

测试之美 360技术嘉年华

第五季

对无线测试平台中进行大批量
设备部署

1

如何适配版本和监控 界面变化

处理界面变化 进行版本适配
处理阻塞弹窗 进行机型适配

2

如何解决手机联 网问题

WIFI信道干扰 利用虚拟路由
无法使用5G 使用USB上网

如何.....

3

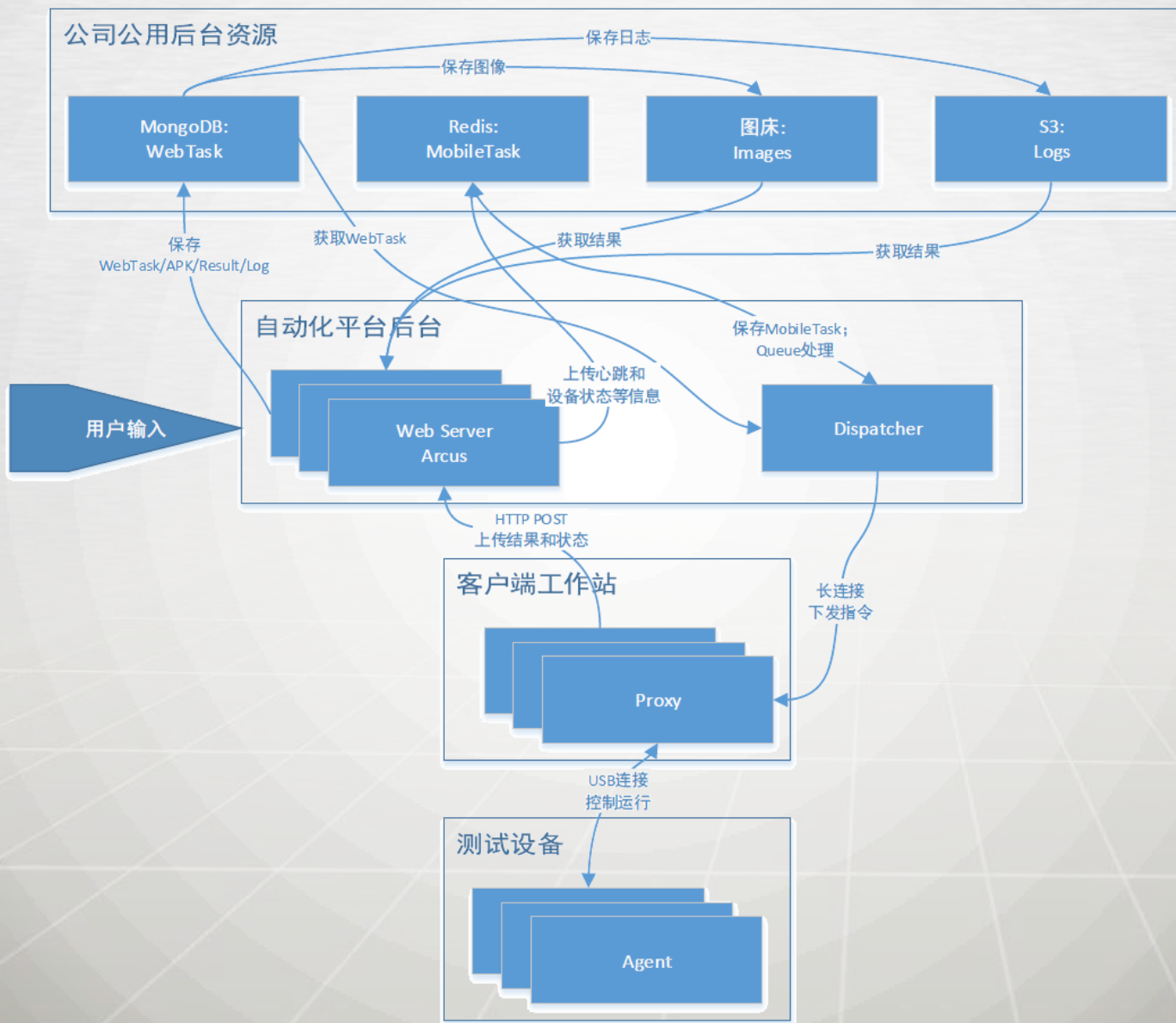
如何提高设备稳定 性

长时间运行 定期清理
死机、无响应 远程可控

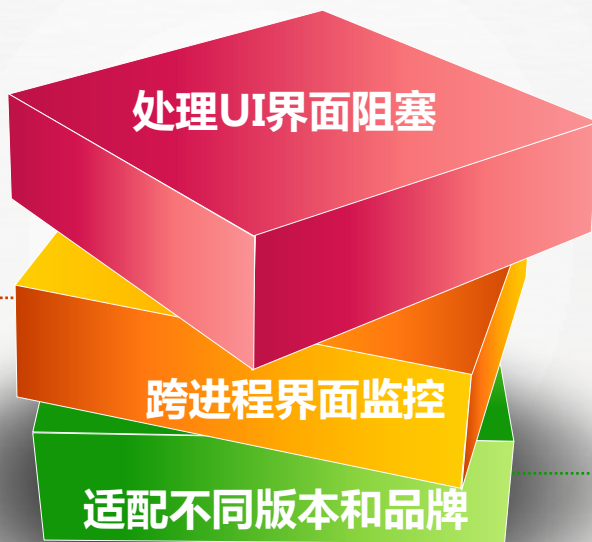
4

如何实现为业务 定制

需求各异 实现插件化
开发任务量大 激发用户潜能



- Instrumentation或者Accessibility均无法满足需求
- Android API 16、17 中利用UI Automator功能通过反射调用UIDevice对象实例的方法
- 利用getAutomatorBridge()方法获取UI Automator服务，本质Accessibility Service



- 针对不同品牌手机中时常会弹出各种确认界面，例如权限提示、安装阻塞、自动运行等
- 配置UI阻塞界面操作行为名单，每次界面变化遍历一次，找到配置项执行点击、滑动等一系列操作

- 脱离对ROOT的依赖支持更多设备
- 更多使用系统服务，如：UI Automator、Accessibility
- 使用云端配置和代码热替换，快速部署

使用UiDevice的getAutomatorBridge()方法

Provides access the [QueryController](#) and [InteractionController](#)

Returns:

[ShellUiAutomatorBridge](#)

```
84     UiAutomatorBridge getAutomatorBridge() {  
85         if (mUiAutomationBridge == null) {  
86             throw new RuntimeException("UiDevice not initialized");  
87         }  
88         return mUiAutomationBridge;  
89     }  
90 }
```



UiAutomatorBridge

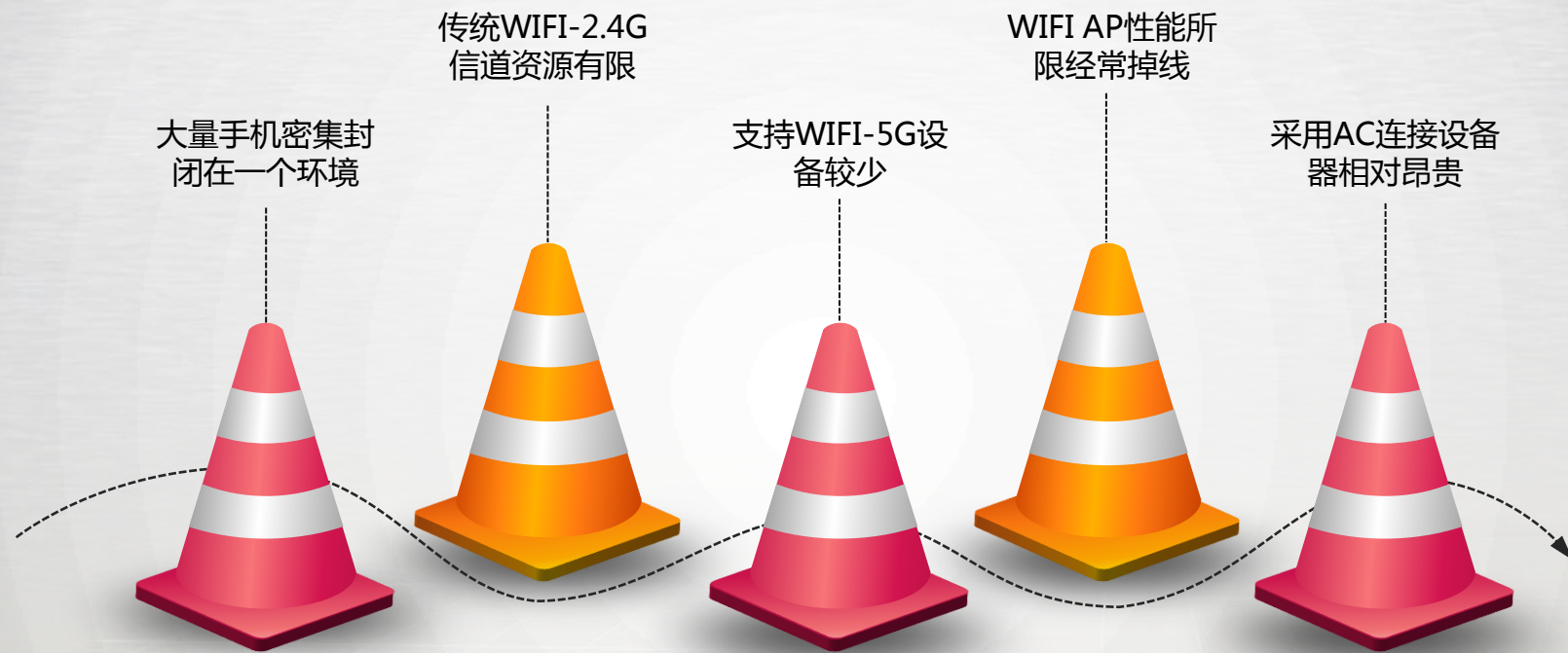
- `UiAutomatorBridge(UiAutomation) : void`
- `executeCommandAndWaitForAccessibilityEvent(Run)`
- `getDefaultDisplay() : Display`
- `getInteractionController() : InteractionController`
- `getQueryController() : QueryController`
- `getRootInActiveWindow() : AccessibilityNodeInfo`**
- `getRotation() : int`
- `getSystemLongPressTime() : long`
- `injectInputEvent(InputEvent, boolean) : boolean`
- `isScreenOn() : boolean`
- `performGlobalAction(int) : boolean`
- `setCompressedLayoutHierarchy(boolean) : void`
- `setOnAccessibilityEventListener(UiAutomation.OnAc`
- `setRotation(int) : boolean`

可云控的UI阻塞元素处理机制

品牌筛选

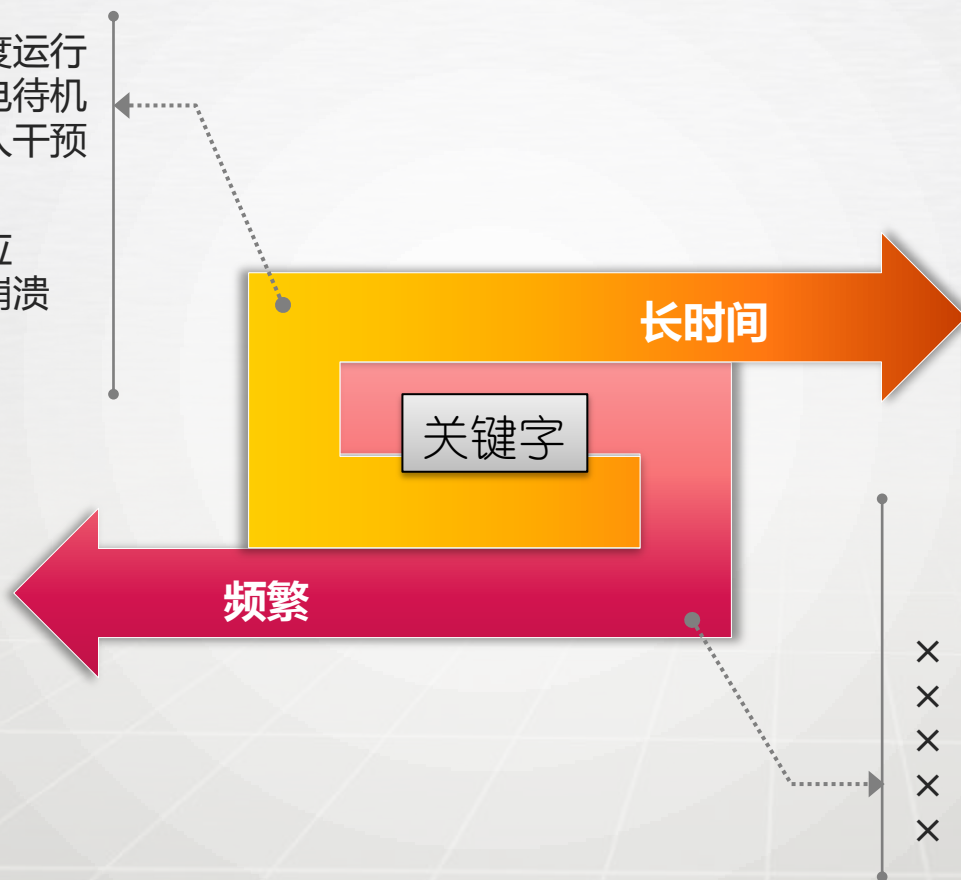
```
<config>
  <BrandItems brand="vivo">
    <!-- phase 代表 适用阶段 1 任何时候都生效 2 非任务运行期间生效 3 或者未配置 只在跑任务时生效 4 关闭-->
    <BrandItem desc="安装程序的界面">
      <condition text="在手机上安装未知来源的应用" class="android.widget.TextView" />
      <handler text="好" package="android" class="android.widget.Button" action="16" wait="1" phase="1"/>
    </BrandItem>
    <BrandItem desc="安装程序的界面">
      <condition text="它将获得以下方面的权限" class="android.widget.TextView" />
      <handler text="下一步" class="android.widget.Button" action="16" wait="1"/>
    </BrandItem>
    <BrandItem desc="点击权限框">
      <condition text="权限" class="android.widget.TextView" />
      <handler text="允许" class="android.widget.Button" action="16" wait="1"/>
    </BrandItem>
  </BrandItems>
</config>
```

查找满足condition的元素
即执行handler中规定的操作





- × 长期高强度运行
- × 长时间充电待机
- × 长时间无人干预
- × 手机死机
- × 操作无响应
- × 系统服务崩溃



- × 频繁安装卸载
- × 频繁执行Shell命令
- × 频繁USB-SOCKET操作
- × 频繁调用系统服务
- × 频繁掉电关机



硬件

- ✓ 大功率USB-HUB
- ✓ 原装USB数据线
- ✓ 高性能服务器

软件

- ✓ 设备空闲时杀掉后台进程
- ✓ 累计出现Socket等通讯错误自动重启手机
- ✓ 强大后台管理系统，可对单台、多台设备更新程序、截屏、远程控制

自行接入无限扩充

私有设备 独占不排队
设备在身边 测试在眼前

免费好用机型全

免费无成本 机型多又全
专业管理 无需操心



P 平台

测试项目单一 兼容、脚本、UI遍历
测试顺序固定 安装、启动、卸载
测试结果单一 统一回收前端展示

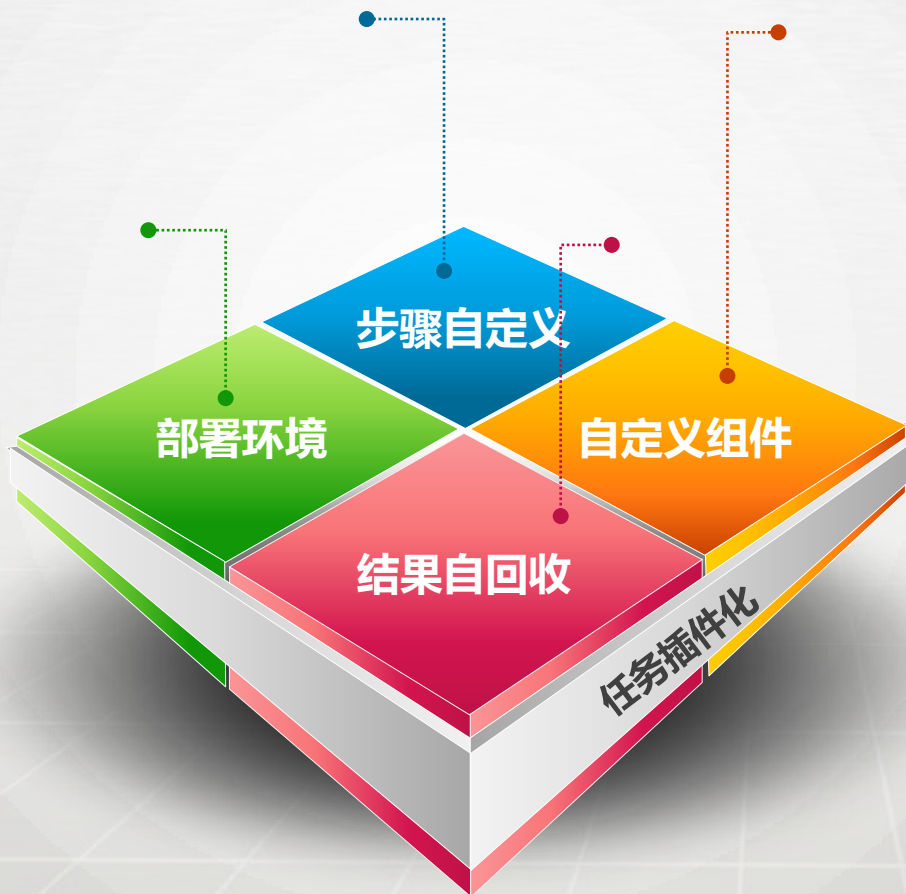


U 用户

精简步骤 去掉不需要的
测试环境部署 推送文件、安装APK
测试结果自回收 回收统计更自如

- 更改任务执行顺序
- 加入自定义步骤

- 推送测试文件（病毒样本、媒体文件，等等）

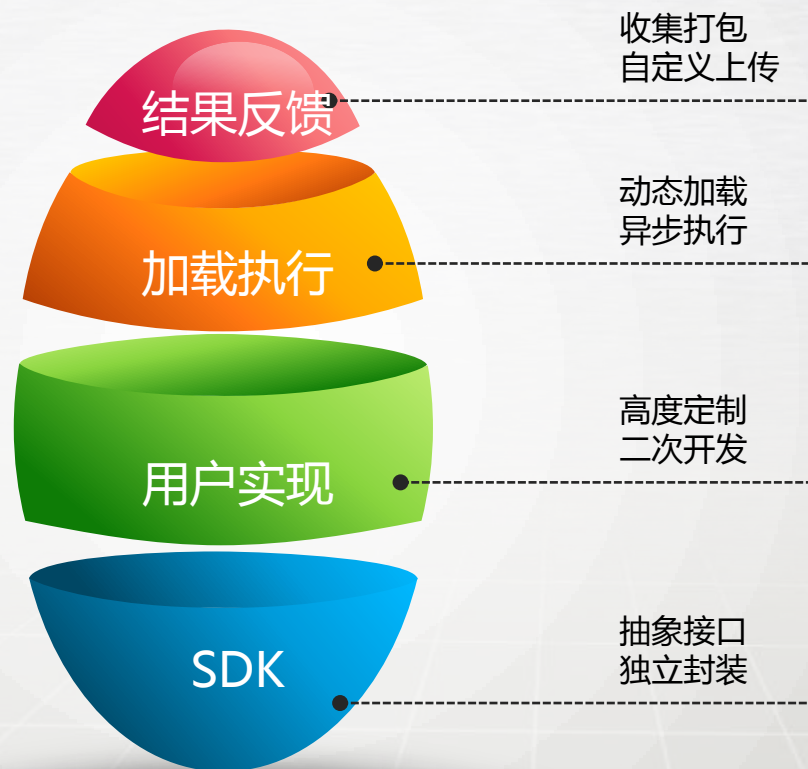


- 可调用自定义JAR
- 调用其他自定义组件

- 测试结果文件用户自行打包
- 发送到指定位置或者展示在前端

任务插件化

提供SDK接口 文档
用户实现接口 形成JAR包



THANKS!