

# Introduction to ModemManager



aleksander.es

QMI **LTE** libmbim  
data-rate kernel  
IoT **qmi\_wwan** MBIM latency  
ModemManager  
AT **Mobile Broadband** M2M  
3GPP **USB** user-space  
libqmi

# Who am I?



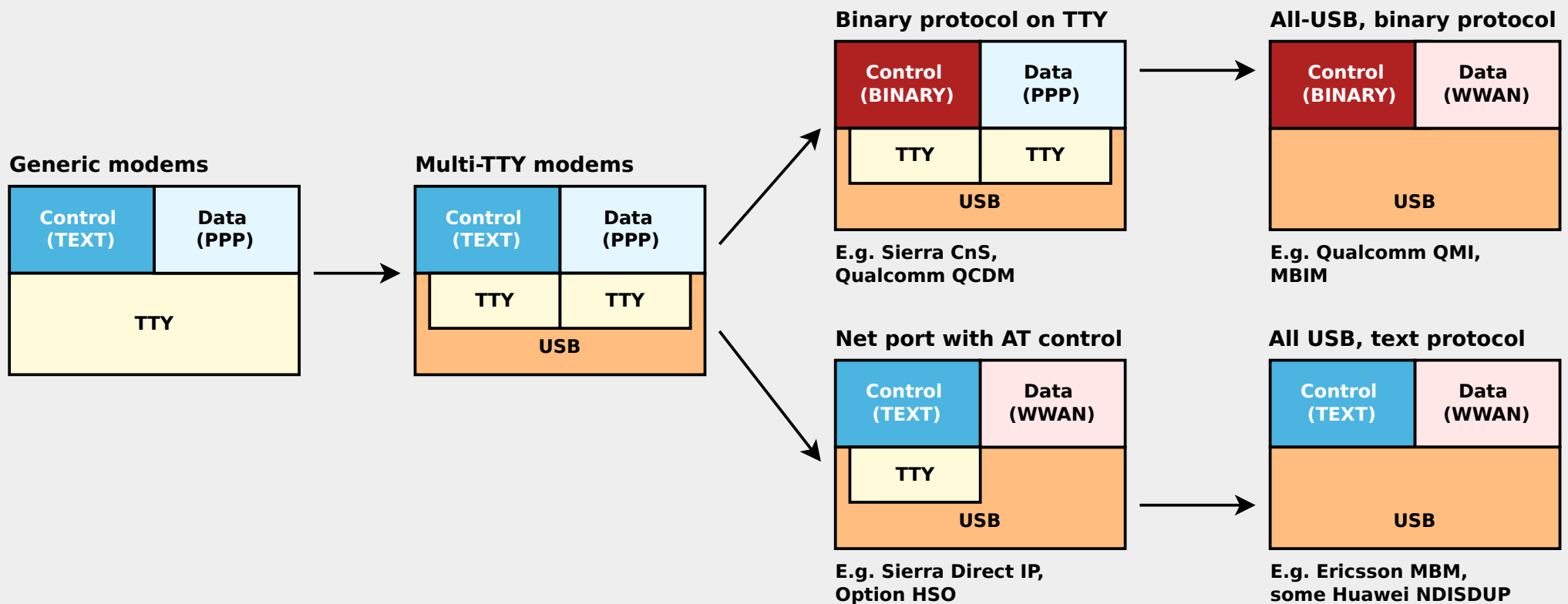
aleksander.es



# WWAN modems have evolved...



aleksander.es



... plus Qualcomm SoCs with integrated modems, newer PCIe based devices (mhi, iosm...)

# Table of contents (1/2)



aleksander.es

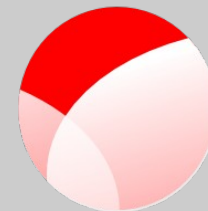
- Where does MM fit in the system
  - From the bottom to the top
  - Userspace system daemons
  - Hierarchy of tools
- MM internals
  - MM in DBus
  - MM interfaces
  - MM state machine(s)
  - MM protocols and plugins
  - libmm-glib
  - mmcli

# Table of contents (2/2)



aleksander.es

- libmbim internals
  - JSON database
  - Backends: chardev file
  - mbim-proxy
  - mbimcli
- libqmi internals
  - JSON database
  - Backends: chardev file, QRTR, MBIM
  - qmi-proxy
  - qmicli
  - qmi-firmware-update



aleksander.es

# Where does MM fit in the system?

# From the bottom to the top



aleksander.es

- **Userspace: user programs**

- GNOME Shell
- GNOME Control Center

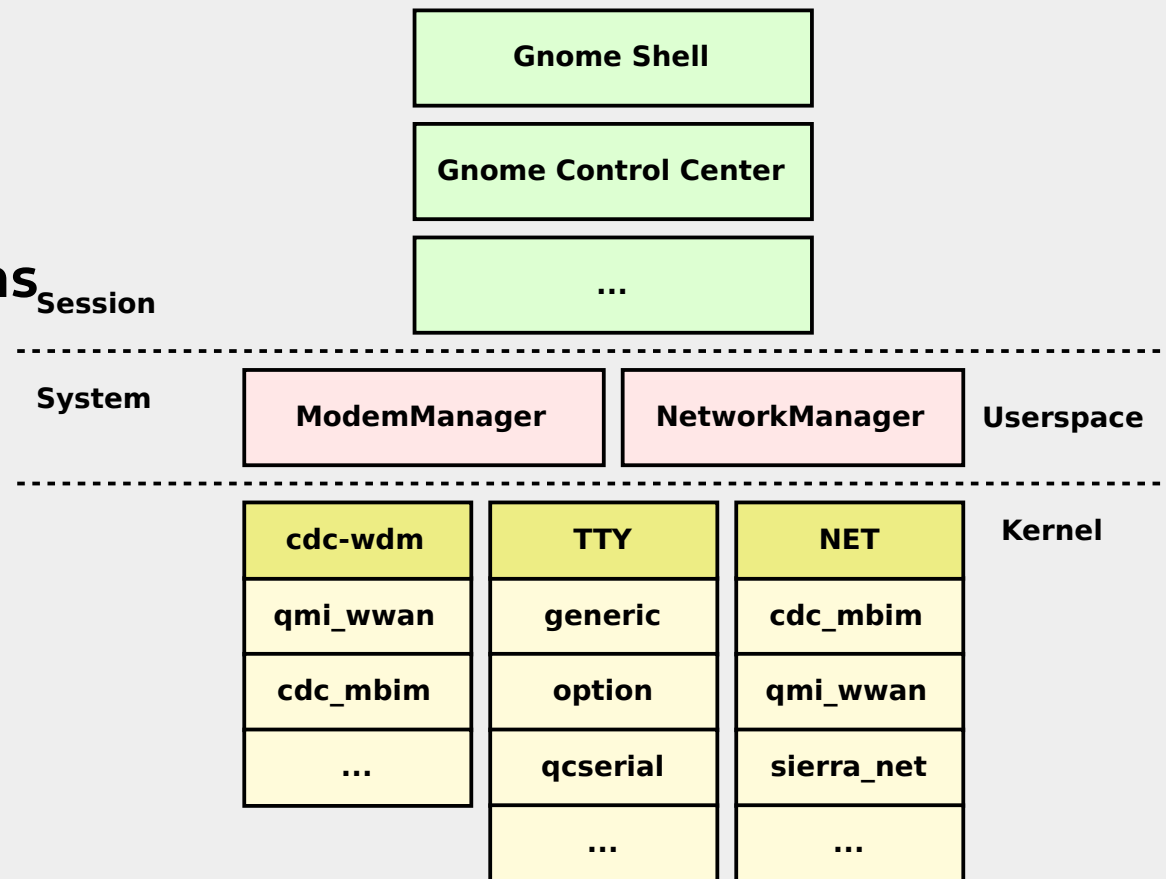
- **Userspace: system daemons**

- NetworkManager
  - pppd, dhclient...

- ModemManager
  - libqmi, libmbim...

- **Kernel**

- Control port drivers
- Data port drivers



# Userspace system daemons



aleksander.es

- **ModemManager**

- Control port(s)
  - SIM unlocking, Network registration, Signal quality, Location information, Connection setup (APN, auth...)
  - Primary & secondary ports
  - External libraries for some binary protocols (e.g. libmbim, libqmi)

- **NetworkManager**

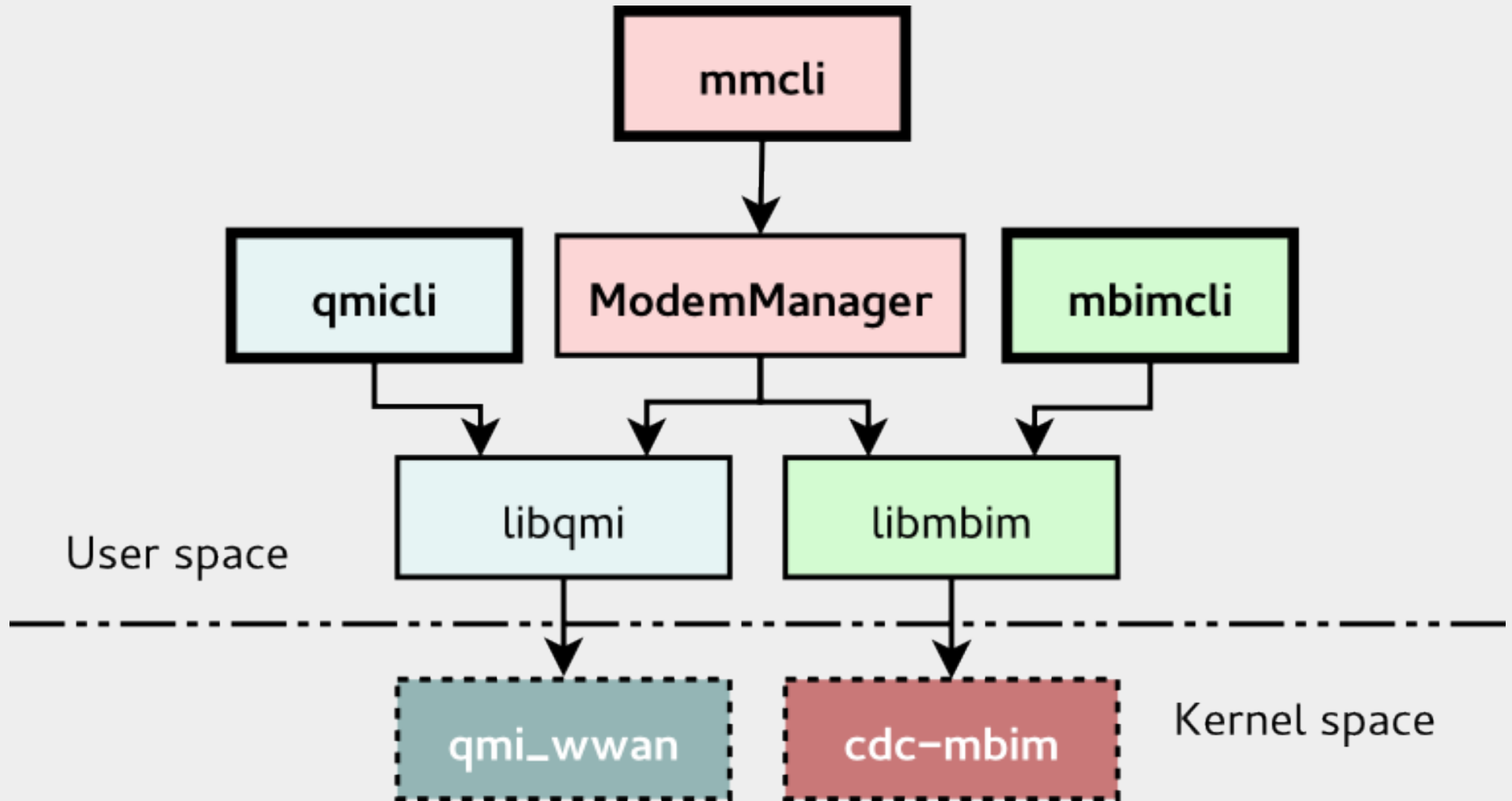
- TTY Data port
  - PPP (launches pppd)
- NET Data port
  - DHCP (launches dhclient or dhcpcd)
  - Static (netlink with kernel)

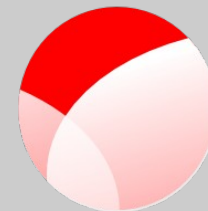


# Hierarchy of tools



aleksander.es





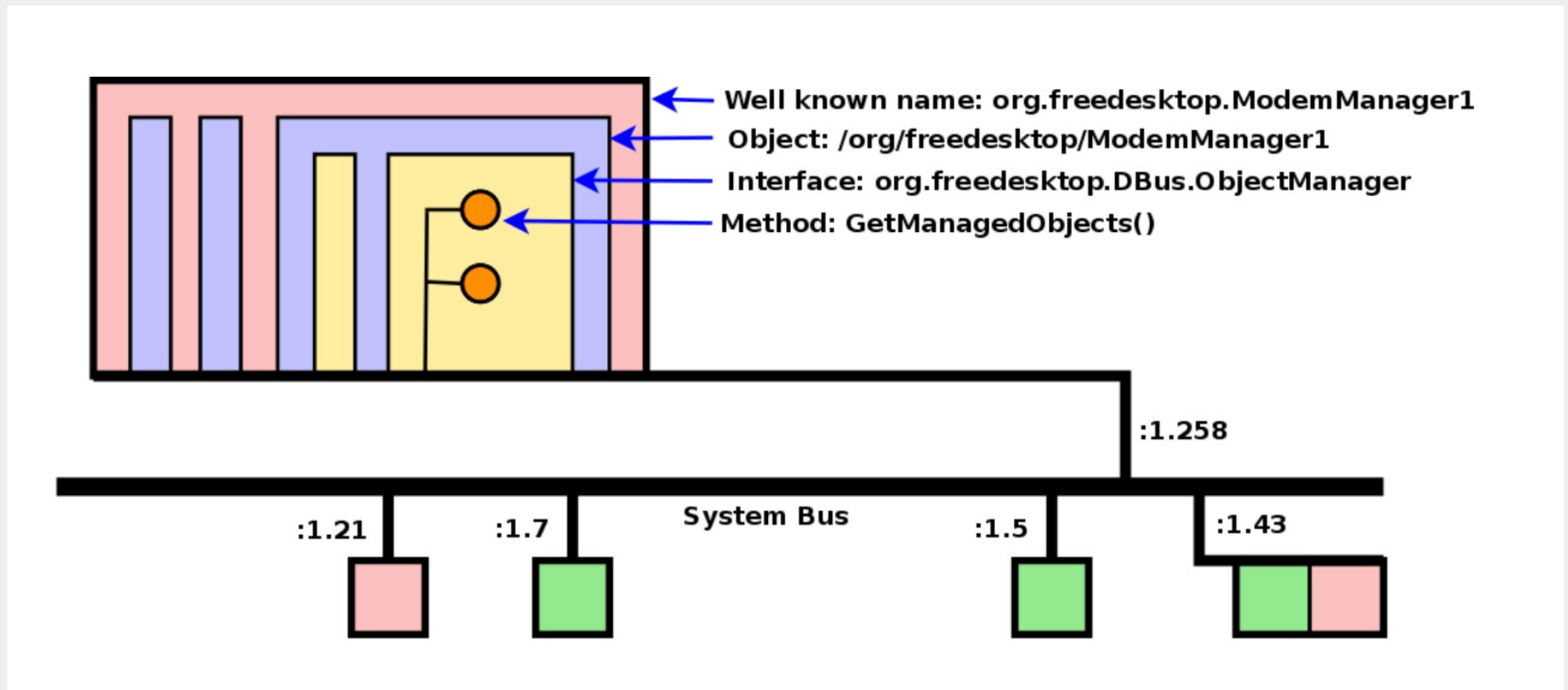
aleksander.es

# MM internals

# MM in DBus



aleksander.es



# MM in DBus



aleksander.es

- Exposed objects:
  - Manager object
  - Modem objects
  - SIM objects
  - Bearer objects
  - Voice call objects
  - SMS objects

# MM in DBus



aleksander.es

The screenshot shows the D-Feet application window titled "System Bus" and "Session Bus". The left pane lists various system services, with **org.freedesktop.ModemManager1** selected. The right pane displays the following details:

- Address:** unix:path=/var/run/dbus/system\_bus\_socket
- Name:** org.freedesktop.ModemManager1
- Unique name:** org.freedesktop.ModemManager1
- Object path:**
  - ↳ /org/freedesktop/ModemManager1
  - ↳ /org/freedesktop/ModemManager1/Modem/0
  - ↳ /org/freedesktop/ModemManager1/SIM/0
  - ↳ /org/freedesktop/ModemManager1/SMS/0
  - ↳ /org/freedesktop/ModemManager1/SMS/1
- Interfaces:**
  - ↳ org.freedesktop.DBus.Introspectable
  - ↳ org.freedesktop.DBus.Peer
  - ↳ org.freedesktop.DBus.Properties
  - ↳ **org.freedesktop.ModemManager1.Sms** (selected)
- Methods:**
  - Send () → ()
  - Store (UInt32 storage) → ()
- Properties:**
  - Array of [Byte] Data (read)
  - Boolean DeliveryReportRequest (read)
  - Int32 Class (read)
  - String DischargeTimestamp (read)
  - String Number (read)
  - String SMSC (read)
  - String Text (read) = BUZON MOVISTAR vie, 09 - 16:23
  - 1 llamada de 648844996
  - 
  - Ha recibido una llamada sin mensaje. Si desea devolver la llamada pulse sobre el numero
  - String Timestamp (read)
  - Struct of (UInt32, Variant) Validity (read)
  - UInt32 DeliveryState (read)
  - UInt32 MessageReference (read)
  - UInt32 PduType (read)
  - UInt32 ServiceCategory (read)

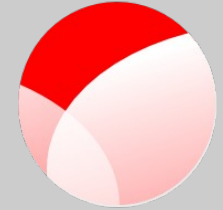
# MM interfaces



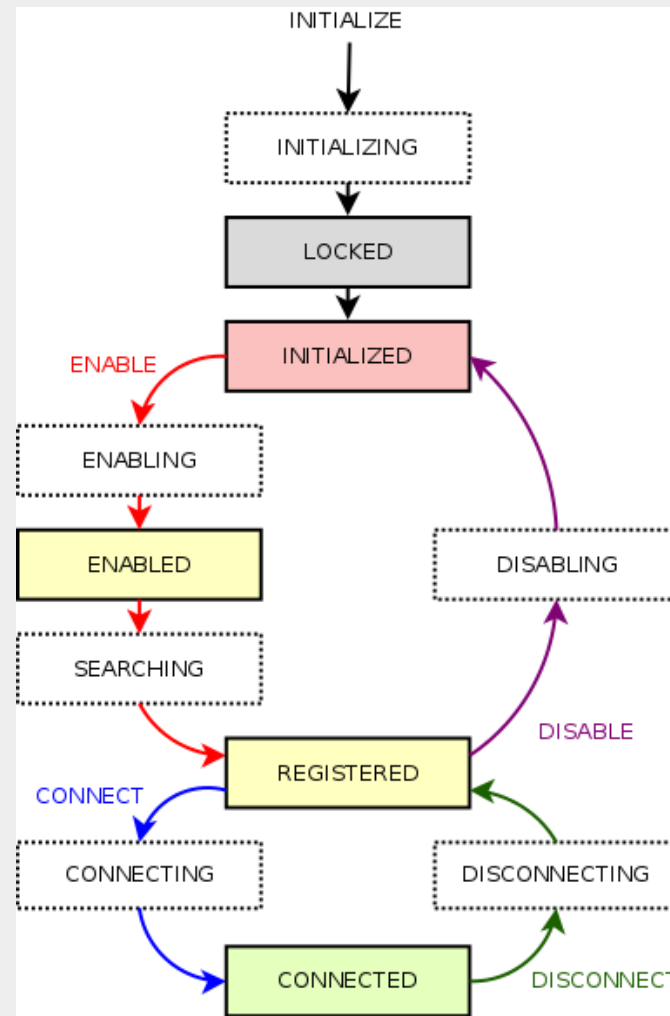
aleksander.es

- Modem object:
  - **Modem**, Modem3gpp, ModemCdma, ModemSignal, ModemLocation, Modem3gppUssd, Modem3gppProfileManager, ModemVoice, ModemMessaging, **ModemFirmware**, ModemSimple...
- SIM object:
  - Sim
- Bearer object:
  - Bearer

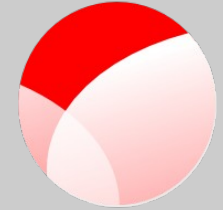
# MM state machine(s)



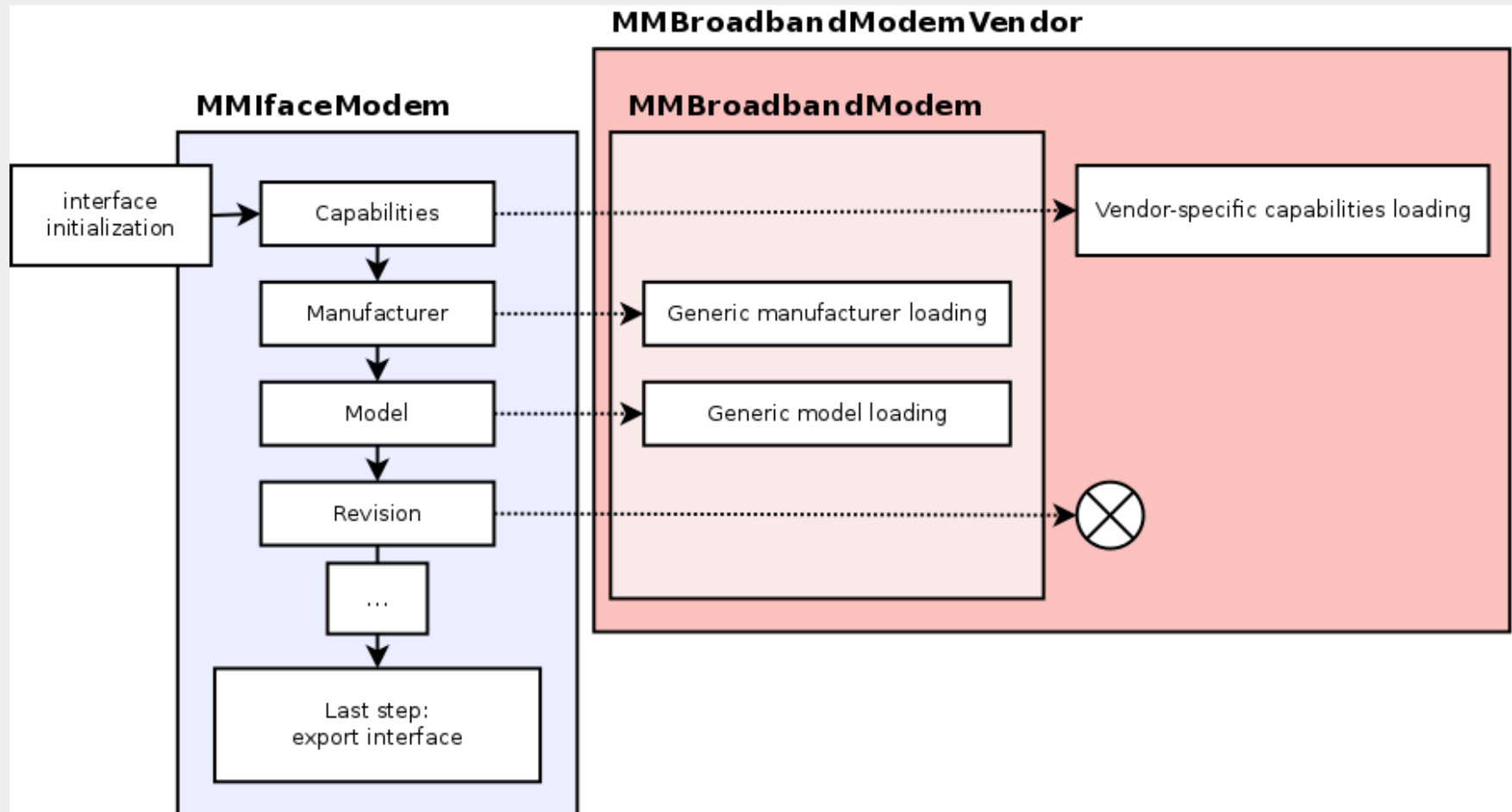
aleksander.es



# MM protocols and plugins



aleksander.es





# libmm-glib



aleksander.es

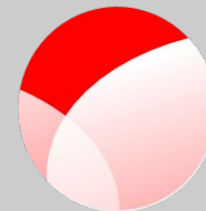
- Client library to interface with the ModemManager daemon
- Glib main loop based
  - Can be integrated in any kind of C/C++ project easily.
- GObject introspection capable
  - Can be integrated in any kind of Python, Javascript (or any other G-I capable setup) project easily
- Stable API across releases
  - API-ABI compatible since 2013 (when MM 1.0 was released).
- Used by NetworkManager, gnome-shell, ....

# mmcli



aleksander.es

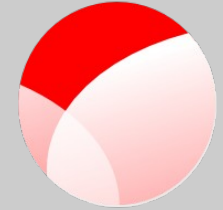
- Command line tool to interface with the ModemManager daemon
  - Originally, just a tester for libmm-glib.
- Easily usable from shell scripts
  - The key tool in the openwrt+netifd integration.
  - Additional JSON and key-value outputs



aleksander.es

# libmbim internals

# JSON database



aleksander.es

```
// *****
{ "name"      : "Emergency Mode",
  "service"   : "Basic Connect",
  "type"      : "Command",
  "since-ex"  : { "set"      : "1.4",
                  "query"   : "1.4",
                  "response" : "1.4",
                  "notification" : "1.8" },
  "query"     : [],
  "set"       : [ { "name"      : "State",
                    "format"   : "guint32",
                    "public-format" : "MbimEmergencyModeState" } ],
  "response"  : [ { "name"      : "State",
                    "format"   : "guint32",
                    "public-format" : "MbimEmergencyModeState" } ],
  "notification" : [ { "name"      : "State",
                       "format"   : "guint32",
                       "public-format" : "MbimEmergencyModeState" } ] },
```

# Backends: chardev



aleksander.es

- libmbim currently only has one single backend: MBIM control ports must be exposed as character devices by the kernel
  - E.g. cdc-wdm ports exposed by the cdc-mbim driver in upstream Linux.

# mbim-proxy



aleksander.es

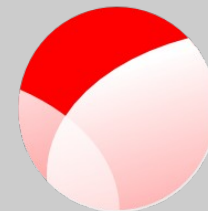
- Allows running multiple programs using the MBIM protocol on the same device at the same time
- The only process that access the MBIM port, all additional programs connect to the proxy
- ModemManager always starts a mbim-proxy
  - So, when MM is running mbimcli actions must always be run through the mbim-proxy.

# mbimcli



aleksander.es

- One “action” for each MBIM request/response.
- May be run at the same time as ModemManager through the mbim-proxy
- Link management options for multi PDN multiplexing setup



aleksander.es

# libqmi internals



# JSON database



aleksander.es

```
{ "name"      : "Set Operating Mode",
  "type"      : "Message",
  "service"   : "DMS",
  "id"        : "0x002E",
  "since"     : "1.0",
  "input"     : [ { "name"      : "Mode",
                    "id"        : "0x01",
                    "type"      : "TLV",
                    "since"     : "1.0",
                    "format"    : "guint8",
                    "public-format" : "QmiDmsOperatingMode" } ],
  "output"    : [ { "common-ref" : "Operation Result" } ] },
```

# Backends: chardev, QRTR, MBIM



aleksander.es

- libqmi has multiple backends:
  - QMI control ports exposed as character devices by the kernel.
    - E.g. cdc-wdm ports exposed by the cdc-mbim driver in upstream Linux.
    - E.g. rpmsg ports in Qualcomm SoCs like the MSM8916 or MSM8974.
  - QRTR nodes in the QRTR bus.
    - E.g. in Qualcomm SoCs like the SDM845
  - MBIM control ports in Qualcomm based modems, exposed as character devices by the kernel.
    - i.e. using QMI over MBIM.

# qmi-proxy



aleksander.es

- Allows running multiple programs using the QMI protocol on the same device at the same time
- The only process that access the QMI port, all additional programs connect to the proxy
- ModemManager always starts a qmi-proxy
  - So, when MM is running qmicli actions must always be run through the qmi-proxy.

# qmicli



aleksander.es

- One “action” for each QMI request/response.
- Additional actions for monitoring QMI indications.
- May be run at the same time as ModemManager through the qmi-proxy
- May be run on MBIM modems or in QRTR-capable SoCs
- Link management options for multi PDN multiplexing setup
- Allocated clients may be re-used over and over
  - (i.e. no need to allocate a new client for every action)

# qmi-firmware-update



aleksander.es

- Firmware upgrade tool for Sierra Wireless Qualcomm-based modems
- QMI DMS Set Firmware Preference
- Sahara, QDL, Firehose
- For more info:
  - `qmi-firmware-update --help-examples`

# Thanks!



aleksander.es

## +Aleksander Morgado

**Freelance GNU/Linux developer**

aleksander@aleksander.es

<https://aleksander.es>