

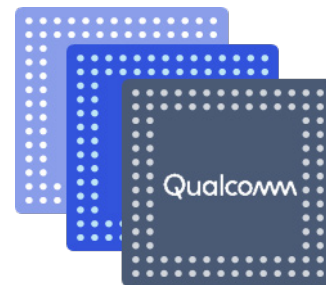
# SDM670/SDM710 Linux Android 软件架构 概述

80-PD126-1SC 版本 C

机密和专有信息 – Qualcomm Technologies, Inc.

禁止公开披露：如若发现本文档在公共服务器或网站上发布，请报告至：[DocCtrlAgent@qualcomm.com](mailto:DocCtrlAgent@qualcomm.com)。

限制分发：未经 Qualcomm 配置管理部门的明确批准，不得向 Qualcomm Technologies, Inc. 或其关联公司的员工之外的任何人分发。



# 机密和专有信息 – Qualcomm Technologies, Inc.



**机密和专有信息 – Qualcomm Technologies, Inc.**

**禁止公开披露：**如若发现本文档在公共服务器或网站上发布，请报告至：[DocCtrlAgent@qualcomm.com](mailto:DocCtrlAgent@qualcomm.com)。

**限制分发：**未经 Qualcomm 配置管理部门的明确批准，不得向 Qualcomm Technologies, Inc. 或其关联公司的员工之外的任何人分发。

未经 Qualcomm Technologies, Inc. 的明确书面许可，不得使用、复印、复制或修改其全部或部分内容，或以任何方式向其他人泄露其内容。

本文档中所提及的所有 Qualcomm 产品为 Qualcomm Technologies, Inc. 和/或其子公司的产品。

Qualcomm、Kryo、Hexagon、MSM 和 Snapdragon 是 Qualcomm Incorporated 在美国及其他国家/地区所注册的商标。Qualcomm ChipCode 是 Qualcomm Incorporated 的商标。其他产品和品牌名称可能是其各自所有者的商标或注册商标。

本技术资料可能受美国和国际出口、再出口或转让（统称“出口”）法律的约束。严禁违反美国和国际法律。

Qualcomm Technologies, Inc.  
5775 Morehouse Drive  
San Diego, CA 92121  
U.S.A.

© 2017–2018 Qualcomm Technologies, Inc. 和/或其子公司。保留所有权利。

# 修订历史记录

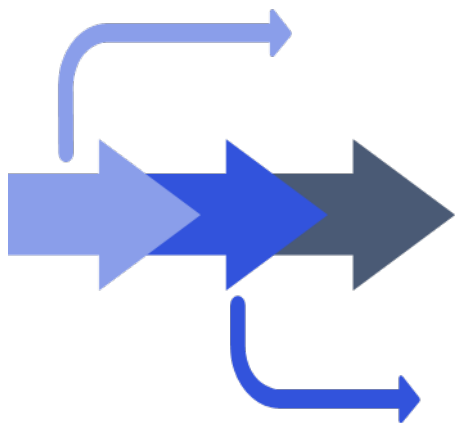
版本	日期	说明
A	2017 年 8 月	初始版本
B	2017 年 11 月	本文档进行了编辑性修改；务必通篇阅读
C	2018 年 4 月	更新了文档，与 SDM710 和 SDM670 CS 的详细信息和配置相匹配。通篇阅读

# 目录

---

- 目标
- SDM670/SDM710 应用程序处理器
- 启动序列
- 内核 4.9
- Android O
- APPS 启动加载程序
- 分区变更
- 进程间通信 (IPC)
- 电源管理
- 软件交付内容
- 调试概述
- 支持
- 参考资料
- 问题?

Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huqin.com



Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaiqin.com

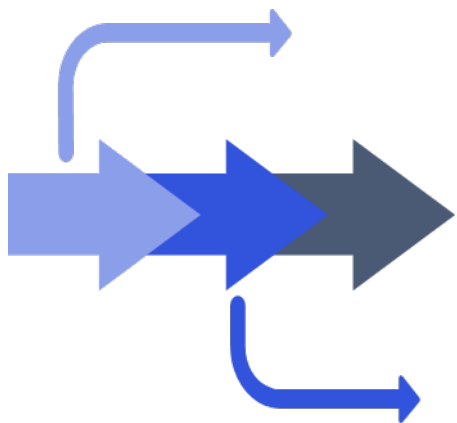
## 目标

---

# 目标

---

- OEM 管理人员和工程师可用本文档来：
  - 了解 SDM670/SDM710 芯片组 HLOS 的新功能及产品改进
  - 了解应用启动加载程序、内核和 Android BSP 的新功能
  - 了解硬件和软件迁移的详细信息，帮助规划开发工作
  - 了解支持 SDM670/SDM710 芯片组进行开发和产品化而不可或缺的新工具、工具版本、Qualcomm Technologies Inc. (QTI) 软件交付内容
  - 了解为 SDM670/SDM710 芯片组提供的新调试支持
  - 高级软件组件
  - 参考文档
- 本文档是 OEM 工程师和管理人员了解 SDM670/SDM710 芯片组内核、应用启动加载程序和 Android BSP 新增功能以及项目规划信息的切入点



## SDM670/SDM710 应用程序处理器

---

# SDM670/SDM710 应用程序处理器子系统

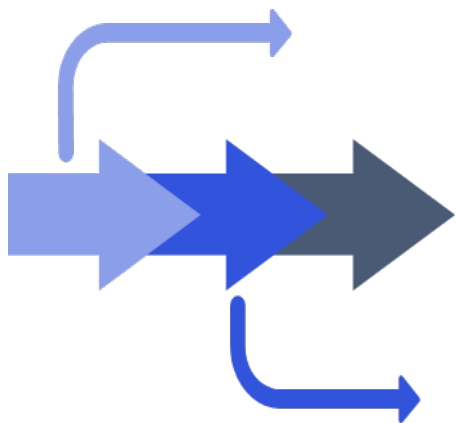
硬件参数	SDM660		SDM670		SDM710	
	Qualcomm® Kryo™ 200 CPU		Kryo 360		Kryo 360	
核心类型	Kryo 200 大核	Kryo 200 小核	Kryo 360 大核	Kryo 360 小核	Kryo 360 大核	Kryo 360 小核
架构	ARM v8	ARM v8	ARM v8	ARM v8	ARM v8	ARM v8
F <sub>max</sub>	2.2 GHz	1.843 GHz	约 2.0 GHz	约 1.7 GHz	约 2.2 GHz	约 1.7 GHz
CPU 数量	4	4	2	6	2	6
二级缓存	1 MB 二级缓存	1 MB 二级缓存	256 KB/核心	128 KB/核心	256 KB/核心	128 KB/核心
三级缓存	不适用		1 MB		1 MB	
系统缓存	不适用		512 KB		512 KB	
NEON/Crypto、VFP	有	有	有	有	有	有
注释	—	—	每个核心的专用二级缓存， 单线程操作总缓存 > 1 MB		每个核心的专用二级缓存， 单线程操作总缓存 > 1 MB	



# CPU 时钟和电压控制

- 高性能大核群集和低功耗 小核 群集连同相应二级缓存以不同频率在不同时钟源中运行：在同一群集中，所有核心和 L2 保持同步。
- 1 MB 共享三级缓存（与 APC0 共享电源，但 L3 频率可与 小核不同）。



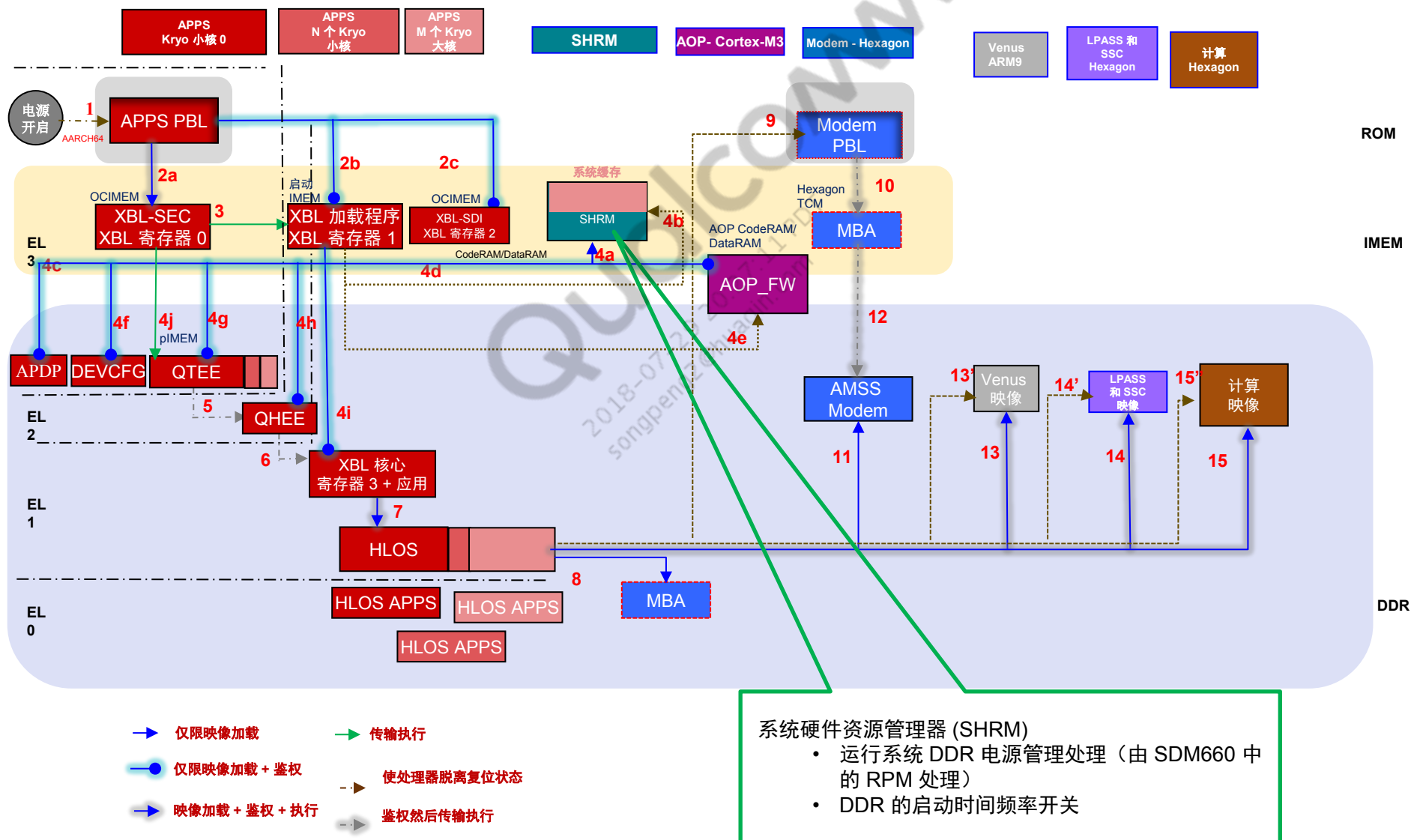


Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaqin.com

## 启动序列

---

## SDM670/SDM710 冷启动流程



## SDM670/SDM710 冷启动流程（续）

1. Kryo/Cortex A53 应用程序 CPU 已退出复位模式。
2. 在 Kryo/Cortex A53 中，APPS PBL 执行以下操作：
  - a. 将 XBL SEC（区域 0）从存储设备加载到 OCIMEM 并进行鉴权。
  - b. 将 XBL（区域 1）从存储设备加载到 L2TCM 并进行鉴权。
  - c. 将 XBL SDI（系统调试映像）(区域 2)加载到 OCIMEM 并进行鉴权，然后跳转至 XBL 区域 1。
3. XBL SEC 在 EL3 模式下运行安全配置，然后在 EL1 模式下执行 XBL。
4. XBL 按以下顺序加载其它固件：
  - a. 将 Qualcomm® 受信任执行环境 (QTEE) 映像从启动设备加载到 pIMEM 并进行验证。
  - b. 将 QHEE 映像从启动设备加载到 DDR 并进行鉴权。
  - c. 将 XBL 核心（区域 3）和 ABL 映像从启动设备加载到 DDR 并进行鉴权。
  - d. 将 RPM 固件映像从启动设备加载到 RPM 代码 RAM 并进行鉴权。
  - e. XBL SEC 将执行传输到 QTEE。
5. QTEE 使 RPM 脱离复位模式，以执行 RPM 固件。
6. QTEE 建立安全环境，然后执行 QHEE 映像。
7. QHEE 执行 XBL 核心（或 XBL 区域 3）和 XBL 核心安装，然后运行 UEFI 应用 (abl.elf)。

## SDM670/SDM710 冷启动流程（续）

---

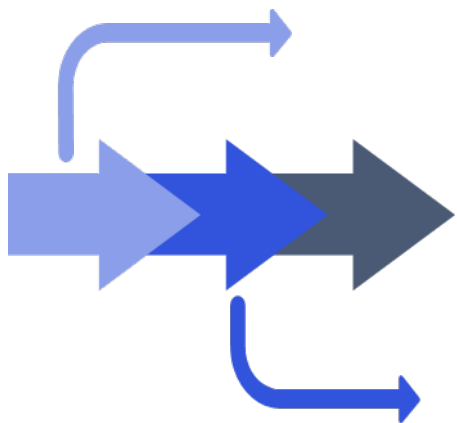
8. Linux 加载程序应用程序（ABL 的组成部分）通过验证启动加载 HLOS 内核并进行验证。
9. HLOS 内核通过外设映像加载程序 (PIL) 将 MBA 加载到 DDR。
10. HLOS 内核使 Qualcomm® Hexagon™ DSP Modem 脱离复位模式。
11. Modem PBL 将 MBA 从 DDR 复制到 Modem TCM、对 MBA 进行鉴权，然后跳转到 MBA 映像。
12. HLOS 通过 PIL 将 AMSS Modem 映像加载到 DDR。
13. MBA 对 Modem 映像进行鉴权，然后跳转至 Modem。
14. HLOS 通过 PIL 将 Venus 和 LPASS 映像加载到 DDR。
15. HLOS 使 Venus 和 LPASS 脱离复位模式，然后通过安全 SMC 调用启动执行。

# SDM670/SDM710 冷启动流程（续）

组件	基于处理器	加载源	ZI/RW 分配位置	执行位置	功能
应用程序主启动加载程序 (APPS PBL)	Kryo 小核 0/性能群集 A53 Core0	—	L2TCM、RPM CodeRAM、VMEM	ROM	<ul style="list-style-type: none"><li>▪ 建立 APPS 安全 RoT、支持 APPS 安全启动和 XBL 加载器、启动设备解密以及紧急下载模式。</li><li>▪ 在 L2 TCM、片上内存 (OCIMEM) 和 RPM CodeRAM 中加载和并检验 XBL1 ELF 段。</li></ul>
可扩展启动加载程序 (XBL)	Kryo 小核 0/性能群集 A53 Core0	eMMC/UFS	L2 TCM、VMEM、LPDDR4X、OCIMEM	L2 TCM、VMEM	<ul style="list-style-type: none"><li>▪ 启动内存子系统（总线、DDR、时钟和 CDT）、加载/鉴权 QTEE、QHEE、RPM_FW、XBL 核心映像、通过 USB 和 Sahara 进行内存转储、支持看门狗调试保持、支持 RAM 转储到 SD、支持 USB 驱动程序、USB 充电、热检查、支持 PMIC 驱动程序和 DDR 定型。</li><li>▪ 以单个 ELF 区段的形式包含 XBL SEC，以锁定 xPU，并在执行其余 XBL 之前将控制从安全 EL3 切换为非安全 EL2 或 EL1。</li></ul>
QTEE	Kryo 小核/大核/Cortex A53 核心	eMMC/UFS	OCIMEM、LPDDR4X	OCIMEM、LPDDR4 (pIMEM)	<ul style="list-style-type: none"><li>▪ 等同于 TrustZone BSP</li><li>▪ 建立安全运行系统执行环境</li><li>▪ 配置 xPU</li><li>▪ 支持熔丝驱动程序</li></ul>
Qualcomm Hypervisor 执行环境 (QHEE)	Kryo 小核 0/Cortex A53 核心	eMMC/UFS	LPDDR4X	LPDDR4X	<ul style="list-style-type: none"><li>▪ 配置 SMMU</li><li>▪ 提供虚拟化支持</li></ul>
RPM_FW	Cortex M3	eMMC/UFS	RPM DataRAM/MessageRAM	RPM CodeRAM	发挥资源电源管理器的作用。

# SDM670/SDM710 冷启动流程（续）

组件	基于处理器	加载源	ZI/RW 分配位置	执行位置	功能
XBL 核心（UEFI 或 LK）	Kryo 小核 0/性能群集 A53 Core0	eMMC/UFS	LPDDR4X	LPDDR4X	HLOS 特有的功能丰富的启动加载程序（UEFI 功能）
HLOS	Kryo 小核/大核 /Cortex A53 核心	eMMC/UFS	LPDDR4X	LPDDR4X	HLOS 内核和应用程序
Modem 主启动加载程序 (PBL)	Modem Hexagon	–	Hexagon TCM	ROM	<ul style="list-style-type: none"><li>▪ 建立 Modem 安全 RoT</li><li>▪ 将 MBA 从 DDR 加载到 Hexagon TCM</li><li>▪ 鉴权/解密 MBA</li></ul>
Modem 启动鉴权程序 (MBA)	Modem Hexagon	eMMC/UFS	Hexagon TCM	Hexagon TCM	启用对 Modem 固件的安全鉴权和解密
Modem AMSS	Modem Hexagon	eMMC/UFS	LPDDR4X	LPDDR4X	Modem 固件



Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaqin.com

## 内核 4.9

---



# Linux 内核 4.9

- SDM670/SDM710 芯片组使用 Linux 内核 4.9：作为 Android O Treble 项目的组成部分，所有电路板和厂商驱动程序均以可加载内核模块 (LKM) 的形式加载。
- SDM660 使用内核 4.4
- 与内核 4.4 相比，内核 4.9 的主要变更包括：

更改产生影响的设备驱动程序	说明	OEM 产生影响 (是/否)
dma_attrs 为无符号长整型，而非结构体	<ul style="list-style-type: none"><li>▪ 将驱动程序移植到内核 4.9 时，务必使用 dma_attrs</li><li>▪ QTI 驱动程序已更新，可使用新数据结构类型</li></ul>	有
工作队列锁定检测器	QTI 已在内核 4.4 中向后移植	否
CPU 热插拔基础架构	<ul style="list-style-type: none"><li>▪ 在内核 4.9 中重新编写，但向后兼容</li><li>▪ 内核 4.10+ 中取消了向后兼容，未来的内核版本必须遵循新接口或 API – 目前无影响</li></ul>	否
SLUB 空闲列表随机化	减弱攻击者对内核堆状态的控制，但 QTI 目前不会启用此功能，否则将破坏解析程序	否
硬件固化的 usercopy (copy_to/from_user)	防止内核写入错误或恶意写入内核驱动程序	否

# Linux 内核 4.9（续）

调试工具	说明	OEM 产生影响 (是/否)
dtx_diff tool	调试工具以源格式和二进制格式比较设备树	否

其它变更	说明	OEM 产生影响 (是/否)
GPIO 芯片是真实器件	除非写入 GPIO 控制器，否则不产生影响	否
kallsyms 为相对地址	在 64 位 架构中，kallsyms 地址表缩小一半， 因为内核符号之间的偏移以 32 位形式表示。平均节省数百千字节的永久 .rodata	否
GCC 插件支持	<ul style="list-style-type: none"><li>▪ 取决于 GCC 版本</li><li>▪ 开发人员向编译程序添加新功能时无需修改编译程序本身</li></ul>	否
ro_after_init 部分	模块布局中具有 更多页面对齐部分	否

# 可加载内核模块

- Treble (Android O) 要求厂商、ODM 分别实现内核 4.9
  - 所有内核必须在内部支持启动和安装分区
  - 所有电路板外设驱动程序必须以内核模块形式加载
  - 所有 SoC 内核必须支持 LKM
    - CONFIG\_MODULES=y
    - CONFIG\_MODULE\_UNLOAD=y
    - CONFIG\_MODVERSIONS=y
    - CONFIG\_MODULE\_SRCVERSION\_ALL=y
  - 对于要求支持验证启动的设备，内核模块必须位于验证分区中
  - 内核模块必须位于只读分区中
  - 不强制进行内核模块签名

## 可加载内核模块（续）

- LKM 位置：

- 模块必须从经过验证的只读分区载入
- 模块不得位于 /system 路径，从而保持独立框架空中下载升级 (FOTA) 目标
- 来自 SoC 厂商的模块必须位于 /vendor/lib/modules 位置。这些模块不得依赖于 ODM 模块
- 来自 ODM 的模块必须位于 /odm/lib/modules
- 恢复需要使用的模块（来自 SoC 厂商和 ODM）必须位于 /lib/modules 的恢复 rootfs 中
- 建议通过连续加载或卸载测试运行模块

## 可加载内核模块（续）

### ■ LKM 添加方式：

1. 要在厂商映像中创建 LKM，则将以下代码添加到 BoardConfig.mk（以 LKM 的名称代替 module\_name）：

```
BOARD_VENDOR_KERNEL_MODULES := \  
$(KERNEL_MODULES_OUT)/module_name.ko
```

2. 如果在恢复模式下需要使用模块，则添加以下代码：

```
BOARD_RECOVERY_KERNEL_MODULES := \  
$(KERNEL_MODULES_OUT)/module_a.ko
```

3. 要通过 init 进程加载模块，将以下代码添加到 init.target.rc（对于恢复模式，则添加到 /bootable/recovery/etc/init.rc）：

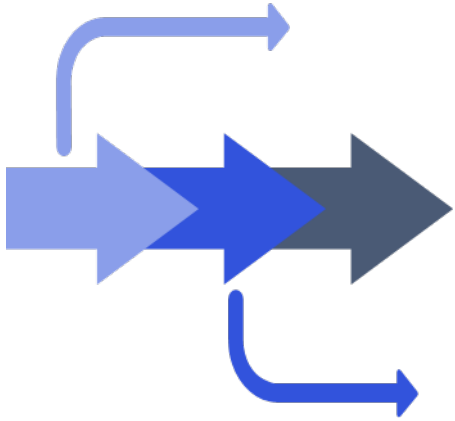
```
on early-init  
exec -- /system/bin/modprobe -a -d /vendor/lib/modules module_name
```

### ■ 驱动程序转换为 LKM 的示例：

- snd-soc-wsa881x.ko – 扬声器放大驱动程序 wsa881x

# 设备树叠加 (DTO)

- 分为用于 SoC 特定数据的基础设备树 (DT) 文件和用于电路板特定数据的叠加 DT 文件。
  - 基础 DT 位于启动分区，叠加 DT 位于新 DTBO 分区。
  - 启动加载程序从相应分区中查找 DT blob。
  - 启动加载程序使用设备树编译器 (DTC) 对包含叠加 blob 的主 DT blob 进行修补。合并 DT 传递到内核。
  - DT 源文件位置：/kernel/arch/arm/boot/dts/。
  - 叠加 DT 源文件将“-overlay.dts”作为文件名后缀。
- 例如
  - sdm845.dts -- SoC 的基本 DT 源文件。DT 源文件包含的所有 DTSI 文件不得包含任何电路板特定绑定。
  - sdm845-mtp-overlay.dts -- MTP 板的叠加 DT 源文件。此文件包含的所有 DTSI 文件不得包含任何 SoC 特定绑定。
  - OEM 板特有的数据应添加到叠加 DT 源文件。不包含具有 SoC 特定绑定的 DTSI。



Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaqin.com

## Android O

---

# Android O

---

- Treble
  - Android 框架更改拆分系统和厂商映像
  - 针对 LKM 的内核 4.9 变更
  - 针对 DTO 的内核 4.9 变更
- 启动加载程序：Verified Boot 2.0 和 DTO 的变更

Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaqin.com



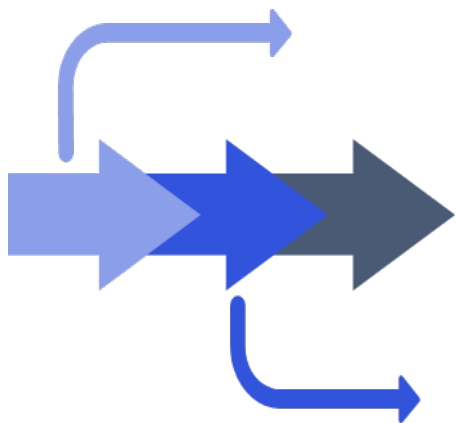
# Android O（续）

## ■ Treble Android 框架变更

- C/C++ 头文件中HAL 声明用HAL 接口定义语言 (HIDL) 重新。
- HAL 接口变为进程间 binder通信。框架在客户端侧运行，厂商实施方案在服务器侧的独立进程（HAL 守护进程）中运行。
- VNDK 包含厂商接口使用的通用基础库。厂商 HAL 不得与 VNDK 以外的 ASOP 框架存在任何依赖性。
- 对于修改 QTI HAL 或实现自有 HAL 的 OEM：
  - HAL 进行了重新设计，将实际功能从其接口中分离（输入/输出和类型）。
  - 确定 Android 框架的所有 HAL 链接。
  - 确保所有链接仅对照 VNDK 库，不参考其它任何库。
  - 如果在 QTI HAL 中进行更改或使用 QTI 库，应与 QTI 工程团队合作协调变更事宜。
  - QTI HAL 版本编号已与 QTI 协调。

## ■ 系统和厂商映像分离

- 之前的 AOSP 更新与系统映像中的厂商模块捆绑在一起
- Treble 中的映像分为两个映像：
  - 包含核心框架的系统映像
  - 厂商映像



Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huqin.com

## APPS 启动加载程序

---

# UEFI APPS 启动加载程序变更

## ■ DTO

- Android O 通过 DTO 方案将设备树分为主 SoC 特定组件和电路板特定组件。
  - 主 DT: 仅SoC部分和默认配置, 由 SoC 厂商提供。
  - 叠加 DT: 设备特定配置, 由 ODM/OEM 提供。
- 统一可扩展固件接口 (UEFI) 负责:
  - 从相应分区 (DTBO 和启动) 中查找 SoC 特定 (主 DT blob) 和电路板特定 (叠加 DT blob) 设备树 blob。
  - 使用 DTC 对包含叠加 blob 的主 DT blob 进行修补。
  - 创建一个传递到内核的组合 DT。

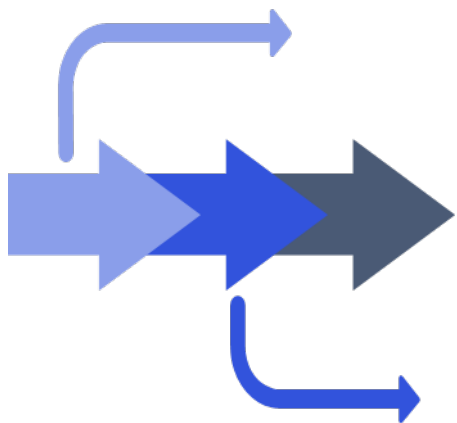
# Verified Boot 2.0

---

- 密钥管理：Android 现在允许对每个分区单独签名。
  - 每个分区使用由分区所有者生成的自有私钥签名。
  - 各分区可能有不同的所有者。例如，Google 可能对系统分区签名，OEM 对厂商分区签名。
- 各分区的公钥提供给 OEM 并保存在密钥存储分区。密钥存储分区由 OEM 使用 其私钥签名。
- 在启动过程中：
  - UEFI 使用 OEM 公钥验证密钥存储分区以及获取所有分区的公钥。
  - UEFI 使用 Google 提供的 libavb 验证所有分区的元数据。
  - UEFI 验证所有分区的签名。
  - UEFI 从 KERNEL\_CMDLINE 描述符中读取 dm-verity 参数并将其传递给内核。

## Verified Boot 2.0（续）

- 防回滚
  - Android 现在分配与 OS 版本相关的非负数 (rollback\_index)。
  - 每个分区均有专属回滚索引。
    - 每个分区的 A 和 B 插槽共享回滚索引。
    - 例如，boot\_a/boot\_b、system\_a/system\_b、vendor\_a/vendor\_b 和 odm\_a/odm\_b 是共享四个回滚索引的八条插槽。
  - 回滚索引以单调递增（即从不减小）。如果修改需要回滚索引递增（即回滚索引与新补丁、安全补丁等级或 API 等级更改一同发布时），则回滚索引递增。
  - 防回滚通过将允许的回滚索引定义为 min\_rollback\_index 进行启用。
- 防回滚的 UEFI 支持：
  - 跟踪每个分区的 min\_rollback 索引值。
  - 如果分区的回滚索引小于其 min\_rollback\_index，设备启动。



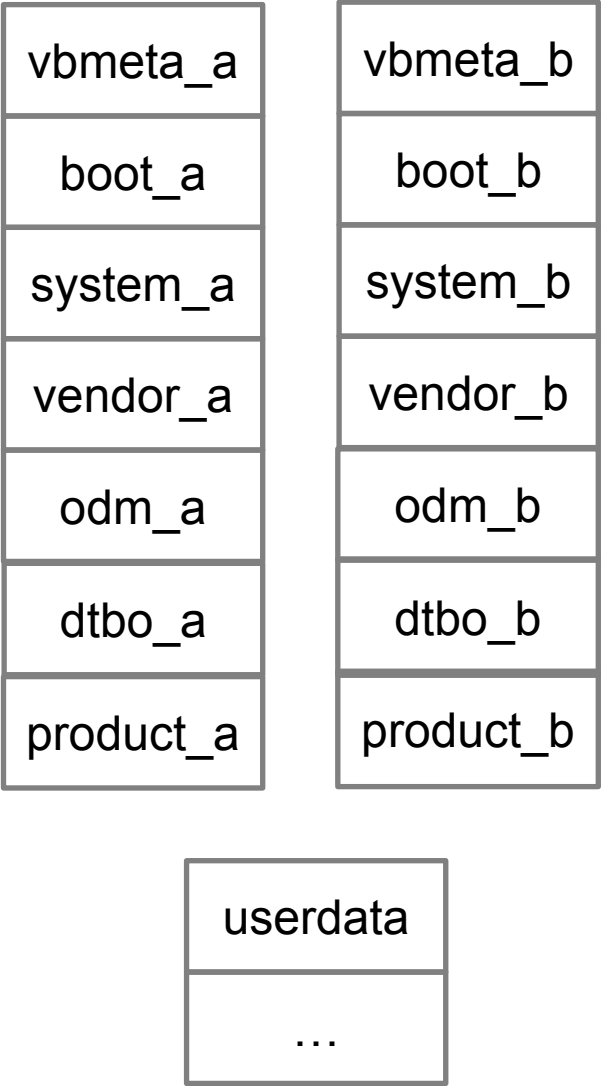
Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaqin.com

## 分区变更

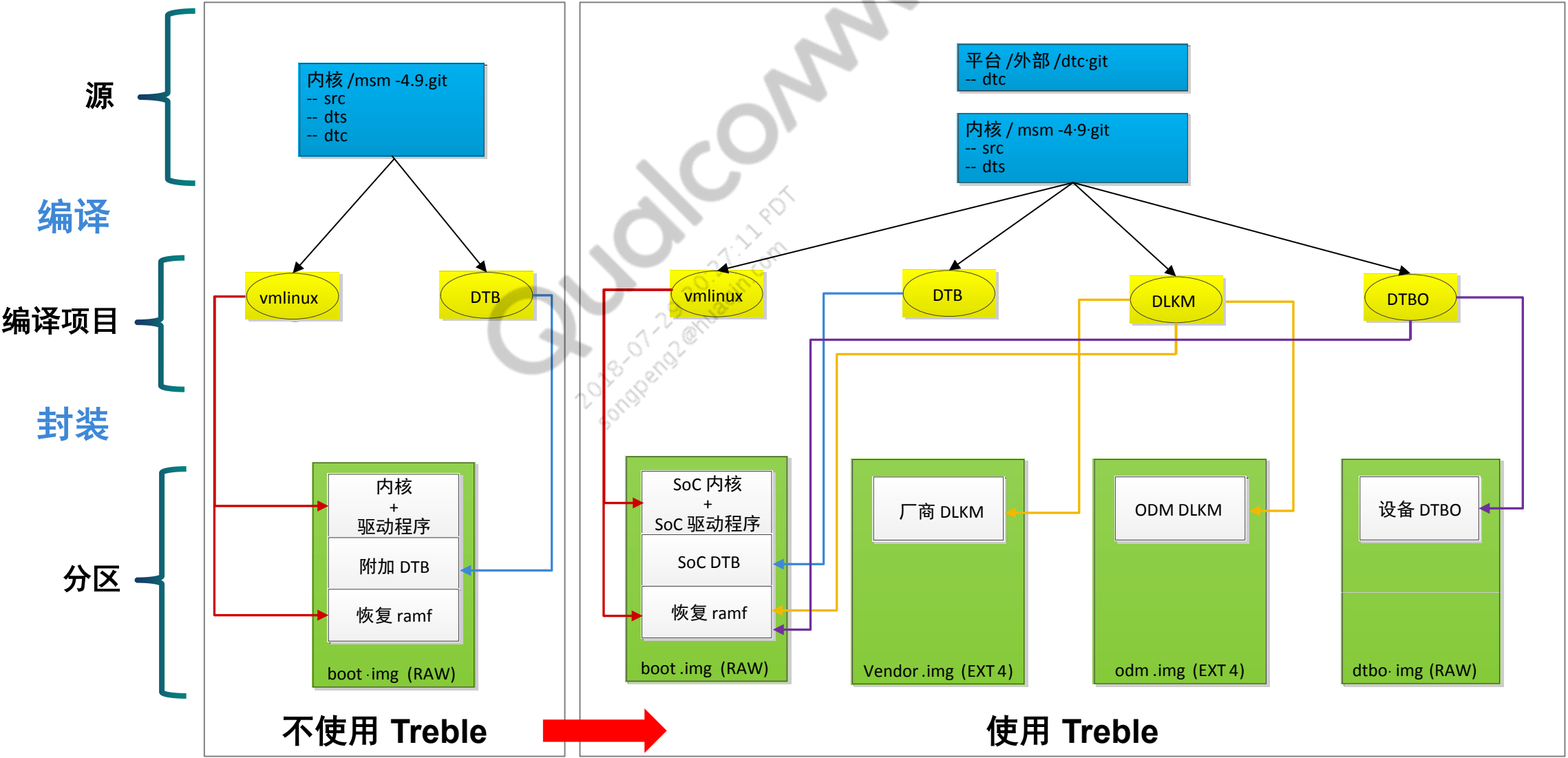
---

# 分区变更

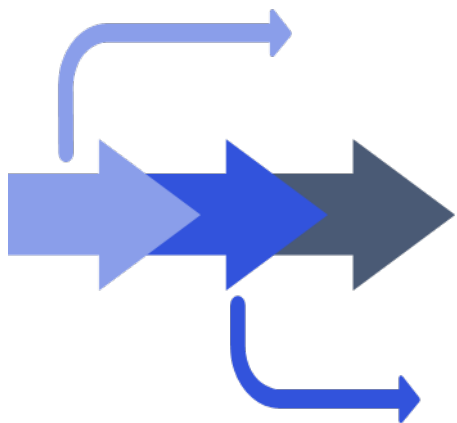
特性	添加分区
A/B 分区（默认）	<ul style="list-style-type: none"><li>添加 A 和 B 分区</li><li>system_a.img、system_b.img</li><li>vendor_a.img、vendor_b.img</li></ul>
Treble 框架	分离 system.img 和 vendor.img
Verified Boot 2.0 （回滚保护）	添加 vbmeta_a 和 vbmeta_b 分区
DTO、LKM	添加 dtbo_a.img、dtbo_b.img (DT) odm_a.img、odm_b.img (DLKM)



# 编译和封装变更







Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaqin.com

## 进程间通信 (IPC)

---

# IPC 机制

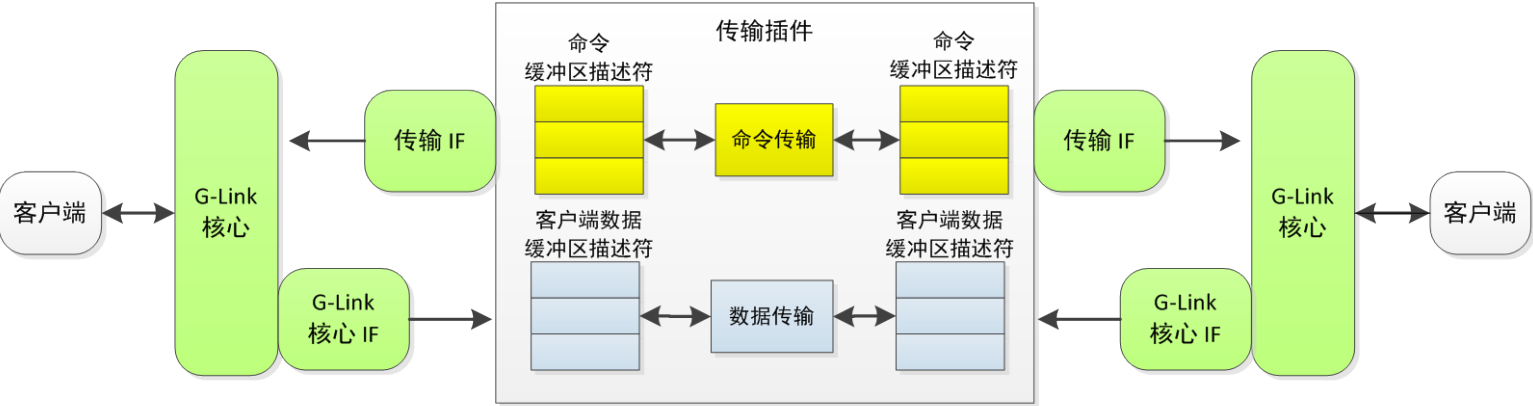
---

- 共享内存 (SMEM): 所有处理器共享的物理内存
- Qualcomm MSM™ 芯片组接口: 采用 IDL 定义的消息传递框架
- G-Link 通用链路: 点对点链路层传输

Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaqin.com

# 通用链路 (G-Link)

- 点对点链路层传输，代替 SMD：客户端包括 Linux 内核、Modem、RPM、LPASS 和 SLPI
- 功能
  - 一路或多路物理传输的多路复用逻辑通道
  - 支持的物理传输
    - 共享存储器
    - 基于复本（如 UART）
    - 基于 DMA
  - 添加新传输句柄的插件架构
  - 性能可扩展
  - 占用空间可扩展，适用于 RPM 等资源受限的系统
  - 发生错误或子系统重启时出现故障缓冲区所有权返回给所有者
  - 带内信号传输
  - QoS 流量优先级确定



虚拟命令和数据队列将始终存在，  
但可能在单次物理传输中多路复用。

## 通用链路 (G-Link) (续)

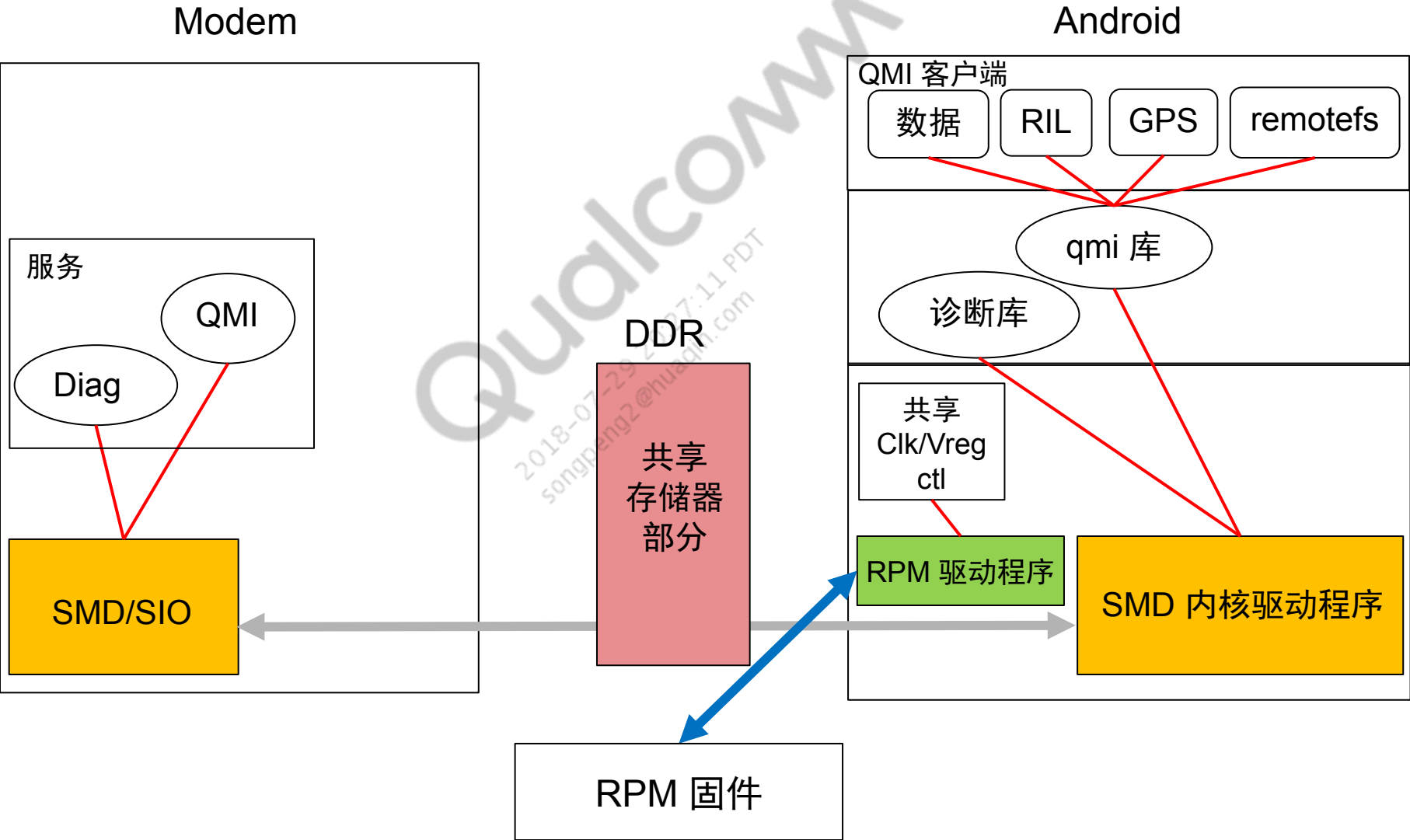
---

- 主要概念
  - 异步 API
  - 反向压力流量控制机制；避免在客户端未准备好接收时发送数据包
  - 基于传输特定 MTU 的透明数据包分片
  - 循环数据包调度
- 更多详细信息，参见 *G-Link (Generic Link) Overview* (80-P2598-1)

# QMI 背景知识

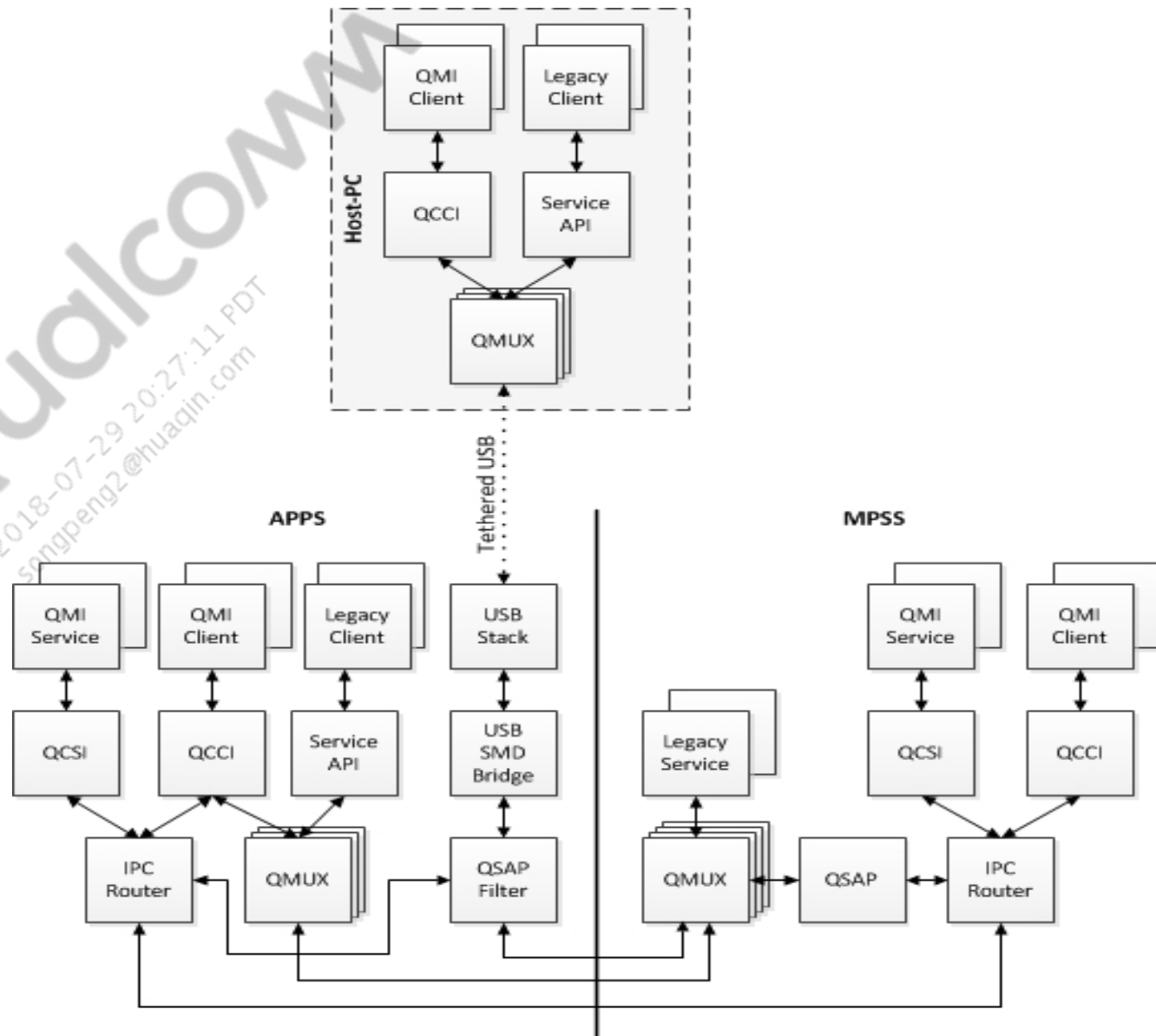
- 采用接口描述语言 (IDL) 定义的消息传递框架
  - Android 无线接口层 (RIL) 通过其传递 Modem 消息
    - 执行方
      - 客户端 – 发出请求
      - 服务 – 处理请求、发送响应和指示
    - 消息类型
      - 请求 – 设置/获取参数值并配置指示
      - 响应 – 已请求操作的结果及关联数据
      - 指示 – 通知客户端状态更改
    - IDL
      - 通过 IDL 定义的 QMI 消息
      - IDL 编译器为服务和客户端生成 C 头文件和源文件
  - 更多详细信息, 参见 *QMI Framework Overview* (80-P4684-6)

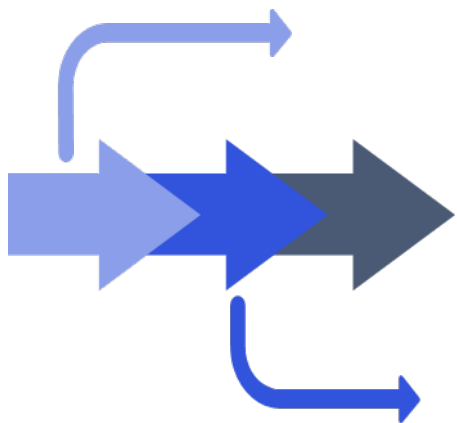
# QMI 和服务



# QMI 高级架构

- QMI 多路复用器 (QMUX) (传统) : 传统点对点路由
- QMI 通用客户端接口 (QCCI): 连接注册的 QMI 服务、发送请求以及接收响应或指示
- QMI 通用客户端接口 (QCSI): 用于实现服务
- IPC 路由器 (QMUX v2): 点对点消息路由、名称服务发现和流量控制
- QMI 服务访问代理 (QSAP): 允许传统 QMUC 点对点客户端通过 IPC 路由器进行通信





Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaqin.com

## 电源管理

---



# CPU 低功耗模式

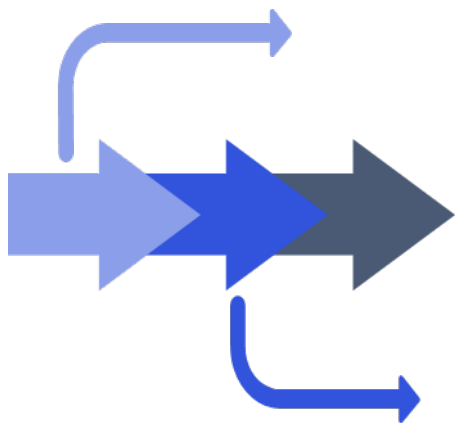
低功耗模式	说明
C1	核心时钟选通
C2	<ul style="list-style-type: none"><li>核心内存保持</li><li>L1/L2 保持并开启</li></ul>
C3	<ul style="list-style-type: none"><li>核心电源深度睡眠</li><li>L1 和 L2 关闭</li><li>核心关闭</li><li>探测关闭</li></ul>
C4	C3 状态 + PLL 关闭（群集）+ Vdd 关闭（仅限大核）

# L3 低功耗模式

低功耗模式	说明
D1	L3 时钟选通
D2	<ul style="list-style-type: none"><li>▪ L3 和 GNOC 时钟选通</li><li>▪ 探测开启</li></ul>
D4	<ul style="list-style-type: none"><li>▪ L3 时钟/探测关闭</li><li>▪ L3 缓存清空并由硬件关闭</li></ul>

# Linux Android 支持的系统低功耗模式

模式索引	说明
1	应用 L3 缓存深度睡眠 (D4) + 系统休眠（RSC 握手）+ Cx 关闭 + Mx 保持
2	模式 1 + 系统缓存关闭 + AOSS 休眠



Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huqin.com

## 软件交付内容

---

# 软件组件

专有软件交付内容：<https://chipcode.qti.qualcomm.com>

专有源码



开源软件交付内容：<https://Codeaurora.org>

开源



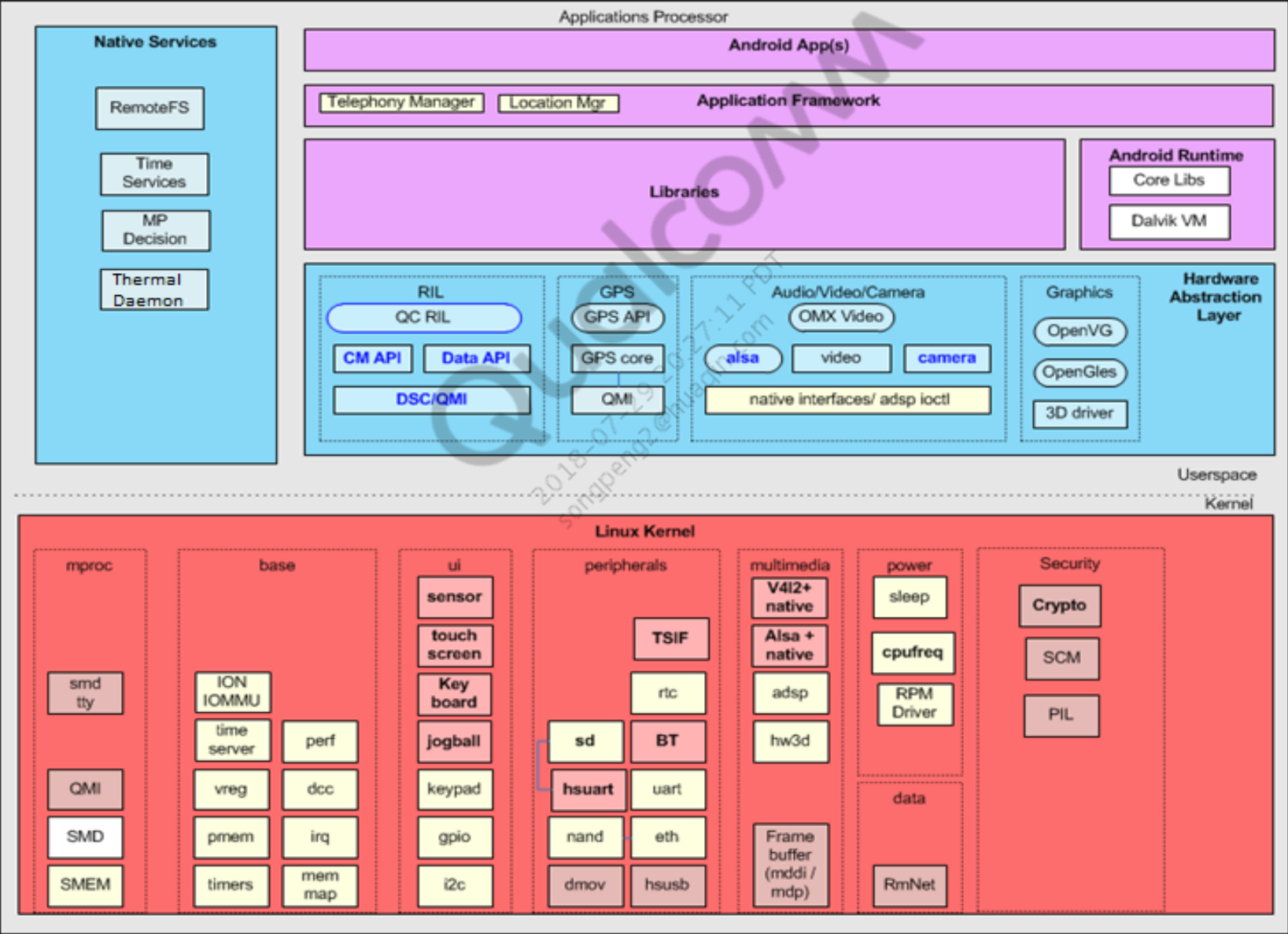


# 软件工具和补丁

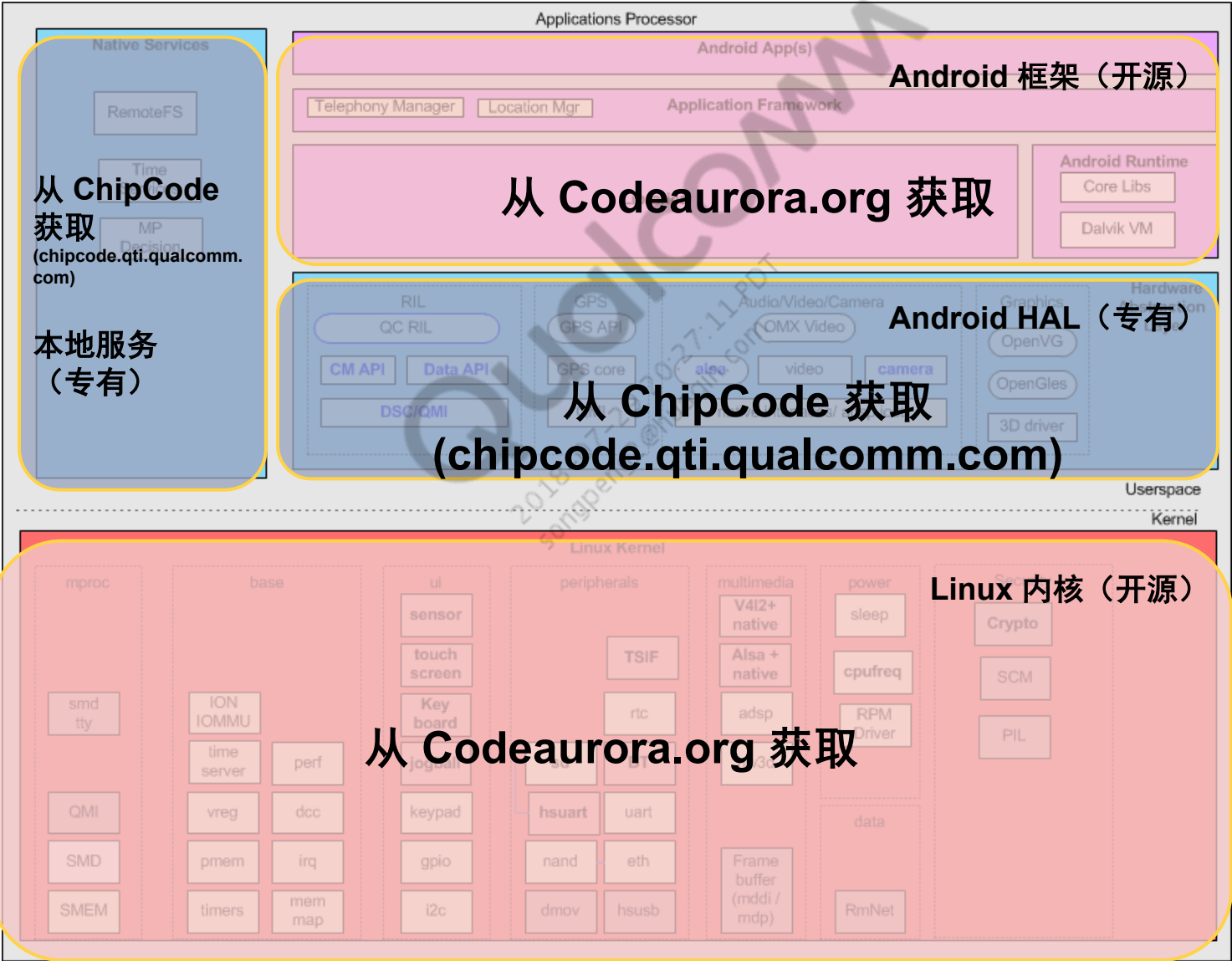
---

- 文档和工具门户以及新芯片组项目起点：  
<https://createpoint.qti.qualcomm.com>
- 关于使用 Git 存储库存储软件代码的 Qualcomm ChipCode™ 门户，访问：  
<https://chipcode.qti.qualcomm.com>
- 补丁
  - 专有软件补丁分发：  
<https://chipcode.qti.qualcomm.com>
  - 开源软件补丁分发：  
<https://www.codeaurora.org/patches/quic/la>

# 软件交付内容 – 应用程序处理器



# 软件交付内容 – 应用程序处理器（续）

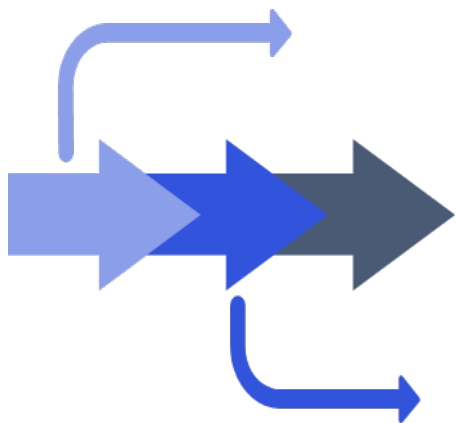




# 编译环境

---

- Linux 内核和用户空间
  - Ubuntu 14.04
  - Java 8 (OpenJDK)
  - gcc 4.9.0 工具链
  - Qualcomm® Snapdragon™ 处理器 64 LLVM
- 启动和安全环境（TrustZone (TZ) 和安全监视器）
  - ARM 工具 (LLVM)
- 调试 – 来自Lauterbach 的 64 位 TRACE32



Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaqin.com

## 调试概述

---

# 调试

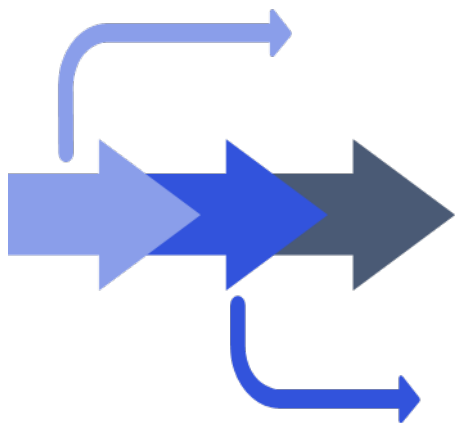
- 嵌入式 USB 调试 (EUD)
  - 允许 OEM 使用无适配器的低成本调试器
  - 允许 OEM 继续使用无适配器的 Lauterbach 调试器
- EUD 硬件支持下列三种主要工作模式：
  - 旁路模式 – EUD 有效关闭或禁用
  - 调试模式 – EUD (usb-hub) 存在且处于激活状态
  - 安全模式 – EUD 允许受限访问
- 出于安全目的，EUD 由熔丝引导

EUD 熔丝	熔丝说明	
fuse_eud_dis (具有覆盖功能)	1	EUD 处于激活状态
	0	EUD 保持非激活状态，且处于旁路模式
fuse_ignr_csr	1	向 csr_eud_en 写入 1 可避免 EUD 进入调试（忽略）
	0	插入 HSUSB 电缆后，向 csr_eud_en 写入 1 导致 EUD 进入调试模式

## 调试（续）

---

- APB 扫描转储机制适用于当前支持 JTAG 扫描转储机制的所有核心。例如 GPU、WCSS、Hexagon 处理器和 BIMC
- AOSS/RPMh 调试支持
  - 无论 CX 处于何种状态，调试器均可与 DFD 基础架构进行通信
  - 采集位于 CX 之外的模块（即 AOSS 中的模块）中的跟踪和分析数据，与 CX 电源状态无关
  - RPM 硬件固化调试，例如硬件事件、CTI、AO 处理器调试和 PCB 监视器
  - APB 扫描转储到 RPMh HM
- 崩溃调试流程过程中的应用程序启动处理器 APB 扫描转储可提取或保存通用寄存器值



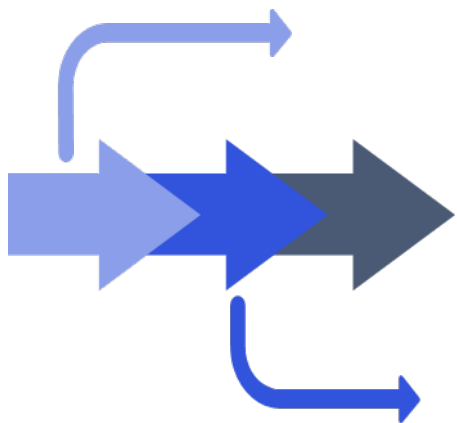
Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huaqin.com

支持

文档	
标题	文档号
Qualcomm Technologies, Inc.	
SDM670/SDM710 Linux Android Software User Manual	SP80-PD126-4
SDM670/SDM710 Boot and CoreBSP Architecture Overview	80-PD126-2
SDM670/SDM710 RPM Hardening Overview and Debug	80-PD126-21
SDM670/SDM710 Linux Peripherals (UART, SPI, I2C) Overview	80-PD126-5
SDM670/SDM710 Power Overview	80-PD126-6
SDM670/SDM710 Security Overview	80-PD126-3
SDM670/SDM710 APSS Software Migration Overview	80-PD126-37
Android O Migration Guide	80-PC301-1
GLINK (Generic Link) Overview	80-P2598-1

# 缩略词

缩写	
缩略词或术语	定义
AOSS	实时响应子系统 (Always on subsystem)
APB	高级外设总线 (Advanced peripheral bus)
BW	广播窗口 (Broadcast window)
CCI	缓存一致性互连 (Cache coherency interconnect)
DFD	调试设计 (Design for debug)
EUD	嵌入式 USB 调试 (Embedded USB debug)
FOTA	固件空中下载 (Firmware over the air)
GUID	全局唯一标识符 (Globally unique identifier)
LKM	可加载内核模块 (Loadable kernel module)
NoC	片上网络 (Network on chip)
QMUX	QMI 多路复用器 (QMI multiplexor)
RPM	资源和电源管理器 (Resource power manager)
RPMh	资源电源管理器硬件固化(Resource power manager hardening)
SMD	共享内存驱动程序 (Shared memory driver)
SMEM	共享存储器 (Shared memory)
SMSM	共享内存状态机 (Shared memory state machine)
SMMU	系统内存管理单元 (System memory management unit)
UEFI	统一可扩展固件接口 (Unified Extensible Firmware Interface)



Qualcomm  
2018-07-29 20:27:11 PDT  
songpeng2@huawei.com

问题？

<https://createpoint.qti.qualcomm.com>

---