AUGUST 2025

WHITEPAPER

Implementing Application Processor TEEs using RISC-V Supervisor Domains

WHITEPAPER

Ravi Sahita (Editor), Vedvyas Shanbhogue, Samuel Ortiz (Rivos Inc.), Krste Asanovic, Samuel Holland, Yann Loisel (SiFive), Andy Dellow, Eckhard Delfs, Osman Koyuncu (Qualcomm), Wojciech Ozga, Guerney Hunt (IBM), Anup Patel, Radim K, Greg Favor (Ventana Microsystems), Luis Fiolhas (Semidynamics), Nick Kossifidis (FORTH), Jiewen Yao (Intel), Yuan Skye (Beijing Institute of Open Source Chip), Siqi Zhao (Alibaba), Paul Ku (Andes), Tuo Li (Chinese Academy of Sciences), Philipp Tomsich (Vrull), Ruud Derwig (Synopsys), Baskaran Chidambaram (MIPS), Andrea Gallo (RISC-V)

Contents

Glossary

TEE

Trusted execution environment (TEE) is a set of hardware and software mechanisms that allow creating an attestable trusted computing base and isolated execution environment. A TEE provides confidentiality and integrity for the code and data loaded into it. A TEE may be used to provide security services at a higher trust level than the general operating environment.

CoVE

RISC-V non-ISA ABI to support confidential-compute scenarios. CoVE or Confidential VM Extensions describes the threat model, reference architecture and ABIs for confidential virtual machine workloads on RISC-V platforms.

Supervisor Domain

A domain refers to an environment or context that includes a set of system resources and a set of system entities that have the right to access the resources as defined by a common security policy, security model, or security architecture [17].

In the context of this paper, a supervisor domain is a supervisor execution context isolated from other supervisor execution contexts on the same platform.

RDSM

Root Domain Security Manager - Software that runs in RISC-V M-mode and manages isolation between supervisor domains. Used in various use cases such as TEE, partitioning and fault isolation.

TSM

TEE Security Manager (TSM) is software that operates in a supervisor domain that is used for workload isolation in the RISC-V Confidential VM Environment (CoVE) architecture. The CoVE TSM is a software module that enforces security properties on a platform, specific to the CoVE threat model.

Introduction

RISC-V is a versatile and extensible ISA that supports a wide range of computing platforms—from deeply embedded systems to high-performance application processors and accelerators. Many of these platforms process sensitive, high-value data that demands strong confidentiality, integrity, and verifiable execution guarantees. Similarly, safety-critical systems require robust fault isolation and partitioning to ensure the integrity of mission-critical operations.

Modern computing environments often involve multi-tenant or multi-level workloads, necessitating scalable isolation mechanisms. RISC-V addresses these needs through its privileged ISA and associated extensions, enabling the creation of **supervisor domains** — isolated execution contexts that can enforce trust boundaries or mutual isolation between workloads.

Diverse use cases can be addressed, including:

- **Trusted Execution Environments (TEEs):** Enabling secure enclaves for access control, data confidentiality and integrity, supply chain assurance, and intellectual property protection.
- **Confidential Computing:** Isolating code and data even from more privileged software
- **Fault Isolation:** Supporting functional safety through code integrity enforcement, authenticated firmware updates, and system manageability.
- **Static and Dynamic Resource Partitioning:** Facilitating multi-level security (MLS) and workload separation across trust boundaries.

This non-normative application note outlines how to implement these isolation-oriented use cases using RISC-V supervisor domains. It describes both **ISA-defined mechanisms** (hardware-level) and **non-ISA mechanisms** (hardware and software) that enable instantiation and management of isolated supervisor contexts.

At a foundational level, the privileged ISA extension supports isolation between two supervisor domains—for example, to implement secure and non-secure worlds. At a more advanced level, the same architectural framework can support multiple concurrent domains, enabling complex trust hierarchies and use cases such as **multi-level security systems** and **confidential computing** [10].

To further strengthen isolated workloads, RISC-V supports additional security features such as **Control Flow Integrity (CFI)** and other software-based hardening techniques. For a comprehensive overview of ratified and in-development security mechanisms beyond the scope of this note, refer to the RISC-V Security Model Specification [3].

Executive Summary

Isolated processor contexts	Hart isolation is based on privilege levels [1] and supervisor domains* [2]: • U - User/Application - can be in any RDSM-selected isolated supervisor domains supported by the hart • S - Supervisor - can be in any RDSM selected isolated supervisor domains supported by the hart • HS - Supervisor/ Hypervisor - can be in any RDSM selected isolated supervisor domains supported by the hart • M - Machine - Root domain manager - manages transitions across supervisor domains
Memory Isolation across isolated contexts (processor accesses)	• Hart MMU with PMP [1], SPMP [13], Smepmp [1], Sv [1], Smmpt* [2]. • RDSM manages isolation across supervisor domain contexts using a Memory Protection Table (*Smmpt**)
Context Switching	• M-mode RDSM software handles state transitions between domains • S/HS-mode software handles context switching for workloads within an isolated supervisor domain (e.g. TSM)
Isolation within supervisor domain	• Achieved via use of Sv [1] / SPMP* by the Supervisor Domain Security Manager (e.g., OS, hypervisor, runtime) which manages isolation across VMs or process contexts.
Memory encryption	• Platform-specific (out of scope of the RISC-V ISA) - see requirements in the RISC-V Security Model Specification* [3].
SoC support for isolated contexts	• IOMMU [4] with IO-MPT* [2] • IOPMP [12] and similar IO rule checkers* [15] • AIA [5] with Smsdia* [2]
Software support	• M-mode RDSM • S/HS-mode TEE Security Manager (TSM) that implements the CoVE* ABI-specific to use case such as Salus [14] • Applications within a supervisor domain (no changes required if hosted within a RISC-V Confidential VM [6]

* Specifications that are in development (STABLE)

Reference Architecture

RISC-V privileged ISA provides architecture support for (vertical) isolation via privilege levels (M, S(H), and (V)S, (V)U as shown below. The privileged ISA extensions defined as part of the Supervisor Domain specification provides a second dimension that allows (horizontal) isolation via creation of more than one isolated supervisor domain. Supervisor Domain specification is currently in the RVI ratification process as a stable specification. Supervisor domains allow for more than one isolated supervisor contexts to be instantiated - the policies of workload assignment to an isolated supervisor domain (for e.g. with labels such as “secure” or “nonsecure”) are expected to be customized by software and configuration policies based on usage requirements. The policies for managing TEEs within each supervisor domain may be distinct because they are isolated.

A basic (nominal) 2-domain system is shown (in figure 1) below, with some example labels as non-secure and secure:

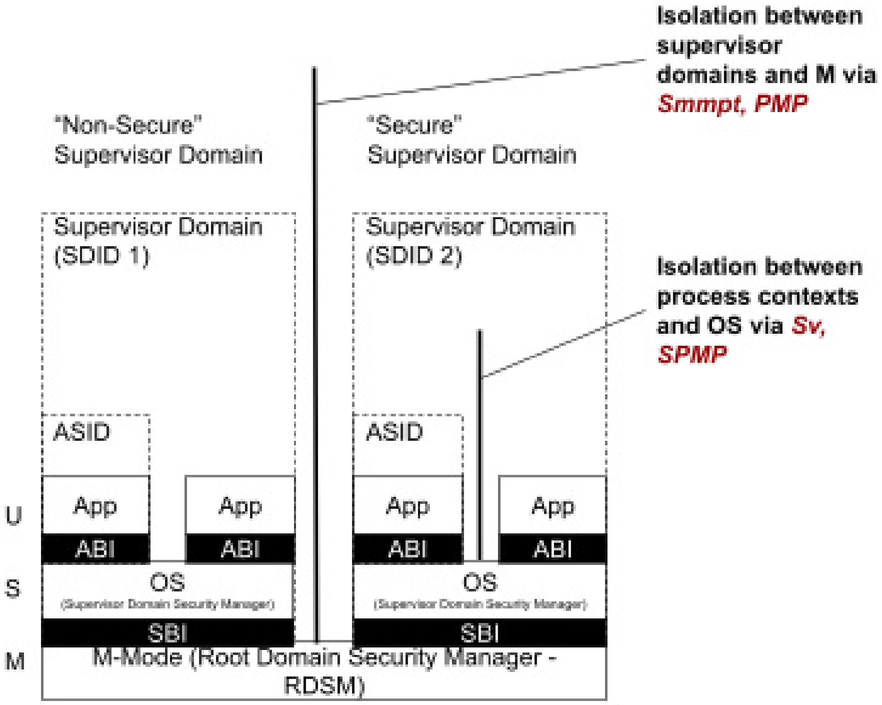
Isolation between supervisor domains and M via *Smmpt, PMP*
"Non-Secure" Supervisor Domain
"Secure" Supervisor Domain
Supervisor Domain (SDID 1)
Supervisor Domain (SDID 2)
Isolation between process contexts and OS via *Sv, SPMP*
ASID
ASID
U
S
M
App
App
App
App
ABI
ABI
ABI
ABI
OS
(Supervisor Domain Security Manager)
OS
(Supervisor Domain Security Manager)
SBI
SBI
M-Mode (Root Domain Security Manager - RDSM)

Figure 1: Nominal secure, non-secure two-domain TEE

As shown below (in figure 2), these nominal 2 basic domains may also be used with H-extension to further isolate workloads within a domain as virtual machines.

Further, supervisor domains provide broader architectural flexibility for additional isolation levels (note flexibility in variety of configurations in the supervisor domains from M+U to M+HS+U within a domain:

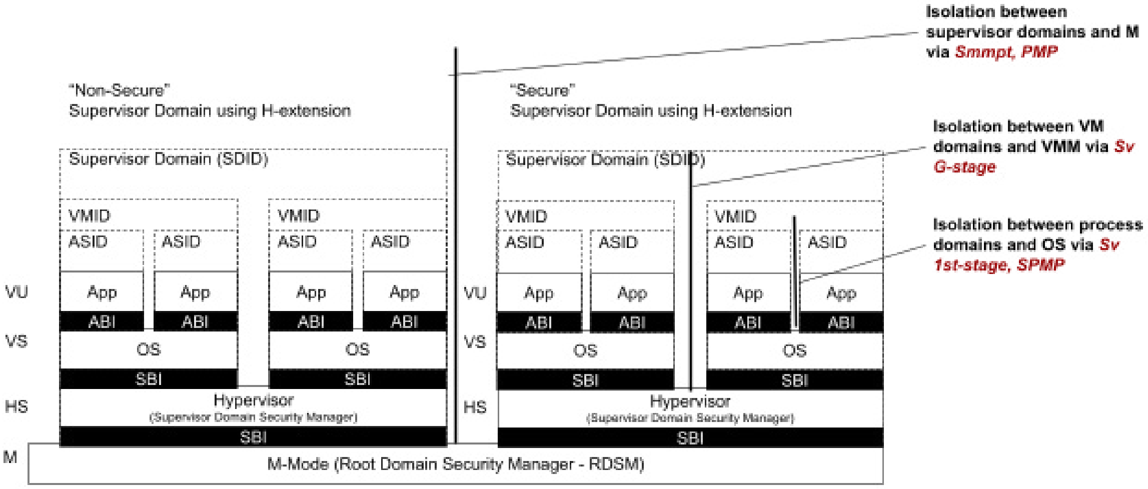
Isolation between supervisor domains and M via Smmpt, PMP
"Non-Secure" Supervisor Domain using H-extension
"Secure" Supervisor Domain using H-extension
Isolation between VM domains and VMM via Sv G-stage
Isolation between process domains and OS via Sv 1st-stage, SPMP
Supervisor Domain (SDID)
VMID
ASID
App
ABI
OS
SBI
Hypervisor
(Supervisor Domain Security Manager)
VU
VS
HS
M
M-Mode (Root Domain Security Manager - RDSM)

Figure 2: Nominal secure, non-secure two-domain TEE with Hypervisor extension

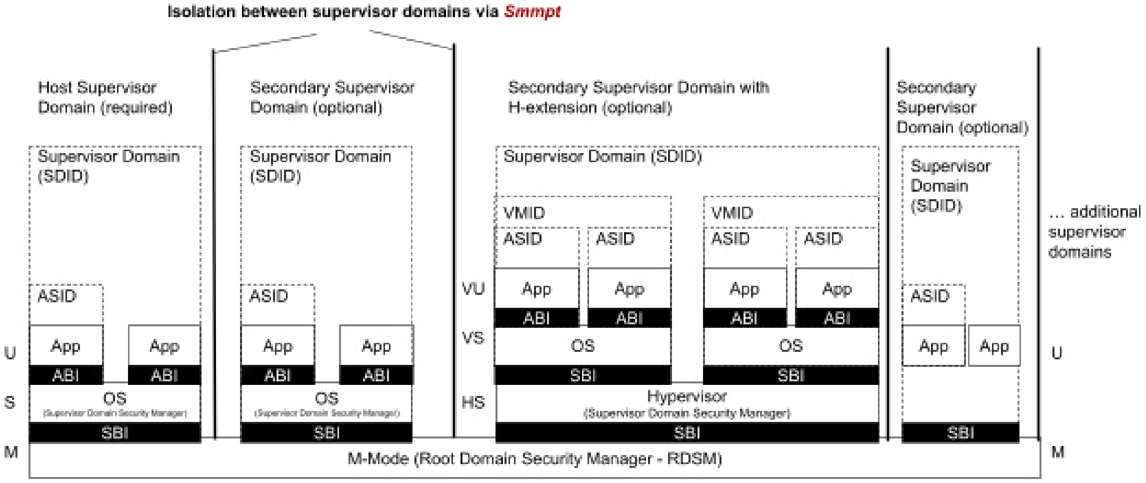
Isolation between supervisor domains via Smmpt
Host Supervisor Domain (required)
Secondary Supervisor Domain (optional)
Secondary Supervisor Domain with H-extension (optional)
Secondary Supervisor Domain (optional)
Supervisor Domain (SDID)
VMID
ASID
App
ABI
OS
(Supervisor Domain Security Manager)
SBI
Hypervisor
(Supervisor Domain Security Manager)
... additional supervisor domains
U
S
M
VU
VS
HS
M-Mode (Root Domain Security Manager - RDSM)

Figure 3: N-domain TEE with mixed privilege level usage

Privilege Levels and Supervisor Domains

Privilege levels are used to provide protection between different components of the software stack, and attempts to perform operations not permitted by the current privilege mode will cause an exception to be raised. These exceptions will normally cause traps into an underlying execution environment. The machine level has the highest privileges and is the only mandatory privilege level for a RISC-V hardware platform. Code run in machine-mode (M-mode) is the highest privilege mode, as it has low-level access to the machine implementation. M-mode can be used to manage secure execution environments on RISC-V. User-mode (U-mode) and supervisor-mode (S-mode) are intended for conventional application and operating system usage respectively. Supervisor mode can use the Hypervisor extension for virtual machine monitors to manage virtual machines (which can run unmodified operating systems and applications).

Supervisor domain (Smsdid) is associated with a set of physical address regions that are isolated from other supervisor domains on the same platform, with the Root Domain Security Manager (RDSM) as the software TCB. The RDSM operates in M-mode which manages the isolation properties of the physical address spaces via the supervisor domain ISA extension. The access the RDSM has to lower privilege may be restricted via Smepmp. A supervisor domain identifier (SDID) is associated with the hart operating in the context of a supervisor domain to facilitate physical address isolation on a per supervisor domain basis. Supervisor domains must rely on a Trusted Computing Base (TCB) which consists of the RDSM (software) and hardware (e.g. hart, SoC, Root-of-trust) that enforces the isolation properties for the supervisor domain. The RDSM may utilize PMP/Smepmp and/or the Smmpt (Machine-level Memory Protection Tables) extension to isolate physical memory between supervisor domains. Isolation of the workloads within a supervisor domain is the responsibility of the software (e.g. OS/hypervisor) managing the supervisor domain and using the virtual memory protection ISA (Sv or SPMP).

Implementation-specific notes: A key goal of using supervisor domains is to be able to reduce the TCB for each domain. The quantification of the TCB is done via measurements of HW and SW and reported via cryptographically-verified reports - this process is called attestation [10]. Usages may enable the attestation of each supervisor domain independently from other domains. Sensitive data may be entrusted to a particular domain after verifying the trust properties statically (via boot) or dynamically (via attestation). These trust properties are established as part of the hardware and software supply chain, system configuration and can be evaluated using attestation mechanisms.

The Root Domain Security Manager

The RDSM software manages the isolation of domains, and the transitions between them. It executes in M-Mode, with fixed entry points from lower privilege domains. It is responsible for saving and restoring state on transitions between domains, and enabling access to resources allowed for the active domain. In addition to the HW, the RDSM is considered a key part of the TCB and the RDSM's measurements are expected to be verified as part of the secure boot [16] on the platform as well as included in attestation reports to any external relying party to enable use cases such as attested TEEs and confidential computing. Dynamic measurements and attestation of the workloads as they are created and destroyed may then be facilitated by the static TCB.

Switching between Supervisor Domains

The ECALL instruction is used to make a request from the supervisor domain software to the RDSM. When executed in U-mode, the supervisor domain software handles the ECALL if delegated. When executed from S-mode, the request is handled by the RDSM that uses Smsdid CSRs to activate the required supervisor domain on the hart and context switches any security state (typically in registers) for the hart. SBI extensions define the ECALL ABI convention for context switching to another supervisor domain, e.g. see the RISC-V CoVE specification for the COVH and COVG SBI extension that define the ABI between the host OS and the TSM. Similarly, returning from a supervisor domain to a prior supervisor domain is achieved either synchronously by an ECALL initiated serviced by the RDSM that context switches the security state of the hart or via asynchronous event (interrupt/exception) which allows the RDSM to evaluate the event condition and either shuts down, resumes the original supervisor domain or another supervisor domain.

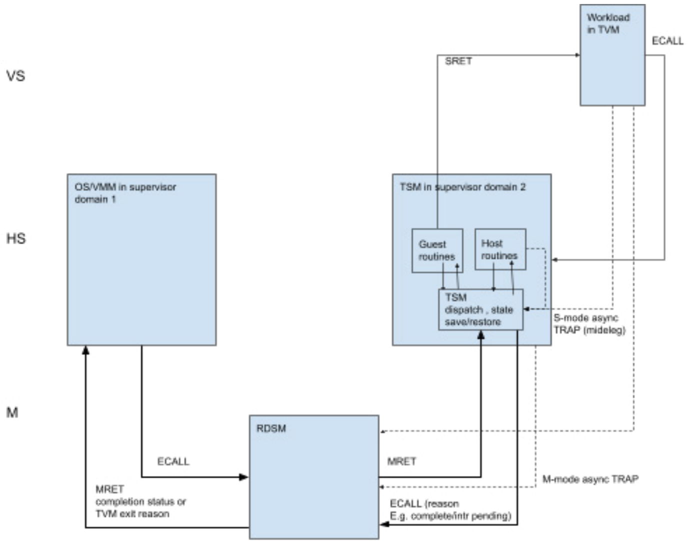
VS
HS
M
Workload in TVM
ECALL
SRET
OS/VMM in supervisor domain 1
TSM in supervisor domain 2
Guest routines
Host routines
TSM dispatch , state save/restore
S-mode async TRAP (mideleg)
RDSM
ECALL
MRET
MRET completion status or TVM exit reason
M-mode async TRAP
ECALL (reason E.g. complete/intr pending)

Figure 4: Security Context switching between two supervisor domains

Memory Isolation

PMP/Smepmp, Sv and Smmpt (M-level Memory Protection Tables)

Refer to the logical access-control model for memory accesses in Figure 5. To support secure processing and contain faults, it is desirable to limit the physical addresses accessible by the supervisor domain active on a hart. An optional physical memory protection (PMP) unit provides per-hart machine-mode control registers to allow physical memory access privileges (read, write, execute) to be specified for each physical memory region. The PMP values are checked in parallel with the PMA checks, and PMP rules are typically static in nature. PMP checks are applied to all accesses whose effective privilege mode is S or U, including instruction fetches and data accesses in S and U mode, and data accesses in M-mode when the MPRV bit in mstatus is set and the MPP field in mstatus contains S or U. PMP checks are also applied to page-table accesses for virtual-address translation, for which the effective privilege mode is S. Optionally, PMP checks may additionally apply to M-mode accesses, in which case the PMP registers themselves can be locked, so that even M-mode software cannot change them until the hart is reset. Using Machine Mode Lockdown (MML), M-mode may only execute code from a fixed set of executable regions, without the ability to add new ones after the MML bit is set. In this mode which is defined as part of the Smepmp extension, M-mode is also restricted from accessing and/or executing S or U mode memory, which prevents a class of privilege escalation attack vectors. All PMP registers are WARL which allows vendors to have a predefined ruleset as early as when the hart comes out of reset. PMP violations are always trapped precisely at the processor.

When page-based virtual memory (Sv) is used, it composes with the physical memory protection. When paging is enabled, instructions that access virtual memory may result in multiple physical-memory accesses, including implicit references to the page tables. The PMP checks apply to all of these accesses. The effective privilege mode for implicit page table accesses is S. Implementations with virtual memory are permitted to perform address translations speculatively and earlier than required by an explicit memory access, and are permitted to cache them in address translation cache structures, e.g.TLBs - including possibly caching the identity mappings from effective address to physical address used in Bare translation modes and M-mode. The PMP settings for the resulting physical address may be checked (and possibly cached) at any point between the address translation and the explicit memory access. Hence, when the PMP settings are modified, M-mode software must synchronize the PMP settings with the virtual memory system and any PMP or address-translation caches. This is accomplished by executing a global SFENCE.VMA, after the PMP CSRs are written. Additional synchronization may be required when the hypervisor extension is implemented. If page-based virtual memory is not implemented, memory accesses check the PMP settings synchronously, so no synchronization is needed.

When supervisor domain memory isolation (Smmpt) is used, it composes with page-based virtual memory and PMP. The Smmpt extension specifies a M-mode Memory Protection Table that enables the RDSM to program XWR permissions for physically-addressed memory (or device-mapped regions) by a hart/device operating within a supervisor domain. Associating a hart/device with a supervisor domain implies that any physical-addressable region access occurring in the context of the supervisor domain is subject to Memory Protection Table (MPT) access-checks for that domain. Hence, software or hardware accesses that originate from supervisor domains other than the allowed supervisor domain can be explicitly prevented or explicitly allowed. The RDSM is in the TCB of all supervisor domains and is expected to manage the isolation via MPTs. RDSM access to domain physical memory may be curtailed by the use of Smepmp. In typical security usages, write accesses to the MPT structures must be restricted and managed by the RDSM. Harts and devices may be assigned access to memory for a supervisor domain. For all accesses using a physical address, the SDID is the supervisor domain identifier programmed into a CSR. This CSR is programmed on the hart by the Root Domain Security Manager (RDSM). The assignment of the hart/device to a supervisor domain may be static (e.g. device assignment to a VM) or dynamic (e.g. scheduling a VM virtual cpu within a domain). The MPT for the supervisor domain active on the hart is programmed on the hart along with the supervisor domain identifier. The MPT does not perform any address translation; it simply provides access permissions for the physically addressed regions/pages to enforce the isolation properties per the use case requirements. The in-memory structures used for MPT must themselves be access-limited to the RDSM by use of the MPT structures to disallow any supervisor domain from accessing the structures unless explicitly delegated by the Root Domain Security Manager (RDSM) to a particular domain (per use case policies).

Smwg [15] proposes a hart ISA extension to enable an agent context that initiates a transaction to a physical address to mark the transaction with the world identifier (WID) of the agent context. The transaction is only allowed to complete successfully if the targeted resource has the appropriate access permissions (read or write) on that address for the WID on the transaction. Smwg is designed for the case where the allocation of agent contexts and resources to worlds is performed before or at reset/boot time, and not changed dynamically when the system is running unless there is a system reset.

Similar to PMP, IOPMP [12] is a rule-based checker. Additionally, IOPMP recognises a context, I/O agent, or a channel of a device by an RRID (Request Role ID), which is carried by a transaction. IOPMP checks a transaction by a three-tuple of an RRID, an access region, and an access type of read/write/execution. Every RRID can be associated with multiple sets of rules, while each rule has a region and permitted access types. Since every RRID has a bitmap to its associated rule sets, changing the rule sets of an RRID takes a single write. Thus, IOPMP can quickly dynamically switch the mapping from RRID to its rules, not limited by whether RRID is changeable. IOPMP stores its rules in internal storage, which can mitigate physical attacks on the rules and fix the latency of checking a transaction. IOPMP can fully or partially lock its rules and the above mappings to prevent sensitive data from being accessed maliciously when the RDSM is compromised, which is designed for better defense-in-depth.

Private and Shared Memory Assignment to Supervisor Domains

The RDSM may perform static or run-time configuration of the MPT. A memory region (consisting of one or more pages) may be (re)assigned from one domain to another at run-time e.g. this is done by revoking the permission for one domain, clearing the memory, and assigning permissions to another domain. Run-time configuration may be performed via M-mode CSRs, MPT in-memory structures and MFENCE.SPA (and variants) instructions.

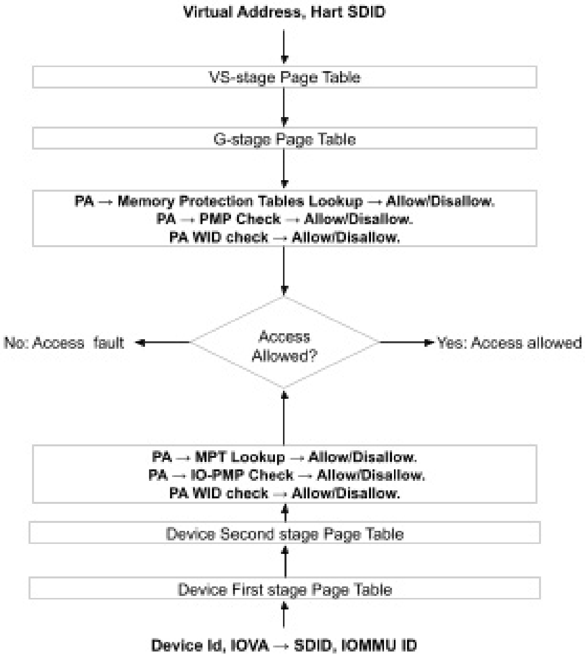
Virtual Address, Hart SDID
VS-stage Page Table
G-stage Page Table
PA → Memory Protection Tables Lookup → Allow/Disallow.
PA → PMP Check → Allow/Disallow.
PA WID check → Allow/Disallow.
No: Access fault
Access Allowed?
Yes: Access allowed
PA → MPT Lookup → Allow/Disallow.
PA → IO-PMP Check → Allow/Disallow.
PA WID check → Allow/Disallow.
Device Second stage Page Table
Device First stage Page Table
Device Id, IOVA → SDID, IOMMU ID

Figure 5: Sv and G-stage address translation and permissions enforcement including MPT lookup

Converting memory regions access-permissions to revoke access from one domain and grant to another (or vice versa), may involve platform-specific operations based on the enforcement mechanism, such asTLB/cache flushes that must be enforced by the RDSM and hardware via the Smsdid fence ISA. The RDSM is expected to change the settings and flush caches if necessary, so the system is only incoherent during the transition between domain assignment settings. This transitory state should not be visible to lower privilege levels (i.e. supervisor domains). There are also security aspects to be considered during (re)configuration, e.g., clearing memory used by the current SD before assigning it to another SD, as well as micro-architectural side-channels/covert-channels.

A hart/device may perform accesses to memory exclusively accessible to its supervisor domain, or to memory shared globally with one or more supervisor domains. Memory sharing between supervisor domains is achieved via the RDSM managing the MPT permissions to make physical memory regions accessible to the required supervisor domains. Memory sharing can be set up between supervisor domains active on a hart or accessing memory via a device assigned to the supervisor domain under the purview of an IOMMU domain. Access to physical addresses initiated from a hart or a device assigned a supervisor domain identifier may be denied by virtue of the permissions in the MPT lookup - such disallowed accesses from a hart cause a trap to the RDSM to report a fault. In the case of a device access disallowed by the MPT, the IO subsystem may log an error for the RDSM which may delegate it to a supervisor domain. (See more details about device accesses with supervisor domain in the IO device section below).

Interrupt Isolation

Some isolated supervisor domains may have devices assigned to them for secure I/O operations. Devices designed to carry out I/O operations typically signal the completion of an I/O task or an associated error via an external interrupt. These external interrupts might be aggregated and indicated by an interrupt controller linked to the RDSM. The RDSM, in the course of handling an external interrupt, should forward the interrupt to the relevant supervisor domain. The efficiency of handling external interrupts by a supervisor domain can be enhanced if the external interrupts could be directly assigned to the supervisor domain - the Smsdia [2] extension enables such functionality. Such direct delivery of external interrupts necessitates a supervisor interrupt domain that can be linked to specific supervisor domains. To accommodate such supervisor domains, To accommodate supervisor domains, Smsdia extends the IMSIC and APLIC specified by the RISC-V Advanced Interrupt Architecture (AIA) specification [5] to support multiple supervisor interrupt domains, enabling an individual supervisor interrupt domain to be associated with each such supervisor domain. A supervisor interrupt domain in an IMSIC consists of a supervisor-level interrupt file and optionally, one or more guest-level interrupt files. A supervisor interrupt domain in both an APLIC and an IMSIC could be linked with a supervisor domain, with the APLIC configured to route interrupts to the associated IMSIC as MSIs to the corresponding interrupt domain of the IMSIC. Interrupt controllers other than APLIC and/or IMSIC that support multiple supervisor domains may also serve a supervisor domain.

Upon scheduling a supervisor domain for execution on a hart, Smsdia provides CSRs (shown in figure below) to the RDSM as a mechanism to select a supervisor interrupt domain in an interrupt controller as a source of S- and VS-level external interrupts for a hart using the CSR. Selecting a supervisor interrupt domain also chooses the linked S and VS-level interrupt pending signals as those detected by the hart for initiating the associated traps. Smsdia also provides the RDSM with an interface to be alerted if an external interrupt, whether at the S- or VS-level, is pending for any supervisor interrupt domains not currently active on a hart. The RDSM may leverage this notification to inform its scheduling decisions. To facilitate this functionality, the Smsdia extension introduces CSRs, along with a local supervisor domain external interrupt.

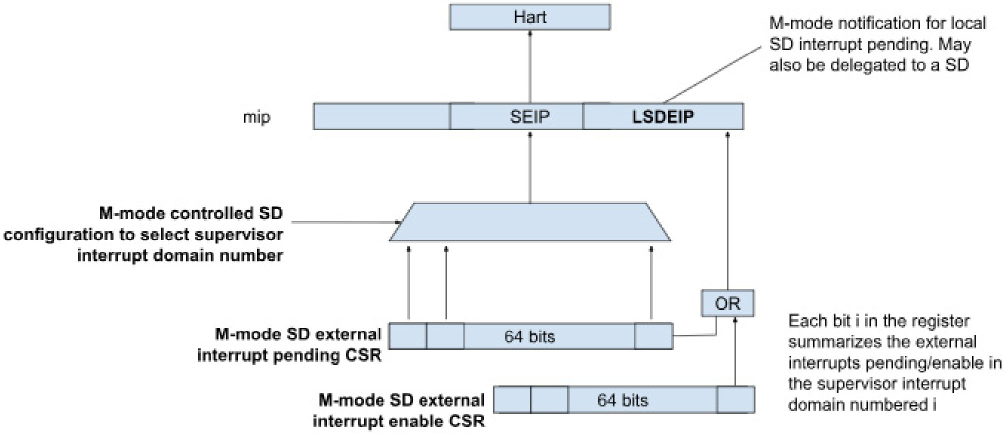
Hart
M-mode notification for local SD interrupt pending. May also be delegated to a SD
mip
SEIP
LSDEIP
M-mode controlled SD configuration to select supervisor interrupt domain number
OR
M-mode SD external interrupt pending CSR
64 bits
Each bit i in the register summarizes the external interrupts pending/enable in the supervisor interrupt domain numbered i
M-mode SD external interrupt enable CSR
64 bits

Figure 6: Smsdia-defined CSRs and controls in bold

IO Device Isolation

Supervisor domains may be granted control over DMA-capable devices. When such direct device association is supported, the system might also incorporate multiple instances of IOMMU. Each IOMMU instance can be tied directly to a supervisor domain, allowing that domain to manage address translation and protection for DMA that originates from devices under its control. To uphold isolation properties, the DMA from the devices and the IOMMU linked with a supervisor domain must adhere strictly to the access protections encoded in the MPT of the respective supervisor domain - this is enforced via the IO-MPT extension programmed by the RDSM. Additionally, using the MPT, the RDSM enforces that the IOMMU memory-mapped programming regions are access-restricted to the supervisor domain the IOMMU is assigned to.

At any given time, a solitary supervisor domain is scheduled for execution on a RISC-V hart by the root domain security manager (RDSM). As part of this scheduling, the RDSM programs a pointer to the MPT into a CSR within the hart. Unlike the RISC-V harts, DMA-capable devices connected to a supervisor domain remain continuously active. Such devices might initiate DMA even if the associated domain is not currently active on any RISC-V harts.

As a result, the MPT of all supervisor domains must be constantly active for DMA protection. Furthermore, the IO subsystem must possess the capability to select the appropriate MPT for enforcement based on the identity of the device initiating the DMA. The IO-MPT non-ISA extension enables this functionality.

The I/O subsystem that implements IO-MPT offers the following functions:

- Supervisor Domain Classifier (SDCL): This classifier within the I/O subsystem interprets the attributes of a DMA request and determines the appropriate MPT for that request.
- MPT Checker (MPTCHK): This function ensures that stipulated access controls by *Smmpt* are applied to the memory regions accessed by the DMA. It uses the MPT identified by the SDCL. Collectively, these two functionalities form a logical block in the I/O subsystem, referred to as the I/O MPT checker (see figure below).

Additional information about IO device assignment to supervisor domains can be found in the CoVE-IO specification [11].

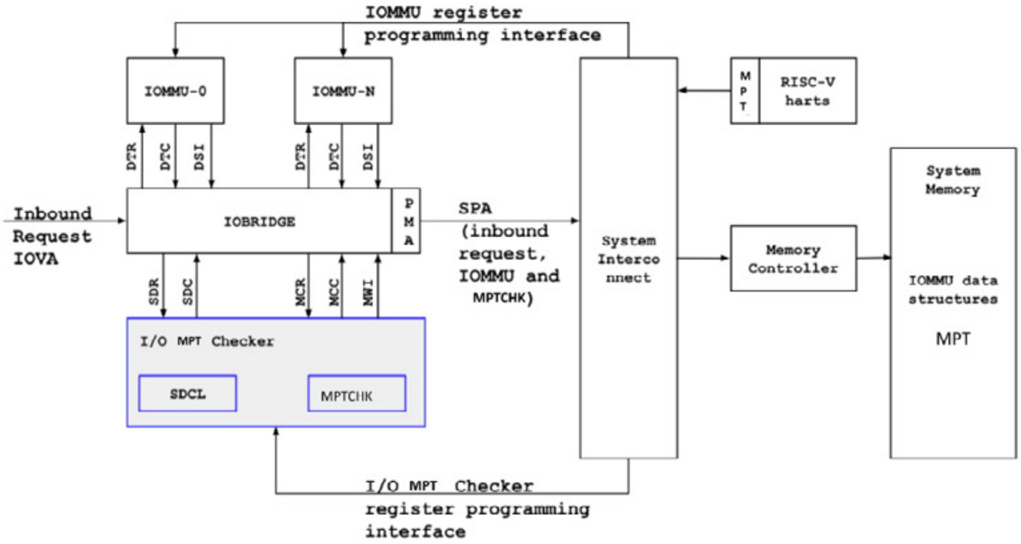

Figure 7: I/O MPT Checker

Invasive Debug and Trace

Smsdedbg and Smsdetrc enable an RDSM to manage isolation of supervisor domains when invasive mechanisms such as external debug and trace are being utilized. External debug/trace refers to debugging/tracing software running on a separate target RISC-V platform, via a debug/trace control/data transport that provides external access to the debug module/trace encoder.

Smsdedbg defines the sdedbgalw bit in the SD configuration CSR that specifies if a supervisor domain is allowed to be externally-debugged. Similarly, Smsdetrc defines the sdetrcalw bit that specifies if a supervisor domain is allowed to be externally-traced. These two extensions only affect debug and trace orchestrated by an external actor - on the other hand, self-hosted debug (and trace) are contained within the workload being debugged with the context isolation managed by the RDSM. Note that Smsdedbg and Smsdetrc may be used as independent extensions even when a single supervisor domain is being utilized on a platform.

The configuration for these settings for a supervisor domain is expected to be obtained from the manifest/configuration of the supervisor domain and must be reported in the attestation report for the supervisor domain and its workloads. The RDSM should manage this supervisor domain configuration state for context switching using memory isolated from all untrusted supervisor domains.

QoS, Performance Monitoring, RAS Isolation

With isolated supervisor domains, the resource accesses from a supervisor domain or the RDSM must not be observable by entities that are not within their TCB using the resource usage monitors. Similarly, the resource allocations for a supervisor domain or the RDSM must not be influenced by entities outside their TCB. To support this security objective, the following capabilities are supported by the Capacity and Bandwidth QoS Register Interface (*CBQRI*) non-ISA extension and *Smqosid* ISA extension for supervisor domains.

The non-ISA supervisor domain *CBQRI* extension is specified to enable the RDSM to allocate a QoS register interface (QRI) exclusively to a supervisor domain from multiple QRIs in the capacity and bandwidth controllers within the SoC. The RDSM may also mediate access to a QRI among multiple supervisor domains.

The *Smqosid* ISA extension provides the second capability to associate a QRI identifier (QRID) with requests originating from supervisor domains and the RDSM. The QRID, along with the Resource Control identifier (RCID) and AT, is used to identify the resource allocation configurations in a capacity or bandwidth controller. The QRID, along with the Monitoring Counter Identifier (MCID), is used to identify the ID of the counter used to monitor the resource usage in a capacity or bandwidth controller. The Smqosid extension enables the scenario where two or more security domains share a common set of capacity and bandwidth controllers. In this setup, access to the QRI of these controllers is restricted to RDSM, which then provides mediated access for these security domains to configure capacity or bandwidth allocation and read the corresponding monitoring counters. A common QRID is associated with requests from these domains, and the RDSM allocates sets of RCID and MCID for each domain's use. The RDSM is expected to configure these fields such that each SD can select from a disjoint range of values for RCID and MCID. The RDSM may then delegate configuration of RCID and MCID within those disjoint ranges by configuring a set of length fields to restrict malicious or mis-configuration by a supervisor domain.

Memory Encryption and Additional Mechanisms

Additional protection/isolation for memory associated with a supervisor domain is orthogonal (and usage-specific). Such additional protection for memory may be derived by the use of cryptography and/or access-control mechanisms, and they may be linked to the supervisor domain identifier (SDID) programmed by the RDSM. Attacks such as breaching confidentiality, integrity using replay-protection should be considered - refer to [3]. The mechanisms chosen for these additional protection methods are otherwise independent of the supervisor domain ISA and non-ISA extensions and may be platform-specific. The TCB of a particular supervisor domain (and devices that are bound to it) may be independently evaluated via attestation of the HW and SW TCB by a relying party using standard Public-Key Infrastructure-based mechanisms and standards-based attestation frameworks e.g. IETF RATS [8].

Scalability

Scalability Aspect	Properties
Number of supervisor domains	Unlimited
ISA complexity	Reduced complexity of new ISA and non-ISA through utilization of existing RISC-V privileged ISA and non-ISA mechanisms, augmented with modular extensions as necessary
Resource assignment	Configurable assignment of IOMMU, supervisor interrupt domains and devices to supervisor domains. E.g. enables heterogeneous isolated processing
Application compatibility	No modification of application workloads to enforce isolation and other security properties.
Memory management	Composes with Sv and H-extension page table modes to support sophisticated memory management algorithms.
Memory isolation	Isolated supervisor domains may dynamically manage memory exclusively assigned or shared between supervisor domains. Memory may be managed at architectural page sizes (1 GiB, 2 MiB, 4 KiB etc.).
Workload diversity	Supervisor domain uses existing privileged ISA and adds extensions that support modes that allow for embedded devices or for data-center application processors. Rich OS environment support allows for workload abstractions as used today.
Trust models	The supervisor domain architecture does not specify policies of trust models - it enables applications of various trust models and security usage requirements. Some use cases may require additional platform capabilities - see the more detailed security model specification for requirements [3].
Platform lifecycle and operational phases	Support all phases of platform operation, including debugging, performance analysis, and operational deployment. Supervisor domain architecture allows for isolation properties to be implemented while co-existing with external debug and trace controls by providing the RDSM with controls to enforce per security requirements

Software Interfaces

GlobalPlatform-compliant TEE

The goal of the GlobalPlatform TEE Software Architecture [18] is to enable Trusted Applications (TAs) to provide isolated and trustworthy capabilities, which can then be used through Client Applications (CAs). Via the GlobalPlatform architecture, a Trusted Execution Environment (TEE) running a TEE OS is designed as a companion to a non-secure REE OS (e.g. Linux kernel) to host trusted applications (TAs). The TA interface with the TEE Internal Core API v1.3.1 which is the API exposed to Trusted Applications and the TEE Client API v1.0, which is the API describing how to communicate with a TEE. Those APIs are defined in the GlobalPlatform API [7] specifications. The non-secure OS is referred to as the Rich Execution Environment (REE) in TEE specifications. It is typically a Linux OS flavor as a GNU/Linux distribution or the AOSP. There is an active RISE project for a GlobalPlatform-compliant TEE on RISC-V that aims to leverage the PMP, IOPMP, Sv and Smmpt extensions [9].

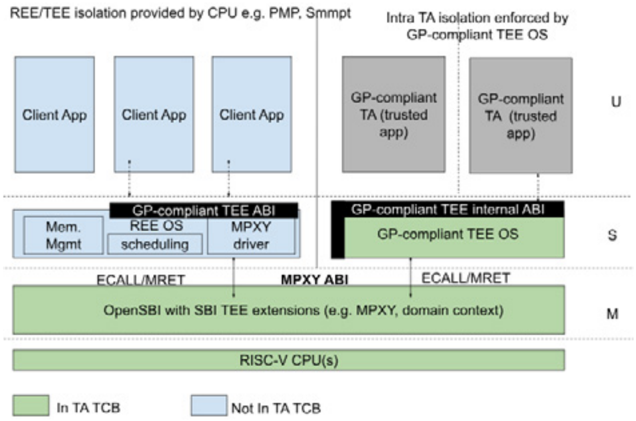
REE/TEE isolation provided by CPU e.g. PMP, Smmpt
Intra TA isolation enforced by GP-compliant TEE OS
Client App
Client App
Client App
GP-compliant TA (trusted app)
GP-compliant TA (trusted app)
U
GP-compliant TEE ABI
Mem. Mgmt
REE OS scheduling
MPXY driver
GP-compliant TEE internal ABI
GP-compliant TEE OS
S
ECALL/MRET
MPXY ABI
ECALL/MRET
OpenSBI with SBI TEE extensions (e.g. MPXY, domain context)
M
RISC-V CPU(s)
In TA TCB
Not In TA TCB

Figure 8: Global Platfrom TEE Software Architecture

RISC-V CoVE (Confidential VM Extension)

CoVE is an ABI to host confidential VM workloads on a RISC-V platform with Hypervisor extension so that the hosting OS and hypervisor (and VMM) can be excluded from the TCB of hosted TEE Virtual Machines (TVMs). The hosting domain is a rich-OS environment such as Linux and the VMM such as KVM. The confidential supervisor domain is managed by a TEE Security Manager (TSM) which operates at a privilege level HS or higher e.g. an H-extension-aware runtime such as Salus [14]. The figure below shows the reference architecture, however other deployment models are possible, with some described in the CoVE specification [6].

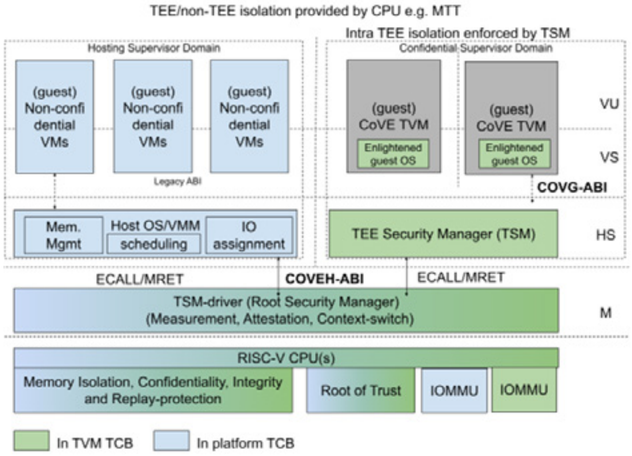
TEE/non-TEE isolation provided by CPU e.g. MTT
Intra TEE isolation enforced by TSM
Hosting Supervisor Domain
Confidential Supervisor Domain
(guest) Non-confi dential VMs
(guest) Non-confi dential VMs
(guest) Non-confi dential VMs
Legacy ABI
(guest) CoVE TVM
Enlightened guest OS
(guest) CoVE TVM
Enlightened guest OS
VU
VS
COVG-ABI
Mem. Mgmt
Host OS/VMM scheduling
IO assignment
TEE Security Manager (TSM)
HS
ECALL/MRET
COVEH-ABI
ECALL/MRET
TSM-driver (Root Security Manager) (Measurement, Attestation, Context-switch)
M
RISC-V CPU(s)
Memory Isolation, Confidentiality, Integrity and Replay-protection
Root of Trust
IOMMU
IOMMU
In TVM TCB
In platform TCB

Figure 9: CoVE ABI

The CoVE ABI [6] specifies two main interfaces:

- COVH – ABI between OS/VMM and the TSM. COVH provides interfaces for: TSM and TVM Measurement and Remote-attestation, Memory Conversion between Domains, TVM HW state isolation & execution, Interrupt Mgmt and Debug/Performance. For systems without a network connection, CoVE specifies a local attestation mechanism.
- COVG – ABI between the TVM and the TSM. COVG provides interfaces for extending dynamic measurements, and getting attestation credentials.

There is an active RISE project for CoVE changes related to Linux/KVM [9].

Call to Action and Timeline

- Participate in development of HW emulation platform for Smmpt via QEMU in RISE security WG - targeting ratification of ISA and non-ISA in 2025 (Q3).
- Participate in open source secure implementations of the RDSM via openSBI and RISE projects - task items target for end of year (2025).
- Participate in open source projects such as GlobalPlatform-compliant TEE and RISC-V CoVE to initiate public RFC discussions by the end of this year (2025).
- Participate in reviewing the RISC-V security model - currently in freeze review.

References

1. RISC-V Privileged ISA:
   https://github.com/riscv/riscv-isa-manual/releases/download/20240411/priv-isa-asciidoc.pdf
2. RISC-V Supervisor domains (Priv ISA extension) proposed specification:
   https://github.com/riscv/riscv-smmtt/releases/download/v0.1.2/smmtt-spec.pdf
3. RISC-V Security Model - proposed non-ISA specification:
   https://github.com/riscv-non-isa/riscv-security-model/releases/download/vtheme/riscv-platform-security-model.pdf
4. RISC-V IOMMU:
   https://github.com/riscv-non-isa/riscv-iommu/releases/download/v1.0.0/riscv-iommu.pdf
5. RISC-V Advanced Interrupt Architecture (AIA):
   https://github.com/riscv/riscv-aia/releases/download/1.0/riscv-interrupts-1.0.pdf
6. CoVE ABI - proposed non-ISA specification:
   https://github.com/riscv-non-isa/riscv-ap-tee/releases/download/v0.7/riscv-cove.pdf
7. GlobalPlatform API:
   https://optee.readthedocs.io/en/stable/architecture/globalplatform_api.html#globalplatform-api
8. IETF RATS:
   https://datatracker.ietf.org/doc/rfc9334/
9. RISE TEE projects - CoVE & GlobalPlatform-compliant TEE:
   https://lf-rise.atlassian.net/wiki/spaces/HOME/pages/8590760/LK_02_012+-+KVM+CoVE+host+support
   https://lf-rise.atlassian.net/wiki/spaces/HOME/pages/8586860/SBI_00_05+-+OpenSBI+RPMI+MM+Support
10. Confidential Computing Consortium:
    https://confidentialcomputing.io/
11. CoVE-IO ABI:
    https://github.com/riscv-non-isa/riscv-ap-tee-io/releases/download/v0.2.0/riscv-cove-io-v0.2.0.pdf
12. RISC-V non-ISA IOPMP:
    https://github.com/riscv-non-isa/iopmp-spec/releases
13. RISC-V Supervisor PMP (privileged ISA extension) proposed specification:
    https://github.com/riscv/riscv-spmp/releases
14. RISC-V hypervisor for TEE development (Salus):
    https://github.com/rivosinc/salus
15. Worldguard specification 0.4 - proposed fast track for ISA (Smwg):
    https://lists.riscv.org/g/security/attachment/711/0/worldguard_rvia_spec-v0.4.pdf
16. UEFI 2.11 Secure boot:
    https://uefi.org/specs/UEFI/2.11/32_Secure_Boot_and_Driver_Signing.html
17. NIST Special Publication 800-53 Revision 5:
    https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf
18. GlobalPlatform TEE System Architecture:
    https://globalplatform.org/wp-content/uploads/2022/05/GPD_SPE_009-GPD_TEE_SystemArchitecture_v1.3_PublicRelease_signed.pdf