

# Sachbericht zum Verwendungsnachweis Teil II 2025

## Verbundvorhaben

### MANTRA5G

#### Modular, Adaptive and iNteroperable Test fRAmework for 5G/6G (MANTRA5G)

Gefördert durch das Bundesamt für Sicherheit in der Informationstechnik (BSI)

Konsortialführung: <b>TÜV Informationstechnik GmbH, Am TÜV 1, 45307 Essen</b>	Förderkennzeichen: <b>01MO23026A</b>
Laufzeit des Vorhabens: von: <b>01.06.2023</b> bis: <b>30.11.2024</b>	
Berichtszeitraum: von: <b>01.06.2023</b> bis: <b>30.11.2024</b>	Datum: <b>28.05.2025</b>

#### Projektpartner:

1. Exceeding Solutions GmbH, Otto-Eißfeld-Straße 4, 06120 Halle (Saale)

## Inhaltsverzeichnis

<b>1</b>	<b>Aufgabenstellung</b>	<b>4</b>
<b>1.1</b>	<b>Kurzdarstellung des durchgeführten Vorhabens</b>	<b>4</b>
1.1.1	Rahmendaten zum Vorhaben	4
1.1.2	Grundlegende Idee und Motivation	4
1.1.3	Zielstellung des Vorhabens	4
1.1.4	Projektplan der ursprünglichen Vorhabensbeschreibung	5
<b>1.2</b>	<b>Eingehende Darstellung der durchgeführten Aufgaben</b>	<b>6</b>
1.2.1	Kickoff-Meeting (14.06.23)	6
1.2.2	Anforderungsanalyse (AP2)	6
1.2.3	Identifikation von Open Source Tools zur Testautomatisierung (AP3)	7
1.2.4	Meilenstein-Meeting (23.11.23)	8
1.2.5	Identifikation von Tools zur Nachbildung von 5G Funktionalität (AP3)	8
1.2.6	Hintergrundanalyse zu konkreten Testabläufen (AP4)	9
1.2.7	Einschätzung der Tools durch praktisches Aufsetzen (AP5)	10
1.2.8	Entwurf einer Zwischenschicht API (AP6)	10
1.2.9	Entwicklung und Präsentation eines MANTRA5G Demonstrators (AP7)	10
<b>1.3</b>	<b>Eingehende Darstellung der erzielten Ergebnisse</b>	<b>11</b>
1.3.1	Anforderungs-Checkliste	11
1.3.2	5G Recherche und Testfallentwicklung	13
1.3.3	Testsetup und Einschätzung der Open-Source Tools	15
1.3.4	Aufbau der Systemlandschaft	17
1.3.5	Entwicklung des Demonstrators	17
<b>1.4</b>	<b>Einschätzung des durchgeführten Vorhabens</b>	<b>18</b>
1.4.1	Vergleich zur ursprünglichen Vorhabensbeschreibung	18
1.4.2	Fazit und Ausblick	19
<b>2</b>	<b>Die wichtigsten Positionen des zahlenmäßigen Nachweises</b>	<b>19</b>
<b>2.1</b>	<b>TÜVIT</b>	<b>19</b>
<b>2.2</b>	<b>Exceeding Solutions</b>	<b>20</b>
<b>3</b>	<b>Notwendigkeit und Angemessenheit der geleisteten Arbeit</b>	<b>21</b>
<b>4</b>	<b>Voraussichtlicher Nutzen und Verwertbarkeit</b>	<b>21</b>
<b>4.1</b>	<b>TÜVIT</b>	<b>21</b>
<b>4.2</b>	<b>Exceeding Solutions</b>	<b>21</b>

<b>5</b>	<b>Fortschritt bei anderen Stellen</b>	<b>22</b>
<b>6</b>	<b>Erfolge und geplante Veröffentlichungen</b>	<b>22</b>

# 1 Aufgabenstellung

## 1.1 Kurzdarstellung des durchgeführten Vorhabens

Das folgende Kapitel stellt einen Bezug zur eingereichten Vorhabensbeschreibung dar und fasst die wesentlichen Ziele des Vorhabens zusammen:

### 1.1.1 Rahmendaten zum Vorhaben

Das diesem Bericht zu Grunde liegende Vorhaben MANTRA5G wurde im Rahmen des Förderaufrufs „Cybersicherheit und digitale Souveränität in den Kommunikationstechnologien 5G/6G“ durch das BSI und dem Kennzeichen 01MO23026A gefördert. Das Vorhaben wurde vom 01.06.2023 bis 30.11.2024 bearbeitet.

Der Projektverbund bestand aus zwei Partnern:

- TÜVIT: als Prüfdienstleistern für IT-Produkte und Systeme auf Basis national und international anerkannter Standards und Kriterien und seit Juli 2022 erste anerkannte Prüfstelle für das Zertifizierungsschema NESAS CCS-GI
- Exceeding Solutions: aktiv in der Entwicklung von Testkomponenten, einziger Anbieter einer vollumfänglichen Testmaschine für intelligente Messsysteme (Smart Meter Gateway) und aktuell im Prozess, den Geschäftsbereich um 5G Prüftechnik zu erweitern

### 1.1.2 Grundlegende Idee und Motivation

5G als Schlüsseltechnologie der digitalen Transformation birgt sowohl Chancen als auch Risiken. Der vermehrte Einsatz in kritischen Infrastrukturen und die starke Verbreitung von Mobilfunk erhöht die Fläche für mögliche Angriffe und die deren Auswirkungen. Um die Sicherheit von Netzen und Diensten zu erhöhen und gewährleisten, sind Netzbetreiber gesetzlich verpflichtet, Sicherheitsmaßnahmen zu implementieren. Ab 2026 dürfen kritische Komponenten nur dann eingesetzt werden, wenn sie zuvor geprüft und zertifiziert wurden. Das für kritische Komponenten des 5G Kernnetzes vorgesehene Zertifizierungsschema in Deutschland ist NESAS CCS-GI. Hierbei wird die Einhaltung geforderter Sicherheitseigenschaften durch standardisierte Sicherheitstests geprüft. Prüf- und Zertifizierungsstellen in Deutschland erwarten eine Vielzahl an Prüfungen. Zum Zeitpunkt des Projektstarts existierten keine vollumfänglichen Testtools, die eine effiziente Prüfung nach NESAS CCS-GI ermöglichen.

### 1.1.3 Zielstellung des Vorhabens

Im Rahmen des durchgeführten Vorhabens wurde die Frage untersucht, wie ein geeignetes Testsystem für Prüfungen nach NESAS CCS-GI aussehen könnte. Im Detail wurde analysiert, welche Anforderungen an ein NESAS CCS-GI Testsystem gelten, wie eine Automatisierung bzw. Teilautomatisierung der Testabläufe erreicht werden kann, welche Tools im Open-Source existieren und welche Tools sich für den Einsatz im Testsystem eignen.

Das angestrebte Ergebnis war die Spezifikation eines modularen, adaptiven, interoperablen Testframeworks für die Prüfung von 5G Mobilfunkkomponenten nach NESAS und NESAS CCS-GI sowie die Implementierung eines Demonstrators basierend auf Open-Source Lösungen.

### 1.1.4 Projektplan der ursprünglichen Vorhabensbeschreibung

Der zu Projektbeginn vorgestellte Projektplan für das Vorhaben MANTRA5G gliedert sich in 7 Arbeitspakete, deren zeitlicher Verlauf in Abbildung 1 dargestellt ist.

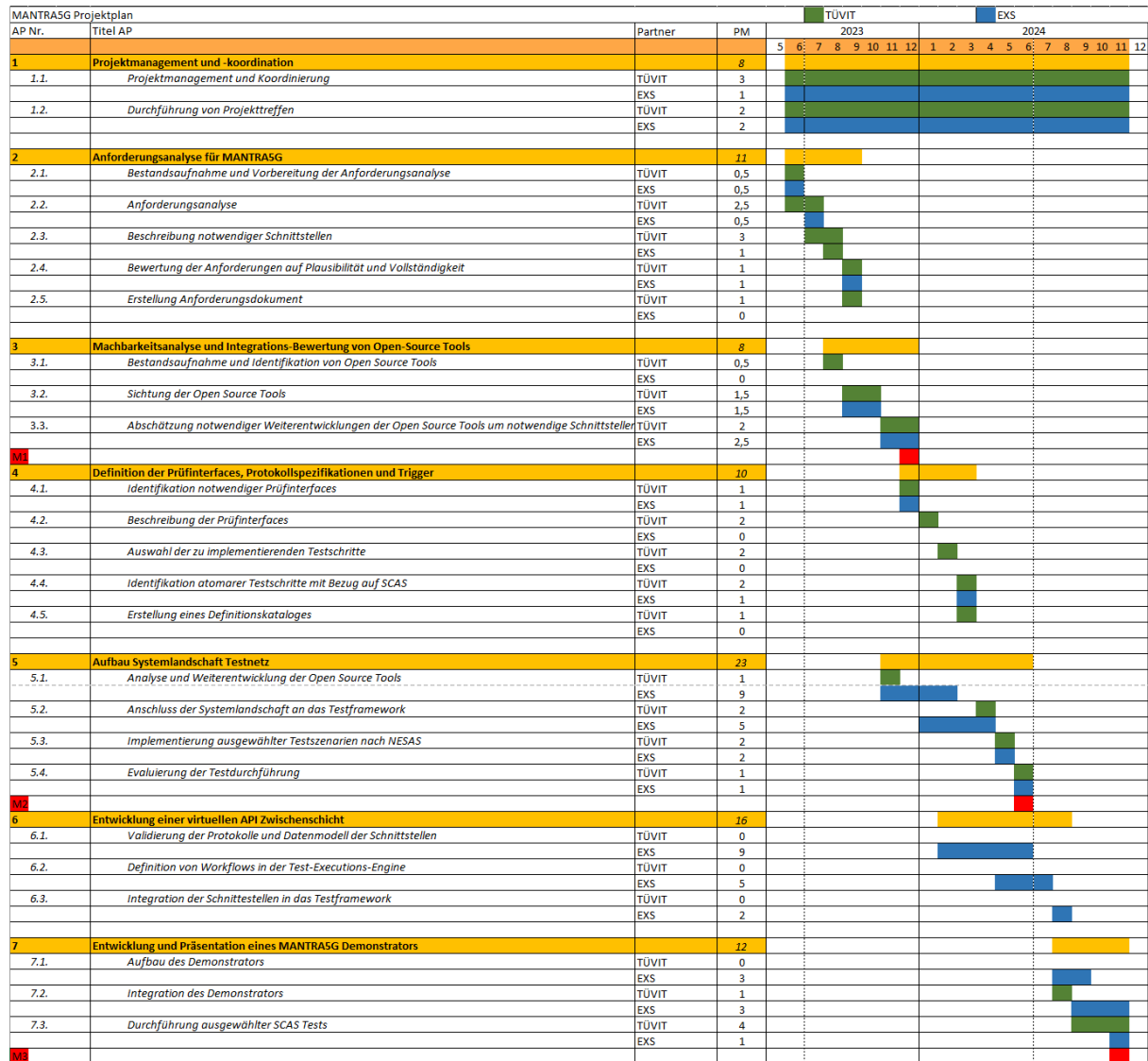


Abbildung 1: Projektplan aus der Vorhabensbeschreibung

Im Rahmen des Vorhabens waren die folgenden Meilensteine vorgesehen

Meilenstein	Monat	Beschreibung
<b>M1</b>	7	Anforderungsanalyse und Machbarkeitsanalyse abgeschlossen und erstes Spezifikationsdokument erstellt
<b>M2</b>	13	Definition der Prüferinterfaces sowie Aufbau der Systemlandschaft abgeschlossen
<b>M3</b>	18	Lauffähiger Demonstrator für MANTRA5G

## 1.2 Eingehende Darstellung der durchgeführten Aufgaben

In den folgenden Abschnitten werden die im Rahmen des Projektes durchgeführten Arbeiten in zeitlicher Reihenfolge ausführlich dargestellt:

### 1.2.1 Kickoff-Meeting (14.06.23)

Zum Projektstart wurde ein Kickoff-Meeting durchgeführt. Dabei konnten sich alle Projektbeteiligten kennenlernen und den fachlichen Austausch beginnen. Im Rahmen des Vor-Ort Treffens wurden die allgemeinen Projektziele sowie das Vorgehen anhand einer vorbereiteten Präsentation besprochen. Außerdem wurden die konkreten Meilensteine festgelegt und die Zeitplanung abgestimmt.

### 1.2.2 Anforderungsanalyse (AP2)

Um Umfang und Funktion der MANTRA5G Toolbox planen zu können, wurde in Q3 2023 mit einer Anforderungsanalyse begonnen. Das Framework soll in der Lage sein, effektive Prüfungen nach anerkannten Prüfschemata umzusetzen. Für die Prüfung von 5G Komponenten sind das internationale Prüfschema NESAS der GSMA und das nationale Prüfschema NESAS CCS-GI vorgesehen. Um im Rahmen der Tool-Spezifikation alle Anforderungen aus den genannten Schemata abzudecken, wurden zunächst die Verfahrensdokumente systematisch analysiert. Konkret wurden die Dokumente der GSMA<sup>1</sup> sowie die Dokumente des BSI<sup>2</sup> betrachtet.

Die Verfahrensdokumente beinhalten Informationen zu allgemeinen Abläufen und Voraussetzungen für eine NESAS bzw. NESAS CCS-GI Prüfung. Aus den Dokumenten wurden Use-Cases und Anforderungen abgeleitet und dokumentiert. Der Fokus hierbei lag vor allem auf der gewünschten Bedienung des zu entwickelnden Frameworks (z.B. vorgesehene Nutzerrollen) und der technischen Rahmenbedingungen, um die Vorgaben der Prüfschemata zu erfüllen (z.B. Reproduzierbarkeit von Testergebnissen).

Die TÜV Informationstechnik war bei Einführung des neuen Prüfschemas an der Pilotierung von NESAS CCS-GI beteiligt. Im Rahmen der Pilotphase wurden alle nach NESAS CCS-GI vorgesehenen Prozesse durchlaufen und eine vollständige Prüfung einer 5G Basisstation durchgeführt. Daher konnte hier auf breites, praxisbezogenes Wissen zurückgegriffen werden. Die bereits notierten Anforderungen an das zu entwickelnde Testsystem wurden um praktische Anwendungsfälle und Erkenntnisse aus der Pilotphase erweitert.

Neben den klar definierten Prozessen und Prüfabläufen gibt NESAS auch vor, welche Tests für die unterschiedlichen Prüfkomponten anwendbar sind. Die Tests sind in den SCAS-Dokumenten der 3GPP beschrieben. Diese teilen sich auf in einem generischen Testkatalog [TS 33.117]<sup>3</sup>, der auf alle Komponenten angewendet werden muss und Produktklassenspezifische Testkataloge, die nur angewendet werden, wenn die zu prüfende Komponente die entsprechende Funktion umsetzt.

---

<sup>1</sup> <https://www.gsma.com/solutions-and-impact/industry-services/assurance-services/nesas-documents/>

<sup>2</sup> [https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Zertifizierung-und-Anerkennung/Zertifizierung-von-Produkten/Zertifizierung-nach-NESAS/Dokumente/Dokumente\\_node.html](https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Zertifizierung-und-Anerkennung/Zertifizierung-von-Produkten/Zertifizierung-nach-NESAS/Dokumente/Dokumente_node.html)

<sup>3</sup>

<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2928>

Durch Analyse der unterschiedlichen SCAS-Dokumente, insbesondere der Test-Vorbedingungen und einzelnen Testschritte, konnten weitere Anforderungen an die MANTRA 5G Toolbox zusammengetragen werden. Der Fokus dieser technischen Aspekte war es vor allem, die benötigten externen Komponenten, sowie die zugehörigen Schnittstellen zu identifizieren.

In Q4 2023 wurde die initiale Anforderungsanalyse abgeschlossen. In einer kurzen Review-Phase wurden die Anforderungen auf Plausibilität bewertet.

Parallel dazu wurde mit einer vorläufigen Einschätzung begonnen, welche SCAS-Testfälle sich für eine automatisierte Durchführung eignen. Das Kriterium für die initiale Einschätzung war, inwieweit sich unterschiedliche Herstellerarchitekturen auf den Testablauf auswirken. Als gut automatisierbar wurde ein Testfall eingeschätzt, wenn ausschließlich Stand-Tools genutzt werden können (z.B. Paket-Generatoren, Wireshark, TLS-Testtool, nmap). Als nicht automatisierbar wurde ein Testfall eingeschätzt, wenn der zu prüfende Mechanismus auf einer herstellerspezifischen Umsetzung basiert bzw. Implementierungsdetails voraussetzt (z.B. mögliche Überlastsituationen und Reaktionen auf diese).

Damit ergab sich zum Abschluss des ersten Meilensteins eine gemeinsame Wissensbasis zu Prüfungen nach NESAS CCS-GI, eine Liste an Anforderungen an das zu entwickelnde Testsystem und eine initiale Einschätzung an Testfällen, die sich zur Automatisierung eignen.

### **1.2.3 Identifikation von Open Source Tools zur Testautomatisierung (AP3)**

Als eines der Projektziele wurde der Einsatz von Open-Source Software angestrebt, zum einen als Rahmenwerk zur Automatisierung von Testabläufen, zum anderen zur Emulation von 5G Funktionen.

In Q3 2023 wurde über Internetrecherche mit einer Bestandsaufnahme begonnen, welche Tools und Lösungen zur Automatisierung von Testabläufen verfügbar sind. Auf Basis der vorhandenen Dokumentation wurde eine Ersteinschätzung vorgenommen.

Auf Grund der weitreichenden Erfahrung eines eigens entwickelten Testsystems aus den hauseigenen Reihen von EXS, wurden Anforderungen an diverse Open Source Tools gestellt, welche für die Umsetzung der Testfälle geeignet erscheinen. Der überzeugendste Kontrahent war dabei das Tool „Robot Framework“. Diese Entwicklungsgrundlage bildet ein passendes Rundumpaket, welches EXS aus der Expertise anderer Umsetzungen mit diesem Tool, bestätigen kann. Konkret wird Robot Framework für die Entwicklung der Testfälle genutzt, welche in das eigens entworfene Testsystem für 5G-Komponenten eingebunden wurde.

In Q4 2023 wurde eine Testinstanz von Robot Framework aufgesetzt und weiter untersucht. Um die Praxistauglichkeit einzuschätzen, wurden zunächst 3 Testfälle aus dem generischen SCAS Testkatalog umgesetzt und in einem kommerziellen Aufbau von Amarisoft getestet.

Das Robot Framework ist ein Open-Source-Testautomatisierungs-Framework, das sich besonders gut für Akzeptanztests und Akzeptanztest-getriebene Entwicklung (ATDD) eignet. Ein wesentlicher Vorteil ist seine benutzerfreundliche, schlüsselwortbasierte Syntax, die es auch Nicht-Programmierern ermöglicht, automatisierte Tests zu schreiben und zu verstehen. Durch seine modulare Architektur lässt sich das Framework leicht erweitern. Es unterstützt zahlreiche Bibliotheken wie Selenium für Webtests oder Appium für mobile Anwendungen und kann mit Python oder Java individuell angepasst werden. Ein weiterer Pluspunkt ist die klare Trennung von Testlogik und Implementierung, was Wartbarkeit und Lesbarkeit verbessert. Zudem erzeugt Robot Framework automatisch übersichtliche Testberichte und Logs, die

Fehleranalyse und -verfolgung erleichtern. Nicht zuletzt fördert das Framework eine gute Team-Zusammenarbeit, da es eine gemeinsame Sprache für Entwickler, Tester und Fachabteilungen schafft. Im Vergleich zu vielen anderen Testtools punktet das Robot Framework vor allem durch seine einfache, lesbare Syntax auf Basis von Schlüsselwörtern. Dadurch können auch Tester ohne Programmierkenntnisse automatisierte Tests schreiben und verstehen, was die Zusammenarbeit zwischen Fachbereichen und Entwicklern erleichtert. Ein weiterer Vorteil ist die große Flexibilität: Das Framework ist plattformunabhängig, lässt sich leicht durch eigene oder vorhandene Bibliotheken (z. B. Selenium, REST, Database) erweitern und unterstützt verschiedene Testarten – von Web- über API- bis hin zu Desktop-Tests. Zudem erzeugt Robot Framework automatisch strukturierte Berichte und Logs, was die Analyse erleichtert. Im Gegensatz zu vielen anderen Tools ist es vollständig Open Source – ohne Lizenzkosten – und hat eine aktive Community, die stetig neue Erweiterungen entwickelt.

Der initiale Einsatz von Robot Framework wurde somit als positiv bewertet, sodass entschieden wurde, das Tool in den folgenden Arbeitspaketen als Grundstein für die Testautomatisierung einzusetzen.

#### **1.2.4 Meilenstein-Meeting (23.11.23)**

Zum Abschluss der Anforderungsanalyse und initialen Identifikation möglicher Tools zur Testautomatisierung wurde ein projektinterner Austausch zwischen den Projektpartnern organisiert. Das Treffen wurde Vor-Ort in Essen durchgeführt. Im Zuge des technischen Treffens wurden bisherige Ergebnisse gesammelt, diskutiert und weitere Schritte besprochen. Außerdem wurde eine Übersicht zu Meilensteinen und Arbeitspaketen sowie zum aktuellen Stand und Fortschritt vorbereitet und dem DLR in Form einer Präsentation vorgestellt.

#### **1.2.5 Identifikation von Tools zur Nachbildung von 5G Funktionalität (AP3)**

Neben dem Rahmenwerk zur Automatisierung sah die Projektbeschreibung vor, Open-Source Software auch zur Nachbildung der benötigten 5G Funktionalität einzusetzen. In Q1 2024 wurde begonnen mögliche Lösungen zu identifizieren und zu untersuchen. Gängige Tools, die ein breites Spektrum an 5G Funktionalität implementieren, sind Open5GS und free5GC. Beide Tools werden häufig im Rahmen von Forschung und Entwicklung genutzt.

In Q2 2024 wurden zunächst Open5GS in Kombination mit UERANSIM untersucht. Dazu wurde Open5GS auf einer virtuellen Maschine mit Ubuntu 22.04 aufgesetzt und in Betrieb genommen. Die Mindestanforderungen für die virtuelle Maschine waren zwei CPU-Kerne, 4 GB RAM und 20 GB Festplattenspeicher. Vor der Installation von Open5GS mussten MongoDB und NodeJS installiert werden. Die Installation erfolgte mithilfe der von Open5GS bereitgestellten Dokumentation, und es bestand die Möglichkeit, das WebUI zu installieren. Zur Simulation von Endgeräten (UE) wurde zusätzlich UERANSIM verwendet.

Eine der Herausforderungen beim Aufsetzen von Open5GS war die korrekte Konfiguration der einzelnen Funktion, insbesondere beim Versuch komplexe Testaufbauten zu simulieren. Die Vielzahl an Parametern und Funktionen führte dazu, dass man schnell den Überblick verlieren konnte. Zudem erwies sich das gezielte Auslösen einzelner Trigger als weniger intuitiv als ursprünglich erwartet.

Im Rahmen der Vorabuntersuchung zeigt Open5GS viele Vorteile. Positiv hervorzuheben ist die Möglichkeit, Fehler durch die Analyse von Log-Dateien effektiv zu identifizieren und zu beheben, was die Stabilität und Zuverlässigkeit des Systems maßgeblich unterstützte. Darüber hinaus erwies sich die Community hinter Open5GS als äußerst hilfreich. Sie stellt eine

Vielzahl an Informationen und Dokumentationen zur Verfügung, die sowohl beim Inbetriebnehmen als auch beim Auftreten von Fehlern wertvolle Unterstützung boten.

Open5GS unterstützt auf den ersten Blick bereits eine Vielzahl von Mobilfunkprotokollen, was es zu einer attraktiven Wahl für produktklassenspezifische SCAS macht. Die kontinuierliche Weiterentwicklung im Open-Source-Kontext lässt darauf hoffen, dass in Zukunft noch mehr Protokolle abgedeckt werden, wodurch alle standardisierten Tests abgedeckt sind.

Aus den genannten Gründen wurden die Entscheidung getroffen, Open5GS für den Aufbau des 5G Testnetzes in AP5 zu nutzen.

### **1.2.6 Hintergrundanalyse zu konkreten Testabläufen (AP4)**

Im Rahmen von AP4 wurden alle Aspekte bearbeitet, die sich auf Anforderungen an die konkrete Implementierung von Testfällen bzw. die konkrete Umsetzung der SCAS-Tests beziehen.

Die SCAS liefern eine Liste an vordefinierten Testfällen. Bereits bei der Entwicklung solcher Testfälle durch die 3GPP oder anderer Standardisierungsorganisationen wird darauf geachtet, dass Testfälle einheitlich beschrieben sind, d.h. einem festen Format folgen, und spezifisch genug sind, dass sie von NESAS Prüfstellen durchgeführt werden können (siehe [FS.50]<sup>4</sup>). Gleichzeitig sind die Testfälle aber auch so generisch formuliert, dass sie unterschiedliche Implementierungen des Standards zulassen und somit auf Produkte unterschiedlicher Hersteller anwendbar sind. Im Rahmen einer Prüfung, muss ein Hersteller relevante Sicherheitsfunktionen beschreiben und testrelevante Informationen (z.B. eine Auflistung der implementierten Protokolle und Schnittstellen) zuliefern.

Ziel des aktuellen Vorhabens war es, konkrete Testfälle in das Framework zu integrieren um die Funktionsweise des Frameworks als Demonstrator zu präsentieren. Eine erste notwendige Vorarbeit lag darin, zunächst ganz allgemein notwendige Protokolle und Prüfinterfaces näher zu identifizieren um zu bestimmen, inwieweit sich Aspekte herstellerunabhängig umsetzen lassen. In Q4 2023 beginnend, wurde dazu zum einen der generische Testkatalog (TS 33.117) und zum anderen die produktklassenspezifischen Testkataloge für AMF (TS 33.512), UPF (TS 33.513), SMF (TS 33.515) und NRF (TS 33.1518) untersucht.

Für die mögliche Umsetzung der Tests des generischen Testkatalogs, konnte auf Erfahrungen aus dem Pilotprojekt zurückgegriffen werden. In einem ersten Schritt wurden die im Rahmen der Pilotierung durchgeführten Tests, genutzten Tools und Testabläufe auf Wiederverwendbarkeit untersucht. Außerdem wurde bestimmt, welche der manuell durchgeführten Tests sich durch einen automatisierten Ablauf vereinfachen lassen würden.

Zusätzlich zu den Erfahrungen aus dem Pilotprojekt, wurde auch eine Recherche nach neuen, potenziell geeigneten Test-Tools durchgeführt. Dabei zeigte sich jedoch, dass es ohne eine konkrete TOE schwierig ist, die Anwendbarkeit und Eignung solcher Werkzeuge verlässlich zu bewerten. Testverfahren und -abläufe sind in vielen Fällen stark produkt- bzw. herstellereinspezifisch, sodass eine allgemeingültige Aussage zur Einsatzfähigkeit einzelner Tools nur eingeschränkt möglich ist.

Für jeden Testfall aus den oben genannten produktklassenspezifischen Testkatalogen wurden die Vorbedingungen und Testschritte analysiert, um zu identifizieren, welche Schnittstellen für

---

<sup>4</sup> [https://www.gsma.com/solutions-and-impact/technologies/security/gsma\\_resources/fs-50-security-assurance-specification-development-guidelines/](https://www.gsma.com/solutions-and-impact/technologies/security/gsma_resources/fs-50-security-assurance-specification-development-guidelines/)

die zu untersuchenden Kommunikationsabläufe relevant sind, d.h. im Rahmen des automatisierten Testens angesprochen bzw. aufgezeichnet werden müssen. Begleitet wurde dies durch eine Recherche zu relevanten Protokollen, Nachrichten und Parametern, anhand derer sich ein Testergebnis ableiten lässt.

AP4 wurde in Q3 2024 abgeschlossen.

### **1.2.7 Einschätzung der Tools durch praktisches Aufsetzen (AP5)**

In AP5 wurden die Erkenntnisse aus AP3 und AP4 zusammengeführt und weitergehend analysiert.

Auf Seiten der TÜVIT wurden in Q2 2024 die identifizierten 5G Open-Source Tools im Gesamtkontext der geforderten Testszenarien bewertet. Dazu wurden einzelne Testfälle im Detail nachvollzogen und die Recherche-Ergebnisse aus AP4 mit real verfügbaren Nachrichten und Konfigurationsparametern im Open5GS Netz abgeglichen. Konkret wurden Testfälle der NRF untersucht. Dies ermöglichte es, Sicherheitsfunktionen zu identifizieren, die bisher nicht in den Open-Source Tools implementiert sind und eine Weiterentwicklung erfordern.

Nachdem in Q1 2024 Robot Framework als grundlegendes Mittel der Wahl für eine Testfallautomatisierung erklärt wurde, wurden exemplarische Arbeiten in einem bereits bei EXS vorhandenen 5G-Campusnetz getätigt. Zunächst handelte es sich dabei um 10 Testfälle aus dem generischen Testfallkatalog TS 33.117.

In Q2 2024 wurde der Fokus, nachdem das Grundgerüst der Testfallmaschine erprobt war, auf die Aufarbeitung der Testergebnisse und dessen Weitergabe, sowie Verwertbarkeit gesetzt. Dabei konnte auf die Erfahrung mit dem bereits langjährig auf dem Markt befindlichen Exceeding Solutions Tool, dem Testframework für Smart Meter-Komponenten, gebaut werden.

Da der Fokus auf Software-lastige Testfälle gelegt wurde, wurde in Q3 2024 die Entwicklung der Testmaschine entsprechend gestaltet. So war es möglich Core-Komponenten auf den Prüfstand zu stellen und die End-zu-End-Kommunikation der einzelnen Funktionen zu überwachen, welche relevant für die entsprechenden Testfälle sind. Die Adressierung findet dabei über eine portspezifische Anwahl statt und kann somit eindeutig und leicht identifizierbar zugeordnet werden. Über diese Eigenschaft lassen sich externe Funktionen (bspw. AMF, SMF, UPF, ...) einschleusen und testen.

### **1.2.8 Entwurf einer Zwischenschicht API (AP6)**

Für die Entwicklung einer Zwischenschicht API wurde eine REST-API mit OpenAPI-Spec verwendet, um spezifische Funktionen zu kapseln und aus dieser generischen API-Schicht bereitzustellen. Ziel des Unterfangens ist es eine vom TOE unabhängige generische Funktion zu bilden. Im Proxy werden dann in Abhängigkeit des TOEs spezifische Funktionen bzw. Module gerufen. Für die Testfallerstellung ist nur die generische Funktion notwendig. Bei weiteren TOEs muss nur der Proxy bzw. das spezifische Modul angepasst werden, nicht mehr der Testfall bzw. die Execution Engine.

### **1.2.9 Entwicklung und Präsentation eines MANTRA5G Demonstrators (AP7)**

In AP7 wurde der Testsystemdemonstrator finalisiert. Nachdem alle Komponenten zunächst einzeln auf ihre Tauglichkeit und perspektivische Interoperabilität getestet wurden, konnte diese nun letztendlich zusammengeführt und Schritt für Schritt im Verbund erprobt werden. Große Entwicklungsschritte zeigten sich bei der Implementierung des Rahmenwerks zur

Testfallverwaltung und bei dem ersten alleinstehenden Testsystem, mit konsequent unabhängigen Komponenten.

Dieses wurde im Verbund erprobt und weiterentwickelt, bis ein Demonstrator bereitstand, welche die gesetzten Anforderungen, für den vorgegebenen zeitlichen Rahmen, erfüllen konnte. Fokus lag dabei auf dem Aufzeigen der Möglichkeiten und den gegebenen Potentialen bei einer Weiterentwicklung. Letztendlich konnten einige Testfälle aus den Katalogen TS 33.117, AMF 33.512 und SMF 33.515 programmiert und erfolgreich getestet werden.

### 1.3 Eingehende Darstellung der erzielten Ergebnisse

In den folgenden Abschnitten werden die wesentlichen Ergebnisse des Vorhabens ausführlich dargestellt.

#### 1.3.1 Anforderungs-Checkliste

Ein gemeinsames Verständnis über das zu entwickelnde System und die geforderten Rahmenbedingungen ist ein wesentliches Kriterium für den reibungslosen Projektverlauf. In Diskussions- und Review-Runden wurden die wesentlichen Eigenschaften der MANTRA Toolbox besprochen.

Dabei ging es zum einen um die Anforderungen an das Testsystem-Rahmenwerk, d.h. die vorgesehene Nutzerinteraktion, die interne Verwaltung von Testfällen, Testergebnissen und Protokolldateien sowie die interne Umsetzung von Testabläufen. Zum anderen ging es um die Anwendungsebene, d.h. Anforderungen, die sich aus der konkreten Implementierung einzelner SCAS-Testfälle ergeben.

Die resultierenden Anforderungen wurde in Form von User Stories und Anwendungsfällen (siehe Abbildung 1) bzw. in Form einer Checkliste dokumentiert (siehe Abbildung 2).

## 2 User Stories

### 2.1 Definition von Nutzern und Rollen

Rolle	Beschreibung
Tester	Ein Tester führt Prüfungen an Komponenten aus, die bereits in den Testaufbau integriert sind. Er kann einzelne Tests starten, wiederholen oder eine Gruppe von Test automatisiert durchführen. Diese Prüfaufgabe ergibt sich im „Daily Business“, wenn eine Komponente der Produktprüfung unterzogen wird. Die generierten Nachweise dienen als Input für den Prüfbericht. Während des Prüfens können Testfallbezogene Parameter (z.B. zu Prüfende Schnittstelle / zu prüfendes Protokoll) variiert und dann einzelne oder mehrere Testschritte erneut durchgeführt werden. Die allgemeinen Konfigurationen zum Testsetup (z.B. IP-Adresse des Prüfgegenstandes / IP-Adressen in der Prüfumgebung, Protokollabläufe im Hintergrund wie UE-Registrierung) werden vom Tester jedoch nicht verändert.
Test-Ingenieur	Ein Test-Ingenieur richtet den Prüfaufbau für eine neue zu prüfende Komponente ein. Dazu richtet er die allgemeinen Konfigurationen zum Testsetup ein (z.B. IP-Adressen, UE-Konfigurationen wie SUP/IMS/IMCC, kryptographisches Material,...). Die Komponente wird durch den Test-Ingenieur entsprechend der <u>Herstellerdokumente</u> oder mit Unterstützung des Herstellers in Betrieb genommen. Anschließend wird der Prüfgegenstand mit der Testumgebung verbunden, indem notwendige Protokollverbindungen aufgebaut und getestet werden.  Optional: Wurde diese Komponente bereits evaluiert, können unterschiedlich alte hinterlegte Einstellungen im Testframework geladen werden, um z.B. die notwendigen Schnittstellen entsprechend der <u>Herstellerangaben</u> zu konfigurieren.  Handelt es sich um eine Komponente, die in dieser Form noch nicht geprüft wurde, wird eine entsprechend neue Prüfkongfiguration angelegt. Diese lässt sich entweder aus einer bereits bestehenden Prüfkongfiguration ableiten und anpassen oder von Grund auf neu erstellen.
Test-Entwickler	Test-Entwickler implementieren neue Testabfolgen, sofern dies aus einem der folgenden Gründe notwendig ist:

### 2.2 Anwendungsszenarien

ID	Thema	Beschreibung	Kommentar
AS.1	Durchführen von Tests	Als Tester möchte ich einen Testfall oder mehrere Testfälle aus einer Liste der relevanten Testfälle auswählen und starten können. Innerhalb des Testframeworks werden die Prüfabläufe automatisch durchgeführt. Falls ein manuelles Eingreifen notwendig ist, wird mir das als Tester angezeigt. Das Ergebnis der Tests (PASS/FAIL) wird mir angezeigt. Im Anschluss exportiere ich die generierten Testnachweise und Testkonfigurationen oder habe dafür in den Einstellungen der Testframework einen Speicherort angegeben. Die exportierten Testnachweise sind vollständig und entsprechend den Vorgaben des BSI.	
AS.2	Einbinden der Prüfkompone-nente	Als Testingenieur bin ich dafür verantwortlich, das Testsystem aufzubauen und einzurichten. Dazu nehme ich den Prüfgegenstand entsprechend der Herstellerdokumentation in Betrieb. Bei Bedarf erhalte ich bei der Inbetriebnahme Unterstützung durch den Hersteller.  Innerhalb der existierenden Testabläufe gibt es Platzhalter für herstellereinspezifische Befehle und Umsetzungen (z.B. neuen Nutzer anlegen / Firewall-Regeln ändern). Diese fülle ich entsprechend der <u>Herstellervorgaben</u> mit eigenen Testschritten und Abfolgen (z.B. Login über <u>Commandozeile</u> . Ausführen der folgenden Befehle, ...). Die korrekte Umsetzung kann ich anhand von internen Checks in den Testabläufen überprüfen. Erst wenn keine internen Fehler auftreten (ERROR) und plausible Ergebnisse sichtbar sind, kann die reguläre Prüfung des Gegenstandes vom Tester durchgeführt werden.  Zu der Einrichtung des Testsetups gehört außerdem die Definition der allgemein gültigen globalen Konfiguration für den Prüfgegenstand (z.B. IP-Adressen aller Kompo-	

Abbildung 1: User-Stories (Auszug)

Die User-Stories beschreiben die Interaktion von Nutzern mit dem Testsystem bzw. das Verhalten der Toolbox, das ein Nutzer erwarten kann. Jeder Anwendungsfall enthält ein Szenario bzw. einen Prozessablauf, der vom zu entwickelnden System umgesetzt werden muss.

### 3 Allgemeines Testkonzept (TC)

#### 3.1 Prüfgegenstand (PuE)

ID	Beschreibung
Inf_TC_PuE_01	Die zu prüfende 5G Komponente wird im Folgenden als Prüfgegenstand bezeichnet.
Inf_TC_PuE_02	5G Komponenten können als klassische Netzwerkfunktionen (HW+SW) vorliegen, als virtuelle Netzwerkfunktion oder als cloud-native Netzwerkfunktion.
Req_TC_PuE_01	Das Framework betrachtet den Prüfgegenstand als Black-Box. Es werden keine Funktionen oder Tools vorausgesetzt, die auf dem Prüfgegenstand installiert sind.
Req_TC_PuE_02	Das Testframework kann über herstellerspezifische Schnittstellen Einstellungen am Prüfgegenstand vornehmen. Dies dient der Vorkonfiguration des TOE. (TOE in den Betriebszustand bringen, z.B. über SSH, Weboberfläche, Management-Einheit) Wird in weiteren Anforderungen konkretisiert.
Inf_TC_PuE_03	Der Hersteller des Prüfgegenstandes liefert einer Architekturübersicht, Handbücher und Schnittstellenbeschreibungen zum Prüfgegenstand. Zum Testzeitpunkt ist den Testframework-Nutzern bekannt, welche Einstellungen am Prüfgegenstand wie vorgenommen werden können.
Bsp_TC_PuE_01	Firewall-Regeln oder Paketfilter-Eigenschaften eines Prüfgegenstandes können sich auf die Testergebnisse auswirken. Ein Hersteller würde im Nutzerhandbuch beschreiben, wie Firewall-Regeln konfiguriert werden, sodass der Nutzer des Testframeworks diese Einstellungen vornehmen kann.

Abbildung 2: Anforderungs-Checkliste (Auszug)

Die Checkliste beinhaltet unterschiedliche Notationsebenen, die entsprechend durch ihre ID gekennzeichnet sind:

- „INF“: stellen Hintergrundinformationen dar, die die folgenden Anforderungen in einen Kontext einordnen und dem Entwickler ein besseres Verständnis für das Gesamtsystem geben
- „REQ“: sind die tatsächlichen Anforderungen, die ein System vollständig umsetzen muss
- „Bsp“: liefern weitere Beispiele, die Anforderungen konkretisieren bzw. bildlicher darstellen.

Mit der Checkliste werden Anforderungen aus unterschiedlichen Themenfeldern abgedeckt. Konkret wurden die folgenden Aspekte bearbeitet und dokumentiert:

- Anforderungen zum allgemeinen Testkonzept
- Anforderungen zur internen Organisation von Tests
- Anforderungen zu Nutzerinteraktionen
- Anforderungen zum Herstellerspezifisches Setup
- Anforderungen zu notwendigen Protokollen
- Anforderungen zu vorgesehenen Tools

Die Checkliste ist so gestaltet, dass sie später eine Bewertung des entwickelten Testsystems ermöglicht.

### 1.3.2 5G Recherche und Testfallentwicklung

Die SCAS-Dokumente geben in generischer Form vor, welche Anforderungen an eine Komponente gelten und wie die Einhaltung dieser Sicherheitseigenschaften von einer Prüfstelle getestet werden soll. Im Detail verweist dabei jeder Testfall auf weitere Dokumente der 3GPP um das übergeordnete Szenario zu beschreiben.

In Summe gibt es 22 Komponenten-Klassen, die nach NESAS CCS-GI geprüft werden müssen. Die zugehörigen Protokolle und Prozeduren sind in mehr als 500 Spezifikationsdokumenten der 3GPP beschrieben. Die Recherche hat ergeben, dass die SCAS-Dokumente unterschiedliche Testarten beschreiben. Um dies zu verdeutlichen und das allgemeine Vorgehen zu beschreiben, werden im Folgenden zwei Tests näher erläutert:

Der erste Testfall bezieht sich auf die Registrierung einer Netzwerkfunktion bei der NRF. Die zugehörige Prozedur ist in [TS 129.510] beschrieben und exemplarisch in Abbildung 3 dargestellt.

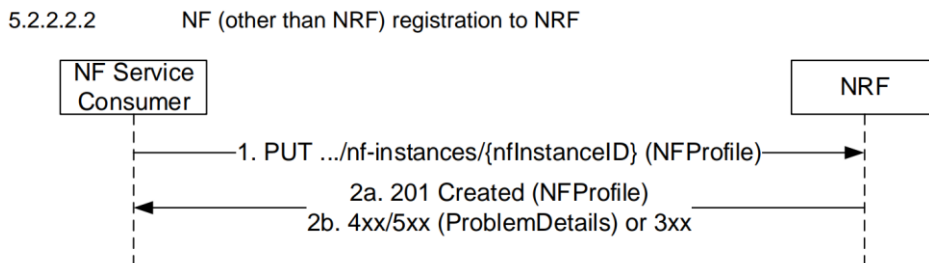


Figure 5.2.2.2.2-1: NF Instance Registration

Abbildung 3: Prozedur NF Instance Registration [TS 129.510]

Zu erkennen ist, dass für den Testfall lediglich eine externe Komponente benötigt wird, um die initiale Kommunikation zu triggern, und die Antwort auf dem gleichen Kanal erfolgt. Ein solcher Testfall lässt sich gut und einfach umsetzen. Die Herausforderung steckt allerdings im Detail. Der Nachricht mitgeliefert werden muss das „NF-Profile“, das aus einer Vielzahl an Einzelparametern besteht. Ist hier nur ein Wert falsch gesetzt, könnte der Test fehlschlagen, obwohl die zu prüfende Komponente sich standardkonform verhält. Es zeigt sich also, dass ein Detailwissen zum 3GPP Standard zwingend erforderlich ist, um auch einfache Testfälle umzusetzen.

Der zweite Testfall bezieht sich auf das PDU Session Establishment der SMF. Die zugehörige Prozedur ist in [TS 123.502] beschrieben und exemplarisch in Abbildung 5 dargestellt. Für diesen Testfall ist die Interaktion zwischen vielen Komponenten erforderlich. Innerhalb des Testsetups muss sichergestellt werden, dass alle Kommunikationsabläufe reibungslos funktionieren.

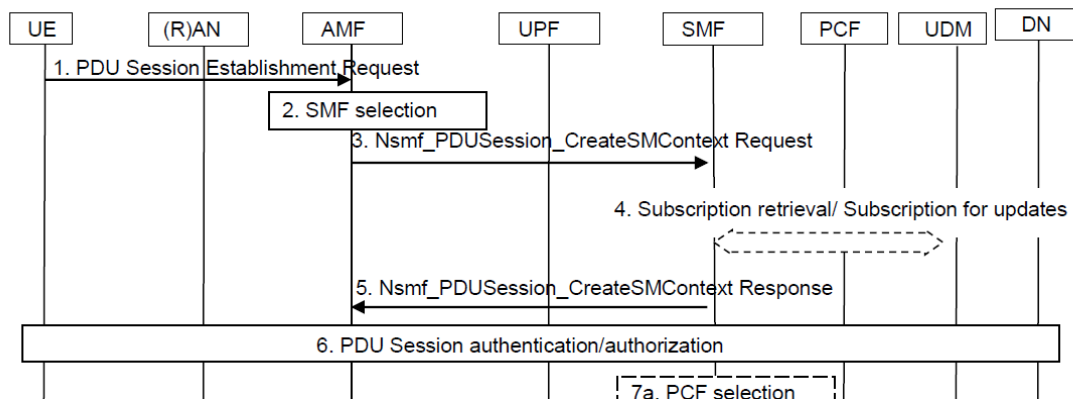


Abbildung 4: Prozedur PDU Session Establishment [TS 123.502]

Um die Testfälle entsprechend zu designen und sicherzustellen, dass die benötigten Vorbedingungen hergestellt werden können, wurden für jeden Testfall die relevanten 3GPP Spezifikationsdokumente, die Szenarien, Protokolle, Prüfschnittstellen und Trigger in Form einer Excel Tabelle dokumentiert (siehe Abbildung 5). Die Analyse wurde für den generischen Testkatalog und die 5G Produktklassen AMF, UPF, SMF und NRF durchgeführt und beinhaltet wesentliche Hintergrundinformationen zum konkreten Entwurf der Testfälle.

#	A	B	C	D	E	F	G	H	I	J
#	Section	Name	ID	Herstellerinformation mit Auswirkung auf die Testumsetzung	Komponenten der Prüfumgebung/ Parametrisierung der Tools	Allgemeiner Ablauf, [SCAS_33.512]	Hintergrund / Szenario / Einordnung	Referenz/ Hintergrundinformationen		
7	4.2.2.3.2	NAS NULL Integrity protection	TC_NAS_INT_INF_AMF		1) Registrierung des UE im Netz ("normale" Registrierung / "emergency registration") und erfolgreicher Authentifizierung 2) Herstellen des Sicherheitskontextes (Security Mode Command Procedure) 3) Auswahl des am höchsten priorisierten Algorithmus, der von AMF und UE unterstützt wird	Refinements in AIS_N2_v1.1 UE	<p><b>Hintergrund:</b> Basierend auf der Spezifikation muss eine AMF NIA0, 128-NIA1 und 128-NIA2 unterstützen und KANN optional auch 128-NIA3 unterstützen. NIA0 darf im regulären Betrieb nicht verwendet werden. NIA0 ist "plaintext with no encryption". Bei der Registrierung kann ein UE den Status "Emergency" mit übergeben. Dann gibt es an manchen Stellen abweichende Prozesse. Bei unauthentifizierten Emergency-Call darf NIA0 z.B. verwendet werden (laut 33.501, laut AIS N2 anscheinend nicht).</p> <p><b>Scenario:</b> Bei Registrierung sendet UE initiale NAS Nachricht. Diese enthält unter anderem die UE security capabilities. Innerhalb der AMF sind ebenfalls Algorithmen hinterlegt, die eine AMF prinzipiell verwenden kann, inklusive deren Priorität. Die AMF wählt den am besten passenden Algorithmus aus.</p>	[TS_33.501, 6.7.1.1/ 6.7.2]; Algorithm selection / Security Mode command procedure [TS_33.501, 5.5.2]; Requirement: Deactivation of NIA0 [TS_24.501, 4.4.2.1c / 4.4.2.1d / 4.4.1 / 5.4.2.3 / 9.11.3.34]; NAS-Protokoll [TS_23.502, 4.2.2.2.1]; Registration procedures [TS_24.301]; NAS Protokoll (konkrete Nachrichten)		
8	4.2.2.3.3	NAS integrity algorithm selection and use	TC_NAS_INT_SELECTION_USE_AMF		1) UE sendet Registrations-Request 2) Die Security Mode Command Complete Nachricht wird aufgezeichnet 3) Der von der AMF gewählte Algorithmus und die UE Security Capabilities werden ausgeliefert	UE, AUSF, UDM	<p><b>Hintergrund:</b> Um einem Man-in-the-middle Angriff vorzubeugen, liefert die AMF die empfangenen UE Capabilities erneut an das UE.</p> <p><b>Scenario:</b> Bei Registrierung wird Security Mode Command Procedure durchlaufen</p>	[TS_33.501, 5.5.2 / 6.7.1.1/ 6.7.2]; Algorithm selection / Security Mode command procedure [TS_23.502, 4.2.2.2.1]; Registration procedures [TS_24.301]; NAS Protokoll (konkrete Nachrichten)		

Abbildung 5: Dokumentation der Prüfinterfaces, Protokolle und Trigger (Auszug)

Zudem wurde für jeden Testfall eine Einschätzung der Automatisierbarkeit vorgenommen und in der Tabelle notiert. Da im Rahmen des Projektes keine Herstellerkomponente zur Verfügung stand, wurde die Abschätzung auf Basis theoretischer Überlegungen sowie der durchgeführten Prüfungen im NESAS Pilotprojekt vorgenommen. Es zeigt sich, dass sich

- 15 der generischen Testfälle voraussichtlich herstellerunabhängig und automatisiert umsetzen lassen,
- 28 der generischen Testfälle für bestimmte Architekturen (z.B. Unix-Systeme) einheitliche umsetzen lassen, und
- für 20 der generischen Testfälle eine Automatisierung an dieser Stelle nicht möglich ist, da die Umsetzung starken Herstellerinput erforderlich macht.

Die Produktklassenspezifischen Tests lassen sich hingegen automatisiert umsetzen, da eine Liste an Triggern und Befehlen durch die 3GPP Spezifikation vorgegeben ist. Hier müssen ggf. lediglich einzelne Hersteller-Dialekte beachtet werden. Im Rahmen dieses Projektes besteht die Idee darin, dies über eine Hersteller-API zu realisieren.

### 1.3.3 Testsetup und Einschätzung der Open-Source Tools

Die Testfallspezifikation in den SCAS-Dokumenten geben vor, dass eine zu prüfende 5G Komponente in ihrer vorgesehenen Einsatzumgebung getestet werden soll. Um die 5G Core Funktionen und Sicherheitseigenschaften prüfen zu können, sind je nach Testfall Interaktionen mit umliegenden Komponenten erforderlich.

Open5GS ermöglicht es, ein gesamtes 5G Kernnetz bestehend aus den folgenden Funktionen nachzubilden: NRF - NF Repository Function, SCP - Service Communication Proxy, SEPP - Security Edge, Protection Proxy, AMF - Access and Mobility Management Function, SMF - Session Management Function, UPF - User Plane Function, AUSF - Authentication Server Function, UDM - Unified Data Management, UDR - Unified Data Repository, PCF - Policy and Charging Function, NSSF - Network Slice Selection Function, BSF - Binding Support Function. UERANSIM emuliert ein 5G Endgerät und die Basisstation. Damit lassen sich zunächst relevante Prozeduren wie NF-Registrierung, UE Registrierung, Sitzungs- und Verbindungsaufbau, abbilden. Das dafür notwendige Testsetup ist in Abbildung 6 dargestellt.

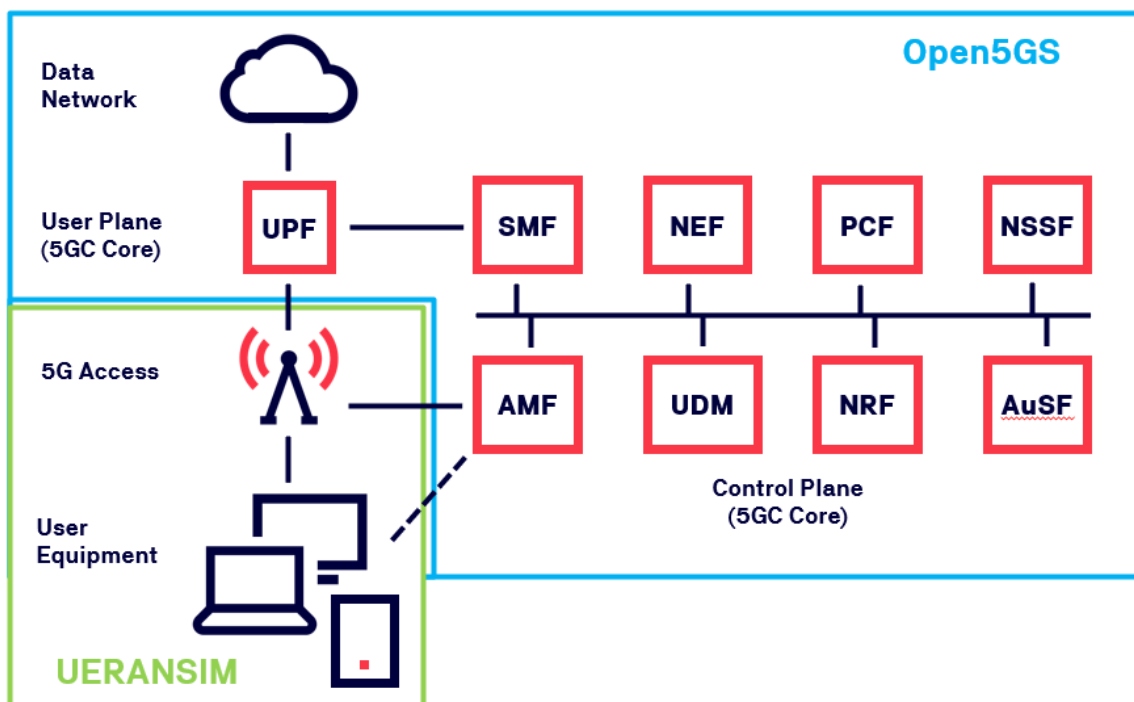


Abbildung 6: Testsetup mit OpenSource Tools

Basierend auf den Rechercheergebnissen aus AP4 wurden unterschiedliche Prozeduren in Open5GS nachgebildet. Hierbei zeigten sich die Vorteile und Grenzen des Tools.

Im Prinzip ist die vollständige 5G Kernnetz funktionalität vorhanden. Komponenten lassen sich über .yaml Dateien konfigurieren und einzeln hochfahren. Die globalen Prozeduren können von außen getriggert werden, z.B. durch Einwahl eines Endgerätes. Intern wird über Logdateien festgehalten welche Nachrichten zwischen welchen Komponenten ausgetauscht werden. Diese Aspekte machen Open5GS zu einem attraktiven und nutzbaren Tool für die produkt-klassenspezifischen SCAS-Tests.

Aus dem praktischen Testen heraus zeigt sich, dass sich Open5GS in erster Linie dazu eignet, die globalen Prozeduren umzusetzen, dass es aber nicht möglich ist Einzeltrigger aus einer

Funktion heraus zu senden. Da dies für unterschiedliche Testfälle jedoch benötigt wird, ist eine Anpassung des Tools erforderlich.

Hierbei ist es lediglich möglich, simulierte API-Anfragen zu senden, die jedoch nicht von der simulierten Funktion selbst verarbeitet werden können. Das bedeutet, interne Prozesse lassen sich nicht vollständig und gezielt auslösen.

Darüber hinaus wurde Open5GS nicht nur in seiner Standardform als lokal laufender Prozess eingesetzt, sondern zunächst auch containerbasiert mit Docker realisiert. In einem weiteren Schritt erfolgte der Betrieb innerhalb eines Kubernetes-Clusters. Diese Umsetzung erfolgte mit Blick auf die spätere Nutzung durch Hersteller oder Betreiber, da davon ausgegangen werden kann, dass ein 5G Core in produktiven Cloud-Umgebungen überwiegend containerisiert eingesetzt wird.

Ein weiterer Aspekt, der während der praktischen Untersuchungen aufgefallen ist, betrifft nicht standardmäßig aktivierte Sicherheitseigenschaften, insbesondere mTLS (mutual TLS).

In Open5GS ist TLS bzw. mTLS standardmäßig nicht aktiviert. Es besteht jedoch die Möglichkeit, die Verschlüsselung über manuelle Konfiguration zu aktivieren. Eine native, automatisierte Integration für TLS/mTLS ist in Open5GS derzeit nicht vorgesehen. Das bedeutet, dass die konkrete Umsetzung insbesondere die Einbindung und Verwaltung von Zertifikaten in der Verantwortung des Betreibers liegt und individuell erfolgen muss.

Zur Realisierung von mTLS wurde im Kubernetes-Cluster eine Public Key Infrastructure (PKI) mit Hilfe von Smallstep/AUTOCERT aufgebaut. Diese PKI verwaltet die Zertifikate der einzelnen Netzfunktionen (z. B. AMF, SMF, UPF) und ermöglicht deren regelmäßigen Austausch. Durch gezielte Konfiguration der Open5GS-Komponenten konnte so eine gegenseitige Authentifizierung über mTLS etabliert werden.

Die größte Herausforderung bestand im komplexen Zusammenspiel zwischen Kubernetes, Containerisierung und den besonderen Anforderungen eines 5G-Kernnetzwerks. Das initiale Aufsetzen, Konfigurieren und sichere Betreiben eines derart verteilten Systems erfordert tiefgehende Kenntnisse sowohl im Bereich 5G als auch im Cloud- und Kubernetes-Ökosystem.

Im Rahmen der Untersuchungen wurde neben Open5GS auch das 5G-Core-Projekt Free5GC betrachtet und erfolgreich in einer Kubernetes-Umgebung aufgebaut und betrieben. Der Fokus lag dabei auf der Evaluierung der grundsätzlichen Einsatzfähigkeit für SCAS-Testzwecke.

Diese Erkenntnis ist entscheidend für die Auswahl und Weiterentwicklung von Testumgebungen, da eine vollständige Abdeckung der SCAS-Testfälle mit den aktuellen Open-Source-Implementierungen nur eingeschränkt möglich ist.

Ein Beispiel dafür sind Unterschiede in den API-Anfragen und Nachrichtenformaten, die sich zwischen den Releases verändern können. Daher ist es notwendig, im Einzelfall genau zu prüfen, inwieweit Herstellerkomponenten mit Open-Source-Komponenten interoperabel sind, um realistische und valide Testergebnisse zu gewährleisten.

Dabei erwies sich Free5GC als weniger umfangreich und umständlicher ergänzbar als Open5GS. Weitere Versuche wurden daher mit letzteren durchgeführt.

#### **1.3.4 Aufbau der Systemlandschaft**

Basierend auf Open5GS und srsRAN wurde bei Exceeding Solutions ein eigenes 5G-Campusnetz in Hard- und Software aufgesetzt. Somit stand uns eine editierbare Grundlage für unser eigens kreierte Testsystem zur Verfügung. Dabei wurden zwei wesentliche Unterscheidungsmerkmale in den Testfällen getroffen. Zum einen Testfälle für die Kategorie „Softwaretests“ und zum anderen für einen „Hardwareteil“. Hardwaretests beinhalteten dabei die Aspekte einer physischen Anwendung. Einer Anwendung der Testfalldurchführung auf der Funkebene bezieht damit den physischen Datenlayer mit in das Testszenario ein, so dass eine vollständige End-zu-Ende Kommunikation in dem jeweiligen Testszenario durchgeführt wurde. Reine Softwaretests beinhalteten nur die interne Kommunikation zwischen den einzelnen Funktionen bzw. sind somit im OSI-Model eher auf den höheren Layern verortet. Beide Kategorien wurden im Rahmen des Forschungsprojekts untersucht, wobei der Fokus, auf Grund von Projektmanagemententscheidungen, zeitlich bedingt, ganz klar auf der Softwareseite lag. Somit wurde in der Praxis ein abgesetztes PC-System hergerichtet, auf welchen die 5G-Software, als Grundlage für das MANTRA-Testsystem, läuft und die entsprechende Funkhardware angeschlossen. Für Hardwaretests standen uns ein separates System von Amarisoft zur Verfügung und ein 5G-fähiges Handy, sowie weitere 5G-Modems. Diese Komponenten wurden als Funktestpartner verwendet. Um Interferenzen mit anderen Funkpartnern zu anderen zeitgleichen Projekten und kritischen Komponenten zu vermeiden, wurde, um eine eindeutige Verifizierbarkeit des Testsystems zu gewährleisten, ein Großteil der Tests in einer abgeschirmten Umgebung durchgeführt (Abschirmzelt).

Die Rechercheergebnisse der TÜV-Informationstechnik zeigen, dass sich die Idee einer Hersteller-API besonders für die Produktklassenspezifischen Testfälle eignet. Die 3GPP gibt die Struktur einer Ressourcen URI vor unter der Service Consumer einen bestimmten Service aufrufen können. Hersteller können ggf. minimal von der Namenskonfiguration abweichen, würde ihre API aber in Form einer OpenAPI Spezifikation beschreiben. Anders sieht es bei den generischen Testfällen aus. Hier besteht die Vermutung, dass für unterschiedliche Hersteller aufgrund unterschiedlicher Produktarchitekturen teilweise unterschiedliche Testabläufe notwendig werden. Da uns im Rahmen des Forschungsprojekts kein konkreter Prüfgegenstand eines Herstellers vorliegt, kann dieser Aspekt nicht weiter untersucht werden.

#### **1.3.5 Entwicklung des Demonstrators**

Nachdem Eingehens das System auf seine zu erwartende Campusnetzfunktionalität untersucht wurde, wurde Robot Framework als Testfallentwicklungssystem und ein eigens gebautes Rahmenwerk als übergeordnete Testmaschine dazu genommen. Das Rahmenwerk lässt die Möglichkeiten der Bedienung und Verwaltung, in automatisierter Form, des Testsystems zu. Dem Anwender ist es hier möglich Kategorien von Testfällen aber auch einzelne Test auszuwählen, sich den Testfallverlauf anzugucken, Resultate anzeigen zu lassen und diese zu exportieren. Die Rubriken sind hierbei zunächst allgemein nach den Testfallkatalogkategorien „generisch“ und „spezifisch“ eingeteilt. Die Testfälle werden dann im Hintergrund in Robot Framework ausgeführt. Bei der Programmierung der einzelnen Testfälle lag ein großes Augenmerk auf der Automatisierbarkeit in der Ausführung der Testfälle aber auch in der Testfallentwicklung. So wurden einzelne und sich wiederholende Programmschritte, wie bspw. Login oder Abfragen, in separate Abschnitte oder Funktionen gepackt, die aus einer Library aufgerufen werden können. Mit Hilfe dieser „Microsteps“ kann die Testfallentwicklung zeitsparend oder teilweise automatisiert durchgeführt werden. Für eine automatisierte Testfallentwicklung fehlte uns jedoch die Zeit ein Rahmenwerk für diesen Aufbau zu entwickeln und stellte somit nicht den Fokus der letztendlichen Umsetzung dar. Vielmehr ging es hierbei darum die Machbarkeit herauszufinden und sich einem praktischen Nutzen dessen anzunähern. Die internen Kommunikationen zwischen den einzelnen

Funktionen, welche testfallspezifisch relevant sind, lassen sich im Nachgang im Log ansehen und können ebenfalls als Testresultat gespeichert werden. Somit ist eine Transparenz des Testsystems gegeben, auch herstellerunabhängig agieren zu können. Lediglich müssen für die Softwaretests die zu testenden Parts in das Testsystem gebracht und entsprechend zugeordnet werden. Da die internen Funktionen spezifische Adressen haben, kann eine Kommunikation mit der „Prüfstandkomponente“ gekoppelt werden. Herstellerabhängige Akzente müssen dabei in den Testfällen oder in einer Hersteller-API gemanagt werden. Leider war es uns in der Projektzeit nicht möglich weitere externe 5G-Komponenten auf unseren Prüfstand zu stellen und eine höhere Diversität zu erzeugen. Somit verblieben wir mit den uns verfügbaren Komponenten (eigenes 5G-System; Amarisoft-System; 5G-Handy; diverse 5G-Modems) und testeten diese oder auch manipulierte diese gezielt.

## **1.4 Einschätzung des durchgeführten Vorhabens**

Die folgenden Abschnitte zeigen eine Gegenüberstellung der Ziele und Ergebnisse des durchgeführten Vorhabens.

### **1.4.1 Vergleich zur ursprünglichen Vorhabensbeschreibung**

Im Rahmen des Vorhabens wurden alle Arbeitspakete wie geplant durchgeführt. In Bezug auf zeitlichen Aufwand haben sich leichte Abweichungen zum ursprünglichen Plan ergeben, ohne jedoch die Gesamtaufwände zu beeinflussen:

- AP3 wurde nicht vollständig bis Q4 2023 abgeschlossen. Es hat sich gezeigt, dass eine Bewertung der Tools auf Eignung und Integrationsmöglichkeiten (d.h. Möglichkeiten und Grenzen) nicht allein durch Sichtung der Dokumentation erfolgen konnte. Stattdessen wurde die Einschätzung durch praktisches Testen der Tools in AP5 untermauert. Notwendige Weiterentwicklungen ließen sich erst nach Bearbeitung von AP4 und AP5 identifizieren.
- In AP4 hat sich gezeigt, dass ein direkter Rückschluss von Vorbedingungen und Testschritten, wie sie in den SCAS-Dokumenten vorgegeben sind, auf einzelne Dokumente der 3GPP Spezifikation nicht möglich ist. Stattdessen sind Protokolle und Prozeduren in mehr als 500 Spezifikationsdokumenten verteilt beschrieben. Um zu identifizieren, welche Schnittstellen und Trigger für einen Testfall erforderlich sind, muss der Testfall in das übergeordnete Szenario eingeordnet und auch die Interaktion mit weiteren Komponenten berücksichtigt werden. Eine Recherche hat sich als deutlich aufwendiger herausgestellt, sodass AP4 parallel zu AP5 weitergeführt und erst zeitgleich mit diesem in Q3 2024 beendet wurde.

Inhaltlich haben sich die folgenden Verschiebungen des jeweiligen Schwerpunkts als sinnvoll herausgestellt.

- Aufgrund der oben beschriebenen aufwändigen Recherche in AP4, wurde die Erstellung einer umfassenden Tabelle als Vorarbeit für die zukünftige Entwicklung von Testfällen als sinnvoll erachtet.
- In AP4 konnte eine Einschätzung zur herstellerunabhängigen Implementierung und Automatisierbarkeit von Testschritten nur auf Basis theoretischer Überlegungen erfolgen, da im Rahmen des Vorhabens kein konkretes Herstellerprodukt bzw. keine Informationen zu konkreten Herstellerarchitekturen verfügbar war.
- Von TÜVIT wurde kein Testequipment angeschafft. Wesentliche Untersuchungen konnten in den Räumlichkeiten der Exceeding Solutions GmbH durchgeführt werden. Im Laufe des Vorhabens wurde basierend auf ersten Ergebnissen die Entscheidung

getroffen den Fokus auf den Test von Software-Kernnetz-Komponenten zu legen. Dadurch rückte der Aufbau einer physikalischen Teststrecke in den Hintergrund.

- Die im Rahmen von AP7 in den Demonstrator integrierten Testfälle konnten nicht gegen einen realen Prüfgegenstand getestet werden. Um dennoch eine Einschätzung zu ermöglichen, wurde eine Callbox von Amarisoft sowie Open5GS als Testnetz genutzt, was jedoch durch den schon vorher motivierten Ende-zu-Ende Charakter einer realen Testumgebung entspricht.

#### 1.4.2 Fazit und Ausblick

Mit den durchgeführten Arbeiten konnte wesentliche Vorarbeit für die Entwicklung eines umfänglichen NESAS CCS-GI Testframeworks geleistet werden. Es wurden geeignete Open-Source Tools untersucht, die einen Vielversprechenden Ansatz liefern und basierend auf den Rechercheergebnisse in Zukunft angepasst und weiterentwickelt werden können, um alle relevanten Szenarien und Anforderungen aus NESAS CCS-GI abzudecken. Der Umfang der 3GPP Spezifikation hat sich als deutlich komplexer herausgestellt als ursprünglich gedacht. Damit wird auch die Entwicklung konkreter Testfälle als zeit- und ressourcenintensiv eingeschätzt. Eine große Bedeutung kommt den herstellereigenen Implementierungen der unterschiedlichen Sicherheitsmechanismen zu. Je nachdem wie ähnliche gewissen Mechanismen von unterschiedlichen Herstellern umgesetzt werden, ergeben sich hierbei möglicherweise Synergien. Im Rahmen des Vorhabens konnte dieser Aspekt jedoch nicht weiter betrachtet werden, da keine konkreten Herstellerprodukte bzw. Kenntnisse zu unterschiedlichen Herstellerarchitekturen verfügbar waren.

Die allgemeine Machbarkeit in Bezug auf Umsetzung und Automatisierung von SCAS-Testfällen konnte über den Demonstrator nachgewiesen werden. Die Ergebnisse des Vorhabens können in folgenden Prüfungen und Projekten genutzt und weiterentwickelt werden.

## 2 Die wichtigsten Positionen des zahlenmäßigen Nachweises

Der zahlenmäßige Nachweis umfasst alle projektbezogenen Ausgaben, die in der Projektlaufzeit angefallen sind. Die finanzielle Dokumentation der Verwendung der Fördermittel wurde in Profionline eingereicht.

### 2.1 TÜVIT

Die im Zuwendungsbescheid angegebenen Materialkosten wurden nicht genutzt. Auf Seiten der TÜVIT wurde kein eigenes Testequipment angeschafft.

Folgende TÜVIT Reisen wurden durchgeführt:

- Kick-Off-Meeting in Merseburg, 14.06.23
- Prüfstellentreffen NESAS in Freital, 10.04.24

Die im Zuwendungsbescheid angegebenen sonstigen unmittelbare Vorhabenkosten wurden für die Ausleihe/Miete eines Prüfgegenstandes europäischer Hersteller eingeplant. Da im Rahmen des Projektes keine Zusammenarbeit mit Herstellern möglich war, wurden die sonstigen Vorhabenkosten nicht genutzt.

Die Personalkosten erreichten nicht die volle Zuwendungshöhe.

Weitere Informationen finden sich im Sachbericht zum Verwendungsnachweis Teil III der TÜVIT.

## 2.2 Exceeding Solutions

Wir haben den Ausgabenplan insofern eingehalten, dass keine außerplanmäßigen Kosten entstanden sind.

Die im Zuwendungsbescheid enthaltenen Materialkosten in der Höhe von 36.110,00€ haben wir nicht ausgeschöpft. Mit der Erstellung der Belegliste für den gesamten Förderzeitraum werden wir einen Betrag von 27.006,54€ abrechnen.

Der Kostenpunkt „sonstige unmittelbare Vorhabenkosten“ können wir nicht abrechnen. Die Arbeiten für diese Arbeiten/Schulungen haben wir zur Erfüllung der Arbeiten nicht benötigt. Der Grund war, dass die Bauteile für unsere Zwecke nicht auf dem Markt verfügbar waren. Der Fokus richtete sich auf die Entwicklung der Softwaretestfälle.

Für Kostenpunkt „Reisekosten“ in der Höhe von 5.200,00€ haben und werden wir keine Beträge abrechnen. Die Nutzung des Dienstwagens und dessen Kosten konnten abgerechnet werden.

Die Personalkosten erreichten nicht die volle Zuwendungshöhe. Durch den Wegfall der geplanten Arbeiten für die Bearbeitung von Hardwaretest und den Weggang von Personal entstanden weniger abrechenbare Personalkosten.

Von Exceeding Solutions wurde das folgende Testequipment angeschafft:

Abschirmzelt	Ein Abschirmzelt für 5G-Funk ist ein speziell entwickeltes Zelt, das elektromagnetische Strahlung – insbesondere hochfrequente Signale wie 5G – abschirmt. Es besteht aus leitfähigen Materialien wie versilbertem Stoff oder metallbeschichtetem Gewebe, die Funkwellen reflektieren oder absorbieren.
Antennentechnik	Für die Evaluierung der physischen Datenlayer wurden verschiedene Antennen beschafft und im Hinblick der Tauglichkeit der Verwendung in einer Labor-Umgebung evaluiert.
Software Defined Radios	Ein Software Defined Radio (SDR) ist ein Funkempfänger und -sender, bei dem viele Funktionen, die früher durch Hardware (z. B. Filter, Modulatoren) realisiert wurden, per Software gesteuert werden. Das bedeutet: Ein Großteil der Signalverarbeitung – wie Demodulation, Decodierung oder Frequenzwahl – findet auf einem Computer oder Mikrocontroller statt. SDRs sind dadurch extrem flexibel: Ein einziges Gerät kann unterschiedlichste Funkstandards (z. B. UKW, DAB, GSM, WLAN) empfangen oder senden – einfach durch Wechsel der Software. Sie werden in der Forschung, im Amateurfunk, bei Sicherheitsbehörden und zunehmend auch in der Industrie eingesetzt. Beim verwendeten Modell handelt es sich dabei um ein Ettus USRP B210, welches sich durch seine Leistung und seinen abdeckbaren Frequenzbereich gut für das Vorhaben eignet.

Desktop-PCs	Zum Aufsetzen der entsprechenden Infrastrukturen wurden Mini-PCs auf Basis von x86 Chips und Ubuntu Linux Distributionen beschafft.
-------------	-------------------------------------------------------------------------------------------------------------------------------------

### 3 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Das Vorhaben leistet einen Beitrag zur Entwicklung von Testtools im 5G Bereich und ermöglicht damit eine effiziente Prüfung nach anerkannten Prüf- und Zertifizierungsschemata wie GSMA NESAS und NESAS CCS-GI. Es konnten unterschiedliche Fragestellungen adressiert werden, die sich aus dem komplexen Testansatz nach NESAS CCS-GI ergeben. Die im Laufe des Vorhabens tatsächlich durchgeführten Arbeiten waren allesamt notwendig und sinnvoll, um zu dem gewünschten Ergebnis zu gelangen. Es wurden wesentliche Vorarbeiten getätigt um Umfang und Funktion eines NESAS CCS-GI Testframeworks einschätzen und dieses implementieren zu können. Das Konzept zur Entwicklung einer universellen Testplattform war erfolgreich und der Demonstrator bietet ein fundiertes Rahmenwerk für zukünftige Prüfungen und Weiterentwicklungen.

### 4 Voraussichtlicher Nutzen und Verwertbarkeit

Der voraussichtliche Nutzen und Verwertungsstrategie wird im Folgenden kurz skizziert. Detaillierte Informationen finden sich auch im Sachbericht zum Verwendungsnachweis Teil III der jeweiligen Projektpartner.

#### 4.1 TÜVIT

In einem ersten Schritt kann das entwickelte Testframework in seiner limitierten Form als Demonstrator in kommenden NESAS CCS-GI Prüfungen der TÜV Informationstechnik genutzt werden, um die bereits implementierten SCAS-Tests automatisiert durchzuführen. Das Grundgerüst soll während der Nutzung im NESAS Prüflabor ergänzt und weiterentwickelt werden. Die durchgeführten Hintergrundrecherche bieten einen guten Ansatz für die Implementierung konkreter Testabfolgen.

#### 4.2 Exceeding Solutions

Des Weiteren wurde die strategische weitere Verwendung des Robot Frameworks für Testszenarien im Hinblick auf andere Tests von kritischen Infrastrukturen, die im Kerngeschäft der Exceeding Solutions GmbH durchgeführt werden, evaluiert und als Alternative für Eigenentwicklungen erkannt. So wurden die zyklischen vollautomatisierten Tests der Testplattform der neuen Technischen Richtlinie 03109-1 2.0 (Anforderungen an die Funktionalität, Interoperabilität und Sicherheit, die die Einzelkomponenten in einem intelligenten Messsystem (Smart Metering System)) in Zusammenarbeit mit dem Bundesamt für Sicherheit in der Informationstechnik auf eben jener Basis implementiert.

## 5 Fortschritt bei anderen Stellen

Um Sicherheitsuntersuchungen an 5G-Komponenten durchzuführen und zu verbessern, wurde vom BSI mit dem Aufbau des 5G/6G Security Lab "TEMIS" (Test Environment for Mobile Infrastructure Security) in Freital begonnen. In dem Labor werden Produkte verschiedenen Hersteller eingesetzt und getestet. Während des Vorhabens konnte eine Vernetzung zum Test-Team des BSI aufgebaut und während einer Vor-Ort Besichtigung eine fachliche Diskussion zu Testthemen angestoßen werden. Das Ziel besteht darin, die Zusammenarbeit auch nach Abschluss des Vorhabens weiterzuführen.

Die Firma Montsecure bietet mit ihrem Produkt SEAL<sup>5</sup> seit kurzem eine kommerzielle Testlösung für ausgewählte Produktklassen und SCAS-Tests an.

## 6 Erfolgte und geplante Veröffentlichungen

- Pressemitteilung zum Projektstart. Links zu den Veröffentlichungen:
  - <https://www.tuvit.de/de/aktuelles/pressemitteilungen/pressemitteilungen-detail/article/mantra5g-neues-forschungsprojekt-zwischen-tuevit-und-exceeding-solutions-gestartet/>
  - <https://www.pressebox.de/pressemitteilung/tuev-informationstechnik-gmbh/mantra5g-neues-forschungsprojekt-zwischen-tuevit-und-exceeding-solutions-gestartet/boxid/1172459>
  - [https://www.linkedin.com/feed/update/urn:li:activity:7107968287338061825?updateEntityUrn=urn%3Ali%3Afs\\_feedUpdate%3A%28V2%2Curn%3Ali%3Aactivity%3A7107968287338061825%29](https://www.linkedin.com/feed/update/urn:li:activity:7107968287338061825?updateEntityUrn=urn%3Ali%3Afs_feedUpdate%3A%28V2%2Curn%3Ali%3Aactivity%3A7107968287338061825%29)
  - [https://www.linkedin.com/posts/t-v-nord-group\\_mobilfunk-kritis-5g-activity-7109794814056886272-fZ0c?utm\\_source=share&utm\\_medium=member\\_desktop](https://www.linkedin.com/posts/t-v-nord-group_mobilfunk-kritis-5g-activity-7109794814056886272-fZ0c?utm_source=share&utm_medium=member_desktop)
  - <https://www.xing.com/pages/tuvinformationstechnikgmbh/news> (Posting 14.09.23)
- Veröffentlichung der Website: [www.mantra5g.de](http://www.mantra5g.de)
- Präsentation der Projektidee auf den folgenden DLR-Info-Veranstaltungen
  - Auftaktveranstaltung, 29.08.23
  - Fachaustauschmeeting TSP 03, 17.11.23
  - Zwischenstand und Austauschmeeting, 23.11.23
- Präsentation des Vorhabens auf dem Prüfstellentreffen NESAS in Freital, 10.04.24
- Präsentation des Vorhabens auf der Veranstaltung „5G/6G Forum“ des BSI in Dresden, 26.11.24: [5G/6G Forum: Cybersicherheit und digitale Souveränität - Sicherheit gemeinsam gestalten - cybersicherheit-bsi.de](#)
- Nennung des Vorhabens während der Panel Diskussion der fuse5G Veranstaltung in Bochum, 04.02.25: <https://fuse5g.nl/>
- Veröffentlichungen Exceeding Solutions:
  - Merseburger Digitaltage, 01.09.23
  - Exceeding Days, 13.09.23
  - Abschlussveranstaltung 5G-POUST, 13.11.24

---

<sup>5</sup> <https://montsecure.com/seal/>