OSC源创会
年终盛典 2016

Hybrid技术在Flyme的应用实践

BY 杨德模

2016.12.04

About me

OSC源创会
年终盛典 2016

- 2011.7-2014.6 腾讯 WebQQ、Q+、手Q、Q群
- 2014.7-2015.10 唯品会 特卖会移动平台前端架构
- 2015.11- 魅族科技，Hybrid App架构设计和落地、前端生态和基础设施建设

- https://github.com/chemdemo

大纲

OSC源创会
年终盛典 2016

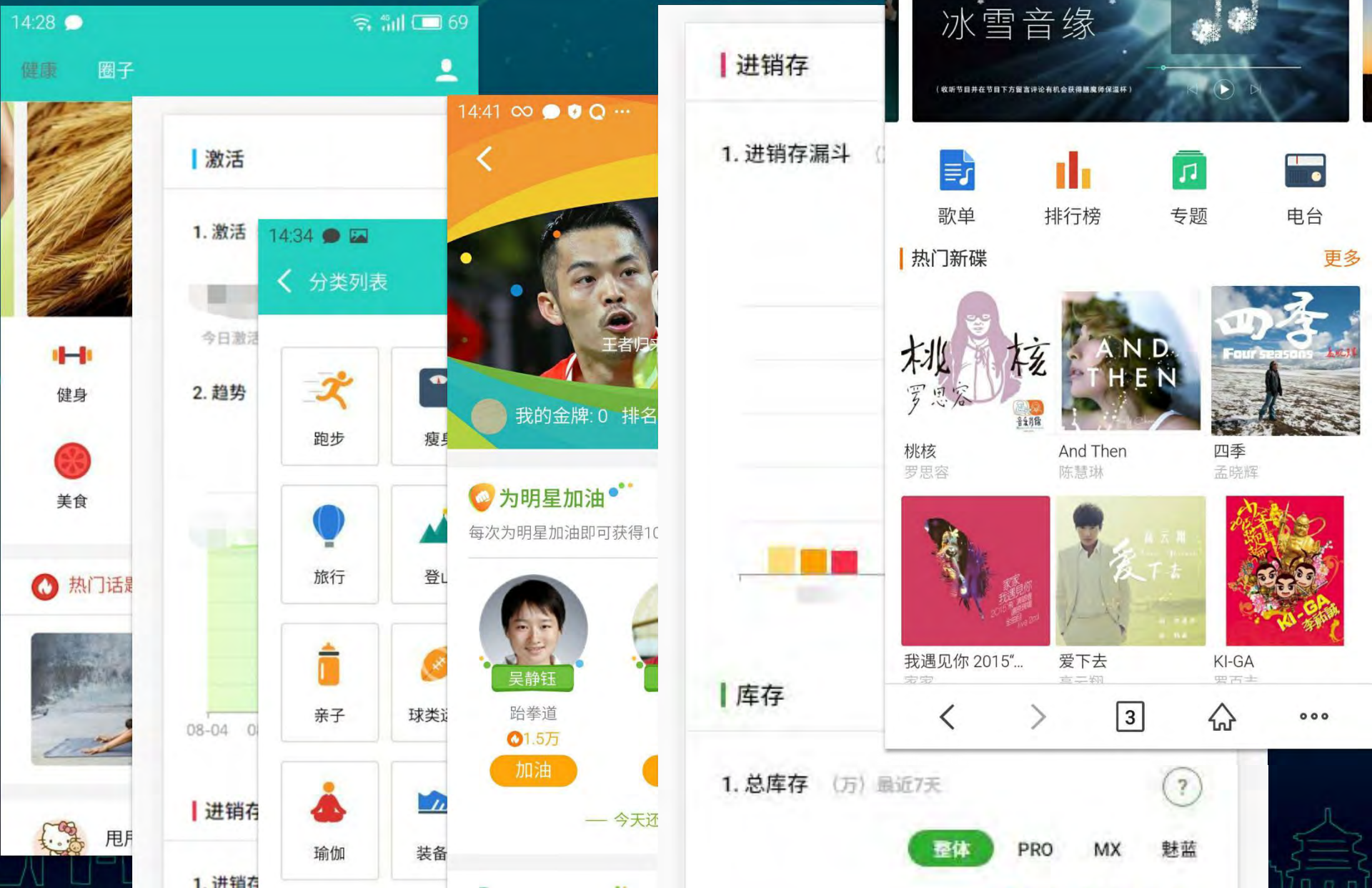
1. Flyme hybrid 原有架构简析

2. 通用Hybrid App开发体系的建设

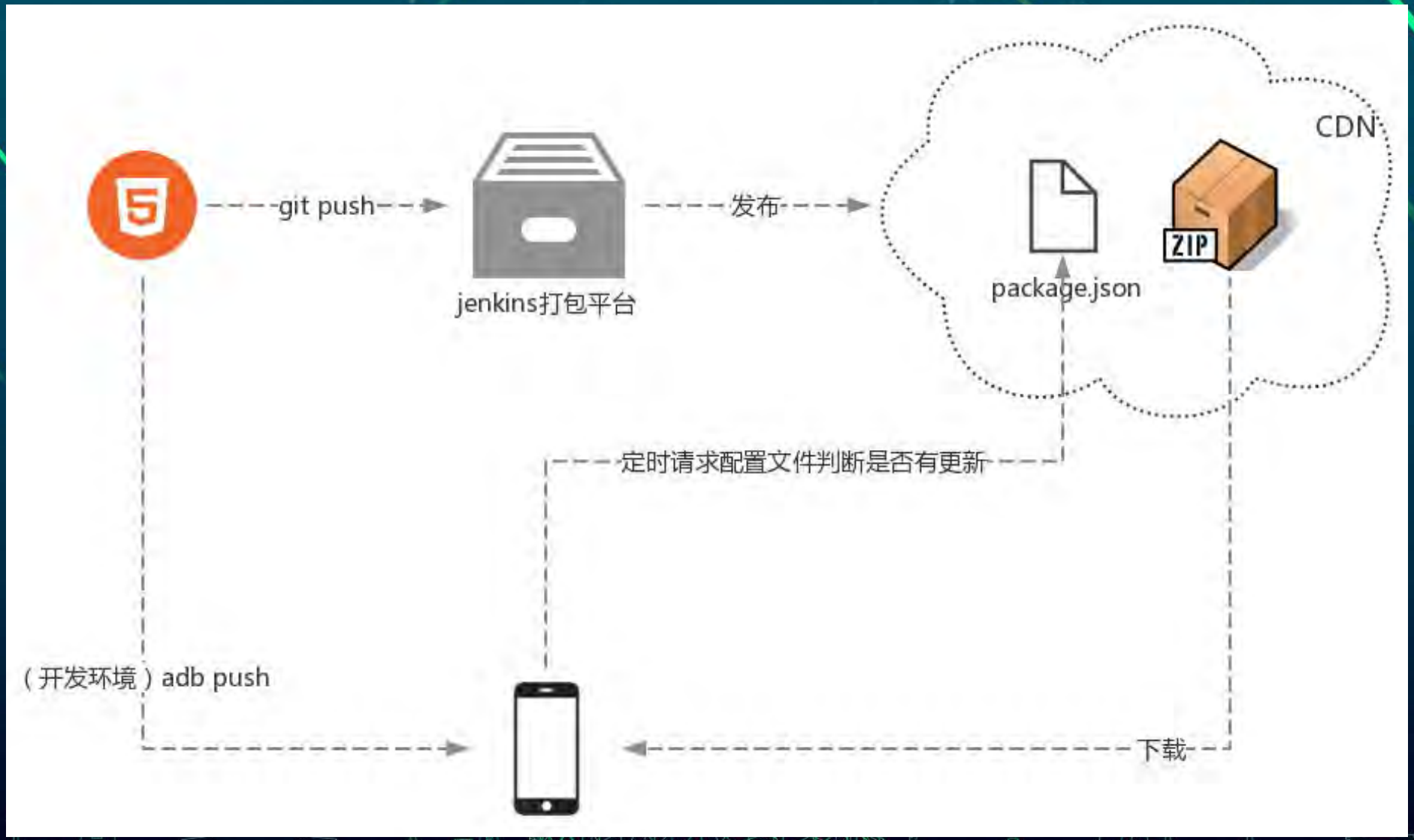
   - 通讯
   - 资源管理
   - 性能优化

3. 前端工程、组件生态、工具链

Hybrid Apps on FlymeOS

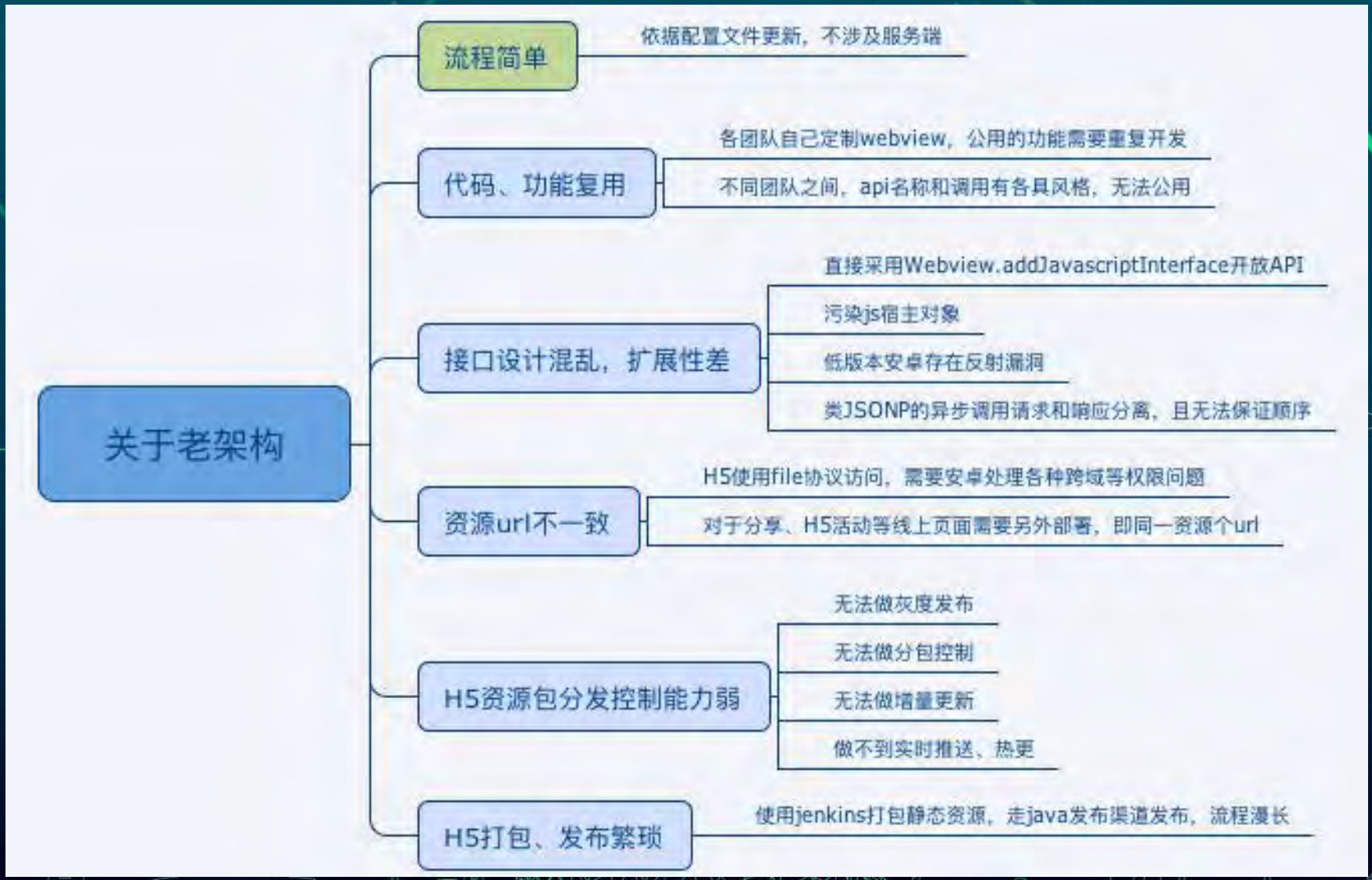


Flyme Hybrid 原有架构流程



Flyme hybrid 原有架构

OSC源创会 年终盛典 2016



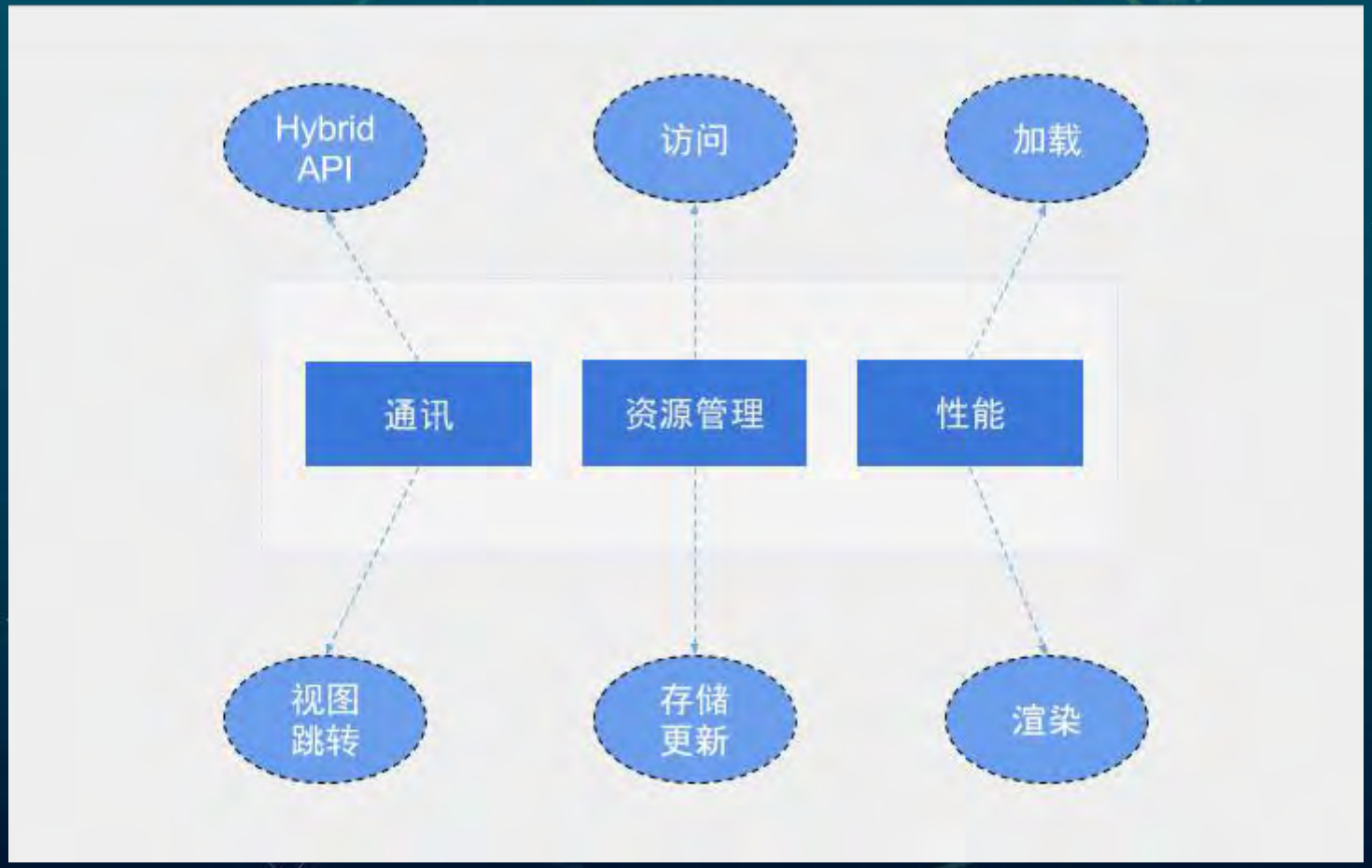
改进思路

OSC源创会
年终盛典 2016

问题	解决方案
代码复用	模块化、SDK、前端工程
接口设计	模块化，优雅设计
访问url不一致	在线和离线均采用唯一url
H5包分发控制	包动态更新接口+push通道
H5编译、打包、发布	自建包管理平台+前端工具

通用的Hybrid架构

OSC源创会
年终盛典 2016



设计理论：https://github.com/chemdemo/chemdemo.github.io/issues/12

通讯——基本原理

OSC源创会
年终盛典 2016

- Android调用H5

```
webview.loadUrl("javascript: alert('hello world')");
```

- Android调用H5：shouldOverrideUrlLoading

```
@Override
public boolean shouldOverrideUrlLoading(WebView view, String url) {
    // 自定义的schema
    if (url.indexOf("myschema://") != -1) {
        ... // other code
        // 返回true 则表明webview已经“消费”了H5的request事件
        return true;
    }
    // 返回false，webview将用默认的方式处理H5的request事件
    return false;
}
```

通讯——我们的实现

OSC源创会
年终盛典 2016

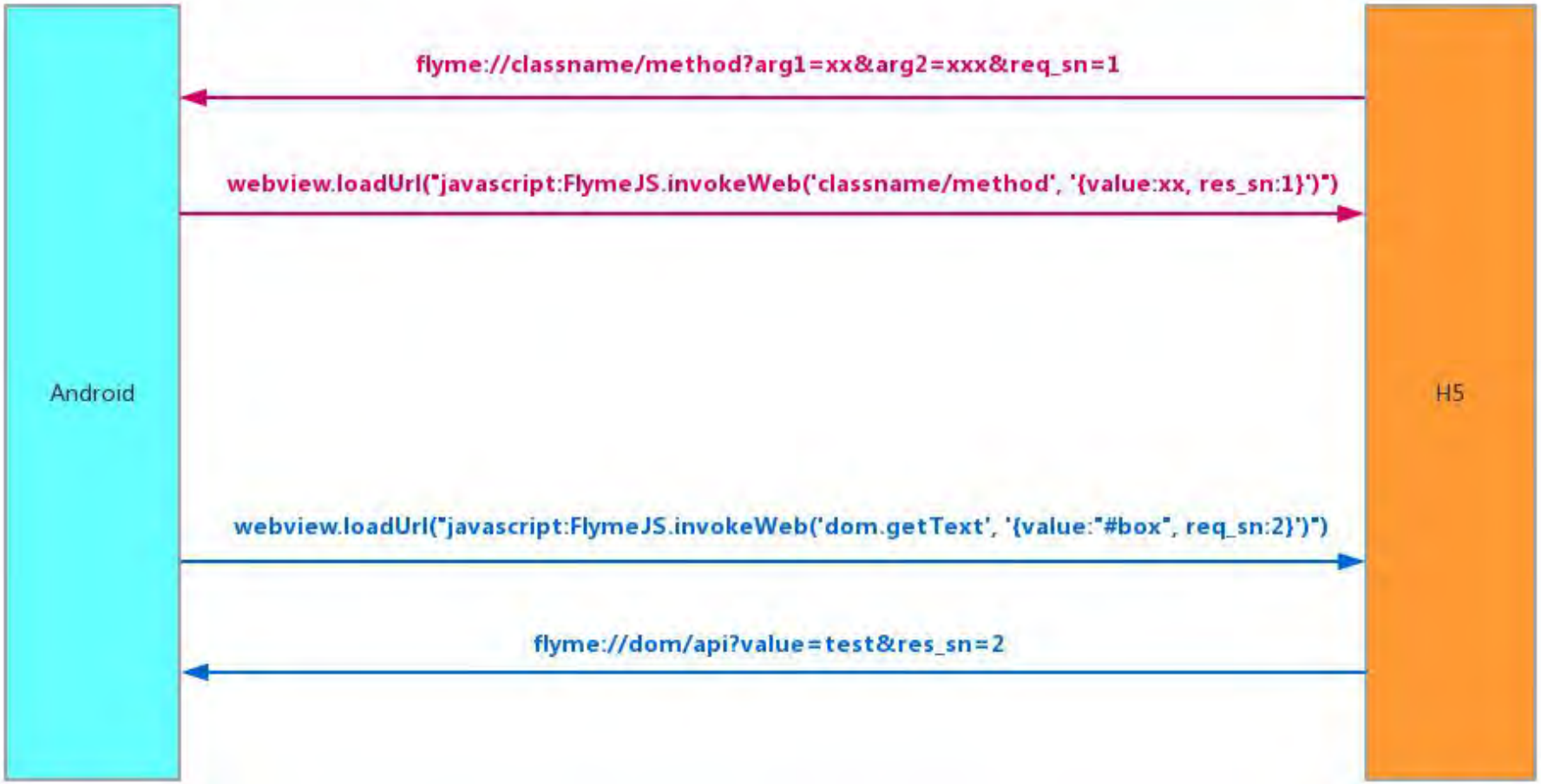
- 在SDK内部封装桥协议约定
- H5的调用封装成以下格式：

```
flyme://[类名]/[方法]?[参数...]
```

- 业务层继承SDK，并根据实际需求扩展Handler
- JS SDK遵循ES6的模块化写法，并提供完善的二次编译打包工具

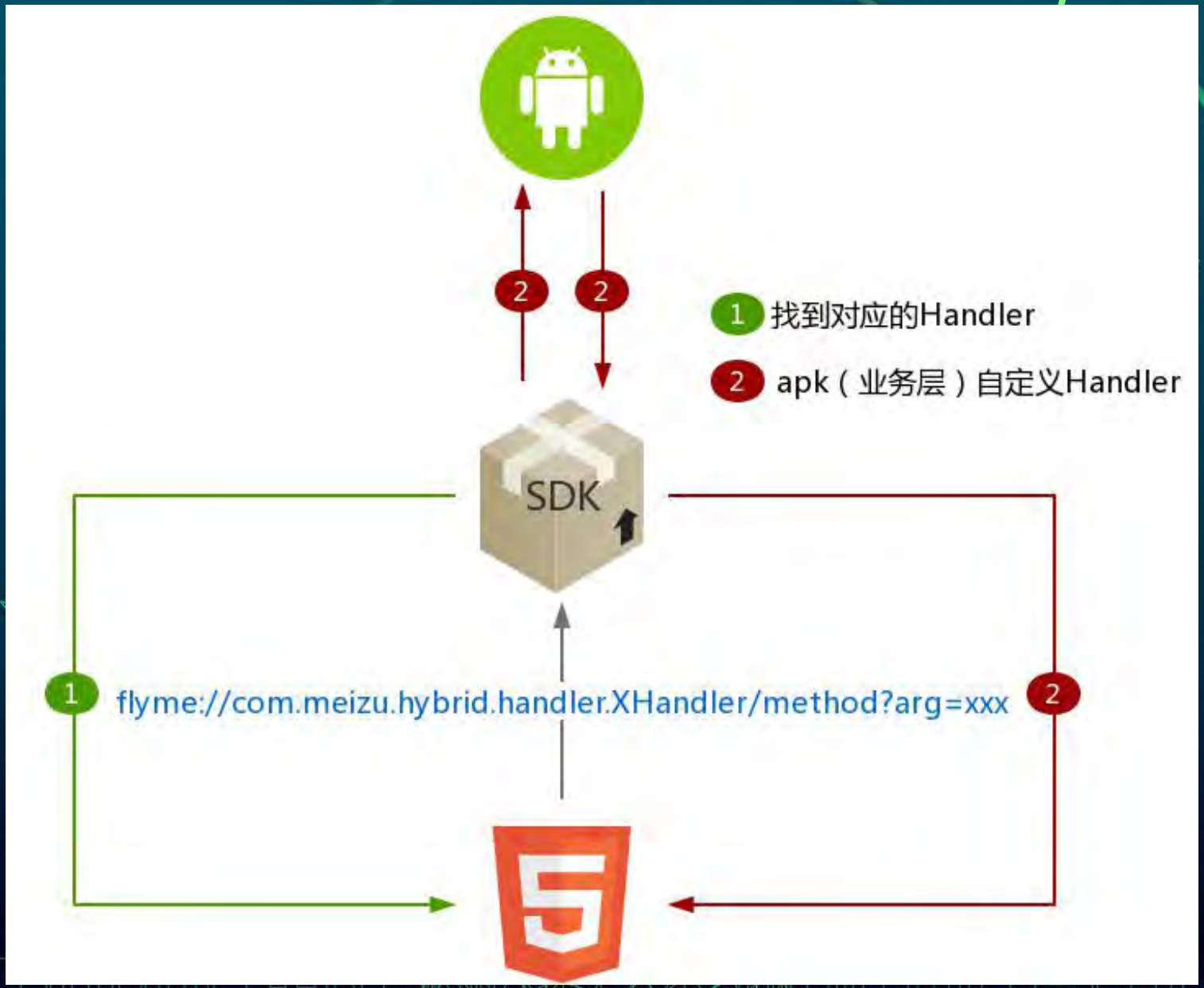
通讯——异步调用

OSC源创会
年终盛典 2016



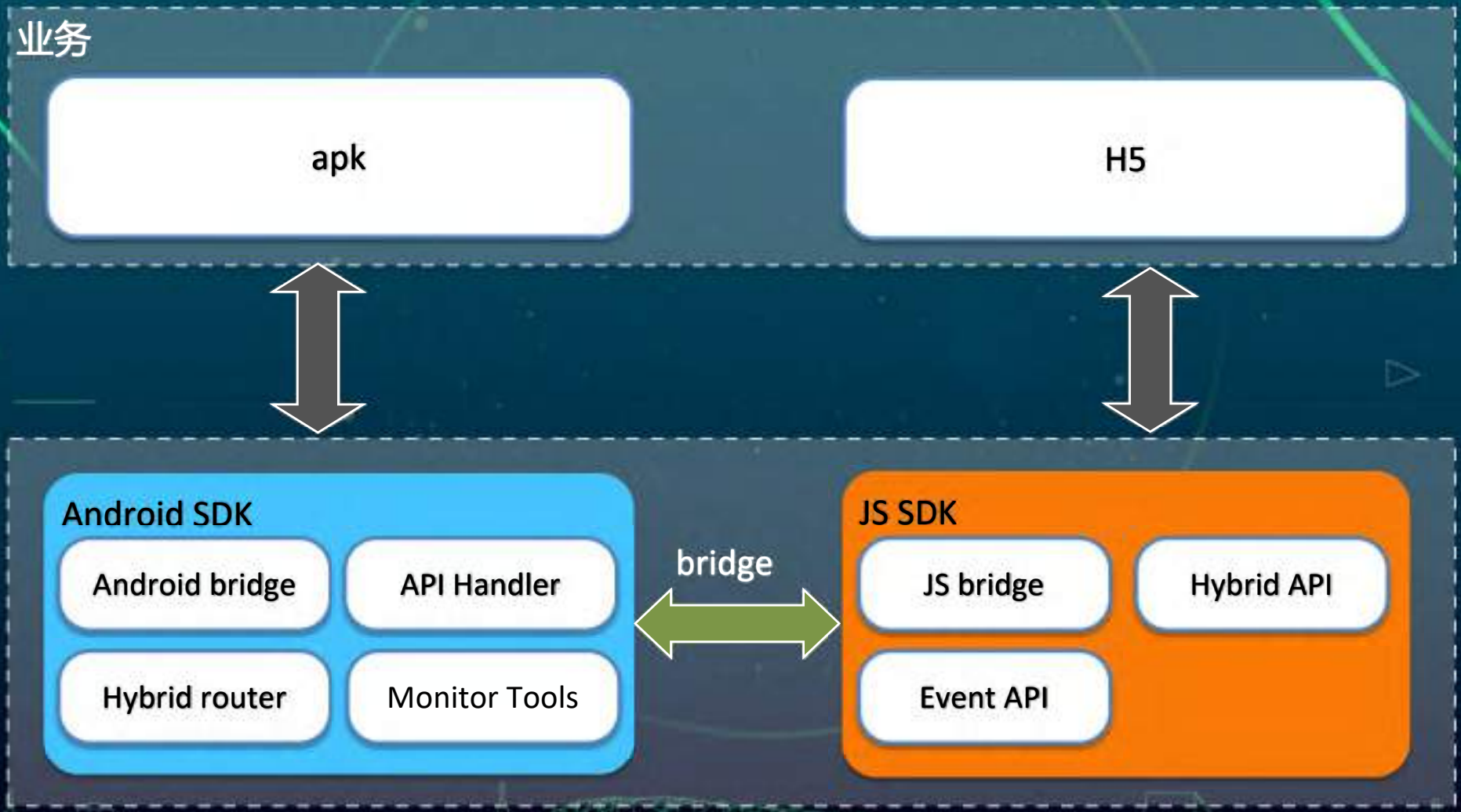
通讯——（回调）调用过程

OSC源创会
年终盛典 2016



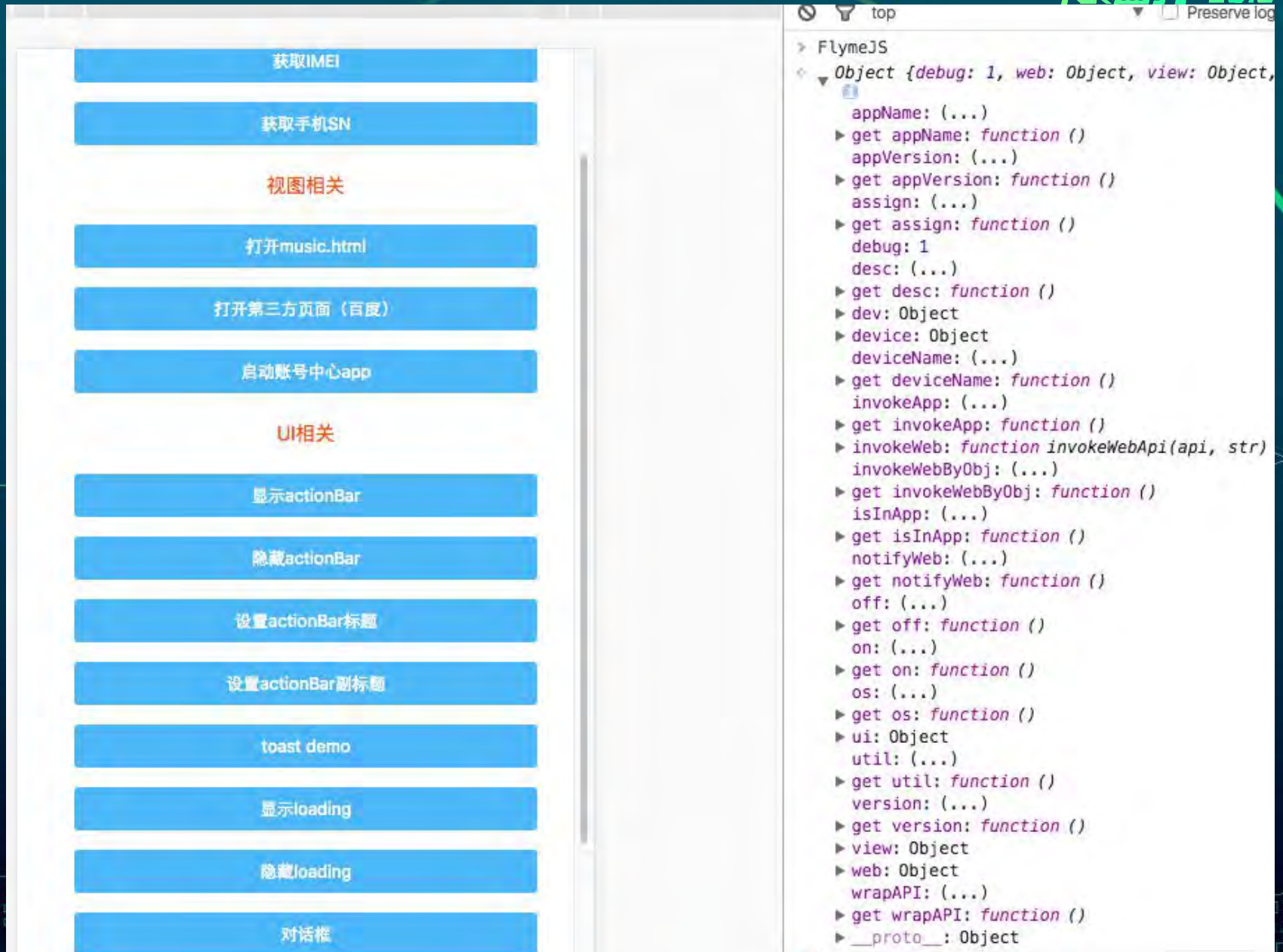
Hybrid SDK

OSC源创会
年终盛典 2016



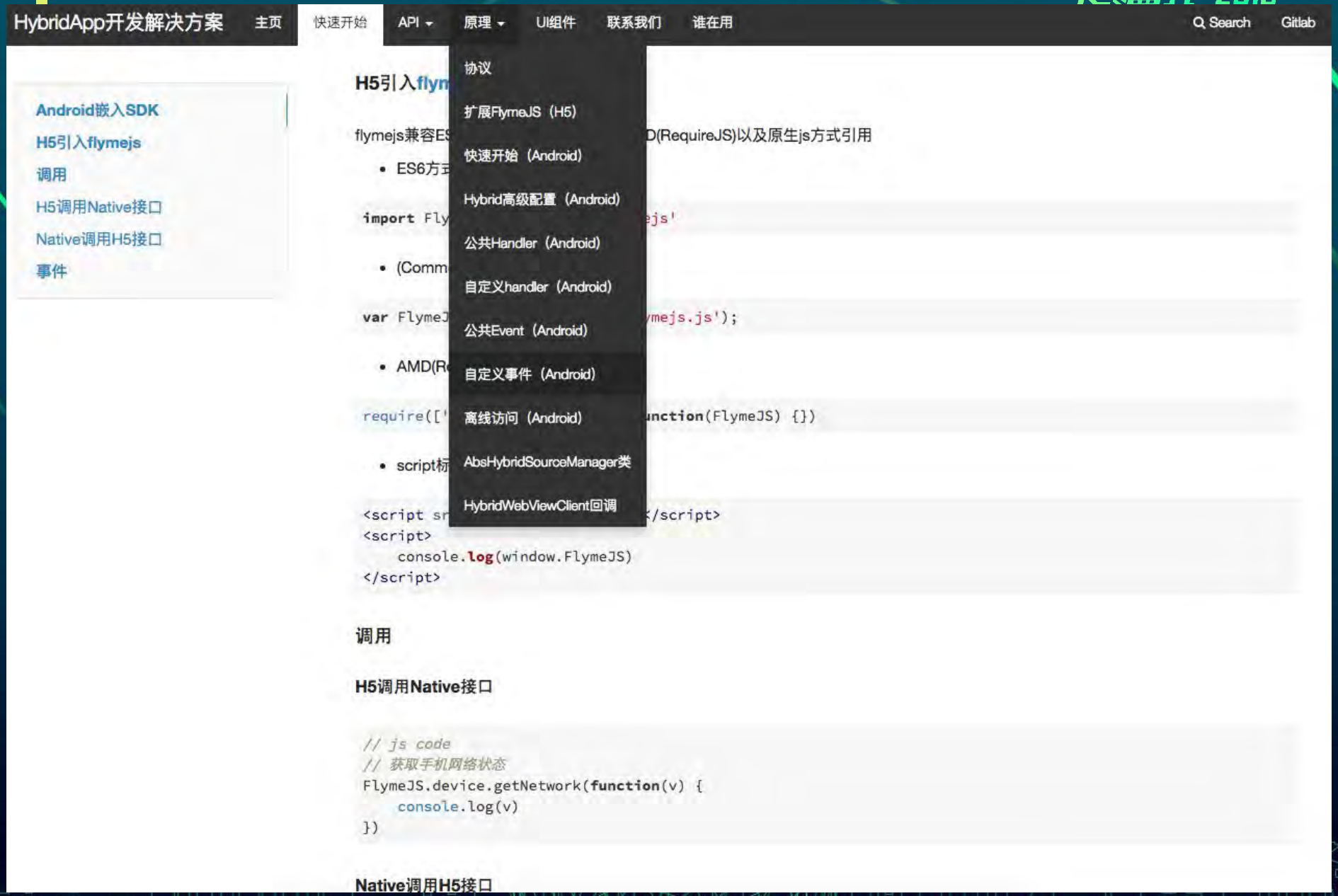
模块化的API设计

OSC源创会
年终盛典 2016

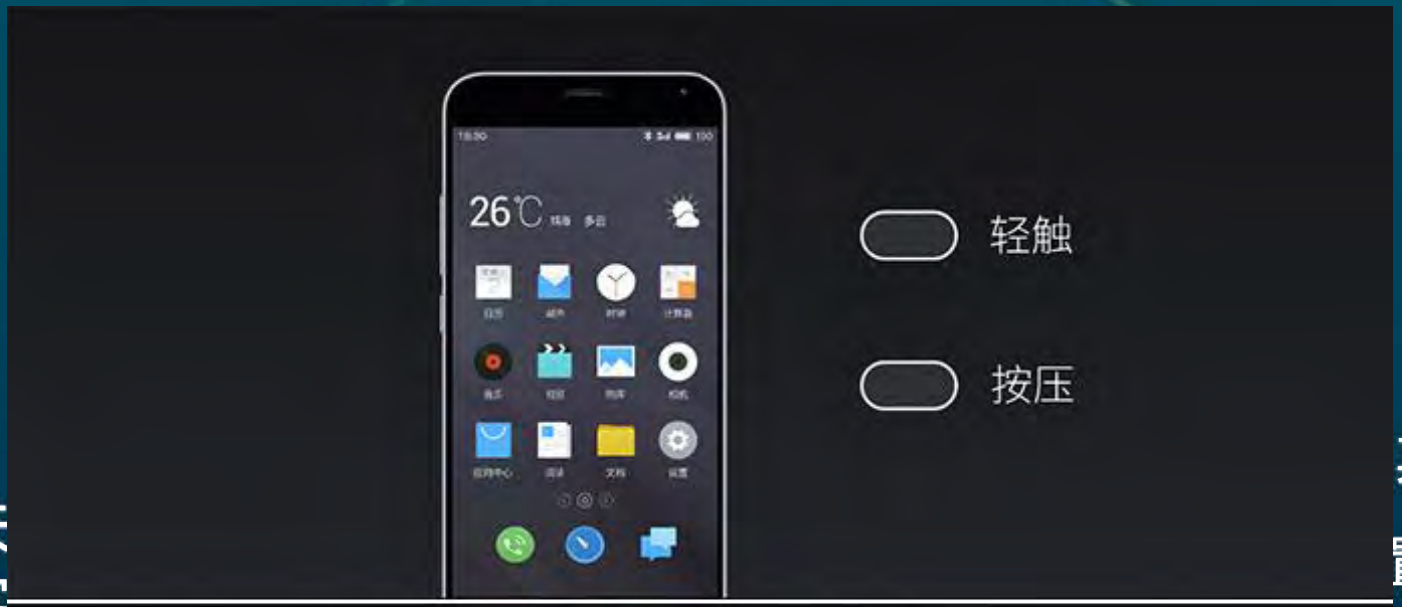


文档和Q&A支持

OSC源创会
年终盛典 2016



H5作为数据接收方



未来未来自客的数据

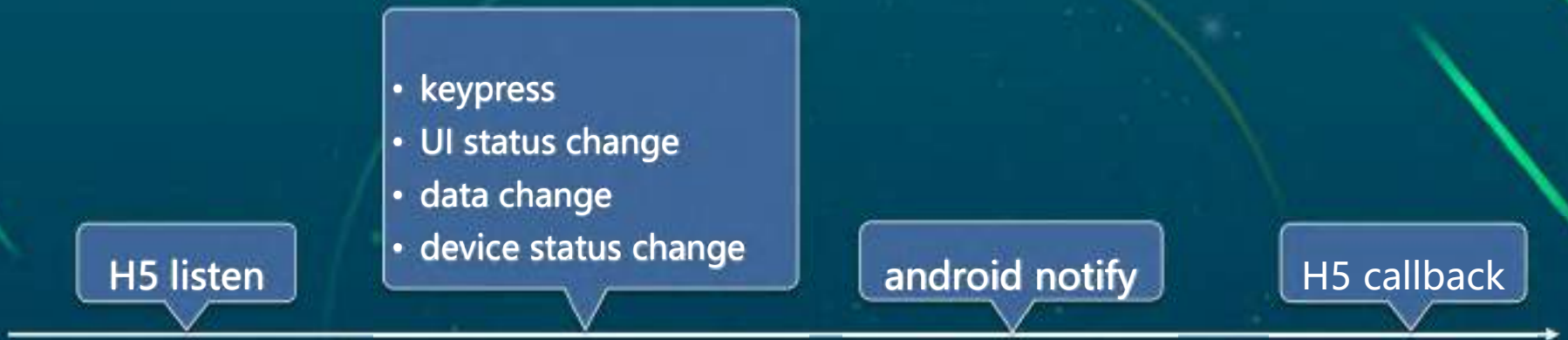
其他按键

置等

状态变更、等

反向通知——事件

OSC源创会
年终盛典 2016



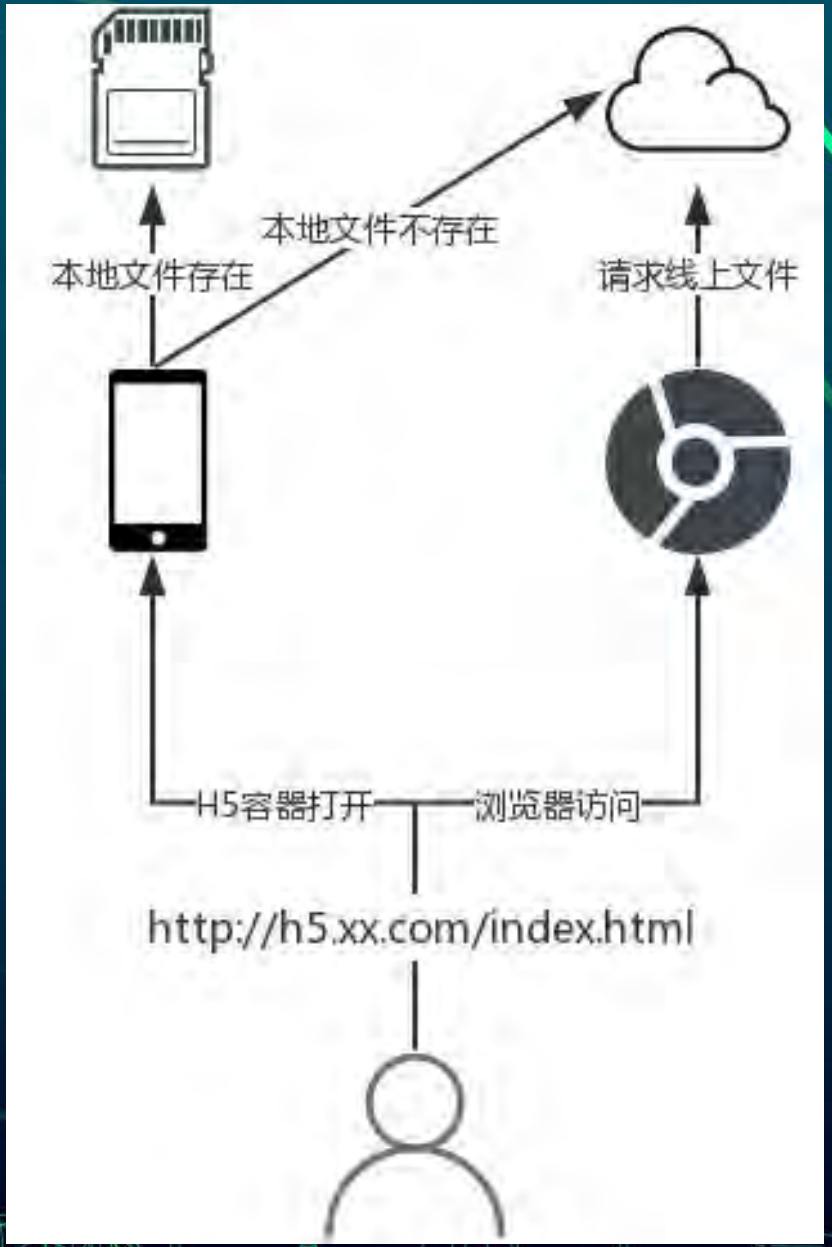

```
// H5拦截mback按键
FlymeJS.on(‘mback’, function mbackHandler(date) {
      console.info(data)
})

// 客户端notify
Webview.loadUrl(“javascript:
      FlymeJS.notify(‘com.meizu.hybrid.event.MBack',
      JSON.stringify({value: ‘xxx’}))
”)
```

H5资源访问

OSC源创会

- 相同的资源使用唯一url定位
- 离线文件不存在或过期则走线上
- 基本覆盖所有H5应用场景



资源定位规则

OSC源创会
年终盛典 2016

- H5资源文件打包路径不受约束，根据正则规则将线上url匹配离线文件
- 客户端实现路径重定向、重写等功能

```
"routes": [
    {
        "regex": "//h5.meizu.com/(.+)",
        "result": "//h5.meizu.com/$1",
        "rule": "replace"
    },
    {
        "regex": "//h(\\d+).meizu.com/(.+)",
        "result": "//h5.meizu.com/$2",
        "rule": "override"
    }
]
```

OSC源创会
年终盛典 2016

几种Web缓存方案对比

方案	原理	优点	缺点
Web缓存	HTTP头	纯Web方案，简单	伪离线 无网不可用
manifest	HTML5新接口	纯Web方案，简单	更新有坑
离线包	请求拦截+本地替换	无网可用，资源易于控制	需要后台、客户端等资源和人力的配合
Cache API	W3C新草案	纯web方案，可实现更细粒度地定制资源	更新依赖Service Worker，两者皆不成熟，webview兼容问题

资源包管理平台设计

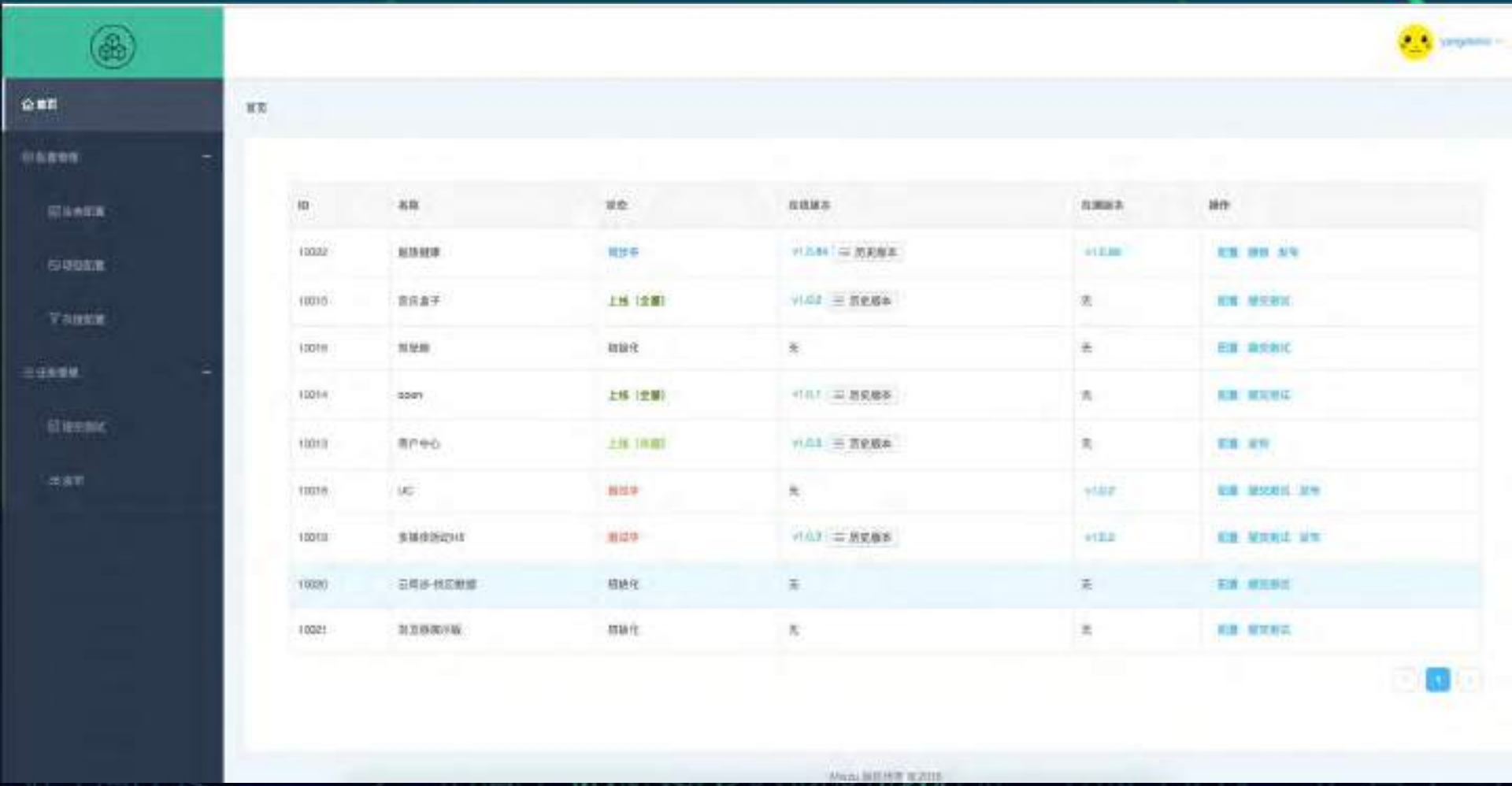
OSC源创会
年终盛典 2016

- 托管H5离线包，前端只需要轻量的包管理
- CI、测试、灰度、发布一体化
- 版本管理、增量对比
- 本地资源包更新检查接口
- 静默更新（服务端主动推送）

资源包管理平台

OSC源创会
年终盛典 2016

轻量的H5离线包托管平台



待发包测试

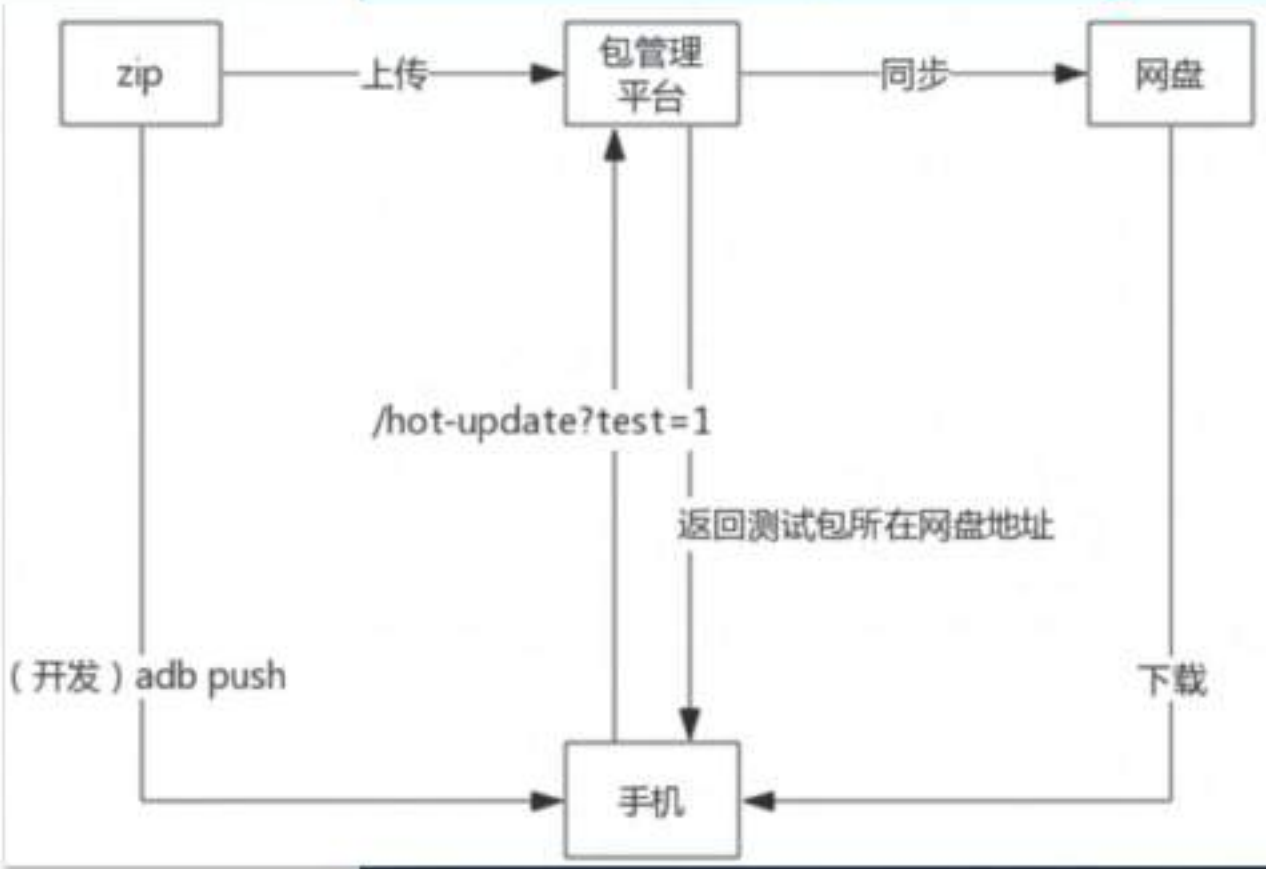
首页 / 提交测试 / 10002

项目: 用户中心
状　态: 测试中
在线版本: 1.0.2
测试版本: 1.0.2
测试包地址: ucdemo.zip

文件列表

h5.meizu.com/uc/a.html
h5.meizu.com/uc/b.html
package.json
uc-res.meizu.com/uc/css/a.min.css
uc-res.meizu.com/uc/css/b.min.css
uc-res.meizu.com/uc/css/common.min.css
uc-res.meizu.com/uc/img/webpack.png
uc-res.meizu.com/uc/js/a.min.js
uc-res.meizu.com/uc/js/b.min.js
uc-res.meizu.com/uc/js/common.min.js
uc-res.meizu.com/uc/js/venders.min.js

重新上传包　同步到CDN

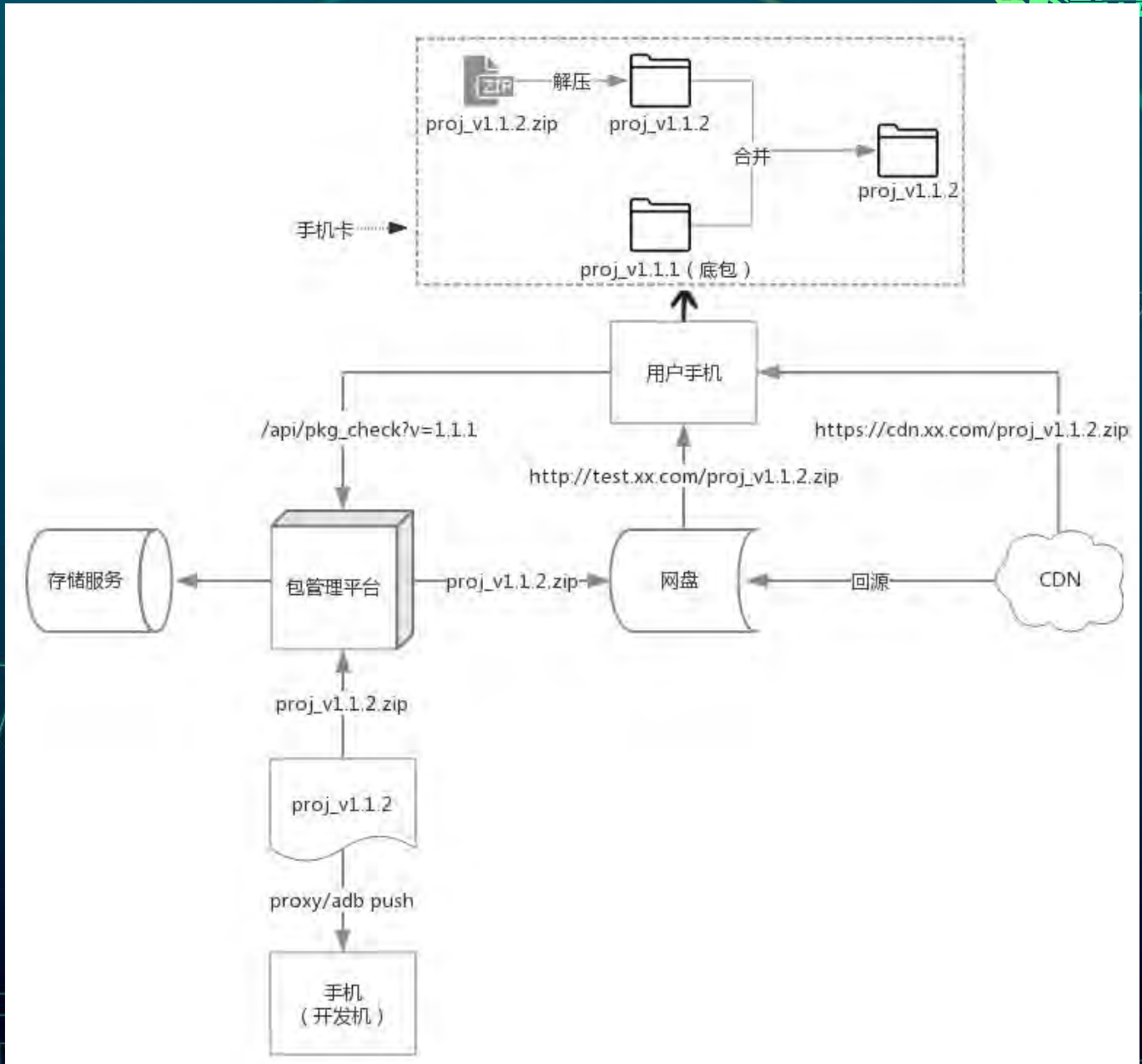


离线包分发控制

多种灰度方式，更精确地控制离线资源更新范围

离线包更新流程

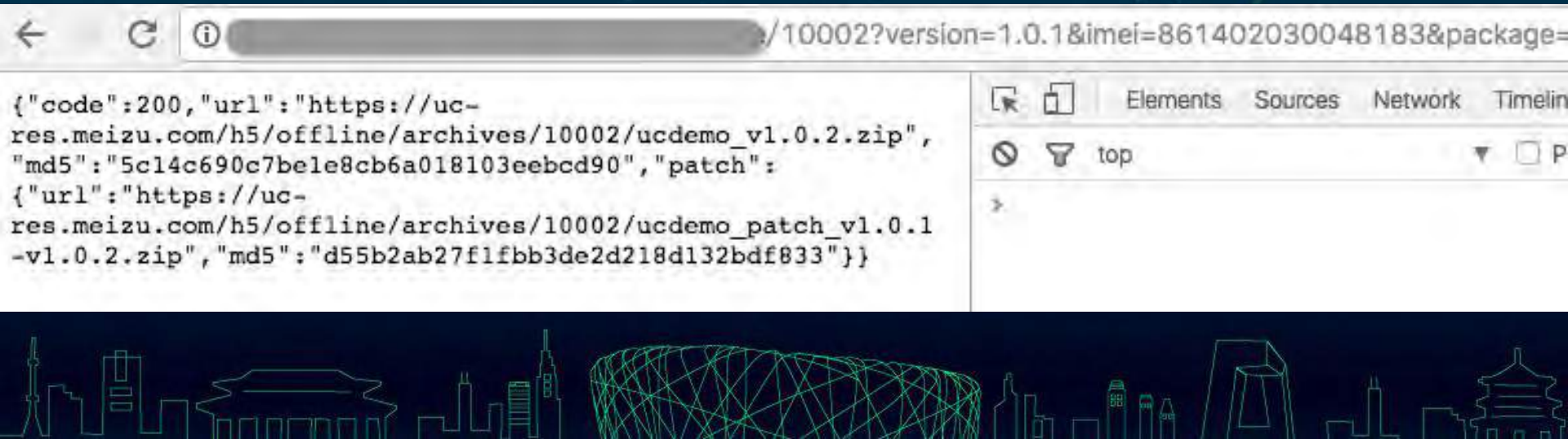
OSC源创会
年终盛典 2016



离线包瘦身——增量包

OSC源创会
年终盛典 2016

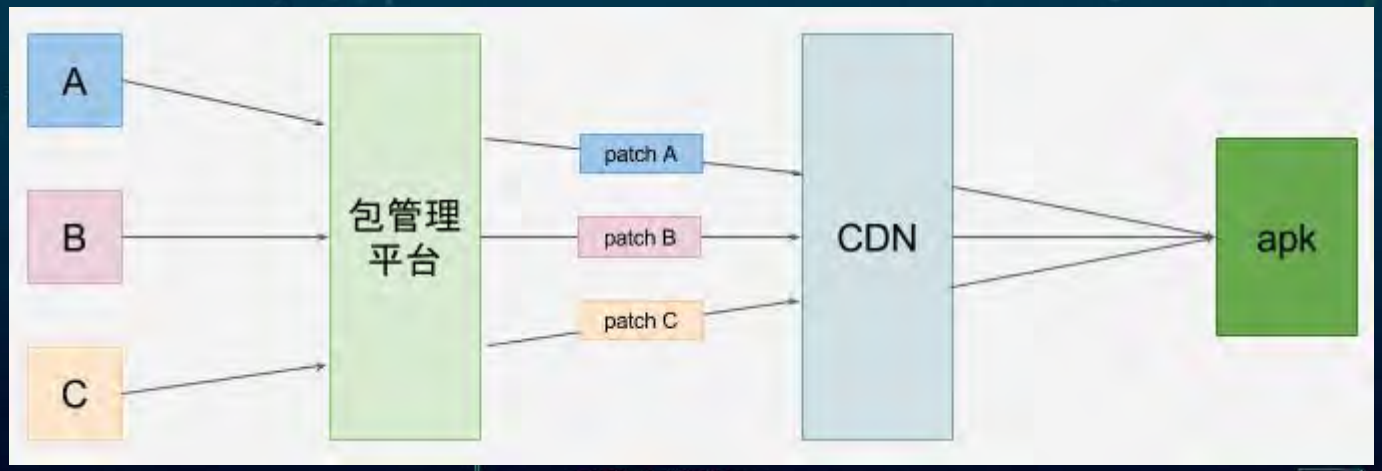
- 上传测试包时同时自动生成增量包
- 更新时同时返回增量和全量包地址
- 增量包在客户端进行hash校验防篡改
- 文件级别的增量diff



离线包瘦身——拆包

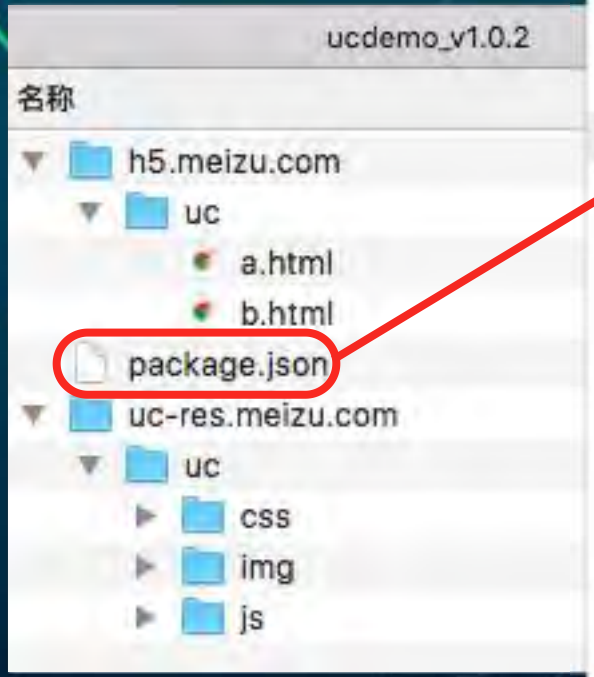
OSC源创会
年终盛典 2016

- 同一个apk内的H5按功能分为多个模块
- 模块之间的更新频率不同
- 生成离线包时只打包改动的模块进行更新
- 在客户端合并多包



离线包规范

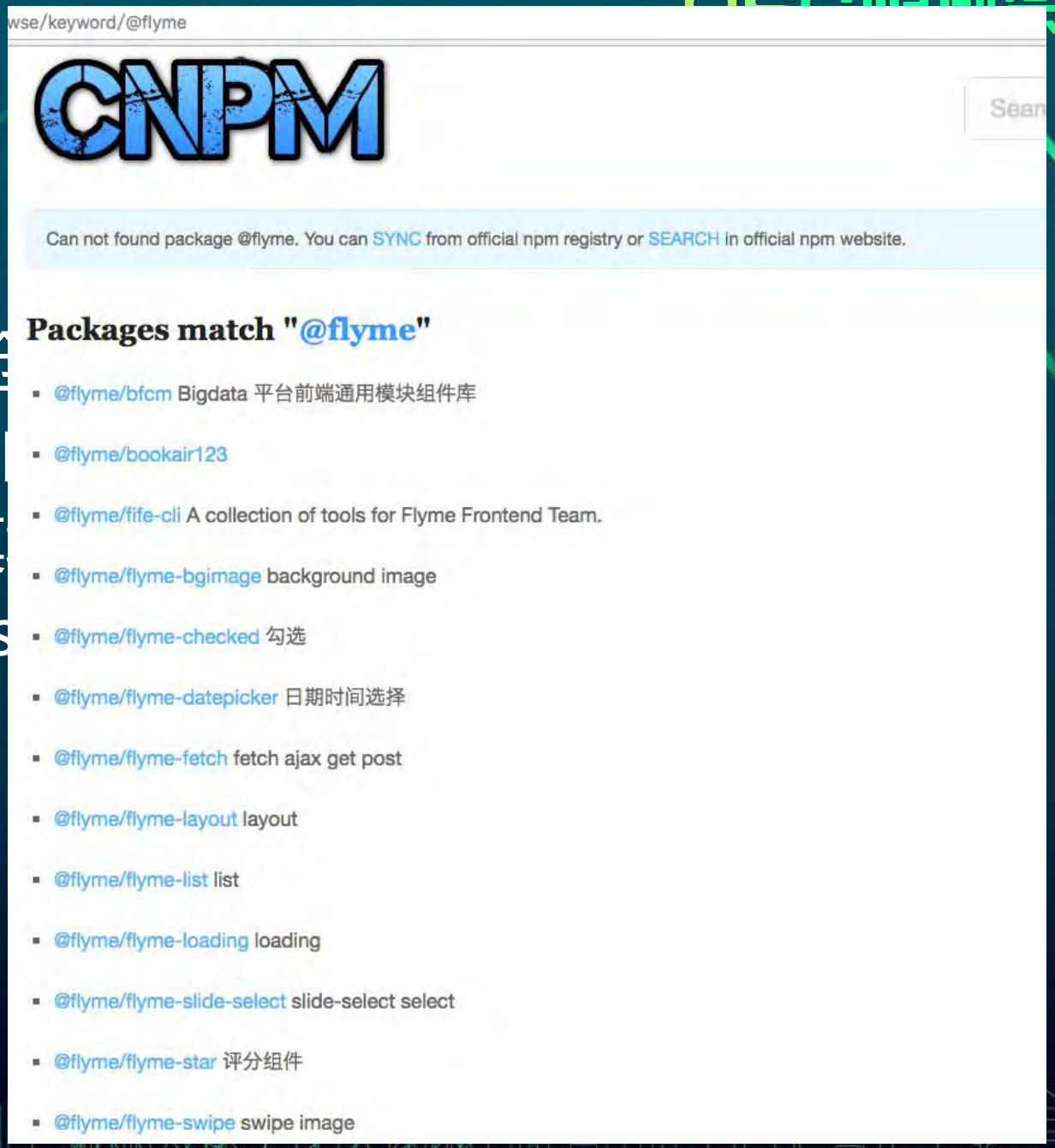
OSC源创会
年终盛典 2016




```
{
    "apk": "com.meizu.usercenter", // apk名
    "id": 10002, // 包id
    "moduleId": 0, // apk内的模块序号
    "version": "1.0.2", // 包版本号
    "routes": [ // H5资源访问规则
        {
            "regex": "//h5.meizu.com/(.+)",
            "result": "//h5.meizu.com/$1",
            "rule": "replace"
        },
        {
            "regex": "//h(\\d+).meizu.com/(.+)",
            "result": "//h5.meizu.com/$2",
            "rule": "override"
        },
        {
            "regex": "//abc.meizu.com/xxx/([^.]+)?.jsp?(.*)",
            "result": "//abc.meizu.com/pages/$1.html?$2",
            "rule": "replace"
        }
    ]
}
```

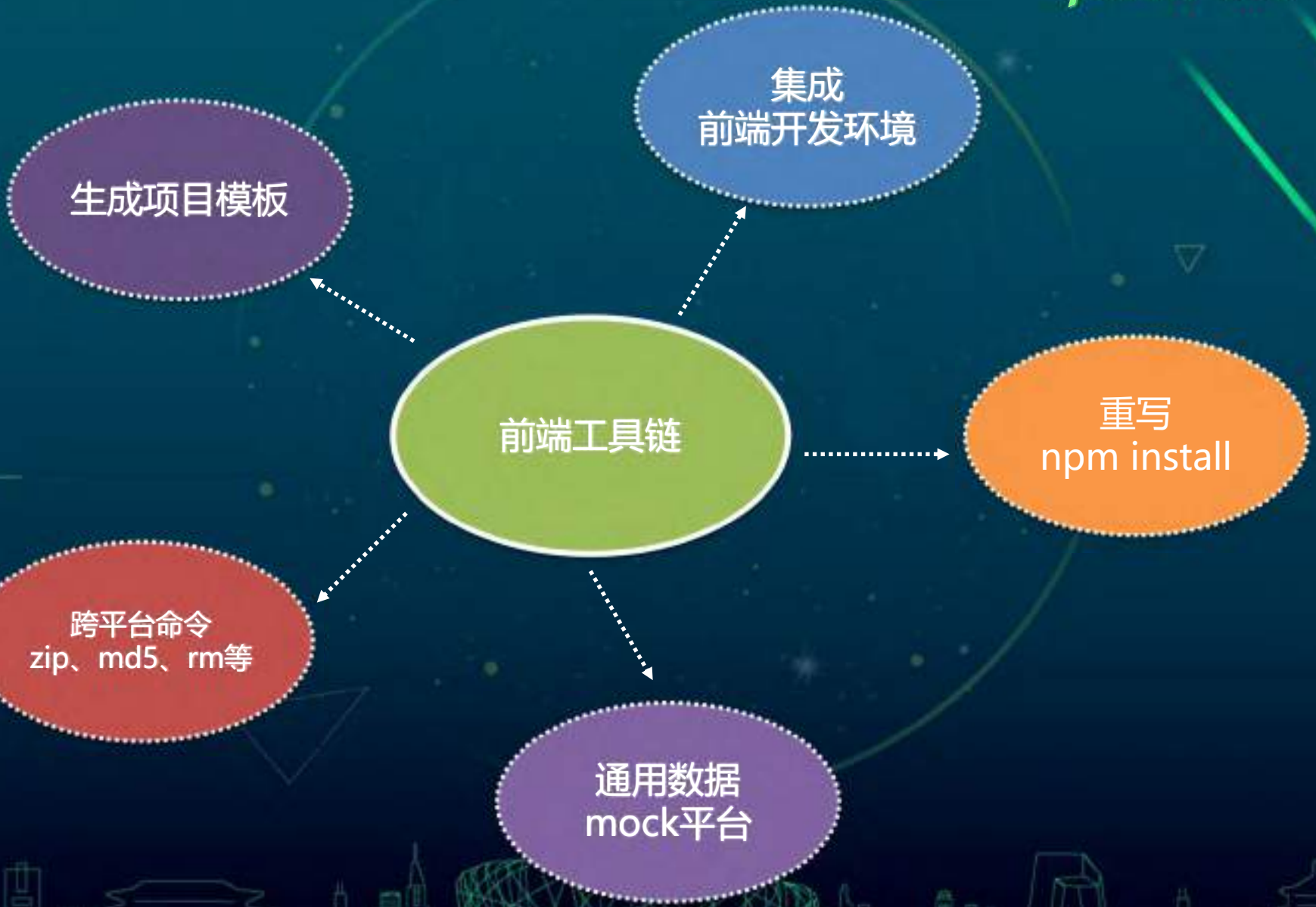
前端组件生态

- 私有npm仓
- 基于React
- 遵循ES6模
- 组件publis



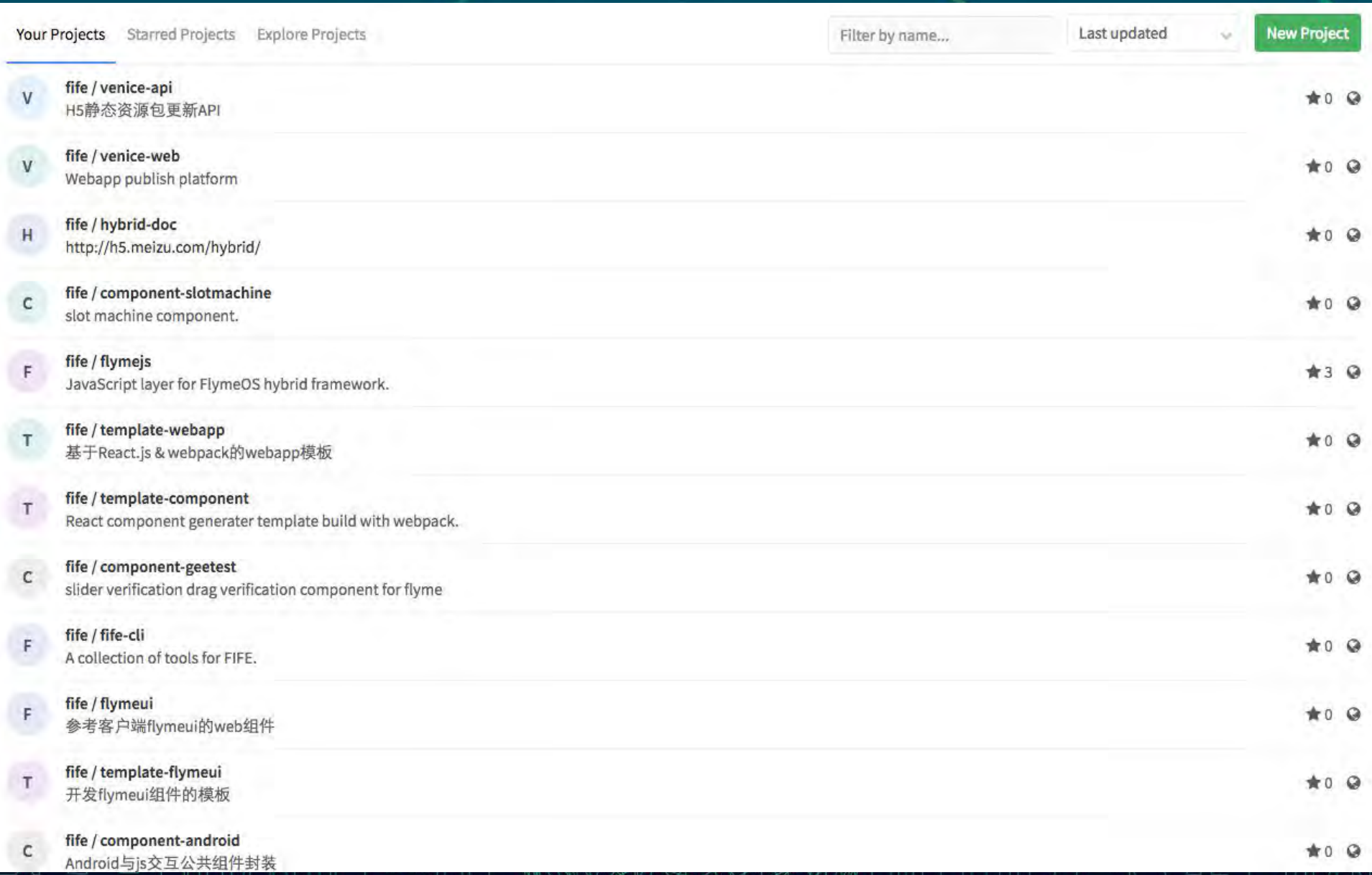
工具

OSC源创会
年终盛典 2016



逐渐完善的工具链

OSC源创会
年终盛典 2016



Flyme Hybrid 体系

OSC源创会
年终盛典 2016

（类）Hybrid开发方案对比

OSC源创会
年终盛典 2016

方案	原理	优点	缺点	适用范围
ReactNative/ Weex	js书写 原生代码运行	高性能、跨平台	接入门槛高	大公司、 创业团队
Hybrid	定制且易扩展的 webview	方案通用、 稳定、技术细节 容易把控	技术实现不通用 开发成本高 性能不够好	大中型团队
Ionic/Cordova	高度定制的 webview	丰富的api和完善的工具、插件， 纯web方案	高度定制导致扩展性差， 性能差，依赖官方后台 实现热更	个人开发者

Q&A

OSC源创会
年终盛典 2016

谢谢！