

eSIM management on Qualcomm phones

FrOSCon 2024

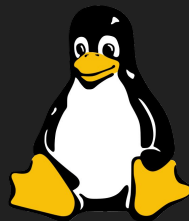
2024-08-18

About me

Luca Weiss

postmarketOS, Linux kernel, OpenRazer

Android Platform Engineer @ Fairphone



fosstodon.org/@z3ntu

FAIRPHONE

Agenda

- What is eSIM?
 - Removable eSIM, LPA, etc.
- Qualcomm modem basics
 - QMI, QRTR
- Open-source implementation
- My “eSIM Manager” GUI app

Notes about this talk

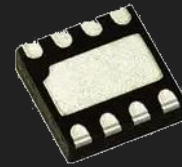
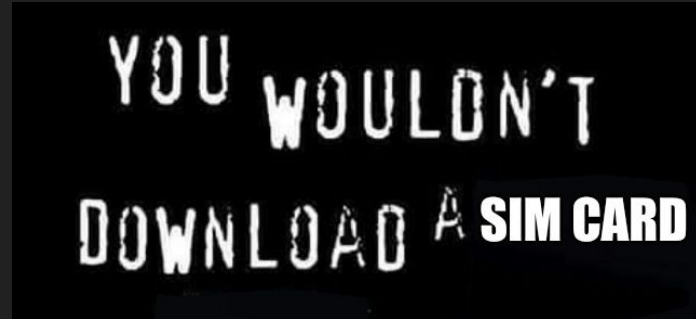
- There will be simplifications
- There will (probably) be mistakes
- I hope you learn things regardless

eSIM & eUICC

eSIM = embedded Subscriber Identity Module

eUICC = embedded Universal integrated circuit card


- Traditional: Physical SIM card from operator
- eSIM: No physical card required, can download it instead
- First adopted in smartwatches
- Talking about consumer eUICC (SGP.22)
 - Not M2M eUICC (SGP.02) or IoT eUICC (SGP.32)
- Most components in architecture are certified by GSMA
 - Except for Local Profile Assistant (LPA)
 - No non-certified profiles on your commercial eSIM
 - No certified profiles on your home-grown eSIM



Removable eSIM?

- eUICC is normally a chip on PCB
- But you can also put it on normal e.g. 4FF form factor
- Management works the same
- Android has support for app authorization via ARA-M
 - Using OMAPI
 - ARA-M is just an Android-level restriction
 - iOS has no support for removable eSIM management (exc.: STK)

sysmocom
systems for mobile communications GmbH

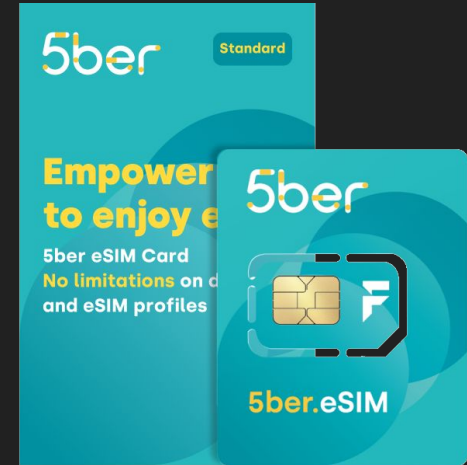


sysmocom
eUICC1-C2G

- Expertise in protocol R&D from 2G to 5G, RAN to CN
- Support and development for Osmocom • open5gs
- small-cell cellular base station hardware
- GSM, UMTS and LTE networks in the box (NITB)
- SIM cards, accessories, tracers, remote SIM, eUICC

Please support <https://osmocom.org/>
a community creating projects related to Open source mobile communications including (among many other things) the **pySim** software you can use to work with this eUICC and its eSIMs. Osmocom relies on contributions, whether by code, documentation improvements or financially.

sysmoEUIICC1-C2G



Local Profile Assistant (LPA)

- Software which runs on the UE (“user equipment”, e.g. phone)
- Interfaces with the SM-DP+ (via HTTPS)
 - SM-DP+ : Subscription Manager - Data Preparation Plus
- Interfaces with the eUICC
- Purposes:
 - Manage eSIM profiles (enable, disable, delete)
 - Download eSIM profiles from the SM-DP+
 - Retrieve ‘notifications’ from the eUICC and send them to the SM-DP+
- Android: proprietary Google LPA as part of GMS bundle (`com.google.android.euicc`)
- Similar with iOS, Windows, etc.

eSIM in Linux Mobile space

- Could do eSIM management e.g. on Android and reboot back
 - Example: download eSIM, enable it, reboot back to Ubuntu Touch
- Configured eSIM behaves like regular SIM card
- No open-source LPA until recently
 - “Ipac” project with initial commit 2023
 - Only PC/SC and AT interface until even more recently

Modem communication basics on Qualcomm

- QRTR/QIPCRTR - Qualcomm IPC router

- Network protocol - address family: AF_QIPCRTR
- Transport: Shared Memory Driver (SMD) channel, or e.g. MHI for external modems
- Think of UDP, but with 'node' number and 'port' number instead of IPv4 and port number
- Running service:
 - Service Version Instance Node Port
 - 11 1 0 0 82 User Identity Module service
- Service: Identifies a given service (e.g. 4097 = DIAG service)
- Version: Sometimes used for versioning services for backwards compatibility
- Instance: Semi-random number, important if a service is running multiple times
- *^ above values are registered at nameserver, used for lookup to get port*
- Node: Processor that service is running on (e.g. 0=modem, 1=Linux, 5=adsp, 10=cdsp)
- Port: Port number to send packets to (like port in TCP/IP), needs to be unique

Modem communication basics on Qualcomm

- QMI - Qualcomm MSM Interface
 - 1. Way to encode/encode structured data into binary
 - 2. Collection of defined interfaces, specific to Qualcomm devices
- Type-Length-Value (TLV) encoding
- Messages are requested by client & will get response
- Indications are unsolicited messages, e.g. notifications
- Functions grouped into services
 - e.g. UIM service (0x0B): call 0x28 is “Change PIN”
 - Input parameter old PIN + new PIN
 - Output: Result
- List in libqmi repository: `data/*.json`

Managing eSIMs on Qualcomm

- Any LPA ‘driver’ in Ipac needs these operations:
 - Open logical channel
 - Send APDU (and receive response)
 - Close logical channel
- We send QMI-encoded messages over QRTR socket to modem
- Operations are separate QMI calls in UIM (User Identity Module) service
 - “Open logical channel” (0x0042), AID as parameter, returns Channel ID
 - “Send APDU” (0x003B), Channel ID and APDU contents as parameter, returns APDU resp.
 - “Close logical channel” (0x003F), Channel ID as parameter

First proof of concept

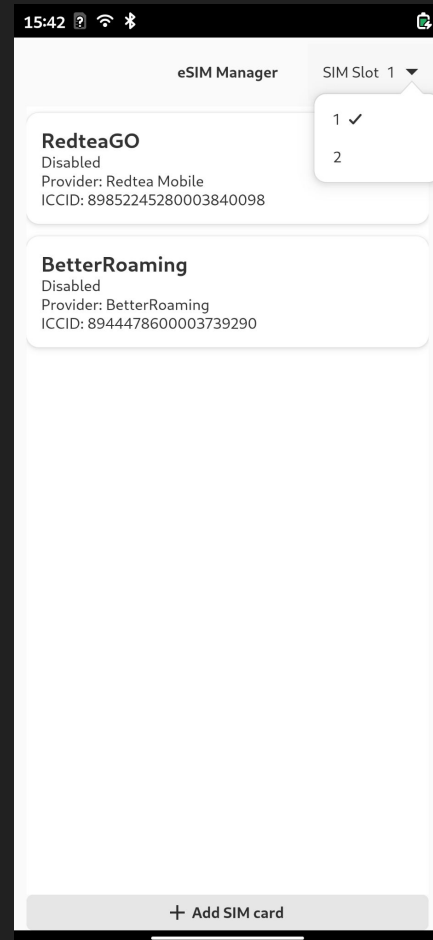
- Required QMI calls implemented in libqmi
 - Command line util “qmcli” with new switches such as --uim-send-apdu=slot,channel,apdu
- “lpac” - C-based eUICC LPA
 - Stdio ‘driver’: Prints requests on stdout in json format, expects responses on stdin
- Developed Python wrapper script around that
 - Parses json request, calls qmcli, sends json reply back to lpac
- It works!

Implementation in “Ipac”

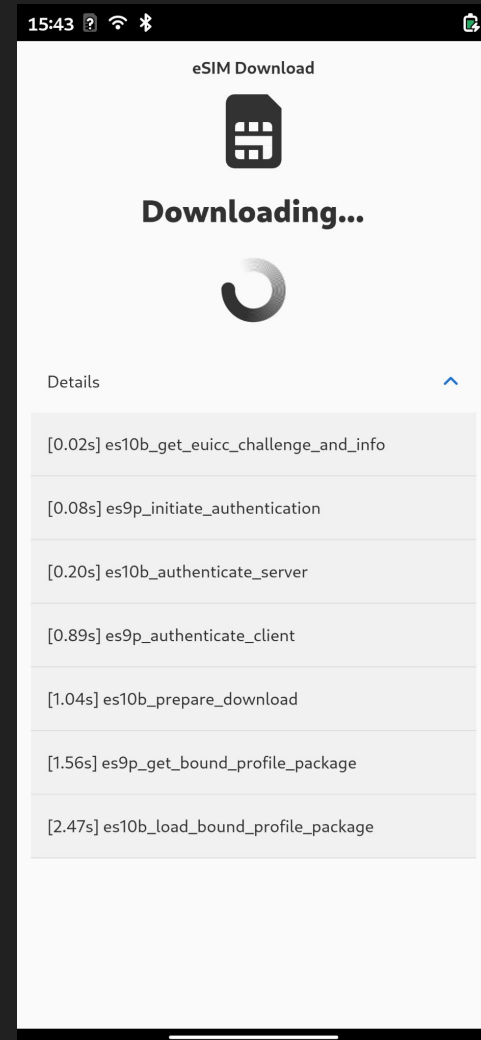
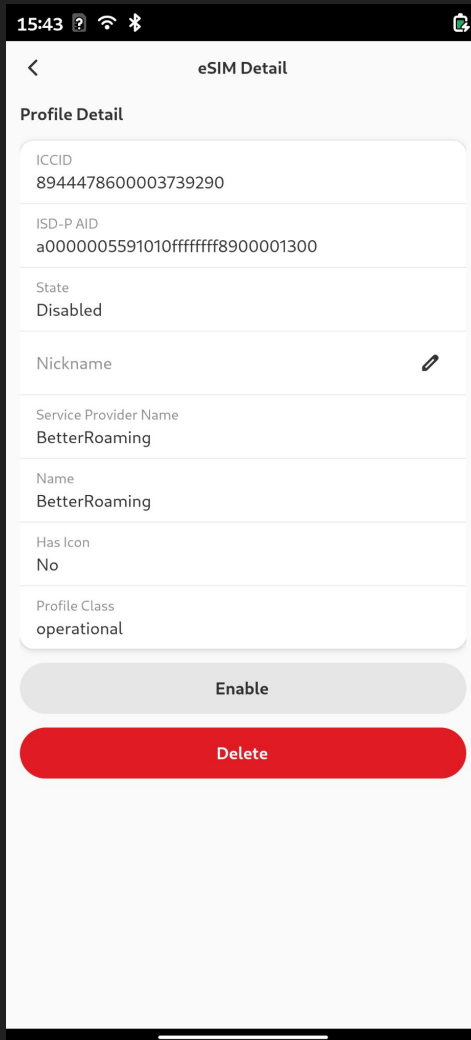
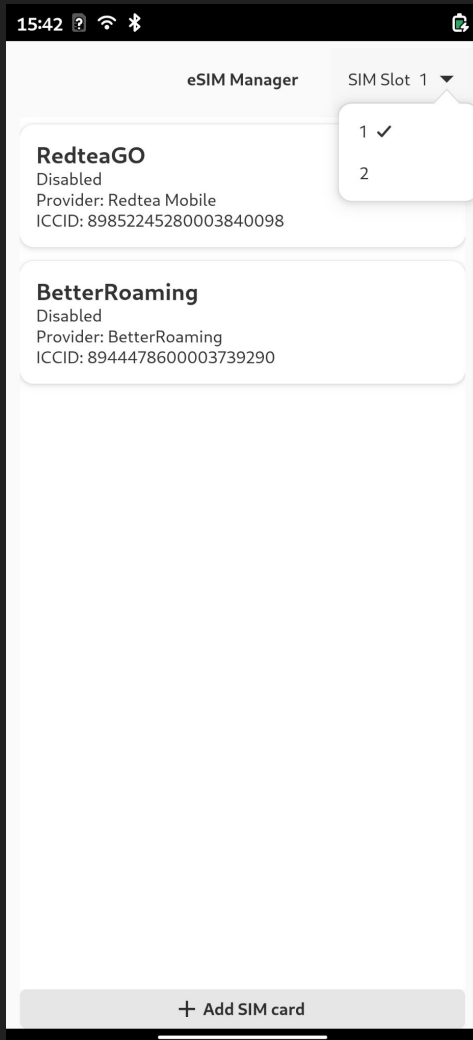
- Using glib async API of libqmi & libqrtr inside synchronous Ipac API
 - Not super easy, but not the first person to do that
- Replicated wrapper script functionality in C
- Also works!
- QMI-over-QRTR backend was merged - included in Ipac v2.0.2
 - Phones, smartwatches(?), tablets(?)
- QMI-over-chardev backend was merged July 24th
 - e.g. USB-attached modems: Quectel RM520N
- QMI-over-MBIM backend still missing: [Ipac#94](#)
 - e.g. PCI-attached modems: Foxconn T99W175

eSIM Manager app

- June 2024 released “eSIM Manager” (lpa-gtk)
- GUI for lpacli
- GTK4 & libadwaita
- Basic but functional as of now
- Missing features - please contribute!
 - There's a dummy backend you can run easily without an actual eSIM
 - Non-code contributions (design advice or logo) are very welcome!







Thank you!