BOOMERANG
CONSULTING

FORMATION

- Management

boomerang-consulting.com

BOOMERANG
CONSULTING

Bienvenue sur cette formation Boomerang Consulting

- Guilian GANSTER
- 6 ans en tant que développeur freelance spécialisé dans les startups innovantes
- 4 ans en tant que formateurr

Boomerang Consulting

Fort de 20 années d'expertises dans la formation professionnelle continue, notamment en informatique, Boomerang Consulting est un organisme qui cherche à vous offrir la qualité optimum pour vos formations avec :

- Un accueil de qualité et de proximité,
- Un service commercial, administratif et pédagogique à votre disposition,
- Un positionnement technologique étendu avec des formateurs experts, qualifiés et certifiés pour certains,
- Une qualité de services et une réponse à vos besoins grâce aux contenus pédagogiques **SUR-MESURE**.

BOOMERANG CONSULTING

Centre de formation certifié

- L’engagement pour la qualité en étant certifié Qualiopi.
- Centre de formation spécialisé dans les domaines de l’IT, Ressources humaines et développement personnel.

Qualité et satisfaction

- Des formateurs certifiés et expérimentés.
- Un taux de satisfaction client supérieur à 93% suite aux évaluations de fin de stage.

Horaire et convocations

- 9h00- 17h30 (pause 12h15 - 13h30)
- 15 mn de pause le matin
- 1h de pause déjeuner
- 15 mn de pause l'après midi
- Dernier jour à 16h30 (si le plan de cours est terminé uniquement)

BOOMERANG CONSULTING

Prérequis

- Sur quels langages avez vous l'habitude de travailler ?
- Avez vous déjà utilisé un langage bas niveau (C, Rust, Go, …)
- Quelles sont vos attentes/cas d'usage à l'issue de cette formation ?
- Droits admin sur vos PC ?

BOOMERANG CONSULTING

Objectifs pédagogiques

A l'issue de cette formation, vous serez capable de :

- Comprendre, lire et écrire des classes en langage C++
- Traduire une conception objet en langage C++
- Développer des applications en langage C++

Les feuilles d’émargement

Matin et après midi

- Première chose à faire : signer les feuilles d’émargement. Merci de bien vouloir faire des signatures conformes.
- Les croix et initiales ne sont pas autorisées.

Les évaluations de fin de stage

- Dernier jour de la formation : le centre de formation à l'obligation contractuelle de fournir vos évaluations à votre entreprise avant 15h, donc au retour de la pause déjeuner, 2 ou 3 h avant la fin de la formation, je vous ferai remplir les évaluations formateur.

BOOMERANG CONSULTING

Déroulé et structure de la formation

- Équilibre théorie / pratique: 50%
- Chaque notion sera illustré d'un bloc théorique suivi d'un exercice pratique. Une slide avec un code minimale sera fournis pour la syntaxe. Les exercices pratiques reflètent dans la mesure du possible des cas réels.

BOOMERANG CONSULTING

Support de cours et outils pédagogiques

- Les énoncés des TP de validation des acquis sont présent sur ces slides.
- Les corrigés et différents codes d'exemple seront sur Github et/ou sur un Drive et accessibles en fin de formation.

Suivi quotidien et adaptabilité

- Google Forms de demi-journée pour la validation des acquis et l'adaptabilité.

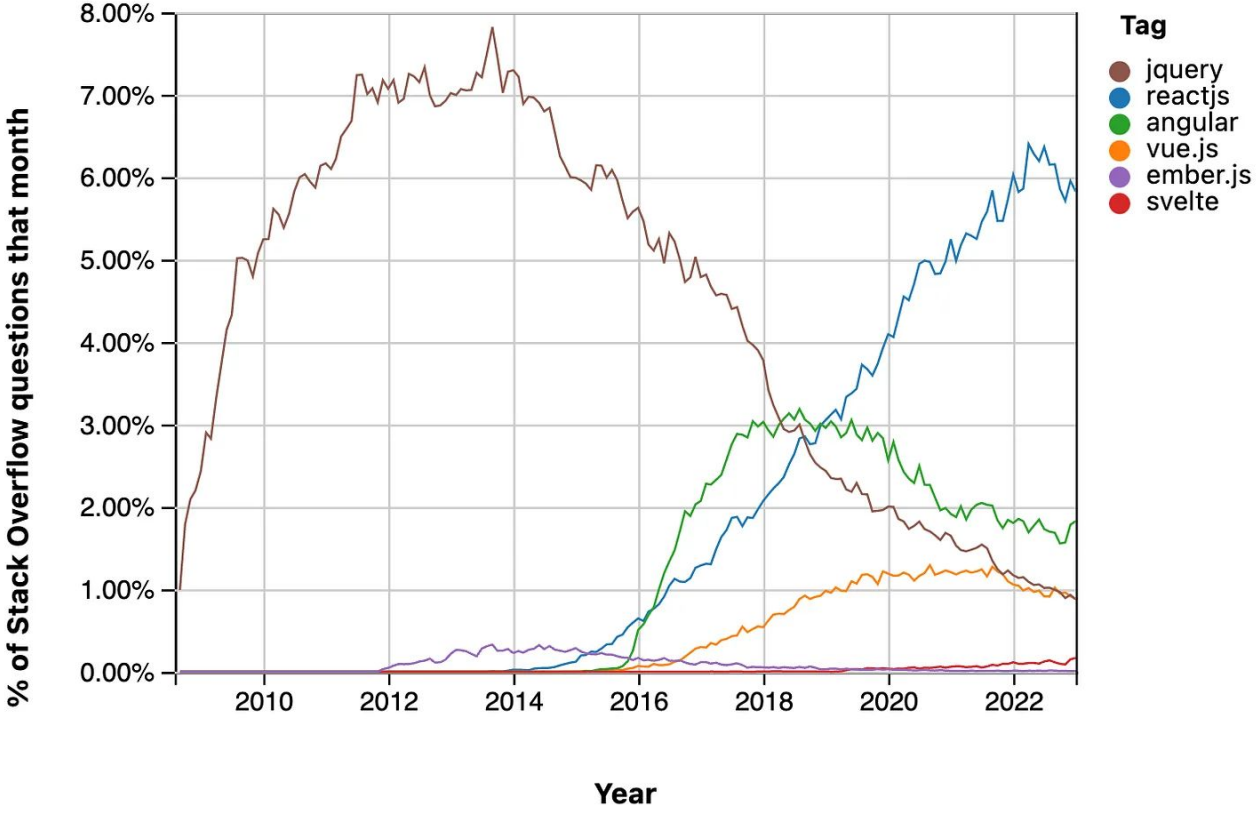
  A chaque fin de demi journée, les stagiaires devront répondre aux questions suivantes par l'intermédiaire d'un Google Form :

- Le rythme vous convient il ?
- L'équilibre théorie pratique vous convient il ?
- Les notions abordées jusque là sont elles acquises ?
- Y a t'il des notions sur lesquelles revenir avant de poursuivre le cours ?

BOOMERANG
CONSULTING

Chapitre 1 :
Histoire et Ecosystème

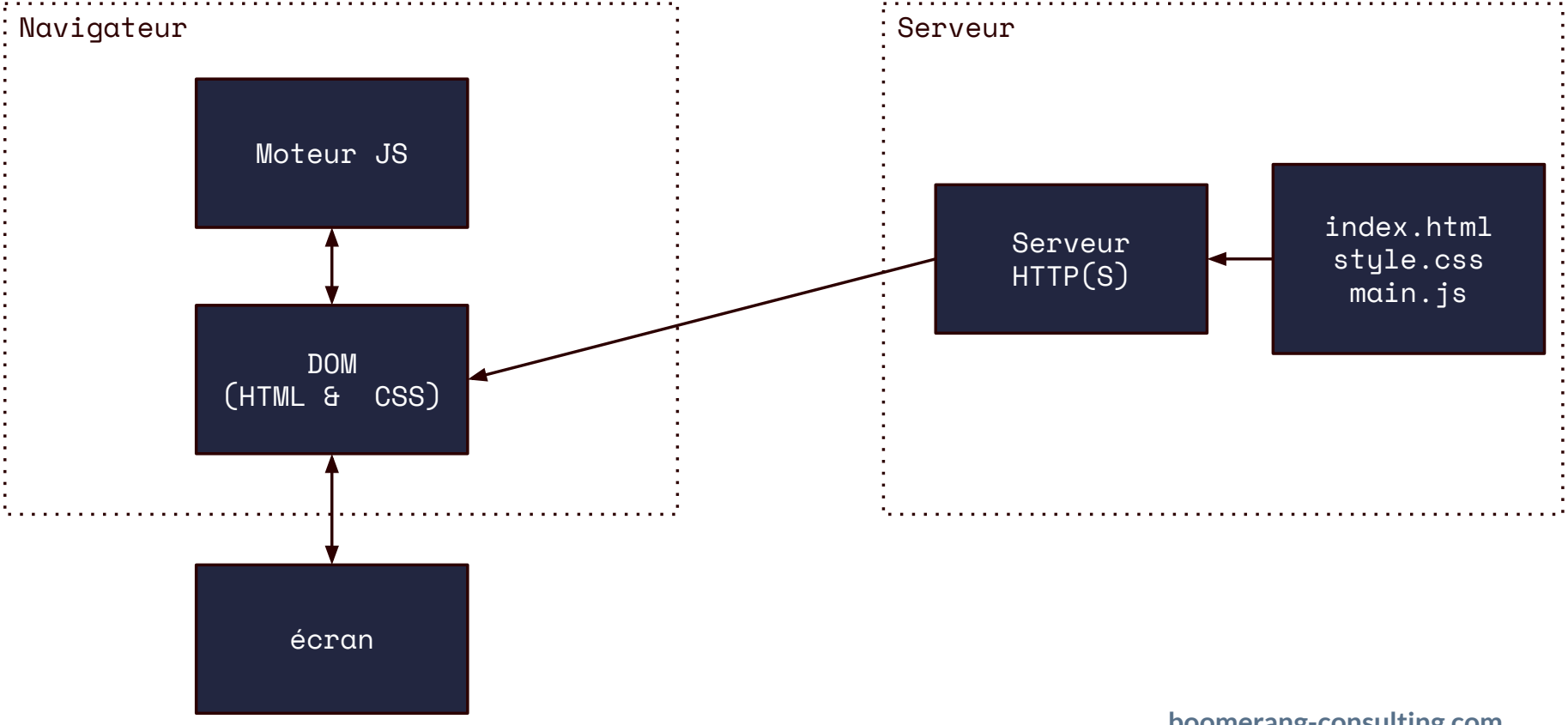
BOOMERANG
CONSULTING



BOOMERANG CONSULTING

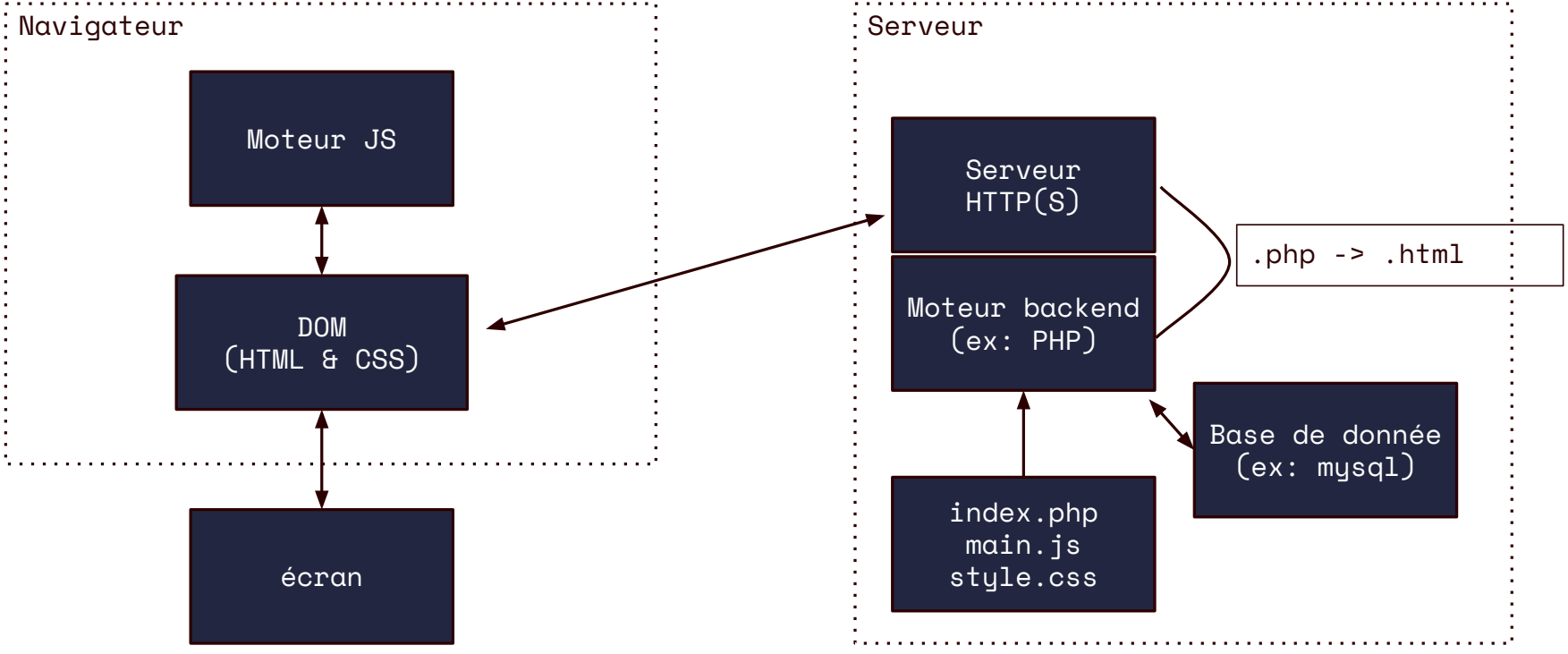
BOOMERANG CONSULTING

Site web statique



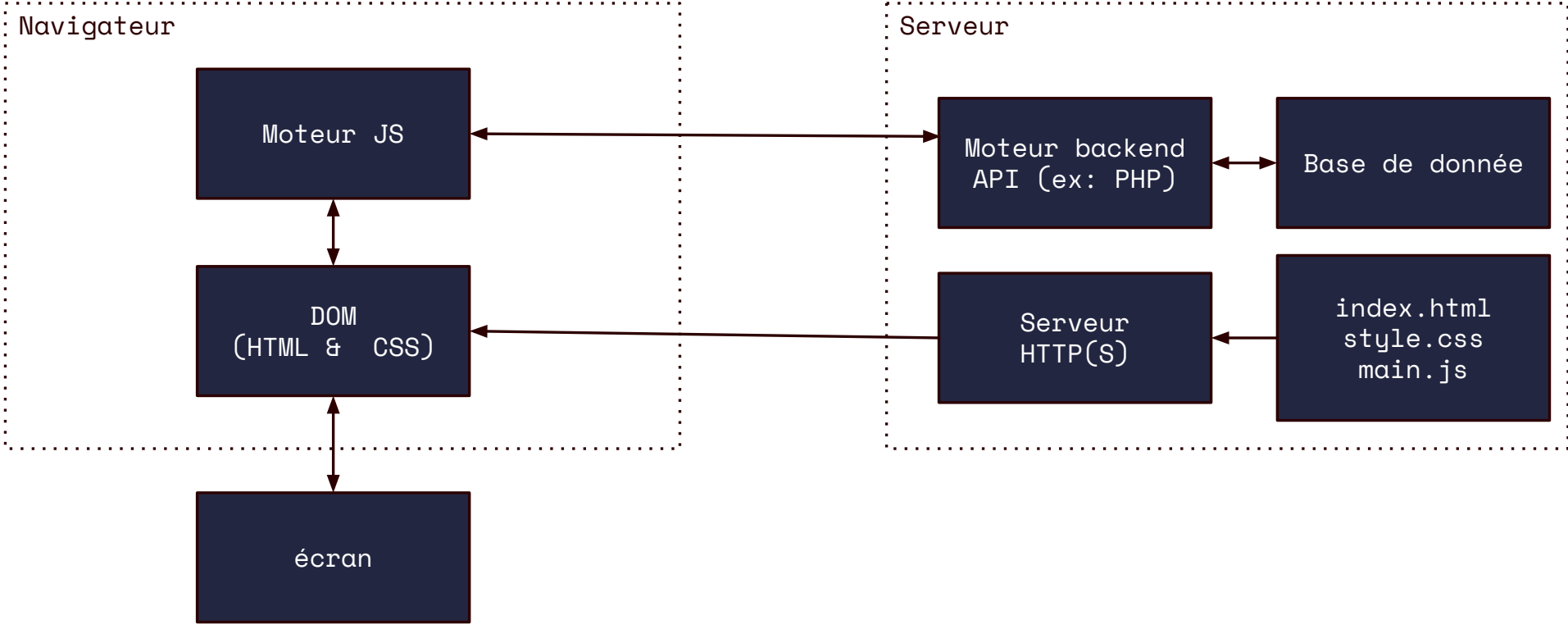
BOOMERANG CONSULTING

Site web dynamique (php/ROR/...)



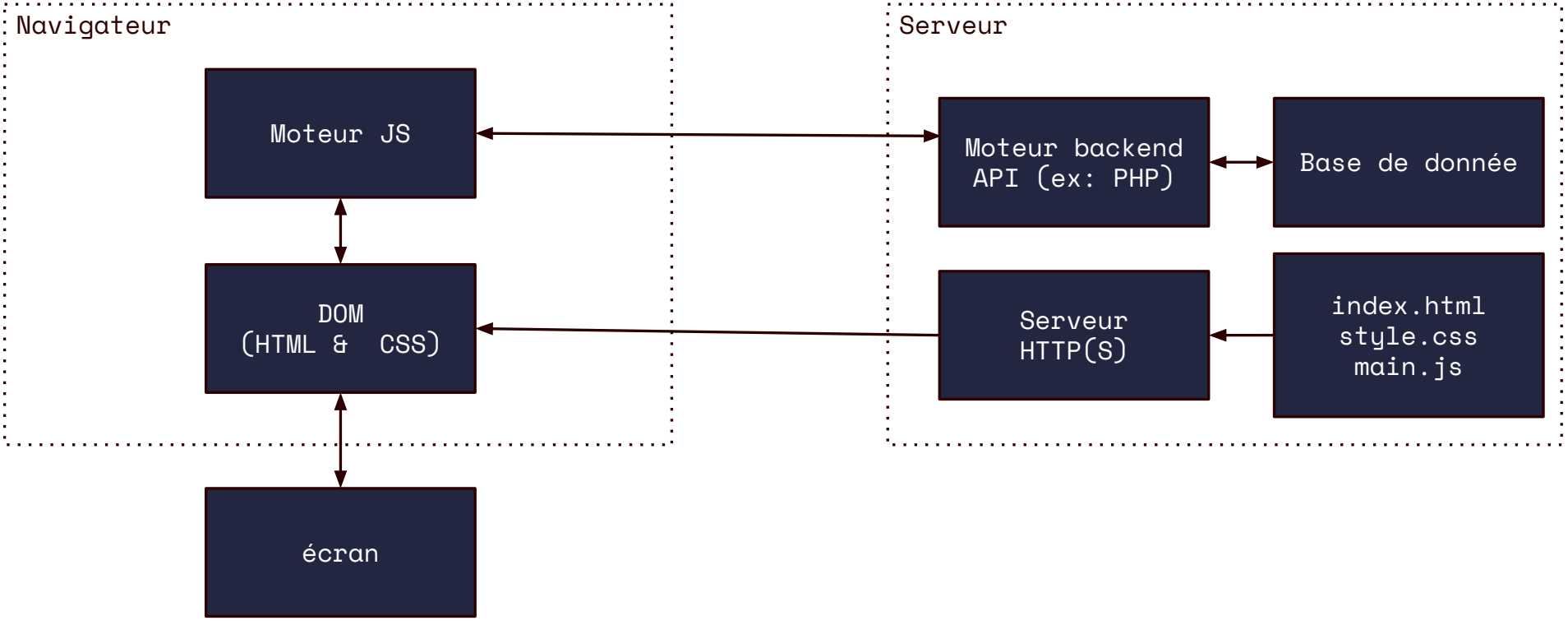
BOOMERANG CONSULTING

SPA



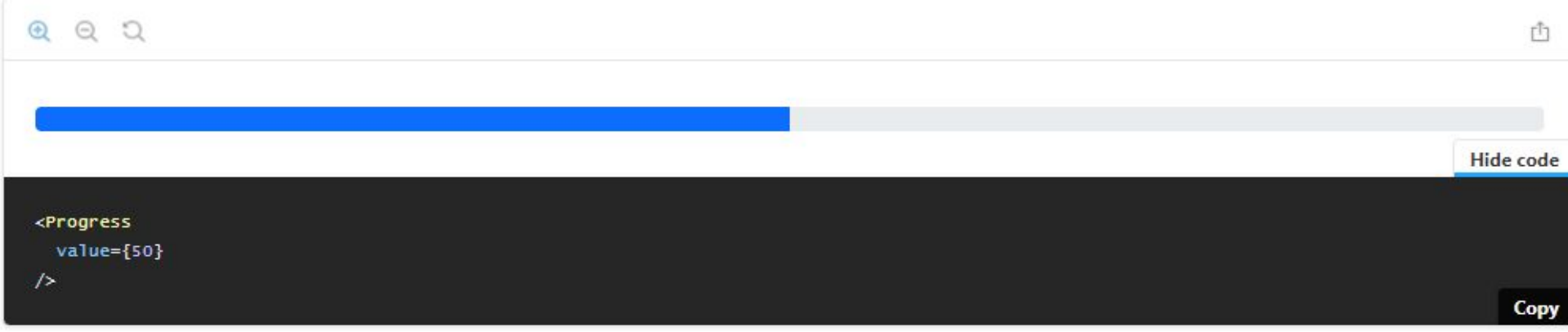
BOOMERANG CONSULTING

SSR



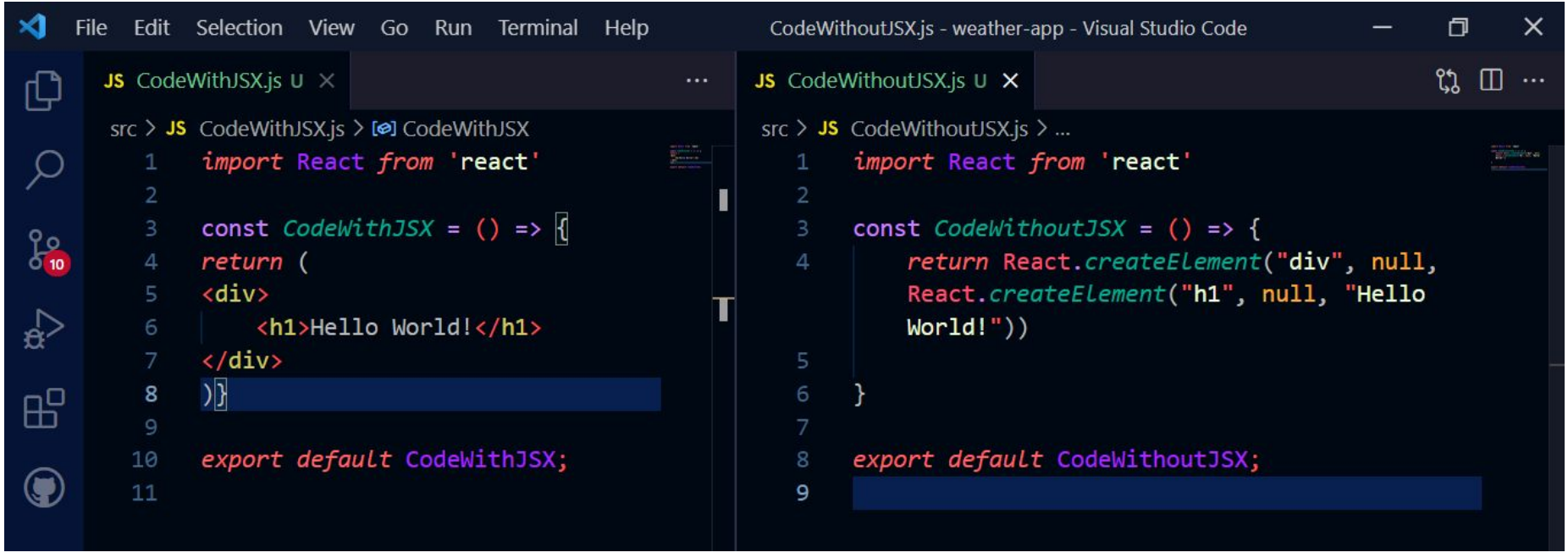
BOOMERANG
CONSULTING

Philosophie



BOOMERANG CONSULTING

JSX



L'environnement

BOOMERANG
CONSULTING

ES6

BOOMERANG CONSULTING

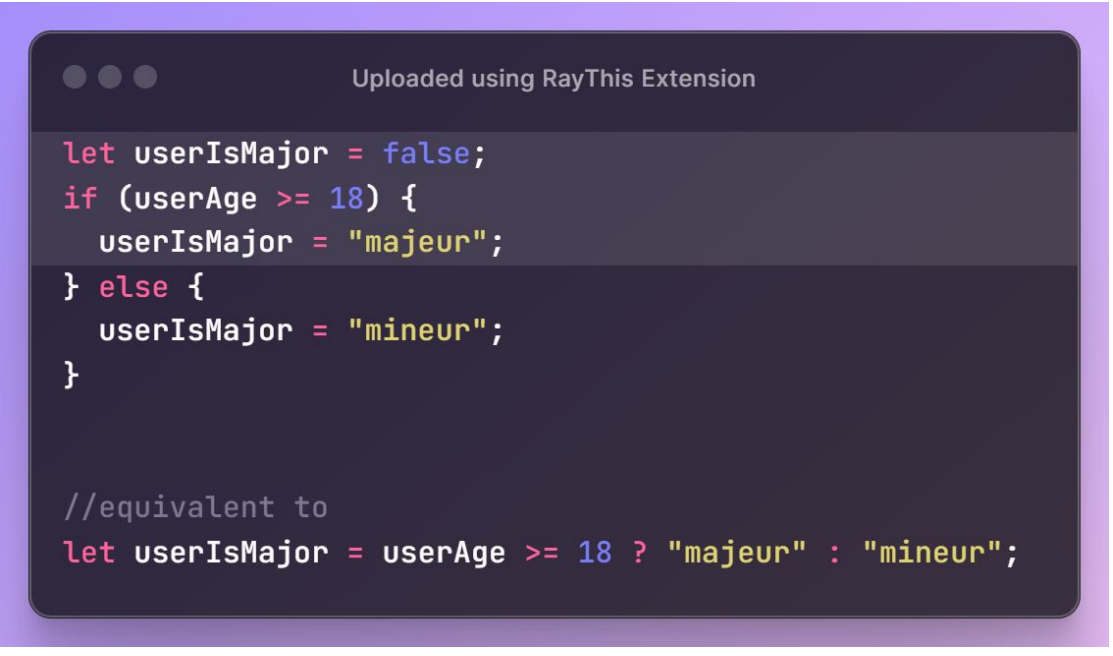
var vs let

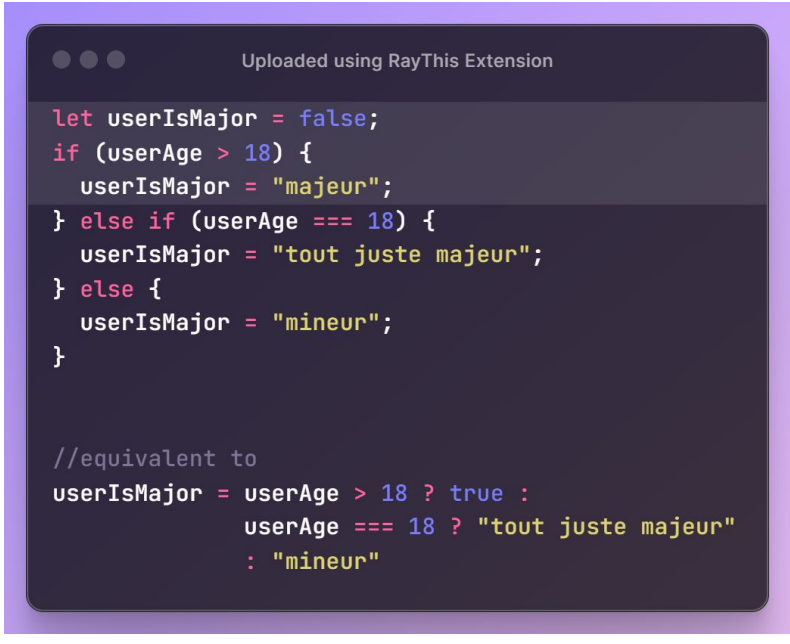
var vs let

```
function varTest() {
  var x = 1;
  if (true) {
    var x = 2;  // c'est la même variable !
    console.log(x);  // 2
  }
  console.log(x);  // 2
}

function letTest() {
  let x = 1;
  if (true) {
    let x = 2;  // c'est une variable différente
    console.log(x);  // 2
  }
  console.log(x);  // 1
}

varTest();
letTest();
```

BOOMERANG CONSULTING

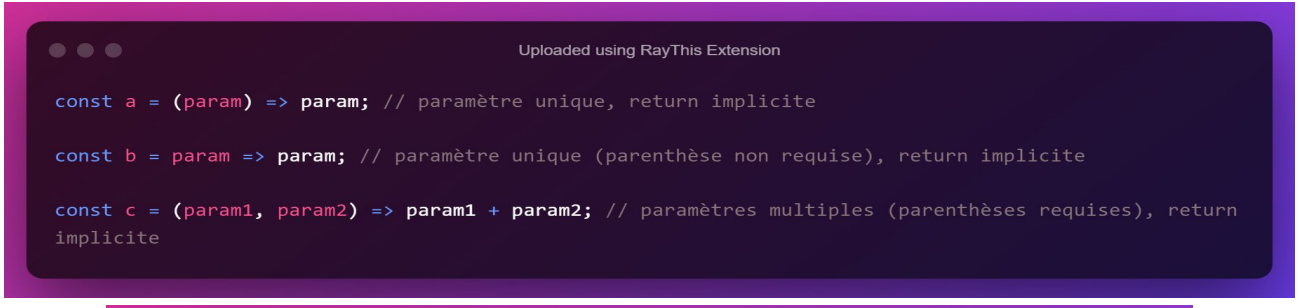
ternaires





BOOMERANG CONSULTING

arrow functions



BOOMERANG
CONSULTING

array functions

fonctions flêchées

```
//from this
function foo() {
  console.log("bar");
}

//to this
const foo = () => {
  console.log("bar");
}

//or this
const foo = () => console.log("bar);
```

array functions

```
//from this
const array = [1, 2, 3];

for (let i = 0; i < array.length; i++) {
  console.log(array[i]);
}

//to this
const array = [1, 2, 3];

array.map(i => console.log(i));
```

BOOMERANG CONSULTING

Destructuration & spread opérator

structuration

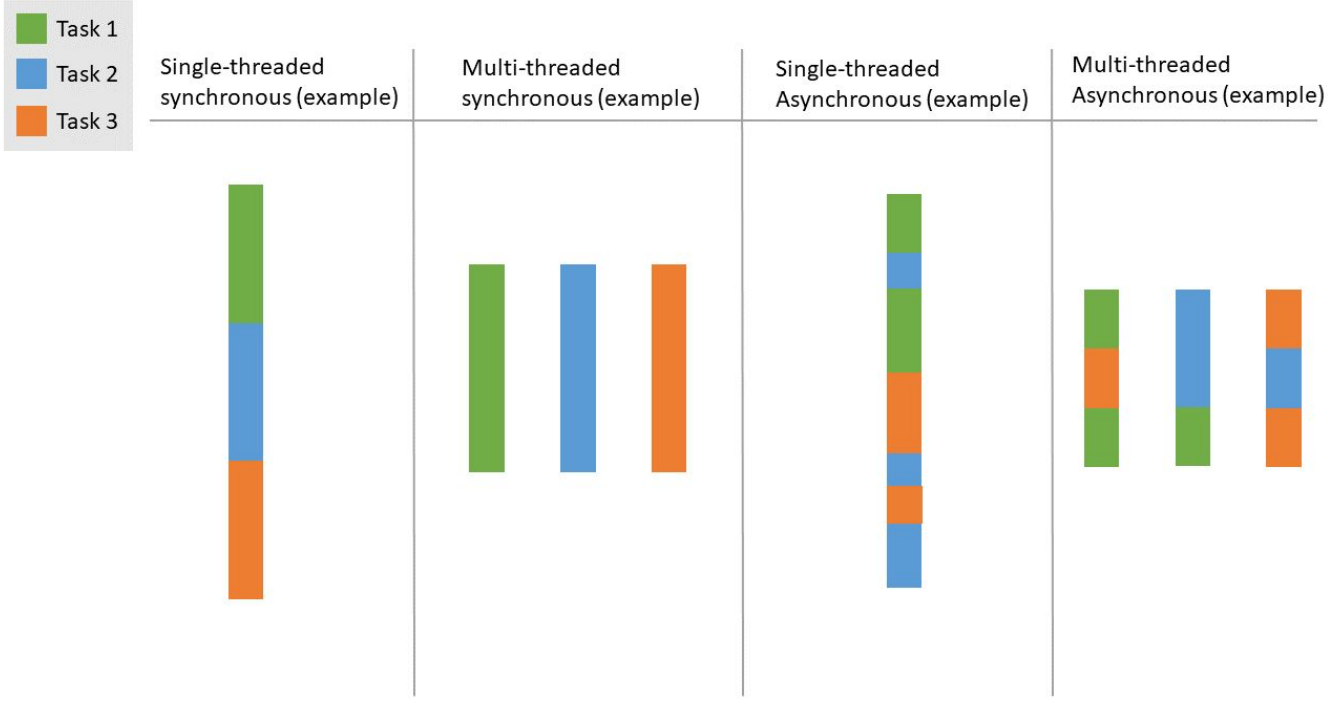
```
//array structuration
let arrayStructuration = [1, 2]
console.log(arrayStructuration); //[1, 2]

//array copy
let arrayCopy = [...arrayStructuration, 3, 4]
console.log(arrayCopy); //[1, 2, 3, 4]

//aray destructuration
let [a, b] = arrayCopy;
console.log(a);//1
console.log(b);//2
```

structuration

```
//object structuration
let obj = {
    message: "hello world"
}
console.log(obj); //{message: "hello world"}

//object add key
obj.type = "success";
console.log(obj); //{message: "hello world", type: "success"}

//object copy
let copy = {...obj}

//object destructuration
const {message} = obj;
console.log(message); //"hello world"
```

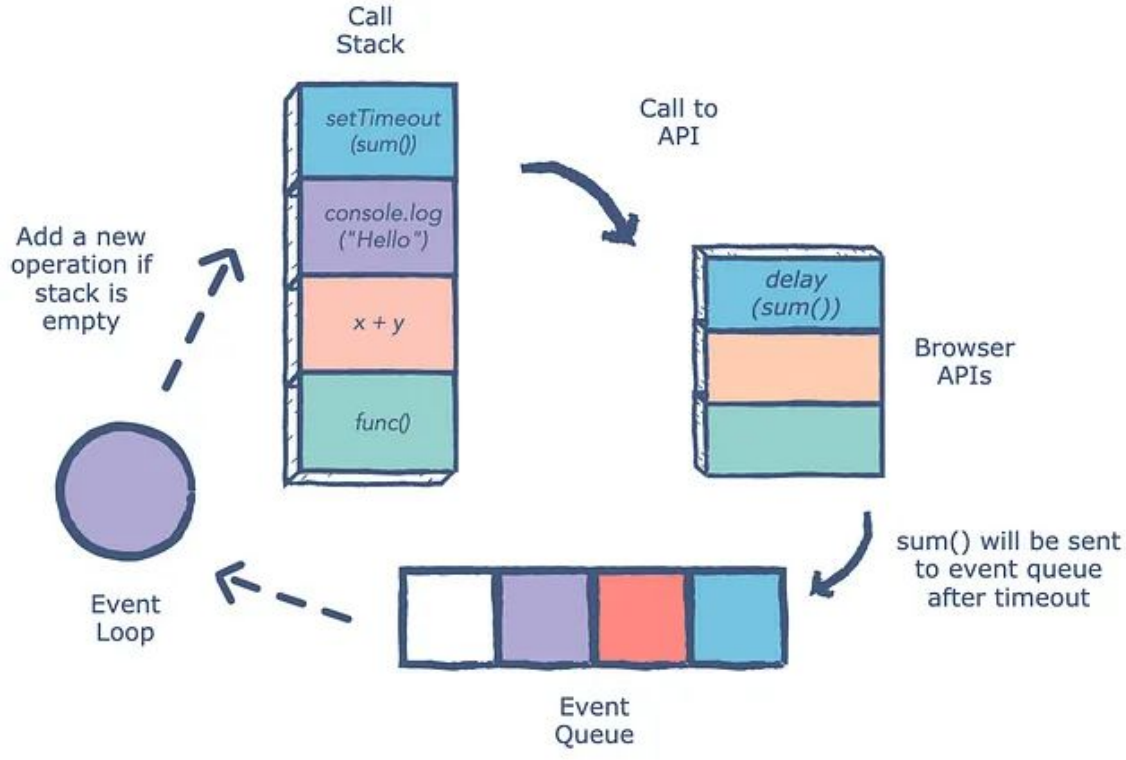
BOOMERANG CONSULTING

async



BOOMERANG CONSULTING

async



BOOMERANG CONSULTING

Asynchrone

axios.get(url)

BOOMERANG CONSULTING

resolve & reject

```
function asyncFunction() {
  return new Promise((resolve, reject) => {
    // Simulation d'une opération asynchrone
    setTimeout(() => {
      const randomNumber = Math.random();

      if (randomNumber < 0.5) {
        // Résoudre la promesse avec le nombre aléatoire
        resolve(randomNumber);
      } else {
        // Rejeter la promesse avec une erreur
        reject(new Error('Une erreur s\'est produite !'));
      }
    }, 1000);
  });
}
```

```
// Utilisation de try/catch pour capturer l'erreur
async function executeAsyncFunction() {
  try {
    const result = await asyncFunction();
    console.log('Résultat :', result);
  } catch (error) {
    console.error('Erreur :', error.message);
  }
}
```

```
// Utilisation de .catch pour capturer l'erreur
asyncFunction()
  .then(result => {
    console.log('Résultat :', result);
  })
  .catch(error => {
    console.error('Erreur :', error.message);
  });
```

BOOMERANG CONSULTING

Parallèlisme

Waterfall

```
async function asyncOperation1() { /* ... */ }
async function asyncOperation2() { /* ... */ }
async function asyncOperation3() { /* ... */ }

async function executeAsyncOperations() {
  try {
    const result1 = await asyncOperation1();
    const result2 = await asyncOperation2(result1);
    const result3 = await asyncOperation3(result2);
    console.log('Résultat final :', result3);
  } catch (error) {
    console.error('Erreur :', error);
  }
}

executeAsyncOperations()
```

Parallel

```
async function asyncOperation1() { /* ... */ }
async function asyncOperation2() { /* ... */ }
async function asyncOperation3() { /* ... */ }

async function executeParallelOperations() {
  try {
    const [result1, result2, result3] = await Promise.all([
      asyncOperation1(),
      asyncOperation2(),
      asyncOperation3()
    ]);
    console.log('Résultat 1 :', result1);
    console.log('Résultat 2 :', result2);
    console.log('Résultat 3 :', result3);
  } catch (error) {
    console.error('Erreur :', error);
  }
}

executeParallelOperations();
```

Parallel

```
const asyncQueue = [];

function addToQueue(asyncTask) {
  asyncQueue.push(asyncTask);
  if (asyncQueue.length === 1) {
    processQueue();
  }
}

function processQueue() {
  const asyncTask = asyncQueue[0];
  asyncTask()
    .then(result => {
      console.log('Résultat de la tâche :', result);
      asyncQueue.shift();
      if (asyncQueue.length > 0) {
        processQueue();
      }
    })
    .catch(error => {
      console.error('Erreur de la tâche :', error);
      asyncQueue.shift();
      if (asyncQueue.length > 0) {
        processQueue();
      }
    });
}

// Exemple d'utilisation
addToQueue(asyncOperation1);
addToQueue(asyncOperation2);
```

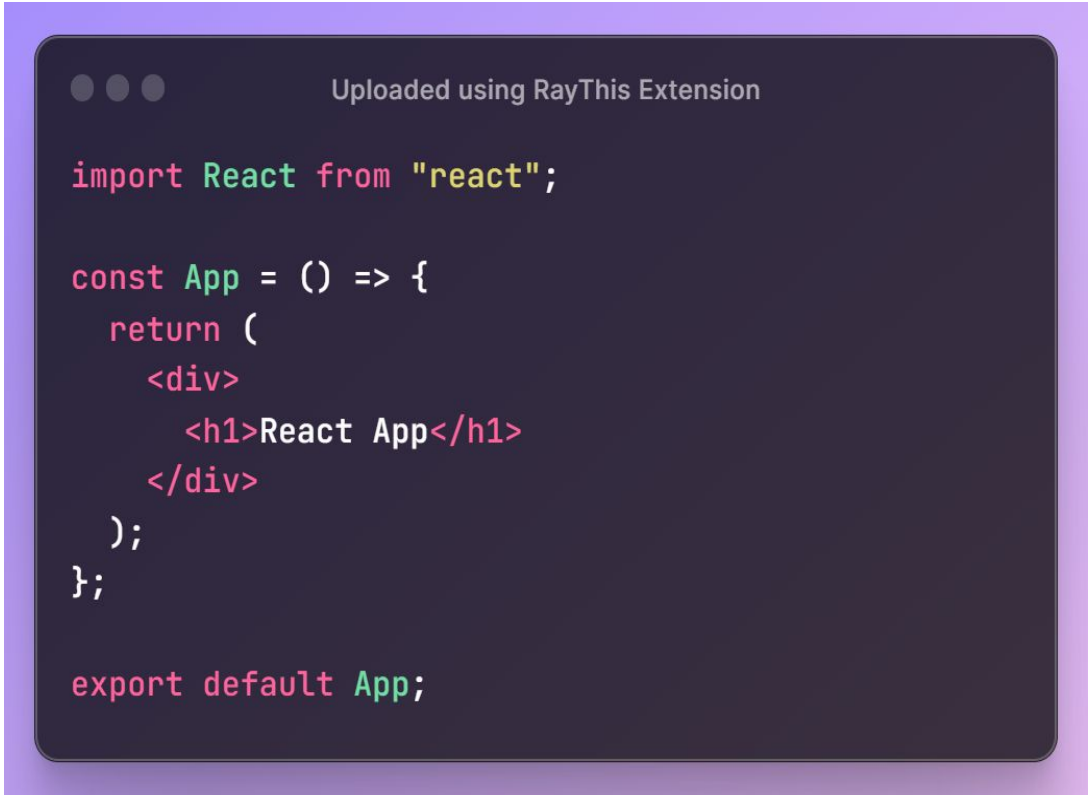
BOOMERANG CONSULTING

Vite

```
PS C:\Users\guili\Desktop> npm create vite@latest
Need to install the following packages:
create-vite@5.1.0
Ok to proceed? (y) y
√ Project name: ... demo-react
√ Select a framework: » React
√ Select a variant: » JavaScript

Scaffolding project in C:\Users\guili\Desktop\demo-react...

Done. Now run:

  cd demo-react
  npm install
  npm run dev

PS C:\Users\guili\Desktop> cd .\demo-react\
PS C:\Users\guili\Desktop\demo-react> npm install
```

BOOMERANG CONSULTING

JSX



BOOMERANG
CONSULTING

Uploaded using RayThis Extension

```
import { createContext, useContext, useState } from "react";

const ctx = createContext();

const Provider = ({children}) => {
  const [state, setState] = useState(0);

  return (
    <ctx.Provider value={[state, setState]}>
      {children}
    </ctx.Provider>
  )
}

const useCtx = () => {
  const [state, setState] = useContext(ctx);

  return [state, setState];
}

export {Provider};
export default useCtx;
```

Uploaded using RayThis Extension

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'
import {Provider} from './contexts/ctx.jsx';

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <Provider>
      <App />
    </Provider>
  </React.StrictMode>,
)
```

Uploaded using RayThis Extension

```
import Button from "./Button";
import useCtx from "./contexts/ctx";

function App() {
  const [state, setState] = useCtx();


  return (
    <>
      <Button onClick={() => setState(state+1)}>click me: {state}</Button>
    </>
  )
}

export default App
```

BOOMERANG
CONSULTING

Découpage en composants

Uploaded using RayThis Extension

```
import React from "react";
import Component from "./Component";

const App = () => {
  return (
    <div>
      <Component />
    </div>
  );
};

export default App;
```

Uploaded using RayThis Extension

```
import React from "react";

const Component = () => {
  return (
    <div>
      <h1>My first component</h1>
    </div>
  );
};

export default Component;
```

BOOMERANG
CONSULTING

props

Uploaded using RayThis Extension

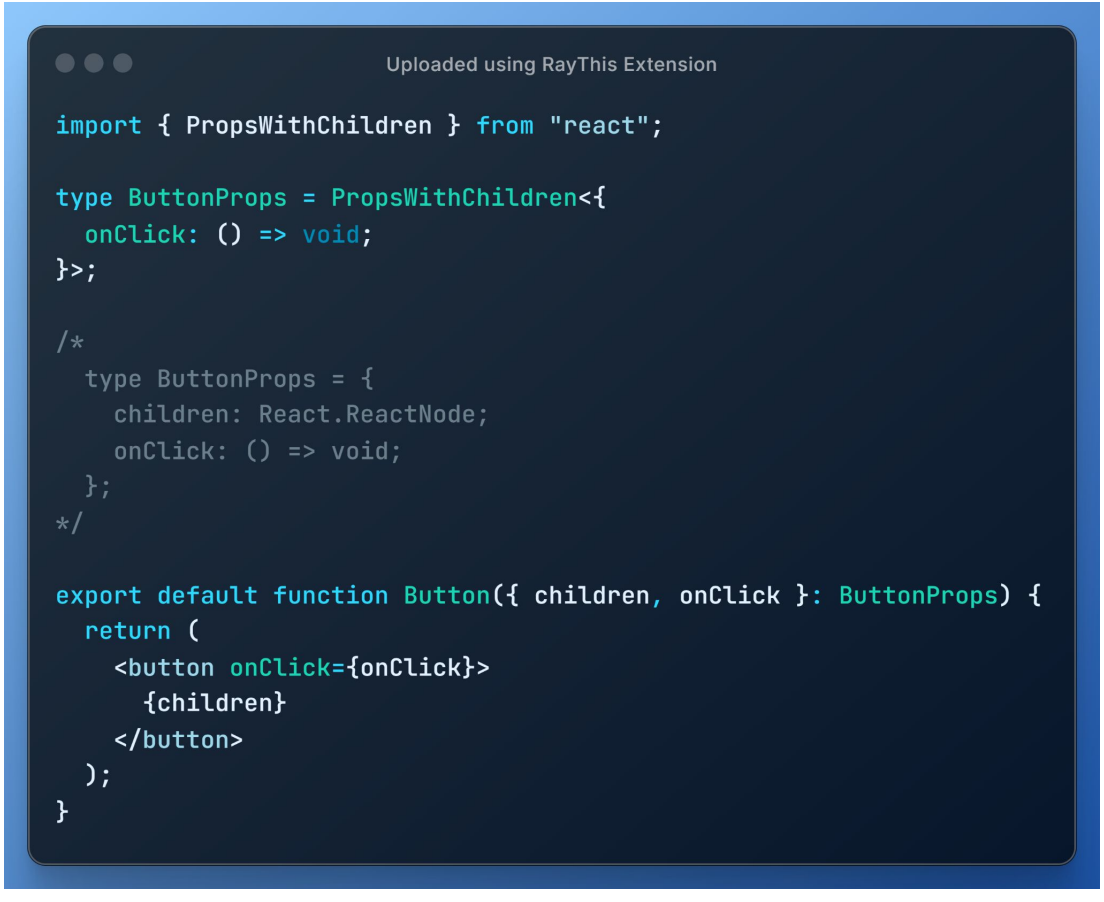
```
import React from "react";
import Component from "./Component";

const App = () => {
  return (
    <div>
      <Component title="hello world" />
    </div>
  );
};

export default App;
```

Uploaded using RayThis Extension

```
import React from "react";

const Component = (props) => {
  const { title } = props;

  return (
    <div>
      <h1>{title}</h1>
    </div>
  );
};

export default Component;
```

BOOMERANG CONSULTING

props with TS



BOOMERANG CONSULTING

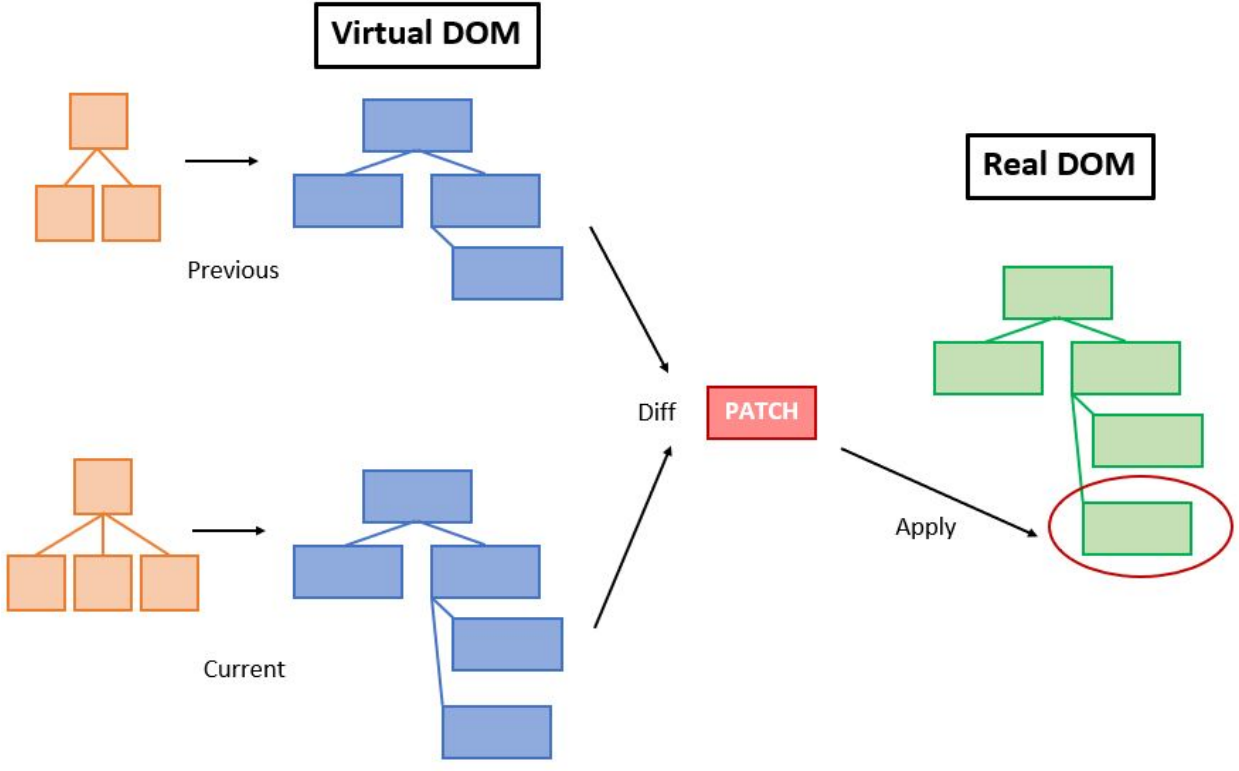
Gestion d'évènements

Uploaded using RayThis Extension

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

export default function EventPage() {
  return (
    <div>
      <button onClick={() => alert("hello world")}>Click me</button>
    </div>
  );
}
```

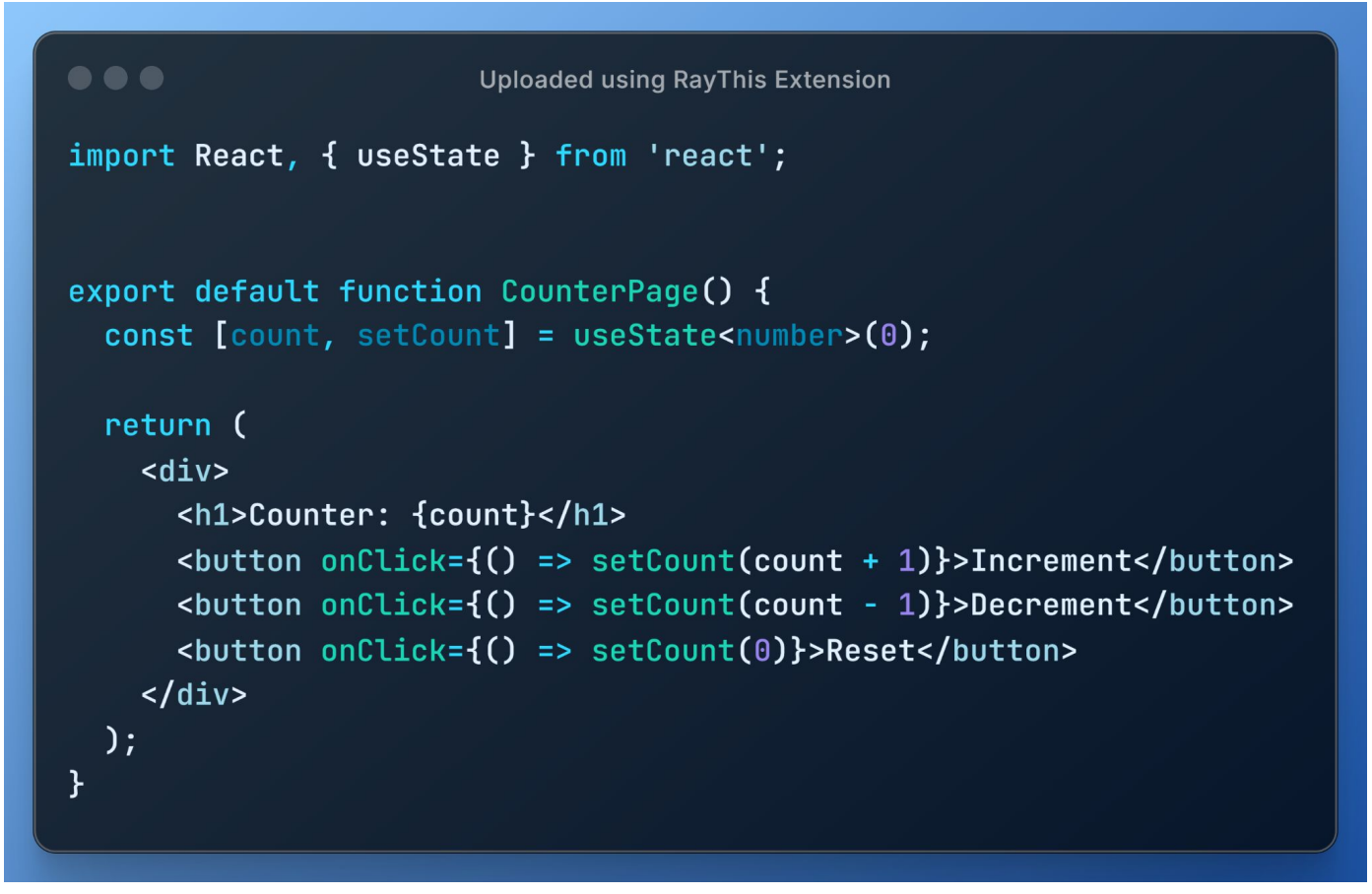
BOOMERANG CONSULTING

DOM Virtuel



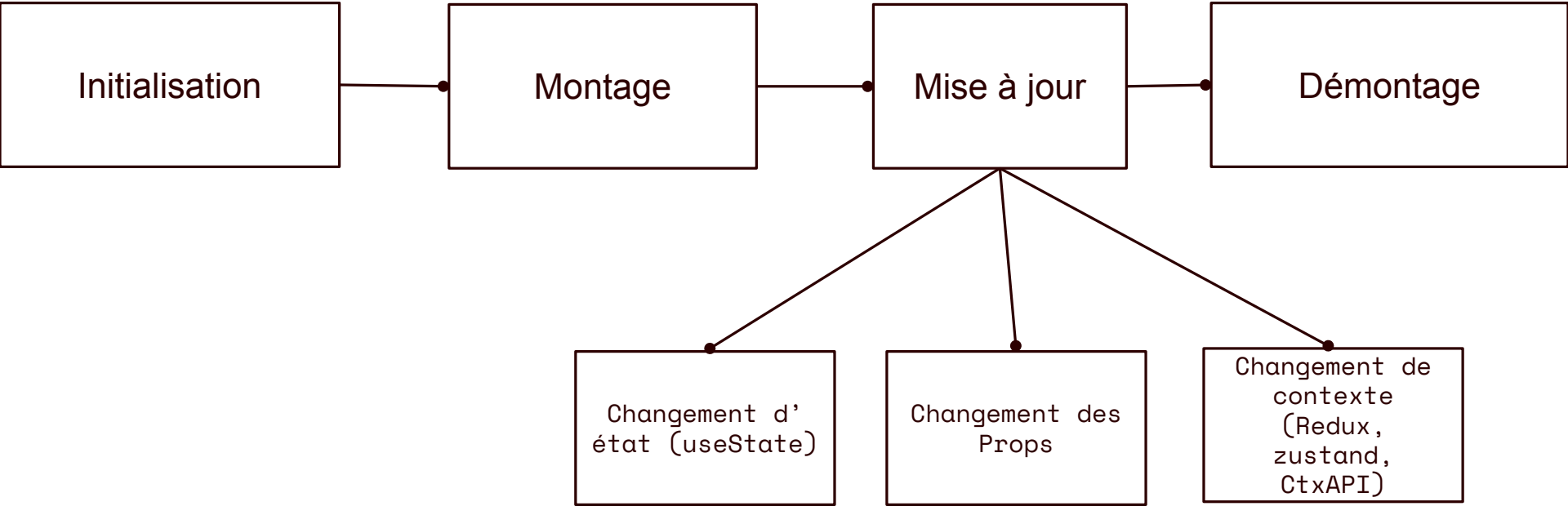
BOOMERANG CONSULTING

Gestion d'état



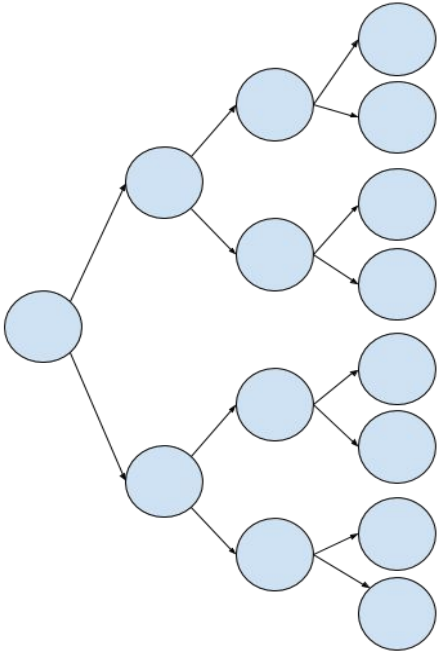
BOOMERANG CONSULTING

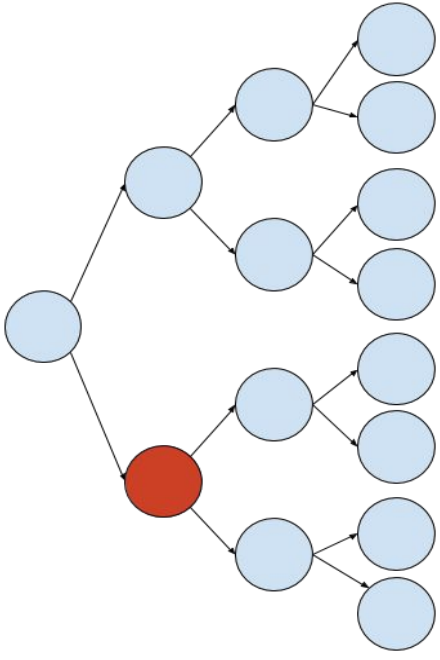
Cycle de vie



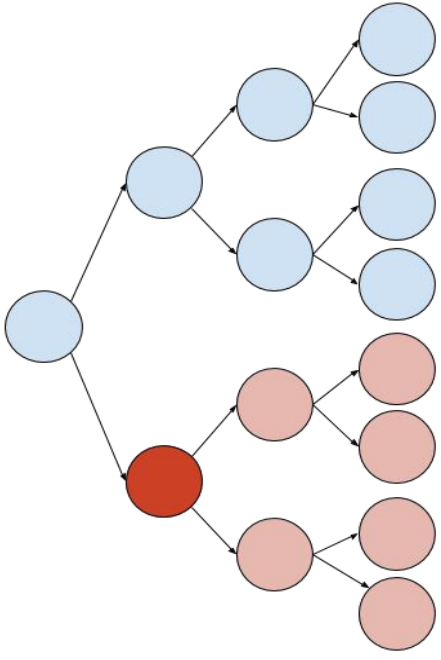
BOOMERANG CONSULTING

Cycle de vie





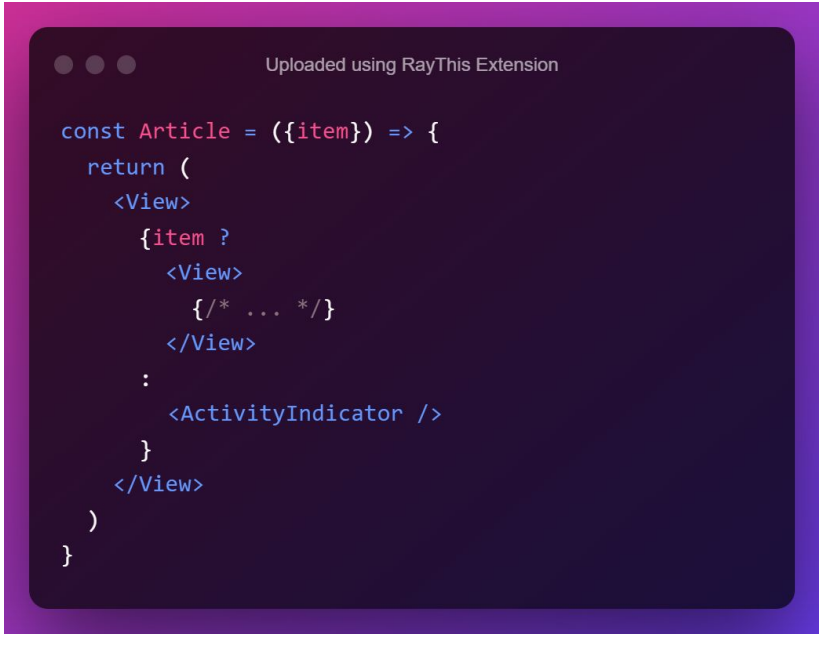
Changement d'état ou props



Propagation aux enfants (children)

BOOMERANG CONSULTING

Conditional rendering



BOOMERANG CONSULTING

Conditional rendering

BOOMERANG CONSULTING

Array rendering

BOOMERANG CONSULTING

2way data binding

Uploaded using RayThis Extension

```
import React, {useState} from "react";
import styles from "./App.module.scss";

const App = () => {
  const [name, setName] = useState("");

  return (
    <div className={styles.container}>
      <input
        type="text"
        value={name}
        onChange={(e) => setName(e.target.value)}
      />
      <p>{name}</p>
    </div>
  );
};

export default App;
```

BOOMERANG CONSULTING

Effet de bords et abonnements

Mount

```
useEffect(() => {

}, [])
```

Set "[]" empty dependency array to run an effect only on "mount"

Update

```
useEffect(() => {

})
```

Do not pass a dependency array at all to run an effect on each component update

```
useEffect(() => {

}, [dependencies])
```

Set dependencies array to run an effect only if any dependency change

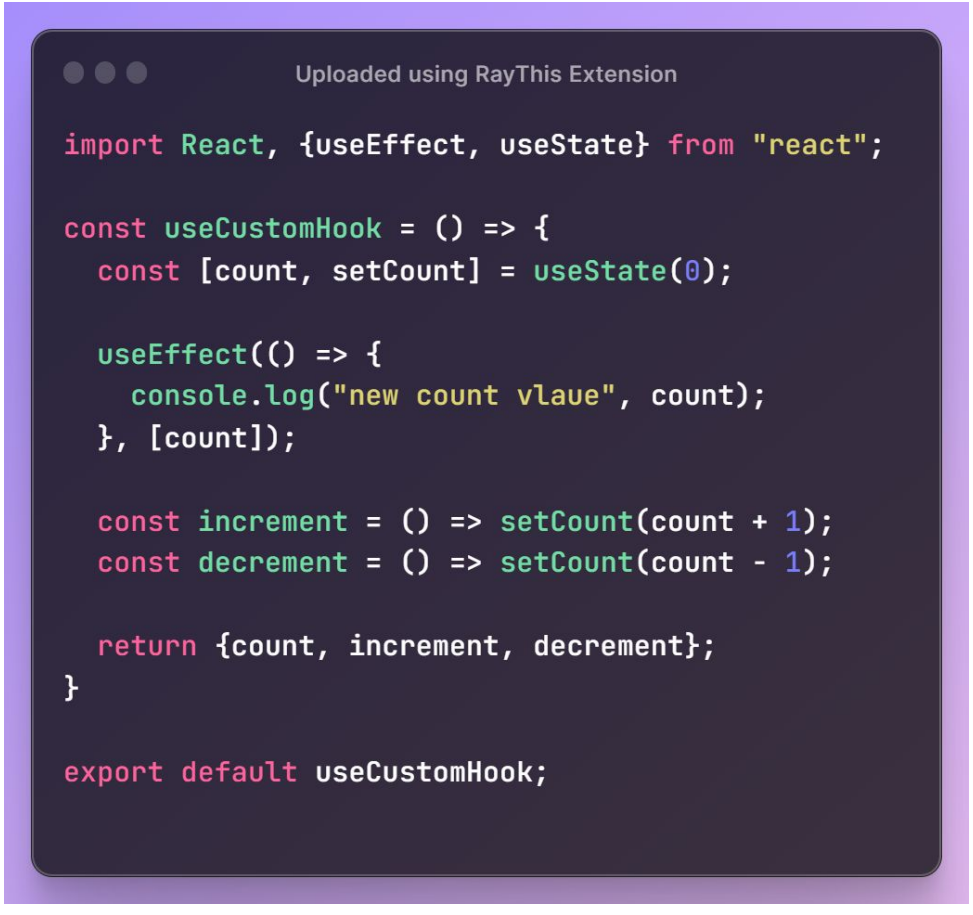
Unmount

```
useEffect(() => {
  return () => {

  }
}, [])
```

Set return function to run on component unmount lifecycle.

BOOMERANG CONSULTING

Hooks customs



BOOMERANG CONSULTING

hook customs

pas de hook conditionnel

commence par “use”

Nettoyez vos effets

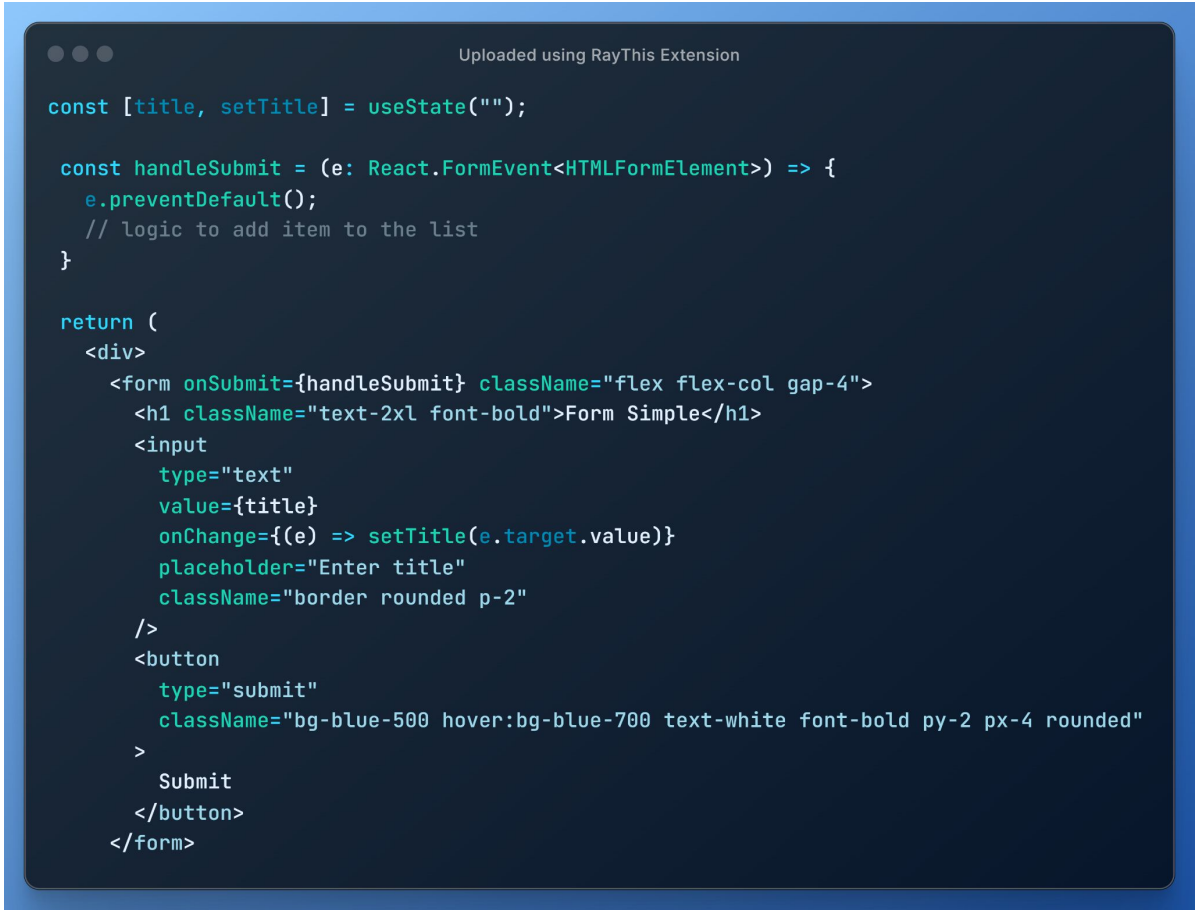
préférer les retours cassants

Découper en hook custom

Favoriser le code réutilisable

BOOMERANG CONSULTING

Formulaires



BOOMERANG CONSULTING

Formulaires

Uploaded using RayThis Extension

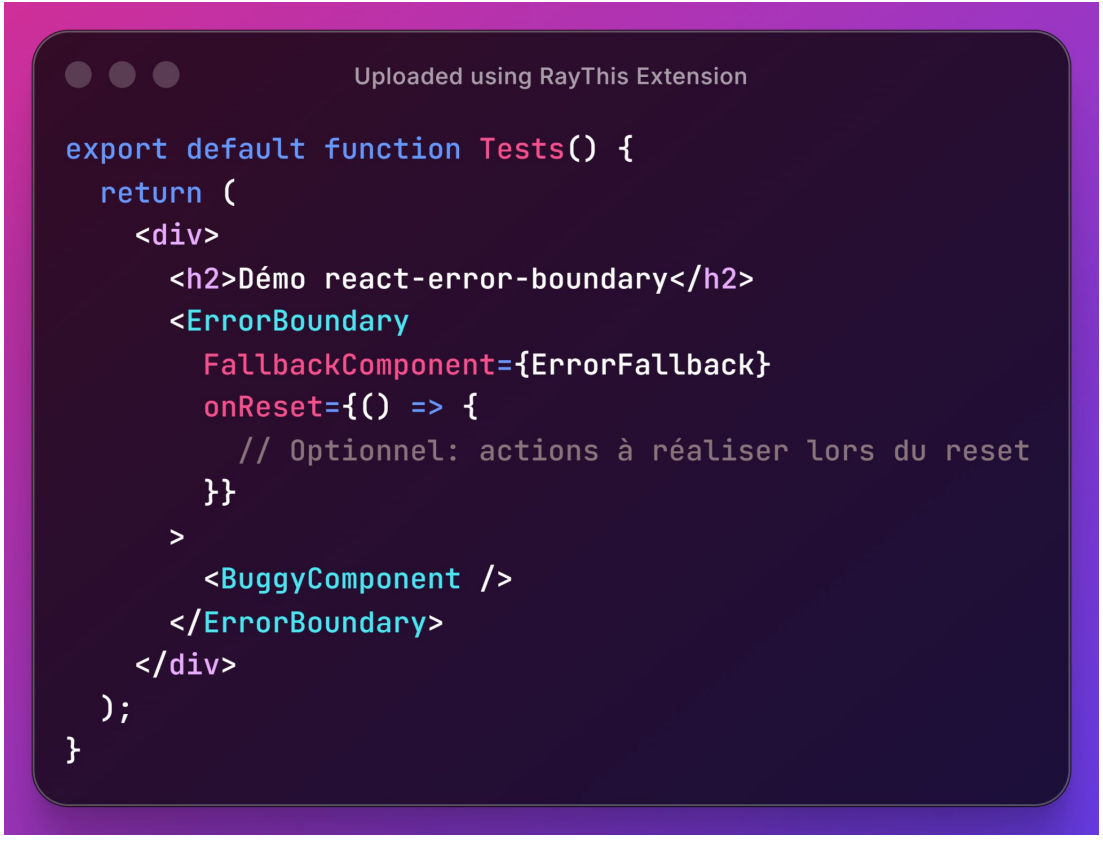
```zod
import { z, ZodType } from "zod"; // Add new import

export const UserSchema: ZodType<FormData> = z
 .object({
   email: z.string().email(),
   githubUrl: z
     .string()
     .url()
     .includes("github.com", { message: "Invalid GitHub URL" }),
   yearsOfExperience: z
     .number({
       required_error: "required field",
       invalid_type_error: "Years of Experience is required",
     })
     .min(1)
     .max(10),
   password: z
     .string()
     .min(8, { message: "Password is too short" })
     .max(20, { message: "Password is too long" }),
   confirmPassword: z.string(),
 })
 .refine((data) => data.password === data.confirmPassword, {
   message: "Passwords do not match",
   path: ["confirmPassword"], // path of error
 });
```

Uploaded using RayThis Extension

```tsx
function Form() {
  const {
    register,
    handleSubmit,
    formState: { errors },
    setError,
  } = useForm<FormData>();

  const onSubmit = async (data: FormData) => {
      console.log("SUCCESS", data);
  }

  return (
      <form onSubmit={handleSubmit(onSubmit)}>
        <div className="grid col-auto">
          <h1 className="text-3xl font-bold mb-4">
            Zod & React-Hook-Form
          </h1>
          <FormField
            type="email"
            placeholder="Email"
            name="email"
            register={register}
            error={errors.email}
          />
```

BOOMERANG CONSULTING

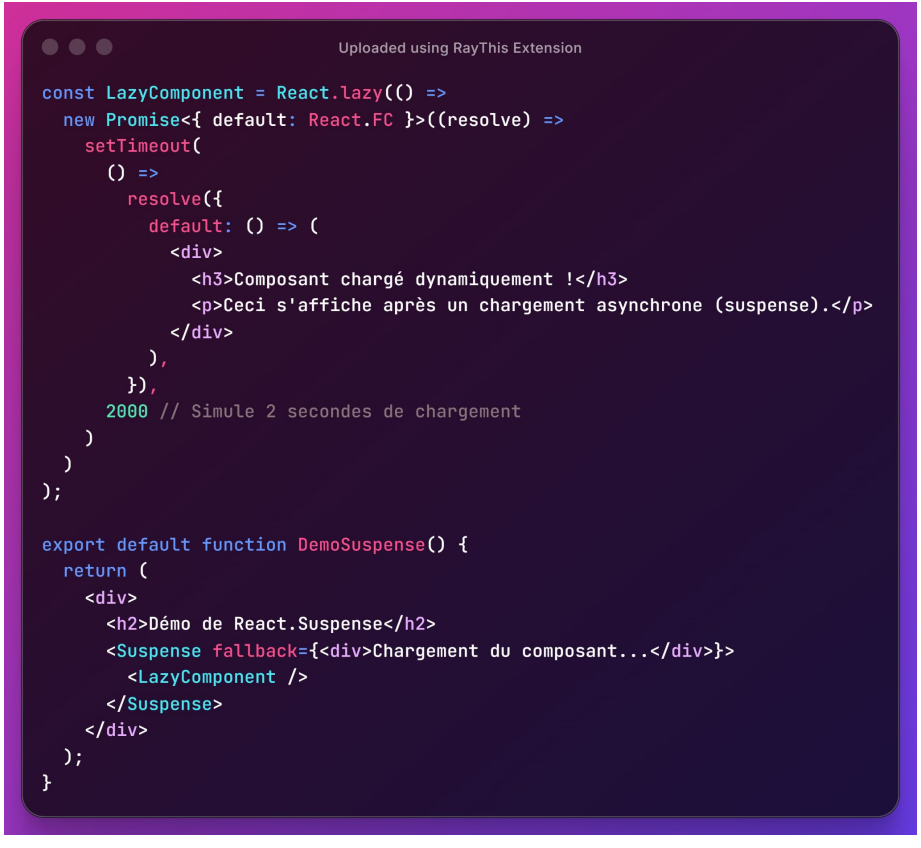
ErrorBoundary



npm i
react-error-boundary

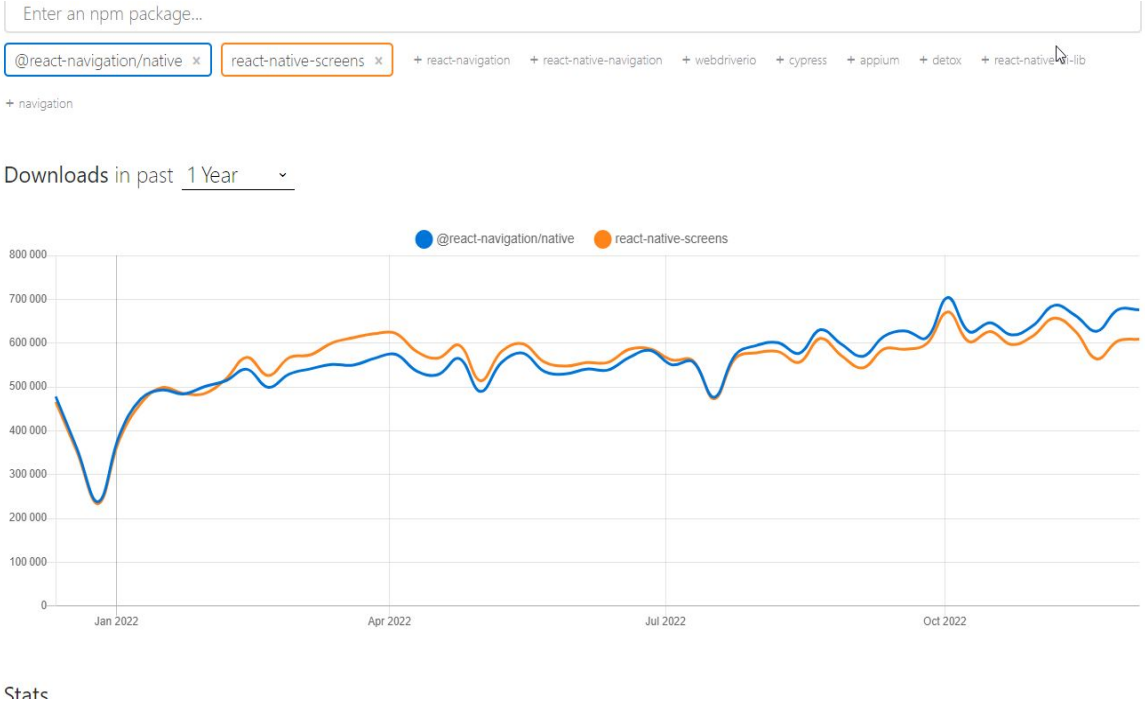
BOOMERANG CONSULTING

Suspense



BOOMERANG CONSULTING

npm & dépendances externes



BOOMERANG CONSULTING

Routing côté client

```
import Link from "next/link";


export default function HomePage() {
  return (
  <Link href="/about">
    About page
  </Link>
  );
}
```

BOOMERANG CONSULTING

React Hook Form + zod

Uploaded using RayThis Extension

```ts
import { z, ZodType } from "zod"; // Add new import

export const UserSchema: ZodType<FormData> = z
 .object({
   email: z.string().email(),
   githubUrl: z
     .string()
     .url()
     .includes("github.com", { message: "Invalid GitHub URL" }),
   yearsOfExperience: z
     .number({
       required_error: "required field",
       invalid_type_error: "Years of Experience is required",
     })
     .min(1)
     .max(10),
   password: z
     .string()
     .min(8, { message: "Password is too short" })
     .max(20, { message: "Password is too long" }),
   confirmPassword: z.string(),
 })
 .refine((data) => data.password === data.confirmPassword, {
   message: "Passwords do not match",
   path: ["confirmPassword"], // path of error
 });
```

Uploaded using RayThis Extension

```tsx
function Form() {
  const {
    register,
    handleSubmit,
    formState: { errors },
    setError,
  } = useForm<FormData>();

  const onSubmit = async (data: FormData) => {
      console.log("SUCCESS", data);
  }

  return (
      <form onSubmit={handleSubmit(onSubmit)}>
        <div className="grid col-auto">
          <h1 className="text-3xl font-bold mb-4">
            Zod & React-Hook-Form
          </h1>
          <FormField
            type="email"
            placeholder="Email"
            name="email"
            register={register}
            error={errors.email}
          />
```

État complexe: useReducer

Uploaded using RayThis Extension

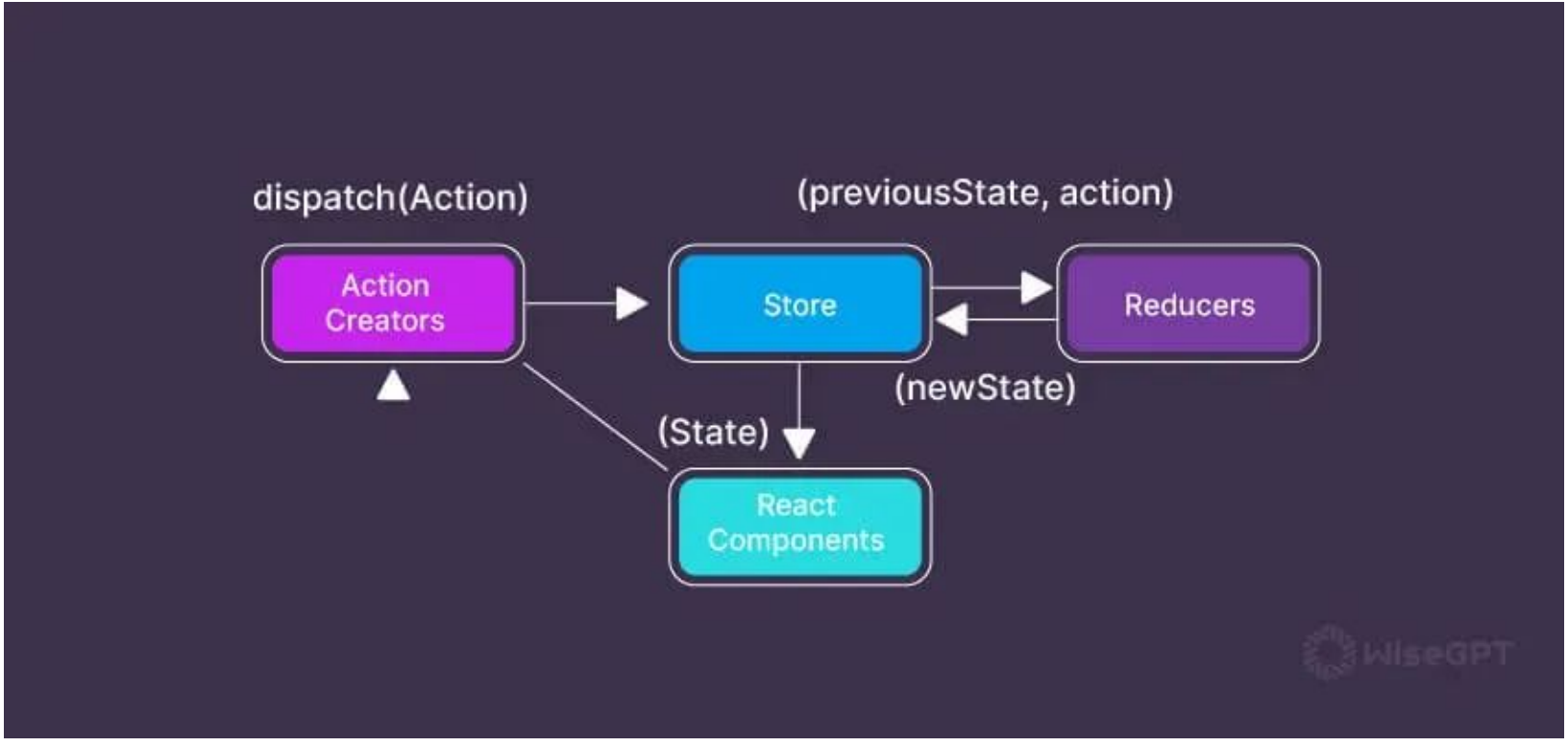
```
import { useReducer } from "react";

function cartReducer(state, action) {
  switch (action.type) {
    case "ADD_ITEM":
      return [...state, action.item];
    case "REMOVE_ITEM":
      return state.filter((item) => item.id !== action.id);
    case "CLEAR_CART":
      return [];
    default:
      return state;
  }
}
```

Uploaded using RayThis Extension

```
export default function Cart() {
  const [cart, dispatch] = useReducer(cartReducer, []);

  return (
    <>
      <button
        onClick={() =>
          dispatch({ type: "ADD_ITEM", item: { id: 1, name: "Parfum" } })
        }
      >
        Ajouter un parfum
      </button>

      <ul>
        {cart.map((item) => (
          <li key={item.id}>
            {item.name}
            <button
              onClick={() => dispatch({ type: "REMOVE_ITEM", id: item.id })}
            >
              Supprimer
            </button>
          </li>
        ))}
      </ul>

      <button onClick={() => dispatch({ type: "CLEAR_CART" })}>Vider</button>
    </>
  );
}
```

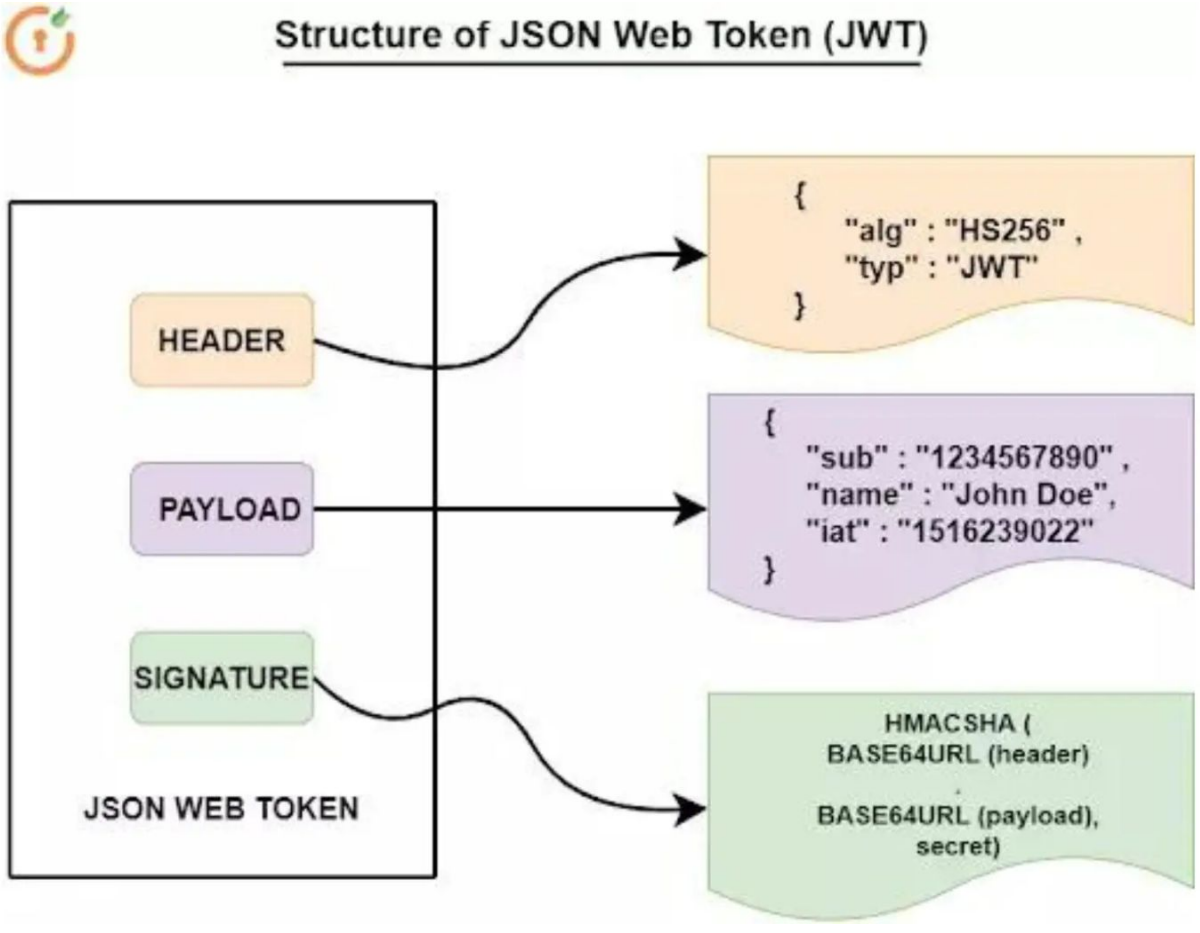
BOOMERANG CONSULTING

Redux



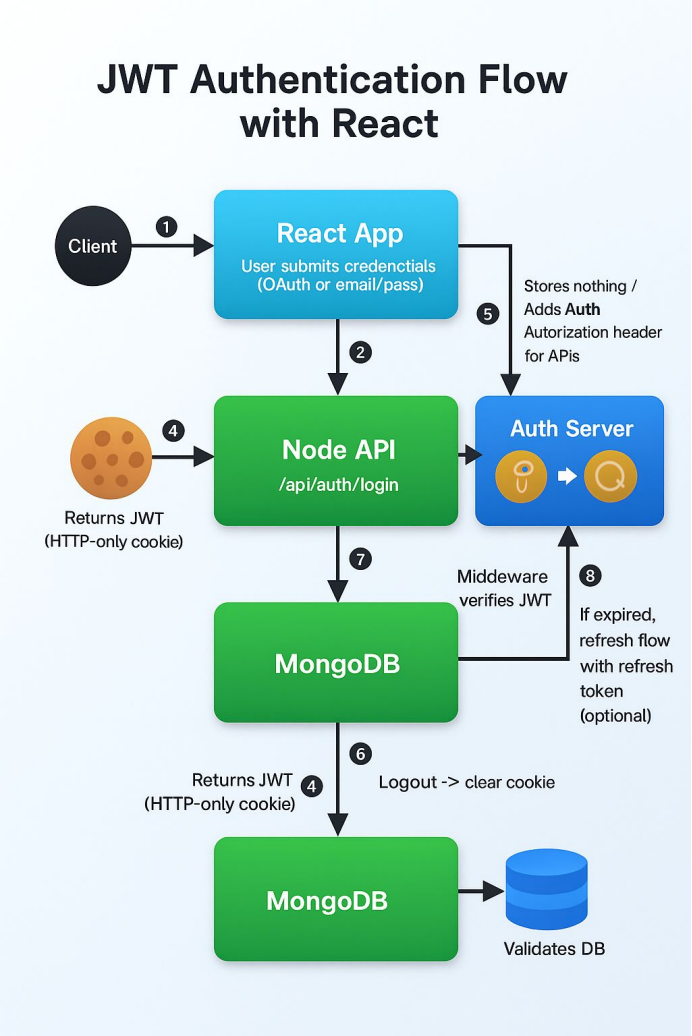
BOOMERANG CONSULTING

JWT



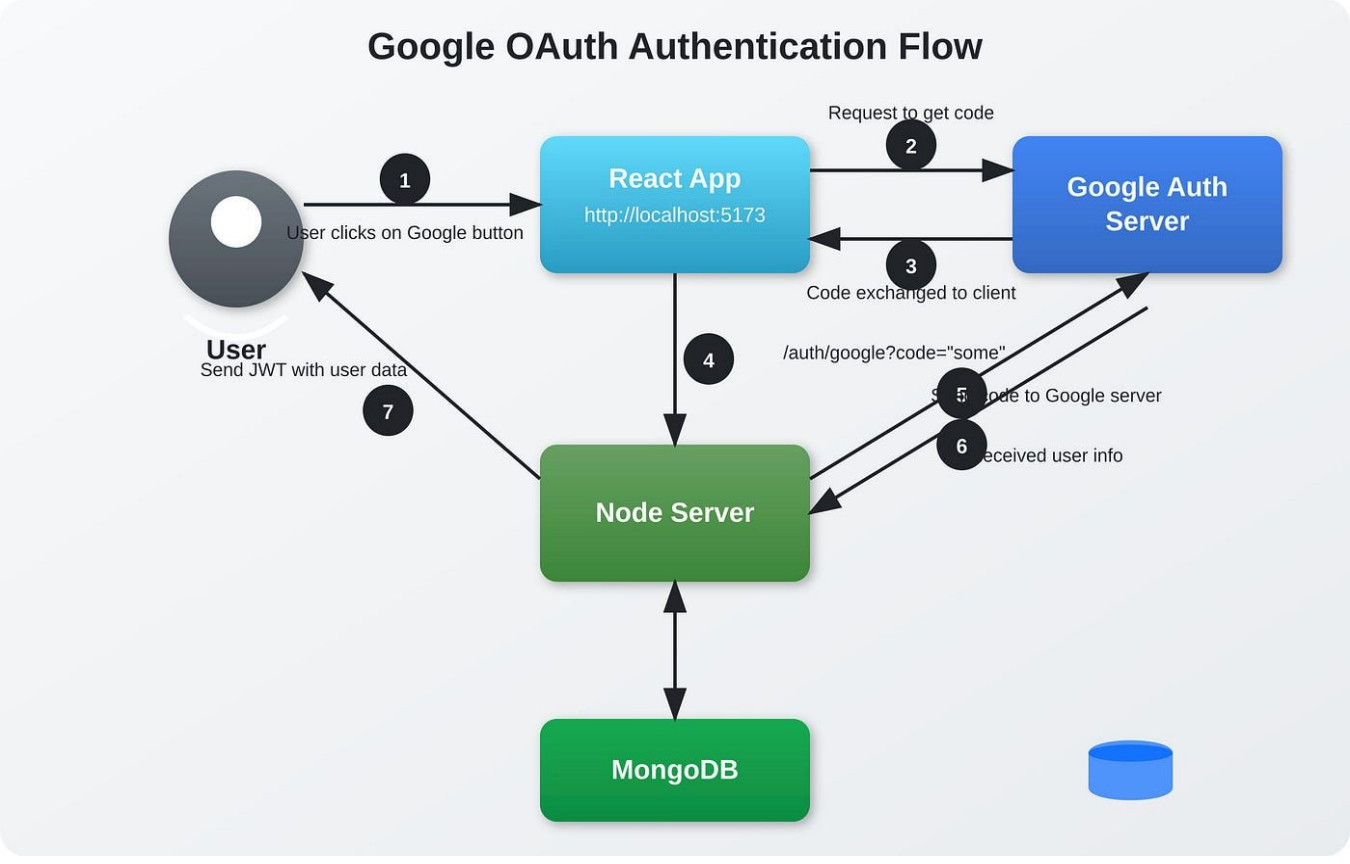
BOOMERANG CONSULTING

JWT



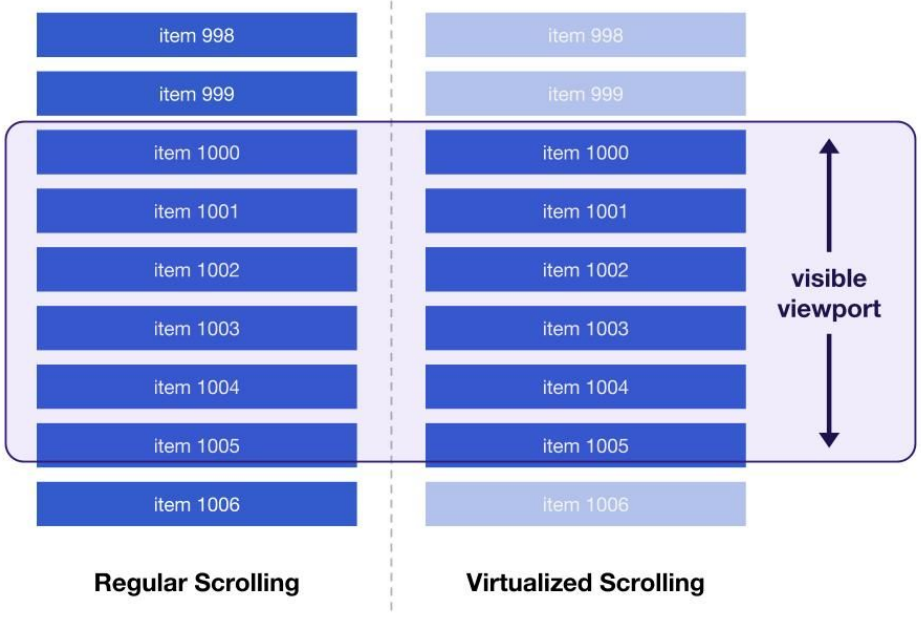
BOOMERANG CONSULTING

OAUTH

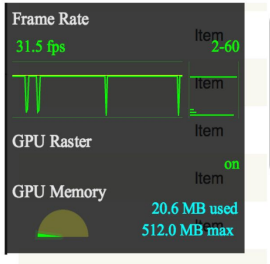


BOOMERANG CONSULTING

liste virtuelle (tanstack/virtual)



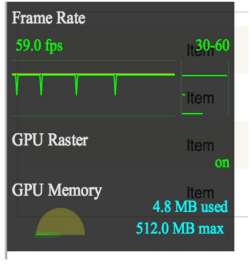
Rendering a list of 10K items



10K rows in the DOM
Slower initial render
Sluggish scrolling

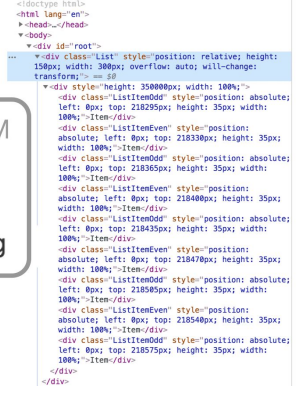
242.7 ms Rendering

Rendering 10K items w/virtualization



Minimal rows in DOM
Fast initial render
~60fps

2.4 ms Rendering



BOOMERANG CONSULTING

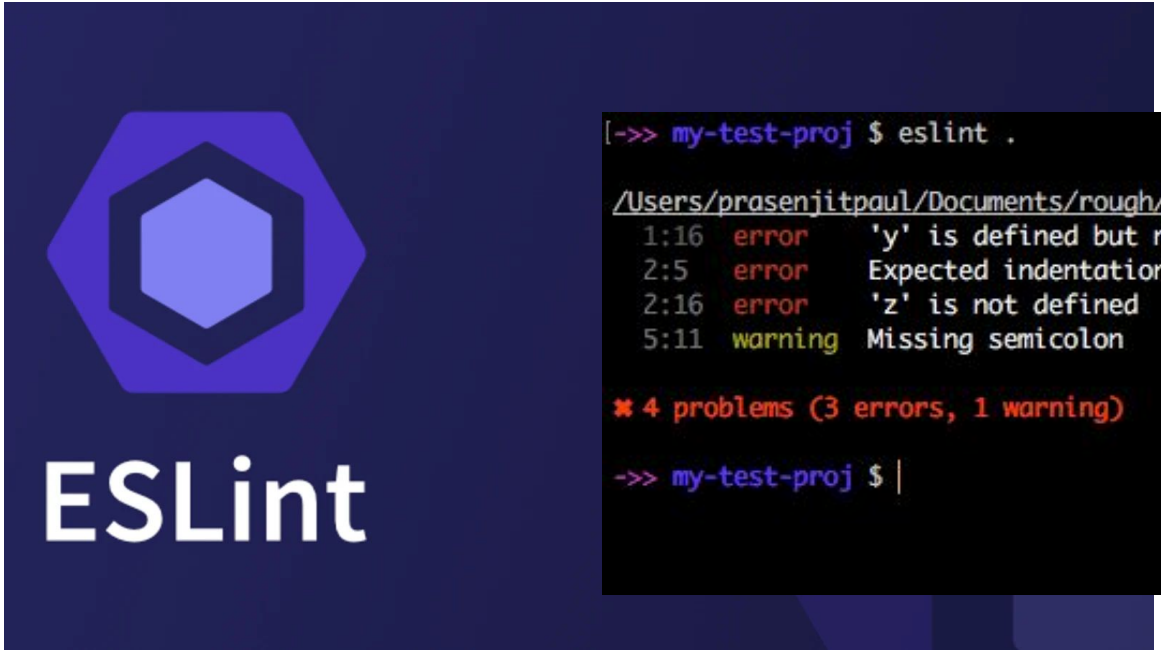
liste virtuelle (tanstack/virtual)

BOOMERANG CONSULTING

tests

BOOMERANG CONSULTING

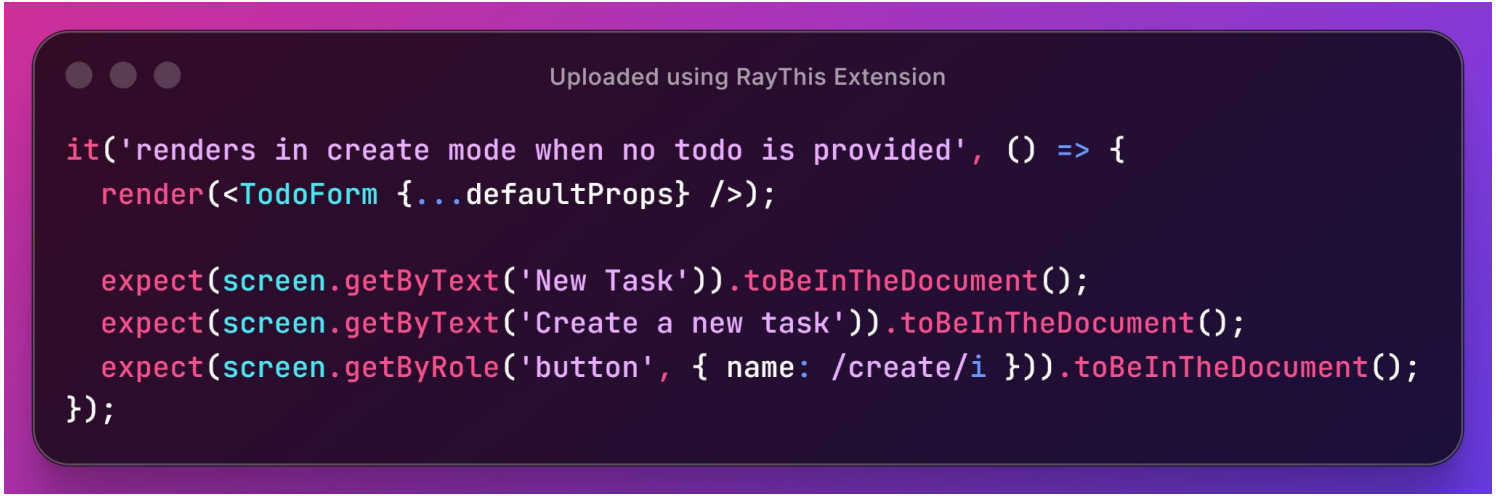
eslint




```
[->> my-test-proj $ eslint .

/Users/prasenjitpaul/Documents/rough/my-test-proj/index.js
  1:16  error    'y' is defined but never used                no-unused-vars
  2:5   error    Expected indentation of 2 spaces but found 4 indent
  2:16  error    'z' is not defined                           no-undef
  5:11  warning  Missing semicolon                            semi

✖ 4 problems (3 errors, 1 warning)

->> my-test-proj $
```

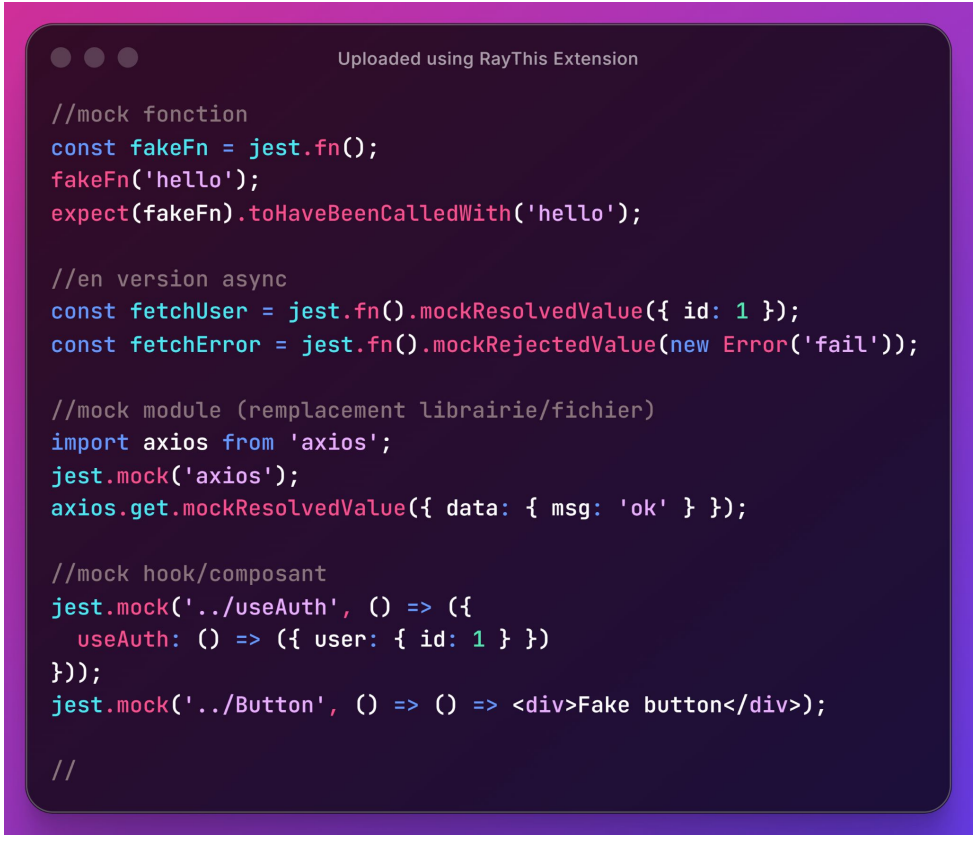
BOOMERANG CONSULTING

Tests présence



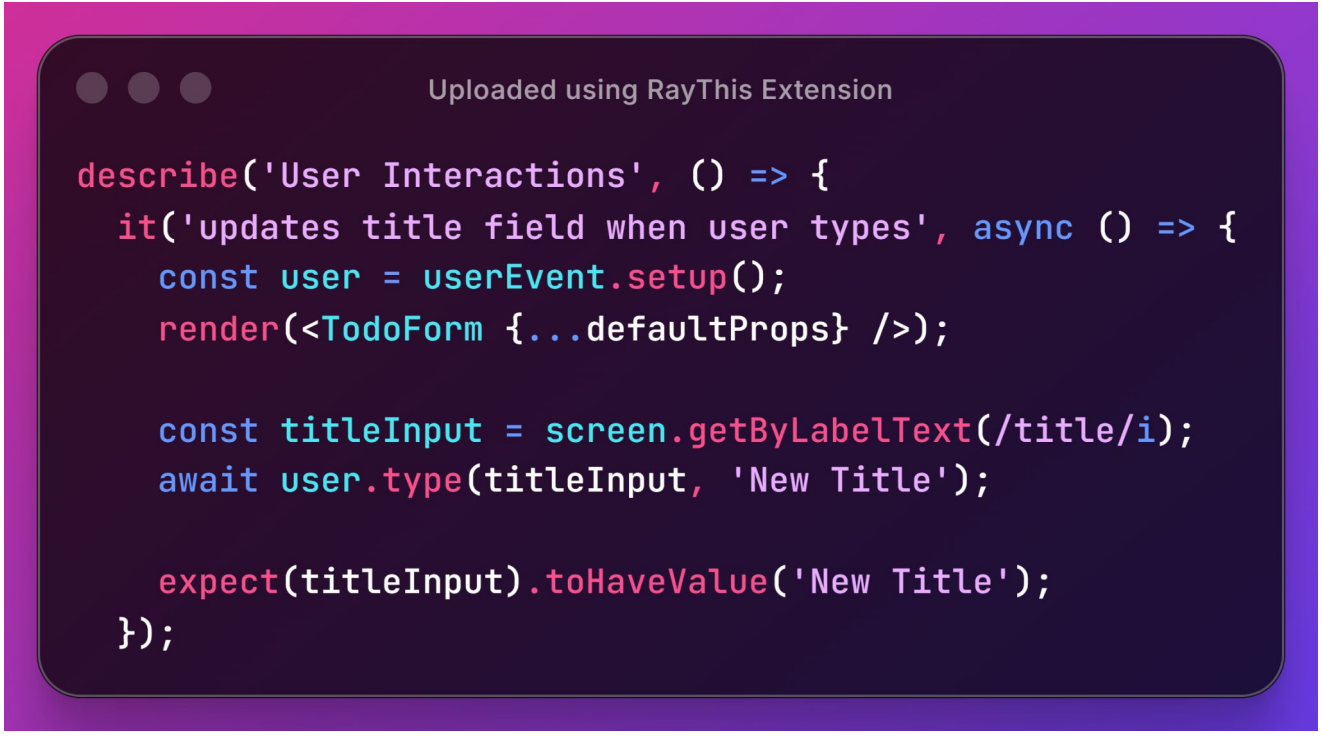
BOOMERANG CONSULTING

mocks



BOOMERANG CONSULTING

Tests évènements



BOOMERANG CONSULTING

Tests asynchrone

BOOMERANG CONSULTING

Tests hooks