<u>i18n & L10n:</u>

- Internationalization (i18n) is the process of designing a software application so that it can be adapted to various languages and regions without engineering changes.
- I.e. The purpose is to make an application language agnostic.
- Localization (L10n) is the process of adapting internationalized software for a specific region or language by adding locale specific components and translating text.
- I.e. The purpose is to adapt an application for a specific language (locale).
- Internationalization is the process of designing and developing your software so it can be adapted and localized to different cultures, regions, and languages while localization is the adaptation of your software to meet the language, culture, and other requirements of each locale.
- Internationalization helps you build your software t with future markets and languages in mind. It's the process of neutralizing the code, content, and design so that, down the road, it'll be easier to adapt your product to additional cultures without having to completely re-engineer it. Localization typically follows internationalization.
- Note: Localization is not only about language translation.

We also have to deal with the following:

- Number format
- Date/Time
- Punctuation
- Sort orders
- Units and conversion
- Currency
- Paper size
- Page layout
- You can configure your locale preference in your browser.
- The language that the website is in can be found in the request header.

E.g.

```
▼ Request Headers view source
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
```

- Some ways to make your websites language agnostic:
 - Store the language preference in the URL.
 - Store the language preference in the user's profile.
 - Store the language preference in a cookie.

Web Services:

- Is the predecessor to web APIs.
- A **web service** is a standardized medium to propagate communication between the client and server applications on the internet. Web services are mostly used between web servers (B2B).
- A client would invoke a series of web service calls via requests to a server which would host the actual web service. These requests are made through what is known as Remote Procedure Calls (RPC). RPCs are calls made to methods which are hosted by the relevant web service. The requests are typically made through HTTP but can also be made through SMTP. The main component of a web service design is the data which is transferred between the client and the server using XML. This provides a common platform for applications developed in various programming languages to talk to each

other. Web services use SOAP for sending the XML data between applications. This data is called a SOAP message, which is an XML document.

 SOAP (Simple Object Access Protocol) is known as a transport-independent messaging protocol. SOAP is based on transferring XML data called SOAP Messages. Each message contains an XML document. Only the structure of the XML document follows a specific pattern, not the content.

Here is what a SOAP message consists of:

- Each SOAP document needs to have a root element known as the <Envelope> element. The root element is the first element in an XML document.
- The "envelope" is in turn divided into 2 parts. The first is the header, and the next is the body.
- The header contains the routing data which is basically the information which tells the XML document to which client it needs to be sent to.
- The body will contain the actual message.

	request		
x<br <so env soa enc</so 	<pre>kml version="1.0"?> bap:Envelope xmlns:soap="http://www.w3.or yelope" ap:encodingStyle="http://www.w3.org/2001/ coding"></pre>	rg/2001/12/soap- /12/soap-	
<sc < <td><pre>bap:Body> cm:GetPrice xmlns:m="http://www.w3schools</pre></td><td>s.com/prices"></td></sc 	<pre>bap:Body> cm:GetPrice xmlns:m="http://www.w3schools</pre>	s.com/prices">	
<td>soap:Envelope> ex</td> <td>ample from www.w3school.com</td>	soap:Envelope> ex	ample from www.w3school.com	
Ĵ	Soap Request (over HTTP)	Ĵ	
Client	t	Server	
SOAP Response			
SOAP	Response		
SOAP I	Soap Response (over HTTP)		
Cl [*] xmi<br <soap ence soap enco</soap 	<pre>Kesponse Soap Response (over HTTP) I version="1.0"?> p:Envelope xmlns:soap="http://www.w3.or lope" :encodingStyle="http://www.w3.org/2001/ ding"></pre>	g/2001/12/soap- 12/soap-	
SOAP f	<pre>Kesponse Soap Response (over HTTP) I version="1.0"?> p:Envelope xmlns:soap="http://www.w3.or lope" :encodingStyle="http://www.w3.org/2001/ ding"> p:Body> :GetPriceResponse xmlns:m="http://www.w es"> <m:price>1.90</m:price> m:GetPriceResponse xmlns:m="http://www.w es"> 1.90 m:GetPriceResponse xmlns:m="http://www.w es"> 1.90 m:GetPriceResponse> ap:Body> </pre>	g/2001/12/soap- 12/soap- 3schools.com/	

- The **WSDL** (Web services description language) file is an XML-based file which basically tells the client application what the web service does.

A web service cannot be used if it cannot be found. The client invoking the web service should know where the web service actually resides.

Secondly, the client application needs to know what the web service actually does, so that it can invoke the right web service. This is done with the help of the WSDL, known as the Web services description language. By using the WSDL document, the client application would be able to understand where the web service is located and how it can be utilized.

The WSDL file provides a way to describe your web service.



WSDL	How do we talk together? WSDL Client Serve	er
<message na<br=""><p </p </message> <message na<br=""><part nam<="" td=""><th>me="getPriceRequest"> art name="Item" type="xs:string"/> me="getPriceResponse"> e="Price" type="xs:double"/></th><td></td></part></message>	me="getPriceRequest"> art name="Item" type="xs:string"/> me="getPriceResponse"> e="Price" type="xs:double"/>	
 <porttype n<br=""><operatio <input <output </output </input </operatio </porttype>	ame="glossaryPrice"> n name="getPrice"> message="getPriceRequest"/> message="getPriceResponse"/> on>	
<binding ty<br=""><soap:bi http" /> <operati <soap: <input <outpu <th><pre>pe="glossaryTerms" name="b1"> nding style="document" transport="http://schemas.xmlsoap.org/soap/ on> operation soapAction="http://example.com/getPrice"/> ><soap:body use="literal"></soap:body> t><soap:body use="literal"></soap:body> on></pre></th><td></td></outpu </input </soap: </operati </soap:bi </binding>	<pre>pe="glossaryTerms" name="b1"> nding style="document" transport="http://schemas.xmlsoap.org/soap/ on> operation soapAction="http://example.com/getPrice"/> ><soap:body use="literal"></soap:body> t><soap:body use="literal"></soap:body> on></pre>	

- Universal Definition Language (UDDI) is a standard for describing, publishing, and discovering the web services that are provided by a particular service provider. It provides a specification which helps in hosting the information on web services. It provides a way to advertise your web service.
- While web services are a good idea, they have not been widely adopted because:
 - Web services are very flexible but very complex architecture.
 - Standards for web services evolve faster than development frameworks.
 - Ad-hoc solutions adopted by the main actors of the web.
 I.e. JSON replaced XML as JSON is more human and computer readable.
 Furthermore, web APIs replaced web services.

Advertising, Analytics and Tracking:

- Advertising:
- One of the oldest and most popular ways to generate revenue for a webpage.
- E.g.



What's happening is that an advertiser will go to an agency with some ads and ask the agency to find some content publishers to publish their ads on the content publisher's website. The content agency then looks for content publishers to showcase the ads on their website so that visitors of that website can see the ads.

Note: Sometimes, advertisers go directly to the content publisher. Examples of this are Google and Facebook.

- There are 2 main popular models of online advertising:

1. Click Banners:

-

- On other people's websites.
 - Examples include:
 - Pay per click
 - Pay per view
 - Pay per transaction

2. Sponsored Links:

- On the search engine result page.
- Buying keywords (bidding price).
- Google does this

- If you embed ads in your webpage/webapp, the ad network rewards you with cash every time a visitor clicks on an ad on your webpage.
- For the web programmer, a javascript snippet to be inserted in the webpage will perform ajax requests to the ad networking company. The ad is shown in an iframe.
- For the visitor, a third party cookie will track his/her visits through different sites to display more relevant ads.
- Web Scraping and Click Fraud:
- There are 2 main types of advertising fraud:
 - 1. Web Scraping:
 - Web Scraping occurs when a website extracts, collects and aggregates data from other websites to make it seem legit. Then, when users search for certain keywords, search engines will prioritize those websites.

I.e. They are spamming search engines (spamdexing).

The goal is to attract visitors to your website and fool them to click on ads.

- 2. Click Fraud:
- **Click fraud** is having a bot (a computer program) that automatically clicks on:
 - ads displayed on your website to increase your earnings or
 - ads anywhere on the web but targeting specific ads to increase the expenses of your competitors.
- Web Analytics:
- Is used to maximize your revenue from advertisement.
- It looks at the following information:
 - Which website guides the users to your website?
 - What are the keywords that they typed in the search engine that guide them to your website?
 - What do they do on your website?
 - How long do they stay?
 - What pages do they look at?
 - Where are they from geographically?
 - What are their hobbies?
 - How old are they?
- There are 2 techniques for web analytics:

1. Log file analysis:

- Is done on the server side.
- Server side code analyzes the web server logs.
- Can collect IP addresses, rate of requests, etc.

2. Page tagging analysis:

- Is done on the client side.
- Javascript code analyzes the user's interactions.
- Can collect the type of device the user is on (laptop/cell phone/desktop) and the brand of the device (Apple/Samsung/Google)
- You can do web analytics on your own by mixing log analysis and page tagging or you can ask another company, like Google, to do it for you. If you do web analytics as a service, you can do page tagging only.
- Web Tracking:
- **Third-party cookies** are cookies with a unique ID to identify the same user visiting different websites. Google and other websites place third-party cookies in their websites or ads to learn more information about their users.

- **Browser fingerprinting** refers to tracking techniques that websites use to collect information about you. Some information collected include:
 - the User agent header
 - the Accept header
 - the Connection header
 - the Encoding header
 - the Language header
 - the list of plugins
 - the platform
 - the cookies preferences (allowed or not)
 - the Do Not Track preferences
 - the timezone
 - the screen resolution and its color depth
 - the use of local storage
 - the use of session storage
 - the presence of AdBlock
 - the list of fonts
 - Browsing in **privacy mode** disables the browser data storage.
- I.e. It disables the following:
 - (frontend) web cache
 - HTTP cookies
 - HTML5 local storage
 - Flash/Silverlight cookies

However, it does not protect against browser extensions.

 Do Not Track is a HTTP header field that was proposed in 2009. If you have this on, the browser will tell the website to not to use third party cookies. This is completely useless because websites can decide whether or not to honor such a request.

I.e. The browser asks the website to not use third party cookies, but it is up to the website to decide if they want to comply or not.