

# Building iPhone Applications with Flash CS5

Mike Chambers  
Principal Product Manager  
Developer Relations  
Flash Platform



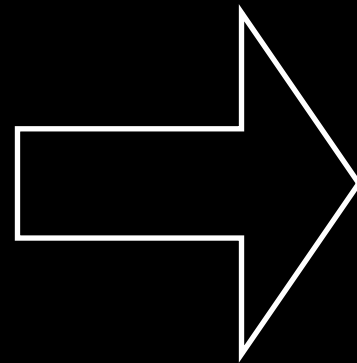
What did Adobe  
announce at Max?

# 2

Two key announcements concerning developing on the Flash Platform.



Flash Pro CS5



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



# Flash Player 10.1



Smartphone enabled

Multitouch, accelerometer, screen orientation

Optimized memory, power, hardware acceleration

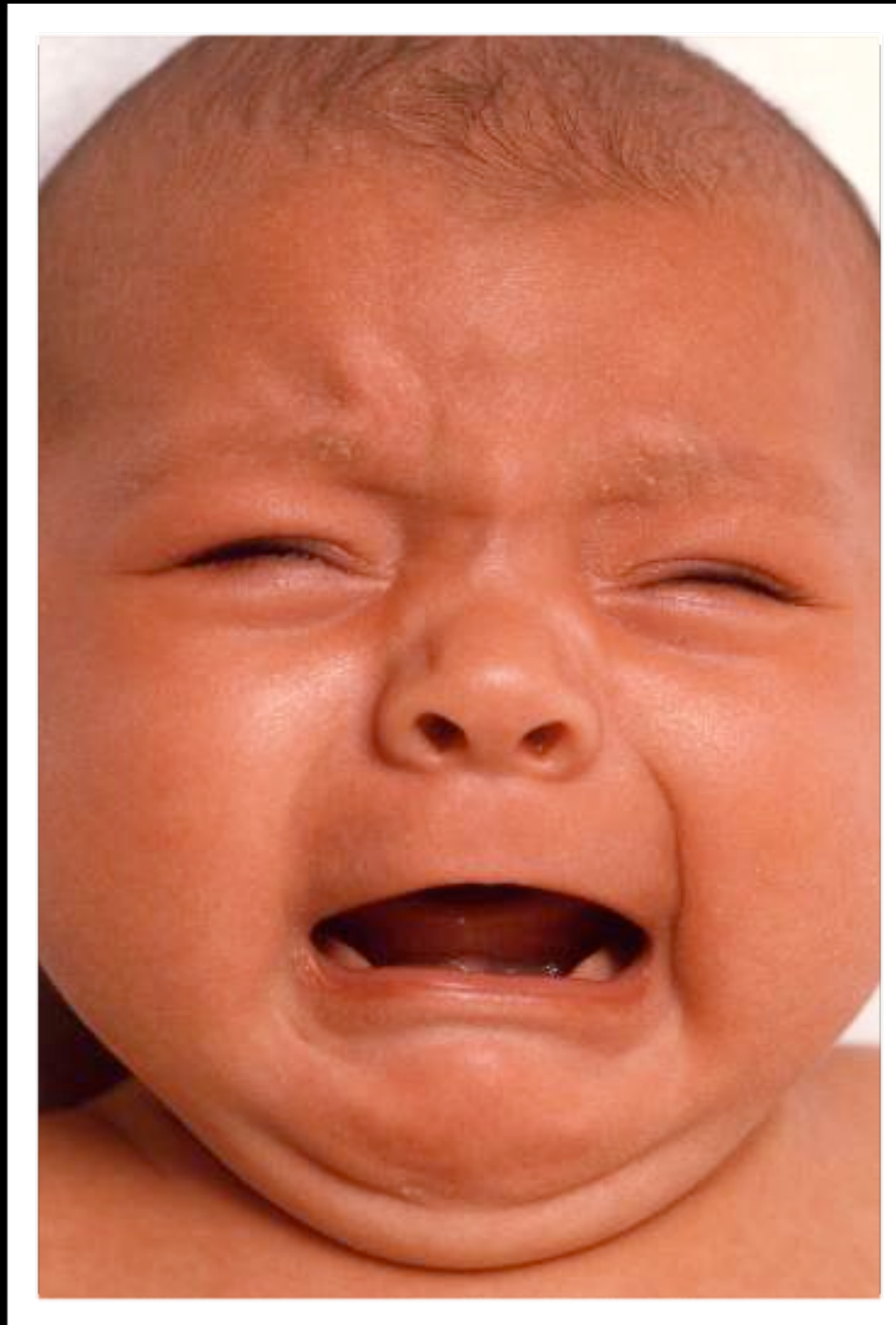
RAW Microphone Access

Global Error Handler





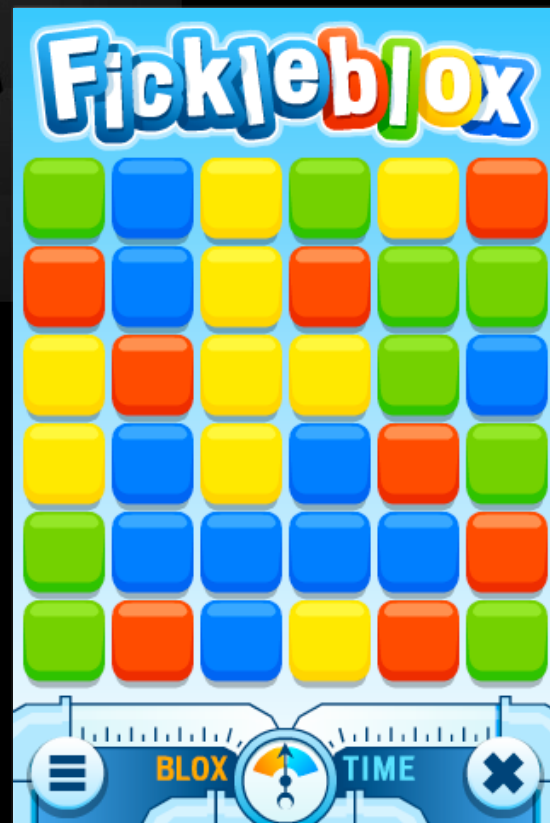
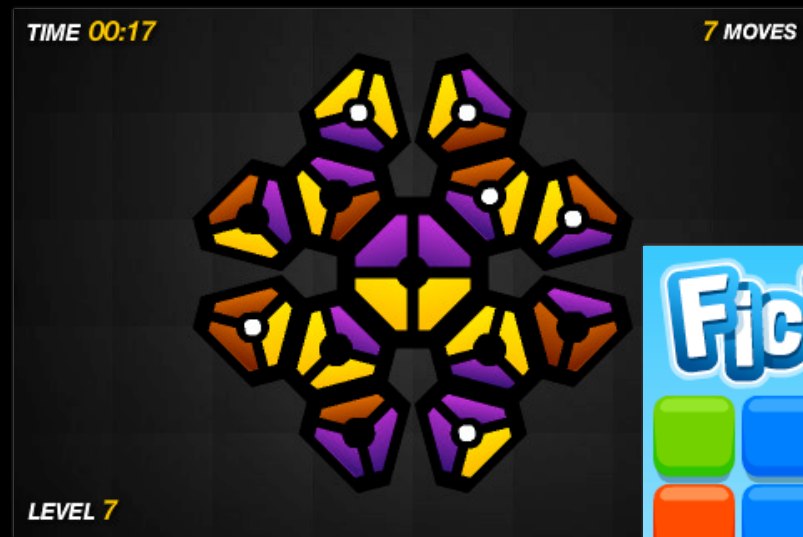
Flash Player 10.1 will be available across both browsers, desktops and devices. This includes in Adobe AIR 2.0, as well as Flash Player for Palm Pre, Android and Windows Mobile devices.



## Did not announce 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.



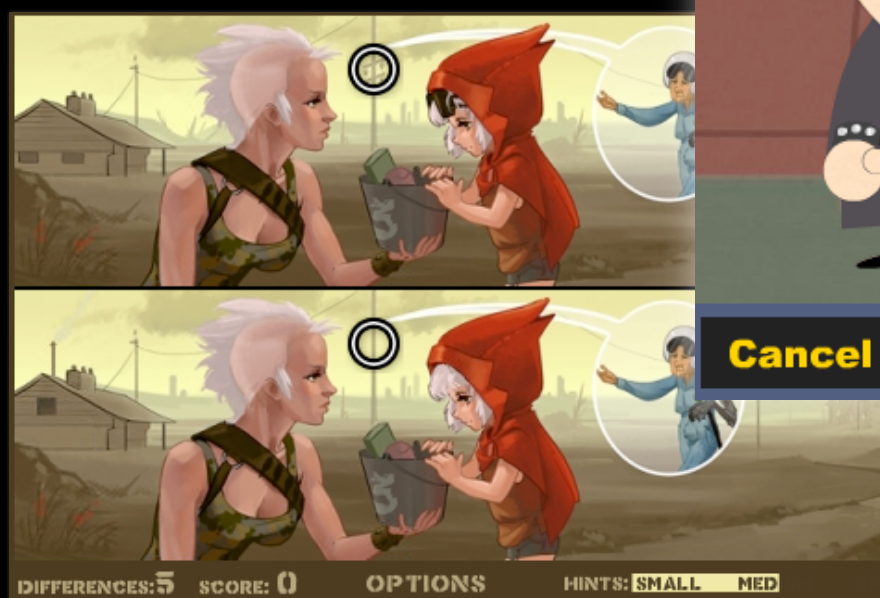


Play Game

High Scores

How To Play

Settings



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





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

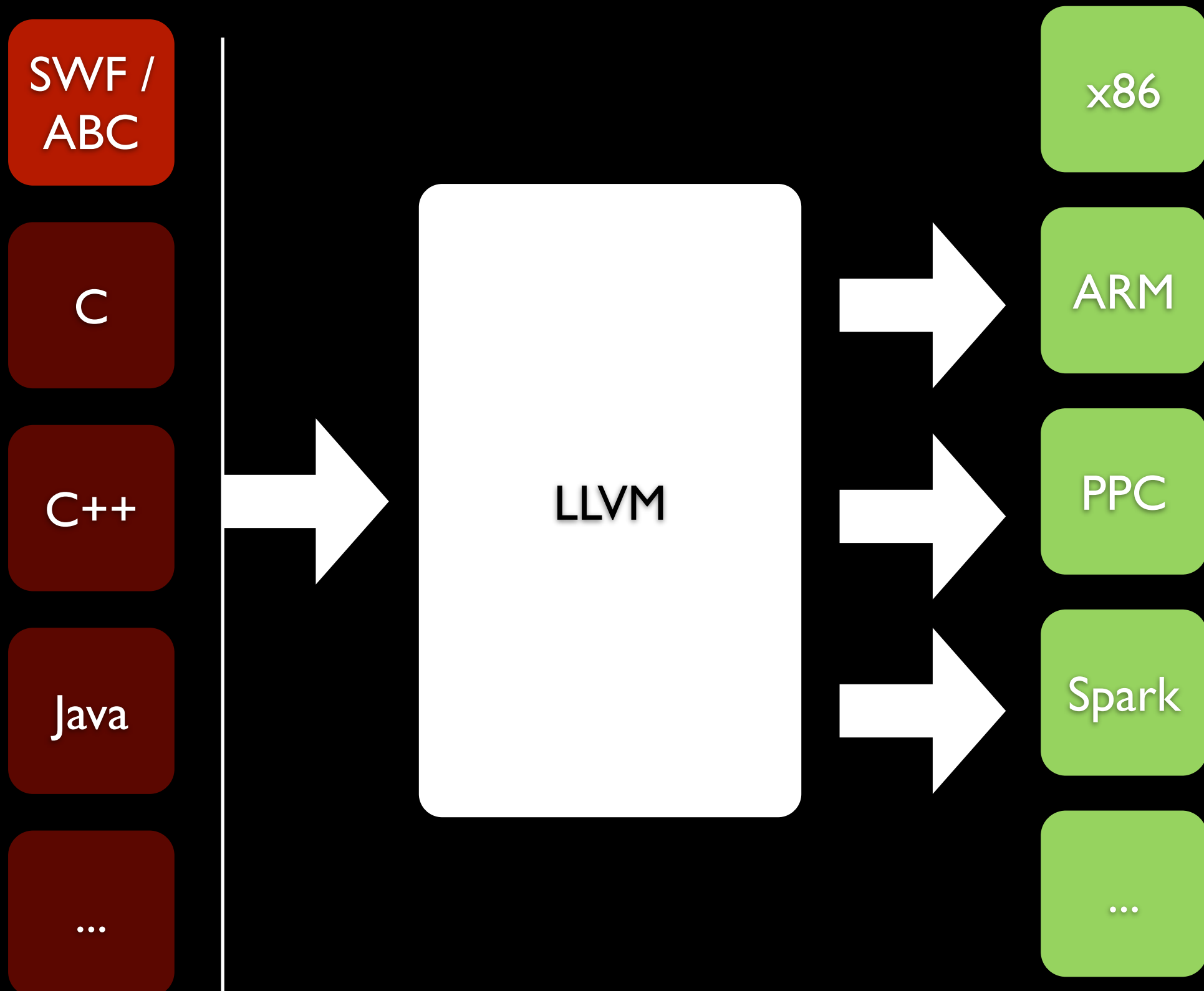
Used in Alchemy



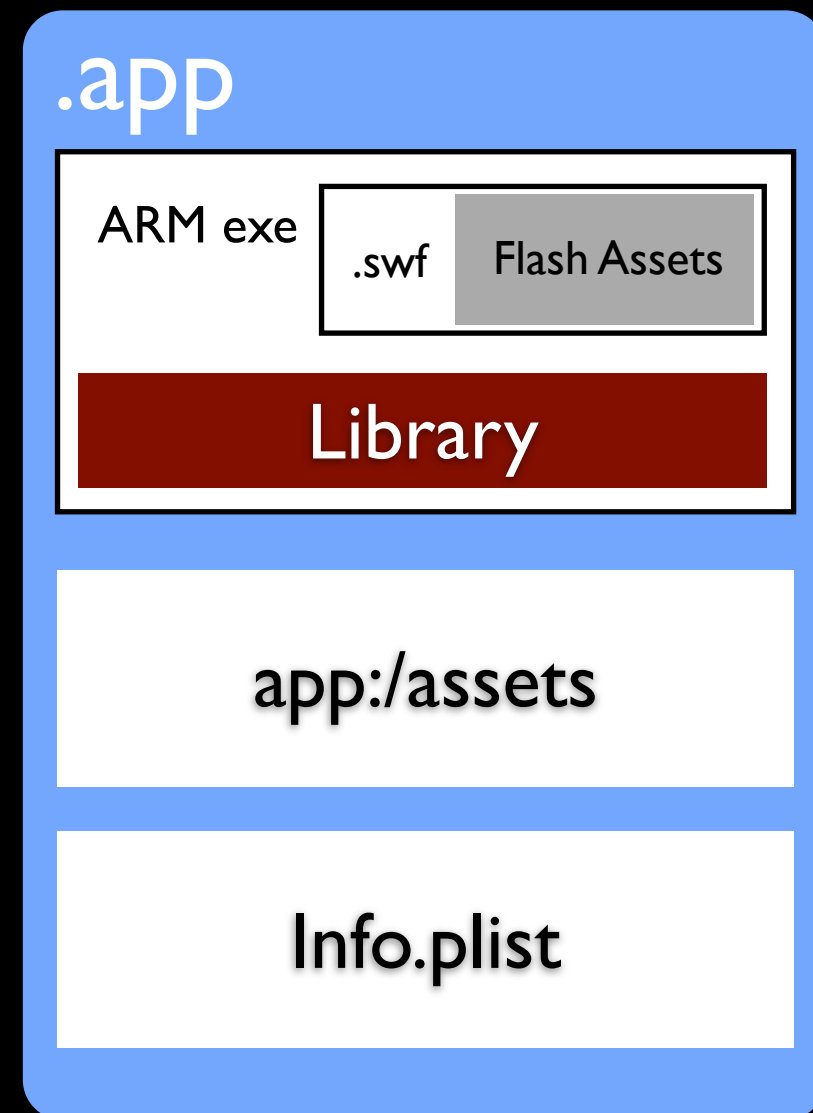
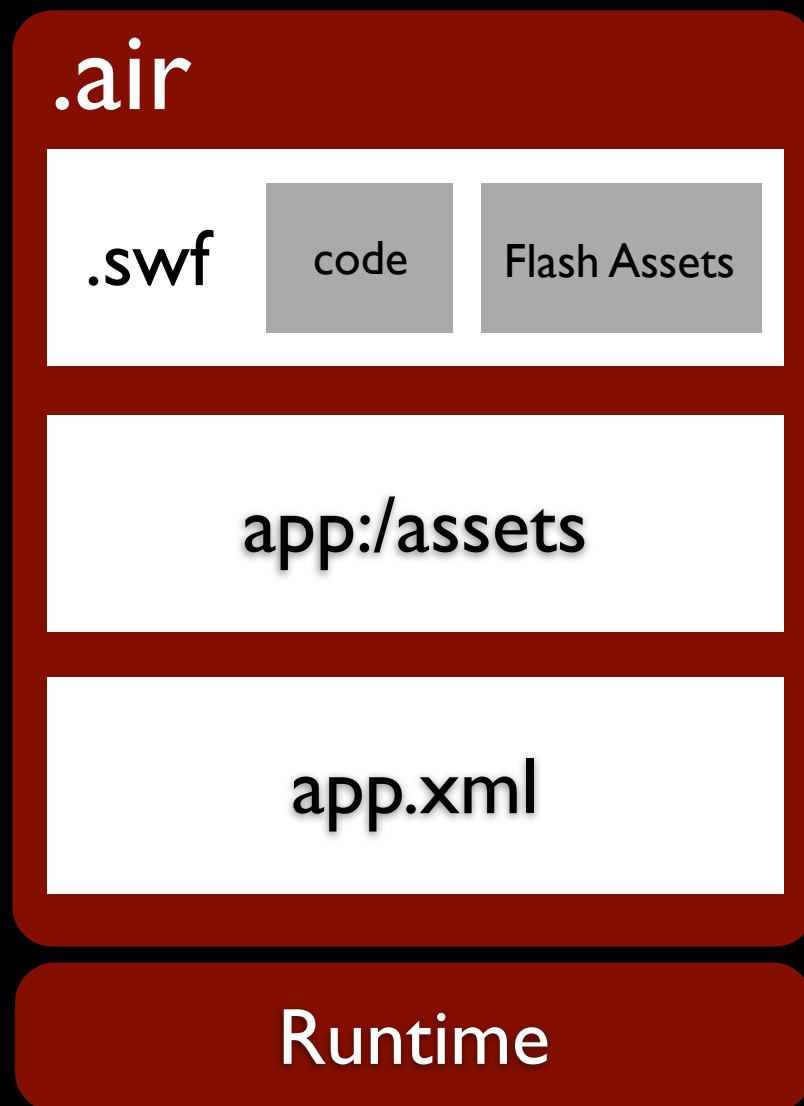
Front End

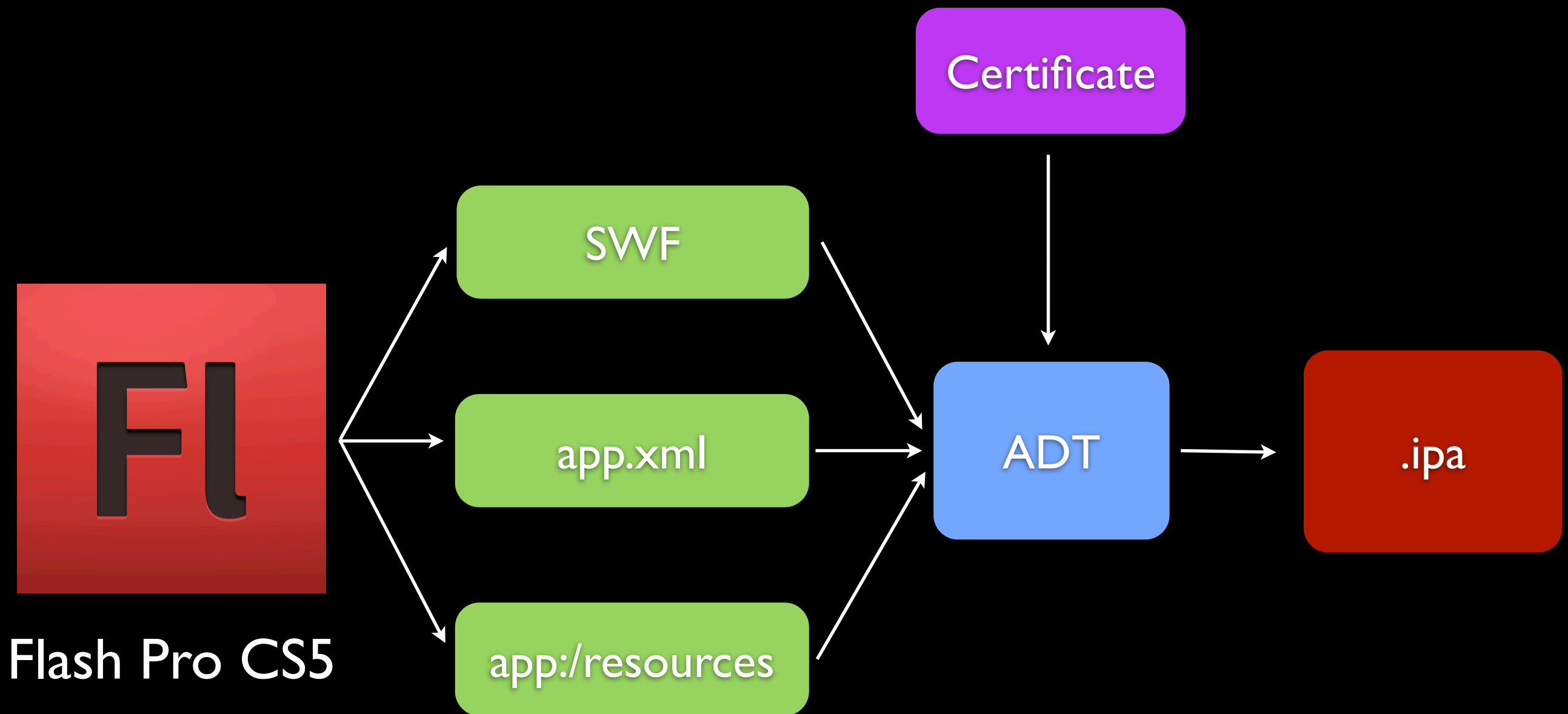
Compilation / Optimization

Back End



# .AIR vs .APP

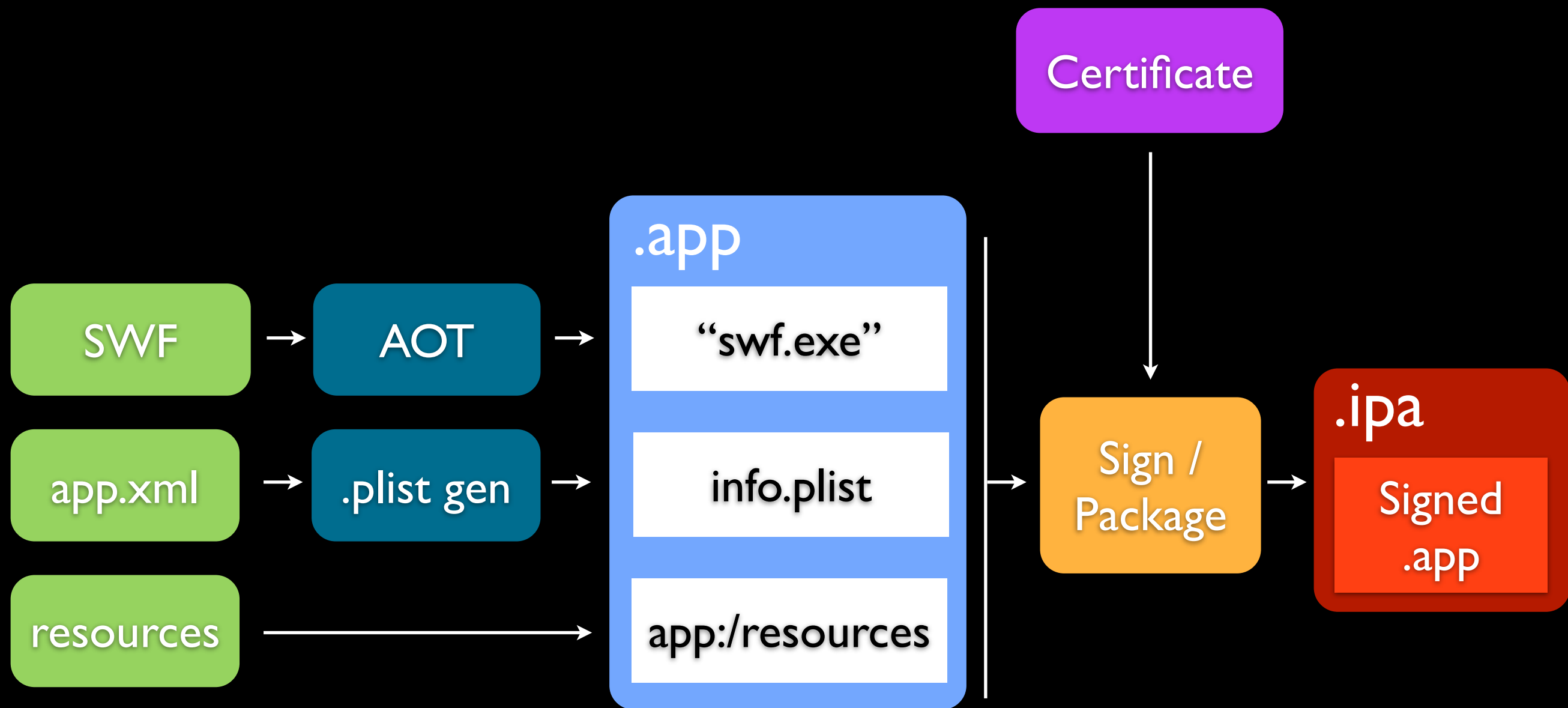




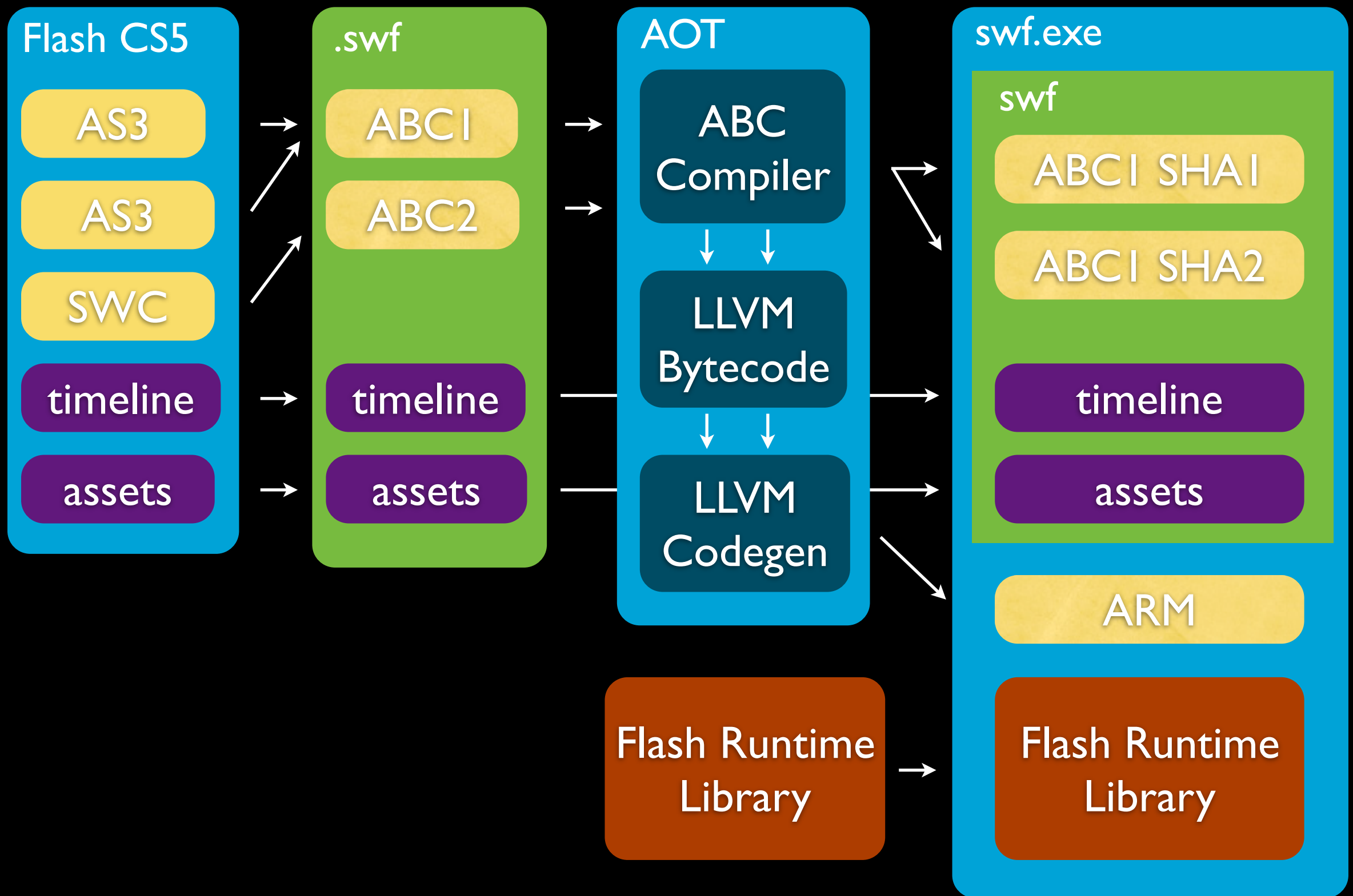
# AOT Compilation

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

# ADT



# AOT Compilation

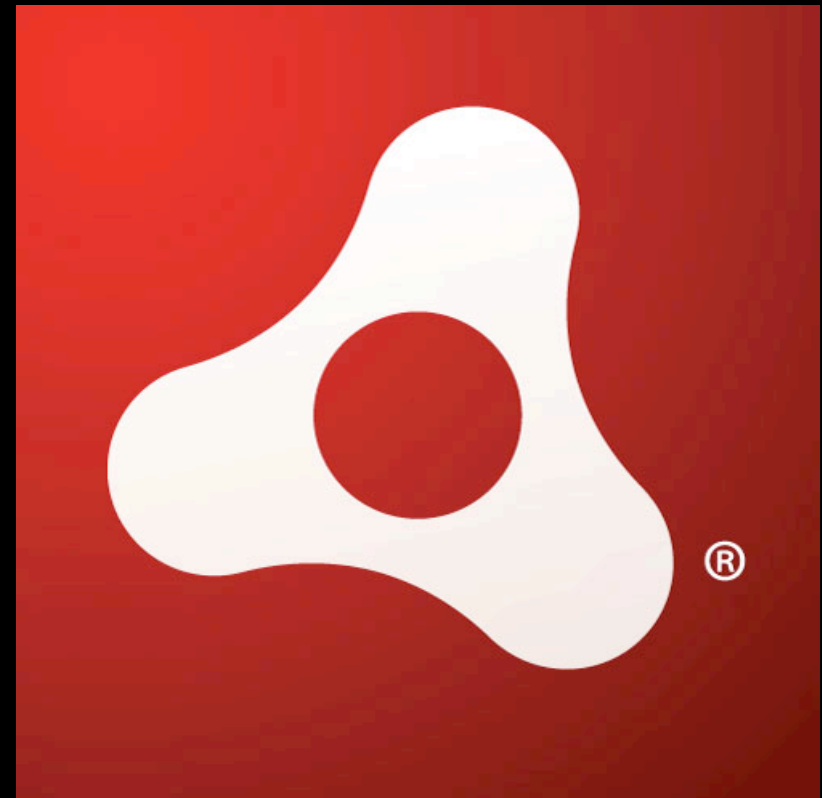


# Developing Content

# APIs



Flash Player 10.1

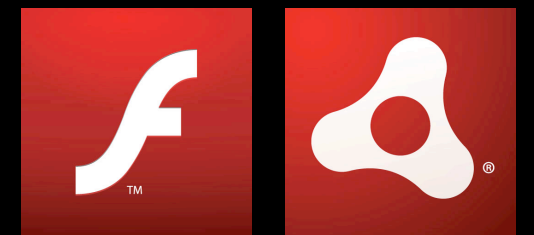


Adobe AIR 2.0



# New APIs

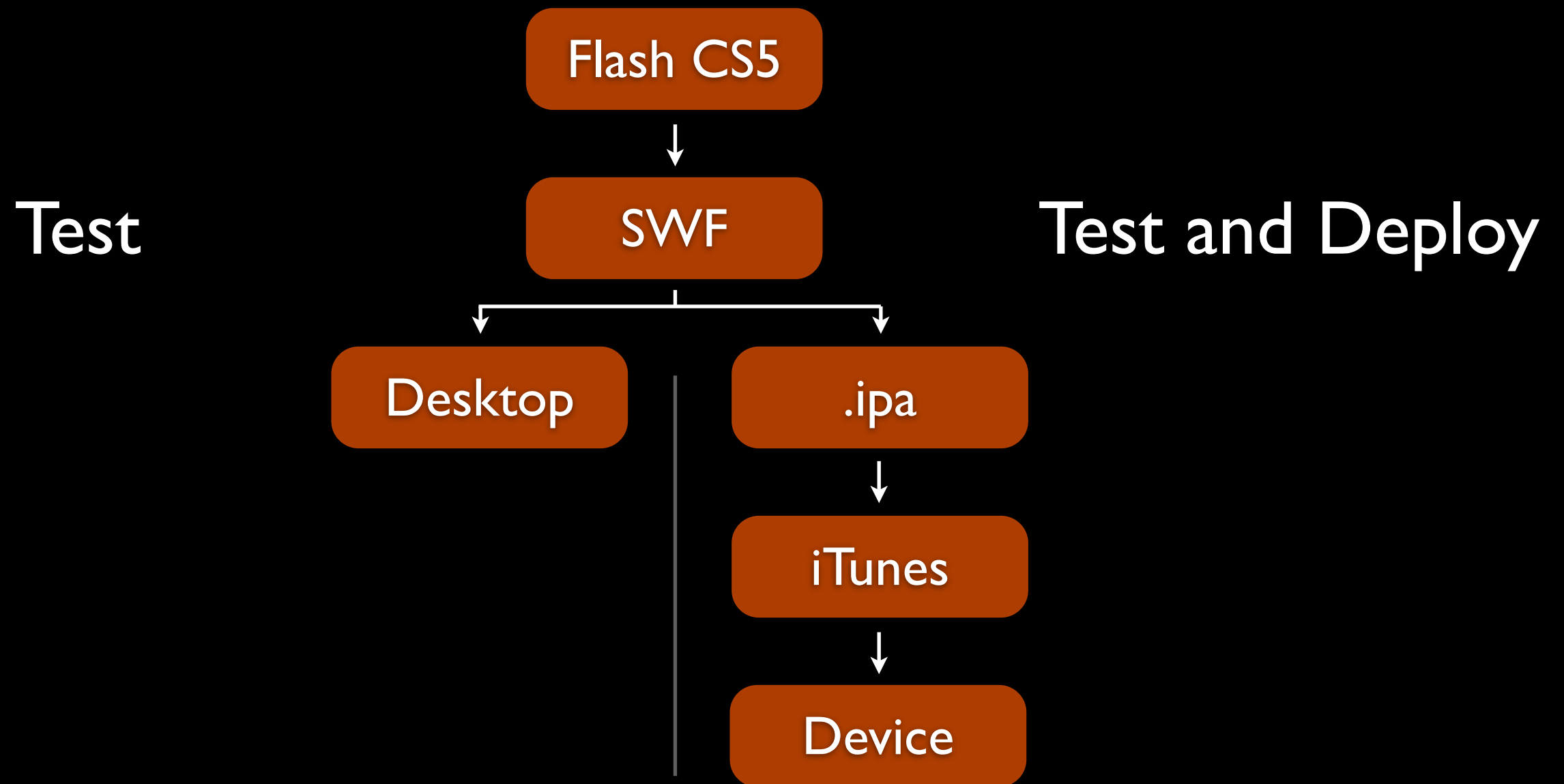
- MultiTouch
- Screen Orientation
- MediaLibrary
- Accelerometer
- Geo-location
- Cut / Copy / Paste
- Native TextInput
- tel:, mailto:, maps:, video:



# Do not reply on...

- Keyboard events outside of textfields
- Rollover Hover
- ActionScript 1 or 2
- Loading a SWF with ActionScript
- LocalConnection
- PixelBender
- H.264 and Speex codecs
- Microphone / Video Camera access

# Development Workflow



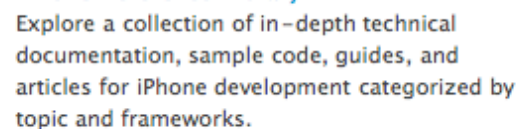


Hi, Guest | Register | Log In

Log in

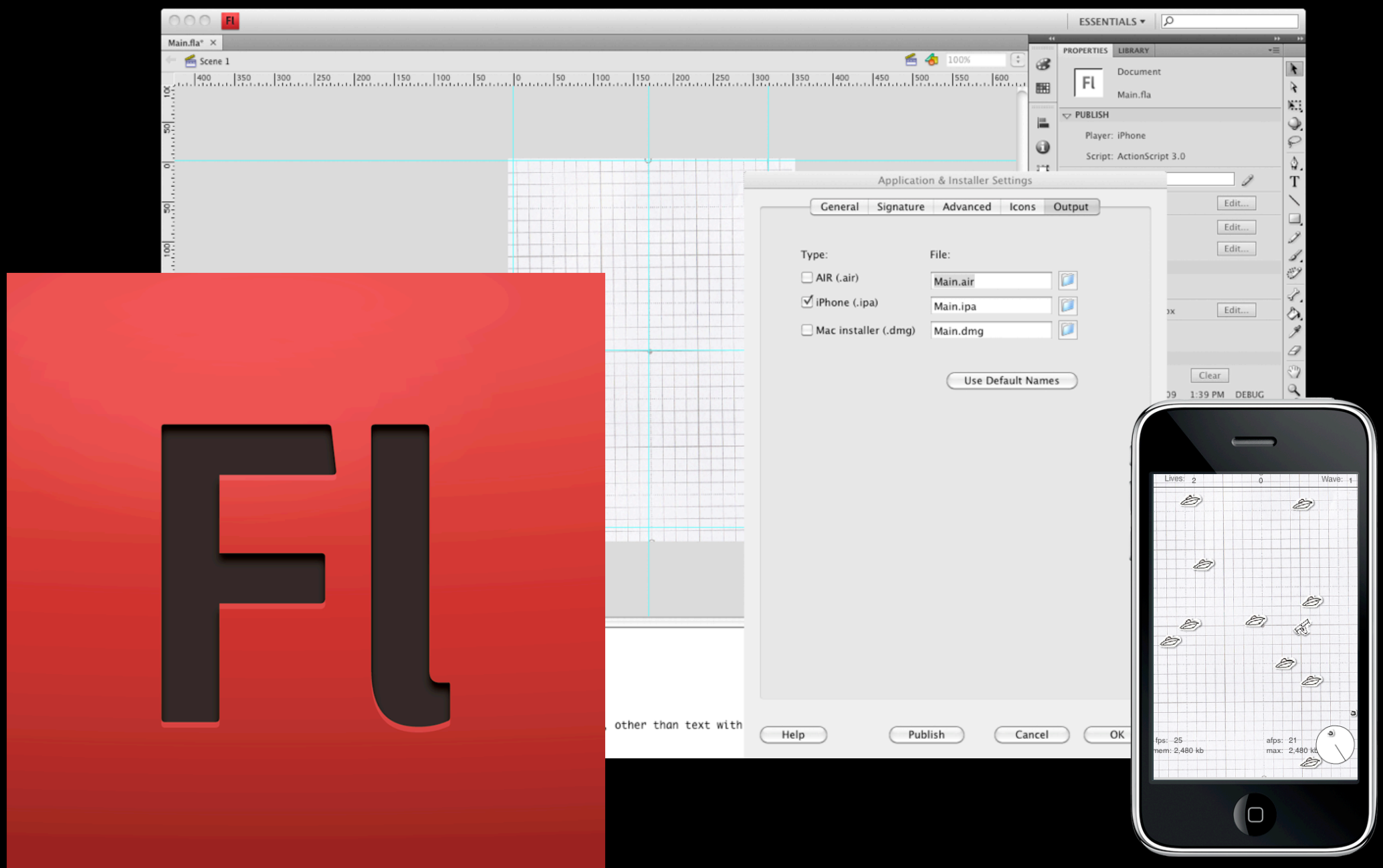
Search iPhone Reference Library

## Featured Content



### Watch Apple experts discuss getting started videos

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



# Rendering

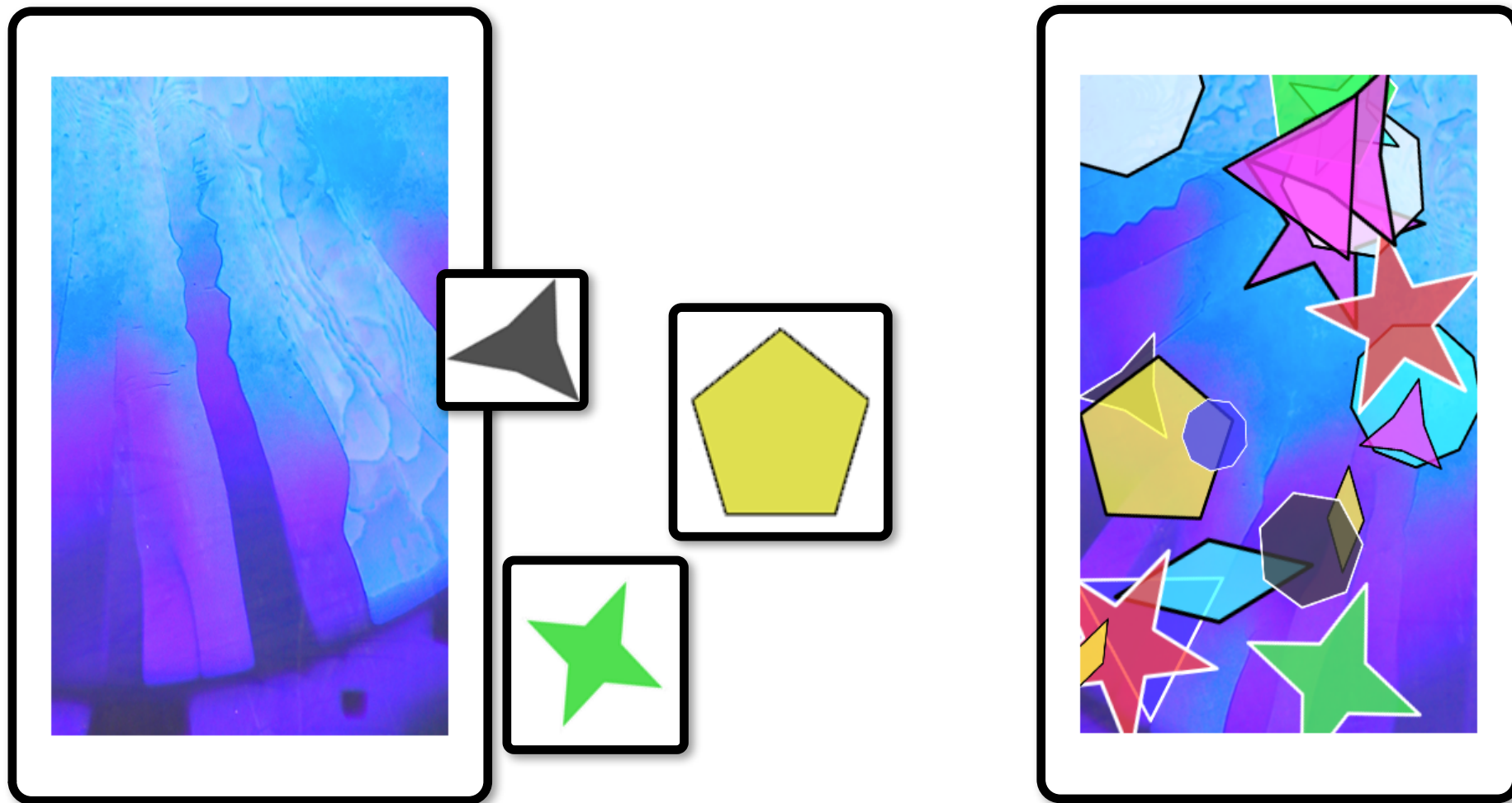
- Current Apps in iTunes store use Software Rendering
- Adding Support for GPU Composition

# Software Rendering





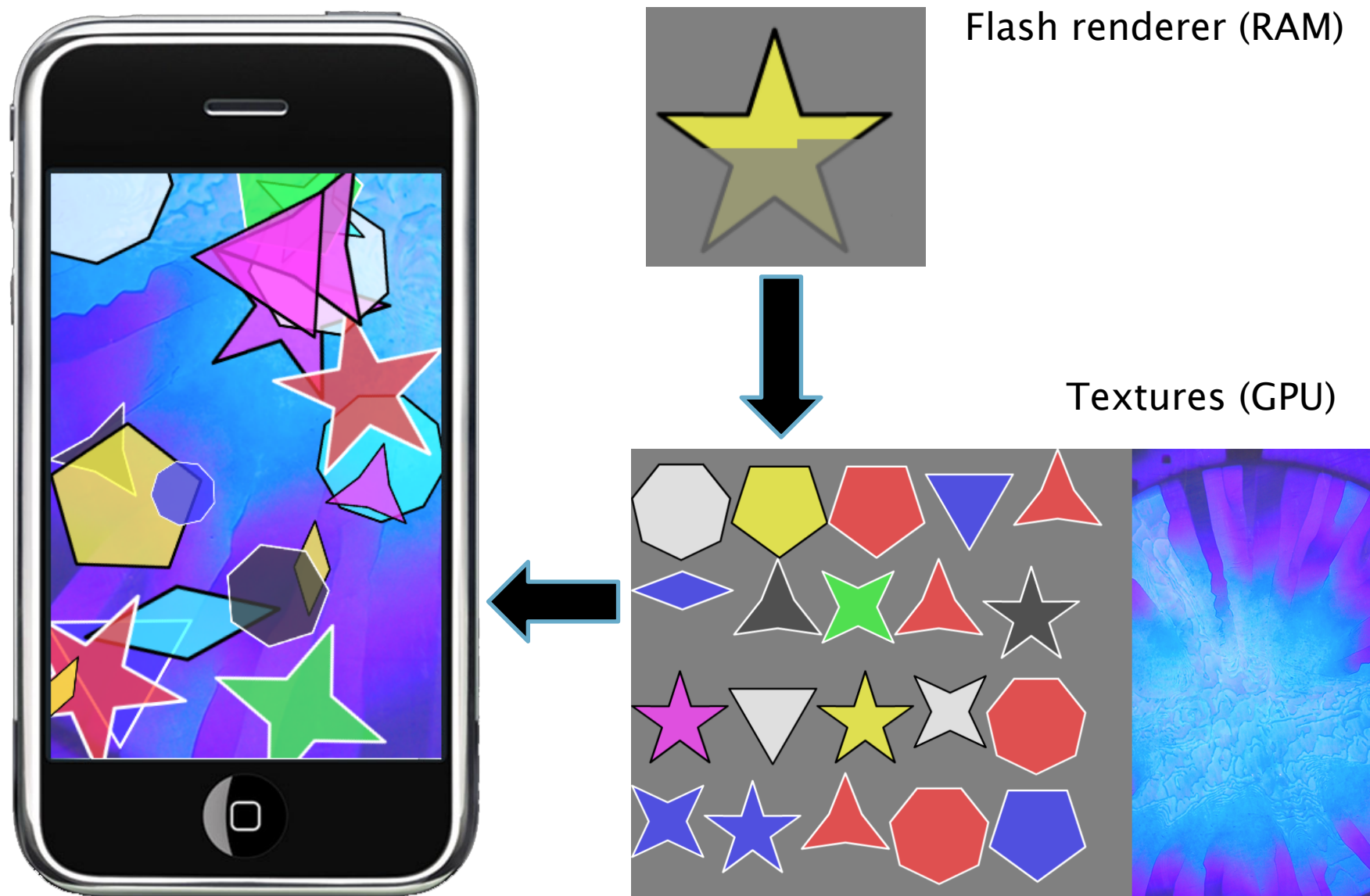
# GPU Compositing



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`
- `cacheAsSurface:Boolean`
- Using 2.5D Apis
  - Such as setting `z` property



What can you do to be ready?



!=



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 Internactions and most importantly performance.

# Screen Size



Fullscreen	320x480
With Status Bar	320x460 480x300



# Fonts

American Typewriter	Hello World
American Typewriter Condensed	Hello World
Arial	Hello World
Arial Rounded MT Bold	<b>Hello World</b>
Courier New	Hello World
Georgia	Hello World
Helvetica	Hello World
Marker Felt	<b>Hello World</b>
Times New Roman	Hello World
Trebuchet MS	Hello World
Verdana	Hello World
Zapfino	<i>Hello World</i>

- Helvetica is default
- Use device fonts for input fields

# Primary input is via Finger



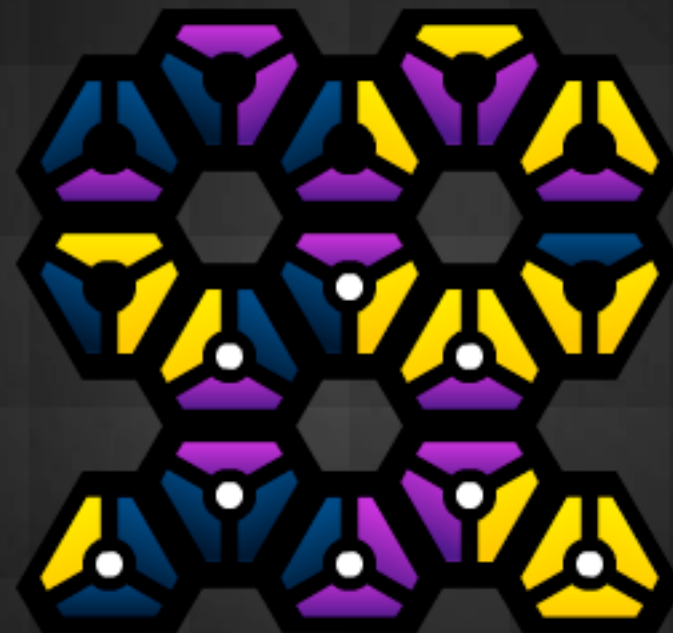
- Less accurate than mouse
  - Create larger hit area
  - No modifier keys
- 
- Remember Finger obscures screen



# CHROMACIRCUIT

TIME 00:41

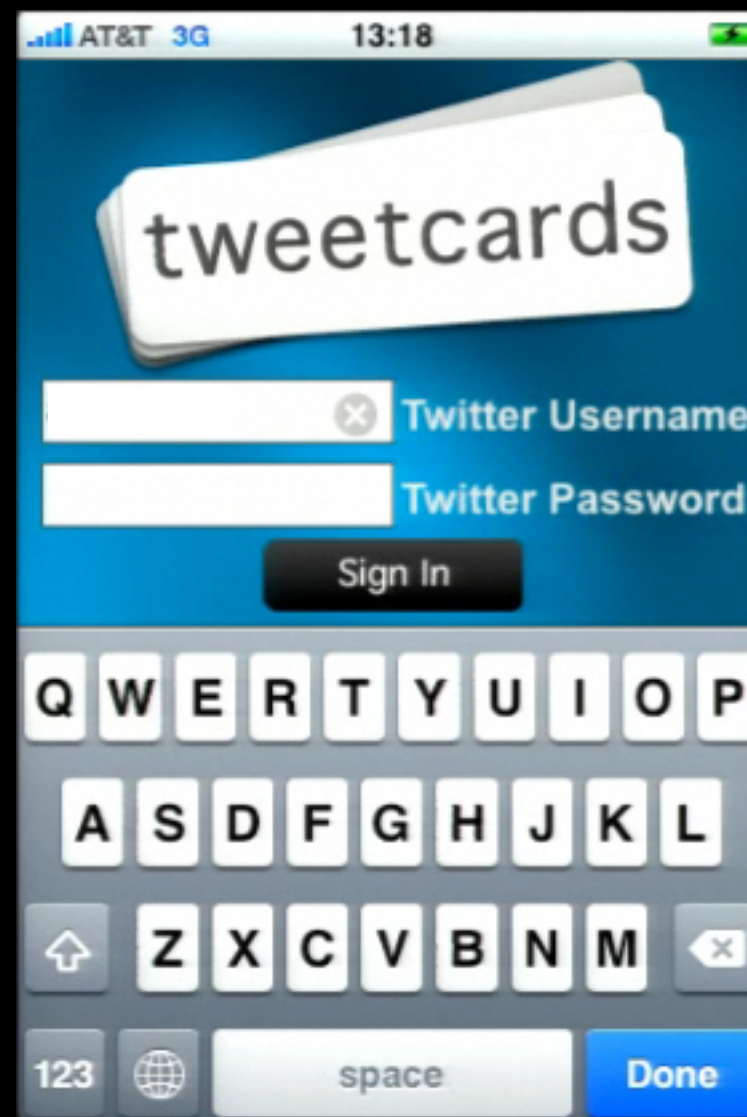
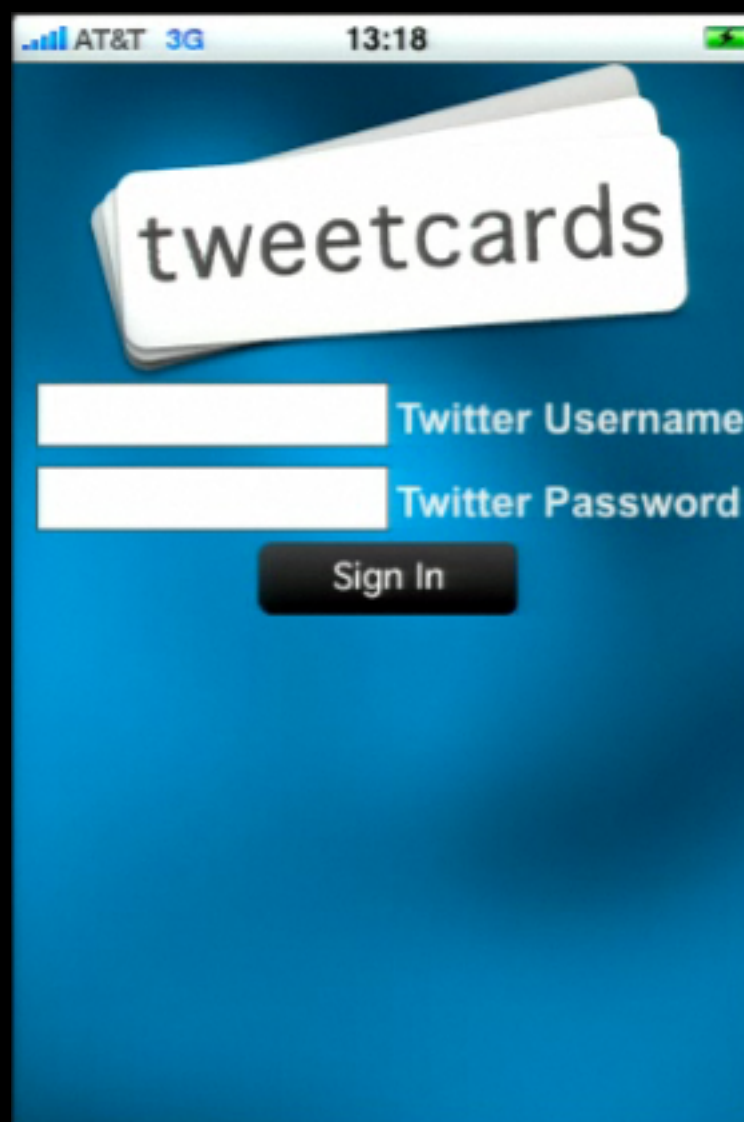
14 MOVES



LEVEL 4



# Text Input

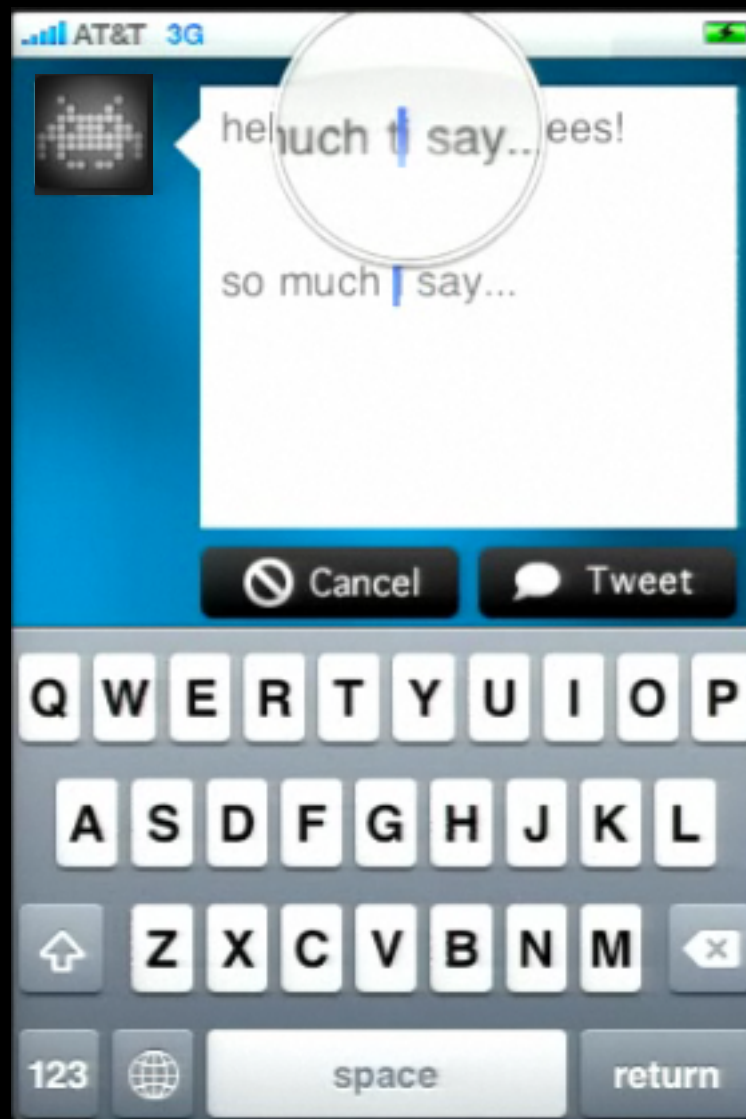


Remember iPhone has a software based keyboard. Make sure to leave room for it to open, and not cover text inputs.

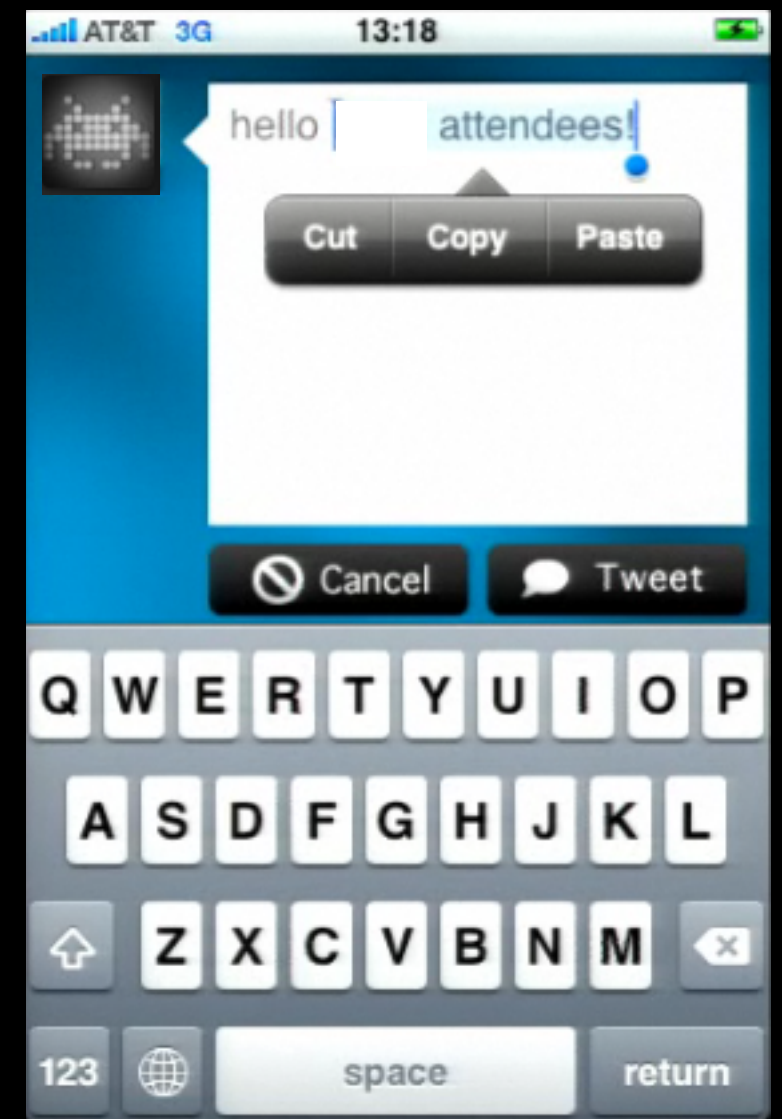
undo



Magnifying glass



cut/copy/paste



Native text input controls are used, so you have access to everything you would expect.

# Designing Flash Applications for the iPhone

Arno Gourdol

<http://bit.ly/arnoiphone>

# Apple Mobile Human Interface Guidelines

<http://bit.ly/mobilehig>

# Code Optimizations

- Most will improve performance / memory usage on desktop
- Some AOT specific



# 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	



# Reuse Instances

- 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);
    }
}
```

# Clean Up

- Clean up unneeded event handlers
- Remove unneeded Timer events and ENTER\_FRAME handlers

# Event Dispatching

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

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

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

# Mouse Events

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

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

private function onEnterFrame(e:Event):void
{
    if(mouseX < 0 || mouseY < 0)
    {
        //do something
    }
}
```

Can listen for MOUSE\_UP  
MOUSE\_DOWN events to toggle



# Function Calls (AOT)

- Arguments are passed on the stack
  - Reduce number of arguments
- AS3 Functions marked as “final” can be inlined
- Deep recursion can overflow stack
  - Out of memory will terminate app

Adobe is also working on a number of ActionScript 3 performance improvements for both JIT and AOT compiled content.

More info in the future.

# Designing and Developing for the Multiscreen web

## Thibault Imbert

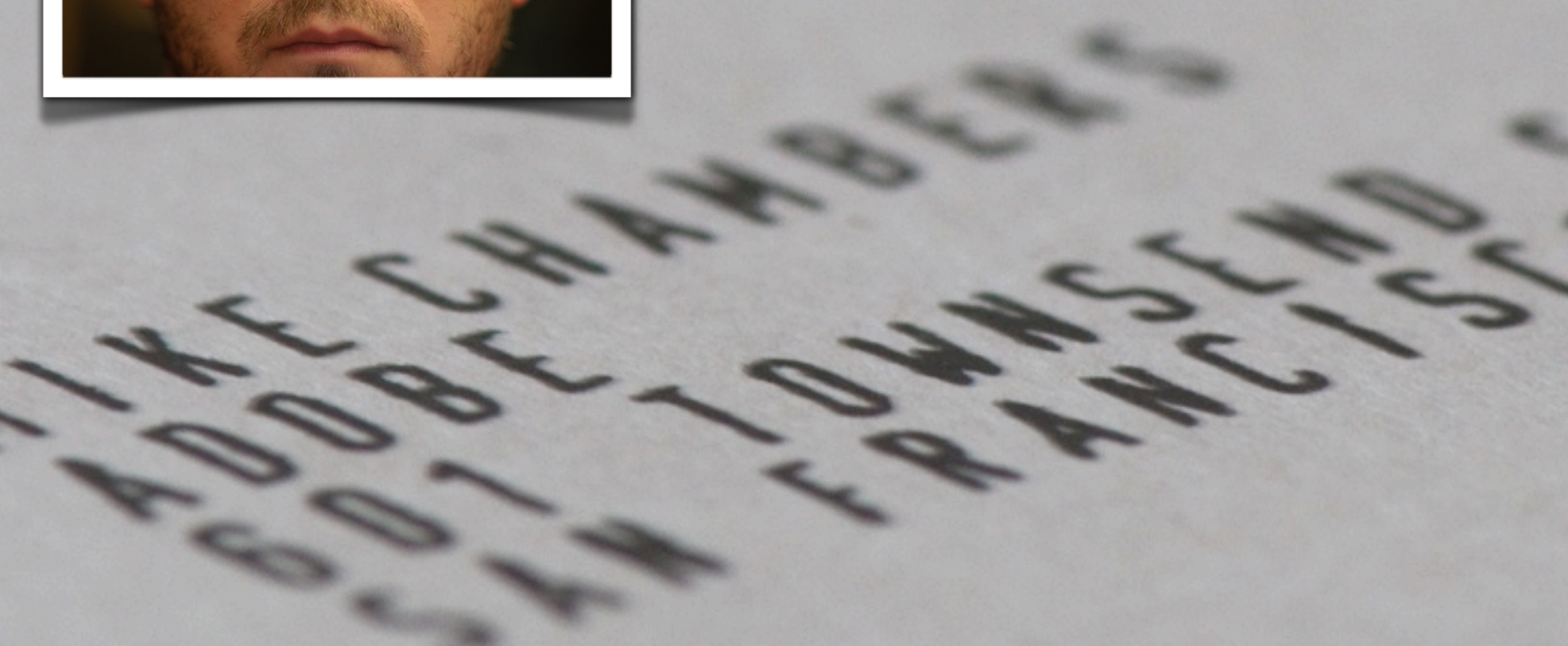
# Mike Chambers



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

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

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



# Baby Crying

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

