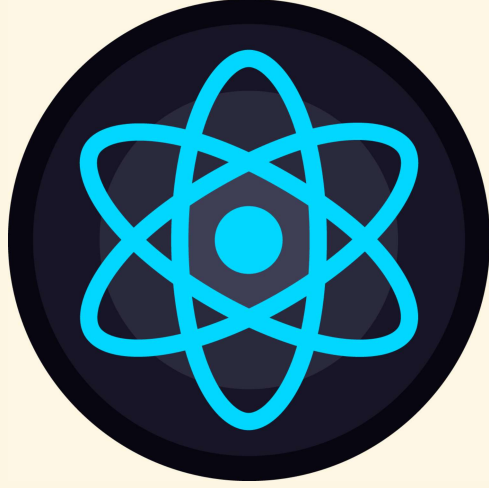


CONTEXTE



PROPS-DRILLING



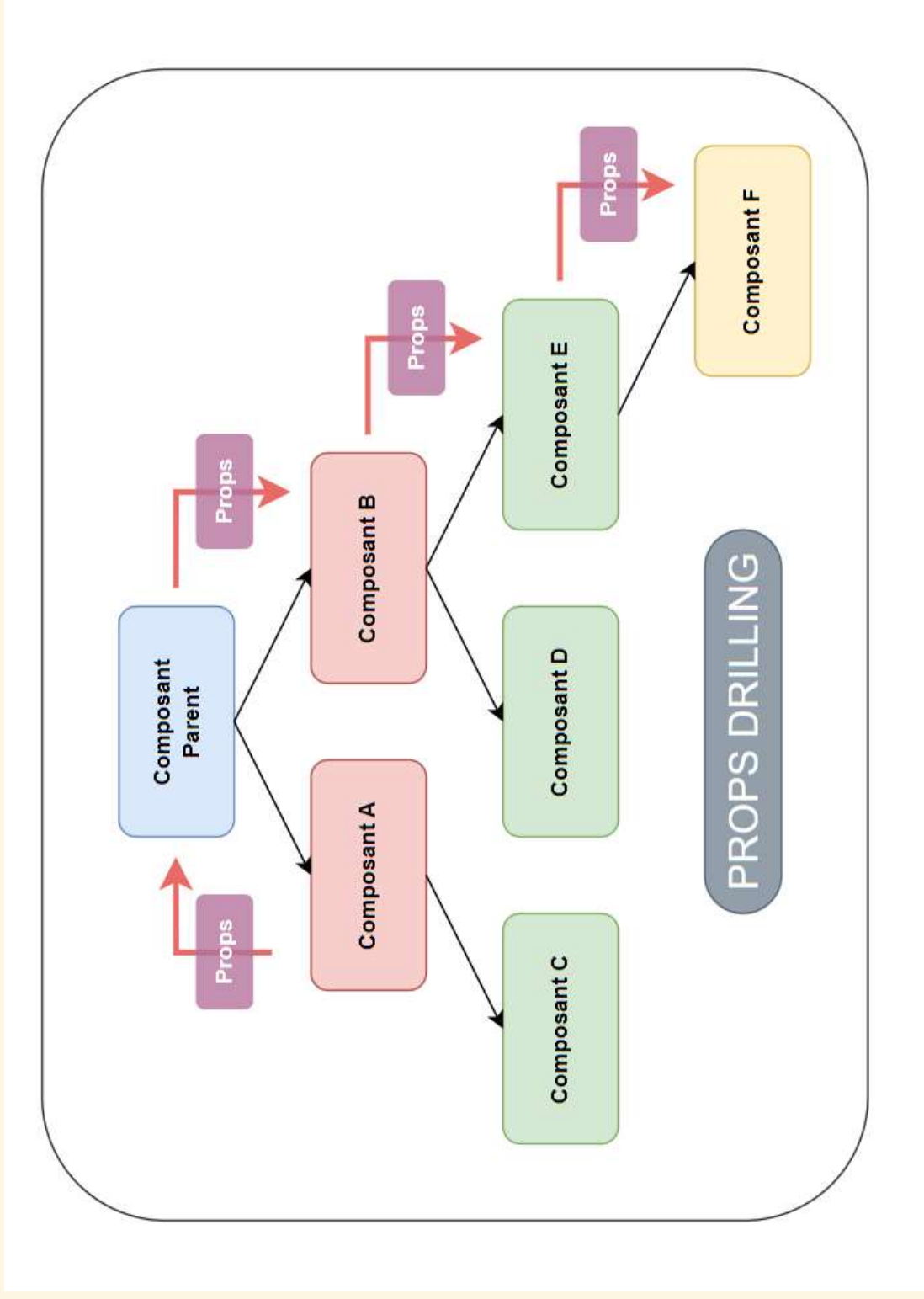
DÉFINITION

- Consiste à passer les données d'un composant parent à un composant enfant en utilisant les props.
- Deviens fastidieux plus votre arborescence de composants est profonde.
- Peut rendre le code difficile à lire et à maintenir





EN IMAGE





STATE MANAGEMENT

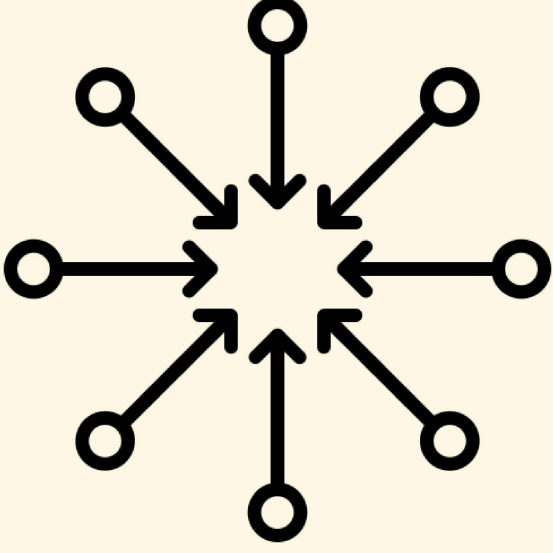


ETAT D'UNE APPLICATION

C'est une collection de données qui peuvent être modifiées au cours de l'exécution de l'application.



DÉFINITION

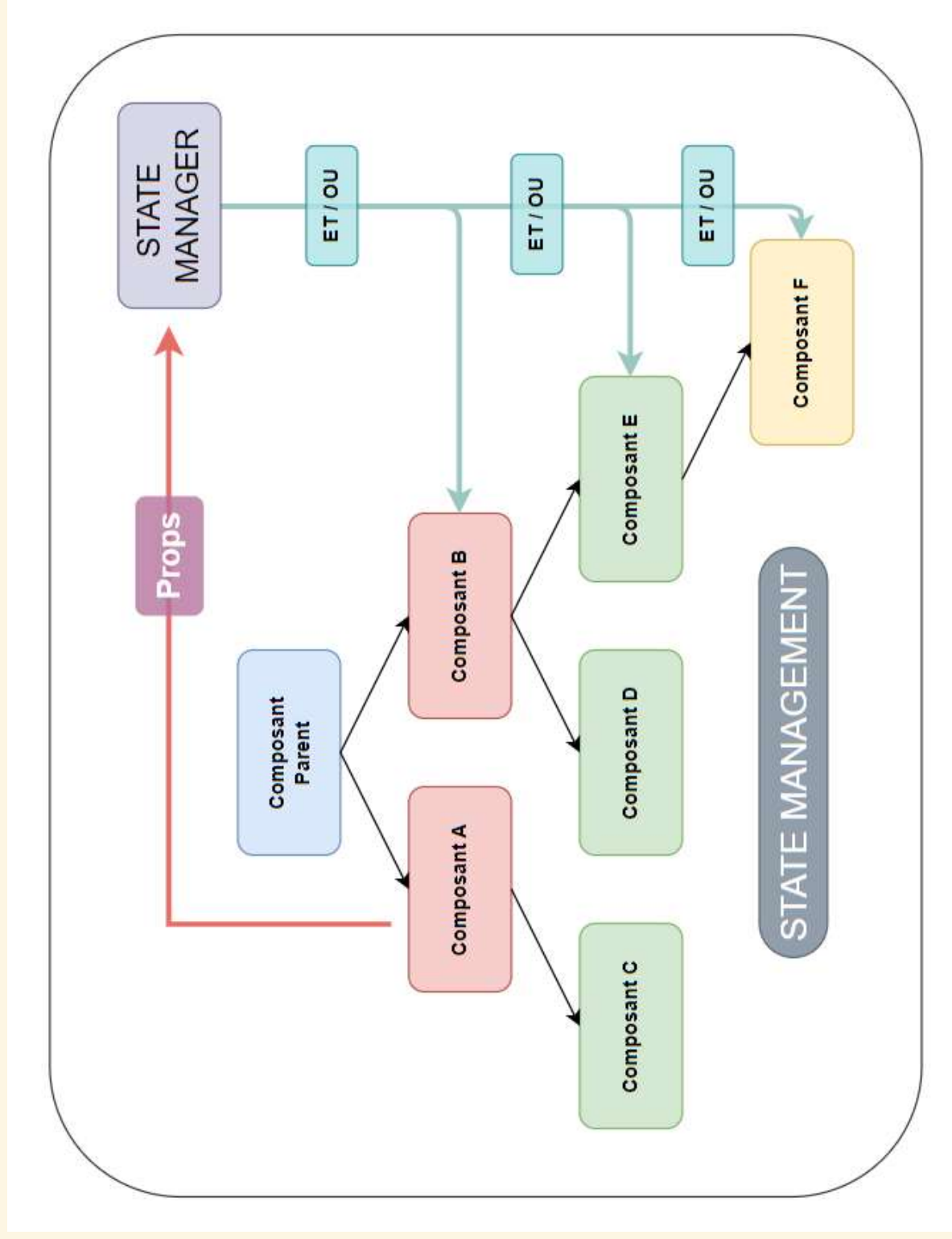


Centralise la gestion de l'état dans un endroit unique pour rendre le code plus facile à maintenir et à comprendre

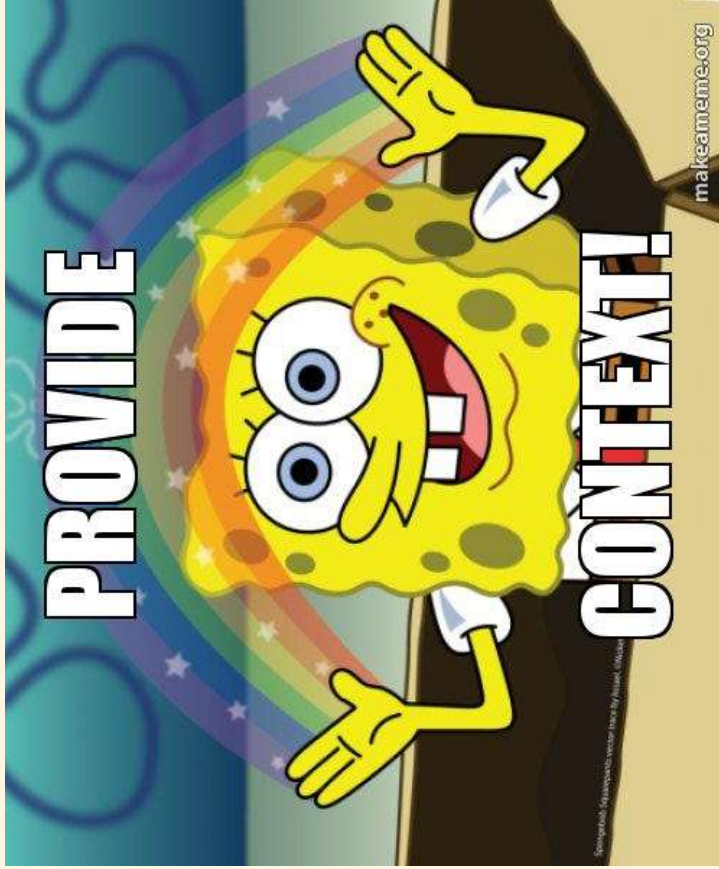


EN IMAGE





LE CONTEXTE



DÉFINITION

- Mécanisme de partage de données entre plusieurs composants.
- Défini par un composant parent et accessible par tous les composants enfants de ce composant
- Il est utilisé pour éviter de passer des props à travers l'arborescence.



FONCTIONNEMENT - ÉTAPE 1

- Créez un répertoire "context" dans votre projet
- Définissez le contexte en créant un objet "context" avec une valeur par défaut :

```
import React, { createContext } from "react";

const MyContext = React.createContext({
  theme: "light",
});

export default MyContext;
```



FONCTIONNEMENT - ÉTAPE 2

- Enveloppez l'élément qui sera le parent de tous les composants qui auront accès au contexte dans un élément "Provider" :

```
import { MyContext } from "./context";

function App() {
  return (
    <MyContext.Provider value={{ theme: "dark" }}>
      <div>
        {/* Les composants enfants qui ont accès au contexte vont ici */}
      </div>
    </MyContext.Provider>
  );
}
```



FONCTIONNEMENT - ÉTAPE 3

- Accédez au contexte dans les composants enfants en utilisant **useContext** :

```
import { useContext } from "react";
import MyContext from "../context";

// Composant enfant qui a accès au contexte
function ChildComponent() {
  // Accédez au contexte en utilisant le hook "useContext"
  const context = useContext(MyContext);
  return <div>Le thème actuel est : {context.theme}</div>;
}
```



DEMO



LA SUITE

Par ici !

