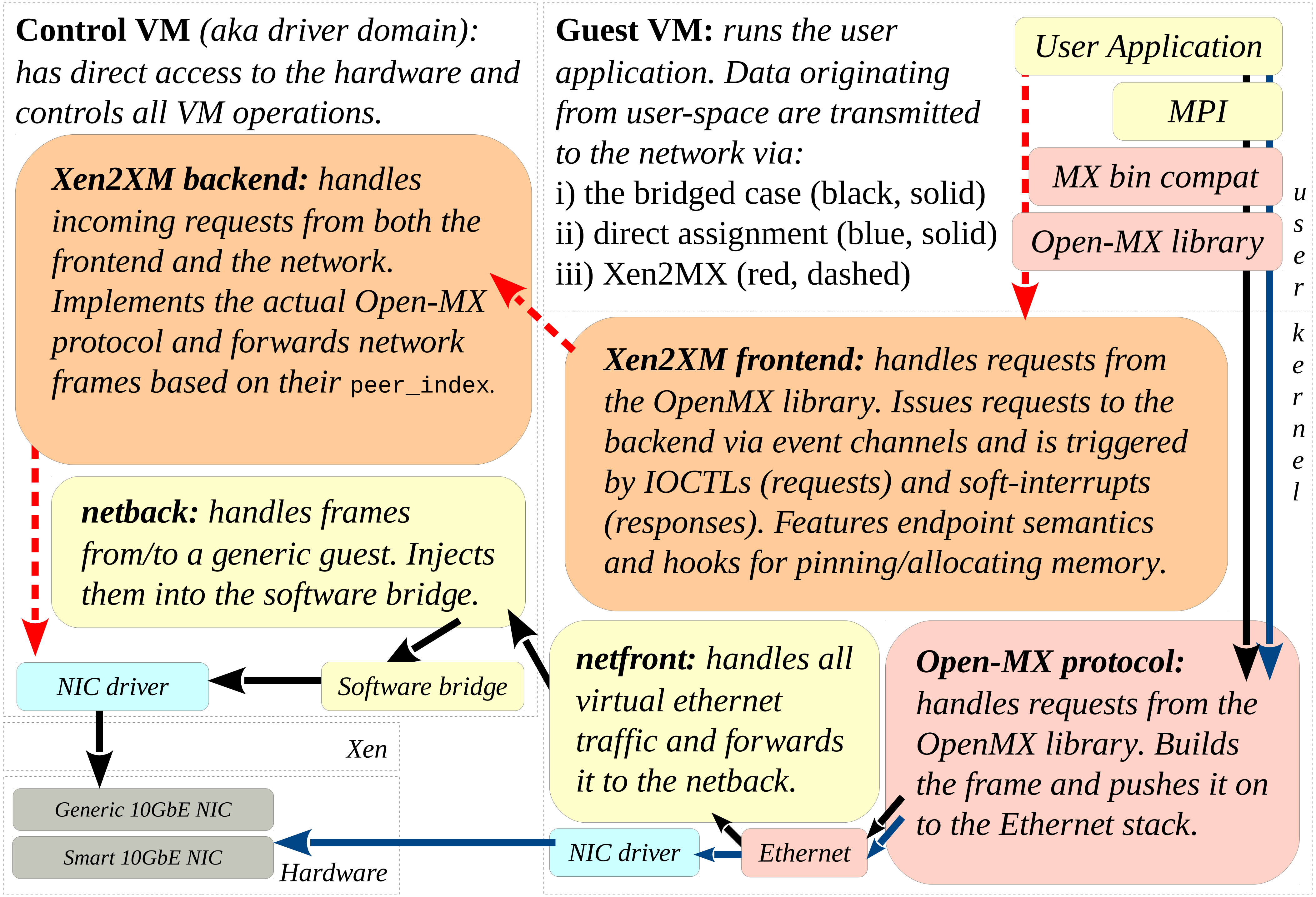
Control VM *(aka driver domain): has direct access to the hardware and controls all VM operations.*
Xen2XM backend: *handles incoming requests from both the frontend and the network. Implements the actual Open-MX protocol and forwards network frames based on their* `peer_index`.
netback: *handles frames from/to a generic guest. Injects them into the software bridge.*
NIC driver
Software bridge
Xen
Generic 10GbE NIC
Smart 10GbE NIC
Hardware
Guest VM: *runs the user application. Data originating from user-space are transmitted to the network via:*
i) the bridged case (black, solid)
ii) direct assignment (blue, solid)
iii) Xen2MX (red, dashed)
User Application
MPI
MX bin compat
Open-MX library
user
kernel
Xen2XM frontend: *handles requests from the OpenMX library. Issues requests to the backend via event channels and is triggered by IOCTLs (requests) and soft-interrupts (responses). Features endpoint semantics and hooks for pinning/allocating memory.*
netfront: *handles all virtual ethernet traffic and forwards it to the netback.*
Open-MX protocol: *handles requests from the OpenMX library. Builds the frame and pushes it on to the Ethernet stack.*
NIC driver
Ethernet