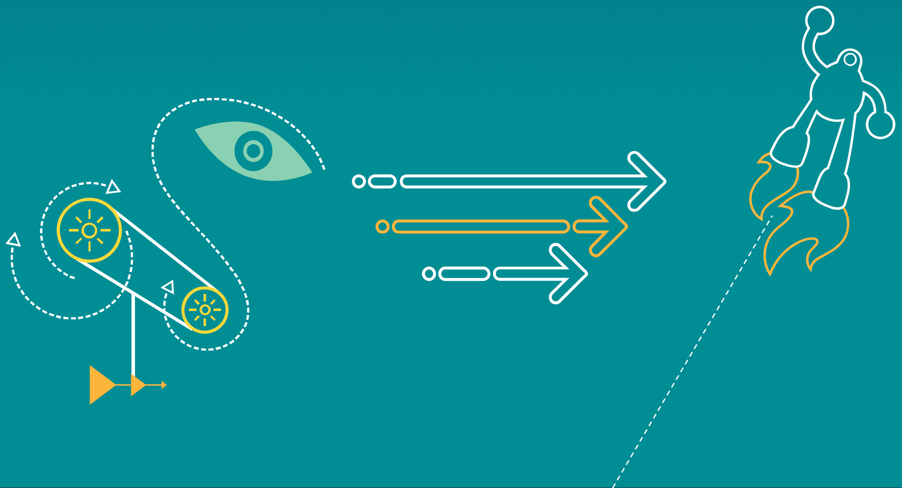
Cross-/Flexi-Mapping Support Overview

Qualcomm Technologies, Inc.

80-NA157-256 C



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

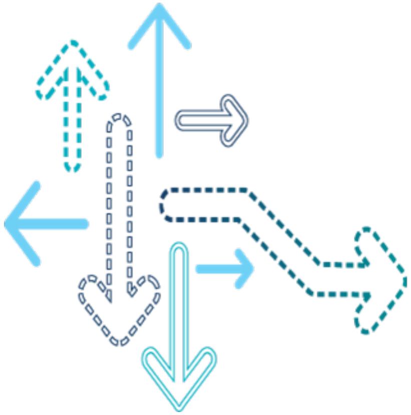
Revision History

Revision	Date	Description
A	July 2014	Initial release
B	October 2015	Added Slides 24 and 27
C	November 2015	Updated slide 25

Contents

- Feature Overview
- Interface Changes
- Cross-Mapping During Bootup
- Cross-Mapping – User-Initiated Network Mode Change
- Additional Information
- Additional Features
- Cross-Mapping – Android Marshmallow
- References
- Questions?

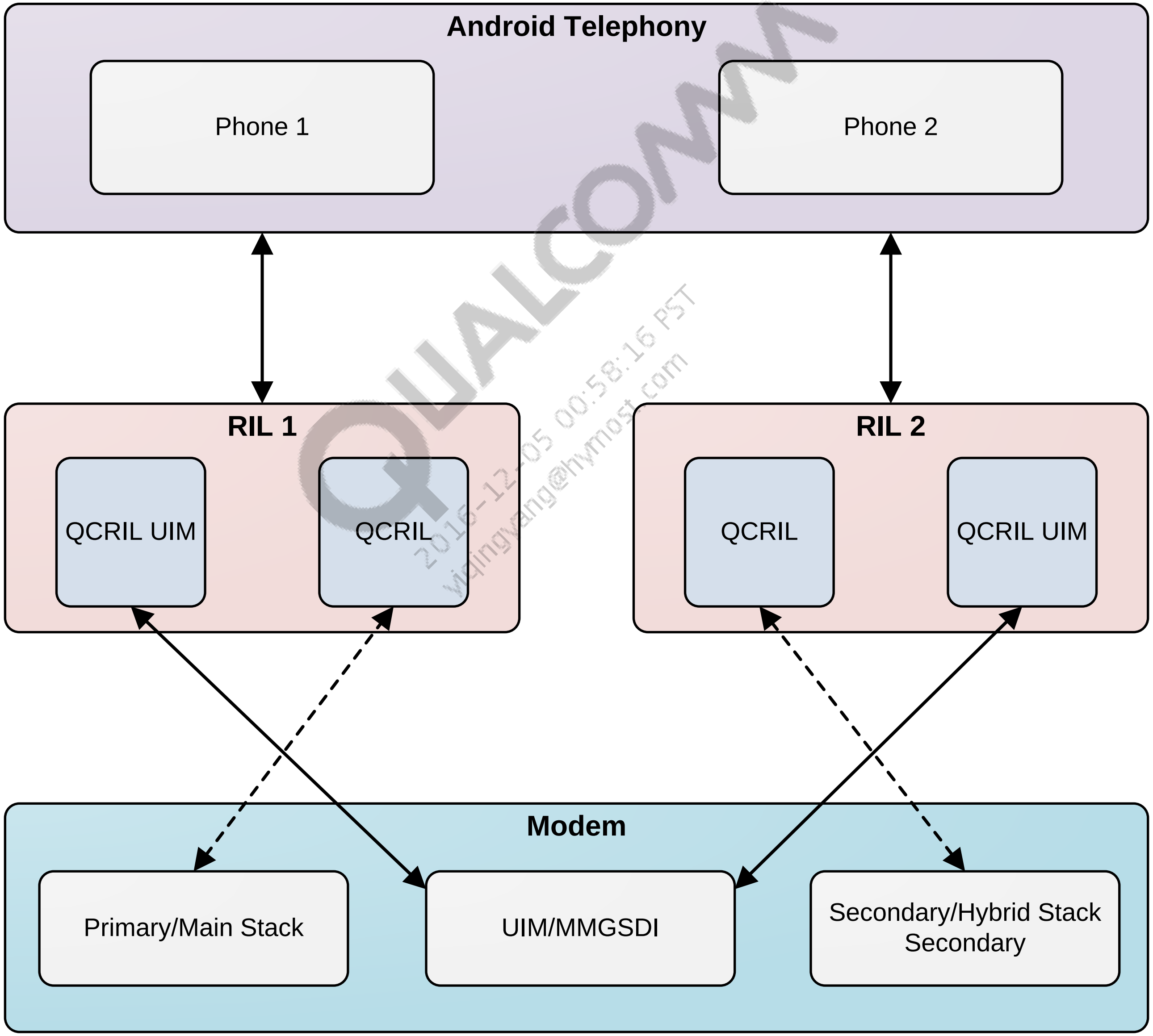
Feature Overview



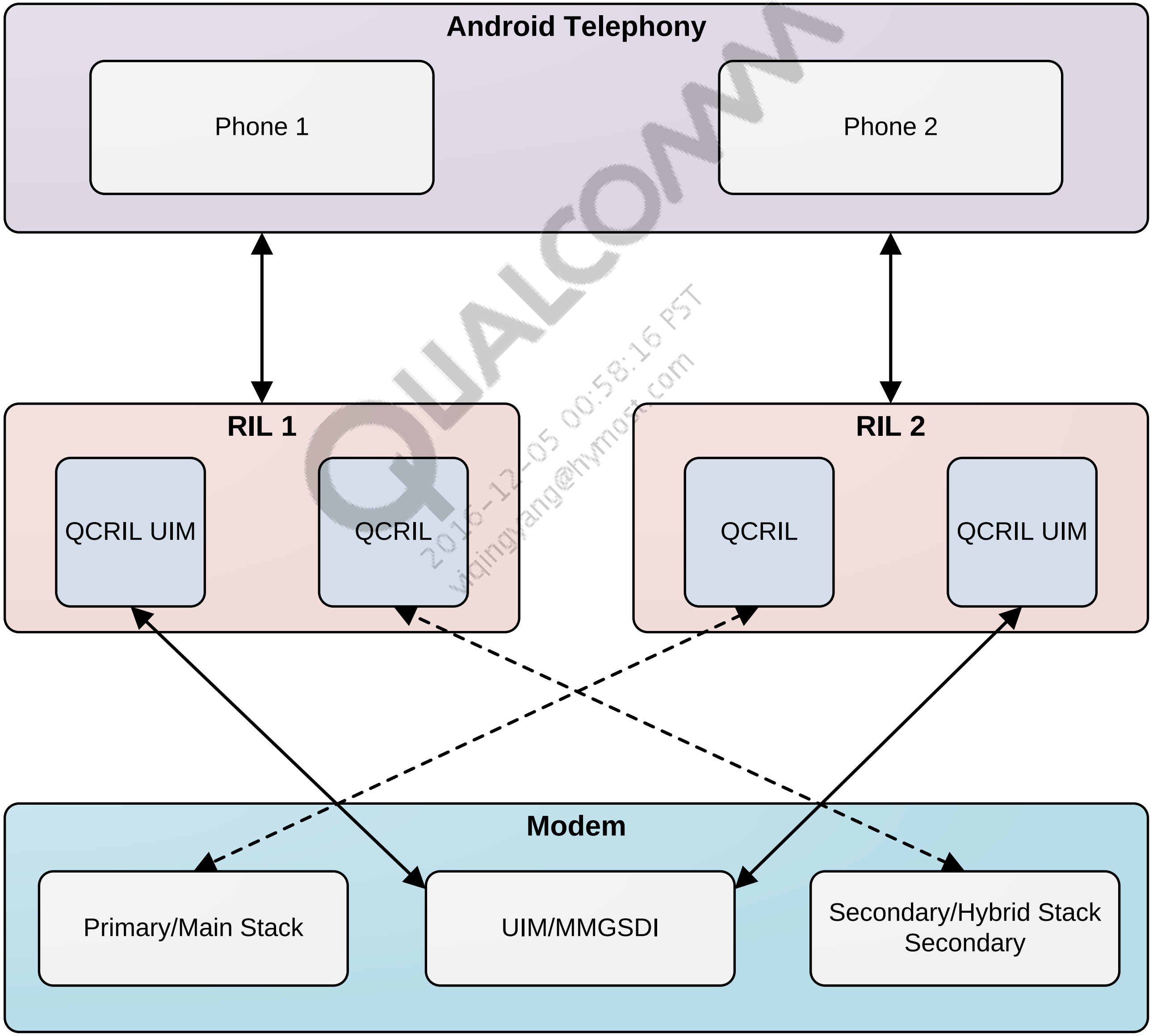
Feature Overview

- China Mobile Communications Corporation (CMCC) recommends Cross-/Flexi-Mapping.
- Cross-mapping supports dynamic mapping of modem stacks based on the network mode set for a subscription.
- On device bootup, Android telephony queries the modem capabilities on each of the stacks mapped to the subscription and stores them.
- Based on the network mode set for a subscription, Android telephony checks if the present stack supports the mode preference.
- If Network mode is not supported on a stack mapped to the current subscription, Android telephony performs cross-mapping of the stack to support Network mode.

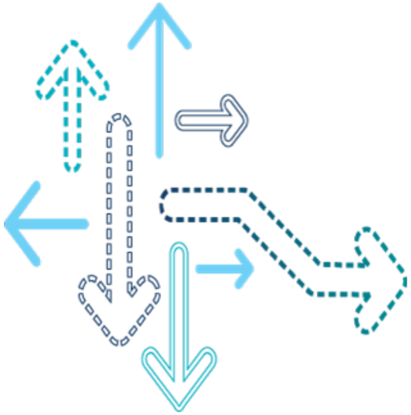
Straight-Mapping Architecture



Cross-Mapping Architecture



Interface Changes



Android Telephony and RIL Interface

- Two OEM HOOK functions are added to support cross-mapping from Android telephony and RIL:
  - QCRIL_EVT_HOOK_UPDATE_SUB_BINDING
  - QCRIL_EVT_HOOK_GET_MODEM_CAPABILITY

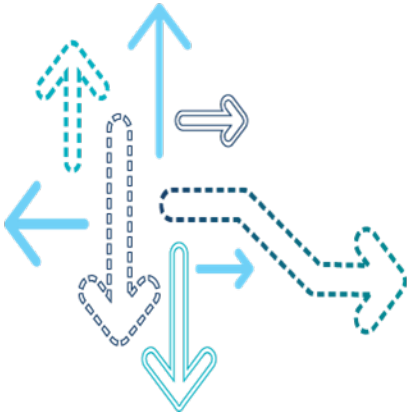
QCRIL_EVT_HOOK_UPDATE_SUB_BINDING

- Informs RIL to unbind/bind to a modem stack
- Request ID – QCRIL_EVT_HOOK_BASE + 36
- Request payload byte array (2 bytes)
  - byte[0] – Stack ID (0/1/2)
  - byte[1] – Enable/disable (0/1)
- Response payload – None

QCRIL_EVT_HOOK_GET_MODEM_CAPABILITY

- Gets modem capabilities from QCRIL
- QCRIL fetches this information from the modem using the existing QMI_DMS (QMI_DMS_GET_DEVICE_CAPABILITY)
- Request ID – QCRIL_EVT_HOOK_BASE + 35
- Request payload – None
- Response payload (byte[ ] – Byte array → 6 bytes)
  - byte[0] – Stack ID (0/1/2)
  - byte [1 to 4] – Bit vector representing the Radio Access Technology (RAT) supported per ril.h
  - byte[5] – subs_voice_data_capability

Cross-Mapping During Bootup



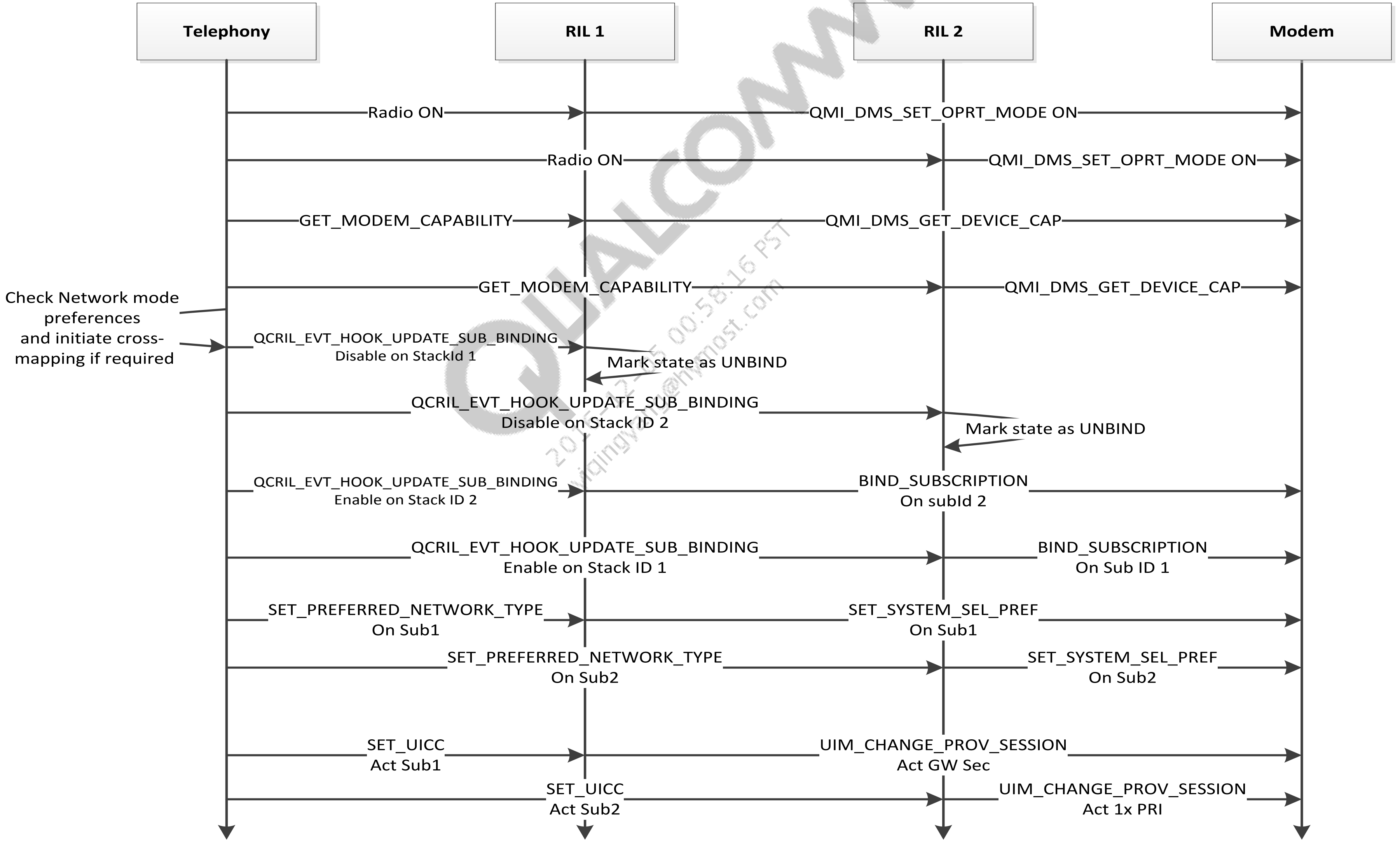
Android Telephony Preliminary Tasks

- Queries modem capabilities when radio is available
- Determines if cross-mapping is necessary based on modem capabilities and the current Network mode preference of the subscription
- If Android telephony detects that the modem is not capable of Network mode, cross-mapping is initiated

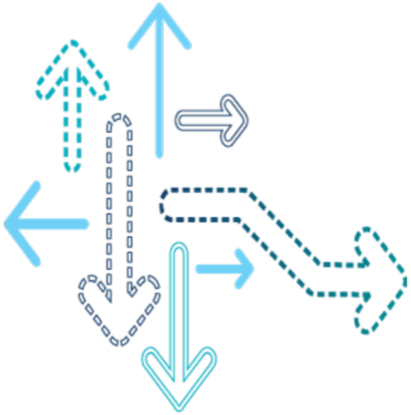
Cross-Mapping Tasks During Bootup

1. Android telephony requests RIL to disable subscription binding on all subscriptions, which keeps RIL in the UNBIND state.
2. After successful unbinding, Android telephony requests RIL to bind on a new stack ID on all subscriptions.
3. RIL issues BIND_SUBSCRIPTION on a new subscription ID with the modem.
4. After successful binding, a request for SET_SYSTEM_SEL_PREF is sent to the modem on all subscriptions with Network mode preferences.
   - If this or any previous steps fail, all steps are reverted.
5. Once the previous steps are successful, the subscriptions are activated and the designated data subscription is set.

Cross-Mapping During Bootup Call Flow


Telephony
RIL 1
RIL 2
Modem
Radio ON
QMI_DMS_SET_OPRT_MODE ON
Radio ON
QMI_DMS_SET_OPRT_MODE ON
GET_MODEM_CAPABILITY
QMI_DMS_GET_DEVICE_CAP
GET_MODEM_CAPABILITY
QMI_DMS_GET_DEVICE_CAP
Check Network mode preferences and initiate cross-mapping if required
QCRIL_EVT_HOOK_UPDATE_SUB_BINDING Disable on StackId 1
Mark state as UNBIND
QCRIL_EVT_HOOK_UPDATE_SUB_BINDING Disable on Stack ID 2
Mark state as UNBIND
QCRIL_EVT_HOOK_UPDATE_SUB_BINDING Enable on Stack ID 2
BIND_SUBSCRIPTION On subId 2
QCRIL_EVT_HOOK_UPDATE_SUB_BINDING Enable on Stack ID 1
BIND_SUBSCRIPTION On Sub ID 1
SET_PREFERRED_NETWORK_TYPE On Sub1
SET_SYSTEM_SEL_PREF On Sub1
SET_PREFERRED_NETWORK_TYPE On Sub2
SET_SYSTEM_SEL_PREF On Sub2
SET_UICC Act Sub1
UIM_CHANGE_PROV_SESSION Act GW Sec
SET_UICC Act Sub2
UIM_CHANGE_PROV_SESSION Act 1x PRI

Cross-Mapping – User-Initiated Network Mode Change



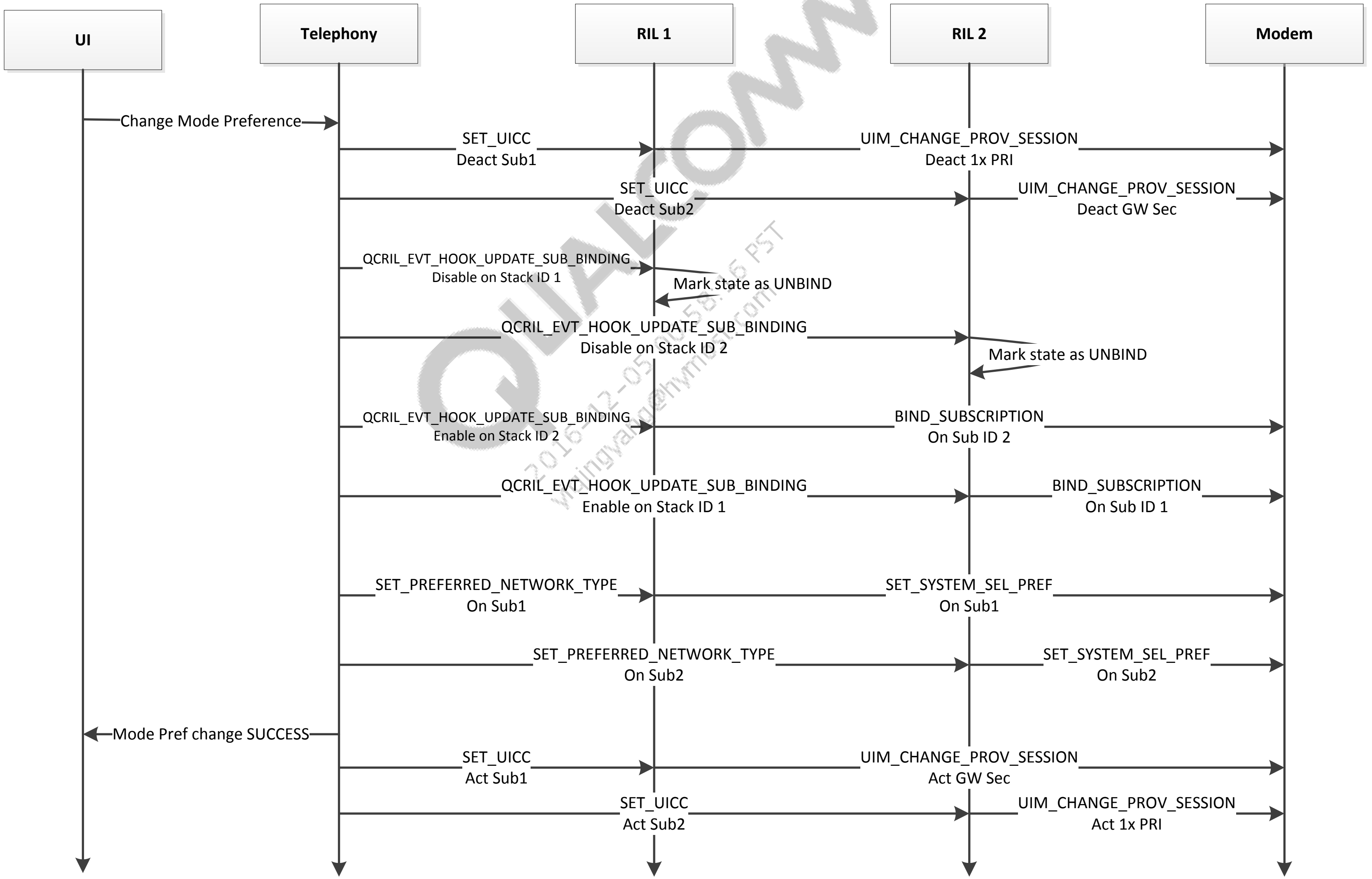
Cross-Mapping – User-Initiated Network Mode Change

- Based on the network mode set by the user for the subscription, Android telephony checks if the present stack is capable of Network mode.
- If Android telephony detects that the modem is not capable of Network mode, then Android telephony initiates cross-mapping.

Cross Mapping Tasks – User-Initiated Network Mode Change

1. Android telephony deactivates all activated subscriptions.
   - Upon successful deactivation, Android telephony requests RIL to disable the subscription binding on all subscriptions.
   - RIL remains in the UNBIND state.
2. Upon successful unbinding, Android telephony requests RIL to bind on new stack IDs on all subscriptions.
   - RIL issues BIND_SUBSCRIPTION on the new subscription ID with the modem.
3. A request for SET_SYSTEM_SEL_PREF is sent to the modem on all subscriptions with a new mode selected.
   - If this or any previous steps fail, all steps are reverted.
4. Android telephony reactivates any deactivated subscriptions and sets the designated data subscription.

Cross-Mapping – User Initiated Network Mode Change Call Flow



UI

Telephony

RIL 1

RIL 2

Modem

Change Mode Preference

SET_UICC
Deact Sub1

UIM_CHANGE_PROV_SESSION
Deact 1x PRI

SET_UICC
Deact Sub2

UIM_CHANGE_PROV_SESSION
Deact GW Sec

QCRIL_EVT_HOOK_UPDATE_SUB_BINDING
Disable on Stack ID 1

Mark state as UNBIND

QCRIL_EVT_HOOK_UPDATE_SUB_BINDING
Disable on Stack ID 2

Mark state as UNBIND

QCRIL_EVT_HOOK_UPDATE_SUB_BINDING
Enable on Stack ID 2

BIND_SUBSCRIPTION
On Sub ID 2

QCRIL_EVT_HOOK_UPDATE_SUB_BINDING
Enable on Stack ID 1

BIND_SUBSCRIPTION
On Sub ID 1

SET_PREFERRED_NETWORK_TYPE
On Sub1

SET_SYSTEM_SEL_PREF
On Sub1

SET_PREFERRED_NETWORK_TYPE
On Sub2

SET_SYSTEM_SEL_PREF
On Sub2

Mode Pref change SUCCESS

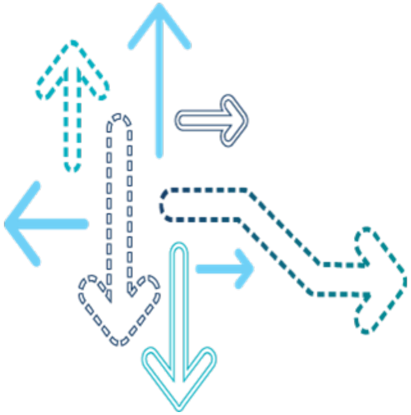
SET_UICC
Act Sub1

UIM_CHANGE_PROV_SESSION
Act GW Sec

SET_UICC
Act Sub2

UIM_CHANGE_PROV_SESSION
Act 1x PRI

Additional Information



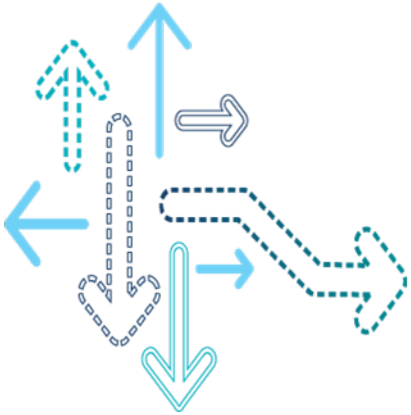
Additional Information

- Android telephony must revert the subscription mapping if any of the following operations fail:
  - Subscription deactivation
  - Unbind/bind
  - Mode preference request
- The modem must report modem capability as GSM-only on the secondary or tertiary stack.
- Cross-mapped subscriptions must be deactivated before unbind/bind requests.
  - During Mobile-Originated (MO)/Mobile-Terminated (MT) calls, MO/MT SMS functionality is not expected until subscriptions are activated.

Additional Information

- Airplane mode enable/disable and emergency calls must be supported even when cross-mapping is in progress.
- Android telephony does not initiate the cross-mapping procedure if an emergency call is running.
- When exchanging SIMs in slots, Network mode must be explicitly set to trigger cross-mapping.

Additional Features



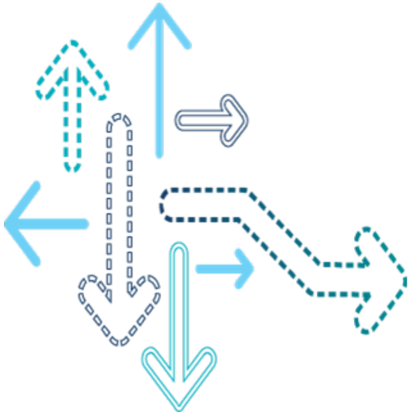
Primary Card Features

- Features are designed for CMCC.
- Based on the IIN value of the SIM card, the device detects whether the inserted SIM card is a CMCC card. If it is a CMCC card, then it is set as primary card.
- After the primary card is detected, the network mode on the Primary card subscription is set to 20 (TD-SCDMA, GSM/WCDMA, and LTE) if a USIM application is present or 18 (TD-SCDMA, GSM/WCDMA) if a SIM application is present. Network mode on another subscription is set as GSM. This initiates cross-mapping if required.
- Designated Data Subscription (DDS) is set on the Primary card irrespective of the previous DDS preference.
- If both cards are CMCC cards, a notification is provided during bootup to select one card as Primary card. The user can choose Primary card from the SIM settings menu.
- This feature is controlled through ADB property *persist.radio.primarycard*. If the property is set to TRUE, then the primary card feature is enabled.

7 + 5 Mode

- This mode supports LTE on the primary/secondary stack which has DDS. Without this feature, when DDS is on the secondary stack, LTE cannot be supported on that stack.
- The primary stack supports 7-mode, i.e., LTE-FDD, LTE-TDD, WCDMA, TD-SCDMA, GSM, HRPD, CDMA. The secondary stack supports 5-mode, i.e., LTE-FDD, LTE-TDD, WCDMA, TD-SCDMA, GSM. These are the static capabilities or max capabilities that can be supported on the stacks.
- Stacks operate in different modes based on the DDS set by the user, e.g., if DDS is on the primary stack, then the stack operates in 7-mode and the secondary stack operates in GSM. If the user changes DDS to secondary stack, then the primary stack operates in CDMA, HRPD, and GSM, and the secondary stack operators in LTE-FDD, LTE-TDD, WCDMA, TD-SCDMA, and GSM.
- If DDS changes due to the MMS received, then the stacks' operating mode does not change. The stacks operate in different modes only for user-triggered DDS changes.

Cross-Mapping – Android Marshmallow



Cross-Mapping – Android Marshmallow

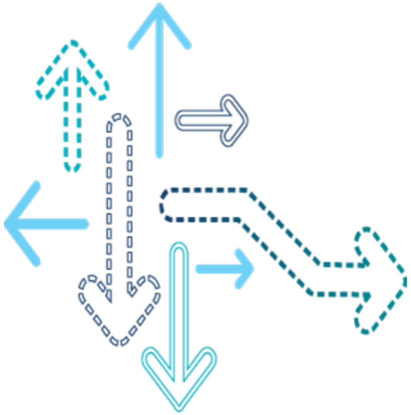
- Google provides cross-mapping support.
- Google added the following new RIL requests/indications:
  - RIL_REQUEST_GET_RADIO_CAPABILITY
  - RIL_REQUEST_SET_RADIO_CAPABILITY
  - RIL_UNSOL_RADIO_CAPABILITY
- QTI-added OEM HOOK functions are no longer used:
  - QCRIL_EVT_HOOK_GET_MODEM_CAPABILITY
  - QCRIL_EVT_HOOK_UPDATE_SUB_BINDING
- As per Google Marshmallow code, cross-mapping is triggered based on the DDS change; e.g., if the user changes DDS from SIM1 to SIM2, then cross-mapping is triggered and maps primary stack to SIM2 and secondary stack to SIM1. Network mode preferences on SIM1 and SIM2 are not changed.
- State machine is implemented in ProxyController.java while triggering flex map. Different states are SET_RC_STATUS_IDLE, STARTING, STARTED, APPLYING, SUCCESS, and FAIL.

Cross-Mapping – Android Marshmallow

- QTI supports trigger cross-mapping based on the user-initiated network mode change.
- ADB property persist.radio.flexmap_type is added to either of the following values:
  - none – Cross-mapping is disabled
  - dds – Cross-mapping is triggered based on the DDS change as per the Google Marshmallow code
  - nw_mode – Cross-mapping is triggered based on the user-initiated network mode change
  - If the ADB property is not set, then cross-mapping is triggered based on the user-initiated network mode change.
- The QTI file QtiRadioCapabilityController.java decides whether cross-mapping should be triggered or not when the user changes the network mode from the UI. To trigger cross-mapping, the Google provided setRadioCapability() API is invoked in ProxyController.java, else Network mode is set.

References

Acronyms	
Term	**Definition**
CMCC	China Mobile Communications Corporation
DDS	Designated Data Subscription
MO	Mobile-Originated
MT	Mobile-Terminated



Questions?

https://createpoint.qti.qualcomm.com