If "selenium-server-standalone" jar file has all the library and methods to run a Selenium script, then why should we use "selenium-java" jars? I read else where that it is used for Language Binding. Is that the case? Can someone please confirm?

In Selenium RC (Remote Control), it was mandatory to use selenium-server-standalone.jar jar file because RC involves a server and RC Core will then use this server to establish a communication between the browser and your code. All this is powered by the selenium-server-standalone.jar. along with library functions present in your code.  
  
But with Selenium WebDriver, there is no need for selenium-server-standalone.jar jar file, because the WebDriver API directly communicates with the browser's native language and overcomes the Same Origin Policy issue. So, the selenium-java.jar jar files is used instead for testing with WebDriver.

<http://selenium-release.storage.googleapis.com/2.44/selenium-server-standalone-2.44.0.jar> & <http://selenium-release.storage.googleapis.com/2.44/selenium-java-2.44.0.zip>  
  
I'm aware that the  first version was Selenium RC and the second version is Selenium 2.0(Webdriver). So, my question is, does the latest version include all the JARS required to run even Selenium RC(Selenium Server). And vice versa, does Selenium Server JAR support all the methods run as part of Selenium Webdriver? What are the differences between these two JARS?

Selenium Server: This Was extensively used in Selenium RC for performing local and remote tests because of Same Orgin Policy. However, in Selenium WebDriver, there is no need for Selenium Server JAR for performing local tests. Only for remote tests using Grid, it is needed.  
  
And with the other zip package, we have all the classes & methods as libraries for allowing Selenium WebDriver to achieve automation while testing the web pages.