Figure 6-3. *Three-tier dog application*

The interface tier contains the `lab.html` file and the `dog_interface.php` program. The business rules tier includes the `dog.php` class and the `get_breeds` class. The `dog_data` class is in the data tier. Any requests from the `dog_interface` program for communications with the business rules tier are handled by the `dog_container` class. Any requests for communications with the data tier (from the business rules tier) are also handled by the `dog_container` class. Access to the data tier is only available from or through the business rules tier. Access to the interface tier only occurs from or through the business rules tier.

The interface tier does not know the location/name of any class or method in the business rules tier. This information is discovered by using the `dog_container`. The business rules tier does not know the location/name of any class or method in the data tier. This information is also discovered by using the `dog_container`. This allows complete independence of each tier, which allows changes to occur in one tier without requiring changes in the other two tiers.

Do It

1. Download all the files for the `dog` application from this chapter. Adjust file names in the `dog_application` XML file to discover how missing files are handled in the application. Does the application handle these problems properly? Adjust the `dog_data` XML file to include badly formed XML data. Run the application. Does it handle these problems properly? Empty the `dog_data` XML file (except for the dogs and dog tags). Run the application. Does it handle this situation properly? For any instance that causes the application to error instead of raising an exception, attempt to adjust the code to anticipate the problem and raise an exception.

219