Arquitectures:  
  
**Front-end** : React (SAP: Single App Page), basically it’s a one page rendered that simulate navigation between pages, just switching on/off components. Example: Navbar could be a component that is always on (visible). SAP as React has less impact on client’s dispositive because it doesn’t reload full page, only reload a component when it’s needed.  
  
The way of how to be programming in React, will depends on programming paradigm used at backend.  
  
**Backend**: using Laravel with PHP. There are 2 paradigms, server-side-rendering (SSR), client-side-rendering (CSR).

* server-side-rendering (SSR): Laravel it’s a technology that works mainly with this paradigm. It could be hard to pair with React or Vue for front-end, so a solution is use Inertia framework. Inertia transform Laravel responses into a React’s component (server-side) and send it to client-side. However, it also has some disadvantages, such as:  
  + It is expensive and resource-intensive for the engineering team.
  + It tends not to work with third-party JavaScript.
  + The page is displayed quicker, but users still can’t interact with content until all resources have been loaded.
  + It requires frequent server requests and full page reloads.
  + It may degrade performance if the application is heavy or complex.
* client-side-rendering (CSR): Another approach is use “response()->json” Laravel’s method, and send responses as JSON to frontend side, and control routing and rendering with React (client-side). This method is not friendly for non-React programmers, but makes possible the use of libraries like react router, react context, react flux, react redux, and of course, all the libraries of nodejs like Material UI or Bootstrap for visuals.  
    
  example of response()->json method:  
  Graphical user interface, text

  Description automatically generated