

< HDC.Together >

华为开发者大会 2020

# HarmonyOS应用测试实践

## HarmonyOS应用测试挑战

- > 小手表、大屏幕，App如何在多形态设备上快速部署与测试？
- > 碰一碰、扫一扫，App如何快速构建分布式业务的测试条件？
- > 来回跑、反复查，App如何在复杂场景中快速识别问题并定位根因？

# 整体解决方案 < DECC测试框架 >

一站式多语言测试框架，支撑全流程的测试活动，看护应用质量



## DECC

测试执行调度层

xDevice Test Runner

测试框架层

CTest

CppTest

JUnit

JSUnit

UITest

测试基础库

APIs (Audio、Graphic...)

Tools (Trace、Dump、KingKong...)



## 具备三大特征：

### ✓ 高效开发集成框架

1行配置、1步调用、1键到底 ...

### ✓ 支持多语言、跨语言

C/C++/Java/JS/Python ...

### ✓ 支持多设备、跨设备

手表、大屏、车机 ...

# < DECC测试框架 > 功能丰富 高效易用

## 1行配置

结合IDE，最少配置步骤，得到可用框架

## 1步调用

提供足够丰富的API封装，直接调用，极少人工干预

## 1键到底

一键执行，获取清晰友好的结果反馈协助定位

// 一行配置，即可集成JUnit测试能力

```
dependencies {
    implementation fileTree(dir: 'libs', include:
['*.jar'])
    testImplementation
'harmonyosx.testkit:harmonyjunitrunner:0.2'
    testCompile 'junit:junit:4.12'
}
```

DECC-JUnit

240+

DECC-JSUnit

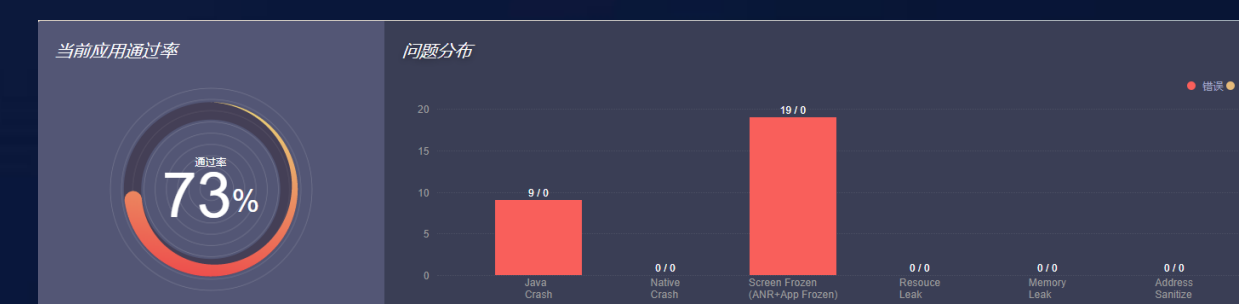
70+

DECC-UITest

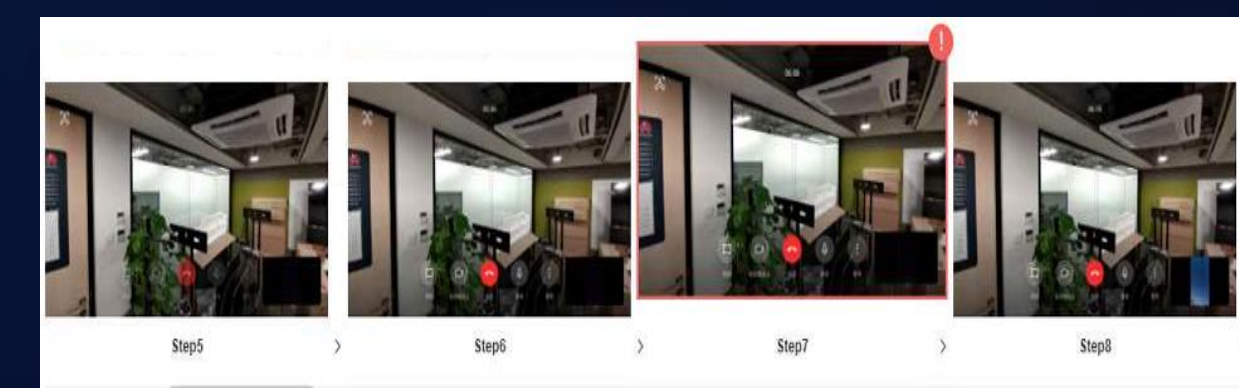
150+

// 断言： 是/否、相等、非空...  
 // 线程处理： 启动Ability、获取上下文...  
 // 原子操作： 连接Wi-Fi、切换用户...  
 // 界面能力： 点击、双击、拖动、双指...  
 // 扩展能力： 循环、并发、多用例组合...

// 可看到整体执行结果



// 可对单用例执行结果提供丰富定位手段

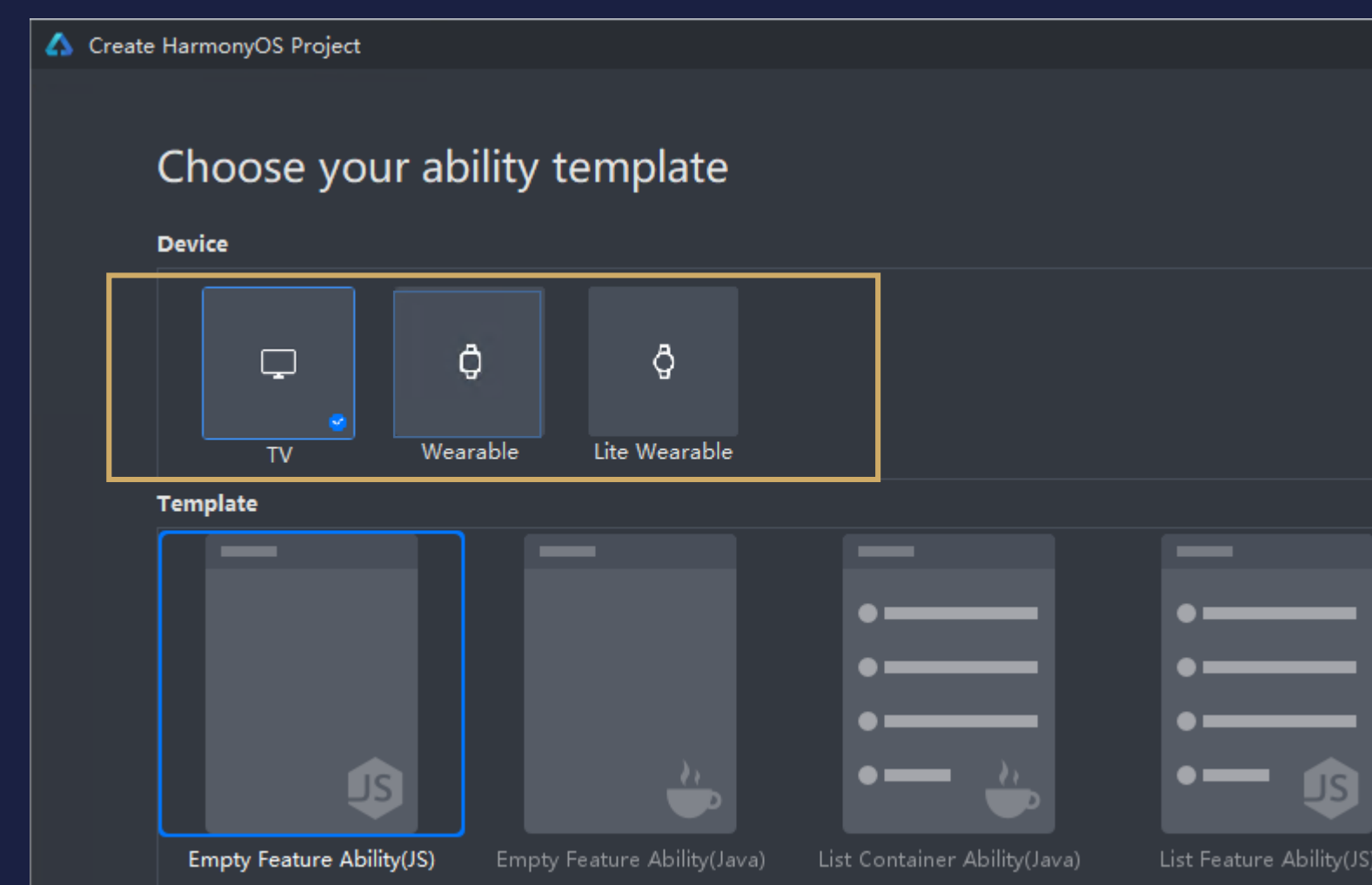
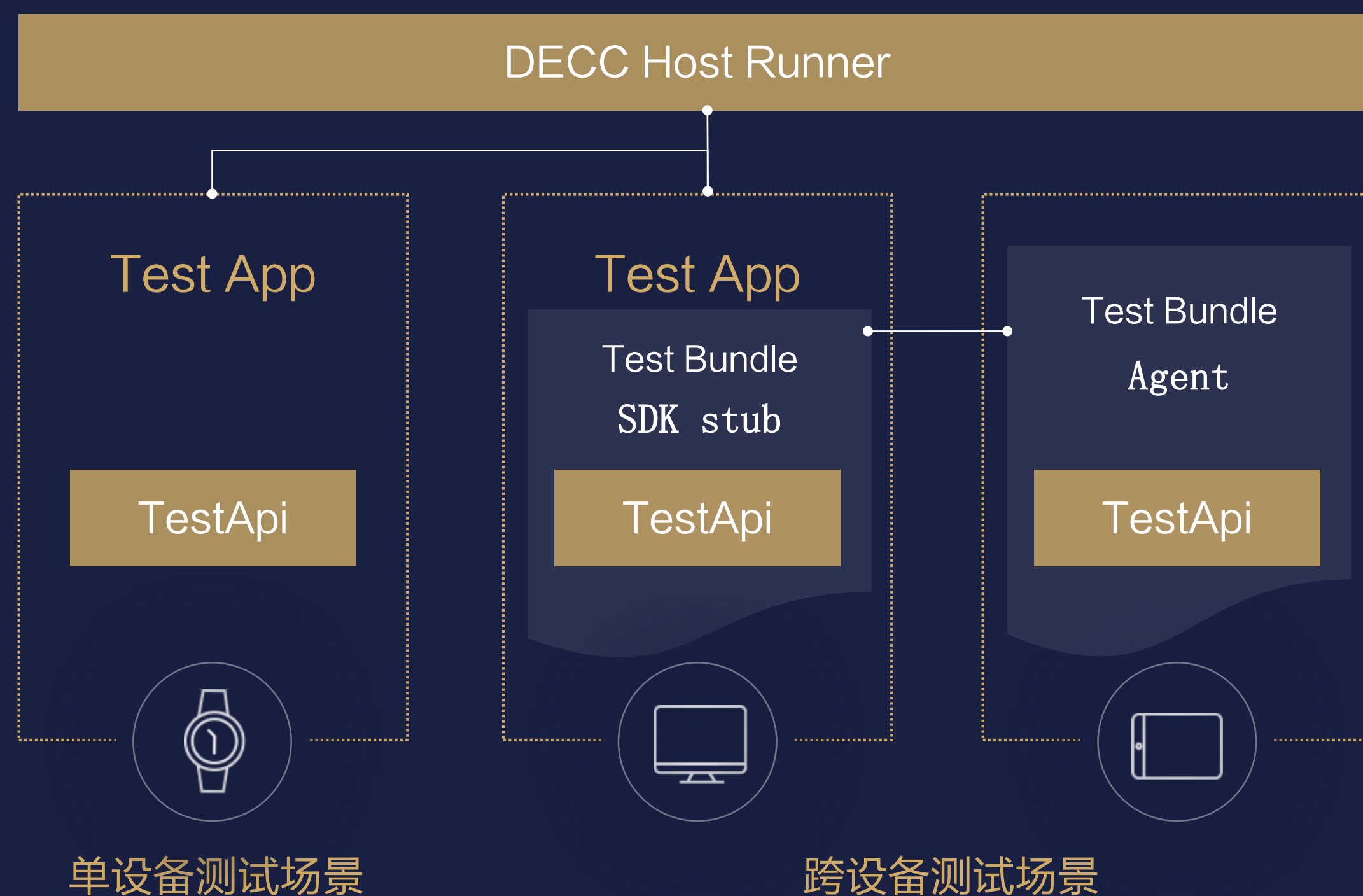




# < DECC测试框架 > 支持多语言、多类型测试



## < DECC测试框架 > 支持多设备、跨设备测试



- ✓一步构建 多设备部署
- ✓一次编写 跨设备执行

## < DECC-JSUnit > JavaScript单元测试

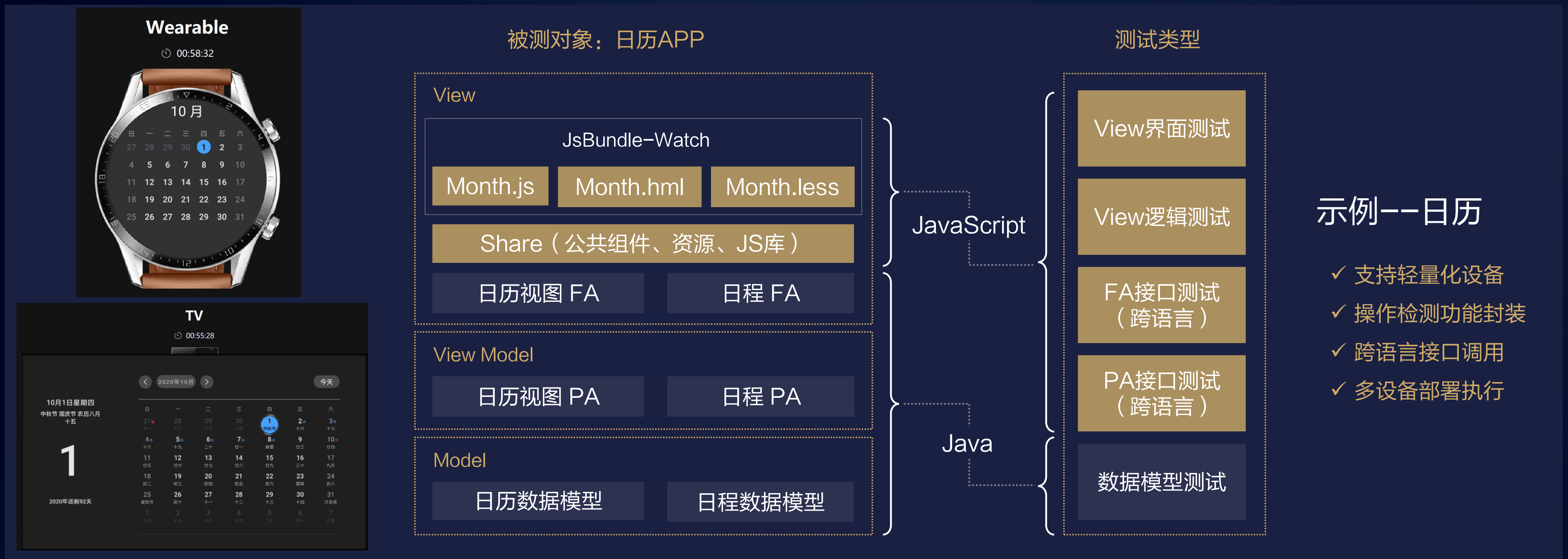
JS单元测试框架：提供70+封装好的API能力，调用方便，支持轻设备到富设备各种形态的JS测试



插件化机制，按需灵活裁剪或添加组件

- ✓ 轻量化：可裁剪至最小，放入<32K 空间的设备中执行
- ✓ 跨设备：用例一次编写，可跨设备调用接口执行
- ✓ 跨语言：无需切换测试框架，直接跨语言调用API

# < DECC-JSUnit > 日历Demo





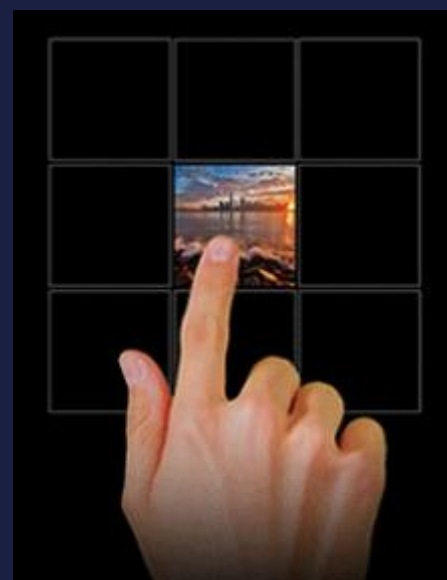
**TV**  
00:55:28



# < DECC-JSUnit > JavaScript单元测试

## < DECC-UITest > 系统集成测试

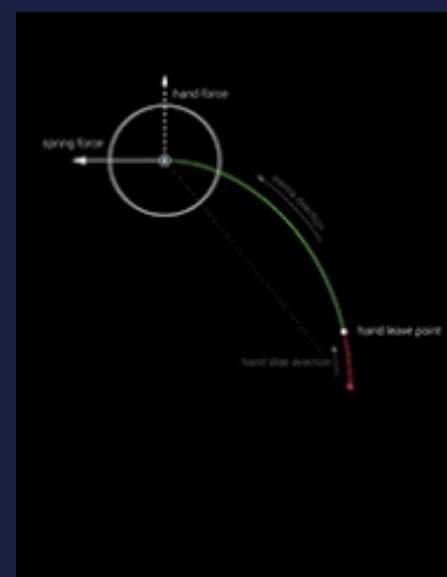
UI集成测试框架：提供150+封装好的API能力，支持UI界面自动化测试



- ✓ 支持UI操作处理，支持对系统状态进行模拟与检查

### UI操作能力

- 界面整体：DUMP、快速截图、智能识别
- 控件处理：点击、滑动、双击、长按、多指
- 硬件模拟：指纹、呼吸灯、休眠、低电量



- ✓ 支持对控件的动效、渲染等视觉类效果进行检测

### 检测能力

- 检查能力：控件检查、文本检查、动效检查
- 文件操作：纯文本、XML、数据库、日志

- ✓ 支持对轻设备进行界面自动化测试



轻量化设备上只需部署少量几个接口，即可通过Host主机获得全部的UI测试能力

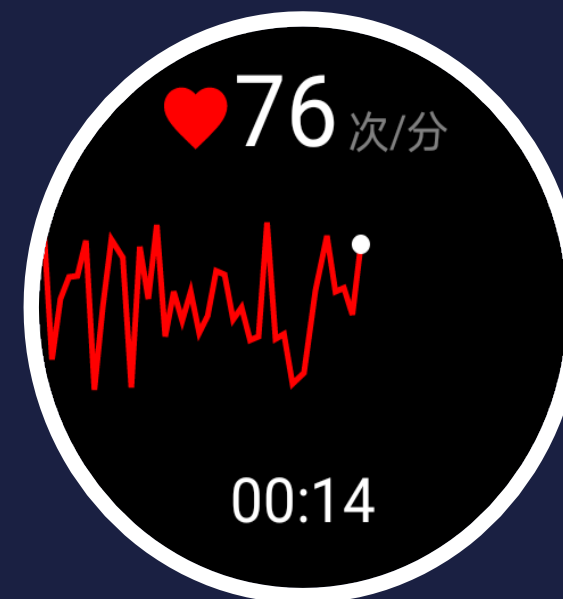
## < DECC-UITest > 手表UI测试Demo



轮盘索引



视频播放



心率监测



海量字体



帧动画



天气圆角

### 示例——查看手表特性

- ✓ 可查找：Dump查看界面控件树信息
- ✓ 可控制：点击、滑动等操作
- ✓ 可检测：获得控件状态数据，判断处理结果
- ✓ 可分析：拉通对比分析，如内存性能等

# < DECC-UITest > 系统集成测试



PyCharm interface showing a Python file named `DumpTreeTest.py` in the `src` directory of the `xdevice` project.

The code defines a `DumpTreeTest` class inheriting from `TestCase`. The class includes methods for initialization, clicking a button, setting up the test environment, and performing a test step.

```
1  # -*- coding: utf-8 -*-
2  # from devicetest import * # @UnusedWildImport
3  from aw import * # @UnusedWildImport
4  from _core.constants import DeviceTestType
5  from _devicetest, devicetest.core.test_case import TestCase
6  import bin2png_test as b2p
7
8
9  class DumpTreeTest(TestCase):
10     def __init__(self, controllers):
11         self.TAG = self.__class__.__name__
12         self.tests = [
13             "test_step1",
14         ]
15         TestCase.__init__(self, self.TAG, controllers)
16
17     def click(self, button, timeout=800):
18         common_lite.click(self.devices[0], button[0], button[1], timeout)
19
20     def setup(self):
21         print("预置工作:初始化设备.....")
22
23     def test_step1(self):
24         print("执行用例.....")
25         common_lite.ui_dump(self.devices[0], timeout=800)
26
27     def teardown(self):
28         print("收尾工作:解除挂载.....")
29
```

The left sidebar shows the project structure, including directories like `src`, `test`, and `testcases`, and files like `init.py`, `harmony_test.bat`, and `README.md`.

The bottom status bar indicates the current file is `DumpTreeTest.py` and the method is `__init__()`.

激活 Windows  
转到“设置”以激活 Windows。

## < DECC-KingKong > 稳定性测试



### 关键测试服务:

- ✓ 智能的压测服务 (Intelligent Stress Test Service)
- ✓ 完善的定位服务 (Complete Location Service)
- ✓ 丰富的检测服务 (Rich Detection Service)
- ✓ 远程真机服务 (Remote Real Device Service)

KingKong在EMUI历史上发挥了重要作用，移植和适配后将对三方开发者开放能力

## < DECC-KingKong > 智能压测服务















### 压测服务：

- ✓ 支持分布式场景的跨设备并发压测
- ✓ 支持自定义压测场景，满足差异业务诉求
- ✓ 基于AI的智能遍历，覆盖更充分、效率更高



# < DECC-KingKong > 检测与定位服务

APP信息	安装启动		稳定性测试							操作	
	安装	启动	Java Crash	Native Crash	Screen Frozen	Total	Resource Leak	Memory Leak	Address Sanitize		
A 7.0.10	✓	✓	1	检测内容		4	✖5	✓0	✓0	--	  
B 11.20.0.10	✓	✓	0	0	11	✖11	✓0	✓0	--	  	
C 10.7.0.7700	✓	✓	1	0	0	✓1	✓0	✓0	--	 	
D 0.2.00.21442	✓	✓	2	0	0	✓2	✓0	✓0	--	 	
E 1.1.20.10040	✓	✓	0	0	0	✓0	✓0	✓0	--	 	






```

5  SYSVMType: Maple
6  APPVMType: Art
7  Foreground: Yes
8  Ability: com.huawei.meettime/.activity.MainActivity
9  Reason: MainThread blocked for 5 seconds
10 Lifetime: 135
11 RootCauseAnalysis:
12 MainThread blocked due to ongoing binder transactions:
13 "main" prio=5 tid=1 Blocked
14 ->- waiting to lock <0x00314445> (a java.lang.Class<huawei.app.ContextImpl>) held by
15 "platform-default-thread-5" prio=5 tid=3 Native
16 ->- locked <0x00314445> (a java.lang.Class<huawei.app.ContextImpl>)
17 TrustStack:
18 "platform-default-thread-5" prio=5 tid=3 Native
19   at com.bytedance.services.account.impl.AccountServiceImpl.getSpiceData(SourceFile:

```

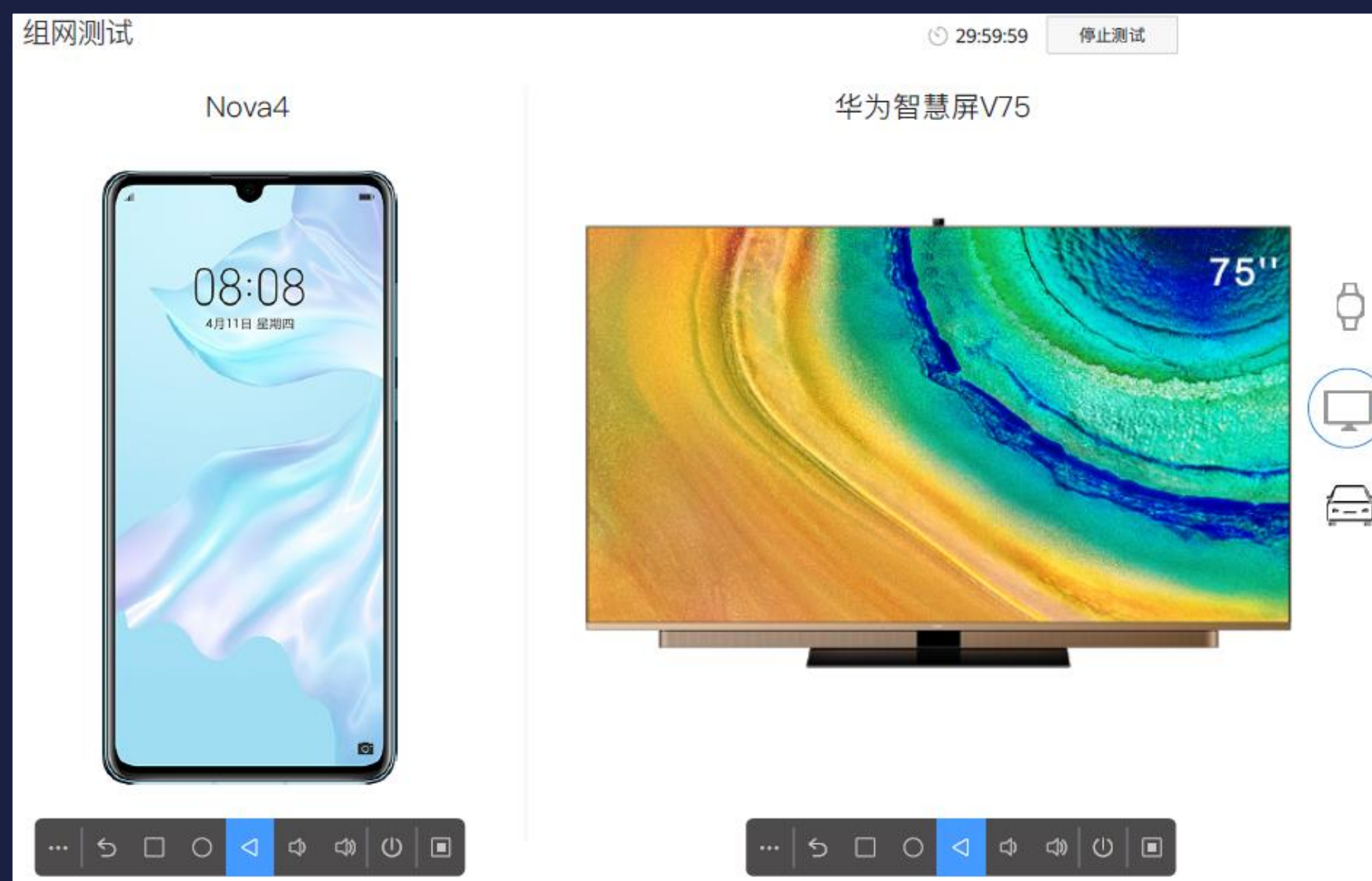
丰富的检测服务： ✓ 内存检测    ✓ Crash    ✓ Freeze

完善的定位服务： ✓ 操作日志    ✓ 失败截图    ✓ 跨端关联分析



## < DECC-KingKong > 远程真机测试服务

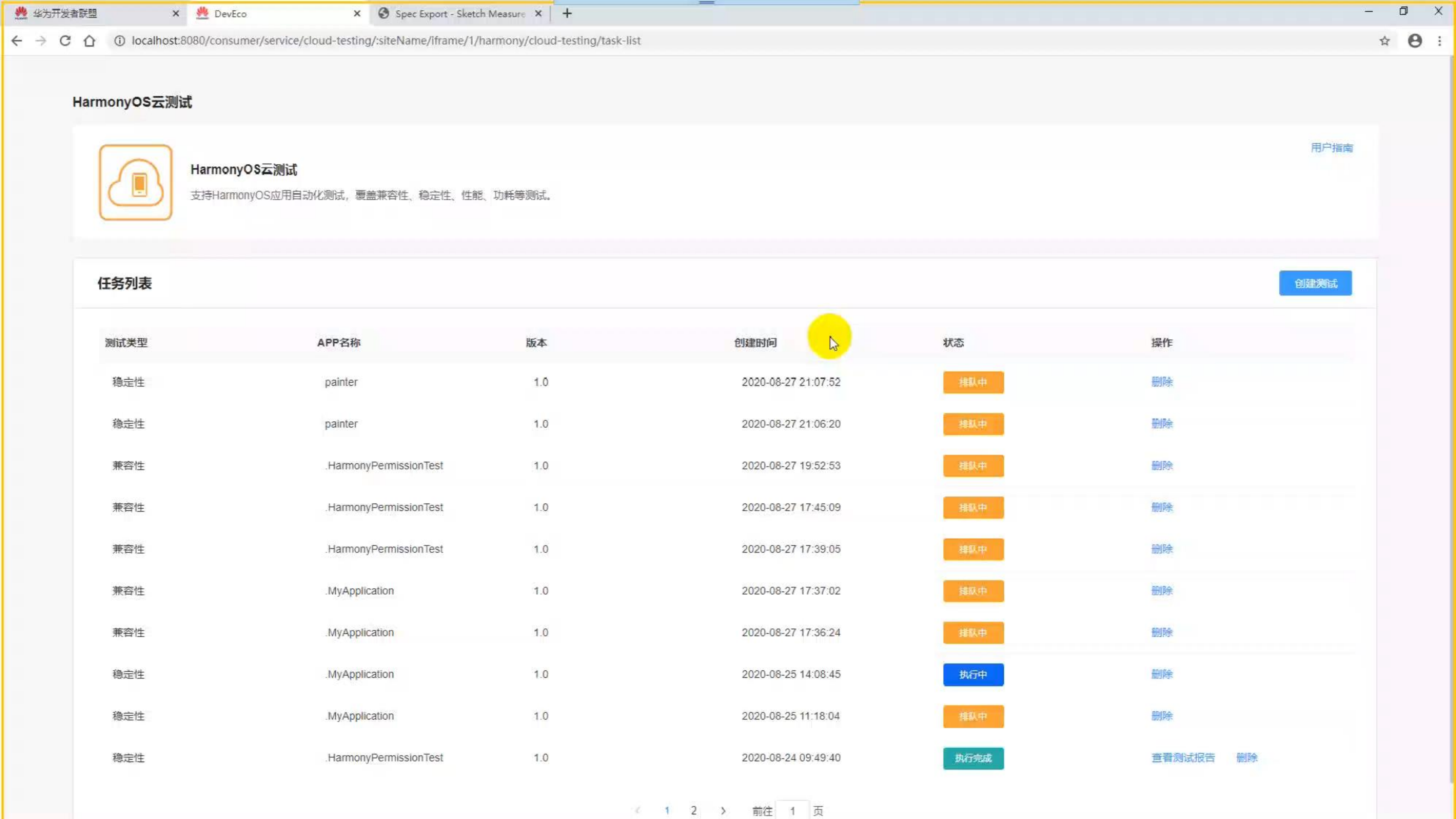
分布式超级终端：支持1+智慧屏、智能手表、车机等场景HarmonyOS应用测试



### 远程真机测试能力：

- ✓ 多屏互动，低时延云端测试
- ✓ 分布式应用调用链定位、定界能力
- ✓ 分布式日志一键导出

# < DECC-UITest > 稳定性测试



## HarmonyOS云测试



### HarmonyOS云测试

支持HarmonyOS应用自动化测试, 覆盖兼容性、稳定性、性能、功耗等测试。

[用户指南](#)

## 任务列表

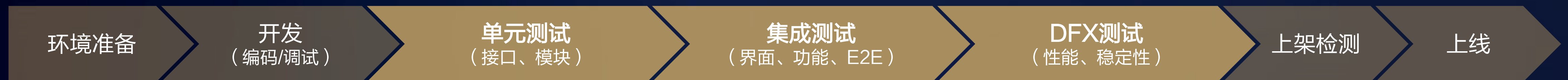
[创建测试](#)

测试类型	APP名称	版本	创建时间	状态	操作
稳定性	painter	1.0	2020-08-27 21:07:52	排队中	删除
稳定性	painter	1.0	2020-08-27 21:06:20	排队中	删除
兼容性	.HarmonyPermissionTest	1.0	2020-08-27 19:52:53	排队中	删除
兼容性	.HarmonyPermissionTest	1.0	2020-08-27 17:45:09	排队中	删除
兼容性	.HarmonyPermissionTest	1.0	2020-08-27 17:39:05	排队中	删除
兼容性	.MyApplication	1.0	2020-08-27 17:37:02	排队中	删除
兼容性	.MyApplication	1.0	2020-08-27 17:36:24	排队中	删除
稳定性	.MyApplication	1.0	2020-08-25 14:08:45	执行中	删除
稳定性	.MyApplication	1.0	2020-08-25 11:18:04	排队中	删除
稳定性	.HarmonyPermissionTest	1.0	2020-08-24 09:49:40	执行完成	<a href="#">查看测试报告</a> 删除



## 总结 < DECC测试解决方案 >

一站式多语言测试框架，支撑全流程的测试活动，看护应用质量



### DECC

测试执行调度层

xDevice Test Runner

测试框架层

CTest

CppTest

JUnit

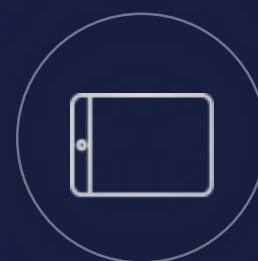
JSUnit

UITest

测试基础库

APIs (Audio、Graphic...)

Tools (Trace、Dump、KingKong...)



### 具备三大特征：

#### ✓ 高效开发集成框架

1行配置、1步调用、1键到底 ...

#### ✓ 支持多语言、跨语言

C/C++/Java/JS/Python ...

#### ✓ 支持多设备、跨设备

手表、大屏、车机 ...





欢迎关注HarmonyOS开发者微信公众号