App Frameworks

#WWDC17

Focus Interaction in tvOS 11

Session 224

Matt Ricketson, UIKit Engineer
Jon Staff, UIKit Engineer

New features and enhancements

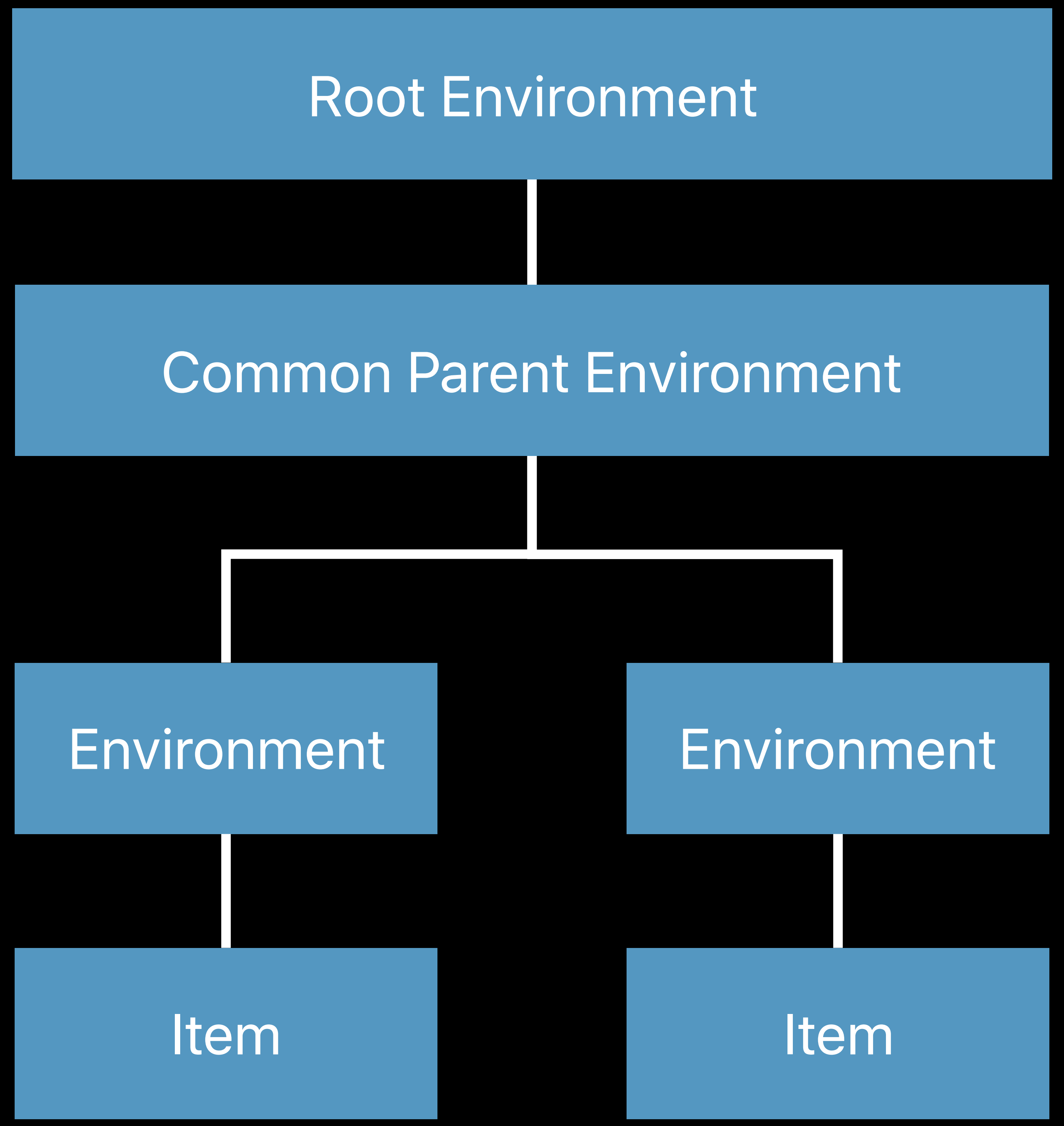
Support for SceneKit

New debugging tools

Review

Review

Focus environment hierarchy



Root Environment

Common Parent Environment

Environment

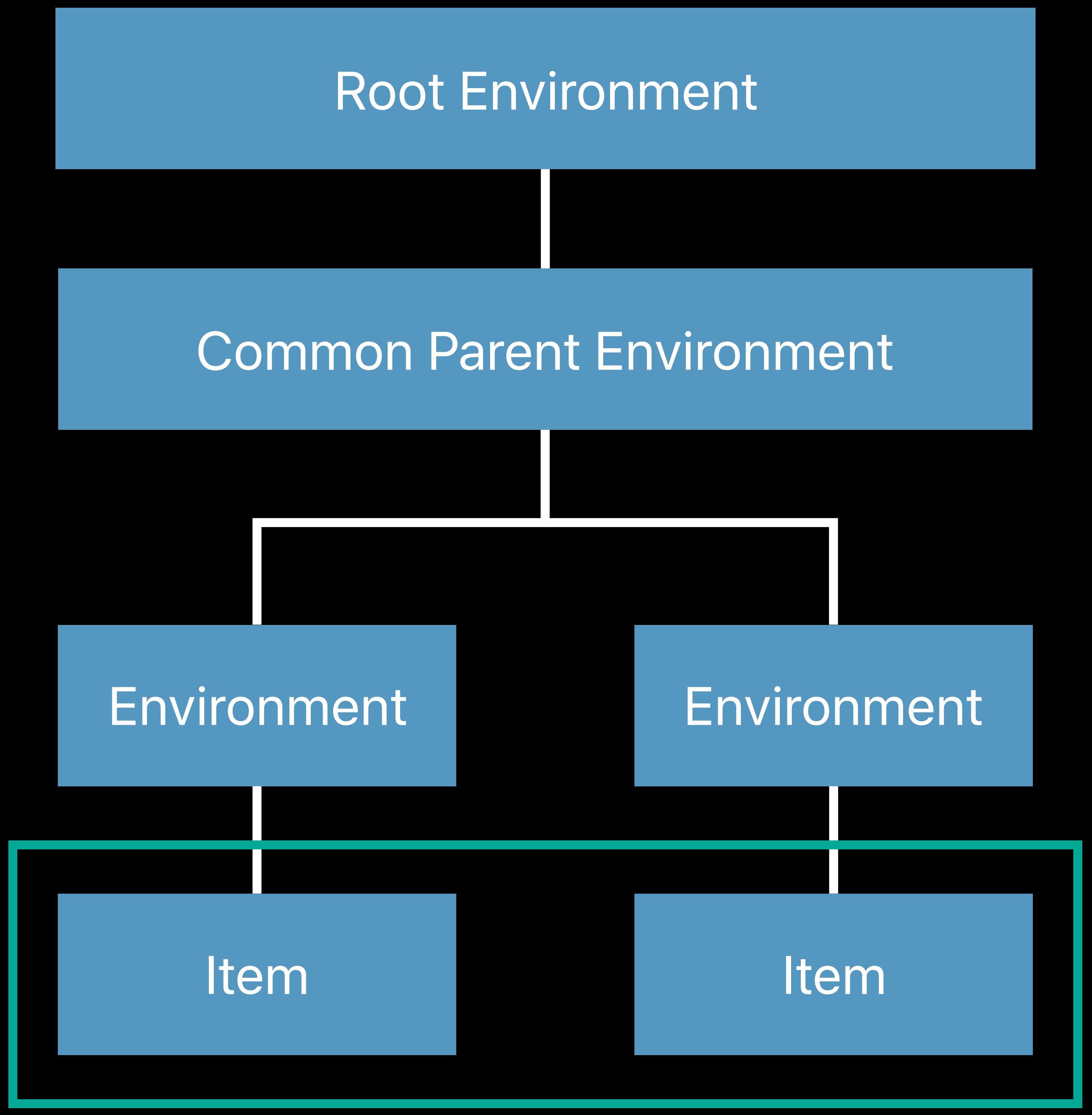
Environment

Item

Item

Review

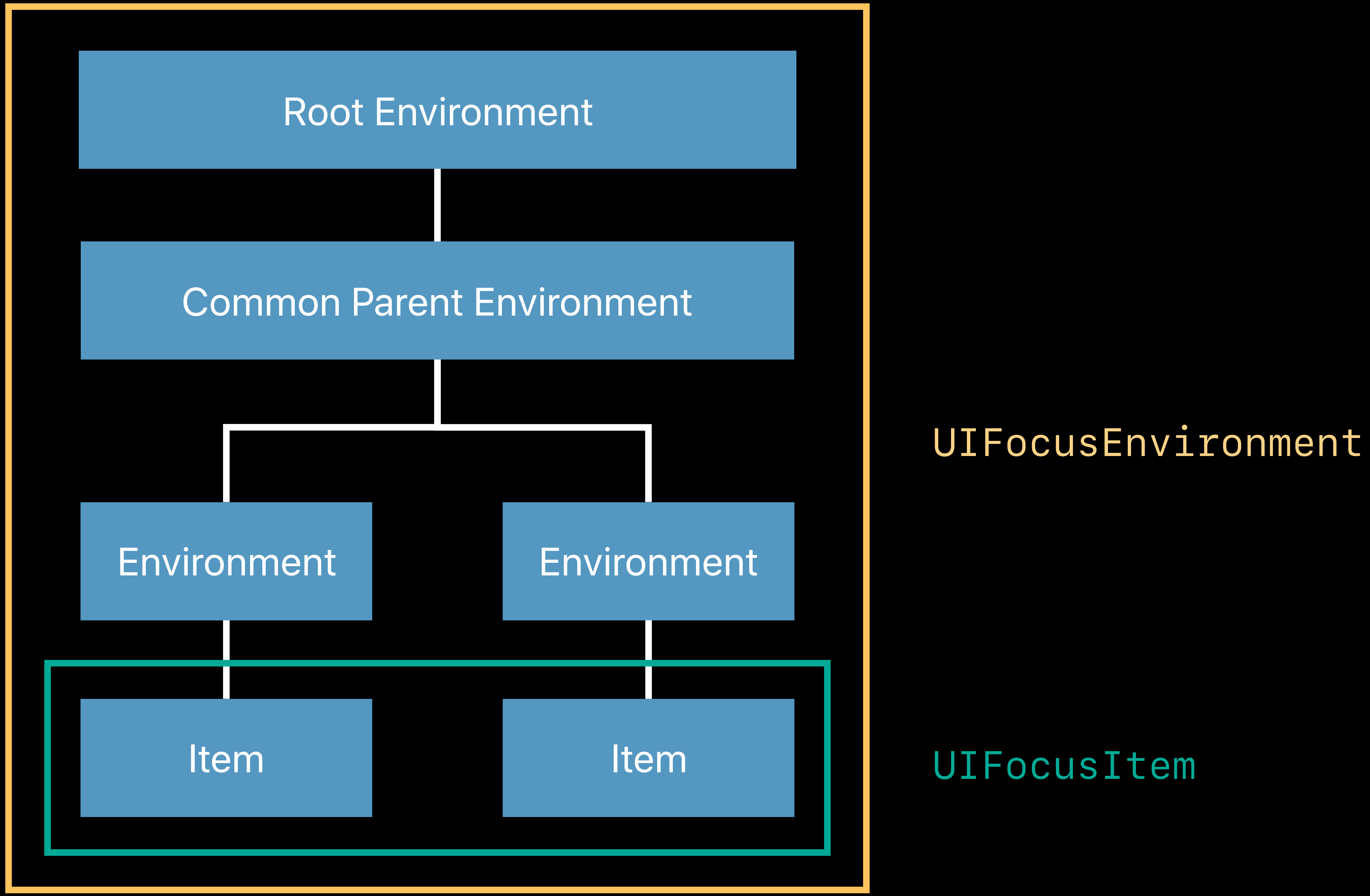
Focus environment hierarchy



`UIFocusItem`

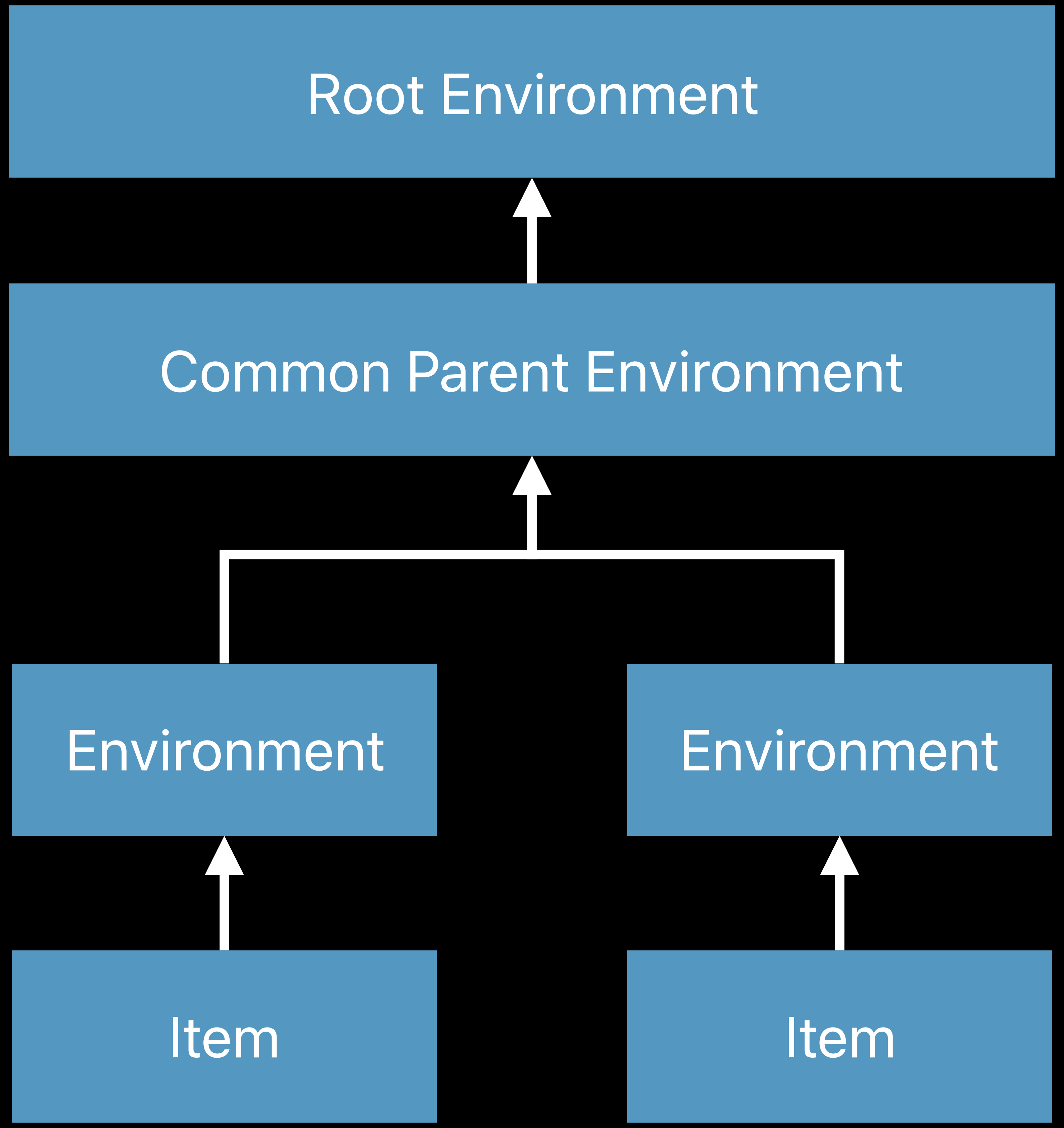
Review

Focus environment hierarchy



Review

Focus environment hierarchy



Review

Types of focus updates

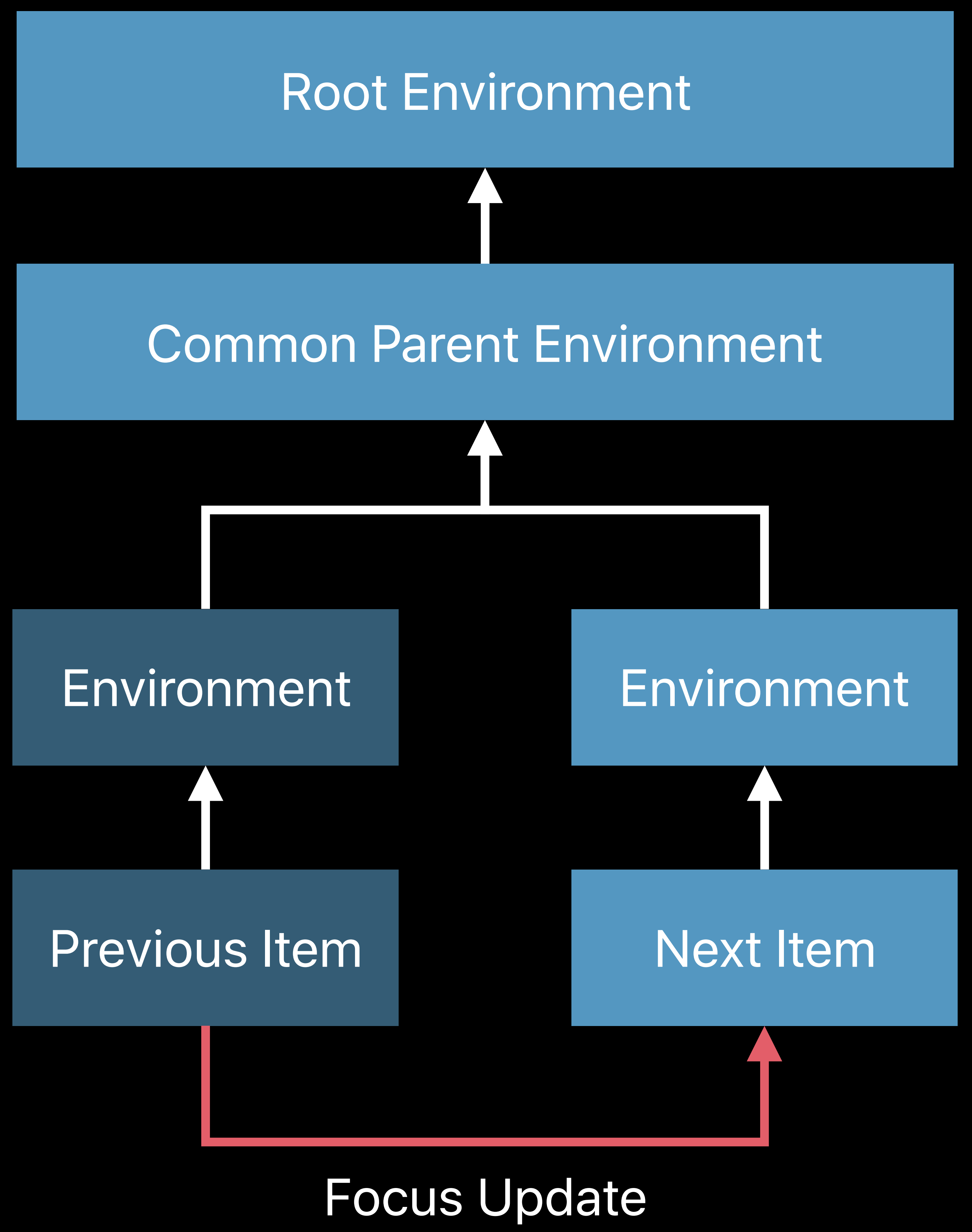
User-initiated focus movement

Programmatic focus updates

- Requested via `setNeedsFocusUpdate`

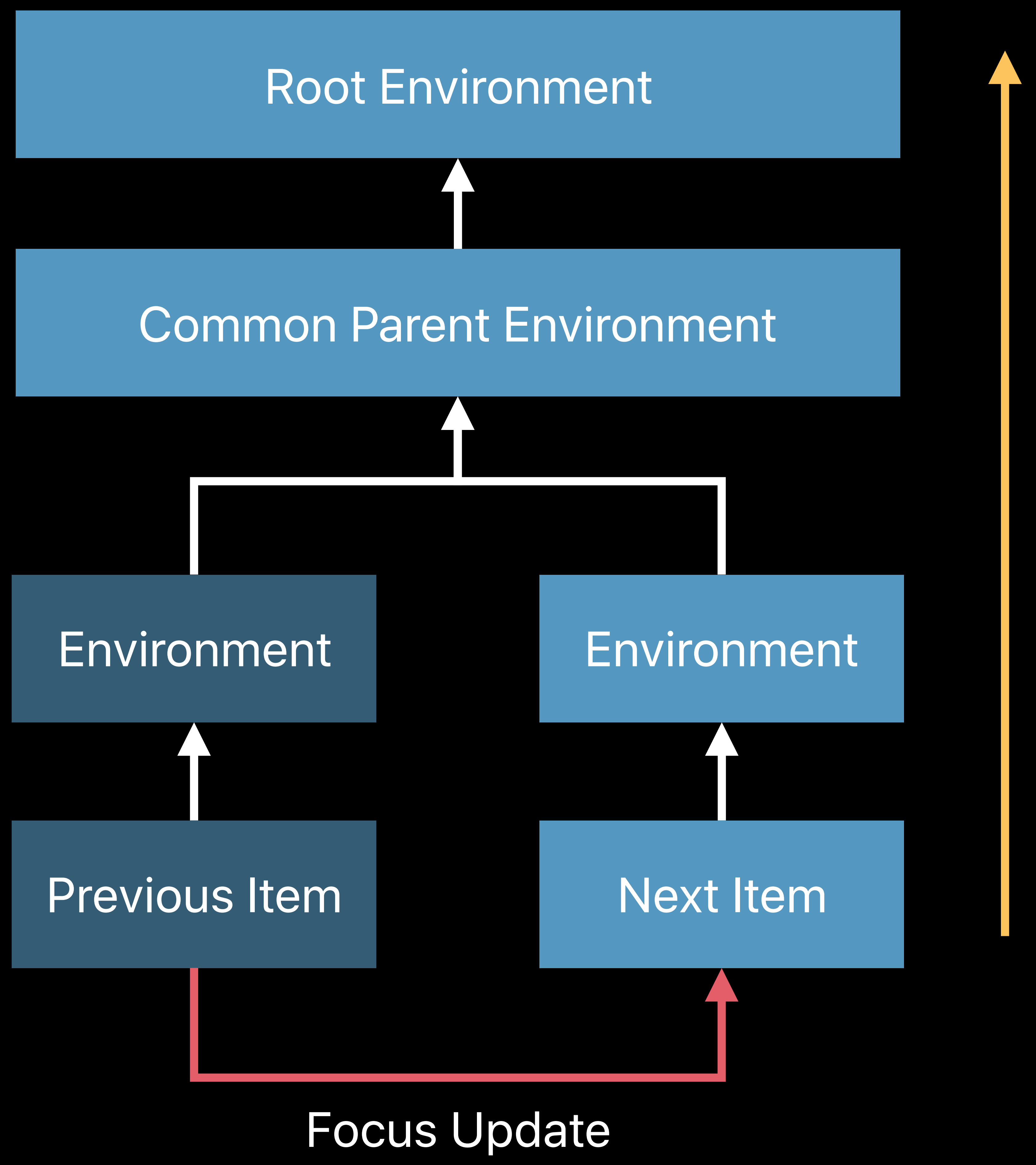
Review

Focus updates



Review

Focus updates

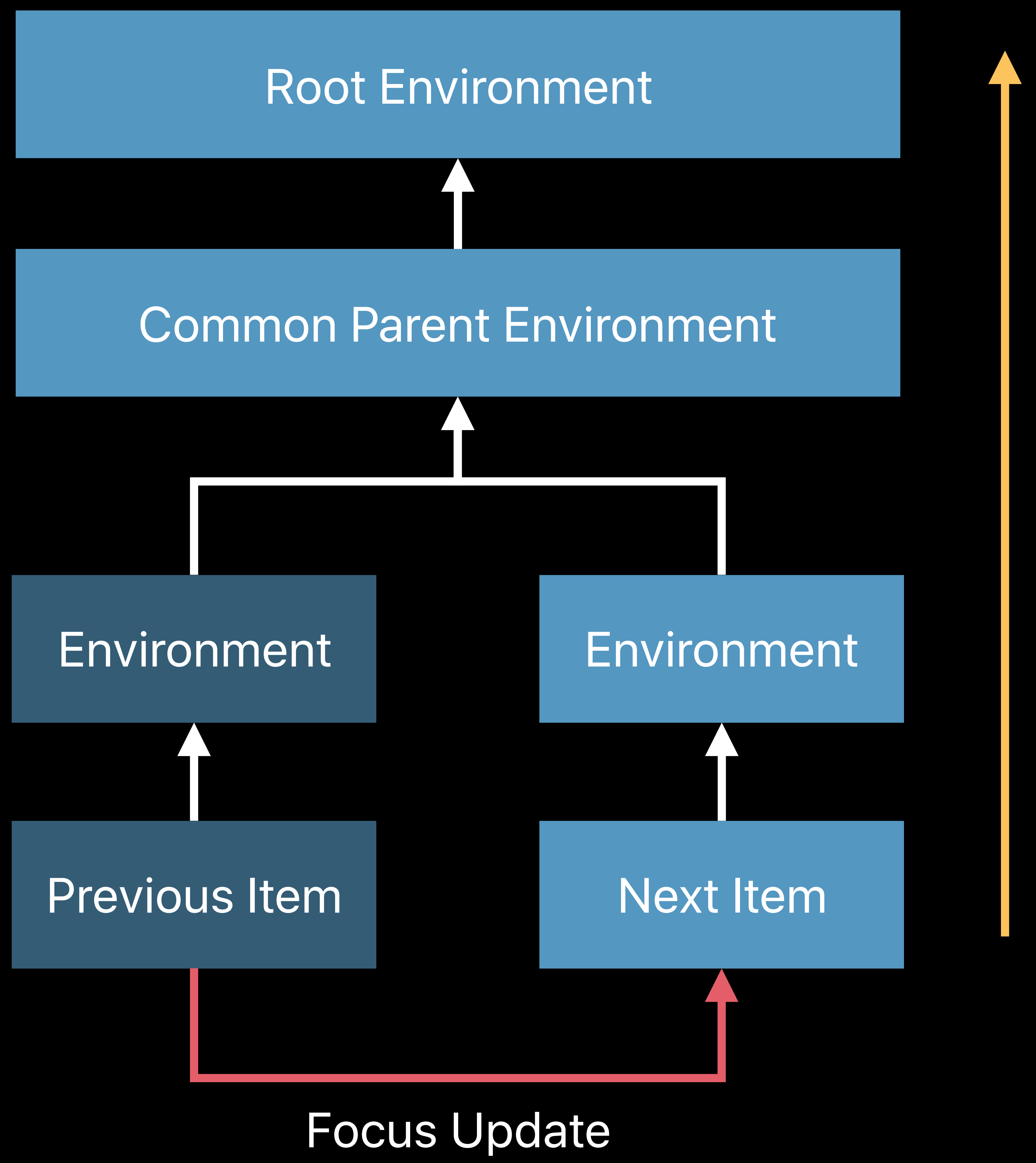


UIFocusEnvironment

```
func didUpdateFocus(in:with:)
```

Focus Update Notifications

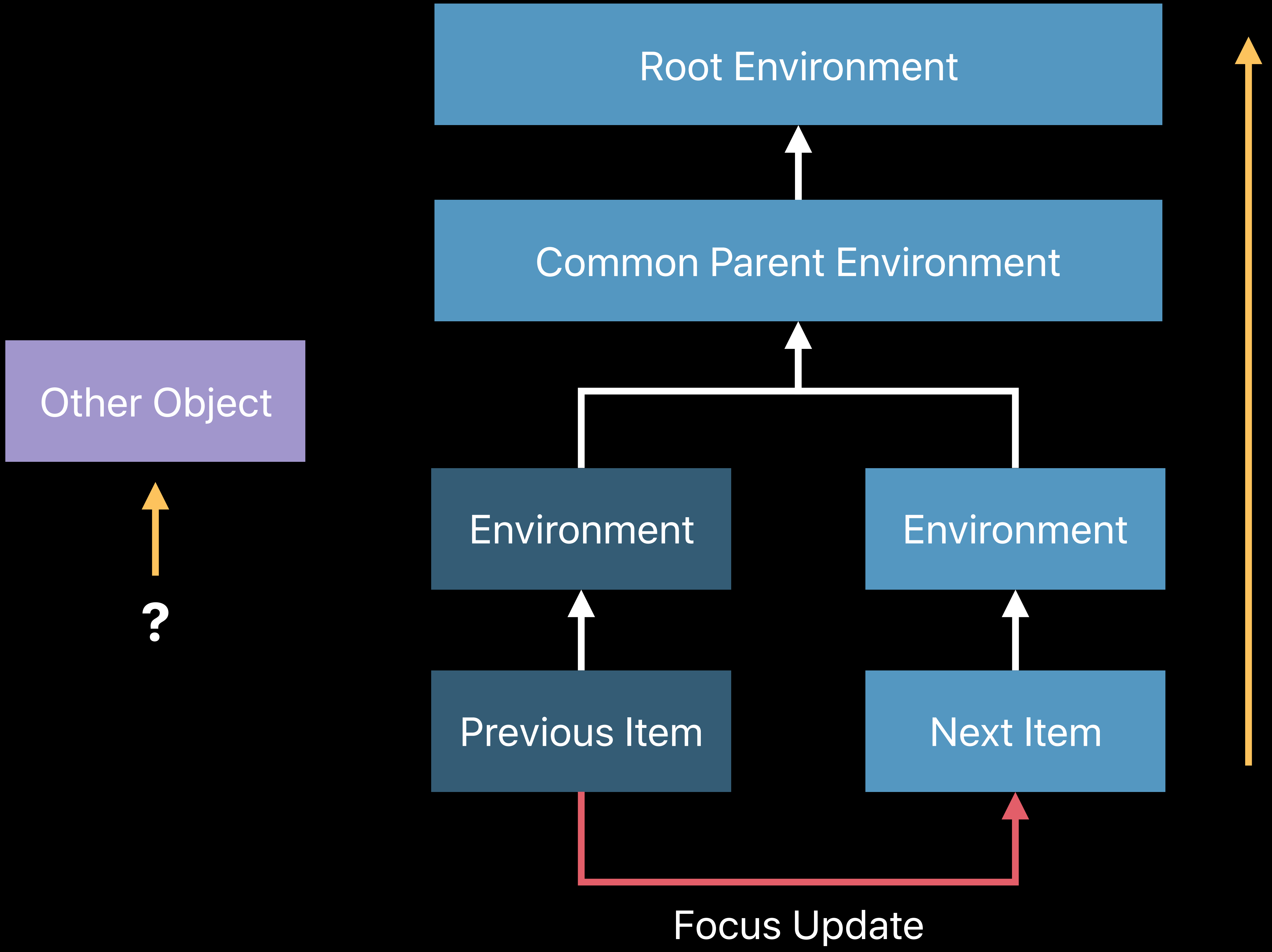
Focus Updates



UIFocusEnvironment

```
func didUpdateFocus(in:with:)
```

Focus Updates



UIFocusEnvironment

```
func didUpdateFocus(in:with:)
```

Focus Update Notifications

NEW

```
static let UIFocusDidUpdate: Notification.Name
```

Sent on every focus update

Sent after calling `didUpdateFocus(in:with:)` on all relevant focus environments

User info dictionary includes both the context and animation coordinator

Focus Update Notifications

User info keys

NEW

`UIFocusUpdateContextKey`	Provides the `UIFocusUpdateContext` instance that describes the update.
`UIFocusUpdateAnimationCoordinatorKey`	Provides the `UIFocusAnimationCoordinator` instance associated with the update.

Focus Update Notifications

NEW

```
static let UIFocusMovementDidFail: Notification.Name
```

Sent when the user tries, but fails, to move focus in a certain direction

Not sent for programmatic focus updates that fail

Also provides a `UIFocusUpdateContext` instance

Use for providing helpful feedback to the user

Protocol Extensions

UIFocusEnvironment and UIFocusItem

Protocol Extensions

UIFocusItem

NEW

Check if an item is currently focused:

```
let isFocused = item.isFocused
```

Protocol Extensions

Focus environment containment

NEW

Check if a focus environment contains another environment:

```
if environment1.contains(environment2) {
    // do something
}
```

```
// MyViewController.h

override func didUpdateFocus(in context: UIFocusUpdateContext,
                             with coordinator: UIFocusAnimationCoordinator) {
    super.didUpdateFocus(in: context, with: coordinator)

    if let nextFocusedItem = context.nextFocusedItem, self.contains(nextFocusedItem) {
        // focus is inside the view controller
    }

}
```

Focus Animation Enhancements

Jon Staff, UIKit Frameworks Engineer

Focus Animations

Review

Use the coordination API for matching system animations

Provided whenever a focus update occurs

```
open class UIFocusAnimationCoordinator : NSObject {

    open func addCoordinatedAnimations(_ animations: (() -> Void)?,
                                       completion: (() -> Void)? = nil)


    ...

}
```

Focus Animations

Review

UIKit manages focus animations

Animation duration is influenced by the speed of focus movement

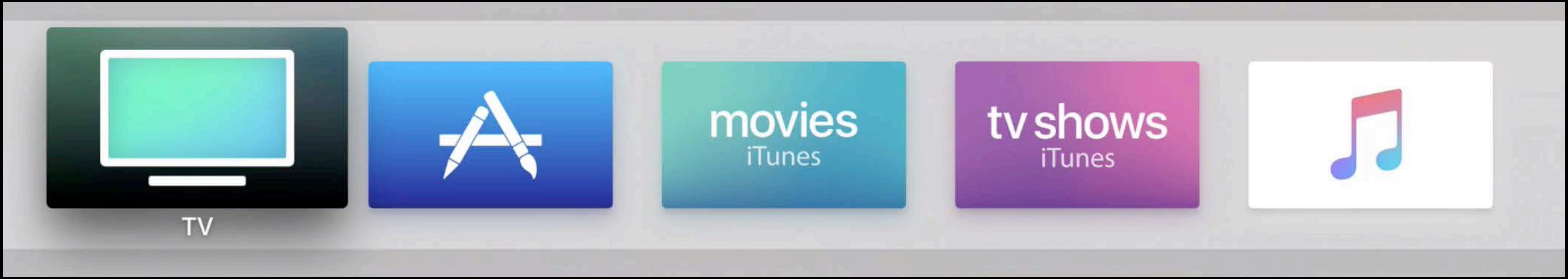
- Quicker movements have quicker animations
- Slower movements have slower animations
- Animations for off-screen items are delayed

Focus Animations

Review

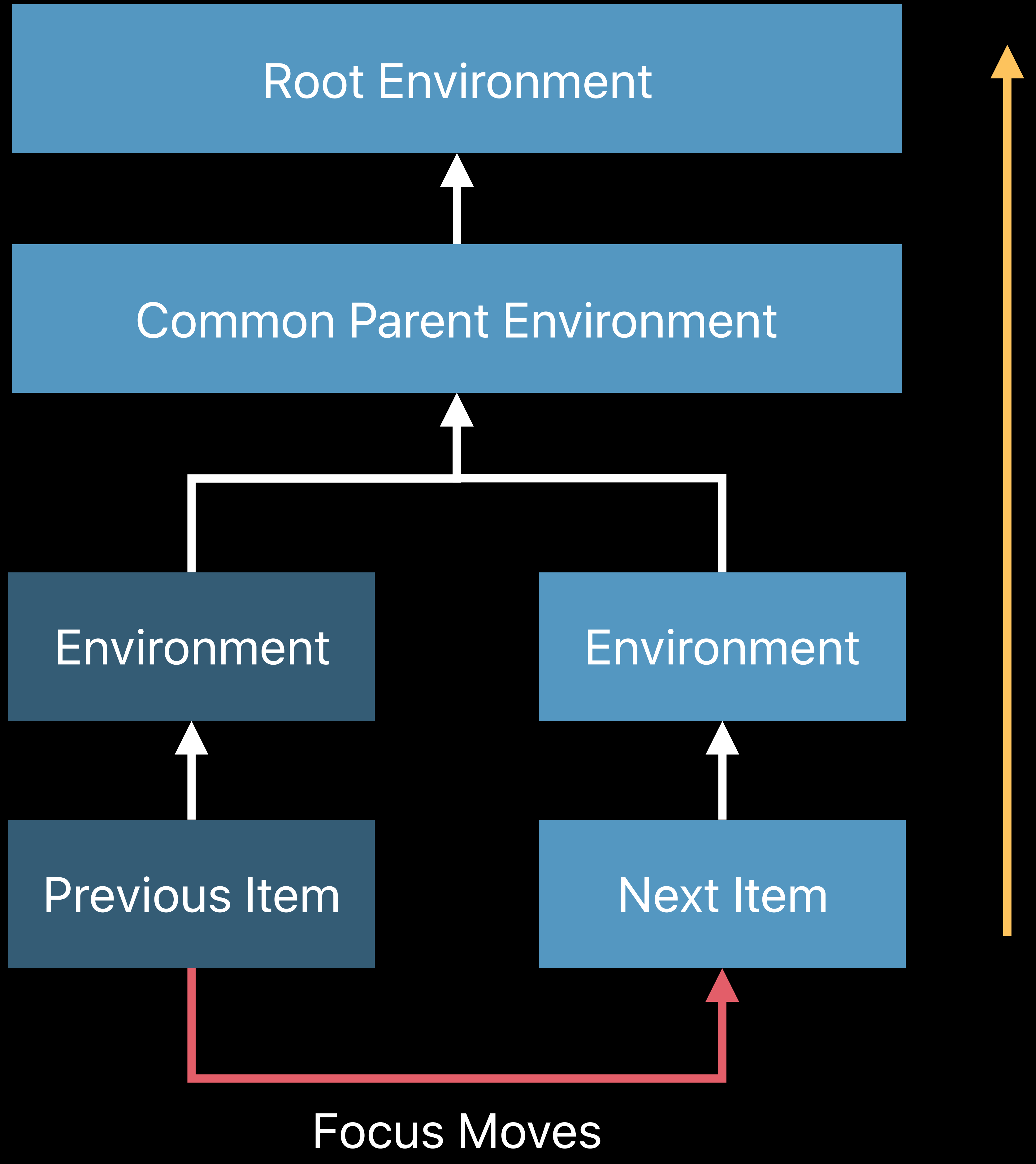
Focusing animations are quicker and more prominent

Un-focusing animations are slower and subtler



Focus Animations

Review



UIFocusEnvironment

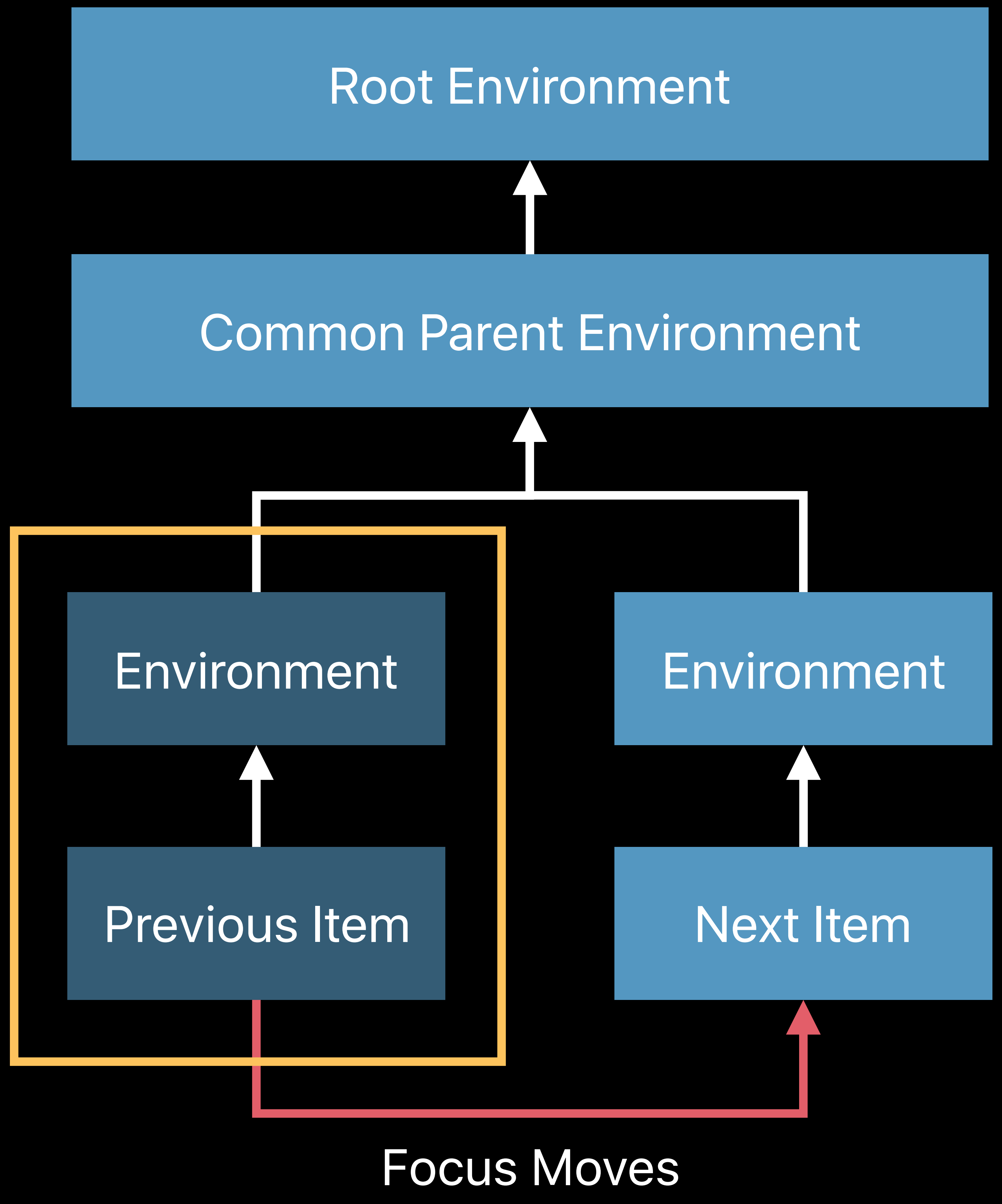
```
func didUpdateFocus(in:with:)
```

Focus Animations

Review

```
func addCoordinatedAnimations(_:completion:)
```

Un-focusing Animations

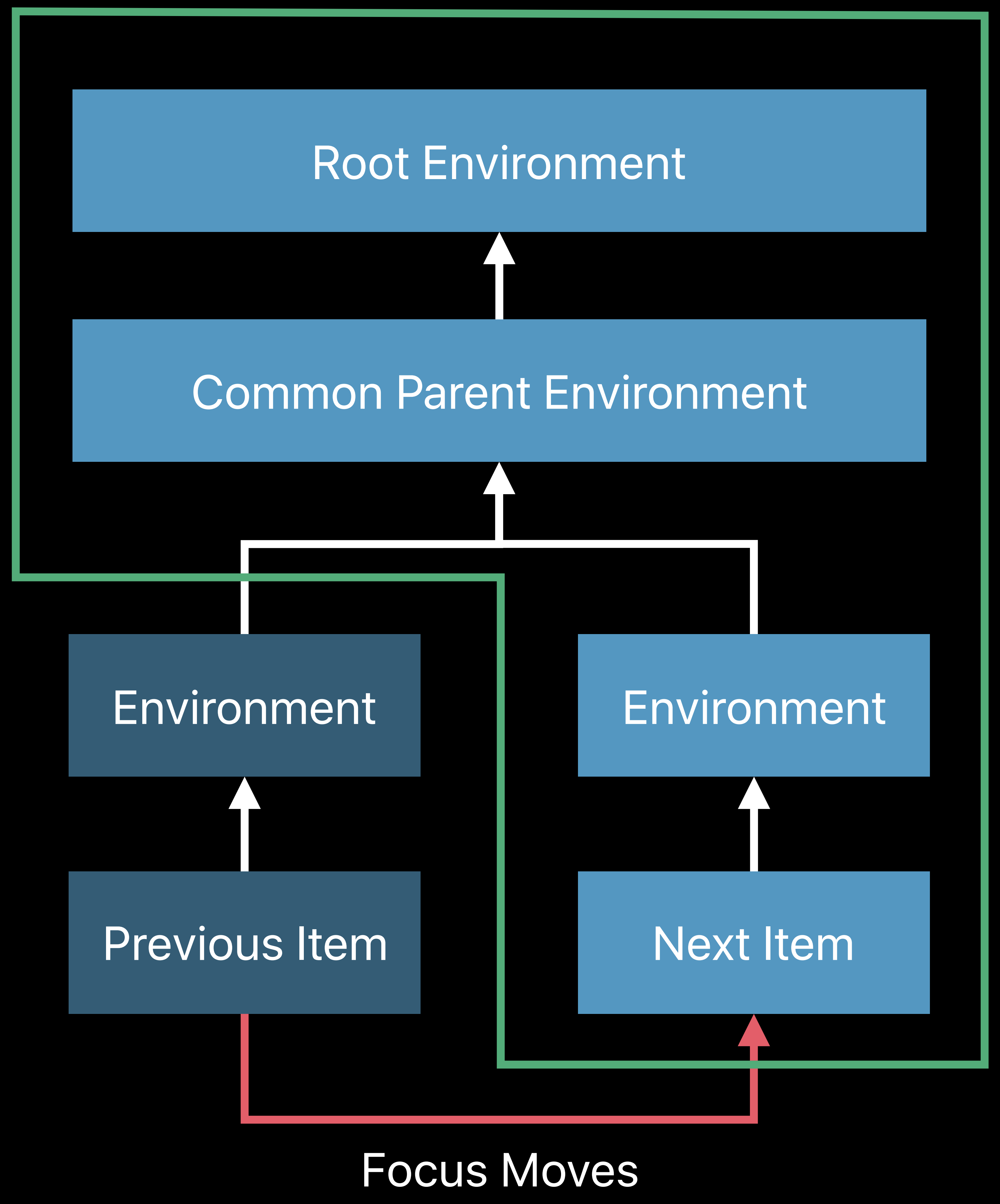


Focus Animations

Review

```
func addCoordinatedAnimations(_:completion:)
```

Focusing Animations



Targeted Coordinated Animations

Review

NEW

Targeted Coordinated Animations

Review

NEW

```
open class UIFocusAnimationCoordinator : NSObject {

    open func addCoordinatedAnimations(_ animations: (() -> Void)?,
                                       completion: (() -> Void)? = nil)

    open func addCoordinatedFocusingAnimations(_ animations: ((UIFocusAnimationContext) -> Void)?,
                                               completion: (() -> Void)? = nil)

    open func addCoordinatedUnfocusingAnimations(_ animations: ((UIFocusAnimationContext) -> Void)?,
                                                 completion: (() -> Void)? = nil)

}
```

Focus Animations

New API

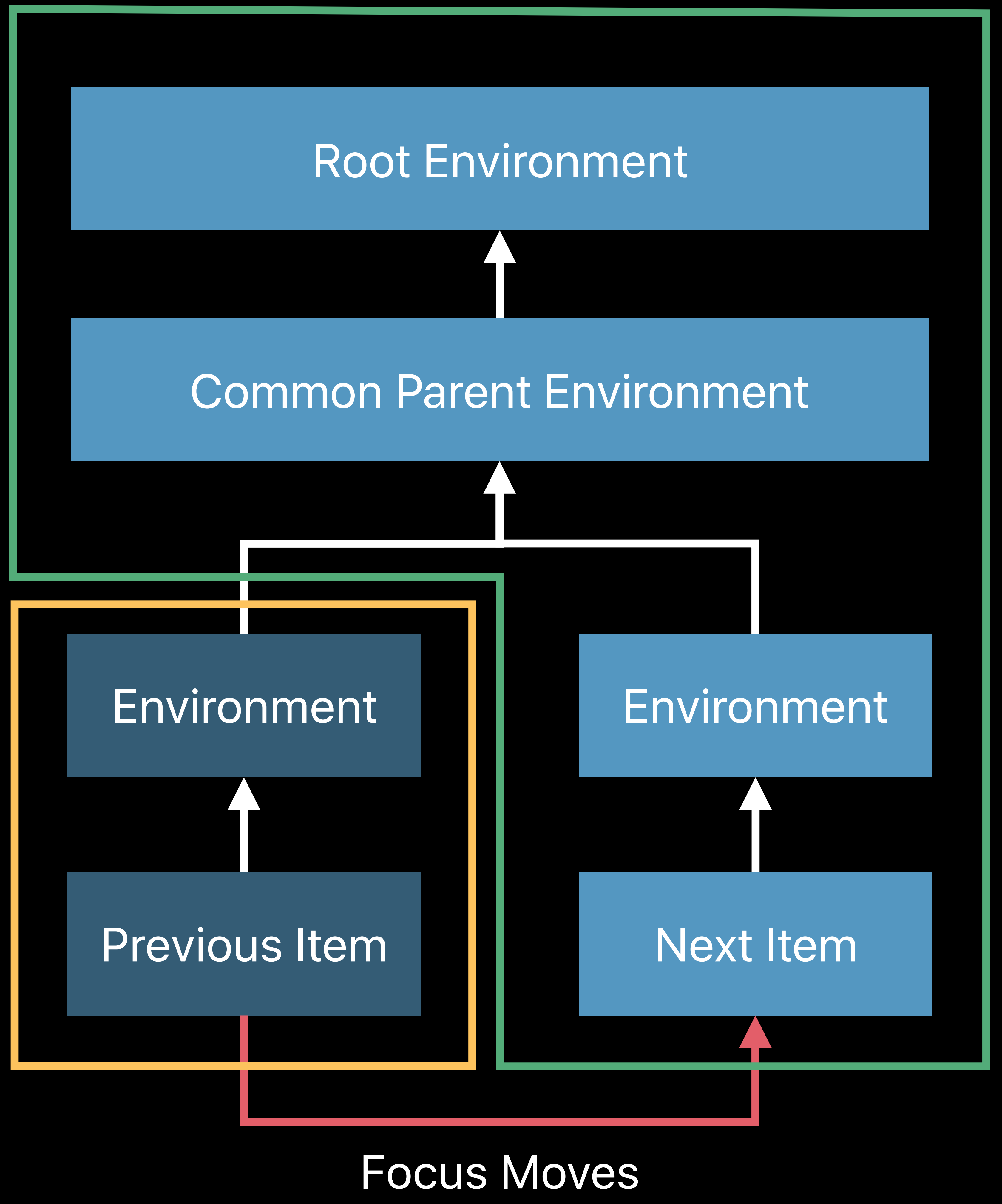
NEW

```
func addCoordinatedFocusingAnimations(_:completion:)
```

Focusing Animations

```
func addCoordinatedUnfocusingAnimations(_:completion:)
```

Un-focusing Animations



Demo

Targeted coordinated animations

Targeted Coordinated Animations

New API

NEW

Timing information provided by `UIFocusAnimationContext`

```
public protocol UIFocusAnimationContext : NSObjectProtocol {

    /// The duration of the main animations in seconds.
    public var duration: TimeInterval { get }

}
```

```swift
override func didUpdateFocus(in context: UIFocusUpdateContext,
                             with coordinator: UIFocusAnimationCoordinator) {

       coordinator.addCoordinatedFocusingAnimations({ animationContext in
            let duration = animationContext.duration / 2.0

            UIView.animate(withDuration: duration,
                           delay: duration, // start at the halfway point
                           options: [.overrideInheritedDuration],
                           animations: {
                             // some animation
                           })
        })
}
```

```
override func didUpdateFocus(in context: UIFocusUpdateContext,
                             with coordinator: UIFocusAnimationCoordinator) {

       coordinator.addCoordinatedFocusingAnimations({ animationContext in
            let duration = animationContext.duration / 2.0

            UIView.animate(withDuration: duration,
                           delay: duration, // start at the halfway point
                           options: [.overrideInheritedDuration],
                           animations: {
                             // some animation
                           })
        })
}
```

```
override func didUpdateFocus(in context: UIFocusUpdateContext,
                             with coordinator: UIFocusAnimationCoordinator) {

       coordinator.addCoordinatedFocusingAnimations({ animationContext in
            let duration = animationContext.duration / 2.0

            UIView.animate(withDuration: duration,
                           delay: duration, // start at the halfway point
                           options: [.overrideInheritedDuration],
                           animations: {
                             // some animation
                           })
        })
}
```

```
override func didUpdateFocus(in context: UIFocusUpdateContext,
                             with coordinator: UIFocusAnimationCoordinator) {

       coordinator.addCoordinatedFocusingAnimations({ animationContext in
            let duration = animationContext.duration / 2.0

            UIView.animate(withDuration: duration,
                           delay: duration, // start at the halfway point
                           options: [.overrideInheritedDuration],
                           animations: {
                             // some animation
                           })
        })
}
```

UIFocusAnimationContext

Better non-UIKit support

Consistent feel for focus interaction across frameworks

```swift
override func didUpdateFocus(in context: UIFocusUpdateContext,
                             with coordinator: UIFocusAnimationCoordinator) {
    let node = myNode
    coordinator.addCoordinatedFocusingAnimations({ animationContext in
        let fadeAction = SKAction.fadeAlpha(to: 0.5, duration: animationContext.duration)
        node.run(fadeAction)
    })
}
```

Summary

Use the existing API when targeting isn’t needed

Use the new coordination APIs for targeting animations

Use the animation context for timing information

Custom Focus Sounds

Focus Sounds

Volume adjustments based on speed

- Faster movements have quieter sounds
- Slower movements have louder sounds

Panning adjustments based on location

- Items on the right side of the screen will pan right
- Items on the left side of the screen will pan left

Custom Focus Sounds

New API

NEW

Play custom sounds when focus movement occurs

Opt out of UIKit-played sounds

Custom Focus Sounds

New API

NEW

Register a sound with an identifier

Provide a sound identifier during a focus update

Sound matching the identifier is played

Custom Focus Sounds

New API

NEW

Class method on `UIFocusSystem`

```
open class UIFocusSystem : NSObject {

    /// Registers a sound file for a given identifier.
    open class func register(_ soundFileURL: URL,
                             forSoundIdentifier identifier: UIFocusSoundIdentifier)
}
```

Custom Focus Sounds

New API

NEW

Global registration

Register only once per sound identifier

Register early

Accepts standard iOS sound file formats

Custom Focus Sounds

New API

NEW

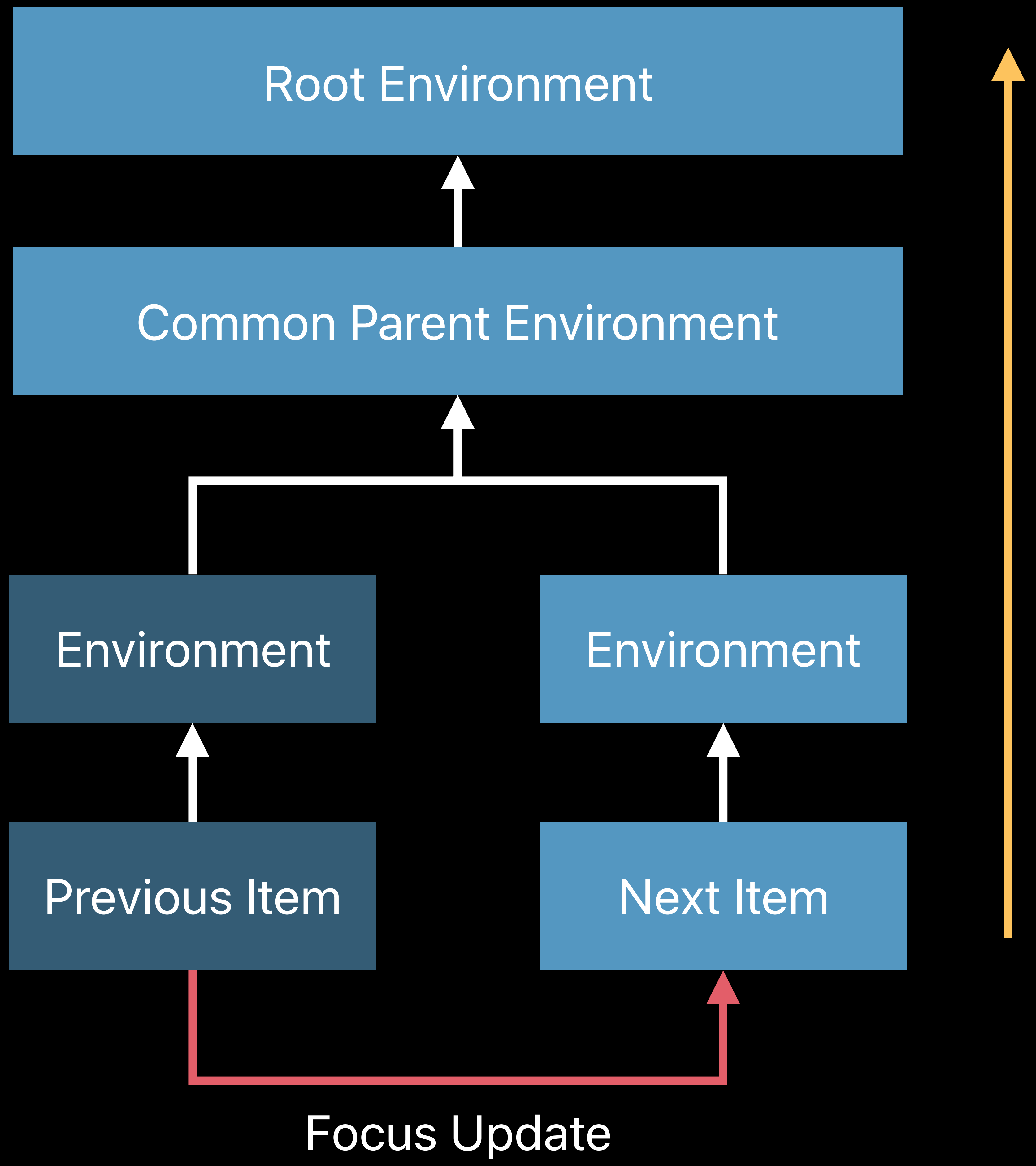
Optional method on `UIFocusEnvironment`

```
public protocol UIFocusEnvironment : NSObjectProtocol {

    ...

    optional public func soundIdentifierForFocusUpdate(in context: UIFocusUpdateContext) ->
UIFocusSoundIdentifier?

    ...

}
```

Custom Focus Sounds

New API

NEW



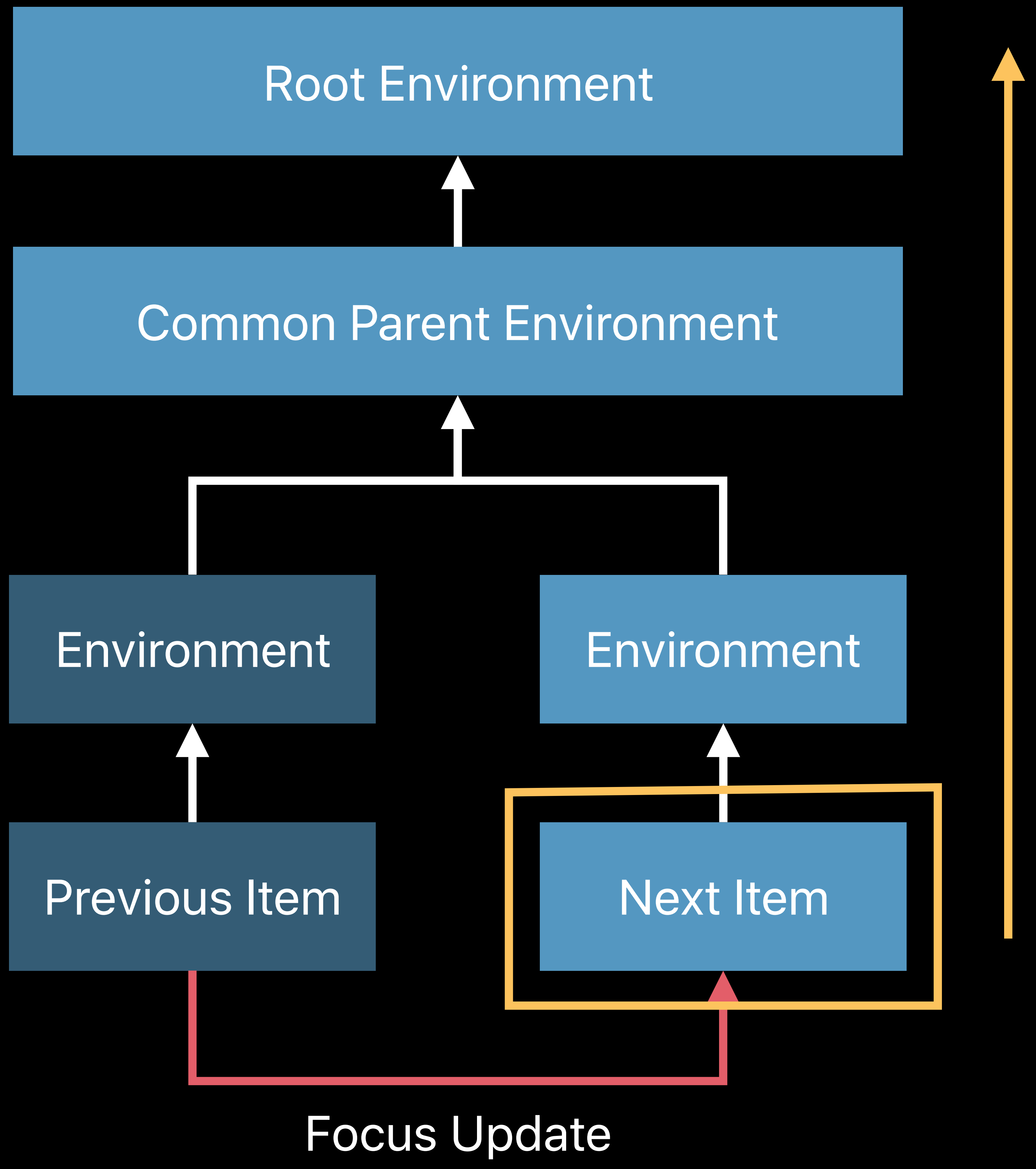
UIFocusEnvironment

```
func soundIdentifierForFocusUpdate(in:)
```

Custom Focus Sounds

New API

NEW

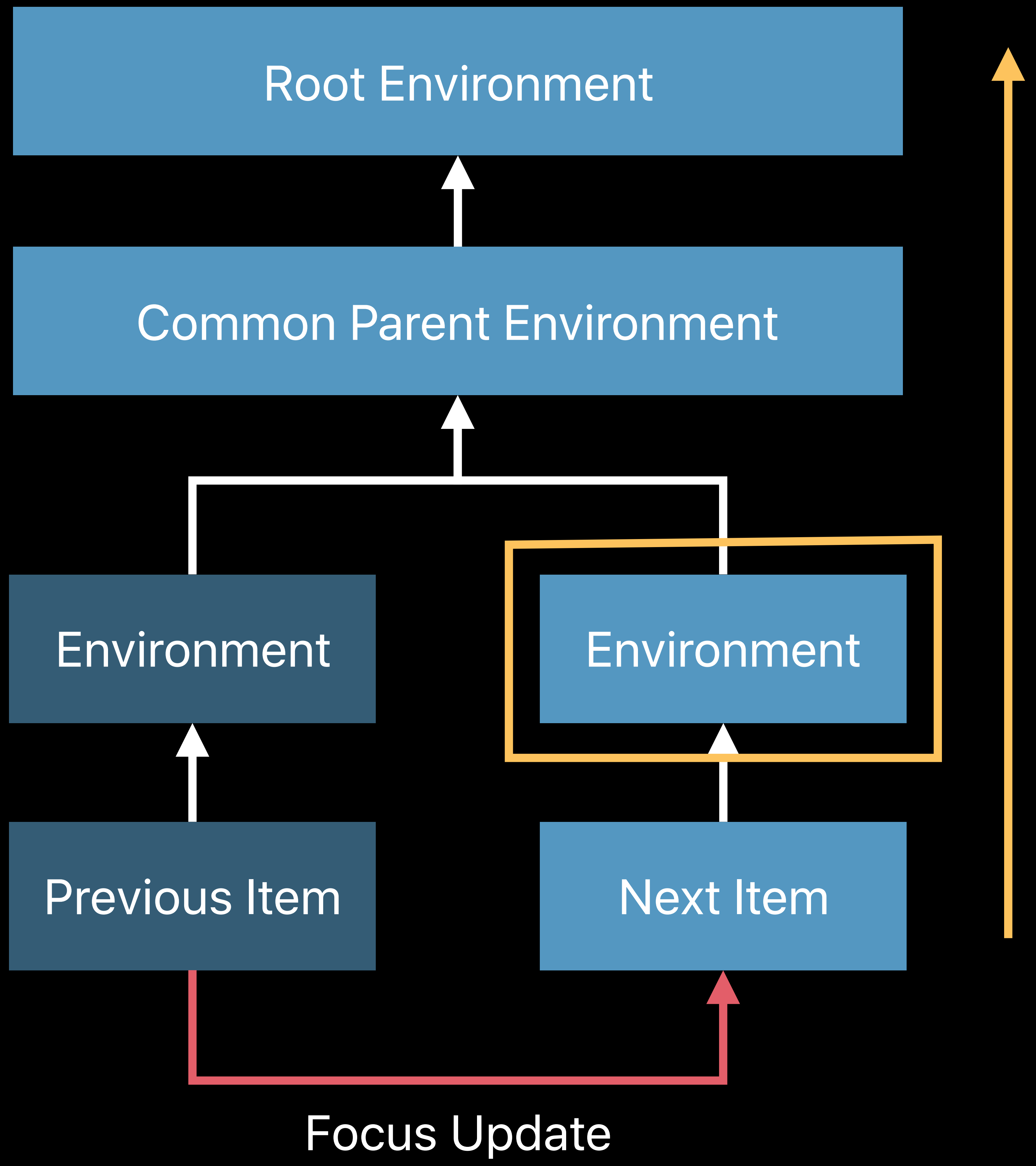


```
UIFocusEnvironment

func soundIdentifierForFocusUpdate(in:)
```

Custom Focus Sounds

New API

NEW



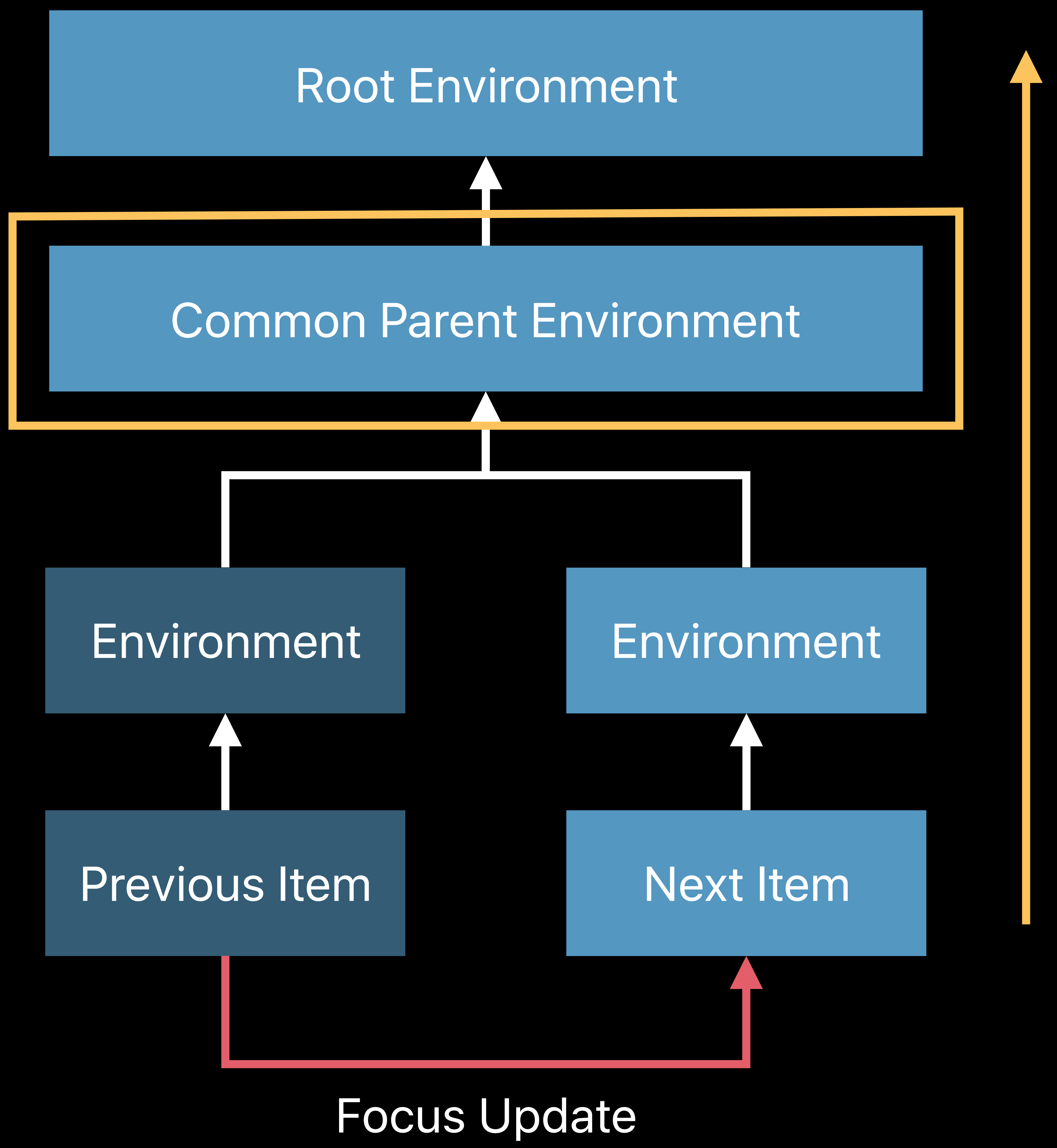
UIFocusEnvironment

```
func soundIdentifierForFocusUpdate(in:)
```

Custom Focus Sounds

New API

NEW



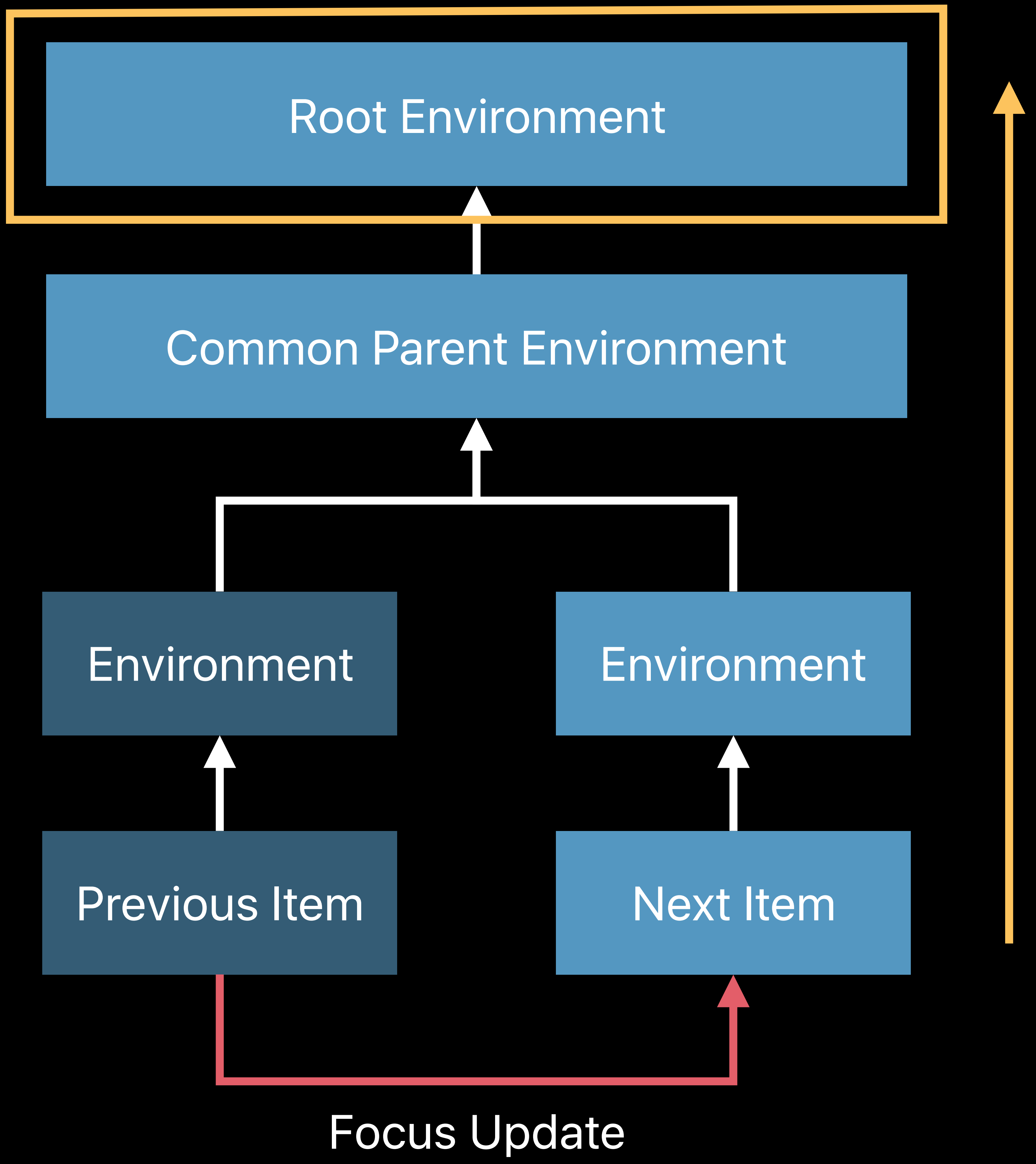
UIFocusEnvironment

```
func soundIdentifierForFocusUpdate(in:)
```

Custom Focus Sounds

New API

NEW



UIFocusEnvironment

```
func soundIdentifierForFocusUpdate(in:)
```

Custom Focus Sounds

New API

NEW

Sound Identifier	Description
Custom Sound Identifier (e.g. `.myCustomIdentifier`)	Plays the sound registered with this sound identifier.
`.default`	Explicitly force the default UIKit sound to play.
`.none`	Opt out of UIKit-played sounds for this focus update.

Custom Focus Sounds

New API

NEW

Conditionally playing a custom sound

```swift
override func soundIdentifierForFocusUpdate(in context: UIFocusUpdateContext)
-> UIFocusSoundIdentifier? {
    if let nextFocusedItem = context.nextFocusedItem,
        soundEnabledEnvironment.contains(nextFocusedItem) {
        return myCustomSoundIdentifier
    }
    return nil
}
```

Custom Focus Sounds

Opting out of sounds

Users expect focus movements to play sounds

Only opt out of focus sounds if your users expect it

- i.e. user-toggled sound settings in a game

Respect user device settings for navigation sounds

Custom Focus Sounds

Best practices

Custom sounds should enhance the user experience

Differentiate between different elements

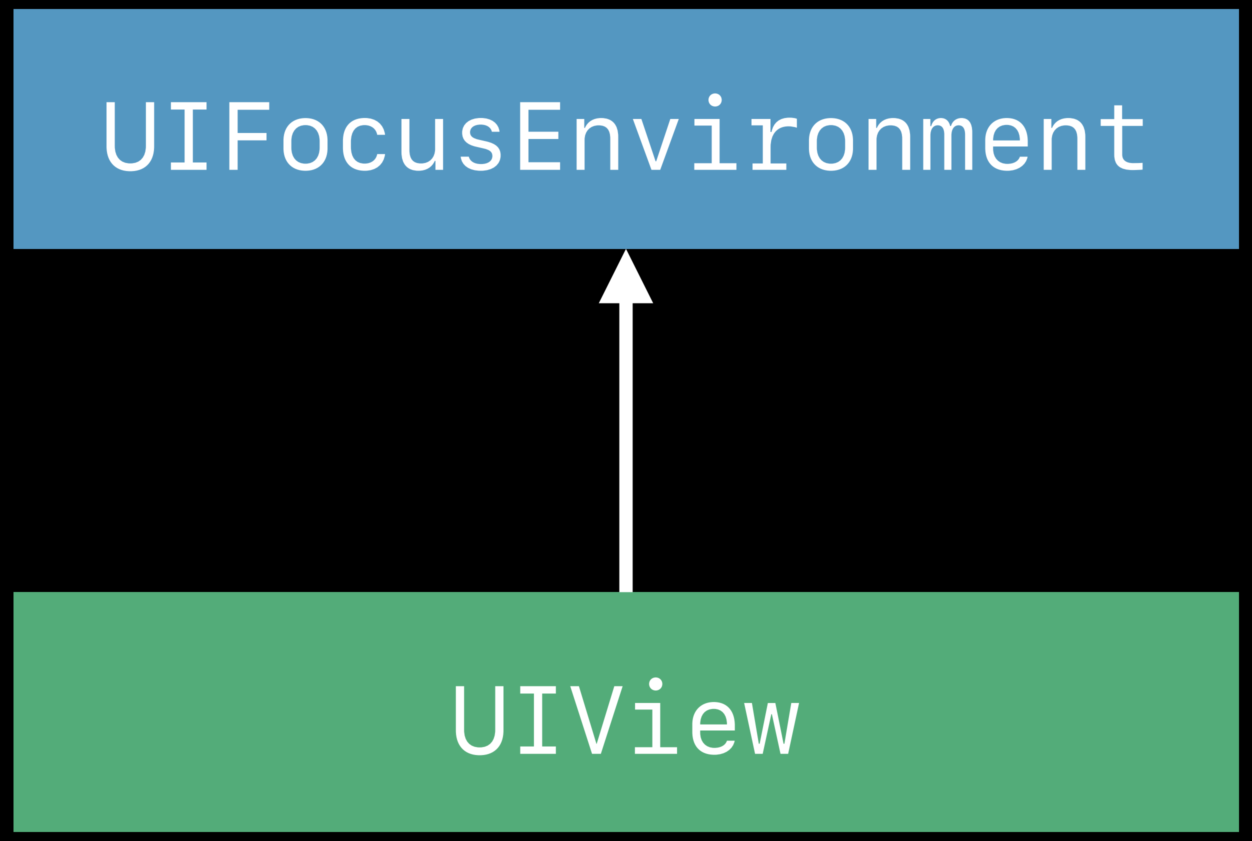
- i.e. Size or element type

Support for SceneKit

Matt Ricketson, UIKit Frameworks Engineer

Focus Item Types

tvOS 9

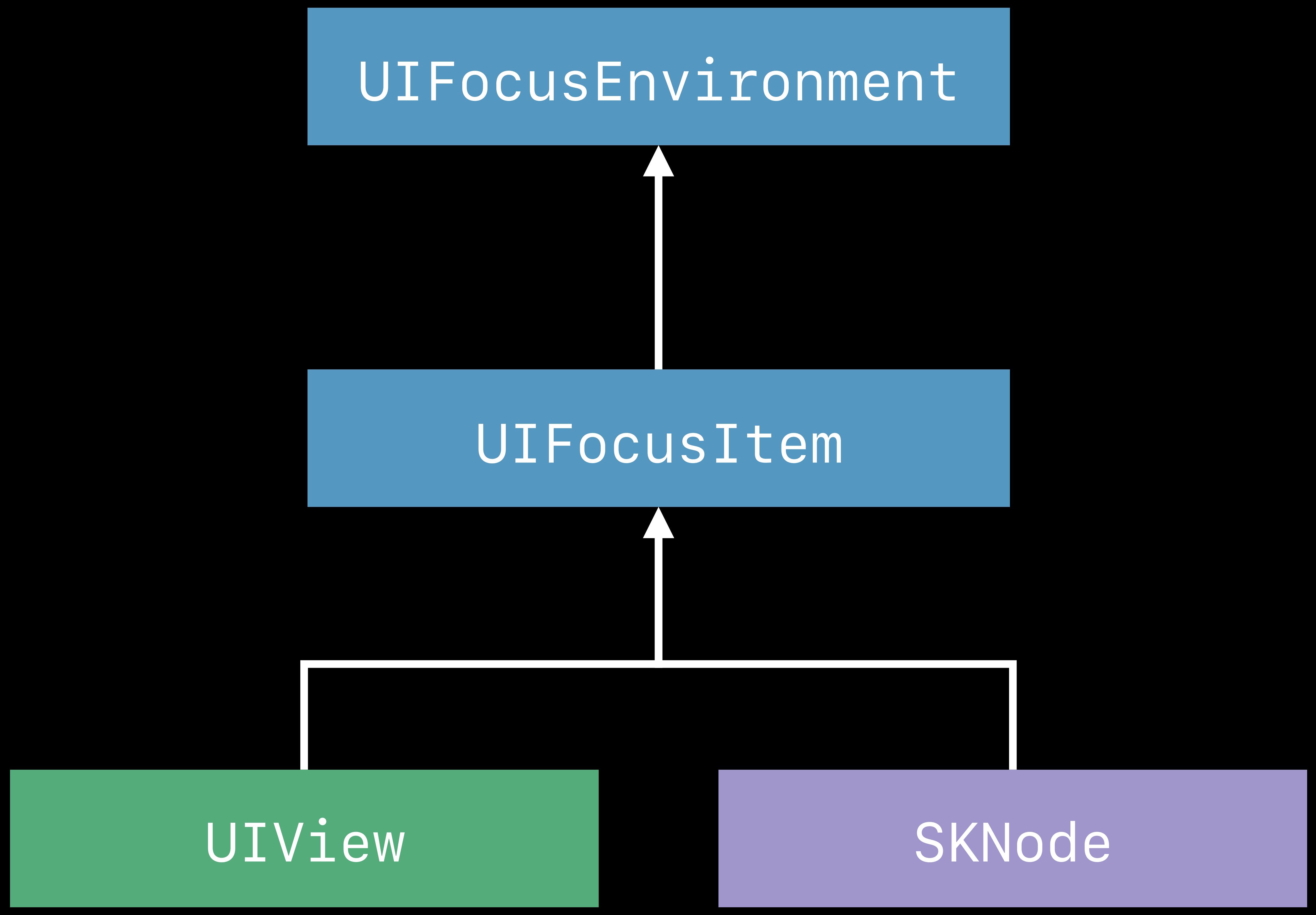


UIFocusEnvironment

UIView

Focus Item Types

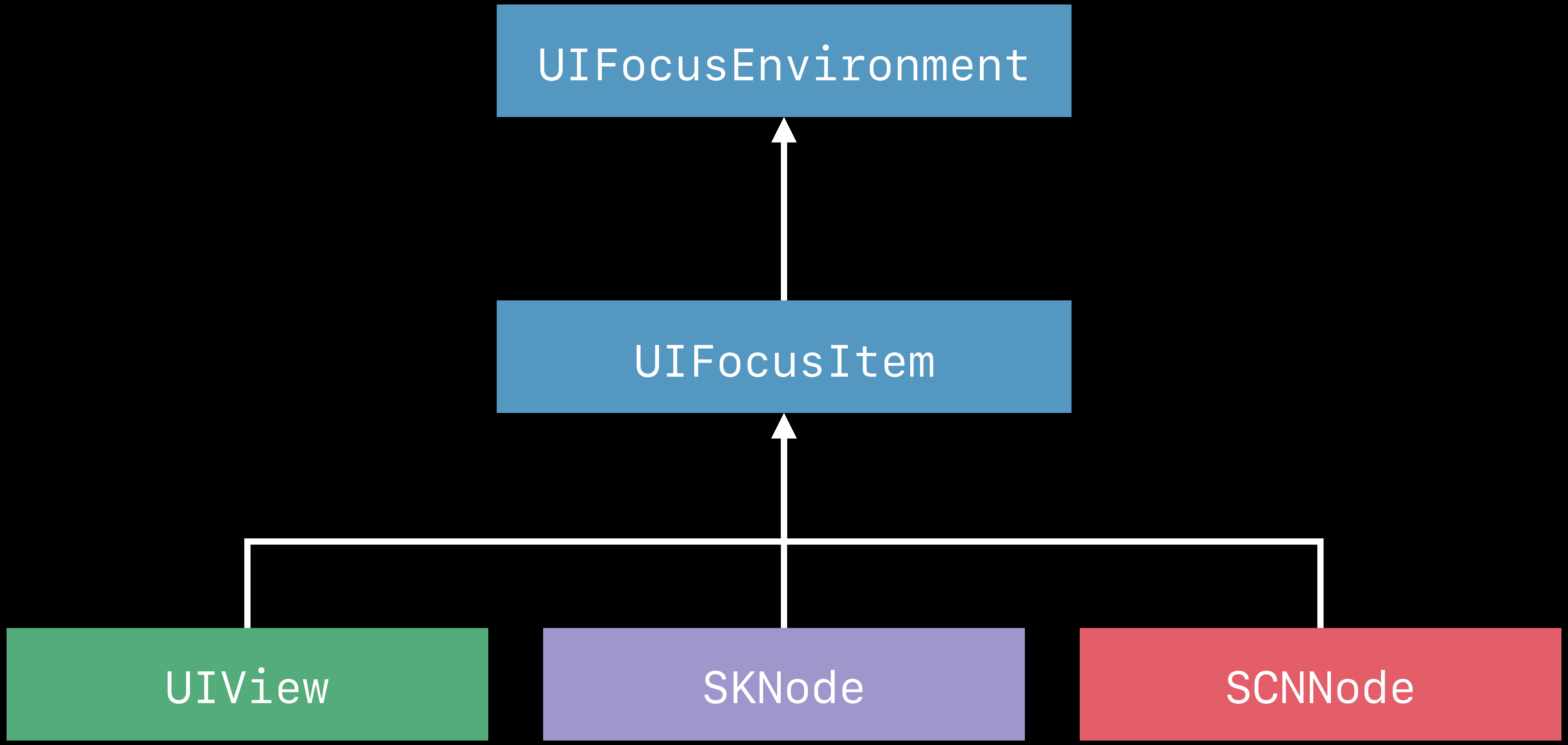
tvOS 10



Focus Item Types

tvOS 11

NEW



Support for SceneKit

Making nodes focusable

NEW

Opt-in to focus support for `SCNNode`

```
node.focusBehavior = .focusable
```

Also available on `SKNode`

Demo

Focus support in SceneKit

Support for SceneKit

Summary

Focus movement is two-dimensional, based on screen visibility

Focusable SpriteKit scenes on SceneKit surfaces are allowed

Supports moving focus between UIKit, SpriteKit, and SceneKit

All `UIFocusEnvironment` APIs available

SceneKit: What's New		WWDC 2017
Going Beyond 2D with SpriteKit	Executive Ballroom	Friday 10:00AM

New Debugging Tools

Focus Update Logging

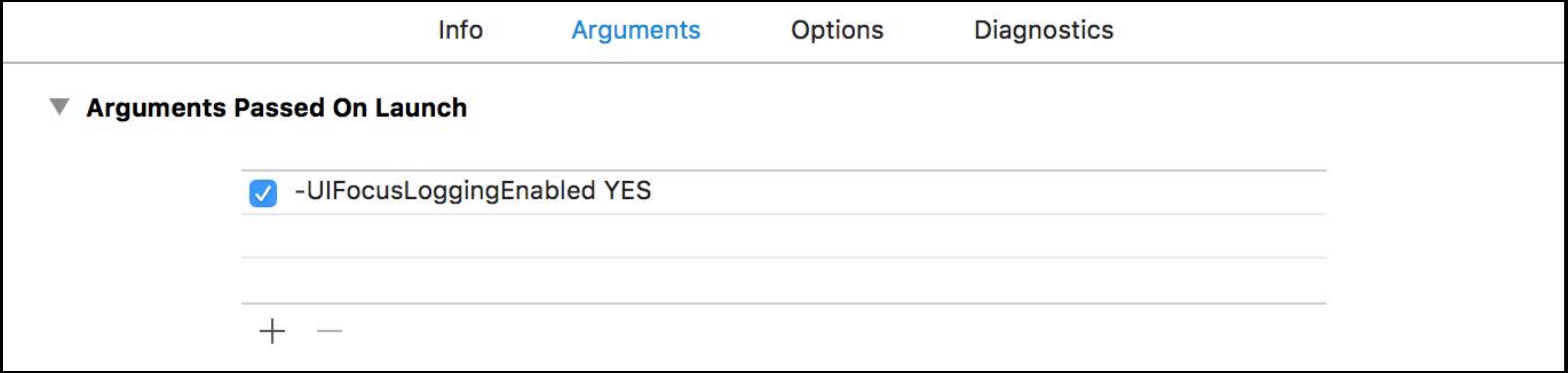
Focus Update Logging

NEW

Automatically logs focus updates while an application is running

Outlines preferred focus logic

Not for use in production apps (impacts performance)



▼ Arguments Passed On Launch

☑ -UIFocusLoggingEnabled YES

UIFocusDebugger

UIFocusDebugger

NEW

Used for diagnosing focus-related issues

Structured as a command line tool built into UIKit

Use from LLDB while app is paused, using `po`

Not for use in application logic

UIFocusDebugger

Status

NEW

```
(lldb) po UIFocusDebugger.status()
```

Provides the currently focused item

UIFocusDebugger

Simulating focus updates

NEW

```
(lldb) po UIFocusDebugger.simulateFocusUpdateRequest(from: environment)
```

Simulates focus update requests from a specific environment

- i.e. a successful call to `environment.setNeedsFocusUpdate()`

Outlines preferred focus logic used to determine new focus item

UIFocusDebugger

Checking focusability

NEW

```
(lldb) po UIFocusDebugger.checkFocusability(for: item)
```

Diagnoses potential issues affecting item focusability

Works on `SKNode` and `SCNNode` in addition to `UIView`

Replaces `_whyIsThisViewNotFocusable`

UIFocusDebugger

Checking focusability

NEW

```
(lldb) po UIFocusDebugger.checkFocusability(for: item)
```

UIView is hidden	Ancestor UIView is hidden
UIView alpha is zero	Ancestor UIView alpha is zero
UIView user interaction is disabled	Ancestor UIView user interaction is disabled
UIView is not in the a window	Ancestor UIView is not in a window
UIView returns false from canBecomeFocused	Ancestor UIView returns true from canBecomeFocused
UIControl is disabled	SpriteKit node not focusable
UITextView is not selectable	SpriteKit node user interaction disabled
UISegmentedControl has no enabled segments	SceneKit node not focusable
Focus items occluding other focus items	

UIFocusDebugger

Status

NEW

```
(lldb) po UIFocusDebugger.help()
```

Prints detailed instructions for all debugging commands

Demo

Focus Debugging

Summary

New focus update notifications, `isFocused`, and `contains()`

Provide consistent animations by targeting coordinated animations

Provide custom focus sounds for more immersive experiences

Use focus interaction in SceneKit for more consistent user experiences

Use `UIFocusDebugger` and focus update logging to diagnose focus issues

More Information

https://developer.apple.com/wwdc17/224

Related Sessions

Advances in TVMLKit		WWDC 2017
What’s New in tvOS		WWDC 2017
SceneKit: What's New		WWDC 2017
Going Beyond 2D with SpriteKit	Executive Ballroom	Friday 10:00AM

WWDC17