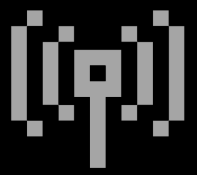
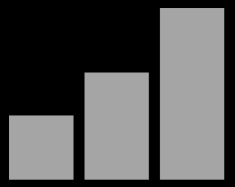
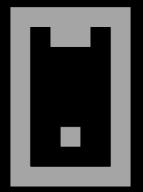
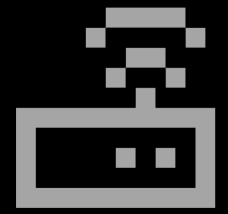



What your phone won't tell you

Uncovering fake base stations on iOS devices

Lukas Arnold

37C3 UNLOCKED

IMSI Catchers Everywhere

Die Bundesregierung

"German Federal Police used IMSI Catchers in at least 38 instances in 2022"

The Register

Secret Service, ICE break the law over and over with fake cell tower spying

Investigations 'at risk' from sloppy surveillance uncovered by audit probe

MOTHERBOARD
TECH BY VICE

With $20 of Gear from Amazon, Nearly Anyone Can Make This IMSI-Catcher in 30 Minutes

whoami

- Hi, I'm Lukas 👋
- Master's Student @ TU Darmstadt
- Student Researcher @ SEEMOO

TECHNISCHE UNIVERSITÄT DARMSTADT

SEEMOO

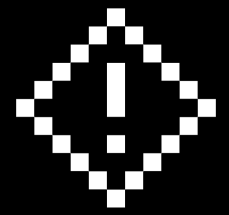
SECURE MOBILE NETWORKING

emergenCITY

ATHENE

Nationales Forschungszentrum für angewandte Cybersicherheit

Cellular Security



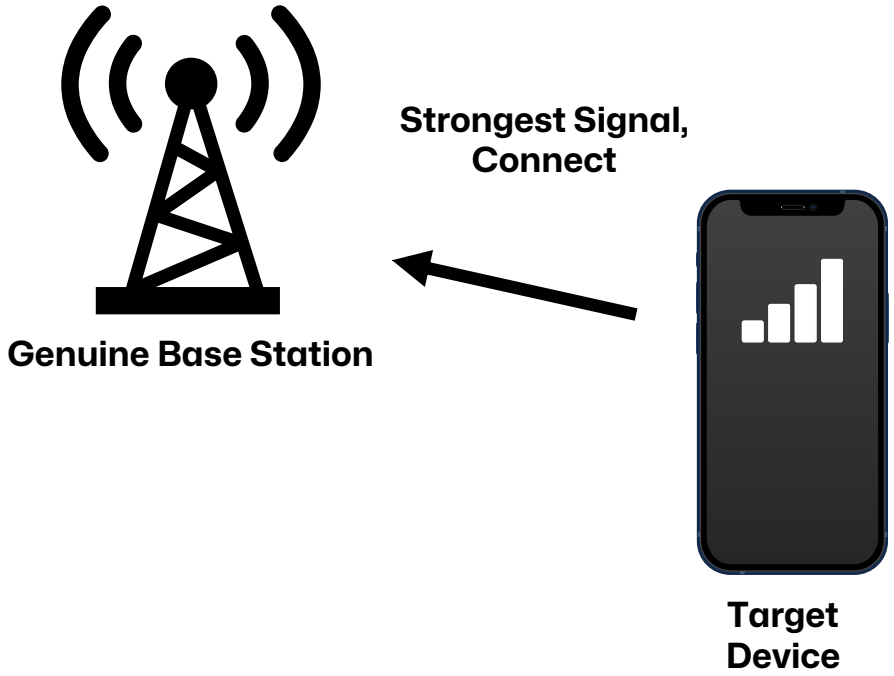
Rogue Base Stations

Adversaries can

- **block** over-the-air signals
- **intercept** over-the-air signals
- **modify** over-the-air signals

They cannot

- physically access the target
- infect the target beforehand



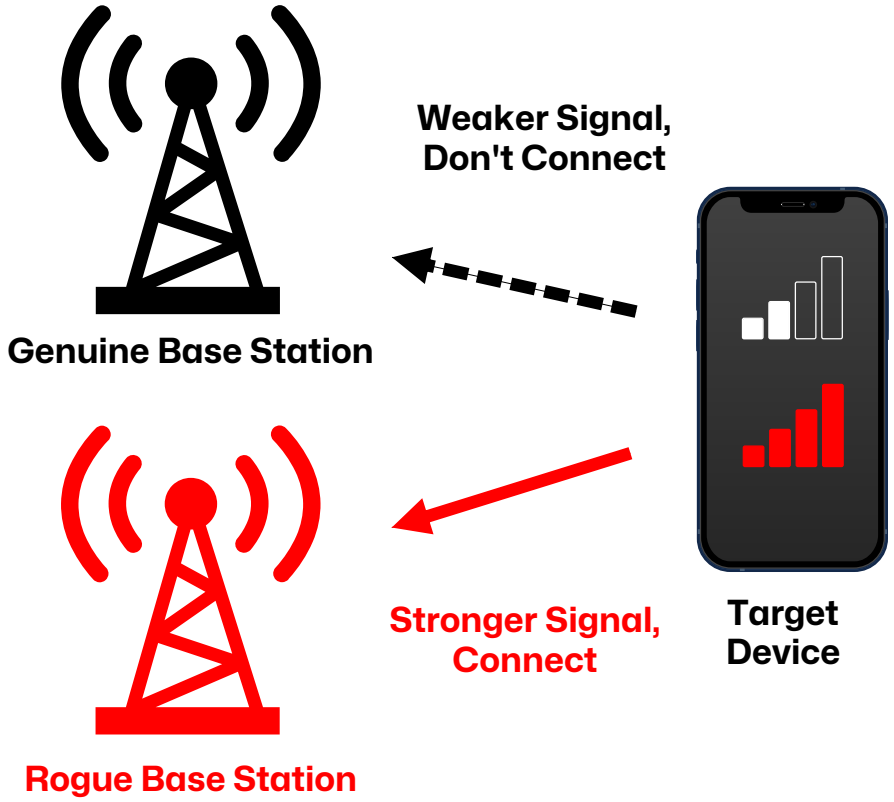
Rogue Base Stations

Adversaries can

- **block** over-the-air signals
- **intercept** over-the-air signals
- **modify** over-the-air signals

They cannot

- physically access the target
- infect the target beforehand



Known Attack Vectors (1)

~~4G~~ ⬇️
~~3G~~ ⬇️
2G ⬅️

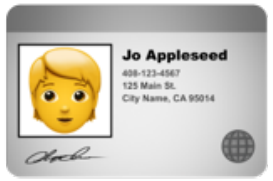
Missing Authentication & Integrity Checks

- Only UE auth. in 2G
- Flaws also in 3G & 4G
- Traffic Interception & Manipulation

Downgrade Attacks

- Smartphones still support 2G
- Jam newer frequency bands

Known Attack Vectors (2)



Identity Information Leakage

- IMSI / IMEI in 2G, 3G, 4G
- SUCI / SUPI in 5G
- Location

Firmware and Mobile Operating System RCE

- Basebands as a zero-click attack surface

Attacker Capabilities

Regular Attackers

- Reasonable budget
- Open-source software & public knowledge

State-sponsored Attackers

- Unlimited budget
- Collaborate with network operators in jurisdiction

Mitigations

App-based

- App monitors base-band parameters
- Instant warnings

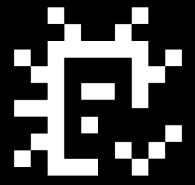
Sensor-based

- Dedicated sensors
- Collaborate

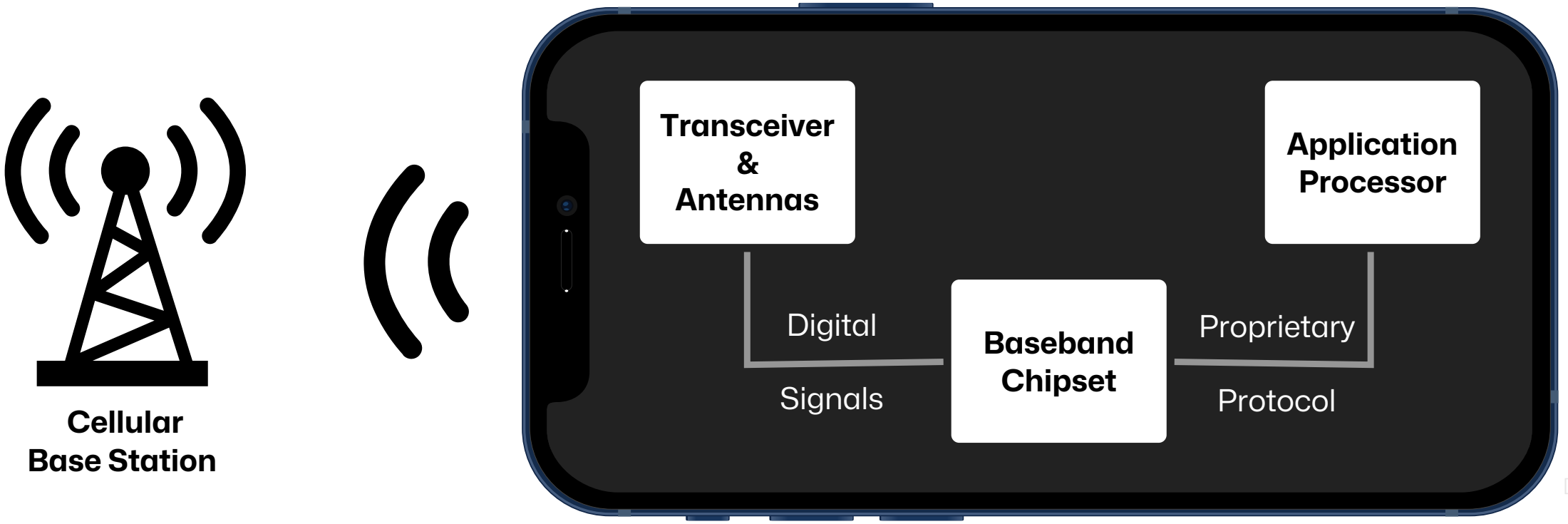
Network-based

- Utilize data of base stations & core network

Tampering with iPhone Basebands



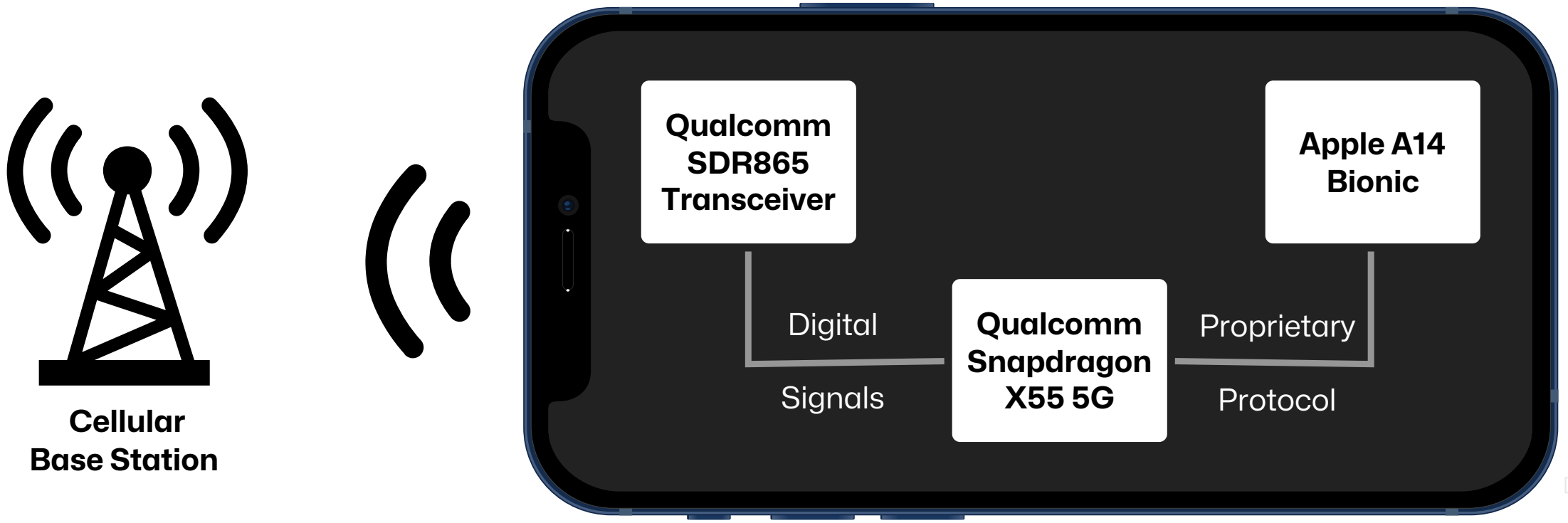
What is the baseband?



[1]

Baseband: Implements complex **cellular protocol stack**

Baseband in the iPhone 12



[1]

Baseband: Implements complex **cellular protocol stack**

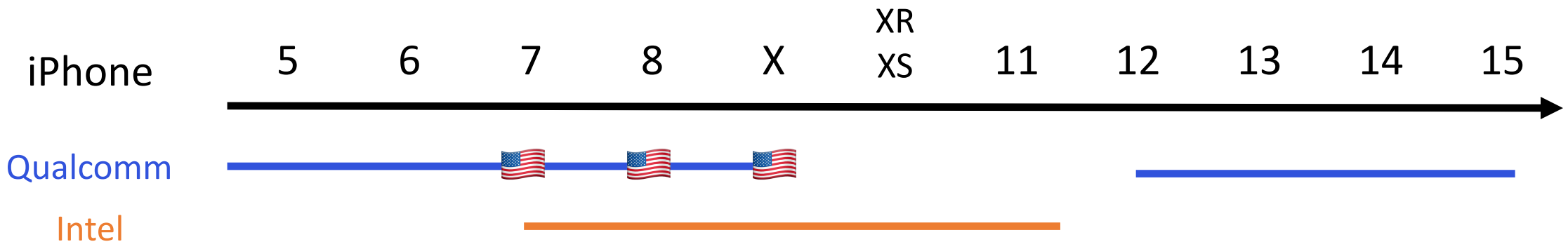
iPhone Basebands

Qualcomm

- Protocol: **Qualcomm MSM Interface (QMI)**
- Focus of my Bachelor's thesis

intel

- Protocol: **Apple Remote Invocation (ARI)**
- Reverse-engineered by Tobias Kröll (ARIstoteles dissector)

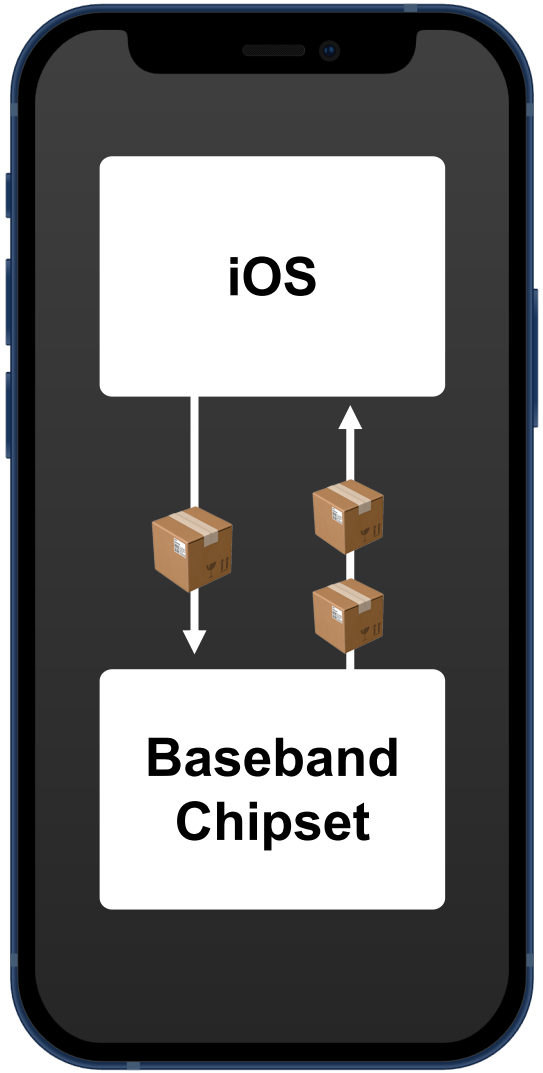


Baseband Interface Protocols

- Binary protocols with packet-like structures
- Packets consist of
  - Headers
    - Type (Request, Response, Indication)
    - Service ID
    - Message ID
  - Data
    - Type-Length-Value's (TLVs)

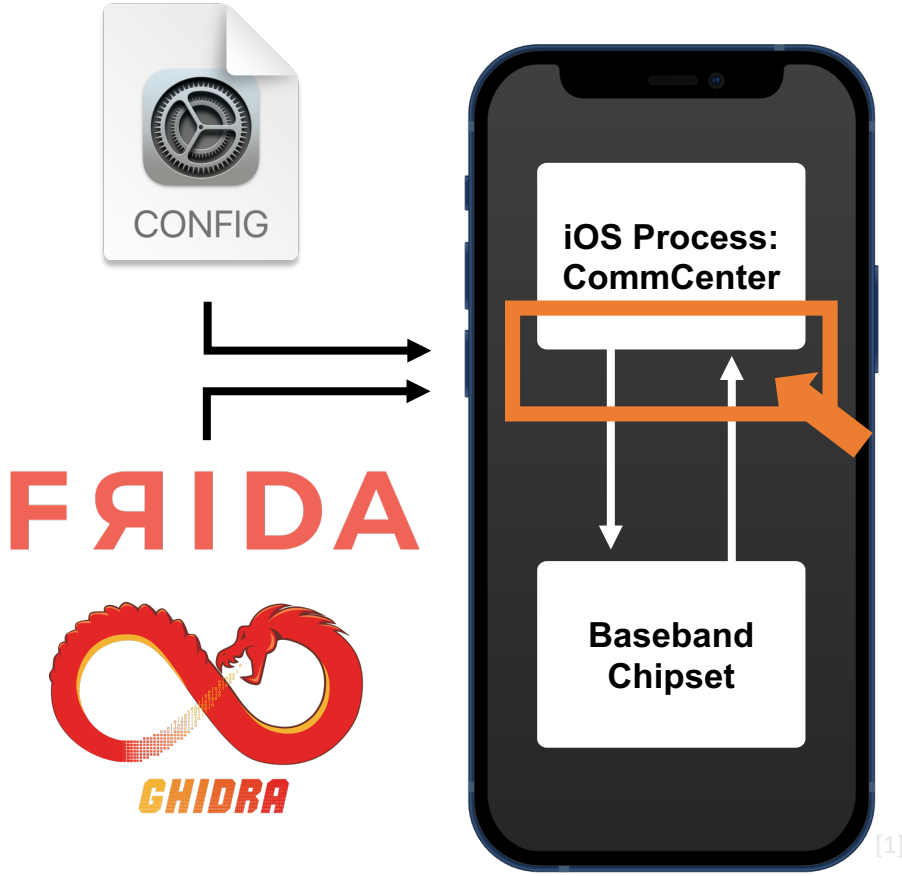
ARI STOTELES

libqmi



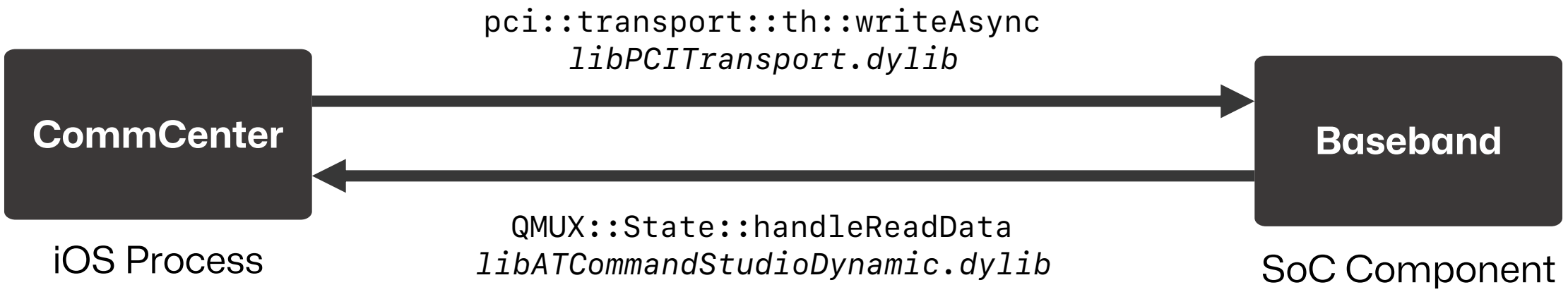
iOS Baseband Architecture

- **CommCenter** handles all things related to cellular communication
  - iOS System Process
  - Baseband Communication
- Examine the iOS RX / TX architecture for BB packets
  - Apple Baseband Debug Profile
  - RE Tools: Frida & Ghidra
  - jiska: Fuzzing the phone in the iPhone



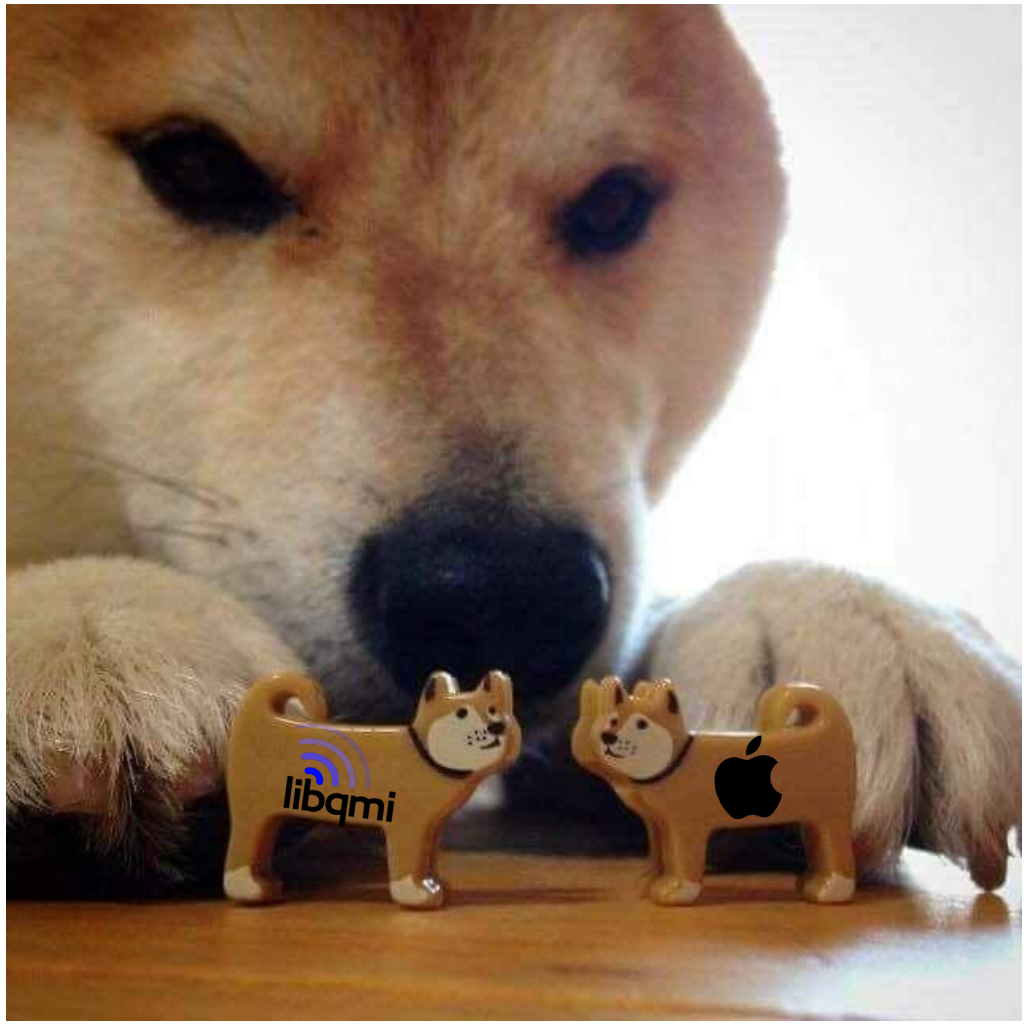
[1]

iOS Baseband Architecture

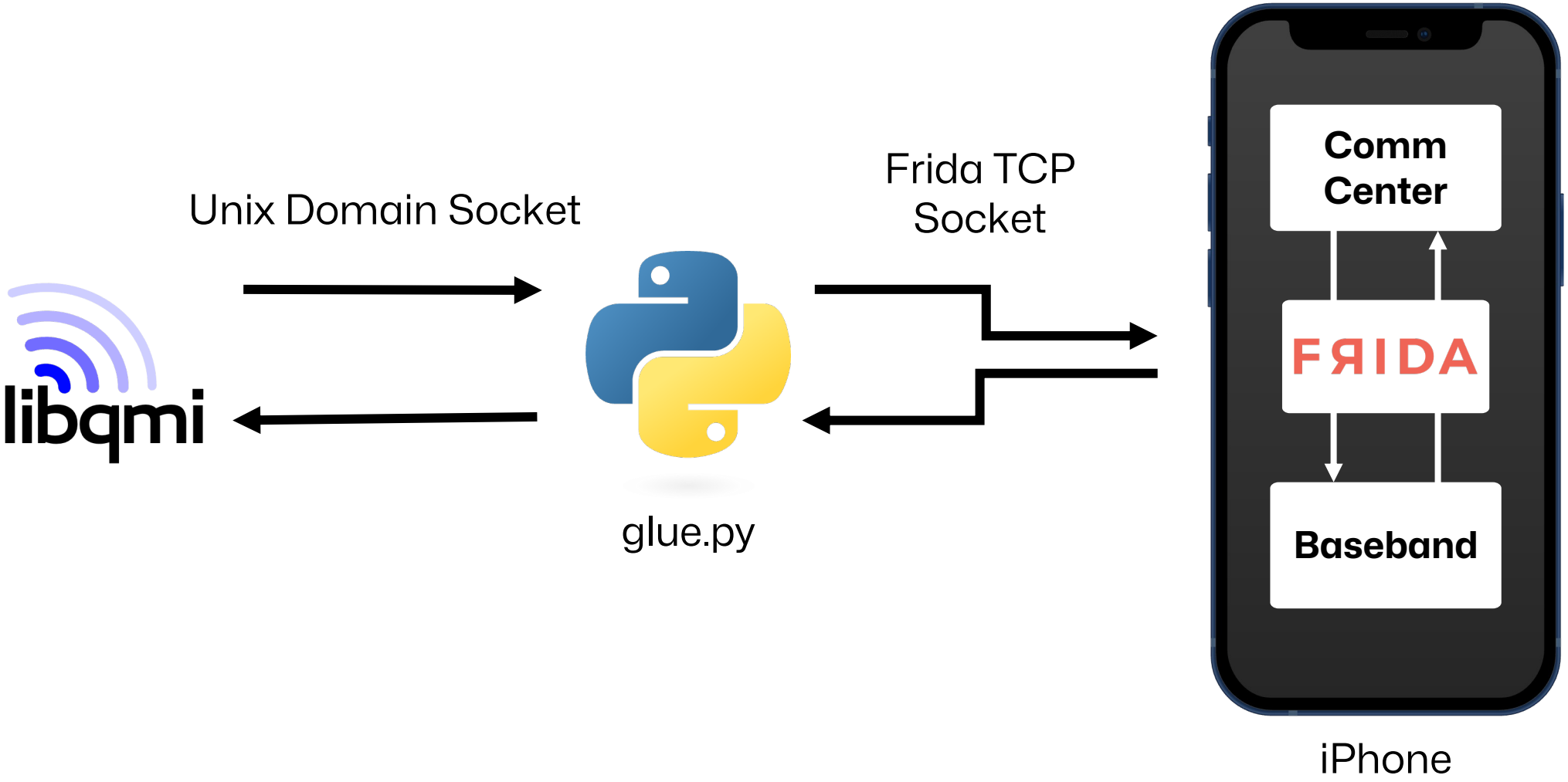


iOS & Baseband communicate over
Peripheral Component Interconnect (PCI)

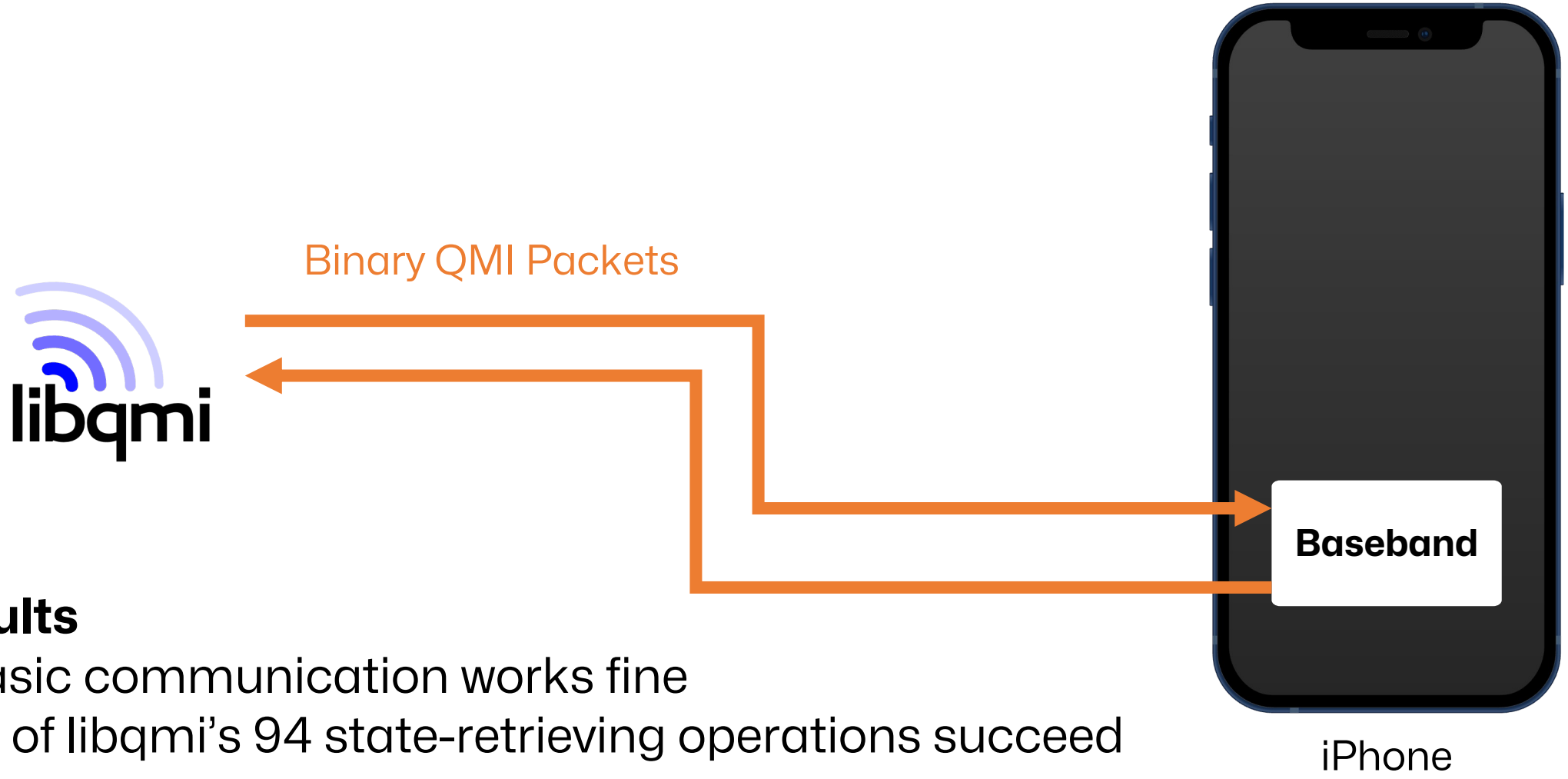
Connecting What Belongs Together



Connecting What Belongs Together



Connecting What Belongs Together



Results

- Basic communication works fine
- 22 of libqmi's 94 state-retrieving operations succeed

Evaluating QMI Services

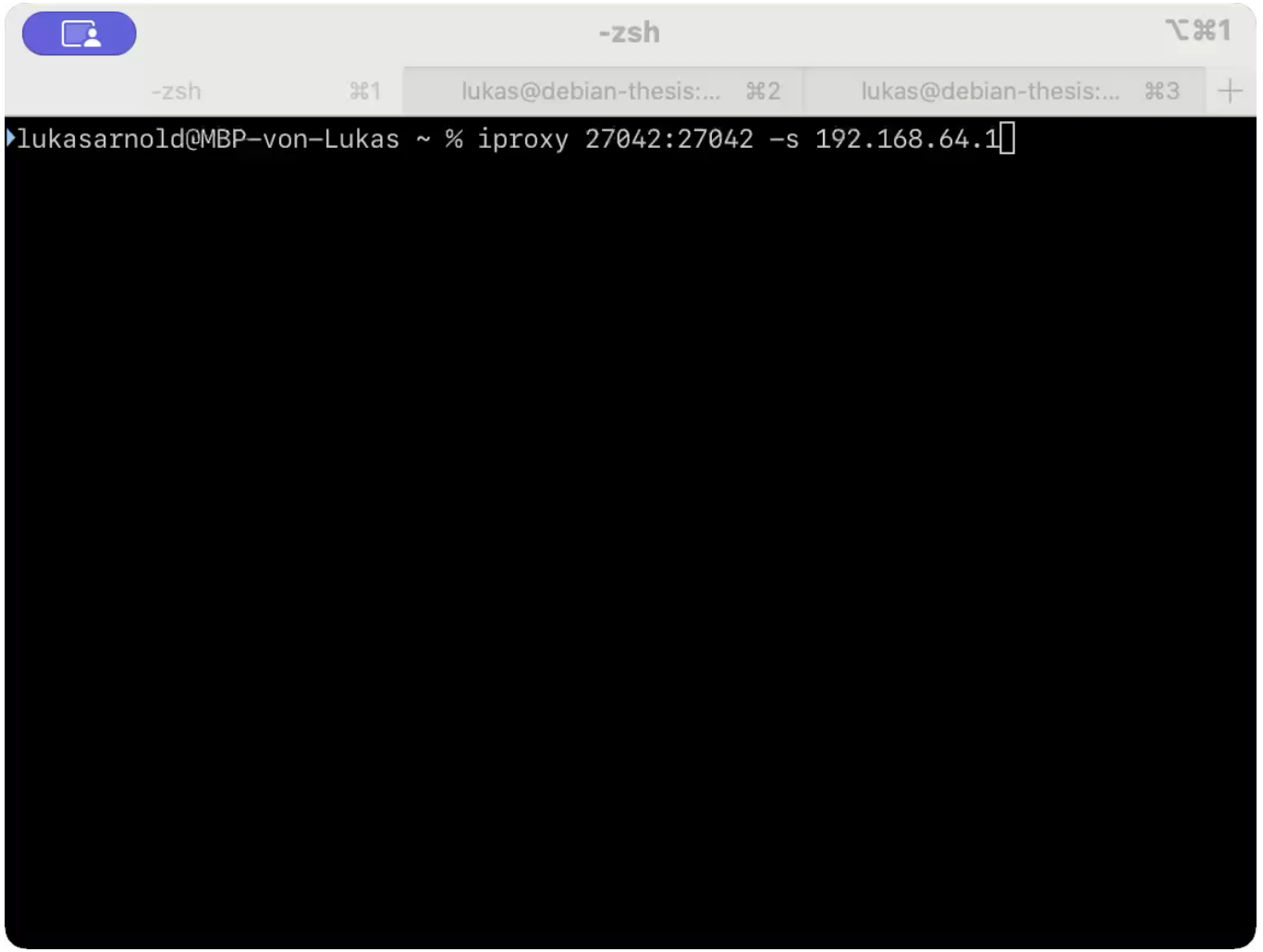
- QMI Services bundle similar QMI messages
- List of QMI services = Overview of baseband capabilities

Snapdragon X55
- Query using libqmi client
- 39 Services

iOS 17.1
- Extract from firmware
- 31 Services

Baseband provides more services than iOS uses

Interacting with the Baseband



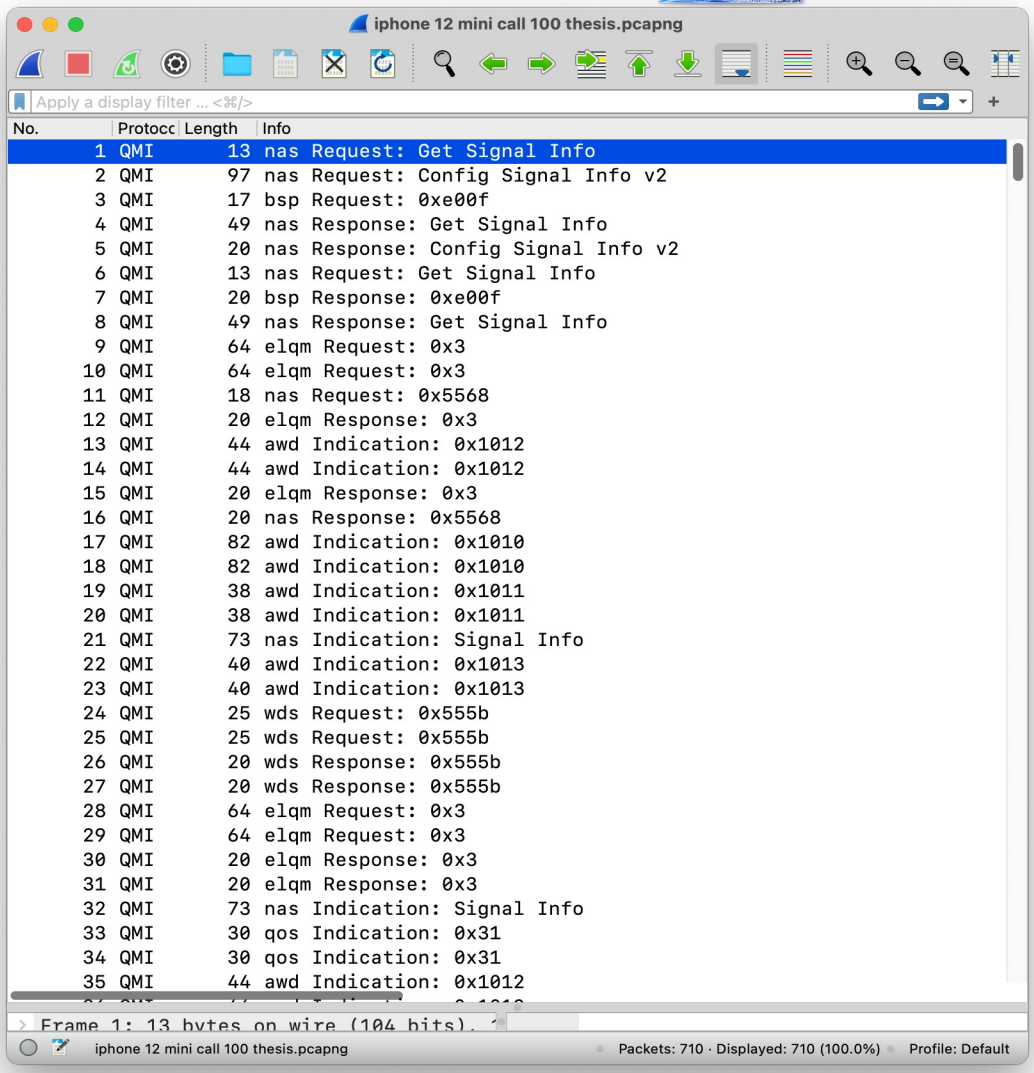
QMI Wireshark Dissector

Existing dissector for QMI

- Based on libqmi, but built for USB modems (@dnlplm)
- Adapt to iOS

Three approaches to **extract QMI packets** from iPhones

- On- and Offline
- Jailbroken and non-jailbroken iPhones



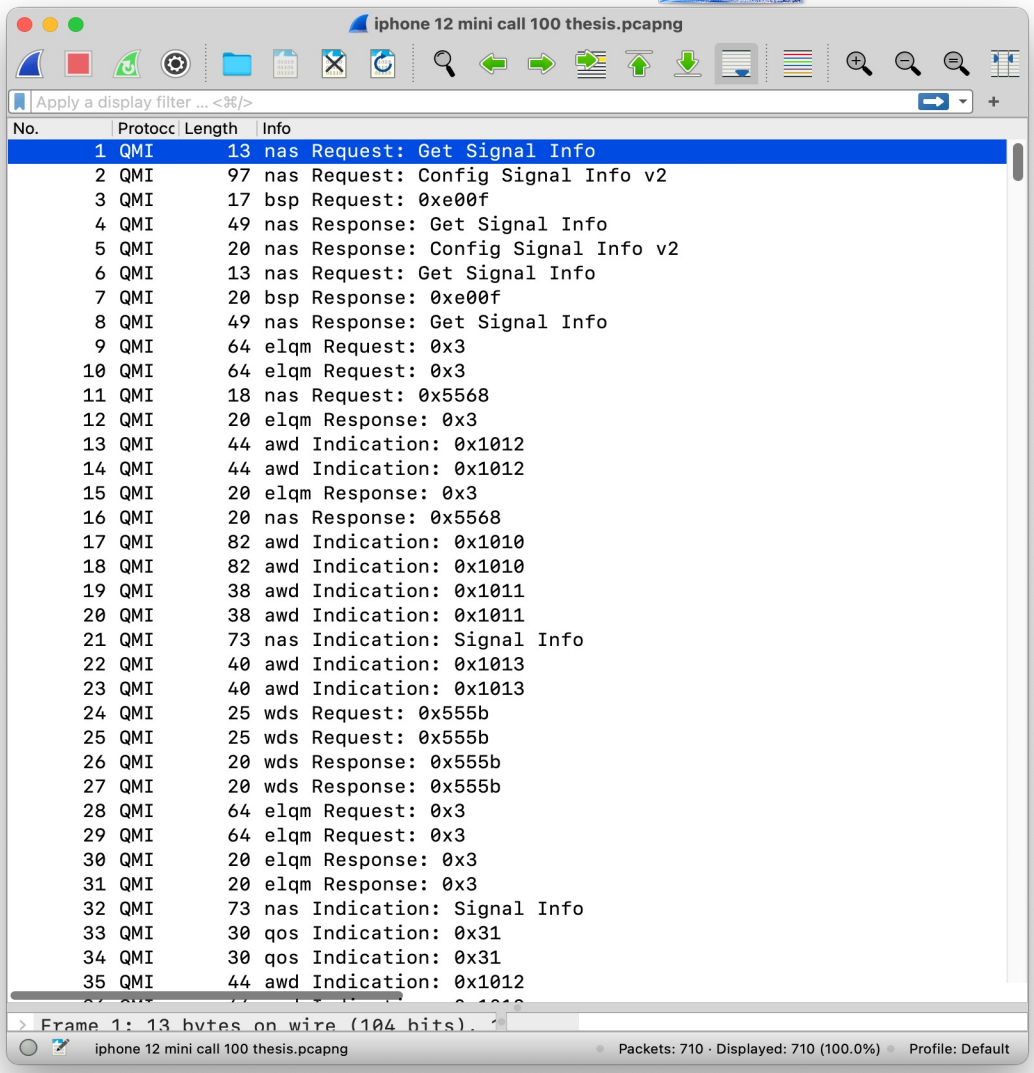
QMI Wireshark Dissector

Dissector **Issue**

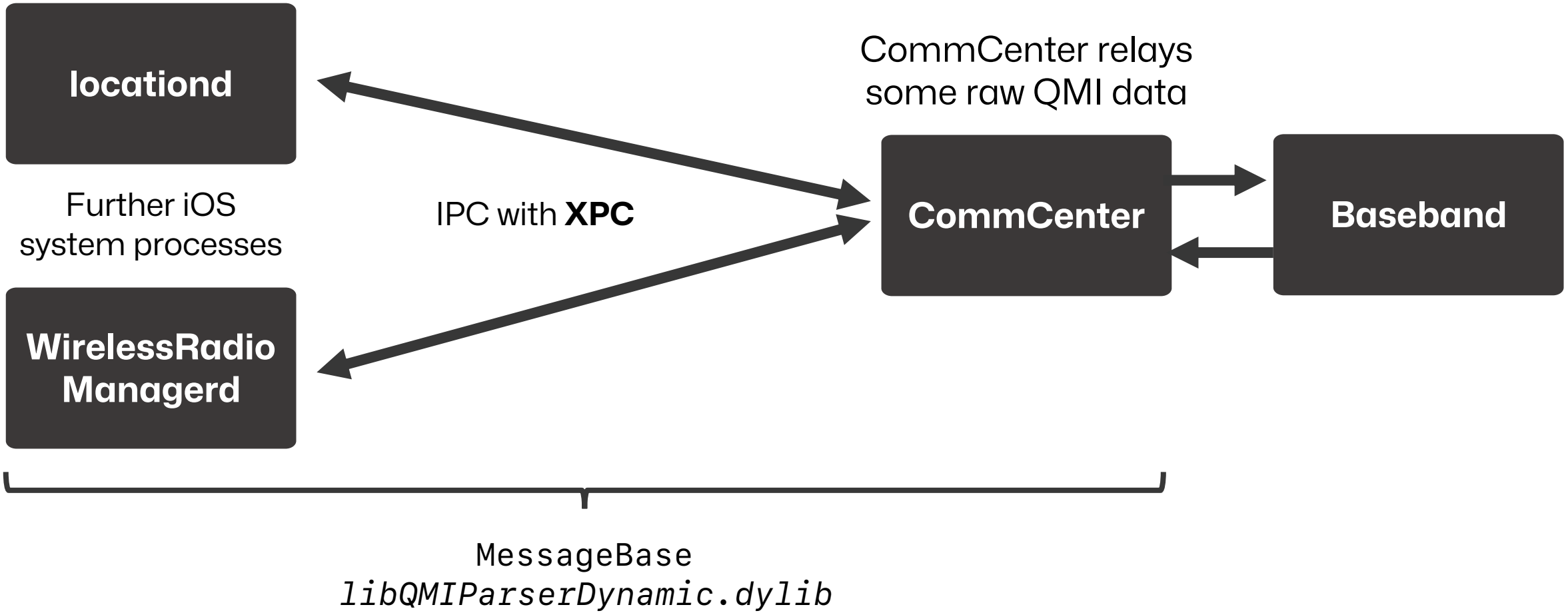
- Misses many of iOS' message identifiers
- Crucial for understanding the baseband comm.

Resolve the Issue

- Research how iOS parses and builds QMI packets



Processing Binary QMI Data

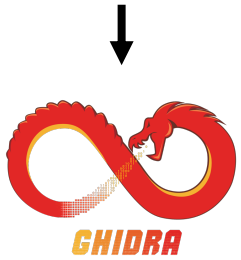


Extracting QMI IDs from iOS

Automatic Message ID Extraction Workflow

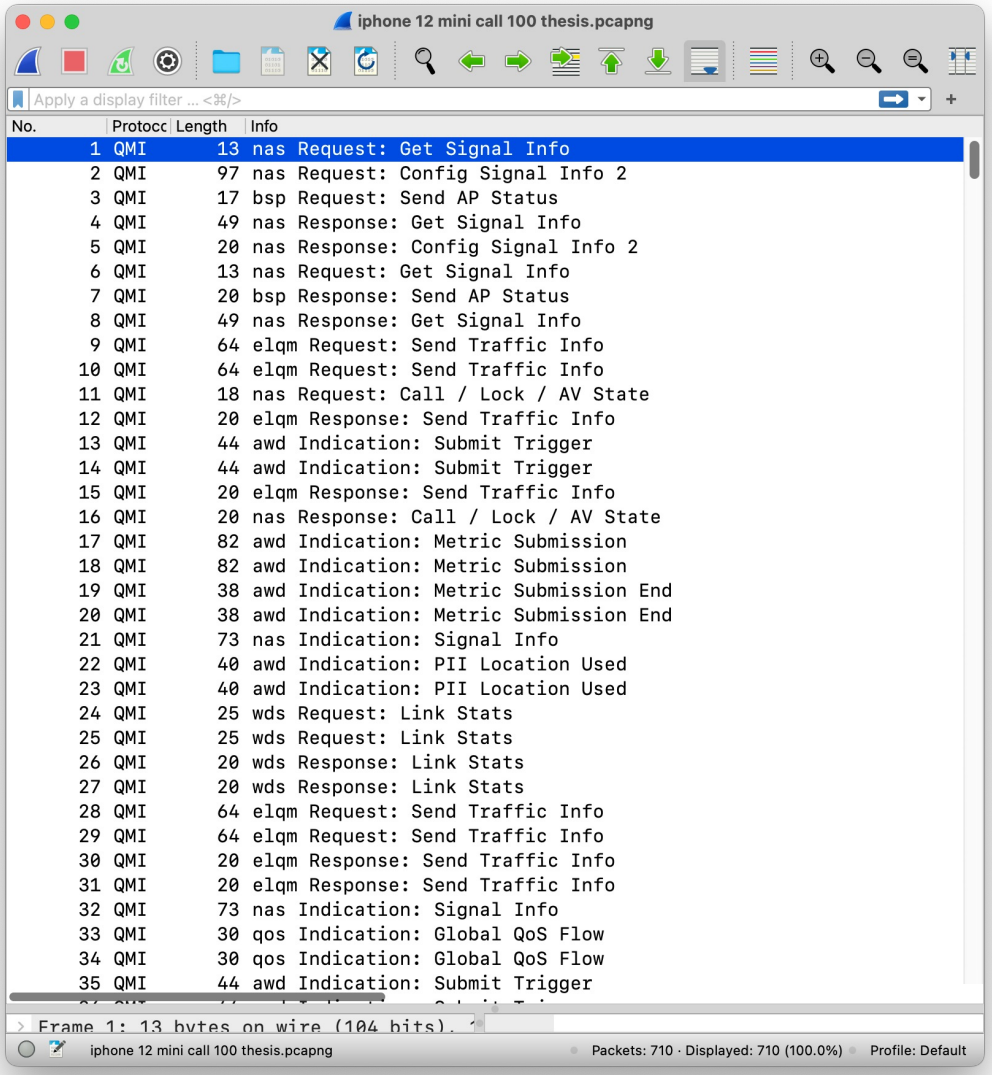
- Based on MessageBase class

iOS Firmware



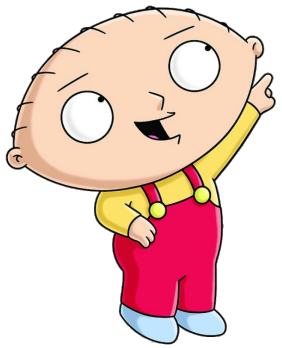
Extract Message IDs

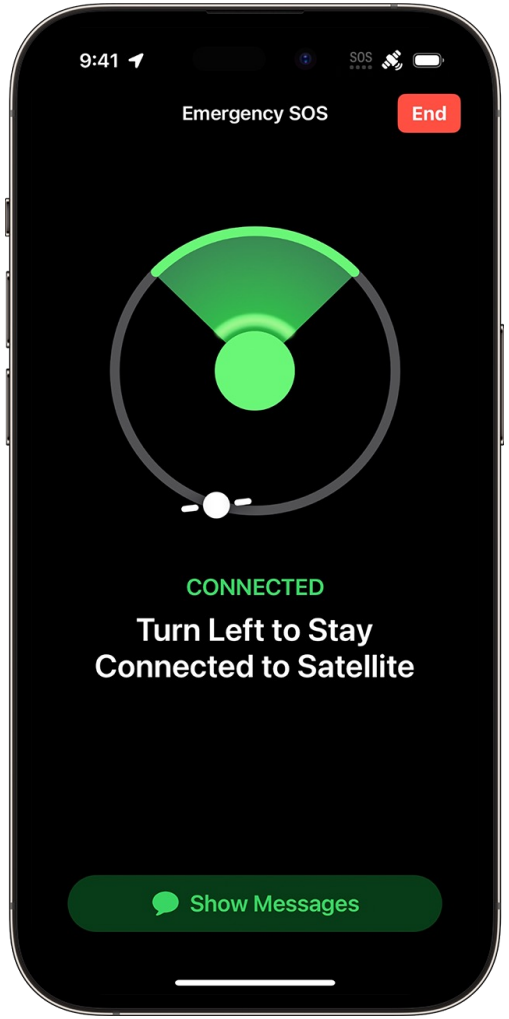
Convert into libqmi Definitions



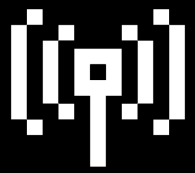
Emergency SOS via Satellite

- Record packets & analyze with QMI Wireshark dissector
- Novel QMI Stewie Service on iPhone 14 baseband (Snapdragon X65)
- Bifröst: Apple's Rainbow Bridge for Satellite Communication



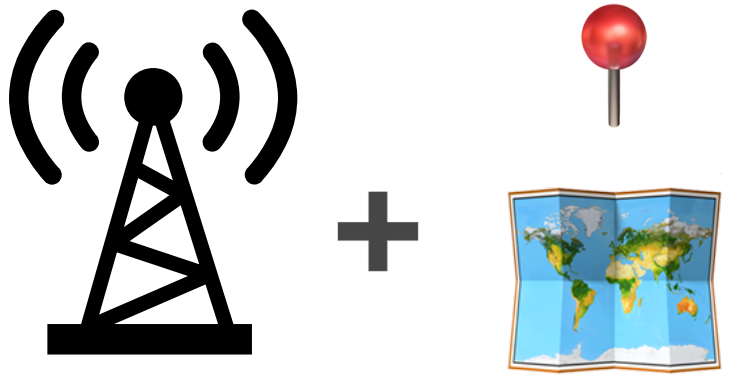


RBS Detection



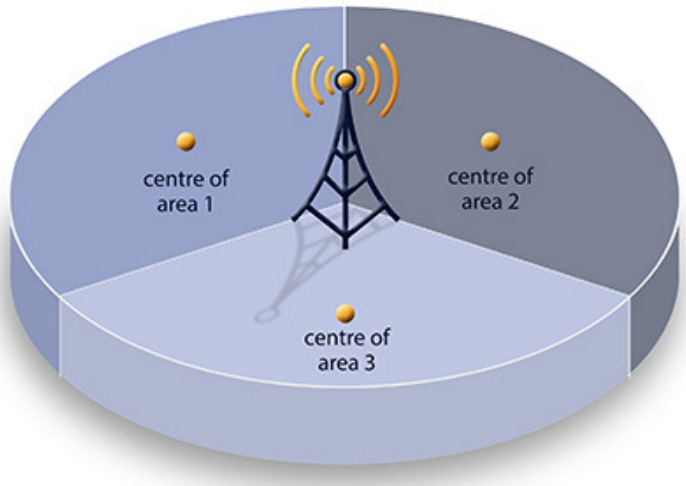
Cell Location Databases

- Cell towers hold multiple cells
- Cell location databases link cell identification with location



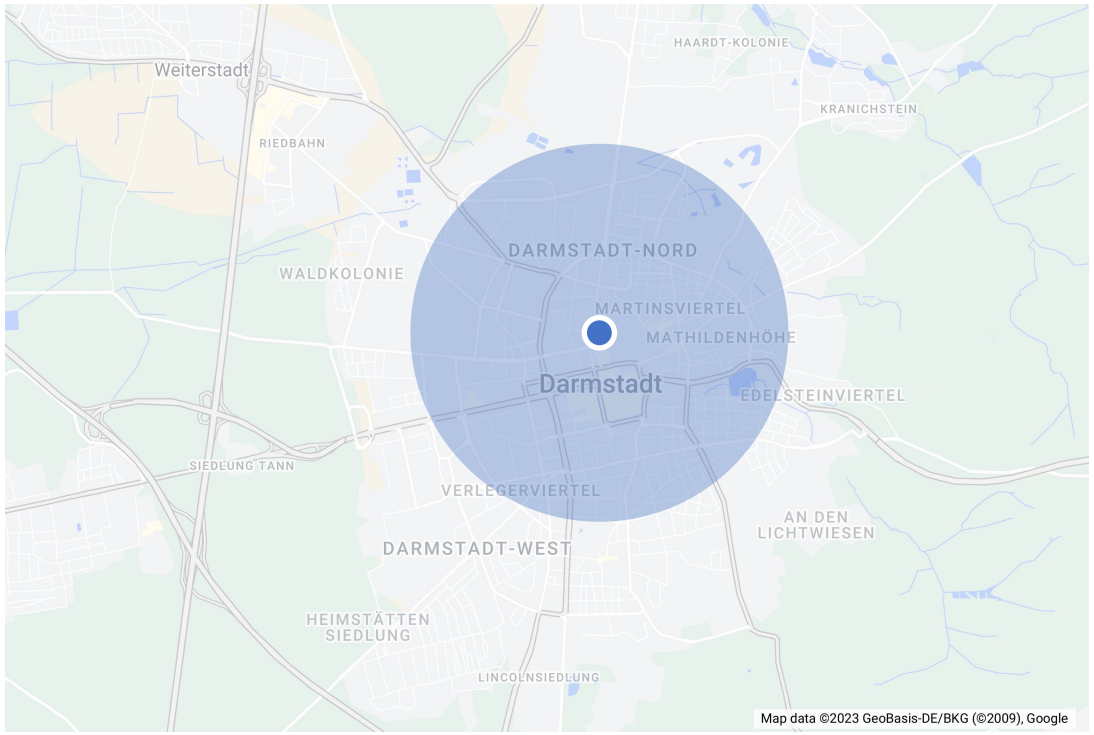
Cellular Base Station

cell tower with 3 cells, each with 120° angle

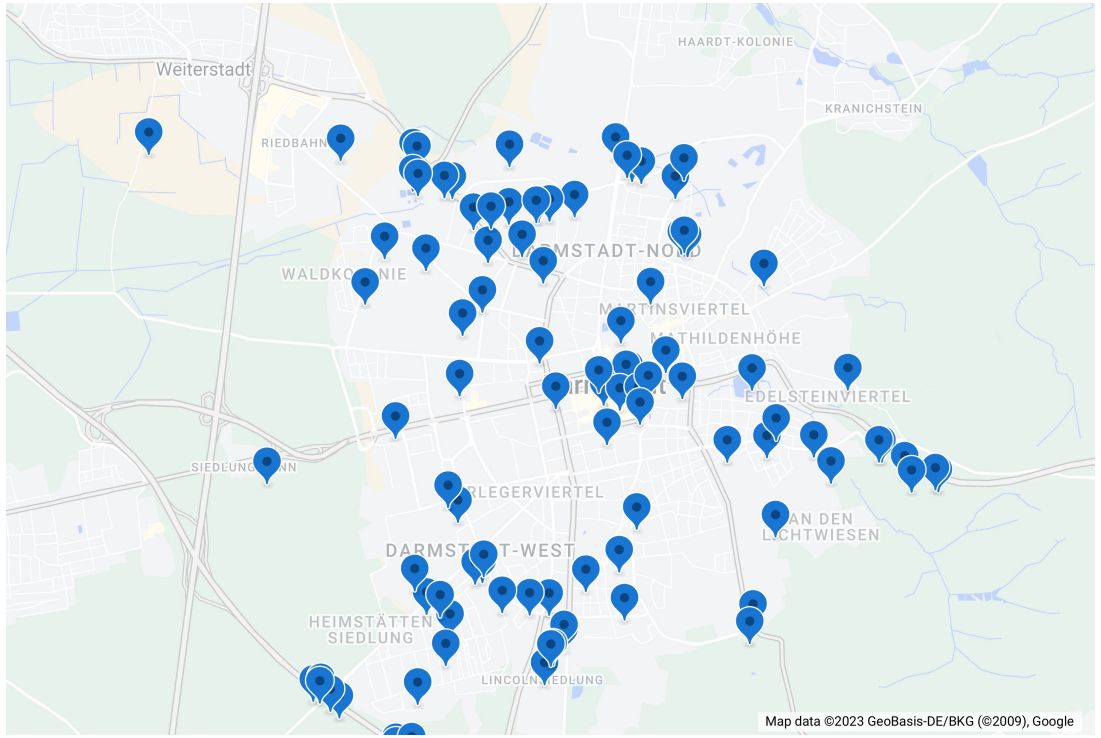


Applications of Cell Location DBs

Determine Location



Index Cellular Networks



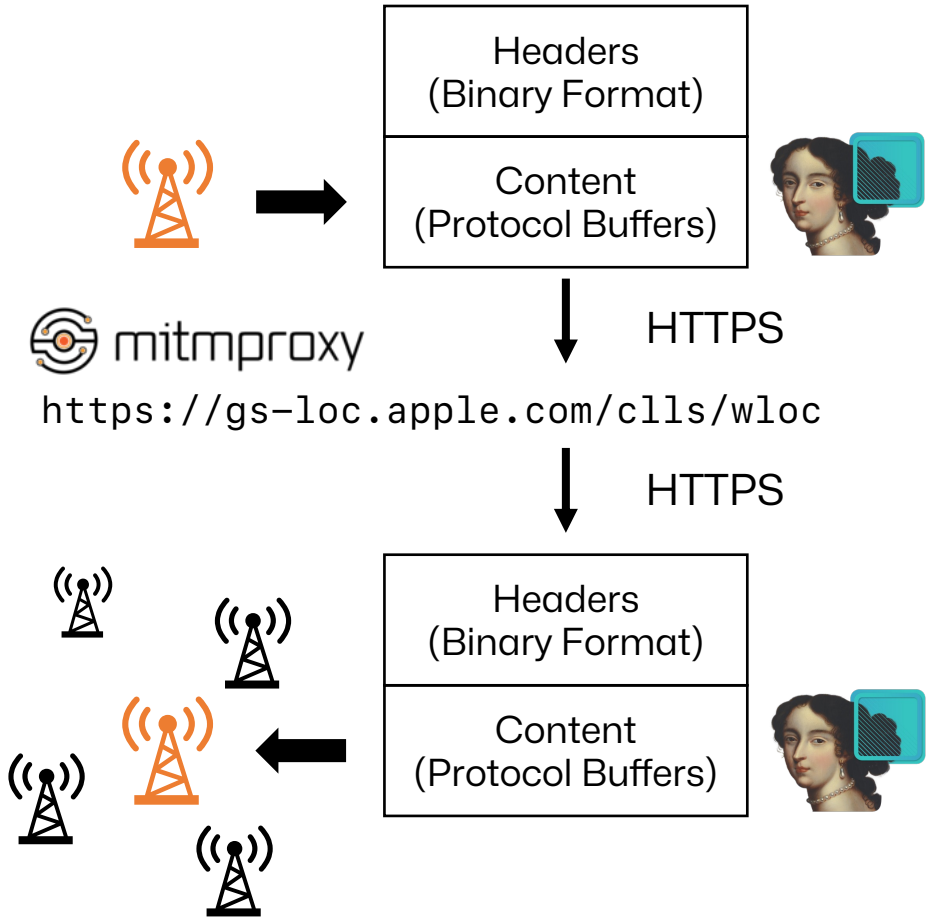
Apple Location Services

- iPhone use multi-sensory approach to determine location
- Apple Location Services
  - Provides aggregated data to devices
  - Devices contribute data to DB
- Research into ALS
  - Wi-Fi MACs (Mika Tuupola)
  - Cell IDs (Our Contribution)

[1]

ALS HTTPS Endpoint

- HTTP Body consists of
  - Apple-custom binary headers
  - Content encoded with Protobuf
- Request Parameter
  - Identification for 1 cell
- Successful Response
  - Location of up to 100 nearby cells
  - Locations for 20 cellular areas



ALS HTTPS Endpoint

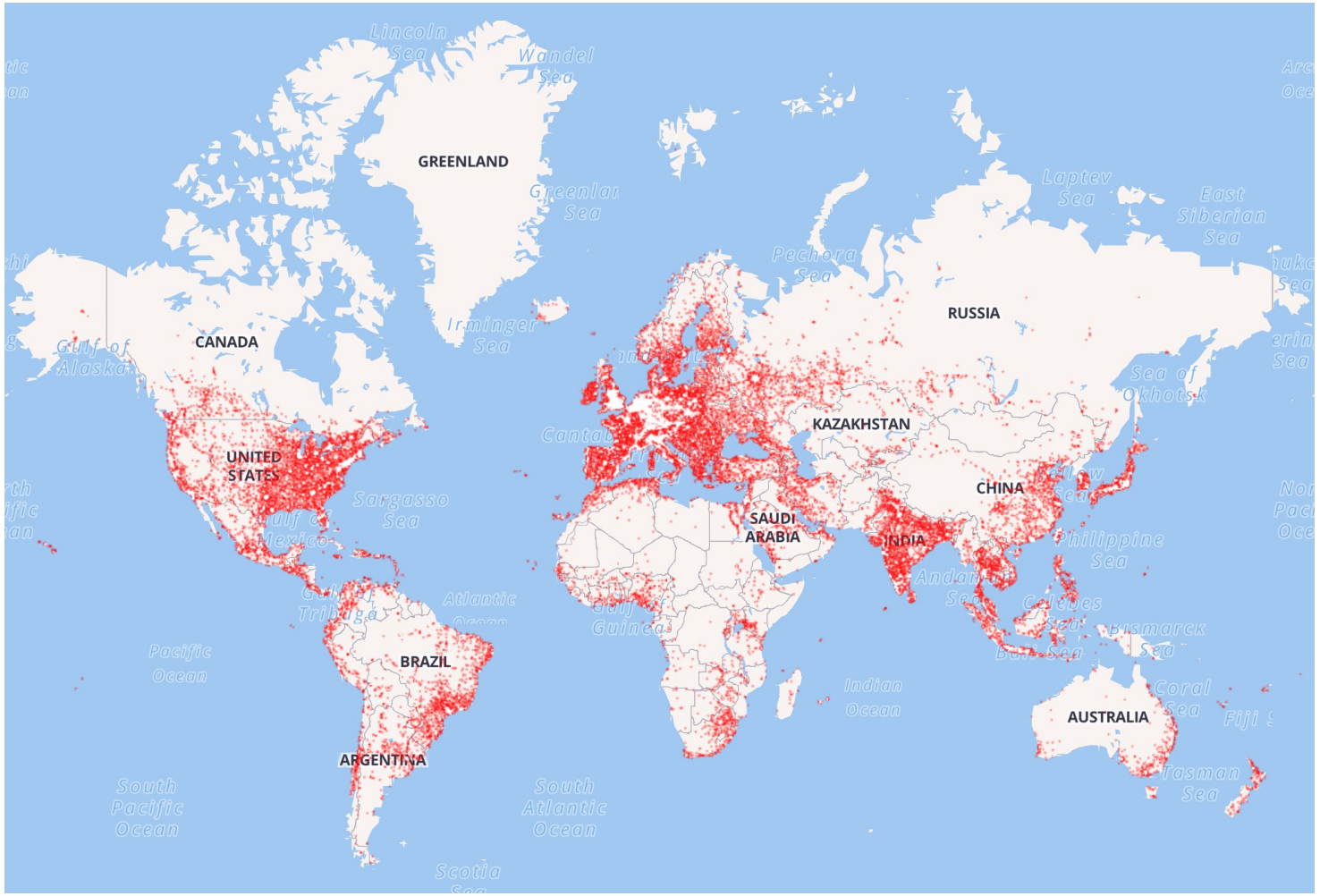
- HTTP Body consists of
  - Apple-custom binary headers
  - Content encoded with Protobuf
- Request Parameter
  - Identification for 1 cell
- Successful Response
  - Location of up to 100 nearby cells
  - Locations for 20 cellular areas

```
message LteCell {
  optional int32 mcc = 1;
  optional int32 mnc = 2;
  optional int32 cellID = 3;
  optional int32 tacID = 4;
  optional Location location = 5;
  optional int32 uarfcn = 6;
  optional int32 pid = 7;
}
```

ALS Characteristics

- Contributions by over 2 billion active devices
- Good coverage across networks & generations
  - Miss Rate in Europe < 1%
- New cells take multiple days to end up in the DB

OpenCelliD

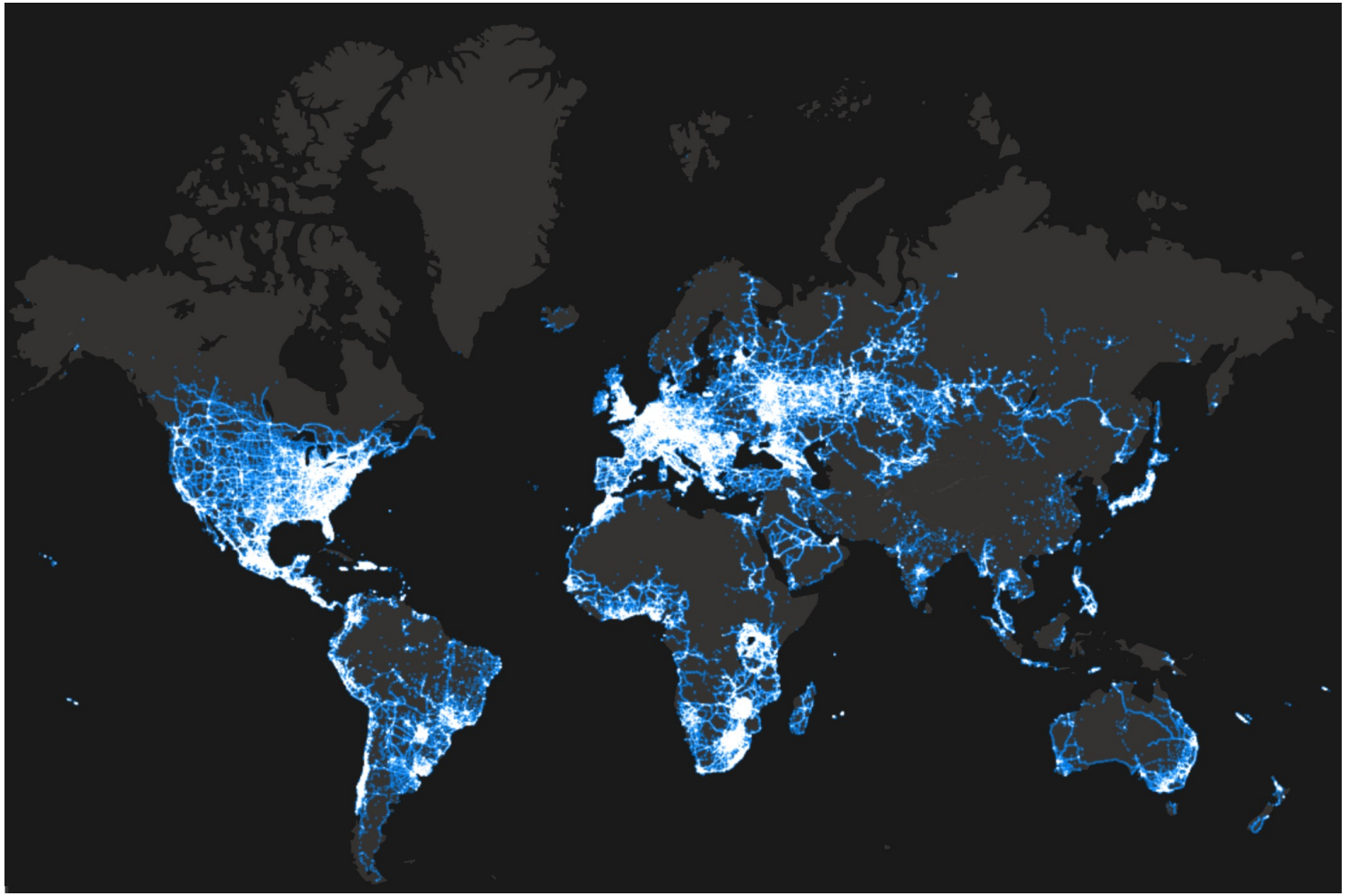


Size:
47 million cells

Vendor:
Unwired Labs

unwiredlabs
OpenCellid

Mozilla Location Service

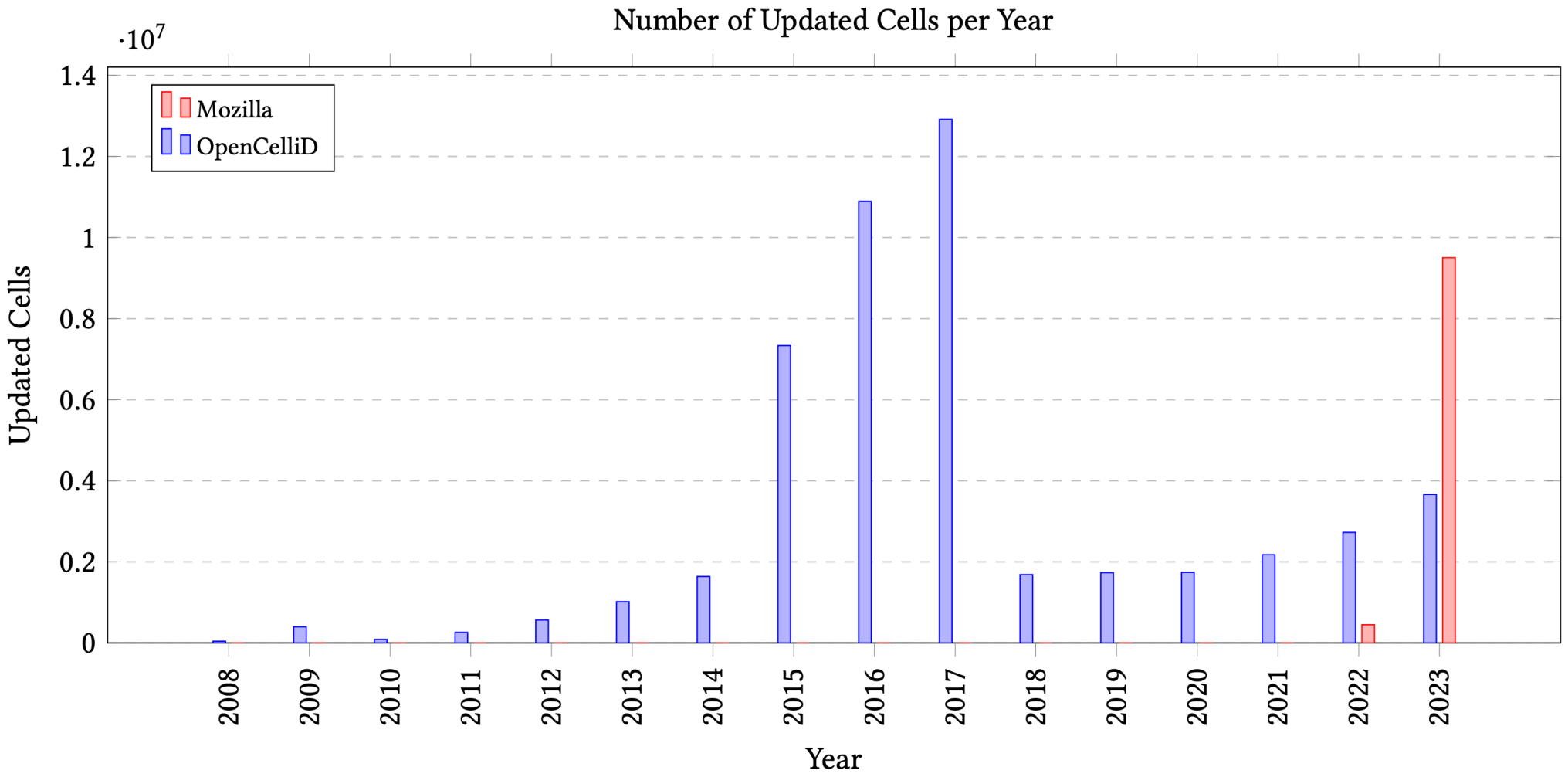


Size:
63 million cells
(9 million cells)

Vendor:
Mozilla

moz://a
location service

Evaluating Database Freshness



Evaluating Database Coverage

- Experiment 🔬
  - Repeat for 10 combinations of country & database
- Mozilla Location Service 🦊
  - Contains low % of out-of-service cells
  - Covers only a small % of a country's networks
- OpenCelliD 🔌
  - Contains a larger % of out-of-service cells
  - Covers a larger % of a country's networks

RBS Detection

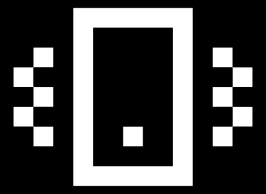
- Based on
  - Cell Measurements
  - QMI & ARI Packets
  - Locations

- Heuristics evaluate combined datasets
  - Score (0 P - 100 P) for each cell measurement
  - Measurements grouped into categories
    - Untrusted (0 P – 49 P)
    - Suspicious (50 P – 94 P)
    - Trusted (95 P – 100 P)

Detection Criteria

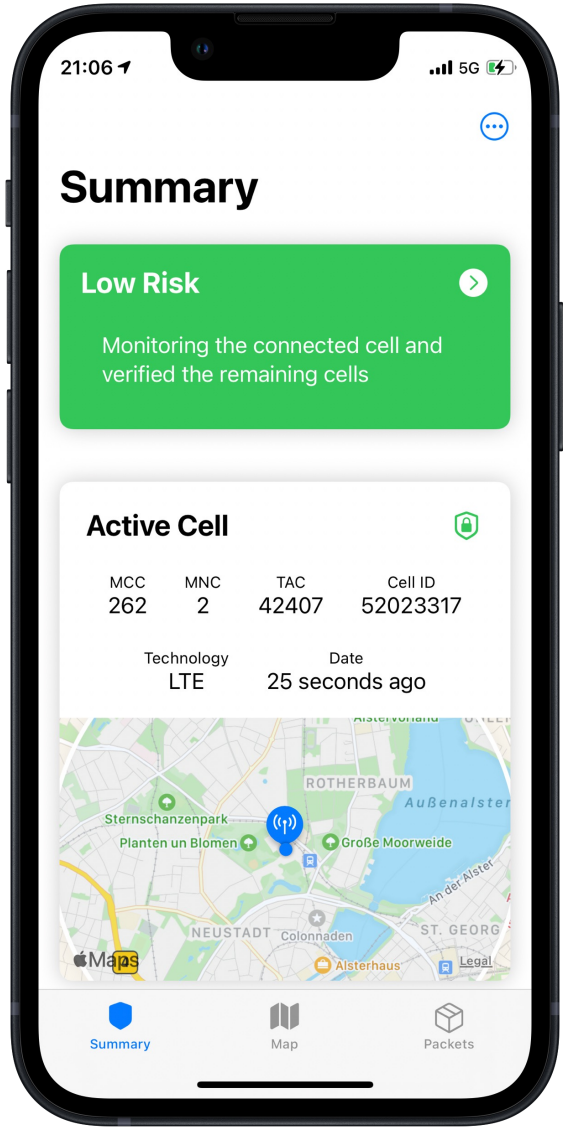
1. Existence of Cell in ALS Database (20 P)
   - RBSes are only active for a short amount of time → Not recorded in ALS
2. Distance between ALS Cell and User Location (20 P)
   - distance(cell, smartphone) + error margins > 75km ?
3. Comparison of Cell's Frequency Channel and PID (8 P)
   - RBSes may use other channels & PIDs not to cause interference
4. Bandwidth (2 P)
   - More expensive SDRs are required to utilize full channel bandwidth
5. Network Reject Packet (30 P)
   - Detect failed authentication between UE & BS
6. Signal Strength (20 P)
   - RBSes trick targets into connecting by advertising a higher signal strength

CellGuard



CellGuard iOS App

- Implements detection algorithm
- Records & links required data points
- Compatible with iOS 14 – 17
- Created with Swift & SwiftUI

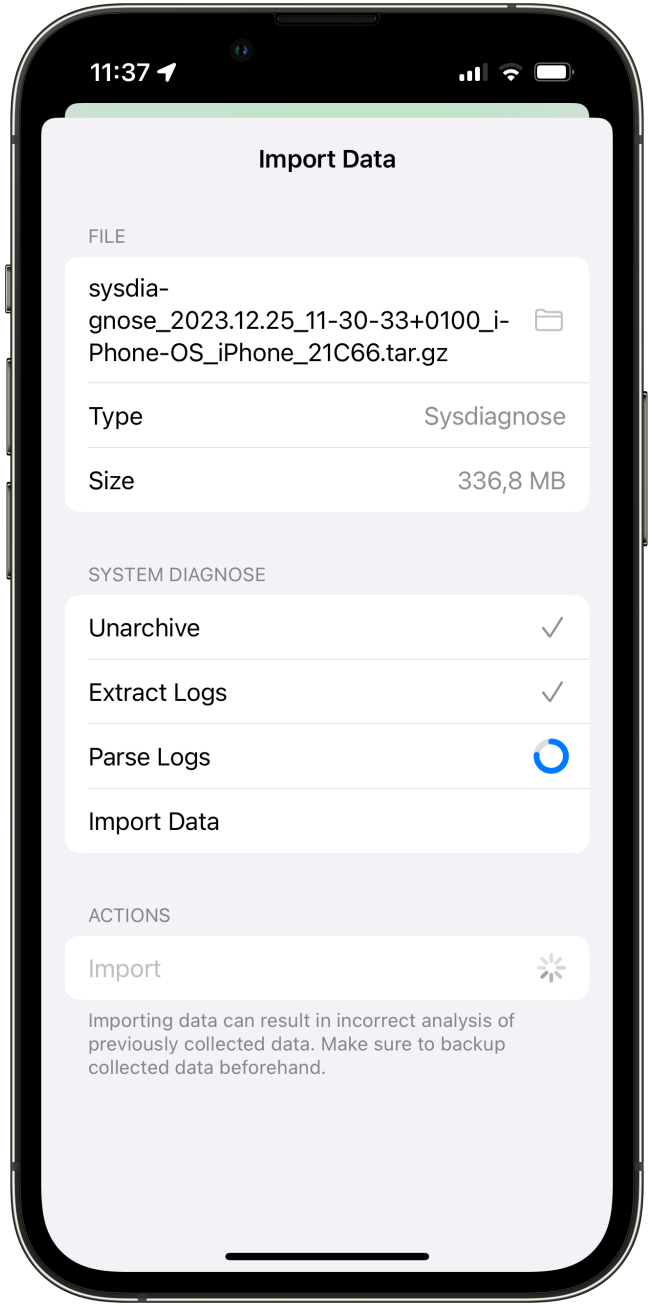


Jailbroken iPhones

- Tweak
  - External component with elevated privileges
  - Modifies default actions of iOS
  - Requires jailbroken iOS
- CellGuard's tweaks
  - Hook into iOS' CoreTelephony framework
  - Collect & cache data (cells / packets)
  - Provide data via local TCP sockets

Non-Jailbroken iPhones

- System Diagnoses
  - Diagnostics snapshot of an Apple device
  - Officially supported by Apple
  - Contain past system logs
- To import data into CellGuard
  - Install the Baseband debug profile
  - Create a sysdiagnose
  - Share sysdiagnose with CellGuard



Non-Jailbroken iPhones

- System Diagnoses
  - Diagnostics snapshot of an Apple device
  - Officially supported by Apple
  - Contain past system logs
- To import data into CellGuard
  - Install the Baseband debug profile
  - Create a sysdiagnose
  - Share sysdiagnose with CellGuard
- CellGuard extracts data from sys-diagnoses with macos-unifiedlogs

macos-unifiedlogs

Which processing time? I run on every smart toaster

Apple's Console

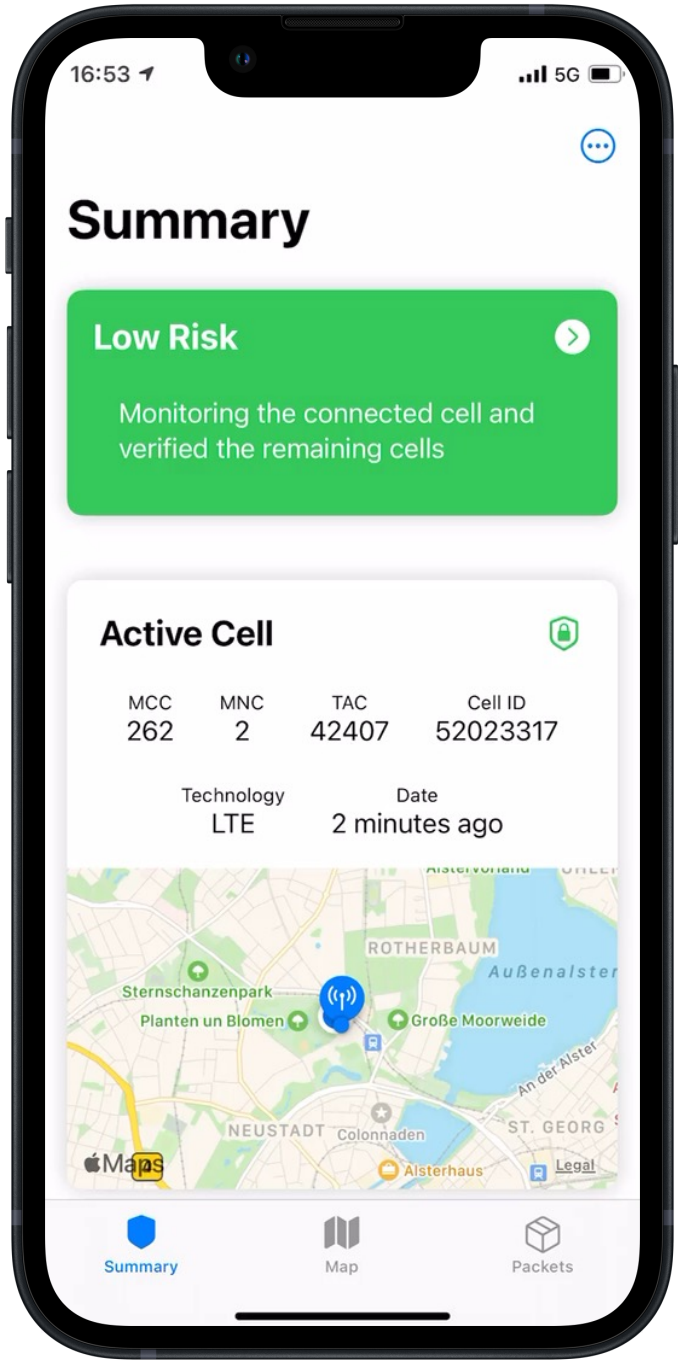
Please wait 10 minutes and use a Mac

CellGuard Demo

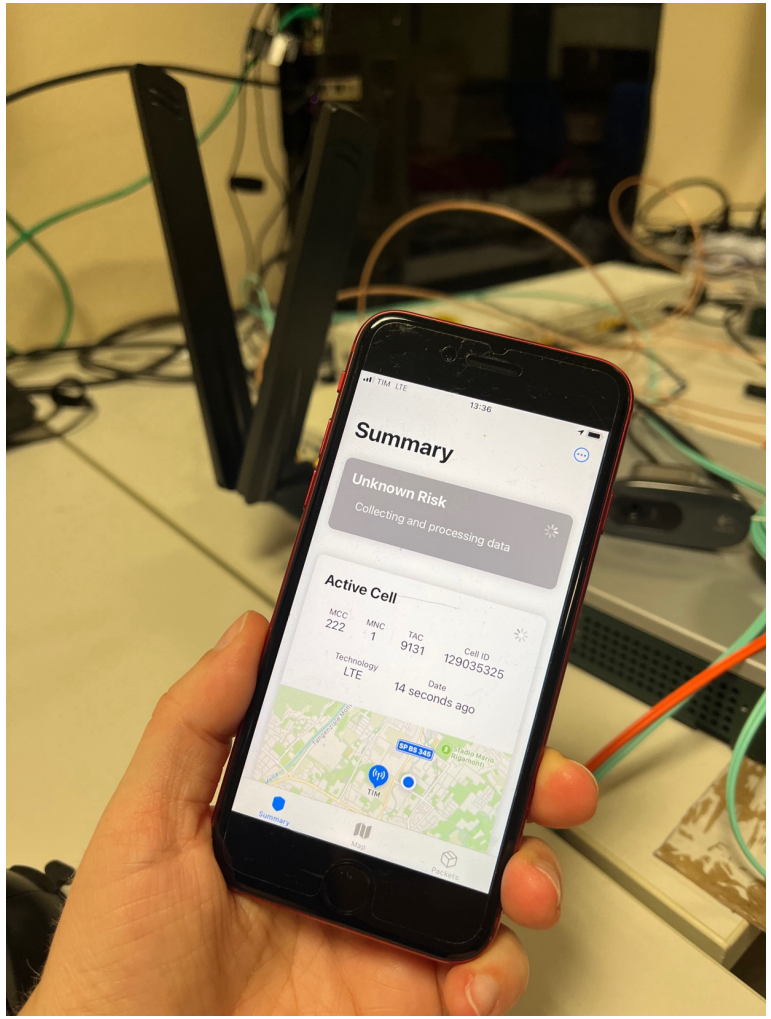
Summary
- Cells
- Cell Details
- Cell Measurements
- Settings

Map
- ALS Cell Details

Packets
- Packet Filter
- Packet Details

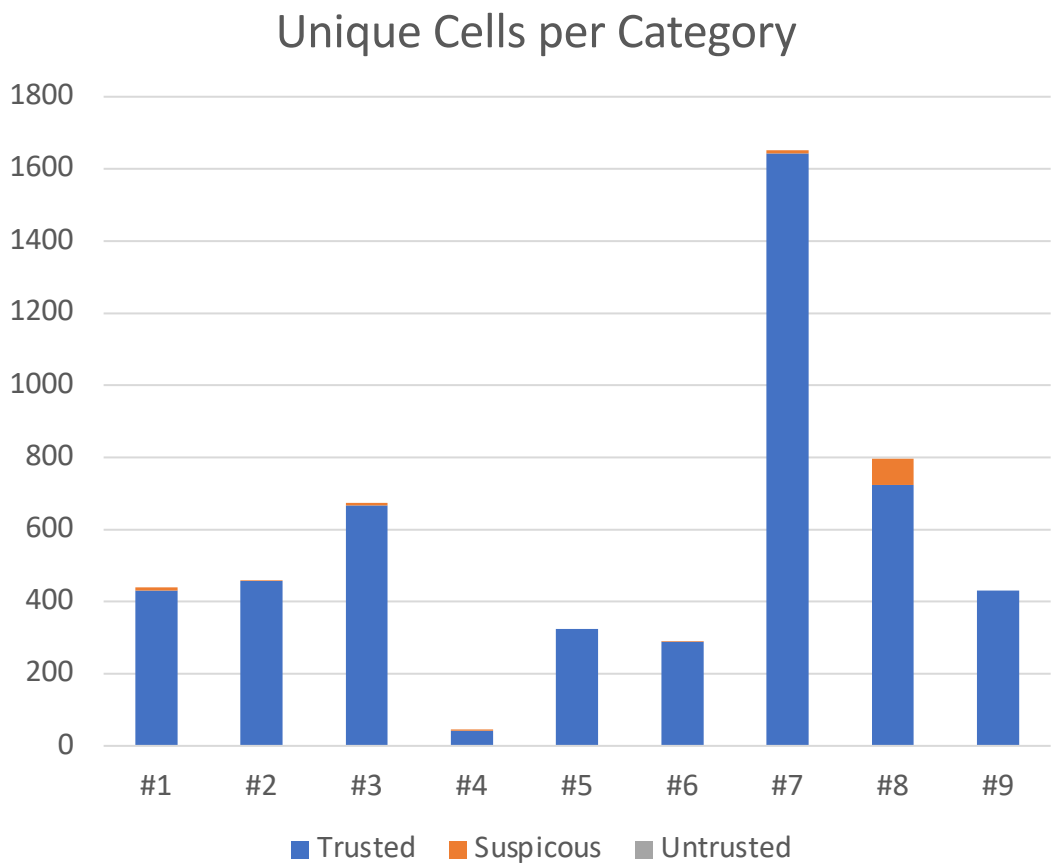


Evaluation in Lab with SDRs



- Set up custom LTE IMSI catcher in lab environment
- Monitor baseband packets with tooling & CellGuard
- Improve CellGuard's detection algorithm based on findings

Evaluation in the Wild



- Collected data
  - In six European countries
  - For over six months
- Low False-Positive-Rate
  - Except when moving at very high speeds (> 500 km/h) on an iPhone SE (#8)
  - Almost all cells were in ALS
- Tested CellGuard with 2G, 3G, 4G, 5G cellular networks

wen eta?

Goals

Provide

insights into iOS

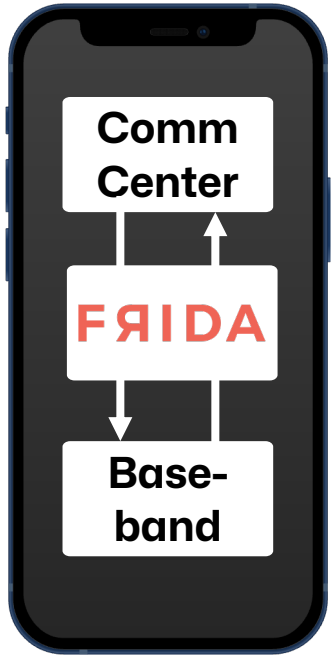
Monitor

illicit use of RBSes

Protect

high-risk groups

So far, we've achieved

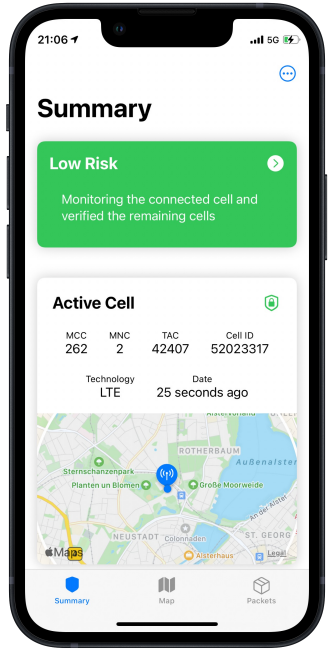


Created novel tooling for QMI on iPhones

unwiredlabs OpenCellid

moz://a location service

Reverse-engineered & evaluated ALS



Developed CellGuard to monitor RBSes

We’re working on

Release: 2024

Improving

detection algorithm

Enhancing

user-friendliness

Smart notifications

for data import

Image Sources

- Thieves: Home Alone
- iPhone 12 mini: Rafael Fernandez, CC BY-SA 4.0, via Wikimedia Commons
- Qualcomm X55 Baseband: Qualcomm, Snapdragon X55 5G Modem-RF System
- Emergency SOS: Apple, Emergency SOS via Satellite Screenshot
- Stewie: Family Guy, Stewie Griffin
- Antenna Segments: OpenCelliD, Antenna Segments
- Meme Templates: imgflip.com
- Icons: Pixelarticons
- Fonts: Mona Sans, VCR OCD Faux

Q&A

Question for later?

larnold@seemoo.de

@lukasarnld

@lukasarnld@mastodon.social

Special thanks to jiska and Prof. Matthias Hollick