

Accessibility in iOS

Developing for everyone

Session 202

Sam White

iOS Accessibility

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Accessibility

Supporting different abilities



Agenda

What we will cover

- Accessibility features in iOS
 - What's new
- Accessibility tutorial
 - Basic
 - Advanced
 - New
- Guided Access API



iOS Accessibility Features

VoiceOver



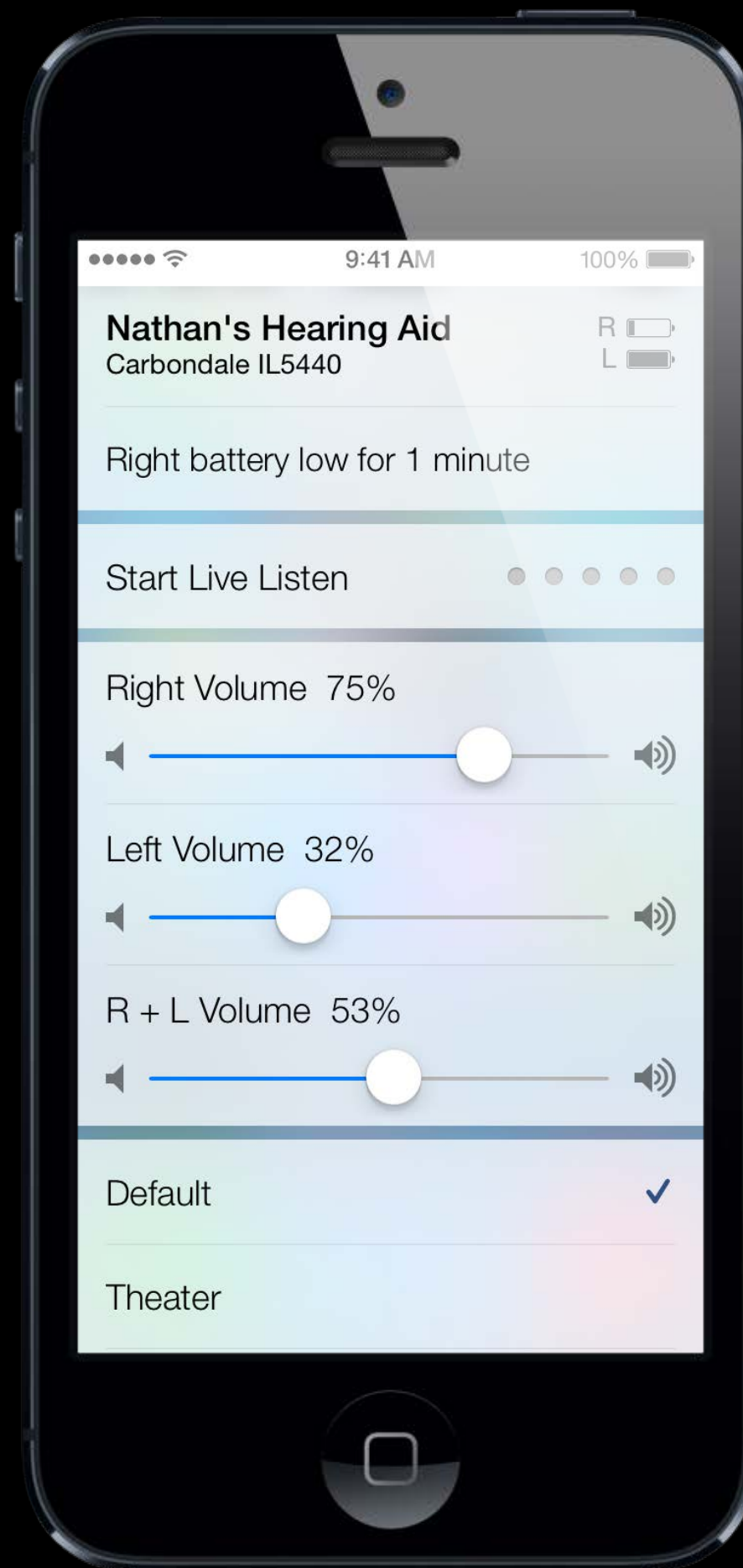
iOS Accessibility Features

Guided Access



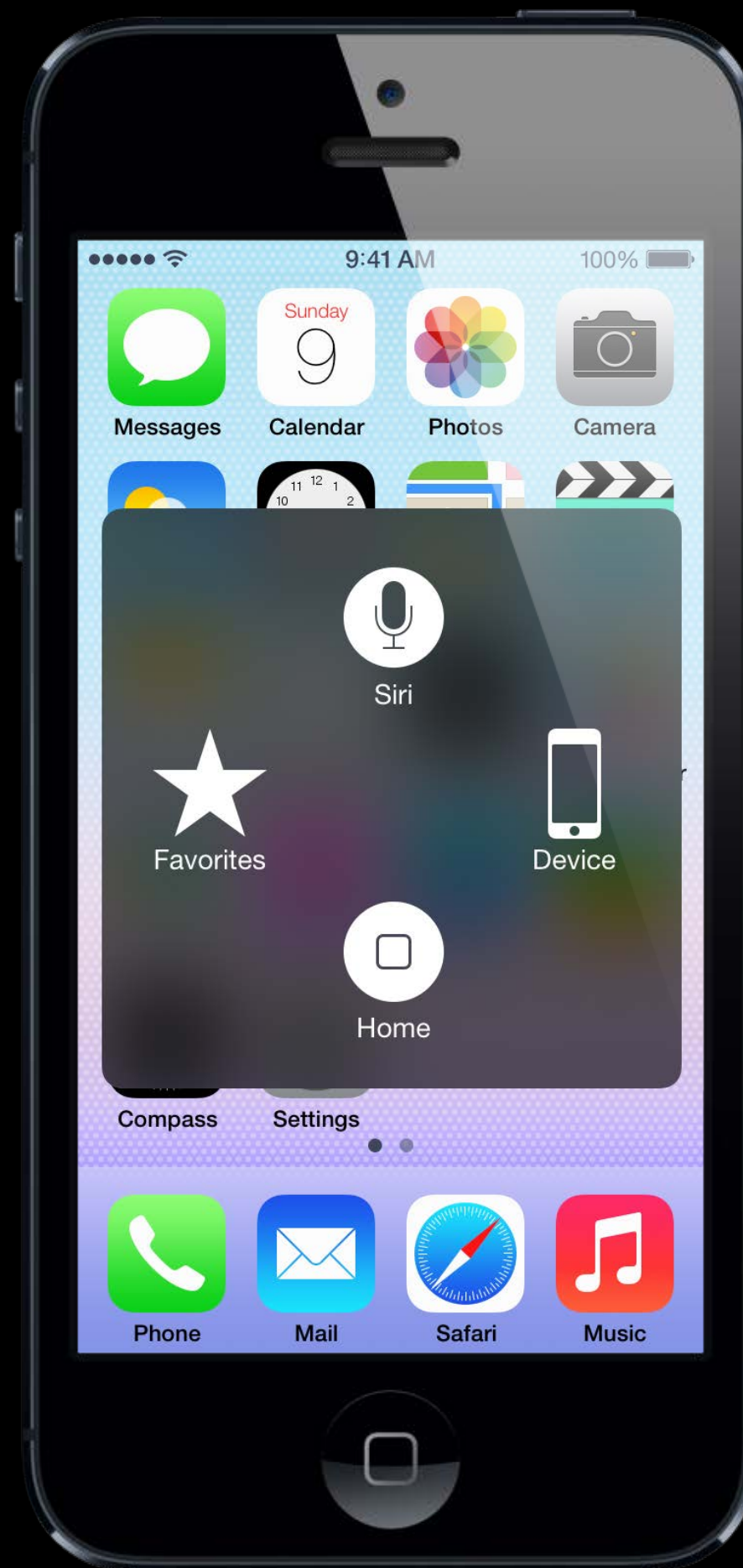
iOS Accessibility Features

Made for iPhone Hearing Aids



iOS Accessibility Features

AssistiveTouch



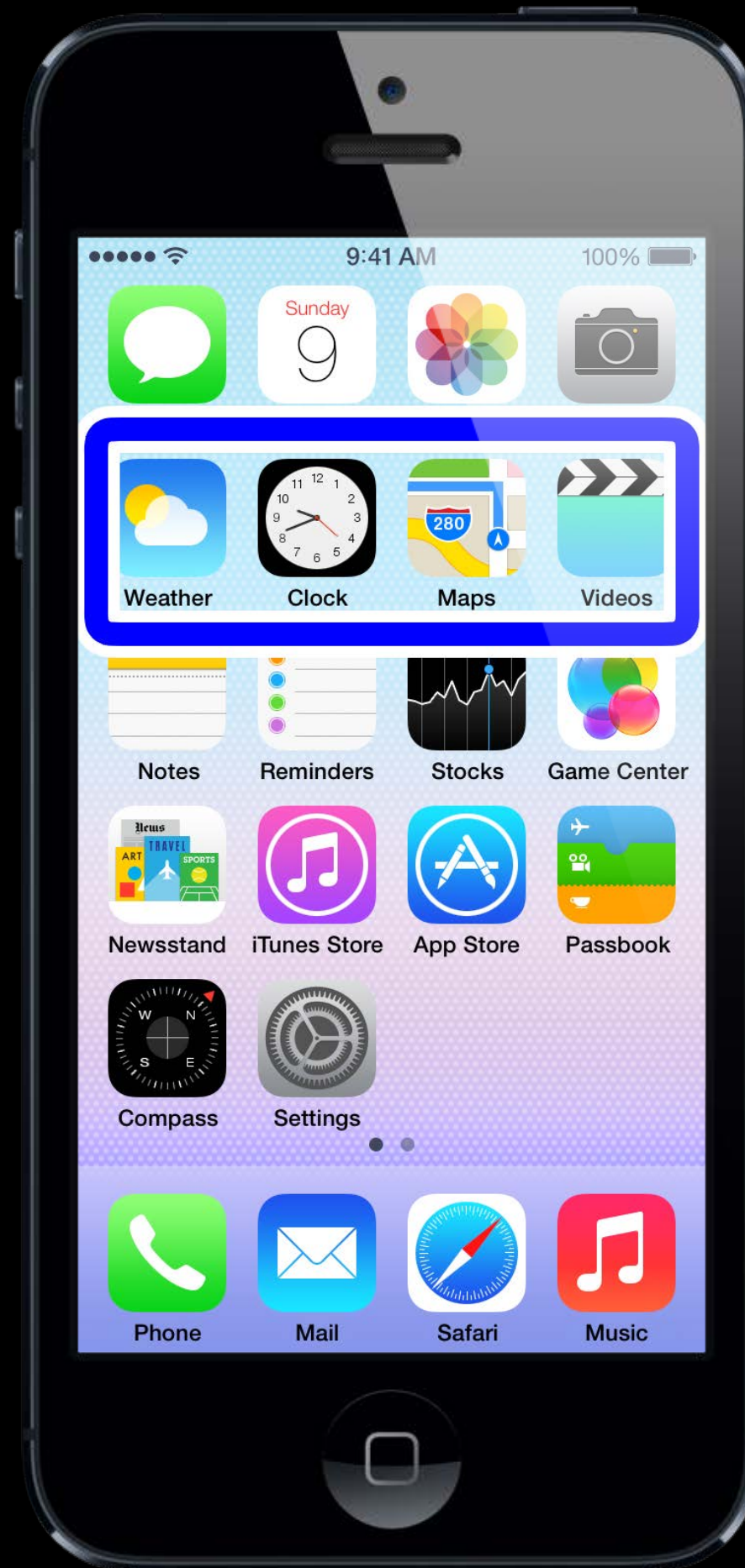
Accessibility

Supporting different abilities



New in iOS 7

Switch Control



Demo

Switch Control

Accessibility Tutorial

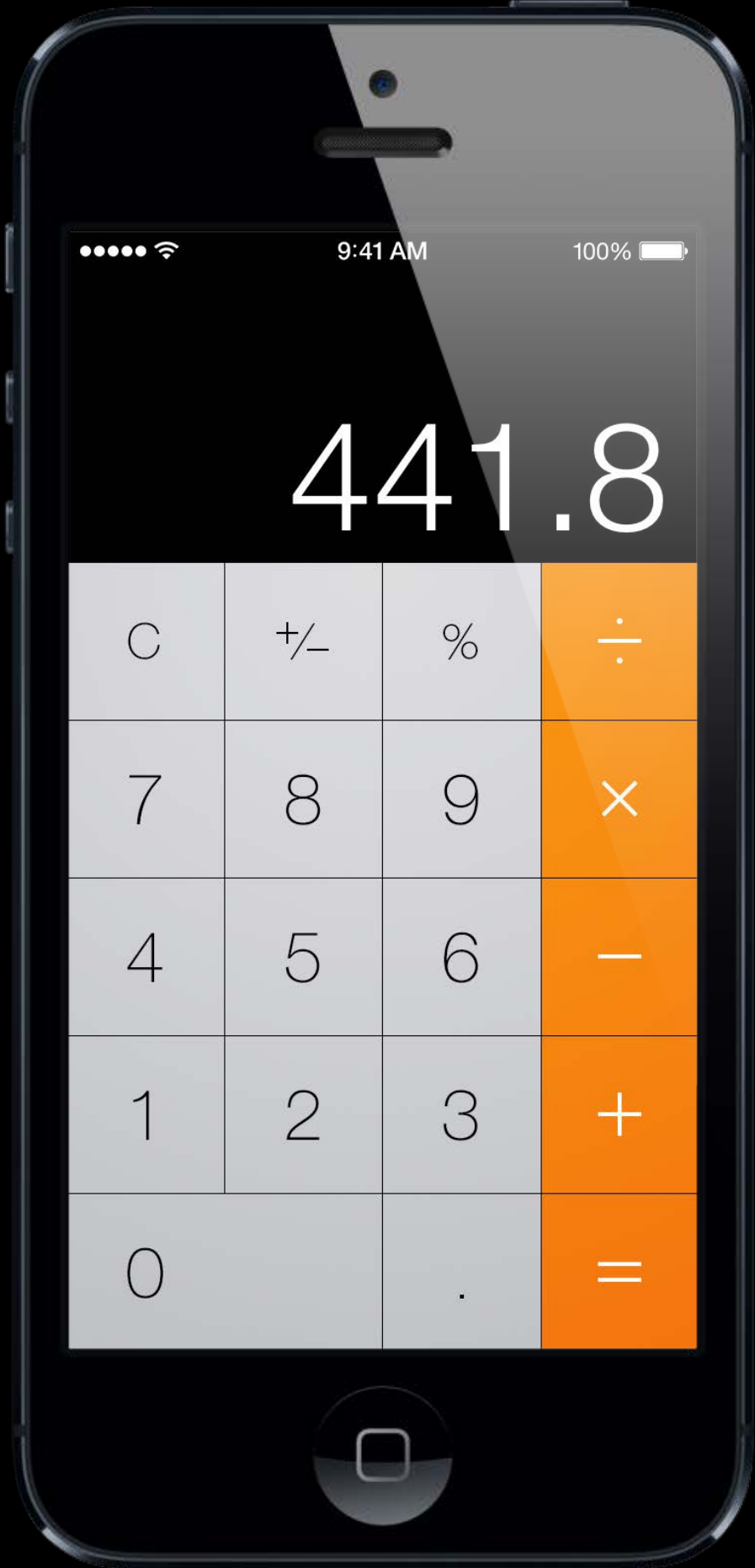
What you'll learn

- UIAccessibility
- Accessibility audit
- UIAccessibility API
 - Basic
 - Advanced
 - New
- Guided Access API



UIAccessibility

VoiceOver

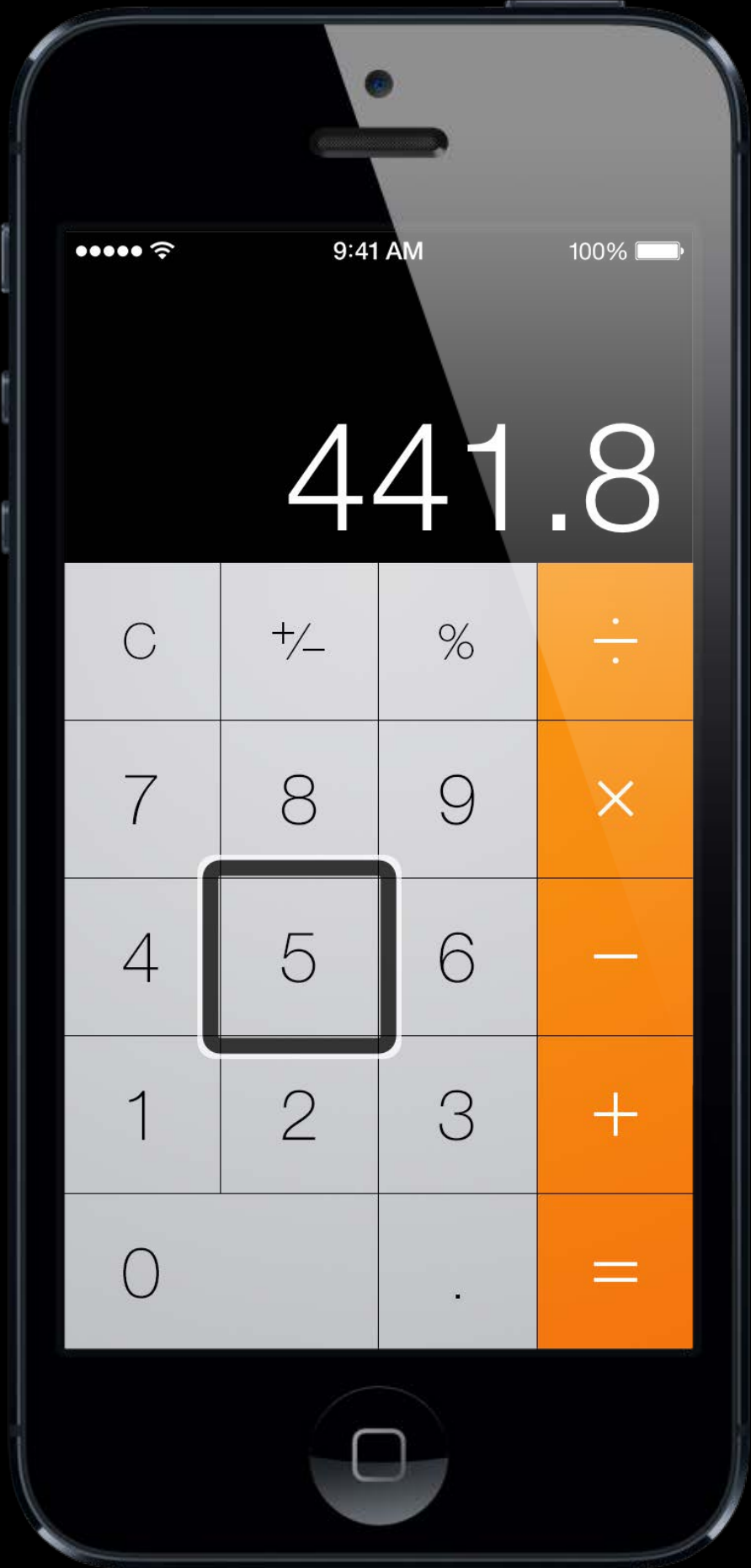


UIAccessibility

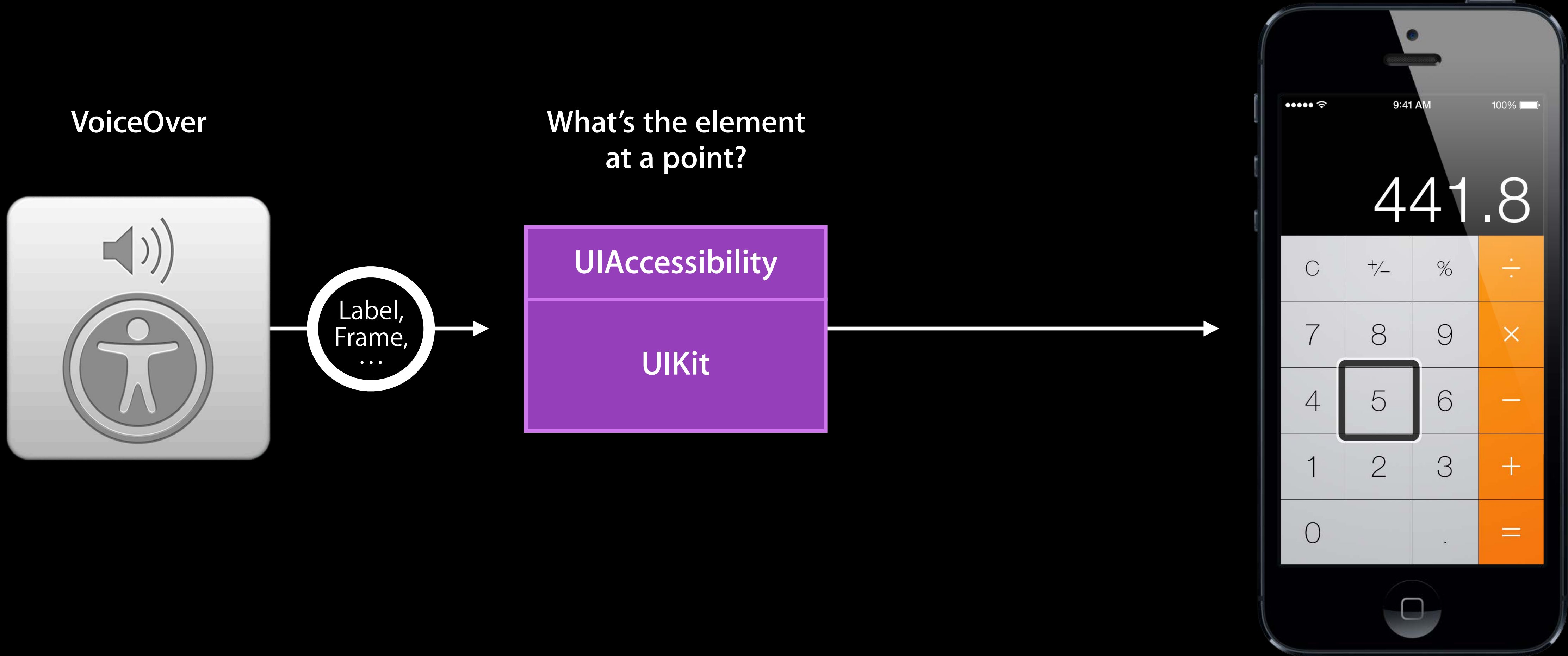
VoiceOver



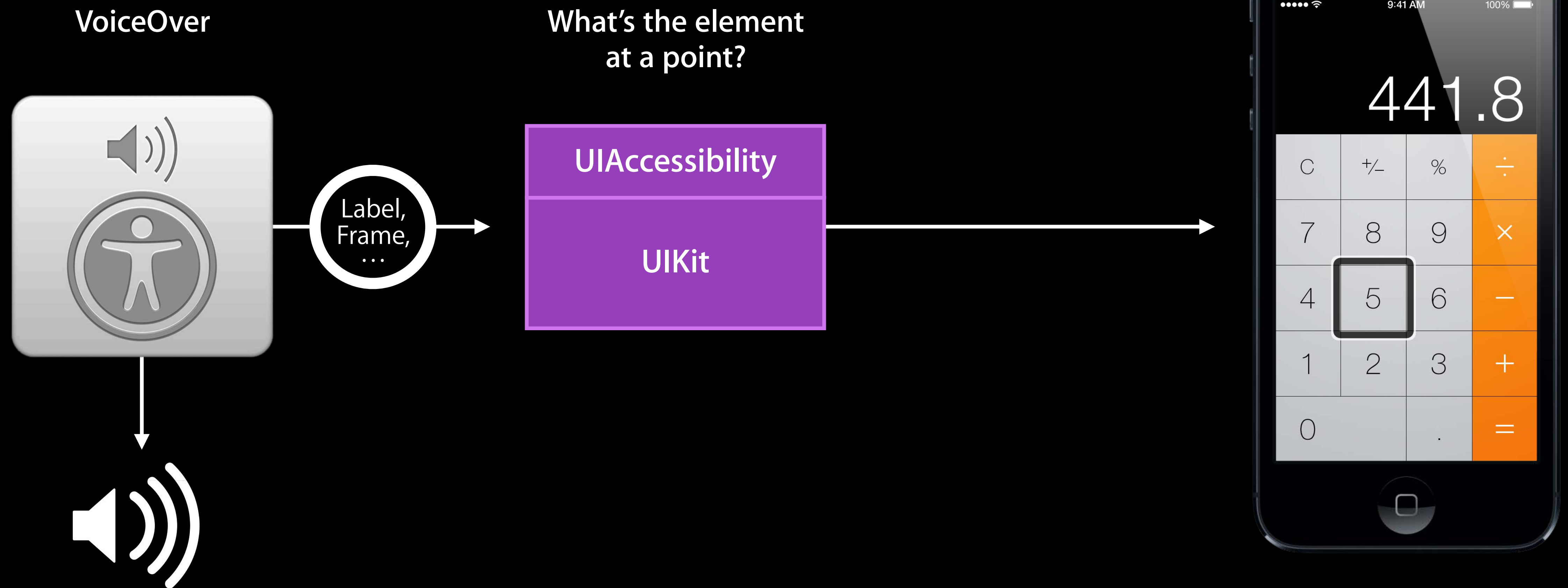
What's the element at a point?



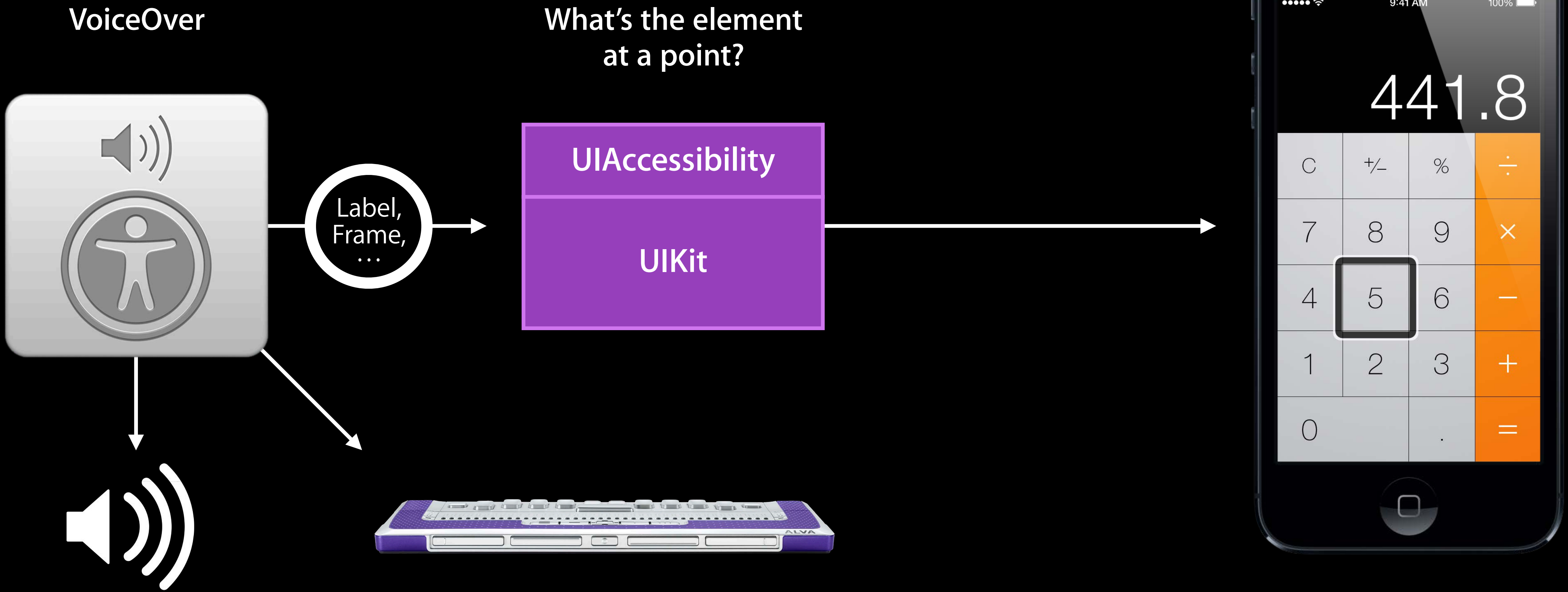
UIAccessibility



UIAccessibility

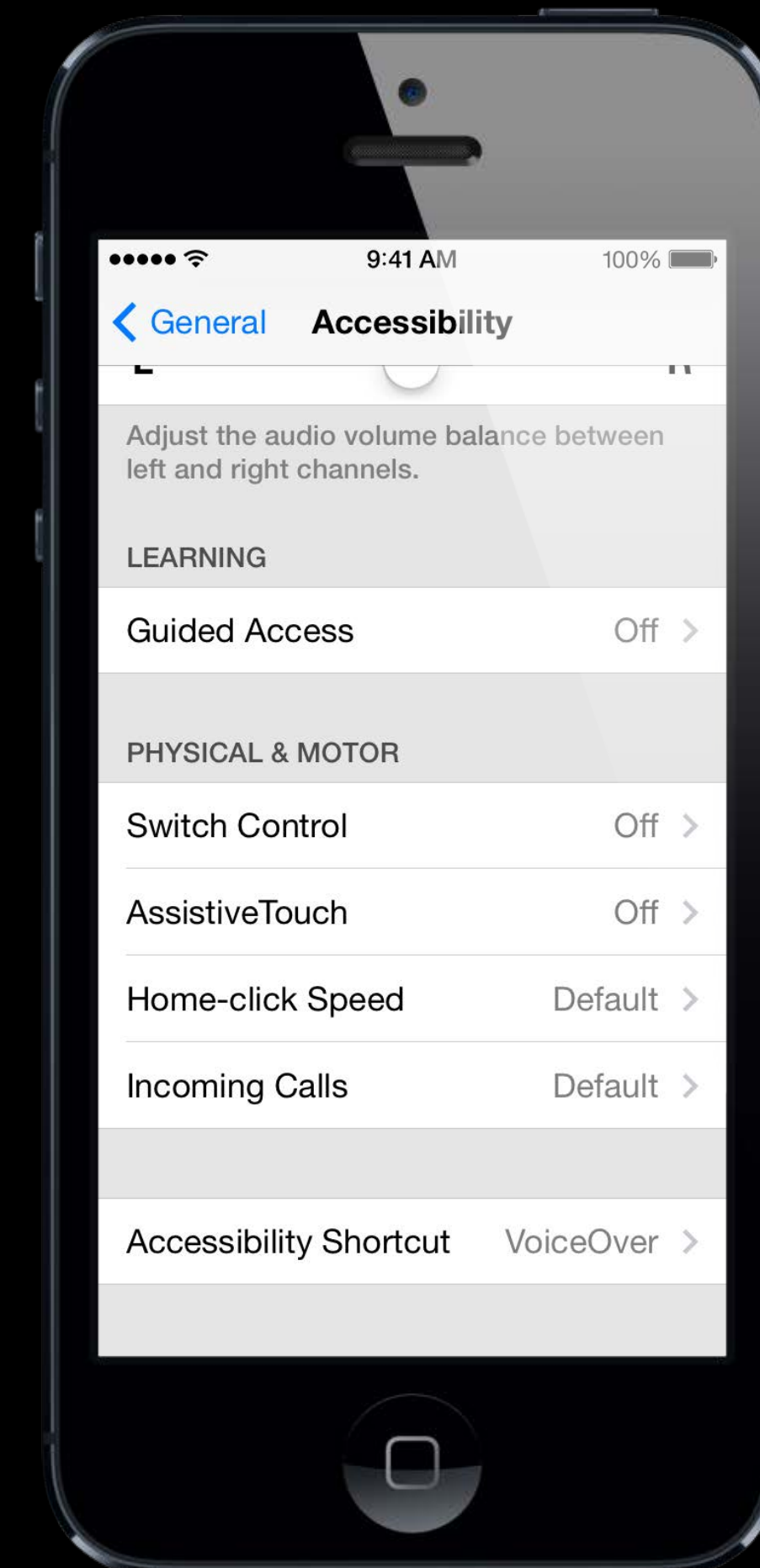


UIAccessibility



Accessibility Audit

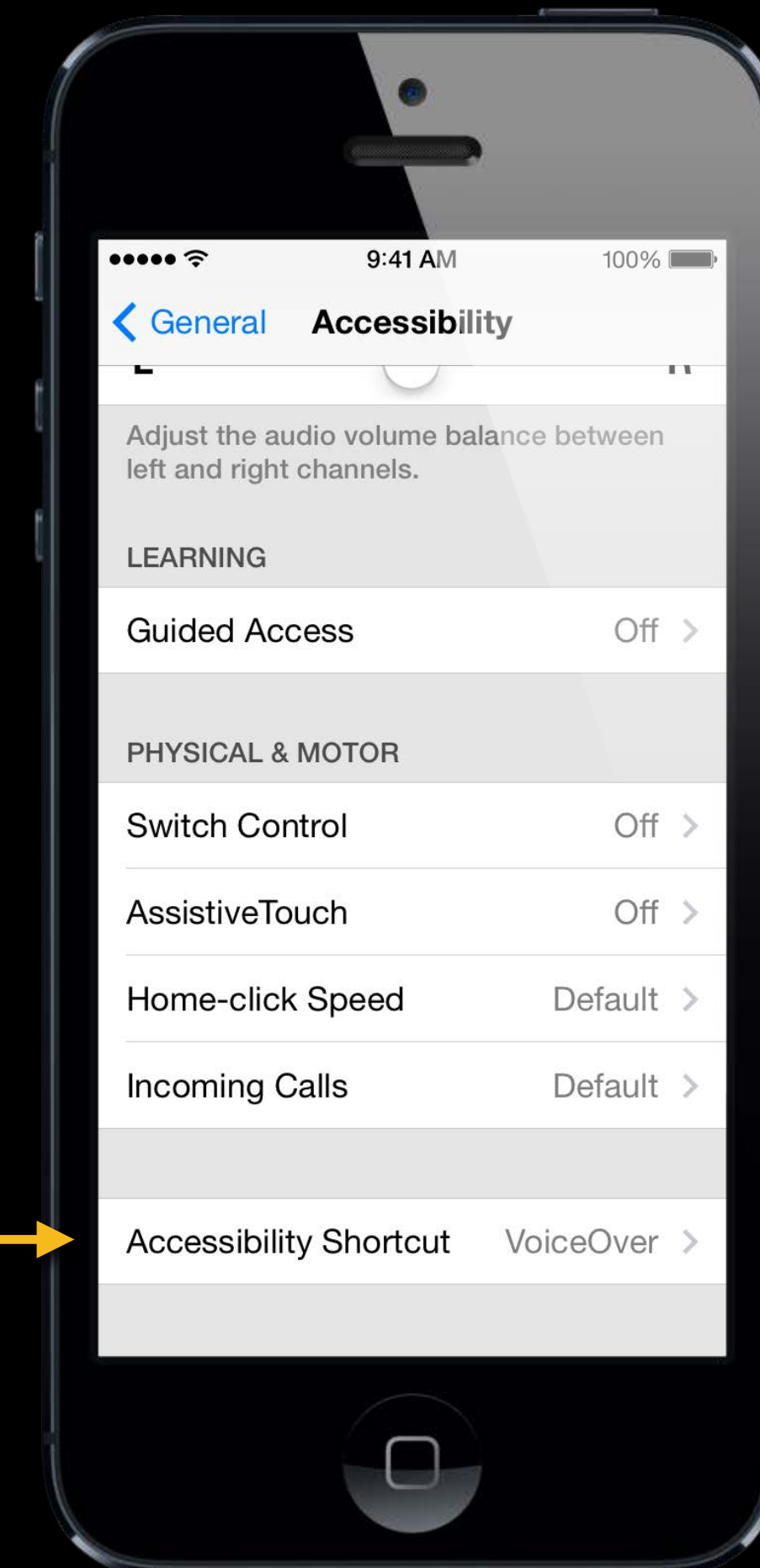
- Simulator
 - Accessibility inspector
- Device
 - Accessibility feature
- Settings > General > Accessibility



Accessibility Audit

- Simulator
 - Accessibility inspector
- Device
 - Accessibility feature
- Settings > General > Accessibility

Accessibility Shortcut →



Demo

Accessibility audit

Basic Accessibility Attributes

UIAccessibility.h

`@property BOOL isAccessibilityElement`

- Return YES to make accessibility features see this element
- Default is YES for UIKit controls and labels

`@property (copy) NSString *accessibilityLabel`

- Textual representation of the element

Basic Accessibility Attributes

UIAccessibility.h

`@property (copy) NSString *accessibilityHint`

- Provides additional usage information to users

`@property (copy) NSString *accessibilityValue`

- Textual representation of the element's value

`@property UIAccessibilityTraits accessibilityTraits`

- Defines behavior and intended use
- Bitmask of unsigned integers

Accessibility Traits



Accessibility Traits

Header



Accessibility Traits

Header



Static Text

Accessibility Traits

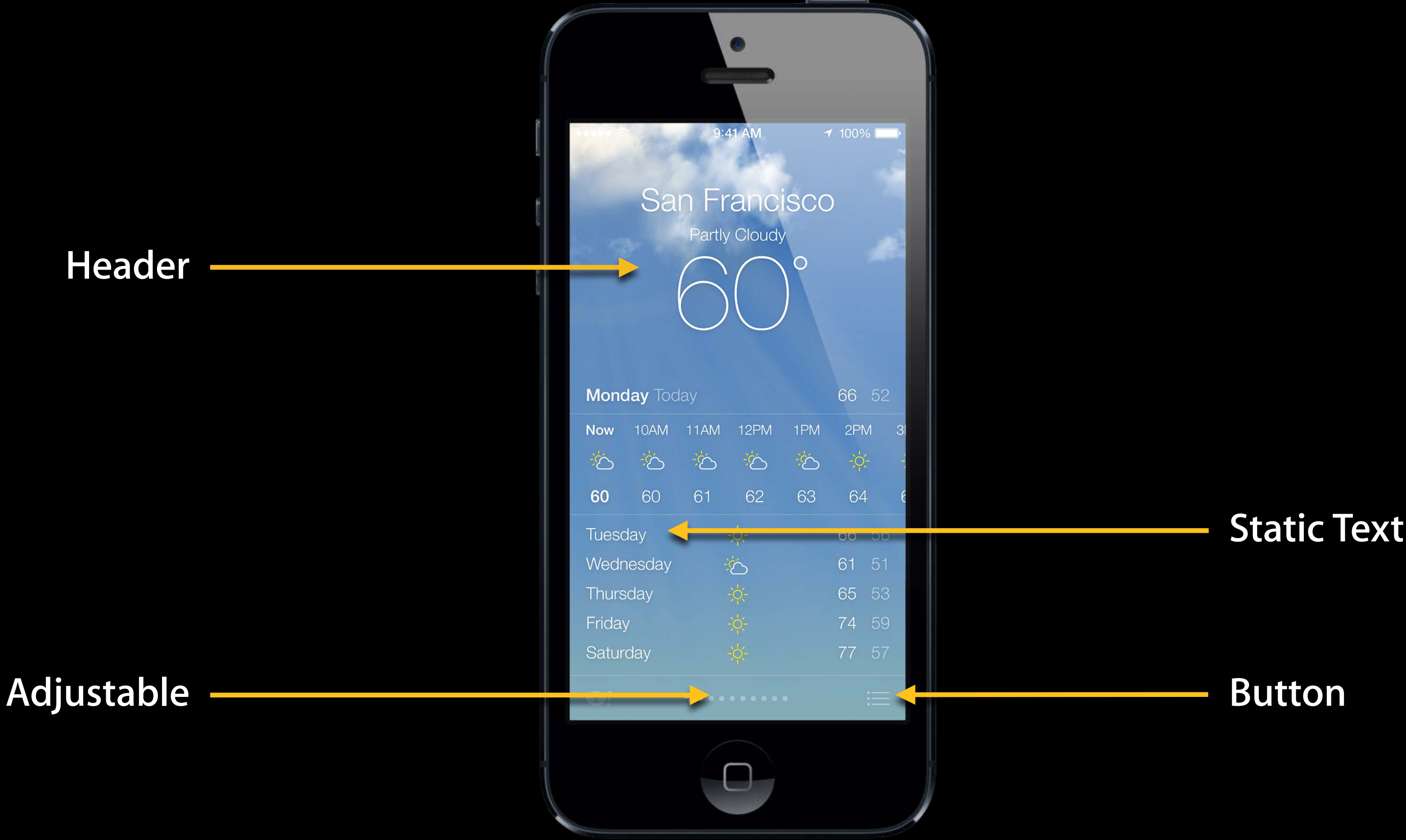
Header



Static Text

Button

Accessibility Traits



Accessibility in Interface Builder

Static accessibility attributes

▼ Accessibility

Accessibility Enabled

Label

Hint

Traits

- Button
- Image
- Static Text
- Search Field
- Plays Sound
- Keyboard Key
- Summary Element
- Updates Frequently
- User Interaction Enabled

Link

Selected

Accessibility in Interface Builder

Static accessibility attributes

▼ Accessibility

Accessibility Enabled

Label

Hint

Traits

- Button
- Image
- Static Text
- Search Field
- Plays Sound
- Keyboard Key
- Summary Element
- Updates Frequently
- User Interaction Enabled

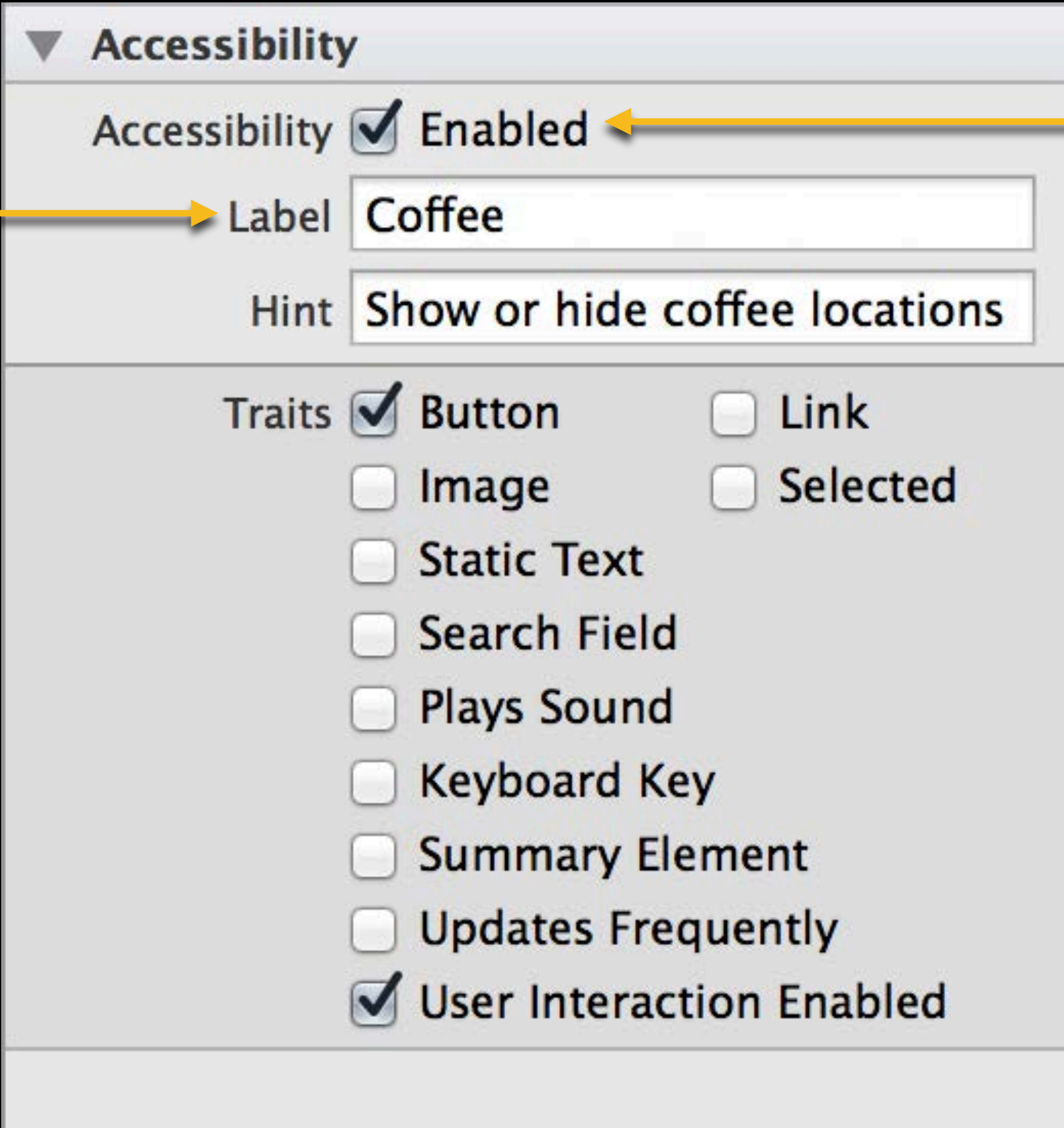
Link

Selected

isAccessibilityElement

Accessibility in Interface Builder

Static accessibility attributes



The screenshot shows the Accessibility panel in Interface Builder. The panel is titled "Accessibility" and contains the following settings:

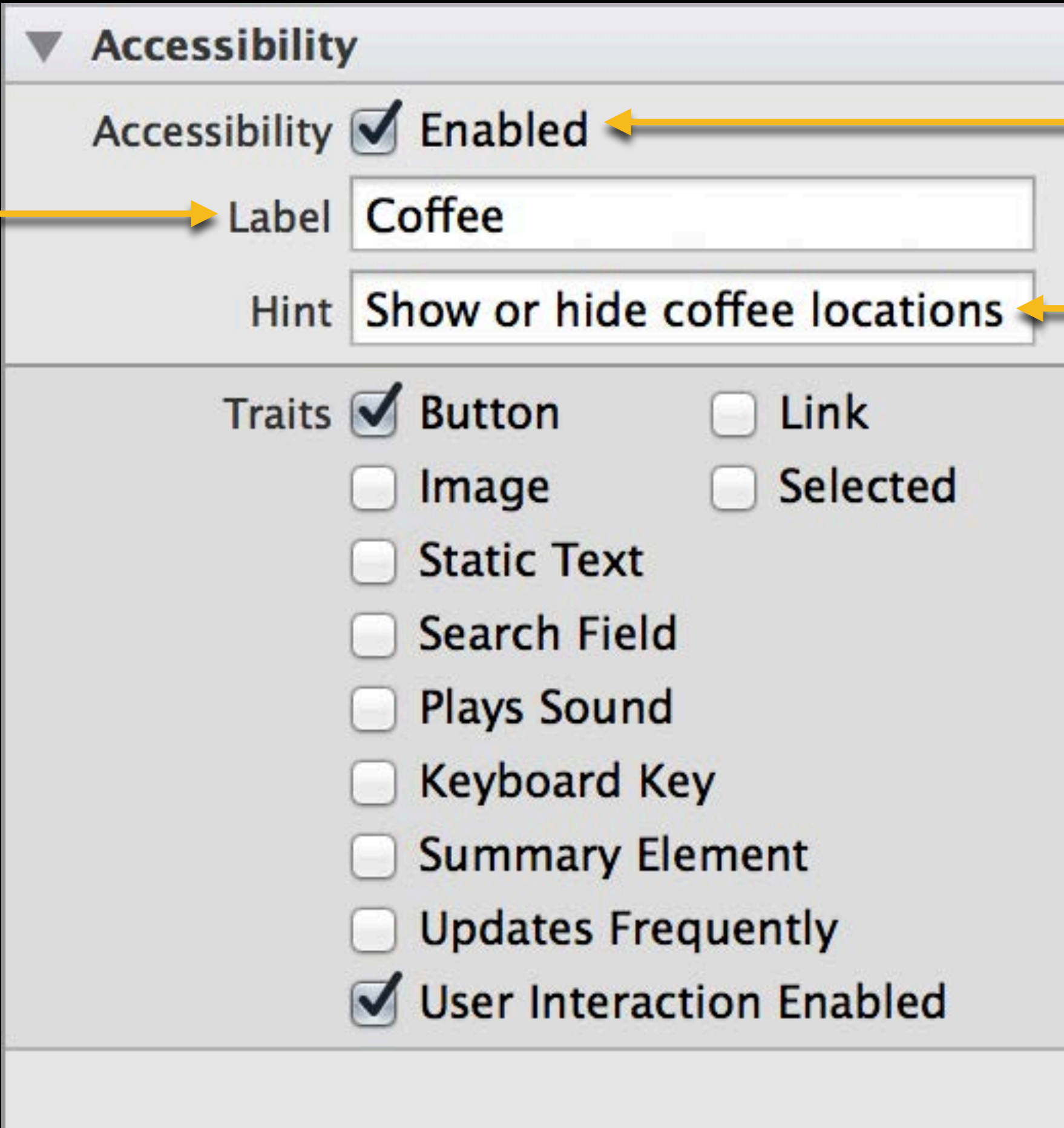
- Accessibility** Enabled
- Label**
- Hint**
- Traits**
 - Button
 - Image
 - Static Text
 - Search Field
 - Plays Sound
 - Keyboard Key
 - Summary Element
 - Updates Frequently
 - User Interaction Enabled
 - Link
 - Selected

Annotations with yellow arrows point to the following elements:

- isAccessibilityElement** points to the "Accessibility" checkbox.
- accessibilityLabel** points to the "Label" text field.

Accessibility in Interface Builder

Static accessibility attributes



The screenshot shows the Accessibility panel in Interface Builder. It is divided into three main sections: Accessibility, Label, and Traits. The Accessibility section has a checkbox for 'Accessibility' which is checked and labeled 'Enabled'. The Label section has a text field containing 'Coffee'. The Traits section has a list of checkboxes for various traits: Button (checked), Link, Image, Selected, Static Text, Search Field, Plays Sound, Keyboard Key, Summary Element, Updates Frequently, and User Interaction Enabled (checked). Three yellow arrows point from external text labels to specific elements in the panel: 'isAccessibilityElement' points to the 'Enabled' checkbox, 'accessibilityLabel' points to the 'Coffee' text field, and 'accessibilityHint' points to the 'Show or hide coffee locations' text field.

Accessibility Enabled ← isAccessibilityElement

accessibilityLabel → Label Coffee

Hint Show or hide coffee locations ← accessibilityHint

Traits Button Link
 Image Selected
 Static Text
 Search Field
 Plays Sound
 Keyboard Key
 Summary Element
 Updates Frequently
 User Interaction Enabled

Accessibility in Interface Builder

Static accessibility attributes

The screenshot shows the Accessibility panel in Interface Builder. The panel is titled "Accessibility" and contains several sections:

- Accessibility:** A checkbox labeled "Enabled" is checked. An arrow points from the text "isEnabledAccessibilityElement" to this checkbox.
- Label:** A text field contains the word "Coffee". An arrow points from the text "accessibilityLabel" to this field.
- Hint:** A text field contains the text "Show or hide coffee locations". An arrow points from the text "accessibilityHint" to this field.
- Traits:** A list of accessibility traits with checkboxes:
 - Button
 - Image
 - Static Text
 - Search Field
 - Plays Sound
 - Keyboard Key
 - Summary Element
 - Updates Frequently
 - User Interaction EnabledAn arrow points from the text "accessibilityTraits" to the "Traits" label.

Accessibility in Code

Static accessibility attributes

```
- (void)awakeFromNib {  
    UIView *headerView = self.headerView;  
    headerView.isAccessibilityElement = YES;  
    headerView.accessibilityLabel = @"Moscone West";  
    headerView.accessibilityTraits = UIAccessibilityTraitHeader;  
}
```


Accessibility in Code

Dynamic accessibility attributes

```
- (NSString *)accessibilityLabel {  
    return @"Floor";  
}  
  
- (NSString *)accessibilityValue {  
    int floor = self.floor;  
    return [NSString stringWithFormat:@"%d", floor];  
}
```

Demo

Basic UIAccessibility API

Advanced Accessibility

UIAccessibilityContainer protocol

```
- (NSInteger)accessibilityElementCount {  
    return self.rooms.count;  
}  
  
- (id)accessibilityElementAtIndex:(NSInteger)index {  
    return [self.rooms objectAtIndex:index];  
}  
  
- (NSInteger)indexOfAccessibilityElement:(id)element {  
    return [self.rooms indexOfObject:element];  
}
```

Advanced Accessibility

UIAccessibilityContainer protocol

```
- (NSArray *)rooms {
    if ( _rooms != nil ) {
        return _rooms;
    }
    _rooms = [[NSMutableArray alloc] init];
    ...
    CGRect roomFrame = [self frameForRoomAtIndex:i];

    [_rooms addObject:room];
    ...
    return _rooms;
}
```


Advanced Accessibility

UIAccessibilityContainer protocol

```
- (NSArray *)rooms {
    if ( _rooms != nil ) {
        return _rooms;
    }
    _rooms = [[NSMutableArray alloc] init];
    ...
    CGRect roomFrame = [self frameForRoomAtIndex:i];
    UIAccessibilityElement *room =
    [
        [_rooms addObject:room];
        ...
        return _rooms;
    ]
}
```

Advanced Accessibility

UIAccessibilityContainer protocol

```
- (NSArray *)rooms {
    if ( _rooms != nil ) {
        return _rooms;
    }
    _rooms = [[NSMutableArray alloc] init];
    ...
    CGRect roomFrame = [self frameForRoomAtIndex:i];
    UIAccessibilityElement *room =
        [[UIAccessibilityElement alloc] initWithAccessibilityContainer:self];
    [_rooms addObject:room];
    ...
    return _rooms;
}
```

Advanced Accessibility

UIAccessibilityContainer protocol

```
- (NSArray *)rooms {
    if ( _rooms != nil ) {
        return _rooms;
    }
    _rooms = [[NSMutableArray alloc] init];
    ...
    CGRect roomFrame = [self frameForRoomAtIndex:i];
    UIAccessibilityElement *room =
        [[UIAccessibilityElement alloc] initWithAccessibilityContainer:self];
    room.accessibilityFrame =
        [_rooms addObject:room];
    ...
    return _rooms;
}
```

Advanced Accessibility

UIAccessibilityContainer protocol

```
- (NSArray *)rooms {  
    if ( _rooms != nil ) {  
        return _rooms;  
    }  
    _rooms = [[NSMutableArray alloc] init];  
    ...  
    CGRect roomFrame = [self frameForRoomAtIndex:i];  
    UIAccessibilityElement *room =  
        [[UIAccessibilityElement alloc] initWithAccessibilityContainer:self];  
    room.accessibilityFrame =  
        UIAccessibilityConvertFrameToScreenCoordinates(roomFrame, self.view);  
    [_rooms addObject:room];  
    ...  
    return _rooms;  
}
```


Advanced Accessibility

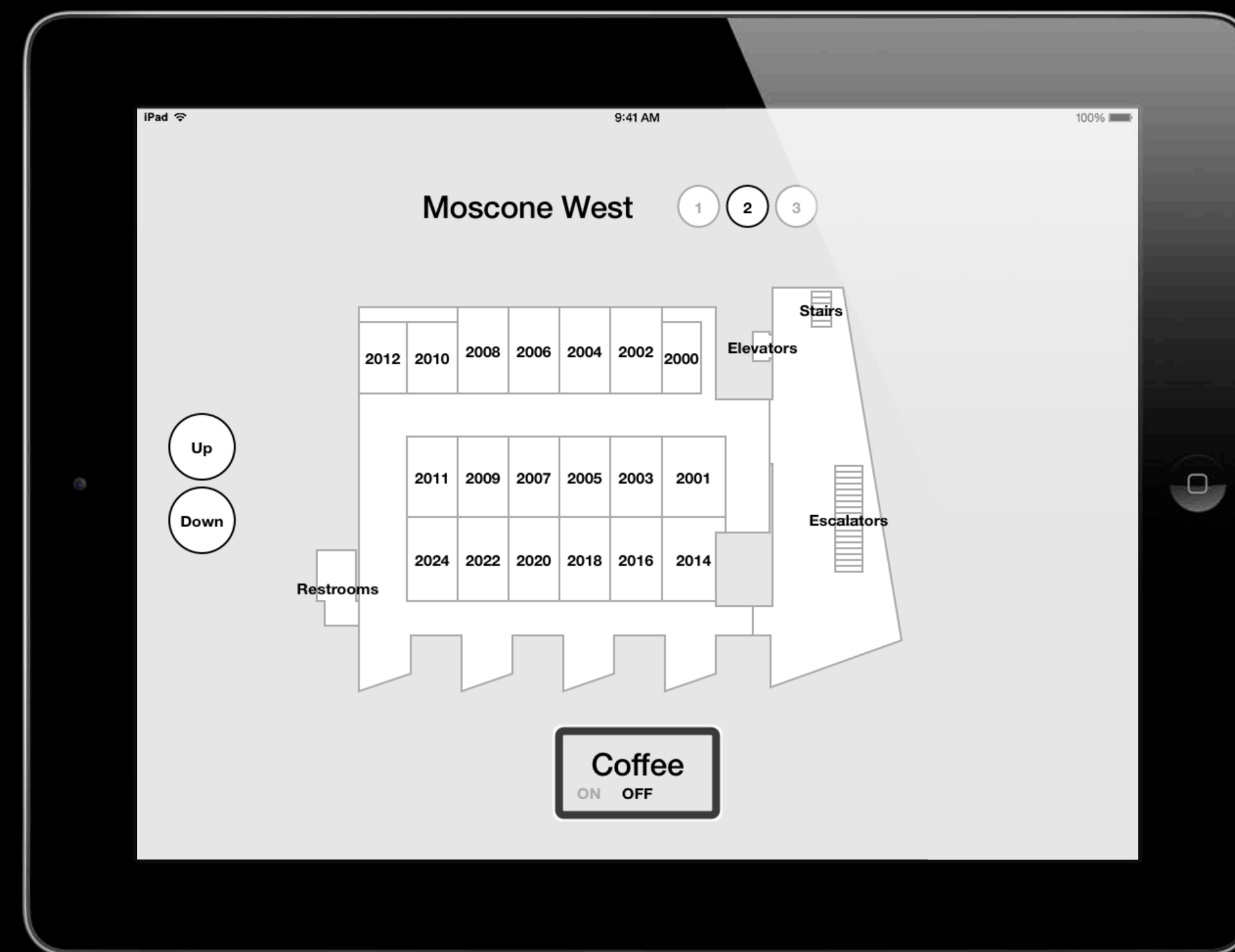
UIAccessibilityContainer protocol

```
- (NSArray *)rooms {
    if ( _rooms != nil ) {
        return _rooms;
    }
    _rooms = [[NSMutableArray alloc] init];
    ...
    CGRect roomFrame = [self frameForRoomAtIndex:i];
    UIAccessibilityElement *room =
        [[UIAccessibilityElement alloc] initWithAccessibilityContainer:self];
    room.accessibilityFrame =
        UIAccessibilityConvertFrameToScreenCoordinates(roomFrame, self.view);
    room.accessibilityLabel = [self labelForRoomAtIndex:i];
    [_rooms addObject:room];
    ...
    return _rooms;
}
```

Advanced Accessibility



- (BOOL)accessibilityActivate
 - Implement on an element to handle the default action

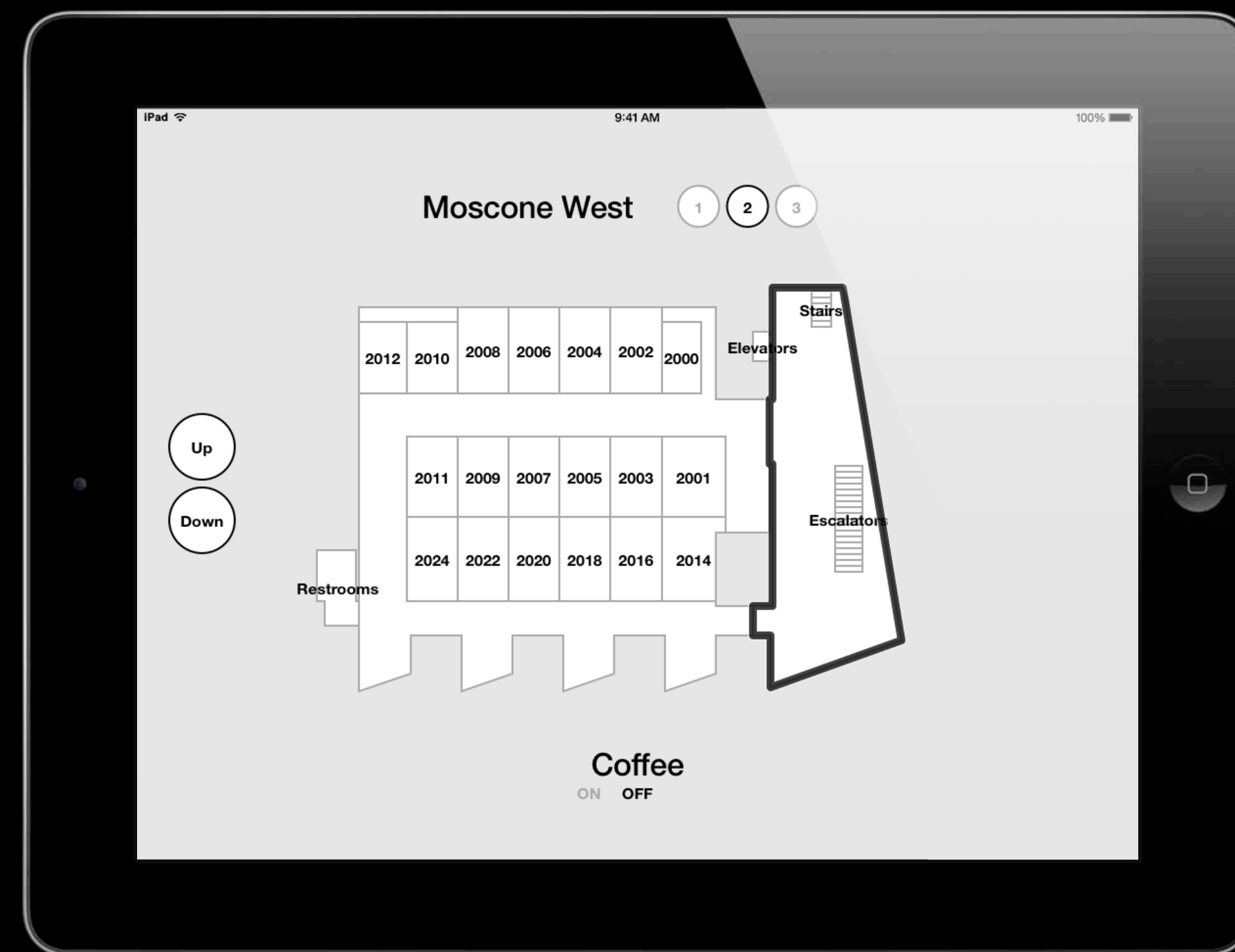


Advanced Accessibility



@property (copy) UIBezierPath *accessibilityPath

- Preferred over accessibilityFrame when present



Demo

Advanced and new UIAccessibility API

Guided Access

What we learned

- Hardware buttons are useful

Guided Access

What we learned

- Hardware buttons are useful
- Circling areas to ignore touch doesn't address visual distractions

Guided Access

What we learned

- Hardware buttons are useful
- Circling areas to ignore touch doesn't address visual distractions
- Developers are eager to adopt

Demo

Guided Access API

Guided Access API

UIGuidedAccessRestrictions.h

- Implement on UIApplicationDelegate
- State persistence comes free



Guided Access API



```
static NSString *ControlsRestrictionId =
    @"com.MyCompany.MyApp.ControlsRestrictionId"

- (NSArray *)guidedAccessRestrictionIdentifiers {
    return @[ ControlsRestrictionId ];
}

- (NSString *)textForGuidedAccessRestrictionWithIdentifier:
    (NSString *)restrictionId {
    return @"Controls";
}

- (NSString *)detailTextForGuidedAccessRestrictionWithIdentifier:
    (NSString *)restrictionId {
    return @"Adjust floor down/up";
}
```

Guided Access API



```
- (void)guidedAccessRestrictionWithIdentifier:  
  (NSString *)restrictionId didChangeState:  
  (UIGuidedAccessRestrictionState)newState {  
    BOOL enabled = ( newState != UIGuidedAccessRestrictionStateDeny );  
    [self setControlsEnabled:enabled];  
}
```

```
UIGuidedAccessRestrictionState  
  UIGuidedAccessRestrictionStateForIdentifier(NSString *restrictionId);
```

We heard you...

Text to Speech API

AVSpeechSynthesis.h



```
AVSpeechSynthesizer *synthesizer = [[AVSpeechSynthesizer alloc] init];
```

Text to Speech API

AVSpeechSynthesis.h



```
AVSpeechSynthesizer *synthesizer = [[AVSpeechSynthesizer alloc] init];  
AVSpeechUtterance *utterance = [AVSpeechUtteranceWithString:@"Hello!"];
```

Text to Speech API

AVSpeechSynthesis.h



```
AVSpeechSynthesizer *synthesizer = [[AVSpeechSynthesizer alloc] init];  
AVSpeechUtterance *utterance = [AVSpeechUtteranceWithString:@"Hello!"];  
[speechSynth speakUtterance:utterance];
```

Text to Speech API

AVSpeechSynthesis.h



```
AVSpeechSynthesizer *synthesizer = [[AVSpeechSynthesizer alloc] init];  
AVSpeechUtterance *utterance = [AVSpeechUtteranceWithString:@"Hello!"];  
[speechSynth speakUtterance:utterance];
```

Works in Safari!

More Information

Jake Behrens

UI Frameworks Evangelist
behrens@apple.com

Documentation

Accessibility Programming Guide for iOS
Search <http://developer.apple.com/> for Accessibility

UIAccessibility Protocol Reference
Search <http://developer.apple.com/> for UIAccessibility

VoiceOver User Manual
<http://support.apple.com/manuals/iphone>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

Accessibility in OS X

Pacific Heights
Tuesday 10:15AM

Designing Accessories for iOS and OS X

Nob Hill
Tuesday 10:15AM

Labs

Accessibility and Speech

Services Lab A
Tuesday 12:45PM



Summary



- Adding accessibility is easy

Summary



- Adding accessibility is easy
- More users

Summary



- Adding accessibility is easy
- More users
- Great feedback

Summary



- Adding accessibility is easy
- More users
- Great feedback
- We take accessibility seriously, so should you

 WWDC2013