OPEN SOURCE FOR ESIM INTEGRATION

Harald Welte <hwelte@sysmocom.de>

ABOUT THE SPEAKER

- Linux user + admin since 1994
- FOSS developer since late 1990s
- former kernel hacker (mostly netfilter/iptables)
- working with smart cards for 25+ years
- working on FOSS implementations of cellular protocol stacks since 2008
  - see osmocom.org for most of that work
- running a small company of FOSS cellular hackers called sysmocom
  - we have SIM card and eUICC products + do a lot of eSIM related R&D

RECAP: CLASSIC SIM

- SIM card introduced with GSM (2G)
- single-purpose smart card
- fundamental purpose of classic SIM:
  - storage of configuration data
  - storage of contacts
  - storage of SMS
  - *secure element* holding the secret per-subscriber key material
- From UMTS onwards, it’s technically an USIM, not a SIM
  - USIM is a logical application on an UICC

ESIM IN A NUTSHELL

- virtualisation of the USIM
- de-coupling of *SIM profile* from *physical chip*
- the eSIM capable chip is now called **eUICC**
- eUICC can hold one or more *SIM profiles*

ESIM FUNDAMENTALS

UICC

The classic ETSI chip card for USIM, ISIM, etc.

eUICC

The eSIM capable UICC

USIM

The SIM type first introduced with UMTS/3G

SM-DP+

The server from where eSIM profiles are downloaded

LPA

The software on the UE connecting eUICC to SM-DP+

TERMINOLOGY: ESIM VS EUICC

- eUICC is unambiguously the physical chip
- eSIM sometimes is used to refer to
  - the **eSIM profile**, i.e. a logical USIM downloaded into the eUICC
  - the eUICC [which I consider to be wrong use of terminology]

To avoid confusion, I’m always using eUICC when referring to the chip.

WHAT IS AN ESIM PROFILE?

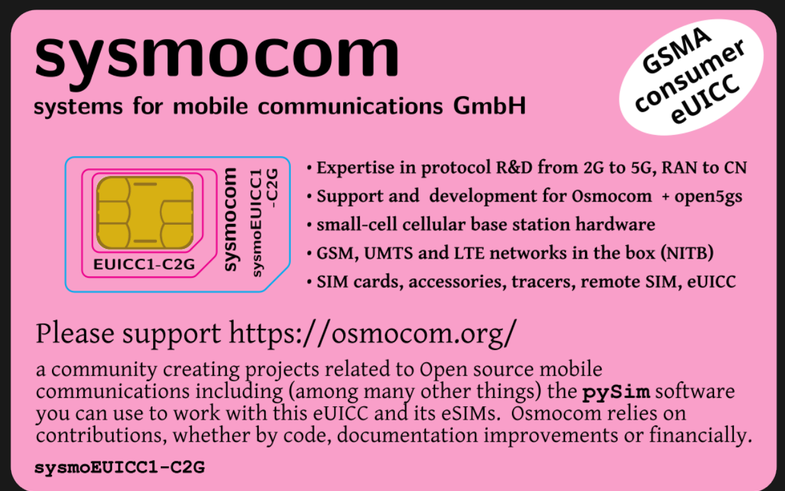
- think of it as glorified tarball of USIM filesystem + metadata
  - can also contain executable code (java applets), just like classic USIM
- parsed by eUICC to build something that looks like the classic USIM to cellular modem
- signed by HSM in SM-DP+ so eUICC can verify authenticity
- encrypted with eUICC-specific public key to prevent cloning
- [before download] identified by *activation code* usually represented as QR-code

EUICC FUNDAMENTALS

- electrical interface just like USIM/UICC
- data transmission just like USIM/UICC
- eUICC with activated eSIM profile behaves just like USIM
  - a cellular modem doesn’t need any change. It talks to the USIM as usual
- mechanical form-factor *usually* soldered instead of pluggable

REMOVABLE EUICCS

- an eUICC doesn't have to be soldered
  - removable eUICCs in plastic card form factor
  - to retrofit older UEs without eSIM support
  - for R&D, testing and debugging



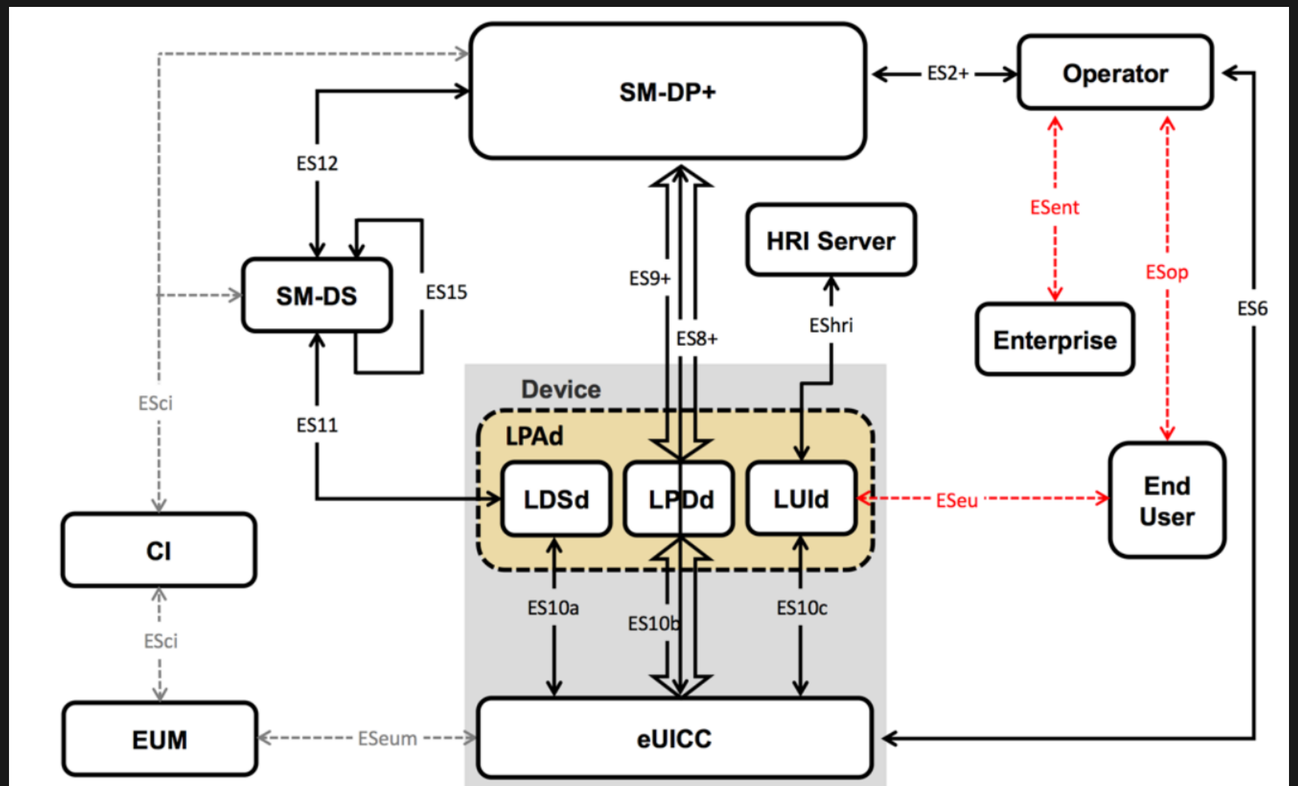
ESIM RECAP: THE THREE FLAVOURS

Spec	**eSIM RSP flavour**
SGP.01 + SGP.02	M2M eSIM RSP
SGP.21 + SGP.22	Consumer eSIM RSP
SGP.31 + SGP.32	IoT eSIM RSP

In this talk we focus on **SGP.21/22 Consumer eSIM**, though some parts are generally applicable.

CONSUMER ESIM MODEL

- uses client-driven *pull* model to provision and manage profiles by the end user
- HTTPS/TCP/IP protocol used for profile download
- Introduces SM-DS and LPA (both don’t exist in prior M2M)
- no *provisioning profile* required, as device can e.g. use WiFi to download a profile



Remote SIM Provisioning for Consumer Architecture

CONSUMER ESIM: KEY ELEMENTS

- eUICC: chip with certificate + private key
- SM-DP+: server from which eSIM profile is downloaded
- LPA: client software performing download via HTTPS
  - side-note: exotic encoding: HTTP contains JSON containing base64-encoded ASN.1 DER

CONSUMER ESIM: SM-DP+

SM-DP+

Subscription Manager - Data Preparation (Enhanced)

- secure back-end server
- mutually authenticates with the eUICC, verifying its certificate
- creates the eSIM profile containing operator key material + configuration
- signs and encrypts it with eUICC-specific public key, so only that single eUICCs private key can decrypt it
- most often operated as SaaS by an independent entity under contract of multiple MNO/MVNOs

CONSUMER ESIM: LPA

LPA: Local Profile Assistant

- software on the client device
  - can optionally also be fully built-in to the eUICC (LPAe)
  - standard case with smartphones today is on the device (LPAd)
- provides user interface for profile management
- talks to local eUICC and to remote SM-DP+

CONSUMER ESIM: LPA OPERATIONS

- profile download (SM-DP+ to eUICC)
- local profile management
  - enable profile
  - disable profile
  - delete profile
- processing of notifications (eUICC to SM-DP+)

GSMA ESIM SECURITY

GSMA SAS (Security Accreditation Scheme)

- SAS-SM for SM-DP+ operators
- SAS-UP for eUICC manufacturers
- very comprehensive security requirements covering most but not all parts of a related product, but also includes topics staff hiring processes, physical site access security and business continuity.
- accreditation requires ongoing re-audits every two years

Only accredited vendors receive X.509 certificates derived from GSMA CA (root-of-trust)

Operating a SM-DP+ or issuing eUICCs is very costly due to SAS accreditation

ESIM VS. OPEN SOURCE SOFTWARE

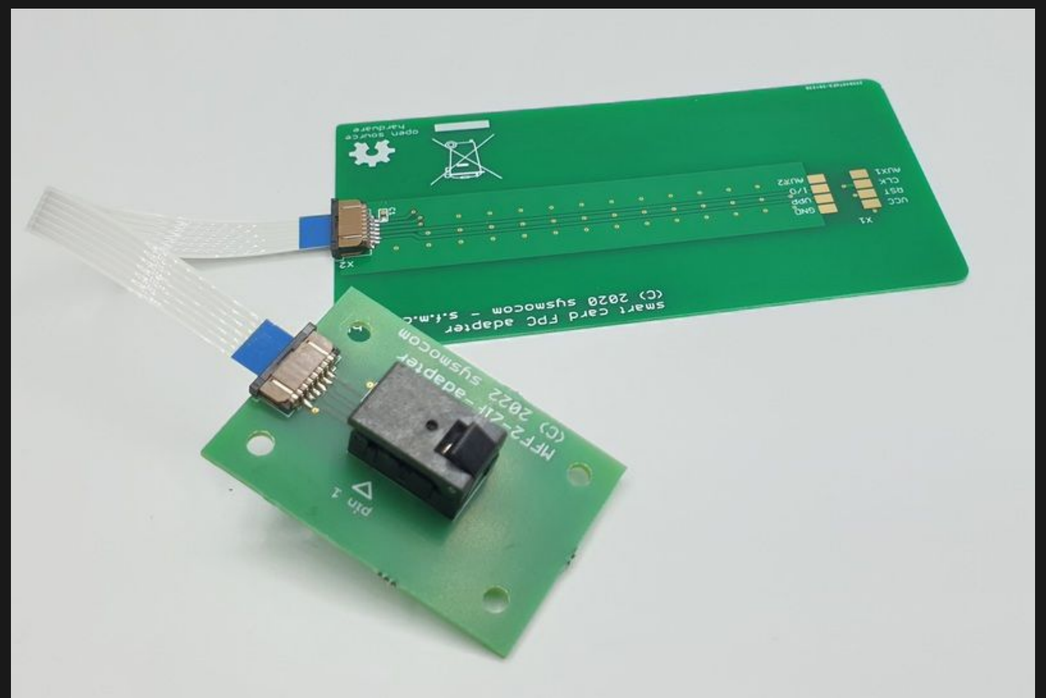
- technically all protocol specs are public
  - FOSS implementations of all elements are possible in theory
- GSMA accreditation is very expensive
  - unlikely to happen given the associated cost
- only the LPA does not need any kind of accreditation, so FOSS LPA are possible
  - `lpac` available from https://github.com/estkme-group/lpac
  - `LPAdesktop` available from https://github.com/Truphone/LPAdesktop
- lab-grade SM-DP+ available for private / custom CA
  - `osmo-smdpp` is part of `pySim`, see https://downloads.osmocom.org/docs/pysim/master/html/osmo-smdpp.html

THE lpac LPA IMPLEMENTATION

- stand-alone C-language FOSS implementation of the SGP.21/22 LPA
- https://github.com/estkme-group/lpac
- Modular architecture with APDU and HTTP *drivers*
- UI/OS integration with Windows, Linux, macOS, Android, Linux Mobile
- Interface towards UI or management implemented via JSON output on stdout

HOW `lpac` TALKS TO THE EUICC

- for desktop use: Interfaces eUICC via `pcsc-lite`, like any smartcard
- use USB CCID smart card reader with mechanical adapter, as needed



lpac PROFILE DOWNLOAD

```
$ ./lpac profile download -s testsmdpplus1.example.com -m 1234
{"type":"progress","payload":{"code":0,"message":"es10b_get_euicc_challenge","data":null}}
{"type":"progress","payload":{"code":0,"message":"es10b_get_euicc_info","data":null}}
{"type":"progress","payload":{"code":0,"message":"es9p_initiate_authentication","data":null }}
{"type":"progress","payload":{"code":0,"message":"es10b_authenticate_server","data":null}}
{"type":"progress","payload":{"code":0,"message":"es9p_authenticate_client","data":null}}
{"type":"progress","payload":{"code":0,"message":"es10b_prepare_download","data":null}}
{"type":"progress","payload":{"code":0,"message":"es9p_get_bound_profile_package","data": null}}
{"type":"progress","payload":{"code":0,"message":"es10b_load_bound_profile_package","data": null}}
{"type":"lpa","payload":{"code":0,"message":"success","data":null}}
```

lpac LIST INSTALLED PROFILES

```
$ ./lpac profile list | json_pp
{
  "payload" : {
    "code" : 0,
    "data" : [
      {
        "iccid" : "89000123456789012341",
        "isdpAid" : "A0000005591010FFFFFFFF8900001100",
        "profileClass" : 2,
        "profileName" : "GSMA Generic eUICC Test Profile 1A",
        "profileState" : 0,
        "serviceProviderName" : "GSMA Test 1A"
      },
      {
        "iccid" : "89000123456789012358",
        "isdpAid" : "A0000005591010FFFFFFFF8900001200",
        "profileClass" : 2,
        "profileName" : "OsmocomProfile",
        "profileState" : 0,
        "serviceProviderName" : "OsmocomSPN"
      }
    ],
    "message" : "success"
  },
  "type" : "lpa"
}
```

lpac PROFILE ENABLE, DISABLE, DELETE

```
$ ./lpac profile enable 89000123456789012358
{"type":"lpa","payload":{"code":0,"message":"success","data":null}}

$ ./lpac profile disable 89000123456789012358
{"type":"lpa","payload":{"code":0,"message":"success","data":null}}

$ ./lpac profile delete 89000123456789012358
{"type":"lpa","payload":{"code":0,"message":"success","data":null}}
```

CONSUMER ESIM SUPPORT IN EMBEDDED LINUX DEVICES

Assumptions:

- we’re building an embedded device containing a cellular modem
- we want it to have SGP.21 consumer eSIM support

Consequences:

- we need an eUICC instead of a SIM/USIM
- we need LPA software to download/manage eSIM profiles

REPLACING SIM/USIM WITH EUICC

- no hardware change if removable plastic-card eUICC is used
- small hardware change if solder-type eUICC is used
  - replace mini/micro/nano-SIM slot with MFF2 package footprint

ACCESSING EUICC FROM LINUX BASED LPA

- eUICC is attached to cellular modem like classic [U]SIM
- same electrical interface and protocol
- LPA software will reside on Linux-running application processor

Result: LPA needs access to the SIM card interface

ACCESSING EUICC FROM LINUX BASED LPA

Common methods:

- AT-command based transport
  - analog modems used to have AT commands back in the 1980s
  - cellular modems often also used AT commands (3GPP TS 27.007)
  - there are AT commands to open channels + exchange APDUs (`AT+CCHO`, `AT+CCHC`, `AT+CGLA`)
- QMI based transport on Qualcomm modems
  - UIM (User Identity Module) Service
- other vendor-specific proprietary mechanisms

TESTING ESIM SUPPORT

- testing using real-world cellular subscriptions is expensive
- GSMA specified SGP.26 test certificates
  - requires special eUICC with certificates within SGP.26 root of trust
  - root CA cert + private key is publicly specified!
  - anyone can create + sign eSIM profiles
- unlimited number of profile downloads e.g. from
  - https://test.rsp.sysmocom.de/ or
  - self-hosted `osmo-smdpp` instance

Of course there's no access to commercial production network with your self-signed/issued eSIM profiles :)

SGP.01/02 M2M ESIM MODEL

- centralized, remote control
- does not need LPA software on client
- works entirely over SIM OTA channels
  - modem support varies
  - operator support varies
- requires special M2M subscription with MNO/MVNO
- cannot download consumer eSIM profiles
- didn’t deliver on promise of easy migration from one MNO/MVNO to another.

OUTLOOK: SGP.31/32 IOT ESIM MODEL

- best of both worlds
- can use any consumer eSIM profile of any MNO/MVNO
- like consumer eSIM but with LPA split in two halves
  - **upper half** of LPA becomes eIM software running on backend server
  - **lower half** of LPA becomes IPA
- currently at the brink of production deployment
  - first GSMA SAS accredited eUICC and eIM are showing up

FURTHER INFORMATION

- https://euicc-manual.osmocom.org/
- https://www.gsma.com/esim/wp-content/uploads/2018/12/esim-whitepaper.pdf
- some of my other eSIM talks
  - https://media.ccc.de/v/camp2023-57190-demystifying_esim_technology
  - https://media.ccc.de/v/osmodevcall-20241016-laforge-gsma-iot-esim
- https://programm.froscon.org/2024/system/event_attachments/attachments/000/000/857/original/FrOSCon_2024_-_eSIM_management_on_Qualcomm_phones.pdf
- Forum at https://discourse.osmocom.org/c/sim-card-technology/esim-euicc/8

EOF

End of File