



# Editing Media with AV Foundation

Overview and best practices

**Eric Lee**  
iPhone Engineering

# What You'll Learn

- Why and when you should use AV Foundation editing
- Concepts underlying manipulation of timed-based media
- What editing tasks you can accomplish using AV Foundation

# Sample Code for This Session

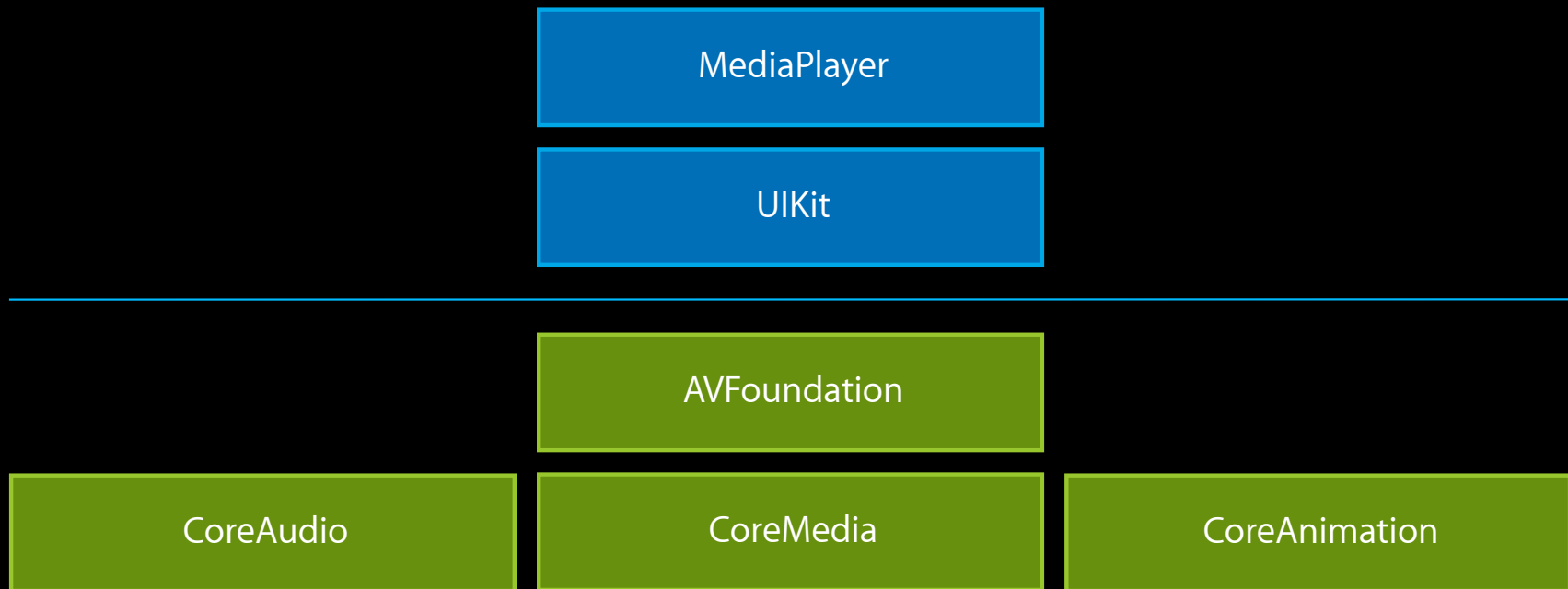
iOS 4

- AVPlayerDemo
- AVEditDemo

**Materials available at:**

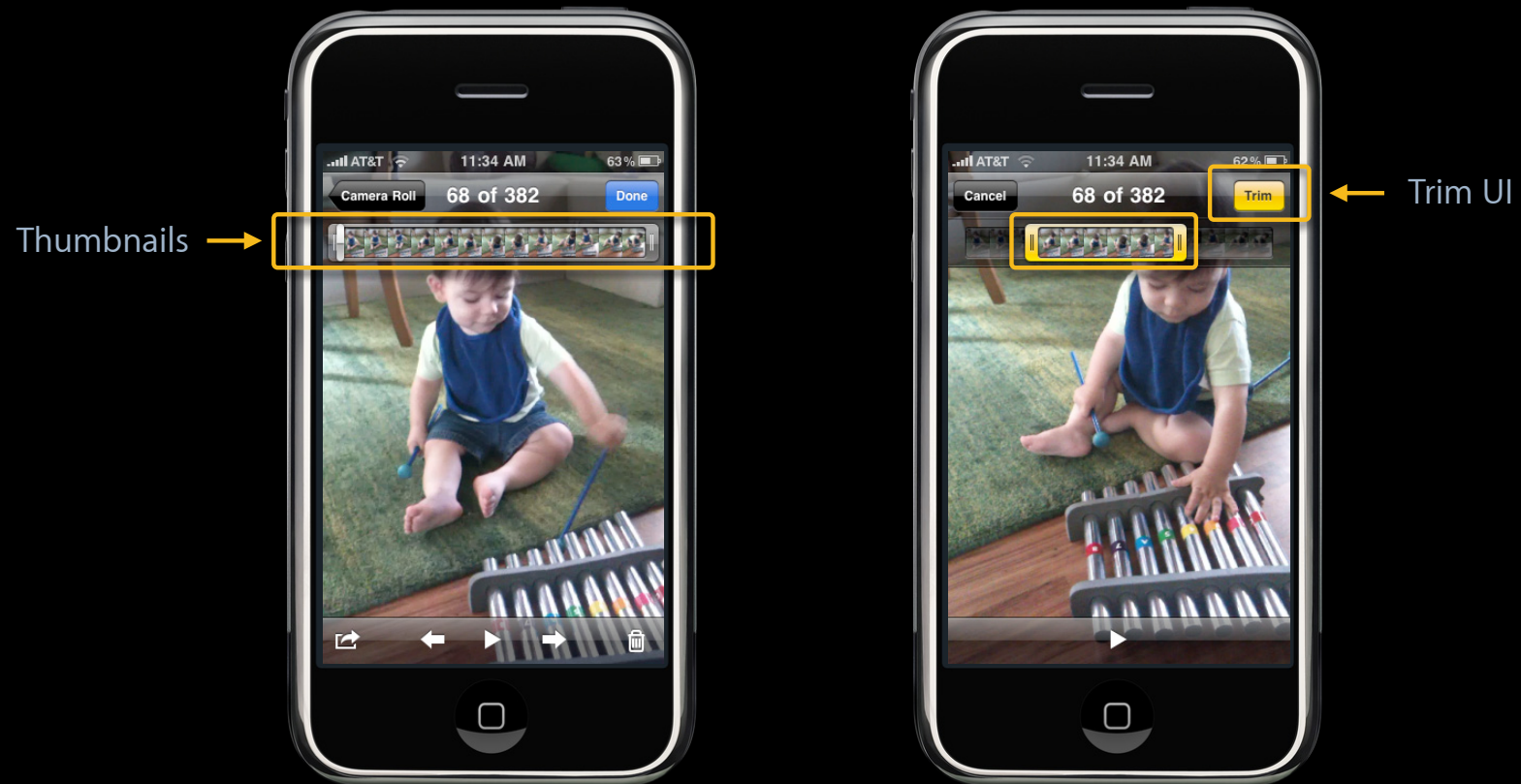
<http://developer.apple.com/wwdc/attendee/>

# Technology Framework



# Editing in iPhone OS 3.x

## A user feature

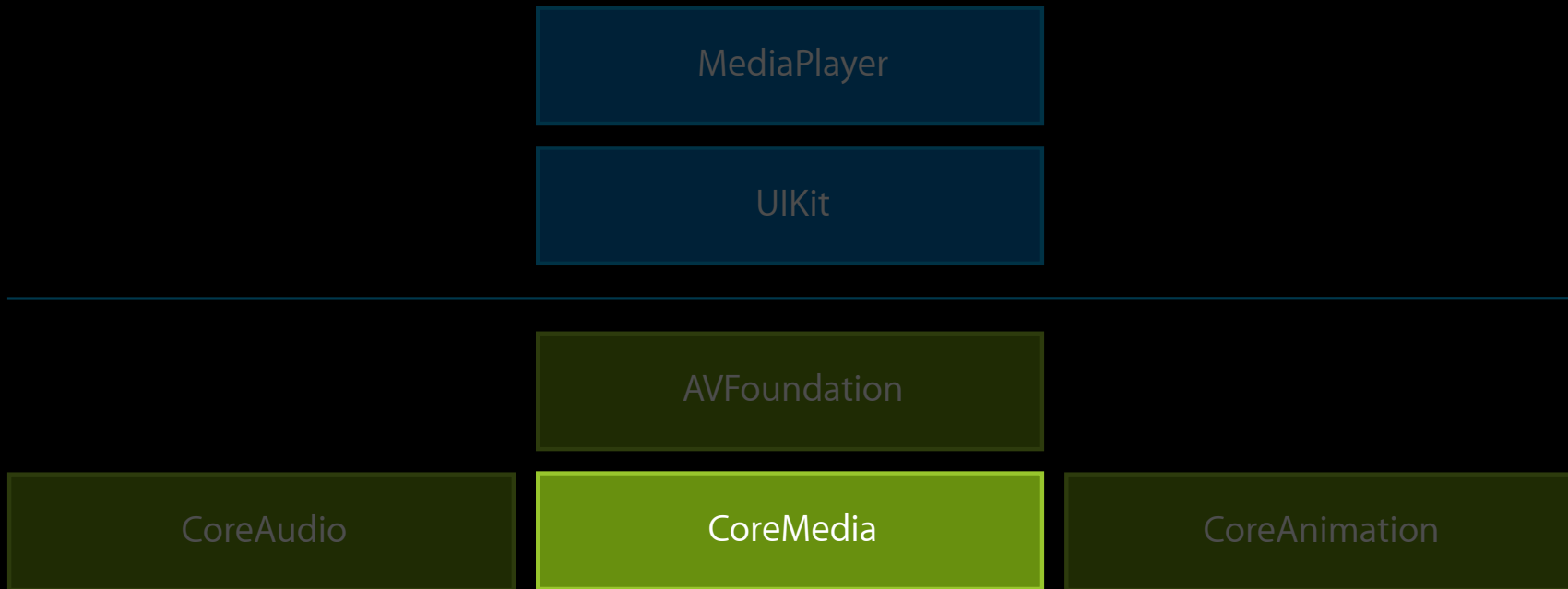


# Editing APIs in AV Foundation

## Scenarios

- Create an image for a time
- Trim a movie to a time range
- Cutting together multiple clips
- Audio mixing
- Video transitions
- Incorporating Core Animation in movies

# Fundamentals

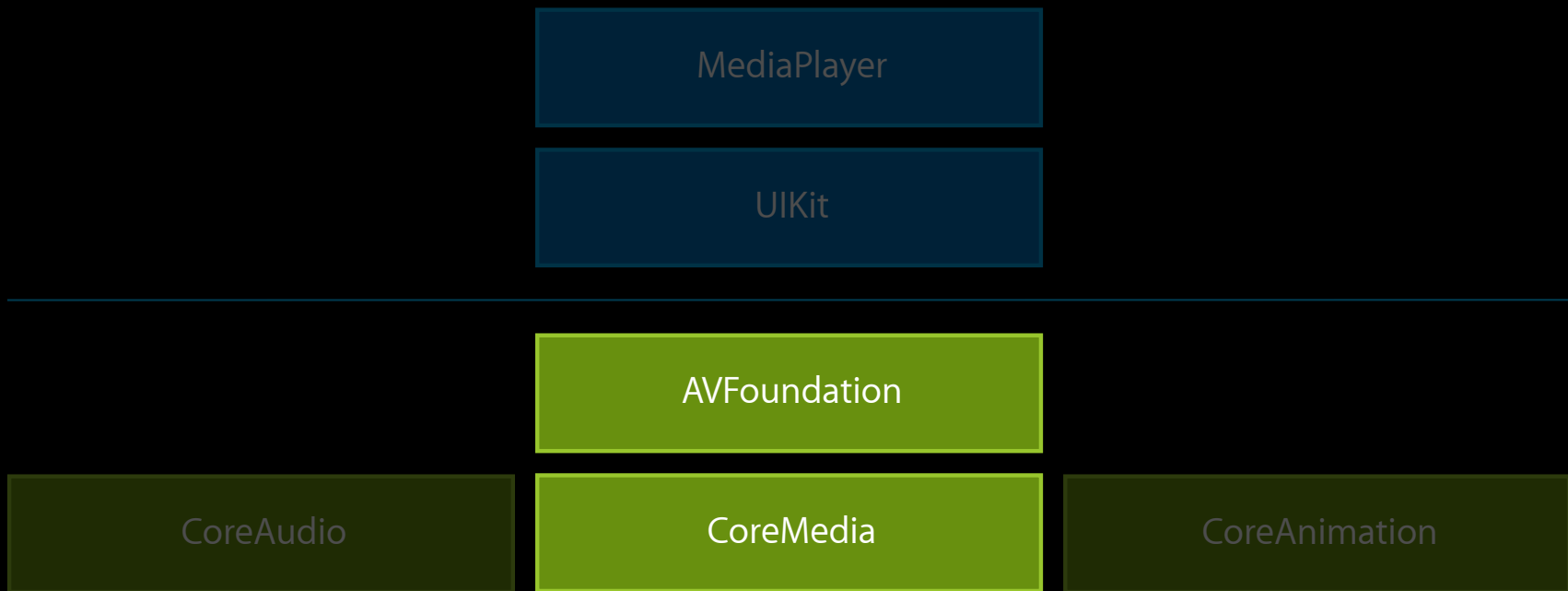


# CMTIME

## Struct type for rational time

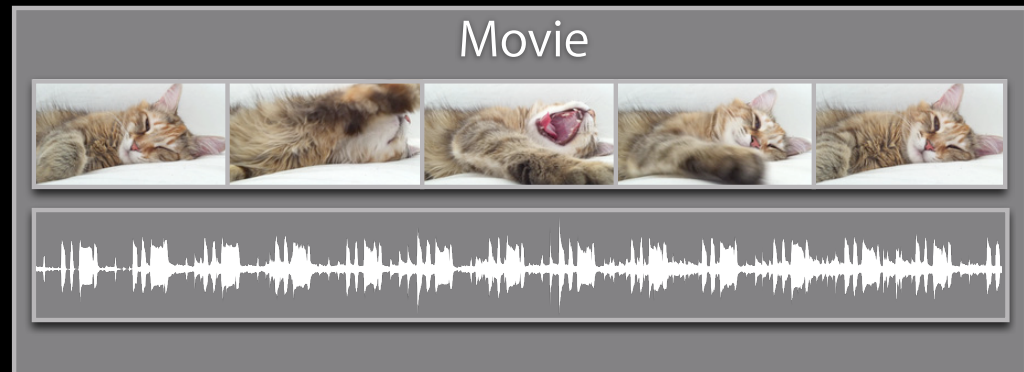
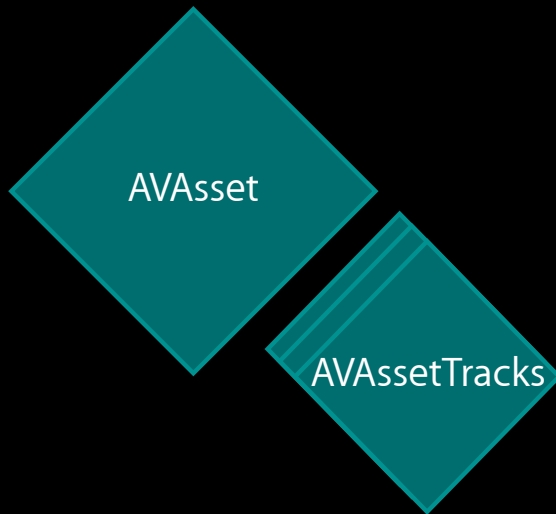
- `CMTIME t = CMTIME_Make( time value, time scale );`
- `kCMTIME_Zero`, `kCMTIME_Invalid`
- `x = CMTIME_Add( y, z );`
- `if( CMTIME_COMPARE_INLINE( t1, <=, t2 ) ) { ... }`
- `CMTIME_Range r = CMTIME_Range_Make( start time, duration );`
- `CMTIME_Mapping m = CMTIME_Mapping_Make( source range, target range );`

# Moving Up



# Movies and AVAssets

- AVURLAsset represents movies in files
- An AVAssetTrack represents a particular track inside the movie
- Use AVPlayerItem to play an AVAsset



# Editing APIs in AV Foundation

## Scenarios

- Create an image for a time
- Trim a movie to a time range
- Cutting together multiple clips
- Audio mixing
- Video transitions
- Incorporating Core Animation in movies

# Demo

Custom playback UI

AVPlayerDemo

# Grab Image at Time

Grab images from an AVAsset using AVAssetImageGenerator

```
AVAssetImageGenerator *imageGenerator =  
    [AVAssetImageGenerator  
        assetImageGeneratorWithAsset:myAsset];  
  
[imageGenerator generateCGImagesAsynchronouslyForTimes:timeArray  
    completionHandler:handlerBlock];  
  
// need to retain imageGenerator until you get the images
```

# Image Generation Completion Block

## Check the result

```
AVAssetImageGeneratorCompletionHandler handlerBlock =  
    ^(CMTime requestedTime, CGImageRef image, CMTime actualTime,  
      AVAssetImageGeneratorResult result, NSError *error){  
    switch (result) {  
        case AVAssetImageGeneratorSucceeded:  
            /* image is valid */  
            break;  
        case AVAssetImageGeneratorFailed:  
            /* error */  
            break;  
        case AVAssetImageGeneratorCancelled:  
            /* cancelled */  
            break;  
    }  
}
```

# Editing APIs in AV Foundation

## Scenarios

- Create an image for a time
- Trim a movie to a time range
- Cutting together multiple clips
- Audio mixing
- Video transitions
- Incorporating Core Animation in movies

# Export and Trimming

## Export an AVAsset to a new file using AVAssetExportSession

- Presets for different sizes, bitrates, etc.
- Optionally set timeRange to trim
- Optionally add metadata

```
AVAssetExportSession *exportSession = [[AVAssetExportSession alloc]  
    initWithAsset:asset presetName:AVAssetExportPresetMediumQuality];
```

```
exportSession.outputURL = ...;  
exportSession.outputFileType = AVFileTypeQuickTimeMovie;
```

```
exportSession.timeRange = CMTimeRangeMake(startTime, duration);
```

```
exportSession.metadata = ...;
```

```
[exportSession exportAsynchronouslyWithCompletionHandler:handlerBlock];
```

# Export Completion Block

## Check the status

```
void (^handlerBlock)(void) = ^{  
    switch (exportSession.status) {  
        case AVAssetExportSessionStatusCompleted:  
            /* export complete */  
            break;  
        case AVAssetExportSessionStatusFailed:  
            /* export error (see exportSession.error) */  
            break;  
        case AVAssetExportSessionStatusCancelled:  
            /* export cancelled */  
            break;  
    }  
}
```

# A Word on Error Handling

## Handle failures gracefully

- AVAssetExportSession will not overwrite files
- AVAssetExportSession will not write files outside of your sandbox

# Export and Multitasking

## Handle failures gracefully

- Other apps that start playback will interrupt a background export
- Even in the foreground, an incoming phone call will interrupt export

# Editing APIs in AV Foundation

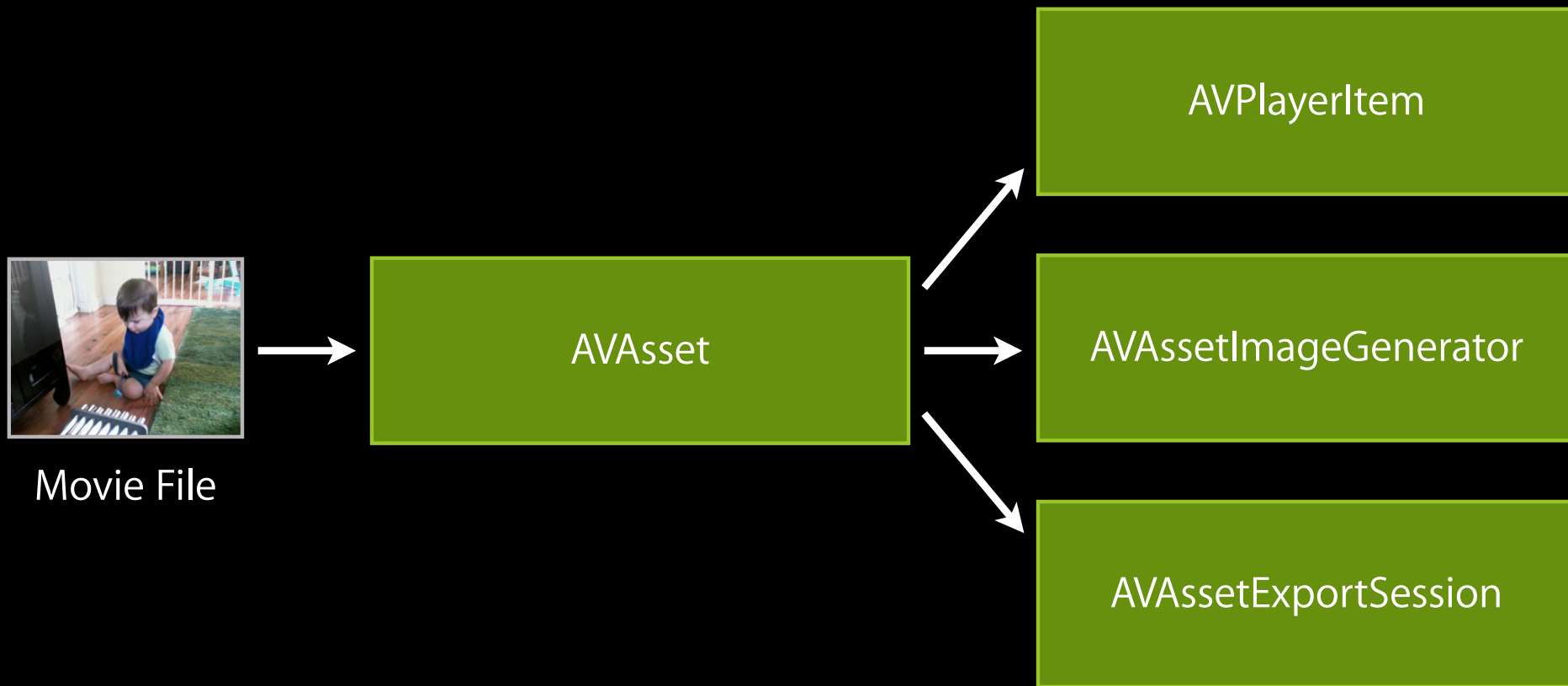
## Scenarios

- Create an image for a time
- **Trim a movie to a time range**
- Cutting together multiple clips
- Audio mixing
- Video transitions
- Incorporating Core Animation in movies

# AVAsset and AVComposition

Cornerstones of editing

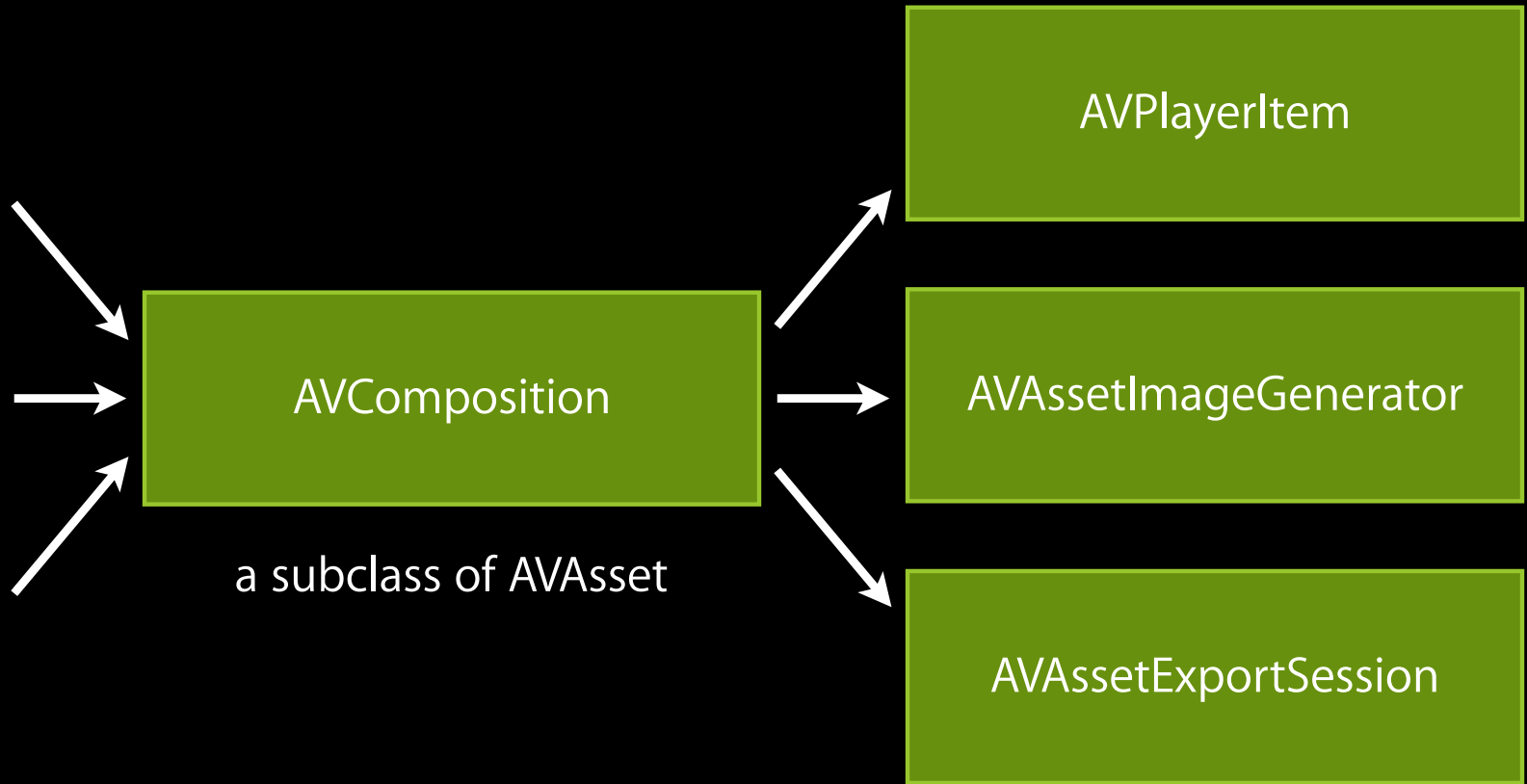
# AVAsset as a Source Object



# AVComposition



Movie Files



# Editing APIs in AV Foundation

## Scenarios

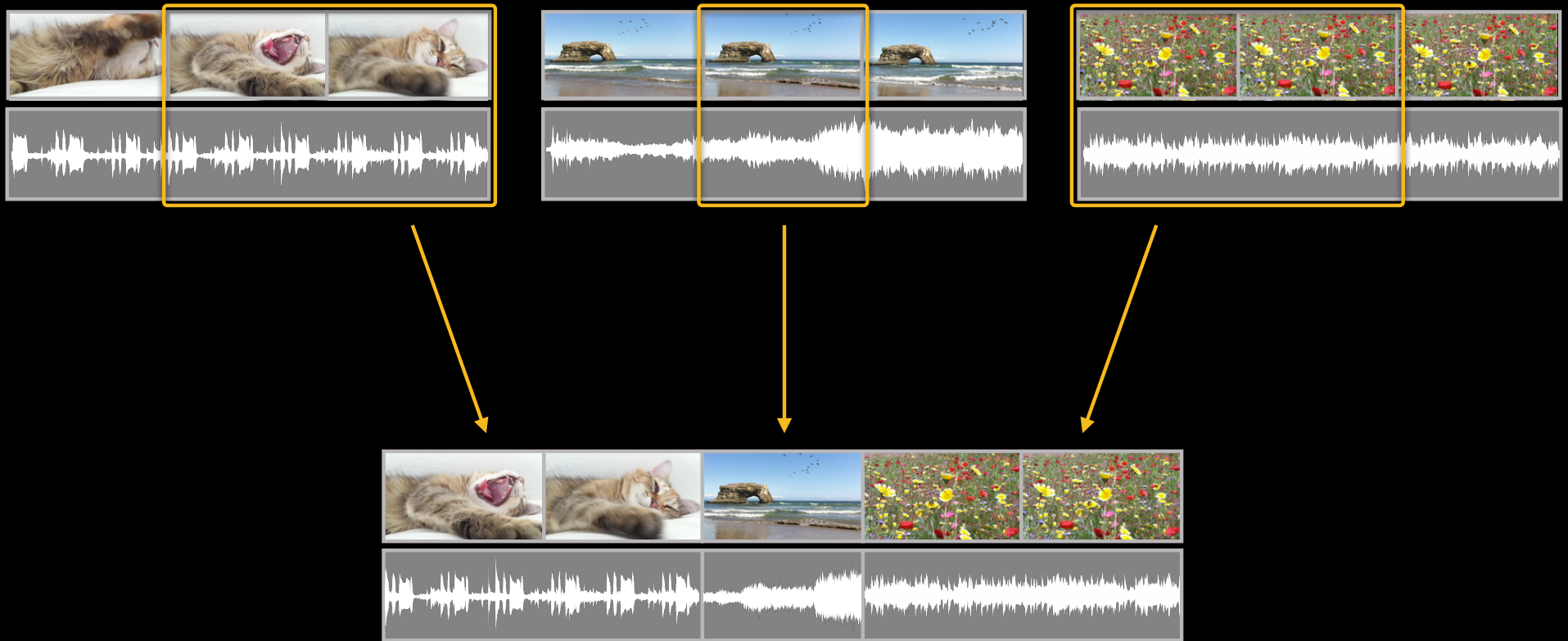
- Create an image for a time
- Trim a movie to a time range
- **Cutting together multiple clips**
- Audio mixing
- Video transitions
- Incorporating Core Animation in movies

# Demo

Cutting together movie clips

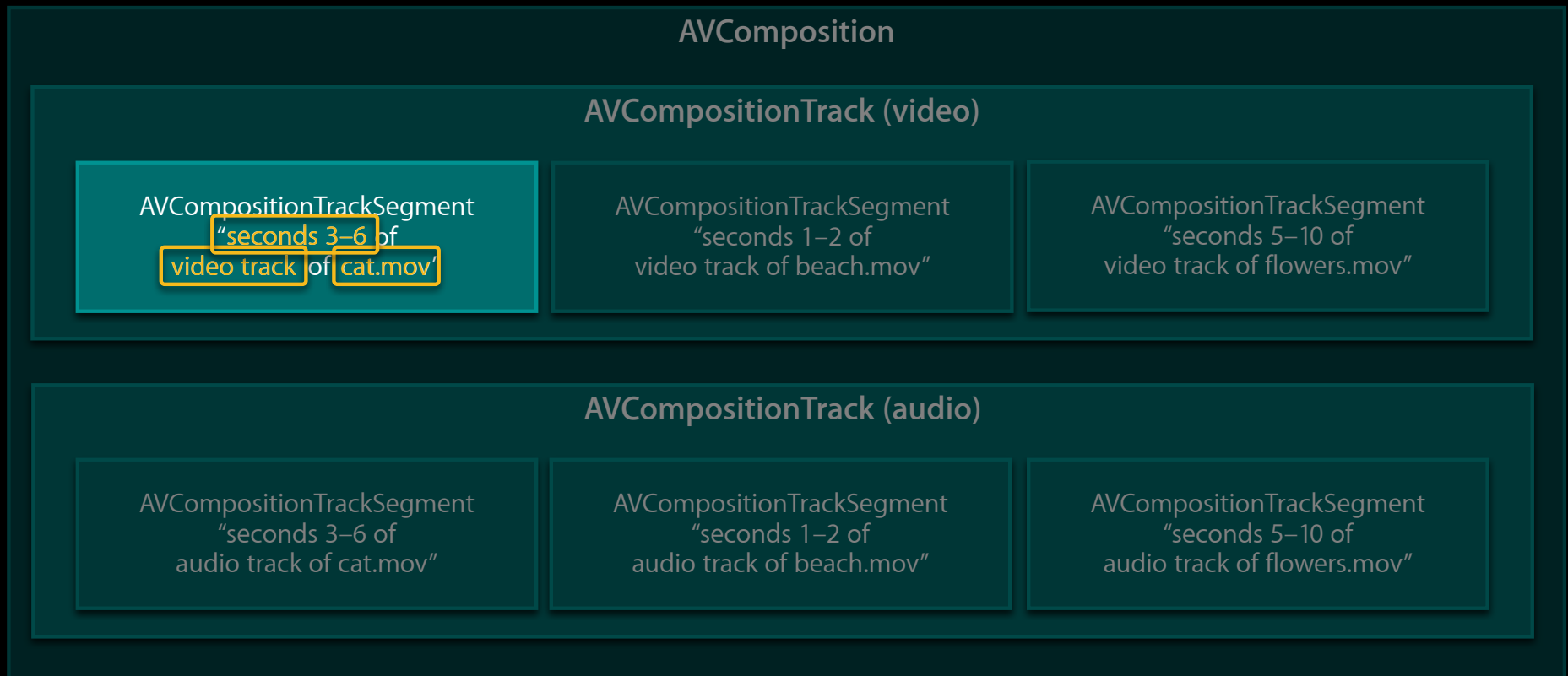
Sample code: `AVEditDemo` (see `SimpleEditor.m`)

# Composing a Timeline



# AVComposition

AVComposition assembles asset segments on a timeline



# AVMutableComposition

- You can edit across all tracks of a composition:
  - `[composition insertTimeRange:... ofAsset:... atTime:... error:...];`
- You can edit a single track:
  - `[compositionTrack insertTimeRange:... ofTrack:... atTime:... error:...];`
- You can change the segment array directly:
  - `[compositionTrack setSegments:...];`

# AVEditDemo

See `buildSequenceComposition` in `SimpleEditor.m`

```
AVMutableComposition *composition = [AVMutableComposition composition];
```

```
AVMutableCompositionTrack *compositionVideoTrack =  
    [composition addMutableTrackWithMediaType:AVMediaTypeVideo  
        preferredTrackID:...];
```

```
[compositionVideoTrack insertTimeRange:...  
    ofTrack:clipVideoTrack atTime:... error:...];
```

# When Not to Mutate Compositions

Don't modify an `AVMutableComposition` during playback, image generation, or export



- Make a copy for these tasks; then it's safe to modify the original:

```
AVPlayerItem *playerItem =  
    [AVPlayerItem playerItemWithAsset:  
        [[mutableComposition copy] autorelease]];
```



- Switch player to new item:

```
[player replaceCurrentItemWithPlayerItem:playerItem];
```

# Editing APIs in AV Foundation

## Scenarios

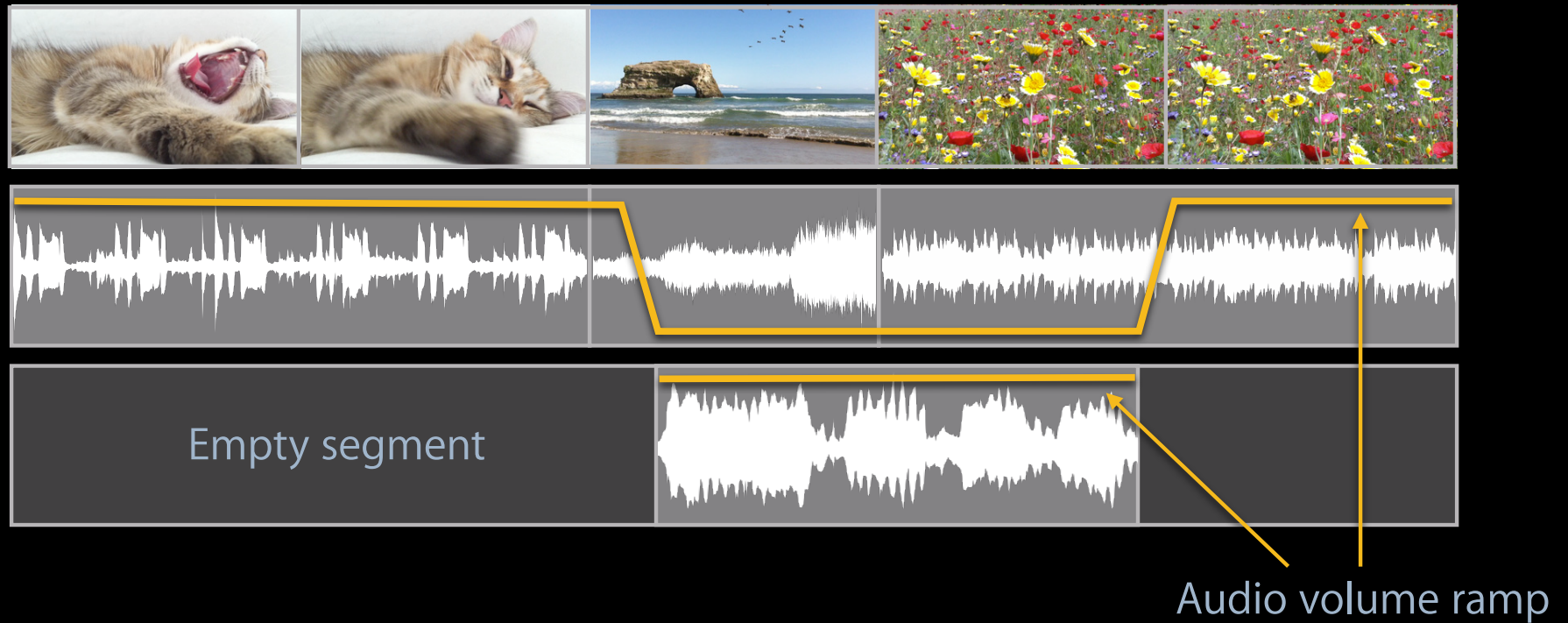
- Create an image for a time
- Trim a movie to a time range
- **Cutting together multiple clips**
- Audio mixing
- Video transitions
- Incorporating Core Animation in movies

# Demo

Mixing in additional audio

Sample code: `AVEditDemo` (see `SimpleEditor.m`)

# Adding a Commentary Track



# AVComposition and AVAudioMix

## AVComposition



## AVAudioMix



# AVAudioMix

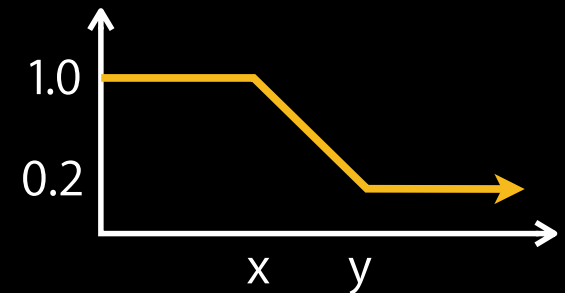
Tool for adding volume adjustments

- Has array of AVAudioMixInputParameters
  - Each adjusts the volume level of one track
  - Tracks without AVAudioMixInputParameters get default volume

# AVMutableAudioMix

```
AVMutableAudioMixInputParameters *trackMix =  
    [AVMutableAudioMixInputParameters  
        audioMixInputParametersWithTrack:mainAudioTrack];
```

```
[trackMix setVolume:1.0 atTime:kCMTimeZero];  
  
[trackMix setVolumeRampFromStartVolume:1.0  
    toEndVolume:0.2  
    timeRange:CMTimeRangeMake(x, y-x)];  
...
```



```
AVMutableAudioMix *audioMix = [AVMutableAudioMix audioMix];  
audioMix.inputParameters = [NSArray arrayWithObject:trackMix];
```

# Using AVAudioMix

- To apply AVAudioMix for playback:
  - `playerItem.audioMix = audioMix;`
- To apply AVAudioMix for export:
  - `exportSession.audioMix = audioMix;`

# Editing APIs in AV Foundation

## Scenarios

- Create an image for a time
- Trim a movie to a time range
- Cutting together multiple clips
- **Audio mixing**
- Video transitions
- Incorporating Core Animation in movies

# Demo

## Video transitions

Sample code: `AVEditDemo` (see `SimpleEditor.m`)

# AVComposition and AVVideoComposition

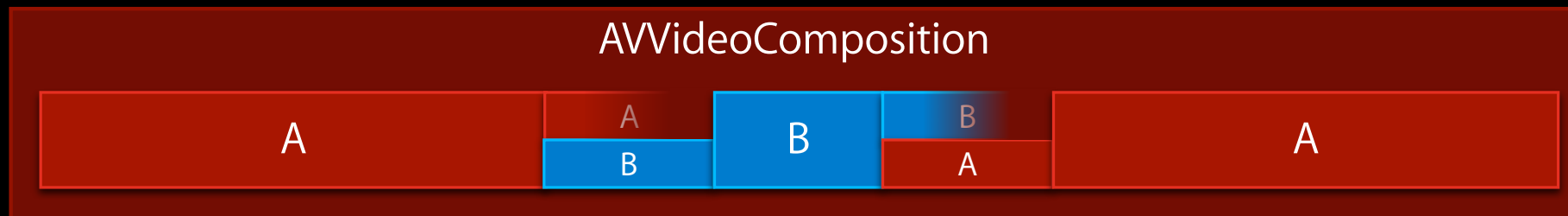
## AVComposition



## AVVideoComposition

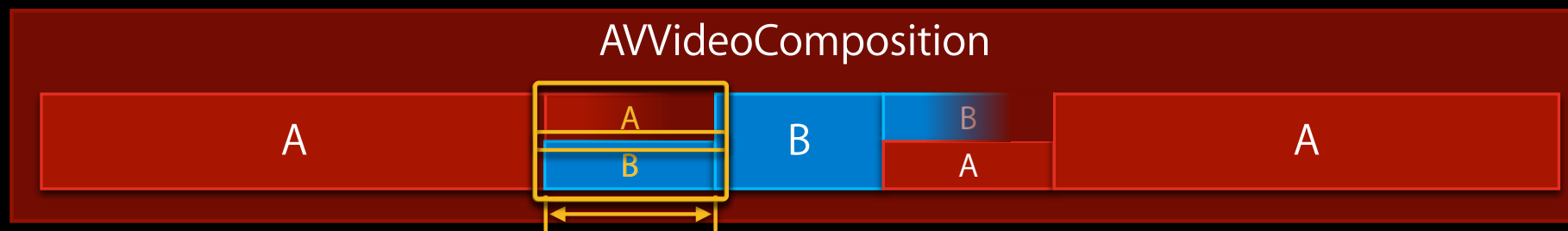


# AVVideoComposition



- Has an array of *AVVideoCompositionInstructions*
  - Each instruction describes the output video in terms of input layers (*AVVideoCompositionLayerInstruction*)
    - Each layer has an opacity and an affine transform
    - Opacity can be tweened — e.g., for a cross-fade
    - Affine transform can be tweened — e.g., for a push transition

# AVVideoCompositionInstruction



```
AVMutableVideoCompositionInstruction *transition =  
    [AVMutableVideoCompositionInstruction videoCompositionInstruction];
```

```
transition.timeRange = ...;
```

```
AVMutableVideoCompositionLayerInstruction *fromLayer =  
    [AVMutableVideoCompositionLayerInstruction  
     videoCompositionLayerInstructionWithAssetTrack:trackA];
```

```
// Fade out trackA by setting a ramp from 1.0 to 0.0.  
[fromLayer setOpacityRampFromStartOpacity:1.0 toEndOpacity:0.0  
 timeRange:...];
```

```
AVMutableVideoCompositionLayerInstruction *toLayer = ...
```

```
transition = [NSArray arrayWithObjects:fromLayer, toLayer, nil];
```

# AVVideoComposition

```
AVMutableVideoComposition *videoComposition =  
    [AVMutableVideoComposition videoComposition];
```

```
videoComposition.instructions = [NSArray arrayWithObject:transition];
```

```
videoComposition.frameDuration = CMTimeMake(1, 30);
```

```
videoComposition.renderSize = CGSizeMake(1280, 720);
```

```
videoComposition.renderScale = 0.5; // for playback only
```

# Using AVVideoComposition

- For playback:
  - `playerItem.videoComposition = videoComposition;`
- For image generation:
  - `assetImageGenerator.videoComposition = videoComposition;`
- For export:
  - `assetExportSession.videoComposition = videoComposition;`

# Pitfalls

- AVVideoCompositionInstructions must not overlap or contain gaps



- AVVideoComposition must not be shorter than AVComposition



# Hardware Requirements for Video Composition

- iPhone 3GS or later
- iPod touch (3rd generation)

# Temporal Video Compression

Advanced

- Not restricted to editing at key frames
- Use AVVideoComposition and alternating segments between two video tracks to give AV Foundation more time for catchup decoding

I P P P P | P P P P | P P P P P ...



# Other Uses for AVVideoComposition

Advanced

- Two assets with different video sizes inserted into an AVComposition will end up in different video tracks by default
  - AVPlayer will only play one of them
  - Add an AVVideoComposition
- Movies captured in portrait or reverse landscape orientation will have a preferredTransform set on the video track
  - The transform will be ignored if the asset is placed in an AVComposition
  - Use AVVideoComposition to reinstate the rotation

# Editing APIs in AV Foundation

## Scenarios

- Create an image for a time
- Trim a movie to a time range
- Cutting together multiple clips
- Audio mixing
- **Video transitions**
- Incorporating Core Animation in movies

# Demo

## Core Animation in Movies

Sample code: `AVEditDemo` (see `SimpleEditor.m`)

# Core Animation in Movies

## AVComposition



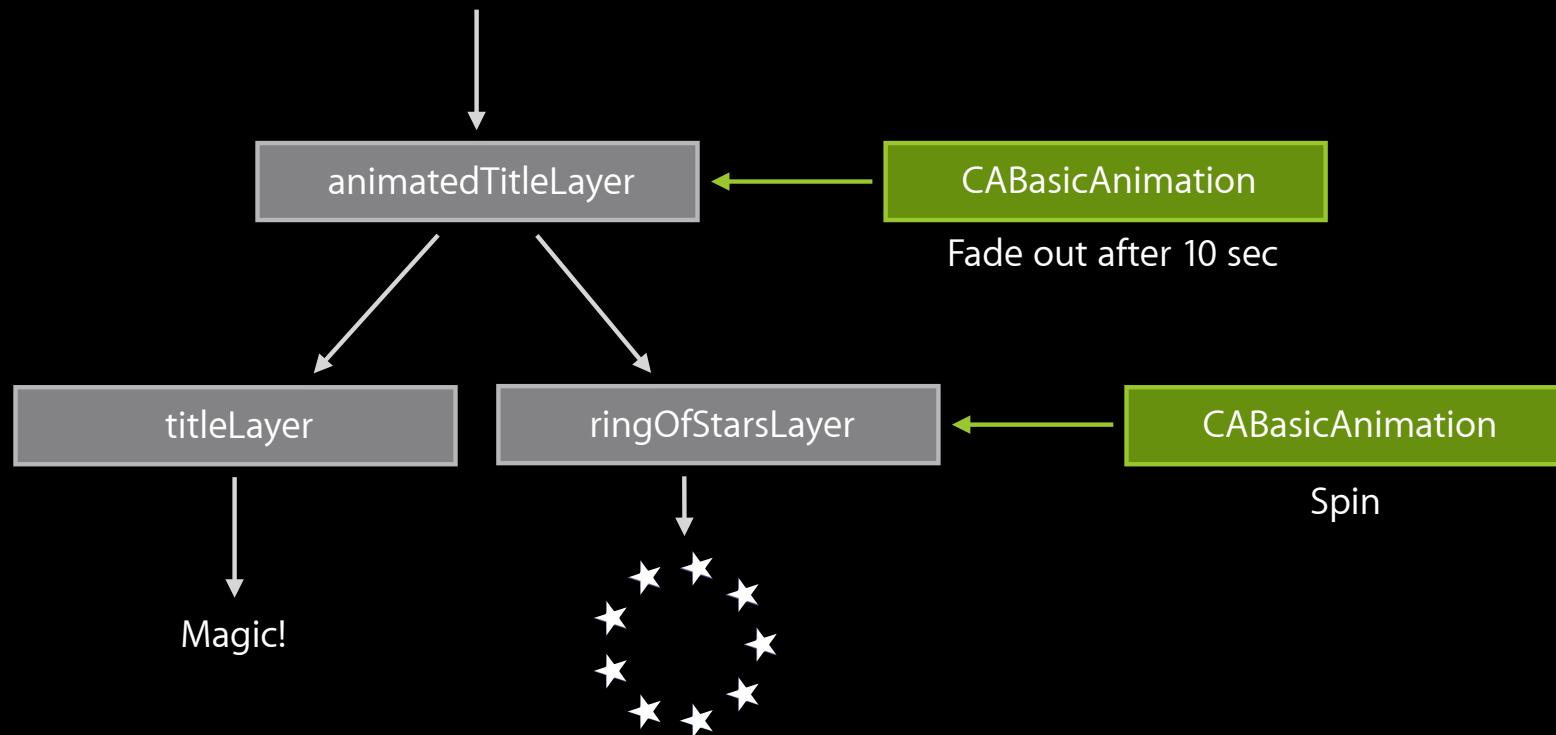
## CALayers



## CAAnimations

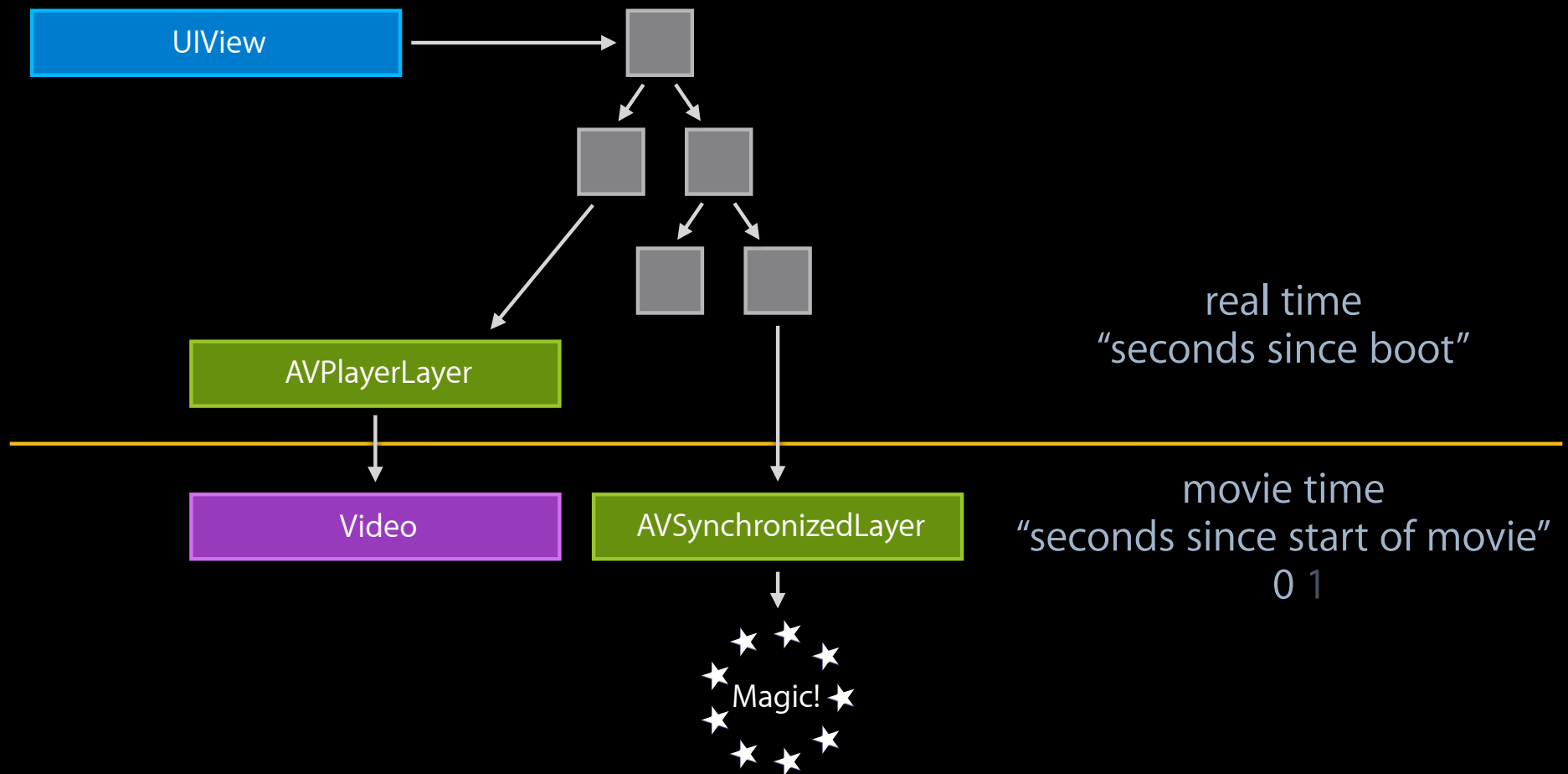


# AVEditDemo: Core Animation



Attend Sessions 424 and 425 for more information on Core Animation

# Animation, Video, and Time

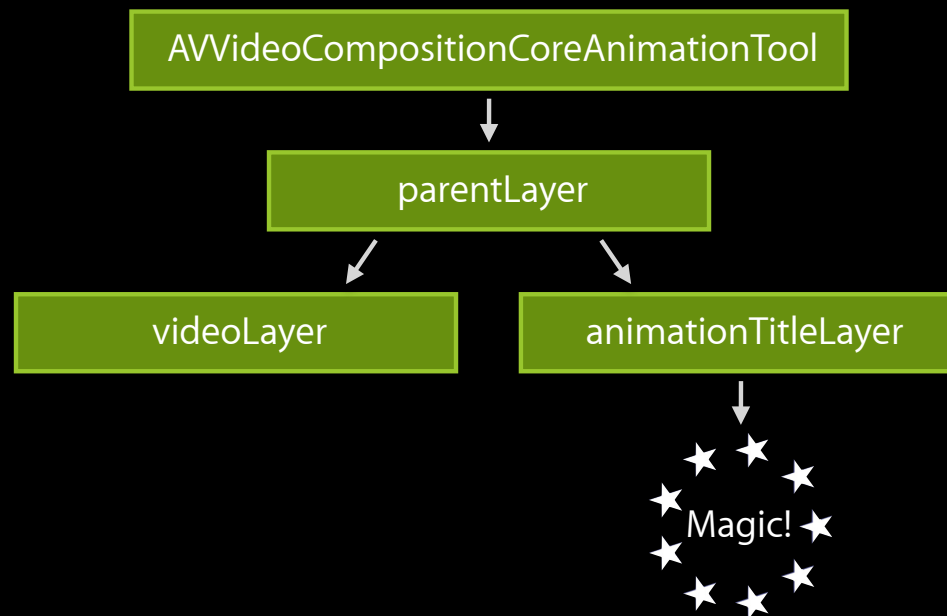


# Playback with Core Animation

- Use `AVSynchronizedLayer` to make animation use movie timing

# Export with Core Animation

- Use `AVVideoCompositionCoreAnimationTool` to integrate Core Animation as a “post-processing” video stage



- Set `compositionInstruction.enablePostProcessing` to `NO` to skip Core Animation rendering when not needed

# Multitasking

- Core Animation use in the background will cause the export to fail

# Core Animation Gotchas

Core Animation features that are convenient for real-time animation but don't work well in movies

- Zero `beginTime` is automatically translated to `CACurrentMediaTime()`
  - Use a small nonzero number: e.g., `1e-100` or `-1e-100` (`AVCoreAnimationBeginTimeZero`)
- Animations are automatically removed after Core Animation thinks they have passed
  - `animation.removedOnCompletion = NO;`

# Core Animation Gotchas

Core Animation features that are convenient for real-time animation but don't work well in movies

- Disable implicit animations

```
[CATransaction begin];  
    [CATransaction setDisableActions:YES];  
    // your layer code here  
[CATransaction commit];
```

# Stretch It Out

- Core Animation contributions may continue past the end of an AVComposition

AVComposition

Core Animation Animations



- Need to explicitly indicate how long playback or export should run
  - Set `playerItem.forwardPlaybackEndTime` for playback
  - Set `assetExportSession.timeRange` for export

# Summary

- Create an image for a time AVAssetImageGenerator
- Outputting a movie AVAssetExportSession
- Combining multiple clips AVComposition
- Audio volume adjustment AVAudioMix
- Video transitions AVVideoComposition
- Incorporating Core Animation AVSynchronizedLayer and AVVideoCompositionCoreAnimationTool

# More Information

## Eryk Vershen

Media Technologies Evangelist  
[evershen@apple.com](mailto:evershen@apple.com)

## Documentation

AV Foundation Framework Reference  
<http://developer.apple.com/iphone>

## Apple Developer Forums

<http://devforums.apple.com>

# Related Sessions

|   |                                   |
|---|-----------------------------------|
| Discovering AV Foundation (Repeat)                      | Nob Hill<br>Thursday 4:30PM       |
| Using the Camera with AV Foundation                     | Presidio<br>Tuesday 4:30PM        |
| Core Animation in Practice, Part 1                      | Nob Hill<br>Thursday 11:30AM      |
| Core Animation in Practice, Part 2                      | Nob Hill<br>Thursday 2:00PM       |
| Introducing Blocks and Grand Central Dispatch on iPhone | Russian Hill<br>Wednesday 11:30AM |

# Labs

AV Foundation Lab #1

Graphics & Media Lab C  
Wednesday 9:00AM

AV Foundation Lab #2

Graphics & Media Lab B  
Friday 11:30AM

Core Animation Lab

Graphics & Media Lab D  
Thursday 3:15PM





