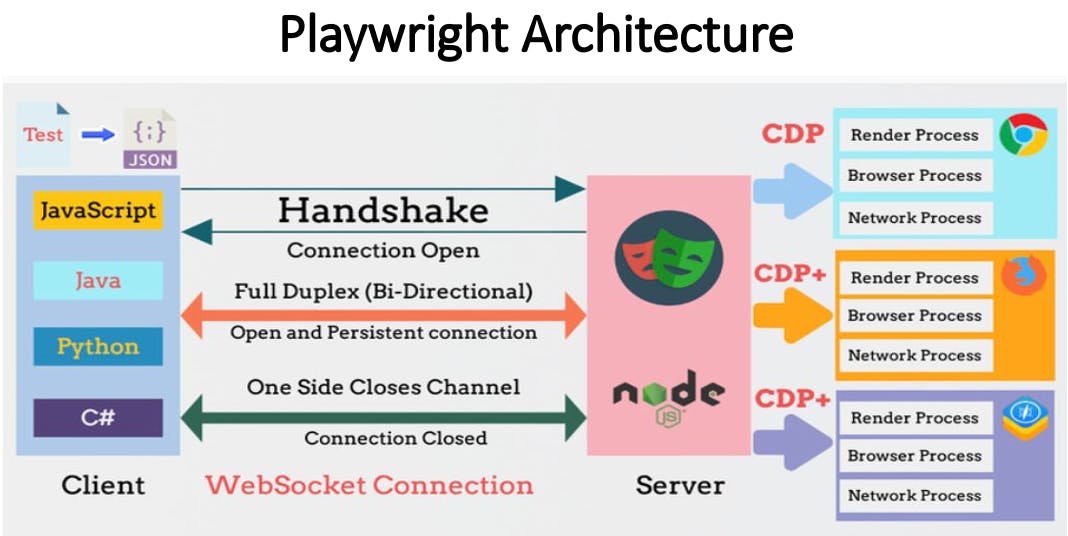
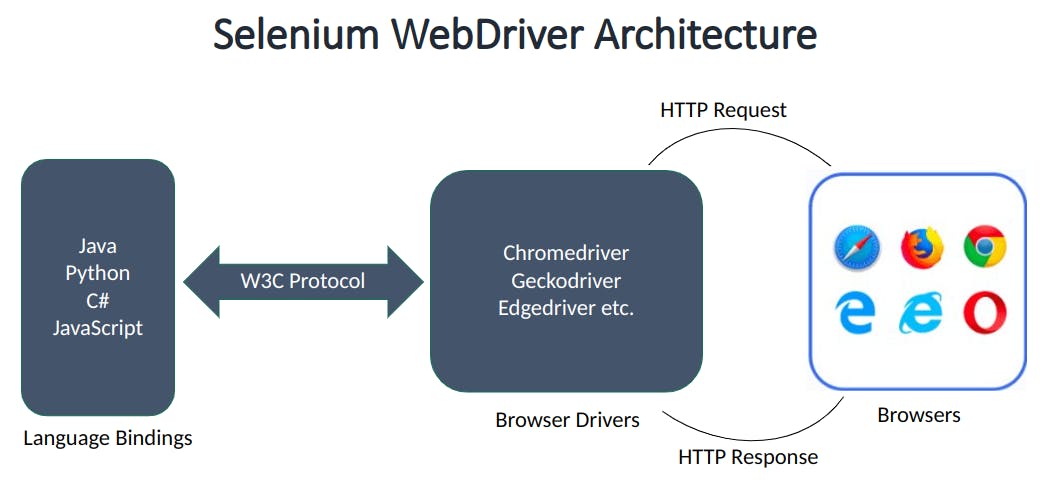
Architecture of Playwright framework

A primary reason for Playwright's speed especially when compared to Selenium is its architecture. Playwright makes use of [WebSockets](https://en.wikipedia.org/wiki/WebSocket) instead of HTTP connection to communicate with end browsers and send/receive requests. The architecture of Playwright would look as follows:





Establishing communication with WebSockets is extremely beneficial as it creates one connection and keeps that connection alive until either party disconnects explicitly. In this case, in the ideal situation, that disconnect will come when all the test cases are executed. The complete process can be documented in steps as follows:

* A WebSocket connection request is sent from the client to the server.
* If the configurational parameters are correct, the connection is accepted.
* The server acknowledges the connection establishment to the client.
* The test execution begins on the same connection.
* All the test cases can use the same connection for their execution.
* Once the execution is completed, the connection is disbanded.

This architecture is in vast contrast to HTTP which is used by Selenium. In HTTP protocol, the server establishes a connection that is accepted and acknowledged by the server. But this connection will last only for that particular request. Once the request completes, the connection breaks and a new connection initiates another request. This will incur a lot of overheads considering that each test execution is going to be a separate request. The more tests in your suite, the more time you can expect. This could be a significant reason we saw a higher execution time for Selenium in the previous section.