



RIPE Réseaux IP
Européens

Pocket Atlas probe

Reflections on the RIPE Atlas probe on Android

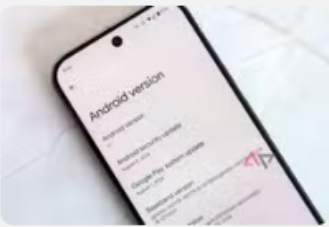
Alex Semenyaka | RIPE 90 | 2025-05-16



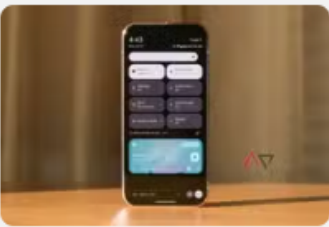
Android's native Linux Terminal app is live in Google's latest update



RELATED



Android could so native Linux tern



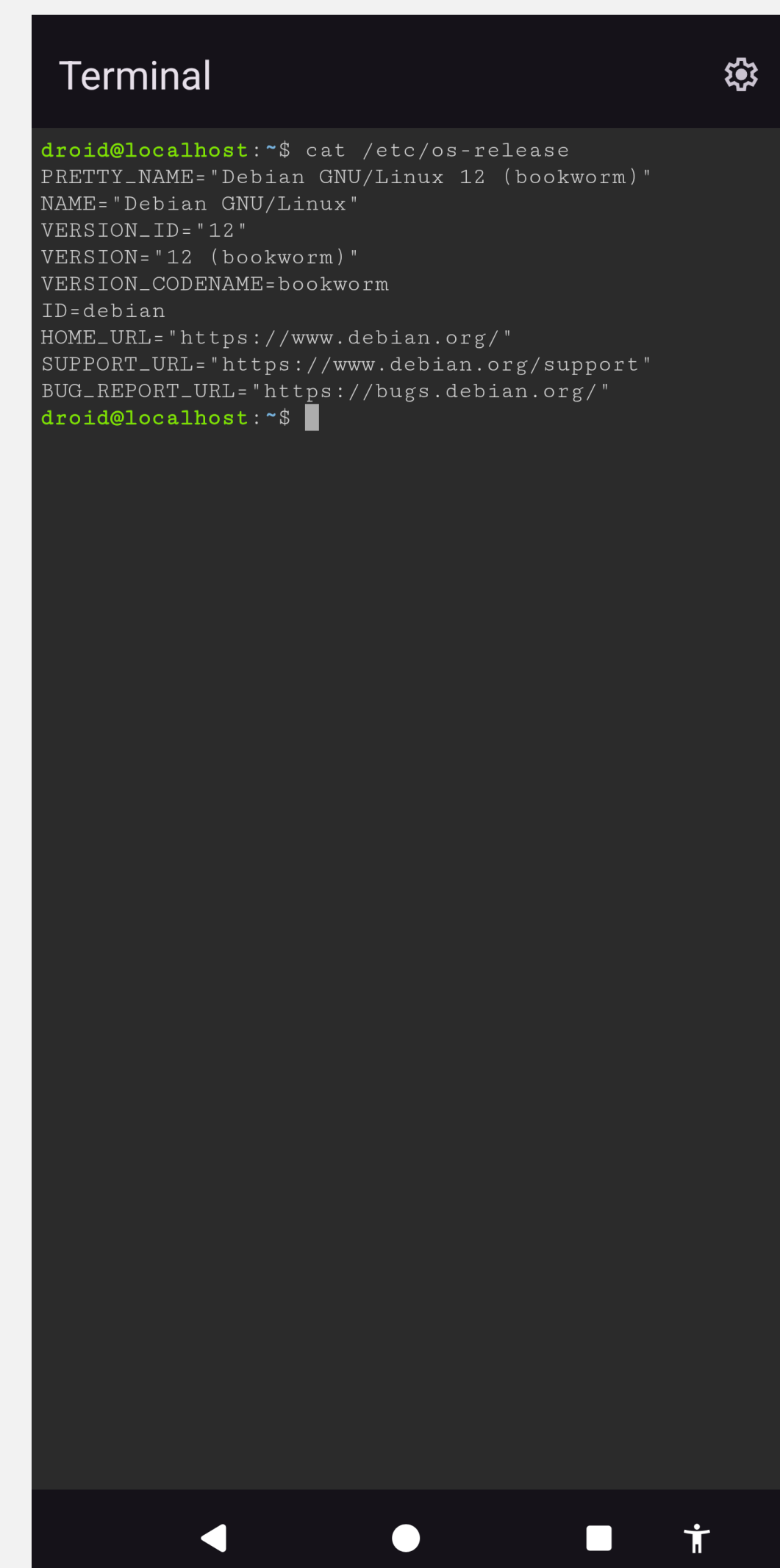
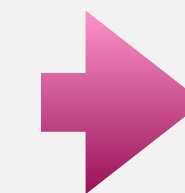
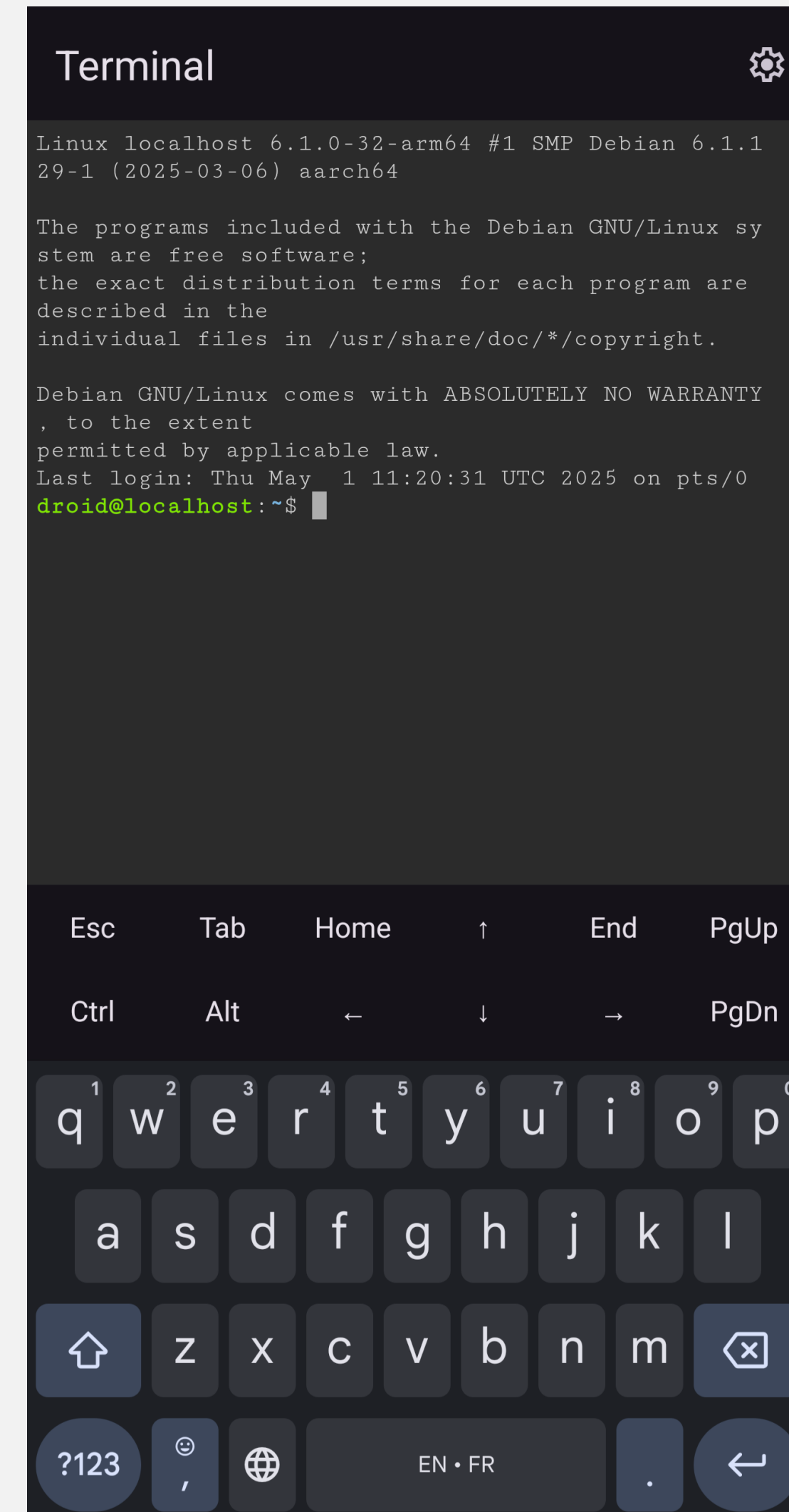
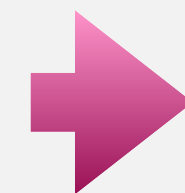
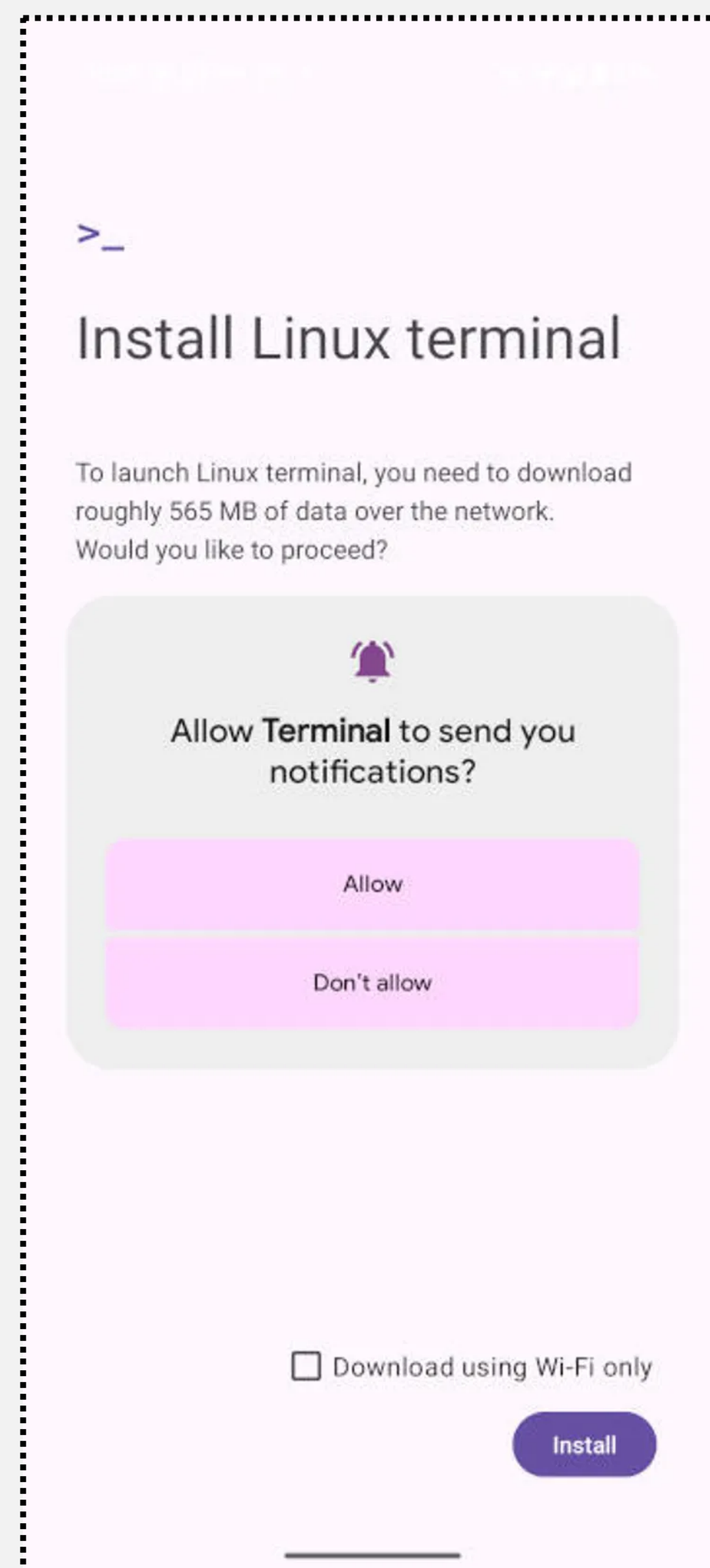
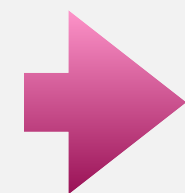
Android 15's fina brings upgrades Gboard, and mor



YouTube's fake r conspiracy deep channels suspen



How it looks like



Under the hood



- Underneath Terminal runs the Android Virtualization Framework (AVF)
- “Android Virtualization Framework (AVF) provides secure and private execution environments”
- “AVF is ideal for security-oriented use cases that require stronger, even formally verified, isolation assurances over those offered by Android’s app sandbox”
 - (<https://source.android.com/docs/core/virtualization>)
- **Now, we can play with raw sockets and such**



Let's add some crazyness

- Hmmmm, and can we install the RIPE Atlas probe there?
- Follow <https://github.com/RIPE-NCC/ripe-atlas-probe-doc/blob/master/manuals/Debian12-binary.en.md>
- Use binary-arm64
- ???
- Yes, we can

It works



Terminal

```
droid@localhost:~$ ps ax | grep atlas
 359 ?      Ss        0:00 /usr/bin/bash /usr/sb
in/ripe-atlas
 462 ?      Ss        0:00 /usr/libexec/ripe-atl
as/measurement/telnetd -b 127.0.0.1 -p 2023 -P /ru
n/ripe-atlas/pids/telnetd-port2023-pid.vol
 464 ?      Ss        0:00 /usr/libexec/ripe-atl
as/measurement/perd -c /var/spool/ripe-atlas/crons
/main -A 9801 -P /run/ripe-atlas/pids/perd-main.pi
d.vol
 466 ?      Ss        0:00 /usr/libexec/ripe-atl
as/measurement/eperd -c /var/spool/ripe-atlas/cron
s/7 -A 9807 -P /run/ripe-atlas/pids/perd-7.pid.vol
-O /var/spool/ripe-atlas/data/new/7 -i 7
 467 ?      S        0:00 /usr/bin/bash /usr/sb
in/ripe-atlas
 469 ?      S        0:00 /usr/libexec/ripe-atl
as/measurement/oooqd /var/spool/ripe-atlas/crons/o
neoff -A 9809 -P /run/ripe-atlas/pids/oooqd.pid.vo
l -i 9
 518 ?      S        0:00 /usr/bin/ssh -i /etc/
ripe-atlas/probe_key -o ServerAliveInterval 60 -o
StrictHostKeyChecking yes -o UserKnownHostsFile /r
un/ripe-atlas/status/known_hosts -R 2023:127.0.0.1
:2023 -L 8080:127.0.0.1:8080 -p 443 atlas@ctr-dub-
sw01.atlas.prod.ripe.net KEEP
 1046 pts/0  S+        0:00 grep atlas
droid@localhost:~$
```

Probe 1010739

This probe is **Connected** (since 2025-05-13 15:53:55 UTC)

OVERVIEW

NETWORK

RESULTS

MANAGE

Probe Information ⓘ

ID

1010739

DESCRIPTION

ARCHITECTURE

Software

FIRMWARE VERSION

5110

SHARED PUBLICLY

Yes

USER TAGS

SYSTEM TAGS

system: Auto GEOIP city

system: IPv4 Capable

system: IPv4 RFC1918

system: Software

ASN V4

2121

(RIPE-MEETING-AS - Reseaux IP Europeens Network Coordination Centre (RIPE NCC))

COUNTRY

Portugal (PT)

FIRST CONNECTED

2025-04-10 17:32:14 (UTC)

LAST CONNECTED

2025-05-13 16:13:26 (UTC)

TOTAL UPTIME

0d 6h 3m 2s

ROUTER TYPE ⓘ

N/A

MAC

N/A

BANDWIDTH ALLOWED ⓘ

N/A

DNS ⓘ

Off

Location ⓘ

Traffic ⓘ

No Traffic Data Available.

Current Measurements

There are currently 470 measurements running on this probe.

See Measurements ↗

Well-known Targets ⓘ

facebook.com (v6)

Deviation: 0%

Avg: N/A

Loss: -1%

google.com (v6)

Deviation: 0%

Avg: N/A

Loss: -1%

wikipedia.org (v6)

Deviation: 0%

Avg: N/A

Loss: -1%



Do we support it?

Do you support mobile measurements?

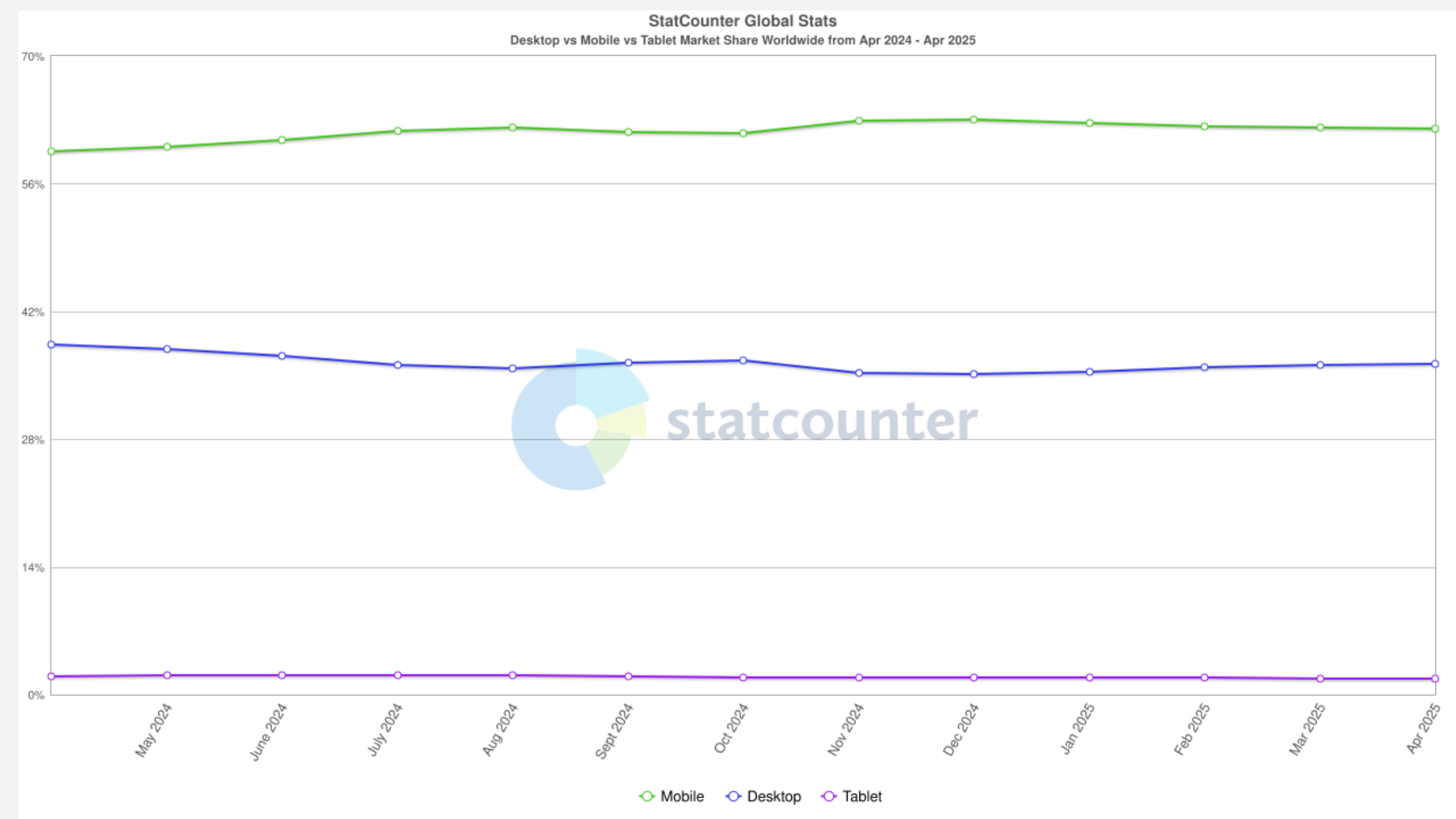
"Mobile measurements" can mean two separate topics:

- The ability to run a probe on a mobile device, such as an app (in the Android or Apple stores). This is not supported for several reasons:
 - Doing so would considerably affect the power consumption of the device as it is always-on and always active (i.e. measurements are constantly happening).
 - Permissions available to such apps are unlikely to be sufficient to support expected probe behaviour.
 - The resulting data would be hard to interpret as devices tend to switch between wifi and mobile connections, or even 3G/4G/5G, resulting in latency, IP and other changes.
- The ability to measure using mobile networks, in other words to host a probe on a wireless connection:
 - This is already possible: several hosts run probes connected via Starlink or some "fixed wireless" connections.
 - In these cases the probe is a regular client device behind the router; one that requires a wired connection to the (home) router.

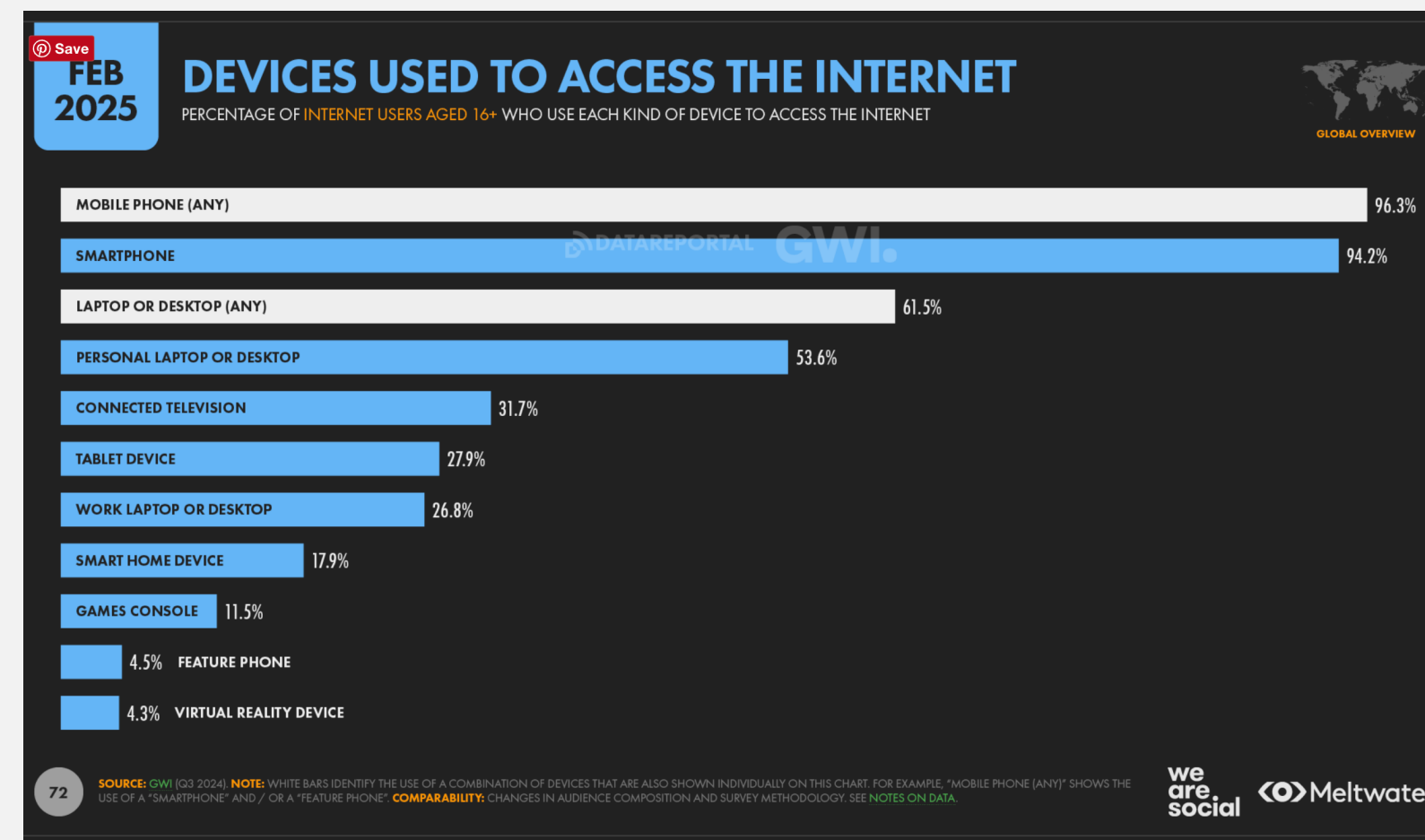
(<https://atlas.ripe.net/docs/faq/technical-details#do-you-support-mobile-measurements>)

= NO

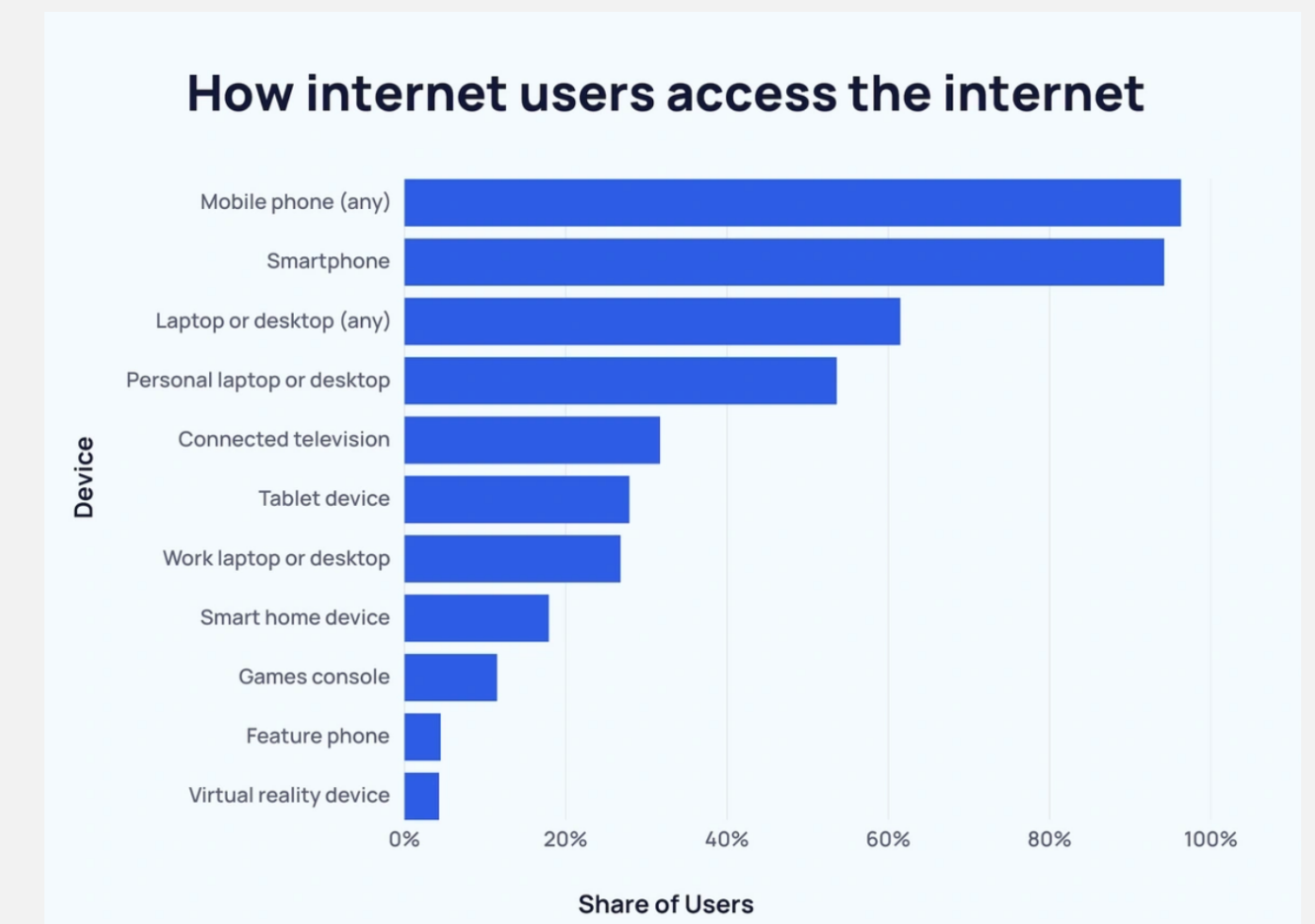
However...



<https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>



<https://datareportal.com/reports/digital-2025-sub-section-device-trends>



<https://explodingtopics.com/blog/mobile-internet-traffic>

We have moved into the era of Mobility First.
(Some time ago)



And what does it mean?

- We will have more and more such installations
 - <https://www.ripe.net/media/documents/116-2023-06-Slides-Rotterdam2-Louis-Petiniaud.pdf>
- However, on the whole, we find ourselves in a lose-lose situation
 - We are not using the potential of mobile devices
 - We do not take advantage of the opportunity to enrich our data
 - GPS data are not available in the Google Linux Terminal
 - Researchers who need these capabilities will use under-the-table options, as I have shown
 - ...which, without support and proper tagging, will lead to data pollution



Time for some dubious fabrications

Perhaps it is time to reconsider our approach?

This is not a *statement*, but a *question*.

Maybe **we don't** need to do this.

But the decision requires an understanding of all aspects.



Well, it is not that simple

- Battery draining
 - It is not necessary to run the mobile probe all the time
 - Several “mobile probes” in one location, active at different times, in total still can do a reasonable job
 - There are different scheduling models that can be implemented to be chosen from
 - Active all the time
 - The app is only active when the screen is active (the user is using the phone)
 - The app is only active when the phone is charging
 - The app is active once an hour
 - The app is only active when open
 - It is not necessary to maintain the permanent connection with controllers



Well, it is not that simple

- Data interpretation
 - The probe would explicitly tag the interface they use at the moment (cellular, Wi-Fi, or other)
 - *Maybe* these probes should not be selected for measurements without a specific and explicit request
 - *Maybe* not all probes were created equal



Well, it is NOT that simple

- Android restrictions
 - No silver bullet 😞 - but something can be done
- Using bundled apps
 - We used to have it in the version 1: `"result": "traceroute to 2001:7fd::1 (2001:7fd::1), 32 hops max, 16 byte packets NEWLINE 1 2001:67c:2e8:13::2 1.907 ms 1.909 ms 1.945 ms NEWLINE 2 2001:7f8:1::a502:5152:1 14.162 ms 2.026 ms 1.872 ms NEWLINE 3 2001:7fd::1 1.876 ms 1.872 ms 1.878 ms NEWLINE"`
- Using AVF backend combined with non-virtualized frontend
- Using third-party solutions based on NDK
 - <https://github.com/wangjing53406/traceroute-android-executable> - UDP traceroute
 - <https://github.com/mahmudur85/android-traceroute> - ICMP traceroute



Some conclusions

- Going this way has both pros and cons
- The **anonymity** of such probes/measurements **should be** additionally provided and controlled
- The variability of measurements will be much less (especially in the case of ping and traceroute)
- We should not expect a steady stream of measurements from a single sample
- Data can be enriched with information that is not available anywhere else: GPS, LACs, etc
- It becomes possible to co-operate with other parties
- Potentially, much larger coverage can be obtained
- We will need to think about new, mobile-specific principles of probe operation, measurement and interpretation of results
- It all costs money
 - Both to develop and then to store more data
- Maybe a crowdsourced approach could help



Questions, feedback & **dialogue**





RIPE Réseaux IP
Européens

THANK YOU