NS2 Network and System Security Laboratory KAIST

ORANalyst:
Systematic Testing Framework for Open RAN Implementations

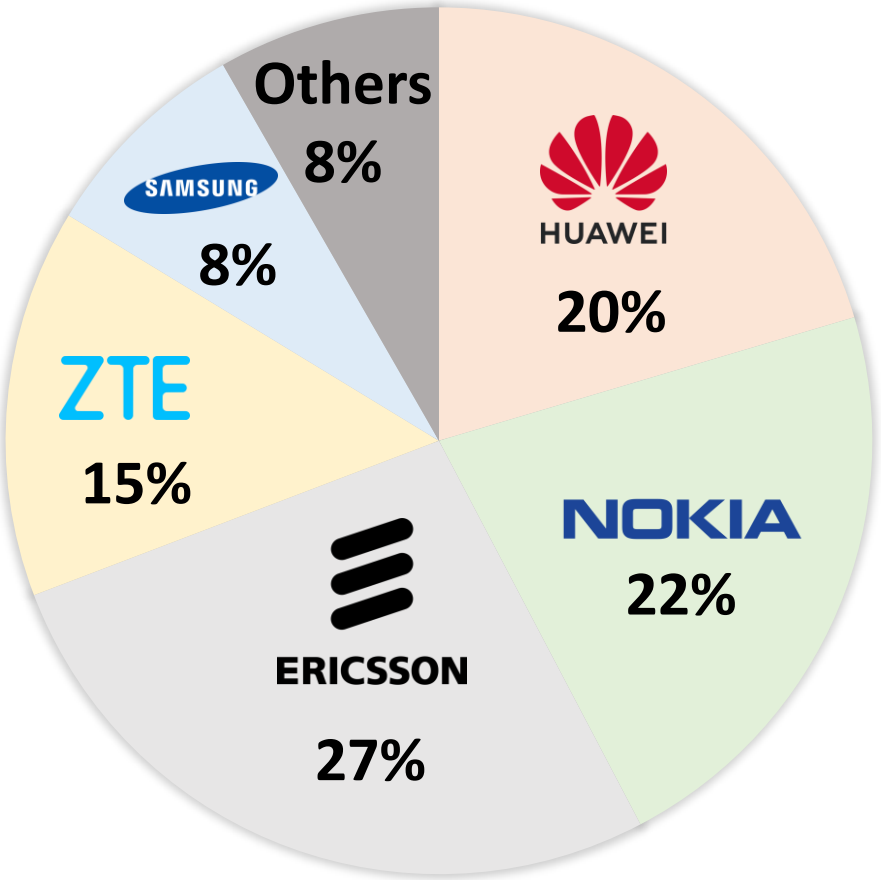
Tianchang Yang, Syed Md Mukit Rashid, Ali Ranjbar, Gang Tan, and Syed Rafiul Hussain

USENIX Security ‘24

Presenter: Isu Kim, 27 Nov. 2024

Open RAN (O-RAN)

WORLD RAN MARKET SHARE 2021 [1]



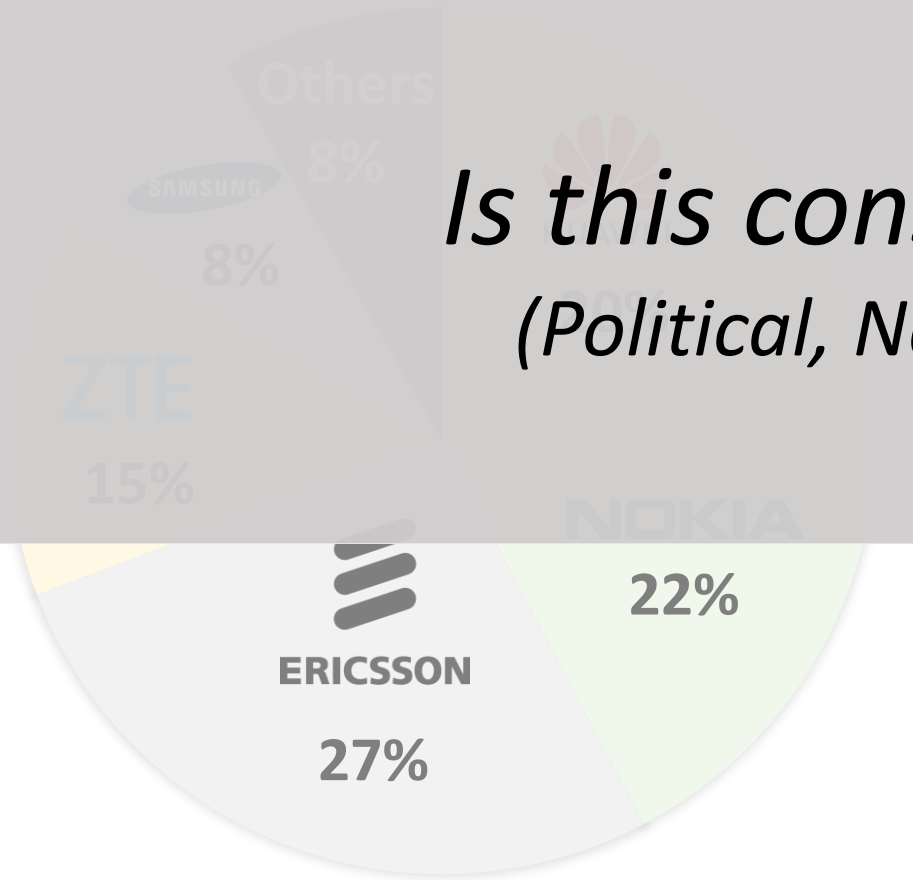
Company	Country of Origin
SAMSUNG	Republic of Korea
ZTE	People's Republic of China
HUAWEI	People's Republic of China
NOKIA	Finland
ERICSSON	Sweden

50.0+ Billion Dollars 2021

[1] https://techblog.comsoc.org/2022/01/25/mobile-experts-ericsson-1-in-ran-market-huawei-falls-to-3/

Open RAN (O-RAN)

WORLD RAN MARKET SHARE 2021 [1]



Company	Country of Origin
SAMSUNG	Republic of Korea
ZTE	People's Republic of China
HUAWEI	People's Republic of China
NOKIA	Finland
ERICSSON	Sweden

Is this consolidation okay?

(Political, National, Economical)

50.0+ Billion Dollars 2021

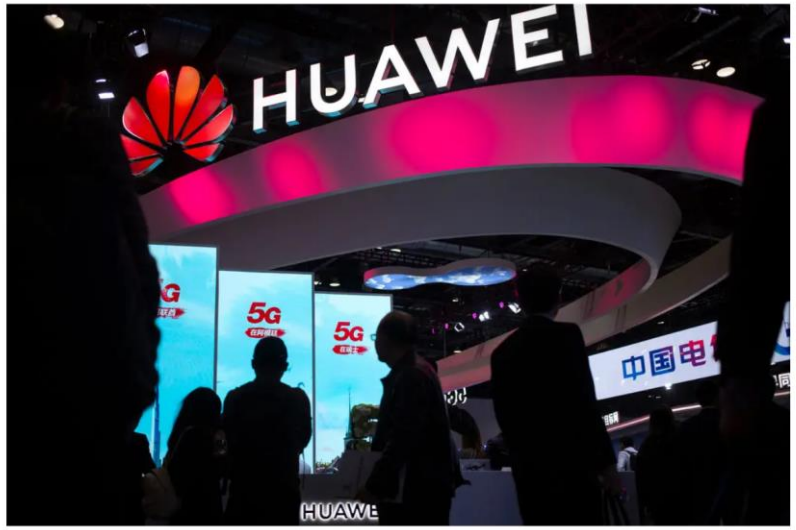
[1] https://techblog.comsoc.org/2022/01/25/mobile-experts-ericsson-1-in-ran-market-huawei-falls-to-3/

Open RAN (O-RAN)

White House Official Says Huawei Has Secret Back Door to Extract Data

The allegation that Huawei maintains access to the data that flows through its network is the latest step in a campaign to thwart the Chinese telecom giant's rise.

Share full article



BROADCOM Products Solutions Support and Services Company How To Buy Support Portal English Search

Support and Services / Symantec Security Center / Virus Definitions & Security Updates / Attack Signatures / Attack: ZTE Router Backdoor Activity

Print Share Page

Attack: ZTE Router Backdoor Activity

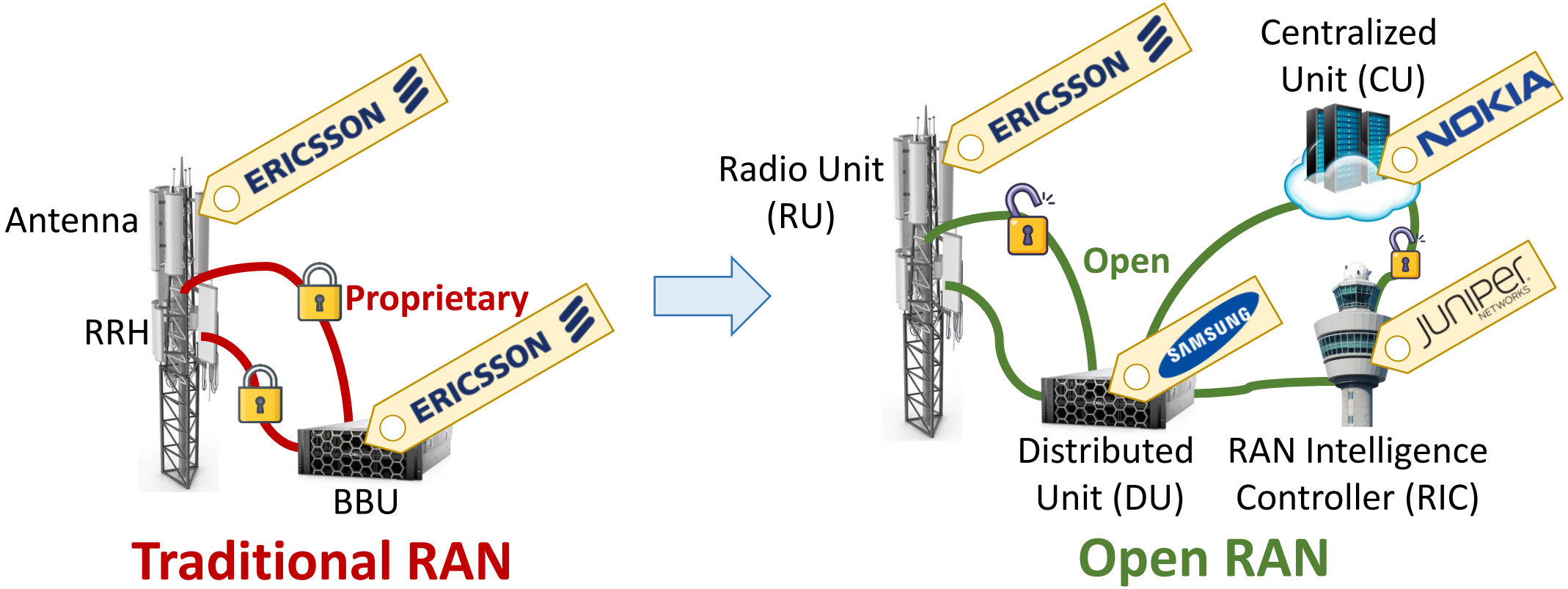
Severity:High

This attack could pose a serious security threat. You should take immediate action to stop any damage or prevent further damage from happening.

Ericsson: The spiral of lies that cost the Swedish telecom giant dearly

Entangled in a corruption scandal, the equipment manufacturer will pay a new fine of nearly €200 million. 'Le Monde' reports on how the company hindered the work of the US justice system, particularly in Iraq.

Open RAN (O-RAN)



"Break vendor lock-in by open interfaces"

Open RAN (O-RAN)

Microsoft Maintains Open RAN Momentum

NEWS 04 January 2023 | 3 minute read

Written by Dan Meyer, Executive Editor, SDX Central

Microsoft is developing a radio access network (RAN) analytics and control technologies targeted at supporting virtualized RAN (vRAN) gear from third-party vendors running on Microsoft's Edge platforms and builds on the industry's broader work on a RAN intelligent controller (RIC) specifications.

CATEGORIES AND TAGS

Telecoms

International

OPEN RAN

WIRELESS NETWORKING TECH

DIVERSIFICATION

KT, 제주도 5G 망에 오픈랜 시스템 구축

이경탁 기자

입력 2024.10.17. 10:02

SKT, 한국 오픈랜 장비·기술력 글로벌 무대에 알렸다

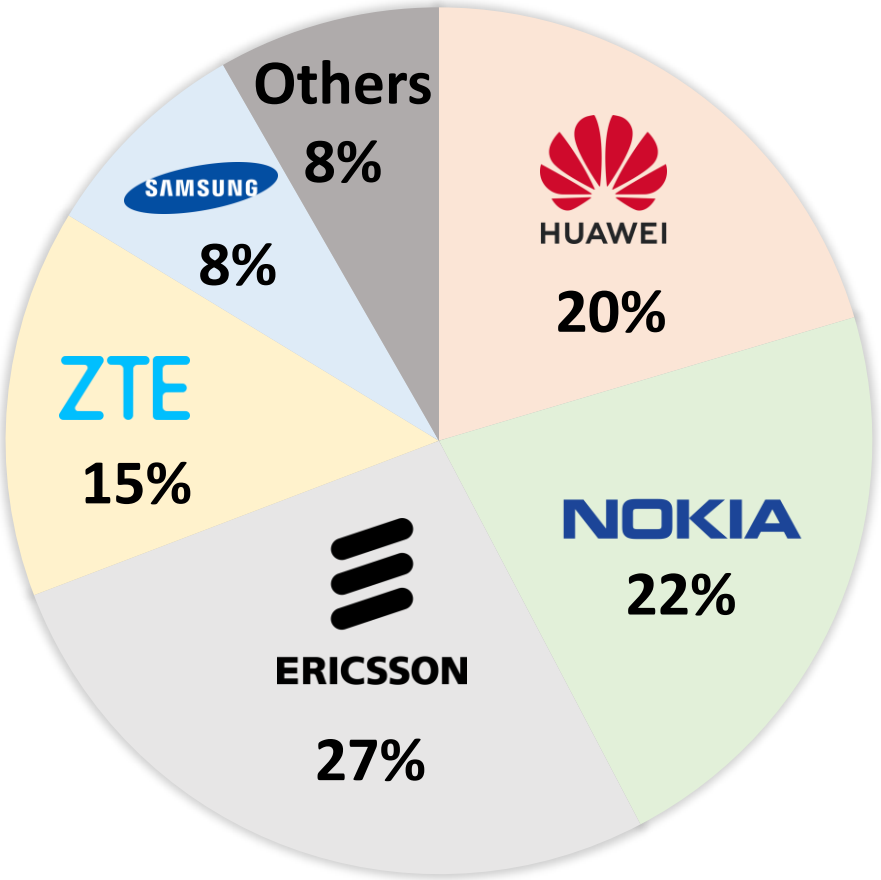
Telecommunications

Google joins the O-RAN ALLIANCE to advance telecommunication networks

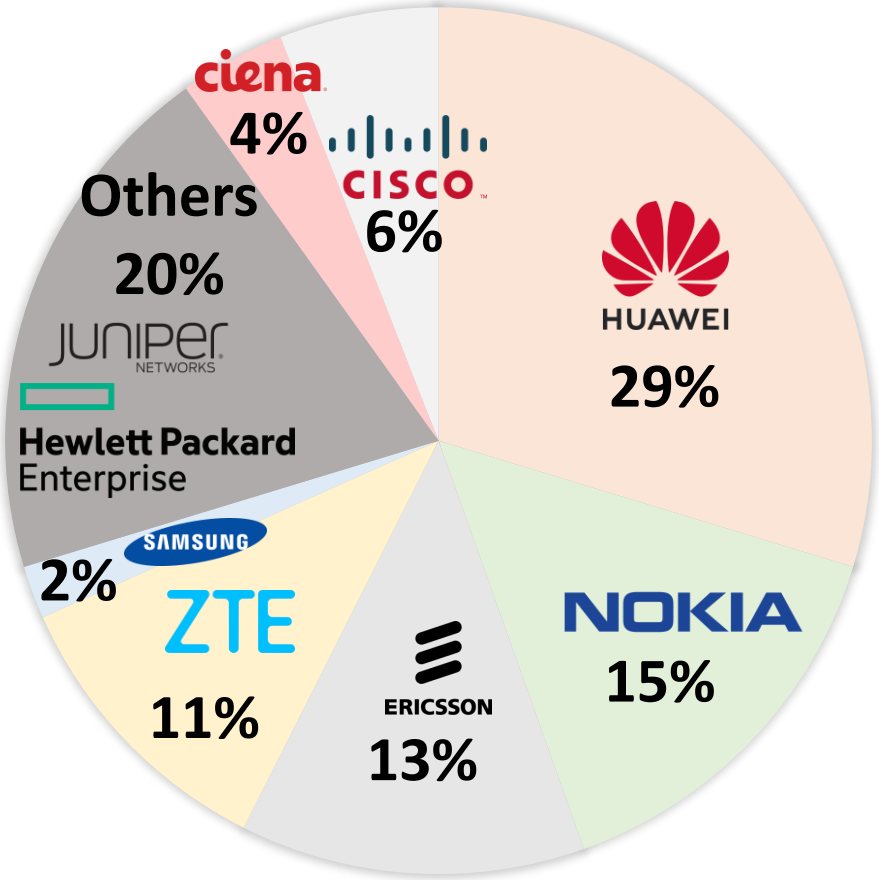
June 29, 2021

Open RAN (O-RAN)

WORLD RAN MARKET SHARE 2021



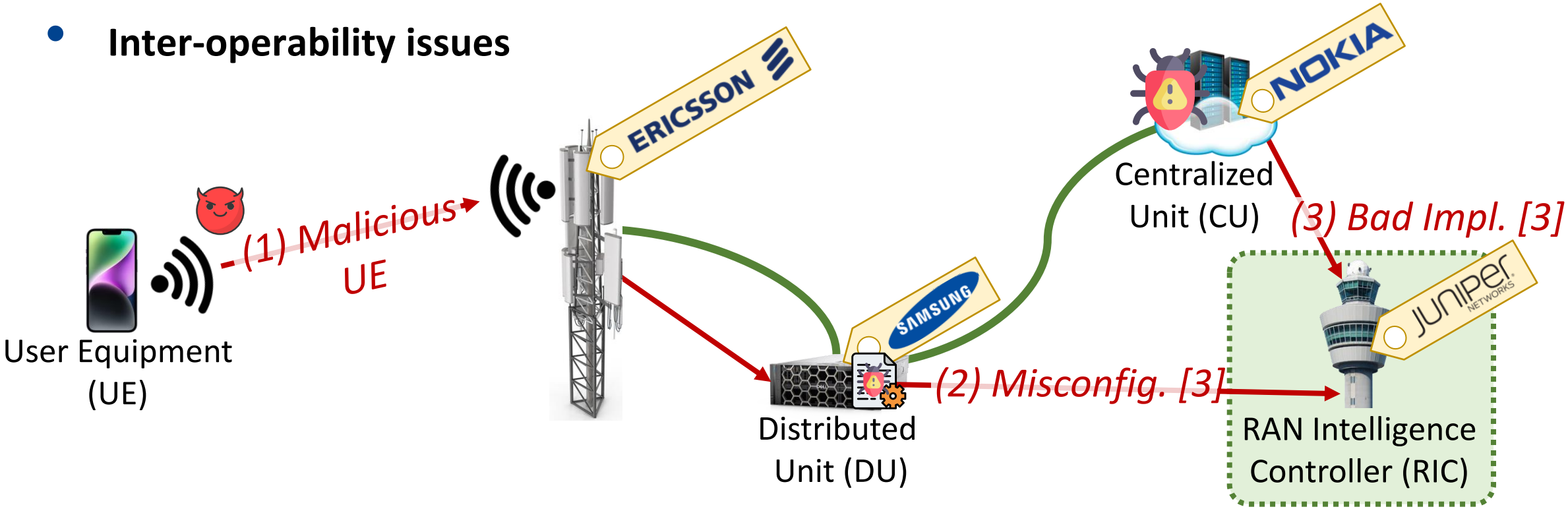
WORLD RAN MARKET SHARE 2023 [2]



[2] https://techblog.comsoc.org/category/ran-market/

Vulnerabilities

- **Inter-operability issues**



"Must ensure that RICs are robust against malicious and unexpected inputs"

[3] LTE security disabled: Misconfiguration in commercial networks

Backgrounds

- **RAN Intelligence Controller (RIC)**

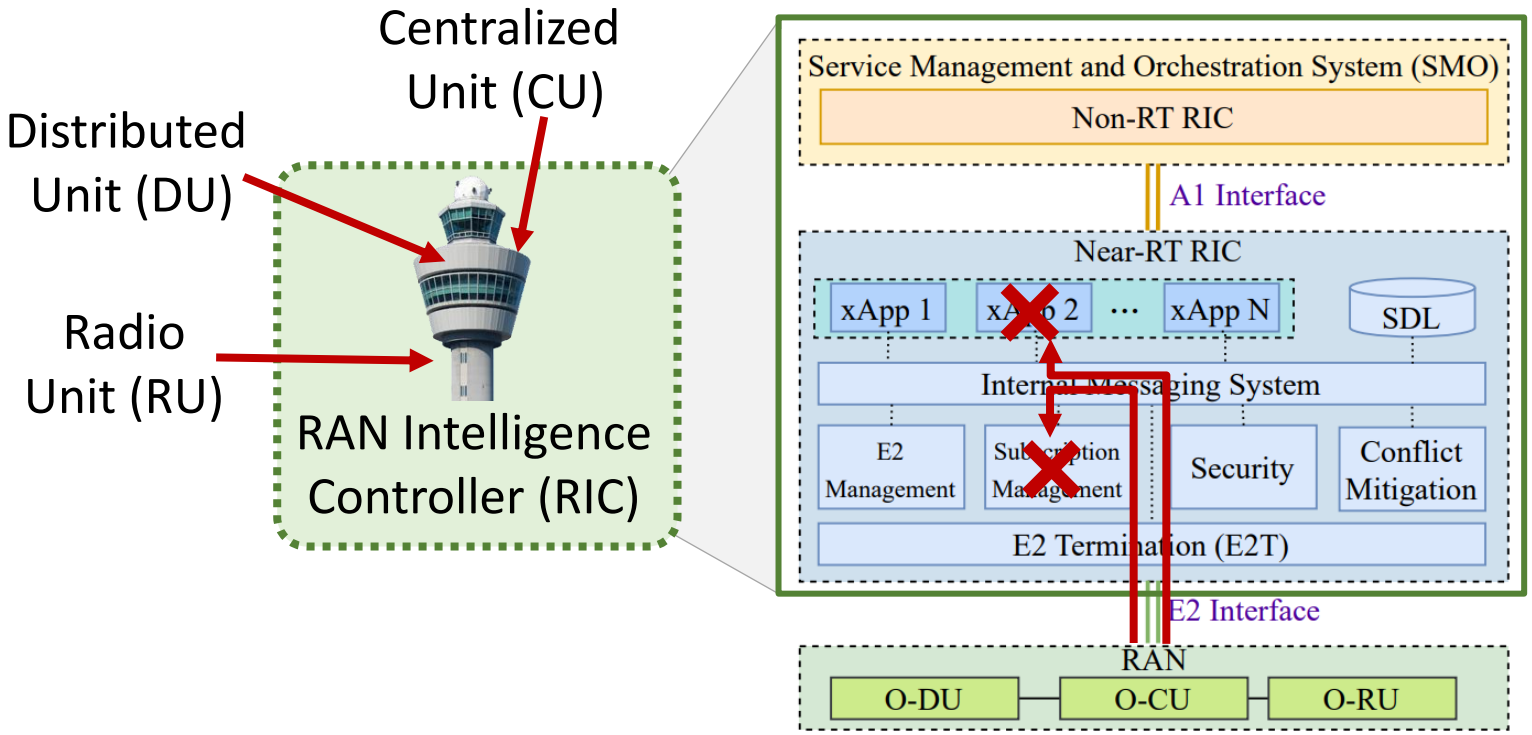


Figure 1: O-RAN RIC Architecture

- Software-centric, service-based, disaggregated architecture
- Each xAPPs can be from a 3rd party
- No standards on internal messaging
  - *gRPC? REST API?*

O-RAN.WG11.Security-Near-RT-RIC-xApps-TR.0-R003-v05.00

6.17 Solution #16: Additional security measures for the E2 interface

6.17.1 Introduction

The Near-RT RIC receives Near real-time information from the E2 Nodes across the E2 interface. While the E2 interface is considered secure with controls that provide confidentiality, integrity, and mutual authentication, the Near-RT RIC should not assume that the data received is valid and trusted. The Near-RT RIC should provide built-in security compliant with a zero-trust architecture based upon the principle that perimeter security is insufficient to protect against internal threats.

Backgrounds

- **RAN Intelligence Controller (RIC)**
  - Software-centric, service-based, disaggregated architecture
  - Each xAPPs can be from a 3rd party
  - No standards on internal messaging

[Research Question]

*"Can we develop an **automated reasoning framework** to analyze the **robustness and operational integrity** of **O-RAN implementations**, providing high-security assurances prior to their commercial deployments?"*

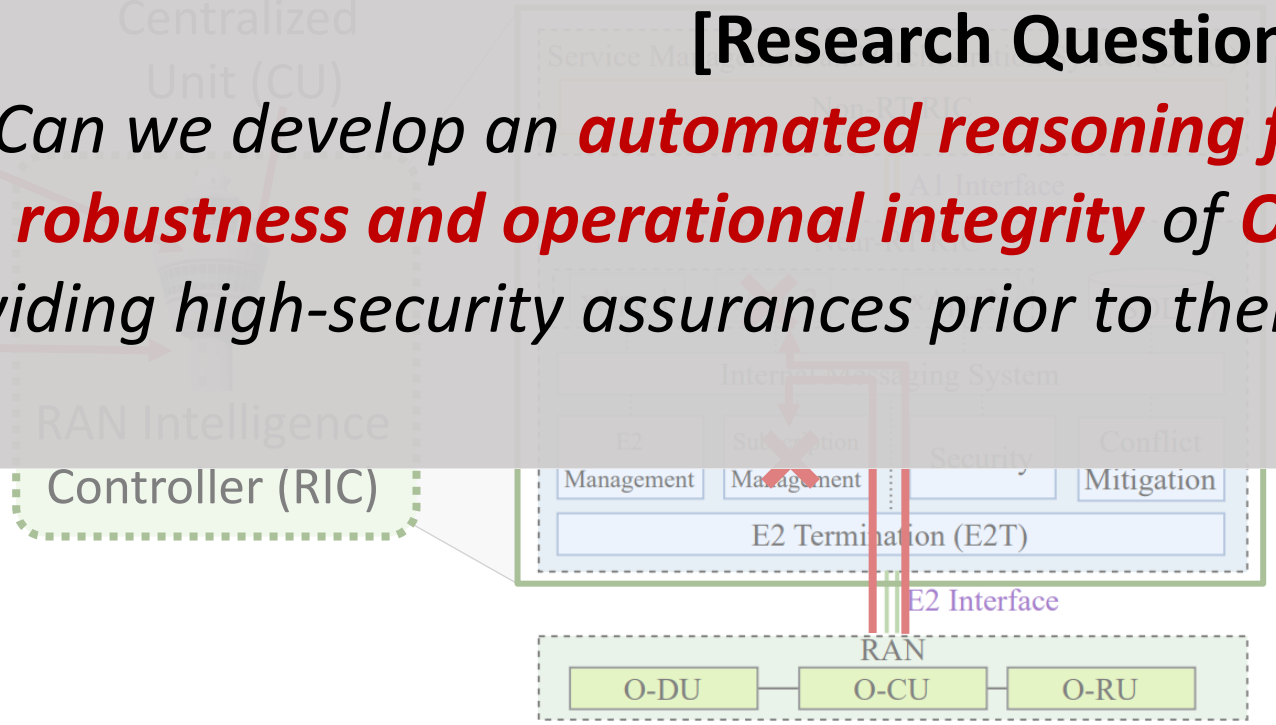


Figure 1: O-RAN RIC Architecture

O-RAN

6.17 Solution #16: Additional security measures for the E2 interface

6.17.1 Introduction

The Near-RT RIC receives Near real-time information from the E2 Nodes across the E2 interface. While the E2 interface is considered secure with controls that provide confidentiality, integrity, and mutual authentication, the Near-RT RIC should not assume that the data received is valid and trusted. The Near-RT RIC should provide built-in security compliant with a zero-trust architecture based upon the principle that perimeter security is insufficient to protect against internal threats.

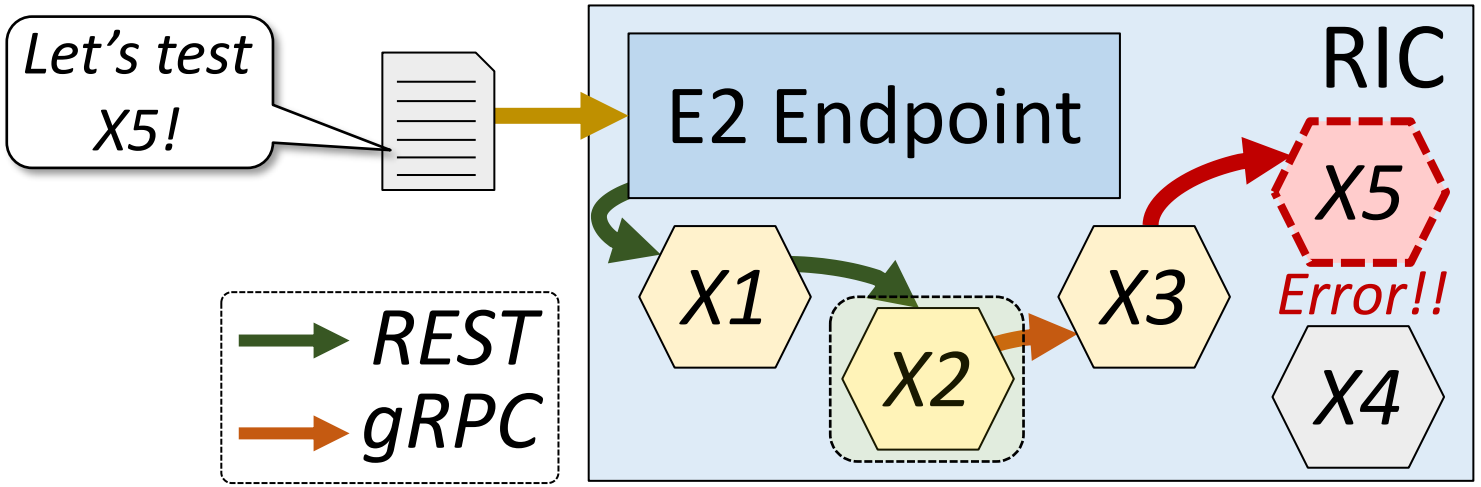
O-RAN Testing

- **Existing testing methods**
  - Fails to provide interconnected insights
  - Does not support O-RAN connections (SCTP)

Fuzzer Category	Examples	Remarks
General	AFL, LibFuzzer, Driller	• Monolithic command-line apps only
Protocol	AFLNET, BooFuzz, Peach	• Testing individual servers only • Labor-intensive and error-prone task
Microservice	Evomaster RPC	• Manual driver code creation
API	Restler, Evomaster	• Depends on analyzing response messages

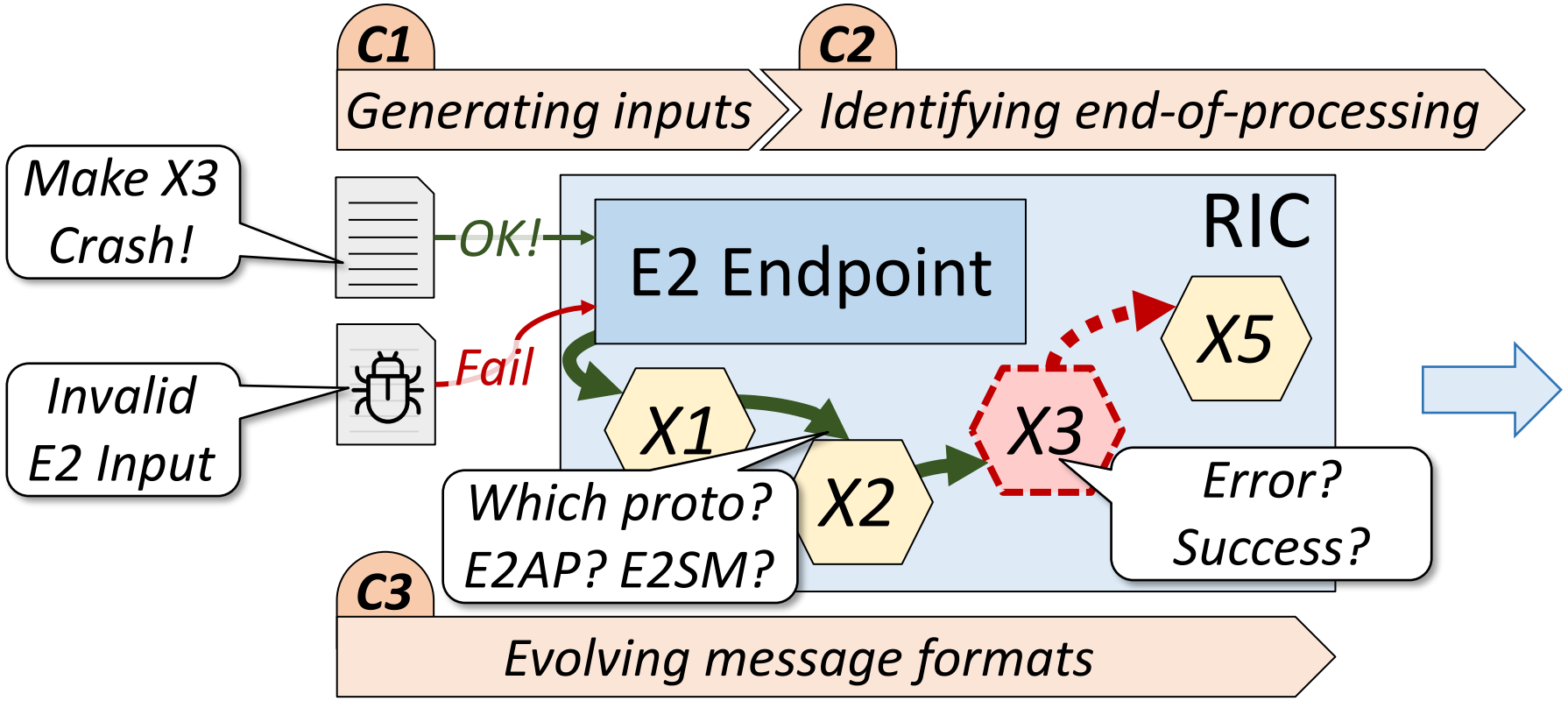
ORANALYST - Motivation

- **ORANalyst – An end-to-end testing framework**
  - Testing in isolation can…
    - Be too labor-intensive making stubs
    - Make unrealistic inputs, resulting false positive
  - RIC communications are unspecified (gRPC? REST API?)



Testing only C2?

ORANALYST - Challenges

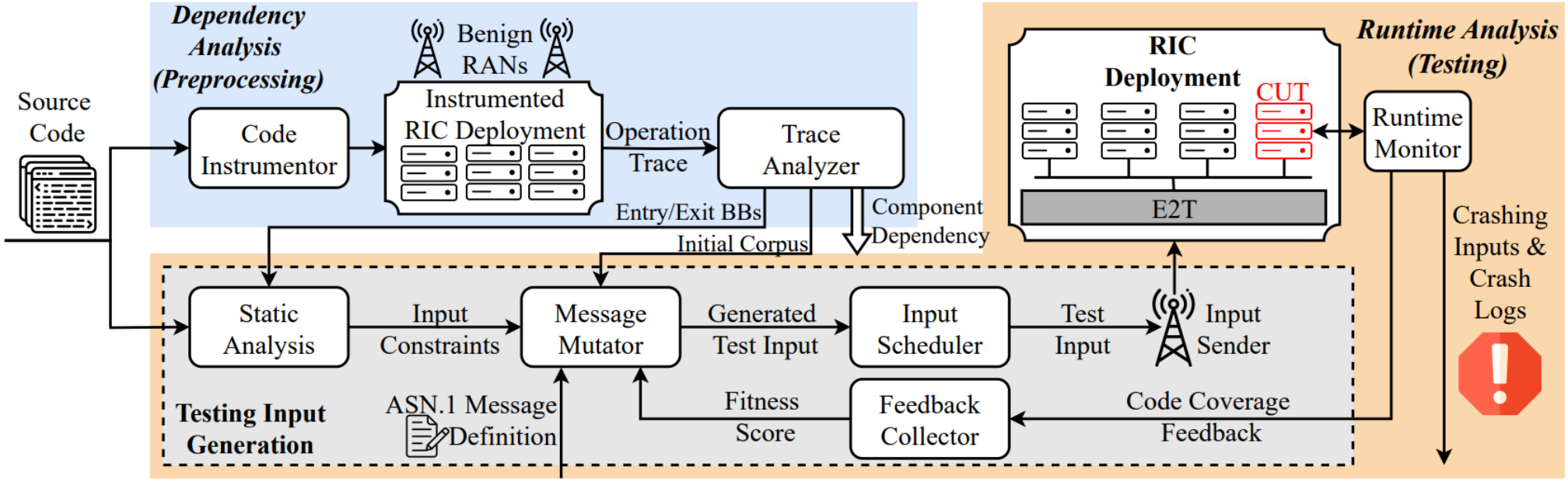


As the fuzzing terms…

- **POET**: C1, C3
- **Courier**: E2 Endpoint
- **Oracle**: C2

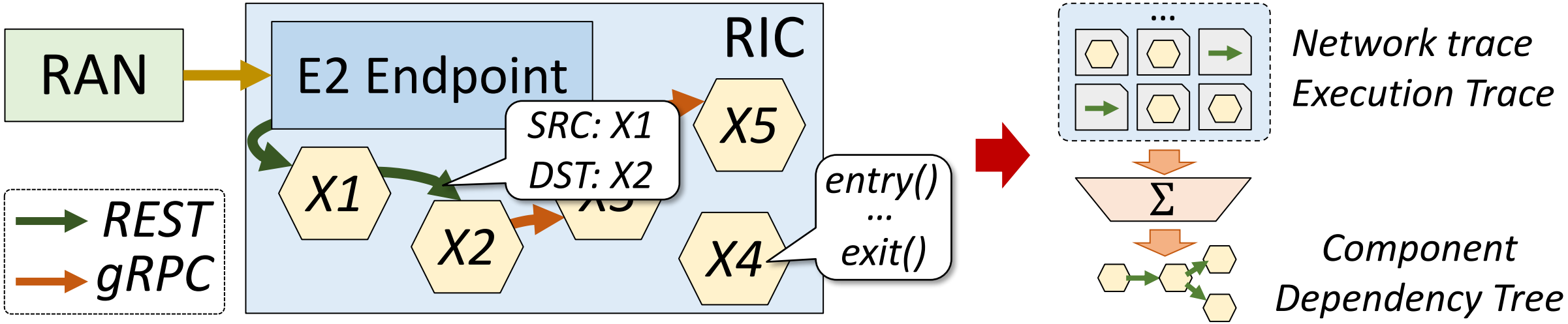
ORANALYST - Design

- **Overview:**
  - Goal: end-to-end, grammar-guided, feedback-driven fuzzing framework
  - Two stage operation: *"dependency analysis"* and *"runtime testing"*



ORANALYST - Design

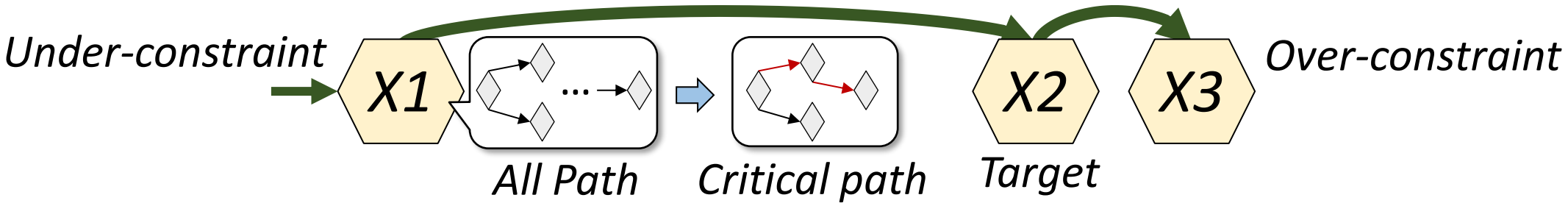
- **ORANalyst – Dependency analysis (C1)**
  - Static analysis can't find the inter-component information flow via network
  - Collect network traffic and execution information for 24 hours of RIC in with benign RAN



"Capture flow of all message types and construct a dependency tree"

ORANALYST - Design

- **ORANalyst – Input constraint generation (C1)**
  - Construct Program Dependency Graph (PDG) [4]
    - Control Dependency Graph (CDG) and Data Dependency Graph (DDG)
    - There are limited number of paths that actually contribute → Critical Path



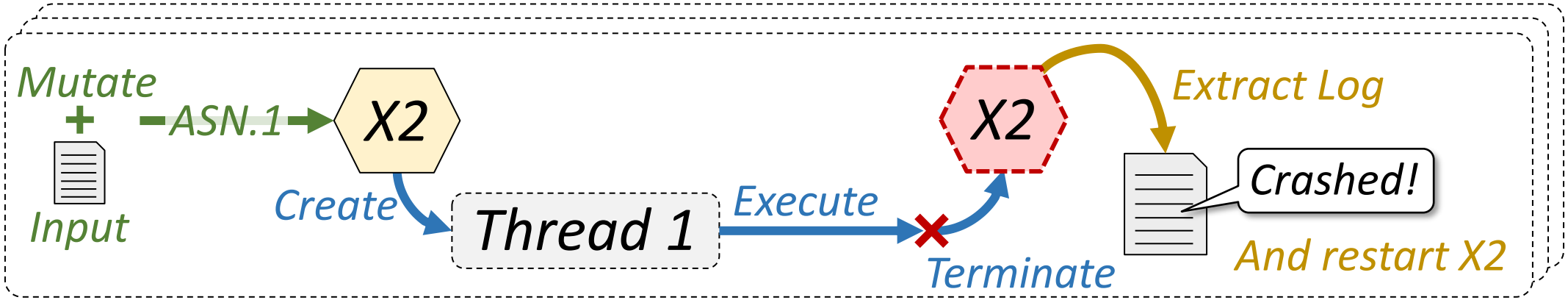
  - Using path conditions, generate input "constraints" for each components

"With critical path and input loops, we can find out the target component"

[4] The program dependence graph and its use in optimization

ORANALYST - Design

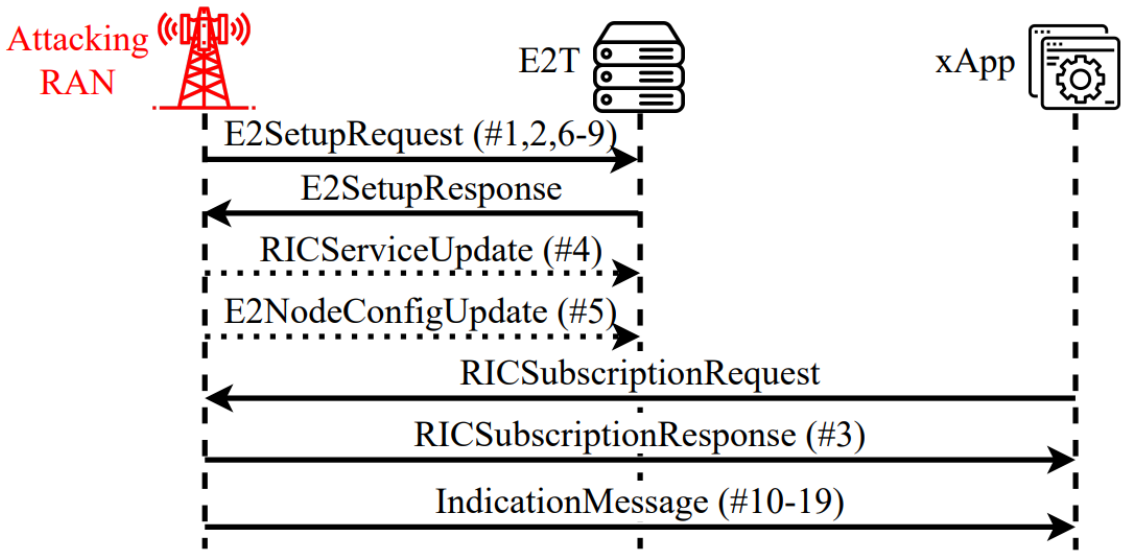
- **ORANalyst – Runtime analysis (C2 & C3)**
  - Generate input messages by mutating fields with ASN.1 grammar
  - Iteratively run feedback loops to calculate the code coverage



"Focuses on testing components at a time, shallow to deeper ones"

Evaluation

- **Setup**
  - 4 xApps and 6 platform components
  - 2 Open RAN RIC implementations
  - 24-hour period for each component
- **Results**
  - **19 issues across 7 components**
    - 17 led to crashes,
    - 2 led to the blockage of communication
  - Types of issues
    - Memory issues, improper error handling
    - All those vulnerabilities were able to crash and DoS the RIC and RAN



Evaluation

- **Comparison with fuzzing tools**
  - With adjustments to support Open RAN implementation

O-RAN-SC Component	E2T				Kpimon					
Fuzzer	crashes	corpus	cover	% decoded	crashes	corpus	bb cover	edge cover	% reaching xApp	% decoded
`ORANalyst`	3	2149	4326	72.35	3	73	1838	910	100/100	55.64
`ORANalyst` w/o input constraints	3	2149	4326	72.35	1	47	1828	907	47.27/59.01	53.50
`ORANalyst` w/o grammar	0	1433	4647	3.9	1	59	1831	906	40.64/80.81	16.76
AFLNET	0	245	3663	21.78	0	41	1824	901	32.81/97.83	12.37
BooFuzz	1	427033*	3655	81.96	1	427033*	1824	899	10.71/11.65	33.40
Radamsa	0	1323	3916	3.76	0	66	1827	901	11.39/78.20	4.40
Radamsa-filter	0	137	3467	100	1	35	1820	896	62.54/62.54	86.13

"ORANalyst without input constraints fail to effectively generate inputs"

Conclusion & Remarks

- **ORANalyst**
  - First end-to-end testing framework for Open RAN implementation
  - Utilizes static analysis and dynamic trace analysis
  - Was able to generate 19 vulnerabilities, which can lead to DoS and crashing RIC
- **Pros**
  - Dependency tracing and targeting specific components seems to be a good approach
  - Can be applicable to not only O-RAN testing, but other microservice architectures as well
- **Cons**
  - Honestly speaking, nothing seems new
    - C2: Implemented just ASN.1 protocols, C3: Capture process related system calls + logs [5]
  - One component at a time, not multiple
  - No consideration on “states”

[5] https://www.starlingx.io/blog/starlingx-oran-o2-application/

Related Works (Before)

- **LTE**
  - *[USENIX SEC'22] DoLTEst: In-depth Downlink Negative Testing Framework for LTE Devices*
  - [IEEE S&P'21] Bookworm Game: Automatic Discovery of LTE Vulnerabilities Through Documentation Analysis
  - [MobiCom'19] A Systematic Way to LTE Testing
  - [NDSS'18] LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE
- **5G**
  - [CCS'19] 5GReasoner: A Property-Directed Security and Privacy Analysis Framework for 5G Cellular Network Protocol
  - [IEEE Access'24] Formal-Guided Fuzz Testing: Targeting Security Assurance From Specification to Implementation for 5G and Beyond

Related Works (After)

- **LTE**
- **5G**
  - [USENIX SEC'24] Logic Gone Astray: A Security Analysis Framework for the Control Plane Protocols of 5G Basebands (Same authors)
  - [IEEE Access'24] Formal-Guided Fuzz Testing: Targeting Security Assurance From Specification to Implementation for 5G and Beyond
  - [IEEE WONS'24] AMFuzz: Black-box Fuzzing of 5G Core Networks
  - [WISEC'24] Security Testing The O-RAN Near-Real Time RIC & A1 Interface
  - [Arxiv 2024] CovFUZZ: Coverage-based fuzzer for 4G&5G protocols

Good Questions

- To solve the path explosion problem in static analysis, the authors selectively analyze some functions and ignore others. Can this lead to false negatives in their approach?
- How does ORANalyst ensure coverage for rarely occurring edge cases in real-world RAN interactions?
- The paper targeted RIC in O-RAN. Also, O-RAN uses a unified interface. What is the difference between O-RAN and other fuzzing papers?
- How is it that there is no standardized protocol? Is O-RAN a small field? What might be the reasons for the absence of a standardized protocol?
- What are the limitations in applying this methodology to proprietary O-RAN deployments instead of open-source ones?

Best Questions

- **Wonyoung Kim**
  - Unlike Traditional RAN, O-RAN allows eNBs to be configured in software, which I believe makes them more vulnerable to physical attacks. For example, a modern operating system can be used in O-RAN, which provides a high advantage to developers as well as attackers. This allows the attacker to conduct more malicious acts. If a base station is compromised, could vulnerabilities related to privilege management be more impactful than memory vulnerability attacks?
- **Younghyo Kang**
  - ORANalyst does not appear to include verification for ‘false-negatives.’ If this fuzzer were to incorporate a verification step comparing the output against a specification, similar to DoLTEst, it could become a more rigorous fuzzer. Do you think this would be feasible in practice?
- **Sihun Yang**
  - How does ORANalyst differentiate between critical vulnerabilities and those that might not be exploitable in real-world scenarios? Can ORANalyst evaluate the practicality of the found vulnerabilities?

Thank You