Use cases	1. Shared guest faults	2. Private guest faults	3. Don't check for mismatches within KVM?	4. Validate shareability in kvm_gmem_get_pfn?	5. Validate shareability on userspace fault?	6. Need to track shareability?
Legacy case: guest_memfd only used for private pages, other memory type (e.g. shmem) used for shared pages	Via userspace_addr	Via kvm_gmem_get_pfn	Must check (guest_memfd won't have shareability to check against)	No need, since guest_memfd is only used for private memory.	No need, since guest_memfd is not used for shared memory.	No
pKVM	Via kvm_gmem_get_pfn	Via kvm_gmem_get_pfn	Don't check	Can't, no concept of private or shared faults	== ALL	Yes
Option 1 for using guest_memfd for coco VMs: for both private and shared pages - guest always faults via kvm_gmem_get_pfn	Via kvm_gmem_get_pfn	Via kvm_gmem_get_pfn	Don't check	`== ALL` or `== GUEST` depending on `fault->is_private`	== ALL	Yes
Option 2 for using guest_memfd for coco VMs: for both private and shared pages - guest services private faults via kvm_gmem_get_pfn and shared faults via userspace_addr	Via userspace_addr	Via kvm_gmem_get_pfn	Don't check	`== GUEST` IOW, validate against `fault->is_private` (validation should always pass since in this setup where `ALWAYS_FAULT_FROM_GMEM` is `false`, only private faults go to `kvm_gmem_get_pfn`)	== ALL	Yes
gmem used for non-coco VMs, remove from direct map, enable mmap (?) **IOW,** **"all shared mode", but guest always fault in using get_pfn** **(Patrick)**	Via kvm_gmem_get_pfn	Via kvm_gmem_get_pfn	Don't check	No need, since there is no concept of shared vs private in non-coco VMs	No need, since there is no concept of shared vs private in non-coco VMs	No
gmem used for non-coco VMs,	Via userspace_addr	Via userspace_addr	Don't check	No need, since there is no concept	No need, since there is no concept	No

remove from direct map, enable mmap **(Nikita)**				of shared vs private in non-coco VMs	of shared vs private in non-coco VMs	
Proposal: flags	Memslot flag: `ALWAYS_FAULT_FROM_GMEM` (default `false`)		Memslot flag: `DONT_CHECK_MISMATCHES` (default `false`, IOW, will check by default)	No flag. Validate if TRACK_SHAREABILITY is set. For pKVM, call another guest_memfd function like `kvm_gmem_get_pfn_no_validate()`	guest_memfd flag: `TRACK_SHAREABILITY` If this flag is set, guest_memfd will ● Initialize the shareability xarray ● Perform conversions	

How do I read this table?

- "Use cases" column lists how guest_memfd might be used by different users
- Using the legacy case as an example, guest_memfd did not support in-place sharing/mmap before, so previously, guest_memfd was only used for private memory, and some other memory type had to be used for shared memory
  - Column 1: Hence shared guest faults would use memory via `slot->userspace_addr`
  - Column 2: Private guest faults go via `kvm_gmem_get_pfn()`
  - Column 3: "Mismatches" means when the fault is private, but `kvm->mem_attr_array` says the page is shared. In this case, KVM exits to userspace for userspace VMM to handle the mismatch. This is generally used for implicit conversions where the guest doesn't request a conversion before accessing the page.
    - Proposal here is to skip checking within KVM, since if guest_memfd is validating shareability, if `fault->private` doesn't match shareability status, KVM will get `-EACCES` from guest_memfd and will exit to userspace with `KVM_EXIT_MEMORY_FAULT` anyway.
    - In the legacy case, KVM must continue to check for mismatches and exit to userspace
  - Column 4: Proposal here is to validate shareability by changing `kvm_gmem_get_pfn()` to pass in `fault->private`. guest_memfd will check that
    - If `fault->private`, shareability must be `GUEST`, if `!fault->private`, shareability must be `ALL`
    - In the legacy case, there's no need to validate shareability since guest_memfd is only used for private memory
  - Column 5: Whether to validate that shareability is `ALL` on userspace fault (in `kvm_gmem_fault()`)

Summary of proposed flags

- `ALWAYS_FAULT_FROM_GMEM`: guest faults are always serviced via `kvm_gmem_get_pfn()`
  - Default: false, i.e. shared guest faults will be serviced via `slot->userspace_addr` and private guest faults will use `kvm_gmem_get_pfn()`
  - If this flag is set, `userspace_addr` is assumed to be set up such that whether `slot->userspace_addr` or `kvm_gmem_get_pfn()` is used, the same physical page is used to service the fault.
- `DONT_CHECK_MISMATCHES`: KVM will not exit to userspace for userspace VMM to handle the mismatch and will instead defer to validation (if any) in guest_memfd to determine whether to exit to userspace with `KVM_EXIT_MEMORY_FAULT`
- `TRACK_SHAREABILITY`: KVM will initialize shareability xarray and perform conversions