

< HDC.Together >

华为开发者大会 2020

从零开始开发一个HarmonyOS应用

## 样例程序介绍

编写第一个HarmonyOS应用

样例程序演示

# 图库应用样例

手机端



一键流转

大屏端

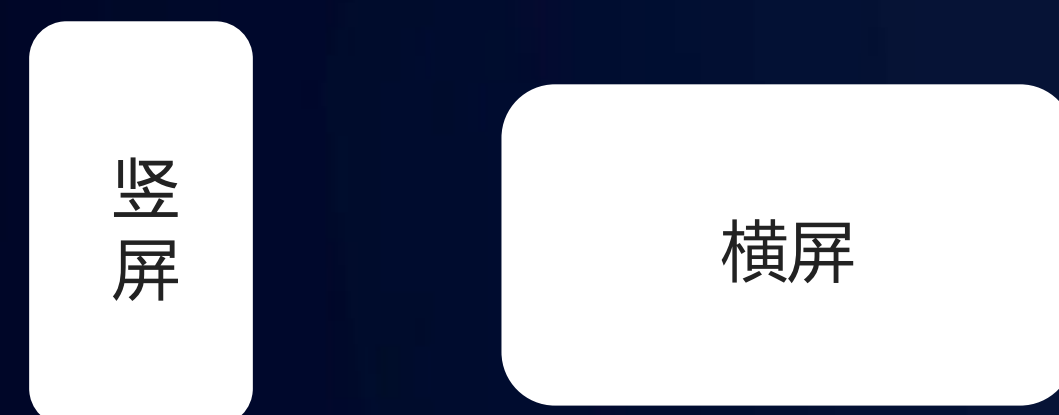


体验1：点击流转按钮，应用从手机流转到大屏，界面布局由竖屏自动切换为横屏。

体验2：遥控器点击确认键，应用从大屏流转到界面，界面布局由横屏自动切换为竖屏

# 功能点分析

## 布局自适应



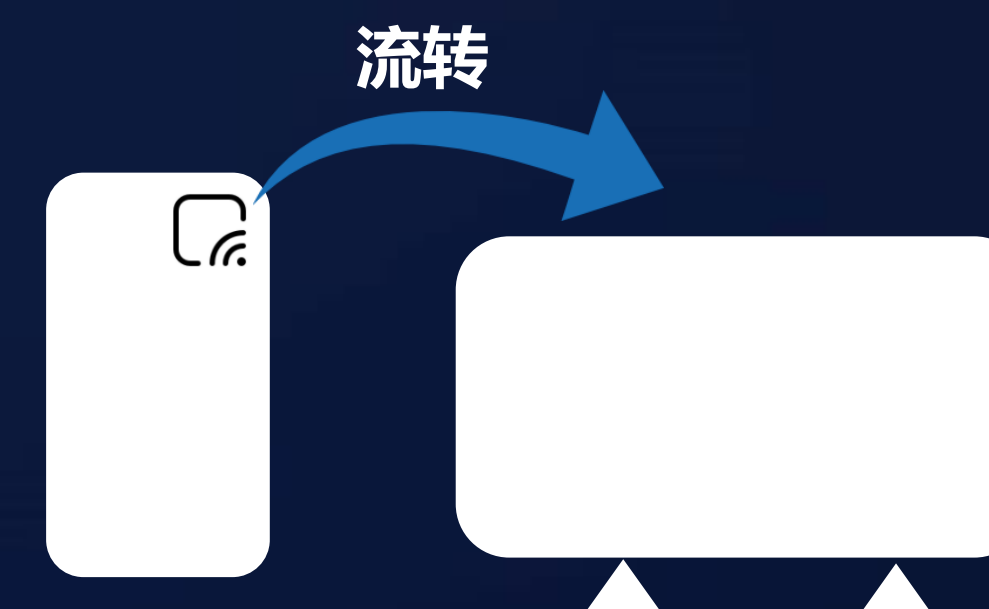
一套布局自适应横屏和竖屏

## 输入事件处理



一套代码兼容触屏、遥控器输入

## 跨设备流转



一键实现跨设备流转

< HDC.Together >

华为开发者大会 2020

样例程序介绍

**编写第一个HarmonyOS应用**

样例程序演示

# 样例应用开发步骤

**Step1:**

创建应用工程



**Step2:**

设计用户界面

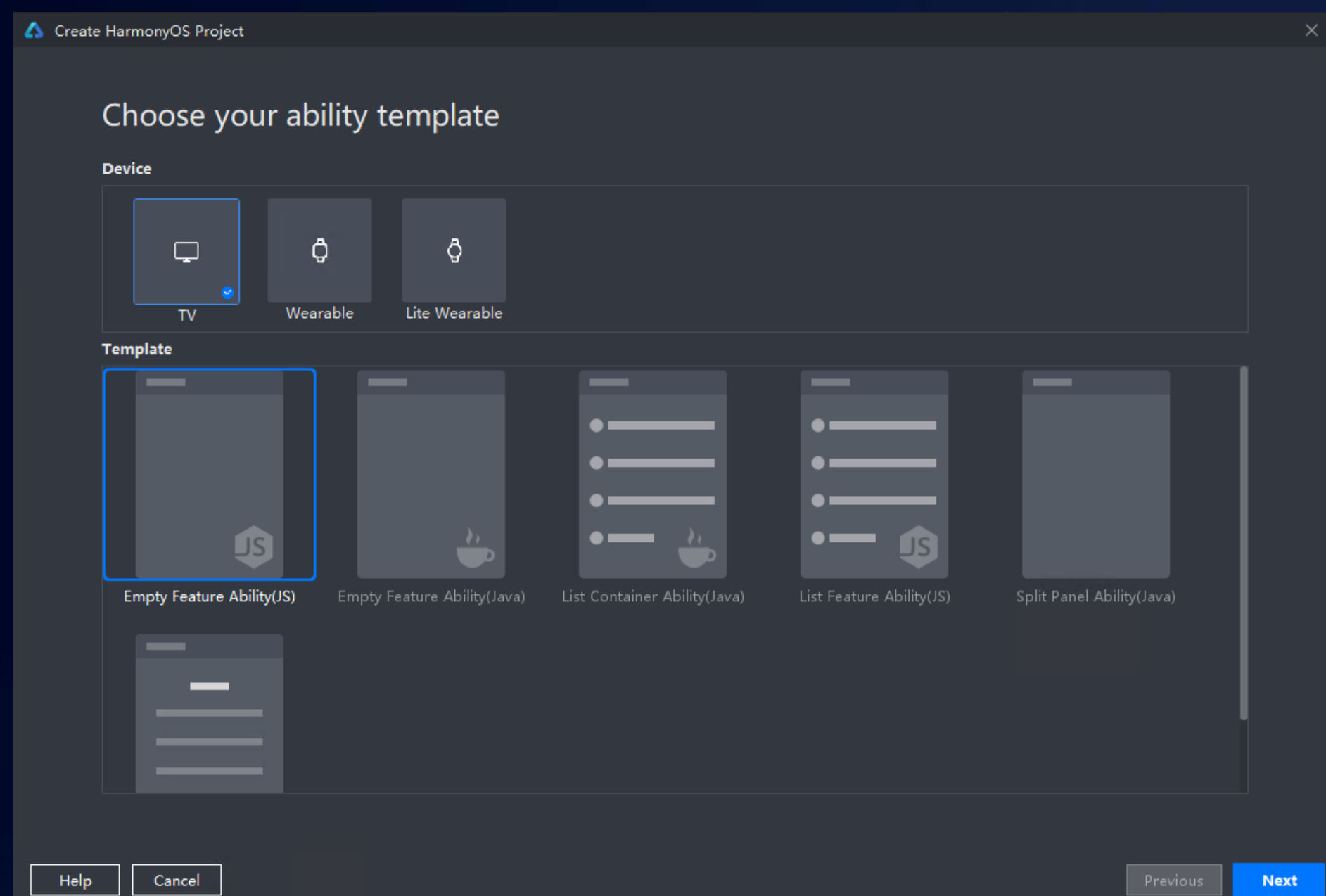


**Step3:**

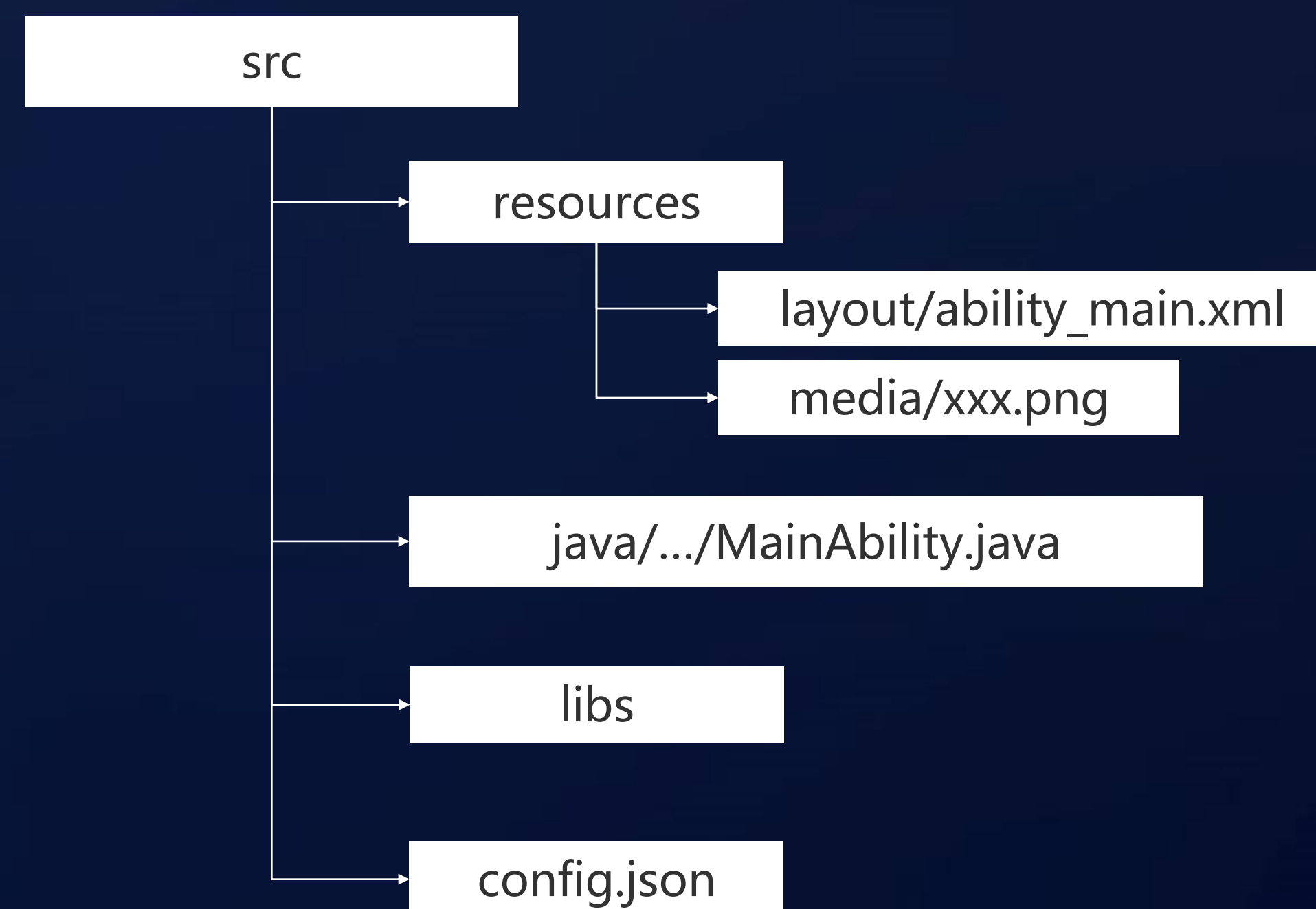
实现跨端流转

# Step1: 使用 DevEco Studio 创建应用工程

## 工程模板选择



## 工程目录结构



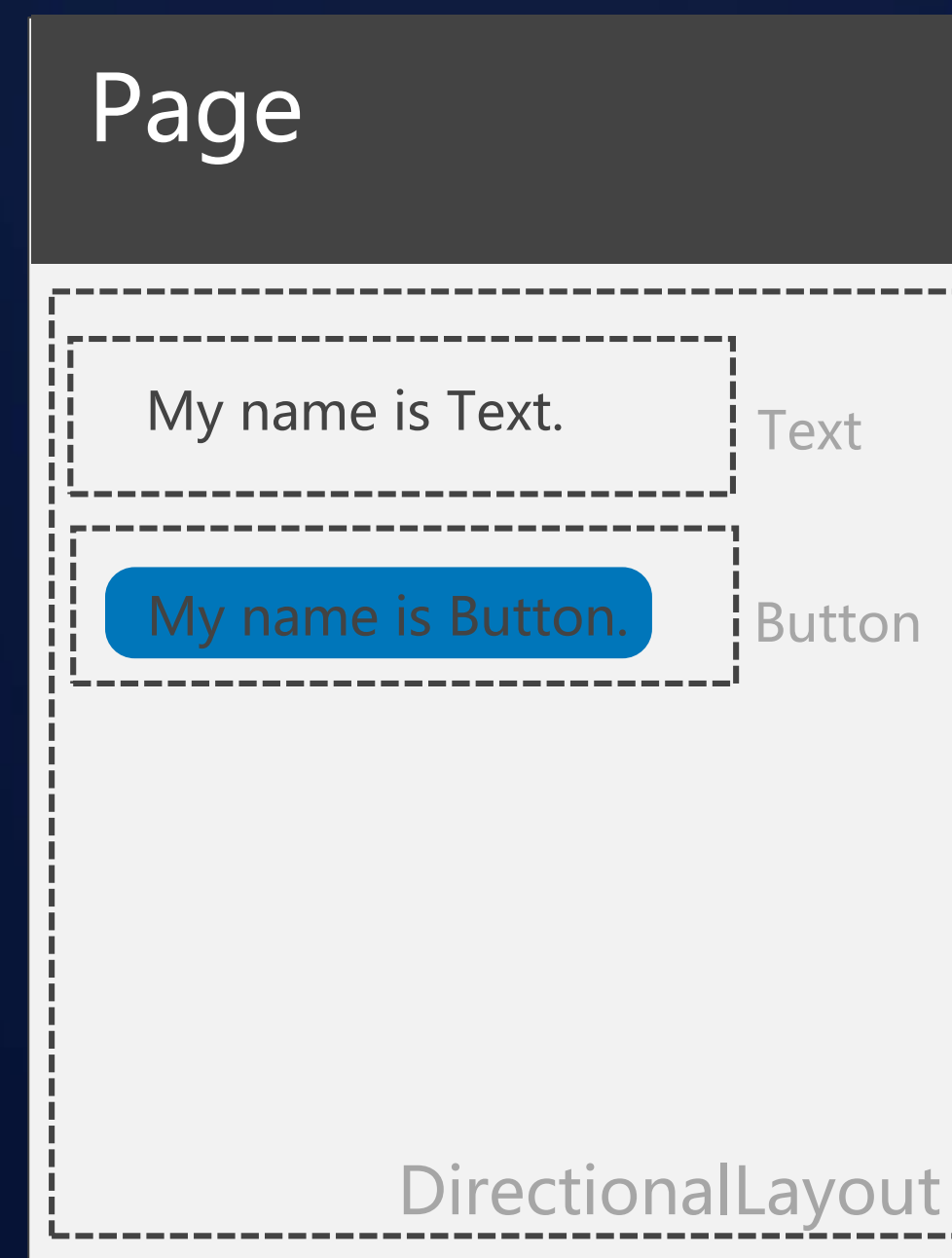


# 工程目录结构

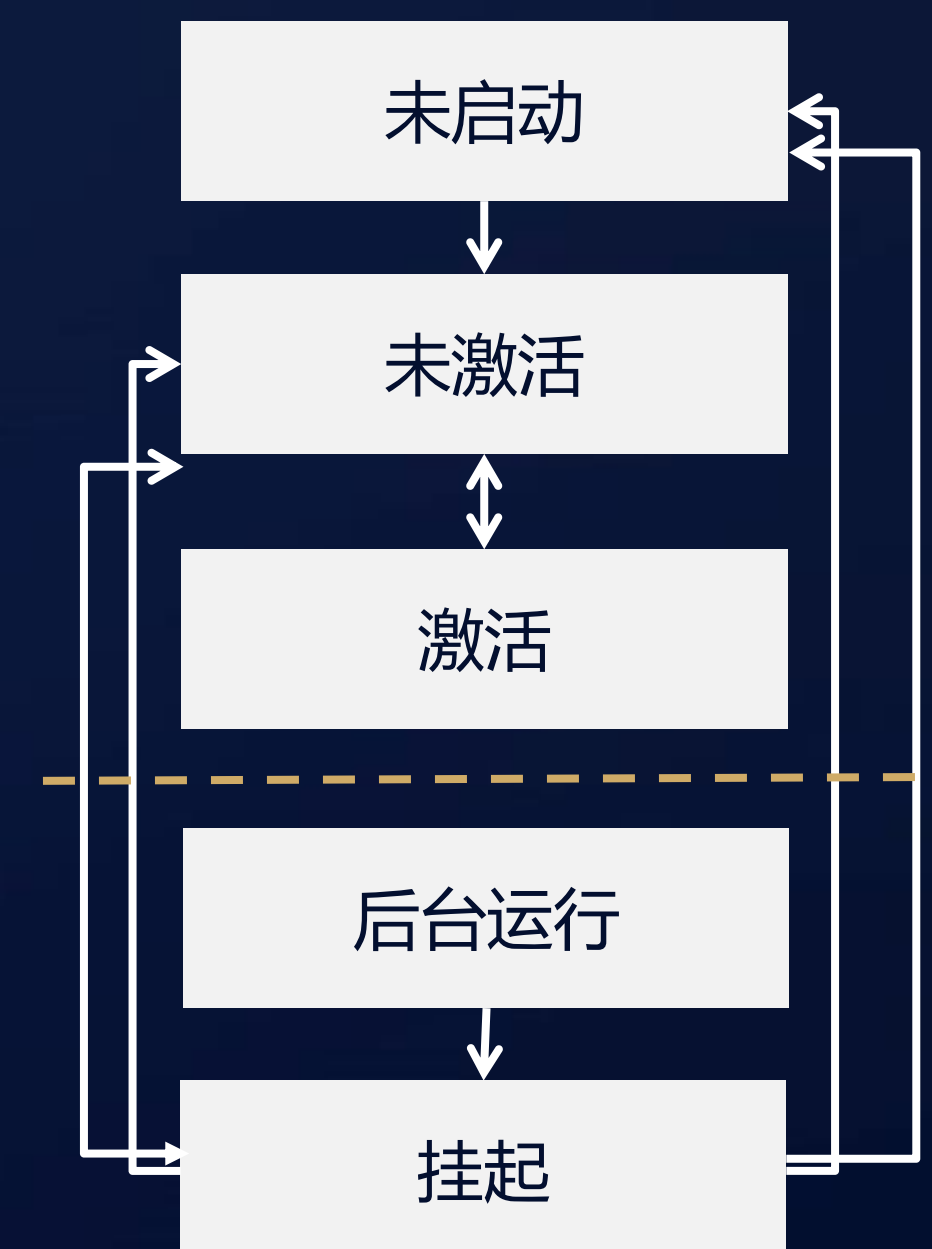
## config.json



## ability\_main.xml



## MainAbility.java



元程序生命周期

< HDC.Together >

华为开发者大会 2020



# 样例应用开发步骤

**Step1:**

创建应用工程



**Step2:**

设计用户界面

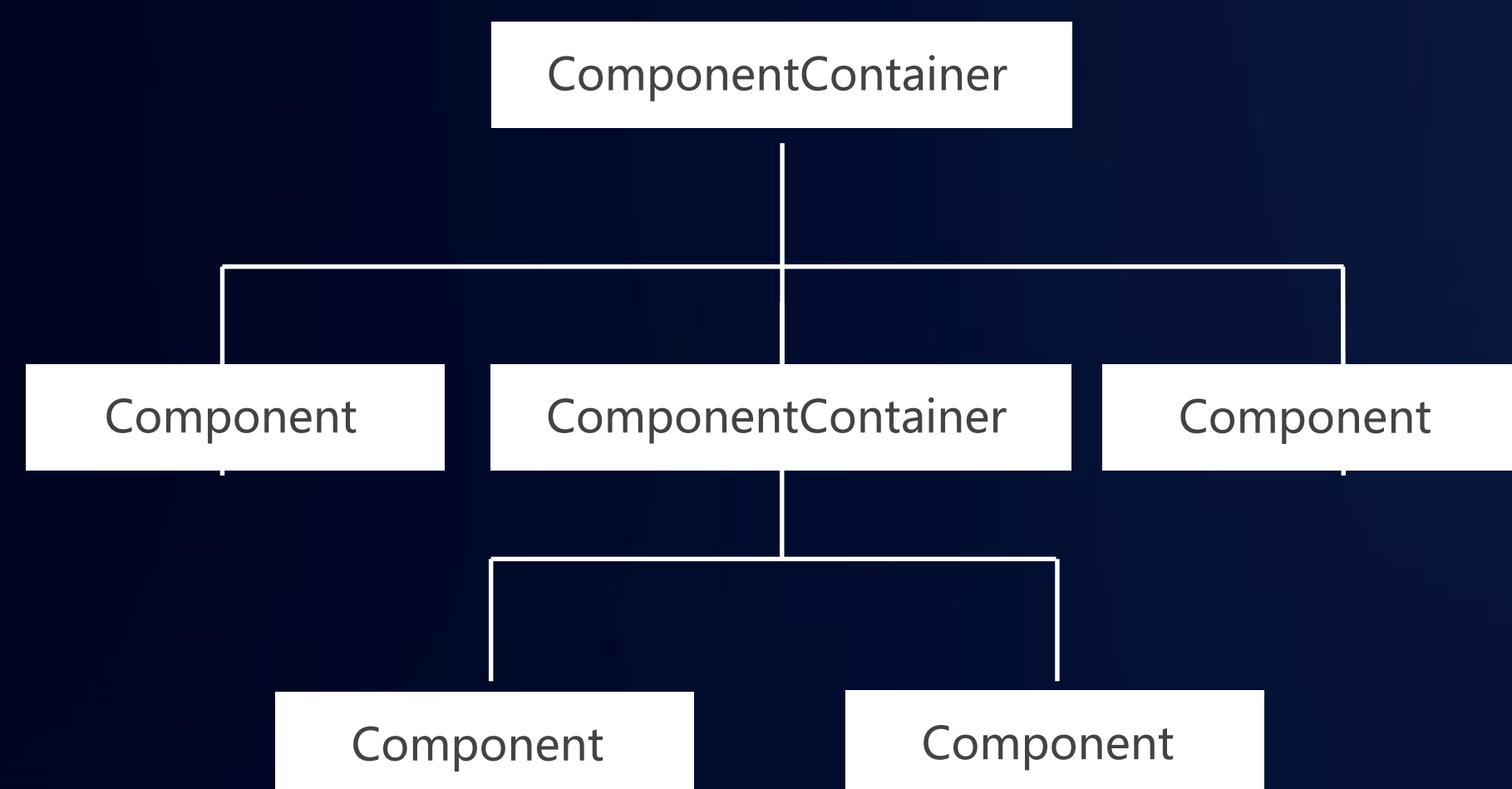


**Step3:**

实现跨端流转

# Step2: 使用 XML 文件进行用户界面设计

## 界面设计元素



## 常用布局控件



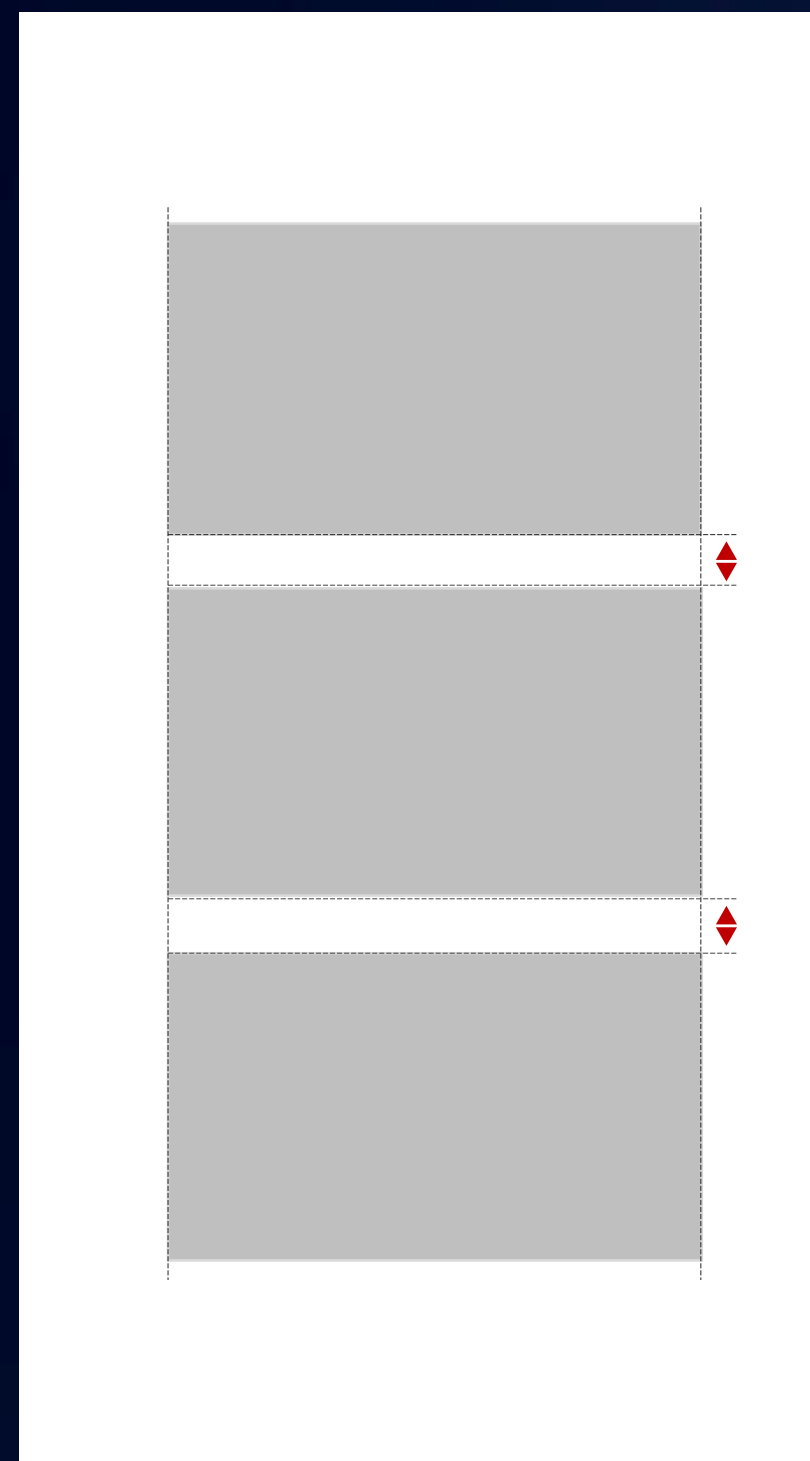
# 用户界面分析

标题部分  
布局相同

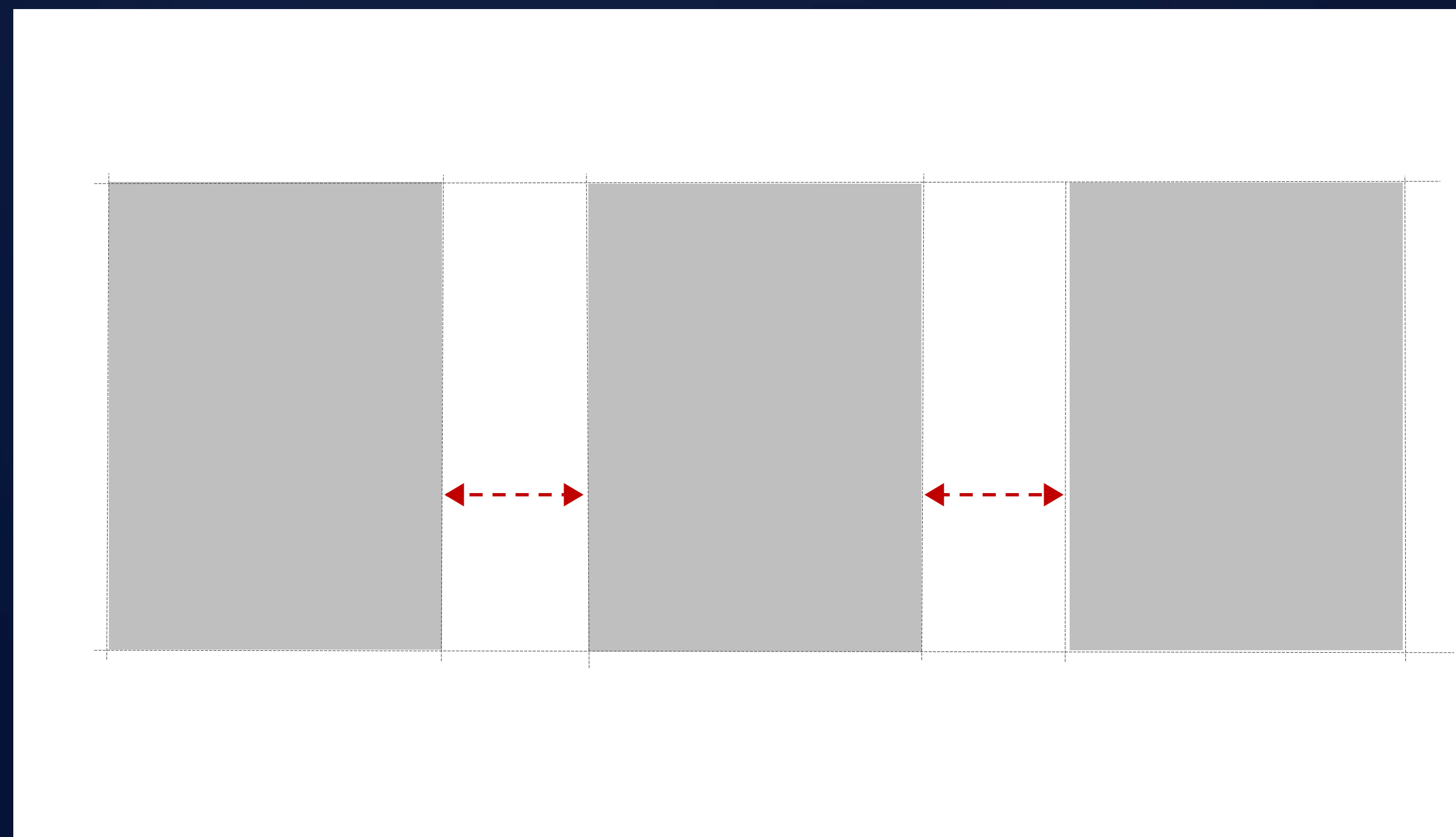
图片分栏由  
竖屏变横屏



# AdaptiveBoxLayout 布局原理



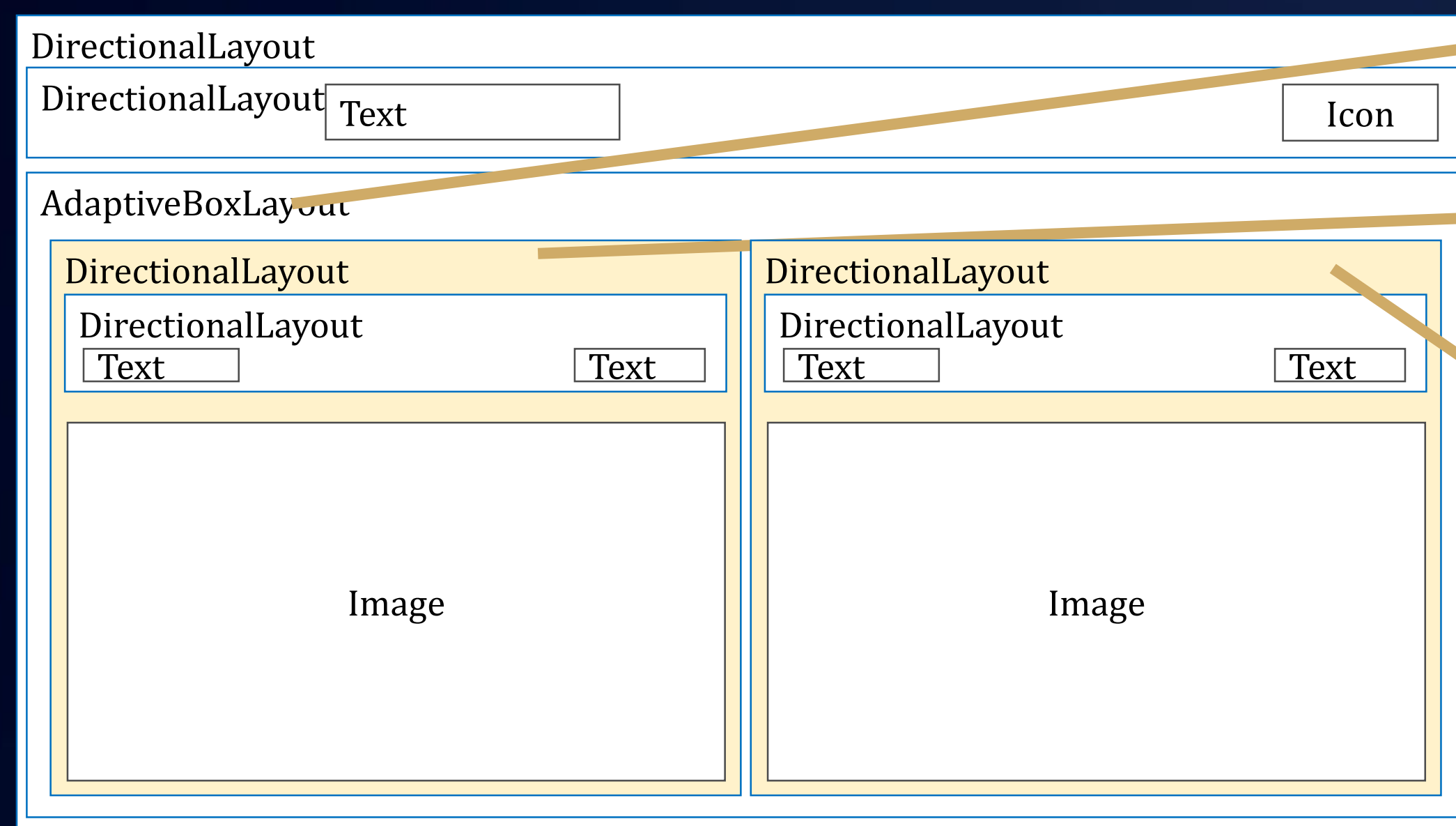
纵向均分



横向均分

# 页面元素分解与XML实现

## 页面元素分解



AdaptiveBoxLayout: 实现横竖屏自适应

## 代码示例: ability\_main.xml

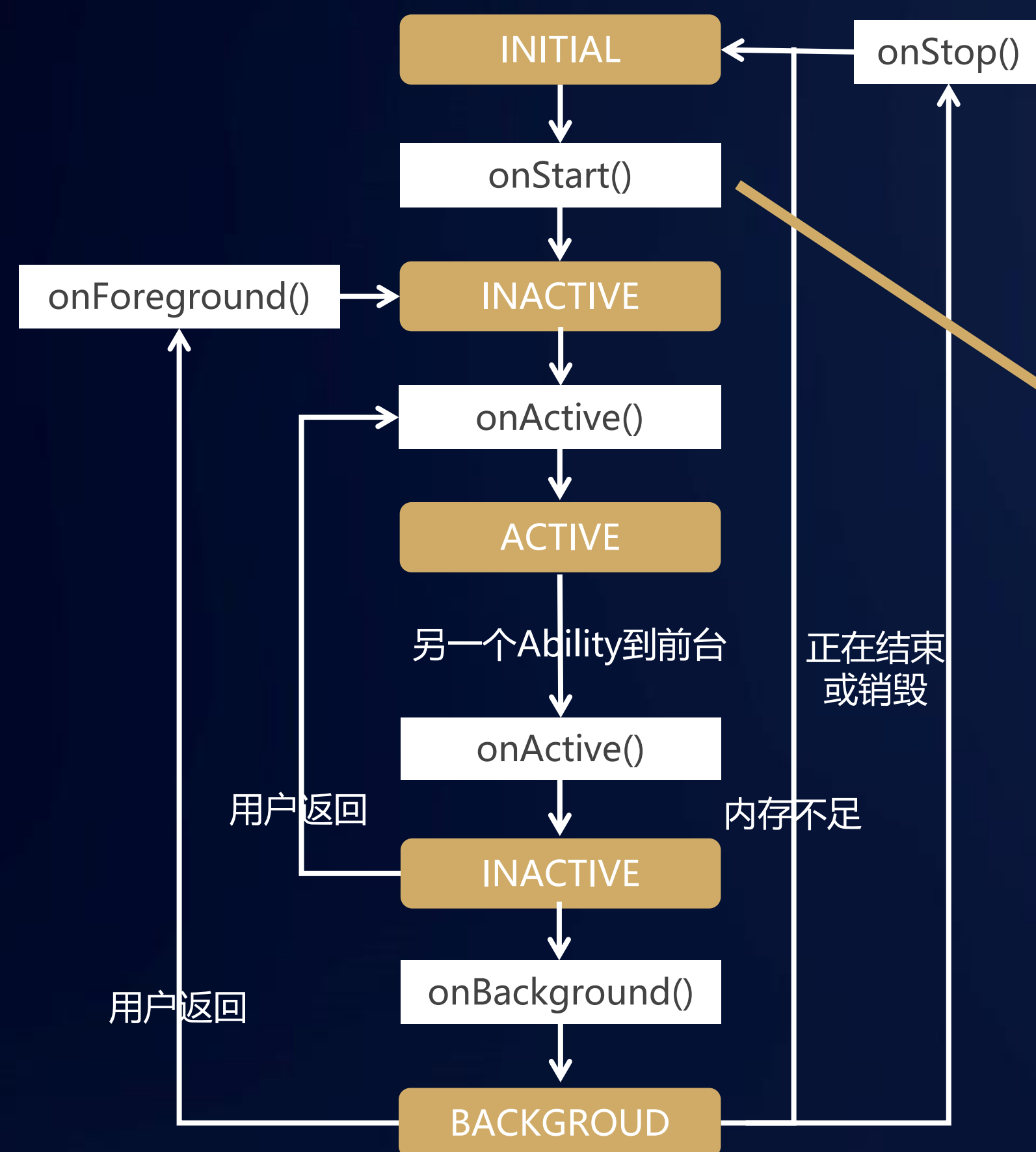
```

<!-- 图库部分 -->
<AdaptiveBoxLayout
    ohos:width="match_parent"
    ohos:height="match_content">

    <!-- 图库推荐 -->
    <DirectionalLayout
        ohos:width="match_parent"
        ohos:height="match_content"
        ohos:orientation="vertical">
        ...
    </DirectionalLayout>

    <!-- 图库照片 -->
    <DirectionalLayout
        ohos:width="match_parent"
        ohos:height="match_content"
        ohos:orientation="vertical">
        ...
    </DirectionalLayout>
</AdaptiveBoxLayout>
    
```

# 在onStart()方法中加载页面布局



## 代码示例: MainAbility.java

```

@Override
public void onStart(Intent intent) {
    super.onStart(intent);
    // 加载页面布局
    super.setUIContent(ResourceTable.Layout_ability_main);
}
  
```

布局文件资源ID

Page的生命周期

< HDC.Together >

华为开发者大会 2020

# 样例应用开发步骤

**Step1:**

创建应用工程



**Step2:**

设计用户界面

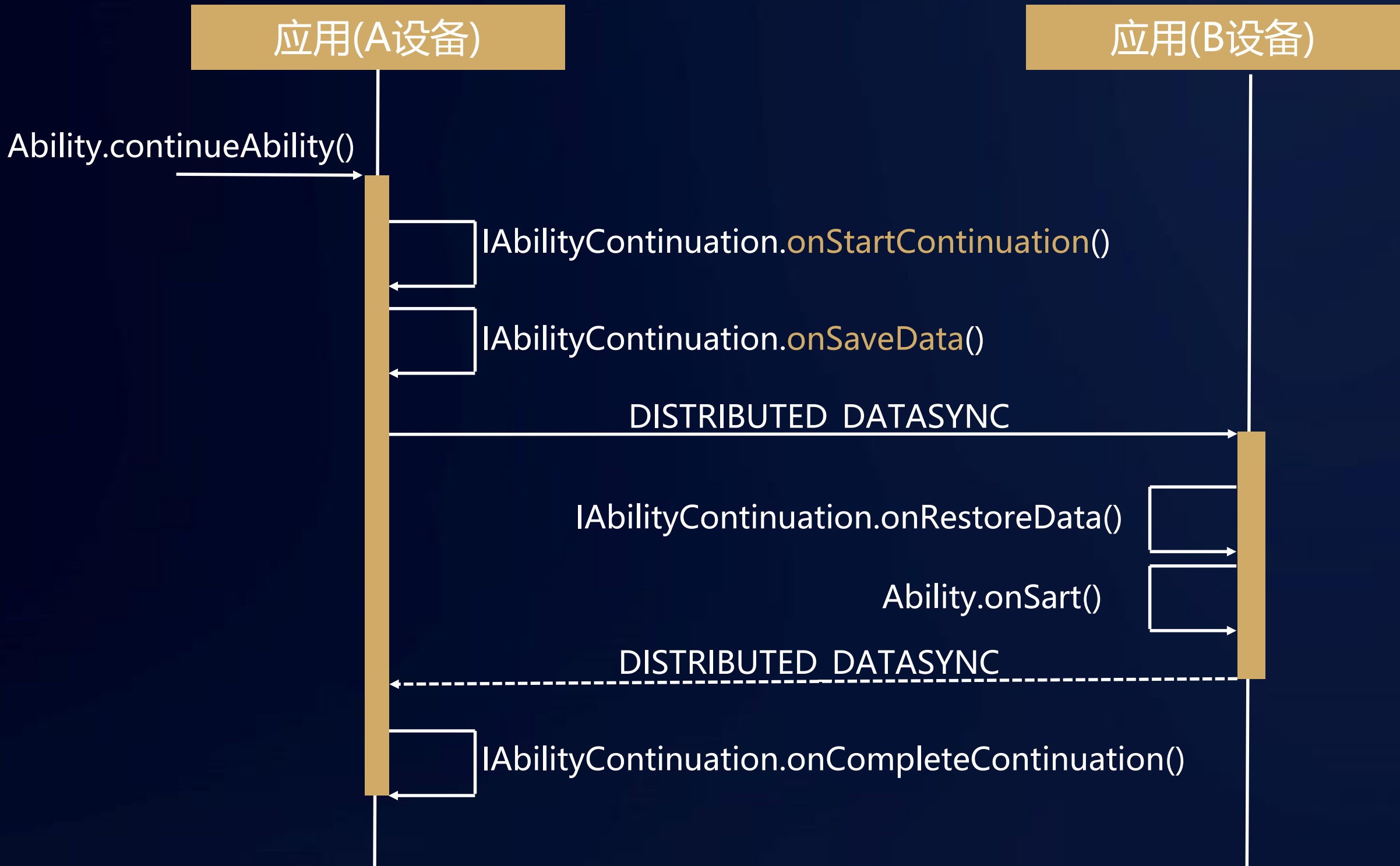


**Step3:**

实现跨端流转



# Step3:实现应用的跨设备流转



## 实现 IAbilityContinuation 接口

接口名称	说明
<code>onStartContinuation()</code>	Page请求迁移后，系统首先回调此方法，开发者可以在此回调中决策当前是否可以执行迁移。
<code>onSaveData()</code>	Page请求迁移后，系统首先回调此方法，开发者可以在此回调中决策当前是否可以执行迁移。
<code>onRestoreData()</code>	源侧设备上Page完成保存数据后，系统在目标侧设备上回调此方法，开发者在此回调中接受用于恢复Page状态的数据。注意，在目标侧设备上的Page会重新启动其生命周期，无论其启动模式如何配置。且系统回调此方法的时机在 <code>onStart()</code> 之前。
<code>onCompleteContinuation()</code>	目标侧设备上恢复数据一旦完成，系统就会在源侧设备上回调Page的此方法，以便通知应用迁移流程已结束。开发者可以在此检查迁移结果是否成功，并在此处理迁移结束的动作，例如，应用可以在迁移完成后终止自身生命周期。

# 申请迁移权限DISTRIBUTED\_DATASYNC

在config.json中  
声明权限

```
"reqPermissions": [  
  {  
    "name": "ohos.permission.DISTRIBUTED_DATASYNC"  
  }  
]
```

在MainAbility.java  
中请求权限

```
if ( verifySelfPermission(SystemPermission.DISTRIBUTED_DATASYNC) ==  
    IBundleManager.PERMISSION_GRANTED ) {  
    this.continueAbility();  
} else {  
    requestPermission(SystemPermission.DISTRIBUTED_DATASYNC);  
}
```

< HDC.Together >

华为开发者大会 2020

# 实现触控事件监听器



触控事件处理

事件监听器: **Component.ClickedListener**

代码示例: **MainAbility.java**

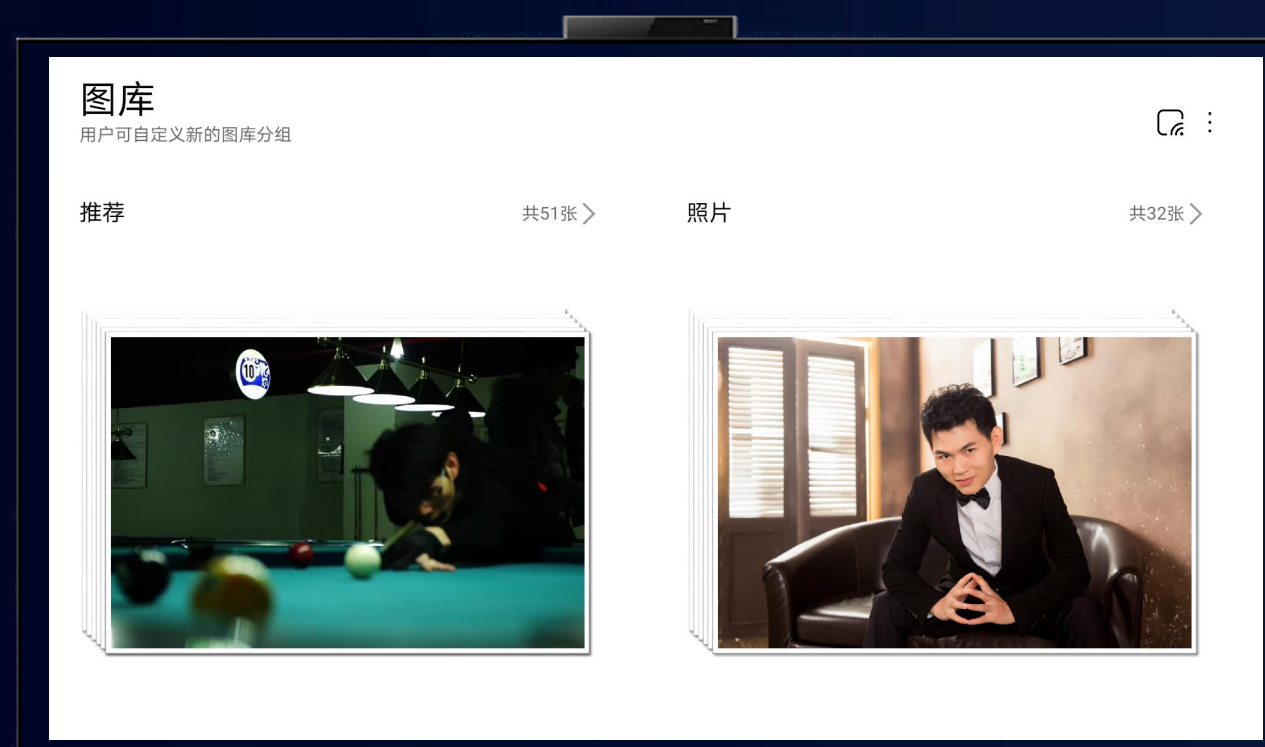
```
@Override
```

```
public void onClick(Component component) {
```

```
    continueAbility();
```

```
}
```

# 实现按键事件监听器



遥控器按键事件处理



事件监听器: **Component. KeyEventListener**

代码示例: **MainAbility.java**

@Override

```
public boolean onKeyEvent(Component component, KeyEvent keyEvent) {  
    if (keyEvent.isKeyDown() && keyEvent.getKeyCode() == KeyEvent.KEY_DPAD_CENTER) {  
        continueAbility();  
        return true;  
    }  
    return false;  
}
```

< **HDC.Together** >

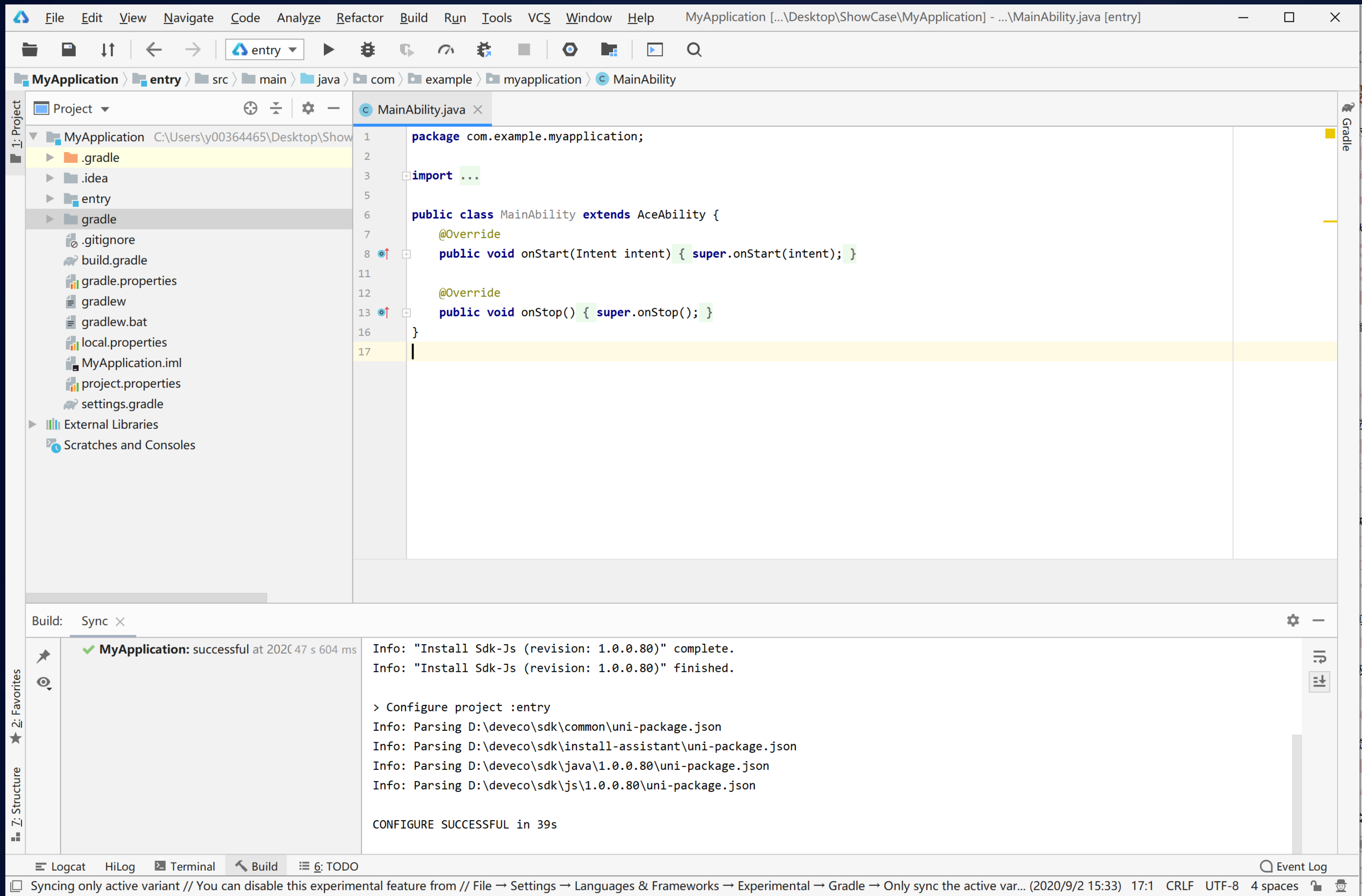
华为开发者大会 2020

样例程序介绍

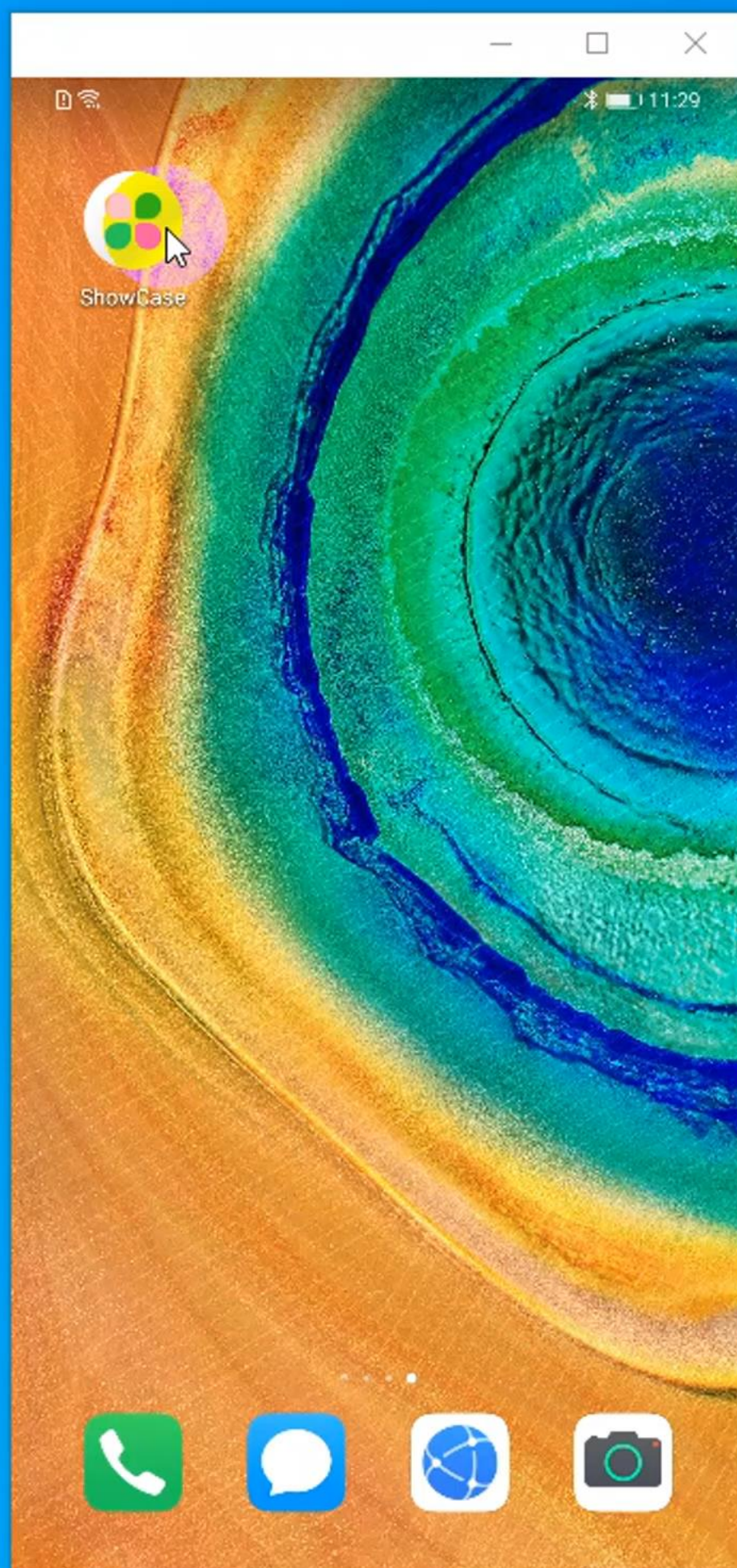
编写第一个HarmonyOS应用

**样例程序演示**

# 样例编码演示









# 总结

- 创建工程：使用DevEco Studio模板创建工程，了解工程的代码结构
- 设计布局：使用xml进行布局描述，AdaptiveBoxlayout实现横竖屏的自适应
- 实现流转：实现使用IAbilityContinuation接口一行代码轻松实现业务跨端流转



欢迎关注HarmonyOS开发者微信公众号