

# 2025 *Activity Report*

RESEARCH CENTRE: Inria Centre at Université Grenoble Alpes

IN PARTNERSHIP WITH: Université de Grenoble Alpes, Institut polytechnique de Grenoble, CNRS

---

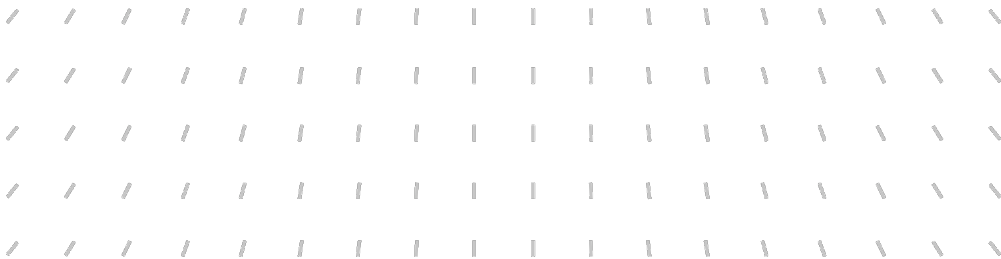
Project-Team

## KRAKOS

Design of performance, robust, secure, flexible, and energy-efficient system software

---

*In collaboration with* Laboratoire d'Informatique de Grenoble (LIG)



## **Project-Team KRAKOS**

*Creation of the Project-Team: 2024 October 01*

Each year, Inria research teams publish an Activity Report presenting their work and results over the reporting period. These reports follow a common structure, with some optional sections depending on the specific team. They typically begin by outlining the overall objectives and research programme, including the main research themes, goals, and methodological approaches. They also describe the application domains targeted by the team, highlighting the scientific or societal contexts in which their work is situated. The reports then present the highlights of the year, covering major scientific achievements, software developments, or teaching contributions. When relevant, they include sections on software, platforms, and open data, detailing the tools developed and how they are shared. A substantial part is dedicated to new results, where scientific contributions are described in detail, often with subsections specifying participants and associated keywords. Finally, the Activity Report addresses funding, contracts, partnerships, and collaborations at various levels, from industrial agreements to international cooperations. It also covers dissemination and teaching activities, such as participation in scientific events, outreach, and supervision. The document concludes with a presentation of scientific production, including major publications and those produced during the year.

## Keywords

### Computer sciences and digital sciences

- A1.1.1. – Multicore, Manycore
- A1.1.9. – Fault tolerant systems
- A1.1.10. – Reconfigurable architectures
- A1.1.13. – Virtualization
- A1.3. – Distributed Systems
- A2.2.3. – Memory management
- A2.2.4. – Parallel architectures
- A2.2.5. – Run-time systems
- A2.6. – Infrastructure software

### Other research topics and application domains

- B6.1. – Software industry
- B6.5. – Information systems
- B6.6. – Embedded systems
- B6.7. – Computer Industry (hardware, equipments...)

## Contents

<b>Project-Team KRAKOS</b>	<b>1</b>
<b>1 Team members, visitors, external collaborators</b>	<b>5</b>
<b>2 Overall objectives</b>	<b>7</b>
2.1 Presentation . . . . .	7
2.2 Objectives . . . . .	7
<b>3 Research program</b>	<b>8</b>
3.1 Methodology . . . . .	8
3.1.1 M1 - Virtualization . . . . .	8
3.1.2 M2 - Profiling, Tracing and Monitoring . . . . .	8
3.2 Research Axes . . . . .	8
3.2.1 A1 - Machine Virtualization . . . . .	8
3.2.2 A2 - Mutant Kernels and Key Abstractions for Concurrency . . . . .	9
3.2.3 A3 - Disaggregation . . . . .	11
3.2.4 A4 - Fault Tolerance . . . . .	12
<b>4 Application domains</b>	<b>14</b>
4.1 Overview . . . . .	14
4.2 Hypervisor Layer . . . . .	14
4.2.1 Target Hypervisors . . . . .	14
4.2.2 Cloud Deployment Models . . . . .	14
4.2.3 Cloud Service Models . . . . .	14
4.3 Operating System Layer . . . . .	15
4.4 Middleware and Orchestration . . . . .	15
4.5 Domain-Specific Applications . . . . .	15
4.5.1 Genomics and Bioinformatics . . . . .	15
4.5.2 Memory-Intensive Applications . . . . .	15
4.5.3 Microservices Architectures . . . . .	15
4.6 Cross-Cutting Application Characteristics . . . . .	16
<b>5 Social and environmental responsibility</b>	<b>16</b>
5.1 Energy Efficiency and Green Computing . . . . .	16
5.1.1 Participation in Standardization Initiatives . . . . .	16
5.2 Diversity, Equity, and Inclusion . . . . .	16
5.2.1 Leadership in DEI Initiatives . . . . .	16
5.2.2 Gender Diversity Reflection . . . . .	17
5.3 Open Science and Reproducible Research . . . . .	17
<b>6 Highlights of the year</b>	<b>17</b>
6.1 Team Creation and Inauguration . . . . .	17
6.2 HDR and PhD Defenses . . . . .	17
6.3 Major Publications . . . . .	17
6.4 Awards and Recognition . . . . .	18
6.5 International Collaborations . . . . .	18
6.6 Conference Organization and Leadership . . . . .	18
6.7 Industrial Partnerships . . . . .	18

<b>7</b>	<b>Latest software developments, platforms, open data</b>	<b>19</b>
7.1	New Software . . . . .	19
7.1.1	USM . . . . .	19
7.1.2	MigCheck . . . . .	19
7.1.3	vPIM . . . . .	19
7.1.4	Faho . . . . .	19
7.1.5	GoodKit . . . . .	19
7.1.6	B-Side . . . . .	20
7.1.7	P4Cemaker . . . . .	20
7.1.8	DirtBuster . . . . .	20
7.2	New platforms . . . . .	20
7.2.1	IBARA - Portable Micro-Cluster for Africa . . . . .	20
7.2.2	Grid'5000 . . . . .	20
7.2.3	SLICES-FR . . . . .	21
<b>8</b>	<b>New results</b>	<b>21</b>
8.1	Physical Memory Management in Userspace (USM) . . . . .	21
8.2	Processing-in-Memory Virtualization . . . . .	21
8.3	Heterogeneous VM Migration . . . . .	21
8.4	Understanding Intel User Interrupts . . . . .	22
8.5	System Call Identification for Security . . . . .	22
8.6	SIMBox Fraud Detection . . . . .	22
8.7	P4Cemaker . . . . .	22
8.8	Pre-Stores . . . . .	23
8.9	Carbon Footprint of Storage in Datacenters . . . . .	23
8.9.1	IBARA - Portable Micro-Cluster for teaching . . . . .	23
<b>9</b>	<b>Bilateral contracts and grants with industry</b>	<b>24</b>
<b>10</b>	<b>Partnerships and cooperations</b>	<b>24</b>
10.1	International Initiatives . . . . .	24
10.1.1	Associated Teams . . . . .	24
10.1.2	Other International Collaborations . . . . .	24
10.2	National Initiatives . . . . .	25
10.2.1	ANR Projects . . . . .	25
10.2.2	PEPR Projects . . . . .	25
10.2.3	Inria Challenges (Défis) . . . . .	25
10.2.4	Regional Projects . . . . .	25
10.3	Collaboration with Other Research Teams . . . . .	26
10.4	Conference and Workshop Organization . . . . .	26
<b>11</b>	<b>Dissemination</b>	<b>26</b>
11.1	Invited Talks . . . . .	26
11.2	Scientific Expertise . . . . .	26
11.3	Research Administration . . . . .	27
11.4	Teaching - Supervision - Juries . . . . .	27
11.4.1	Teaching . . . . .	27
11.4.2	Supervision . . . . .	27
<b>12</b>	<b>Scientific production</b>	<b>27</b>
12.1	Publications of the year . . . . .	27
12.2	Cited publications . . . . .	28

# 1 Team members, visitors, external collaborators

## Research Scientist

- Baptiste Lepers [INRIA, Advanced Research Position, HDR]

## Faculty Members

- Alain Tchana [Team leader, GRENOBLE INP, Professor]
- Noel De Palma [UGA, Professor]
- Fabienne Dechamboux [UGA, Professor]
- Renaud Lachaize [UGA, Associate Professor]
- Vania Marangozova [UGA, Professor]
- Nicolas Palix [UGA, Associate Professor]
- Thomas Ropars [UGA, Associate Professor]

## Post-Doctoral Fellows

- Celestin Bessala Bessala [FLORALIS, Post-Doctoral Fellow, from Jul 2025 until Oct 2025]
- Kenta Ishiguro [GRENOBLE INP, Post-Doctoral Fellow, from Mar 2025]
- Daniel Ndjodo Bessala [UGA, Post-Doctoral Fellow, from Aug 2025]

## PhD Students

- Ivane Adam [UGA]
- Paul Breuil [ENSMP]
- Fonyuy-Asheri Caleb [INRIA]
- Maxime Collette [INRIA, from May 2025]
- Ifechukwu Ejiofor [UGA, from Oct 2025]
- Papa Assane Fall [INRIA, from Oct 2025 until Nov 2025]
- Jordan Gounou Fondjo [GRENOBLE INP]
- Gabriel Job Antunes Grabher [UGA]
- Yves Kone [TOULOUSE INP]
- Jean-Luc Mahop Ma Ngos [UGA]
- Gregoire Mugnier [UGA, until Jun 2025]
- Arnel Nguetoum Mewoupea [UGA, from Apr 2025]
- Yannick Nzali Koagne [UGA]
- Arnold Okala Nanga [ORANGE, CIFRE]
- Damase Onana [Vates]
- Benjamin Priour [HUAWEI, CIFRE, from Sep 2025]
- Brice Tegua Wakam [ORANGE]

## Technical Staff

- Louis Duval [GRENOBLE INP, Engineer, from Apr 2025]
- Andre Freyssinet [UGA, Engineer]
- Franck Kamokoue Sikati [GRENOBLE INP, from Nov 2025]
- Tony Kwenkeu [GRENOBLE INP, Engineer, from Oct 2025]
- Armel Nguetoum Mewoupea [UGA, Engineer, until Mar 2025]
- Albin Petit [INRIA, Engineer]
- Jules Seban [INRIA, Engineer, from Dec 2025]
- Remi Segretain [UGA, Engineer]

## Interns and Apprentices

- Elouan Barraud [UGA, Intern, from Feb 2025 until May 2025]
- Merveille Biada Tchuisseu [GRENOBLE INP, Intern, from Oct 2025]
- Maxime Bodart [GRENOBLE INP, Intern, until Jul 2025]
- Julien Brelot [GRENOBLE INP, Intern, from Mar 2025 until Jul 2025]
- Marie-Line Da Costa Bento [INRIA, Intern, from Jun 2025 until Sep 2025]
- Greg Depoire–Ferrer [ENS Lyon, from Feb 2025]
- Kevin Efremov [GRENOBLE INP, Intern, until Aug 2025]
- Ifechukwu Ejiofor [FLORALIS, Intern, from Feb 2025 until Jun 2025]
- Thomas Fourier [INRIA, Intern, from Apr 2025 until Sep 2025]
- Kimia Khadamlou [UGA, Intern]
- Fideline Kuetche [GRENOBLE INP, from Sep 2025]
- Meyo Charlotte Lysana Georgia [INRIA, Intern, from Sep 2025]
- Weihao Ni [INRIA, Intern, from Mar 2025 until Aug 2025]
- Corentin Oparowski [INRIA, Intern, from May 2025 until Aug 2025]
- Jad Salameh [UGA, Intern]
- Jules Seban [GRENOBLE INP, Intern, from Mar 2025 until Aug 2025]
- Franck Tamwo [GRENOBLE INP, Intern, from Sep 2025]
- Yann Brady Tchounkeu Djabou [INRIA, Intern, from Jun 2025 until Aug 2025]
- Niels Terese [INRIA, Intern, until Apr 2025]
- Xiaoxiang (William) Wu [INRIA, Intern, until Apr 2025]
- Yuben Yang [INRIA, Intern, until Jul 2025]
- Alexander Yanovskyy [UGA, Intern, from Feb 2025 until Jul 2025]

## Administrative Assistant

- Annie Simon [INRIA]

## 2 Overall objectives

### 2.1 Presentation

Created on October 1st, 2024, **KrakOS** is the Systems group at Inria Centre at Université Grenoble Alpes. The team name pays homage to Sacha Krakowiak, emeritus professor from Grenoble whose work has significantly influenced the local and international scientific community in operating systems research.

Data centers are an essential pillar of computing infrastructures. They host the vast majority of applications used daily by businesses and individuals, along with associated data. Applications are increasingly diverse and must meet ever-stronger efficiency constraints in terms of responsiveness, data volumes, and energy consumption. To meet these needs, data centers are designed with complex multi-level architectures, characterized by:

- **Large scale:** Number of physical and virtual servers, volumes of internal and external requests
- **Density and resource sharing:** Number of applications cohabiting on each physical server
- **Hardware heterogeneity:** At the server scale and at the data center scale
- **Multiple accelerators:** NVM, GPU, TPU, PIM, FPGA, etc.
- **Extremely advanced microarchitectures:** AMP, NUMA, DDIO, SGX, etc.

System layers (hypervisor, operating system, centralized or distributed runtime) play a critical role due to the control they exercise over both hardware resources and software activities: they directly impact the security, stability, and efficiency of the data center, and therefore the applications it hosts.

Numerous works from the scientific community have highlighted the growing inadequacy between the characteristics of current system layers and those of the data centers described above. Current systems are delicate to maintain, evolve, observe/supervise, optimize, make reliable, and secure, especially as each of these objectives conflicts with the others. Generally, these difficulties lead to under-exploitation of the potential of hardware resources. These inefficiencies are amplified by the significant and growing reduction in time scales for both the latencies of certain hardware resources and the durations of application tasks ("microsecond-scale" computing).

### 2.2 Objectives

The **KrakOS** team aims to revisit the fundamental principles that have governed the construction of system layers until now in order to take into account the modernity of data centers and anticipate future developments. KrakOS targets five main objectives and the inherent trade-offs between them:

1. **Performance**, characterized by application metrics such as execution time, throughput, latency, as well as statistical indicators on the variability of these metrics;
2. **Fault tolerance and high availability**;
3. **Velocity of development, testing, and deployment** (to enable rapid consideration of new requirements);
4. **Expressiveness and flexibility of programming interfaces (APIs)**, to simplify the work of application programmers;
5. **Energy efficiency**. KrakOS aims to achieve the above objectives while maintaining (at minimum) the energy efficiency of systems or improving it.

Like any system research team, KrakOS aims to invent new abstractions, concepts, policies, mechanisms, and techniques. Prototyping and empirical evaluation are the preferred methods to validate our proposed contributions. Theoretical proofs are rarely performed in this domain given the complexity of the studied systems.

## 3 Research program

### 3.1 Methodology

KrakOS has a unique approach in its scientific methodology:

1. Revisit and question the relevance of established solutions in systems (Process and Thread actions, for example);
2. Revisit and question the relevance of solutions that have not succeeded (microkernels, for example);

KrakOS validates its results primarily empirically. For this, the Grid'5000 research platform and its successor SLICES-FR will be our main experimental grounds.

#### 3.1.1 M1 - Virtualization

To achieve the stated objectives, KrakOS relies primarily on virtualization. Virtualization is a fundamental tool at the heart of building computer systems. It enables optimal resource utilization, isolation/security, uniformity in resource access, and facilitates the design of fault tolerance techniques.

We consider virtualization in its original sense, as defined by Sacha Krakowiak: the virtualization of a component is the design of an "ideal" abstraction of that component for other components or users. In this definition, the virtualized component can be a physical component (device, machine or grouping of machines) or software (a machine is a stack of virtual machines that goes from the motherboard to the browser, for example).

#### 3.1.2 M2 - Profiling, Tracing and Monitoring

Empirical observation and therefore observability are at the heart of systems research. They allow identifying and understanding limitations and problems: bottlenecks, sources of inefficiency and resource waste, performance anomalies, bugs and complex failures (at hardware and software levels).

KrakOS aims to contribute to the production of profiling and tracing tools adapted to the modernity of data centers. Among the challenges posed by the latter, we can cite the stack of complex and highly distributed layers. In a virtualized cloud environment, for example, it is extremely difficult to reconstruct the path taken by an I/O request that traverses the virtual machine, host system, network, storage system, and disk, then retraces the path in reverse.

More generally, the evolution of latencies and throughputs of emerging communication and storage devices, coupled with strong quality of service constraints of cloud applications, require new approaches allowing an acceptable and flexible compromise between precision, efficiency, and intrusiveness (code and privacy). Regarding privacy, for example, it is necessary for data center operators to comply with regulations (GDPR, for example). Monitoring tools must be able to trace the I/O activities of virtual machines without observing customer data.

### 3.2 Research Axes

KrakOS will pursue four research axes simultaneously. These axes are deeply interconnected and address the five main objectives of KrakOS: performance, fault tolerance and high availability, velocity of development, API expressiveness and flexibility, and energy efficiency.

#### 3.2.1 A1 - Machine Virtualization

KrakOS investigates fundamental problems in machine virtualization that have become increasingly critical as datacenters evolve toward greater heterogeneity, incorporate diverse hardware accelerators, and face ever more stringent performance requirements. While virtualization has been the cornerstone technology enabling cloud computing for over two decades, the assumptions underlying current virtualization systems—designed for relatively homogeneous server fleets with CPU-centric workloads—are increasingly mismatched with modern datacenter realities. These mismatches manifest as performance bottlenecks, operational challenges, and missed opportunities to leverage emerging hardware capabilities.

The research addresses five interconnected challenges. **VM live migration at scale** has become problematic as datacenter hardware diversifies. Migrating virtual machines across heterogeneous hardware platforms—from older to newer CPU generations, between different vendors’ processors, or to systems with different accelerator configurations—requires maintaining both functional correctness and performance characteristics. Industry has highlighted the severity of this problem: Microsoft has noted that hardware heterogeneity in Azure contributes significantly to resource fragmentation, where incompatibility between hosts prevents optimal VM placement and leads to hundreds of millions of dollars in efficiency losses. **I/O virtualization efficiency** remains a persistent challenge despite decades of research. Current virtualization approaches impose significant performance overhead on I/O-intensive applications due to additional software layers, context switches between guest and host, and memory copies. As storage devices transition to NVMe SSDs and networking speeds reach 100+ Gbps, these overheads become increasingly unacceptable—the challenge is to design virtualization mechanisms that approach bare-metal performance while maintaining the isolation and management benefits that motivate virtualization in the first place.

**Hardware accelerator support** presents difficulties because emerging accelerators like PIM (Processing-in-Memory), GPUs, TPUs, and FPGAs were designed without virtualization in mind. Each accelerator type presents unique challenges: GPUs have complex memory hierarchies and scheduling requirements, PIM devices tightly couple computation with memory access patterns, and FPGAs require load-time configuration that complicates sharing. Virtualizing these devices while maintaining both performance (near-native execution speed) and isolation (preventing one tenant from observing or interfering with another) requires deep co-design of hardware features and virtualization software. **Nested virtualization**—where virtual machines run inside other virtual machines—has long been considered impractical for production use due to performance overhead. However, nested virtualization is increasingly important for scenarios like testing cloud infrastructure, providing cloud-within-cloud services, and enabling sophisticated isolation architectures. Building on recent hardware advances and algorithmic improvements, KrakOS aims to make nested virtualization practical for production deployment. Finally, **security** challenges in virtualized environments continue to evolve as attack surfaces expand and new vulnerability classes emerge. Beyond traditional concerns about hypervisor bugs that could allow guest escape, modern threats include side-channel attacks that exploit shared hardware resources, malware operating within guest VMs that must be detected from outside, and supply-chain attacks targeting virtualization infrastructure itself.

KrakOS addresses these challenges through coordinated research across multiple dimensions. The team designs and implements novel I/O virtualization mechanisms that minimize overhead through techniques such as direct device assignment with enhanced isolation, optimized data paths that reduce memory copies, and hardware-software co-design that leverages emerging virtualization features in I/O devices. For emerging accelerators, KrakOS develops transparent virtualization approaches exemplified by the vPIM project for Processing-in-Memory devices, which provides full virtualization support while maintaining near-native performance and strong isolation guarantees. The team creates migration protocols and feasibility testing tools, such as MigCheck, that can predict whether a VM can successfully migrate to a target host before attempting the migration—preventing failures that cause service disruptions and wasted resources. Security is strengthened through multiple approaches: formal verification techniques that can prove properties about critical hypervisor code, hardware-software co-design that leverages trusted execution environments and memory encryption, and virtual machine introspection frameworks (such as GoodKit) that enable security monitoring from the hypervisor without compromising guest privacy or performance. Throughout this research, KrakOS maintains a strong focus on open-source hypervisors—particularly Xen and KVM—to ensure that innovations can be rapidly adopted in production cloud environments and benefit the broader systems community rather than remaining theoretical contributions.

### 3.2.2 A2 - Mutant Kernels and Key Abstractions for Concurrency

#### Sub-axis 1: Mutant Kernels - Outsourcing OS Services to User Space

KrakOS studies the extensibility of monolithic kernels through a novel approach: outsourcing system services and abstractions from kernel space to user mode. This research direction is inspired by the microkernel philosophy but operates under fundamentally different constraints and opportunities. While traditional microkernels aim for minimalism—reducing the kernel to a small, provably correct core—the mutant kernel approach preserves the rich feature set of monolithic kernels that applications depend on while gaining the flexibility, safety, and evolvability benefits of user-space implementation. This philosophy aligns

with recent industry trends where major systems are being redesigned to support user-space implementations of traditionally kernel-resident services.

Recent work in the systems community demonstrates both the promise and current limitations of this approach. Systems like uFS [13] (file system), Snap [14] (networking stack), and ghOSt [12] (scheduler) have shown that moving individual services to user space can improve flexibility and enable rapid innovation. However, current approaches suffer from three fundamental limitations. First, they consider outsourcing only a single service at a time, neglecting the complex interactions between system services that occur in real kernels. Second, they rely exclusively on classical abstractions like the Process, which provides insufficient nuance to distinguish between ordinary application code and semi-privileged system services that require special treatment—higher scheduling priority, access to privileged resources, and protection from interference by untrusted applications. Third, no existing framework addresses efficient and secure cooperation between multiple outsourced services, despite the fact that services like memory management, scheduling, and I/O must closely coordinate.

KrakOS addresses these limitations through several research directions. The team pursues a holistic study of OS service outsourcing that considers multiple services simultaneously and their necessary interactions. This requires designing new abstractions specifically for system services—abstractions that sit conceptually between ordinary processes and kernel code, with appropriate privileges, protection mechanisms, and scheduling guarantees. The team explores using high-level and provable languages (such as Rust or formally verified subsets of C) for implementing these services, taking advantage of user-space deployment to leverage stronger type systems and verification tools than are practical in kernel development. Security and isolation mechanisms must be carefully adapted to the needs of semi-privileged services, providing protection both from untrusted applications and between mutually distrusting system services. Finally, efficient user-kernel communication interfaces are essential—the performance overhead of crossing protection boundaries must be minimized since system services handle operations at microsecond granularity.

#### **Sub-axis 2: Key Abstractions for Concurrency and Isolation**

The fundamental abstractions that developers use to structure concurrent and distributed programs have remained largely unchanged since their introduction in the 1960s and 1970s. The Process abstraction, introduced by Dijkstra in 1965 and subsequently implemented in pioneering systems like MULTICS, provides isolated address spaces and resource management. The Thread abstraction was later derived to accommodate concurrent shared-memory programming within a single address space. For over fifty years, developers have been forced to make a static choice between these two abstractions during application development, a choice with profound implications for performance, scalability, and fault tolerance.

This creates a fundamental dilemma. Multi-threaded applications benefit from efficient communication through shared memory but can only scale to the size of a single machine—they cannot leverage the computational resources of an entire datacenter. Conversely, multi-process applications can scale to datacenter-scale by distributing work across many machines, but suffer from heavy communication overhead since inter-process communication requires expensive serialization, network transmission, and deserialization. The challenge is particularly acute because developers typically lack complete control over where their applications will be deployed—an application designed for a single large machine may later need to scale across multiple machines, or vice versa, but the chosen abstraction is baked into the application’s architecture.

KrakOS’s vision, articulated in [15], proposes a radical rethinking of these fundamental abstractions. Rather than forcing developers to choose between Processes and Threads, KrakOS designs a system interface that facilitates the integration of new abstractions at the same architectural level—abstractions that can provide different trade-offs between isolation, performance, and scalability. This research leverages modern hardware protection mechanisms including Intel SGX (secure enclaves), Intel MPK (memory protection keys for fast domain switching), Arm TrustZone (secure execution environments), and CHERI capabilities (hardware-enforced fine-grained memory protection). By separating three orthogonal concerns—execution flows (units of sequential execution), protection domains (boundaries for isolation and security), and communication mechanisms (how execution flows interact)—the programming interface allows applications to compose these elements in ways that match their specific needs rather than being constrained by the Process/Thread dichotomy.

The research extends beyond API design to system runtime implementation. KrakOS develops runtimes capable of dynamically selecting the most relevant communication and isolation mechanisms for each application based on current deployment context, workload characteristics, and performance requirements. For example, when co-located on a single machine, components might communicate through shared memory;

when distributed, the same application might transparently switch to network communication. This dynamic adaptation requires sophisticated runtime support that can make these decisions efficiently and transparently. Finally, the team extends compilers and code generators to enable simplified or even transparent use of these new abstractions, allowing developers to express high-level intent rather than low-level mechanism choices, with the compiler and runtime collaborating to select appropriate implementations.

### 3.2.3 A3 - Disaggregation

The rise of cloud computing, enabled by virtualization technologies, has paradoxically led to server fragmentation—the chronic underutilization of hardware resources within individual servers. While virtualization allows multiple workloads to share a physical machine, the granularity of allocation remains at the server level, leading to situations where some servers have excess CPU capacity while others have unused memory, yet these resources cannot be efficiently shared across server boundaries. Resource disaggregation addresses this fundamental limitation by enabling more flexible allocation of hardware resources at finer granularities. The economic impact is substantial: Microsoft estimates that even a 1% reduction in fragmentation within its Azure cloud platform would generate savings of hundreds of millions of dollars annually [11], highlighting both the scale of the problem and the potential impact of effective solutions.

KrakOS pursues research on two complementary approaches to disaggregation, each with distinct characteristics and challenges: software-based (soft) disaggregation and hardware-based (hard) disaggregation.

**Soft Disaggregation (Software-based)** Soft disaggregation retains the traditional "server-centric" paradigm where the fundamental building block remains a complete server machine, but modifies software layers—particularly hypervisors and operating systems—to allow virtual machines to dynamically leverage hardware resources from multiple physical servers within the same rack. This approach benefits from emerging high-speed interconnection technologies like CXL (Compute eXpress Link), which provide memory-semantic access across physical server boundaries with latencies approaching those of local DRAM.

KrakOS investigates several research directions within soft disaggregation. For **memory disaggregation**, the team revisits fundamental OS algorithms including memory management, synchronization primitives, and checkpointing mechanisms to account for NUMA (Non-Uniform Memory Access) and CXL-based topologies where memory access latencies vary significantly depending on physical location. The research explores how user-space service delegation (as discussed in axis A2 on mutant kernels) can simplify the implementation of disaggregation mechanisms and improve system resilience by isolating complex memory management policies in recoverable user-space services. For **I/O disaggregation**, KrakOS optimizes data communication streams through automatic and transparent migration or distribution of TCP and QUIC sessions across multiple network interfaces, coupled with global and opportunistic management of memory buffers that can reduce data copies and eliminate bottlenecks in distributed application communication.

**Hard Disaggregation (Hardware-based)** Hard disaggregation represents a more radical approach requiring deep redesign of both hardware and system software architectures. Rather than organizing a rack as a collection of complete server machines (server-centric), hard disaggregation builds racks as clusters of specialized resource boards (resource-centric architecture). Each resource board, or "blade," provides only one type of resource—CPU boards contain only processors and minimal local memory, memory boards provide large pools of DRAM, storage boards host persistent storage devices, and so forth. These specialized boards are interconnected through an ultra-fast network fabric whose performance and reliability characteristics approach those of traditional within-server buses, fundamentally different from commodity inter-rack datacenter networks.

This architectural transformation opens vast design spaces that KrakOS explores systematically. The team investigates the adequate scale of disaggregated racks—determining optimal numbers of boards and their interconnection topologies to balance performance, cost, and fault isolation. Research on board dimensioning examines trade-offs between "light" boards (highly specialized with minimal resources beyond their primary

function) and "heavy" boards (incorporating more local resources for reduced network dependency). Network communication management presents unique challenges: the system must efficiently handle loopback traffic (communication within a virtual server), intra-rack traffic (between boards in the same rack), and inter-datacenter traffic (between racks or to external networks), each with vastly different performance characteristics. Energy efficiency optimizations explore how disaggregation enables fine-grained power management—for example, powering down unused memory boards or consolidating computation onto fewer CPU boards during low-load periods.

A critical and underexplored aspect of hard disaggregation is software stack design. KrakOS develops hypervisors and guest operating systems with paravirtualized interfaces specifically designed to virtualize a disaggregated rack into elastic virtual servers that can dynamically grow and shrink by adding or removing resource boards. This software stack must support existing server-centric applications without modification, enable both intra-rack virtual server migration (moving between resource configurations) and inter-datacenter migration (moving complete virtual servers between racks or datacenters), and enforce strict isolation across multiple dimensions including configuration (preventing misconfigurations from affecting other tenants), performance (ensuring one tenant cannot degrade another's performance), fault isolation (preventing failures from propagating), and security/privacy (protecting tenant data and computation from observation or interference). Notably, most existing research on disaggregation focuses on simple isolation abstractions such as Linux processes and containers; KrakOS's work on full virtual machine support for disaggregated architectures addresses a gap in current research while providing the strong isolation properties required for production multi-tenant cloud environments.

### 3.2.4 A4 - Fault Tolerance

System designers often neglect fault tolerance during initial development, focusing primarily on functionality and performance. This approach can render initially effective solutions impractical when resilience requirements are considered, necessitating costly redesigns or abandonment of otherwise promising ideas. KrakOS researchers take a different approach by considering fault tolerance as a first-class concern from the design phase, integrating resilience mechanisms into the fundamental architecture rather than retrofitting them later. The team has identified specific approaches to incorporate fault tolerance for each of the three preceding research axes (machine virtualization, mutant kernels, and disaggregation), ensuring that innovations in these areas can be deployed in production environments where failures are inevitable.

**Fault Tolerance for Virtual Machines** Observability—the practice of monitoring system execution—is essential for numerous critical functions including crash detection, hang detection, intrusion detection, and performance monitoring. However, implementing effective observability for virtual machines creates a fundamental dilemma known as the Observer/Observed problem. On one hand, the Observer and Observed must reside in distinct fault domains to prevent fault propagation; if they share a fault domain, a failure in the Observed system can corrupt or crash the Observer, defeating the purpose of monitoring. On the other hand, the Observer requires easy and efficient access to the Observed system's state to perform meaningful monitoring without introducing prohibitive performance overhead.

This dilemma becomes particularly acute in virtualized environments. A single VM can host multiple applications, making it a complex entity to monitor. Embedding both Observers and Observed components within the same VM using traditional non-virtualized abstractions (such as separate processes) proves ineffective because a VM crash necessarily crashes all contained processes, including any Observers. Existing approaches, such as the out-of-VM observability framework proposed by Ding et al. [17], attempt to solve this by dedicating a separate VM for observation. However, this architecture introduces performance-costly observation mechanisms because cross-VM communication is significantly more expensive than intra-VM operations. Moreover, it leads to substantial resource waste since each user VM requires a corresponding observer VM, effectively doubling memory, CPU, and management overhead.

KrakOS proposes a novel approach: integrating the Observer into the VMM (Virtual Machine Monitor) membrane. A VM consists of two components: the guest OS that executes applications (a black box from the datacenter manager's perspective) and the VMM that virtualizes hardware and manages the guest. The key insight is that the VMM and guest OS share the same address space, yet the guest remains isolated through hardware virtualization mechanisms like extended page tables. By extending the VMM to incorporate the Observer as a second guest alongside the guest OS, KrakOS achieves both isolation (the Observer

runs in a separate protection domain) and efficient access (the Observer can directly examine guest state without crossing VM boundaries). This architecture enables multiple critical use cases including ransomware detection through behavioral monitoring, addressing the semantic gap between hypervisor and guest OS by maintaining high-level semantic information, real-time security monitoring with minimal overhead, and performance anomaly detection that can identify subtle degradation patterns.

**Fault Tolerance for Mutant Kernels** . When OS services are externalized to user space (as described in research axis A2 on mutant kernels), they become significantly more vulnerable to failures than their kernel space counterparts. While an application process crash typically affects only that specific application, an OS service crash can impact all applications depending on that service, yet user-space services lack the protection and recovery mechanisms traditionally afforded to kernel components. This creates a fundamental challenge: how to provide the flexibility and safety benefits of user-space implementation while maintaining the reliability expectations of critical system services.

KrakOS explores three complementary approaches to address this challenge. The first approach proposes designing a new first-class abstraction specifically for OS services that acknowledges their special status—distinct from both ordinary application processes and kernel code. This abstraction would provide appropriate protection, scheduling priority, and recovery mechanisms tailored to the unique requirements of system services, aligning directly with the broader research agenda on new concurrency and isolation abstractions discussed in axis A2.

The second approach leverages a kernel fallback mechanism where both user-space and kernel-space versions of an OS service coexist. In case of user-space service failure, the system can temporarily rely on the default kernel implementation during maintenance and recovery. However, this approach introduces the significant challenge of transferring or synchronizing state between versions that may employ different policies (such as Most Recently Used versus Least Recently Used for page replacement) and maintain different internal data structures. Solving this state reconciliation problem requires either designing services with compatible internal representations or developing sophisticated state translation mechanisms.

The third approach employs user-space redundancy by replicating OS services across multiple address spaces. While replication is a well-established technique for fault tolerance, applying it to OS services presents unique challenges. Maintaining replica state coherence requires coordination protocols, but traditional replication algorithms introduce performance overhead that is unacceptable for latency-critical system services. OS services must handle queries at microsecond scale—for example, page fault handling cannot tolerate millisecond-scale coordination delays introduced by consensus protocols. Therefore, KrakOS must develop new replication techniques specifically optimized for the extreme performance requirements of system services while still providing meaningful fault tolerance guarantees.

**Fault Tolerance for Disaggregation** . The goal of this research direction is to provide correctness and availability guarantees for disaggregated systems, particularly in hard disaggregation designs where resources are physically separated into specialized boards. Prior work on disaggregation has primarily addressed conceptual models and performance optimization, often disregarding reliability concerns that become critical in production deployments.

This research specifically targets hardware crash failures where one or more resource boards crash within a rack—a failure mode that differs fundamentally from traditional node crashes in server-centric architectures. The smaller granularity of failures in disaggregated systems fundamentally reshapes both the challenges and opportunities for fault tolerance, as losing a single memory board affects multiple virtual servers simultaneously in ways that differ qualitatively from losing an entire node. Applications are more likely to encounter failures in disaggregated contexts because the disaggregation of resources increases the number of independent components that can fail, effectively multiplying failure probabilities. The impact and optimal recovery strategy depend critically on the type of failed board (CPU, memory, disk, or network) and its specific configuration, such as cache size on CPU boards or memory capacity on memory boards. Consequently, no one-size-fits-all approach can be effective—different failure scenarios demand

fundamentally different recovery mechanisms.

KrakOS pursues several research directions to address these challenges. The team is developing new formalisms for reasoning about failures, communication patterns, and consistency guarantees in disaggregated infrastructure, as existing theoretical frameworks assume monolithic server architectures. New failure models are essential because existing models cannot accurately capture the hybrid nature of disaggregated systems, where intra-rack communication over ultra-fast fabrics differs fundamentally from inter-rack communication over commodity networks. The team investigates suitable consistency models for applications executing in disaggregated datacenters, balancing the tension between strong guarantees that simplify application development and relaxed models that enable better performance. Finally, cache coherence protocols must be redesigned to handle failures gracefully—for example, determining correct behavior when a CPU core that exclusively owns a cached object crashes, potentially leaving other cores with stale data or blocked on unavailable resources.

## 4 Application domains

### 4.1 Overview

The research efforts of KrakOS target data centers that run all types of applications, unlike the HPC (High-Performance Computing) domain which focuses on specific scientific workloads. KrakOS aims at accommodating various application types while maintaining a key constraint: non-degradation of performance for one application type in favor of another (unless explicitly specified as a desired policy).

Operating across multiple system layers (hypervisor, operating system, and middleware), KrakOS addresses several target areas with specific characteristics and requirements.

### 4.2 Hypervisor Layer

#### 4.2.1 Target Hypervisors

KrakOS focuses on open-source hypervisors that dominate cloud deployments. **Xen** is used extensively in cloud environments, particularly valued for its strong security properties and robust isolation mechanisms that enable safe multi-tenant deployments. **KVM**, integrated directly into the Linux kernel, has been widely adopted by major cloud providers due to its performance characteristics and seamless integration with existing Linux infrastructure. By focusing on these two dominant hypervisors, KrakOS ensures that research contributions can be rapidly adopted in production cloud environments.

#### 4.2.2 Cloud Deployment Models

KrakOS research addresses both private and public cloud deployment models, each with distinct characteristics and requirements. In **private clouds**, where applications belong to a single entity, best-effort resource management is often permissible, allowing the focus to remain on overall efficiency and performance optimization across the entire infrastructure. In contrast, **public clouds** hosting applications from different owners require that each application receives its subscribed amount of resources, necessitating strict isolation mechanisms and rigorous SLA (Service Level Agreement) enforcement to prevent interference between tenants and ensure contractual obligations are met.

#### 4.2.3 Cloud Service Models

KrakOS deliberately limits its scope to the most complex cloud service models where systems research can have the greatest impact. **Infrastructure as a Service (IaaS)** represents the traditional VM-based cloud model with startup times on the order of minutes and complex requirements for resource management, live migration, and multi-tenant isolation. **Function as a Service (FaaS)**, one of the newest and most challenging cloud models, demands ultra-fast startup times at the microsecond scale, elastic scaling that can respond to rapid workload changes, and fine-grained resource allocation mechanisms that can efficiently multiplex short-lived function invocations. The constraints of FaaS fundamentally challenge traditional operating system and virtualization assumptions, making it a particularly rich area for systems innovation.

### 4.3 Operating System Layer

KrakOS targets **Linux** as its primary operating system due to several compelling factors. Linux enjoys widespread adoption in both cloud and enterprise environments, making research contributions immediately relevant to production deployments. Its open-source nature enables deep modifications and experimental reimplementations of core subsystems, essential for systems research. The rich ecosystem and strong community support ensure that innovations can be integrated into mainline development and benefit from collaborative improvement. Finally, Linux's presence across the computing spectrum—from massive cloud datacenters to resource-constrained edge devices—ensures that KrakOS research on Linux has broad applicability.

### 4.4 Middleware and Orchestration

KrakOS addresses several critical middleware layers that sit between applications and infrastructure. **Message-Oriented Middleware (MOM)** plays a vital role in application interoperability within distributed systems, enabling inter-application communication, service decoupling, and asynchronous message processing that allows systems to scale and evolve independently. The **Edge-Cloud continuum** represents an increasingly important deployment model where computation must be distributed across multiple tiers—from resource-constrained edge devices to massive cloud datacenters—requiring sophisticated mechanisms for latency-sensitive application placement and resource management across heterogeneous distributed environments. **Kubernetes** serves as the primary focus for container orchestration research, as it has become the de facto standard for automated deployment, scaling, and management of containerized applications, with direct connections to KrakOS research on disaggregation and resource management. Finally, middleware for **large-scale data processing**, encompassing both real-time stream processing and batch analytics, presents challenges in efficiently managing data movement and computation placement, directly connecting to the team's work on storage and memory management optimization.

### 4.5 Domain-Specific Applications

#### 4.5.1 Genomics and Bioinformatics

Through the **ANR PicNIC project** in collaboration with ICO (Institut de Cancérologie de l'Ouest), KrakOS addresses critical challenges in genomic data processing. The research focuses on reducing data movements in genomic datacenters, optimizing execution times for complex genomic analysis pipelines, minimizing energy consumption, and improving data-intensive workload performance. Genomic applications present unique challenges with extremely large datasets ranging from terabytes to petabytes, complex multi-stage computational pipelines with diverse resource requirements, I/O-intensive operations that can bottleneck on storage systems, and critical needs for data locality optimization to avoid expensive data transfers. These characteristics make genomics an ideal testbed for KrakOS research on disaggregation, efficient I/O, and energy-aware resource management.

#### 4.5.2 Memory-Intensive Applications

Given fundamental memory resource limitations in datacenters and the need to accelerate disk-intensive applications, KrakOS specifically targets memory-intensive workloads. **Key-value stores** such as Memcached serve as caching systems that maintain critical data in memory for low-latency access, requiring efficient in-memory data structure management and horizontal scalability across multiple servers. **Graph processing applications** perform large-scale analytics on graph structures, characterized by random memory access patterns that challenge traditional memory hierarchies and demand sophisticated memory management to maintain performance at scale.

#### 4.5.3 Microservices Architectures

Microservices have emerged as the dominant programming model for modern Internet services, presenting both opportunities and challenges for systems research. These architectures consist of distributed, loosely-coupled services that can be independently deployed and scaled, often implemented in multiple programming

languages (polyglot development), with complex inter-service communication patterns. For KrakOS research, microservices introduce challenges in fine-grained resource allocation (as individual services may have vastly different resource needs), service discovery and intelligent routing, fault tolerance mechanisms that prevent cascading failures across service dependencies, and comprehensive performance monitoring and observability that can track requests across dozens of service invocations.

## 4.6 Cross-Cutting Application Characteristics

KrakOS research addresses applications spanning an enormous range of characteristics, ensuring that proposed solutions are robust and generally applicable. **Latency requirements** vary from microsecond-scale responsiveness demanded by FaaS functions to minutes or hours acceptable for batch processing jobs. **Resource consumption** ranges from lightweight serverless functions consuming mere megabytes of memory to resource-intensive analytics requiring hundreds of gigabytes and multiple accelerators. **Deployment patterns** include single-tenant applications in private clouds, multi-tenant services in public clouds, and hybrid deployments spanning cloud and edge infrastructure. **Data patterns** encompass data-intensive applications like genomics where I/O dominates execution time, and compute-intensive simulations where CPU and accelerator performance are critical. This diversity of application domains ensures that KrakOS solutions must be general, robust, and applicable to real-world production environments across multiple industries rather than optimized for narrow use cases.

# 5 Social and environmental responsibility

## 5.1 Energy Efficiency and Green Computing

Energy efficiency is embedded as a core concern throughout KrakOS research activities, reflecting the team's commitment to reducing the environmental footprint of computing systems. The team conducts research on energy-aware virtualization mechanisms that optimize power consumption without sacrificing performance, addressing the growing challenge of datacenter energy costs and carbon emissions. This includes developing novel resource management algorithms that consider energy as a first-class optimization criterion alongside traditional performance metrics. The team has developed specialized tools for measuring and optimizing energy consumption at multiple system layers. These measurement frameworks provide the foundation for understanding energy behavior and designing more efficient systems.

### 5.1.1 Participation in Standardization Initiatives

Nicolas Palix serves as mission leader for "Action Monitoring" within **GDRS Écoinfo**, the national research network dedicated to eco-responsible digital practices. In this role, he coordinates efforts to establish best practices and metrics for evaluating the environmental impact of digital technologies across French research institutions. The team actively contributes to **AFNOR SPEC 2314 on Frugal AI**, working to define standards and best practices for resource-efficient artificial intelligence. This standardization effort aims to ensure that AI systems can deliver high performance while minimizing computational resource consumption and energy usage, making AI technologies more accessible to organizations with limited infrastructure. The three-year **IAoundé Project**, funded by Région AURA, focuses specifically on frugal AI research and capacity building, promoting sustainable computing practices particularly for resource-constrained environments in developing countries.

## 5.2 Diversity, Equity, and Inclusion

### 5.2.1 Leadership in DEI Initiatives

Alain Tchana serves as a member of the **ACM (Association for Computing Machinery) Diversity, Equity, and Inclusion Council**, a prestigious appointment that recognizes his leadership in promoting inclusive practices in computing research and education. Through this role, Tchana advocates for increased representation of underrepresented groups in computer science research and education, drawing on his extensive experience building partnerships between European and African institutions. He works to ensure

equitable access to computing resources and opportunities, particularly for researchers and students from developing countries who face systemic barriers to participation in international research. Within KrakOS and the broader systems research community, we foster inclusive research practices that value diverse perspectives and create welcoming environments for all researchers.

### 5.2.2 Gender Diversity Reflection

The team engages in ongoing self-assessment through internal discussions specifically focused on "how to increase the number of women in the team," recognizing that gender diversity remains a critical challenge in systems research. KrakOS implements conscious recruiting practices designed to encourage applications from underrepresented groups, including targeted outreach to diverse student populations and careful attention to inclusive language in job postings and internship descriptions. The team prioritizes creating a welcoming and supportive environment for all members, with policies that promote work-life balance and accommodate diverse needs. While the team acknowledges significant work remains to achieve representative diversity, these ongoing efforts—particularly the financial commitment to women interns—reflect a genuine commitment to structural change rather than symbolic gestures.

## 5.3 Open Science and Reproducible Research

KrakOS maintains a strong commitment to open science principles, recognizing that scientific progress depends on transparent sharing of methods, data, and results. The team publishes open-source software and tools including Faho (PIM operating system), vPIM (PIM virtualization), MigCheck (migration feasibility testing), GoodKit (VM introspection), and B-Side (system call identification), making these research artifacts freely available to the community. Team members actively contribute to major open-source projects including the Xen hypervisor and Linux kernel, ensuring that research innovations can benefit production systems used worldwide. The team regularly organizes workshops and seminars for knowledge dissemination, including tutorials at conferences like CompAS 2025, the Workshop Défi OS, and the Xen Project Winter Meetup, fostering dialogue between researchers and practitioners.

## 6 Highlights of the year

### 6.1 Team Creation and Inauguration

KrakOS was officially created on October 1, 2024, as an Inria project-team in partnership with Université Grenoble Alpes, Grenoble INP, and CNRS. The team's inauguration ceremony took place on November 25, 2024, at the Inria Centre at Université Grenoble Alpes. This event was honored by the attendance of Sacha Krakowiak, the distinguished emeritus professor after whom the team is named, symbolizing the continuity between pioneering work in operating systems research in Grenoble and KrakOS's mission to advance the field for modern datacenter environments.

### 6.2 HDR and PhD Defenses

Baptiste Lepers successfully defended his Habilitation à Diriger des Recherches (HDR) on December 12, 2024, marking a significant milestone for the team and recognizing his contributions to operating systems research, particularly in the areas of scheduling, memory management, and system performance optimization. Two PhD students completed their doctoral work in 2024-2025: Papa Assane Fall and William Wu.

### 6.3 Major Publications

The team achieved remarkable publication success at premier systems conferences, demonstrating the quality and impact of KrakOS research. Papers were accepted at **EuroSys 2025** and **NSDI 2025**, two of the most selective venues in systems research. Two papers were accepted at **APSys 2025**. Additional acceptances include **SIGMETRICS 2025** on Intel User Interrupts performance analysis, **ASIACCS 2025** on SIMBox fraud detection, and two papers at **Middleware 2024 and 2025** on Processing-in-Memory virtualization and binary-level system call identification.

## 6.4 Awards and Recognition

Team members and alumni received prestigious recognition for their research contributions. Anne-Josiane Kouam was honored with the Prix Science Ouverte 2025, recognizing her commitment to open science principles and her work on fraud detection in telecommunications that balances security with privacy preservation. Yasmine Djebrouni received the Accessit (honorable mention) for the GDR RSD Thesis Award 2025. Stella Bitchebe earned the Accessit for the GDR RSD Thesis Award 2024 for her thesis on nested virtualization optimization.

## 6.5 International Collaborations

KrakOS expanded its international research network through multiple funding mechanisms and partnership programs. The team secured funding through the **France Berkeley Fund** for collaboration with Natacha Crooks at UC Berkeley. An **Associated Team proposal with the University of British Columbia**, co-led with Mohammad Shahradsad, is under review to advance responsible cloud computing research. The **Associated Team with ENSPY Cameroon**, co-led with Thomas Bouetou, was approved and supports the IAoundé frugal AI initiative. An **Associated Team with the University of Sydney**, partnering with Vincent Gramoli, enables blockchain systems research and PhD co-supervision. Additionally, a **Mourou/Strickland Program** collaboration with Mohammad Shahradsad at UBC facilitates advanced research exchanges.

The team maintained active international mobility with significant research visits: Willy Zwaenepoel from the University of Sydney spent six months at KrakOS, contributing expertise in distributed systems; Gohar Irfan Chaudhry from MIT visited for two weeks for collaborative research discussions; Alain Tchana conducted extended research stays at MIT (2.5 months) and UBC (2 weeks); Maxime Collette and Alain Tchana visited ETH Zurich for collaborative discussions; and multiple researchers exchanged visits between Cameroon and Grenoble, strengthening the IAoundé partnership.

## 6.6 Conference Organization and Leadership

Team members held prominent leadership positions in the systems research community: Artifact Evaluation Chair for OSDI/ATC 2025 and SOSP 2025, Shadow PC Chair for EuroSys 2025. Team members served on program committees of major conferences including EuroSys 2025 and 2026, SIGMETRICS 2025, NSDI 2025, ASPLOS 2025, Middleware 2025, NCA 2025, SOSP 2026 and FAST 2026. Vania Marangozova-Martin served as President of the system track for CompAS 2025, the premier French-language systems conference. Team members also participated in numerous PhD defense committees, serving as presidents, reviewers, and CSI (Comité de Suivi Individuel) members, contributing to doctoral education across France.

KrakOS organized several community events including the **Xen Project Winter Meetup**, co-organized with Vates on January 30-31, 2025, bringing together international contributors to the Xen hypervisor. The team also organized the **Workshop Défi OS** on December 13, 2024, fostering collaboration among French research teams working on operating systems challenges.

## 6.7 Industrial Partnerships

KrakOS actively pursued industrial partnerships to ensure research relevance and facilitate technology transfer. With **Vates**, a leading French virtualization company, the team submitted proposals for a LabCom VirtDisk-Lab and a BPI project, and established one CIFRE PhD thesis on virtualization and storage systems. A **MIAI Industrial Chair proposal** was submitted jointly with Vates and EasyVirt, focusing on confidential computing and AI-based health workloads. The team secured one **CIFRE thesis with Huawei Technologies**, advancing research on AI optimization. Two ongoing **CIFRE theses with Orange Labs** address virtual machine introspection and machine failure detection, contributing to operational challenges in large-scale cloud deployments. While not all funding applications were successful, these partnerships demonstrate KrakOS's commitment to bridging academic research and industrial needs.

## 7 Latest software developments, platforms, open data

### 7.1 New Software

#### 7.1.1 USM

**USM** is a comprehensive framework for developing and deploying memory management policies in Linux entirely in userspace. Unlike traditional approaches where memory management policies are embedded in the kernel, USM adopts a microkernel-inspired architecture that moves policy implementation to userspace while retaining critical mechanisms in the kernel. The framework provides complete coverage of memory management aspects including page allocation, page eviction decisions (what pages to evict, when to evict them, and where to store evicted content), and integrated policies that coordinate these decisions. USM enables rapid development and safe experimentation with novel memory management strategies without requiring kernel modifications or system reboots.

The source code has been publicly released to support further research and adoption by the systems community. Development is led by Alain Tchana, Renaud Lachaize, Papa Fall and Jean-Pierre Lozi.

#### 7.1.2 MigCheck

**MigCheck** is a tool designed to test the feasibility of virtual machine migration in heterogeneous hardware environments before actual migration attempts. The tool performs comprehensive analysis of hardware compatibility, examining CPU instruction set architectures to predict potential migration issues. By identifying incompatibilities early, MigCheck prevents costly migration failures and service disruptions in production cloud environments. The tool is currently in the maturation phase and was submitted for LSI Carnot funding. Development is led by Alain Tchana, Renaud Lachaize, and Kenta Ishiguro.

#### 7.1.3 vPIM

**vPIM** provides comprehensive virtualization support for Processing-in-Memory (PIM) devices, enabling multiple virtual machines to efficiently share PIM hardware while maintaining strict isolation and performance guarantees. The system implements novel scheduling and resource management policies specifically designed for the unique characteristics of PIM architectures, where computation occurs directly within memory arrays. vPIM addresses critical challenges in PIM virtualization including memory allocation, DPU (Data Processing Unit) scheduling, and performance isolation between co-located tenants. The source code has been publicly released to support further research and adoption by the systems community. Contact: Alain Tchana.

#### 7.1.4 Faho

**Faho** is an operating system specifically designed for UPMEM Processing-in-Memory devices, providing dynamic and efficient sharing of Data Processing Units (DPUs) among multiple applications. Unlike traditional batch-oriented approaches, Faho implements time-sharing mechanisms that allow unpredictable job arrivals to be handled efficiently while maintaining fairness and high utilization. The system includes sophisticated scheduling algorithms that account for the unique characteristics of PIM hardware, including limited on-chip memory and the cost of data movement between host and PIM devices. Faho's source code is publicly available, facilitating integration into existing PIM-based systems. Development is led by Alain Tchana and Renaud Lachaize.

#### 7.1.5 GoodKit

**GoodKit** provides an efficient and robust virtual machine introspection framework that enables security monitoring, debugging, and analysis of guest VM. The framework is designed to minimize performance overhead while providing comprehensive visibility into VM internals, including memory access patterns, system call activity, and kernel data structures. GoodKit addresses the semantic gap problem that traditionally plagues VM introspection by maintaining high-level semantic information about guest OS structures. This capability is essential for security applications such as intrusion detection, malware analysis, and compliance monitoring in cloud environments. The source code is publicly available. Development is led by Alain Tchana and Renaud Lachaize.

### 7.1.6 B-Side

**B-Side** performs sophisticated binary-level static identification of system calls in compiled applications, enabling security analysis and monitoring without requiring access to source code. The tool employs advanced program analysis techniques including control flow reconstruction, symbolic execution, and pattern matching to accurately identify system call sites even in heavily optimized or obfuscated binaries. B-Side is particularly valuable for security auditing of closed-source software, legacy applications, and potentially malicious code where source access is unavailable. Contact: Alain Tchana.

### 7.1.7 P4Cemaker

**P4CEMaker** is a novel system designed to semiautomatically accelerate existing RDMA-based consensus protocols through the use of a programmable switch. We demonstrated the usefulness of P4CEMaker by accelerating four different consensus protocols, achieving up to 2 times performance improvement in around a day of work per protocol. Contact: Baptiste Lepers.

### 7.1.8 DirtBuster

**DirtBuster** is a tool that identifies scenarios in which the CPU caches perform suboptimally. CPU caches have been heavily optimized to cache DRAM, but are not increasingly used to cache data coming from other memory devices (e.g., persistent memory, CXL memory, FPGA memory). In such scenarios, caches may perform suboptimally. By using a combination of static and dynamic analysis, DirtBuster identifies applications and code regions that are likely to suffer from suboptimal cache behavior. Developers can then add hints to direct the cache (these hints are also suggested by DirtBuster). Contact: Baptiste Lepers.

## 7.2 New platforms

### 7.2.1 IBARA - Portable Micro-Cluster for Africa

**IBARA** is a portable and autonomous micro-cluster specifically designed for teaching, research, and service hosting in countries facing electrical infrastructure challenges. This innovative platform integrates high-performance micro-computers within a compact transportable suitcase, enabling rapid deployment of computing capabilities in isolated locations or areas with severe electricity deficits.

IBARA's distinguishing feature is its hybrid power system guaranteeing uninterrupted operation. The platform operates on mains electricity when available, automatically switches to a storage battery (automotive-type) during power outages, and maintains battery charge through an integrated foldable solar panel, enabling completely off-grid operation. This design addresses the reality of frequent power interruptions in many African regions while providing the reliable computing infrastructure essential for modern education and research.

Within the IAoundé project framework, IBARA enables KrakOS to deliver hands-on teaching on cloud computing, virtualization, and distributed systems at partner universities (University of Yaoundé 1, ENSPY) without requiring expensive datacenter infrastructure. Students gain practical experience with modern technologies—virtual machines, distributed applications, container orchestration—using a system designed for their specific infrastructural constraints. As a research platform, IBARA supports experiments on energy-aware scheduling, resilient system design, and efficient resource utilization, aligning with KrakOS's work on green computing while addressing real-world deployment challenges. The platform also provides practical hosting capabilities for local university services, reducing dependence on distant cloud providers and supporting digital sovereignty.

**Key Features:** Portable micro-cluster in transportable suitcase; hybrid power (mains/battery/solar); autonomous operation in isolated locations; supports teaching, research, and local hosting

### 7.2.2 Grid'5000

KrakOS extensively uses Grid'5000, a large-scale distributed computing testbed for experimental research. The platform provides access to diverse hardware configurations essential for validating virtualization and resource management research.

### 7.2.3 SLICES-FR

The team participates in SLICES-FR, the French component of the European SLICES infrastructure for large-scale experimental research in networking, distributed computing, and IoT.

## 8 New results

### 8.1 Physical Memory Management in Userspace (USM)

Papa Assane Fall's PhD research addressed a critical challenge in datacenter memory management. Main memory is a critical resource in datacenters due to its major impact on application performance and server costs. However, Linux's memory management (MM) system, designed to be general-purpose, is not always optimal for the diverse workload requirements encountered in production cloud environments.

Fall introduced **USM (User-Space Memory)**, the first complete framework for rapid development of memory management policies in Linux. USM adopts a microkernel-inspired design that enables MM policies to run entirely in userspace, aligning with KrakOS's broader research agenda on mutant kernels (axis A2). This architecture addresses several key requirements for extensible memory management including generality (supporting diverse policy types), simplicity (reducing development complexity), safety (preventing policy bugs from crashing the kernel), reconfigurability (enabling dynamic policy changes), transparency (maintaining compatibility with existing applications), and observability (providing detailed insights into memory behavior).

**Participants:** Assane Fall, Jean-Pierre Lozi, Renaud Lachaize, Alain Tchana.

### 8.2 Processing-in-Memory Virtualization

The team made significant advances in Processing-in-Memory (PIM) virtualization through two complementary systems. First, **vPIM** [16] provides a comprehensive virtualization solution for PIM devices, addressing the challenge of efficiently virtualizing emerging PIM architectures. This system enables multiple virtual machines to share PIM hardware while maintaining performance isolation between tenants, a critical requirement for cloud environments. Second, the team developed **Faho**, a time-sharing system designed to optimally manage UPMEM PIM resources when independent jobs arrive unpredictably in the system. Faho implements sophisticated scheduling policies that balance fairness, throughput, and energy efficiency, demonstrating that PIM systems can effectively support multi-tenant workloads in production cloud environments. This work is under review at ISCA.

**Participants:** Maxime Collette, Ni Weihao, Renaud Lachaize, Alain Tchana.

### 8.3 Heterogeneous VM Migration

Research on heterogeneous virtual machine migration led to the development of **MigCheck**, a tool that tests migration feasibility across different hardware platforms before attempting actual migration. This work is crucial for cloud operators managing diverse hardware fleets, as it prevents migration failures that can lead to service disruptions and resource waste. This research addresses a critical operational challenge in modern cloud datacenters where hardware heterogeneity is increasing due to rapid technology evolution. This work is under review at EuroSys.

**Participants:** Kenta Ishiguro, Fonyuy-Asheri Caleb, Eloua Barraud, David Bromberg, Renaud Lachaize, Alain Tchana.

## 8.4 Understanding Intel User Interrupts

Yves Koné's work on Intel User Interrupts provides deep insights into this emerging hardware feature, which enables user-space applications to receive hardware interrupts without kernel intervention. Through comprehensive performance characterization and analysis, the research demonstrates both the opportunities and limitations of this new mechanism. This work was accepted at SIGMETRICS 2025 and contributes to understanding how modern hardware features can be leveraged to improve application performance at the microsecond scale.

**Participants:** Yves Kone, Louis Duval, Pascal Felber, Daniel Hagimont, Renaud Lachaize, Alain Tchana.

## 8.5 System Call Identification for Security

The team developed **B-Side**, a tool that enables binary-level static identification of system calls in compiled applications. B-Side provides a foundation for security monitoring and analysis tools that work without requiring source code access, addressing a critical need in security auditing of closed-source software and legacy systems. This work was published at Middleware 2025.

**Participants:** Gaspard Thévenon, Kevin Nguetchouang, Kahina Lazri, Pierre Olivier, Alain Tchana.

## 8.6 SIMBox Fraud Detection

Josiane Kouam's work on detecting SIMBox fraud through latency anomalies (**SigN**) demonstrates how system-level monitoring can address real-world security challenges in telecommunications. SIMBox fraud, where international calls are illegally routed through mobile networks to avoid charges, costs telecom operators billions of dollars annually. The SigN system leverages subtle timing differences in call routing to identify fraudulent traffic patterns without requiring deep packet inspection or customer data access, making it both privacy-preserving and GDPR-compliant. This research was accepted at ASIACCS 2025 and is being deployed in collaboration with telecom operators in Africa to combat fraud while respecting user privacy.

**Participants:** Josiane Kouam, Aline Carneiro, Philippe Martins, Cédric Adjih, Alain Tchana.

## 8.7 P4Cemaker

Paul Breuil worked on P4CEMaker, a novel system designed to semi-automatically accelerate existing RDMA-based consensus protocols using a programmable switch. Central to the design of P4CEMaker is the insight that, despite the diversity of algorithmic approaches used by consensus protocols (e.g., fault detection and leader election), they rely on a common set of networking operations such as scattering and gathering values, which can be offloaded to programmable switches. P4CEMaker consists of two components: a dynamic analysis tool that automatically detects these network operations and provides developers with precise call-graph information showing where and how they are executed in the code, and a versatile hardware acceleration library that enables these operations to run in hardware with minimal code changes. Paul used P4CEMaker to accelerate four different consensus protocols, achieving up to a 2× performance improvement with roughly one day of work per protocol. P4Cemaker was published in ICDCS 2025.

**Participants:** Jakob Nibler, Thomas Ropars.

## 8.8 Pre-Stores

William Wu worked on improving the performance of CPU caches when they are used to cache memories other than regular DRAM. These scenarios are becoming common (persistent memory, remote memory accessed via CXL, etc.). William introduced the notion of software pre-storing - the converse of software prefetching. With software pre-fetching, instructions are inserted in the code to asynchronously move data up in the memory hierarchy. With software pre-storing, instructions are inserted to direct the CPU to asynchronously move data down in the memory hierarchy. Pre-storing can be implemented by using existing processor instructions. Software pre-storing provides performance benefits for write-heavy applications on emerging architectures.

William identified application scenarios in which software pre-storing is beneficial, and developed a tool, DirtBuster, that identifies applications and code regions that can benefit from pre-storing. He evaluated the concept of software pre-storing and the DirtBuster tool on two CPU architectures (ARM and x86) and two types of cacheable memories (PMEM and cache-coherent DRAM accessed through an FPGA). He demonstrate dperformance improvements for key-value stores, HPC applications, message passing, and Tensorflow, by up to 2.3x. The work was published in EuroSys'25.

**Participants:** Xiaoxiang Wu, Baptiste Lepers, Willy Zwaenepoel.

## 8.9 Carbon Footprint of Storage in Datacenters

The team works on the analysis on the carbon footprint of storage in the cloud. During the year, our work has focused on studying the impact of the storage technology (HDD vs SSD) on the trade-off between performance and carbon footprint, considering the case of key-value stores. The work of Jakob Nibler has demonstrated that this type of database, commonly used in datacenters, there could be situations where using HDDs instead of SSDs can be better from the carbon footprint point of view. This is especially true if the applications are unable to take full advantage of the high performance of SSD devices, and if the energy powering the datacenters has a low carbon intensity. These results open new research directions for reducing the environmental impacts of Cloud infrastructures.

**Participants:** Jakob Nibler, Thomas Ropars.

### 8.9.1 IBARA - Portable Micro-Cluster for teaching

**IBARA** is a portable and autonomous micro-cluster specifically designed for teaching, cloud and big data research, and hosting services in African countries. This innovative platform addresses a critical challenge: providing reliable computing infrastructure in environments with unstable or unavailable electrical power. IBARA integrates a set of high-performance micro-computers within a compact, easily transportable suitcase, making it an ideal solution for rapid deployment of computing capabilities in isolated locations or areas with severe electricity deficits.

The platform's major strength lies in its hybrid power system that guarantees uninterrupted operation under varying power conditions. When standard electrical current is available, IBARA operates on mains power like conventional computing infrastructure. However, when power outages occur—a frequent occurrence in many African regions—the system immediately switches to a storage battery (automotive-type battery) ensuring continuous operation without data loss or service interruption. To maintain long-term autonomy, the battery is kept charged through a small foldable solar panel that can be integrated into the suitcase or connected externally, enabling completely off-grid operation in sunny conditions typical of many African deployments.

**Participants:** Blandine Ntchoutta, Alain Tchana.

## 9 Bilateral contracts and grants with industry

- **Vates.** KrakOS maintains a strong partnership with **Vates**, a leading French virtualization company. Donald Onana started his CIFRE PhD thesis in November 2025, focusing on VM observability. The collaboration extends through the **MIAI Industrial Chair** (under review) on confidential computing and AI-based health workloads, combining expertise in secure virtualization with machine learning applications in healthcare. A potential CIFRE thesis for Louis Duval is under discussion. This partnership is further strengthened through the **ANR YUPIM project**, which brings together academic and industrial expertise in Processing-in-Memory virtualization.
- **Orange Labs.** The team collaborates with **Orange Labs** through two CIFRE PhD theses. Dufy Tegua focuses on virtual machine introspection for security and monitoring, while Eric Okala works on machine failure detection and recovery mechanisms in cloud environments. These collaborations are integrated within the **ANR SecondChance project** and the **ANR SCALER project**.
- **Huawei.** KrakOS established a CIFRE partnership with **Huawei Technologies**, supporting Benjamin Priour's PhD research on AI workload optimization.

## 10 Partnerships and cooperations

### 10.1 International Initiatives

KrakOs is involved in the Important Project of Common European Interest on Next Generation Cloud Infrastructure and Services (IPCEI-CIS). More specifically, KrakOs contributes to the E2CC (Eco Edge to Cloud Continuum) project.

#### 10.1.1 Associated Teams

- **University of British Columbia (Canada).** KrakOS has submitted a proposal for an Inria Associated Team with **Mohammad Shahradsat at the University of British Columbia**, focusing on responsible cloud computing with emphasis on energy efficiency, carbon-aware scheduling, and sustainable datacenter operations. The proposal is currently under review.
- **The Cameroon (ENSPY, University of Yaoundé).** An Inria Associated Team with **Thomas Bouetou at ENSPY and University of Yaoundé 1** has been accepted, centered on frugal AI research and capacity building in resource-constrained environments. This partnership strengthens long-term collaboration with Cameroonian institutions and supports the IAoundé project objectives.
- **The University of Sydney (Australia).** The team established an Inria Associated Team with **Vincent Gramoli at the University of Sydney**, focusing on blockchain systems, distributed consensus protocols, and high-performance distributed ledger technologies. This collaboration includes co-supervision of PhD student Paul Breuil and facilitates student mobility between France and Australia. In addition, Xiaoxiang Wu and Yuben Yang, two PhD students of Baptiste Lepers (hired before Baptiste joins the team) during six months.

#### 10.1.2 Other International Collaborations

- **The IAoundé project**, funded by the PAI AURA, establishes a formal collaboration with Cameroonian institutions including University of Yaoundé 1 and ENSPY, promoting frugal computing research. Seven researchers from Cameroon visited us during the year and we realized eight visits in Cameroon.
- KrakOS collaborates with **Pierre Olivier at the University of Manchester, UK**, on operating systems and virtualization research, including co-supervision of PhD students and joint publications on systems security.
- The team partners with **Pascal Felber at the University of Neuchâtel, Switzerland**, on leveraging modern hardware features for improving performance.

- Collaboration with **Natacha Crooks at UC Berkeley, USA**, is supported by the France Berkeley Fund, focusing on building a uniform framework for memory management and thread scheduling.
- **Adam Belay at MIT** hosted Alain Tchana for research discussions on operating systems for microsecond-scale computing and datacenter efficiency.
- The team has collaboration with **Timothy Roscoe at ETH Zurich, Switzerland**, with multiple research visits by Alain Tchana, Baptiste Lepers, and Maxime Collette, exploring systems architecture and hardware-software co-design.
- KrakOs collaborates with the team of Fumio Machida (University of Tsukuba) on the modeling of performance anomalies in micro-services applications. Gabriel Antunes Grabber visited the team for 3 months (April-June 2025) thanks to a UGA IDEX formation grant.

## 10.2 National Initiatives

### 10.2.1 ANR Projects

- The **ANR PRCE YUPIM project**, led by Principal Investigator Alain Tchana, is currently ongoing and focuses on advancing Processing-in-Memory virtualization technologies for next-generation cloud infrastructures.
- **ANR PRME KNext**, under the leadership of Baptiste Lepers, was submitted to develop next-generation kernel architectures that leverage emerging hardware features and new concurrency abstractions for improved performance and security.
- The **ANR PRC XRay project**, with Nicolas Palix as Scientific Responsible, was submitted to develop advanced static analysis tools for Linux kernel code, improving security and reliability through automated verification techniques.

### 10.2.2 PEPR Projects

- KrakOS is actively involved in the **PEPR Cloud**, contributing to three projects: DIVA, STEEL, and TARANIS.

### 10.2.3 Inria Challenges (Défis)

- The **Défi Inria OS** (Operating Systems Challenge) provided substantial support to KrakOS, funding three PhD theses, one postdoctoral position, and two six-month engineering positions, enabling the team to pursue ambitious research directions in modern operating systems design.

### 10.2.4 Regional Projects

- KrakOS received funding from **Région Auvergne-Rhône-Alpes** for the three-year **IAoundé project** focused on frugal AI research, strengthening partnerships with African institutions and promoting sustainable computing practices in resource-constrained environments. We have also received funding from the UGA International Research Booster for the same collaboration.
- Two **LIG Émergence projects** were accepted.
- KrakOS received funding from **LabEx Persyval-Lab** for research on virtualization of UPMEM Processing-in-Memory (PIM) technology, advancing the integration of emerging memory-centric computing architectures into cloud environments.

### 10.3 Collaboration with Other Research Teams

- KrakOS collaborates with the **WIDE team at Inria Rennes** through the co-supervision of one PhD student (Fonyuy-Asheri Caleb), focusing on VM live migration on heterogeneous processors. While located in Rennes, Fonyuy-Asheri Caleb visits KrakOS every two months for at least one week.
- The team works with the **STACK team at Inria** on building a carbon aware FaaS framework, co-supervising two master's interns.
- Collaboration with the **Whisper team at Inria** involves the co-supervision of three PhD students working on memory management, semantic gap, and bug finding in Linux.
- KrakOS partners with the **SEPIA team at IRIT (Toulouse)** to co-supervise three PhD students on topics related to memory management in virtualized systems, security and IO improvement.
- The team collaborates closely with **AGEIS Lab at UGA** on GDPR compliance and data protection research, co-supervising three master's interns and two postdoctoral researchers.

### 10.4 Conference and Workshop Organization

- The team co-organized the **Xen Project Winter Meetup** on January 30-31, 2025 in Grenoble, bringing together international contributors and users of the Xen hypervisor.
- KrakOS organized the **Workshop Inria Défi OS** on December 13, 2024, in Grenoble, facilitating collaboration and knowledge exchange among French research teams working on operating systems.
- The team organized the **IAoundé Conference** with events in June 2025 in Grenoble and August 2025 in Cameroon, promoting frugal AI and systems research in partnership with African institutions.
- KrakOS organized the **Workshop VMPSec** (Virtualization, Migration, Performance and Security) in June 2025 in Grenoble, addressing critical challenges in modern virtualization technologies.
- The team supported the creation of a new **Nuit de l'Info 2025** site in Cameroon for students participating in the IAoundé project, extending this popular French student programming competition to Africa.

## 11 Dissemination

### 11.1 Invited Talks

- **Alain Tchana** gave invited talks at multiple prestigious venues including GT SSLR (GDR Sécurité) in Paris, ETH Zurich, UBC, Seine AI workshop organized by Huawei, the 128-bit RISC-V European workshop at HiPEAC Barcelona, and the midi de la recherche at ENSIMAG.
- **Baptiste Lepers** delivered invited talks at ETH Zurich, gave a keynote at JSI Inria, and presented at the PizzaTalk series at LIG.
- **PhD students** presented their accepted papers at major international conferences including Middleware 2024, APSys 2025, EuroSys 2025, NSDI 2025, SIGMETRICS 2025, and ASIACCS 2025.
- The team organized a tutorial on virtualization at CompAS 2025, sharing expertise and best practices with the French-speaking systems research community.

### 11.2 Scientific Expertise

- **Nicolas Palix** serves as mission leader for "Action Monitoring" within GDRS Écoinfo and contributes to AFNOR SPEC 2314 on Frugal AI standardization.
- **Renaud Lachaize** is a member of the MIAIA Cluster selection committee.
- **Fabienne Boyer** serves as the representative of LIG at the MACI scientific council.

- **Alain Tchana** served as external reviewer for ERC Advanced Grants 2026.
- Team members serve on program committees of several international conferences including EuroSys, SOSp, ASPLOS, Middleware, NSDI, CCGrid, IC2E and NCA.

### 11.3 Research Administration

- **Alain Tchana** served as member of CoNRS (Comité National de la Recherche Scientifique) and serves on the ACM DEI Council (Diversity, Equity, and Inclusion).
- **Renaud Lachaize** is a member of the SIGOPS ASF staff.
- **Noël De Palma** serves as head of LIG (Laboratoire d’Informatique de Grenoble).
- **Thomas Ropars** is a member of the GDR RSD board (In charge of the relations between the GDR and the conferences and schools).

### 11.4 Teaching - Supervision - Juries

#### 11.4.1 Teaching

All permanent team members are faculty members (enseignants-chercheurs) with teaching responsibilities at Université Grenoble Alpes and Grenoble INP.

To broaden the recruitment sphere, team members also teach at institutions beyond Grenoble, including ENS de Lyon, attracting talented students from diverse academic backgrounds to systems research.

In addition to their national teaching responsibilities, team members regularly conduct teaching missions abroad, particularly at partner universities in Cameroon such as University of Yaoundé 1 and ENSPY, where they deliver courses on operating systems, virtualization, and distributed systems.

#### 11.4.2 Supervision

The team supervised 18 PhD students in 2024, 3 postdocs, and more than 20 interns. KrakOS maintains a very open internship policy, recognizing that internships are an essential pathway for engaging students in systems research and cultivating the next generation of researchers in operating systems and distributed computing. The team also runs a mentoring program for students in Cameroon, providing guidance and support to students at partner universities such as University of Yaoundé 1 and ENSPY, helping them develop research skills and pursue advanced studies in computer systems.

## 12 Scientific production

### 12.1 Publications of the year

#### International journals

- [1] Y. Kone, L. Duval, R. Lachaize, P. Felber, D. Hagimont and A. Tchana. ‘Understanding Intel User Interrupts’. In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems*. Proceedings of the ACM on Measurement and Analysis of Computing Systems 9.2 (3rd June 2025), pp. 1–32. DOI: [10.1145/3727132](https://doi.org/10.1145/3727132). URL: <https://hal.science/hal-05115268>.

#### International peer-reviewed conferences

- [2] P. Breuil and B. Lepers. ‘P4CEMaker: automated hardware acceleration of consensus protocols’. In: ICDCS 2025 - IEEE 45th International Conference on Distributed Computing Systems. Glasgow, United Kingdom: IEEE, 21st July 2025, pp. 681–691. DOI: [10.1109/ICDCS63083.2025.00072](https://doi.org/10.1109/ICDCS63083.2025.00072). URL: <https://inria.hal.science/hal-05317064>.

- [3] B. Ekane, D. Mvondo, R. Lachaize, Y.-D. Bromberg, A. Tchana and D. Hagimont. ‘DISC: Backpressure Mitigation In Multi-tier Applications With Distributed Shared Connection’. In: *Proceedings of the 22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*. NSDI 2025 - 22nd USENIX Symposium on Networked Systems Design and Implementation. Philadelphia (Pennsylvania), United States, 28th Apr. 2025, pp. 55–70. URL: <https://hal.science/hal-05051628>.
- [4] K. Ishiguro, K. Hayama, A. Yokoyama, H. Yamada and T. Hirotsu. ‘Nesting Overlay File Systems with ShadowWhiteout’. In: *APSys ’25: 16th ACM SIGOPS Asia-Pacific Workshop on Systems*. Seoul, South Korea: ACM, 11th Oct. 2025, pp. 15–21. DOI: [10.1145/3725783.3764408](https://doi.org/10.1145/3725783.3764408). URL: <https://hal.science/hal-05474746>.
- [5] G. Job Antunes Grabher, F. Machida and T. Ropars. ‘Modeling Anomaly Detection in Cloud Services: Analysis of the Properties that Impact Latency and Resource Consumption’. In: *Proceedings of the IEEE/ACM 18th International Conference on Utility and Cloud Computing*. UCC 2025 - IEEE/ACM 18th International Conference on Utility and Cloud Computing. Nantes, France, 2025, pp. 1–10. DOI: [10.1145/3773274.3774269](https://doi.org/10.1145/3773274.3774269). URL: <https://hal.science/hal-05374713>.
- [6] Y. Kone, L. Duval, R. Lachaize, P. Felber, D. Hagimont and A. Tchana. ‘Understanding Intel User Interrupts’. In: *Abstracts of the 2025 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. SIGMETRICS 2025 - ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. Vol. 53. 1. Stony Brook, United States: ACM, June 2025, pp. 127–129. DOI: [10.1145/3726854.3727326](https://doi.org/10.1145/3726854.3727326). URL: <https://hal.science/hal-05156424>.
- [7] J. Nibler and T. Ropars. ‘Carbon Footprint of Storage in Data Centers: the Impact of Using Ssds for Key-Value Stores’. In: *2025 IEEE 25th International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Proceedings*. CCGrid 2025 - IEEE 25th International Symposium on Cluster, Cloud and Internet Computing. Tromsø, Norway: IEEE, 19th May 2025, pp. 305–314. DOI: [10.1109/CCGRID64434.2025.00064](https://doi.org/10.1109/CCGRID64434.2025.00064). URL: <https://hal.science/hal-05447447>.
- [8] A. Tchana, J. Gounou, P. A. Fall, R. Lachaize, H. Tapamo, C. P. B. Bessala and V. Quema. ‘Crazy : Unifying Memory and CPU Management Subsystems’. In: *APSys ’25: 16th ACM SIGOPS Asia-Pacific Workshop on Systems*. Seoul (Korea), South Korea: ACM, 11th Oct. 2025, pp. 1–7. DOI: [10.1145/3725783.3764398](https://doi.org/10.1145/3725783.3764398). URL: <https://hal.science/hal-05474726>.
- [9] X. Wu, B. Lepers and W. Zwaenepoel. ‘Pre-Stores: Proactive Software-guided Movement of Data Down the Memory Hierarchy’. In: <https://dl.acm.org/doi/pdf/10.1145/3689031.3696097>. EuroSys 2025 - Twentieth European Conference on Computer Systems. Rotterdam, Netherlands: ACM, 30th Mar. 2025, pp. 1161–1176. DOI: [10.1145/3689031.3696097](https://doi.org/10.1145/3689031.3696097). URL: <https://hal.science/hal-05149954>.

## Reports & preprints

- [10] I. Fajjari, Y. Nzali Koagne, J. Soares and V. Marangozova. *D 4.2-1 Use Case Prototypes*. Université Grenoble Alpes; Orange Innovation; Orange Direction EOLAS, Mar. 2025. URL: <https://hal.science/hal-05288437>.

## 12.2 Cited publications

- [11] P. Ambati, Í. Goiri, F. Frujeri, A. Gun, K. Wang, B. Dolan, B. Corell, S. Pasupuleti, T. Moscibroda, S. Elnikety, M. Fontoura and R. Bianchini. ‘Providing SLOs for resource-harvesting VMs in cloud platforms’. In: *Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation*. OSDI’20. USA: USENIX Association, 2020 (cit. on p. 11).
- [12] J. T. Humphries, N. Natu, A. Chaugule, O. Weisse, B. Rhoden, J. Don, L. Rizzo, O. Rombakh, P. Turner and C. Kozyrakis. ‘ghOSt: Fast & Flexible User-Space Delegation of Linux Scheduling’. In: *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*. SOSP ’21. Virtual Event, Germany: Association for Computing Machinery, 2021, pp. 588–604. DOI: [10.1145/3477132.3483542](https://doi.org/10.1145/3477132.3483542). URL: <https://doi.org/10.1145/3477132.3483542> (cit. on p. 10).

- [13] J. Liu, A. Rebello, Y. Dai, C. Ye, S. Kannan, A. C. Arpaci-Dusseau and R. H. Arpaci-Dusseau. ‘Scale and Performance in a Filesystem Semi-Microkernel’. In: *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles. SOSP ’21*. Virtual Event, Germany: Association for Computing Machinery, 2021, pp. 819–835. doi: [10.1145/3477132.3483581](https://doi.org/10.1145/3477132.3483581). URL: <https://doi.org/10.1145/3477132.3483581> (cit. on p. 10).
- [14] M. Marty, M. de Kruijf, J. Adriaens, C. Alfeld, S. Bauer, C. Contavalli, M. Dalton, N. Dukkipati, W. C. Evans, S. Gribble, N. Kidd, R. Kononov, G. Kumar, C. Mauer, E. Musick, L. Olson, E. Rubow, M. Ryan, K. Springborn, P. Turner, V. Valancius, X. Wang and A. Vahdat. ‘Snap: a microkernel approach to host networking’. In: *Proceedings of the 27th ACM Symposium on Operating Systems Principles. SOSP ’19*. Huntsville, Ontario, Canada: Association for Computing Machinery, 2019, pp. 399–413. doi: [10.1145/3341301.3359657](https://doi.org/10.1145/3341301.3359657). URL: <https://doi.org/10.1145/3341301.3359657> (cit. on p. 10).
- [15] A. Tchana, D. Goepf, S. Bitchebe and R. Lachaize. ‘xOS: The End Of The Process-Thread Duo Reign’. In: *Proceedings of the 14th ACM SIGOPS Asia-Pacific Workshop on Systems. APSys ’23*. Seoul, Republic of Korea: Association for Computing Machinery, 2023, pp. 1–8. doi: [10.1145/3609510.3609817](https://doi.org/10.1145/3609510.3609817). URL: <https://doi.org/10.1145/3609510.3609817> (cit. on p. 10).
- [16] D. Tegui, J. Chen, S. Bitchebe, O. Balmau and A. Tchana. ‘vPIM: Processing-in-Memory Virtualization’. In: *Proceedings of the 25th International Middleware Conference. Middleware ’24*. Hong Kong, Hong Kong: Association for Computing Machinery, 2024, pp. 417–430. doi: [10.1145/3652892.3700782](https://doi.org/10.1145/3652892.3700782). URL: <https://doi.org/10.1145/3652892.3700782> (cit. on p. 21).
- [17] S. Zhao, X. Ding, W. Xu and D. Gu. ‘Seeing Through The Same Lens: Introspecting Guest Address Space At Native Speed’. In: *Security Symposium (USENIX Sec’17)*. USENIX, 2017 (cit. on p. 12).