



REDUX & NGRX

Redux pour Angular



PRESENTATION DE REDUX

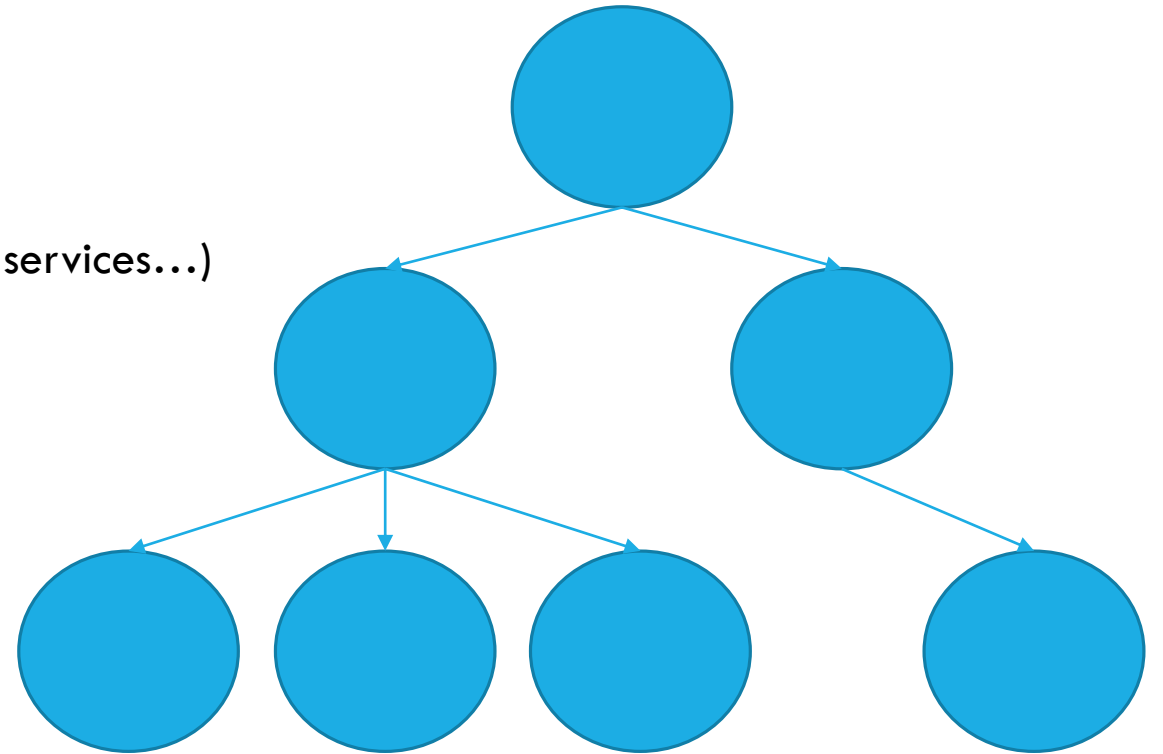
REDUX EN BREF

- Design pattern apparu en 2013
- Inspiré de FLUX (architecture pensée pour React)
- Propose une nouvelle façon pour gérer et organiser « **l'état** » d'une application
- Indépendant du framework de rendu (Angular, React, Vue JS...)

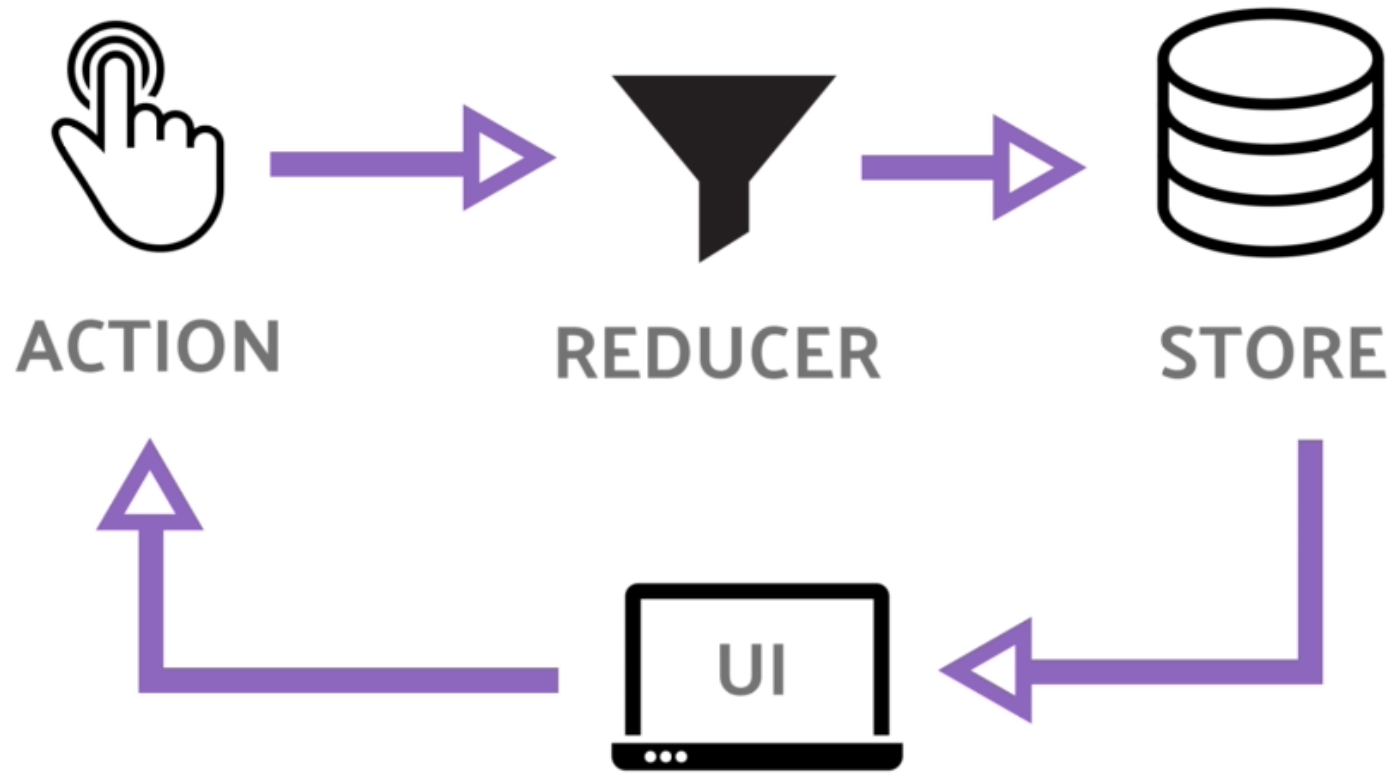


LE PROBLEME

- Applications de plus en plus complexes
- Etat difficile à gérer dans les SPA
 - Données (issues d'API, en session, en local, dans des services...)
 - Etat de l'UI (spinner, pop-up, pagination ...)
 - Etat des routes (route courante...)
- Etat dispersé entre les composants



REDUX UNIDIRECTIONAL DATA-FLOW



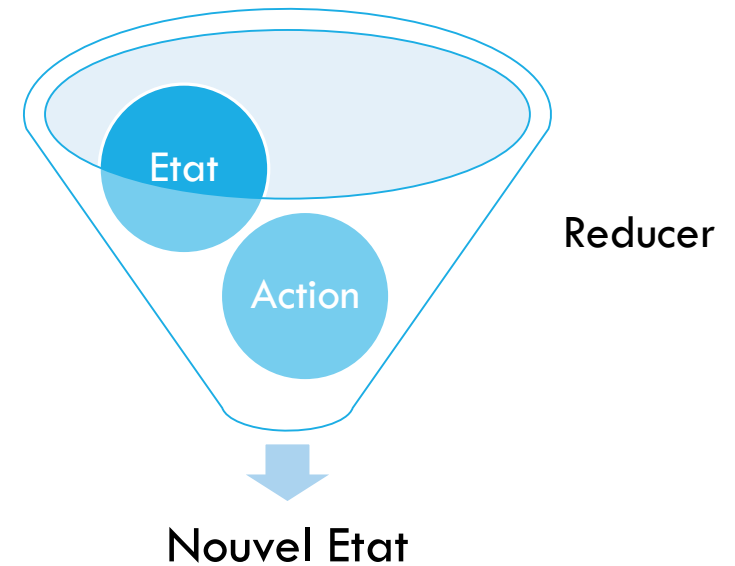
STORE ET REDUCERS

Store

- Etat de l'application à un moment donné

Reducer

- Fonction pure renvoyant un nouvel état en fonction d'une action et de l'état en cours



3 PRINCIPES

- « Single source of truth »
- L'état est en LECTURE seule
- L'état ne peut être modifié que par des fonctions pures (les reducers)

AVANTAGES

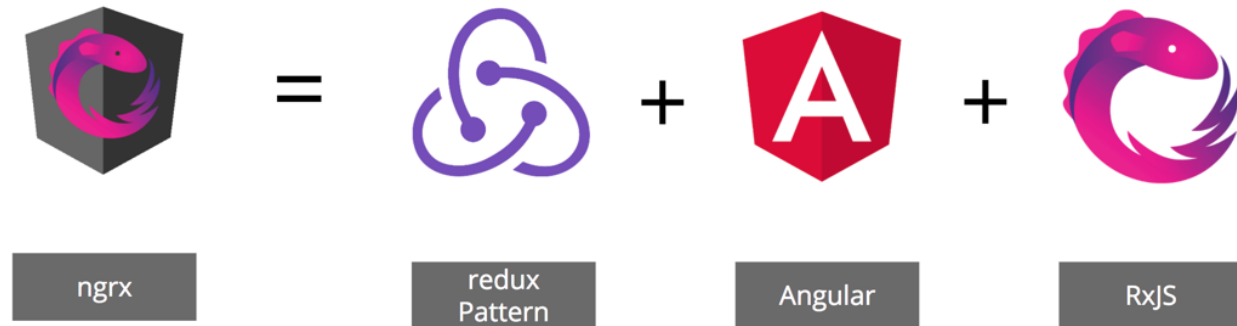
- Application prédictive
- Etat centralisé
- « Time travel » debugging
- Indépendant du framework front-end



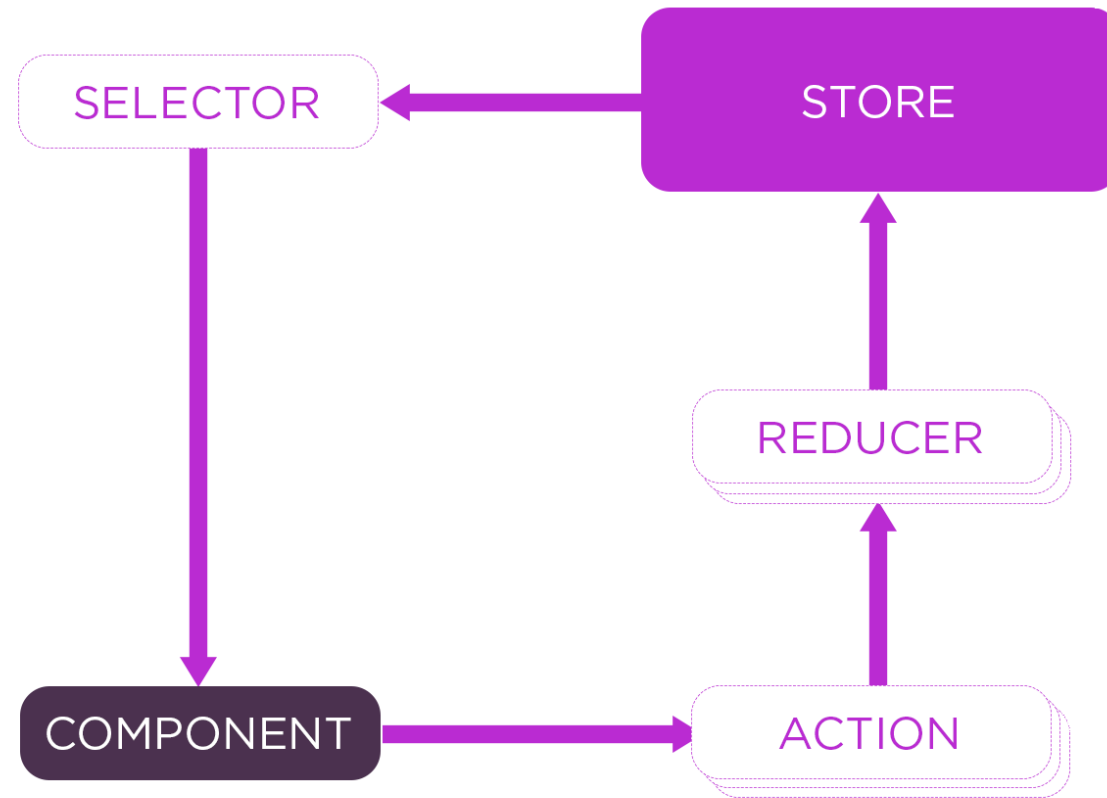
PRESENTATION DE NGRX

NGRX EN BREF

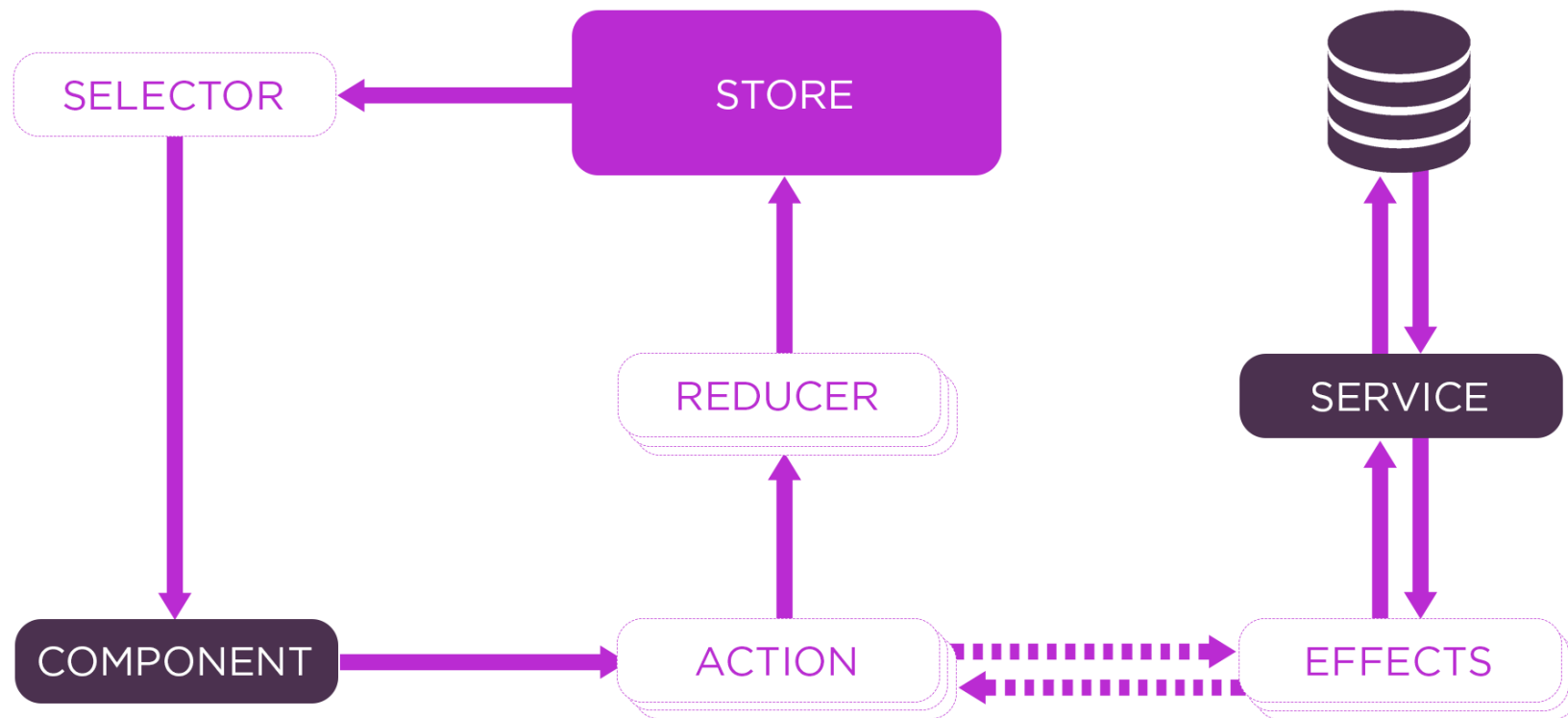
- Librairie pour Angular
- Implémente Redux
- Plusieurs « sous-module »
 - ngrx/store
 - ngrx/store-devtools
 - ngrx/effects
 - Etc...



NGRX/STORE



NGRX/EFFECTS



CREDITS IMAGES

- Images sur les pages « Unidirection data-flow » et « NgRx en bref »
 - Daouda Diallo - <https://medium.com/code-divoire/cr%C3%A9ation-dune-application-angular-avec-ngrx-pattern-redux-845f9ac7606e>
- Images sur les pages « NgRx/store » et « NgRx/effects »
 - Site officiel de NGRX - <https://ngrx.io/guide/store>