A Tale of Tweaking

Filippo Bigarella

April 13, 2014

About Me

- 19 years old student based in Italy
  - Sorry for my English!
  - Studying Computer Science at the University of Trento

About Me

- Joined the Jailbreak community as a developer in late 2010
- First jailbroken in November 2009: didn't even know what a jailbreak was back then
- blackra1n!
- Winterboard
- !?

The Beginning

- First tweak: StartDial

The Beginning

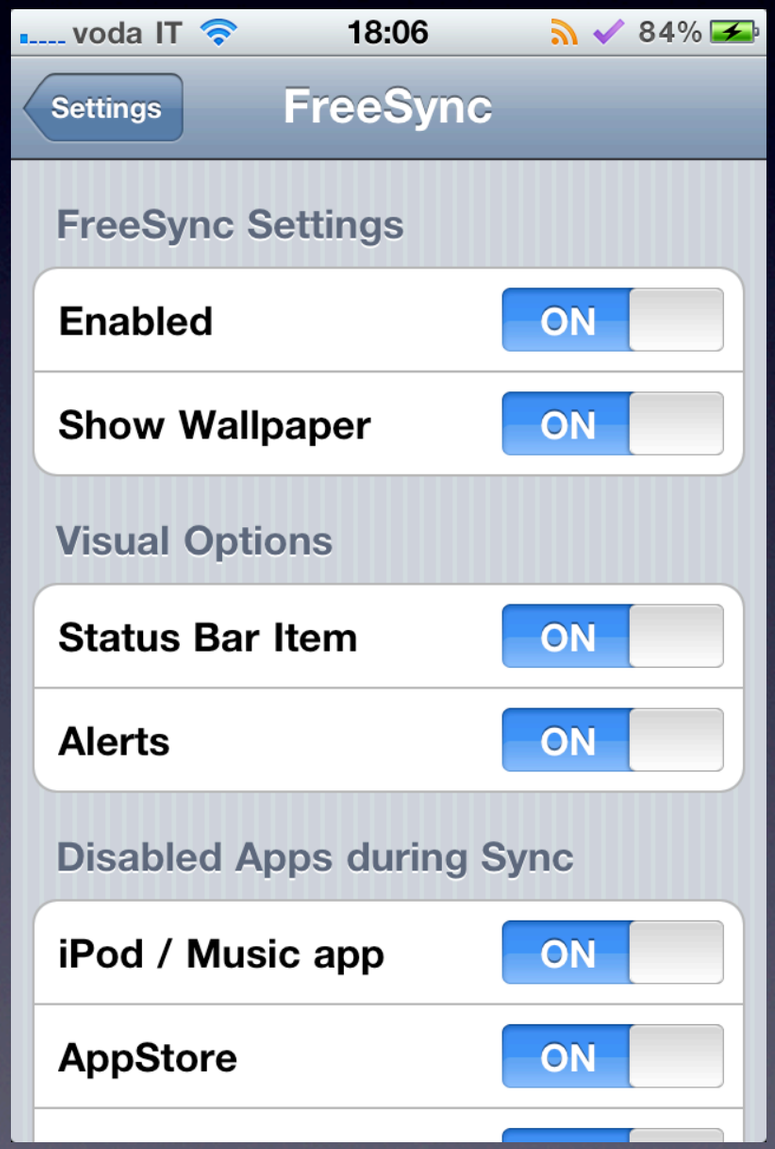
- First tweak: StartDial
  - “random” project
    - Very simple tweak
  - Didn’t know much about Objective-C and the runtime
    - Many things to learn

Towards the Cydia Store

- Making tweaks was amazing!
- Mess around with iOS
- Understand how things work
- SpringBoard fascinated me
- Started digging into it even more!

My first steps in the Cydia Store

- FreeSync
  - Worked alongside @qwertyoruiop
  - Use your device while syncing it!



My first steps in the Cydia Store

- FreeSync
- Wrote along with @qwertyoruiop
- Use your device while syncing it!
- Visual effects during sync

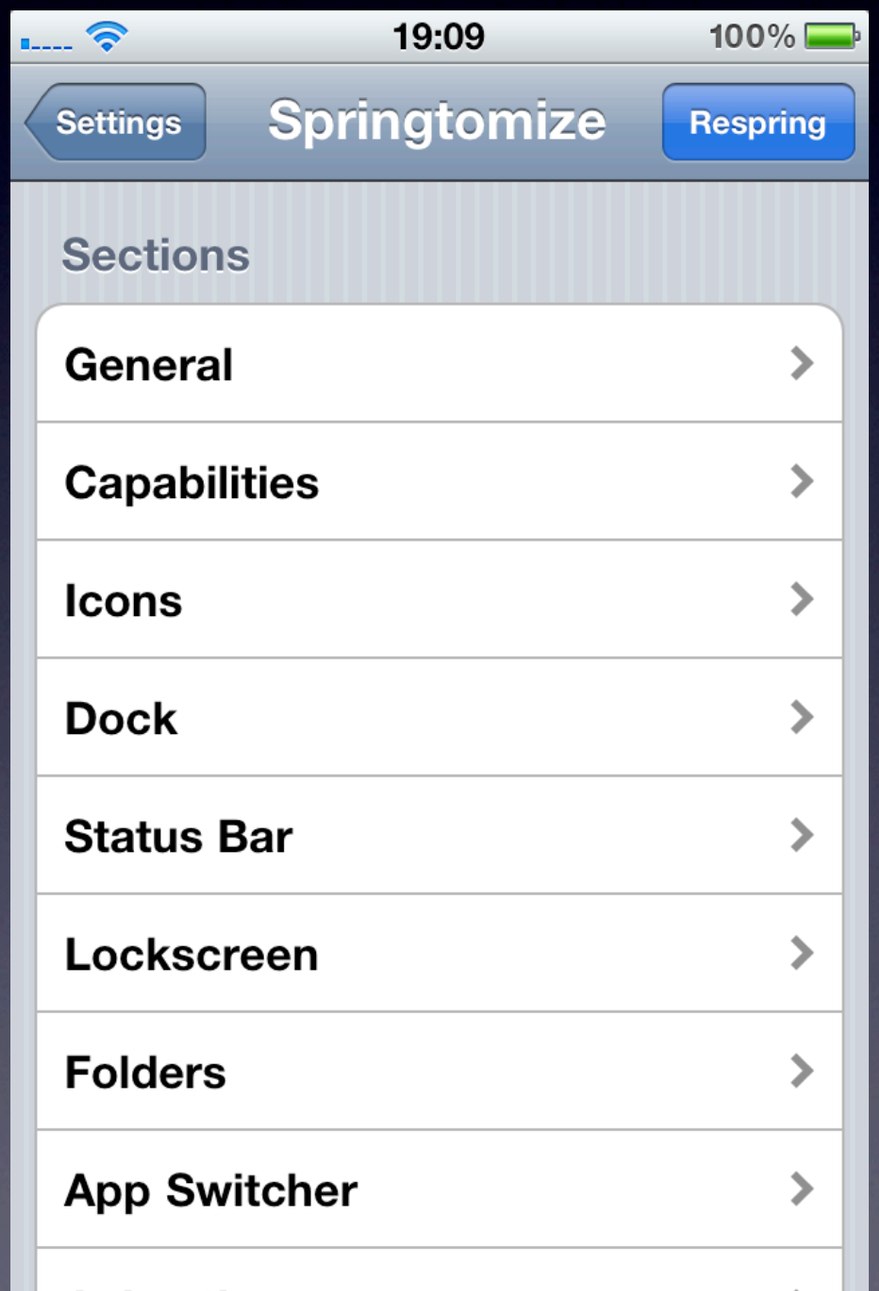
My first steps in the Cydia Store

- FreeSync
  - Wrote along with @qwertyoruiop
- Use your device while syncing it!
- Visual effects during sync
- Obsolete with iOS 5 :(

The birth of Springtomize

- As I said, SpringBoard fascinated me
- I began trying to make as many changes as I could
- Tweak ALL the things
  - People liked the idea
- Springtomize!

The birth of Springtomize



The birth of Springtomize

- Released on March 12th, 2011 (iOS 4)

The birth of Springtomize

- Trying to concentrate as many features as possible into a single, easy-to-use interface

... and its flaws

- "Monolithic" project
- All the code contained in a single file
- Difficult to maintain
- No scope separation

... and its flaws

- Expensive Settings (First version only)
- Stored in a plist

... and its flaws

- Expensive Settings (First version only)
- Stored in a plist (that's fine)

... and its flaws

- Expensive Settings (First version only)
  - Stored in a plist (that's fine)
  - ... NEVER stored in memory!
    - Read the whole plist *every* time

... and its flaws

- Expensive Settings (First version only)

```
%hook SBIconController

-(void)scrollViewDidScroll:(id)scrollView
{
    reloadprefs;
    ifd(@"Enabled", YES)
    {
        ifd(@"STDisableSpot", NO)
        {
            SBSearchView *searchView_ = MSHookIvar<SBSearchView *>(self, "_searchView");
            searchView_.hidden = YES;
        }
    }
    %orig;
}

%end
```

... and its flaws

- Expensive Settings (First version only)

```
#define reloadprefs NSDictionary* dict = [NSDictionary dictionaryWithContentsOfFile:settingsFile]

%hook SBIconController

-(void)scrollViewDidScroll:(id)scrollView
{
    reloadprefs;
    ifd(@"Enabled", YES)
    {
        ifd(@"STDisableSpot", NO)
        {
            SBSearchView *searchVi                    earchView *>(self, "_searchView");
            searchView_.hidden = YE
        }
    }
    %orig;
}

%end

#define ifd(x, y) if ([dict objectForKey:x] ? [[dict objectForKey:x] boolValue] : y)
```

... and its flaws

- Expensive Settings (First version only)

```
%hook SBIconController

-(void)scrollViewDidScroll:(id)scrollView
{
    NSDictionary* dict = [NSDictionary dictionaryWithContentsOfFile:settingsFile];
    if ([dict objectForKey:@"Enabled"] ? [[dict objectForKey:@"Enabled"] boolValue] : YES)
    {
        if ([dict objectForKey:@"STDisableSpot"] ? [[dict objectForKey:@"STDisableSpot"] boolValue] : YES)
        {
            SBSearchView *searchView_ = MSHookIvar<SBSearchView *>(self, "_searchView");
            searchView_.hidden = YES;
        }
    }
    %orig;
}

%end
```

... and its flaws

- Expensive Settings (First version only)
  - Fixed with an update
- Plist read only once, everything stored in memory with static variables

```
static BOOL STFoldersInDock=NO;
static BOOL STBadges=NO;
static BOOL STJitter=NO;
static BOOL STDisableSpot=NO;
static BOOL STFolderBadges=NO;
static BOOL STFolderIconName=NO;
```

... and its flaws

- Expensive Settings (First version only)
  - Fixed with an update
  - Plist read only once, everything stored in memory with static variables

```
STFoldersInDock = [[dict objectForKey:@"STFoldersInDock"] boolValue] ?: NO;
STBadges = [[dict objectForKey:@"STBadges"] boolValue] ?: NO;
STJitter = [[dict objectForKey:@"STJitter"] boolValue] ?: NO;
STDisableSpot = [[dict objectForKey:@"STDisableSpot"] boolValue] ?: NO;
STFolderBadges = [[dict objectForKey:@"STFolderBadges"] boolValue] ?: NO;
STFolderIconName = [[dict objectForKey:@"STFolderIconName"] boolValue] ?: NO;
```

... and its flaws

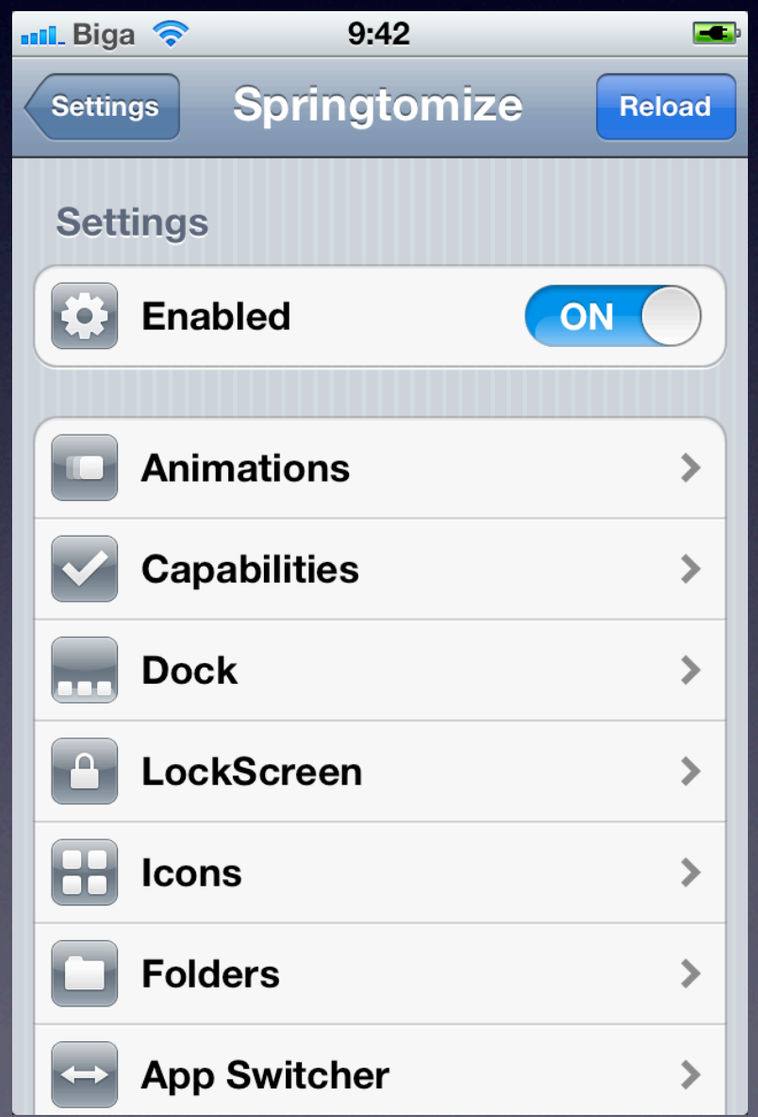
- Settings bundle implemented using only property lists
- Not very flexible

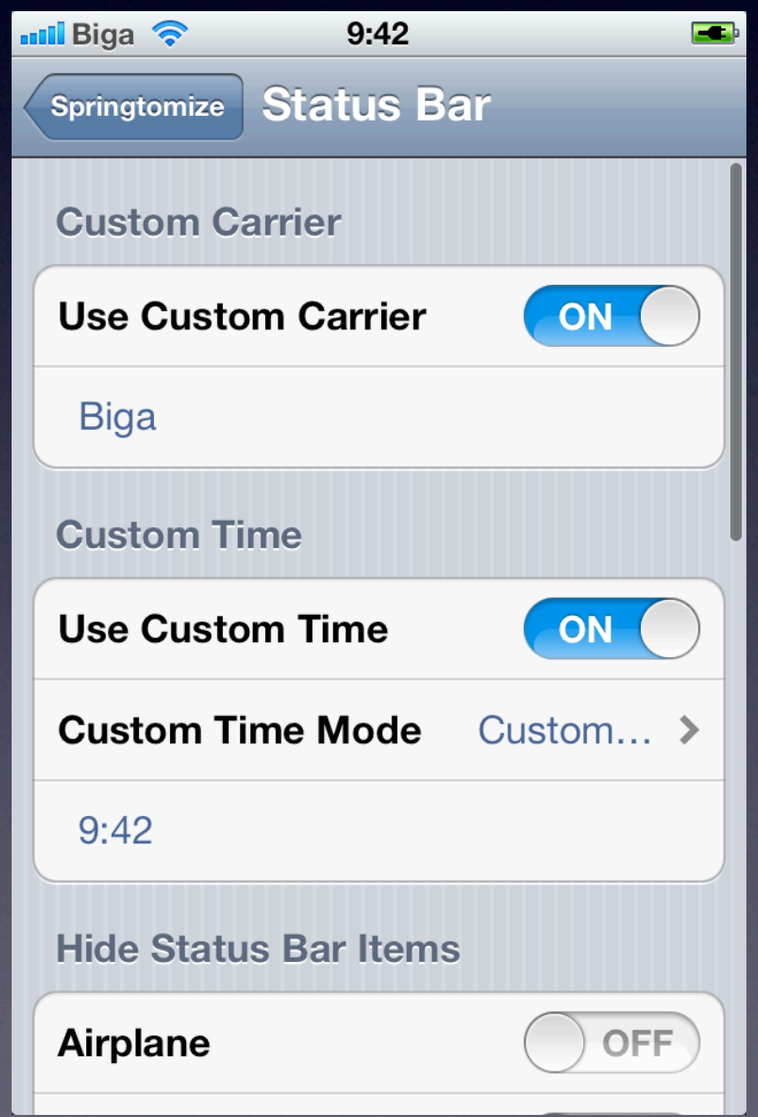
Springtomize 2

- Released on December 1st, 2011 (iOS 5)

Springtomize 2

- Deeper customization





Springtomize 2

- Re-engineered project
  - ~20,000 lines of code split across 40 files
  - 3 components:
    - Tweak
    - Settings bundle (+ application)
    - Daemon

Springtomize 2

- Improved, pseudo-dynamic settings handling

Springtomize 2

- 19 (free) updates in 2 years
- Bug fixes
- Tons of new features
- iOS 6.x support

Flaws²

- Messy structure
  - Some files needed to be included in others to be compiled (!?)
    - Long compilation time
- Still no separation of scope between code units

Flaws^2

- Unstructured Hooks
  - A lot of code needed to be replicated to support a minimal change
- Messy build environment
  - “hacked up” headers
  - Manual additions / fixups required

The Arrival of iOS 7

- Many changes to the interface
- SpringBoard included
- Broken private APIs
- Broken jailbreak-dev tools and general jailbreaks itself
- Steep upgrade path from Springtomize 2

Springtomize 3

- Project started in early October
- Built as a whole new project
- Fine-tuned build environment
  - Build verification
  - Automated CVS (git)

Springtomize 3: Getting Ready

- Need to do a lot of work inside SpringBoard
  - Get a "compilable" set of headers to speed up development
  - Dump + fix all the headers
  - Scripts to fix and test headers
  - No need to copy/paste class declarations into a messy header

Springtomize 3: The Project

- 4 different components working together
  - Springtomize library
  - Tweak
  - UI Library
  - Settings Bundle

Springtomize 3:
The Project

- 4 different components working together
  - Springtomize library
  - Tweak
  - UI Library
  - Settings Bundle

Springtomize 3: Settings

- Big number of settings to manage
- How to store them (memory / disk)?
- How to handle user modifications?

Springtomize 3: Settings (Storage)

- On disk:
  - Just use Property Lists
  - Easy to maintain
  - Fast
  - No overhead of using external libraries

Springtomize 3: Settings (Storage)

- In memory
  - Static variables make the code messy
  - Can’t properly manage multiple code units
  - No proper way to “group” them by context

Springtomize 3: Settings (Storage)

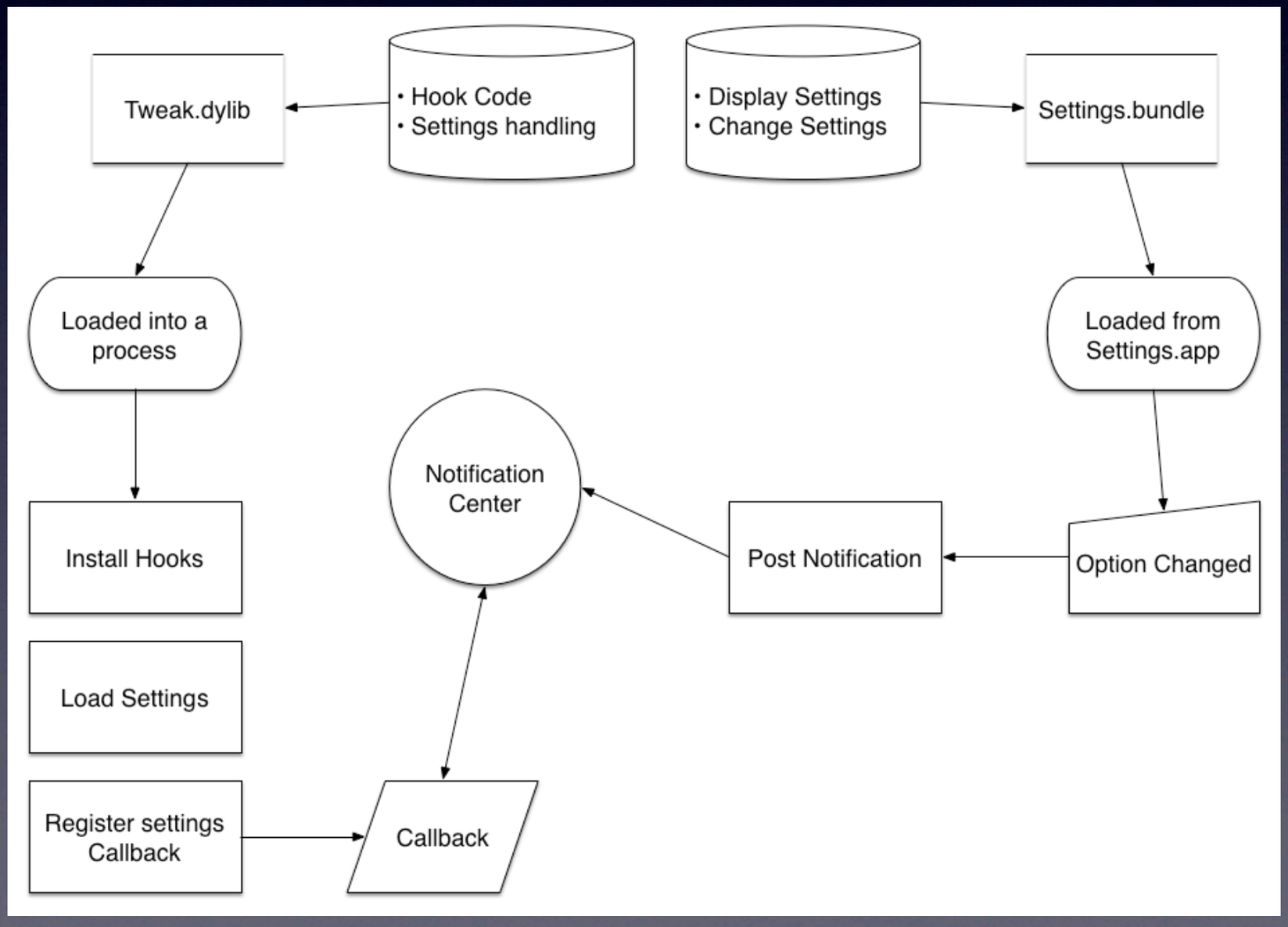
- In memory
  - Context-aware structures!
  - Easily group variables by context
  - Consistent ways to read / save / change options
  - => Consistent handling

Springtomize 3: Dynamic Settings

- The user must be able to easily modify options
- "Best effort" policy
  - Apply anything that's possible, ask for respring for deeper changes
- How to handle changes when you have a lot of options?

Springtomize 3: Dynamic Settings

- Classic "dynamic settings" approach

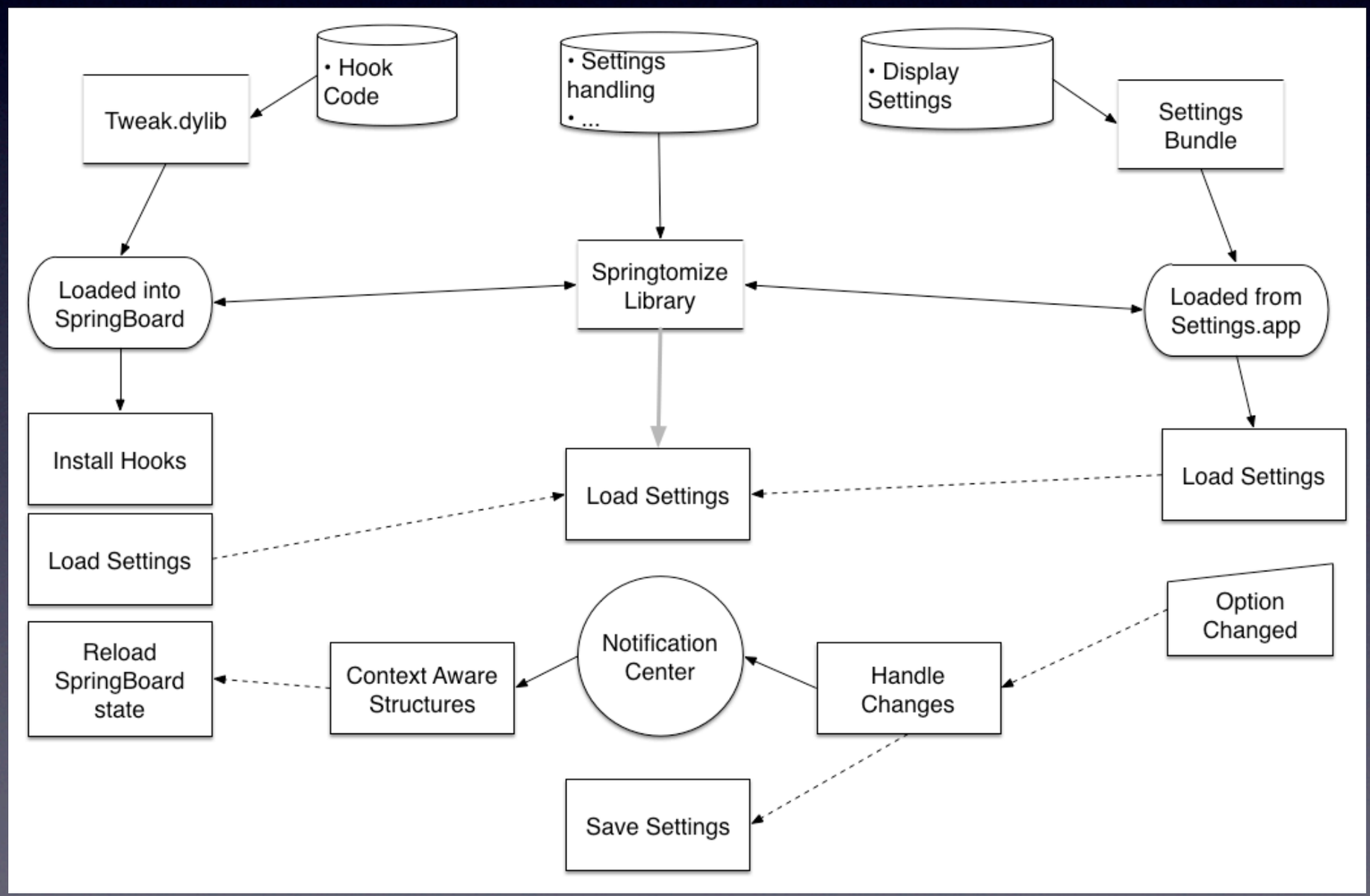


Springtomize 3: Dynamic Settings

- Classic "dynamic settings" approach
  - works for a limited number of settings
    - reload all settings, all the time OR
    - register multiple notifications
      - low flexibility

Springtomize 3: Dynamic Settings

- Springtomize 3 Approach



Springtomize 3: Dynamic Settings

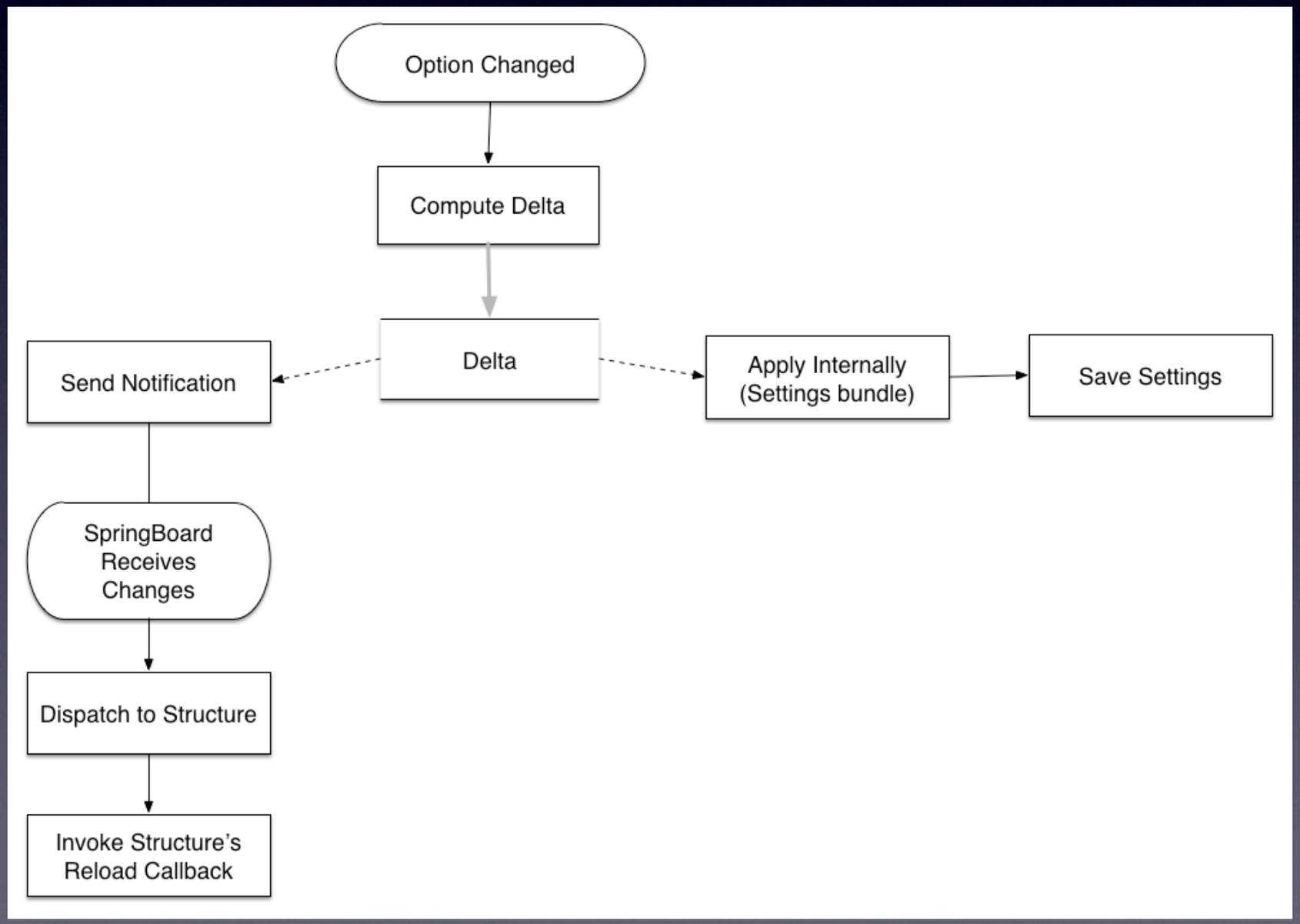
- Code to handle settings implemented in a common library
- Single message sent through the system
  - Automatically dispatched to the right structure
    - Reload only what's needed to apply changes

Springtomize 3: Dynamic Settings

- The notification contains the changes
  - “delta” of settings sent across the system
  - Update only what’s needed
  - No need to read anything from disk

Springtomize 3: Dynamic Settings

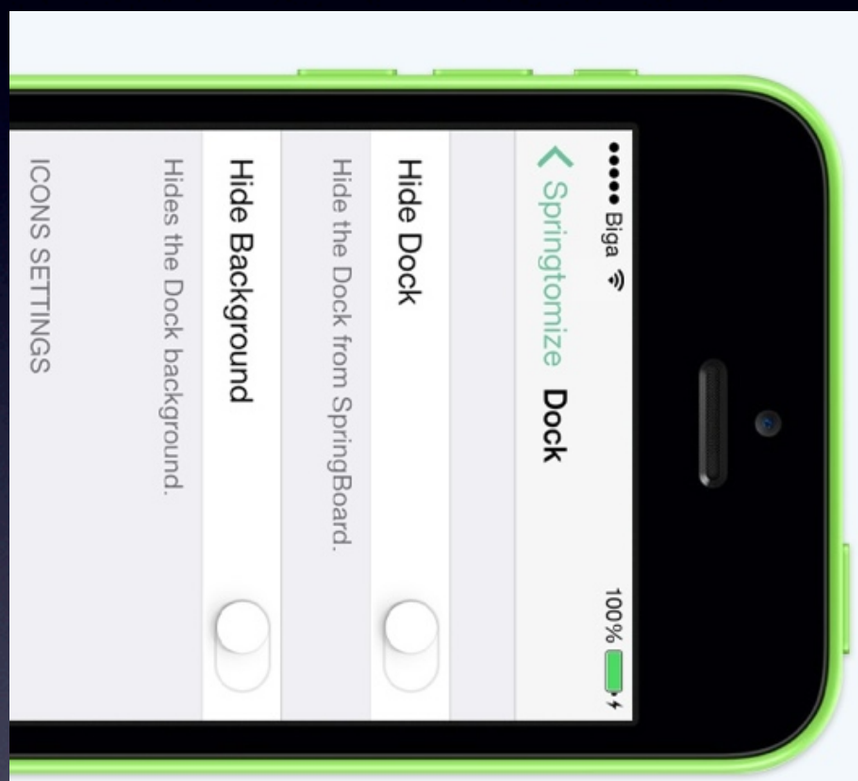
- Reload Process

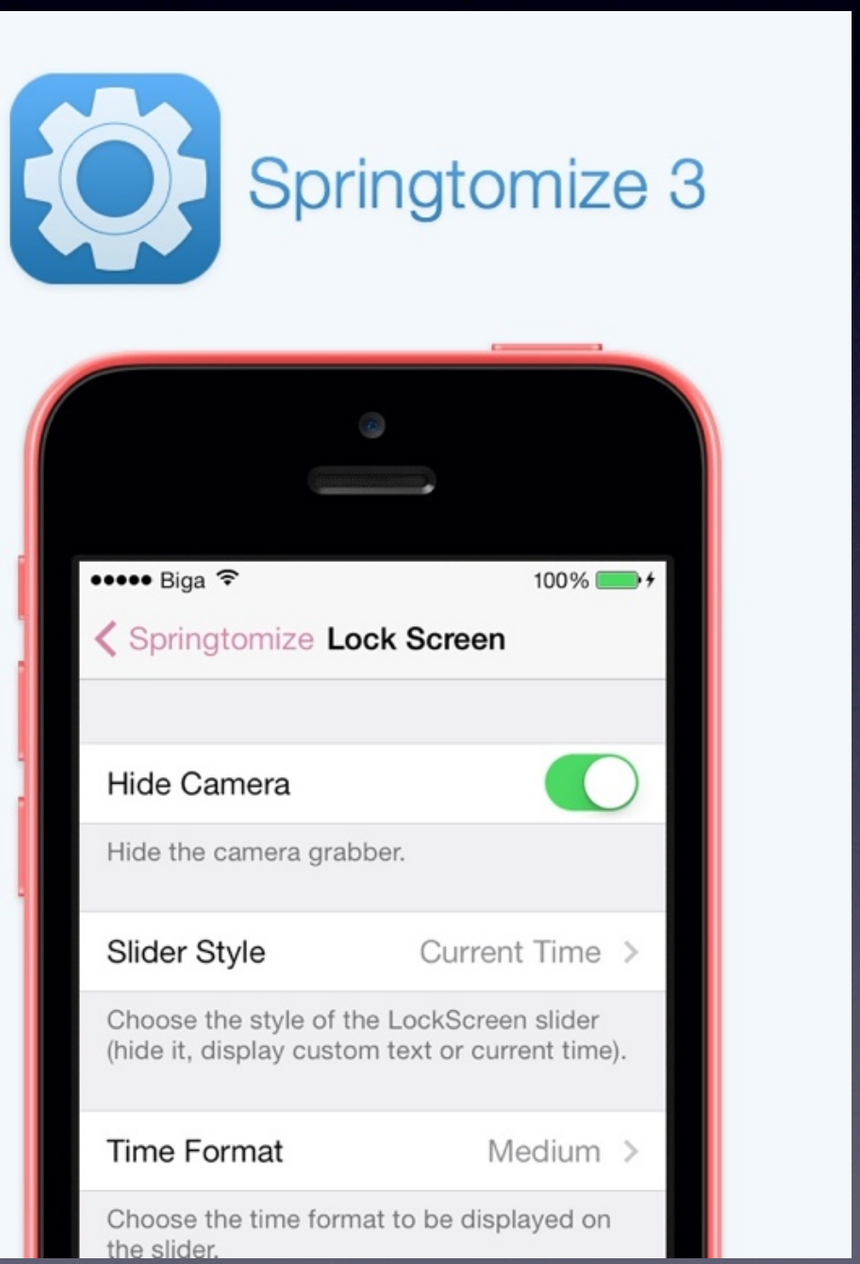


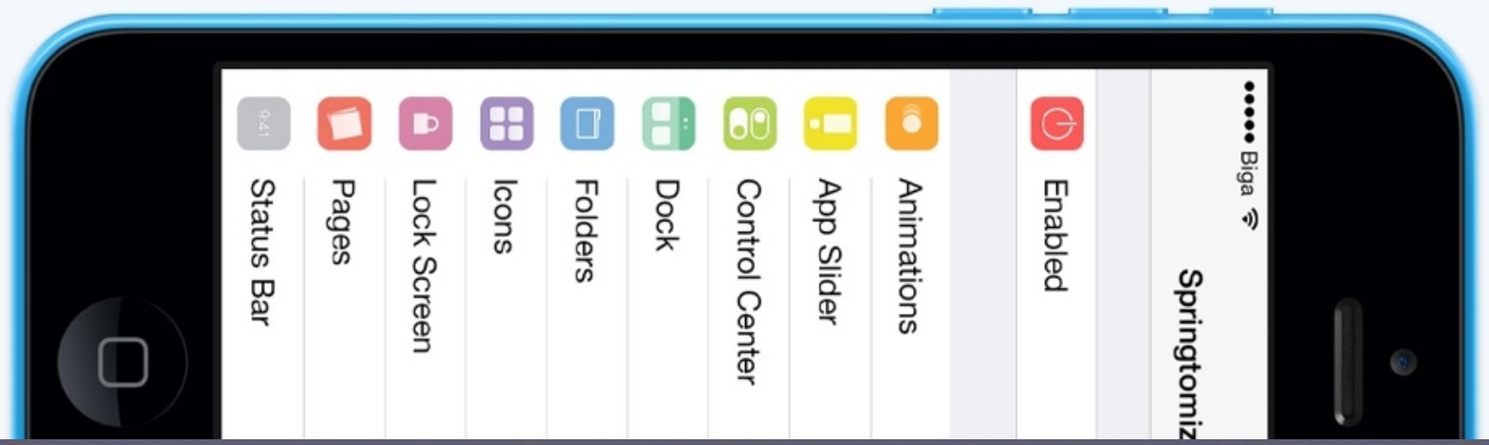
Springtomize 3:
64 bit

- New arm64 architecture of the iPhone 5S
- Need to pay more attention to data types
- Inspect both 32 and 64 bit binaries
- Write portable code

Springtomize 3:
A new interface







Springtomize 3:
A new interface

- New graphic elements by Surenix
- Improved UX
- “Inline” documentation for every option
- Dynamic panels and options
- Everything implemented in code

Springtomize 3: Localization

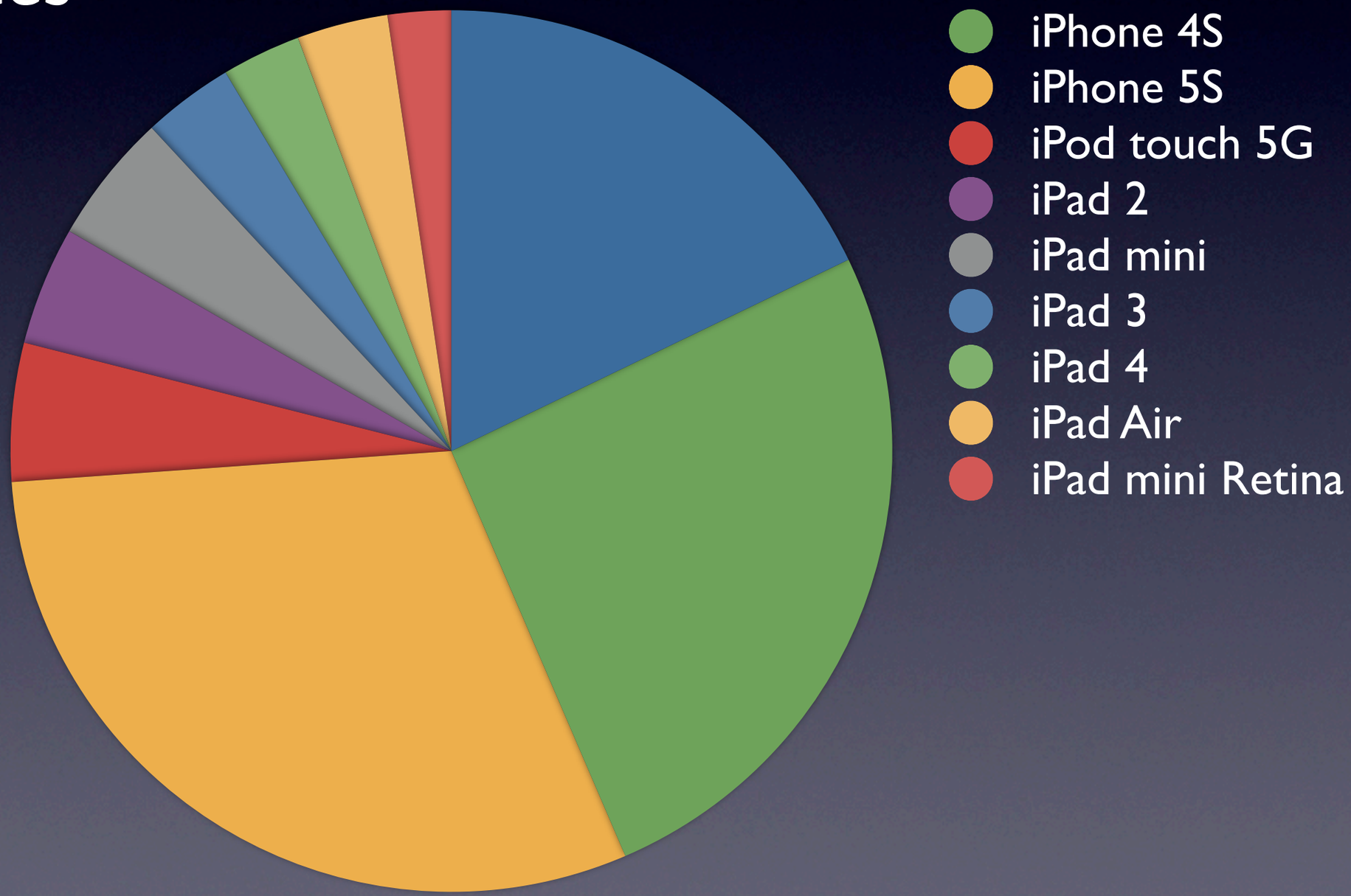
- Settings written with localization in mind
- Automatically updated
- Translated in 20 different languages
  - @Anderson69s, @aphunex, @vroeeem, @RobinShady, @SwissHttp, @ThefferA, @Grolubao, @TommyWelle, @nalbilia, @ipgoogle, @NitiiZZ, @Antonisem_, @Aut0pear, @Commandor, @Stleamist, @Agal1n, @stevenfky, @vicryabov, @denizbatun

Springtomize 3: Stats

- Released on January 25th, 2014
- 6 updates
- 15+ new features already
- Installed on more than 900,000 devices
- ~600 new devices every hour

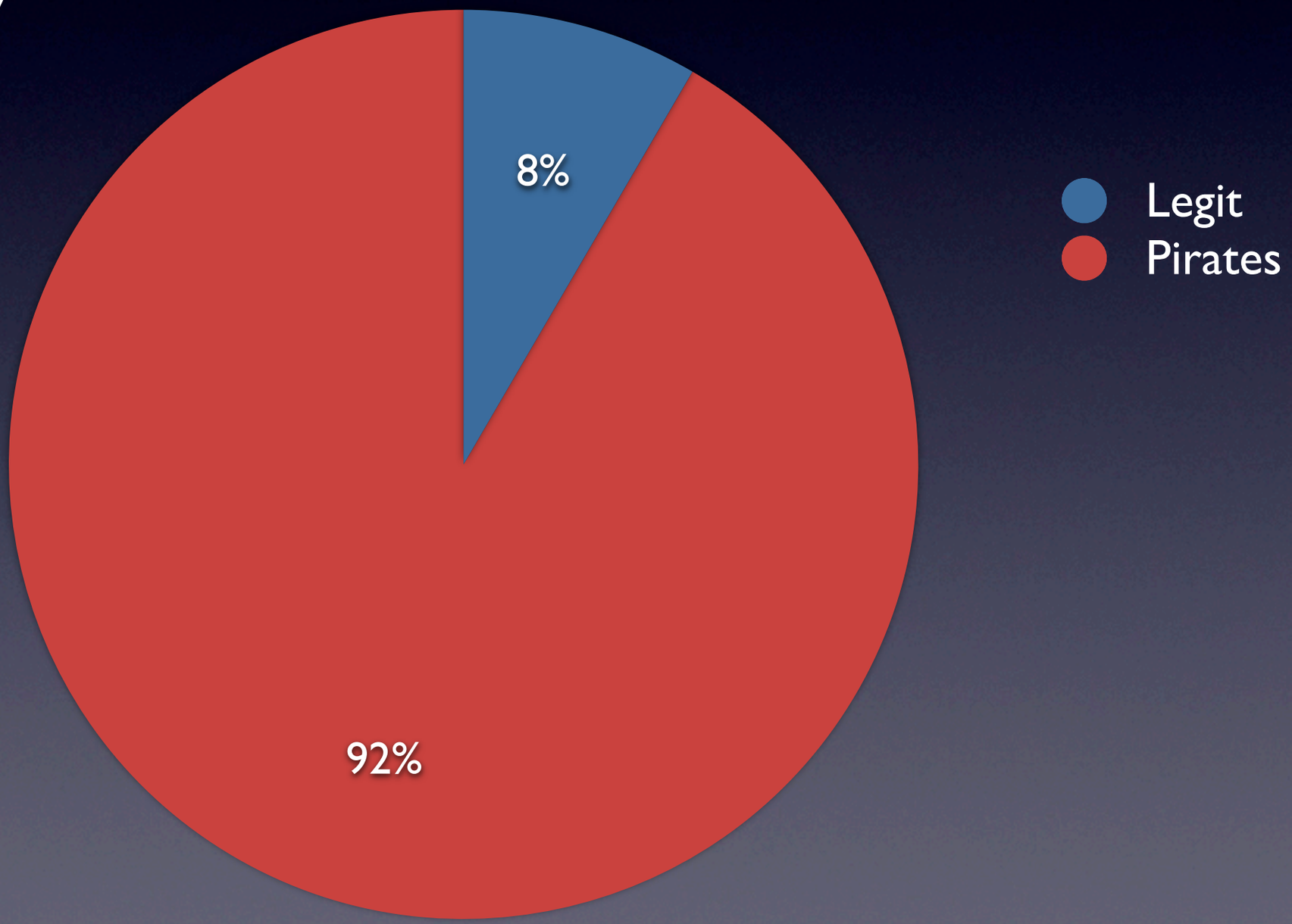
Springtomize 3: Stats

- Devices



Springtomize 3: Stats

- Piracy



Springtomize 3:
Get involved!

- /r/springtomize on Reddit

Conclusions

- Update coming later this month with new features
- Thanks everyone who helped me develop Springtomize 3

Questions?

- Contact
- filippo@filippobiga.com
- @FilippoBiga