

# Updates of Android on RISC-V

Mao Han

April. 19<sup>th</sup>, 2022

# Milestone

Basic support  
on RVB-ICE

Full feature Android  
on RISC-V

Upstream inclusion of  
RISC-V support

RISC-V based Android  
certificated product

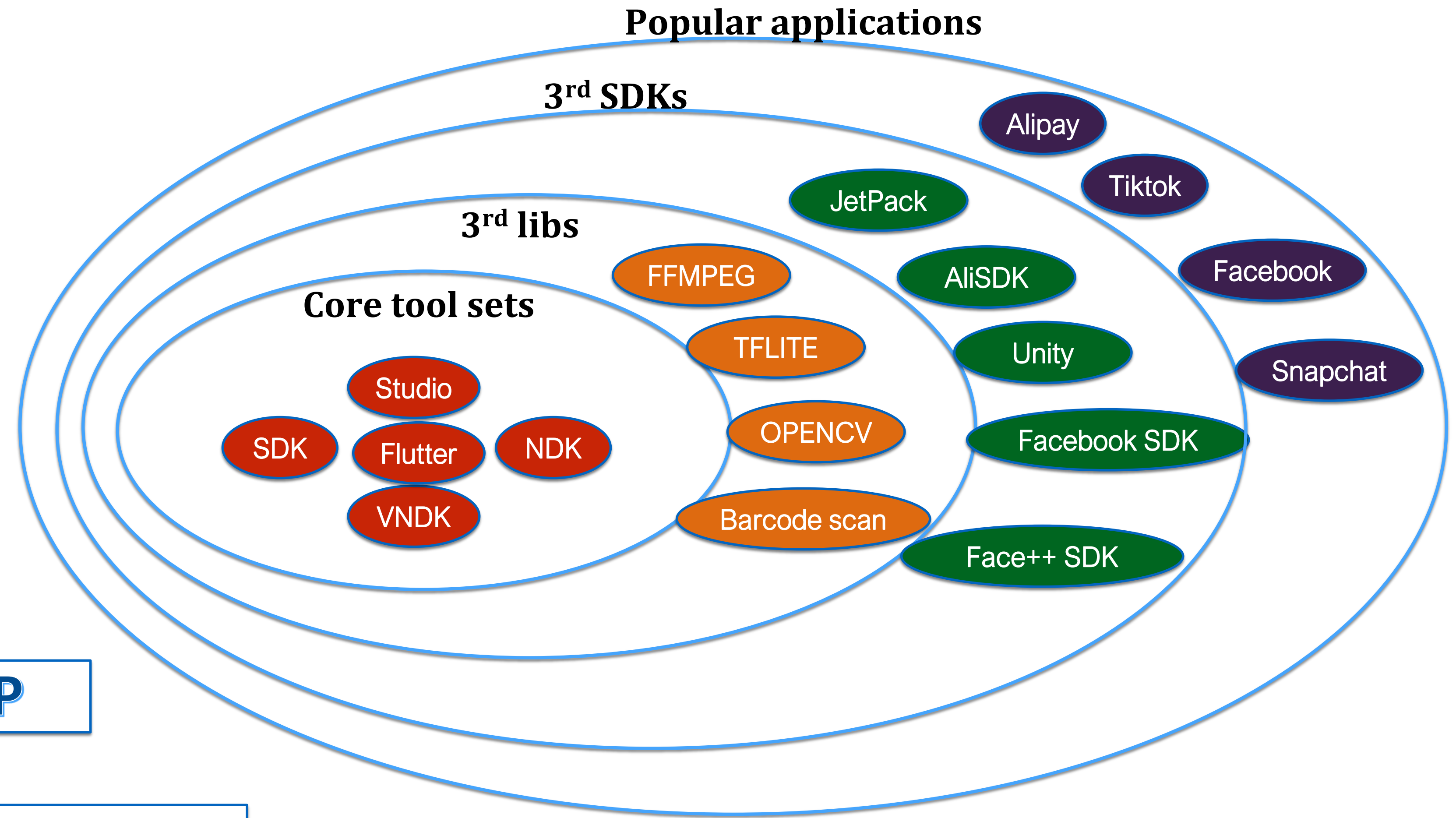
## Challenge

- Poor IP support
- Incompatibility
- Lack of verification

- Poor code quality
- Failing compliant test

- Poor Optimization
- Poor SDK & APP support
- No reference design

# Android Ecosystem

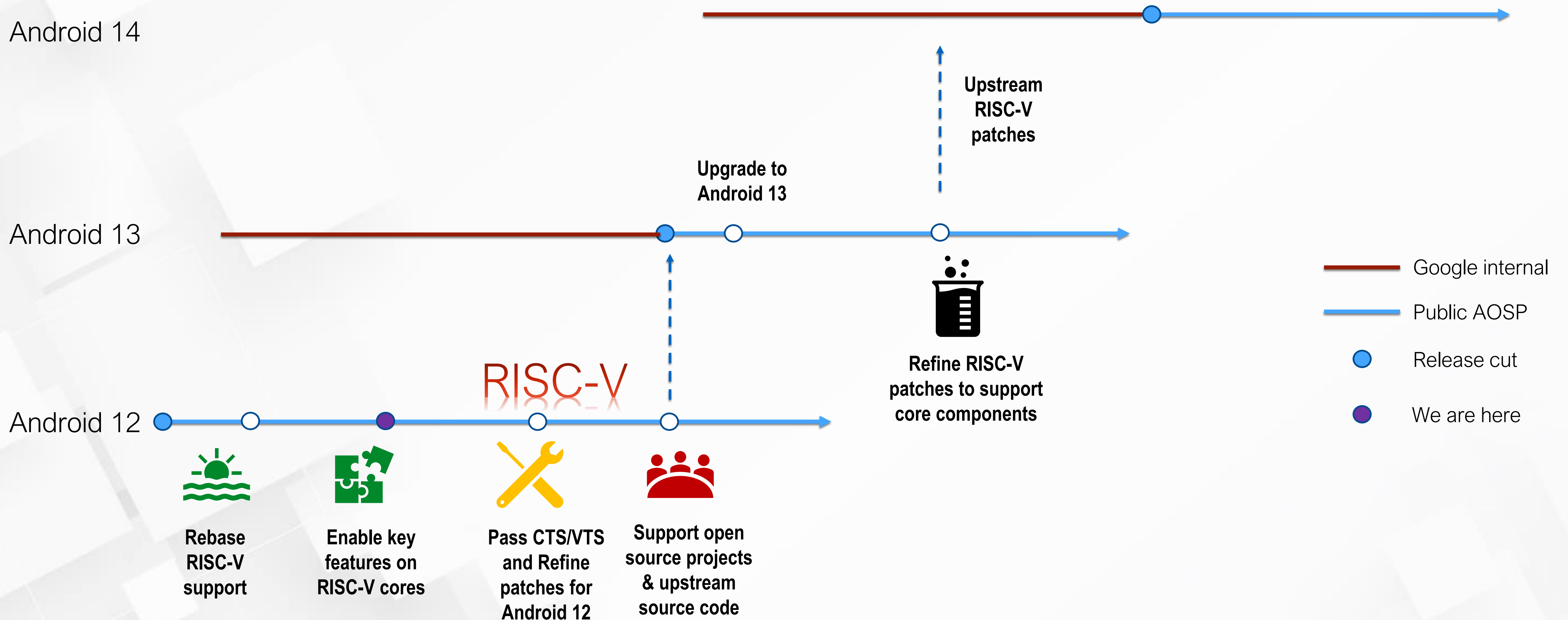


AOSP Android

SoC Android BSP

Device Android BSP

# AOSP Upstream Plan: Why Android 12 NOW



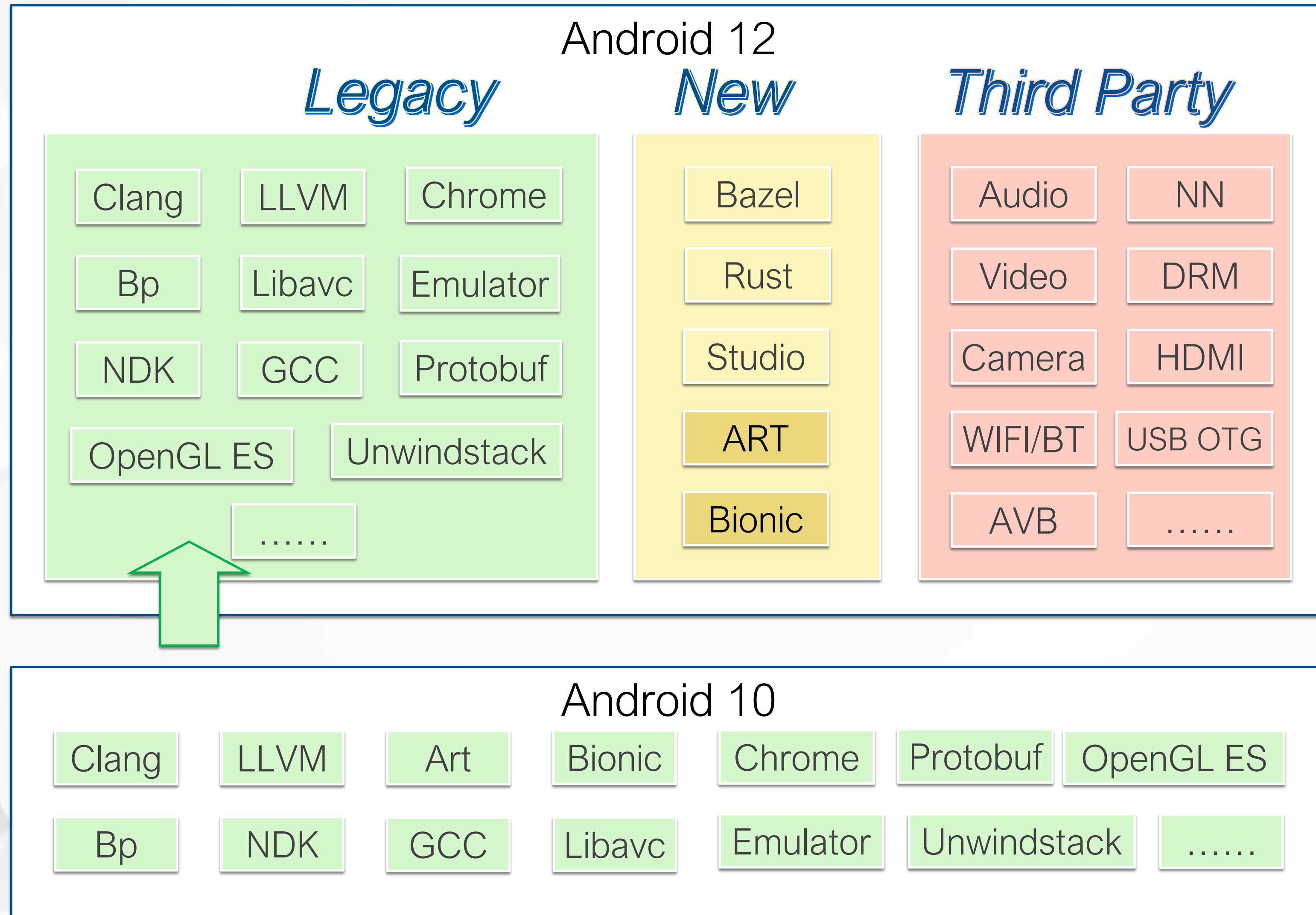
# Hardware Requirements for Android Development

- 2+ Core at *XuanTie* C910 (rv64imafdcv) level
- 4G+ DDR Memory
- GPU (*OpenGL ES*, *OpenCL*)
- Display (MIPI/HDMI)
- USB
- Multi-Channel Audio output & input
- HW Video/Picture codec
- ISP, multiple MIPI lanes
- Neural Network Accelerator





# What We need to Support Android 12 on RISC-V



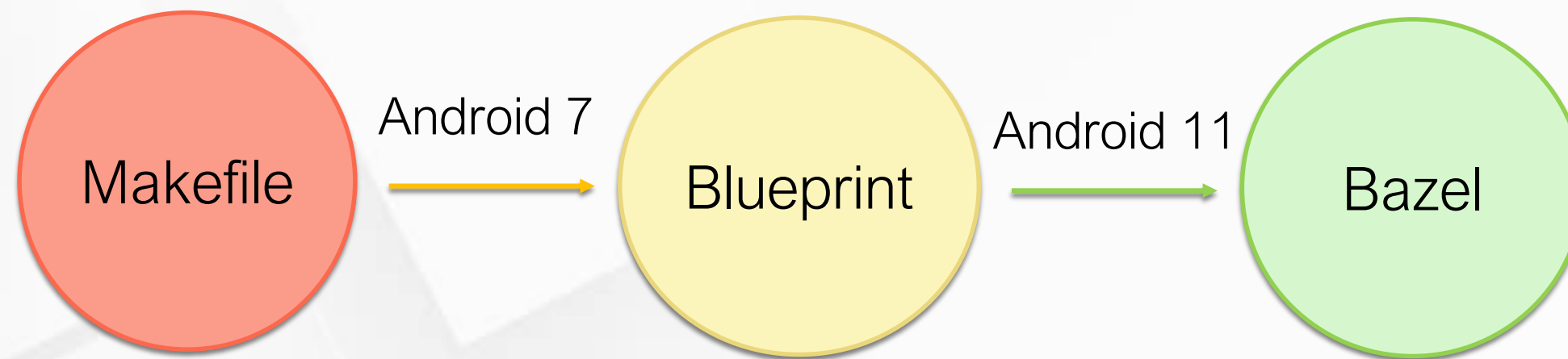
Android 12 on RISC-V

# New Features

## Build system

- Slow
- Inflexible
- Complex

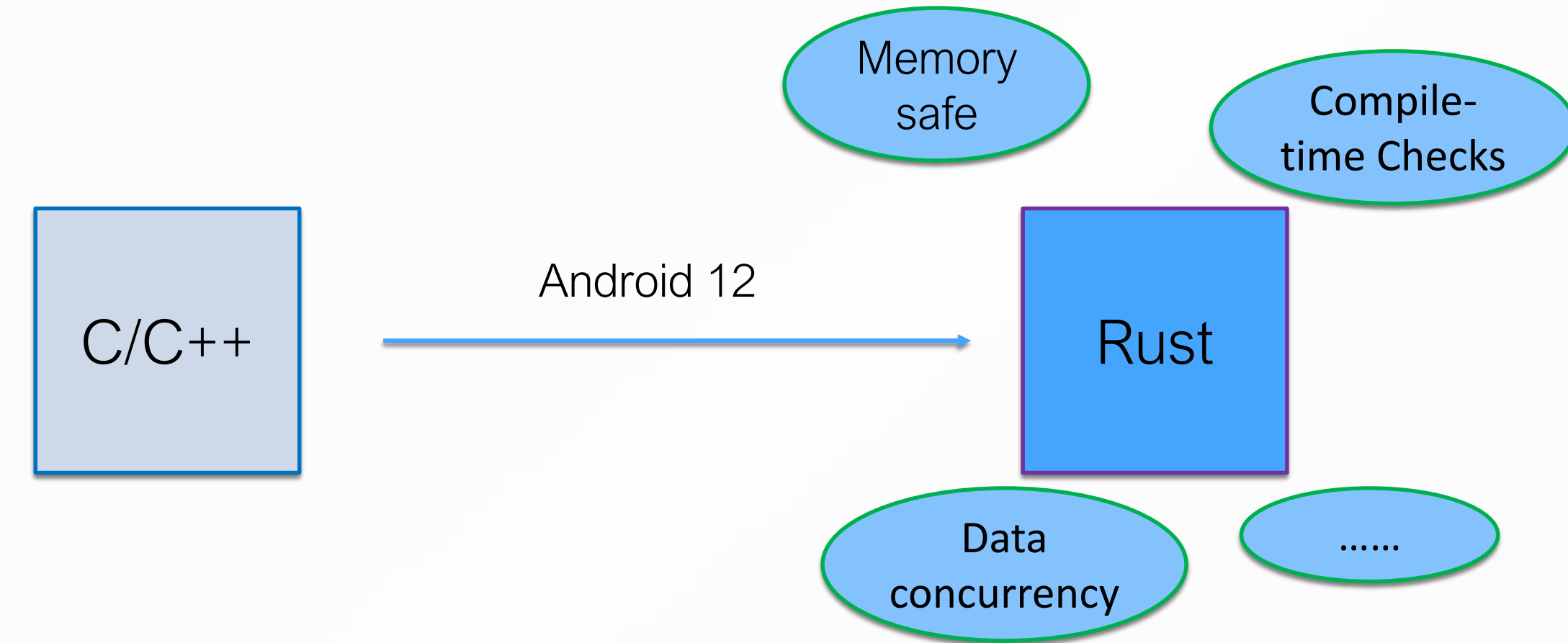
- Faster
- Flexible
- Simpler



RISC-V support:

- Configurability
- Setting & constraints

## Rust



RISC-V support:

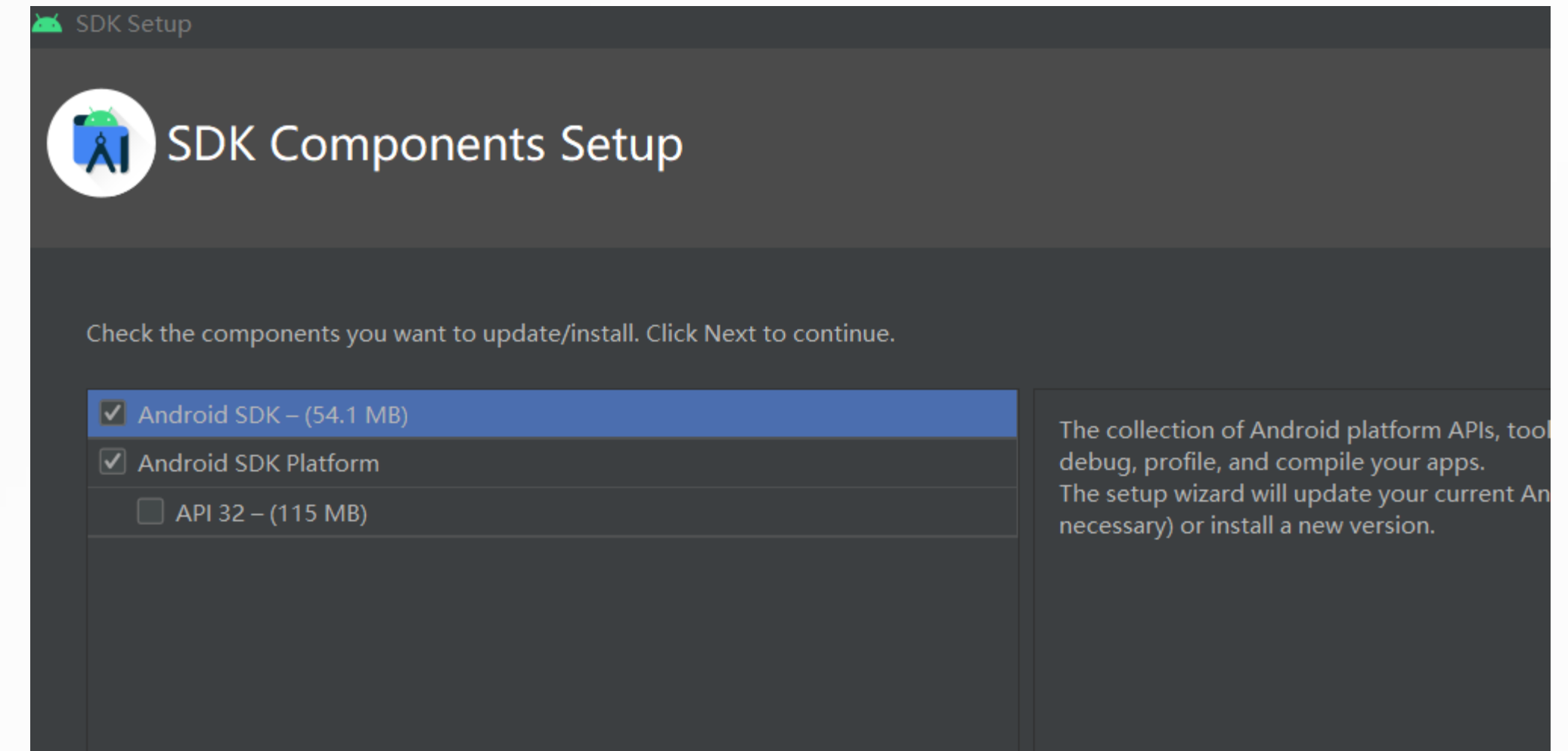
- Target triplets & search path
- LIBC API
- Crates

New prebuilt project repo, changes in 5+ architecture related gits, 20+ files

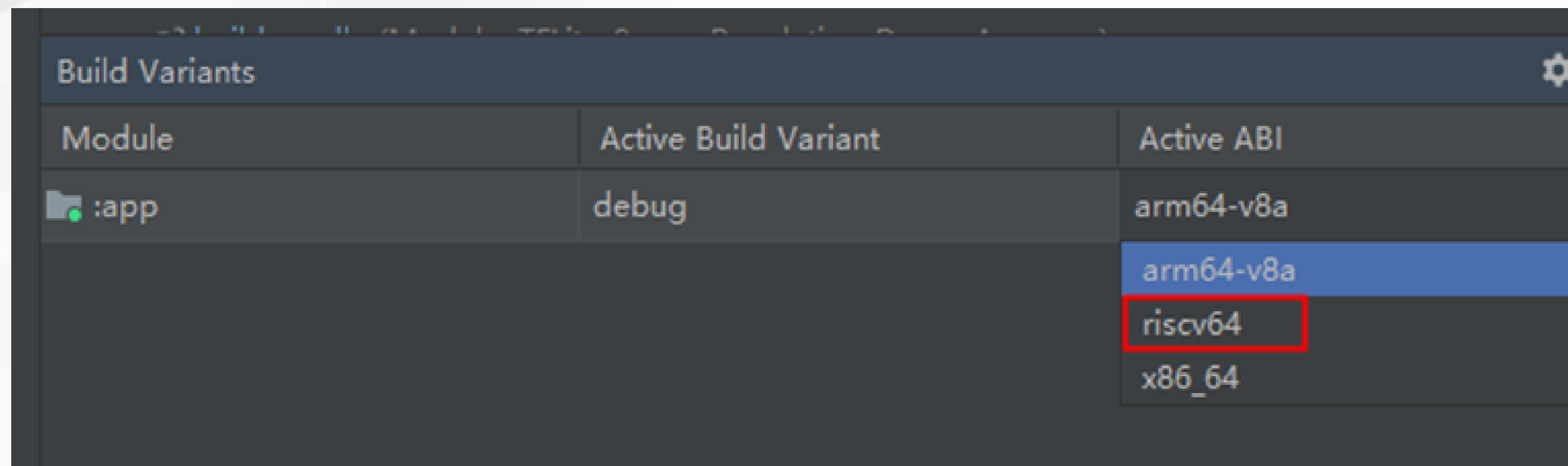
# Support Android Studio

```
18 ~/ndk/ndk-project ./ndk/checkbuild.py
./ndk/checkbuild.py base-toolchain binutils canary-readme changelog clang cpufeatures gte
ue ndk-build ndk-build-shortcut ndk-stack ndk-stack-shortcut ndk-which-sho
script-toolchain simpleperf source.properties sysroot system-stl toolchain v
/home/maoh/ndk/ndk-project/out/linux/android-ndk-r20/build/ndk-build -j192 V=1
lc' BIONIC_PATH=/home/maoh/ndk/ndk-project/bionic NDK_PROJECT_PATH=null AP
MK=/home/maoh/ndk/ndk-project/external/libcxx/Application.mk NDK_OUT=/home
libcxx/libs LIBCXX_FORCE_REBUILD=true
PLATFORM not set. Defaulting to minimum supported version android-16.
/home/maoh/ndk/ndk-project/out/libcxx/libs/armeabi-v7a/* /home/maoh/ndk/ndk-project/out
/home/maoh/ndk/ndk-project/out/libcxx/libs/x86_64/* /home/maoh/ndk/ndk-project/out/
/home/maoh/ndk/ndk-project/out/libcxx/libs/armeabi-v7a/gdbserver /home/maoh/ndk/ndk-proj
/home/maoh/ndk/ndk-project/out/libcxx/libs/x86_64/gdbserver
```

Generate NDK for Windows(MINGW)



Import SDK with RISC-V NDK support



Build APP for RISC-V

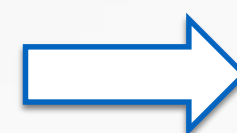
Must-have:

- Emulator in Window SDK
- Profiling tools
- Other missing dependent packages/libraries

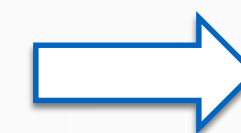


# Running TF lite on RISC-V

NN HAL  
support



Build TF lite  
demo APK



Install and  
execute

```

1  apply plugin: 'com.android.application'
2  apply plugin: 'de.undercouch.download'
3
4  android {
5      compileSdkVersion 31
6
7      defaultConfig {
8          applicationId "org.tensorflow.lite.examples.superresolution"
9          minSdkVersion 29
10         targetSdkVersion 31
11         versionCode 1
12         versionName "1.0"
13         externalNativeBuild {
14             cmake {
15                 arguments '-DANDROID_STL=c++_shared'
16             }
17         }
18         ndk {
19             abiFilters 'x86_64', 'arm64-v8a', 'riscv64'
20         }
21     }
22
23     buildTypes {
24         release {
25             minifyEnabled false
26             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
27         }
28     }
29 }

```

Build: Build Output × Build Analyzer ×

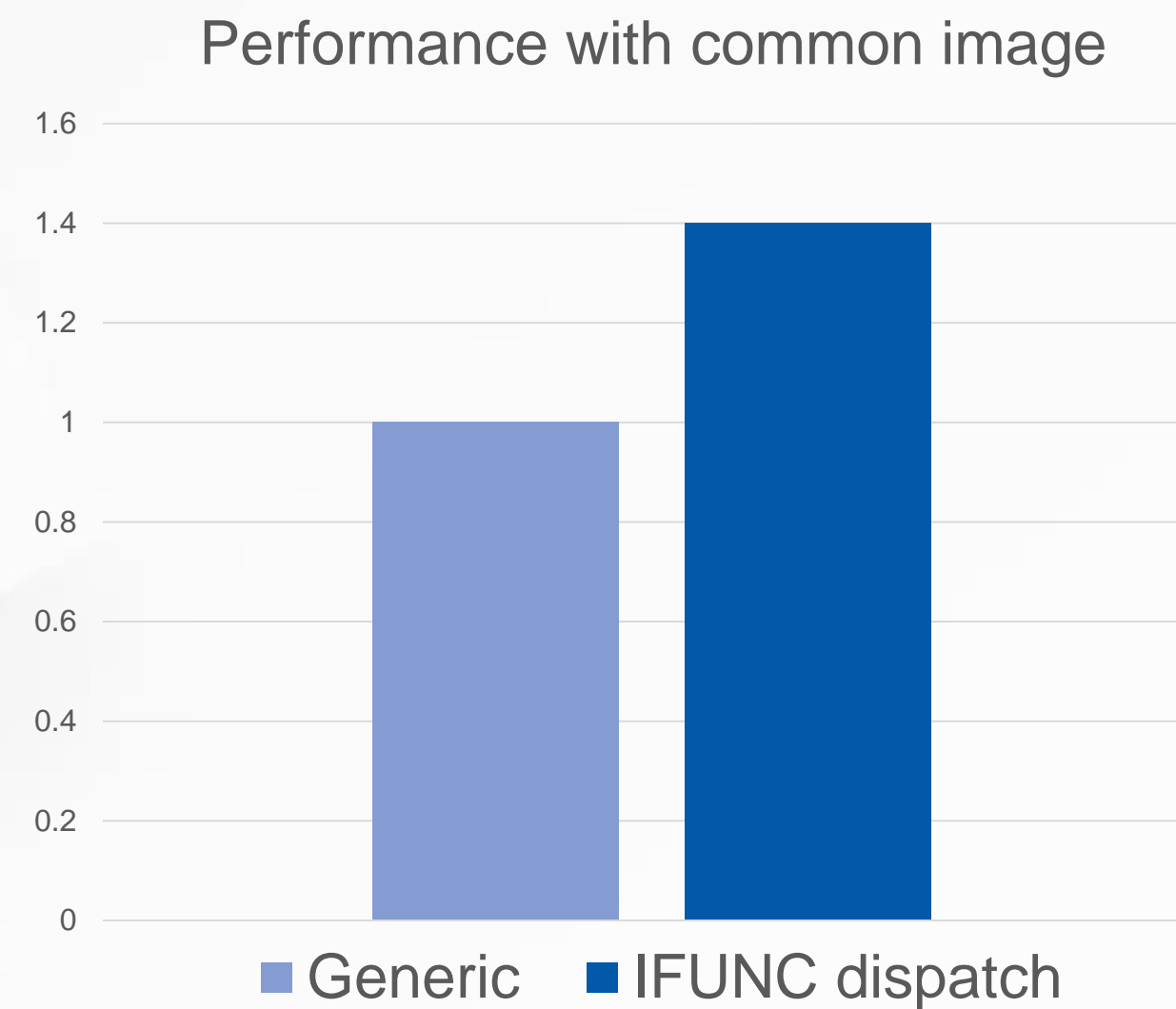
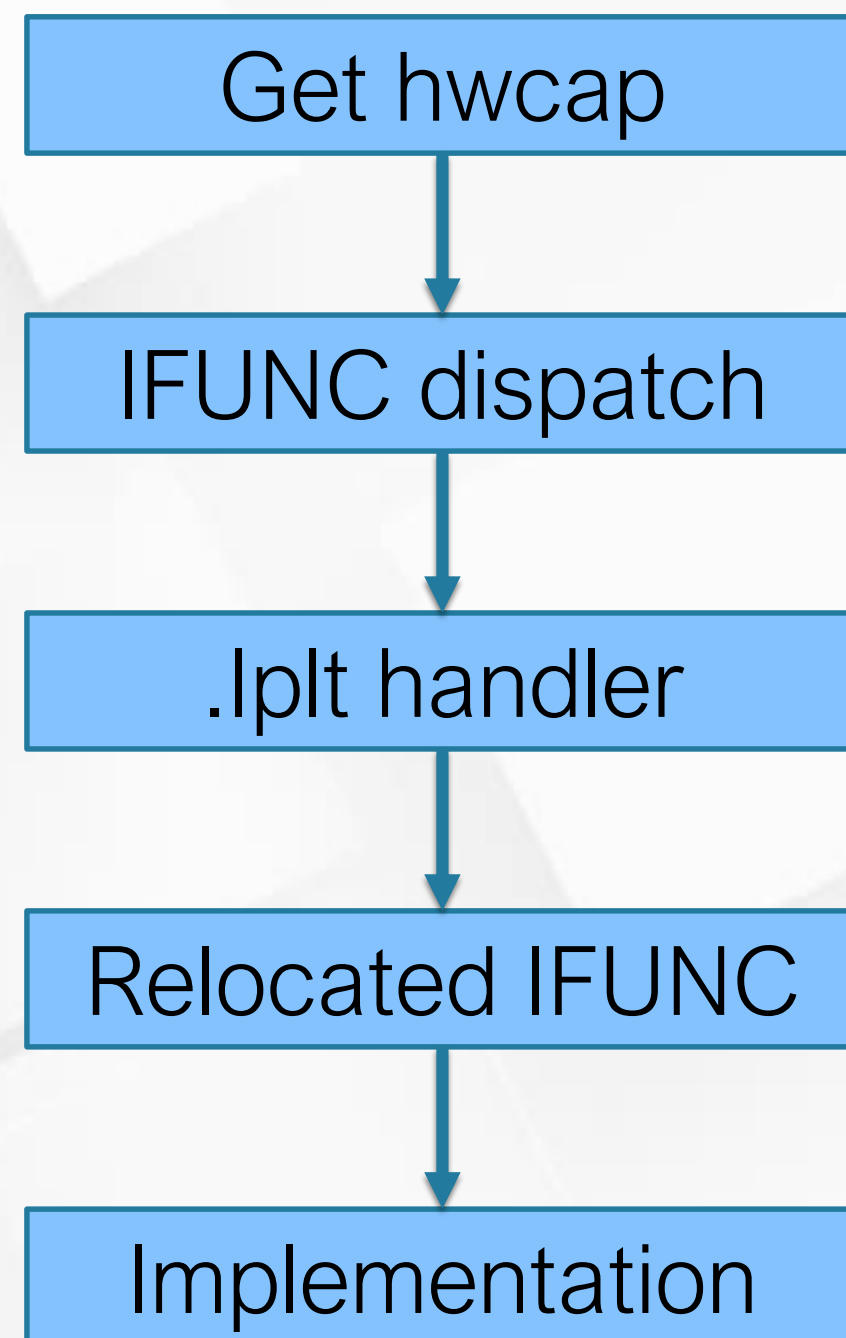
Build: finished At 2022/4/13 16:27 with 1 warning  
 [CXX5106] NDK was located by using ndk.dir property. This method is deprecated and will be removed in the future.

Clean SuperResolution arm64-v8a,lib\_tensorflowlite arm64-v8a  
 Clean SuperResolution x86\_64,lib\_tensorflowlite x86\_64



# Features Require Major Changes

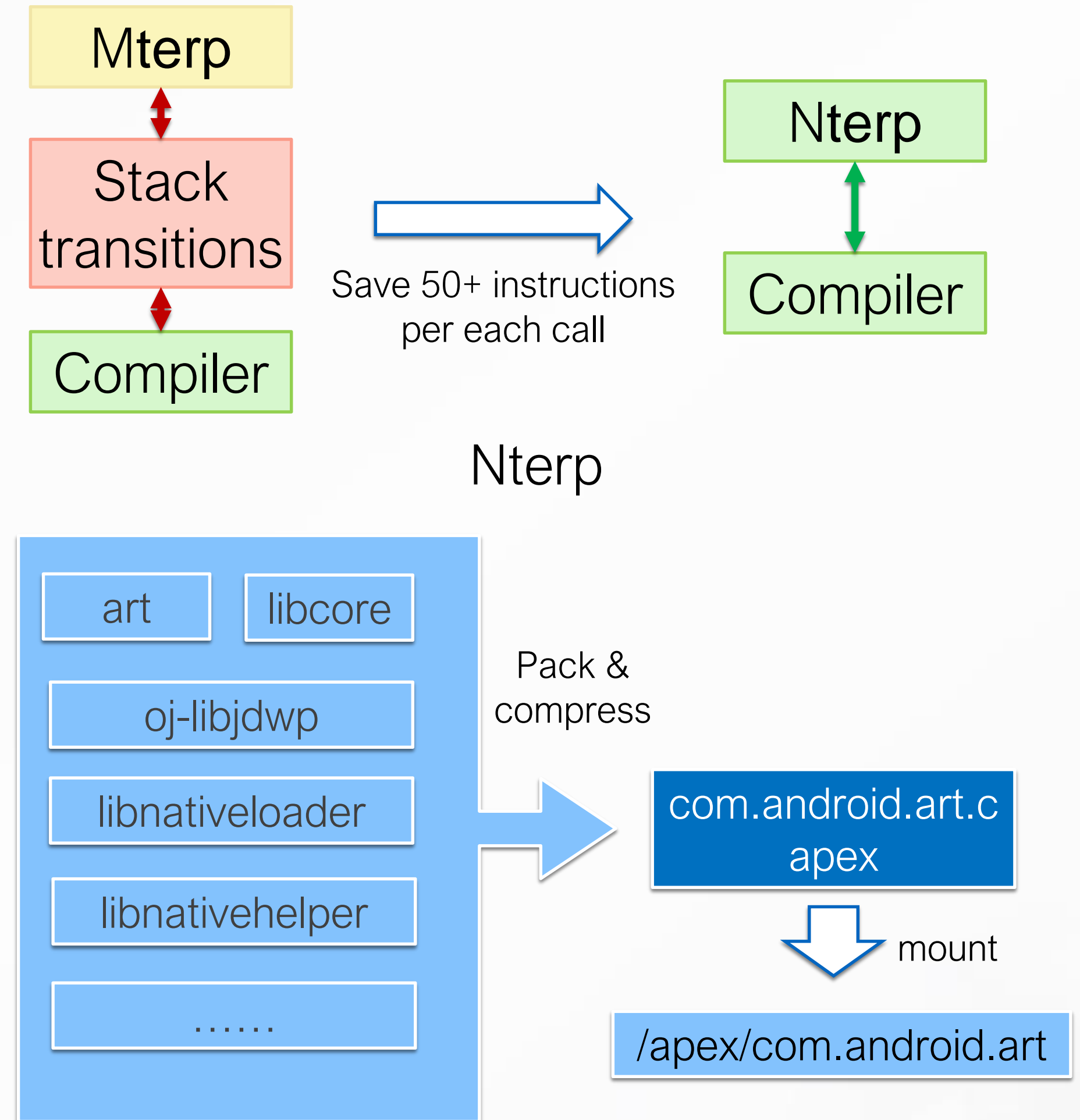
## Bionic



- Dynamic syscalls generation
- Kernel UAPI update
- New common relocation

~6k lines changes, 30+ commits

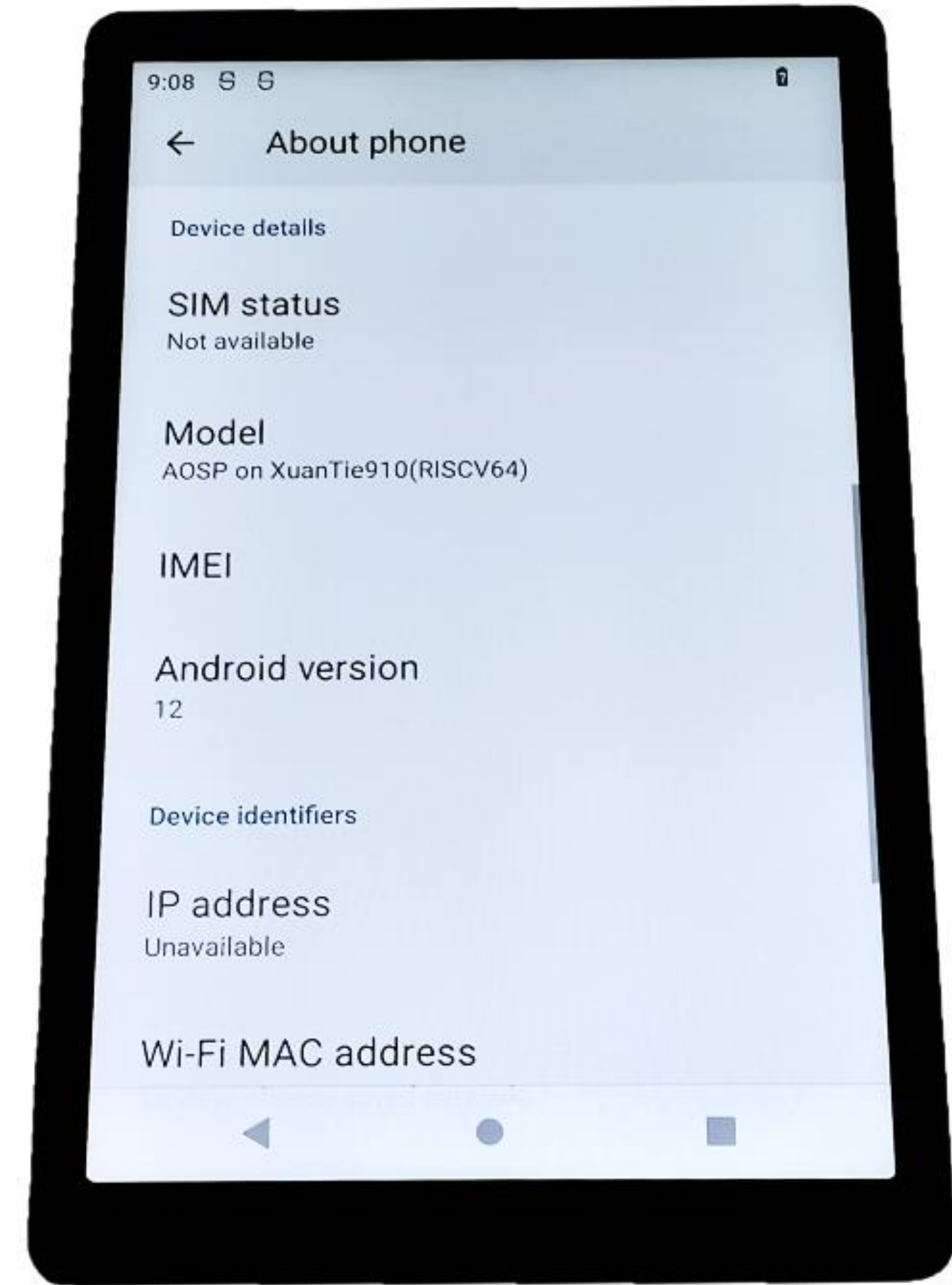
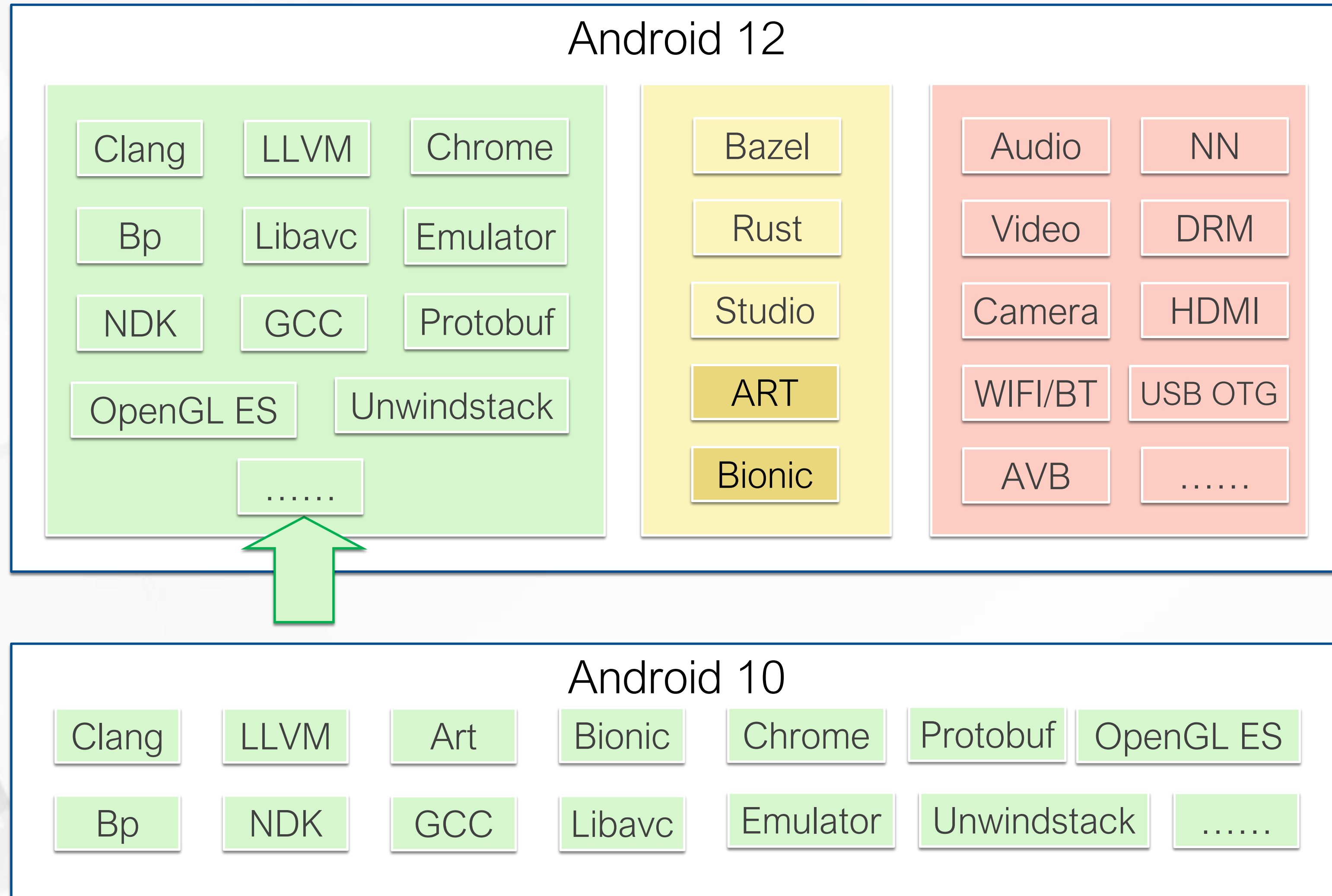
## ART



## Modular ART

~10K lines changes, 60+commits: JNI, New API, Functionality changes

# How to Support Android 12 on RISC-V



Android 12 on RISC-V



# Vendor Module Integration on RISC-V

- Audio playback
- DRM hwcomposer
- USB OTG
- WIFI/BT
- Video playback
- Camera

Poor RISC-V  
support

Incompatible

Challenges

Not yet  
verified on  
RISC-V

## Features - Video

Incompatible service bit width

Build configs not verified

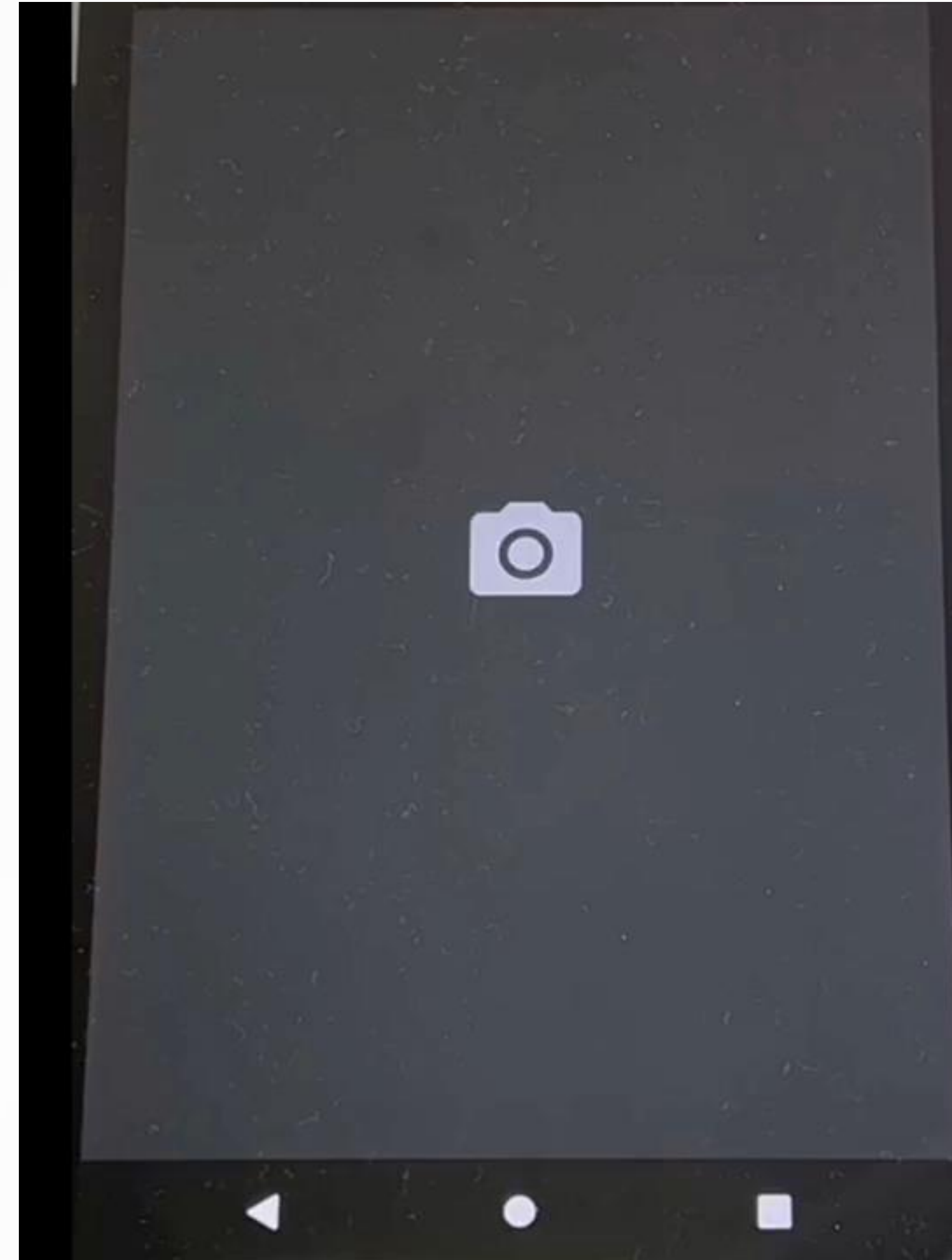
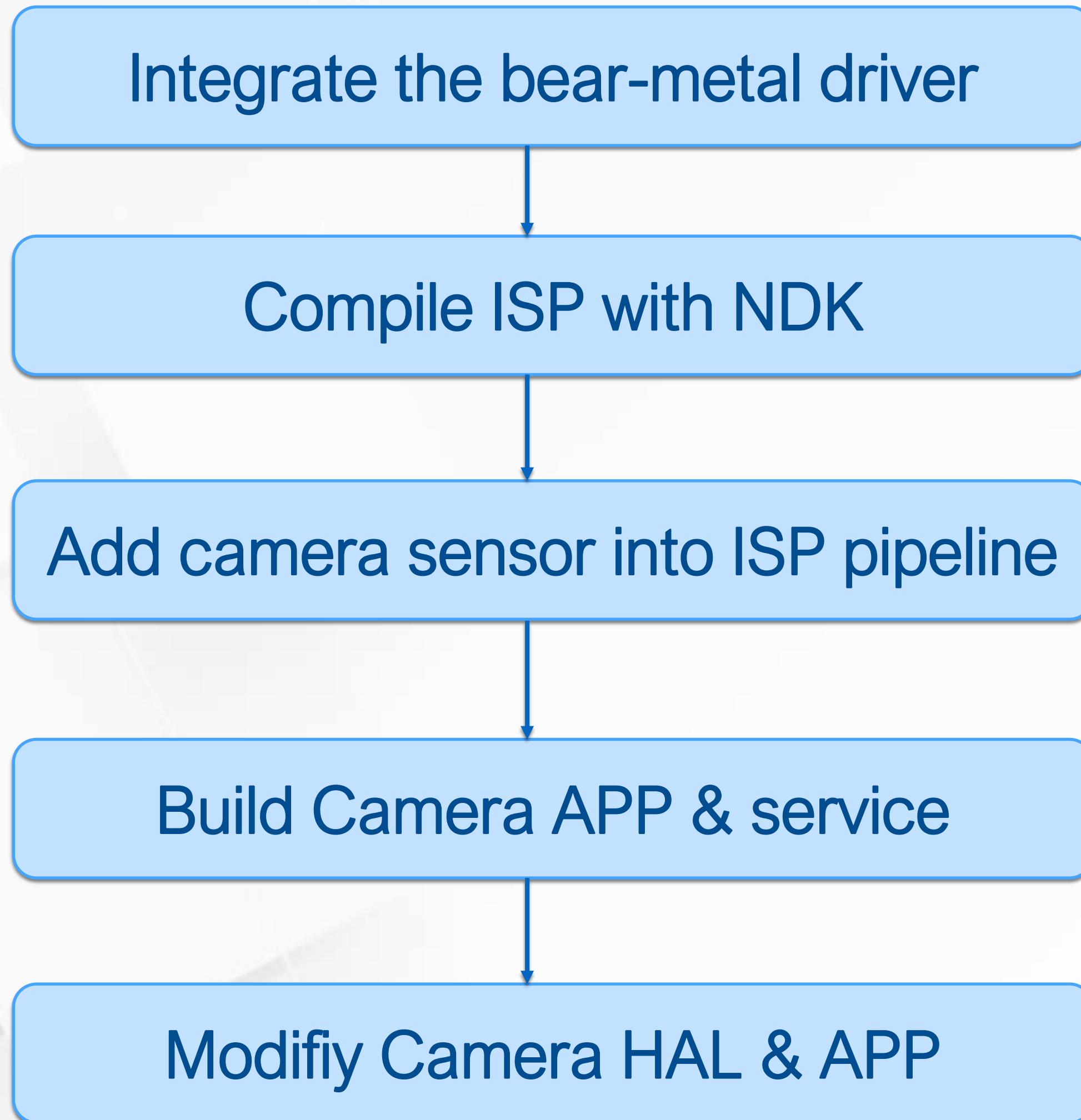
Conflict in OMX header

Missing NWB handler

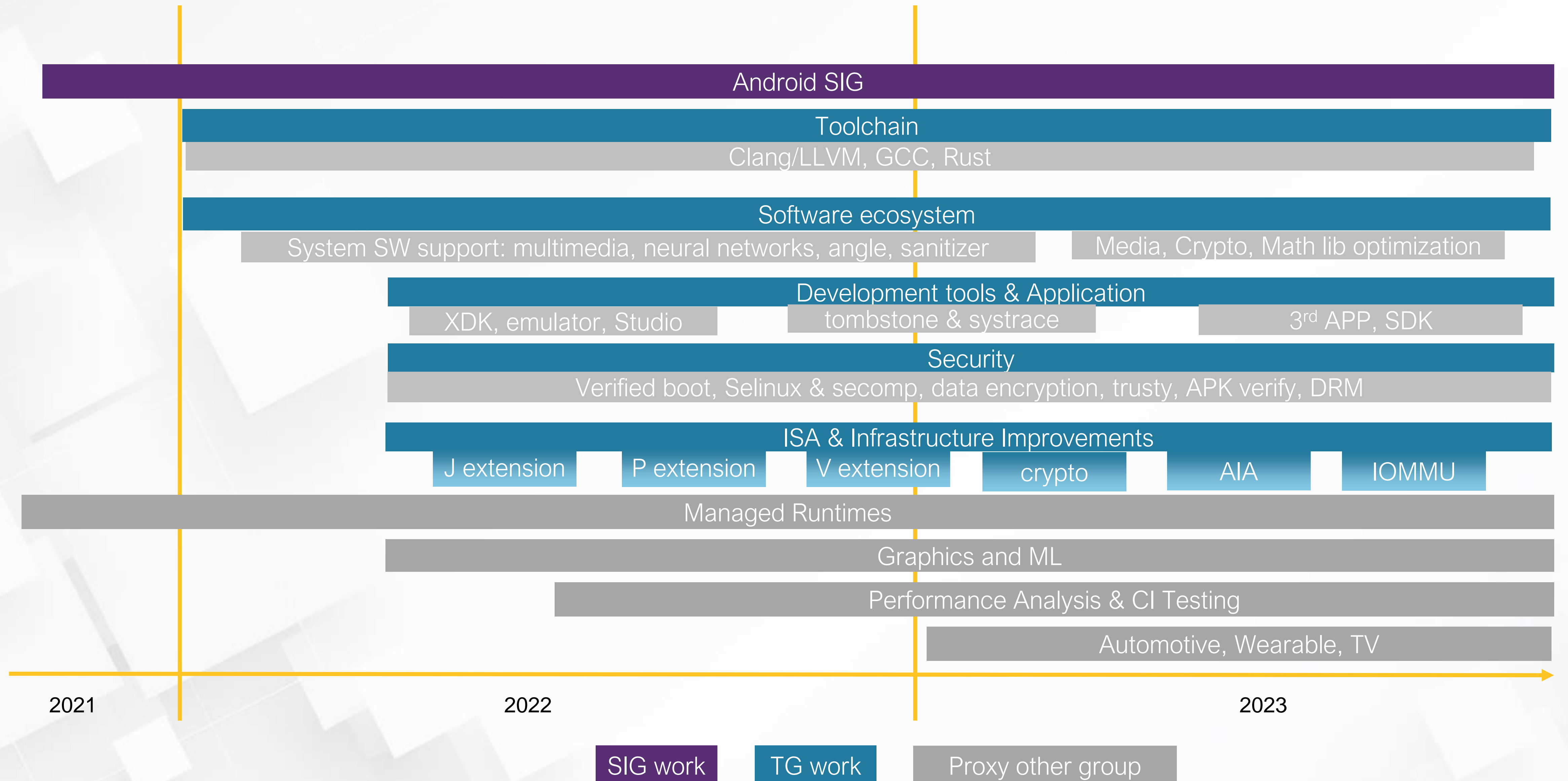




## Features - Camera

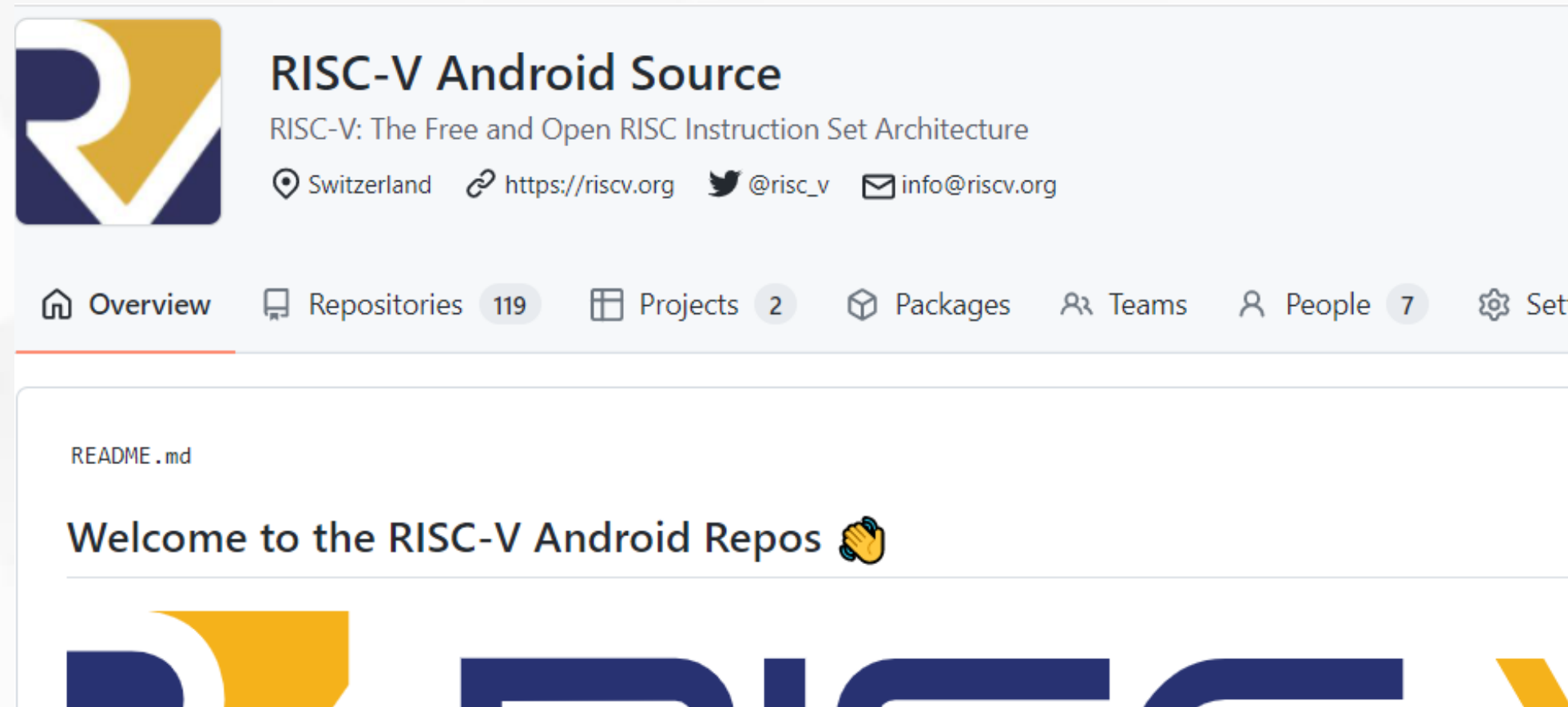


# RISC-V Android SIG





Look for the codes/binaries :  
<https://github.com/riscv-android-src>



Open chip community (English page):  
[occ.t-head.cn/community/risc\\_v\\_en](http://occ.t-head.cn/community/risc_v_en)



Build Android12 on RISC-V:  
<https://github.com/riscv-android-src/riscv-android/blob/main/doc/android12.md>

### Build Android 12 on RISC-V

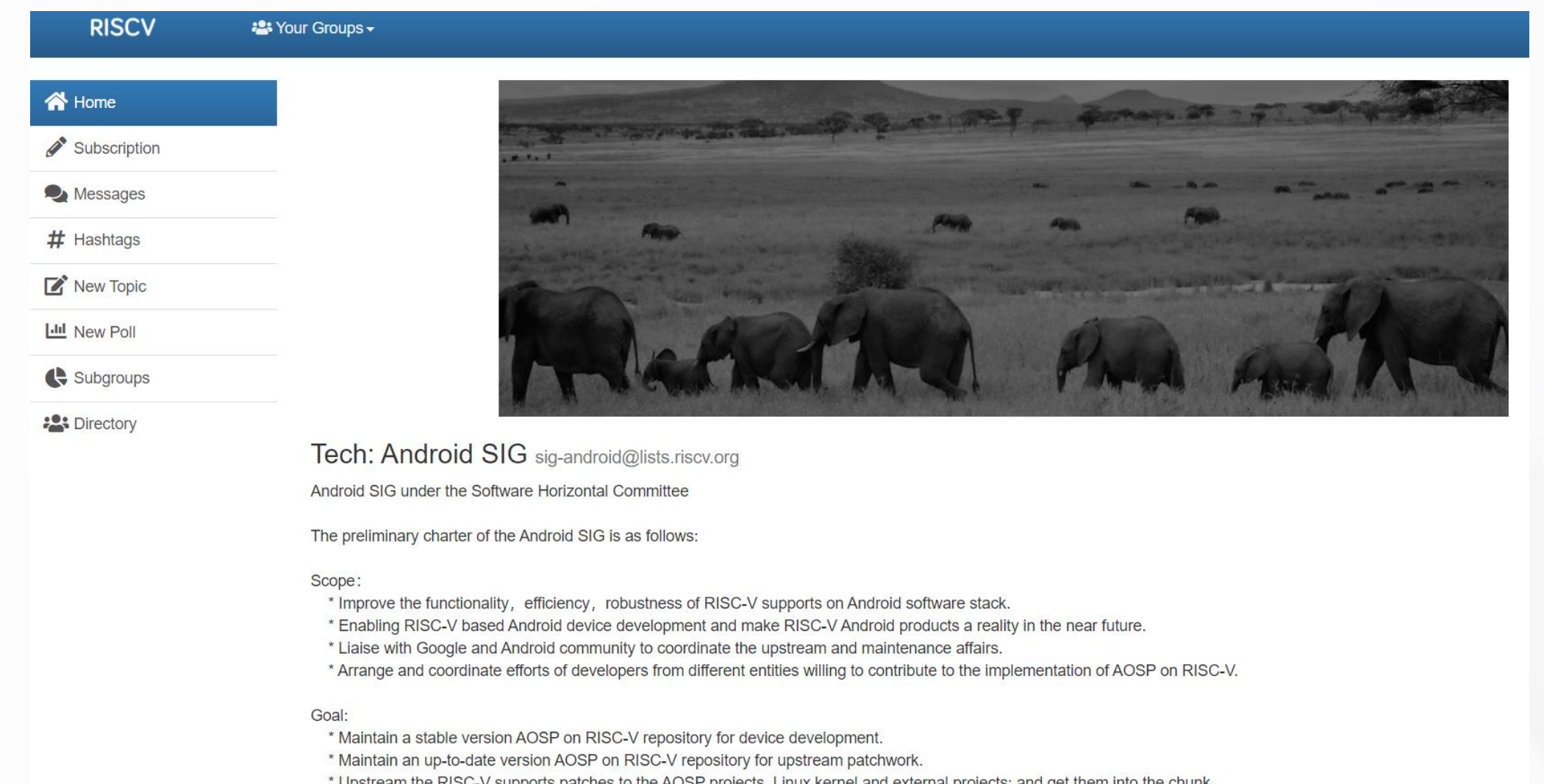
To download the RISC-V Android source tree to your working directory:

```
mkdir ~/riscv-android-src && cd mkdir ~/riscv-android-src
repo init -u git@github.com:riscv-android-src/manifest.git -b riscv64-android-12.0.0_dev
repo sync
cd prebuilts/rust/
git lfs pull
cd -
cd prebuilts/clang/host/linux-x86/
git lfs pull
mkdir prebuilts/runtime/mainline/runtime/sdk/android/riscv64/include/bionic/libc/kernel/uapi/asm-riscv64
rm external/angle/Android.bp
touch packages/modules/ArtPrebuilt/com.android.art-riscv64.apex
```

Build full emulator image with command:

```
source build/envsetup.sh
lunch sdk_phone64_riscv64
m -j
```

Join the discussion:  
<https://lists.riscv.org/g/sig-android>



# THANK YOU

