Contributions à la sécurité à la frontière logicielle/matérielle

Soutenance d'Habilitation à Diriger des Recherches

Pascal Cotret

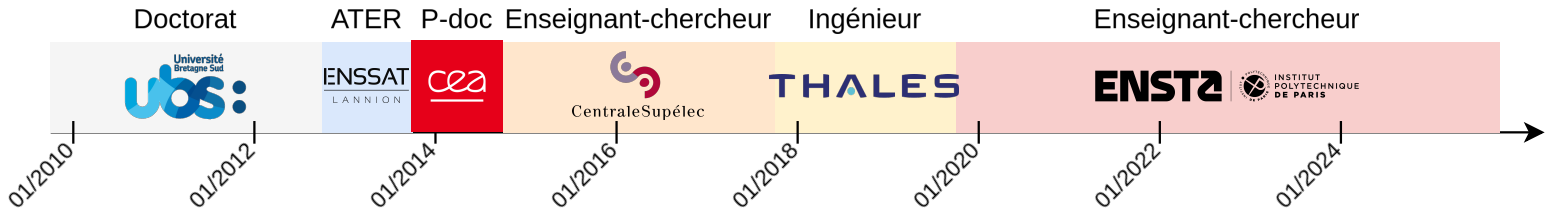
Lab-STICC, ENSTA campus de Brest

27 novembre 2025

Table des matières

À propos

1. Parcours



ANR SecReSoC

Cominlabs HardBlare

PEC Secure VMs

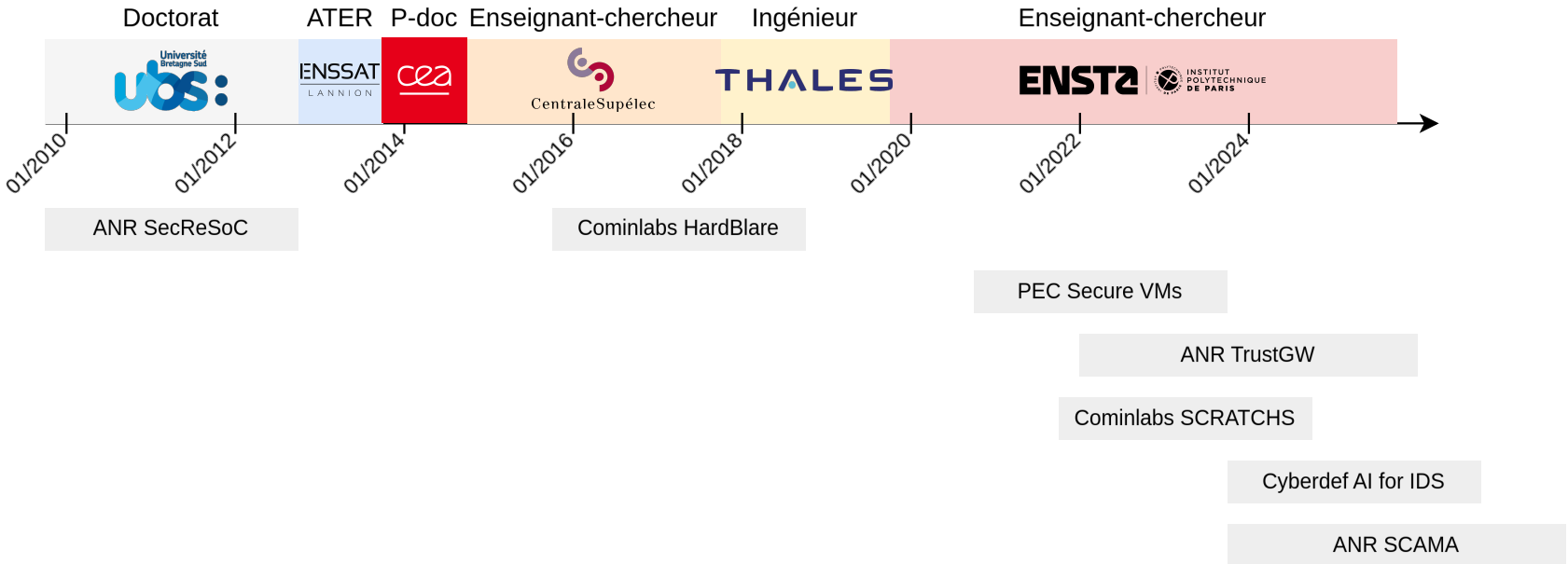
ANR TrustGW

Cominlabs SCRATCHS

Cyberdef AI for IDS

ANR SCAMA

1. Parcours

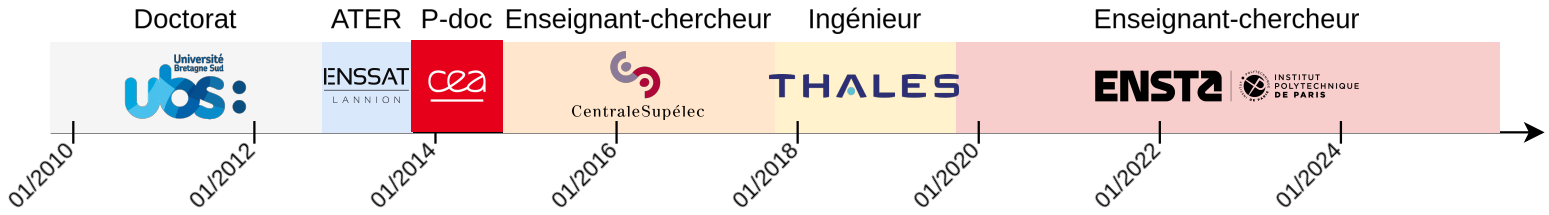


Thales

- Radio-communications militaires[1]
- Activités non développées dans la soutenance

1. Thales Group. *Thales inaugure à Cholet son nouveau centre d'excellence en radiocommunications, pour répondre aux besoins croissants des forces armées.* `https://www.thalesgroup.com/fr/actualites-du-groupe/communiques-de-presse/thales-inaugure-cholet-son-nouveau-centre-dexcellence-en`. Communiqué de presse. Thales Group, juin 2025.

1. Parcours



ANR SecReSoC

Cominlabs HardBlare

PEC Secure VMs

ANR TrustGW

Cominlabs SCRATCHS

Cyberdef AI for IDS

ANR SCAMA

Activités présentées

CentraleSupélec Rennes et ENSTA Brest

Les publications associées aux projets sont en **ocre**

Contexte

2. Contexte

Cybersécurité, un problème majeur

- Systèmes vulnérables très nombreux
- Activités civiles et militaires

Cible : systèmes embarqués

- Objets connectés, systèmes industriels ou cyber-physiques, etc.
- Des systèmes complexes :
  - Microcontrôleurs et RTOS (ARM Cortex-M)
  - Système avec OS type Linux ou Android et processeurs plus complexes (ARM Cortex-A, x86)




2. Contexte

RISC-V

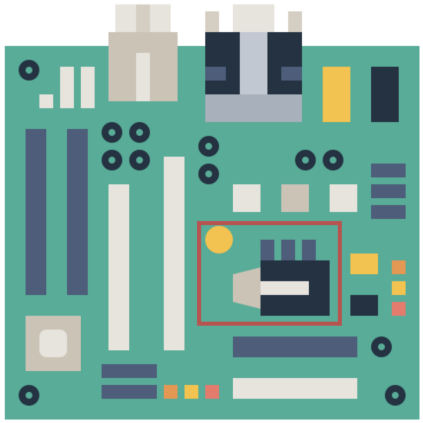
- Des industriels [1] : SiFive, Codasip, Keysom...
- Applications diverses : IA [Gau25], audio/vidéo, sécurité

Sécurité matérielle et architecture [2]

- Sur le logiciel : compilateur, environnement de confiance
- Sur le matériel :
  - ✓ Extensions du jeu d'instructions, protection mémoire, etc.
  - ✗ EM, acoustique ou par injection de fautes

1. Jon Gold (adapté par Jean Elyan). "5 géants des puces s'allient autour de RISC-V pour s'affranchir d'ARM". In : *Le Monde Informatique* (2023). URL : `https://www.lemondeinformatique.fr/actualites/lire-5-geants-des-puces-s-allient-autour-de-risc-v-pour-s-affranchir-d-arm-91215.html`.
2. Tao Lu. "A Survey on RISC-V Security : Hardware and Architecture". In : *CoRR* abs/2107.04175 (2021). arXiv : `2107.04175`. URL : `https://arxiv.org/abs/2107.04175`.

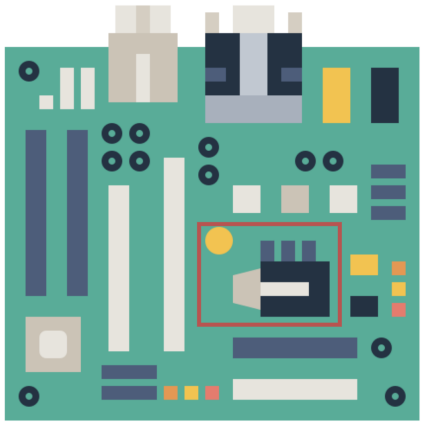
2. Contexte

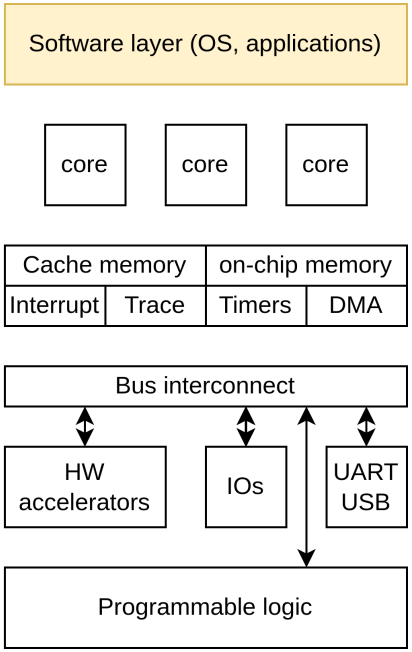


Système embarqué

- Une couche logicielle : système d'exploitation, applications
- Une couche matérielle : processeurs, mémoires, périphériques. . .

2. Contexte



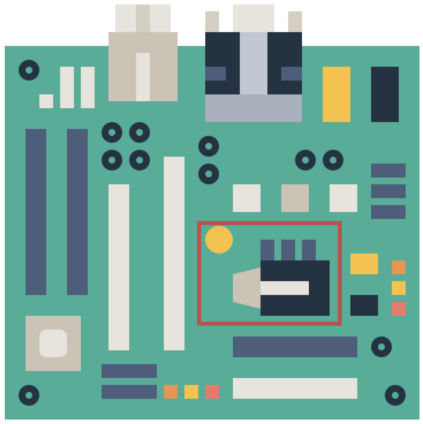


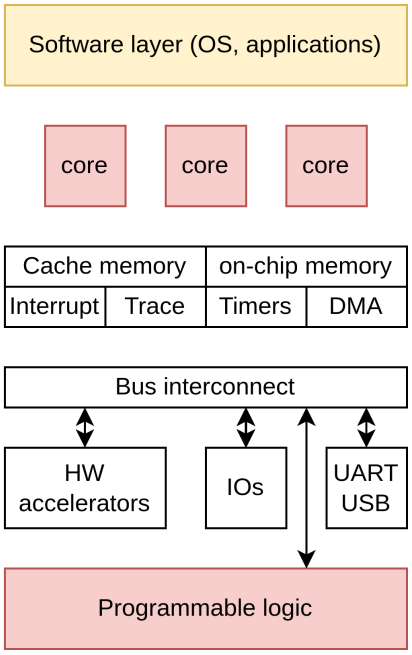
Système embarqué

- Une couche logicielle : système d'exploitation, applications
- Une couche matérielle : processeurs, mémoires, périphériques. . .

Compromis entre logiciel et matériel pour améliorer la sécurité globale du système

2. Contexte



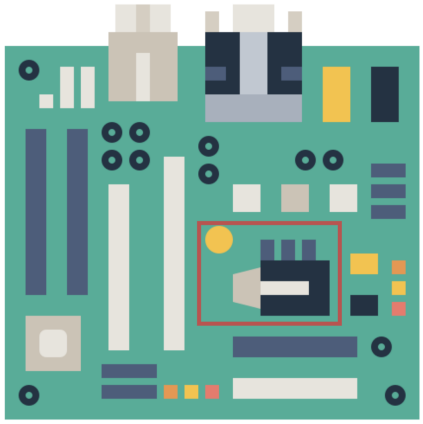


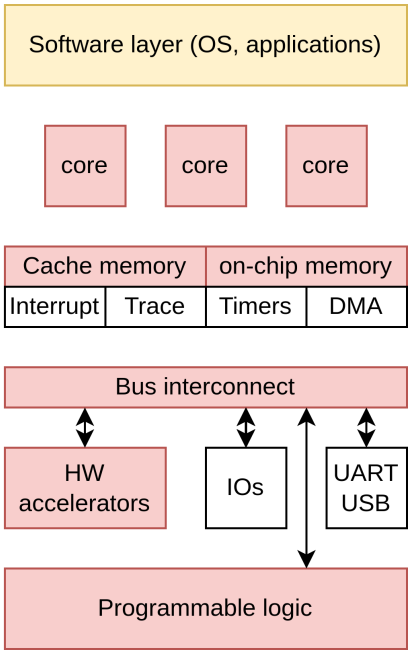
Axe 1

Comment le matériel peut aider à la sécurité logicielle ?

- Cominlabs HardBlare
- PEC VM sécurisée

2. Contexte



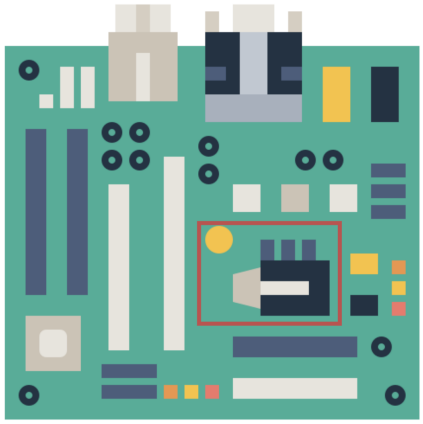


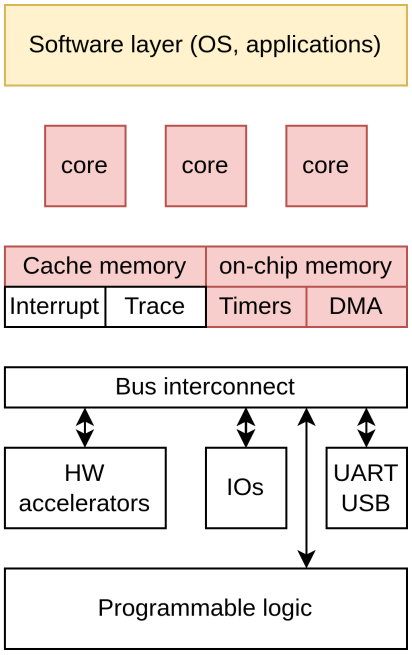
Axe 2

Extensions matérielles pour des applications de sécurité

- ANR TrustGW
- Chaire cyberdéf navale

2. Contexte





Axe 3

Sécurité au niveau de la microarchitecture

- Cominlabs SCRATCHS
- ANR SCAMA

Activités de recherche

3.1. Comment le matériel peut aider à la sécurité logicielle ?

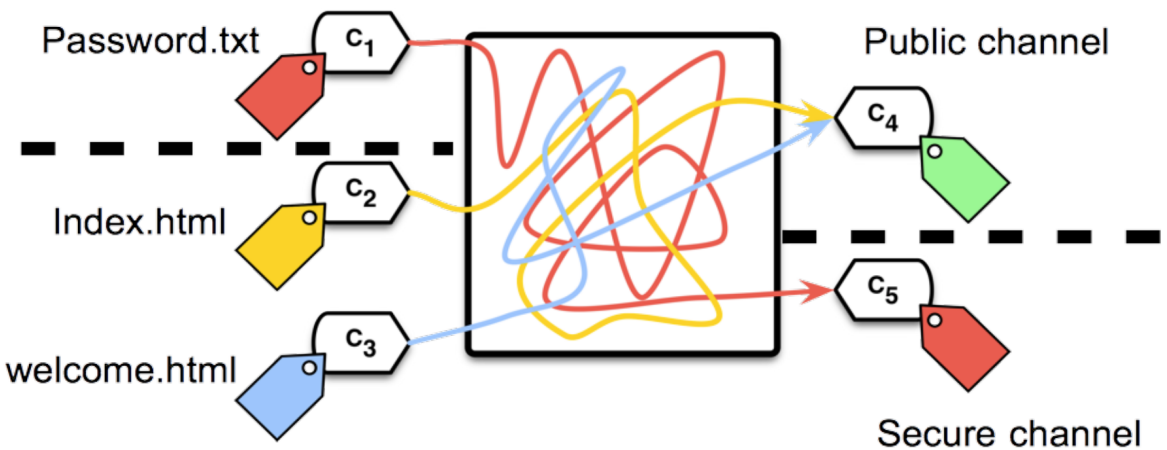
3.1. DIFT : suivi de flot de contrôle

Motivation

Une approche générique pour détecter des attaques sur la confidentialité et l'intégrité des données à différents niveaux

Différents niveaux de DIFT

- On attache des labels (ou **tags**) à des données et on spécifie une politique de suivi d'information, une relation entre les tags
- Á l'exécution, on propage les tags et on détecte les éventuelles violation de la politique mise en place



3.1. Différents niveaux de DIFT

Gros grain : niveau OS

- Fichiers, pages mémoire

✓ Teinte plus simple, surcoût temporel

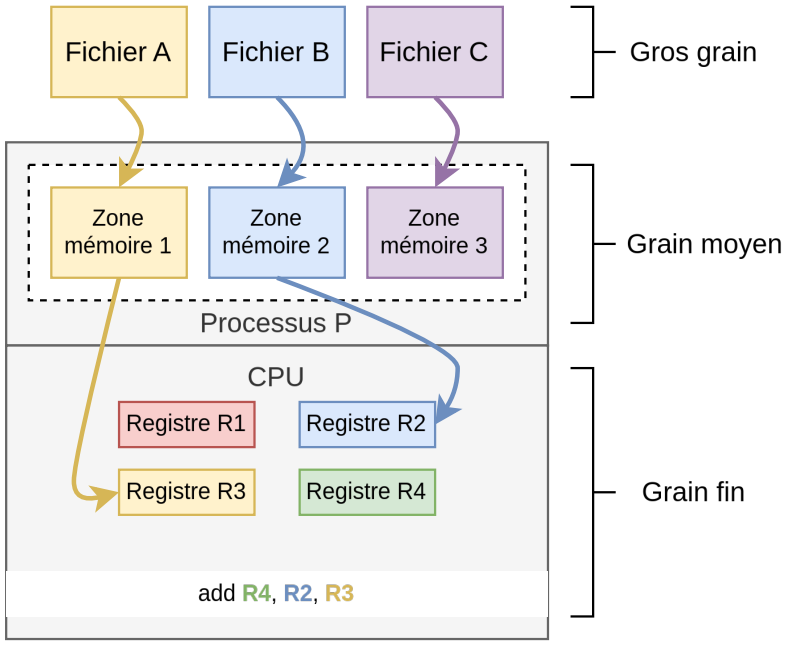
✗ Sur-approximation

Grain fin : niveau langage machine

- Registres, mots mémoire

✓ Suivi précis

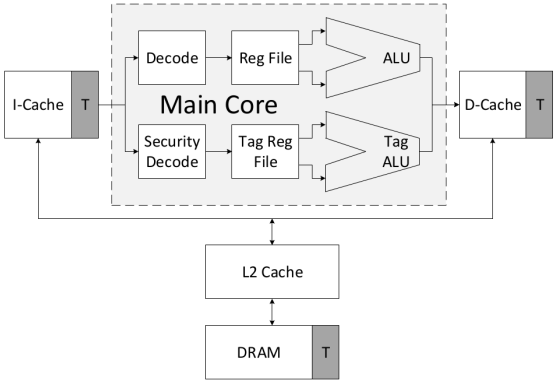
✗ Surcoût temporel

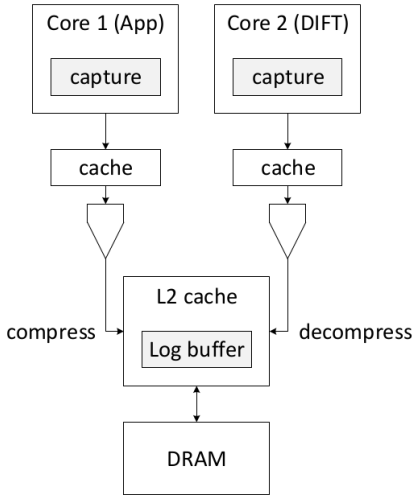


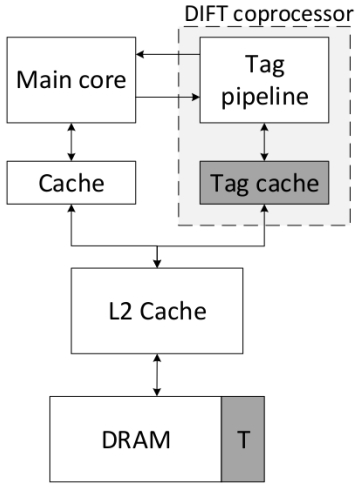
Approche

- Combinaison des deux niveaux
- Moniteur de sécurité en matériel

3.1. DIFT assisté par matériel







In-core [DKK07; Dha+15]

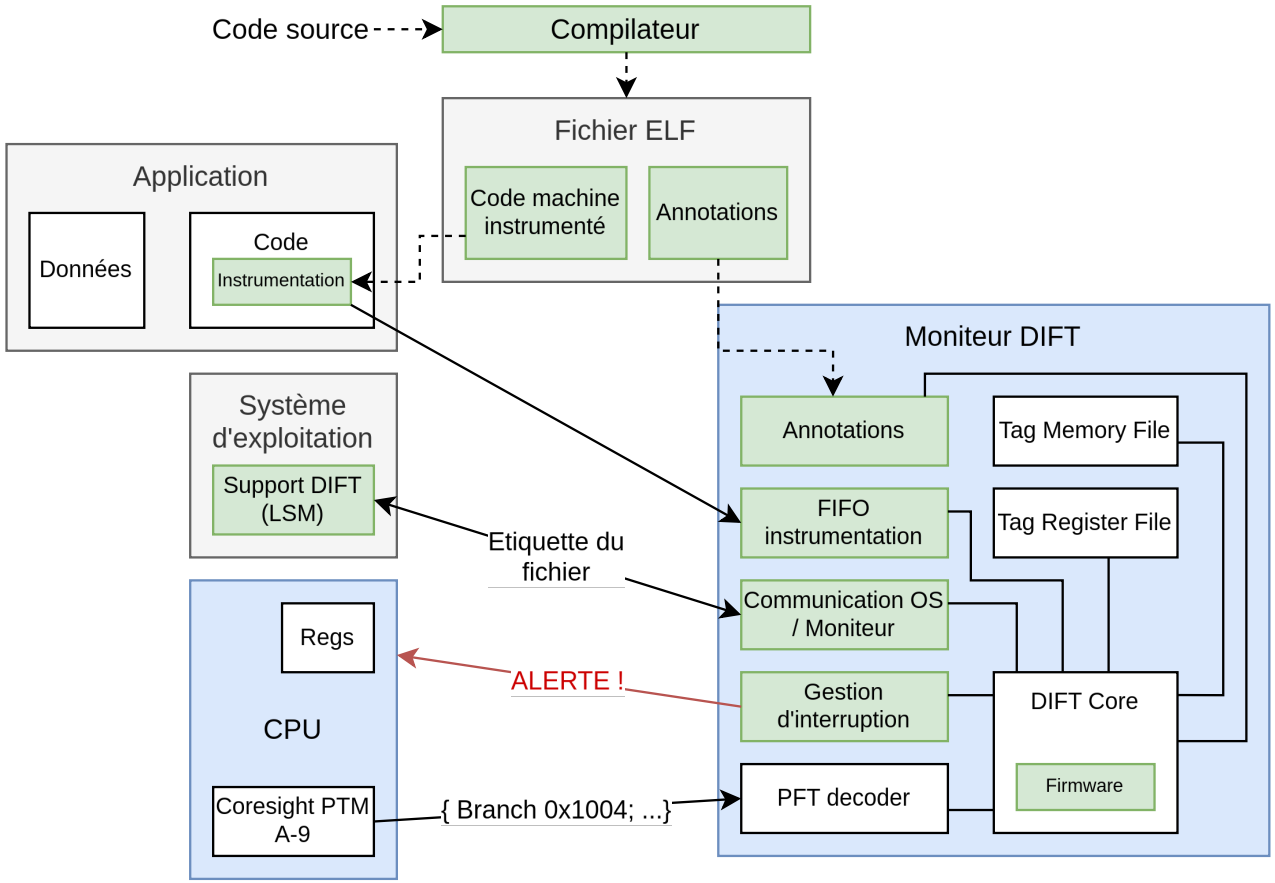
Offloading [Che+08; Ruw+08]

Off-core [KDK09; Heo+15; Lee+16]

Approche	Approche	Cible	Portabilité *hardcore*	Isolation moniteur
[DKK07; Dha+15]	In-core	LEON3	✗	✗
[Che+08; Ruw+08]	Offloading	LEON3	✗	✗
[KDK09]	Off-core	LEON3	✗	✗
[Heo+15; Lee+16]	Off-core	LEON3	✓	✗
HardBlare [Wah+17; Wah+18b]	Off-core	ARM	✓	✓

3.1. HardBlare : une approche off-core pour ARM

FPGA Zynq (Zedboard) + OS basé sur Yocto



- Composants de debug CoreSight (PTM, ETM,. . .).
- Modification **logicielles** à plusieurs niveaux.
- Moniteur DIFT **matériel**.

3.1. Comparaison avec d'autres approches *off-core*

	Raksha [KDK09]	FlexCore [Den+10]	Heo et al. [Heo+15]	FPL'17 [Wah+17]	AsianHOST'18 [Wah+18b]
Surface	+6.4%	+14.8%	+14.5%	**+0.47%**	+0.95%
Consommation	N/A	**6.3%**	24%	16%	16.2%
Communication	N/A	N/A	60%	**5.4%**	335%
Hardcore	✗	✗	✗	✓	✓
Multi-tâches	✗	✗	✗	✗	✓

- ✓ Flexible car différentes politiques peuvent être configurées
- ✓ Pas de sur-approximation
- ✓ Supporter le suivi de plusieurs applications en même temps
- ✓ TEE TrustZone pour complètement isoler le moniteur

Architectures

- ✗ Dépendance aux évolutions de l'architecture ARM [3]
- ✓ RISC-V : Raft (RAID'23 [Wan+23]), Palmiero et al. (HPEC'18 [Pal+18])

3. `https://developer.arm.com/Architectures/CoreSight%20Architecture`

3.1. Vulnérabilités des navigateurs

En **octobre 2025**, le CERT-FR[1] a levé plusieurs alertes majeures sur des vulnérabilités découvertes dans plusieurs navigateurs web.

AVIS DU CERT-FR

Objet: Vulnérabilité dans Google Chrome

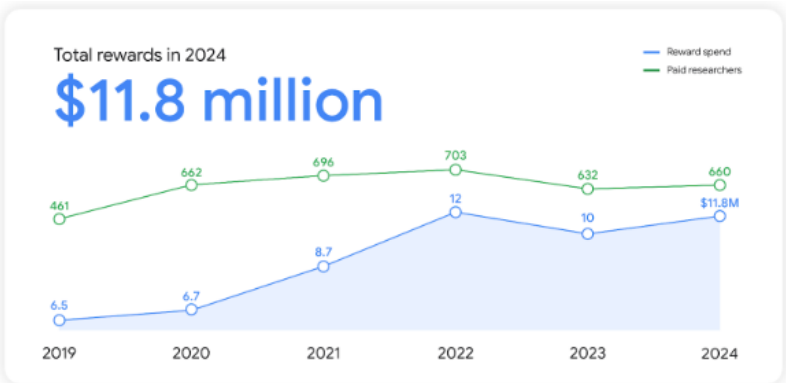
GESTION DU DOCUMENT

Référence	CERTFR-2025-AVI-0875
Titre	Vulnérabilité dans Google Chrome
Date de la première version	15 octobre 2025
Date de la dernière version	15 octobre 2025
Source(s)	Bulletin de sécurité Google Chrome du 14 octobre 2025

Objet: Multiples vulnérabilités dans Microsoft Edge

GESTION DU DOCUMENT

Référence	CERTFR-2025-AVI-0862
Titre	Multiples vulnérabilités dans Microsoft Edge
Date de la première version	10 octobre 2025
Date de la dernière version	10 octobre 2025
Source(s)	Bulletin de sécurité Microsoft Edge CVE-2025-11458 du 09 octobre 2025 Bulletin de sécurité Microsoft Edge CVE-2025-11460 du 09 octobre 2025

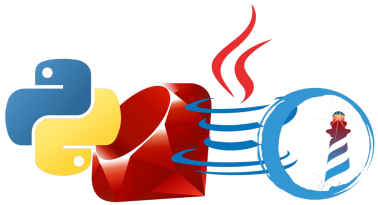


Le **“bug bounty program”**[2] en 2024 a pu récompenser **660** chercheurs dont **$3,4M** que pour Chrome seulement.
Le lancement de **v8ctf** souligne une volonté de sécurisation de leur navigateur.

1. CERT-FR. *Avis de sécurité*. `https://www.cert.ssi.gouv.fr/avis/`. 2025
2. Google Security Blog. *Vulnerability Reward Program : 2024 in Review*. `https://security.googleblog.com/2025/03/vulnerability-reward-program-2024-in.html`. Mars 2025

3.1. Déploiement de machines virtuelles

Exécution JS web



Support langage

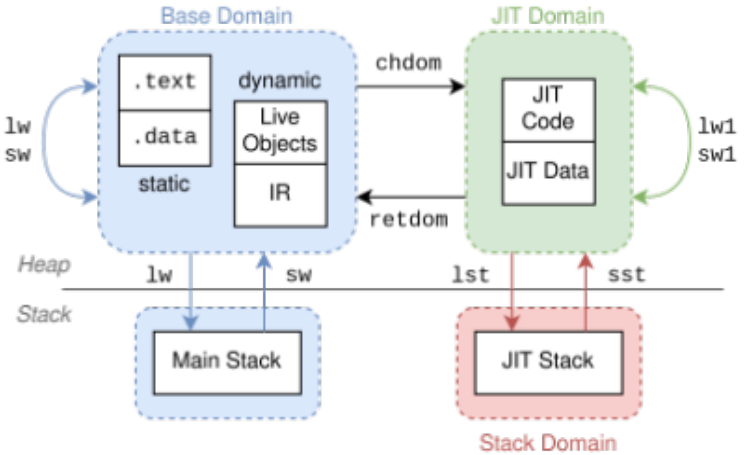
Android

Distribution APK

Déploiement de machines virtuelles (VMs) pour supporter l'exécution d'applications
$\Rightarrow$ Milliards de systèmes d'information !

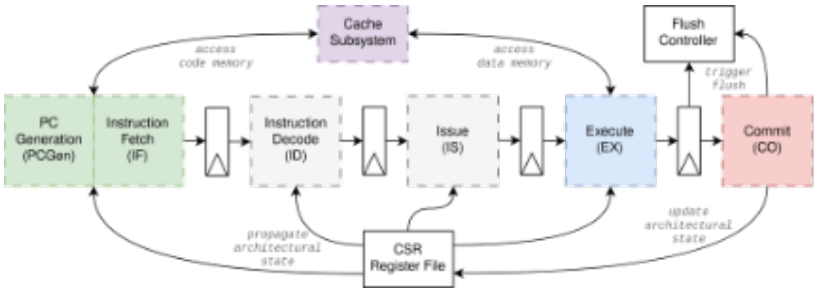
Une vulnérabilité dans ces systèmes compromet les données de millions d'utilisateurs et donne un accès direct à leur système d'information

3.1. Solution d'isolation



Garanties :

- Séparation de l'**accès aux données**
- Contrôle des échanges de **flot d'exécution**
- **"Shadow-stack"** pour le code JIT
- Contrôle des **appels système**



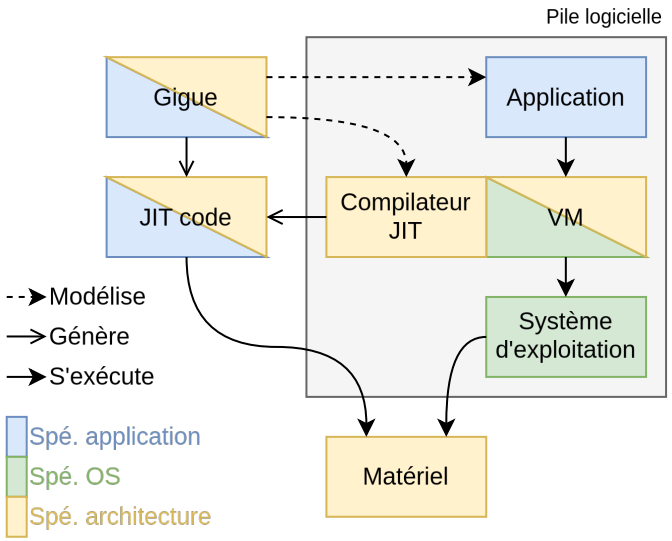
Processeur CVA6 :

- Architecture RISC-V open-source
- Maintenu par l'OpenHW Group

OPENHW®

THALES

3.1. Comment évaluer la solution d'isolation ?



Générateur de charges aléatoire suivant une distribution prédéfinie et un modèle d'exécution semblable au JIT

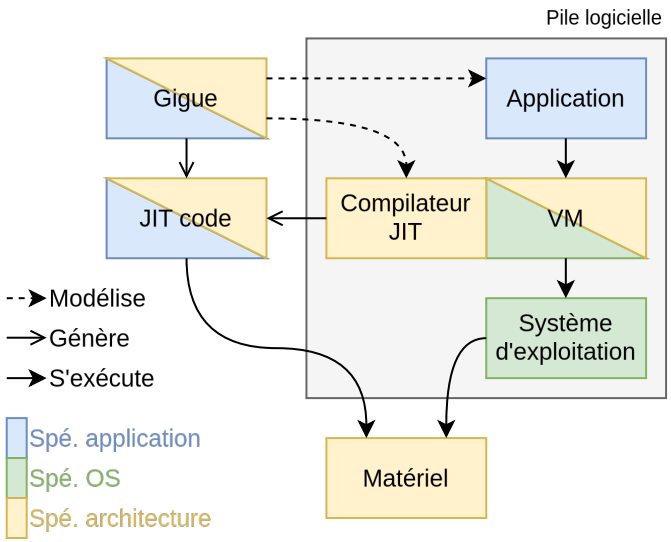
Objectifs :

- Paramétrisation des binaires pour qualifier une application/VM
- Instrumentation des binaires générés avec des instructions dédiées
- Interfaçage avec des processeurs open-source existants

Validation :

- Vérification des binaires avant leur distribution
- Extension de l'environnement d'exécution aux nouvelles instructions

3.1. Comment évaluer la solution d'isolation ?

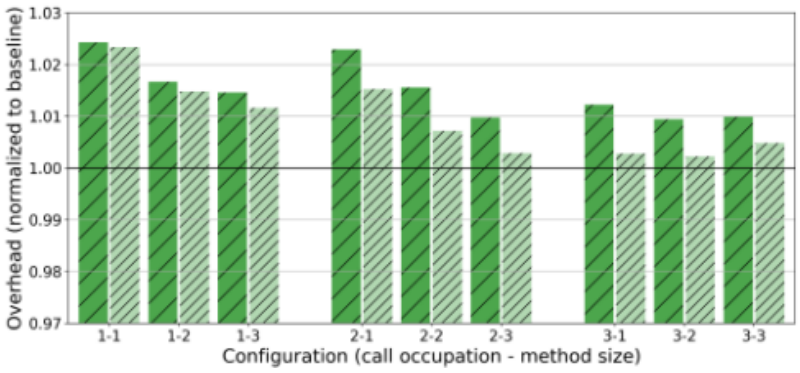


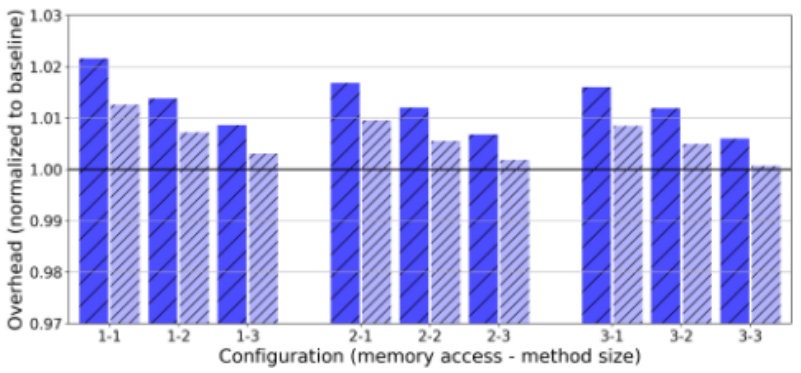
Générateur de charges aléatoire suivant une distribution prédéfinie et un modèle d'exécution semblable au JIT

Accès pour le développeur matériel à des exemples concrets pour tester la solution en simplifiant la pile technologique [1]

1. Quentin Ducasse et al. "Gigue : A JIT Code Binary Generator for Hardware Testing". In : *2023 Workshop on Virtual Machines and Language Implementations*. Oct. 2023

3.1. Vers une VM sécurisée





Surcoûts :

$< 2.5\%$ pénalité en performance

$< 1\%$ ressources matérielles

Densité des appels	Accès mémoire	Taille des méthodes
1%	4%	400 octets
3%	12%	600 octets
6%	20%	800 octets

3.1. Axe 1 - Synthèse

Comment le matériel a pu aider à la sécurité logicielle ?

DIFT assisté par le matériel sur ARM [Wah+17 ; Wah+18a ; Wah+18b]

- Perspectives sur RISC-V [Wan+23] et attaques en fautes associées [Pen24]
- Spécifications de traces RISC-V [RIS25]

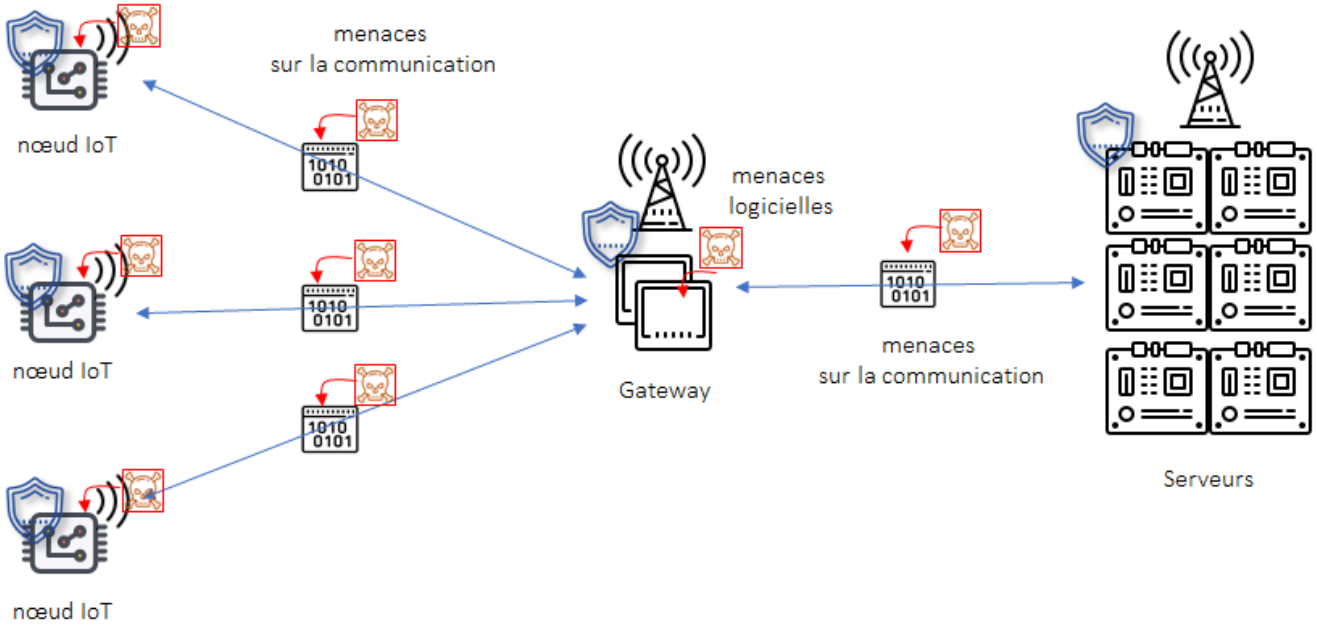
Compilation à la volée et RISC-V [4] [DCL25 ; DCL23 ; Duc+22]

- Sécurité multi-domaine
- Intégration dans de futurs travaux (projet ANR SCAMA)

4. Outils open-source sur `https://github.com/QDucasse/` : `gigue`, `jitdomain-tests`,`cva6-jitdomain`

3.2. Extensions matérielles pour des applications de sécurité

3.2. Cas #1 - TrustGW : une passerelle sécurisée



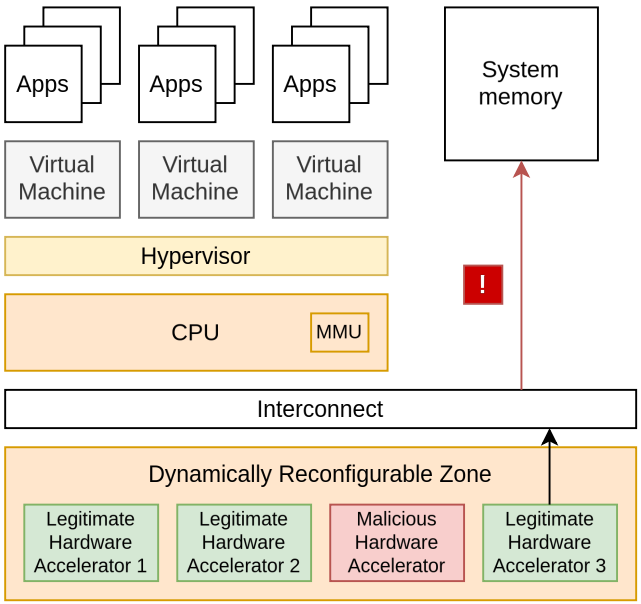
Environnement

- Processeur RISC-V CVA6 (+ extension H)
- Hyperviseur Bao [5]

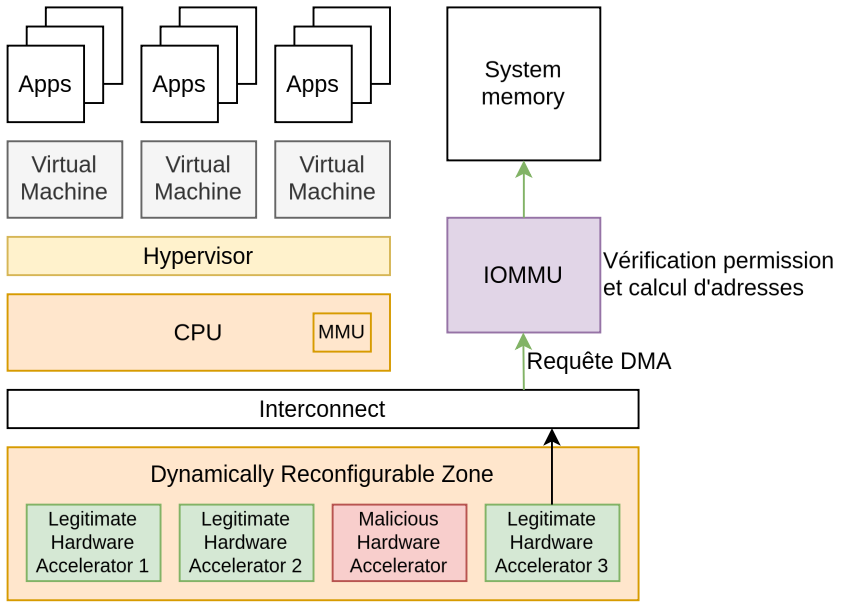
Quelles sont les menaces sur le SoC de la passerelle ?

5. Travaux de l'université de Minho : `https://github.com/bao-project/bao-hypervisor`

3.2. Cas #1 - Utilisation d'une IOMMU

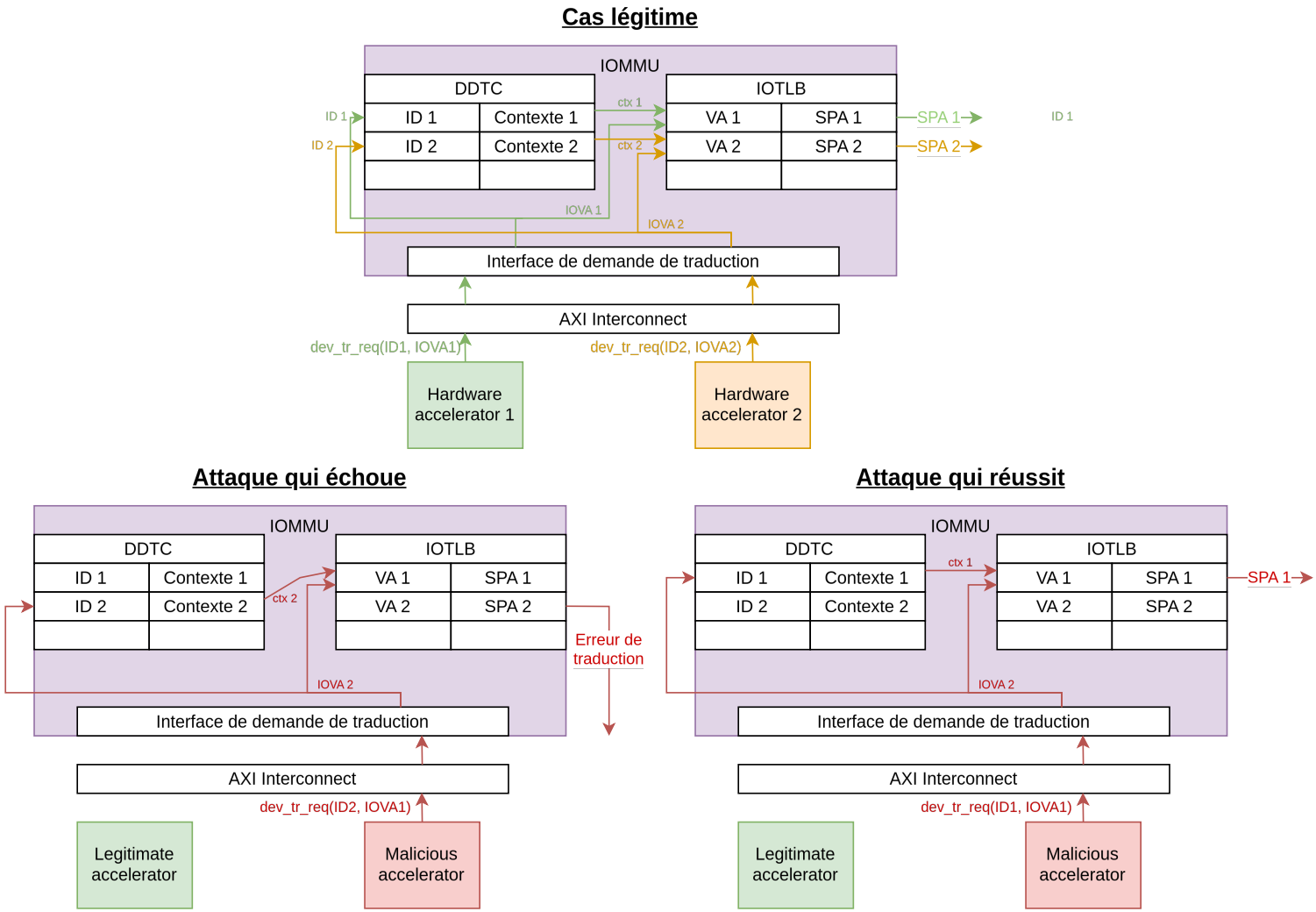


Architecture sans IOMMU



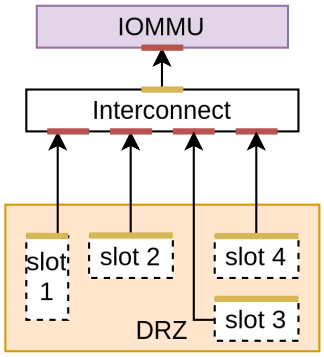
Architecture avec IOMMU

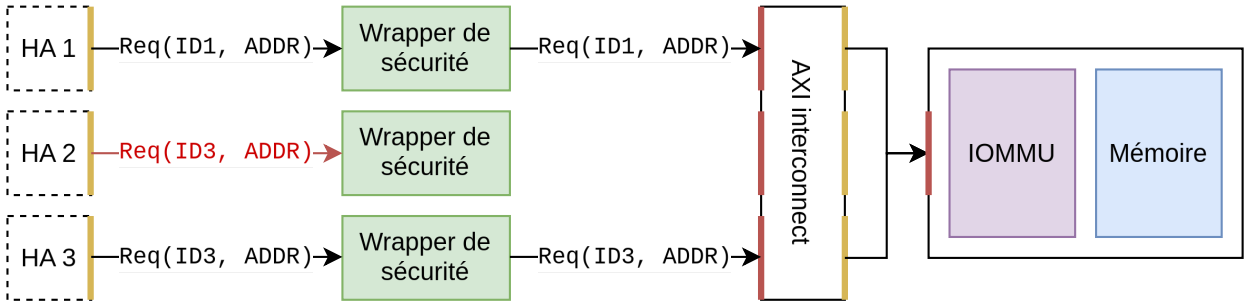
3.2. Cas #1 - Attaque par usurpation d'ID [Jen+25] [6]



6. Aya Jendoubi et al. "Security of Dynamically Reconfigurable RISC-V Systems : I/O Attack Focus". In : *39th Annual IEEE International Parallel & Distributed Processing Symposium (IEEE IPDPS 2025) : 32nd Reconfigurable Architecture Workshop*. Milan, Italy, juin 2025. URL : https://hal.science/hal-05117047.

3.2. Cas #1 - Contremesure

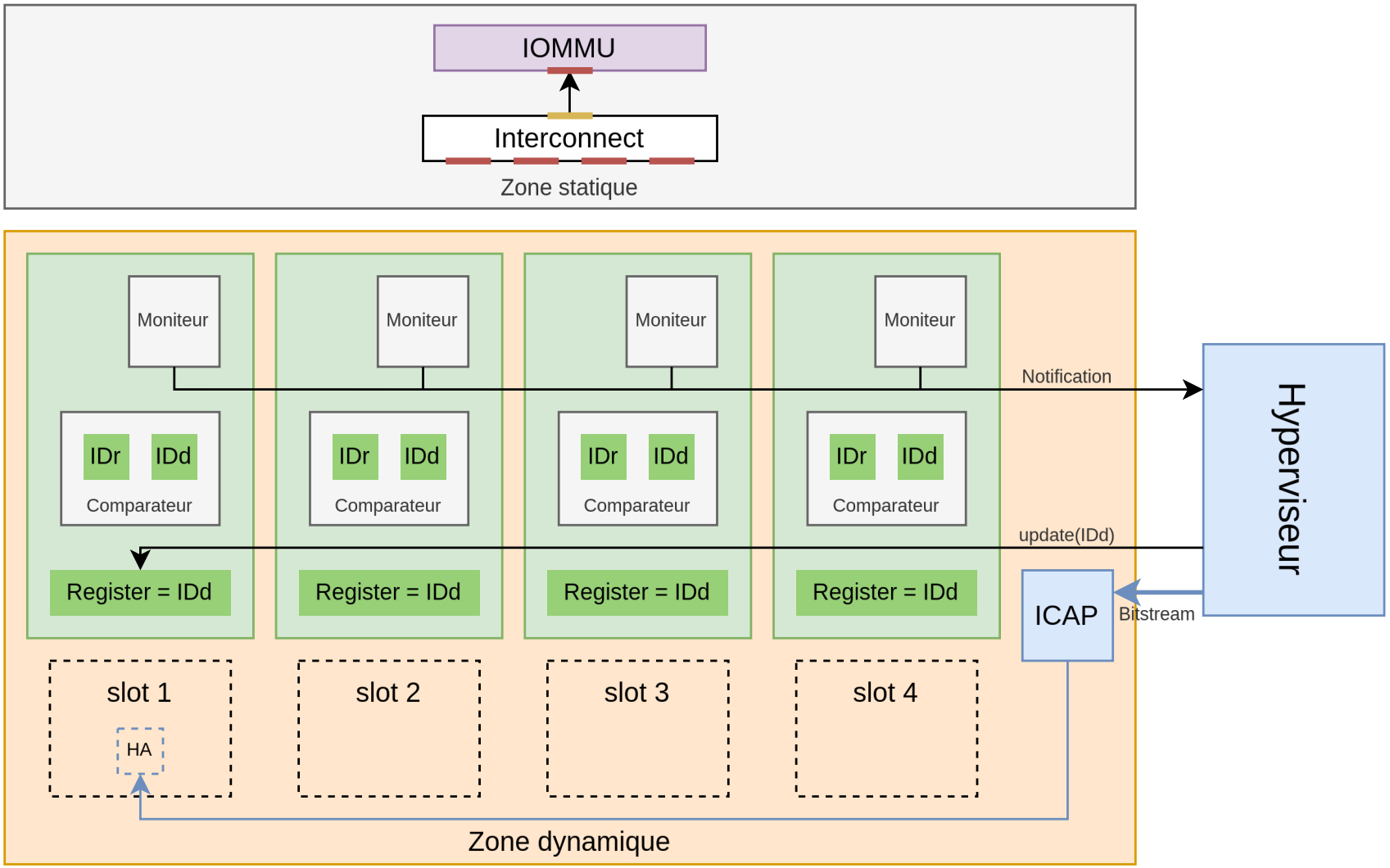




Objectifs

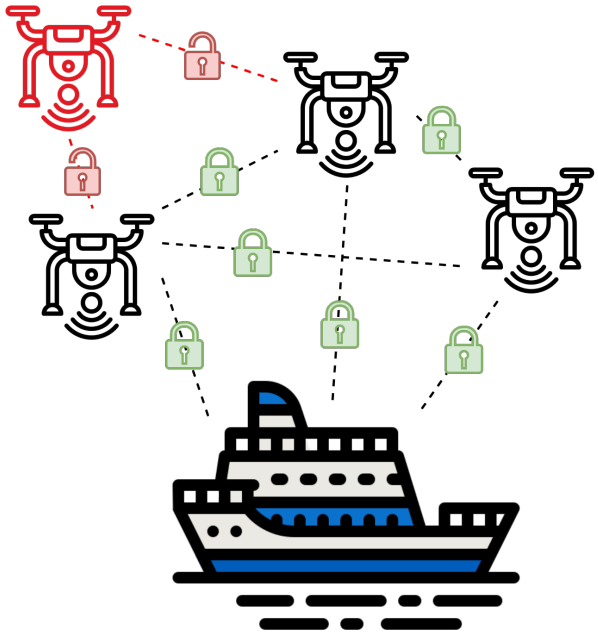
- Wrapper pour filtrer les accès légitimes ou non
- Surcoût temporel
- Comportement en cas d'accès illégitime ?

3.2. Cas #1 - Vérification d'authenticité [7]



7. Soumission en cours au workshop DASIP 2026. Jendoubi et al. "ARMOR : Accelerator Runtime Monitoring and cOntrolled identity enfoRcement"

3.2. Cas #2 - Matériel pour de l'IA efficiente



Travaux en collaboration avec Naval Group

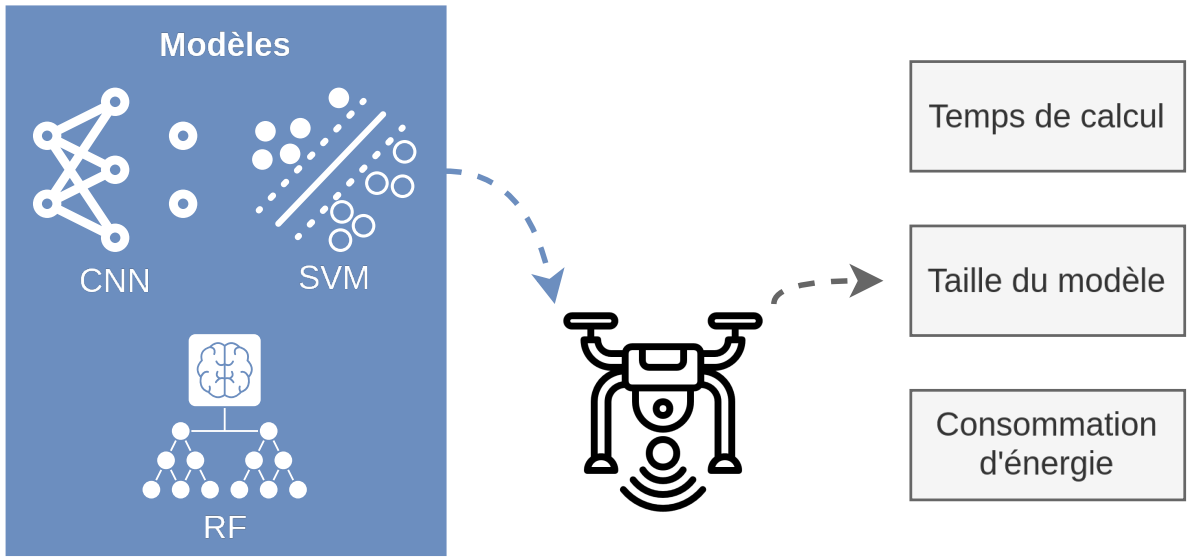
Principal objectif

Embarquer des modèles d'IA sur des drones

Plusieurs applications

Détection d'intrusion, détection d'objets sur images, etc.

3.2. Cas #2 - Obstacles technologiques [Gar+25] [8]

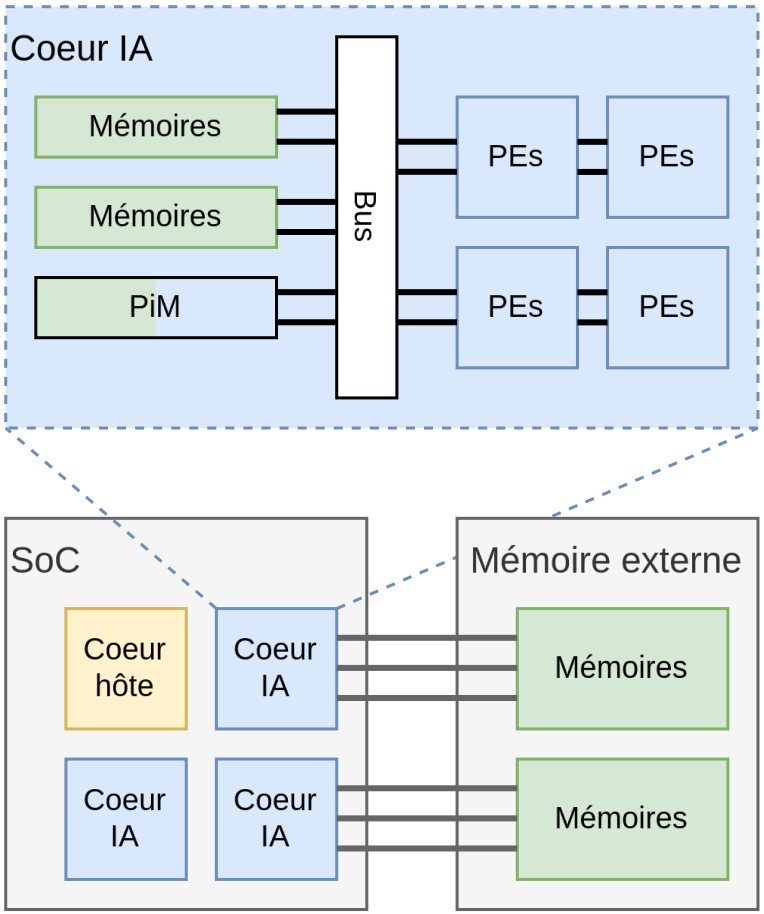


Question de recherche

Comment l'architecture matérielle électronique du drone peut-elle être modifiée pour améliorer l'efficacité de l'inférence des modèles d'IA ?

8. Pierre Garreau et al. "A survey on versatile embedded Machine Learning hardware acceleration". In : t. 167. Oct. 2025, p. 103501. DOI : https://doi.org/10.1016/j.sysarc.2025.103501. URL : https://www.sciencedirect.com/science/article/pii/S1383762125001730.

3.2. Cas #2 - Architecture matérielle embarquée



Question 1

Comment ordonnancer une charge de travail IA sur une architecture matérielle optimisée avec des ressources restreintes ?

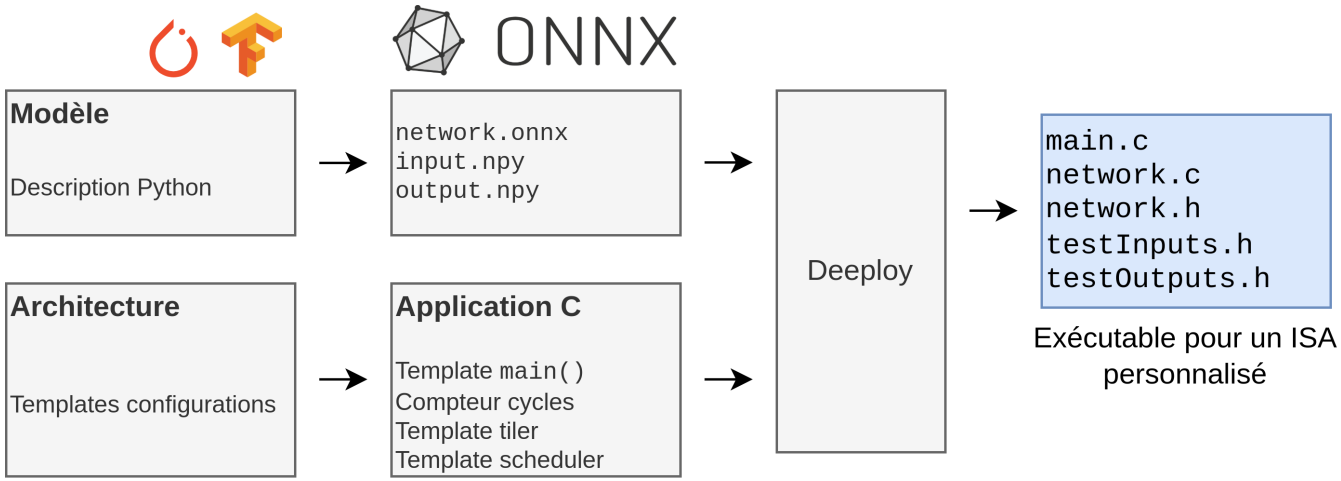
Question 2

Comment utiliser le contexte du drone en temps réel pour ordonnancer la charge de travail IA ?

- SoC avec cluster de calcul (cf. Pulpissimo [9])

9. https://github.com/pulp-platform/pulpissimo

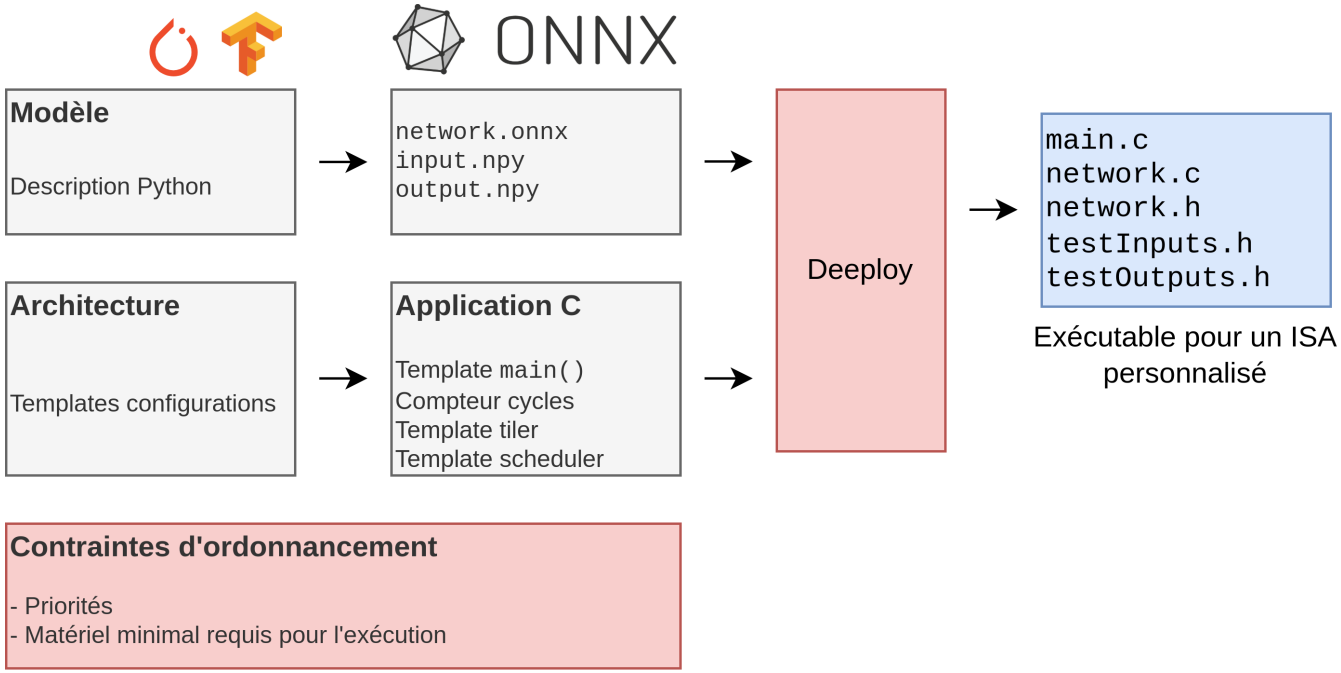
Cas #2 - Modèles d'IA sur système embarqué RISC-V



Du modèle Python à l'exécutable pour ISA personnalisé grâce à Deeploy [10]

10. Philip Wiese et al. "Toward Attention-based TinyML : A Heterogeneous Accelerated Architecture and Automated Deployment Flow". In : *IEEE Design & Test* (2025), p. 1-1. DOI : 10.1109/MDAT.2025.3527371.

Cas #2 - Modèles d'IA sur système embarqué RISC-V

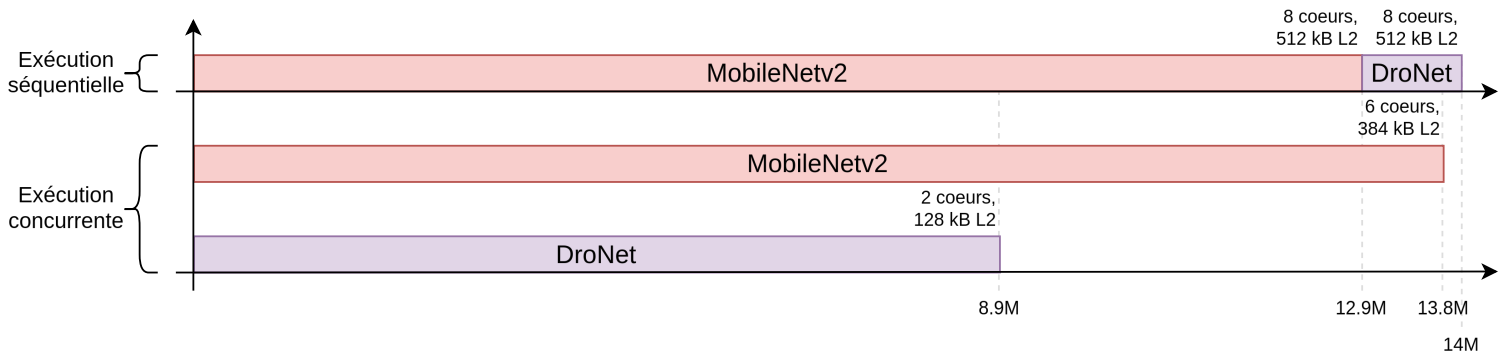


Modifications apportées à Deeploy pour exécuter plusieurs modèles

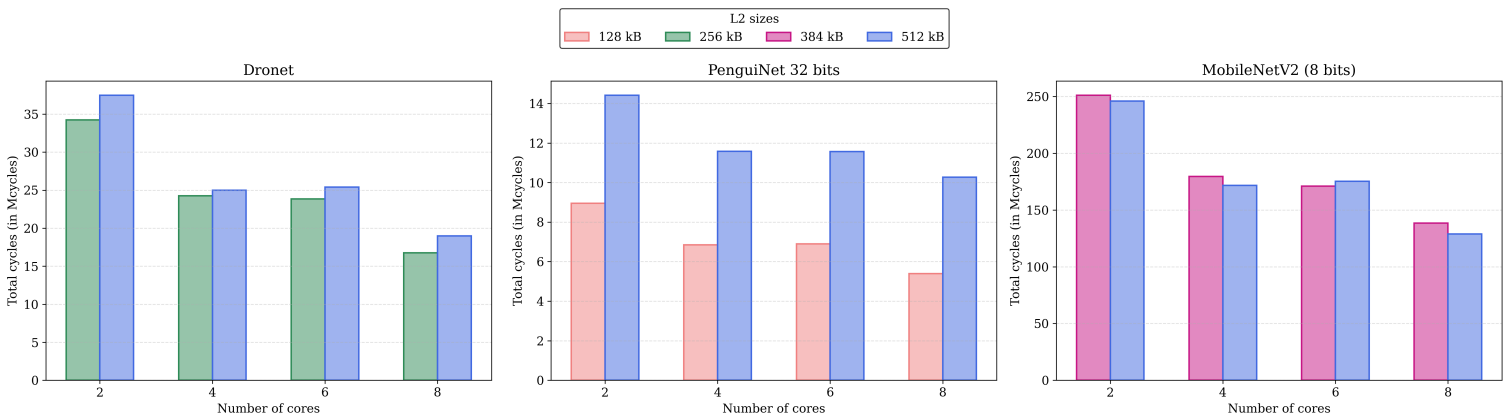
Critères de décision :

- c_i : nombre de coeurs utilisés
- T_i : cycles pour charger et exécuter le modèle
- P_i : priorité de la tâche
- $L1_i$ et $L2_i$: les contraintes mémoire pour chaque tuile (entrées, sorties, poids)

Cas #2 - Exécution multi-modèles [11]



Exécution concurrente de MobileNetV2 (8-bits quantized) et DroNet sur 8 coeurs et 512 kB de mémoire L2



Exécution de Dronet, PenguiNet et MobileNetV2

11. Soumission en cours à ISCAS 2026. Garreau et al. “Priority-Aware Scheduling of Multi-Model, Multi-Precision DNN Inference on Resource-Constrained Embedded Multi-Cores”

3.2. Axe 2 - Synthèse

Hyperviseurs [Jen+25]

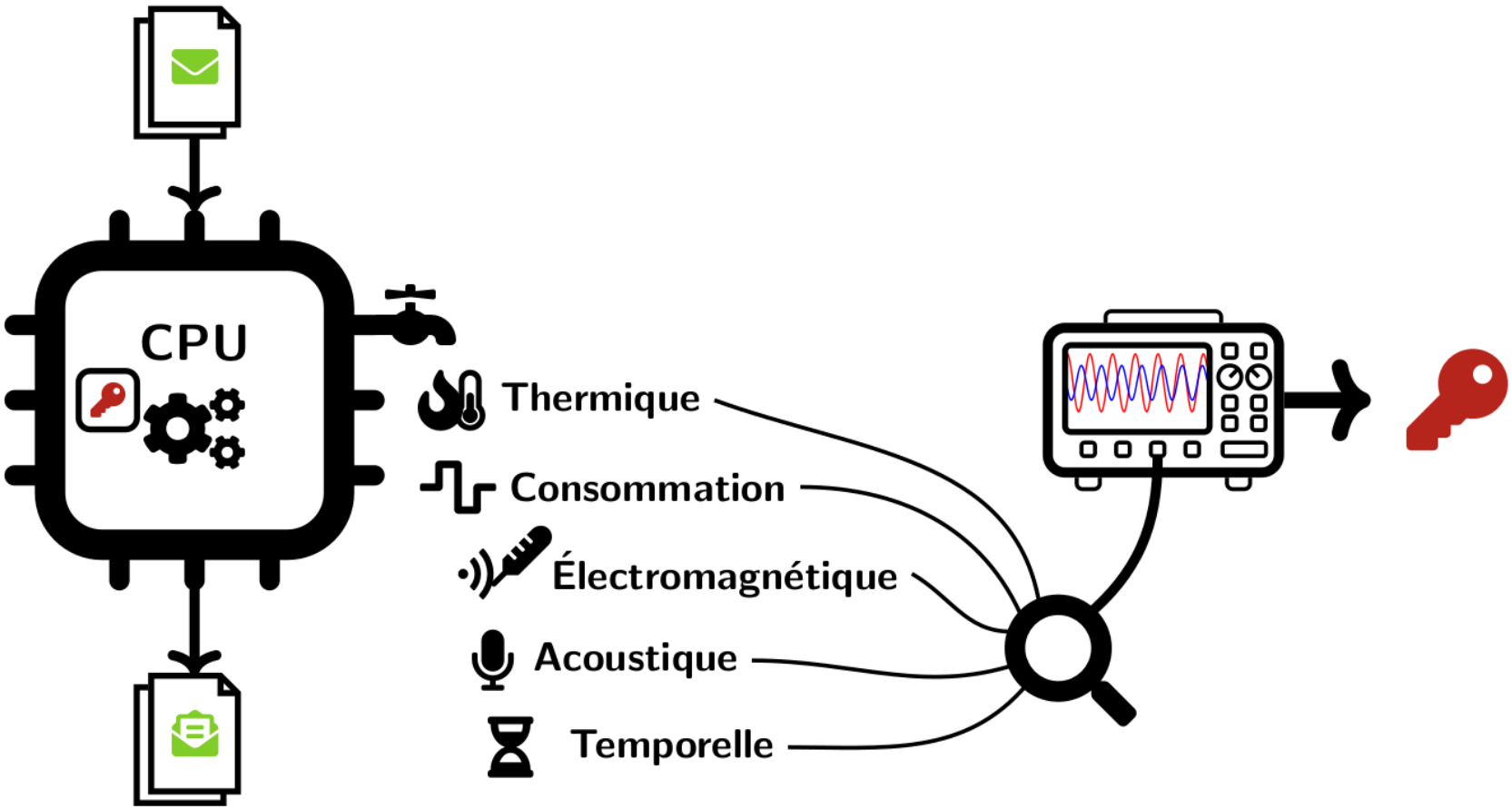
- Faiblesses dans le protocole AXI [Zon+25]
- Intégration du wrapper de sécurité dans Bao

Accélérateurs IA [Gar+25]

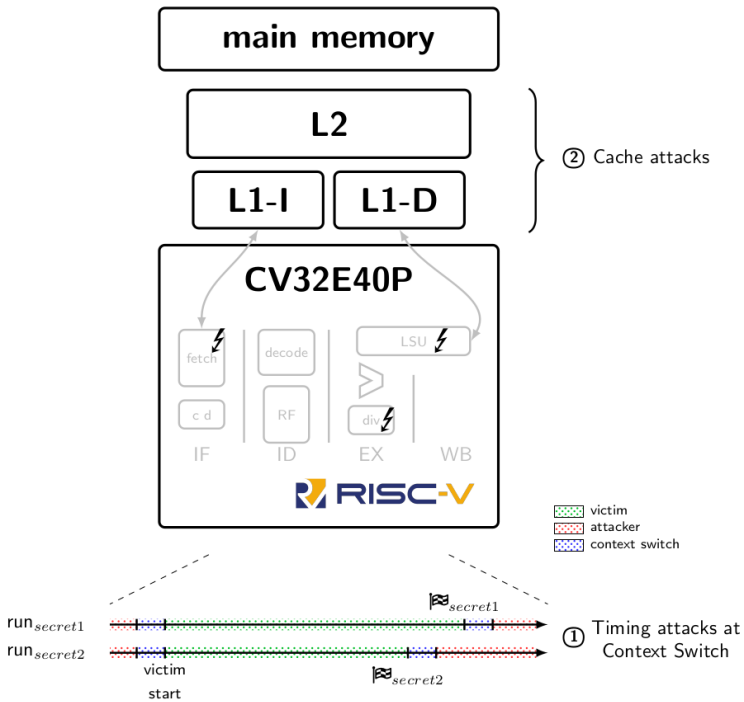
- Application à des datasets IDS type UAV-NIDD [Had+25]

3.3. Sécurité au niveau de la microarchitecture

3.3. Attaques par canaux auxiliaires



3.3. Fuites temporelles



Sources de fuites exploitées par 🕵 :

Branchements :
`if (condition(secret))`

Opérations à temps variable :
`dividend/secret;`

Index pour accès mémoire :
`array[secret];`

Les données stockées dans les mémoires cache

Basés sur la randomisation (ScrambleCache [JHN21], ScatterCache [Wer+19])

- ✓ Autonome
- ✗ Besoin de mettre à jour la sécurité (et donc d'invalider le cache)

Partitionnement à gros grain (NoMoCache [Dom+12])

- ✓ Support de sécurité pour l'OS et les applications
- ✗ Peut fortement affecter les performances

Contremesures

Conçues en tenant compte de systèmes complexes

- Ne correspondent pas aux exigences/possibilités des systèmes embarqués
- Se concentrent souvent sur le Last Level Cache (grand cache partagé)

Pas directement compatibles avec les systèmes embarqués

Partitionnement à grain fin (PLcache [WL07])

- Apporte un support de sécurité pour l'OS et les applications
- Impacte légèrement les performances

PLcache

Un bon candidat pour notre mécanisme de sécurité

- Introduit de nouvelles instructions permettant de réserver des lignes de cache
- S'adapte aux exigences et aux contraintes des systèmes embarqués

3.3. SCRATCHS - Mécanisme de verrouillage

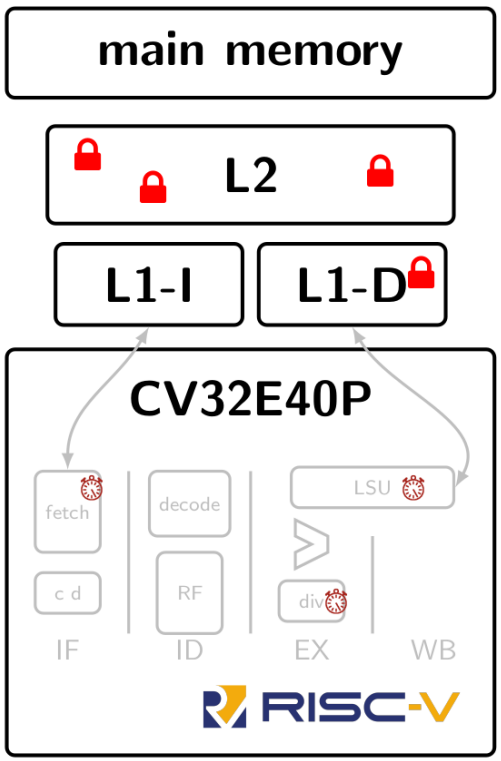
Projet Cominlabs SCRATCHS

- Side-Channel Resistant Applications Through Co-designed Hardware/Software
- Assurer une exécution en temps constant, efficace et à la demande

3.3. SCRATCHS - Mécanisme de verrouillage

Projet Cominlabs SCRATCHS

- Side-Channel Resistant Applications Through Co-designed Hardware/Software
- Assurer une exécution en temps constant, efficace et à la demande



Extension du jeu d'instructions :

- Instructions `lock` et `unlock`

🔒 `lock` garde la ligne dans le cache :

- Garantie d'un accès à temps constant
- La ligne verrouillée ne peut pas être évincée
- Atténue Evict+Time et Prime+Probe

🔓 `unlock` libère la ligne verrouillée

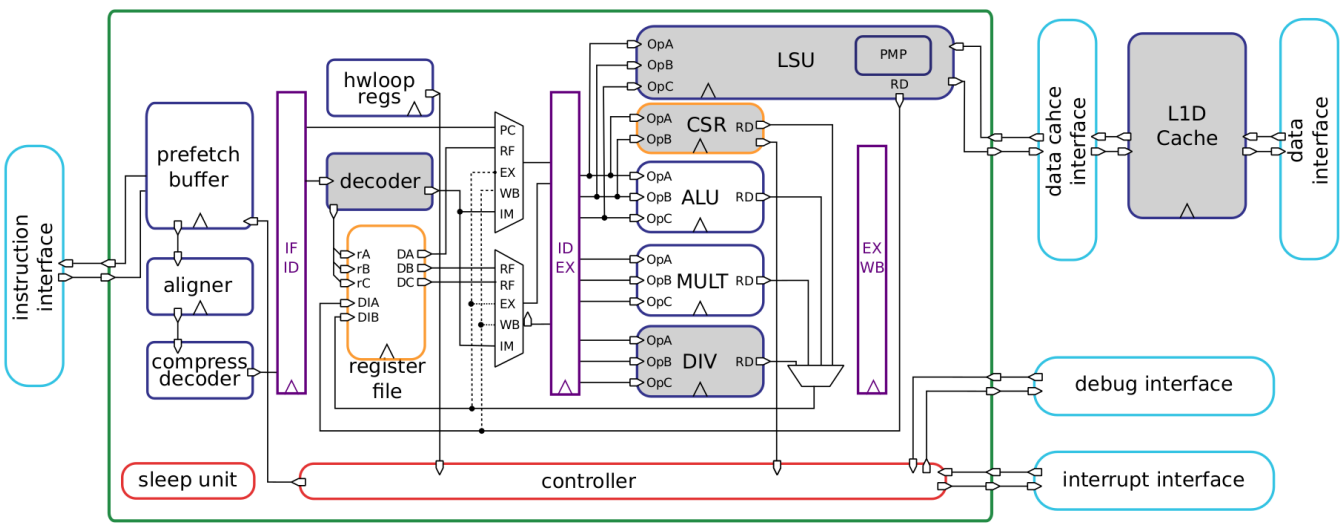
- La donnée peut être évincée

3.3. Implémentation du mécanisme de lock - Cache *set-associative*

Primitives logicielles :

```
1  void fct(int* sensitive_table, int* input){
2    // verrouillage
3    for(int i=0;i<sizeof(sensitive_table);i+=16)
4      __LOCK(&sensitive_table, i);
5
6    algo(sensitive_table,input);
7    // déverrouillage
8    for(int i=0;i<sizeof(sensitive_table);i+=16)
9      __UNLOCK(&sensitive_table, i);
10 }
```

Modifications matérielles :

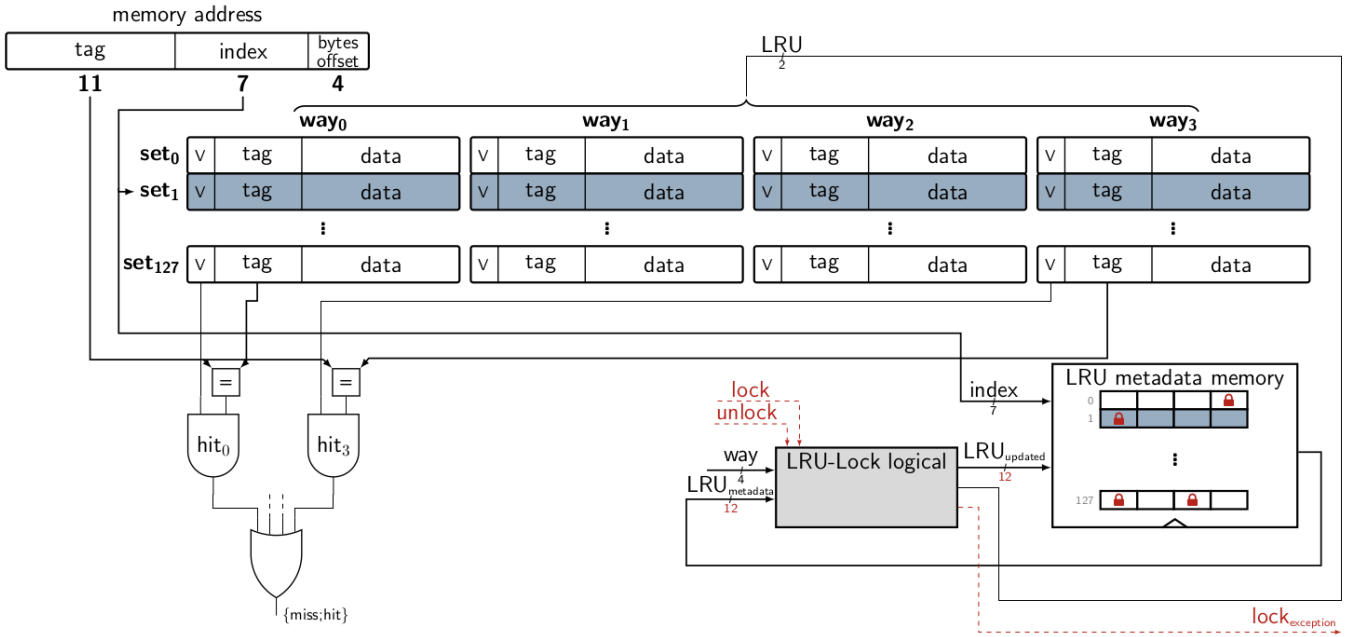


3.3. Implémentation du mécanisme de lock - Cache *set-associative*

Primitives logicielles :

```
1  void fct(int* sensitive_table, int* input){
2    // verrouillage
3    for(int i=0;i<sizeof(sensitive_table);i+=16)
4      __LOCK(&sensitive_table, i);
5
6    algo(sensitive_table,input);
7    // déverrouillage
8    for(int i=0;i<sizeof(sensitive_table);i+=16)
9      __UNLOCK(&sensitive_table, i);
10 }
```

Modifications matérielles :



3.3. Implémentation du mécanisme de lock - Cache *set-associative* [Gau+24] [12]

En résumé :

- Faible surcoût en surface (<3% sur le cache)
- Impact faible sur la performance globale
- Le mécanisme de verrouillage offre une sécurité **fine** et **à la demande** contre les attaques par analyse temporelle

Une combinaison avec un cache aléatoire peut-elle repousser les limites ?

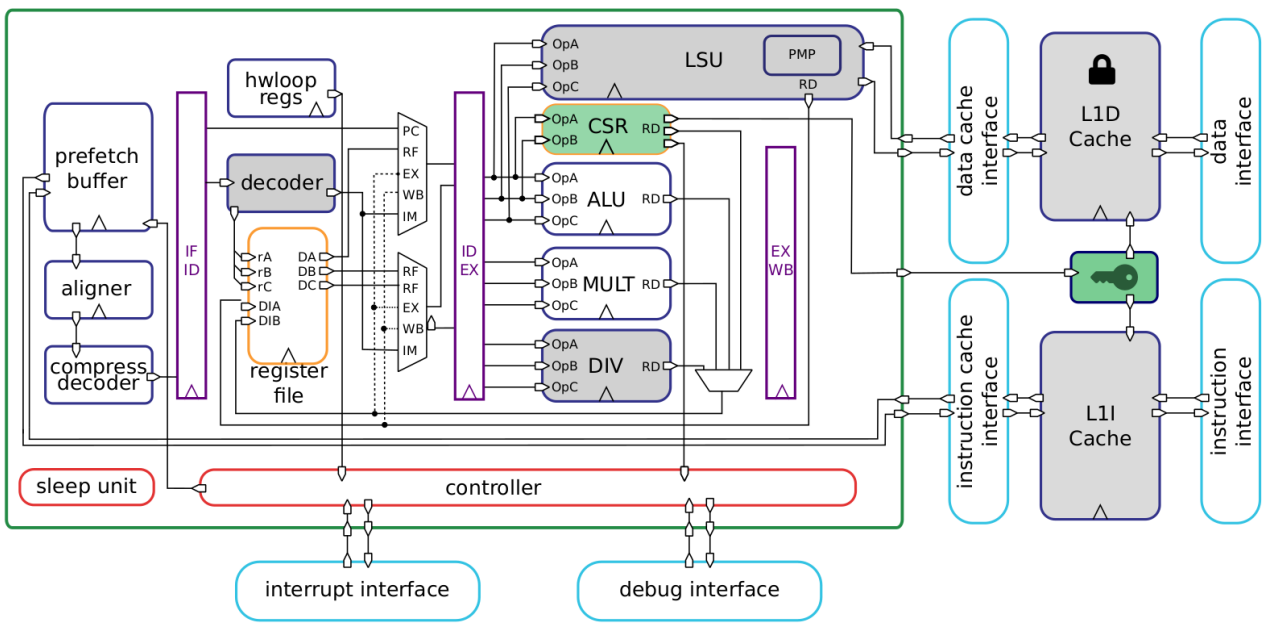
- Identifier le nombre exact de lignes de cache verrouillées
- Limiter l'utilisation du verrouillage
- Renouveler fréquemment les clés afin de maintenir la sécurité

12. Nicolas Gaudin et al. "A Fine-Grained Dynamic Partitioning Against Cache-Based Timing Attacks via Cache Locking". In : *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. Juill. 2024.

3.3. Extension avec un cache randomisé [Pet+24] [13]

- Se prémunir d'une nouvelle attaque qui construit un ensemble d'éviction (comme dans Prime+Prune+Probe)
- Fournir des garanties de sécurité lorsqu'un ensemble d'éviction est en place
  - Garder les applications critiques invulnérables
  - Empêcher l'attaquant d'inférer le nombre de lignes de cache verrouillées

Modifications matérielles :



13. Moritz PETERS et al. "On The Effect of Replacement Policies on The Security of Randomized Cache Architectures". In : *19th ACM ASIA Conference on Computer and Communications Security (ACM ASIACCS 2024)*. Juill. 2024.

3.3. Résumé sur le cache randomisé

En résumé :

- Fournir des **garanties de sécurité** pour les applications critiques
  - Cache aléatoire biaisé pour l'ensemble des applications
  - Mécanisme de verrouillage réservé aux applications critiques
- **Les applications critiques restent immunisées** contre les attaques basées sur le temps
- Cette implémentation permet de se défendre contre de **nouvelles techniques** de construction d'ensembles d'éviction
- L'implémentation du mécanisme de verrouillage implique un **faible surcoût** en surface (<2,5% sur le cache)

3.3. Résumé sur le cache randomisé

En résumé :

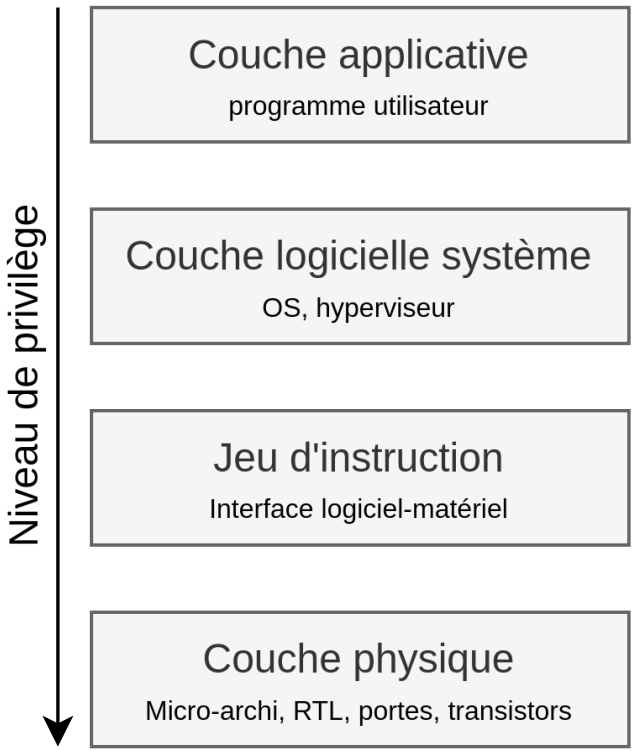
- Fournir des **garanties de sécurité** pour les applications critiques
  - Cache aléatoire biaisé pour l'ensemble des applications
  - Mécanisme de verrouillage réservé aux applications critiques
- **Les applications critiques restent immunisées** contre les attaques basées sur le temps
- Cette implémentation permet de se défendre contre de **nouvelles techniques** de construction d'ensembles d'éviction
- L'implémentation du mécanisme de verrouillage implique un **faible surcoût** en surface (<2,5% sur le cache)

Quels sont les attaques microarchitecturales potentielles sur l'environnement logiciel ?

3.3. Attaques par canaux auxiliaires et informatique confidentielle (*confidential computing*)

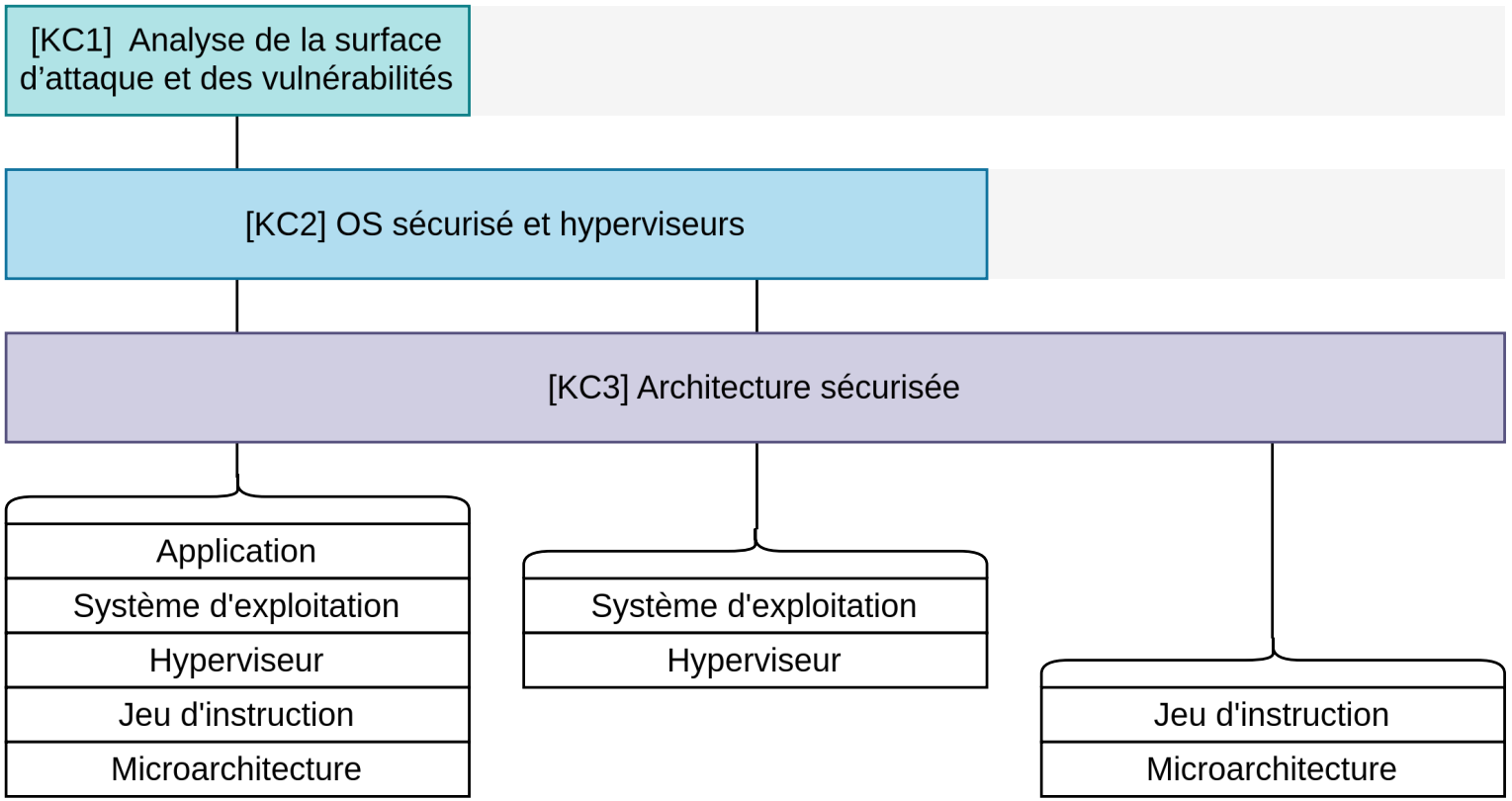
Fuite d'information possible même si le logiciel est de confiance
⇒ Le logiciel peut être chiffré (RSA, AES, etc.)

Le matériel sous-jacent est vulnérable
⇒ Informations sur la microarchitecture, informations sur l'exécution du programme



3.3. Projet ANR SCAMA

Secure-by-Design Computing Against Microarchitectural Attacks



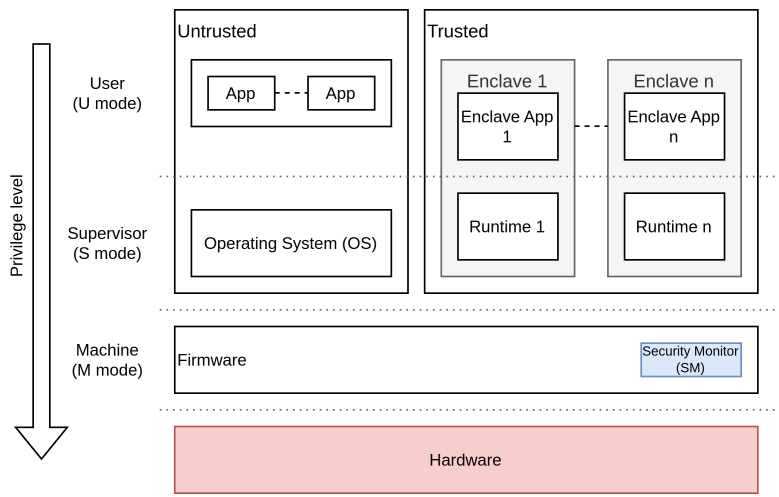
3.3. Contremesures logicielles pour des attaques temporelles sur les mémoires cache

Environnement d'exécution de confiance - TEE Keystone sur RISC-V :

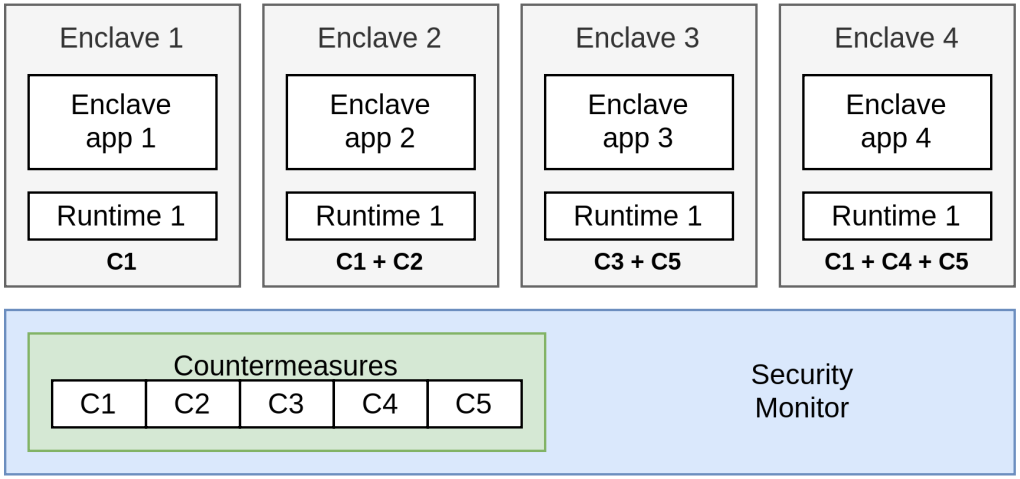
- Analyse détaillée du modèle de sécurité de l'environnement utilisé dans le projet
- Identifier ses faiblesses contre des attaques temporelles sur les mémoires caches

Nos objectifs :

- Développer de nouveaux mécanismes de protection
- Déployer le TEE renforcé sur processeur RISC-V CVA6
- Évaluer les coûts en performance et en sécurité



3.3. Vers une sécurité modulaire



- Différents mécanismes de sécurité (C1 à C5)
- Sécurité combinées de plusieurs mesures ?

Contremesures envisagées

- Exécution temps constant [OST06]
- Injection de bruit [MDS12]
- Cache Flushing [ZR13]
- Architecture de caches [Rib+22]
- Extensions spécifiques aux TEEs [DFS20]

3.3. HermiCache

Comparaison de HermiCache avec d'autres mécanismes d'isolation de cache adaptés aux TEEs

Solution	**Type d'isolation**	**Plateforme d'évaluation**	**Métrique**	**Surcoût (%)**	**TEE utilisé**
HybCache [DFS20]	Way Partitioning	gem5	IPC	0–15	-
Chunked-Cache [Des+21]	Set Partitioning	gem5	Miss rate	Variable	-
Composable Cachelets [Tow+22]	Fine-grained (cachelets)	gem5	IPC	0–27	Intel SGX [CD16]
SENSE [San+24]	Event Notification	gem5	Runtime	2–130	Intel SGX [CD16]
VeriCache [Yin+25]	Set Partitioning	gem5	IPC	0–60	Intel SGX [CD16]
HermiCache	**Fine-grained**	**FPGA + gem5**	**IPC**	**0–11**	Keystone [Lee+20]

Sécurité modulaire au plus près du matériel au meilleur compromis performance/sécurité

3.3. Axe 3 - Synthèse

- Architecture de cache avec contremesures pour temps constant **[Gau+24; Pet+24]**
- Sauvegarde des verrous ⇒ Postdoc projet LockOS “Embedded Operating System for Hardware Cache Locking support”
- Sécurité des caches avec TEE [14]

14. Soumission en cours à HOST 2026 : “HermiCache : Enclave-Aware Cache Replacement for Trusted Execution Environments”

Conclusion

Activités de recherche

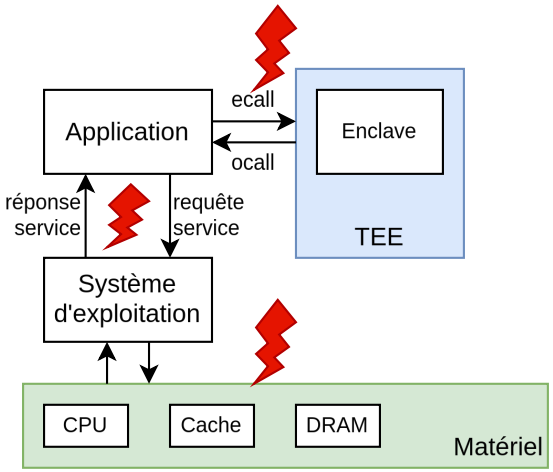
La sécurité à la frontière logicielle/matérielle :

- Depuis la microarchitecture
- Jusqu'à l'intégration de composants reconfigurables associés à des processeurs embarqués

Perspectives - Sécurité et microarchitecture

- Comment la microarchitecture peut impacter la sécurité du système ?
- Détection et contremesures d'attaques sur des systèmes avec du logiciel et du matériel

4. Fuites dans les enclaves



- *Working group* TEE RISC-V [15]
- Fuites dans les enclaves SGX [Zha+24]
- Vulnérabilités sur le *context switching* (EvilCS [JBM24])

Fuites microarchitecturales dans un TEE RISC-V

- D'autres sources de fuites que les caches ? (CVA6, CV32E40S)
- Métriques applicables à plusieurs TEEs (OpenMZ, Penglai, etc.)

15. `https://github.com/riscv-non-isa/riscv-ap-tee`

4. Hyperviseur RISC-V et microarchitecture

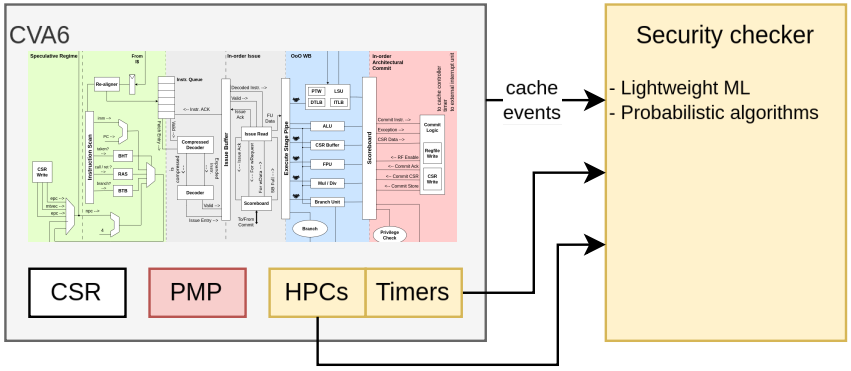
- Fuites sur un bus dans un environnement ARM [ROP24]
- Vulnérabilités dans les spécifications des bus (AXI [ZHS25], APB [Fer+17])

Activités

- De l'hyperviseur à la microarchitecture ($\Rightarrow$ Université de Minho)
- Canaux auxiliaires dans les bus AXI

$\Rightarrow$ Détection de chevaux de Troie matériels

4. Coprocesseur RISC-V pour monitoring d'accès illicites



Accès mémoire illicites (IMAs) et trojans matériels (HTs) détectés par le PMP.
Le PMP peut être compromis (élévation privilèges)

Approche

- Suivi de signaux micro-architecturaux
- ML et/ou probabilité pour la supervision

Travaux existants

✗ Basés sur du logiciel : [Sá+23 ; Che+22]

✓ Travaux existants sur gem5 [Pal+21 ; Rib+24]

Activités

- Investigations sous gem5 et Verilator pour le CVA6
- Détecteurs temps réel en matériel

Projets et encadrements

Co-encadrements :

- 6 thèses
- 4 stagiaires

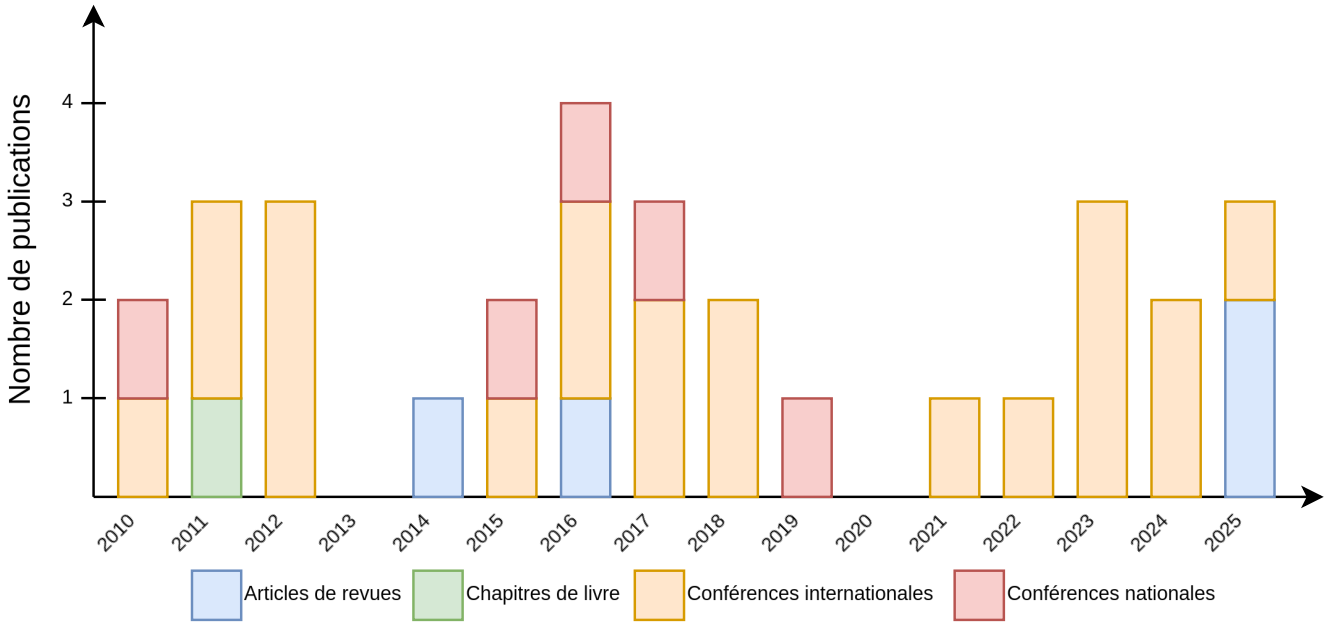
Projets collaboratifs :

- HardBlare (Cominlabs 2015-2018)
- TrustGW (ANR 2022-2026)
- SCAMA (ANR 2024-2028)
- UAV-IDS (Chaire cyberdéf navale 2024-2027)

Collaborations :

- INRIA Rennes
- INRIA Lille
- LTCI
- Hubert-Curien
- LIRMM
- NavalGroup Ollioules

4. Publications [19]



- Soumissions en cours : Elsevier Micpro [16], IEEE ISCAS [17], RAW [18]
- Des outils open-source : `https://github.com/QDucasse/{gigue,jitdomain-tests,cva6-jitdomain}`

16. Suite VM sécurisée
17. Travaux IA et IDS
18. Travaux ANR TrustGW
19. Liste complète des publications : `https://pcotret.gitlab.io/publication/`

Contributions à la sécurité à la frontière logicielle/matérielle

Soutenance d'Habilitation à Diriger des Recherches

Pascal Cotret

Lab-STICC, ENSTA campus de Brest

27 novembre 2025

Références i

[CD16] Victor Costan et Srinivas Devadas. “Intel SGX explained”. In : *Cryptology ePrint Archive (IACR)* (jan. 2016). URL : https://eprint.iacr.org/2016/086.

[CER25] CERT-FR. *Avis de sécurité*. https://www.cert.ssi.gouv.fr/avis/. 2025.

[Che+08] Shimin Chen, Michael Kozuch, Theodoros Strigkos, Babak Falsafi, Phillip B. Gibbons, Todd C. Mowry, Vijaya Ramachandran, Olatunji Ruwase, Michael Ryan et Evangelos Vlachos. “Flexible Hardware Acceleration for Instruction-Grain Program Monitoring”. In : *SIGARCH Comput. Archit. News* 36.3 (juin 2008), p. 377-388. ISSN : 0163-5964. DOI : 10.1145/1394608.1382153. URL : https://doi.org/10.1145/1394608.1382153.

[Che+22] Kevin Cheang, Cameron Rasmussen, Dayeol Lee, David W. Kohlbrenner, Krste Asanović et Sanjit A. Seshia. *Verifying RISC-V Physical Memory Protection*. 2022. arXiv : 2211.02179 [cs.CR]. URL : https://arxiv.org/abs/2211.02179.

[DCL23] Quentin Ducasse, Pascal Cotret et Loïc Lagadec. “Gigue : A JIT Code Binary Generator for Hardware Testing”. In : *2023 Workshop on Virtual Machines and Language Implementations*. Oct. 2023.

[DCL25] Quentin Ducasse, Pascal Cotret et Loïc Lagadec. “War on JITs : Software-Based Attacks and Hybrid Defenses for JIT Compilers - A Comprehensive Survey”. In : *ACM Comput. Surv.* (avr. 2025). ISSN : 0360-0300. DOI : 10.1145/3731598. URL : https://doi.org/10.1145/3731598.

[Den+10] Daniel Y. Deng, Daniel Lo, Greg Malysa, Skyler Schneider et G. Edward Suh. “Flexible and Efficient Instruction-Grained Run-Time Monitoring Using On-Chip Reconfigurable Fabric”. In : *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*. Déc. 2010, p. 137-148. DOI : 10.1109/MICRO.2010.17.

[Des+21] Ghada Dessouky, Alexander Gruler, Pouya Mahmoody, Ahmad-Reza Sadeghi et Emmanuel Stapf. “Chunked-Cache : On-Demand and Scalable Cache Isolation for Security Architectures”. In : *CoRR* abs/2110.08139 (2021). arXiv : 2110.08139. URL : https://arxiv.org/abs/2110.08139.

[DFS20] Ghada Dessouky, Tommaso Frassetto et Ahmad-Reza Sadeghi. “HybCache : Hybrid Side-Channel-Resilient Caches for Trusted Execution Environments”. In : *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, août 2020, p. 451-468. ISBN : 978-1-939133-17-5. URL : https://www.usenix.org/conference/usenixsecurity20/presentation/dessouky.

[Dha+15] Udit Dhawan, Catalin Hritcu, Raphael Rubin, Nikos Vasilakis, Silviu Chiricescu, Jonathan M. Smith, Thomas F. Knight, Benjamin C. Pierce et Andre DeHon. “Architectural Support for Software-Defined Metadata Processing”. In : *SIGARCH Comput. Archit. News* 43.1 (mars 2015), p. 487-502. ISSN : 0163-5964. DOI : 10.1145/2786763.2694383. URL : https://doi.org/10.1145/2786763.2694383.

[DKK07] Michael Dalton, Hari Kannan et Christos Kozyrakis. “Raksha : a flexible information flow architecture for software security”. In : *SIGARCH Comput. Archit. News* 35.2 (juin 2007), p. 482-493. ISSN : 0163-5964. DOI : 10.1145/1273440.1250722. URL : https://doi.org/10.1145/1273440.1250722.

Références ii

[Dom+12] Leonid Domnitser, Aamer Jaleel, Jason Loew, Nael Abu-Ghazaleh et Dmitry Ponomarev. “Non-monopolizable caches : Low-complexity mitigation of cache side channel attacks”. In : *ACM Trans. Archit. Code Optim.* 8.4 (jan. 2012). ISSN : 1544-3566. DOI : 10.1145/2086696.2086714. URL : https://doi.org/10.1145/2086696.2086714.

[Duc+22] Quentin Ducasse, Guille Polito, Pablo Tesone, Pascal Cotret et Loïc Lagadec. “Porting a JIT compiler to RISC-V : Challenges and Opportunities”. In : *MPLR - Managed Programming Languages and Runtimes 2022*. Sept. 2022.

[Fer+17] Nicole Fern, Ismail San, Çetin Kaya Koç et Kwang-Ting Tim Cheng. “Hiding Hardware Trojan Communication Channels in Partially Specified SoC Bus Functionality”. In : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36.9 (2017), p. 1435-1444. DOI : 10.1109/TCAD.2016.2638439.

[Gar+25] Pierre Garreau, Pascal Cotret, Julien Francq, Jean-Christophe Cexus et Loïc Lagadec. “A survey on versatile embedded Machine Learning hardware acceleration”. In : t. 167. Oct. 2025, p. 103501. DOI : https://doi.org/10.1016/j.sysarc.2025.103501. URL : https://www.sciencedirect.com/science/article/pii/S1383762125001730.

[Gau+24] Nicolas Gaudin, Vianney Lapôtre, Pascal Cotret et Guy Gogniat. “A Fine-Grained Dynamic Partitioning Against Cache-Based Timing Attacks via Cache Locking”. In : *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. Juill. 2024.

[Gau25] François Gauthier. “SiFive lance cinq processeurs RISC-V conçus pour supporter les charges de travail de l’IA”. In : *L’Embarqué* (sept. 2025). URL : https://www.lembarque.com/article/sifive-lance-cinq-processeurs-risc-v-concus-pour-supporter-les-charges-de-travail-de-lia.

[Goo25] Google Security Blog. *Vulnerability Reward Program : 2024 in Review*. https://security.googleblog.com/2025/03/vulnerability-reward-program-2024-in.html. Mars 2025.

[Had+25] Hassan Jalil Hadi, Yue Cao, Muhammad Khurram Khan, Naveed Ahmad, Yulin Hu et Chao Fu. “UAV-NIDD : A Dynamic Dataset for Cybersecurity and Intrusion Detection in UAV Networks”. In : *IEEE Transactions on Network Science and Engineering* 12.4 (2025), p. 2739-2757. DOI : 10.1109/TNSE.2025.3553442.

[Heo+15] Ingoo Heo, Minsu Kim, Yongje Lee, Changho Choi, Jinyong Lee, Brent Byunghoon Kang et Yunheung Paek. “Implementing an Application-Specific Instruction-Set Processor for System-Level Dynamic Program Analysis Engines”. In : *ACM Trans. Des. Autom. Electron. Syst.* 20.4 (sept. 2015). ISSN : 1084-4309. DOI : 10.1145/2746238. URL : https://doi.org/10.1145/2746238.

[JBM24] Aruna Jayasena, Richard Bachmann et Prabhat Mishra. “EvilCS : An Evaluation of Information Leakage through Context Switching on Security Enclaves”. In : *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Mars 2024, p. 1-6. DOI : 10.23919/DATE58400.2024.10546809.

Références iii

[Jea23] Jon Gold (adapté par Jean Elyan). “5 géants des puces s’allient autour de RISC-V pour s’affranchir d’ARM”. In : *Le Monde Informatique* (2023). URL : https://www.lemondeinformatique.fr/actualites/lire-5-geants-des-puces-s-allient-autour-de-risc-v-pour-s-affranchir-d-arm-91215.html.

[Jen+25] Aya Jendoubi, Jean-Christophe Prévotet, Philippe Tanguy et Pascal Cotret. “Security of Dynamically Reconfigurable RISC-V Systems : I/O Attack Focus”. In : *39th Annual IEEE International Parallel & Distributed Processing Symposium (IEEE IPDPS 2025) : 32nd Reconfigurable Architecture Workshop*. Milan, Italy, juin 2025. URL : https://hal.science/hal-05117047.

[JHN21] Amine Jaamoum, Thomas Hiscock et Giorgio Di Natale. “Scramble Cache : An Efficient Cache Architecture for Randomized Set Permutation”. In : *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2021, p. 621-626. DOI : 10.23919/DATE51398.2021.9473919.

[KDK09] Hari Kannan, Michael Dalton et Christos Kozyrakis. “Decoupling Dynamic Information Flow Tracking with a dedicated coprocessor”. In : *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*. Juin 2009, p. 105-114. DOI : 10.1109/DSN.2009.5270347.

[Lee+16] Jinyong Lee, Ingoo Heo, Yongje Lee et Yunheung Paek. “Efficient Security Monitoring with the Core Debug Interface in an Embedded Processor”. In : *ACM Trans. Des. Autom. Electron. Syst.* 22.1 (mai 2016). ISSN : 1084-4309. DOI : 10.1145/2907611. URL : https://doi.org/10.1145/2907611.

[Lee+20] Dayeol Lee, David Kohlbrenner, Shweta Shinde, Krste Asanovic et Dawn Song. “Keystone : An Open Framework for Architecting Trusted Execution Environments”. In : *Proceedings of the Fifteenth European Conference on Computer Systems*. EuroSys ’20. Avr. 2020.

[Lu21] Tao Lu. “A Survey on RISC-V Security : Hardware and Architecture”. In : *CoRR* abs/2107.04175 (2021). arXiv : 2107.04175. URL : https://arxiv.org/abs/2107.04175.

[MDS12] Robert Martin, John Demme et Simha Sethumadhavan. “TimeWarp : rethinking timekeeping and performance monitoring mechanisms to mitigate side-channel attacks”. In : *SIGARCH Comput. Archit. News* 40.3 (juin 2012), p. 118-129. ISSN : 0163-5964. DOI : 10.1145/2366231.2337173. URL : https://doi.org/10.1145/2366231.2337173.

[OST06] Dag Arne Osvik, Adi Shamir et Eran Tromer. “Cache Attacks and Countermeasures : The Case of AES”. In : *Topics in Cryptology – CT-RSA 2006*. Sous la dir. de David Pointcheval. Berlin, Heidelberg : Springer Berlin Heidelberg, fév. 2006, p. 1-20. ISBN : 978-3-540-32648-9.

[Pal+18] Christian Palmiero, Giuseppe Di Guglielmo, Luciano Lavagno et Luca P. Carloni. “Design and Implementation of a Dynamic Information Flow Tracking Architecture to Secure a RISC-V Core for IoT Applications”. In : *2018 IEEE High Performance extreme Computing Conference (HPEC)*. 2018, p. 1-7. DOI : 10.1109/HPEC.2018.8547578.

Références iv

[Pal+21] Alessandro Palumbo, Luca Cassano, Pedro Reviriego, Giuseppe Bianchi et Marco Ottavi. “A Lightweight Security Checking Module to Protect Microprocessors against Hardware Trojan Horses”. In : *2021 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. 2021, p. 1-6. DOI : 10.1109/DFT52944.2021.9568291.

[Pen24] William Pensec. “Enhanced Processor Defence Against Physical and Software Threats by Securing DIFT Against Fault Injection Attacks”. Theses. Université Bretagne sud, déc. 2024. URL : https://hal.science/tel-04862037.

[Pet+24] Moritz Peters, Nicolas Gaudin, Jan Philipp Thoma, Vianney Lapôtre, Pascal Cotret, Guy Gogniat et Tim Güneysu. “On The Effect of Replacement Policies on The Security of Randomized Cache Architectures”. In : *19th ACM ASIA Conference on Computer and Communications Security (ACM ASIACCS 2024)*. Juill. 2024.

[Rib+22] Jordi Ribes-González, Oriol Farràs, Carles Hernández, Vatistas Kostalabros et Miquel Moretó. *A Security Model for Randomization-based Protected Caches*. Cryptology ePrint Archive, Paper 2022/440. 2022. DOI : 10.46586/tches.v2022.i3.1-25. URL : https://eprint.iacr.org/2022/440.

[Rib+24] Stefano Ribes, Fabio Malatesta, Grazia Garzo et Alessandro Palumbo. “Machine Learning-Based Classification of Hardware Trojans in FPGAs Implementing RISC-V Cores”. In : *ICISSP 2024 - 10th International Conference on Information Systems Security and Privacy*. Rome, Italy, fév. 2024, p. 1-8. DOI : 10.5220/0012324200003648. URL : https://hal.science/hal-04685628.

[RIS25] RISC-V International / riscv-non-isa. *RISC-V Processor Trace Specification*. https://github.com/riscv-non-isa/riscv-trace-spec. GitHub repository, latest release version 2.0 (tag : 2.0-20250616), released June 23, 2025. Juin 2025.

[ROP24] Cristiano Rodrigues, Daniel Oliveira et Sandro Pinto. “BUSted !!! Microarchitectural Side-Channel Attacks on the MCU Bus Interconnect”. In : *2024 IEEE Symposium on Security and Privacy (SP)*. 2024, p. 3679-3696. DOI : 10.1109/SP54263.2024.00062.

[Ruw+08] Olatunji Ruwase, Phillip B. Gibbons, Todd C. Mowry, Vijaya Ramachandran, Shimin Chen, Michael Kozuch et Michael Ryan. “Parallelizing dynamic information flow tracking”. In : *Proceedings of the Twentieth Annual Symposium on Parallelism in Algorithms and Architectures*. SPAA ’08. Munich, Germany : Association for Computing Machinery, juin 2008, p. 35-45. ISBN : 9781595939739. DOI : 10.1145/1378533.1378538. URL : https://doi.org/10.1145/1378533.1378538.

[Sá+23] Bruno Sá, Francisco Marques, Manuel Rodriguez, José Martins et Sandro Pinto. “Holistic RISC-V Virtualization : CVA6-based SoC”. In : *Proceedings of the 20th ACM International Conference on Computing Frontiers*. CF ’23. Bologna, Italy : Association for Computing Machinery, août 2023, p. 389-390. ISBN : 9798400701405. DOI : 10.1145/3587135.3591436. URL : https://doi.org/10.1145/3587135.3591436.

Références v

[San+24] Fan Sang, Jaehyuk Lee, Xiaokuan Zhang, Meng Xu, Scott Constable, Yuan Xiao, Michael Steiner, Mona Vij et Taesoo Kim. “SENSE : Enhancing Microarchitectural Awareness for TEEs via Subscription-Based Notification”. In : jan. 2024. doi : 10.14722/ndss.2024.24176.

[Tha25] Thales Group. *Thales inaugure à Cholet son nouveau centre d’excellence en radiocommunications, pour répondre aux besoins croissants des forces armées*. https://www.thalesgroup.com/fr/actualites-du-groupe/communiques-de-presse/thales-inaugure-cholet-son-nouveau-centre-dexcellence-en. Communiqué de presse. Thales Group, juin 2025.

[Tow+22] Daniel Townley, Kerem Arikan, Yu David Liu, Dmitry Ponomarev et Oğuz Ergin. “Composable Cachelets : Protecting Enclaves from Cache Side-Channel Attacks”. In : *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA : USENIX Association, août 2022, p. 2839-2856. isbn : 978-1-939133-31-1. url : https://www.usenix.org/conference/usenixsecurity22/presentation/townley.

[Wah+17] Muhammad Abdul Wahab, Pascal Cotret, Mounit Nasr Allah, Guillaume Hiet, Vianney Lapôtre et Guy Gogniat. “ARMHEx : A hardware extension for DIFT on ARM-based SoCs”. In : *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*. Juill. 2017, p. 1-7.

[Wah+18a] Muhammad Abdul Wahab, Pascal Cotret, Mounit Nasr Allah, Guillaume Hiet, Vianney Lapôtre, Guy Gogniat et Arnab Kumar Biswas. “A MIPS-based coprocessor for information flow tracking in ARM SoCs”. In : *2018 International Conference on Reconfigurable Computing and FPGAs (Reconfig)*. Déc. 2018, p. 1-8.

[Wah+18b] Muhammad Abdul Wahab, Pascal Cotret, Mounit Nasr Allah, Guillaume Hiet, Vianney Lapôtre, Guy Gogniat et Arnab Kumar Biswas. “A novel lightweight hardware-assisted static instrumentation approach for ARM SoC using debug components”. In : *Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. Déc. 2018, p. 1-6.

[Wan+23] Yu Wang, Jinting Wu, Haodong Zheng, Zhenyu Ning, Boyuan He et Fengwei Zhang. “Raft : Hardware-assisted Dynamic Information Flow Tracking for Runtime Protection on RISC-V”. In : *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*. RAID ’23. Hong Kong, China : Association for Computing Machinery, oct. 2023, p. 595-608. isbn : 9798400707650. doi : 10.1145/3607199.3607246. url : https://doi.org/10.1145/3607199.3607246.

[Wer+19] Mario Werner, Thomas Unterluggauer, Lukas Giner, Michael Schwarz, Daniel Gruss et Stefan Mangard. “ScatterCache : Thwarting Cache Attacks via Cache Set Randomization”. In : *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA : USENIX Association, août 2019, p. 675-692. url : https://www.usenix.org/conference/usenixsecurity19/presentation/werner.

[Wie+25] Philip Wiese, Gamze Islamoglu, Moritz Scherer, Luka Macan, Victor J.B. Jung, Alessio Burrello, Francesco Conti et Luca Benini. “Toward Attention-based TinyML : A Heterogeneous Accelerated Architecture and Automated Deployment Flow”. In : *IEEE Design & Test* (2025), p. 1-1. doi : 10.1109/MDAT.2025.3527371.

Références vi

[WL07] Zhenghong Wang et Ruby B. Lee. “New cache designs for thwarting software cache-based side channel attacks”. In : *SIGARCH Comput. Archit. News* 35.2 (juin 2007), p. 494-505. ISSN : 0163-5964. DOI : 10.1145/1273440.1250723. URL : https://doi.org/10.1145/1273440.1250723.

[Yin+25] Lingfeng Yin, Haixia Wang, Yongqiang Lyu, Chaojian Hu et Dongsheng Wang. “VeriCache : Formally Verified Fine-Grained Partitioned Cache for Side-Channel-Secure Enclaves”. In : *IEEE Transactions on Dependable and Secure Computing* (2025), p. 1-12. DOI : 10.1109/TDSC.2025.3525628.

[Zha+24] Xiaohan Zhang, Jinwen Wang, Yueqiang Cheng, Qi Li, Kun Sun, Yao Zheng, Ning Zhang et Xinghua Li. “Interface-Based Side Channel in TEE-Assisted Networked Services”. In : *IEEE/ACM Transactions on Networking* 32.1 (fév. 2024), p. 613-626. DOI : 10.1109/TNET.2023.3294019.

[ZHS25] Melisande Zonta, Nora Hinderling et Shweta Shinde. “Xray : Detecting and Exploiting Vulnerabilities in Arm AXI Interconnects”. In : *2025 Design, Automation & Test in Europe Conference (DATE)*. Avr. 2025, p. 1-7. DOI : 10.23919/DATE64628.2025.10992968.

[Zon+25] Melisande Zonta-Roudes, Andres Meza, Nora Hinderling, Lucas Deutschmann, Francesco Restuccia, Ryan Kastner et Shweta Shinde. “eXpect : On the Security Implications of Violations in AXI Implementations”. In : *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. ICCAD ’24. Newark Liberty International Airport Marriott, New York, NY, USA : Association for Computing Machinery, avr. 2025. ISBN : 9798400710773. DOI : 10.1145/3676536.3676844.

[ZR13] Yinqian Zhang et Michael K. Reiter. “Düppel : retrofitting commodity operating systems to mitigate cache side channels in the cloud”. In : *CCS*. CCS ’13. Berlin, Germany : Association for Computing Machinery, nov. 2013, p. 827-838. ISBN : 9781450324779. DOI : 10.1145/2508859.2516741. URL : https://doi.org/10.1145/2508859.2516741.