

REACT

@rimthong & @BenjaminDreux

QU'EST-CE QUE REACT?

- Techno de Facebook.
- On l'appelle le V de MVC, mais pas exact.
- Ça ressemble un peu aux Web Components.

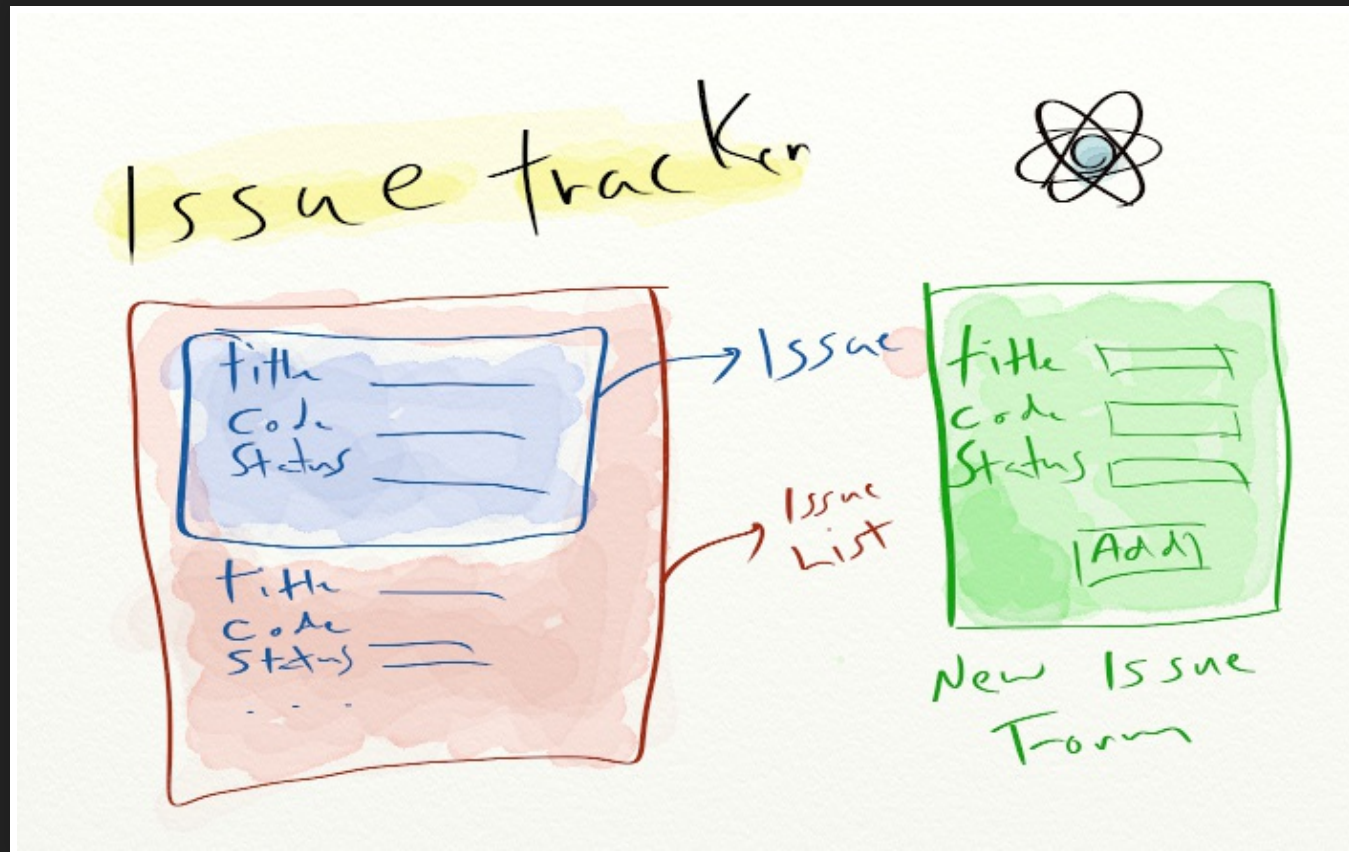
REACT EST... ÉTRANGE

- Controverse sur les bonnes pratiques (onClick)
- HTML dans le Javascript (JSX)

REACT EST... RAPIDE

- Utilisation de virtual DOM

FAISONS UNE APP!



VUE D'UN ISSUE

```
var Issue = React.createClass({
  render: function() {
    return (
      <div>
        <div>Title: {this.props.issue.title}</div>
        <div>Status: {this.props.issue.status}</div>
        <div>Reporter: {this.props.issue.reporter}</div>
        <div>Description: {this.props.issue.description}</div>
      </div>
    );
  }
});
```

LISTE DES ISSUES

```
var IssueList = React.createClass({
  getInitialState: function() { return {issues: [{id:1, ...}]}; },
  render: function(){
    var elements = this.state.issues.map(function(issue){
      return <li key={issue.id}> <Issue issue={issue}/> </li>
    });
    return <ul> {elements} </ul>
  }
});
```

FORMULAIRE

```
var NewIssueForm = React.createClass({
  addIssue:function(e){
    var issue = {title: this.refs.title.getDOMNode().value};
    ...
  },
  render: function() {
    return
      <form className="new-issue-form" onSubmit={this.addIssue}>
        <div className="form-group">
          <label > Title: </label>
          <input className="form-control" ref="title" />
        </div>
        ...
        <button>Add</button>
      </form>
  }
});
```


C'est tout pour React

BONNE JOURNÉE

... presque

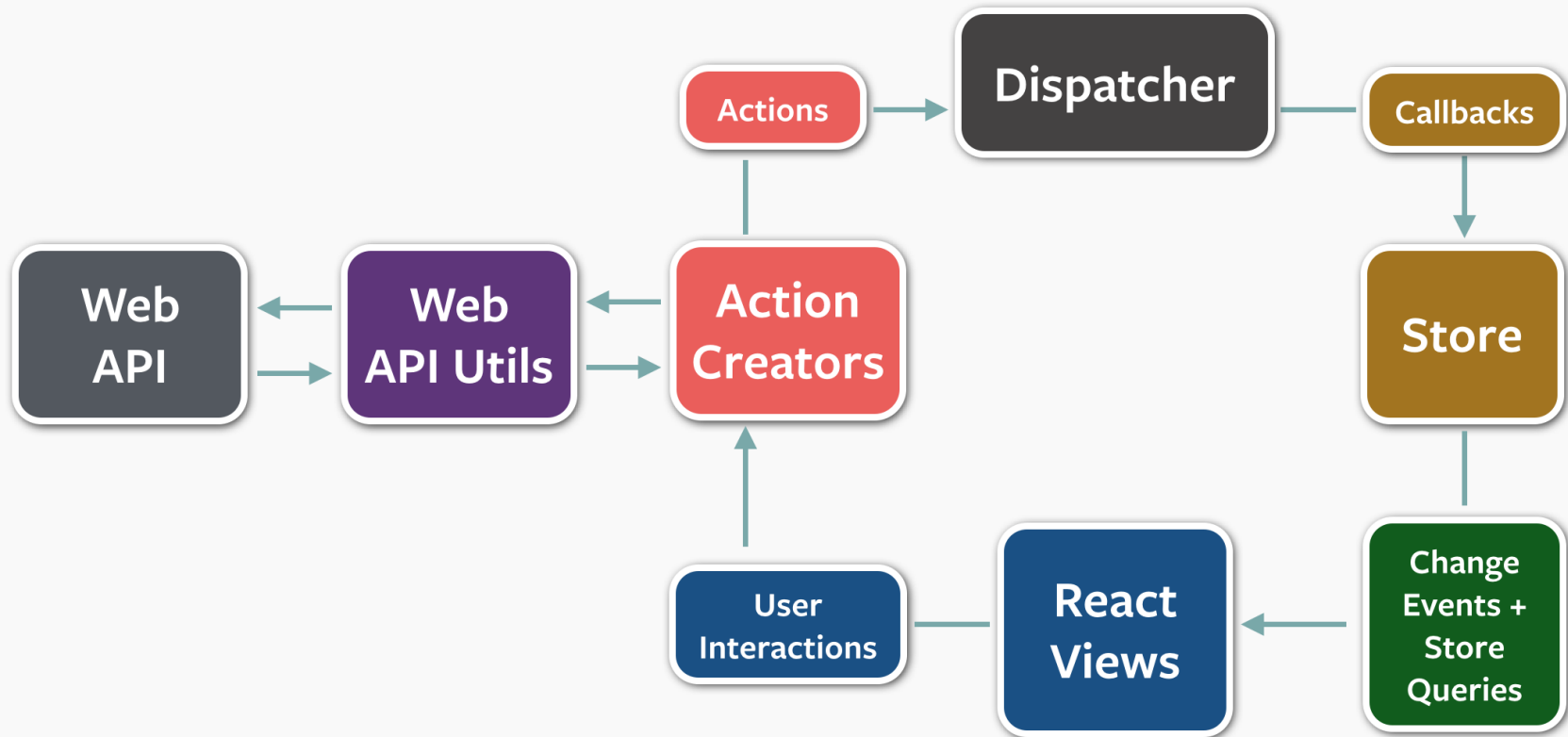
REACT

- Gros trip de Dev
- render gratuit
- permet le pattern flux

FLUX

- Comment on branche le tout?

WHAT THE FLUX??



DISPATCHER

```
var Dispatcher = require("flux").Dispatcher;  
var IssueDispatcher = _.extend(new Dispatcher(), {});
```

REACTION DU STORE

```
IssueDispatcher.register(function(action){  
  switch(action.type){  
    case 'issue:add':  
      IssueStore.addIssue(action.payload);  
      break;  
    case 'issue:fetch':  
      IssueStore.set(action.payload);  
      break;  
  }  
});
```

STORE

```
var CHANGE_EVENT = 'changed';
var IssueStore = _.extend({}, EventEmitter.prototype, {
  issues: [],
  emitChange: function() {this.emit(CHANGE_EVENT);},
  addChangeListener: function(callback){this.on(CHANGE_EVENT, callback)},
  removeChangeListener: function(callback){this.removeListener(CHANGE_EVENT, callback)},
  getIssues: function(){return this.issues},
  addIssue:function(issue){
    this.issues.push(issue);
    this.emitChange();
  },
  set:function(newIssues){
    this.issues = newIssues;
    this.emitChange();
  }
});
```

COMPOSANTES

```
var NewIssueForm = React.createClass({
  addIssue:function(e){
    var issue = {title: this.refs.title.getDOMNode().value};
    IssueActionCreator.createNewIssue(issue);
    e.stopPropagation(); e.preventDefault();
  },
  render: function() {
    return (
      <form className="new-issue-form" onSubmit={this.addIssue}>
        <div className="form-group">
          <label > Title: </label>
          <input className="form-control" ref="title" />
        </div>
        <button>Add</button>
      </form>
    );
  }
});
```


ACTION CREATOR

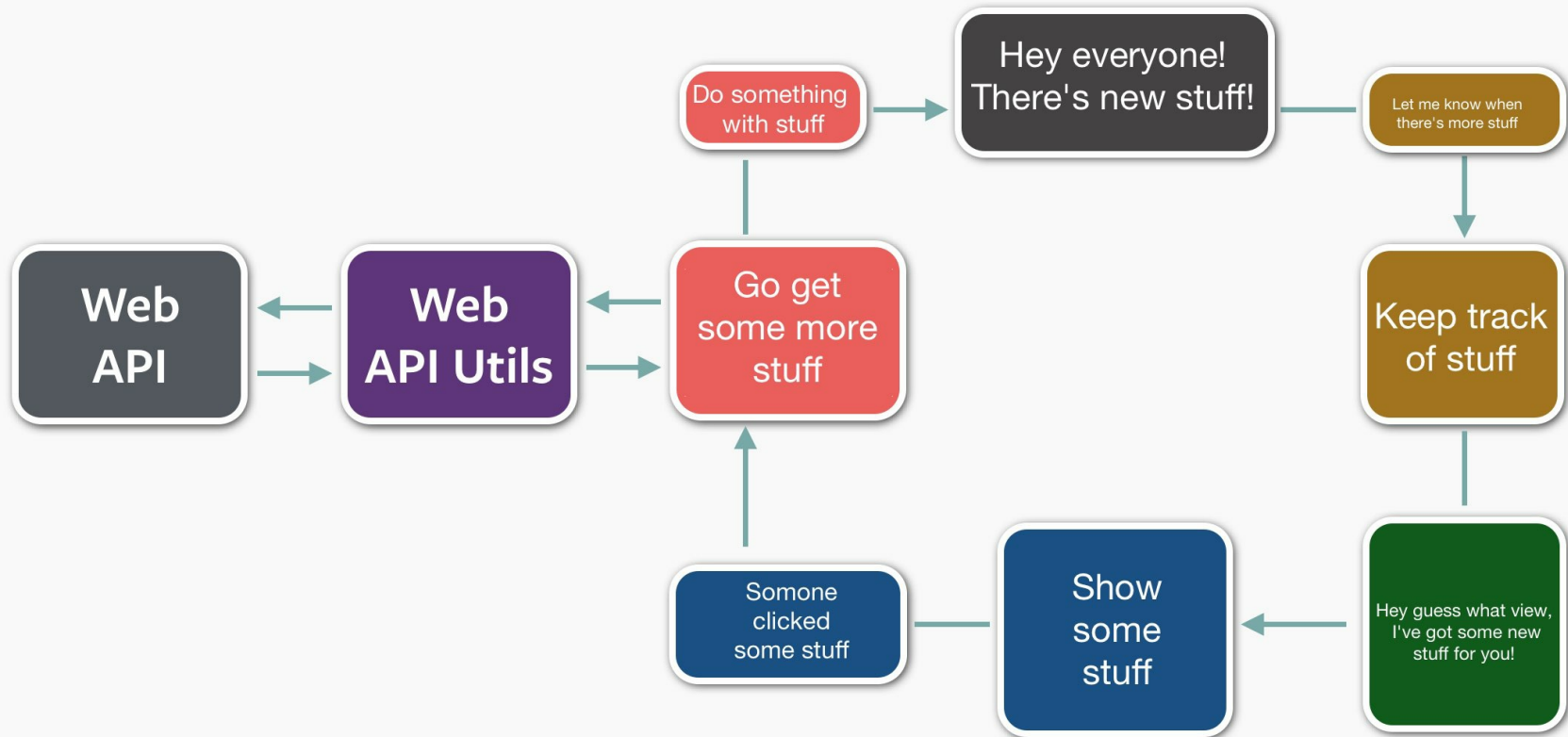
```
var IssueActionCreator = {
  createNewIssue:function(issue){
    //Appel au backend
    IssueDispatcher.dispatch({
      type:"issue:add",
      payload: issue
    });
  },
  fetchIssues:function(){
    // Appel au backend
    var issues = [{id:1, title:"Changer la couleur des bouton"},
                  {id:2, title:"J'aime pas la couleur du bouton"}]

    IssueDispatcher.dispatch({
      type:'issue:fetch',
      payload: issues
    });
  }
}
```

INTERFACE CLIENT API

```
// INSERT YOUR CODE HERE
```

FLUX, C'EST JUSTE ÇA !!



TEST

Jest

- Jasmine ++ (it.only, xit, pit, xdescribe)
- Auto mocking
- Output de comparaison json A la elixir
- Pas un requis pour tester du REACT
- Implementation de dom

React.addon.TestUtils

- Rendu de composant
- Simulation d'evenement utilisateur (key, click...)
- Mock de composant

EXEMPLE DE TEST

```
jest.dontMock( '../js/NewIssueForm' );  
  
it.only(trigger an action on click, function(){  
  
  React = require('react/addons');  
  TestUtils = React.addons.TestUtils;  
  
  var form = TestUtils.renderIntoDocument(<NewIssueForm/>);  
  var button = TestUtils.findRenderedDOMComponentWithTag(form, 'bu  
  
  TestUtils.Simulate.submit(button)  
  
  expect(IssueActionCreator.createNewIssue.mock.calls.length).toBe  
});
```

BROWSERIFY

- Module A la node dans le frontend aujourd'hui
- Pipeline de transformation de code (jsx, coffee, minify, uglify...)
- package.json:

```
"scripts": {
  "start": "watchify -o bundle.js -v -d js/*",
  "test": "jest",
  "build": "NODE_ENV=production browserify . | uglifyjs -cm > bu
},
"browserify": {
  "transform": [
    "envify",
    "reactify"
  ]
},
```

ALTERNATIVES

- Raynos/mercury (Uber) -> immutabilité par défaut
- Riot.js (Muut, Disqus like) -> Diff simpliste
- Mithril -> Angular avec Virtual DOM
- Ractive -> Backbone avec Virtual DOM
- OM ou Quiescent (Interface Clojure script pour React)
- ELM (language appart)

REACT VS. WEB COMPONENTS

Les techno de base ne sont pas supporté nativement
aujourd'hui, voir:

- <http://caniuse.com/#search=object.observe>
- <http://caniuse.com/#search=shadowdom>
- <http://caniuse.com/#search=import>
- <http://caniuse.com/#search=template> (moins vrai)

Sauf pour chrome

DES QUESTIONS?