A Trust-minimized Platform for Secure AI Accelerator Integration

Friedrich Pauls, Sebastian Haas, Yogesh Verma
Barkhausen Institut, Dresden, Germany
forename.surname@barkhauseninstitut.org

***Abstract*—We present a secure tiled MPSoC architecture enabling trust-minimized integration of third-party AI accelerators. Leveraging a microkernel operating system and hardware-enforced isolation via Trusted Communication Units, our adapted Accelerator Support Module securely encapsulates AI cores with modest area overhead. Implementation results in 22 nm FDSoI demonstrate effective isolation and low resource usage, enabling privacy-preserving, on-premise AI computing.**

***Index Terms*—Hardware/Software Co-Design, Isolation, Network-on-Chip, Operating System, Privacy, Security, Tiled Architecture**

I. Introduction

Imagine a voice assistant that lives entirely on your personal home server. It transcribes your speech, understands your intent, and carries out commands, all without ever sending audio to the cloud. You say: "Add a calendar entry for tomorrow at 10am." The assistant transcribes the audio using one AI module, classifies the command using a second, and finally routes the result to a calendar service. Each step is performed by a separate AI agent, each confined to a minimal security domain. No single module has access to the full audio, full transcript, and control logic at once. The result: secure functionality without surveillance.

This scenario captures a broader vision: local, privacy-preserving AI computing. From document tagging to smart home management and photo classification, many everyday AI tasks can be done on–premise, but only if the hardware platform supports strong, default isolation between modules. Most accelerators today are black-box IP cores with unknown trustworthiness. Once integrated into a system-on-chip (SoC), they pose significant risks if not properly contained.

This paper explores a solution based on a secure tiled MPSoC platform designed to integrate third-party accelerators, such as AI cores, without compromising system integrity. The platform builds on the M^3 architecture, which combines a tiled hardware design with a microkernel operating system (OS) and hardware-enforced communication via *Trusted Communication Units* (TCUs). We extend this architecture with an adapted *Accelerator Support Module* (ASM), a tile wrapper that allows secure integration of untrusted AI accelerators.

Consider the example shown in Figure 1 (a): A privacy-preserving voice assistant implemented using two AI accelerators: one for speech-to-text and another for intent classification. Each accelerator resides in its own tile. The voice-to-text tile has access only to microphone data and produces annotated text, which is passed via an OS-mediated and TCU-enforced channel to the intent-classifier tile. This model ensures that neither module sees more data than needed, and prevents lateral movement in case of compromise.

While related NoC-centric security architectures like SiFive Shield use memory protection units and message passing to create a tiled security architecture [1], M^3 uses the principle of *isolation-by-default*, each AI accelerator can be confined to its own tile, communicating only through explicit OS-managed channels. Even when multiple accelerators collaborate, as in the voice assistant example, each operates with strictly bounded permissions. This drastically reduces the system's trusted computing base and limits the impact of a compromised component.

Section I-A introduces the architecture and isolation concept, Section II details the AI accelerator integration, followed by implementation results from our 11-tile prototype including an ASM with AI accelerator in Section III.

A. The M^3 Architecture and Isolation-by-Default

The M^3 platform is a hardware/software co-design that targets secure, flexible integration of heterogeneous compute components, including general-purpose processors and specialized accelerators. It is based on a tiled manycore SoC architecture [2]. Each tile encapsulates a processing element, local memory, and a dedicated TCU [3]. These tiles are interconnected via a network-on-chip (NoC), and the overall system is orchestrated by a microkernel operating system (OS) running on a designated kernel tile.

The key security principle of M^3 is *isolation-by-default*. In contrast to conventional MPSoCs where cores share memory and buses by default, in M^3 no tile can communicate with any other unless explicitly permitted. This is enforced at the hardware level by TCUs. Each TCU mediates all memory and message-passing interactions and ensures that only authorized channels, established by the OS, are active. The OS thus becomes the sole entity responsible for setting up secure interactions, while the TCUs enforce these decisions in hardware.

This model offers two major benefits: (1) it drastically reduces the *trusted computing base* (TCB) since only the kernel tile, the OS, the NoC, TCUs, and *used* accelerators must be trusted; and (2) it limits the blast radius of compromised components. Even if an application, accelerator, or driver is

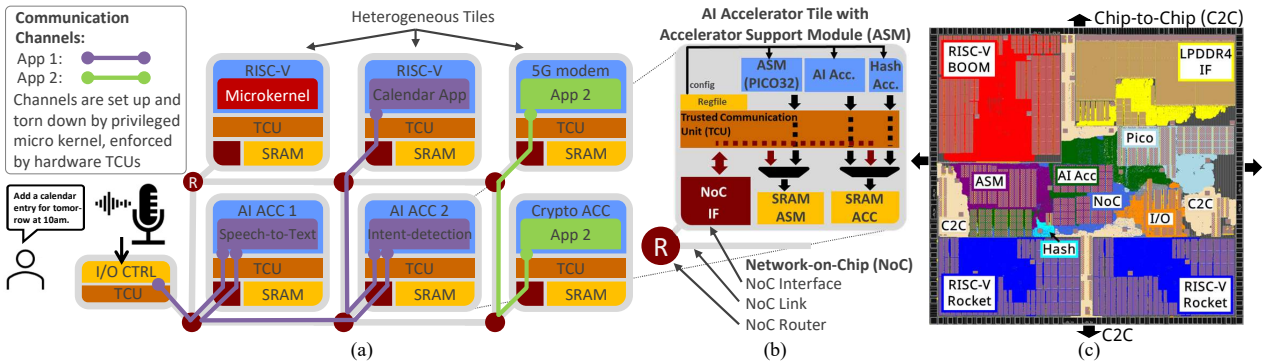

Fig. 1: (a) General architecture secure HW/OS platform and use case. (b) ASM with AI accelerator. (c) Post-routing layout.

malicious or vulnerable, it remains confined to its own tile and cannot interact with other parts of the system.

Moreover, the M^3 supports dynamic configuration and revocation of communication channels. This makes the system particularly suitable for multi-tenant or multi-agent AI workloads, where isolated services need to interact under tightly controlled policies.

II. Secure AI Accelerator Integration using ASM

One of the key challenges in building such a secure and flexible platform is the integration of third-party or vendor-specific hardware accelerators, particularly AI cores, as these often lack standardized interfaces and do not conform to the communication and memory protocols required by M^3. Furthermore, their internal logic is typically opaque, making it difficult to evaluate their trustworthiness or apply formal verification.

To address this, we use an adapted ASM, as depicted in Figure 1 (b), a dedicated tile architecture designed to securely integrate such accelerators. Each ASM tile consists of three main components: (1) a lightweight RISC-V core (e.g., PicoRV32) acting as a protocol handler between M^3 and the accelerator, (2) a TCU and (3) the AI accelerator. To limit overhead, the ASM can also host additional accelerators, e.g. for hashing. The RISC-V core controls the accelerators and interacts with the M^3 OS, while the TCU enforces access control and communication boundaries.

We integrated a 32 GOPS 8-bit fixed-point AI accelerator from [4] featuring a 64-MAC array, vector units for activation and pooling, and support for convolutional and pooling operators, with 16kB local SRAM, 4x64bit data interfaces to 128kB tile SRAM and a 32bit config interface.

The ASM wraps the accelerators in a hardware-managed security envelope. It allows only memory-mapped access to designated SRAM regions and exposes a narrow configuration interface. All external communication—whether for data transfer, command issuance, or result propagation—is regulated by the TCU and must be authorized by the OS.

Tile	#/chip	Area	SRAM area		TCU area		Memory config
		[mm^2]	[mm^2]	[%]	[mm^2]	[%]*	[kB]
Boom	1	1.60	1.02	64	0.10	6.2	L2: 256, I+D: 2x16
Rocket	2	1.14	0.77	68	0.10	8.7	L2: 256, I+D: 2x16
ASM	1	0.72	0.57	79	0.10	7.7	128
+ AI		0.51	0.23	44			128
+ Hash		0.03	0.01	44			4
DDR-IF	1	1.99	0.17	8	0.01	0.4	
I/O	1	0.27	0.20	76	0.09	33.4	64
Chip-2-Chip	4	0.19	0.0	36	0.01	3.1	
Pico	1	0.43	0.30	68	0.04	9.8	128
Chip Total:		10.28	5.56	54			

* TCU area relative to total tile area. For ASM, includes AI and Keccak accelerators.

TABLE I: Tile areas, memory configuration and TCU overhead.

III. Implementation and Results

To evaluate the ASM and TCU area overhead, we implemented and synthesized an instance of an M^3 architecture in a 22 nm FDSoI GlobalFoundries process, typical conditions (25 °C, 0.8 V). The architecture consists of 11 tiles with respective areas, as shown in Table I, and a 2x2 star-mesh NoC (bandwidth 16 bytes/cycle). All tiles are secured by a TCU. Depending on tile needs, TCUs are available with different feature sets (e.g. memory protection), leading to varying sizes. Fig. 1 (c) shows the post-routing layout of our implementation. The total chip area is 10.28 mm^2 from which 5.56 mm^2 (54%) is SRAM. The overhead area for the TCUs is below 10% for the accelerators.

IV. Conclusion

We demonstrated that our ASM design enables secure, low-overhead integration of AI accelerators in the M^3 platform. By leveraging isolation-by-default and hardware-enforced access control, the architecture can supports privacy-preserving, multi-agent AI workloads with minimal trust assumptions.

Acknowledgment

This research was funded by the German Federal Ministry of Education and Research (BMBF), funding number 16ME0527.

References

[1] J. Prior, "SiFive Shield: An Open, Scalable Platform Architecture for Security," 2019. [Online]. Available: https://www.sifive.com/blog/sifive-shield-an-open-scalable-platform-architecture

[2] G. Fettweis *et al.*, "A Low-Power Scalable Signal Processing Chip Platform for 5G and Beyond - Kachel," in *53rd Asilomar Conference on Signals, Systems, and Computers*, 2019.

[3] S. Haas *et al.*, "A trusted communication unit for secure tiled hardware architectures," in *2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2022, pp. 1–4.

[4] P. Bernardo *et al.*, "Compiler-aware ai hardware design for edge devices," in *Proc. 8th Int. Workshop on Edge Systems, Analytics and Networking*. New York, NY, USA: Association for Computing Machinery, 2025.