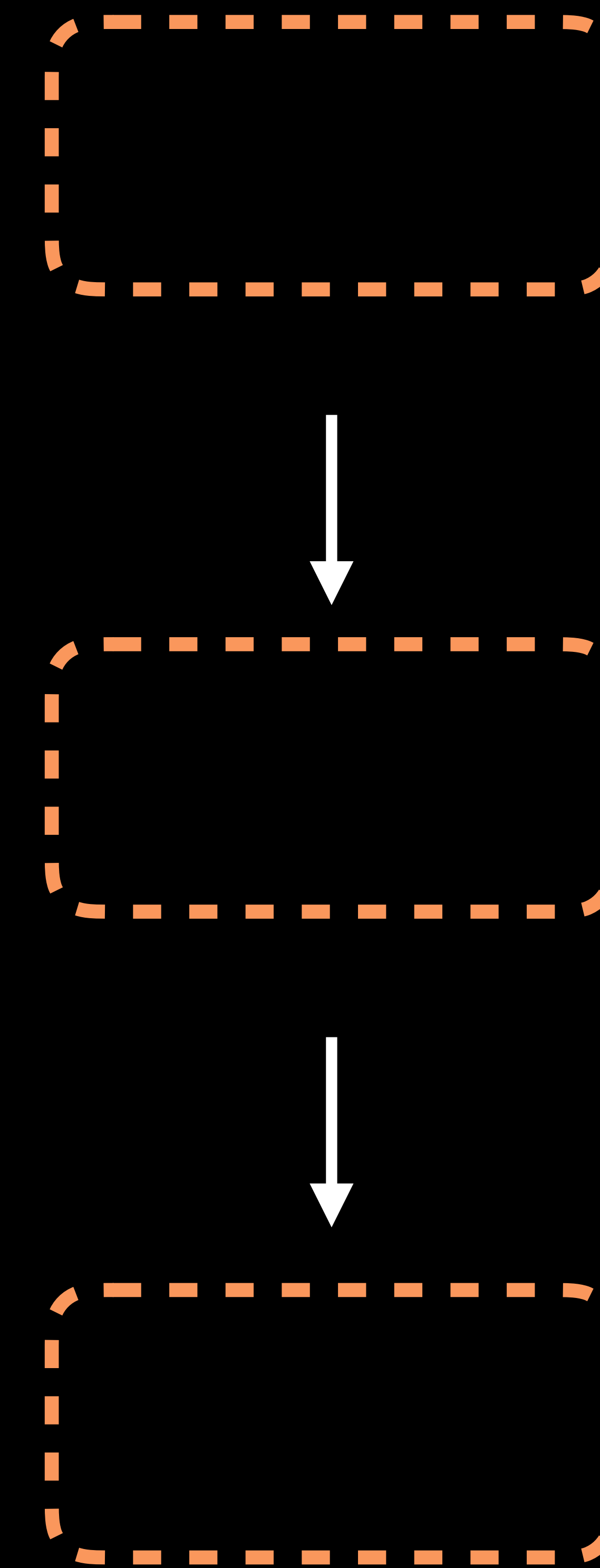
Update Constraints



Layout



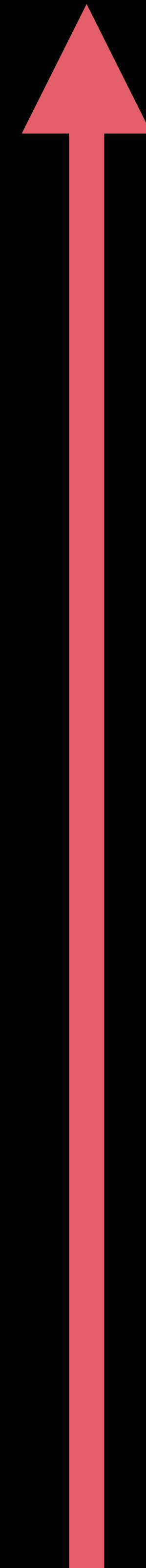
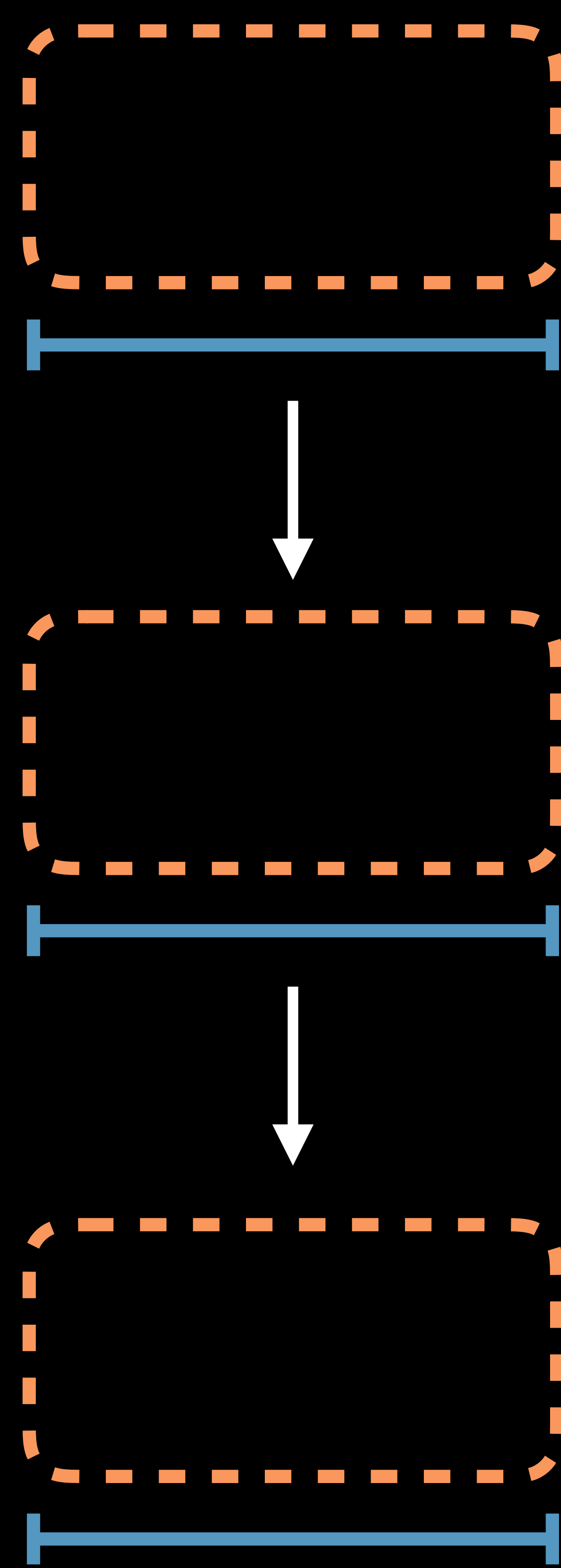
Display



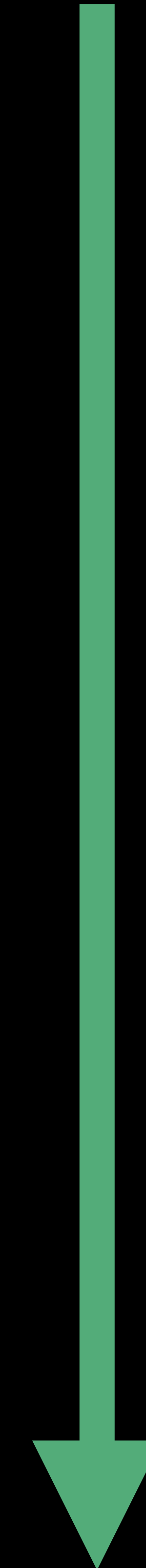
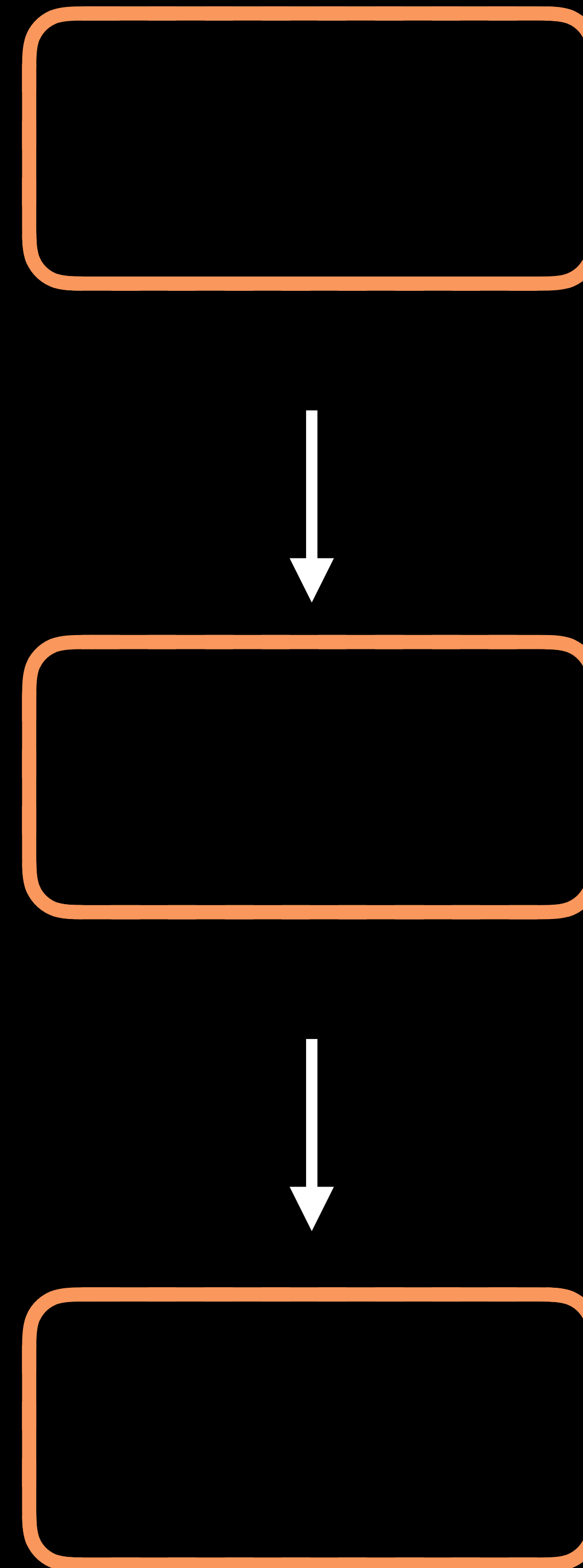
The Render Loop

What is updateConstraints?

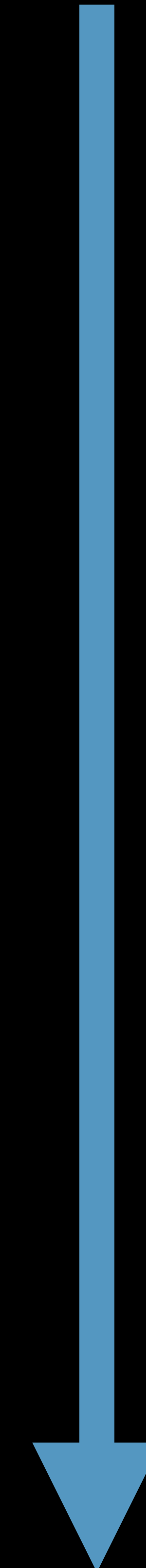
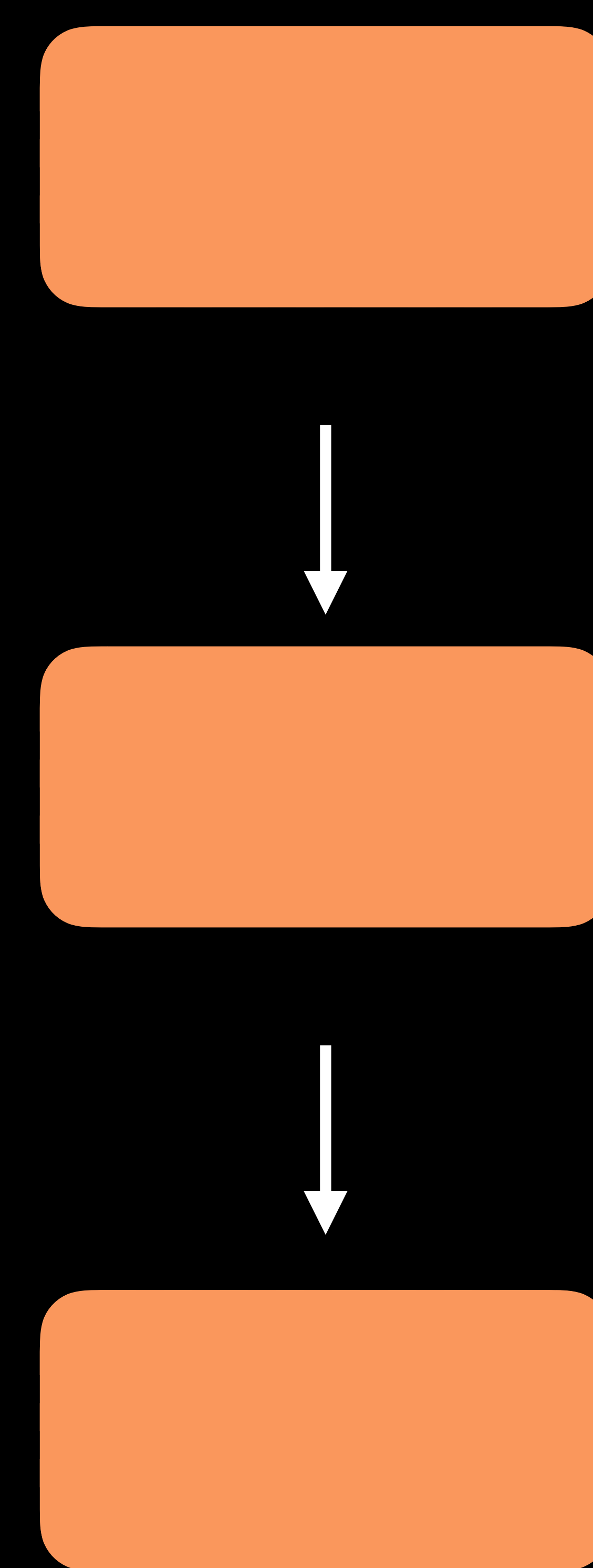
Update Constraints



Layout



Display



The Render Loop

What is updateConstraints?

Update Constraints	Layout	Display
<code>updateConstraints()</code>	<code>layoutSubviews()</code>	<code>draw(_:)</code>
<code>setNeedsUpdateConstraints()</code>	<code>setNeedsLayout()</code>	<code>setNeedsDisplay()</code>
<code>updateConstraintsIfNeeded()</code>	<code>layoutIfNeeded()</code>	-

// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline,]
                                                metrics: nil,
                                                views: views)
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)

    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```

// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline,]
                                                metrics: nil,
                                                views: views)
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)
    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```


// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func layoutSubviews() {
    text1.removeFromSuperview()
    text1 = nil
    text1 = UILabel(frame: CGRect(x: 20, y: 20, width: 300, height: 30))
    self.addSubview(text1)

    text2.removeFromSuperview()
    text2 = nil
    text2 = UILabel(frame: CGRect(x: 340, y: 20, width: 300, height: 30))
    self.addSubview(text2)
    super.layoutSubviews()
}
```

// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func layoutSubviews() {
```

```
    text1.removeFromSuperview()
```

```
    text1 = nil
```

```
    text1 = UILabel(frame: CGRect(x: 20, y: 20, width: 300, height: 30))
```

```
    self.addSubview(text1)
```

```
    text2.removeFromSuperview()
```

```
    text2 = nil
```

```
    text2 = UILabel(frame: CGRect(x: 340, y: 20, width: 300, height: 30))
```

```
    self.addSubview(text2)
```

```
    super.layoutSubviews()
```

```
}
```


// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline],
                                                metrics: nil,
                                                views: views)
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)

    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```

// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints+=NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline],
                                                metrics: nil,
                                                views: views)
    myConstraints+=NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)

    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```



```
// This is ok! Doesn't do anything unless self.myConstraints has been nil'd out
override func updateConstraints() {
    if self.myConstraints == nil {
        var constraints = [NSLayoutConstraint]()
        let views = ["text1":text1, "text2":text2]
        constraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                    options: [.alignAllFirstBaseline],
                                                    metrics: nil,
                                                    views: views)

        constraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                    options: [],
                                                    metrics: nil,
                                                    views: views)

        NSLayoutConstraint.activate(constraints)
        self.myConstraints = constraints
    }
    super.updateConstraints()
}
```





```
// This is ok! Doesn't do anything unless self.myConstraints has been nil'd out
override func updateConstraints() {
    if self.myConstraints == nil {
        var constraints = [NSLayoutConstraint]()
        let views = ["text1":text1, "text2":text2]
        constraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                    options: [.alignAllFirstBaseline],
                                                    metrics: nil,
                                                    views: views)

        constraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                    options: [],
                                                    metrics: nil,
                                                    views: views)

        NSLayoutConstraint.activate(constraints)
        self.myConstraints = constraints
    }
    super.updateConstraints()
}
```

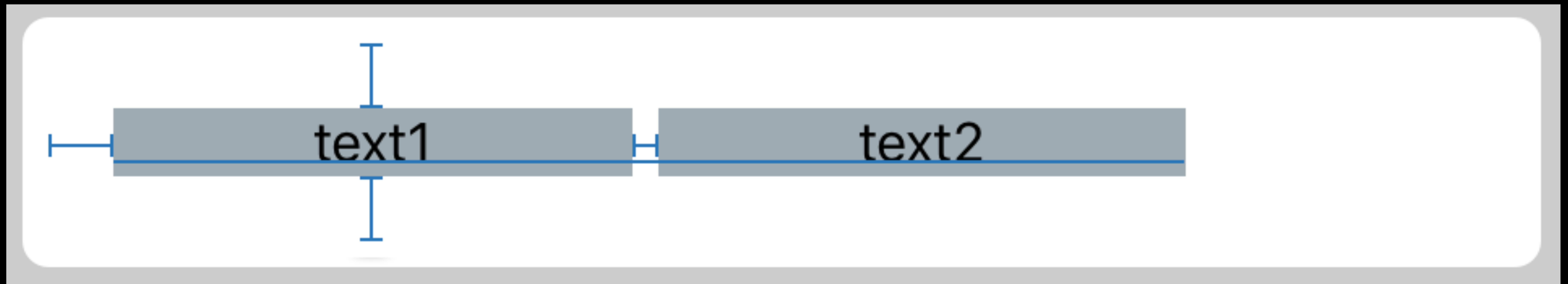

The Render Loop

Do you really need to be part of it?

Useful: Avoiding wasted work

Dangerous: Runs a lot

Interface Builder



// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline],
                                                metrics: nil,
                                                views: views)
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)

    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```

// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline],
                                                metrics: nil,
                                                views: views)
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)
    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```

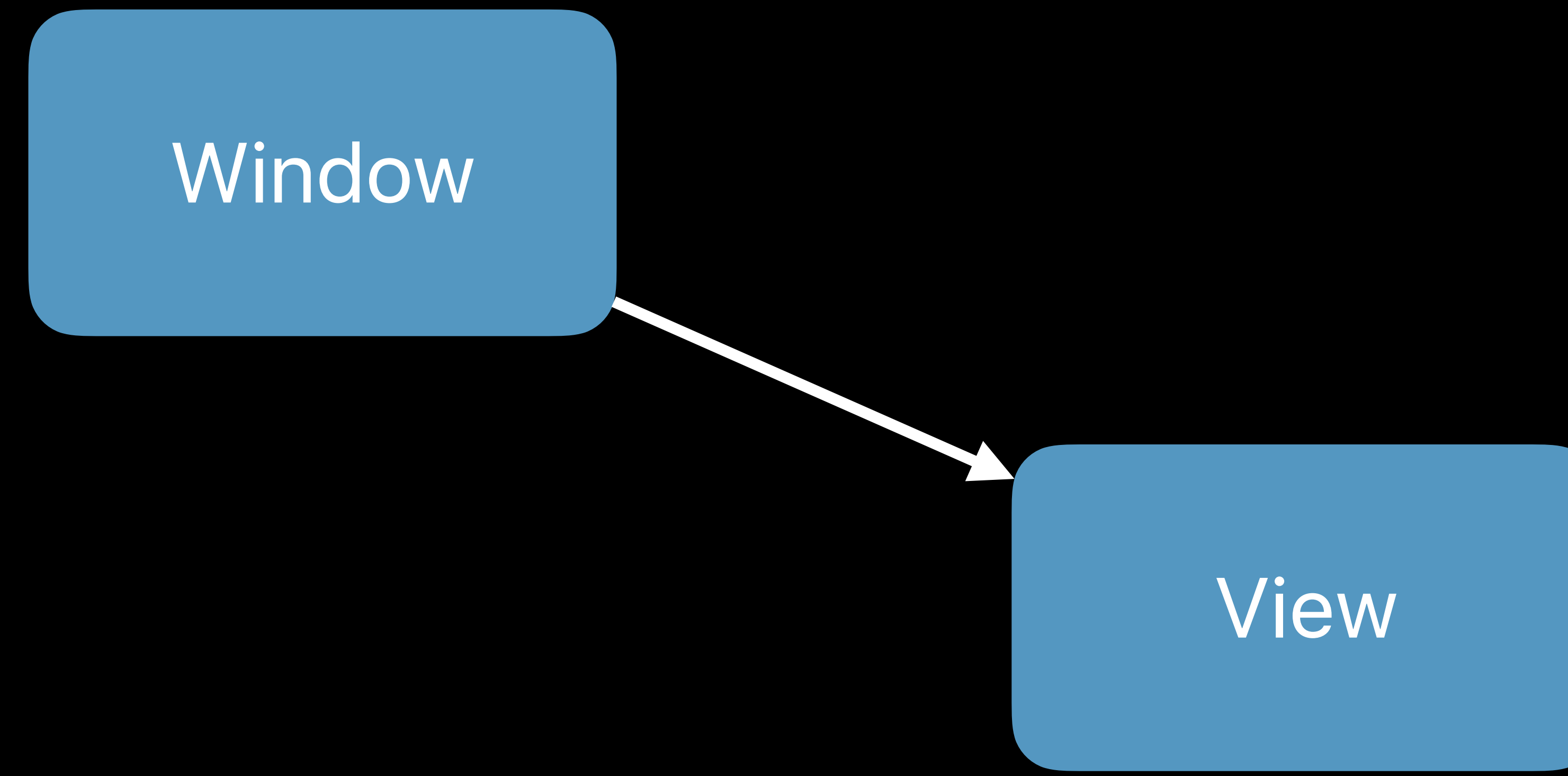
Activating a Constraint

Internal structure

View

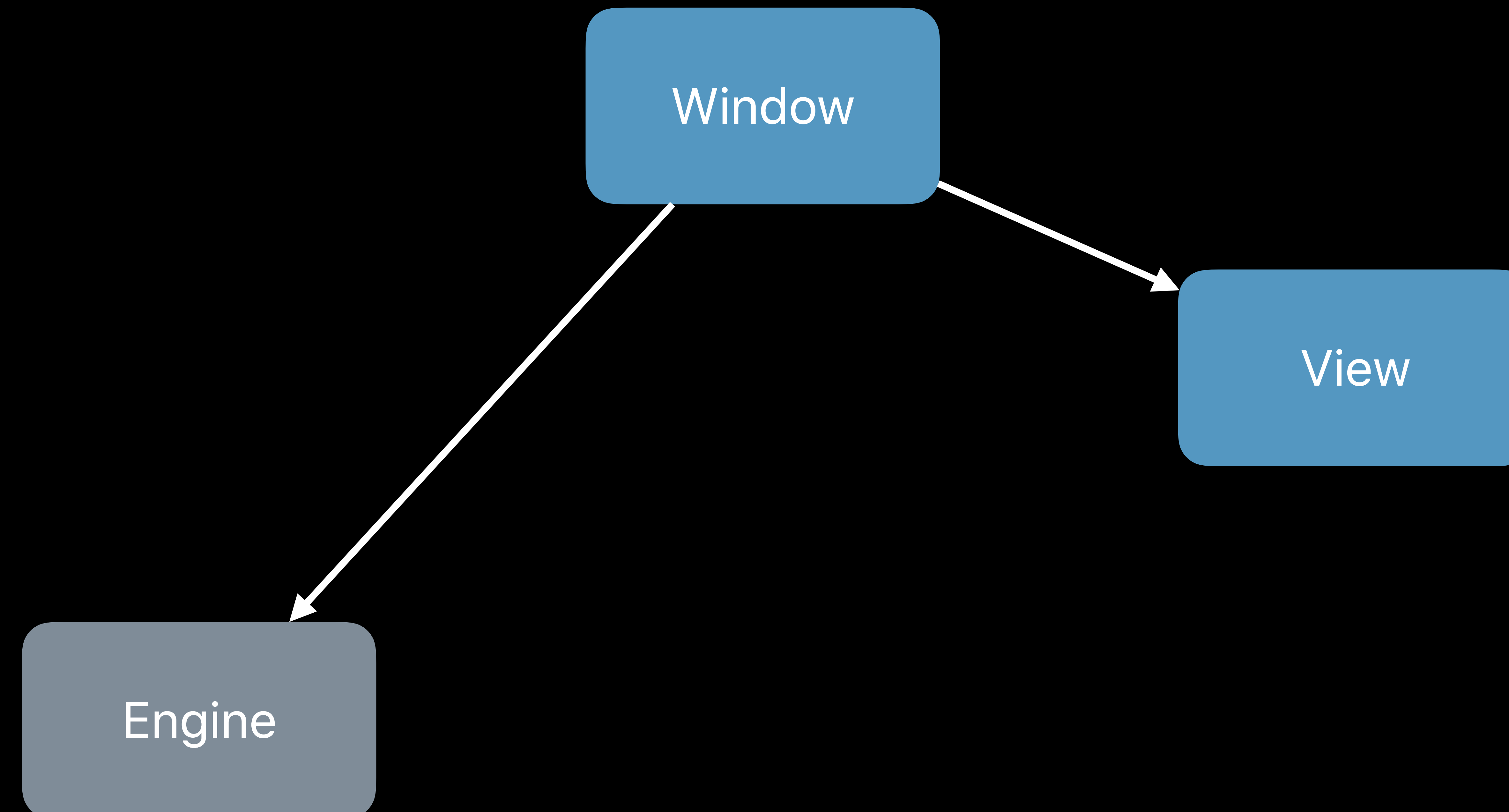
Activating a Constraint

Internal structure



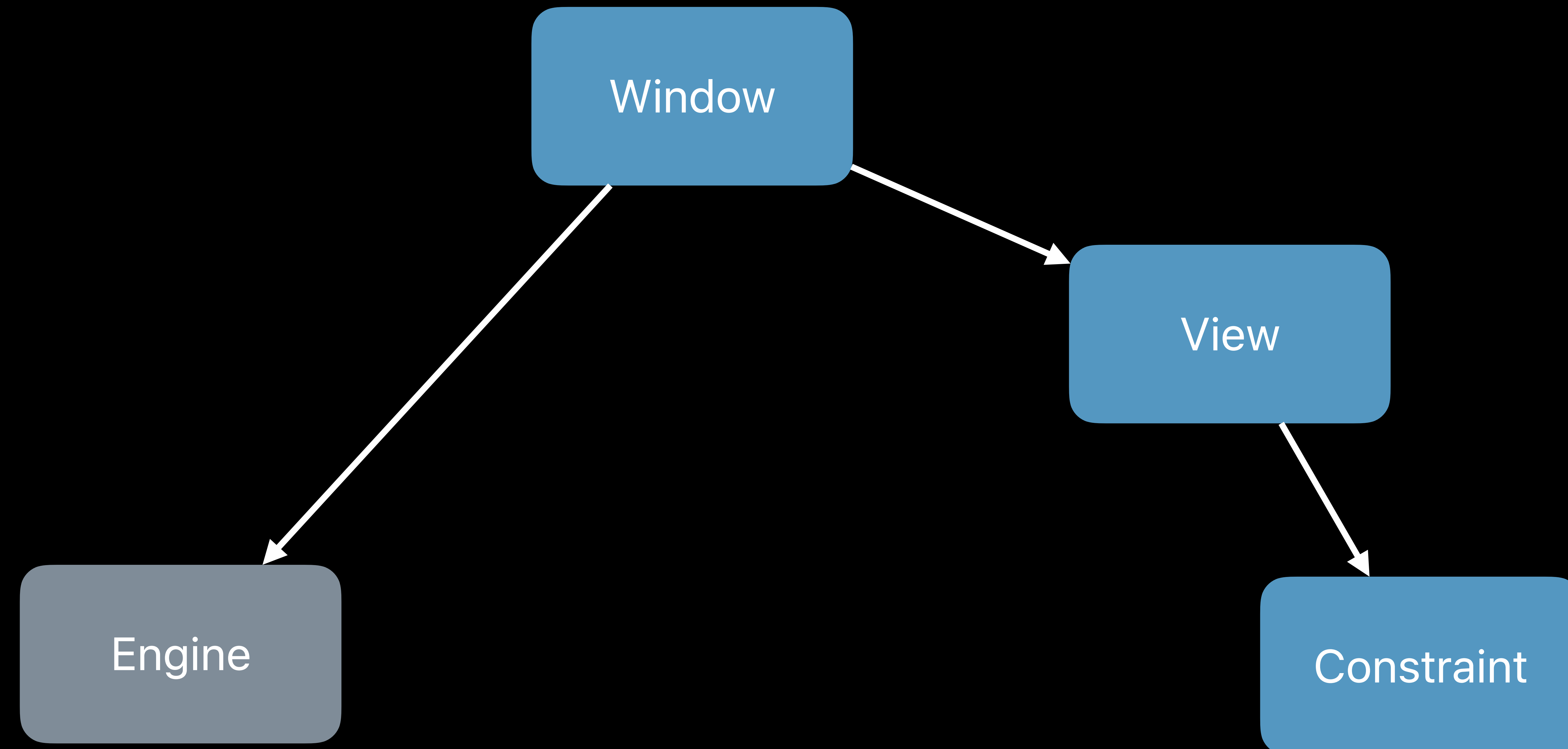
Activating a Constraint

Internal structure



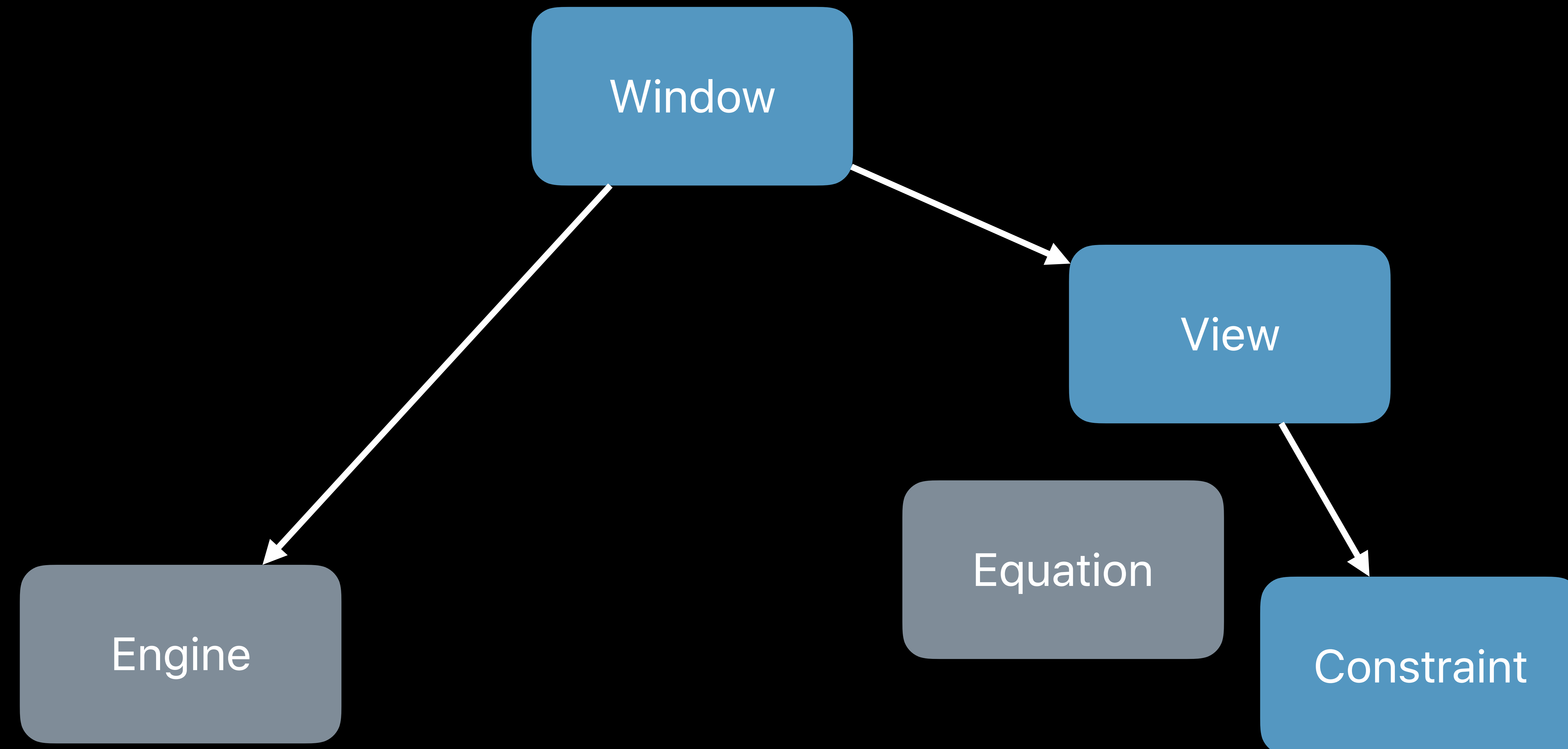
Activating a Constraint

Internal structure



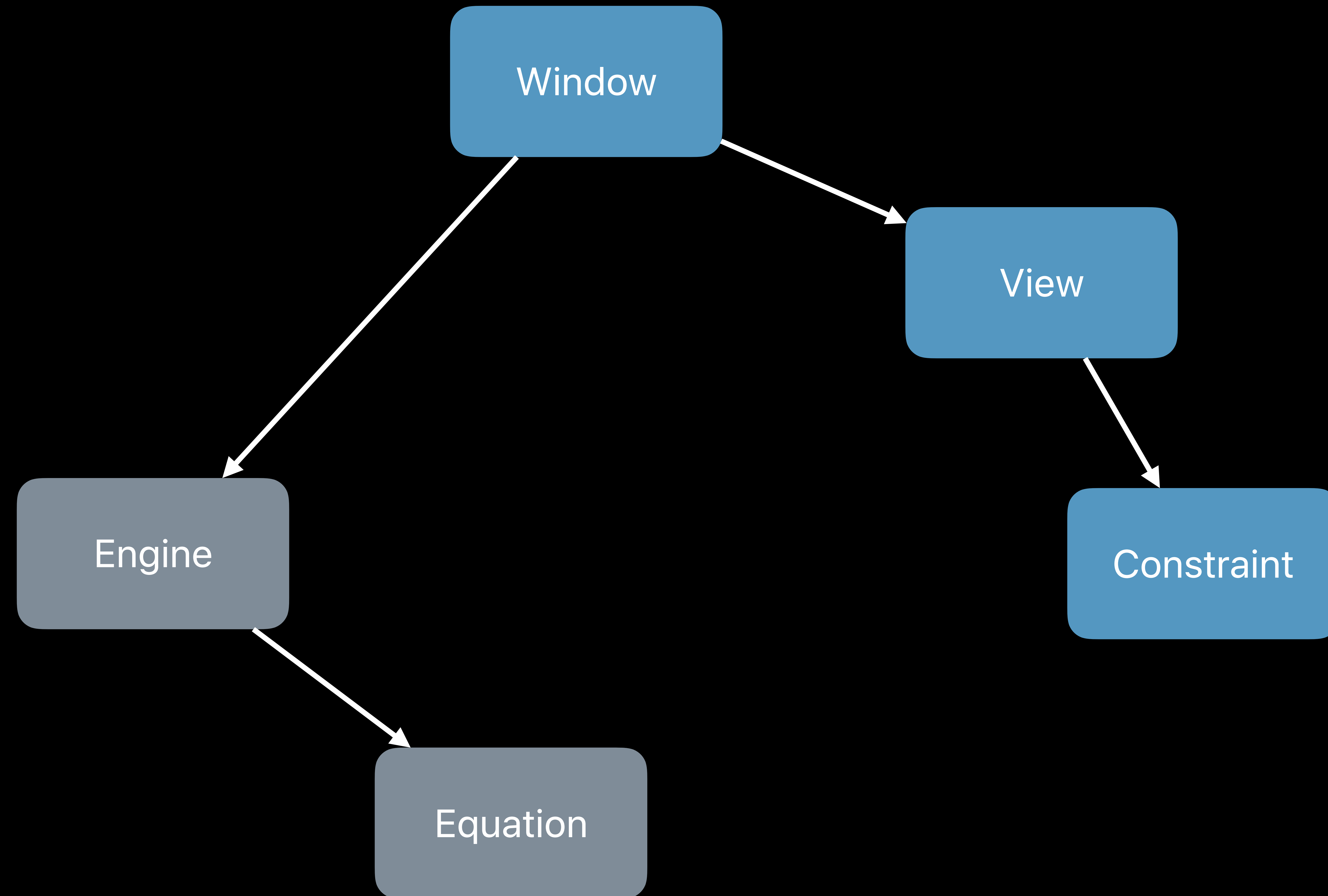
Activating a Constraint

Internal structure



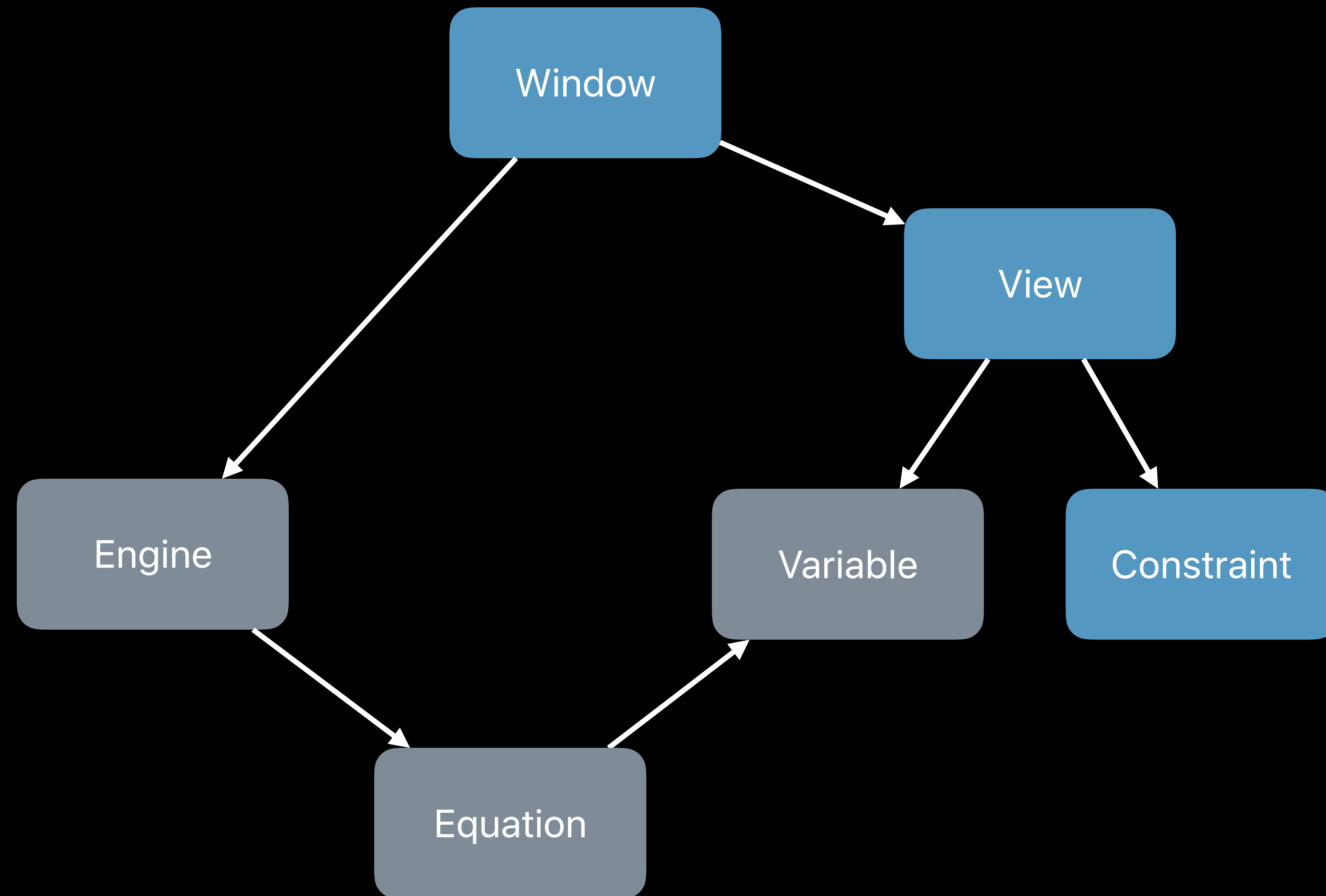
Activating a Constraint

Internal structure



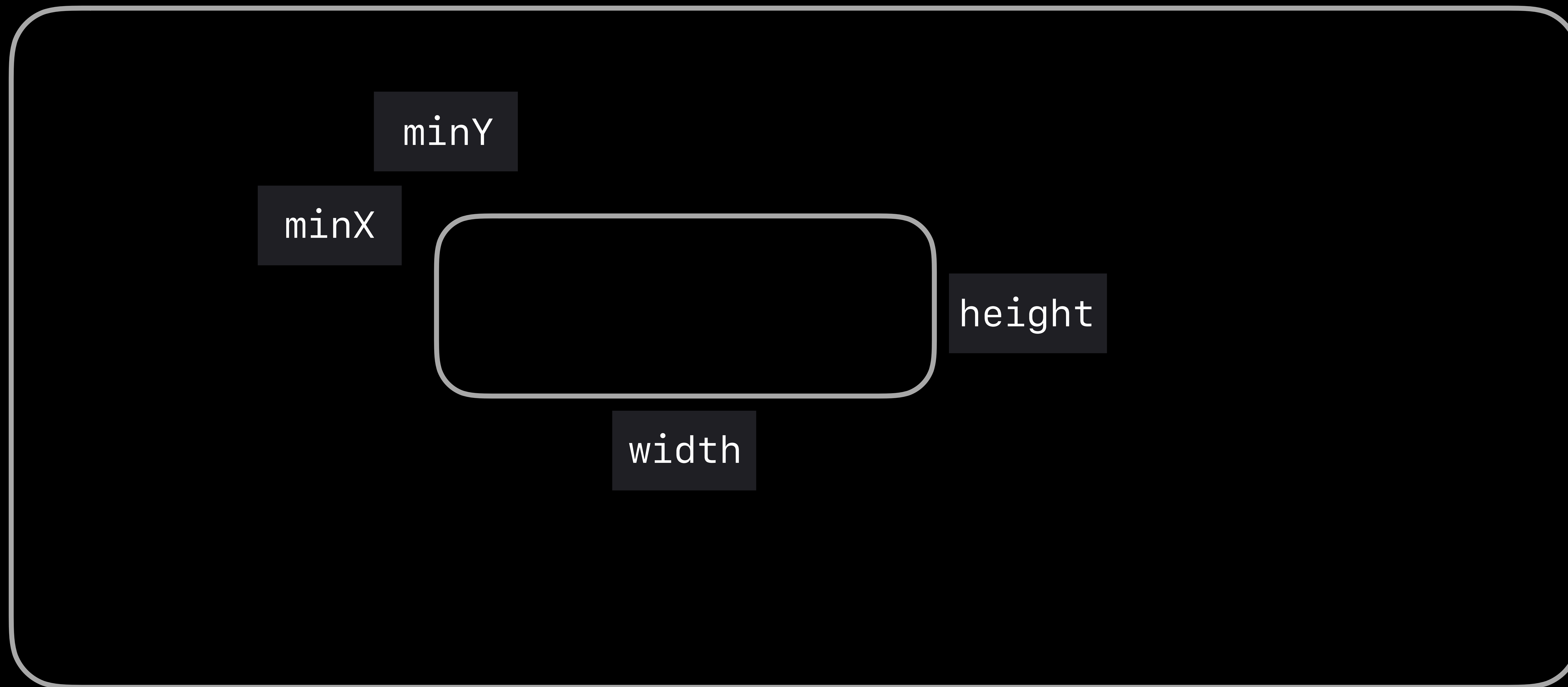
Activating a Constraint

Internal structure

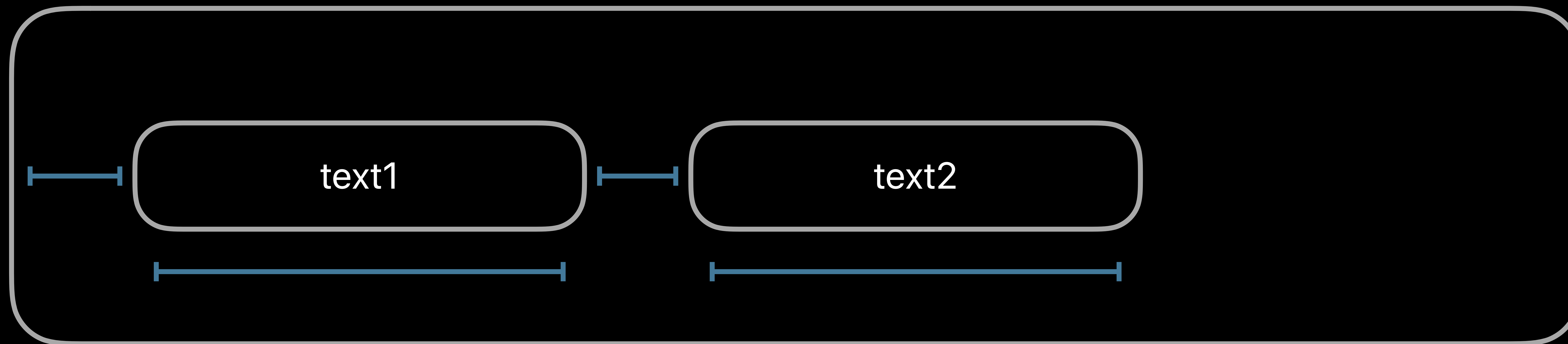


Variables

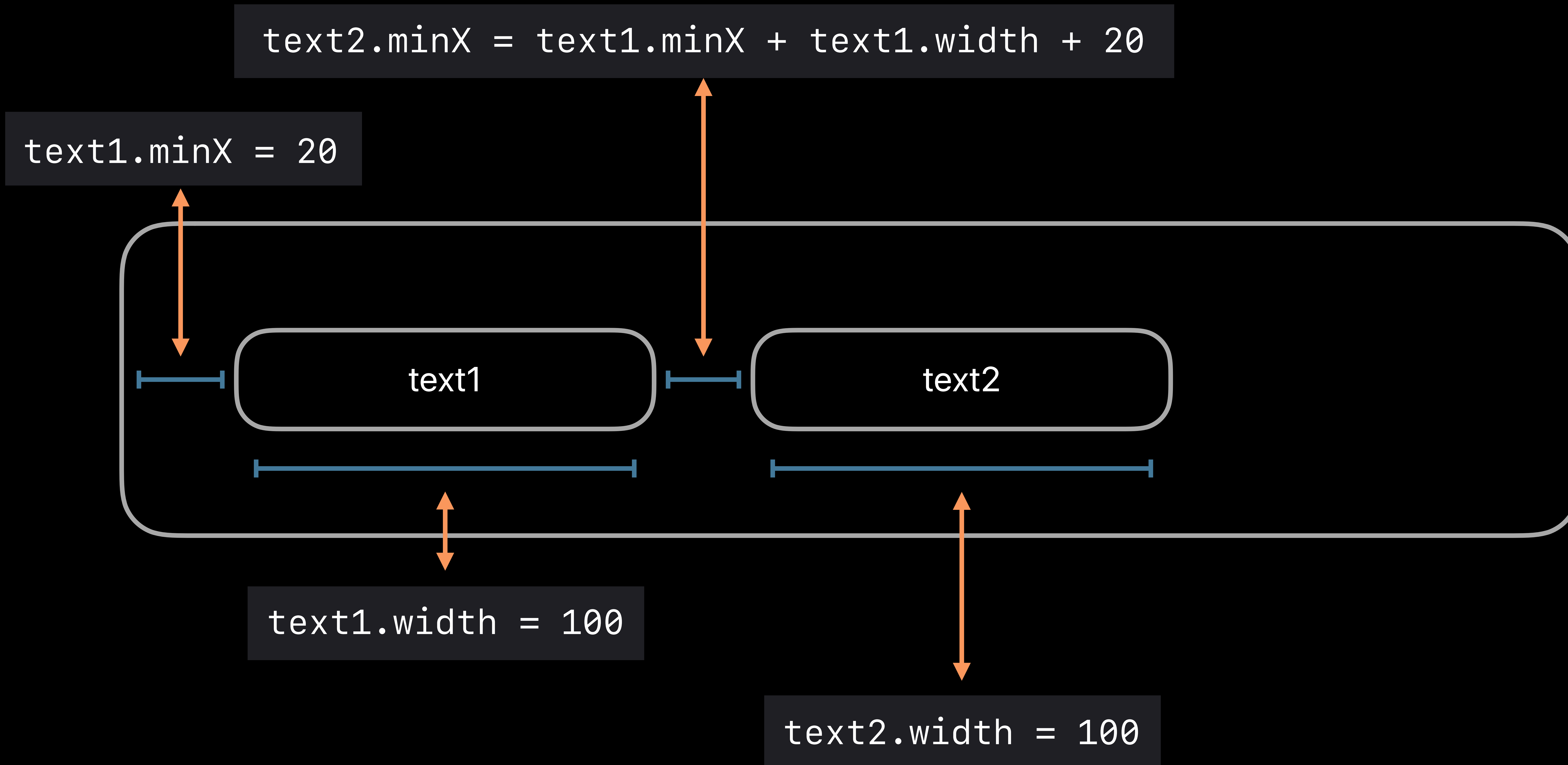
Unknowns layout must calculate



Activating a Constraint

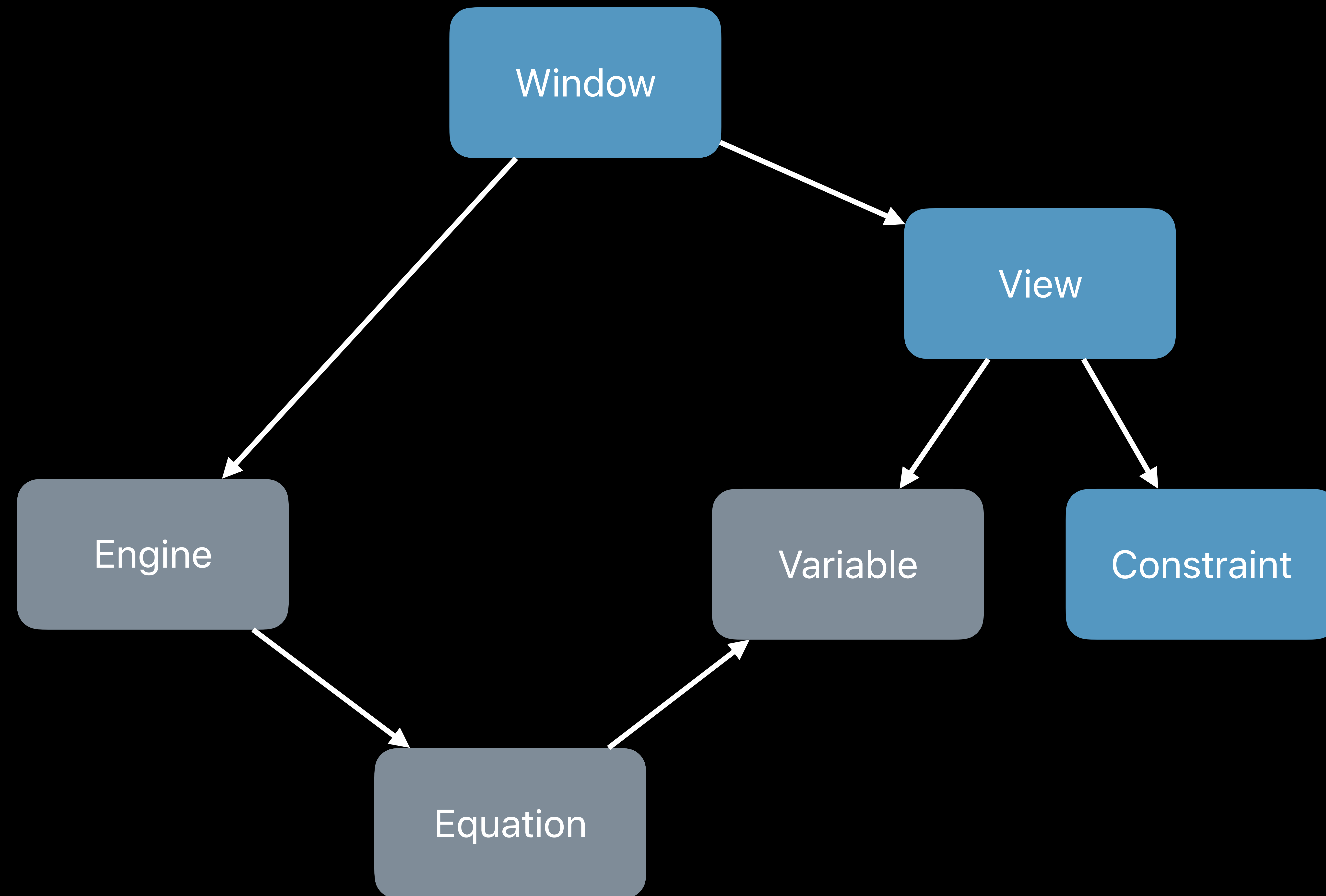


Activating a Constraint

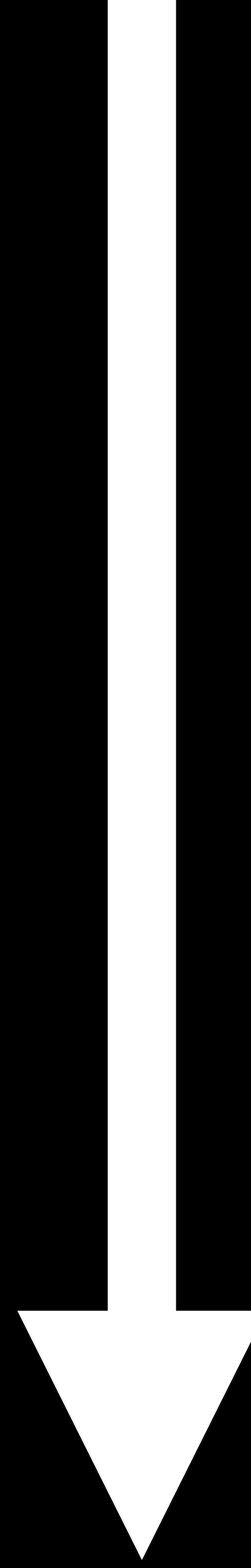


Activating a Constraint

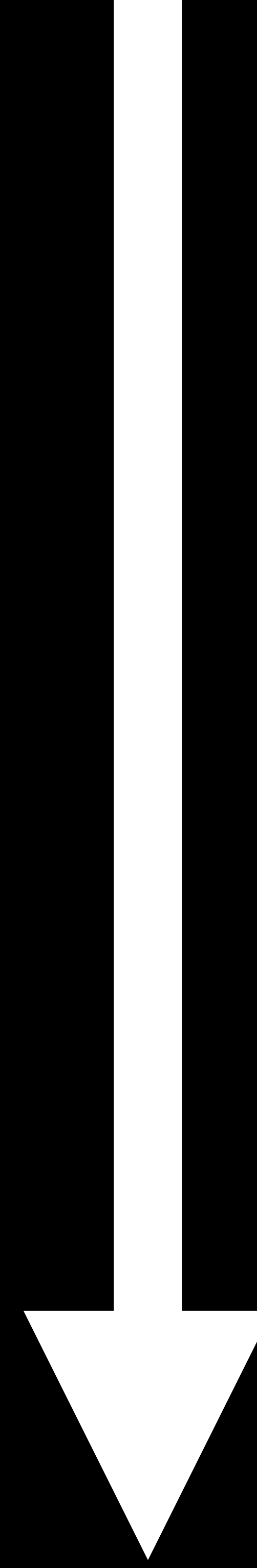
Internal structure



Engine

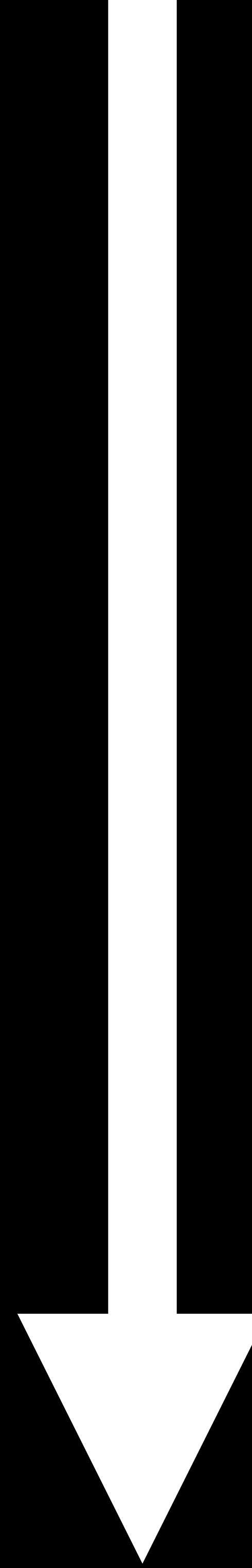


Engine



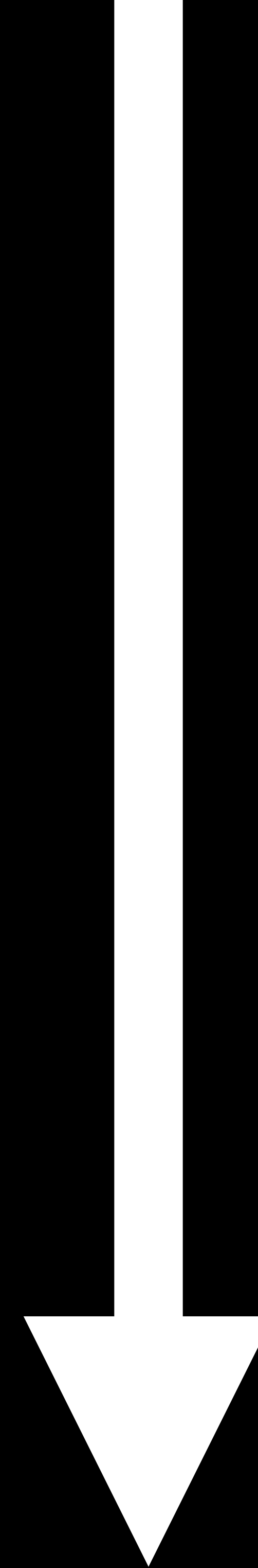
```
text1.minX = 8
```


Engine



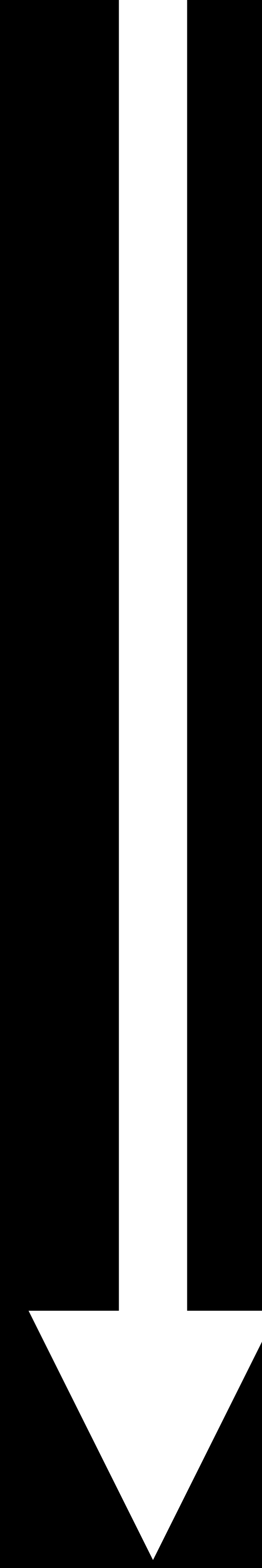
```
text1.minX = 8  
text1.width = 100
```

Engine



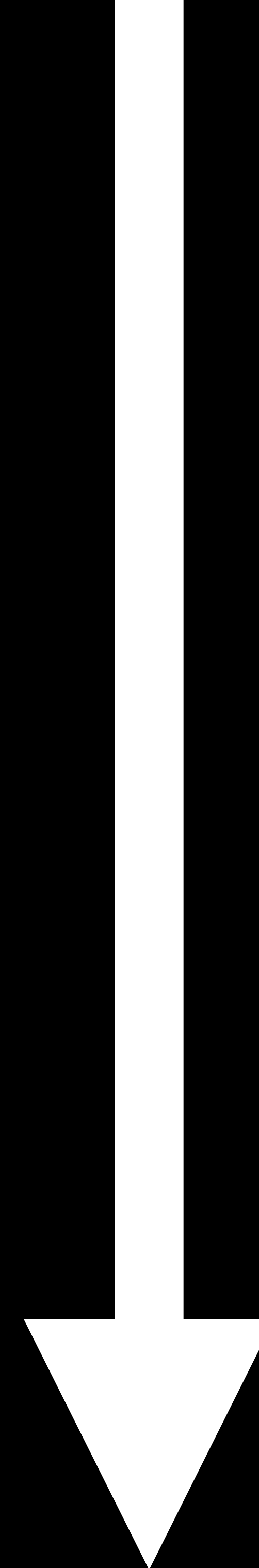
```
text1.minX = 8  
text1.width = 100  
text2.minX = text1.minX + text1.width + 20
```

Engine



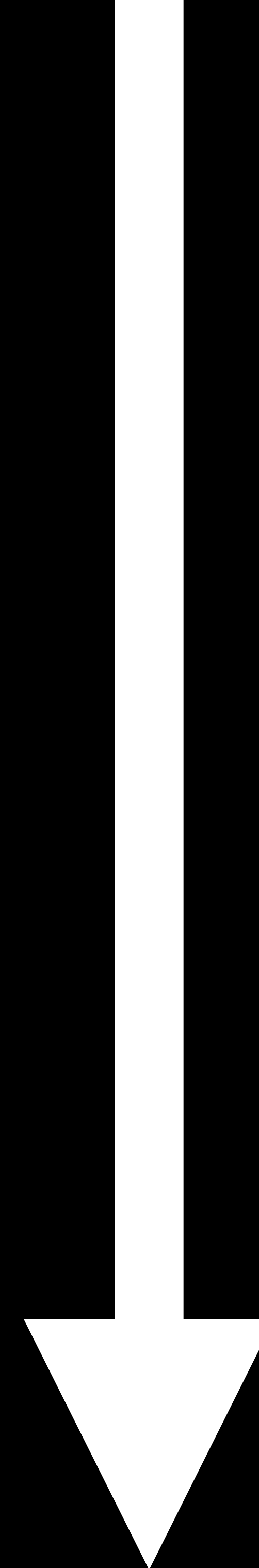
```
text1.minX = 8  
text1.width = 100  
text2.minX = text1.minX + text1.width + 20
```

Engine



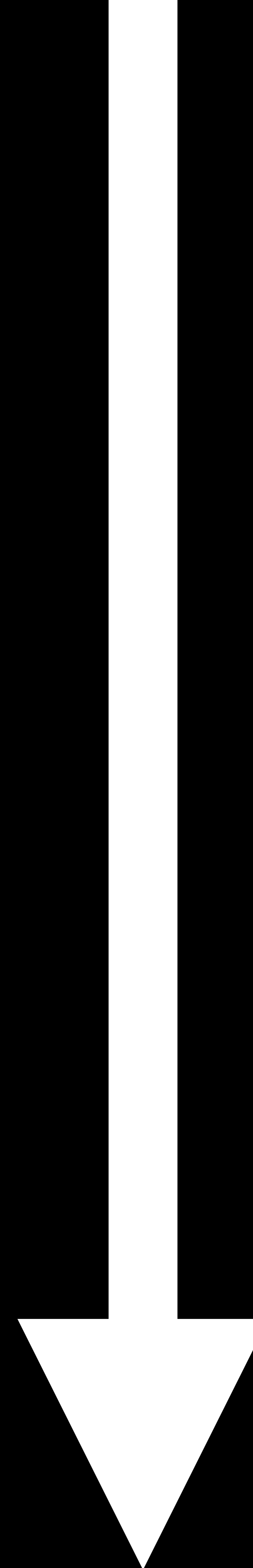
```
text1.minX = 8  
text1.width = 100  
text2.minX =           +           + 20
```


Engine



```
text1.minX = 8  
text1.width = 100  
text2.minX = 128
```

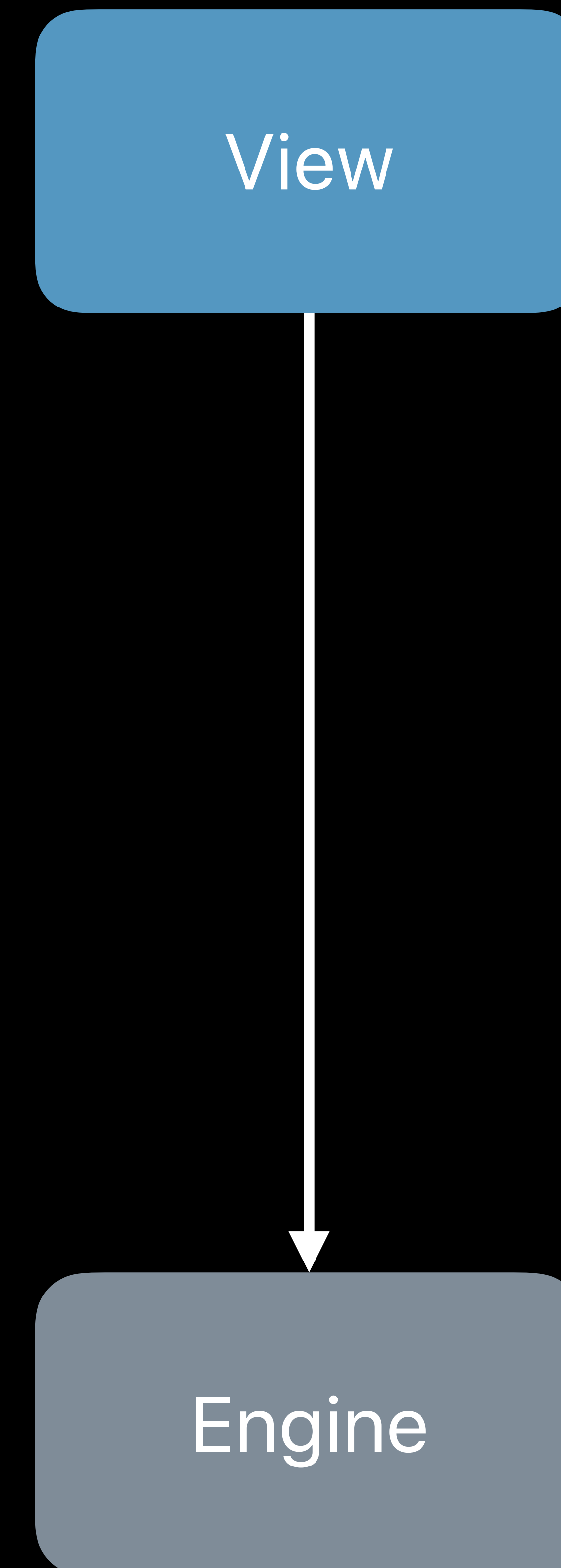
Engine



```
text1.minX = 8  
text1.width = 100  
text2.minX = 128  
text2.width = 100
```

Activating a Constraint

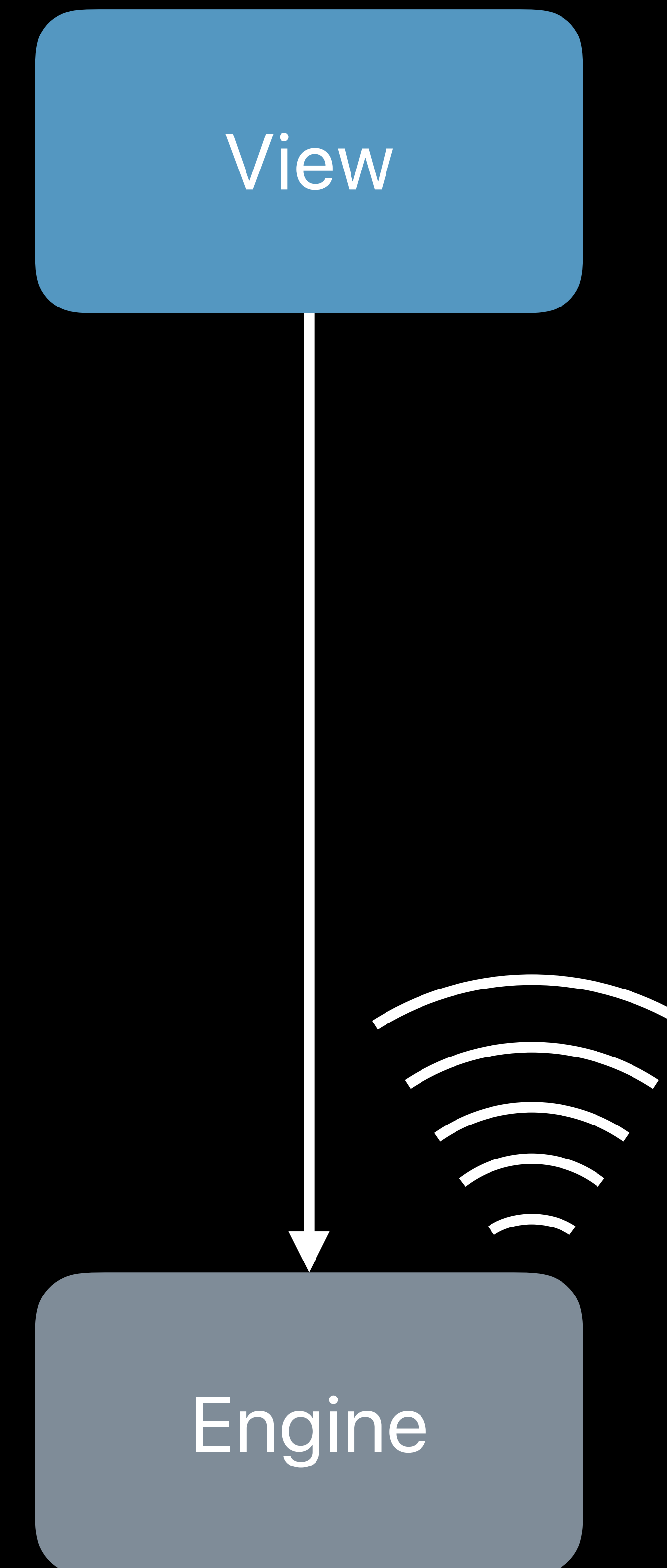
Triggering the layout pass



Activating a Constraint

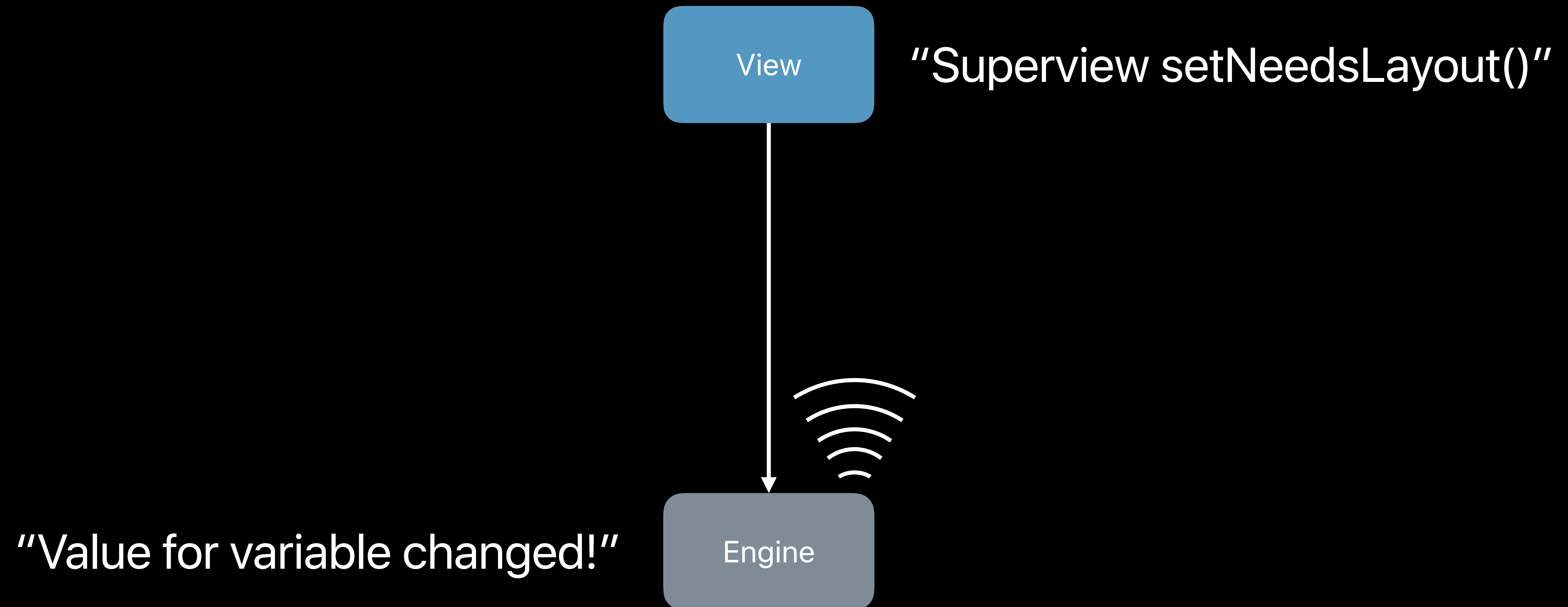
Triggering the layout pass

“Value for variable changed!”

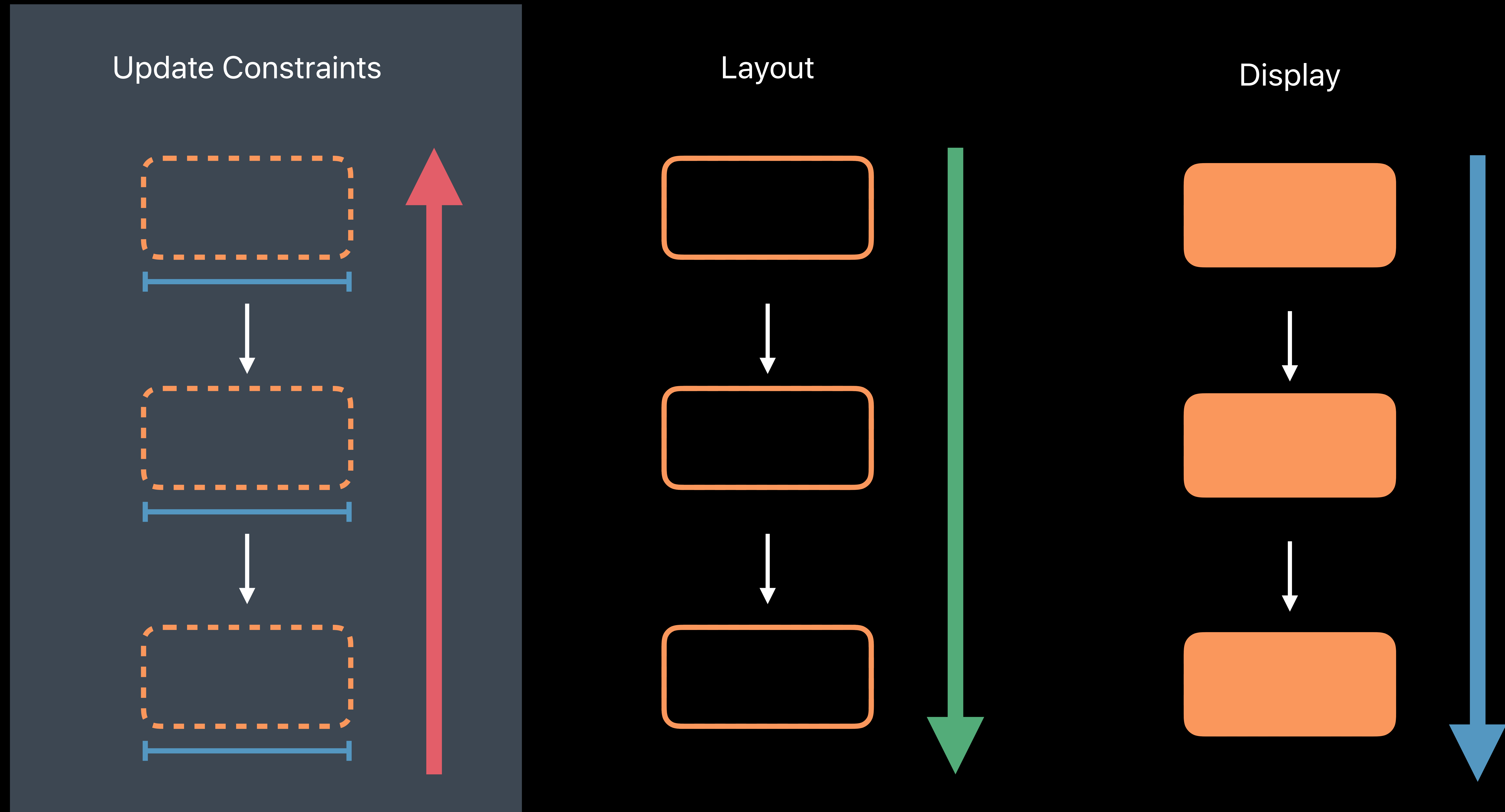


Activating a Constraint

Triggering the layout pass

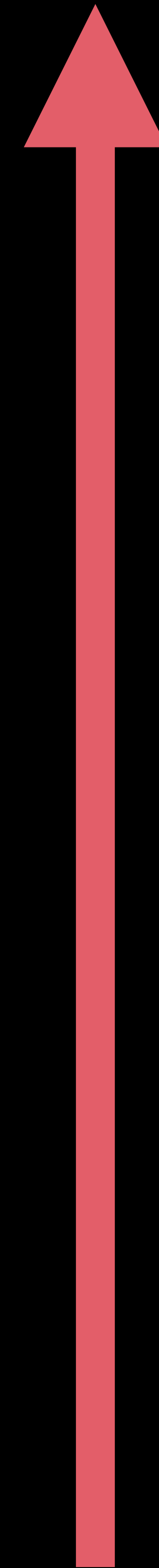
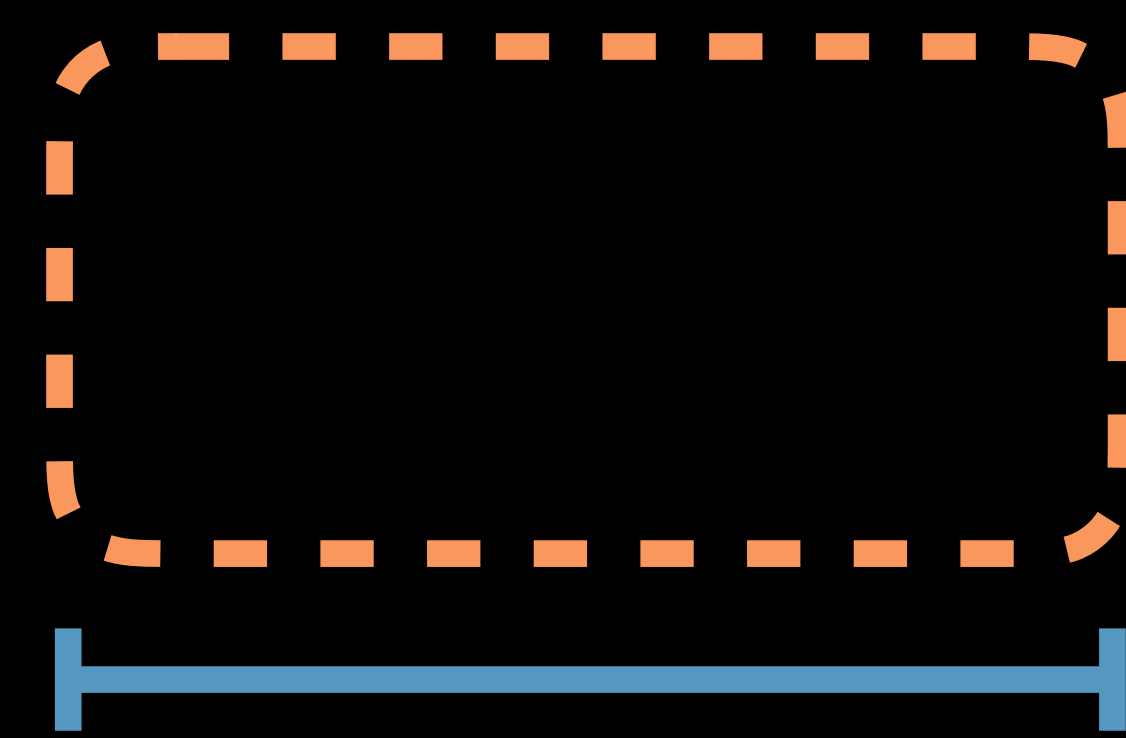
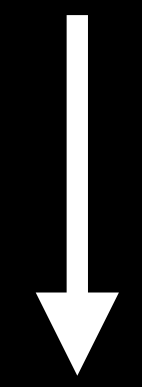
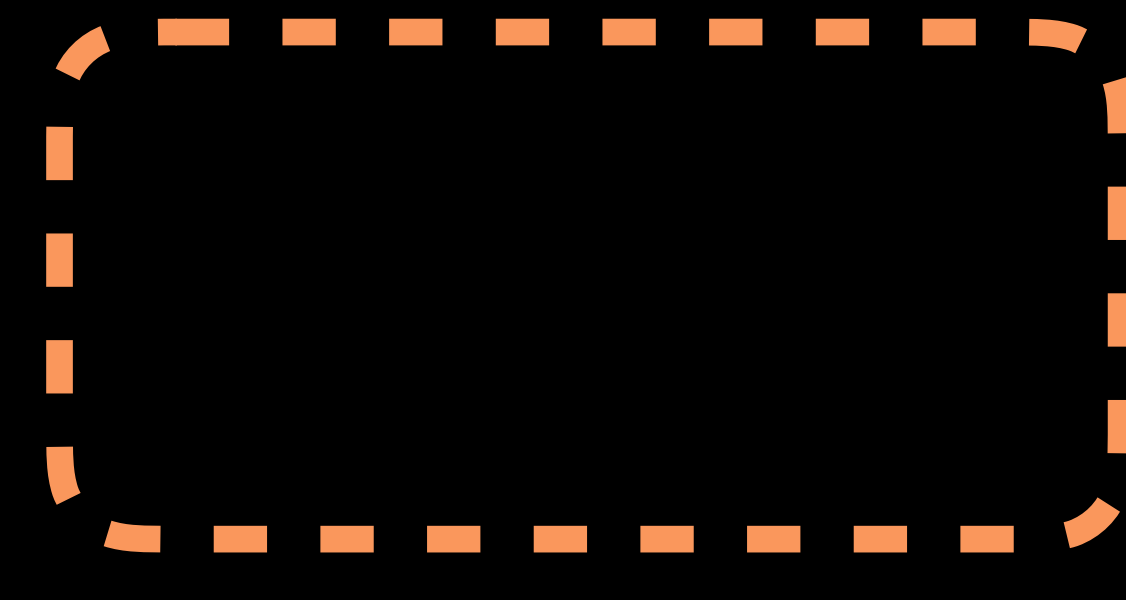
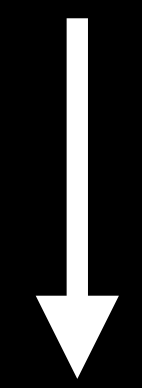
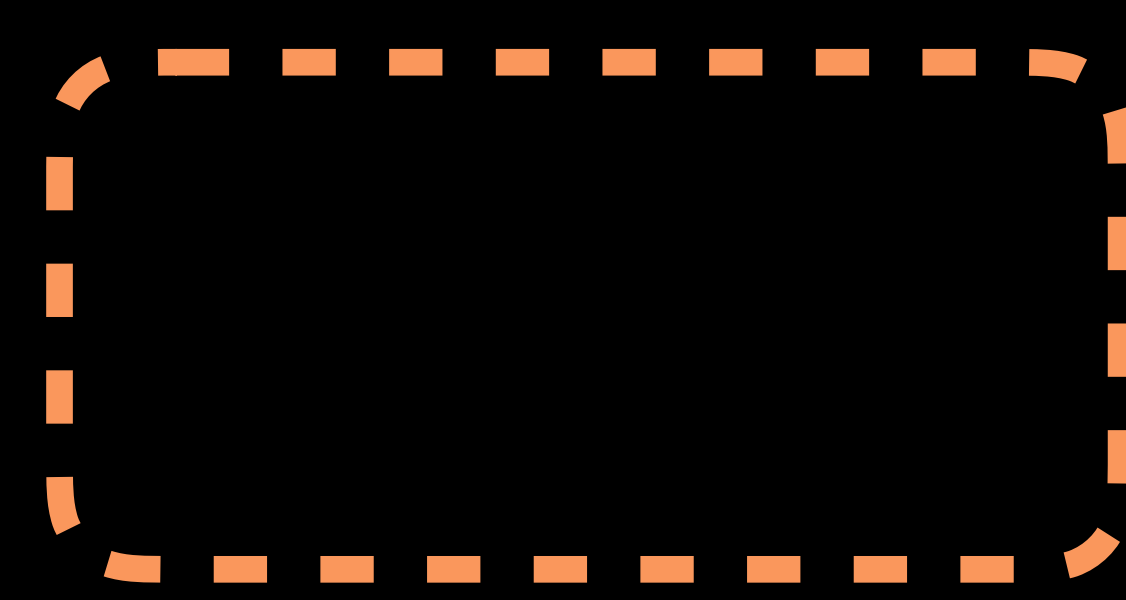


The Render Loop

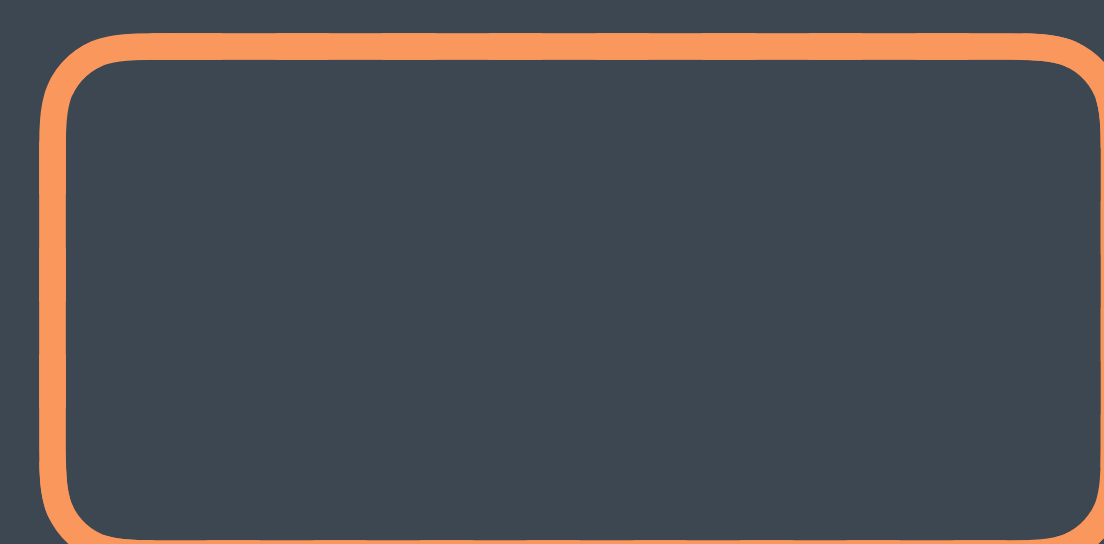


The Render Loop

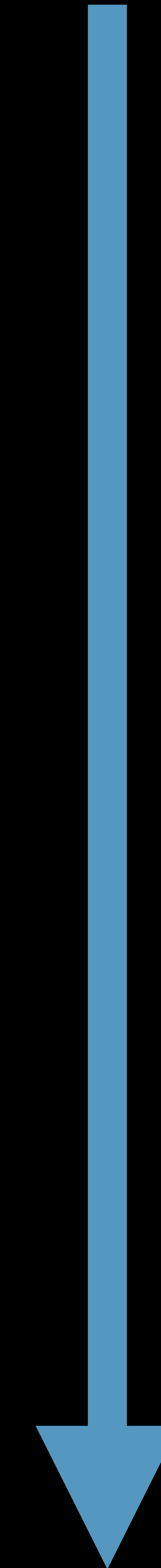
Update Constraints



Layout

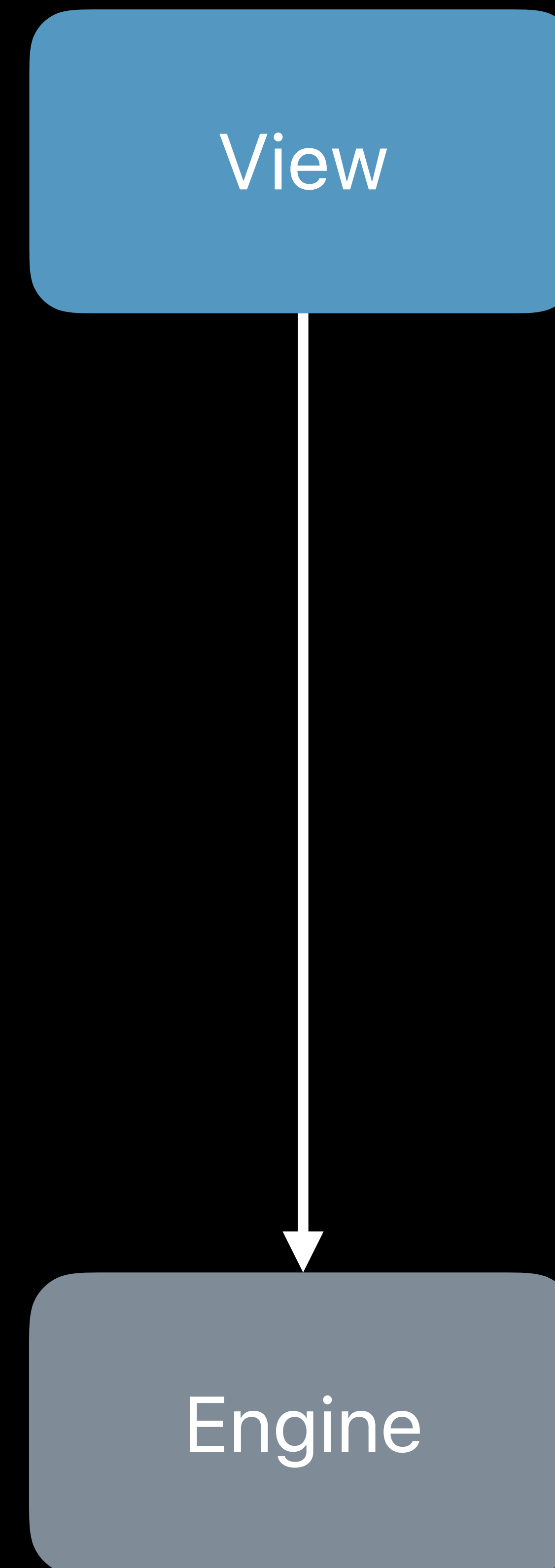


Display



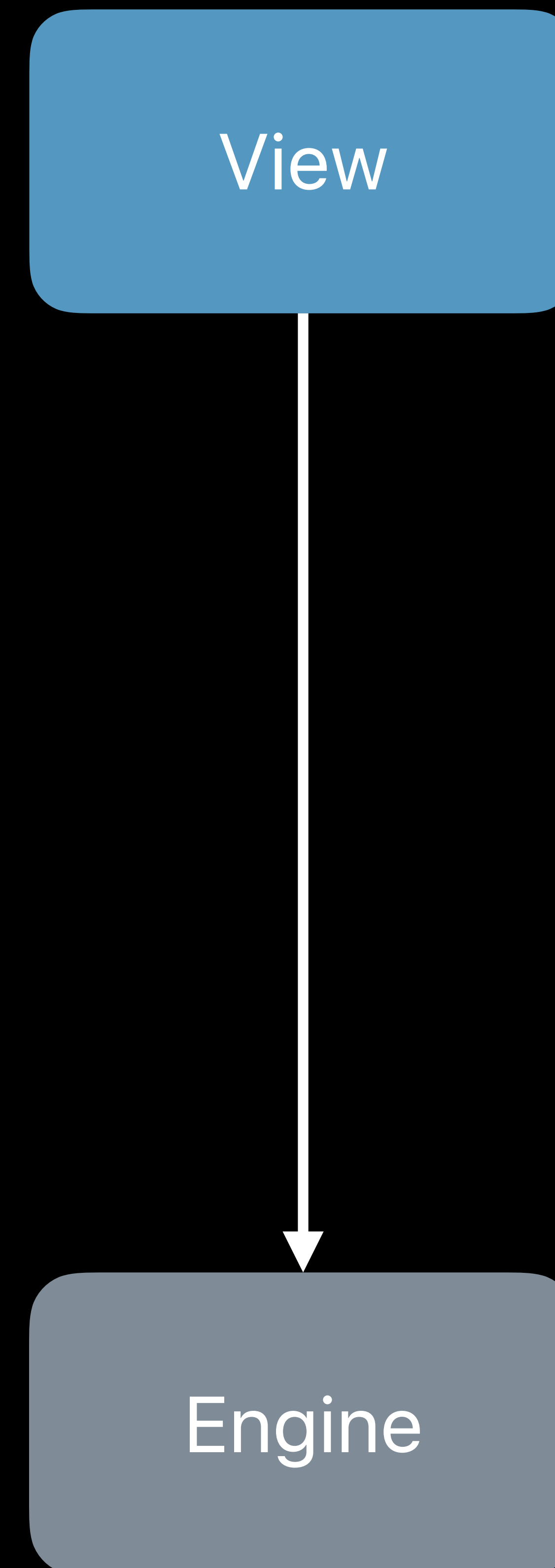
UIView.layoutSubviews()

Copying data from engine to subview



UIView.layoutSubviews()

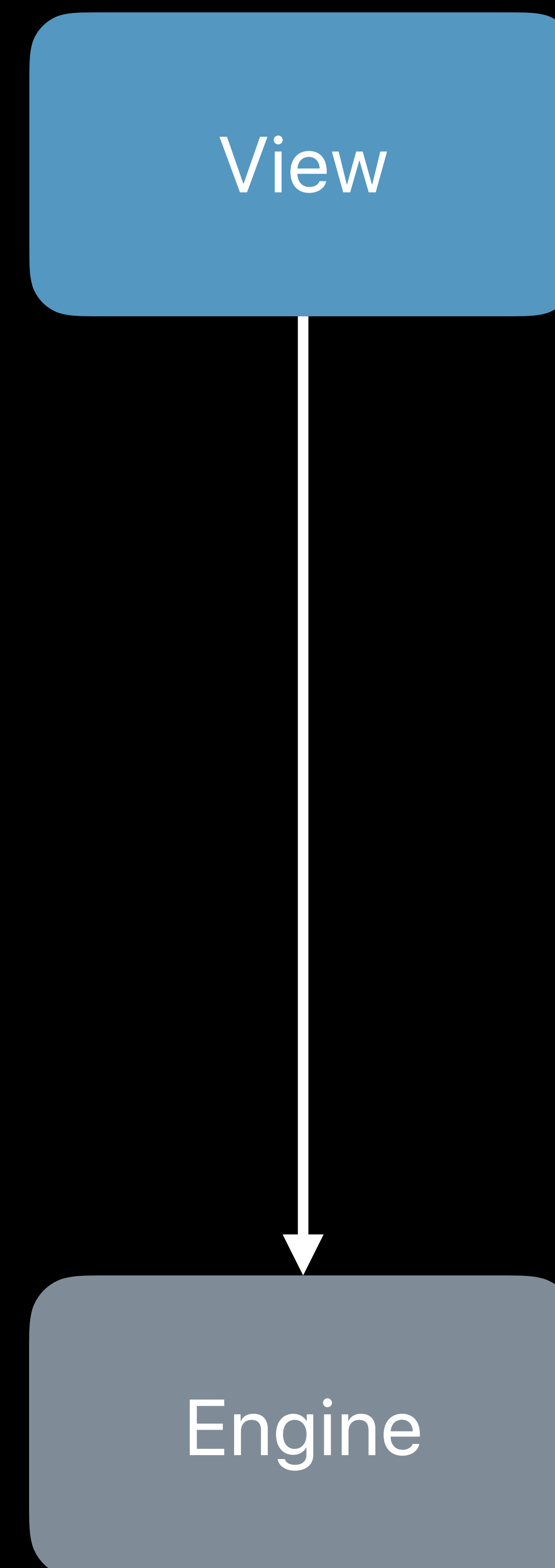
Copying data from engine to subview



“Engine, what are the values for the variables?”

UIView.layoutSubviews()

Copying data from engine to subview

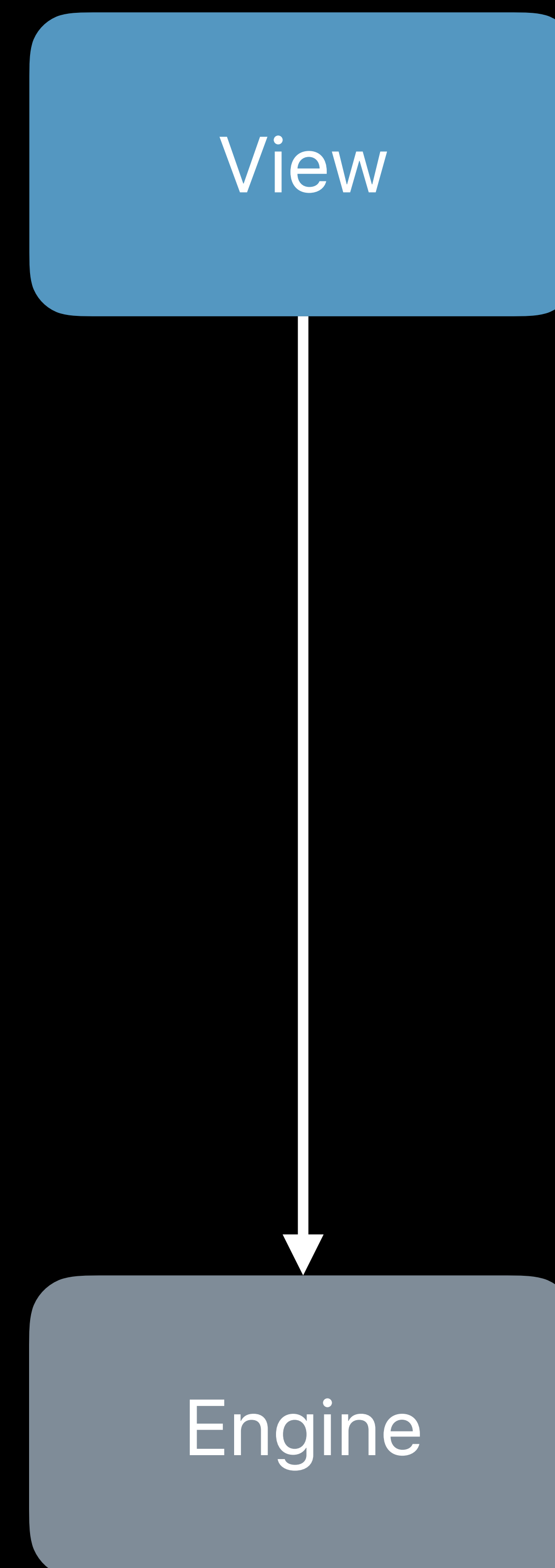


"Engine, what are the values for the variables?"

"Subview setBounds(), subview setCenter()"

UIView.layoutSubviews()

Copying data from engine to subview



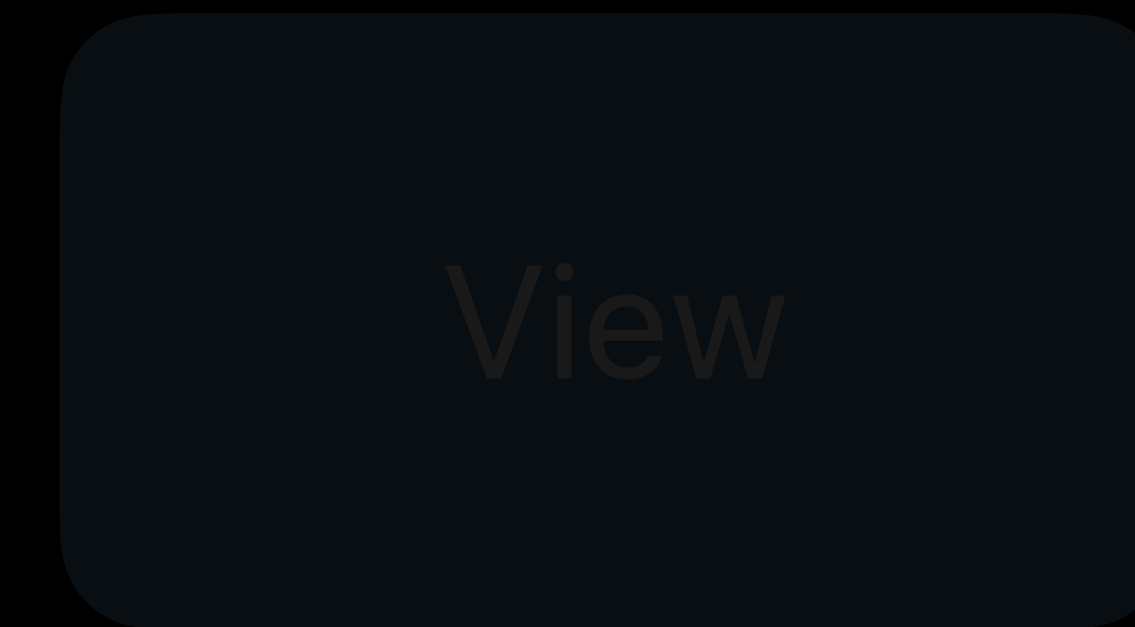
“Engine, what are the values for the variables?”

“Subview setBounds(), subview setCenter()”

`UIView.layoutSubviews()`

Copying data from engine to subview

Done!



"Engine, what are the values for the variables?"

"Subview.setBounds(), subview.setCenter()"

// Don't do this! Removes and re-adds constraints potentially at 120 frames per second



```
override func updateConstraints() {
    NSLayoutConstraint.deactivate(myConstraints)
    myConstraints.removeAll()
    let views = ["text1":text1, "text2":text2]
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "H:|-[text1]-[text2]",
                                                options: [.alignAllFirstBaseline],
                                                metrics: nil,
                                                views: views)
    myConstraints += NSLayoutConstraint.constraints(withVisualFormat: "V:|-[text1]-|",
                                                options: [],
                                                metrics: nil,
                                                views: views)
    NSLayoutConstraint.activate(myConstraints)
    super.updateConstraints()
}
```

Churning

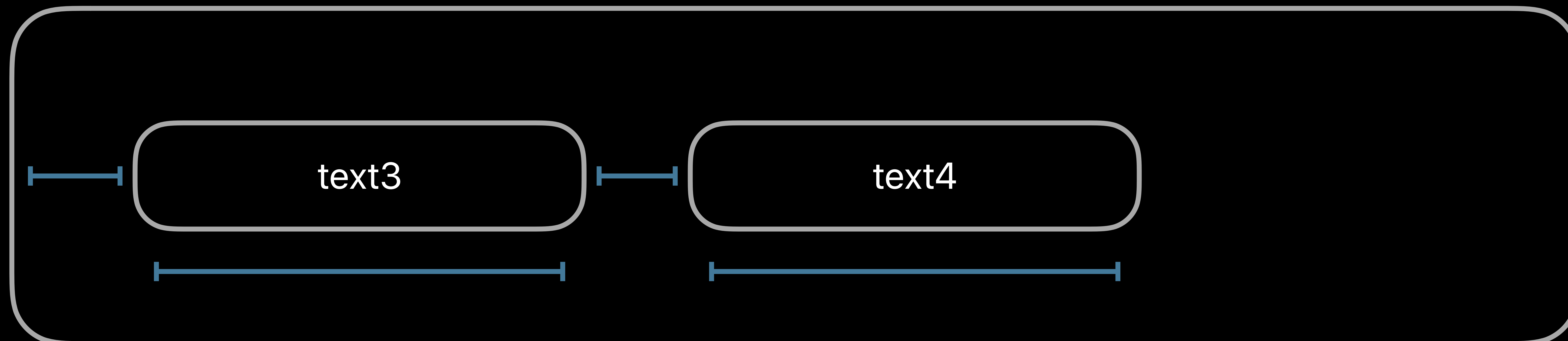
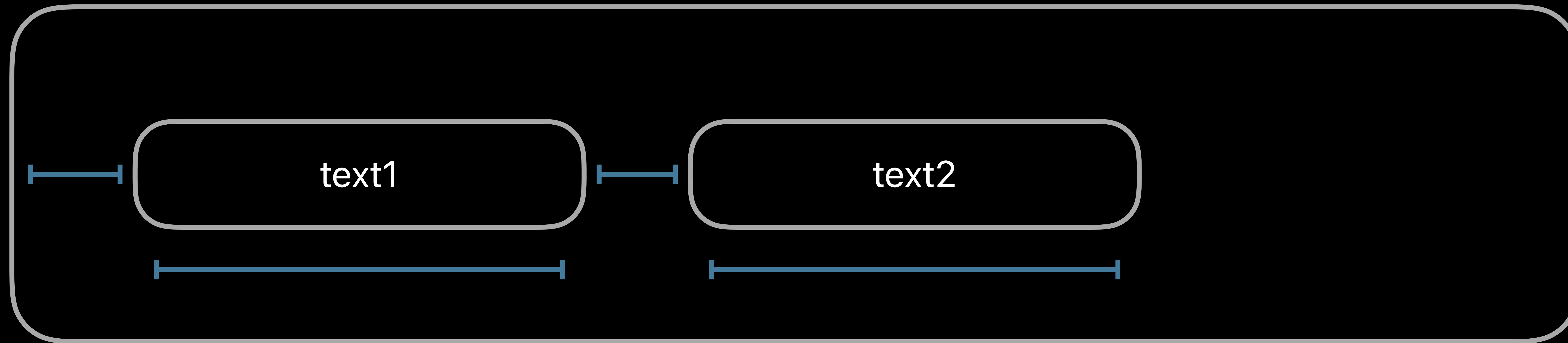


Churning



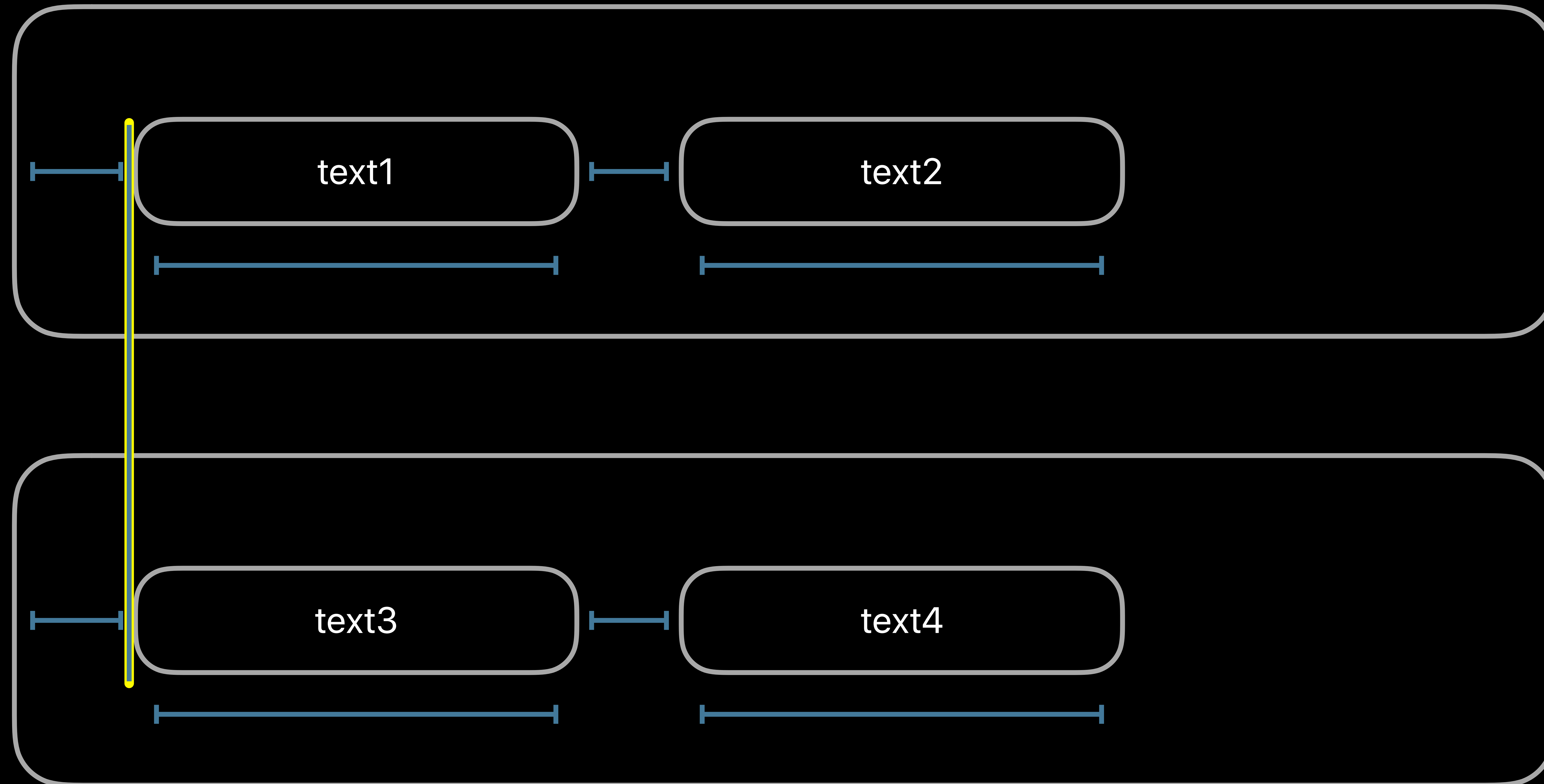
Local Versus Global Layout

You don't pay for what you don't use



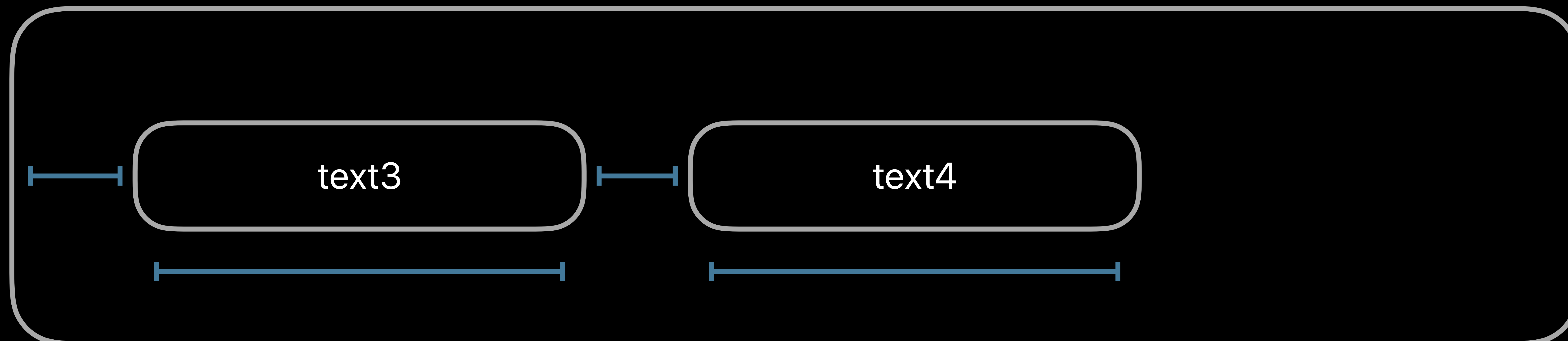
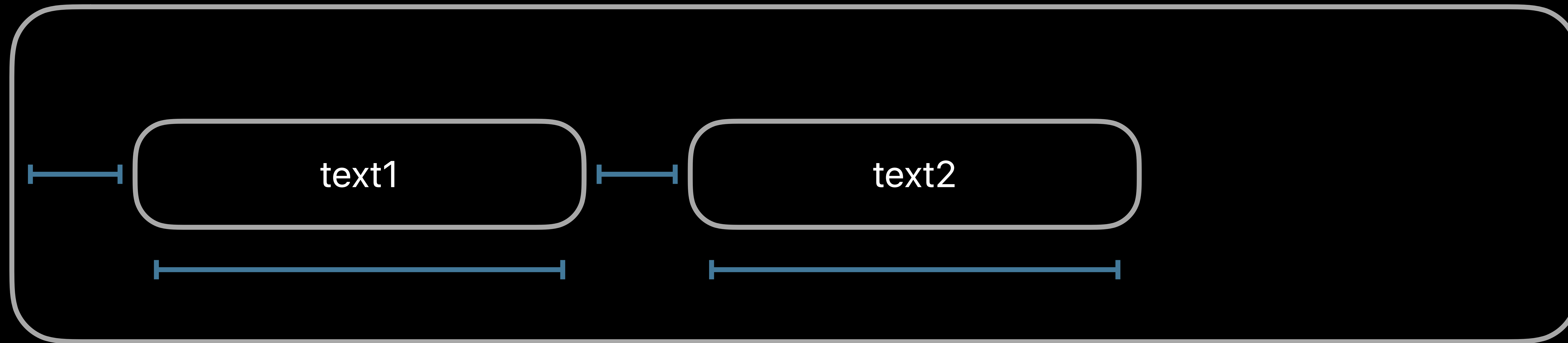
Local Versus Global Layout

You don't pay for what you don't use



Local Versus Global Layout

You don't pay for what you don't use



Unrelated Views Don't Interact



Unrelated Views Don't Interact

```
text1.minX = 8  
text1.width = 100  
text2.minX = 20 + text1.minX + text1.width
```

```
text3.minX = 8  
text3.width = 100  
text4.minX = 20 + text3.minX + text3.width
```


Unrelated Views Don't Interact

```
text1.minX = 8  
text1.width = 100  
text2.minX = 128
```

```
text3.minX = 8  
text3.width = 100  
text4.minX = 128
```

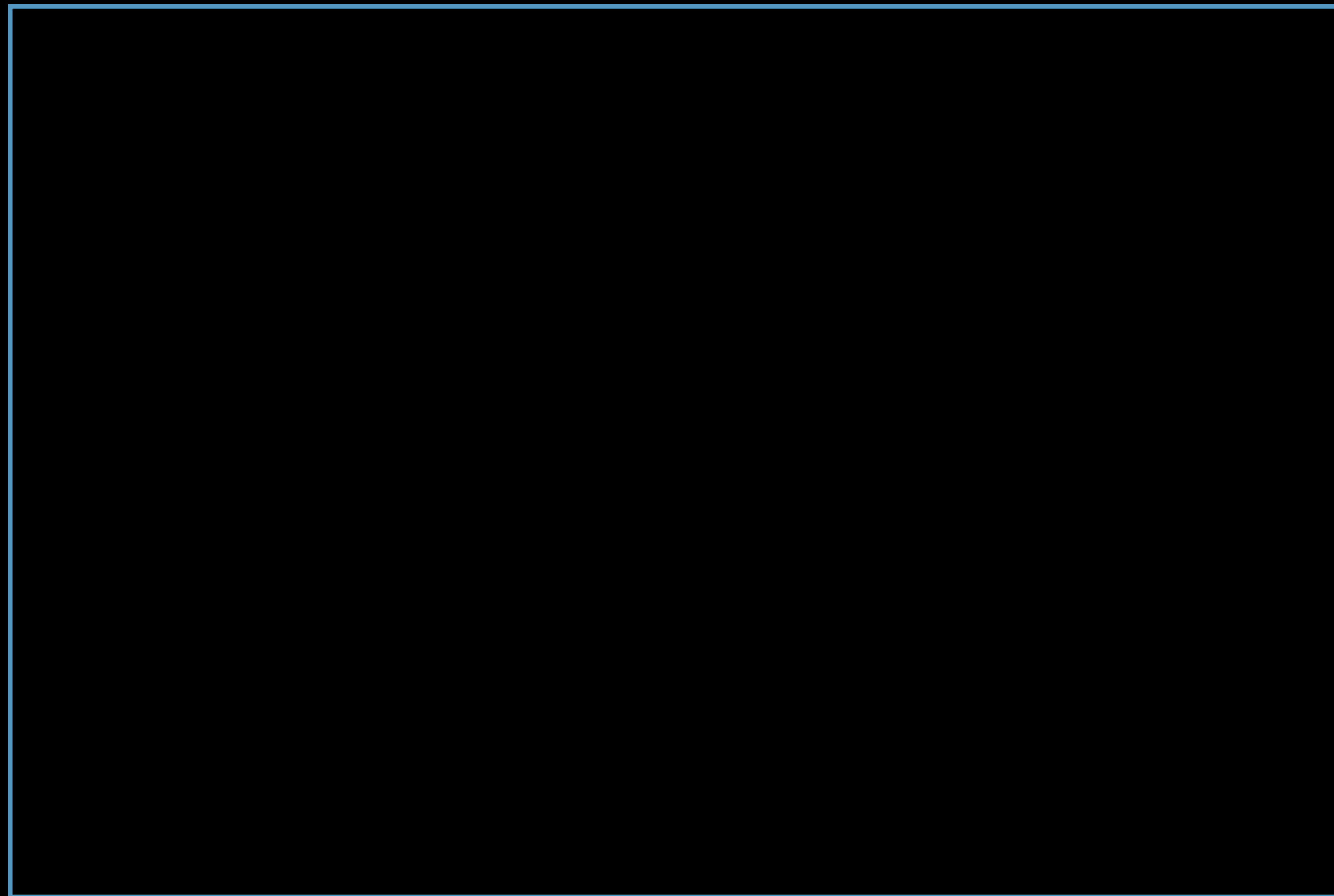
Unrelated Views Don't Interact

```
text1.minX = 8  
text1.width = 100  
text2.minX = 128  
text2.width = 100
```

```
text3.minX = 8  
text3.width = 100  
text4.minX = 128  
text4.width = 100
```

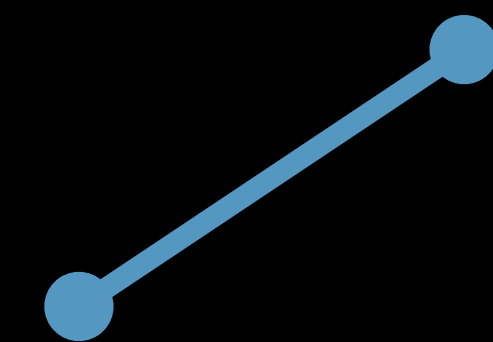
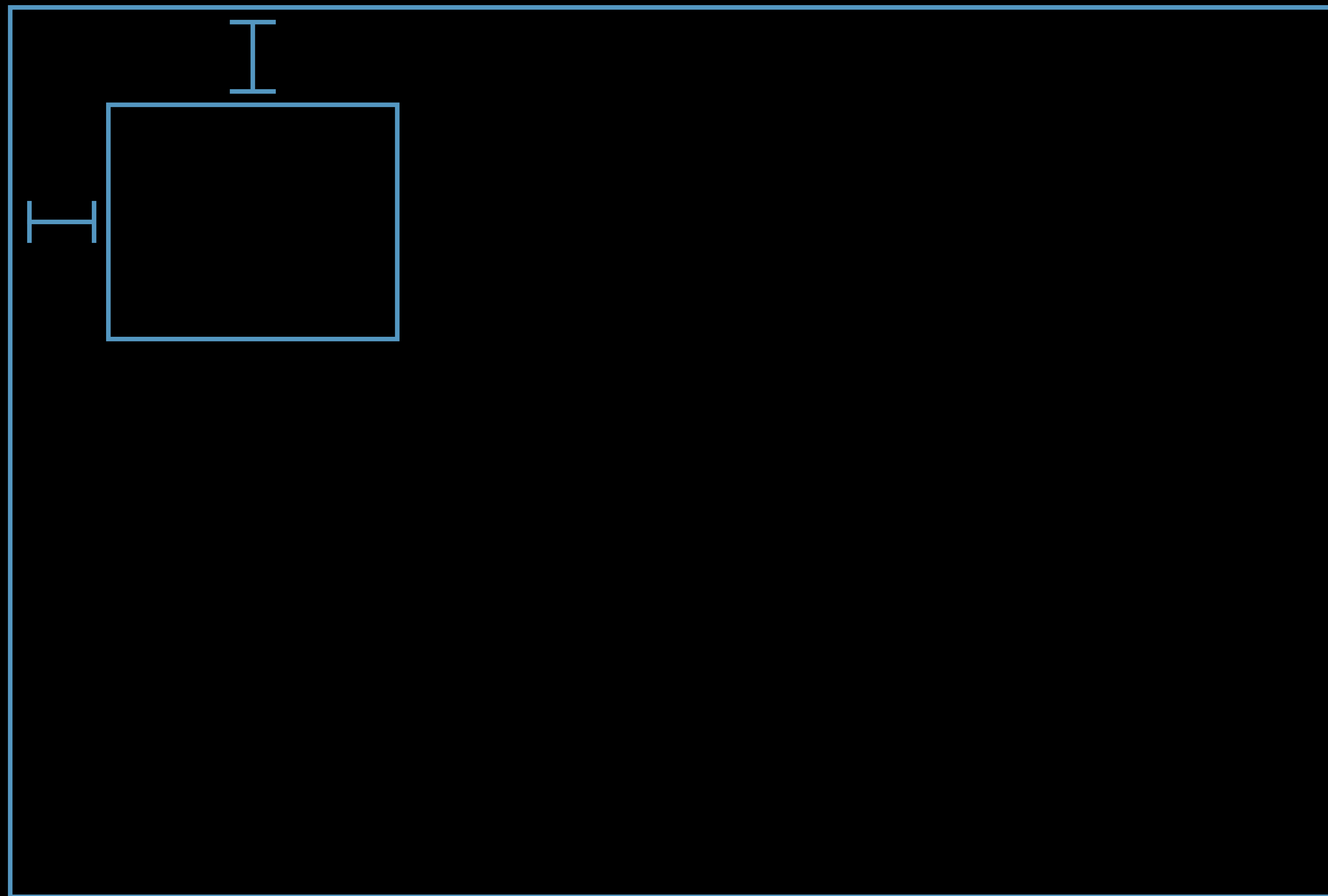
Performance Scales Linearly When No Interaction

How much work would it take by hand?



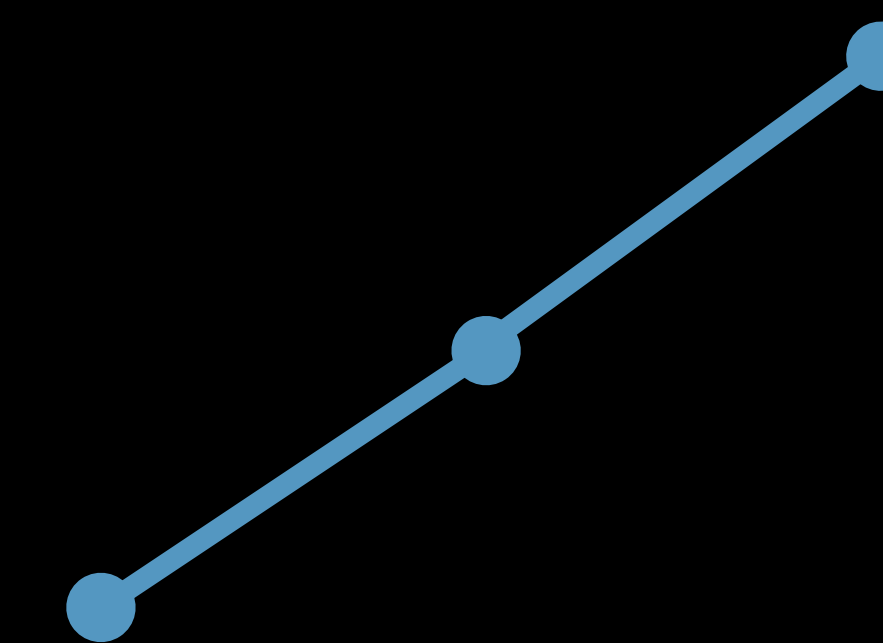
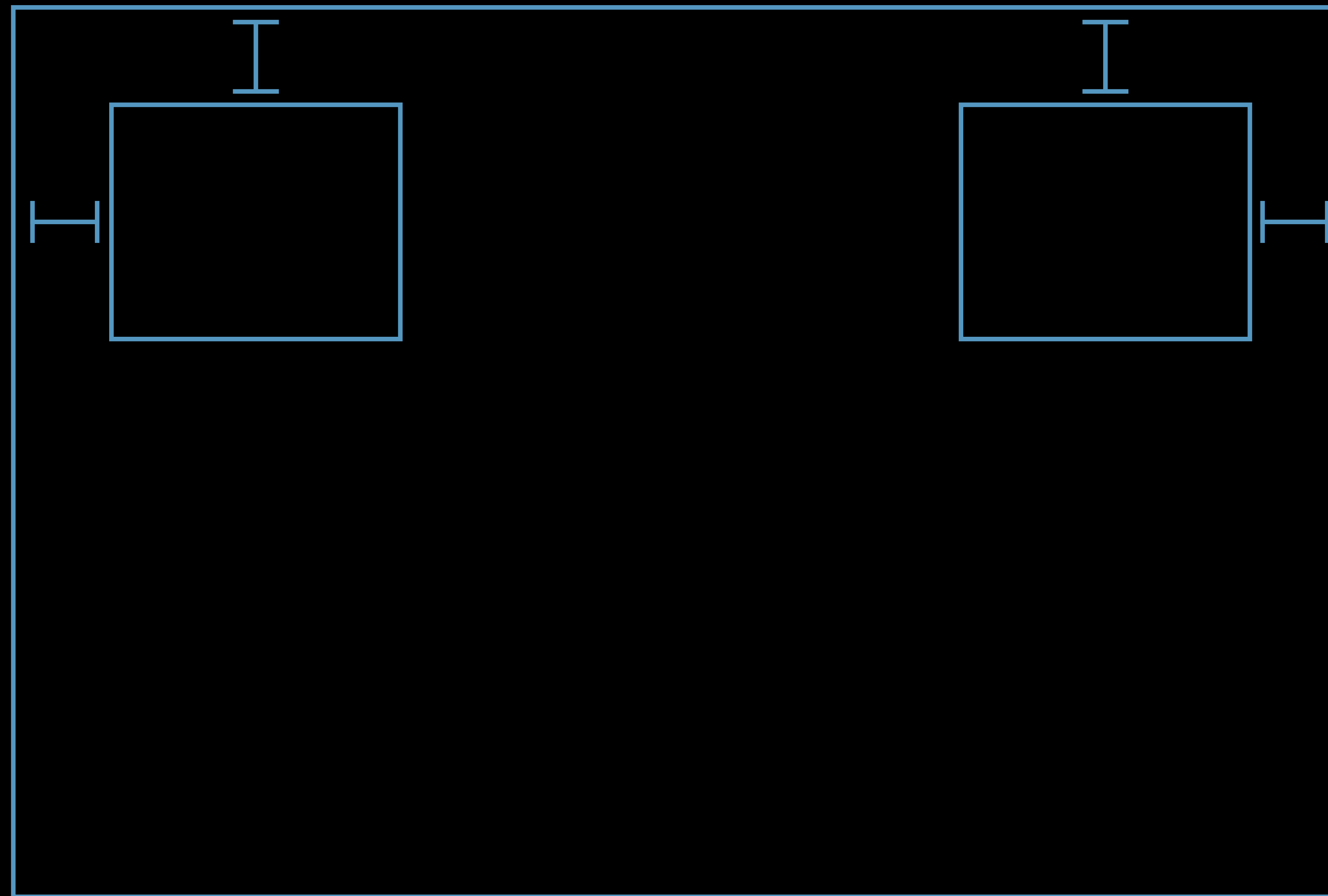
Performance Scales Linearly When No Interaction

How much work would it take by hand?



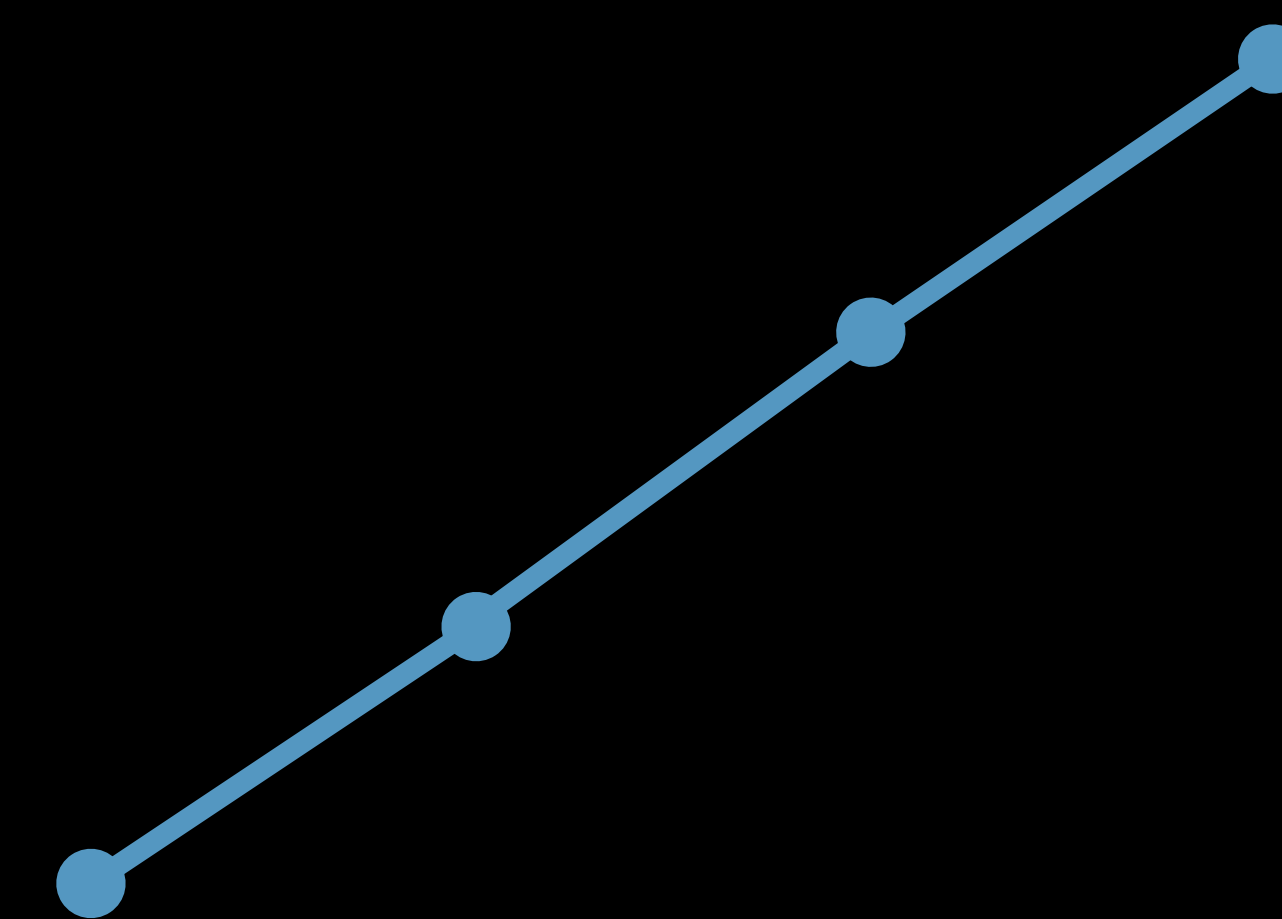
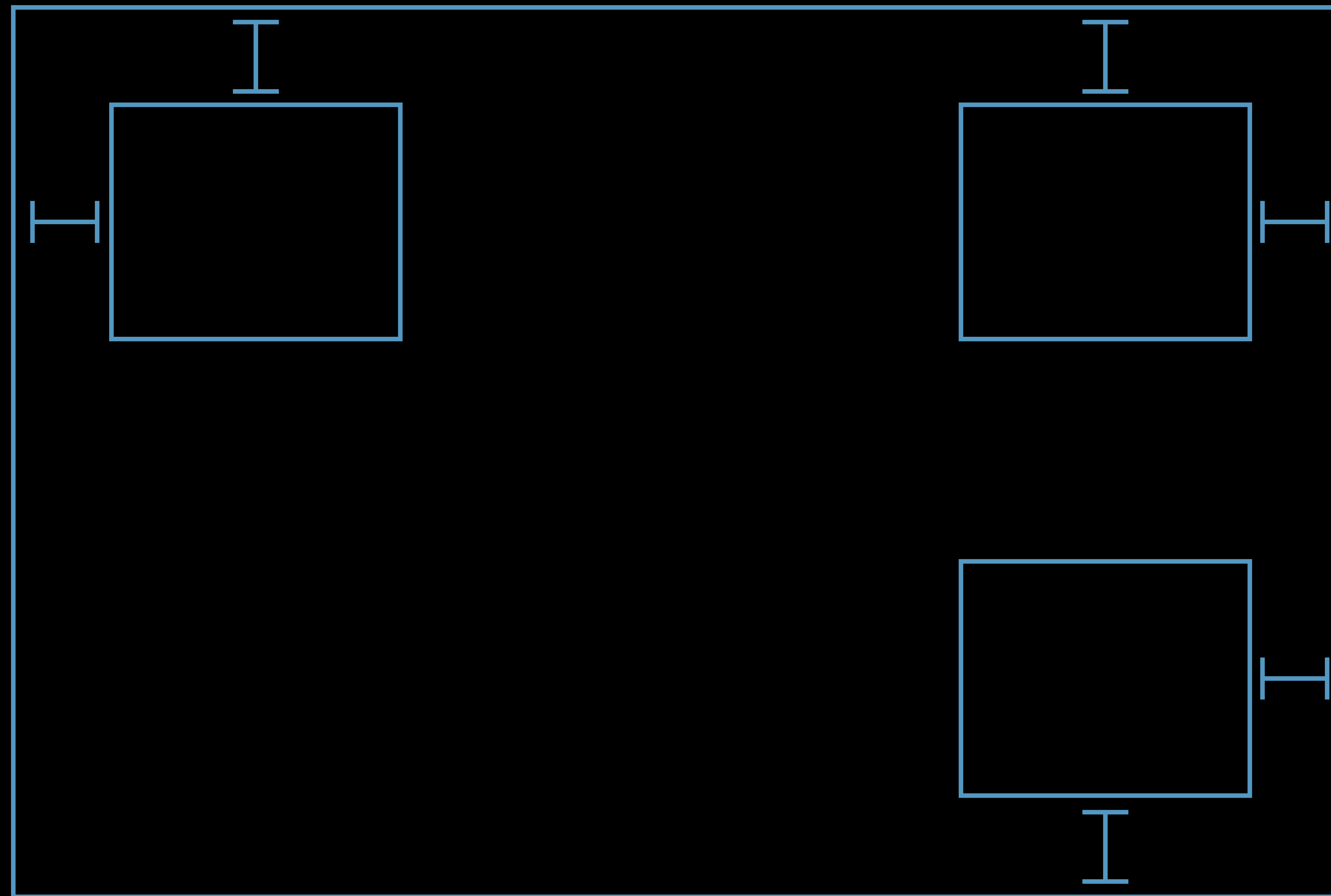
Performance Scales Linearly When No Interaction

How much work would it take by hand?



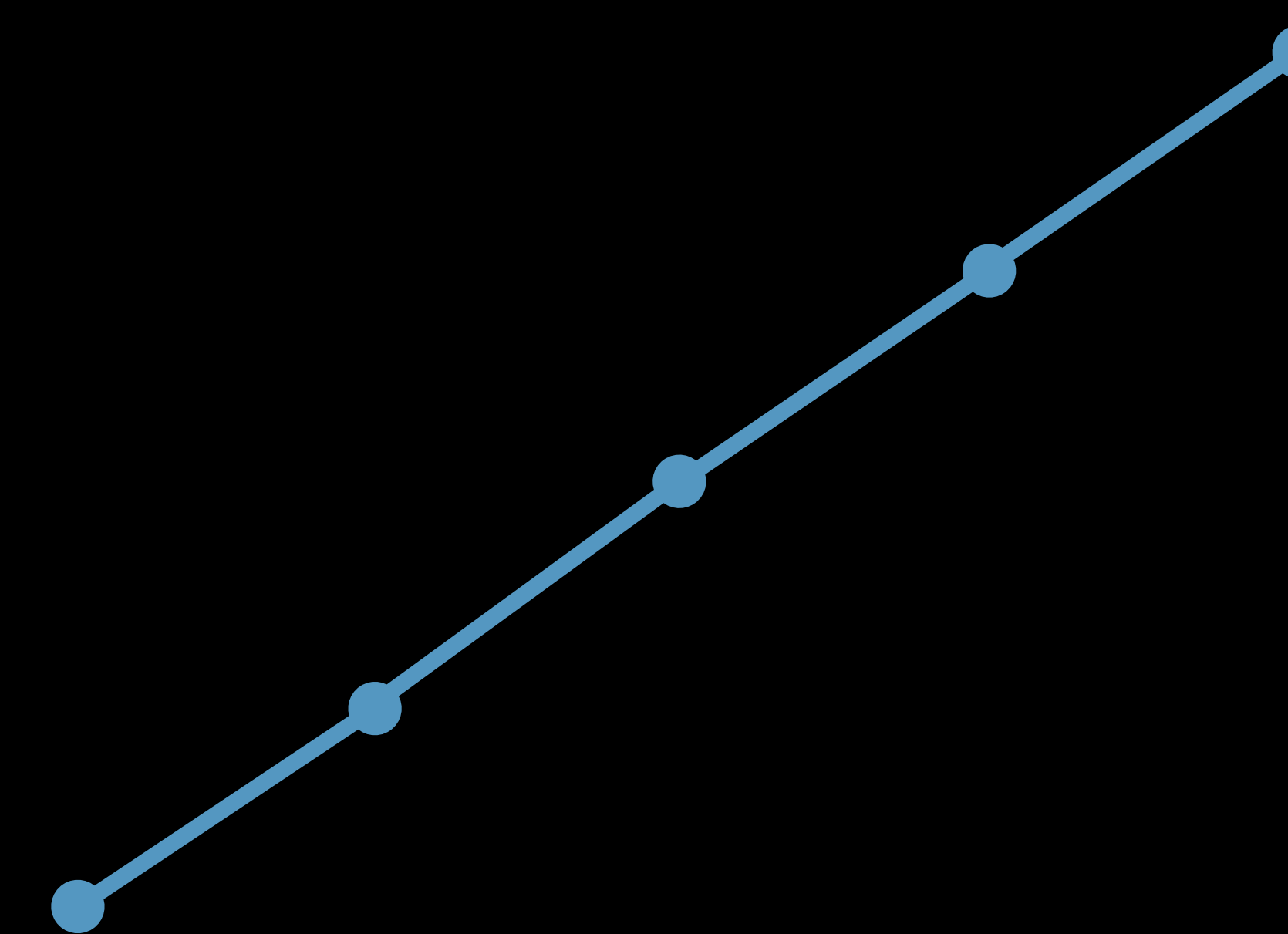
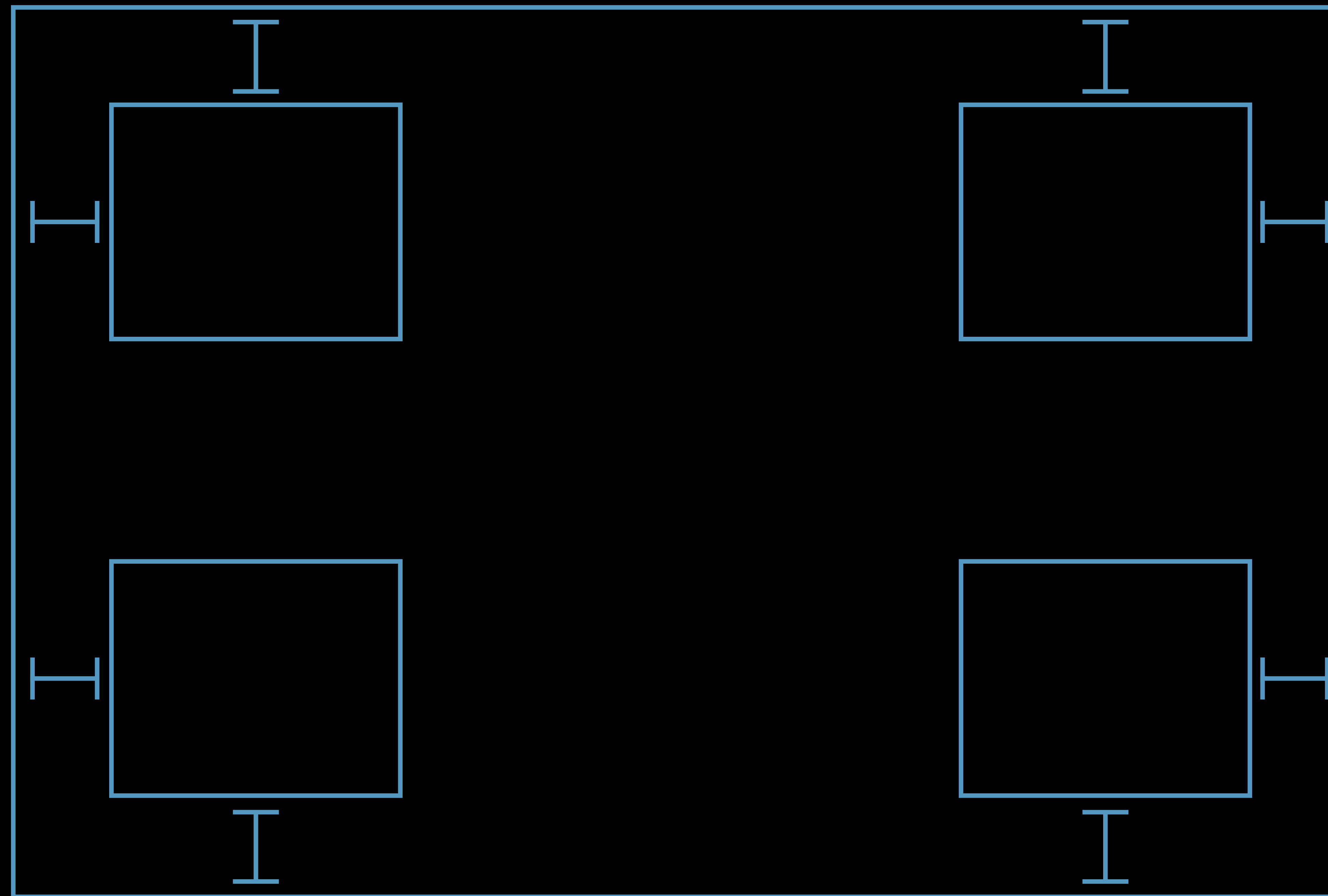
Performance Scales Linearly When No Interaction

How much work would it take by hand?



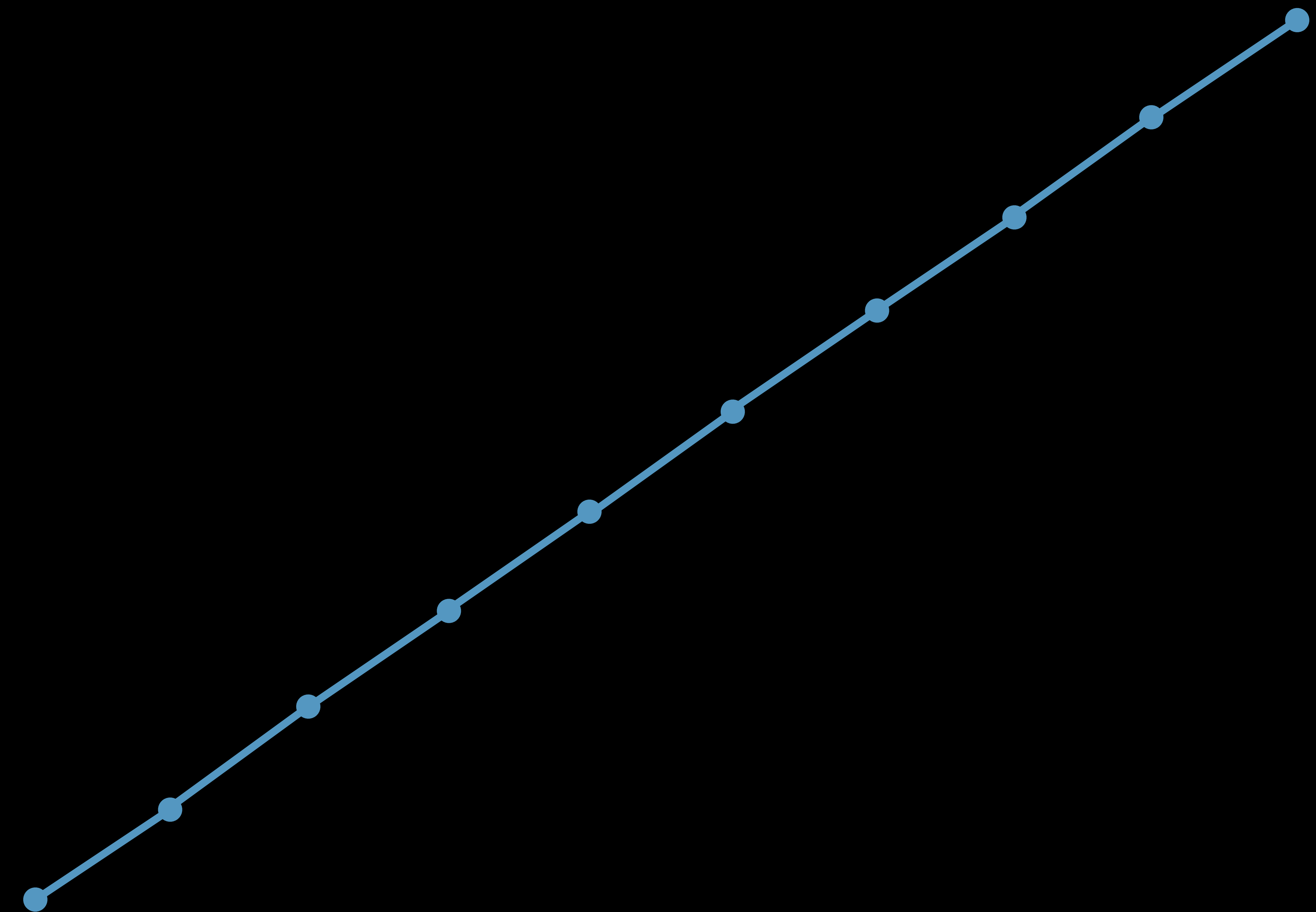
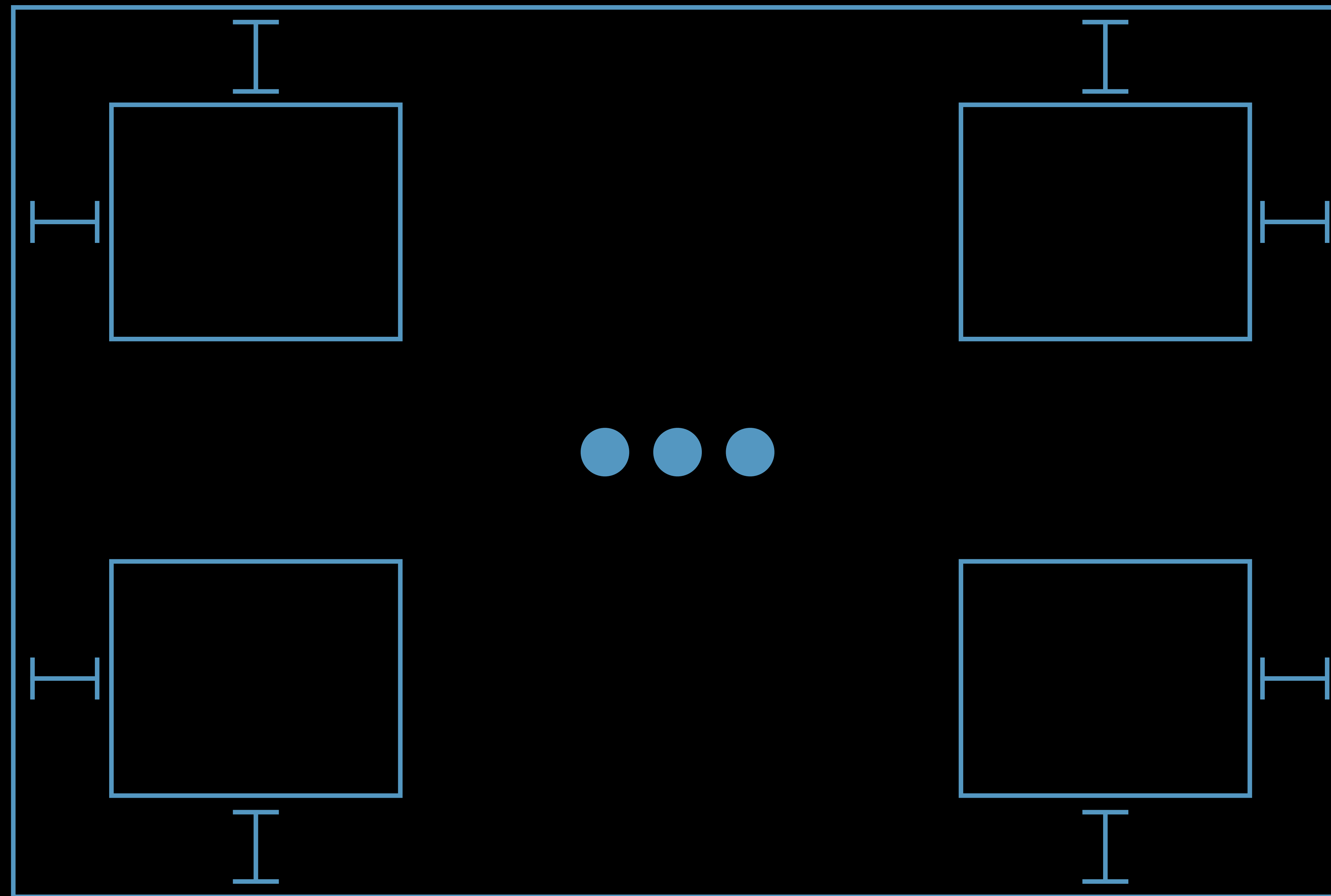
Performance Scales Linearly When No Interaction

How much work would it take by hand?



Performance Scales Linearly When No Interaction

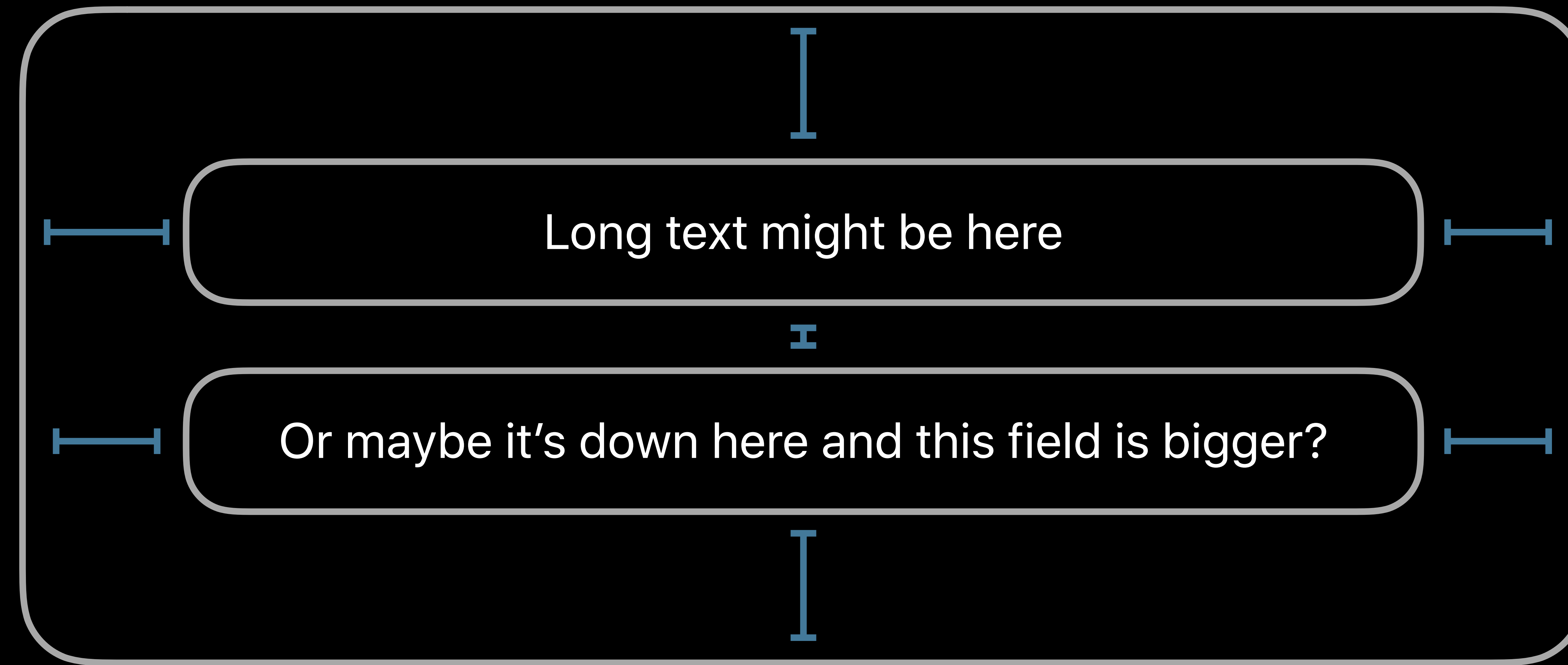
How much work would it take by hand?



The engine is a layout cache
and dependency tracker.

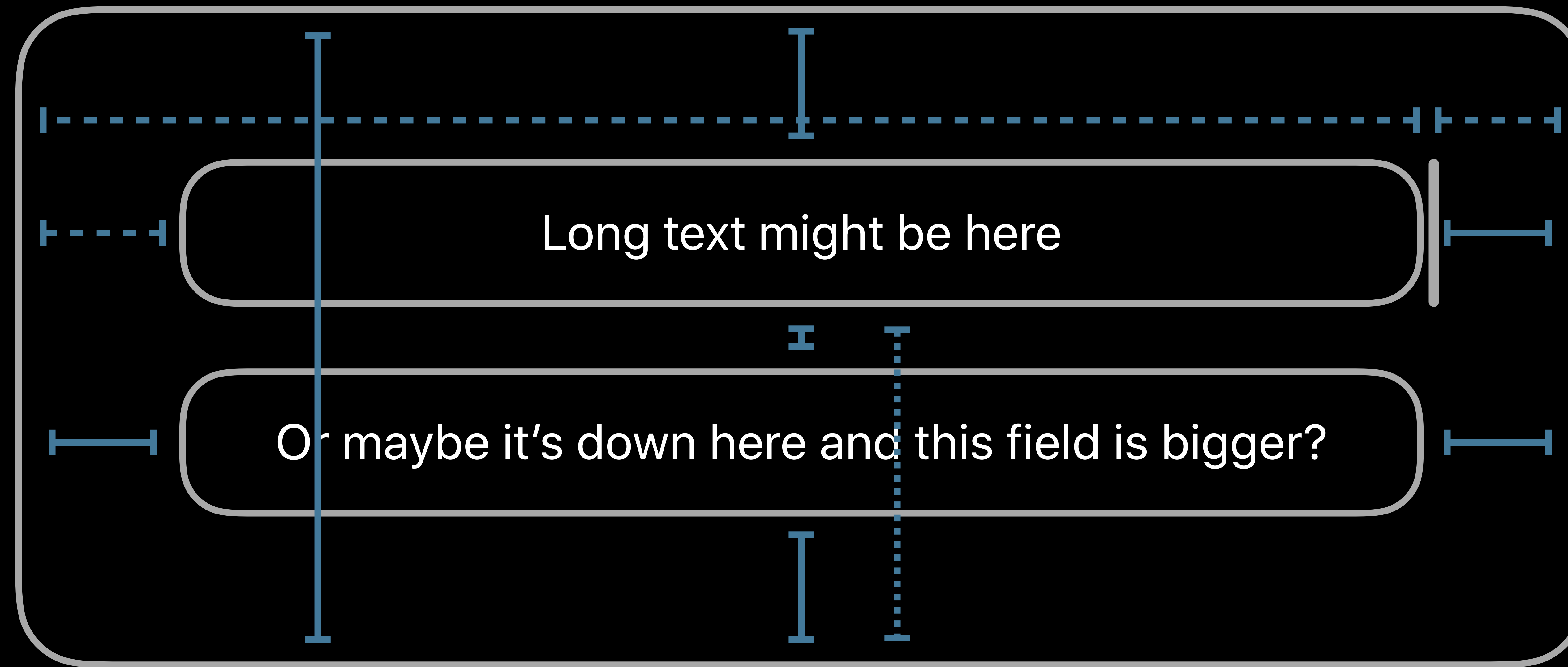
Model The Problem Naturally

Don't avoid using constraints



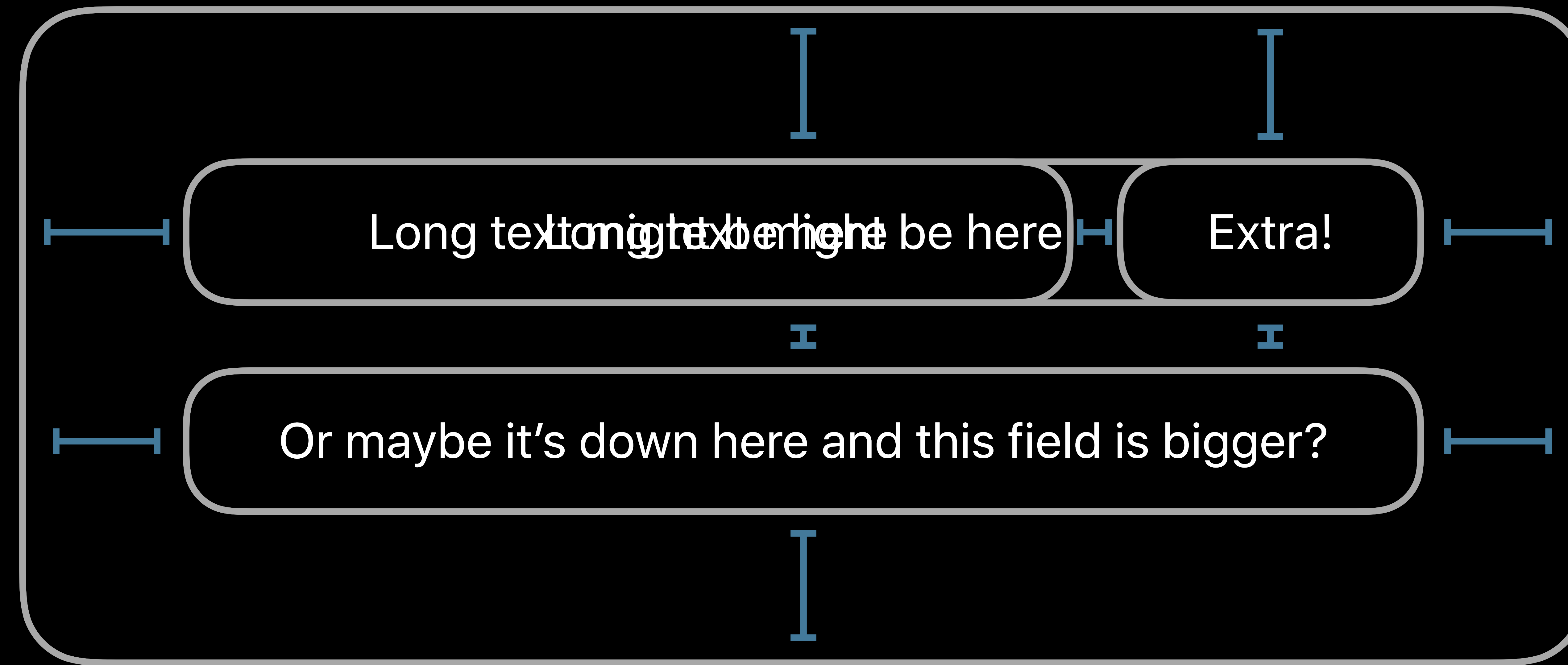
Model The Problem Naturally

Don't wedge two layouts into one set of constraints



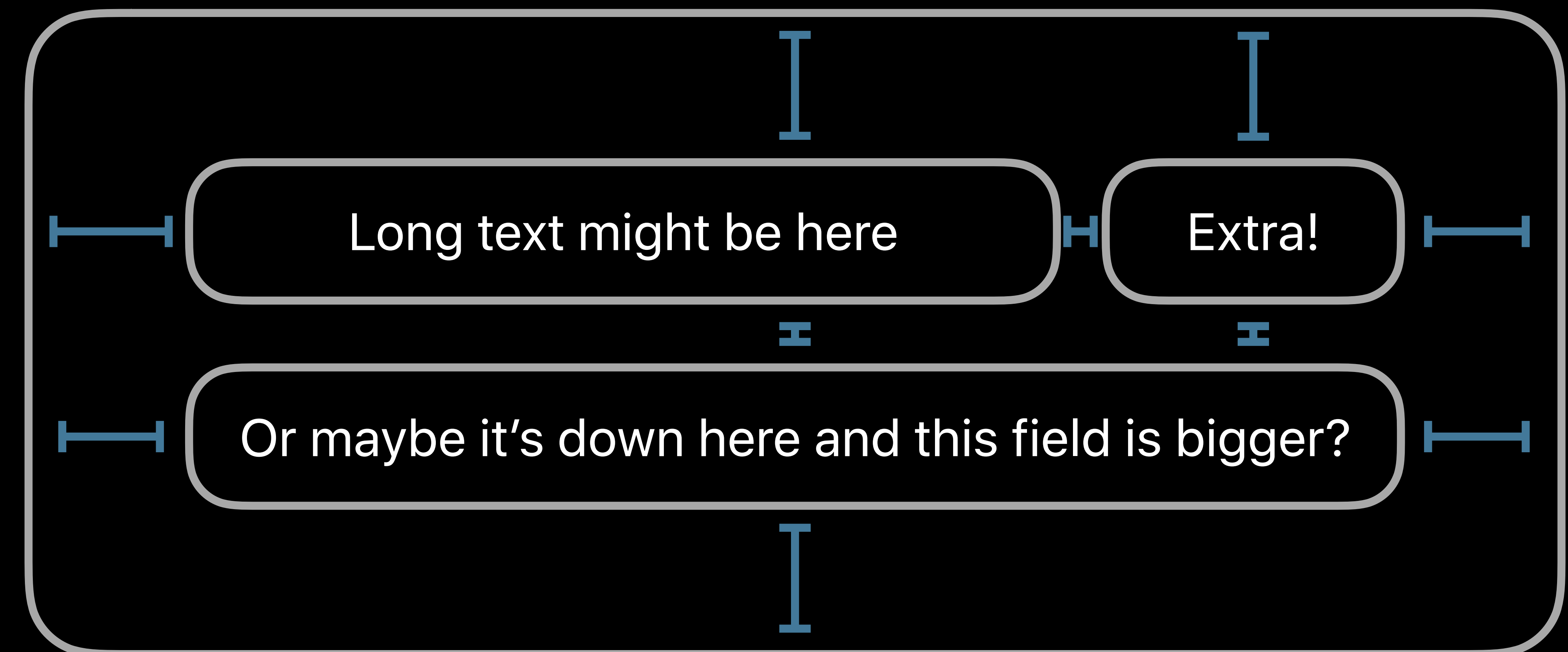
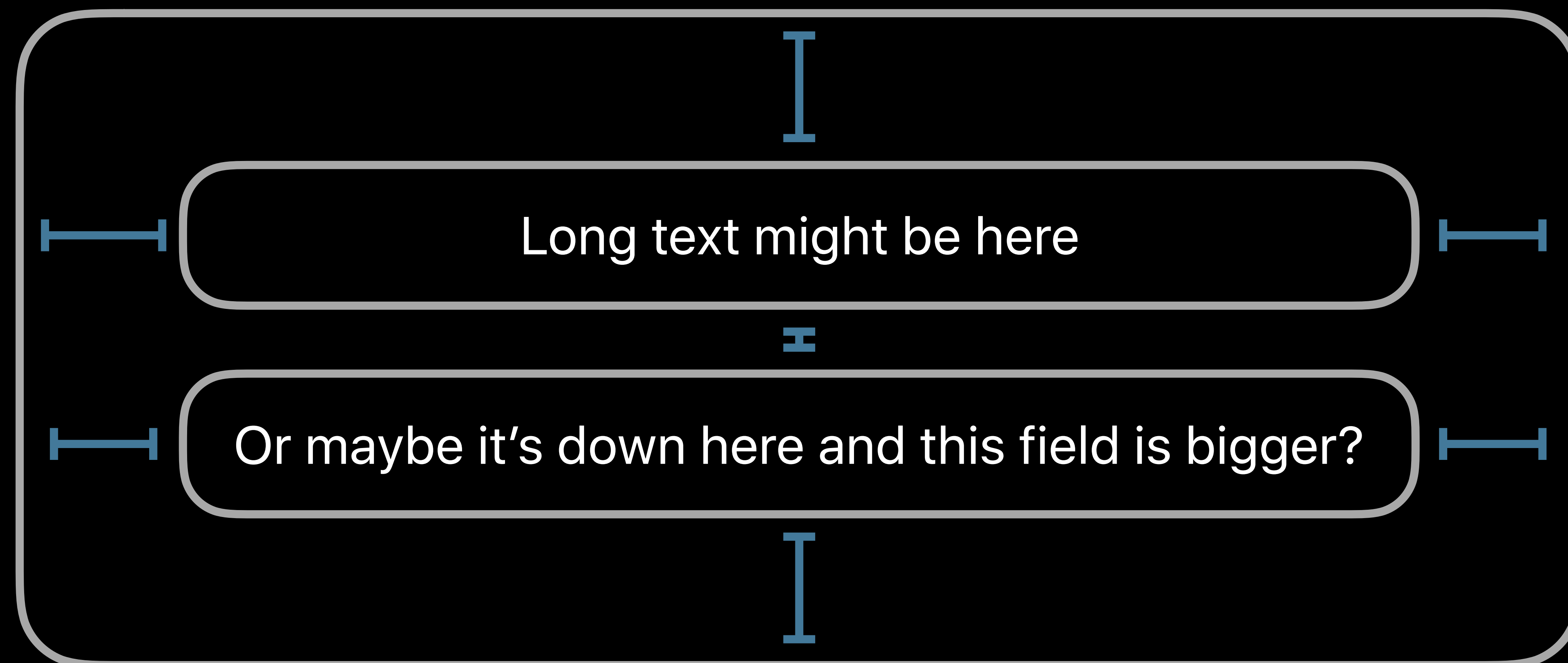
Model The Problem Naturally

Don't wedge two layouts into one set of constraints



Model The Problem Naturally

Don't wedge two layouts into one set of constraints



Is That Everything?

Costs of other auto layout features

Inequality

NSLayoutConstraint set constant

Priority

Is That Everything?

Costs of other auto layout features

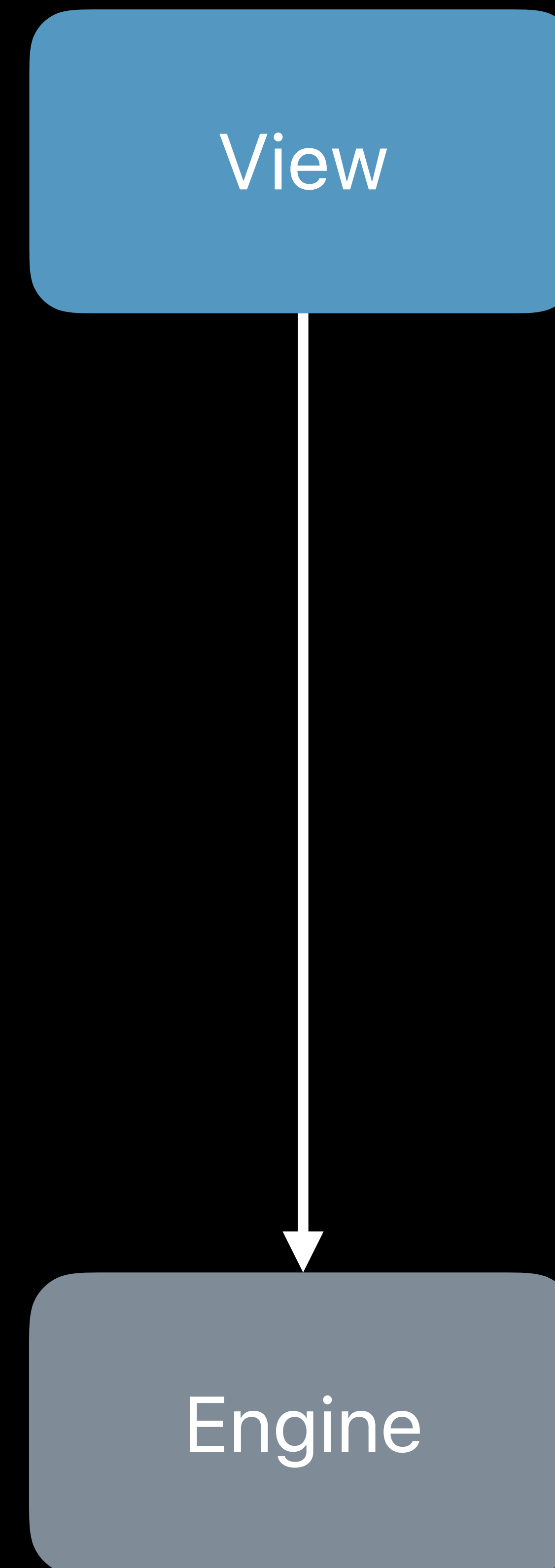
Inequality

NSLayoutConstraint set constant

Priority

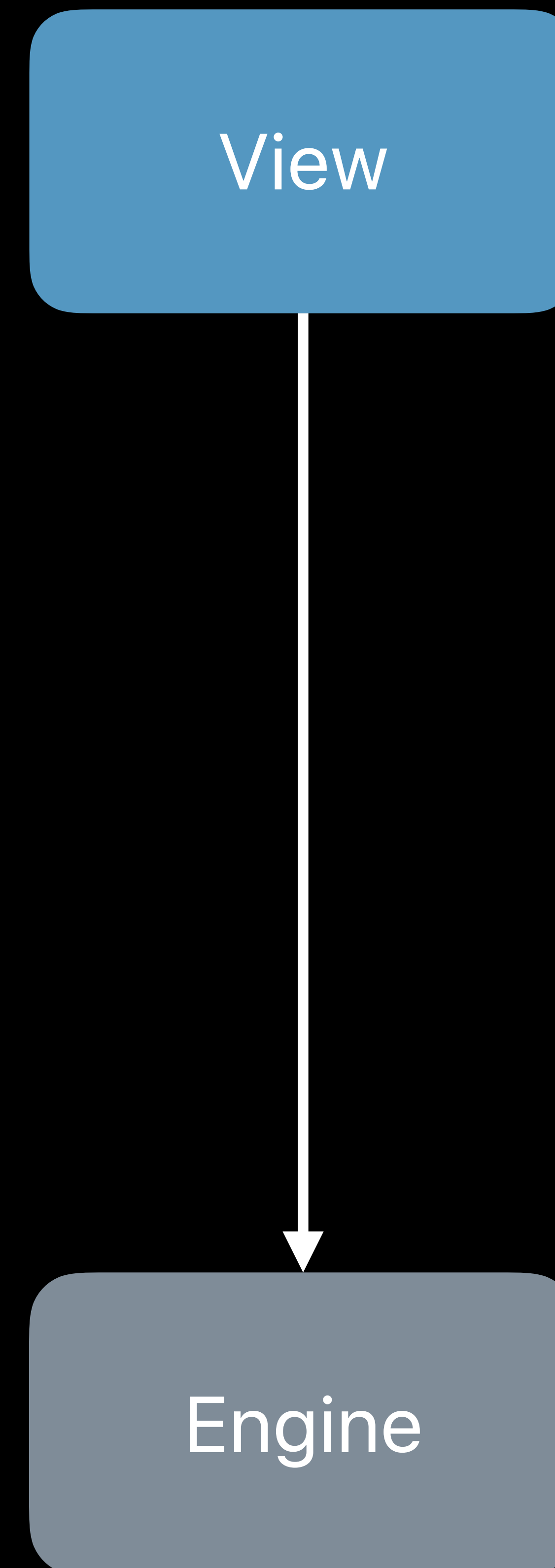
Error Minimization

Constraints with non-required priority



Error Minimization

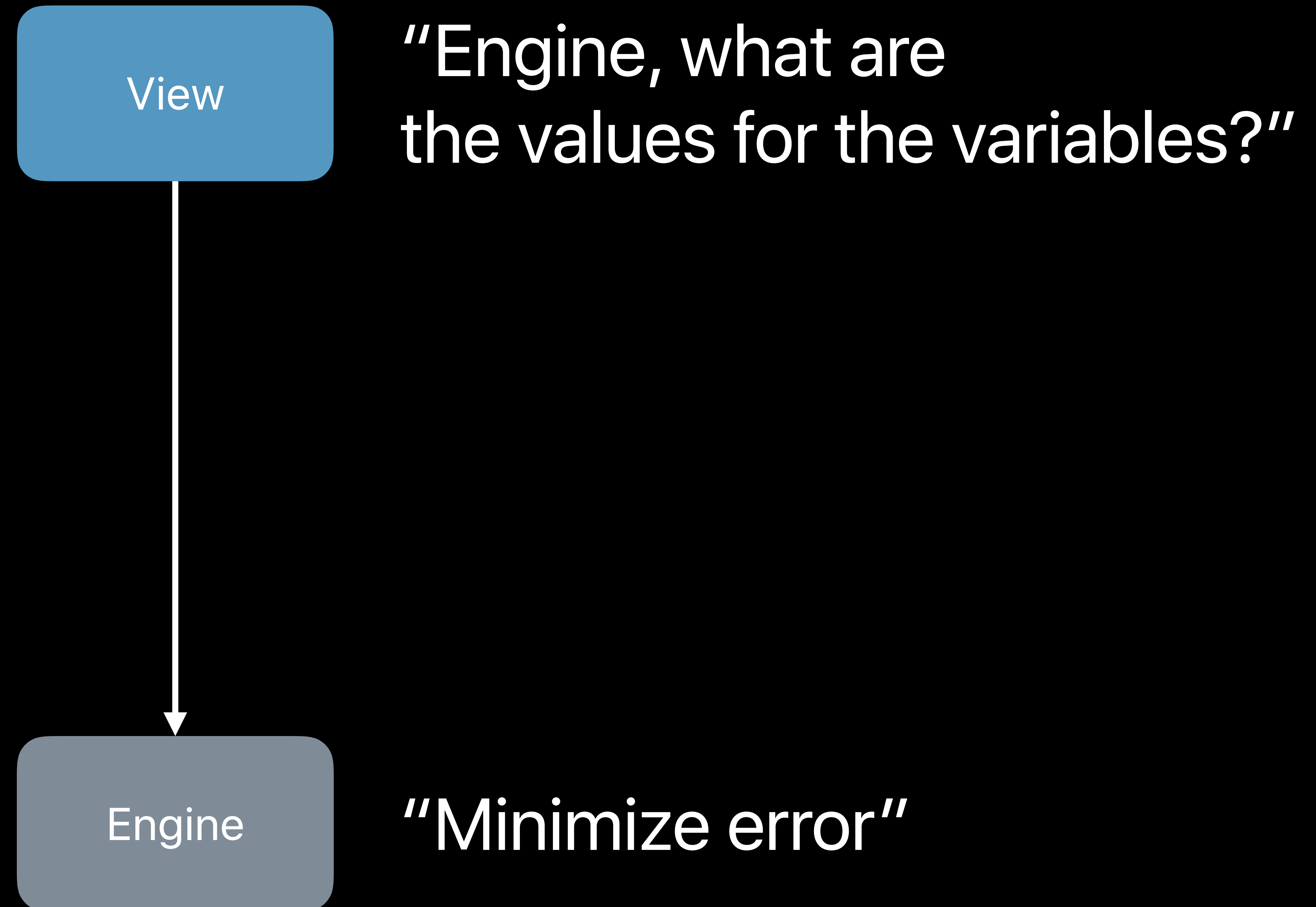
Constraints with non-required priority



“Engine, what are the values for the variables?”

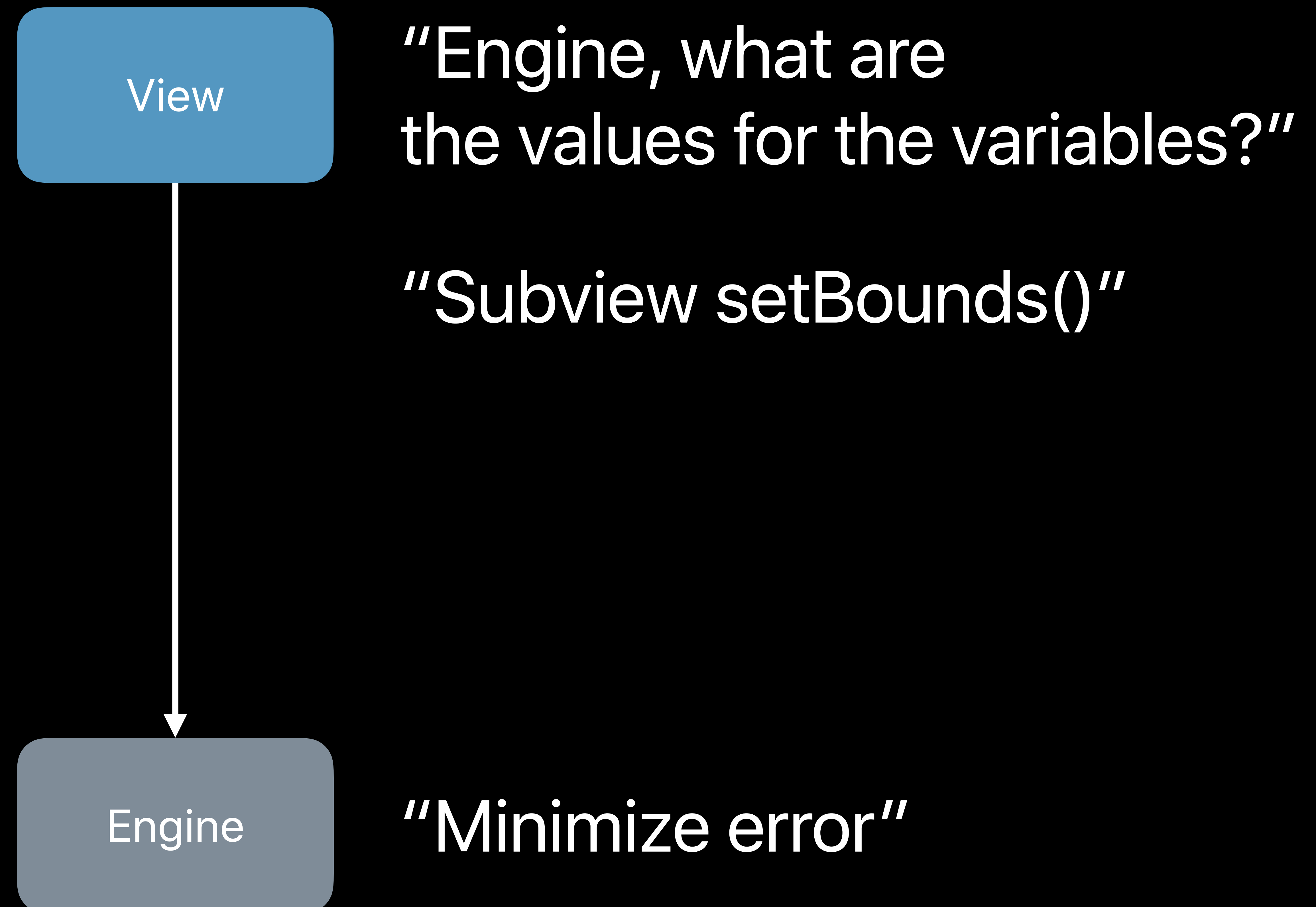
Error Minimization

Constraints with non-required priority



Error Minimization

Constraints with non-required priority



Performance Intuition

Don't churn

Basic algebra

Engine is a layout cache and tracker

You pay for what you use

iOS 12 improvements

Internals and intuition

Building efficient layouts

iOS 12 improvements

Internals and intuition

Building efficient layouts

#WWDC18

Building Efficient Layouts

Kasia Wawer, iOS Keyboards

Constraint Churn

New solution, same layout

Extra work happens with no visible result

Enough churn can affect performance

Building a Layout



Author

Title

June 6th, 2018

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Building a Layout

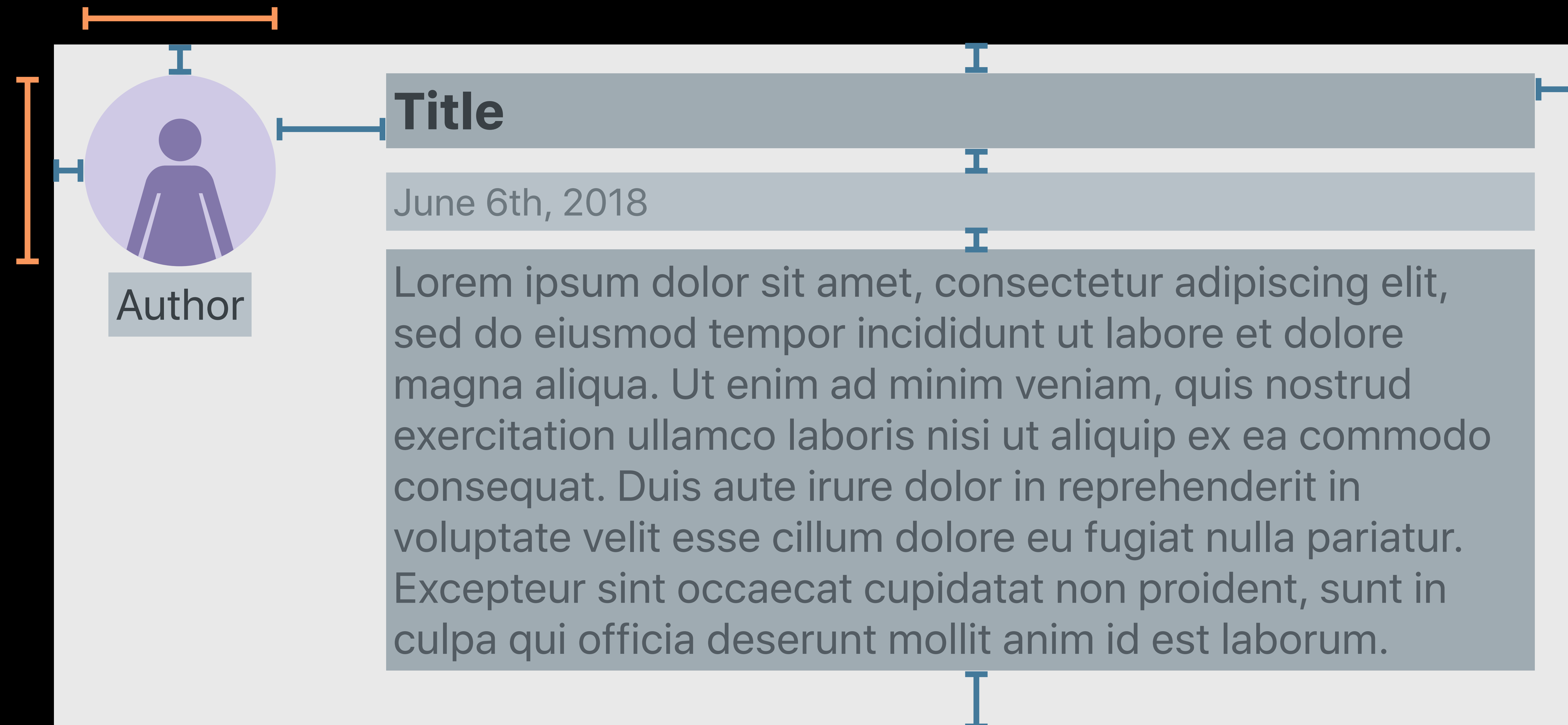
Author

Title

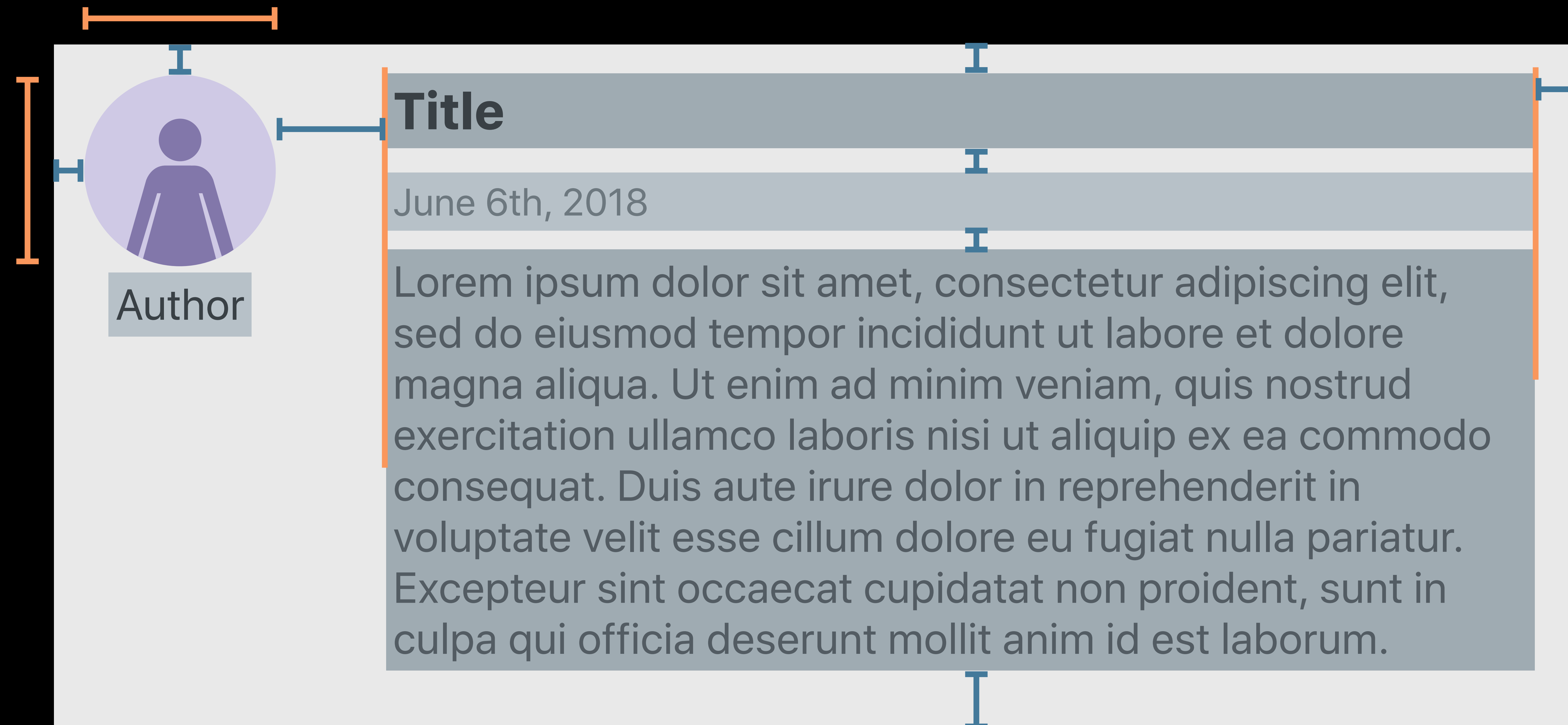
June 6th, 2018

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

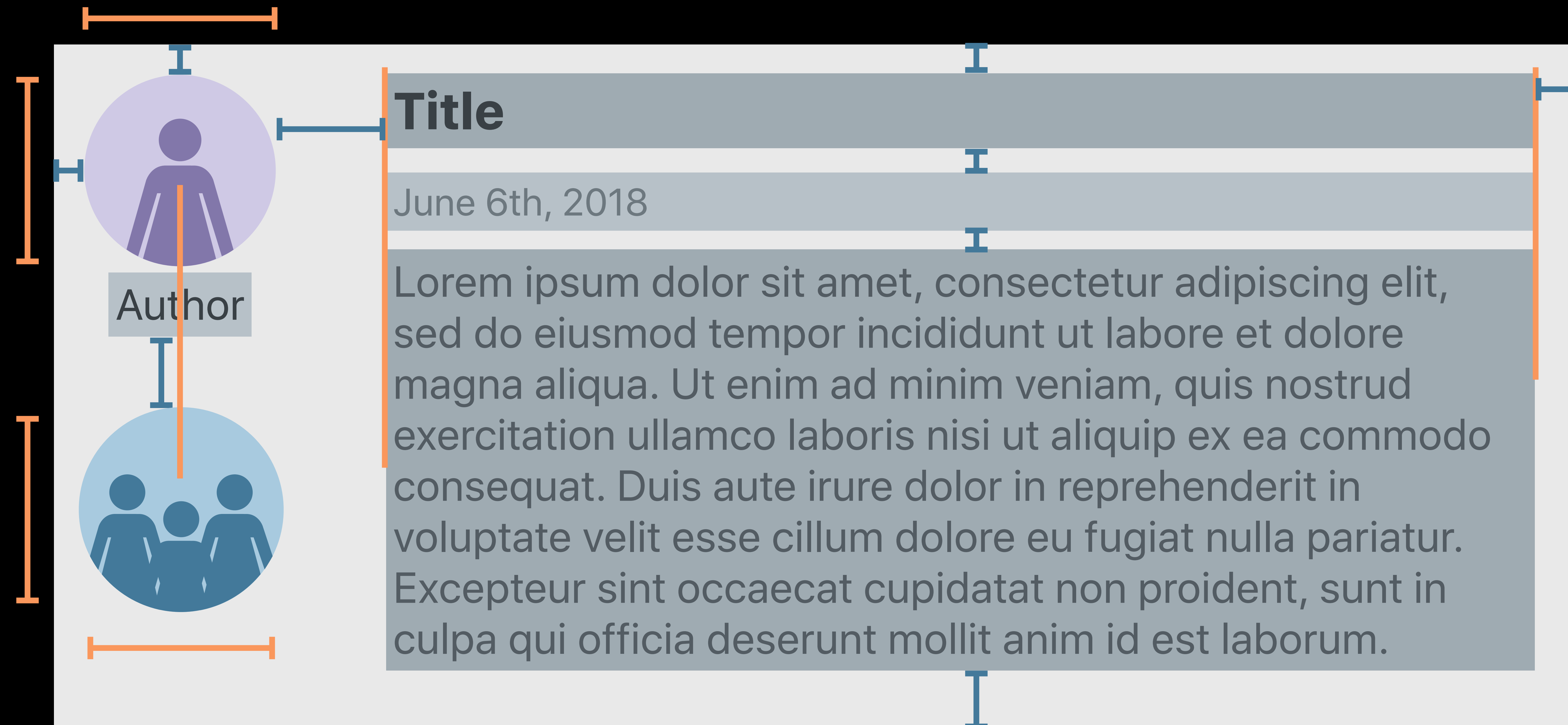
Building a Layout



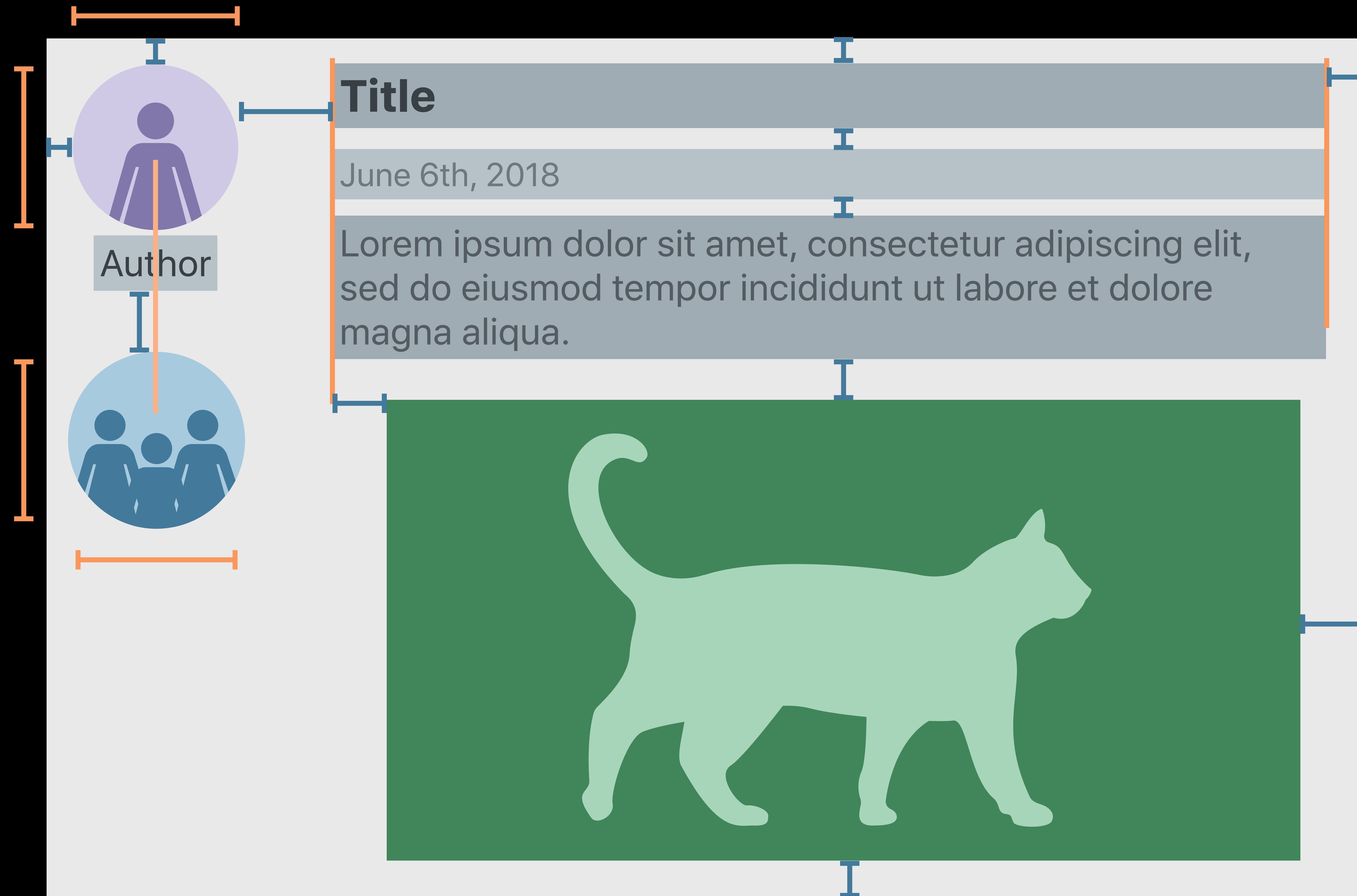
Building a Layout



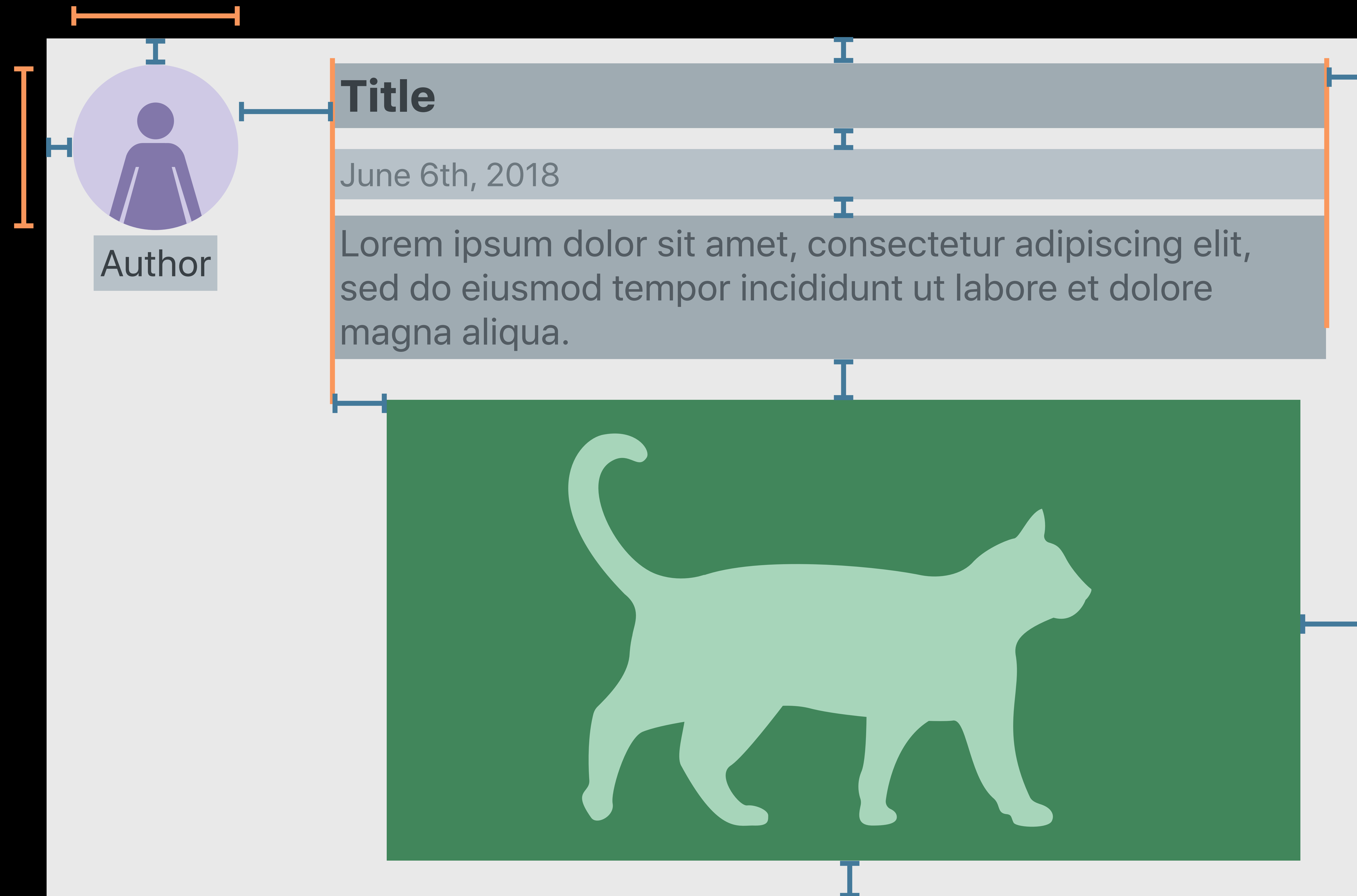
Building a Layout



Building a Layout



Building a Layout



Edit

Trip Log



Misty ocean

April 23, 2018



I love this view. Look at how the fog mists over the rocks. I think I saw a seal sitting on one of them, and definitely some gulls. I wish I had more time to just sit and ponder the water, but there's too much to do!



Dzisiejsza lekcja

April 23, 2018

Monica

Lubię wiele zwierząt. Koty są najlepsze.



Cat friend

April 24, 2018

Monica



I met the sweetest cat today! He was orange and absolutely huge. He was not willing to stick around while I got the

Edit

Trip Log



Misty ocean

April 23, 2018



I love this view. Look at how the fog mists over the rocks. I think I saw a seal sitting on one of them, and definitely some gulls. I wish I had more time to just sit and ponder the water, but there's too much to do!



Dzisiejsza lekcja

April 23, 2018

Monica

Lubię wiele zwierząt. Koty są najlepsze.



Cat friend

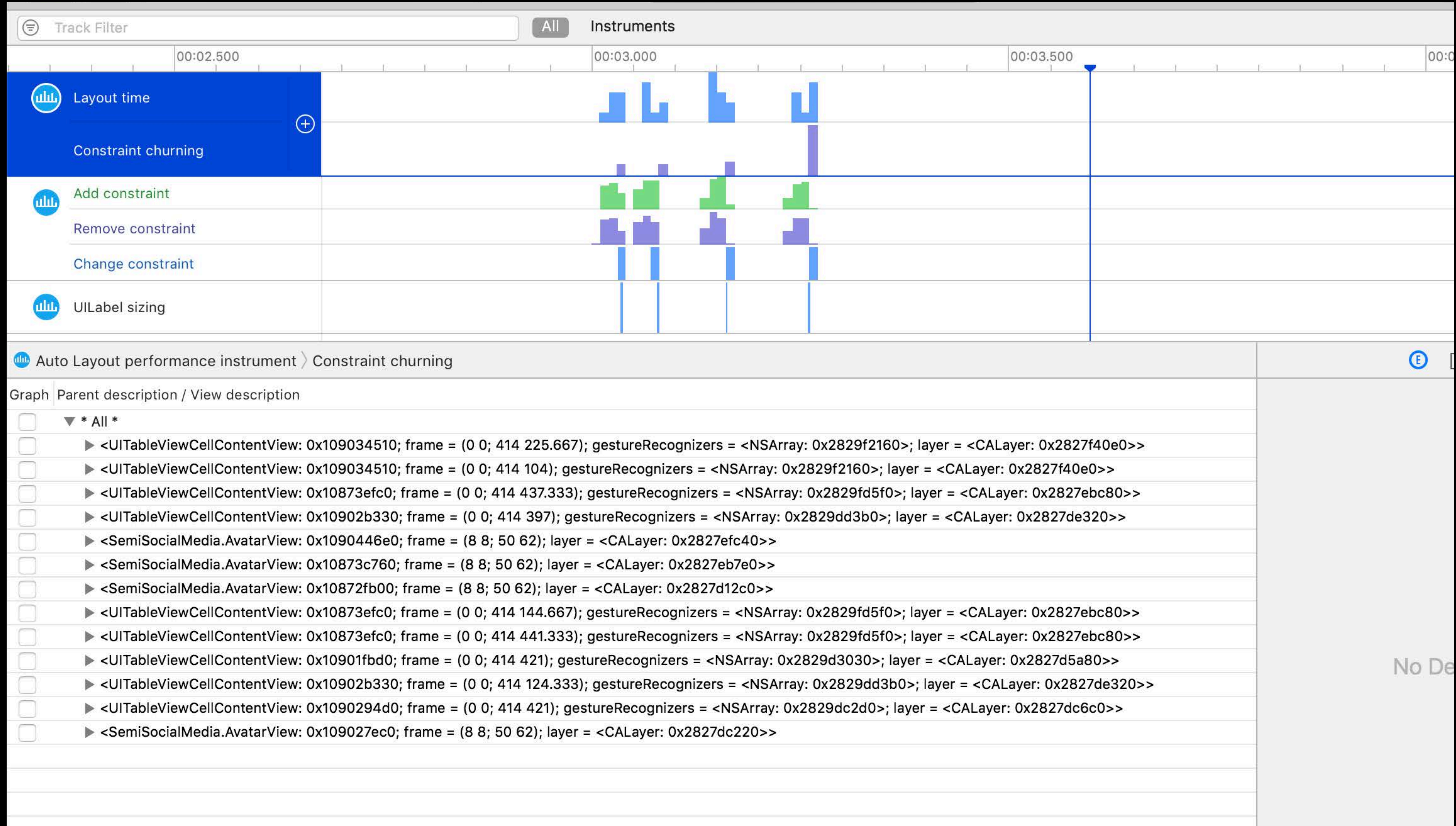
April 24, 2018

Monica

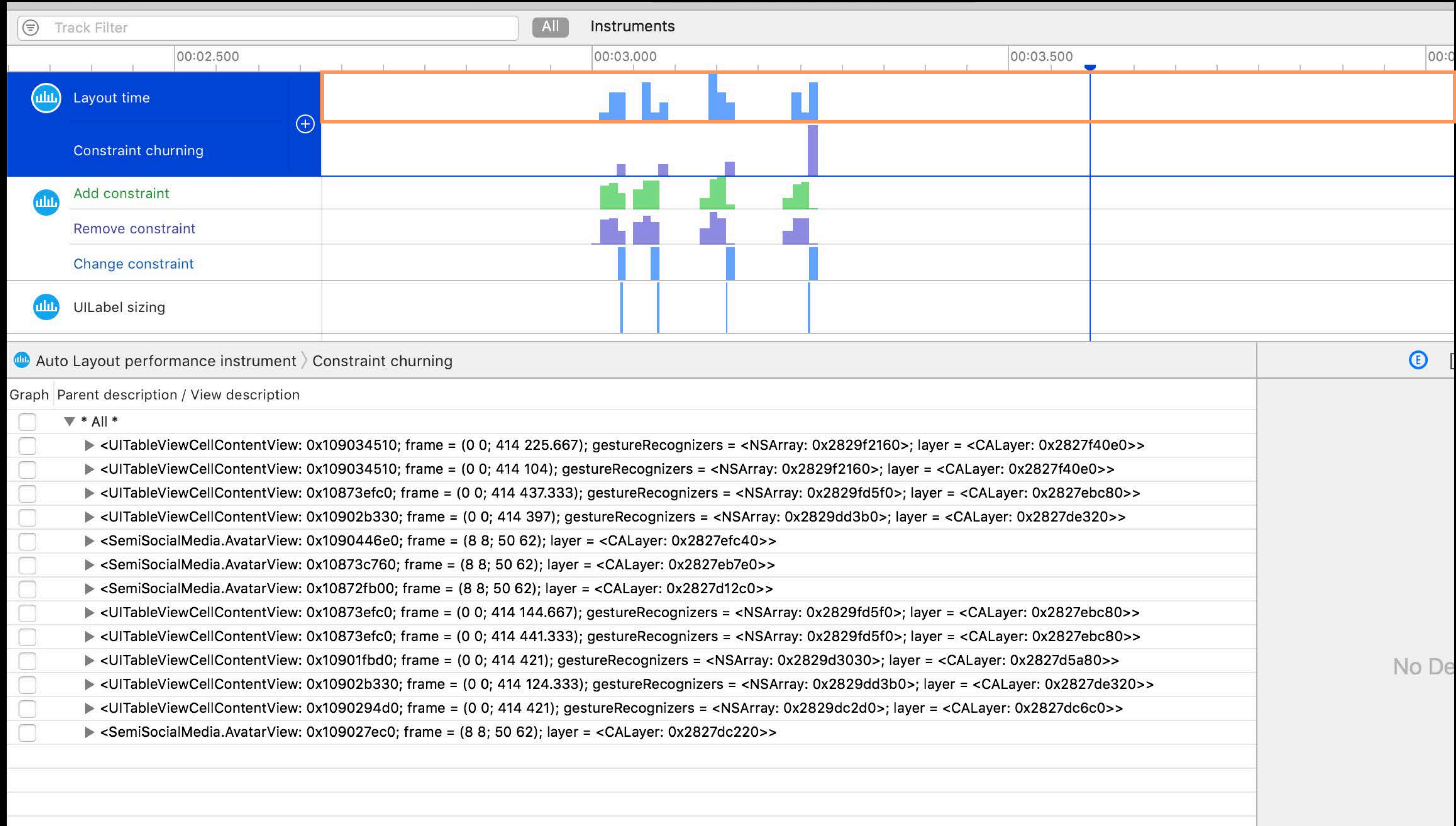


I met the sweetest cat today! He was orange and absolutely huge. He was not willing to stick around while I got the

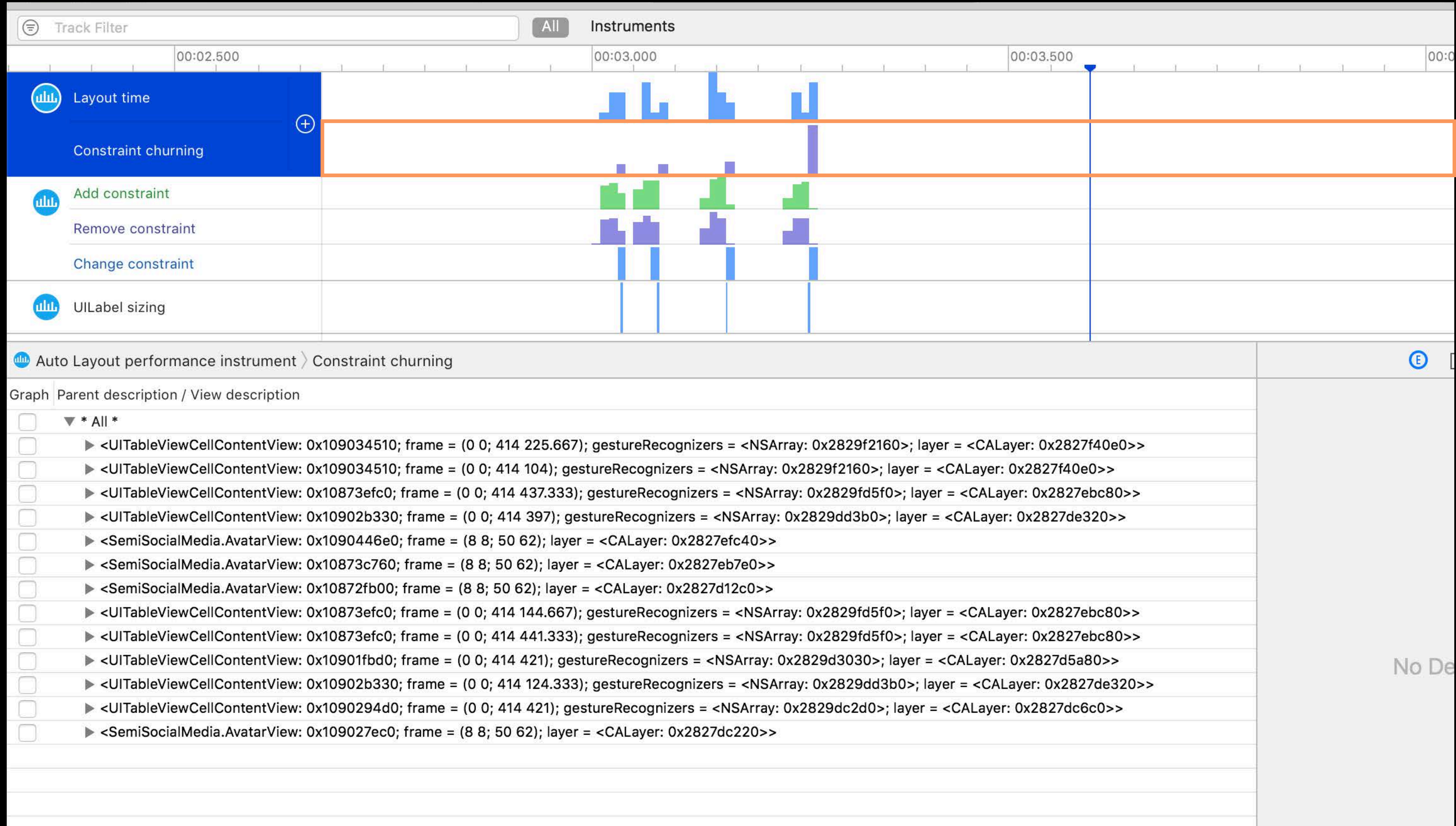
Finding the Problem



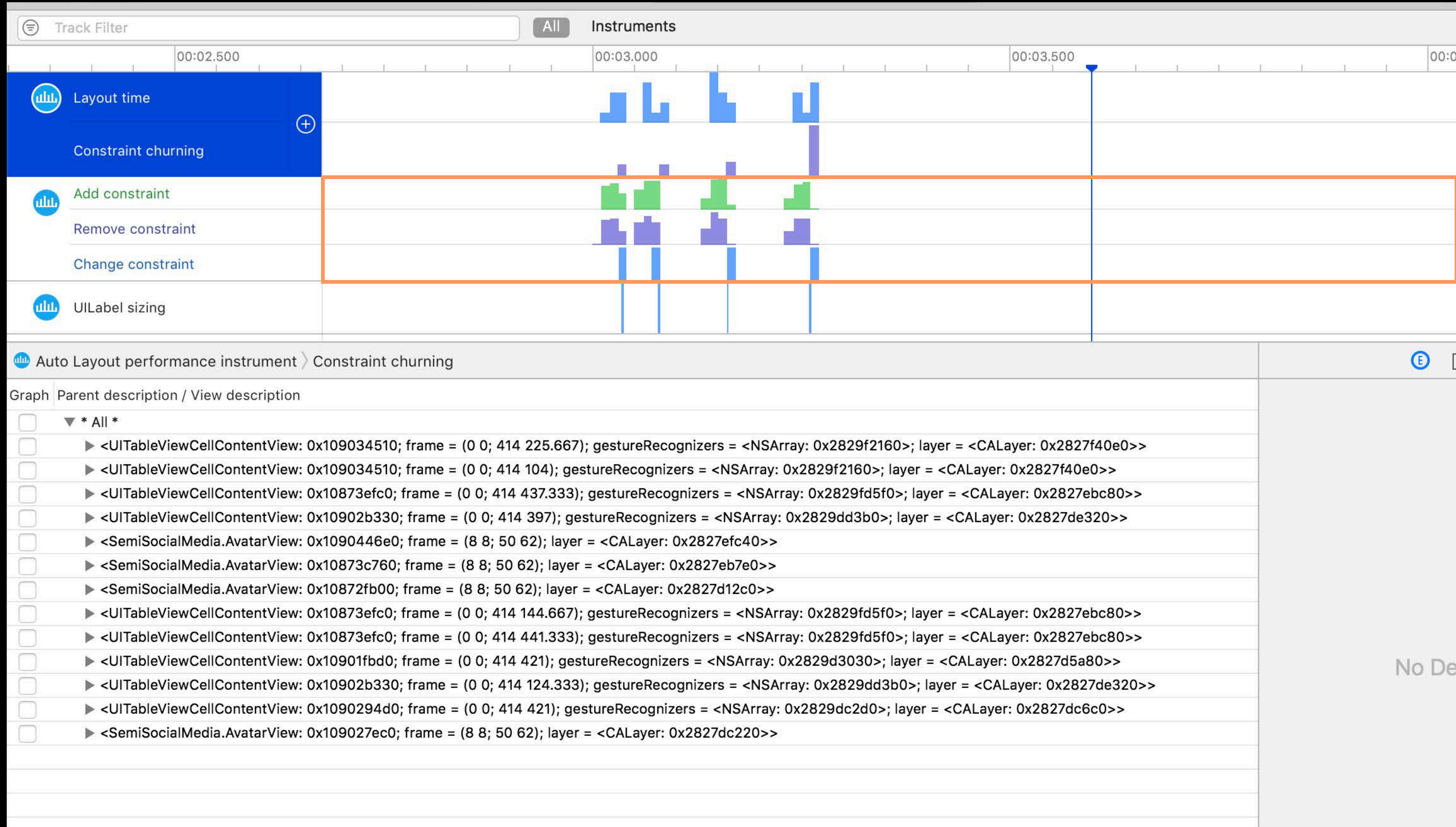
Finding the Problem



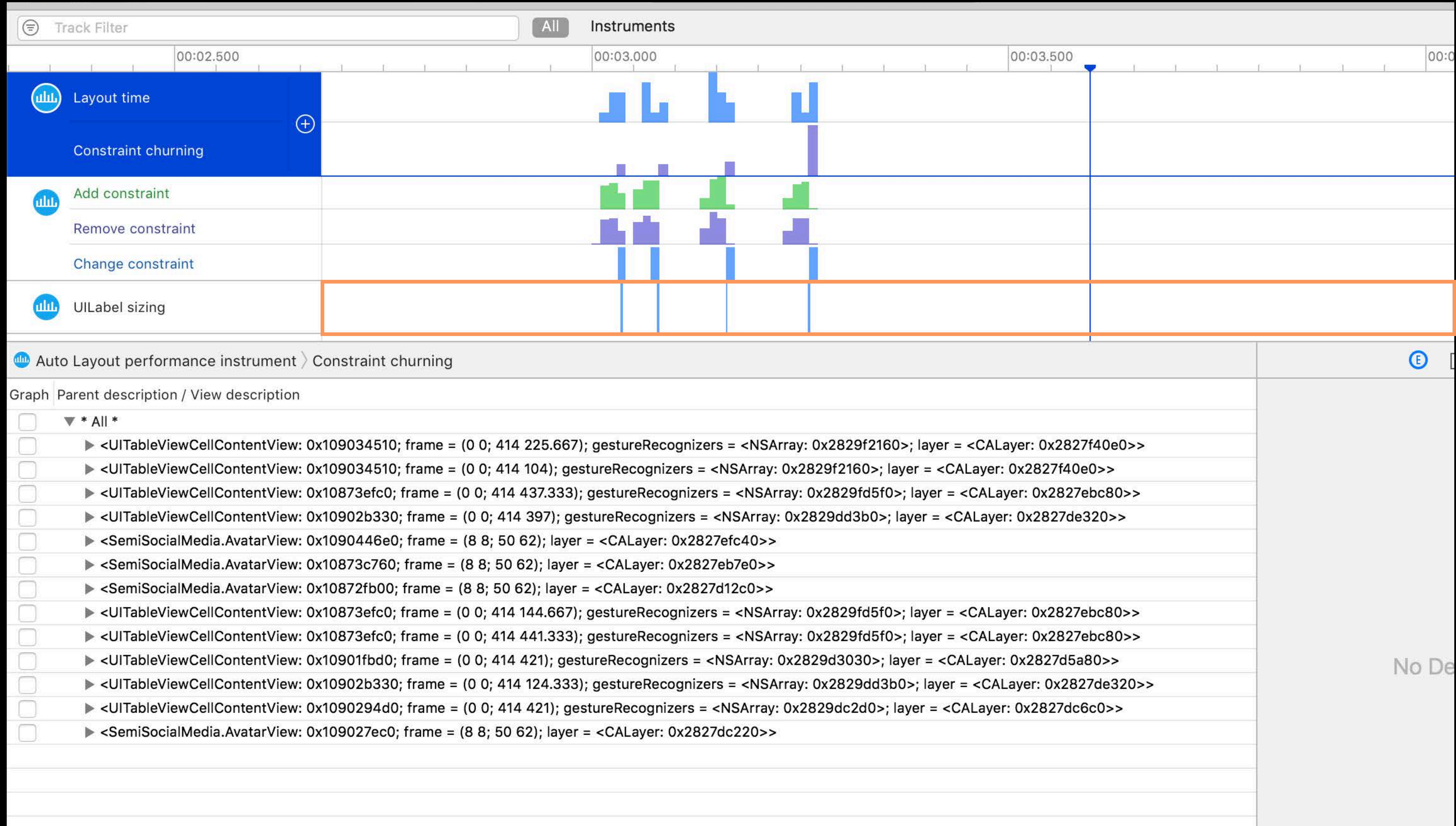
Finding the Problem



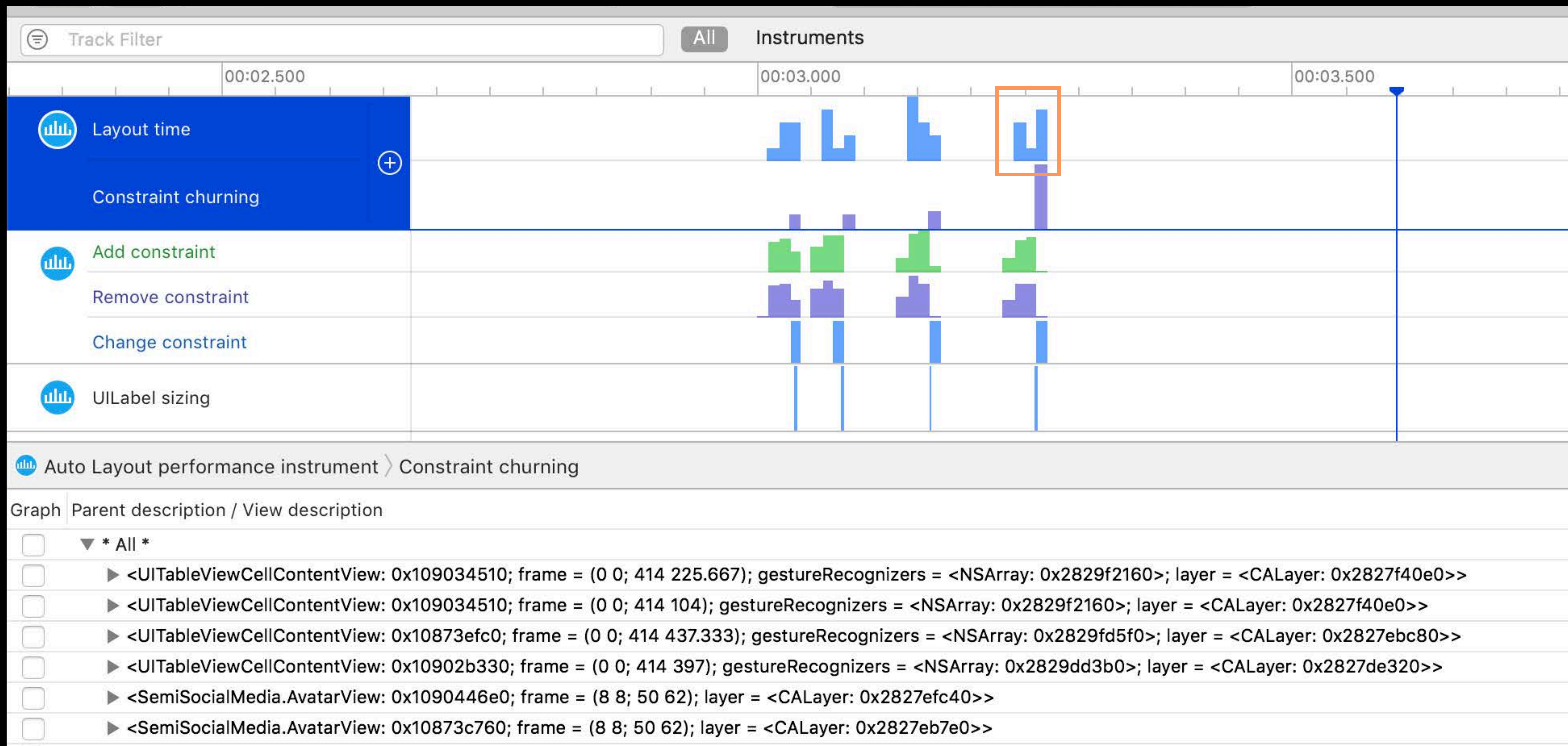
Finding the Problem



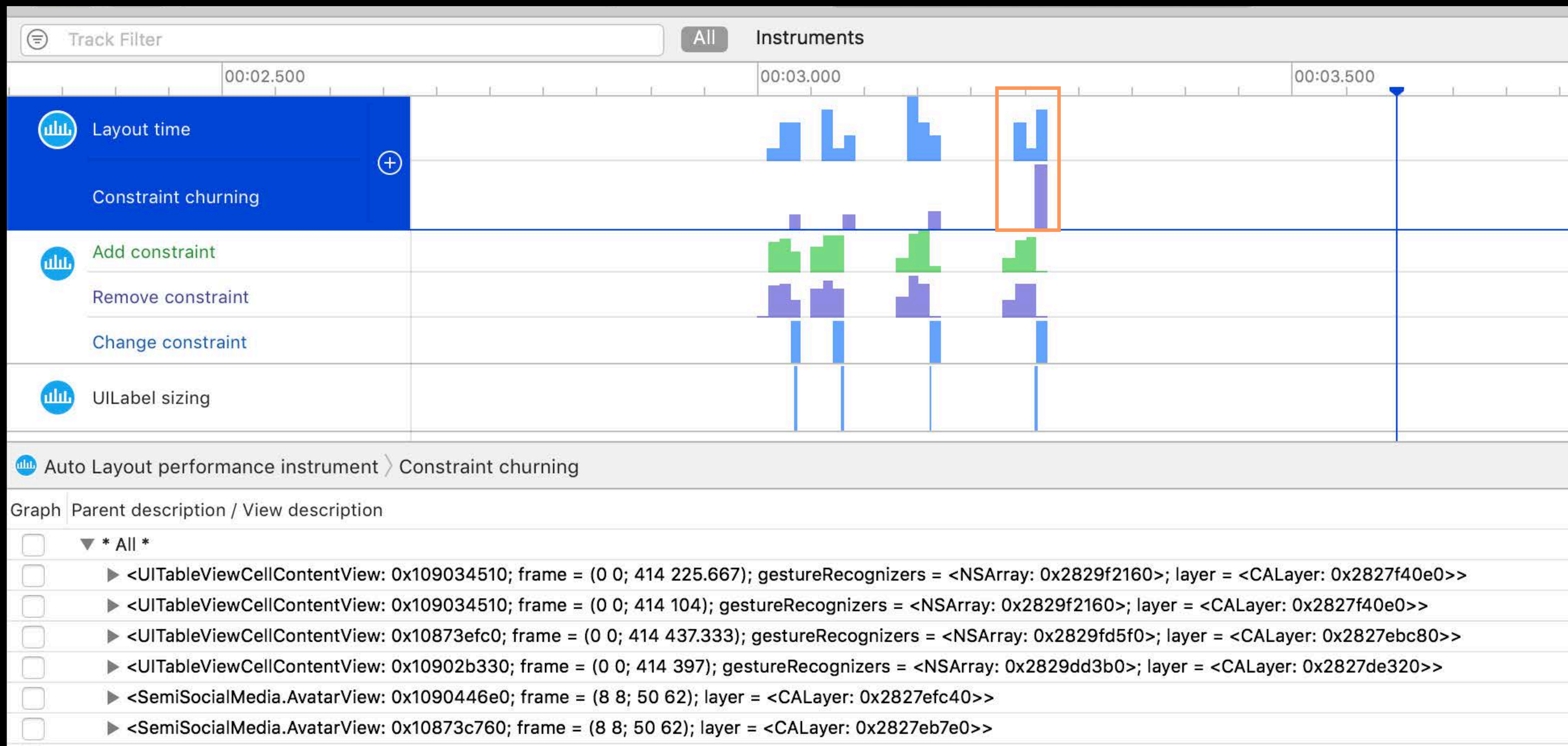
Finding the Problem



Finding the Problem



Finding the Problem



Finding the Problem

Auto Layout performance instrument > Constraint churning

Graph Parent description / View description

<input type="checkbox"/>	▼ * All *
<input type="checkbox"/>	▼ <UITableViewCellContentView: 0x10873efc0; frame = (0 0; 414 437.333); gestureRecognizers = <NSArray: 0x2829fd5f0>; layer = <CALayer: 0x2827ebc80>>
<input type="checkbox"/>	<UILabel: 0x10873e7b0; frame = (74 8; 324 17); text = 'Quite a hike'; userInteractionEnabled = NO; layer = <UILabelLayer: 0x2804b3e80>>
<input type="checkbox"/>	<UILabel: 0x10873eaa0; frame = (74 55.3333; 324 81.3333); text = 'Hiked up the steps to the...'; userInteractionEnabled = NO; layer = <UILabelLayer: 0x2804b8050>>
<input type="checkbox"/>	<SemiSocialMedia.AvatarView: 0x10873c760; frame = (8 8; 50 62); layer = <CALayer: 0x2827eb7e0>>
<input type="checkbox"/>	<UILabel: 0x108737f70; frame = (74 33; 324 14.3333); text = 'May 25, 2018'; userInteractionEnabled = NO; layer = <UILabelLayer: 0x2804b3f70>>
<input type="checkbox"/>	▼ <SemiSocialMedia.AvatarView: 0x10873c760; frame = (8 8; 50 62); layer = <CALayer: 0x2827eb7e0>>
<input type="checkbox"/>	<UILabel: 0x10873e290; frame = (0 50; 50 12); text = 'Monica'; userInteractionEnabled = NO; layer = <UILabelLayer: 0x2804b2df0>>
<input type="checkbox"/>	<UIImageView: 0x10873e060; frame = (0 0; 50 50); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x2827eb660>>

Finding the Problem

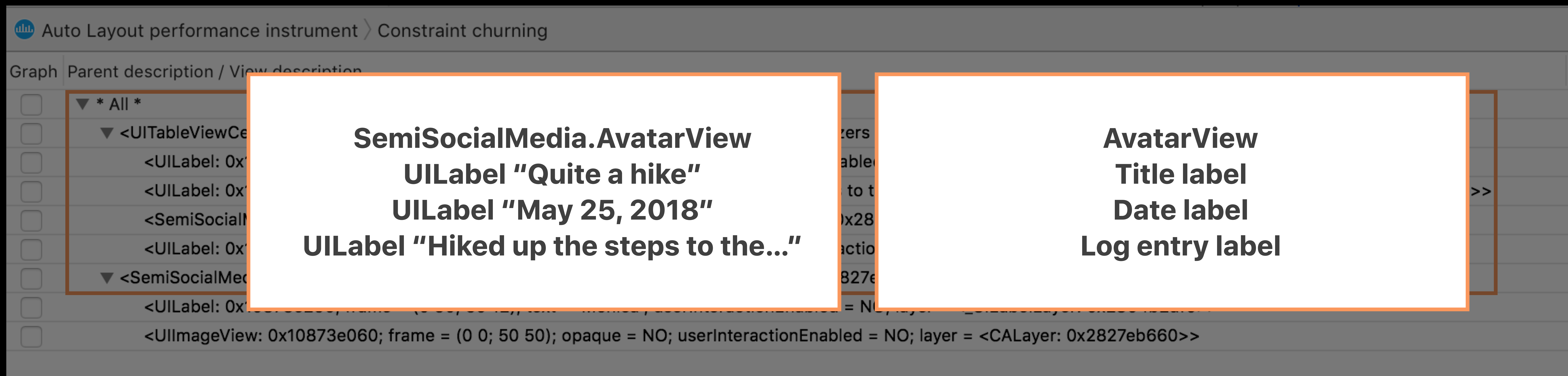
Auto Layout performance instrument > Constraint churning

Graph Parent description / View description

- ▼ * All *
- ▼ <UITableViewCellContentView: 0x10873efc0; frame = (0 0; 414 437.333); gestureRecognizers
- <UILabel: 0x10873e7b0; frame = (74 8; 324 17); text = 'Quite a hike'; userInteractionEnabled
- <UILabel: 0x10873eaa0; frame = (74 55.3333; 324 81.3333); text = 'Hiked up the steps to t
- <SemiSocialMedia.AvatarView: 0x10873c760; frame = (8 8; 50 62); layer = <CALayer: 0x28
- <UILabel: 0x108737f70; frame = (74 33; 324 14.3333); text = 'May 25, 2018'; userInteractio
- ▼ <SemiSocialMedia.AvatarView: 0x10873c760; frame = (8 8; 50 62); layer = <CALayer: 0x2827e
- <UILabel: 0x10873e290; frame = (0 50; 50 12); text = 'Monica'; userInteractionEnabled = NO
- <UIImageView: 0x10873e060; frame = (0 0; 50 50); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x2827eb660>>

SemiSocialMedia.AvatarView
UILabel "Quite a hike"
UILabel "May 25, 2018"
UILabel "Hiked up the steps to the..."

Finding the Problem



Finding the Problem

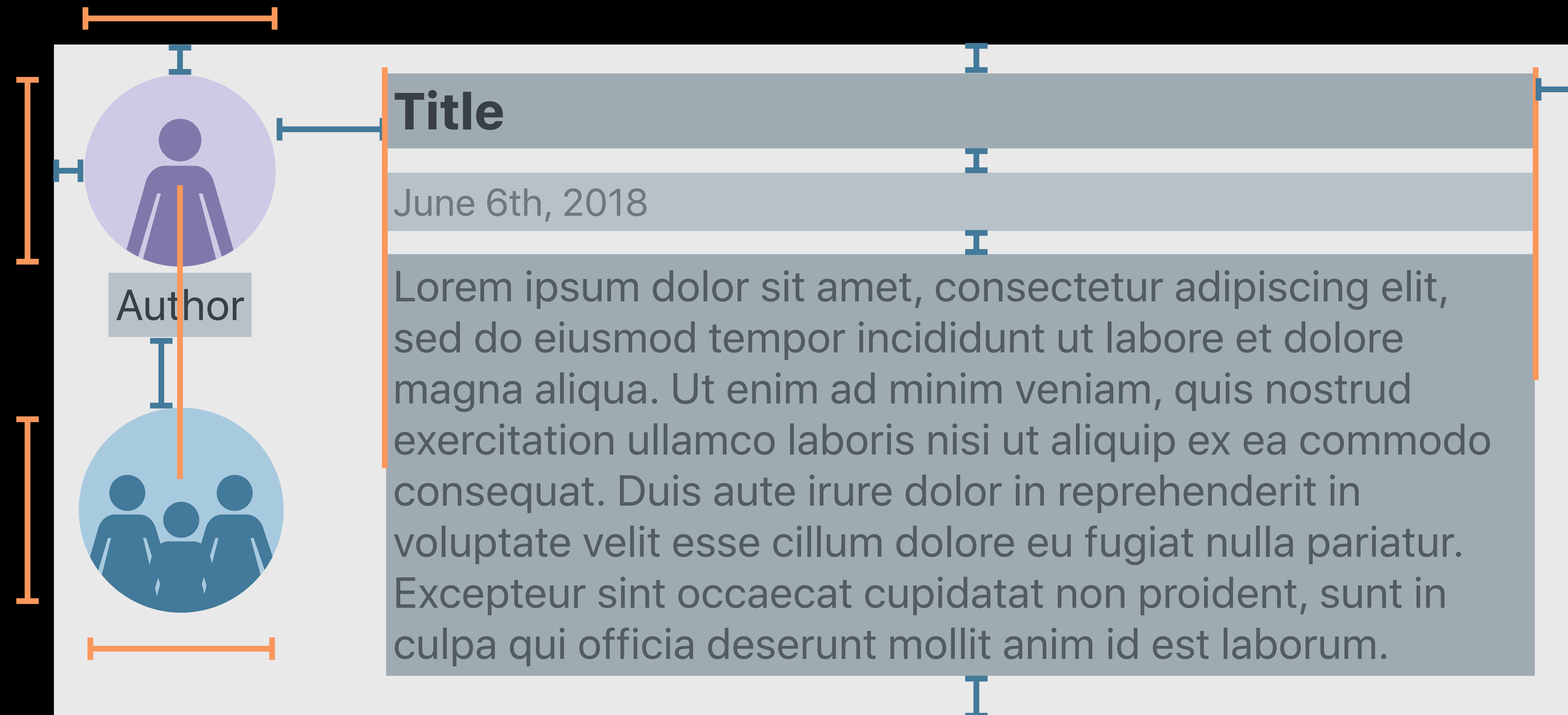
Auto Layout performance instrument > Constraint churning

Graph Parent description / View description

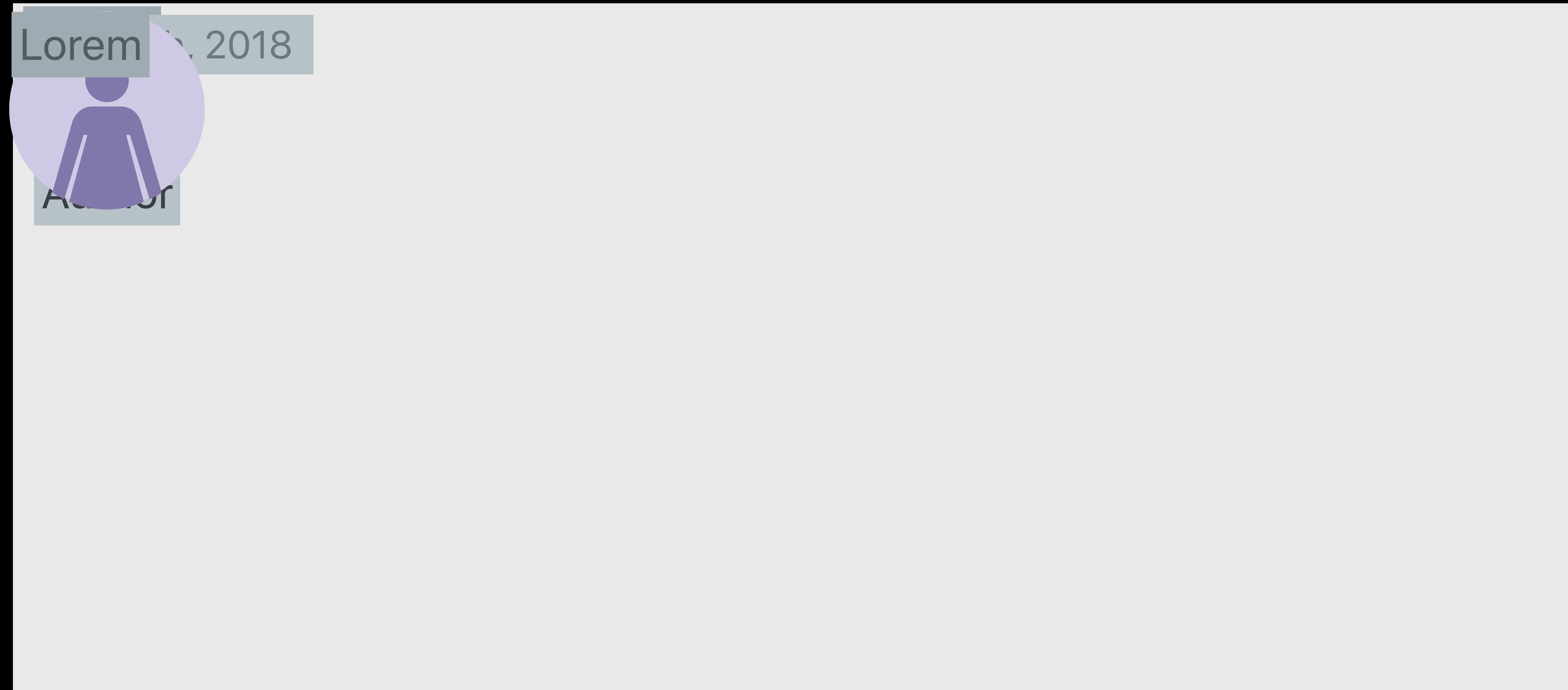
<input type="checkbox"/>	▼ * All *	
<input type="checkbox"/>	▼ <UITableViewCellContentView: 0x10873efc0	f0>; layer = <CALayer: 0x2827ebc80>>
<input type="checkbox"/>	<UILabel: 0x10873e7b0; frame = (74 8; 3	elLayer: 0x2804b3e80>>
<input type="checkbox"/>	<UILabel: 0x10873eaa0; frame = (74 55.3	abled = NO; layer = <UILabelLayer: 0x2804b8050>>
<input type="checkbox"/>	<SemiSocialMedia.AvatarView: 0x10873c	
<input type="checkbox"/>	<UILabel: 0x108737f70; frame = (74 33;	<UILabelLayer: 0x2804b3f70>>
<input type="checkbox"/>	▼ <SemiSocialMedia.AvatarView: 0x10873c76	
<input type="checkbox"/>	<UILabel: 0x10873e290; frame = (0 50; 5	er: 0x2804b2df0>>
<input type="checkbox"/>	<UIImageView: 0x10873e060; frame = (0 0; 50 50); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x2827eb660>>	

AvatarView
Title label
Date label
Log entry label

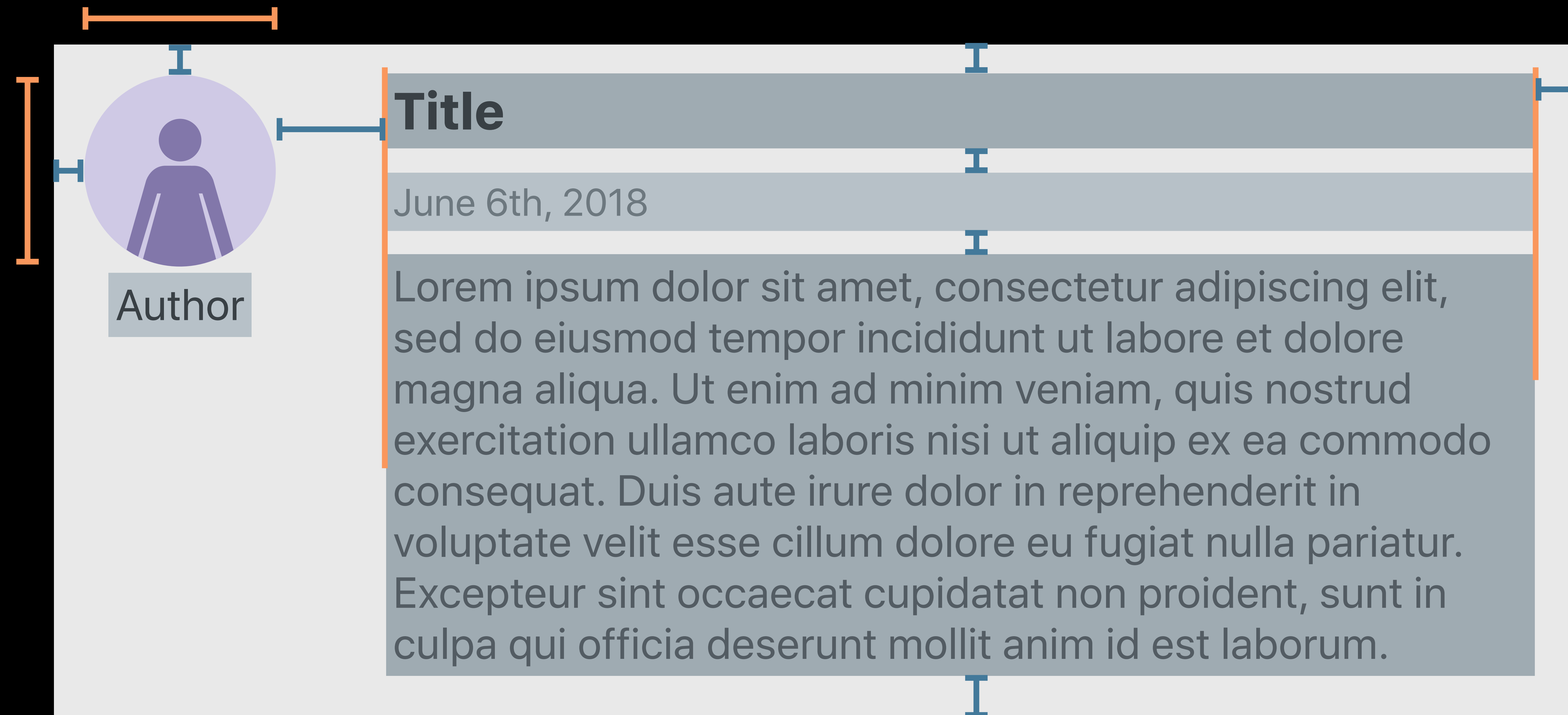
Building a More Performant Layout



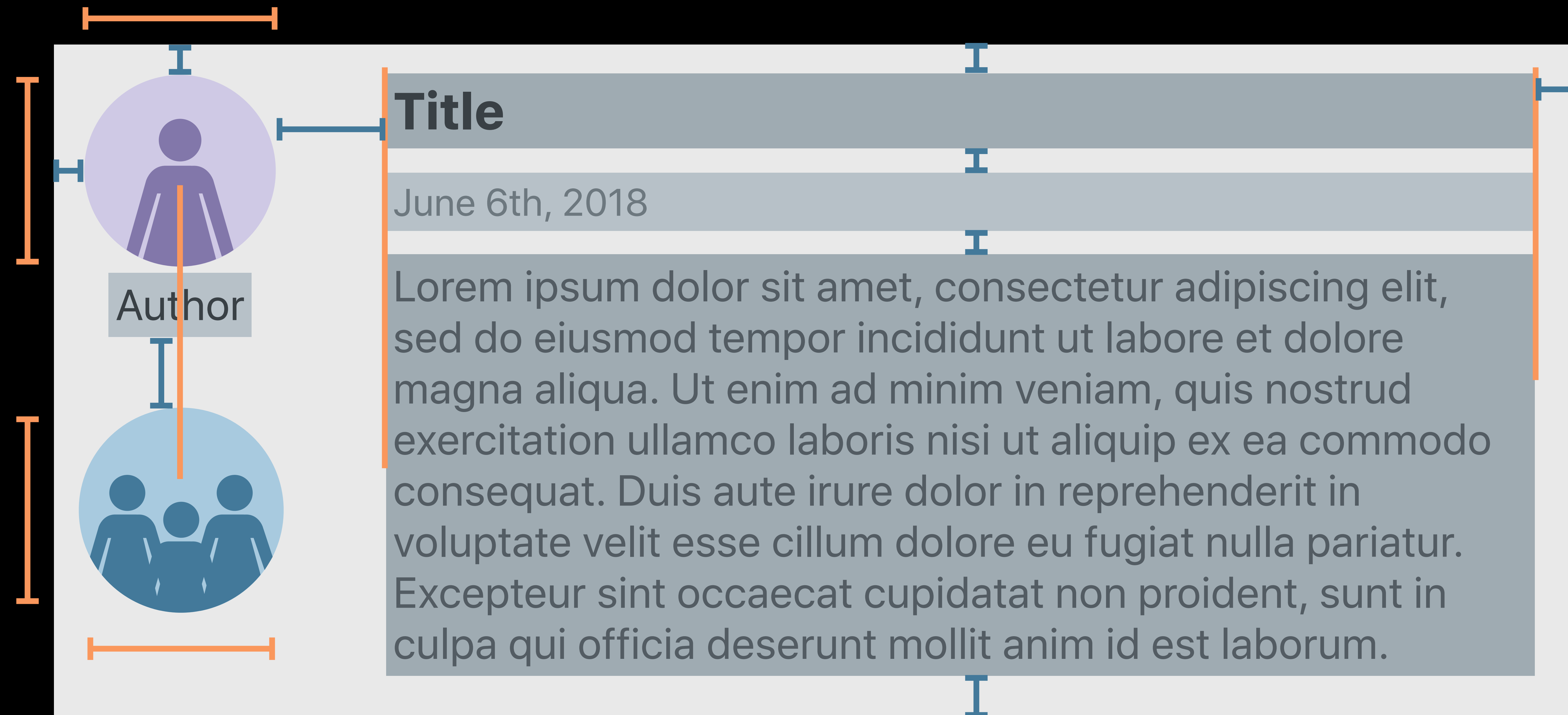
Building a More Performant Layout



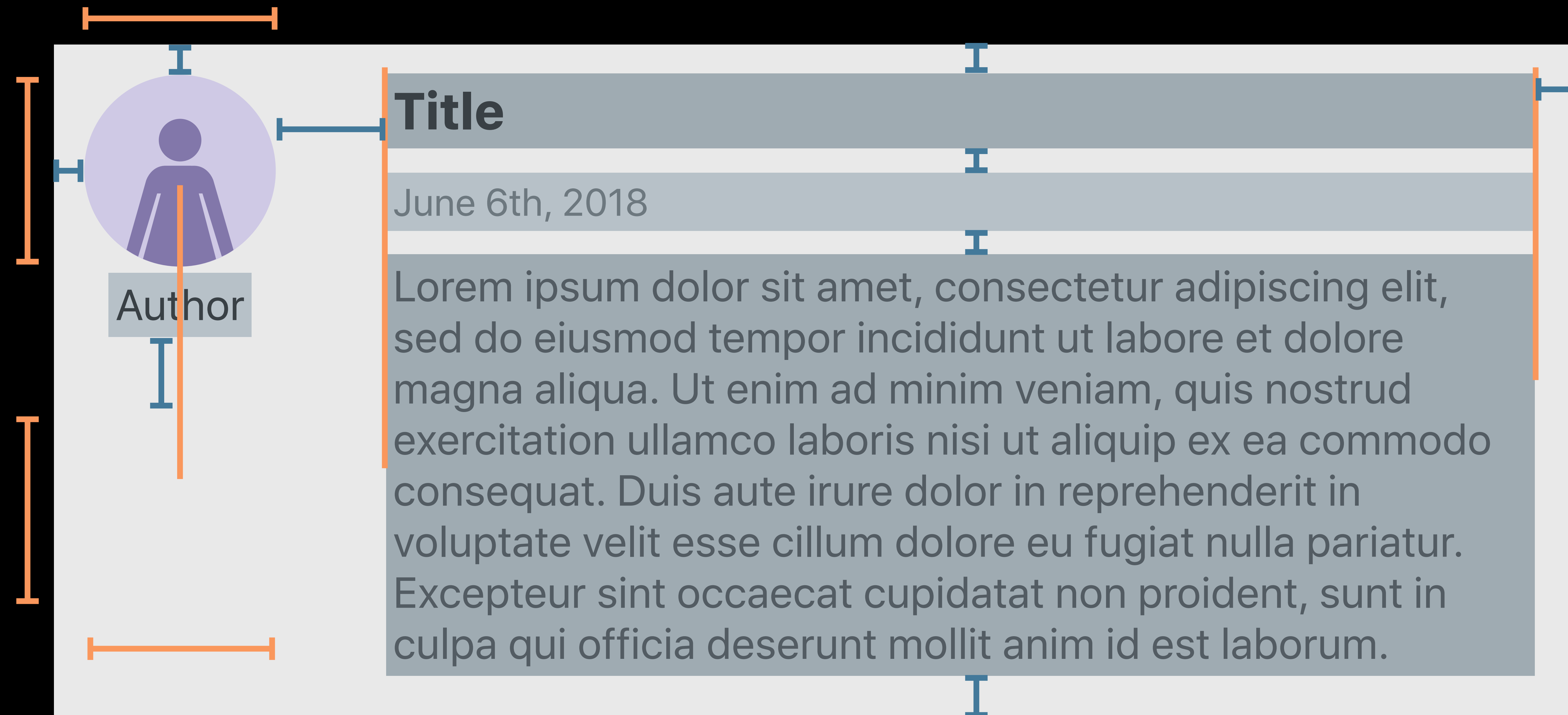
Building a More Performant Layout



Building a More Performant Layout



Building a More Performant Layout



Building a More Performant Layout



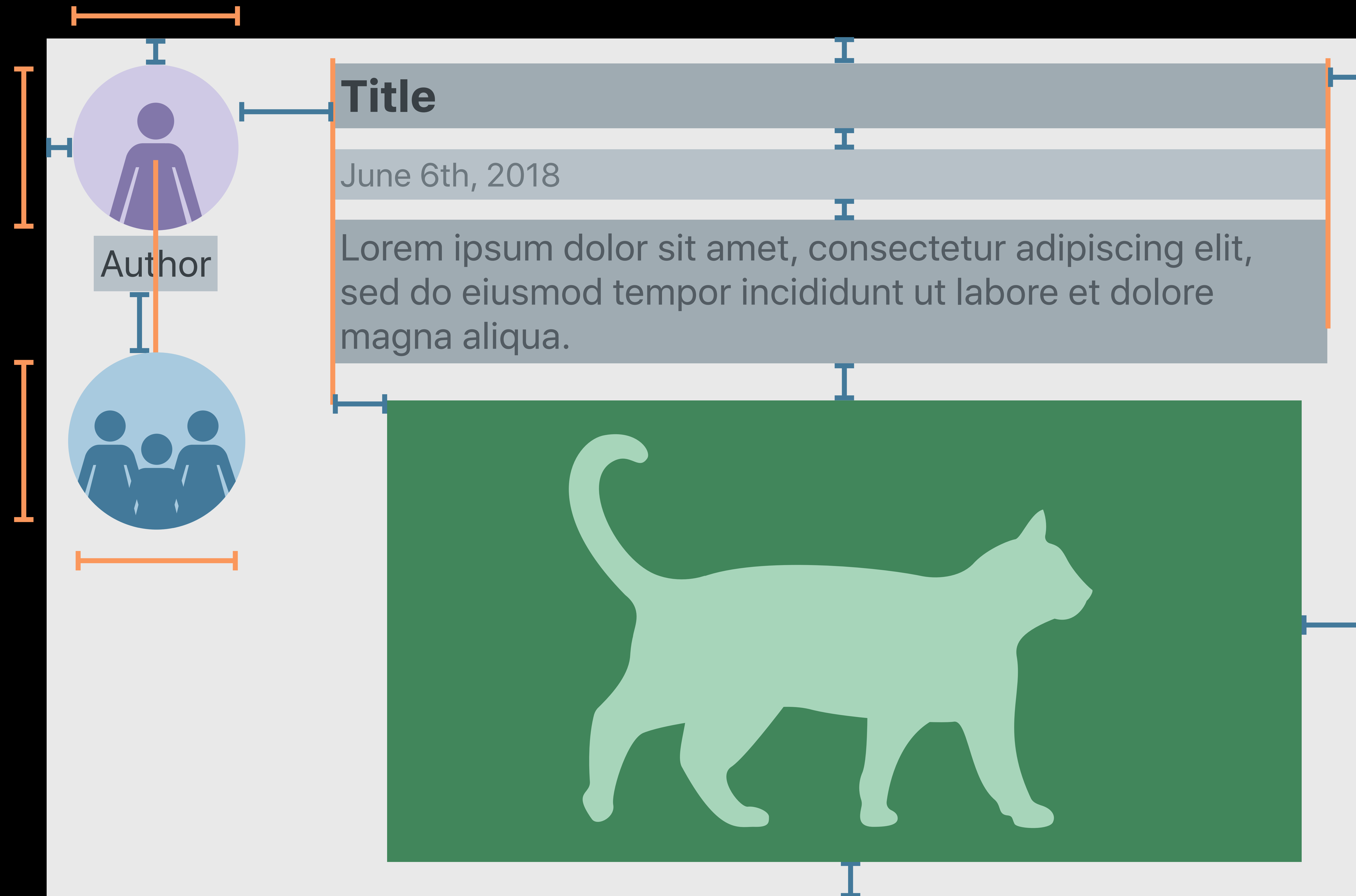
Author

Title

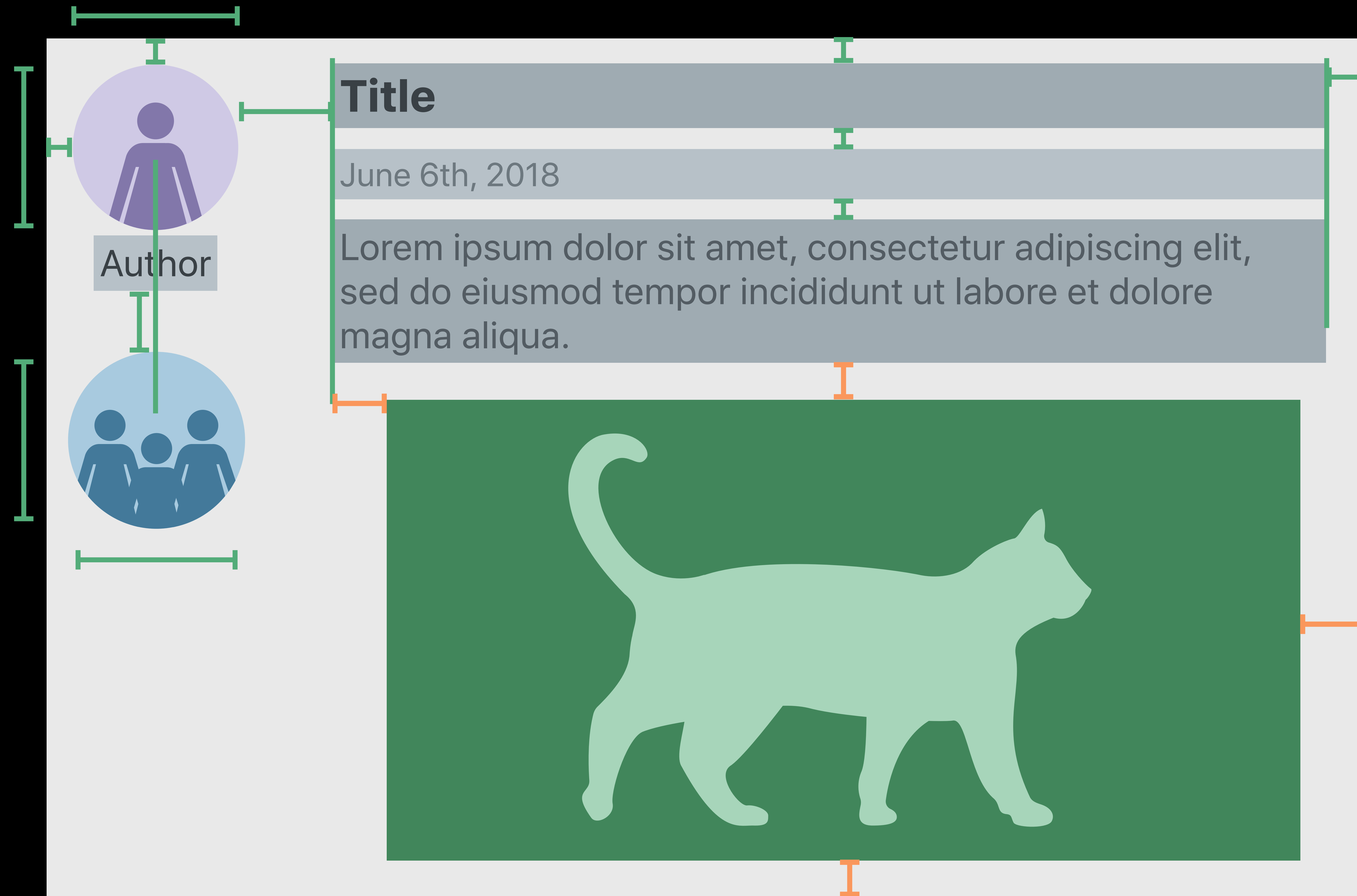
June 6th, 2018

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

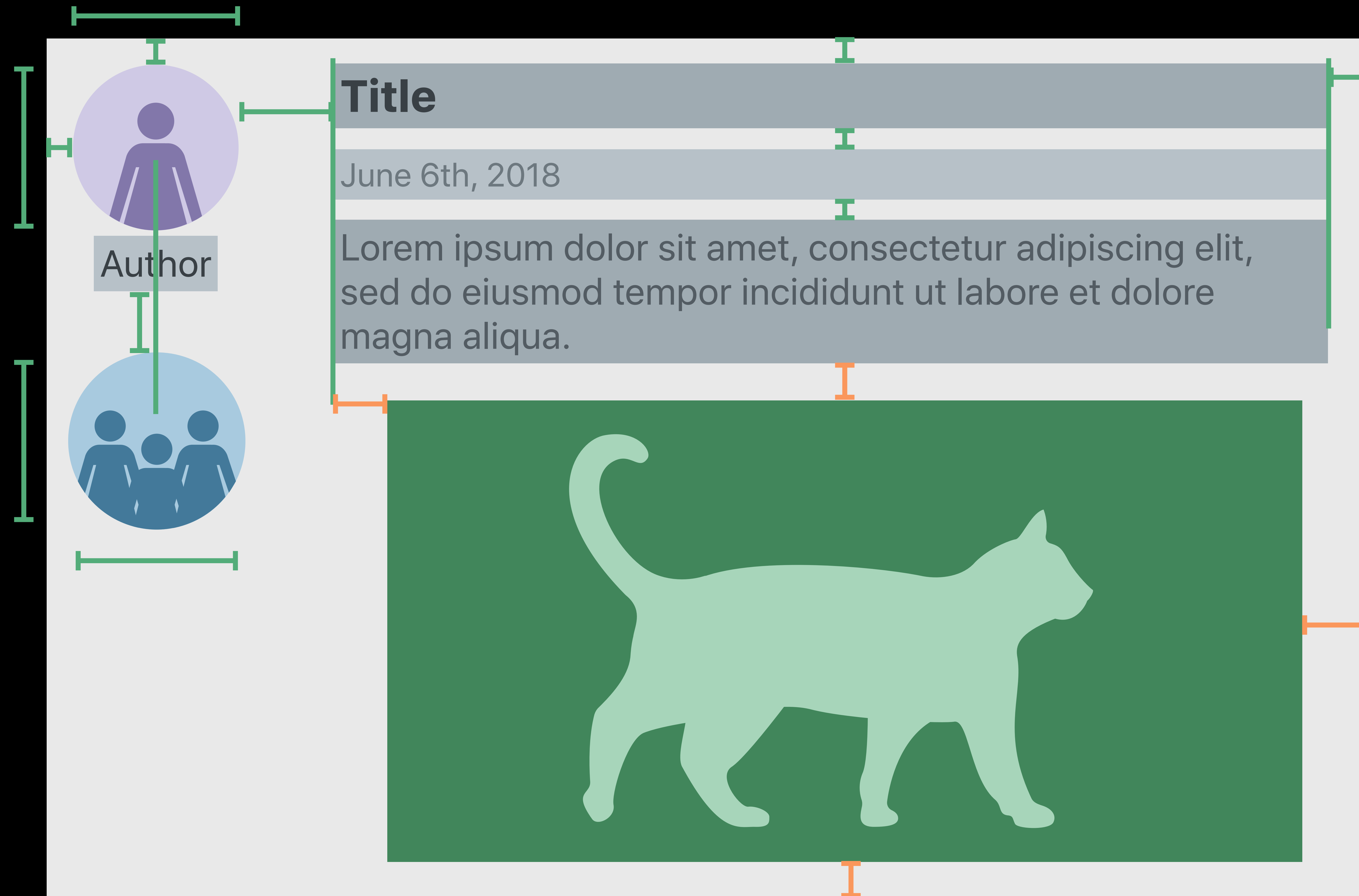
Building a More Performant Layout



Building a More Performant Layout



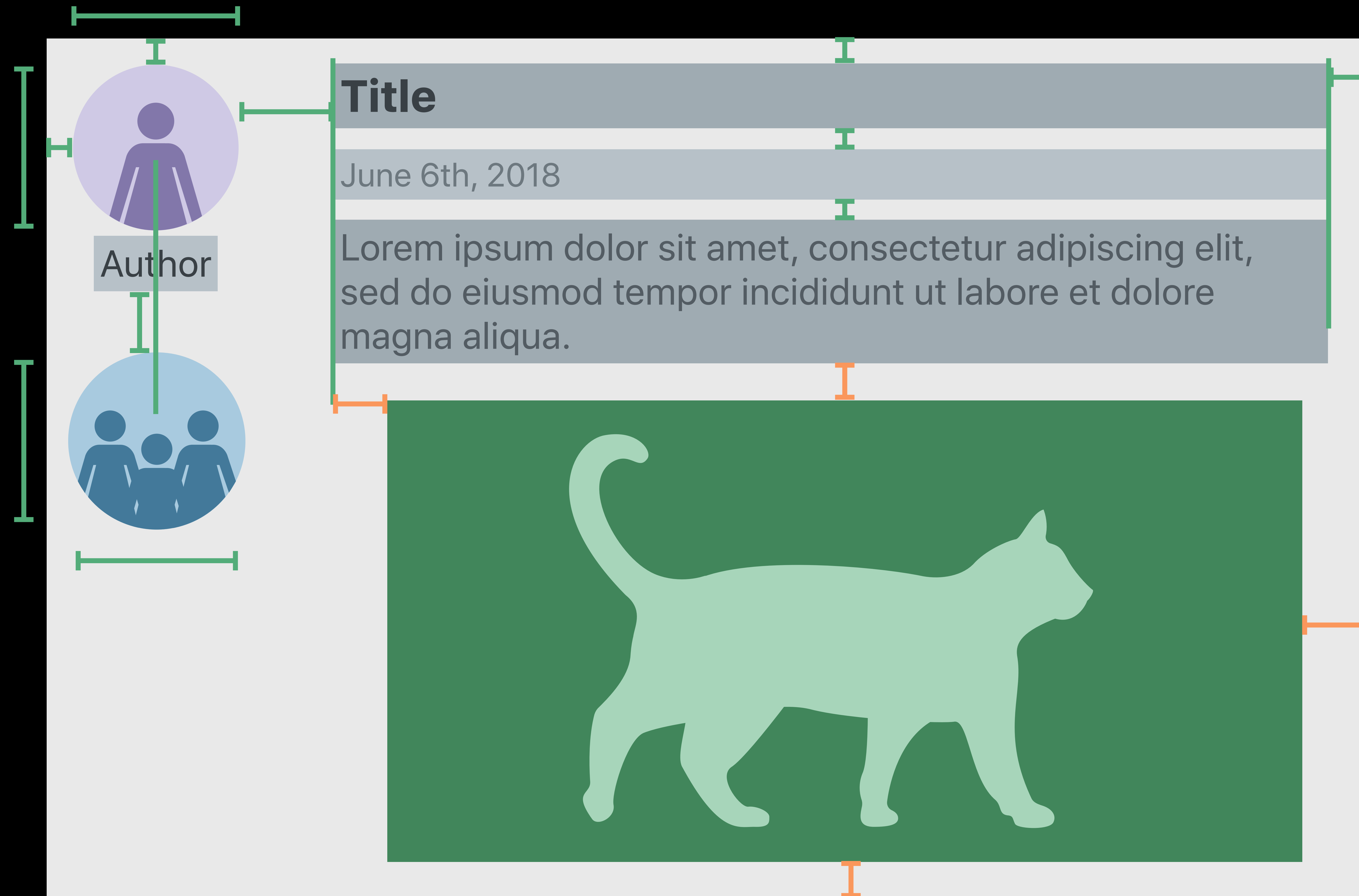
Building a More Performant Layout



imageConstraints =

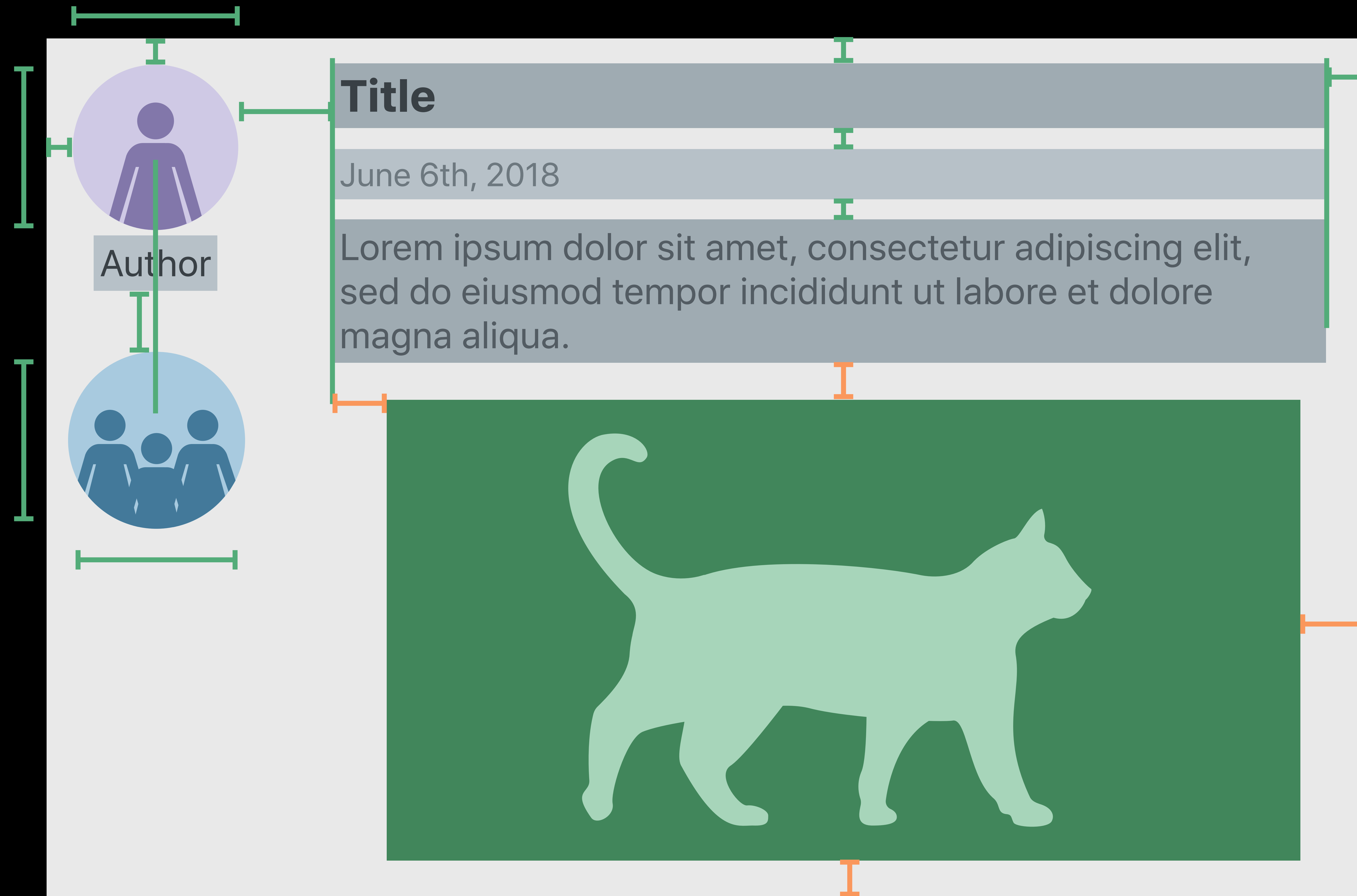
┌─┐ ┌─┐ ┌─┐ ┌─┐

Building a More Performant Layout



```
imageConstraints =  
[ , I, , I ]
```

Building a More Performant Layout



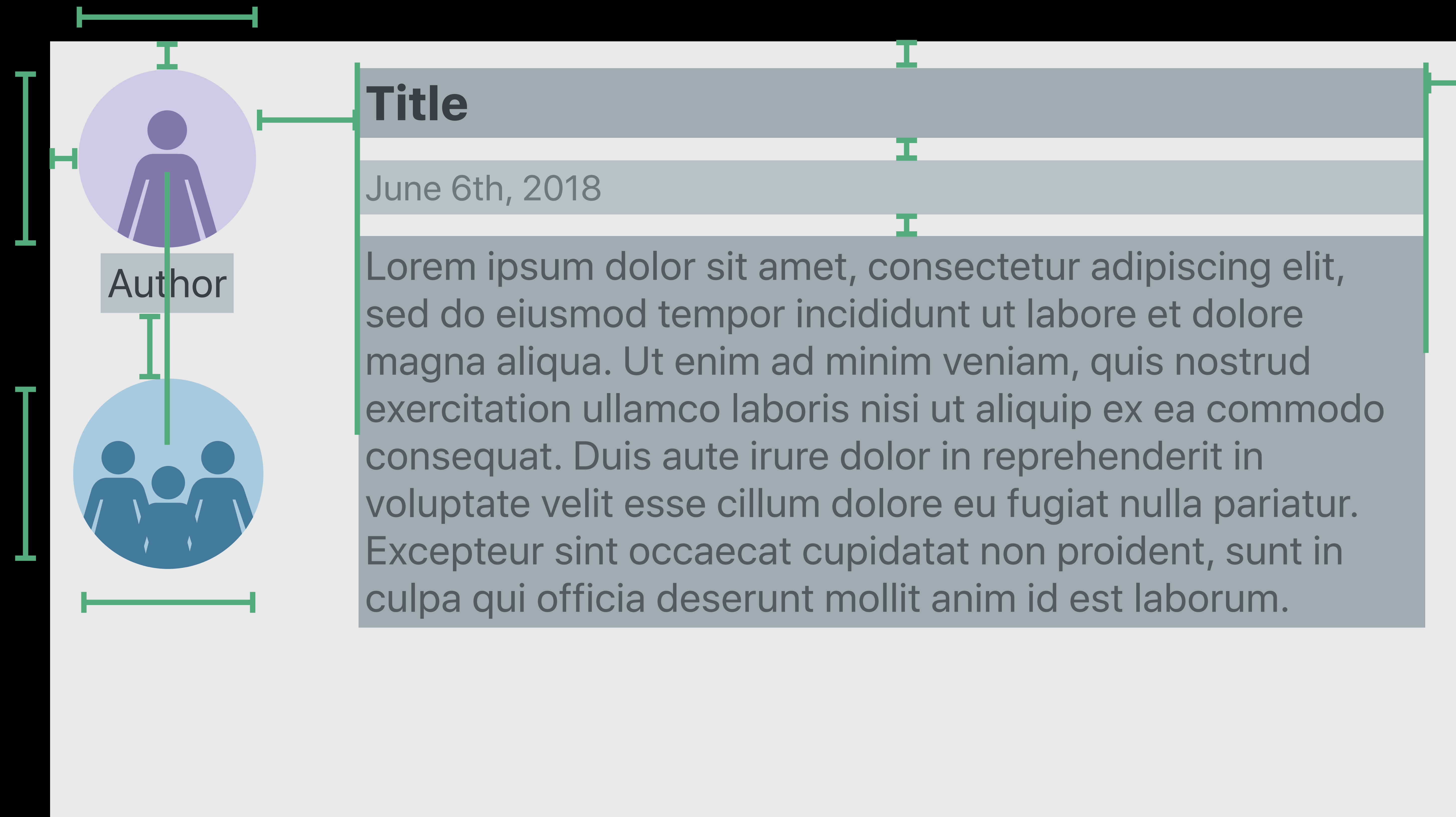
```
imageConstraints =
```

```
[ H, I, H, I ]
```

```
noImageConstraints =
```

```
[ I ]
```

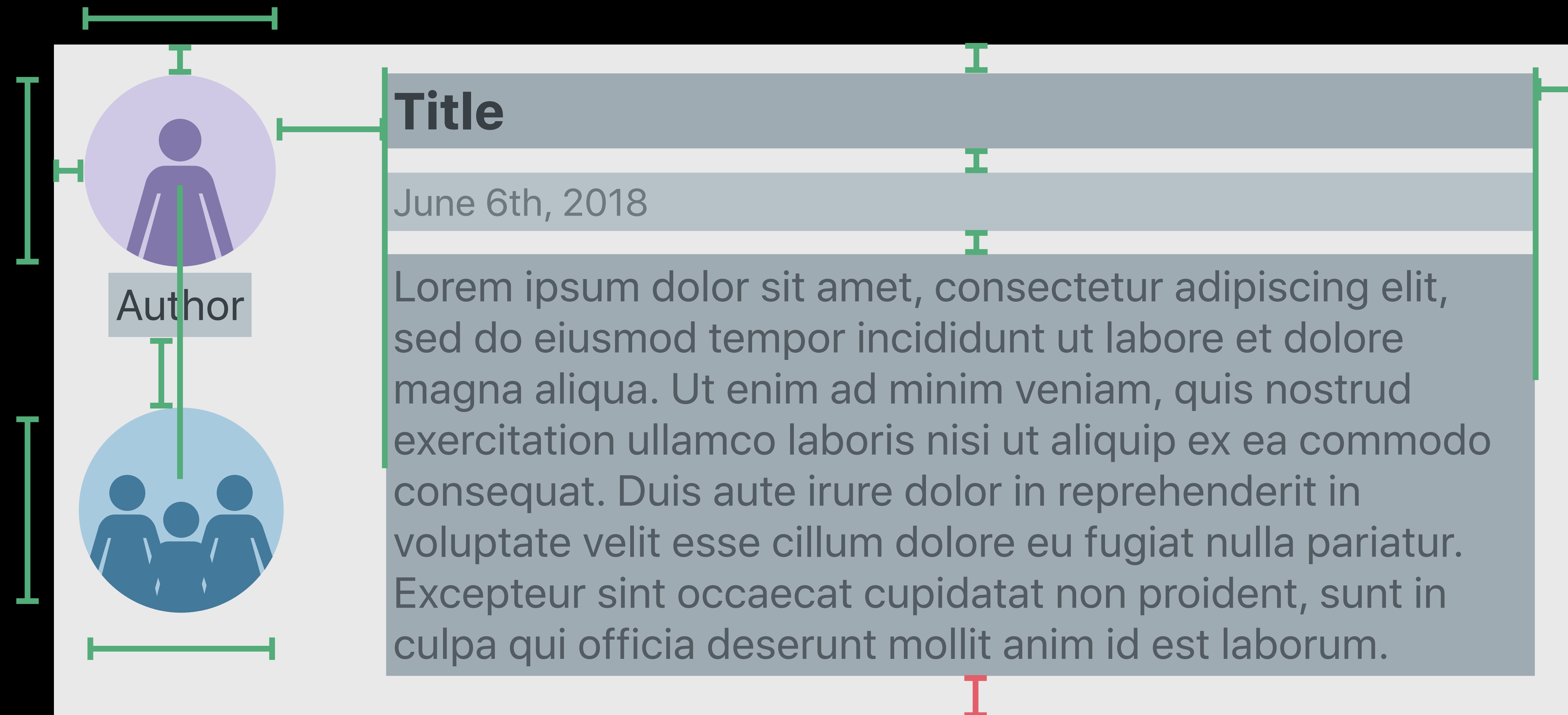
Building a More Performant Layout



```
imageConstraints =  
[ H, I, H, I ]
```

```
noImageConstraints =  
[ I ]
```

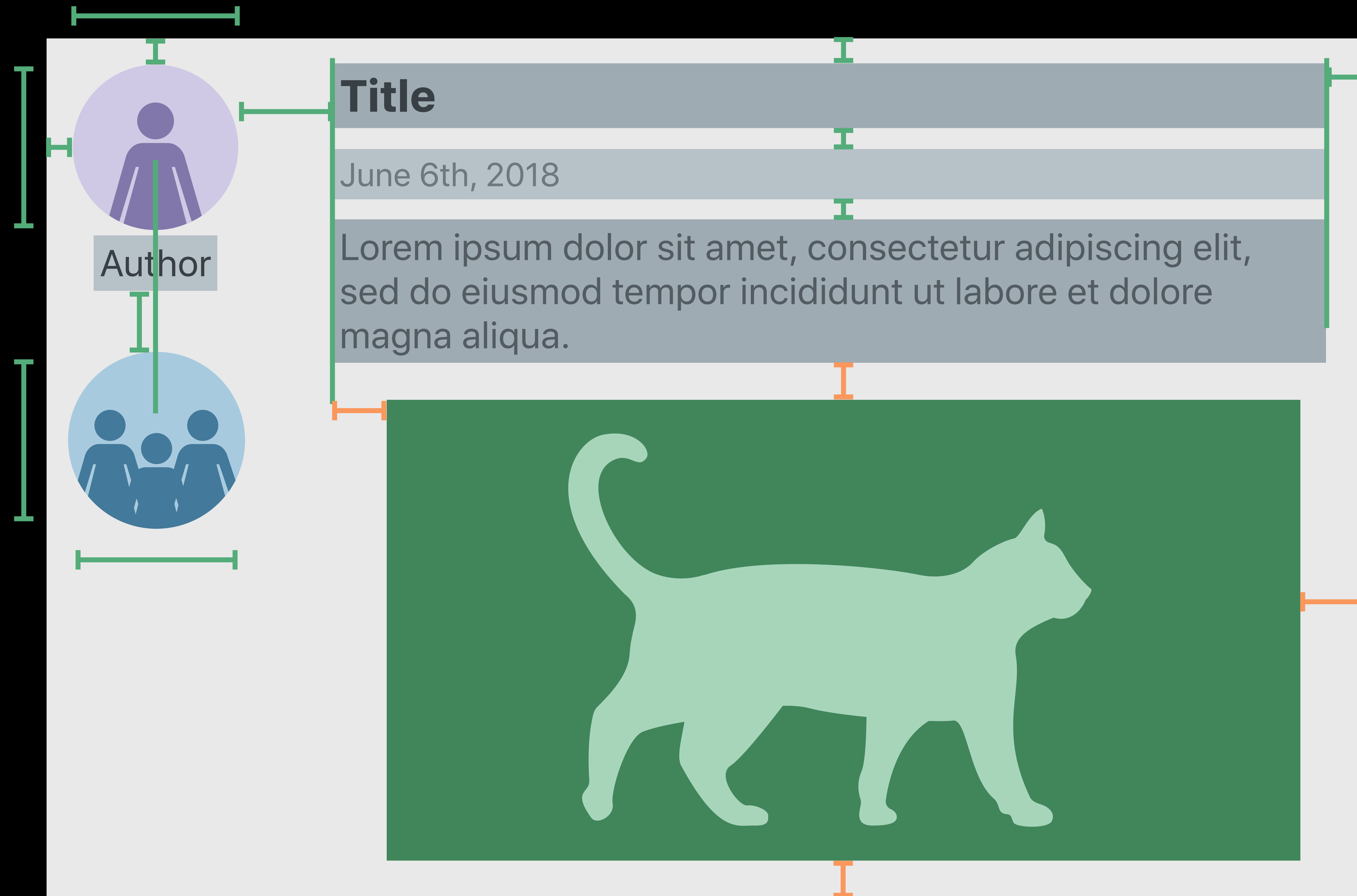

Building a More Performant Layout



```
imageConstraints =  
[ H, I, H, I ]
```

```
noImageConstraints =  
[ I ]
```

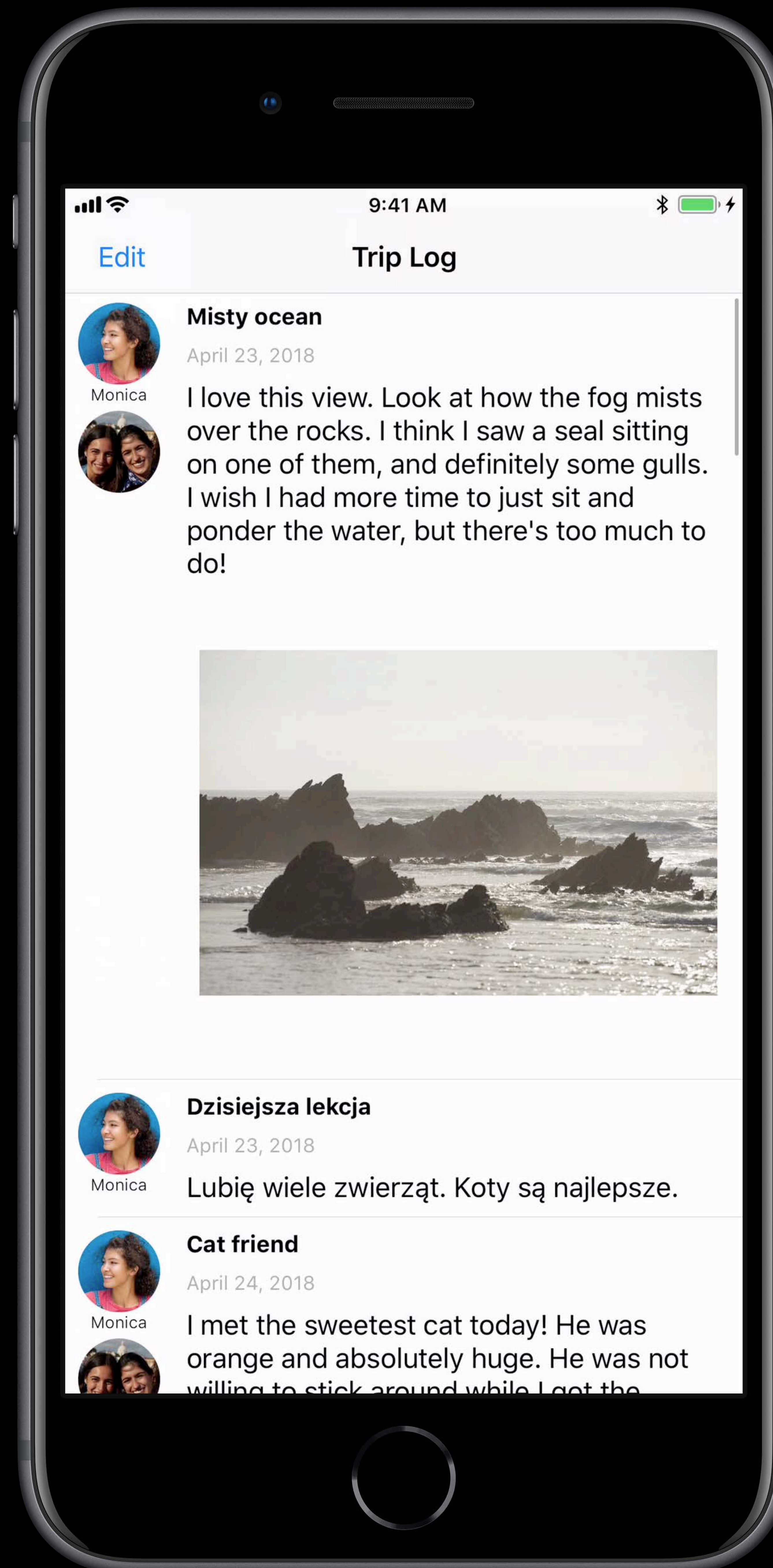
Building a More Performant Layout



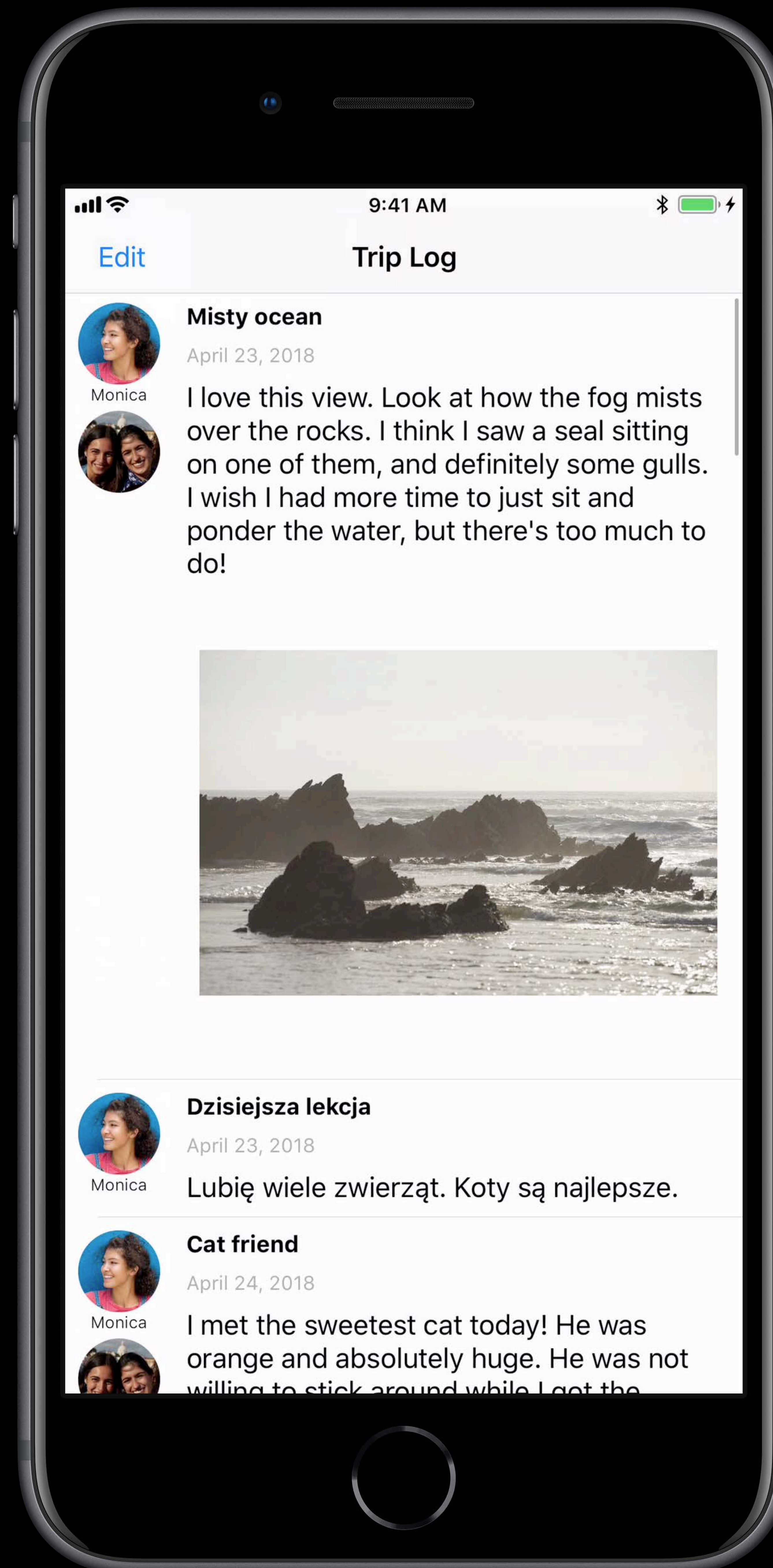
```
imageConstraints =  
[ H, I, H, I ]
```

```
noImageConstraints =  
[ I ]
```

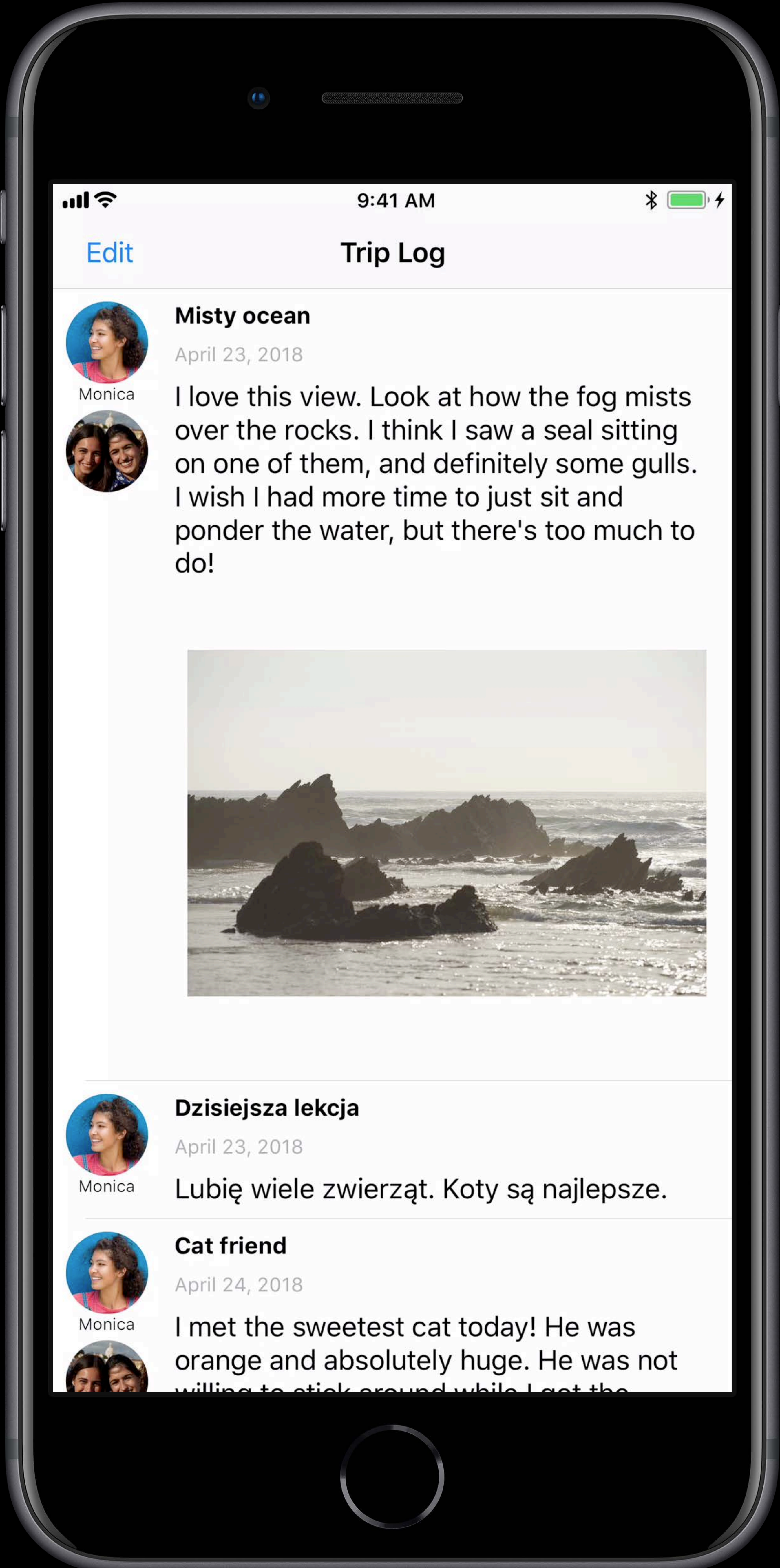

Before



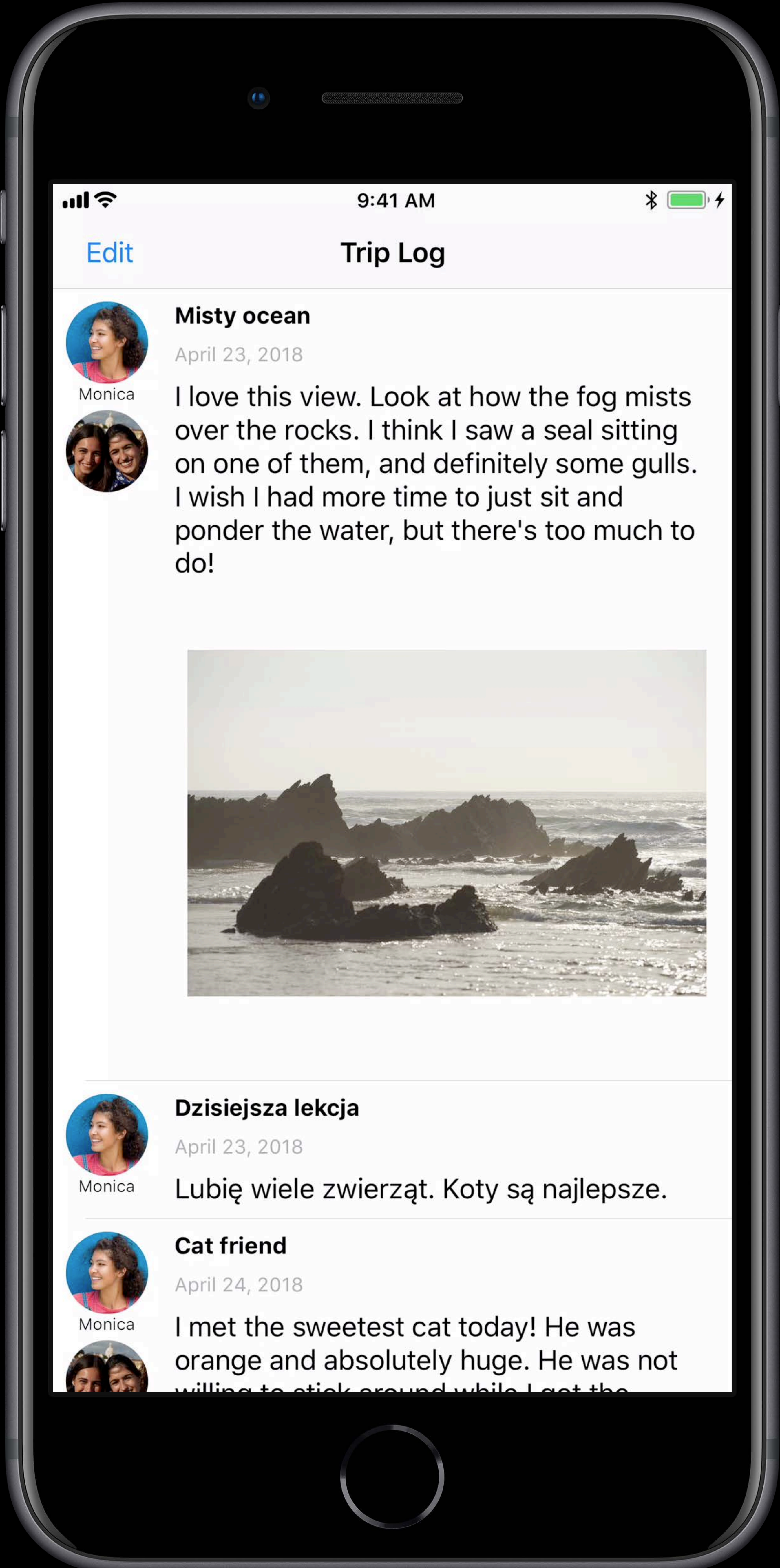
Before



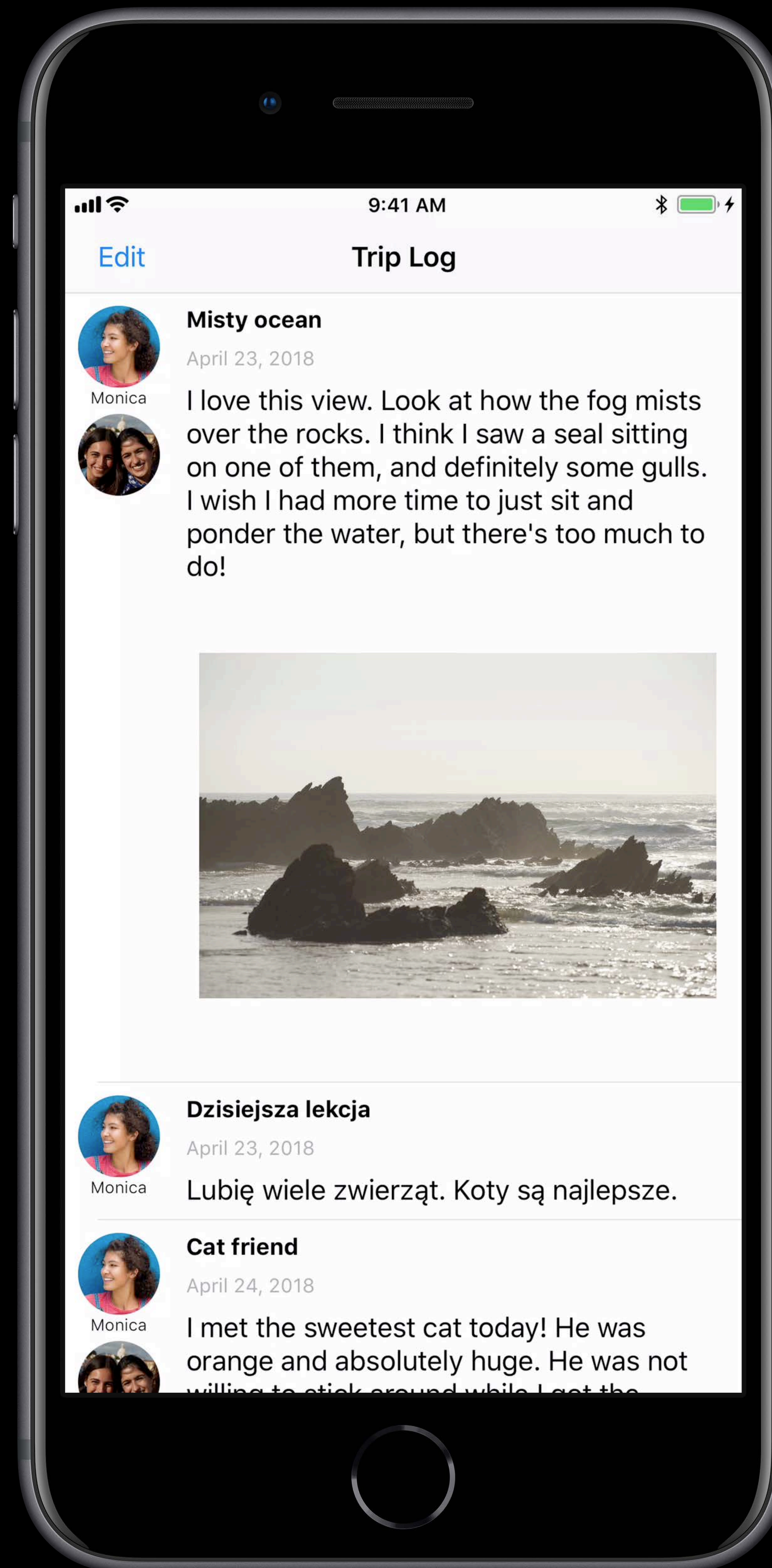
After



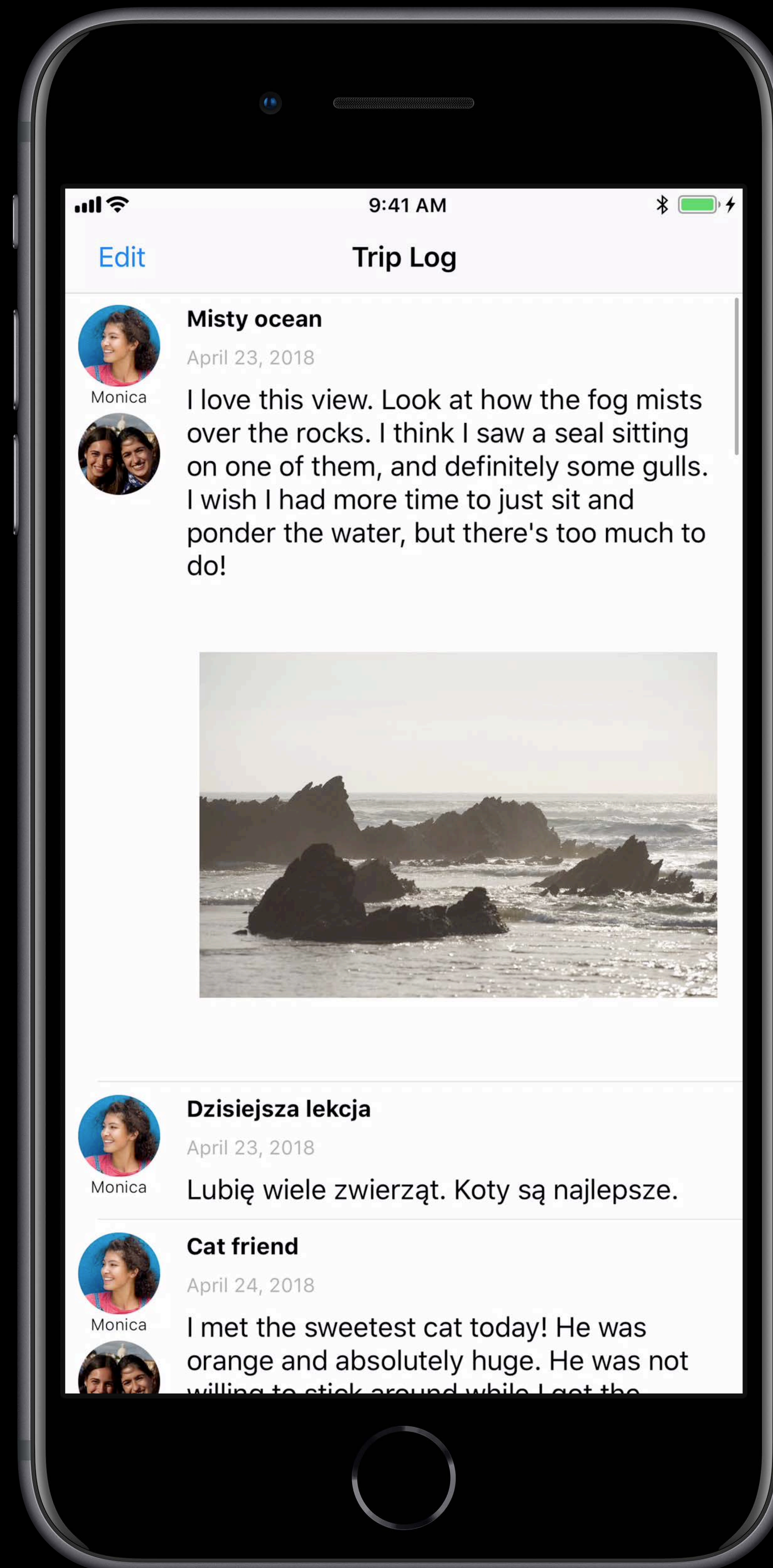
After



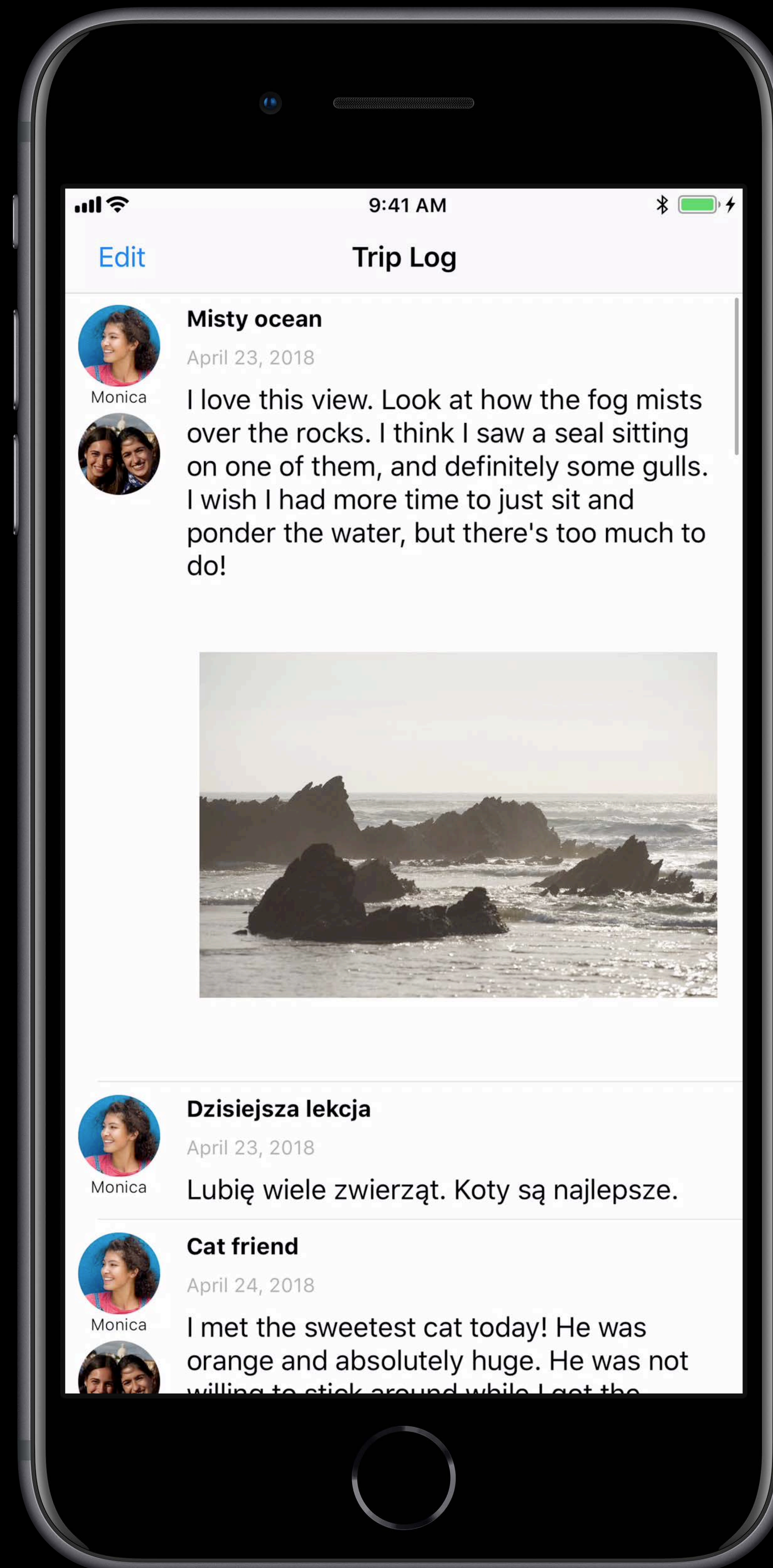
After (iOS 11)



After (iOS 12)



After (iOS 12)



Constraint Churn

Avoid removing all constraints

Add static constraints once

Only change the constraints that need changing

Hide views instead of removing them

Intrinsic Content Size

Not all views need it

Views with non view content:

- UIImageView returns its image size
- UILabel returns its text size

UIView uses it to make constraints

Overriding

Optionally override for text measurement

- Return size if known without text measurement
- Use `UIView.noIntrinsicMetric` and constraints

Overriding

Optionally override for text measurement

- Return size if known without text measurement
- Use `UIView.noIntrinsicMetric` and constraints

```
override var intrinsicContentSize: CGSize {  
    return CGSize(width: UIView.noIntrinsicMetric, height: UIView.noIntrinsicMetric)  
}
```

System Layout Size Fitting Size

```
systemLayoutSizeFitting(_ targetSize:)
```

Used when combining with frames

Getting Size from the Engine

Engine is created

Constraints are added

Layout is solved

Size of the top view is returned

Engine is discarded



Getting Size from the Engine

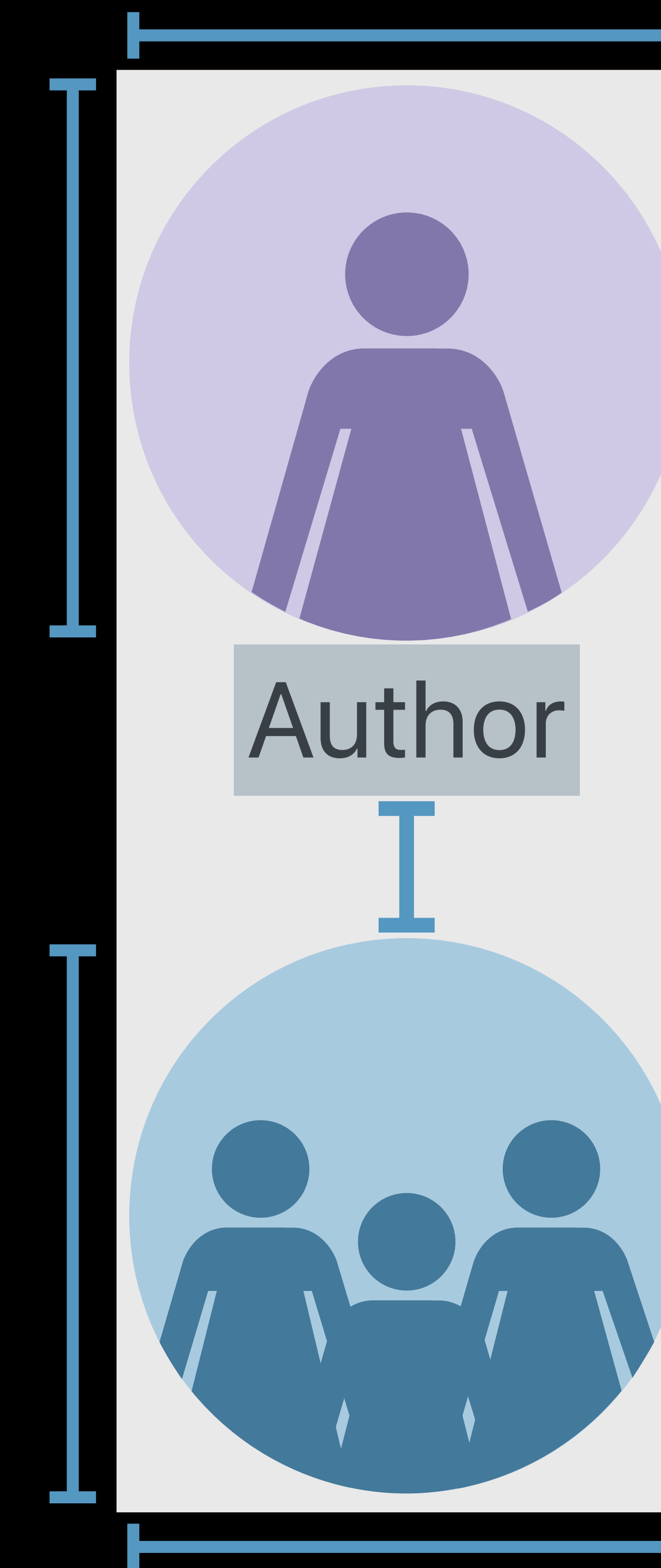
Engine is created

Constraints are added

Layout is solved

Size of the top view is returned

Engine is discarded



Getting Size from the Engine

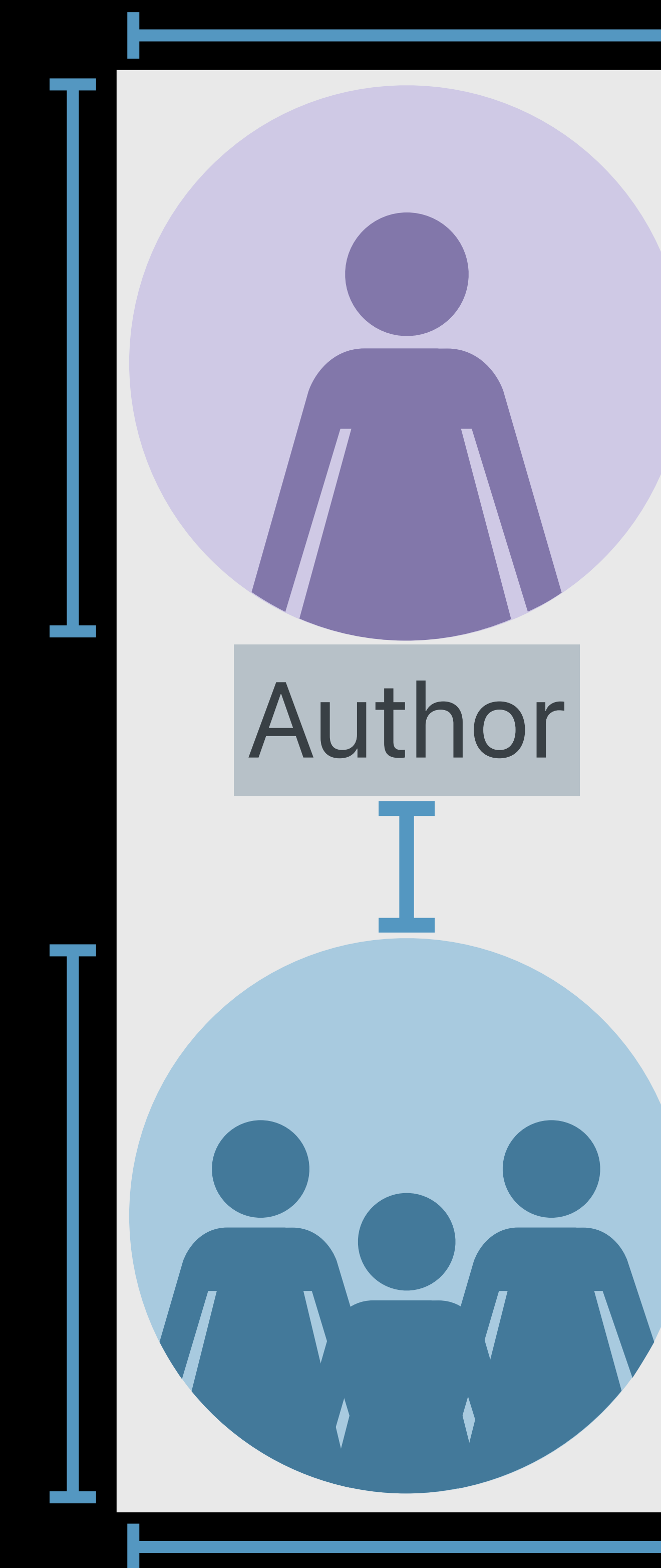
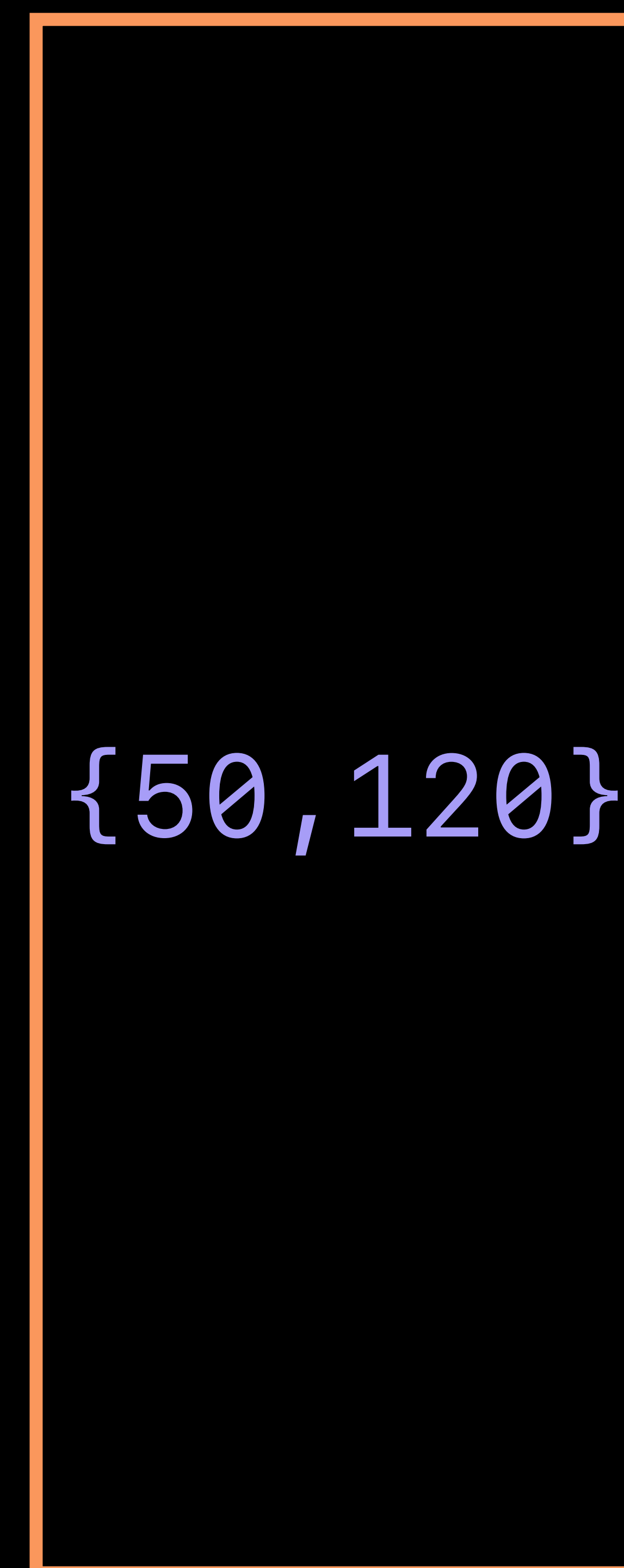
Engine is created

Constraints are added

Layout is solved

Size of the top view is returned

Engine is discarded



Getting Size from the Engine

Engine is created

Constraints are added

Layout is solved

`{50, 120}`

Size of the top view is returned

Engine is discarded

Unsatisfiable Constraints

There is no solution for the specified constraints

Logging will help you to debug

Can mask other issues

See related session for debugging tips

Ready, Set, Go!

Think before you update constraints

Tune as needed

Enjoy your faster layouts

More Information

<https://developer.apple.com/wwdc18/220>

 **WWDC18**