

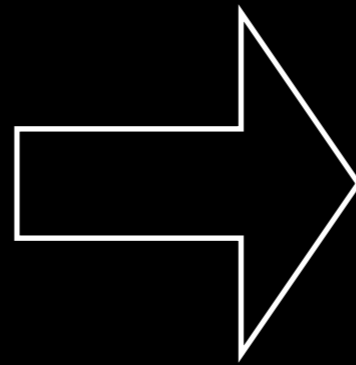
# Building High Performance iPhone Applications with Flash CS5

Mike Chambers  
Principal Product Manager  
Developer Relations  
Flash Platform





Flash Pro CS5



Adobe Flash CS5 will include support for packaging stand-alone apps for the Apple iPhone

# Flash Player 10.1



Smartphone enabled

Multitouch, accelerometer, screen orientation

Optimized memory, power, hardware acceleration

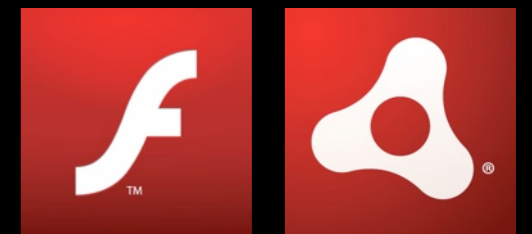
RAW Microphone Access

Global Error Handler

Public Beta on Labs Now!

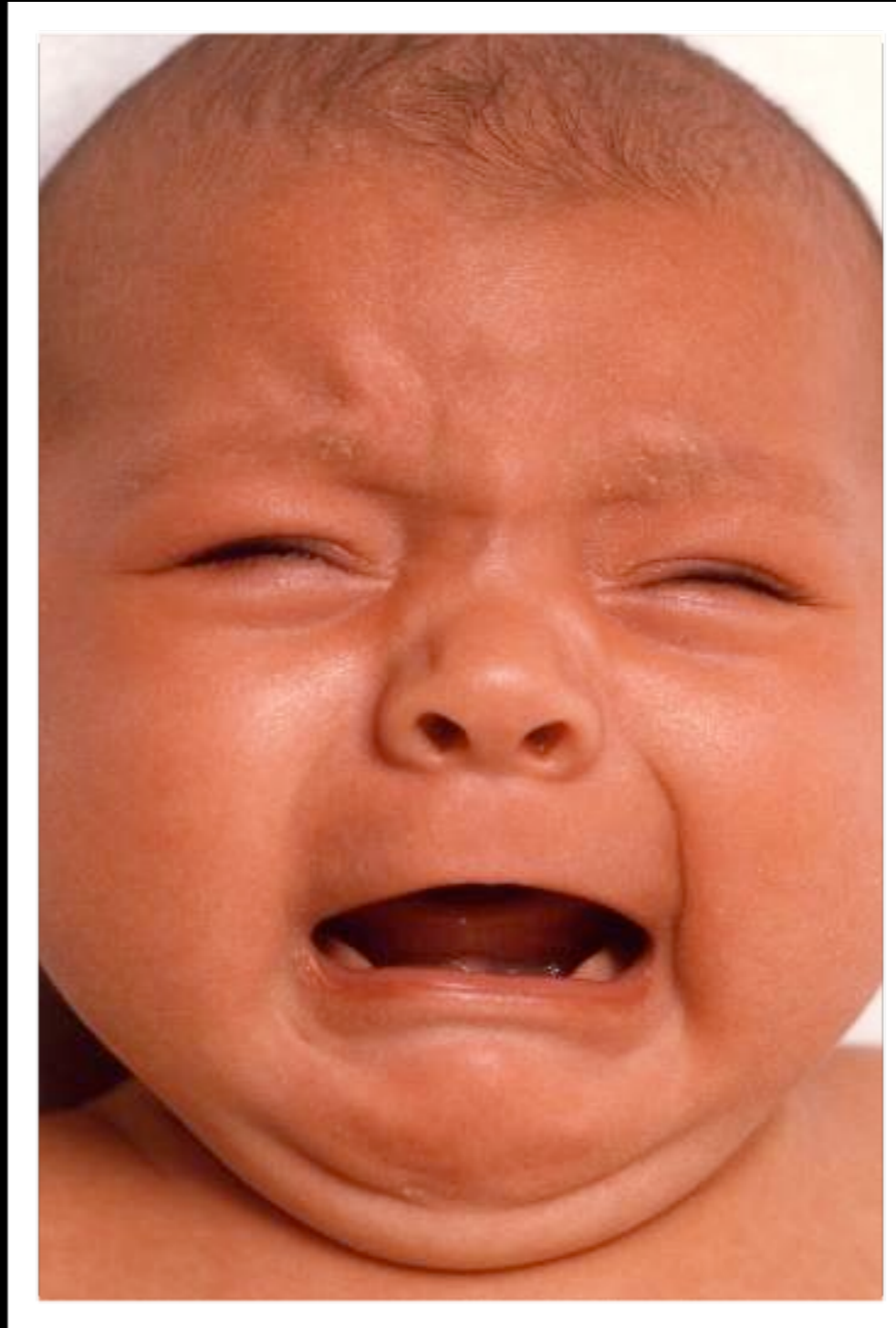
# New APIs

- MultiTouch
- Screen Orientation
- MediaLibrary
- Accelerometer
- Geo-location
- Cut / Copy / Paste
- Native TextInput
- tel:, mailto:, maps:, video:



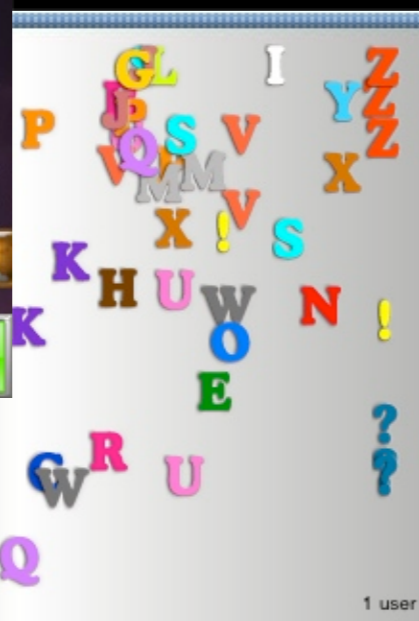
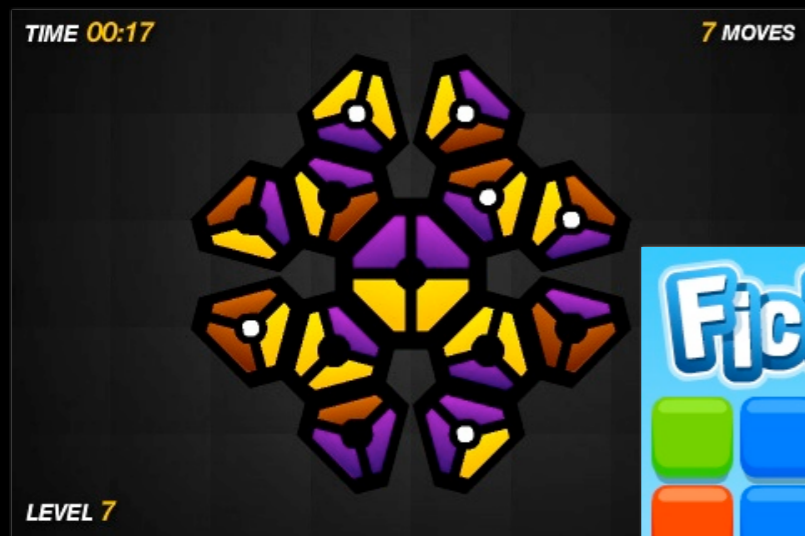


Flash Player 10.1 will be available across both browsers, desktops and devices. This includes in Adobe AIR 2.0, as well as Flash Player for Palm Pre, Android and Windows Mobile devices.



## No Flash Player for Safari Mobile on iPhone

Basically, we need apple's help with this. However, as the Palm Pre and Android Flash Player demos have shown, Flash content can perform well on this class of devices.



Already a number of iphone applications created with Flash on the itunes app store.

Level 1

# Alchemist

Gold  
0

LEVEL 1

00000950

A progress bar is visible at the top of the grid.

SCORE 301

COMBO 02

TIME 20

DRUMS 13

SKILL SHOT!



How does it work?

# LLVM

Low Level Virtual Machine

Open source compiler infrastructure designed for optimizing programs written in arbitrary programming languages

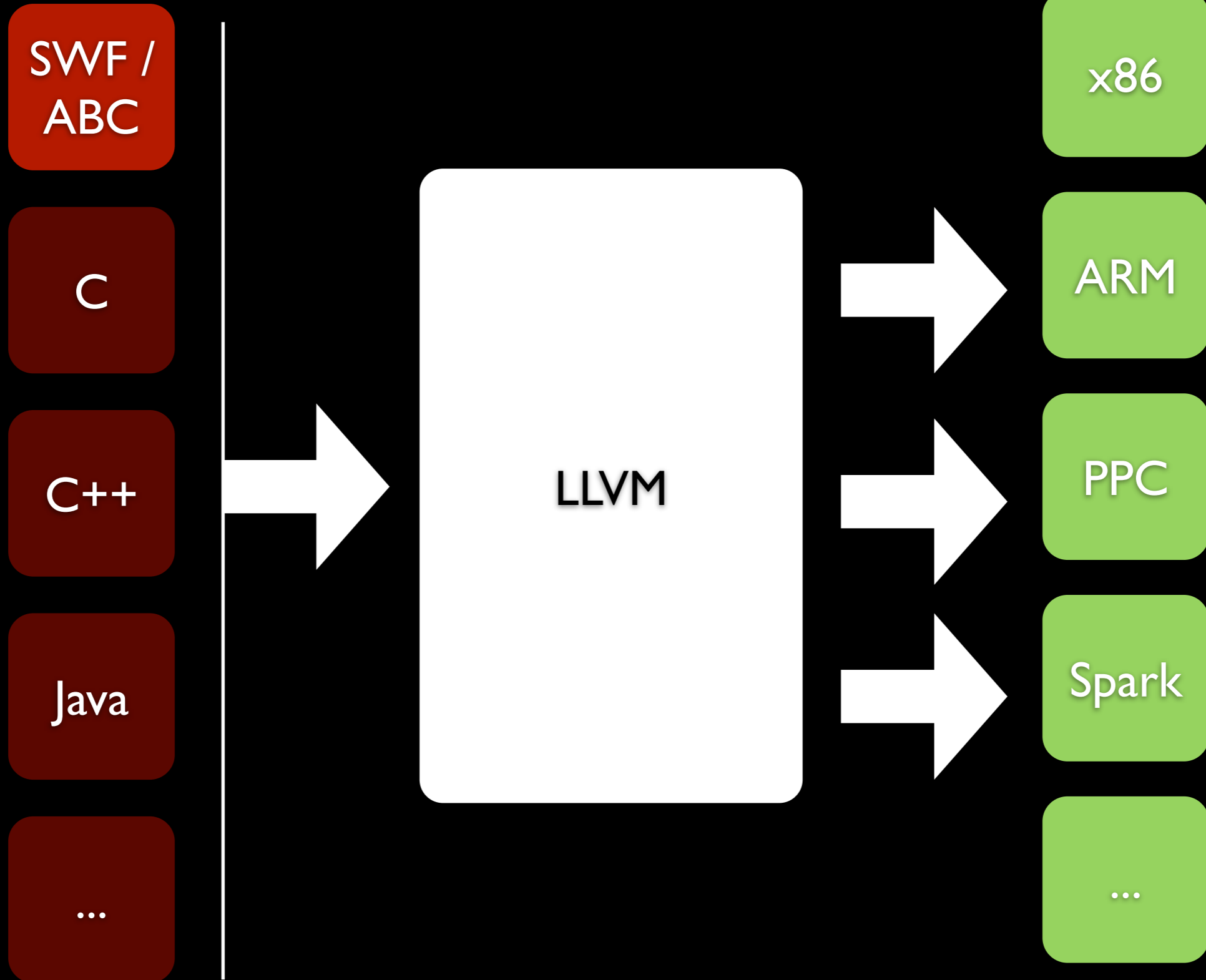
Capable of generating machine code for various targets including x86 and ARM processors

Used in Alchemy

Front End

Compilation / Optimization

Back End



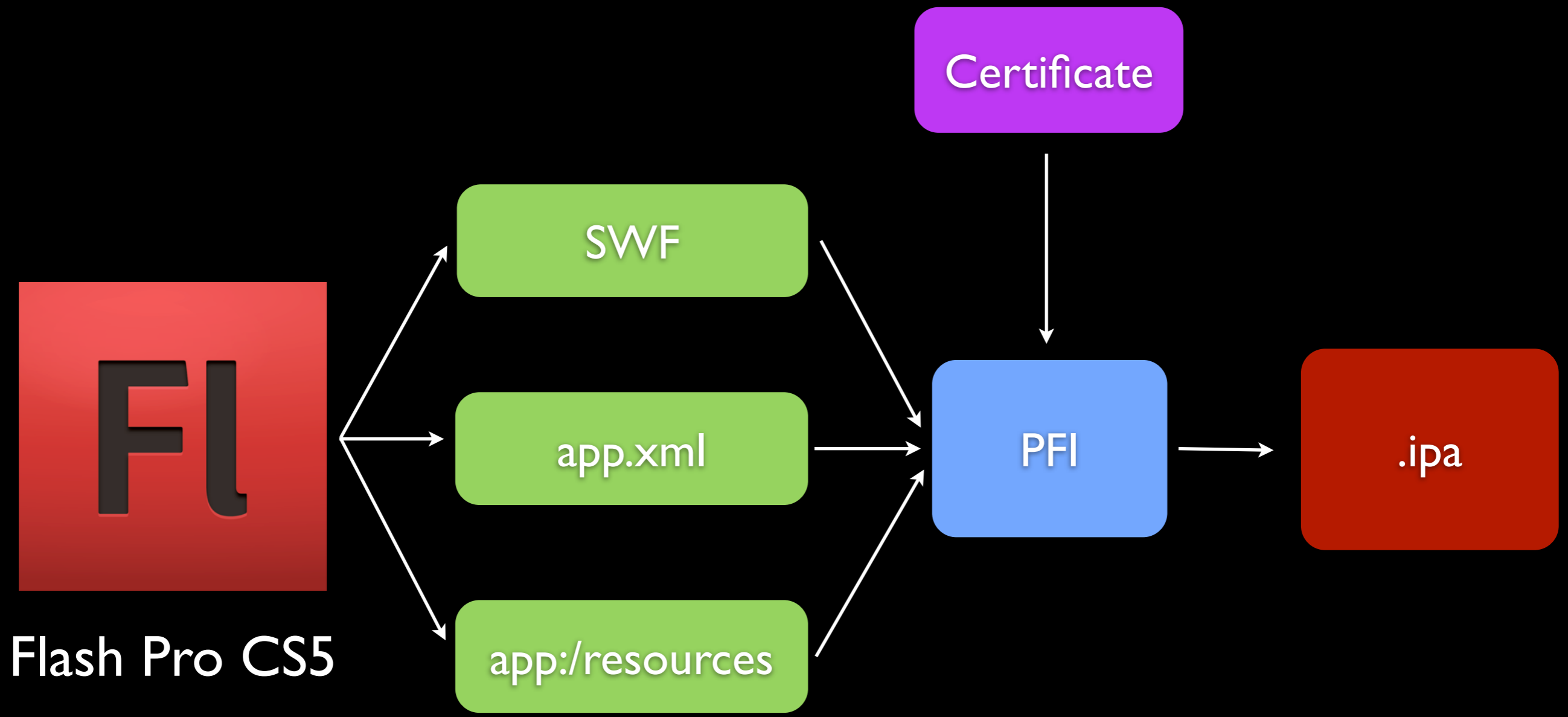
Adobe built a SWF / ABC front-end for LLVM

Develop

Publish

Package

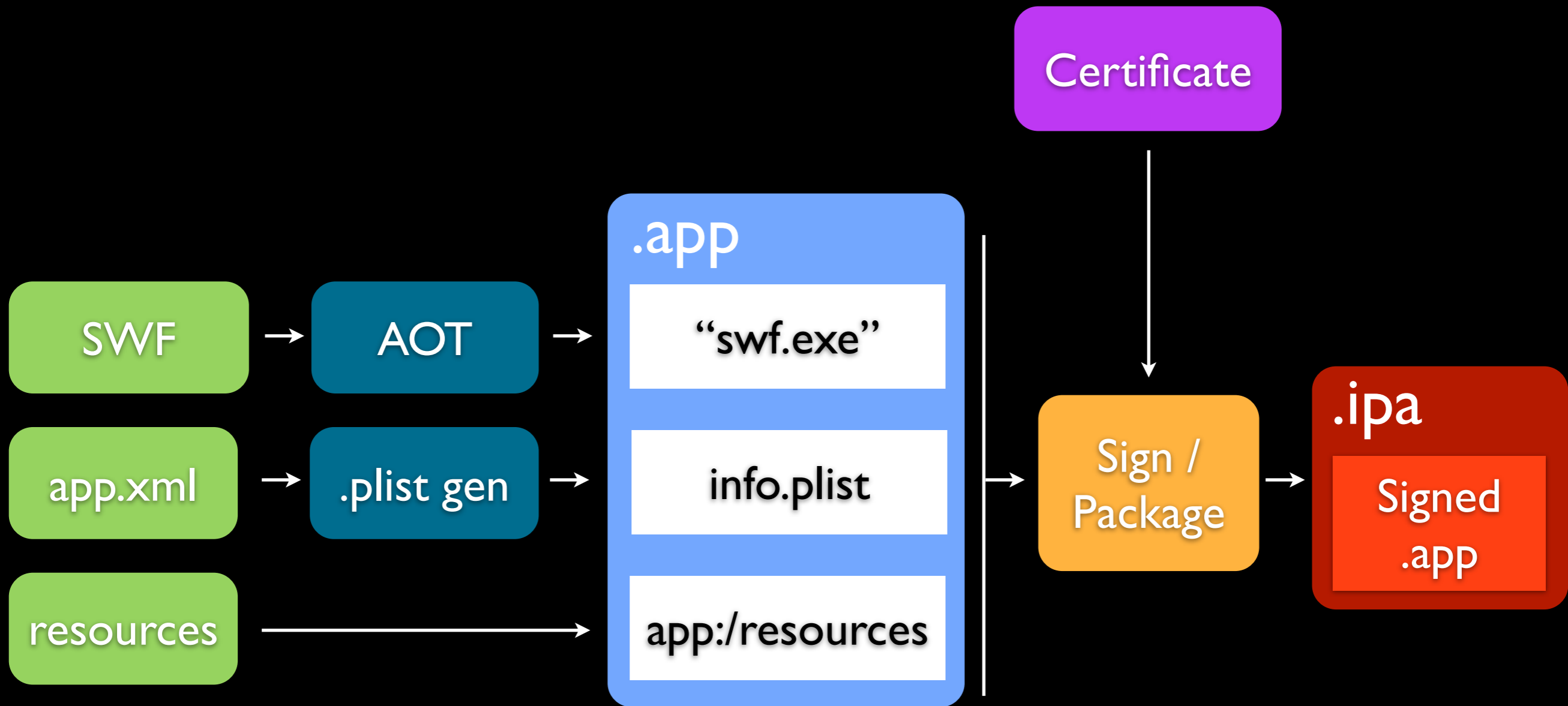
Deploy



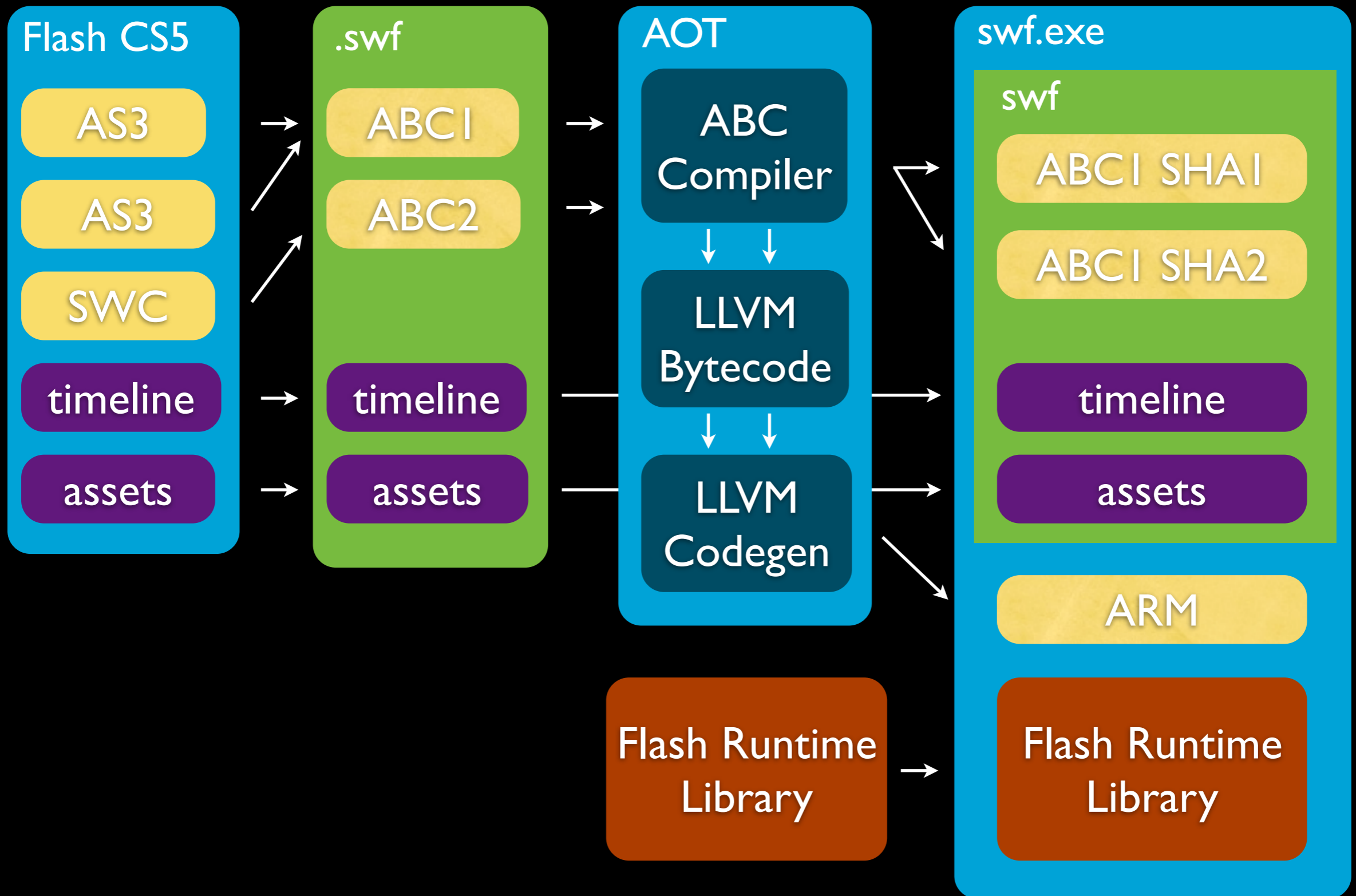
# AOT Compilation

- Ahead of Time (AOT) compilation
- iPhone License Restricts interpreters
  - Cant JIT code
  - No Interpreter
- Compiles ABC bytecode from SWF
- LLVM Base compile toolchain

# PFI



# AOT Compilation



AOT Compilation. Notice that there is a library form of the Flash Player runtime included in the application. ABC bytecode is compiled to native arm code.

# Developing Content



# APIs

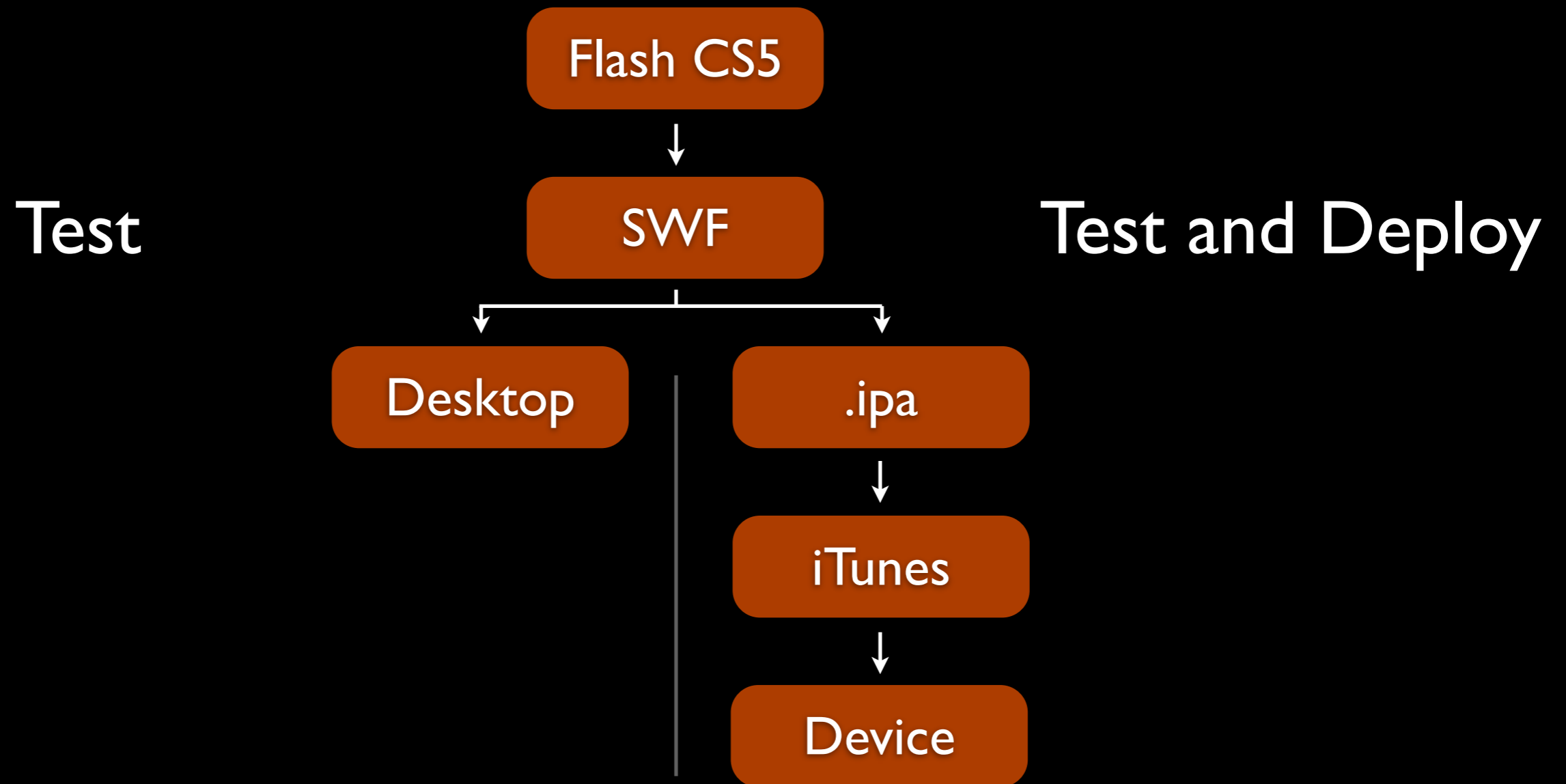


Flash Player 10.1



Adobe AIR 2.0

# Development Workflow





# iPhone Dev Center

Hi, Guest | Register | Log In

Log in to get the most out of the iPhone Dev Center. [Log in](#)

The iPhone Dev Center provides access to technical resources and information to assist you in developing with the latest technologies in iPhone OS. Log in with your Registered iPhone Developer Apple ID and password, or sign-up as a free Registered iPhone Developer today.

## Developing for iPhone OS 3.1

Search iPhone Reference Library

### Technical Documentation



#### Getting Started Documents

Developers new to iPhone OS can read about the tools, frameworks, development best-practices, and design methods for creating innovative world-class iPhone applications.



#### iPhone Reference Library

Explore a collection of in-depth technical documentation, sample code, guides, and articles for iPhone development categorized by topic and frameworks.

### Featured Content

- [iPhone Application Programming Guide](#)
- [iPhone Development Guide](#)
- [iPhone Human Interface Guidelines](#)
- [Your First iPhone Application](#)
- [Learning Objective-C: A Primer](#)

## iPhone Developer Program

### App Store Resource Center

Find details on everything from how to prepare for submitting an app to managing an app once it's been posted. [Log in](#)



### News and Announcements

Check out this regularly updated section for a range of information including tips on submitting apps, turnaround time for app review, and more. [Log in](#)



To access iPhone SDK 3.1.2 and additional technical resources and information, [log in](#) with your Registered iPhone Developer Apple ID and password, or [sign up](#) as a free Registered iPhone Developer today.

### Download iPhone SDK 3.1.2

Registered iPhone Developers can download iPhone SDK 3.1.2, which includes the Xcode IDE, iPhone



### Getting Started Videos

Watch Apple experts discuss everything from getting started with iPhone SDK, to the tools and



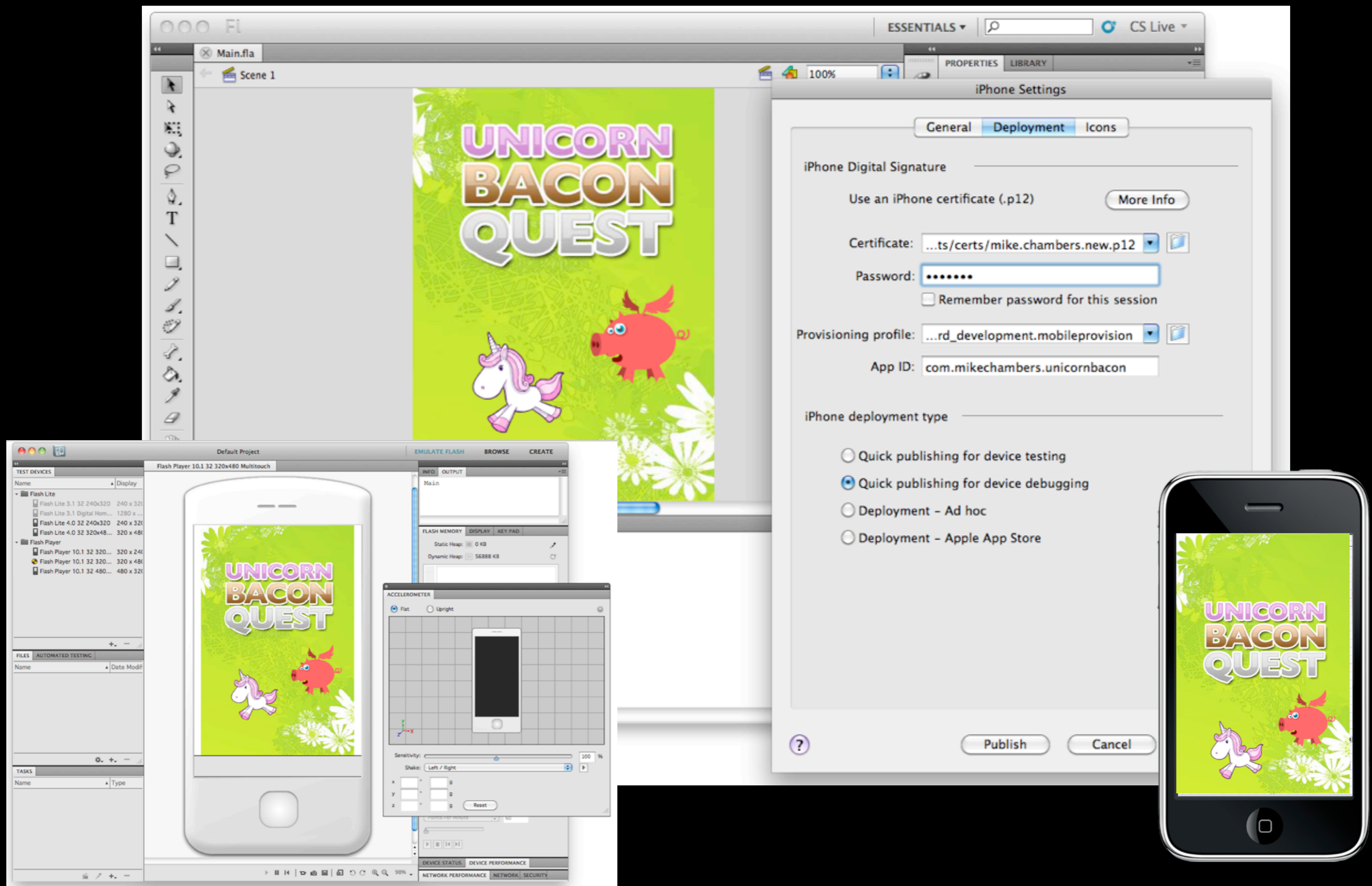
Watch Apple experts discuss everything from getting started with iPhone SDK, to the tools and

### Join the iPhone Developer Program

The iPhone Developer Program offers a complete process for developing and distributing iPhone or iPod touch applications. [Learn More](#)

The iPhone Developer Program offers a complete process for developing and distributing iPhone or iPod touch applications. [Learn More](#)

In order to deploy an ipa / app to the device, you must be part of the Apple developer program, and have the appropriate certificates and provisioning profiles setup.



# Performance Tips and Tricks

- Most will improve performance / memory usage on desktop
- General ActionScript performance techniques also apply to device



! =

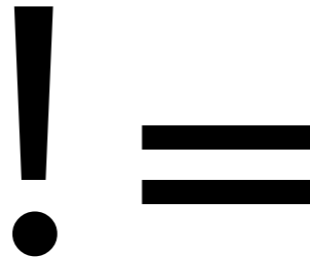


iPhone development is mobile development. It has a significantly slower processor than what you are used to on your desktop.

Screen Size

UI Interactions

Performance



This affects Screen Size, UI Interactions and most importantly performance. We will focus on performance.

# Test and Profile Code



## Profile External Application



# AS3 Performance Testing Harness

Grant Skinner

[bit.ly/as3performance](http://bit.ly/as3performance)

```
performancetests.GraphicsTests (5 iterations)
Testing different approaches for drawing.

method.....ttl ms...avg ms
tare [2]                0      0.00
drawPath                104     20.80
drawPathShort           107     21.40
fullPath                142     28.40
reference                103     20.60
shortReference          105     21.00
withGraphics            939    187.80

performancetests.Functions (5 iterations)
Testing impact of function COs.

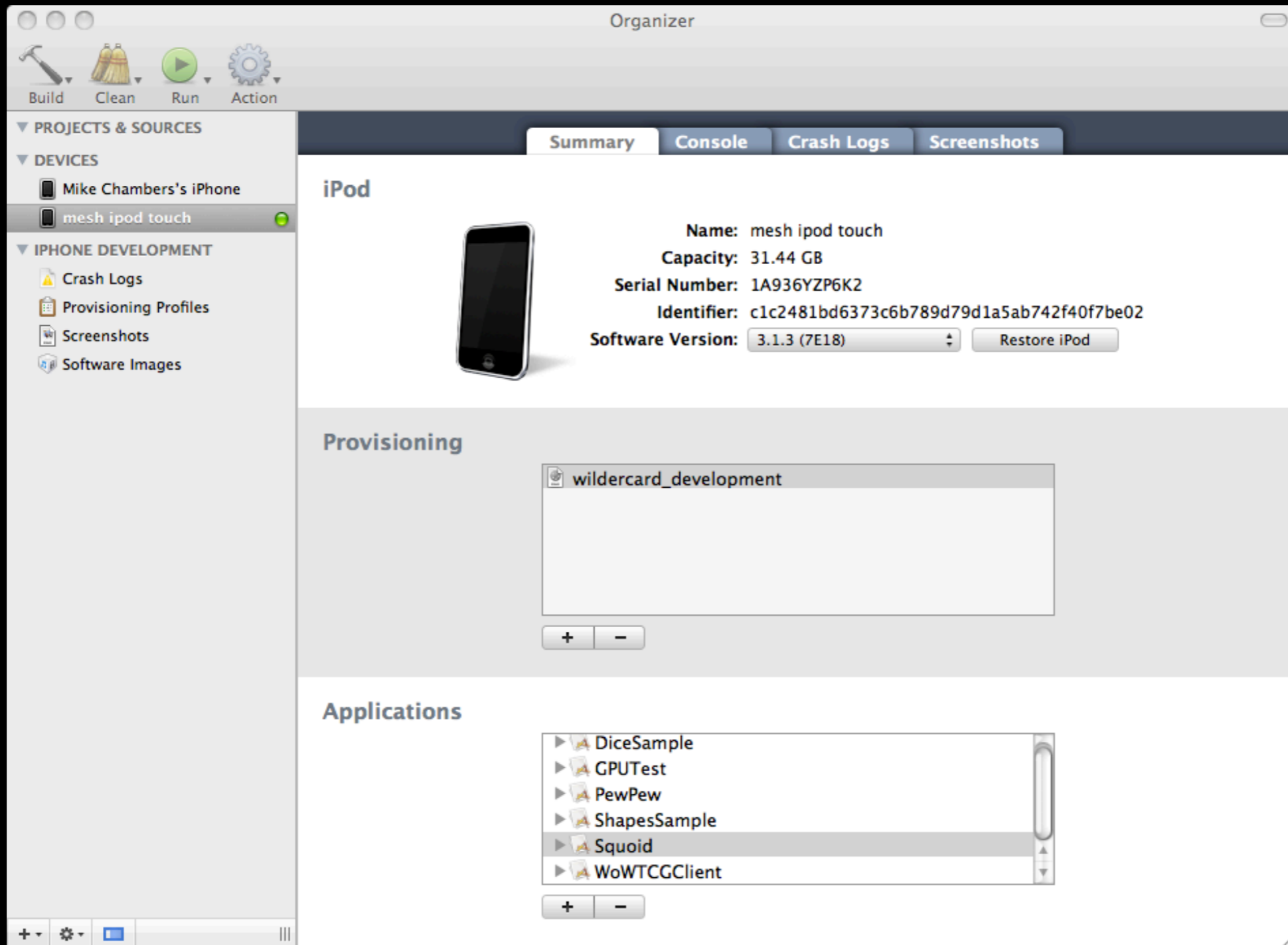
method.....ttl ms...avg ms
tare [3]                3       0.60
anonymous               707    141.40
anonymousRef            92     18.40
method                  30      6.00
reference                80     16.00
```

Make sure to test optimization results.

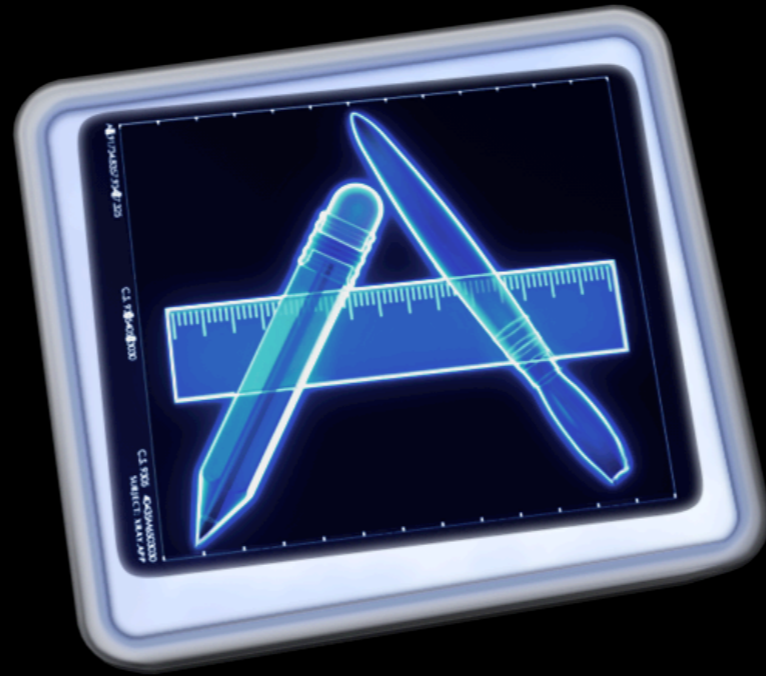
# Remote Device Debugging from Flash CS5



Debug > Begin Remote Debug Session > ActionScript 3



Mac Only



# Instruments

Mac Only



Shark

Mac Only

# Simple Game Framework

[github.com/mikechambers](https://github.com/mikechambers)

# Rendering

- Hardware Composition is available for DisplayObjects
- Can greatly improve performance
- Uses OpenGL ES-1
- APIs will be available on additional platforms



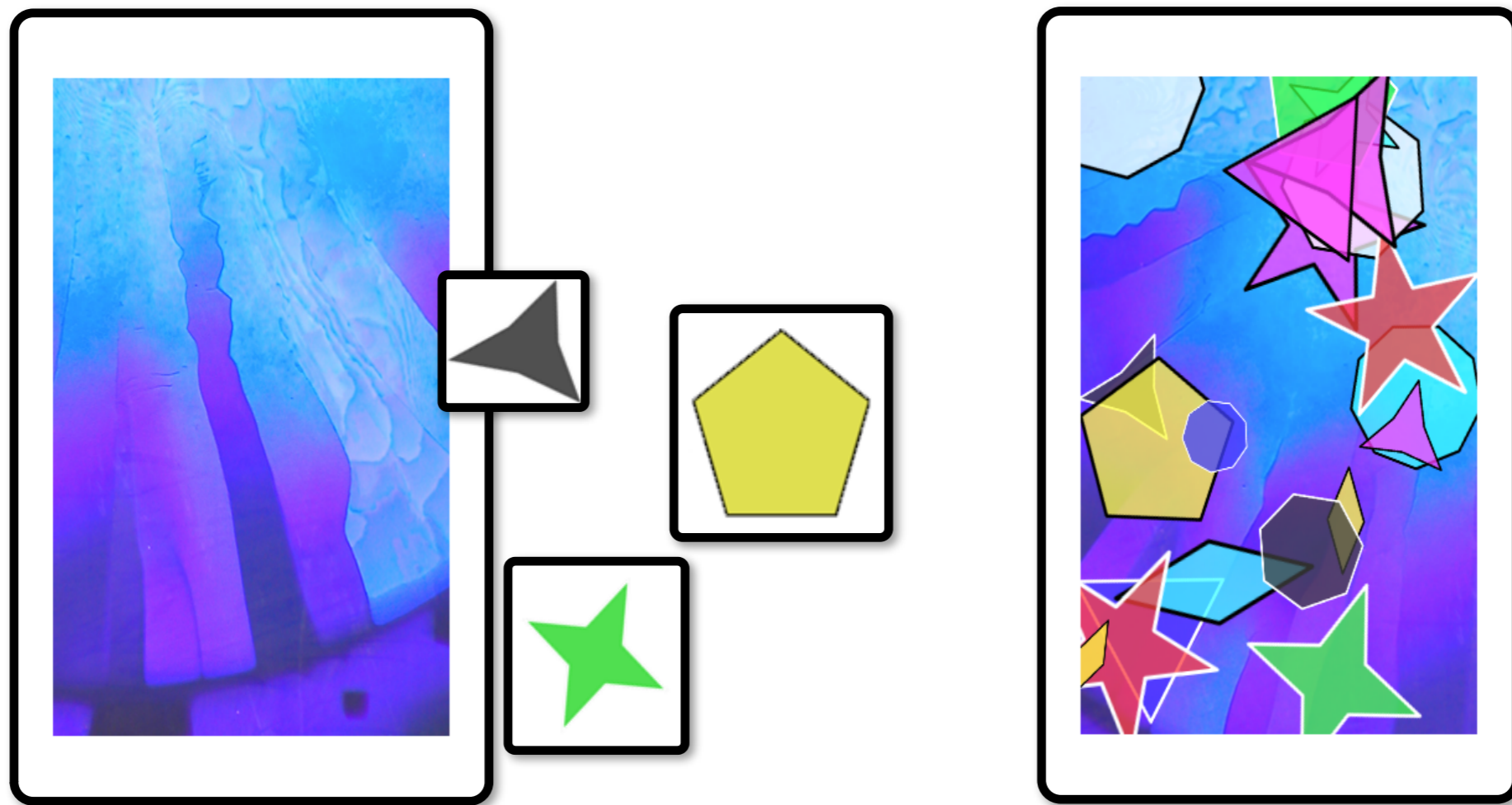


# Software Rendering



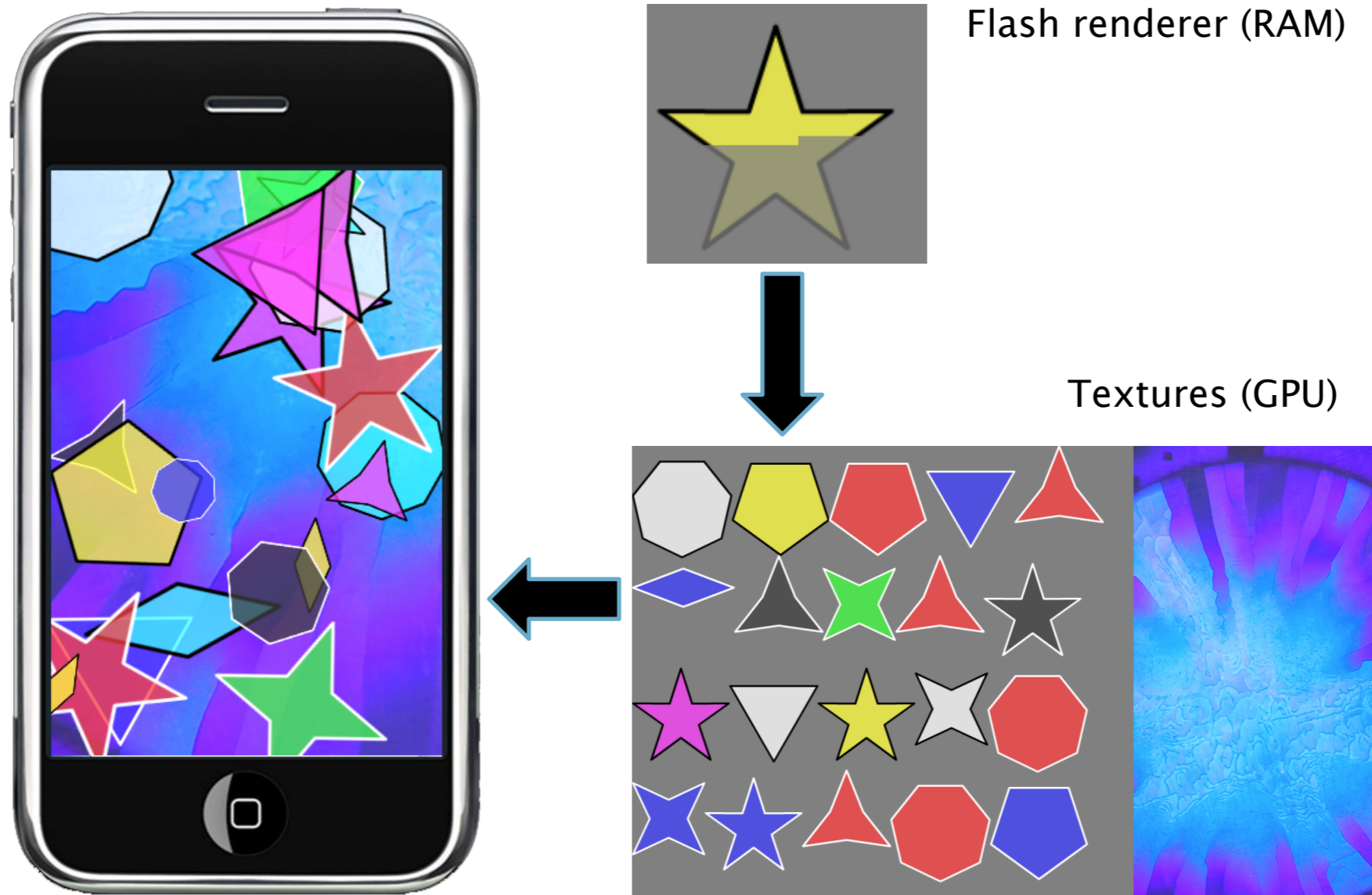
Software rendering renders entire display (including composition) via software. CPU intensive.

# GPU Compositing



CPU Composition. Individual elements rendered via software, but are then composited / put together in hardware. Can be very fast.

# GPU Pipeline with Cached Surfaces



Individual elements can be cached and then composited in hardware.

# Using GPU Composition

- `cacheAsBitmap: Boolean`
- `cacheAsBitmapMatrix: Matrix`
- Using 2.5D Apis
  - Such as setting `z` property



NEW!

```
<initialWindow>  
  <renderMode>gpu</renderMode>  
  ...  
</initialWindow>
```

# cacheAsBitmap

## cacheAsBitmap

- Geometric Translations
- Changes in  $X / Y$
- Tweens

## cacheAsBitmapMatrix

- Geometric Dilation
- Rotation
- Scaling

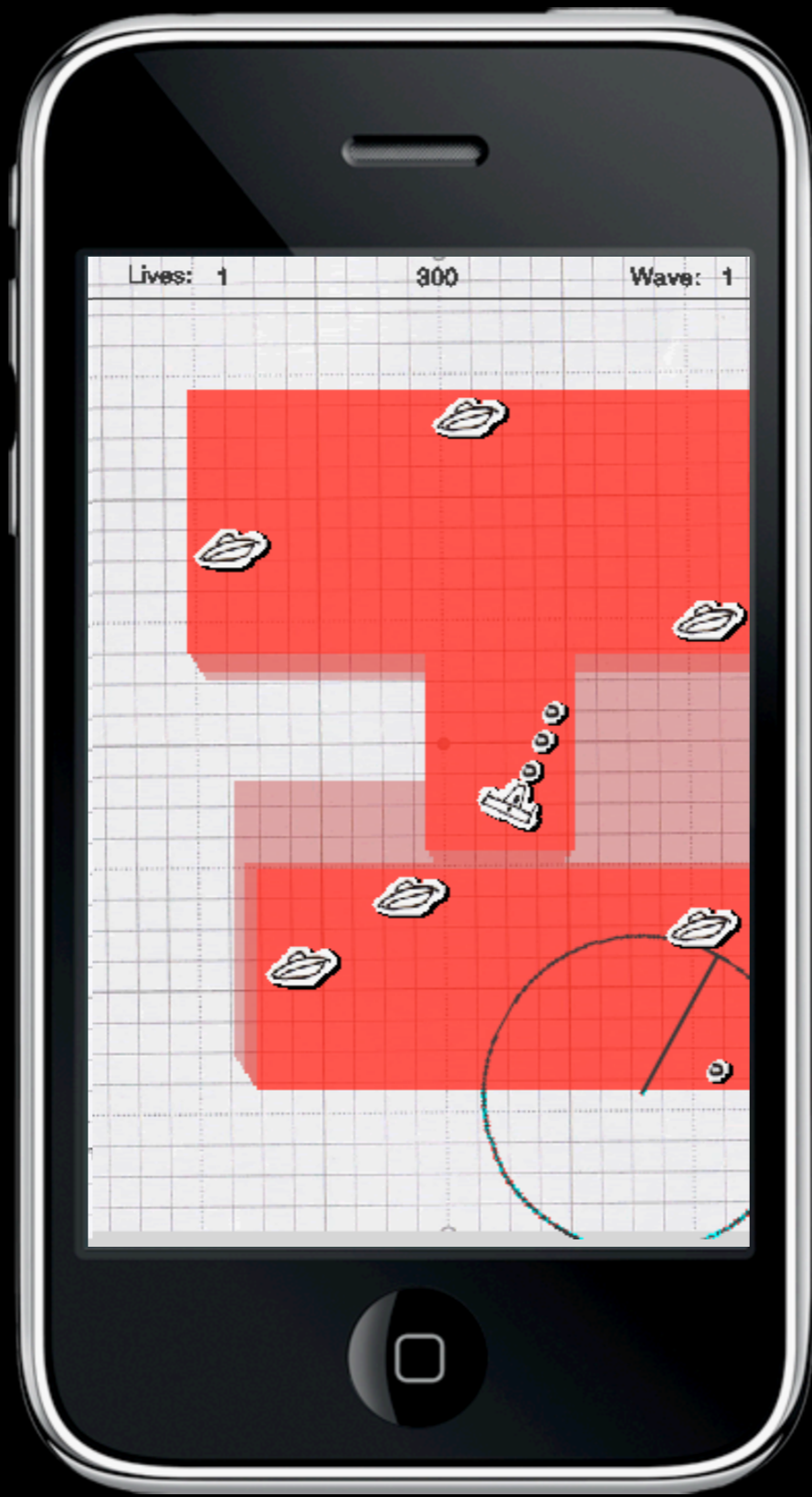
- Can set `cacheAsBitmap` in IDE or Code
- Have to subclass `DisplayObject` to set `cacheAsBitmapMatrix` (can't set in IDE)
- SGF : `CachedSprite.as`

# Debugging HWA

## CTTextureUploadTracking

```
<iPhone>  
  <InfoAdditions><![CDATA[  
    <key>CTTextureUploadTracking</key>  
    <true/>  
  ]]></InfoAdditions>  
</iPhone>
```

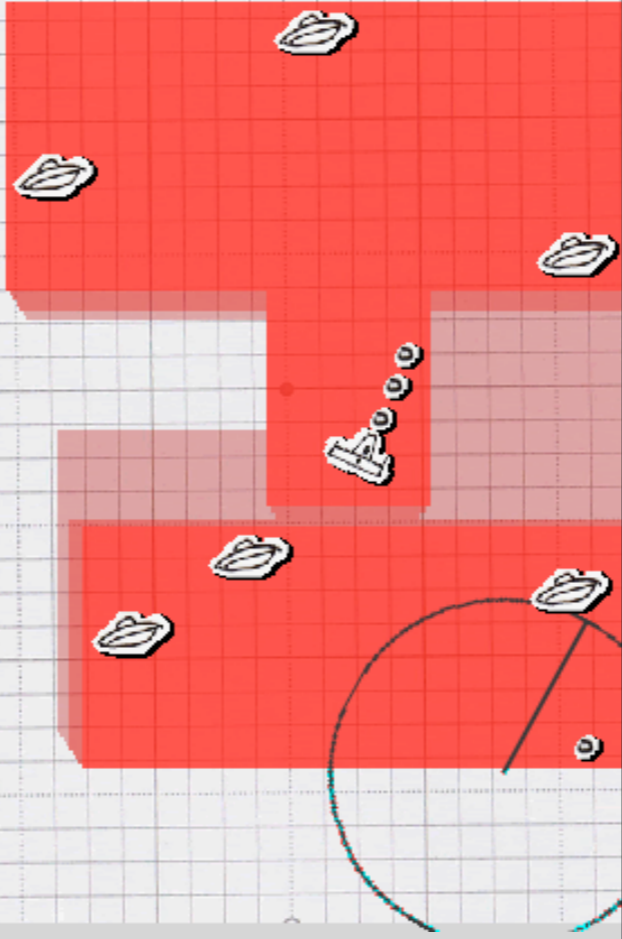




Lives: 1

300

Wave: 1



Keeo

# Use GPU!!!

- Hardware accelerate everything
  - Except for items that re-draw often
- Break up display objects to minimize re-draws
- Don't cache clips that have children that move / change
- Keep display list shallow

# Prevent Redraws

- Items moving over non-cached DisplayObjects
- Visual content changes
  - Drawing API
  - Children moving / changing
- ColorTransformations
- Removing from display list

# Minimize Instance Allocations

- Reuse object instances
- Memory Allocation very expensive
- Reduces Garbage collection
- Reduces CPU / Initialization costs

```
private function doSomething():void
{
    for(var i:int = 0; i < 100; i++)
    {
        var p:Point = new Point();
        p.x = 5;
        p.y = i * 5

        checkPoint(p);
    }
}
```

```
private function doSomething():void
{
    var p:Point = new Point();
    for(var i:int = 0; i < 100; i++)
    {
        p.x = 5;
        p.y = i * 5

        checkPoint(p);
    }
}
```

```
private function foo():void
{
    var s:Vector.<Number> = new Vector().<Number>;
    //...
}
```

```
private var s:Vector.<Number> = new Vector().<Number>();
private function foo():void
{
    var len:int = s.length;
    for(var i:int = 0; i < len; i++)
    {
        s[i] = null;
    }

    //...
}
```



```
private var s:Vector.<Number> = new Vector().<Number>();
private function foo():void
{
    s.length = 0;

    //...
}
```

# Object Pooling

- Reuses Object instances
- Avoid constant initialization and garbage collection
- Particularly useful for DisplayObjects composited by GPU

# Pooling DisplayObjects

- Re-use cached DisplayObjects
- Keep on stage (off screen)
- Can give major performance boosts
- PewPew

GameObjectPool.as  
SoundManager.as

# Time Management

- In General, ENTER\_FRAME performs better than Timer
- Use single listener, and then dispatch
- SGF :TickManager.as

# Event Dispatching

- Can be very expensive
- Requires several memory allocations
- Consider using callbacks in CPU intensive areas
- Consider Reusing Event instances

```
private function doSomething():void
{
    //...
    dispatchEvent(new Event("done"));
}

private function doSomethingBetter(callback:Function):void
{
    //...
    callback();
}
```

# Event Propagation

- Can be very expensive, especially on display list instances.
- `Event.stopPropagation();`
- `Event.stopImmediatePropagation();`



```
private function onClick(e:MouseEvent):void
{
    e.stopPropagation();

    //... do stuff
}
```

# Mouse / Touch Events

- Can disable with
  - `mouseEnabled`
  - `mouseChildren`
- Don't use `MouseEvent.MOUSE_MOVE`
  - Check Mouse position at interval
  - Example

```
private function init():void
{
    addEventListener(Event.ENTER_FRAME, onEnterFrame);
}

private function onEnterFrame(e:Event):void
{
    this.x = mouseX;
    this.y = mouseY;
}
```

Can listen for `MOUSE_UP`  
`MOUSE_DOWN` events to toggle

# Optimizing Mobile Content for the Adobe Flash Platform

Thibault Imbert

[bytearray.org/?p=1363](http://bytearray.org/?p=1363)

# Mike Chambers



[www.mikechambers.com](http://www.mikechambers.com)

[twitter.com/mesh](https://twitter.com/mesh)

[mesh@adobe.com](mailto:mesh@adobe.com)

All links at : <http://www.mikechambers.com/blog/2009/10/17/resources-for-learning-more-about-flash-to-iphone/>

Baby Crying

<http://www.flickr.com/photos/bbaunach/1055569383/>

