

Next 介紹

Eddy Chang

✉ hello@eddychang.me



Next 是什麼

- 建立在 Node.js 上，基於 React 技術熱門開源碼網站框架
- 主要用於伺服器端渲染(SSR)與伺服器端生成(SSG)
- 2016 年開始發展，目前由 Vercel 公司主導開發維護
- Netflix、GitHub、Uber、微軟、星巴克、任天堂等為實例客戶

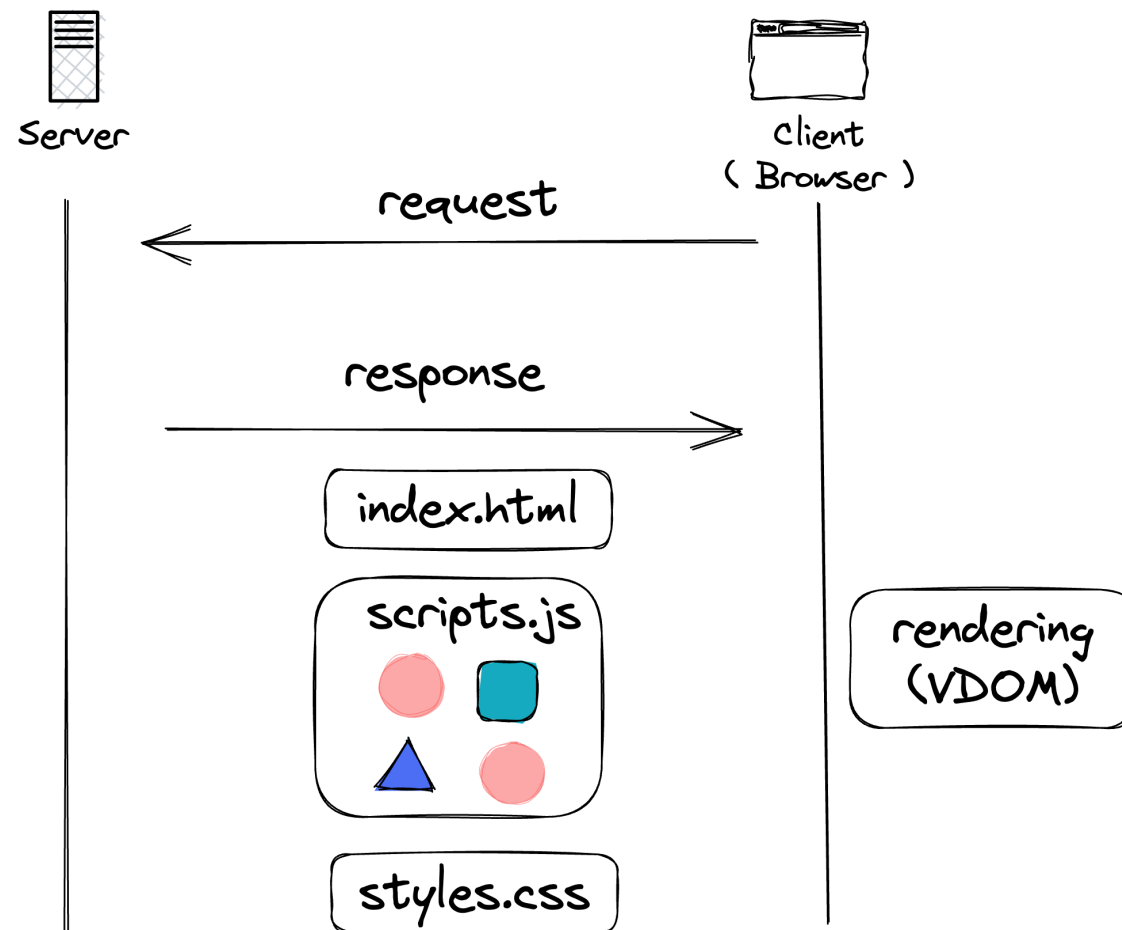
Next.js 特色

- 易於學習使用: 基於 React 開發，具有 React 基礎很容易可以上手使用
- 內建 CSS 功能: CSS-in-JS 功能與 styled-jsx
- TypeScript 自動支援
- 多樣的資料 Fetch 技術: SSG/SSR
- 檔案系統的路由

CSR

Client Side Render

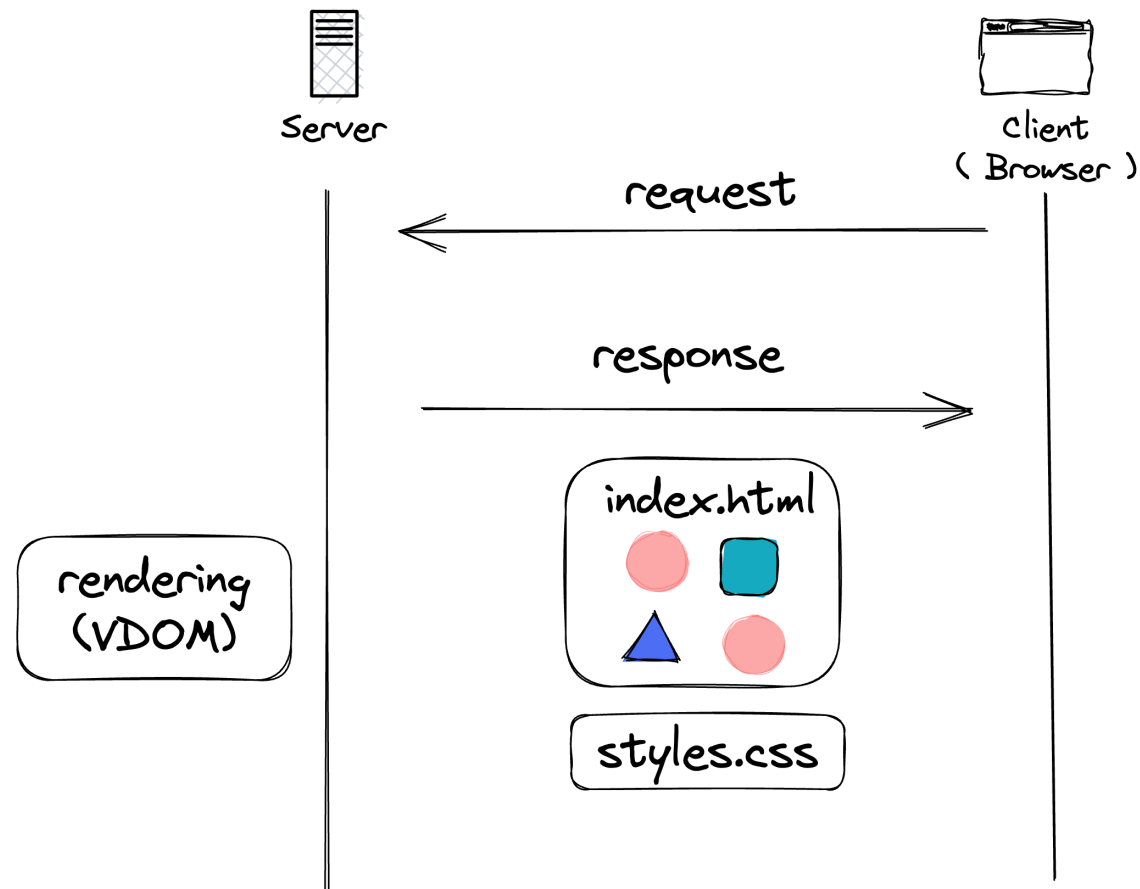
網頁上的所見 HTML 內容，是在客戶端(瀏覽器端)執行後進行渲染再提供它們，一開始伺服器會回應一個空白內容的 HTML 文件。JS 框架如 React, Vue, Angular 的 SPA(單頁式應用)指的就是以 CSR 的方式進行。



SSR

Server Side Render

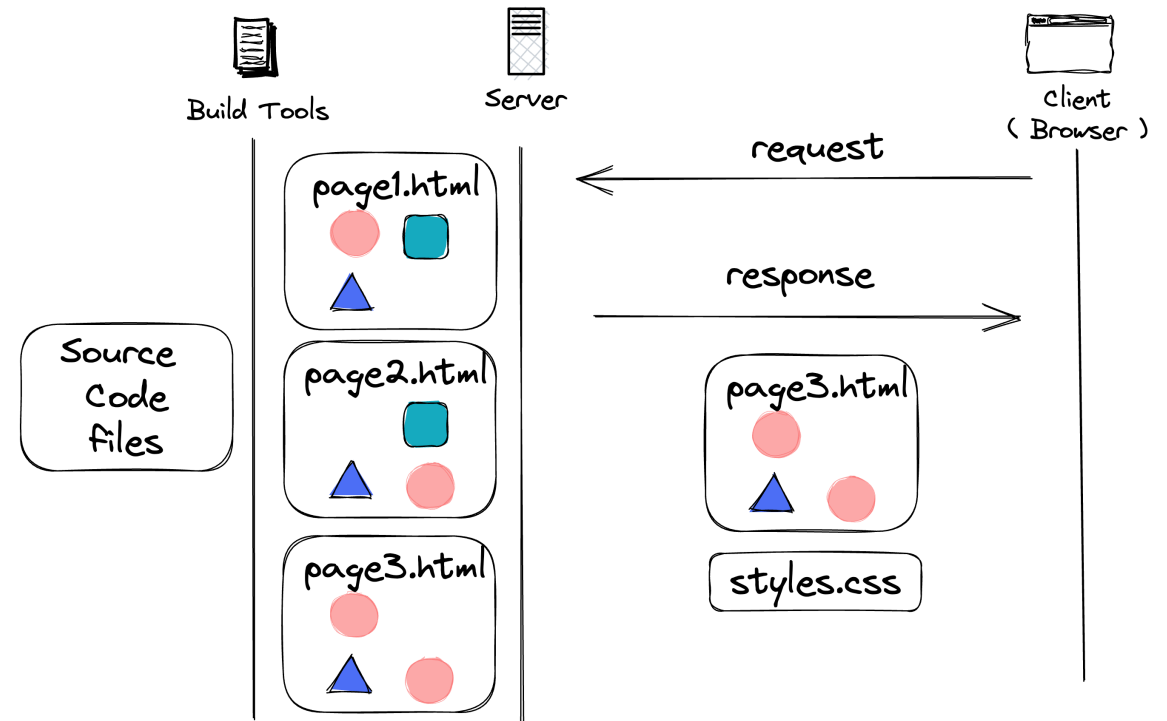
網頁上的所見 HTML 內容，伺服器每次收到客戶端要求後，才在伺服器端進行渲染產生頁面，再回應給客戶端。與 SSG 的最大不同處，在於它雖然是一種預渲染 (pre-render) 機制，但需要對每次要求伺服器都要進行渲染的過程。



SSG

Server Side Generation

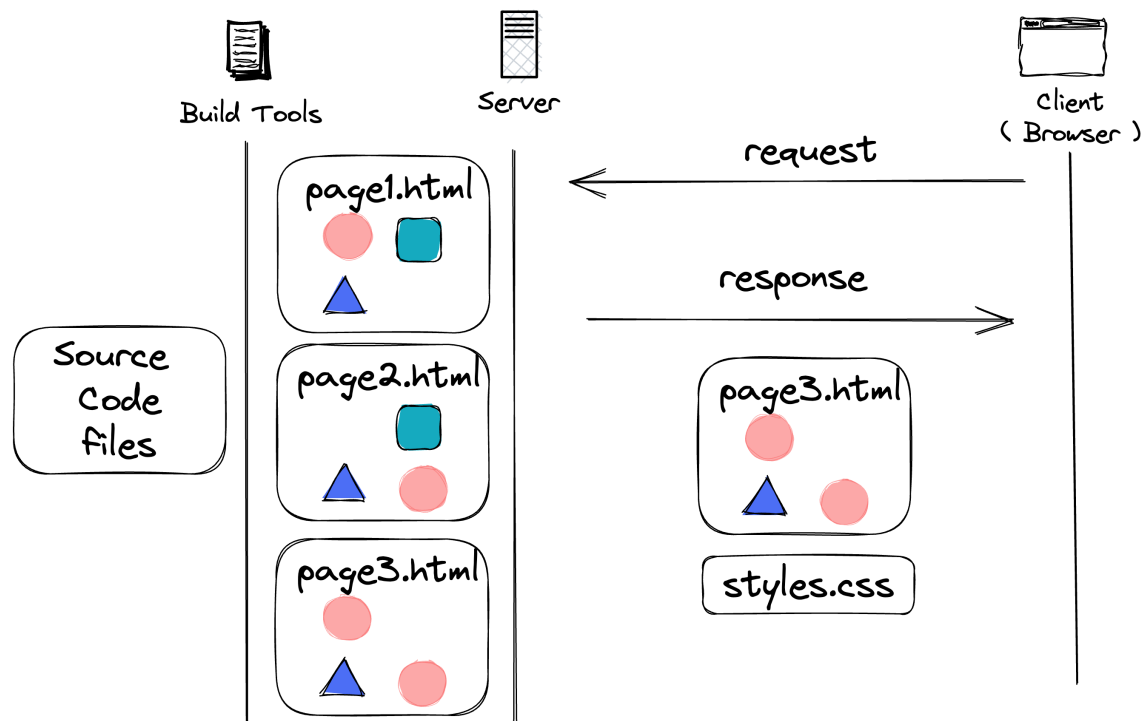
網頁上的所見 HTML 內容，是由伺服器**預先**在伺服器端渲染建立產生 HTML 文件，再視要求回應給客戶端。



ISR

Incremental Static Regeneration

ISR 是一種漸進的 SSG 改善方式，可以周期性地建立或驗證過期的頁面，然後重新產生。它也是一種預渲染(pre-render)機制，屬於進階式的應用。



應用技術比較

種類	CSR	SSR	SSG
說明	在客戶端(瀏覽器)渲染	在伺服器端渲染	在打包期間渲染產生靜態頁面
優點	與伺服器反應時間最少，開發彈性高	SEO 好，開發彈性高好	SEO 好，伺服器負擔輕，達成更好的快取機制
缺點	SEO 差	伺服器負擔重	多了打包建立過程與時間，只適合純靜態網頁

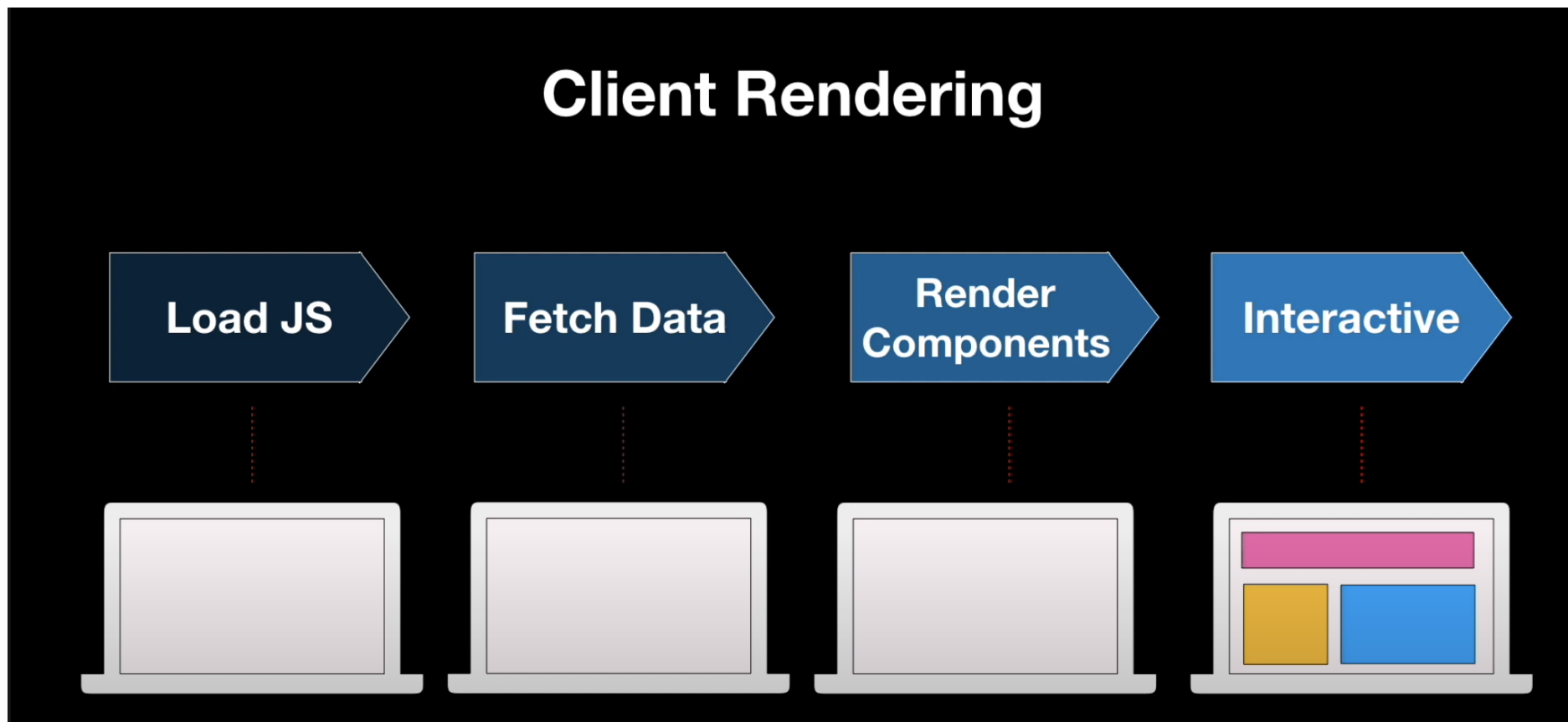
註: 各技術實務上可混合使用，以上是從某單一技術應用比較

靜態網頁(Static) vs 動態網頁(Dynamic)

種類	靜態網頁	動態網頁
說明	單純使用 HTML/CSS/JS 製作的網頁，每個路徑對應一個頁面	伺服器每次依要求來產生內容(通常存在資料庫中)的網頁
技術	HTML/CSS/JS	PHP, Nodejs, .net...
優點	容易開發製作，主機服務效率高、便宜	容易改變內容，開發各種功能
缺點	無法提供功能性(登出入，表單填寫)...	伺服器負載重，主機費用高

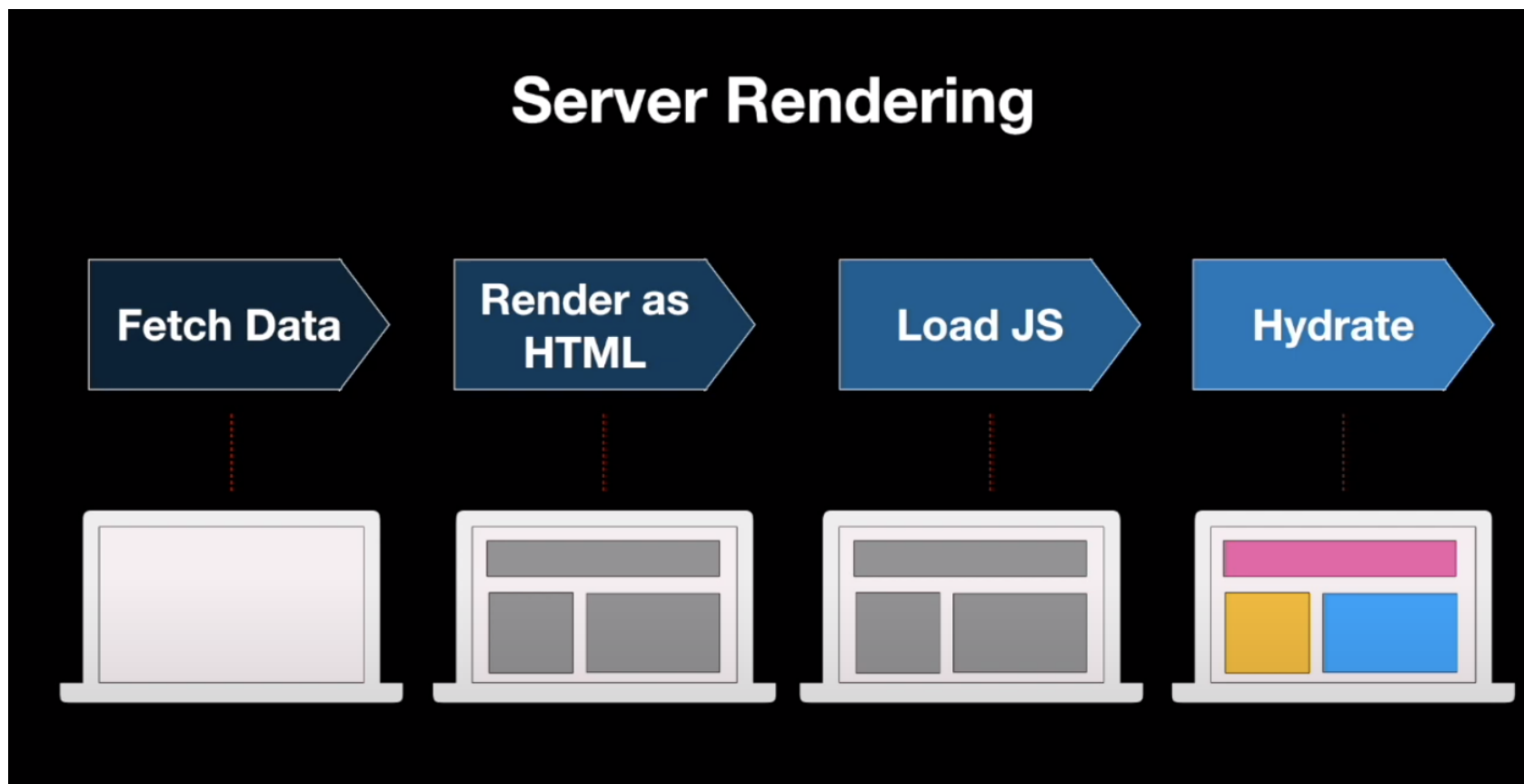
註: 大致上可以認為"靜態"或"動態"網頁的角度是以"伺服器端"的應用技術來決定的。在客戶端執行的 JS 如果有 AJAX、fetch 等技術應用，可以說它是一個不單純的靜態網頁而已。

客戶端渲染(CSR)



from Shaundai in [React Conf 2021](#)

伺服器端渲染(SSR)



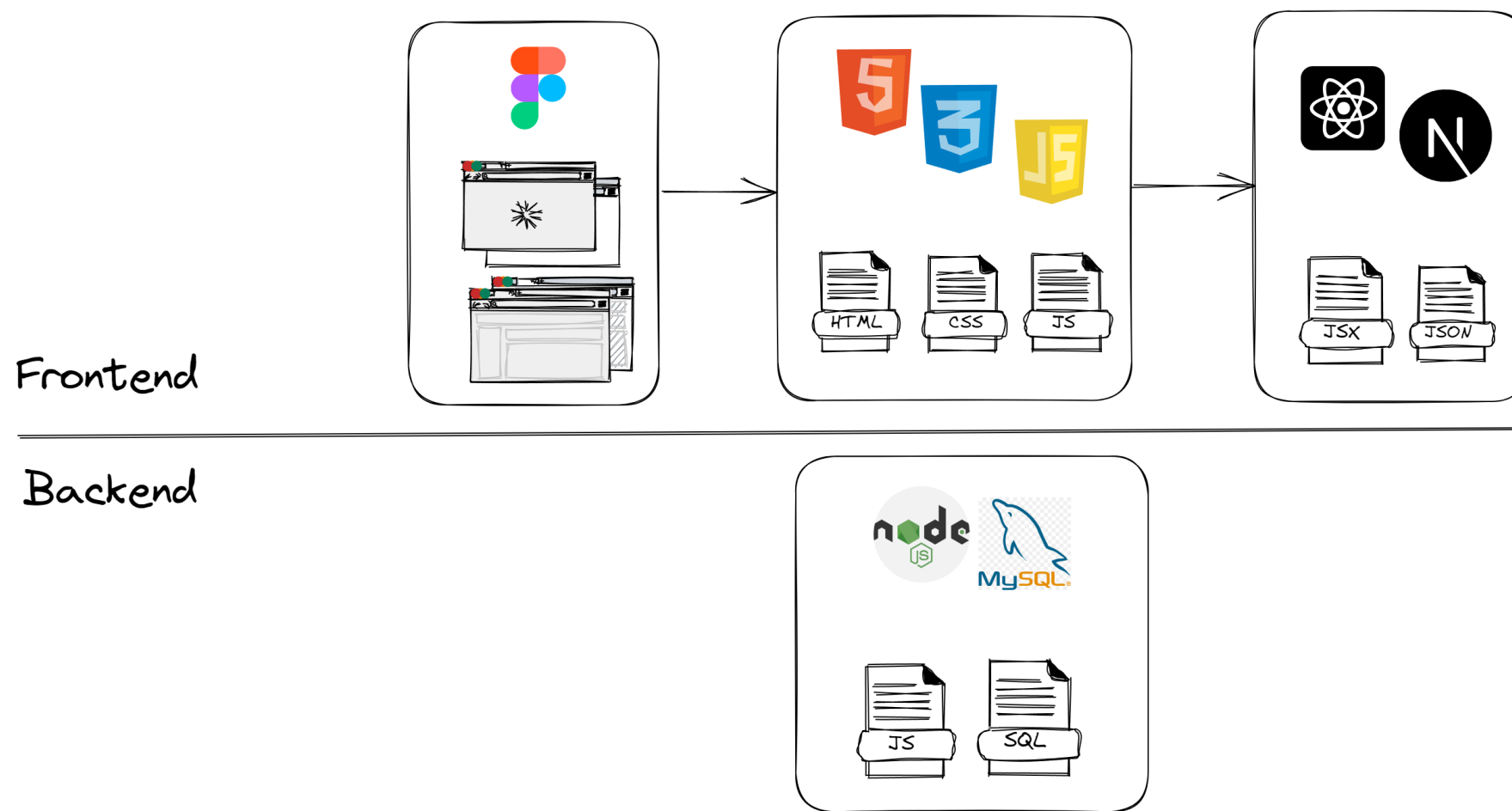
from Shaundai in [React Conf 2021](#)

水合作用(hydration)

SSR 專用術語，使用類似於 React 原本的瀏覽器 DOM render 技術，但在伺服器進行。取而代之的是，會先讓使用者載入一個已具有 HTML DOM 元素網頁(類似框架灰頁)，再載入對應的 JS 碼，之後進行 **水合化(hydrate)**，進行檢查已載入的 DOM 的結構是否相符，並將相對應的事件加上，使之成為 CSR 元件，具有事件監聽和互動性。

更多參考: [官方 React 18 SSR 架構說明\(有圖解\)](#) / [hydration SSR 圖解說明](#)

實作說明(圖解)



實作說明(Next vs Node)

- Next: React 應用 + SSG 技術(產生靜態頁面)與路由
- Node/MySQL: RESTful 伺服器 + 資料庫 + 會員認證...等

實作流程說明

1. **Figma**: 設計網站中各網頁呈內容
2. **HTML/CSS/JS**: 將設計雛形轉為各 HTML/CSS 頁面
3. **JSX 化**: 將各 HTML 頁面，轉為 JSX 檔案與各元件，導入 Next 中與建立路由
4. **分離出資料模型**: 將所需動態獲得資料集中到 JSON 檔案，定義各變數名與資料模型
5. **前端網站雛形實作**: 實作前端網頁功能與各特效、動態操作介面
6. **資料庫設計與 SQL 實作**: 設計各資料表，與實作各所需 SQL
7. **RESTful 伺服器 API 實作**: 實作各所需 RESTful API
8. **整合前後端應用**: 整合前後端應用，實作會員認證與整合外部 API(登入/付款/物流...)

Next vs Vite vs CRA

種類	Next	Vite	CRA
說明	以 Node 為基礎, React 的 SSG/SSR 框架	高速的前端建置工具	React 官方出品的專案樣版
打包編譯速度	極快(Turpo/swc)	極快(預設 esbuild , 可擴充)	慢(Webpack/babel)
針對應用	CRA(SPA)/SSG/SSR	CSR(SPA) , 可擴充 SSR/SSG	CSR(SPA)
實例客戶	全球各大公司	主要在中國	適合入門