XWIKI NATIS

Les librairies JS

Michael
XWIKI NATIS

XDNATIS

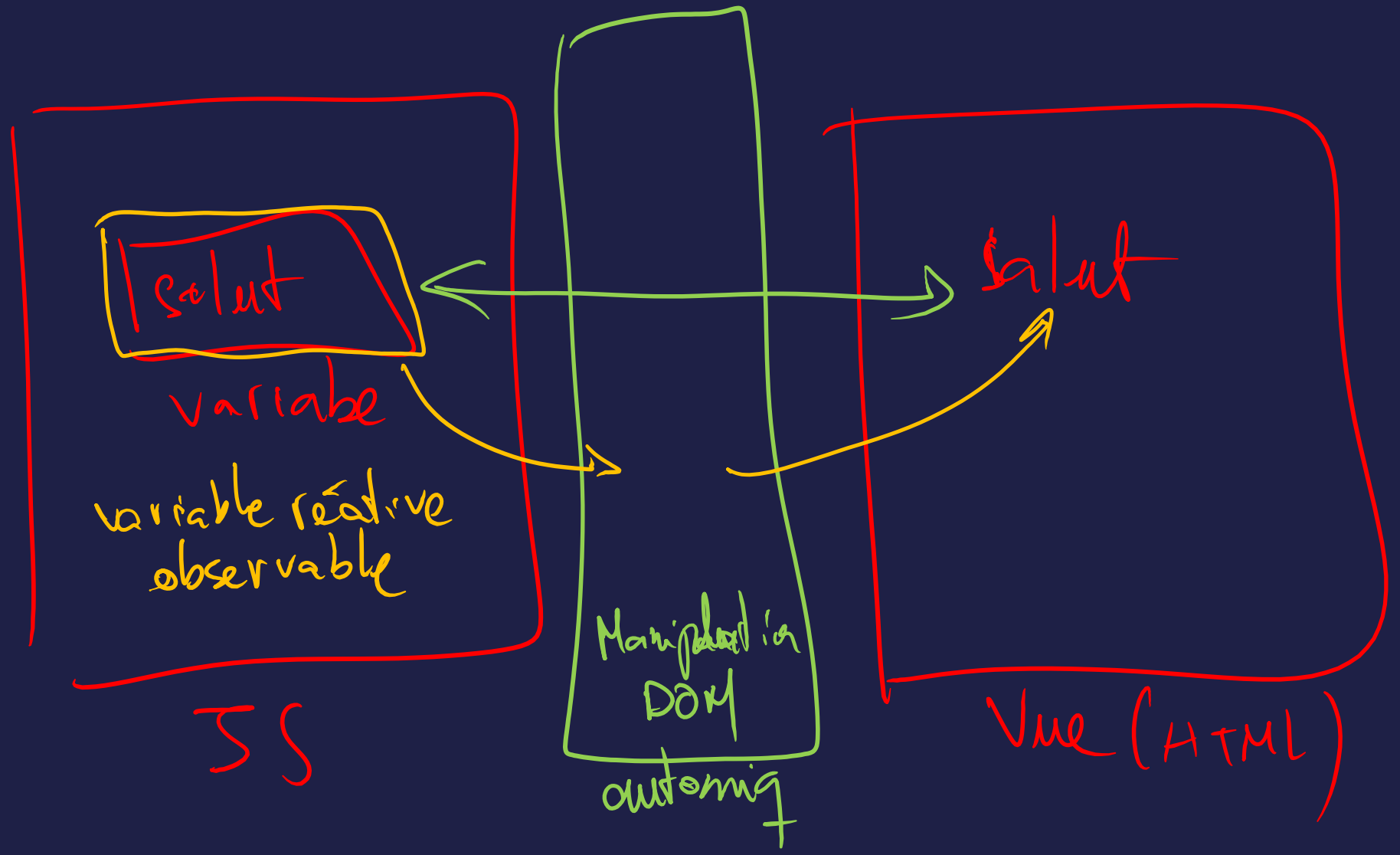
Compétence demandée :

Comprendre le fonctionnement de Knockout et Reactjs

XEBIA NATIS

1. La programmation réactive
2. L'architecture MVVM
3. Knockout
4. Reactjs

XDNATIS


salut
variable
variable réactive
observable
JS
Manipulation
DOM
écoute
$salut
Vue (HTML)

XEBIA NATIS

LA PROGRAMMATION REACTIVE

XEBIA NATIS

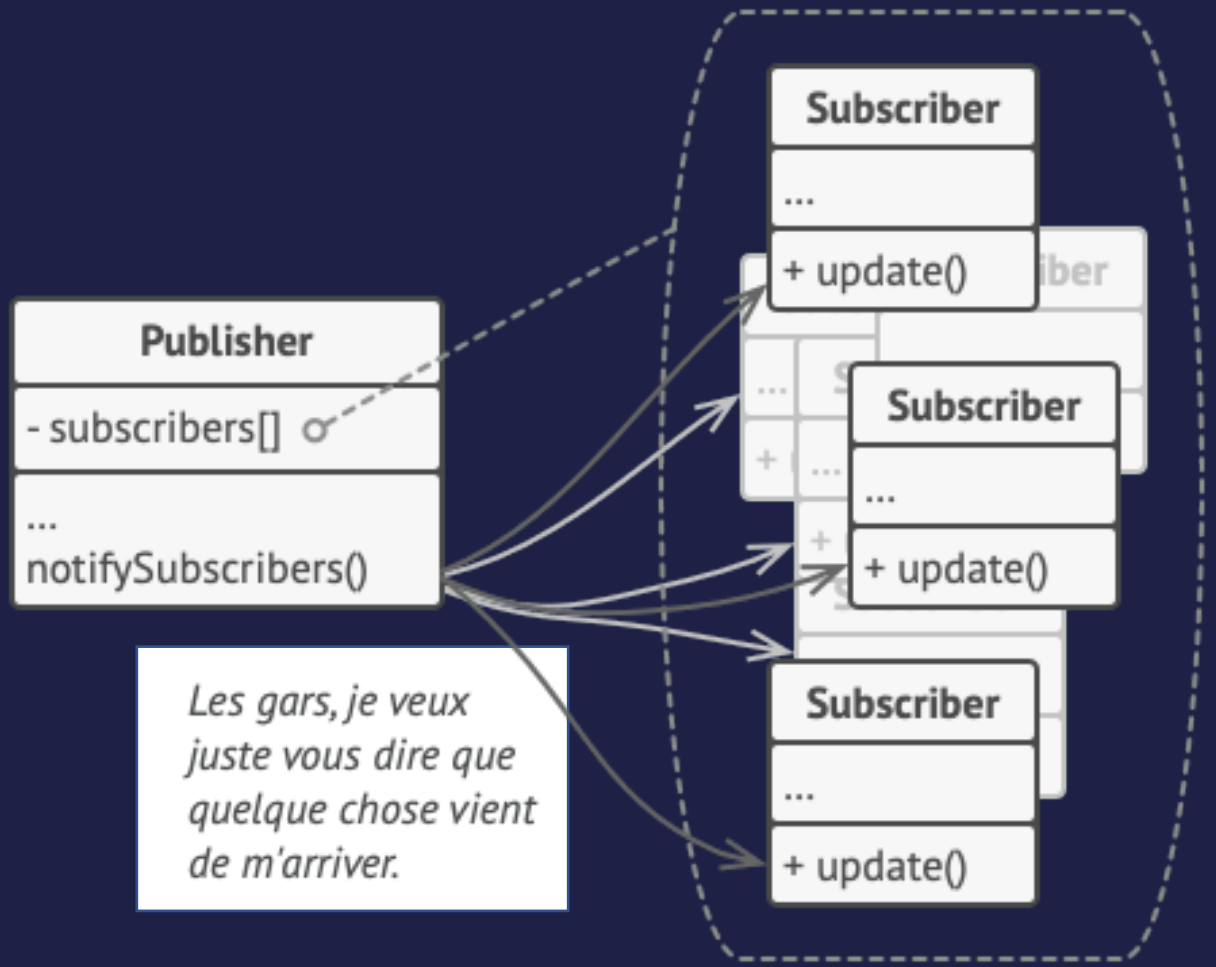
En informatique, la programmation réactive est un paradigme de programmation visant à conserver une cohérence d'ensemble en propageant les modifications d'une source réactive (modification d'une variable, entrée utilisateur, etc.) aux éléments dépendants de cette source.

XEBIA NATIS

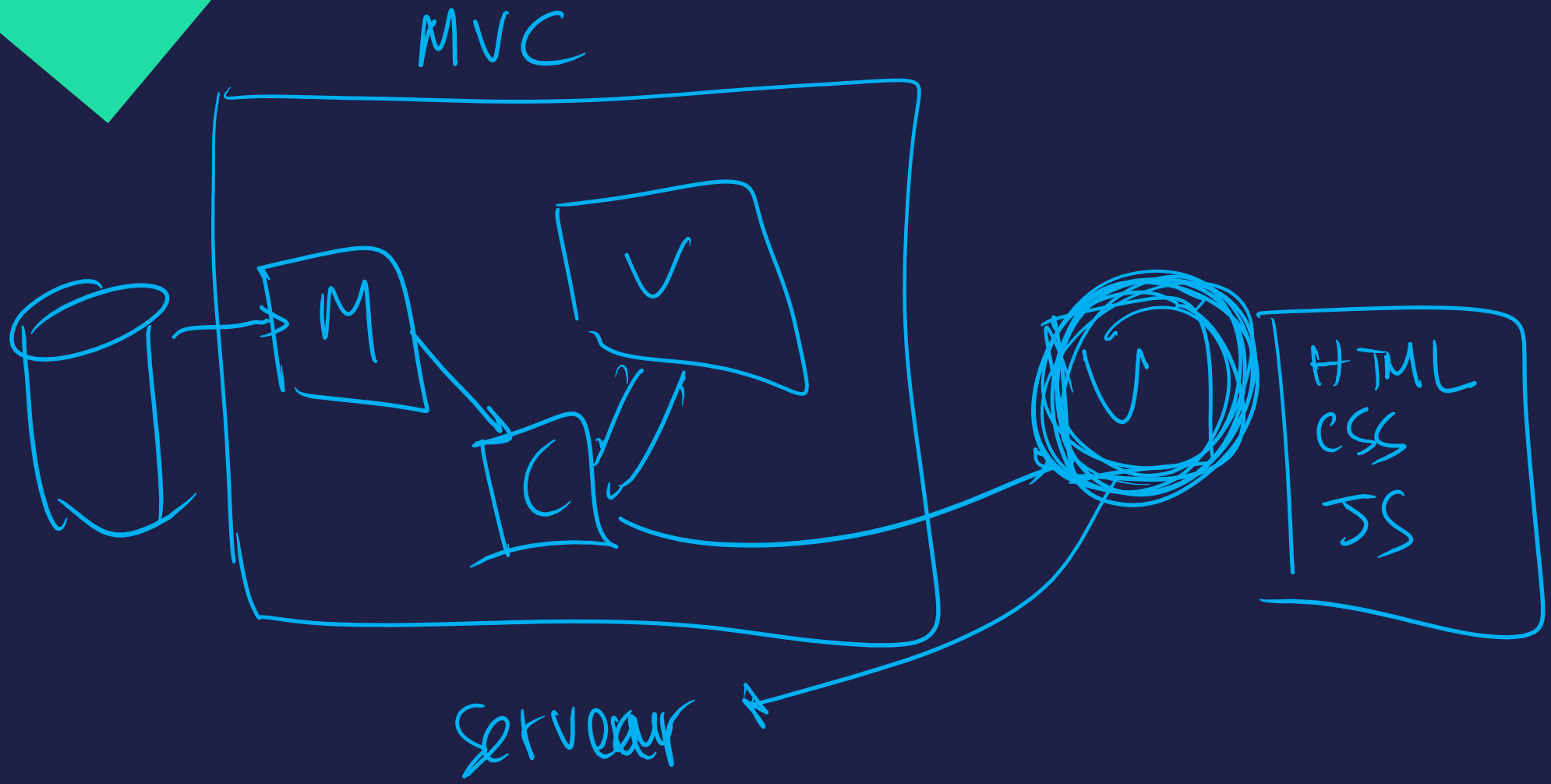
La programmation réactive est possible grâce au design pattern Observer.

L’Observer est un design pattern dans lequel un objet, nommé le sujet, maintient une liste de ses dépendants, appelés observateurs, et les notifie automatiquement de tout changement d'état, généralement en appelant l'une de leurs méthodes.

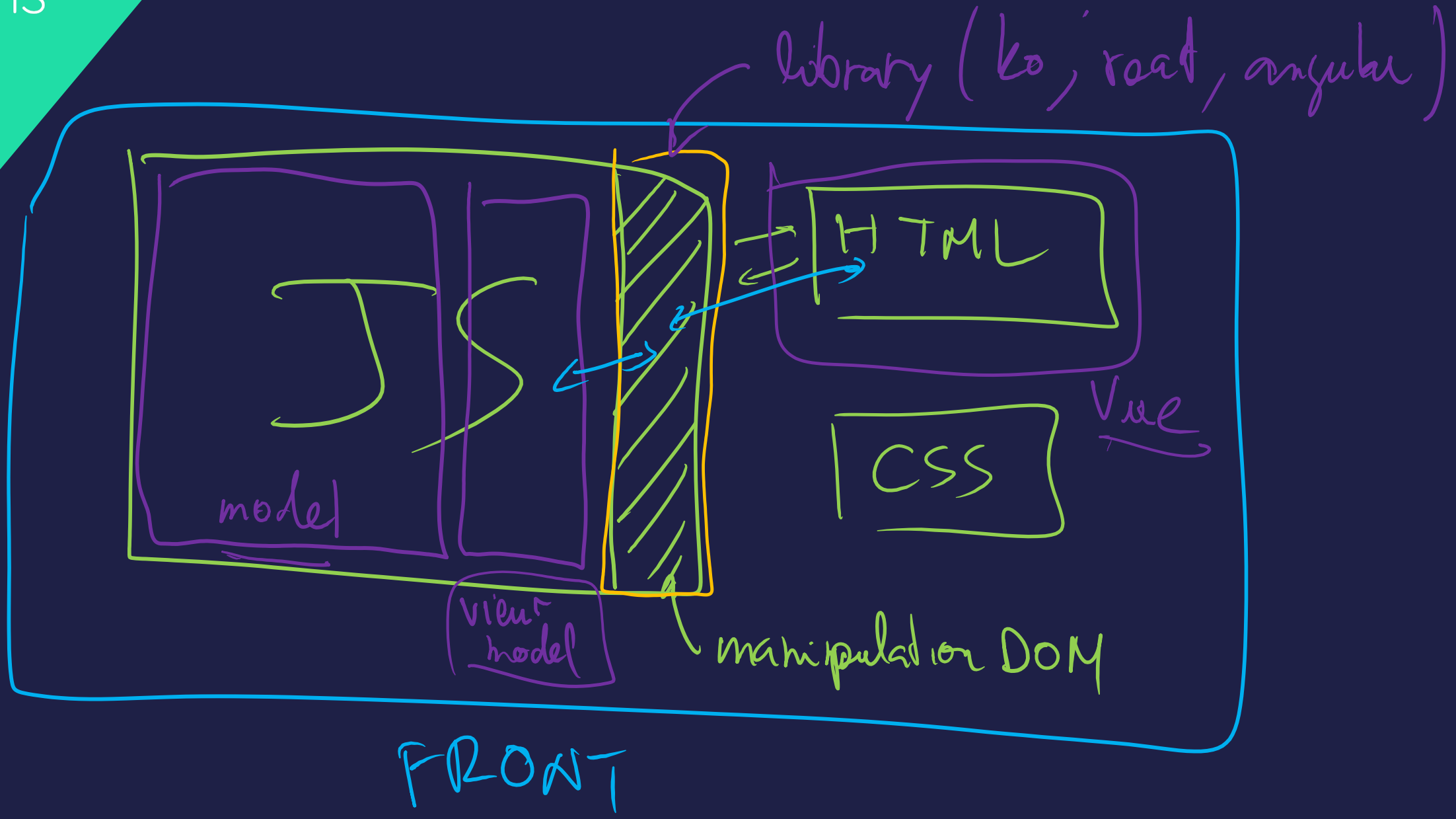
XEBIA NATIS


Publisher
- subscribers[]
...
notifySubscribers()
Subscriber
...
+ update()
Subscriber
...
+ update()
Subscriber
...
+ update()
Les gars, je veux juste vous dire que quelque chose vient de m'arriver.

XDNATIS



XEBIA NATIS


library (ko, react, angular)
JS
model
view model
HTML
CSS
Vue
manipulation DOM
FRONT

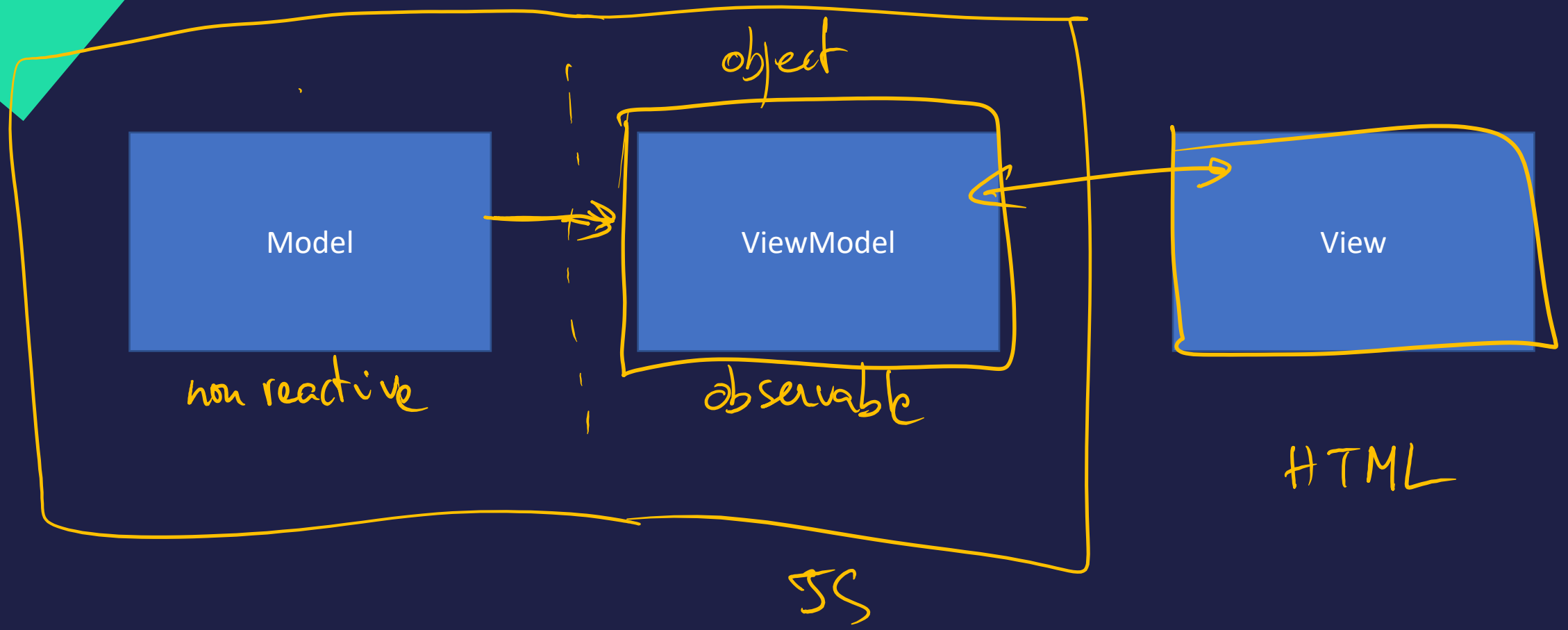
XEBIANATIS

ARCHITECTURE MVVM

XEBIA NATIS

L’architecture MVVM, de l'anglais Model, View, ViewModel, est une architecture et une méthode de conception utilisée dans le génie logiciel notamment pour le développement front-end.

XEBIA NATIS


object
Model
ViewModel
View
non reactive
observable
HTML
JS

XEBIA NATIS

Apparu en 2004, MVVM est originaire de Microsoft et adapté pour le développement des applications basées sur les technologies Windows Presentation Foundation (WPF) par exemple.

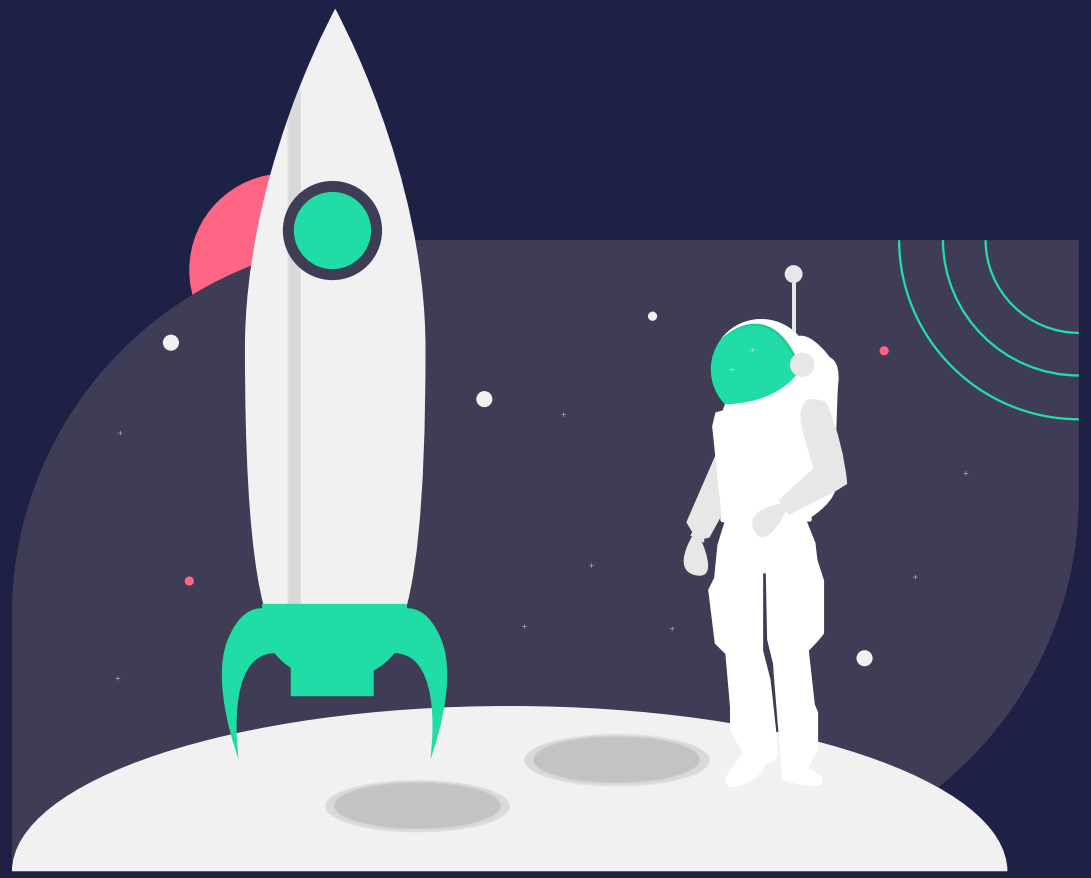
Cette méthode permet, tel le modèle MVC (modèle-vue-contrôleur), de séparer la vue de la logique et de l'accès aux données en accentuant les principes de liaison et d'événement.

XONATIS

XEBIA NATIS

1. La programmation réactive
2. L'architecture MVVM
3. Knockout
4. Reactjs

XWIKI NATIS



XONATIS

XWIKI NATIS

Knockout

XWIKI NATIS

Knockout est une vielle librairie qui a été l'une des premières à implémenter l'architecture MVVM.

Elle instaure un processus de développement simple, mais qui a ses limitations lorsque le projet devient complexe.

XKNATIS

Le remplaçant de Knockout le plus populaire est Vue.

Son fonctionnement de base est quasi-identique et Vue prend en charge les cycles de vie, une réactivité à l'enfant beaucoup plus simple par exemple.

XEBNATIS

```
<div>
    <h1>Fitness tracker</h1>
    <input type="text" placeholder="Poids" data-bind="value: weight" />
    <button data-bind="click: addToHistory">Ajouter</button>
    <div data-bind="foreach: history">
        <div data-bind="text: $data"></div>
    </div>
</div>
<script>
    class ViewModel {
        weight;
        history;

        constructor() {
            this.weight = ko.observable();
            this.history = ko.observableArray();
        }

        addToHistory() {
            this.history.push(this.weight);
        }
    }
    (function() {
        const viewModel = new ViewModel();
        ko.applyBindings(viewModel);
    })();
</script>
```

XEBIANATIS

Pour ajouter Knockout a un projet, il suffit de l'inclure dans une balise script en HTML

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/knockout/
3.5.0/knockout-min.js"></script>
```

XWIKI NATIS

Pour aller plus loin

https://knockoutjs.com/

XONATIS

XEBIA NATIS

Reactjs

XWIKI NATIS

Facebook

XWIKI NATIS

Uber

XWIKI NATIS

Instagram

XEBIALABS

Codecademy

XWIKI NATIS

Yahoo Mail!

XEBIA NATIS

New York Times

XWIKI NATIS

Paypal

XONATIS

Netflix

XWIKI SAS

Etc.

XWIKI SAS

XWIKI NATIS

Le but des librairies et frameworks JS modernes

XWIKI NATIS

Le but des librairies et frameworks JS modernes

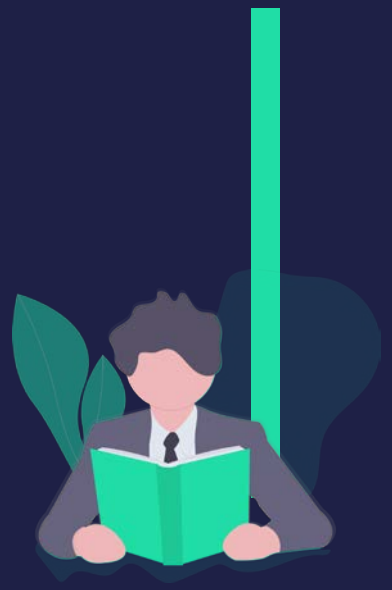
Produire une page statique (static page)

XWIKI NATIS

Une page statique : une page qui est renvoyée au client telle qu'elle a été stockée sur le serveur

XWIKI NATIS

Une page statique : une page qui est renvoyée au client telle qu'elle a été stockée sur le serveur



XWIKI NATIS

- Avoir un changement instantané
- Pouvoir faire des traitements en parallèle
- Améliorer le UX

XNATIS

Processus de développement FRONT

XWIKI SAS

XDNATIS

La doc est magnifique

https://fr.reactjs.org/

XONATIS

XWIKI NATIS

CONSTRUCTION D'UN PROJET

XEBIA NATIS

```
npm init react-app quizz
cd quizz
npm run start
```

XONATIS

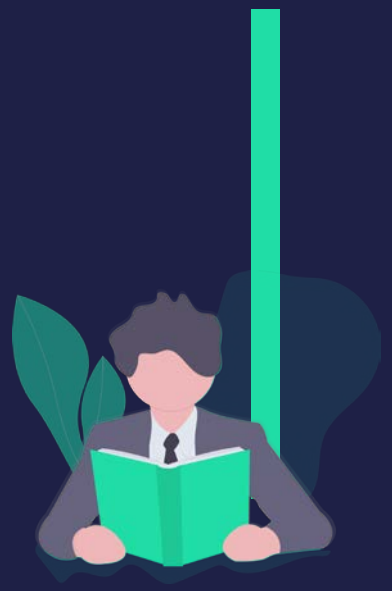
XWIKI NATIS

LES COMPOSANTS

XONATIS

XWIKI NATIS

Une page est un composant fait de composants



XEBIANATIS

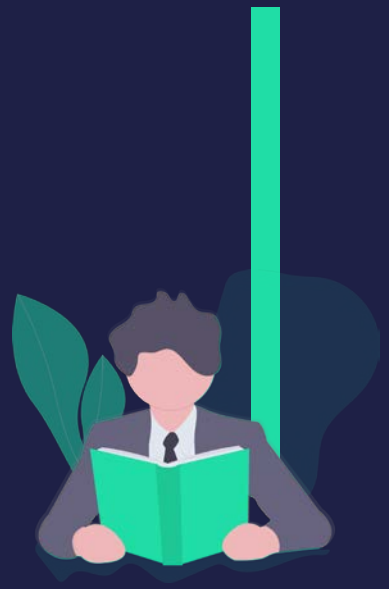
```
function App() {
    return (
        <h1>It works!</h1>
    );
}

export default App;
```

```
class App {
    render() {
        return (
            <h1>It works!</h1>
        );
    }
}

export default App;
```

XEBIA NATIS



Les functional components sont des fonctions

Les class components sont des classes

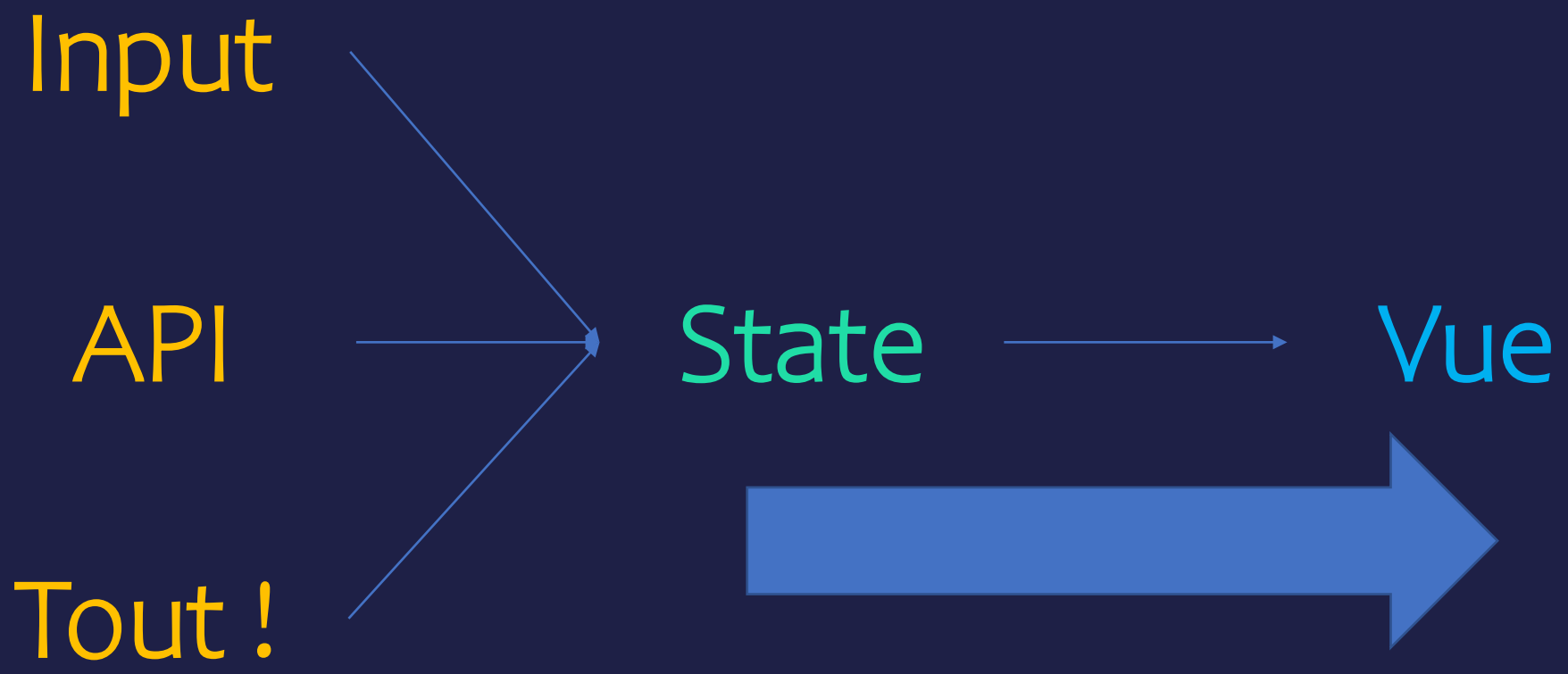
XWIKI SAS

XEBIA NATIS

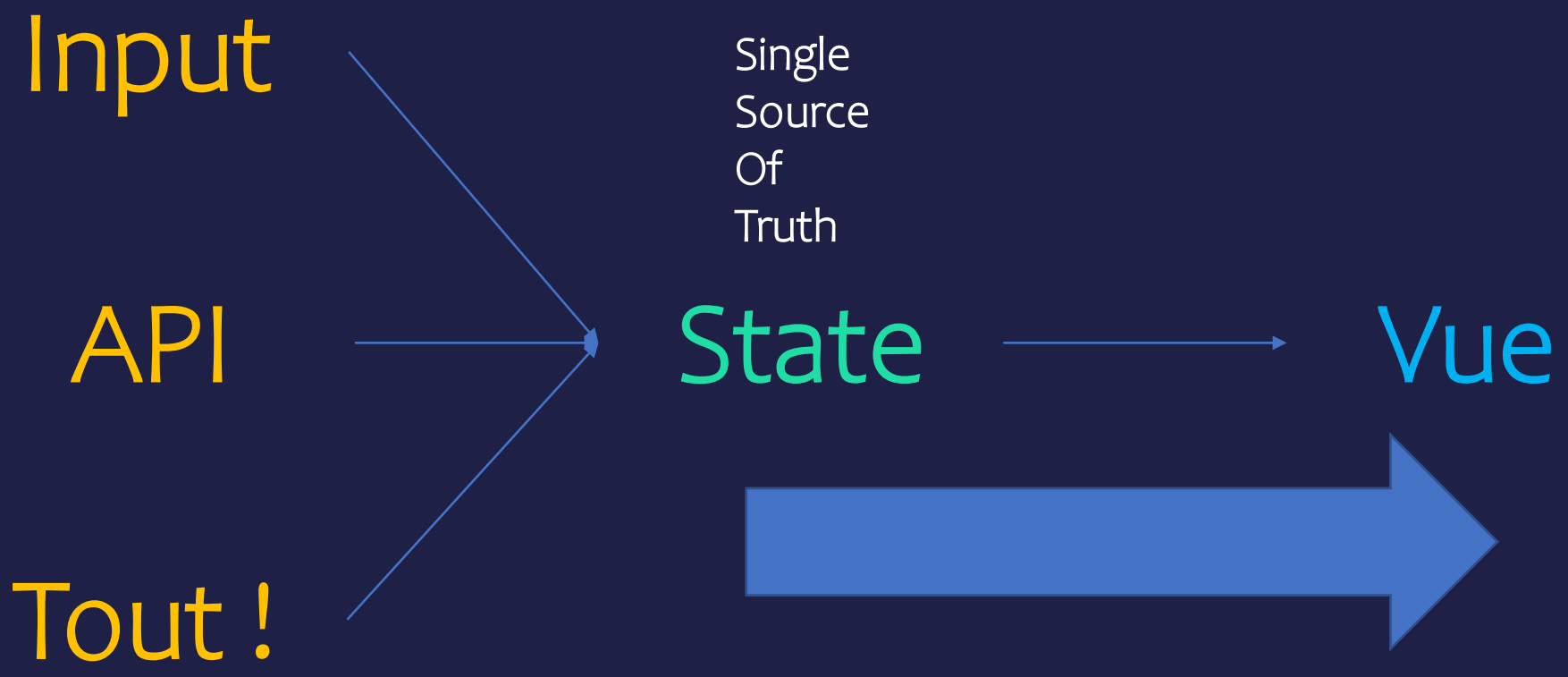
LE STATE

XEBIA NATIS

XDNATIS



XWIKI NATIS



XWIKI NATIS

```
import { useState } from 'react';

function App() {

    const [minuterie, setMinuterie] = useState(3);

    return (
        <div>
            <div>{ minuterie }</div>
            <button onClick={() => setMinuterie(6)}>Augmenter</button>
        </div>
    );
}

export default App;
```

XONATIS

XEBIA NATIS

LES CONTROLLED COMPONENTS

XPNATIS

```jsx
import { useState } from 'react';

function Home() {

    const [email, setEmail] = useState('');

    return (
        <input type="email" value={email} onChange={e => setEmail(e.target.value)} />
    );
}

export default Home;
```

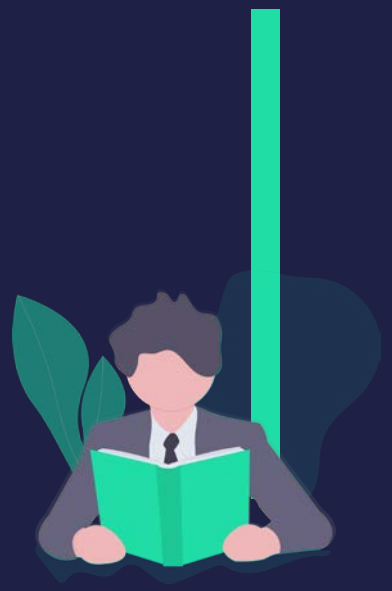
XONATIS

XWIKI NATIS

Pour les formulaires,
N'oubliez pas de changer
le state !

XWIKI NATIS

Pour les formulaires,
N'oubliez pas de changer
le state !



XONATIS

XWIKI NATIS

LE CONTEXTE

XWIKI NATIS

Context

Partagé par tous !

XWNATIS

```
import React from 'react';

const ContextGlobal = React.createContext({
    email: null,
    age: null
});

export default ContextGlobal;
```

XEBNATIS

```
import { useContext } from 'react';
import ContextGlobal from '../common/ContextGlobal';

function Home() {
    const context = useContext(ContextGlobal);

    const handleClick = () => {
        context.email = email;
    };

    return (
        <button onClick={handleClick}>Enregistrer</button>
    );
}

export default Home;
```

XONATIS

XEBIA NATIS

LES PROPERTIES

XDNATIS

XEBIA NATIS

```
function App() {
    return (
        <MonJolieBouton nomDuBouton="Acheter maintenant"></MonJolieBouton>
    );
}

export default App;
```

```
function MonJolieBouton(props) {
    return (
        <button className="css-jolie">{props.nomDuBouton}</button>
    );
}

export default MonJolieBouton;
```

XEBIA NATIS

```
function App() {
    return (
        <MonJolieBouton nomDuBouton="Acheter maintenant"></MonJolieBouton>
    );
}

export default App;
```

```
function MonJolieBouton(props) {
    return (
        <button className="css-jolie">{props.nomDuBouton}</button>
    );
}

export default MonJolieBouton;
```

XWIKI SAS

XEBIA NATIS

Pour aller plus loin

https://fr.reactjs.org/