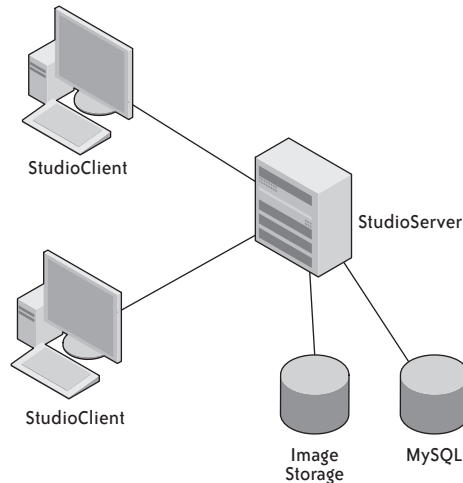


So any workstation in the studio must be interchangeable, but “interchangeable” presents some problems. The images for a single session can consume close to a gigabyte.

We briefly contemplated building the workstations as a peer-to-peer network with distributed replication. Ultimately, we opted for a more traditional client-server model, as shown in Figure 4-5.



*FIGURE 4-5. Studio deployment*

The server is equipped with larger disks than the clients, and they are RAIDed for resilience. The server runs a MySQL database to hold structured data about customers, sessions, and orders. Most of the space, however, is devoted to storing the customers’ photographs.

Because the studios are remote and the associates are not technically adept, we knew it would be important to make the “plumbing” invisible. Associates should never have to look at filesystems, investigate failures, or restart jobs. They should certainly never log into the database server! At worst, if a network cable should be bumped loose, once it is plugged back in, everything should work as normal and also should automatically recover from that temporary problem.

With that end in mind, we approached the system and application architecture.

### **Image repositories**

To make the workstations interchangeable, the most essential feature would be automatic transfer of images, both from the workstation where the photographer loaded them to the server and from the server to another workstation.

The studio client and studio server both use a central component called an image repository. It deals with all aspects of storing, loading, and recording images, including their metadata. On