

Microfrontends

What are micro frontends ?

1. Take a monolithic application and divide it into smaller multiple apps ?
2. Each smaller app is responsible for it's own feature
3. Smaller apps become easier to understand, code and maintain
4. Multiple engineering team can work in isolation

Pros

1. Better scalability
2. Faster development
3. You can use multiple frameworks for your apps
4. Deployment independence
5. Codebase is smaller and easier to manage/understand
6. Easier testing the smaller features

Cons

1. Integration (Deployment, assembly, configuration) between micro frontends can be difficult
2. With a small team, it might not be worth the struggle
3. Duplication of dependency

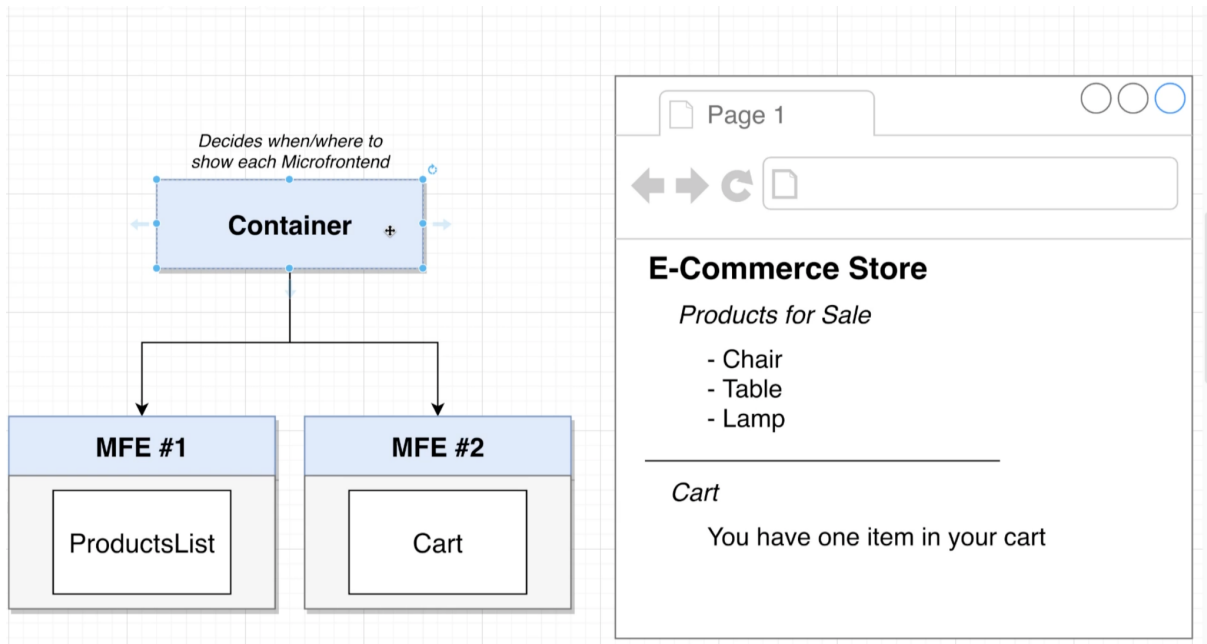
Inflexible requirements for micro frontends

1. Near Zero Coupling between child - child and container - child (As low as you can keep)
2. No shared state
3. Sharing libraries using Module federation plugin is "OKAY"
4. Css from one project should not impact other project
5. Version control (Monorepo or separate shouldn't have any impact)
6. Container shouldn't assume anything about the child (framework or any code dependency)

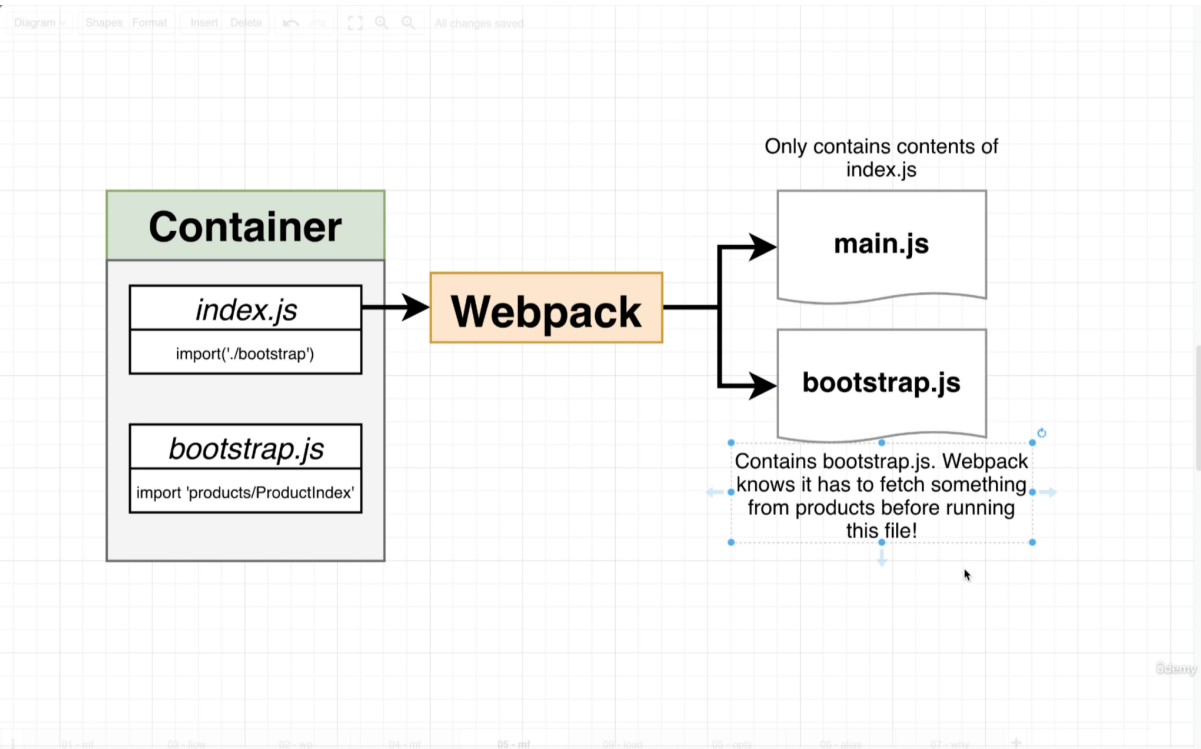
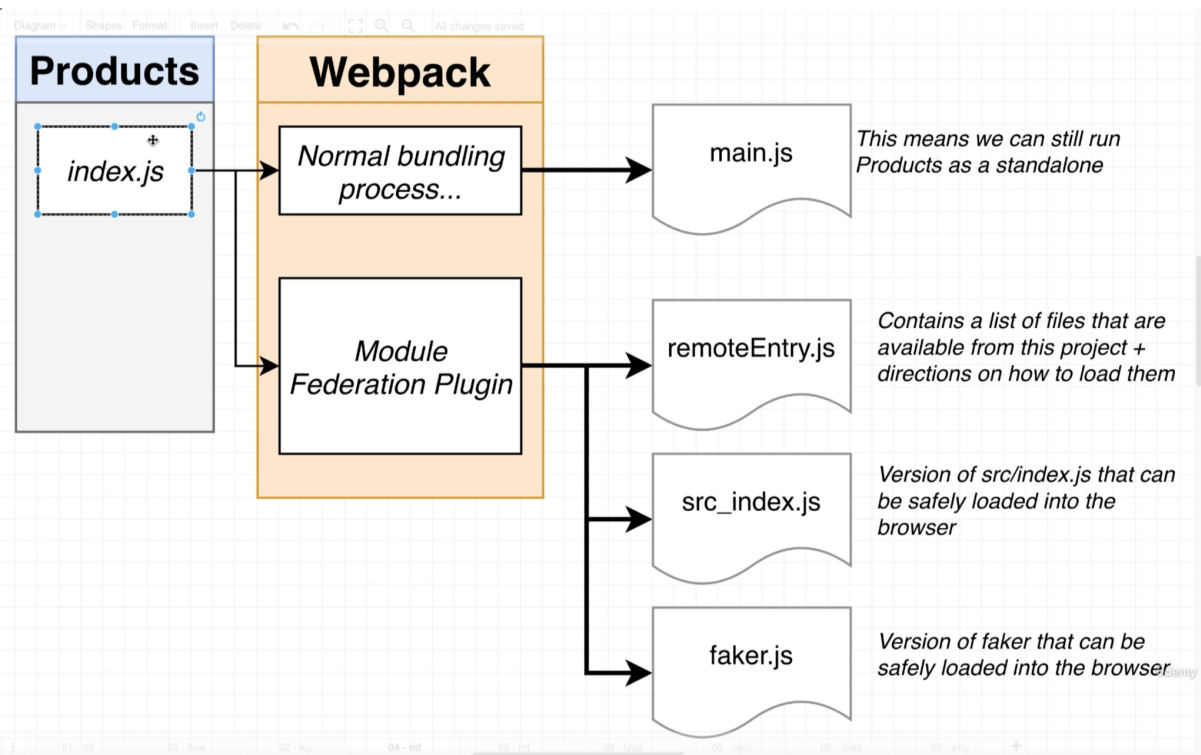
How are micro frontends integrated among themselves ?

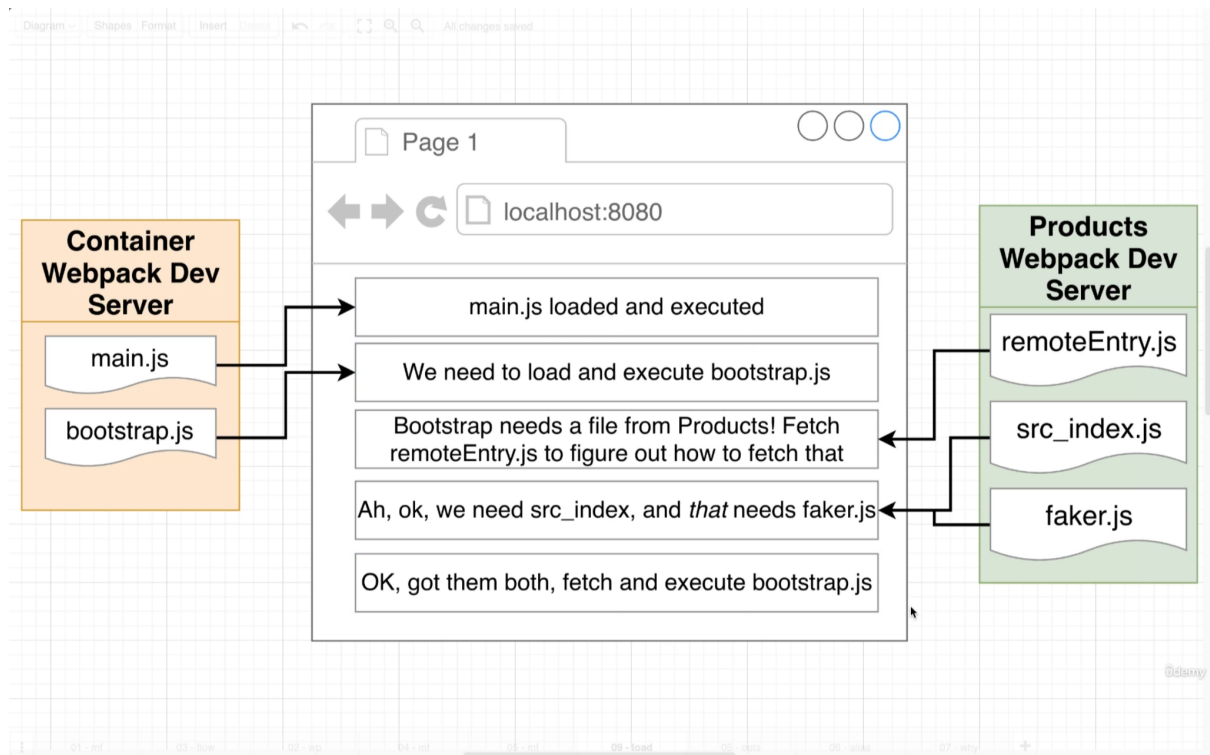
1. Build time: Before container gets loaded into browser, it gets access to other micro frontends code.
2. Run time: After container gets loaded into browser, it gets access to other micro frontend.
3. Server Integration: while sending down JS to load into container, a server decide whether to load other micro frontend or not.

Architecture



Integration:





Resources

1. <https://micro-frontends.org/>