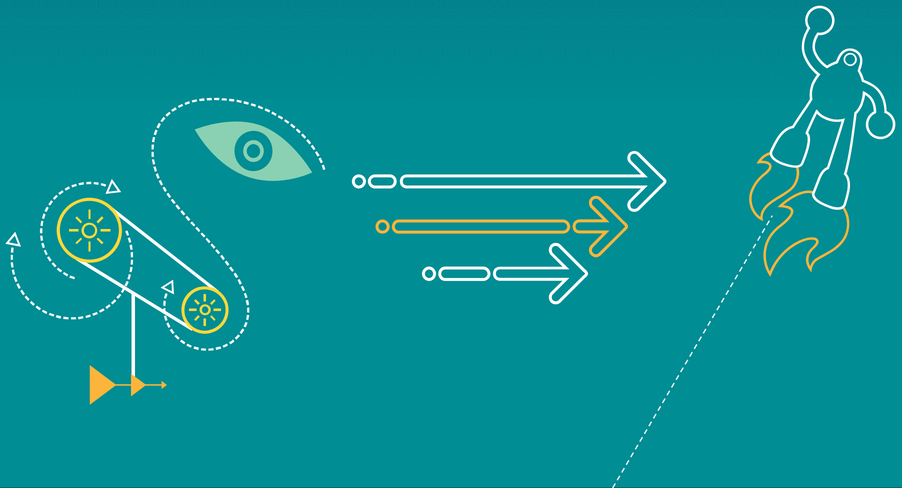
Security TrustZone QSEE Overview

Qualcomm Technologies, Inc.



80-NL239-5 F

Confidential and Proprietary – Qualcomm Technologies, Inc.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies, Inc. or its affiliated companies without the express approval of Qualcomm Configuration Management.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to: DocCtrlAgent@qualcomm.com.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. All Qualcomm Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer (“export”) laws. Diversion contrary to U.S. and international law is strictly prohibited.

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

© 2013-2015 Qualcomm Technologies, Inc. and/or its affiliated companies. All rights reserved.

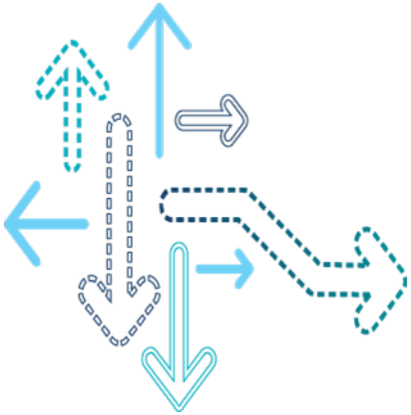
Revision History

Revision	Date	Description
A	December 2013	Initial release
B	April 2014	Added slide 6; updated slides 5 and 27
C	November 2014	Updated slides 4 to 7, 12, and 13
D	April 2015	Updated slide 6
E	April 2015	Added slides 7, 9-10, 22-25, and 33-34; updated slides 6 and 35
F	July 2015	Updated slides 5 to 7

Contents

- MSM8909/MSM8916/MSM8936/MSM8939/MSM8952/MSM8956/ MSM8976 TrustZone
- Software Customization
- Debugging
- FAQs
- References
- Questions?

MSM8909/MSM8916/MSM8936/MSM8939 /MSM8952/MSM8956/MSM8976 TrustZone



Deltas Between MSM8909/MSM8916/MSM8936/MSM8939 /MSM8952/MSM8956/MSM8976

	MSM8916	MSM8936	MSM8939/MSM8952/MSM8956/MSM8976	MSM8909
Warm boot counters	MSM8916 is a quad-core chipset. During bootup, core zero gets cold-booted and the other cores get warm-booted. The warm boot counters maintained by ARM TrustZone get updated.	Similar to MSM8916	MSM8939 is an octa-core chipset. During bootup, core zero of the performance clusters gets cold-booted and the other cores get warm-booted. The cores in the power cluster also go through warm boot. TrustZone maintains the warm boot counters for all eight cores.	Similar to MSM8916
Cache Coherent Interface (**CCI)**	There is no CCI in MSM8916 as it is a quad-core chipset.	Similar to MSM8916	CCI is present because the chipset is an octa-core. Until now, CCI configuration is done by TrustZone at bootup.	Similar to MSM8916
Hypervisor	Applicable	Applicable	Applicable	Not applicable

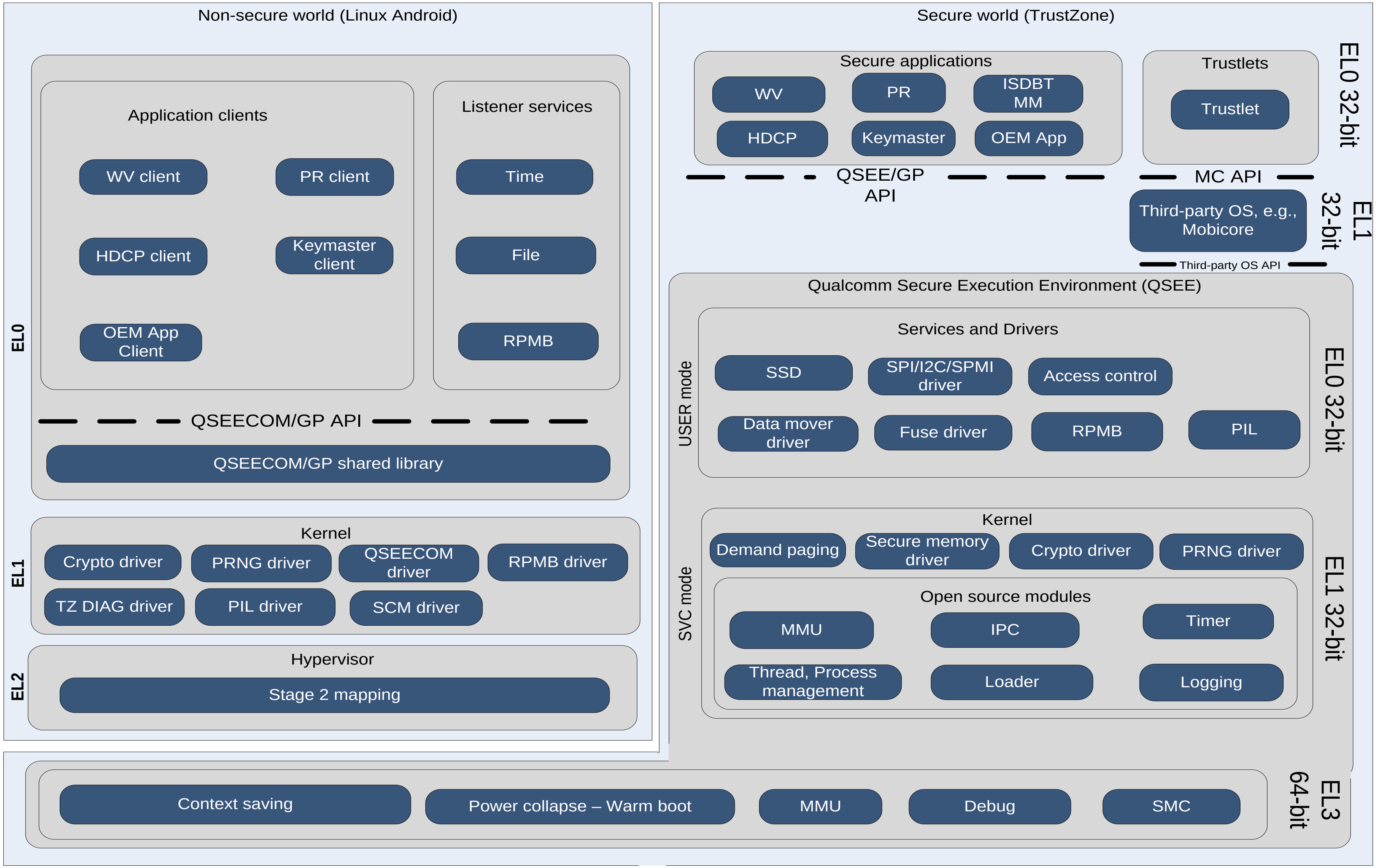
New Feature List in MSM8952/MSM8956/MSM8976

- Inline Crypto Engine (ICE) for FDE use cases
- Qualcomm Malware Protection (QMP) ver. 1.0 – Malware protection
- Integrated Services Digital Broadcasting-Terrestrial (ISDB-T) – Broadcast security feature
- SafeSwitch – Remote SIM lock feature
- Unified image encryption feature (only in MSM8956/MSM8976)
- Debug policy enabled – ramdump collection in secure boot (only in MSM8956/MSM8976)

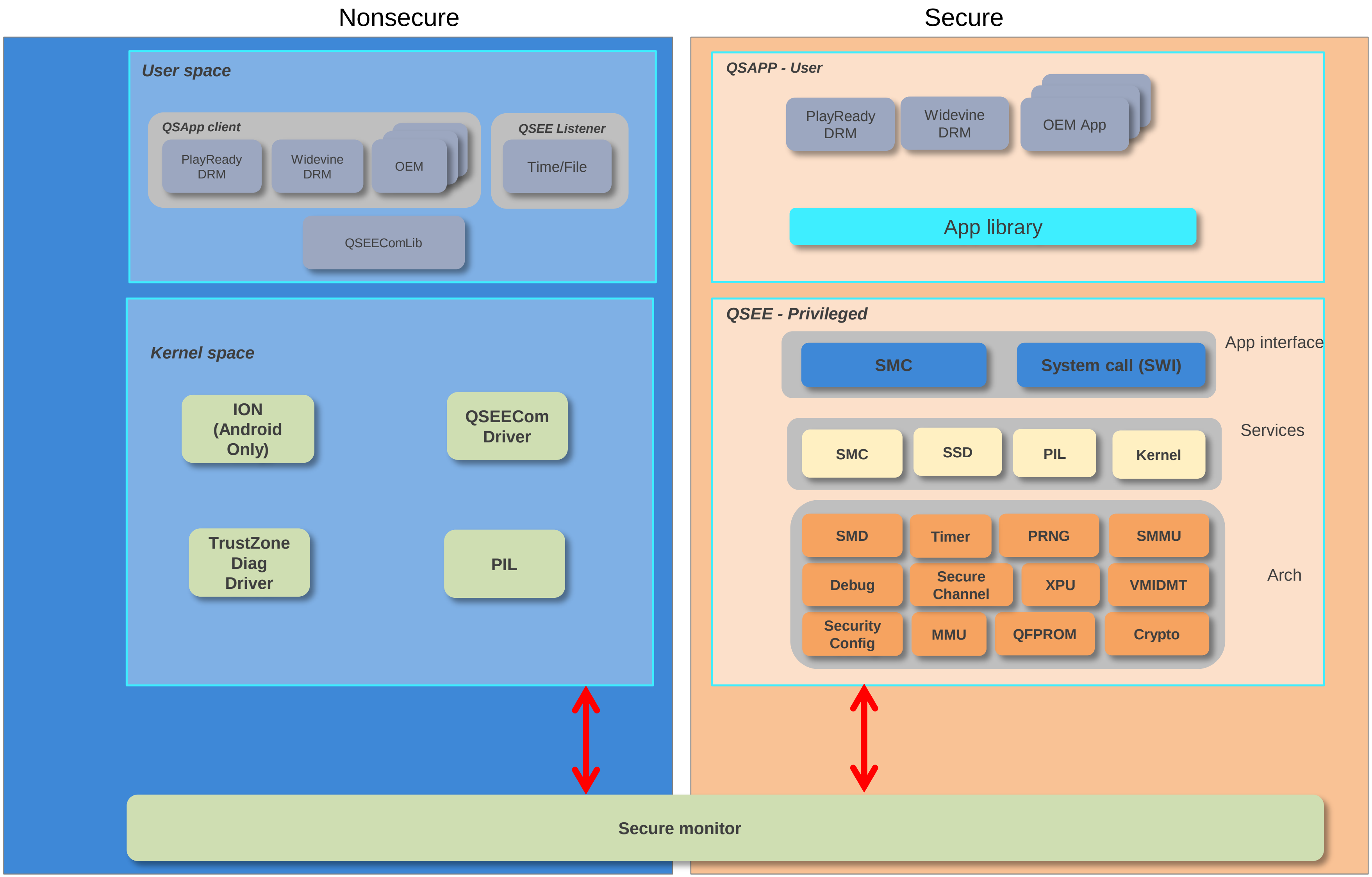
Qualcomm Secure Execution Environment (QSEE)

- TrustZone-enabled Cortex-A53 processors; MSM8909 has Cortex-A7 processor.
  - Security configuration registers, such as Security Configuration Register (SCR)
  - MMU page tables for secure and nonsecure
  - Secure or nonsecure interrupts
  - New core mode – Exception level 3 (Secure Monitor) mode; MSM8909 has only the Secure Monitor mode
- Qualcomm Access Control (QuAC)
  - Virtual Master ID Matching Table (VMIDMT)
  - xPU3
  - System MMU (SMMU)
  - QuAC Reinitialization Block (QRIB)
- TrustZone software
  - Configuring the security of Cortex-A53 or Cortex-A7 and QuAC

QSEE 3.0 Security Framework Block Diagram

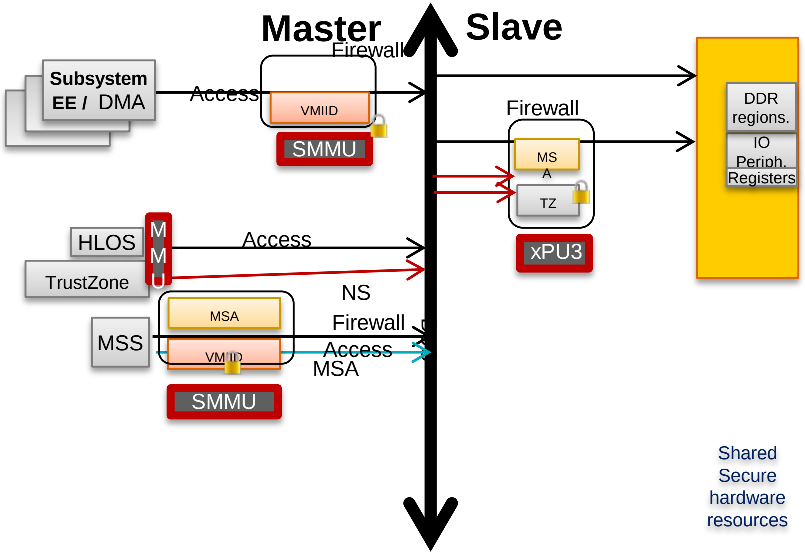


TrustZone Software Architecture for MSM8909



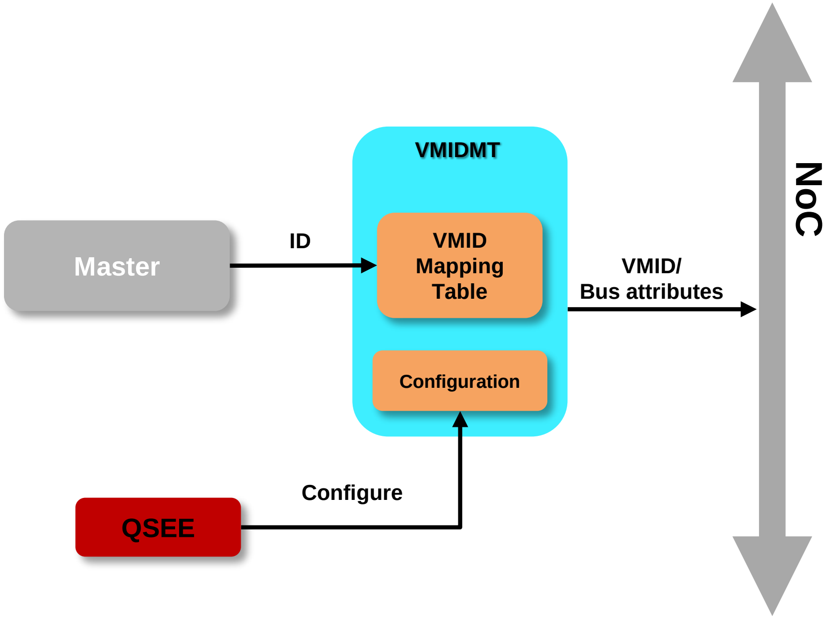
QuAC

- At the master side, the SMMU unit checks the access permissions for any memory access. Only if the access passes SMMU, it goes to the slave side.
- As a second-level check, External Protection Units (xPUs) present at the slave side control the access using VMIDs generated from the master.
- Three security domains
  - TrustZone
  - Modem
  - VMID-based
- Domain control hardware components
  - VMIDMT
  - xPUs
- Only TrustZone can configure the QuAC
- There is an exception to the MSA-based partitions. After TrustZone marks a partition as modem-only access, those partitions are accessed only by the modem. Additionally, all permission changes, and so on, are done by the modem.



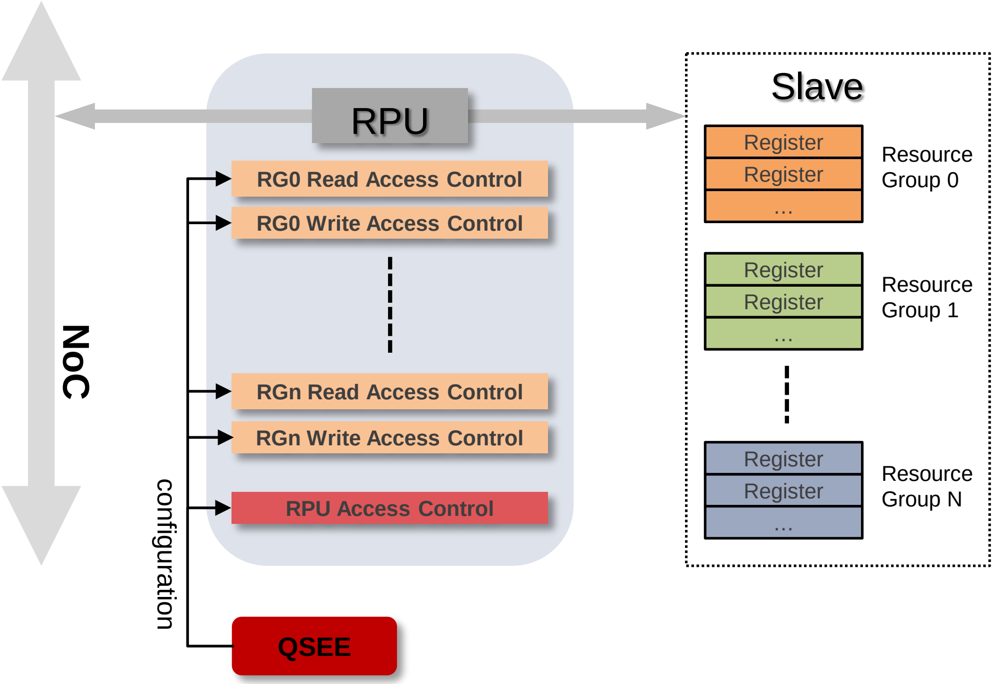
QuAC – VMIDMT

- Access control at the master side
- Generates domain control signals
  - xPROTNS for the TrustZone domain
  - xMssSelfAuth for the MSS domain
  - VMID for the VMID-based domain
- Up to 32 VMIDs supported in the system
- Configured by TrustZone



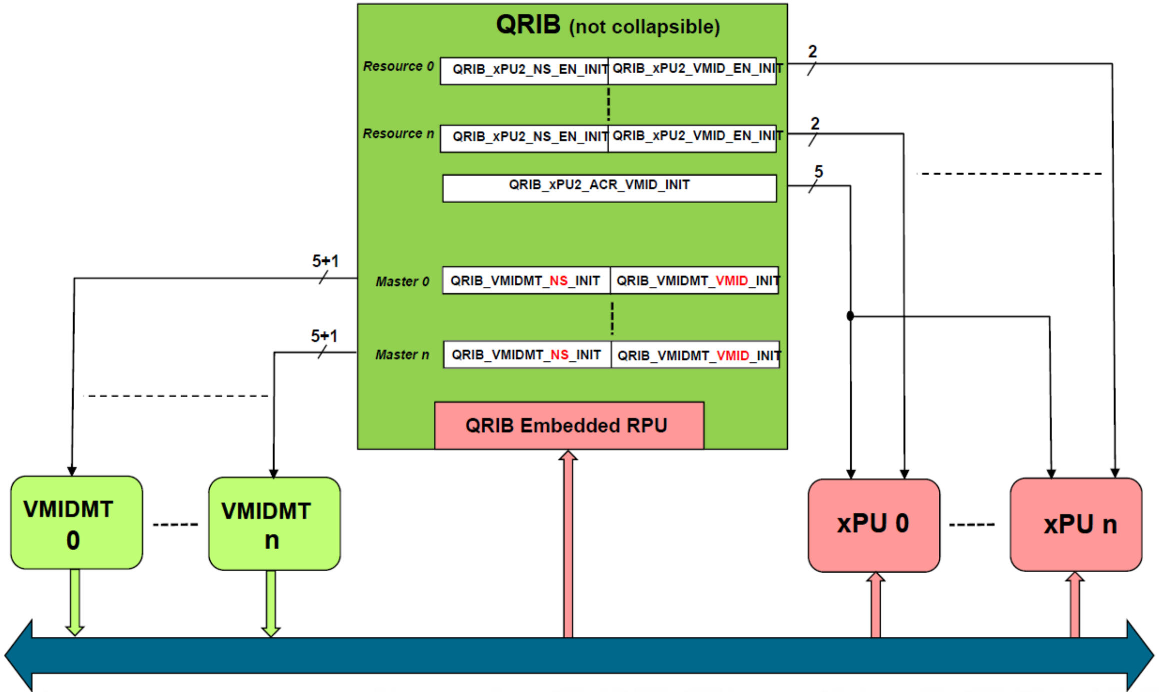
QuAC – xPU

- Combinations of multiple PUs
  - Memory Protection Unit (MPU)
  - Register Protection Unit (RPU)
  - Address Protection Unit (APU)
- Access control at the slave side
- Conditionally grants access to a master, group of masters, or to a set of resource groups
- Configured by TrustZone



QuAC – QRIB

- Some QuACs are power-collapsible
- New hardware block that maintains critical and minimal restoration information and supplies the values to the xPU component during wake-up from power collapse
- Noncollapsible QuAC components do not require QRIB automatic reinitialization



TrustZone Software

- Loaded and authenticated by Secondary Boot Loader (SBL1)
- Configures the security of Cortex-A53 cores and QuAC. In MSM8909, it configures the security of Cortex-A7 cores.
- Runs from DDR
- Entered by an SMC instruction or secure interrupts
- Components in the secure side
  - Monitor
    - Monitor software
    - FIQ handlers
    - Dump L1/L2 caches
  - QSEE
    - In Privileged mode
    - Monitor software, kernel, HAL, device drivers, PIL, SMC handling, syscall handling, and so on.
- Qualcomm Secure Applications (QSAPP)
    - In User mode
    - Secure applications, that is, Microsoft PlayReady and Widevine
    - Interface to QSEE
    - Secure File System (SFS)

TrustZone Software (cont.)

- Hypervisor/Qualcomm Hypervisor Execution Environment (QHEE) image; MSM8909 does not have Hypervisor.
  - Used to configure SMMU
  - Configures the second-level page table of SMMU, where it is used for multioperational system. MSM8909 has only the first level of SMMU configured by QSEE.
  - For any unauthorized access, where SMMU is not mapped, the error handling is trapped into Hypervisor to handle the error

QSEE Software

- Components in the nonsecure side
    - QSEECom client
        - HLOS module requests secure services to QSAPP in the secure side
    - QSEECom listener
        - Serves the requests for nonsecure services originating from QSAPP
    - QSEECom library
        - Running in User mode, provides the API interface to QSEECom clients and listeners
    - TZDiag driver
        - Allows HLOS to retrieve TrustZone diagnostic data from the secure side
    - QSEECom driver
        - Running in Kernel mode, maintains the communication channel between the secure world and nonsecure world in the nonsecure side
    - Secure Channel Manager (SCM) driver
        - SCM in Kernel mode implements the SCM interface in the nonsecure side

Use Cases and Services

- Boot
  - Cold boot
    - Loaded and authenticated by SBL1
    - Sets up stack pointers for different secure modes, that is, SVC, Undefined mode, Abort mode, and so on
    - Changes VBAR to a TrustZone vector table, which is used for the next warm boot
    - Configures security of the cores and QuAC
    - Secure state is established on cold boot
  - Warm boot
    - CPU starts in Secure mode; hence, warm boot always starts from TrustZone
    - TrustZone ensures that CPU and QuAC are known good states and then returns to HLOS
  - Power collapse
    - Power collapse terminates in TrustZone
    - Before power collapse
      - Notifies QSEE clients and QSAPPS of the power collapse
      - Flushes all TrustZone cache regions

Use Cases and Services (cont.)

- PILAuth
  - Authenticates subsystem images and brings subsystems out of reset
  - Authenticates secure applications
- Resource protection
  - Static configuration of QuAC blocks and dynamic MPU configurations
  - MSA partitions
    - During static xPU configuration, TrustZone configures three EBI partitions for Modem Self-Authentication (MSA)
    - Only MSA can access those partitions after configuration
  - Secure channels
    - Keeps the communications between subsystems secret from HLOS
      - TrustZone and MSS
      - TrustZone and LPASS
    - TrustZone generates a key on every cold boot that is used to encrypt or decrypt messages

Use Cases and Services (cont.)

- SYSCALL
  - SYSCALL is the SMC instruction-based function interface for HLOS to request the secure services of TrustZone
    - Enables or disables secure watchdog
    - Fuse read or write
    - Sets the address of memory dump buffer
    - Dynamic MPU protection
- Secure debug
  - Secure watchdog
    - Only secure watchdog can reset the system while another nonsecure watchdog bite is routed to TrustZone as secure debug FIQs
    - Only TrustZone can configure and pet the secure watchdog
  - Fatal error handling
    - Nonsecure watchdog bite
    - SGI15
    - RPM WDT bite
    - RPM fatal error
    - AHB slave timeout
    - Network-on-Chip (NoC) error

Use Cases and Services (cont.)

- Content protection; not supported in MSM 8909
  - Secure video playback
    - Decryption of secure contents
- Other secure services
  - Cache dumps (L1/L2) via SYSCALL; not supported in MSM 8909
  - BSP drivers running in TrustZone
    - Crypto
    - QFPROM
    - PRNG
      - PRNG is configured only by TrustZone
      - Generated random numbers are retrieved by any entity in the system via SYSCALL

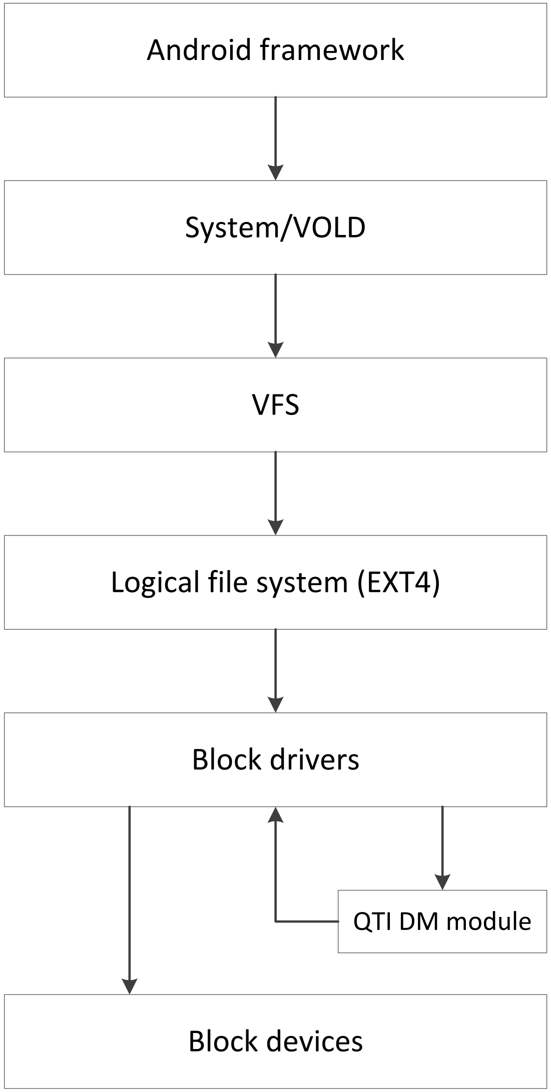
SFS Overview

- There are two independent SFSs, one on TrustZone and another on the modem
- After devices have been secured, the two SFS systems use different hardware crypto engines for file data encryption and decryption, so that they are secure from each other.
- The SFS Anti-rollback protection feature, first introduced on APQ8084 is supported on MSM8994 and future chipsets
- Anti-rollback protection is implemented through eMMC storage's RPMB hardware support and an RPMB driver in TrustZone
- During development time, OEMs must provision their devices with a test RPMB key using the sampleapp, TrustZone app
- For commercial devices, the production RPMB key is provisioned automatically by TrustZone when the secure boot and JTAG disable efuses are blown
- For SFS security, a device's RPMB key can only be provisioned once; hence, a device provisioned with a test key cannot be provisioned again with a production key
- OEMs must plan their development devices carefully due to the above restriction

Hardware-Based Full Disk Encryption Overview

- Android provides a software-based mechanism to protect user data through disk encryption
  - Currently, it supports encryption of a user data partition and any nonremovable SD card but does not provide encryption facility for any other partition
- Android disk encryption is based on the dm-crypt Linux kernel module
- Uses the mapped device feature of the kernel and works at the block layer
- QTI developed a hardware-based design that uses a new device mapper-based module that encrypts and decrypts data on a comparatively larger packet. This design provides a significant boost to encryption throughput.
- The hardware-based solution provides the following benefits:
  - Improves performance
  - Reduces power consumption through hardware crypto
  - Enhances security since crypto key is not stored in RAM, which can potentially be dumped by hackers

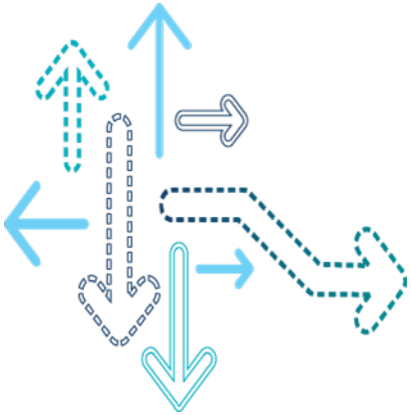
Hardware-Based Full Disk Encryption Overview (cont.)



Hardware-Based Keymaster/Keystore Overview

- Hardware-based keymaster uses TrustZone application APIs to ensure that the key data stored is not accessible by HLOS
    - Currently, only the RSA key type is supported.
- The keyblob generated is encrypted by a key accessible to TrustZone only. It is stored in the file system on the HLOS end
- HLOS components include implementation of the keymaster API defined in keymaster.h:
    - generate_keypair
    - import_keypair
    - get_keypair_public
    - delete_keypair
    - delete_all
    - sign_data
    - verify_data
- Each of these APIs call into TrustZone for TrustZone to process the request
- The TrustZone component includes a keymaster TrustZone application. This application implements the above APIs, and is generated as split binary images in the TrustZone image build.

Software Customization



Logging

- TrustZone logging control
  - Selects an output target for TrustZone logging, TrustZone ring buffer, or JTAG
  - Allows TrustZone to determine whether to dump xPU syndrome registers
  - Allows or disallows TrustZone logging

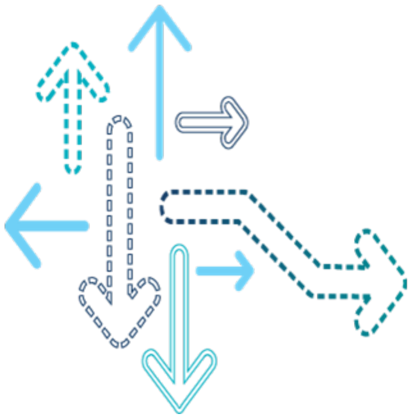
OEM SYSCALL

- OEMs can implement their own secure services
  - `TZBSP_SYSCALL_CREATE_CMD_ID` // create command ID
  - `TZBSP_DEFINE_SYSCALL` // define SYSCALL
  - Implementing the function defined by `TZBSP_DEFINE_SYSCALL`
- References
  - trustzone_images/core/securemsm/trustzone/qsee/include/tzbsp_syscall_pub.h
  - trustzone_images/core/securemsm/trustzone/qsee/arch/msm8x16/src/tzbsp_syscall_def.c

OEM Key for SFS

- OEMs can use their own unique key for SFS
  - `void qsee_oem_set_kdf_derive_key(void **key, size_t *key_len)`

Debugging



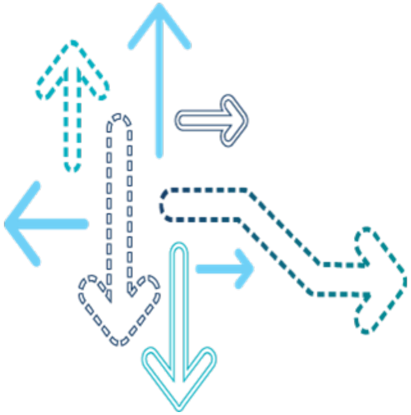
Debugging with Trace32

- To determine whether an applications processor is in Secure mode, Nonsecure mode, or Monitor mode, open a T32 register window, and look at the lower left corner for sec, nsec, or mon.
- Use Z to access secure memory
  - For example – d.dump AZ: 0xFE81000 (Z: TrustZone, A: Physical)
- Good locations for breakpoints in TrustZone
  - tzbsp_syscall – All SYSCALLs go through this function
  - tzbsp_mon_fiq_handler – Handles FIQs
  - tzbsp_mon_irq_handler – Handles IRQ exit when in the secure side
  - tzbsp_smc_handler – Handles SMC calls from both the secure and nonsecure sides

Debugging with Logging

- TrustZone diagnostic area
  - Counters (bookkeeping)
    - Warm boot entry or exit
    - Power collapse termination entry or exit
  - Logging buffer
    - Circular buffer (ring buffer)
    - Contains TrustZone logging strings including the debug information on an xPU violation and ABT timeout
    - TrustZone logging is disabled when secure boot is enabled

FAQs



FAQs

Q. How to compile/clean the TrustZone image in Linux environment?

A. Run the following command to:

- Build images:

```
./build.sh CHIPSET=<CHIPSET> tz sampleapp tzbsp_no_xpu playready
widevine securitytest keymaster commonlib
```

- Clean the build:

```
./build.sh CHIPSET=<CHIPSET> tz sampleapp tzbsp_no_xpu playready
widevine securitytest keymaster commonlib –c
```

Q. How to compile/clean the TrustZone image in Windows environment?

A. Run the following command to:

- Build images:

```
build.cmd CHIPSET=msm8909 tz sampleapp tzbsp_no_xpu playready
widevine securitytest keymaster commonlib
```

- Clean the build:

```
build.cmd CHIPSET=msm8909 tz sampleapp tzbsp_no_xpu playready
widevine securitytest keymaster commonlib –c
```

References

Acronyms	
Term	**Definition**
CCI	Cache Coherent Interface
ICE	Inline Crypto Engine
ISDB-T	Integrated Services Digital Broadcasting-Terrestrial
QHEE	Qualcomm Hypervisor Execution Environment
QMP	Qualcomm Malware Protection
QRIB	QuAC Reinitialization Block
QSAPP	Qualcomm Secure Applications
QSEE	Qualcomm Secure Execution Environment
QuAC	Qualcomm Access Control
SCM	Secure Channel Manager
SCR	Security Configuration Register
SFS	Secure File System
VMIDMT	Virtual Master ID Matching Table
xPU	External Protection Units

Questions?

https://createpoint.qti.qualcomm.com