



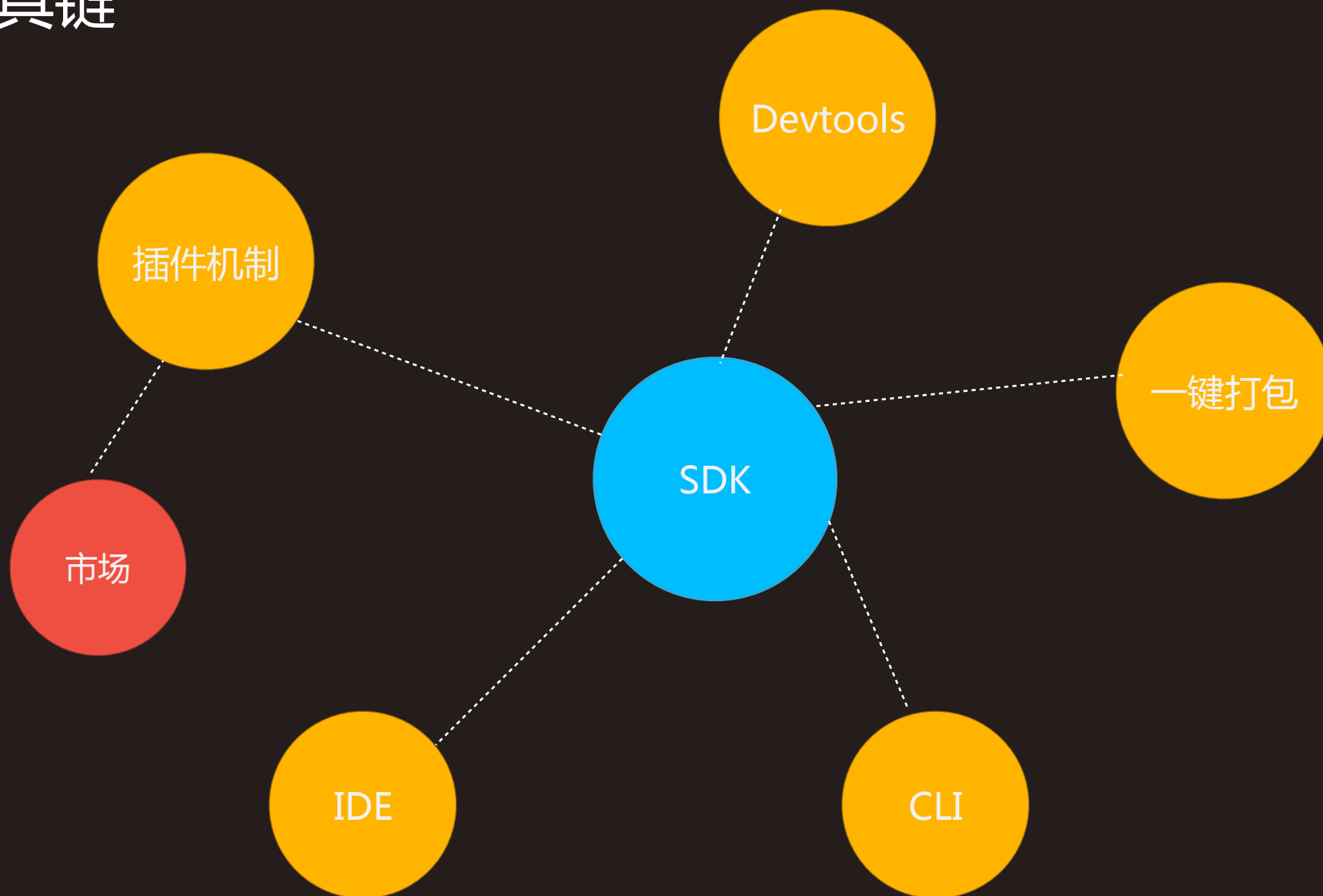
WEEEX Conf

畅爽开发的保证 —— 探秘Weex工具链

Weexpack与插件机制

Weex开发中的痛点

Weex工具链



Weex SDK支持的扩展方式

Module

非UI的特定功能

例如 sendHttp、openURL 等

Component

Native控件

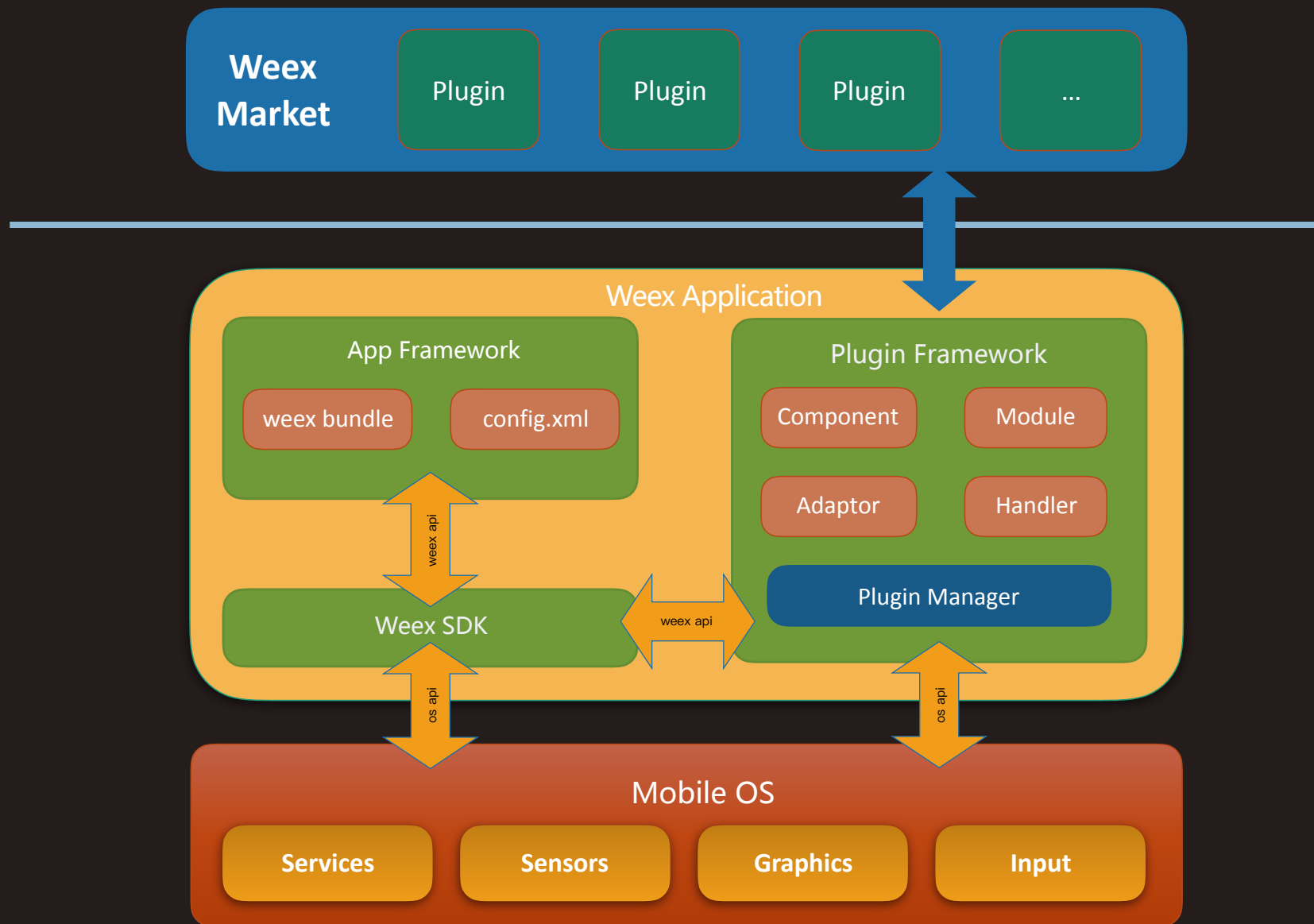
例如 RichText、DatePicker 等

Adapter / Handler

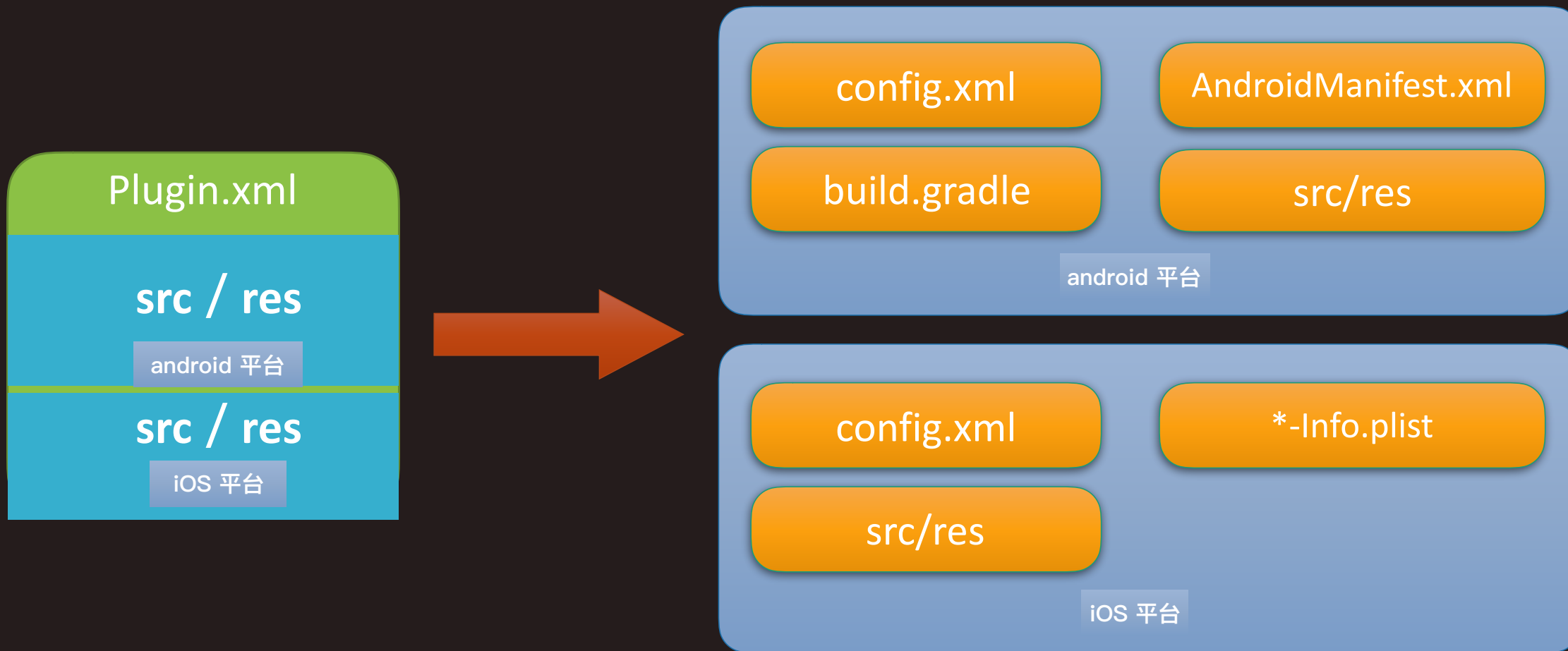
实现某些sdk接口

例如图片加载器等

插件机制

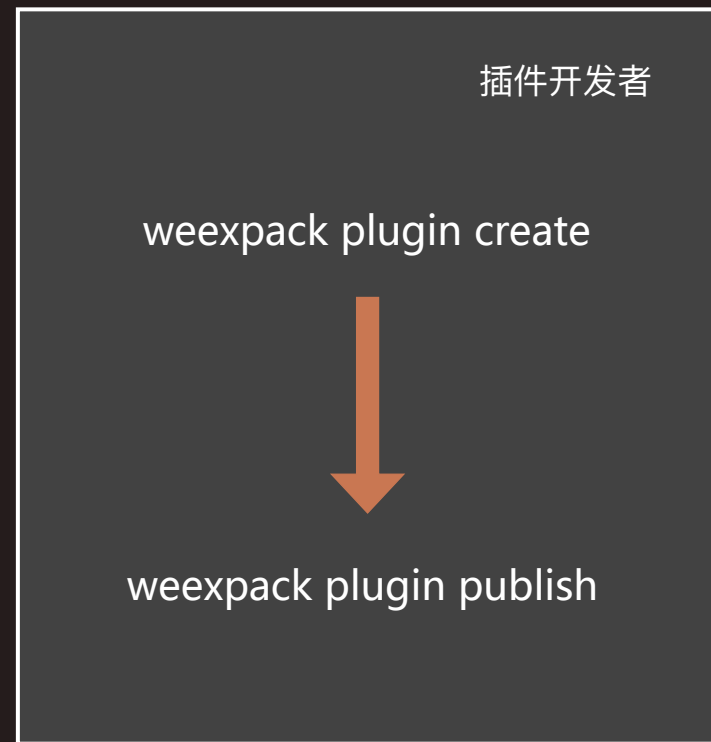
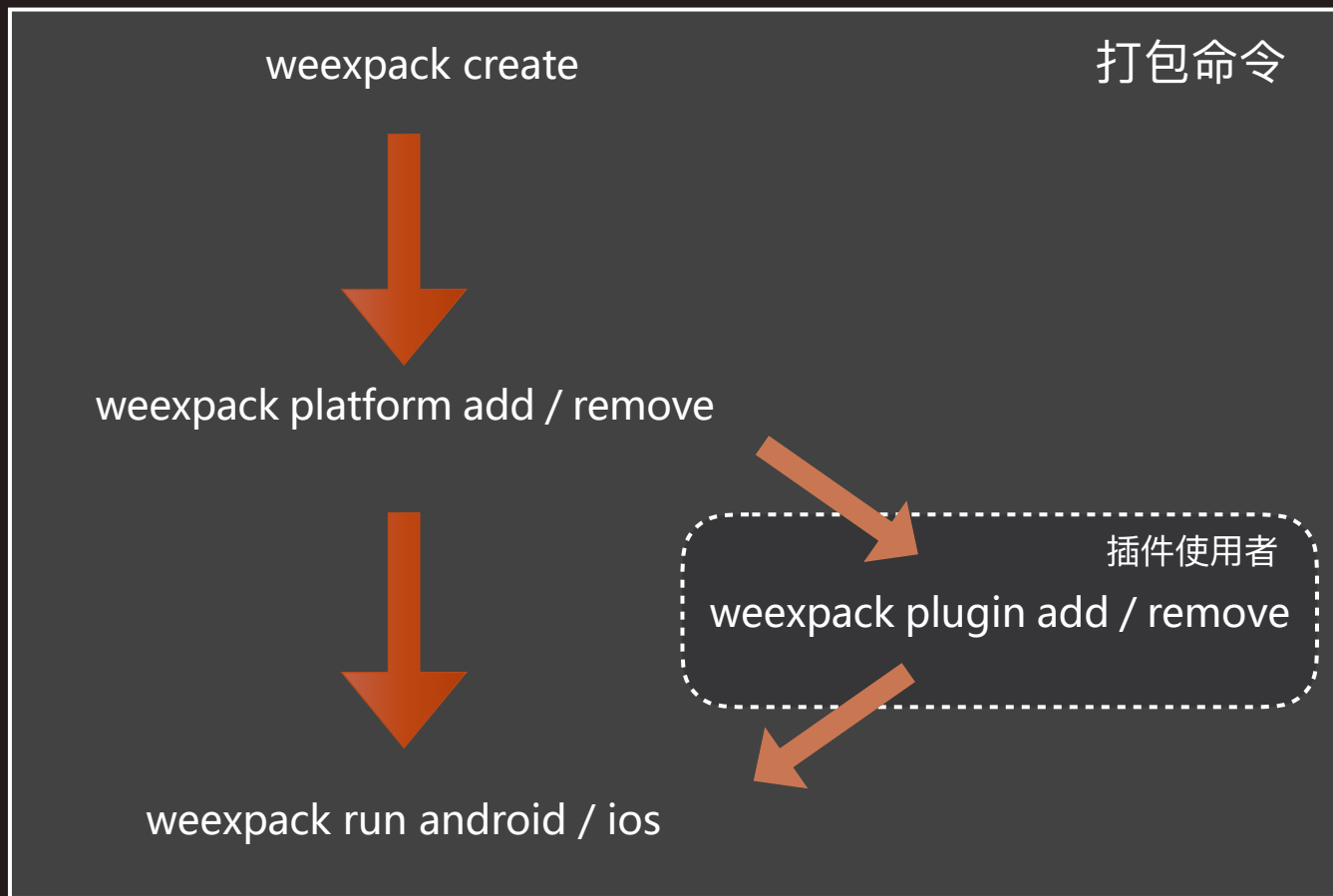


插件到插件容器的映射



打包及插件相关命令

weexpack命令行工具 <https://github.com/weex-team/weex-pack>



插件的安装使用

借助weexpack命令能完成安装使用插件的所有环节

create

创建weex工程

创建一些weex工程必需的配置文件及目录

platform add / remove

添加支持的平台

当前支持android和ios两个平台，平台的模版可以自定义，支持从本地、github、npm添加

plugin add / remove

添加需要的插件

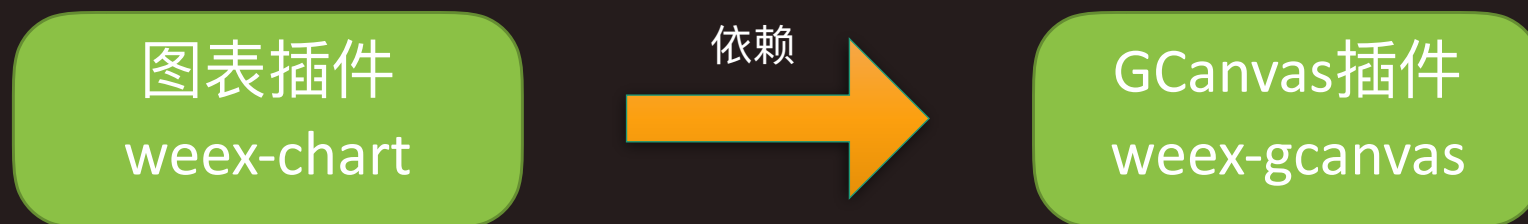
支持从市场、本地、github下载安装插件及其依赖包

build weexplugin library

编译构建插件容器

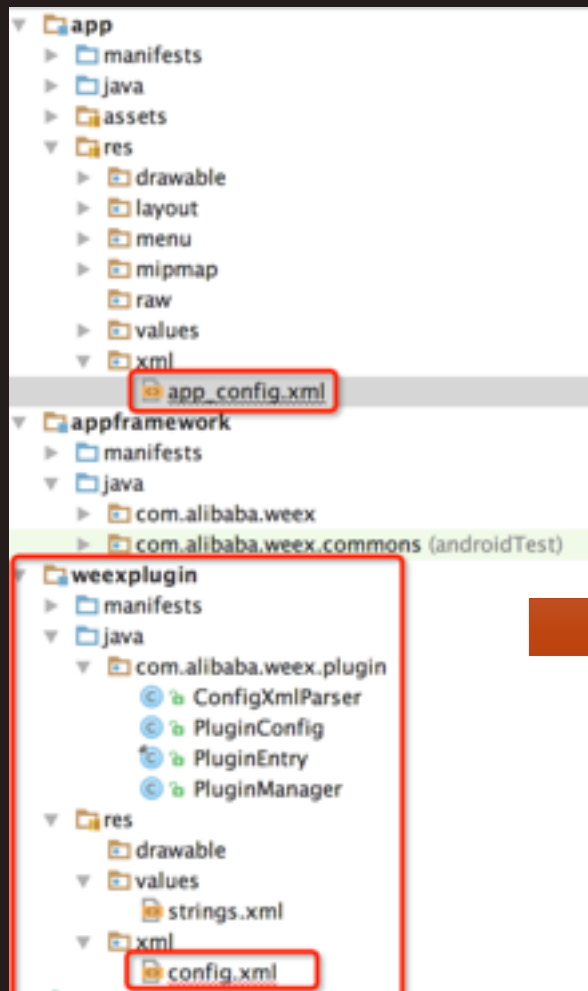
所有的插件都被安装在一个可被称为插件容器的目标工程中，其可被编译为一个二进制的依赖库，在你的App打包时引用该二进制包即可

打包及插件机制应用示例



1. weexpack create MyApp // 创建 weexpack工程
2. cd MyApp && npm install // 切换到工程目录下安装依赖
3. weexpack platform add android // 添加android平台模版
4. weexpack plugin add weex-chart // 安装weex-chart插件
5. cp -ri ../weex-chart-examples/* src // 在src目录下编写we代码
6. weexpack run android // 连接设备，打包运行

打包及插件机制应用示例——平台模版安装



打包及插件机制应用示例——插件安装

```
<plugin id="weex-gcanvas"
  version="0.1.6">

  <name>weex-gcanvas</name>
  <description></description>
  <license>Apache 2.0</license>
  <keywords>weex,canvas</keywords>
  <repo>https://*.git</repo>
  <issue>https://*</issue>

  <platform name="android">

    <config-file target="config.xml" parent="/*">
      <feature name="WXGCanvasModule">
        <param name="category" value="module"/>
        <param name="android-package" value="com.taobao.weex.ui.module.GcanvasModule"/>
        <param name="api" value="gcanvas"/>
      </feature>
    </config-file>

    <config-file target="config.xml" parent="/*">
      <feature name="WXGCanvasComponent">
        <param name="category" value="component"/>
        <param name="android-package" value="com.alibaba.weex.extend.component.WXGCanvasComponent"/>
        <param name="api" value="gcanvas"/>
      </feature>
    </config-file>

    <source-file src="native/android/WXGCanvasGLSurfaceView.java"
      target-dir="src/com/alibaba/weex/extend/component"/>
    <source-file src="native/android/WXGCanvasComponent.java" target-dir="src/com/alibaba/weex/extend/component"/>
    <source-file src="native/android/GcanvasModule.java" target-dir="src/com/taobao/weex/ui/module"/>

    <framework src="com.taobao.android:gcanvas_library_weex:1.0.1-SNAPSHOT@aar"/>

  </platform>
</plugin>
```

plugin.xml

```
<feature name="WXGCanvasModule">
  <param name="category" value="module" />
  <param name="android-package" value="com.taobao.weex.ui.module.GcanvasModule" />
  <param name="api" value="gcanvas" />
</feature>
<feature name="WXGCanvasComponent">
  <param name="category" value="component" />
  <param name="android-package" value="com.alibaba.weex.extend.component.WXGCanvasComponent" />
  <param name="api" value="gcanvas" />
</feature>
```

config.xml

```
dependencies {
  ... compile fileTree(include: ['*.jar'], dir: 'libs')
  ... androidTestCompile 'junit:junit:4.12'
  ... androidTestCompile 'com.android.support.test.espresso:espresso-core:2.2.2'
  ... androidTestCompile 'com.android.support.test:runner:0.5'
  ... /*source dependency*/
  ... // compile project(':weex_sdk')
  ... compile 'com.taobao.android:weex_sdk:0.8.0'
  ... compile project(':appframework')
  ... compile 'com.squareup.okhttp:okhttp:2.3.0'
  ... compile 'com.squareup.okhttp:okhttp-ws:2.3.0'
  ... compile 'com.squareup.okio:okio:1.0.1'
  ... compile 'com.alibaba:fastjson:1.1.46.android'
  ... compile 'com.android.support:support-v4:23.1.1'
  ... compile 'com.android.support:appcompat-v7:23.1.1'
  ... compile 'com.android.support:design:23.1.1'
  ... compile 'com.android.support:support-annotations:23.1.1'
  ... compile 'com.taobao.android:weex_inspector:0.0.8.5'
  ... compile 'com.journeyapps:zxing-android-embedded:3.4.0'

  ... // PLUGIN GRADLE EXTENSIONS START
  ... compile 'com.taobao.android:gcanvas_library_weex:1.0.1-SNAPSHOT@aar'
  ... // PLUGIN GRADLE EXTENSIONS END
}
```

build.gradle

插件机制的优点

1

按需打包

插件可按需灵活配置打包，app完全自主。

2

独立升级

插件可独立升级，支持细粒度的版本控制。

3

支持二次开发

插件可以二进制库或者源码形式发布，方便二次开发。

4

流程自动化

提供完善的命令行工具，自动合并代码、资源及编译脚本。

5

面向定制

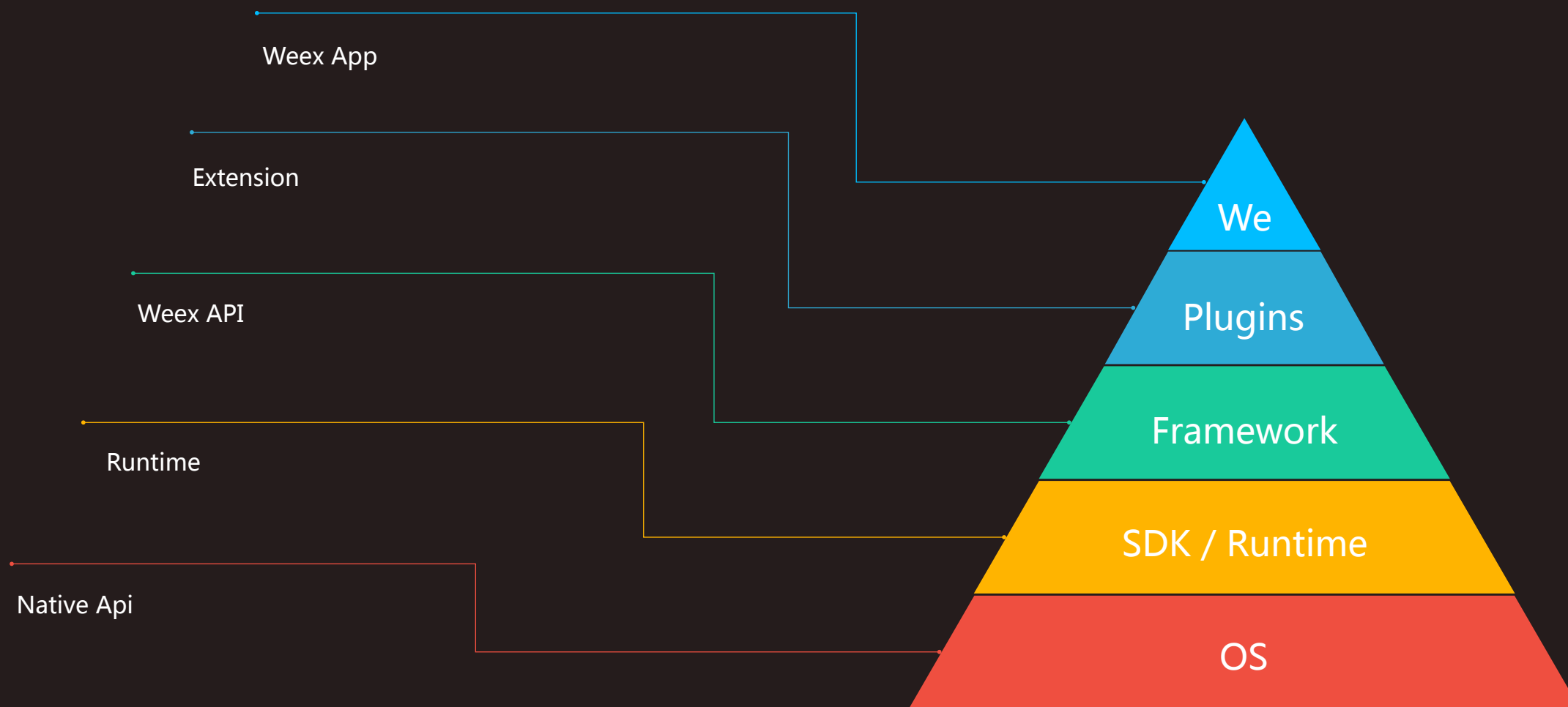
可按需定制自己的插件仓库及平台模版。

6

集成一键打包


插件工具和打包工具彻底打通，降低开发app的门槛。

展望——插件化的Weex App



插件市场

插件市场是weex各种扩展的信息聚合点，在市场里可以自由的搜索、下载、发布插件和寻求合作者共同开发。



Weex第三方组件市场

weexpack Weex pack tool 0.2.3
webpack Packs CommonJs/AMD modules for ... 2.1.0-beta.28
weex-plugin--weex-gcanvas 0.1.3
weex

Weex Engineering Development Kit

[中文版文档](#)

Weexpack introduction

Weexpack is our next generation of engineering development kits. It allows developers to create weex projects with simple commands and run the project on different development platforms.

pre-environmental requirements

- supported on Mac linux windows.



WEEEX Conf

畅爽开发的保证--探秘Weex工具链

调试工具Devtools篇

Weex的调试需求

换位思考-开发者在想什么？

我需要断点调试js，最好能直接在we文件里打断点

能不能看到native的各种网络加载信息，如加载bundle的信息和时间

开发者

我想查看dom层级结构(前端)/native view 层级结构(native端)

其他问题

Weex Devtool主要功能一览

Debugger

远程调试weex源码，可呈现原始的we代码结构并打断点。

展示渲染节点的层级结构。可随时切换dom tree和native view tree。
可以高亮某一个节点

Inspector

Debugger

extension

Others

多实例调试
Screencast
remote control
hot-reload
profiling

chrome devtool 的extension
display as design

extension

Weex Devtool 探秘

远程调试

1

抽象问题

远程运行时与RPC

2

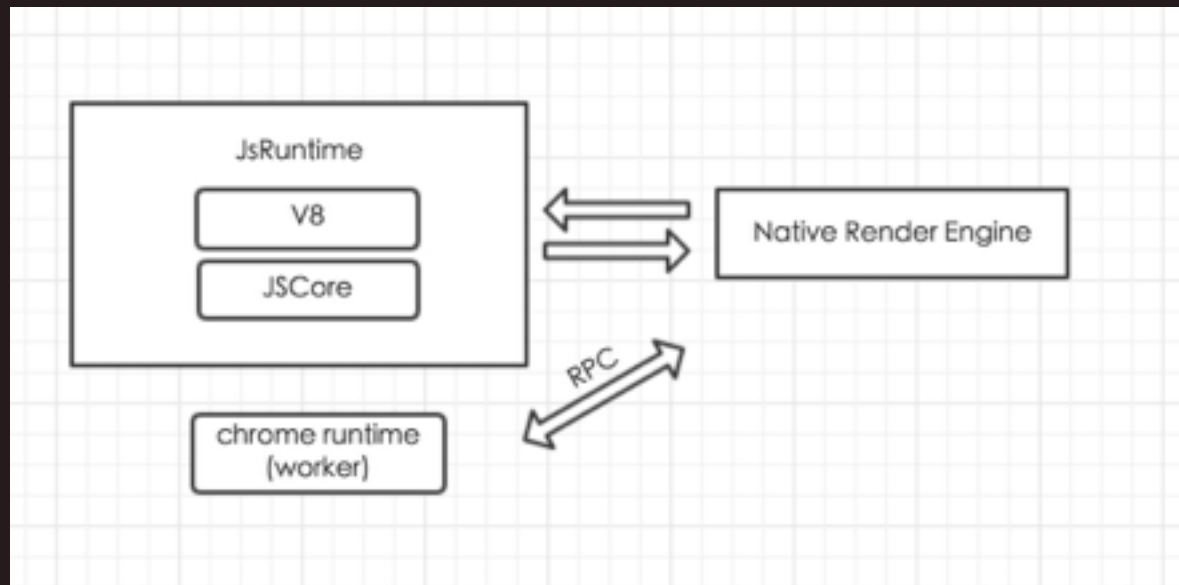
解决问题

运行时环境哪家强

3

优化问题

Bundle加载与Sourcemap



Weex Devtool

为我所用吧Chrome Inspector

- Chrome Inspector
- Remote Debugger Protocol
- A extension Why ?

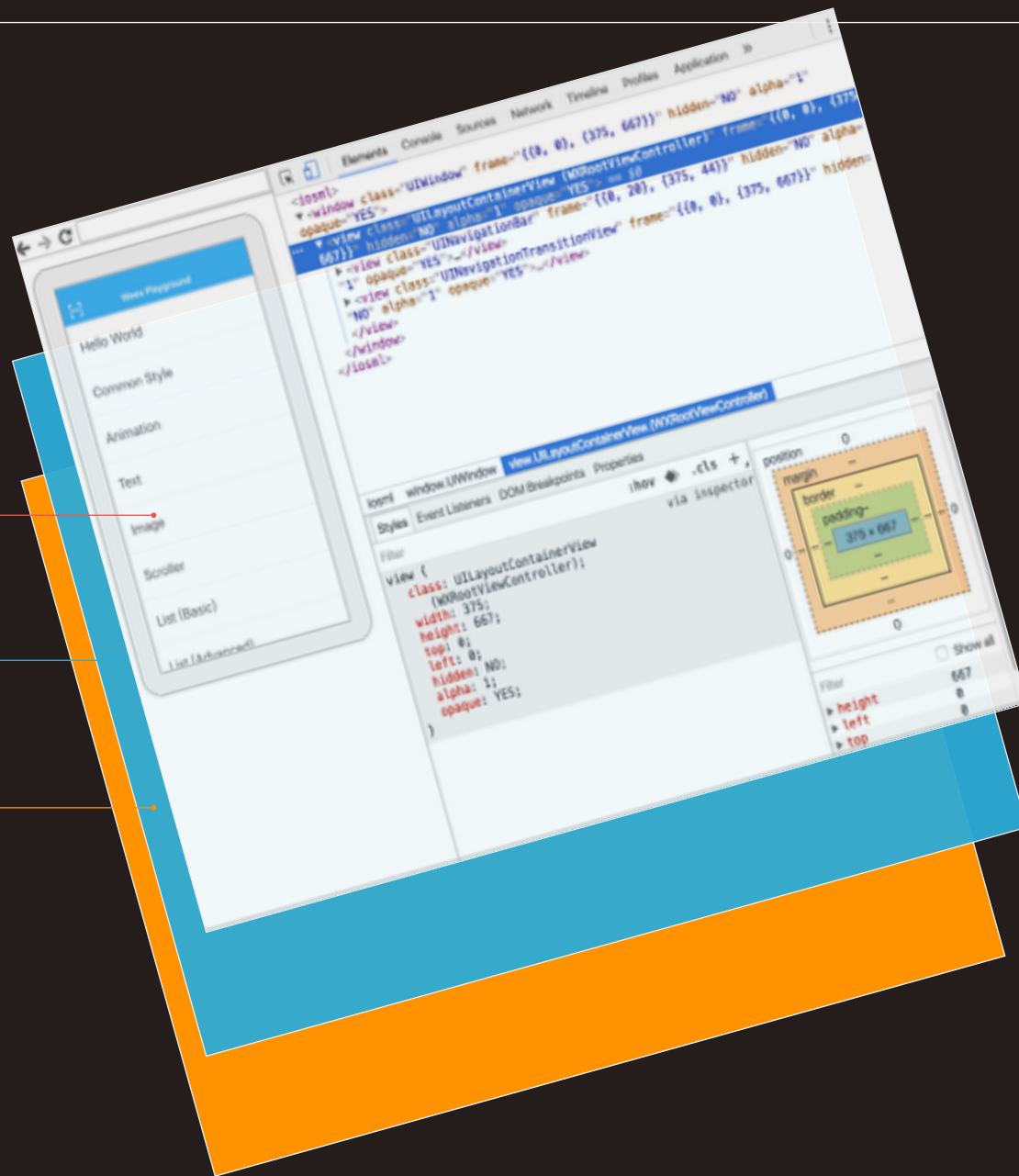
Weex Devtool 探秘

整体架构

Devtool FrontEnd(Runtime)

Node Server

Native Inspector



Weex Devtool

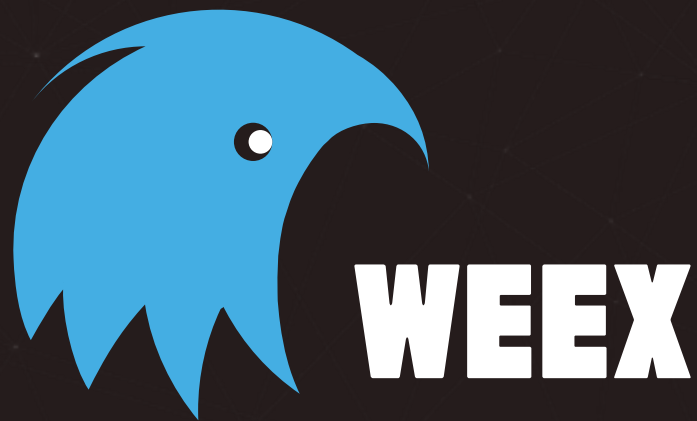
困境与展望

- 调试体验太分离
- 部分功能需要extension提供
- Weex Devtool 1.0.0 !



Q&A

- <https://github.com/weexteam/weex-toolkit>
- <https://github.com/weexteam/weex-devtool>
- <https://github.com/weexteam/weex-pack>



<https://github.com/alibaba/weex>