OpenHarmony3.1 产品兼容性规范

OpenHarmony3.1 产品兼容性规范

文档版本 01

发布日期 2022-06-30

版权所有 (c) 2021-2022 OpenHarmony 兼容性工作组,保留一切权利。

版权许可

本产品兼容性规范文档的著作权由贡献本产品兼容性规范文档的 OpenHarmony 兼容性工作组拥有。本产品兼容性规范文档仅以只读形式向您提供,仅供您用于开发完全遵从 OpenHarmony 产品兼容性规范的产品的目的使用,您不得修改或创建本产品兼容性规范文档的衍生作品。您可以不经修改地原样分发本产品兼容性规范文档,但必须保留本产品兼容性规范文档中的所有声明。

商标声明

本产品兼容性规范文档不涵盖任何关于开放原子开源基金会和 OpenHarmony 项目的商标、名称或标志的许可。未经开放原子开源基金会书面事先明示许可,任何第三方不得以任何形式使用开放原子开源基金会和 OpenHarmony 项目的商标、名称或标志。

免责声明

OpenHarmony 项目会不定期对本产品兼容性规范文档的内容进行更新。

本产品兼容性规范文档在提供时不附带任何明示或默示的担保。在任何情况下,开放原子开源基金会或版权所有者不对任何人因使用本兼容性规范文档而引发的任何直接或间接损失承担责任,不论因何种原因导致或者基于何种法律理论,即使其曾被建议有此种损失的可能性。

目录

1 概述	
1.1 OpenHarmony 系统简介	1
1.2 OpenHarmony 兼容性目标	2
1.3 OpenHarmony 产品兼容性规范文档说明	3
2 介绍	
2 介绍	4
3 基础系统兼容性	6
3.1 轻量系统(mini system)兼容性	6
3.1.1 硬件	
3.1.1.1 CPU	
3.1.1.2 图形显示	6
3.1.1.3 内存和存储	6
3.1.1.4 通信	7
3.1.1.5 摄像头	7
3.1.1.6 音频	7
3.1.1.7 USB	7
3.1.1.8 加解密和安全隔离运行环境	7
3.1.2 软件	7
3.1.2.1 最小集 API 兼容性	7
3.1.2.2 运行时兼容性	14
3.1.2.3 应用兼容性	14
3.1.2.4 应用包格式	15
3.1.2.5 Ark UI 框架	
3.1.3 分布式	15
3.1.4 性能和功耗	15
3.1.5 安全	15
3.1.6 系统和软件升级	15
3.1.7 开发工具和开发选项	15
3.1.8 DFX (Design for X)	15
3.2 小型系统(small system)兼容性	16

3.2.1 硬件	16
3.2.1.1 CPU	16
3.2.1.2 图形显示	16
3.2.1.3 内存和存储	16
3.2.1.4 通信	16
3.2.1.5 摄像头	16
3.2.1.6 音频	16
3.2.1.7 USB	16
3.2.1.8 加解密和安全隔离运行环境	16
3.2.2 软件	17
3.2.2.1 最小集 API 兼容性	17
3.2.2.2 运行时兼容性	27
3.2.2.3 应用兼容性	28
3.2.2.4 应用包格式	28
3.2.2.5 Ark UI 框架	
3.2.3 分布式	28
3.2.4 性能和功耗	28
3.2.5 安全	28
3.2.6 系统和软件升级	28
3.2.7 开发工具和开发选项	28
3.2.8 DFX (Design for X)	28
3.3 标准系统(standard system)兼容性	29
3.3.1 硬件	29
3.3.1.1 CPU	29
3.3.1.2 图形显示	29
3.3.1.3 内存和存储	29
3.3.1.4 通信	29
3.3.1.5 摄像头	29
3.3.1.6 音频	29
3.3.1.7 USB	29
3.3.1.8 加解密和安全隔离运行环境	29
3.3.2 软件	30
3.3.2.1 最小集 API 兼容性	30
3.3.2.2 运行时兼容性	46
3.3.2.3 应用兼容性	46
3.3.2.4 应用包格式	46
3.3.2.5 JS UI 框架	47
3.3.3 分布式	47
3.3.4 性能和功耗	47
3.3.5 安全	47

3.3.6 系统和软件升级	47
3.3.7 开发工具和开发选项	47
3.3.8 DFX (Design for X)	47
4 可选系统能力兼容性	48
4.1 图形图像	49
4.2 电池管理服务	49
4.3 电源管理服务	49
4.4 USB 服务	49
4.5 AI 引擎	50
4.6 媒体	50
4.7 音频	51
4.8 相机	51
4.9 图像编解码	52
4.10 电路域电话短信服务	53
4.11 通话管理服务	53
4.12 输入法服务框架	53
4.13 定时&时间时区服务	
4.14 密钥和凭据	54
4.15 应用权限管理	54
4.16 应用完整性校验	55
4.17 多模输入服务	55
4.18 Ability assistant 工具	55
5 硬件兼容性	56
5.1 内存和存储	56
6 软件兼容性	<i>5</i> 7
6.1 API 兼容性	57
6.2 HDI 兼容性	
6.3 运行时兼容性	
6.4 应用兼容性	
6.5 应用包格式	59
6.6 Ark UI 框架	59
7 分布式兼容性	60
7.1 分布式硬件	60
7.2 分布式软总线	60
7.3 分布式数据管理	62
7.4 分布式任务调度	63

8	性能和功耗兼容性	63
9	系统和软件升级	65
10	> 开发工具和开发选项	66
1:	1 兼容性测试套件	67
12	2 修订记录	68

人 概述

- 1.1 OpenHarmony 系统简介
- 1.2 OpenHarmony 兼容性目标
- 1.3 OpenHarmony 产品兼容性规范文档说明

1.1 OpenHarmony 系统简介

OpenHarmony 是一款面向全场景、全连接、全智能时代的开源操作系统,采用部件化设计,支持在 128KiB 到 xGiB 内存资源的设备上运行,设备开发者可基于目标硬件能力自由选择系统部件进行集成。

为保证在不同硬件上易集成,OpenHarmony 定义了三种基础系统类型,设备开发者通过选择基础系统类型完成必选部件集配置后,便可实现其最小系统的开发。三种基础系统类型的参考定义:

• 轻量系统 (mini system)

面向 MCU 类处理器的设备,硬件资源极其有限,支持的设备最小内存为 128KiB,可以提供多种轻量级网络协议,轻量级的图形框架,以及丰富的 IOT 总线读写部件等。可支撑的产品如智能家居领域的连接类模组、传感器设备、穿戴类设备等。

小型系统 (small system)

面向具备 MMU 的应用处理器的设备,支持的设备最小内存为 4MiB,可以提供更高的安全能力、标准的图形框架、视频编解码的多媒体能力。可支撑的产品如智能家居领域的 IP Camera、电子猫眼、路由器以及行车记录仪等。

● 标准系统(standard system)

面向应用处理器的设备,支持的设备最小内存为 128MiB,可以提供增强的交互能力、GPU 以及硬件合成能力、更多控件以及动效更丰富的图形能力、完整的应用框架。可支撑的产品如带屏 IOT 设备、轻智能手机等。

如上几种基础系统中所支持的最小内存,字节的次方单位采用 IEC 60027-2 标准中定义的二进制前缀。

OpenHarmony 也提供了一系列可选的系统部件,方便设备开发者按需配置,以支撑其特色功能的扩展或定制开发。

1.2 OpenHarmony 兼容性目标

鉴于 OpenHarmony 组件可以按需拼装,即允许不同设备提供不同的系统能力,并且由于 OpenHarmony 源代码开源,可能导致同一个功能的源码经过不同设备开发修改后在软硬件规格方面也会有差异。但每个用户和开发者都需要一个共同的生态系统,即在此生态系统中的设备是兼容可互通的,以及设备上可运行应用的用户体验是一致的。因此,需要为设备开发定义统一的兼容性设计要求,以保证所有基于 OpenHarmony 开发的设备都能够兼容这个共同的生态系统,并且可以通过明确的设备认证流程来加入这个生态。

从用户和开发者视角的 OpenHarmony 兼容性目标如下:

- 用户视角
 - a. 单设备功能可用:
 - i. 单设备功能/特性具有延续性;
 - ii. 合法渠道安装的应用功能可用。
 - b. 设备组合功能可用:
 - i. 基于使用场景,可以通过组合相关设备完成场景功能;
 - ii. 基于使用场景,有统一的场景体验,不依赖具体厂家硬件差异;
 - iii. 基于使用场景,以及当前设备组合状态,符合预期的一致业务体验;
 - iv. 设备加入/退出已有场景组合时,业务体验可以预期;
 - v. 设备可以在不同使用场景下,参与设备组合并完成功能。
- 开发者视角
 - a. 应用开发者: 应用可根据设备兼容性要求正确分发, 跨设备/跨版本兼容运行, 给用户提供一致的业务体验。
 - b. 设备开发者:
 - i. 用于指导集成 OpenHarmony 系统的统一的软硬件兼容性技术规范;
 - ii. 明确的设备认证/发布要求与指导;
 - iii. 统一的基础协议兼容性要求,以支撑实现设备间互通。

OpenHarmony 通过如下三个部分达成兼容性目标:

- OpenHarmony 开源项目的源代码。开发者可以免费获得并移植到设备。
- OpenHarmony 产品兼容性规范文档 (PCS), 定义设备兼容性的标准。开发者必须 遵循此文档定义的规则进行设备开发。
- OpenHarmony兼容性测试套件(XTS),提供验证设备兼容性的执行机制。开发者可以使用此套件进行兼容性验证和评估。

1.3 OpenHarmony 产品兼容性规范文档说明

OpenHarmony 产品兼容性规范文档(Product Compatibility Specifications,以下简称产品兼容性规范文档)定义了与 OpenHarmony 系统兼容的产品必须满足的要求,满足本产品兼容性规范文档定义要求的产品,才可能被认定与 OpenHarmony 系统兼容。

本产品兼容性规范文档从 API 兼容性、分布式接口兼容性方面对生态设备进行定义,以促进 OpenHarmony 应用在所有 OpenHarmony 生态设备上兼容性运行和分布式互通。本产品兼容性规范文档还从软硬件基础规格等方面对生态设备进行定义,以保证用户在不同 OpenHarmony 生态设备上相同使用场景的体验一致性。

需要说明的是,本产品兼容性规范文档的作用是整理已经明确的兼容性要求,并非综合全面的设备兼容性开发指导,会根据业务需求持续更新。OpenHarmony 开源项目包含首选的 OpenHarmony 开源实现,设备开发时建议优先选择该首选的 OpenHarmony 开源实现,如果修改该首选的开源实现,请详细阅读本产品兼容性规范要求的兼容性规则并确保遵循其规则。

2 介绍

用词约定

本文中出现的"必须(MUST)"、"必需(REQUIRED)"、"会(SHALL)"、"不会(SHALL NOT)"、"应(SHOULD)"、"不应(SHOULD NOT)"、"建议(RECOMMENDED)"、"强烈建议(STRONGLY RECOMMENDED)"、"可以(MAY)"、"可选(OPTIONAL)"和"禁止(MUST NOT)"用词遵循 RFC2119 中定义的 IETF 标准。

文档结构

第3 节基础系统兼容性定义特定系统类型的兼容性规范,第4 节可选系统能力兼容性定义可选系统能力的兼容性规范,第5节 硬件兼容性及之后定义普遍适用于所有 OpenHarmony 系统类型的兼容性规范。

针对规范所需"基础系统 ID"定义如下:

- M: 适用于轻量系统 (mini system)。
- S: 适用于小型系统(small system)。
- STD: 适用于标准系统(standard system)。
- ALL: 适用于所有系统。

针对"条件必选"兼容性规范定义如下:

• C: Conditional, 有条件必选规范, 设备选择集成某个系统能力时需要遵循。

针对"通用"兼容性规范定义如下:

• G: General, 通用规范, 所有系统可引用。

针对规范定义所需的"TYPE"定义如下:

- HARDWARE: 适用于硬件兼容性的规范。
- SOFTWARE: 适用于软件兼容性的规范。
- UPDATE: 适用于设备与 APP 升级兼容性的规范。
- DISTRIBUTED: 适用于分布式兼容性的规范。

- PERFORMANCE: 适用于性能兼容性的规范。
- POWER: 适用于功耗兼容性的规范。
- SECURITY: 适用于安全兼容性的规范。
- TOOLS: 关于开发工具与选项兼容性的规范。
- CERTIFICATION: 关于兼容性测试的规范。
- DFX: Design for X,关于可靠性、可测试性、可服务性、可供应性、可伸缩性、 能效与环境设计等方面的兼容性规范。

强制规范定义

对每个"必须 (MUST)"规范定义的格式:

- 1. 针对基础系统定义的 ID 格式为【基础系统 ID-TYPE-XXXX】(例如【M-HARDWARE-0100】)。
- 2. 针对条件必选定义的 ID 格式为【C-基础系统 ID[| 基础系统 ID...]-TYPE-XXXX】 (例如【C-M|S-HARDWARE-0100】)。
- 3. 通用规范定义的 ID 格式为【G-TYPE-XXXX】(例如【G-HARDWARE-0100】)。

强烈建议规范定义

对每个"强烈建议(STRONGLY RECOMMENDED)"规范定义的格式:

- 1. 针对基础系统定义的 ID 格式为【基础系统 ID-TYPE-SR-XXXX】(例如【M-HARDWARE-SR-0100】)。
- 2. 针对条件必选定义的 ID 格式为【C-基础系统 ID[| 基础系统 ID...]-TYPE-SR-XXXX】(例如【C-M|S-HARDWARE-SR-0100】)。
- 3. 通用规范定义的 ID 格式为【G-TYPE-SR-XXXX】(例如针对硬件兼容性的强烈建议为【G-HARDWARE-SR-0100】)。

3

基础系统兼容性

采用 OpenHarmony 系统开发的设备,必须选择一种基础系统类型进行必需系统部件的集成,以支持其最小系统的实现。

选择某一种基础系统类型开发的设备必须遵循本章定义的对应基础系统兼容性规范。

- 3.1 轻量系统 (mini system) 兼容性
- 3.2 小型系统(small system)兼容性
- 3.3 标准系统(standard system)兼容性

3.1 轻量系统(mini system)兼容性

3.1.1 硬件

3.1.1.1 CPU

【M-HARDWARE-SR-0100】主 CPU 的 DMIPS 大于 100 MIPS。

3.1.1.2 图形显示

如果提供图形显示能力,必须遵循 4.1 图形图像定义的硬件兼容性规范。

3.1.1.3 内存和存储

必须遵循 5.1 内存和存储中定义的硬件兼容性规范。

最小内存和存储

为了定义可以运行符合 OpenHarmony 兼容性规范的最小系统,以及完成设备的基本业务功能:

【M-HARDWARE-SR-0200】设备整体内存空间≥128KiB。

【M-HARDWARE-SR-0201】设备整体可读写非易失存储空间≥2MiB。

【M-HARDWARE-SR-0202】用户可读写非易失存储空间≥128KiB。

3.1.1.4 通信

必须遵循 7.2 分布式软总线中定义的硬件兼容性规范。

3.1.1.5 摄像头

如果设备支持摄像头,必须遵循 4.8 相机定义的硬件兼容性规范。

3.1.1.6 音频

如果设备支持音频输入或输出,必须遵循 4.7 音频定义的硬件兼容性规范。

3.1.1.7 USB

如果设备支持 USB, 必须遵循 4.4 USB 服务定义的硬件兼容性规范。

3.1.1.8 加解密和安全隔离运行环境

如果设备支持硬件加解密模块,必须遵循 4.14 密钥和凭据定义的硬件加密兼容性规范。如果设备支持 TEE,必须遵循 4.14 密钥和凭据定义的硬件 TEE 兼容性规范。

3.1.2 软件

3.1.2.1 最小系统部件集和 API 兼容性

必须遵循 6.1 API 兼容性定义的兼容性规范。

【M-SOFTWARE-0100】必须满足 Mini 系统最小部件集定义和部件对应的 API 兼容性 要求。要求如下:

表 3-1 轻量最小系统部件集和 API 兼容性列表

子系统	必选部 件	源码目录	仓	功能	需要兼容的 API
内核	liteos_m	kernel/lit eos_m	kernel_l iteos_m	LiteOS-M 内 核,支持 riscv32、 ARM 和 C- SKY 架构。	liteos_m 内核提供的 CMSIS 或 POSIX 标准 API
驱动	device_dr iver_fram ework	drivers/fr amework drivers/ad apter	drivers_ framew ork drivers_ adapter	HDF (Hardware Driver Foundation) 驱动框架。	

子系	必选部	源码目录	仓	功能	 需要兼容的 API
统	件	<i>""</i>	ן	73110	帝女兼谷的人で
语言 编译 器运 行时	musl 或 三方运 行时库	third_part y/musl 或 其他自定 义的三方 目录	third_p arty_m usl 或 三方仓	c 运行时 库。	系统基础库 libc 提供的 API
DFX	hilog_lite	base/hivi ewdfx/hil og_lite	hiviewd fx_hilo g_lite	轻量系统和 小型系统的 流水日志功 能。	如下头文件中定义的 API: base/hiviewdfx/hilog_lite/int erfaces/native/kits/hilog_lite/ log.h base/hiviewdfx/hilog_lite/int erfaces/native/kits/hilog_lite/ hiview_log.h
	hievent_l ite	base/hivi ewdfx/hie vent_lite	hiviewd fx_hiev ent_lite	在轻量和小,是一个,是一个,是一个,是一个,是一个,是一个,是一个,是一个,是一个,是一个	
	hiview_li te	base/hivi ewdfx/hi view_lite	hiviewd fx_hivi ew_lite	轻量 DFX 初始化,控制轻量 DFX 部件的按需启动。	
启动恢复	syspara_l ite	base/start up/syspar a_lite	startup_ syspara _lite	轻量和小型 系统的系统 属性设置、 读取。	如下头文件中定义的 API: base/startup/syspara_lite/inte rfaces/kits/parameter.h
	bootstrap _lite	base/start up/bootst rap_lite	startup_ bootstra p_lite	轻量和小型 系统启动引 导。	
分布 式软 总线	dsoftbus	foundatio n/commu nication/d softbus	commu nication _dsoftb us	近场设备间 分布式通信 能力管理, 提供不区分 链路和传输 发现和传输 能力。	
系统 服务	samgr_lit	foundatio n/distribu	distribu tedsche	轻量和小型 系统上提供	/

子系 统	必选部件	源码目录	仓	功能	需要兼容的 API
管理	е	tedschedu le/samgr_ lite	dule_sa mgr_lit e	系统服务注 册和发现能 力。	
安全	device_a uth	base/secu rity/devic eauth	security _device auth	设备间可信 关系的建 立、维护、 使用、撤销 等全生命 期的管理。	
	huks	base/secu rity/huks	security _huks	向应用提供 密钥库能 力,包括密 钥管理及密 钥的密码学 操作等功 能。	如下头文件中定义的 API: base/security/huks/interfaces/ innerkits/huks_standard/main /include/*.h
升级	ota_lite	base/upda te/ota_lite	update_ ota_lite	设备远程升级。	如下头文件中定义的 API: base/startup/ota_lite/interface s/kits/*.h
编译构建	build	build productde fine/com mon	build_li te product define_ commo n	编译框架。	
公共 基础 类库	utils_lite	utils/nativ e/lite	utils_na tive_lite	为各个部件 和应用提础 库,包括存储、文件操作、定时 器、Dump 系统属性。	如下头文件中定义的 API: utils/native/lite/include/*.h
三方开源库	bounds_c hecking_ function	third_part y/bounds _checkin g_functio n	third_p arty_bo unds_c hecking _functi on	安全库函数。	
	cmsis	third_part y/cmsis	third_p arty_cm	Cortex-M 处 理器系列的	CMSIS 标准 API。

子系 统	必选部 件	源码目录	仓	功能	需要兼容的 API
			sis	与供应商无 关的硬件抽 象层。	
	littlefs	third_part y/littlefs	third_p arty_litt lefs	为微控制器 设计的小故 障安全文件 系统。	
	fatfs	third_part y/FatFs	third_p arty_Fa tFs	FatFs 是面 向小型嵌入 式系统的一 种通用的 FAT 文件系 统。	
	optimize d_routine s	third_part y/optimiz ed- routines	third_p arty_op timized _routin es	ARM 处理器 相关的库的 优化实现。	
	freebsd	third_part y/FreeBS D	third_p arty_Fr eeBSD	适用于服务 器、桌面和 嵌入式的操 作系统。	
	mbedtls	third_part y/mbedtls	third_p arty_m bedtls	嵌入式设备 上的 TLS 协 议的轻量级 实现。	
	openssl	third_part y/openssl	third_p arty_op enssl	Open SSL 工 具集,包括 libssl 和 libcrypto。	
	cjson	third_part y/cJSON	third_p arty_cJ SON	可用于 C 语 言的轻量 JSON 解 析。	/
	libcoap	third_part y/libcoap	third_p arty_lib coap	轻量级 Web 应用程序协 议。	/
	boringssl	third_part y/borings sl	third_p arty_bo ringssl	Google 创建 的 OpenSSL 分支,不能 保证 API 或	

子系统	必选部 件	源码目录	仓	功能	需要兼容的 API
				ABI 的稳定 性。	

表中"需要兼容的 API"列注明"/"表示该必选子系统当前还不涉及对应用开放的 API。三方开源库中列出的部件为 OpenHarmony 必选部件依赖的三方开源库。

【M-SOFTWARE-0101】为保证设备之间提供一致的与设备和版本相关的描述信息,系统对这些信息定义了统一的格式规范,设备开发必须遵循这些规范。设备信息规范定义如下表:

表 3-2 设备信息接口列表

API 接口	接口定义及返回值要求	举例
GetDeviceType ()	设备类型。最多 32 字符。设 备生命周期内固定。	linkiot ipcamera
GetManufactu re()	公司英文名简称。最多 32 字 符。设备生命周期内固定。	HUAWEI
GetBrand()	品牌英文名称。最多 32 字符。设备生命周期内固定。	HUAWEI
GetMarketNa me()	外部产品名称。也叫传播 名,用户可见。最多 32 字 符。设备生命周期内固定。	Mate 30
GetProductSer ies()	产品系列英文名称。用以对不同类型产品进行分类管理。最简情况下,系列下可能只有一个型号,此时可以使用型号信息代替系列。最多32字符。设备生命周期内固定。	TAS
GetProductMo del()	型号。设备关键信息之一,用以标识一类设备。用户可见,通常打印在设备铭牌上,是不同产品的区分标识。该值也是进行认证所需的关键数据,需要采用英文描述。最多32字符。设备生命周期内固定。	TAS-AL00
GetSoftwareM odel()	内部软件子型号。多硬件共 软件时区分软件分支。最多	TAS-AL00

API 接口	接口定义及返回值要求	举例
	32 字符。设备生命周期内固定。	
GetHardware Model()	硬件版本号。最多 32 字符。 设备生命周期内固定。	TASAL00CVN1
GetSerial()	设备序列号。标识设备的关键信息,在多个设备间流转业务时一般需严格保持一致,用户可见。最多 64 字符。设备生命周期内固定。	随设备差异。 如 2d5058f4*************1f258689 、FB91****E03759 等。
GetOsFullNam e()	操作系统及版本号。系统名 与版本号之间以英文字符'-' 连接,不能包含空格。最多 64字符。随软件版本变更。	OpenHarmony-3.1.1.1
GetDisplayVer sion()	用户可见的软件版本号。注 意是整个系统的软件版本号 而非 OpenHarmony 版本号。 最多 64 字符。随软件版本变 更。	1.0.0.6
GetBootloader Version()	Bootloader 版本号。最多 64 字符。随软件版本变更。	u-boot-v2019.07
GetSecurityPat chTag()	安全补丁标签。标识当前 OS 的安全补丁级别。最多 64 字 符。随软件版本变更。	2021-01-01
GetAbiList()	abi 兼容性列表。多个取值以 英文逗号','分隔,仅在有生 态的系统且系统中包含 native 应用时使用。最多 64 字符。设备生命周期内固 定。	riscv-liteos
GetSdkApiVer sion()	系统软件 API Version。设备 当前版本 API 版本,仅在有 生态的系统使用,一般是整 数。随软件版本变更。	3
GetFirstApiVe rsion()	设备首版本的系统软件 API Version。仅有生态的系统使 用,一般是整数。设备生命 周期内固定。	3
GetIncremental Version()	差异版本号。在设备型号确定的情况下,即在"设备类型"+"公司"+"品牌"+"产品系列	1.0.0.6

API 接口	接口定义及返回值要求	举例
	"+"操作系统及版本号"+"型号"+"内部硬件子型号"+"内部软件子型号"均相同的情况下,此值必须可以唯一标识软件版本。随软件版本变更。	
GetVersionId()	版本 Id。由多个字段拼接而成: \$(设备类型) + '' + \$(公司英文名简称) + '' + \$(品牌英文名称) + '' + \$(产品系列英文名称) + '' + \$(操作系统及版本号) + '' + \$(型号) + '' + \$(系统软件 API level) + '' + \$(差异版本号) + '' + \$(差异版本号) + '' + \$(差异版本号) + '' + \$(趋建类型)。在所有厂家的所有设备范围中,可以唯一标识版本。最多 127 字符。多个组成字段间以英文符号''分隔。随软件版本变更。	1、组成字段不包含分隔符: linkiot/MANU/BRAND/SERIES(AA)/ OpenHarmony- 3.1.1.1/MODEL(AA)/SM/3/1.0.0.6/rel ease 2、组成字段"型号"的取值 "M1/M2"包含了分隔符"/",须对 其做%2F 编码: linkiot/MANU/BRAND/SERIES(AA)/ OpenHarmony- 3.1.1.1/M1%2FM2/SM/3/1.0.0.6/relea se 3、组成字段"品牌"的取值"TOP ONE 品牌"包含空格或者汉字(强烈建议不要包含空格): •【建议】品牌不要取空格,汉字直接输入(要保持 utf-8 编码): linkiot/MANU/TOPONE 品牌/SERIES(AA)/OpenHarmony- 3.1.1.1/MODEL/SM/3/1.0.0.6/relea se •【可选】满足规范,对空格做%20编码,汉字直接输入(要保持 utf-8编码): linkiot/MANU/TOP%20ONE 品牌/SERIES(AA)/OpenHarmony- 3.1.1.1/MODEL/SM/3/1.0.0.6/relea se •【可选】不影响实际含义,适当突破规范,直接输入空格和汉字(utf-8编码): linkiot/MANU/TOP ONE 品牌/SERIES(AA)/OpenHarmony- 3.1.1.1/MODEL/SM/3/1.0.0.6/relea se
GetBuildType(构建类型。同一基线代码的 不同构建类型,比如 debug、release, log、nolog	release:nolog

API 接口	接口定义及返回值要求	举例
	等。可以用多个标识以英文 冒号':'分隔。最多 32 字符。 随软件版本变更。	
GetBuildUser(构建 user。最多 32 字符。随 软件版本变更。	jenkins
GetBuildHost(构建 host。最多 32 字符。随 软件版本变更。	linux
GetBuildTime(构建时间。Epoch Time,自 1970 年至今的秒数。最多 32 字符。随软件版本变更。	1294902266
GetDevUdid()	设备标识符。注意与设备序列号进行区分,设备标识符一般是基于设备序列号进一步计算后得出,两者并不相同。长度 64 字符。设备生命周期内固定。	随设备差异。比如 B7C35073***********************************

上表中:

- 1. 对于包含分隔符的返回值,为避免数据与分隔符相同进而产生歧义,需要进行编解码:
- 整体遵循 RFC3986 规范。
- 按 RFC3986 2.4. When to Encode or Decode 章节规范对带有分隔符的返回值进行百分号编解码 (注意分隔符不转码)。
- 若返回值包含多字节字符 (比如汉字), 需先采用 utf-8 编码后再按 RFC3986 规范进行编解码。
- 2. API 接口列加粗斜体说明:
- 加粗斜体表示其返回值在设备生命周期不变化,普通字体表示软件版本升级后可能变化。

【M-SOFTWARE-0102】禁止删减 productdefine/common/base/mini_system.json 定义的最小系统部件集配置。

3.1.2.2 运行时兼容性

必须遵循 6.3 运行时兼容性中定义的运行时兼容性规范。

3.1.2.3 应用兼容性

如果设备支持安装三方应用,则设备开发必须遵循 6.4 应用兼容性中定义的应用兼容性规范。

3.1.2.4 应用包格式

如果设备支持安装三方应用,则设备开发必须遵循 6.5 应用包格式中定义的应用兼容性规范。

3.1.2.5 Ark UI 框架

带屏设备如果选择集成 Ark UI 框架,则设备开发必须遵循 6.6 Ark UI 定义的兼容性规范。

3.1.3 分布式

如果采用分布式软总线协议进行组网,必须遵循 7.2 分布式软总线定义的兼容性规范。

3.1.4 性能和功耗

必须遵循 8 性能和功耗兼容性定义的兼容性规范。

3.1.5 安全

强烈建议集成 4.14 密钥和凭据系统能力并遵循其定义的兼容性规范。

强烈建议集成 4.15 应用权限管理系统能力并遵循其定义的兼容性规范。

强烈建议集成 4.16 应用完整性校验系统能力并遵循其定义的兼容性规范。

3.1.6 系统和软件升级

必须遵循 9 系统和软件升级定义的兼容性规范。

3.1.7 开发工具和开发选项

必须遵循 10 开发工具和开发选项定义的兼容性规范。

3.1.8 DFX (Design for X)

【M-DFX-0100】流水日志特指设备运行期间会持续生成的具备流水性质的日志文件, 打印流水日志必须使用 hilog 接口,并且按照接口要求定义模块标识,确保日志可追溯。

3.2 小型系统(small system)兼容性

3.2.1 硬件

3.2.1.1 CPU

【S-HARDWARE-0100】 具备 MMU。

3.2.1.2 图形显示

如果提供图形显示能力,必须遵循 4.1 图形图像定义的硬件兼容性规范。

3.2.1.3 内存和存储

必须遵循 5.1 内存和存储中定义的硬件兼容性规范。

最小内存和存储

为了定义可以运行符合 OpenHarmony 兼容性规范的最小系统,以及完成设备的基本业务功能:

【S-HARDWARE-SR-0200】设备整体内存空间≥4MiB。

【S-HARDWARE-SR-0201】设备整体可读写非易失存储空间≥16MiB。

3.2.1.4 通信

必须遵循 7.2 分布式软总线中定义的硬件兼容性规范。

3.2.1.5 摄像头

如果设备支持摄像头,必须遵循 4.8 相机定义的硬件兼容性规范。

3.2.1.6 音频

如果设备支持音频输入,必须遵循 4.7 音频定义的硬件兼容性规范。

3.2.1.7 USB

如果设备支持 USB,必须遵循 4.4 USB 服务定义的硬件兼容性规范。

3.2.1.8 加解密和安全隔离运行环境

如果设备支持硬件加解密模块,必须遵循 4.14 密钥和凭据定义的硬件加密兼容性规范。如果设备支持 TEE,必须遵循 4.14 密钥和凭据定义的硬件 TEE 兼容性规范。

3.2.2 软件

3.2.2.1 最小系统部件集和 API 兼容性

必须遵循 6.1 API 兼容性和 6.2 HDI 兼容性定义的兼容性规范。

【S-SOFTWARE-0100】必须满足 Small 系统最小部件集定义和部件对应的 API 兼容性 要求。要求如下:

表 3-3 小型最小系统部件集和 API 兼容性列表

子系统	必选部件	必选条件	源码目录	仓	功能	需要兼容的 API
内核	liteos_a 或 linux	M	kernel/lit eos_a 或 kernel/li nux	kernel_liteos_a 或 kernel_linux_p atches kernel_linux_c onfig kernel_linux_5 .10 kernel_linux_4 .19 kernel_linux_b uild	LiteOS-A 向积耗能域主A构 或 link 和个选 外、和的,流M。 以 4.10本 人.10本 人.10本 人.10两可	内核提供的 CMSIS 或POSIX 标准 API
驱动	device_ driver_fr amewor k	M	drivers/f ramewor k drivers/a dapter	drivers _frame work drivers _adapt er	HDF (Hardware Driver Foundation)驱动框 架。	
语言 编译 器运 行时	musl 或 三方运 行时库	M	third_pa rty/musl 或其他 自定义 的三方 目录	third_p arty_m usl 或三方 仓	c 运行时 库。	系统基础库 libc 提供的API
启动	syspara_	M	base/star	startup	轻量和小	如下头文件中定义的

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
恢复	lite		tup/sysp ara_lite	_syspar a_lite	型系统的 系统属性 设置、读取。	API: base/startup/syspara_lite/in terfaces/kits/parameter.h
	bootstra p_lite	M	base/star tup/boot strap_lit e	startup _bootst rap_lite	轻量和小型系统启动引导。	
	init	M	base/star tup/init_ lite	startup _init_li te	用程理含置保外理户启,权、活退和出人,和出土,和出土。	如下头文件中定义的 API: base/startup/init_lite/interf aces/kits/init_sync.h base/startup/init_lite/interf aces/kits/include/syscap_n dk.h
	appspaw n_lite	支持 Ope nHa rmo ny JS 应用	base/star tup/apps pawn_lit e	startup _appsp awn_lit e	小应器接程的化程其限用序入型用,受序命应,对,应框口系孵负应框令用设应并用架。统化责用架孵进置权调程的	
分布式线	dsoftbus	М	foundati on/com municati on/dsoft bus	commu nicatio n_dsoft bus	近间通管供链备传力。	
	ipc_lite	М	foundati on/com municati on/ipc li	commu nicatio n_ipc_l	小型系统 的进程间 通信能	如下头文件中定义的API: foundation/communication

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
			te	ite	力。	/ipc_lite/interfaces/kits/lite ipc_adapter.h foundation/communication /ipc_lite/interfaces/kits/ndk _list.h foundation/communication /ipc_lite/interfaces/kits/seri alizer.h
系统服等理	samgr_li te	M	foundati on/distri butedsch edule/sa mgr_lite	distribu tedsche dule_sa mgr_lit e	轻和小上提服 系统系 系统 系统册和 定统册和 定统明和 现能力。	
	safwk_li te	M	foundati on/distri butedsch edule/saf wk_lite	distribu tedsche dule_sa fwk_lit e	轻量系统 和小型系 统上的基 础服务进 程。	
	dmsfwk _lite	支持 Ope nHa rmo ny JS 应用	foundati on/distri butedsch edule/d msfwk_l ite	distribu tedsche dule_d ms_fw k_lite	小上设管供控组力分景用型负备理访制件,布下协系责组,问远的支式的同统跨件提和程能持场应。	
元能 力	aafwk_li te	支 持 Ope nHa rmo ny JS 应 用	foundati on/aafw k/aafwk _lite	aafwk_ aafwk_ lite	轻量和小型系统上的应用开发框架。	如下头文件中定义的API: foundation/aafwk/aafwk_li te/interfaces/kits/ability_lit e/*.h
包管 理	appexec fwk_lite	支 持 Ope	foundati on/appe xecfwk/	appexe cfwk_a ppexec	轻量和小 型安装包 管理框	如下头文件中定义的 API: foundation/appexecfwk/ap

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
		nHa rmo ny JS 应 用	appexecf wk_lite	fwk_lit e	架。	pexecfwk_lite/interfaces/ki ts/bundle_lite/*.h
ArkU I	ace_engi ne_lite	支持 Ope nHa rmo ny JS 应用 带屏	foundati on/ace/a ce_engin e_lite	ace_en gine_lit e	轻量和小型系统上的 JS UI 框架。	
DFX	hilog_fe atured_li te	M	base/hivi ewdfx/hi log_lite	hiview dfx_hil og_lite	轻量系统 和小型系 统的流水 日志功 能。	如下头文件中定义的 API: base/hiviewdfx/hilog_lite/i nterfaces/native/kits/hilog_ lite/hiview_log.h
安全	device_a uth	M	base/sec urity/dev iceauth	securit y_devi ceauth	设信建护用等周理备关立、、使撤生的明系、使撤免管	
	huks	M	base/sec urity/hu ks	securit y_huks	向供能活理的操作用钥,钥密码等为密及密件等的 计图图 计算的 计算的 计算的 计算的 计算术 化二甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基	
升级	ota_lite	М	base/upd ate/ota_l ite	update _ota_lit e	设备远程 升级。	如下头文件中定义的 API: base/startup/ota_lite/interfa ces/kits/*.h

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
编译 构建	build	M	build productd efine/co mmon	build_l ite product define_common	编译框 架。	/
公共 基础 类库	utils_lite	M	utils/nati ve/lite	utils_n ative_li te	为件提的库存件定 各和供基,储操时 中应通础包、作器 所用 括文、、 系 明性。	如下头文件中定义的 API: utils/native/lite/include/*.h
研发 工具 链	syscap_codec	支 持 Ope nHa rmo ny JS 应 用	developt ools/sys cap_cod ec	develo ptools_ syscap _codec	系统能力 编解码工 具。	
三方 开源 库	bounds_ checkin g_functi on	M	third_pa rty/boun ds_chec king_fun ction	third_p arty_bo unds_c heckin g_funct ion	安全库函 数。	
	littlefs	M	third_pa rty/littlef s	third_p arty_lit tlefs	为微控制 器设计的 小故障安 全文件系 统。	
	fatfs	M	third_pa rty/FatFs	third_p arty_Fa tFs	FatFs 是面 向小型嵌 入式系统 的一种通 用的 FAT	/

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
					文件系 统。	
	mbedtls	M	third_pa rty/mbed tls	third_p arty_m bedtls	嵌入式设 备上的 TLS 协议 的轻量级 实现。	
	openssl	M	third_pa rty/open ssl	third_p arty_op enssl	Open SSL 工具集, 包括 libssl 和 libcrypto。	
	libcoap	М	third_pa rty/libco ap	third_p arty_li bcoap	轻量级 Web 应用 程序协 议。	
	boringss 1	M	third_pa rty/borin gssl	third_p arty_bo ringssl	Google 创 建的 OpenSSL 分支,不 能保证 API 或 ABI 的 稳定性。	
	json	M	third_pa rty/json	third_p arty_js on	C++ JSON 库。	
	mksh	M	third_pa rty/mksh	third_p arty_m ksh	交互式和 Shell 脚本 的命令解 释器。	/
	toybox	M	third_pa rty/toyb ox	third_p arty_to ybox	命令行工具集。	
	selinux	M	third_pa rty/selin ux	third_p arty_se linux	安全增强 型 Linux	/
	libuv	М	third_pa rty/libuv	third_p arty_li buv	libuv 是一 个高性能 的,事件 驱动的 I/O	

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
					库,并且 提供了跨 平台(如 windows, linux)的 API。	
	nuttx	M	third_pa rty/Nutt X	third_p arty_N uttX	成熟的实 时操作系 统。	/
	freebsd	M	third_pa rty/Free BSD	third_p arty_Fr eeBSD	适用于服 务器、嵌入 面的操作 系统。	
	optimize d_routin es	M	third_pa rty/opti mized- routines	third_p arty_op timized _routin es	ARM 处理 器相关的 库的优化 实现。	
	linux	М	third_pa rty/Linu x_Kerne 1	third_p arty_Li nux_K ernel	linux 4.x 内核。	
	node	支持 Ope nHa rmo ny JS 应用 带 屏)	third_pa rty/node	third_p arty_no de	开源的跨 平台的 JavaScript 运行时环 境。	
	jerryscri pt	支 持 Ope nHa rmo ny JS	third_pa rty/jerry script	third_p arty_je rryscri pt	轻量 JavaScript 引擎。	

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
		应用带屏)				

- 1. 表中"需要兼容的 API"列注明"/"表示该必选子系统当前还不涉及对应用开放的 API。三方开源库中列出的部件为 OpenHarmony 必选部件依赖的三方开源库。
- 2. 必选条件定义部件的选择条件,"M"表示 Mandatory 强制, 描述详细选择条件的项表示满足条件则该部件必选。

【S-SOFTWARE-0101】为保证设备之间提供一致的与设备和版本相关的描述信息,系统对这些信息定义了统一的格式规范,设备开发必须遵循这些规范。设备信息规范定义如下表:

表 3-4 设备信息接口列表

API 接口	接口定义及返回值要求	举例
GetDeviceType ()	设备类型。最多 32 字符。设 备生命周期内固定。	linkiot ipcamera
GetManufactu re()	公司英文名简称。最多 32 字符。设备生命周期内固定。	HUAWEI
GetBrand()	品牌英文名称。最多 32 字符。设备生命周期内固定。	HUAWEI
GetMarketNa me()	外部产品名称。也叫传播 名,用户可见。最多32字 符。设备生命周期内固定。	Mate 30
GetProductSer ies()	产品系列英文名称。用以对不同类型产品进行分类管理。最简情况下,系列下可能只有一个型号,此时可以使用型号信息代替系列。最多32字符。设备生命周期内固定。	TAS
GetProductMo del()	型号。设备关键信息之一, 用以标识一类设备。用户可 见,通常打印在设备铭牌 上,是不同产品的区分标	TAS-AL00

API 接口	接口定义及返回值要求	举例
	识。该值也是进行认证所需的关键数据,需要采用英文描述。最多 32 字符。设备生命周期内固定。	
GetSoftwareM odel()	内部软件子型号。多硬件共 软件时区分软件分支。最多 32字符。设备生命周期内固 定。	TAS-AL00
GetHardware Model()	硬件版本号。最多 32 字符。 设备生命周期内固定。	TASAL00CVN1
GetSerial()	设备序列号。标识设备的关键信息,在多个设备间流转业务时一般需严格保持一致,用户可见。最多 64 字符。设备生命周期内固定。	随设备差异。 如 2d5058f4******************1f258689 、FB91****E03759 等。
GetOsFullNam e()	操作系统及版本号。系统名 与版本号之间以英文字符'-' 连接,不能包含空格。最多 64字符。随软件版本变更。	OpenHarmony-3.1.1.1
GetDisplayVer sion()	用户可见的软件版本号。注 意是整个系统的软件版本号 而非 OpenHarmony 版本号。 最多 64 字符。随软件版本变 更。	1.0.0.6
GetBootloader Version()	Bootloader 版本号。最多 64 字符。随软件版本变更。	u-boot-v2019.07
GetSecurityPat chTag()	安全补丁标签。标识当前 OS 的安全补丁级别。最多 64 字 符。随软件版本变更。	2021-01-01
GetAbiList()	abi 兼容性列表。多个取值以 英文逗号','分隔,仅在有生 态的系统且系统中包含 native 应用时使用。最多 64 字符。设备生命周期内固 定。	riscv-liteos
GetSdkApiVer sion()	系统软件 API Version。设备 当前版本 API 版本,仅在有 生态的系统使用,一般是整 数。随软件版本变更。	3
GetFirstApiVe	设备首版本的系统软件 API	3

API 接口	接口定义及返回值要求	举例
rsion()	Version。仅有生态的系统使用,一般是整数。设备生命周期内固定。	
GetIncremental Version()	差异版本号。在设备型号确定的情况下,即在"设备类型"+"公司"+"品牌"+"产品系列"+"操作系统及版本号"+"型号"+"内部硬件子型号"+"内部软件子型号"均相同的情况下,此值必须可以唯一标识软件版本。随软件版本变更。	1.0.0.6
GetVersionId()	版本 Id。由多个字段拼接而成: \$(设备类型) + '' + \$(公司英文名简称) + '' + \$(品牌英文名称) + '' + \$(操作系统及版本号) + '' + \$(型号) + '' + \$(系统软件 API level) + '' + \$(差异版本号) + '' + \$(差异版本号) + '' + \$(差异版本号) + '' + \$(差异版本号) + '' + \$(趋建类型)。在所有厂家的所有设备范围中,可以唯一标识版本。最多 127 字符。多个组成字段间以英文符号''分隔。随软件版本变更。	1、组成字段不包含分隔符: linkiot/MANU/BRAND/SERIES(AA)/ OpenHarmony- 3.1.1.1/MODEL(AA)/SM/3/1.0.0.6/rel ease 2、组成字段"型号"的取值 "M1/M2"包含了分隔符"/",须对 其做%2F 编码: linkiot/MANU/BRAND/SERIES(AA)/ OpenHarmony- 3.1.1.1/M1%2FM2/SM/3/1.0.0.6/relea se 3、组成字段"品牌"的取值"TOP ONE 品牌"包含空格或者汉字(强 烈建议不要包含空格): •【建议】品牌不要取空格,汉字直接输入(要保持 utf-8 编码): linkiot/MANU/TOPONE 品牌/SERIES(AA)/OpenHarmony- 3.1.1.1/MODEL/SM/3/1.0.0.6/relea se •【可选】满足规范,对空格做%20编码,汉字直接输入(要保持 utf-8 编码): linkiot/MANU/TOP%20ONE 品牌/SERIES(AA)/OpenHarmony- 3.1.1.1/MODEL/SM/3/1.0.0.6/relea se •【可选】不影响实际含义,适当突破规范,直接输入空格和汉字(utf-8 编码): linkiot/MANU/TOP ONE 品牌

API 接口	接口定义及返回值要求	举例	
		/SERIES(AA)/OpenHarmony- 3.1.1.1/MODEL/SM/3/1.0.0.6/relea se	
GetBuildType(构建类型。同一基线代码的不同构建类型,比如debug、release,log、nolog等。可以用多个标识以英文冒号':'分隔。最多 32 字符。随软件版本变更。	release:nolog	
GetBuildUser(构建 user。最多 32 字符。随 软件版本变更。	jenkins	
GetBuildHost(构建 host。最多 32 字符。随 软件版本变更。	linux	
GetBuildTime(构建时间。Epoch Time,自 1970 年至今的秒数。最多 32 字符。随软件版本变更。	1294902266	
GetDevUdid()	设备标识符。注意与设备序列号进行区分,设备标识符一般是基于设备序列号进一步计算后得出,两者并不相同。长度 64 字符。设备生命周期内固定。	随设备差异。比如 B7C35073***********************************	

上表中:

- 1. 对于包含分隔符的返回值,为避免数据与分隔符相同进而产生歧义,需要进行编解码:
- 整体遵循 RFC3986 规范。
- 按 RFC3986 2.4. When to Encode or Decode 章节规范对带有分隔符的返回值进行百分号编解码 (注意分隔符不转码)。
- 若返回值包含多字节字符(比如汉字), 需先采用 utf-8 编码后再按 RFC3986 规范进行编解码。
- 2. API 接口列加粗斜体说明:
- 加粗斜体表示其返回值在设备生命周期不变化,普通字体表示软件版本升级后可能变化。

【S-SOFTWARE-0102】禁止删減 productdefine/common/base/small_system.json 定义的最小系统部件集配置。

3.2.2.2 运行时兼容性

必须遵循 6.3 运行时兼容性中定义的运行时兼容性规范。

【S-SOFTWARE-SR-0200】建议 C++库采用 libc++, 支持 C++17 标准。

3.2.2.3 应用兼容性

必须遵循 6.4 应用兼容性中定义的应用兼容性规范。

3.2.2.4 应用包格式

必须遵循 6.5 应用包格式中定义的应用兼容性规范。

3.2.2.5 Ark UI 框架

带屏设备如果选择集成 JS UI 框架,则设备开发必须遵循 6.6 Ark UI 框架定义的兼容性规范。

3.2.3 分布式

必须遵循 7.1 分布式硬件定义的兼容性规范。

必须遵循 7.2 分布式软总线定义的兼容性规范。

必须遵循 7.4 分布式任务调度定义的兼容性规范。

3.2.4 性能和功耗

必须遵循 8 性能和功耗兼容性定义的兼容性规范。

3.2.5 安全

强烈建议集成 4.14 密钥和凭据系统能力并遵循其定义的兼容性规范。

强烈建议集成 4.15 应用权限管理系统能力并遵循其定义的兼容性规范。

强烈建议集成 4.16 应用完整性校验系统能力并遵循其定义的兼容性规范。

3.2.6 系统和软件升级

必须遵循 9 系统和软件升级定义的兼容性规范。

3.2.7 开发工具和开发选项

必须遵循 10 开发工具和开发选项定义的兼容性规范。

3.2.8 DFX (Design for X)

【S-DFX-0100】打印流水日志必须使用 hilog 接口,并且要按照接口要求定义模块标识,确保日志可追溯。

3.3 标准系统(standard system)兼容性

3.3.1 硬件

3.3.1.1 CPU

【STD-HARDWARE-0100】 具备 MMU。

3.3.1.2 GPU

【STD-HARDWARE-SR-0101】建议支持图形处理 GPU。

3.3.1.3 图形显示

如果提供图形显示能力,必须遵循 4.1 图形图像定义的硬件兼容性规范。

3.3.1.4 内存和存储

必须遵循 5.1 内存和存储中定义的硬件兼容性规范。

最小内存和存储

为了定义可以运行符合 OpenHarmony 兼容性规范的最小系统,以及完成设备的基本业务功能:

【STD-HARDWARE-SR-0200】建议设备整体内存空间≥128MiB。

【STD-HARDWARE-SR-0201】建议设备整体可读写非易失存储空间≥2GiB。

3.3.1.5 通信

必须遵循 7.2 分布式软总线中定义的硬件兼容性规范。

3.3.1.6 摄像头

如果设备支持摄像头,必须遵循 4.8 相机定义的硬件兼容性规范。

3.3.1.7 音频

如果设备支持音频输入,必须遵循 4.7 音频定义的硬件兼容性规范。

3.3.1.8 USB

如果设备支持 USB, 必须遵循 4.4 USB 服务定义的硬件兼容性规范。

3.3.1.9 加解密和安全隔离运行环境

如果设备支持硬件加解密模块,必须遵循 4.14 密钥和凭据定义的硬件加密兼容性规范。

如果设备支持 TEE, 必须遵循 4.14 密钥和凭据定义的硬件 TEE 兼容性规范。

3.3.2 软件

3.3.2.1 最小系统部件集和 API 兼容性

必须遵循 6.1 API 兼容性和 6.2 HDI 兼容性定义的兼容性规范。

【STD-SOFTWARE-0100】必须满足 Standard 系统最小部件集定义和部件对应的 API 兼容性要求。要求如下:

表 3-5 标准最小系统部件集和 API 兼容性列表

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
内核	linux	M	kernel/li nux	kernel_linux_p atches kernel_linux_c onfig kernel_linux_5 .10 kernel_linux_4 .19 kernel_linux_b uild	linux 内 核,4.19 和 5.10 两个版 本可选。	内核提供的 CMSIS 或 POSIX 标准 API
驱动	device_ driver_fr amewor k	M	drivers/f ramewor k drivers/a dapter	drivers _frame work drivers _adapt er	HDF (Hardware Driver Foundation)驱动框 架。	
语言 编译 器运 行时	musl	M	third_pa rty/musl	third_p arty_m usl	c 运行时 库。	系统基础库 libc 提供的API
启动恢复	init	M	base/star tup/init_ lite	startup _init_li te	用户态进程 启动管理, 还包含权限 设置、进程 保活和意外	如下头文件中定义的 API: base/startup/init_lite/interf aces/kits/init_sync.h base/startup/init_lite/interf

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
					退出处理。	aces/kits/include/syscap_n dk.h 如下 JS 模块定义的 API: @ohos.deviceinfo @ohos.systemparameter
	appspaw n	支持 Ope nHa rmo ny JS 应用	base/star tup/apps pawn_st andard	startup _appsp awn	应器,受理是 所负用的应设权 是是是一个, 是是一个, 是是一个, 是是一个, 是是一个, 是是一个, 是是一个, 是是一个, 是是一个, 是是一个, 是是一个, 是一个,	
分布 式软 总线	dsoftbus	M	foundati on/com municati on/dsoft bus	commu nicatio n_dsoft bus	近场设备间 分布式通信 能力管理, 提供不区分 链路和传输 发现和传输 能力。	
	ipc	M	foundati on/com municati on/ipc	commu nicatio n_ipc	提供进程间 通信能力。	如下 JS 模块定义的 API: @ohos.rpc
系统 服务 管理	samgr	M	foundati on/distri butedsch edule/sa mgr	distribu tedsche dule_sa mgr	提供系统服 务注册和发 现能力。	
	safwk	M	foundati on/distri butedsch edule/sa fwk	distribu tedsche dule_sa fwk	基础服务进程。	/
	dmsfwk	支 持 Ope nHa rmo	foundati on/distri butedsch edule/d msfwk	distribu tedsche dule_d ms_fw k	提供访问和 控制远程组 件的能力, 支持分布式	如下 JS 模块定义的 API: @ohos.DISTRIBUTEDdsc hedule

子系统	必选部件	必选条件	源码目录	仓	功能	需要兼容的 API
		ny JS 应 用			场景下的应 用协同。	
元能力	ability_r untime	支持 Ope nHa rmo ny JS 应用	foundati on/aafw k/standa rd	aafwk_ standar d	对 Ability 的命统和用支持的 和用支撑的 和用支撑的 Ability应和内力。 Ability应和内力。	如下 JS 模块定义的 API: @ohos.ability.wantconstan t @ohos.ability.featureabilit y @ohos.app.abilitymanager
	ability_b ase	支持 Ope nHa rmo ny JS 应用	foundati on/aafw k/standa rd/interf aces/inn kits	aafwk_ standar d	元能力基础 框架	
理	bundle_f ramewor k	支 持 Ope nHa rmo ny JS 应 用	foundati on/appe xecfwk/ standard	appexe cfwk_s tandard	应用程序开 发框架。	如下 JS 模块定义的 API: @ohos.bundle
DFX	hiviewdf x_hilog	М	base/hiv iewdfx/h ilog/fra mework s/native	hiview dfx_hil og	为系统内部 的服务提供 流水日志功 能。	如下头文件中定义的API: base/hiviewdfx/hilog/interfaces/native/kits/include/hilog/log.h 如下 JS 模块定义的API:

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
						@ohos.hilog
	hilog_se rvice	M	base/hiv iewdfx/h ilog/serv ices	hiview dfx_hil og	为系统和应 用提供日志 打印、记录 用户操作、 系统运行状 态等。	
	hitrace_ native	М	base/hiv iewdfx/h itrace	hiview dfx_hit race	系统内调用 链维测功 能。	/
	hiappev ent_nati ve	支 持 Ope nHa rmo ny JS 应 用	base/hiv iewdfx/h iappeve nt	hiview dfx_hia ppeven t	应用时间打点。	如下头文件中定义的 API: base/hiviewdfx/hiappevent /interfaces/native/kits/inclu de/hiapp_event.h 如下 JS 模块定义的 API: @ohos.hiAppEvent
	hisyseve nt_nativ e	M	base/hiv iewdfx/h isyseven t	hiview dfx_his ysevent	通过生生 建	
	hicollie_ native	М	base/hiv iewdfx/h icollie	hiview dfx_hic ollie	提供了软件 看门狗功 能。	/
	faultlog gerd	М	base/hiv iewdfx/f aultlogg erd	hiview dfx_fa ultlogg erd	运行时崩溃 临时日志的 生成。	/
访问 控制	access_t oken	支 持 Ope nHa rmo ny JS 应	base/sec urity/acc ess_toke n	securit y_acce ss_toke n	程序权限的 访问控制, 控制者程序 是否有权限 访问相应的资源。	如下 JS 模块定义的 API: @ohos. abilityAccessCtrl

子系 统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
古州	arranth a	用士	farmiati		田工从畑外	加工及特种点义的
事件 通知	eventha ndler	支 持 Ope nHa rmo ny JS 应 用	foundati on/appe xecfwk/ standard /libs/libe venthan dler	appexe cfwk_s tandard	用于阻性 一种 是异 力 比	如下 JS 模块定义的 API: @ohos.events.emitter
公共 基础 类库	utils_bas e	М	utils/nati ve	utils_n ative	提供了一些 常用的 C、 C++开发增 强 API。	如下 JS 模块定义的 API: @ohos.utils
ArkU	ace_engi ne	支持 Ope nHa rmo ny JS 应用 带屏)	foundati on/ace/a ce_engi ne	ace_ac e_engi ne	开提类类等和CSS力Weba。 以I 准动,beba。 Weba。 以I 推动,beba。 然如 类编	1. JS 模块定义的 API: @system.app @ohos.app @system.configuration @ohos.configuration @ohos.router @ohos.router @ohos.prompt 2. 內置 JS API: console.debug log info war n error(message) setTimeout(handler[, delay[,args]]) clearTimeout(timeoutID) setInterval(handler[, delay[,args]]) clearInterval(intervalID) 3. JS UI 组件: div、list、list-item、 stack、swiper、chart、 image、image-animator、 input、marquee、picker- view、progress、 qrcode、slider、switch、

子系 统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
Ark	ark_js_r untime	支持 Ope nHa rmo ny JS 应用 带 屏	ark/js_ru ntime	ark_js_ runtim e	方舟编译器 JS 运行 时。	text /
	arkcomp iler_runt ime) 支持 Ope nHa rmo ny JS 应用带屏)	ark/runti me_core	ark_ru ntime_ core	方舟运行时 的公共组 件。	
	arkcomp iler_fron tend	支持 Ope nHa rmo ny JS 应用 带屏)	ark/ts2a bc	ark_ts2 abc	方舟运行时 子系红具, 持将 JavaScript 文件转为 方舟文件。	
安全	huks	M	base/sec urity/hu ks	securit y_huks	向应用提供 密钥库能 力,包括密 钥管理及密 钥的密码学 操作等功	/

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
					能。	
	device_a uth	M	base/sec urity/de viceauth	securit y_devi ceauth	设备间可信 关系维护、 使用、撤销 等全生命理。 期的管理。	
升级	updater	М	base/upda ate/upda ter	update _updat er	升级工具,解析和校验升级包,启动升级流程。	如下头文件中定义的 API: base/update/updater/interfa ces/kits/include/misc_info/ *.h base/update/updater/interfa ces/kits/include/package/*. h base/update/updater/interfa ces/kits/include/updatekits/ *.h
	update_s ervice	M	base/upd ate/upda teservice	update _updat eservic e	升级服务, 提供的有数。 用的、包包取和 级包取和触发 , 大级。 升级。	如下 JS 模块定义的 API: @ohos.update.d.ts
编译 构建	build	M	build productd efine/co mmon	build build_1 ite product define_ commo n	编译框架。	
研发工具 链	syscap_codec	支 持 Ope nHa rmo ny JS 应	developt ools/sys cap_cod ec	develo ptools_ syscap _codec	系统能力编 解码工具。	

子系统	必选部件	必选条件	源码目录	仓	功能	需要兼容的 API
		用				
三方开源库	bounds_ checkin g_functi on	M	third_pa rty/boun ds_chec king_fu nction	third_p arty_bo unds_c heckin g_funct ion	安全库函数。	/
	mbedtls	M	third_pa rty/mbe dtls	third_p arty_m bedtls	嵌入式设备 上的 TLS 协议的轻量 级实现。	/
	openssl	M	third_pa rty/open ssl	third_p arty_op enssl	Open SSL 工具集,包 括 libssl 和 libcrypto。	/
	cjson	M	third_pa rty/cJSO N	third_p arty_cJ SON	可用于 C 语言的轻量 JSON 解 析。	/
	libcoap	M	third_pa rty/libco ap	third_p arty_li bcoap	轻量级 Web 应用 程序协议。	/
	boringss 1	M	third_pa rty/borin gssl	third_p arty_bo ringssl	Google 创 建的 OpenSSL 分支,不能 保证 API 或 ABI 的 稳定性。	
	json	M	third_pa rty/json	third_p arty_js on	C++ JSON 库。	/
	mksh	М	third_pa rty/mksh	third_p arty_m ksh	交互式和 Shell 脚本 的命令解释 器。	
	toybox	M	third_pa rty/toyb ox	third_p arty_to ybox	命令行工具 集。	

子系统	必选部 件	必选条件	源码目录	仓	功能	需要兼容的 API
	selinux	M	third_pa rty/selin ux	third_p arty_se linux	安全增强型 Linux	/
	libuv	M	third_pa rty/libuv	third_p arty_li buv	libuv 是一 个高性能 的,事件驱 动的 I/O 库,并且提 供了跨平台 (如 windows, linux)的 API。	
	libxml2	M	third_pa rty/libx ml2	third_p arty_li bxml2	XML 的 C 语言解释 器。	/
	bzip2	M	third_pa rty/bzip 2	third_p arty_bz ip2	数据压缩工 具。	/
	lz4	M	third_pa rty/lz4	third_p arty_lz 4	提供 LZ4 数据格式的 压缩和解压 缩接口。	/
	libdrm	M	third_pa rty/libdr m	third_p arty_li bdrm	用户态访问 DRM 的 库。	/
	e2fsprog s	M	third_pa rty/e2fsp rogs	third_p arty_e2 fsprogs	ext2/ext3/ex t4 文件系 统工具集。	/
	libpng	支持 Ope nHa rmo ny JS 应用 借 屏	third_pa rty/libpn g	third_p arty_li bpng	PNG 标准 库。	

子系统	必选部件	必选条件	源码目录	仓	功能	需要兼容的 API
)				
	jsoncpp	支持 Ope nHa rmo ny JS 应用	third_pa rty/jsonc pp	third_p arty_js oncpp	序列化反序 列 JSON 格 式的开源 C++库。	
	zlib	支持 Ope nHa rmo ny JS 应用	third_pa rty/zlib	third_p arty_zli b	数据压缩 库。	
	icu	支持 Ope nHa rmo ny JS 应用	third_pa rty/icu	third_p arty_ic u	提供 C、 C++和 JAVA 操作 Unicode 的 函数。	
	node	支持 Ope nHa rmo ny JS 应用 带屏)	third_pa rty/node	third_p arty_no de	开源的跨平 台的 JavaScript 运行时环 境。	
	libjpeg	支 持 Ope	third_pa rty/libjp eg	third_p arty_li bjpeg	读写 JPEG 图片文件的 C 库。	/

子系 统	必选部 件	必选条件	源码目 录	仓	功能	需要兼容的 API
		nHa rmo ny JS 应 用 (帯 屏)				
	benchm ark	支持 Ope nHa rmo ny JS 应用	third_pa rty/benc hmark	third_p arty_be nchmar k	性能测试库。	
	libunwin d	支持 Ope nHa rmo ny JS 应用	third_pa rty/libun wind	third_p arty_li bunwin d	获取程序的 调用栈和异 常处理和跳 转。	
	ejdb	支 持 Ope nHa rmo ny JS 应 用	third_pa rty/ejdb	third_p arty_ej db	嵌入式 JSON 数据 库引擎。	
	iowow	支 持 Ope nHa rmo ny JS 应	third_pa rty/iowo w	third_p arty_io wow	基于跳跃表的 KV 存储。	

子系 统	必选部 件	必选条件	源码目 录	仓	功能	需要兼容的 API
		用				
	sqlite	支 持 Ope nHa rmo ny JS 应 用	third_pa rty/sqlite	third_p arty_sq lite	轻量的 SQL 数据 库引擎。	
	jsframe work	支持 Ope nHa rmo ny JS 应用 带 屏)	third_pa rty/jsfra mework	third_p arty_jsf ramew ork	构建移动跨 平台 UI 的 框架。	
	boost	支 持 Ope nHa rmo ny JS 应 用	third_pa rty/boost	third_p arty_bo ost	提供可移植的同行评议过的 C++库。	
	quickjs	支持 Ope nHa rmo ny JS 应用 带 屏)	third_pa rty/quic kjs	third_p arty_qu ickjs	轻量的嵌入 式 JavaScript 引擎。	

子系统	必选部件	必选条件	源码目录	仓	功能	需要兼容的 API
	curl	支 持 Ope nHa rmo ny JS 应 用	third_pa rty/curl	third_p arty_cu rl	支持数据以 URL 语法 传输的库和 命令行工 具。	
	glfw	支 持 Ope nHa rmo ny JS 应 用	third_pa rty/glfw	third_p arty_gl fw	OpenGL 的 多平台库。	
	expat	支 持 Ope nHa rmo ny JS 应 用	third_pa rty/expat	third_p arty_ex pat	C编写的 XML解释 器库。	

□ 说明

- 1. 表中"需要兼容的 API"列注明"/"表示该必选子系统当前还不涉及对应用开放的 API。三方开源库中列出的部件为 OpenHarmony 必选部件依赖的三方开源库。
- 2. 必选条件定义部件的选择条件,"M"表示 Mandatory 强制,描述详细选择条件的项表示满足条件则该部件必选。

【STD-SOFTWARE-0101】为保证设备之间提供一致的与设备和版本相关的描述信息,系统对这些信息定义了统一的格式规范,设备开发必须遵循这些规范。设备信息规范定义如下表:

表 3-6 设备信息接口列表

API 接口	接口定义及返回值要求	举例
GetDeviceType	设备类型。最多32字符。设	linkiot

API 接口	接口定义及返回值要求	举例
0	备生命周期内固定。	ipcamera
GetManufactu re()	公司英文名简称。最多 32 字符。设备生命周期内固定。	HUAWEI
GetBrand()	品牌英文名称。最多 32 字符。设备生命周期内固定。	HUAWEI
GetMarketNa me()	外部产品名称。也叫传播 名,用户可见。最多 32 字 符。设备生命周期内固定。	Mate 30
GetProductSer ies()	产品系列英文名称。用以对不同类型产品进行分类管理。最简情况下,系列下可能只有一个型号,此时可以使用型号信息代替系列。最多32字符。设备生命周期内固定。	TAS
GetProductMo del()	型号。设备关键信息之一,用以标识一类设备。用户可见,通常打印在设备铭牌上,是不同产品的区分标识。该值也是进行认证所需的关键数据,需要采用英文描述。最多32字符。设备生命周期内固定。	TAS-AL00
GetSoftwareM odel()	内部软件子型号。多硬件共 软件时区分软件分支。最多 32字符。设备生命周期内固 定。	TAS-AL00
GetHardware Model()	硬件版本号。最多 32 字符。 设备生命周期内固定。	TASAL00CVN1
GetSerial()	设备序列号。标识设备的关键信息,在多个设备间流转业务时一般需严格保持一致,用户可见。最多 64 字符。设备生命周期内固定。	随设备差异。 如 2d5058f4************************************
GetOsFullNam e()	操作系统及版本号。系统名 与版本号之间以英文字符'-' 连接,不能包含空格。最多 64字符。随软件版本变更。	OpenHarmony-3.1.1.1
GetDisplayVer sion()	用户可见的软件版本号。注 意是整个系统的软件版本号	1.0.0.6

API 接口	接口定义及返回值要求	举例
	而非 OpenHarmony 版本号。 最多 64 字符。随软件版本变 更。	
GetBootloader Version()	Bootloader 版本号。最多 64 字符。随软件版本变更。	u-boot-v2019.07
GetSecurityPat chTag()	安全补丁标签。标识当前 OS 的安全补丁级别。最多 64 字 符。随软件版本变更。	2021-01-01
GetAbiList()	abi 兼容性列表。多个取值以 英文逗号','分隔,仅在有生 态的系统且系统中包含 native 应用时使用。最多 64 字符。设备生命周期内固 定。	riscv-liteos
GetSdkApiVer sion()	系统软件 API Version。设备 当前版本 API 版本,仅在有 生态的系统使用,一般是整 数。随软件版本变更。	3
GetFirstApiVe rsion()	设备首版本的系统软件 API Version。仅有生态的系统使 用,一般是整数。设备生命 周期内固定。	3
GetIncremental Version()	差异版本号。在设备型号确定的情况下,即在"设备类型"+"公司"+"品牌"+"产品系列"+"操作系统及版本号"+"型号"+"内部硬件子型号"+"内部软件子型号"均相同的情况下,此值必须可以唯一标识软件版本。随软件版本变更。	1.0.0.6
GetVersionId()	版本 Id。由多个字段拼接而成: \$(设备类型) + '/' + \$(公司英文名简称) + '/' + \$(品牌英文名称) + '/' + \$(产品系列英文名称) + '/' + \$(操作系统及版本号) + '/' + \$(内部软件子型号) + '/' + \$(系统软件 API level) + '/' + \$(差异版本号) + '/' + \$(构建类型)。在所有厂家的所有设备范围中,可以唯一标识版	1、组成字段不包含分隔符: linkiot/MANU/BRAND/SERIES(AA)/ OpenHarmony- 3.1.1.1/MODEL(AA)/SM/3/1.0.0.6/rel ease 2、组成字段"型号"的取值 "M1/M2"包含了分隔符"/",须对 其做%2F 编码: linkiot/MANU/BRAND/SERIES(AA)/ OpenHarmony- 3.1.1.1/M1%2FM2/SM/3/1.0.0.6/relea

API 接口	接口定义及返回值要求	举例
	本。最多 127 字符。多个组成字段间以英文符号'/'分隔。随软件版本变更。	se 3、组成字段"品牌"的取值"TOP ONE 品牌"包含空格或者汉字(强 烈建议不要包含空格): •【建议】品牌不要取空格,汉字直接输入(要保持 utf-8 编码): linkiot/MANU/TOPONE 品牌 /SERIES(AA)/OpenHarmony- 3.1.1.1/MODEL/SM/3/1.0.0.6/relea se •【可选】满足规范,对空格做%20 编码,汉字直接输入(要保持 utf-8 编码): linkiot/MANU/TOP%20ONE 品牌 /SERIES(AA)/OpenHarmony- 3.1.1.1/MODEL/SM/3/1.0.0.6/relea se •【可选】不影响实际含义,适当突破规范,直接输入空格和汉字 (utf-8 编码): linkiot/MANU/TOP ONE 品牌 /SERIES(AA)/OpenHarmony- 3.1.1.1/MODEL/SM/3/1.0.0.6/relea se
GetBuildType(构建类型。同一基线代码的不同构建类型,比如debug、release, log、nolog等。可以用多个标识以英文冒号':'分隔。最多 32 字符。随软件版本变更。	release:nolog
GetBuildUser(构建 user。最多 32 字符。随 软件版本变更。	jenkins
GetBuildHost(构建 host。最多 32 字符。随 软件版本变更。	linux
GetBuildTime(构建时间。Epoch Time,自 1970 年至今的秒数。最多 32 字符。随软件版本变更。	1294902266
GetDevUdid()	设备标识符。注意与设备序列号进行区分,设备标识符一般是基于设备序列号进一步计算后得出,两者并不相同。长度 64 字符。设备生命	随设备差异。比如 B7C35073***********************************

API 接口	接口定义及返回值要求	举例
	周期内固定。	A055

□ 说明

上表中:

- 1. 对于包含分隔符的返回值,为避免数据与分隔符相同进而产生歧义,需要进行编解码:
- 整体遵循 RFC3986 规范。
- 按RFC3986 2.4. When to Encode or Decode 章节规范对带有分隔符的返回值进行百分号编解码 (注意分隔符不转码)。
- 若返回值包含多字节字符(比如汉字), 需先采用 utf-8 编码后再按 RFC3986 规范进行编解码。
- 2. API 接口列加粗斜体说明:
- **加租斜体**表示其返回值在设备生命周期不变化,普通字体表示软件版本升级后可能变化。

【STD-SOFTWARE-0102】禁止删减 productdefine/common/base/standard_system.json 定义的最小系统部件集配置。

3.3.2.2 运行时兼容性

必须遵循 6.3 运行时兼容性中定义的运行时兼容性规范。

【STD-SOFTWARE-SR-0200】建议 C++库采用 libc++, 支持 C++17 标准。

【STD-SOFTWARE-0200】为了方便应用的书写,JS 运行时必须支持 ES2017 中的 async/await 特性。

3.3.2.3 应用兼容性

必须遵循 6.4 应用兼容性中定义的应用兼容性规范。

【STD-SOFTWARE-0300】任何替代 ohos 核心应用程序的版本都必须遵守 ohos 核心应用程序提供的相同 want,系统实现者必须支持 ohos SDK 提供的所有 want。

【STD-SOFTWARE-0301】系统实现者不应对使用 ohos SDK 提供的 want 的系统应用附加任何特权,不应阻止三方应用绑定并承担对这些 want 的控制。

【STD-SOFTWARE-0302】ohos 设备必须发送公共事件 want 以响应适当的系统事件来通知三方应用硬件或软件环境的改变。

□ 说明

Want 定义了一个 Ability 的启动信息格式,本地或者跨设备调用 Ability 时,需要按此格式封装启动信息,并通过 start Ability/connect Ability 接口携带。

【STD-SOFTWARE-0303】系统实现者不应在 commonevent 命名空间中添加新的公共事件类型,不应更改或扩展 ohos 核心应用使用的任何公共事件。

3.3.2.4 应用包格式

必须遵循 6.5 应用包格式中定义的应用兼容性规范。

3.3.2.5 JS UI 框架

带屏设备如果选择集成 JS UI 框架,则设备开发必须遵循 6.6 Ark UI 框架定义的兼容性规范。

3.3.3 分布式

必须遵循 7.1 分布式硬件定义的兼容性规范。

必须遵循 7.2 分布式软总线定义的兼容性规范。

必须遵循 7.4 分布式任务调度定义的兼容性规范。

3.3.4 性能和功耗

必须遵循8性能和功耗兼容性定义的兼容性规范。

3.3.5 安全

强烈建议集成 4.14 密钥和凭据系统能力并遵循其定义的兼容性规范。

强烈建议集成 4.15 应用权限管理系统能力并遵循其定义的兼容性规范。

强烈建议集成 4.16 应用完整性校验系统能力并遵循其定义的兼容性规范。

3.3.6 系统和软件升级

必须遵循 9 系统和软件升级定义的兼容性规范。

3.3.7 开发工具和开发选项

必须遵循 10 开发工具和开发选项定义的兼容性规范。

开发工具

【STD-TOOLS-0100】必须支持使用 hdc 设备连接器提供的命令行和通信协议与设备进行交互。

3.3.8 DFX (Design for X)

【STD-DFX-0100】打印流水日志必须使用 hilog 接口,并且要按照接口要求定义模块标识,确保日志可追溯。

可选系统能力兼容性

OpenHarmony 提供了一系列可选的系统部件,方便设备开发者按需配置,以支撑其特色功能的扩展或定制开发。系统将这些可选的系统部件组合为一系列描述为特性或功能的系统能力,以方便设备开发者理解和选择。本章为每种可选择的系统能力定义兼容性规范。

如果系统中集成了某个系统能力,则必须遵循该系统能力定义的兼容性规则,并强烈 建议考虑其相应的兼容性建议。

- 4.1 图形图像
- 4.2 电池管理服务
- 4.3 电源管理服务
- 4.4 USB 服务
- 4.5 AI 引擎
- 4.6 媒体
- 4.7 音频
- 4.8 相机
- 4.9 图像编解码
- 4.10 电路域电话短信服务
- 4.11 通话管理服务
- 4.12 输入法服务框架
- 4.13 定时&时间时区服务
- 4.14 密钥和凭据
- 4.15 应用权限管理
- 4.16 应用完整性校验
- 4.17 多模输入服务

4.1 图形图像

显示能力

如果提供显示能力:

【C-ALL-HARDWARE-0200】必须在硬件 profile 中设置支持显示的 display 字段标识。

支持 2D 和 3D 加速

如果支持图形显示,并且支持图形的 2D 或 3D 硬件加速:

【C-ALL-HARDWARE-0201】默认情况下必须启用硬件加速。

如果使用 Linux 内核并且提供 3D 的硬件加速能力:

【C-ALL-HARDWARE-0202】必须同时支持 OpenGL ES1.1 和 OpenGL ES2.0。

【C-ALL-HARDWARE-0203】必须支持可在 libGLESv2.so 库中导出相应的 OpenGL ES 2.0 定义的函数符号。

4.2 电池管理服务

【C-STD-SOFTWARE-0300】如果设备支持电池,则必须提供电池电量信息和充放电状态的查询能力和接口。

4.3 电源管理服务

【C-STD-SOFTWARE-0400】必须提供系统休眠、唤醒和低功耗管理能力。

【C-STD-SOFTWARE-0401】必须在用户明确执行设备进入不活动状态的操作之后进入休眠状态。

【C-STD-SOFTWARE-0402】如果设备已实现 ACPI 中所定义的 S3/S4 电源状态,必须满足仅在应用不需要系统资源时进入 S3/S4 状态。当应用需要系统资源时,则必须退出 S3/S4 状态。

4.4 USB 服务

USB device mode

如果设备实现支持 USB device 模式:

【C-ALL-HARDWARE-0500】端口必须可连接到具有标准A型或C型的主机模式的USB端口。

【C-ALL-HARDWARE-0501】必须通过 GetHardwareProfile()中在 USB 标准设备描述符中报告 iSerialNumber 的正确值。

USB host mode

当设备实现支持主机模式的 USB 端口:

【C-ALL-HARDWARE-0502】必须支持连接标准 USB 外围设备。

【C-ALL-HARDWARE-0503】必须通过 GetHardwareProfile()中 usbhost 提供是否支持 USBHOST 的标记。

4.5 AI 引擎

【C-S-SOFTWARE-0600】AI sdk 需按算法调用顺序, 封装 client 对外提供接口; 对于异步插件对应的 sdk, 需要实现 client 提供的回调接口 IClientCb。

【C-S-SOFTWARE-0601】AI sdk 接口实现中,需要保存与 client 交互的相关通用数据。

【C-S-SOFTWARE-0602】AI sdk 与 plugin 需要使用编解码模块,将特定算法数据转换成 AI 引擎的通用数据类型。

【C-S-SOFTWARE-0603】AI sdk 中需要对以编解码返回的出参数据类型进行内存释放。

【C-S-SOFTWARE-0604】AI plugin 需要实现 server 提供的 IPlugin 接口。

【C-S-SOFTWARE-0605】AI plugin 需要使用 AI 引擎提供的统一数据通道。

如果提供视频输入能力:

【C-ALL-SOFTWARE-SR-0600】强烈推荐支持 CV 能力。

如果提供语音输入能力:

【C-ALL-SOFTWARE-SR-0601】强烈推荐支持 ASR 能力。

4.6 媒体

【C-S|STD-SOFTWARE-0700】必须支持 PCM/WAV 格式的音频编码,编码采样频率必须支持 8K, 16K, 44.1K, 48K。

【C-S|STD-SOFTWARE-0701】必须支持 MPEG-2 AAC Main/MPEG-2 AAC LC (Low Complexity)格式的音频编码,编码采样频率支持 16K, 44.1K, 48K, 可选支持 11.025K, 22.05K, 48K, 176.4K, 192K。

【C-STD-SOFTWARE-0702】必须支持 MPEG-4 AAC Profile (AAC LC)格式解码,支持单/双声道,支持 8~48KHz 采样率,支持 8/16/24bit 位深,建议支持 32bit 位深,支持 MPEG-4(.mp4,.m4a)容器格式。

【C-STD-SOFTWARE-0703】必须支持 H.264 格式 Baseline Profile Level 3 编码规格,必须支持 SD(尺寸小于 720×480)编码规格,应该支持 HD(尺寸大于 1280×720)编码规格。

【C-STD-SOFTWARE-0704】必须支持 H.264 格式 Baseline Profile/Main Profile/HighProfile 解码,MPEG-4(.mp4)容器格式。

【C-S|STD-SOFTWARE-SR-0701】建议支持 MPEG-4 AAC LD (Low Delay)/MPEG-4 AAC HE (High Efficiency) AACPlusV1/V2(3GPP)格式的音频编码。

4.7 音频

麦克风硬件兼容性

如果设备支持麦克风:

【C-ALL-HARDWARE-0800】必须通过 GetHardwareProfile()中 microphone 提供是否支持麦克风的标记,以支持从配置信息判断设备是否支持麦克风硬件。

【C-ALL-HARDWARE-0801】必须支持单或双 MIC 输入或 Line-In 输入的一种或多种方式。

【C-ALL-HARDWARE-0802】必须支持 16bit 音频输入。

音频输出硬件兼容性

如果设备实现包括扬声器或者其他用于音频输出的音频/多媒体输出端口外设,例如 3.5mm 音频插孔,HDMI,或使用 USB 音频类的 USB 主机模式端口。其他类型的音频输出端口,例如支持基于无线电的音频输出和蓝牙等音频输出不在本章节定义。

【C-ALL-HARDWARE-0803】必须通过 GetHardwareProfile()中 aout 提供是否支持音频输出的标识,以支持从配置信息判断设备支持的音频输出端口种类。

【C-S|STD-HARDWARE-SR-0800】强烈建议支持低时延耳返特性。

【C-S|STD-HARDWARE-SR-0801】强烈建议支持 3A(Acoustic Echo Cancelling、Automatic Gain Control、Active Noise Control)特性。

软件兼容性

【C-S|STD-SOFTWARE-0800】音频播放必须支持 16K, 44.1K, 48K 采样率音频。

【C-S|STD-SOFTWARE-0801】音频播放必须支持单声道和双声道音频。

【C-S|STD-SOFTWARE-0802】音频播放必须支持 16 位 PCM 编码格式音频 。

【C-S|STD-SOFTWARE-0803】音频采集必须支持单声道、编码格式为 16 位 PCM、采样率为 16K,48K 的音频。

【C-S|STD-SOFTWARE-0804】必须支持 MP3 格式解码, 支持单/双声道, 支持 8~320kbps 的固定码率、变码率模式, 支持 MP3(.mp3)容器格式。

4.8 相机

硬件兼容性

【C-ALL-HARDWARE-0900】必须通过 GetHardwareProfile()中 camera 字段提供是否支持摄像头硬件的标记,以支持从配置信息判断设备是否支持摄像头硬件。

视频输入硬件兼容性

【C-ALL-HARDWARE-0901】必须支持 8-/10-/12-bit RGB Bayer DC timing VI 其中的一种或多种模式。

【C-ALL-HARDWARE-0902】必须支持 MIPI, LVDS/sub-LVDS, and HiSPi 一种或多种模式。

ISP 硬件兼容性

【C-ALL-HARDWARE-0904】必须支持 3A(Auto Exposure/Auto Focus/Auto White Balance)调节。

软件兼容性

【C-S|STD-SOFTWARE-0900】必须提供摄像头设备能力查询功能,包含摄像头个数,方向,类型,宽高比和分辨率信息。

【C-S|STD-SOFTWARE-0901】必须提供摄像头数据流采集功能,包含预览流、录像流和拍照流。预览流、录像流必须支持 YUV420SP 格式,拍照流必须支持 JPEG 编码格式。

【C-S|STD-SOFTWARE-0902】建议提供摄像头 3A(Auto Exposure/Auto Focus/Auto White Balance)能力进行开关控制和模式切换的功能。

4.9 图像编解码

【C-STD-SOFTWARE-1000】必须支持 JPEG 格式编码。

【C-S-SOFTWARE-1001】必须支持 PNG 格式编码。

【C-S-SOFTWARE-1002】必须支持 WebP 格式编码。

【C-S-SOFTWARE-1003】必须支持 JPEG 格式解码,支持基础和渐进式解码,支持 JPEG(.jpg)容器格式。

【C-S-SOFTWARE-1004】必须支持 GIF 格式解码,支持静态和动态 GIF,支持 GIF(.gif) 容器格式。

【C-S-SOFTWARE-1005】必须支持 PNG 格式解码,支持 PNG(.png)容器格式。

【C-S-SOFTWARE-1006】必须支持 BMP 格式解码,支持 BMP(.bmp)容器格式。

【C-S-SOFTWARE-1007】必须支持 WebP 格式解码,支持 WebP(.webp)容器格式。

【C-S-SOFTWARE-SR-0100】建议支持 HEIF(HEIC)格式解码,支持 HEIF(.heif)、HEIF(.heic)容器格式。

【C-S-SOFTWARE-1008】必须支持 Raw 格式,支持 ARW (.arw)、CR2 (.cr2)、DNG (.dng)、NEF (.nef)、NRW (.nrw)、ORF (.orf)、PEF (.pef)、RAF (.raf)、RW2 (.rw2)、SRW (.srw)容器格式。

4.10 电路域电话短信服务

【C-STD-SOFTWARE-1100】必须提供电路域电话核心服务包含的 SIM 卡和搜网能力的完整实现。

□说明

- 如果设备实现包含电话硬件,则必须至少支持电话核心服务包含的 SIM 卡和搜网 API 实现, 其他能力视设备规格决定是否支持,比如语音通话、短信、蜂窝数据等。
- 2) 如果设备实现不包含电话硬件,则电话核心服务相关 API 可以为空实现。

4.11 通话管理服务

【C-STD-SOFTWARE-1200】必须提供通话管理功能的完整实现。

□□说明

通话管理服务主要负责管理 CS(Circuit Switch, 电路交换)、IMS(IP Multimedia Subsystem, IP 多媒体子系统)和 OTT(over the top, OTT 解决方案)三种类型的通话。

- 1)如果设备支持通话功能(CS通话、IMS通话、OTT通话中的一种或多种),则必须支持通话管理服务的API实现。
- 2) 如果设备不支持通话功能,则通话管理服务相关 API 可以为空实现。

4.12 输入法服务框架

【C-STD-SOFTWARE-1300】必须支持第三方输入法应用。

【C-STD-SOFTWARE-1301】必须完整实现输入法管理框架。

【C-STD-SOFTWARE-1302】必须预安装英文输入法应用。

4.13 定时&时间时区服务

【C-STD-SOFTWARE-1400】必须限制通过定时器拉起已经退出的进程的行为。

□ 说明

定时器支持以回调或悬挂意图的方式通知创建该定时器的服务或者应用进行超时处理。系统应该限制定时器拉起已经退出的进程的行为,比如应用/服务通过定时器随意拉起非常驻的后台服务或者弹窗行为,以避免对用户体验或者设备功耗造成影响。

【C-STD-SOFTWARE-1401】必须通过权限管控修改系统时间时区的行为。

4.14 密钥和凭据

硬件兼容性

如果设备支持硬件加解密模块,必须支持以下规范:

【C-ALL-HARDWARE-SR-1500】强烈推荐支持硬件 AES256 加解密算法。

【C-ALL-HARDWARE-1501】必须支持硬件 HASH-SHA256、HMAC SHA256 算法。

【C-ALL-HARDWARE-1502】必须支持硬件 RSA、ECC 签名校验算法。

【C-ALL-HARDWARE-1503】必须支持硬件真随机数生成,满足 FIPS140-2 随机测试标准。

如果设备支持 TEE, 硬件必须支持以下规范:

【C-ALL-HARDWARE-1504】CPU 必须支持安全和非安全状态。

【C-ALL-HARDWARE-1505】如果 CPU class 为 cortex-A,必须支持 TZASC 安全内存配置和 TZPC 安全外设配置。

【C-ALL-HARDWARE-1505】如果 CPU class 为 cortex-M,必须支持 SAU 或 IDAU 安全内存配置。

软件兼容性

【C-ALL-SOFTWARE-1500】必须支持安全随机数。

【C-ALL-SOFTWARE-1501】必须支持 HASH 算法,至少包括 SHA256。

【C-ALL-SOFTWARE-1502】必须支持 HMAC 和 HKDF 算法,至少包括 HMAC-SHA256。

【C-ALL-SOFTWARE-1503】必须支持 AES 对称加密算法,分组模式支持 CBC、GCM,密钥长度支持 128、192、256 位。

【C-S|STD-SOFTWARE-1504】必须支持 AES 对称加密算法,分组模式支持 ECB、OFB、CFB、CTR、CCM、密钥长度支持 128、192、256 位。

【C-S|STD-SOFTWARE-1505】必须支持非对称加密算法 RSA,包括 2048 位、3072 位、4096 位,能够实现秘钥管理、加解密以及签名服务。

【C-S|STD-SOFTWARE-1506】支持椭圆曲线签名算法 ECDSA 和秘钥协商算法 ECDH,至少支持 P256 曲线;支持椭圆曲线签名算法 ED25519 和秘钥协商算法 X25519、25519 曲线。

4.15 应用权限管理

【C-S|STD-SOFTWARE-1600】必须支持权限模型,并且执行权限模型文档中定义的每个权限,不应省略、更改或忽略任何权限。

【C-STD-SOFTWARE-1601】禁止应用申请 restricted 权限,除非应用包中含有 restricted 权限的权限证书。

【C-S|STD-SOFTWARE-1602】必须为用户提供专用的界面或接口用于授权和管理动态权限(user grant),必要时用户可以撤销授权。

【C-S|STD-SOFTWARE-1603】禁止给任何的预装应用授予任何的动态权限(user grant)。

□ 说明

如下情况例外:

- * 预装应用使用动态权限前获得用户同意;
- * 预装应用为动态权限相关的默认应用,例如,短信动态权限允许授予给默认短信应用。

【C-STD-SOFTWARE-1604】设备必须为用户提供专门的界面或接口用于授权和管理应用访问其他设备的能力

【C-STD-SOFTWARE-1605】分布式场景下,主体设备(分布式业务发起方)必须为用户提供专门的界面或接口用于授权和管理应用访问其他设备的能力。

4.16 应用完整性校验

【C-ALL-SOFTWARE-1700】必须对应用进行签名,用于应用完整性和来源验证。

【C-ALL-SOFTWARE-1701】在安装应用时,必须对应用的签名进行校验,确保应用来源可信和未被篡改,禁止安装签名校验失败的应用。

4.17 多模输入服务

【C-STD-SOFTWARE-1800】电源键、Home 键、静音键必须系统处理,禁止发送给应用,返回键必须发送到应用处理。

【C-STD-SOFTWARE-1801】禁止修改触摸输入事件和按键输入事件的投递规则。

4.18 Ability assistant 工具

【C-ALL-SOFTWARE-1900】必须遵守 aa 工具命令行语义。

5 硬件兼容性

5.1 内存和存储

5.1 内存和存储

内存,简称为 RAM。

存储, 定义为非易失性存储, 分成以下几种类型:

- 1. ROM,只读存储器。
- 2. Writeable NVM(nonvolatile memory,以下简称 NVM),芯片内或芯片外可读写非易失存储器,例如片内内置 Flash 或外接 Flash 器件。
- 3. portable NVM,用于外部可拆卸存储器,如 SDCard。

【G-HARDWARE-0100】外部可拆卸存储必须通过 mount 方式安装到系统根目录,例如 SDCard 必须被 mount 成/sdcard。

软件兼容性

- 6.1 API 兼容性
- 6.2 HDI 兼容性
- 6.3 运行时兼容性
- 6.4 应用兼容性
- 6.5 应用包格式
- 6.6 Ark UI 框架

6.1 API 兼容性

【G-SOFTWARE-0100】必须提供 OpenHarmony 已公开 API 以及 OpenHarmony 社区源代码中所有用"@systemapi"注解的 API 的完整实现,并确保其行为与 API 文档的定义一致。

【G-SOFTWARE-0101】禁止通过更改任何方法或类签名或者通过移除类或类字段来修改 OpenHarmony 已公开 API 的定义和行为。

【G-SOFTWARE-0102】修改 API 的底层实现时,不应影响任何已公开 API 的定义和行为。

【G-SOFTWARE-0103】即使省略了 OpenHarmony 已公开 API 的某些硬件功能,也必须保持 API 存在并以合理的方式运行。

【G-SOFTWARE-0105】不应区分设备来确保 API 行为兼容性,比如采用设备白名单的方法。

【G-SOFTWARE-0106】不应在 ohos.*命名空间中的 API 添加任何已公开元素(例如类或接口,或现有类或接口的字段或方法)、或"@systemapi"注解的接口。

【G-SOFTWARE-0107】在 ohos 命名空间外扩展 API 时,不应位于归其他组织所有或提及其他组织的命名空间内。

【G-SOFTWARE-0108】禁止更改 Native API 的定义和行为。

【G-SOFTWARE-0109】禁止添加或移除 Native 库中的公共函数。

【G-SOFTWARE-0110】禁止更改"@test"注解的 API 定义和行为。

6.2 HDI 兼容性

【G-SOFTWARE-0200】必须按照"驱动开发指南"中开发文档中所述规范进行 HDI 开发。

【G-SOFTWARE-0201】如果现有驱动无法满足接口定义,应新增接口实现,不应修改现有实现。

【G-SOFTWARE-0202】如果系统服务部件中需要与某个被规定为可选部件的硬件部件交互,但设备实现不具备该组件,仍必须提供该组件 HDI 的完整类定义,并以合理的方式将该 HDI 的行为实现为空操作,不允许未实现的 HDI 返回异常。

6.3 运行时兼容性

【G-SOFTWARE-0300】C 运行时的 API 必须遵循 C99 standard、POSIX.1.2008 规范。

【G-SOFTWARE-0301】JS 运行时支持 ES5、ES6 语法规范。

□ 说明

ES6 以下几个特性明确不支持:

- 3. with 语句
- 4. eval
- 5. Function

【G-SOFTWARE-0302】JS 运行时环境必须运行在严格模式。

6.4 应用兼容性

Ability 的基本组成和运行单元是 FA/PA,Ability 的生命周期切换本质上是 FA/PA 的生命周期切换。用户对一个 FA/PA 的操作(如:在一个界面上打开一个新的界面,或者按 Back 键返回上一个界面,又或者按 Home 键回到主菜单),往往是多个 FA/PA 配合进行生命周期切换才能完成。

【G-SOFTWARE-0400】禁止修改现有 want 的显示匹配行为或语义。

【G-SOFTWARE-0401】不应更改 Ability 的生命周期或生命周期回调语义。

【G-SOFTWARE-0402】应实现一个可让用户轻松切换到前一个 FA 的快捷方式。

6.5 应用包格式

【G-SOFTWARE-0500】必须能够安装和运行由 OpenHarmony 打包工具生成的 hap 包。

【G-SOFTWARE-0501】必须支持使用 hap 包签名方案签名验证".hap"文件。

【G-SOFTWARE-0502】 扩展 .hap 或者 config.json 清单时,采用的方式不应导致相应文件无法在其他 OpenHarmony 兼容的设备正确安装和运行。

【G-SOFTWARE-0503】应该为用户提供卸载已安装应用的入口。

【G-SOFTWARE-0504】禁止在没有任何用户提示的情况下静默安装应用。

【G-SOFTWARE-0505】针对显式安装的应用,禁止在没有用户确认的情况下静默卸载应用。

6.6 Ark UI 框架

【G-SOFTWARE-0600】禁止修改 config.json 中 js 标签配置规则。

【G-SOFTWARE-0601】禁止修改 HML、CSS、JS 语法。

【G-SOFTWARE-0602】禁止修改 JS 组件,及其属性和接口定义。

【G-SOFTWARE-0603】类 web 范式限制应用访问的绝对路径应统一用"/"开头、相对路径统一用"./"或"../"开头。

【G-SOFTWARE-0604】类 web 范式限制应用私有目录的路径前缀应为 internal://app/,仅本应用可见,随应用卸载删除。

【G-SOFTWARE-0605】类 web 范式禁止应用使用../等方式访问父目录,比如 internal://app/../。

【G-SOFTWARE-0606】禁止修改支持多语言、定义资源文件和引用资源的规则。

了分布式兼容性

- 7.1 分布式硬件
- 7.2 分布式软总线
- 7.3 分布式数据管理
- 7.4 分布式任务调度
- 7.5 分布式文件
- 7.6 分布式通知

7.1 分布式硬件

【G-DISTRIBUTED-0100】禁止修改分布式设备管理点对点认证交互协议。

【STD-DISTRIBUTED-0101】禁止修改分布式相机交互协议。

【STD-DISTRIBUTED-0102】禁止修改分布式硬件信息存储规则。

【STD-DISTRIBUTED-0113】禁止修改 OpenHarmony 分布式屏幕交互协议。

7.2 分布式软总线

网络协议

面向的设备如果有网络的需求:

【G-HARDWARE-0200】必须支持 IPV4 协议栈,必须提供 Linux BSD Socket API 或者 LWIP BSD Socket API 之一。

【G-HARDWARE-0201】必须支持 DHCPv4/Client。

WLAN

如果设备支持 WLAN:

【G-HARDWARE-0202】必须支持 IEEE 802.11b/g/n。

【G-HARDWARE-0203】必须支持 STA 和 AP 模式。

【G-HARDWARE-0204】必须通过 GetHardwareProfile()中 WLAN 字段提供是否支持wlan 硬件的标识。

如果设备支持 IEEE 802.11 定义的 WLAN 省电模式:

【G-HARDWARE-0205】必须在应用使用 WLAN 服务 API 中的性能或者低延迟模式中关闭省电模式。

【G-HARDWARE-0206】必须支持多播 DNS(mDNS),并且在任何操作时间不应过滤 mDNS 数据包。

【G-HARDWARE-0207】必须支持 DHCPv4 Client/Server。

【G-HARDWARE-SR-0500】作为 AP 时, 建议支持≥2 个 STA 接入。

【G-HARDWARE-0208】必须支持 WFA WPA/WPA2personal,建议支持 WPS2.0。

【G-HARDWARE-SR-0600】建议支持≥1 个 SDIO2.0 Slave 接口。

蓝牙

如果设备支持蓝牙

【G-HARDWARE-SR-0700】推荐支持 BLE(Bluetooth Low Energy)。

【G-HARDWARE-SR-0701】推荐支持 PTA(Packet Traffic Arbitration)。

【G-HARDWARE-0211】必须通过 GetHardwareProfile()中 bluetooth 字段提供是否支持蓝牙硬件的标识。

分布式软总线

【G-DISTRIBUTED-0200】禁止修改 OpenHarmony 分布式软总线基于 CoAP 的设备发现管理协议。

【G-DISTRIBUTED-0201】禁止修改 OpenHarmony 分布式软总线基于蓝牙 BLE 的设备发现管理协议。

【G-DISTRIBUTED-0202】禁止修改 OpenHarmony 分布式软总线设备组网的管理协议,包括设备信息交换协议及心跳规则和机制。

【G-DISTRIBUTED-0203】禁止修改或终止设备保活策略和心跳规则,如 TCP 的 keepalive 周期以及 BLE 广播间隔和扫描占空比。

【G-DISTRIBUTED-0204】禁止修改组网设备类型定义,需遵照统一的设备类型设定约束。

【G-DISTRIBUTED-0205】禁止修改 OpenHarmony 分布式软总线设备间传输通道的管理协议。

□ 说明

上述规则中的管理协议指软总线发布的报文格式的定义,字段定义以及字段的含义、交互的顺序。协议在保证兼容性的基础上可受控扩展,但是禁止删除和修改已有定义。

【G-DISTRIBUTED-0206】禁止修改传输的默认协议(TCP/UDP),新增或者变更默认传输协议必须通过协商机制来实现。

【G-DISTRIBUTED-0207】禁止修改设备 UUID、NetworkId 生成规则。

【G-DISTRIBUTED-0208】必须支持蓝牙、WiFi 或以太网等软总线依赖的通信能力中的一种或者多种。

【G-DISTRIBUTED-0209】禁止修改软总线默认使用的 BLE 广播的 UUID (0xFDEE),如果发生变更就无法和使用原 0xFDEE 地址为 UUID 的蓝牙设备实现互通。

【G-DISTRIBUTED-0210】使用消息传输接口,消息大小不超过 4K 字节,超过时需要业务对消息进行分包处理,或者改为使用字节传输接口。字节传输可支持最大 4M 字节(注意:不同产品因硬件限制,此上限可能有变化)。

【G-DISTRIBUTED-0211】出于安全和兼容性考虑,不应修改安全相关逻辑 (permission.json 检查,设备认证,会话认证),不应变更软总线依赖的安全 (HiChain) 和协议部件 (CoAP, JSON)。

【G-DISTRIBUTED-0212】禁止修改 RPC 中定义的数据结构和接口,并提供对应的完整实现。

【G-DISTRIBUTED-0213】设备的 NetworkId 仅在设备在线时可用,设备下线后其 NetworkId 也随之失效。不应将 NetworkId 作为设备的长期的标识;不应修改 NetworkId 的生命周期,比如在对端设备下线后,继续对外提供本地缓存的对端设备 NetworkId,否则可能引发不可预知的问题。

【G-DISTRIBUTED-SR-0200】传输层的安全不代表业务安全的达成,建议机密数据的传输应在使用软总线接口前就完成安全加密,采用业务到业务间的端到端安全机制。

7.3 分布式数据管理

【G-DISTRIBUTED-0300】禁止修改分布式数据服务中用于设备间数据同步的协议帧、包结构以及加解密算法。

【G-DISTRIBUTED-0301】禁止删除和修改 service_meta 数据库存贮的 KvStoreMetaData 的已有字段以及 Key 的生成方法。

【G-DISTRIBUTED-0302】禁止改变分布式同步的数据的 key 和 value 的规格。

□ 说明

进行分布式同步的数据,必须满足每个 key 不超过 1KiB,每个 value 不超过 4MiB。

【G-DISTRIBUTED-0303】禁止修改系统定义的用于存储 FeatureAbility、ParticleAbility的 map 表的格式。

7.4 分布式任务调度

【G-DISTRIBUTED-0400】禁止修改跨设备之间 DMS 通信协议,包括 Transaction ID 和数据结构定义。

【G-DISTRIBUTED-0401】禁止修改系统服务 SAID 的定义。

【G-DISTRIBUTED-0402】设备必须支持 RPC。

7.5 分布式文件

【G-DISTRIBUTED-0500】禁止修改 OpenHarmony 分布式文件系统设备间传输通道的管理协议。

7.6 分布式通知

【G-DISTRIBUTED-0600】禁止修改分布式通知消息数据存储的 Key 值生成规则。

【G-DISTRIBUTED-0601】禁止修改分布式通知消息数据存储的 Value 值对应的数据结构,如数据结构版本号名称、ID 名称等。

【G-DISTRIBUTED-0602】禁止修改分布式通知框架定义的系统通知类型,如BASIC TEXT,MUTITEXT等。

性能和功耗兼容性

IO 性能

如果设备支持 Wi-Fi 2.4G:

【G-PERFORMANCE-SR-0100】强烈建议支持 2.4G 的网络 IO 性能≥10Mbps。

低功耗模式

低功耗模式可以有效的减少芯片的功耗,需要芯片提供多种低功耗的控制来动态降低 芯片的功耗:

【G-POWER-SR-0100】强烈建议提供时钟门控功能,在模块空闲的时候,可以关闭相应的时钟。

【G-POWER-SR-0101】强烈建议提供模块的工作频率动态调整功能。

【G-POWER-SR-0102】强烈建议提供外设器件低功耗管理模式。视产品形态包括但不限于 LCD、WiFi、Modem、BT、NFC、Sensor 等。

【G-POWER-SR-0103】 强烈建议提供 SensorHUB 代理 Sensor 事件。

【G-POWER-SR-0104】强烈建议支持电源低功耗管控,如器件动态上下电。

【G-POWER-SR-0105】强烈建议支持系统低功耗模式,支持后台应用低功耗模式:长时任务、短时任务、Job代理。

系统和软件升级

【G-UPDATE-0100】设备必须提供整个系统软件升级的机制。升级机制包括:在线升级,USB线缆升级或者移动存储卡升级等。

【G-UPDATE-0101】设备升级过程必须支持不擦除用户数据。

【G-UPDATE-0102】设备升级前必须对升级包进行基于公私钥对的签名校验。

【G-UPDATE-SR-0100】签名校验方式强烈建议使用 SHA-256 进行哈希计算,通过 RSA3072 进行验证。

【G-UPDATE-0103】如果设备具备连接因特网能力,必须支持在线下载升级包并完成设备的升级。

【G-UPDATE-0104】设备升级过程中必须支持掉电保护功能。

【G-UPDATE-SR-0101】推荐对全部系统分区进行 A/B 备份或者只对关键分区进行 A/B 备份。

开发工具和开发选项

开发工具

【G-TOOLS-0100】必须支持 OpenHarmony 调试命令控制台。

【G-TOOLS-0101】必须支持 OpenHarmony 调试命令集合。

【G-TOOLS-0102】必须支持 OpenHarmony 调试所需权限。

【G-TOOLS-0103】必须支持 OpenHarmony 调试节点的完整性。

【G-TOOLS-0104】必须满足 OpenHarmony 日志、trace 等格式输入输出要求。

【G-TOOLS-0105】必须满足 OpenHarmony hiprofiler 相关系统数据获取格式。

【G-TOOLS-0106】必须满足 OpenHarmony 工具链 API 与命令的兼容不发生改变。

【C-STD-TOOLS-0107】如果设备支持 ability runtime,必须支持 aa 工具命令行。

【C-STD-TOOLS-0108】如果设备支持 bundle_framework, 必须支持 bm 工具命令行。

【C-STD-TOOLS-0109】如果设备支持 ArkUI 框架,必须支持 Wukong 工具命令行。

开发选项

【G-TOOLS-0107】必须支持进入 OpenHarmony 调试模式。

【G-TOOLS-0108】必须支持 OpenHarmony 调试配置。

兼容性测试套件

【G-CERTIFICATION-0100】必须通过 OpenHarmony 要求的兼容性测试。

【G-CERTIFICATION-0101】必须使用配套 OpenHarmony 发布的最新兼容性测试套件进行测试。

🗀 说明

兼容性测试以 OpenHarmony 官网兼容性测试专区发布的最新测试要求和测试套件为准。

12 修订记录

2022/06/30: OpenHarmony3.1 产品兼容性规范首次发布。