D3.1

Components for Trustworthy Disaggregated Computing Architecture

Funded by
the European Union

Revision v1.0

Work package	WP3
Task	T3.1, T3.2, T3.3, T3.4
Dissemination level	PU – Public, fully open. e.g., website
Deliverable type	R – Document, report (excluding periodic and final reports)
Due date	30-09-2023
Submission date	29-09-2023
Deliverable lead	Barkhausen Institut (BI)
Version	v1.0
Authors	Pavlos Alexias (WINGS), Kristoffer Andersson (EAB), Nils Asmussen (BI), Marco Bertuletti (ETH), Vanessa Daccache (NOK), Efstathios Katranaras (SEQ), Michael Roitzsch (BI), Ross Staton (NOK), Fredrik Tillman (EAB), Yichao Zhang (ETH), Enrico Guarino (TIM), Gian Michele Dell'Aera (TIM)
Contributors	Work package partners (see below)
Reviewers	Efstathios Katranaras (SEQ), Viktor Razilov (TUD)

Abstract

To describe the overall project architecture, we first review the target use cases the project has previously laid out in deliverable D2.1. The architecture we propose fundamentally improves the trustworthiness and efficiency of future Beyond-5G and 6G mobile networks. From this architecture, we identify components, where advancements beyond the state of the art are needed. These demands drive the technical development in work packages 4 and 5.

Keywords

architecture, heterogeneous accelerators, disaggregation, virtualization, trusted execution, remote attestation, signal processing, interconnects, radio fingerprinting

Document Revision History

Version	Date	Description of change	Contributor(s)
v0.1	24-07-2023	initial outline version	Nils Asmussen (BI), Michael Roitzsch (BI)
v0.2	15-09-2023	partner contributions to all sections	Pavlos Alexias (WINGS), Kristoffer Andersson (EAB), Marco Bertuletti (ETH), Vanessa Daccache (NOK), Efstathios Katranaras (SEQ), Michael Roitzsch (BI), Ross Staton (NOK), Fredrik Tillman (EAB), Yichao Zhang (ETH), Enrico Guarino (TIM), Gian Michele Dell'Aera (TIM), Nils Asmussen (BI)
v1.0		final version after review	Nils Asmussen (BI)

Contributing Partners

Abbreviation	Company name
BI	BARKHAUSEN INSTITUT
EAB	ERICSSON
CYB	CYBERUS TECHNOLOGY
EUR	EURECOM
SEQ	SEQUANS
TIM	TELECOM ITALIA
WINGS	WINGS ICT SOLUTIONS
NOK	NOKIA NETWORKS GERMANY
NNF	NOKIA NETWORKS FRANCE
IIIV	NNF/IIIV LABS

Disclaimer

The information, documentation, and figures available in this deliverable are provided by the COREnext project's consortium under EC grant agreement **101092598** and do not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

Copyright Notice

Executive Summary

The COREnext deliverable D2.1 explained three use case families targeted by the project. This deliverable translates these use cases into component challenges. We survey technology building blocks and synthesize an overall project architecture, from which we derive innovation needs in digital and analogue components. Key components of the architecture are:

- power-efficient signal processing,
- power-efficient high-throughput interconnects,
- radio link authentication and infrastructure attestation, and
- a heterogeneous compute platform with trusted execution environments.

These needs are then addressed in work packages four and five over the course of the project. Feedback from these work packages will influence subsequent iterations of the COREnext architecture described here.

Table of Contents

List of Figures

List of Tables

Acronyms and Definitions

AAL	Acceleration Abstraction Layer
AAL-LPU	Acceleration Abstraction Layer – Logical Process Unit
AALI	Acceleration Abstraction Layer Interface
AUSF	Authentication server function
BiCMOS	Bipolar complementary metal–oxide–semiconductor
DSP	Digital signal processor
EIR	Equipment identity registry
E2	Interface between Near-RT RIC and O-RAN nodes
FPGA	Field programmable gate array
Gbps	Gigabits per second
GHz	Gigahertz
GPU	Graphics processing unit
IoT	Internet of things
JCAS	Joint communication and sensing
MEC	Mobile edge cloud
MMIC	Monolithic microwave integrated circuit
Near-RT RIC	Near real-time RAN intelligent controller
NG Core	Next generation core
O-CU	Open centralized unit
O-CU-CP	O-RAN central unit – control plane
O-CU-UP	O-RAN central unit – user plane
O-DU	Open distributed unit
O-RAN	Open radio access network
O-RU	Open radio unit
OFH	Open Fronthaul
POLA	Principle of least authority
RF	Radio frequency
SEV	Secure encrypted virtualization
SGX	Software guard extensions
SiGe	Silicon–germanium
SoC	System-on-chip
TDX	Trust domain extensions
TEE	Trusted execution environment
TLS	Transport layer security
WP	Work package

1 Introduction

The COREnext projects sets out to **fundamentally improve trustworthiness and efficiency** of future Beyond-5G and 6G mobile networks. With 6G, we expect machine-to-machine communication to play an increasingly larger role. Extrapolating from 4G, which focusses on human-to-machine communication like the mobile web and video streaming, 5G already showcases machine-to-machine interaction in industrial environments. Factory automation and robotic control loops are being deployed over 5G. With an adequate infrastructure and communication performance, 6G can bring features like personal robots, autonomous driving, and interactive extended reality experiences to the mass market.

However, for mass-market acceptance, trustworthiness is a key requirement, which the upcoming 6G infrastructure must consider from design phase to deployment. This demand is especially imperative given features like **joint communication and sensing,** where the same physical antenna can be used for both mobile communication and radar-based sensing of the environment. Suddenly, every component in the network that is allowed to communicate can potentially subvert its antenna access to scan the physical space around the user. Without **foundational architectural measures** against such privacy disasters, some of the most interesting 6G applications may not be acceptable to a general audience.

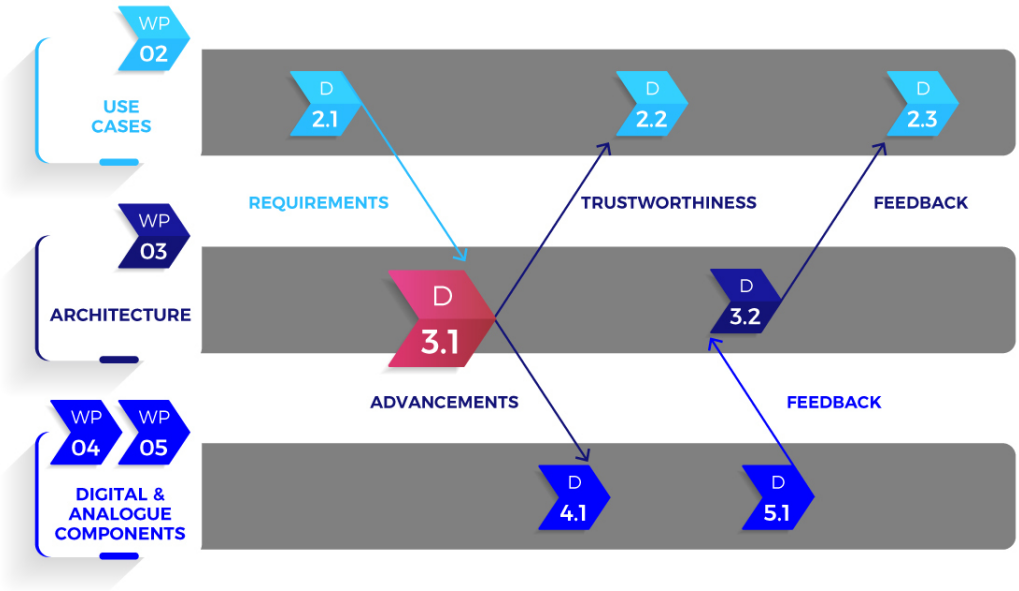

Figure 1: Information flow between deliverables

In this deliverable, we review the target use cases the project has previously laid out in deliverable D2.1. The overall flow (see Figure 1) translates the requirements from these use cases into **necessary component advancements.** Within this process, this deliverable formulates a project-wide architecture, which addresses the use case requirements. From this architecture, we identify clusters of components, where advancements beyond the state of the art are needed. This component demand drives the technical development in work packages 4 (for digital components) and 5 (for analogue components). Deliverable D4.1 will review this need and derive concrete

technical contributions by partners. Future deliverables will provide feedback from component level back to architecture and use cases, complementing the top-down with a bottom-up pass.

In the following sections, we first summarize the use cases and their requirements from deliverable D2.1 and sketch an attacker model for each use case (Section 2). Then, we survey existing technology building blocks to form a common terminology as well as understand gaps in the state of the art (Section 3). We propose an architecture for core network, base station, and terminal devices that addresses our use case requirements by postulating advancements in components (Section 4). We then contextualize our architecture against existing solutions and practical concerns (Section 5). Finally, we summarize the needed component advancements that further project work should address (Section 6).

2 Architectural Requirements from Use Cases

The list of use cases that will be analysed in COREnext is defined in the deliverable D2.1 that was focused in identifying three use case families encompassing a wide range of use cases: enhanced human communication and entertainment, enhanced machine communication, and intelligent management. Inside each family, a specific use case is selected for the COREnext requirement definition:

- The **extended reality (XR) use case** represents all the innovative use cases regarding novel human communication or entertainment. As the success of human interactions depends on the capability of reality reproduction, this use case requires performance in network capacity and latency.
- The **automotive infrastructure use case** represents use cases involving non-human communication that is required for human safety, exploiting the lower reaction time of non-human actuators.
- The **smart city use case** represents the use cases based on data management for optimization of capability/resource, exploiting a large amount of information provided by network elements (RAN nodes, edge nodes, sensor nodes).

The three use cases identified: XR, automotive infrastructure, and smart city will constitute reference use cases in the COREnext project. In deliverable D2.1, a table was created, listing for each use case the level of relevance for the requirements identified for the COREnext project. A copy of this table can be found below (Table 1).

Requirements	XR	Automotive Infrastructure	Smart City
Privacy, reliability, integrity	High	High	High
Trustworthy analogue access	High	High	High
Trustworthy distributed code execution	Medium	High	Medium
Energy-efficient connectivity	High	High	High
Energy-constrained devices	Low	Low	High
Ultra-low latency	High	Medium	Low
High-Capacity Connectivity	High	Low	Low

Table 1: Use case requirements and relevance

The requirements of **reliability, integrity, and privacy** are relevant for all the identified use cases. COREnext wants to guarantee a high level of security in all the proposed use cases,

introducing end-to-end trustworthiness from analogue radio to digital processing. Similarly, **energy efficiency** plays a crucial role in next generation networks, including the three proposed use-cases, to maintain a sustainable system: reducing environmental impact, saving costs for system owners, enabling energy-constrained use cases, and promoting a positive public image.

Each defined use case has its specific role in stressing the COREnext requirements:

- The **XR use-case** highlights the performance requirements. In this use case, data transmission can be summarized by a **latency** requirement and a **capacity** requirement. A latency requirement refers to the maximum tolerable delay in the reception of transmitted data that is acceptable for the desired user experience. Delays in XR applications can lead to a noticeable discrepancy between the user's physical movements or actions and the corresponding virtual response, resulting in a degraded experience and potentially causing motion sickness or discomfort. Therefore, low latency – resulting in a reduction of these delay effects – as well as a guaranteed minimal transmission capacity are crucial for delivering a compelling and immersive XR experience. Low latency and high capacity therefore avoid a reduction of the information quality in terms of image resolution sound fidelity or a loss of information for any other sense involved in the communication. The devices used in this use case are not always under the control of the network operator, because a domestic user is free to install or update any application they want. Consequently, the COREnext platform must ensure that only trusted applications can get access to sensing/communication capabilities of the modem. Similarly, the COREnext platform must securely host third-party applications, avoiding attacks and guaranteeing confidentiality and integrity of private information.
- The **automotive infrastructure use case** highlights the trustworthiness of non-human communication where applications running inside cars, road infrastructure, and maybe other vehicles (like bicycles) are relevant for driver's safety. This use case requires a distributed code execution architecture where trustworthiness is crucial to ensure security, reliability, and integrity of the decisions taken by all the non-human actors of the system. Critical aspects are controlling access to distributed components, controlling the integrity of code execution, monitoring and auditing distributed code execution, and keeping the distributed system up to date with security patches, bug fixes, and software updates. The improvement on trustworthiness has to be introduced without introducing additional latency that could sacrifice the "speed of acting".
- The **smart city use case** highlights the concurrent connectivity of a high number of sensors (IoT devices) where it is relevant to support an efficient data transmission technique that generates small packets of data sent with long idle periods in between to minimize the power consumption of the sensors. Despite the limited connection performance of the IoT devices, the platform must guarantee security for safety-critical and privacy-sensitive applications.

Given these focus requirements of the use case families, we can consider potential attacks relevant to these use cases. More detailed attacker models will be discussed in deliverable D2.2, which reconsiders these use cases from a trustworthiness angle, but a summary is given here to inform our architecture design:

- The XR devices will likely be based on **non-European processor and application platforms,** which may be considered untrustworthy. Apps running on those devices will have

access to a sensor-rich environment, so the key attack we consider is a **privacy compromise** due to apps combining location, camera, and other sensor data.

- With cars offloading code into the automotive infrastructure to benefit from rich sensor information, the same infrastructure will be used by multiple clients simultaneously. These clients will not necessarily trust each other and indeed, a **breach of software isolation and integrity** is a relevant attack. Such a breach can be conducted by a malicious client uploading attack software into the infrastructure or even by **gaining physical access to infrastructure components** deployed in the field. Consequences range from reduced service quality to cars executing unsafe driving manoeuvres.
- Smart city equipment is equally susceptible to physical attacks on devices. In addition, improving energy efficiency of these devices requires the integration of accelerators and other special-purpose hardware originating from third-party vendors. Each of these hardware components can contain **exploitable hardware vulnerabilities that enable denial of service attacks or violations of the confidentiality or integrity of other components in the system.** Given that a system-on-chip typically integrates several third-party hardware components, we observe a **large hardware attack surface** that increases the likelihood of successful attacks. Deploying such devices in critical infrastructure adds to the severity of the resulting problems.

3 Technology Building Blocks

The use cases and derived attacker models establish challenging requirements for the COREnext project, primarily grouped around the two main goals of efficiency and trustworthiness. Before we propose an architecture to address these challenges, we survey existing technology building blocks. This overview serves two purposes:

- We bring all readers to a common understanding of concepts and terminology.
- We review the state-of-the-art in different, project-relevant technology fields. We identify matches of existing solutions to the project challenges as well as boundaries and open problems of these solutions.

We group technology building blocks from radio interface to application software, moving from shared antennas and hardware for signal processing to software deployment and application isolation.

3.1 Joint Communication and Sensing

Wide bandwidths and large antenna arrays, which are the hallmark of typical high resolution radar systems, are also reminiscent of modern communication systems. Successive generations of communication systems have climbed in frequency, and we are now at a stage where many key radar bands for high resolution sensing are merging with communication bands. For example, some of the used radar bands like K (18 GHz-26.5GHz) and Ka (26.5 GHz – 40 GHz) are close to popular mmWave communication bands. Furthermore, the bandwidth of modern communication systems is large, thus enabling opportunities for Joint Communication and Sensing (JCAS). Radar technology and wireless telecommunications have coexisted for decades and most of the efforts so far have been on interference management to make the two technologies co-exist without disturbing one another. However, this has caused additional costs for infrastructure and inefficiencies in spectrum usage. The objective of JCAS is to share the spectrum more efficiently and **reuse the existing wireless network infrastructure for sensing,** i.e., providing sensing and localization capabilities as part of a wireless communication network service. In this context, sensing could refer to 'radar-like' functionality, i.e., the ability to detect the presence, movement, and other characteristics of objects under the coverage of the wireless network. However, it can also refer to other types of sensing, such as detection of general characteristics of the environment, local weather conditions, etc.

The main benefit of JCAS, compared to the deployment of separate networks is that the capability can be introduced at large scale with relatively low incremental cost by piggybacking on infrastructure deployed for a communication purpose. Massive communication infrastructure already exists, and it is foreseen that even more dense deployments will be available in the future which would allow further enhanced sensing capabilities. This opens the possibility not only for mono-static radar applications – where transmission of the radar signal and reception of the reflected signal are handled by the same node – but also for various multi-static setups where transmission and reception can be handled by different collaborating nodes. Also, if implemented properly, integration of sensing into communication networks has the benefit of better spectrum utilization compared to assigning separate spectrum chunks for the two applications.

However, sensing and communications are traditionally addressing completely different sets of use cases and requirements. In its simplest form, sensing uses a known signal that is sent in a particular direction and by analysing the reflected signal various parameters such as channel response, target-presence, and target properties like position, shape, size, velocity, etc., can be estimated with a certain level of accuracy and latency. In contrast, in communication, key performance indicators include data-rate, latency, coverage and reliability. This leads to sensing signal characteristics such as bandwidth, time duration, periodicity, power, etc., being different compared to a communication system.

Another aspect when combining communication and sensing relates to security, and the trustworthiness associated with **shared hardware for multiple systems.** Applications using the communication infrastructure must be prevented from having unauthorized access to information that is processed on the same platform. This will be equally important for terminals which carry even more sensor information, e.g., camera, positioning etc. Trusted solutions to **isolate sensing and communications** within a shared modem will be instrumental for continued digitalization of society. This cannot be solved at the application layer only but must be guaranteed by the hardware itself and related firmware, as applications are not trusted per default.

3.2 Heterogeneous Accelerators

Wireless communication like the one in mobile networks requires signals from the antenna to be processed before the payload of transmitted messages becomes usable. With data rates expected to reach 100 Gbps [1], processing the data requires a high computational throughput. High-performance general-purpose processors could be used to run software-implementations of these computations, but this would use a lot of electrical energy. Dedicated hardware solutions can solve the same task with a much lower energy investment, thus increasing overall energy efficiency.

The spectrum of available processing hardware can be categorized in terms of its flexibility and programmability on the one hand and energy efficiency on the other hand. **General-purpose processors** are fully programmable because they can run arbitrary software. Because a general-purpose processor invests a portion of its energy consumption to understand and manage the flow of software instructions, it is often the least efficient option for high-throughput computation.

Special-purpose hardware accelerators do not have this problem and consequently offer the most energy-efficient solution. While they are limited to a few or even a single task, they require little to no programming to fulfil it. The system around the accelerator organizes the incoming and outgoing data flows, but the accelerator itself autonomously performs its computation without interpreting software. While being superior with respect to energy-efficiency, the computation performed is baked into the hardware. A later change to the algorithms requires designing, manufacturing, and deploying new accelerators.

Flexible accelerators like graphics processing units (GPUs) offer a middle ground because they are programmable but are architecturally restricted to a specific problem domain. Similarly, **instruction set extensions** to general-purpose processors try to combine the advantages of efficient fixed-function computation in hardware with the flexibility of a programmable processor.

Because of their large-scale deployments, we believe mobile networks require energy-efficient hardware and they make heavy use of purpose-built hardware already today. However, the accelerators necessary for the signal processing in future networks are still to be researched and the trade-off between flexibility and efficiency is to be evaluated. Similar trade-offs exist in areas like image processing and artificial intelligence, where accelerators are industrial practice today.

3.3 Virtualization and Disaggregation

Virtualization is a technique that abstracts a physical machine to share it among multiple distrusting tenants. Each tenant receives a **virtual machine** that is indistinguishable from a physical machine for the tenant. Multiple virtual machines can run on the same physical machine and thereby share this machine among multiple tenants. Virtualization is commonly used in data centres to use the physical machines more efficiently and maximize their utilization. Since different tenants do typically not trust each other, isolation between different tenants is very important. The virtualization solution therefore needs to either assign different resources to different tenants (e.g., different memory regions) or multiplex the same resource in time among the tenants (e.g., the same CPU but during different times). Furthermore, tenants need to be prevented from accessing the resources of other tenants and therefore are running on top of a trusted software called **hypervisor**. The hypervisor isolates itself from all tenants and different tenants from each other and uses nowadays typically hardware-supported virtualization (e.g., Intel VT-x) to make that efficient.

In the recent past, data centres are not only providing storage devices and network capabilities to virtual machines, but increasingly also accelerators such as GPUs and FPGAs. However, classical data centres with a fixed set of such resources per machine have more and more trouble to adapt to the dynamic nature of applications and fully utilize the data centre's resources. For that reason, data centres are becoming increasingly **disaggregated.** The idea is to have separate pools of resources that are connected via a network fabric. Besides more modularity and therefore easier maintenance and replacement of components, disaggregation allows data centre operators to assign resources more flexibly to applications and thereby increase their overall utilization. For data centres in the mobile network area, the disaggregated resources will not only contain storage devices and accelerators, but also MMIO antennas. As all of these resources are then accessed over a network fabric, this fabric needs to be constructed using **high throughput and low latency interconnects** to fulfil the requirements of applications.

3.4 Container and Orchestration

Virtual machines are a way to securely share the same physical machine as described in the previous section. However, virtual machines are rather heavy weight, because each tenant has a full machine available requiring an operating system besides the desired application. Virtual machines have therefore a rather large memory footprint and long boot times. **Containers** are a more lightweight approach to share the same physical machine. Like with virtual machines, containers are isolated from each other, but unlike virtual machines, share the same operating-system kernel. For that reason, this type of virtualization is also called OS-level virtualization.

Containers therefore trade some of the isolation provided by virtual machines for less memory footprint and shorter start up times.

Containers are not only used due to their cost advantages, but also because of their easy **deployment.** That is, software can be packaged as a self-contained archive and therefore deployed in any environment. Containers therefore avoid the problems that come with software packages that rely on other software being installed in a particular version and configuration on the target system. Besides the deployment, running several containers requires **orchestration** to manage their life cycle and assign/re-assign them to different resources during this life cycle. In the context of O-RAN (see 5.1), these resources also include the interaction with communication networks.

Like for virtual machines, **strong isolation** between containers is important as different containers typically belong to different mutually distrusting tenants. In the context of O-RAN with access to communication networks, isolation is even more important. For that reason, the shared underlying hardware and software platform should be as small and reliable as possible.

3.5 Component Isolation and Access Control

Software running on one hardware platform can originate from different security domains: O-RAN or edge cloud applications should be isolated from each other and from the software of the infrastructure provider. An important means to improve trustworthiness is to enforce **strict isolation** between such software components. Whenever a component is faulty or has been taken over by an attacker due to a vulnerability, strict isolation limits the blast radius of such a fault.

However, software cannot function in perfect isolation because it needs to interact with other components and its environment. Therefore, strict isolation must be accompanied by controlled communication. To again improve trustworthiness, we should follow the security design pattern called the **Principle of Least Authority (POLA),** whereby only necessary accesses are allowed and everything else is denied by default. In contrast to an allow-by-default approach, this is also called a **deny-by-default** approach.

POLA is currently employed primarily in the software world. High-security microkernels like L4 enable strict isolation between software components [2]. For COREnext however, we want to extend this protection to the hardware-level. We expect hardware platforms to be composed of a mix of general-purpose processors and accelerators, which we do not necessarily all trust equally. Thus, it makes sense to apply similar strict isolation and communication control principles to hardware components as is already done for software.

As an operating system abstraction, so-called **capabilities** are a good fit to implement a POLA system. A capability is an unforgeable access token that grants the right to use a communication channel (or another resource). Proof of possession is needed before any communication is allowed. Thus, capabilities are a natural mechanism for a deny-by-default system. Capability-based access control for all hardware and software interactions would constitute a good starting point for a **secure-by-design** system. However, hardware-enforced capabilities are only nascent research concepts in novel processor architectures like CHERI [3] and in M³ [4, 5] for systems-on-chip.

To enforce a communication policy system-wide, all communication channels must be supervised such that only allowed communication takes place and any other attempts are prevented. The mechanisms to supervise communication channels and manage communication policy must be sufficiently scalable and lightweight. The hardware and software functionalities for isolation and communication control form a set of platform features that all applications on that platform must trust. In this sense, these mechanisms form the **trust anchor** of the system.

3.6 Trusted Execution and Attestation

In a distributed system, we rely on remotely running software to implement an end-to-end use case. A good example is the automotive infrastructure use case, where code running in the infrastructure is potentially trusted by the participating cars. Trust in such distributed code execution can be established by means of **remote attestation.** The hardware on the remote server can produce a cryptographically verifiable proof of the software running on the processor. Thus, the client software – for example in the car – can verify that the expected version of the remote software is being executed.

The process of remote attestation augments regular encrypted and authenticated communication over protocols such as Transport Layer Security (TLS). TLS only assures to the client that the server is in possession of a private key, but this private key could have been stolen and the client is in fact talking to an imposter. With remote attestation, the client receives proof of the software state and an unforgeable identification of the underlying hardware. This much stronger evidence allows the client to assess the integrity of the remotely running software and thus substantiates the client's trust in it.

As an extension of remote attestation, a **Trusted Execution Environment (TEE)** is a container for code execution that is shielded from the surrounding system such that as little as possible outside influence on code execution is allowed. TEEs rely on processor extensions to even remove access by the operating systems into the TEE. Management and deployment services that are part of the system can only treat the TEE as a black box as they cannot observe the computation inside it.

While remote attestation alone provides only a snapshot of the system state, with server-side software running inside a TEE, the client has a guarantee from the hardware of the **long-term integrity** of the software state. Currently, TEEs functionality is offered by commodity processor vendors like Intel (SGX, TDX) or AMD (SEV), but TEEs are limited to the main processor of the system. It is an open research question how to securely integrate accelerators and efficient connectivity into TEEs.

4 Proposed Architecture

We propose an architecture for COREnext that covers an end-to-end technology stack from the device in the user's control up to edge cloud processing of user data. In this way, we look at all software, hardware, and connectivity layers that potentially come in contact with personal data of human users or safety-critical data of IoT equipment.

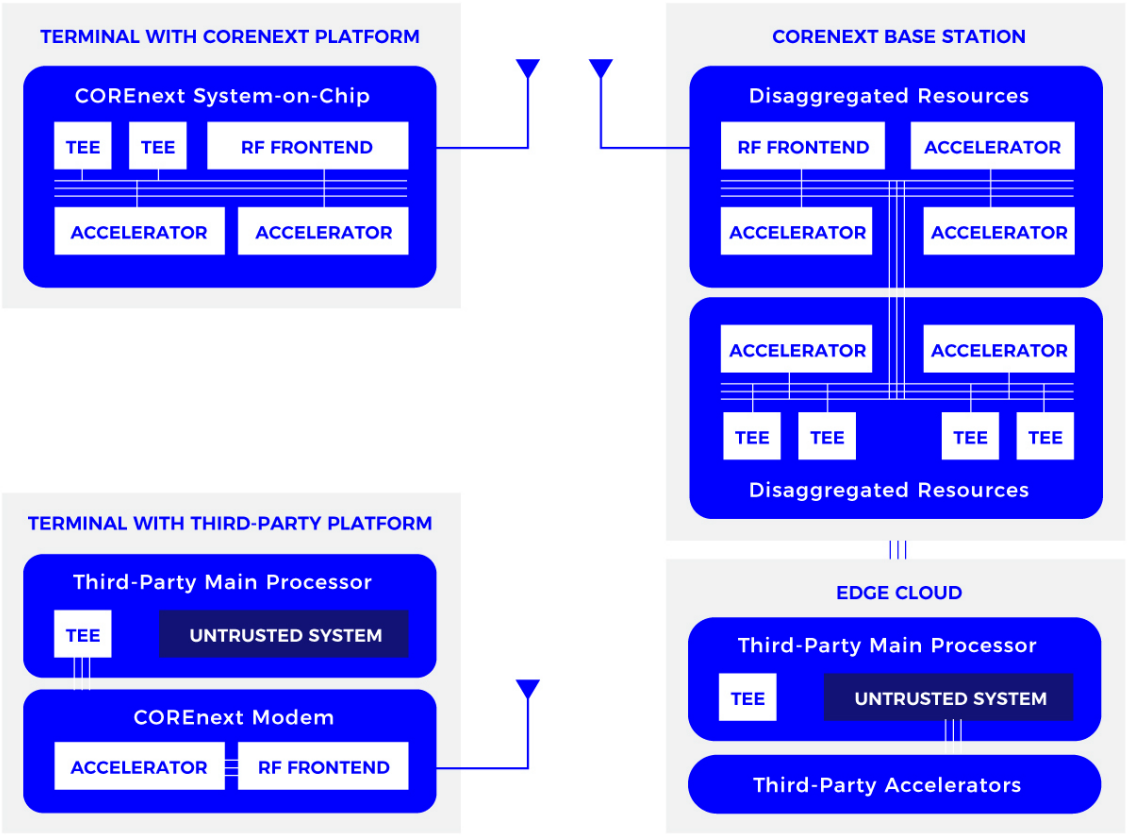

Figure 2: COREnext architecture overview

Figure 2 summarizes the COREnext architecture. On the left, we see different kinds of terminal devices, which can range from end-user devices such as phones or XR glasses to IoT devices in industrial or personal contexts. Terminals include safety-critical devices like connected cars or sensors and controllers in critical infrastructure of a smart city. The right part of the figure shows the network infrastructure, which is comprised of base stations and edge cloud nodes. The separation between the two is variable as some workloads of edge data centres can be moved to base stations (for lower latency or to reduce hardware deployment) or network management tasks can be offloaded from base stations to the cloud. Base stations and edge cloud nodes can also vary dramatically in size and scale, depending on their deployment location and performance needs. But architecturally, they contain the same elements, irrespective of their concrete configuration. The edge cloud is connected to more centralized tiers of cloud infrastructure and ultimately to the internet, which is not shown in the figure.

The following subsections explain terminal and base station architecture in more detail. But we want to highlight some aspects important for the overall understanding here:

- The terminal side of the figure is split into two versions: The upper variant represents devices, where the terminal is designed entirely based on a COREnext application platform, offering superior trustworthiness. The lower variant recognizes that we cannot expect all terminals to follow this path. Instead, we discuss how a COREnext radio system can interface with a third-party application platform, which is based on a third-party processor and operating system.
- Similarly, the right half of the figure is split into an upper part, showing a base station using the COREnext architecture, whereas we expect the edge cloud in the lower part to be using off-the-shelf hardware components.
- Even when employing non-COREnext hardware, the architecture embedding these components should ensure the overall trustworthiness of the system.

The architecture proposal combines many of the building blocks described above: trusted execution environments, accelerators for signal processing, and resource virtualization and disaggregation. More details and open research questions are discussed in the following.

4.1 Terminal with COREnext Application Platform

Focusing on novel machine interactions like in the smart city and automotive infrastructure use cases, we posit that terminals serving as edge devices should be focused on three basic axes:

- **Security:**
  identity assertion, authentication, secure communication
- **Energy efficiency:**
  energy harvesting, computational offloading, duty cycling and power saving
- **Flexibility:**
  multi-connectivity, adaptive routing, support for a wide spectrum of applications, relay operation

In **contemporary terminal architectures,** the device is generally made of several blocks and functions. The user-related functions (such as making a phone call, connecting a data service, handling several interfaces, etc.) are driven by the "host CPU", usually a powerful microprocessor with complete control of almost everything in the device. In addition, the device includes one or multiple modems for wireless connectivity (cellular, Wi-Fi, etc.) which can be a separated component or partly embedded into the host CPU for simpler systems. The cellular modem is made of two main parts, a baseband, and an RF component, but also includes other components such as memories and a power management processor for managing the overall communication process. Main blocks and functions of the modem include the digital signal processor (DSP) for encoding and decoding wireless signals (with channel coding, modulation/demodulation, error correction functions, etc.), the protocol stack (with protocols such as MAC, RRC, PDCP, etc.) for managing the communication with the network, and the RF transceiver for converting the signal from analogue to digital so as it can be processed by the DSP and vice versa. The modem is then connected to the RF frontend, which ensures (with functions such as filtering, amplification, etc.) the adaptation of the RF signal to the proper frequency band, and finally connected to the antenna

subsystem. A network-on-chip, usually hierarchically designed, is used to interconnect the various blocks within the terminal platform.

In COREnext, this general terminal architecture shall be enhanced to emphasize the 3 basic axes of focus.

Regarding the **security** aspect of the devices, terminals should be able to provide confidential, authenticated, and efficient communication with the edge cloud, with protocols such as MQTT, CoAP, HTTP, as well as cryptographic protocol support, through TLS. Moreover, the terminal architecture should ensure the integrity of its local operation against attacks by **isolating software and hardware components** from each other to reduce the impact of vulnerabilities and failures. A COREnext-designed application platform and modem isolates hardware components that can originate from multiple vendors into separate tiles which communicate securely with each other and the external world.

Taking into consideration that the terminals could also refer to devices with high energy demands, energy harvesting capabilities (solar panels, RF harvesters, etc.) combined with computational offloading would be crucial for the performance of the device. **Energy efficiency** can also be improved by including **task-specific accelerators** within a system-on-chip. These traits are also crucial for supporting several years of lifetime for battery-based ultra-low power and complexity devices or even battery-less IoT devices with limited or no energy storage capability that can employ a very simple RF transceiver architecture for either active, semi-passive, or complete passive signal generation [6]. Such devices may also need a redesign of reliable and energy efficient security mechanisms compared to the standard authentication, encryption, data integrity, and authorization protocols. In addition, energy efficient terminals should consider advanced power saving mechanisms such as adaptive and efficient duty cycling and circuitry sleeping, or triggered wake-up that may involve the use of a separate low-power monitoring receiver [7].

Finally, on the **flexible network** aspect, multi-connectivity capabilities and adaptive routing based on dynamic configurations (e.g., varying network conditions, data rate, etc.) could drastically increase the efficiency of the device. In addition, the capability of one device to serve as both a terminal and a base station, allowing interconnection between devices, can provide benefits to the overall network such as extended coverage, improved reliability, load balancing and reduced infrastructure costs. Such a relay operation capability will require concurrent execution of both terminal and base station functions, while ensuring that data forwarding is done securely.

On all three axes mentioned, managing, and exposing device capabilities in a secure manner is crucial to adjust to the specific use case and ensure the continuous trustworthiness of the device.

4.2 Terminal with Third-Party Application Platform

In practical reality, not all terminals will contain a complete COREnext platform with the secure and trusted features we propose. Complex edge and end-user devices that require high computational capabilities may use **non-European processors and application platforms,** which will have access to sensitive data as well as communication and sensing functionality offered by the device. Furthermore, it is crucial to take into consideration that some edge devices could contain

untrusted third-party components. With the term untrusted, we denote hardware components that are not guaranteed to provide data security and privacy compliance.

In order to establish trust across multiple levels – hardware, software, network –, **isolation and compartmentalization** are fundamental design principles towards ensuring that even if one component is compromised, the overall integrity and trustworthiness of the system is maintained. Nevertheless, isolation and compartmentalization still require the existence of secure and trusted communication channels among components.

Communication channels can be grouped based on the nature of the components that communicate with each other. This grouping helps to identify risks and problems, as well as proposing possible solutions based on the nature of the components.

On the **hardware-level,** we can rely on the trustworthiness of the COREnext modem, because it contains the security features we design as part of the project. Those include hardware fingerprinting using AI and strong isolation between components. The modem itself must also be secured from malicious accesses by the untrusted third-party processor platform, for example by shielding unwanted memory accesses and by offering only a narrow and well-defined interface.

Applications running on the third-party platform may want to use the modem to **communicate with cloud and edge services.** This type of communication passes through the COREnext modem protected by cryptographic protocols like TLS. Higher-level protocols for specific use cases such as HTTPS, SSH, or MQTT also apply here. In these cases, the COREnext modem simply carries those encrypted messages as payload and is not further concerned with their content.

However, applications on the third-party platform may also interact with the COREnext modem to access value-added services such as Joint Communication and Sensing (JCAS). In this case, instead of a cloud service, the **modem itself acts as a service provider** to the application. Given the privacy implications of technologies like JCAS, we do not want to offer these functionalities to arbitrary applications running on the untrustworthy third-party platform. Asking for application identities is also insufficient because the third-party platform could lie about them to gain access itself.

Therefore, we rely on the existence of TEEs as isolated environments provided by the third-party processor. These offer **islands of trust** within the untrustworthy third-party software. The modem can use attestation technology to check the authenticity of a TEEs code and only offer JCAS services to vetted applications with regulatory approval.

4.3 COREnext Base Station and Edge Cloud

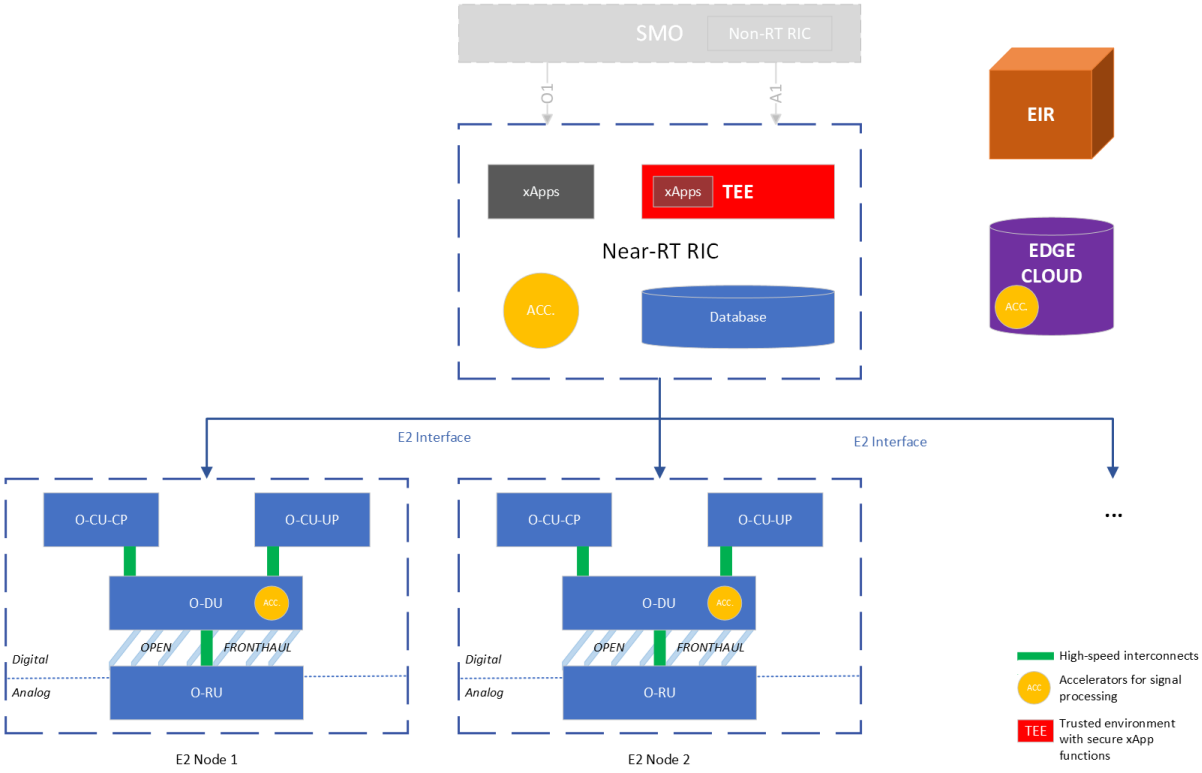

Figure 3: COREnext base station architecture

The architecture design of a COREnext base station leverages the architecture outlined in the O-RAN specifications [8]. The basic principles applied to classic RAN architectures can also be applied here. In the architecture we are proposing, we establish the placement of the previously described blocks of TEE and accelerators according to their needs. Figure 3 visually represents all the components of a base station architecture. We will further relate COREnext to O-RAN in 5.1.

A key component of the broader COREnext base station is the Near-RT RIC (Near Real-Time RAN Intelligent Controller). In current 5G next generation core solutions, the Near-RT RIC is responsible for making near-real-time decisions to optimize the performance of the radio access network, improve user experience, and efficiently allocate resources. In a COREnext architecture, the controller can be extended to include a security module, known as TEE, and accelerators for signal processing, resource virtualization and disaggregation. The TEE is responsible to verify the trustworthiness of the xApps running inside the RIC and to prevent malicious xApps from accessing sensitive data or impacting the RAN performance. Furthermore, TEE can be enhanced to authenticate the hardware inside the E2 nodes and to trigger the E2 nodes to perform authentication of UE terminals that are requesting access to the RAN. The E2 node digital hardware authentication will be further elaborated in WP4 D4.3.

UE Terminals requesting access to a COREnext base station will be authenticated via an RF fingerprint scan as described in WP5 D5.1 and D5.2. The Equipment Identity Register (EIR), responsible for equipment verification and user authentication in current NG core solutions, can be

extended to store more hardware relevant information such as those RF analogue fingerprint scans. The process unfolds as follows:

A security xApp running within the TEE will trigger an RF fingerprint scan for the specified UE terminal via an E2 interface command to the O-DU. It will also check the local database for existing RF fingerprints for this device. If no local scan is available, then a request will be sent to the EIR database to download any existing scans for comparison. If the database of the near-RT RIC is loaded due to a high number of users in the RAN, then the RF scans can be stored and processed on the mobile edge cloud server. The newly obtained RF scan must be compared to the original RF scan for this hardware unit by means of signal processing resources. The latter can be either performed on the O-DU using FPGA accelerators, or offloaded to the mobile edge cloud server if the O-DU is loaded due to a high number of users. If the authentication is unsuccessful then access to xApp services within the TEE is denied for this UE terminal.

The open fronthaul connection between the O-RU and the O-DU is realised by a high-speed RF link. Using such wireless connections will increase the security of the current open fronthaul interface which is not encrypted in current specifications [9]. Details pertaining to high-speed interconnects will be covered in WP5 D5.3.

In section 255.1, we discuss further the existing O-RAN standards and relate it to our architecture.

4.4 Key Innovations Required by the Architecture

With this overview of our proposed COREnext architecture, we can summarize the necessary innovations beyond the state of the art. Figure 4 repeats the architecture figure from the start of this section with these innovations highlighted.

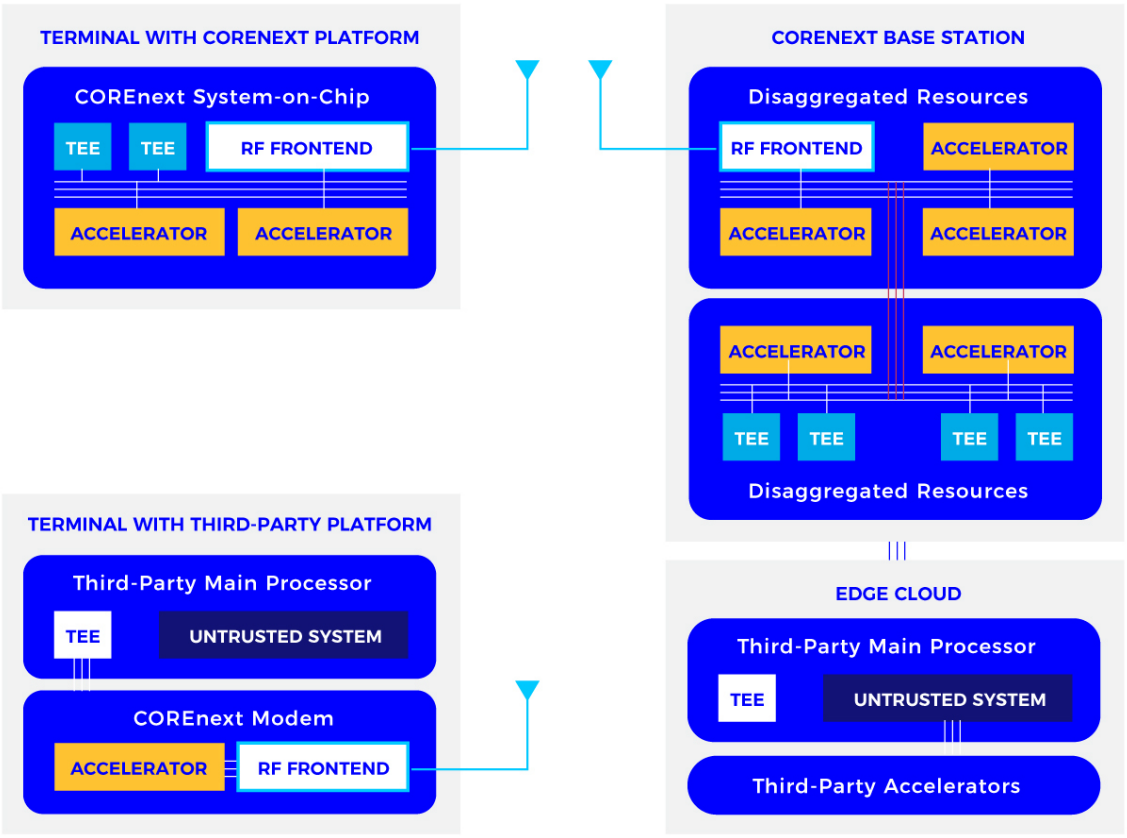

Figure 4: Required innovations to realize the CORENext architecture

- The CORENext and third-party terminals require **novel signal processing accelerators** to achieve the desired energy efficiency. Similar accelerators can be used to improve the efficiency of signal processing on the base station side (marked yellow in the figure).
- To organize hardware resources and their assignment to workloads flexibly, the base station employs a disaggregated design, which groups resources into pools that can be flexibly combined via a fast interconnect. Efficiency, latency, and throughput goals for base stations require a **novel high-throughput interconnect** (marked red in the figure).
- For terminals to offload computation into the cloud and for base stations to reliably identify clients, **radio link authentication and infrastructure attestation** is needed. On the lowest layer, a mutually verified analogue device fingerprint (marked blue in the figure) can provide a novel element to support the trustworthiness.
- Within terminal and base station hardware, computation must be organized in a securely isolated and coordinated fashion. This property must be provable by means of attestation across processors and accelerators, calling for **novel heterogeneous trusted execution environments** (marked cyan in the figure).

5 Solution Scope and Related Work

With a proposal for an end-to-end project architecture in place, this section looks at O-RAN as an existing standard and relates it to our architecture. We show the boundaries of the COREnext project and identify hand-off points for future research efforts. Finally, we discuss concerns for the practical applicability of our research.

5.1 Existing Standards: O-RAN

The O-RAN alliance is defining a new reference architecture for Radio Access Network disaggregation in mobile communication, proposing an open and flexible framework. This architecture is based on the architecture defined in 3GPP [10] that introduces additional open and well-defined standard interfaces increasing the number of disaggregated components. The main result of such architecture is easier interoperability when using network elements from different vendors.

Figure 5 below shows the key components in O-RAN architecture [11], made of software or hardware elements, and interfaces connecting them. Figure 6 below shows how each component is mapped in Control Plane (C-Plane) and User Plane (U-Plane) RAN Protocol Stack. In detail, starting from the lower level of the protocol stack to the higher level, the O-RAN building blocks are:

- O-RU (O-RAN Radio Unit) contains the radio frequencies and the lower portion of the physical layer detailed in [12]. This is the only component that cannot be virtualized.
- O-DU (O-RAN Distributed Unit) is a virtualizable component that oversees Higher part of the PHY (e.g., Channel Coding), MAC Layer and RLC Layer and radio signal processing, like coding, modulation, and demodulation. The O-DU can exploit the OFH interface standardized in [12] to send data to one or more O-RU and it is connected to a single CU using the interface F1 [13] standardized by 3GPP.
- O-CU (O-RAN Central Unit) is a network node where the RAN protocol stack is split in Control Plane (O-CU-CP) and User Plane (O-CU-UP).
  - The O-CU-UP implements the following protocol layers: Packet Data Convergence Protocol (PDCP) and Radio Resource Control (RRC) handling Radio configuration and information messages between User and Network.
  - The O-CU-UP implements the protocol layers: PDCP and SDAP (Service Data Adaptation Layer) [14] with the role to map the QoS (Quality of Service) Flows of the PDU Session to the DRB (Data Radio Bearer) handled by the Radio Access Network.
- Near-RT RIC (Near-Real-Time RAN Intelligent Controller) is a logical function that can control the Radio Access Network. It implements procedures of Radio Resource Management (like handover management, radio resource management, Service Cell/Node addition ...).
- Non-RT RIC (Non-Real-Time RAN Intelligent Controller) is a logical function inside the SMO (Service Manager Orchestrator) used to orchestrate the RAN network. It configures policies and network functions and receives Fault and Performance Information. In general SMO and Non-RT RIC are responsible for Network FCAPS (Fault, Configuration, Accounting, Performance and Security).

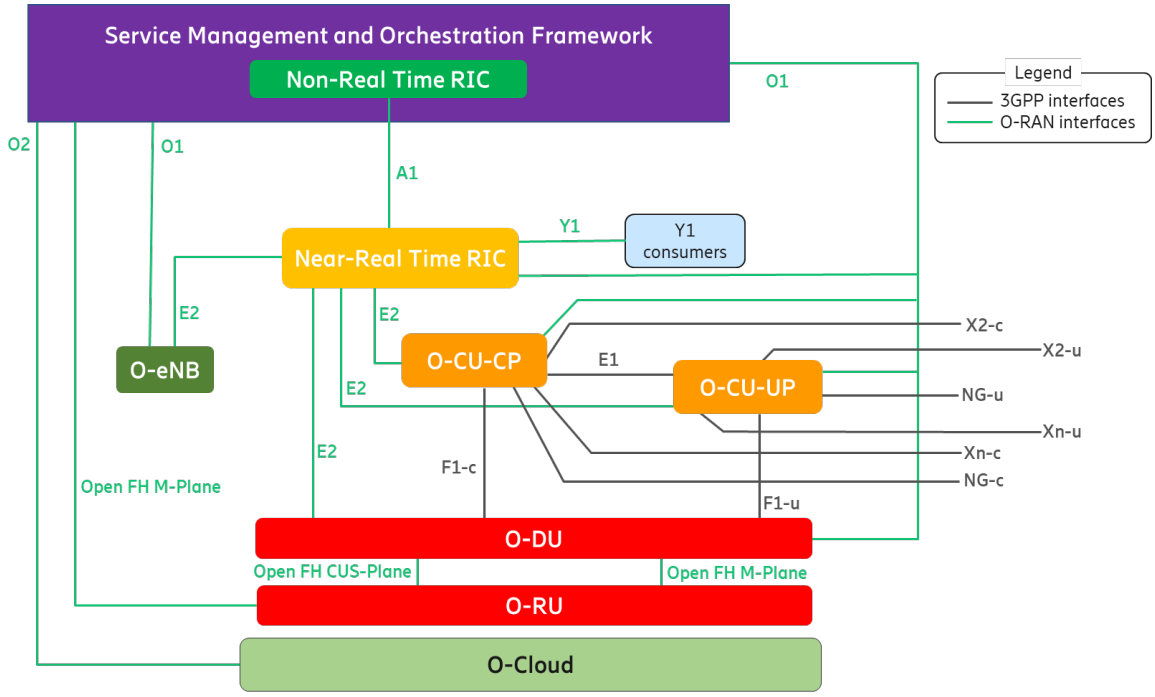

Figure 5: Logical Architecture of O-RAN

O-RAN Alliance defines the interfaces: O1, A1, E2 and O2 showed in Figure 5. The O1 interface is related to configuration and management of each component. The A1 interface oversees the coordination of policies by Non-RT RIC, as well as the transfer of machine learning models and data running on Near RT RIC. The E2 interface enables the Near-RT RIC to directly control the RAN elements (O-DU, O-CU-CP, O-CU-UP). The O2 interface is used to deploy the elements in a virtual environment (defined O-Cloud in O-RAN).

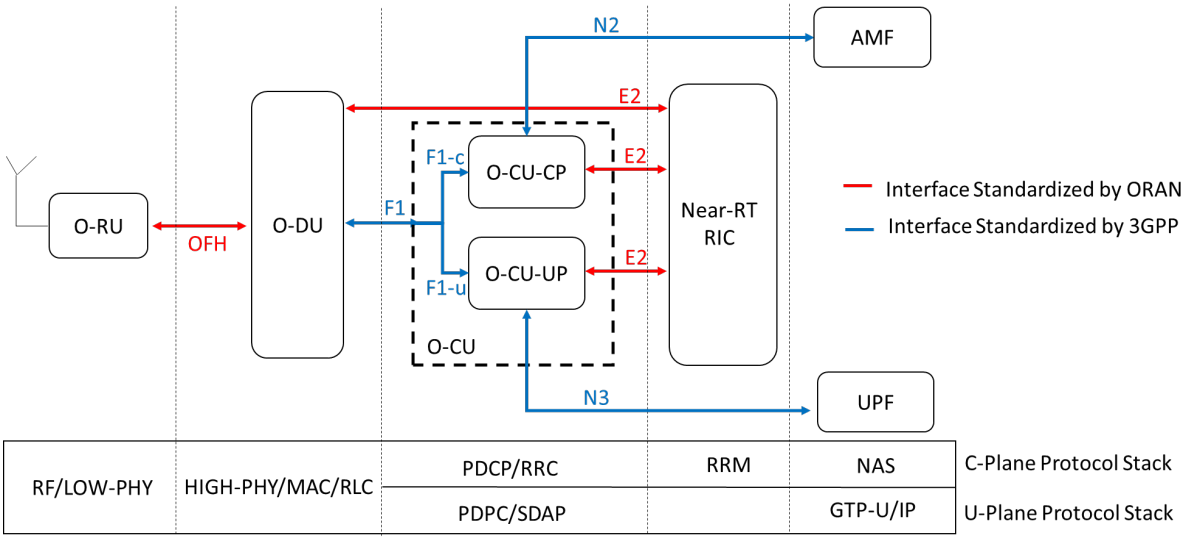

Figure 6: RAN Protocol stack mapping on O-RAN

In a world where privacy and security requirements need to be met, this type of distributed and disaggregated architecture requires each component to be as secure as possible since there are several points in which the platform or the RAN itself could be attacked by a malicious user. Each distributed element can benefit from using the proposed trustworthy architecture, such that every component can trust the other, and to assure that the communication channel is not manipulated or liable of unintended usage.

Hardware acceleration is an important aspect in radio access network, and so O-RAN has defined an architecture providing a layer to interact with specialized hardware accelerators, called AAL (Acceleration Abstraction Layer) [15]. This layer is a component of O2 interface, shown in Figure 5 that can accelerate portions of the virtualized RAN applications (DU, CU, RIC etc.). A clear example is the DU, where the FEC (Forward Error Correction) block is an application workload used for channel coding/decoding that can be offloaded in hardware. In Figure 7, a diagram is depicted showing the applications that can be accelerated exploiting the AAL interfaces. These interfaces are means to discover and configure the AAL-LPU (Logical Process Unit) that are then used to offload the computation to physical hardware accelerator devices.

In O-RAN, the RAN is made of distributed network elements and disaggregated software functions that can be executed in this environment. The DU can mainly benefit from this architecture, being a virtualizable network element with several functions that can be accelerated (signal processing, analogue-to-digital conversion etc.).

Security in AAL architecture is of critical importance to ensure integrity, confidentiality, and availability of data and functions hosted on top of it. COREnext architecture can help harden the API, interfaces, and hardware elements, for instance introducing the TEE in the requiring modules.

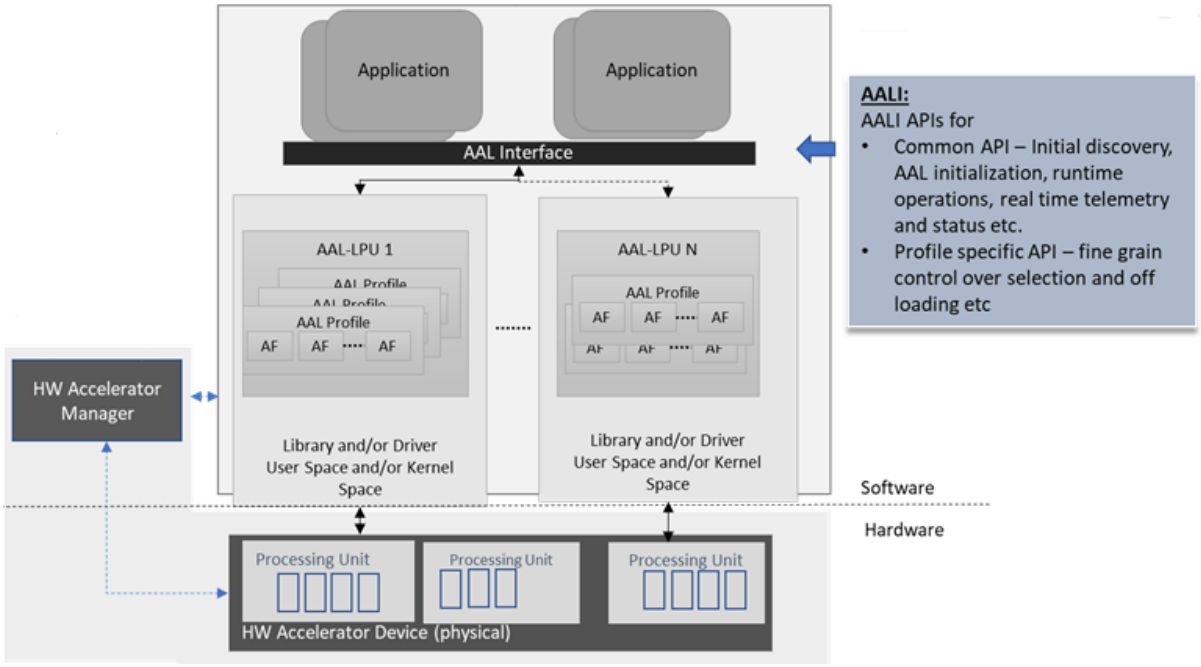

Figure 7: O-RAN Acceleration Abstraction Layer architecture

5.2 Boundaries of COREnext Contribution

COREnext is addressing fundamental challenges and research questions associated to the implantation of 6G trust functions and hardware-enabled security. As the maturity is low and the work experimental in nature, the validation of results will primarily take place using lab facilities and infrastructure among project partners. This is fully in line with the COREnext objectives and paves the way towards further refinements and future potential commercialization activities beyond the project duration. Investments in more realistic and advanced setups will be needed, as well as understanding implications of real-time processing needs and impact of constrained hardware resources.

Low-level hardware development on the actual processing cores, e.g., RISC-V cores, will be necessary to complement the initial activities using standard cores. System software needs to evolve along with the new hardware processing and trustworthiness features developed by COREnext. We expect a need for further research on how to interact with off-the-shelf software stacks like Linux in a secure way, while further expanding custom system software layers to bring the trustworthiness, energy efficiency, and low-latency operation of the COREnext hardware to applications.

In addition, to ensure compliance and compatibility with network standards (e.g., 3GPP) and commercial reliability for end-users, further efforts will be required that go beyond the horizon of COREnext. The level of testing needed to validate such innovations against requirements set by regulators would be an overwhelming stretch compared to the project objectives and ambition.

Finally, life cycle management of equipment (e.g., in an O-RAN context) is an important area for field deployment that will require open and standardized interfaces. This leads to a decoupling of software and hardware network elements, and the need for additional understanding of such interplay – also currently not in scope of COREnext.

5.3 Scalability and Practicality

Deployments of mobile network infrastructure like the one proposed by COREnext in the real world must address concerns of practicality. The developed solution must be **scalable,** which can be difficult to test in research environments. Also, it must be **cost-efficient,** which means that functional and non-functional properties including security and trustworthiness must be weighed against their cost in the market in terms of material, components, and energy.

Recognizing these concerns, the COREnext project runs a dedicated work package (WP7) to compile a **roadmap for industrial adoption** of the technical solutions developed in the project. Within the scope of this work package and its deliverables, we will investigate such practicality concerns.

However, from a technical and architectural standpoint, two arguments can be made:

- Scalability and security have an overlap: Low complexity and rigorously structured systems tend to improve both qualities. The strong isolation we position in our architecture to improve trustworthiness by strengthening its security can thus also keep system complexity in check by way of the **divide-and-conquer** principle.

- The cost of trustworthiness can be reduced, when considering it integrated architecturally from the beginning. When applied as an afterthought, we often must rely on encryption to tunnel sensitive information through untrusted connections or to protect it in untrusted storage. However, with a principled approach to **separating security domains and strictly enforcing communication control,** costly encryption and authentication can be reduced, because sensitive information flows can be routed in a way where they simply do not touch untrusted components.

We intend to demonstrate in COREnext that a trustworthy mobile network infrastructure does not automatically become complex and overhead heavy, but that in fact, the opposite can be achieved. By implementing communication control in hardware, we want to show a lightweight substrate into which accelerators and processors can be plugged to run signal processing, network functions, and O-RAN applications.

6 Components to Develop

After relating our architecture to existing standards and practical concerns, we now lower the proposed architecture into needed component advancements. These advancements were already sketched in Section 4.4 and are now described in more detail. We intend to identify clusters of closely interacting components that require innovation beyond the state of the art. These component advancements then become input for the technical research work in work packages 4 (digital) and 5 (analogue).

Reviewing the architecture, COREnext aims to contribute:

- a trusted base station infrastructure, and
- a trusted terminal infrastructure with either COREnext or third-party application platform

using

- efficient signal processing and acceleration,
- efficient interconnects,
- infrastructure authentication and attestation, and
- trusted digital computation.

	Digital	Analogue
Efficiency	Power-efficient signal processing	Power-efficient high-throughput interconnect
Trustworthiness	Heterogeneous compute platform with TEEs	Radio link authentication and infrastructure attestation

Table 2: COREnext component advancements

Table 2 illustrates, how these contributions can be clustered. We can group component advancements along the efficiency/trustworthiness and along the digital/analogue dimensions. Innovation in each of the four quadrants is required, which we will explain in the following subsections, with more details to be expected in later deliverables from work packages 4 and 5.

6.1 Power-Efficient Signal Processing

In the past few years, the data rates and computational complexity required for wireless communication systems escalated. For example, 5G promises peak data rates of up to 20 Gbps, while 6G is expected to deliver data rates of 100 Gbps or higher [1]. 5G algorithms are indeed moving fast towards 5.5G (2025) and 6G (2030) with over 10x decrease in end-to-end latency required [16]. To meet the objectives of these new generations of cellular networks, there is an increasing demand for novel and enhanced features within the backbone infrastructures.

To sustain these workloads resorting to off-the-shelf computational commodities, such as general-purpose processors, would not give the best performance in terms of energy efficiency, latency, and throughput. COREnext targets the implementation of **application-specific accelerators** to map in hardware the most demanding functions of a 5G/6G base station. In this regard, the project will move in two directions. On one side it will propose designs of specialized

data paths for the most repetitive signal-processing tasks in the workload, having input and output data as a continuous stream over time. The project will also focus on the use of available AI image/video processing accelerators, as Deep Learning models are considered promising candidates to substitute and simplify some of the most demanding parts of the 5G processing chain [17]. On the other side, COREnext will consider more flexible, adaptive, and reconfigurable solutions, where fully programmable RISC-V cores are clustered in hundreds or thousands to exploit large-scale parallelism in the execution of the processing steps [18], or the RVV extension set is adopted to boost performance via data-level parallelism [19]. Thanks to RISC-V extensible instruction set architecture (ISA), the processors could eventually be enhanced with ISA extensions tailored to the 5G domain.

We therefore target a **heterogeneous platform,** where different types of processors and accelerators, each optimized for a specific type of task, are packed within a single system. This is a viable compromise between the flexibility of software implementations and the efficiency of RTL-based ASIC designs. Further details about the accelerators and platform development will be presented in deliverable D4.1.

6.2 Power-Efficient High-Throughput Interconnect

The demand for greater I/O bandwidth within telecom systems and data centres is increasing exponentially caused by the explosive growth of network traffic. However, conventional electrical and optical high-speed interconnects are struggling with the challenges in functional and economical aspects, energy efficiency, and trustworthiness.

Copper-based electrical links are bandwidth limited because of skin losses and require complex circuits for equalization and coding. In optical links, for short distances, the complexity and cost for the electronic-to-optical and optical-to-electronic conversion devices and the chip-to-fibre assembly is an issue. As an alternative, we believe that **terahertz-waves over plastic fibre** will be a much better solution for telecom system and data centre applications. The latest reported achievements in terms of data rate and distance are clearly indicating that terahertz-over-plastic has the potential to replace the existing Cu-based and optical technologies used in high-throughput data interconnects.

In COREnext, a terahertz-over-plastic fibre demonstration will be implemented at 200-280 GHz, targeting for 224 Gbps over a distance from a few cm to a few meters with less than 1 picojoule per bit energy consumption. The advanced SiGe BiCMOS semiconductor technology (600 GHz fmax) will be used for the monolithic microwave integrated circuits (MMICs). The bipolar transistor is more power-efficient than its CMOS counterpart used for microwave-over-plastic at lower frequencies. Another key issue is the implementation of simple, mechanically stable, and elegant solutions for the interconnect between circuit and fibre, guaranteeing a seamless data link operation. For this purpose, antenna-in-package solutions will be investigated, taking advantage of Infineon's advanced packaging scheme (eWLB). This will offer a high level of integration as it will incorporate the state-of-the-art millimetre-wave SiGe technology along with the package, integrating microwave passives and antennas. To the best of our knowledge, neither the frequency range nor the targeted data rate has been attempted before. Thus, the proposed demonstrator system and performance goals are well beyond the state of the art.

6.3 Radio Link Authentication and Infrastructure Attestation

The use of machine learning for RF fingerprinting has recently emerged as a promising technique for physical layer security for 5G and beyond. The basic premise of RF fingerprinting is that each transmitting device has minor manufacturing imperfections and operation impairments that result in unique, subtle characteristics or discrepancies in the radio signals it emits. These discrepancies, although often very limited, can be detected, measured, and processed allowing to create a fingerprint of the device.

The rise of machine learning has revolutionized the field of RF fingerprinting. Machine learning algorithms, especially deep learning models, are capable of recognizing patterns in data through automatic learning. These techniques can extract features from raw or minimally processed RF signals and have demonstrated impressive results in identification accuracy and resilience against signal variations.

By analysing fingerprint information, the base station and terminal device can mutually check each other's authenticity at the physical layer. Additionally, it provides a layer of low-level infrastructure attestation for distributed code execution. By checking the authenticity of the base station, terminals like connected cars can place more trust in the infrastructure they may use to offload code execution.

Technical challenges we are going to address include:

- development of a lightweight machine learning algorithm for RF fingerprinting,
- algorithm-hardware co-design to have a machine-learning accelerator purpose-built for fingerprinting, and
- securing the connection between fingerprint signal extraction to the machine-learning accelerator.

Because it contains an analogue (RF signal extraction) and digital part (machine-learning accelerator), this work will be split across work packages 4 and 5, with additional details reported in deliverables D4.1 and D5.1.

6.4 Heterogeneous Compute Platform with TEEs

To implement a trustworthy base station and terminal architecture, COREnext needs an integrative platform for digital computation that addresses the following three requirements:

- integration of heterogeneous execution units like processors, accelerators, and FPGAs,
- secure isolation of hardware units by potentially untrusted third-party vendors, and
- secure isolation of software in TEEs.

Integration in the scope of COREnext is focused on the system-on-chip (SoC) level, but the concepts should extend to larger scale deployments up to edge cloud data centres.

Heterogeneous accelerators are a must-have for energy-efficient computation. A COREnext-SoC should be able to accommodate different kinds of processors with and without instruction-set extensions as well as programmable and fixed-function accelerators.

At the same time, we expect a need to integrate COREnext-designed execution units together with hardware components by **third-party vendors.** These vendors may not be fully trusted, making it necessary to securely isolate those hardware components from the rest of the system. As discussed in Section 3.5, capabilities serve as a mechanism for communication control. We grant those third-party hardware components only a minimal set of access rights to limit the blast radius should a component be subverted to perform malicious actions.

Our starting point for this research is the **M³ system architecture.** M³ already fulfils the first two platform requirements, because it can integrate different execution units and exercises communication control between them. Its basis is a tile-based SoC design, where each execution unit lives in its own tile. Tiles are connected by an on-chip network, whereby each such connection is policed by a hardware unit called **Trusted Communications Unit (TCU).** The TCU implements capability enforcement in hardware and thus promises to control communication with very little overhead for throughput and latency.

What is currently missing in M³ is support for TEEs. Trusted execution and remote attestation are necessary to extend the M³ security promises to larger-scale distributed systems. In off-the-shelf hardware architectures, interactions between TEEs and accelerators are notoriously difficult. Because it is a clean-slate redesign, we believe M³ can offer higher security TEEs with natural accelerator integration, thus solving a pressing research problem. With support for TEEs and remote attestation, M³ can become a corner stone of the COREnext architecture.

7 Conclusion

The COREnext project focuses on trustworthiness and efficiency for future Beyond-5G and 6G networks. From the COREnext target use cases elaborated in deliverable D2.1, we derived an architecture proposal centred around:

- a trusted base station and core network infrastructure, and
- a trusted terminal infrastructure with either COREnext or third-party application platform

From this architecture, we identified four clusters of interacting components requiring innovation beyond the state of the art:

- efficient signal processing and acceleration,
- efficient interconnects,
- infrastructure authentication and attestation, and
- trusted digital computation.

The necessary advancements will be designed in work packages 4 and 5, and finally be evaluated in work package 6. Initial results from this work will lead to a refined architecture to be reported in deliverable D3.2 and will also feed back into the use cases and trustworthiness requirements collected in work package 2.

8 References

[1] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb and G. C. Trichopoulos, "Wireless Communications and Applications Above 100 GHz: Opportunities and Challenges for 6G and Beyond," *IEEE Access,* vol. 7, p. 78729–78757, 2019.

[2] L. Singaravelu, C. Pu, H. Härtig and C. Helmuth, "Reducing TCB Complexity for Security-sensitive Applications: Three Case Studies," in *1st ACM SIGOPS/EuroSys European Conference on Computer Systems (EuroSys)*, Leuven, Belgium, 2006.

[3] R. N. Watson, J. Woodruff, P. G. Neumann, S. W. Moore, J. Anderson, D. Chisnall, N. Dave, B. Davis, K. Gudka and B. Laurie, "CHERI: A Hybrid Capability-system Architecture for Scalable Software Compartmentalization," in *IEEE Symposium on Security and Privacy*, Sam Jose, CA, USA, 2015.

[4] N. Asmussen, M. Völp, B. Nöthen, H. Härtig and G. Fettweis, "M3: A Hardware/Operating-system Co-design to Tame Heterogeneous Manycores," in *21st International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Atlanta, GA, USA, 2016.

[5] N. Asmussen, M. Roitzsch and H. Härtig, "M³x: Autonomous Accelerators via Context-Enabled Fast-Path Communication," in *USENIX Annual Technical Conference (USENIX ATC)*, Renton, WA, USA, 2019.

[6] "Study on Ambient IoT (Internet of Things) in RAN," in *3GPP TR 38.848 V0.2.0*, 2023.

[7] "Study on low-power Wake-up Signal and Receiver for NR," in *3GPP TR 38.869 V0.2.0*, 2023.

[8] O-RAN Alliance, "O-RAN.WG3.RICARCH-R003-v04.00 Technical Specification, Near-RT RIC Architecture," 2023.

[9] S. Poretsky and J. S. Boswell, "Security Considerations of Open RAN," August 2020.

[10] 3GPP, "NR and NG-RAN Overall description," in *TS 38.300 v17.5.0*, 2023.

[11] O-RAN Alliance, "O-RAN.WG1.OAD-R003-v09.00 Technical Specification, Architecture Description," 2023.

[12] O-RAN Alliance, "O-RAN.WG4.CUS.0-R003-v12.00 Technical Specification, Control, User and Synchronization Plance Specification," 2023.

[13] 3GPP, "F1 Application Protocol (F1AP) Specification," in *TS 38.473 v17.5.0*, 2023.

[14] 3GPP, "Service Data Adaptation Protocol (SDAP) Specification," in *TS 37.324 v17.0.0*, 2022.

[15] O-RAN Alliance, "O-RAN.WG6.AAL-GAnP-R003-v06.00 Technical Specification, Acceleration Abstraction Layer General Aspects and Principles," 2023.

[16] X. You, C.-X. Wang, J. Huang, X. Gao, Z. Zhang, M. Wang, Y. Huang, C. Zhang, Y. Jiang and J. Wang, "Towards 6G Wireless Communication Networks: Vision, Enabling Technologies, and New Paradigm Shifts," *Science China Information Sciences,* vol. 64, p. 1–74, 2021.

[17] L.-H. Shen, K.-T. Feng and L. Hanzo, "Five Facets of 6G: Research Challenges and Opportunities," *ACM Computing Surveys,* vol. 55, no. 11, p. 1–39, 2023.

[18] M. Bertuletti, Y. Zhang, A. Vanelli-Coralli and L. Benini, "Efficient Parallelization of 5G-PUSCH on a Scalable RISC-V Many-Core Processor," in *Design, Automation \& Test in Europe Conference \& Exhibition (DATE)*, Antwerp, Belgium, 2023.

[19] V. Razilov, E. Matúš and G. Fettweis, "Communications Signal Processing Using RISC-V Vector Extension," in *International Wireless Communications and Mobile Computing (IWCMC)*, Dubrovnik, Croatia, 2023.