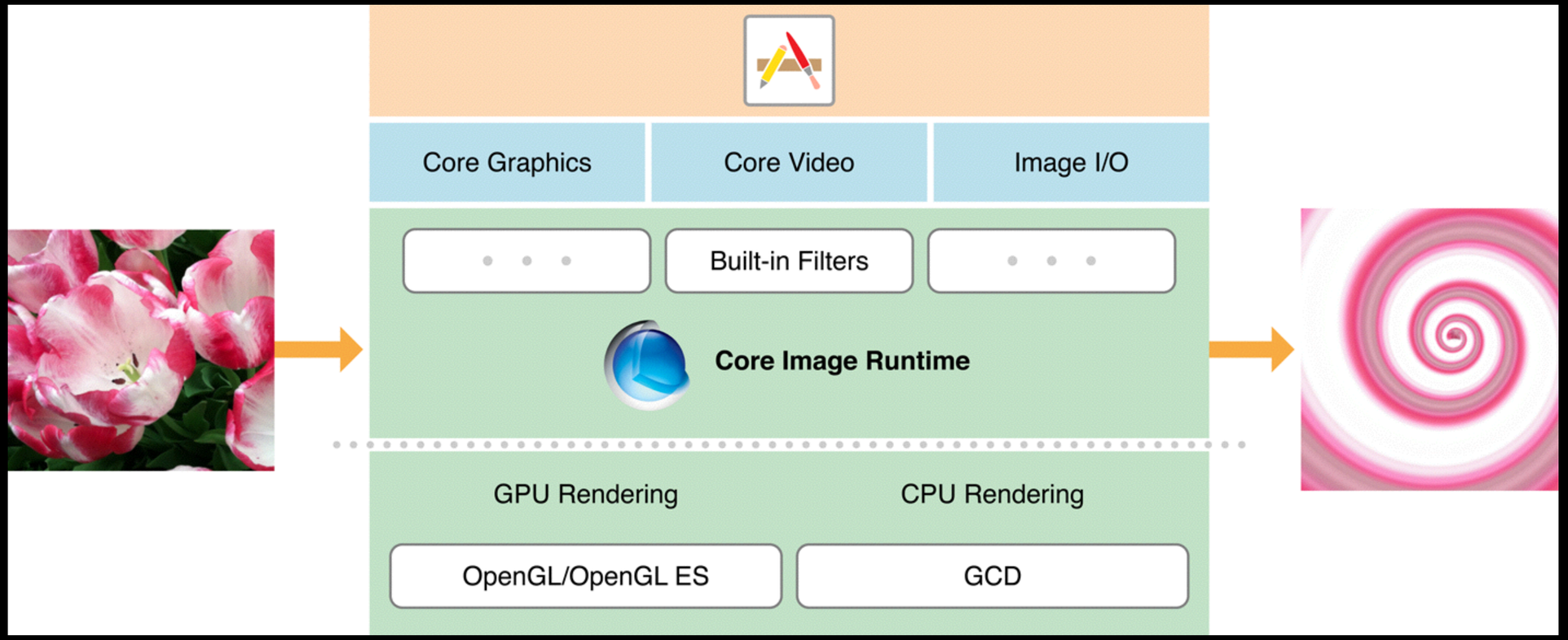
iOS Dev Accelerator
Week 3 Day 3

- Core Image
- AVFoundation

CoreImage

- “Core Image is an image processing and analysis technology designed to provide near real-time processing for still and video images”
- Can use either the GPU or CPU
- “Core Image hides the details of low-level graphic processing….You don’t need to know the details of OpenGL/ES to leverage the power of the GPU”

CoreImage


Core Graphics
Core Video
Image I/O
Built-in Filters
Core Image Runtime
GPU Rendering
CPU Rendering
OpenGL/OpenGL ES
GCD

CoreImage Offerings

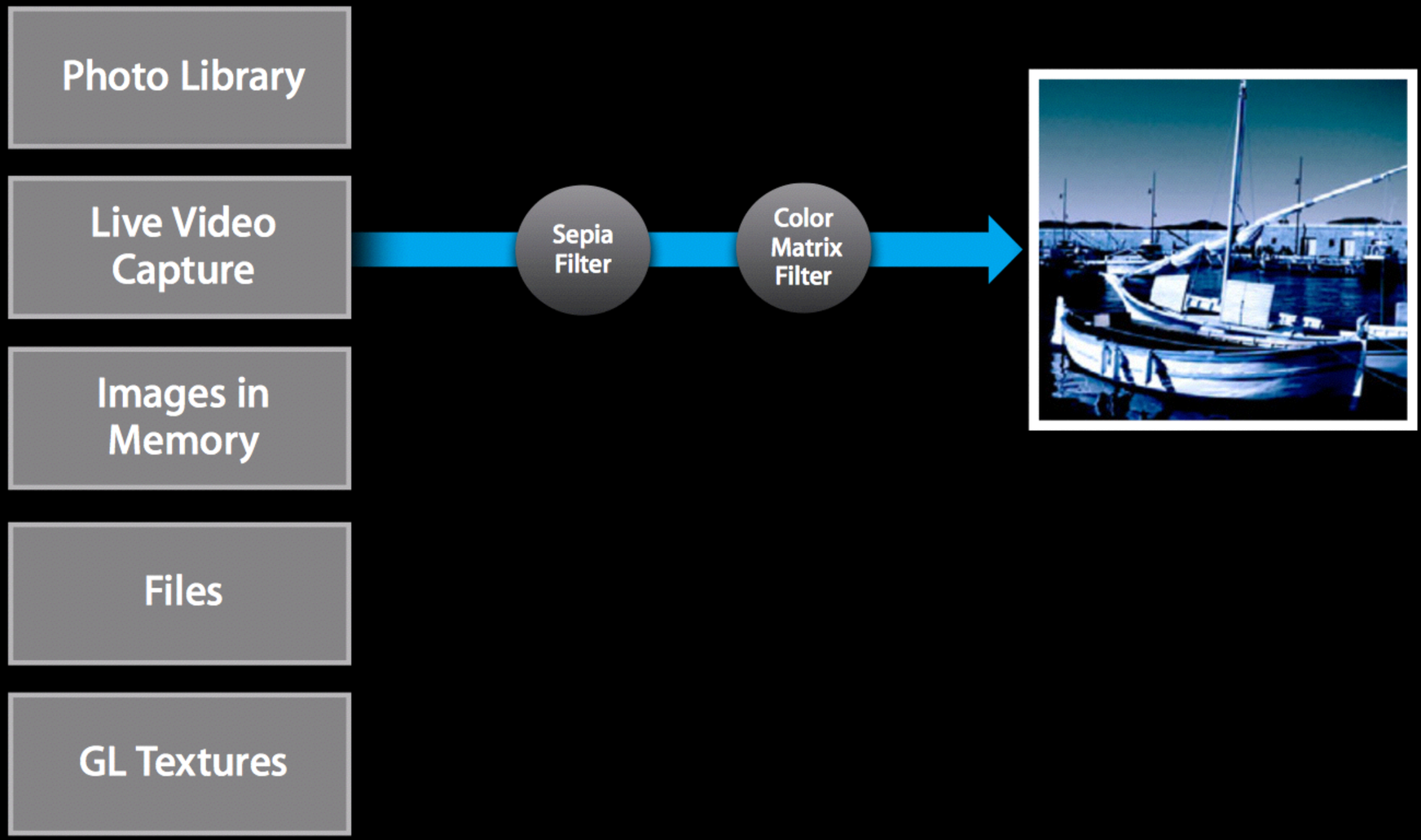
- Built-in image processing filters (90 on iOS)
- Feature detection capability
- Support for automatic image enhancement
- Ability to chain multiple filters together to create custom effects

guy using CoreImage

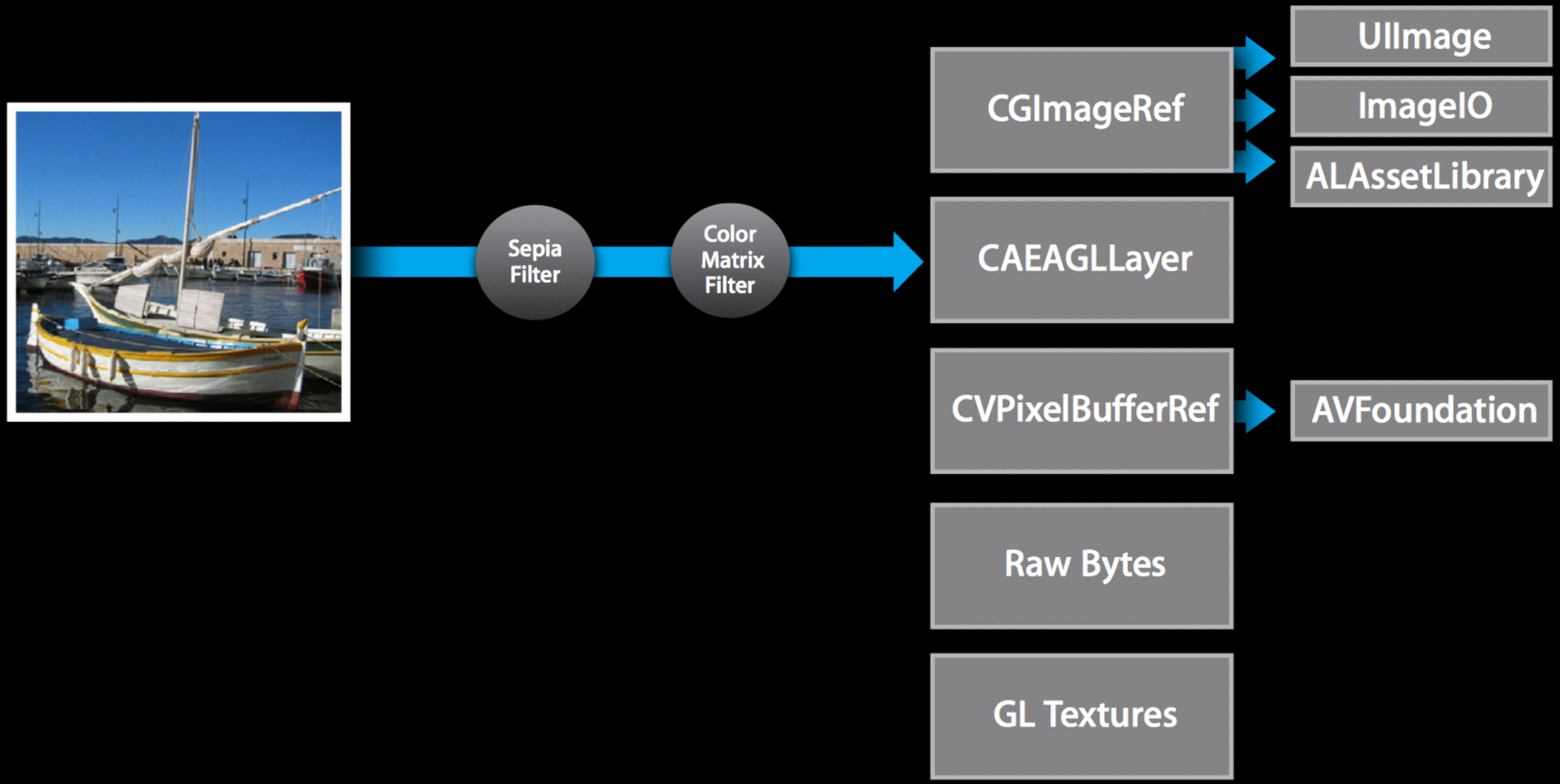
Filtering

- Filters applied on a per pixel basis
- Can be chained together

Filtering Inputs are Flexible



As are the Outputs



CIAdditionCompositing
CIAffineClamp
CIAffineTile
CIAffineTransform
CIBarsSwipeTransition
CIBlendWithMask
CIBloom
CICheckerboardGenerator
CICircleSplashDistortion
CICircularScreen
CIColorBlendMode
CIColorBurnBlendMode
CIColorControls
CIColorCube
CIColorDodgeBlendMode
CIColorInvert
CIColorMap
CIColorMatrix
CIColorMonochrome
CIColorPosterize
CIConstantColorGenerator
CICopyMachineTransition
CICrop
CIDarkenBlendMode
CIDifferenceBlendMode
CIDisintegrateWithMask
CIDissolveTransition
CIDotScreen
CIEightfoldReflectedTile
CIExclusionBlendMode
CIExposureAdjust
CIFalseColor
CIFlashTransition
CIFourfoldReflectedTile
CIFourfoldRotatedTile
CIFourfoldTranslatedTile
CIGammaAdjust
CIGaussianBlur
CIGaussianGradient
CIGlideReflectedTile
CIGloom
CIHardLightBlendMode
CIHatchedScreen
CIHighlightShadowAdjust
CIHoleDistortion
CIHueAdjust
CIHueBlendMode
CILanczosScaleTransform
CILightenBlendMode
CILightTunnel
CILinearGradient
CILineScreen
CILuminosityBlendMode
CIMaskToAlpha
CIMaximumComponent
CIMaximumCompositing
CIMinimumComponent
CIMinimumCompositing
CIModTransition
CIMultiplyBlendMode
CIMultiplyCompositing
CIOverlayBlendMode
CIPerspectiveTile
CIPerspectiveTransform
CIPinchDistortion
CIPixellate
CIRadialGradient
CIRandomGenerator
CISaturationBlendMode
CIScreenBlendMode
CISepiaTone
CISharpenLuminance
CISixfoldReflectedTile
CISixfoldRotatedTile
CISoftLightBlendMode
CISourceAtopCompositing
CISourceInCompositing
CISourceOutCompositing
CISourceOverCompositing
CIStarShineGenerator
CIStraightenFilter
CIStripesGenerator
CISwipeTransition
CITemperatureAndTint
CIToneCurve
CITriangleKaleidoscope
CITwelvefoldReflectedTile
CITwirlDistortion
CIUnsharpMask
CIVibrance
CIVignette
CIVortexDistortion
CIWhitePointAdjust

CIImage

- An Immutable object that represents the recipe for an Image
- Can represent a file from disk or the output of a CIFilter
- Multiple ways to create one:

```
var image  = CIImage(contentsOfURL: url)
```

```
var anotherImage = CIImage(image: UIImage())
```

Also has inits from Raw bytes,
NSData,CGImage,PixelBuffers,etc

CIFilter

- Mutable object that represents a filter
- Produces an output image based on the input

```
var filter = CIFilter(name: "CISepiaTone")
filter.setValue(image, forKey: kCIInputImageKey)
filter.setValue(NSNumber(float: 0.8), forKey: @"inputIntensity")
```

CIContext

- An object through which Core Image draws results
- Can be based on CPU or GPU

```
var context = CIContext(options: nil)
var result = filter.valueForKey(kCIOutputImageKey) as CIImage
var imageRef = context.createCGImage(result, fromRect: result.extent()) as
    CGImageRef
var finalImage = UIImage(CGImage: imageRef)
```

CIKernel

- Custom Core Image Kernels are supported as of iOS 8
  - Built using Core Image Kernel language
  - similar to OpenGL ES2.0 shaders
  - used for advanced Core Image work
  - docs not published yet

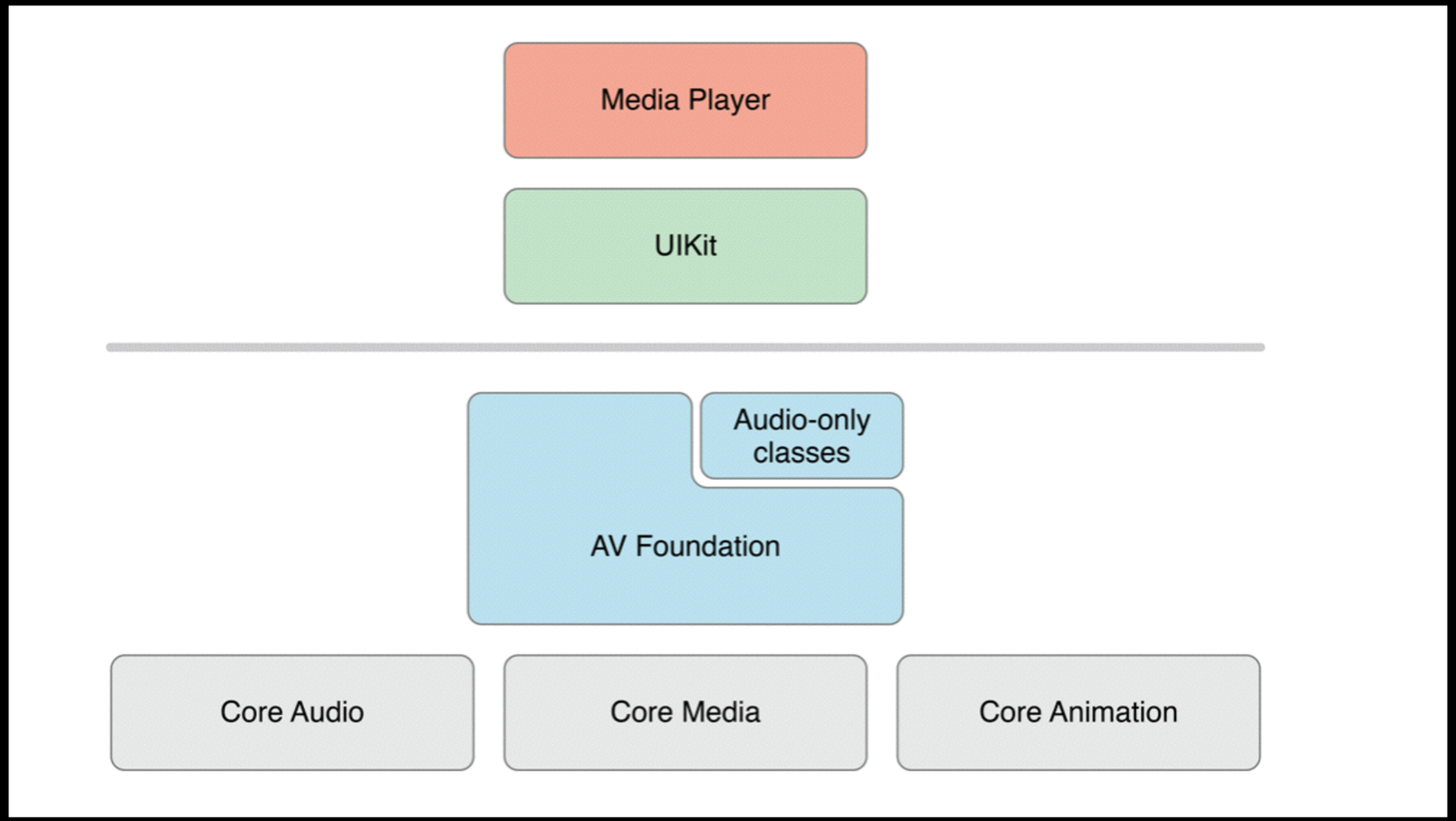
IBDesignable

- Create custom interface builder components
- Designable views must be in a separate framework
- Prefix your class declaration with @IBDesignable

IBInspectable

- Exposes attributes in `InterfaceBuilder`
- Prefix your variable declaration with `@IBInspectable`
- Add `didSet` method to the variable
- Set self.property, then update the view based on the changes

AVFoundation

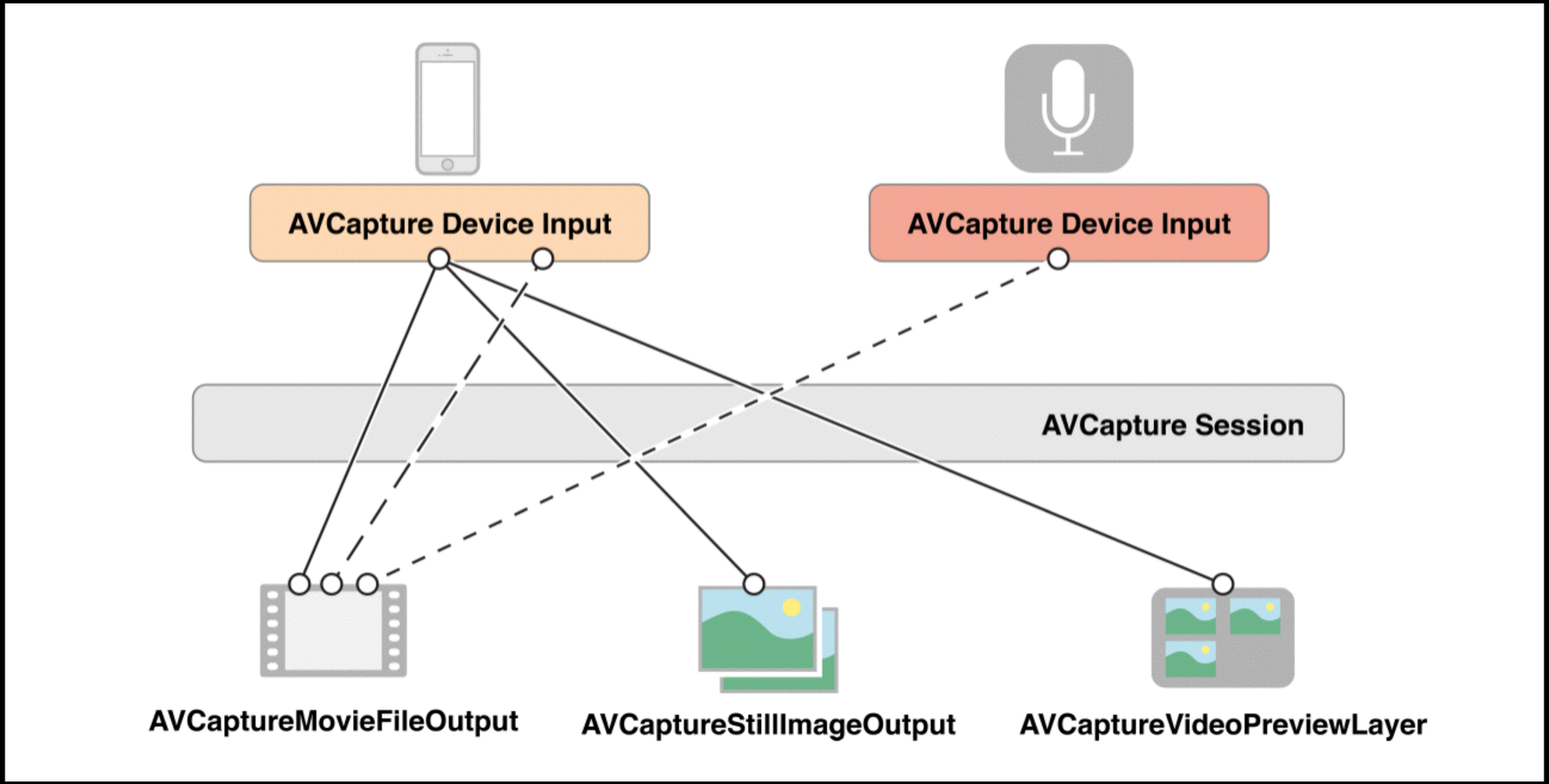


- A framework used to play and create time-based audiovisual media.

AVFoundation Assets

- The primary model class that AVFoundation uses to represent media is AVAsset.
- AVAsset is an aggregated representation of one or more pieces of media data.
- Provides info like the title,duration,natural size,etc.
- Each piece of media data inside the asset is considered a track.
- An asset or track that has been initialized may not ready to be used right away, so the API is highly asynchronous using callbacks.

AVFoundation Capturing



- To manage the capture of media, you create objects to represent inputs and outputs.
- You then use an instance of AVCaptureSession to coordinate the flow of data between them.

AVFoundation Capturing

- You will need the following objects setup for capture:
  - An an instance of AVCaptureDevice to represent the input device, like the phones camera or mic.
  - An instance of AVCaptureInput to configure the ports from the iput device.
  - An instance of AVCaptureOutput to manage the output to a movie file or still image.
  - An instance of AVCaptureSession to coordinate the flow of data from input to output.
  - An instance of AVCaptureVideoPreviewLayer to show the user a preview of what the camera is recording.

AVCaptureDevice

- Represents a physical capture device and the properties associated with the device.
- An instance of AVCapture devices allows you to configure the underlying device.
- Provides input data to an AVCaptureSession

AVDeviceInput

- A concrete sub-class of AVCaptureInput.
- Used to capture data from an AVCapture Device.
- initWithDevice:Error:

AVCaptureSession

- You use this class to coordinate the flow of data from input devices to outputs.
- use addInput: and addOutput: methods to add those streams.
- tell a session to startRunning() when everything is configured.
- Run all session setup and startRunning on a background queue because it is potentially blocking and we want to keep the interface responsive to the user.

AVMetadataObject

- Contains additional data about the image/frame of video
- AVMetadataFaceObject
  - defines features for a single detected face
  - properties include rollAngle, yawAngle, & faceID
- AVMetadataMachineReadableCodeObject
  - features of a detected 1d or 2d barcode