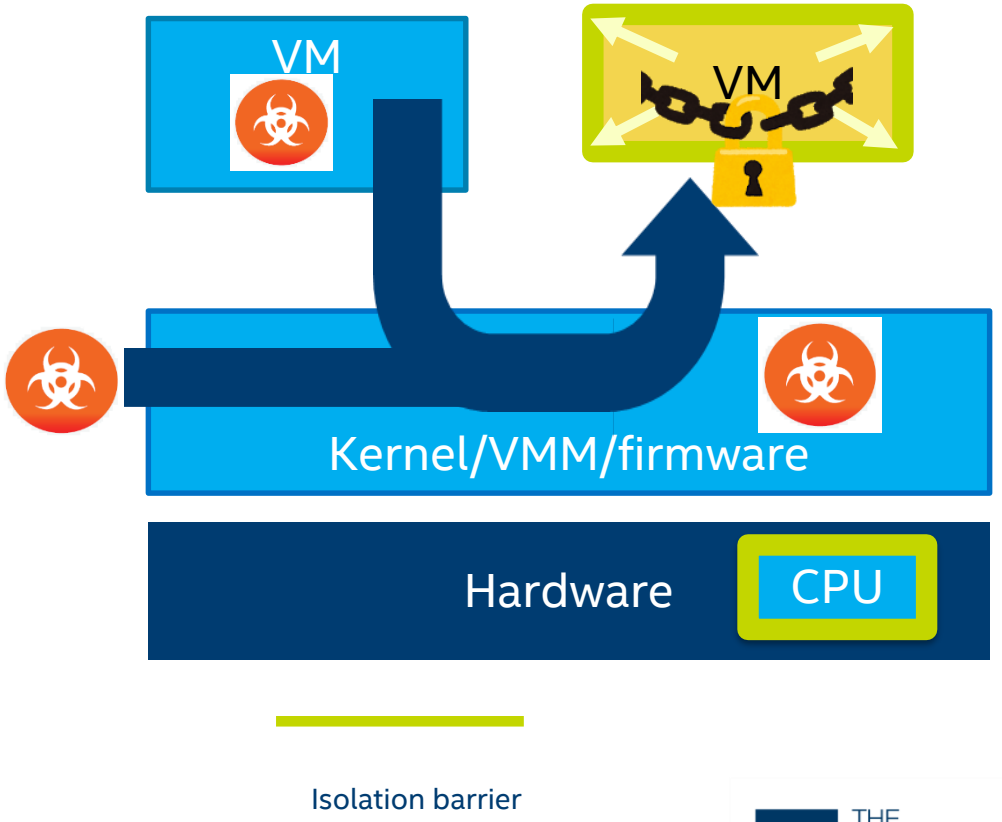
Guest Memory Protection

[redacted]

Isaku Yamahata(Intel)

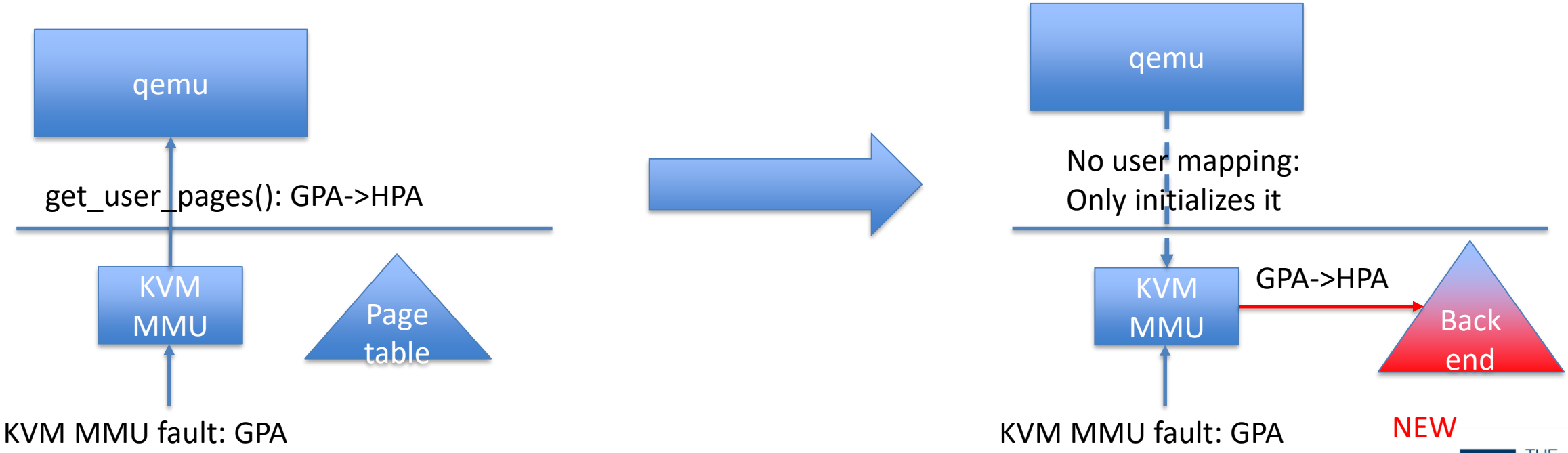
Demand for guest-memory-protection

- Cloud computing is common
- Protecting data in such environment
- Even from VMM/host OS/firmware
- Guest-memory-protection
  - Vendor neutral terminology in qemu/kvm world



Removing user space mapping

- GPA->HPA uses get_user_pages() (or its variant)

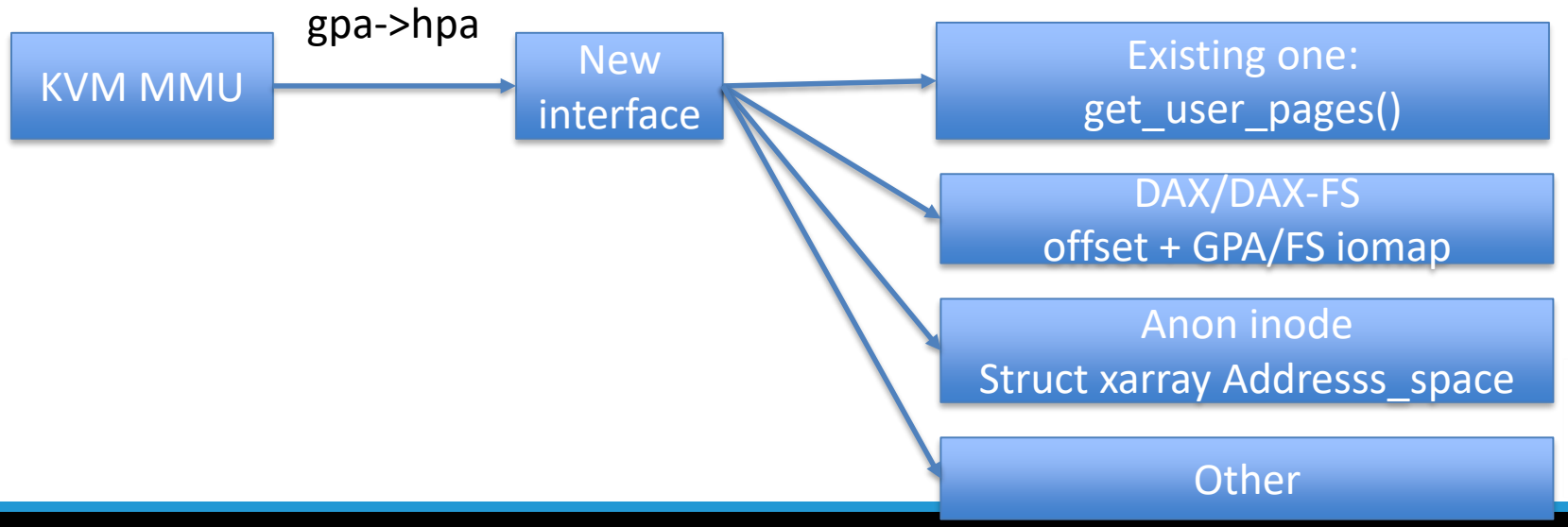


Operations of KVM MMU(x86)

- A new interface for KVM MMU
- Address conversion to resolve KVM MMU fault
  - GPA -> HPA
- Dirty page logging for live-migration
- User fault for postcopy: propagating KVM MMU fault into user space

Allowing multiple backends

- New interface for KVM MMU: GPA -> HPA
  - For various backend
- Update KVM MMU to use it

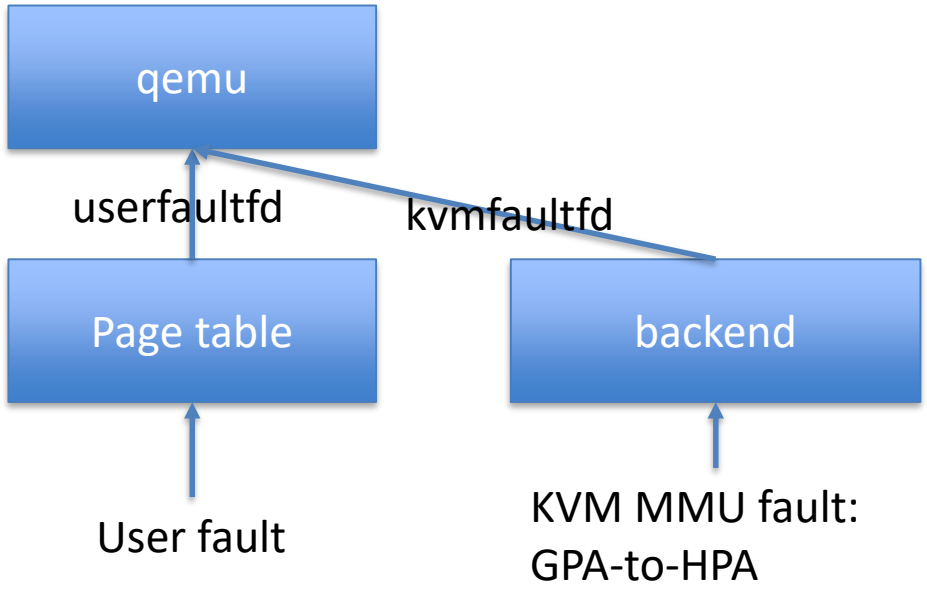


Dirty page logging

- modify mark_page_dirty()
- Instead of marking pte, maintain inside the backend

Postcopy

- Introduce new fd for postcopy
- Mostly same interface to userfaultfd for minimum modification to qemu



Allowing multiple type of VM

- Co-existence of Guest-memory-protected VM and normal VM
- Enhance capability ioctl for VM feature
  - (Some of) KVM capability becomes per-VM, not systemwide
- Enhance Switching device KVM ioctl to VM KVM ioctl for VM feature

More hooks for initialization/teardown

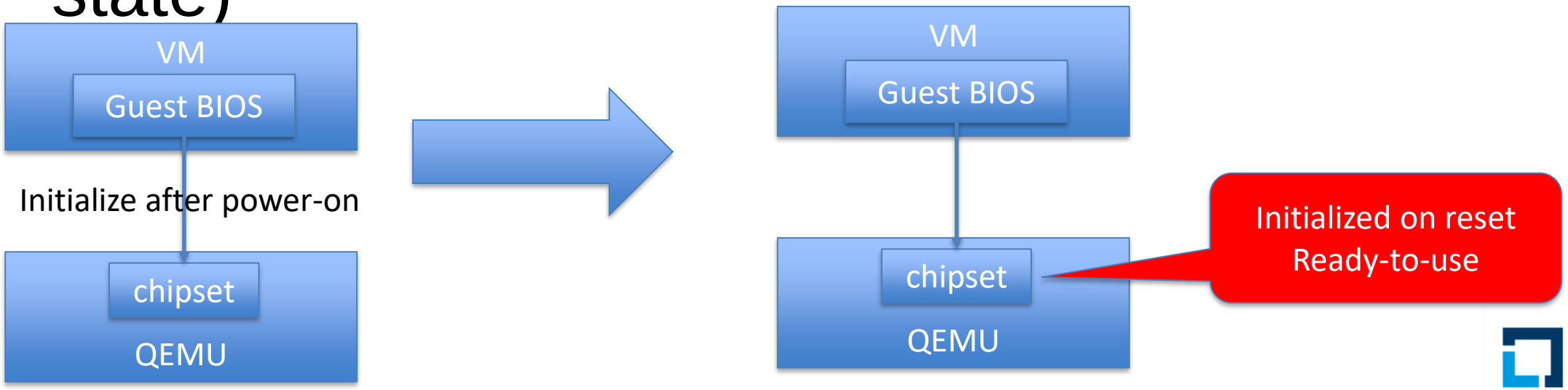
- Introduce VM-type for x86
- Some functionalities aren't useable/must be enabled for memory-protected guest.
  - Hooks to disable/enable/check it
- CPU/CPUID/MSR/memory
  - More hooks for them

Disabling devices/features

- Some of devices/features aren't usable(doesn't make sense) for memory-protected guest
  - More knobs to disable
- Twist of ACPI-table to not-report those devices
- Really disable them in device-emulation

Reducing attack surface

- Eliminating BIOS device initialization
- Disabling initialization only IO(portio/mmio)(freezing (some of)device state)



Proposal to make progress

- Hooks for CPU/memory initialization/teardown
  - More knobs for cpuid/MSR
- More knobs to disable device/registers if appropriate
  - Hook for ACPI table generation
- Add after-reset hook to twist the reset status
- Allowed-list of port-IO/MMIO region which configurable on startup
  - Instead of ad-hoc “if (enabled)” check


KVM
FORUM

Removing qemu mapping to guest memory

- New internal structure: GPA -> HPA
  - Wrapper of: get_user_pages()
  - Struct file: address_space
- Update x86 kvm MMU code to use new interface
- Dirty page logging
- Postcopy
  - Userfaultfd isn’t directly usable.
  - Adds new fd with (mostly) same interface for

Reducing attack surface(cont)

- Don’t allow chipset configuration
- Qemu setup configuration and guest uses it
  - No bios setup because bios is in guest
    - E.g. Don’t allow to change MMCFG
  - Twisting reset state
- Right now, it’s adhoc “if” clauses.
- Disabling unused devices
- BIOS: no pflash
- Additional MSR constraint
  - Currently very adhoc

- Twisting reset state and freeze IO
  - No changes to chipset configuration
- Disabling some devices
  - E.g. legacy device(ISA devices), legacy interrupt controller, SMI
  - Add more device configurations to disable
- Twisting ACPI
  - for disabled devices
  - IRQ table(only MSI)

Disabling devices

- Some devices(especially legacy one) should be disabled/eliminated
- There are sever