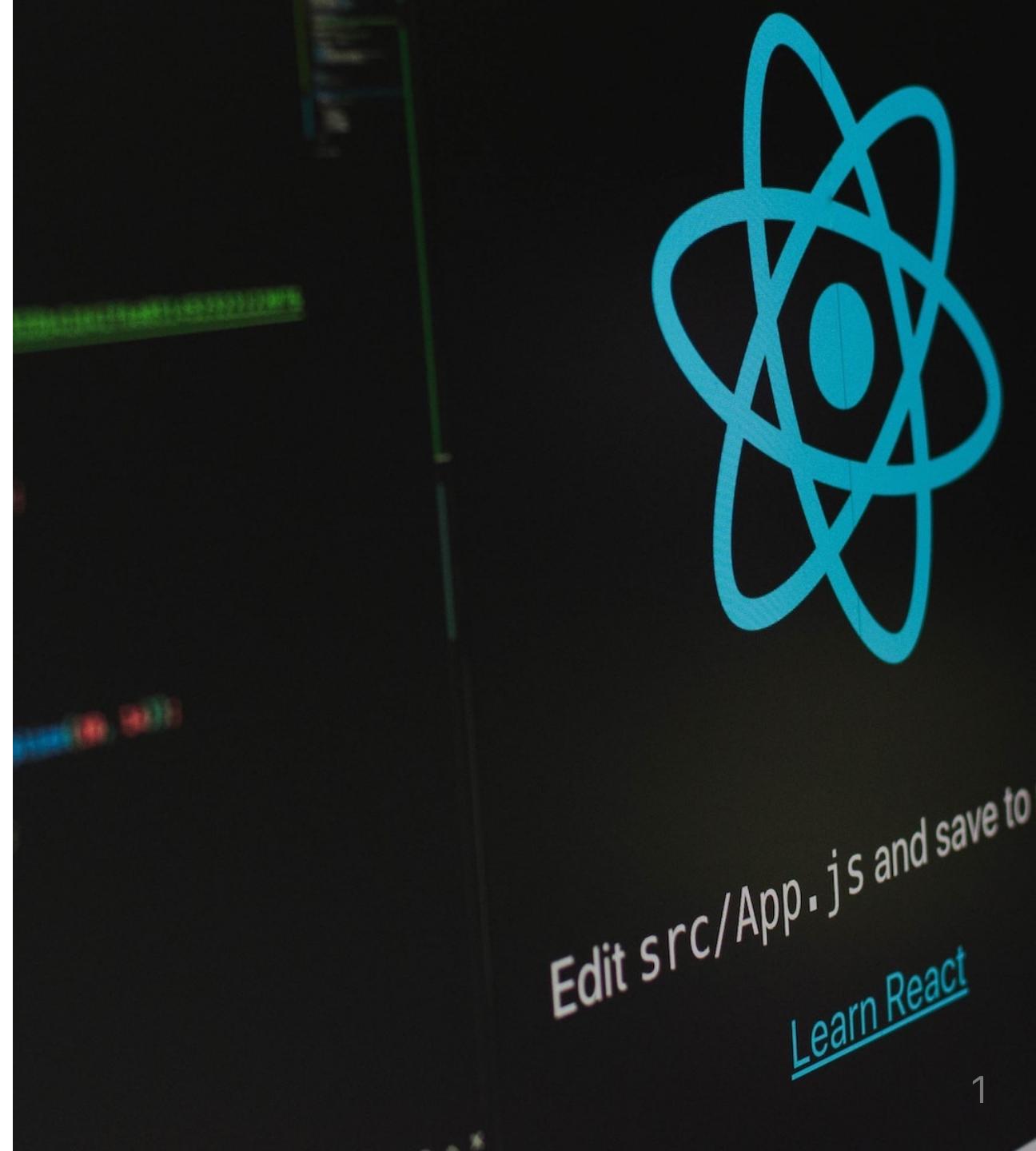


React 介紹

Eddy Chang

✉ hello@eddychang.me



React 是什麼 (what)

- |  用於網站或原生使用者操作介面的函式庫 ~ react.dev
- |  用來實作使用者操作介面的 JavaScript 函式庫 ~ reactjs.org

React 是什麼 (what is)

1. 由 Meta(Facebook) 創造與維護
2. 最早對外發佈於 2013 年，目前為 v18(2022/3)
3. React 通常指的是瀏覽器端的網頁應用程式以 CSR 與 SPA 技術來開發
4. 可延伸至伺服器端的 SSR/SSG 技術(Next)
5. 可延伸至手機 App 開發(React Native)
6. 開放原始碼授權(MIT)

React 不是完整的程式框架(Framework)，相當於軟體 MVC 設計模式的 View(視圖)

React Ecosystem(生態圈)

SSR/SSG: Next, Remix

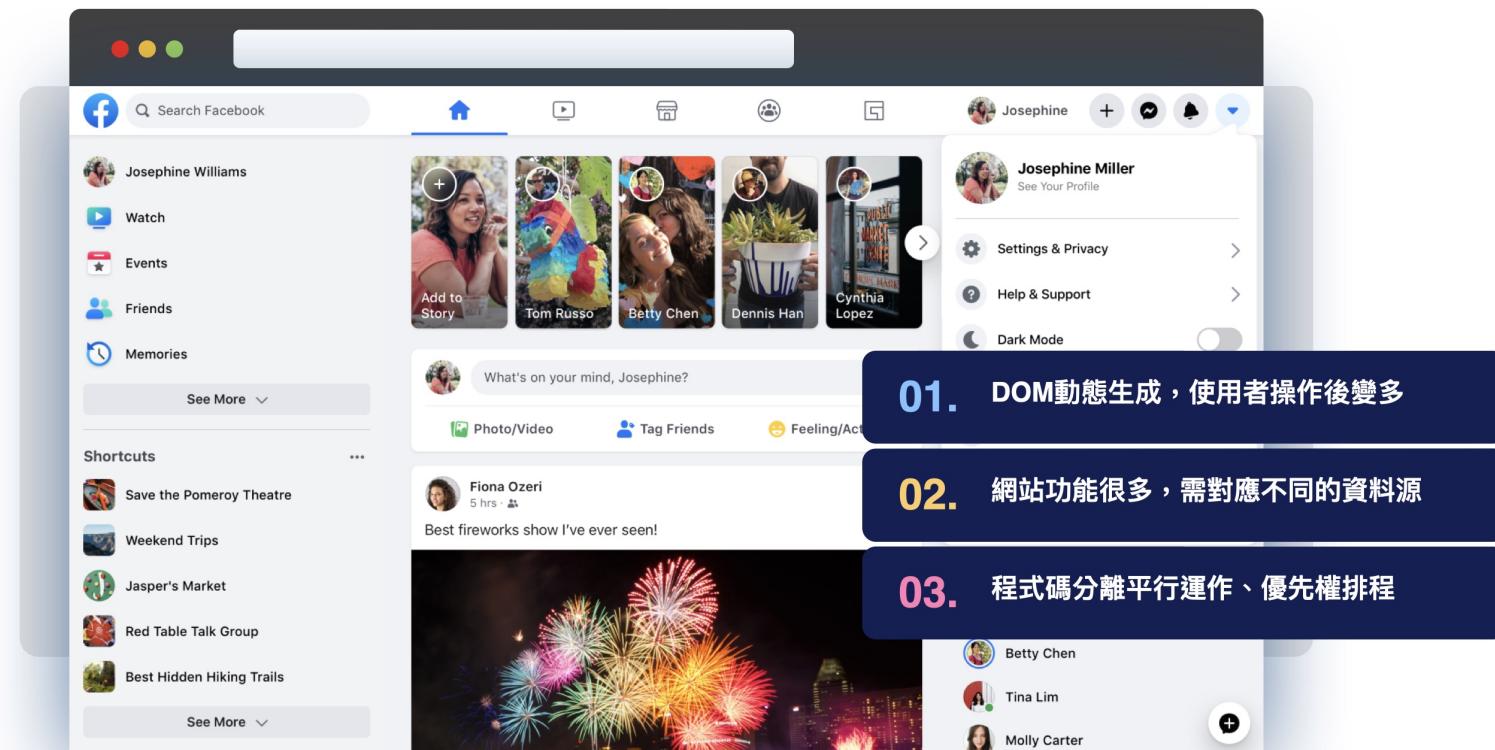
State Management(狀態管理): Redux Toolkit, Zustand

Styling(樣式): Styled Components, Emotion, CSS Modules, Tailwind(整合),

UI libraries(UI 庫): MUI, Ant Design, Chakra UI, React Bootstrap(整合)

Mobile(行動裝置): React Native

Facebook 為什麼創造 React ?



實例客戶(全球)



NETFLIX

yahoo!



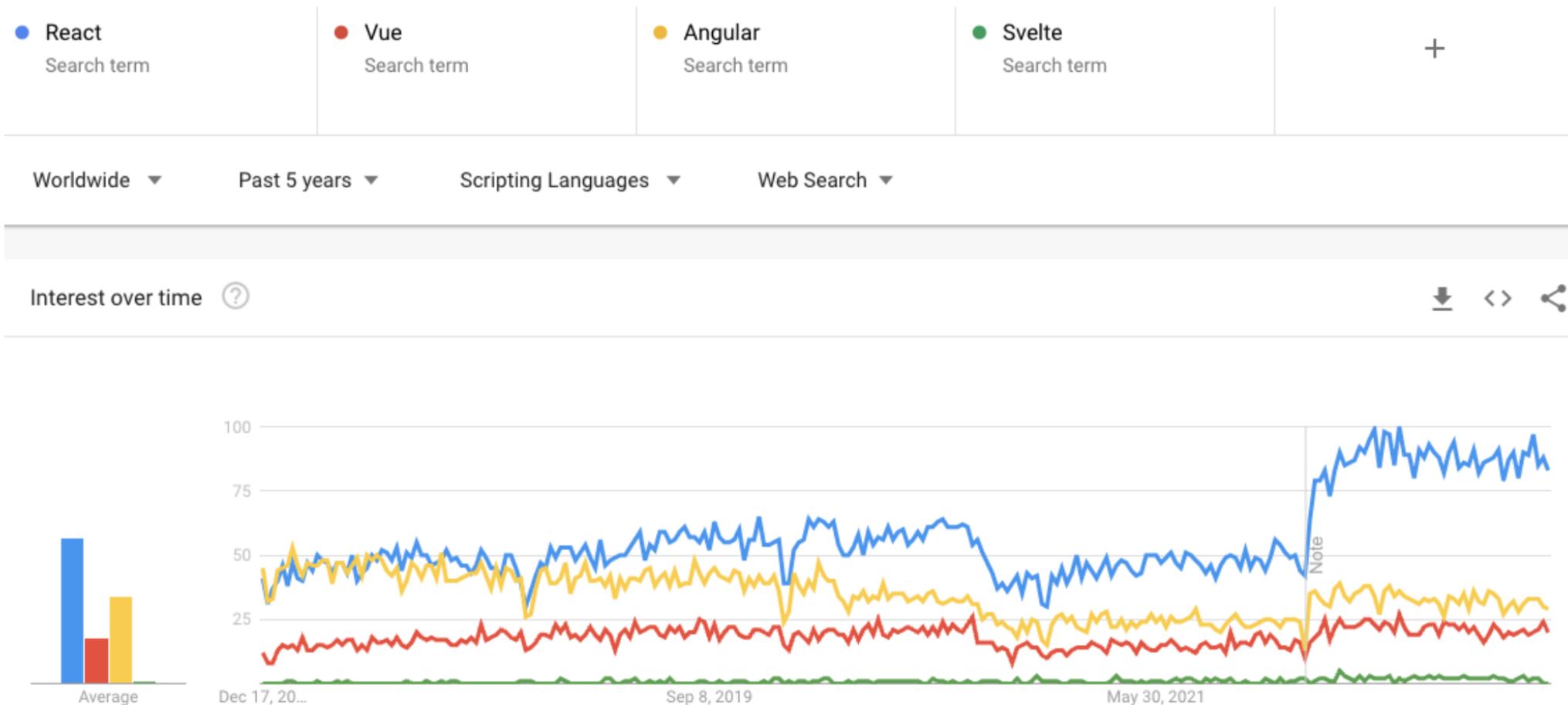
twitch



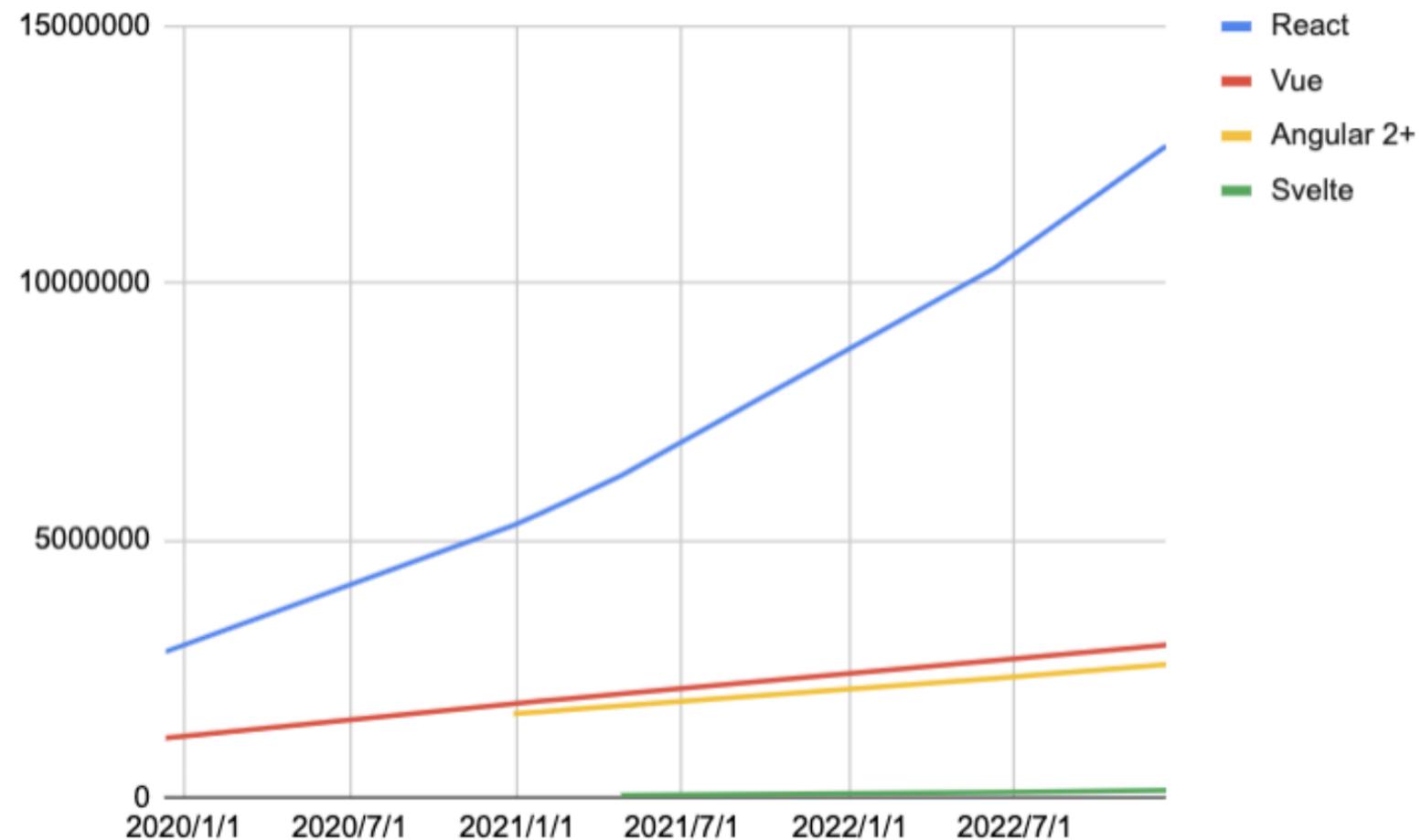
實例客戶(台灣/亞洲)



趨勢 - Google Trends

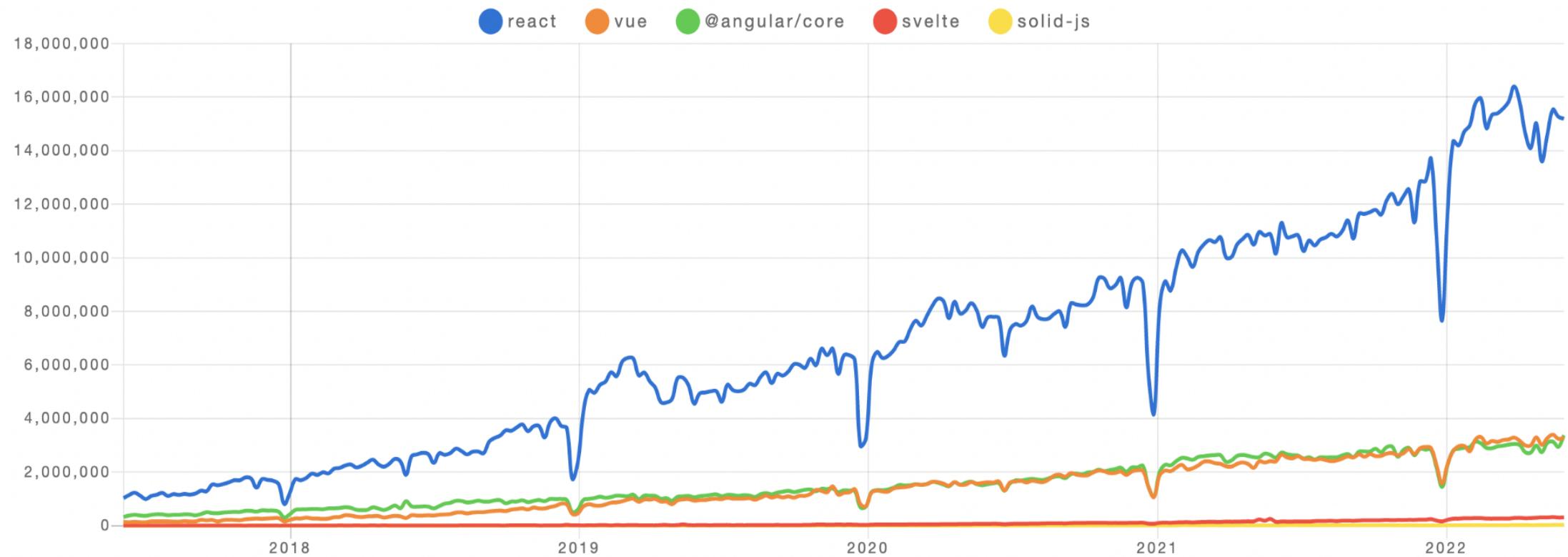


趨勢 - Github repos that depend on

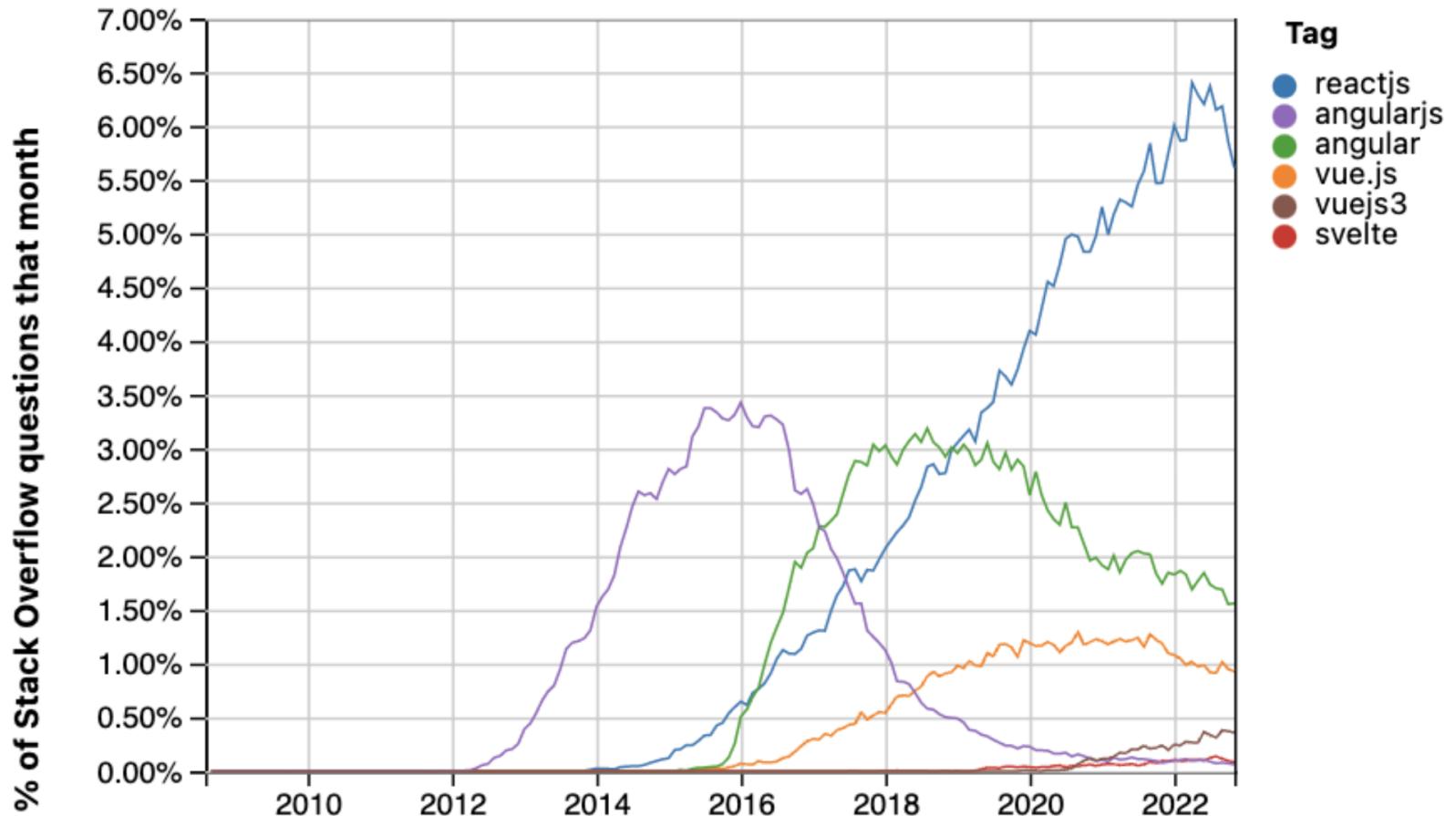


趨勢 - npm downloads

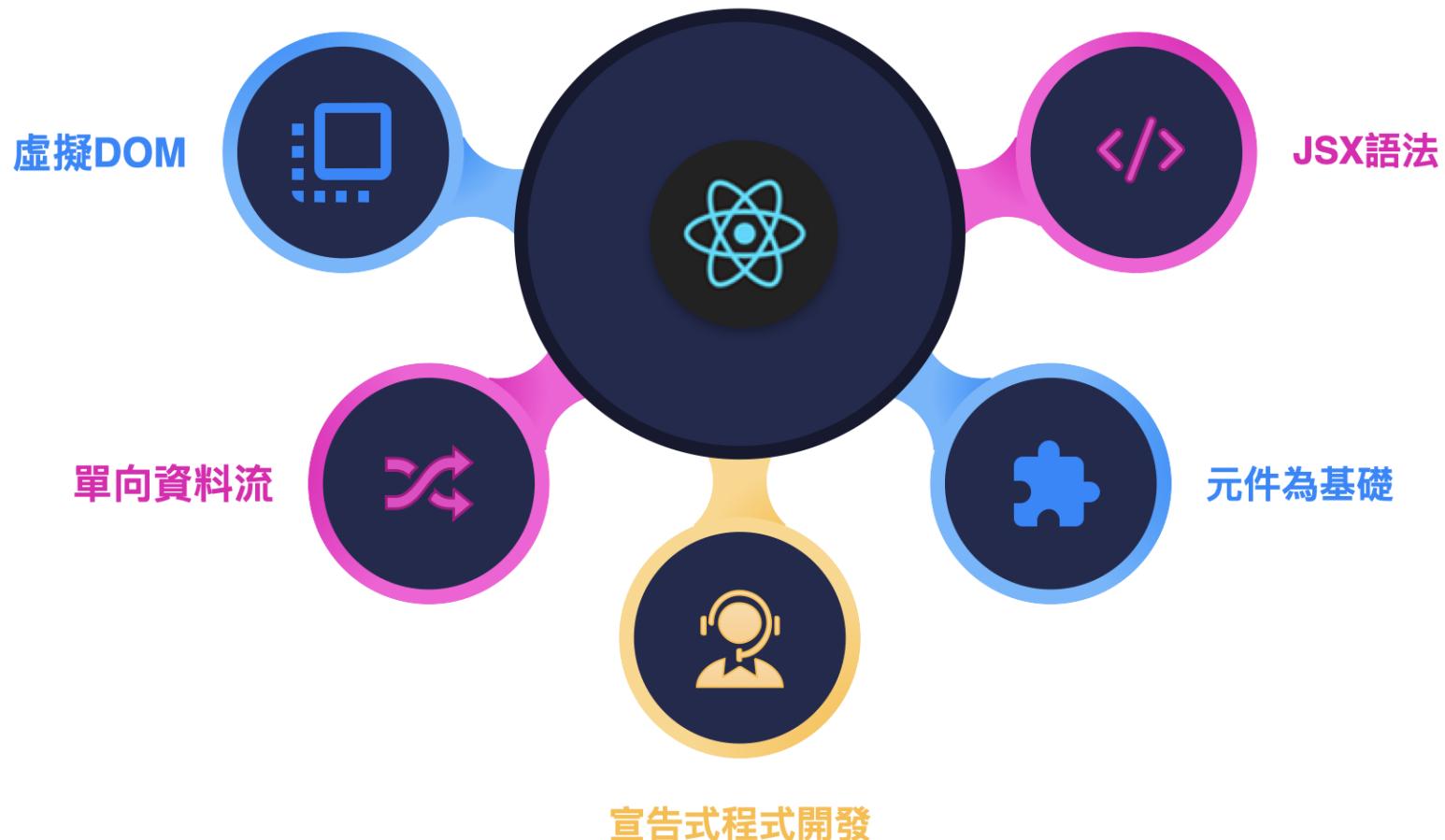
Downloads in past 5 Years ▾



趨勢 - Stack Overflow trends

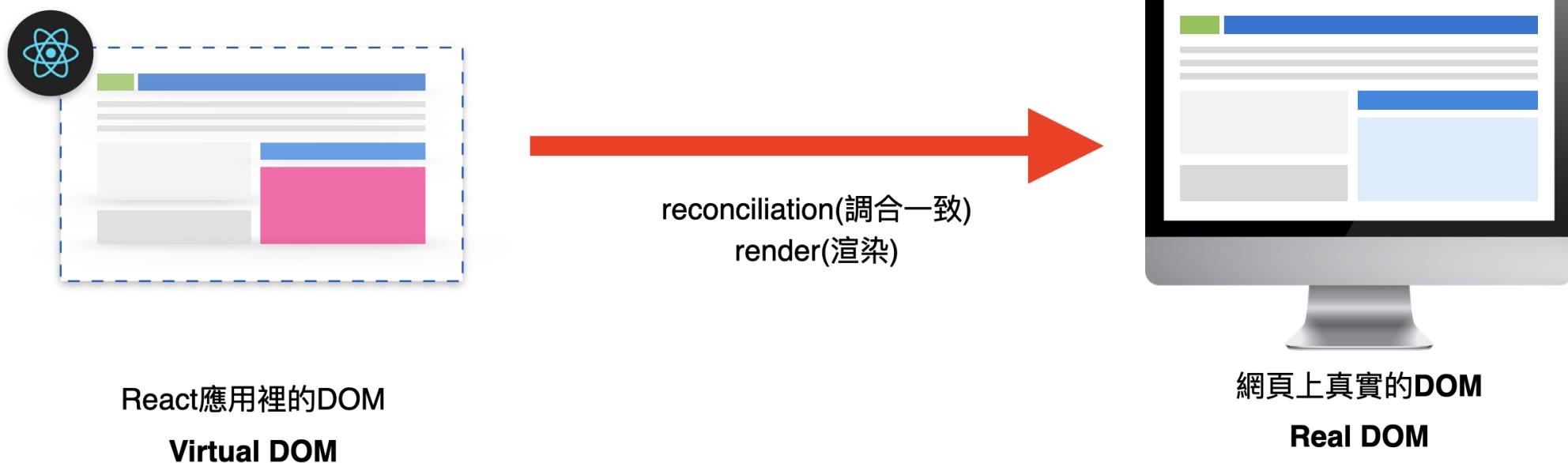


React 五大特性



Virtual DOM(虛擬 DOM)

React 元件中自行管理的 DOM 結構，用於差異比較後再與真實 DOM 作渲染。



JSX

JS 是 JavaScript，X 是 XML。是一種針對 JavaScript 語言語法的擴充，在 JavaScript 程式碼中嵌入類似 HTML 的標記，用來建立虛擬 DOM 的語法。

```
}).then(response => {
  setProgress('finished');
});
```

```
turn (
  <div className={`progress-button ${progress}`}>
    <span className="loading-text">Loading...
      <button className="download-button">
        <span className="button-text">Download</span>
      </button>
      <span className="percentage">${percentage}</span>
    </span>
  </div>
)
```

default App;

以"元件"為基礎

以類別或函式，加入 "狀態
(state)" 與 "屬性(props)" 特性來開發元件，拆分可重覆利用的功能和介面為一個個元件，然後再組合它們。



單向資料流

元件間依靠屬性(props)特性來傳遞資料，只能由父母元件傳遞到子女元件，優點為在複雜的結構中容易除錯與最佳化。



"宣告式"程式開發

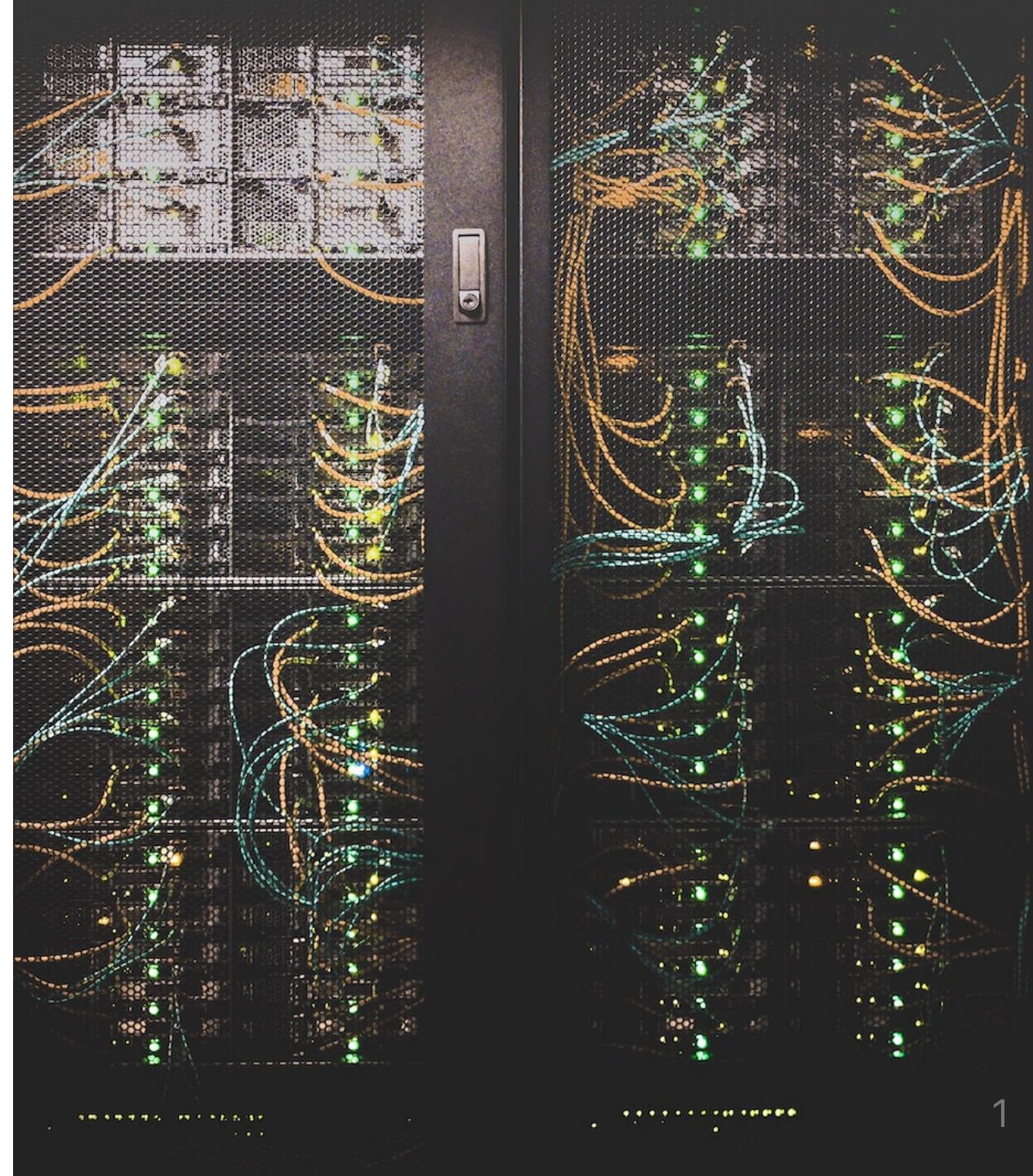
使用類似 HTML 語法來撰寫元件，
DOM 的事件與處理全權交由 React。
React 會協助開發者建立容易維護與
除錯的應用程式，而且足夠快速的進
行 DOM 處理。



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)

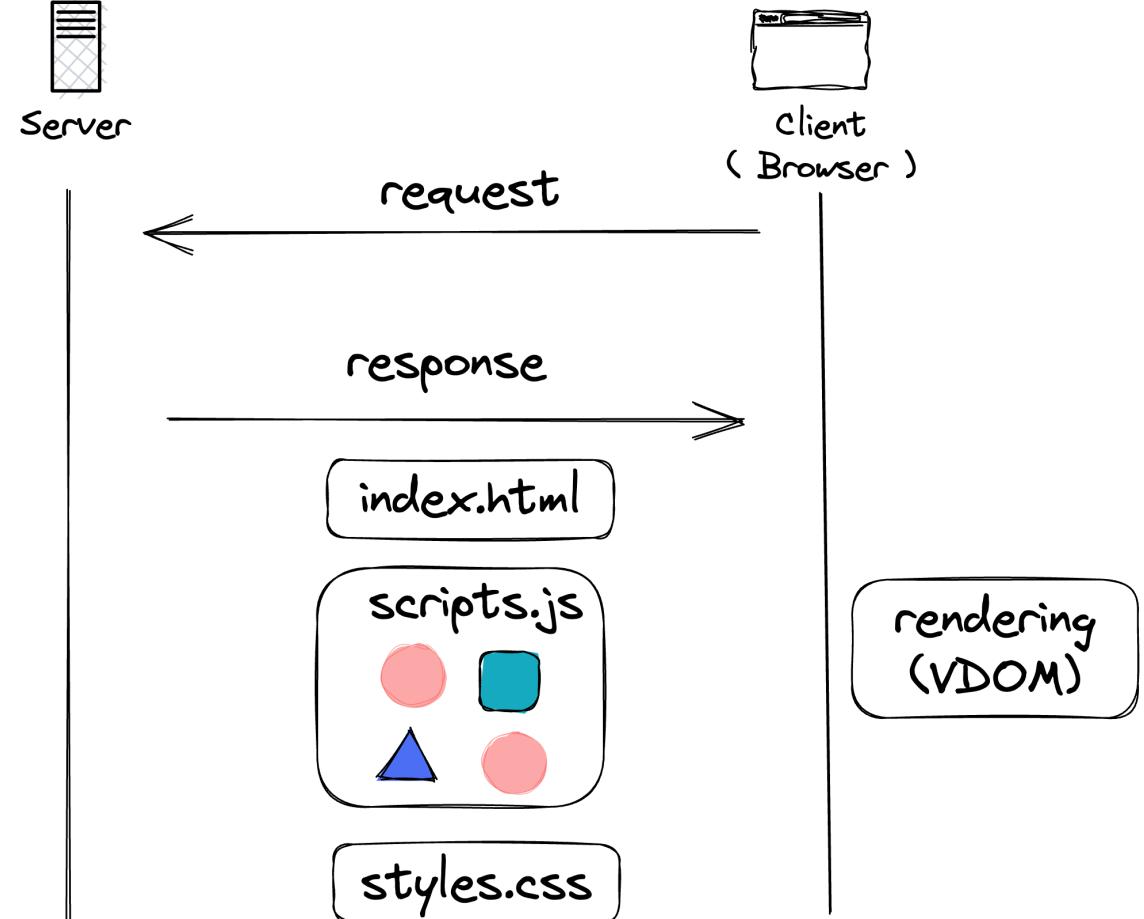
2016 年開始發展，目前由 Vercel 公司主導開發維護

Netflix、GitHub、Uber、微軟、星巴克、任天堂等為實例客戶

CSR

Client Side Render

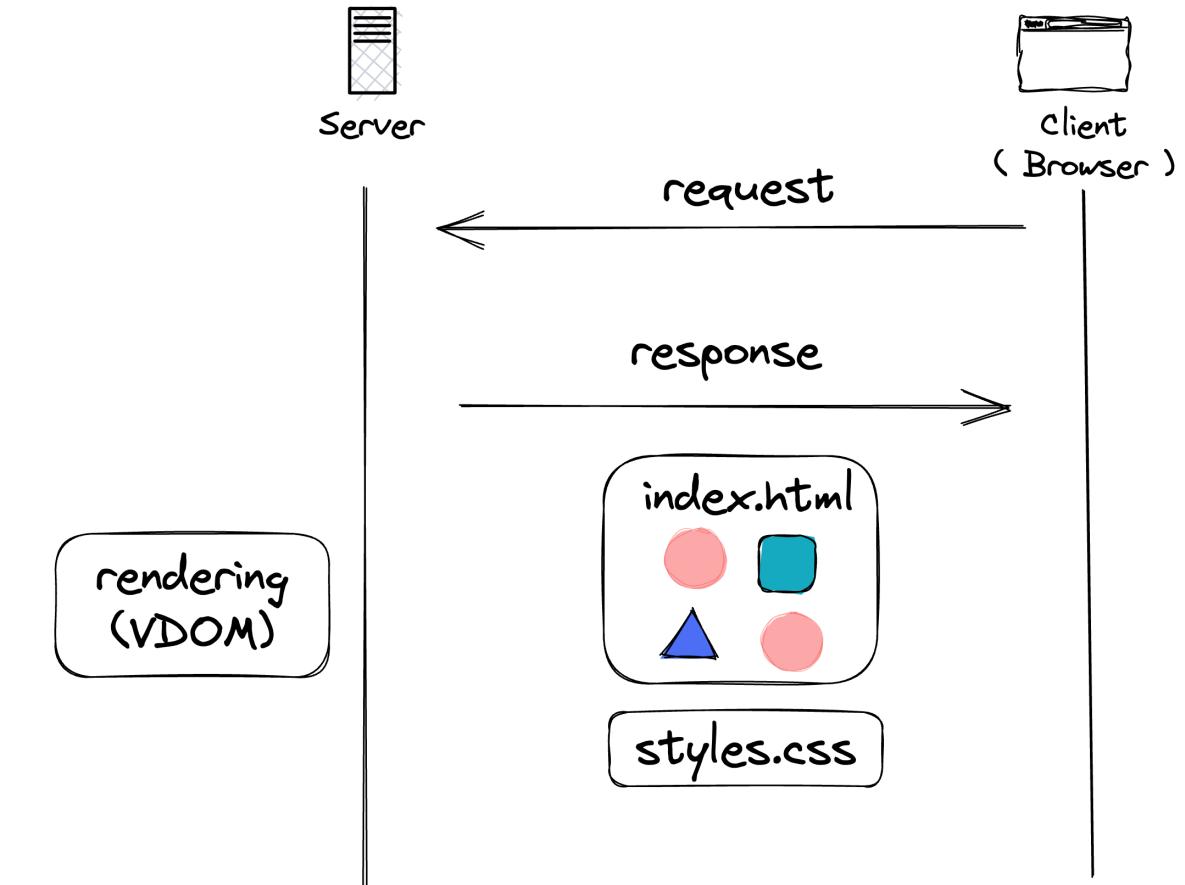
網頁上的所見 HTML 內容，是在客戶端(瀏覽器端)執行後進行渲染再提供它們，一開始伺服器會回應一個空白內容的 HTML 文件。JS 框架如 React, Vue, Angular 的 SPA(單頁式應用)指的就是以 CSR 的方式進行。



SSR

Server Side Render

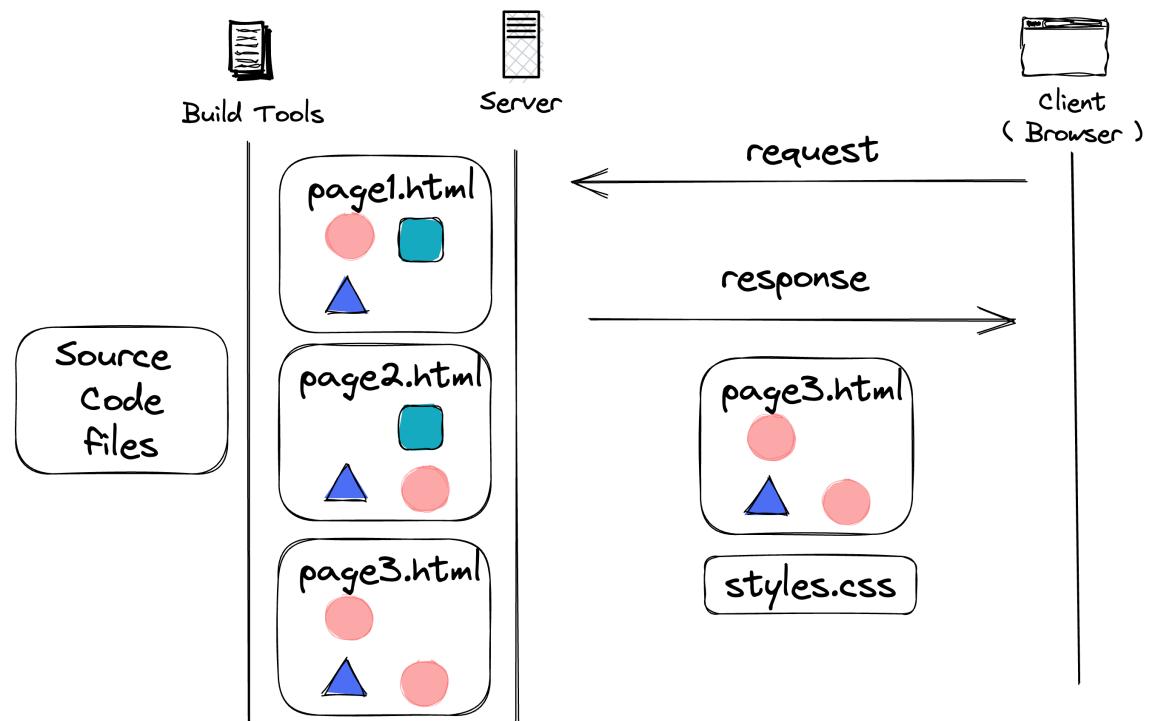
網頁上的所見 HTML 內容，伺服器每次收到客戶端要求後，才在伺服器端進行渲染產生頁面，再回應給客戶端。與 SSG 的最大不同處，在於它雖然是一種預渲染 (pre-render) 機制，但需要對每次要求伺服器都要進行渲染的過程。



SSG

Server Side Generation

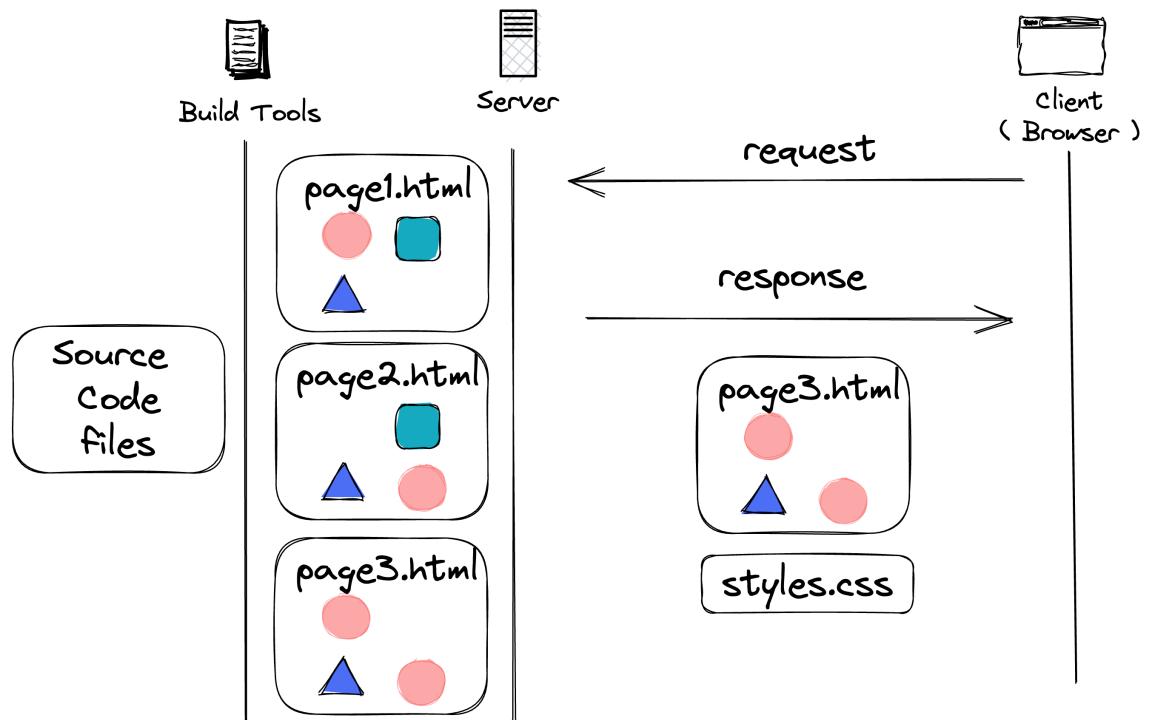
網頁上的所見 HTML 內容，是由伺服器**預先**在伺服器端渲染建立產生 HTML 文件，再視要求回應給客戶端。



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証過期的頁面，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

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

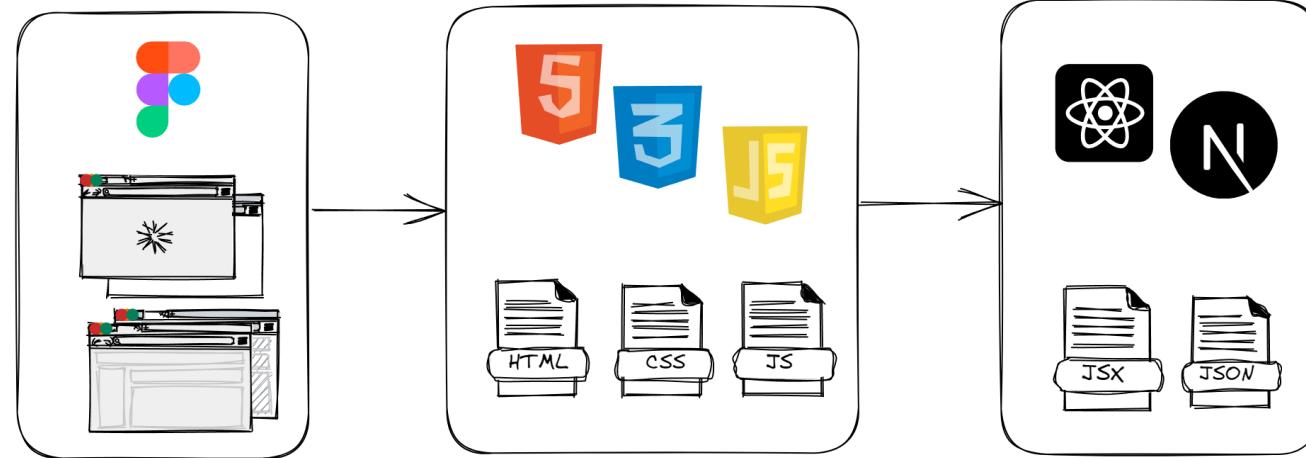
TypeScript 自動支援

多樣的資料 Fetch 技術: SSG/SSR

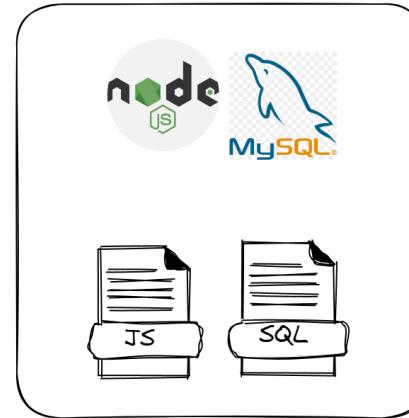
檔案系統的路由

實作說明(圖解)

Frontend



Backend



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 + 會員認証...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，實作會員認証與整合外部 API(登入/付款/物流...)