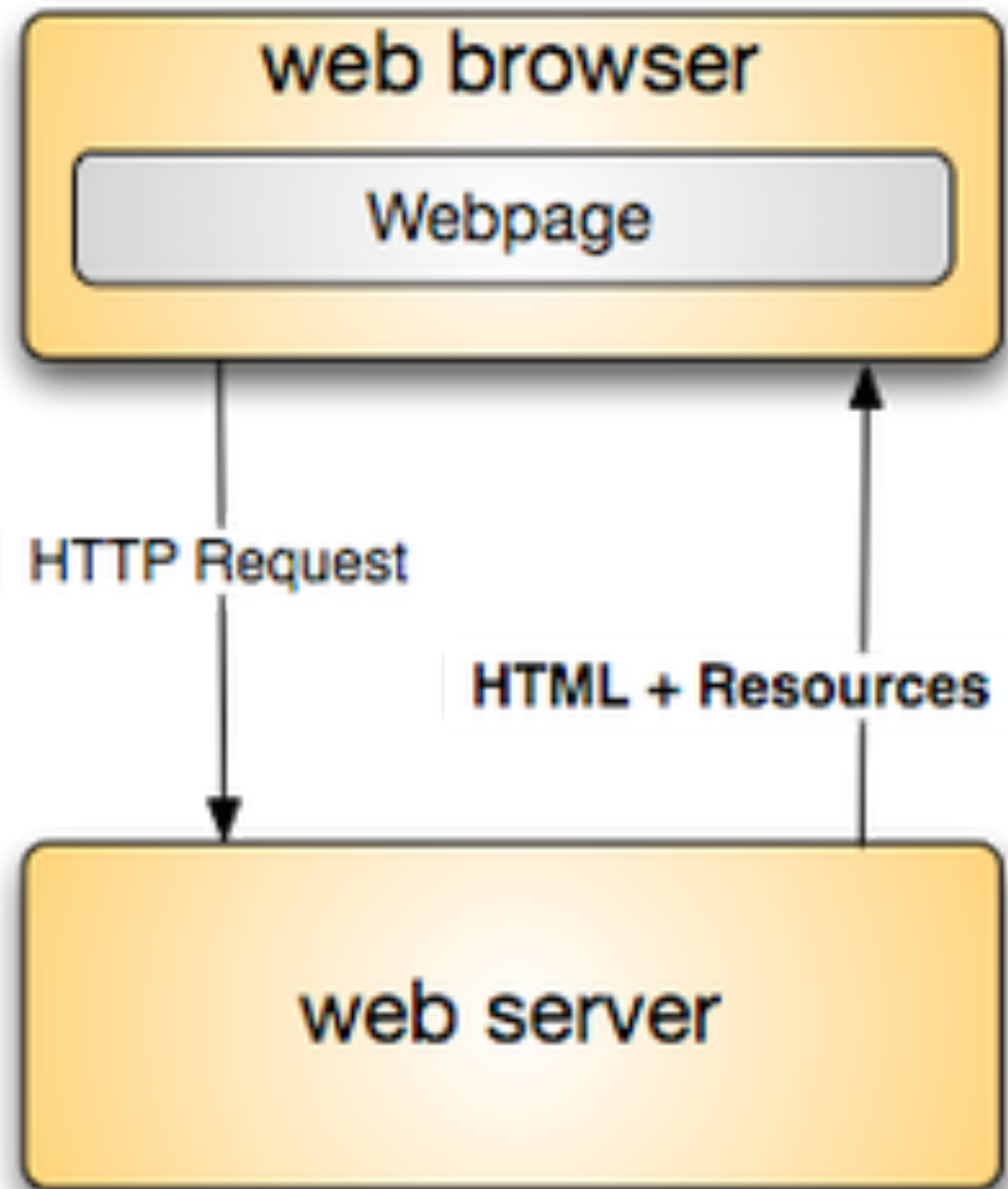


SSR

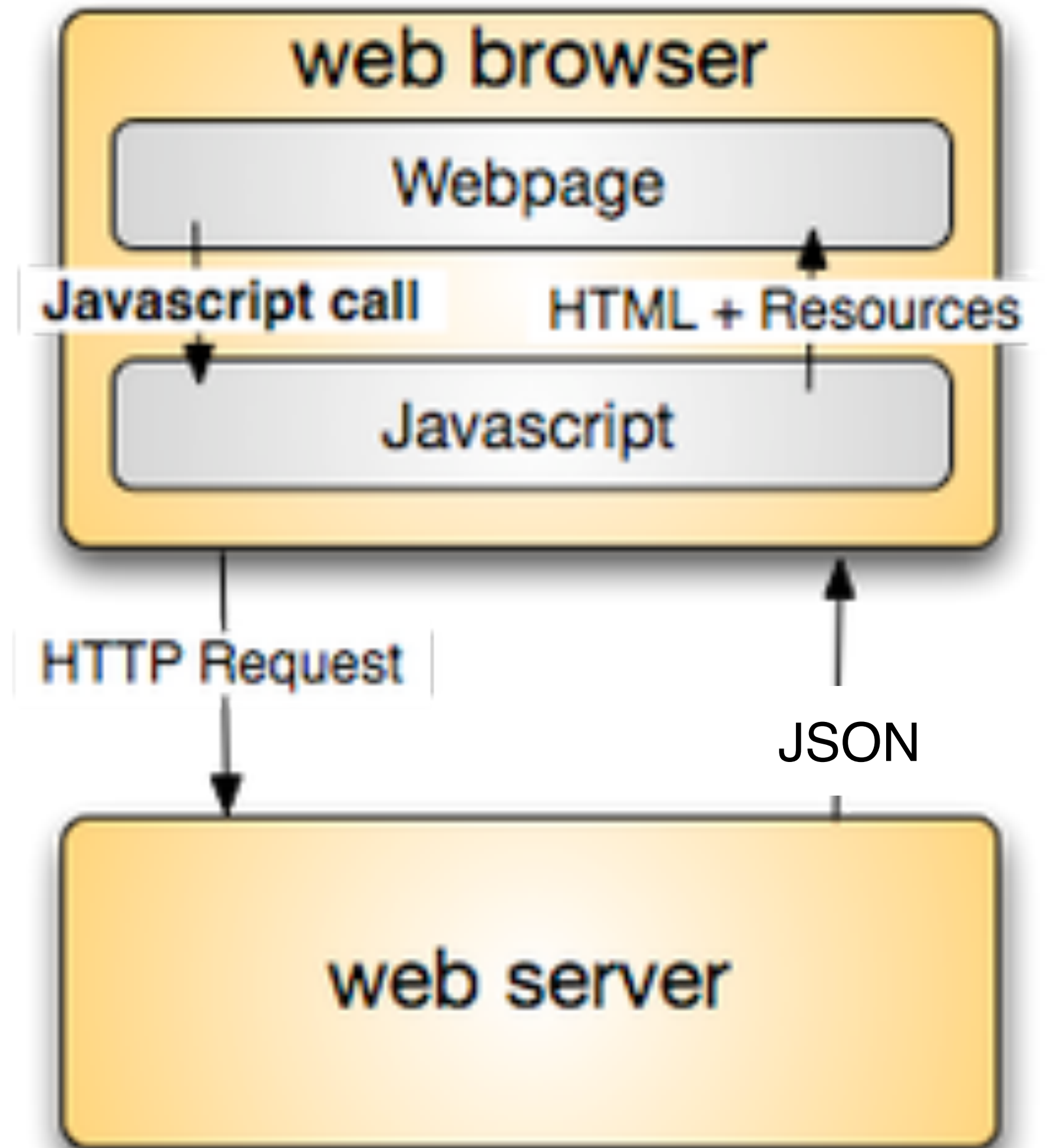


A Review of Server Side Rendering

Traditional web model



AJAX web model



Server Side Rendering (SSR)

- A feature of modern 'meta' frameworks.
- When a route is loaded, the framework may load the page on the server first - build the complete page, and then send the complete page to the browser
- Matches closely conventional 'Server Rendered Applications' - like Node/Hapi/Handlebars or PHP
- Seen as better for Search Engine Optimisation (SEO)



Server Side Rendering (SSR) - Client Side Hydration



- **Client-side hydration** a unique feature of modern frameworks (SvelteKit, Next, Nuxt)
- While the initial HTML is generated on the server:
 - Automatically include client-side JavaScript code that "hydrates" the page once it's loaded in the browser.
 - After the initial page load, client-side JavaScript takes over to handle user interactions and dynamic updates
 - Potentially provides a smoother user experience
- Seen as 'Best of Worlds' combination of SPA and traditional Server Render Applications

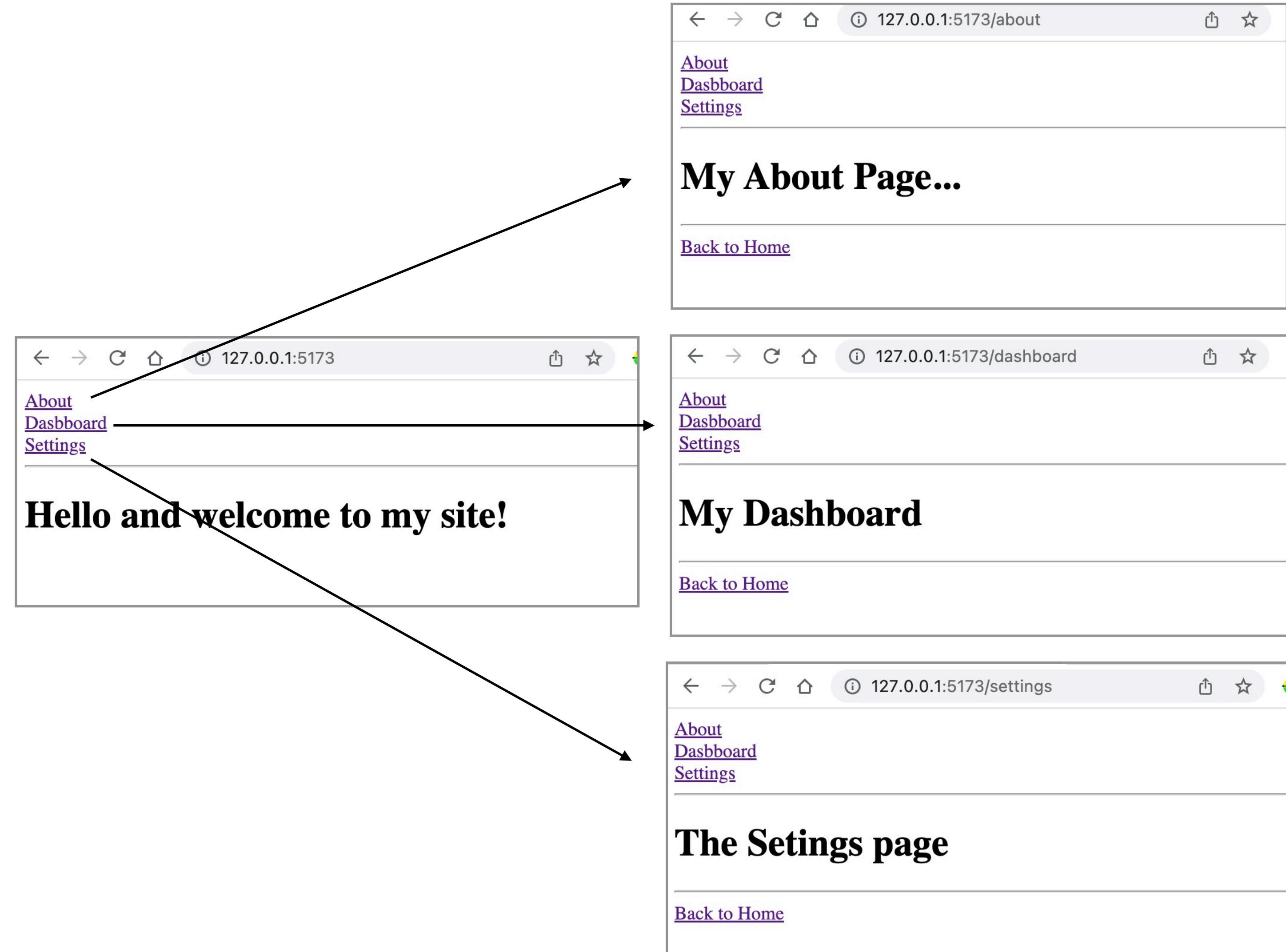
Server Side Rendering (SSR)



- **Improved Initial Page Load Performance:** SSR routes provide fully rendered HTML content to the client, which can significantly reduce the time it takes for users to see content on the page. This is particularly beneficial for content-heavy websites or applications where fast initial page load times are crucial for user experience.
- **Search Engine Optimization (SEO):** Search engine crawlers can more easily index content from SSR routes because they receive fully rendered HTML from the server. This can lead to better search engine rankings and increased visibility for the application in search engine results pages (SERPs).
- **Accessibility and Progressive Enhancement:** SSR routes ensure that content is accessible to users who have JavaScript disabled or are using devices with limited capabilities. By providing server-rendered content as a baseline, applications can progressively enhance the user experience with client-side JavaScript interactions.
- **Improved Performance on Low-Powered Devices:** SSR routes can provide a better user experience on low-powered devices or slower network connections by reducing the amount of client-side processing required to render the initial page.
- **Consistent User Experience:** SSR routes help ensure a consistent user experience by providing a predictable rendering process across different devices and browsers. This can help minimize issues related to client-side rendering inconsistencies.

Client Side Rendering (CSR)

- Client-side rendering (CSR) is the generation of the page contents in the web browser using JavaScript.
- Server-side rendering (SSR) is the generation of the page contents on the server.
- SSR is significantly superior for SEO (Search Engine optimisation)



Server Side Rendering (SSR)

CSR

1.



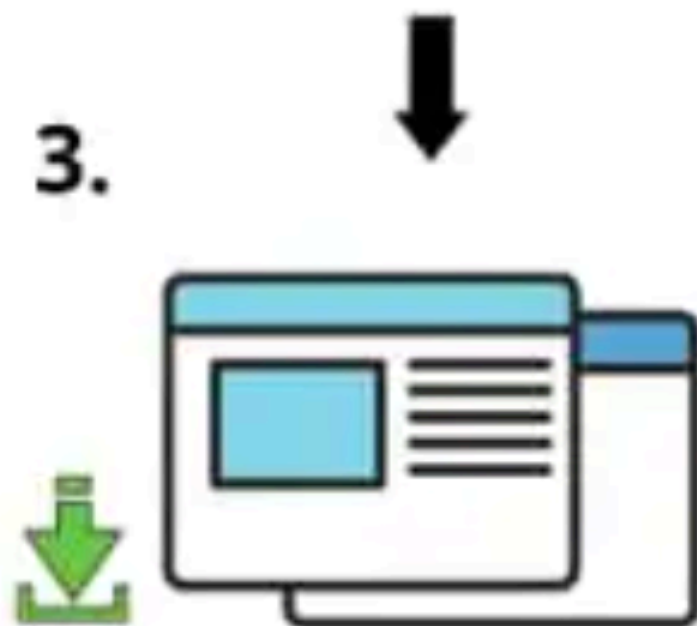
User requests a website

2.



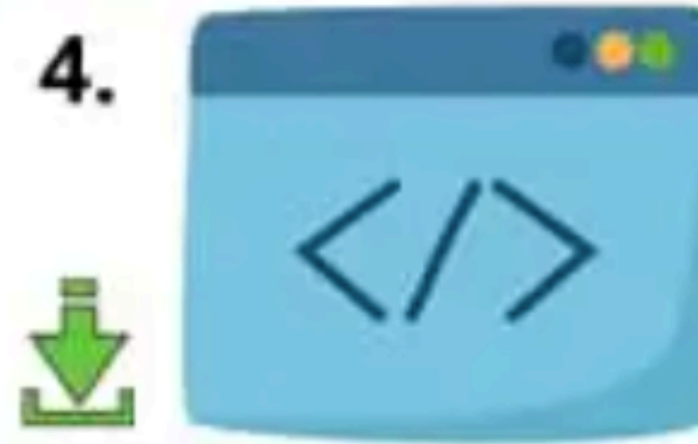
a CDN can quickly serve HTML files with links to JS

3.



Browser downloads the HTML & then the JS, meanwhile the site isn't visible to the user

4.



The browser downloads the Javascripts

5.



The JS is then executed, APIs are called for data, & user sees placeholders

6.



The Server responds with the data asked by the API

7.



The data from the APIs fill the placeholders & the page is now interactive

SSR

1.



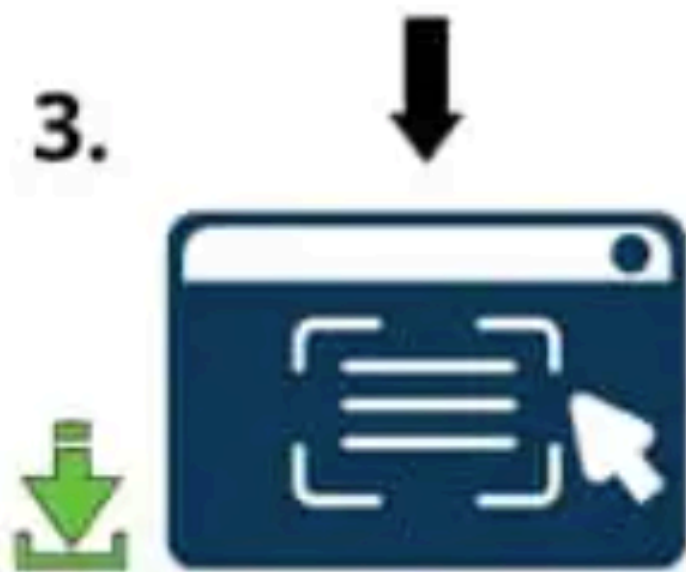
User requests a website

2.



Server creates "Ready to Render" HTML files

3.



The browser can quickly render the HTML but the site isn't interactive

4.



The browser downloads the Javascript

5.



The user can view content & the interactions can be recorded

6.



The browser executes the JS framework

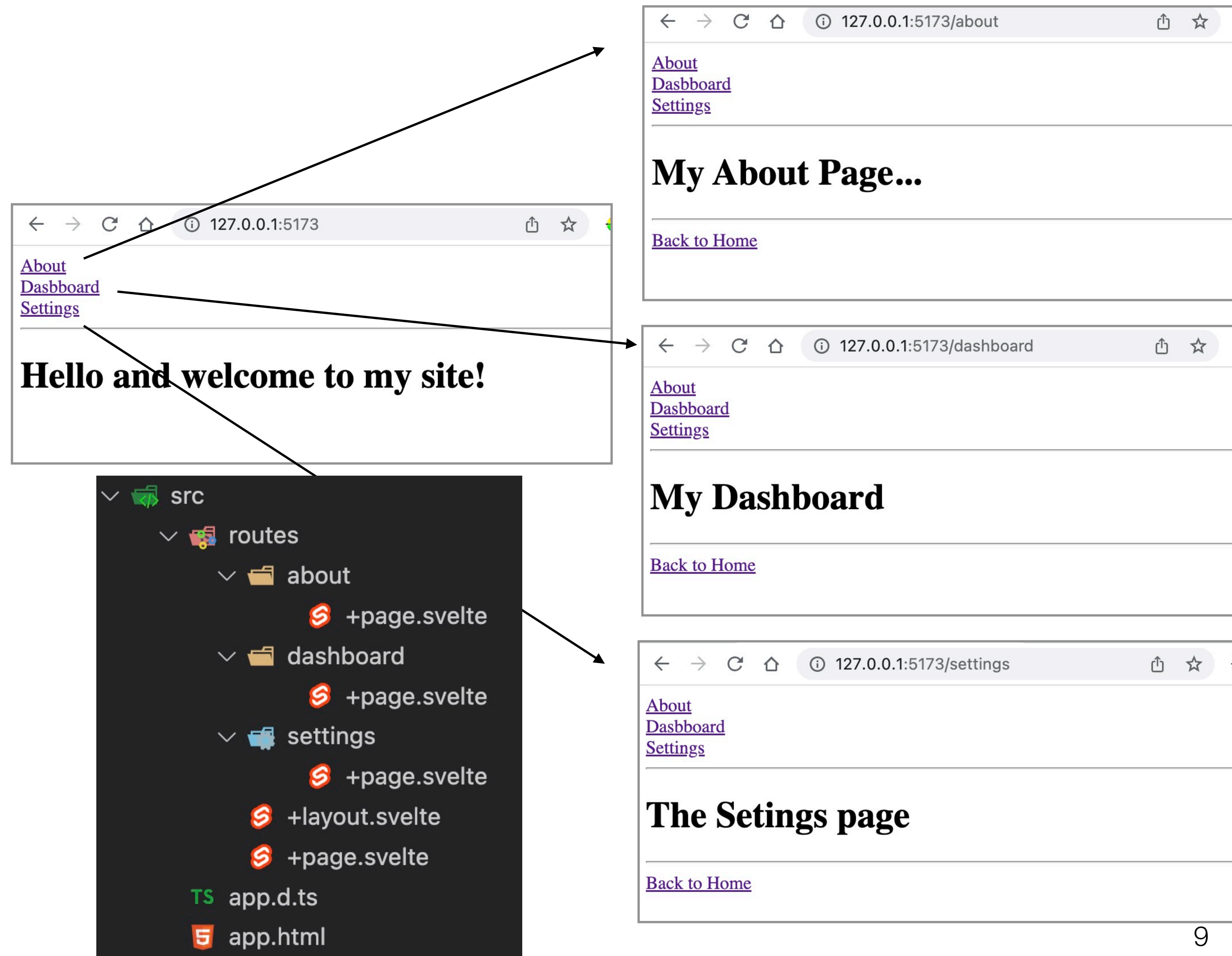
7.



The recorded interactions can be executed & the page is now interactive

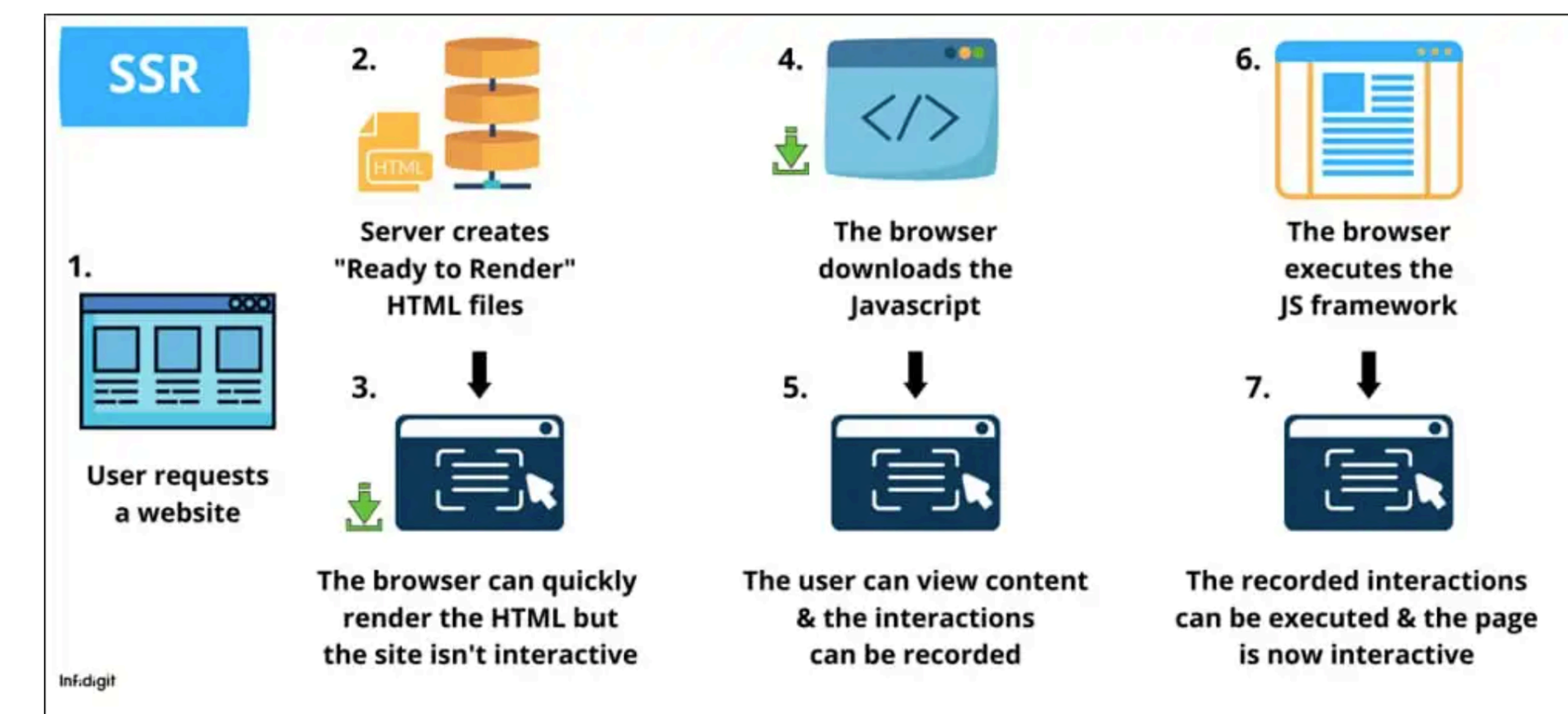
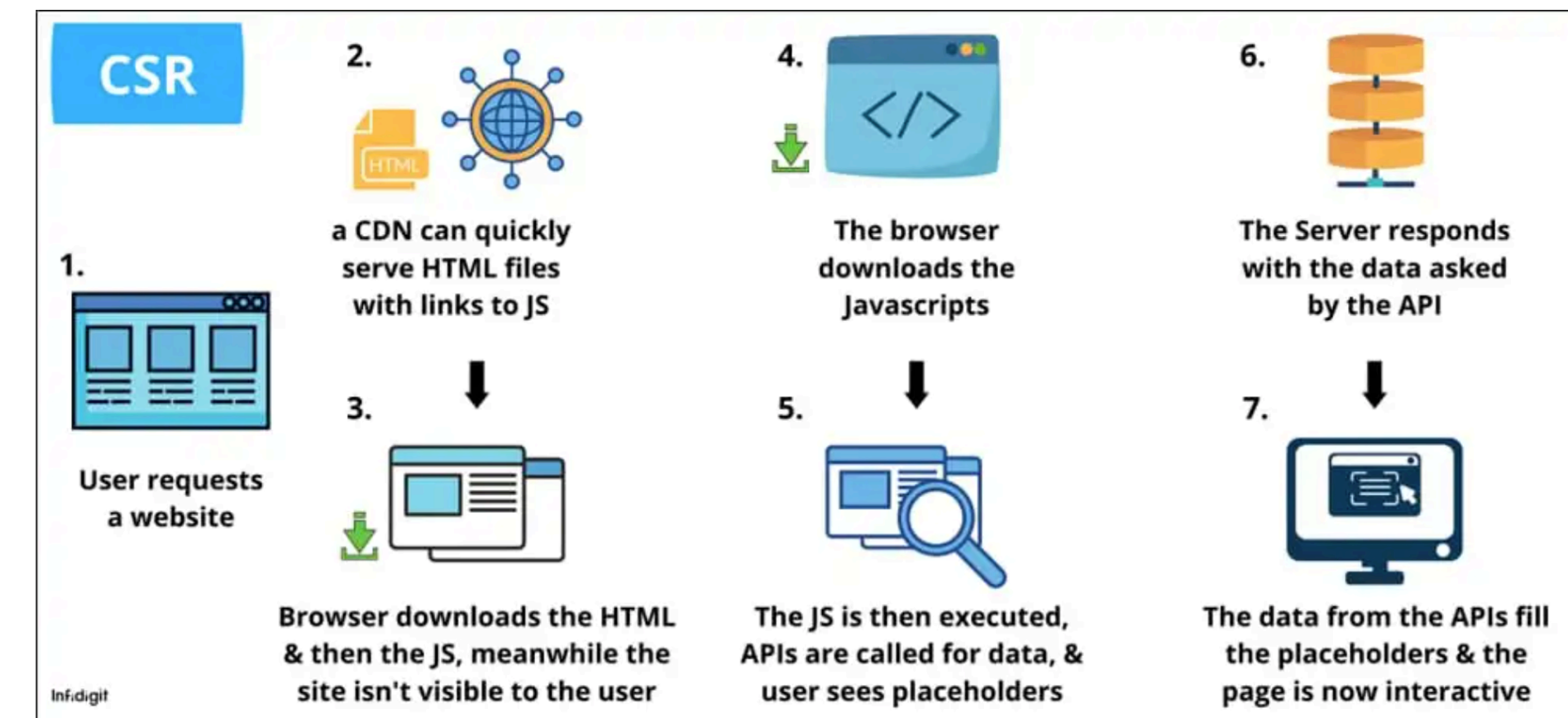
Client Side Routing

- By default, when you navigate to a new page (by clicking on a link or using the browser's forward or back buttons), SvelteKit will intercept the attempted navigation and handle it.
- SvelteKit will then update the displayed contents on the client by rendering the component for the new page
- This process of updating the page on the client in response to attempted navigation is called client-side routing.



SvelteKit Rendering Model

- By default, pages are also server-side (SSR) rendered when you first visit a page.
- When you navigate back to a page that has already been visited, SvelteKit will use client side rendering (CSR)
- This will improve the perceived performance of the app (it will feel like an SPA), but retain potential SEO benefits of an MPA model
- This behaviour can be customised -
 - Full CSR (SPA like)
 - Full SSR (MPA like)
 - Per page behaviour CSR/SSR



SSR



A Review of Server Side Rendering