1. **Browser Architecture**

**The browser main functionality is to**

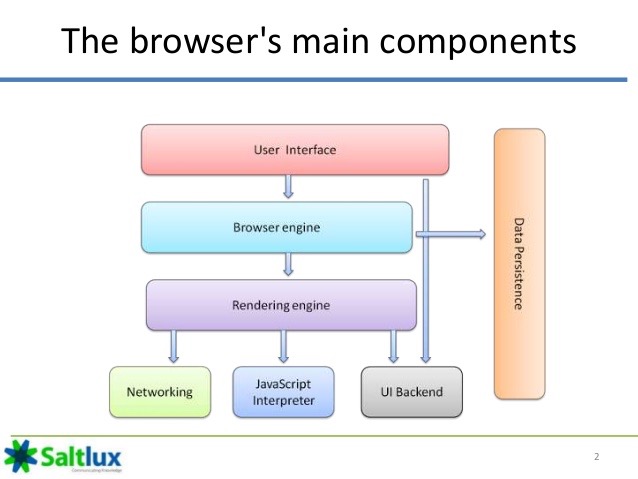
**present the web resource you choose,**

**by requesting it from the server and**

**displaying it on the browser window.**

**The resource format is usually HTML**

**but also PDF, image and more.**

Most basic component that all web browser must exhibit are listed below:

**The user interface**: this includes the address bar, back/forward button, bookmarking menu, etc. Every part of the browser display except the window where you see the requested page.

**The browser engine**: marshals actions between the UI and the rendering engine. Browser Engine provides methods to begin the loading of URL and other high-level browsing actions.

- Reload, Back, Forward actions

- Error messages

- Loading progress

**The rendering engine** : responsible for displaying requested content. For example if the requested content is HTML, the rendering engine parses HTML and CSS, and displays the parsed content on the screen. FROM “STRING” (HTML) TO OBJECT MODEL (DOM)

**Networking**: Access and transfer data on the internet (calls HTTP, HTTPS, FTP). The Networking components handles all aspects of internet communication or security.

**UI backend**: used for drawing basic widgets like combo boxes and windows. This backend exposes a generic interface that is not platform specific. Underneath it uses operating system user interface methods.

**JavaScript interpreter**. Used to parse and execute JavaScript code.

**Data storage**. This is a persistence layer. The browser may need to save all sorts of data locally, such as cookies. Browsers also support storage mechanisms such as localStorage, IndexedDB, WebSQL and FileSystem.

It is important to note that browsers such as Chrome run multiple instances of the rendering engine: one for each tab. Each tab runs in a separate process.

1. **SPA CONCEPT (Single page application)**

A **single-page application (SPA)** is a web application that interacts with the user by dynamically rewriting the current page rather than loading entire new pages from a server. This approach avoids interruption of the user experience between successive pages, making the application behave **more like a desktop application**. In an SPA, either all necessary code – HTML, JavaScript, and CSS – is retrieved with a single page load, or the appropriate resources are dynamically loaded and added to the page as necessary, usually in response to user actions. The **page does not reload at any point in the process**, nor does control transfer to another page, although the location hash or the HTML5 History API can be used to provide the perception and navigability of separate logical pages in the application. Interaction with the single page application often involves dynamic **communication with the web server** behind the scenes.

