

JSPatch 成长之路

陈振焯 (bang)

自我介绍

微信读书 iOS 开发者



自我介绍

微信读书 iOS 开发者

博客

自我介绍

微信读书 iOS 开发者

博客

<http://cnbang.net>

🏠 bang's blog

📁 APPs

☰ 存档

💡 作品

👤 关于

XCode 代码补全插件 – JSPatchX 原理解析

2016-4-26 评论(0) 分类: 技术文章 Tags: ios JSPatch [编辑]

JSPatchX 是 JSPatch Xcode 代码自动补全插件，目前在 github 开源，效果见图：

```
1 require('UIActivityIndicatorView, UITableView, UIView, UIImageView, NSURL')
2
3 defineClass('DBTimelineView', [ 'avatarView' ], {
4   setupSubviews: function() {
5     var view = |
6   }
7 })
8
9
10
11
12
13
14
```

做完一个开源项目照例写篇文章说明下实现原理，主要目的是让想对这个项目做贡献改进的人可以通过文章更容易地了解这个项目的由来，思路，核心原理和流程，降低参与这个项目开发的门槛。

自我介绍

微信读书 iOS 开发者

博客

捣鼓过的东西

自我介绍

微信读书 iOS 开发者

博客

捣鼓过的东西

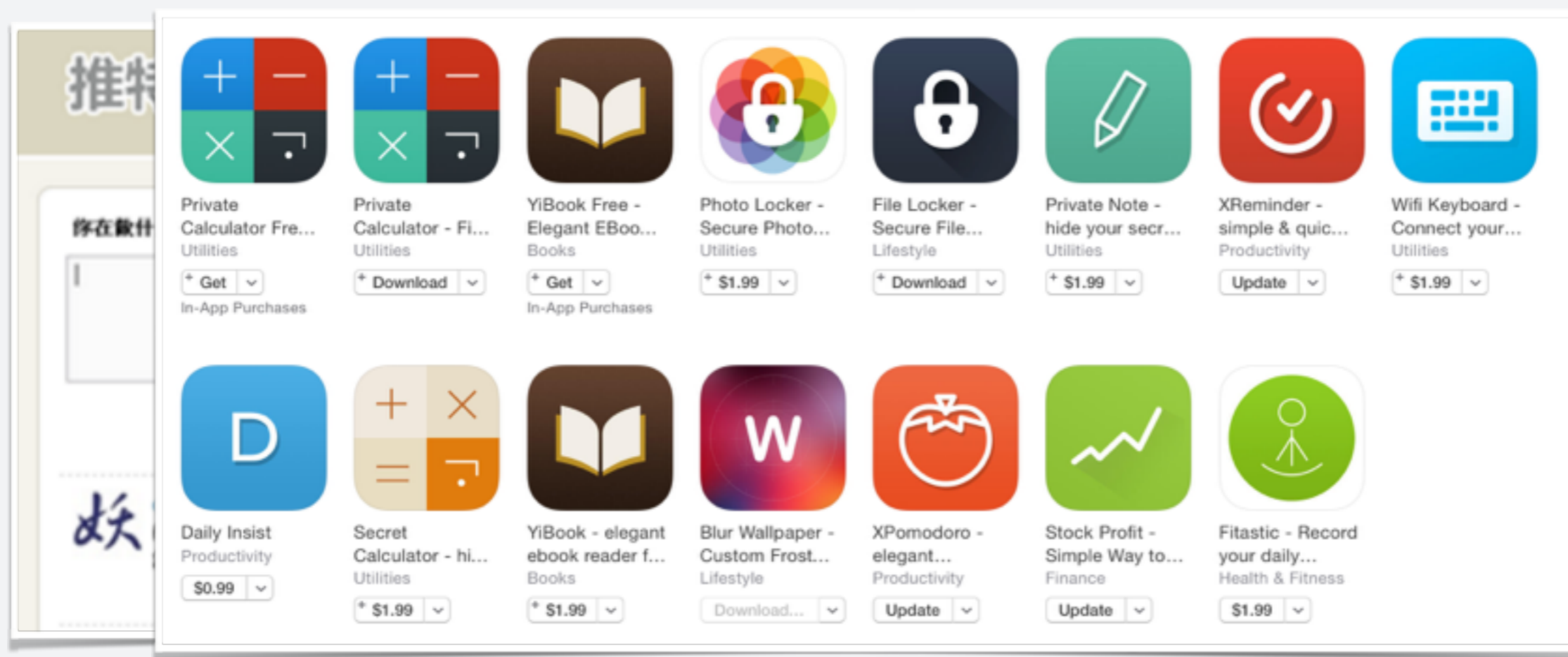
The image shows a screenshot of a Weibo profile page. At the top, there is a header with the text "推特中文圈 Twitese" and navigation links for "首页", "随便看看", "排行榜", and "退出". Below the header, there is a text input area with the prompt "你在做什么?" and a character count "还可以输入 140 个字". A blue "发送" button is located below the input area. To the right of the input area, there is a profile card for the user "bang590" with a profile picture, the name "bang", and statistics: "42 好友", "79 关注者", and "145 消息". Below the profile card, there are four menu items: "全部消息", "我的消息", "@回复我的", and "私信". At the bottom of the page, there is a tweet from user "妖" with the text "qngyina 弹子球。回~ RT @1988jy: 羽毛球RT @Free2E: 篮球白痴们, 都喜欢什么球类啊? RT @1988jy: 篮球白痴联盟RT @BerMQ: 全世界的篮球白痴们团结起来 @kgsoltek @lpistons @wczbb @humanit @1988jy" and a timestamp "通过 Seemik 1分钟前".

自我介绍

微信读书 iOS 开发者

博客

捣鼓过的东西



自我介绍

微信读书 iOS 开发者

博客

捣鼓过的东西

The image shows a screenshot of a GitHub repository page for `bang590 / JSPatch`. The repository is public and has 347 watchers, 5,436 stars, and 1,081 forks. It contains 27 issues, 0 pull requests, and 27 contributors. The repository description states: "JSPatch bridge Objective-C and Javascript using the Objective-C runtime. You can call any Objective-C class and method in JavaScript by just including a small engine. JSPatch is generally use for hotfix iOS App. — Edit". The repository has 282 commits, 1 branch, 10 releases, and 27 contributors. The commit history shows a merge pull request #343 from `gin7758258/master` and several other commits, including "Add demo for tvOS", "Remove iOS code from tvOS build", "Remove private api `JSSynchronousGarbageCollectForDebugging`", "tidy up codes; [bugfix] temporary files won't be cleared if network", and "ignore DS_Store".

On the left side, there is a partial view of an iPhone app store page showing several apps, including "Private Calculator Fre...", "Daily Insist Productivity" (priced at \$0.99), and "Secret Calcula Utilitie".

大纲

- 起步
- 发展
- 下一步

起步



问题

App 线上 bug 修复周期长，成本高

(测试->打包->发布->审核->用户下载)

业界解决方案

waxPatch

waxPatch

停止开发

不支持Block

线程不安全

不支持64位

难以调试

审核风险

没有文档

lua语言陌生

没有测试用例

更好的解决方案

更好的解决方案

JavaScript

更好的解决方案

JavaScript

原生JavaScriptCore引擎

JS 语言终端应用广泛

符合审核规则

更好的解决方案

JavaScript

原生JavaScriptCore引擎

JS 语言终端应用广泛

符合审核规则

有没有人试过?

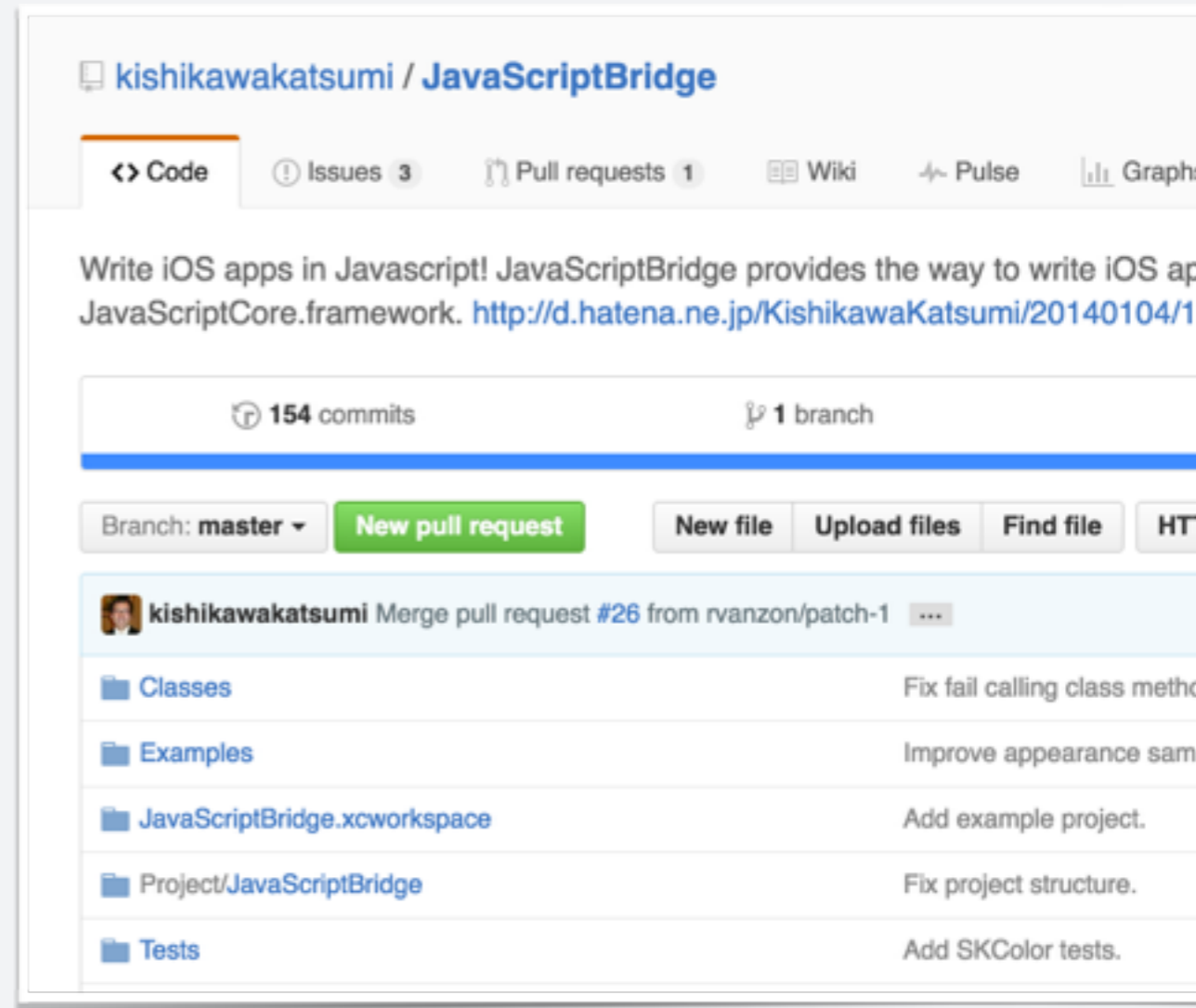
更好的解决方案

JavaScriptBridge

需要编译前定义所有接口

臃肿

实用性低



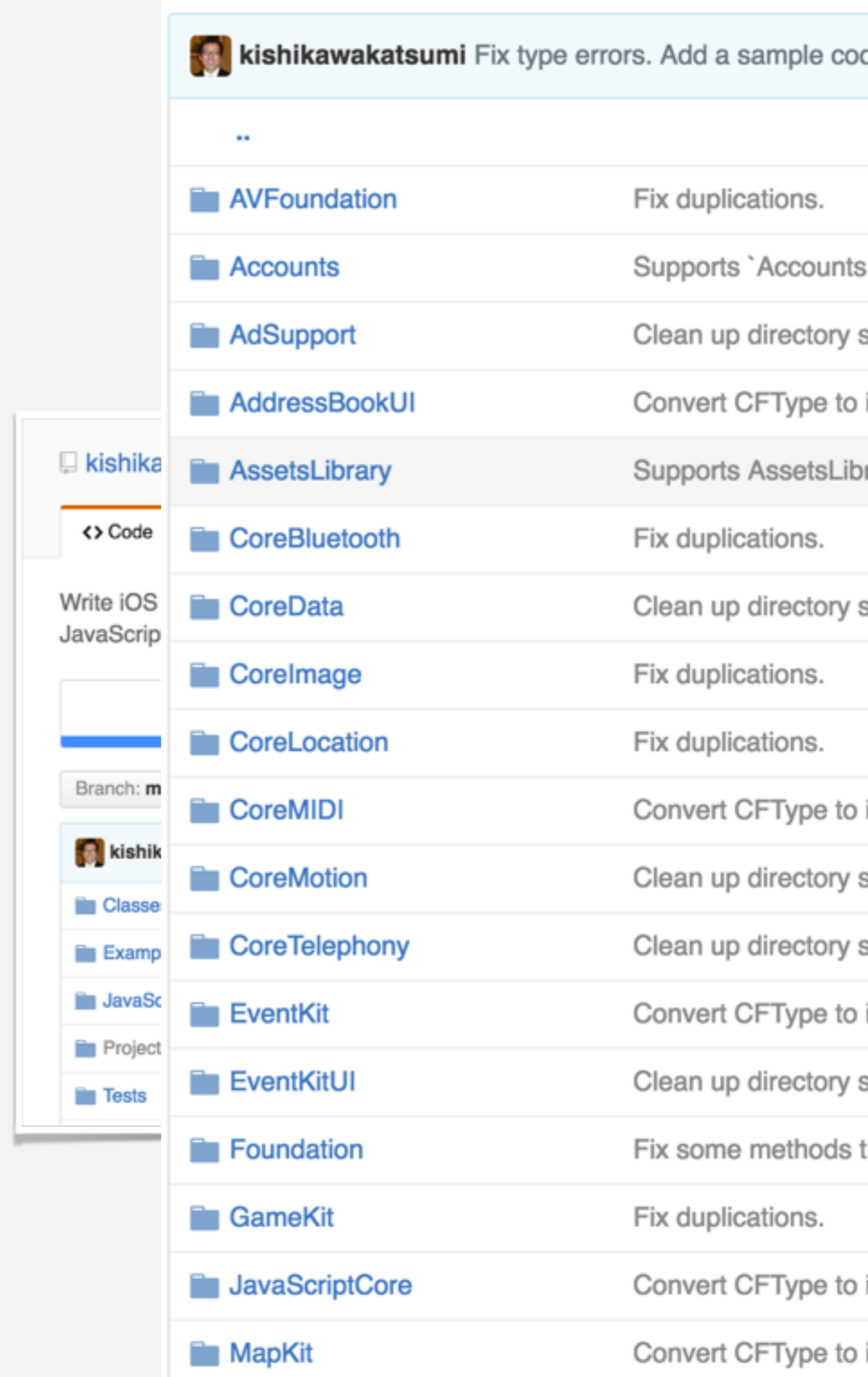
更好的解决方案

JavaScriptBridge

需要编译前定义所有接口

臃肿

实用性低

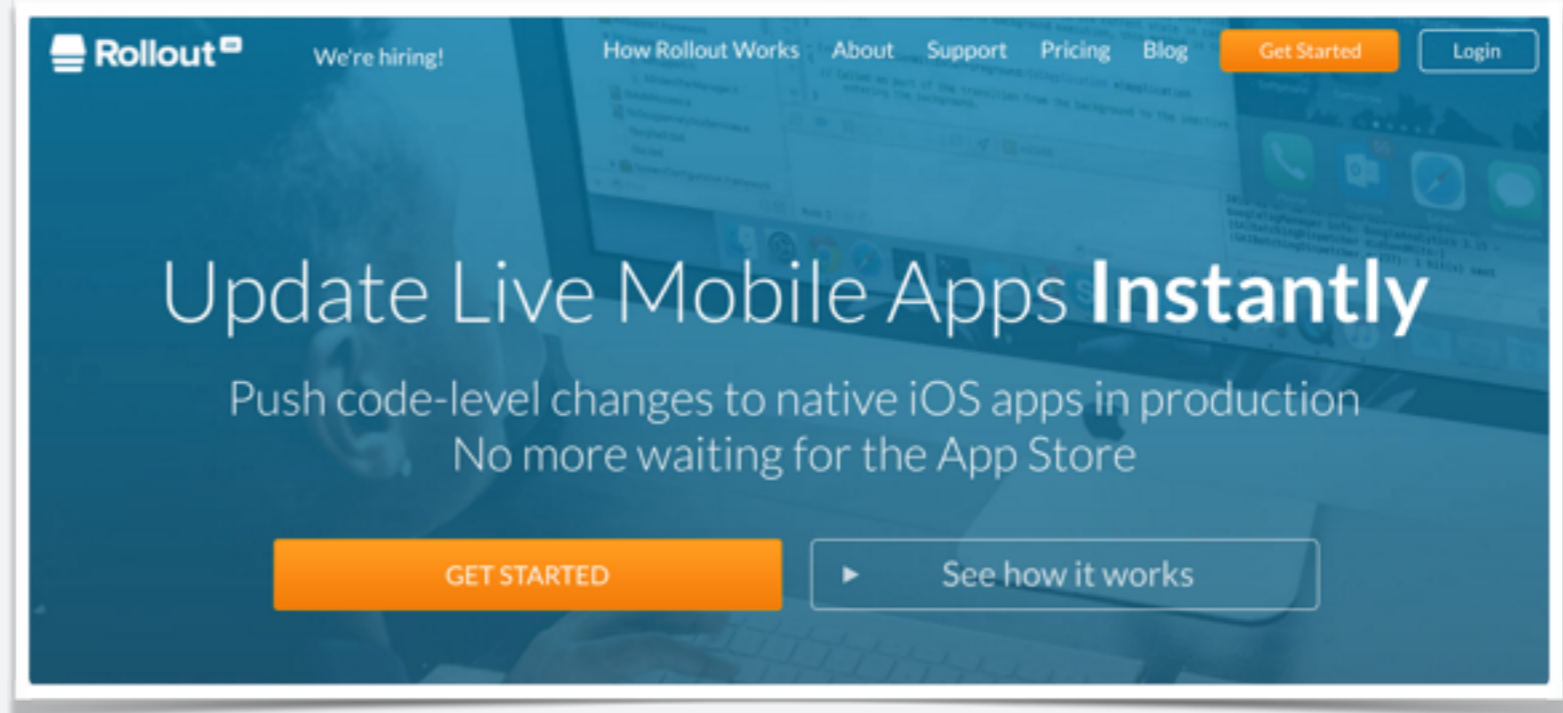


更好的解决方案

Rollout

非开源

写法复杂



```
var obj = R.NSClassFromString('MyObject').call('new')
obj.call('say')self.originalImplementation();
```

核心问题

期望的效果:

UIView.alloc()

JavaScriptCore

className "UIView"
methodName "alloc"

ObjC Runtime

Call [UIView alloc]

核心问题

期望的效果:

UIView.alloc()

JavaScriptCore

className "UIView"
methodName "alloc"

ObjC Runtime

Call [UIView alloc]

```
> UIView.alloc()
```

```
✘ ▶ Uncaught ReferenceError: UIView is not defined(...)
```

核心问题

语言规则： 必须已经是对象成员函数

```
var UIView = {};  
UIView.alloc = function() {  
    //pass "UIView" and "alloc" to OC  
}  
UIView.alloc();
```

核心问题

语言规则： 必须已经是对象成员函数

```
var UIView = {};  
UIView.alloc = function() {  
    //pass "UIView" and "alloc" to OC  
}  
UIView.alloc();
```

```
var UIView = {}  
UIView.alloc = function() {}  
UIView.new = function() {}  
UIView.isEqual = function() {}  
UIView.version = function() {}  
...
```

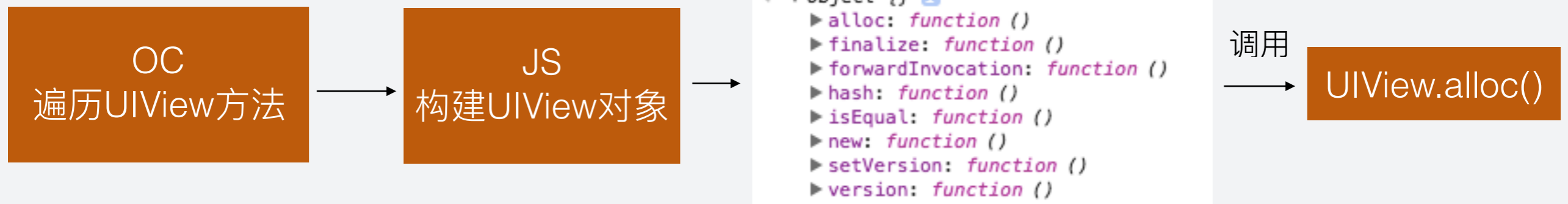

核心问题

每一个类和方法都要事先定义好：



核心问题

每一个类和方法都要事先定义好：



- 方法太多 (NSObject 实例方法830个 类方法118个)
- 占用内存高 (NSObject 1.3M, UIView 2M)

核心问题

每一个类和方法都要事先定义好：



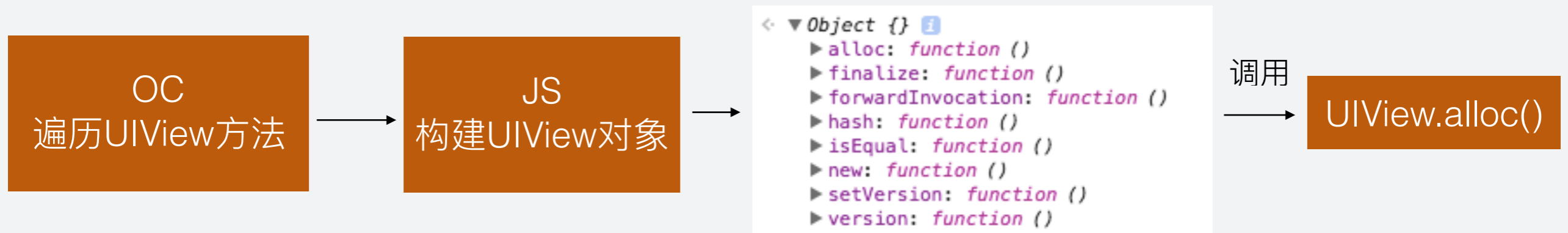
- 方法太多 (NSObject 实例方法830个 类方法118个)
- 占用内存高 (NSObject 1.3M, UIView 2M)

优化？

- 去除私有方法
- JS构建继承链公用基类方法

核心问题

每一个类和方法都要事先定义好：



- 方法太多 (NSObject 实例方法830个 类方法118个)
- 占用内存高 (NSObject 1.3M, UIView 2M)

优化？

- 去除私有方法
- JS构建继承链公用基类方法

✗ 占用内存依旧很高

核心问题

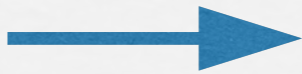
思维定势

必须遵守语言规则

核心问题

预编译

核心问题

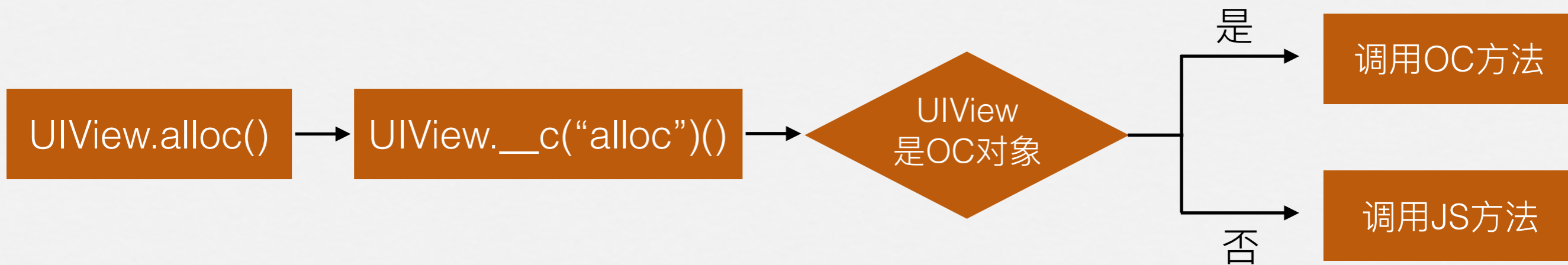
预编译 `UIView.alloc()`  `UIView.__c("alloc")()`

- 正则替换代码
- `__c()` 函数：模拟 Ruby/Lua 元方法机制

核心问题

预编译 `UIView.alloc()` \longrightarrow `UIView.__c("alloc")()`

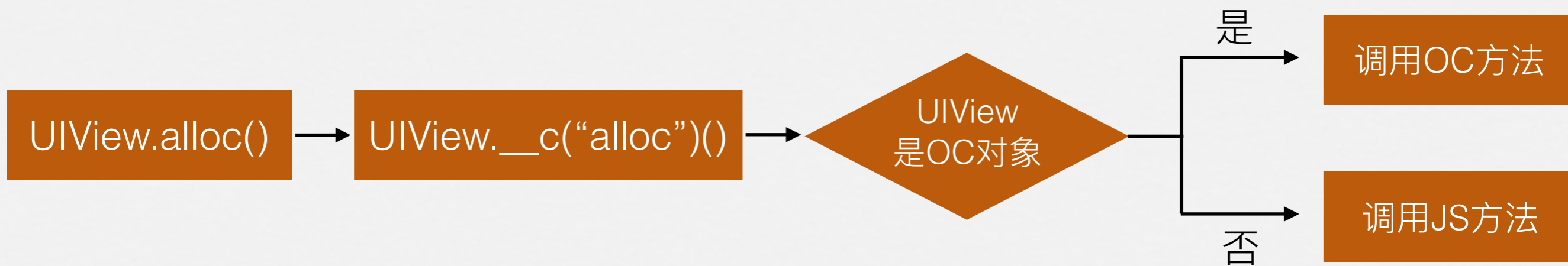
- 正则替换代码
- `__c()` 函数：模拟 Ruby/Lua 元方法机制



核心问题

预编译 `UIView.alloc()` \longrightarrow `UIView.__c("alloc")()`

- 正则替换代码
- `__c()` 函数：模拟 Ruby/Lua 元方法机制



效果:

- 简洁 高效
- 内存占用 \downarrow 99%

完善功能

- 调用/替换OC方法
- 支持64位
- 支持block
- 完整参数类型支持 (C指针 / Class / id* / struct)
- 支持C函数调用

..... <https://github.com/bang590/JSPatch/wiki>

发展



如何做得更好？

如何做得更好？

↓ 降低使用门槛

↑ 提高使用效率

Keep it simple and tiny

保持精巧，保持好用



安全问题

权限大，第三方攻击危害大

方案一：对称加密脚本，固定密钥

- 优点：简单
- 缺点：密钥存客户端，易被破解

方案一：对称加密脚本，固定密钥

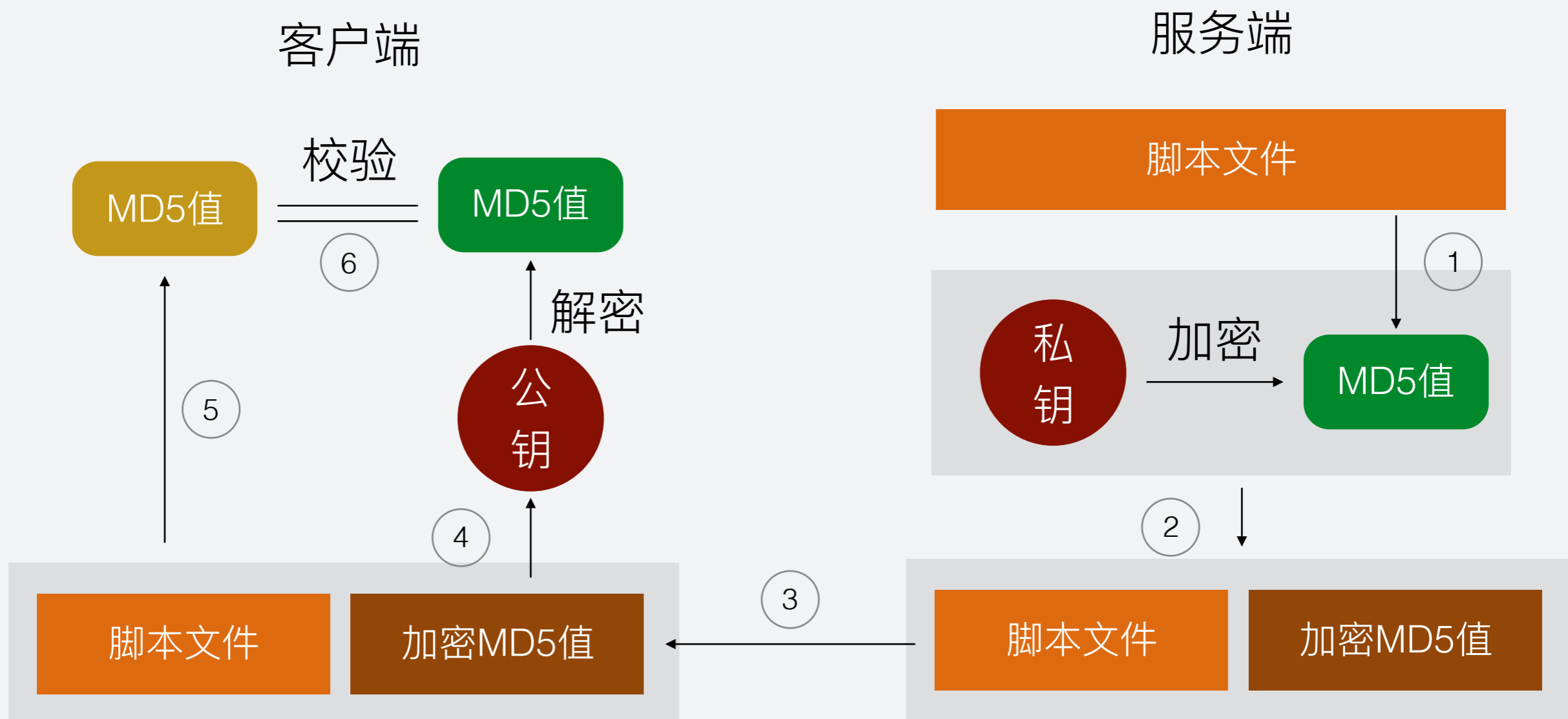
- 优点：简单
- 缺点：密钥存客户端，易被破解

方案二：https传输

- 优点：安全性高
- 缺点：门槛高，部署繁琐

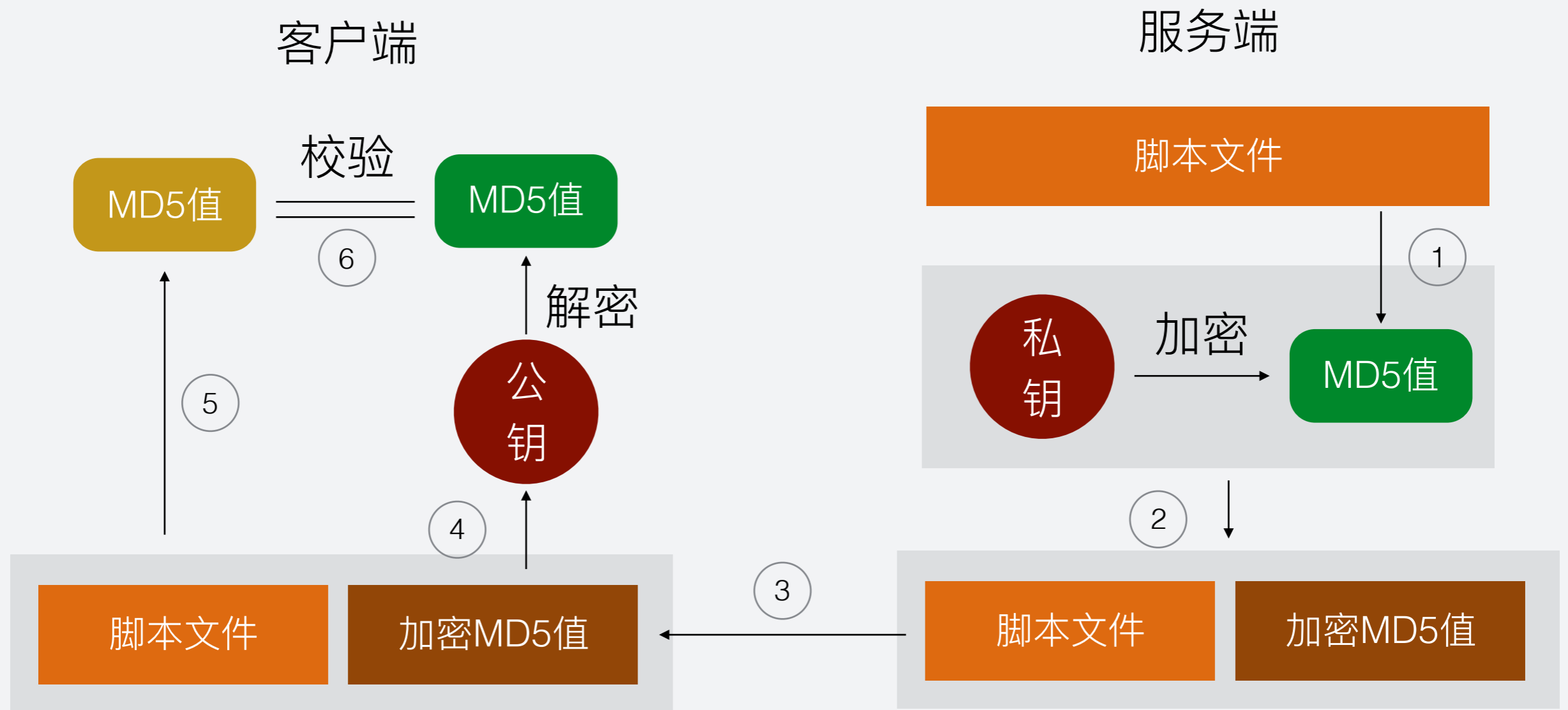
安全策略

方案三：RSA签名验证



安全策略

方案三：RSA签名验证



优点：门槛低，通用性高，部署简单，安全性高

安全策略

JPLoader

bang590 / JSPatch

Unwatch 370

Unstar 5,918

Fork 1,224

Code

Issues 33

Pull requests 0

Wiki

Pulse

Graphs

Settings

Branch: master

JSPatch / Loader /

Create new file

Upload files

Find file

History

rob2468 tidy up codes; [bugfix] temporary files won't be cleared if network ...

Latest commit 446245f on Jan 21

..

libs	Add JSPatch Loader	7 months ago
tools	Add JSPatch Loader	7 months ago
JPLoader.h	Fix compile error	7 months ago
JPLoader.m	tidy up codes; [bugfix] temporary files won't be cleared if network	5 months ago

后台搭建麻烦

JSPatch 平台

<http://jspatch.com>

 JSPatch Platform

[文档](#)

[SDK下载](#)

[登录](#)

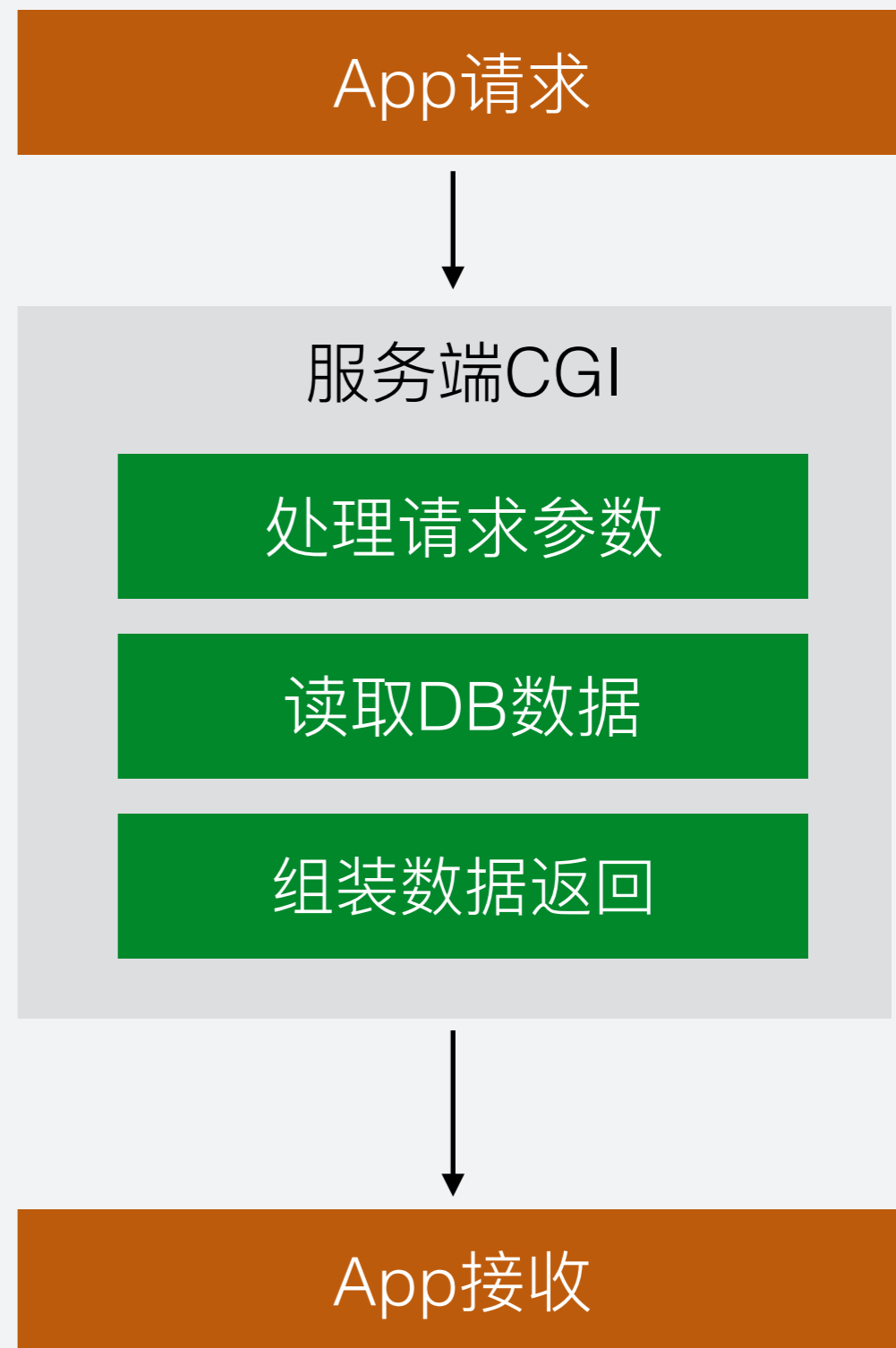
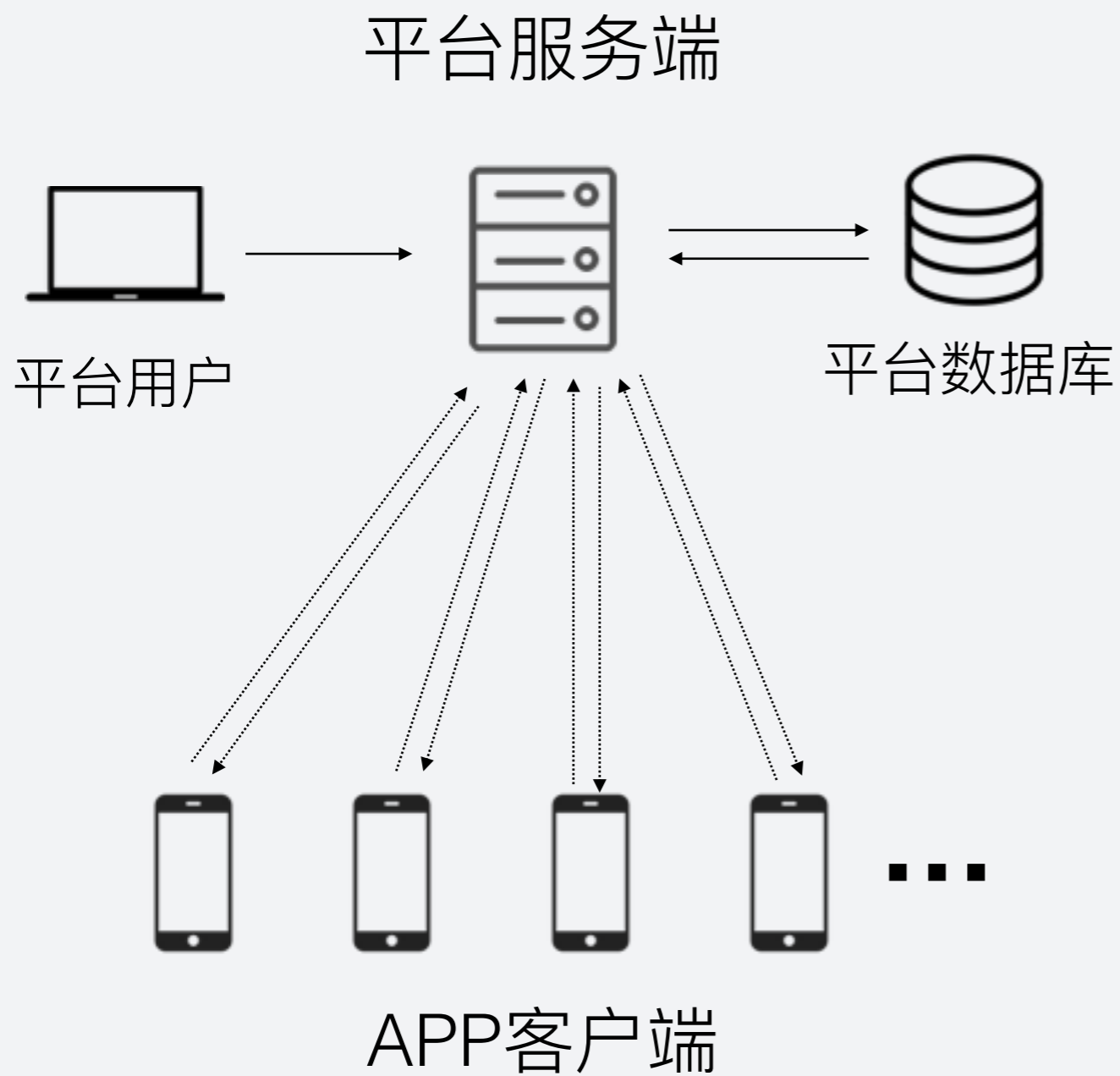
[注册](#)

App 动态更新服务平台

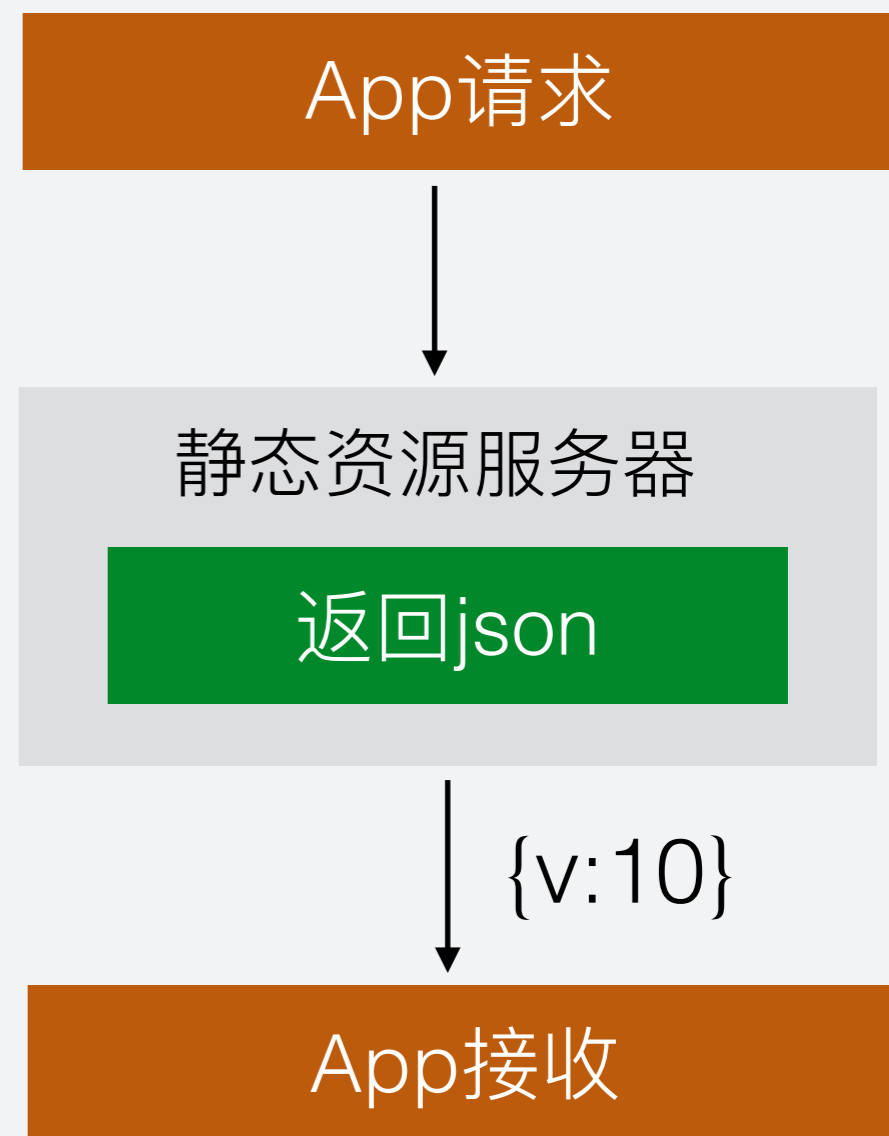
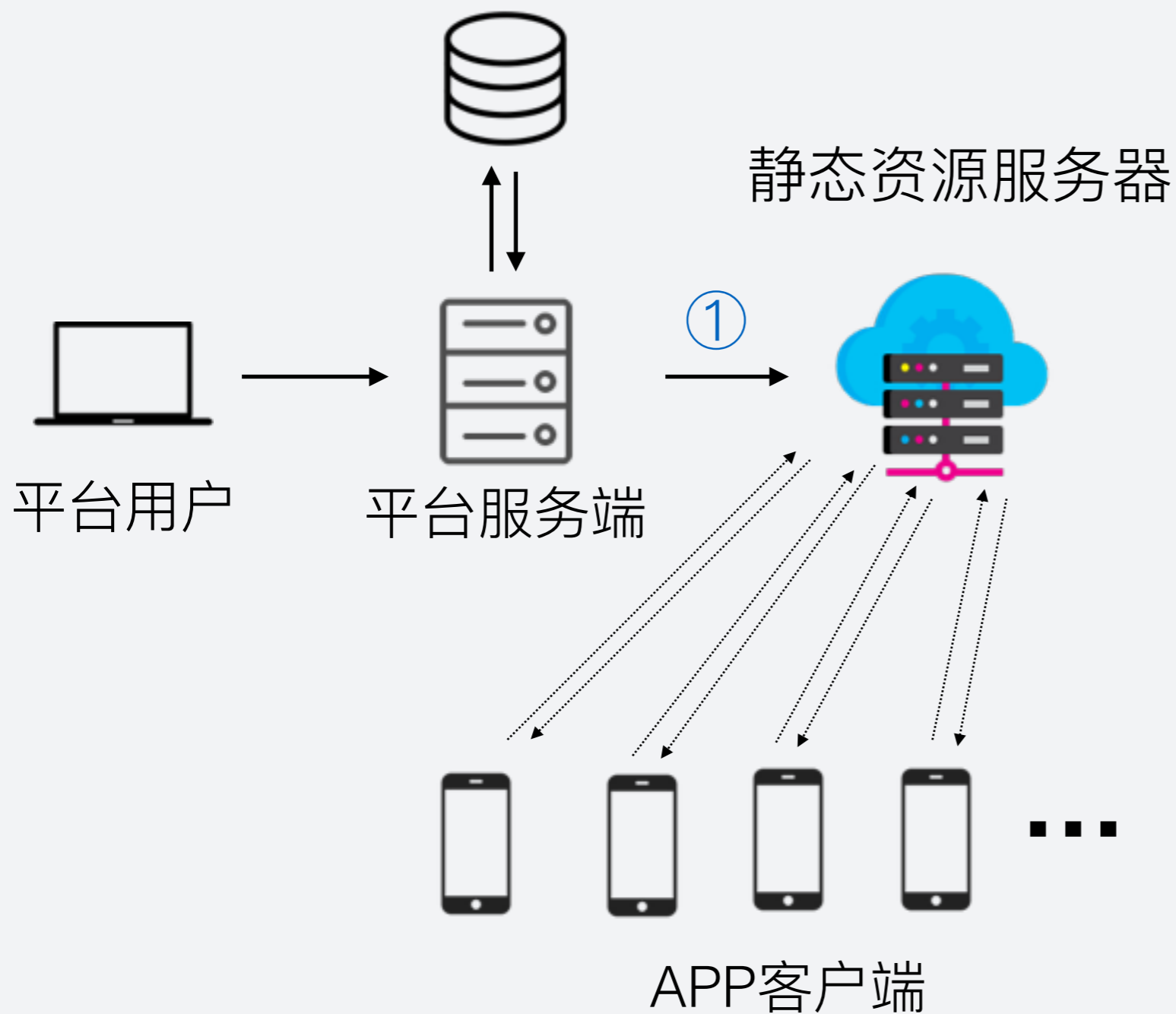
[平台介绍](#)

问题：如何支撑高并发？

JSPatch平台



JSPatch平台



① 文件名: {appKey} 内容: {v:10}

转换代码效率低

人工翻译整个方法内容

JSPatch Convertor



Objective-C

```
@implementation SampleClass
- (void)requestUrl:(NSString *)url param:(NSDictionary *)dict callback:(JPCallback)callback {
    [super requestUrl:url param:dict callback:callback];
    JPRequest *obj = [[JPRequest alloc] initWithUrl:url param:dict];
    obj.successBlock = ^(id data, NSError *err) {
        NSString *content = [JPParser parseData:data];
        if (callback) callback(@{
            @"content": content
        }, err);
        [self.dataSource refresh];
        self.handleRequestSuccess(data);
    };
}
@end
```

Javascript

```
require('JPRequest,JPParser');
defineClass('SampleClass', {
    requestUrl_param_callback: function(url, dict, callback) {
        self.super().requestUrl_param_callback(url, dict, callback);
        var obj = JPRequest.alloc().initWithUrl_param(url, dict);
        obj.setSuccessBlock(block('id,NSError', function(data, err) {
            var content = JPParser.parseData(data);
            if (callback) callback({
                "content": content
            }, err);
            self.dataSource().refresh();
            self.handleRequestSuccess()(data);
        }));
    },
});
```

Convert

总结

降低使用门槛

- 保证易用性
- JSPatch 安全方案 (JPLoader)
- JSPatch 平台
- 完善的文档 / 原理解析文章

提高使用效率

- JSPatch Convertor
- Extensions

超过 2500 个 APP 已接入使用

using.jspatch.org



微信



QQ



支付宝



美团



百度地图



微博



百度



滴滴出行



腾讯视频



QQ浏览器



百度贴吧



京东



网易新闻



携程旅行



唯品会



美团外卖



猫眼电影



饿了么



微信读书



百度阅读



乐视视频



QQ阅读



去哪儿



腾讯地图



口袋购物



美丽说



蘑菇街



1号店



虎牙直播



斗鱼



安居客



天弘爱理财



上海链家



钉钉



百姓网



网易漫画



Boss直聘



神州租车



卷皮折扣



美团外卖商家



虎扑跑步



明道



中华万年历



艺龙出行



孕育管家



Simeji



美柚



战旗TV



银联钱包



知了

下一步



使用JSPatch开发功能模块

React Native / Weex

- 前端 -> 终端 (学习成本)
- 大型(环境配置, 包大小)
- 组件 & 坑

JSPatch

- 学习成本低
- 微型
- 复用所有 OC 组件, 坑少

问题

开发效率 & 运行效率

开发效率

JSPatchX

代码补全

```
1 require('UIActivityIndicatorView, UITableView, UIView, UIImageView, NSURL')
2
3 defineClass('DBTimelineView', [ 'avatarView' ], {
4     setupSubviews: function() {
5
6     }
7 })
8
9
10
11
12
13
14
```

开发效率

JSPatchX

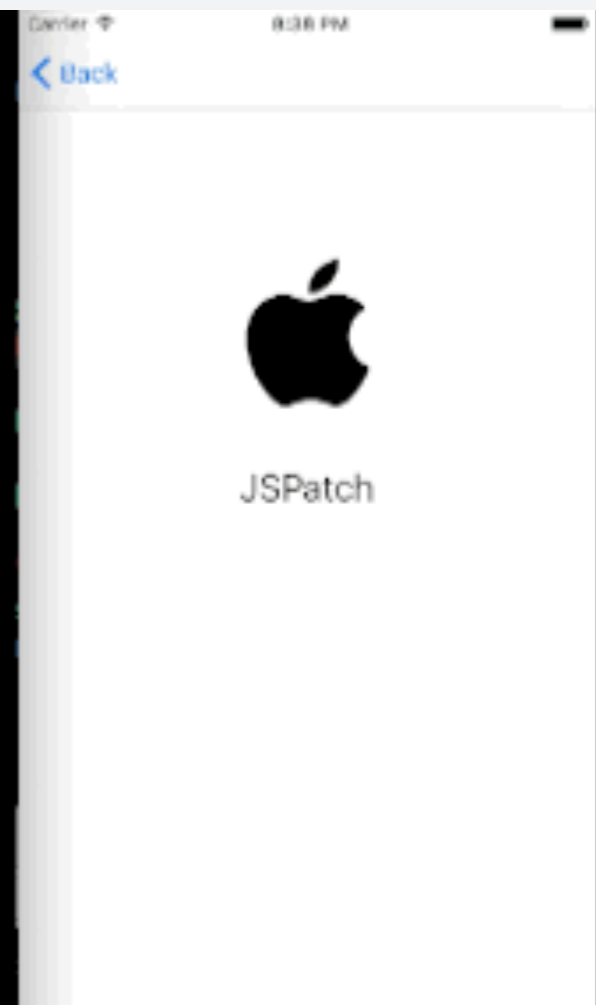
代码补全

```
1 require('UIActivityIndicatorView, UITableView, UIView, UIImageView, NSURL')
2
3 defineClass('DBTimelineView', [ 'avatarView' ], {
4     setupSubviews: function() {
5
6     }
7 })
8
9
10
11
12
13
14
```

PlayGround

即时刷新

```
1 require('UILabel, UIColor, UIFont, UIScreen, UIImageView, UIImage')
2
3 var screenWidth = UIScreen.mainScreen().bounds().width;
4 var screenHeight = UIScreen.mainScreen().bounds().height;
5
6 defineClass('JPDemoController: UIViewController', {
7   viewDidLoad: function() {
8     self.super().viewDidLoad();
9
10    var bgColor = UIColor.colorWithRed_green_blue_alpha(1, 1, 1, 1);
11    self.view().setBackgroundColor(bgColor);
12
13    var size = 100;
14    var imgView = UIImageView.alloc().initWithFrame({x: (screenWidth - size)/2, y:
15      160, width: size, height: size});
16    var img = UIImage.imageWithContentsOfFile(resourcePath('apple.png'));
17    imgView.setImage(img);
18    self.view().addSubview(imgView);
19
20    var imgFrame = imgView.frame();
21    var label = UILabel.alloc().initWithFrame({x: 0, y: imgFrame.height + imgFrame
22      .y + 30, width: screenWidth, height: 40});
23    label.setText("JSPatch");
24    label.setTextAlignment(1);
25    label.setFont(UIFont.systemFontOfSize(24));
26    self.view().addSubview(label);
27  },
28  },
29  );
```



PlayGround

即时刷新

```
1 require('UILabel, UIColor, UIFont, UIScreen, UIImageView, UIImage')
2
3 var screenWidth = UIScreen.mainScreen().bounds().width;
4 var screenHeight = UIScreen.mainScreen().bounds().height;
5
6 defineClass('JPDemoController: UIViewController', {
7   viewDidLoad: function() {
8     self.super().viewDidLoad();
9
10    var bgColor = UIColor.colorWithRed_green_blue_alpha(1, 1, 1, 1);
11    self.view().setBackgroundColor(bgColor);
12
13    var size = 100;
14    var imgView = UIImageView.alloc().initWithFrame({x: (screenWidth - size)/2, y:
15      160, width: size, height: size});
16    var img = UIImage.imageWithContentsOfFile(resourcePath('apple.png'));
17    imgView.setImage(img);
18    self.view().addSubview(imgView);
19
20    var imgFrame = imgView.frame();
21    var label = UILabel.alloc().initWithFrame({x: 0, y: imgFrame.height + imgFrame
22      .y + 30, width: screenWidth, height: 40});
23    label.setText("JSPatch");
24    label.setTextAlignment(1);
25    label.setFont(UIFont.systemFontOfSize(24));
26    self.view().addSubview(label);
27  },
28});
```



确定瓶颈

```
defineClass('JPDribbbleView:UIView', {  
  renderItem: function(item) {  
    ...  
  },  
})  
  
defineClass('JPDribbbleViewController:UIViewController', {  
  render: function(){  
    var view = JPDribbbleView.alloc().init();  
    view.renderItem(item);  
  }  
});
```

运行效率

进入 __c

转换参数

callSelector

字符串转 Class & Selector 调用

JPForwardInvocation

包装/转换参数

调用 JS 方法

转换返回值类型

JS拿到返回值 调用结束

运行效率

进入 __c

转换参数

callSelector

字符串转 Class & Selector 调用

JPForwardInvocation

包装/转换参数

调用 JS 方法

转换返回值类型

JS拿到返回值 调用结束

运行效率

进入 __c

转换参数

callSelector

字符串转 Class & Selector 调用

JPForwardInvocation

包装/转换参数

调用 JS 方法

转换返回值类型

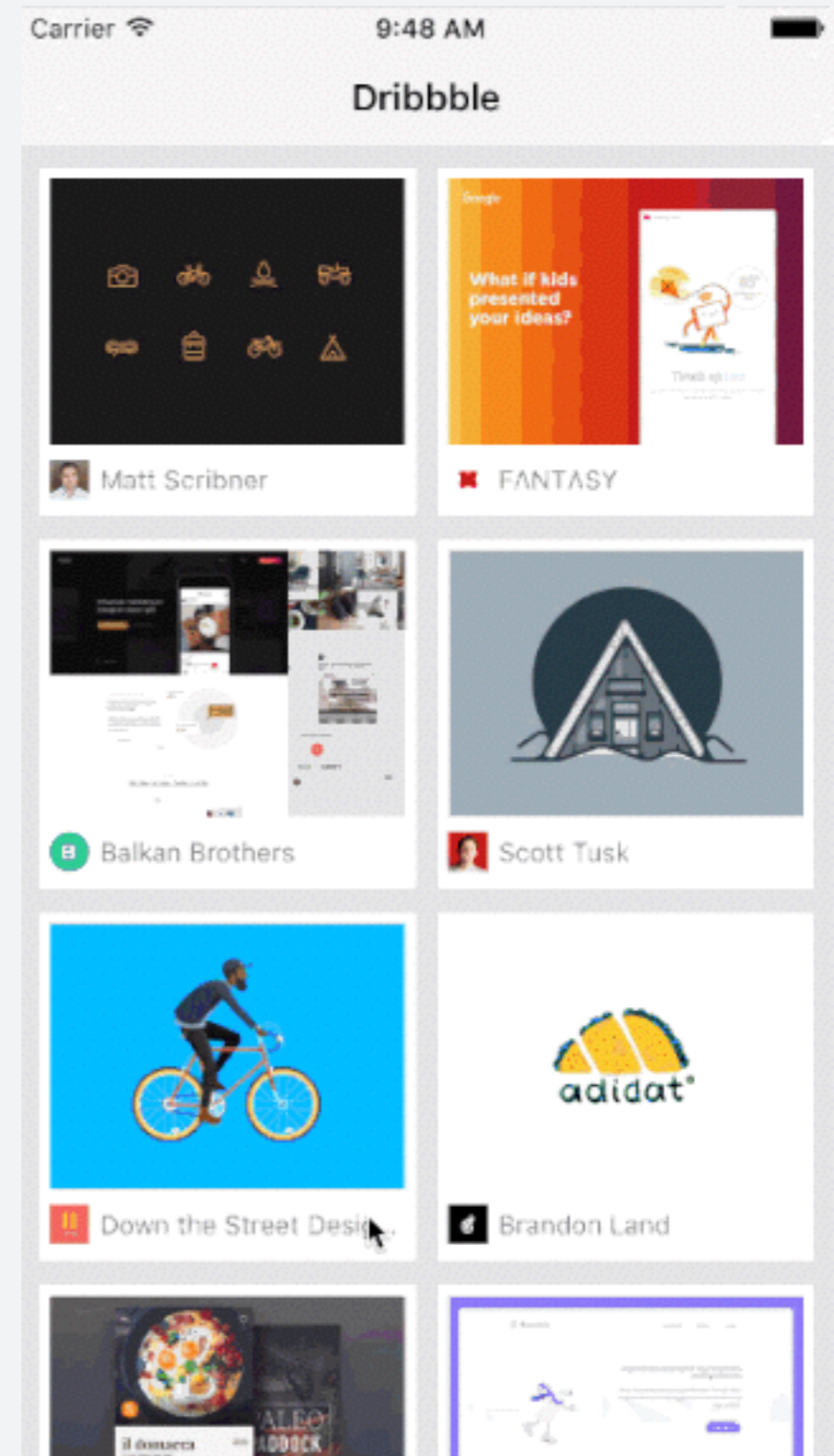
JS拿到返回值 调用结束

性能最高提升 700 倍

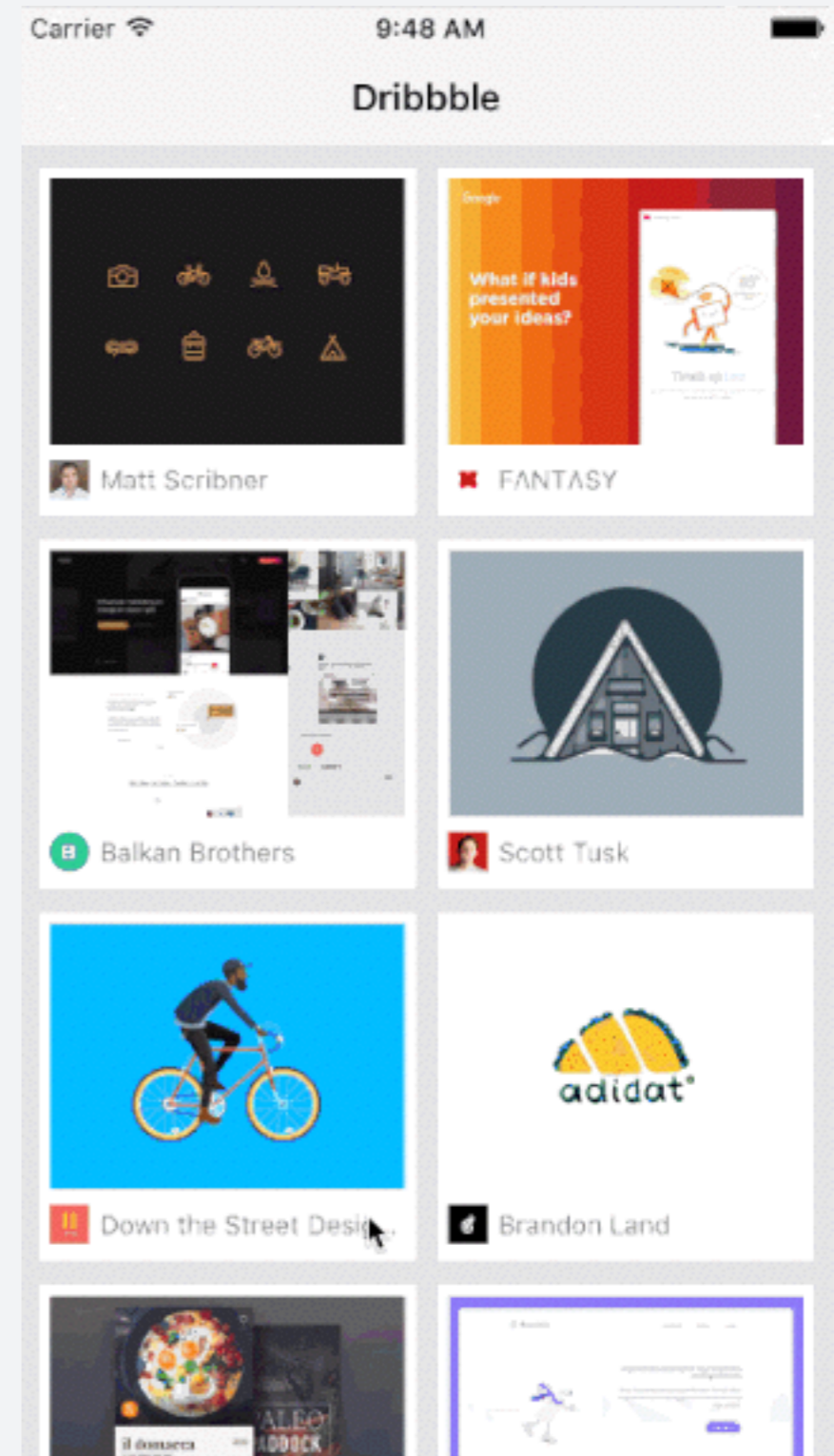
运行效率

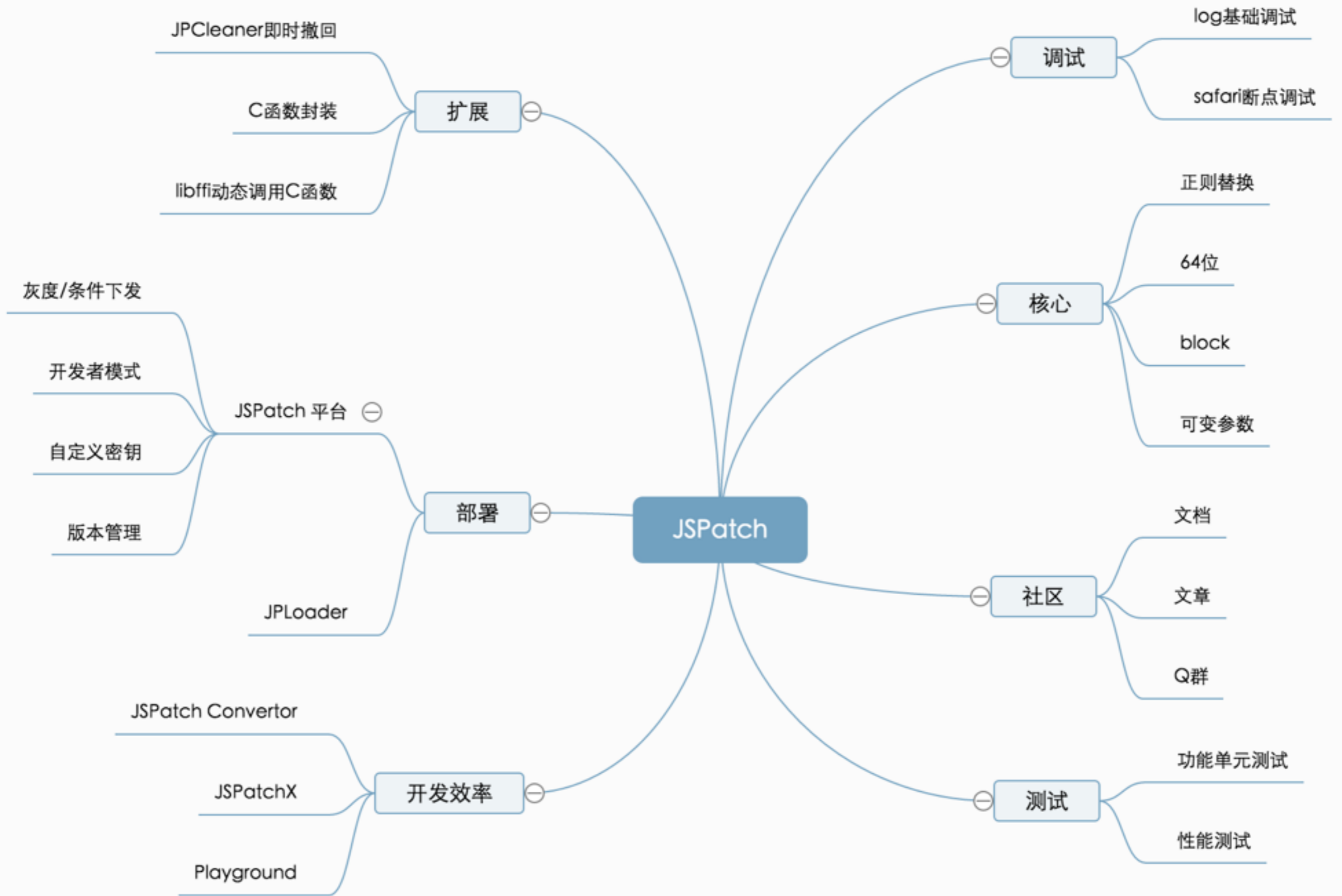
- 提升新增 Property 性能
- 纯 JS Class (写法一致)
- 自动转换 OC 字符串/数组等类型

Dribbble demo



Dribbble demo





THANKS