Devenir **Natif** avec **React**

Du Web au Mobile

24 Novembre 2016

CODEURS EN SEINE


Proxiad
Axe Seine

La **mobilité**, pourquoi pas **React Native** ?

Aujourd'hui

- 3 types majeurs d'applications mobiles
  - Natives
    - Android (Java)
    - iOS (Objective-C, Swift)
    - Windows Phone (C#)
  - Hybrides (Cordova, Teleric, Ionic)
  - Cross-Platform (React Native, Xamarin, Native Script)

Comparatif

	Natives	Hybrides	Cross-Platform
Partage de code	★☆☆☆☆	★★★★☆	★★★★☆
Rapidité de développement	★★☆☆☆	★★★★☆	★★★★★
Performances	★★★★★	★★☆☆☆	★★★★☆
Communauté	★★★★☆	★★★☆☆	★★★★☆
Profil développeur	!= pour chaque plateforme	Commun	Commun

Comparatif

	Natives	Hybrides	Cross-Platform
Partage de code	★☆☆☆☆	★★★★☆	★★★★☆
Rapidité de développement	★★☆☆☆	★★★★☆	★★★★★
Performances	★★★★★	★★☆☆☆	★★★★☆
Communauté	★★★★☆	★★★☆☆	★★★★☆
Profil développeur	!= pour chaque plateforme	Commun	Commun

Comparatif

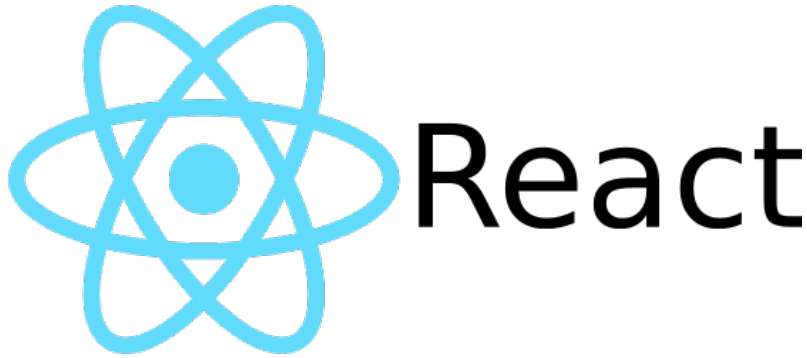
	Natives	Hybrides	Cross-Platform
Partage de code	★☆☆☆☆	★★★★☆	★★★★☆
Rapidité de développement	★★☆☆☆	★★★★☆	★★★★★
Performances	★★★★★	★★☆☆☆	★★★★☆
Communauté	★★★★☆	★★★☆☆	★★★★☆
Profil développeur	!= pour chaque plateforme	Commun	Commun

Comparatif

	Natives	Hybrides	Cross-Platform
Partage de code	★☆☆☆☆	★★★★☆	★★★★☆
Rapidité de développement	★★☆☆☆	★★★★☆	★★★★★
Performances	★★★★★	★★☆☆☆	★★★★☆
Communauté	★★★★☆	★★★☆☆	★★★★☆
Profil développeur	!= pour chaque plateforme	Commun	Commun

React

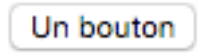
- Librairie **JavaScript**
- UI orientée **composants**
- Créée par **Facebook** (2013)
- **Open-source** (https://github.com/facebook/react)
- Simple & performante
- Très populaire ! (53k+ étoiles)



Un composant React

```
import React from 'react';
import { render } from 'react-dom';

class Button extends React.Component {
  componentWillMount() {
    console.info('Un bouton sauvage apparaît');
  }

  render() {
    return (
      <button onClick={() => alert("T'es tendue!")}>
        Un bouton
      </button>
    );
  }
}

render(<Button />, document.getElementById('root'));
```

- Classe qui étend **React.Component**
- Possède à minima une fonction **render**
- Et si besoin, des méthodes de **cycle de vie**



Le JSX


```jsx
<Button text="Un bouton" />
```

```jsx
React.createElement(Button, {
    text: "Un bouton"
});
```

Un bouton

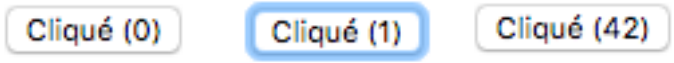
```
<button>Un bouton</button>
```

Les composants **stateful**

```
import React from 'react';
import { render } from 'react-dom';

class CounterButton extends React.Component {
    constructor() {
        super();
        this.state = { counter: 0 };
    }

    incrementCounter() {
        this.setState({ counter: this.state.counter + 1 });
    }

    render() {
        return (
            <button onClick={() => this.incrementCounter()}>
                {this.props.text} ({this.state.counter})
            </button>
        );
    }
}

render(<CounterButton text="Cliqué" />, document.getElementById('root'));
```

Un composant **React** possède :

- Un **état interne**
  - Sa modification déclenche le rendu du composant si nécessaire
- Des **propriétés**
  - De simples attributs passés par les composants parents



Les composants **stateless**

```
import React from 'react';
import { render } from 'react-dom';

function Button(props) {
    return (
        <button onClick={props.onClick}>
            {props.text}
        </button>
    );
}

render(
  <Button
    text="Bouton stateless"
    onClick={() => alert("Je n'ai pas d'état")} />,
  document.getElementById('root')
);
```

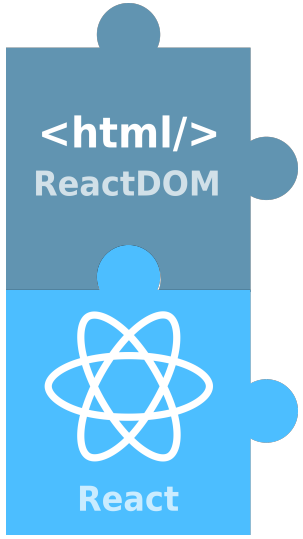
- Pas d'état interne
- Simple fonction
- A prioriser
  - Fonctionnel
  - Testable

La **composition** : un jeu d'enfant !

```
function AppLayout(props) {
  return <div className="app">{props.children}</div>;
}

function MenuItem(props) {
  return <li><a href={props.link}>{props.text.toUpperCase()}</a></li>;
}

function Sidebar() {
  return (
    <ul className="sidebar">
      <MenuItem link="/" text="Accueil" />
      <MenuItem link="/about" text="A propos" />
    </ul>
  );
}

function Site() {
  return (
    <AppLayout>
      <Sidebar />
    </AppLayout>
  );
}
```

- ACCUEIL
- A PROPOS

```
<div class="app">
  <ul class="sidebar">
    <li><a href="/">ACCUEIL</a></li>
    <li><a href="/about">A PROPOS</a></li>
  </ul>
</div>
```


<html/>
ReactDOM
React

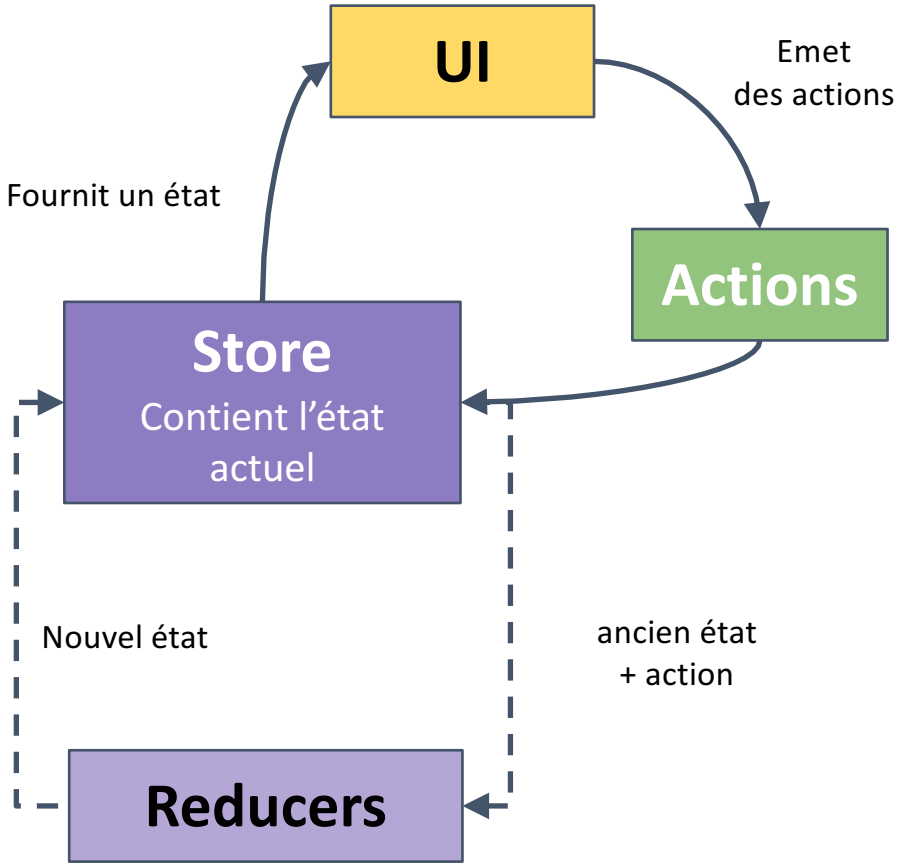
Redux

- Librairie **Javascript** (25k étoiles)
- Gestion d'**état** d'applications JavaScript
- Créée par Dan Abramov (2015)
- **Open-source** (https://github.com/reactjs/redux)
- Très utilisé avec React (react-redux)
- Mais pas seulement

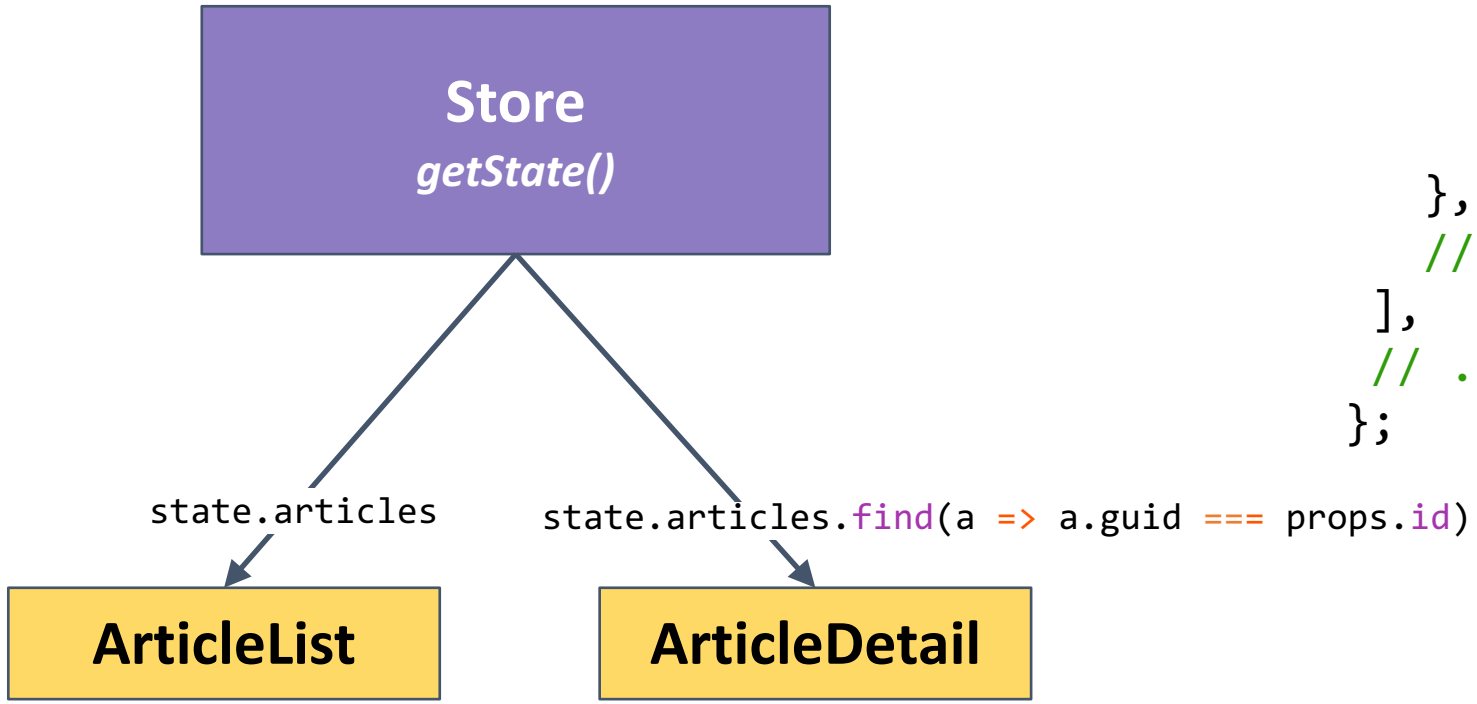
Redux

Redux

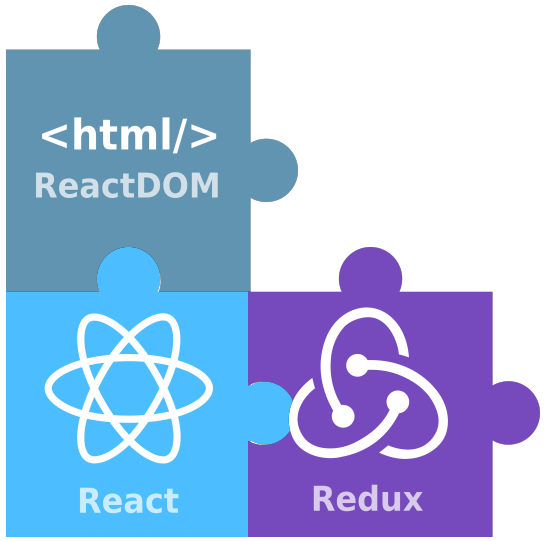
- Gère l'état de l'application
- Déterministe
- 3 principes fondamentaux
  - L'**état** est stocké dans un **unique** objet JS, le **Store**
  - L'**état** est en **lecture seule**. Pour le modifier, on émet des **actions**
  - Le store est modifié avec des fonctions **pures** appelées **réducteurs**



Redux par l’exemple

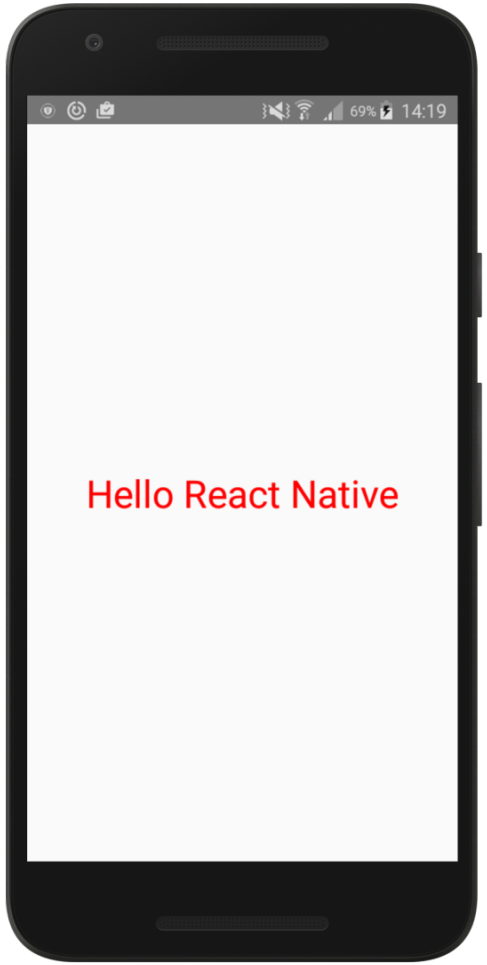



```
const state = {
 articles: [
   {
        guid: ‘myId’,
        title: ‘myTitle’,
        // ...
   },
   // ...
 ],
 // ...
};
```


<html/>
ReactDOM
React
Redux

Une application minimale RN

```
import React from 'react';
import { View, Text, AppRegistry } from 'react-native';

function App() {
  return (
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
      <Text style={{ color: 'red', fontSize: 32 }}>
        Hello React Native
      </Text>
    </View>
  );
}

AppRegistry.registerComponent('monprojet', () => App);
```



React ...**Native** !

- Librairie + CLI pour développer des applications natives en JS (+ de 40k étoiles)
- Créée par Facebook (annoncée en 2015)
- Open-source (https://github.com/facebook/react-native)
- Simple
- Performante
- Fun !

```
npm install -g react-native-cli
react-native init monprojet
cd monprojet
react-native run-android
```

Qui l’utilise ?

Facebook
Facebook Ads Manager
Facebook Groups

Instagram

Airnbnb

Discord

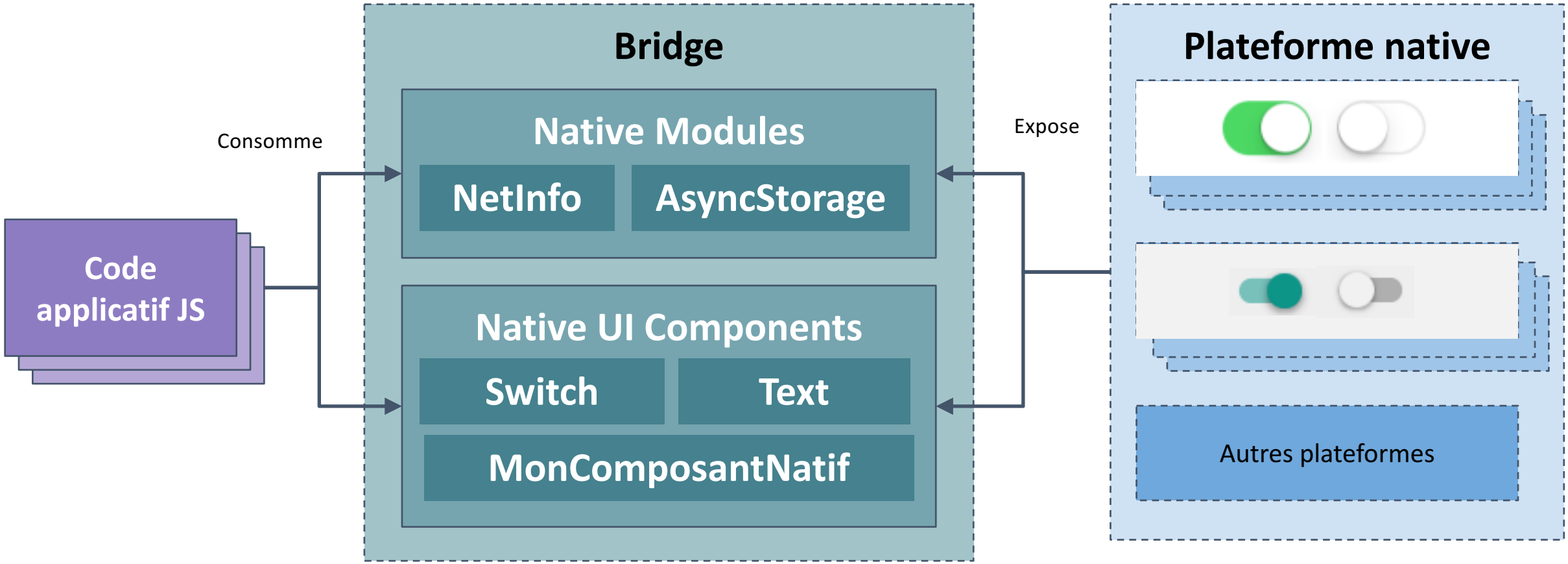
Soundcloud Pulse

Baidu

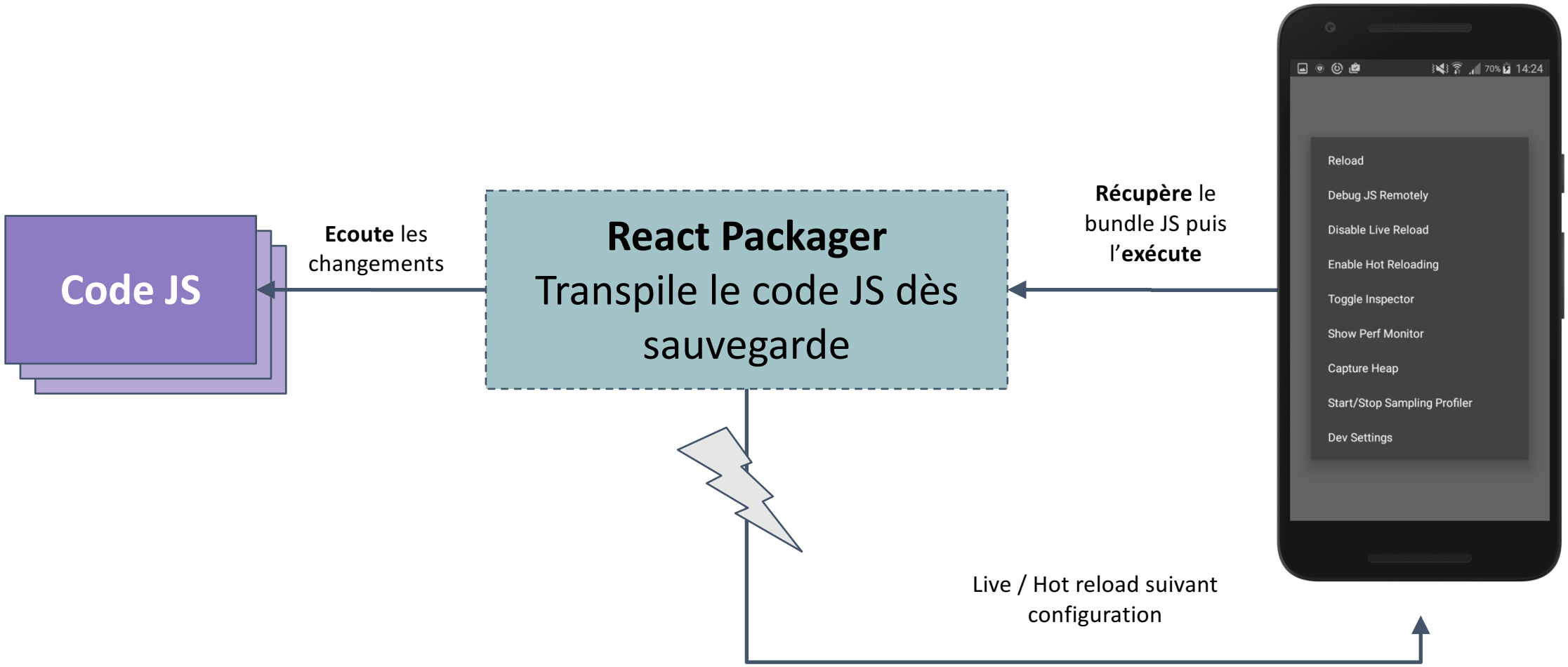
La librairie

- Des API natives
  - NetInfo
  - AsyncStorage
- Des composants
  - Natifs (View, Text, TouchableNativeFeedback, ...)
  - Implémentés en JS (Button, …)
  - Fournis par la communauté, https://github.com/airbnb/react-native-maps
- Une gestion des styles proche du web
  - Propriétés supportées nombreuses (color, backgroundColor, fontSize, …)
  - Layouting Flexbox

Le bridge

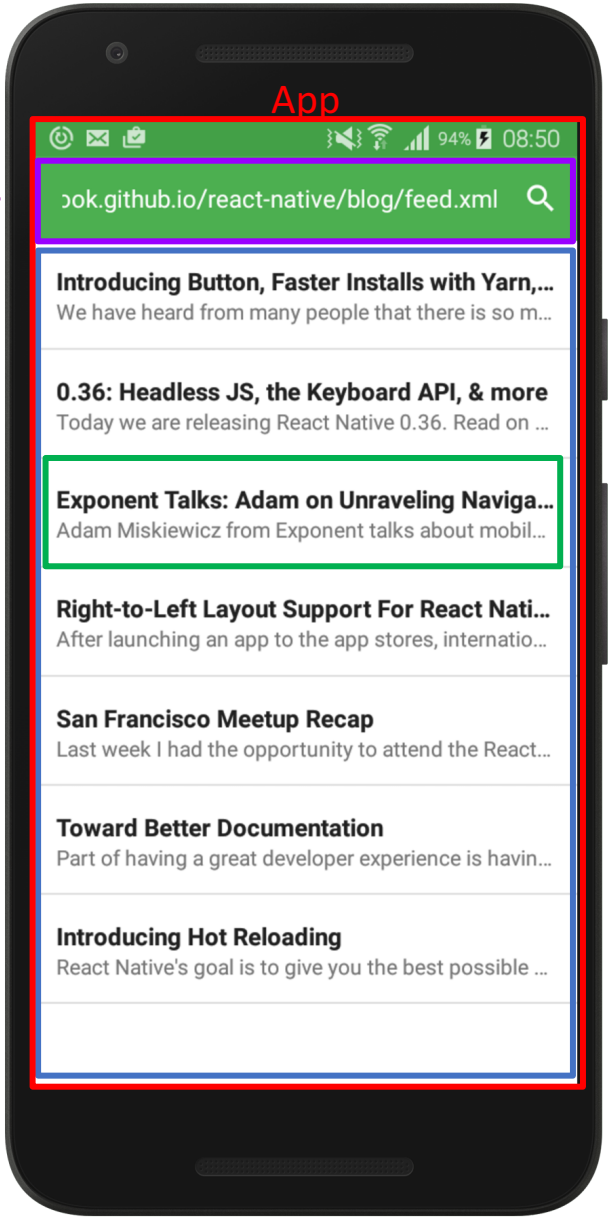

Bridge
Native Modules
NetInfo
AsyncStorage
Native UI Components
Switch
Text
MonComposantNatif
Consomme
Expose
Code applicatif JS
Plateforme native
Autres plateformes

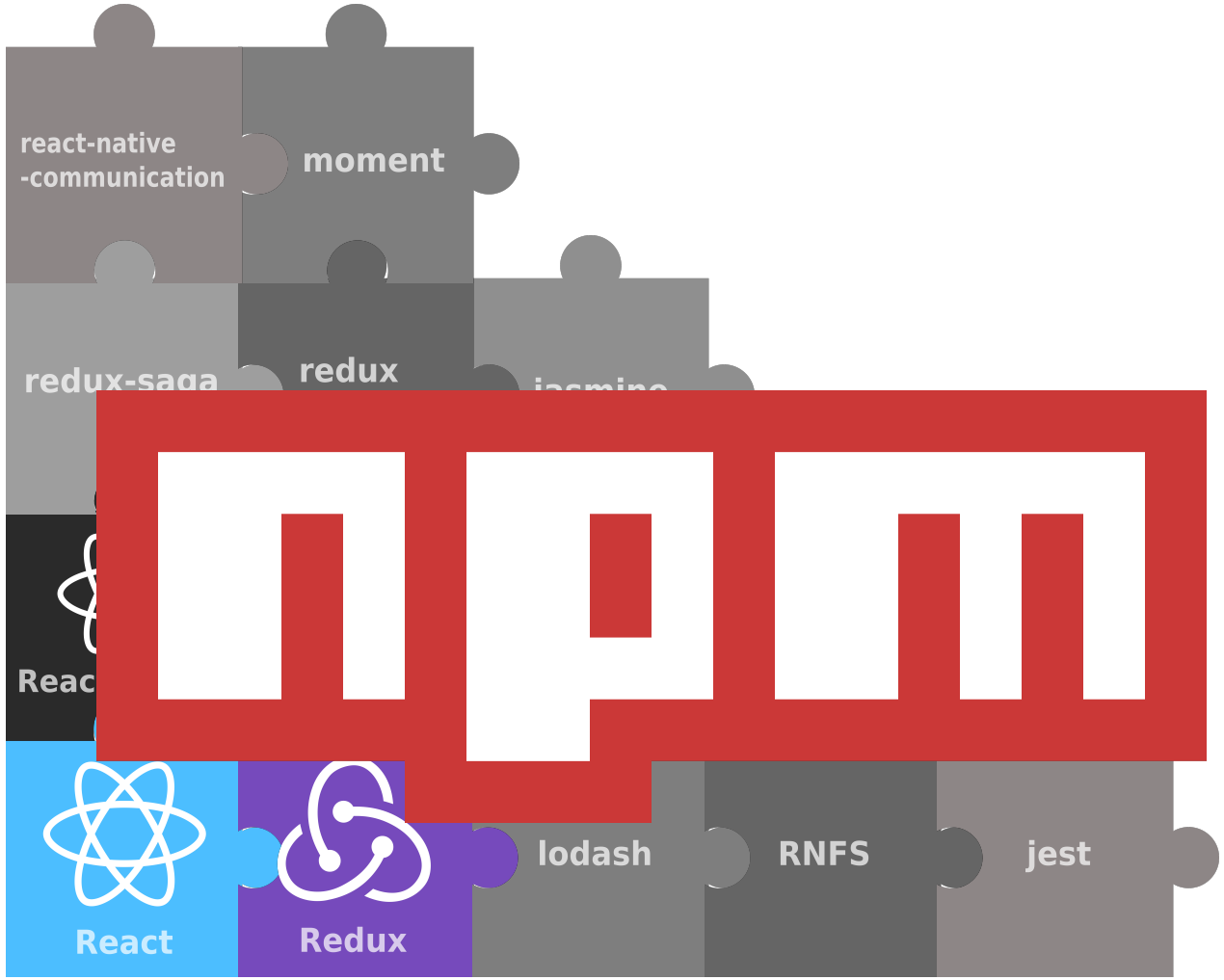
Le packager


Code JS
Ecoute les changements
React Packager
Transpile le code JS dès sauvegarde
Récupère le bundle JS puis l'exécute
Reload
Debug JS Remotely
Disable Live Reload
Enable Hot Reloading
Toggle Inspector
Show Perf Monitor
Capture Heap
Start/Stop Sampling Profiler
Dev Settings
Live / Hot reload suivant configuration

Think in React Native !

- Maquettage (par écran)
- Découpe UI en hiérarchie
- Créer une version statique en react-native
- Définir l'état minimal nécessaire par composant
- Identifier où se trouve l'état (composant / store)
- Gestion des événements UI (MAJ de l'état)




react-native
-communication
moment
redux-saga
redux
jasmine
Reac
npm
React
Redux
lodash
RNFS
jest


IT'S SHOW TIME
BABY
memegenerator.net

Notre point de vue aujourd'hui

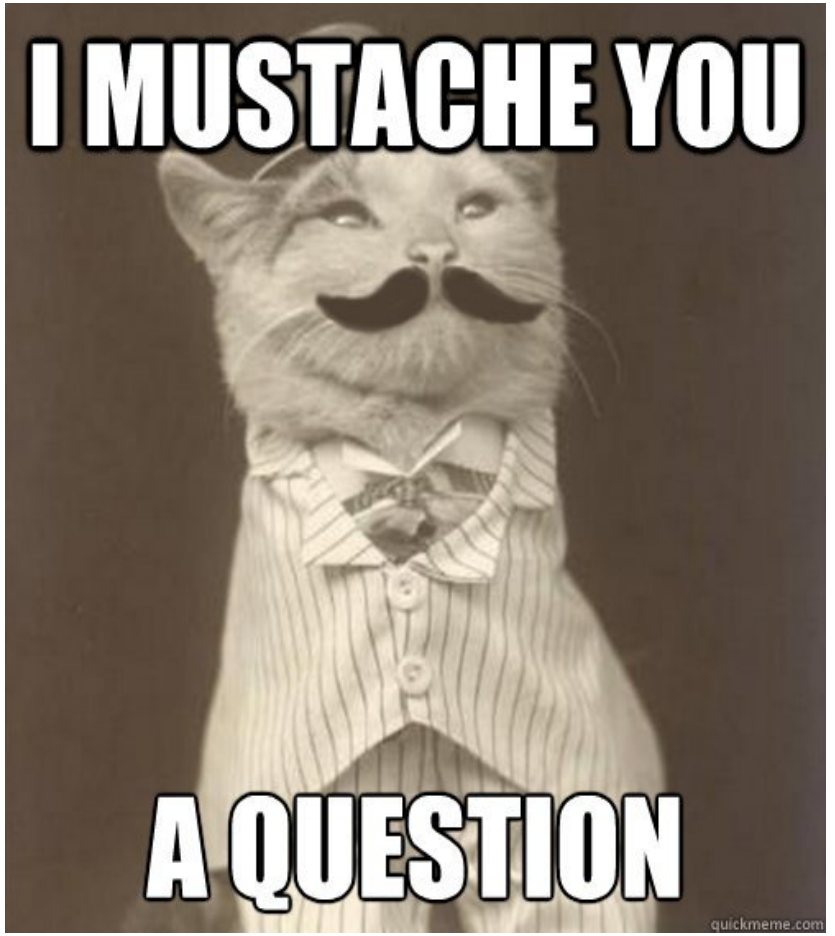
- Axes d'amélioration
  - Plus de composants intégrés
  - Maturité
  - Meilleur support Windows (dév)
- Points positifs
  - Rendu natif (Fluidité et performance)
  - Rapidité de développement
  - Le JavaScript et son univers

Prochainement

https://github.com/proxilabs

https://twitter.com/julien_leicher

https://twitter.com/QBisson



Bibliographie

https://facebook.github.io/react/

https://facebook.github.io/react-native/

http://redux.js.org/

Très bonne série pour apprendre React / Redux

Organisation d'une application Redux

http://stateofjs.com/

https://medium.com/@dan_abramov/you-might-not-need-redux-be46360cf367