

enable efficient and secure communication between domains. Thanks to device channels, Xen's virtual devices achieve near-native performance. Their performance rests on two design principles: copyless transfer and asynchronous notification.

Look at Figure 7-7. This diagram shows how a split device is used. The guest provides the frontend driver with a page of memory, either containing data to be written or to hold data that is read in (1). The frontend driver places a request in the next available slot in the shared ring-buffer, which contains a reference to the provided page (2), and tells the hypervisor to notify the driver domain that a request is pending (3). The backend wakes up and maps the provided page into its address space (4) so that the hardware can interact with it using DMA (5). Finally, the backend notifies the frontend that the request has completed (6), and the frontend notifies the guest application (7).

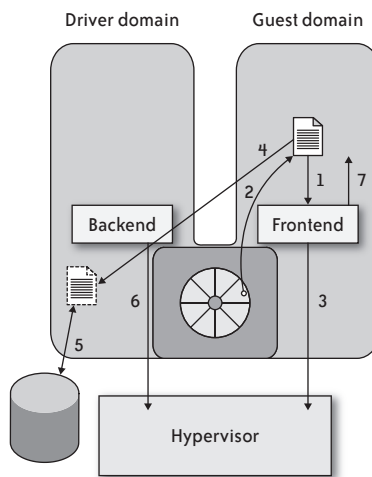


FIGURE 7-7. Anatomy of a split device

Copying data using the CPU is expensive, which is why techniques such as *Direct Memory Access* (DMA) have been developed to transfer device data directly to and from memory without CPU involvement. However, when the data has to move between address spaces, Xen must take special measures to avoid the copy. Xen supports a shared memory mechanism called *grant tables*, whereby each virtual machine maintains a table that defines which of its pages can be accessed by other virtual machines. An index in this table is called a *grant reference*, which, when given to another virtual machine, acts as a *capability*. The hypervisor ensures that only the intended recipient can map the grant reference, which in turn maintains memory isolation. The device channel itself is used to send grant references, which are then used to map buffers for sending or receiving data.

When a new request or response is made, the sender must notify the receiver. This would traditionally use a synchronous notification—akin to a function call—whereby the sender