

MobileSDR: A Mobile Programmable Platform for Wireless Field Tests and Diagnostics

AIMS-2025

Zesen Zhang, Rohith Reddy Vennam, Maiyun Zhang, Yunxiang Chi,
Dinesh Bharadia, Aaron Schulman



Infrastructure Components in Access Network



Last Miles Components

What is wireless measurement

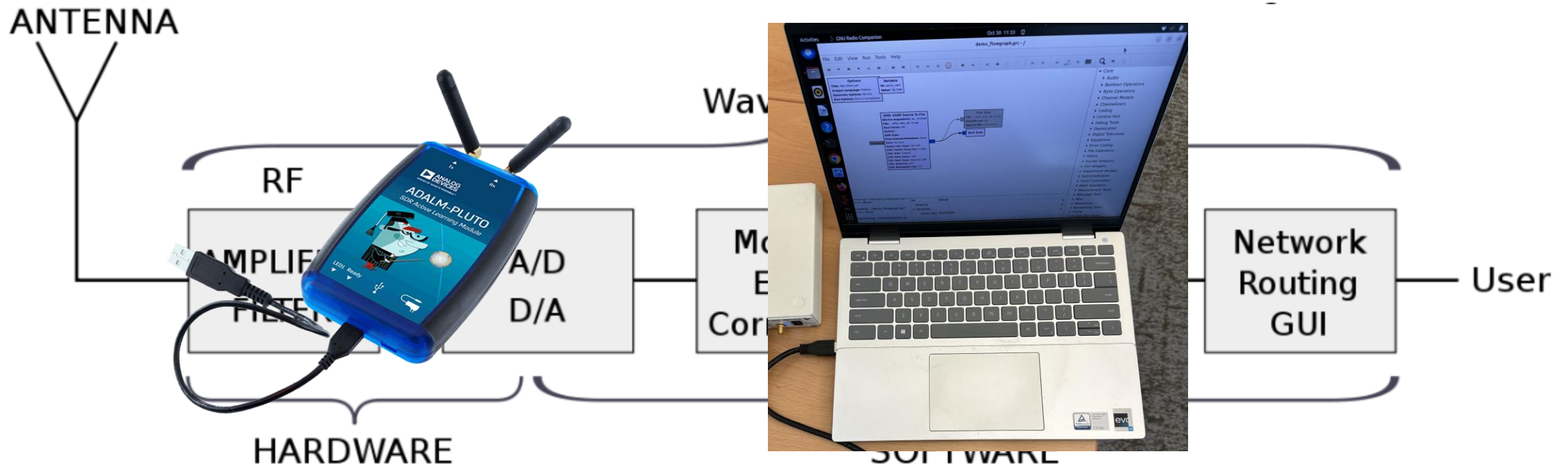
Using sensors to collect analog signal and process with program to capture key information in the signal

- **Sensors:** Collect raw data, such as sound, FM, LoRa, cellular signal
- **Analog-to-digital converter:** Converts the raw data to a digital format if needed.
- **Key feature capture:** FFT, demodulation.
- **Further analysis:** Packet decoding, decryption...

What is SDR



- **Software-defined radio (SDR)** is a radio communication system who uses software to implement components which is conventionally implemented by analog hardware.



Collecting Wireless Data in the field is not easy...

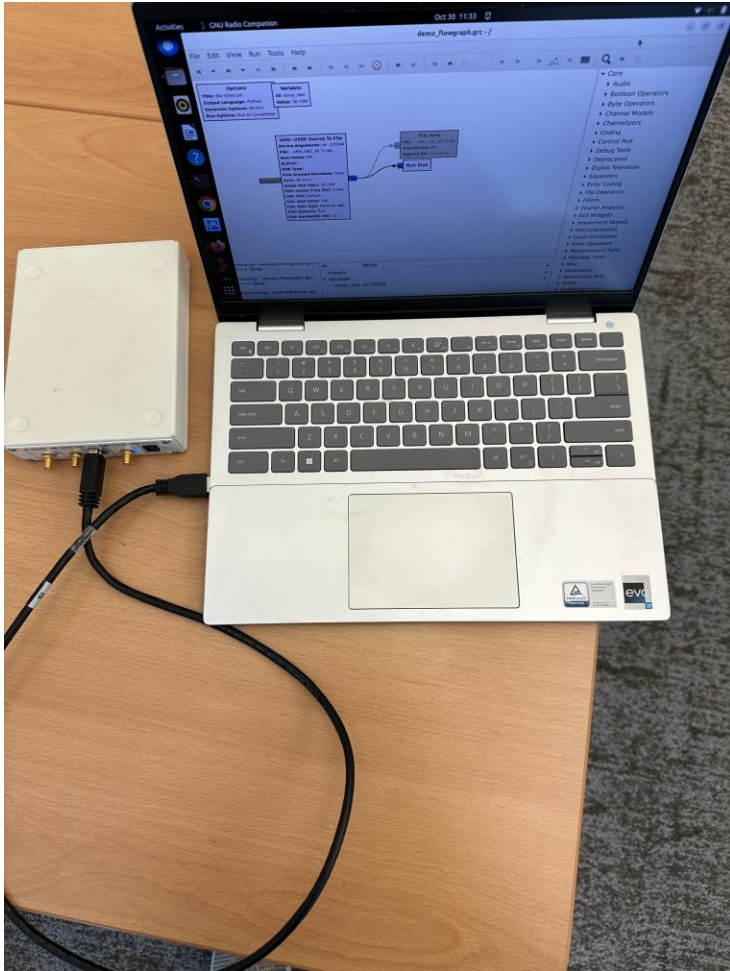


Design program in lab



**Bring a laptop with SDR and
Drive/Walk around to collect data**

What if...



Connect SDR to laptop



Connect SDR to Phone

So that...

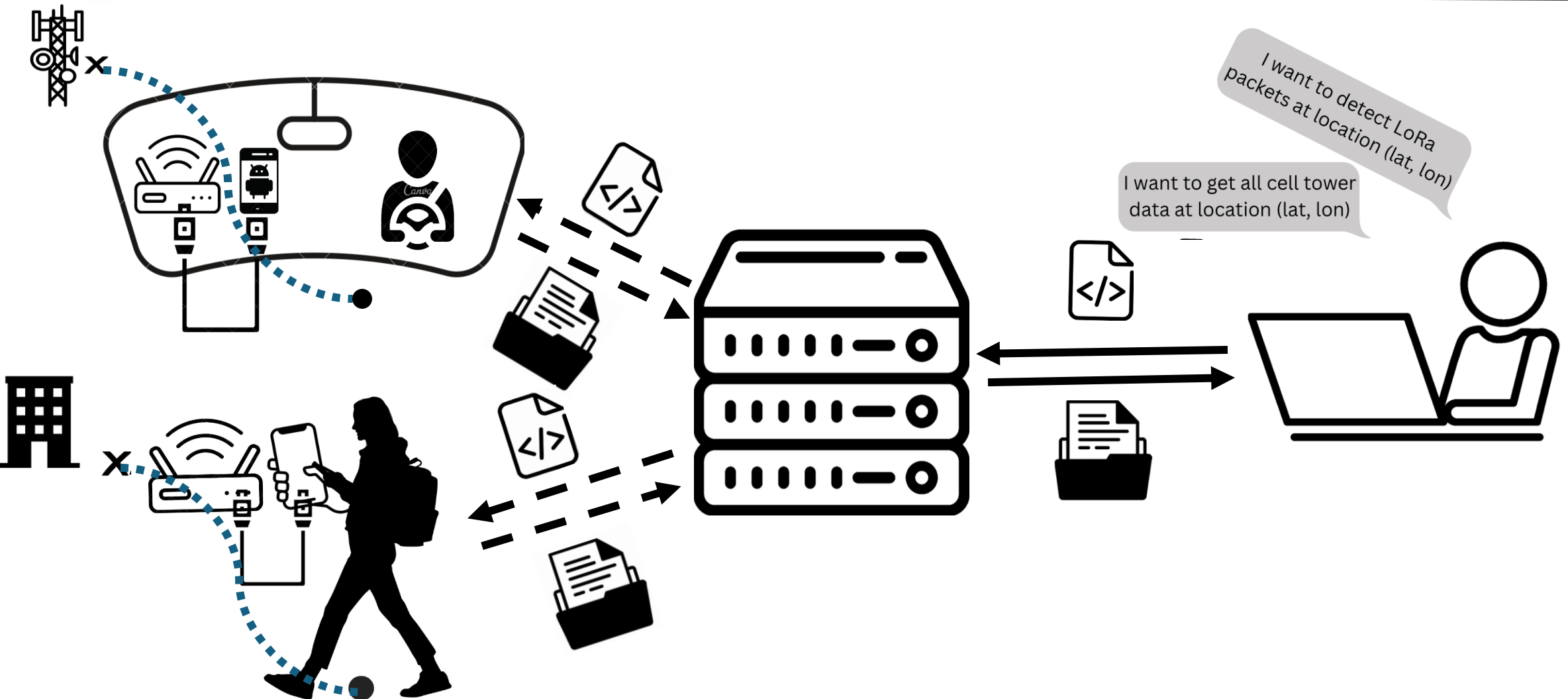


Design program in lab



**Bring a phone with SDR and
Drive/Walk around to collect data**

What if...



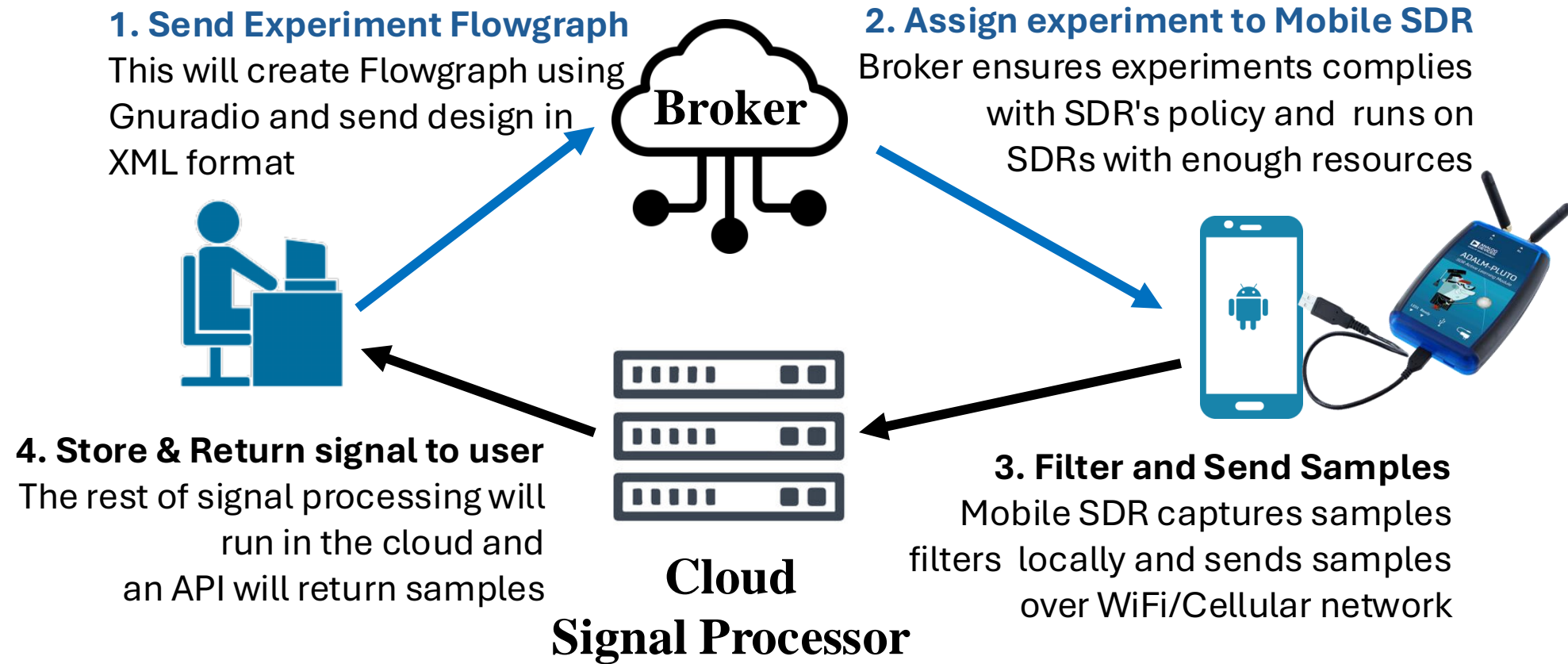
MobileSDR Goal

Separate wireless data collection from program design

Developing a crowdsourced platform

- **Researchers design programs in the lab and publish them on the platform**
- **Users download tasks to their phones and gather data via SDR**
- **Users then upload the collected data to the platform, making it available for everyone**

MobileSDR architecture



To achieve this Goal we need to..

Easy for endpoints to collect data and join the ecosystem

Ensure program security running on endpoint's device

Ensure data integrity

Save battery on the phone

Go through limited computation resource on the phone

To achieve this Goal we need to..

Easy for endpoints to collect data and join the ecosystem

Ensure program security running on endpoint's device

Ensure data integrity

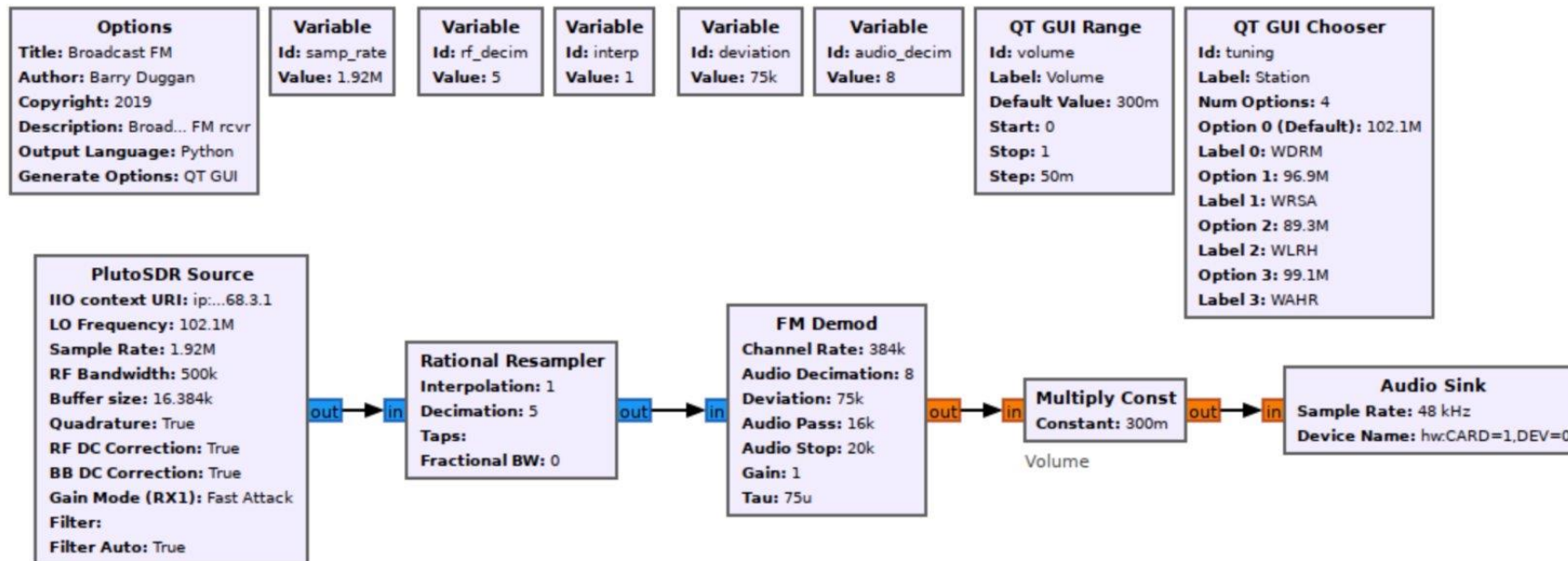
Save battery on the phone

Go through limited computation resource on the phone

Ensure program security running on endpoint

Use domain specific language to restrict what researchers can run on the platform while providing enough flexibility

A **domain-specific language (DSL)** is a computer language specialized to a particular application domain.



Only allow sending XML file instead of binary



Mako template

```
options:
  parameters:
    author: ''
    category: '[GRC Hier Blocks]'
    cmake_opt: ''
    comment: ''
    copyright: ''
    description: ''
    gen_cmake: 'On'
    gen_linking: dynamic
    generate_options: no_gui
    hier_block_src_path: '.: '
    id: androidsdrtest
    max_nouts: '0'
    output_language: python
    placement: (0,0)
    qt_qss_theme: ''
    realtime_scheduling: ''
    run: 'True'
    run_command: '{python} -u {filename}'
    run_options: run
    sizing_mode: fixed
    thread_safe_setters: ''
    title: Not titled yet
    window_size: (1000,1000)
  states:
    bus_sink: false
    bus_source: false
    bus_structure: null
    coordinate: [8, 8]
    rotation: 0
    state: enabled

blocks:
- name: samp_rate
  id: variable
```

```
% if python_version == 2:
from __future__ import print_function
% endif
% if generate_options == 'qt_gui':
from distutils.version import StrictVersion
% endif

if __name__ == '__main__':
    import ctypes
    import sys
    if sys.platform.startswith('linux'):
        try:
            x11 = ctypes.cdll.LoadLibrary('libX11.so')
            x11.XInitThreads()
        except:
            print("Warning: failed to XInitThreads()")
    % endif

#####
##Create Imports
#####
% for imp in imports:
##${imp.replace(" # grc-generated hier_block", "")}
${imp}
% endfor

#####
##Create Class
## Write the class declaration for a top or hier block.
## The parameter names are the arguments to __init__.
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

#
# SPDX-License-Identifier: GPL-3.0
#
# GNU Radio Python Flow Graph
# Title: Not titled yet
# GNU Radio version: v3.8.5.0-6-g57bd109d

from gnuradio import blocks
from gnuradio import gr
from gnuradio.filter import firdes
import sys
import signal
from argparse import ArgumentParser
from gnuradio.eng_arg import eng_float, intx
from gnuradio import eng_notation
from gnuradio import uhd
import time

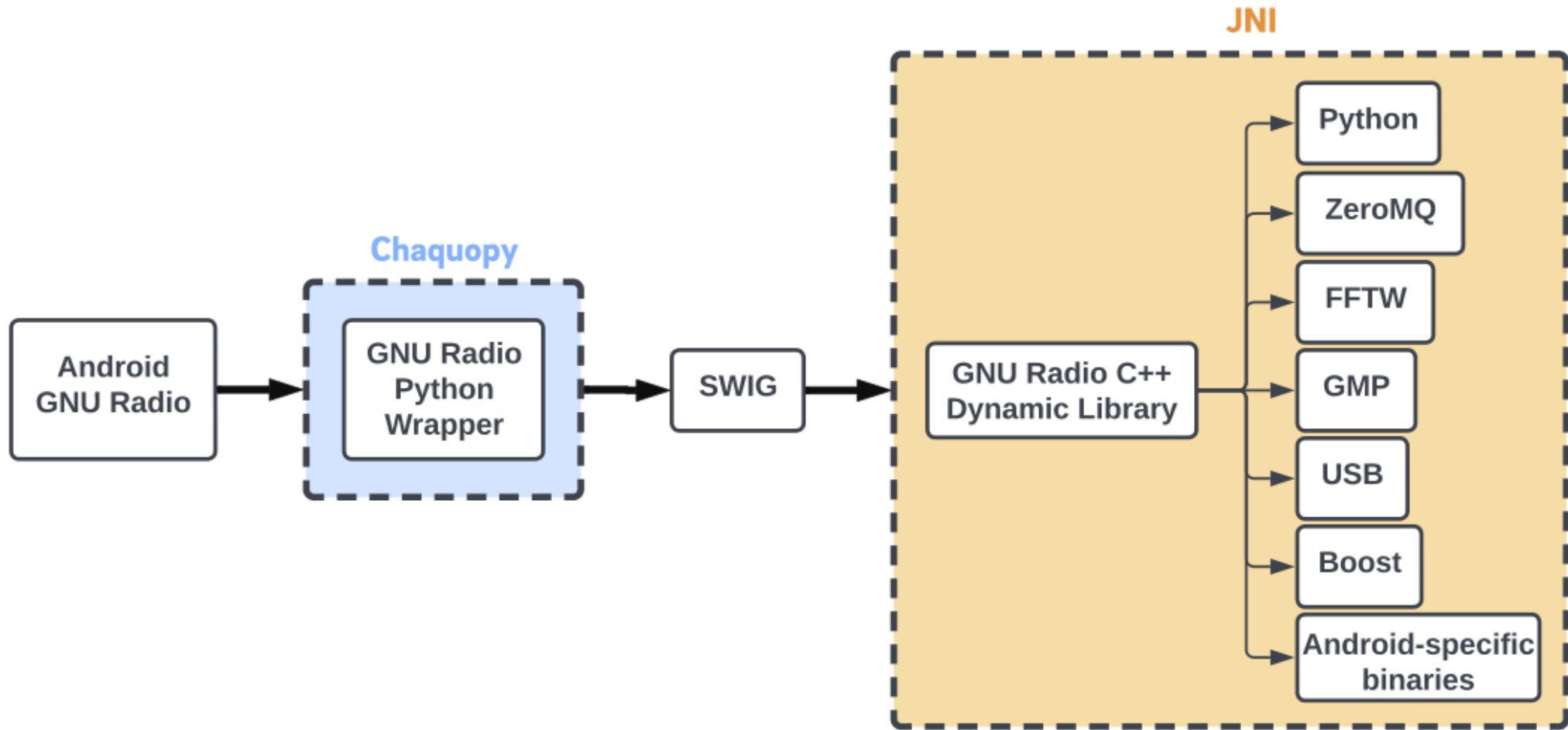
class androidsdrtest(gr.top_block):

    def __init__(self):
        gr.top_block.__init__(self, "Not titled yet")

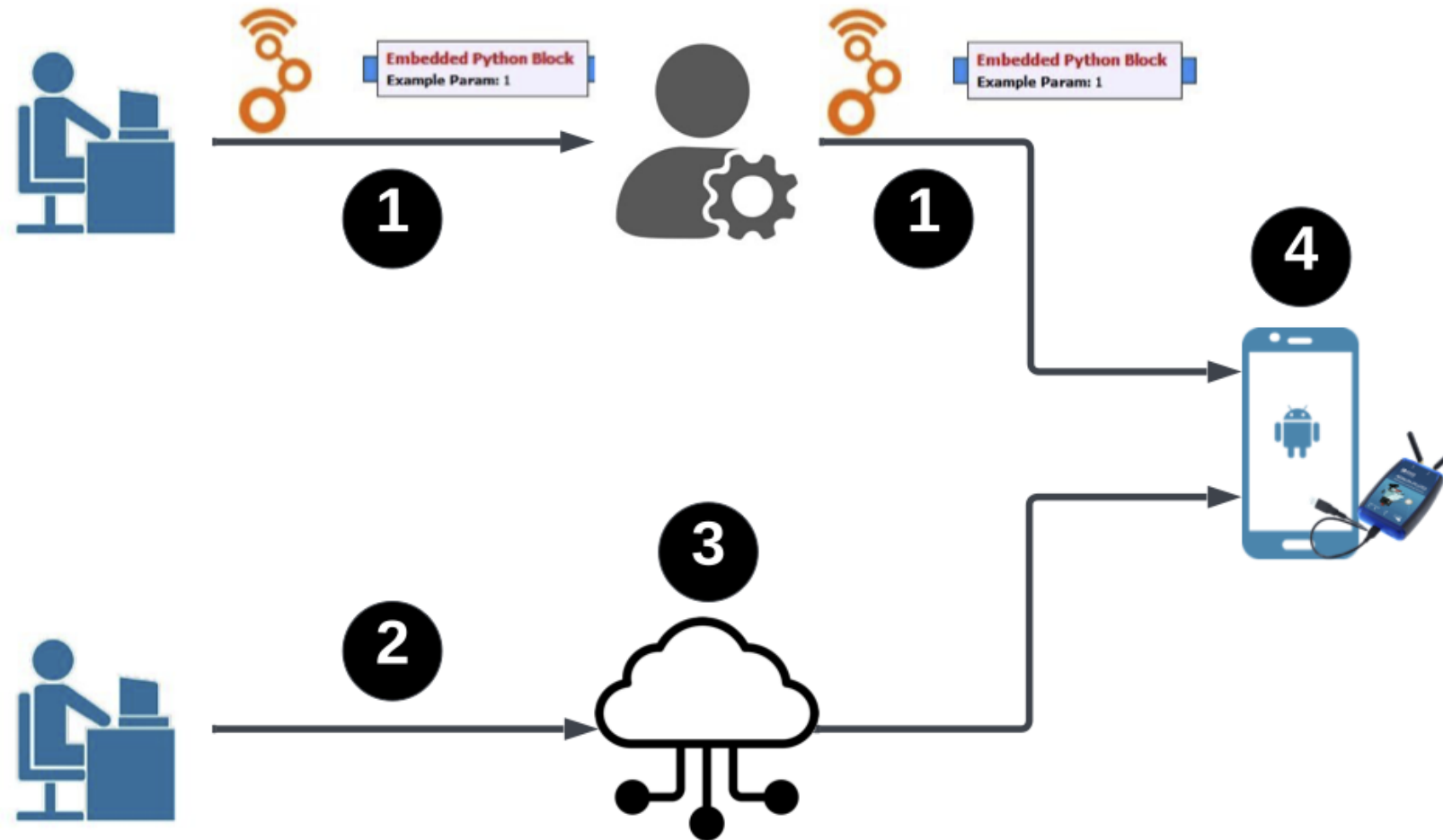
        #####
        # Variables
        #####
        self.samp_rate = samp_rate = 3072000

        #####
        # Blocks
        #####
        self.uhd_usrp_source_to_file_0 = uhd.usrp_source_to_file(
```

Architecture on the endpoint side



Access control



To achieve this Goal we need to..

Easy for endpoints to collect data and join the ecosystem

Ensure program security running on endpoint's device

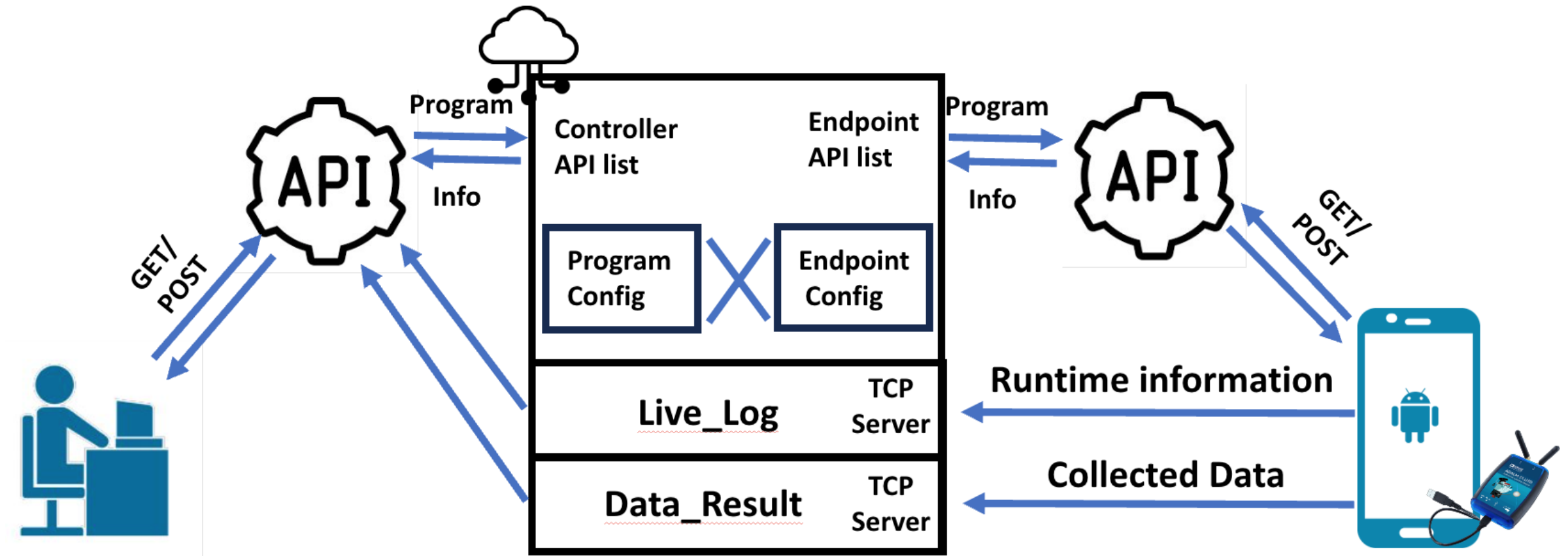
Ensure data integrity

Save battery on the phone

Go through limited computation resource on the phone

Broker automatically sign tasks to endpoint

Broker in the middle control task assignment



Configuration file

```
{  
  "centerlat": "32.86180629",  
  "centerlong": "-117.21663798",  
  "StartTime": "1233333444",  
  "ExpTime": "-1",  
  "range": "1000",  
  "DuplicateTime": "1"  
}
```

Experiment configuration

X

SDR Name: USRP B200

Latitude: 32.86171285

Longitude: -117.21669952

Last Update Time: 2024-10-29T07:04:16Z

Endpoint configuration

Metadata

```
{
  "annotations": [
    {
      "core:freq_lower_edge": 731319999.9987841,
      "core:freq_upper_edge": 746680000.0107517,
      "core:sample_count": 5000000
    }
  ],
  "captures": [
    {
      "core:datetime": "1, 0.250172",
      "core:frequency": 739000000.0047679
    }
  ],
  "global": {
    "core:datatype": "cf32_le",
    "core:hw": "USRP B200",
    "core:sample_rate": 15360000.011967678,
    "core:version": "v0.0.2"
  }
}
```


MobileSDR demo with MIB/SIB decoder

Endpoint List

sd-exp

History Exp

Config File:

Choose File

No file chosen

XML File:

Choose File

No file chosen

Upload

Download result

View Spectrum

MIB/SIB decode

Thanks for your attention

To achieve this Goal we need to..

Easy for endpoints to collect data and join the ecosystem

Ensure program security running on endpoint's device

Ensure data integrity

Save battery on the phone

Go through limited computation resource on the phone

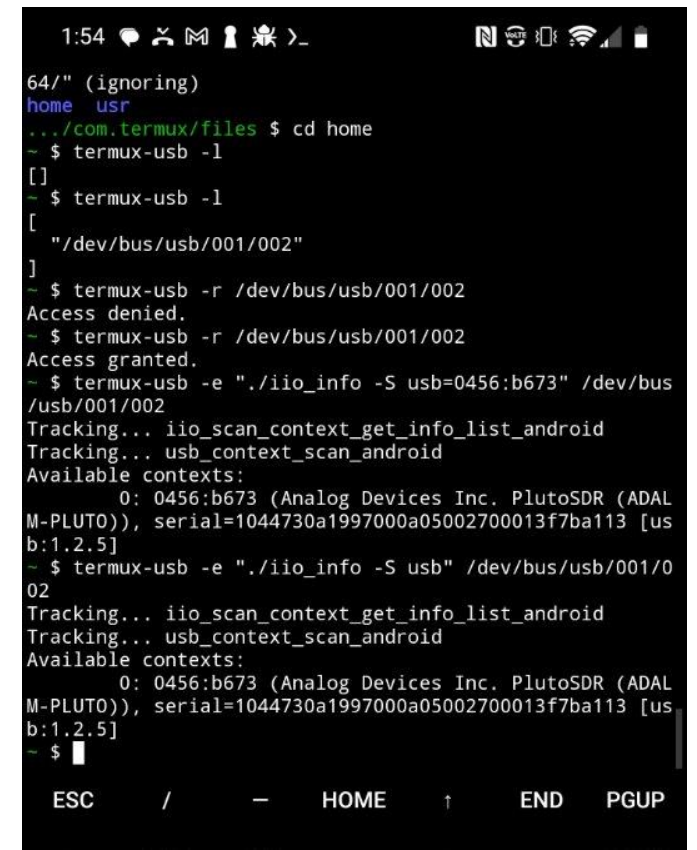
Easy for endpoints to collect data

Allow SDR connected with portable devices (Android phone) to collect data

Cross compile SDR's framework on the phone

1. Cross compile with Android environment on Linux:
2. Compile on Termux:

```
#####  
### CONFIG  
#####  
export TOOLCHAIN_ROOT=${HOME}/Android/Sdk/ndk/21.3.6528147  
export HOST_ARCH=linux-x86_64  
  
#####  
### DERIVED CONFIG  
#####  
export SYS_ROOT=${TOOLCHAIN_ROOT}/sysroot  
export TOOLCHAIN_BIN=${TOOLCHAIN_ROOT}/toolchains/llvm/prebuilt/${HOST_ARCH}/bin  
export API_LEVEL=29  
export CC="${TOOLCHAIN_BIN}/aarch64-linux-android${API_LEVEL}-clang"  
export CXX="${TOOLCHAIN_BIN}/aarch64-linux-android${API_LEVEL}-clang++"  
export LD=${TOOLCHAIN_BIN}/aarch64-linux-android-ld  
export AR=${TOOLCHAIN_BIN}/aarch64-linux-android-ar  
export RANLIB=${TOOLCHAIN_BIN}/aarch64-linux-android-ranlib  
export STRIP=${TOOLCHAIN_BIN}/aarch64-linux-android-strip  
export BUILD_ROOT=$(dirname $(readlink -f "$0"))  
export PATH=${TOOLCHAIN_BIN}:${PATH}  
export PREFIX=${BUILD_ROOT}/toolchain/arm64-v8a  
export PKG_CONFIG_PATH=${PREFIX}/lib/pkgconfig  
export NCORES=$(getconf _NPROCESSORS_ONLN)
```



```
1:54  [status icons]  
64/" (ignoring)  
home usr  
.../com.termux/files $ cd home  
- $ termux-usb -l  
[]  
- $ termux-usb -l  
[  
  "/dev/bus/usb/001/002"  
]  
- $ termux-usb -r /dev/bus/usb/001/002  
Access denied.  
- $ termux-usb -r /dev/bus/usb/001/002  
Access granted.  
- $ termux-usb -e "./iio_info -S usb=0456:b673" /dev/bus/  
usb/001/002  
Tracking... iio_scan_context_get_info_list_android  
Tracking... usb_context_scan_android  
Available contexts:  
  0: 0456:b673 (Analog Devices Inc. PlutoSDR (ADAL  
M-PLUTO)), serial=1044730a1997000a05002700013f7ba113 [us  
b:1.2.5]  
- $ termux-usb -e "./iio_info -S usb" /dev/bus/usb/001/0  
02  
Tracking... iio_scan_context_get_info_list_android  
Tracking... usb_context_scan_android  
Available contexts:  
  0: 0456:b673 (Analog Devices Inc. PlutoSDR (ADAL  
M-PLUTO)), serial=1044730a1997000a05002700013f7ba113 [us  
b:1.2.5]  
- $ █  
ESC / - HOME ↑ END PGUP
```

To achieve this Goal we need to..

Easy for endpoints to collect data and join the ecosystem

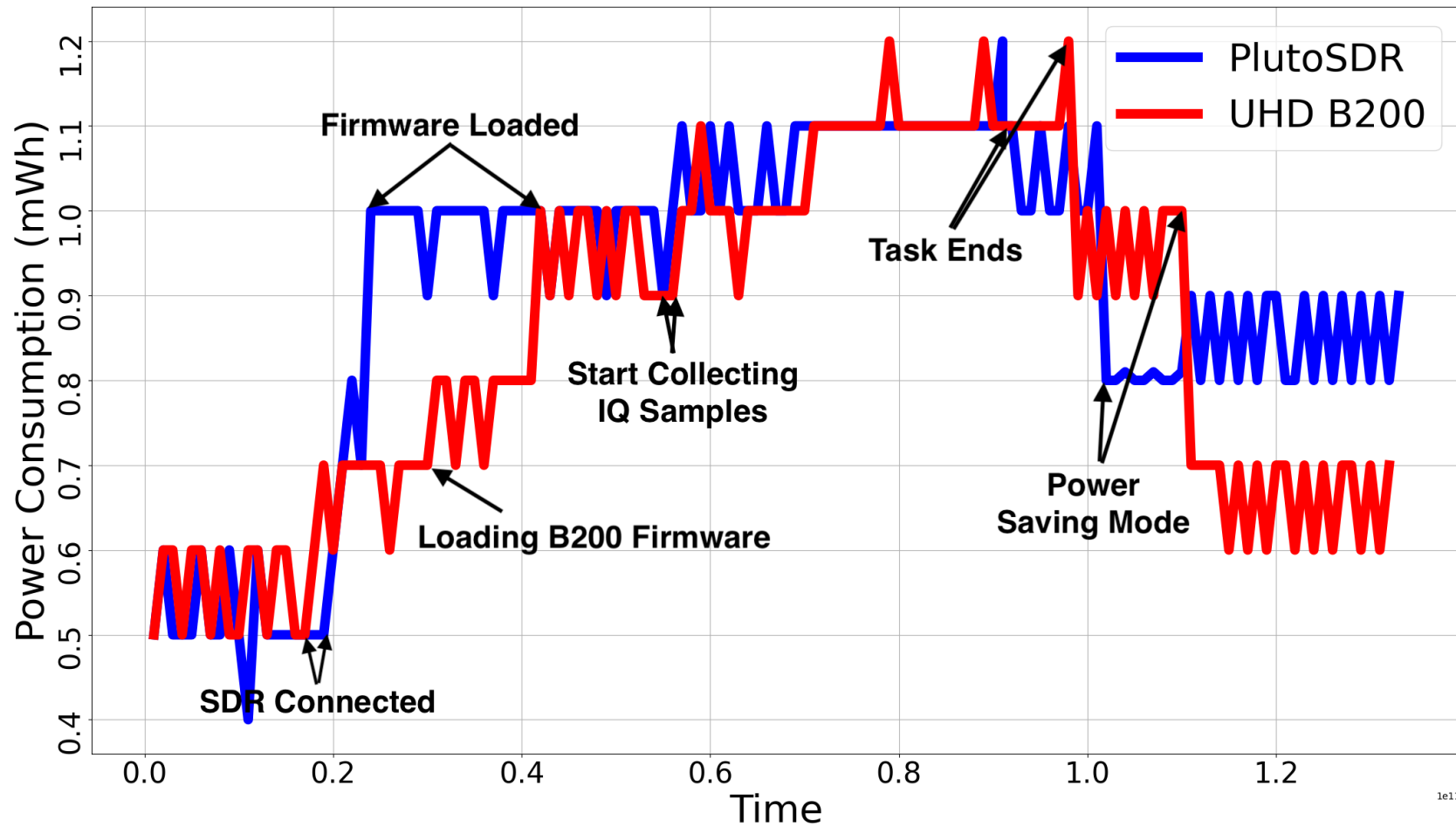
Ensure program security running on endpoint's device

Ensure data integrity

Save battery on the phone

Go through limited computation resource on the phone

Duty Cycle



To achieve this Goal we need to..

Easy for endpoints to collect data and join the ecosystem

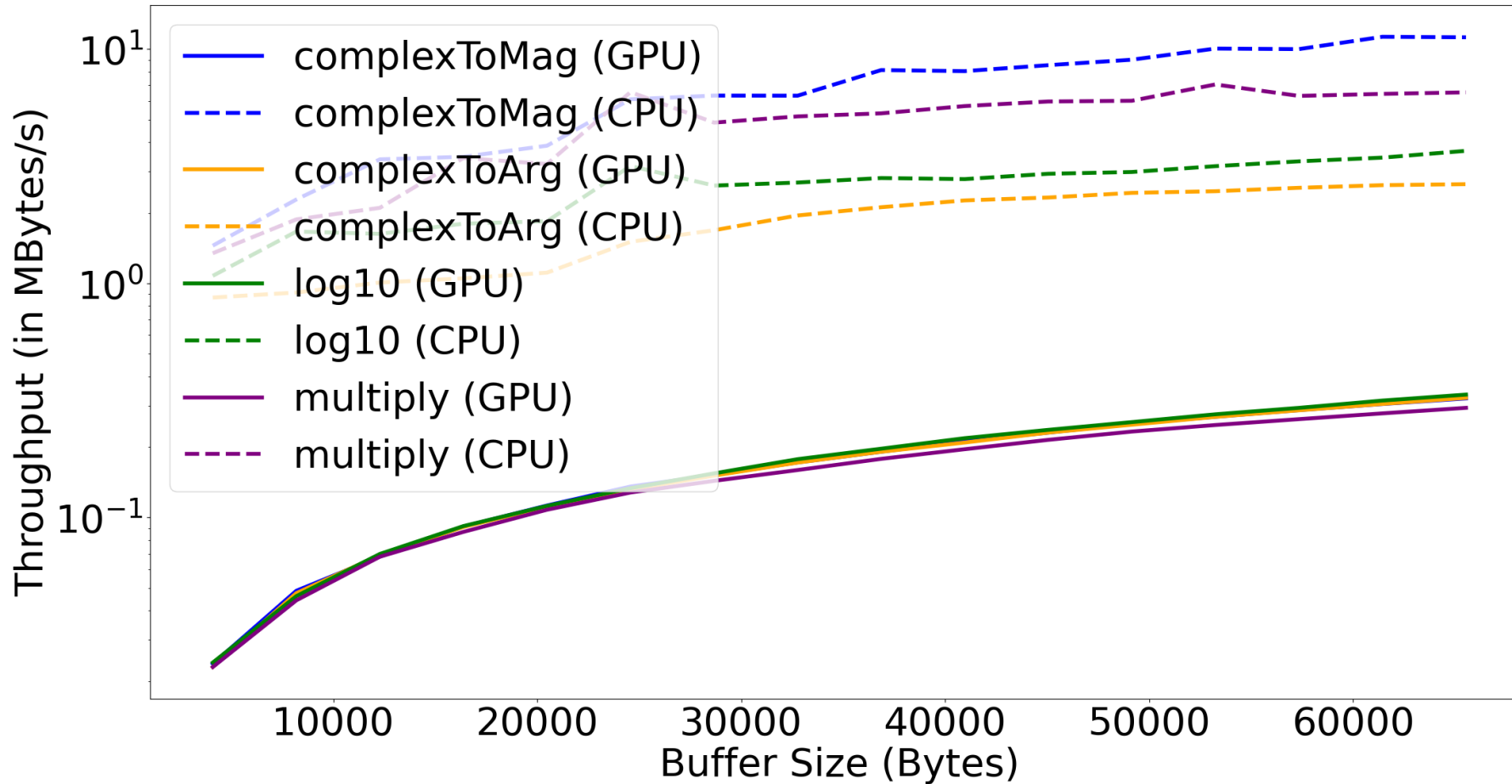
Ensure program security running on endpoint's device

Ensure data integrity

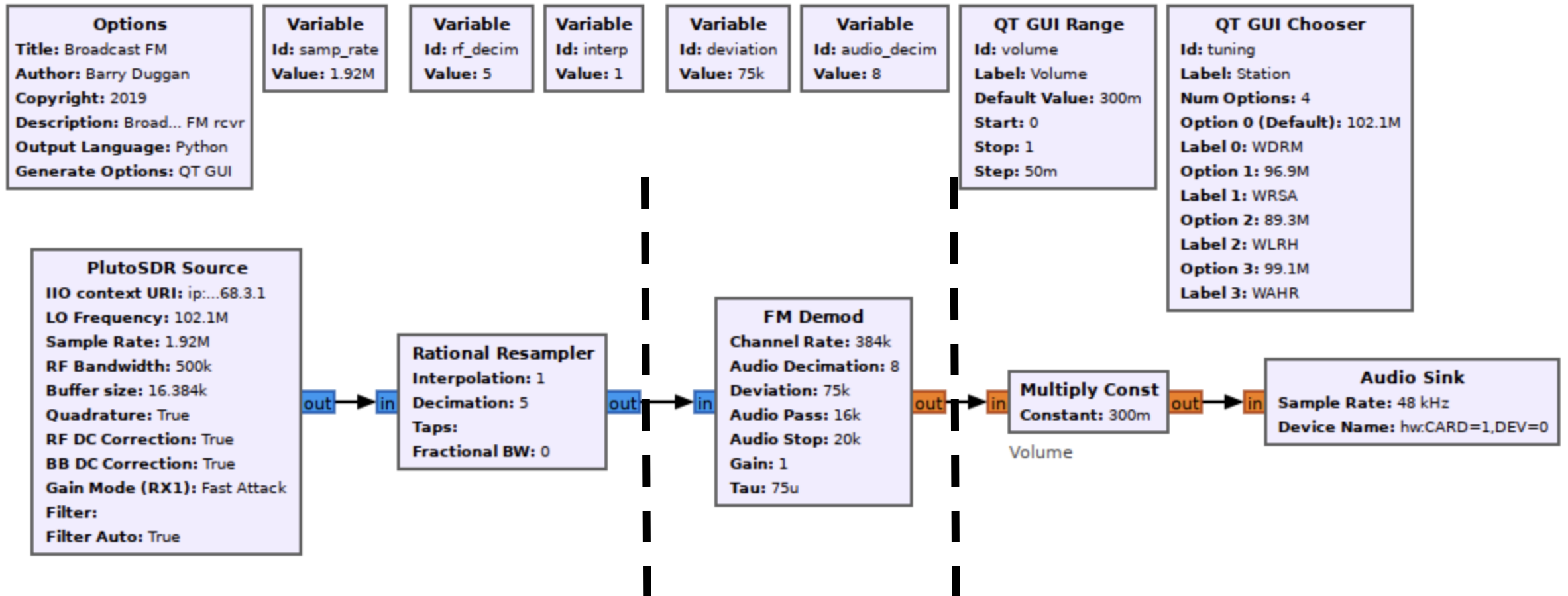
Save battery on the phone

Go through limited computation resource on the phone

Go through computation limitation on the phone



Load tasks on different devices



Load tasks on different devices

