Rich Web Application Revision Week 1

# What is the Web?

## **What is the Web**

The informal definition behind this question is the web is the biggest application on the public internet and the private intranet. In one view as the consumer, the web is a portal to a vast amount of knowledge, opinion, services and a way to connect with others. In another as the developer, the web is a suite of technologies for building the interfaces that the users or machines can interact to get access to all the information.

Think about this question. What technologies make up the web and how do we use them.

## **What is a Web Application?**

Loosely, this is described as the client-side machine system provisioned over the network using the standard modern tools. This is used by the end-user to interact with a server-side interface on a remote device, which a browser or a graphical engine is the standard method. There are many different architecture designs used to build the business and presentation logic responsibilities between the client and server machines. This has evolved over time as devices has improved in the systems power combined with the users demand in improved UX.

## **What is a Rich Web Application?**

This is where the Client has the responsibilities to implement the presentation layer logic, which is the user interface and the user experience. This is the latest evolution in a series of iterative designs changes the web applications have undergone since the first websites. Rich Web Systems employ heavy use of JavaScript to offer a lean data transfer interface with the server, such as Bootstrap.

## **When is it a Web Application?**

The main difference between a website and web applications is the interaction with a server machine that makes the application more dynamic. Websites are designed to display information only. Usually the communication involves a client – server using a protocol like HTTP, combined with a certain type of message standard serialised formats like JSON. Once these are met, the last criteria would be an agreed design pattern using suitable APIs with the JavaScript as part of the application

Some of the technologies used in the modern architecture design are HTML5, CSS3, ECMAScript, Web Sockets, XML, JSON and design patterns like REST

# Websites vs Web Applications

## **What is the difference?**

### Websites

The purpose of the website is to convey information, such as a picture document repository or a gallery. These are statically created to being served to the browser server. Usually HTML and CSS handle most of the UI features. JavaScript was used but CSS is taking over some of its tasks more easily. There is little interaction with the server engine.

### Web Applications

This is heavily involved with the server, mainly interaction with Data Processing functions. For example, a message being sent through a social media app platform. The content is usually generated more dynamically from the server. HTML and CSS are currently being used but may become obsolete soon. There is heavier use of JavaScript to interact with the server interfaces.

## **Data Synchronisation**

Because of the complexity of building web application over websites, data synchronisation becomes an issue that needs to be handled. This is due to multiple sources from servers combined with ensuring the views contain the same information for all potential users.

## **User Experience**

The goal of the interface is to be as intuitively and functional as possible for the user. To rely on this, designers often use idiomatic usage of graphical elements as input nodes. The easier and more quality the UI is for the user, the higher level of complexity of the Web Application. This can be completed through the Dom, Google Metrics tools and AB testing

## **Security**

The issue of the user, their identity and identity verification in the application. When a change happens to the server, it must have been authorised by the application combined with the users’ authentication in the system. The client must also guard against the process and be ready for success and failure scenarios.

# Bootstrapping a Web Application

## **How does the Web Application get started**