

# Enhancing VoIP Apps with CallKit

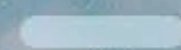
Session 230

Sirisha Pillalamarri CallKit Engineer

Stuart Montgomery CallKit Engineer

Nick Fraioli CallKit Engineer

How do VoIP apps work today?



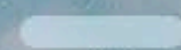
100% 

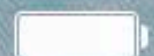
9:41

Thursday, June 16

> slide to unlock





100% 

9:41

Thursday, June 16

> slide to unlock





Messages



Calendar



Photos



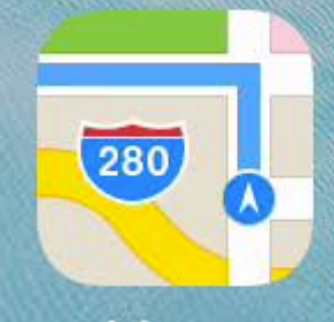
Camera



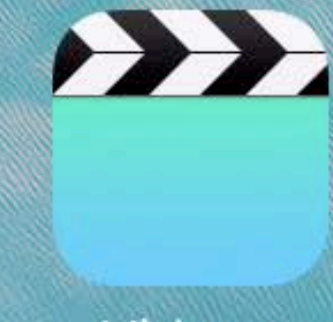
Weather



Clock



Maps



Videos



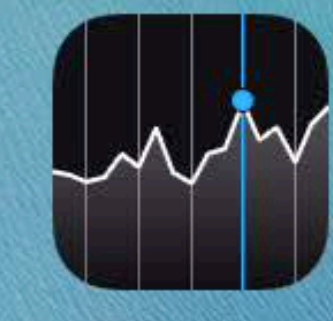
Wallet



Notes



Reminders



Stocks



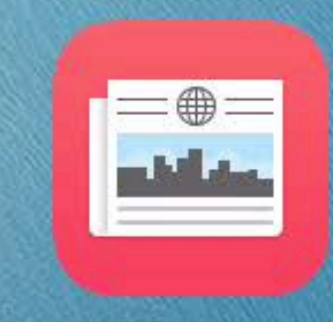
iTunes Store



App Store



iBooks



News



Health



Settings



Phone



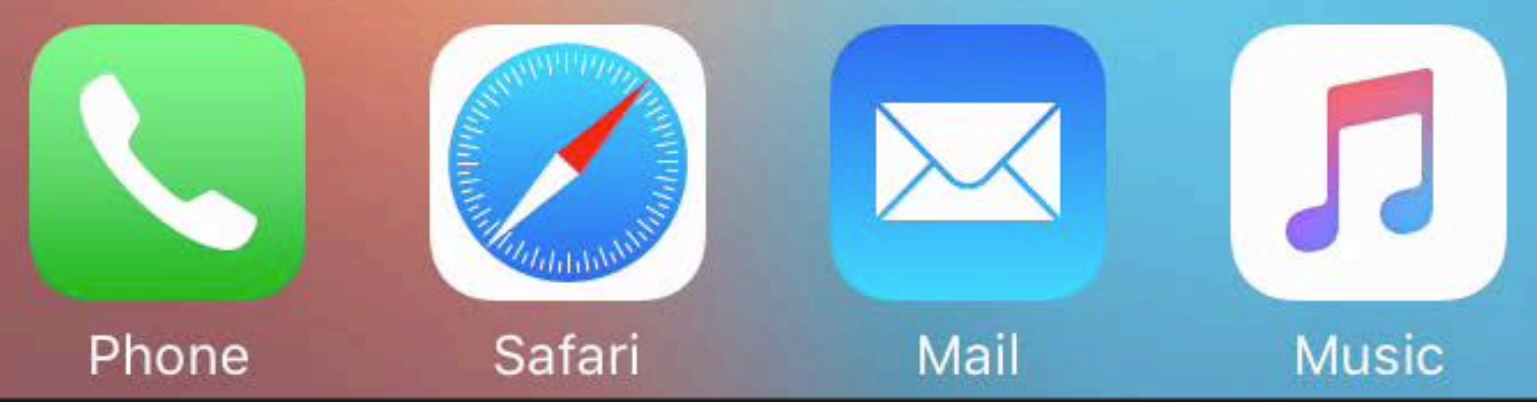
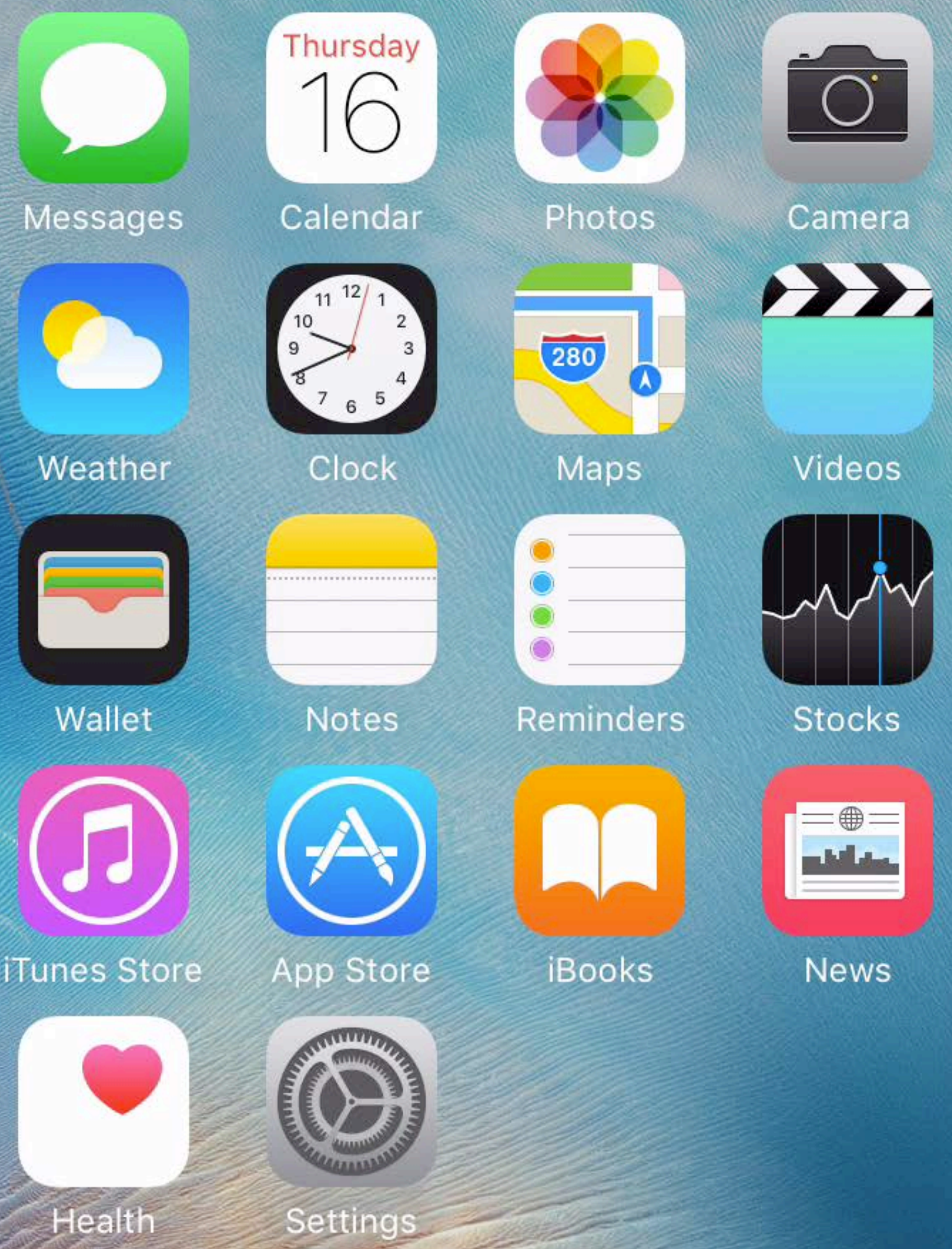
Safari



Mail



Music



Wouldn't it be nice if...



100%

9:41

Thursday, June 16

Press home to open







100%

9:41

Thursday, June 16

Press home to open



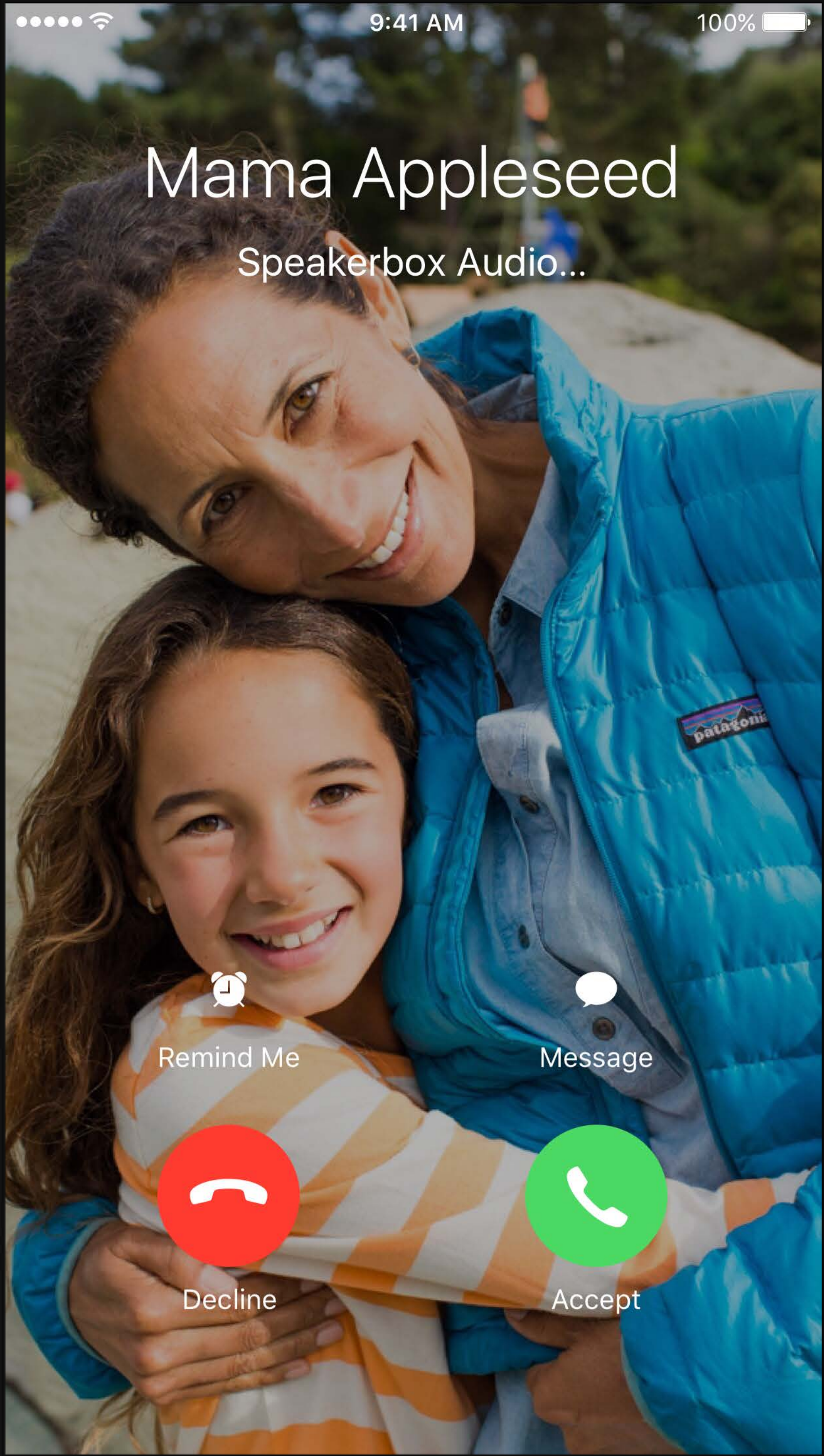


9:41 AM

100%

# Mama Appleseed

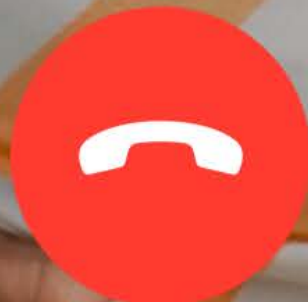
Speakerbox Audio...



Remind Me



Message



Decline



Accept



9:41 AM

100%

Papa Appleseed

HOLD

Mama Appleseed

00:02



mute



keypad



speaker



merge calls



swap



Speakerbox



All Missed

Edit

Mama Appleseed

Yesterday 

Speakerbox Audio



Favorites



Recents



Contacts



Keypad



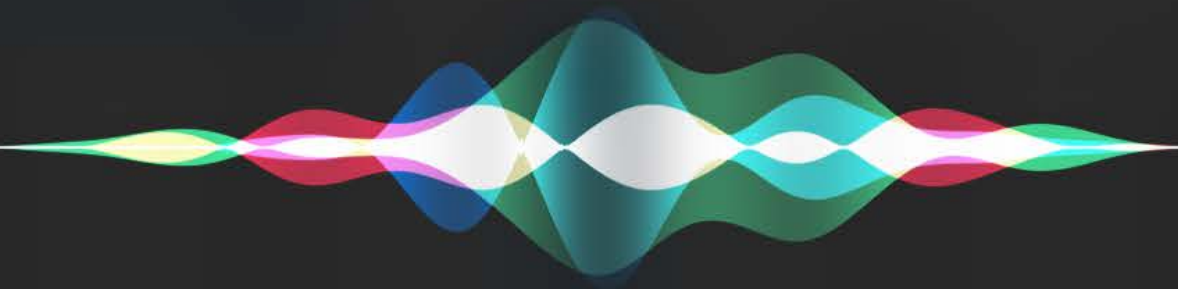
Voicemail



9:41 AM

100% 

What can I help  
you with?





9:41 AM

100%

# Mama Appleseed

Speakerbox Audio 00:02



mute



keypad



speaker



add call



video



Speakerbox





9:41 AM

100%

## Settings

 Airplane Mode

 Wi-Fi >

 Bluetooth >


 Cellular >

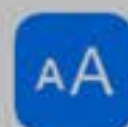
 Personal Hotspot >

 Notifications >

 Control Center >

 Do Not Disturb >

 General >

 Display & Brightness >

 Wallpaper >

 Sounds >

 Siri >



CallKit



# Agenda

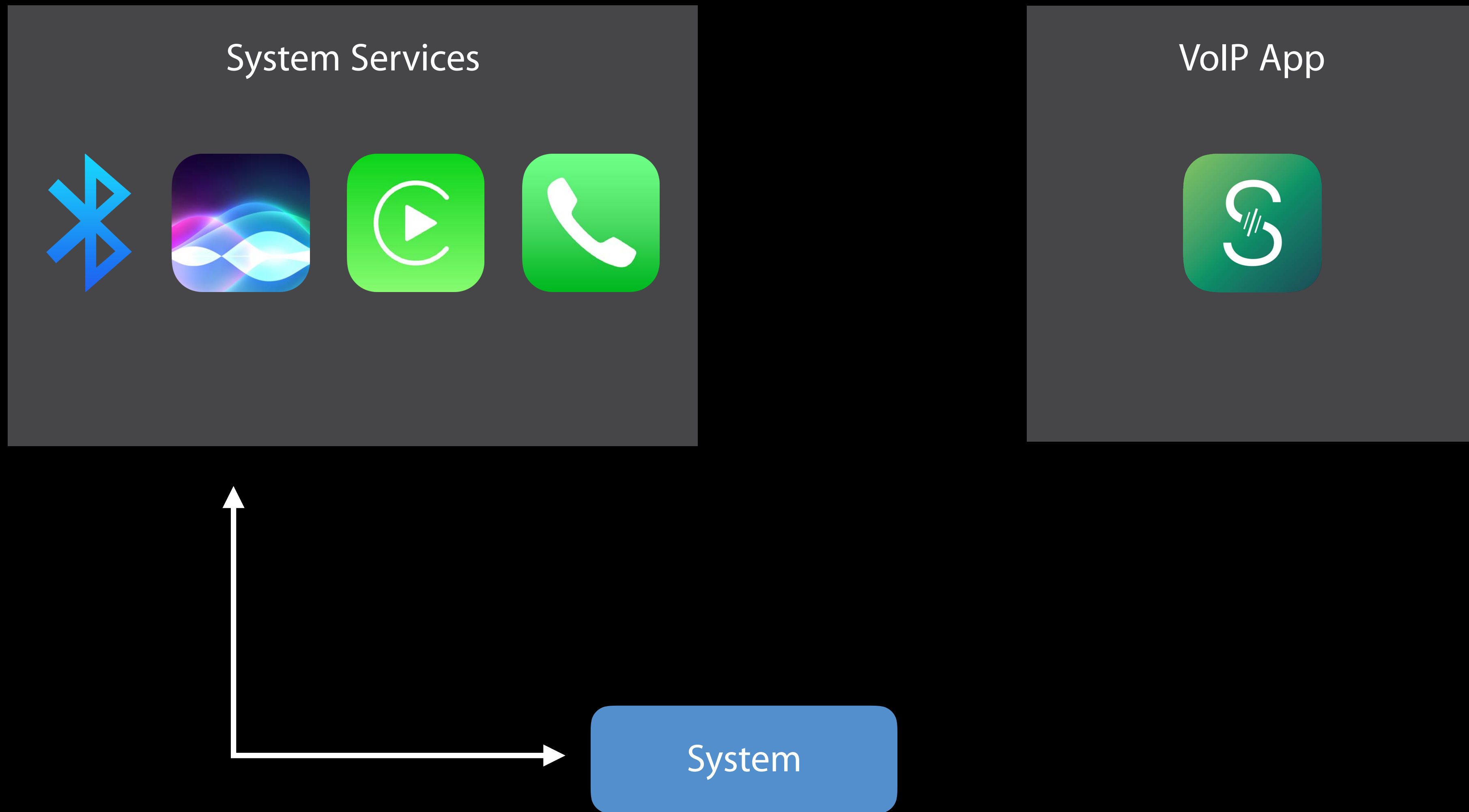
Architecture

Incoming Call

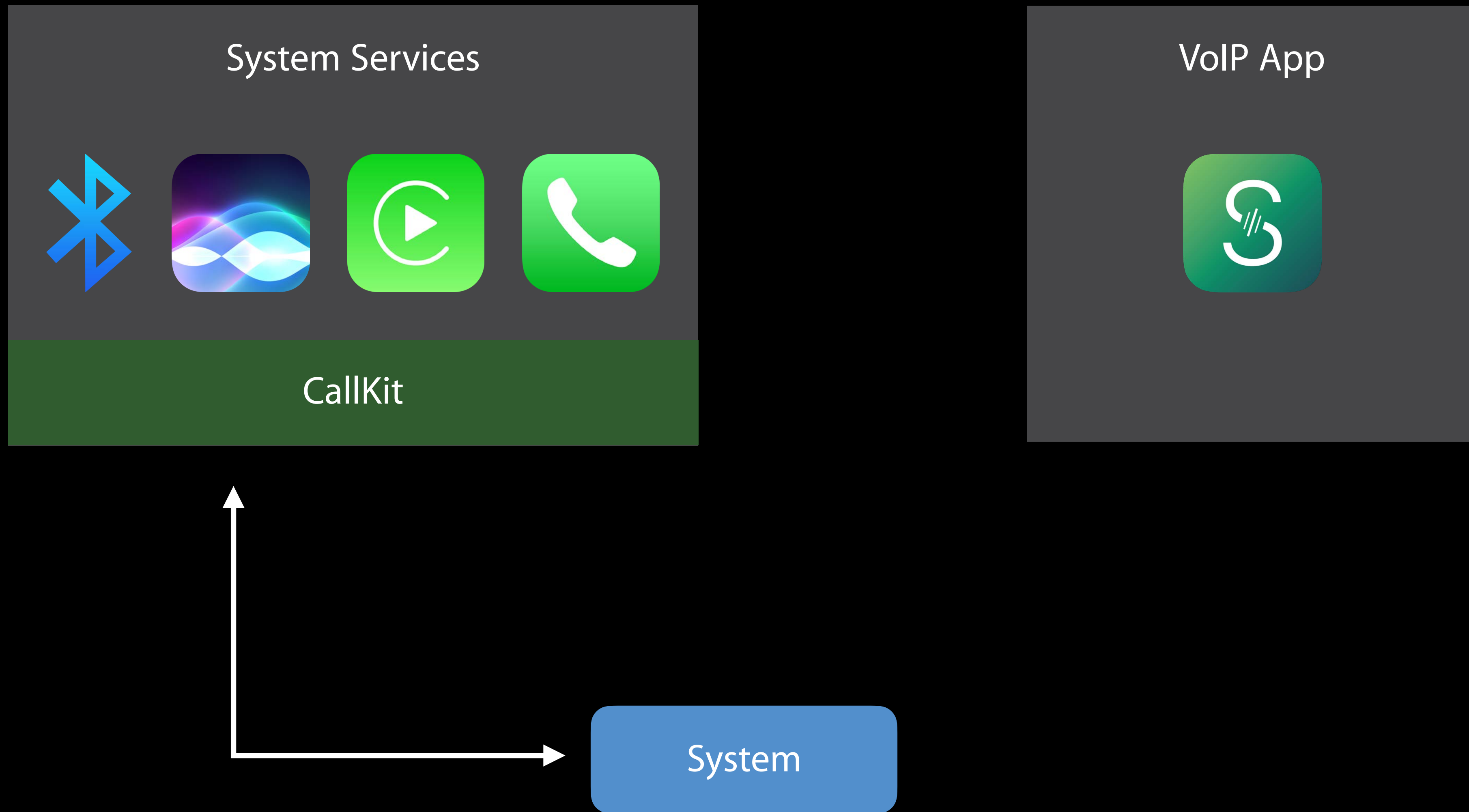
Outgoing Call

API Details

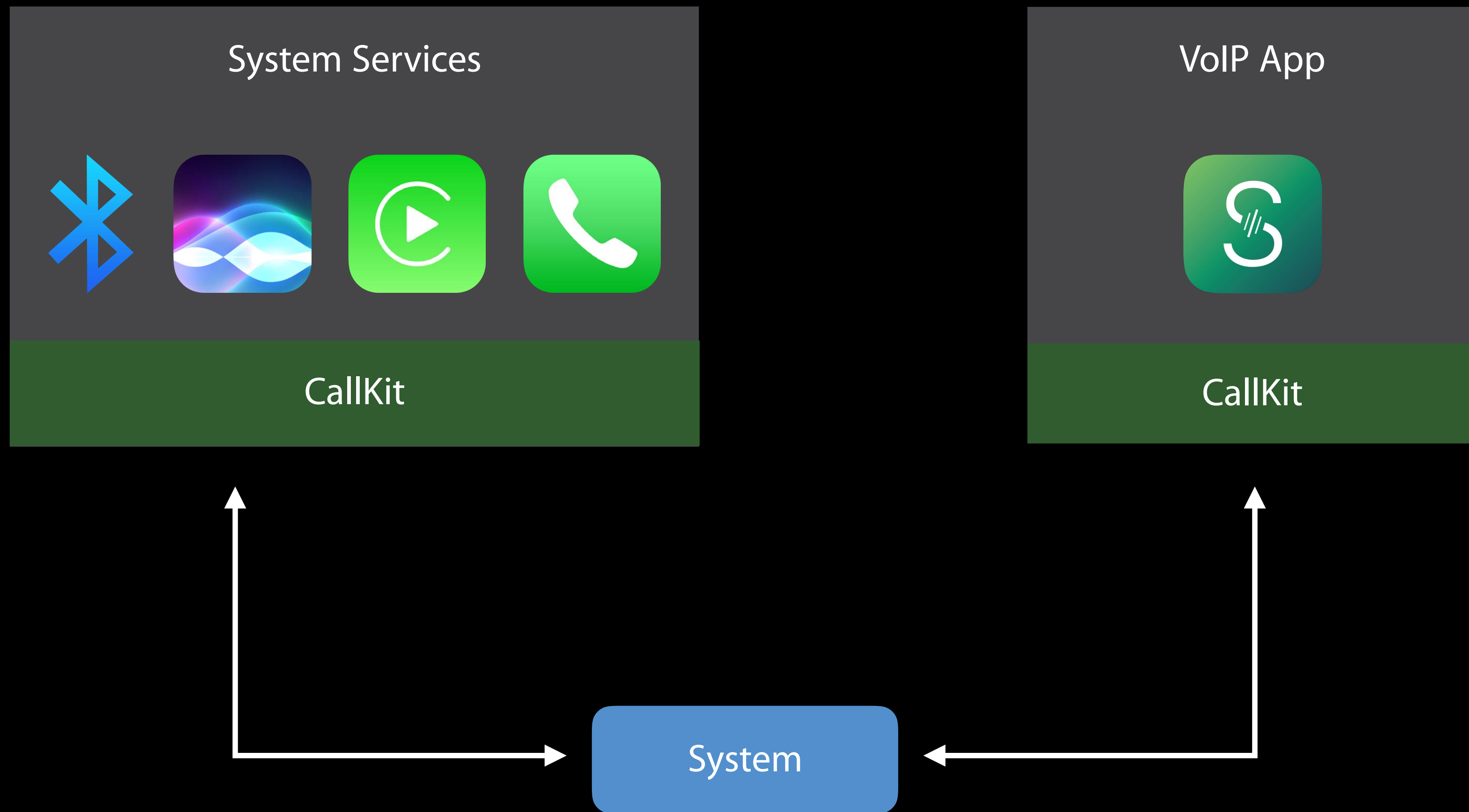
# Architecture



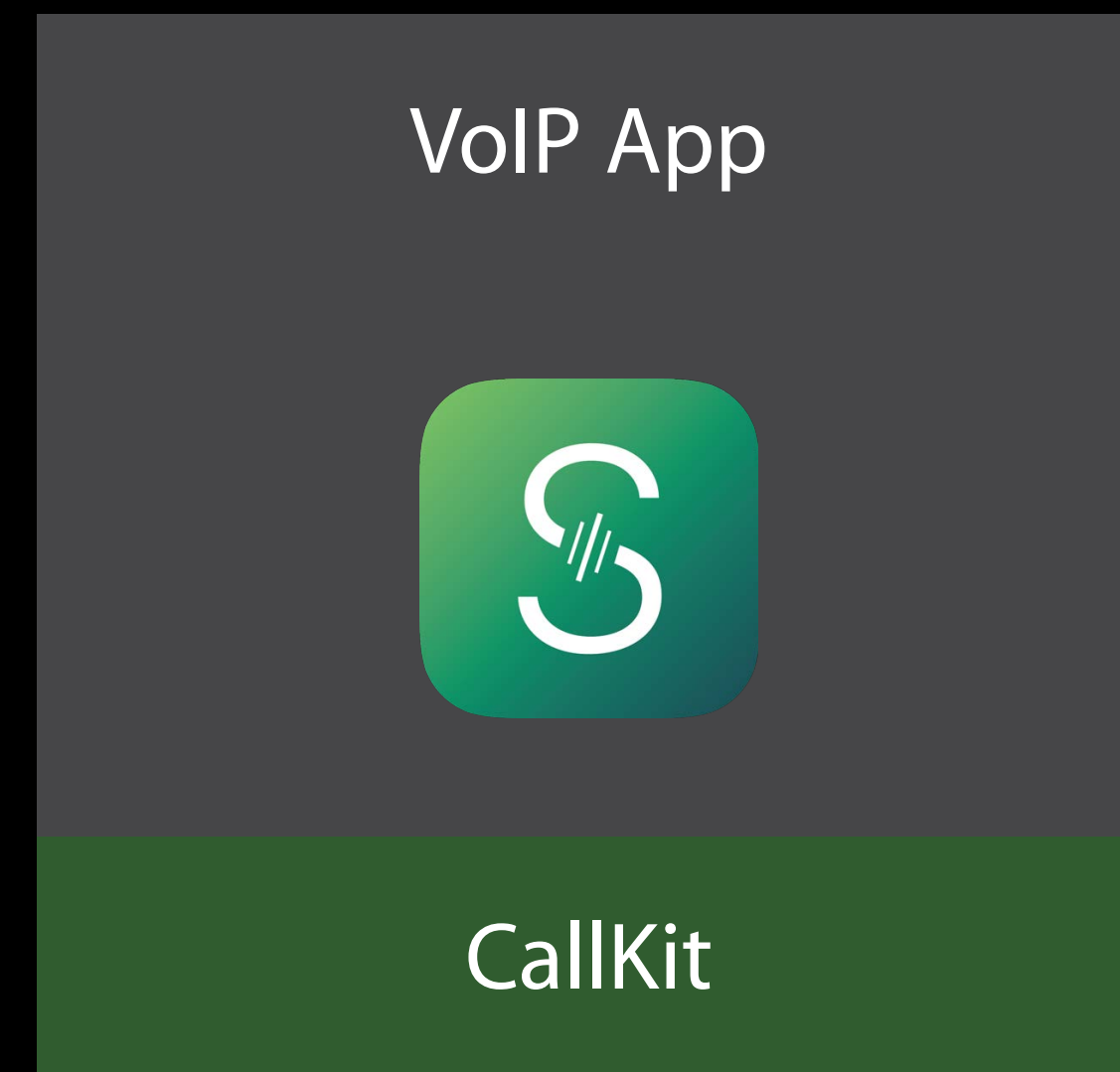
# Architecture



# Architecture



# Architecture



System

# Architecture



# Architecture



System

# Architecture



System

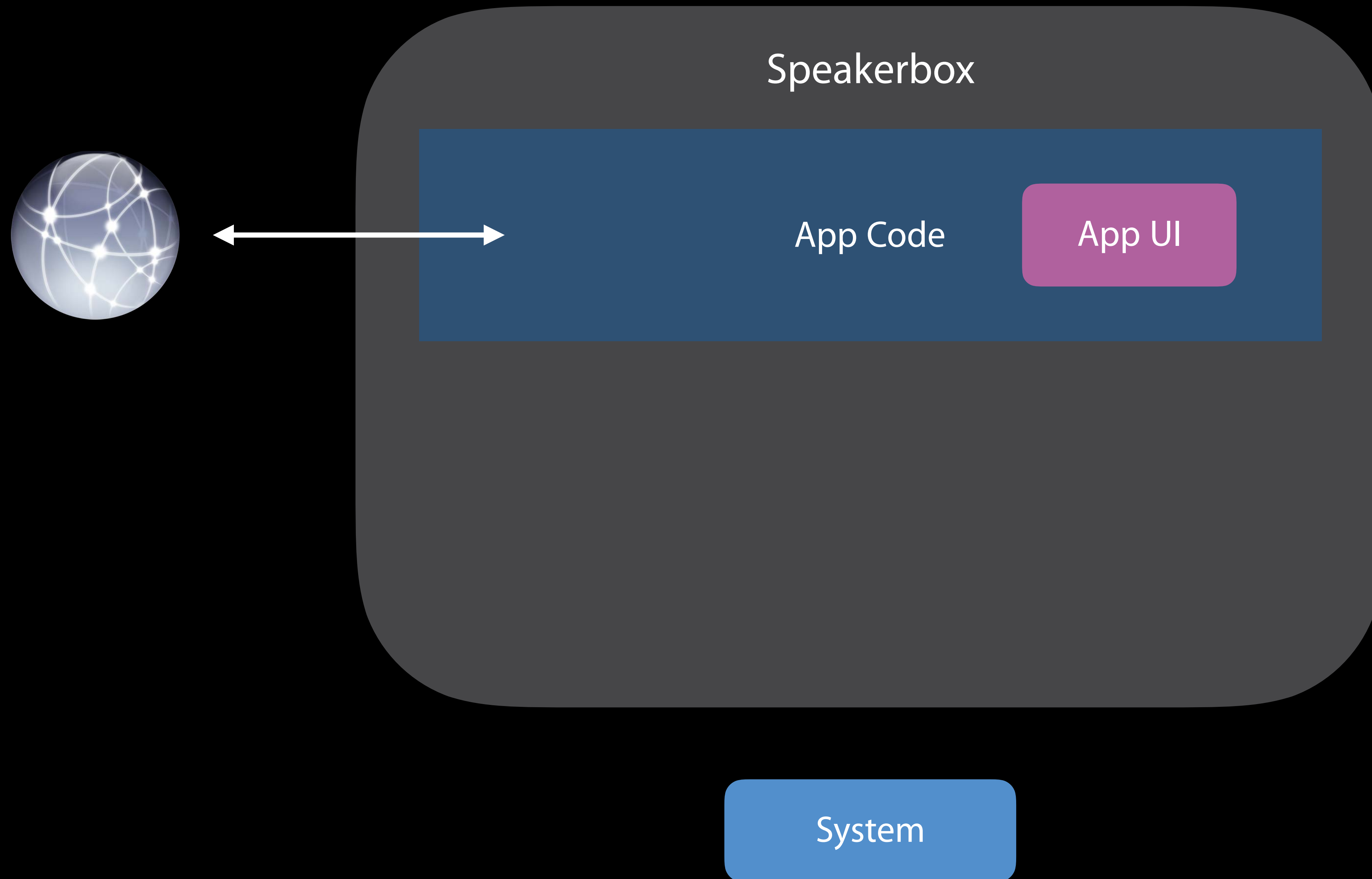


# Architecture

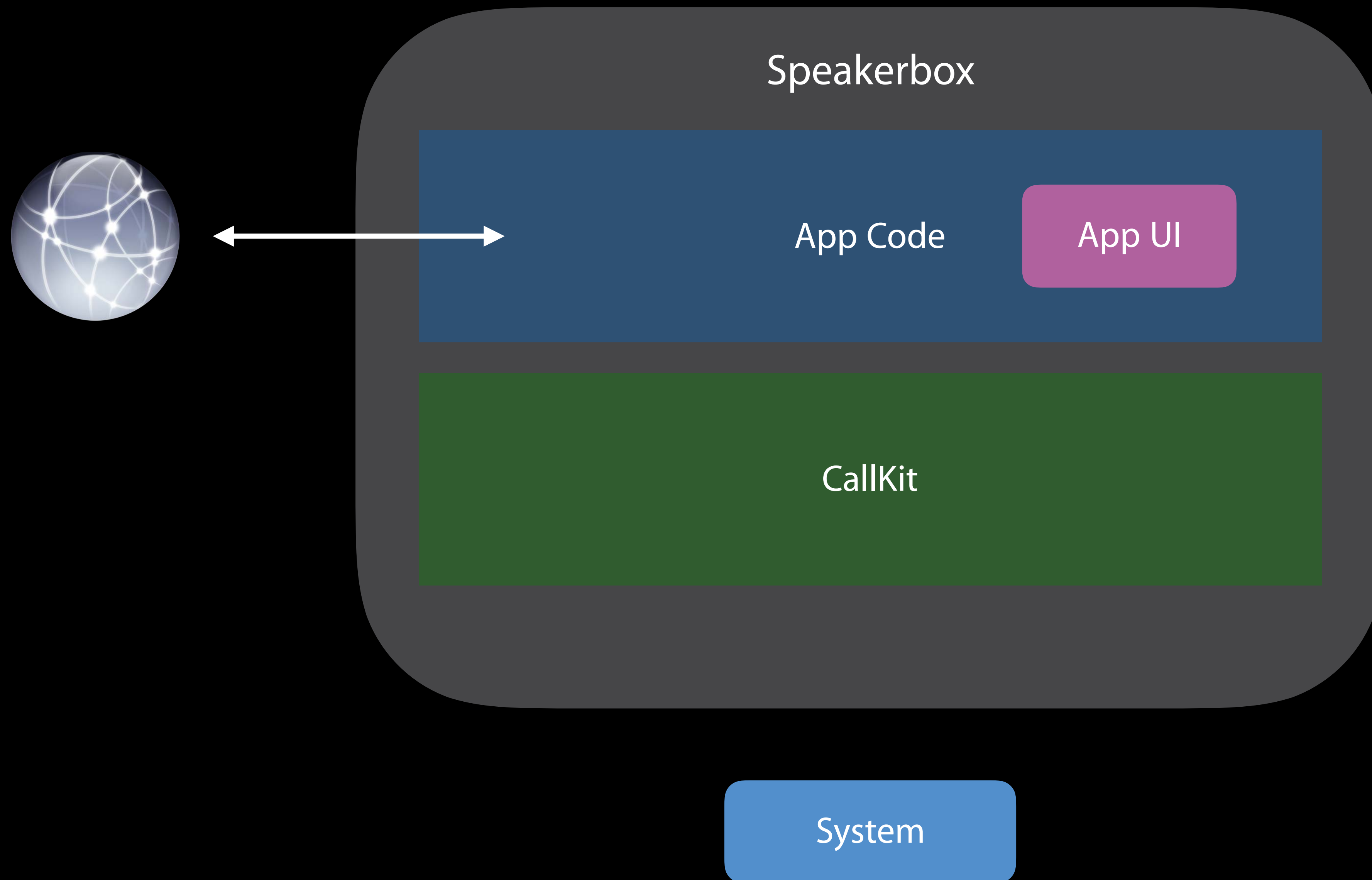
Speakerbox

System

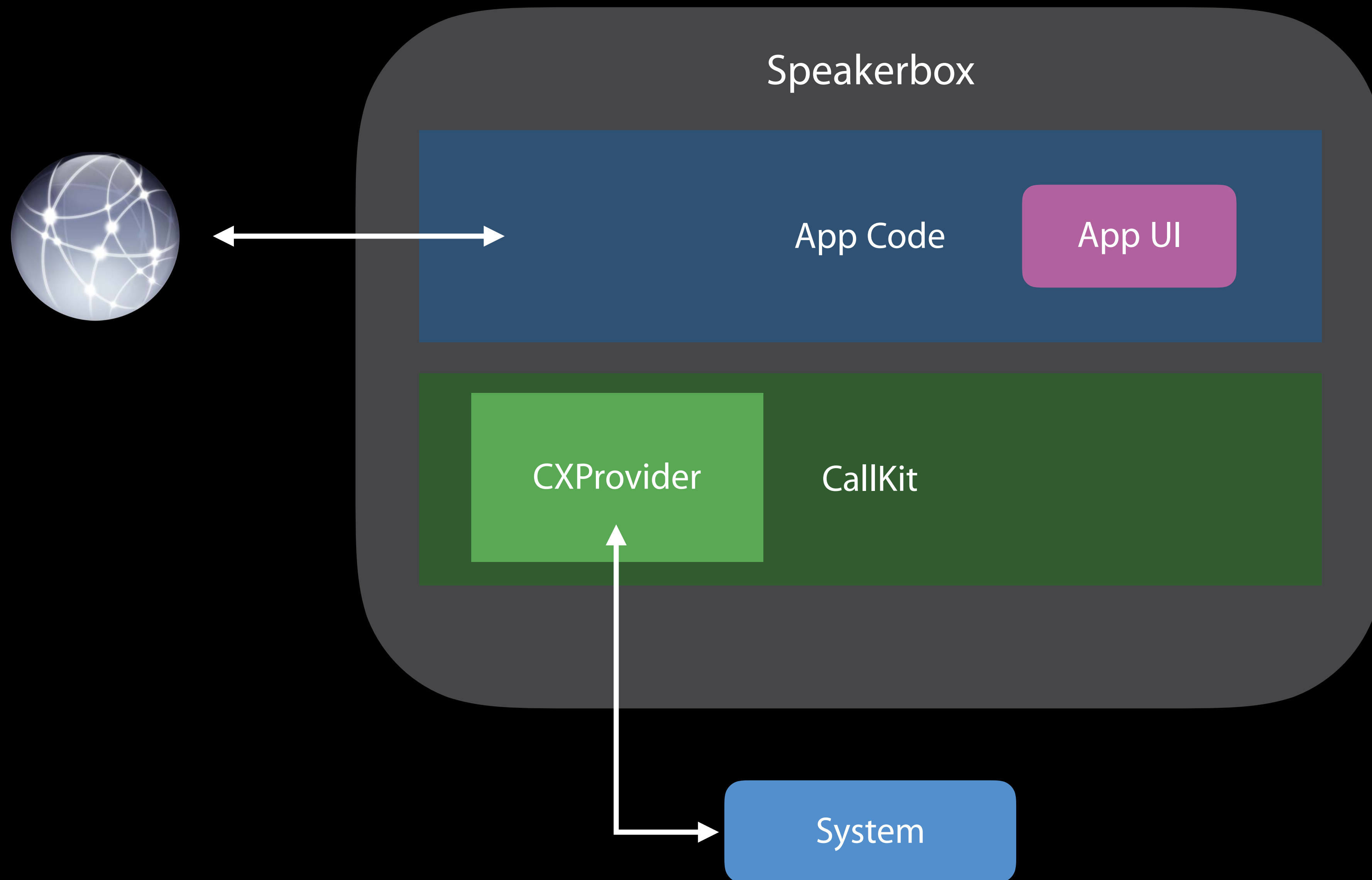
# Architecture



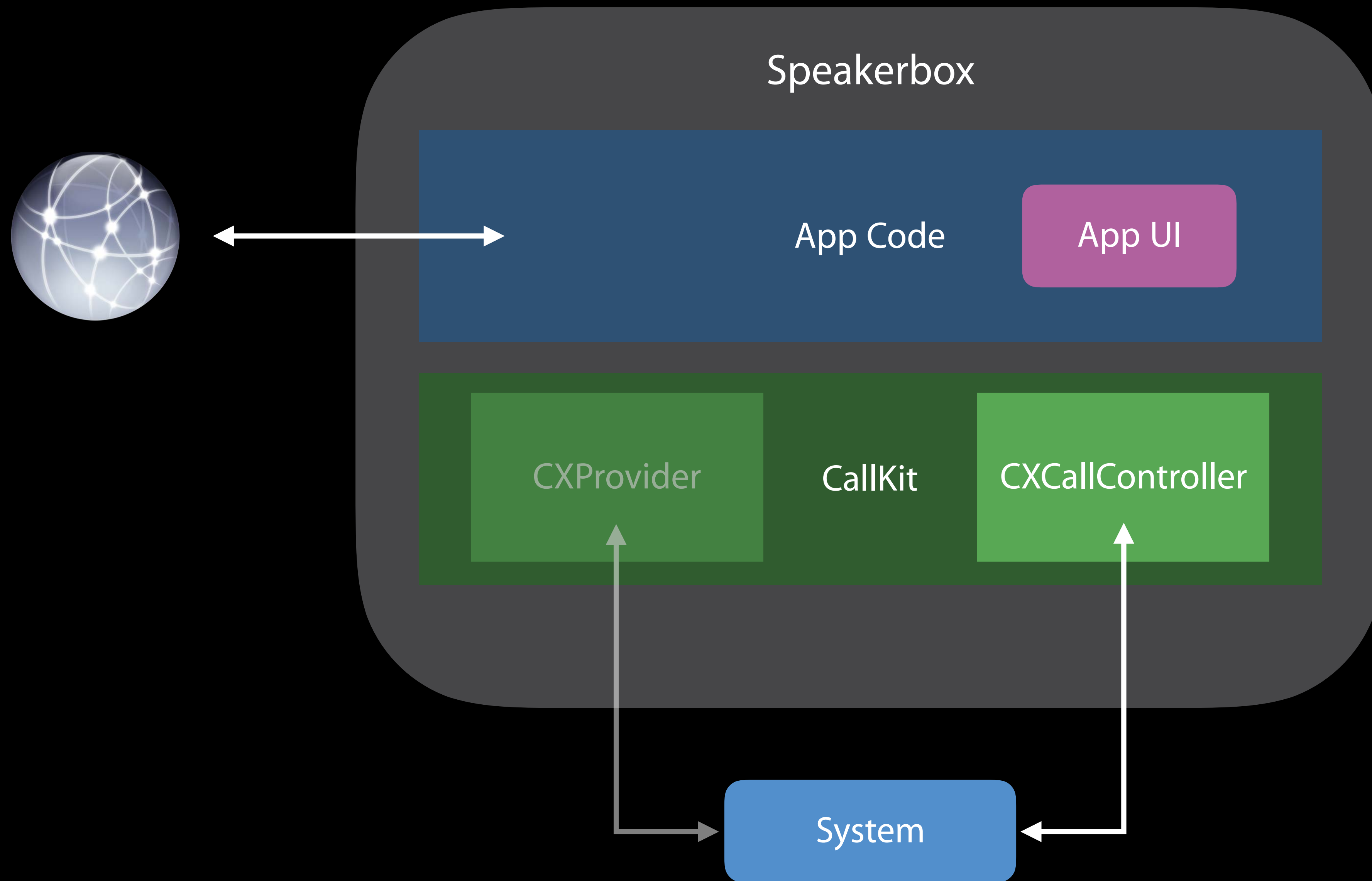
# Architecture



# Architecture



# Architecture



# CXProvider vs. CXCallController

# CXProvider vs. CXCallController

CXProvider

# CXProvider vs. CXCallController

CXProvider

Out-of-band notifications



# CXProvider vs. CXCallController

## CXProvider

Out-of-band notifications

Not user actions

# CXProvider vs. CXCallController

## CXProvider

Out-of-band notifications

Not user actions

External events

- Incoming call

# CXProvider vs. CXCallController

CXProvider

CXCallController

Out-of-band notifications

Not user actions

External events

- Incoming call

# CXProvider vs. CXCallController

## CXProvider

Out-of-band notifications

Not user actions

External events

- Incoming call

## CXCallController

Requests from app

# CXProvider vs. CXCallController

## CXProvider

Out-of-band notifications

Not user actions

External events

- Incoming call

## CXCallController

Requests from app

Local user actions

# CXProvider vs. CXCallController

## CXProvider

Out-of-band notifications

Not user actions

External events

- Incoming call

## CXCallController

Requests from app

Local user actions

Internal events

- Start call

# CXProvider vs. CXCallController

## CXProvider

Out-of-band notifications

Not user actions

External events

- Incoming call

## CXCallController

Requests from app

Local user actions

Internal events

- Start call

Interplay with other providers

- Hold and Start Call

# CXProvider vs. CXCallController

Example uses



# CXProvider vs. CXCallController

Example uses

Use CXProvider to report

Incoming call

Outgoing call connected

Call ended on remote side

# CXProvider vs. CXCallController

## Example uses

Use CXProvider to report

Incoming call

Outgoing call connected

Call ended on remote side

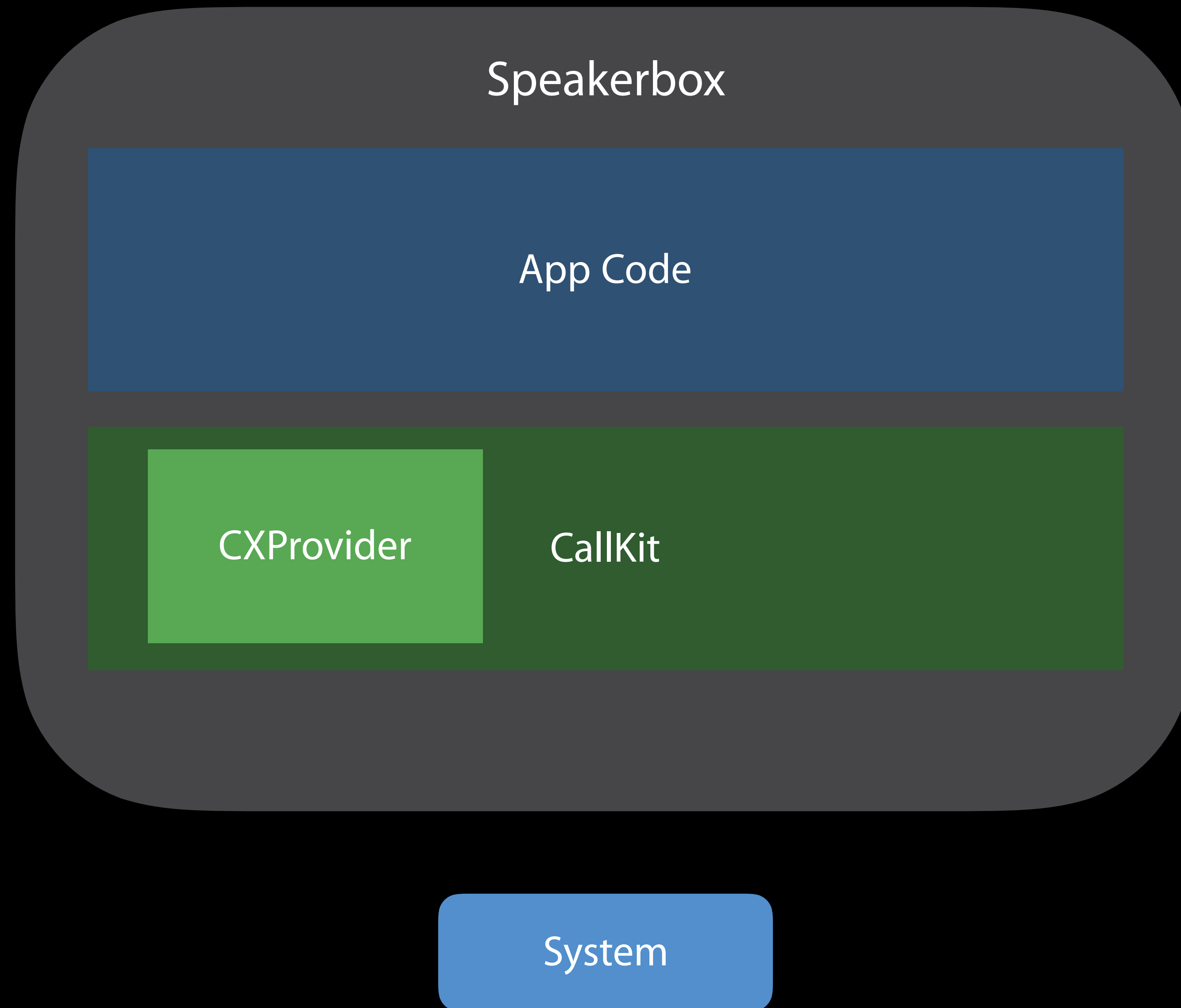
Use CXCallController to request

Start outgoing call

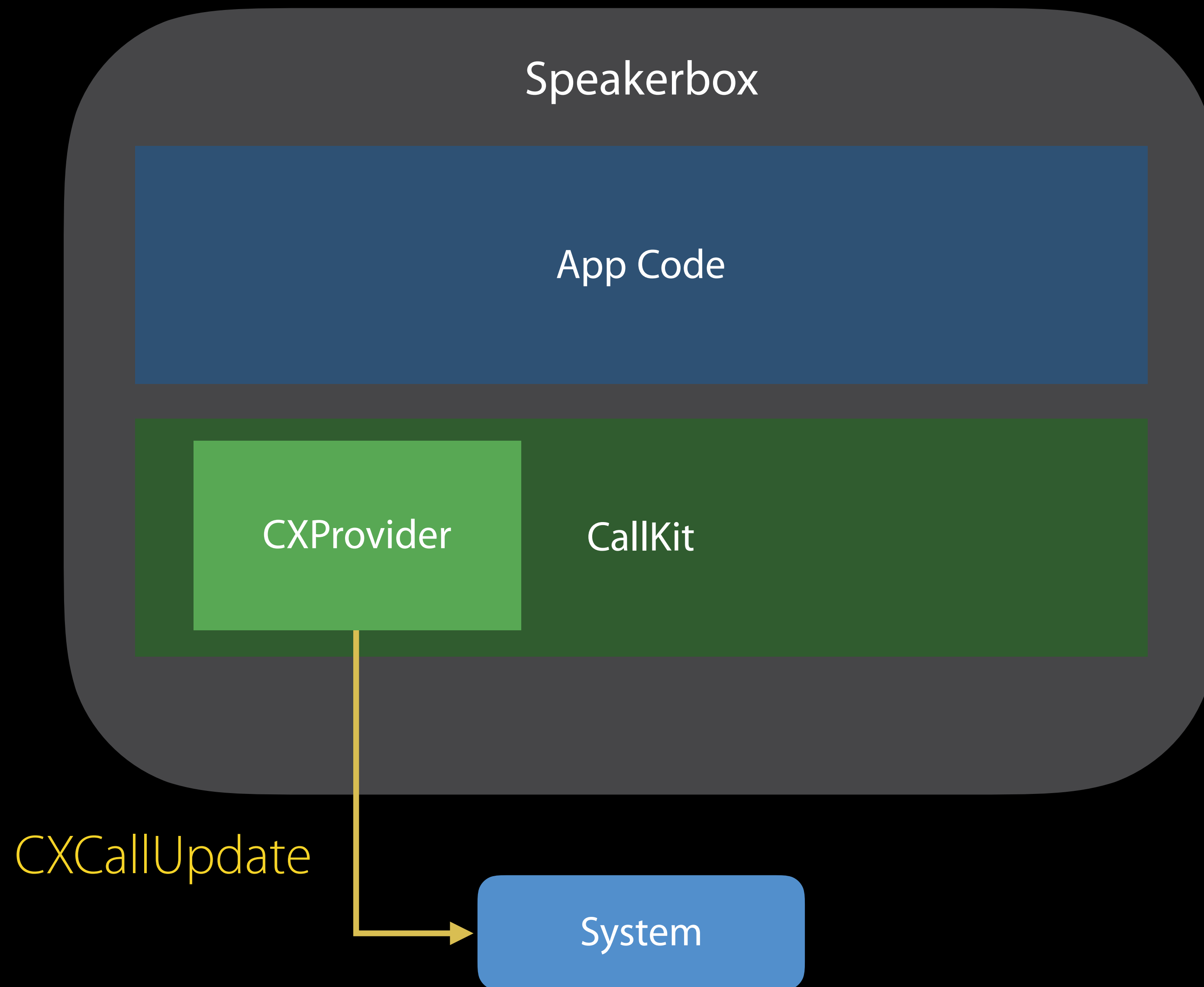
Answer call

End call

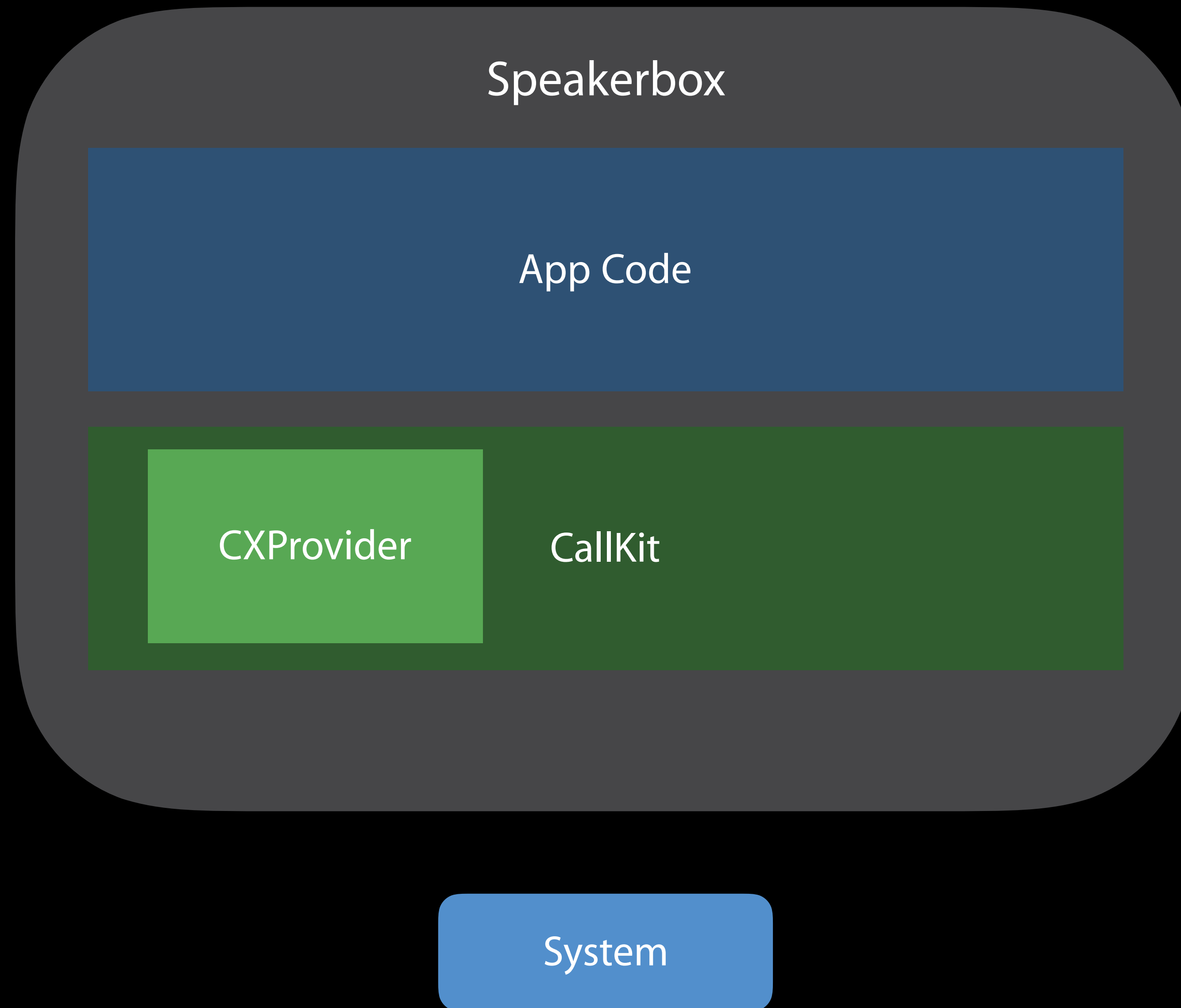
# Architecture



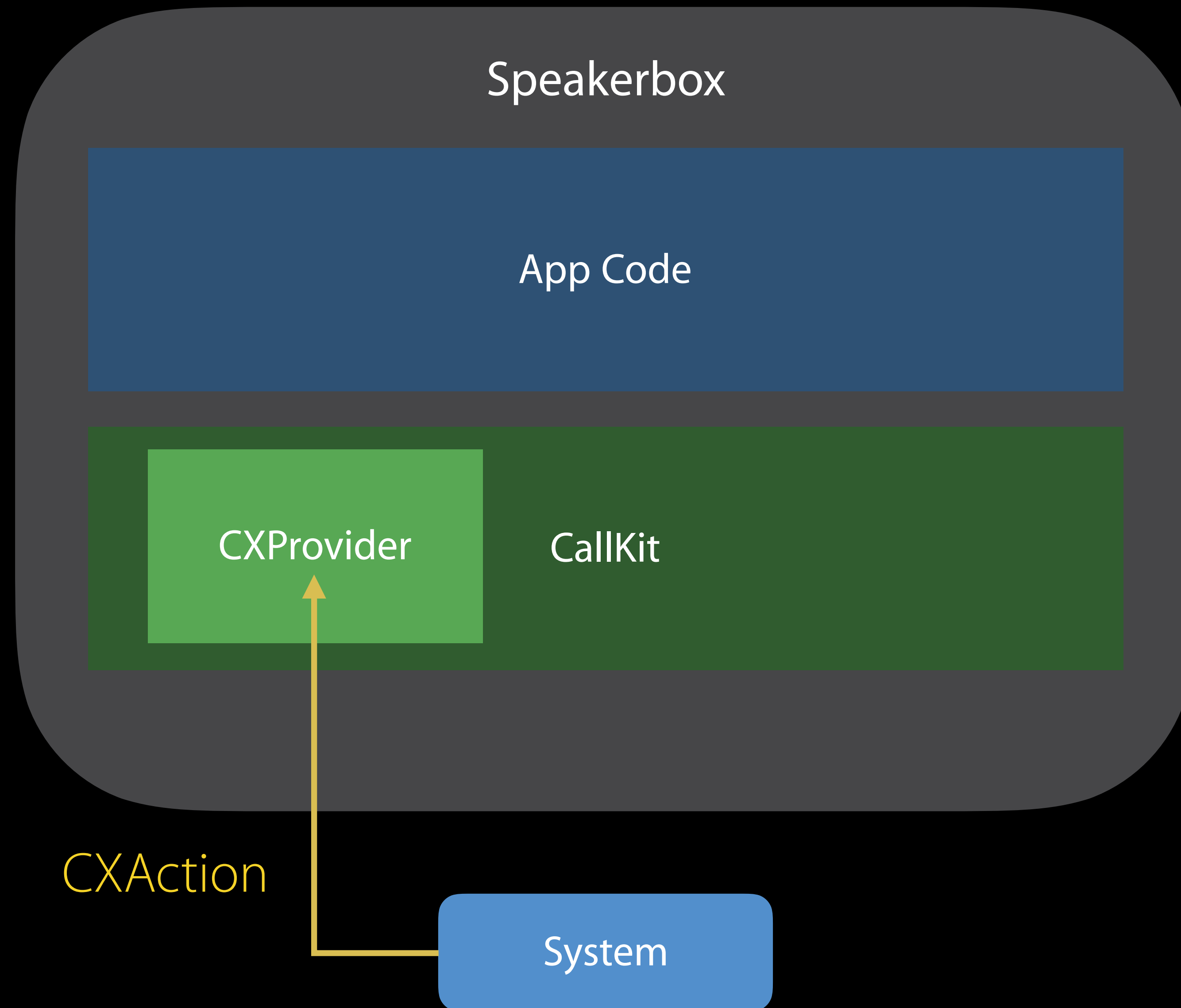
# Architecture



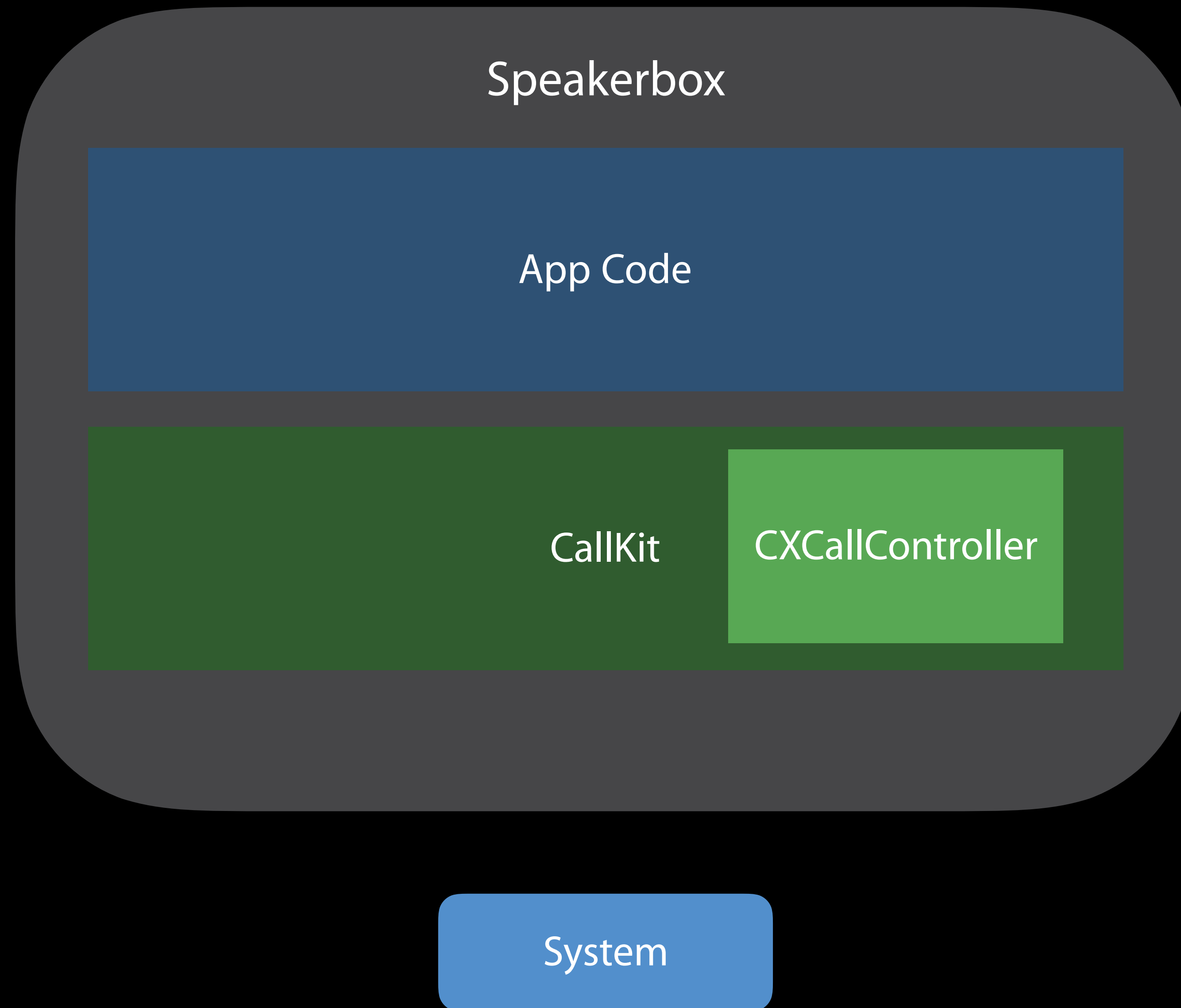
# Architecture



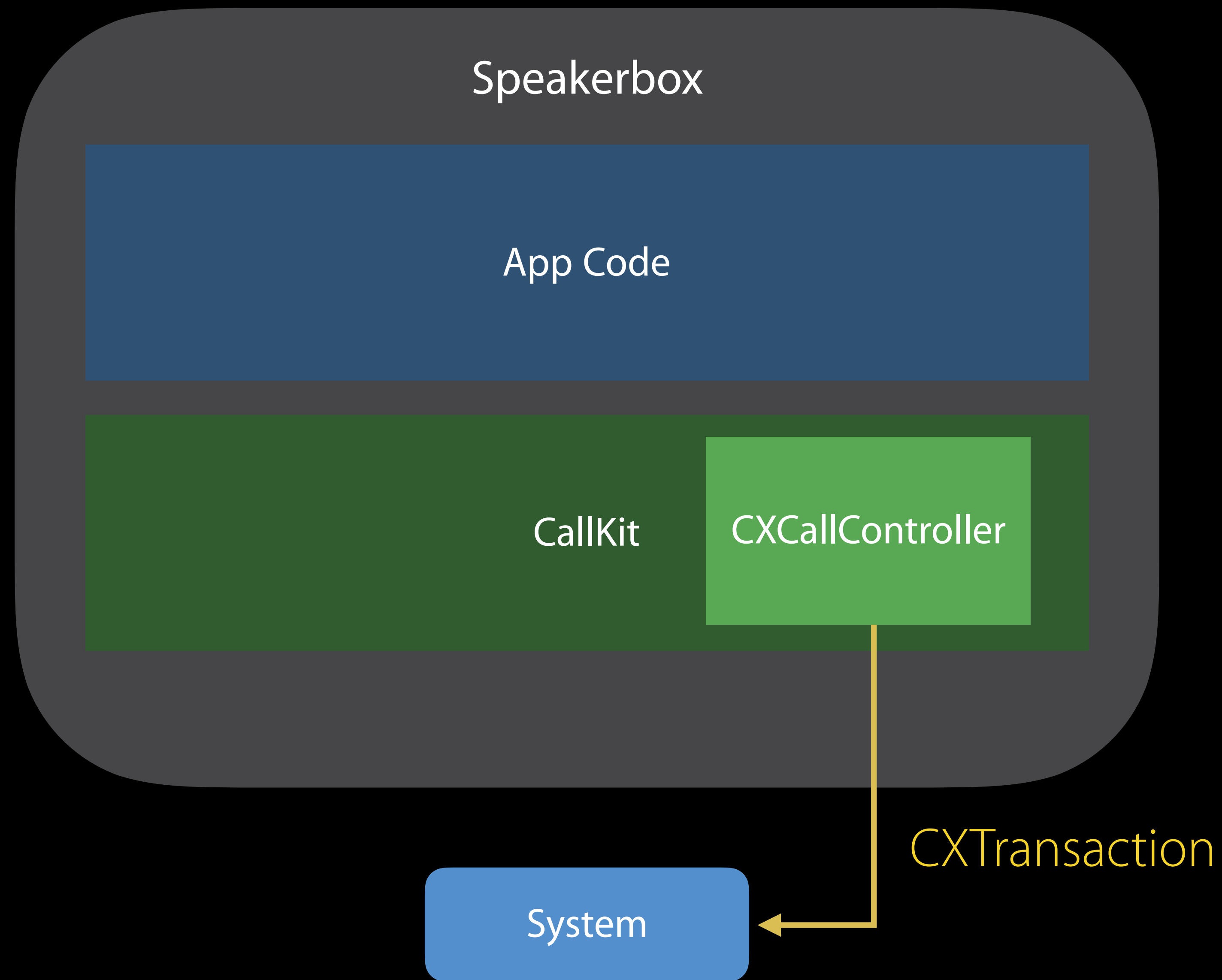
# Architecture



# Architecture



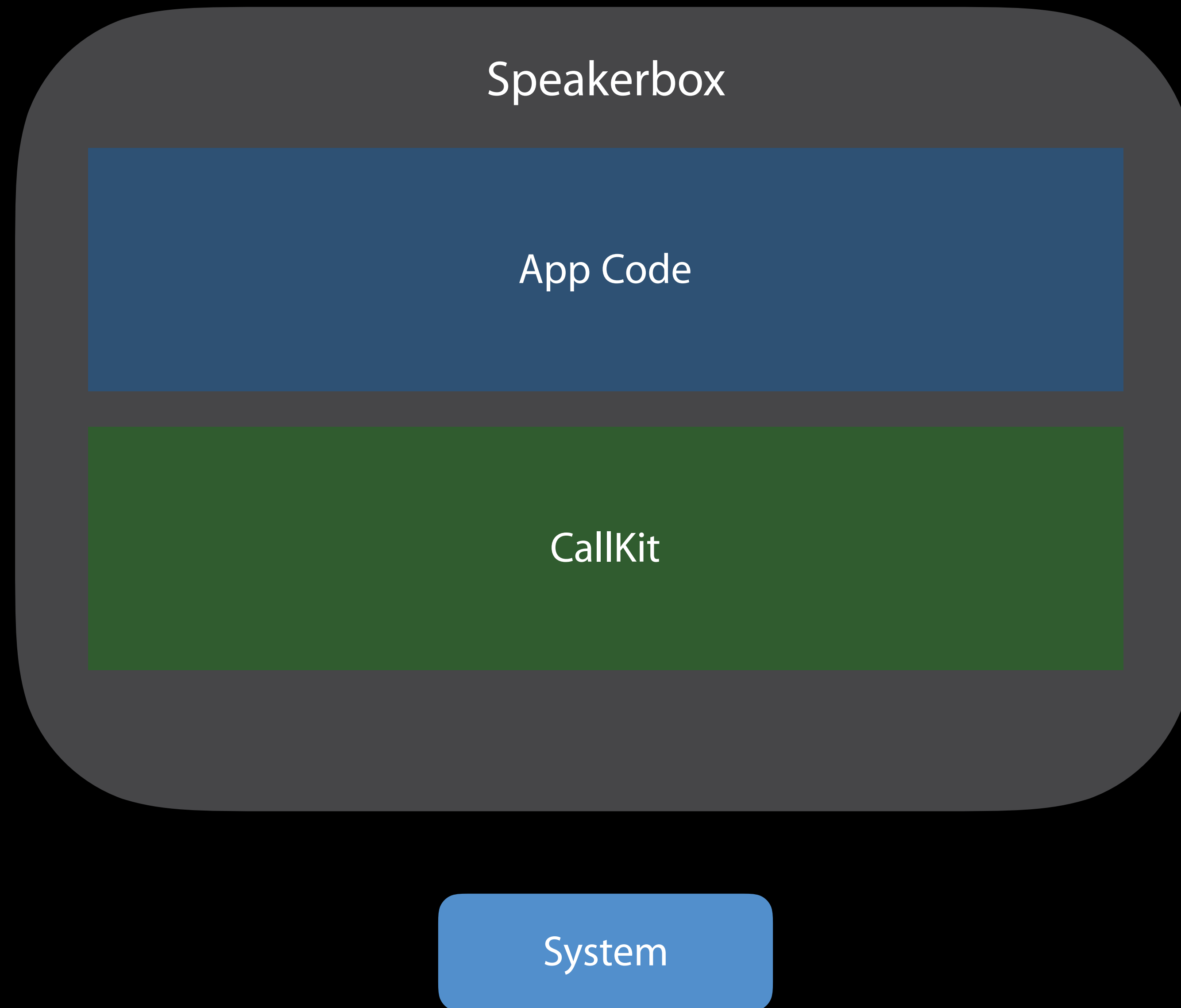
# Architecture



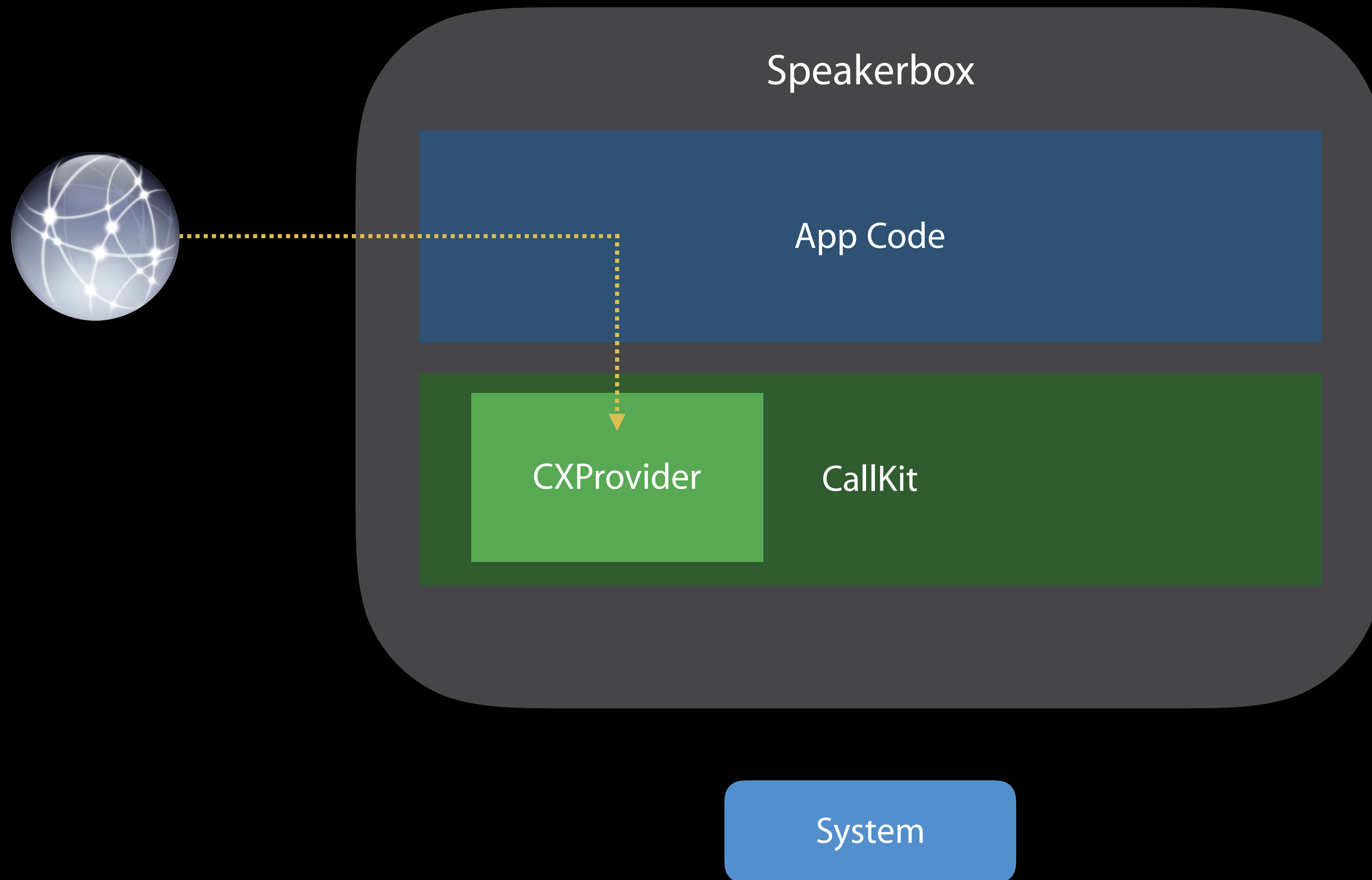


Incoming Call

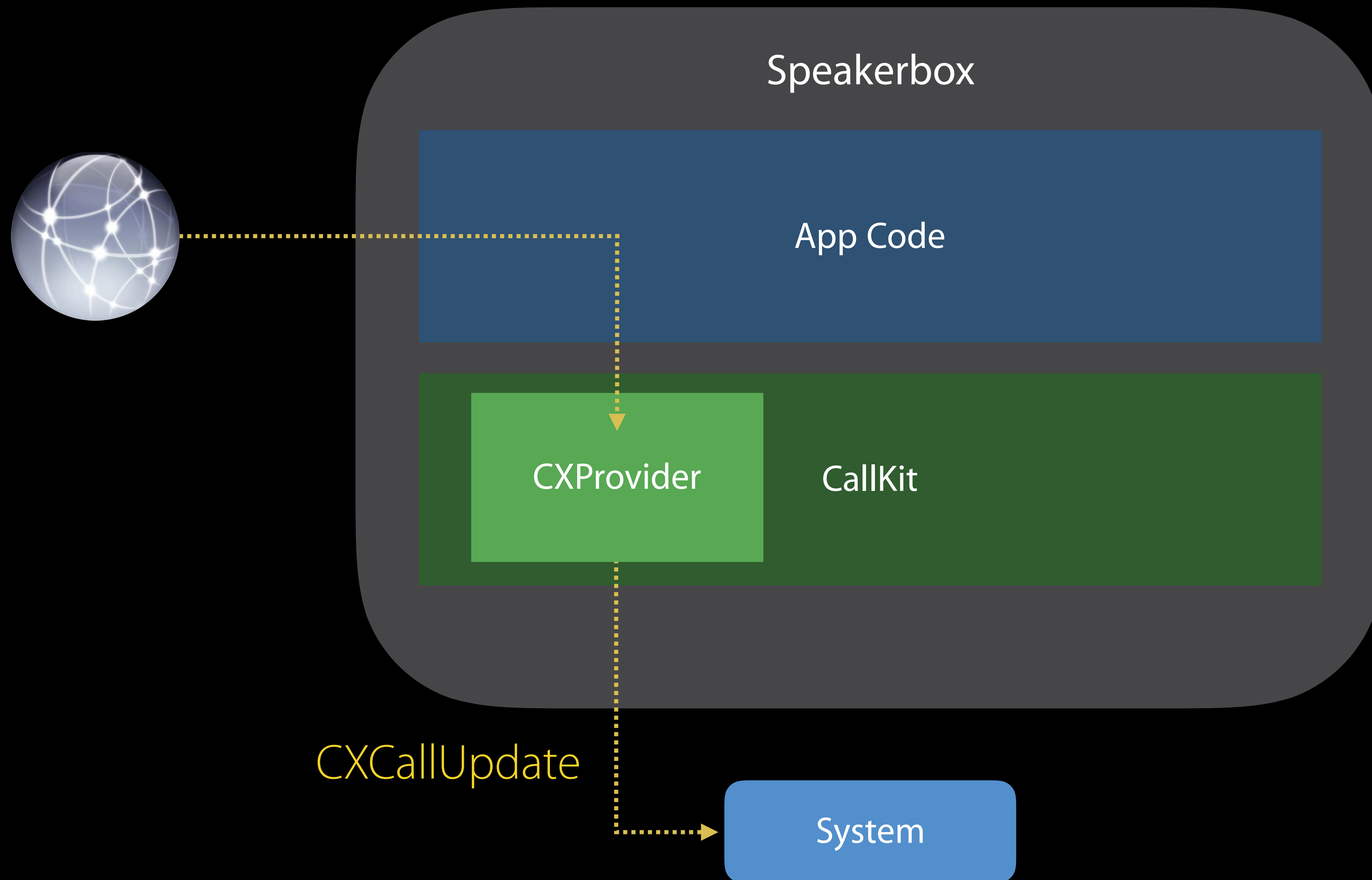
# Architecture



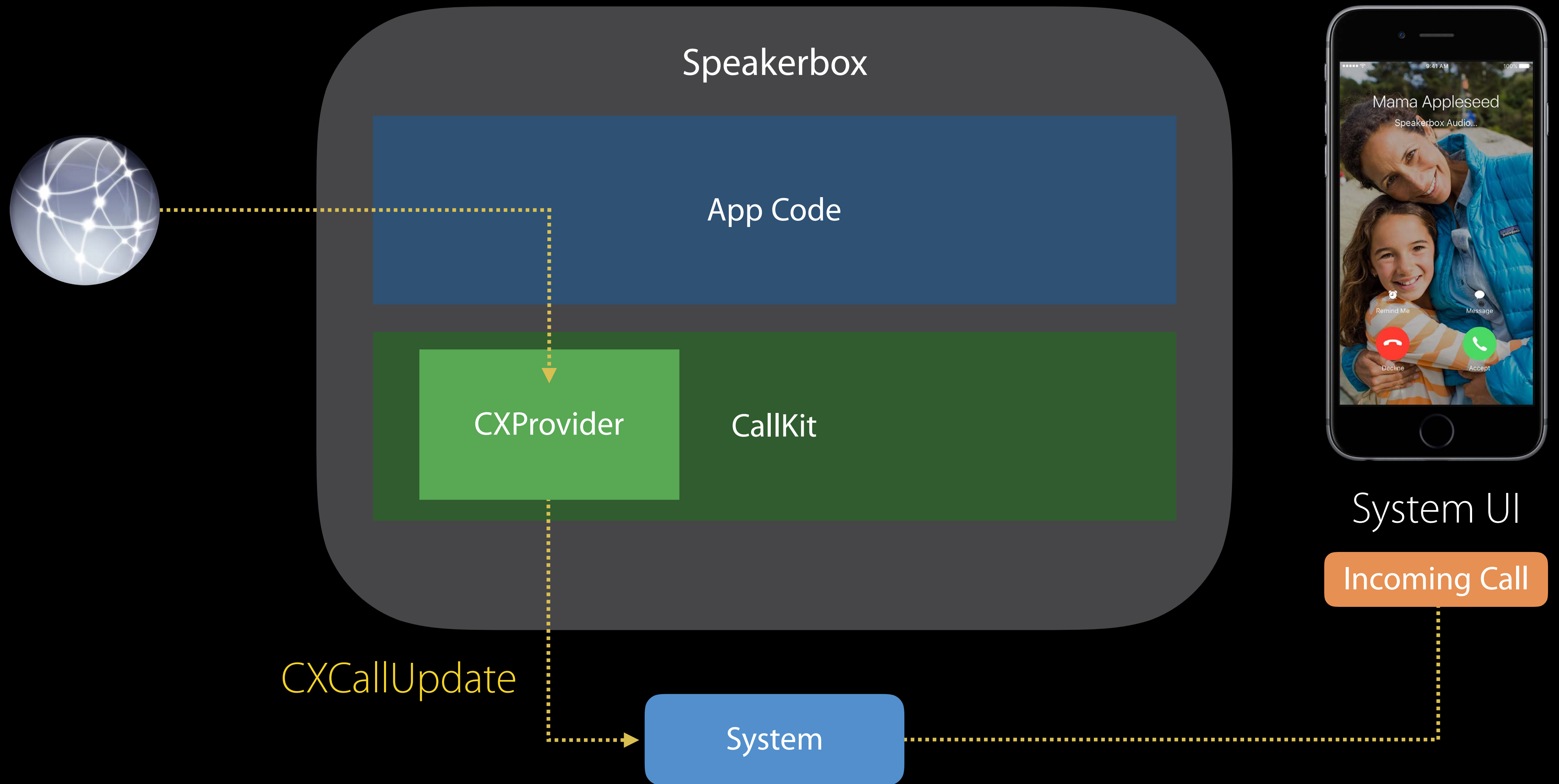
# Architecture



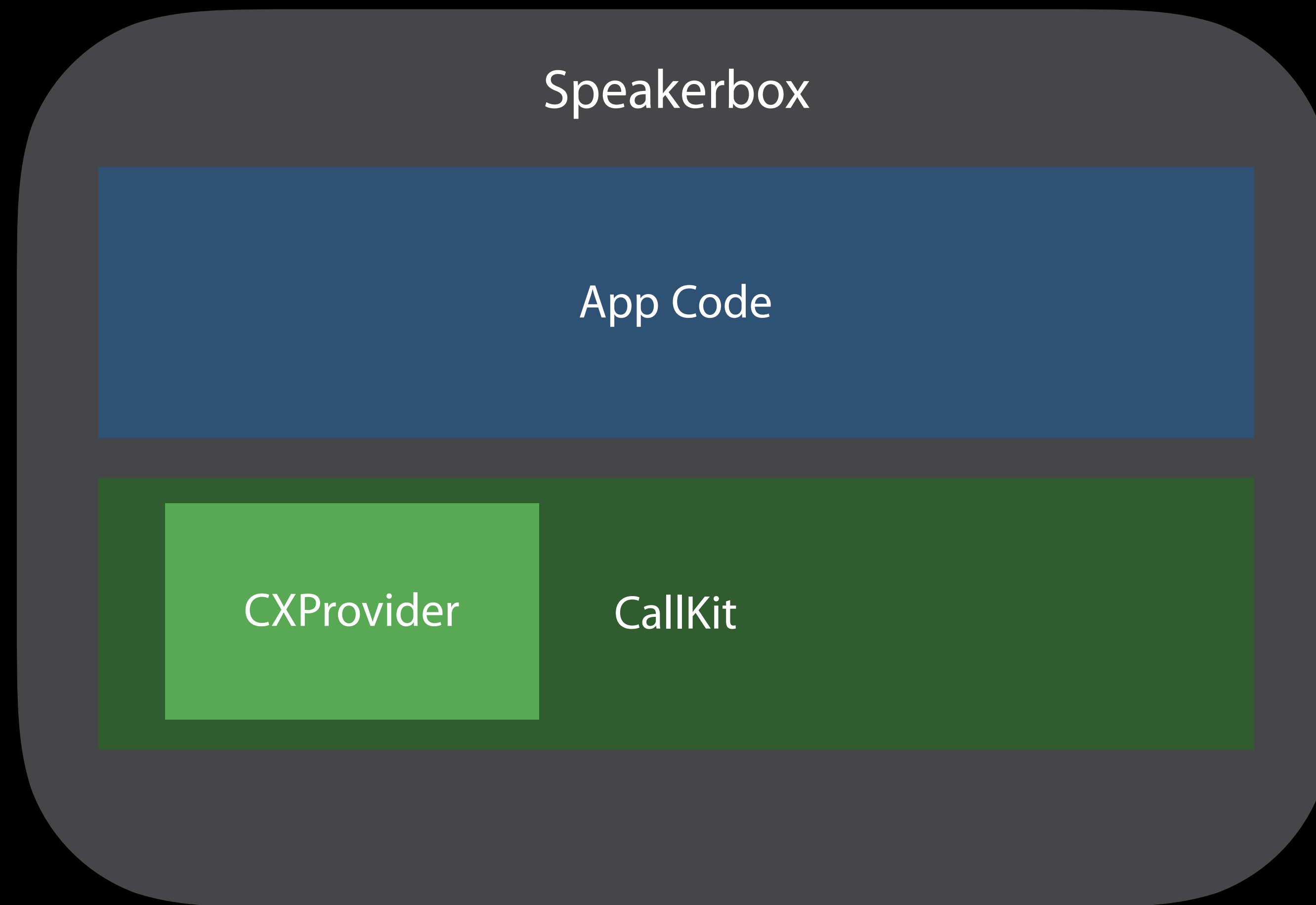
# Architecture



# Architecture



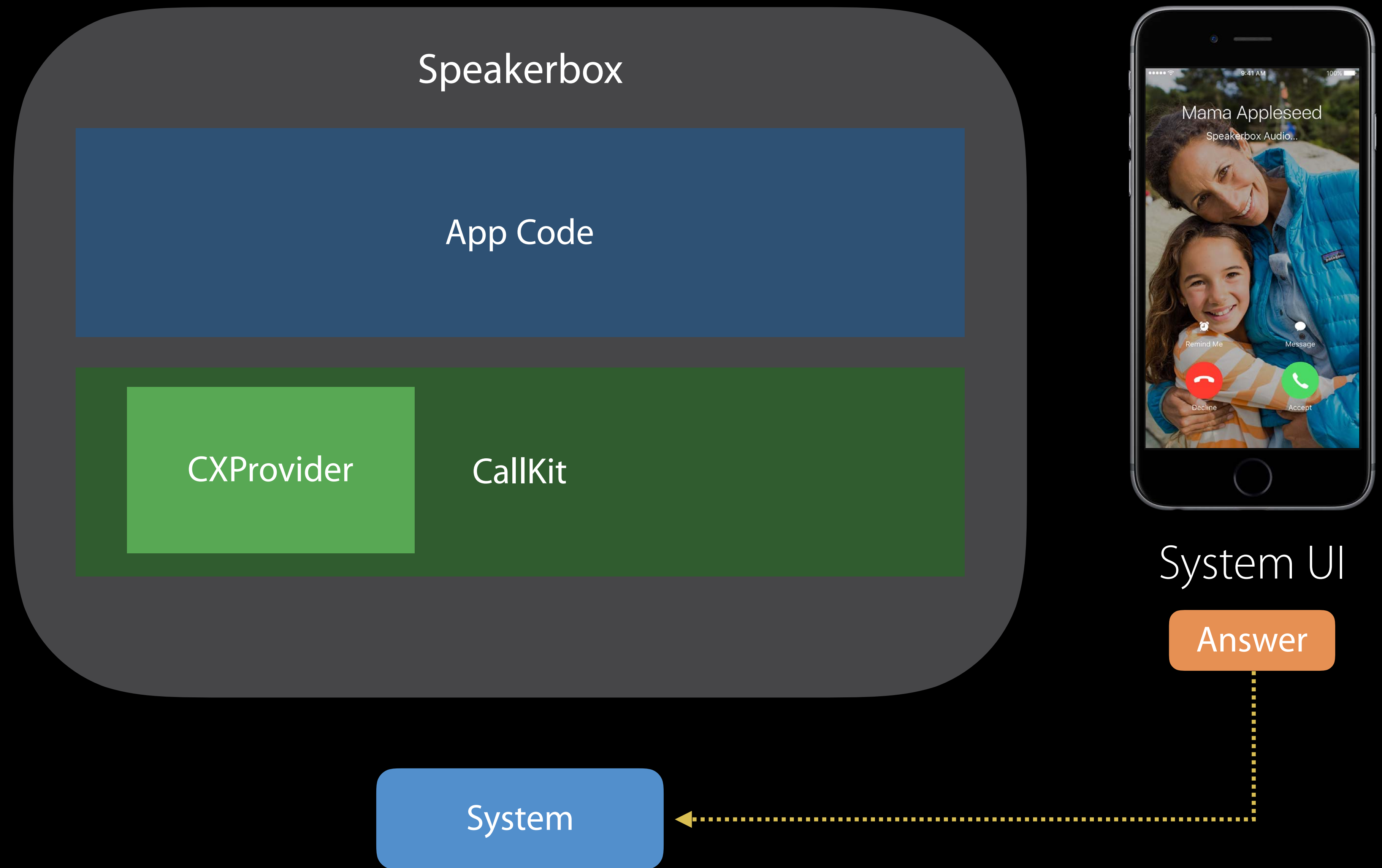
# Architecture



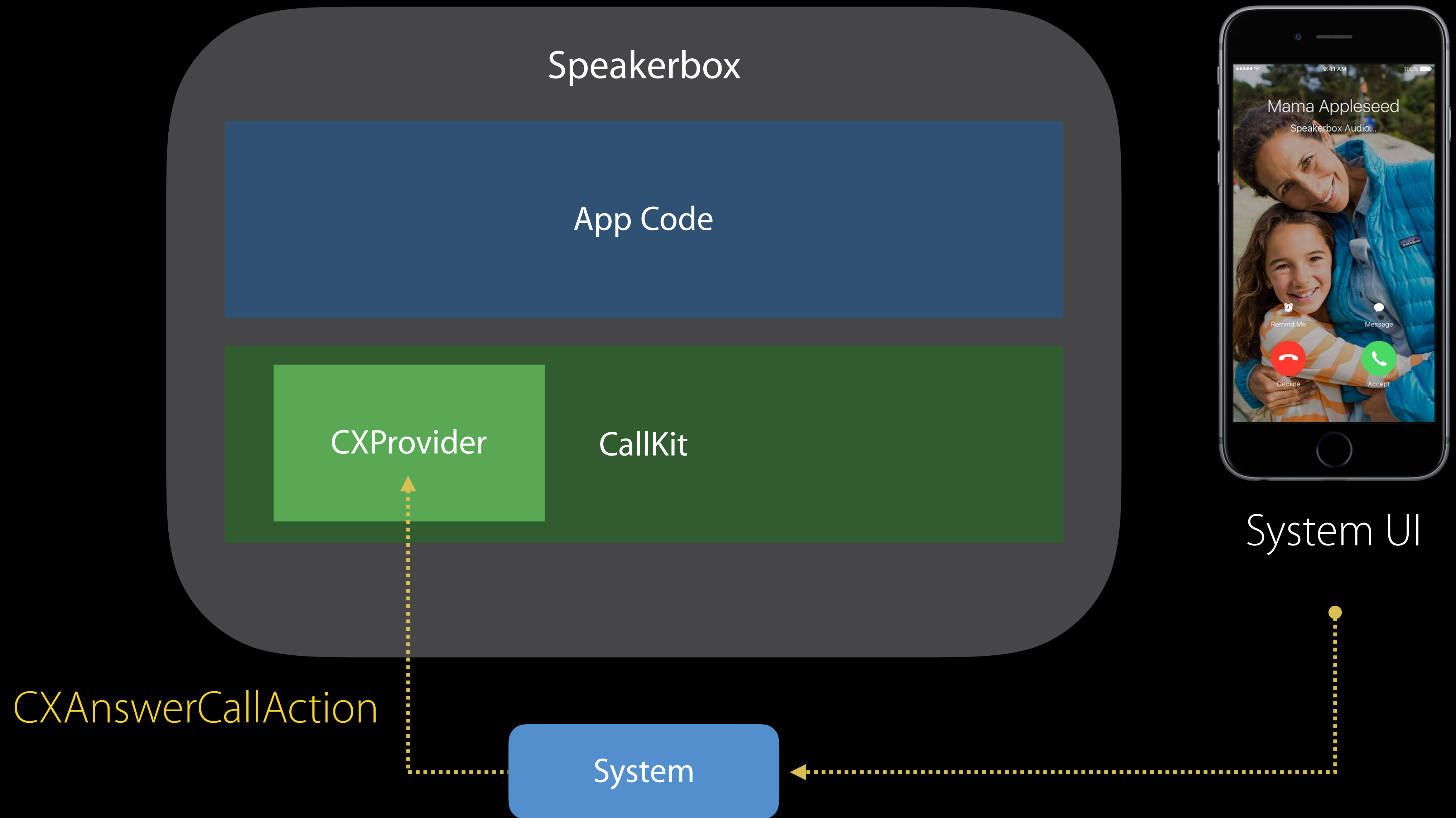
System UI

System

# Architecture

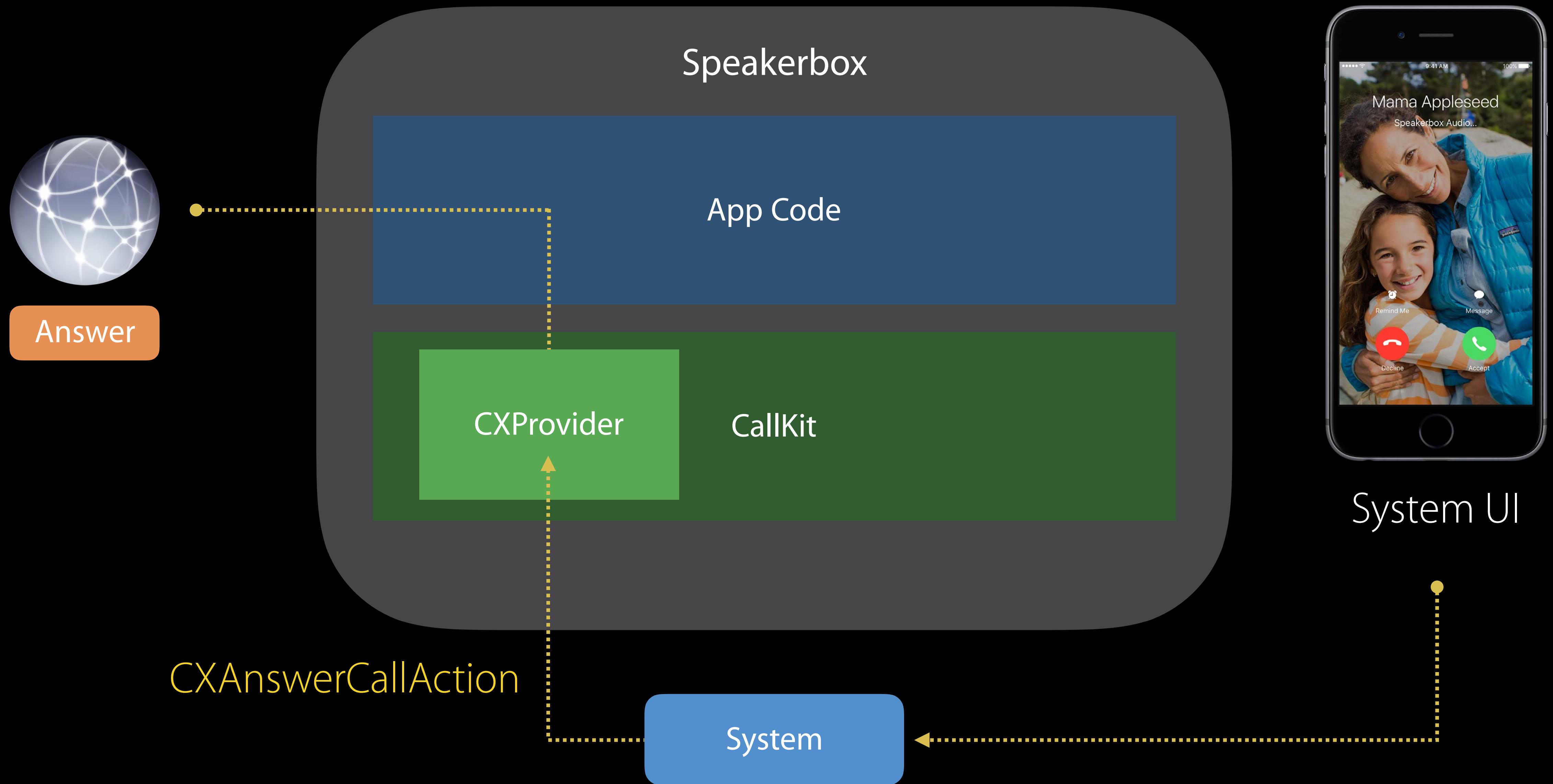


# Architecture

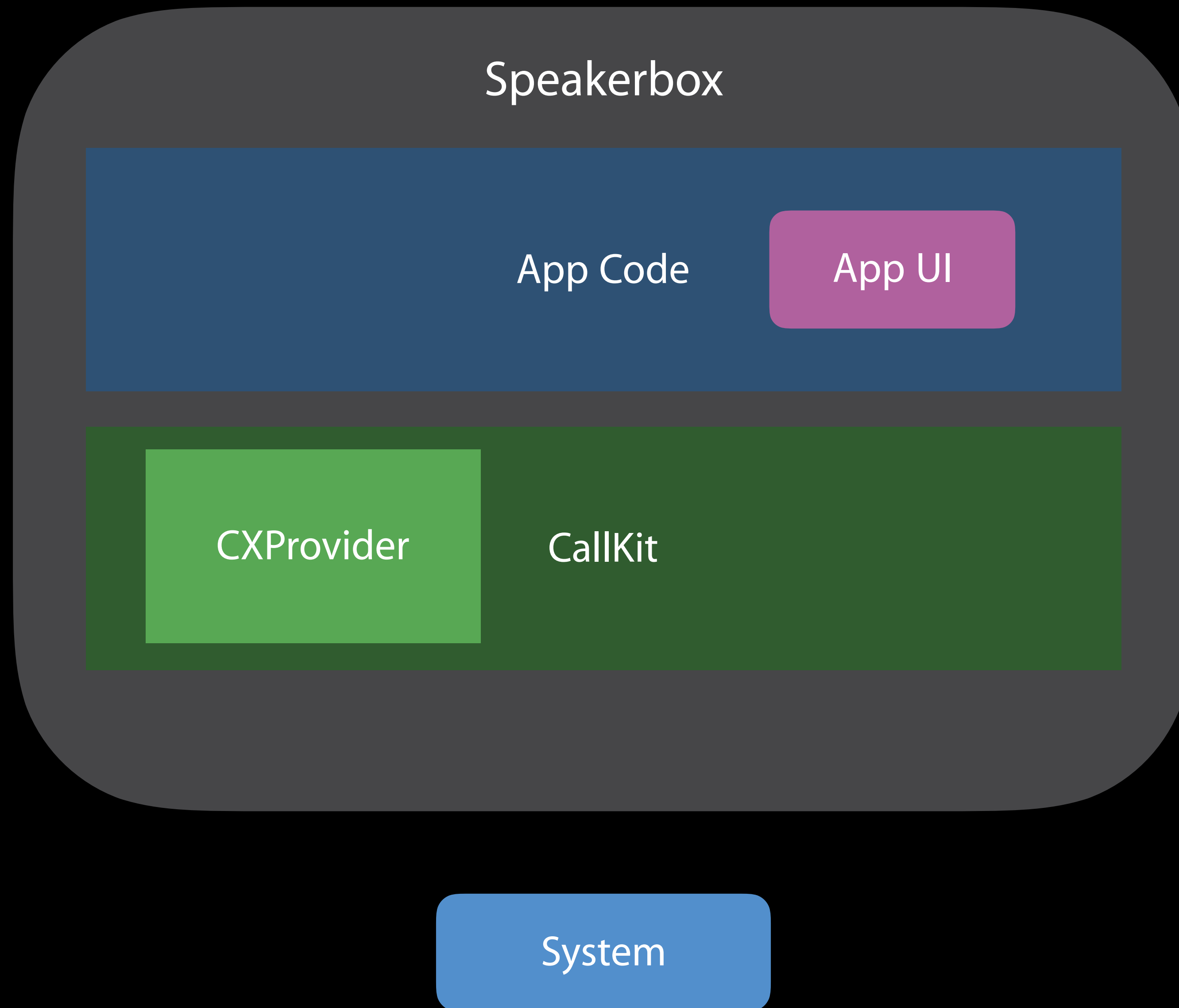




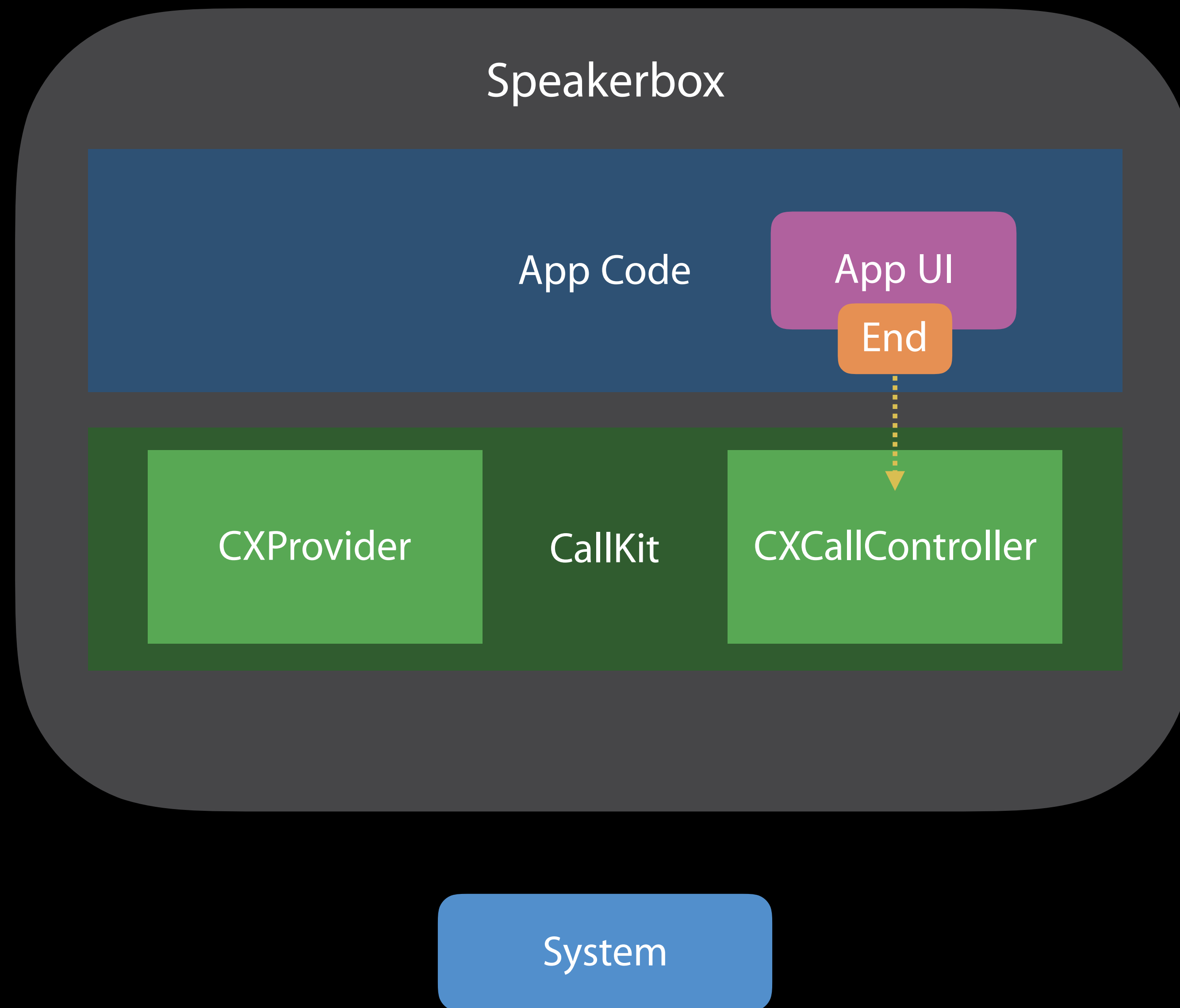
# Architecture



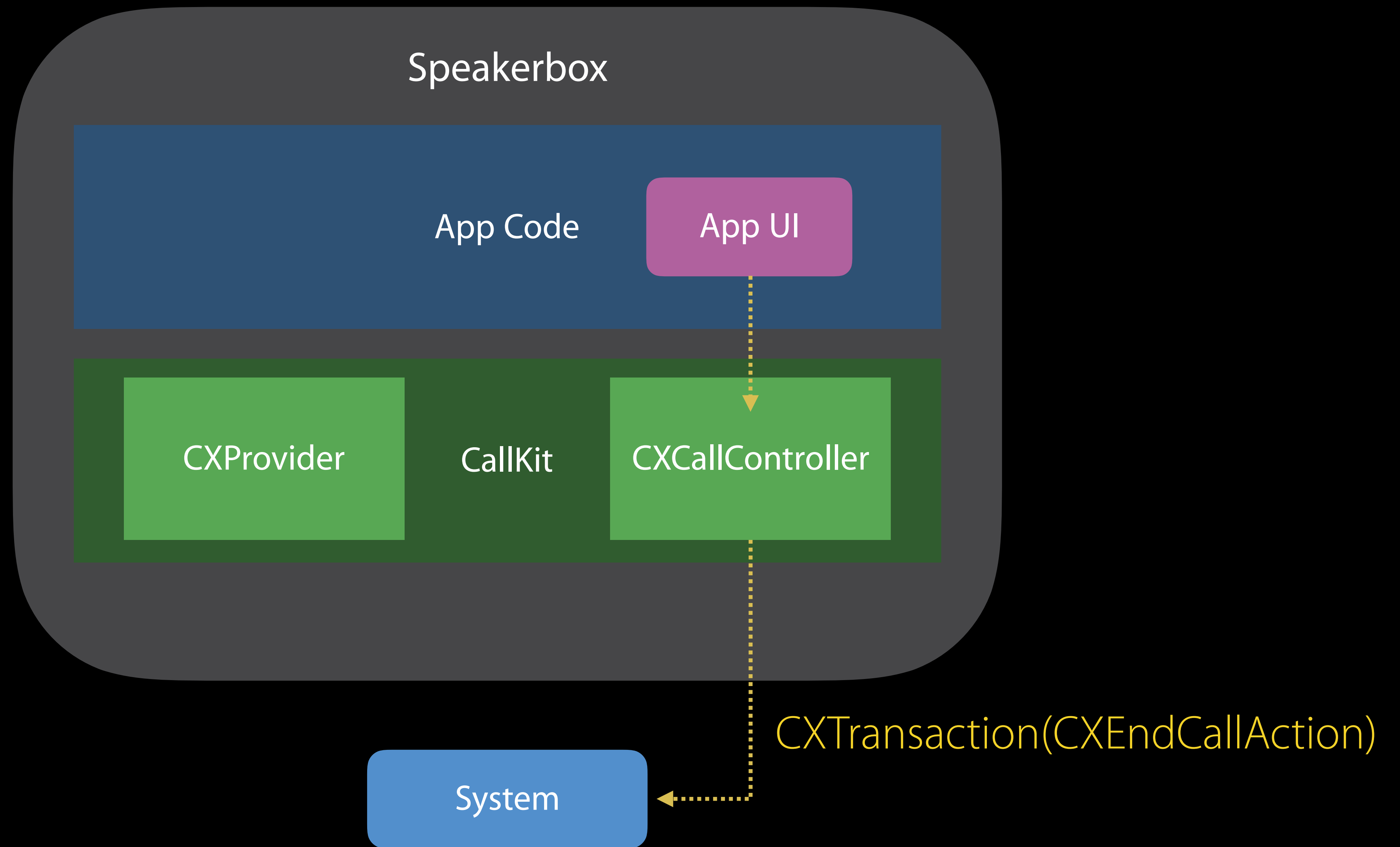
# Architecture



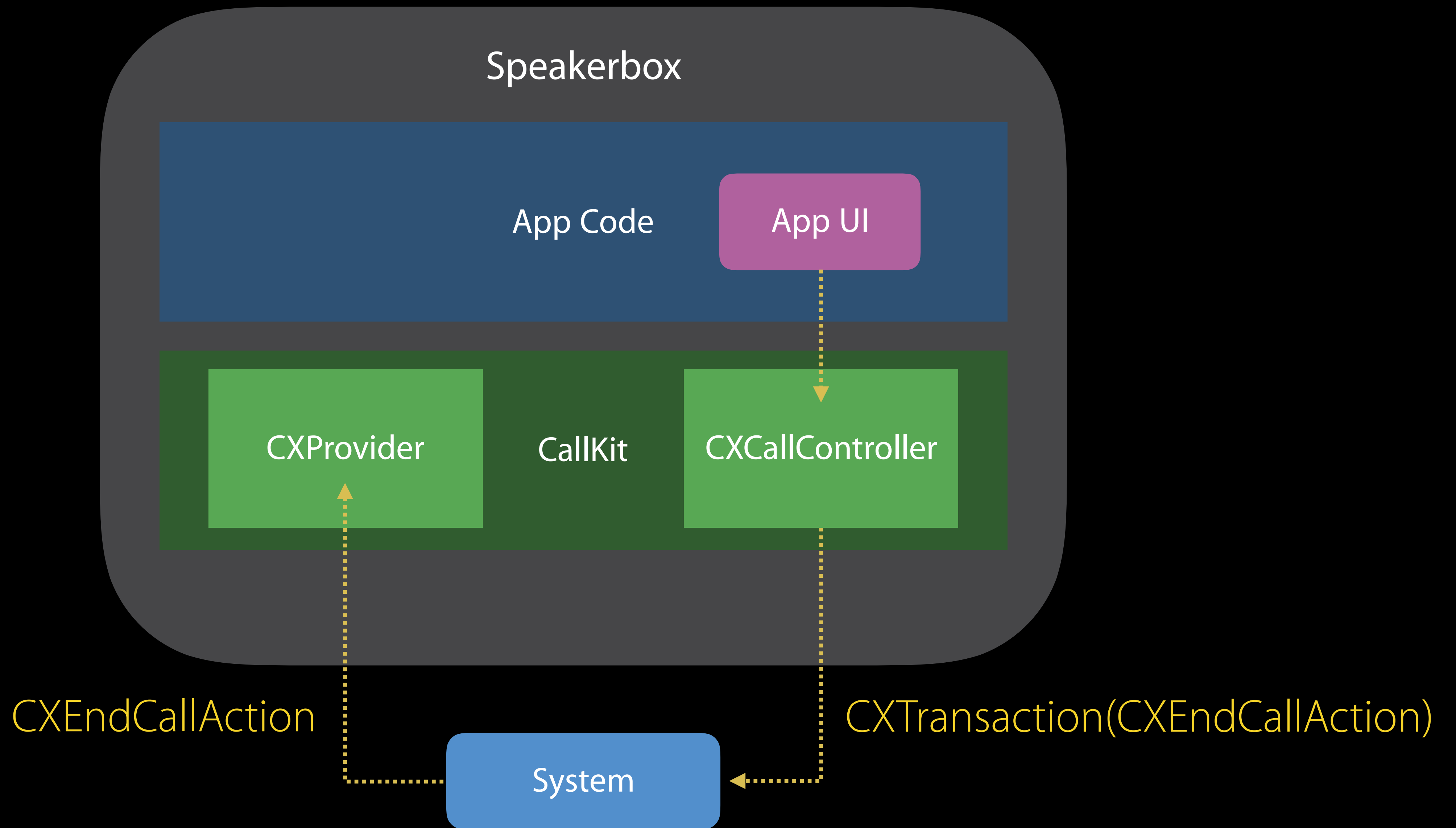
# Architecture



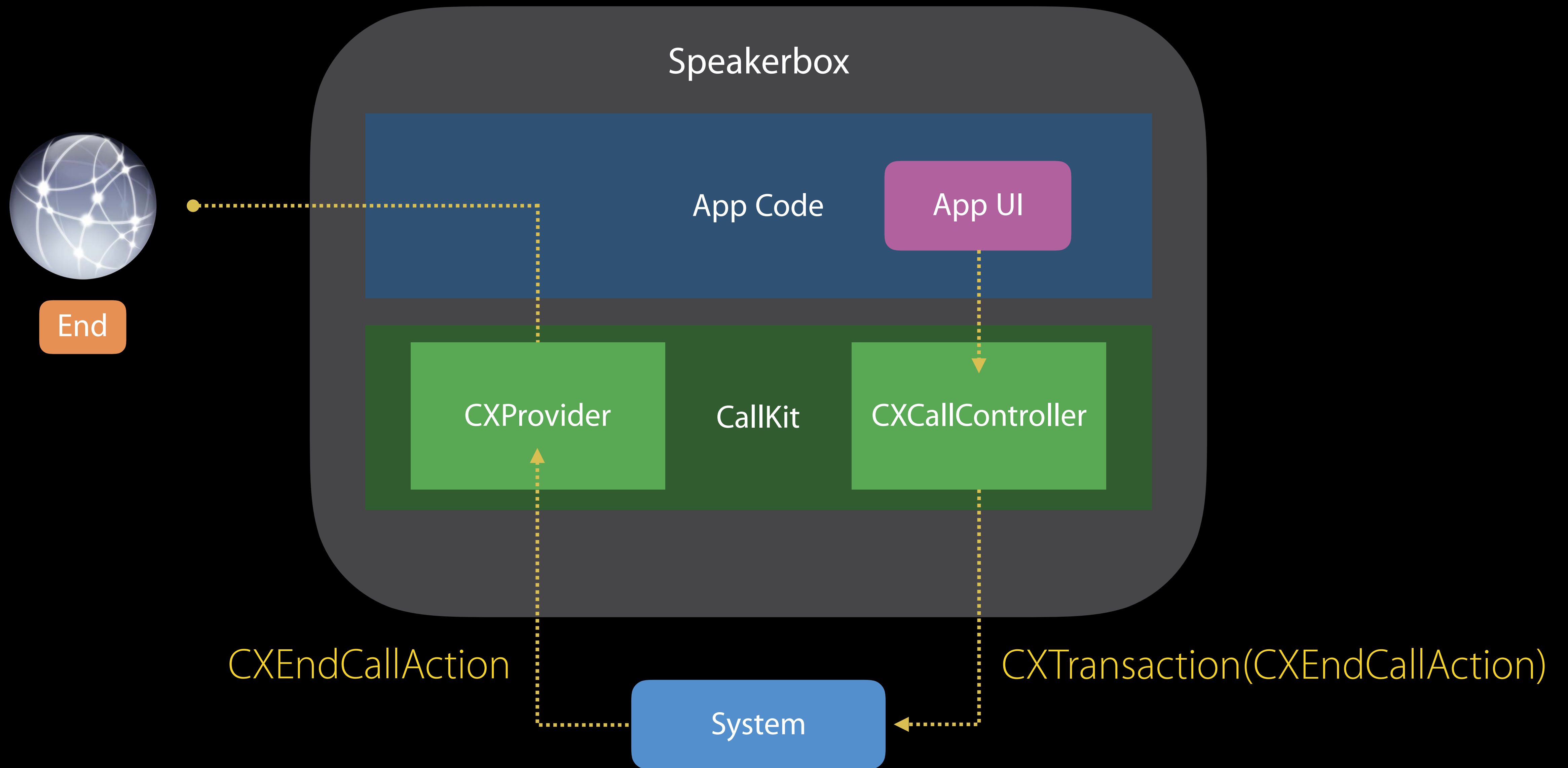
# Architecture



# Architecture



# Architecture



*Demo*

Incoming Call

Stuart Montgomery

# Recap

Incoming call



# Recap

Incoming call

Reported incoming call to system

```
provider.reportNewIncomingCall(with: UUID, update: CXCallUpdate) { error in /* ... */ }
```

# Recap

## Incoming call

Reported incoming call to system

```
provider.reportNewIncomingCall(with: UUID, update: CXCallUpdate) { error in /* ... */ }
```

Performed the CXAnswerCallAction

```
func provider(_ provider: CXProvider, perform action: CXAnswerCallAction) { /* ... */ }
```

# Recap

## Incoming call

Reported incoming call to system

```
provider.reportNewIncomingCall(with: UUID, update: CXCallUpdate) { error in /* ... */ }
```

Performed the CXAnswerCallAction

```
func provider(_ provider: CXProvider, perform action: CXAnswerCallAction) { /* ... */ }
```

Fulfilled the CXAnswerCallAction

```
answerCallAction.fulfill()
```

# Call Actions

CX**Answer**CallAction

CX**Start**CallAction

CX**End**CallAction

CX**SetHeld**CallAction

CX**SetGroup**CallAction

CX**PlayDTMF**CallAction

CX**SetMuted**CallAction

# Managing Multiple Calls



# Managing Multiple Calls

CXTransaction



# Managing Multiple Calls

## CXTransaction

- List of multiple actions



# Managing Multiple Calls

## CXTransaction

- List of multiple actions
- CXEndCallAction and CXAnswerCallAction





# Managing Multiple Calls

## CXTransaction

- List of multiple actions
- CXEndCallAction and CXAnswerCallAction
- Fulfill each action individually



# Managing Multiple Calls

## CXTransaction

- List of multiple actions
- CXEndCallAction and CXAnswerCallAction
- Fulfill each action individually



Outgoing Call

# Outgoing Call

Call controller

# Outgoing Call

Call controller

# Outgoing Call

Call controller

Handle start call intent



Start Call

# Outgoing Call

Call controller

Handle start call intent



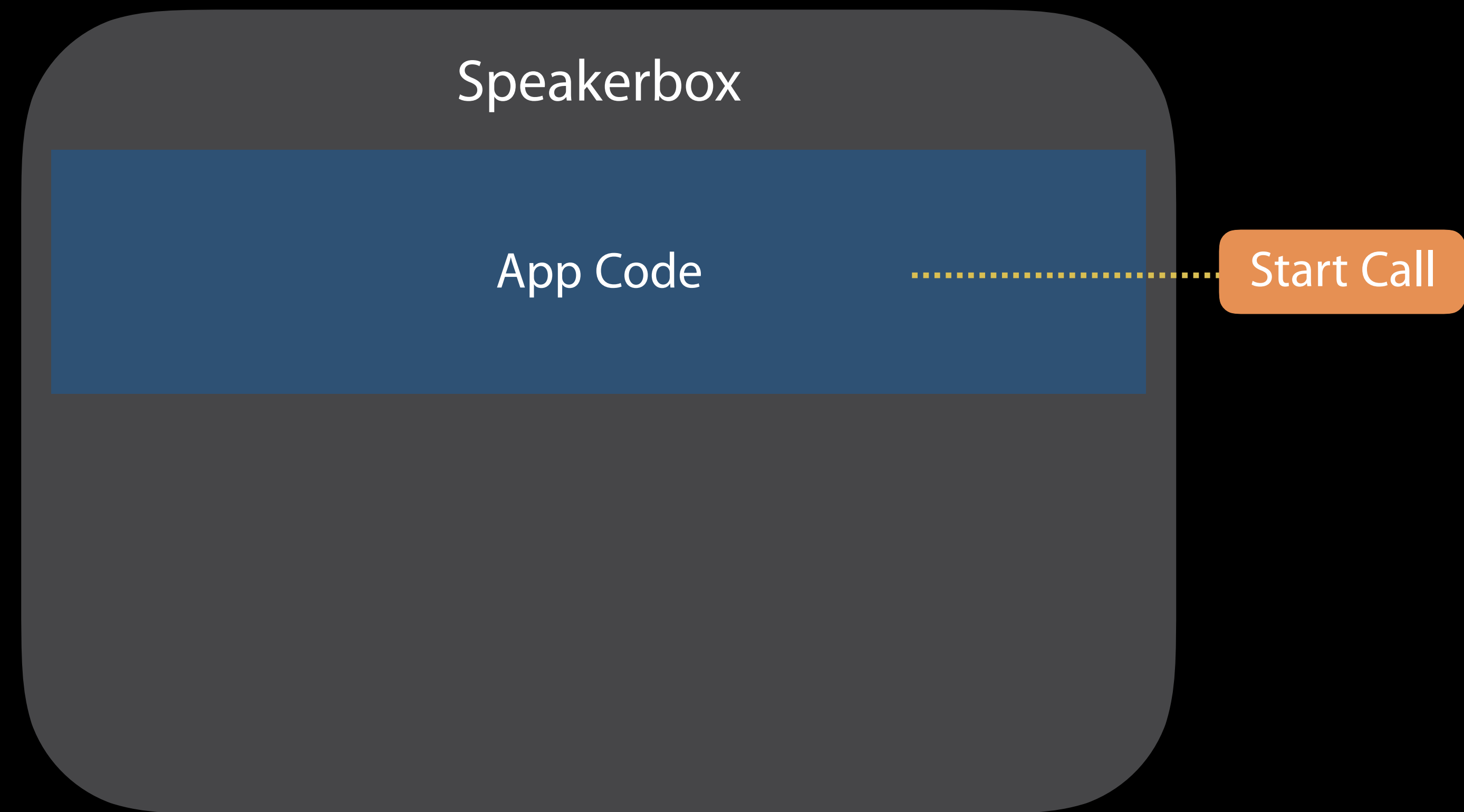
Start Call

# Outgoing Call

## Call controller

Handle start call intent

Create start call action





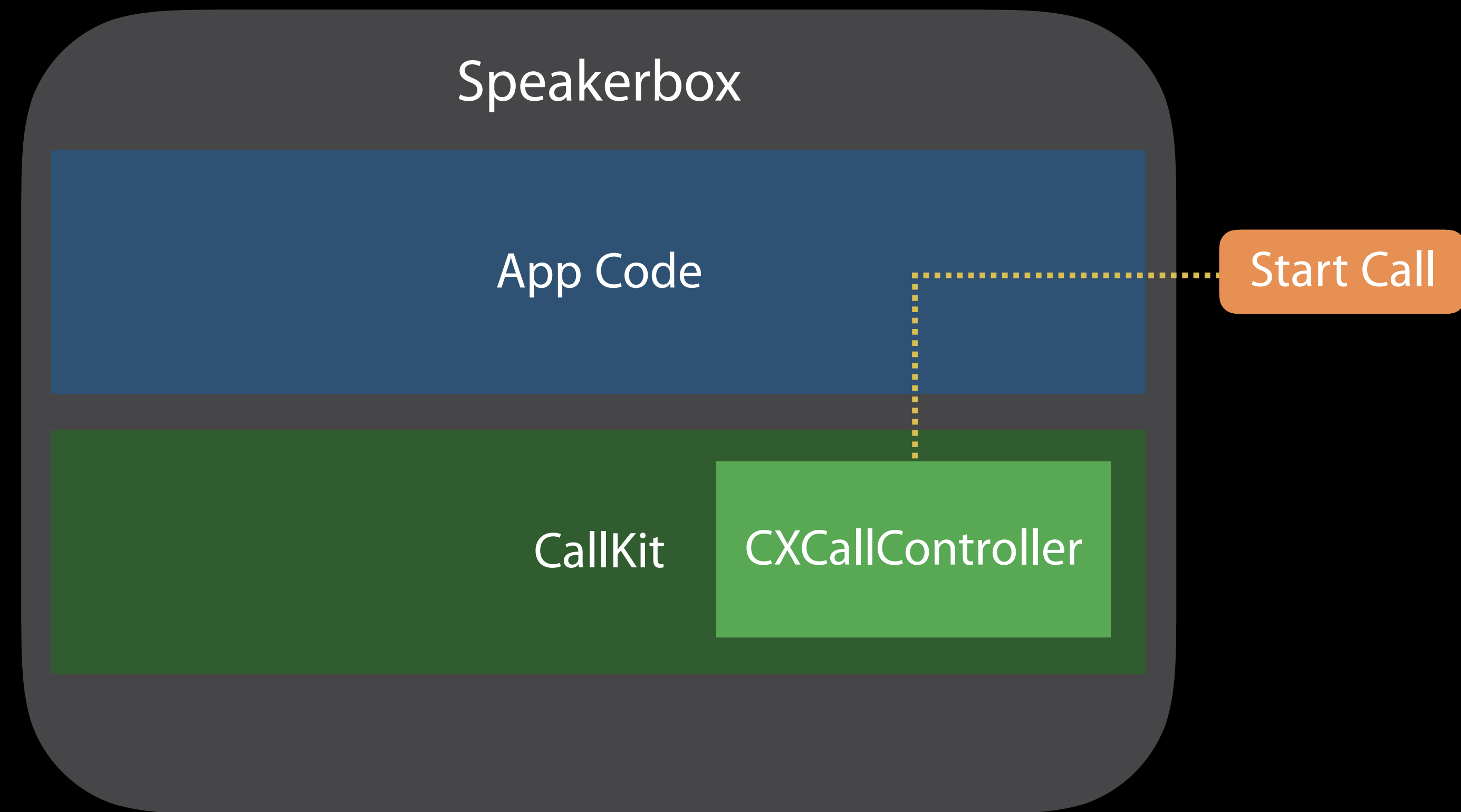
# Outgoing Call

## Call controller

Handle start call intent

Create start call action

Request start call action



# Outgoing Call

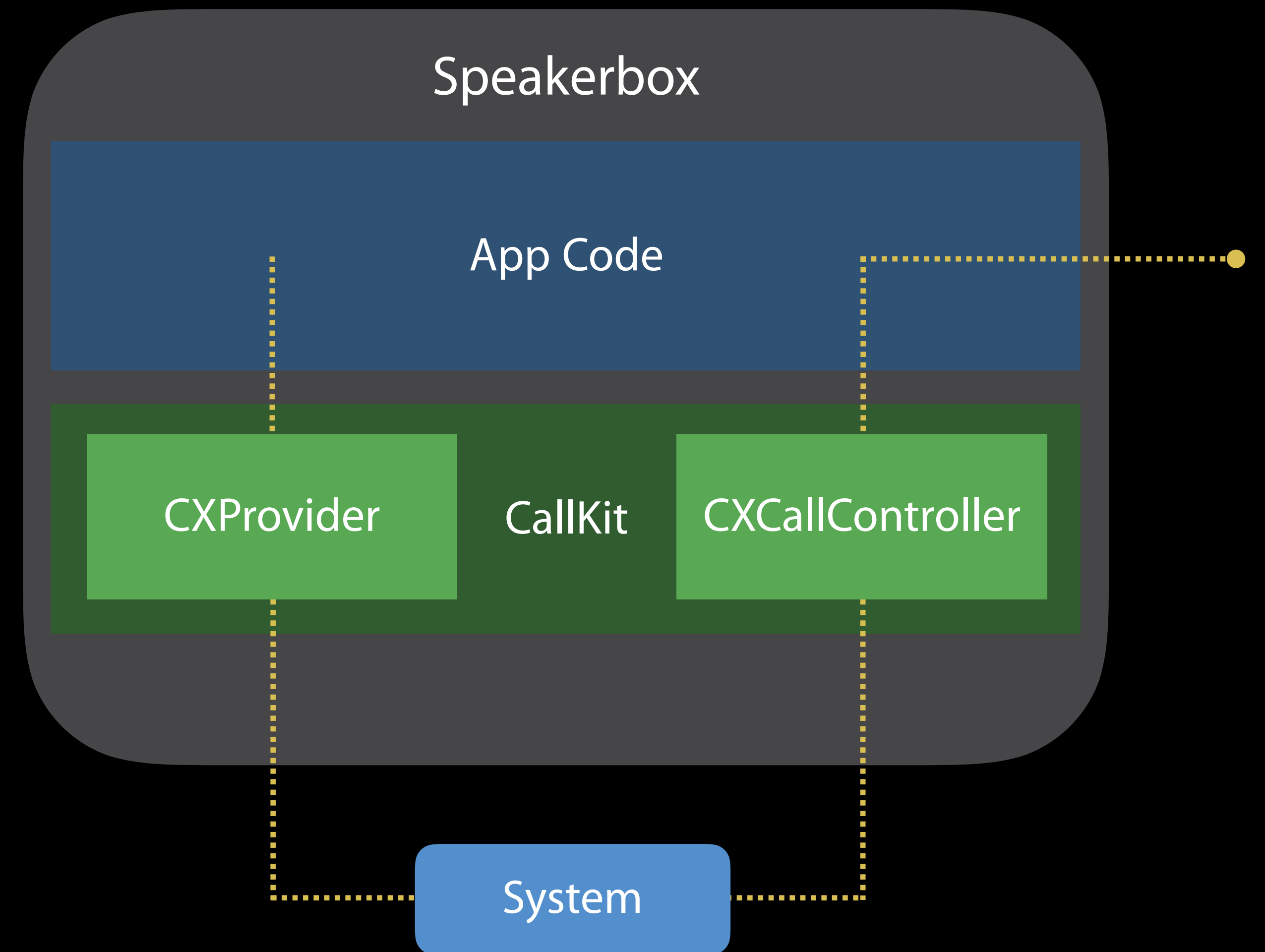
## Call controller

Handle start call intent

Create start call action

Request start call action

Receive start call action



# Outgoing Call

## Call controller

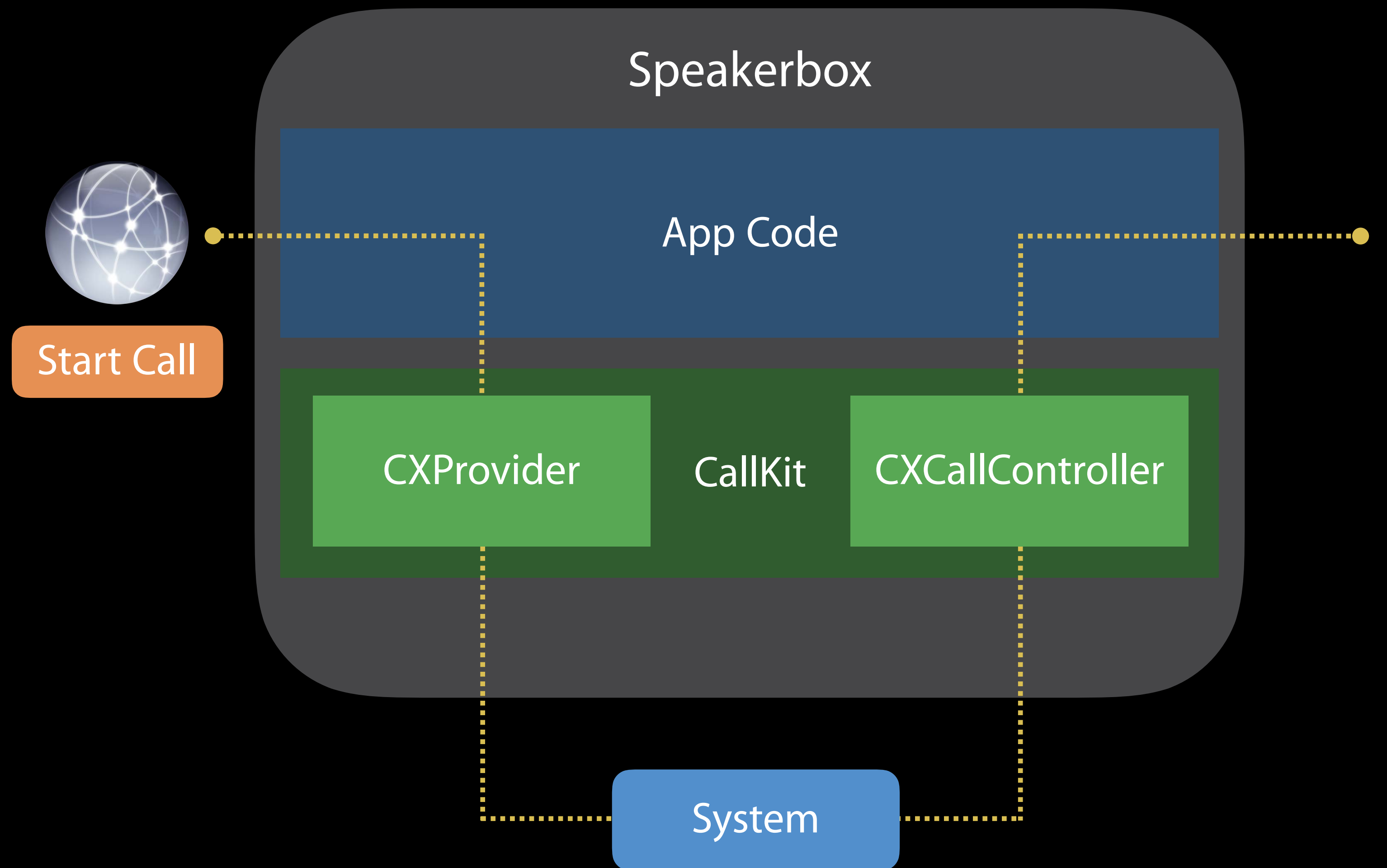
Handle start call intent

Create start call action

Request start call action

Receive start call action

Execute start call action



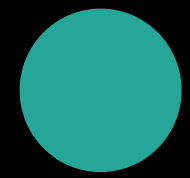
# Outgoing Call

Lifecycle

# Outgoing Call

Lifecycle

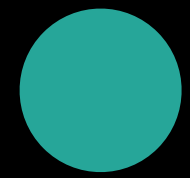
# Outgoing Call Lifecycle



Starting



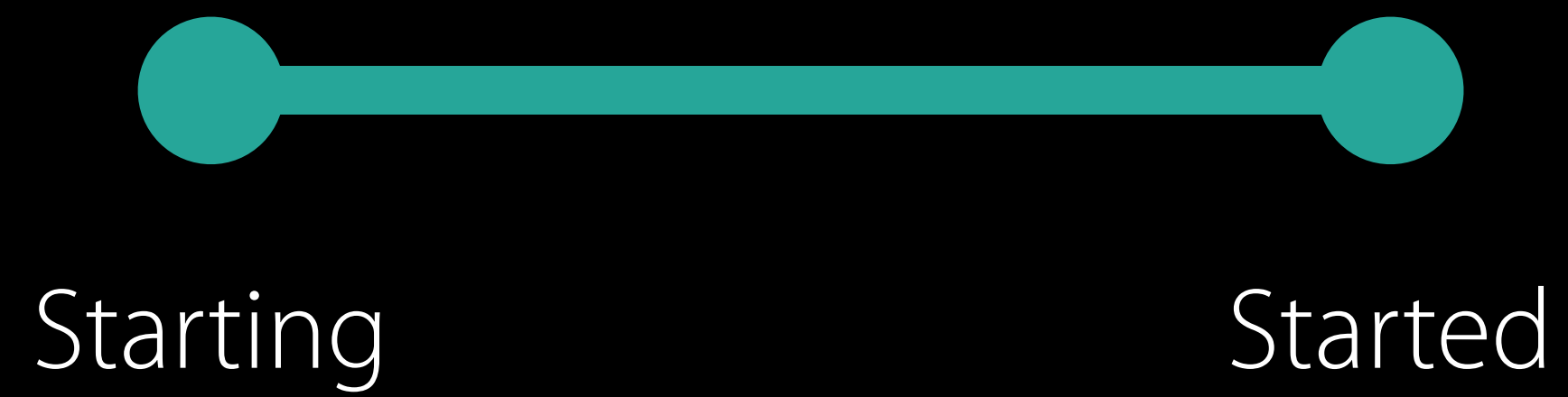
# Outgoing Call Lifecycle



Starting

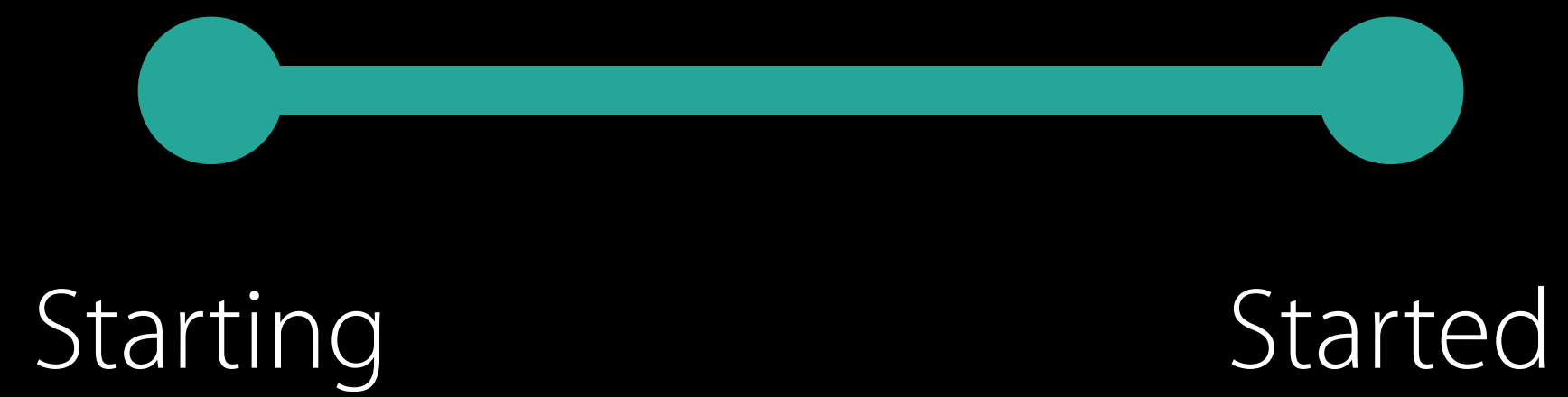


# Outgoing Call Lifecycle

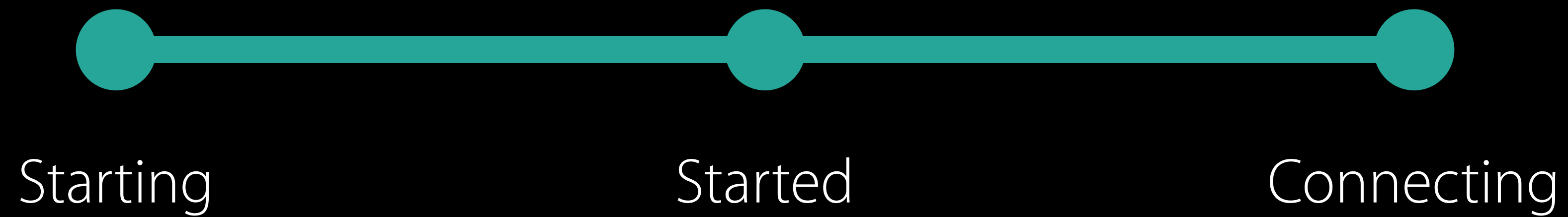




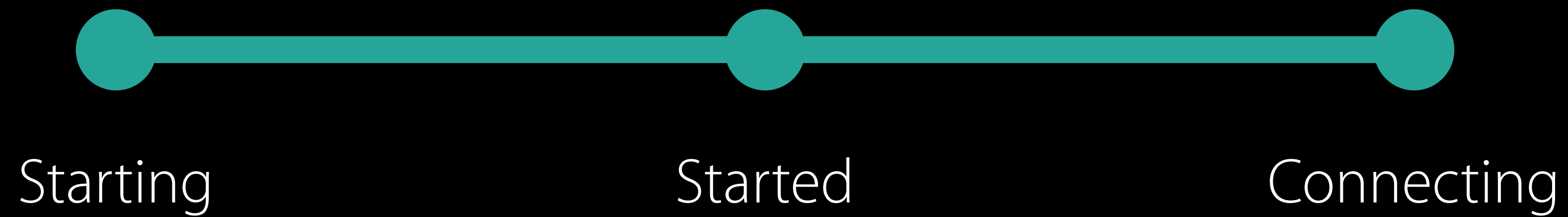
# Outgoing Call Lifecycle



# Outgoing Call Lifecycle



# Outgoing Call Lifecycle



# Outgoing Call Lifecycle



*Demo*

Outgoing Call

Stuart Montgomery

# Recap

Outgoing call

# Recap

## Outgoing call

Request start call action with Call Controller

```
callController.request(startCallTransaction) { error in /* ... */ }
```

# Recap

## Outgoing call

Request start call action with Call Controller

```
callController.request(startCallTransaction) { error in /* ... */ }
```

Fulfill start call action with Provider

```
startCallAction.fulfill()
```



# Recap

## Outgoing call

Request start call action with Call Controller

```
callController.request(startCallTransaction) { error in /* ... */ }
```

Fulfill start call action with Provider

```
startCallAction.fulfill()
```

Send call state updates

```
provider.reportOutgoingCall(with: call.uuid, startedConnectingAt: call.connectingDate)  
provider.reportOutgoingCall(with: call.uuid, connectedAt: call.connectDate)
```

# API Details

# API Details

Authorization

Configuration

Action Errors

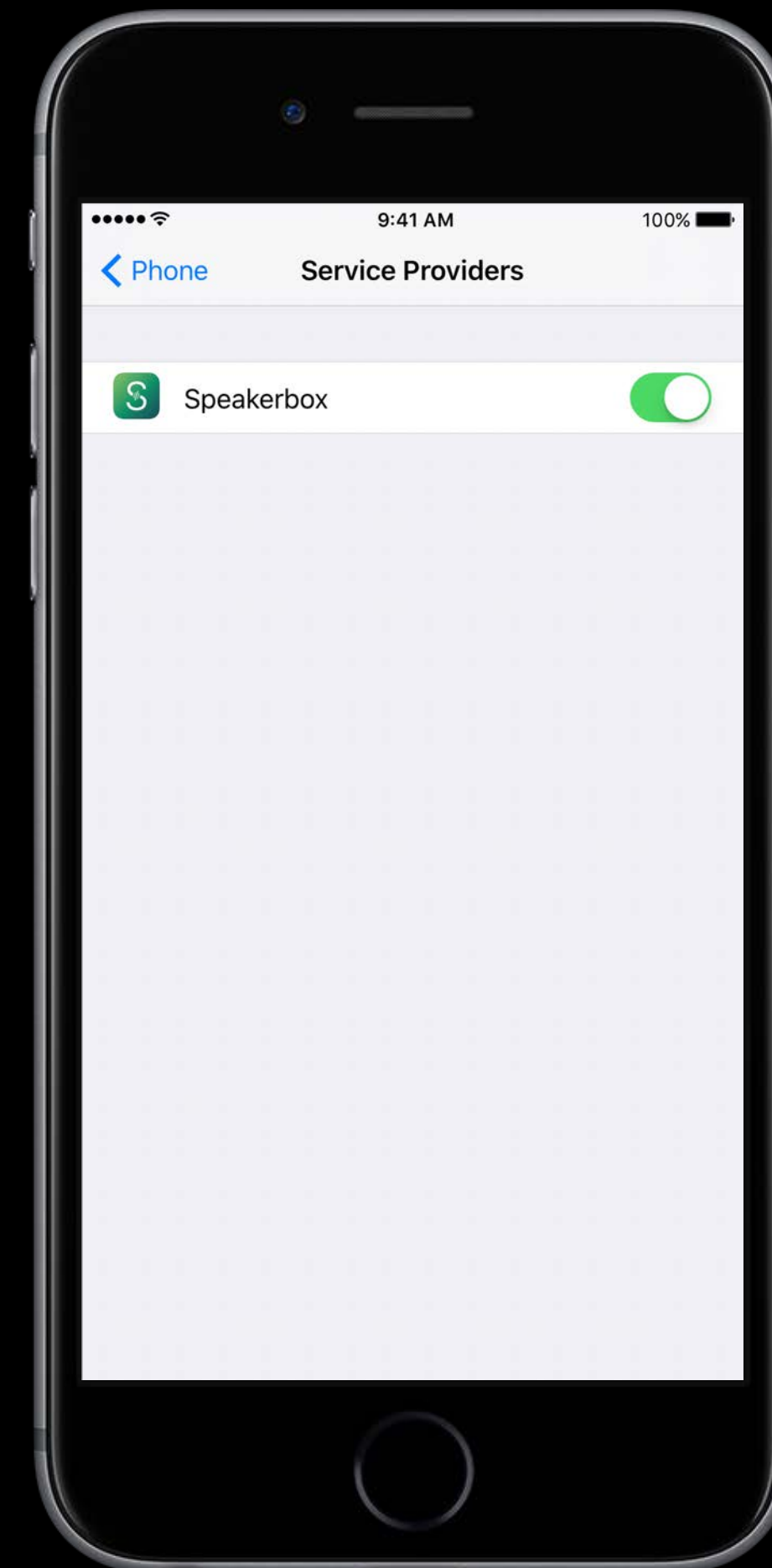
System Restrictions

Audio

# Provider Authorization

# Provider Authorization

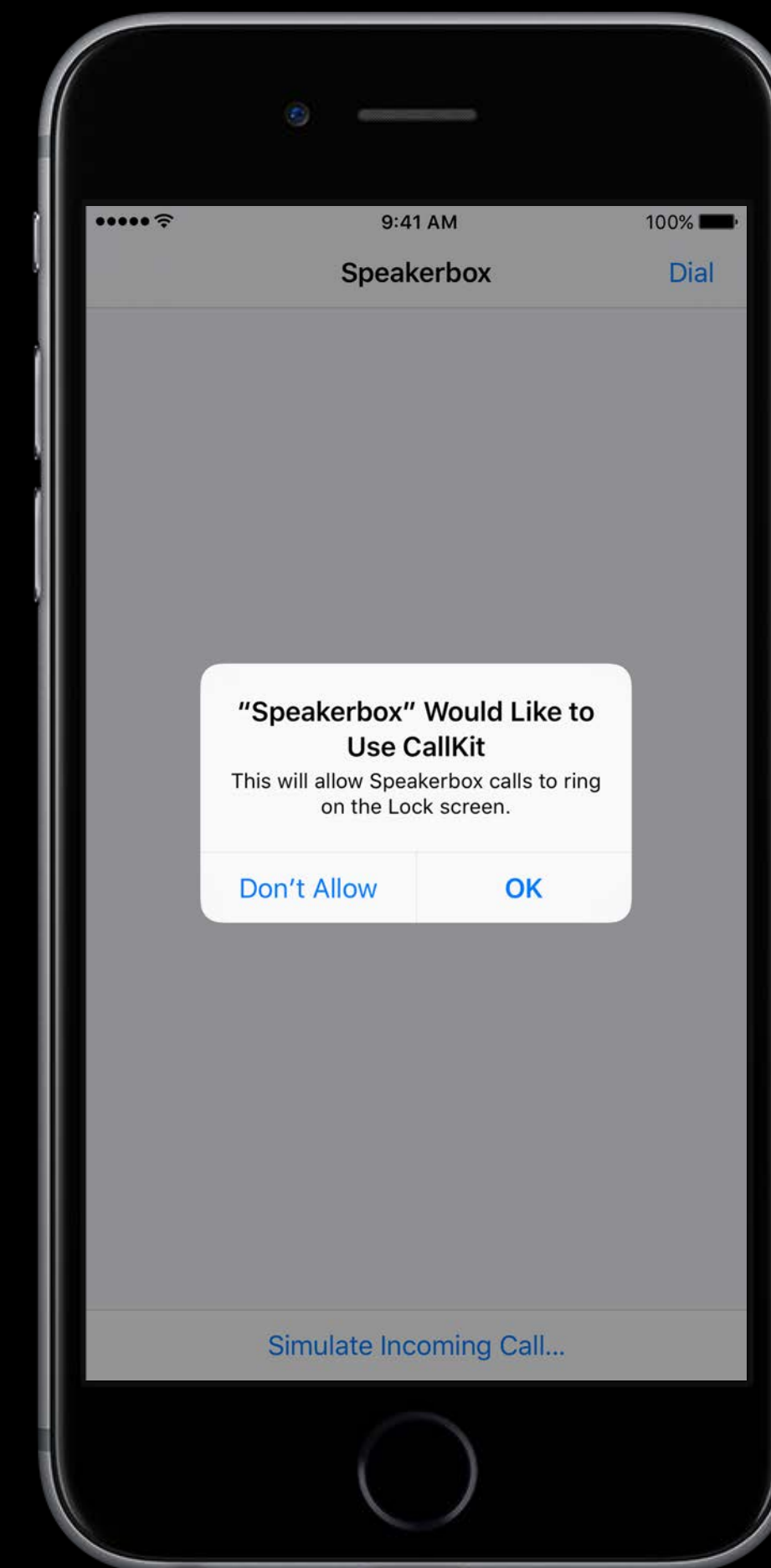
Check authorization status



# Provider Authorization

Check authorization status

Request authorization

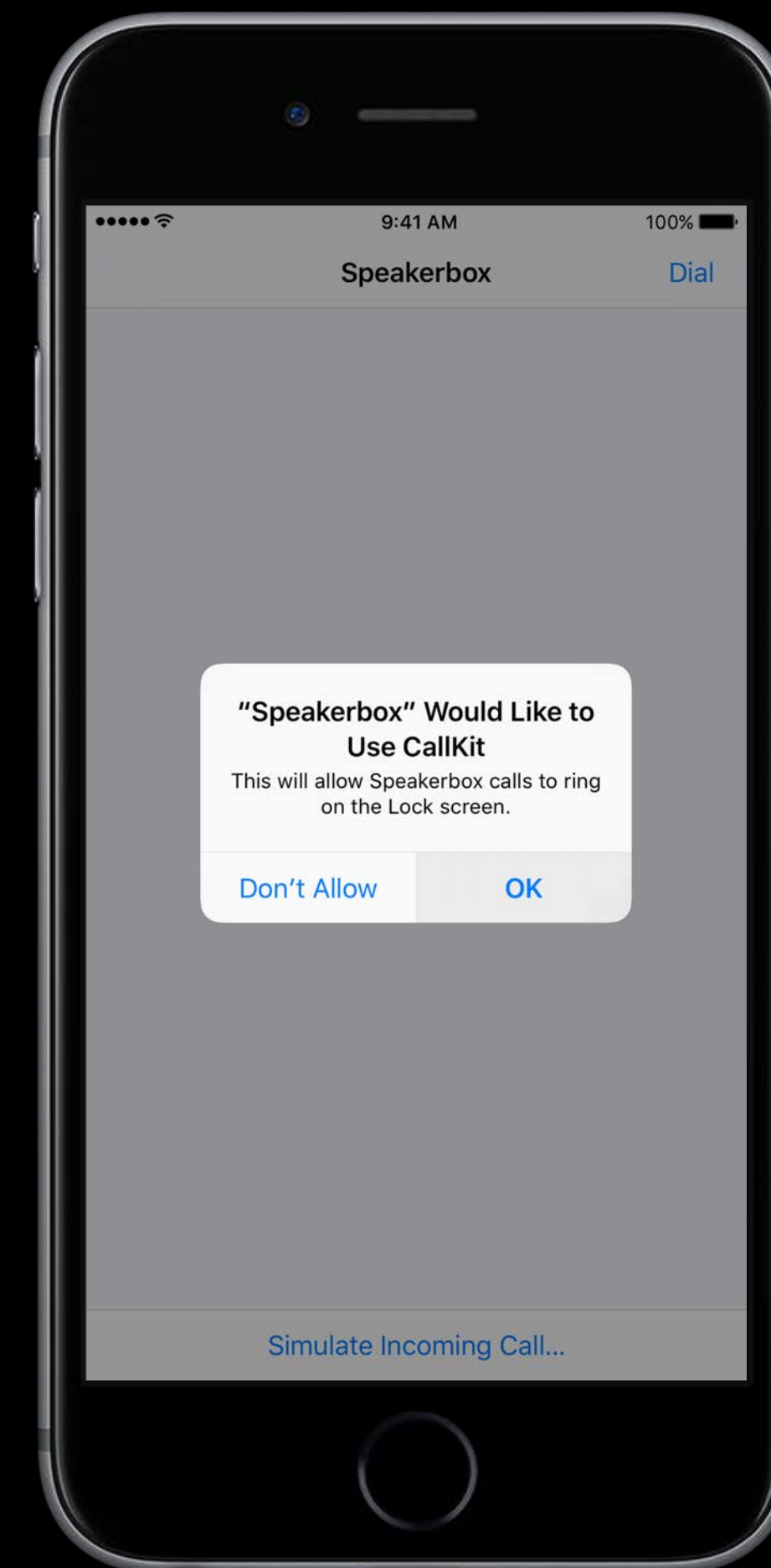


# Provider Authorization

Check authorization status

Request authorization

Observe authorization changes



# Provider Configuration



# Provider Configuration

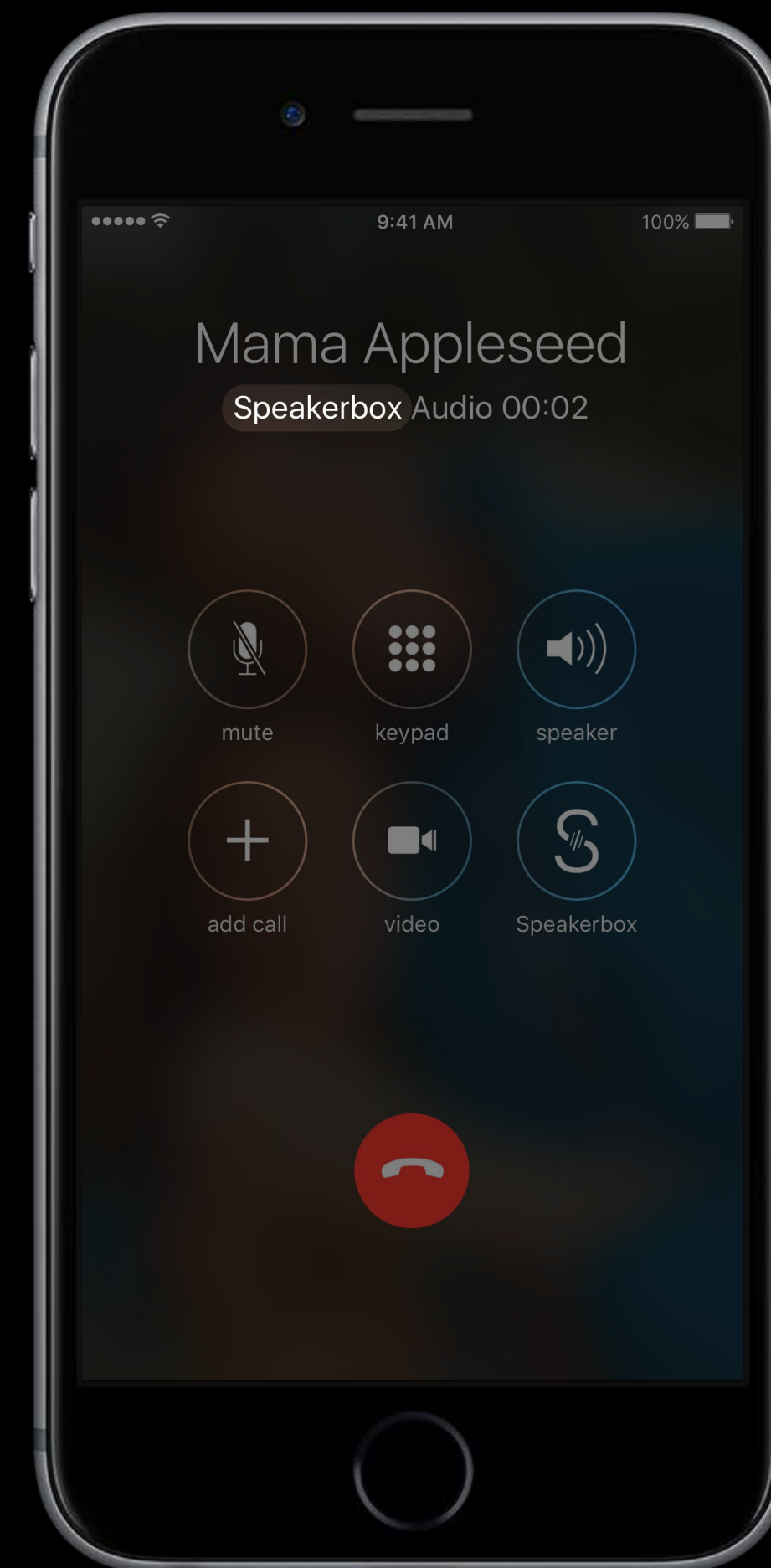
Customized experience



# Provider Configuration

Customized experience

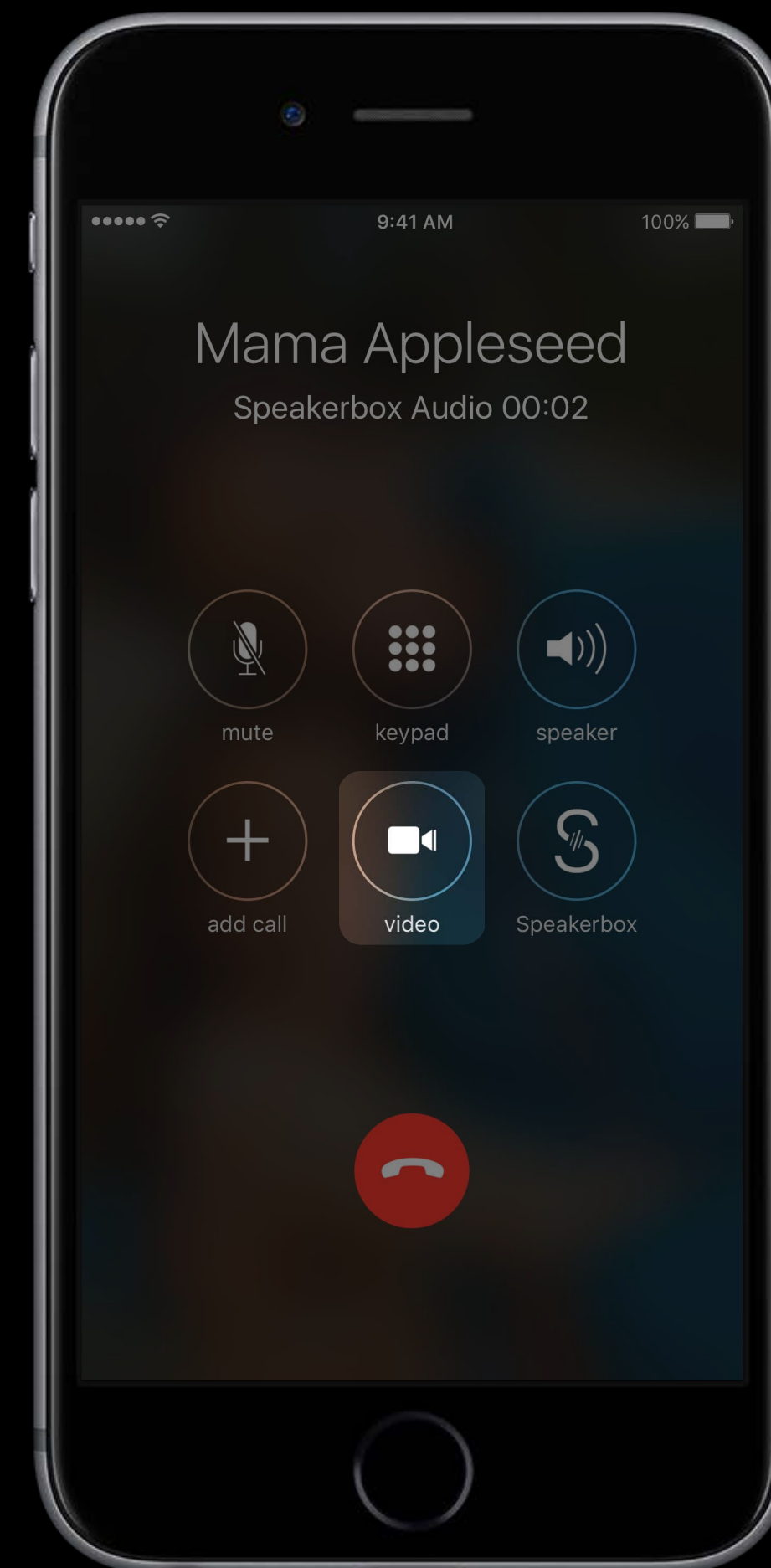
- Localized name for display



# Provider Configuration

Customized experience

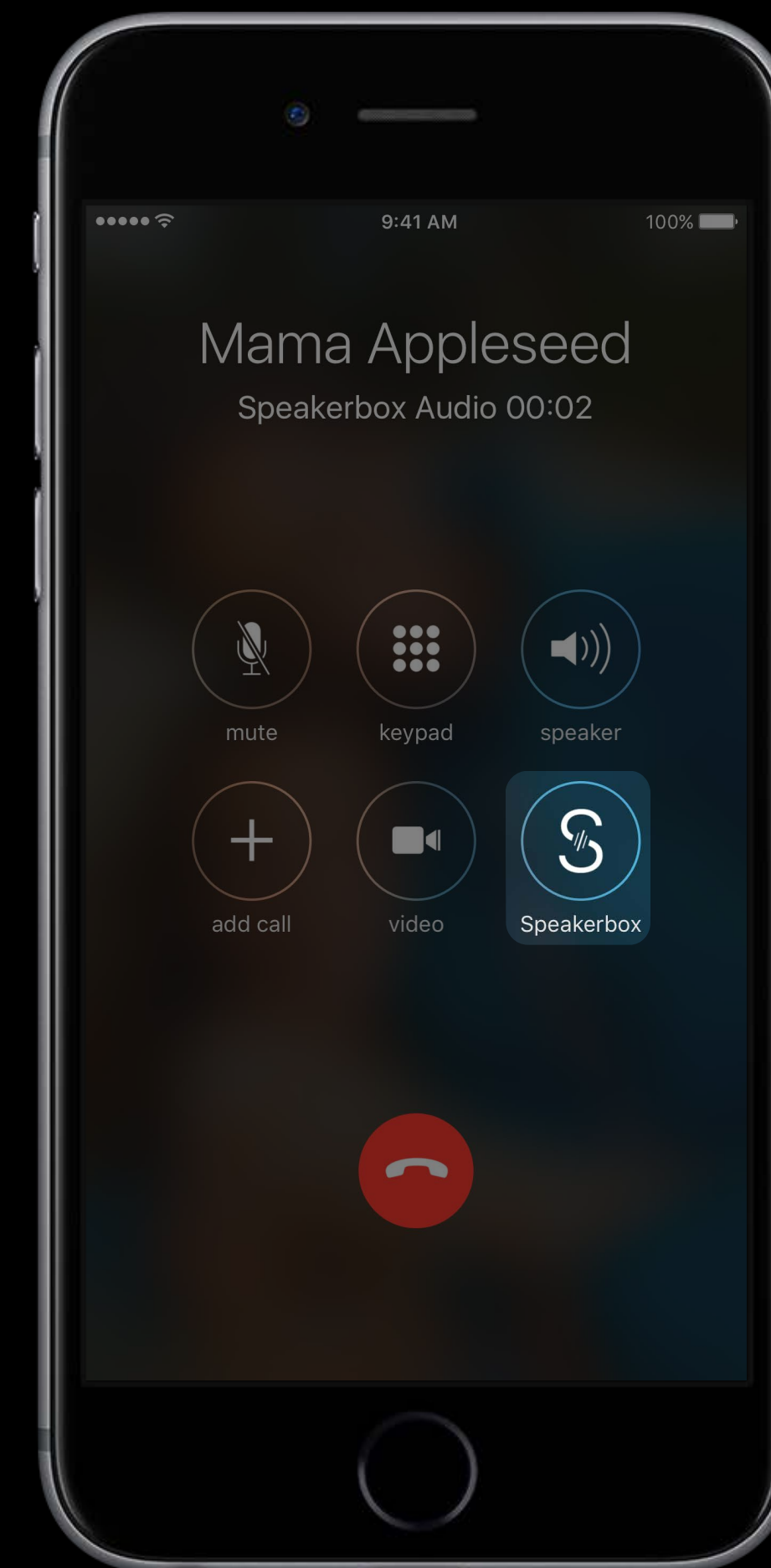
- Localized name for display
- Video support



# Provider Configuration

## Customized experience

- Localized name for display
- Video support
- Masked image icon



# Action Errors

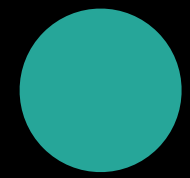
Failing actions appropriately

# Action Errors

Failing actions appropriately

# Action Errors

Failing actions appropriately

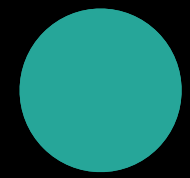


Starting

Receive action

# Action Errors

Failing actions appropriately



Starting

Receive action



# Action Errors

Failing actions appropriately



Starting

Started

Receive action

# Action Errors

Failing actions appropriately



Starting

Started

Receive action

# Action Errors

Failing actions appropriately



Starting

Started

Receive action

Fail action

# Action Errors

Failing actions appropriately



Starting

Started

Receive action

Fail action

# Action Errors

Failing actions appropriately



Receive action

Fail action



# Action Errors

Timeouts

# Action Errors

Timeouts

# Action Errors

Timeouts

Enforce app responsiveness



# Action Errors

## Timeouts

Enforce app responsiveness

Respond to actions in a timely manner

# Action Errors

## Timeouts

Enforce app responsiveness

Respond to actions in a timely manner

React to action timeouts

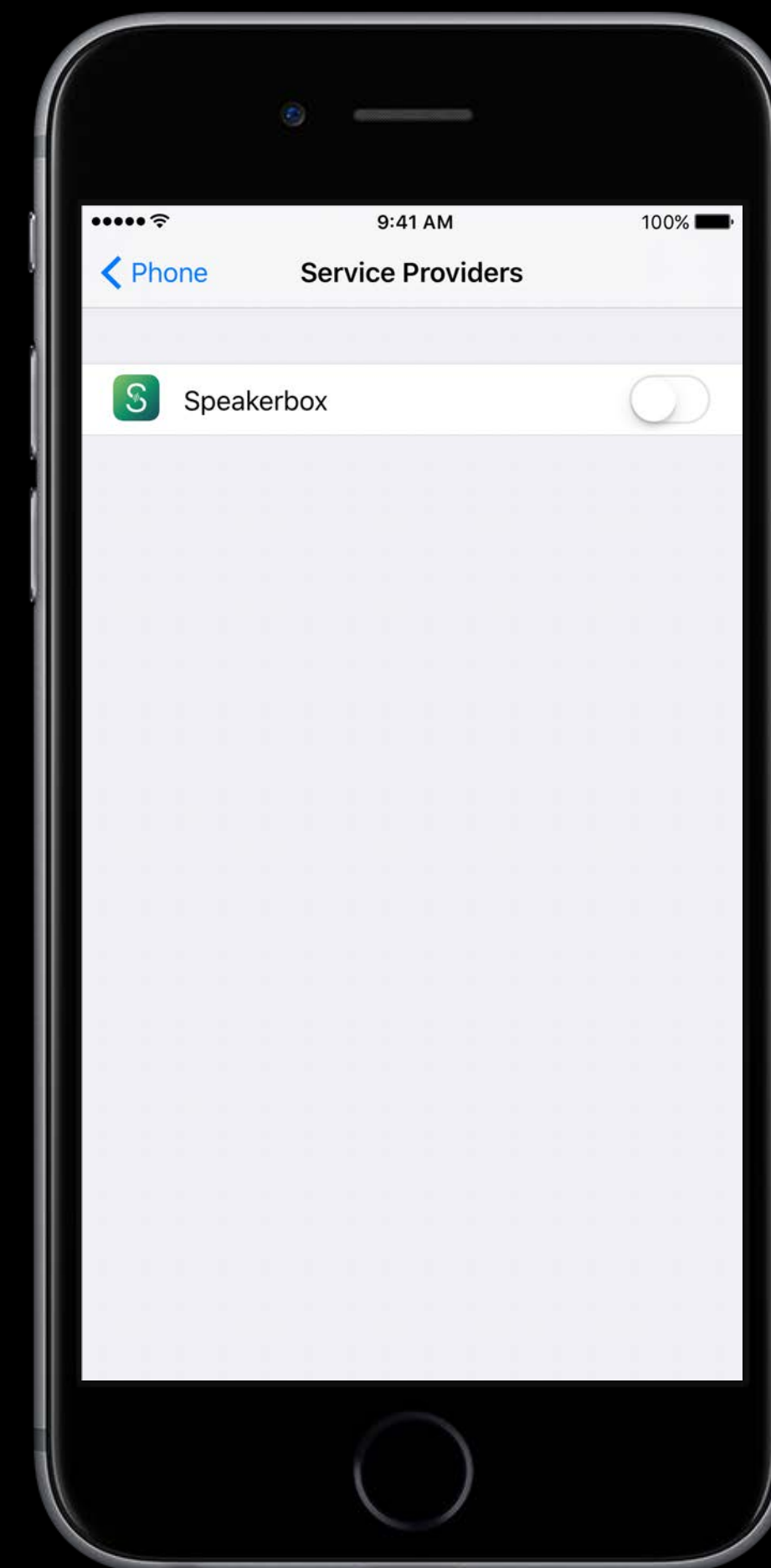
# System Restrictions

Incoming call

# System Restrictions

Incoming call

Not authorized

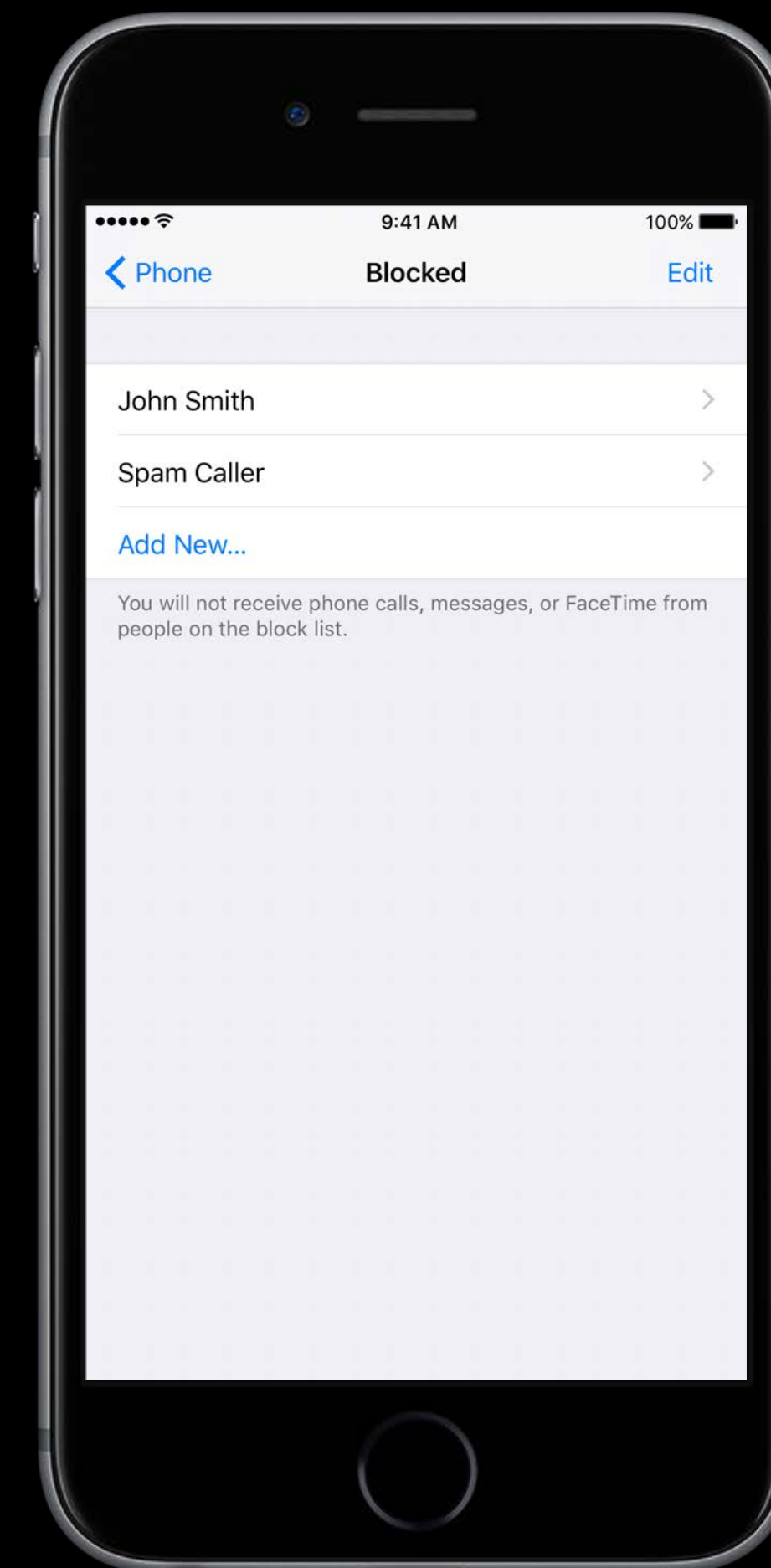


# System Restrictions

Incoming call

Not authorized

Block list



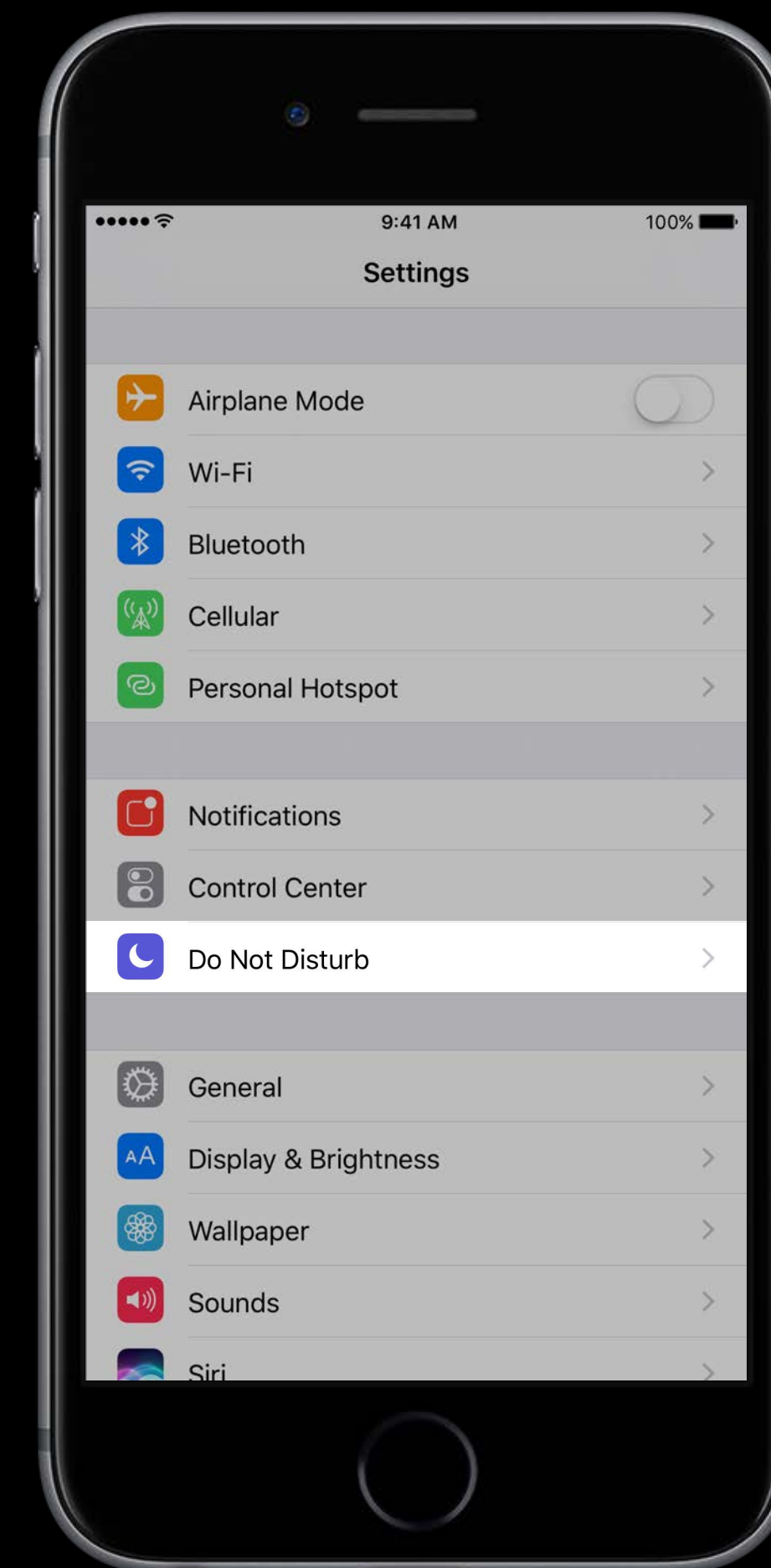
# System Restrictions

Incoming call

Not authorized

Block list

Do not disturb



# System Restrictions

## Incoming call

Not authorized

Block list

Do not disturb

- React to completion handler

```
provider.reportNewIncomingCall(with: UUID(), update: CXCallUpdate()) { error in
    if let incomingCallError = error,
        incomingCallErrorCode = CXErrorCodeIncomingCallError(rawValue: incomingCallError.code)
        where incomingCallErrorCode == .filteredByDoNotDisturb {
        // handle do not disturb
    }
}
```

Audio

Benefits



# Audio

## Benefits

High priority session

# Audio

## Benefits

High priority session

Fewer interruptions

# Audio

## Benefits

High priority session

Fewer interruptions

Audio routing hints

# Audio

Incoming call

---

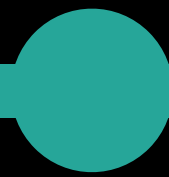
# Audio

Incoming call

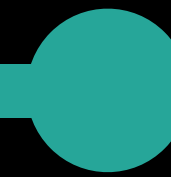
---

# Audio

Incoming call



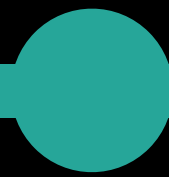
Answer Received



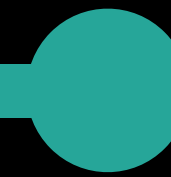
Answer Fulfilled

# Audio

Incoming call



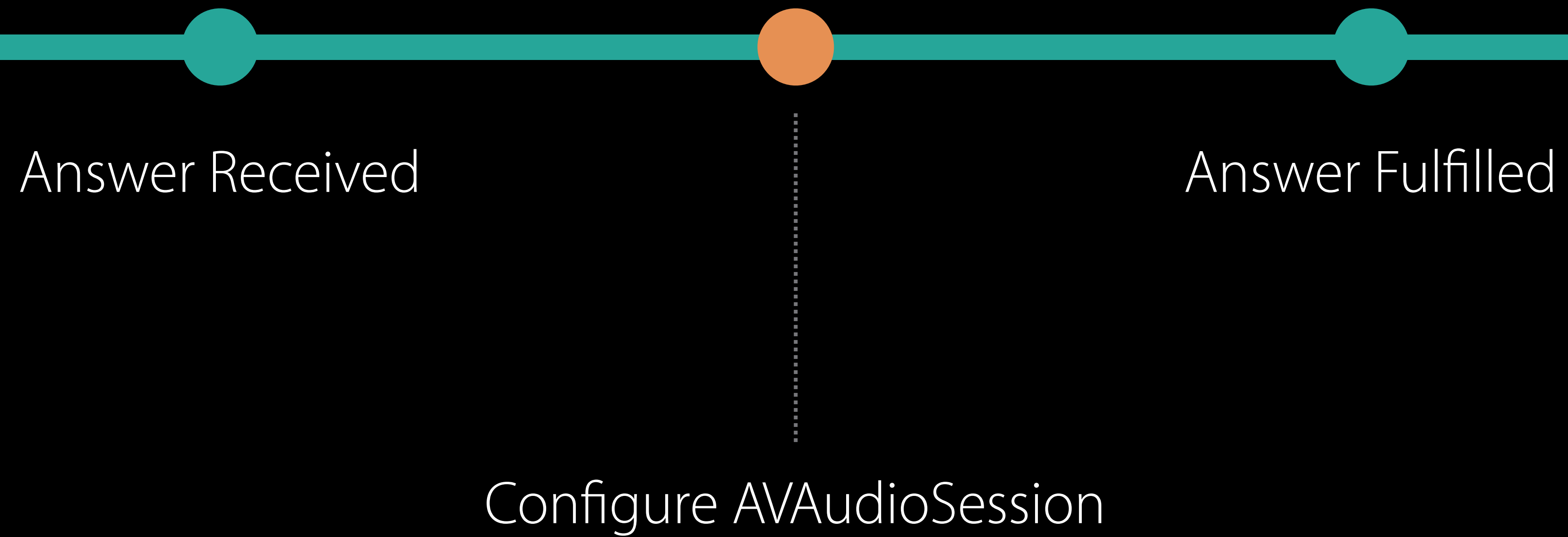
Answer Received



Answer Fulfilled

# Audio

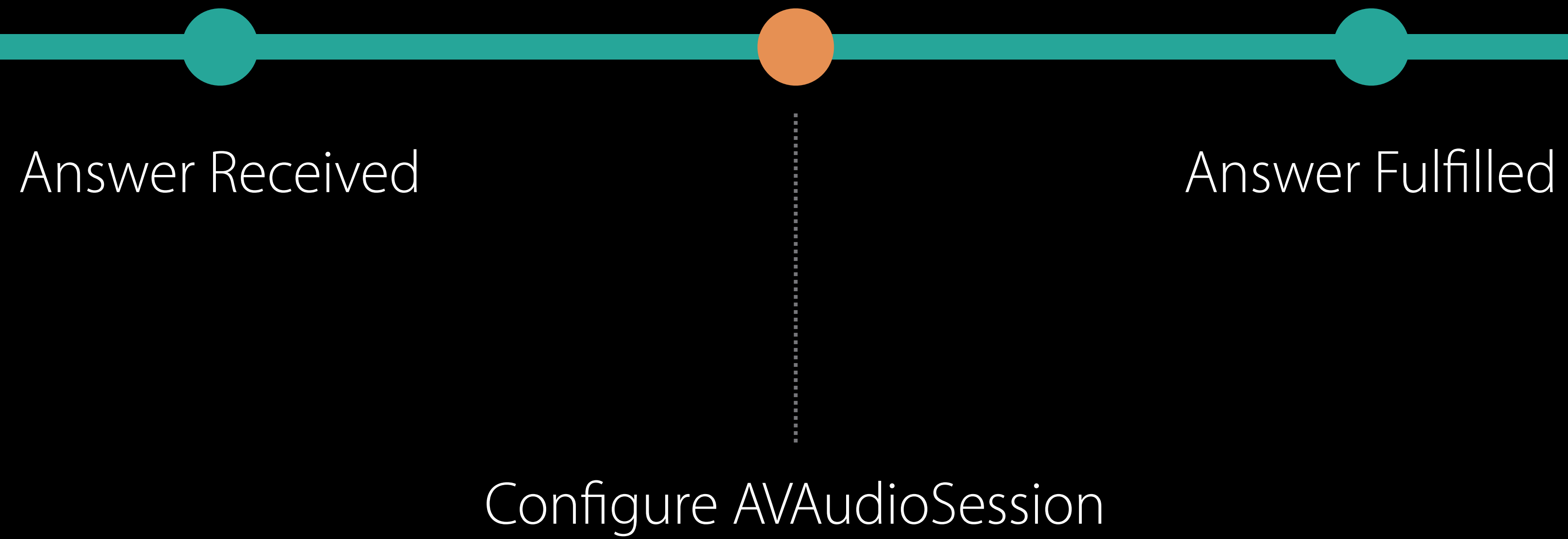
Incoming call





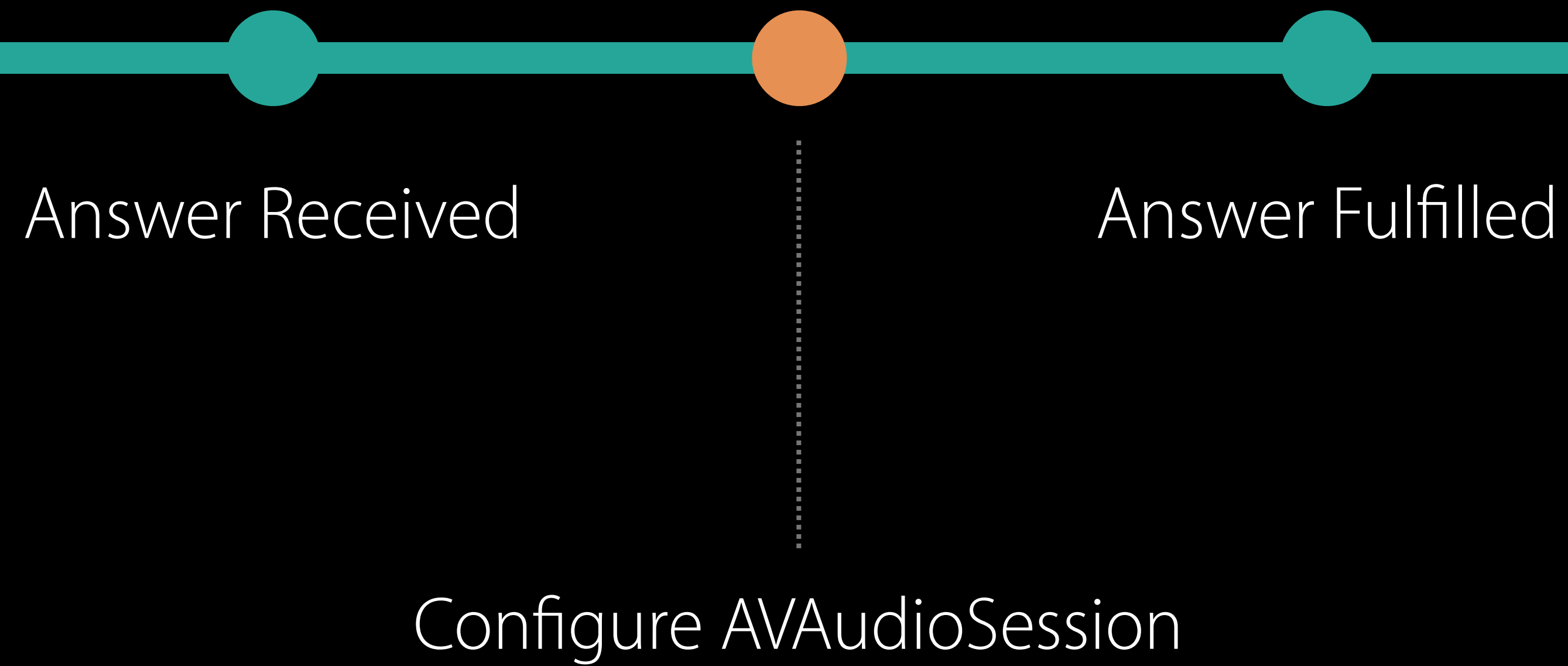
# Audio

Incoming call



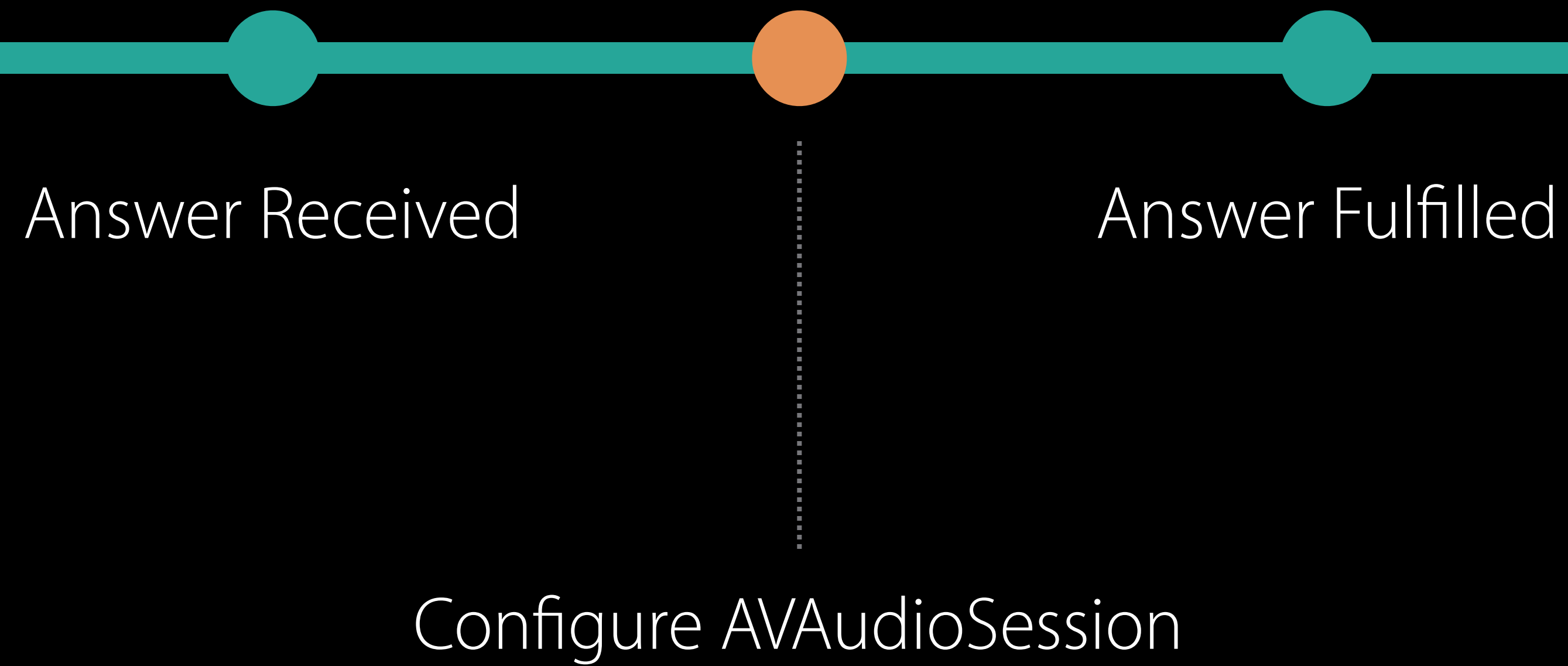
# Audio

Incoming call



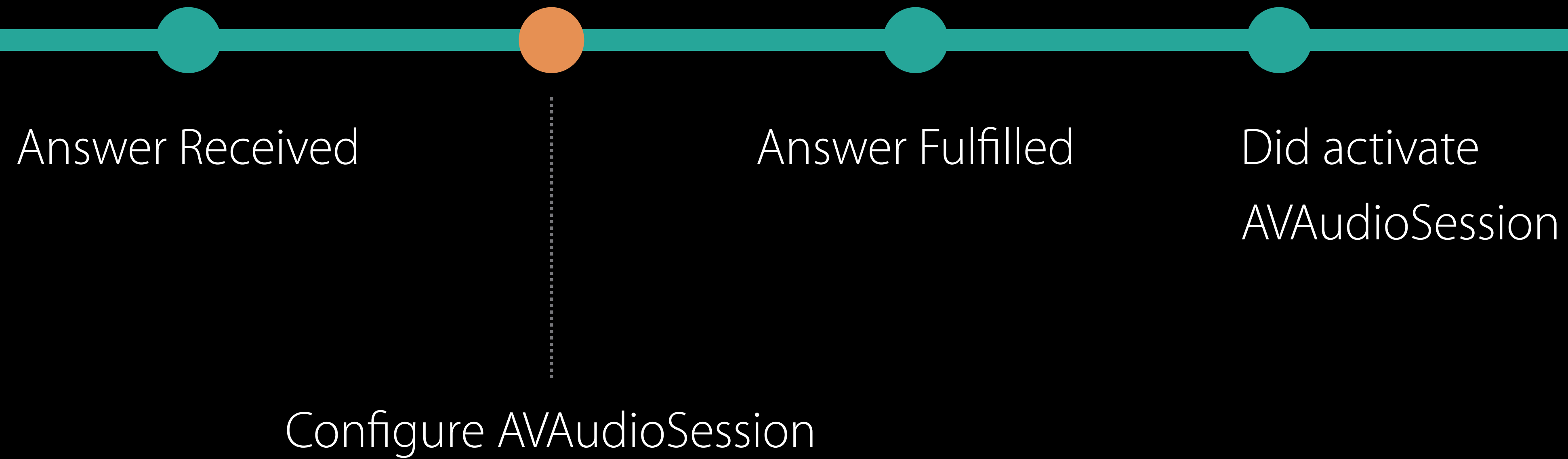
# Audio

Incoming call



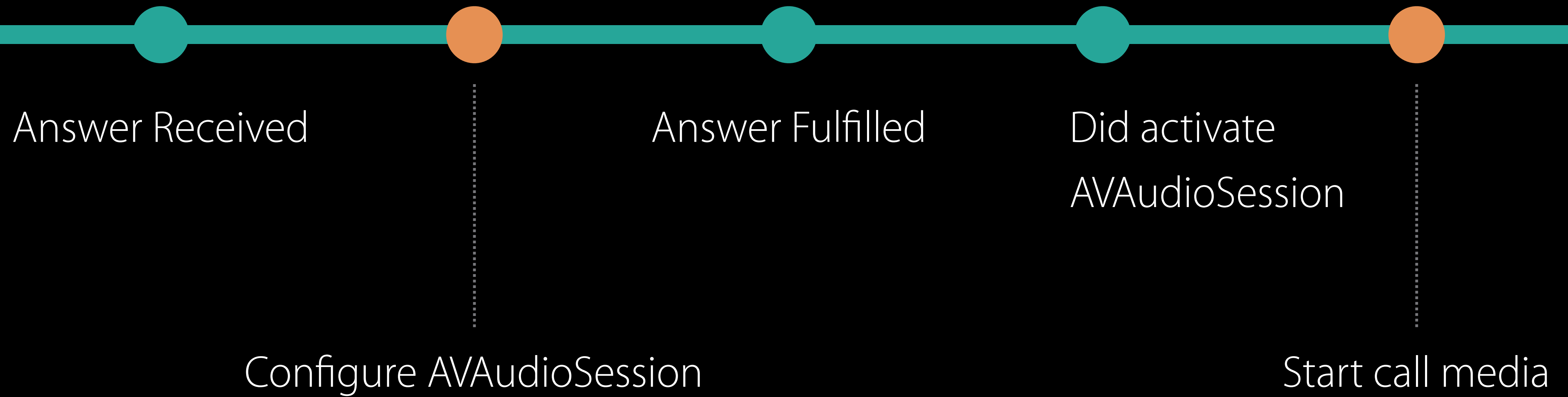
# Audio

Incoming call



# Audio

Incoming call



# Recap

API details

Authorization

Configuration

Action Errors

System Restrictions

Audio

# Summary



# Summary

Integration into the system





# Summary

Integration into the system

Feature parity



# Summary

Integration into the system

Feature parity

Better app visibility



More Information

<https://developer.apple.com/wwdc16/230>

# Related Sessions

---

Introducing SiriKit

Presidio

Wednesday 5:00PM

---

Extending Your Apps with SiriKit

Nob Hill

Thursday 1:40PM

---

Networking with for the Modern Internet

Pacific Heights

Thursday 3:00PM

---

Delivering an Exceptional Audio Experience

Nob Hill

Friday 1:40PM

---

# Labs

---

CallKit Lab

Frameworks Lab A

Friday 9:00AM

---

SiriKit Lab

Frameworks Lab B

Friday 9:00AM

---

Networking Lab 2

Frameworks Lab D

Friday 2:00PM

---

Audio Lab

Graphics, Games, and  
Media Lab D

Friday 3:00PM

---



W

W

D

C

1

6