Basics: Micro-frontend

What is a Micro-frontend?

Micro-frontend have been a hot topic over the last few years. An increasing number of organizations have been using micro-frontends, and looking at its popularity, it might be the future of frontend web development.

Micro-frontend is an architectural style of breaking down a large application into simple, small, and manageable pieces.

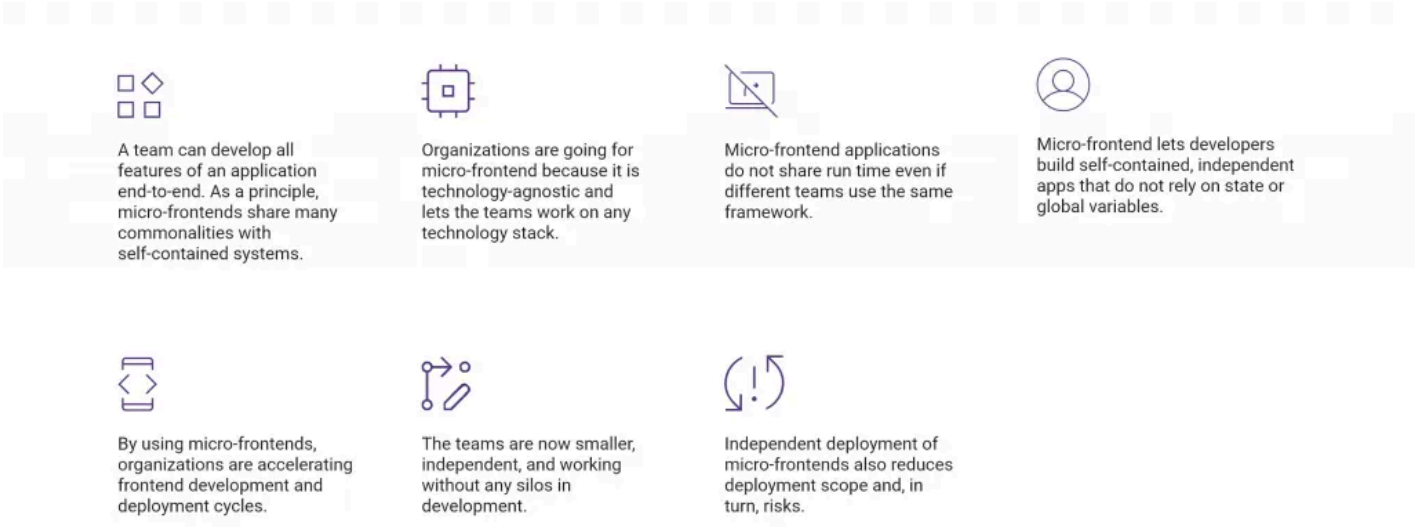
In November 2016, ThoughtWorks Technology Radar first used the term *micro-frontends* and mentioned that organizations should assess this technique.

Why Micro-frontend?

As we all know, frontend development is highly complicated. Over time the frontend layer, often developed by a large team, grows and gets difficult to maintain. That is what we call a **Frontend Monolith.** This monolith leads to communication overheads, making frontend cycles longer.

Micro-frontend breaks down this frontend monolith into smaller portions, each owned by an independent team. Each team has a specialization, and the team members are cross-functional.

Benefits of Micro-frontend



In a nutshell, micro-frontends make frontend development simple, quick, and risk-free like microservices do for backend development.

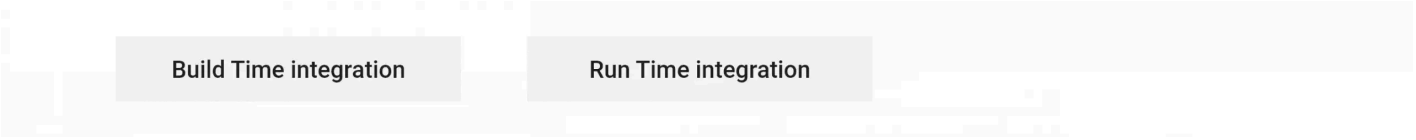
Components of Micro-frontend:

Our “how to create micro-frontends” will be more detailed in the next blog. For now, let us understand the main components of micro-frontend projects:

Micro-frontend, a small project which will be imported into a parent/container project.

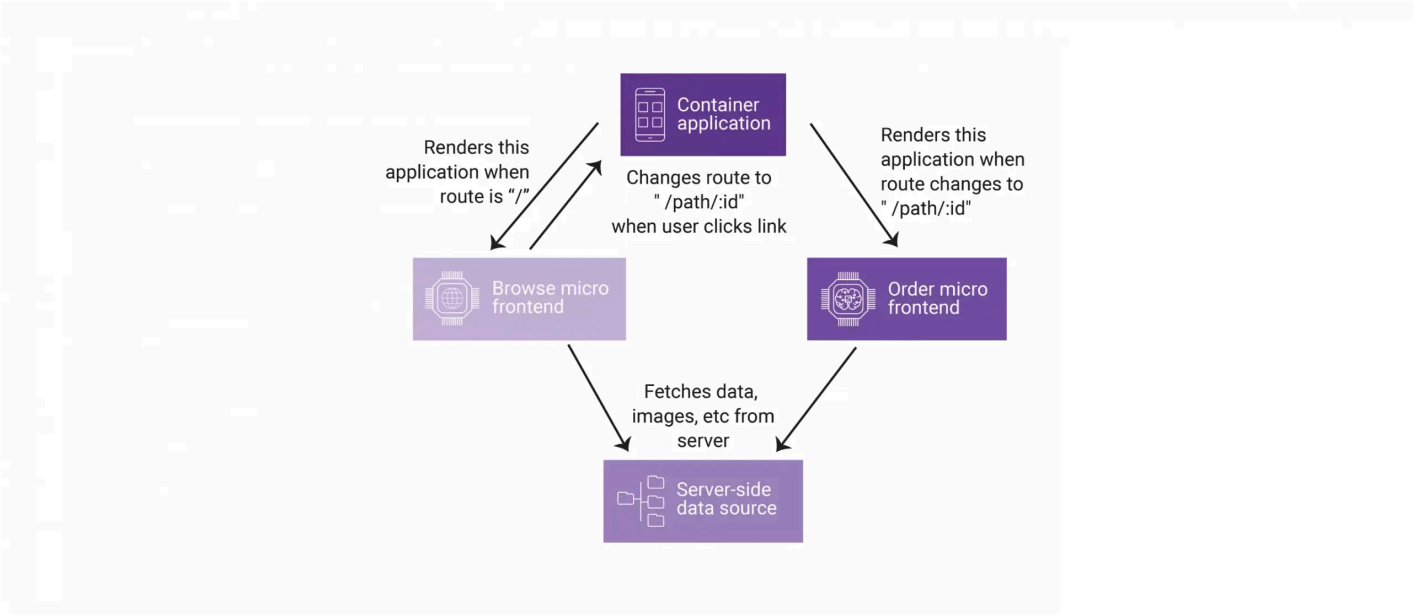
1. **A container/host**, the parent project where all micro-frontends will be hosted.
2. **Micro-frontend framework**, Webpack 5 Module Federation Plugin

The most critical aspect of Micro-frontends is the integration between the host/container and Micro-frontend applications. Micro-frontend applications can be integrated in one of two ways:

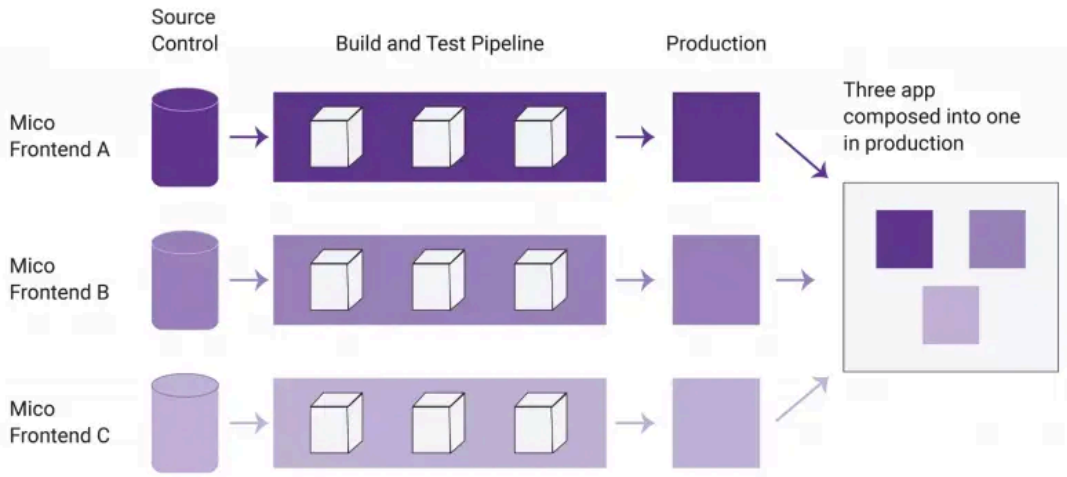


Deploying Micro-frontends:
Deploying Client-Side composition micro-frontends created with Webpack Module federation is extremely easy, efficient, and inexpensive.

Webpack Module Federation:
In June 2019, Jack Jackson, a JavaScript architect, released his masterpiece “Module Federation plugin.” **This plugin allows developers to do a dynamic import to a specific file built and deployed by a separate webpack**. This is often confused with loading a bundle on the page, but it’s not that; if you load the bundle without the webpack, it will load the entire react application, which is more than 1MB. The Module Federation gave a whole new level to micro-frontend development. As a developer, it lets you import a foreign code into your application, share codes in a straightforward way, and be more independent. It might sound complicated, but in reality, all the heavy lifting is done by the webpack.



Every micro-frontend application must have a Continuous Delivery Pipeline (CDP), so it can be built and tested separately. It should also be able to get into production independently without any dependencies. Multiple smaller micro-frontend applications in the production can then be composed together into one large working application.



In the next part, we will learn more about creating micro-frontend architecture and deploying micro-frontend applications in AWS.