D4.1

Concept for Hardware Security Primitives and Heterogenous Acceleration

Funded by
the European Union

Revision v1.0

Work package	WP4
Task	T4.1–4.4
Dissemination level	PU
Deliverable type	R
Due date	30-09-2023
Submission date	25-09-2023
Deliverable lead	TUD
Version	v1.0
Authors	Viktor Razilov (TUD), Marco Bertuletti (ETHZ), Yichao Zhang (ETHZ), Alessandro Vanelli Coralli (ETHZ), Hendrik Borchert (IHP), Markus Ulbricht (IHP), Romain Beurdouche (EUR), Panagiotis Domestichas (WINGS), Pavlos Alexias (WINGS), Andreas Georgakopoulos (WINGS), Mohand Achouche (IIIV), Julien Lallet (NNF), Renaud Santoro (NNF), Michael Roitzsch (BI), Anastasia Grebenyuk (EAB)
Contributors	Work package partners (see below)
Reviewers	José Luis Gonzalez Jimenez (CEA)

Abstract

The COREnext project strives to develop trustworthy and efficient Beyond-5G and 6G mobile networks by means of efficient digital signal processing and mechanisms for isolation and orchestration. We propose the components to enable this vision. For the former, we will design heterogeneous RISC-V-based accelerators for various steps in the radio access network processing chain. For the latter, we will design orchestration solutions on accelerator-, device- and network-level. In later deliverables, we will report on the development efforts and results.

Keywords

heterogeneous accelerators, isolation, orchestration, RISC-V, many-core, vector processor, FEC, MAC scheduling, multi-tenancy, micro-kernel, IoT, radio fingerprinting

Document Revision History

Version	Date	Description of change	Contributor(s)
v0.1	07-07-2023	Initial version of deliverable	Viktor Razilov (TUD)
v0.2	18-08-2023	First draft of deliverable	Viktor Razilov (TUD), Marco Bertuletti (ETHZ), Yichao Zhang (ETHZ), Alessandro Vanelli Coralli (ETHZ), Hendrik Borchert (IHP), Markus Ulbricht (IHP), Romain Beurdouche (EUR), Panagiotis Domestichas (WINGS), Pavlos Alexias (WINGS), Andreas Georgakopoulos (WINGS), Mohand Achouche (IIIV), Julien Lallet (NNF), Renaud Santoro (NNF), Michael Roitzsch (BI), Anastasia Grebenyuk (EAB)
v0.3	08-09-2023	First review	José Luis Gonzalez Jimenez (CEA)
v0.5	15-09-2023	Reviewers' comments addressed	Viktor Razilov (TUD), Romain Beurdouche (EUR)
v1.0	25-09-2023	Submission version	Viktor Razilov (TUD), Romain Beurdouche (EUR), Michael Roitzsch (BI)

Contributing Partners

Abbreviation	Company name
BI	BARKHAUSEN INSTITUT
EAB	ERICSSON
CYB	CYBERUS TECHNOLOGY
EUR	EURECOM
TUD	TECHNISCHE UNIVERSITAET DRESDEN
WINGS	WINGS ICT SOLUTIONS
ETHZ	EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH
IHP	IHP MICROELECTRONICS
NNF	NOKIA NETWORKS FRANCE
IIIV	NNF/IIIV LABS
KAL	KALRAY

Disclaimer

Copyright Notice

Executive Summary

In this deliverable, we derive digital components to build the architecture and its component clusters outlined in D3.1. The component clusters of interest are:

- Power-efficient signal processing and
- Heterogeneous compute platform with trusted execution environments.

We plan to implement the first with the following heterogenous RISC-V-based accelerators:

- Many-core accelerator,
- Vector processing accelerator,
- Forward error correction (FEC) accelerator, and
- MAC scheduling accelerator.

Our main component developments in isolation and orchestration will be:

- Field-programmable gate array (FPGA) multi-tenancy,
- Accelerator virtualization,
- M³, a microkernel-based system for heterogeneous multicores, and
- Artificial intelligence (AI) inference for radio link authentication.

For each of these components, we provide a detailed description of our implementation approach and highlight research challenges. In the follow-up deliverables D4.2 and D4.3 we will report on the component development for the heterogeneous acceleration and the isolation and orchestration mechanisms, respectively.

Table of Contents

List of Figures

Acronyms and Definitions

5G	Fifth generation (mobile network standard)
6G	Sixth generation (mobile network standard)
AI	Artificial intelligence
ASIP	Application-specific instruction set processor
BCCH	Broadcast control channel
BER	Bit error rate
CCCH	Common control channel
CNN	Convolutional neural network
CP	Cloud provider
CPU	Central processing unit
CRC	Cyclic redundancy check
CU	Centralized unit
DL	Downlink
DLP	Data-level parallelism
DNN	Deep neural network
DPU	Data processing unit
DRAM	Dynamic random-access memory
DTU	Data transfer unit
DU	Distributed unit
FEC	Forward error correction
FPGA	Field-programmable gate array
GPU	Graphical processing unit
HARQ	Hybrid automatic repeat request
HPC	High performance computing
IoT	Internet of things
IP	Intellectual property
IQ	In-phase and quadrature
ISA	Instruction set architecture
L1	Level one (memory hierarchy)
LDPC	Low-density parity check
MAC	Medium access control
MANO	Management and orchestration
MIMO	Multiple-input and multiple-output
ML	Machine learning

OFDM	Orthogonal frequency division multiplexing
PA	Power amplifier
PCCH	Paging control channel
PDCP	Packet data convergence protocol
PHY	Physical layer
POLA	Principle of least authority
PUSCH	Physical uplink shared channel
QoS	Quality of service
RACH	Radio access channel
RAN	Radio access network
RF	Radio frequency
RLC	Radio link control
RRC	Radio resource control
RU	Radio unit
RVV	RISC-V Vector
SDAP	Service data adaption protocol
SIMD	Single-instruction multiple-data
SSL	Secure sockets layer
TA	Trusted authority
TCU	Trusted communication unit
TEE	Trusted execution environment
TLS	Transport layer security
TTI	Transmission time interval
UL	Uplink
VIM	Virtual infrastructure manager
VPU	Vector processing unit
VRF	Vector register file
WP	Work package
XR	Extended reality

1 Introduction

Mobile network applications are becoming an increasingly ubiquitous part of our lives. In the Beyond-5G and 6G era, a plethora of devices will sense our everyday environment and communicate to other machines to fulfil various needs of industrial production and consumers. Considering these developments, it is crucial to find solutions that bar malign actors from accessing personal data and keep the energy consumption of the network as low as possible. COREnext's mission is therefore to build trustworthy, yet sustainable, mobile networks.

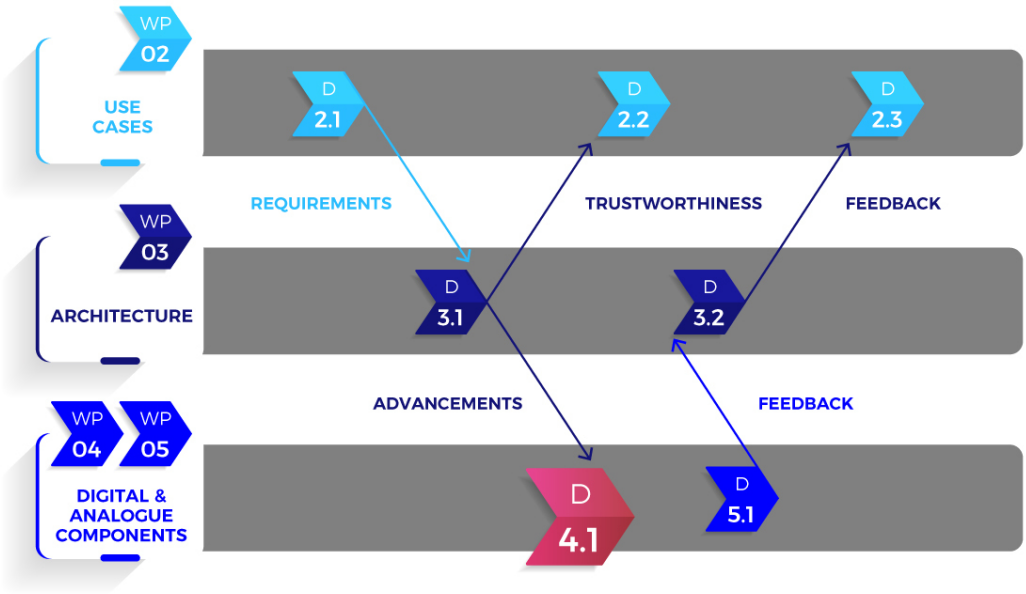

Figure 1: Overview of selected COREnext deliverables and the information flow between them.

An overview of COREnext deliverables and the dependencies between them is provided in Figure 1. COREnext has selected and analysed three prominent 6G use cases that are vulnerable to breaches of trust in the deliverable D2.1:

- extended reality (XR),
- automotive infrastructure,
- smart city.

They represent the use case families of

- enhanced human communication and entertainment,
- enhanced machine communication, and
- intelligent management,

respectively. D2.1 has outlined requirements posed by these use cases. Subsequently, the deliverable D3.1 has transformed these requirements to a project-wide architecture. It identified a set of digital and analogue components that need further development to implement the architecture. Work packages 4 and 5 bring these suggestions to fruition for digital or analogue components, respectively, and generate feedback on the architecture design. This deliverable lays the groundwork

for this undertaking in work package (WP) 4 which deals with digital components. WP5 will present the equivalent deliverable D5.1 about the analogue parts of the network.

The next section summarizes the proposals of D3.1, puts them into context in different parts of the network, and reviews their implication with regards to trustworthiness or efficiency. Section 3 describes the individual component advances to be made on a detailed technical level. Section 4 concludes with a summary and outlook.

2 Architecture

Based on an analysis of the three use cases XR, automotive infrastructure, and smart city, the COREnext deliverable D3.1 has devised an architecture for trusted and efficient base station, core network, and terminal infrastructure. The architecture is composed out of:

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

Of these four component clusters, the first two relate to efficiency and the last two to trustworthiness. The first and the last component require innovation in the digital domain, the middle two in the analogue one. As the scope of this deliverable is on the former, we describe how we aim to implement power-efficient signal processing by means of RISC-V-based acceleration and how hardware security primitives take care of isolation and orchestration in the heterogeneous compute platform with trusted execution environments.

2.1 RISC-V-Based Acceleration

As described in D3.1, COREnext targets the design of a heterogeneous platform that can sustain the high throughput and low power consumption required for base-station processing with application-specific accelerators tailored to the workload needs. In this framework, RISC-V offers relevant opportunities. As an open-source instruction set architecture (ISA), RISC-V allows for extensive customization and optimization, enabling the development of highly specialized accelerators. Its modularity and simplicity, coupled with its growing ecosystem, make RISC-V a good option for designing application-specific accelerators that can efficiently handle the complex workloads of base-station processing, and an attractive choice for industry players in Europe [1].

The project will mainly target the acceleration of the most computationally demanding functions of the processing chain, that according to the proposed O-RAN functional splits can be executed at the distributed unit (DU) or centralized unit (CU) of the network to reduce their cost in terms of latency and execution time [2]. Among the tasks to be accelerated, we consider processing steps of the Lower Media Access Control Layer (MAC), and Lower and Upper Physical Layer (PHY), which can require processing inside a sub-millisecond Transmission Time Interval (TTI).

COREnext aims at providing acceleration of the wireless functions keeping the hardware as flexible and reconfigurable as possible, to keep up with the fast-evolving standards. This target is achieved by two degrees of proximity of the accelerators to the programmable processing elements data path:

- Loosely coupled accelerators are standalone application-specific digital signal processing accelerators that can handle input and output data streams. They are intended as processing islands that can be plugged into the system interconnect, configured, and activated by the system's host. In a basic configuration, a RISC-V programmable processor can activate an accelerator and control its status, by means of reads/writes to its memory-mapped control status registers. RISC-V ISA also allows the implementation of specific instructions for the offloading

and the set-up of the input and output streams. We believe this approach is useful to accelerate the functions in the Lower and Upper-PHY which are characterized by repetitive workloads.

- Tightly coupled accelerators are intended as extensions to the RISC-V ISA. RISC-V extensible ISA also gives the possibility to introduce instructions of key importance for the telecommunications workload, including for example complex multiply and accumulate operations, radix-N butterfly calculations, and CORDIC iterations [3]. The hardware to support these instructions can be included in the data path of fully programmable processors. Custom compiler support for the implemented instructions allows them to be used in low-level programming languages such as C and C++. The ISA enhancement of RISC-V processors allows shifting the paradigm of wireless functions acceleration from the use of dedicated standalone loosely-coupled accelerators to a software-defined approach, ensuring higher flexibility and adaptability to the standard requirements.

On this side, COREnext explores two different architectural solutions to handle the large dimensionality signals of the wireless telecommunication stream in the 5G Lower-PHY processing. First, the project will consider large clusters of scalar RISC-V processing elements with tightly-coupled data memory, leveraging large-scale parallelism to meet the required performances [4]. Second, it will investigate the potential of data-level parallelism (DLP), by leveraging vector processors supporting the RISC-V Vector (RVV) ISA extension set on highly data-parallel tasks [5]. Hybrid solutions are also possible, and acceleration can be obtained by clustering many scalar processors, each with its own Vector Processing Unit (VPU), supporting the RVV instruction set. This combines data and instruction-level parallelism to boost the performance of the processing engine.

2.2 Isolation and Orchestration

The architectural choices done within WP3 and presented within the deliverable D3.1 imply that some effort should be put on designing orchestration and isolation that will fit the heterogeneous, disaggregated, and trustworthy nature of the architecture.

The global architecture stated in D3.1 includes some characteristics that will put constraints on the way to operate the computing infrastructure. The architecture relies on a pool of hardware resources where each resource could be different in many ways. For example, they are either general-purpose or purpose-built, designed for intensive computation tasks or not, fully trusted or not, able to handle multiple tasks in parallel or not, or have even further differences. Digital-analogue converters as well as radio devices should also be considered in the operation of the infrastructure. Among the different digital resource types, there will be TEEs (Trusted Execution Environments) and DPUs (Data Processing Units) that are special boards with built-in high throughput wired network interfaces that are able to process workloads coming through the wired network without going through the bus of a computer. The infrastructure may also interface with third-party base stations or base station components.

In addition to being heterogeneous, the infrastructure would be distributed among distant locations with dissymmetric distribution of workloads and resources to comply with the disaggregated nature of the architecture. Splitting different components of a network between cell sites, edge sites and central sites makes that the computing resources should also be split among those sites in a manner that is relevant to the splitting of the network (software) components.

Managing this complex infrastructure should be done with regards towards the efficiency in terms of resource and energy utilization. Operating this infrastructure should not lead to wasting resources in over-provisioning.

The management of the infrastructure also plays a central role in ensuring trustworthiness. It should ensure an appropriate usage of the infrastructure so that it mitigates attempts of weakness exploits or secret leakage. This must be achieved through multiple mechanisms of threat mitigation and data protection. It includes the monitoring of the activity of software and hardware components – especially their access to data – and the enforcement of restriction policies. It also includes providing the users with encrypted data channels toward authenticated components. This is to comply with the expectation of users for trusted computing by design. The users want their sensitive or valuable data to be protected without having to rely on the ability and willingness of a service provider to ensure its safety. The protection of data should be ensured by design of the infrastructure.

The infrastructure will be shared by multiple network tenants whose workloads should remain in isolation between them and with the infrastructure for the good of both the usage experience and trustworthiness. This is important regarding trustworthiness since it mitigates the threat of potential malicious users.

As was already explained in the building blocks study of D3.1, state-of-the-art virtualization technologies will be used as a base to ensure the management of the infrastructure. Virtualization within cloud-infrastructure is designed to enable the efficient operation of a large computing infrastructure while sharing this infrastructure among multiple users that are isolated between them and from the complexity of the infrastructure.

A virtualization technology mostly relies on some components that perform Management and Orchestration (MANO). MANO components include among other things a Virtual Infrastructure Manager (VIM). The VIM allows to have software components packaged as containers run on an appropriate device. The play of the other components of MANO is to scatter the components across the infrastructure, set them up and manage their runtime.

By incorporating orchestration of resources and isolation of functions into various devices and components, organizations can achieve trustworthy communications. These practices provide centralized control, uniform security policies, and rapid threat response through orchestration while reducing the risk of unauthorized access and data breaches through isolation. Together, they play a crucial role in establishing a secure and reliable operational environment that users can trust to safeguard their data and ensure the integrity of communications. Specifically, orchestration involves the automated process to deploy, configure, integrate & manage an application, service, or resource. It allows for efficient and consistent control of data flow, message routing, and access privileges. Through orchestration we can enforce security policies, manage authentication, and handle encryption across all connected components, ensuring a trustworthy environment.

But the architecture that is aimed integrates some cutting-edge features that may not be yet fully supported by state-of-the-art virtualization technologies. There are therefore some challenges awaiting to design, implement and verify a virtualization technology that would be able to fully handle the aimed architecture.

This architecture will be unique by its heterogeneity. It will require an appropriate MANO that can make a relevant use of the different components.

The MANO should also be aware of disaggregation and of the underlying geographical and functional scattering of resources and containers.

The architecture will integrate new kinds of components (DPUs, TEEs). Those components may have never been used before for the purpose of implementing virtual mobile network functions. This raises numerous questions. It may be necessary to figure out how the VIM should use these components. Special care should be given to mitigating impersonation of these components. A breach in the authentication of these components breaches the protection of users' data since it allows data theft by impersonation. The architecture should be secure by design so that it must endorse the authentication of components.

To achieve this purpose in the field-programmable gate array (FPGA) case, the architecture will integrate a trusted authority (TA). Its role will be to authenticate pairs of FPGA and client applications and guarantee the isolation between the FPGAs and the infrastructure provider. This TA is introduced in part 3.5 of this document.

Trustworthiness may bring additional or stronger requirements compared to what virtualization alone can achieve. Some improvement will have to be done to the isolation to address those requirements.

WP4 will have to deliver a virtualization technology based on state-of-the-art with improvements to the orchestration and isolation to match the requirement of the architecture aimed by the COREnext project.

3 Digital Components

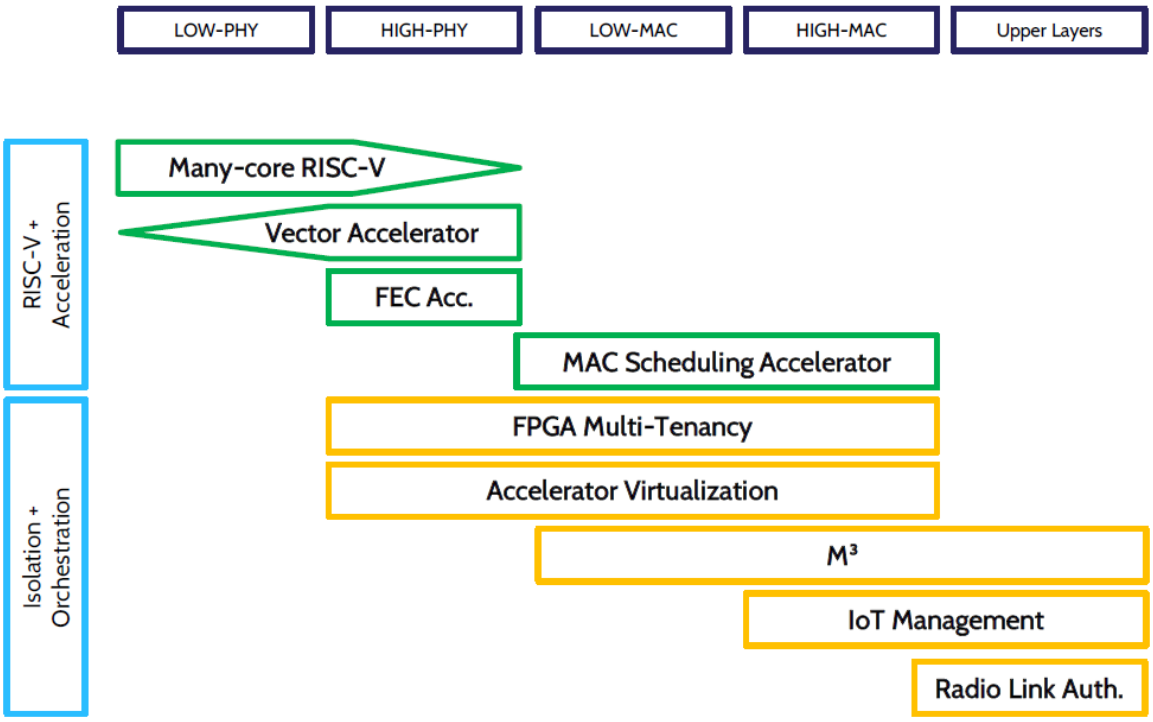

Figure 2: Overview of components in network

For fully functional communication systems - especially for complex ones as defined by 3GPP and O-RAN Alliance – a wide variety of functions must be implemented. The processing steps can be grouped as in the top row of Figure 2.

At first, the radio unit (RU) samples the antenna signal which is then processed in the PHY, which can be divided into LOW-PHY and HIGH-PHY. The former includes orthogonal frequency division multiplexing (OFDM) and digital beamforming, both being tasks that need on-flight processing. These tasks are performed in the RU and then sent to the DU via the fronthaul. On the uplink (UL), HIGH-PHY requires channel estimation and multiple-input and multiple-output (MIMO) decoding of the incoming signal. Both are highly parallel tasks to be executed for a large number of independent sub-carriers [6].

Besides HIGH-PHY, the DU includes the MAC and the Radio Link Control (RLC). MAC modules encompass a variety of tasks, including Random Access Channel (RACH) management, Hybrid Automatic Repeat Request (HARQ) management, downlink (DL) and uplink (UL) data processing, Broadcast Control Channel (BCCH), Paging Control Channel (PCCH), and Common Control Channel (CCCH) processing, as well as MAC Transport Block formation. These tasks are defined in the 3GPP technical specification TS 38.321 [7]. The LOW-MAC generally handles tasks that require close interaction with the PHY, such as HARQ management and MAC Transport Block formation, ensuring efficient and reliable data transmission. The HIGH-MAC consists of all remaining MAC tasks and the CU includes the packet data convergence protocol (PDCP), the service data adaption protocol (SDAP) and the radio resource control (RRC).

As part of this project, we have identified suitable accelerators within the LOW-PHY, the HIGH-PHY and the LOW-MAC to be implemented. Figure 2 shows the digital components COREnext plans to develop in relation to the component clusters (c.f. section 2.1) and the radio access network (RAN) processing chain. Among the heterogeneous RISC-V-based accelerators, there is the many-core RISC-V accelerator geared towards LOW-PHY (c.f., section 3.1) and a vector-processing-based one that is more suitable for HIGH-PHY processing (c.f., section 3.2). The outstandingly demanding forward error correction (FEC) task in the HIGH-PHY and the MAC scheduling are the task of other bespoke accelerators (c.f., section 3.3 and section 3.4, respectively).

The challenge of orchestration and isolation is addressed on multiple levels:

- On the level of a single accelerator that is shared among multiple tenants. FPGA-based accelerators are covered in section 3.5 and the virtualization of any accelerator in section 3.6.
- On the level of multiple untrusted accelerators and processors that are integrated on a single device. The microkernel-based system for heterogeneous many-cores M³ is pitched in section 3.7 as solution for keeping the overall device trustworthy.
- And on the level of multiple devices in an Internet of Things (IoT) as detailed in section 3.8.

Finally, we investigate in section 3.9 inference by artificial intelligence (AI) for radio frequency (RF) fingerprinting for radio link authentication.

3.1 Many-Core RISC-V Accelerator for Low-PHY Processing

In the 5G processing chain, the LOW-PHY layer processing sets challenging requirements in terms of latency and throughput. For example, in a typical use-case for the Physical Uplink Shared Channel (PUSCH), the receiving base station is required to process frequency-multiplexed transmissions counting hundreds of subcarriers on flight, in a timeframe of less than 1ms.

The signal processing required, from the reception by the base-station antenna array to the delivery to HIGH-PHY might include, for example, the following main algorithms: OFDM demodulation, Digital Beamforming, Channel Estimation, Channel Interpolation, and MIMO decoding. This workload maps to highly data-parallel or even data-oblivious operators, such as the Fast Fourier Transform, the Matrix-Matrix and Matrix-Vector Multiplication, and the Element-Wise Matrix Division, which are directly applied to the input large-dimensional vectors [4].

In the case of MIMO decoding, which can be implemented in many ways, but always requires a linear-system inversion as the most demanding, rich in data dependencies, computation step, operations can still be parallelized. In this case, a large number of independent subcarriers can indeed be separately processed.

Deep Learning models are paving their way in 5G and Beyond-5G processing, to speed up parts of the processing chain [8]. In the LOW-PHY Deep Neural Networks (DNNs) can be adopted to replace parts of the receiving chain, such as the Channel Estimation and the MIMO decoding. Deep Learning processing consists of embarrassingly parallel workloads, including Tensor Multiplications and Multi-Dimensional Convolutions.

For these reasons, 5G/Beyond-5G and 6G processing will strongly benefit from execution on parallel hardware. The problem can be addressed with application-specific parallel data paths, but as the standards for 5G signal processing keep evolving at a fast pace, the re-programmability or software-programmability of hardware must be considered a cornerstone of the design process, to guarantee reusability, adaptive behaviour in a wide range of application scenarios, and fast time-to-market of products. Multi-core and Many-Core programmable processors are therefore promising candidates to accelerate 5G processing.

In COREnext WP4, we will focus on the development of Many-Core accelerators built assembling hundreds, up to a thousand programmable cores, supporting a full ISA. A successful architectural pattern for parallel computing is a tightly coupled cluster of processing elements sharing low-latency and high-bandwidth first level (L1) memory. For example, the MemPool and TeraPool design [4] use a hierarchical design approach. A set of lightweight cores tightly coupled to L1 memory macros form the basic Tile building block, represented in Figure 3, which is specifically tailored for massive replication allowing to build clusters with hundreds of processing elements.

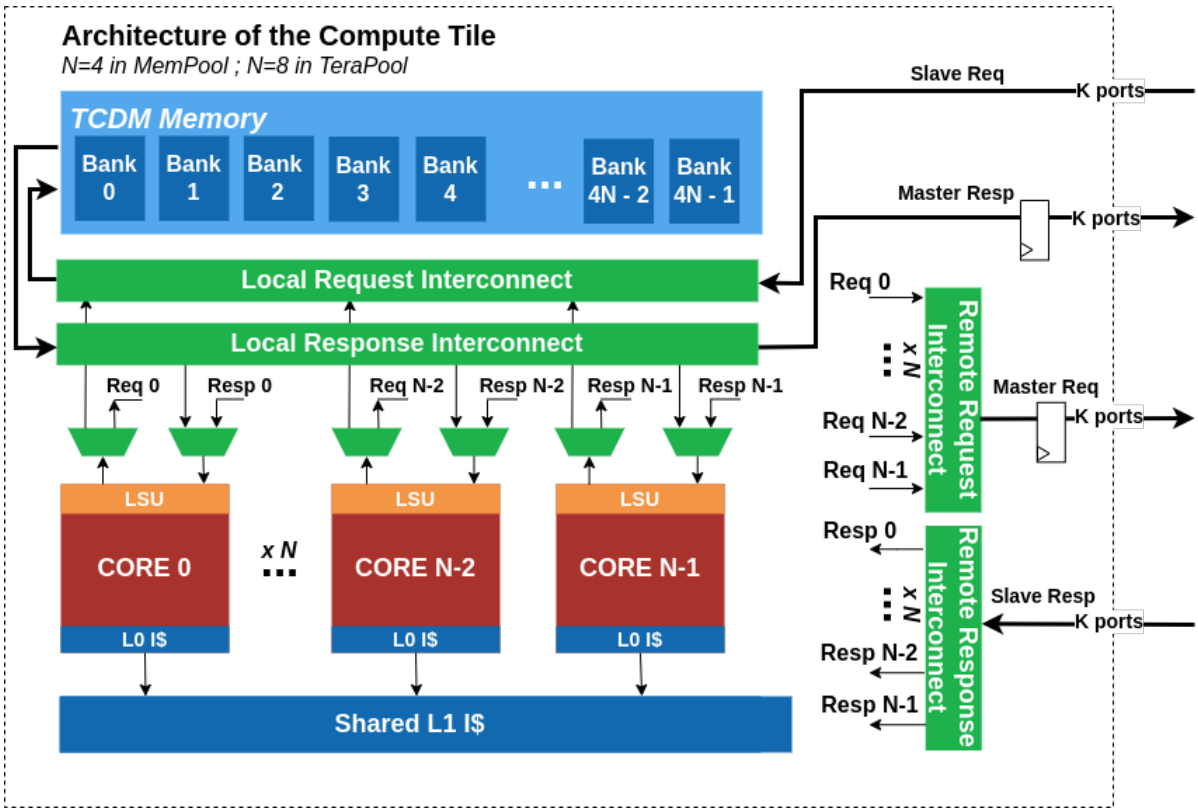

Figure 3: Architecture of a Tile in the MemPool and TeraPool many-core processors

We will address the software implementation of parallel data kernels on these many-core platforms and address demanding use cases for 5G/B5G signal processing. The implemented solutions will be integrated into the overall system architecture that will provide orchestration and trustworthiness via dedicated connections.

To efficiently map the kernels on our programmable cores, we will implement ISA extensions specific to the target workload yet maintaining the cores fully programmable. For this reason, we rely on the open extensible RISC-V ISA.

The efficient processing of large-dimensional signals requires scaling up the memory system as well, to make sure that the speed up of the implemented kernels is not bound by expensive memory transfers. Ideally, hundreds of cores should be able to work in parallel on input vectors, fully residing in the L1 memory hierarchy level. In the implementation of the Many-Core processor, to ensure maximum flexibility in the allocation of data in memory and allow the processor to be easily programmed with standard OpenMP or Halide frameworks, we will implement a multi-banked scratchpad memory. The interconnection required to assemble the cores and the memory is a key enabler of our architecture and will be developed with a hierarchical approach, to make sure that any bank can be accessed by any core in a small number of cycles while keeping the design fully feasible at appealing operational frequencies.

3.2 Programmable Vector Processing Accelerator

Fixed-function accelerators provide excellent efficiency but are difficult to virtualize, as detailed in D3.1. Additionally, custom chip-development and production may not be feasible for every application. Efficient programmable accelerators that lie between general-purpose processors and fixed-function accelerators or application-specific instruction set processors (ASIPs) on the performance-flexibility trade-off curve are therefore desirable.

Vector processors have found a renewed interest for their ability to efficiently exploit DLP in a programmable manner. Popular general-purpose ISAs, e.g., RISC-V and ARM, have added vector processing extensions. While instructions in conventional processors operate on single data items and thus have to be issued multiple times for multiple data items, vector instruction set extensions operate on whole vectors of data. This allows for significant time and energy savings in programs with a lot of DLP as the instruction fetch and decode is one of the major sources of energy overhead in programmable processors.

The underlying model of vector processors differs from array processors, the paradigm of previous single-instruction multiple-data (SIMD) ISA extensions. In the latter, large processing elements consume and produce the operand vectors at once, whereas in the former, the operand vectors may be processed in time-multiplexed chunks. The vector length can thus be significantly larger, further reducing the instruction fetch overhead. Vector processors also tend to utilize the provided processing elements better as they execute instructions in a pipelined fashion through chaining.

Many computationally intensive tasks that exhibit a high degree of DLP lend themselves to vector processing, including communications signal processing and AI. Previous research has investigated the use of conventional RISC-V-based vector processors [5]. While they achieve reasonable performance for communications number-crunching tasks, they are not yet at the efficiency needed for next generation mobile networks. Our goal is therefore to improve their instruction set and microarchitecture, based on the insights found in the virtualized RAN application.

One utilization optimization we found in the preparation and early phase of the COREnext project is the dual vector load: a parallel or interleaved load of two vectors that are input operands to a follow-up instruction. The latter can thus begin execution earlier, before the preceding instructions completed. We analysed this feature theoretically and determined that it is beneficial for compute-

bound and some memory-bound programs. The highest possible speedup is 33 % and we demonstrated a speedup of 21 % in an implementation with about 2 % area overhead [9].

Further, we identified the vector register file (VRF) as another bottleneck. For efficient pipelining, it needs to support multiple concurrent read and write accesses. While banking can mitigate many of the contentions, it also associated with an area overhead. A large VRF, that is efficient for pipelining and reduced instruction fetch overhead, also leads to a high context-switch overhead which needs to be considered when vector processors are to be used in a virtualized environment. We therefore want to investigate architectural alternatives to address these shortcomings.

3.3 FEC Accelerator

The PHY handles transport block segmentation, cyclic redundancy check (CRC) generation, FEC via e. g. low-density parity check (LDPC), rate matching, scrambling, modulation, and many more tasks. In the course of COREnext, the acceleration of CRC and FEC will be tested in particular.

CRC is crucial for error detection within the data and can be used to trigger HARQ retransmissions. They are generated for both transport- and code blocks and thus not only allow for a high error detection probability but also fine-grained and, therefore, efficient data resending in the case of uncorrectable errors. An implementation as RISC-V ISA extension seems appropriate here since CRC without any acceleration could either become a bottleneck or decrease energy efficiency [10]. However, a free-standing accelerator seems excessive at this point. Hence, integration in a freely programmable processor is suitable, which could also be used for other purposes simultaneously (c.f. section 2.1).

FEC is one of the essential processing blocks in many communication systems. Firstly, it provides reliable communication between two endpoints by correcting bit errors in the demodulated data stream. Secondly, it increases energy efficiency and the effective data rate by reducing the number of retransmissions. There are many different algorithms that can be used to implement FEC. Depending on the specific application, certain algorithms may be more appropriate than others.

LDPC is a soft-decision coding scheme with relatively uncomplicated decoding algorithms at hand. Because of that and the superior performance with respect to bit error rate (BER), when compared to hard-decision coding schemes, it quickly dominated FEC processing and has been adapted into many communication standards. We are witnessing ever-increasing data rates, and the barrier of 100 Gb/s wireless communication has already been exceeded several times [11]. We expect data rates of 1 Tb/s or higher in the next few years. Due to this reason, we propose research on hardware architectures supporting unrolled FEC decoders. The LDPC supports message-passing decoding, which is difficult to realize efficiently in a traditional general-purpose processor due to the massive stress on data move operations. Thousands of bits have to travel across the decoding graph.

The LDPC computation steps are not complicated per se. But the number of bits that have to be handled in every clock cycle and exchanged in the processing graph is vast for multiple reasons:

1. All data bits in modern FEC decoders use soft values to represent bit information. These represent the confidence of the bit decision. Usually, that bit confidence is represented by 5 or 6 soft-bit log-likelihood values [12]. Thus, each edge in the decoding graph sends 5 or 6 bits representing one binary data-bit value.

2. All modern decoders are oriented on long codewords. E.g., a single codeword in the 5G-NR standard might accumulate up to 26112 bits, and a decoding graph with 26112 variable nodes and 17664 check nodes is needed [13]. When this data is represented as soft bit values with the minimal specified 5-bit precision [13], [14], this already accumulates to 130560 soft bits, which have to be processed in every iteration.
3. LDPC is an iterative algorithm, which usually has to be performed multiple times to give good BER results [12], [14].

An architecture based on central processing units (CPUs) with 64 Bit data buses cannot effectively handle this amount of data moves. However, when a dedicated hardware decoder is considered, efficiently implementing the decoding graph becomes feasible. The iterative LDPC algorithm can be easily unrolled and pipelined. Each processing node can be designed as an individual hardware node, where the multi-bit graph connections are directly realized as paths in the chip metallization stack. By intelligently designing the paths on the chip, speeds that are not achieved by today's LDPC accelerators could be realized. Among other things, this means low-level intervention in the hardware design through changes in placement and routing. Thus, moving a high number of soft bits and processing them in one clock cycle becomes feasible.

3.4 MAC Scheduling Accelerator

The MAC layer is responsible for various control tasks such as execution of random-access procedures, maintenance of uplink time alignment, ensuring correct data transmission through HARQ in both uplink and downlink data transfer, managing uplink scheduling requests, downlink scheduling, compliance with Quality of Service (QoS) requirements and many others, as well as the creation of transport blocks [7].

Many of these features require the instantiation of complex state machines to represent the control functions. Typically, these state changes must be processed sequentially. Using general-purpose CPUs, such as those based on the RISC-V ISA, as the main processing unit is a suitable approach.

Scheduling DL- and UL-data streams is one of the most important tasks of the MAC. Ultra-high throughput and quality of service (QoS) requirements demand an intelligent, high-performance scheduler with low latency. Here, the scheduler must find a decent solution for a complex optimization problem that includes parameters such as throughput, latency, and general network utilization. In the recent past, artificial intelligence has proven to be capable of solving complex, multivariate problems outstandingly well. On commodity hardware such as graphic processor units (GPUs) these kinds of applications are exceptionally energy hungry. AI accelerators provide this technical advance to a MAC scheduler in an energy-efficient and sustainable manner. We strive for a detailed investigation of this research area and aim to find a suitable accelerator implementation that can be tightly coupled to a RISC-V processor via ISA extension.

3.5 FPGA Multi-Tenancy

FPGA-enabled cloud computing is getting more and more common as cloud providers (CPs) offer hardware accelerated solutions. In this context, clients need confidential remote computing. However Intellectual Properties (IPs) and data are being used and communicated. Cloud security is

critical for a client when choosing a commercial CP. Commercial cloud users expect secure remote computation and access to FPGA accelerators with minimal impact on their design performance. Security mechanisms need to be adapted for an appropriate cloud usage. First, the client needs to ensure that its data is kept private. The client does not want to disclose sensitive IP and data to the CP. To ensure that, the client needs an encrypted channel with the FPGA isolated from the CP. Furthermore, authentication is another important security aspect to establish secure remote connection between a client and the hardware acceleration material. The client needs to ensure that the correct FPGA is used and that no other users may access the allocated resources. Authentication is necessary to manage FPGAs and different cloud service accesses to mitigate client impersonations and data breaches.

Methods used by different CPs lack of transparency concerning data encryption methods, bitstream protection and IP theft. To remove this drawback, it is necessary to use methods and protocols which respect user privacy and intellectual property. A solution to reinforce these aspects is to introduce an intermediate authority between the client and the CP. This authority would be similar to already existing entities in the Public Key Infrastructure mechanism (e.g., certificate authority). Thus, we need an entity that the CP and the client can trust so they do not have to trust each other. The TA [15] [16] [17] [18] [19] serves this purpose. Its role is to authenticate the client-FPGA pair and isolate them from the CP.

Current security models require the client to trust the CP blindly by disclosing sensitive information. In addition, the lack of strong authentication and access control mechanisms, for both the client and the provided FPGA in current solutions, is a major security drawback. In existing solutions, only the client authenticates with the CP. To establish a trust relation, the TA needs to be introduced to the client and have a way to authenticate him without requiring the CP's services. The client needs a transparent authentication scheme with TA to establish the basis of the secure and remote FPGA access. In current FPGA cloud solutions, clients use virtual machines to access their resources. There are no other security measures to protect the resource. Thus, a compromised virtual machine can lead to malicious behaviour and client impersonation.

With the introduction of a TA, we plan to solve problems like user privacy, user, and hardware authentication with third party implication, and create a private channel between the FPGA and the client, isolated from the CP and the TA. Finally, bypassing current tools like virtual machines and offering a direct secure client-FPGA channel ensures privacy and data protection. From a client's perspective, the TA achieves device authentication and isolation from the CP by using the shared secret inside the FPGA. Thus, the client can protect its sensitive IP and data from the CP. From the CP's perspective, the TA achieves tasks like FPGA access management and authentication.

Our proposal is to adapt OAuth 2.0-based access delegation solution for FPGA-accelerated clouds. A remote confidential FPGA environment with a token-based access can be created for the client. Our solution allows to manage and securely allocate heterogeneous resource pools with enhanced privacy & confidentiality for the client. OAuth 2.0 is a secure access delegation open standard where a resource owner can share resources with a client thanks to a common trust placed in a third party (i.e., the TA) [20]. Our solution adapts this protocol for a cloud-enabled FPGA context. As presented in Figure 4, this solution aims to provide an authentication solution for 4 entities simultaneously (FPGA, client, TA, and CP) and achieves perfect isolation between the client and the

CP. In this situation, the CP is considered as the resource owner, the TA is referred as the authorization server and the FPGA is a part of the resource server.

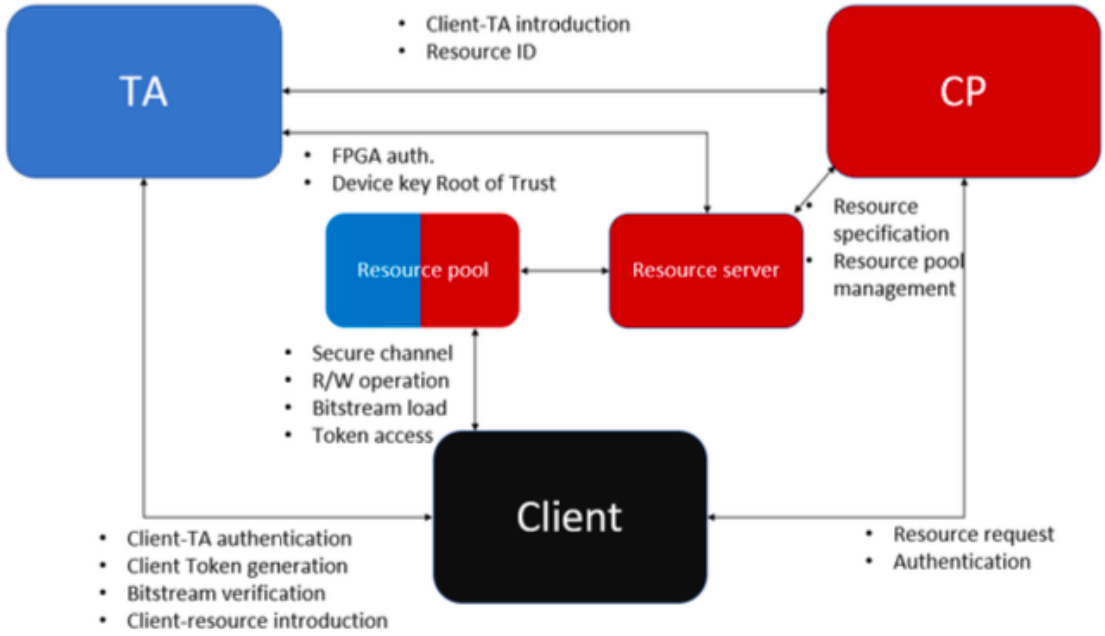

Figure 4: High level architecture

OAuth 2.0 is modified to include FPGA context usage. Due to report size constraint, protocol details will be explained in future technical report. The protocol allows to establish a secured channel between FPGA instance and client. After the token issuance, the client contacts the FPGA to earn access for resources he has been authorized. A transport layer security (TLS) session is set up for secure communication with perfect forward secrecy between the FPGA and the client. The client and the FPGA create their shared secret with algorithms like DHE, ECDHE and then use symmetric encryption algorithms like AES-256-GCM. Once the TLS connection is established, the client sends its token to be authenticated. The FPGA proceeds to token parsing and gives access to the resources the client is authorized to. Further communications between the client and the FPGA will be encrypted. User privacy will be greatly enhanced and isolation from other entities will be achieved.

Modified OAuth 2.0 protocol will be detailed in a future technical report. Our solution enables client benefits from a low-latency single-sign-on authentication for its FPGA thanks to tokenized access. Security and privacy are enhanced for both the CP and the client. Our future work will be focused on the performance evaluation of the proposed solution.

3.6 Accelerator Virtualization

It is needed to ensure that FPGAs and other accelerators can be used with VIM and MANO. We will first investigate how mobile network functions can be entirely or partially deployed to accelerators with a VIM. Kubernetes on top of Docker will be used as a VIM to run a completely disaggregated OpenAirInterface (OAI) RAN on a variety of computing platforms including x86 and ARM servers, FPGAs for baseband acceleration and DPUs. Experiments will help to find potential limitations of

the VIM. This will allow to design improvements and features to fully enable virtualization in the COREnext architecture.

The computing platforms and accelerators that are expected to be used are not yet all supported by OAI. Therefore, some additional work will be done before and in parallel with the tests on the VIM to enable the use of all the expected devices by OAI. Some components may be integrated by the contributors developing these components. It is planned first of all to enable the use of some commercial components like ARM servers, third-party accelerators based on FPGAs and ARM-based DPUs. This will pave the way for components developed within COREnext that are not available for now but may be integrated later within WP6.

Work within WP4 will achieve the design of virtualization for the COREnext architecture which will be then further experimented on top of real COREnext architecture components in WP6.

3.7 M³ – Microkernel-Based System for Heterogeneous Many-Cores

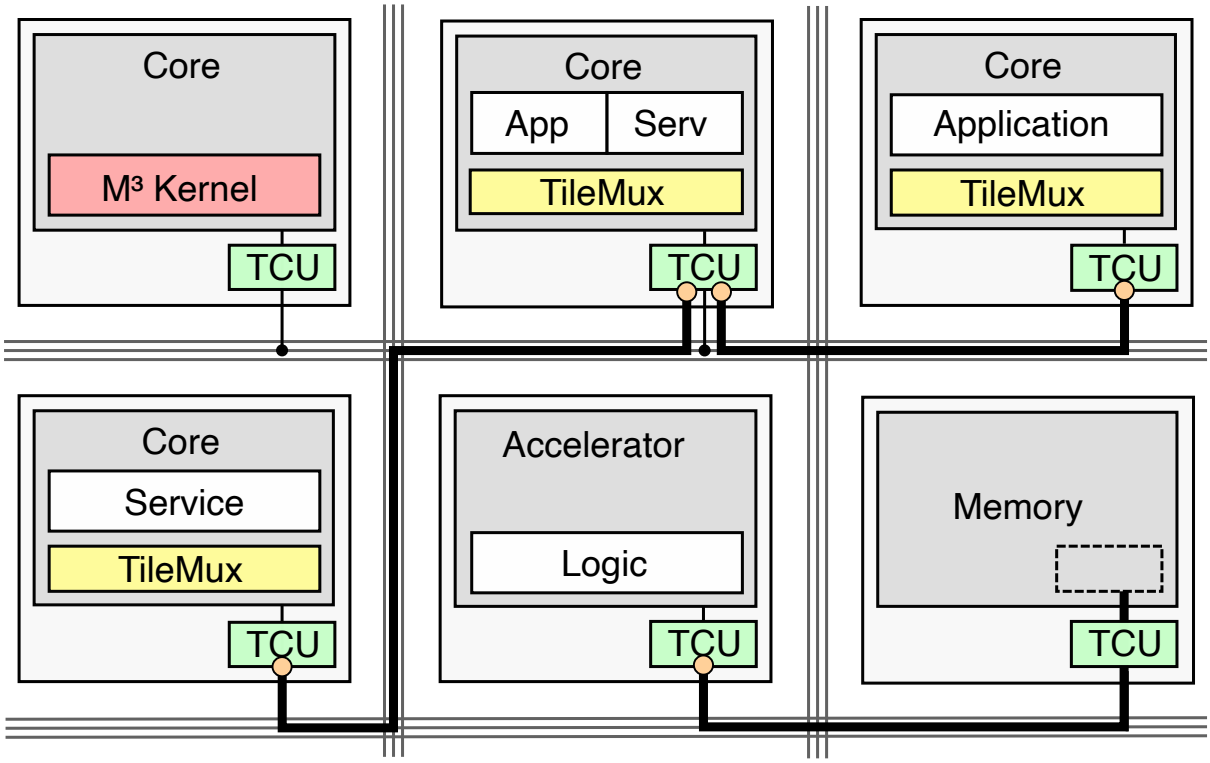

Figure 5: M³ system architecture

M³ [21] proposes a new system architecture based on a hardware/software co-design. On the hardware side, M³ builds upon a tiled architecture, as shown in Figure 5. M³ extends its tiles by adding a new hardware component called trusted communication unit (TCU, also called data transfer unit or DTU in earlier publications). Each tile contains a TCU and either a core, an accelerator, or memory via a memory interface to off-chip dynamic random-access memory (DRAM). The tiles are connected via a network-on-chip. As the TCU is the only way to access tile-external resources, the TCU controls the tile's access permissions. By default, all tiles are isolated from each other. To perform

message-passing between tiles or access memory, a corresponding communication channel (thick black lines in the figure) needs to be established. These communication channels are represented as endpoints in the TCU (orange dots).

On the software side, M³ runs a microkernel (red) on a dedicated kernel tile, and applications and system services on the remaining user tiles. Applications and system services are represented as activities, comparable to processes. An activity on a general-purpose tile executes code, whereas an activity on an accelerator tile uses the accelerator's logic. Activities can use existing communication channels, but only the M³ kernel is allowed to establish such channels. Management of communication rights is based on a capability system, which the microkernel implements. Capability-based permission management fosters the use of strong security policies such as the principle of least authority (POLA).

Applications are placed on different tiles by default, but as shown by M3v [22], tiles with general-purpose cores can also be shared efficiently and securely among multiple applications. For that reason, every core-based user tile runs a multiplexer called TileMux (yellow), which is responsible for isolating and scheduling the applications on its own tile, like a traditional microkernel. However, in contrast to a kernel, each TileMux instance has no permissions beyond its own tile. Instead, only the M³ microkernel can make system-wide decisions.

M³ has been shown to be an excellent platform for the integration of heterogeneous accelerators with computation running on general-purpose cores [23]. Because of the TCU, accelerators are first-class citizens in the system and can interact directly with each other and system services in a data-flow fashion. Traditional platforms require coordination of such flows from a central general-purpose core, which does not scale because the central core quickly becomes a bottleneck. As a result, accelerator integration in M³ is more secure, more scalable, and more energy-efficient compared to traditional system architectures.

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. Although an initial concept exists [24], the problem of TEEs with accelerator integration is currently unsolved in M³ as well as in off-the-shelf hardware architectures. Adding such support to M³ is a major development goal within this project and we believe it will greatly strengthen the position of M³ as an integrative system-level solution for trustworthiness in COREnext.

3.8 IoT Management

The IoT has revolutionized the way we interact with technology, transforming everyday objects into smart devices interconnected through the internet. IoT management and devices are now pervasive in our homes, workplaces, and public spaces, providing us with convenience and efficiency. From smart environment monitoring to connected cars and industrial sensors, IoT devices have become an integral part of our modern lives. However, as the IoT landscape continues to expand, so does the importance of ensuring the trustworthiness of these devices (e.g., are they reliable;

secure; ensure privacy of IoT devices and the data they collect and transmit). With these devices becoming increasingly integrated into critical systems and handling sensitive data, their trustworthiness has a direct impact on user safety, data security, and overall system integrity.

In case IoT devices are compromised the impact can be severe. A breach in the security of an IoT device could lead to unauthorized access to personal information, potential cyberattacks on connected infrastructure, or even endanger lives if safety-critical systems are compromised. Furthermore, data breaches from untrustworthy IoT devices can erode consumer confidence, leading to reluctance in adopting new technologies and hindering the widespread adoption of the IoT. In this context, ensuring the trustworthiness of IoT devices becomes imperative for all stakeholders. By implementing stringent security measures, promoting privacy by design, and fostering a culture of continuous improvement, we can create a more resilient and secure IoT ecosystem.

To ensure trustworthiness in IoT it would be essential to have strong **authentication** and access control by implementing robust authentication mechanisms to ensure that only authorized users or devices can access the IoT system (e.g., utilizing strong passwords, two-factor authentication, and encryption protocols). Also, it is important to have regular **updates** of the firmware and software of IoT devices to fix security vulnerabilities and bugs by providing an easy and automated update process to encourage users to stay current. In addition, the usage of **secure communication** protocols, such as secure sockets layer (SSL)/TLS, to encrypt data transmission between IoT devices and backend systems is important to avoid using default or weak encryption methods. Furthermore, the **encryption** of sensitive data, both during transit and storage, will prevent unauthorized access and ensure data remains confidential, while at the same time it is important to check that the **physical access** to the IoT devices is adequately protected by having proper physical security measures that can prevent unauthorized access.

Other aspects to be considered in trustworthy IoT management and devices can be standardized data formats (XML, JSON, etc.); Flexibility to different communication protocols/patterns, adjusting to varying conditions, e.g., request/response (HTTP(/2), RPC, etc.), publish/subscribe, push/pull (WebSocket, etc.); Multi-connectivity capabilities and Trusted computing TEEs (virtualized, cloud/edge-based).

Trustworthiness in IoT management is important for several reasons. Firstly, it directly impacts user safety and data security. With IoT devices embedded in critical infrastructure, such as healthcare, transportation, and industrial systems, any compromise in their security can have disastrous consequences. Trustworthy devices minimize the risk of cyberattacks, unauthorized access, and data breaches, safeguarding both individuals and organizations from potential harm. Furthermore, trustworthiness paves the way for innovation and progress. In an ecosystem built on trust, businesses can collaborate more effectively, driving collective advancements in IoT technology.

3.9 AI Inference for Radio Link Authentication

This section explores the use of machine learning (ML) technique for RF fingerprinting. The technique 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 hardware impairments can manifest in imperfections such as quadrature imbalance, phase noise, frequency jitter, power amplifier (PA) in-band distortion, intermodulation distortion and reference spurs.

Before the wide spread of ML, measurable properties of the radio signals were extracted using traditional techniques which relied heavily on manual feature engineering and statistical methods. These methods faced limitations such as lack of scalability with increasing data complexity, difficulty in adapting to new situations or changes in the signal environment, and a reliance on extensive domain expertise for feature selection. The rise of ML, however, has revolutionized the field of RF fingerprinting. ML algorithms, especially deep learning models, are capable of recognizing patterns in the 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. These variations may include changes in signal strength due to distance or obstruction, multipath effects, or device interference. Therefore, the ML models can be trained to identify and classify devices based on their RF fingerprints.

The task 4.3 within WP4 aims to develop acceleration solution(s) based on the algorithm-hardware co-design for RF fingerprinting to establish the trustworthiness of a device identity before authorizing any data exchange over a radio link. This task can be broken down into the following subtasks:

1. Develop a lightweight ML algorithm for RF fingerprinting, which would involve researching, designing, and implementing an ML model that is efficient enough to run on resource-constrained devices, but also powerful enough to accurately identify unique RF fingerprints.
2. Algorithm-hardware co-design for implementation to optimize the overall performance, energy efficiency, and cost, which involves implementing the ML model on the hardware and optimizing the implementation to make full use of the hardware resources. This second subtask also involves establishing a trusted connection from the fingerprint signal extraction at the RF frontend to the ML accelerator executing the inference step.

The project officially commenced in April 2023, and we are working closely with the WP5 partners which are primarily involved with the simulation of RF fingerprints data to train ML algorithms.

We began by reviewing a vast number of state-of-the-art publications about this topic to understand their achievements and limitations, challenges in the field, and various use cases. Consequently, we have started to construct a framework that takes raw in-phase and quadrature (IQ) signals data from several simulated transmitter devices and performs device classification using a Convolutional Neural Network (CNN) architecture. We have used the work referenced in [25] as the foundation for this framework.

As an initial test, we ran the training and inference stages using straightforward data from four different transmitter configurations, featuring four different power amplifier models based on measurements and/or post-layout simulations. These configurations exhibit easily distinguishable RF fingerprints, and as expected, we achieved 100% device classification accuracy.

We are now delving deeper into the subtle differences between input configurations. These configurations are generated by using different polynomials to represent the dependency of PA gain

on input power, which represents the amplifier's non-linear behaviour at high input power levels. The goal is to improve our understanding and modelling of these differences in RF fingerprints.

4 Summary

In this deliverable, we proposed component properties derived from the architecture in D3.1, the use cases in D2.1 and the overall project goals. We plan to build power-efficient processing with RISC-V-based accelerators, i.e., a many-core accelerator, a vector processor, an FEC accelerator and a MAC scheduling accelerator. Components which isolate and orchestrate for trustworthiness were proposed on different levels: among a single optionally FPGA-based accelerator shared by multiple tenants, among multiple accelerators and processors, among IoT devices, and in the network by means of AI inference for radio link authentication.

In the future, we want to address the highlighted research challenges. The progress will be reported in D4.2 for the power-efficient signal processing and for the trustworthy computation and orchestration in D4.3, respectively. There is also a parallel development effort for the analogue component clusters in WP5. Both work packages will report their insights back to WP3 which will update the architecture accordingly in D3.2.

5 References

[1] Qualcomm Technologies, "Leading Semiconductor Industry Players Join Forces to Accelerate RISC-V," 04 August 2023. [Online]. Available: https://www.qualcomm.com/news/releases/2023/08/leading-semiconductor-industry-players-join-forces-to-accelerate.

[2] L. M. P. Larsen, A. Checko and H. L. Christiansen, "A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks," *IEEE Communications Surveys & Tutorials,* vol. 21, no. 1, pp. 146-172, 2019.

[3] H. B. Amor, C. Bernier and Z. Přikryl, "A RISC-V ISA Extension for Ultra-Low Power IoT Wireless Signal Processing," *IEEE Transactions on Computers,* vol. 71, no. 4, pp. 766-778, 2022.

[4] 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.

[5] 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, 2022.

[6] 3GPP TR 38.211, "NR; Physical channels and modulation".

[7] 3GPP TR 38.321, "NR; Medium Access Control (MAC) protocol specification".

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

[9] V. Razilov, J. Zhong, E. Matúš and G. Fettweis, "Dual Vector Load for Improved Pipelining in Vector Processors," in *IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)*, Tokyo, Japan, 2023.

[10] A. R. Buzdar, L. Sun, R. Kashif, M. W. Azhar and M. I. Khan, "Cyclic Redundancy Checking (CRC) Accelerator for Embedded Processor Datapaths," *International Journal of Advanced Computer Science and Applications,* vol. 8, no. 2, pp. 321-325, 2017.

[11] A. Karakuzulu, W. A. Ahmad, D. Kissinger and A. Malignaggi, "A Four-Channel Bidirectional D-Band Phased-Array Transceiver for 200 Gb/s 6G Wireless Communications in a 130-nm BiCMOS Technology," *IEEE Journal of Solid-State Circuits,* vol. 58, no. 5, pp. 1310-1322, 2023.

[12] A. Hasani, High-Throughput QC-LDPC Codes for Next-Generation Wireless Communication Systems, Dissertation, BTU Cottbus-Senftenberg, 2021.

[13] A. Verma and R. Shrestha, "Low Computational-Complexity SOMS-Algorithm and High-Throughput Decoder Architecture for QC-LDPC Codes," *IEEE Transactions on Vehicular Technology,* vol. 72, no. 1, pp. 66-80, 2023.

[14] A. Hasani, L. Lopacinski, G. Panic and E. Grass, "550 Gbps Fully Parallel Fully Unrolled LDPC Decoder in 28 nm CMOS Technology," in *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, Grenoble, France , 2022.

[15] H. Englund and N. Lindskog, "Secure acceleration on cloud-based FPGAs – FPGA enclaves," in *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, New Orleans, LA, USA, 2020.

[16] M. E. S. Elrabaa, M. Al-Asli and M. Abu-Amara, "Secure Computing Enclaves Using FPGAs," *IEEE Transactions on Dependable and Secure Computing,* vol. 18, no. 2, pp. 593-604, 2021.

[17] O. Knodel, P. R. Genssler, F. Erxlben and R. G. Spallek, "FPGAs and the Cloud – An Endless Tale of Virtualization, Elasticity and Efficiency," *International Journal on Advances in Systems and Measurements,* vol. 11, no. 3, pp. 230-249, 2018.

[18] K. Eguro and R. Venkatesan, "FPGAs for trusted cloud computing," in *22nd International Conference on Field Programmable Logic and Applications (FPL)*, Oslo, Norway, 2012.

[19] B. Hong, H.-Y. Kim, M. Kim, T. Suh, L. Xu and W. Shi, "FASTEN: An FPGA-Based Secure System for Big Data Processing," *IEEE Design & Test,* vol. 35, no. 1, p. 30.38, 2018.

[20] D. Hardt, "The OAuth 2.0 Authorization Framework," RFC 6749, 2012.

[21] 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.

[22] N. Asmussen, S. Haas, C. Weinhold, T. Miemietz and M. Roitzsch, "Efficient and Scalable Core Multiplexing with M³v," in *27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Lausanne, Switzerland, 2022.

[23] 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.

[24] C. Weinhold, N. Asmussen, D. Göhringer and M. Roitzsch, "Towards Modular Trusted Execution Environments," in *6th Workshop on System Software for Trusted Execution (SysTEX)*, Rome, Italy, 2023.

[25] H. Li, K. Gupta, C. Wang, N. Ghose and B. Wang, "RadioNet: Robust Deep-Learning Based Radio Fingerprinting," in *IEEE Conference on Communications and Network Security (CNS)*, Austin, TX, USA, 2022.