

CoVE in Action

The landscape of
confidential VMs on RISC-V

Ruoqing He

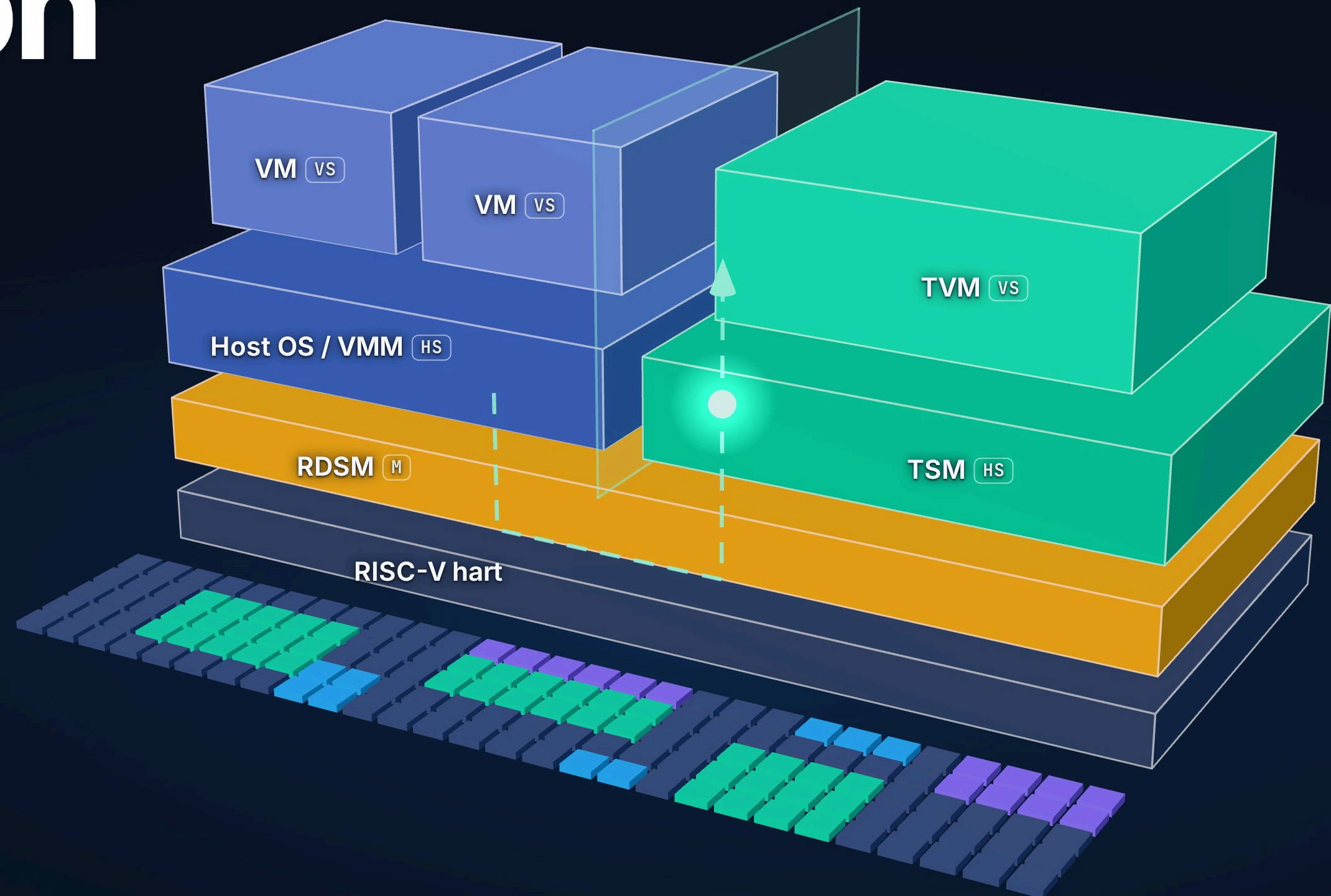
LingCage

Xiaoxia Cui

Damo

Linux Plumbers Conference 2026

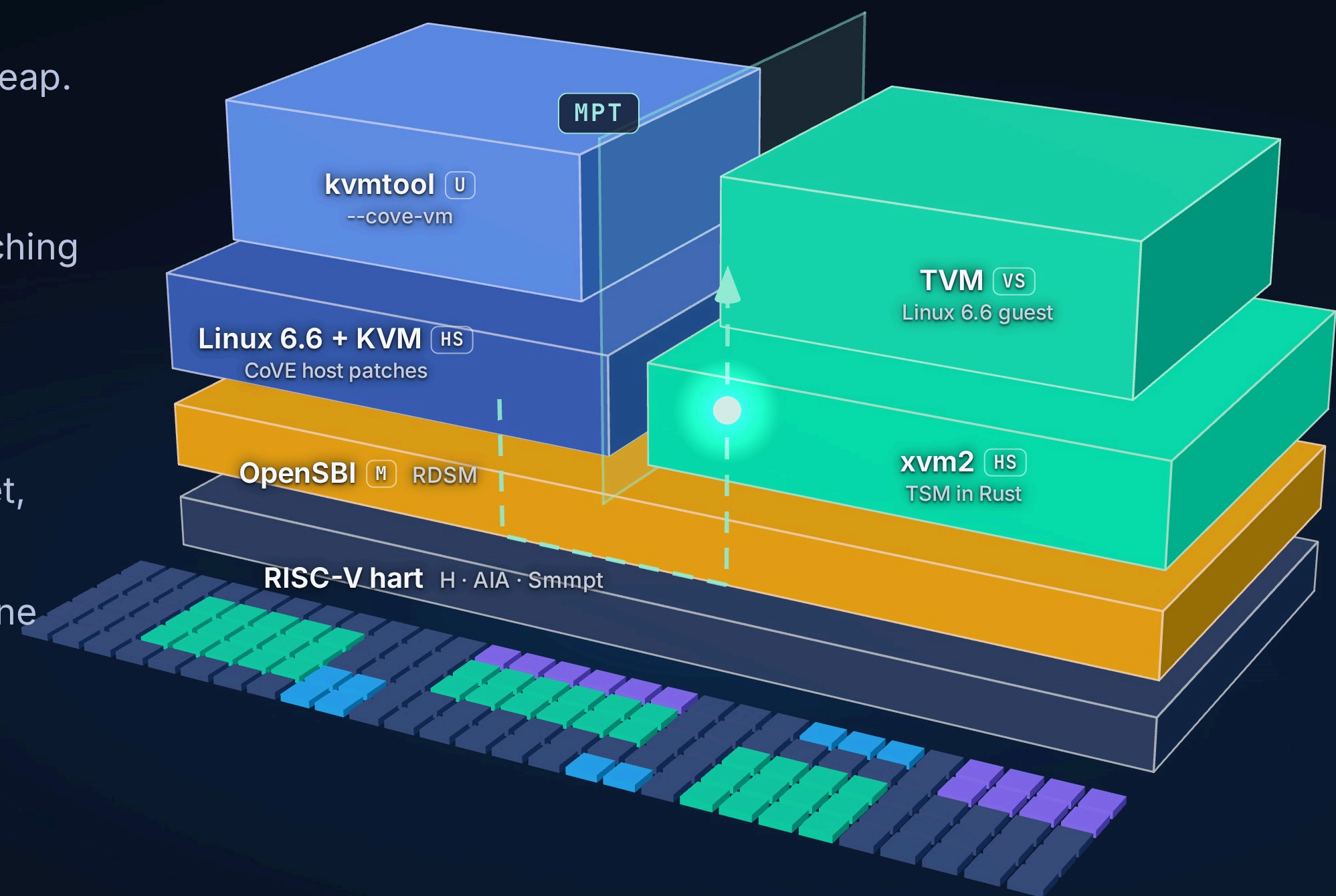
Confidential Computing MC · Prague



Full CoVE DM1 stack boots end to end

- TSM** **xvm2**, written in Rust with `no_std` and no heap. It implements CoVE v0.7 ABI and is open source.
- RDSM** OpenSBI, extended to support domain switching and MPT management.
- HOST** Linux 6.6 with CoVE host patches, KVM as hypervisor and kvmtool as VMM.
- GUEST** Linux 6.6 with CoVE guest patches. virtio net, blk and 9p work over shared pages.
- RUNS ON** Real RISC-V silicon, and QEMU `virt` machine with Smmpt.

Limits for now: one TVM, one vCPU, measurement not implemented yet.



Our PoC

Our PoC is open source. It can be built and run on QEMU with the three commands below.

```
$ ./build.sh tsm
$ ./build.sh run
host # ./run_tvm.sh
```



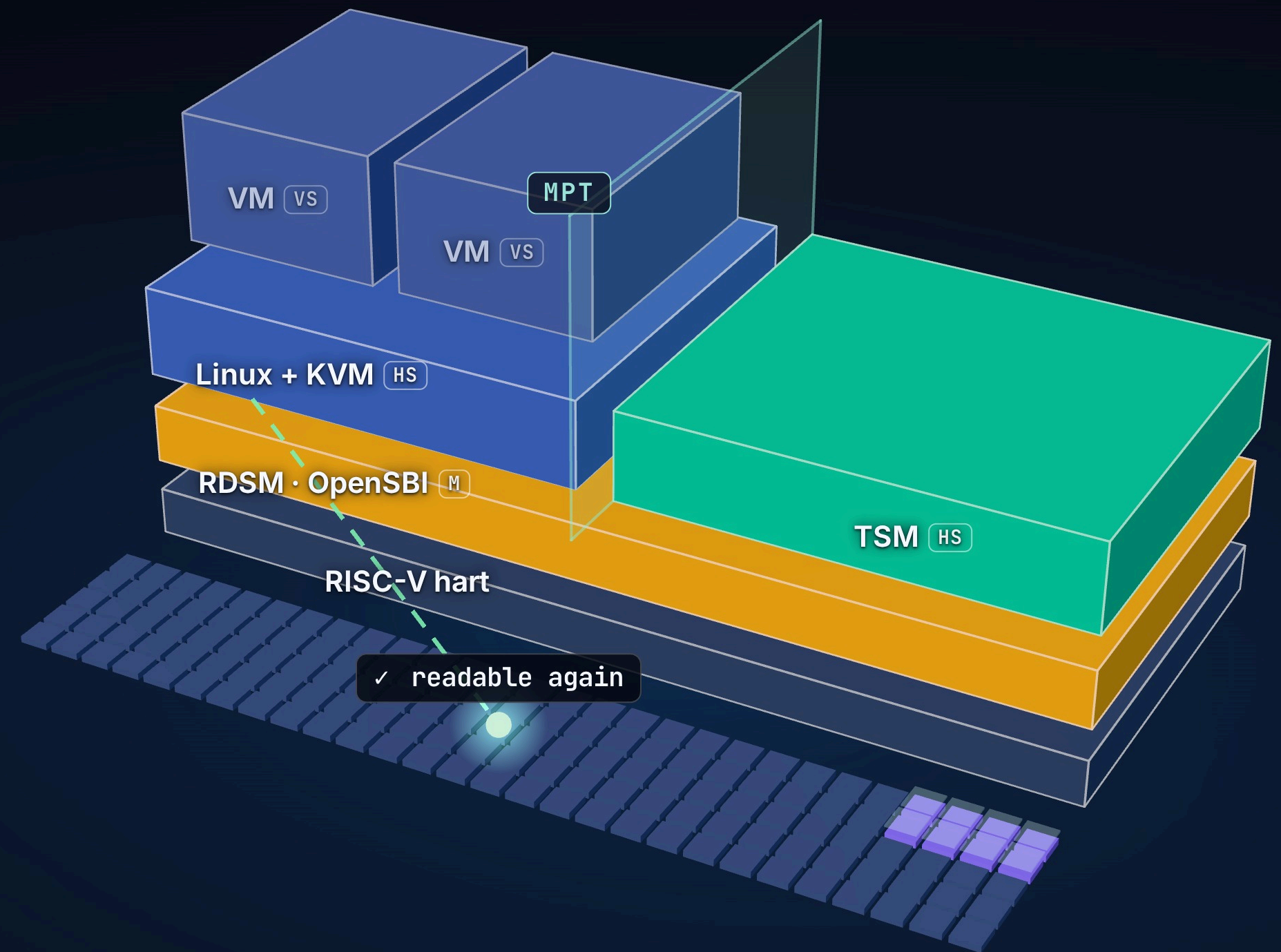
github.com/XUANTIE-RV/cove-dp1-poc

Source code of TSM is in the repository. QEMU and runtime bundle are provided on the release page.

```
# read memory on the host, as root
host # memtool md 0xA0200000
× load access fault, it is memory of TSM

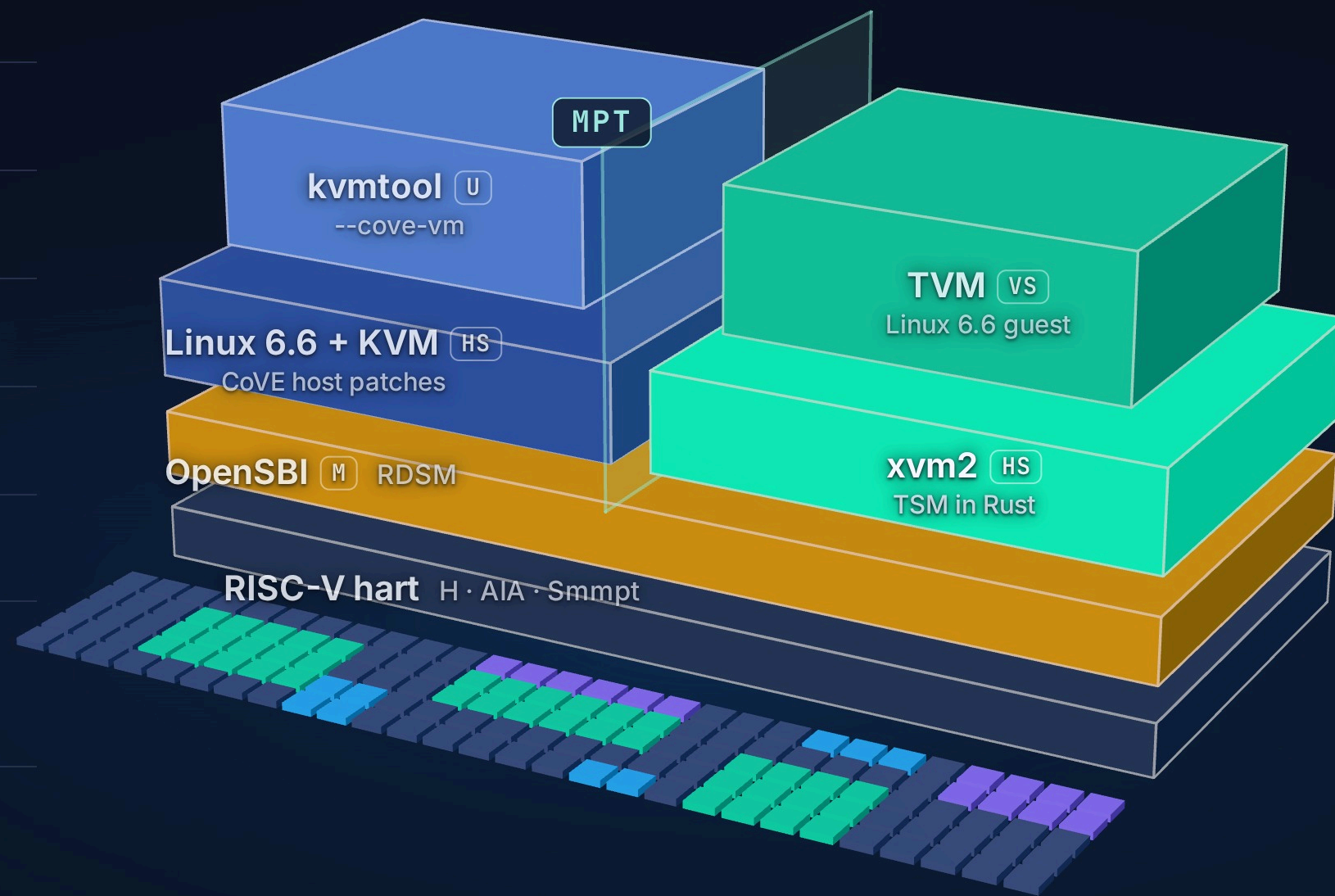
host # memtool md <a page of TVM>
× load access fault while the TVM is running

host # memtool md <same page>
✓ readable again after the TVM is destroyed
```



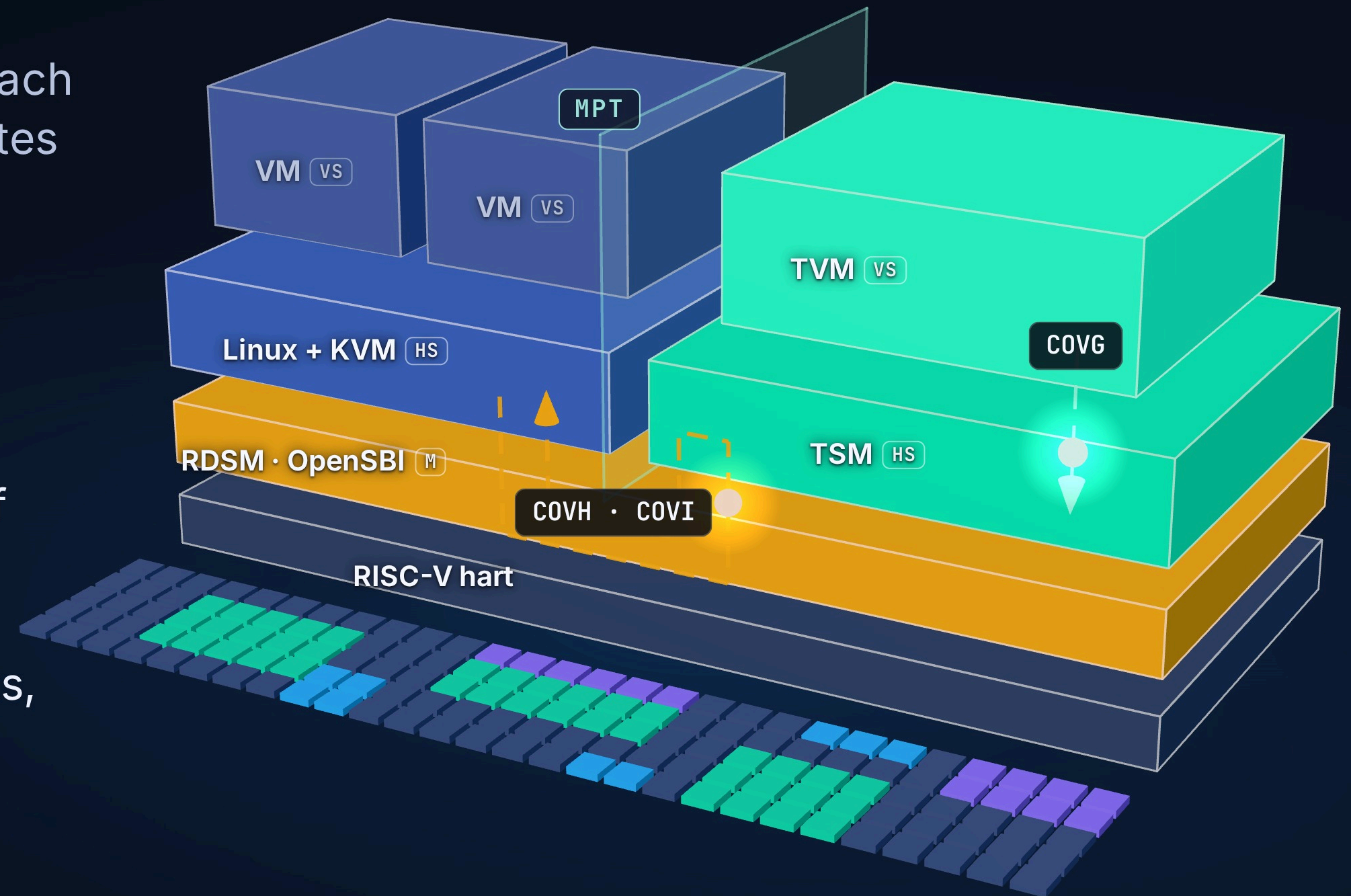
Upstream status of each layer

	UPSTREAM STATUS
Guest kernel	Out of tree. Mainline RISC-V does not support confidential guest yet.
kvmtool	Out of tree, with <code>--cove-vm</code> option.
KVM host	Out of tree. Last posted as RFC in April 2023.
TSM	xvm2 is open source since September 2026.
OpenSBI	Our RDSM changes are not upstream yet.
QEMU	Smmpt series v8 is in review. We provide a XuanTie build.
Specs	CoVE v0.7 and Supervisor Domains are in "Development" state.



DM1: TSM is a peer of host KVM

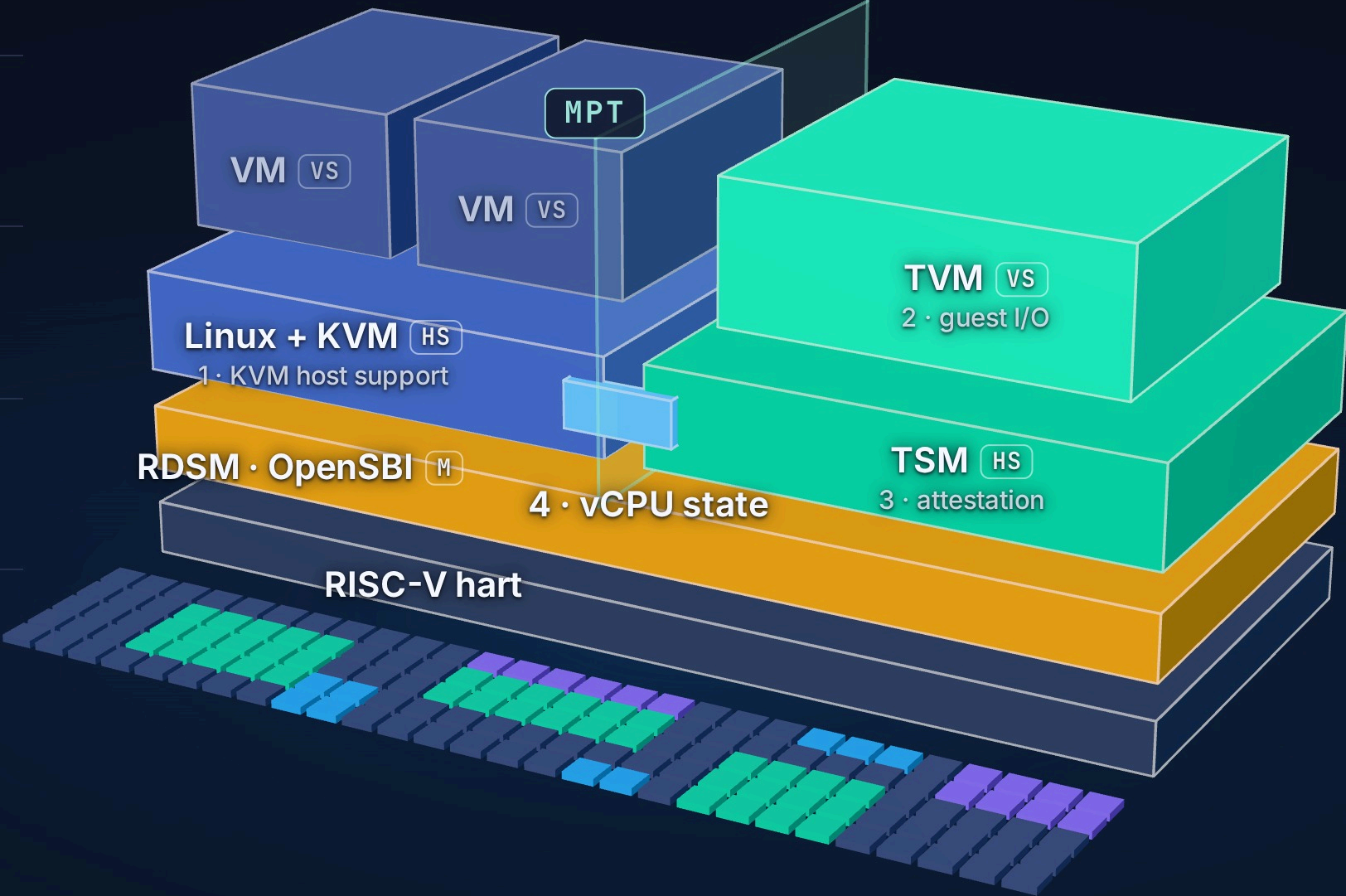
- Linux/KVM and TSM both run in **HS-mode**, each in its own supervisor domain. Hardware isolates them with **MPT**. Host is not nested.
- Each call from host to TSM goes through **RDSM** in M-mode, **TEECALL** to enter and **TEERET** to return.
- TSM manages page tables and vCPU state of the TVM. KVM still allocates memory and schedules vCPUs, through three SBI extensions: **COVH** for host, **COVI** for interrupts, **COVG** for guest.



Four problems we met on the way

PROBLEM		EXPECTED OUTCOME
1	KVM host support COVH · COVI	Rough consensus on upstreaming order
2	Guest I/O COVG · share and unshare	Direction for sharing, by region or by page state
3	Attestation COVG · evidence	People interested in evidence format work
4	vCPU state between KVM and TSM	Direction for the state area, NACL layout or a run page

For each problem we show findings first, then our plan and questions.
Problems 1 to 3 are not specific to DM1. Problem 4 shows up since host is not nested in DM1.



KVM host support: changes since 2023 RFC

2023 RFC

Depends on three series which were not upstream: IPI, AIA, NACL

TODO: fd-based private memory interface

Interrupts rely on AIA with IMSIC guest files

TODO: complete PoC for DM1

Tested with host nested under Salus

The RFC has 48 kernel patches in four groups: cleanups, host side, interrupts and guest side.

OUR STACK TODAY

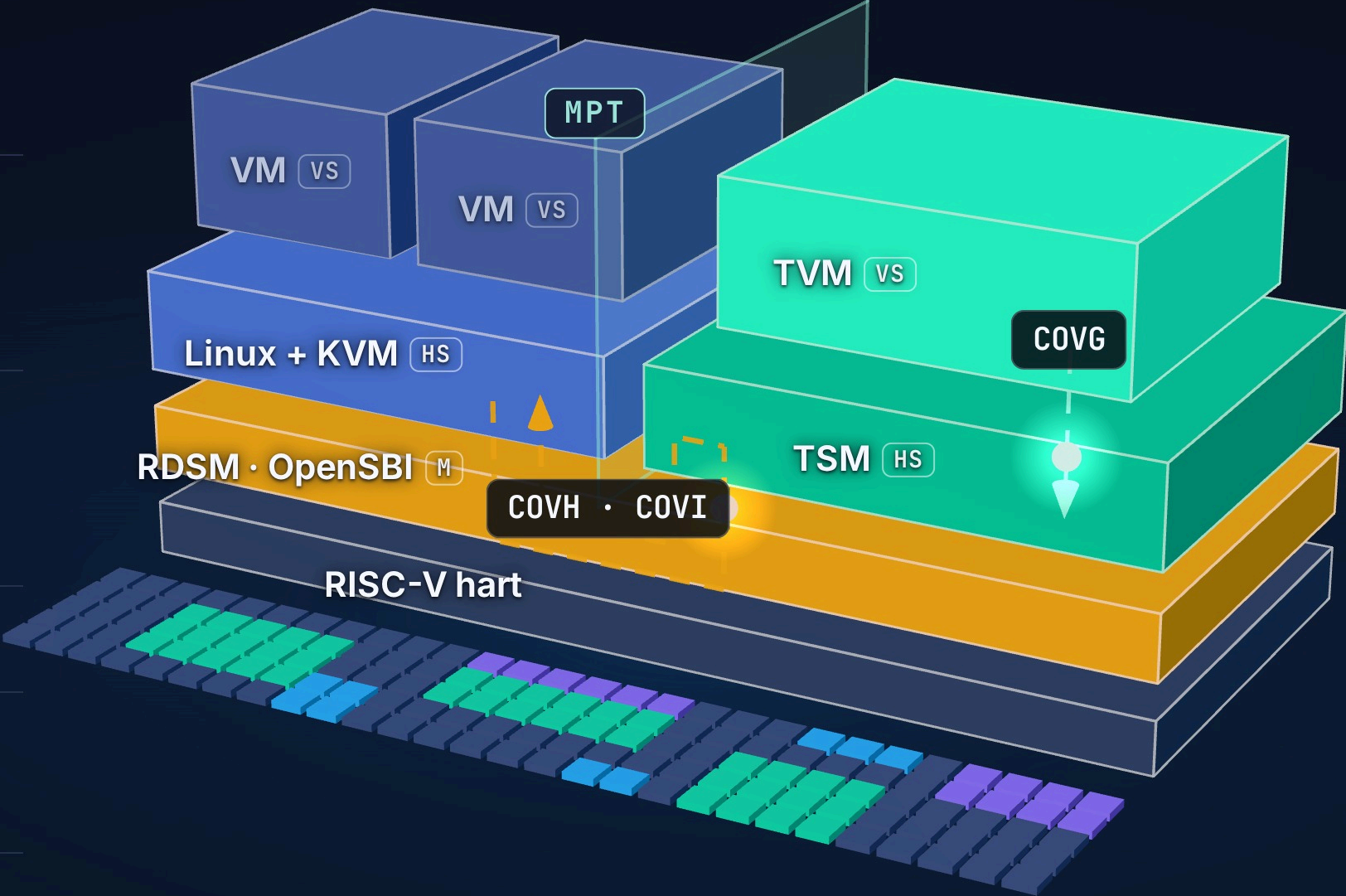
All three are in mainline now

`guest_memfd` is available now. We still pin anonymous pages and convert one 4K page per fault

Same. Our TSM cannot move a guest file yet, so each vCPU is pinned to its hart

This is the stack which boots today

Host is not nested, which exposes one ABI gap, see problem 4



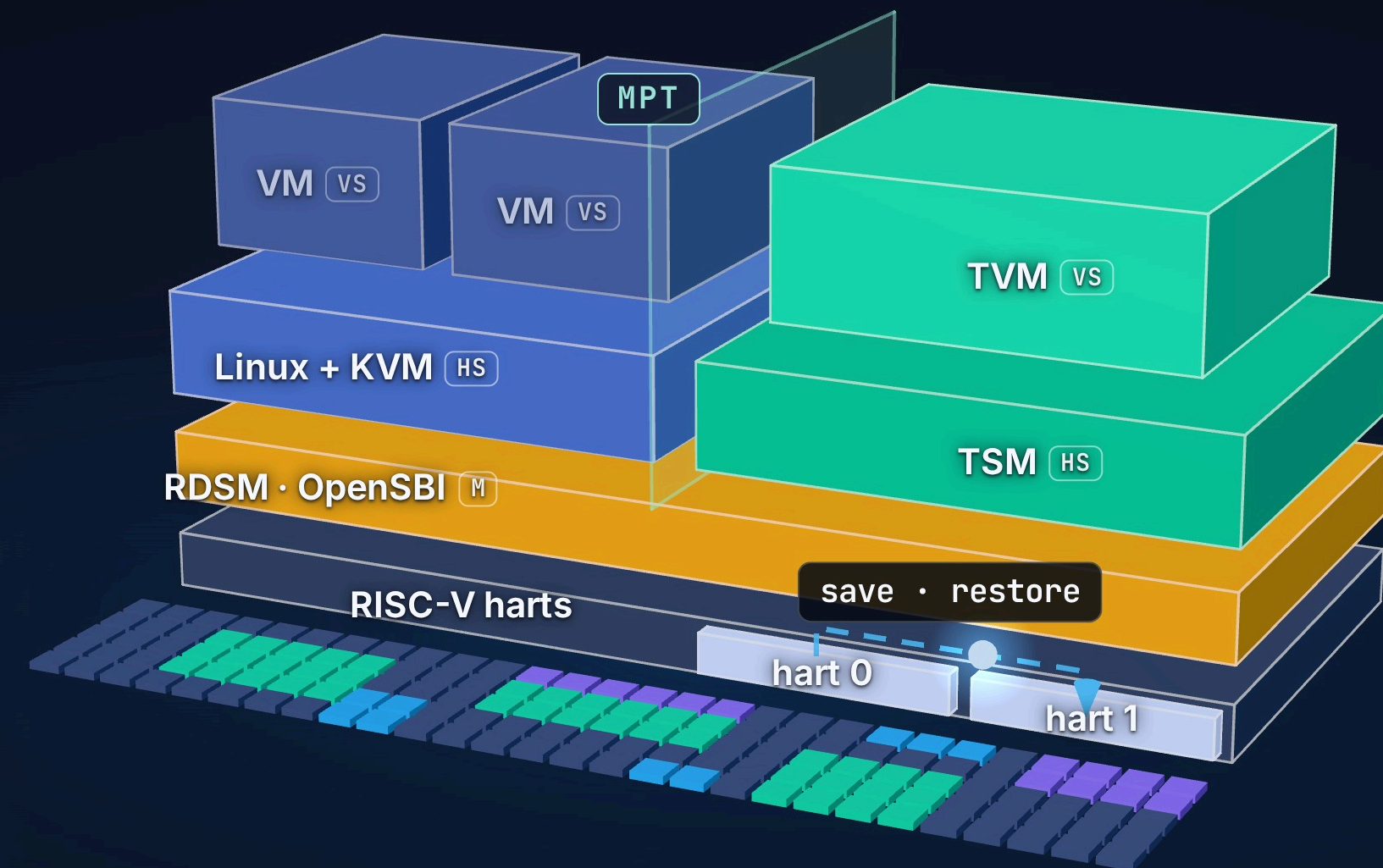
KVM host support: proposed upstreaming order

OUR PLAN

- Freeze spec one extension at a time, starting with **COVG** for guest.
- Upstream guest side first as TDX, SNP and CCA did, since it only depends on **COVG**.
- Then host side, rebased from 6.6 to mainline. Interrupt support comes last.

OPEN QUESTIONS

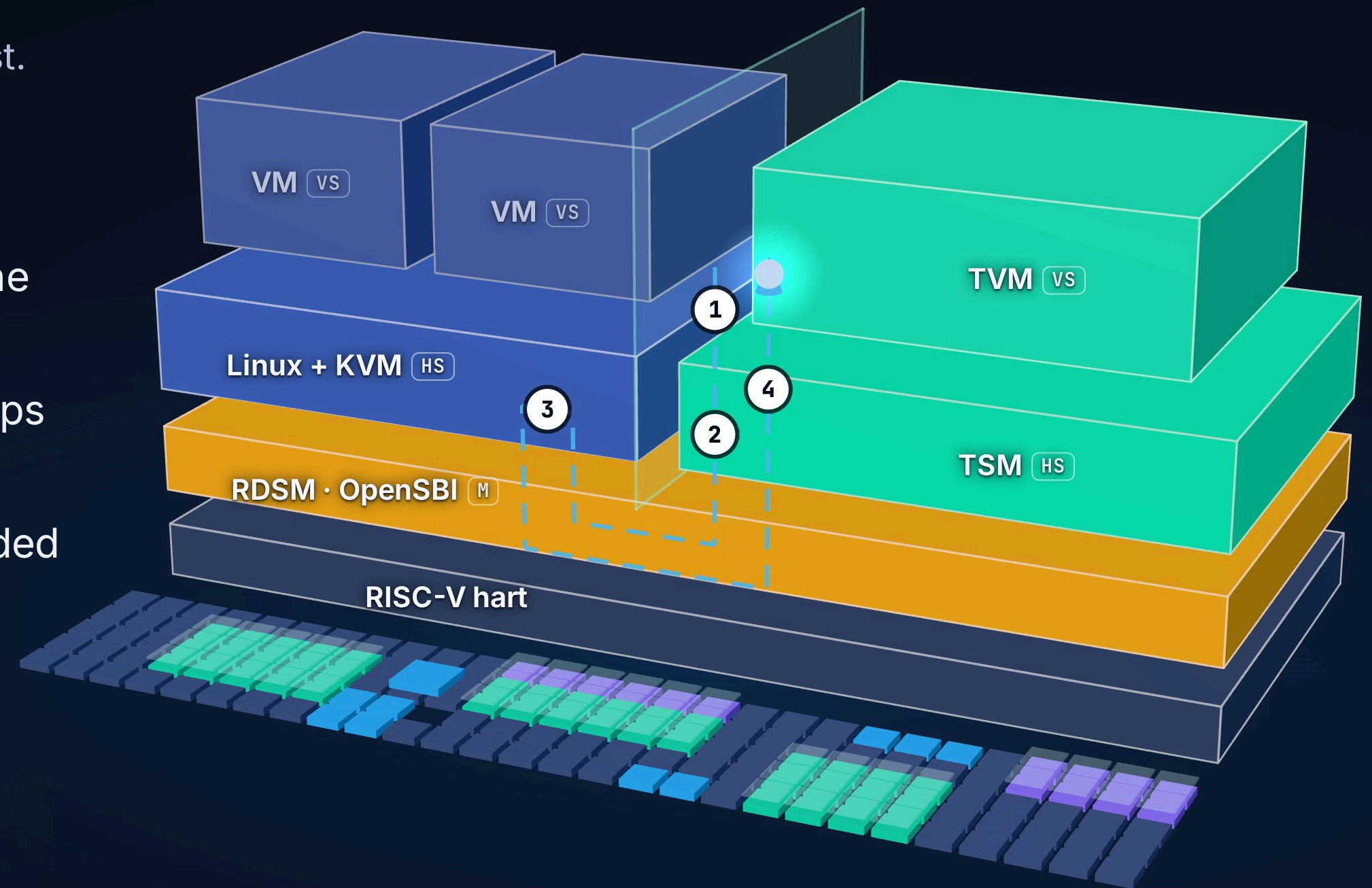
- Is **guest_memfd** required for private memory? Is a first version with pinned pages acceptable to upstream?
- Is it acceptable to keep each vCPU pinned to its hart in a first version?



Guest I/O: path of one share request

I/O is bounced through swiotlb, a pool shared with the host.

- 1 `set_memory_decrypted()` in guest results in `ShareMemoryRegion`. Calling vCPU is blocked.
- 2 TSM checks the range, records it, and forwards the call to the host.
- 3 KVM removes private pages in the range, then maps a shared page with `AddTvmSharedPages`.
- 4 TSM accepts the page only if it is inside the recorded range.



TDX, CCA

Guest sets state of pages, `MapGPA` or `RSI_IPA_STATE_SET`.

CoVE

Guest declares regions. Our TSM tracks **16** regions and merges adjacent ones.



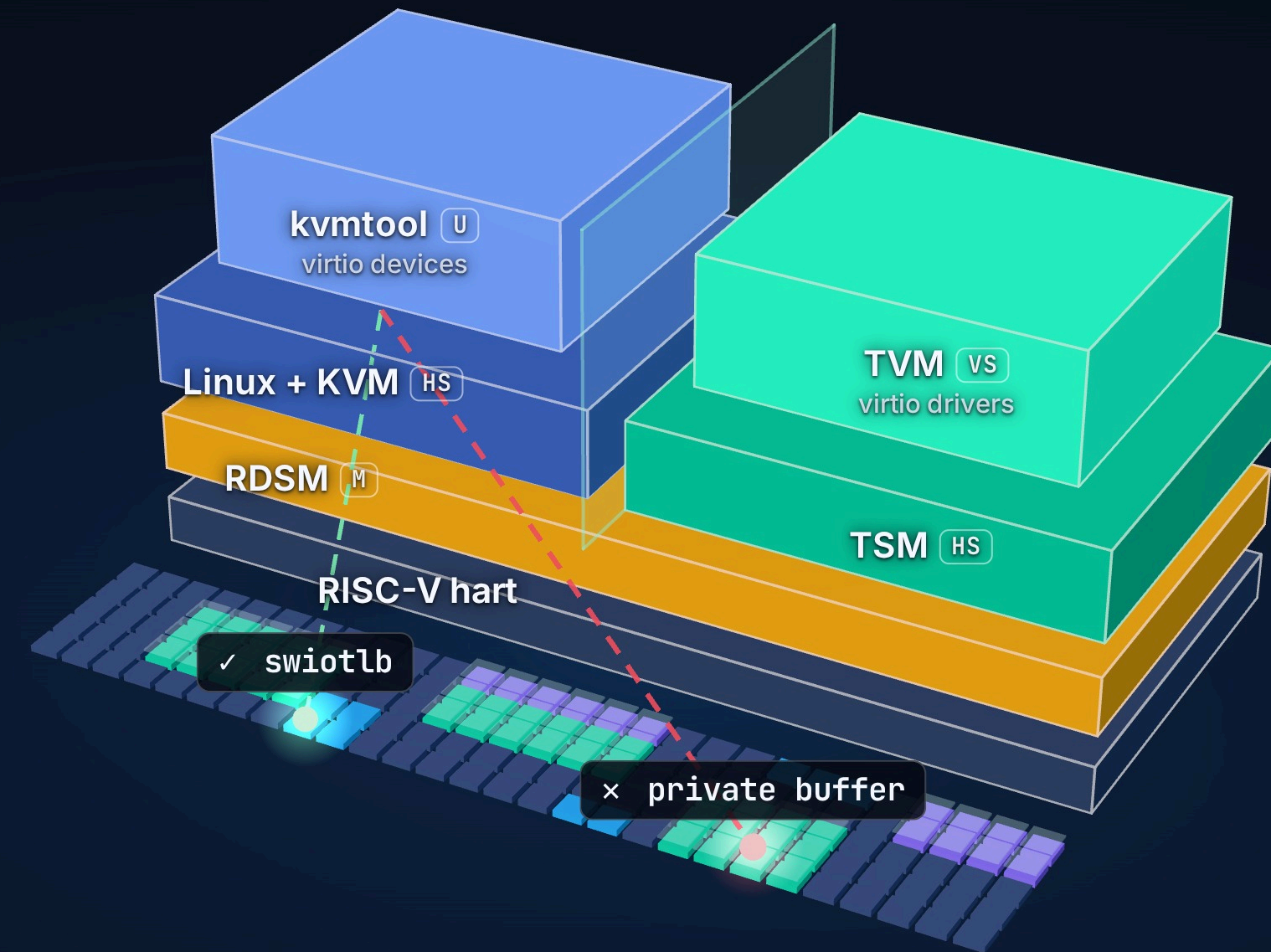
Guest I/O: reuse common code, one semantic does not fit

OUR PLAN

- Reuse common code as is: `cc_platform_has()`, `set_memory_decrypted()`, `swiotlb`, `virtio` with `ACCESS_PLATFORM`.
- RISC-V only adds arch hooks, as x86, arm64 and s390 did.
- One mismatch: the kernel shares memory one allocation at a time, while CoVE shares by region. Our TSM has to merge ranges, otherwise the guest fails to boot.

OPEN QUESTIONS

- Should `ShareMemoryRegion` become an idempotent page state change, as on TDX and CCA?
- Are there other assumptions of common code which do not fit a region model?



Attestation: status of each component from TSM to verifier

	TDX, SNP AND CCA	COVE
Evidence	Defined by each vendor. CCA uses EAT, TDX with DICE returns CWT.	Three chained EATs in spec, profile and labels are "TBD". Our TSM does not produce evidence yet.
Guest kernel	One ABI, configs <code>tsm/report</code> , with providers <code>sev_guest</code> , <code>tdx_guest</code> and <code>arm_cca_guest</code> .	No provider yet. Spec defines <code>sbi_covg_get_evidence()</code> .
Guest agent	guest-components of CoCo read this ABI for those three providers.	No attester yet.
Verifier	Trustee has verifiers <code>snp</code> , <code>tdx</code> and <code>cca</code> , with endorsements from vendors.	No verifier yet, and no endorser for root of trust of each SoC.

This is not specific to DM1. Salus (DM2) returned a DICE certificate to its own test guest, ACE (DM3) does local attestation for systems without network. We are not aware of any CoVE stack which reaches a remote verifier.

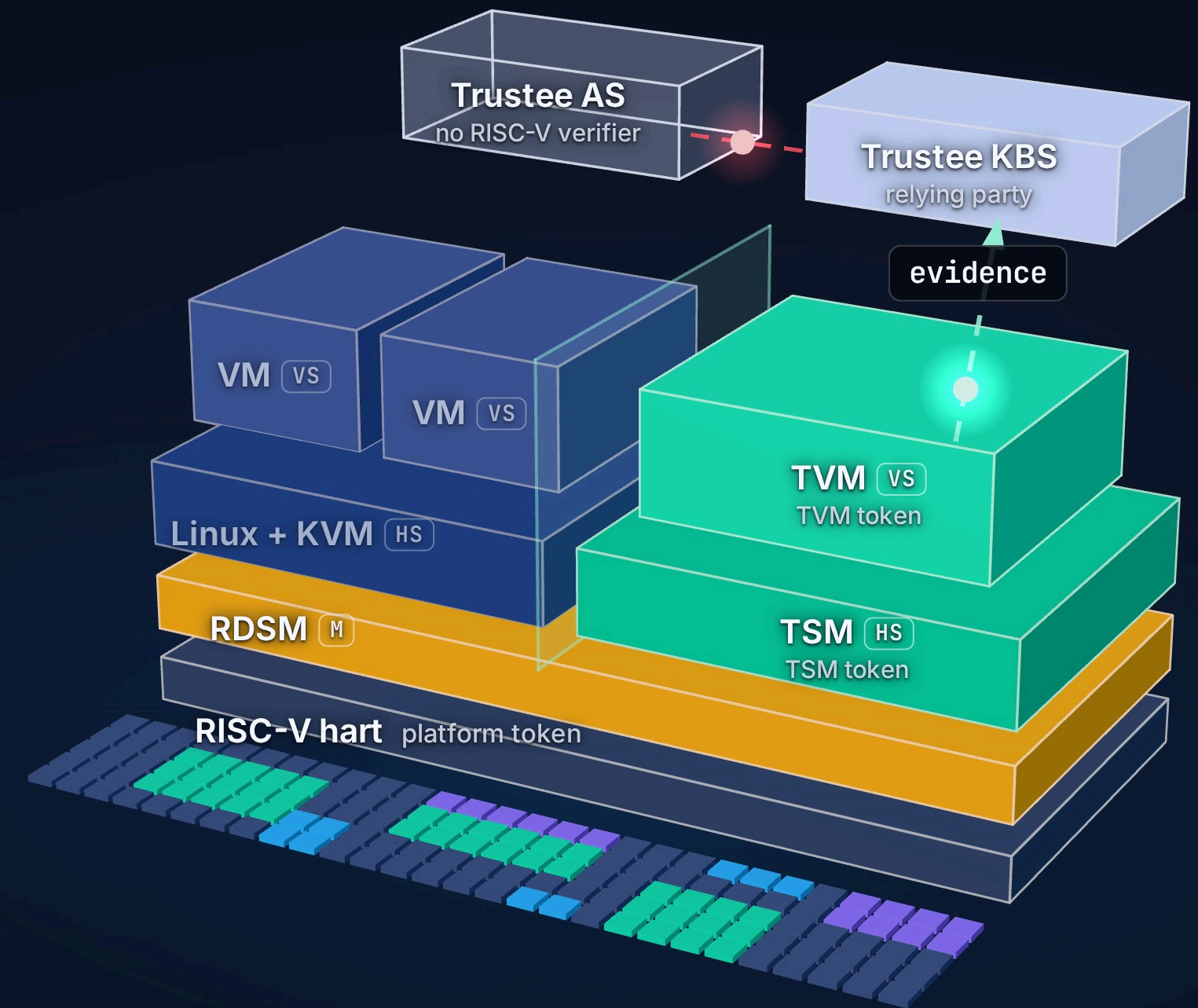
Attestation: reuse kernel ABI, complete evidence format

OUR PLAN

- Kernel: add a provider for common report ABI, configs `tsm/report`, without adding a new interface.
- Format: use three chained EATs defined in spec and complete the profile. CCA token is the closest reference.
- Verifier: add a Trustee plugin after profile is fixed. Our open-source TSM does not compute measurement yet.

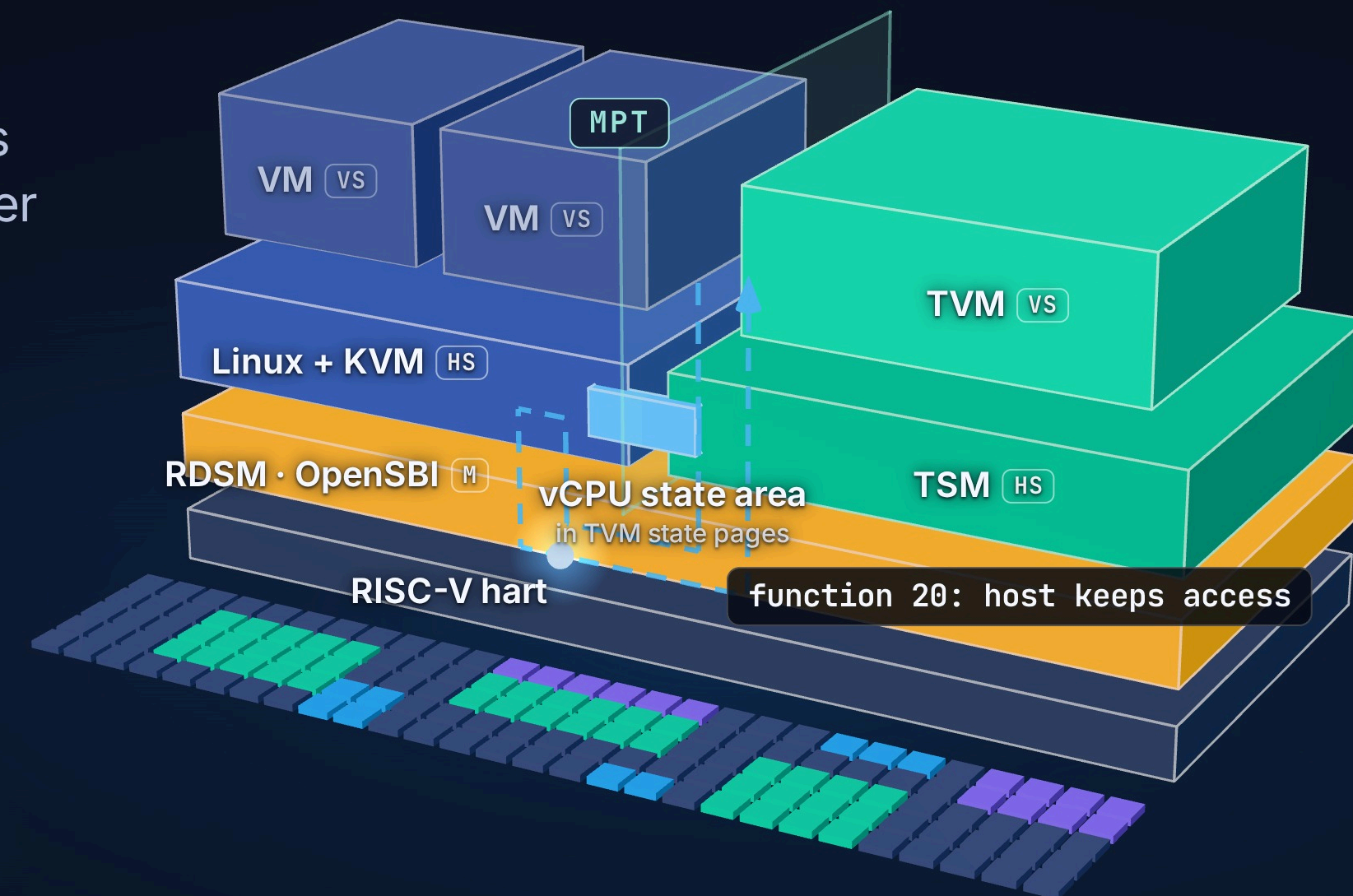
OPEN QUESTIONS

- Who endorses root of trust of each SoC, and who publishes reference values?
- Who is interested in working on the profile with us?



vCPU state: NACL layout comes from nested virtualization

- On each exit KVM needs a few guest registers: fault address, trapped instruction, `a0` to `a7`.
- The spec passes them in **NACL shared memory**, which SBI defines for a nested hypervisor and the one below it. Host was nested under Salus, so 2023 host code says "NACL is mandatory".
- In DM1 the host is not nested. NACL is not provided to it, and the spec does not define how this area is set up.
- Our solution for now: KVM does not require NACL, the area is TVM state pages passed to `CreateTvm`. Spec defines them as confidential, so we need a call which is not in spec to keep them accessible to the host, COVH function 20.



TDX Exit information is returned in registers.

CCA Host passes a run page on each `REC_ENTER`.

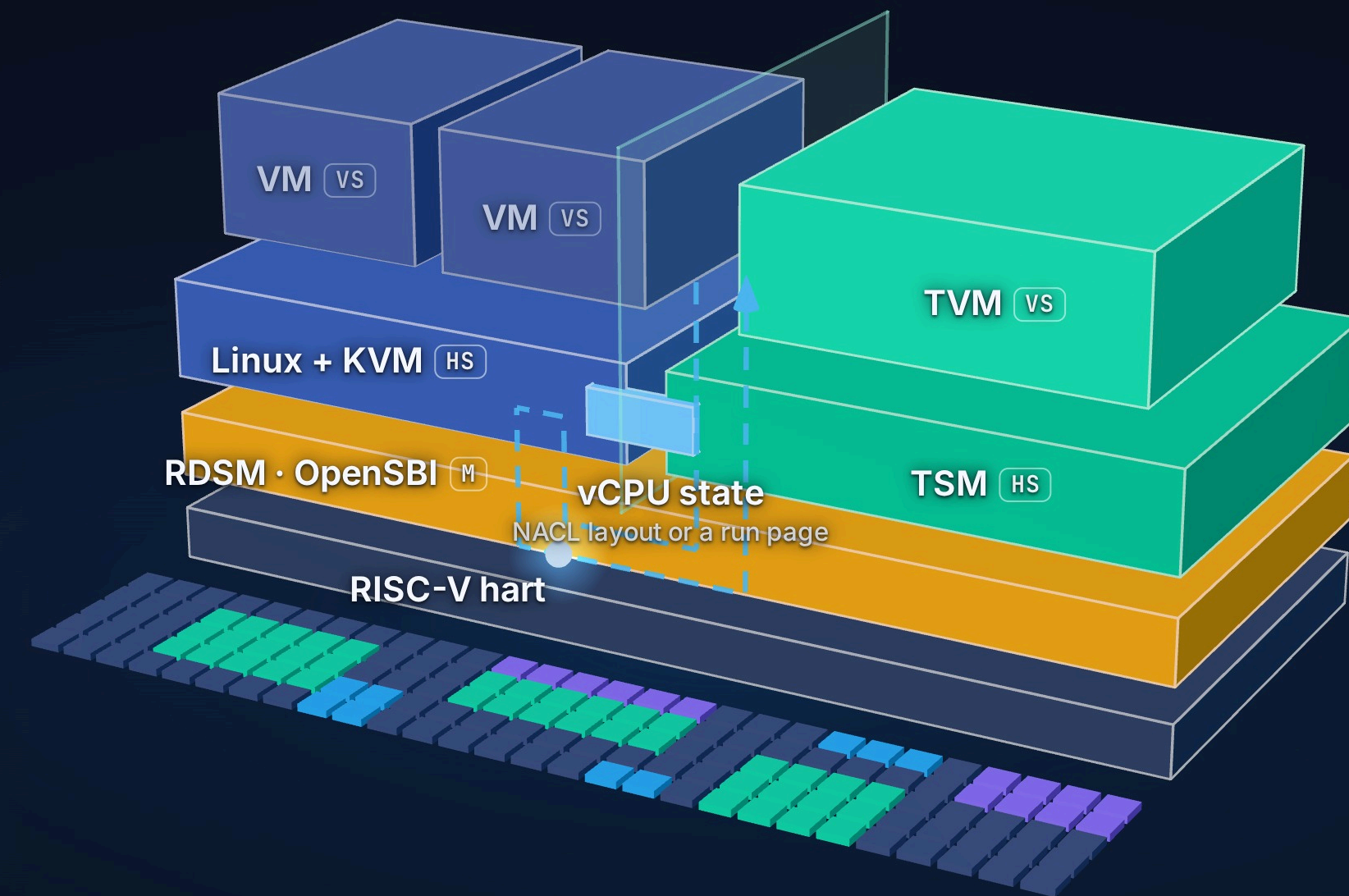
vCPU state: NACL layout or a run page

OUR POSITION

- ABI between KVM and TSM belongs to the spec. KVM should not depend on anything specific to one TSM.
- So function 20 should either be added to spec or be dropped.
- In both cases, the spec needs to define how this area is provided if host is not nested.

OPEN QUESTIONS

- Which option is preferred? **NACL in M-mode firmware**, which SBI spec does not recommend. **A CoVE call** to register the area. Or **a run page** passed on each `RunTvmVcpu`, as CCA does.
- Is anyone still planning a nested host, which NACL was introduced for?



Thank you



github.com/XUANTIE-RV/cove-dp1-poc

Our PoC is open source. Source code of TSM is in the repository, QEMU and runtime bundle are provided on the release page.

