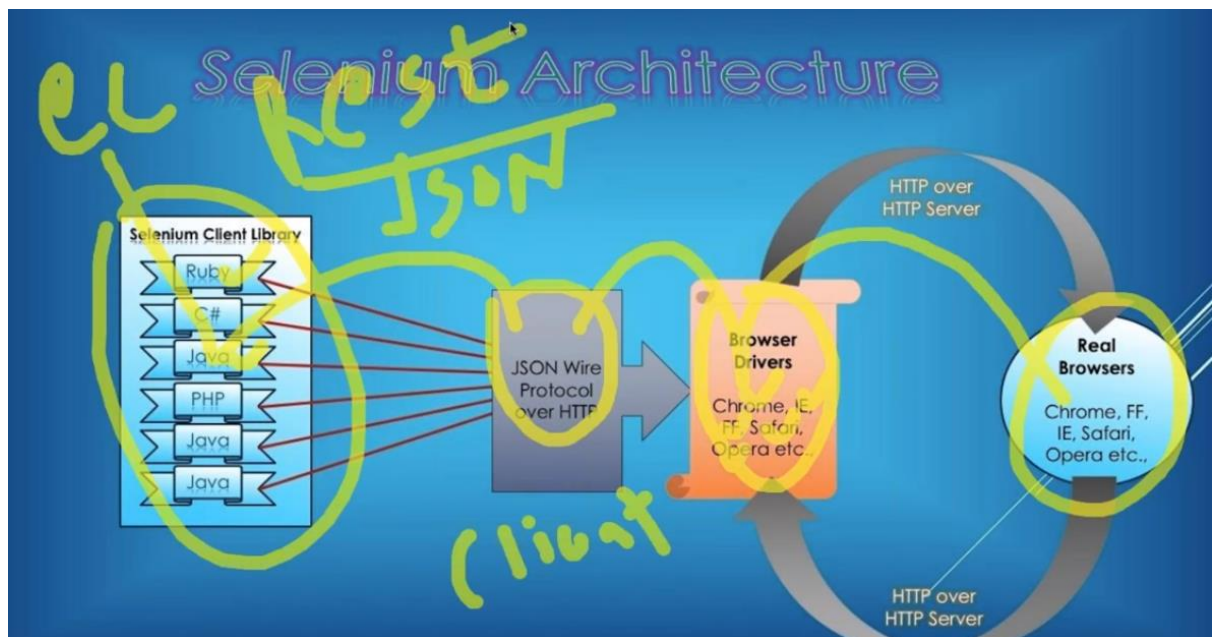


Selenium WebDriver Architecture:

Learn How Selenium WebDriver works internally:

- When the automation script is executed, the following steps happen: +for each Selenium command, a HTTP request is created and sent to the browser driver
- The browser driver uses a HTTP server for getting the HTTP requests the HTTP server determines the steps needed for implementing the Selenium command
- The implementation steps are executed on the browser
- The execution status is sent back to the HTTP server
- The HTTP server sends the status back to the automation script



All implementations of WebDriver that communicate with the browser, or a RemoteWebDriver server shall use a common wire protocol. This wire protocol defines a [RESTful web service](#) using [JSON](#) over HTTP.

The protocol will assume that the WebDriver API has been "flattened", but there is an expectation that client implementations will take a more Object-Oriented approach, as demonstrated in the existing Java API. The wire protocol is implemented in request/response pairs of "commands" and "responses"

POST	/session/:sessionId/frame	Change focus to another frame on the page.
POST	/session/:sessionId/frame/parent	Change focus to the parent content window.
POST	/session/:sessionId/window	Change focus to another window.
DELETE	/session/:sessionId/window	Close the current window.
POST	/session/:sessionId/window/:windowHandle/size	Change the size of the specified window.
GET	/session/:sessionId/window/:windowHandle/size	Get the size of the specified window.
POST	/session/:sessionId/window/:windowHandle/position	Change the position of the specified window.
GET	/session/:sessionId/window/:windowHandle/position	Get the position of the specified window.
POST	/session/:sessionId/window/:windowHandle/maximize	Maximize the specified window if it is not already maximized.
GET	/session/:sessionId/cookie	Retrieve all cookies visible to the current page.
POST	/session/:sessionId/cookie	Set a cookie.