HXOBJC

Băluță Cristian https://github.com/ralcr

Things to cover

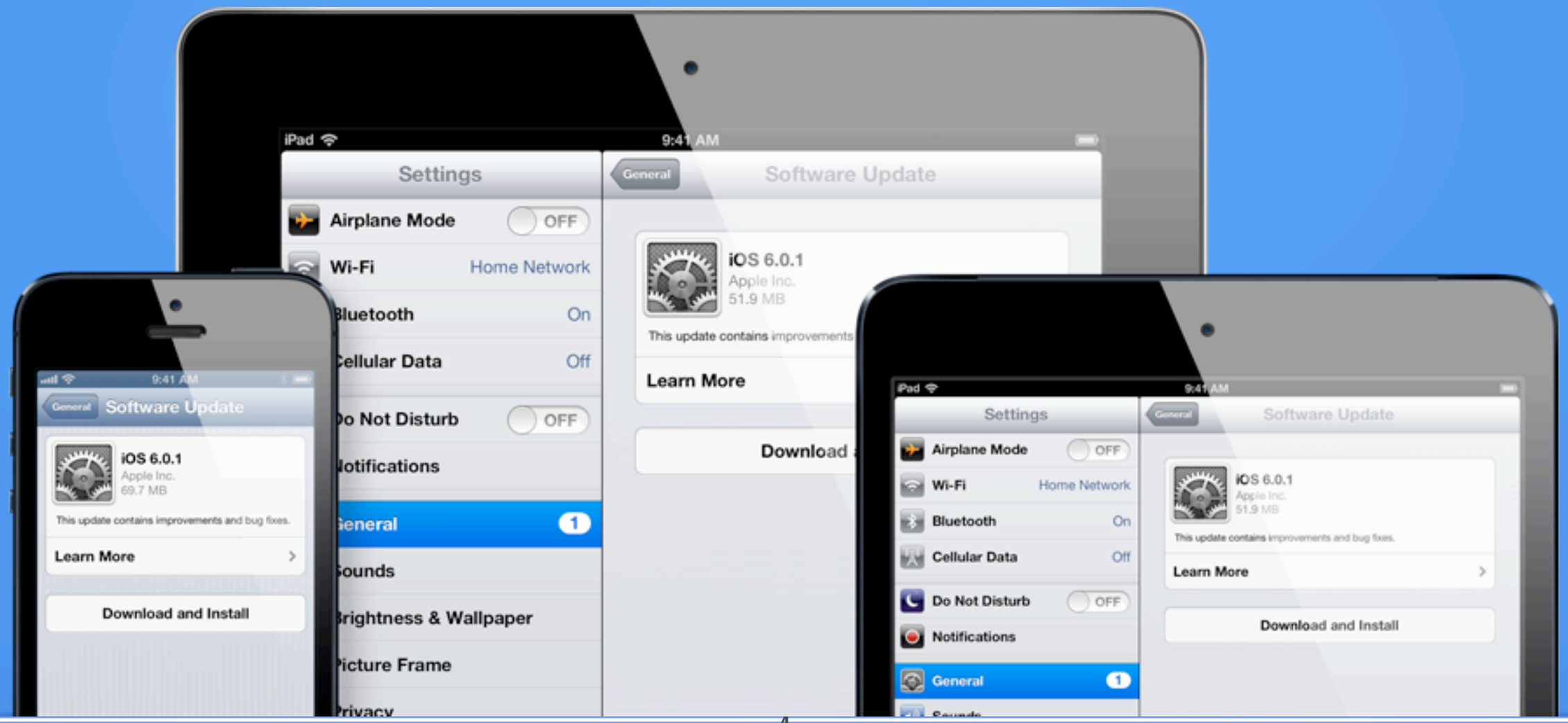
- Architecture
- Implementation
- Cocoa vs Flash
- Future

Why?

- Fun
- No viable alternative
- Speed
- Passion
- Knowledge

iOS apps

can be built with: Xcode, Haxe NME, one of the hundreds of HTML5 frameworks, Flash IDE, Unity



Beautiful syntax

```
- (int)changeColorToRed:(float)r green:
(float)g blue:(float)b;

[myColor changeColorToRed:5.0 green:2.0
blue:6.0];
```

```
function changeColorToRed (red:Float,
green:FLoat, blue:FLoat) :Int;

myColor.changeColorToRed (5.0, 2.0, 6.0);
```

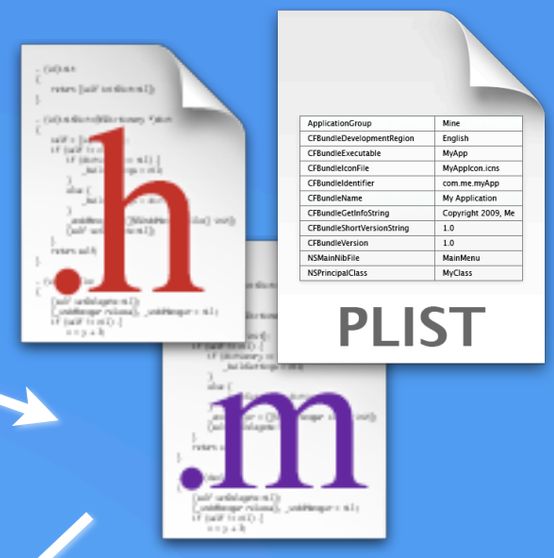
Architecture

hxcocoa

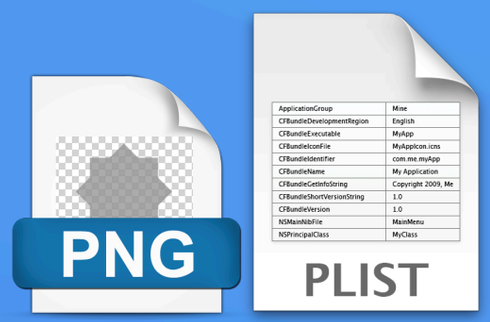
.hx

haxe libs

Hxobjc



SupportingFiles



Compiler flags

- -objc-platform : iphone, ipad, ios, universal
- -objc-version : 4.0 - 6.1
- -objc-bundle-version : 1.0
- -objc-identifier
- -objc-owner
- -objc-bundle-name
- -objc-supporting-files

Compiler flags

- -objc-lib
- -objc-framework
- -objc-linker-flag : ObjC
- -ios-orientation : UIInterfaceOrientationPortrait

Implementation

Genobjc.ml

- 3400 lines of code
- stole code from genas3 and gencpp
- failed to split in genobjc + genxcode
- camelCase
- dozens of repetitive matches of type ‘t’

The magic behind

- String -> NSMutableString category
- Array -> NSMutableArray category
- Anonymous Object -> NSMutableDictionary
- Date -> NSDate category
- primitive -> primitive
- static -> static + getter + setter methods
- dynamic func -> property + method + block
- new -> init. other inits are disabled

The magic behind

- array access -> hx_objectAtIndex + casting
- optional arguments -> if then
- class -> interface + implementation
- interface -> protocol
- enum -> enum

Special classes

- new SEL(method) -> @selector(method:arg:)
- new CGRect -> CGGeometry.CGRectMake()
- Concurrency -> static C methods
- Iterator?

Null

- replaced by nil or [NSNull null] in arrays
- objective-c is a dynamic language
- when it encounters a nil is not doing anything
- your app will not crash
- but you might wonder why is not working

Dynamic

- it’s replaced by ‘id’
- ‘id’ is used as a return of a constructor or can hold any object
- but can’t hold primitive values
- ‘id’ does not require the *

Metadata

- @:include -> #import <someheader.h>
- @:import -> #import “someheader.h”
- @:sel(“methodName:arg1:arg2:”)
- @:framework
- @:weak
- @:c
- @:category
- @:getterBody

Not working

- Reflect, Type, Ereg, Xml, Resources
- primitives as objects
- sys, unit tests
- Dynamic and untyped calls
- complex enums
- Int32, Int64, Utf8, io
- string + primitive concatenation
- import one file into another and vice versa.

Memory

it's done with ARC (automatic ref. counting)

Memory

if you follow good programming principles you'll not have leaks, but if you have try the @:weak metadata

Cocoa, CocoaTouch

Do not reinvent the wheel

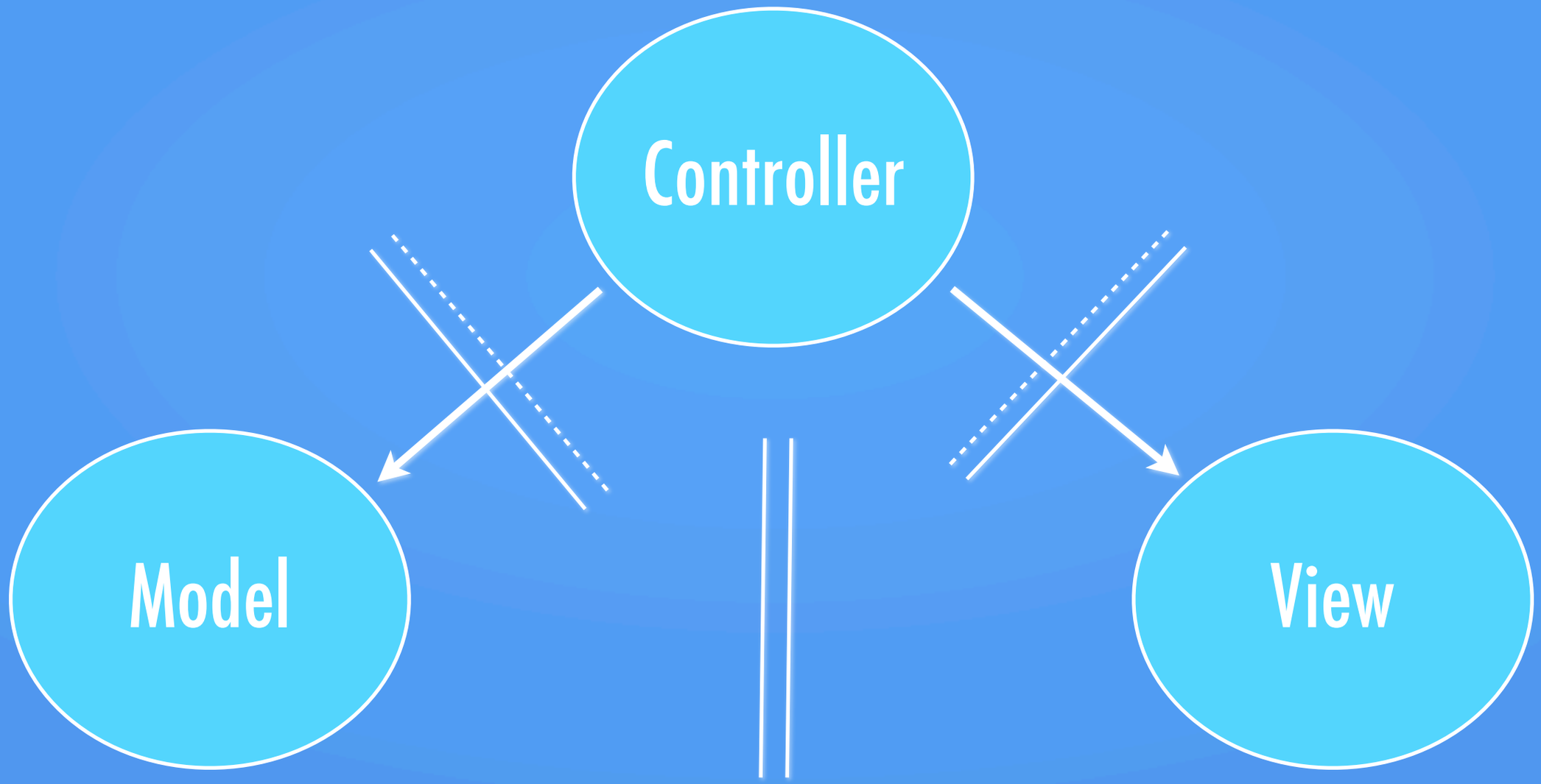
Hxcocoa

- haxelib
- 3 main packages: objc, ios, osx
- objc: foundation, graphics, network, location, store, audio video
- ios: ui, map, gl, assets, addressbook, iAd, mediaplayer, social, telephony
- osx: addkit, webkit
- SupportingFiles : ios launch images and icons

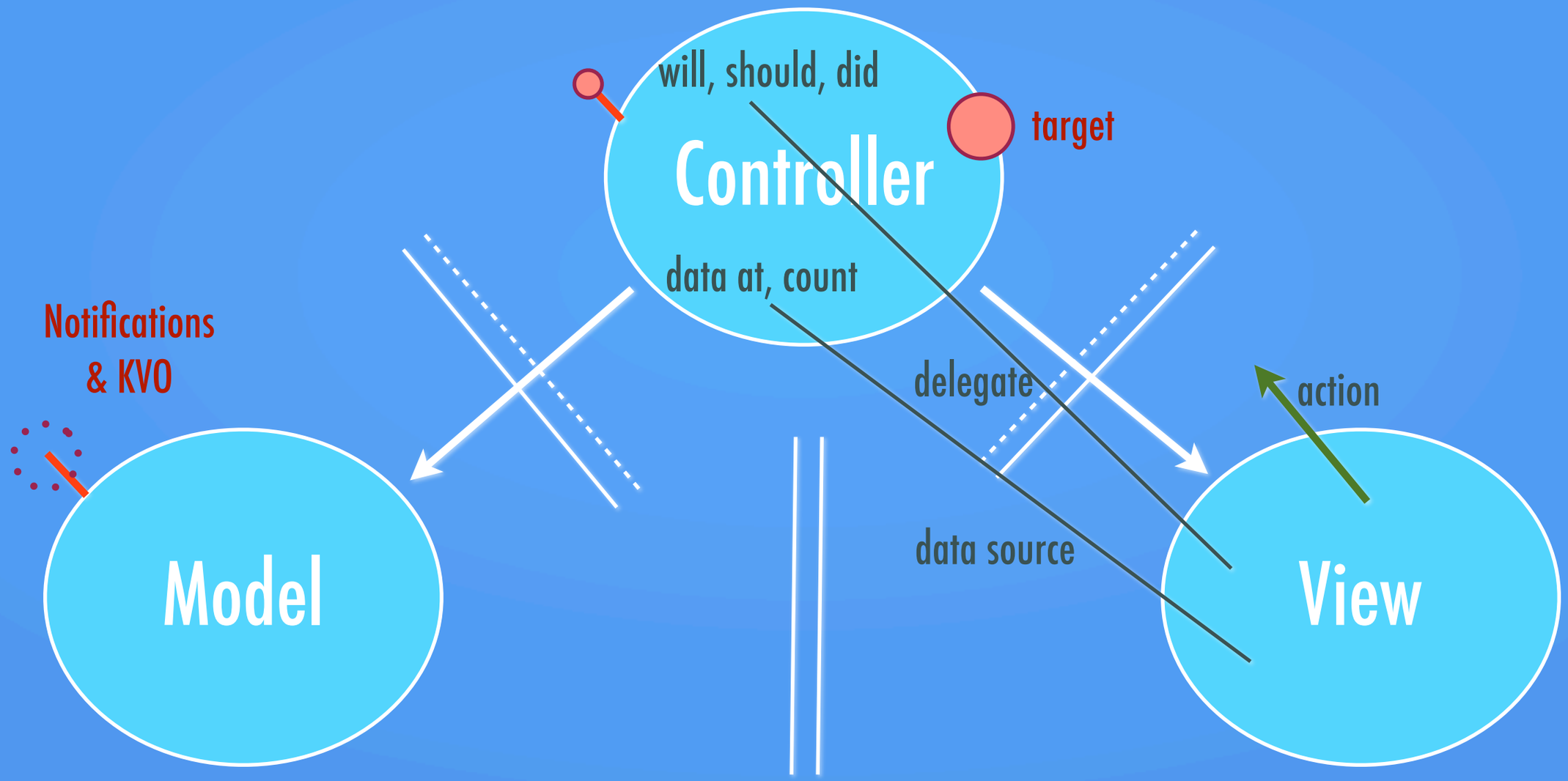
Lib.hx

- each package has it’s own Lib.hx
- generic: log, printf, println, location, getURL
- ios: attach, enumerateLibraryGroups, fetchCameraItemsInGroup, isIpad, isIpod, alert

MVC



MVC


will, should, did
target
Controller
data at, count
Notifications
& KVO
delegate
action
data source
Model
View

ViewController lifecycle

```
public function new ()

override public function viewDidLoad ()
override public function viewWillAppear ()
override public function viewDidAppear ()

override public function didReceiveMemoryWarning ()

override public function viewDidDisappear ()
override public function dealloc ()
```

Flash vs Cocoa

- Sprite -> UIView
- x, y, width, height -> frame.origin, frame.size
- addChild -> addSubview
- clipsToBounds, backgroundColor

```
var sprite = new Sprite();
sprite.x = 0;
sprite.y = 0;
```

```
var sprite = new UIView();
var frame:CGRect = sprite.frame
frame.origin.x = 0;
frame.origin.y = 0;
sprite.frame = frame;
```

Flash vs Cocoa

- EventDispatcher -> Delegate and UINotificationCentre

Flash vs Cocoa

TouchEvent -> UITouch

```
this.addEventListener (TouchEvent.TOUCH_BEGIN, tapHandler);
function touchHandler(e:TouchEvent) {
      var x = e.localX;
      var y = e.localY;
}
```

```
override public function touchesBegan (touches:NSSet,
withEvent:UIEvent) :Void {
       var touches :Array = withEvent.touchesForView(this);
       var aTouch :UITouch = touchesForView.anyObject();
       var pos :CGPoint = aTouch.locationInView(this);
}
```

Flash vs Cocoa

MouseClick -> target-action pattern

```
but.addEventListener (MouseEvent.MOUSE_CLICK, touch);
```

```
but.addTarget (this, new SEL(touch),
UIControlEventTouchUpInside);
```

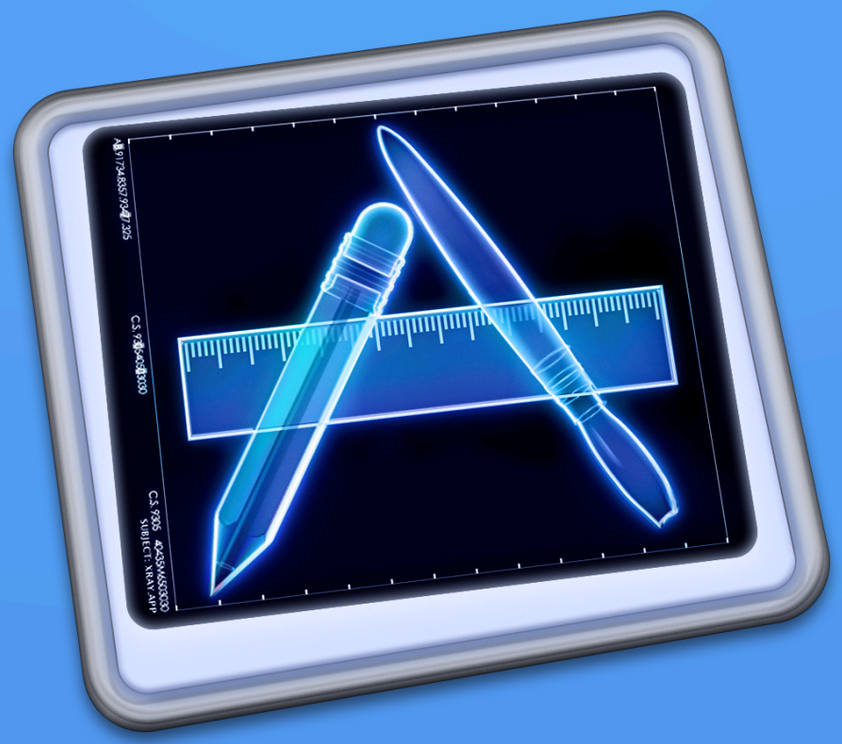
CoreAnimation

- UIView animation
- hidden, frame, alpha, transform
- the properties are set immediately but the appearance is animated

```
UIView.animateWithDuration (duration:Float, delay:Float,
options:UIViewAnimationOptions, animations:Void->Void,
completion:Bool->Void);
```

Debugging

- traces are converted to println with Haxe pos information
- native NSLog
- Instrumens for finding leaks and profiling



Benchmarks

- Compile time:
- Hxobjc - all utest takes 0.8 sec
- NME - seconds to minutes

- Size:
- Hxobjc - almost as the .hx files
- NME - beyond imagination

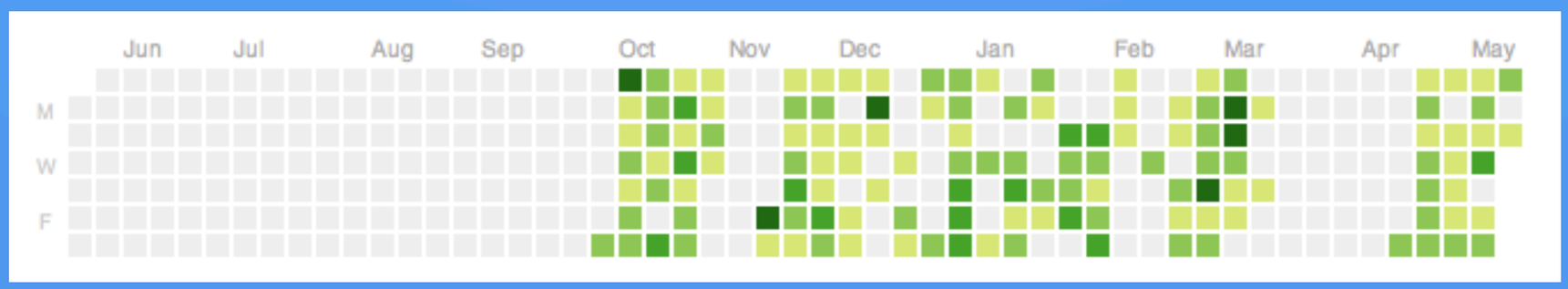
Benchmarks

- Runtime
- 1mil mutable strings takes 2.5 sec comparing to 1mil strings 0.012 sec
- Adding this strings to an array takes 5sec for the mutable string and 0.46 for the simple strings

Future

Future

- Implement all the standard library
- objc and ios externs
- osx externs
- bypass Xcode manual compilation
- sdk.ralcr hxobjc support

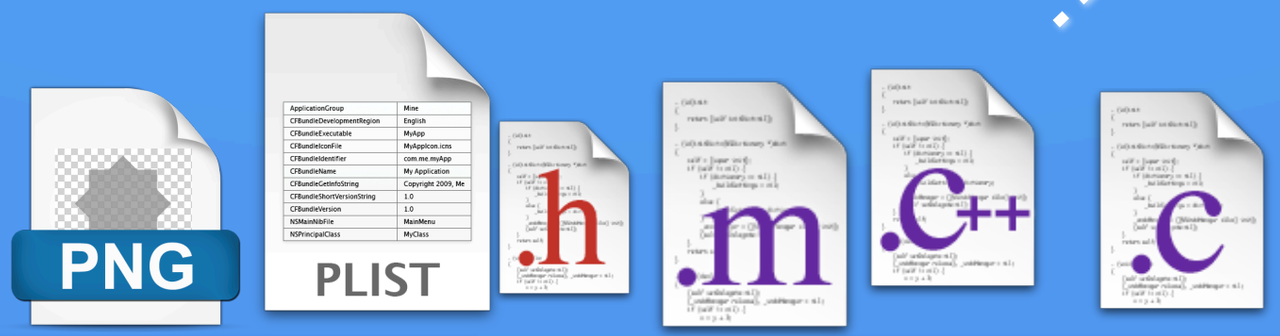


.hx

haxe libs

Hxobjc

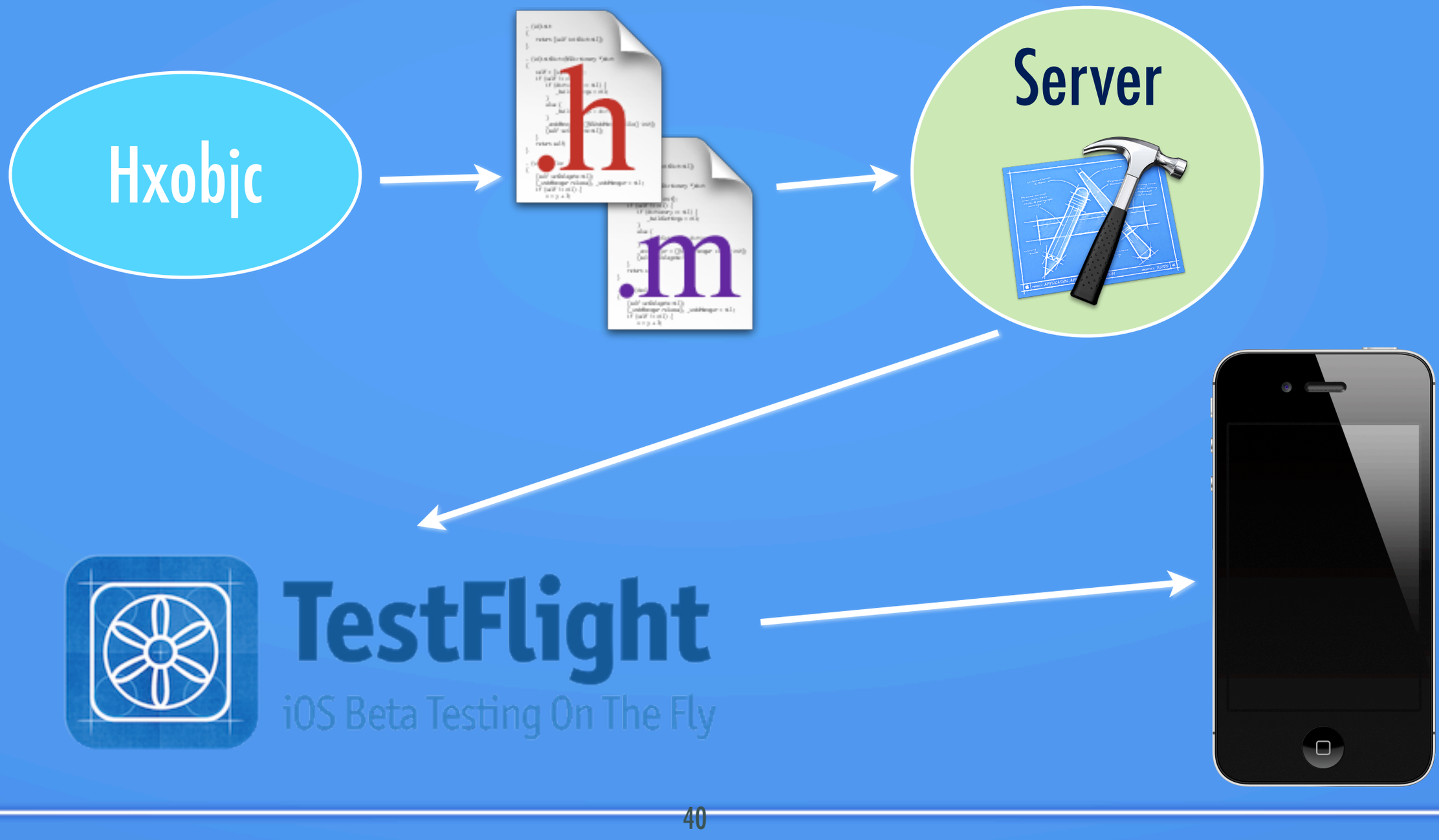
SupportingFiles



Distant future

2025

Compile serverside would be nice



Questions?