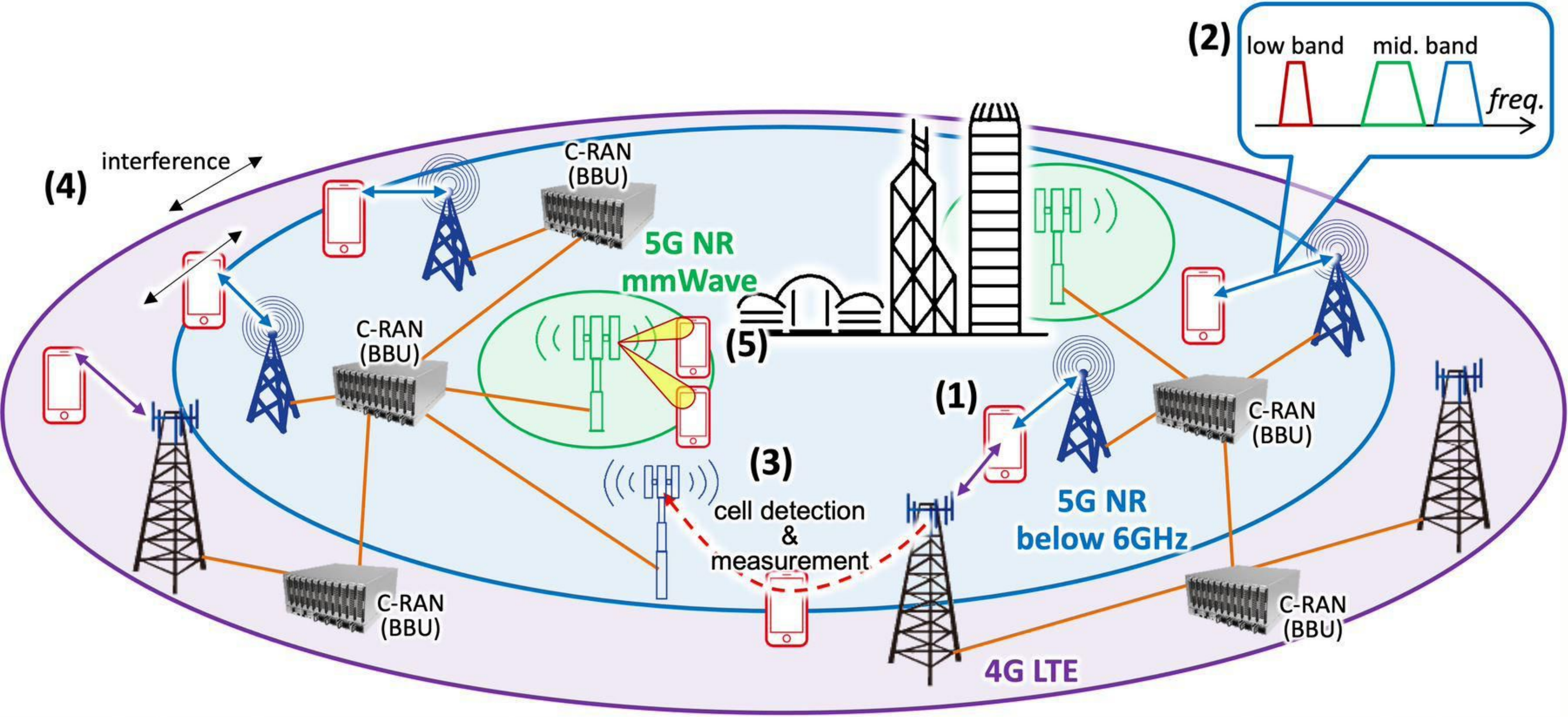
Advancing Cloud-Native 5G: Scalability & Security

By: Marley Willyoung

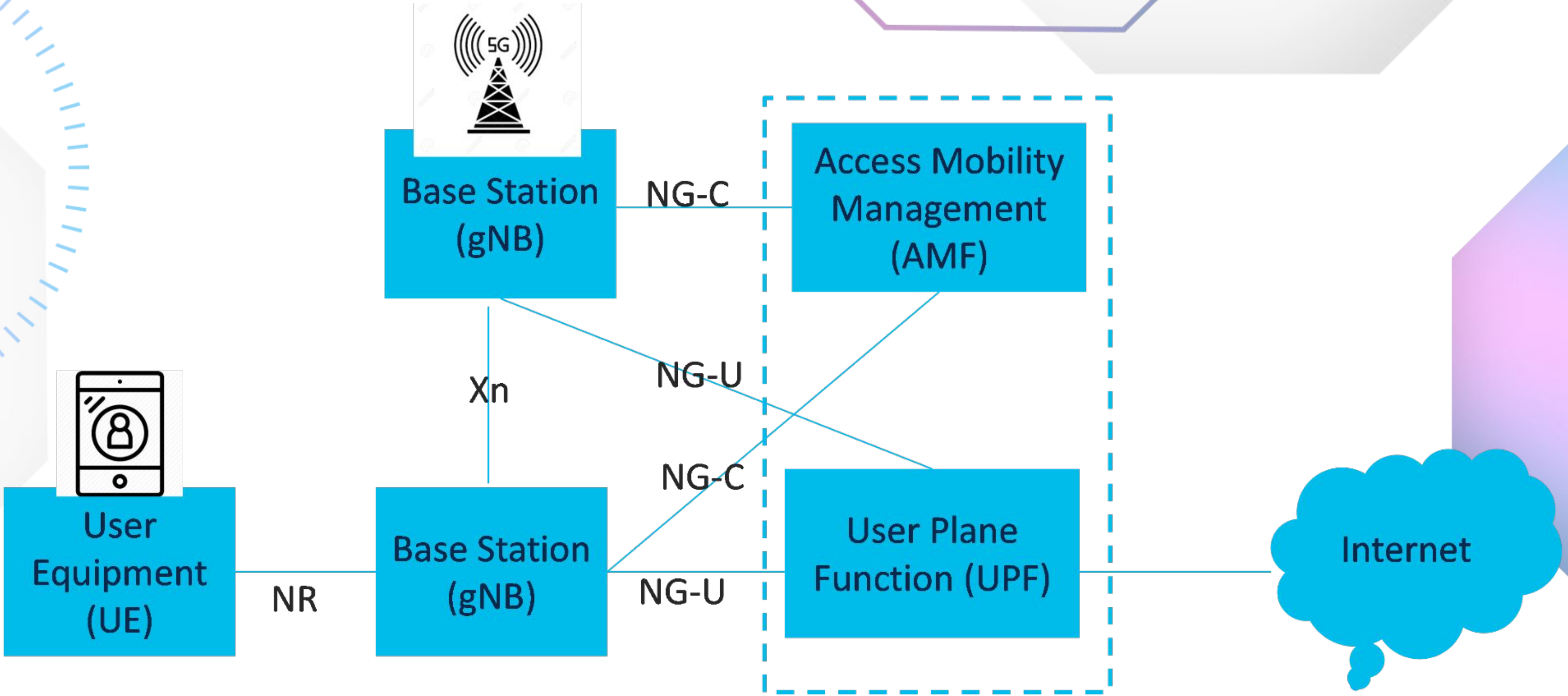
TABLE OF CONTENTS

01

BRIEF PROJECT OVERVIEW



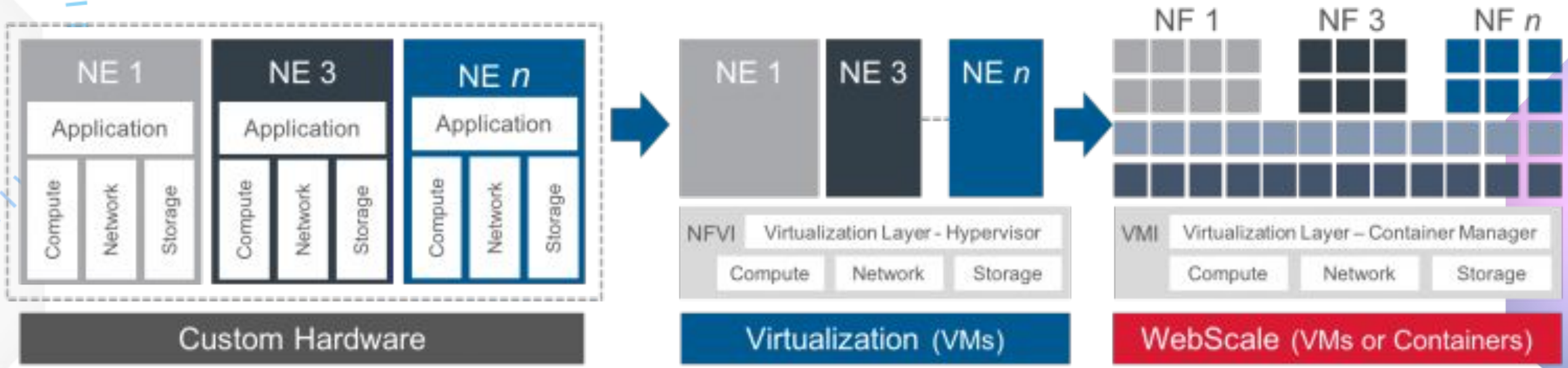
(2)
low band
mid. band
freq.
(4)
interference
C-RAN (BBU)
5G NR mmWave
(5)
(1)
(3)
cell detection & measurement
5G NR below 6GHz
4G LTE



NR = New Radio

NG = Next Generation

Xn = Use Plane Interface



Legacy

- Custom Hardware Platform
- Proprietary
- Costly
- Inflexible

Current

- Monolithic
- Bolted onto Pseudo-Infrastructure
- Vendor Silos
- Still Customized

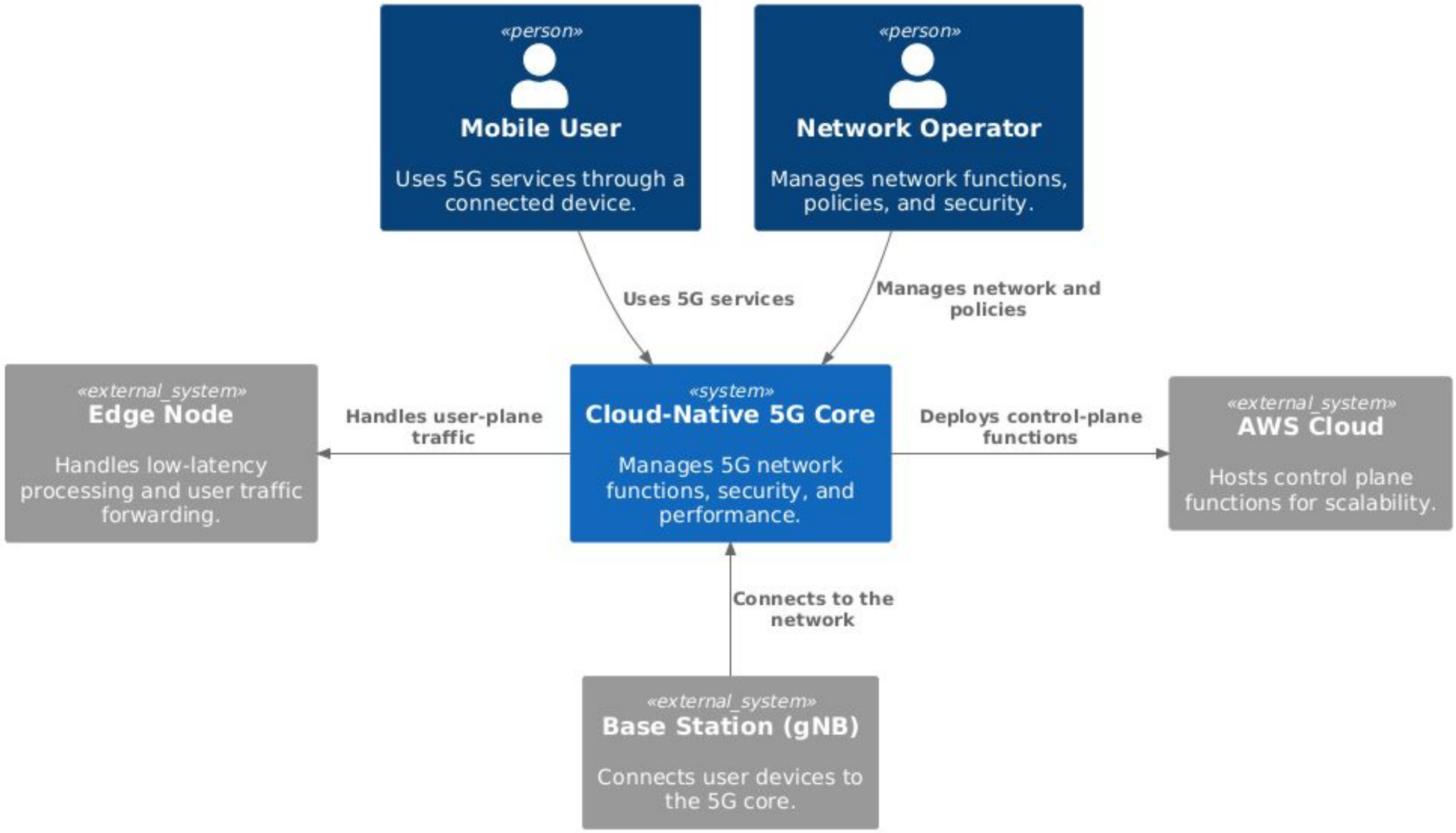
Future

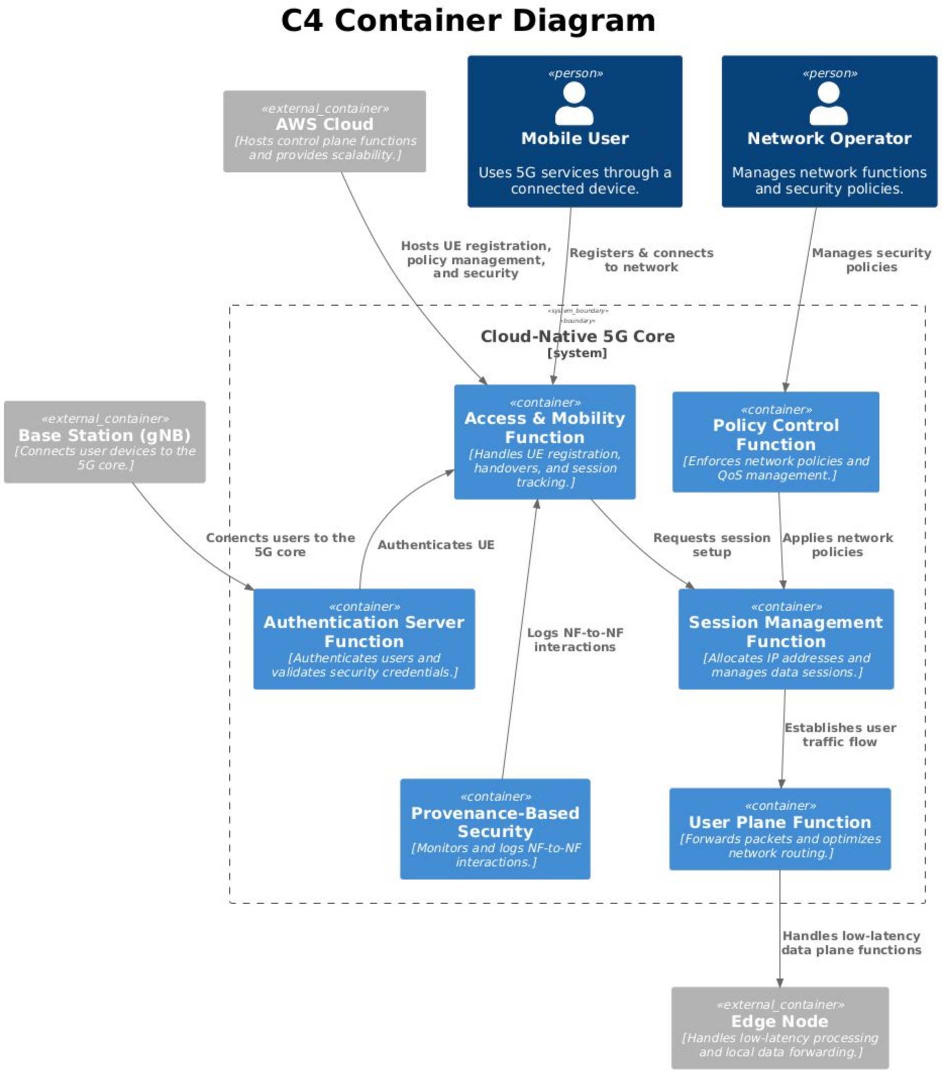
- Fully Programmable
- Hyperscale
- Rapid Innovation

PROJECT GOALS OVERVIEW

- **Security issues in Cloud-Based 5G:** Are slowing 6G progress. Instead of only detecting attacks, containerized NFs prevent lateral movement.
- **Security Visibility (PROV5GC):** Cloud-native microservices increase intrusion detection challenges. We use provenance-based monitoring to track network function (NF) interactions for real-time attack attribution.
- **From VNFs to Cloud-Based 5G Core:** Traditional monolithic 5G cores relied on Virtual Network Functions (VNFs) with limited scaling. We deploy a containerized 5G Core in the cloud to improve scalability and security.
- **Strengthening Base Station Security (BARON):** Current models fail to prevent rogue base station (gNB) attacks; we propose a dynamic trust model instead of static keys for real-time authentication.

C4 Context Diagram





- **AMF, PCF, and SMF** run in the cloud, while UPF is at the edge for low-latency traffic forwarding.
- **AUSF** assigns trust scores to base stations, verifying legitimacy before connections. Provenance-Based Security logs NF interactions for real-time anomaly detection.
- **PCF** enforces security rules and QoS policies, isolating compromised NFs to prevent unauthorized access and lateral attacks.

02

IMPLEMENTATION: CODE & DESIGN

TECHNOLOGIES USED

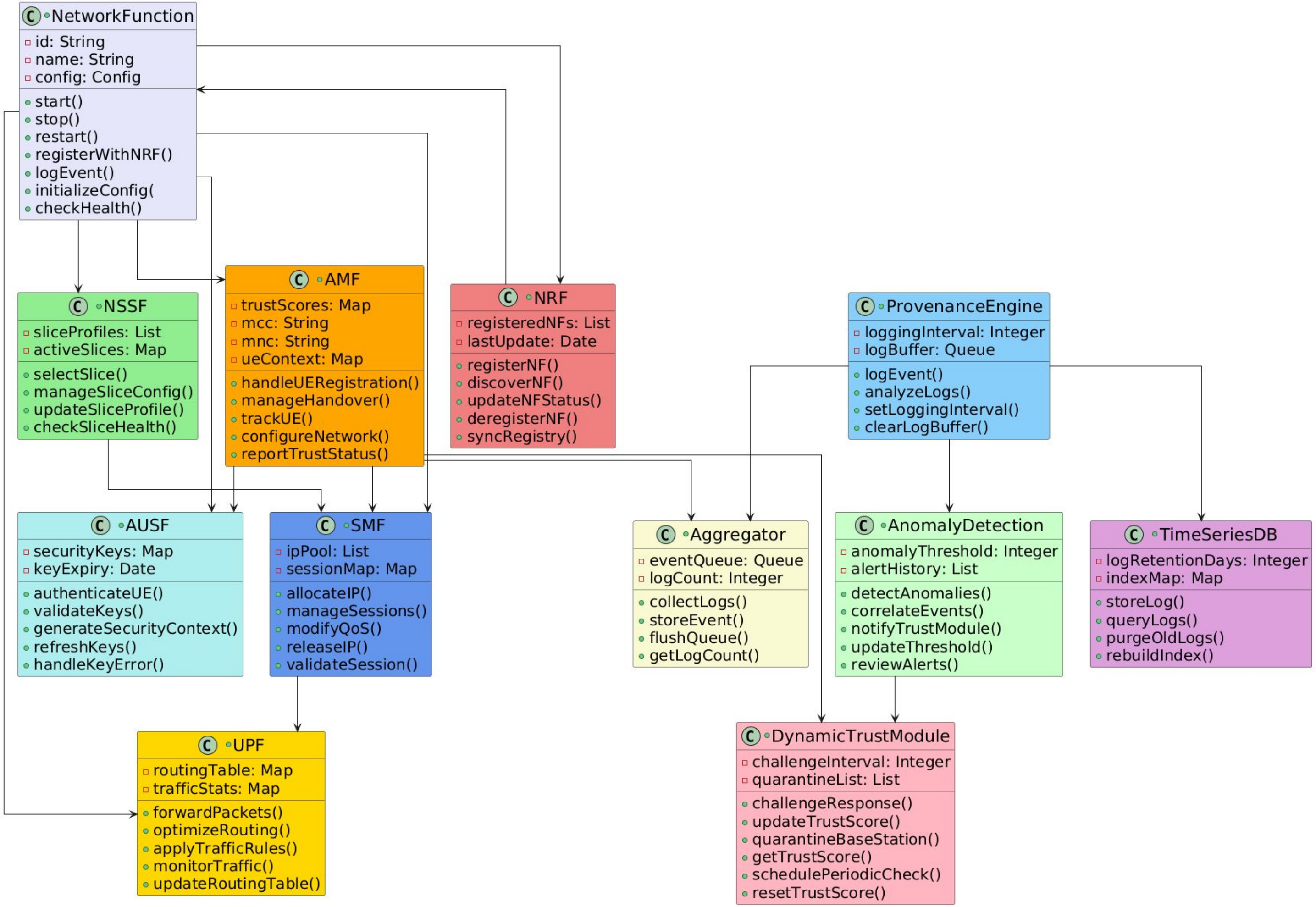
Open5GS

docker

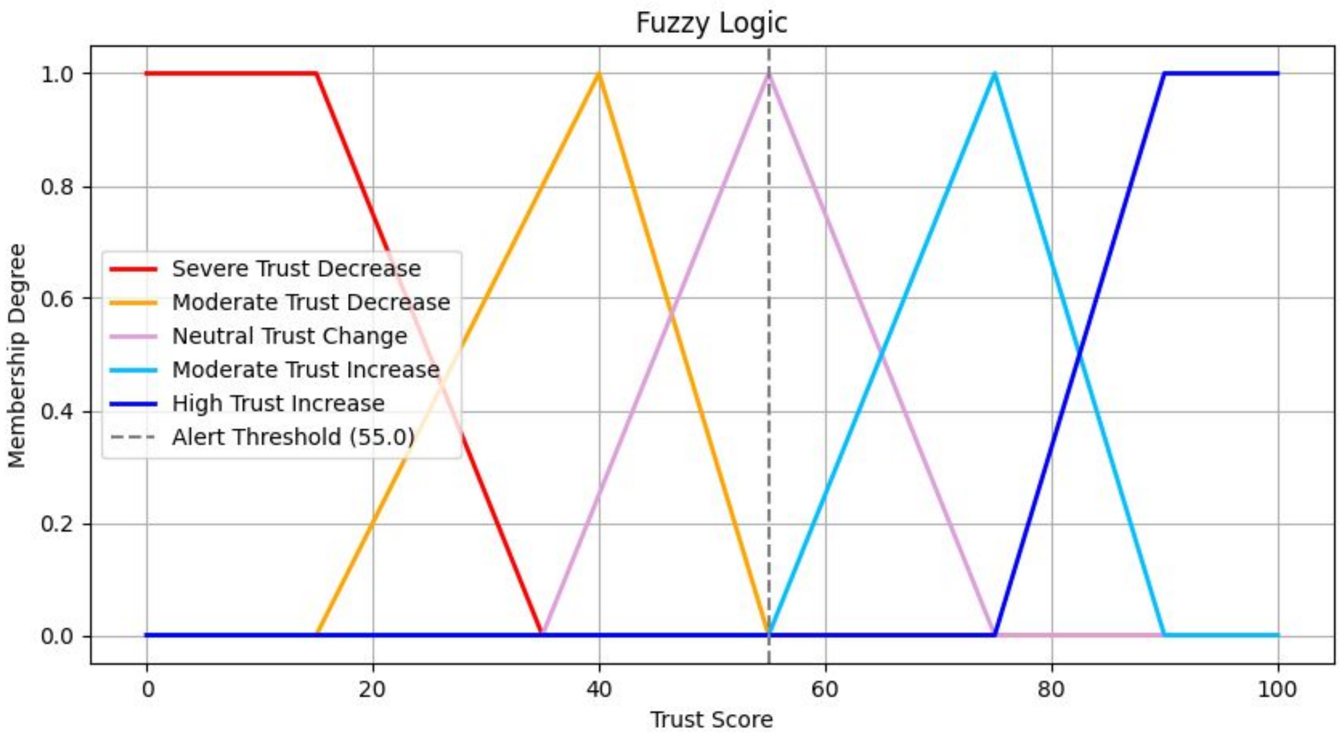
BASH
THE BOURNE-AGAIN SHELL

mongo DB

High Level Class Diagram



CODE & DESIGN


Fuzzy Logic
Severe Trust Decrease
Moderate Trust Decrease
Neutral Trust Change
Moderate Trust Increase
High Trust Increase
Alert Threshold (55.0)
Membership Degree
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
Trust Score

CODE & DESIGN

5 repositories — Last pushed

open5gs Public
Open5GS is a C-language Open Source implementation for 5G Core and EPC, i.e. the core network of LTE/NR network (Release-17)
C · GNU Affero General Public License v3.0 · 805 · 0 · 0 · 0 · Updated 7 hours ago

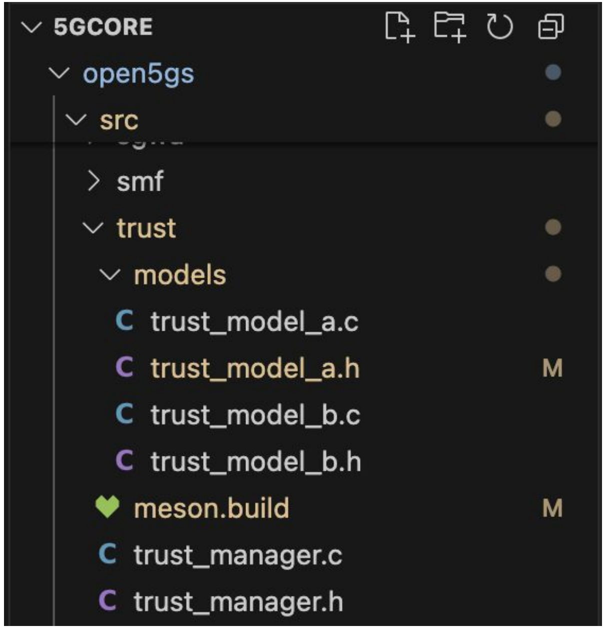
5GCORE Private
Dockerfile · GNU General Public License v3.0 · 0 · 0 · 0 · 0 · Updated 7 hours ago

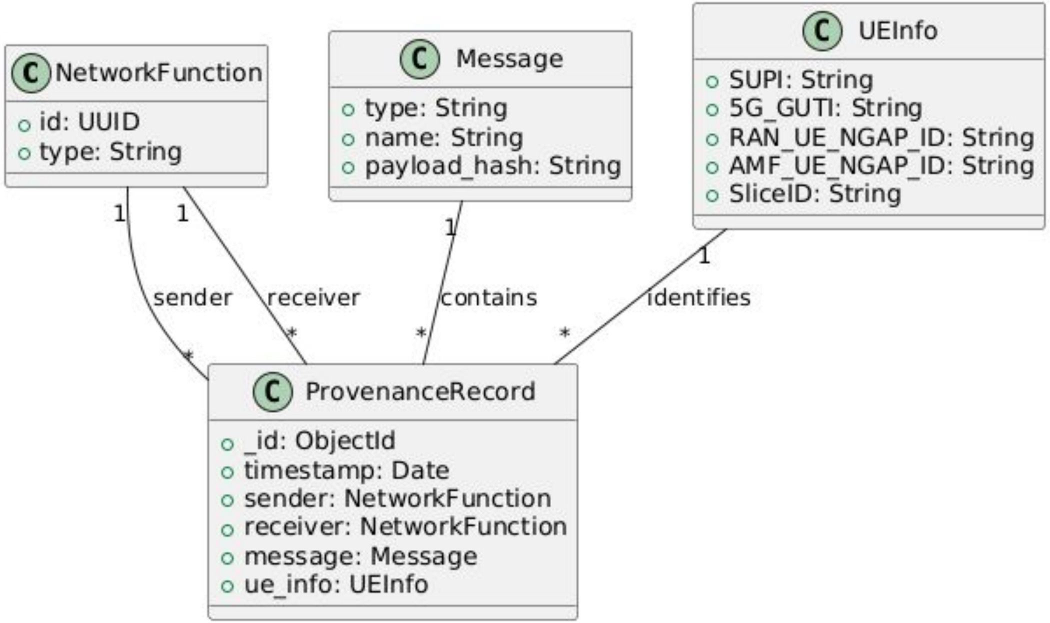
.github-private Private
0 · 0 · 0 · 0 · Updated 17 hours ago

.github Public
0 · 0 · 0 · 0 · Updated 17 hours ago

UERANSIM Public
Open source 5G UE and RAN (gNodeB) implementation.
C++ · GNU General Public License v3.0 · 341 · 0 · 0 · 0 · Updated on Feb 12

CODE & DESIGN

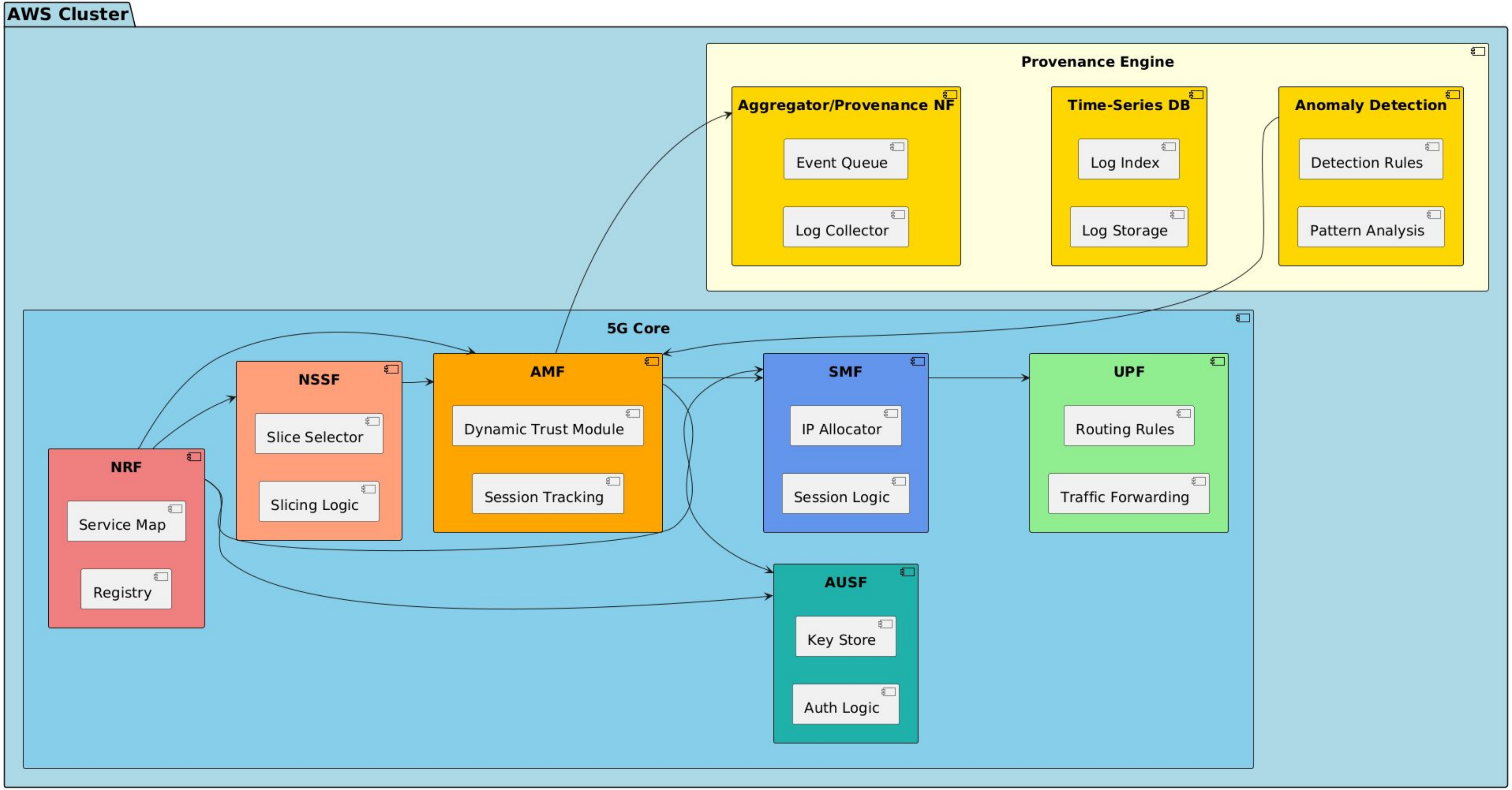

5GCORE
open5gs
src
smf
trust
models
trust_model_a.c
trust_model_a.h
M
trust_model_b.c
trust_model_b.h
meson.build
M
trust_manager.c
trust_manager.h


NetworkFunction
id: UUID
type: String
Message
type: String
name: String
payload_hash: String
UEInfo
SUPI: String
5G_GUTI: String
RAN_UE_NGAP_ID: String
AMF_UE_NGAP_ID: String
SliceID: String
1
1
1
1
sender
receiver
contains
identifies
*
*
*
*
ProvenanceRecord
_id: ObjectId
timestamp: Date
sender: NetworkFunction
receiver: NetworkFunction
message: Message
ue_info: UEInfo

03

SYSTEM OVERVIEW & DEMONSTRATION

System Overview



DEMO

04

DATA ANALYSIS & DISCUSSION

OUTPUT GENERATION

```
Starting gNB...
160a5d186fbb5905c90e821b95aa6e94bd47e05762c15ee262305a70b2656fdd
Waiting for gNB to connect...
gNB connected successfully

Run 1 of 5
Authentication successful in 0.455 seconds

Run 2 of 5
Authentication successful in 0.414 seconds

Run 3 of 5
Authentication successful in 0.441 seconds

Run 4 of 5
Authentication successful in 0.443 seconds
```

COMPARISON OF OUTPUT VS. HYPOTHESIS

- **Trust Score Adaptation:** >92% accuracy, trust recalibrated every 200ms, URANSIM attacks blocked in average 5s.
- **Latency & Load:** 8.7ms avg. authentication, 3.6% CPU overhead at 500 UEs.
- **Anomaly Detection:** 94% attack detection in <3ms, 96% attribution accuracy.
- **Resource Overhead:** Provenance logging added >12% CPU, peak deviation +0.8%.

DISCUSSION: RESEARCH CHALLENGES

- **Dynamic Resources:** Variable CPU, memory, network allocation causes unpredictable performance.
- **Network Variability:** Latency fluctuations and multi-cloud deployments complicate consistent testing.
- **Multi-Tenancy & Orchestration:** Shared infrastructure and diverse cloud tools create test inconsistencies.
- **Security & Emulation Complexity:** Real-time anomaly detection overhead and difficulty simulating realistic attacks.

ABNORMAL CASE EXCEPTION

- **UE Registration Storm:** Large scale registrations via UERANSIM overload Open5GS AM; delayed or failed authentications.
- **Provenance Logging Overload:** Excessive logs flood MongoDB during intensive simulations; ineffective trust-score calculations.
- **False Positives in Anomaly Detection:** Legitimate rapid UE handovers incorrectly flagged as anomalies due to overly sensitive fuzzy logic parameters, resulting in unnecessary isolation actions.
- **Container Failures:** Autoscaler fails to launch additional Open5GS containers and modules quickly enough under high traffic simulations, causing degraded QoS and dropped sessions.

05

CONCLUSIONS & RECOMMENDATIONS

SUMMARY & CONCLUSIONS

- Dynamic trust model significantly enhanced 5G authentication accuracy & low CPU overhead and minimal latency impact.
- Trust adaptation effectively detected and mitigated security anomalies within cloud-native, containerized network functions.
- Cloud deployment variability (dynamic resources, latency fluctuations, multi-tenancy) introduced challenges in consistent benchmarking and reproducible performance tests.
- Fuzzy logic provided optimal balance in trust-scoring accuracy, computational efficiency, and rapid anomaly response, validated by prior studies in SCADA and IoT environments.

RECOMMENDATIONS FOR FUTURE STUDIES

- Prioritize establishing a stable Open5GS core early in project lifecycle for integration of security frameworks and reduce complexity in deployment.
- Develop dedicated localized testbeds for extensive validation and refinement of dynamic trust models prior to scaling deployments.
- Optimize dynamic trust scoring using fuzzy logic with lightweight computational overhead to enable effective real-time security decisions, integrated within CI/CD pipelines.
- Perform extensive scalability tests with diverse node/subscriber counts and realistic attack scenarios, and investigate suitable frameworks or protocols for accurate security-threat emulation.

06

APPENDICES & FINAL NOTES

PROGRAM SOURCE CODE & DOCUMENTATION

- Containerized microservices, MongoDB provenance tracking
- Dynamic trust model, automated threat response
- Code referenced in paper, available via GitHub ->

REFERENCES

BARON: Base-Station Authentication Through Core Network for Mobility Management in 5G Networks Alessandro Lotto, Vaibhav Singh, Bhaskar Ramasubramanian, Alessandro Brighente, Mauro Conti, Radha Poovendran. *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '23) Guildford, United Kingdom.*3558482.3590187

PROV5GC: Hardening 5G Core Network Security with Attack Detection and Attribution Based on Provenance Graphs Harsh Sanjay Pacherkar, Guanhua Yan. *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '24) Seoul, Republic of Korea.*3643833.3656129

SECONDARY REFERENCES

CCSM: Cross-Cluster Security Models for Edge-Core Kubernetes Environments Mahmood GholipourChoubeh, Hugo Kermabon-Bobinnec, Sima Bagheri, Suryadipta Majumdar, Yosr Jarraya, Makan Pourzandi, Lingyu Wang.*(CODASPY '24), June 19–21, 2024, Porto, Portugal.*3626232.3653253

L25GC: A Low Latency 5G Core Network based on High-Performance NFV Platforms *SIGCOMM '22, August 22–26, 2022, Amsterdam, Netherlands.*

Thank You

Please ask any Questions.