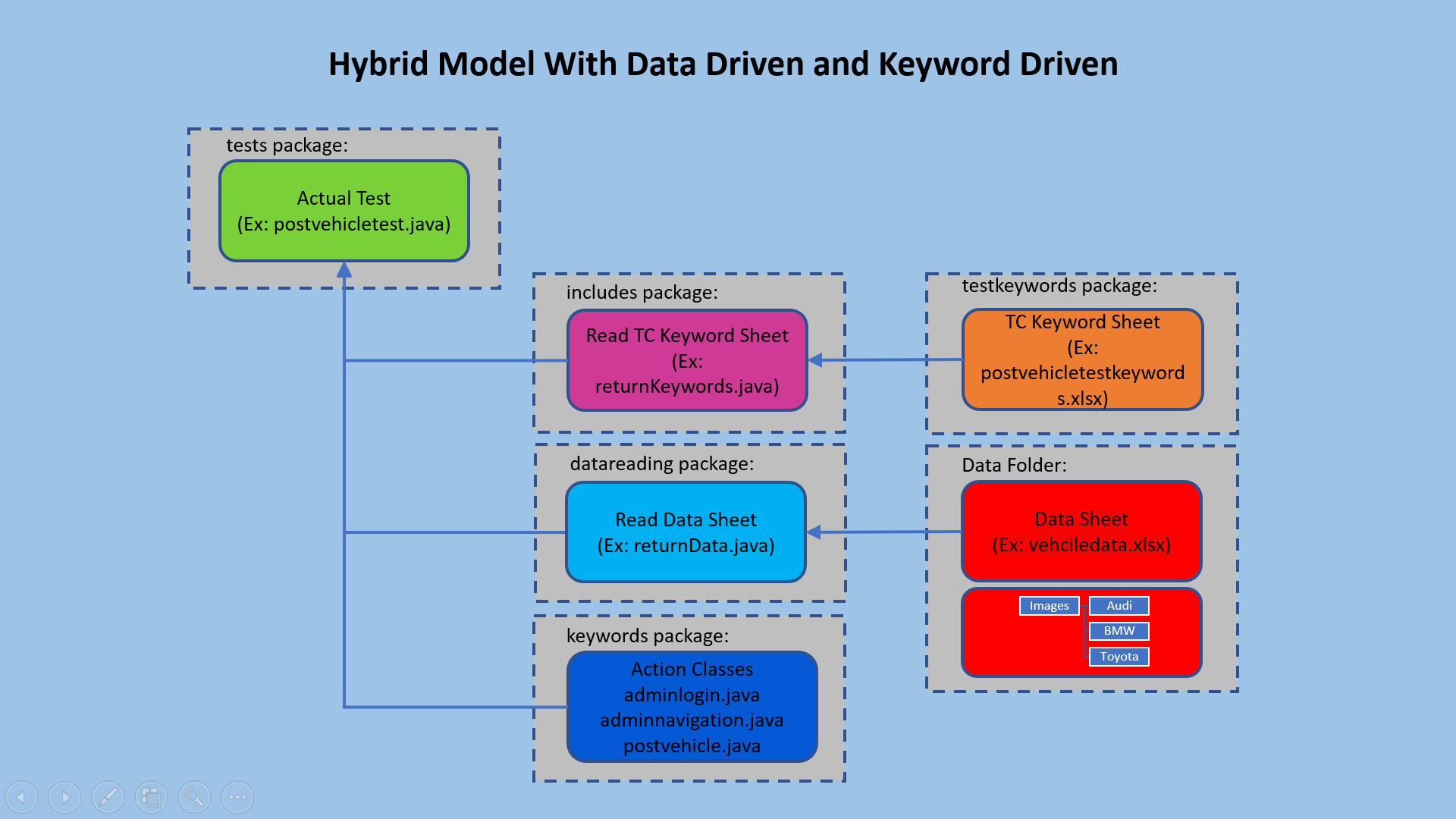
**Hybrid Framework:**

As the name suggests, the Hybrid Testing Framework is a combination of more than one frameworks. The best thing about such a setup is that it leverages the benefits of all kinds of associated frameworks.

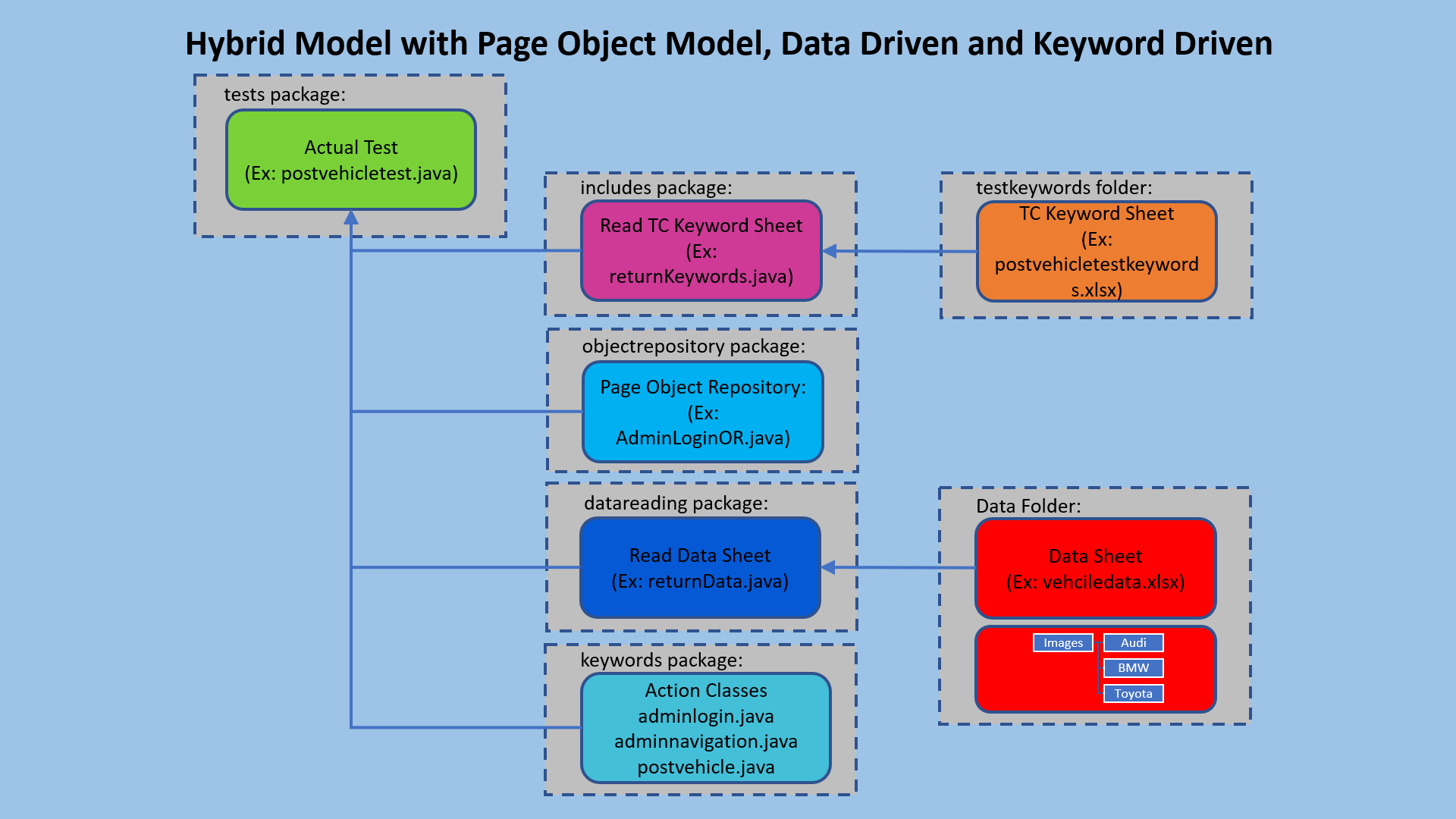
The following diagram shows how the hybrid model works with Keyword driven framework and data driven framework.

For example, in a Keyword driven model, a tester who doesn’t have any automation knowledge can easily add the test scripts to the framework. But if we have to test the same functionality again and again with different sets of data, the tester has to write a lot of steps. So, if we combine this framework with data driven frame work then the test case can be implemented with a smaller number of steps.



In the above diagram you can see that the test data is separated and also the keywords that does each action are written separately. The actual test is written in one case (which is postvehicletest.java) and separate action classes are written for these keywords. When the test case is run it reads the keywords one by one go to the action classes which in turn get the data and does the required action.

We can also separate the objects of each page and put them separately as page object repository (Page Object Model).



If there are any changes to the elements then we can change these directly in the object repository without changing the actual script.

Similarly, if we are using the same functionality again and again we can include modular framework as well. Here since we use keyword driven model, we identify the keywords that belong to a particular functionality and create separate keywords lists for those functionalities and identify these keywords with a particular name so that the actual test steps writing will become much easier. This way the non-automation testers can write the test cases much easily with much smaller number of steps.

