



# Frontends légers avec SolidJs

# Sommaire

---

- 1** Introduction

---
- 2** L'approche classique : 2 Applications

---
- 3** L'approche moderne : Appli. isomorphe

---
- 4** SolidJs : Introduction

---
- 5** SolidJs - Hydratation partielle

---
- 6** SolidJs - Hydratation classique

---
- 7** SolidJs en mode CSR pure

---
- 8** NextJs (Hydratation partielle et classique)

---
- 9** Et après le chargement initial

---
- 10** Conclusion

---
- 11** Annexe : Ressources

---
- 12** Annexe : Autres frameworks intéressants

---

Hello!  
@ SFEIR



[/u/nawfel\\_bgh](#)



[bengherbia.n@sfeir.com](mailto:bengherbia.n@sfeir.com)



Nawfel **Bengherbia**

# Introduction

[sf≡ir]

# Expérience utilisateur

---

- UX
  - [...]
  - { Rapidité du chargement initial }
  - { Rapidité des interactions }

Taille de l'application



# Expérience développeur

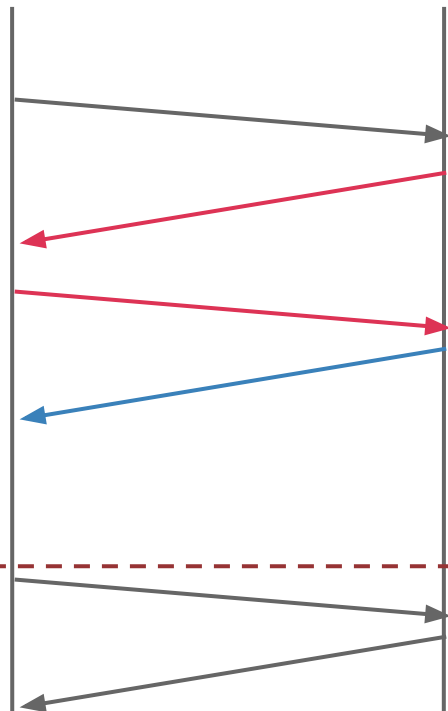
---

- [...]
- Simplicité du code
- Performances par défaut
- Outils de développement et de débogage
  - Coloration syntaxique, typescript
  - Dev tools
- Docs et communauté
- Bibliothèques
  - i18next, material-ui, ag-grid, ...

# Schéma général du chargement d'une page web

Client

Serveur



Première réponse

*Rapide, légère et riche*

Ressources  
(Code + images)

*Taille petite*

Exécution  
initiale

[sf≡ir]



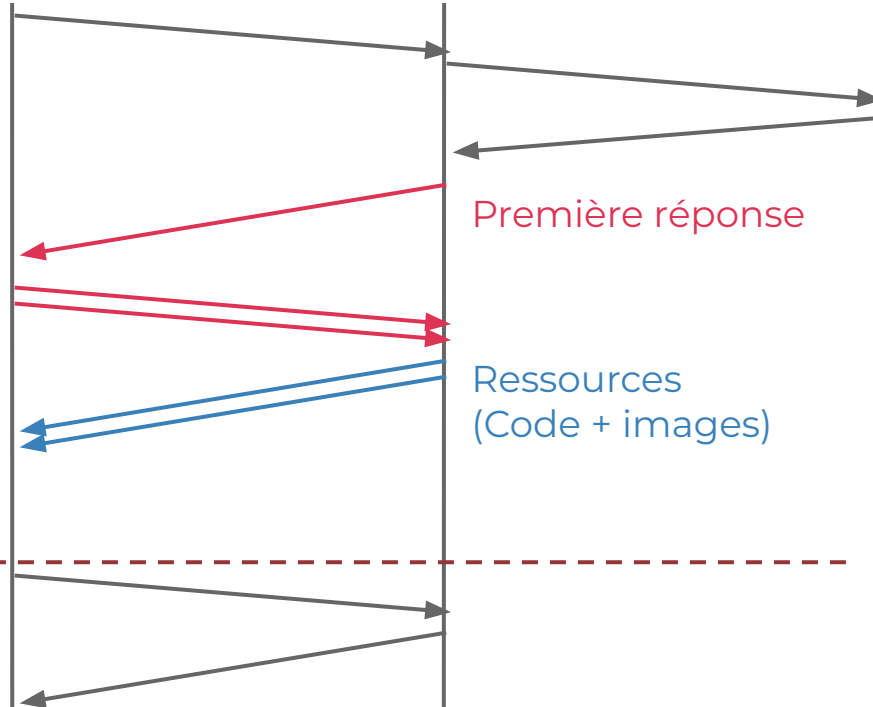


# L'approche : 2 applications



# Amazon.com (SSR + jQuery coté client)

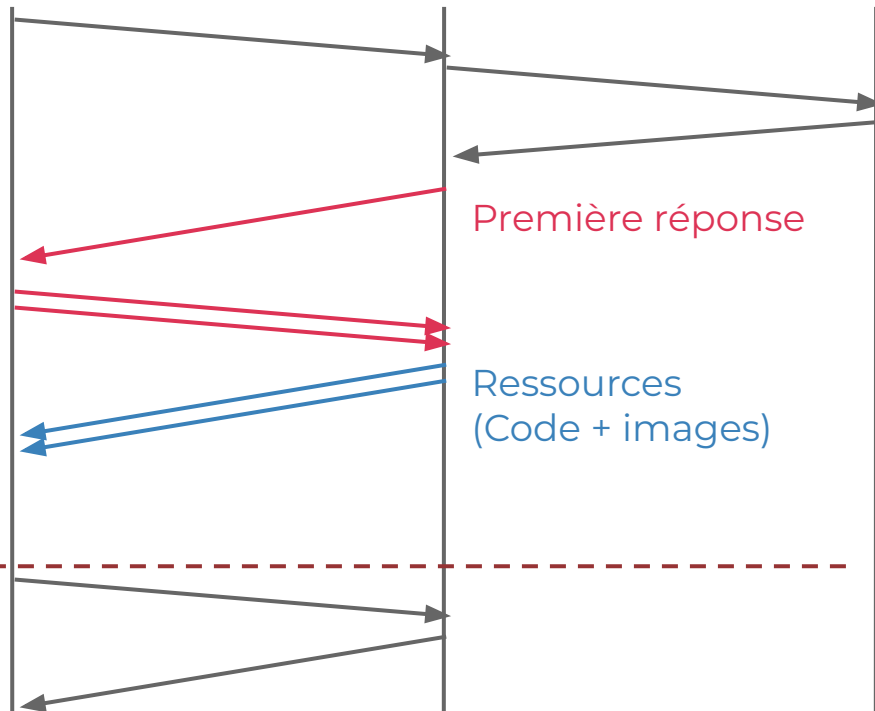
Client                      Back For Front                      Backend



- URL jQuery : **34KB**
- URL Code applicatif
- Rendered HTML
  - URLs images

# Hacker News (SSR + JS vanilla coté client)

Client                      Back For Front                      Backend



- URL Code applicatif: **2.32KB**
- Rendered HTML
  - URLs images

# Pros/Cons de SSR + JS impératif coté client

---

- + Bonne performance
- 2 patterns de développement différents pour le client et le serveur



# Single Page Applications

# Single Page Apps

Client      Serveur de fichiers      Backend

Première réponse

Ressources  
(Code)

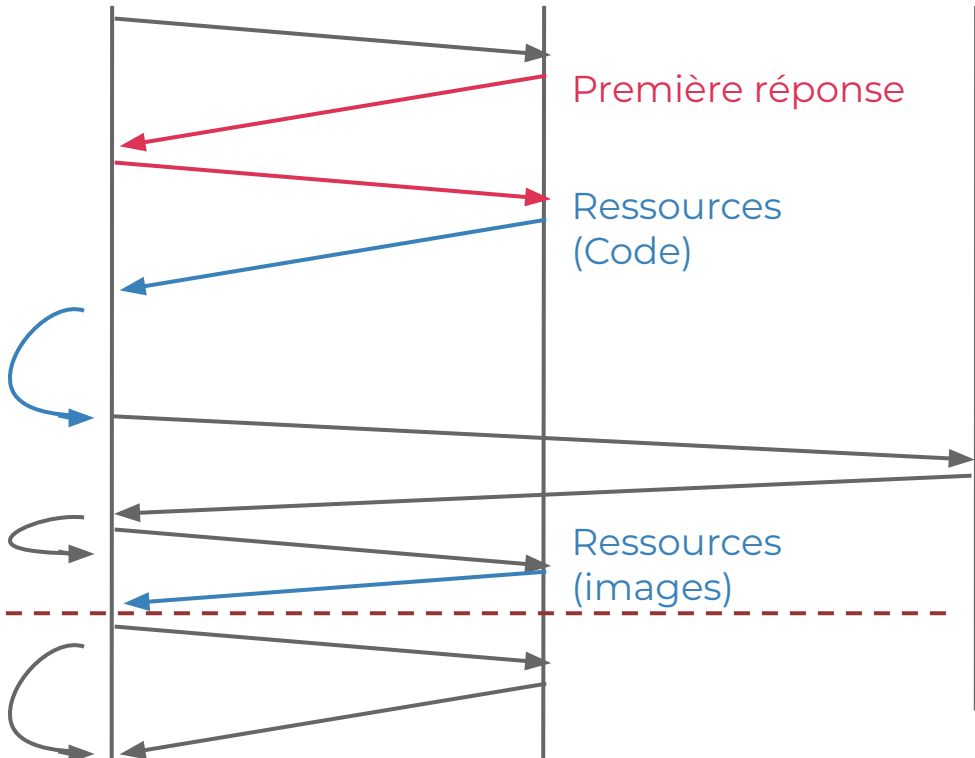
Ressources  
(images)


- URLs bundles Js
  - Framework
  - Code applicatif

Exécution  
initiale : Sans  
contenu

Rendu HTML  
avec contenu

[sf≡ir]

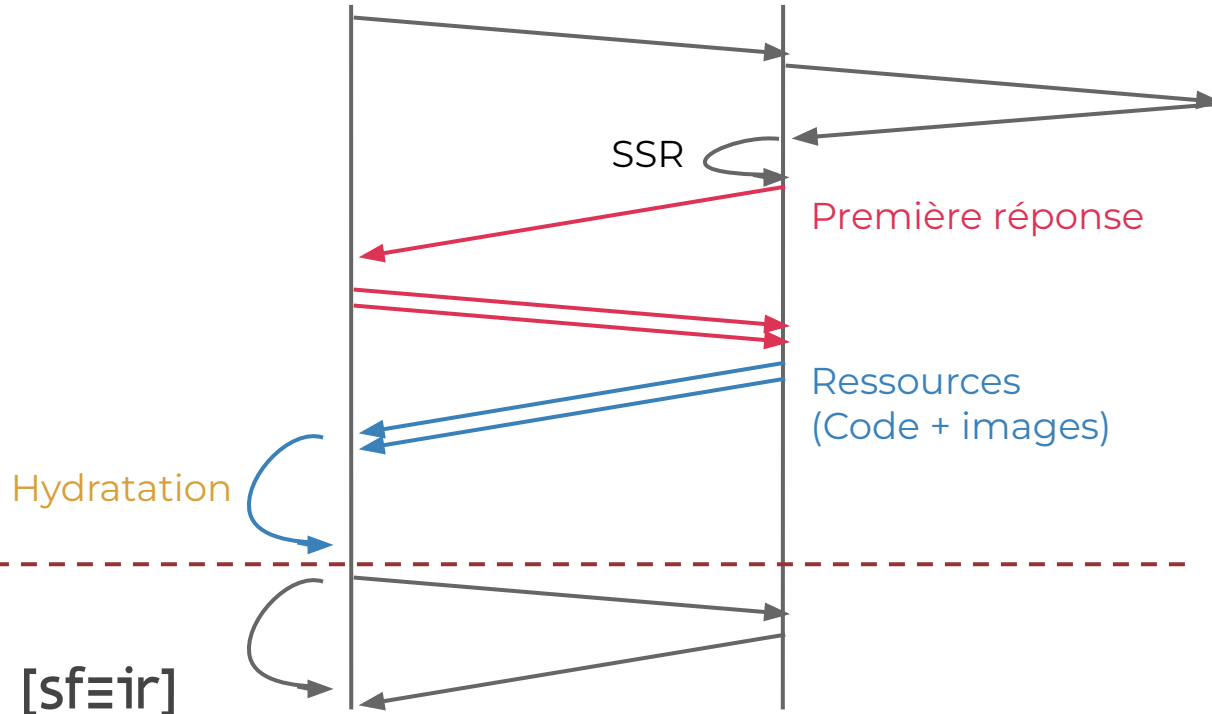




# L'approche : application isomorphe

# Application isomorphe

Client                      Back For Front                      Backend



- URL(s) code
  - Framework
  - Code applicatif
- Rendered HTML
  - URLs images
- Recréer/Désérialiser le state de l'application
- Installer les event handlers





# Introduction à SolidJS

# Solid Js

---

- Open Sourced en 2018 par [@RyanCarniato](#)
- Le framework frontend **le plus rapide** [depuis 2019](#)
- [Version 1.0](#) sortie en juin 2021 (avec support du SSR)
- **SolidStart** (compétiteur de NextJs) toujours en Beta
- Taille : **7.4KB**
- Bibliothèque **Treeshakable**
- Utilise la syntaxe **JSX**, mais **sans VDOM**



main.tsx +

 Display Errors

Result

Output

```
1 import { render } from "solid-js/web";
2 import { createSignal, createEffect } from "solid-js";
3
4 function Counter() {
5   const [count, setCount] = createSignal(1);
6   const increment = () => setCount(count() + 1);
7   createEffect(() => console.log(count()));
8
9   return (
10    <button type="button" onClick={increment}>
11      {count()}
12    </button>
13  );
14 }
15
16 render(() => <Counter />, document.getElementById("app")!);
17
```

3

Console Network Resources Elements Dom Settings

All Error Warning Info

Console was cleared

1

2

3

&gt;

```
main.tsx +  Display Errors
```

```

1 import { render } from "solid-js/web";
2 import { createSignal } from "solid-js";
3
4 function Counter() {
5   const [count, setCount] = createSignal(1);
6   const increment = () => setCount(count() + 1);
7
8   return (
9     <button type="button" onClick={increment}>
10      {count()}
11    </button>
12  );
13 }
14
15 render(() => <Counter />, document.getElementById("app")!);
16

```

```
Result Output
```

```

1 import { template as _$template } from "solid-js/web";
2 import { delegateEvents as _$delegateEvents } from "solid-js/web";
3 import { createComponent as _$createComponent } from "solid-js/web";
4 import { insert as _$insert } from "solid-js/web";
5 const _tmpl$ = /*#__PURE__*/_template(`<button type="button"></button>`, 2);
6 import { render } from "solid-js/web";
7 import { createSignal } from "solid-js";
8 function Counter() {
9   const [count, setCount] = createSignal(1);
10  const increment = () => setCount(count() + 1);
11  return () => {
12    const _el$ = _tmpl$.cloneNode(true);
13    _el$.$click = increment;
14    _$insert(_el$, count);
15    return _el$;
16  }();
17 }
18 render(() => _$createComponent(Counter, {}), document.getElementById("app"));

```

← \_\$insert crée un effet qui traque la valeur de count et met à jour le DOM

**COMPILE MODE**

- Client side rendering
- Server side rendering
- Client side rendering with hydration



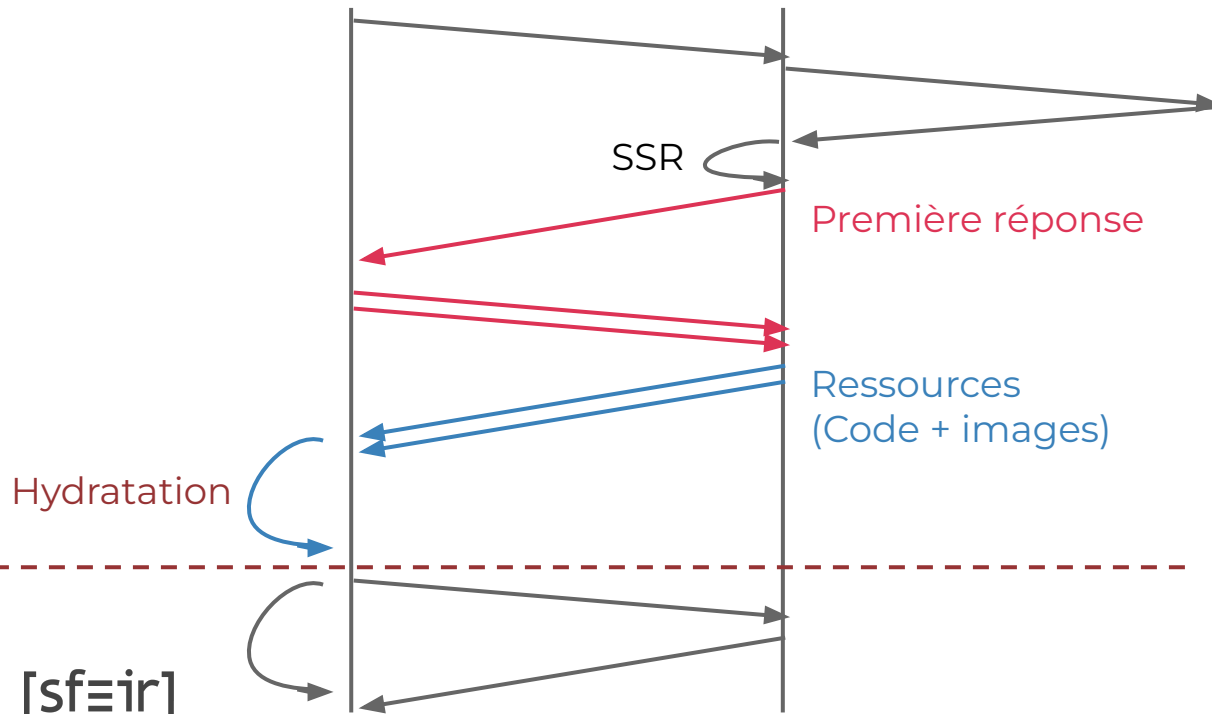


# Hydratation Partielle avec SolidJS

DEPUIS v1.6 (2022-10)

# Hydratation partielle avec Solid Js [\(demo\)](#)

Client                      Back For Front                      Backend



- URLs bundles Js : **<7KB**
  - Framework (Treeshaké)
  - **Composants client**
- Rendered HTML
  - URLs images
  - + **State sérialisé** pour les **composants client**

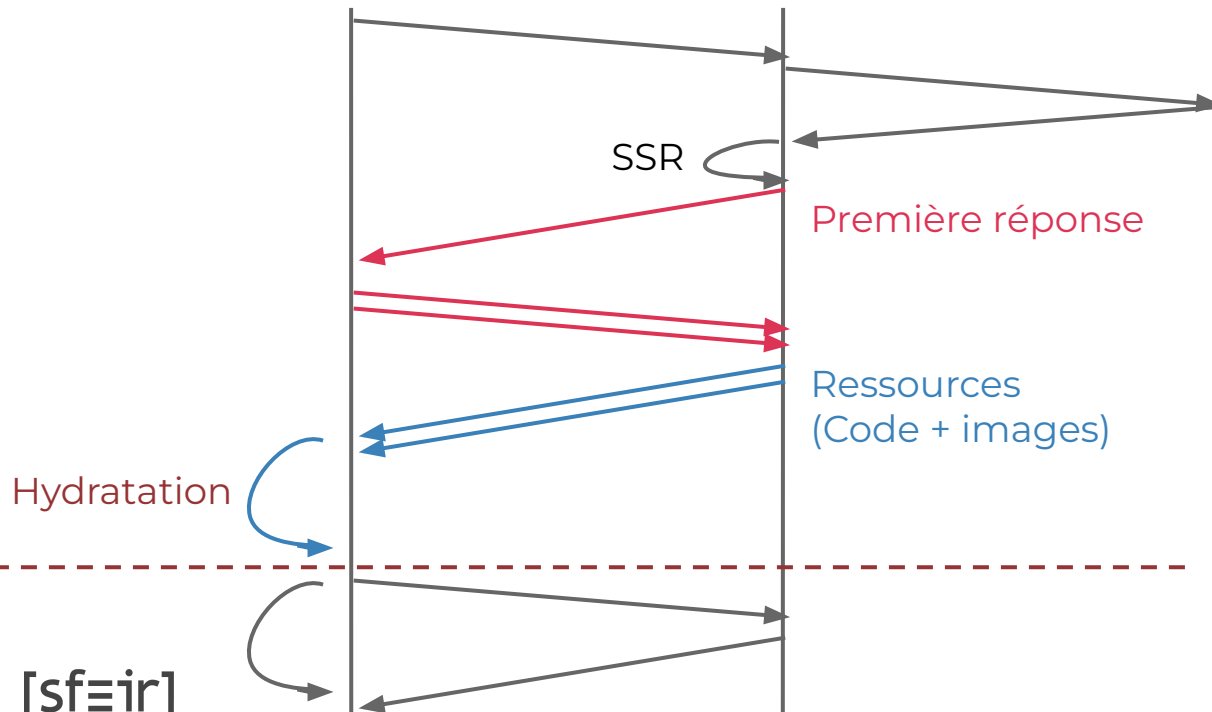




# Hydratation classique avec SolidJs

# Hydratation classique avec Solid Js (demo)

Client                      Back For Front                      Backend



- URLs bundles Js : **18KB**
  - Framework
  - **Composants page**
- Rendered HTML
  - URLs images
  - + **State sérialisé** pour les **composants page**

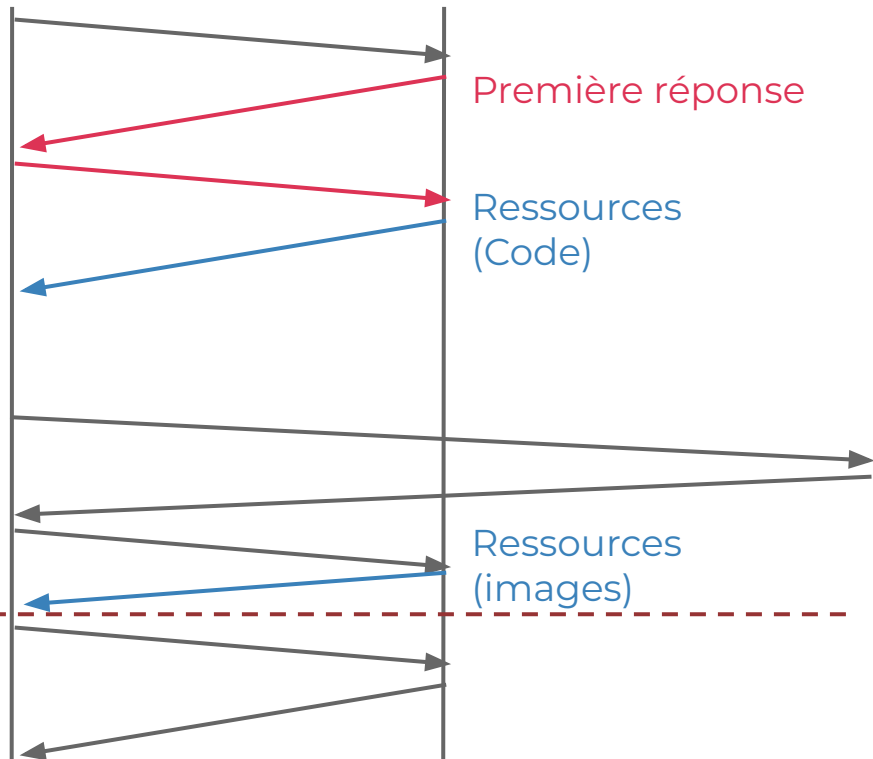


# Client Side Rendering

## avec SolidJs

# Client Side Rendering avec Solid Js (demo)

Client                      Serveur de fichiers                      Backend



- URLs bundles Js : **29KB**
  - Framework
  - **Composants page(s)**
  - **Code appels backend**

Exécution  
initiale : Sans  
contenu

Rendu HTML  
avec contenu

[sf≡ir]

# NextJs

[sf≡ir]



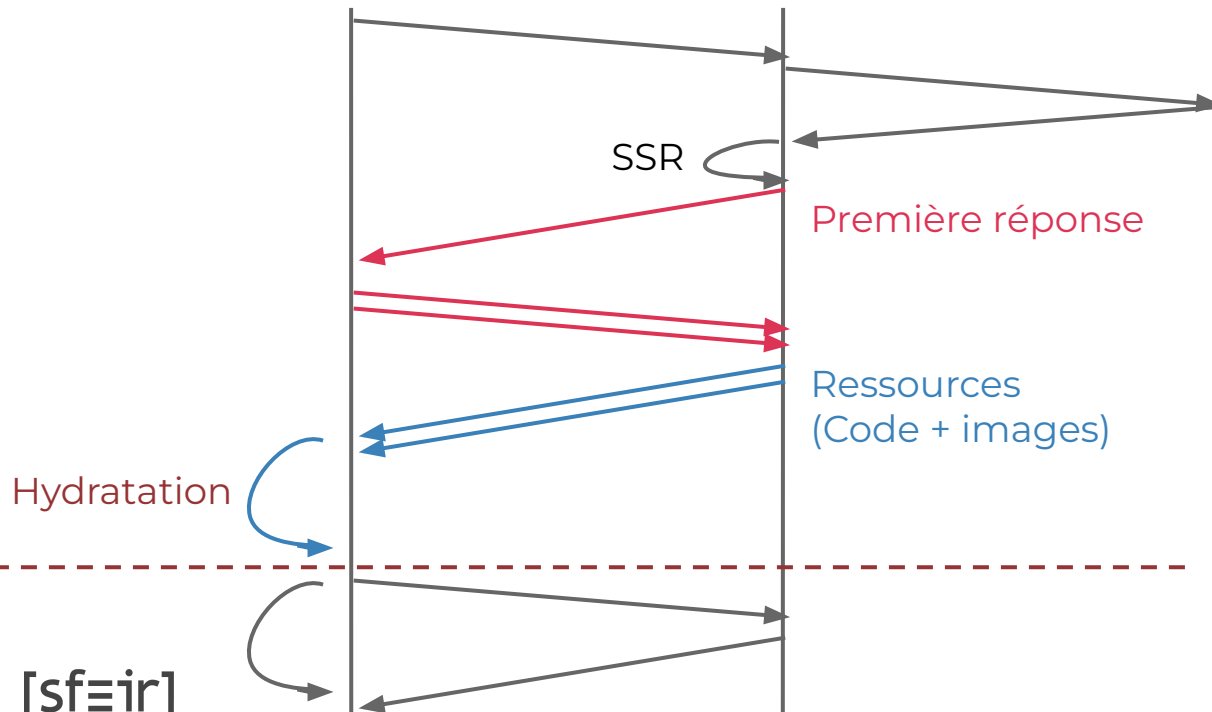


# Hydratation Partielle avec NextJs

DEPUIS v13 (2022-10)

# Hydratation partielle avec Next Js v13 (demo)

Client                      Back For Front                      Backend



- URLs bundles Js : **91,5KB**
  - Framework : **86KB**
    - ReactDOM: **42KB**
  - **Composants client**
- Rendered HTML
  - URLs images
  - + **State sérialisé** pour les **composants client**

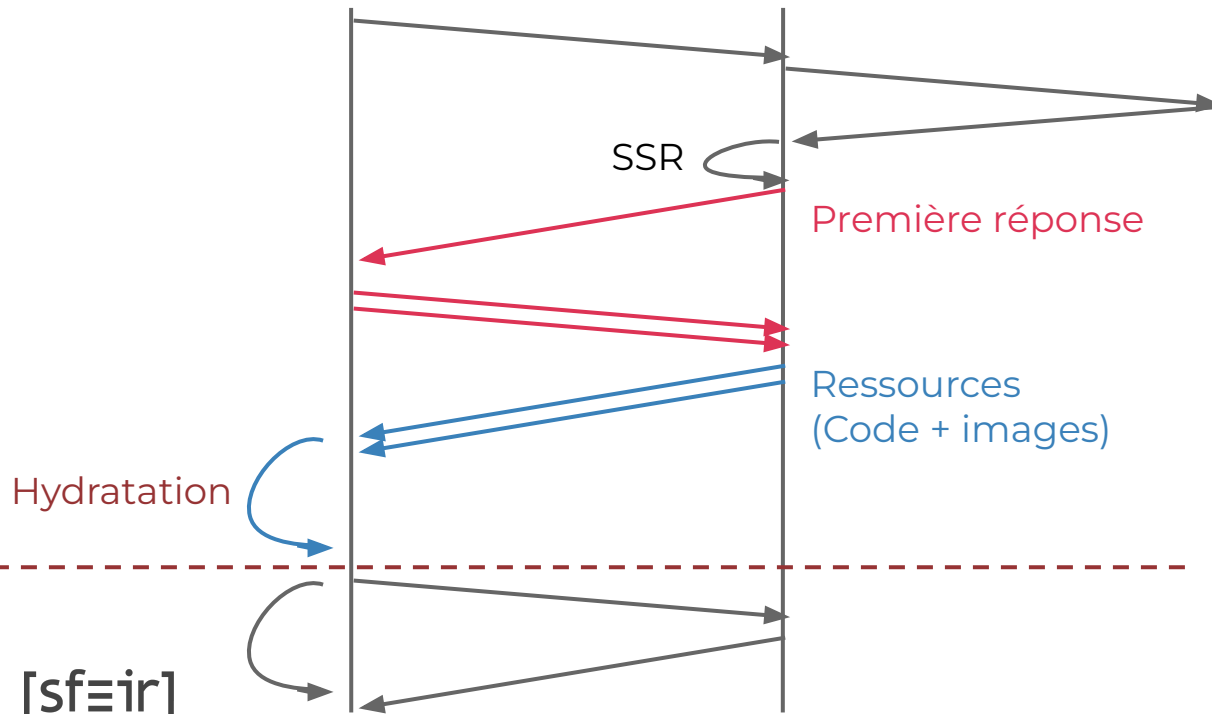




# Hydratation classique avec Next Js

# Hydratation classique avec Next Js v12 [\(demo\)](#)

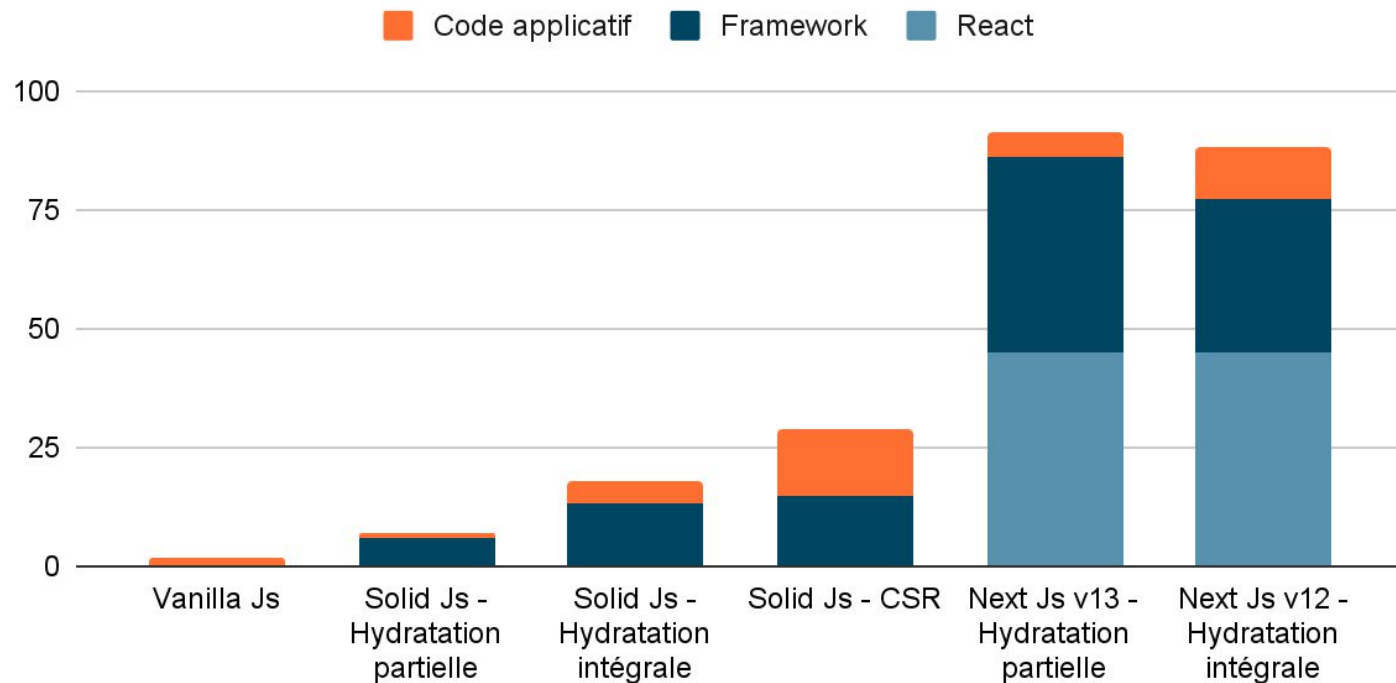
Client                      Back For Front                      Backend



- URLs bundles Js : **88,5KB**
  - Framework : **77KB**
    - ReactDOM: **42KB**
  - **Composants page**
- Rendered HTML
  - URLs images
  - + **State sérialisé** pour les **composants page**

# Recap.

Taille (en KB) du bundle client (Minified + Gzipped) pour la démo Hackernews



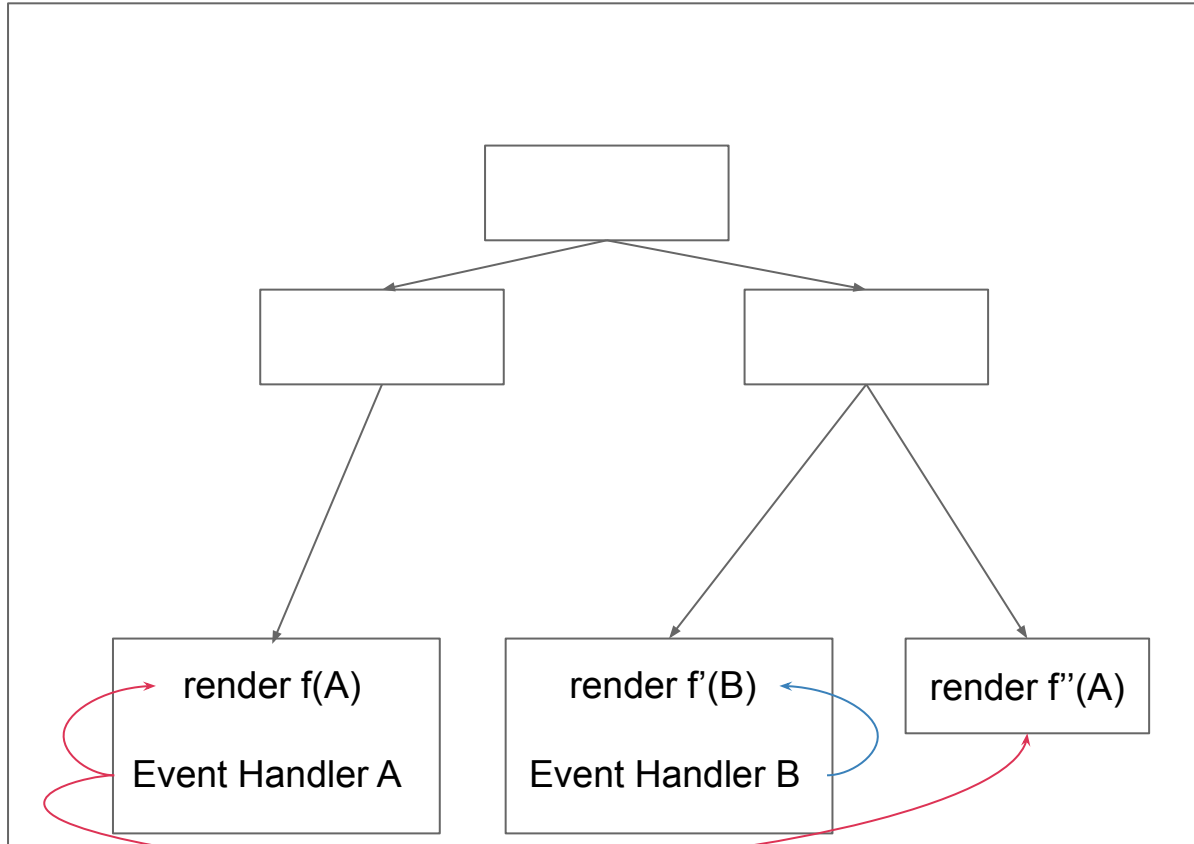


# Et après le chargement initial ?

# Vanilla Js / jQuery

---

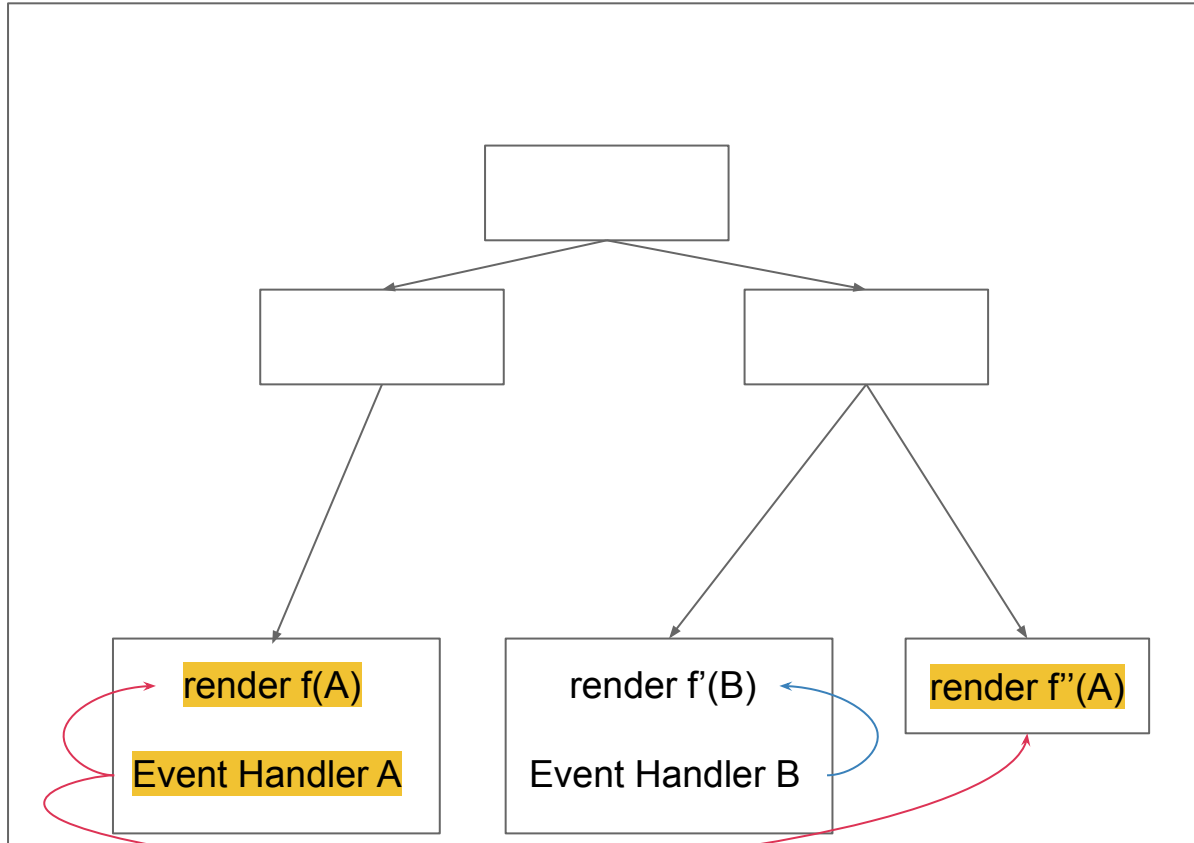
DOM



[sf≡ir]

# Vanilla Js / jQuery

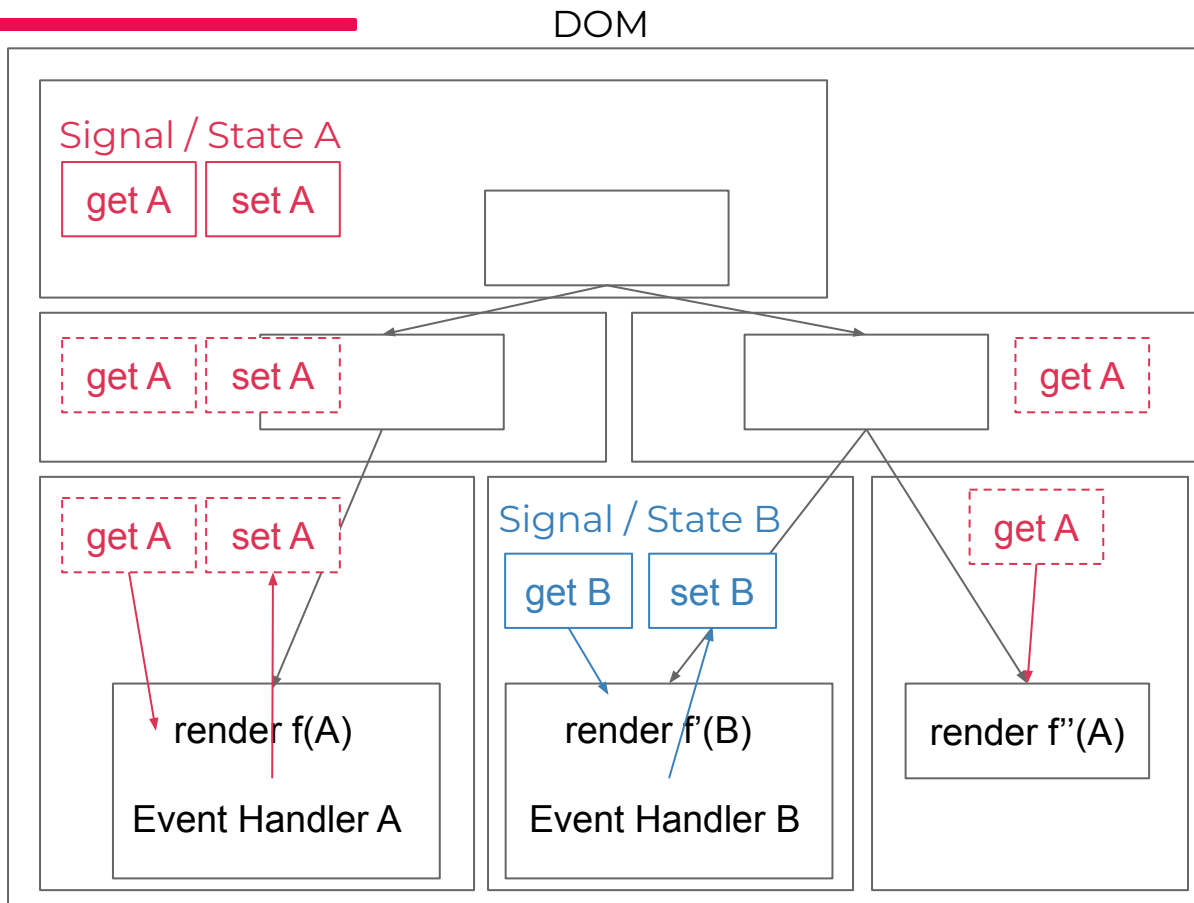
DOM



[sf≡ir]

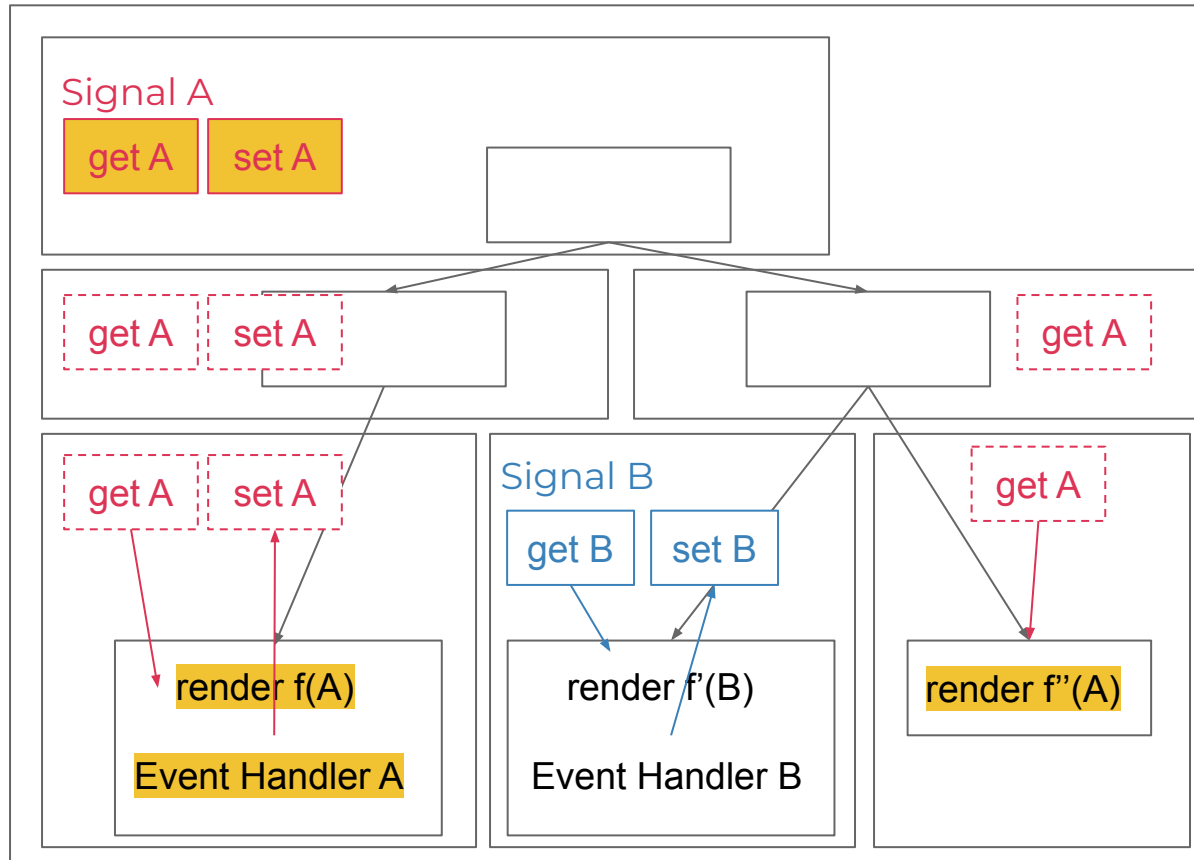
Code exécuté

# Solid Js / React



# Solid Js – Code déclenché par **Event Handler A**

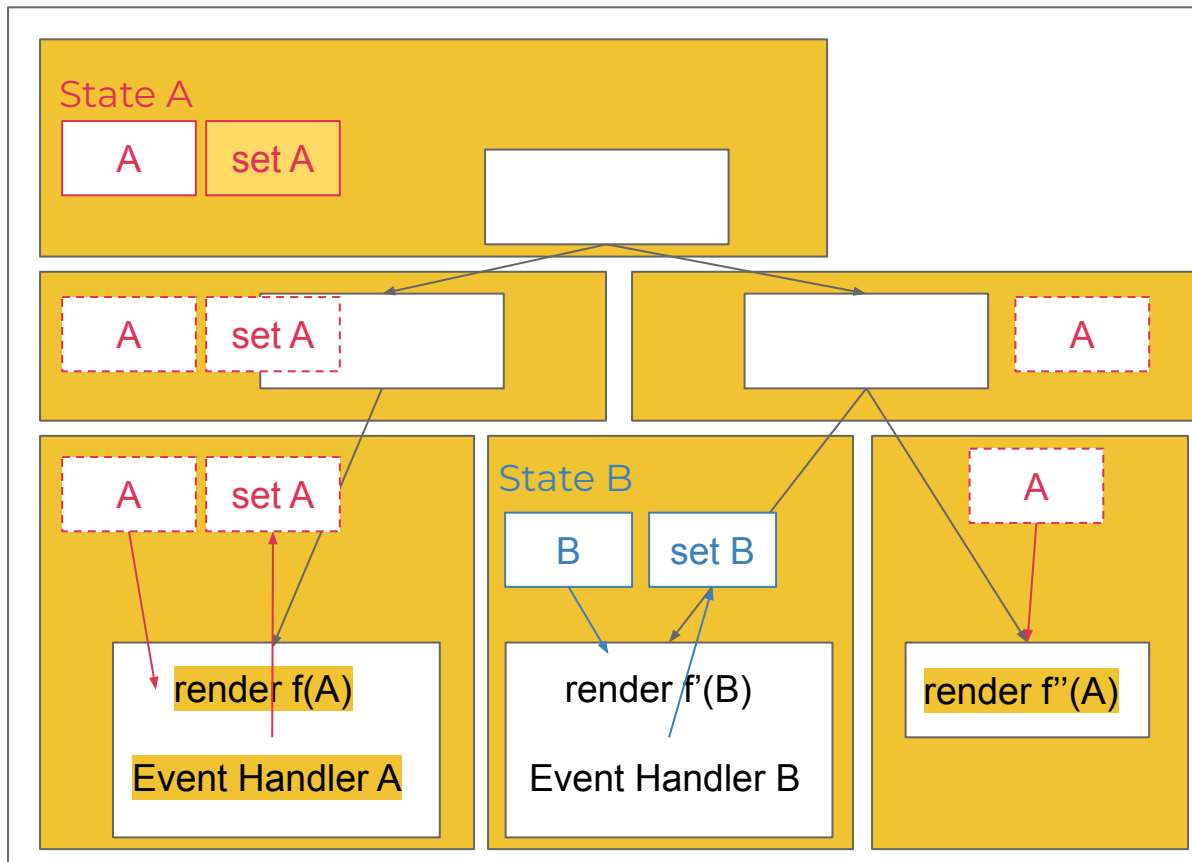
DOM





# React - Code déclenché par Event Handler A

DOM

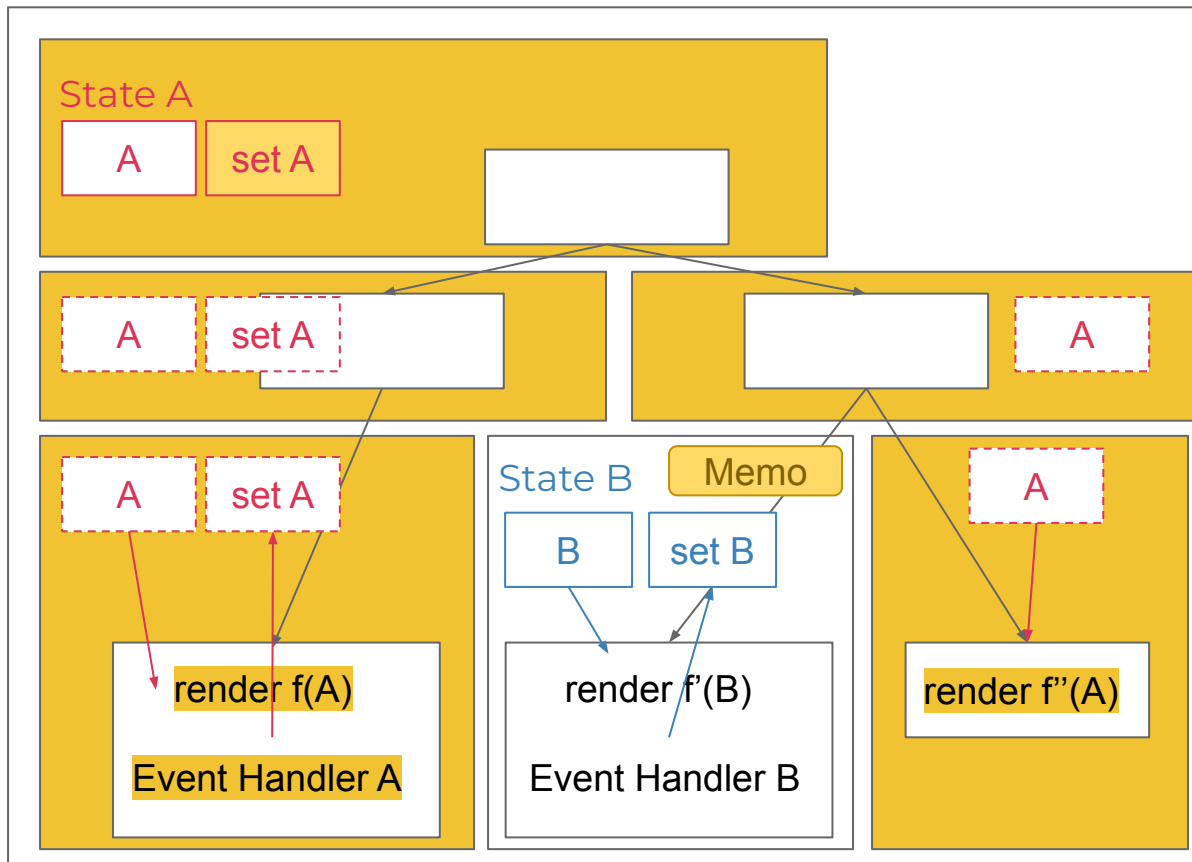


[sf≡ir]

Code exécuté

# React - Code déclenché par Event Handler A

DOM

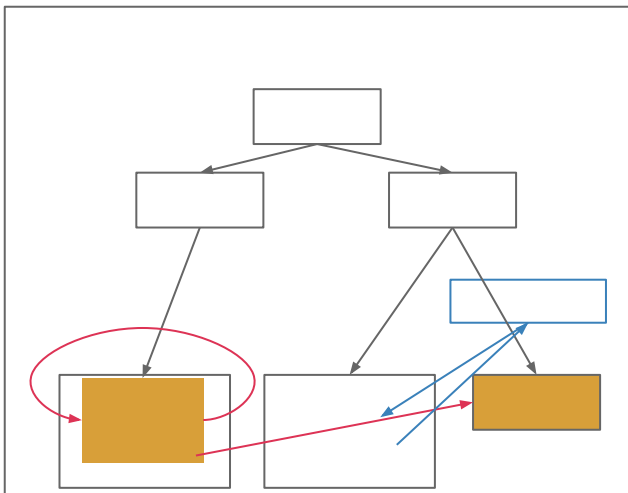


[sf≡ir]

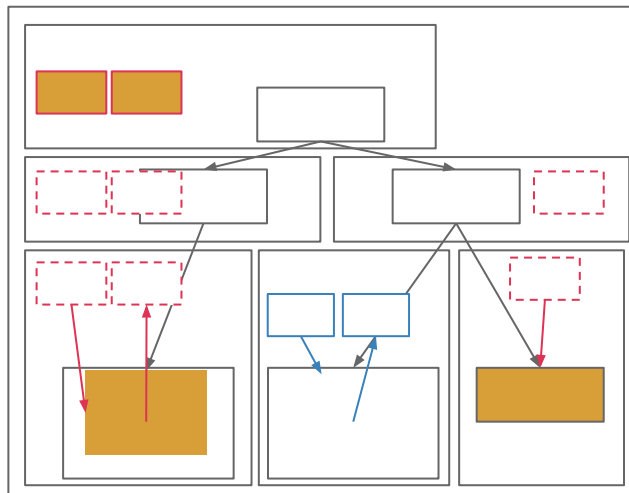
Code exécuté

# Recap.

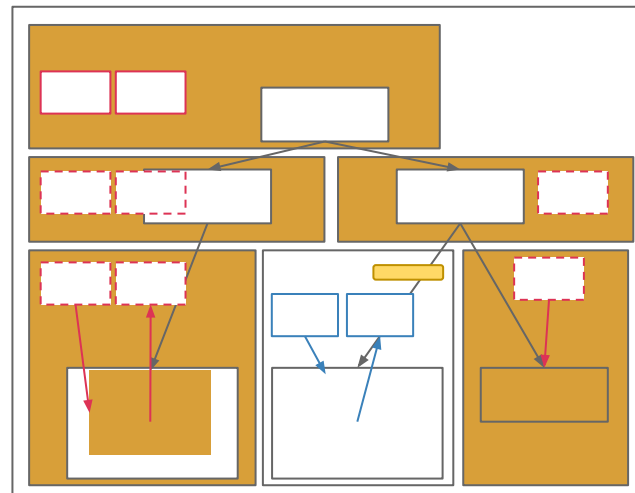
---



Vanilla



Solid Js



React Js

# Conclusion

# Conclusion

---

- SolidJs possède une API similaire aux hooks React
  - ✓ Lisibilité du code
- C'est un framework **lightweight (7KB)** supportant **l'hydratation partielle**
  - ✓ Chargement initial rapide
- Son modèle d'exécution **“Réactivité de granularité fine” (fine-grain)** est basé sur les Signals et Effects (Observables et Observers), et n'est pas lié à la hiérarchie des composants
  - ✓ Bonne performance générale

# Ressources

# Annexe 1 : Ressources

---

- [Guides: Getting started](#)
- [Introduction to Reactivity with SolidJS](#)
- [Chaîne Youtube de Ryan Carniato](#)
- [Getting started with SolidStart](#)
- [Repos des démos \(Par Ryan Carniato\)](#)
- Démos :
  - [Hydratation partielle Solid](#)
  - [Hydratation intégrale Solid](#)
  - [CSR Solid](#)
  - [Hydratation partielle Next](#)
  - [Hydratation intégrale Next](#)



## Annexe 2 : Autres frameworks intéressants

---

- [Astro](#)
  - Hydratation partielle
  - Supporte SolidJs, React, Preact, ... pour les composants client
- [MarkoJs](#)
  - Hydratation partielle “Code elimination”
  - Développé et utilisé par Ebay
  - Version 6 en alpha démontre “resumabilité” + “réactivité fine-grain” (comme SolidJs)
- [Qwik](#)
  - Resumabilité

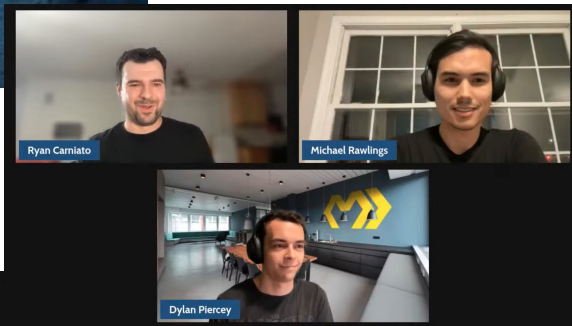
# Annexe 2 : Autres frameworks intéressants

---

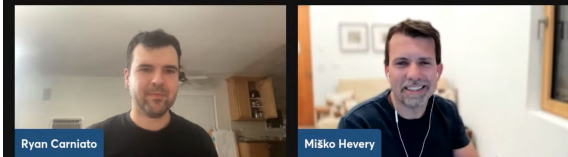
- [Astro chat](#)



- [MarkoJs chat](#)



- [Qwik Talk](#)





**Questions?**

[sf≡ir]