Bringing OpenHarmony to Phones

Lessons from the Oniro Porting Effort

Francesco Pham
Oniro PMC Lead - Huawei

What is Oniro?

Open-source OS platform managed by Eclipse Foundation

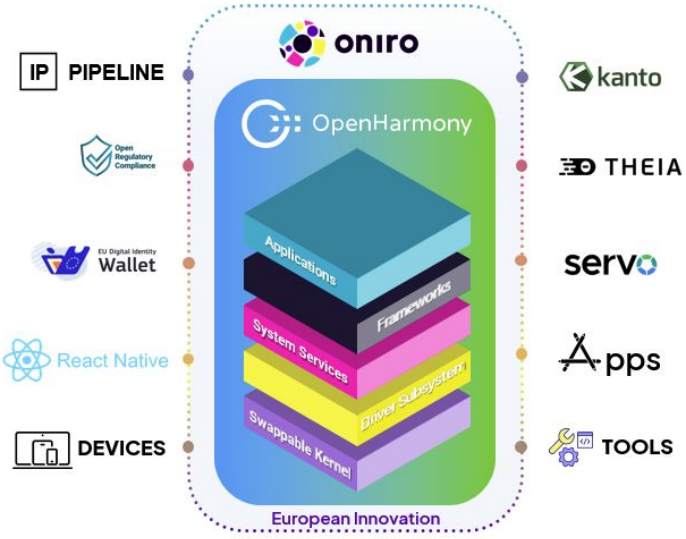
Built on **OpenHarmony** (a mature open-source project):

- Launched in 2020 via Huawei's code donation to the OpenAtom Foundation.
- Serves as the open-source upstream for HarmonyOS

Unique dual-foundation model:

- OpenAtom Foundation (China) + Eclipse Foundation (Europe)
- Shared technology, European governance
- Global scale with regional trust

Mission: Make cross-device platforms accessible for global adoption



From Foundations to Global Access

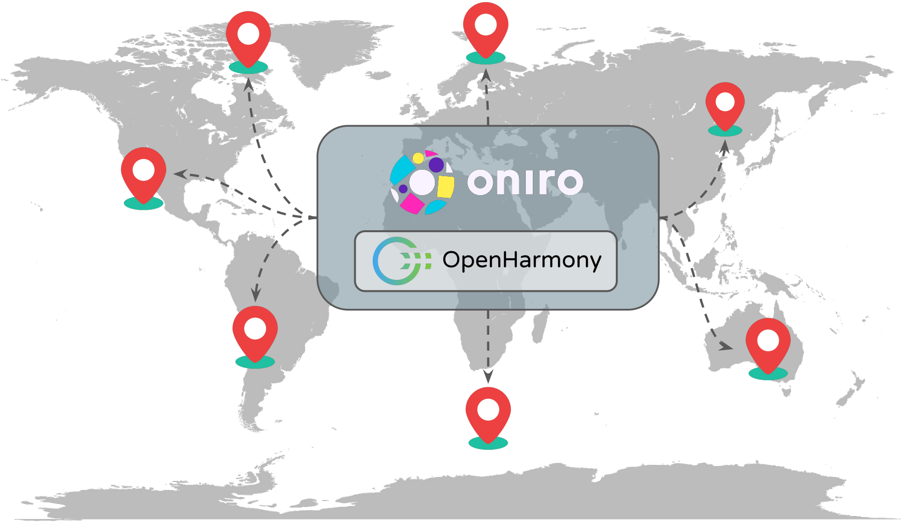
Why Oniro bridges OpenHarmony to the world:

Barriers outside China:

- Platform unfamiliarity (GitCode vs GitHub)
- Language barriers
- Limited access to familiar devices/tools

Oniro removes friction:

- GitHub mirrors with daily synchronization
- English-first documentation
- Globally available hardware support
- Developer-friendly tooling



Unified Cross-Device Architecture

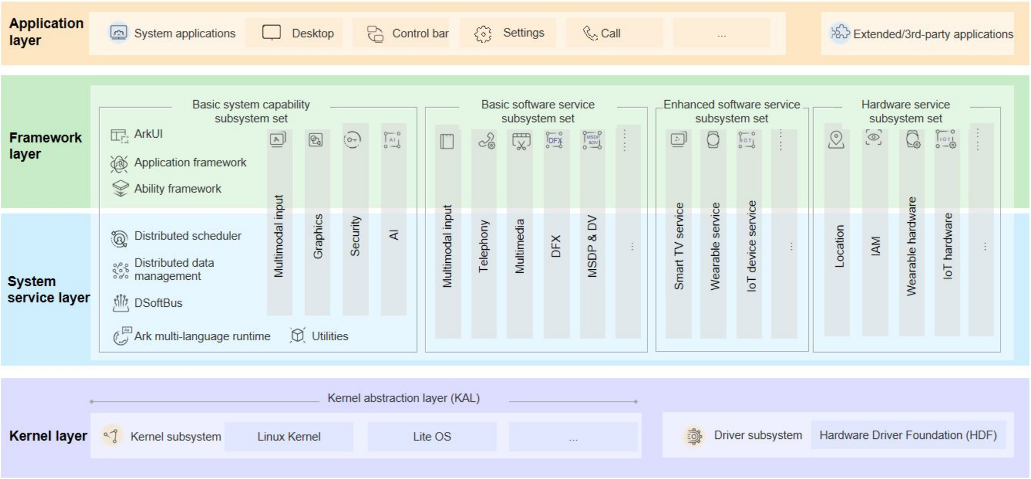
One platform, multiple device classes:

System	Devices	Capabilities
Mini	Sensors, wearables	MCU, lightweight
Small	Cameras, routers	Security, media
Standard	Phones, tablets	Full UI, 3D GPU

The Architecture

Technical Foundation:

- Multi-kernel support: **LiteOS** for mini system, **Linux** for standard system
- HDF: Unified driver abstraction for portability
- Graphics: Rosen (scene-graph compositor)
- Distributed capabilities: DSoftBus, Data sync, remote app invocation, device virtualization ("Super Device")
- App Framework: ArkUI Engine



Application layer

System applications

Desktop

Control bar

Settings

Call

...

Extended/3rd-party applications

Framework layer

System service layer

Basic system capability subsystem set

ArkUI

Application framework

Ability framework

Distributed scheduler

Distributed data management

DSoftBus

Ark multi-language runtime

Utilities

Multimodal input

Graphics

Security

AI

Basic software service subsystem set

Multimodal input

Telephony

Multimedia

DFX

MSDP & DV

...

Enhanced software service subsystem set

Smart TV service

Wearable service

IoT device service

...

Hardware service subsystem set

Location

IAM

Wearable hardware

IoT hardware

...

Kernel layer

Kernel abstraction layer (KAL)

Kernel subsystem

Linux Kernel

Lite OS

...

Driver subsystem

Hardware Driver Foundation (HDF)

Kernel Requirements

You cannot use vanilla mainline.
OpenHarmony requires Linux Kernel 5.10 / 6.6

```
# IPC Backbone
CONFIG_ANDROID_BINDER_IPC=y

# Memory Management
CONFIG_ASHMEM=y            # Unlike Android 12+
CONFIG_PURGEABLE_ASHMEM=y

# Hardware Driver Foundation
CONFIG_DRIVERS_HDF=y        # The magic sauce

# OpenHarmony-Specific Extensions
CONFIG_ACCESS_TOKENID=y     # Security token mapping
CONFIG_SCHED_RTG=y          # Related Thread Groups
```

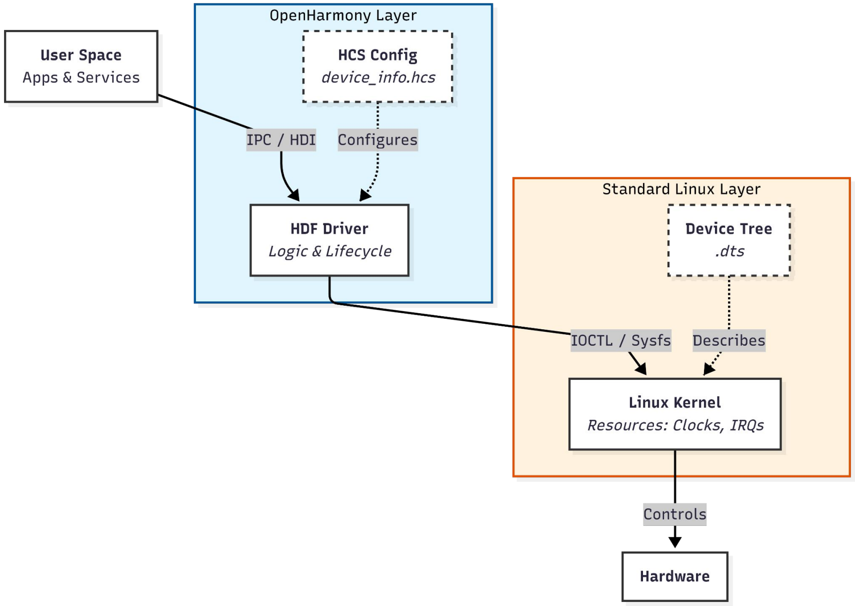
Why Ashmem still? OpenHarmony hasn't migrated to memfd_create yet.

HDF – Hardware Driver Foundation

Problem: How do you abstract drivers across LiteOS (RTOS) AND Linux?

Solution: HDF provides a unified driver framework.

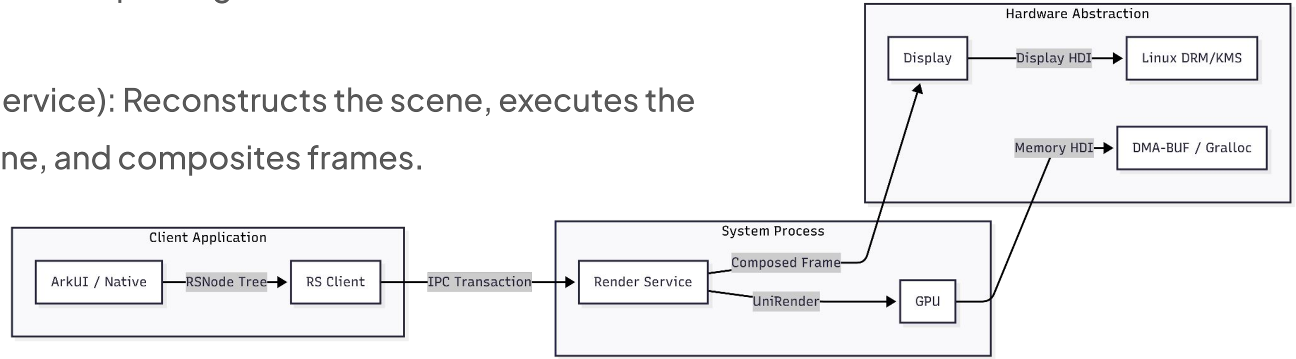
- HDF sits on top of the kernel, acting as a unified middleware.
- It does not replace the Linux Kernel or Device Tree;
- it abstracts the driver logic and service publishing



The Graphics Stack – Rosen Architecture

Client–Server Composition

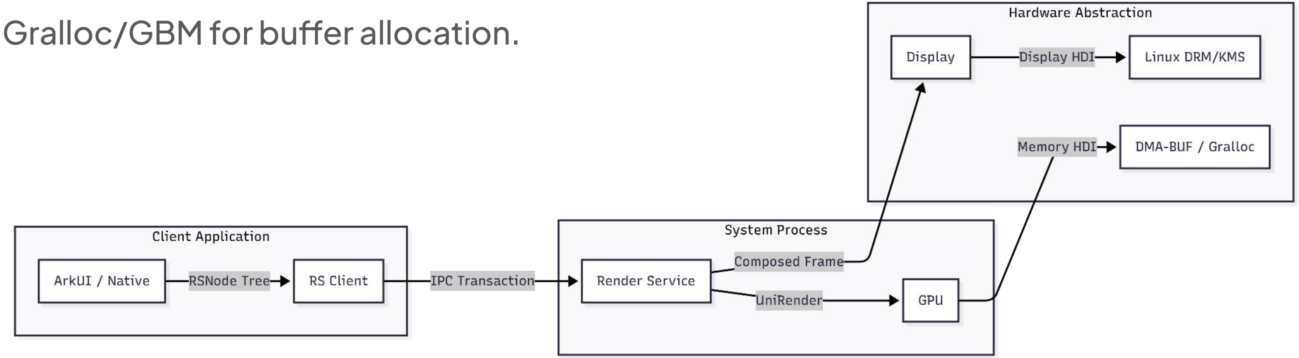
- **Rosen Subsystem:** The core rendering engine, similar to a "Retained Mode" compositor.
- **Split Architecture:**
  - Client (App): Builds a local Scene Graph (RSNode tree) and batches commands.
  - IPC Tunnel: Updates are packaged into Transactions and sent to the server.
  - Server (Render Service): Reconstructs the scene, executes the UniRender pipeline, and composites frames.



The Graphics Stack – Rosen Architecture

The Hardware Link

- **Surface/Memory**: Producer-Consumer model using shared memory and DMA-BUF.
- **HDI (Hardware Device Interface)**:
  - **Display**: Wraps standard Linux DRM/KMS for modesetting/vsync.
  - **Memory**: Wraps Gralloc/GBM for buffer allocation.



The GPU Driver Problem

For GPU driver integration, you need either:

- **Vendor blobs** (Musl-linked)
- **Open source drivers** (build Mesa for your chipset)

But here's the catch…

- Mobile SoC vendors provide drivers for Android (Bionic libc), not OpenHarmony (Musl libc).
- Mesa doesn't support all chipsets yet.

Binary incompatibility:

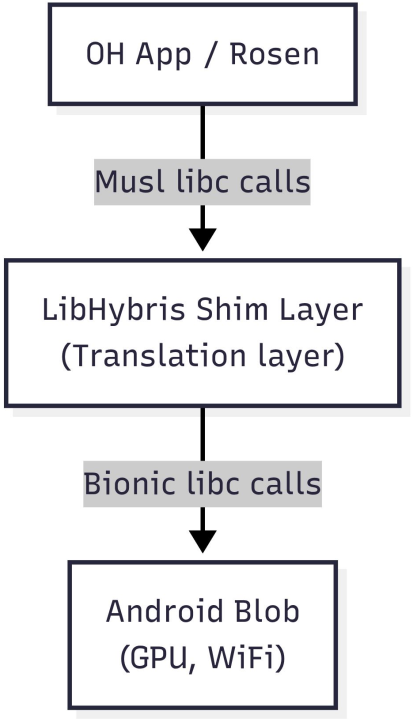
```
Android Driver  →  Calls __aeabi_memcpy (Bionic)
OpenHarmony   →  Has memcpy (Musl)
Result: Segfault 💥
```

LibHybris Solution

Oniro's pragmatic approach: LibHybris integration

- Adaptation layer translating Android calls to OpenHarmony's musl libc
- Enables GPU, Wi-Fi, peripheral support
- **Workaround, not ideal**: Binary blobs remain a black box
- First integration complete, more work ongoing

This could unlock thousands of existing Android devices for FOSS mobile OS use.



Porting to the OnePlus 6T & Volla X23

OnePlus 6T (SDM845)

- Well-supported by the community (postmarketOS/Mainline)
- **Graphics**: Freedreno (Mesa) - fully open
- **Audio**: ALSA userspace (alsa-lib) + HDF Audio HAL

Volla X23

- **Achieved**: Boot system with DRM display output and touch input
- **Challenges**: Less mainline support for Mediatek
- **Next steps**: Further Libhybris integration development

Porting to QEMU Emulator

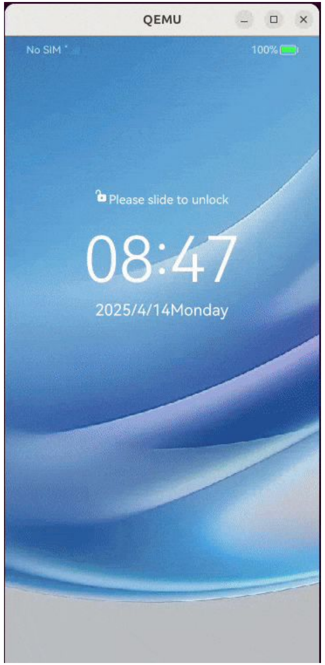
QEMU-based Emulator

- Started from Yocto container approach then moved to native board support
- Full graphical support (Mesa drivers)
- x86_64 architecture → hypervisor acceleration

Enables:

- CI/CD pipelines
- Rapid prototyping
- IDE integration for instant app testing

OpenHarmony accessible to everyone—no physical device needed



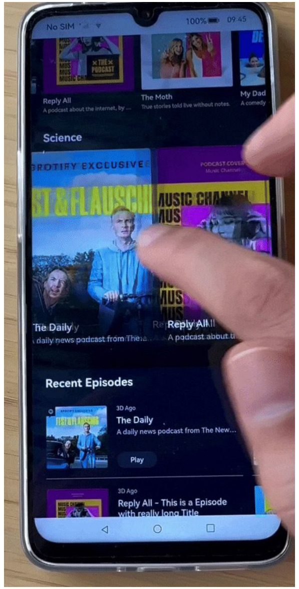
Growing the App Ecosystem

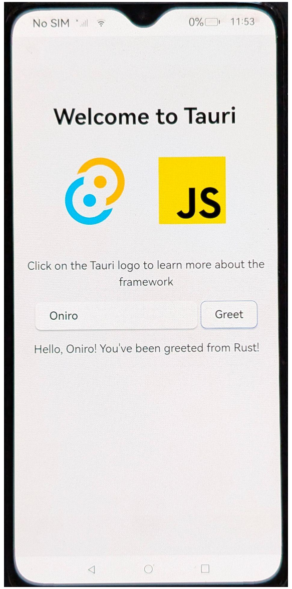
1. **Native ArkTS/ArkUI**

   - Declarative UI (like Flutter/SwiftUI/Jetpack Compose)
   - ArkTS: Extended TypeScript for performance

2. **Cross-platform frameworks**:

   - React Native (hundreds of apps already available)
   - Kotlin Multiplatform & Flutter (under research)
   - Rust frameworks (Tauri, Robius, Dioxus)


Science
SPOTIFY EXCLUSIVE
The Daily
Reply All
Recent Episodes
The Daily
A daily news podcast from The New...
Play
Reply All - This is a Episode with really long Title

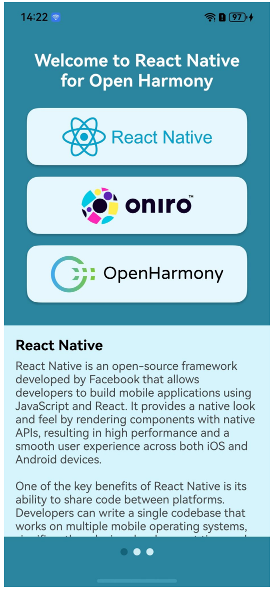


Welcome to Tauri

Click on the Tauri logo to learn more about the framework

Oniro

Hello, Oniro! You've been greeted from Rust!



Welcome to React Native for Open Harmony

React Native

React Native is an open-source framework developed by Facebook that allows developers to build mobile applications using JavaScript and React. It provides a native look and feel by rendering components with native APIs, resulting in high performance and a smooth user experience across both iOS and Android devices.

One of the key benefits of React Native is its ability to share code between platforms. Developers can write a single codebase that works on multiple mobile operating systems,

Developer Experience – Oniro IDE

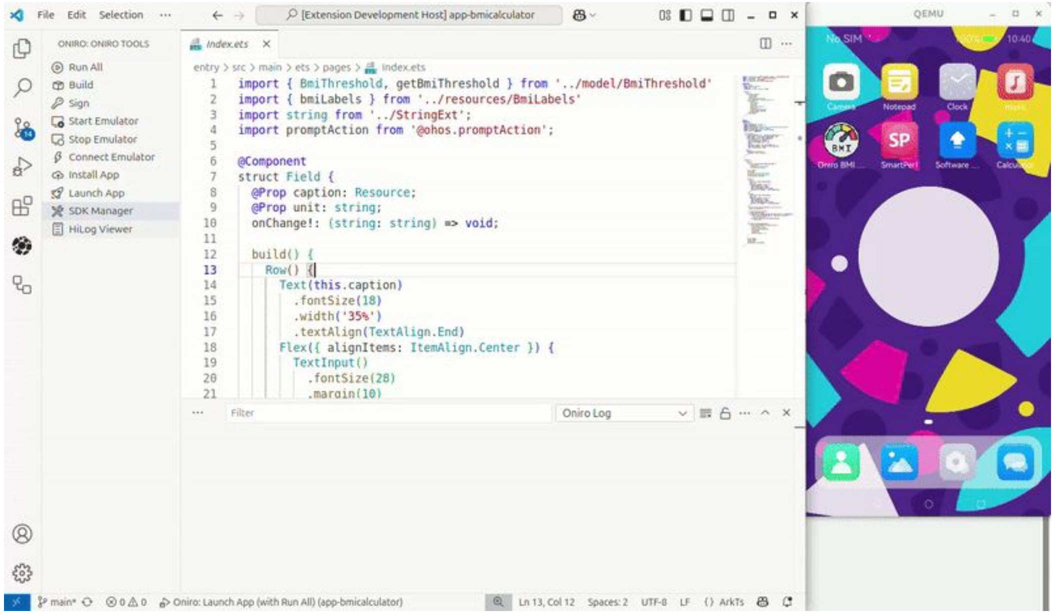
VS Code extension for Oniro/OpenHarmony development

Key features:

- SDK, cmd tools, emulator management
- Signature generation
- Build HAP packages
- Start apps, and filter logs (HiLog)

Impact:

- Fast development workflow in familiar editor
- Linux support (no Windows requirement)
- AI-assisted development ready

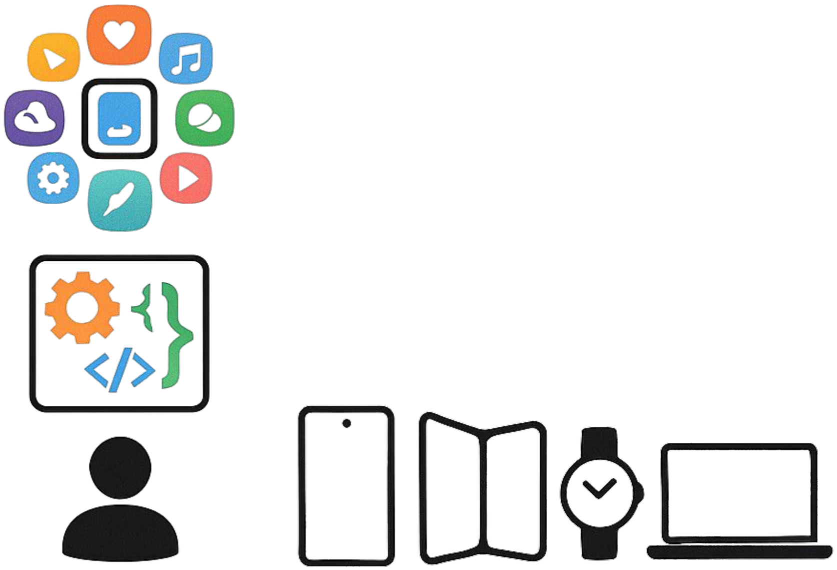


```
import { BmiThreshold, getBmiThreshold } from '../model/BmiThreshold'
import { bmiLabels } from '../resources/BmiLabels'
import string from '../StringExt';
import promptAction from '@ohos.promptAction';

@Component
struct Field {
  @Prop caption: Resource;
  @Prop unit: string;
  onChange!: (string: string) => void;

  build() {
    Row() {
      Text(this.caption)
        .fontSize(18)
        .width('35%')
        .textAlign(TextAlign.End)
      Flex({ alignItems: ItemAlign.Center }) {
        TextInput()
          .fontSize(28)
          .margin(10)
```

Roadmap

Future works:

- Richer app ecosystem through
- Wider third party frameworks integration
- Stronger global developer ecosystem
- More phones & form factors



Join Us in Shaping the Future

Contribute to Oniro development. Let’s create an open, secure mobile ecosystem together.

Think Global and Code Local

oniroproject.org

Find me on LinkedIn