

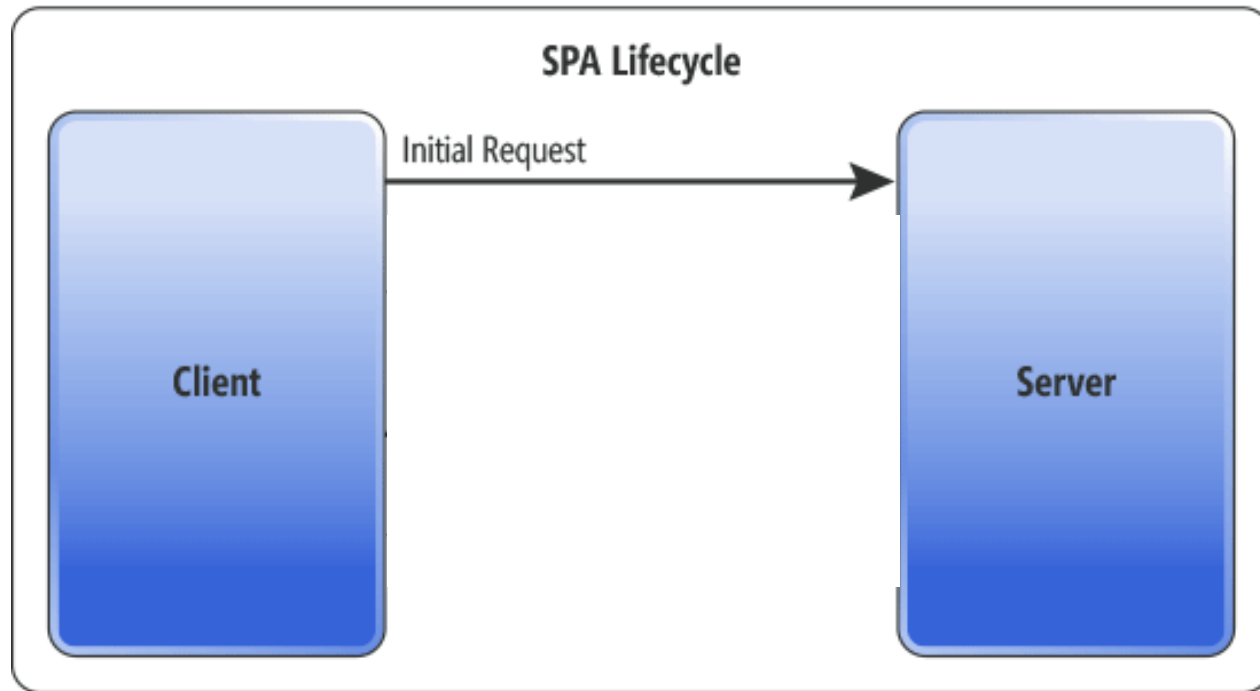
Server-Side Rendering (SSR)

<https://ua.vuejs.org/guide/scaling-up/ssr.html>

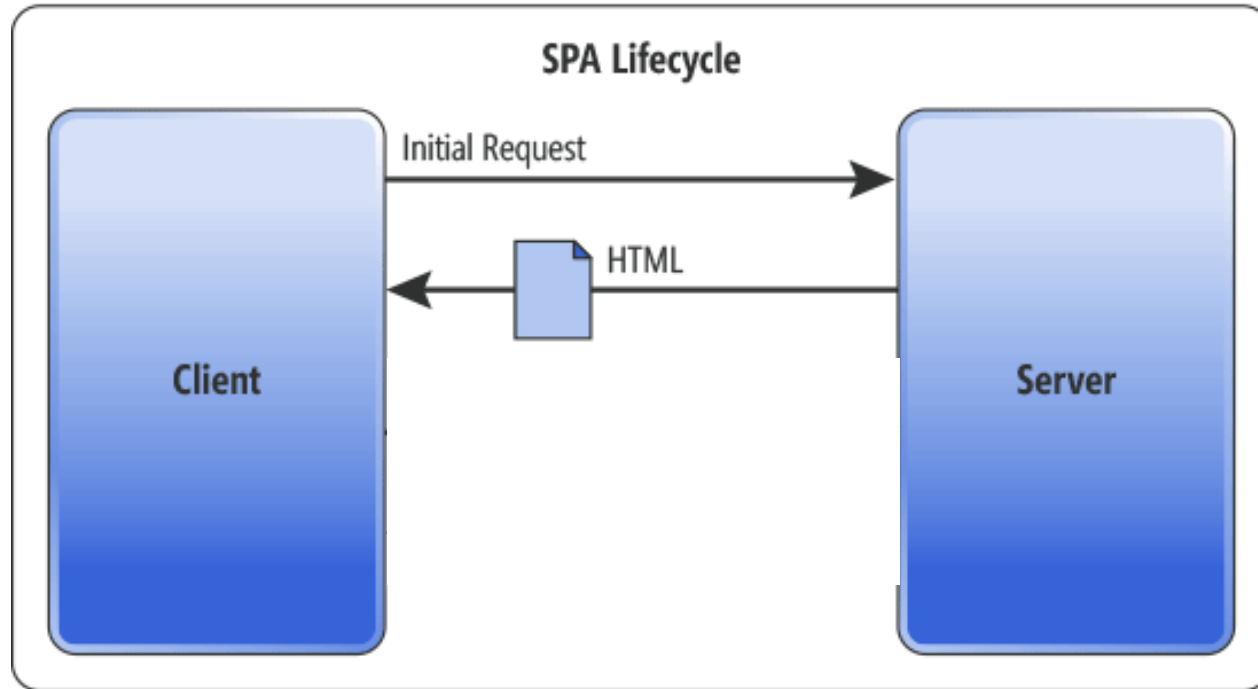
<https://vitejs.dev/guide/ssr>

<https://github.com/bluwy/create-vite-extra/tree/master/template-ssr-vue>

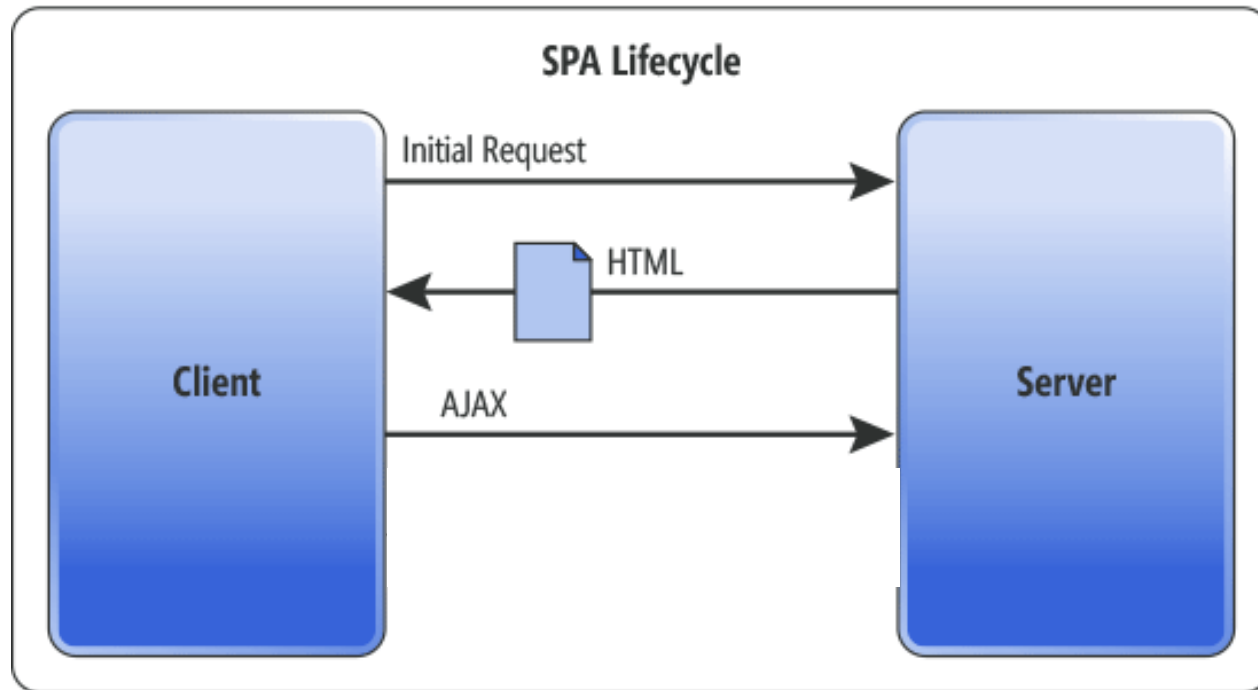
Життєвий цикл SPA додатку



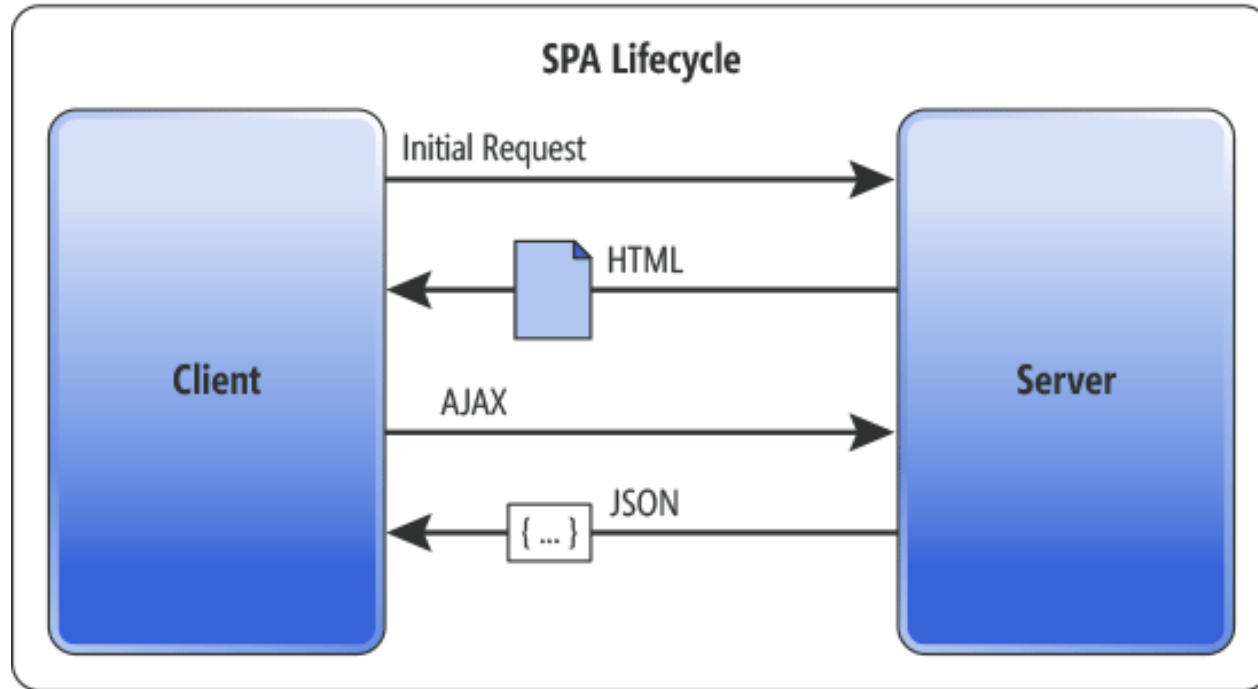
Життєвий цикл SPA додатку



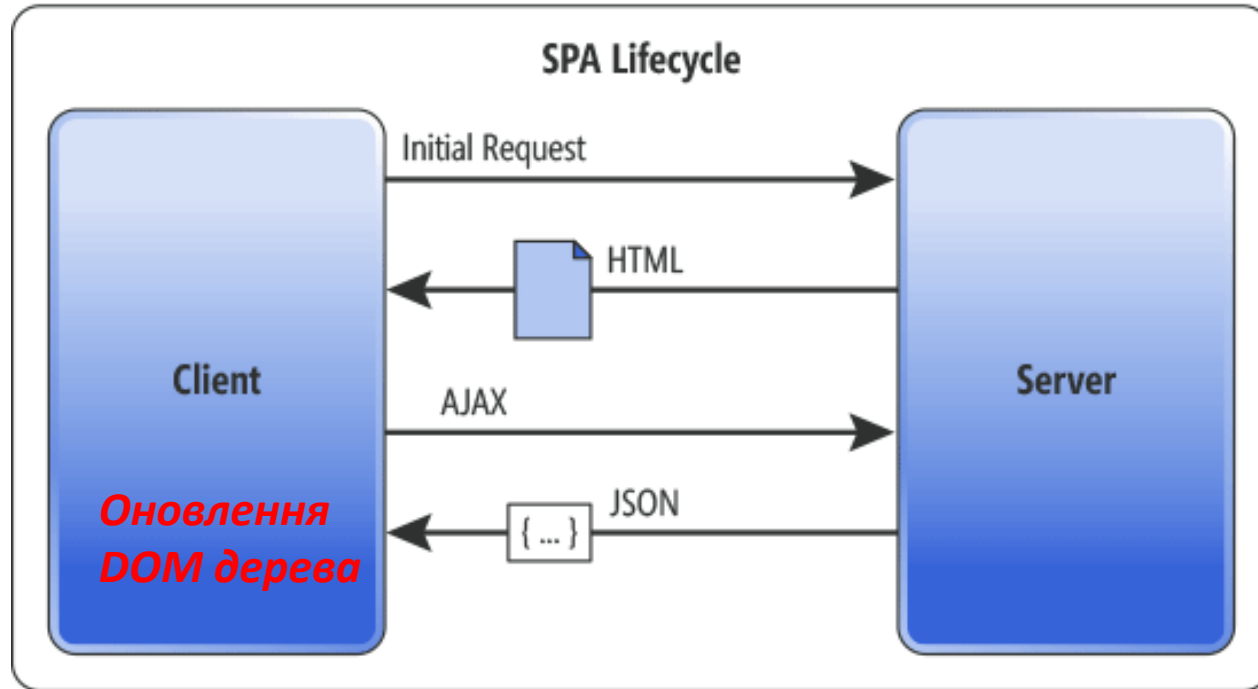
Життєвий цикл SPA додатку



Життєвий цикл SPA додатку



Життєвий цикл SPA додатку



SSR ()

- дозволяє зробити рендер компонентів в рядки HTML на сервері, надіслати їх безпосередньо в браузер
- "гідратувати" статичну розмітку в повністю інтерактивний застосунок на клієнті

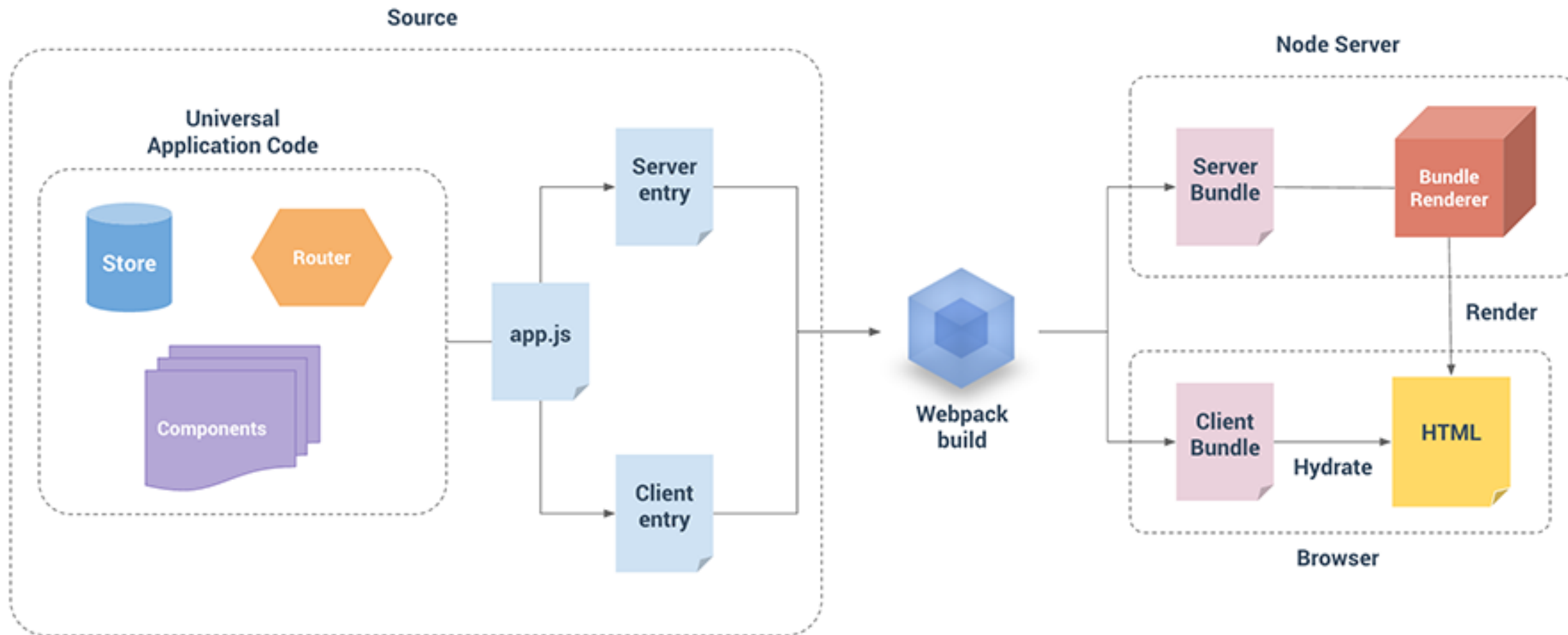
Переваги:

- **Швидший час до готового вмісту:** це більш помітно при повільному з'єднанні або повільних пристроях. Для розмітки, яка рендериться на стороні серверу, не потрібно чекати, поки весь JavaScript буде завантажено та виконано, тому ваш користувач швидше побачить повністю готову сторінку. Крім того, вибірка даних виконується на стороні сервера для першого відвідування, яке, ймовірно, має швидше з'єднання з вашою базою даних, ніж клієнт. Загалом це призводить до покращення показників [Core Web Vitals](#), покращення взаємодії з користувачем і може бути критичним для застосунків, де час до вмісту безпосередньо пов'язаний із коефіцієнтом конверсії.
- **Уніфікована ментальна модель:** ви можете використовувати ту саму мову та ту саму декларативну, компонентно-орієнтовану ментальну модель для розробки всього застосунку, замість того, щоб стрибати між системою шаблонів серверного фреймворку і клієнтським фреймворком.
- **Краща SEO:** сканери пошукової системи безпосередньо побачать повністю рендерину сторінку.

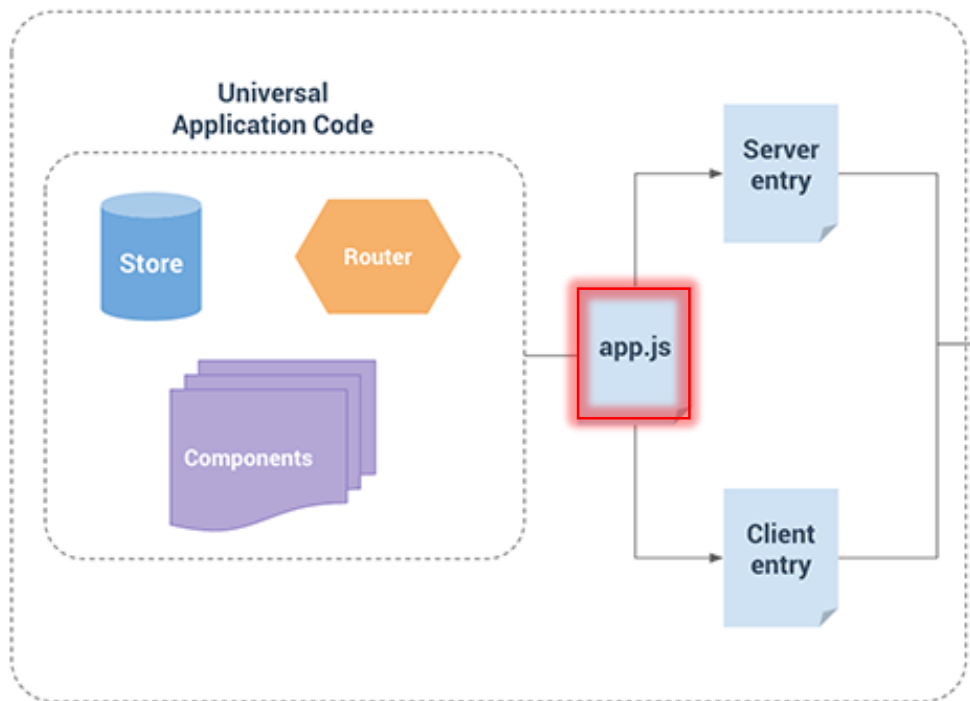
Недоліки:

- **Обмеження розробки.** Специфічний код браузера, можна використовувати лише в межах певних хуків життєвого циклу; деяким зовнішнім бібліотекам може знадобитися спеціальна обробка, щоб їх можна було запускати в застосунку, що рендериться на сервері.
- **Більш складні вимоги до налаштування збірки та розгортання.** На відміну від повністю статичного SPA, який можна розгорнути на будь-якому статичному файловому сервері, застосунок, що рендериться на сервері, потребує середовища, де може працювати сервер Node.js.
- **Більше навантаження на сервер.** Рендеринг повного застосунку в Node.js буде більш інтенсивним для процесора, ніж просто обслуговування статичних файлів, тому, якщо ви очікуєте високий трафік, будьте готові до відповідного навантаження на сервер і мудро використовуйте стратегії кешування

Загальна схема



Source

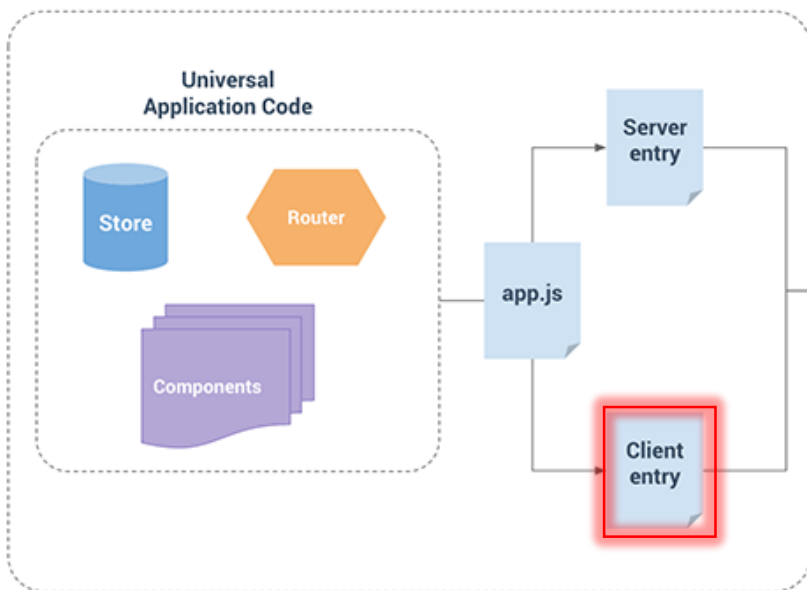


JS app.js src

```
src > JS app.js > ...  
1  import { createSSRApp } from 'vue'  
2  import App from './App.vue'  
3  
4  export function createApp() {  
5    const app = createSSRApp(App)  
6    return { app }  
7  }  
8
```

createSSRApp - це функція, яка дозволяє створювати інстанцію додатку Vue 3 з підтримкою серверного рендерингу (SSR). Вона повертає об'єкт додатку, який можна використовувати для налаштування додатку та його компонентів.

Source

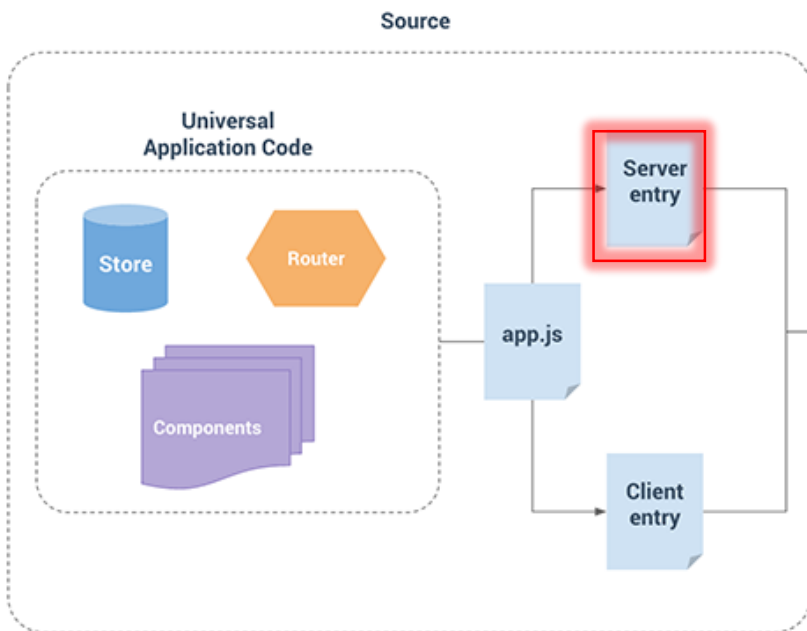


```
JS app.js src ●
src > JS app.js > ...
1  import { createSSRApp } from 'vue'
2  import App from './App.vue'
3
4  export function createApp() {
5    const app = createSSRApp(App)
6    return { app }
7  }
8
```

```
JS entry-client.js src ✕
src > JS entry-client.js > ...
1  import './style.css'
2  import { createApp } from './app'
3
4  const { app } = createApp()
5
6  app.mount('#app')
```

createSSRApp - це функція, яка дозволяє створювати інстанцію додатку Vue 3 з підтримкою серверного рендерингу (SSR). Вона повертає об'єкт додатку, який можна використовувати для налаштування додатку та його компонентів.

renderToString - це функція, яка дозволяє рендерити компонент Vue 3 в HTML-рядок. Вона є частиною пакету `vue/server-renderer` і використовується для створення серверного рендерингу (SSR) додатків на Vue 3.



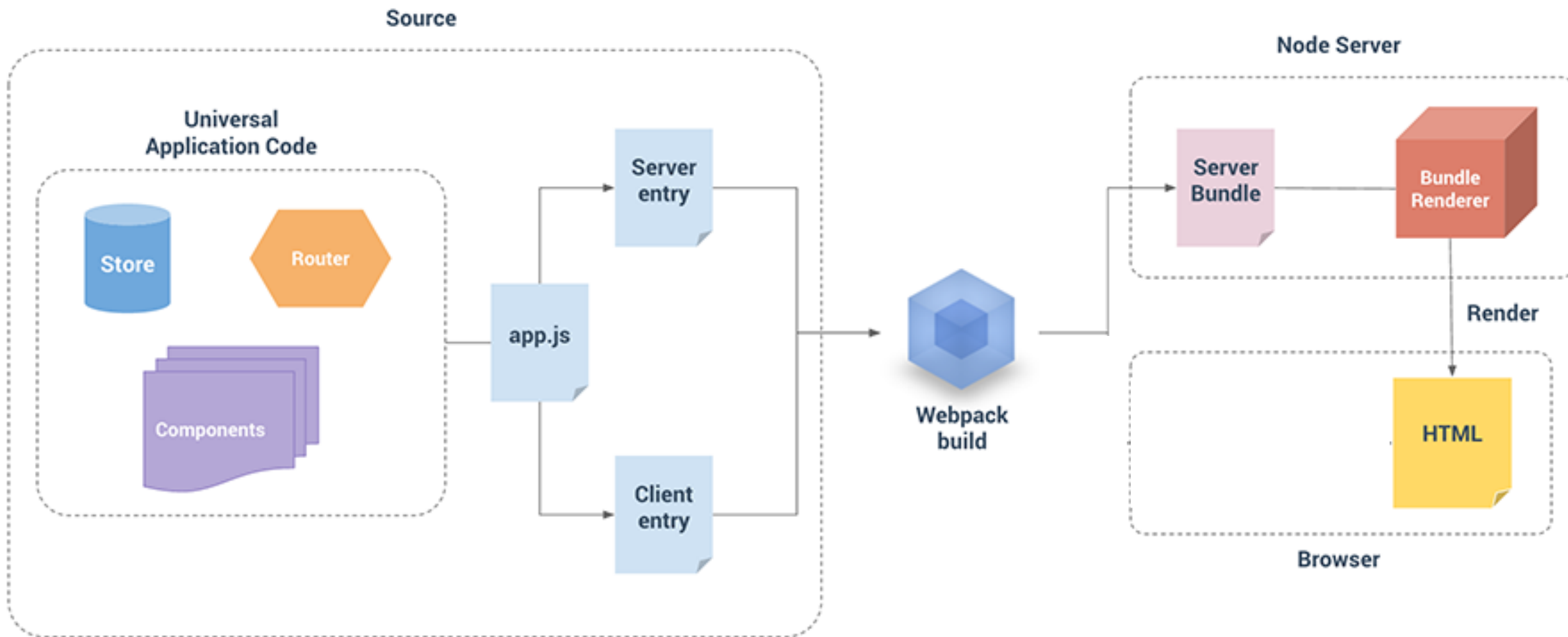
```
JS app.js src
src > JS app.js > ...
1 import { createSSRApp } from 'vue'
2 import App from './App.vue'
3
4 export function createApp() {
5   const app = createSSRApp(App)
6   return { app }
7 }
8
```

```
JS entry-server.js src
src > JS entry-server.js > ...
1 import { renderToString } from 'vue/server-renderer'
2 import { createApp } from './app'
3
4 export async function render() {
5   const { app } = createApp()
6   const ctx = {}
7   const html = await renderToString(app, ctx)
8
9   return { html }
10 }
```

```
JS entry-client.js src X
src > JS entry-client.js > ...
1 import './style.css'
2 import { createApp } from './app'
3
4 const { app } = createApp()
5
6 app.mount('#app')
```

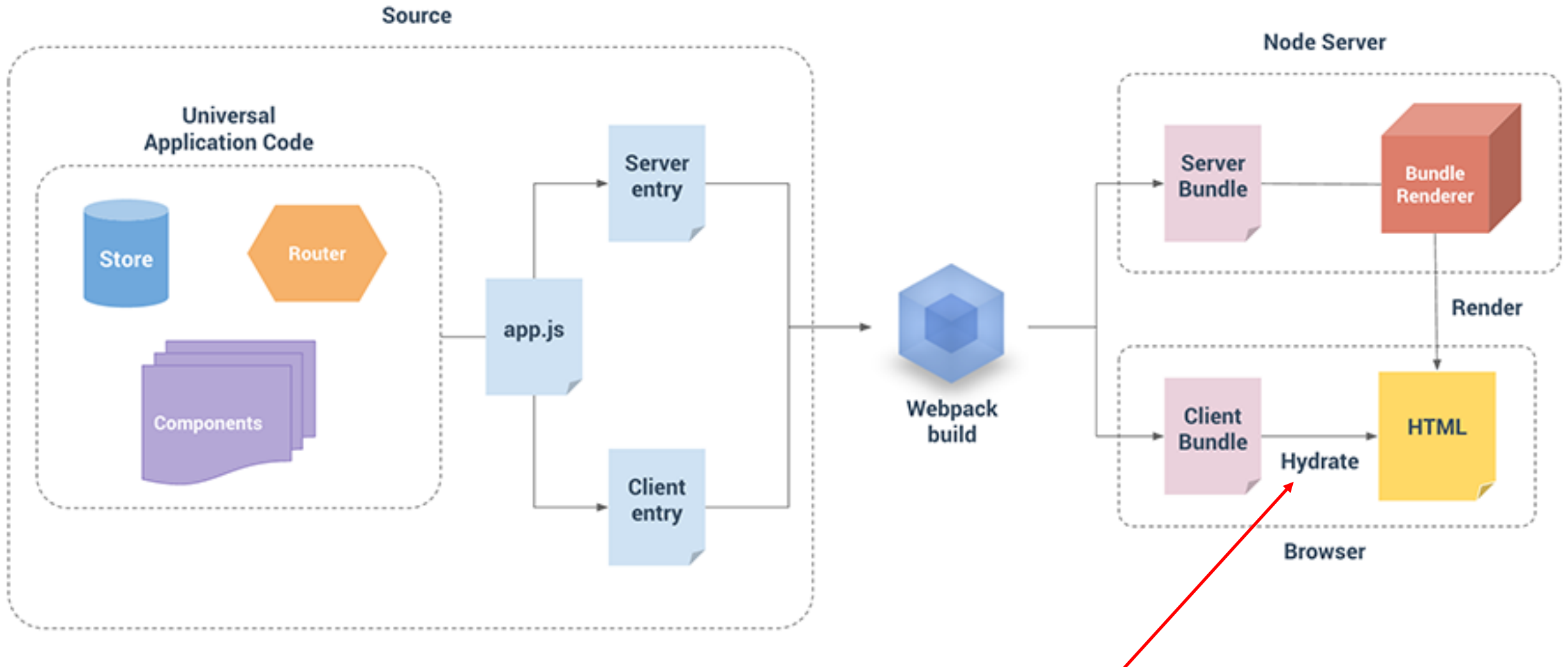
createSSRApp - це функція, яка дозволяє створювати інстанцію додатку Vue 3 з підтримкою серверного рендерингу (SSR). Вона повертає об'єкт додатку, який можна використовувати для налаштування додатку та його компонентів.

Загальна схема



При звертанні до сервера сервер повертає HTML файл, що може бути відображеним у браузері.

Загальна схема



На стороні клієнта для організації інтерактивності виконується гідрація. Гідрація створює той самий застосунок Vue, який було запущено на сервері, зіставляє кожен компонент із вузлами DOM, якими він має керувати, і приєднує слухачі подій DOM.

Загальна схема

