Introduction:

The remote library interface provides means for having test libraries on different machines than where Robot Framework itself is running, and also for implementing libraries using other languages than the natively supported Python and Java. For a test library, user remote libraries look pretty much the same as any other test library, and developing test libraries using the remote library interface is also very close to creating [normal test libraries](http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#creating-test-libraries).

There are two main reasons for using the remote library API:

* It is possible to have actual libraries on different machines than where Robot Framework is running. This allows interesting possibilities for distributed testing.
* Test libraries can be implemented using any language that supports [XML-RPC](http://www.xmlrpc.com/) protocol. There exists ready-made [generic remote servers](https://github.com/robotframework/RemoteInterface#available-remote-servers) for various languages like Python, Java, Ruby, .NET, and so on.

The remote library interface is provided by the Remote library that is one of the [standard libraries](http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#standard-libraries). This library does not have any keywords of its own, but it works as a proxy between the core framework and keywords implemented elsewhere. The Remote library interacts with actual library implementations through remote servers, and the Remote library and servers communicate using a simple [remote protocol](http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#remote-protocol) on top of an XML-RPC channel. The high level architecture of all this is illustrated in the picture below: