Micro Frontends Integration

In this lesson, we will discuss how micro frontends are integrated with each other to make the complete UI of a website

We'll cover the following

- Client-Side Integration Of Micro Frontends
- Server-Side Integration
- Technology & Frameworks Facilitating Server Side Integration

So, once we have different micro frontends ready for our online game store, we need to integrate them together to have a functional website. There are two ways we can do this -

1. By integrating micro frontends on the client

2. By integrating micro frontends on the server

This concept is pretty similar to the *client side & the server-side rendering*, that we've discussed earlier in the course.

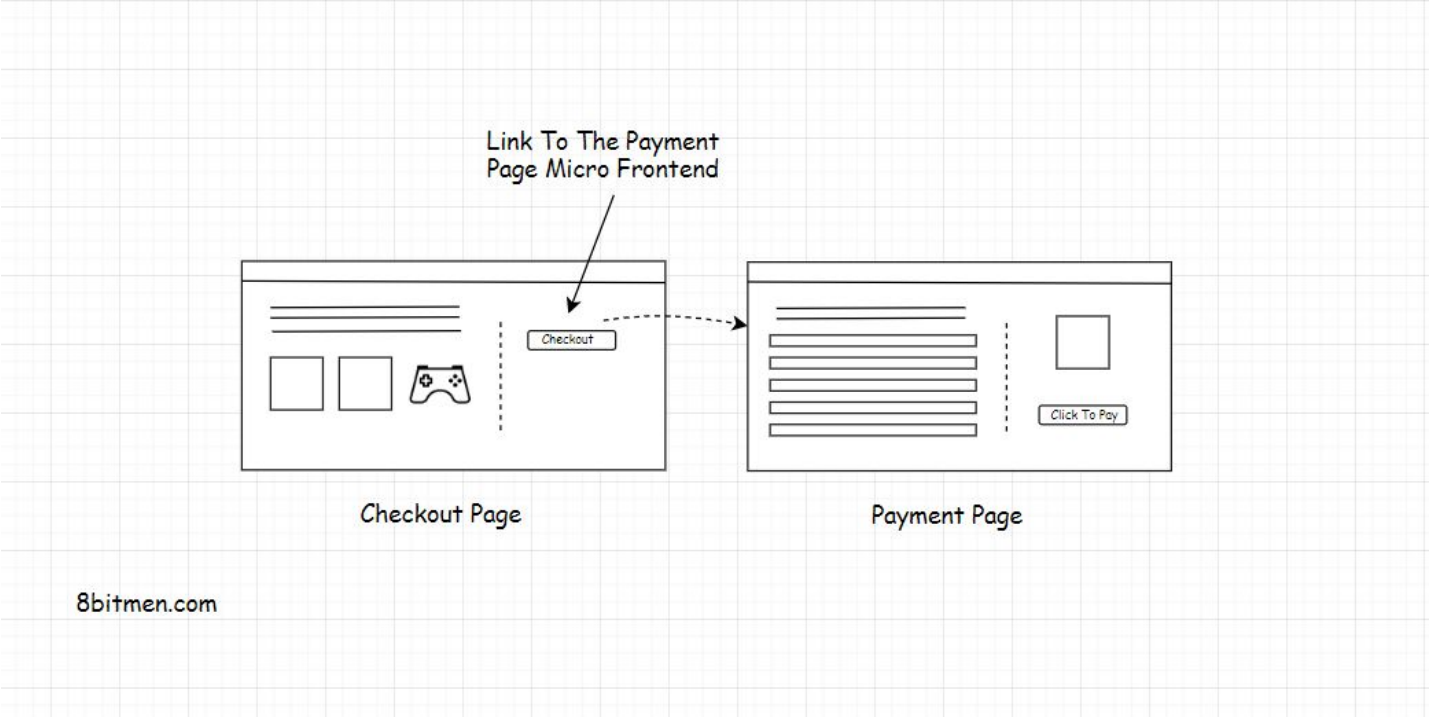
In this micro frontends use case, we need to write additional logic to enable the integration of the *UI* components.

Let's first understand the client-side integration process.

Client-Side Integration Of Micro Frontends

A very basic naïve way of integrating components on the client is having micro frontends with unique links. We just place the links on the website to enable the

user to navigate to a certain micro frontend, like so -



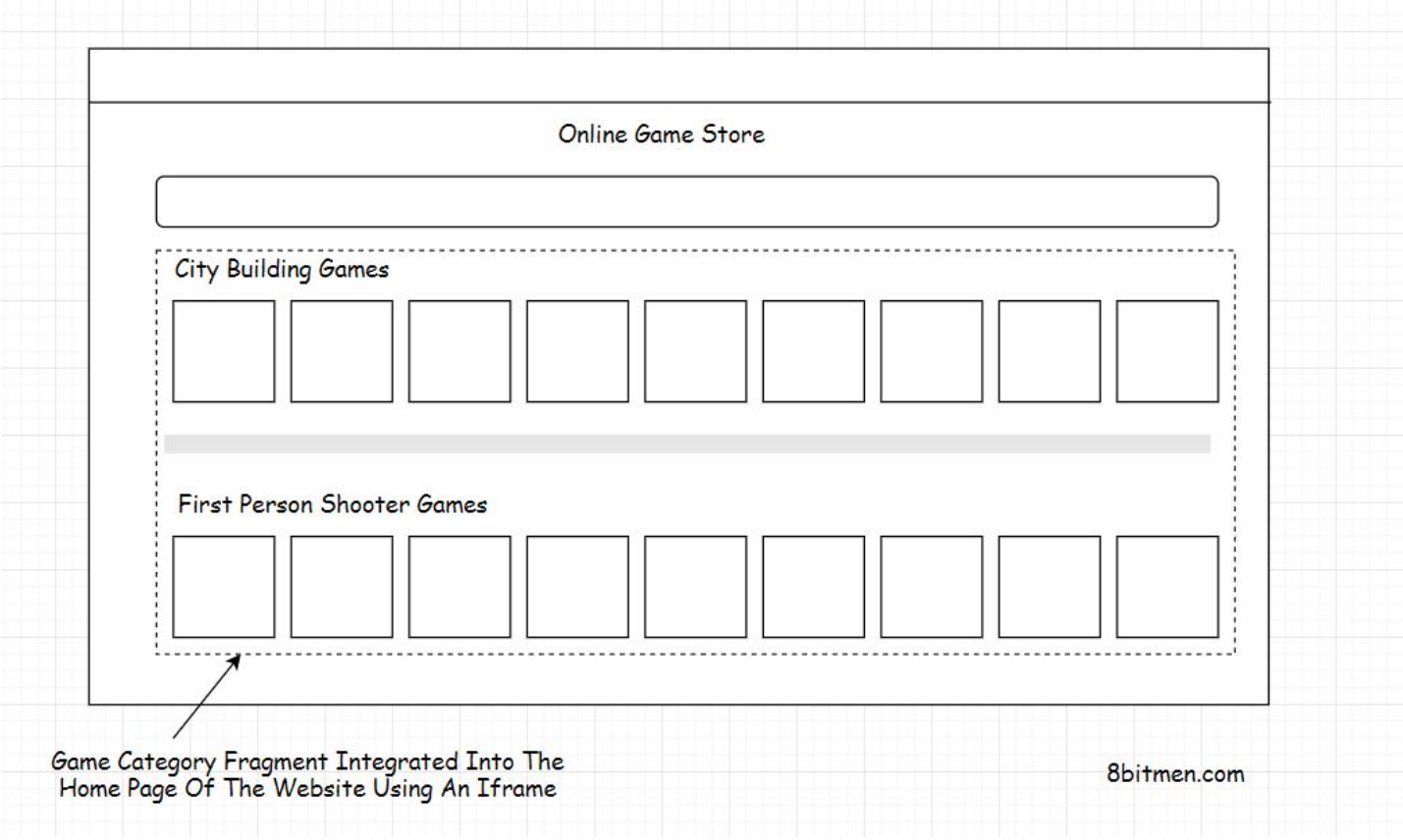
Imagine the *checkout* microservice is hosted on *AWS* having the *URL* - `https://www.aws.amazon.com/onlinegamestore/checkout` & the *payment service* hosted on *Google Cloud* with the *URL* - `https://www.cloud.google.com/onlinegamestore/payment`

When the micro frontends are integrated via basic links when the user navigates from the *checkout page* to the *payment page*, the address in the browser changes from the *AWS URL* to the *Google Cloud URL* that is visible to the end-user.

Following this basic approach, the micro frontends that need to be integrated within a specific page can be done using *Iframes*.

Well, you would have already realized, this approach is not ideal. This is more like the *90s* way of building websites, connecting web pages via direct links & *Iframes*.

Recommended Read - Good reasons why not to use Iframes in page content



A recommended way to integrate components on the client-side is by leveraging a technology called the *Web Components* & frameworks such as *Single SPA*

From the *Mozilla* documentation –

> *Web Components* is a suite of different technologies allowing you to create reusable custom elements — with their functionality encapsulated away from the rest of your code — and utilize them in your web apps.

For further info on *Web Components*, do watch this YouTube video - Web Components: The Secret Ingredient Helping Power The Web

Speaking of the *Single SPA* framework, it's a JavaScript framework for frontend microservices that enables the developers to build their frontend leveraging different JavaScript frameworks.

Recommended Watch – An Introduction To Single Spa

Canopy Tax is cloud-based accounting software for working professionals. It leverages the *Single SPA* framework to build micro frontends for its website.

Recommended Read - A Step By Step Guide To Single Spa By Canopy

Alright, now let's move on to understand the server-side integration process of micro frontends.

Server-Side Integration

When the *UI* components are integrated on the server, on the user request the complete pre-built page of the website is delivered to the client from the server as opposed to sending individual micro frontends to the client and then having them clubbed there.

This cuts down the loading time of the website on the client significantly since the browser does not have to do any sort of heavy lifting.

Just like the client-side integration process, we have to write separate logic on the server to integrate the micro frontends that are requested by the user.

There are several technologies and frameworks available that help us to achieve this.

Technology & Frameworks Facilitating Server Side Integration

Zalando is a popular fashion e-commerce company in Europe that uses *Project Mosaic* to manage its micro frontends.

Recommended Watch - The Zalando Recipe For Scalable Frontends

Open Components is an open-source micro frontends framework used for integrating micro frontends on the server.

Open Table is a *San Francisco* based online restaurant reservation company. It uses *Open Components Framework* to manage their micro frontends. Here is a talk on YouTube that they delivered on their approach.

Server Side Includes SSI is a server-side scripting language used for clubbing the content of multiple web pages on the webserver.

AutoScout24 is one of the leading internet car portals in *Europe*. Their engineering team leverages the *SSI Server Side Includes* technology to build their micro frontends. They gave a talk about their micro frontends approach here

frontends. They gave a talk about their Micro frontends approach here

Recommended Read - Page Building using Micro-Frontends and Server-Side Includes

Podium is another framework that facilitates easy server-side composition of micro frontends.

Finn.no is the largest *Norwegian* classified advertisement website in the number of page views with around *14 million* users a month. The website uses the podium framework to build its micro frontends.

Well, this pretty much sums up our micro frontends chapter, let's move on to the next one...