# Making Apps Adaptive

## Part 1

Session 222

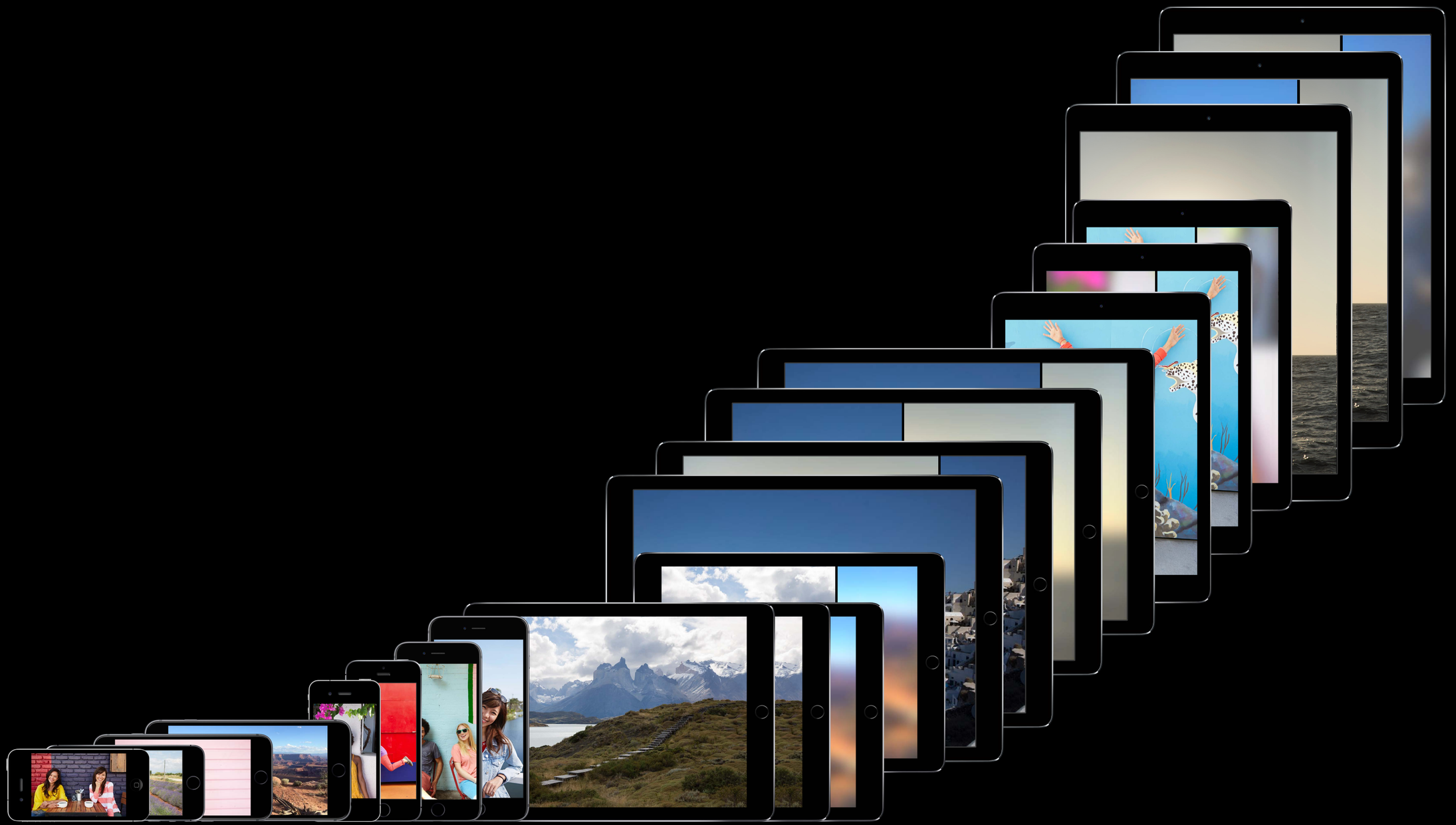
Kevin Cathey Interface Builder Engineer
Brent Shank Interface Builder Engineer

© 2016 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.

Takeaway

# The system is going to do most of the work so you don’t have to.

1,000,000,000



# Dynamic Type

Dynamic Type

Layout Direction

Dynamic Type

Layout Direction

Display Gamut

Dynamic Type

Layout Direction

Display Gamut

Interface Style

# 300+

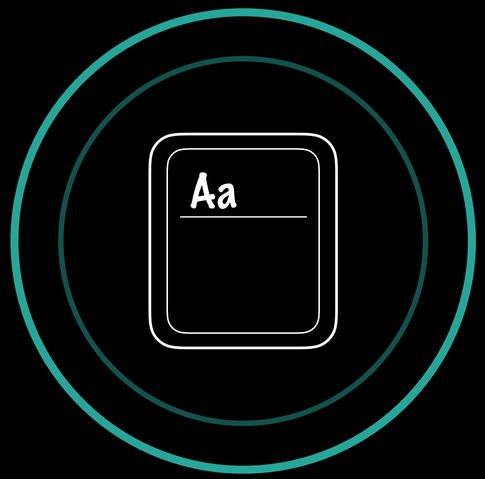
Combinations

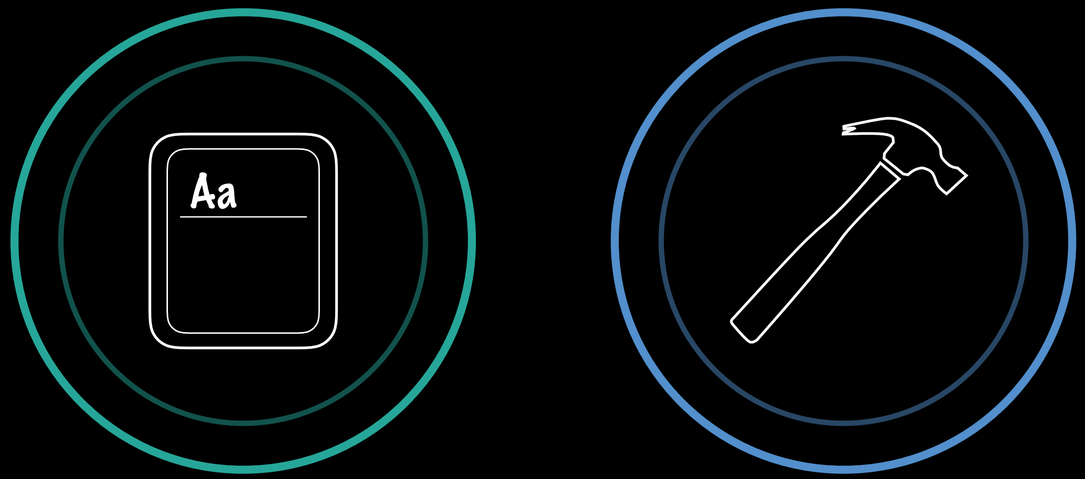
Takeaway

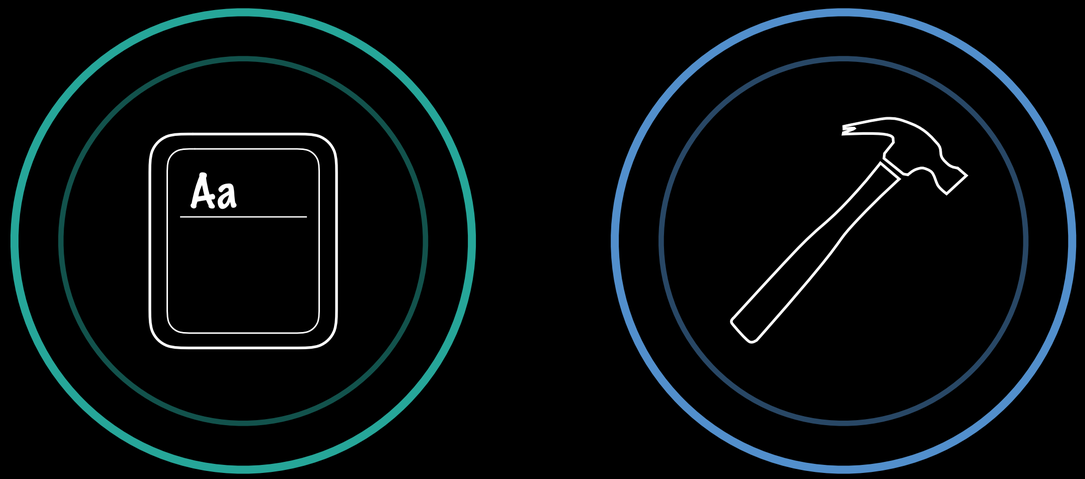
# The system is going to do most of the work so you don’t have to.

Takeaway

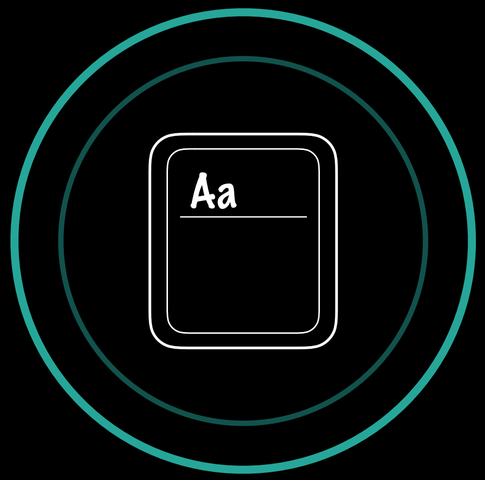
# The system is going to do most of the work so you don’t have to.


Aa


Aa

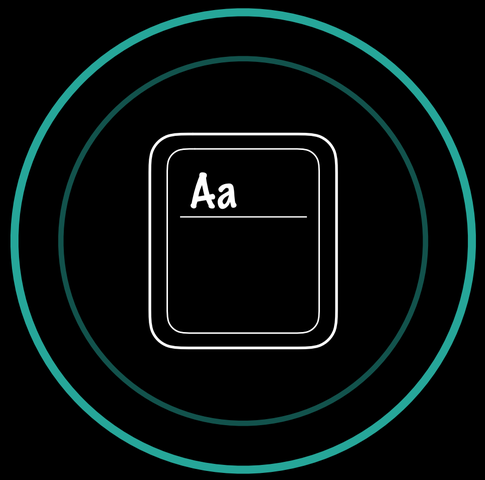



PART 1



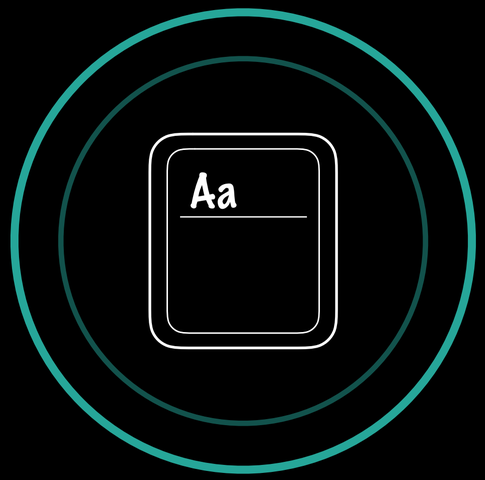

PART 1

PART 2



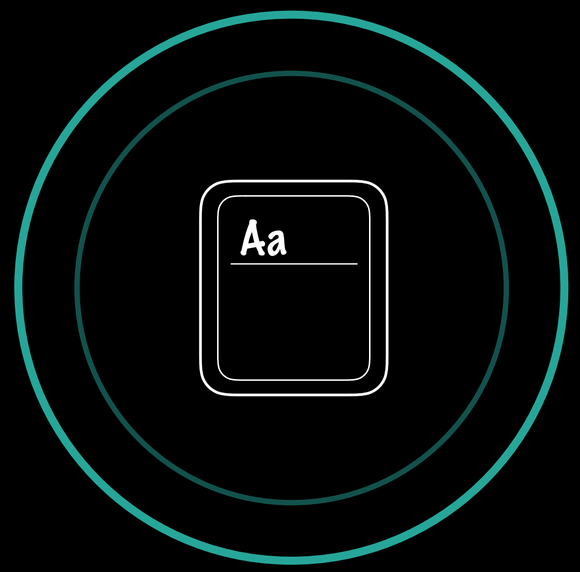

PART 1

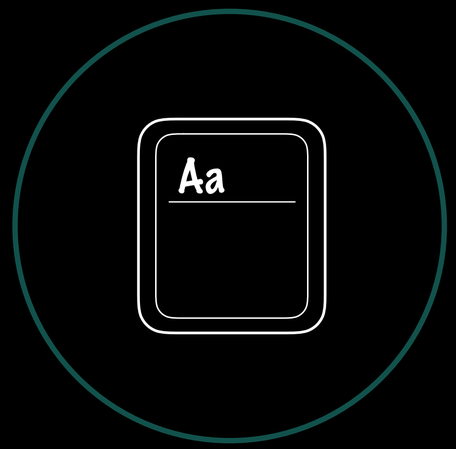
PART 2

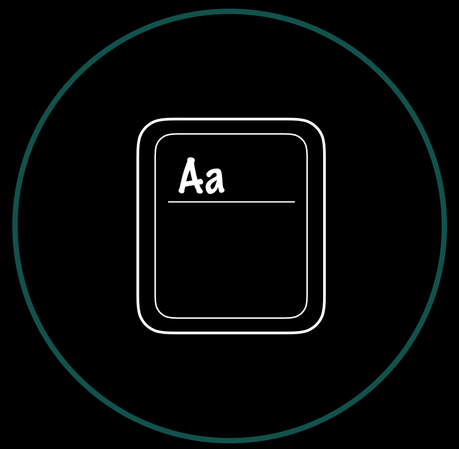



PART 1

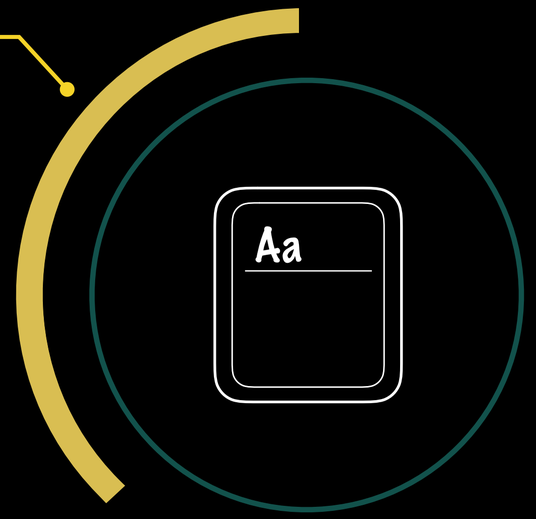
PART 2


Aa

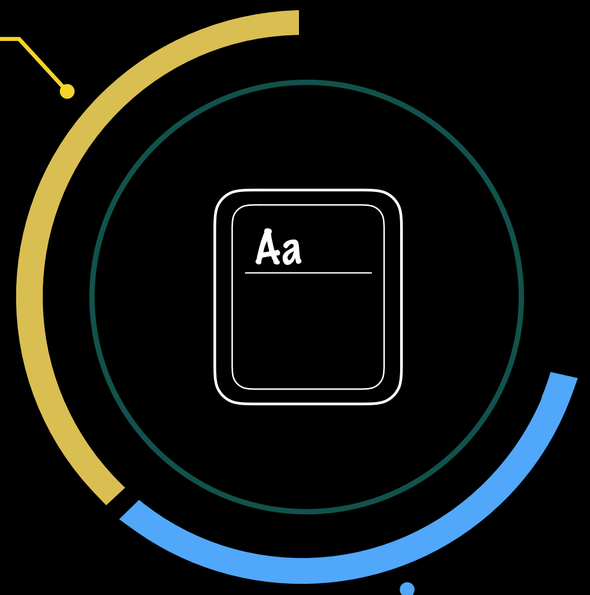

Aa


Aa

What traits are



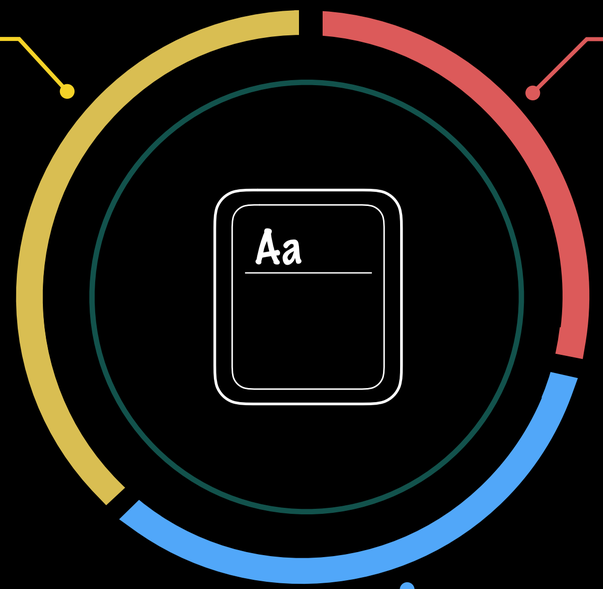
What traits are




Size classes

What traits are

How traits work




Size classes

# Traits

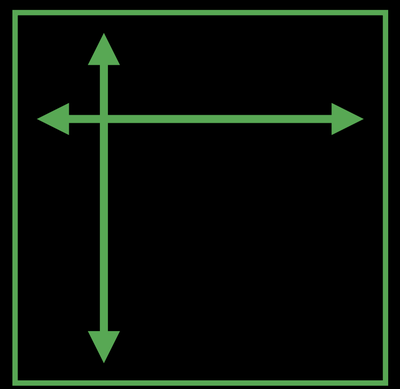
horizontalSizeClass

[horizontalSizeClass = Compact]

```
[horizontalSizeClass = Compact,
verticalSizeClass = Regular,
displayGamut = P3]
```

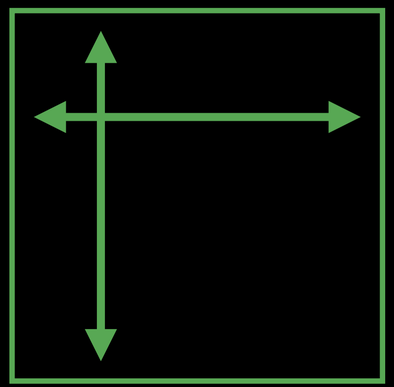
# Trait Examples

# Trait Examples

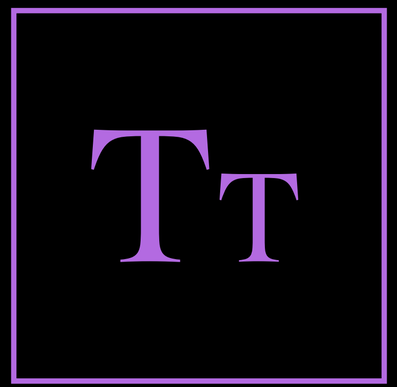


Size Classes

# Trait Examples

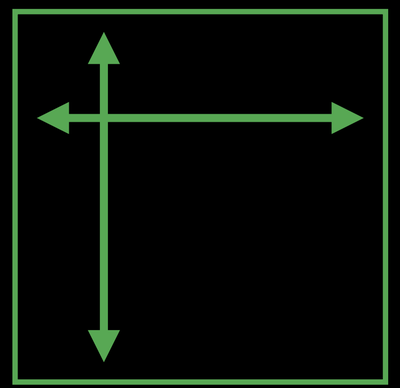


Size Classes

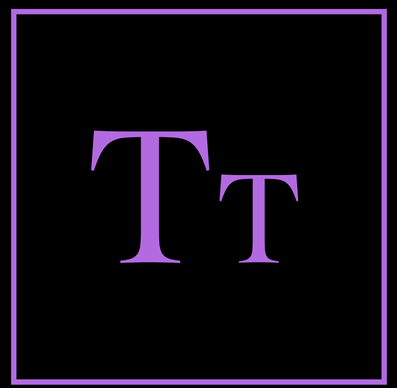



Dynamic Type

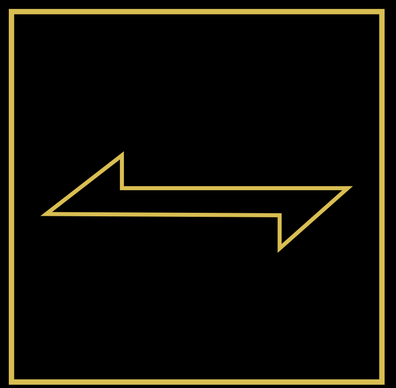
# Trait Examples



Size Classes

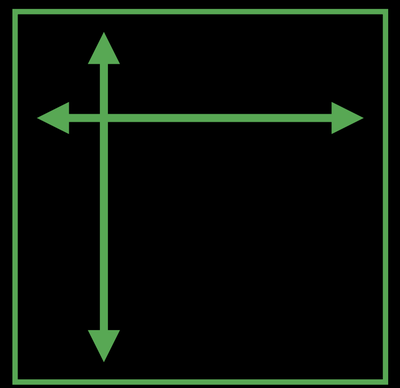


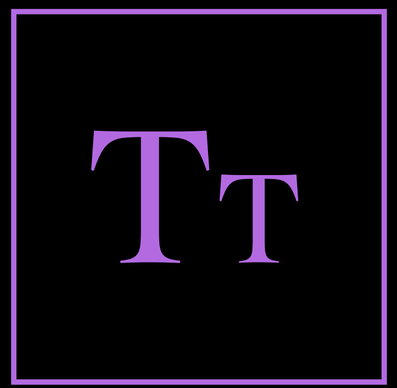

Dynamic Type

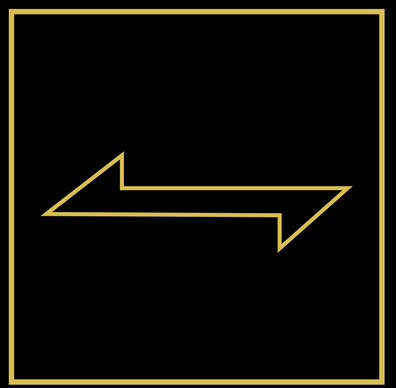


Layout Direction

# Trait Examples





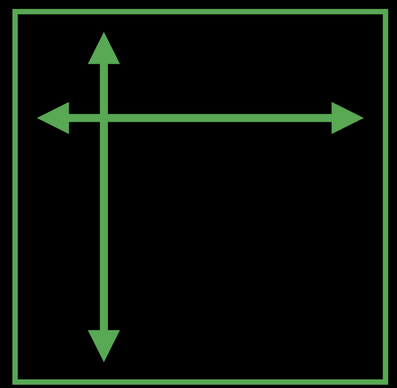
LAYOUT

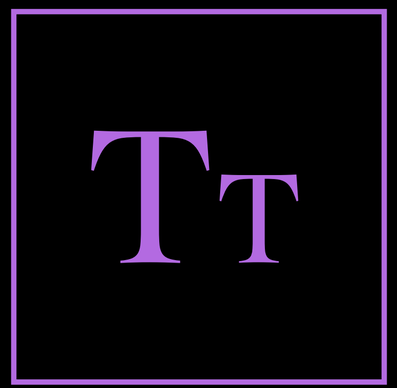
Size Classes

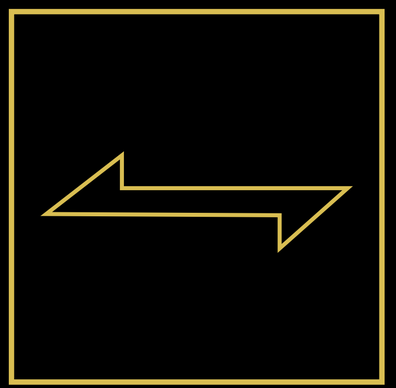
Dynamic Type

Layout Direction

# Trait Examples





LAYOUT

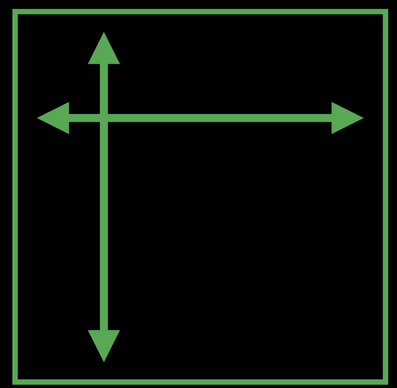
Size Classes

Dynamic Type

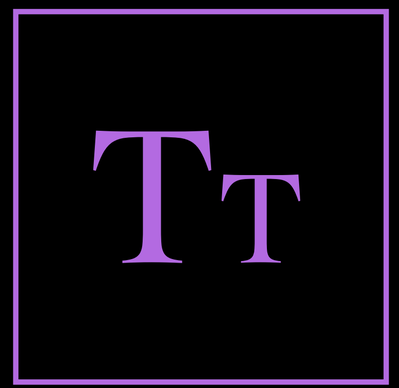
Layout Direction

APPEARANCE

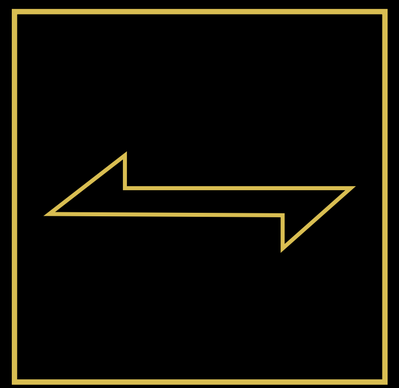
# Trait Examples



Size Classes



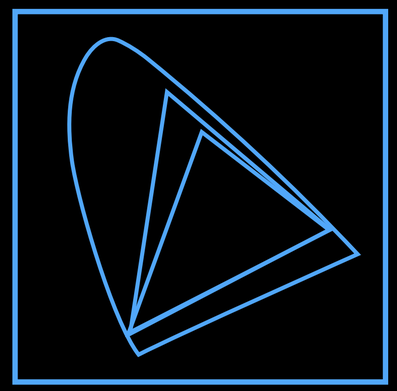

Dynamic Type



Layout Direction

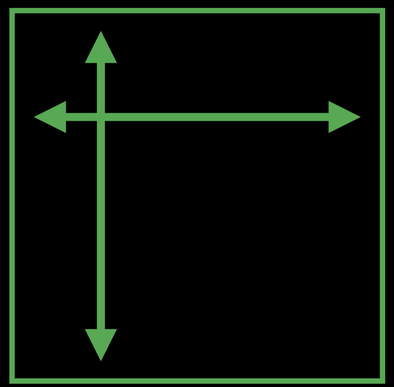
LAYOUT

APPEARANCE

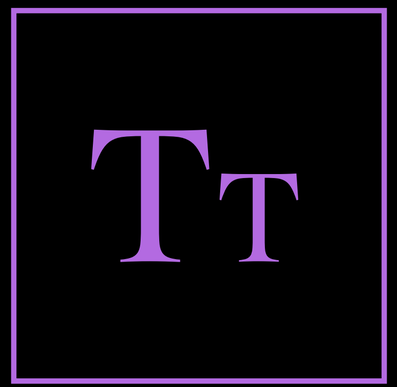


Display Gamut

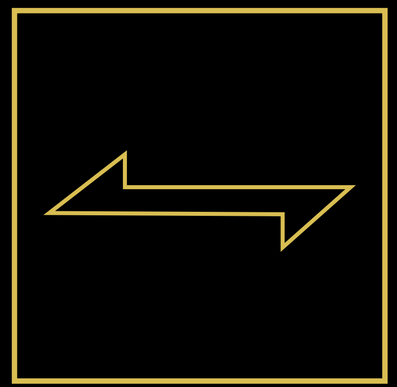
# Trait Examples



Size Classes



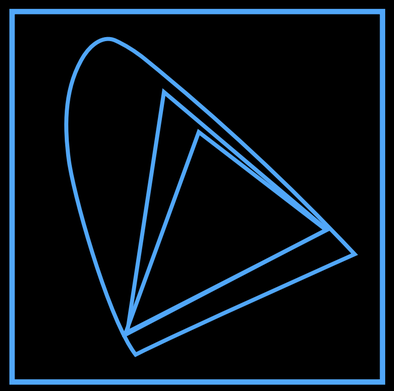

Dynamic Type



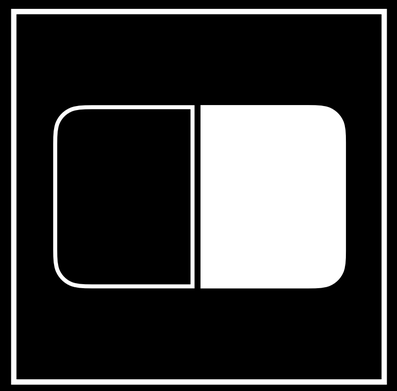
Layout Direction

LAYOUT

APPEARANCE

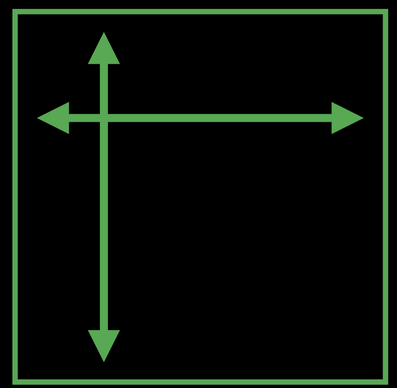


Display Gamut

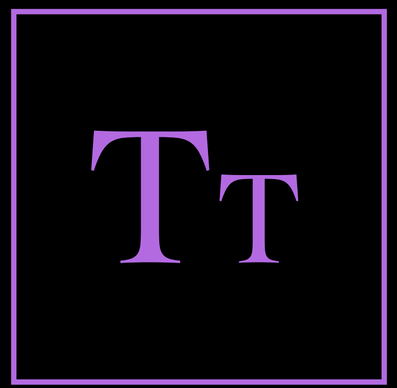


Interface Style

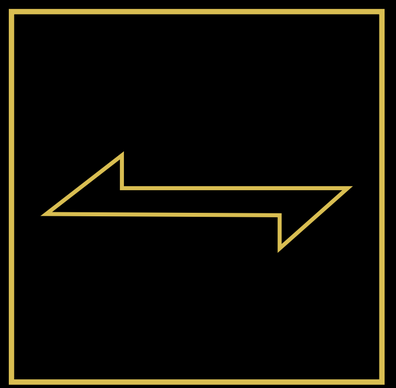
# Trait Examples



Size Classes




Dynamic Type

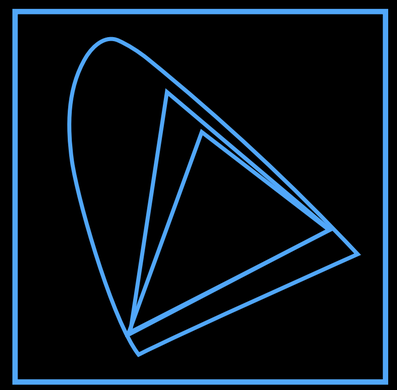


Layout Direction

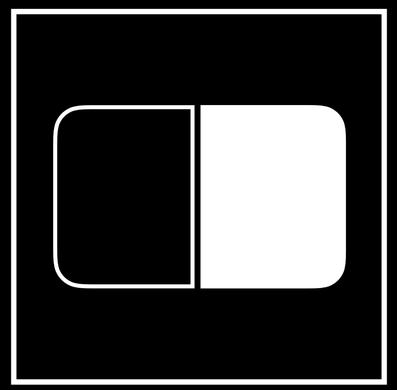
LAYOUT

APPEARANCE

CAPABILITIES

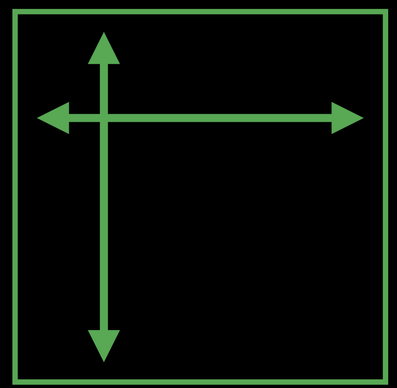


Display Gamut



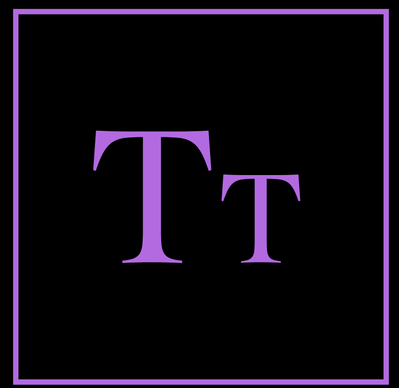
Interface Style

# Trait Examples

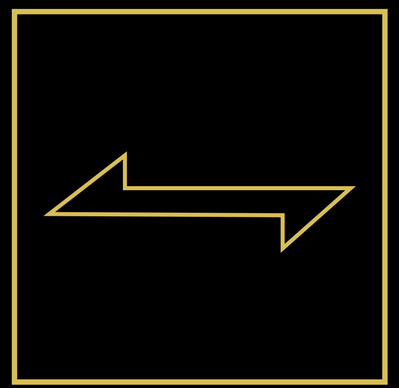


LAYOUT

Size Classes

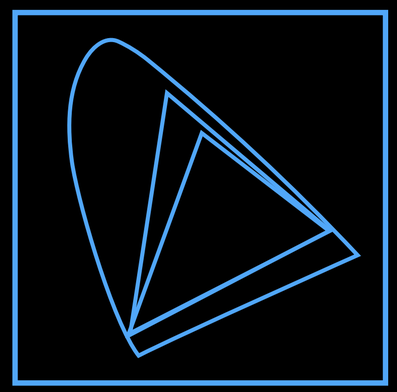



Dynamic Type

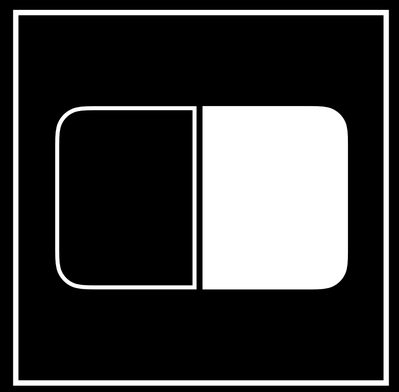


Layout Direction

APPEARANCE

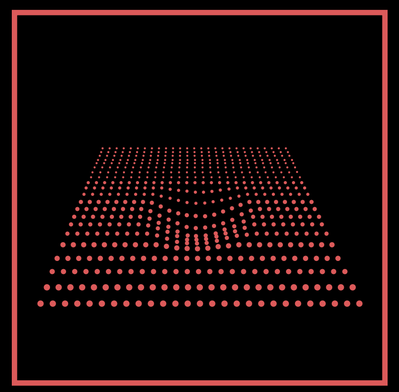


Display Gamut



Interface Style

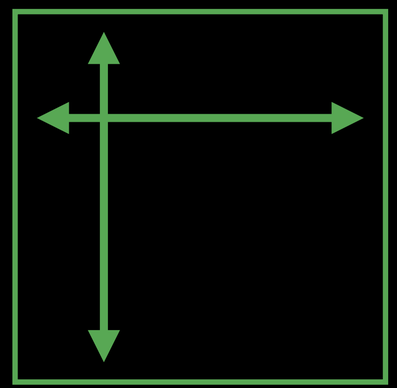
CAPABILITIES



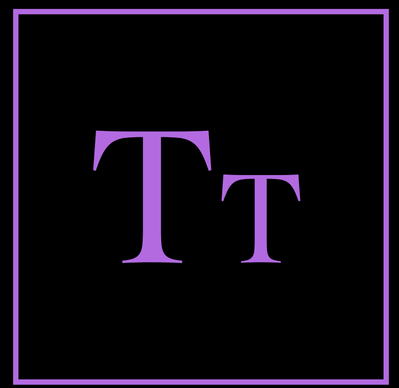
3D Touch

# Trait Examples

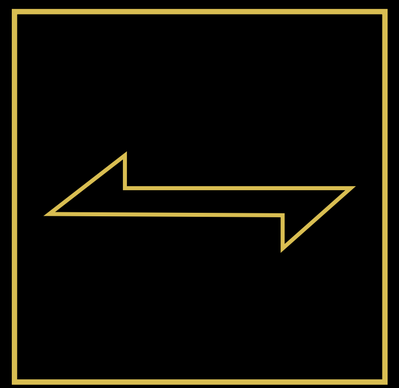
`UITraitCollection`



Size Classes



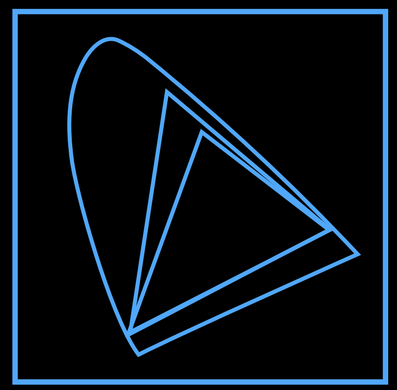
Dynamic Type



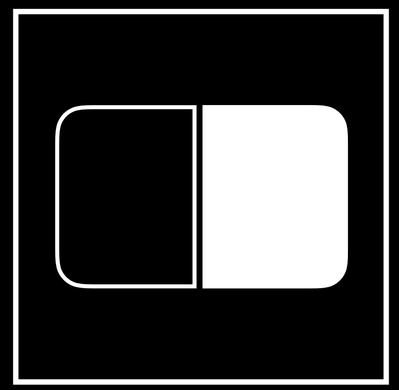
Layout Direction

**LAYOUT**

**APPEARANCE**

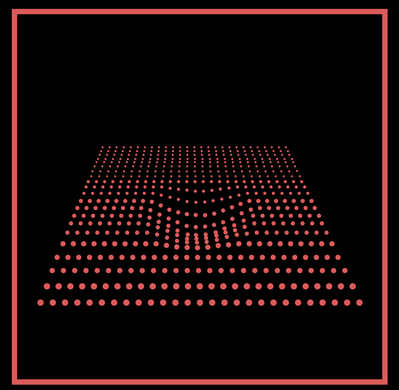


Display Gamut



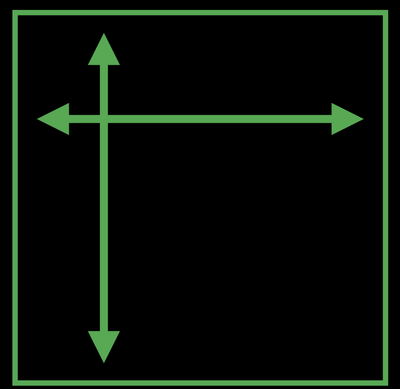
Interface Style

**CAPABILITIES**

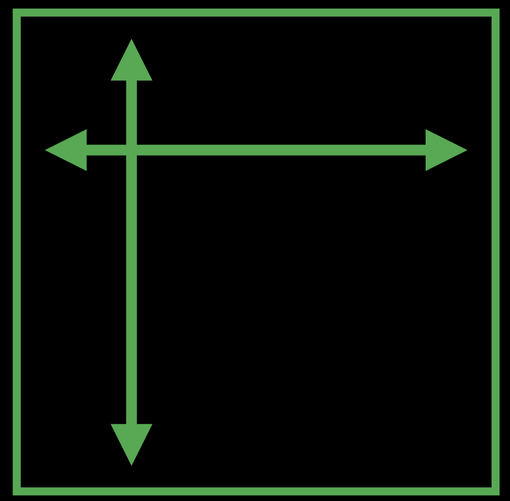


3D Touch

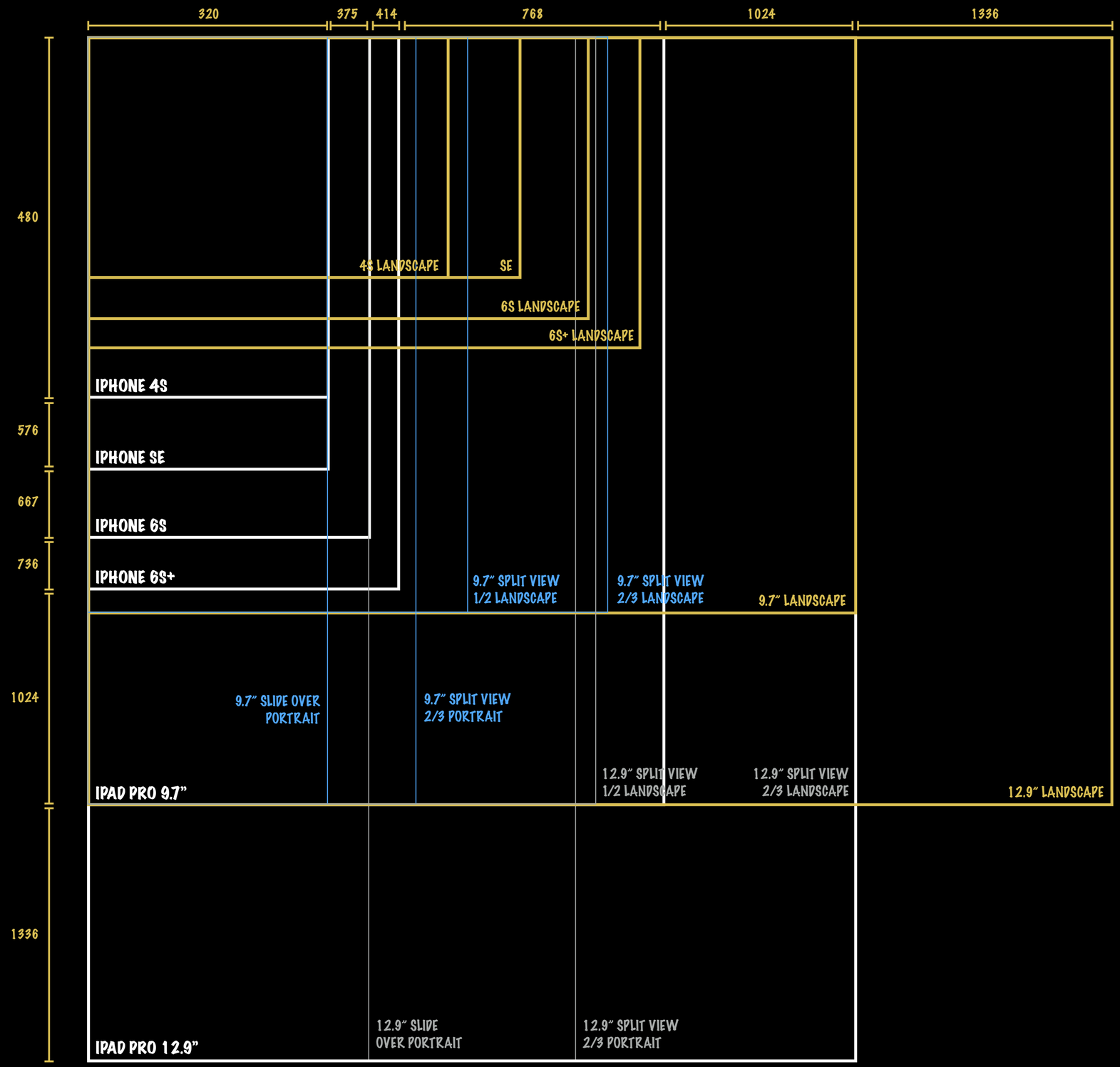
# Trait Examples



Size Classes



Size Classes


320
375
414
768
1024
1336
480
576
667
736
1024
1336
4S LANDSCAPE
SE
6S LANDSCAPE
6S+ LANDSCAPE
IPHONE 4S
IPHONE SE
IPHONE 6S
IPHONE 6S+
9.7" SPLIT VIEW
1/2 LANDSCAPE
9.7" SPLIT VIEW
2/3 LANDSCAPE
9.7" LANDSCAPE
9.7" SLIDE OVER
PORTRAIT
9.7" SPLIT VIEW
2/3 PORTRAIT
12.9" SPLIT VIEW
1/2 LANDSCAPE
12.9" SPLIT VIEW
2/3 LANDSCAPE
12.9" LANDSCAPE
IPAD PRO 9.7"
12.9" SLIDE
OVER PORTRAIT
12.9" SPLIT VIEW
2/3 PORTRAIT
IPAD PRO 12.9"

# Screen Size

320
375
414
768
1024
1336
480
4S LANDSCAPE
SE
6S LANDSCAPE
6S+ LANDSCAPE
9.7" SPLIT VIEW 1/2 LANDSCAPE
9.7" SPLIT VIEW 2/3 LANDSCAPE
9.7" LANDSCAPE
9.7" SLIDE OVER PORTRAIT
9.7" SPLIT VIEW 2/3 PORTRAIT
12.9" SPLIT VIEW 1/2 LANDSCAPE
12.9" SPLIT VIEW 2/3 LANDSCAPE
12.9" LANDSCAPE
1336
IPAD PRO 12.9"
12.9" SLIDE OVER PORTRAIT
12.9" SPLIT VIEW 2/3 PORTRAIT

Screen Size

Orientation

Screen Size

Orientation

Adaptation

Takeaway

# Base layout on available space, not device, orientation, or adaptation

# Reacting to Available Space

# Reacting to Available Space

Coarse Changes

# Reacting to Available Space

Coarse Changes

Fine Changes

# Reacting to Available Space

Coarse Changes

Fine Changes

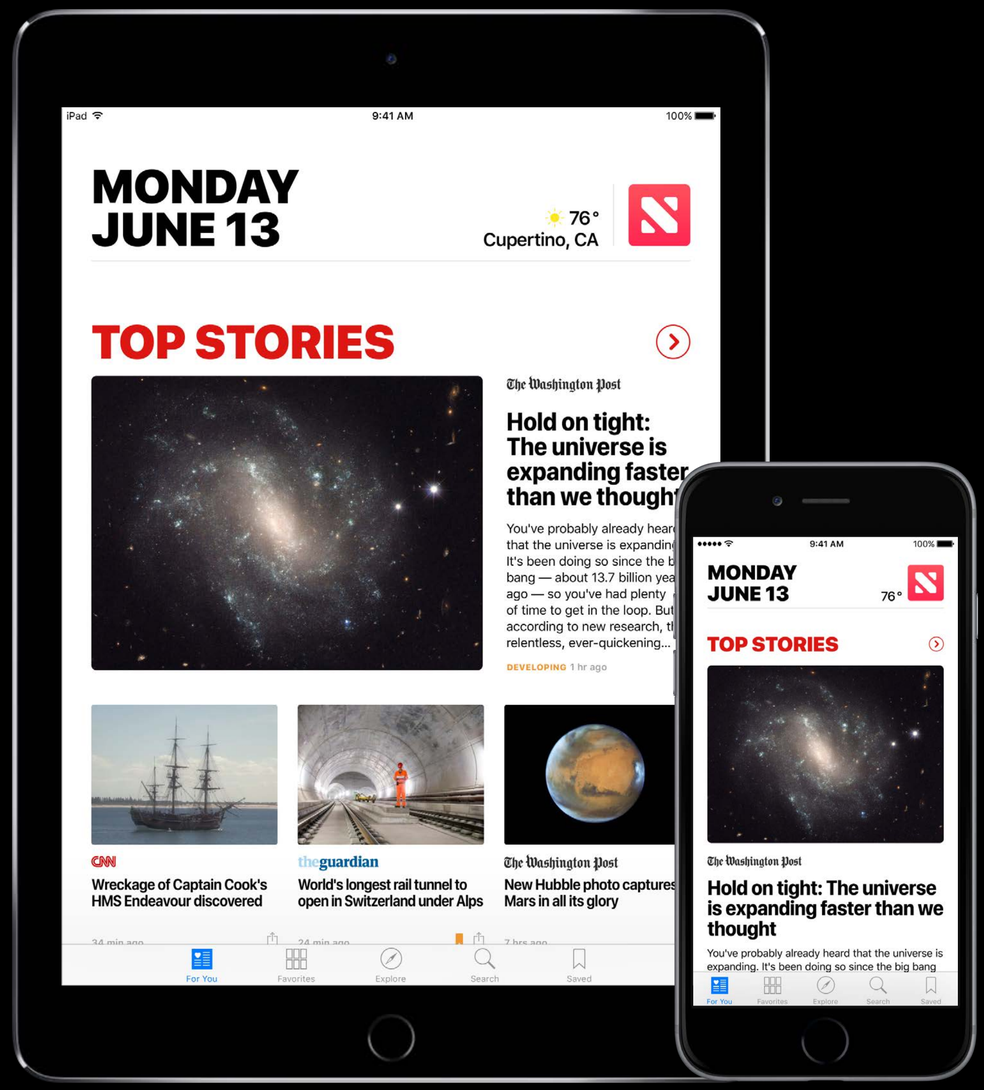
"All problems in computer science can be solved by another level of indirection"

David Wheeler

Takeaway

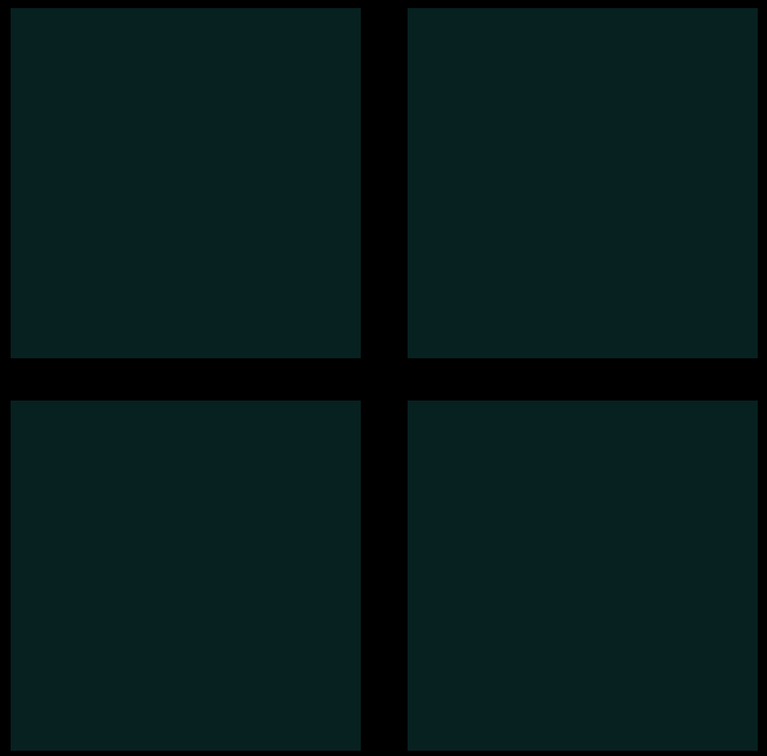
# Size classes express experience


9:41 AM
100%
Messages
Matt
Details
Here's a great photo Emily took on Friday.
It really captures my Frisbee prowess.
You know you dropped it, right?
That's what I meant.
If nothing else, the dinosaur socks look terrific. Haha.
I figured you'd like those.
We need to hang out again, soon. Don't be extinct, OK?
Text Message
Send
Search
nne Fox 9:36 AM
usly?
tt Reiff 9:28 AM
need to hang out again, soon.
't be extinct, OK?
ad Isanhart 9:21 AM
t did you guys think of the
ie? Hope I didn't oversell it.
am Seilaudom 8:58 AM
you get Dawn's email? I think
caps are permanently locked.
tchen Schwartz 1:37 AM
am I up this late?! I think I'm
oming a vampire. But I love...
nie Wong Yesterday
you going to spin class? My
n says yes. My thighs say no.
xander Hunter Yesterday
running 10 minutes late.
ch is early by my standards.
n Steed Yesterday
going to Charleston this
kend. Any restaurant recom...
e Elfving Yesterday
baby picture. She has your
Text Message
Today 9:28 AM


iPad
9:41 AM
100%
MONDAY
JUNE 13
76°
Cupertino, CA
TOP STORIES
The Washington Post
Hold on tight:
The universe is
expanding faster
than we thought
You've probably already heard
that the universe is expanding
It's been doing so since the b
bang — about 13.7 billion yea
ago — so you've had plenty
of time to get in the loop. But
according to new research, t
relentless, ever-quickening...
DEVELOPING 1 hr ago
CNN
Wreckage of Captain Cook's
HMS Endeavour discovered
theguardian
World's longest rail tunnel to
open in Switzerland under Alps
The Washington Post
New Hubble photo captures
Mars in all its glory
For You
Favorites
Explore
Search
Saved
9:41 AM
100%
MONDAY
JUNE 13
76°
TOP STORIES
The Washington Post
Hold on tight: The universe
is expanding faster than we
thought
You've probably already heard that the universe is
expanding. It's been doing so since the big bang
For You
Favorites
Explore
Search
Saved

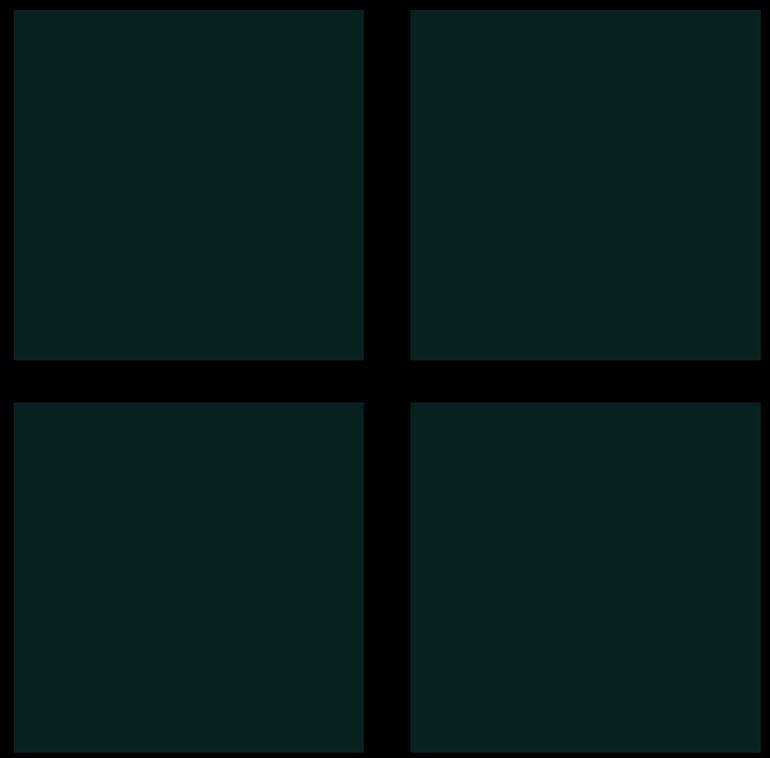
# The Specifics of Size Classes

# The Specifics of Size Classes



# The Specifics of Size Classes

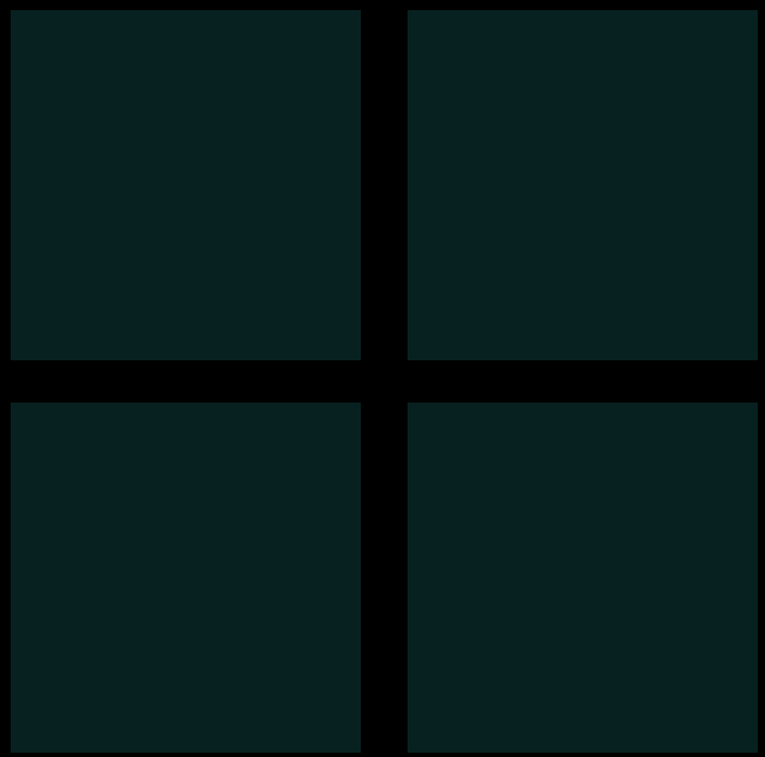
`horizontalSizeClass`



`verticalSizeClass`

# The Specifics of Size Classes

`horizontalSizeClass`

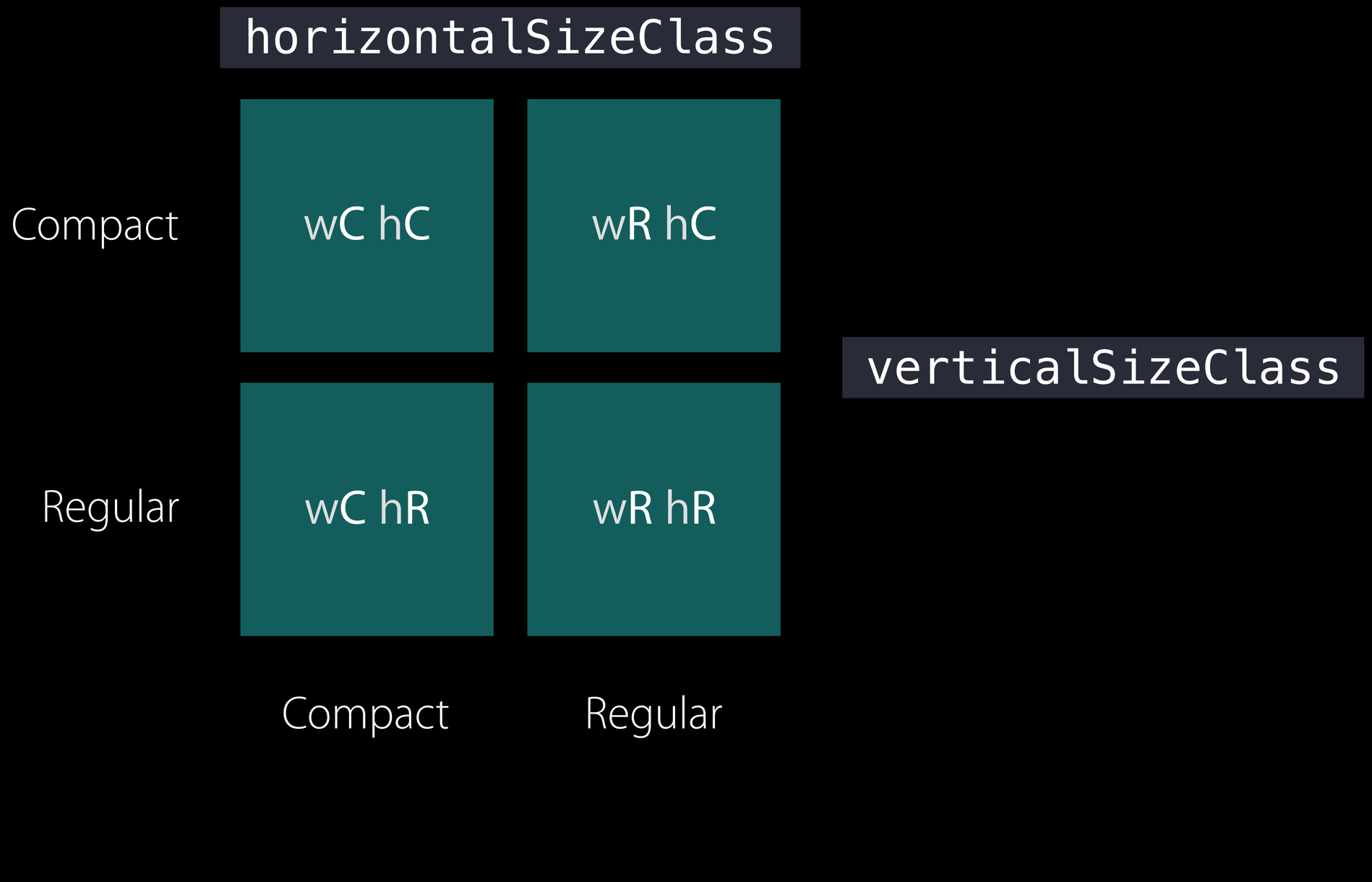


`verticalSizeClass`

Compact

Regular

# The Specifics of Size Classes



# The Specifics of Size Classes

# The Specifics of Size Classes

`horizontalSizeClass`

| `verticalSizeClass` | Compact | Regular |
| --- | --- | --- |
| Compact | wC hC | wR hC |
| Regular |  |  |

# The Specifics of Size Classes

`horizontalSizeClass`

`verticalSizeClass`

| | Compact | Regular |
|---|---|---|
| Regular | wR hC | |
| Compact | | |

# The Specifics of Size Classes

`horizontalSizeClass`

Compact

`verticalSizeClass`

Regular

Compact

Regular

# How Does This Help Me?

# How Does This Help Me?

Only think about four combinations

# How Does This Help Me?

Only think about four combinations

- But most commonly just two

# How Does This Help Me?

Only think about four combinations

- But most commonly just two
- Width is most common

# How Does This Help Me?

Only think about four combinations

- But most commonly just two
- Width is most common

System decides what combination applies

# How Does This Help Me?

Only think about four combinations

- But most commonly just two
- Width is most common

System decides what combination applies

- Size class can change dynamically

# How Does This Help Me?

Only think about four combinations

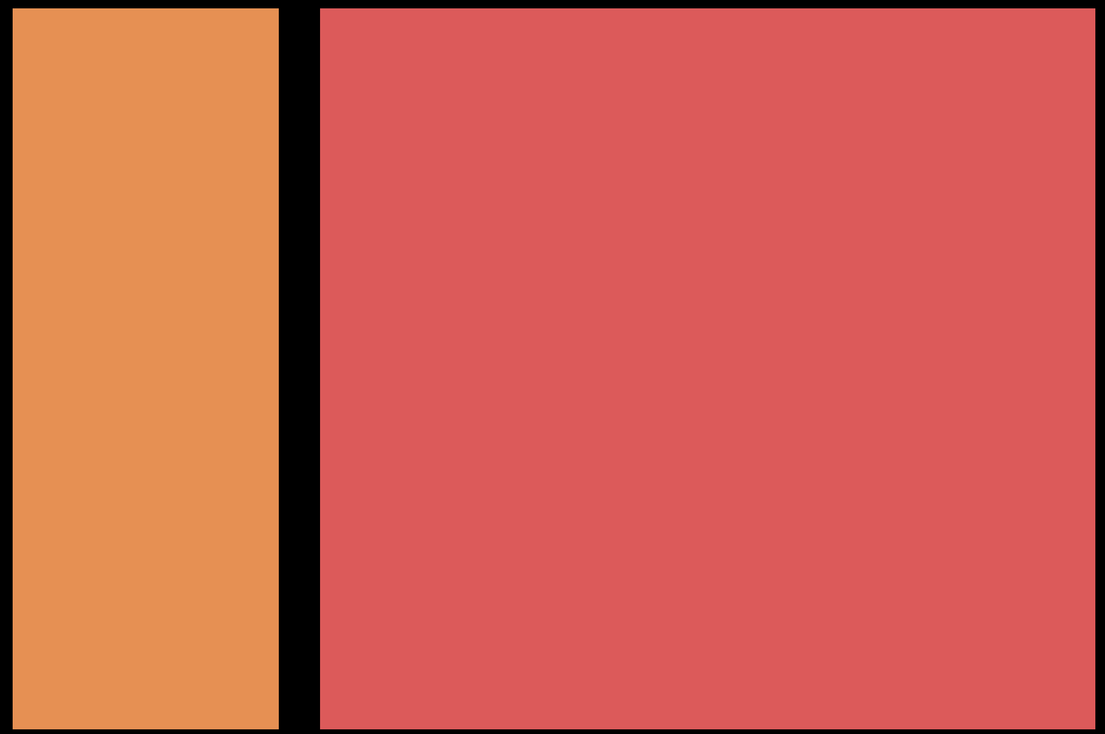
- But most commonly just two
- Width is most common

System decides what combination applies

- Size class can change dynamically
- If you use size classes, system can do the work for you

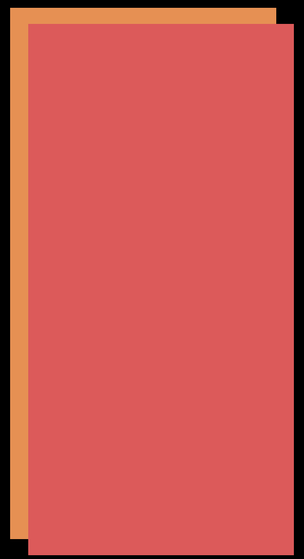
# Dynamic Size Class Changes

## View Controller structure



# Dynamic Size Class Changes

## View Controller structure



# Dynamic Size Class Changes

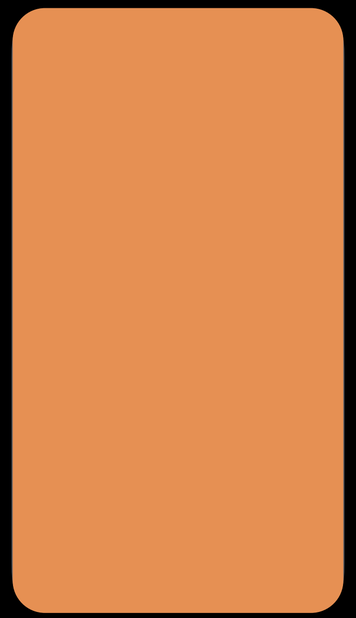
## Presentations

# Dynamic Size Class Changes

## Presentations

# Dynamic Size Class Changes

## Presentations



# Fine Grain Changes

# Fine Grain Changes

Use Auto Layout to specify changes
within a size class

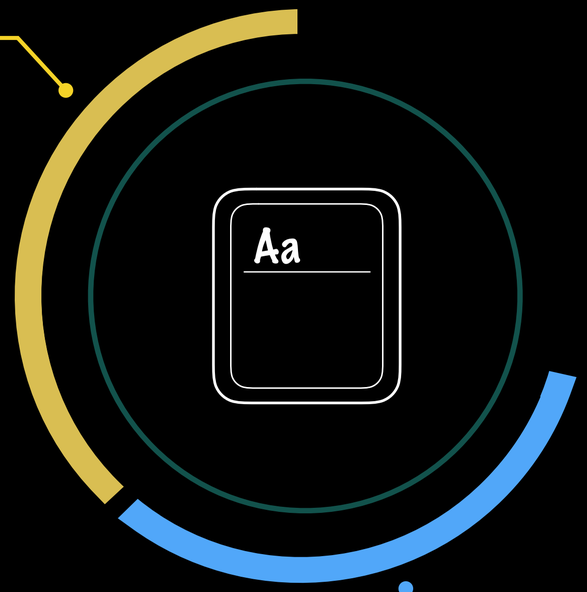
# Fine Grain Changes

Use Auto Layout to specify changes
within a size class

Medium grain changes?

| | | |
|---|---|---|
| Making Apps Adaptive, Part 2 | Presidio | Friday 9:00AM |

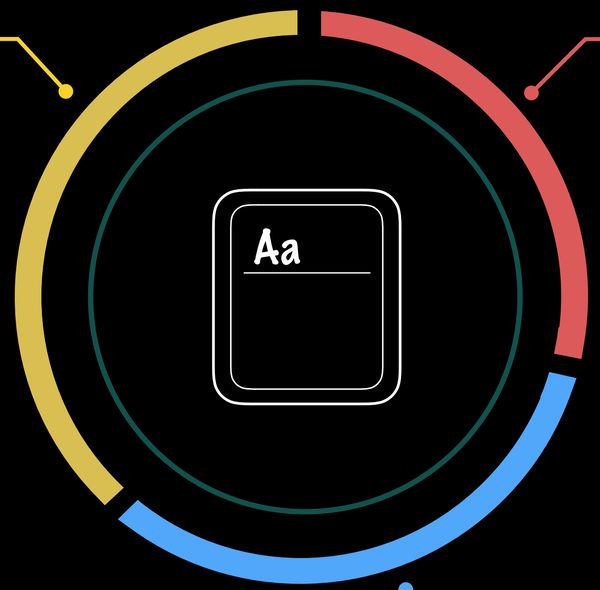
What traits are




Size classes

What traits are

How traits work



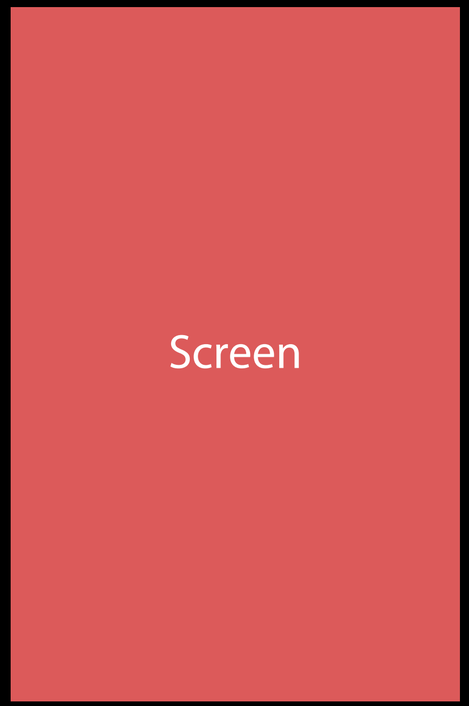

Size classes

# Propagating Changes

## UITraitEnvironment

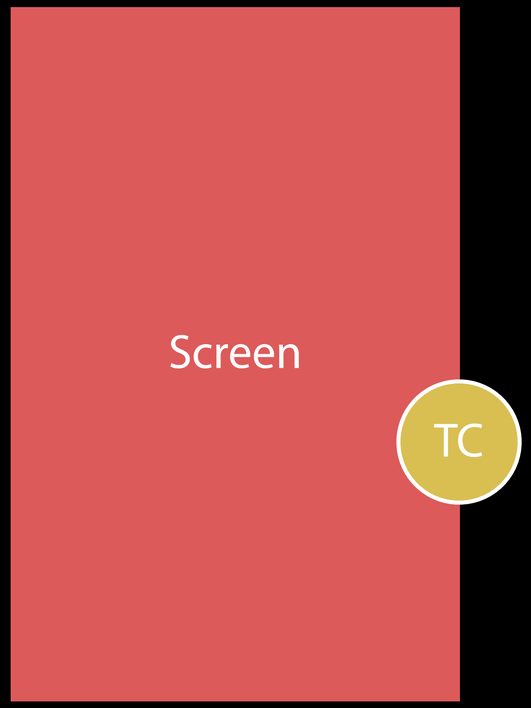
# Propagating Changes

## UITraitEnvironment



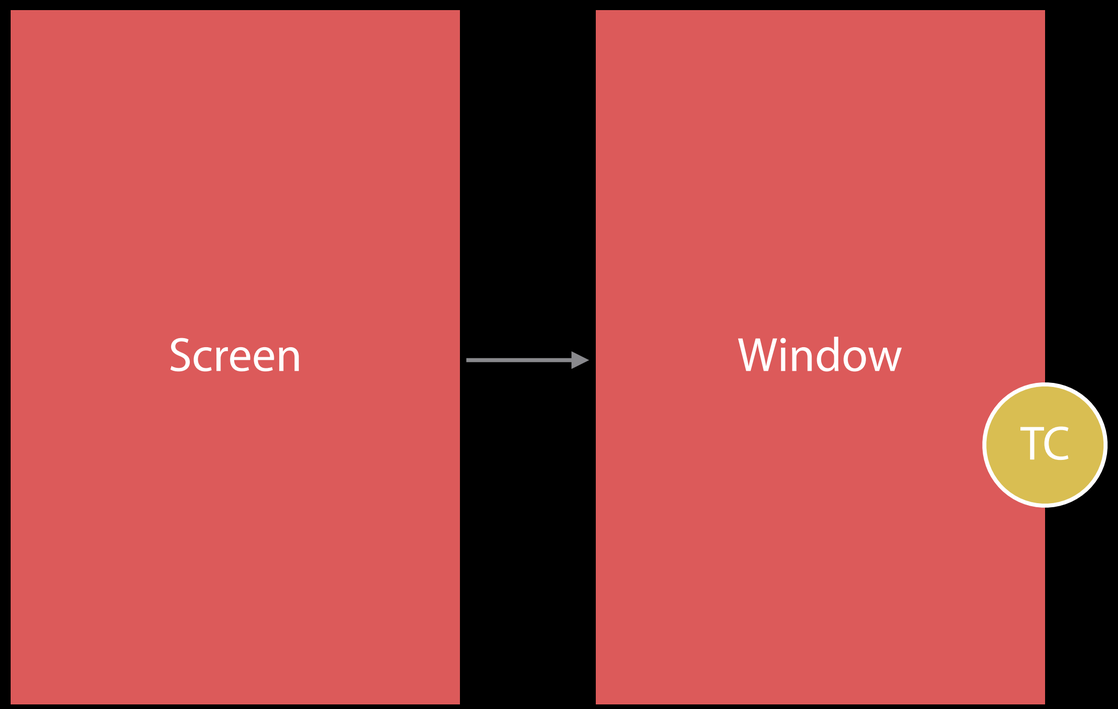
# Propagating Changes

## UITraitEnvironment



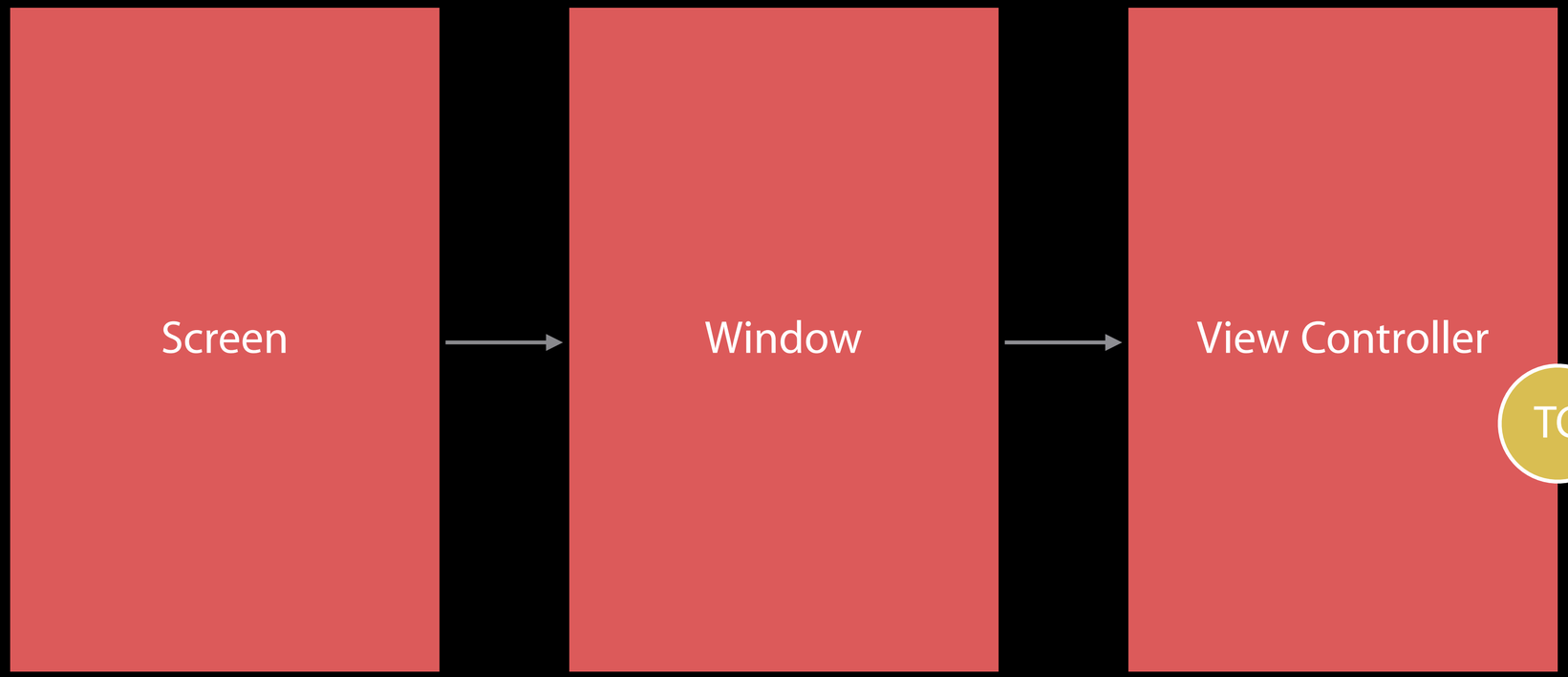
# Propagating Changes

## UITraitEnvironment



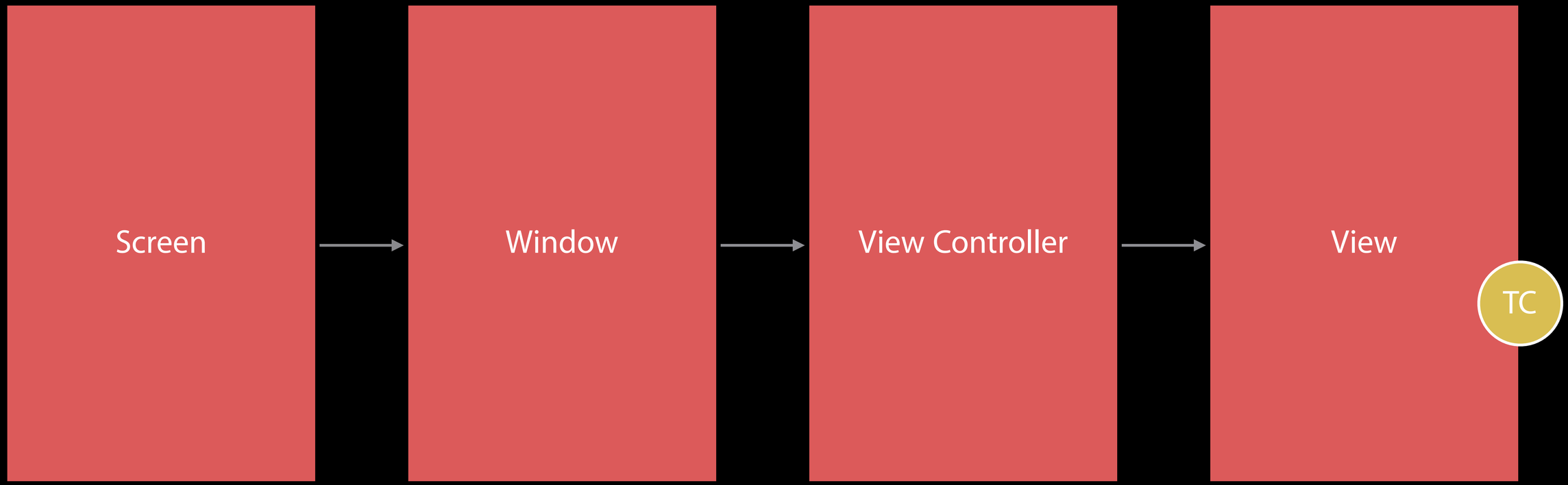
# Propagating Changes

## UITraitEnvironment



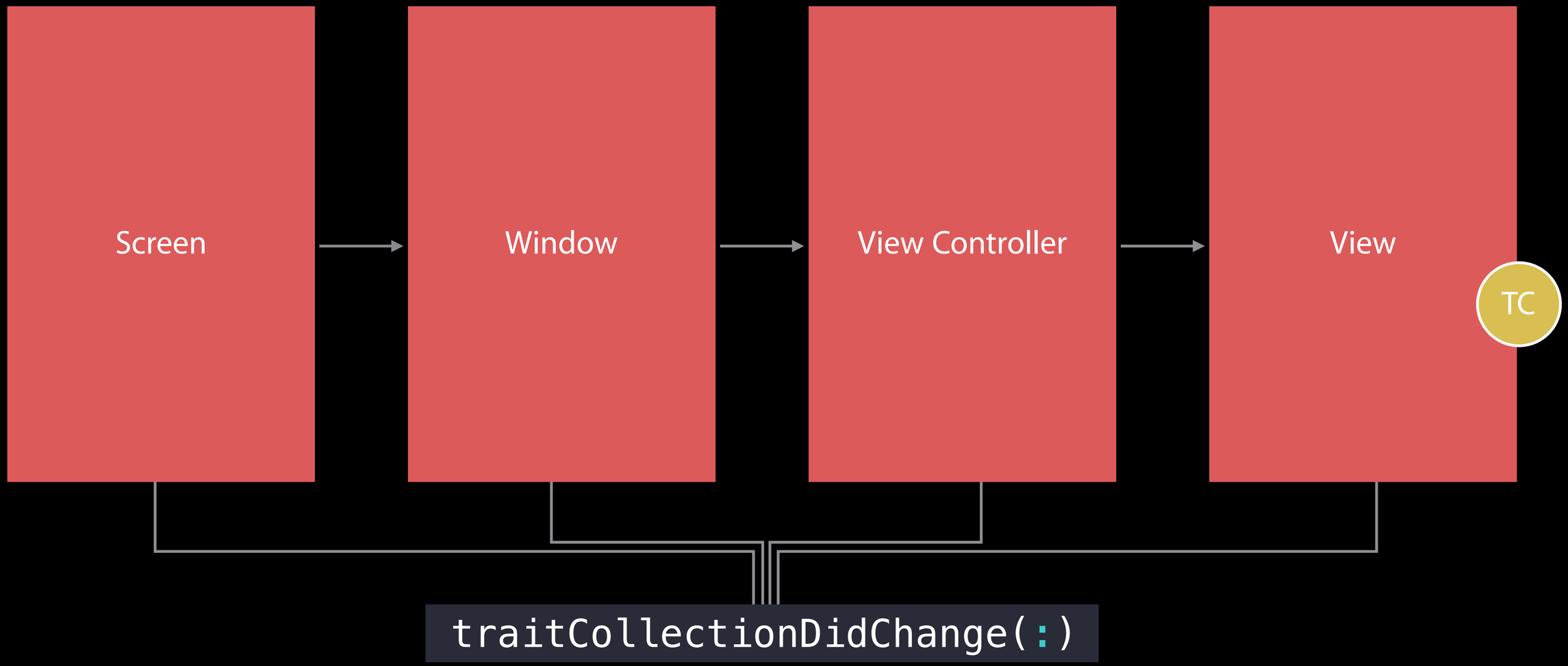
# Propagating Changes

## UITraitEnvironment



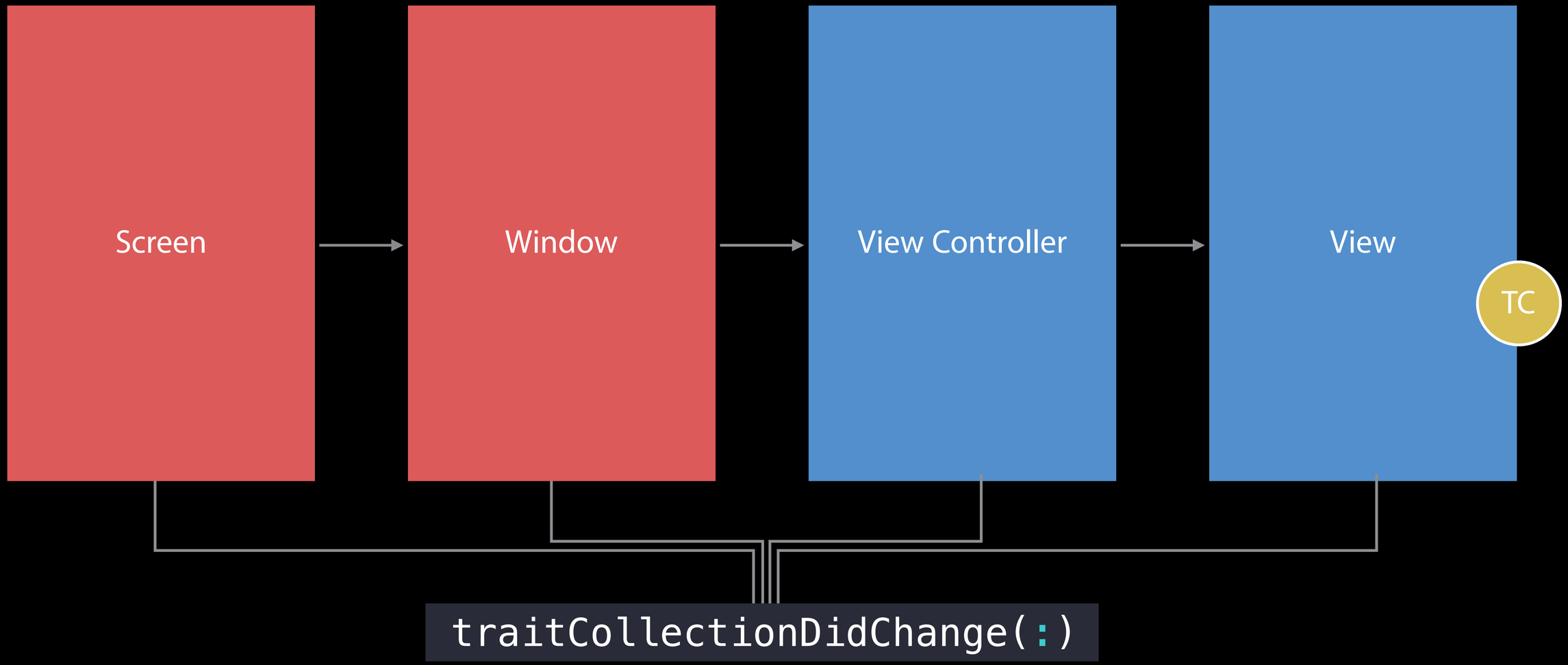
# Propagating Changes

## UITraitEnvironment



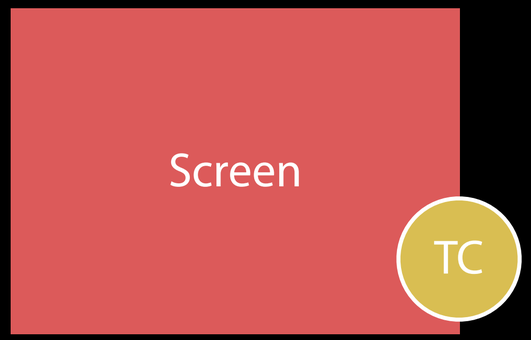
# Propagating Changes

## UITraitEnvironment



# Propagating Changes

## Traits can change during propagation

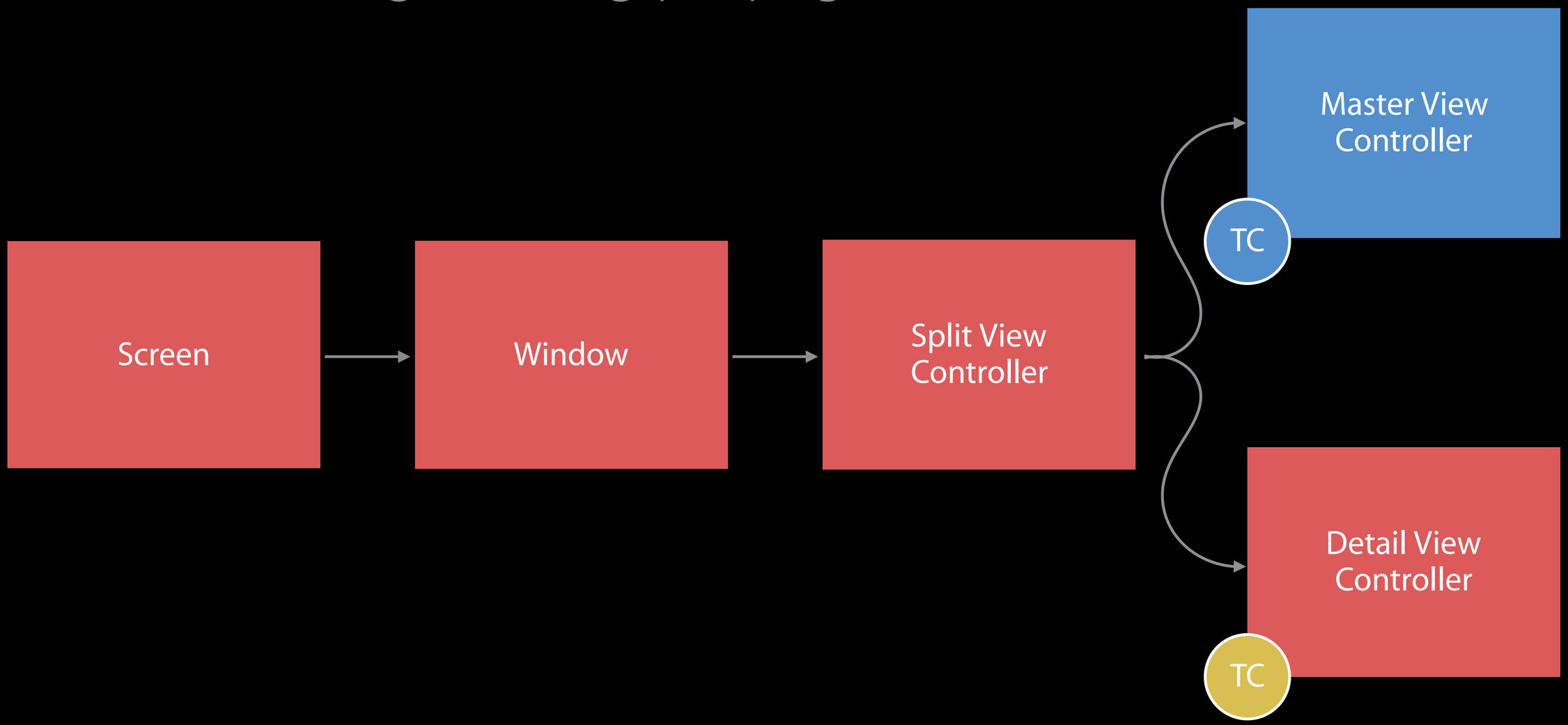


# Propagating Changes

## Traits can change during propagation

# Propagating Changes

## Traits can change during propagation



```swift
override func traitCollectionDidChange(_ previousTraits: UITraitCollection?) {
    super.traitCollectionDidChange(previousTraits)

    if previousTraits?.horizontalSizeClass != traitCollection.horizontalSizeClass {
        switch traitCollection.horizontalSizeClass {
        case .compact:
            setupConstraintsForCompactEnvironment()
        case .unspecified: fallthrough
        case .regular:
            setupConstraintsForRegularEnvironment()
        }
    }
}
```

```swift
override func traitCollectionDidChange(_ previousTraits: UITraitCollection?) {
    super.traitCollectionDidChange(previousTraits)

    if previousTraits?.horizontalSizeClass != traitCollection.horizontalSizeClass {
        switch traitCollection.horizontalSizeClass {
        case .compact:
            setupConstraintsForCompactEnvironment()
        case .unspecified: fallthrough
        case .regular:
            setupConstraintsForRegularEnvironment()
        }
    }
}
```

```swift
override func traitCollectionDidChange(_ previousTraits: UITraitCollection?) {
    super.traitCollectionDidChange(previousTraits)

    if previousTraits?.horizontalSizeClass != traitCollection.horizontalSizeClass {
        switch traitCollection.horizontalSizeClass {
        case .compact:
            setupConstraintsForCompactEnvironment()
        case .unspecified: fallthrough
        case .regular:
            setupConstraintsForRegularEnvironment()
        }
    }
}
```

```swift
override func traitCollectionDidChange(_ previousTraits: UITraitCollection?) {
    super.traitCollectionDidChange(previousTraits)

    if previousTraits?.horizontalSizeClass != traitCollection.horizontalSizeClass {
        switch traitCollection.horizontalSizeClass {
        case .compact:
            setupConstraintsForCompactEnvironment()
        case .unspecified: fallthrough
        case .regular:
            setupConstraintsForRegularEnvironment()
        }
    }
}
```

```swift
override func traitCollectionDidChange(_ previousTraits: UITraitCollection?) {
    super.traitCollectionDidChange(previousTraits)

    if previousTraits?.horizontalSizeClass != traitCollection.horizontalSizeClass {
        switch traitCollection.horizontalSizeClass {
        case .compact:
            setupConstraintsForCompactEnvironment()
        case .unspecified: fallthrough
        case .regular:
            setupConstraintsForRegularEnvironment()
        }
    }
}
```

# traitCollectionDidChange(:)

# traitCollectionDidChange(:)

Called for each `UITraitEnvironment`

# traitCollectionDidChange(:)

Called for each `UITraitEnvironment`

Override and check for specific trait changes

# traitCollectionDidChange(:)

Called for each `UITraitEnvironment`

Override and check for specific trait changes

Some systems react to `traitCollectionDidChange(:)` for you

# traitCollectionDidChange(:)

Called for each `UITraitEnvironment`

Override and check for specific trait changes

Some systems react to `traitCollectionDidChange(:)` for you

- Interface Builder

# traitCollectionDidChange(:)

Called for each `UITraitEnvironment`

Override and check for specific trait changes

Some systems react to `traitCollectionDidChange(:)` for you

- Interface Builder
- Asset catalog

# traitCollectionDidChange(:)

Called for each `UITraitEnvironment`

Override and check for specific trait changes

Some systems react to `traitCollectionDidChange(:)` for you

- Interface Builder
- Asset catalog
- `UIAppearance`

# Takeaways

# Takeaways

Traits describe environment

# Takeaways

Traits describe environment

- Layout, appearance, capabilities

# Takeaways

Traits describe environment

- Layout, appearance, capabilities

Override `traitCollectionDidChange:` to react to trait changes

# Takeaways

Traits describe environment

- Layout, appearance, capabilities

Override `traitCollectionDidChange:` to react to trait changes

Size classes describe experience

# Takeaways

Traits describe environment

- Layout, appearance, capabilities

Override `traitCollectionDidChange:` to react to trait changes

Size classes describe experience

System is going to do most of the work for you

# *Demo*

Brent Shank

# Recap

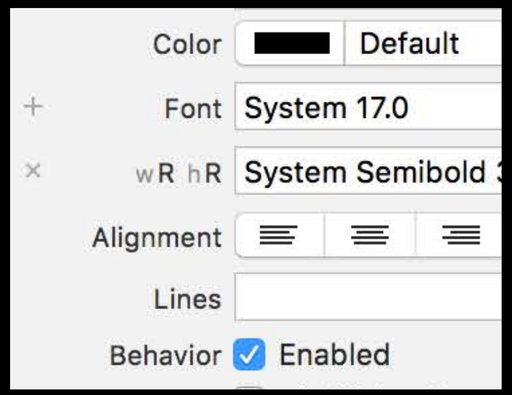
## Building adaptive apps in Interface Builder

# Recap

## Building adaptive apps in Interface Builder

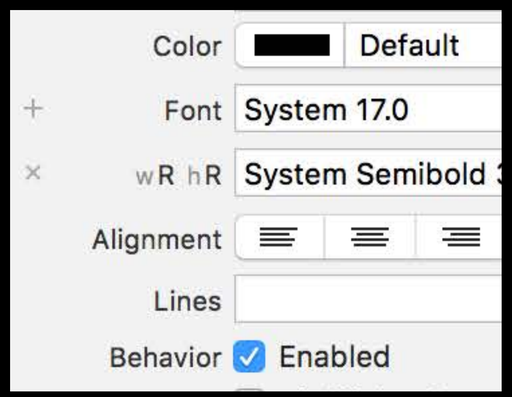
# Recap

## Building adaptive apps in Interface Builder

# Recap

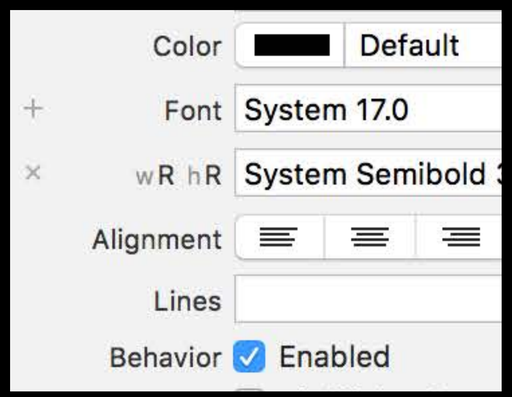
## Building adaptive apps in Interface Builder


# Recap

## Building adaptive apps in Interface Builder





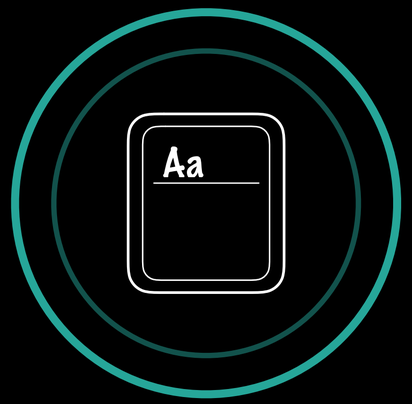

# Summary

# Summary

Traits describe environment

Override `traitCollectionDidChange:` to react to trait changes

Size classes describe experience



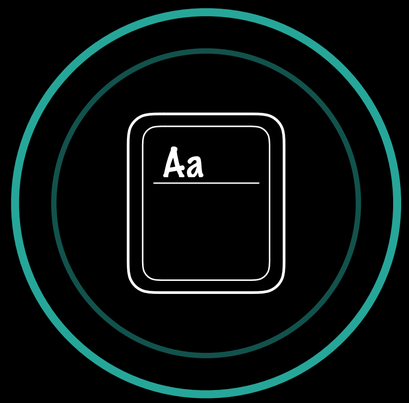
# Summary

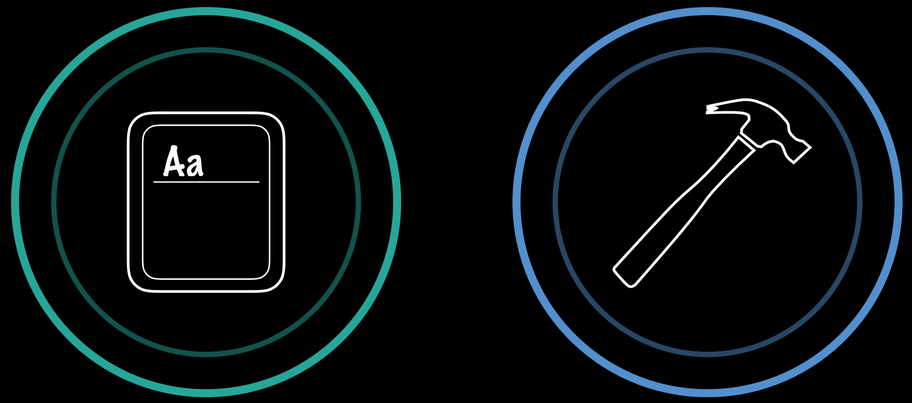
Traits describe environment

Override `traitCollectionDidChange:` to react to trait changes

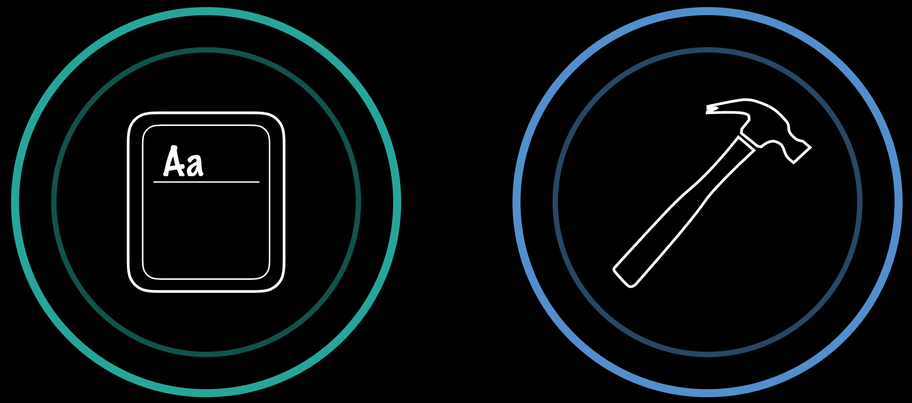
Size classes describe experience

Interface Builder lets you customize on size class and preview on specific configurations



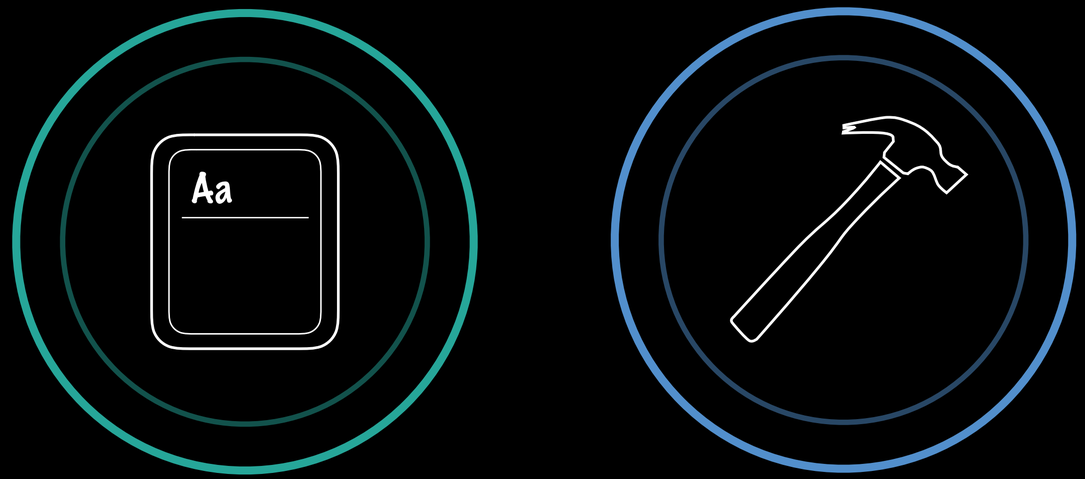




PART 1

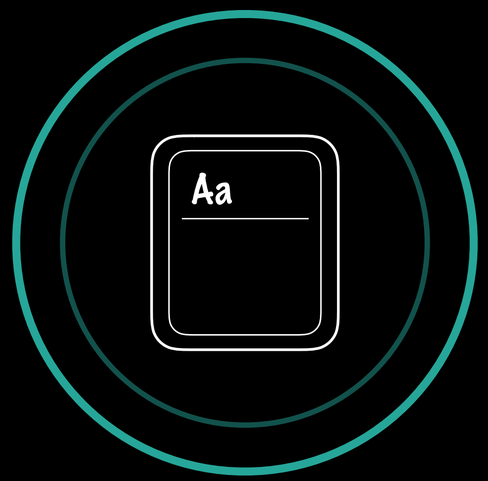



PART 1

# The system is going to do most of the work so you don’t have to.

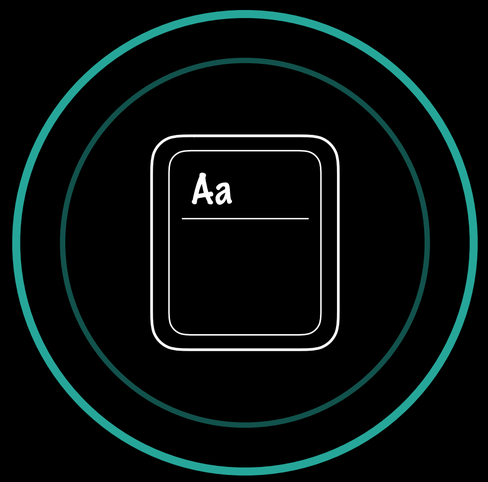



PART 1




PART 1

PART 2




PART 1

PART 2

# Related Sessions

| | | |
|---|---|---|
| Making Apps Adaptive, Part 2 | Presidio | Friday 9:00AM |
| What's New In Auto Layout | Presidio | Friday 3:00PM |

# Labs

| | | |
|---|---|---|
| Interface Builder and Auto Layout Lab | Developer Tools Lab B | Thursday 3:00PM |
| Interface Builder and Auto Layout Lab | Developer Tools Lab C | Friday 9:00AM |

WWDC16