Media #WWDC16

AVCapturePhotoOutput—Beyond the Basics

Addendum to Session 501

Session 511

Brad Ford Apple





























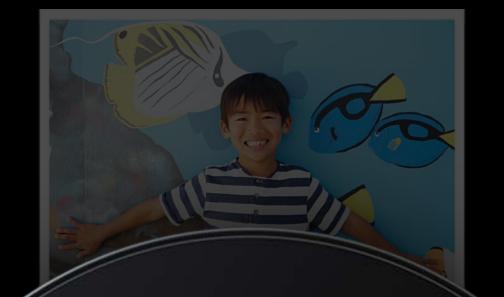
































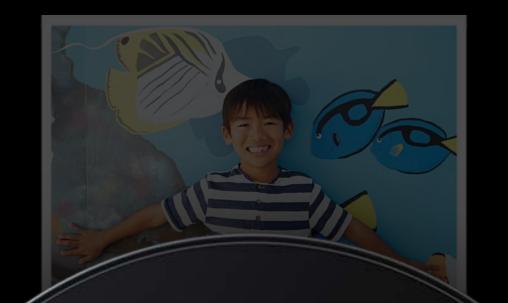




















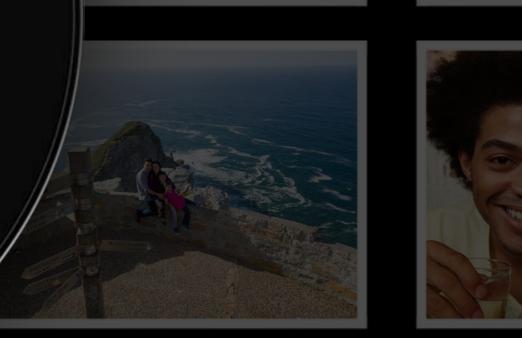


















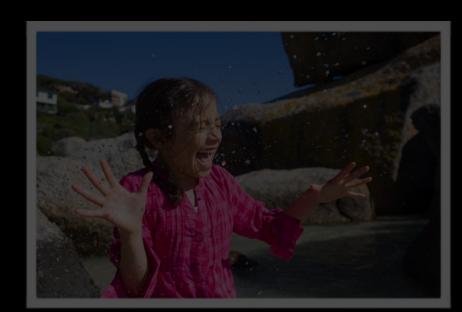




















Wide Color















AVCapturePhotoOutput—Beyond the Basics

AVCapturePhotoOutput—Beyond the Basics

Scene Monitoring

AVCapturePhotoOutput—Beyond the Basics

- Scene Monitoring
- Resource Preparation and Reclamation

AVCapturePhotoOutput—Beyond the Basics

- Scene Monitoring
- Resource Preparation and Reclamation

Camera Privacy Policy Changes





Functional programming model



Functional programming model

Photo settings encapsulation



Functional programming model

Photo settings encapsulation

A delegate-style interface for tracking the progress of photo capture requests



Functional programming model

Photo settings encapsulation

A delegate-style interface for tracking the progress of photo capture requests

Resolving of photo settings to an immutable object

AVCapturePhotoOutput Scene Monitoring



AVCapturePhotoOutput Scene Monitoring



AVCapturePhotoOutput Scene Monitoring



Flash scene

Flash scene

True Tone flash for rear camera

Flash scene

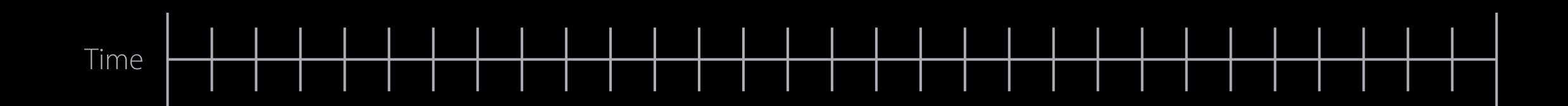
- True Tone flash for rear camera
- Retina flash for front camera

Still Image Stabilization scene

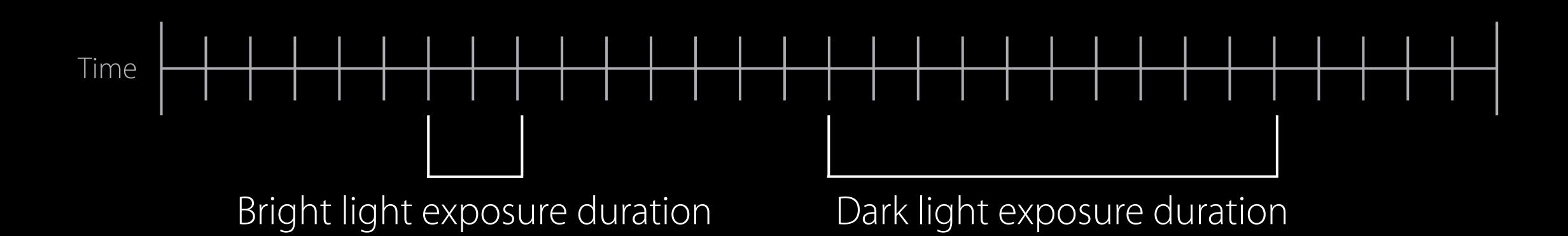
Multiple image fusion capture

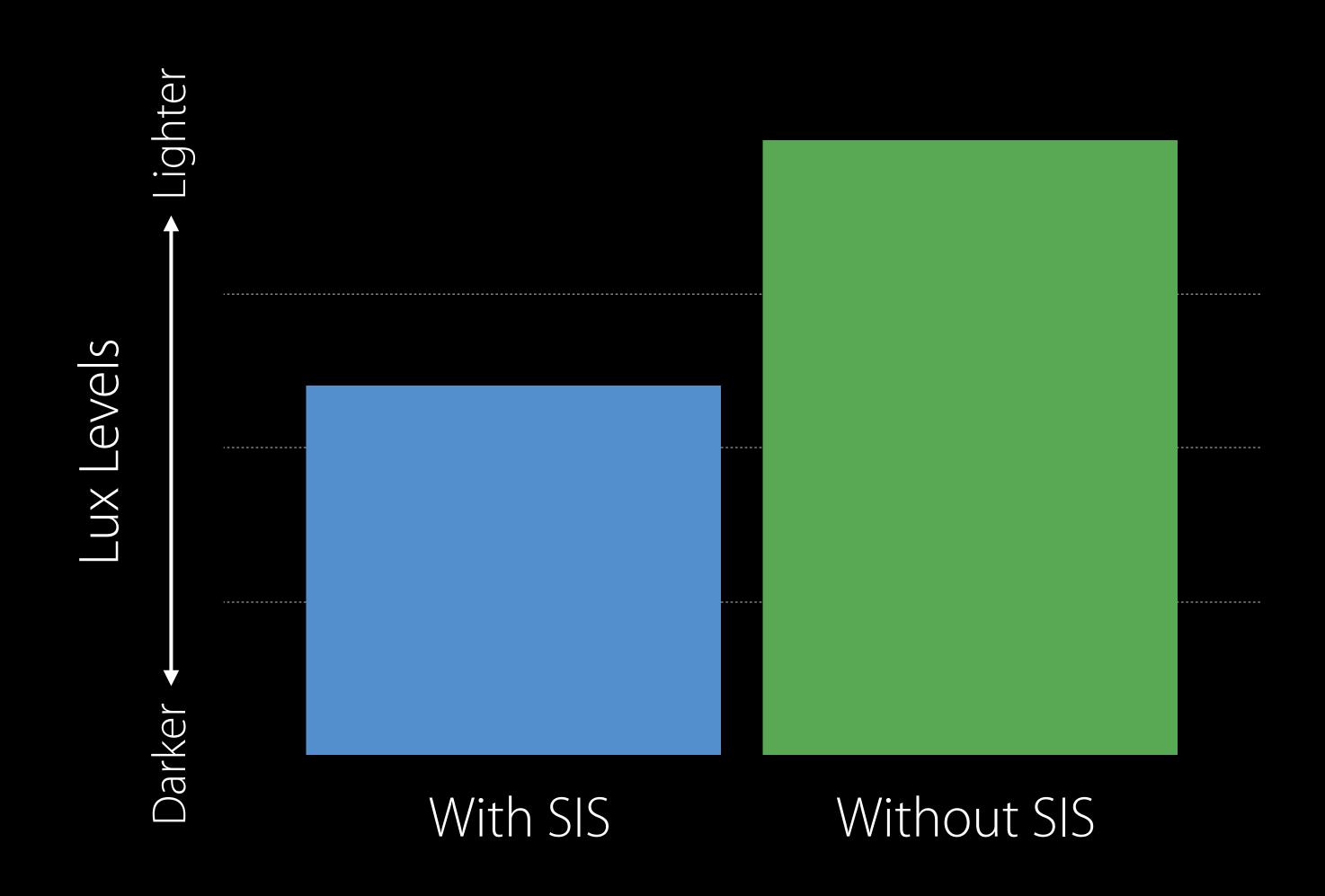
- Multiple image fusion capture
- Reduces low light blur

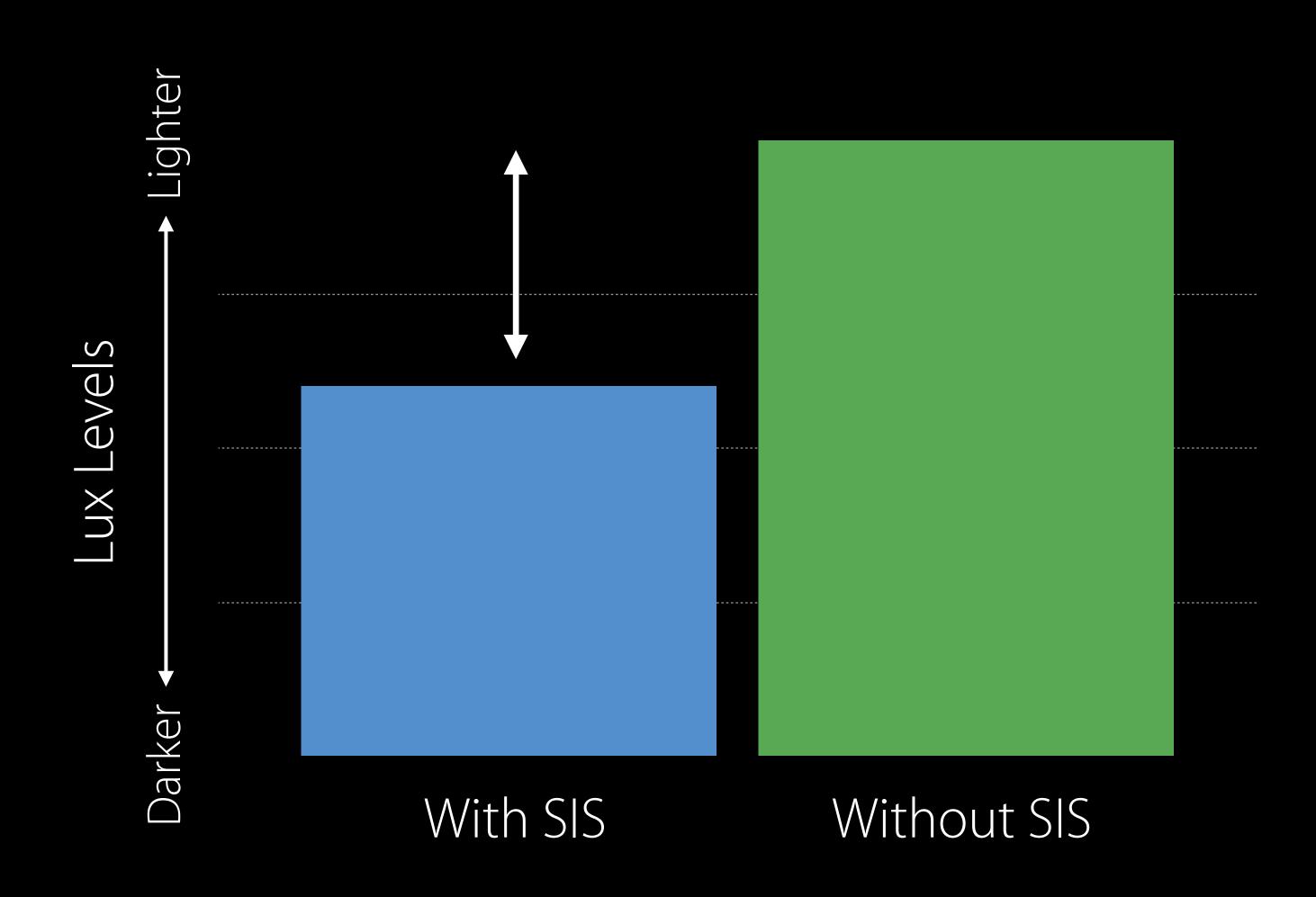
- Multiple image fusion capture
- Reduces low light blur

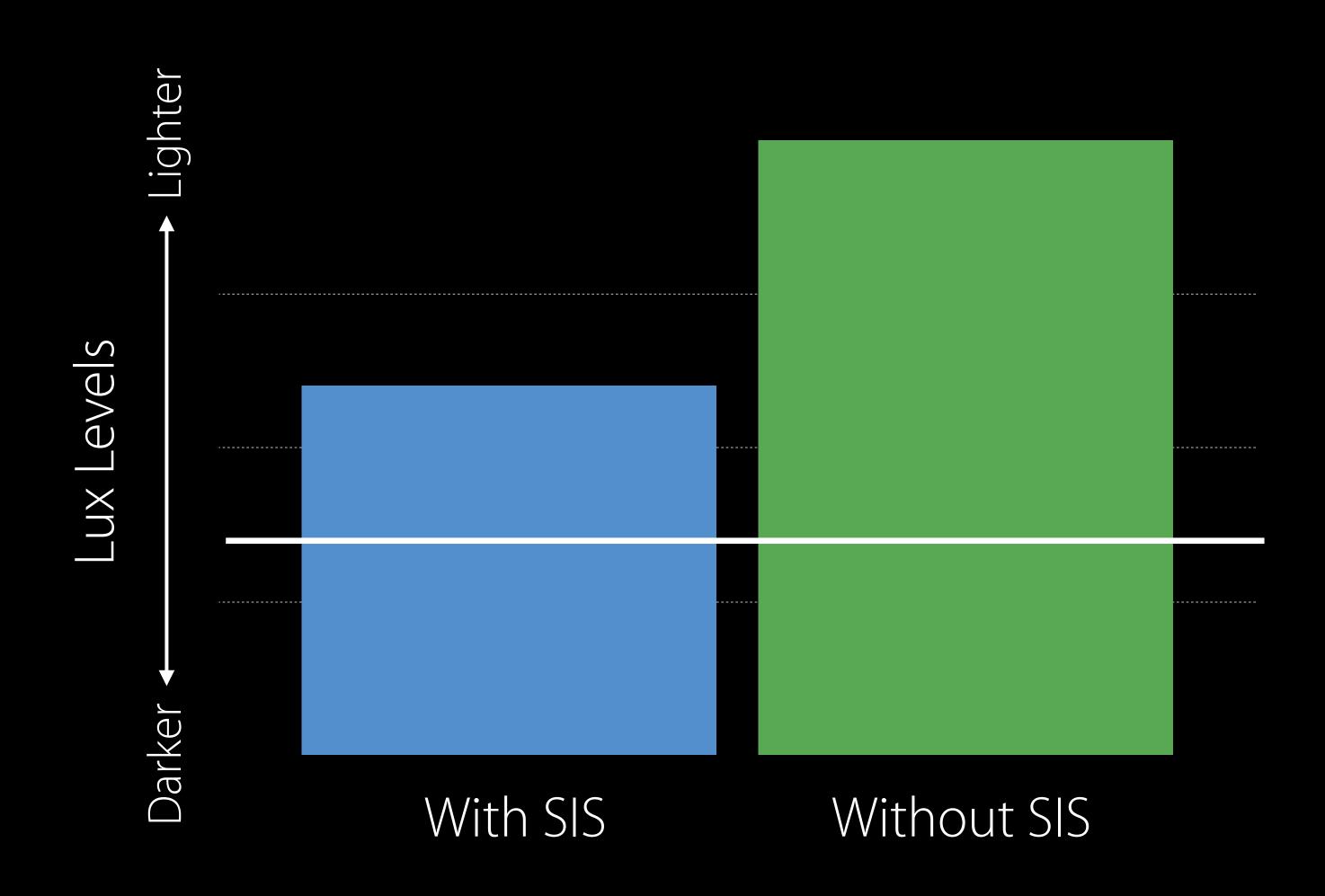


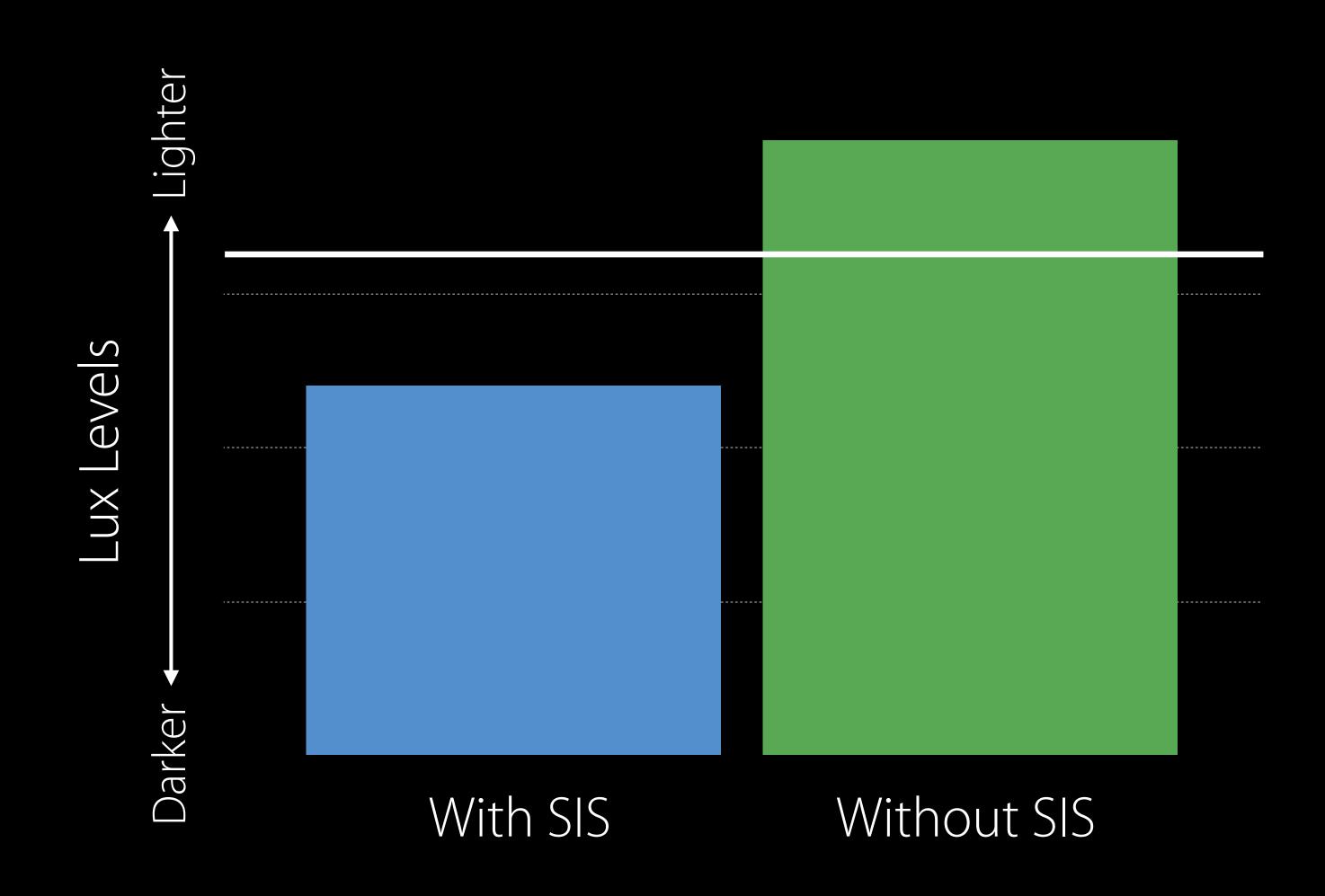
- Multiple image fusion capture
- Reduces low light blur











public var photoSettingsForSceneMonitoring: AVCapturePhotoSettings?

```
public var photoSettingsForSceneMonitoring: AVCapturePhotoSettings?

public var isStillImageStabilizationScene: Bool { get }

public var isFlashScene: Bool { get }
```

```
// Scene monitoring support
func setUpSceneMonitoring()
    let settingsForMonitoring = AVCapturePhotoSettings()
    settingsForMonitoring.flashMode = .auto
    settingsForMonitoring.isAutoStillImageStabilizationEnabled = true
    photoOutput.photoSettingsForSceneMonitoring = settingsForMonitoring
    photoOutput.addObserver(self, forKeyPath: "isFlashScene", options: .new, context: nil)
    photoOutput.addObserver(self, forKeyPath: "isStillImageStabilizationScene",
                                                              options: .new, context: nil)
```

```
// Scene monitoring support
func setUpSceneMonitoring()
    let settingsForMonitoring = AVCapturePhotoSettings()
    settingsForMonitoring.flashMode = .auto
    settingsForMonitoring.isAutoStillImageStabilizationEnabled = true
    photoOutput.photoSettingsForSceneMonitoring = settingsForMonitoring
    photoOutput.addObserver(self, forKeyPath: "isFlashScene", options: .new, context: nil)
    photoOutput.addObserver(self, forKeyPath: "isStillImageStabilizationScene",
                                                              options: .new, context: nil)
```

```
// Scene monitoring support
func setUpSceneMonitoring()
    let settingsForMonitoring = AVCapturePhotoSettings()
    settingsForMonitoring.flashMode = .auto
    settingsForMonitoring.isAutoStillImageStabilizationEnabled = true
    photoOutput.photoSettingsForSceneMonitoring = settingsForMonitoring
    photoOutput.addObserver(self, forKeyPath: "isFlashScene", options: .new, context: nil)
    photoOutput.addObserver(self, forKeyPath: "isStillImageStabilizationScene",
                                                              options: .new, context: nil)
```

```
// Scene monitoring support
func setUpSceneMonitoring()
    let settingsForMonitoring = AVCapturePhotoSettings()
    settingsForMonitoring.flashMode = .auto
    settingsForMonitoring.isAutoStillImageStabilizationEnabled = true
    photoOutput.photoSettingsForSceneMonitoring = settingsForMonitoring
    photoOutput.addObserver(self, forKeyPath: "isFlashScene", options: .new, context: nil)
    photoOutput.addObserver(self, forKeyPath: "isStillImageStabilizationScene",
                                                              options: .new, context: nil)
```

photoSettingsForSceneMonitoring is nullable

photoSettingsForSceneMonitoring is nullable

isStillImageStabilizationScene = false

photoSettingsForSceneMonitoring is nullable

```
isStillImageStabilizationScene = false
```

```
isFlashScene = false
```

```
photoSettingsForSceneMonitoring is nullable
isStillImageStabilizationScene = false
isFlashScene = false

If flashMode = .off, isFlashScene is always false
```

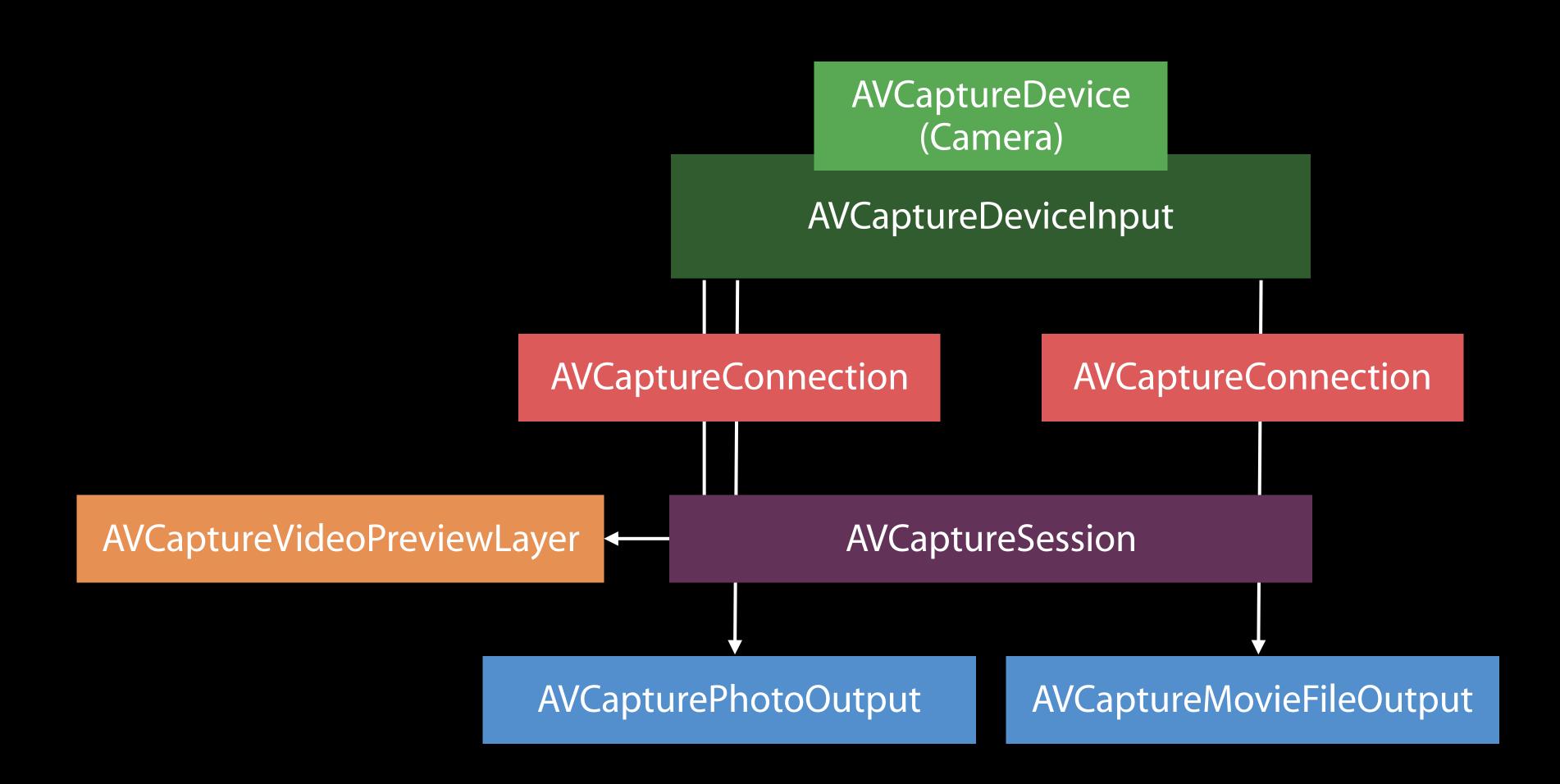
Scene Monitoring Recommendations

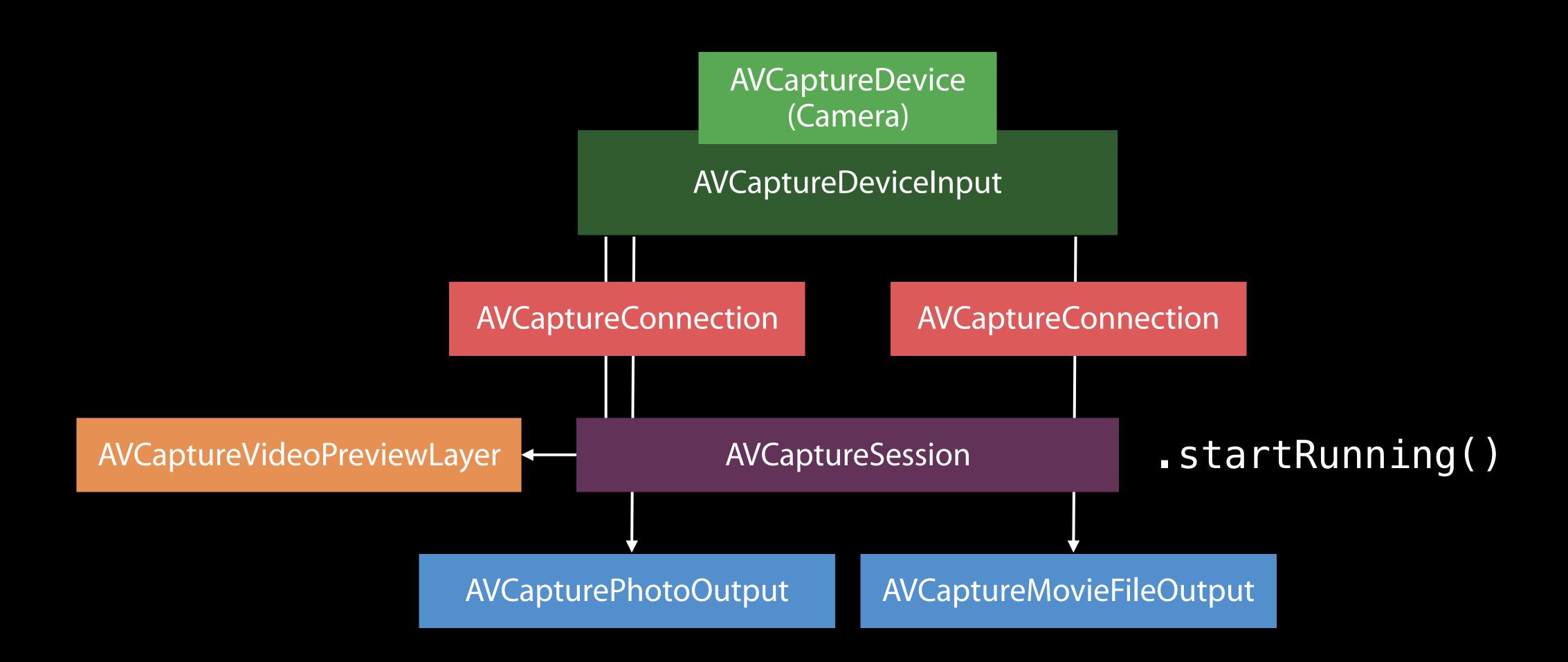
Scene Monitoring Recommendations

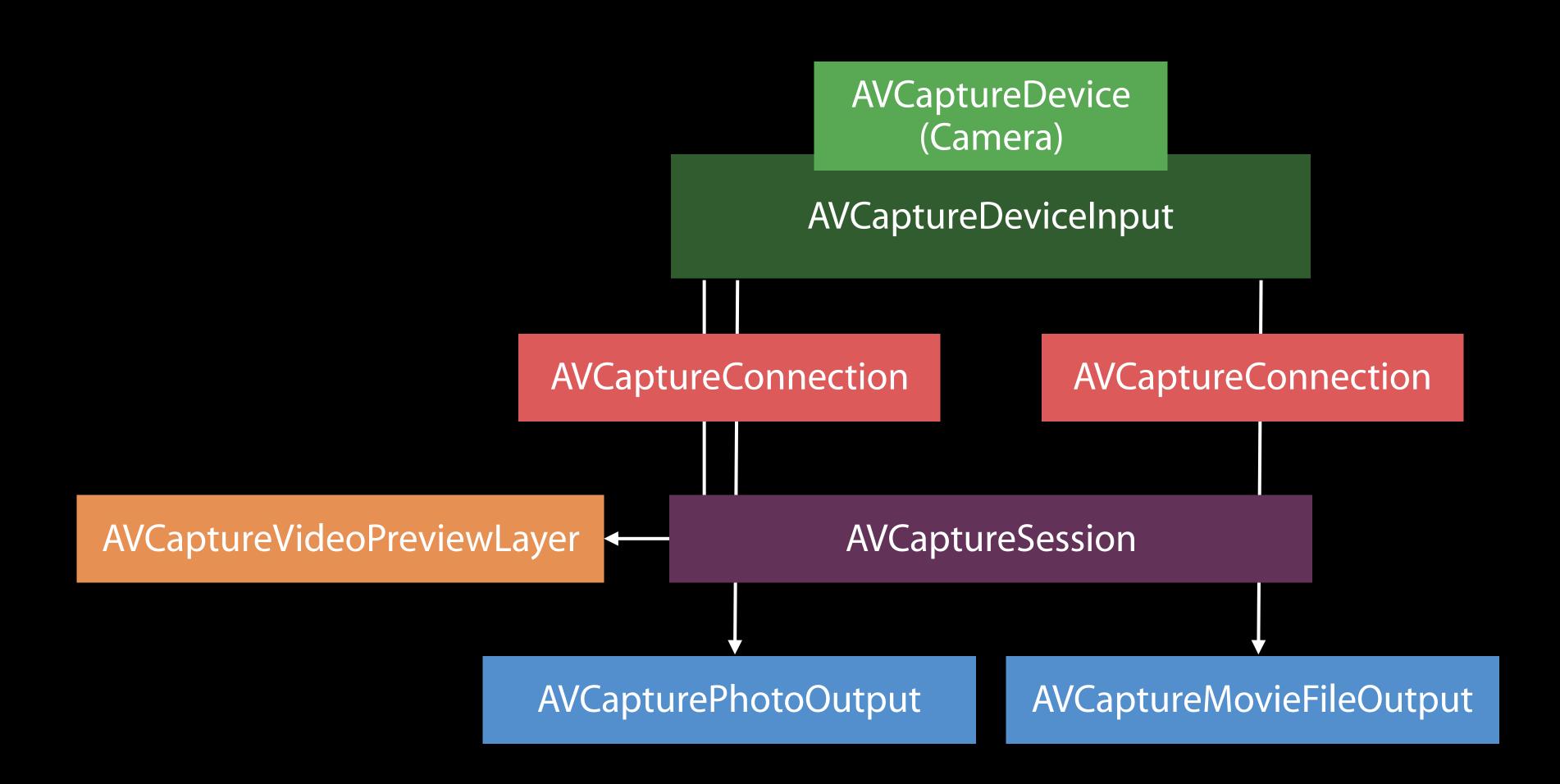
Monitor what you intend to capture

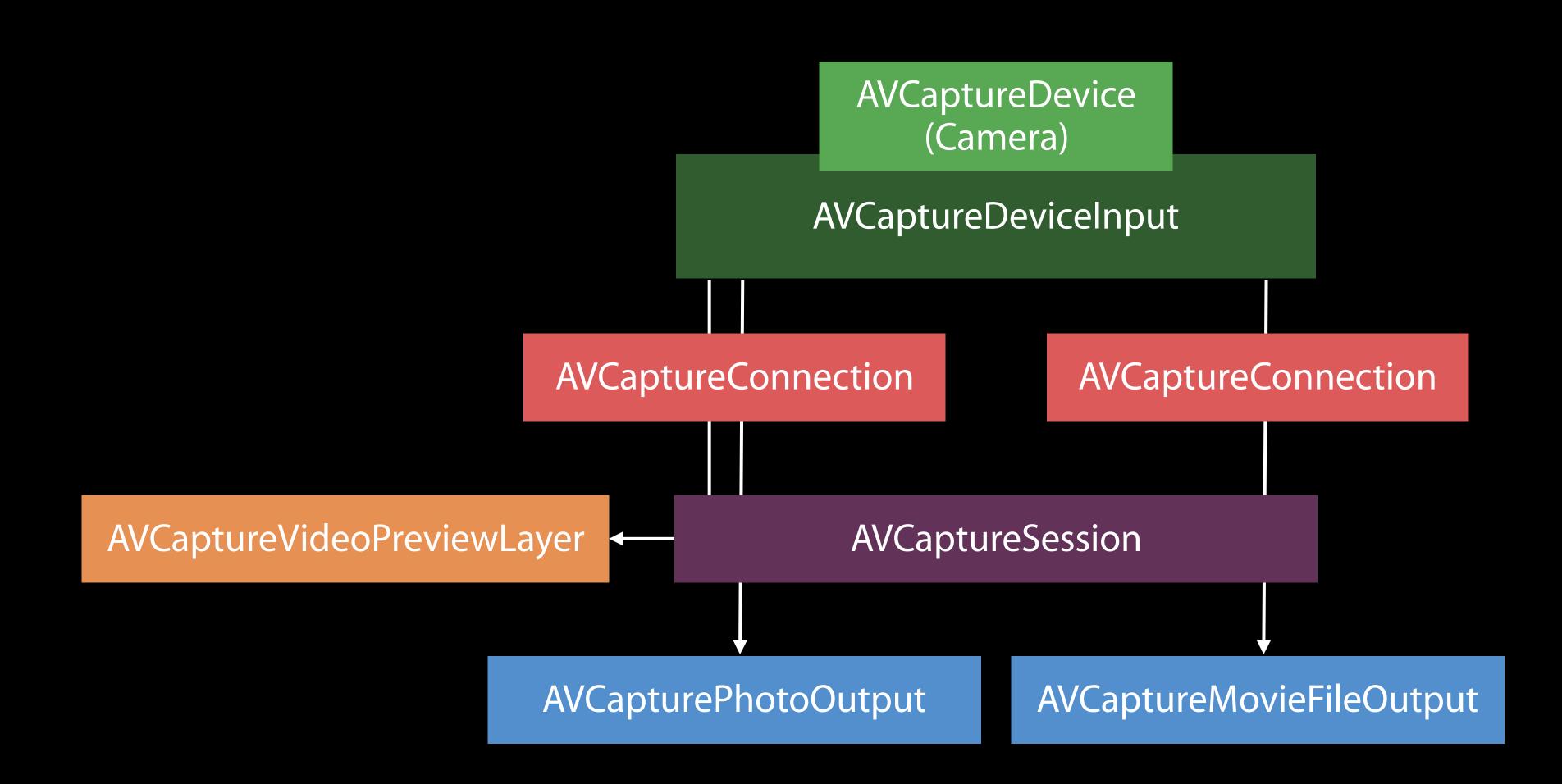


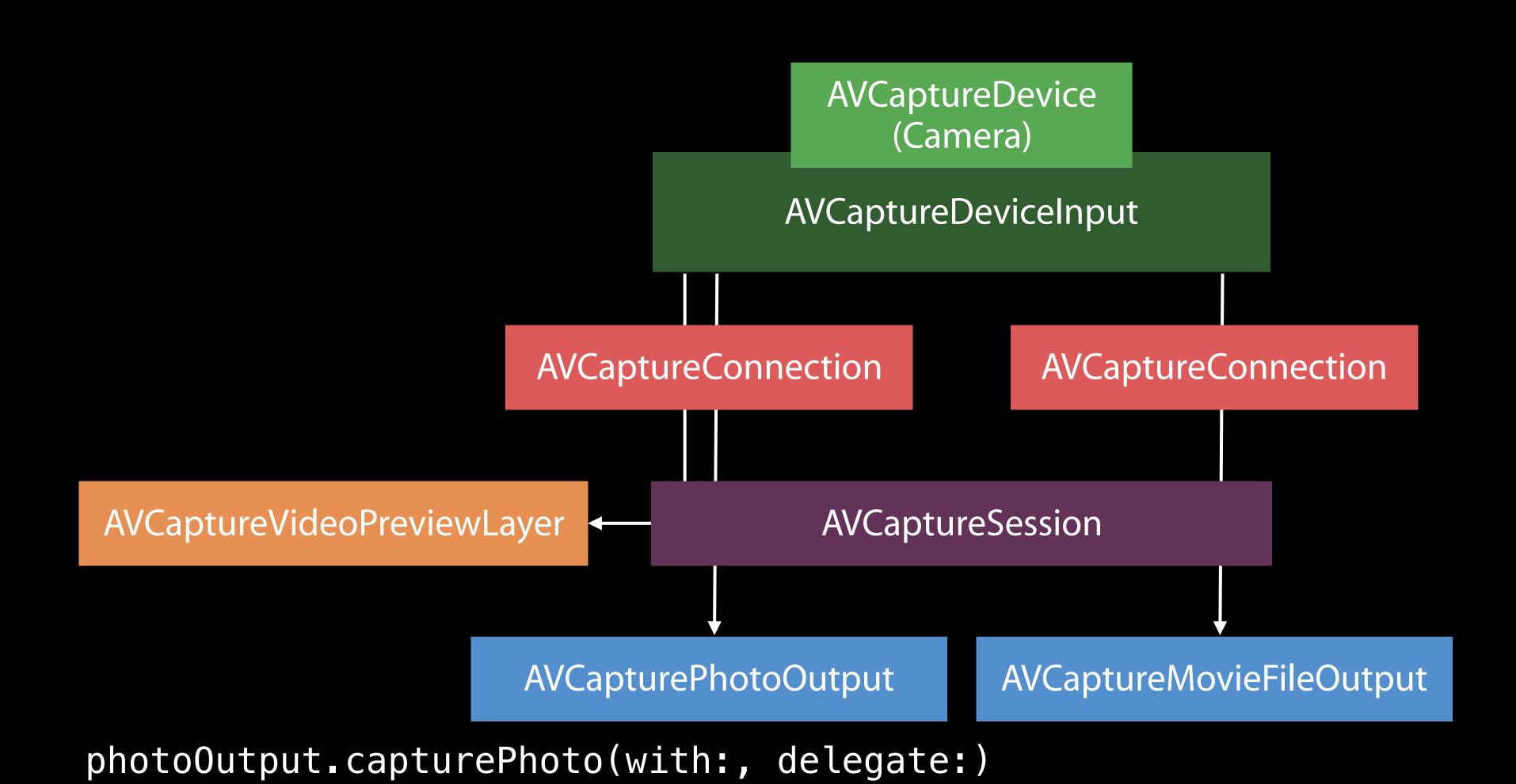
Resource Preparation and Reclamation

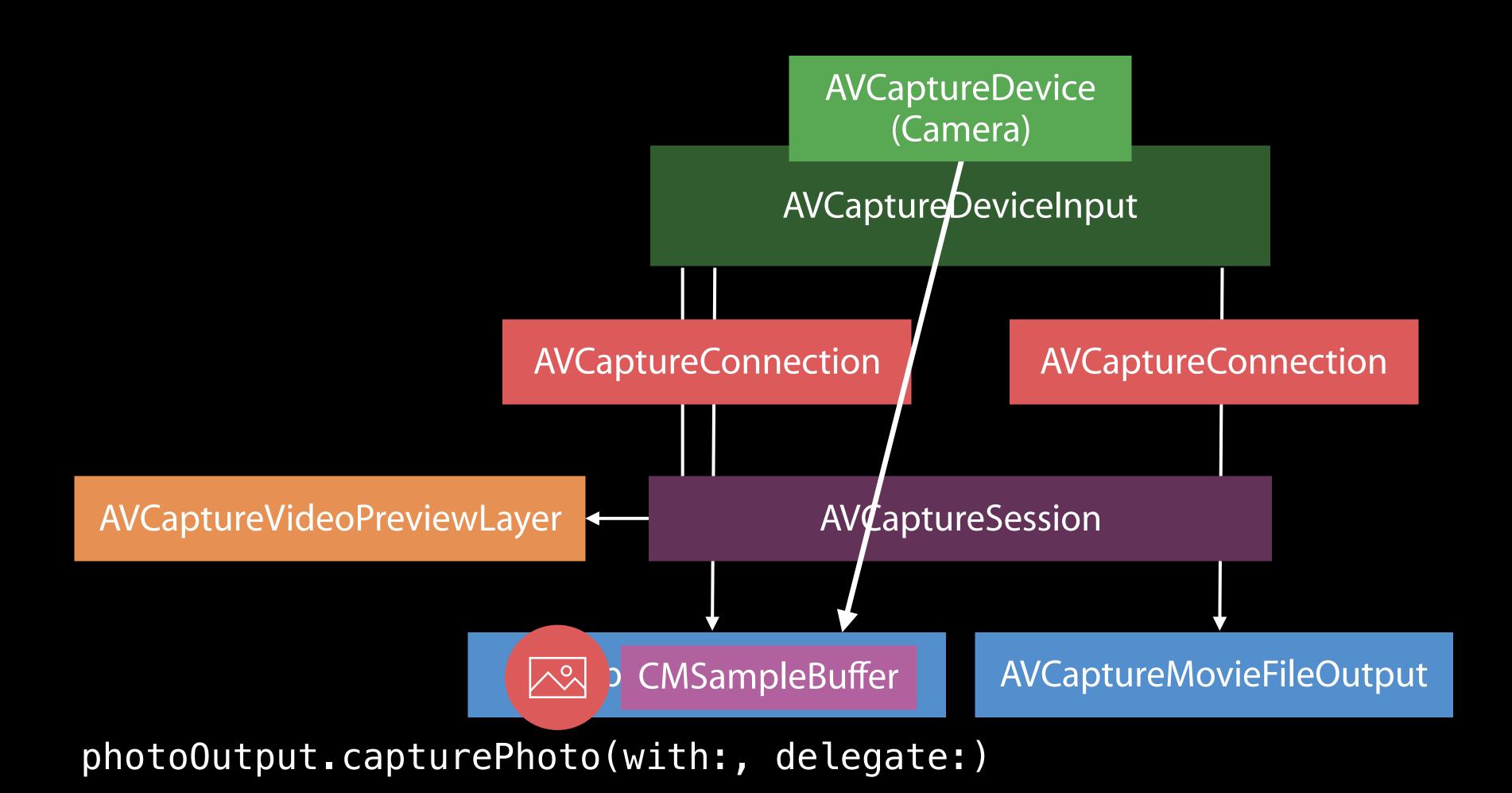








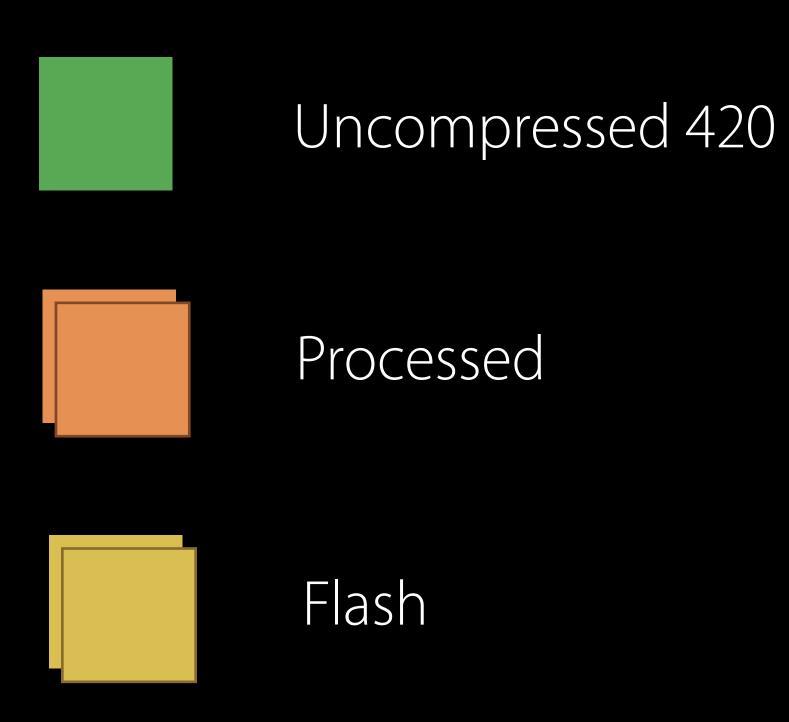


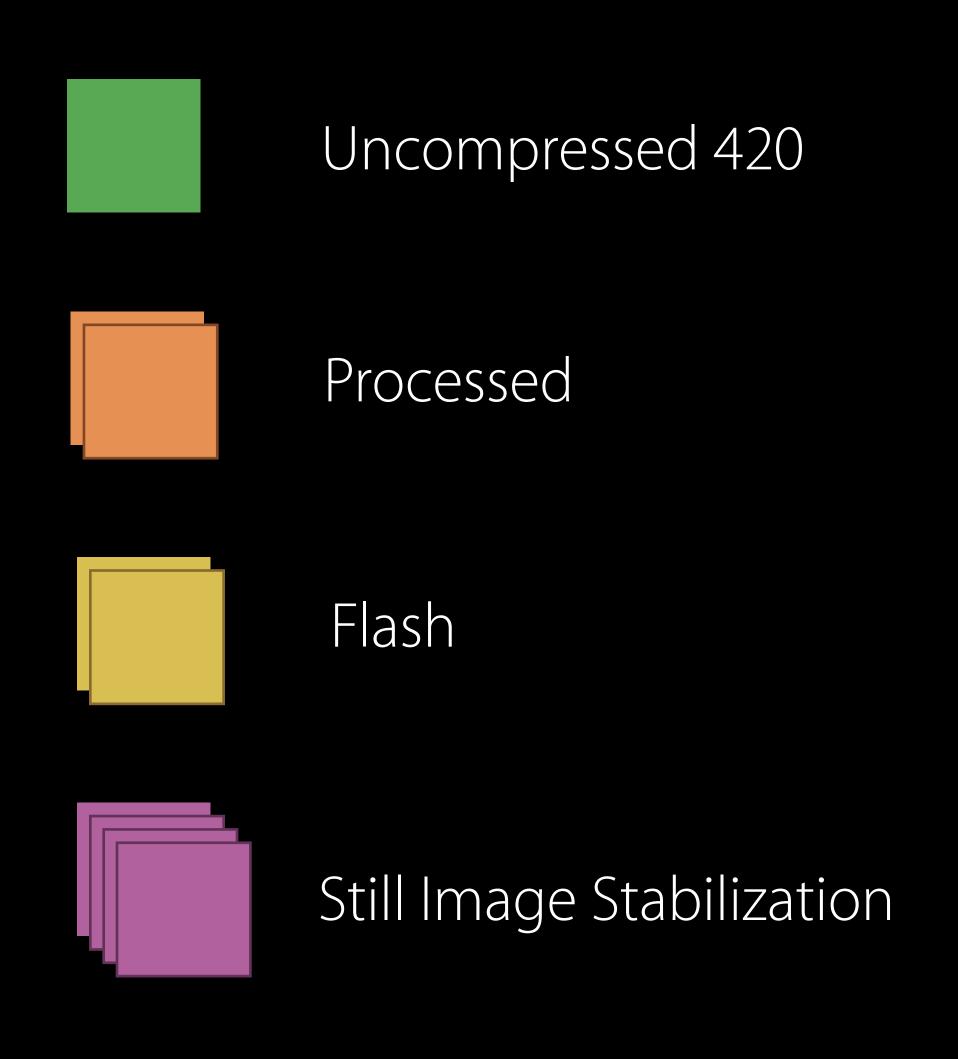


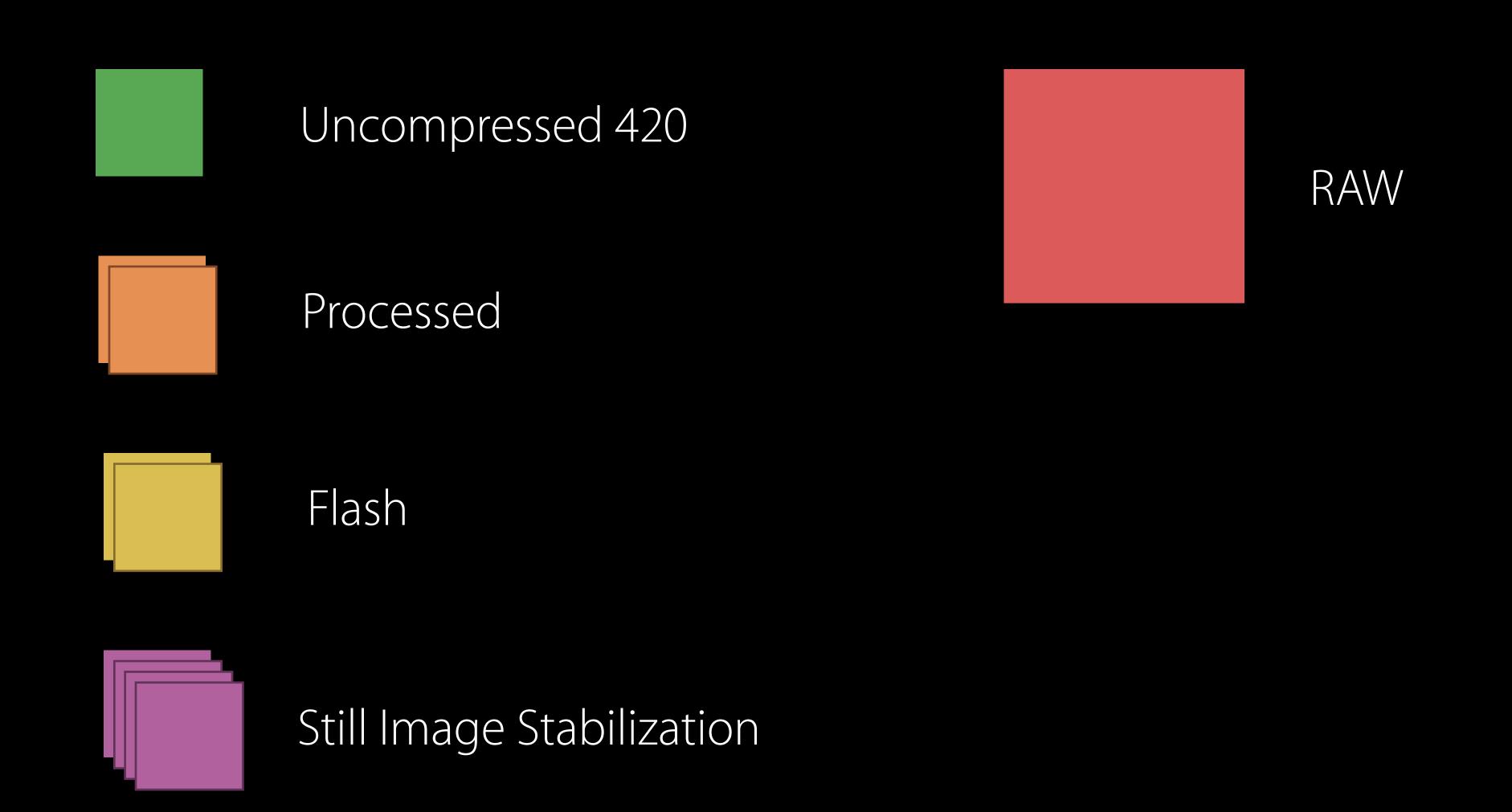


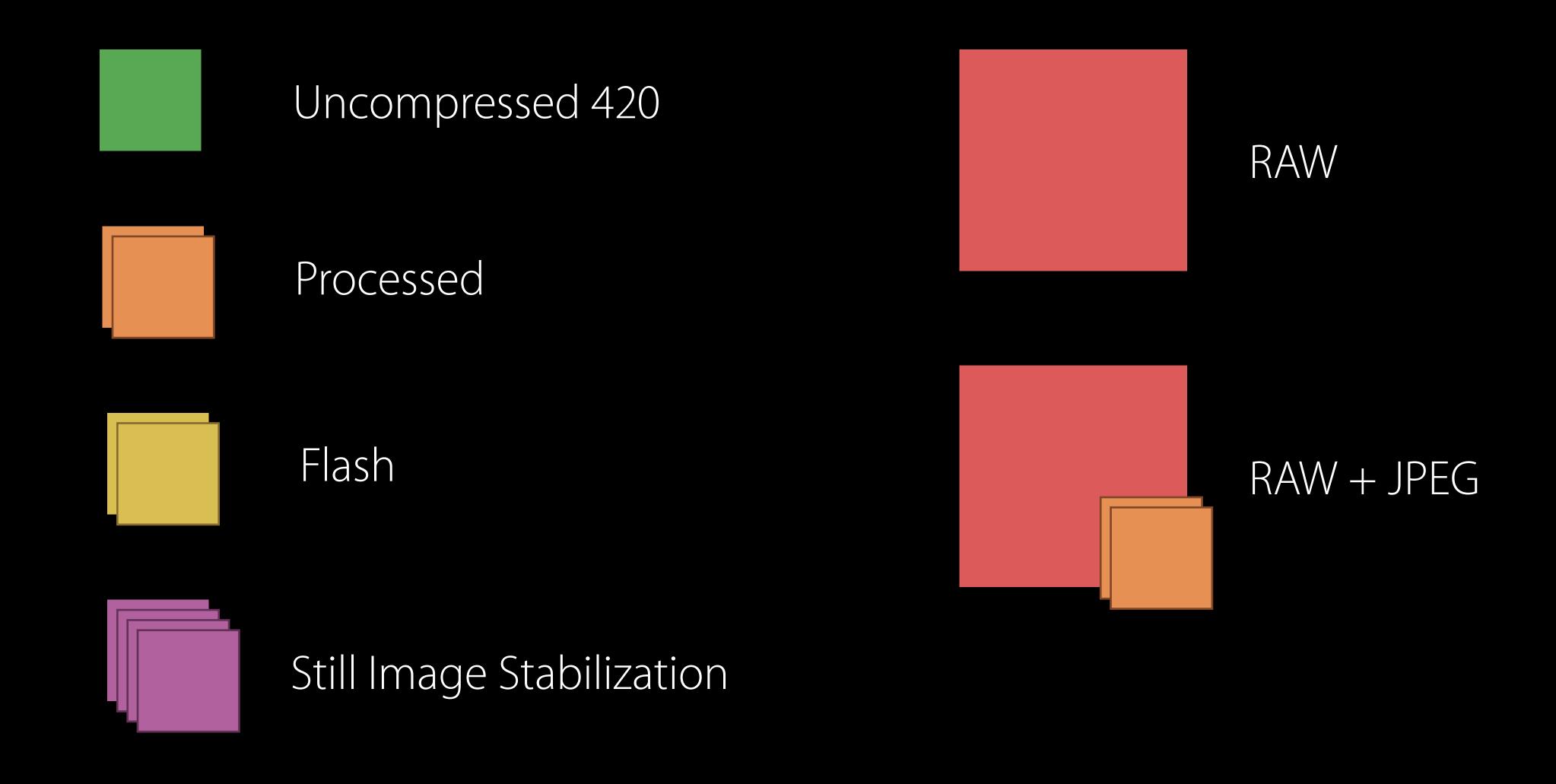


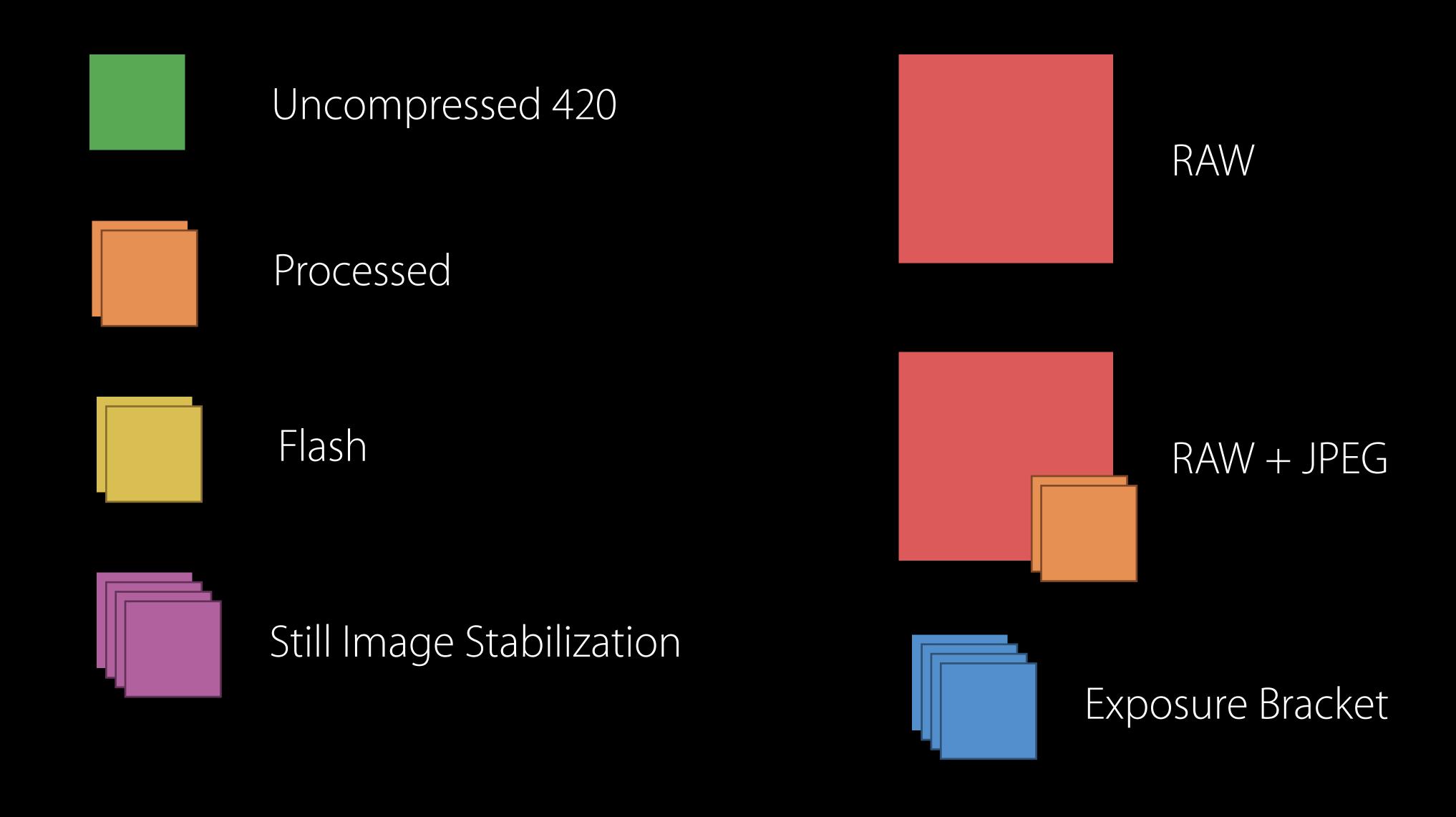


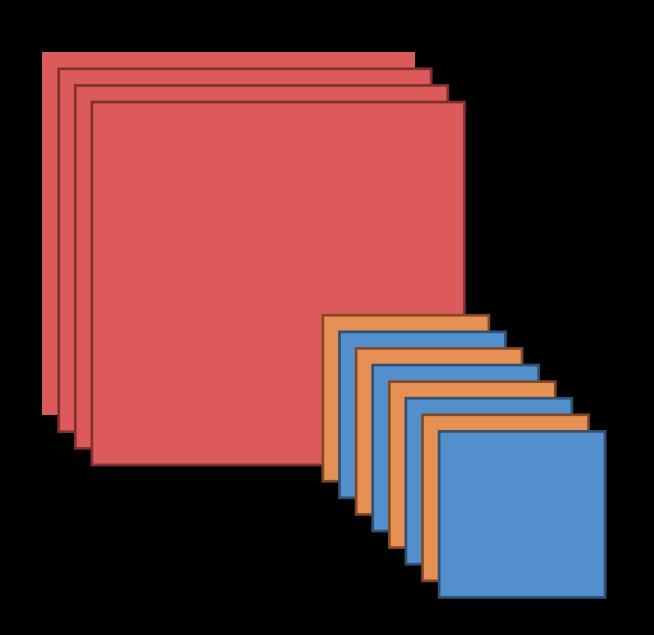












RAW + JPEG Exposure Bracket

Over-preparing

Over-preparing

Bakes a cake every day of the year

Over-preparing

Bakes a cake every day of the year

Under-preparing

Over-preparing

Bakes a cake every day of the year

Under-preparing

Misses the moment



```
public func setPreparedPhotoSettingsArray(
    _ preparedPhotoSettingsArray: [AVCapturePhotoSettings],
    completionHandler handler: ((Bool, NSError!) -> Swift.Void)!)
```

```
public func setPreparedPhotoSettingsArray(
    _ preparedPhotoSettingsArray: [AVCapturePhotoSettings],
    completionHandler handler: ((Bool, NSError!) -> Swift.Void)!)
```

Prepares resources for all the types of capture in your array of settings

Prepares resources for all the types of capture in your array of settings

Reclaims unneeded resources if any

Prepares resources for all the types of capture in your array of settings

Reclaims unneeded resources if any

Reclaims everything if you pass an empty array

Prepares resources for all the types of capture in your array of settings

Reclaims unneeded resources if any

Reclaims everything if you pass an empty array

Calls you back when all resources are prepared

Prepares resources for all the types of capture in your array of settings

Reclaims unneeded resources if any

Reclaims everything if you pass an empty array

Calls you back when all resources are prepared

Returns error if resources couldn't be prepared

Prepares resources for all the types of capture in your array of settings

Reclaims unneeded resources if any

Reclaims everything if you pass an empty array

Calls you back when all resources are prepared

Returns error if resources couldn't be prepared

preparedPhotoSettingsArray default value is [AVCapturePhotoSettings()]

preparedPhotoSettingsArray persists across

preparedPhotoSettingsArray persists across

AVCaptureSession start/stopRunning

preparedPhotoSettingsArray persists across

- AVCaptureSession start/stopRunning
- AVCaptureSession begin/commitConfiguration

preparedPhotoSettingsArray persists across

- AVCaptureSession start/stopRunning
- AVCaptureSession begin/commitConfiguration
 preparedPhotoSettingsArray participates in AVCaptureSession begin/

commitConfiguration deferred work semantics

Stopped AVCaptureSession Behavior

Stopped AVCaptureSession Behavior

setPreparedPhotoSettingsArray calls completion handler when you start running

Stopped AVCaptureSession Behavior

setPreparedPhotoSettingsArray calls completion handler when you start running Multiple setPreparedPhotoSettingsArray calls cancel previous calls

Prepare



Prepare

Prepare before running





Prepare



Re-prepare when your UI changes







```
public var isHighResolutionCaptureEnabled: Bool
```

```
public var isHighResolutionCaptureEnabled: Bool
public var isLivePhotoCaptureEnabled: Bool
```

```
public var isHighResolutionCaptureEnabled: Bool
public var isLivePhotoCaptureEnabled: Bool
public var isLivePhotoAutoTrimmingEnabled: Bool
```

Camera Privacy Policy Changes

Photos and videos are personal, sensitive data

Photos and videos are personal, sensitive data

Beginning in iOS 7, users are notified when an app uses the Camera or Microphone

Photos and videos are personal, sensitive data

Beginning in iOS 7, users are notified when an app uses the Camera or Microphone

"AVCam" Would Like to Access the Camera

"AVCam" Would Like to Access the Microphone

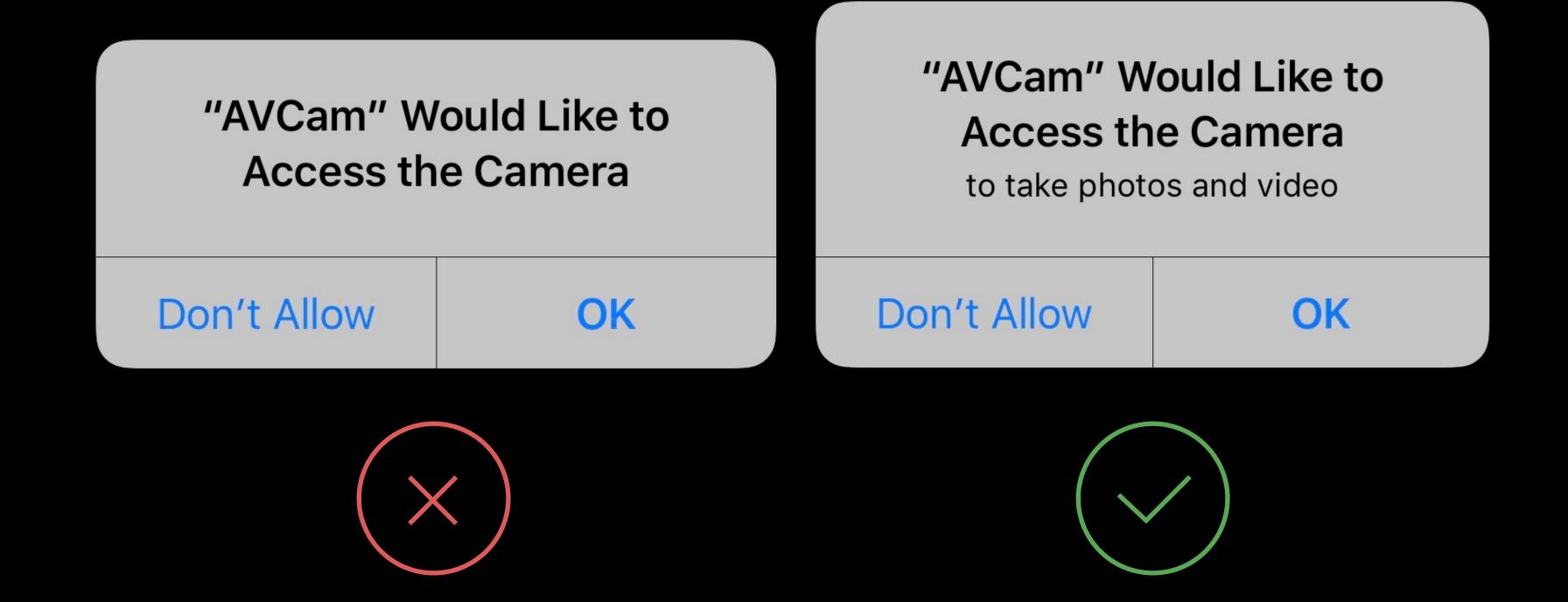
Don't Allow

OK

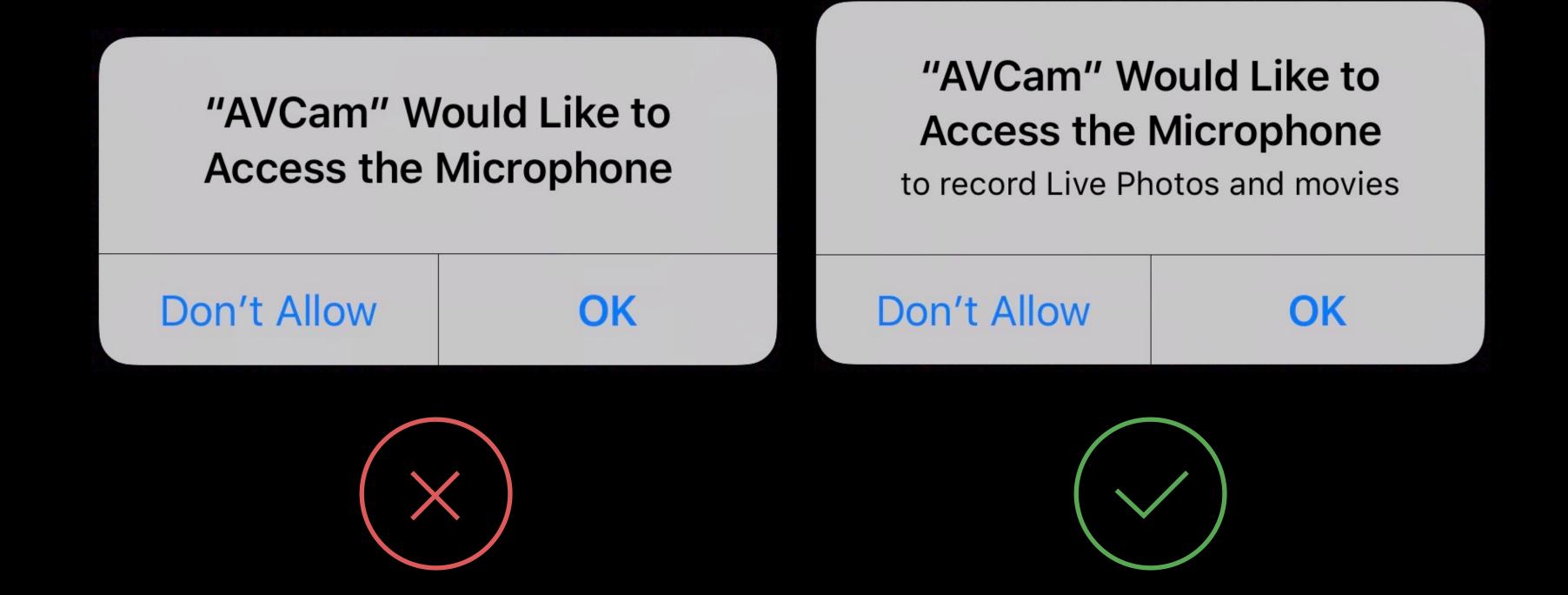
Don't Allow

OK

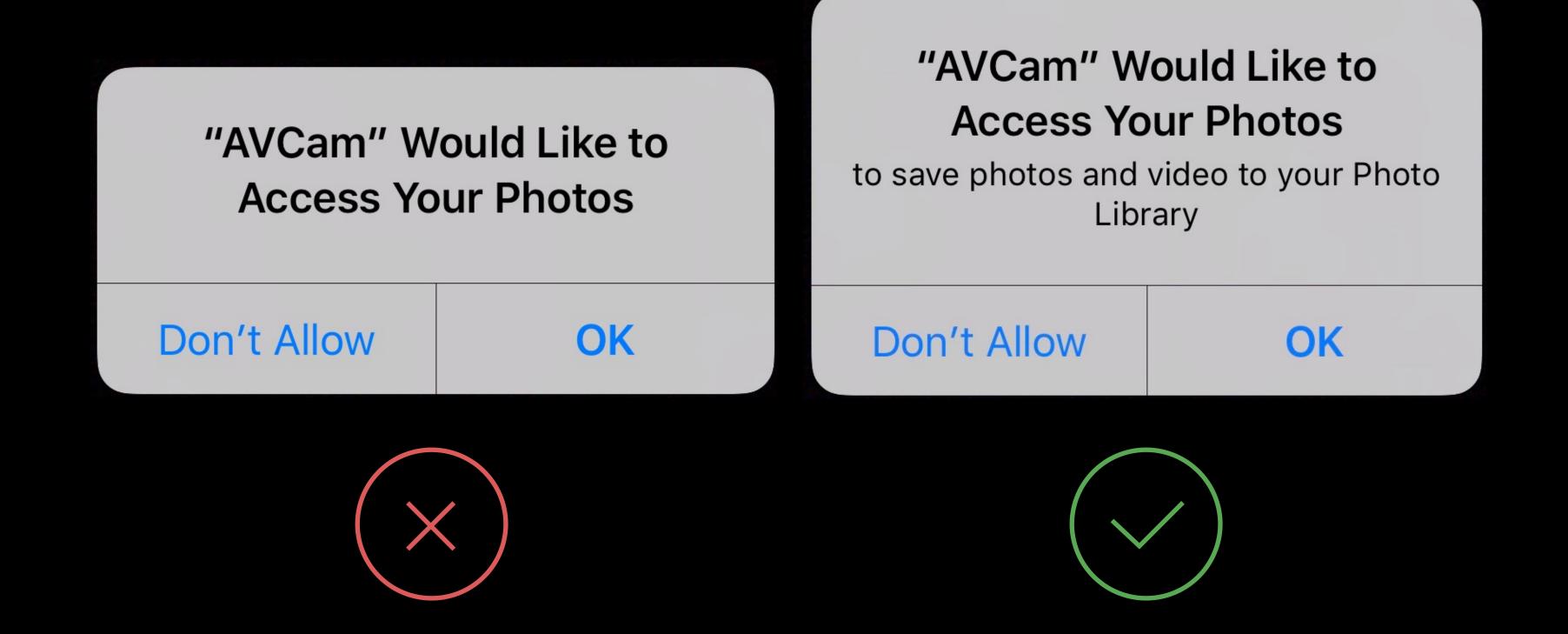
Starting in iOS 10, apps must provide a reason for accessing sensitive data



Starting in iOS 10, apps must provide a reason for accessing sensitive data



Starting in iOS 10, apps must provide a reason for accessing sensitive data



Key		Type	Value
Bundle name	‡	String	\$(PRODUCT_NAME)
Launch screen interface file base name	‡	String	Launch Screen
Localization native development region	‡	String	en ‡
Bundle version	\$	String	1
Privacy - Camera Usage Description 💲 (00	String	to take photos and video
Privacy - Camera Usage Description	‡	Boolean	YES ‡
Privacy - Contacts Usage Description	‡	String	APPL
Privacy - Health Share Usage Description	‡	String	Main
Privacy - Health Update Usage Description	‡	String	to record Live Photos and movies
Privacy - HomeKit Usage Description	÷	String	5.0
Privacy - Location Always Usage Description	÷	String	6.0
Privacy - Location Usage Description	•	String	\$(EXECUTABLE_NAME)
Privacy - Location When In Use Usage Descrip	•	Array	(1 item)
▶ Privacy - Media Library Usage Description	•	Array	(4 items)
Privacy - Microphone Usage Description	•	Boolean	YES
Bundle identifier	•	String	com.example.apple-samplecode.\$(PRODUCT_NAME:rfc1034ident
Bundle creator OS Type code	•	String	????
Application requires iPhone environment	•	Boolean	YES
► Supported interface orientations	.	Array	(4 items)
Privacy - Photo Library Usage Description	i	String	to save photos and videos

```
// Add to Info.plist
<plist version="1.0">
<dict>
   <key>NSCameraUsageDescription
   <string>to take photos and video</string>
   <key>NSMicrophoneUsageDescription
   <string>to record Live Photos and movies
   <key>NSPhotoLibraryUsageDescription
   <string>to save photos and videos
</dict>
</plist>
```

AVCapturePhotoOutput allows fine control of scene monitoring behavior

AVCapturePhotoOutput allows fine control of scene monitoring behavior AVCapturePhotoOutput allows on-demand resource allocation and reclamation

AVCapturePhotoOutput allows fine control of scene monitoring behavior

AVCapturePhotoOutput allows on-demand resource allocation and reclamation

Capture clients must provide a reason for camera, mic, and photos use

More Information

https://developer.apple.com/wwdc16/501

Related Sessions

Advances in iOS Photography	Pacific Heights	Tuesday 11:00AM
Live Photo Editing and RAW Processing with Core Image	Nob Hill	Thursday 11:00AM
Working With Wide Color	Mission	Thursday 1:40PM

Labs

Photo Capture Lab	Graphics, Games, and Media Lab D	Tuesday 1:00PM
PhotoKit Lab	Graphics, Games, and Media Lab D	Tuesday 4:00PM
Color Lab	Frameworks Lab A	Wednesday 1:00PM
Photo Capture Lab	Graphics, Games, and Media Lab D	Thursday 9:00AM
Live Photo & Core Image Lab	Graphics, Games, and Media Lab D	Thursday 1:30PM
Live Photo & Core Image Lab	Graphics, Games, and Media Lab D	Friday 9:00AM
Color Lab	Graphics, Games, and Media Lab C	Friday 4:00PM

W W D C 16