An ephemeral virtual TPM device to allow Remote Runtime Attestation for Confidential Virtual Machines

An ephemeral virtual *TPM device* to allow Remote Runtime Attestation for Confidential Virtual Machines

TPM Devices

- “Measure” key binaries
- Everything you write is immutable until reboot
- Signs everything that it does with a key
- Acts as a Root-of-Trust

An ephemeral virtual TPM device to allow *Remote Runtime Attestation* for Confidential Virtual Machines

Remote Runtime Attestation

- Local Attestation service (“agent”) asks the TPM for “quotes”, i.e. the measures of what it needs to verify
- Remote Attestation services verify that everything is in an expected state, and can perform actions based on the result of the verification

An ephemeral virtual TPM device to allow Remote Runtime Attestation for *Confidential Virtual Machines*

Confidential Virtual Machines

- Trust boundaries redefined
- No trust in the cloud service provider
- Trust is only in the Attestation Report given by the CPU

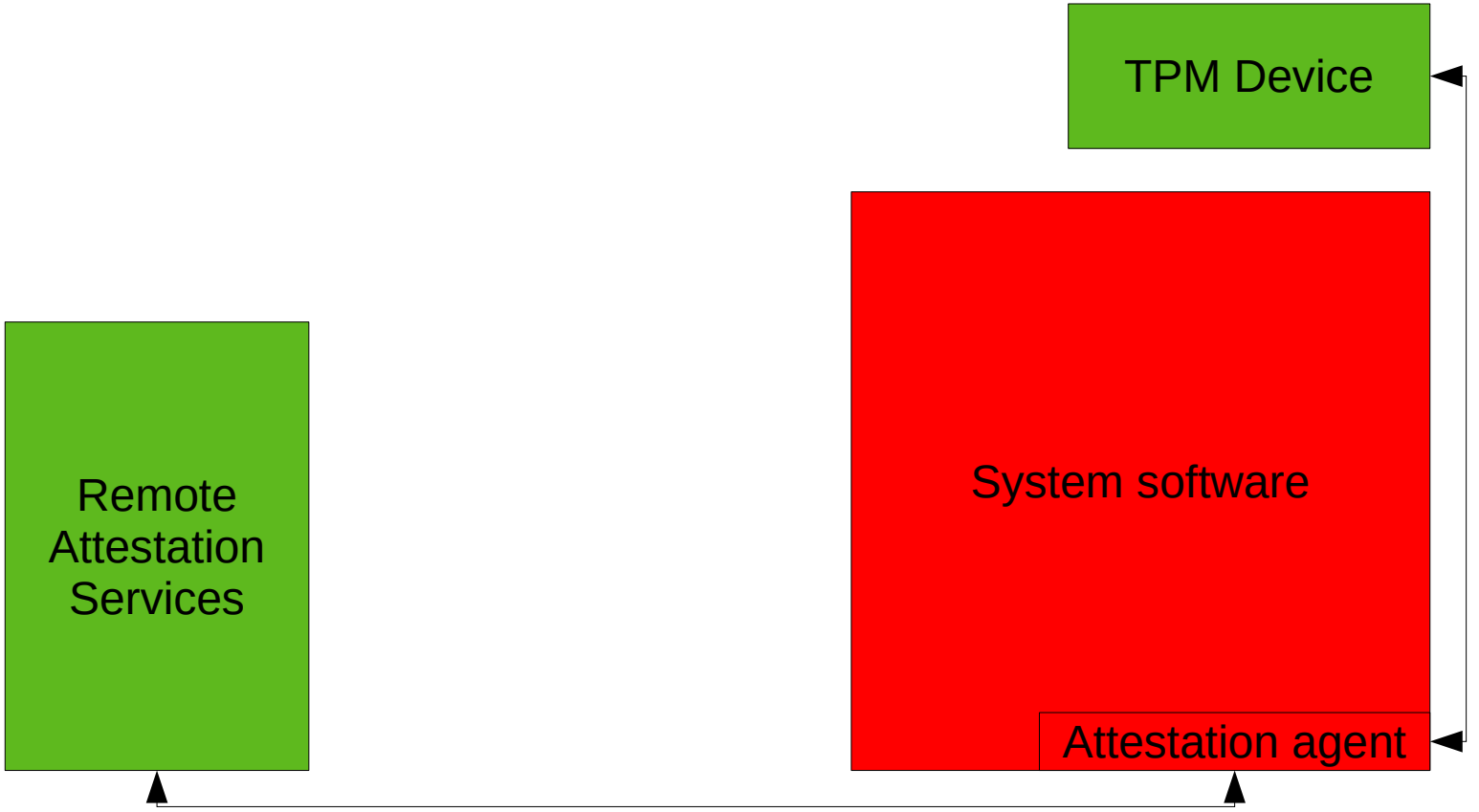
An ephemeral *virtual* TPM device to allow Remote Runtime Attestation for Confidential Virtual Machines

An *ephemeral* virtual TPM device to allow Remote Runtime Attestation for Confidential Virtual Machines

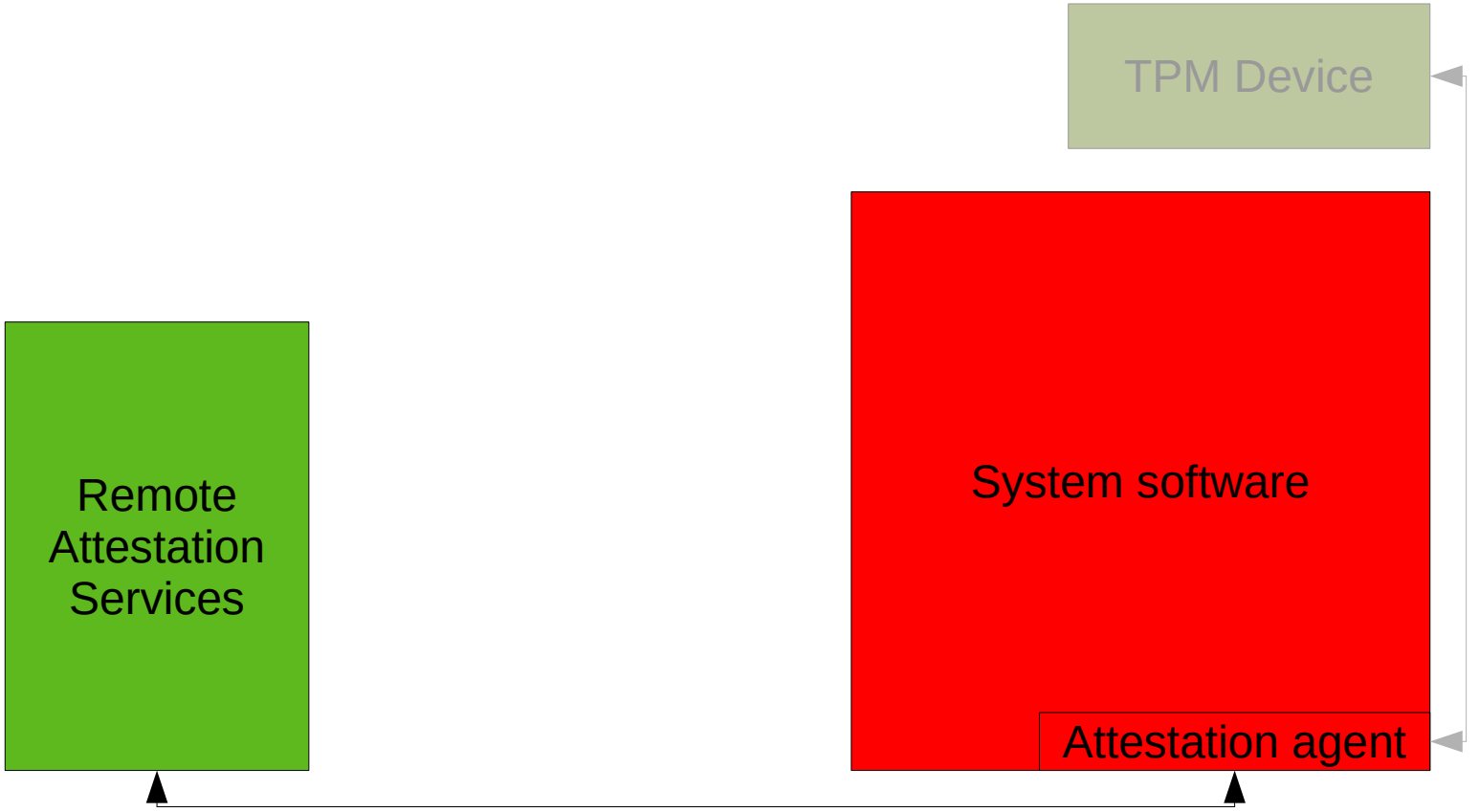
Ephemeral = Stateless

- Smaller code base
- No “secure communication” for state injection headaches
- But:
  - Less features

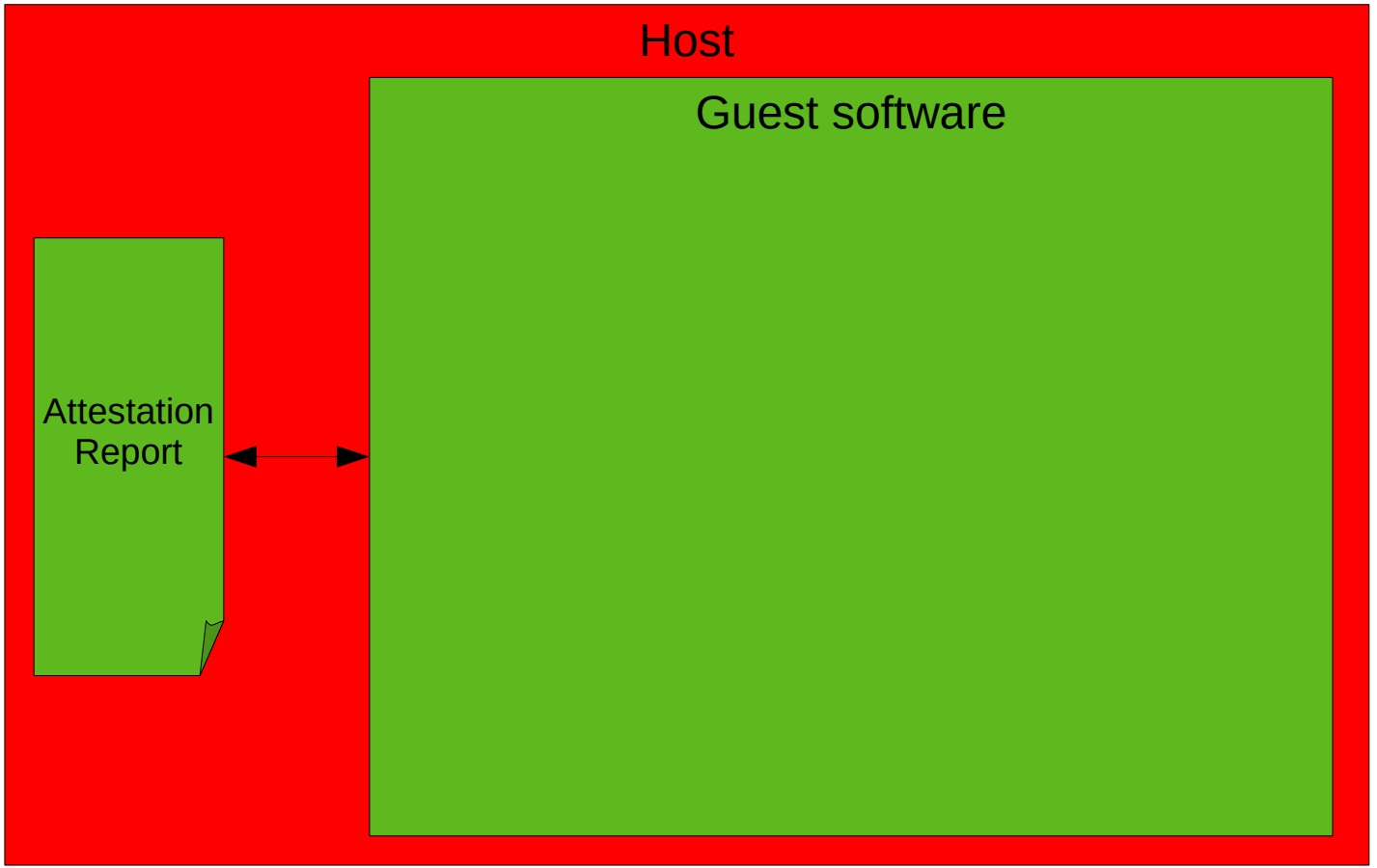
Remote attestation trust model



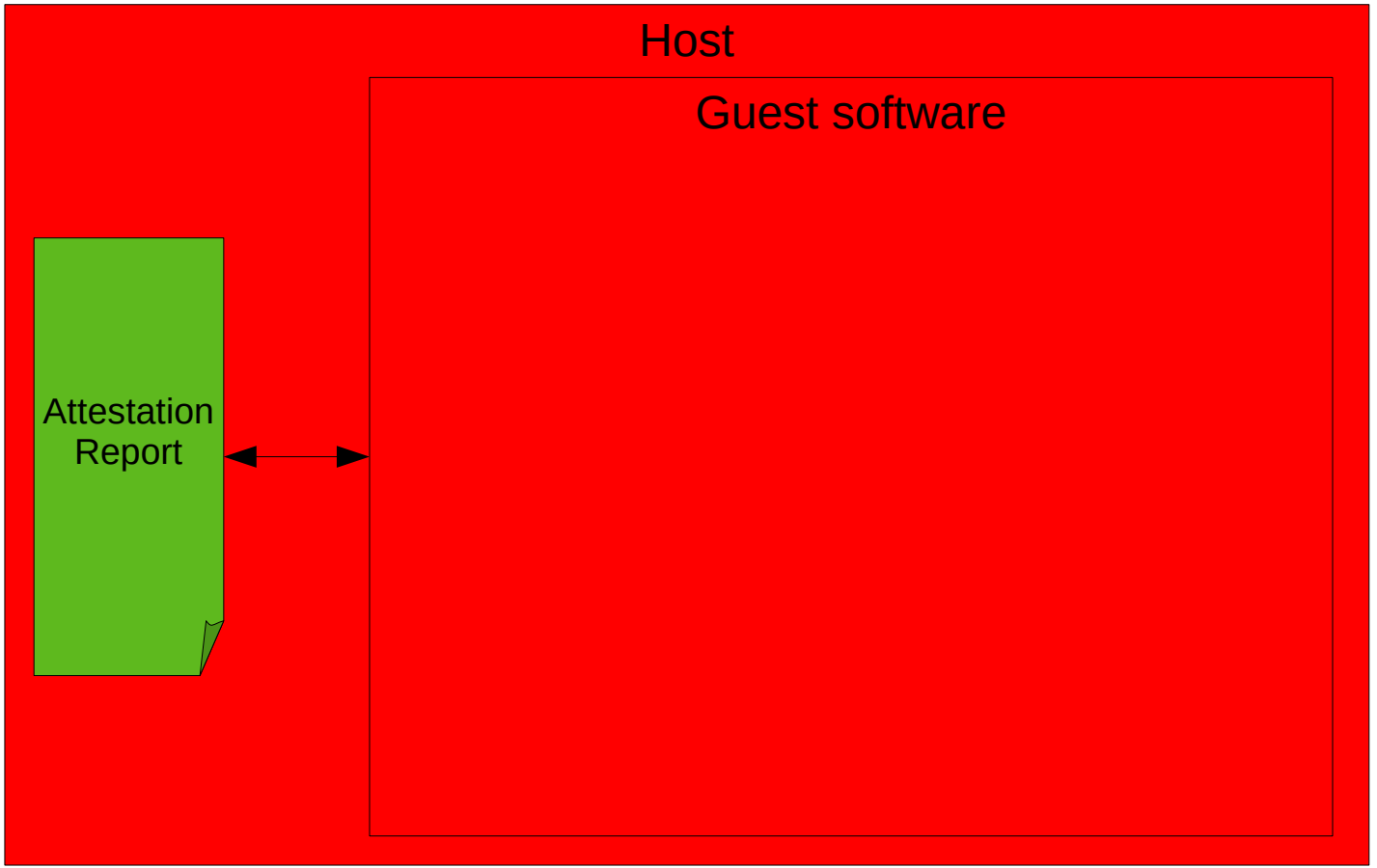
Remote attestation trust model



Confidential VMs trust model



Combined trust models



Where do we run the vTPM?

- Not in the guest (breaks attestation trust-model)
- Not in the host (breaks CVM trust-model)

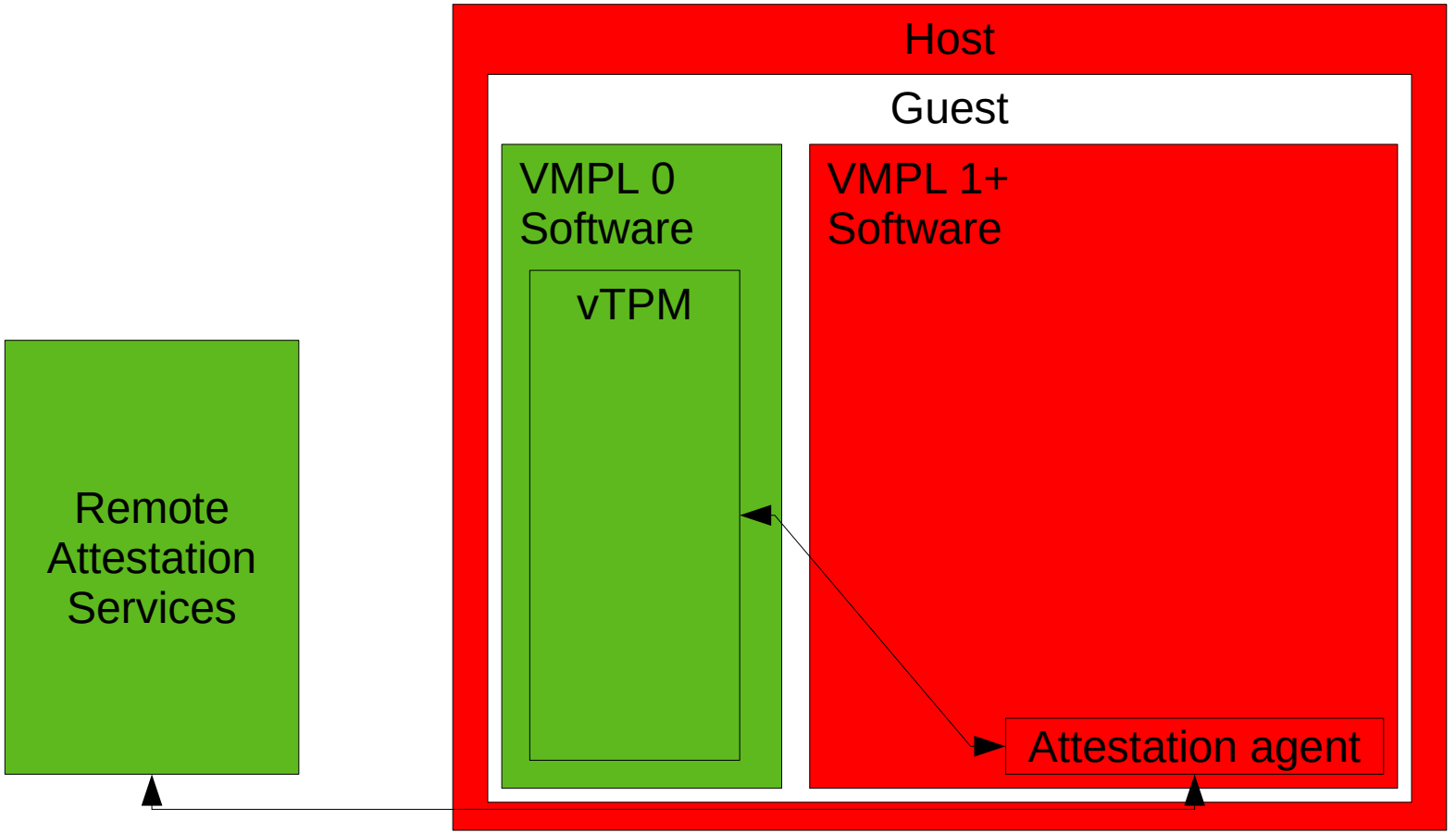
E.g. : AMD’s VMPLs

- Each CVM “enclave” has different privilege levels, 0 to 5 (lower number = higher priority)
- A Higher Priority level has full control of the access from lower levels

Where do we run the vTPM?

- Inside the enclave created for the CVM (secure from the CSP)
- At a higher privilege level than our software stack (secure from the user)

vTPM trust model



How do we trust the vTPM?

- CVM’s Attestation report has boot measurements (that include vTPM)
  - Put it inside the vTPM
  - Get VMPL0 attestation report from vTPM
  - Remote Attestation Service checks digest and signature of Attestation Report

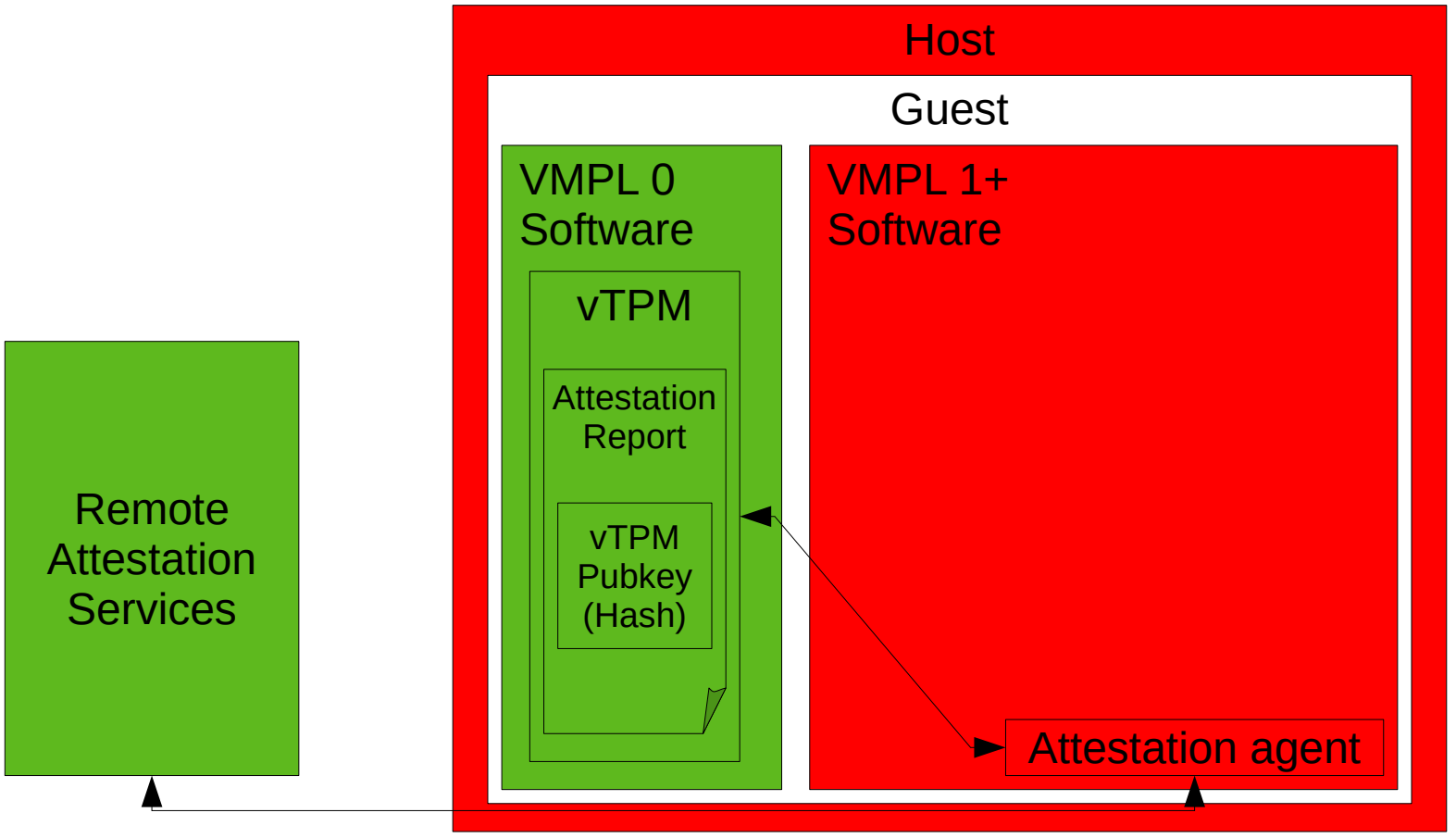
How do we trust the vTPM?

- Standard attestation quotes are signed by a permanent key, but vTPM is stateless, all memory volatile
- CVM’s attestation report measurement don’t include information about keys, i.e. two vTPMs with a different key have the same digest

How do we trust the vTPM?

- Attestation report can include a nonce
- Have the public key as the nonce in the attestation report
- Key pair trust is linked to the attestation report

vTPM trust model



What else?

- Linux-SVSM vs COCONUT and unified solutions
- The Open Source requirement

References

- SVSM-vTPM Proof-of-Concept
- COCONUT Project