MICRO FRONTEND

Saramma Varghese

WHAT IS MICRO FRONTEND?

- It is an architectural style where independent deliverable front end applications are composed into a great whole.
- Key benefits :
  - ➢ Smaller, more cohesive and maintainable codebases
  - ➢ More scalable organizations with decoupled, autonomous team
  - ➢ The ability to upgrade, update or even rewrite the parts of the frontend in a more incremental fashion that was previously possible

OBJECTIVE

- Able to take all of the little micro apps you have created , bring them together in the same webpage and give a great user experience to the users.

- Introduce a stitching layer

  it takes all of your micro apps and bundle them together before serving them to user.

TECHNIQUES USED

❑ **HYPERLINK**

- Configure some settings on server
- Then define each template to render on a separate path
- So each micro app comes on a separate path
- **Pros**:
  - Browser / server is tour stitching layer
  - No js/css conflits. Each app is separate

CONT…

- **Cons**:
  - Complete page load when switching between micro apps
  - Each app loads a complete bundle every time

❑ **IFRAMES**

- Loads each micro app in an frame.
- Stitching layer shows the iframe according to route being opened.
- Instead of loading two apps on click, we can simply load the separate URLs on different iframes

CONT…

- **Pros**:
  - Seamless stitching between micro apps
  - No js / css conflits. Each app in its own iframe
- **Cons**:
  - Routes need to be configured in the container
  - Each iframe creates an extra browser process
  - Each app loads a complete bundle every time

CONT…

- ❑ **Single SPA**
  - It do the same thing as in iframe ,but remove the iframe
  - Loads the multiple single page applications into a single DOM
  - Takes root element ,place in the body and initialize then as it required
  - Create separate container that essentially links all of these and then render every thing into the container

CONT…

- **Pros**:
  - Micro apps load a single DOM
  - Easy set up, most complexity handled by library
- **Cons**:
  - All application share the same global namespace
  - Each app loads a complete bundle every time
  - Deployment need custom tooling

CONT…

- ❑ **WEBCOMPONENTS**
  - All frameworks compile to web components
  - **Pros:**
    - Multiple micro apps share the same view
    - Everything compiles to web components
  - **Cons** :
    - CSS styling conflict
    - A lot of tooling
    - Sharing router / state is difficult

CONT…

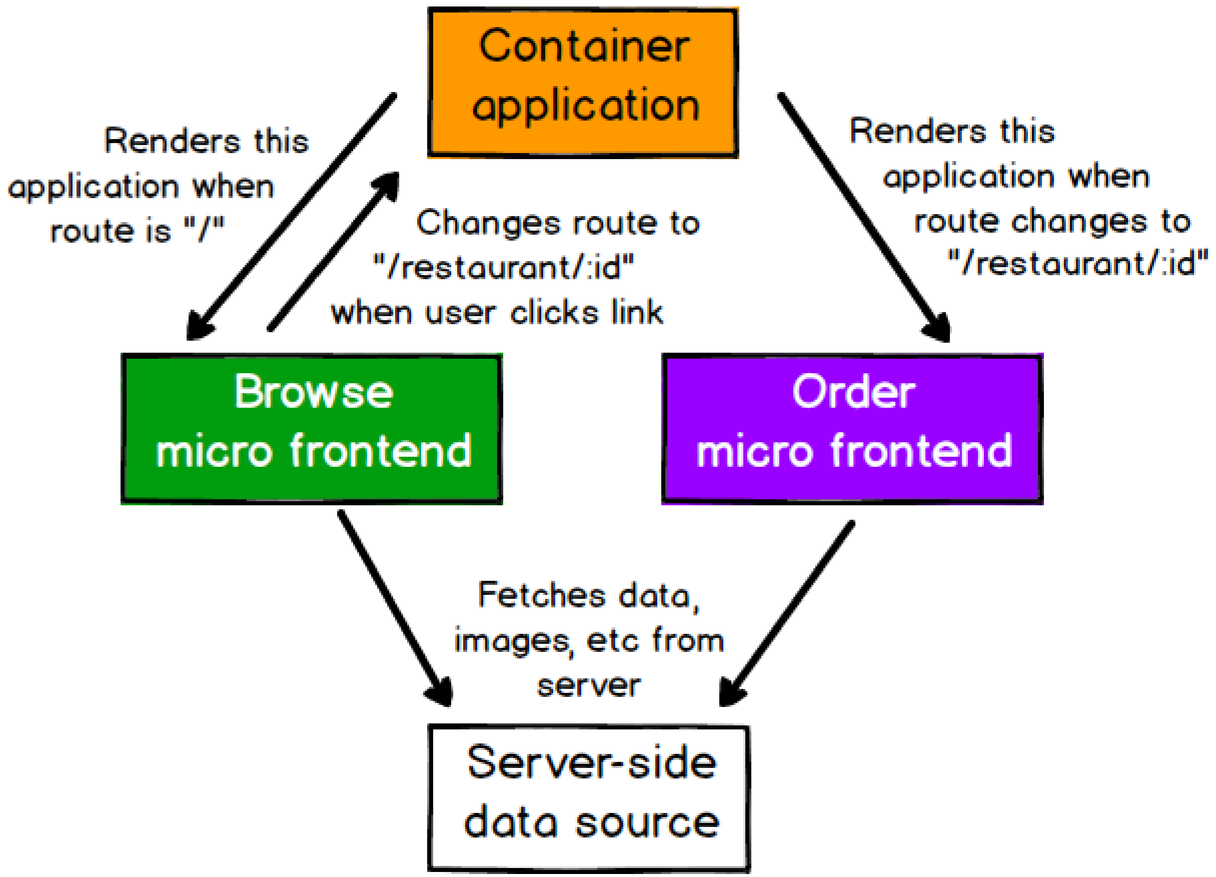
❑ OPEN COMPONENTS

- Open-source Micro Frontend framework
- Allows fast-moving teams to easily build and deploy front-end components.
- **Components**
  - Small, immutable, units of universal code mainly consisting of HTML, javascript, and CSS.
  - They can optionally contain some logic, allowing a server-side node.js application to compose a model that is used to render the view.

CONT…

    - After rendering they are pieces of pure HTML to be injected into any HTML page.
- **Template System**
    - Allows for the support of any client-side technology, hiding away all the configuration complexity while avoiding a specific UI framework.
- **Registry**
    - The registry provides a rest API to consume, retrieve and publish components to a library

MICRO FRONTENDS INTERACTION


Container application
Renders this application when route is "/"
Changes route to "/restaurant/:id" when user clicks link
Renders this application when route changes to "/restaurant/:id"
Browse micro frontend
Order micro frontend
Fetches data, images, etc from server
Server-side data source

CONCLUSION

- We can share the data from one application to another
- We can create a custom element and can use it any js application
- We can create separate components/projects based on features

THANK YOU…