Presentation Script: Integration of Wi-Fi-Only Devices in 5G

Instructions for Training:

- Time yourself: Read through the script at a natural, comfortable pace. Use a stopwatch to see if you're hitting the approximate timings.
- Don't just read: Use the script to understand the key points for each slide. The goal is to speak knowledgeably, not to read from a paper.
- Emphasize key terms: The words in bold are concepts you should stress.
- Pause: The notes like `[pause briefly]` are suggestions for letting a concept sink in before moving on.

Slide 1: Title Slide

(Time: ~30 seconds)

"Good morning/afternoon, everyone. My name is David Araújo, and today I will be presenting my Master's dissertation, titled Integration of Wi-Fi-Only Devices in 5G Core Networks: Addressing Authentication and Identity Management Challenges.

This work was conducted at the University of Aveiro, in collaboration with Altice Labs and the Instituto de Telecomunicações, under the supervision of Dr. Daniel Corujo and Dr. Francisco Fontes."

Slide 2: The Core Problem and Its Significance

(Time: ~1 minute)

The core challenge this dissertation tackles is a significant gap in the current 5G ecosystem. As it stands, Wi-Fi-only devices—meaning those without 5G credentials like a USIM os SIM card — cannot be directly integrated into the 5G Core network using standard methods as they are not capable of communicating and thus, authenticating into the network. This is a major hurdle in today's enterprise and residential environments, where countless devices rely solely on Wi-Fi like IoT.

As 5G continues to expand, this lack of a seamless integration path becomes a critical bottleneck. Solving this problem is essential for achieving true 5G-Wi-Fi convergence and for extending the powerful benefits of 5G—like enhanced mobile broadband and massive IoT support—to this huge ecosystem of legacy and resource-constrained devices.

Slide 3: Research Objectives

(Time: ~1 minute)

To address this challenge, this research was guided by three main objectives:

1. First, to design a secure and robust local authentication mechanism that doesn't depend on 5G credentials.
2. Second, to develop a method for the 5G Core to recognize and, crucially, individually manage each Wi-Fi-only device's connection.
3. And third, to combine these into an integrated solution that requires minimal impact on both the existing 5G architecture and the end devices themselves.

Slide 4: State of the Art - The Gap

(Time: ~1.5 minutes)

To understand the context, it's important to differentiate between two types of non-3GPP devices. There are N5GC devices, which have some limited 5G capabilities and can authenticate. But our focus is on NAUN3 devices — Non-Authenticable Non-3GPP devices. These have no native 5G capabilities and cannot be directly authenticated.

Typically, these devices are simply grouped together behind a residential gateway, which prevents any form of individual management. The central gap this research targets is the lack of a robust mechanism for providing secure, per-device authentication and management for these NUN3 devices.

Slide 5: State of the Art - Managing Device Groups (CGID)

(Time: ~1.5 minutes)

One existing 3GPP concept is the Connectivity Group ID, or CGID. As you can see in the diagram, this allows a group of devices—for example, everything connected to a specific Wi-Fi SSID—to share a single PDU session.

While this provides connectivity, it's a blunt instrument. It offers no per-device traffic granularity. You can't apply a specific security policy to just the printer, or prioritize bandwidth for the television.

More recent developments in 3GPP Release 19 are moving towards enabling per-device traffic distinction. My research anticipates this direction and provides a working, validated proof-of-concept that demonstrates how to achieve this today.

Slides 6 & 7: Framework Concept and Architecture

(Time: ~2 minutes)

So, how did we solve this? Our framework is centered around the concept of a smart 5G Residential Gateway. The key principle here is local intelligence—we placed all the adaptation logic at the network edge, within the 5G-RG itself. This approach ensures minimal impact on the end devices, which require no changes, and minimal disruption to the 5G Core, which sees standard interactions.

[Click to Slide 7]

This slide shows the high-level architecture. On the right, we have the Wi-Fi-only NAUN3 device. On the left, the operator's ISP network, containing the 5G Core and an EAP Authentication Server. In the middle is our

intelligent 5G-RG.

The flow is simple:

1. The NAUN3 device connects to the RG and is authenticated locally via the EAP server.
2. Once authenticated, the RG establishes a unique PDU session for that device.

All traffic from that device is then routed through its dedicated PDU session, making it a distinct, manageable entity within the 5G network.

Slide 8: Authentication Mechanism

(Time: ~1.5 minutes)

For the crucial first step of local authentication, we chose EAP-TLS. This is a highly secure, mutual authentication method based on digital certificates.

The NAUN3 device acts as the Supplicant and holds a client certificate. The 5G-RG, running hostapd, acts as the Authenticator, but more accurately as a relay—it forwards EAP messages inside of RADIUS packets to the Authentication Server. This server, operated by the ISP, is what actually validates the device's certificate and grants or denies access.

This setup provides strong, zero-trust style security without relying on 5G credentials, using standard, well-understood enterprise protocols.

Slide 9: Identity Management (PDU Session as Proxy)

(Time: ~2 minutes)

This brings us to the core innovation of my work. Once a device is locally authenticated, it still needs an identity in the 5G Core. How do we create one?

We do this by having the 5G-RG establish a new, dedicated PDU session and using it as a proxy identity for the NAUN3 device.

As shown in this flow, after a successful EAP authentication, my custom-developed Interceptor application on the RG automatically requests a new PDU session from the 5G Core. The RG maintains a local mapping between the device’s MAC address and its assigned PDU Session ID. This effectively gives a Wi-Fi-only device a unique, manageable 5G identity, using entirely standard 5G mechanisms.

Slide 10: Traffic Management and Policy-Based Routing

(Time: ~1.5 minutes)

With a proxy identity established, we need to ensure traffic is routed correctly. We achieve this with a dynamic, policy-based routing approach, which is also automated by the Interceptor.

First, every incoming packet from the device's MAC address is marked with a unique identifier.

A policy rule then directs these marked packets to a dedicated routing table.

This table contains a single default route, pointing all traffic out through the correct PDU session's virtual interface.

Finally, NAT is applied, so the traffic appears to the outside world as originating from the 5G-Core-assigned IP address.

This ensures complete traffic segregation and allows the 5G Core to manage the flow as a distinct session.

Slides 11 & 12: Testbed and Interceptor

(Time: ~1 minute)

To validate this framework, we built a fully virtualized testbed using Vagrant, Open5GS, and UERANSIM. The environment simulated everything from the end device to the 5G Core.

[Click to Slide 12]

The brain of the solution is the custom Interceptor application I developed. This Go program is the central orchestrator. It monitors hostapd for successful authentications, triggers the creation of new PDU sessions via UERANSIM's command-line tool, configures all the necessary DHCP and routing rules, and, just as importantly, cleans everything up when the device disconnects.

Slide 13: Validation - Successful Onboarding

(Time: ~1 minute)

Our validation tests confirmed the framework functions as designed. First, we verified successful onboarding. Local EAP-TLS authentication was consistently successful. As you can see in the log snippet, this immediately triggered the creation of a unique PDU session on the 'clients' data network, and the 5G Core assigned a unique IP address to that session.

Slide 14: Validation - Connectivity and Isolation

(Time: ~1 minute)

Next, we confirmed end-to-end connectivity and traffic isolation. Using the ping -R command, which records the route of the packet, we could see definitive proof of our mechanism.

The top example shows a device's traffic being routed through the proxy IP 10.46.0.2. The bottom example shows a second device's traffic being routed through a different proxy IP, 10.46.0.3. This proves that traffic is correctly mapped and completely isolated between devices.

Slide 15: Validation - Lifecycle Management

(Time: ~1 minute)

Finally, we validated the full resource lifecycle. When a device disconnected, our Interceptor correctly detected the event, deauthenticated the client, purged all of its associated routing rules and DHCP permissions, and terminated the dedicated PDU session in the 5G Core. This ensures that resources are managed efficiently.

We also measured the onboarding delay. In our proof-of-concept environment, this process took an average of approximately 33 seconds.

Slide 16: Key Contributions

(Time: ~1 minute)

To summarize, my dissertation makes four key contributions:

- It provides a practical, end-to-end framework for integrating these 5G-credential-less devices.
- It introduces the innovative use of per-device PDU Sessions as dynamic proxy identities.
- It demonstrates the tight and practical coupling of strong, local EAP-TLS authentication with 5G session management at the network edge.
- And finally, it delivers a working, validated proof-of-concept, built with open-source tools and custom logic."

Slide 17: Limitations

(Time: ~1 minute)

Of course, this research has some limitations. The proof-of-concept relies on a CLI-based orchestration because of UERANSIM, which contributes to the 33-second onboarding delay. The use of NAT inherently restricts inbound connections to the devices. And finally, our attempts at physical hardware integration were challenging due to the proprietary nature and lack of documentation for the experimental 5G modem we used.

Slide 18: Future Work

(Time: ~1 minute)

These limitations point directly to future work. The next steps would be to adapt the Interceptor to use native modem APIs, like AT commands or QMI, which would drastically reduce the onboarding delay. We could also explore high-performance routing mechanisms like eBPF for scalability, and investigate using the Framed-Route RADIUS attribute to address the NAT limitations. These steps would bring this framework even closer to production-readiness.

Slide 19: Thank You and Q&A

(Time: ~30 seconds)

Thank you for your attention. I am now happy to answer any questions you may have.