

# Distributed functionality of



Jing Zhang  
Shanghai Jiao Tong University



# Resources

- OpenHarmony official website: [openharmony.cn](https://openharmony.cn)
- OpenHarmony gitee: [gitee.com/openharmony](https://gitee.com/openharmony)
- OpenHarmony download: [openharmony.cn/download](https://openharmony.cn/download)
- OpenHarmony application samples: [openharmony/applications\\_app\\_samples](https://openharmony/applications_app_samples)
- Daily Build (binary OpenHarmony image & SDK) : [OpenHarmony CI](#)
- DevEco Studio 4.0: [OpenHarmony v4.0 Release \(2023-10-26\)](#)
- DevEco Device Tool 4.0: [device.harmonyos.com/en/develop/ide](https://device.harmonyos.com/en/develop/ide)

# Tutorial Outline

1. Compile and Install Distributed Calculator application
2. Pair two device by Distributed Music Player app
3. Use Distributed Music Player to transfer audio
4. Use Distributed Calculator to share calculate info

# Environment & Marks

- Compile app in your laptop using DevEco Studio (MacOS/Windows)
- Running pre-compile OpenHarmony in our AWS server
- Running commands to manipulate OpenHarmony emulator in qemu

> stands for macOS/Windows

\$ stands for AWS Linux server

@ stands for qemu

# stands for comments

# AWS connection

1. Put `asplos-oh.pem` into `.ssh/config`
2. Find usable Public IPv4 DNS in server list, remember ✓ after choosing one
3. Add following to your `.ssh/config`

```
Host asplos-1
  User admin
  HostName Public IPv4 DNS
  IdentityFile asplos-oh.pem
```

# Get Images

```
# download two images
$ curl -L -o oh-1.7z \
'https://ipads.se.sjtu.edu.cn:1313/f/7d8cad008a0847ba82da/?dl=1'
$ curl -L -o oh-2.7z \
'https://ipads.se.sjtu.edu.cn:1313/f/69f2c5432b9548e3b931/?dl=1'

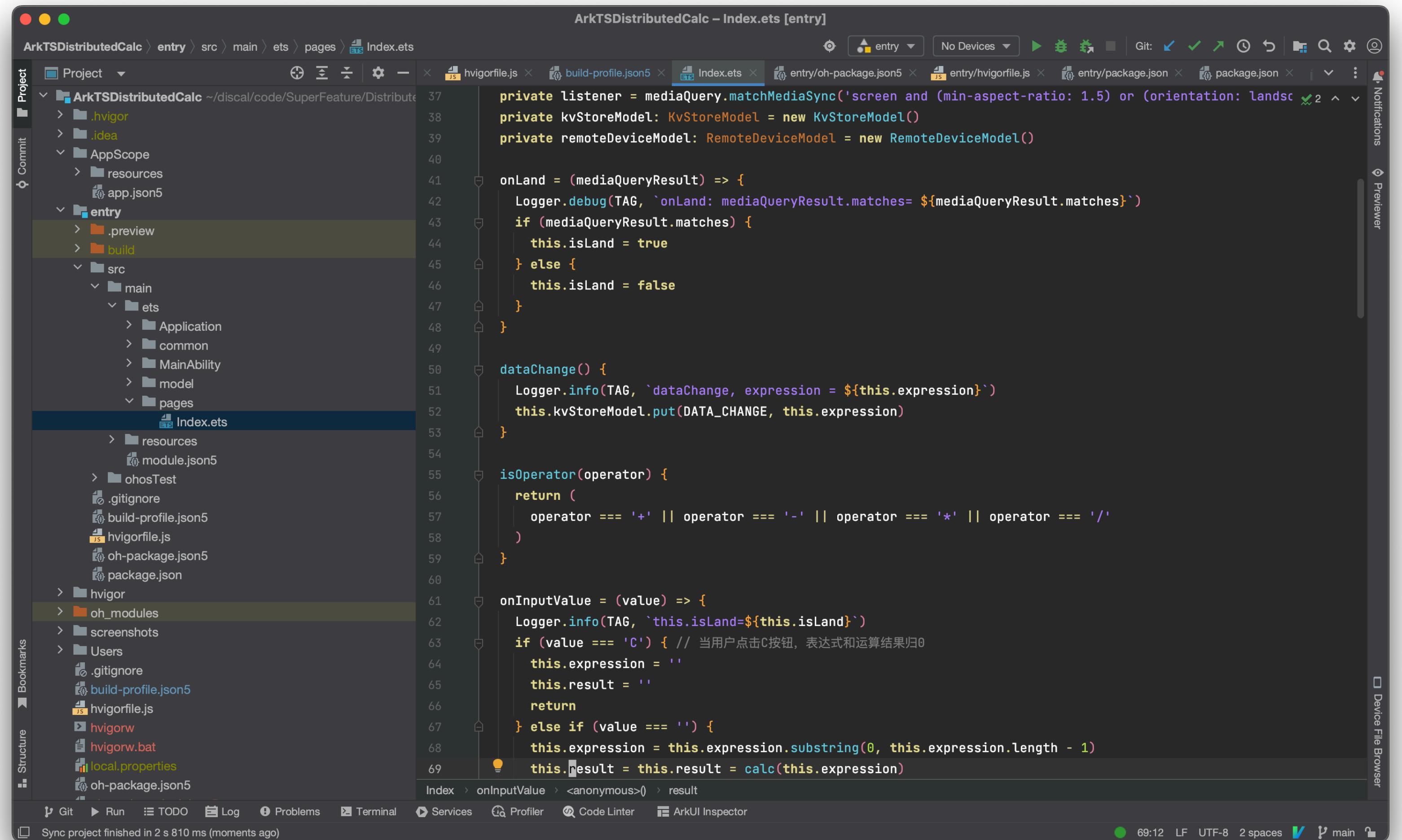
# unzip two packages
$ 7z x oh-1.7z
$ 7z x oh-2.7z
```

# Download DevEco Studio IDE

Win(X64) : [Download](#)

MacOS(X64) : [Download](#)

MacOS(aarch64) :  
[Download](#)



The screenshot shows the DevEco Studio IDE interface with the following details:

- Project View:** Shows the project structure for "ArkTSDistributedCalc". The "entry" folder is expanded, revealing "src", "main", "ets", and "pages" subfolders. "Index.ets" is selected in the tree.
- Code Editor:** The main window displays the content of "Index.ets". The code is written in ETS (Elemental UI Script) and includes logic for handling media queries, managing store models, and performing calculations based on user input.
- Toolbars and Status Bar:** The top bar includes standard IDE icons like file operations, search, and settings. The status bar at the bottom shows "Sync project finished in 2 s 810 ms (moments ago)" and the current time "69:12".

```
private listener = mediaQuery.matchMediaSync('screen and (min-aspect-ratio: 1.5) or (orientation: landscape)')
private kvStoreModel: KvStoreModel = new KvStoreModel()
private remoteDeviceModel: RemoteDeviceModel = new RemoteDeviceModel()

onLand = (mediaQueryResult) => {
  Logger.debug(TAG, `onLand: mediaQueryResult.matches= ${mediaQueryResult.matches}`)
  if (mediaQueryResult.matches) {
    this.isLand = true
  } else {
    this.isLand = false
  }
}

dataChange() {
  Logger.info(TAG, `dataChange, expression = ${this.expression}`)
  this.kvStoreModel.put(DATA_CHANGE, this.expression)
}

isOperator(operator) {
  return (
    operator === '+' || operator === '-' || operator === '*' || operator === '/'
  )
}

onInputValue = (value) => {
  Logger.info(TAG, `this.isLand=${this.isLand}`)
  if (value === 'C') { // 当用户点击C按钮, 表达式和运算结果归0
    this.expression = ''
    this.result = ''
    return
  } else if (value === '') {
    this.expression = this.expression.substring(0, this.expression.length - 1)
    this.result = this.result = calc(this.expression)
  }
}
```

# Download Full-SDK

- Full SDK contains more system API than standard SDK
- We need to use full sdk to build distributed application

Win: [Download](#)

MacOS(aarch64): [Download](#)

MacOS(x86): [Download](#)

9  
└── ets  
└── js  
└── native  
└── previewer  
└── toolchains

# Replace Full-SDK in DevEco

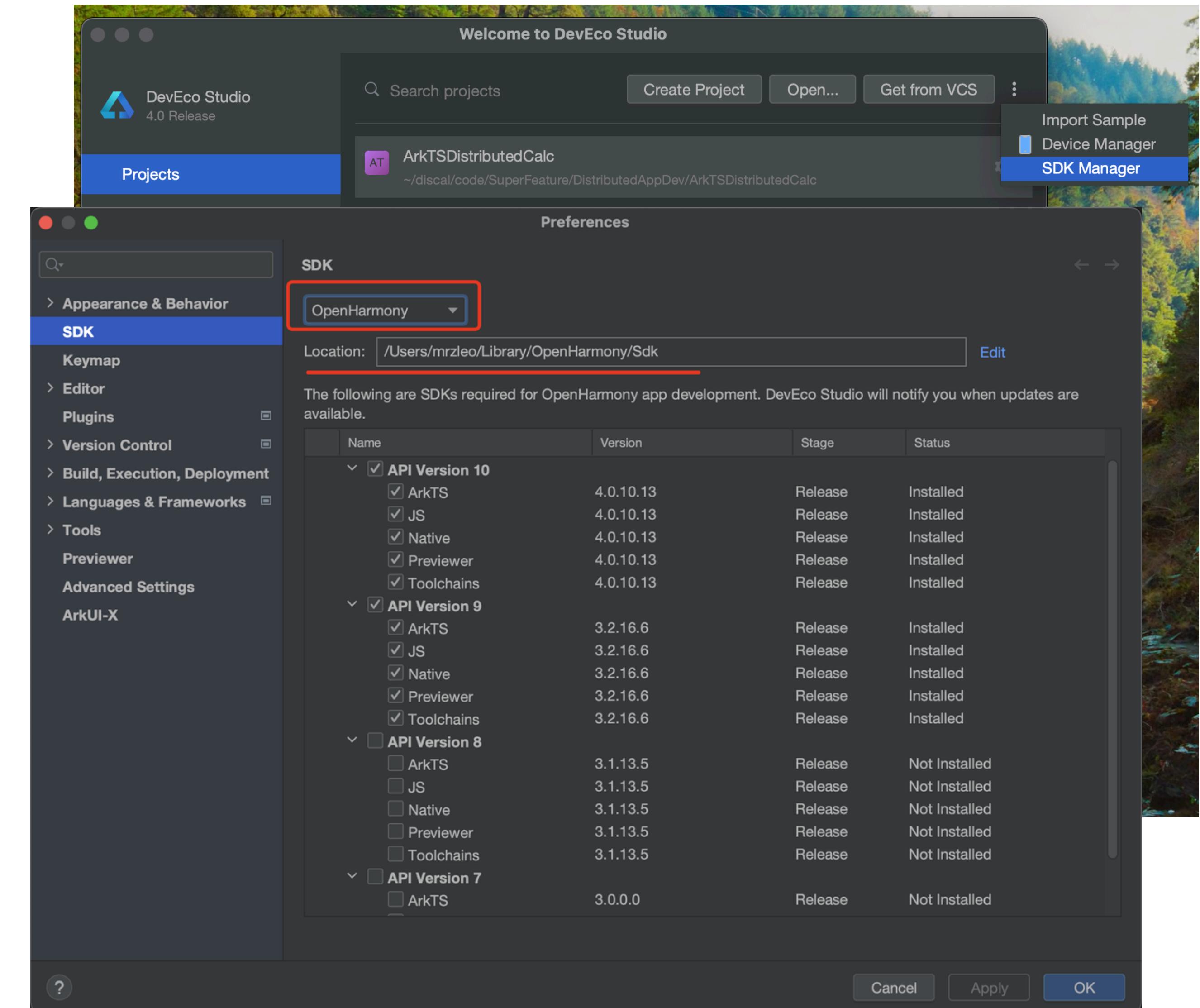
1. Click `SDK Manager` get settings

2. Find sdk path of DevEco

- “OpenHarmony”
- “Location”
- If no path, click Edit and “next”

3. Replace SDK 9:

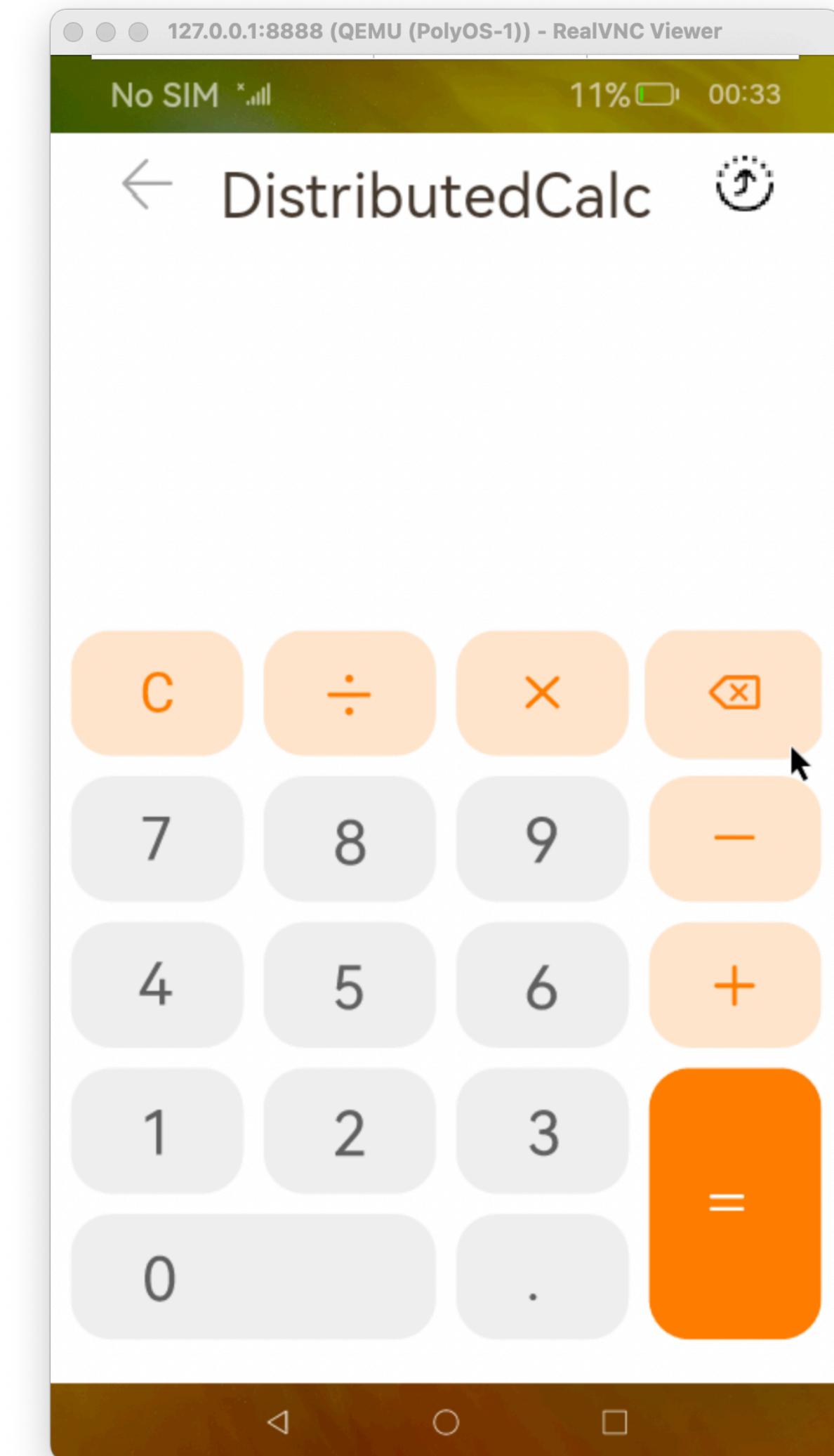
.../OpenHarmony/Sdk/9



# Distributed Calculator Demo

- Distributed Calculator that allows you share calculate information between two devices

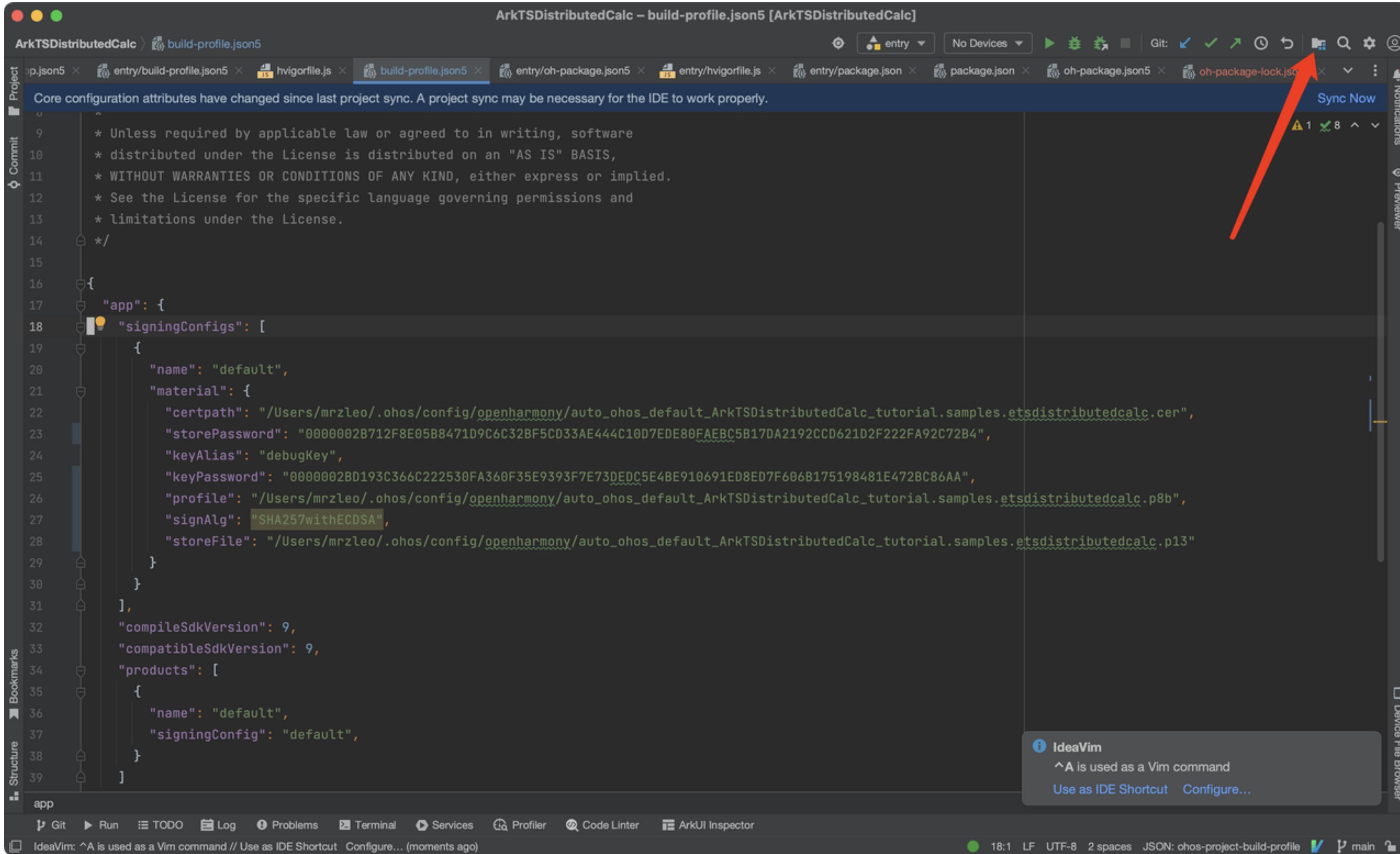
```
# download source code  
> git clone https://github.com/  
openharmony-research/distributedcalc-demo
```



# Compile Distributed Calculator app

1. Generate signature
2. Compile to *Hap*
3. Install app by hdc

# Generate signature



The screenshot shows an IDE interface with a dark theme. The main window displays a JSON configuration file named `build-profile.json5`. The file contains settings for app signing, including certificate paths, store passwords, key aliases, and password profiles. A red arrow points to the `Sync Now` button in the top right corner of the IDE header.

```
ArkTSDistributedCalc - build-profile.json5 [ArkTSDistributedCalc]
ArkTSDistributedCalc > build-profile.json5
Project build-profile.json5 entry/build-profile.json5 h vigorfile.js build-profile.json5 entry/oh-package.json5 entry/h vigorfile.js entry/package.json package.json oh-package.json oh-package-lock.json
Core configuration attributes have changed since last project sync. A project sync may be necessary for the IDE to work properly.
{
  "app": {
    "signingConfigs": [
      {
        "name": "default",
        "material": {
          "certpath": "/Users/mrzleo/.ohos/config/openharmony/auto_ohos_default_ArkTSDistributedCalcTutorial.samples.etsdistributedcalc.cer",
          "storePassword": "0000002B712F8E05B8471D9C6C32BF5CD33AE444C10D7EDE80FAEBC5B17DA2192CCD621D2F222FA92C72B4",
          "keyAlias": "debugKey",
          "keyPassword": "0000002BD193C366C222530FA360F35E9393F7E73DEDCE5E4BE910691ED8ED7F606B175198481E472BC86AA",
          "profile": "/Users/mrzleo/.ohos/config/openharmony/auto_ohos_default_ArkTSDistributedCalcTutorial.samples.etsdistributedcalc.p8b",
          "signAlg": "SHA257withECDSA",
          "storeFile": "/Users/mrzleo/.ohos/config/openharmony/auto_ohos_default_ArkTSDistributedCalcTutorial.samples.etsdistributedcalc.p13"
        }
      }
    ],
    "compileSdkVersion": 9,
    "compatibleSdkVersion": 9,
    "products": [
      {
        "name": "default",
        "signingConfig": "default"
      }
    ]
  }
}
```

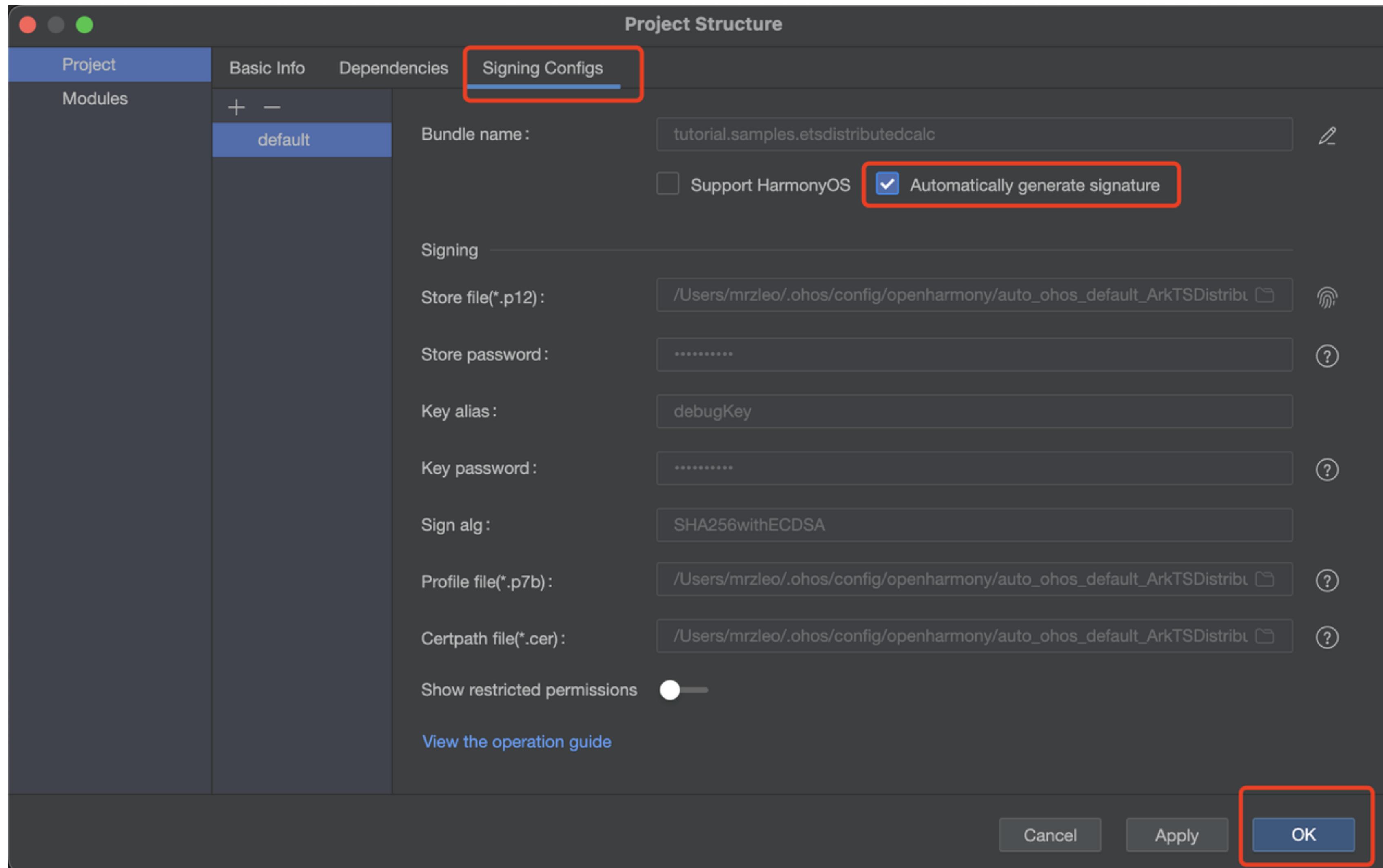
IdeaVim  
^A is used as a Vim command  
Use as IDE Shortcut Configure...

Git Run TODO Log Problems Terminal Services Profiler Code Linter ArkUI Inspector

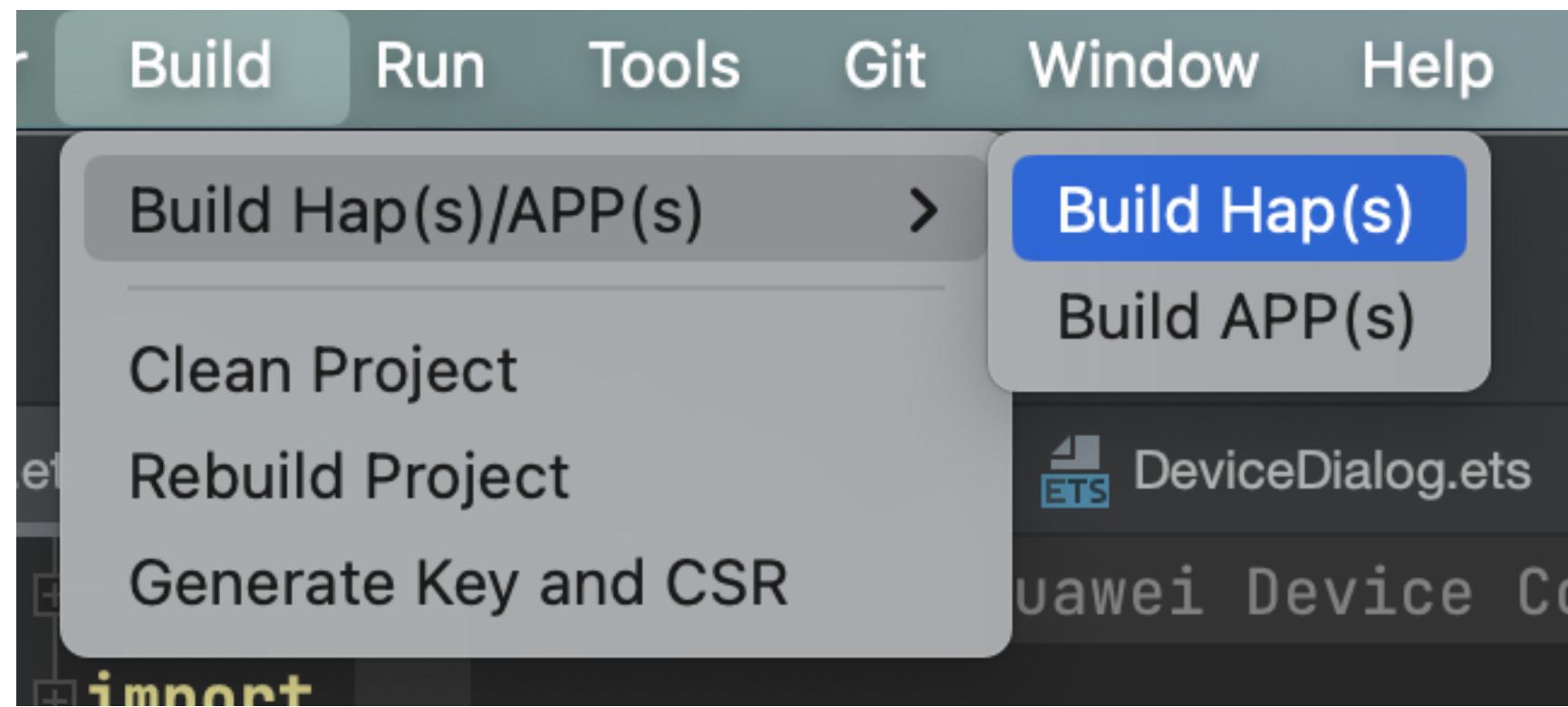
IdeaVim: ^A is used as a Vim command // Use as IDE Shortcut Configure... (moments ago)

18:1 LF UTF-8 2 spaces JSON: ohos-project-build-profile ✓ main

# Generate signature



# Compile to Hap



```
> hvigor UP-TO-DATE :entry:default@MergeProfile...
> hvigor UP-TO-DATE :entry:default@GenerateLoaderJson...
> hvigor Finished :entry:default@BuildNativeWithCmake... after 1 ms
> hvigor UP-TO-DATE :entry:default@MakePackInfo...
> hvigor UP-TO-DATE :entry:default@ProcessProfile...
> hvigor Finished :entry:default@BuildNativeWithNinja... after 1 ms
> hvigor UP-TO-DATE :entry:default@ProcessResource...
> hvigor UP-TO-DATE :entry:default@ProcessLibs...
> hvigor UP-TO-DATE :entry:default@CompileResource...
> hvigor UP-TO-DATE :entry:default@CompileArkTS...
> hvigor Finished :entry:default@BuildJS... after 6 ms
> hvigor UP-TO-DATE :entry:default@PackageHap...
> hvigor UP-TO-DATE :entry:default@SignHap...
> hvigor Finished :entry:assembleHap... after 1 ms
> hvigor BUILD SUCCESSFUL in 230 ms

Process finished with exit code 0
```

```
# Check output
> ls entry/build/default/outputs/default/entry-default-signed.hap
entry/build/default/outputs/default/entry-default-signed.hap
```

# Install App

```
# send to server  
> scp entry/build/default/outputs/default/  
entry-default-signed.hap admin@asplos-1:/  
home/admin/discal.hap
```

```
# boot OpenHarmony emulator, forward ip  
port to localhost using ssh
```

```
> ssh admin@asplos-1 \  
-L 8888:localhost:5920 \  
-L 8889:localhost:5922
```

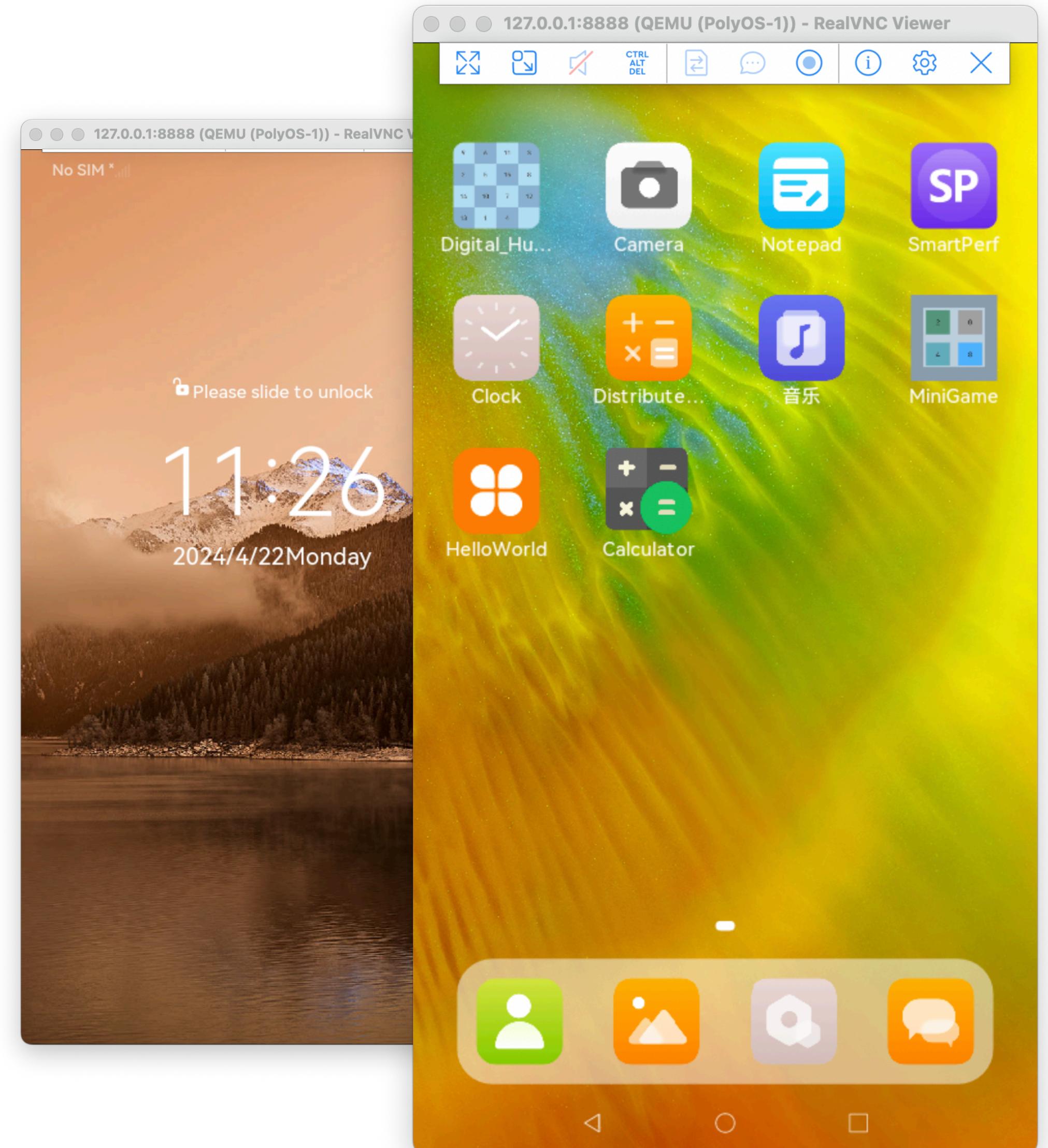
```
$ tmux # 2 windows for 2 emulators
```

```
$ cd ~/oh-1
```

```
$ ./boot.sh
```

```
$ cd ~/oh-2
```

```
$ ./boot.sh
```



# Install in OpenHarmony

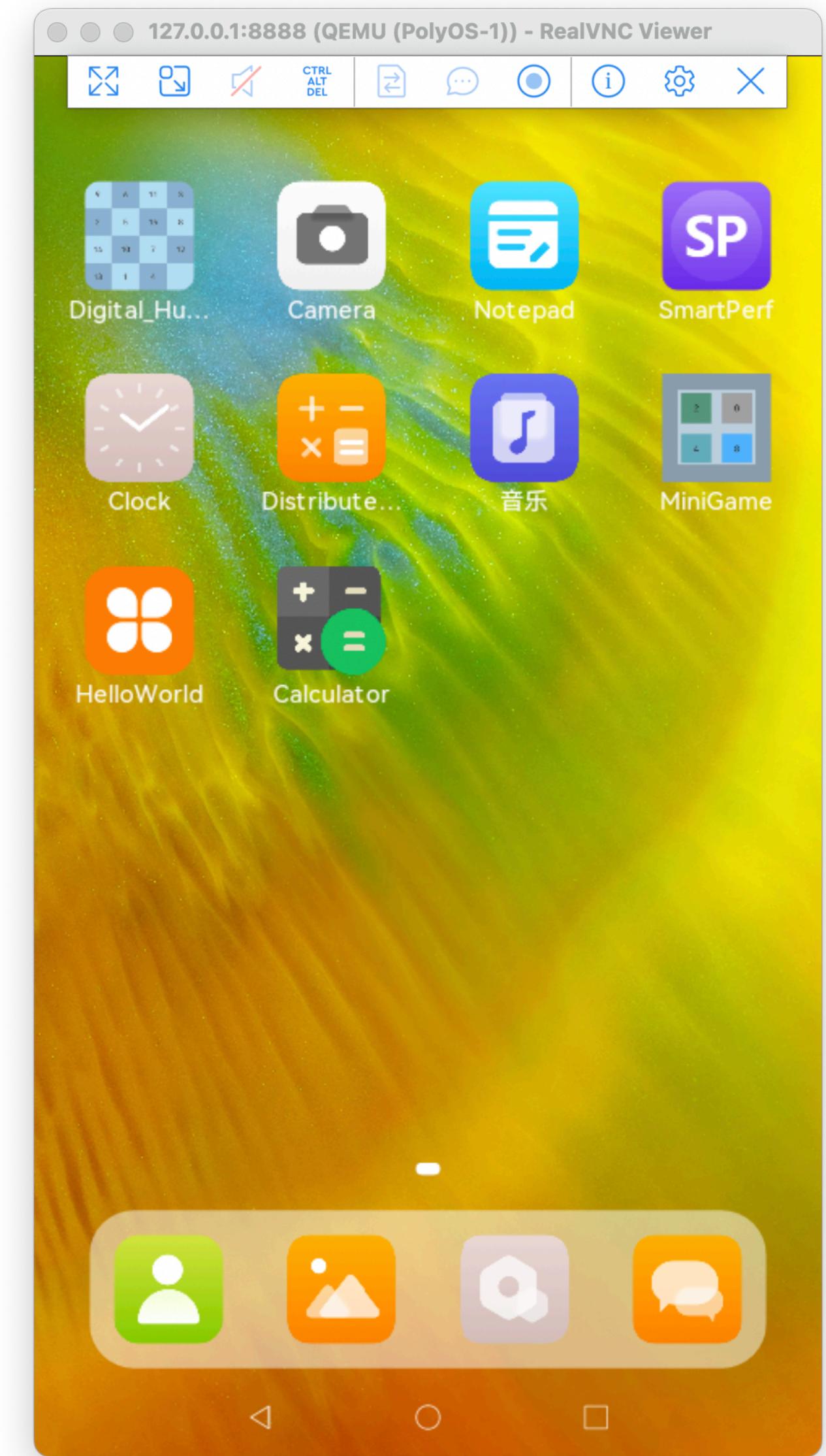
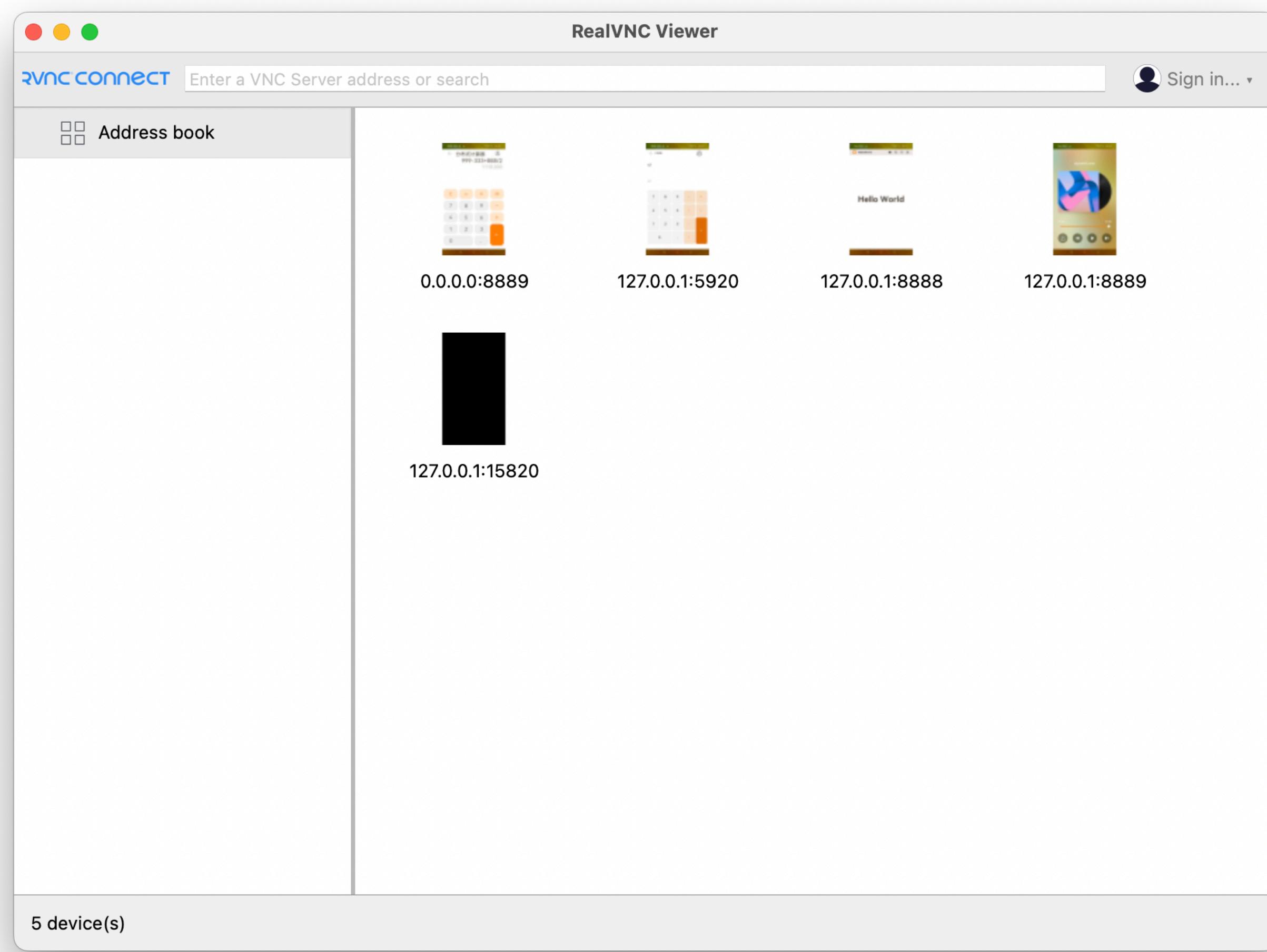
```
# connect to emulator using hdc
$ export OH1=192.168.137.3:5555
$ export OH2=192.168.137.2:5555
$ hdc tconn $OH1
$ hdc tconn $OH2

# install app using hdc
$ hdc -t $OH1 \
    install ~/discal.hap
$ hdc -t $OH2 \
    install ~/discal.hap
```



# Connect to VNC server

```
# use any VNC client to connect to the  
emulator(localhost:8888 and localhost:8889)  
for screen
```



# Settings before start...

```
# connect to oh-1 shell  
$ hdc -t $OH1 shell  
  
# emulate user input, unlock the screen  
@ uinput -T -d 0 0 -m 0 0 0 800 -u 0 800
```

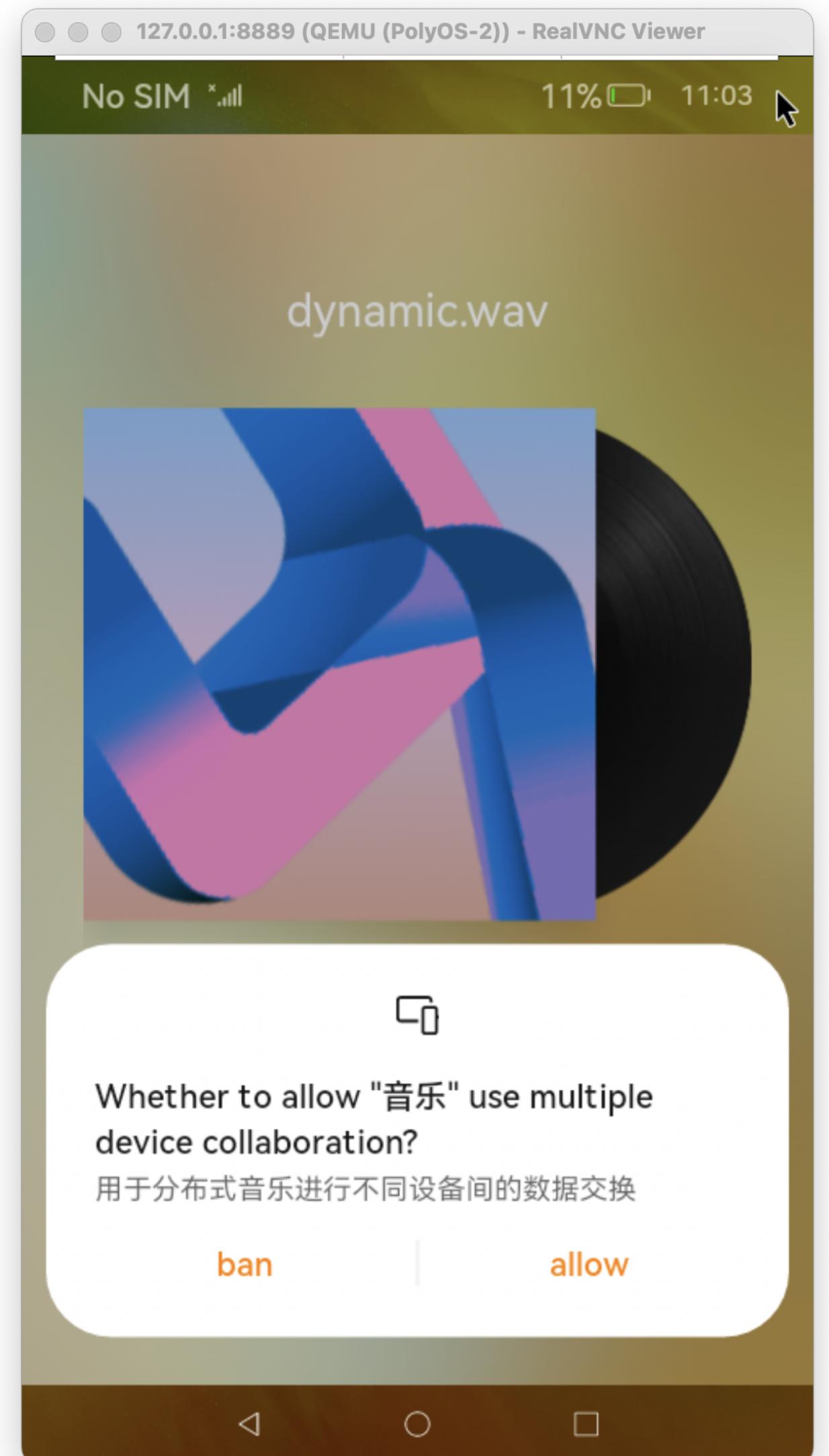
```
# prevent screen sleep  
@ power-shell setmode 602
```

\* do the same for OH2



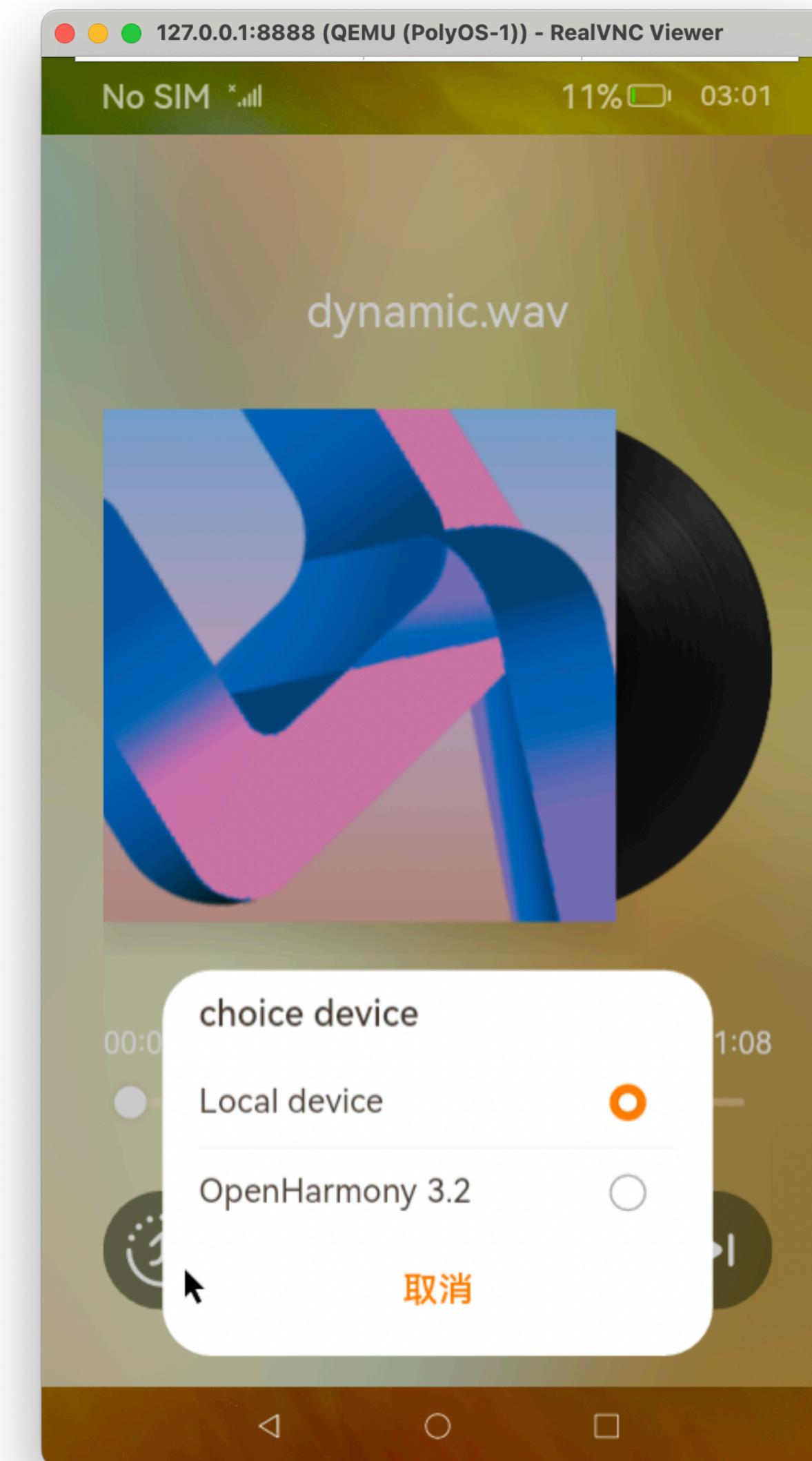
# Run Music Player to pair devices

```
# start Music Player in oh-1  
oh-1@ aa start \  
-a ohos.samples.distributedmusicplayer.MainAbility \  
-b ohos.samples.distributedmusicplayer  
  
# allow permission  
oh-1@ uinput -T -c 350 730
```



# Run Music Player to pair devices

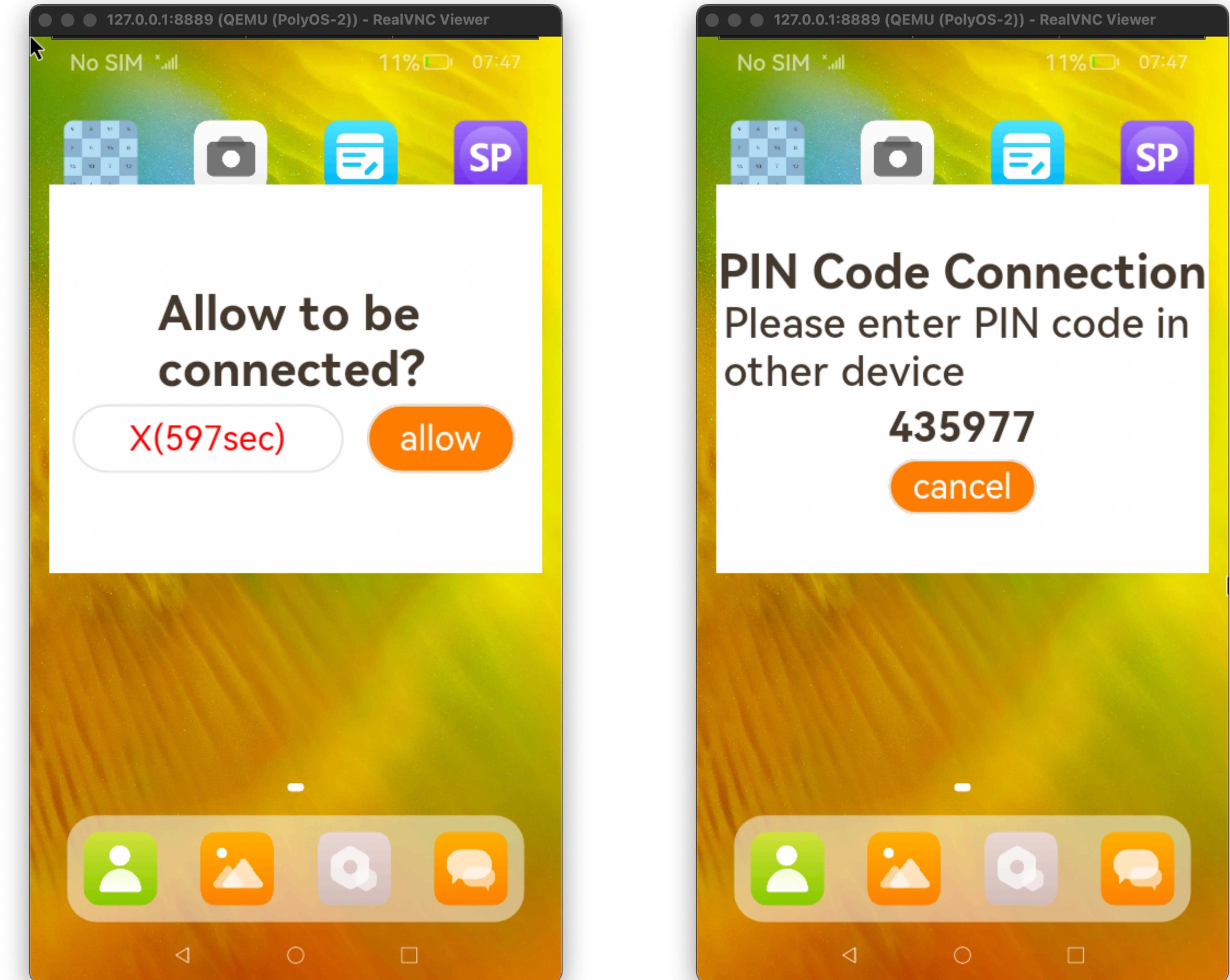
```
# open device list  
oh-1@ uinput -T -c 80 730  
  
# choose remote device  
oh-1@ uinput -T -c 250 700
```



# Run Music Player to pair devices

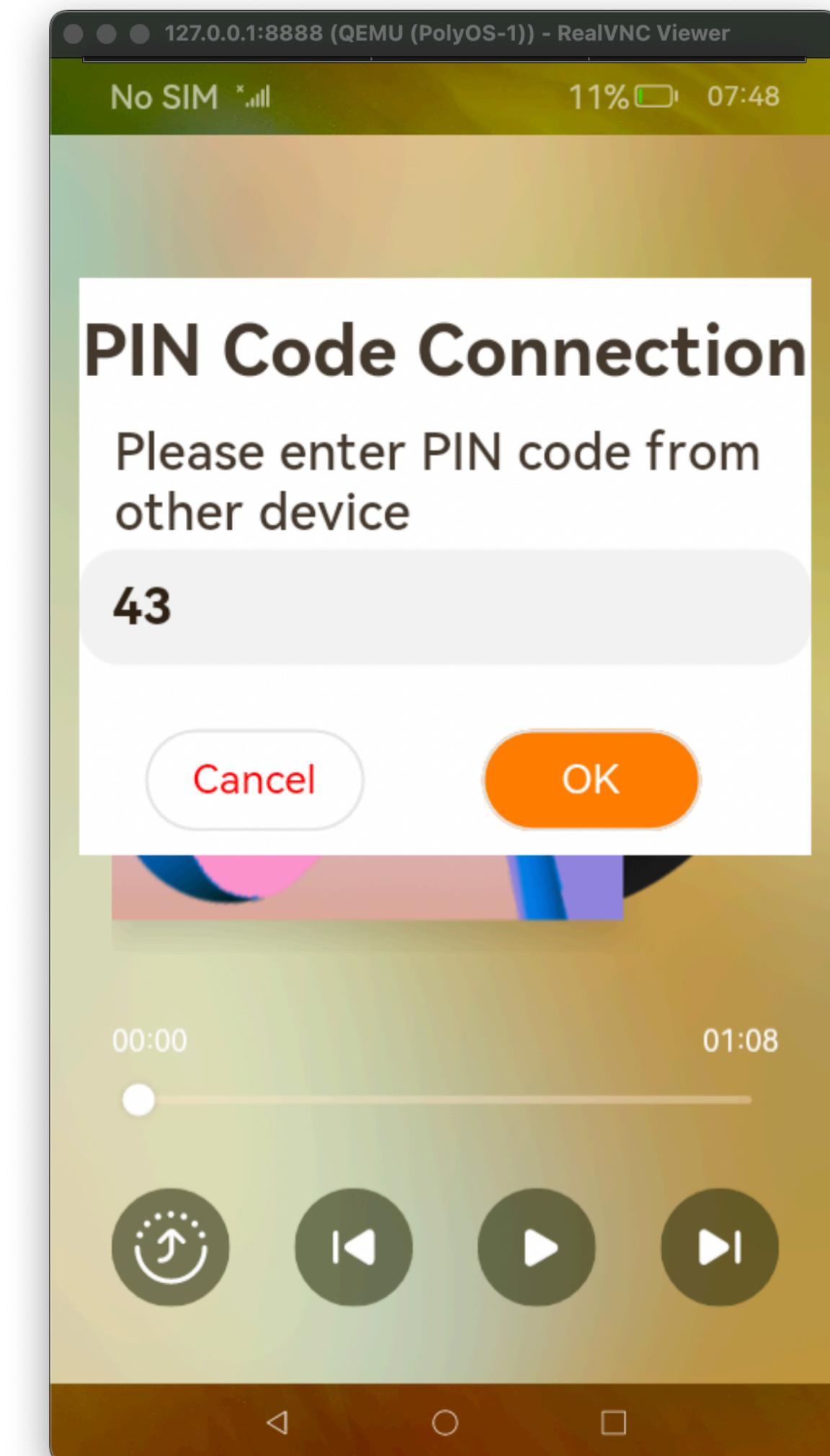
```
# the other device will open  
the pairing request window
```

```
# click allow for PIN code  
oh-2@ uinput -T -c 380 350
```



# Run Music Player to pair devices

```
# enter pin code directly by keyboard  
  
# click yes for confirmation  
oh-1@ uinput -T -c 350 450
```



# Use Distributed Music Player to transfer audio

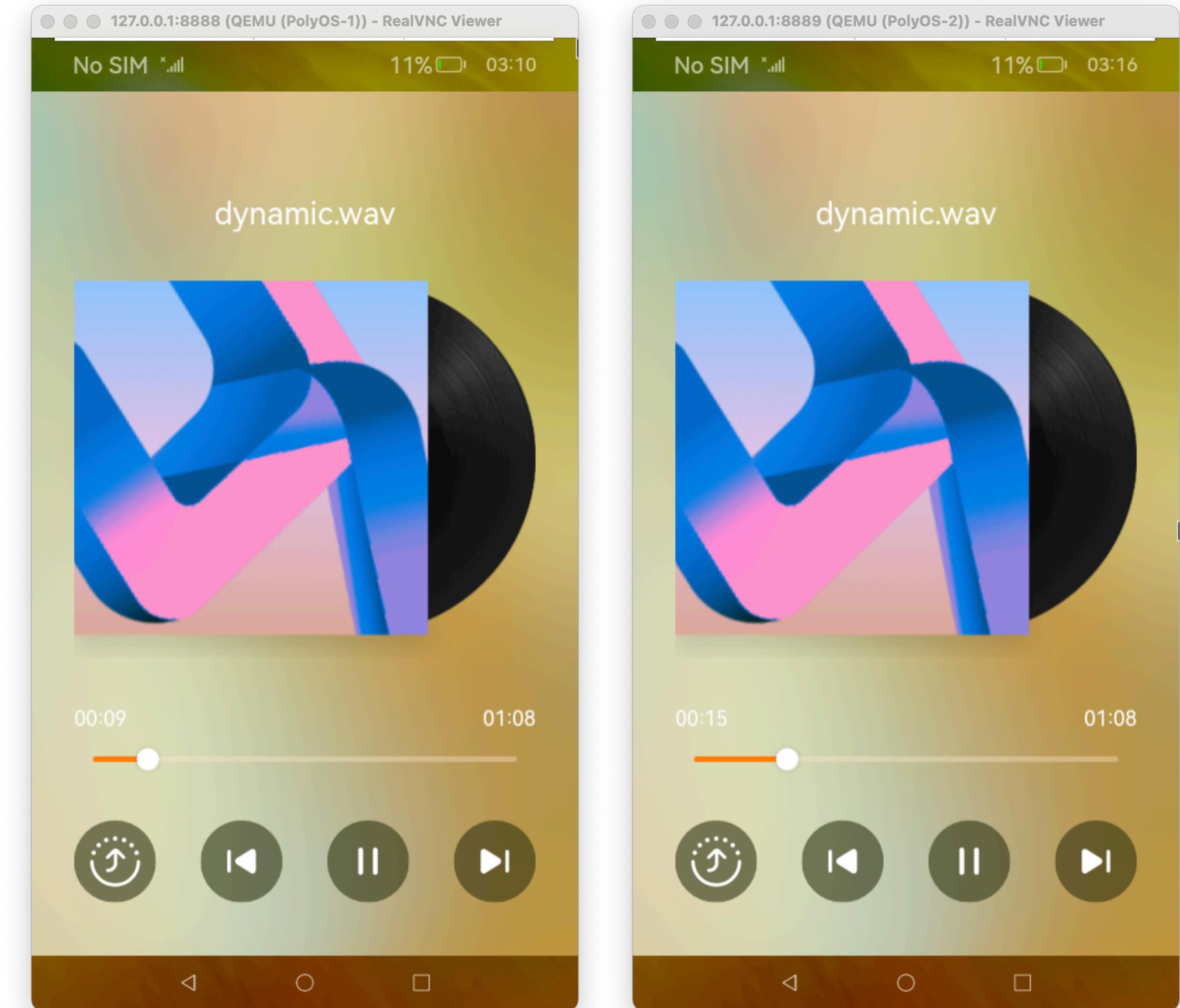
```
# reopen music player in oh-1
oh-1@ aa force-stop \
    ohos.samples.distributedmusicplayer
oh-1@ aa start \
    -a ohos.samples.distributedmusicplayer.MainAbility \
    -b ohos.samples.distributedmusicplayer
```

# Use Distributed Music Player to transfer audio

```
# start playing in oh-1  
oh-1@ uinput -T -c 300 710
```

```
# choose remote device  
oh-1@ uinput -T -c 80 730
```

```
# transfer to oh-2  
oh-1@ uinput -T -c 250 700
```



oh-1

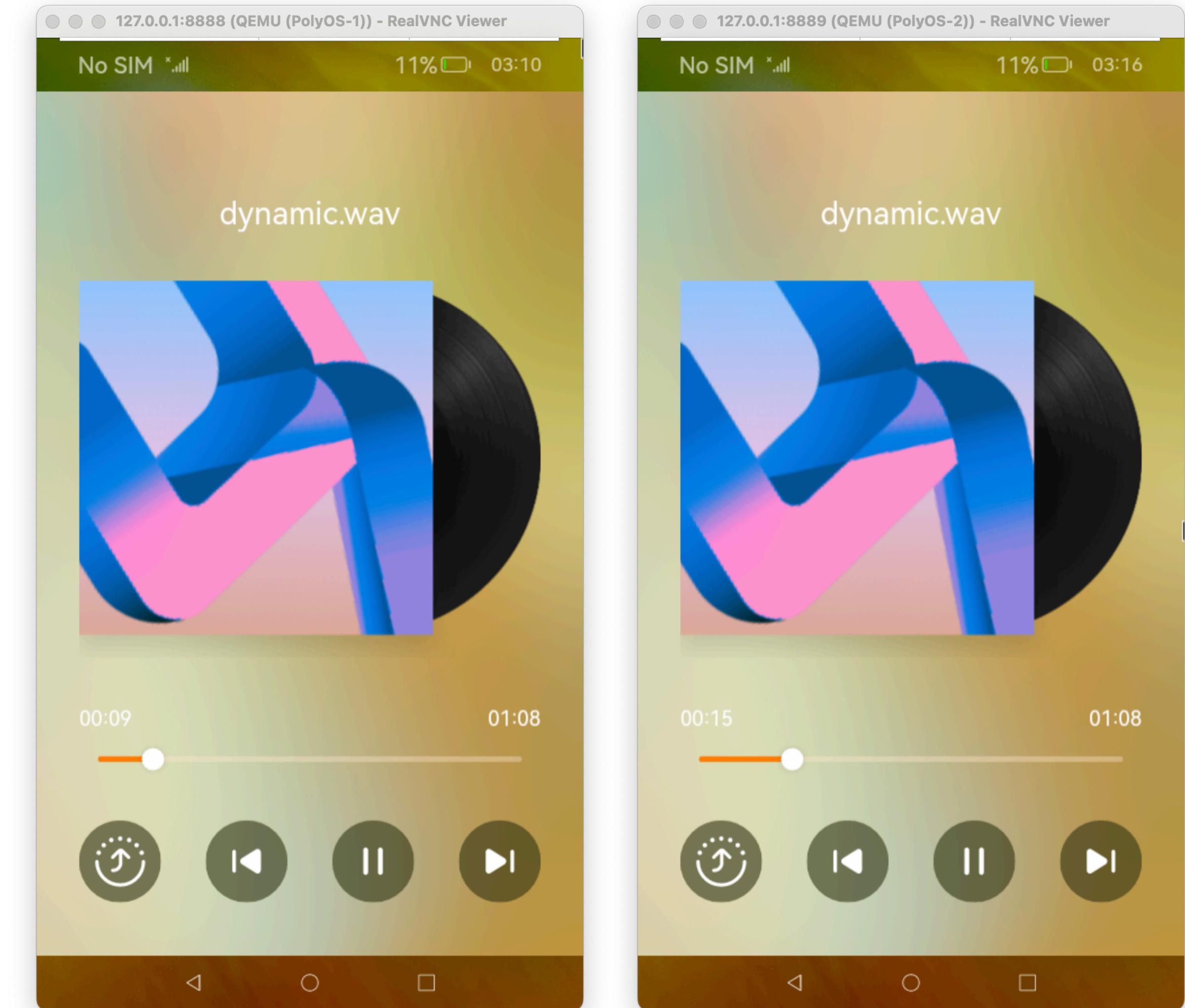
oh-2

# Use Distributed Music Player to transfer audio

```
# music will continue to play in oh-2
```

```
# allow permission  
oh-2@ uinput -T -c 350 730
```

```
# and exit by  
oh-2@ aa force-stop \  
ohos.samples.distributedmusicplayer
```

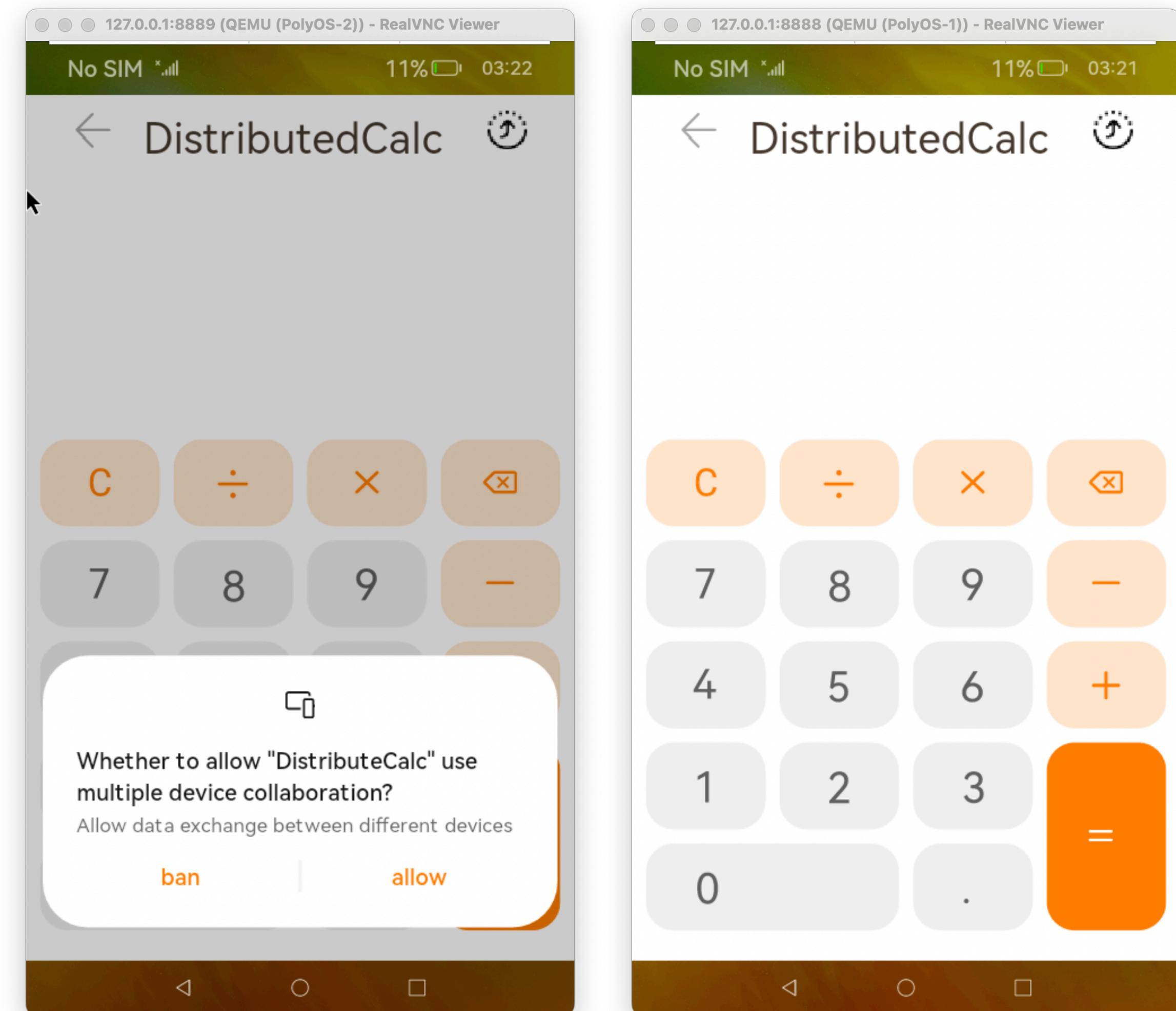


oh-1

oh-2

# Use Distributed Calculator to share calculate info

```
# start distributed calculator in oh-1  
oh-1@ aa start -a MainAbility \  
-b tutorial.samples.etsdistributedcalc  
  
# allow permission  
oh-1@ uinput -T -c 350 740
```



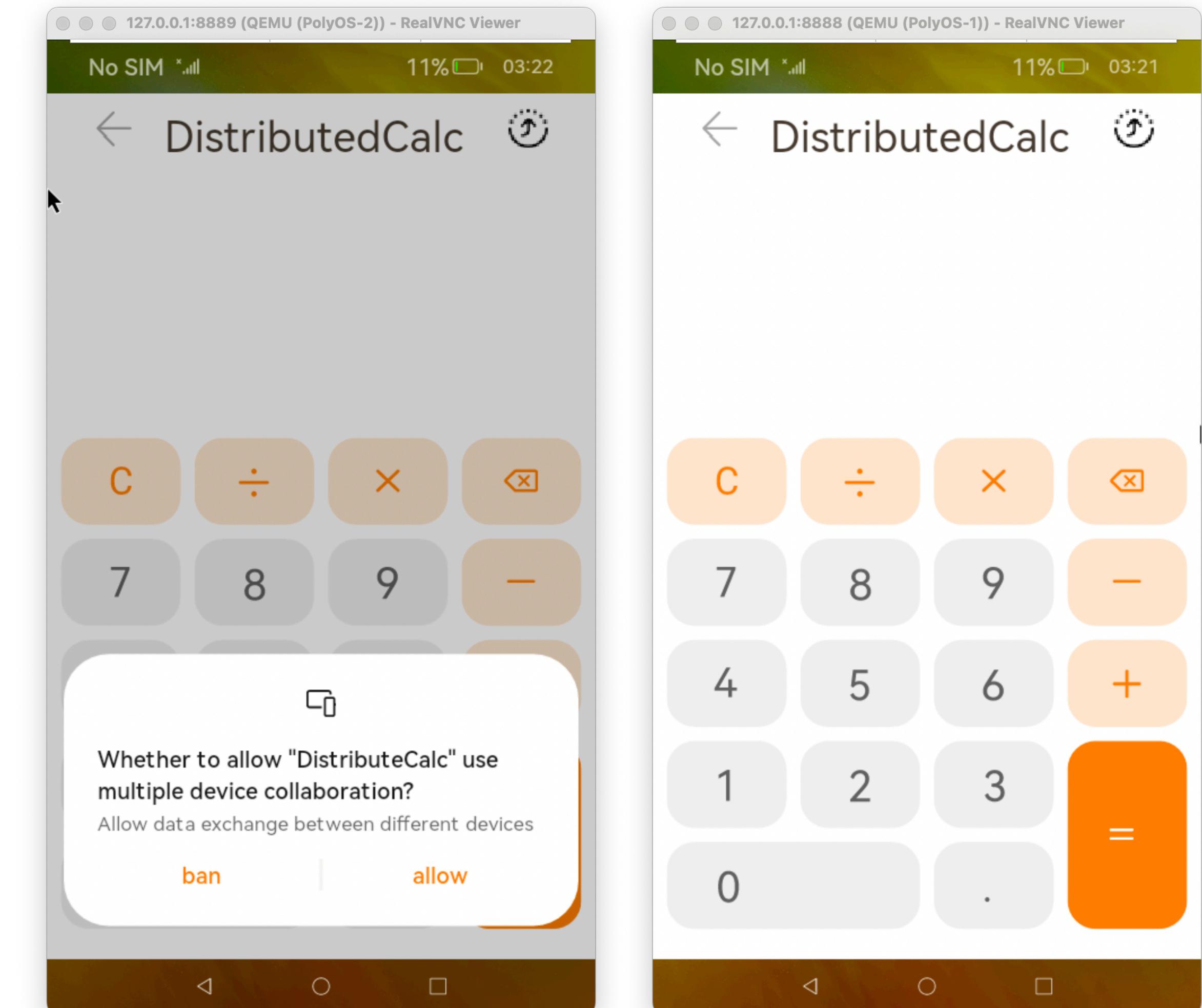
# Use Distributed Calculator to share calculate info

```
# open device list  
oh-1@ uinput -T -c 420 80
```

```
# choose remote device  
oh-1@ uinput -T -c 250 450
```

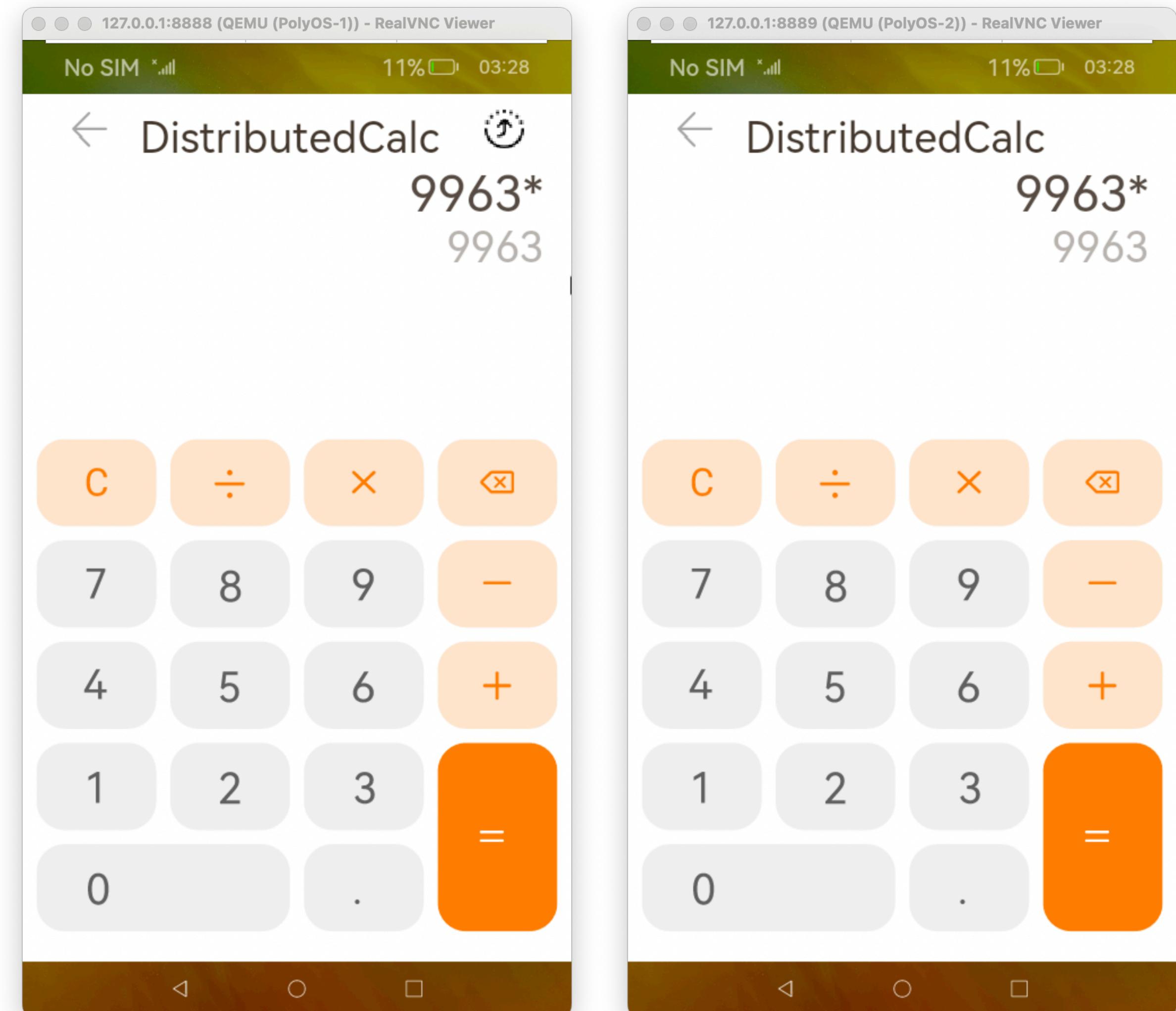
```
# oh-2 will open distributed  
calculator app automatically
```

```
# allow permission  
oh-2@ uinput -T -c 350 740
```



# Use Distributed Calculator to share calculate info

```
# enter some number and operators  
# two devices will sync automatically  
oh-1@ uinput -T -c 300 480 # 9  
oh-1@ uinput -T -c 300 580 # 6  
oh-1@ uinput -T -c 300 680 # 3  
oh-1@ uinput -T -c 300 380 # x
```



# Use Distributed Calculator to share calculate info

```
# stop sharing and remote device will exit the app  
  
# open device list  
oh-1@ uinput -T -c 420 80  
  
# choose localhost to stop sharing  
oh-1@ uinput -T -c 250 400  
  
# exit local app  
oh-1@ aa force-stop tutorial.samples.etsdistributedcalc
```