10th UCAAT
User Conference on Advanced Automated Testing

ETSI
The Standards People

End to End Testing of Mobile Voice and Data Services in RAN Telecom Network

Alexandru-Răzvan Stamp

NOKIA

15/11/2023

What we do

End to end testing of mobile network software with the scope of creating problem reports:

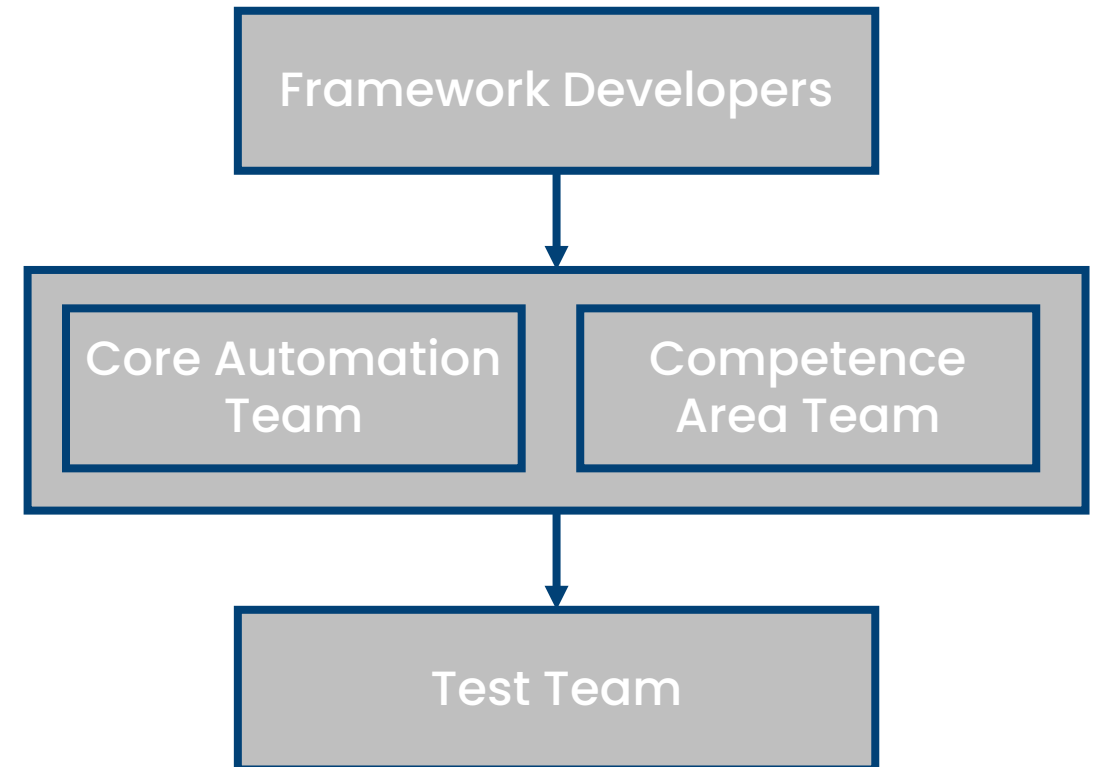- Operation, Administration and Maintenance
- Telecom

Mainly testing Base Transceiver Station (BTS) software in RF Sharing configuration (2 base transceiver stations or more can share the same radio module(s))

This presentation is covering testing of telecom features of the base transceiver station

Automation Team Organization

Implementing automated tests for regression and new feature testing for RFS Team

- Framework Developers
  - Develop libraries and tools that interface with equipment
- Automation Team
  - Develop automated tests using libraries and tools offered by framework developers
- Testers
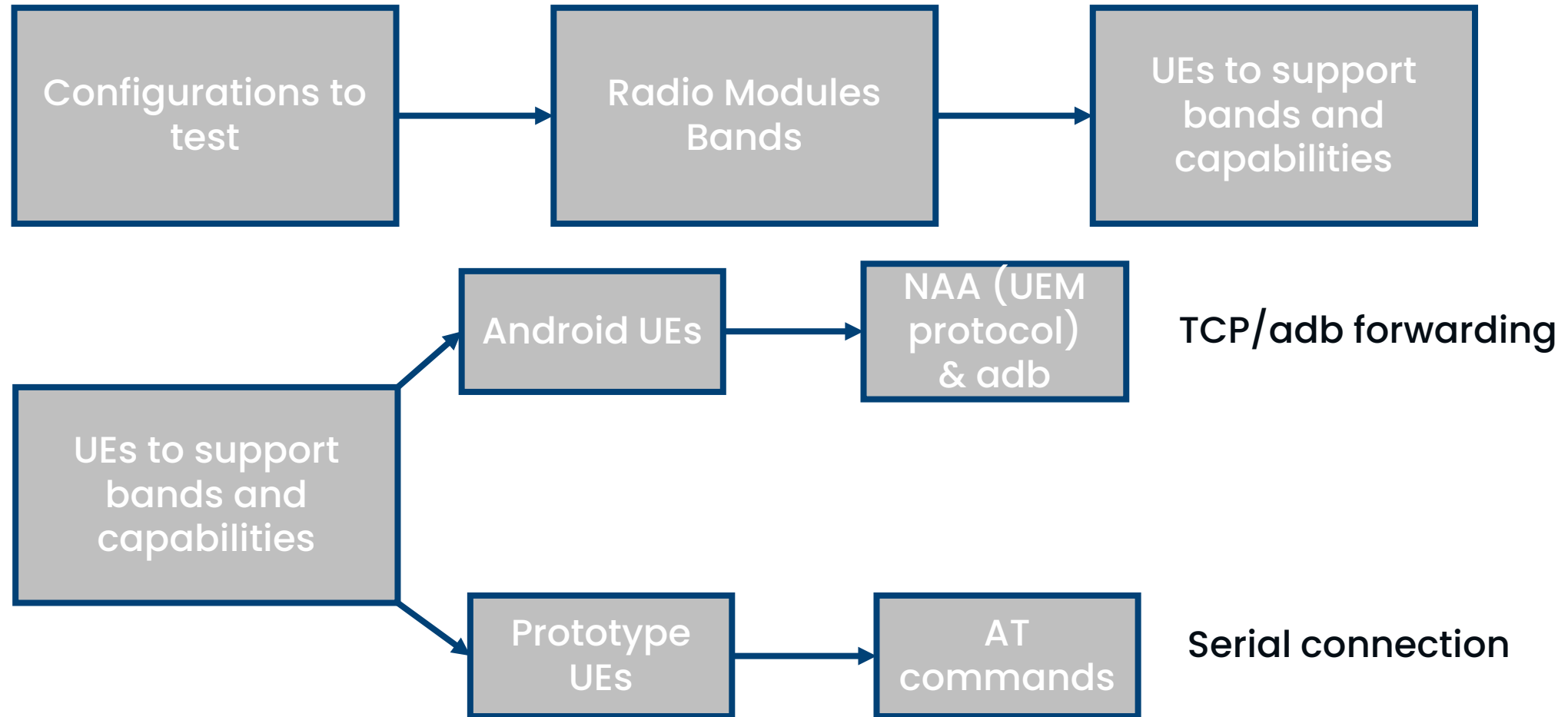 - Configure and execute automated tests developed by automation team

```
┌─────────────────────────────────┐
│     Framework Developers        │
└─────────────────────────────────┘
              │
              ▼
┌─────────────────────────────────┐
│ ┌──────────────┐ ┌────────────┐ │
│ │Core Automation│ │ Competence │ │
│ │     Team     │ │ Area Team  │ │
│ └──────────────┘ └────────────┘ │
└─────────────────────────────────┘
              │
              ▼
┌─────────────────────────────────┐
│           Test Team             │
└─────────────────────────────────┘
```

Equipment

We are testing multiple technologies using multiple user equipment (UE) that work in different frequency bands.

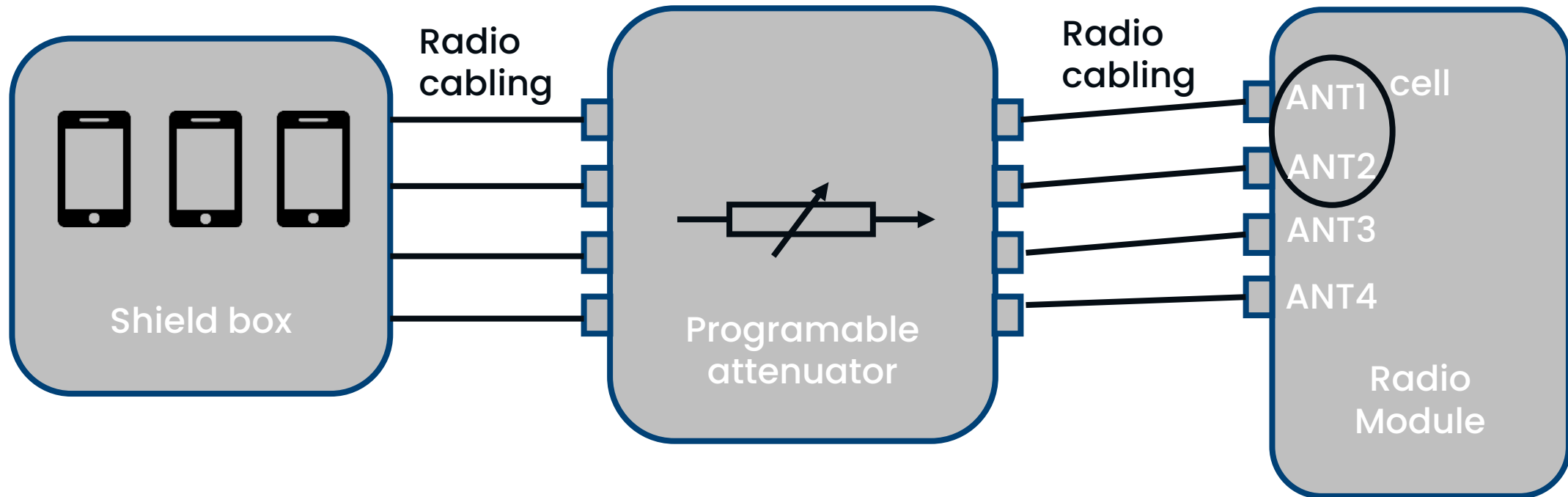| Technology	Service Tested
2G	Voice Call & Data Call
3G	Voice Call & Data Call
4G	Data Call
5G	Data Call

User Equipment used:

Laboratory Environment

Laboratory Environment

Radio cabling

Radio cabling

ANT1 cell

ANT2

ANT3

ANT4

Shield box

Programable attenuator

Radio Module

Lab environment emulating OTA signal propagation

Automation Environment

client

http

SSH

Linux virtual machine:

- Robotframework
- UE config
- Python

Proxy server

Windows PC:

- USB driver
- SSH server
- Python
- adb
- iperf

UEM protocol (TCP), adb / AT commands

USB

UE:
- UE Manager app (NAA)
- iperf

Radio module

Radio cabling

Radio Network

App Server:
- iperf
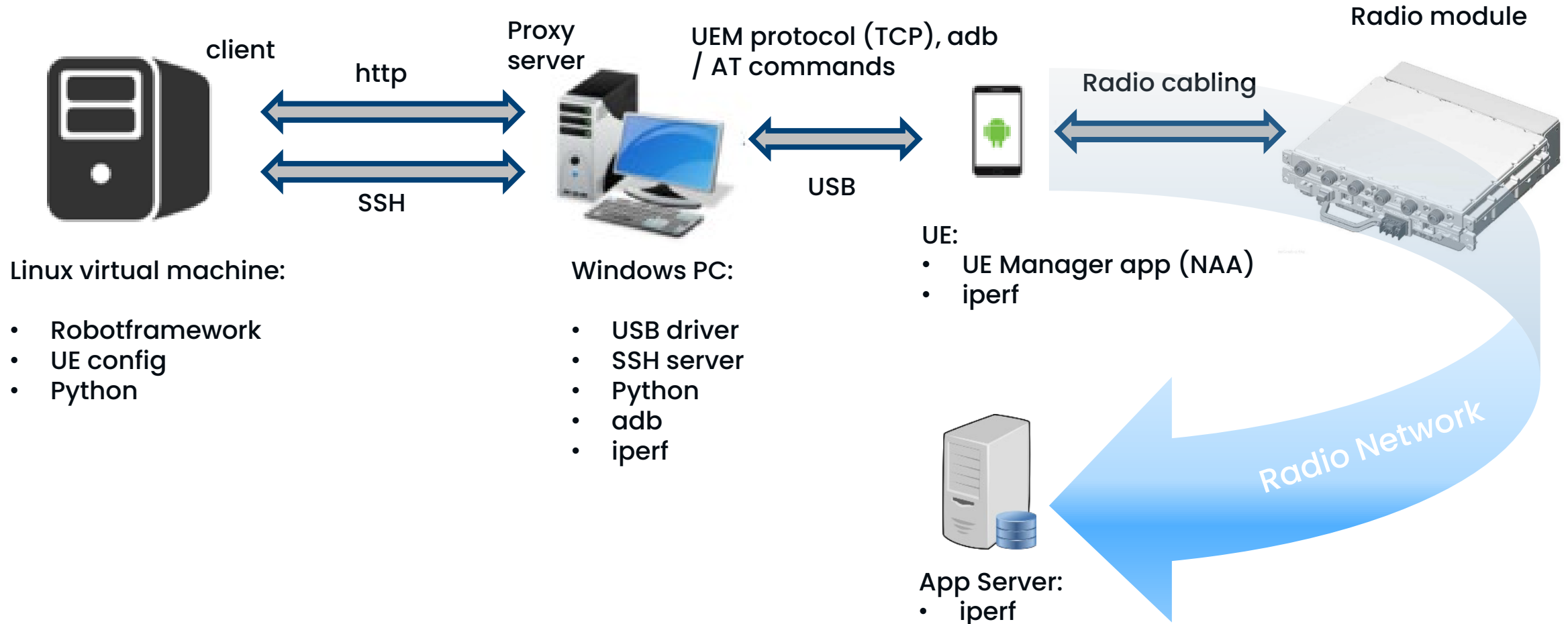
Configuration files

Configuration information is stored under a Python defined object model.

- Mobiles are split into two classes
    - Android based mobile
    - AT based mobile

- Application server class

Each class defines different attributes that we use in the automated tests.

```python
class AndroidUE(UE):
    def __init__(self,
                 serial_number,
                 ue_pc_address,
                 data_transfer=None,
                 ssh_connection_details=None,
                 imei=None,
                 imsi=None,
                 capabilities=None,
                 class_name=PluginAndroidClassType.android,
                 ):
        # type: (...) -> None


class ATUE(UE):
    def __init__(self,
                 com_port,
                 ue_pc_address,
                 data_transfer=None,
                 ssh_connection_details=None,
                 imei=None,
                 imsi=None,
                 capabilities=None,
                 class_name=PluginAndroidClassType.at,
                 ):
        # type: (...) -> None
```

Call Test Flowchart
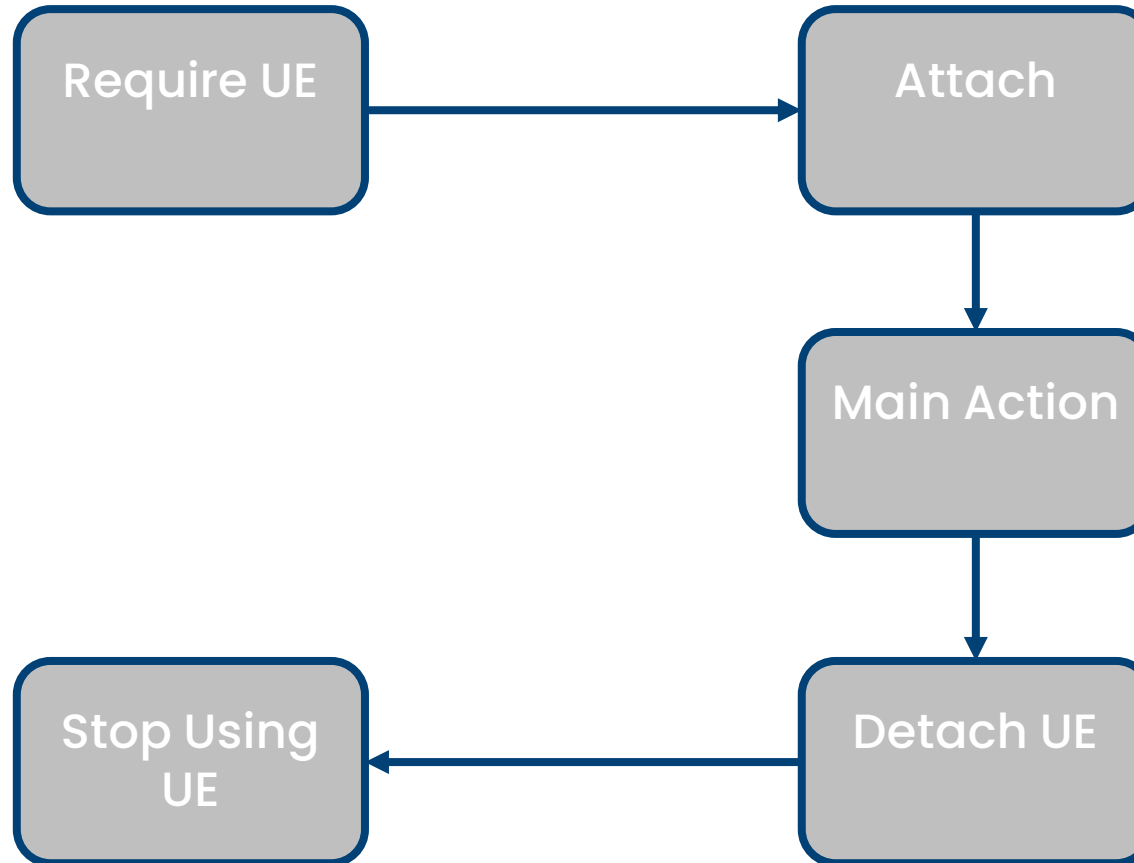
Require & Attach

Require UE:

Input:
- ue alias
- ue capability

Require UE

Search UE that fulfills capability and it's reachable

UE pool from config

_ _ _UE1
_ _ _UE2
_ _ _UE3
...

UE found? — YES → Continue test with selected UE

NO

*Search **another** UE that fulfills capability and it's reachable*

UE pool search completed? — YES → FAILURE no UE available

NO

Attach UE:

Input:
- ue alias
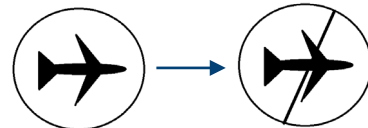
Cell ID

Inputs:
- ue alias
- info(e.g. CELL_ID, IP)

```
Attach UE → Get UE Info → Check if CELL_ID is the expected one
```

CELL_ID   IP

Check if CELL_ID is the expected one → YES → Continue TC actions

Check if CELL_ID is the expected one → NO → Corrective action: reboot UE

Reinitiate attach procedure
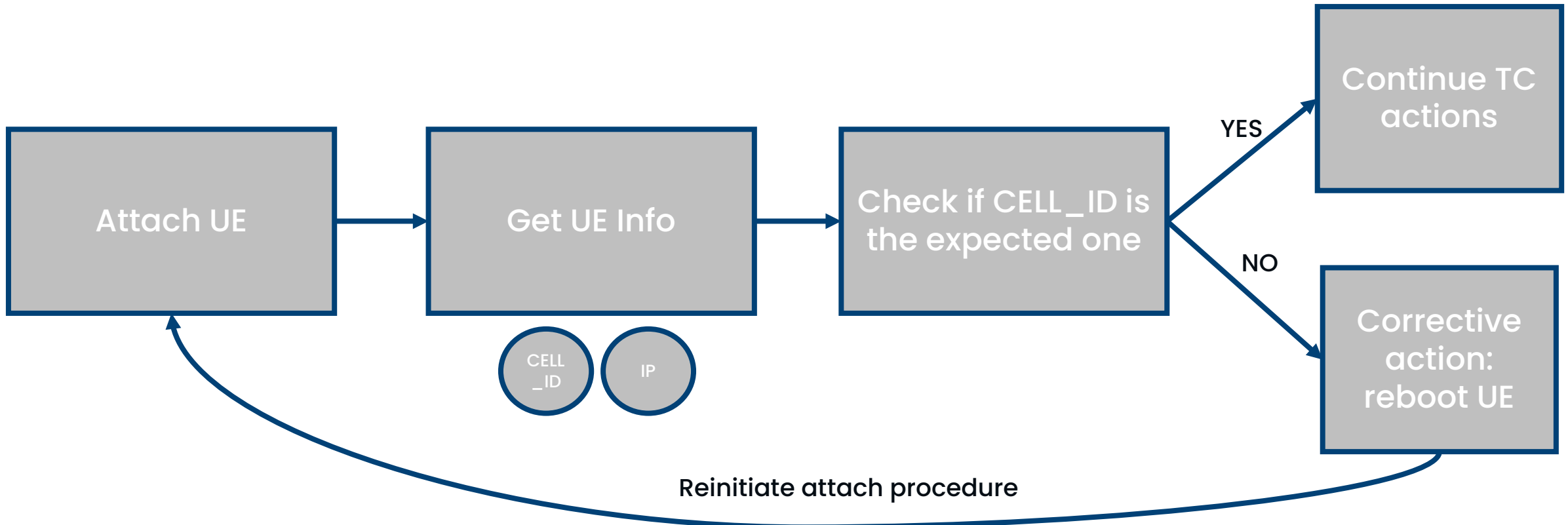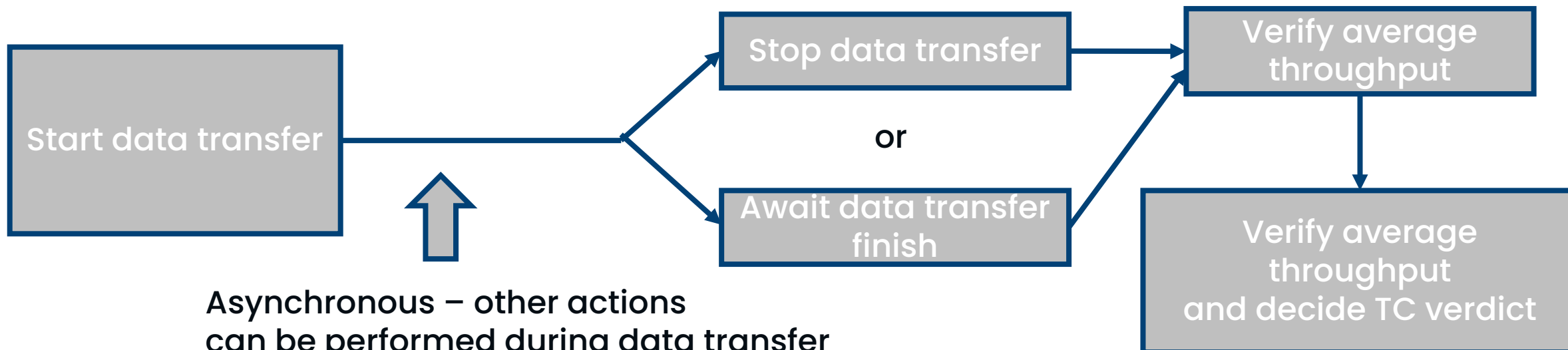
Data Transfer

Inputs:
- ue alias
- dt alias
- direction (UL, DL or both)
- duration
- protocol (UDP/TCP/ICMP etc)

Input:
- dt alias

Inputs:
- dt alias
- min value accepted
- max value accepted

```
Start data transfer    ──→    Stop data transfer    ──→    Verify average throughput
                                      or                              │
                               Await data transfer                    ↓
                               finish          ──→    Verify average throughput
                                                       and decide TC verdict
```

Asynchronous – other actions can be performed during data transfer session

Voice Call

Inputs:
- ue alias
- vc alias
- duration
- to (destination number)

Input:
- vc alias

```
Start voice call  ──────────┐
                            ├──→ Stop voice call ──────→ Decide TC verdict
                            │         or                 passed if voice call
                            └──→ Await voice call ─────→  was not dropped
                                     finish
```
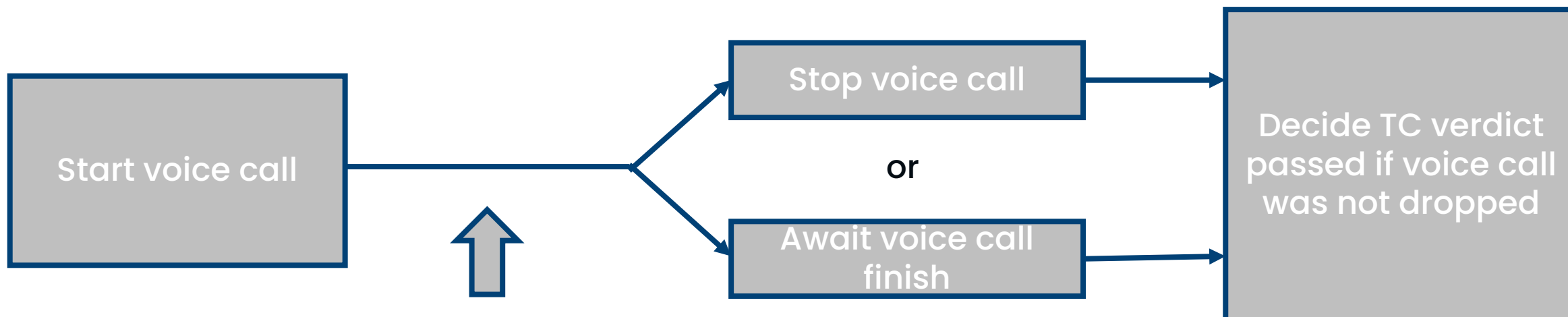
Asynchronous – other actions can be performed during voice call session

Detach & Stop Using UE

Detach UE:

Input:
- ue alias

No further actions can be done with the mobile after detach

Stop Using UE:

Input:
- ue alias

Frees all allocated resources. Can be reused from the mobile pool

Mobility Testing

Select cell(s) by using a programable attenuator (by attenuating unwanted cells).

- Allows testing on multiple cells
- Allows simulation of cell changing

Executed using a library that allows control of multiple programmable attenuators.

Test verdict will be passed if:

- Throughput values are in the selected interval
- Voice call did not drop
- Call performed on desired cell

Logs used for manual investigation of failed cases:

- Robot Logs
- Library Logs
- BTS Logs

Any further questions?

alexandru.stamp@nokia.com
gabriel.stanculescu@nokia.com