

Как мы делали новую iOS-камеру ВКонтакте и VK Клипов



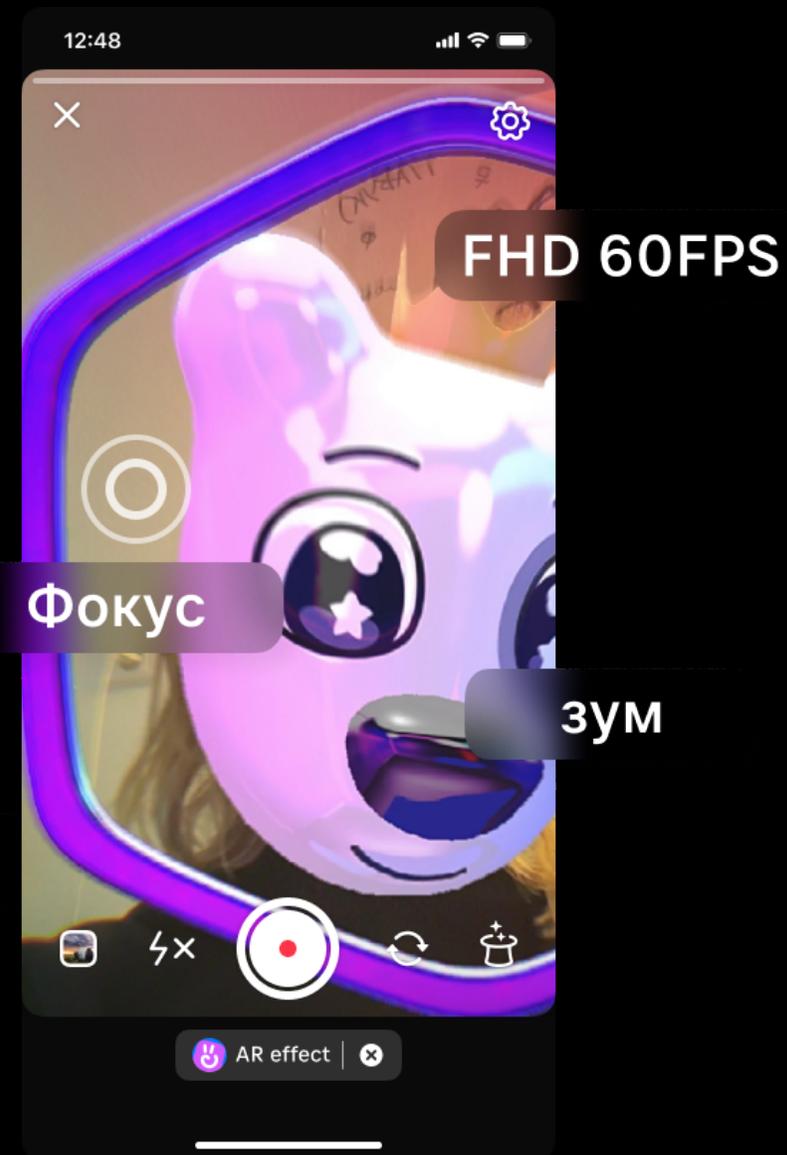
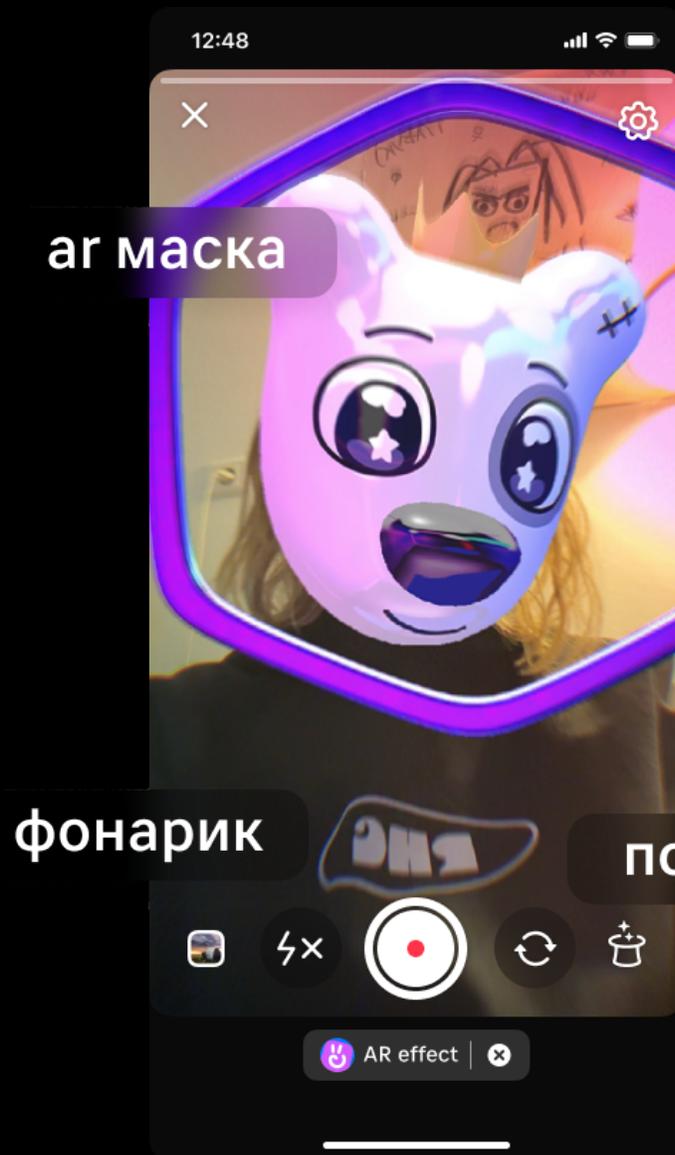


Никита Савинов

@actorFromCalifornia

iOS Team Lead – VK Клипы

Камера ВКонтакте и VK Клипов



О чем этот доклад

Примитивная камера 1.0

Начинаем прокачивать камеру

- Zoom, Torch, Focus & Exposition
- Фронтальная/Задняя камеры
- Кастомные фильтры для кадра
- Camera FPS
- Camera FOV

Качаем дальше. AR

Архитектура

Примитивная камера 1.0

Что мы получаем из коробки

`AVCaptureSession`

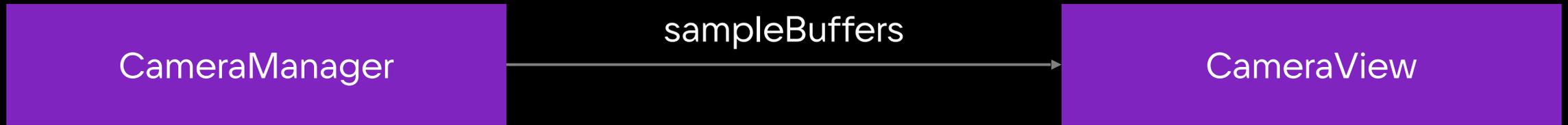
– `AVCaptureInput`

– `AVCaptureOutput`

`AVCaptureVideoPreviewLayer`



Примитивная камера 1.0



Начинаем
прокачивать камеру

Zoom, Torch, Focus & Exposition

1. `device.videoZoomFactor`
2. `device.torchMode`
3. `device.focusMode`
4. `device.exposureMode`

При изменении:

1. Проверки на доступность
2. Обернуть в
 - `lockForConfiguration`
 - `unlockForConfiguration`

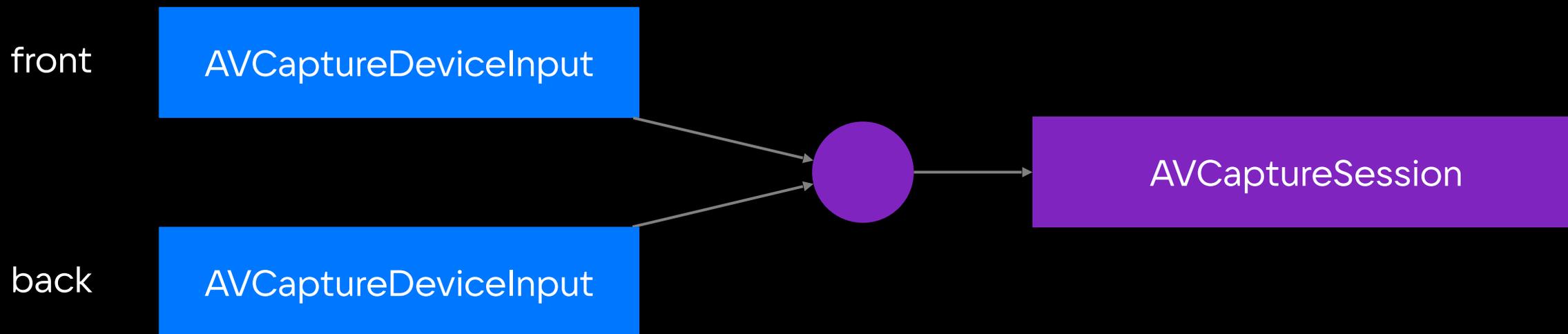


Хотим автоматически менять Focus & Exposition?

```
NotificationCenter.default.addObserver(  
    self,  
    selector: #selector(subjectAreaChanged),  
    name: .AVCaptureDeviceSubjectAreaDidChange,  
    object: currentCamera.cameraDevice  
)
```

Смена камер
(фронтальная / задняя)

Фронтальная/Задняя камеры



Фронтальная/Задняя камеры

При переключении:

1. Взять блокировку на девайс и сессию
2. Удалить инпут текущего входа (`removeInput`)
3. Добавить инпут нового входа (`addInput`)
4. Отпустить блокировку



А с какой камеры кадр?

Проблема:

AVCaptureVideoDataOutputSampleBufferDelegate

не всегда отдает корректную позицию камеры в

connection.inputPorts.first!.sourceDevicePosition

А с какой камеры кадр?

CMSampleBuffer

CMTime

CMAttachments

CMFormatDescription

CVPixelBuffer/
CVBlockBuffer

А с какой камеры кадр?

EXIF – Exchangeable Image File Format

```
let metadata = CMGetAttachment(  
    sampleBuffer, key: "{Exif}" as CFString, attachmentModeOut: nil  
) as? NSDictionary
```

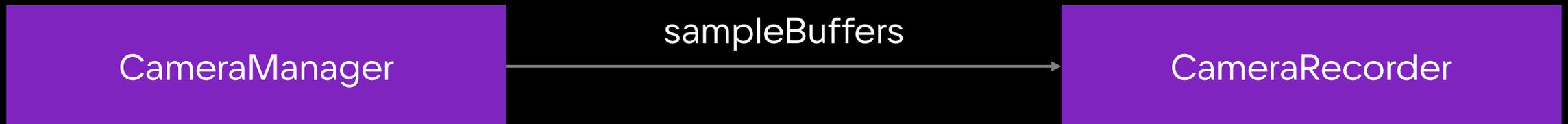
А с какой камеры кадр?

```
if
    let lensModel = metadata?.value(forKey: "LensModel") as? String
    {
        if lensModel.contains("front") {
            return .front
        } else if lensModel.contains("back") {
            return .back
        }
    }
}
```

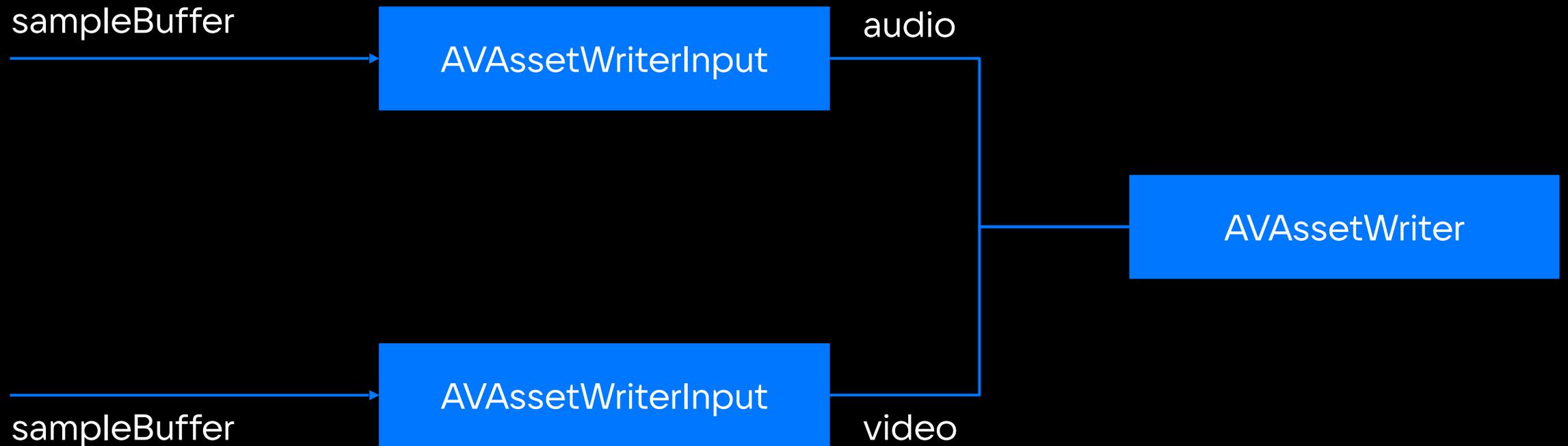
Запись с камеры

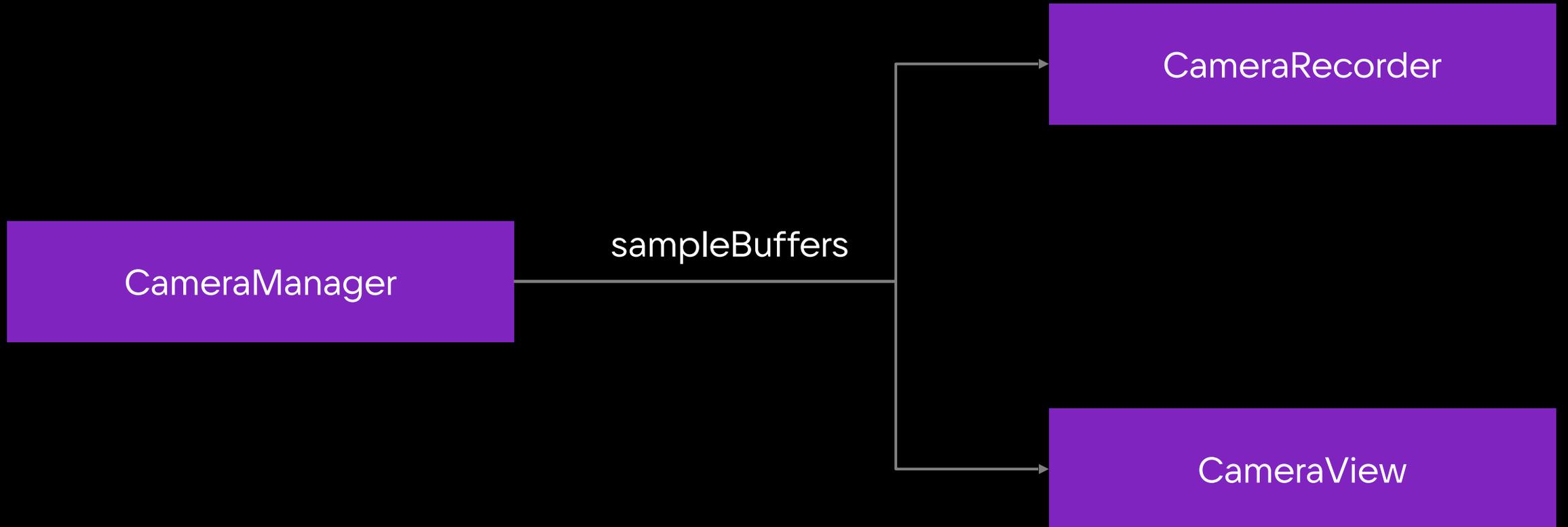
Учимся записывать камеру

1. Создаем AVAssetWriter
2. Настраиваем входы
3. Коннектим входы к AVAssetWriter
4. Запускаем сессию и записываем буфера с камеры



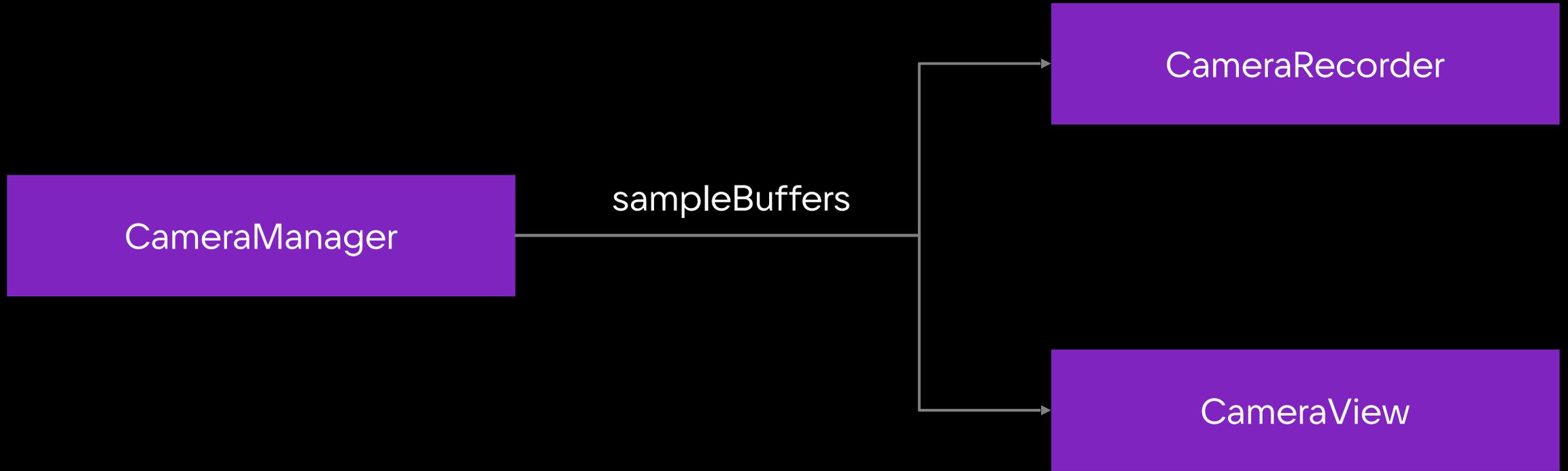
Camera Recorder



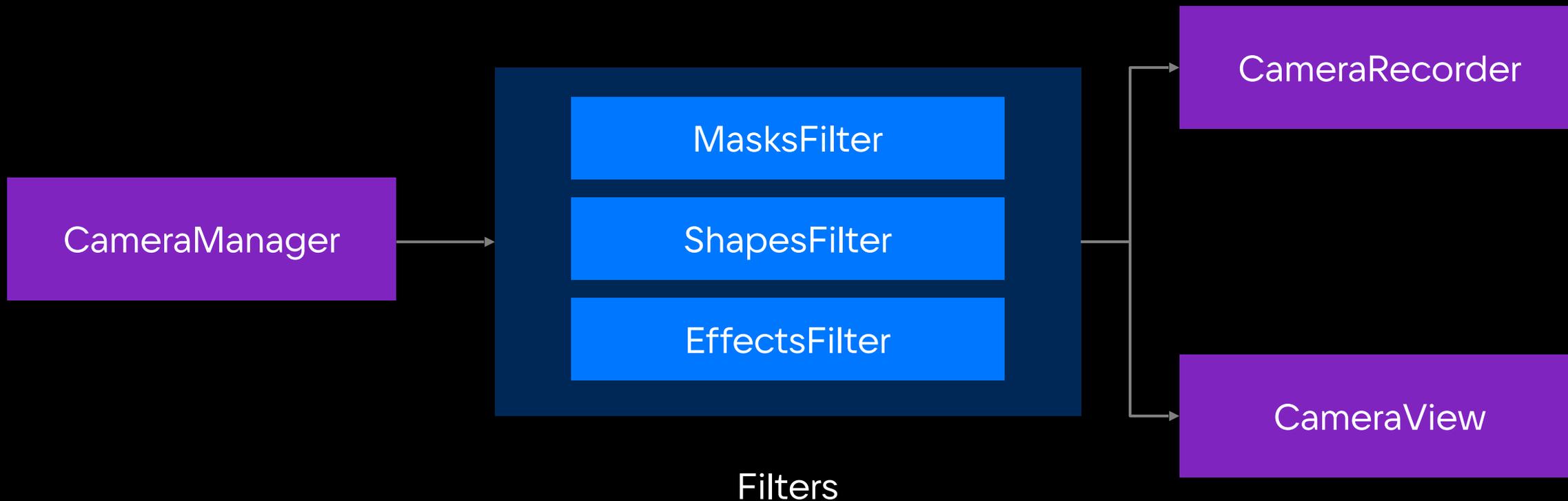


Кастомные фильтры для кадра

Кастомные фильтры для кадра



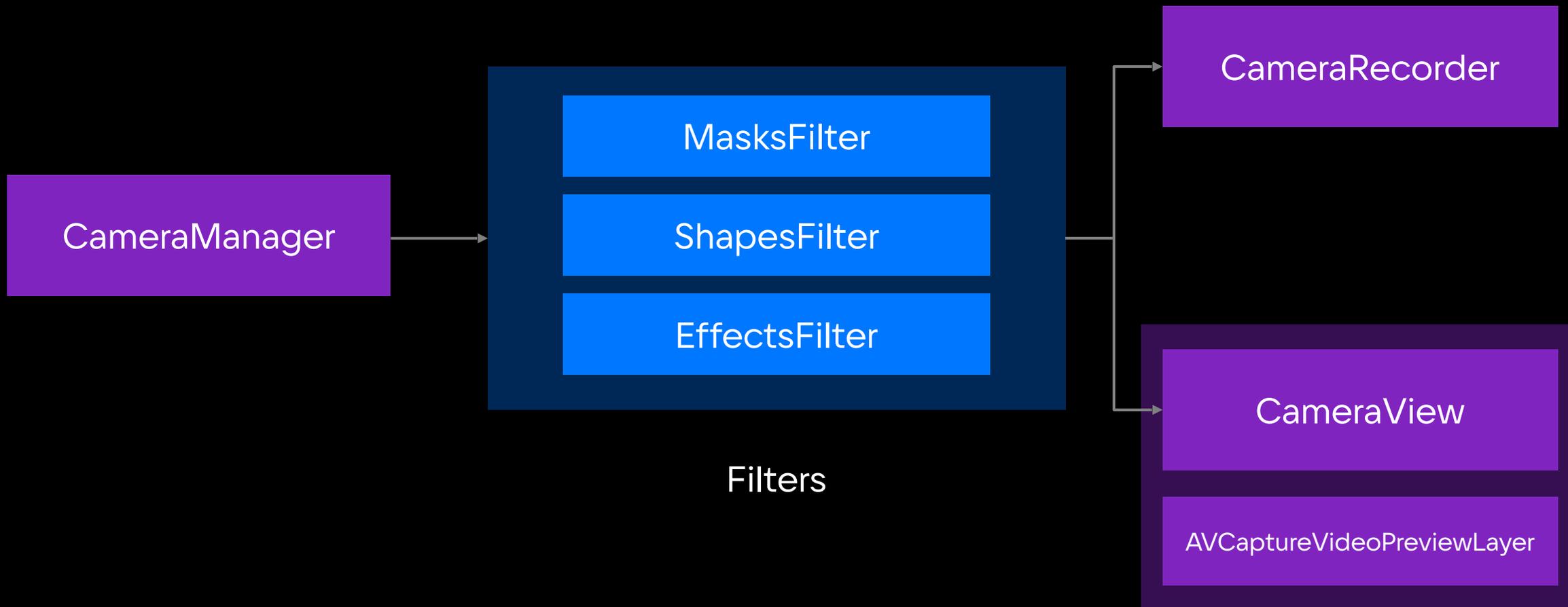
Кастомные фильтры для кадра



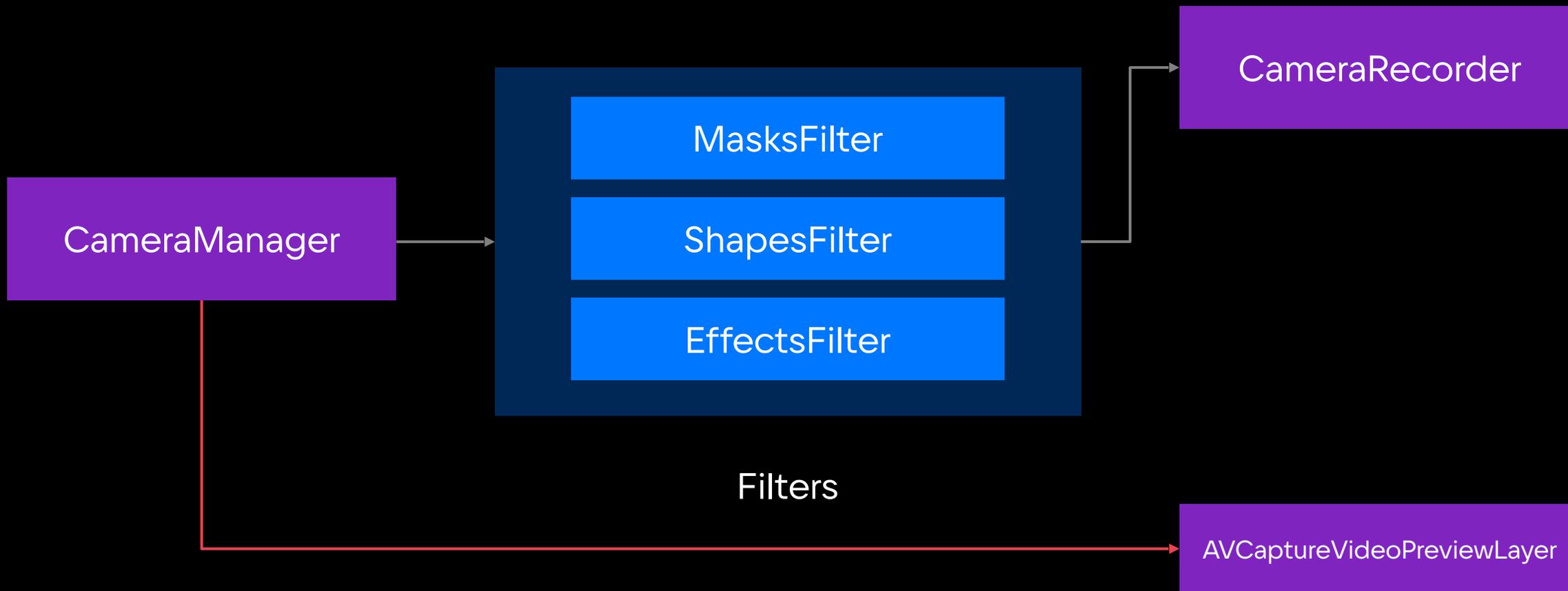
Кастомные фильтры для кадра

```
public protocol VCMCameraFrameProcessing {  
  
    var isReady: Bool { get }  
  
    func process(  
        sampleBuffer: CMSampleBuffer,  
        metaInfo: VCMCameraFilterMetaInfo,  
        completion: (CMSampleBuffer) -> Void  
    )  
}
```

Проблема отображения



Проблема отображения



Решение проблемы отображения



1. videoOrientation
2. mirroring (front / back camera)

Смена FPS

Как менять FPS в камере?

Текущий формат для съемки в камере:

```
let activeFormat = device.activeFormat
```

Как менять FPS в камере?

Текущий формат для съемки в камере:

```
let activeFormat = device.activeFormat
```

У каждого формата есть свой поддерживаемый FPS диапазон:

```
let rateRanges = activeFormat.videoSupportedFrameRateRanges
```

```
_ = rateRanges.first?.minFrameRate
```

```
_ = rateRanges.first?.maxFrameRate
```

Мой FPS не поддерживается 😓

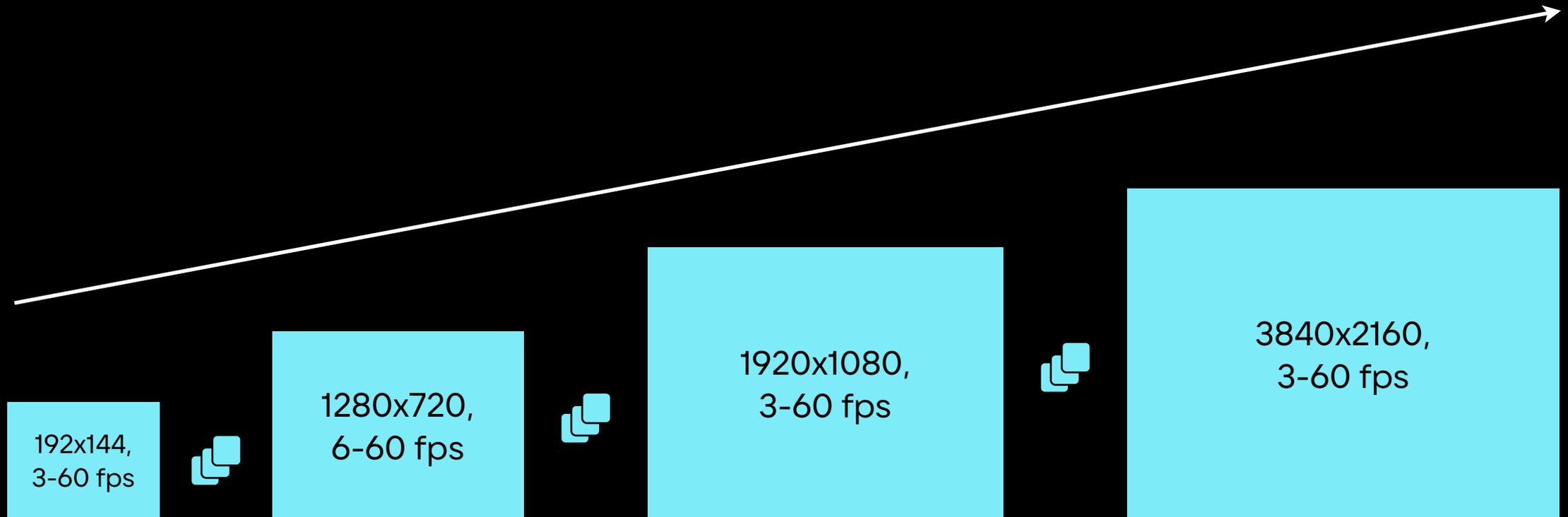
(11db) po device.activeFormat

```
'vide'/'420v' 1920x1080, { 1- 30 fps}, photo  
  dims:{1920x1080,4224x2376}, fov:69.654, supports vis  
  (max strength:Low), max zoom:16.00 (upscales @2.00), AF  
  System:2, ISO:32.0-3072.0, SS:0.000018-1.000000,  
  supports HDR, supports multicom, supports high photo  
  quality
```

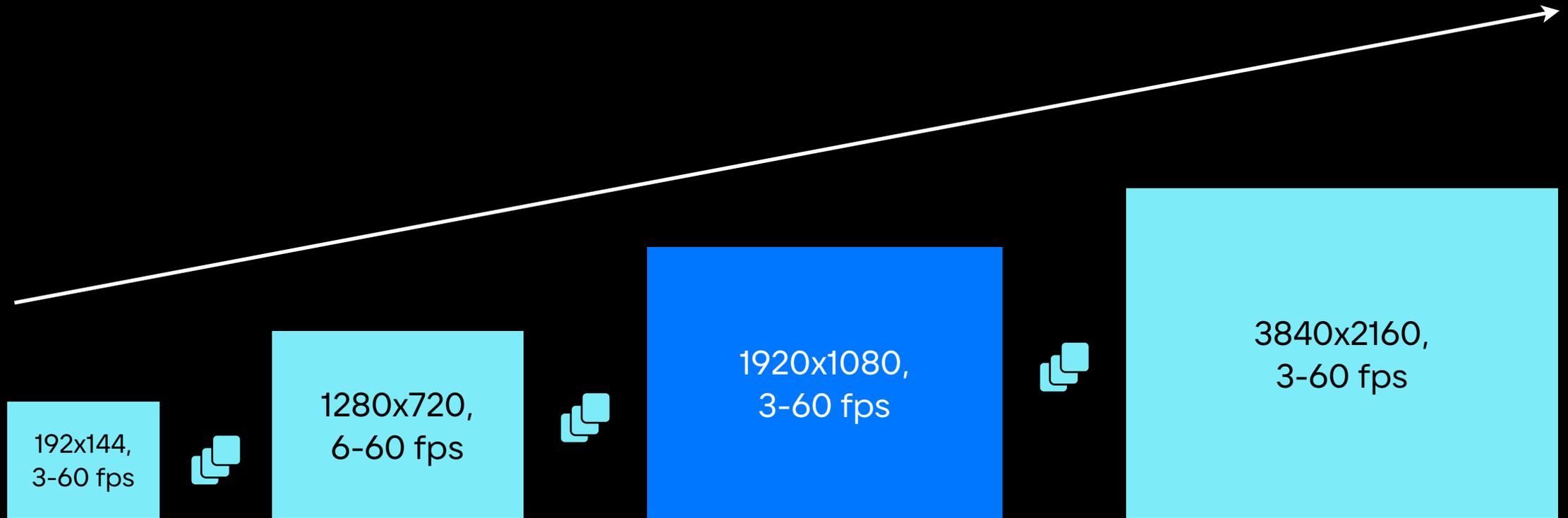
(11db) po device.formats.count

48

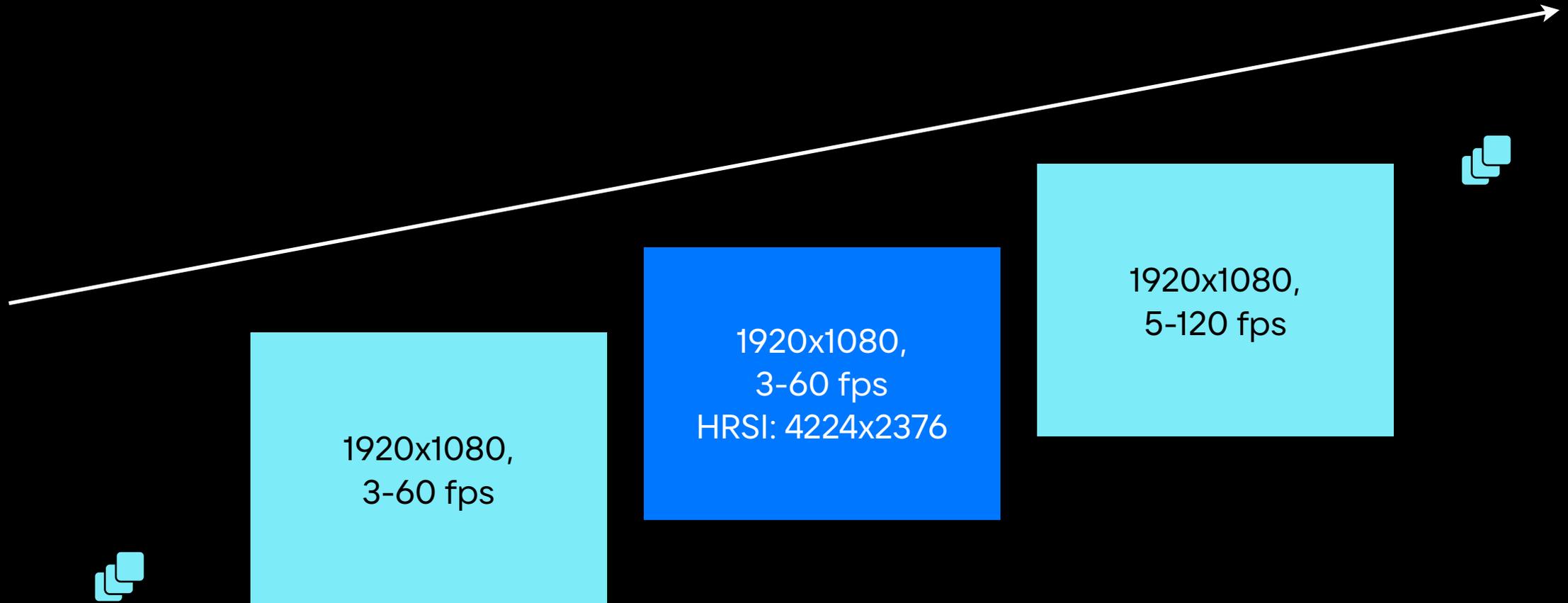
Προ AVCaptureDeviceFormat



Προ AVCaptureDeviceFormat



Προ AVCaptureDeviceFormat



Как менять FPS в камере?

```
device.formats.forEach { format in
    let rateRanges = format.videoSupportedFrameRateRanges
    let supportedRate = !rateRanges.contains {
        Int($0.maxFrameRate) > targetFps.rawValue
    }

    if supportedRate {
        preferredFormat = format
    }
}
```

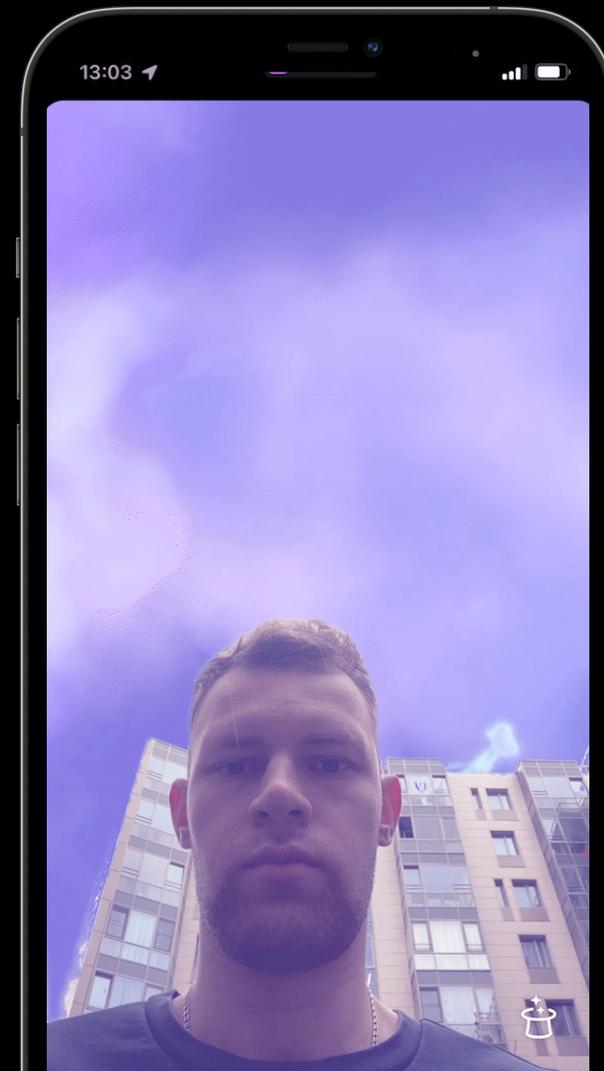
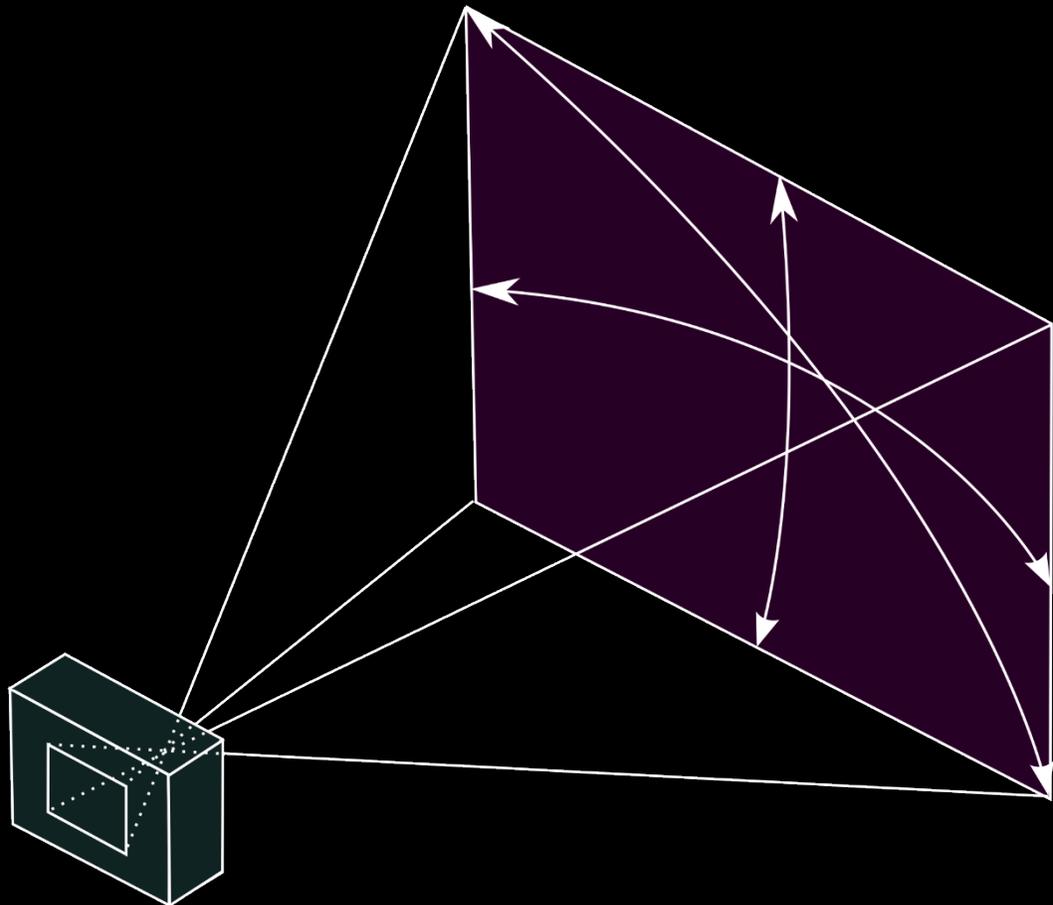
Как менять FPS в камере?

```
device.formats.forEach { format in
    let rateRanges = format.videoSupportedFrameRateRanges
    let supportedRate = !rateRanges.contains {
        Int($0.maxFrameRate) > targetFps.rawValue
    }

    if supportedRate {
        preferredFormat = format
    }
}
```

Field Of View

Что такое Camera FOV и зачем мне это?



Camera FOV

```
var newFOV = currentCamera.cameraDevice.activeFormat
    .geometricDistortionCorrectedVideoFieldOfView

if zoomFactor != 1 {
    newFOV = newFOV * Float.pi / 180 // радианы
    newFOV = 2 * atanf(tanf(newFOV / 2) / Float(zoomFactor))
    newFOV = newFOV * 180 / Float.pi // градусы
}
```

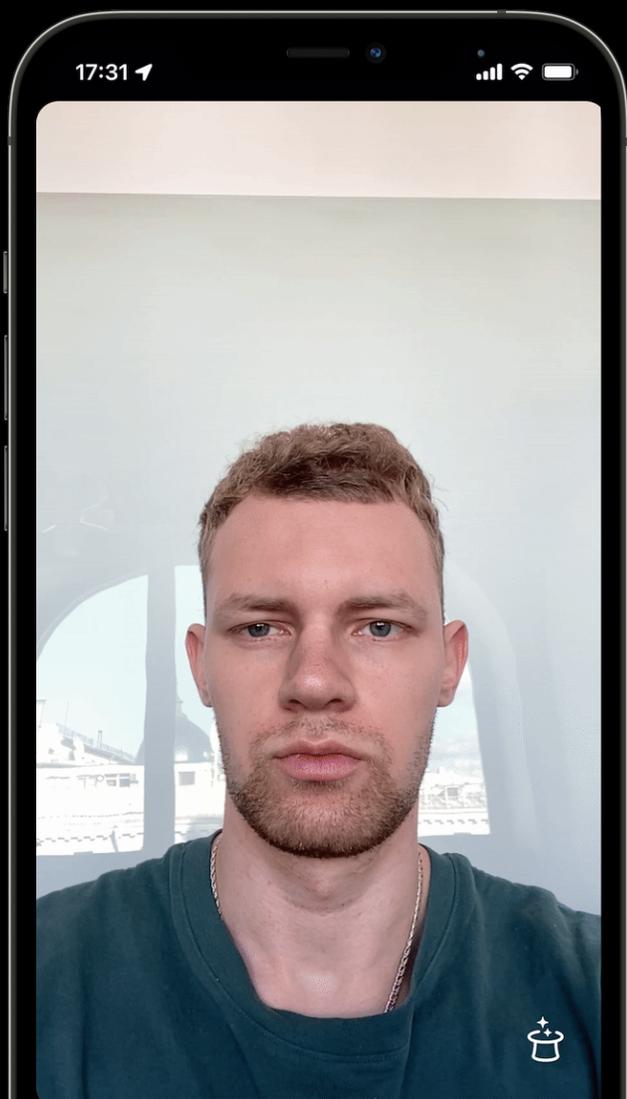
Camera FOV

```
var newFOV = currentCamera.cameraDevice.activeFormat
    .geometricDistortionCorrectedVideoFieldOfView

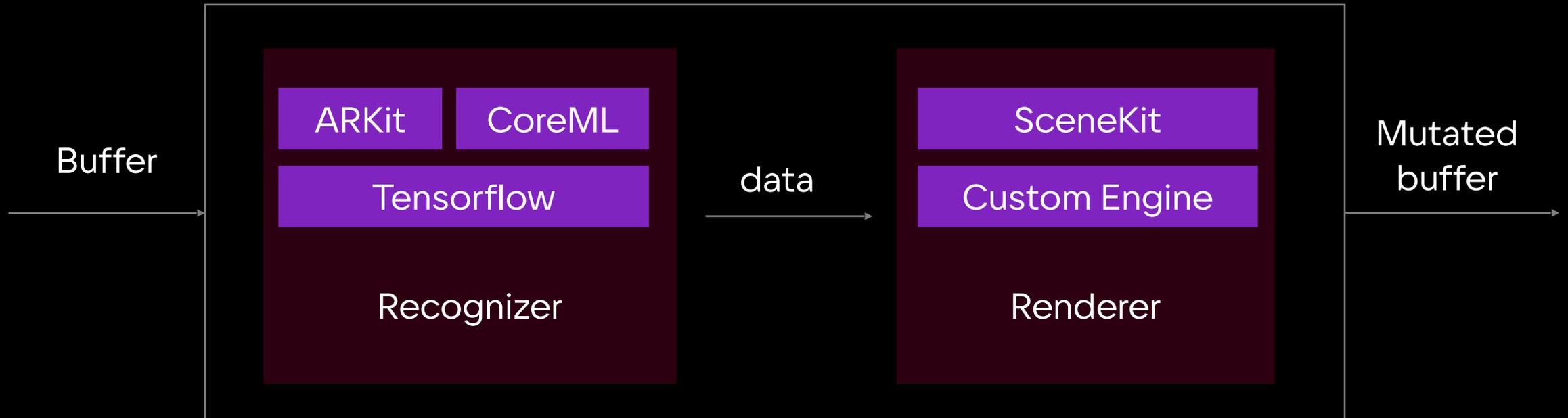
if zoomFactor != 1 {
    newFOV = newFOV * Float.pi / 180 // радианы
    newFOV = 2 * atanf(tanf(newFOV / 2) / Float(zoomFactor))
    newFOV = newFOV * 180 / Float.pi // градусы
}
```

Качаем камеру
дальше. AR

Задача: наложить объекты на части лица



Effects Filter



Custom ML vs ARKit

Tensorflow (mediapipe)

CoreML (mediapipe)

ARKit

Custom ML vs ARKit

	Tensorflow (mediapipe)	CoreML (mediapipe)	ARKit
Скорость (1) (2)	23.67 ms (11 pro max) 54.84 ms (6s)	17.03 ms (11 pro max) 49.31 ms (6s)	~fps (11 pro max) –

(1) – avg time (ms) = all time / launches count

(2) – ЗАВИСИТ ОТ СЛОЖНОСТИ МОДЕЛИ

Custom ML vs ARKit

	Tensorflow (mediapipe)	CoreML (mediapipe)	ARKit
Скорость (1) (2)	23.67 ms (11 pro max) 54.84 ms (6s)	17.03 ms (11 pro max) 49.31 ms (6s)	~fps (11 pro max) –
Количество точек (2)	468	468	1220

(1) – avg time (ms) = all time / launches count

(2) – ЗАВИСИТ ОТ СЛОЖНОСТИ МОДЕЛИ

Custom ML vs ARKit

	Tensorflow (mediapipe)	CoreML (mediapipe)	ARKit
Скорость (1) (2)	23.67 ms (11 pro max) 54.84 ms (6s)	17.03 ms (11 pro max) 49.31 ms (6s)	~fps (11 pro max) –
Количество точек (2)	468	468	1220
Поддержка	iPhone *	iPhone *	iPhone X+ (TrueDepth camera embedded)

(1) – avg time (ms) = all time / launches count

(2) – ЗАВИСИТ ОТ СЛОЖНОСТИ МОДЕЛИ

Запускаем ARKit

Конфигурация ARSession

```
session = ARSession()
```

```
backCameraConfiguration = ARWorldTrackingConfiguration()  
backCameraConfiguration.worldAlignment = .camera
```

```
frontCameraConfiguration = ARFaceTrackingConfiguration()  
frontCameraConfiguration.maximumNumberOfTrackedFaces = 2
```

```
// Run the view's session  
session.run(backCameraConfiguration)
```

Конфигурация ARSession

```
session = ARSession()

backCameraConfiguration = ARWorldTrackingConfiguration()
backCameraConfiguration.worldAlignment = .camera

frontCameraConfiguration = ARFaceTrackingConfiguration()
frontCameraConfiguration.maximumNumberOfTrackedFaces = 2

// Run the view's session
session.run(backCameraConfiguration)
```

Конфигурация ARSession

```
session = ARSession()

backCameraConfiguration = ARWorldTrackingConfiguration()
backCameraConfiguration.worldAlignment = .camera

frontCameraConfiguration = ARFaceTrackingConfiguration()
frontCameraConfiguration.maximumNumberOfTrackedFaces = 2

// Run the view's session
session.run(backCameraConfiguration)
```

Поворот при отображении

ARSession

ARWorldTrackingConfiguration

View (Metal Renderer)



Поворот при отображении

```
// 1. сдвиг центра изображения к точке (0, 0)
transform = transform.translatedBy(x: CGFloat(destWidth) / 2, y: CGFloat(destHeight) / 2)

// 2. отражение по горизонтали
if flipHorizontally {
    transform = transform.scaledBy(x: 1, y: -1)
}

// 3. поворот, если требуется
if rotationFactor != 0 {
    transform = transform.rotated(by: CGFloat(rotationFactor) * CGFloat.pi / 2)
}

// 4. перемещение изображения в центр целевого кадра
transform = transform.translatedBy(x: -CGFloat(srcWidth) / 2, y: -CGFloat(srcHeight) / 2)
```

Поворот при отображении

```
// 1. сдвиг центра изображения к точке (0, 0)
transform = transform.translatedBy(x: CGFloat(destWidth) / 2, y: CGFloat(destHeight) / 2)

// 2. отражение по горизонтали
if flipHorizontally {
    transform = transform.scaledBy(x: 1, y: -1)
}

// 3. поворот, если требуется
if rotationFactor != 0 {
    transform = transform.rotated(by: CGFloat(rotationFactor) * CGFloat.pi / 2)
}

// 4. перемещение изображения в центр целевого кадра
transform = transform.translatedBy(x: -CGFloat(srcWidth) / 2, y: -CGFloat(srcHeight) / 2)
```

Поворот при отображении

```
// 1. сдвиг центра изображения к точке (0, 0)
transform = transform.translatedBy(x: CGFloat(destWidth) / 2, y: CGFloat(destHeight) / 2)

// 2. отражение по горизонтали
if flipHorizontally {
    transform = transform.scaledBy(x: 1, y: -1)
}

// 3. поворот, если требуется
if rotationFactor != 0 {
    transform = transform.rotated(by: CGFloat(rotationFactor) * CGFloat.pi / 2)
}

// 4. перемещение изображения в центр целевого кадра
transform = transform.translatedBy(x: -CGFloat(srcWidth) / 2, y: -CGFloat(srcHeight) / 2)
```

Поворот при отображении

```
// 1. сдвиг центра изображения к точке (0, 0)
transform = transform.translatedBy(x: CGFloat(destWidth) / 2, y: CGFloat(destHeight) / 2)

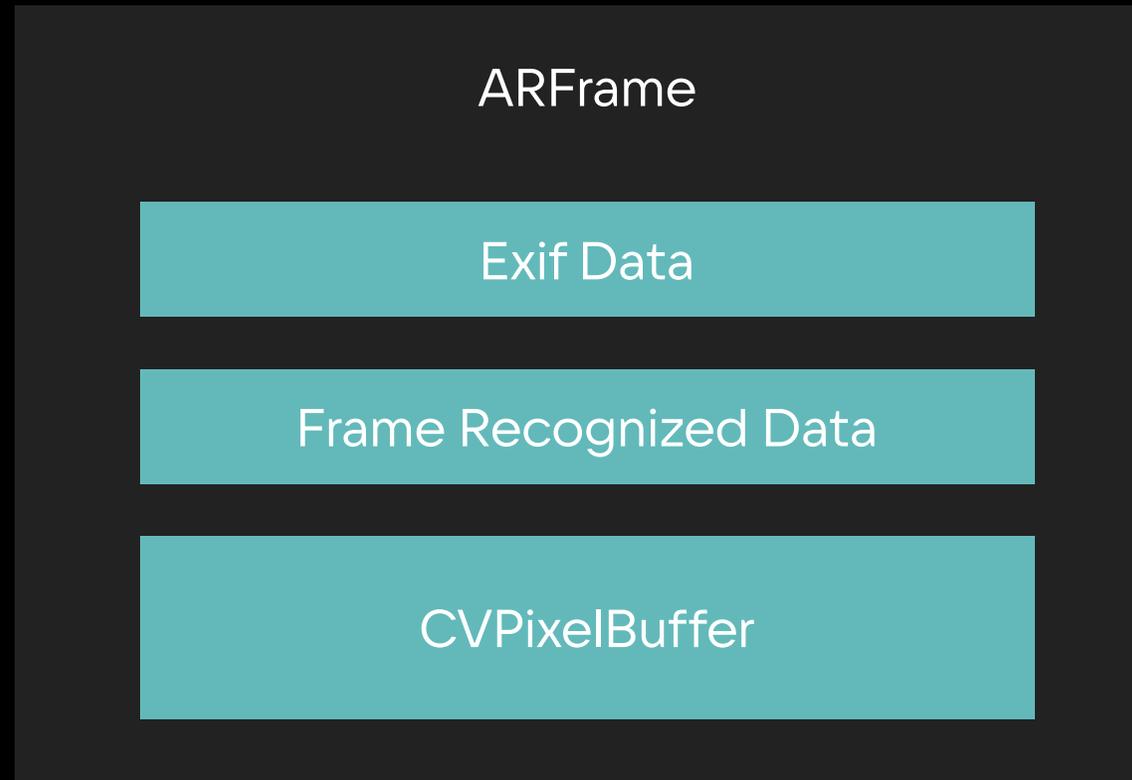
// 2. отражение по горизонтали
if flipHorizontally {
    transform = transform.scaledBy(x: 1, y: -1)
}

// 3. поворот, если требуется
if rotationFactor != 0 {
    transform = transform.rotated(by: CGFloat(rotationFactor) * CGFloat.pi / 2)
}

// 4. перемещение изображения в центр целевого кадра
transform = transform.translatedBy(x: -CGFloat(srcWidth) / 2, y: -CGFloat(srcHeight) / 2)
```

Получение точек лица

ARSession + Face Tracking



ARSession + Face Tracking

Фильтруем лицевые ноды

```
let faceAnchors = frame.anchors
    .compactMap { $0 as? ARFaceAnchor }
    .filter(\.isTracked)
```

ARSession + Face Tracking

Фильтруем лицевые ноды

```
let faceAnchors = frame.anchors
    .compactMap { $0 as? ARFaceAnchor }
    .filter(\.isTracked)
```

Получаем точки лица

```
for anchor in faceAnchors {
    let positions = anchor.geometry.vertices.flatMap { [$0.x, $0.y, $0.z] }
}
```

ARSession + Face Tracking

ARKit

Recognizer

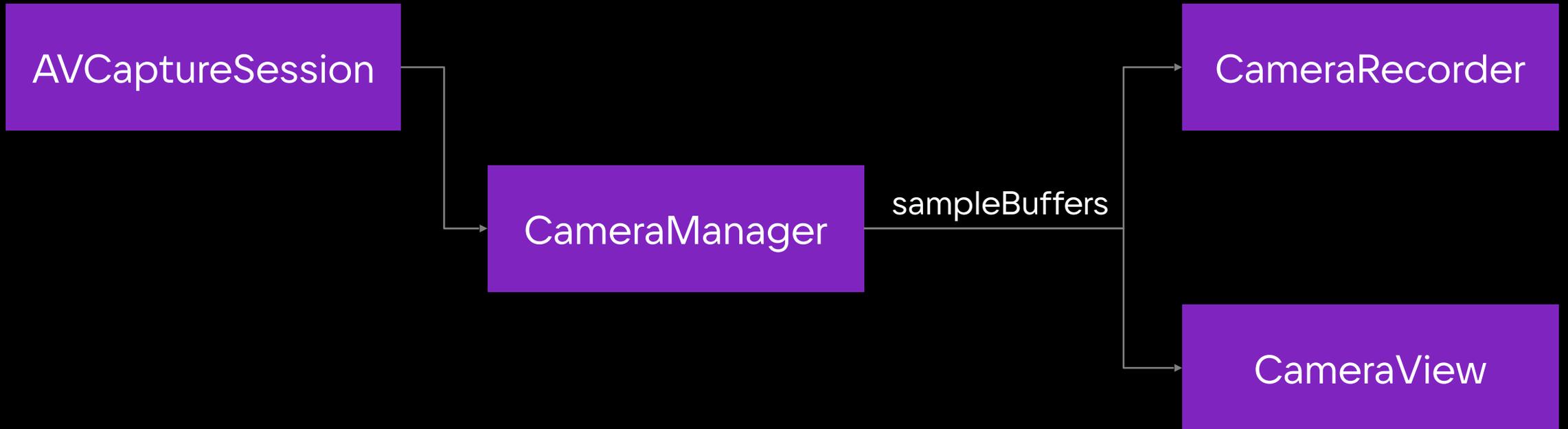
1. Face Anchors Points Positions
2. Bounding Boxes
3. Projection Matrix
4. Camera Matrix
5. ...

Custom
Engine

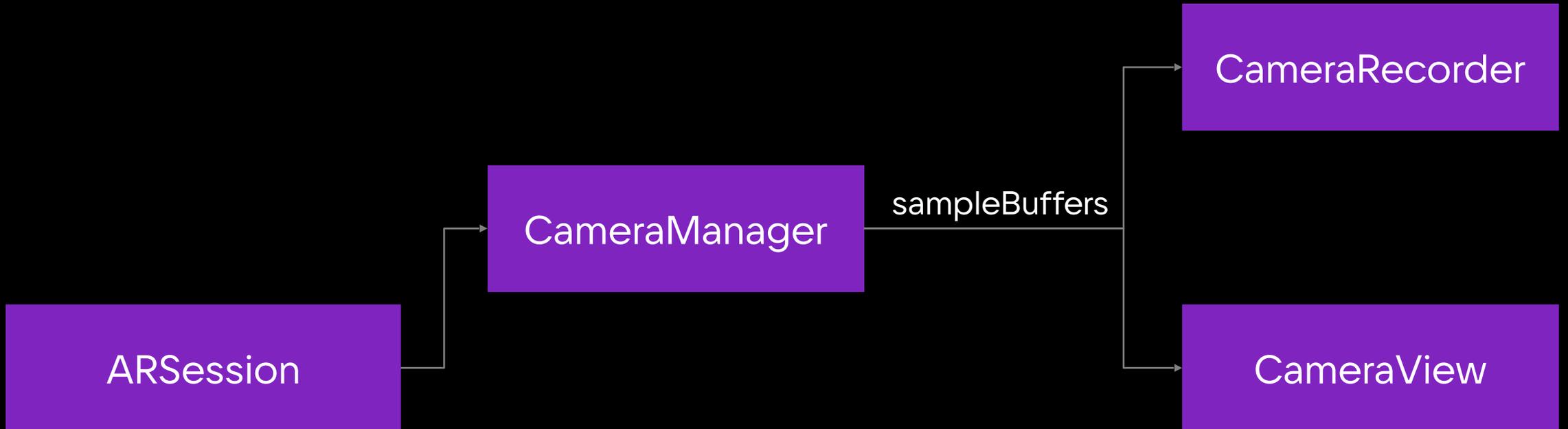
Renderer

Архитектура

Изначальное устройство камеры



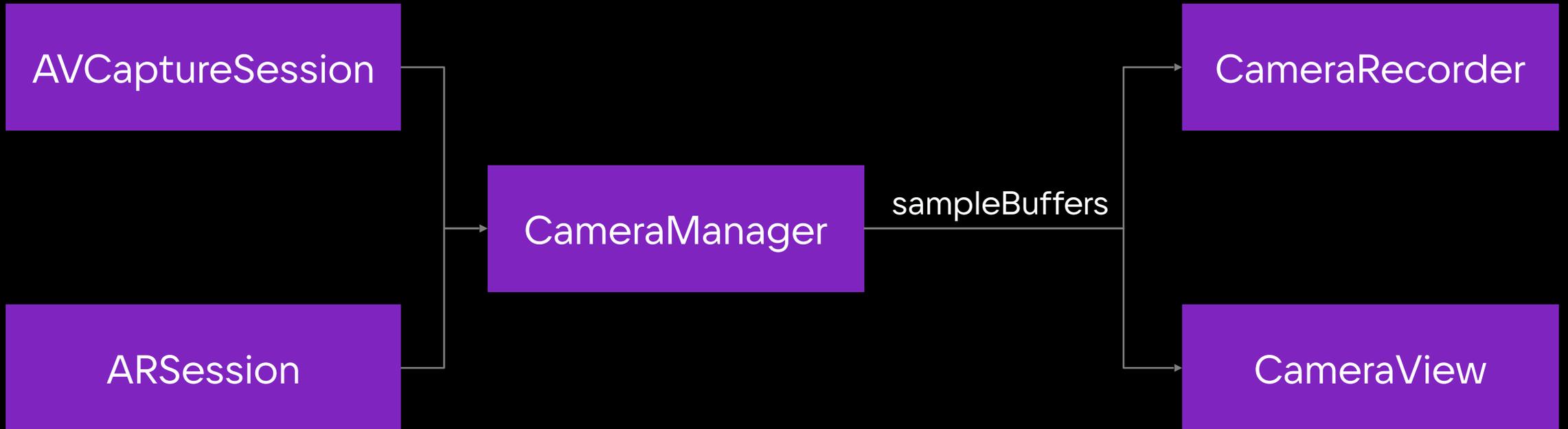
Текущее устройство камеры



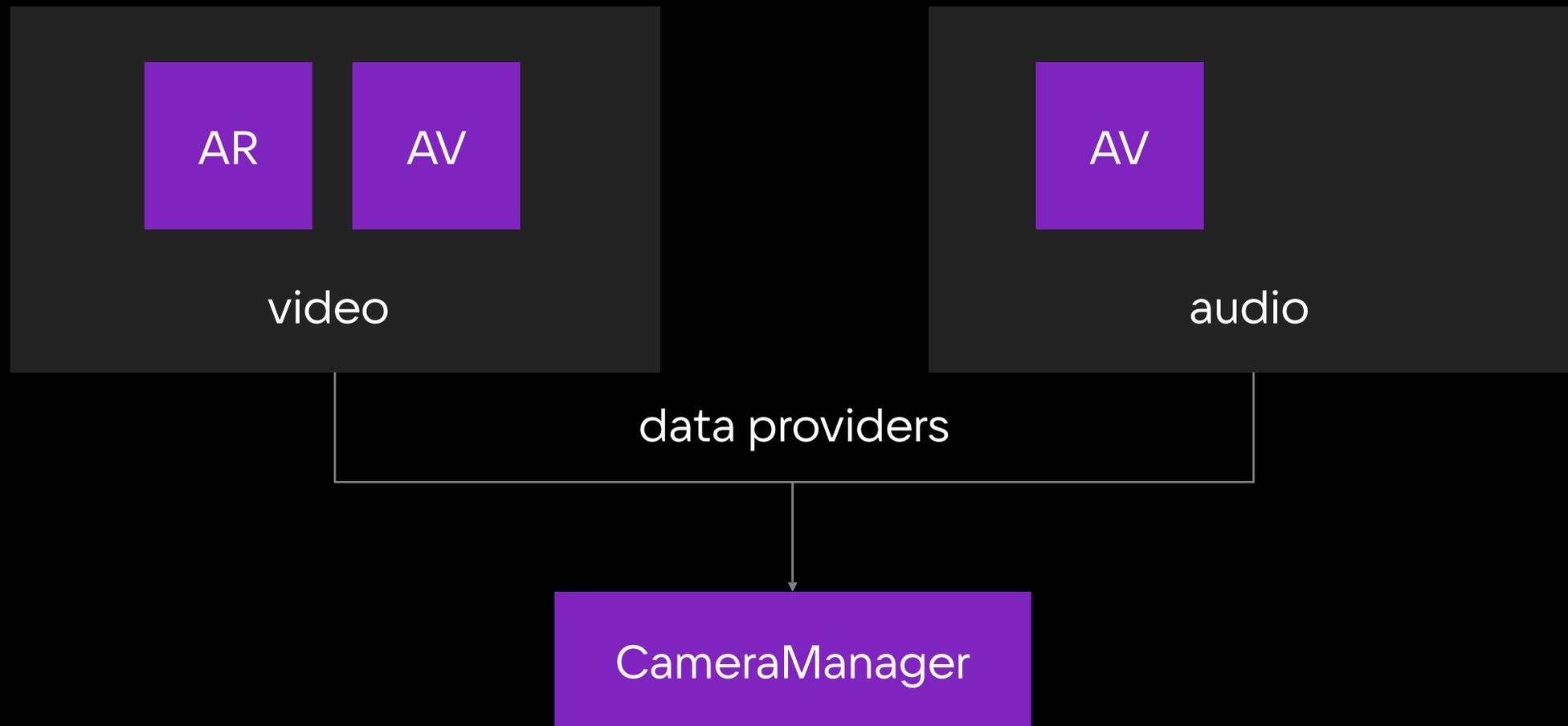
Задача

Поддержать обе возможности записи:
ARSession и **AVCaptureSession**

Желаемое устройство камеры



Провайдеры данных в CameraManager



Переключение провайдеров

```
if
  let videoSource,
  videoSource != activeVideoSource,
  let nextVideoProvider = videoProviders[videoSource]
{
  if let prevVideoProvider = activeVideoProvider {
    prevVideoProvider.stop()
    copyConfiguration(from: prevVideoProvider, to: nextVideoProvider)
  }

  activeVideoSource = videoSource
  nextVideoProvider.start()
}
```

Переключение провайдеров

```
if
  let videoSource,
  videoSource != activeVideoSource,
  let nextVideoProvider = videoProviders[videoSource]
{
  if let prevVideoProvider = activeVideoProvider {
    prevVideoProvider.stop()
    copyConfiguration(from: prevVideoProvider, to: nextVideoProvider)
  }

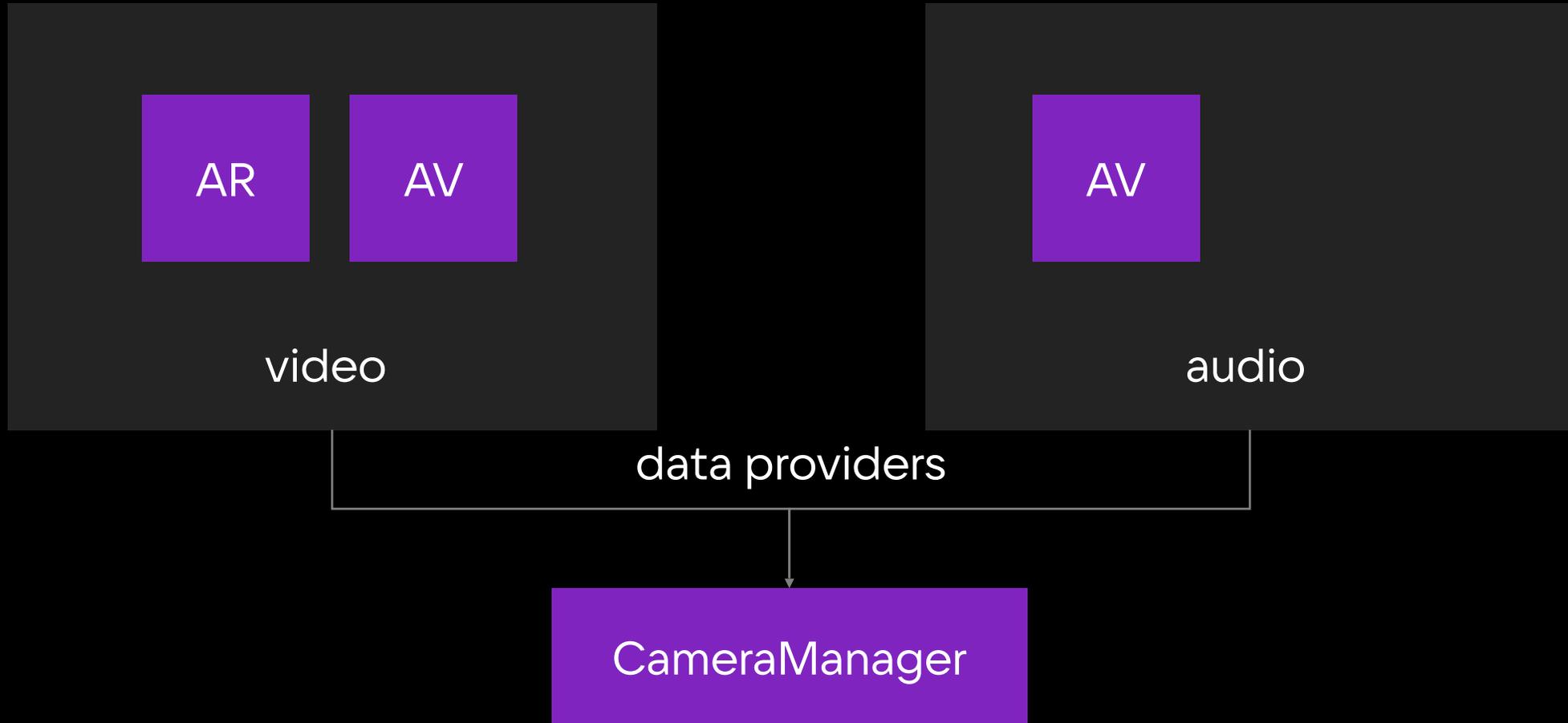
  activeVideoSource = videoSource
  nextVideoProvider.start()
}
```

Переключение провайдеров

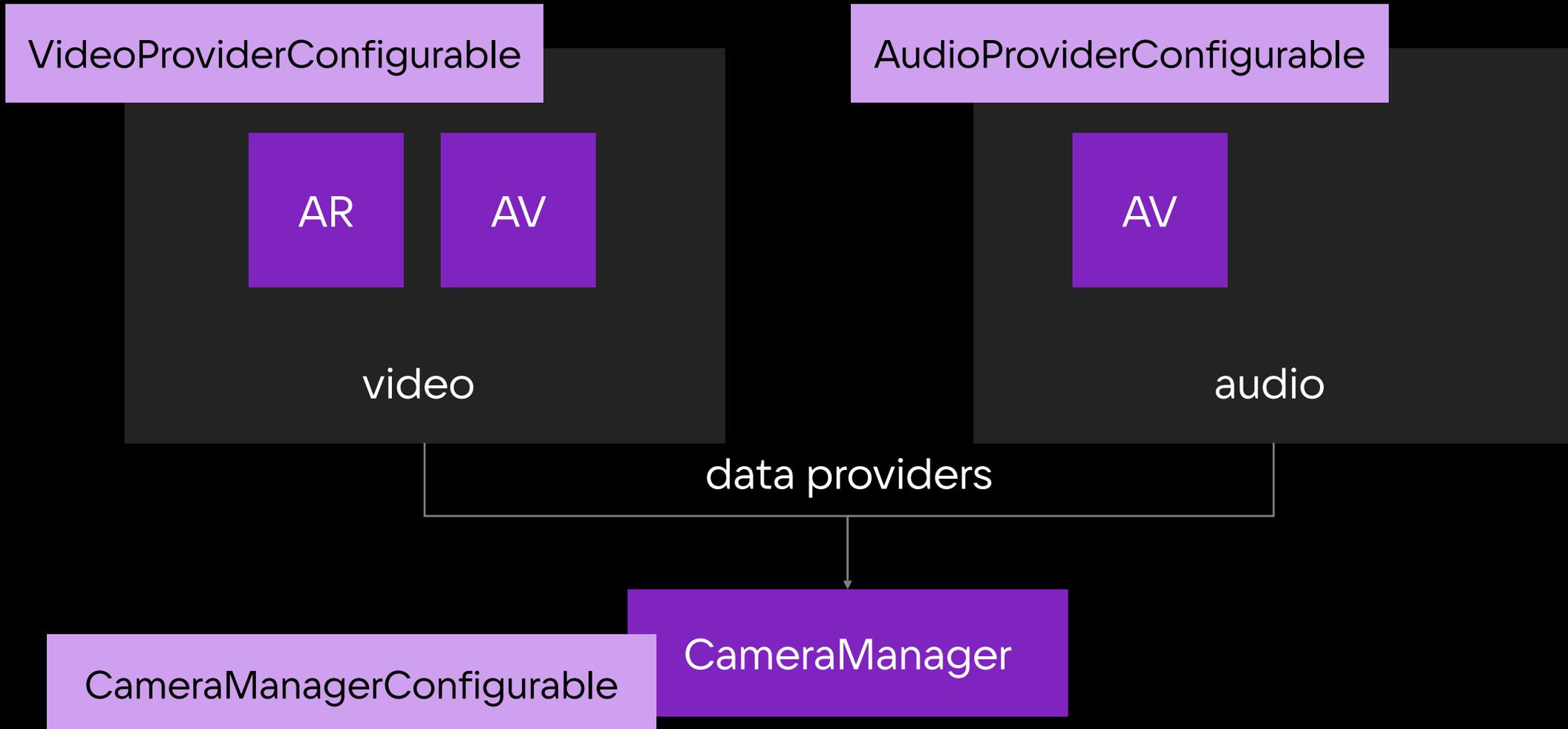
```
if
  let videoSource,
  videoSource != activeVideoSource,
  let nextVideoProvider = videoProviders[videoSource]
{
  if let prevVideoProvider = activeVideoProvider {
    prevVideoProvider.stop()
    copyConfiguration(from: prevVideoProvider, to: nextVideoProvider)
  }

  activeVideoSource = videoSource
  nextVideoProvider.start()
}
```

А как конфигурировать?



А как конфигурировать?



```

func updateConfig(
    _ updateRoutine: @escaping (VCMCameraManagerConfigurator) -> Void,
    completion: (() -> Void)?
) {
    sessionQueue.async {
        defer {
            if completion != nil {
                DispatchQueue.main.async {
                    completion?()
                }
            }
        }

        guard
            let activeAudioProvider = self.activeAudioProvider,
            let activeVideoProvider = self.activeVideoProvider
        else { return }

        let configurator = VCMCameraManagerConfigurator(
            cameraManager: self,
            audioProvider: activeAudioProvider,
            videoProvider: activeVideoProvider
        )

        updateRoutine(configurator)
    }
}

```

```

func updateConfig(
    _ updateRoutine: @escaping (VCMCameraManagerConfigurator) -> Void,
    completion: (() -> Void)?
) {
    sessionQueue.async {
        defer {
            if completion != nil {
                DispatchQueue.main.async {
                    completion?()
                }
            }
        }

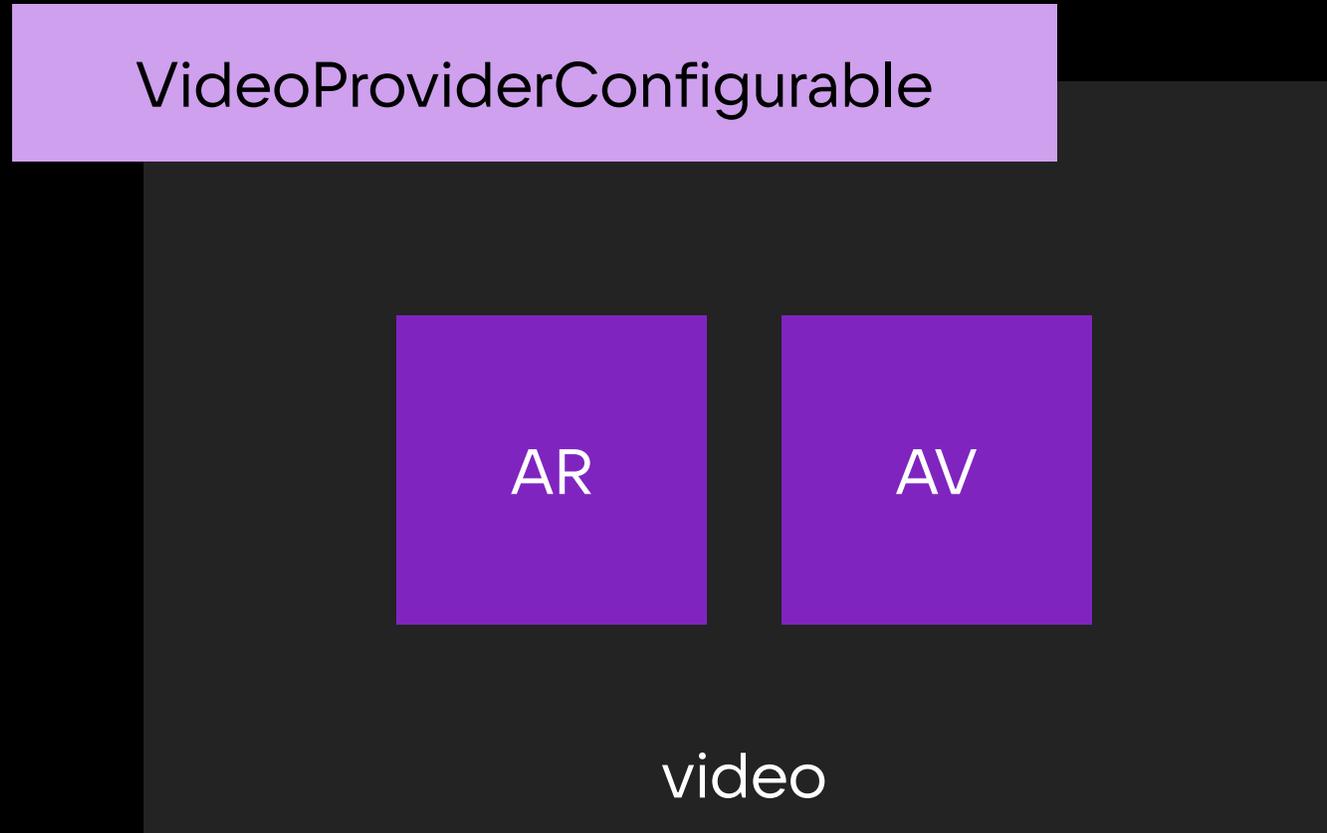
        guard
            let activeAudioProvider = self.activeAudioProvider,
            let activeVideoProvider = self.activeVideoProvider
        else { return }

        let configurator = VCMCameraManagerConfigurator(
            cameraManager: self,
            audioProvider: activeAudioProvider,
            videoProvider: activeVideoProvider
        )

        updateRoutine(configurator)
    }
}

```

Проблема (преимущество) единых интерфейсов



Проблема (преимущество) единых интерфейсов

```
public protocol VCMVideoDataConfigurable {  
    // ...  
    var zoomFactor: CGFloat { get set }  
    // ...  
}
```

Не работает Zoom 🥲

```
// 1. сдвиг центра изображения к точке (0, 0)
transform = transform.translatedBy(x: CGFloat(destWidth) / 2, y: CGFloat(destHeight) / 2)

// 2. отражение по горизонтали
if flipHorizontally {
    transform = transform.scaledBy(x: 1, y: -1)
}

// 3. поворот, если требуется
if rotationFactor != 0 {
    transform = transform.rotated(by: CGFloat(rotationFactor) * CGFloat.pi / 2)
}

// 4. масштабирование
if scale != 1 {
    transform = transform.scaledBy(x: scale, y: scale)
}

// 5. перемещение изображения в центр целевого кадра
transform = transform.translatedBy(x: -CGFloat(srcWidth) / 2, y: -CGFloat(srcHeight) / 2)
```

Итоги: прокачали камеру!

1. Сделали Зуминг, Фонарик, Смену фокуса, Смену экспозиции, Смену камер
2. Кастомные фильтры для каждого кадра
3. Решили задачу поиска лица на кадре с помощью ARKit'a
4. Спроектировали камеру с переключением провайдеров данных



КАК МЕНЯ НАЙТИ

@actorFromCalifornia

