

< HDC.Together >

华为开发者大会 2020

# 调度分布式任务

# 目录

- 分布式调度的功能定位
- 分布式调度的能力范围
- 分布式调度的使用说明

# 分布式调度是什么？

## 自我介绍

分布式调度在超级终端中的定位

# 支持“超级终端”的关键技术与能力



用户体验  
如同使用一个超级设备

三方开发者  
可基于抽象的超级终端开发服务

分布式HarmonyOS



< HDC.Together >

华为开发者大会 2020

## 超级终端的软件能力互助



## 跨设备软件访问的系统服务



## 分布式调度服务在HarmonyOS中的位置



# 分布式调度都能做什么？

## 功能说明

当前分布式调度的能力范围



## 分布式调度能力简介



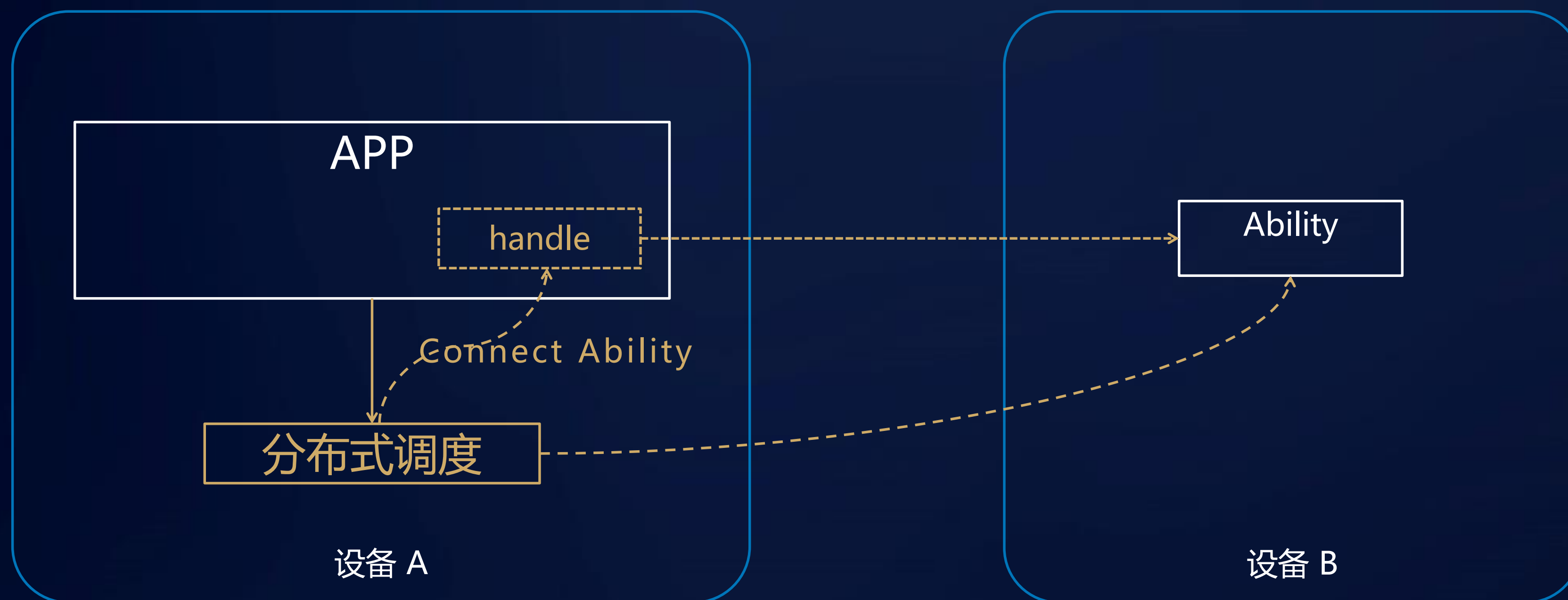
## 寻找可用的软件能力



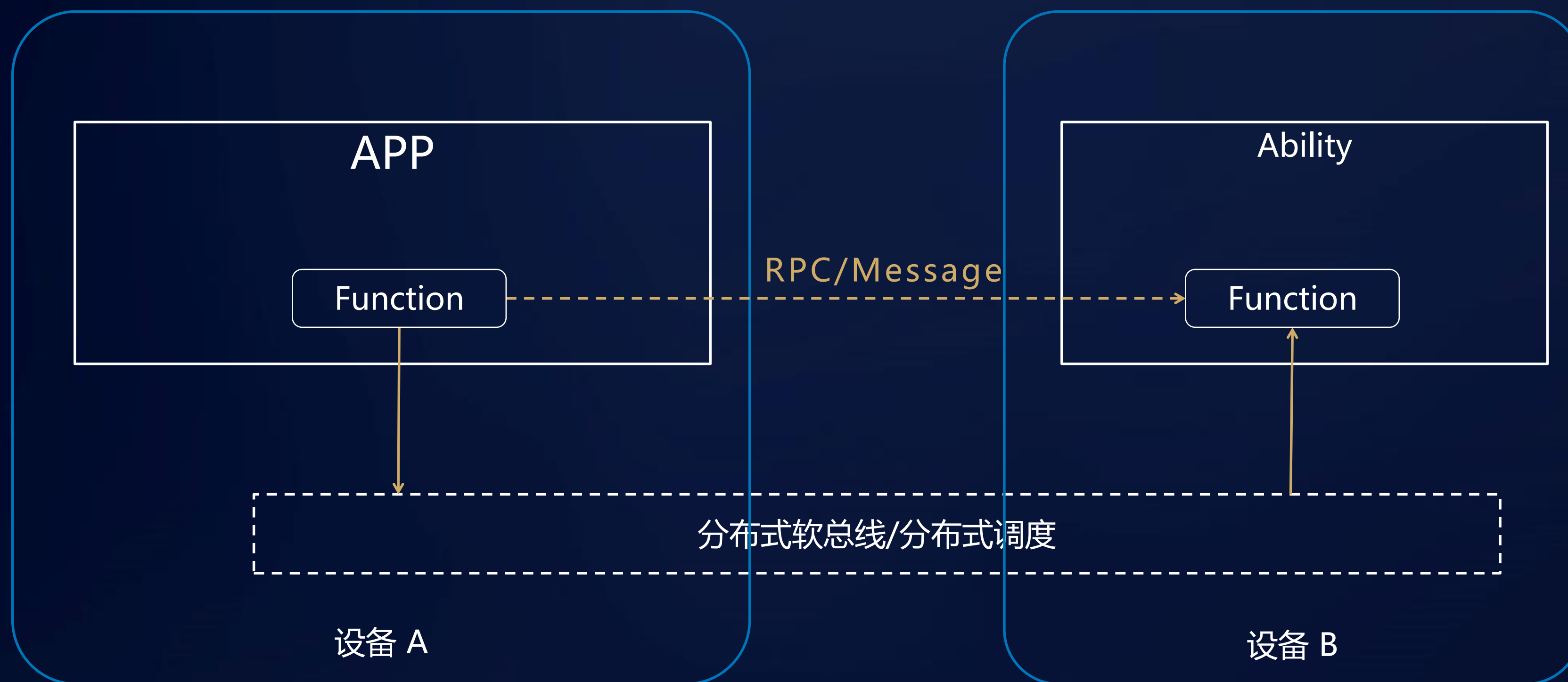
## 启动其他设备上的软件



## 连接/断开其他设备上的软件



## 远程调用其他设备上的软件



## 多层次、全面的安全防护

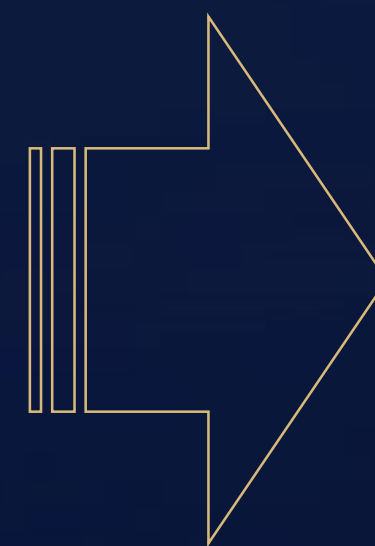


# 如何应用分布式调度能力？

使用方法

介绍分布式应用开发

## 无需等待找到所需要的 Ability



```
public static  
List<AbilityInfo>  
queryRemoteAbilityByIntent  
(Intent intent)
```

分布式调度



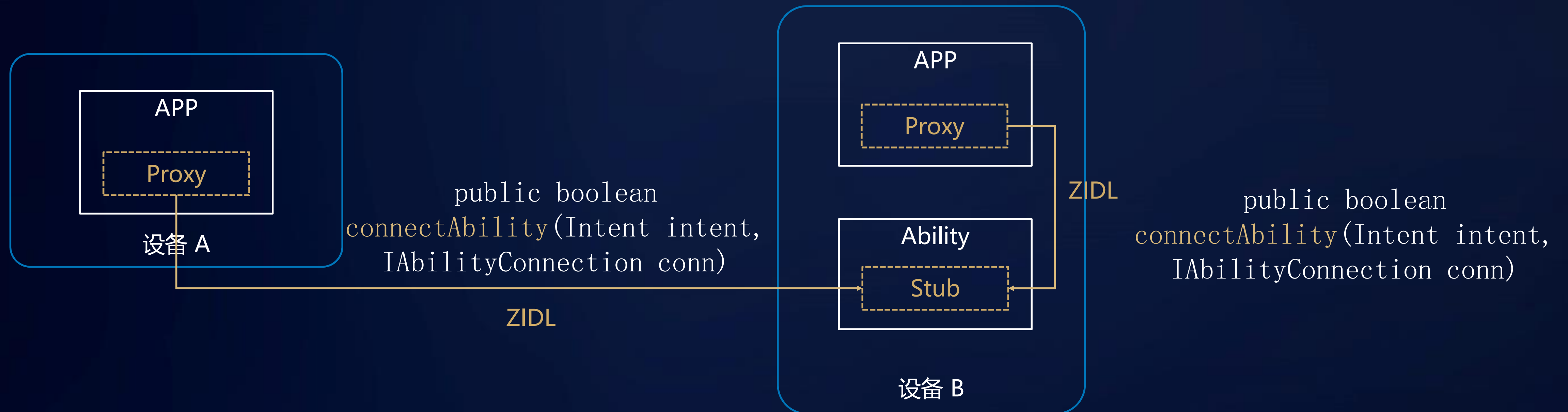
## 一步启动远程 Ability



```
Intent intent = new Intent();
ElementName name = new ElementName(deviceId, "packagename", "classname");
intent.setElement(name);
intent.setFlags(Intent.FLAG_ABILITYSLICE_MULTI_DEVICE);

startAbility(intent);
```

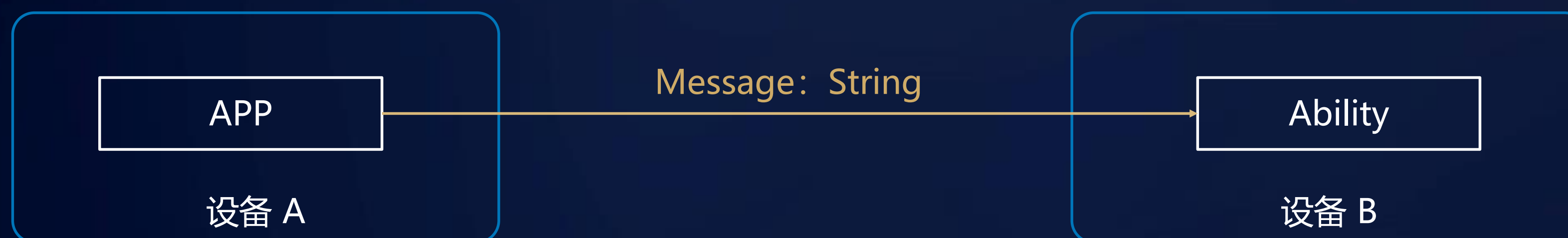
## 像本机一样访问远程 Ability



```
ElementName name = new ElementName(deviceId, "packagename", "classname");
intent.setElement(name);
intent.setFlags(Intent.FLAG_ABILITYSLICE_MULTI_DEVICE);

connectAbility(intent, conn);
```

## 发个消息给远程 Ability



```
p2pClient.send(deviceid, message, sendCallback)
    .addOnSuccessListener(new OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid){};
    })
    .addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(Exception e){};
    });
```



欢迎关注HarmonyOS开发者微信公众号