Google

5G in ModemManager

What a journey

Aleksander Morgado - FOSDEM 2024

Who am I

- Telecommunications Engineer

- +10 years as **ModemManager maintainer** and **developer**
  - Joined the project around MM 0.5
  - Helped reviewing the MM 1.0 DBus API
  - Developed and released libqmi and libmbim
  - Developed and released MM 1.0

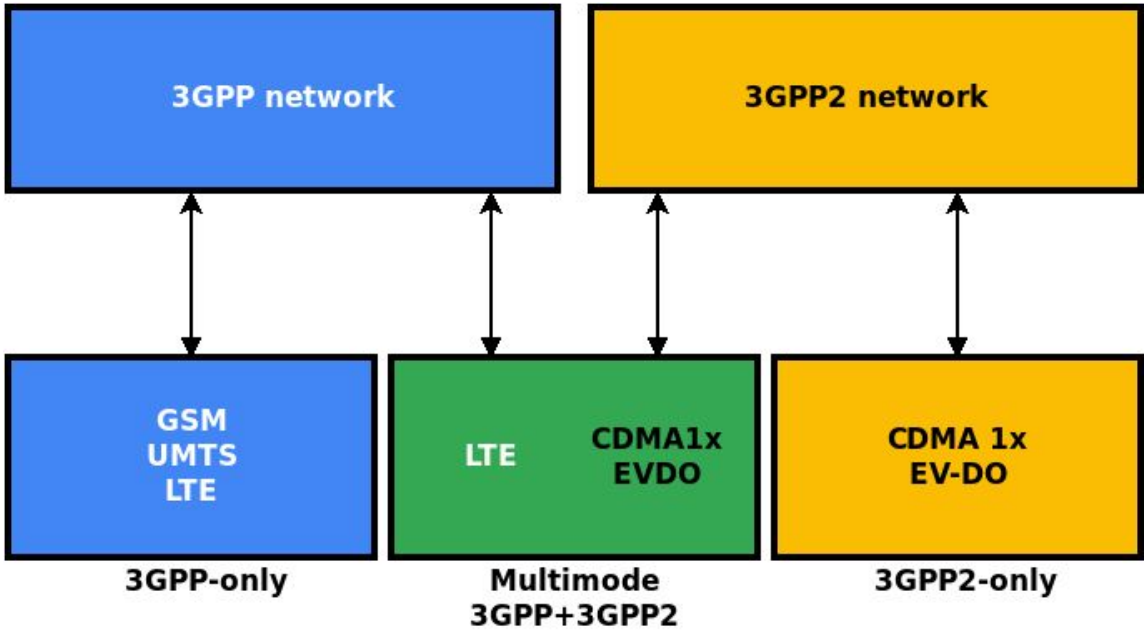
- Working at the **ChromeOS Cellular team** @ Google since 2022

How 4G support was added in MM

MM < 1.0 and the support for 4G

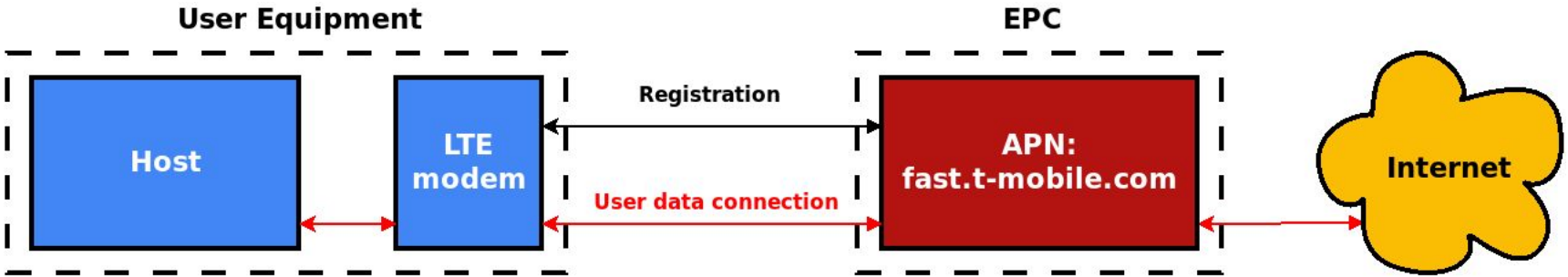
- **4G**
  - `MM_MODEM_GSM_ACCESS_TECHNOLOGY_LTE`

MM 1.0 and the support for 4G



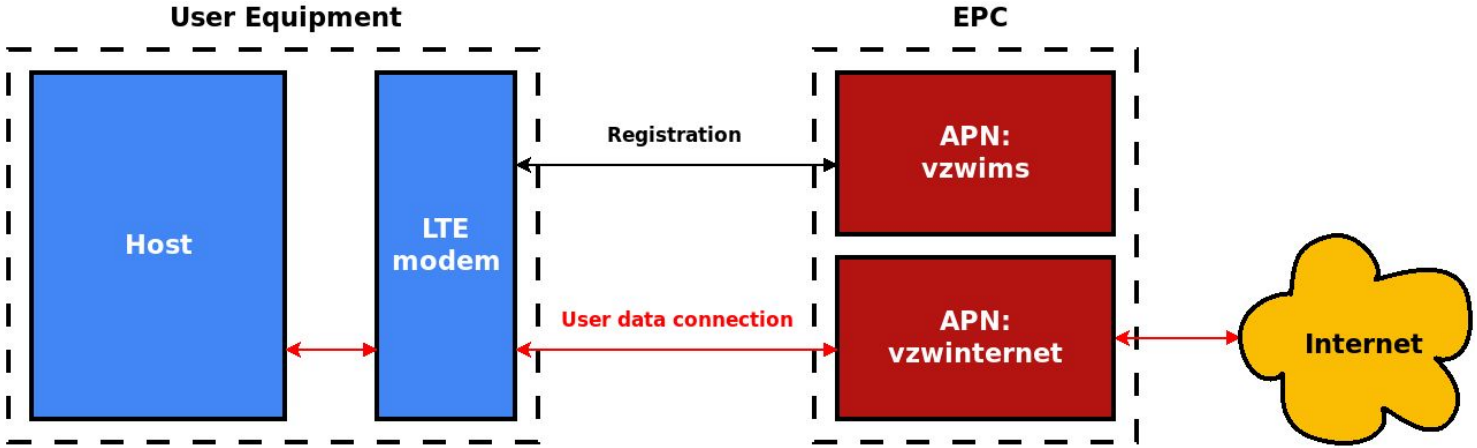
MM 1.0 and the support for 4G

- Ignored initial EPS bearer (attach) settings



MM 1.10 and the 4G attach settings

- Read/write initial EPS bearer settings.
- Show initial EPS bearer status agreed with the network.



What else did we miss in MM 1.0?

- **Profile management** introduced at MM 1.18
  - Finally a clean and consistent way to connect the Verizon class 3 APN.

- Clean differentiation between modems in **bridged mode** vs modems in **router mode**.

Where did we do it well?

- **Multi-PDN support** was included very late, but the existing DBus interfaces made it very easy to implement it without breaking API.

- The codebase allowed adding **Qualcomm SoC support** (e.g. using QRTR and IPA) without major changes and without breaking API.

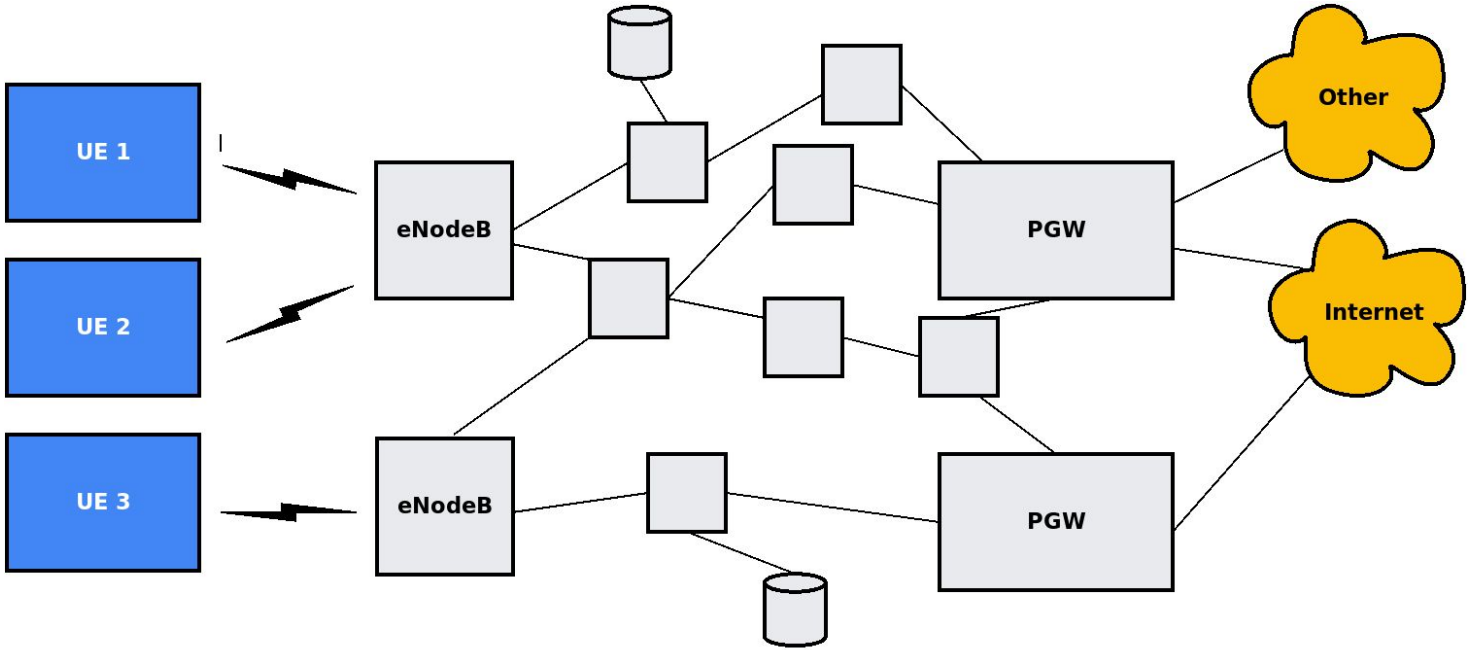
Current 5G support in MM

MM 1.20 and the support for 5G

- **5G SA**
  - `MM_MODEM_ACCESS_TECHNOLOGY_5GNR`
- **5G NSA**
  - `MM_MODEM_ACCESS_TECHNOLOGY_5GNR | MM_MODEM_ACCESS_TECHNOLOGY_LTE`
- Type of 5G (e.g. sub-6 vs mmWave) **not** specified.
- Type of 5G NSA (e.g. EN-DC, NE-DC,...) **not** specified.

Missing 5G features in MM

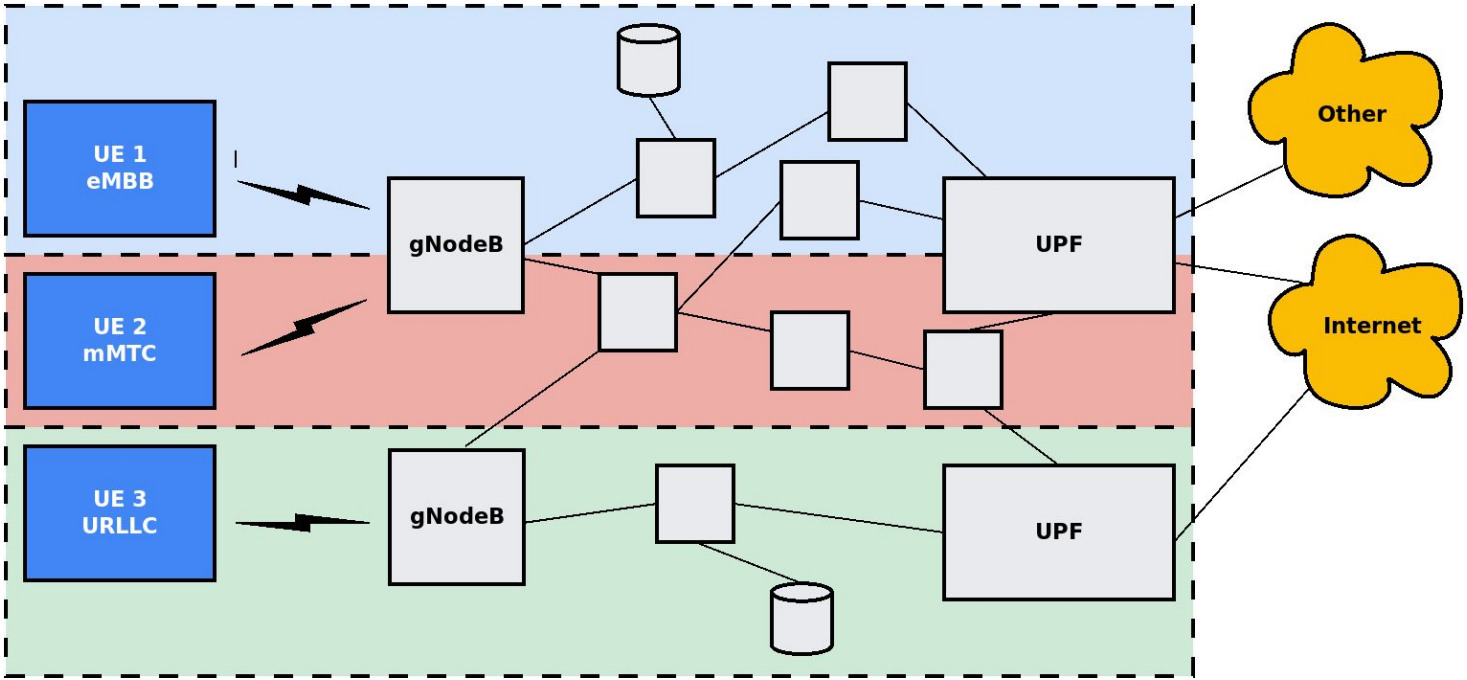
4G networks don’t differentiate UEs



Different UE types in 5G

- Different UEs with different **QoS** parameters
- **eMBB**: enhanced mobile broadband.
- **URLLC**: ultra reliable low latency communications.
- **mMTC**: massive machine-type communications.

5G slicing


UE 1
eMBB
UE 2
mMTC
UE 3
URLLC
gNodeB
gNodeB
UPF
UPF
Other
Internet

5G slicing

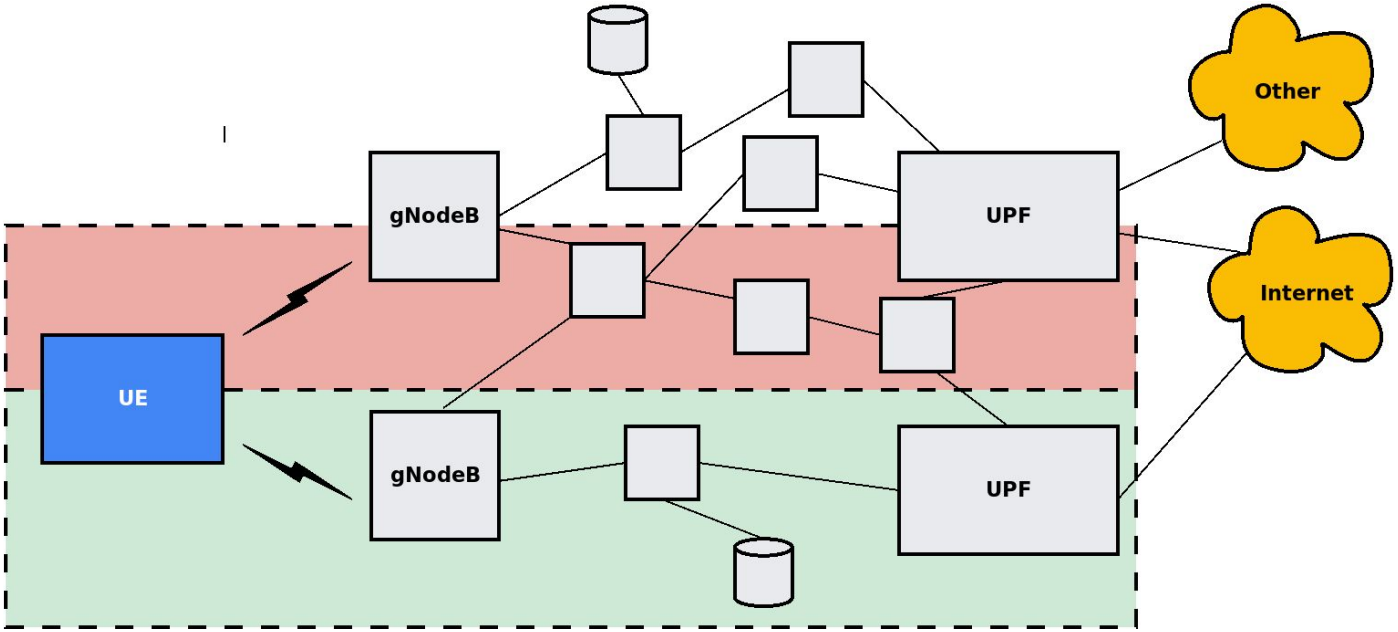
- **A slice is a logical partition of the physical 5G network.**
- Slices are isolated from each other.
- A slice is identified by a **S-NSSAI**
  - Single Network Slice Selection Assistance Information

5G slice selection during registration

- UE requests a **Default Configured NSSAI** during network registration.

- Network reports back
  - **Configured NSSAI**
  - **Allowed NSSAI**
  - Rejected NSSA

Example: separate slices for the same PDN



URSP rules and dynamic 5G slice connections

- The network configures URSP (User Equipment Routing Selection Policy) rules.
- Dynamically select the most suitable network slice for each user device based on QoS, preference, and security needs.
- Great granularity, rules may be per-application, per destination…
- The UE is responsible for applying the rules.

Slicing support in ModemManager

- The host will see **separate multiplexed network interfaces** (e.g. qmapmux0.0, qmapmux0.1) for different QoS flows in different slices.

- Upon a connection attempt to a given PDN, we may want to automatically bring it up over more than one slice.
  - 1 connection attempt, >= 1 multiplexed network interfaces.

- Route/slice selection performed by host.

Other 5G features to consider

- Non-3GPP access support.
- **Non-IP** based 5G connectivity (likely out of scope)

ModemManager for the next 10 years

Google

Remove legacy features

- **Deprecate and remove 3GPP2 support**
  - This includes the full CDMA interface, as well as any logic that checks whether a modem is 3GPP or not.
- Remove POTS support altogether
  - No real code change needed here, but many of the MM 1.0 code rework relied on having support for POTS modems at some point.
- Remove legacy plugins for very old modems

Focus

- **4G and 5G** capable modems should be the primary focus
  - Still support 2G and 3G modems , but on a best effort basis. Only ensure basic functionality works.

- **PCI and USB modems** that expose **network interfaces** should be the primary focus.
  - Still support RS232 and USB modems that rely on AT+PPP, but on a best effort basis. Only ensure basic functionality works.

MM 2.0 DBus API

- It is time for a new 2.0 DBus interface.
- Initially co-exist with the 1.0 DBus interface, even if it means we're duplicating some logic, but completely switch to the new 2.0 interface at some point.
- Not exactly the same process as for the 1.0 DBus interface 13 years ago.

Registration settings

- New APIs to manage settings that the modem will use during the **network selection** and **network attach** procedures.
  - Manual vs automatic settings
  - LTE initial EPS bearer settings
  - EPS mode of operation
  - 5G default network slice settings

Connection management

- Make **profile based connection management** the primary method of operation, instead of giving all settings in the connection attempt.
  - E.g. prefer to say “connect profile 3” instead of “connect with APN=internet, IP type IPv6…”.

- **Multiplexed connections by default**.
  - E.g. creating an ephemeral qmapmux0.0 instead of connecting the parent wwan0 interface.
  - Easy transition to multi-PDN and 5G slicing scenarios.

Dynamic feature interfaces and support

- MM currently exposes all feature interfaces in DBus that are currently supported by the modem, and enables their support by default, even if not needed by the host.
  - E.g. SMS messaging, Voice support.

- We should make the modem:
  - Expose which features are supported.
  - **Enable/disable feature interfaces on demand.**
  - Disabling the interface fully disables everything related to the feature.

Extended wishlist

- AT proxy: allow other applications to perform their own AT control operations over AT ports
- GNSS location out of ModemManager
- Rust (e.g. in binary SMS/QMI/MBIM message parsing)

THANK YOU

https://modemmanager.org