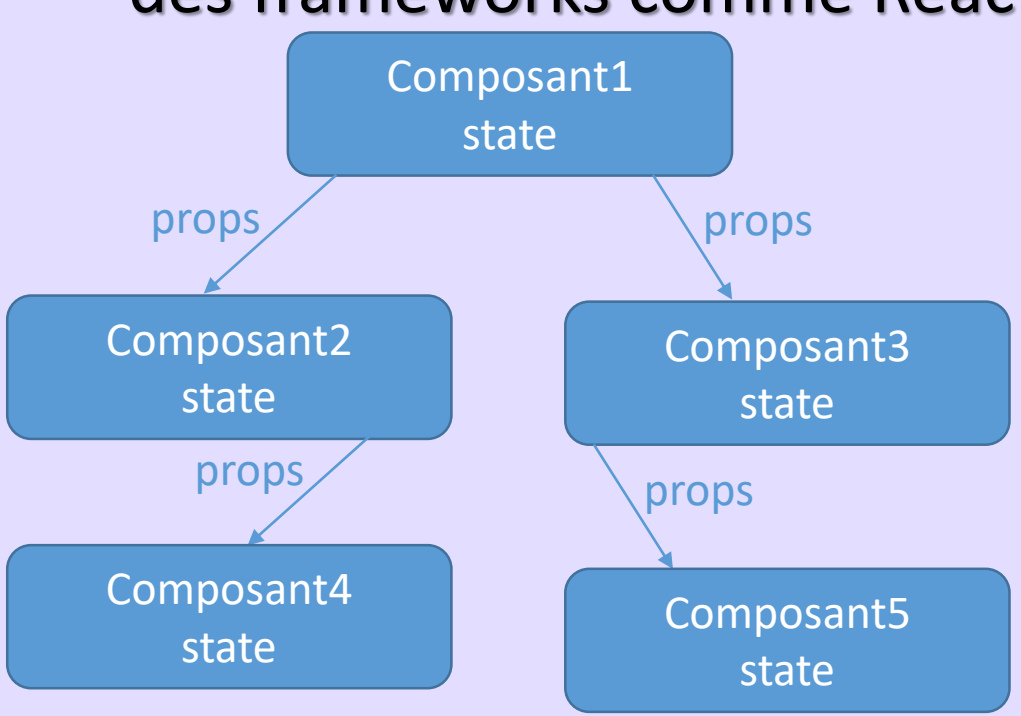
CH11-Introcution à Redux

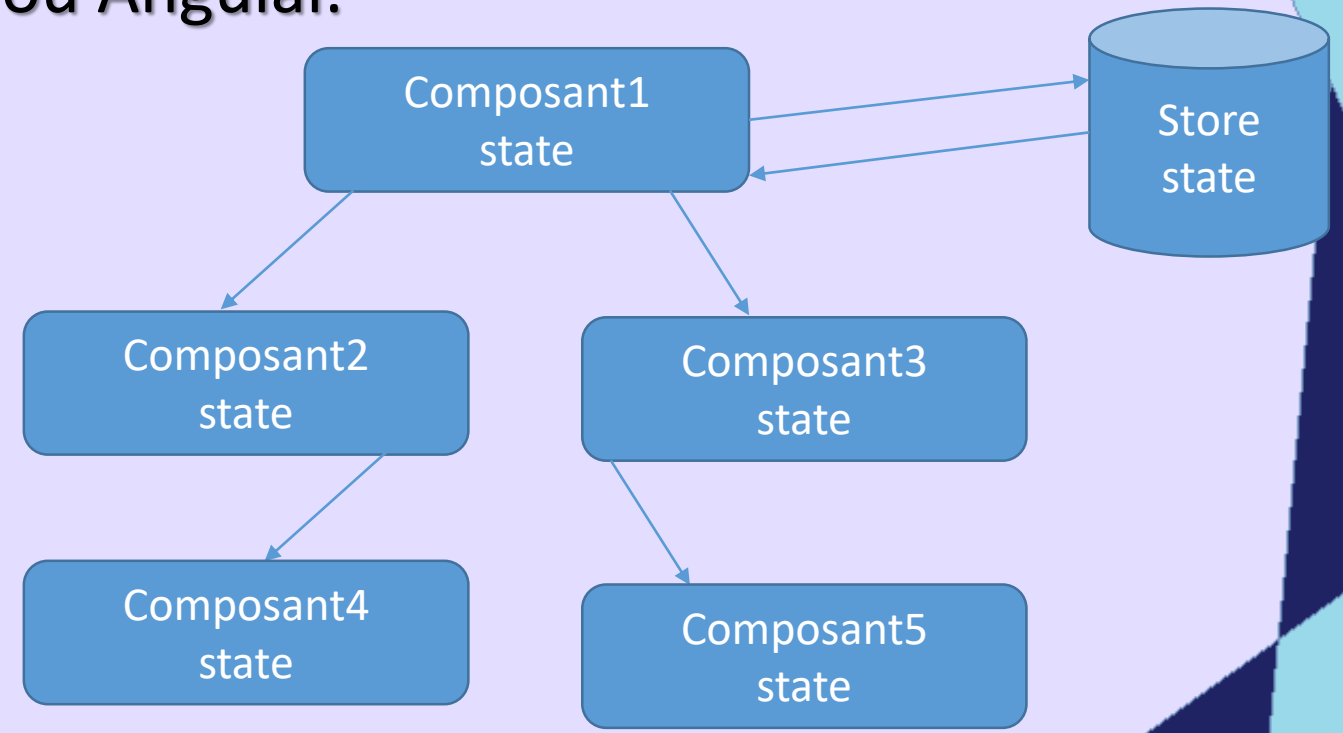
Développement front-end

I- Introduction

- Redux est une bibliothèque open-source Javascript de Gestion d'état pour les applications Web. Elle est plus communément utilisées avec des frameworks comme React ou Angular.



Sans Redux



Avec Redux

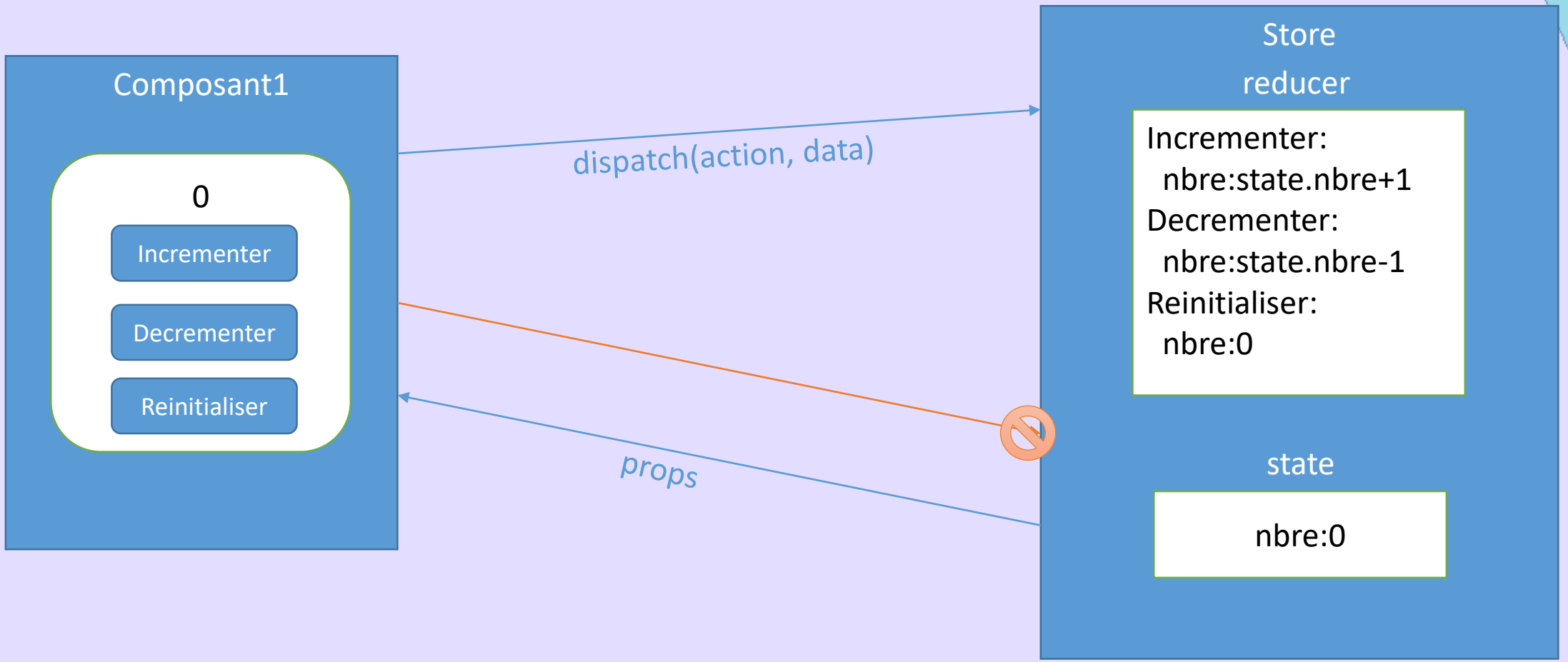
I-Introduction

- Sans Redux:
  - Les données entre les composants sont échangées à travers les props et stockées dans les états state.
  - Chaque donnée transmise entre un composant et un autre doit être stockée dans l'état des deux composants (redondance).
  - Le circuit entre le Composant1 et Composant5 passe obligatoirement par Composant3.

I-Introduction

- Avec Redux:
  - Un magazin des états (store) permet de stocker les états qui peuvent être modifiées et récupérées par tous les composants.
  - Le Composant1 peut stocker (ou modifier) l'état d'une variable directement dans le store, cet état peut être récupéré directement par le Composant5 sans passer par le circuit (Composant3).

II-Fonctionnement



III-Création d’une application avec Redux

- Par défaut, React n’intègre pas Redux. Pour l’activer, il faut ajouter les bibliothèques: redux et react-redux.
- La commande suivante permet de les installer :

```
npm install redux react-redux
```

III-Création d’une application avec Redux

- Créer une nouvelle application React:

```
npx create-react-app chapitre11
cd chapitre11
```

- Installer ensuite les bibliothèques redux et react-redux:

```
npm install redux react-redux
```

III-Création d'une application avec Redux

- Créer ensuite le reducer (reducer.js) :

```
const initialState={nbre:0}
export default function reducer(state=initialState,action){
    switch(action.type){
        case "Incrementer":
            return {...state,nbre:state.nbre+1}
        case "Decrementer":
            return {...state,nbre:state.nbre-1}
        case "Reinitialiser":
            return {...state,nbre:0}
    }
    return state;
}
```

III-Création d'une application avec Redux

- Créer le store et le rendre accessible par React sur index.js:

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import { Provider } from 'react-redux';
import { legacy_createStore } from 'redux';
import reducer from './reducer';
import App from './App';
const store=legacy_createStore(reducer);
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <Provider store={store}>
    <App/>
  </Provider>
);
```

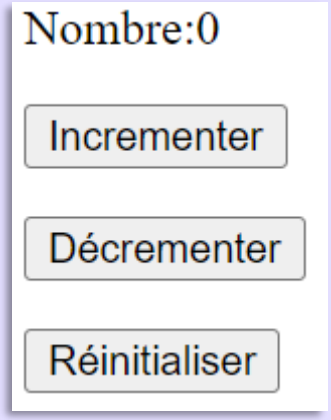
III-Création d'une application avec Redux

- Traiter les états dans le composant App.js:

```
import { useDispatch,useSelector } from "react-redux";
function App() {
  const nbre=useSelector(data=>data.nbre)
  const dispatch=useDispatch();
  return (
    <div className="App">
      <p>Nombre:{nbre}</p>
      <p><button onClick={()=>dispatch({type:"Incrementer"})}
      >Incrementer</button></p>
      <p><button onClick={()=>dispatch({type:"Decrementer"})}
      >Décrementer</button></p>
      <p><button onClick={()=>dispatch({type:"Reinitialiser"})}
      >Réinitialiser</button></p>
    </div>
  );
}
export default App;
```

III-Création d’une application avec Redux

- **Rendu:**



III-Création d'une application avec Redux

- Passer des données dans l'action: (App.js)

```
function App() {
  const nbre=useSelector(data=>data.nbre)
  const [inc,setInc]=useState('');
  const dispatch=useDispatch();
  return (
    <div className="App">
      <p>Incrément:<input type="text" onChange={(e)=>setInc(e.
      target.value)}/></p>
      <p>Nombre:{nbre}</p>
      <p><button onClick={()=>dispatch({type:"Incrementer",
      payload:inc})}>Incrementer</button></p>
      <p><button onClick={()=>dispatch({type:"Decrementer",
      payload:inc})}>Décrementer</button></p>
      <p><button onClick={()=>dispatch({type:"Reinitialiser"})}
      >Réinitialiser</button></p>
    </div>
  );
}
```

III-Création d'une application avec Redux

- Passer des données dans l'action: (reducer.js)

```
const initialState={nbre:0}
export default function reducer(state=initialState,action){
    switch(action.type){
        case "Incrementer":
            return {...state,nbre:state.nbre+parseInt(action.payload)}

        case "Decrementer":
            return {...state,nbre:state.nbre-parseInt(action.payload)}

        case "Reinitialiser":
            return {...state,nbre:0}
    }
    return state;
}
```

III-Création d'une application avec Redux

- Passer des données dans l'action: (rendu)