



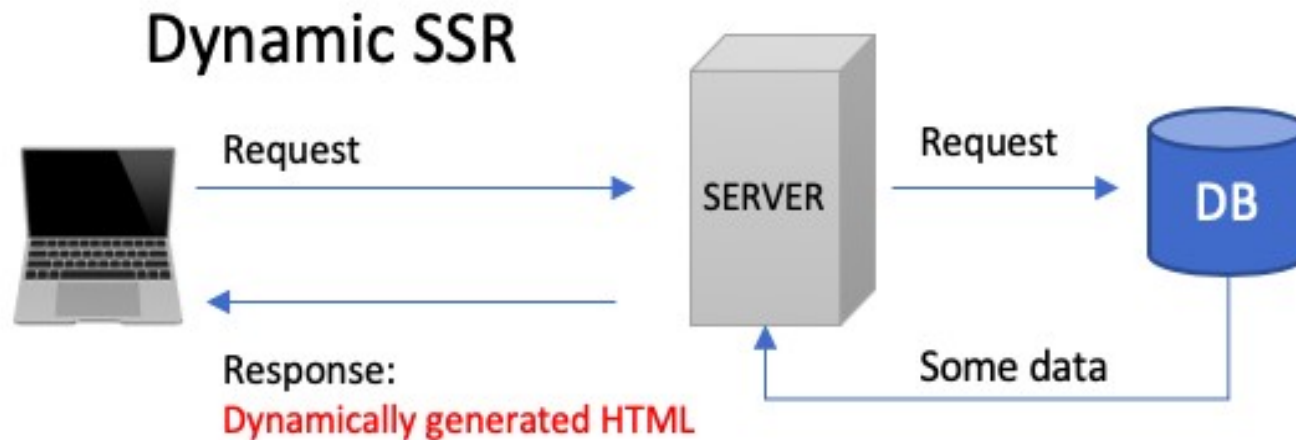
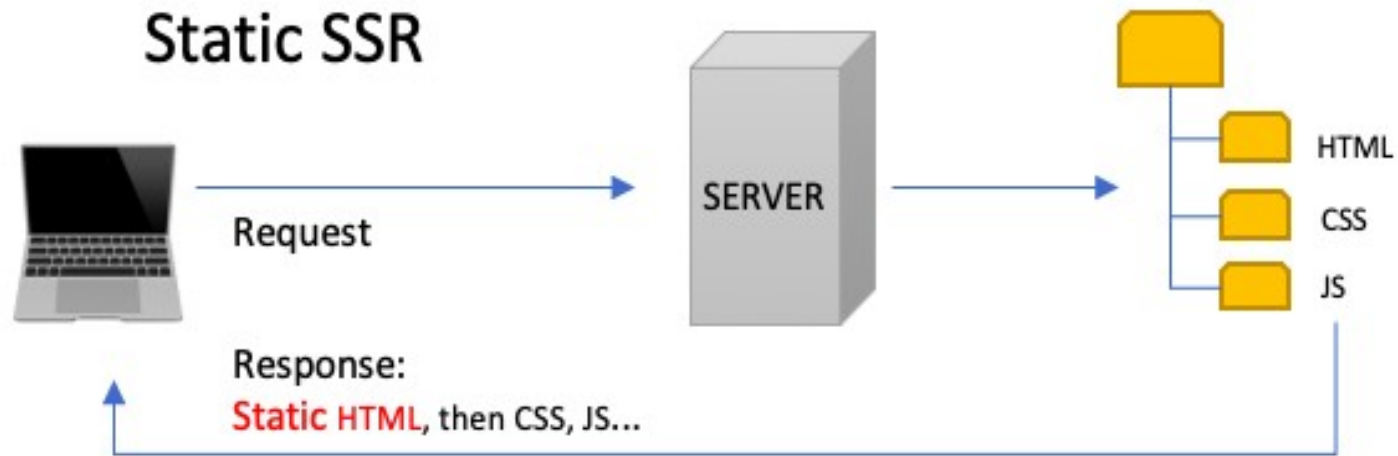
# CSR vs SSR

Les deux concepts...

# Le principe du SSR (1)

- **SSR** signifie **Server-Side Rendering**.
- Traditionnellement, le contenu HTML provenait du serveur, soit sous la forme de pages statiques, soit généré totalement ou partiellement par un langage (PHP par exemple).
- Un changement de page oblige un renvoi de l'ensemble du code par le serveur ; AJAX améliore la situation en effectuant que des renvois partiels.
- Nous sommes dans une logique de type **Back-End**.

# Le principe du SSR (2)

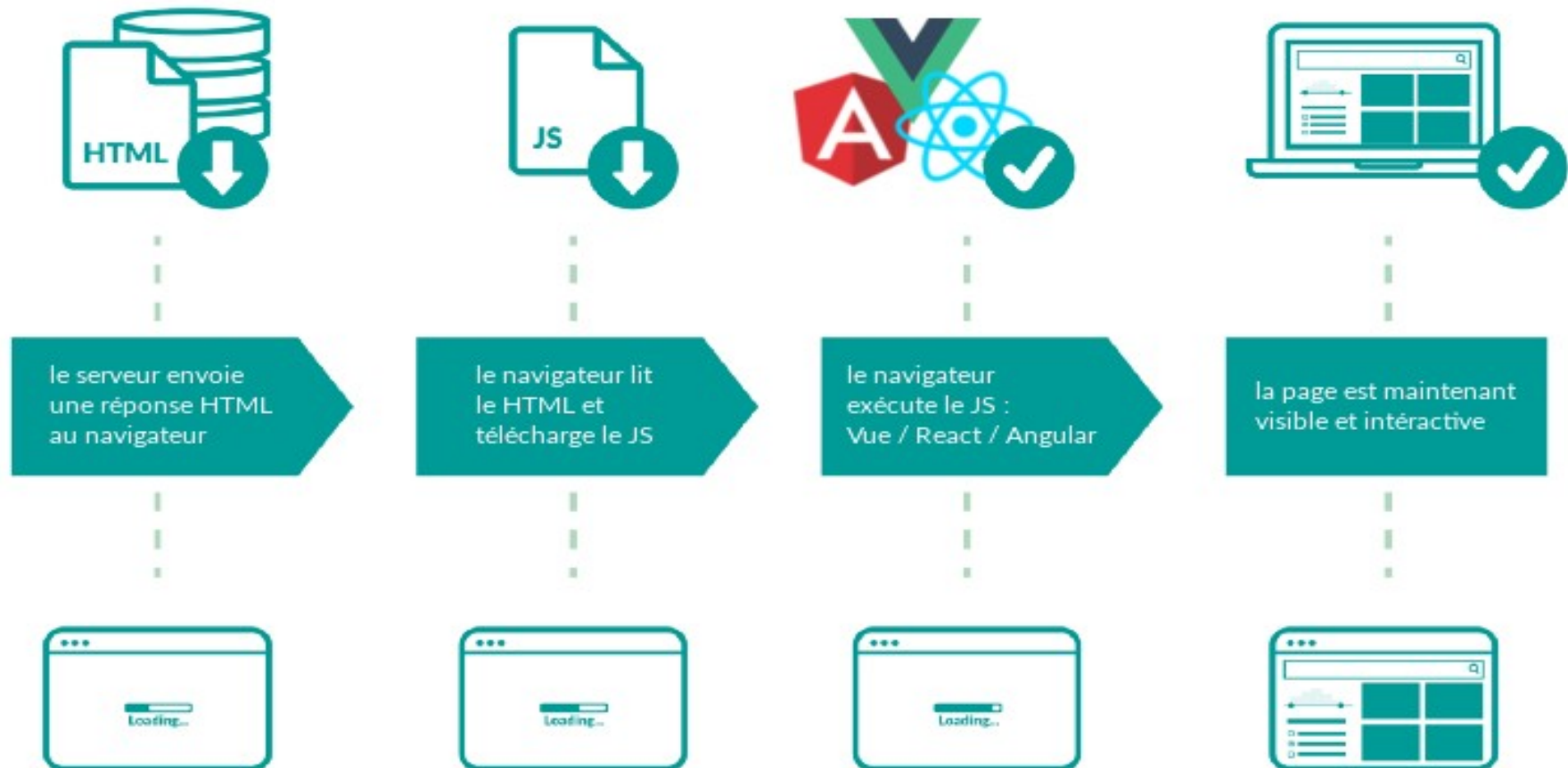


# Le principe du CSR (1)

- **CSR** signifie **Client-Side Rendering**.
- L'interface utilisateur ou contenu HTML est généré par le navigateur par le concept des **Single Page Apps** (SPA) usuellement par un framework JavaScript (React, Angular, Vue,...)
- Le rendu client est dynamique, par le changement du DOM au niveau d'une seule balise HTML.
- Nous sommes dans une logique de type **Front-End**.

# Le principe du CSR (2)

## CSR



# Bilan du CSR

## Positif :

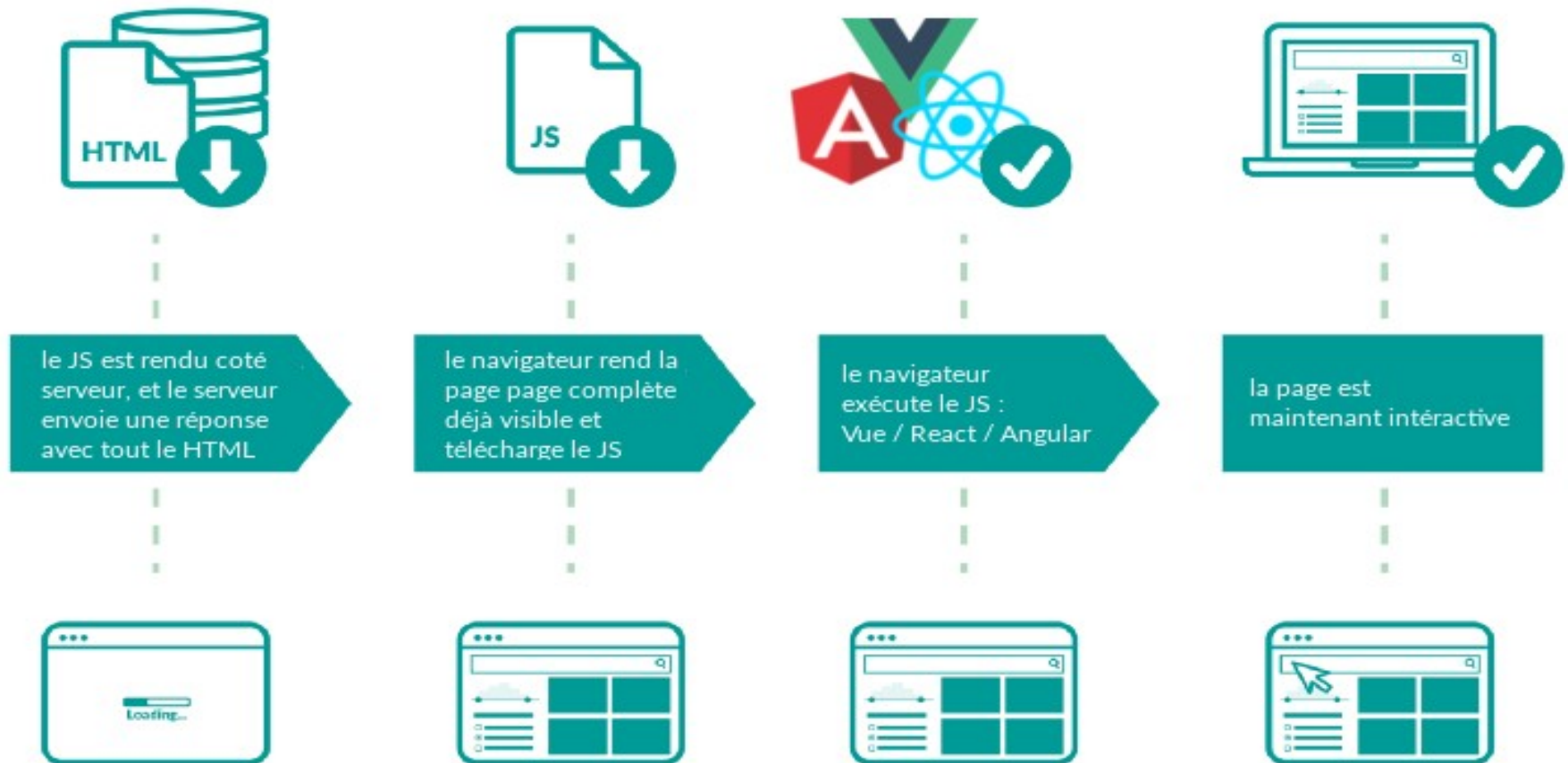
- Le rendu client est interactif et dynamique
- La navigation est rapide car côté navigateur
- Le serveur est faiblement sollicité et plus en sécurité

## Négatif :

- Le chargement de départ est important (plus de code)
- La logique est à l'opposé d'un site statique (utile suivant les cas)
- Le référencement (SEO) est très mauvais

# SSR : le retour ou le rendu côté serveur

## SSR





# L'application universelle ou isomorphique

## Le principe :

- Rendre côté serveur la première page (*landing page*) pour un chargement rapide et un référencement facilité
- Laisser le reste de la navigation à l'application web côté client par l'utilisation du DOM

## Au niveau des frameworks :

- **Next.js** → pour React
- **Angular universal** → pour Angular
- **Nuxt.js** → pour Vue
- Nous sommes dans une logique de type **Full-Stack**.



# Conclusion

## **Il faut utiliser la technologie adaptée en fonction de l'application :**

- Un simple site de présentation nécessitera que des pages statiques ; il existe même des générateurs (Jekyll, Hugo, Pélican)
- Un site de type Intranet gardera la logique du CSR
- Une application Web basée sur des échanges et une demande de visibilité passera au principe de l'application universelle (site de e-commerce, réseau social, etc.)

La tendance est dans l'allègement du code, pour preuve un nouveau venu dans l'univers des frameworks JavaScript : **Svelte**, qui effectue la plupart de la compilation et se passe de DOM virtuel...