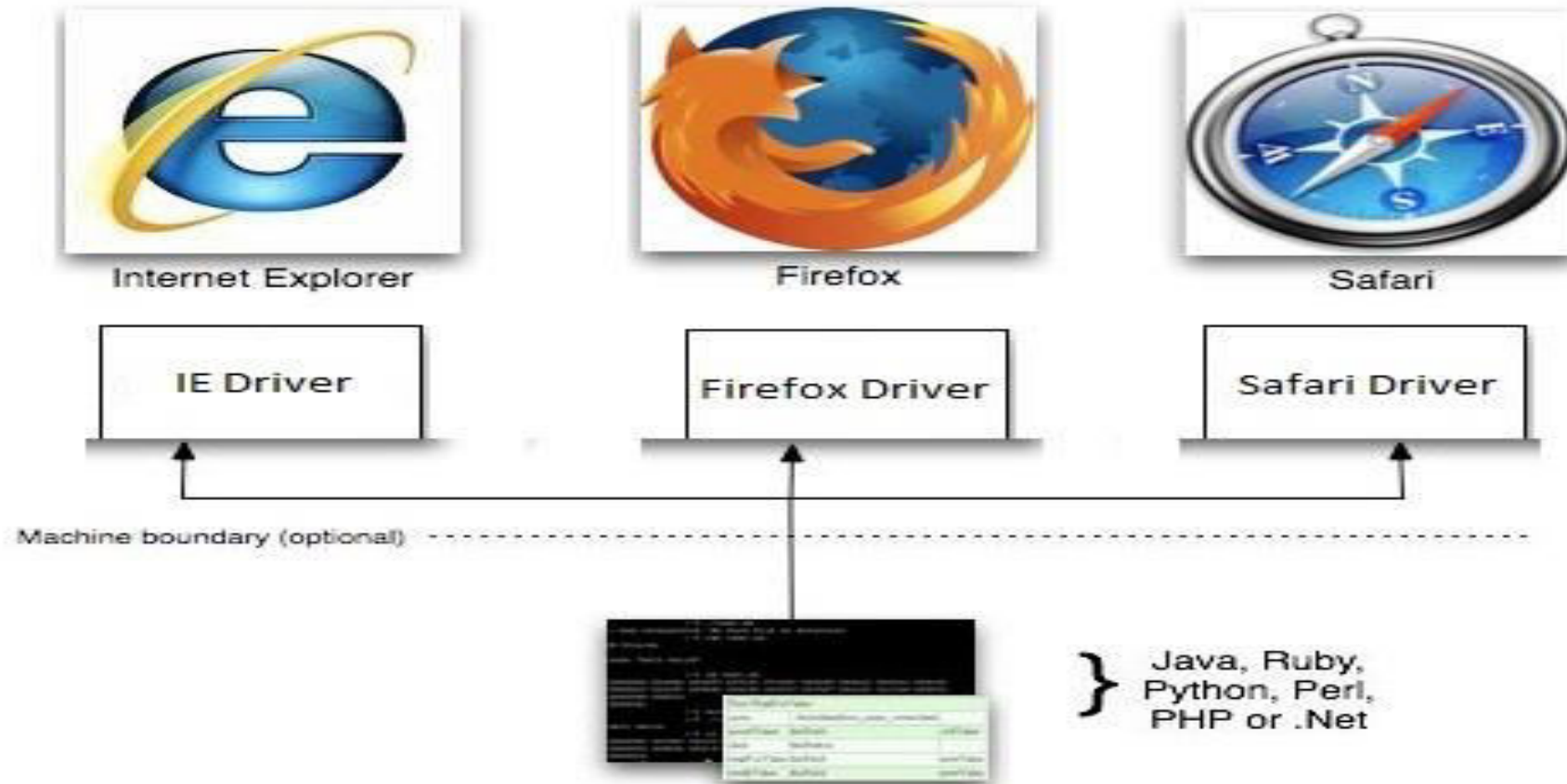# Selenium Webdriver

Architecture of Selenium Webdriver

Setting up Selenium Webdriver

- Selenium can be setup with the help of **"Nuget"**

- "**Nuget**" is package manger for Microsoft Development Platform

- Using "**Nuget**" client, one can consume and release packages

- Right click on the project & select "**Manage Nuget Packages**"

Framework Structure

- *BaseClasses* :- Contain the classes for starting & stopping the Webdriver

- *ComponentHelper* :- Contain the classes specific to the web component

- *Configuration* :- Contain the classes for reading the data from app.config file

- *ExcelReader* :- Contain the classes for reading the data from excel file

Framework Structure

- ***Interfaces***:- Contain the interface for reading the data from file

- ***PageObject*** :- Contain the classes specific Page Object Model

- ***Settings*** :- Contain the classes for constant key

- ***CustomException***:- Contain the classes user define exception

- ***TestScript*** :- Contains the automation script

Configuration File

- We are going to use **"App.config"** file for storing our configuration.

- The configuration should be in the form of "**Key : Value"** pair, where "**Key**" should be unique

- To load the configuration, we will used "**System.Configuration**" which is a in built utility of Visual Studio

- To use it, we need to refer the "**System.Configuration.dll"** file in our project

Configuration File

- The class which help us in reading the configuration from "**App.config**" is "**ConfigurationManager**".

- Inside the **"ConfigurationManager" ,** there is a property **"AppSettings"**

- The **"AppSettings"** has a "**Get**" method, which takes **"Key"** as an argument and return its **"Value"**

- All the configuration inside the "**App.config**" is of type "**string**"

IConfig Interface

- This interface will contain some basic method which are required for our framework such as **"GetBrowser"**.

- All the class which is going to read the data from external file system have to implement this interface.

- The reason of enforcing this restriction, is there are basic methods which are Important for our framework to initialize.