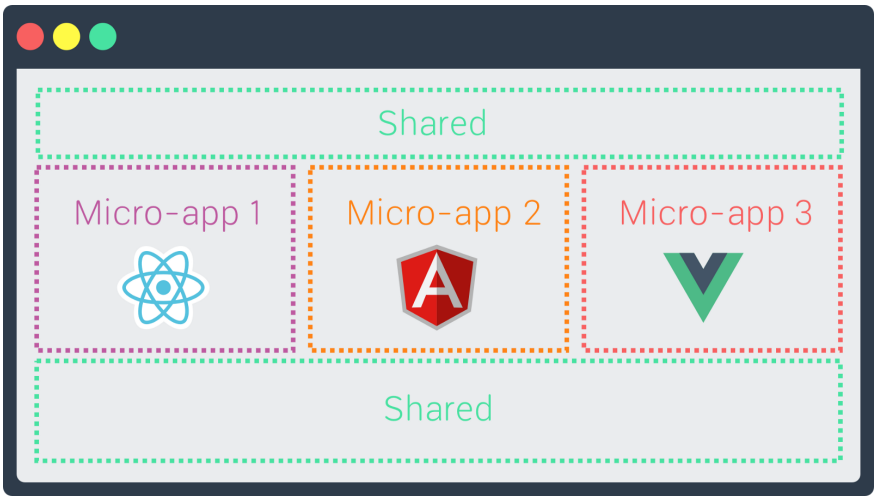
Tom Söderlund Following
CEO of Weld (www.weld.io, @weld_io). Father of two. Feminist. Fan of #ux, #leanstartup, #tabata, #javascript, #espresso
Jul 6 · 3 min read

Micro frontends—a microservice approach to front-end web development

Note: updated with references to HelloFresh, micro-frontends.org, Single-SPA (incl demo) and using a shared event bus.



For web apps, the front end is becoming bigger and bigger, and the back end is getting less important. Our web app at **Weld** (web/app creation tool) is 90% front-end code, with a very thin back end. I can imagine that a majority of new web apps being built today are dealing with a similar situation.

Web apps also change over time, as do development techniques and frameworks. This requires support for allowing different front-end frameworks to co-exist, e.g. older modules built in JQuery or AngularJS 1.x, combined with newer modules built in React or Vue.

The monolithic approach doesn't work for larger web apps

Having a monolithic approach to a large front-end app becomes unwieldly. There needs to be a way of breaking it up into smaller modules that can act independently.

Example:

- `myapp.com/` - landing page built with static HTML.
- `myapp.com/settings` - old settings module built in AngularJS 1.x.
- `myapp.com/dashboard` - new dashboard module built in React.

I would imagine the following is needed:

1. A **shared codebase** in pure JavaScript e.g. managing routing and user sessions. Also some shared CSS. Both should be as thin as possible.
2. A **collection of separate modules**, "mini-apps", built in various frameworks. Stored in different code repositories.
3. A deployment system that **bundles all the modules together** from different repositories and deploys to a server, whenever a module is updated.

The solution: "micro frontends"

But as it turns out, a lot of other people is thinking about the same ideas. The common term is *"micro frontends"*.

Companies like Spotify, Klarna, Zalando, Upwork, and Allegro, and HelloFresh are using the micro frontends approach to build their web apps.

Implementing micro frontends

Here's a few different approaches to implementing micro frontends:

1. The best solution I've seen is the **Single-SPA** "meta framework" to combine multiple frameworks on the same page without refreshing the page (see this demo that combines React, Vue, Angular 1, Angular 2, etc). See Bret Little's explanation here.
2. **Multiple single-page apps that live at different URLs**. The apps use NPM/Bower components for shared functionality.
3. Isolating micro-apps into **IFrames** using libraries and Window.postMessage APIs to coordinate. IFrames share APIs exposed by their parent window.
4. Make the different modules communicate over a **shared events bus** (e.g. chrisdavies/eev). Each module can be built using its own framework, as long as it handles incoming and outgoing events.

5. Using **Varnish Cache** to integrate different modules.
6. **Web Components** as the integration layer.
7. "Blackbox" **React** components.

Resources

- The Single-SPA framework by Canopy Tax (see mention above).
- micro-frontends.org (also on GitHub) managed by Michael Geers, with "techniques, strategies and recipes for building a modern web app with multiple independent teams".
- Project Mosaic by Zalando, a set of libraries to support a microservice style architecture for large scale websites. See also this presentation from Zalando Tech.
- Ask Hacker News: "What do you use to build micro-front ends?"

Read more

- ThoughtWorks Technology Radar: "Micro frontends"
- "Modernizing Upwork with Micro Frontends"—Sep Nasiri, Upwork
- "Introduction to Micro Frontends"—Assaf Gannon
- Presentation: "Micro Frontends: Building a modern webapp with multiple teams"—Michael Geers (JSUnconf.eu 2017 in Hamburg)
- "Microservice Grid and Micro Frontends"—Dejan Glozic
- "The monolithic frontend in the microservices architecture"—Ruben Oostinga, Xebia
- "Managing Frontend in the Microservices Architecture" (Monolithic vs. "Frankenstein" approach)—Bartosz Gałek, Bartosz Walacik, Paweł Wiląd ek at Allegro
- "Including Front-End Web Components Into Microservices"—Viktor Farcic
- "Front-end Microservices at HelloFresh"—Pepijn Senders