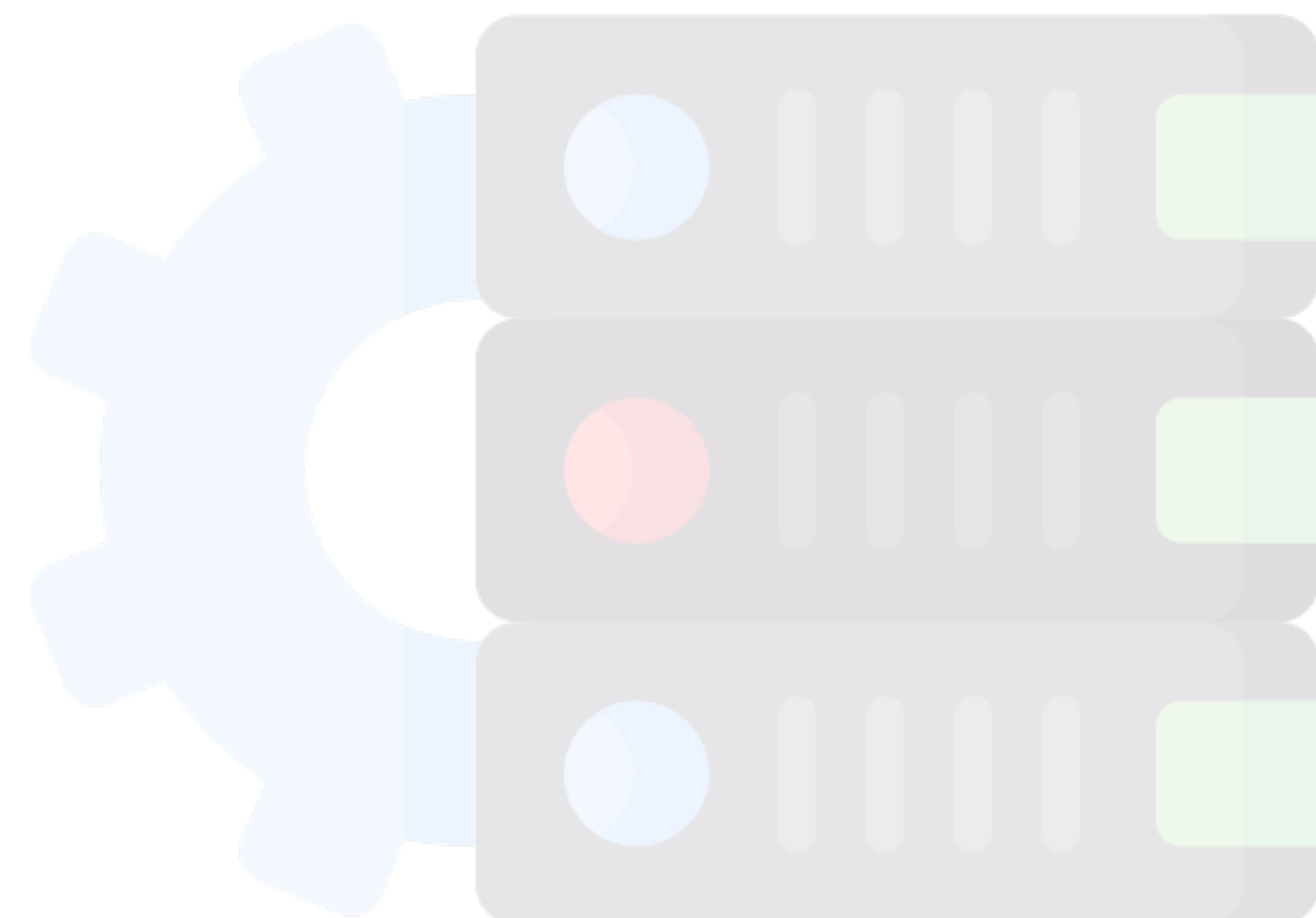




REDUX

Better safe than sorry.

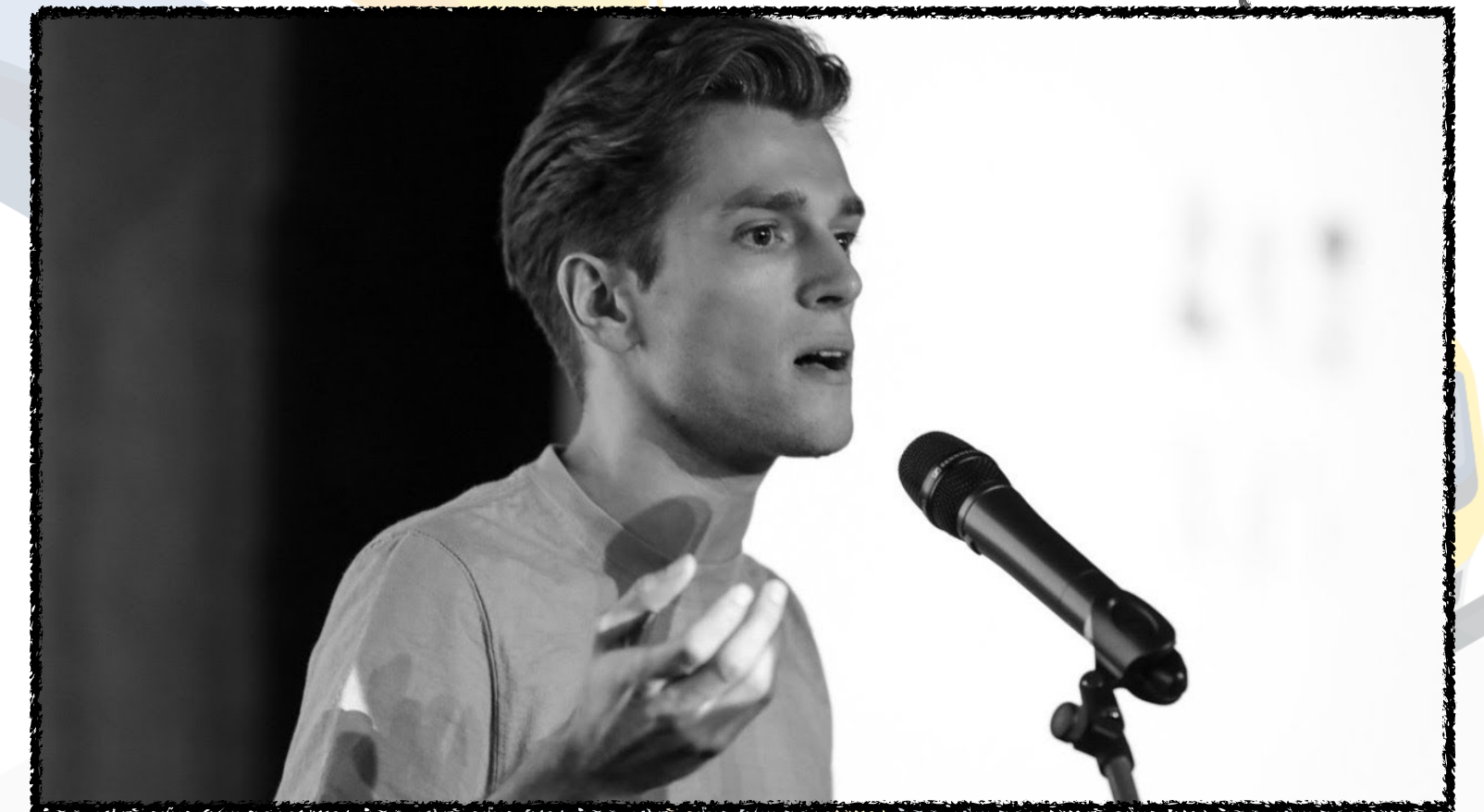


Redux c'est quoi ?

Développé par Dan Abramov depuis 2015.

Redux est une bibliothèque compacte qui fournit un **conteneur d'état** pour les applications **JavaScript**.

Redux met à disposition un **store**, un objet JavaScript qui contient les différentes informations **stockées globalement**.



Redux c'est quoi ?

La **librairie** met à disposition deux **hooks** très pratique, *useSelector* et *useDispatch*.

Ces **hooks** nous permettent de **récupérer** les données *stockées globalement*.

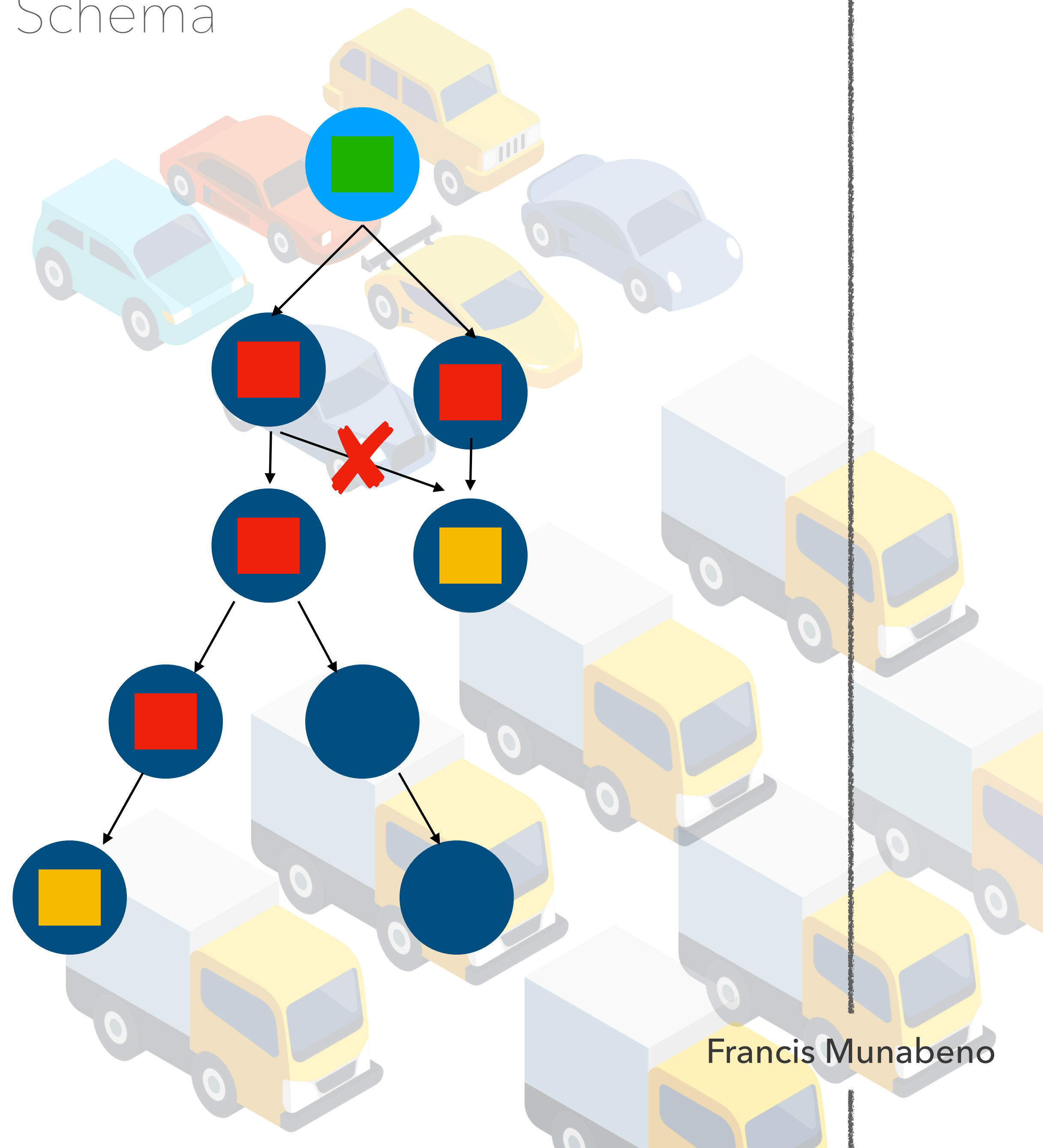
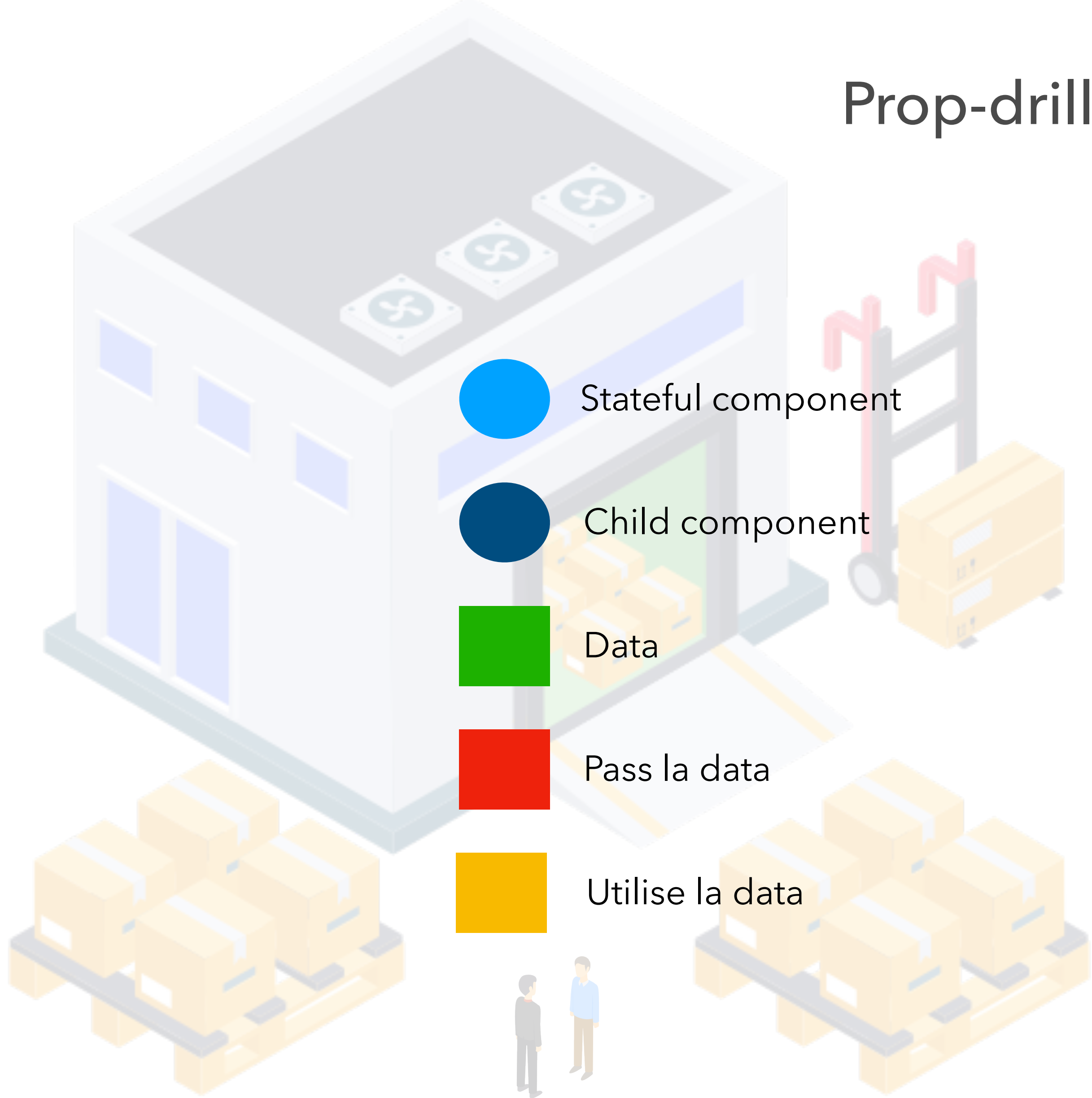
Mais aussi à **déclencher** des **actions** qui permettent de *modifier un state global*.



Intro Redux c'est quoi ?

Francis Munabeno

Prop-drilling Schéma

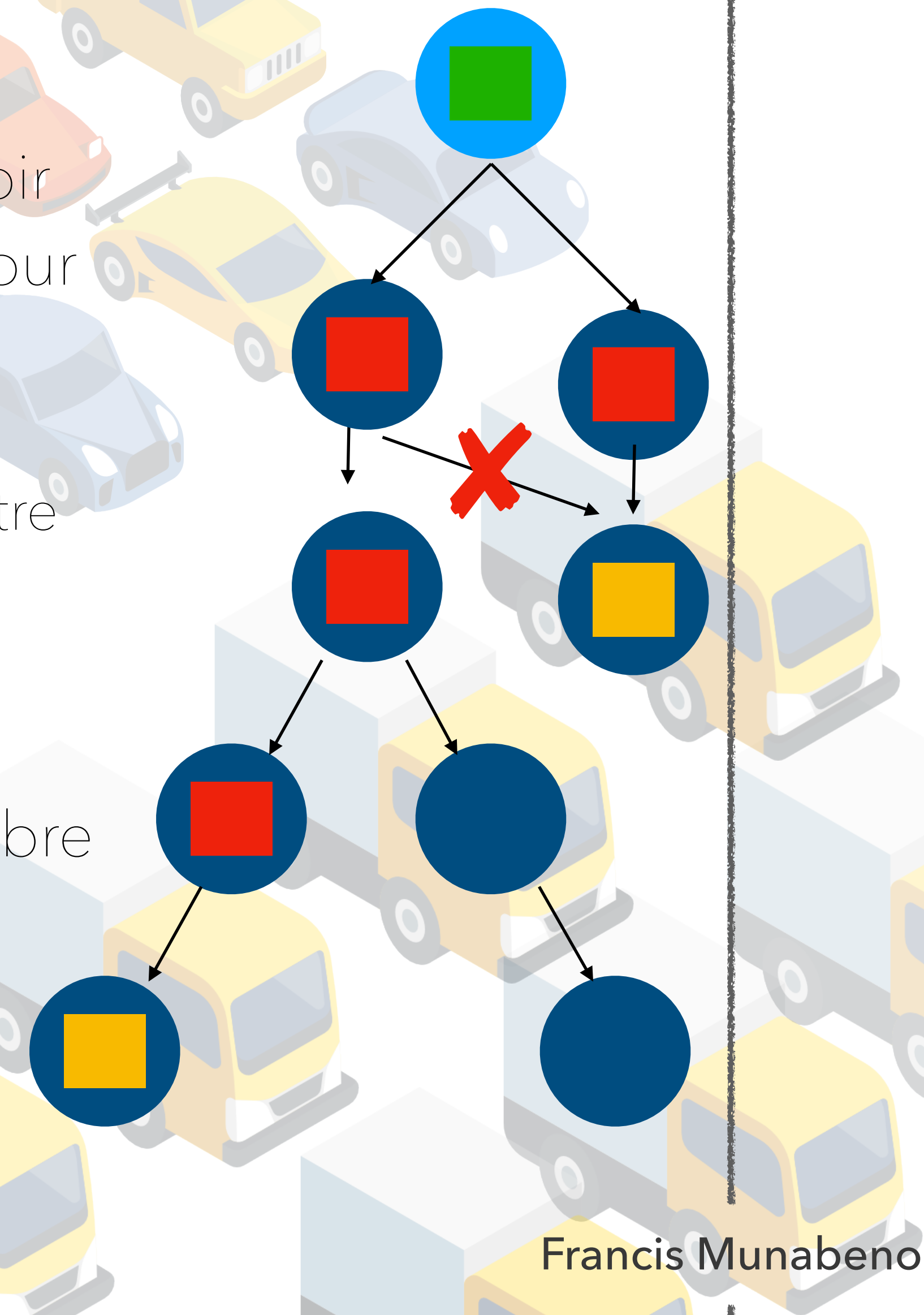


Prop-drilling Explication

Quand **React** commence à se **complexifier**, il est récurrent de voir des cas où un passe des **props** à un composant uniquement pour que celui-ci le transmette à ses **composants enfants**.

Cela ne pose aucun **problème** quand il s'agit de passer une **props** entre 2 ou 3 niveaux de **composant**. Mais quand **l'information** doit passer à plus de **5 niveaux** différents dans l'arborescence des composants.

Cela est **problématique**, car les données **travers** un grand nombre de **composants** alors qu'une **minorité** utilise **l'information** transmise.



Prop-drilling Solution

L'une des **solutions** les plus utilisées est de mettre en place un **state global**.

Informations **accessibles** à tous les niveaux de l'application.

Cela peut être fait avec des **solutions** comme **Redux** ou la **Context API** de React.

Global state

Global state



Prop-drilling pourquoi utiliser Redux

Francis Munabeno

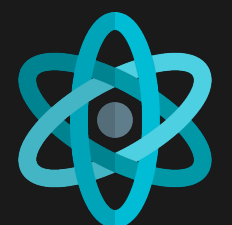
Context Intro

Apparue officiellement en mars 2018, Context permet de stocker des informations dans un **context**.

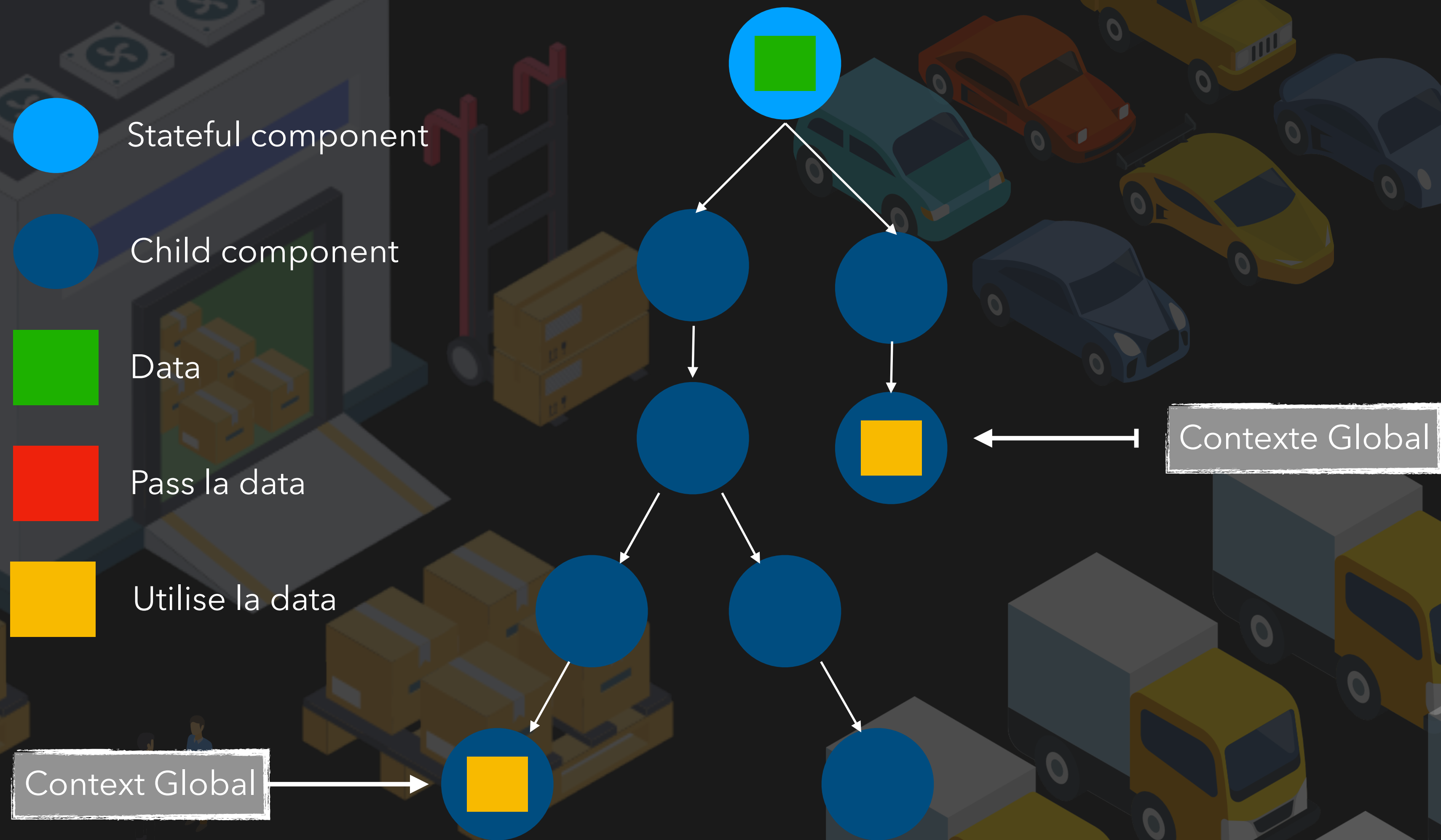
Il met à disposition un **provider** ((composant React) => fournisseur).

Qui permet à tous les **composants** enfants d'accéder à l'information **stockée** dans le **contexte**.

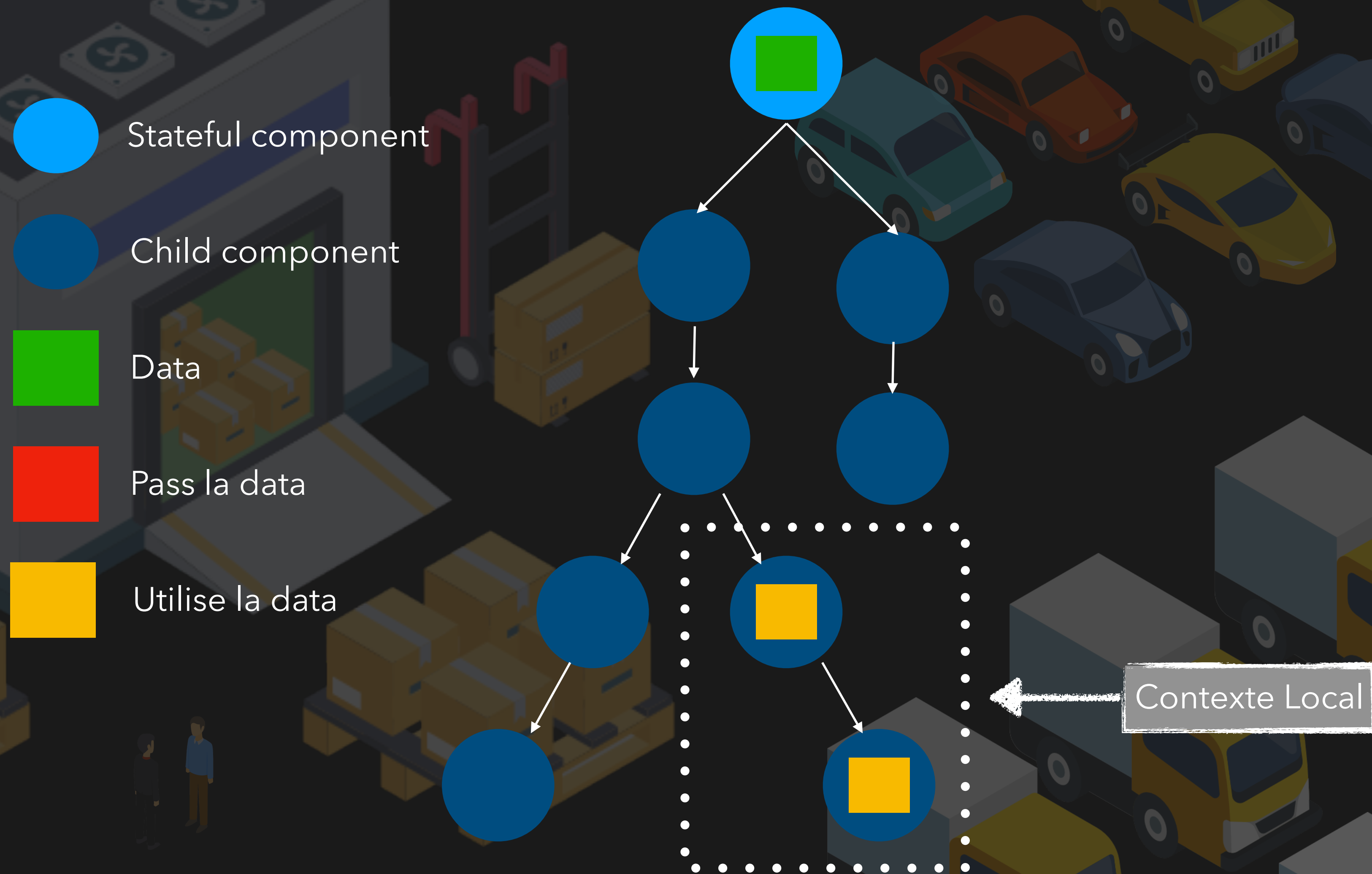
Sans avoir **besoin** de passer par tous les **composants Intermédiaires**.



Context Schema Global



Context Schema Local

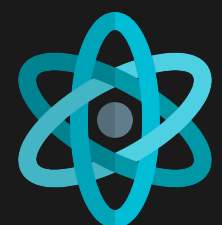


Context mais dans quel context ?

Context est conçu pour **partager** des données qui peuvent être considérées comme globales, **Authentifié, Thème, Langue**.

React met en garde sur **l'utilisation** de **contexte**, car il rend les composants, plus **complexe**.

Mais, pas de panique son utilisation s'est **simplifiée** avec l'arrivée du **hook useContext** mis à disposition **nativement** dans React.

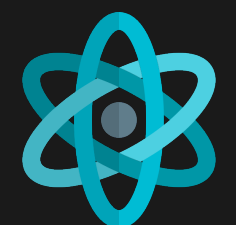


Contexte mais dans quel contexte ?

Francis Munabeno

Context remplace Redux ?

Avons nous encore besoin de Redux ?



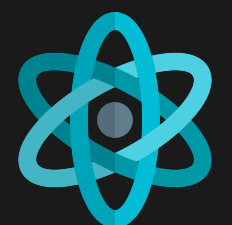
Contexte Contexte Remplace Redux ?

Francis Munabeno

Redux Vs Contexte

Redux est une bonne option lorsque votre application React commence à prendre de l'ampleur et à avoir une base de code conséquente.

Cependant, si vous souhaitez intégrer quelques données globales rapidement et d'une manière plus simplifiée, la Context API semble pouvoir répondre à vos attentes. Elle est intégrée directement dans React.



Redux Avantages ?

Avantages Redux:

- Etat centralisé (Store).
- Optimisations des performances (Redux ne fait pas de rendus inutiles)
- Fonctions de réduction pures.



Redux Avantages ?

Francis Munabeno

Redux Inconvénients ?

Inconvénients Redux:

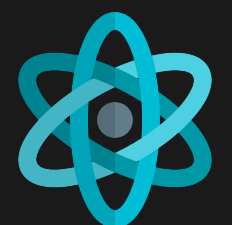
- Difficile à adopter lorsqu'on débute.
- Librairie externe à React.



Context Avantages ?

Avantages Context:

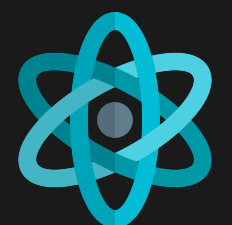
- Facile à prendre en main.
- Solution native de React.
- Contexte locaux (n'englobent pas l'intégralité de l'application mais plutôt une petite quantité de composants).



Context Inconvénients ?

Inconvénients Context:

- L'API Context n'est pas adaptée pour des changement fréquents de state.
- Tous les composants consommant le contexte vont être re-render à chaque changement du state global proposé par le contexte, même si l'information qu'ils utilisent n'est pas changée.



Redux Vs Contexte Conclusion

Aucune de ces deux options n'est prédominante. En fonction de la portée de votre projet et de l'implémentation que vous souhaitez réaliser, une de ces 2 solutions vous permettra de gérer les données globales de votre application.

