

Programmation Réactive

Principes fondamentaux et application au Web

Plan

- ▶ Introduction
- ▶ Les principes de la programmation réactive
- ▶ Exemples
- ▶ Programmation réactive et Web
- ▶ React

Qu'est ce que la programmation réactive ?

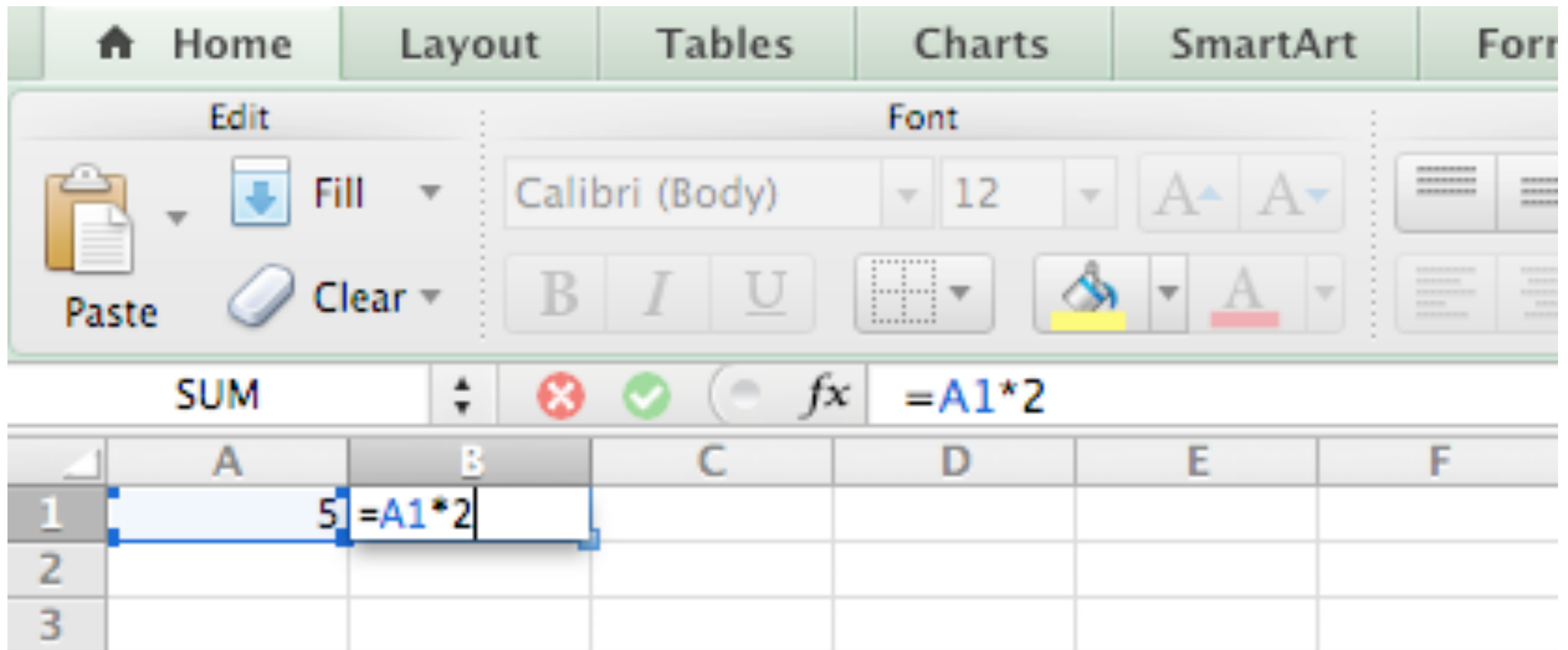
Une approche visant à mieux gérer les flux

Deux types de flux

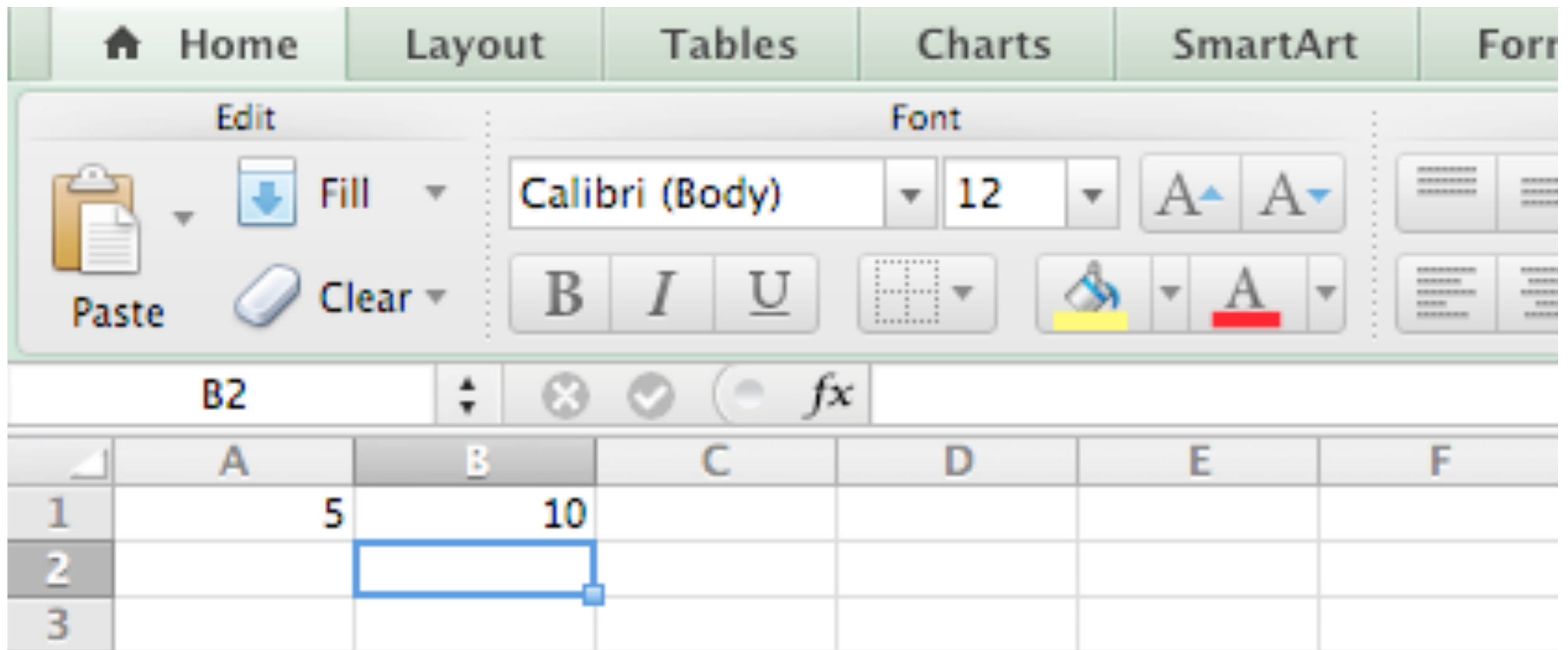
- ▶ Des événements discrets : frappe clavier
- ▶ Des événements continus ou *comportements* : position souris

Idée : dépasser les callbacks ou le patron Observer.

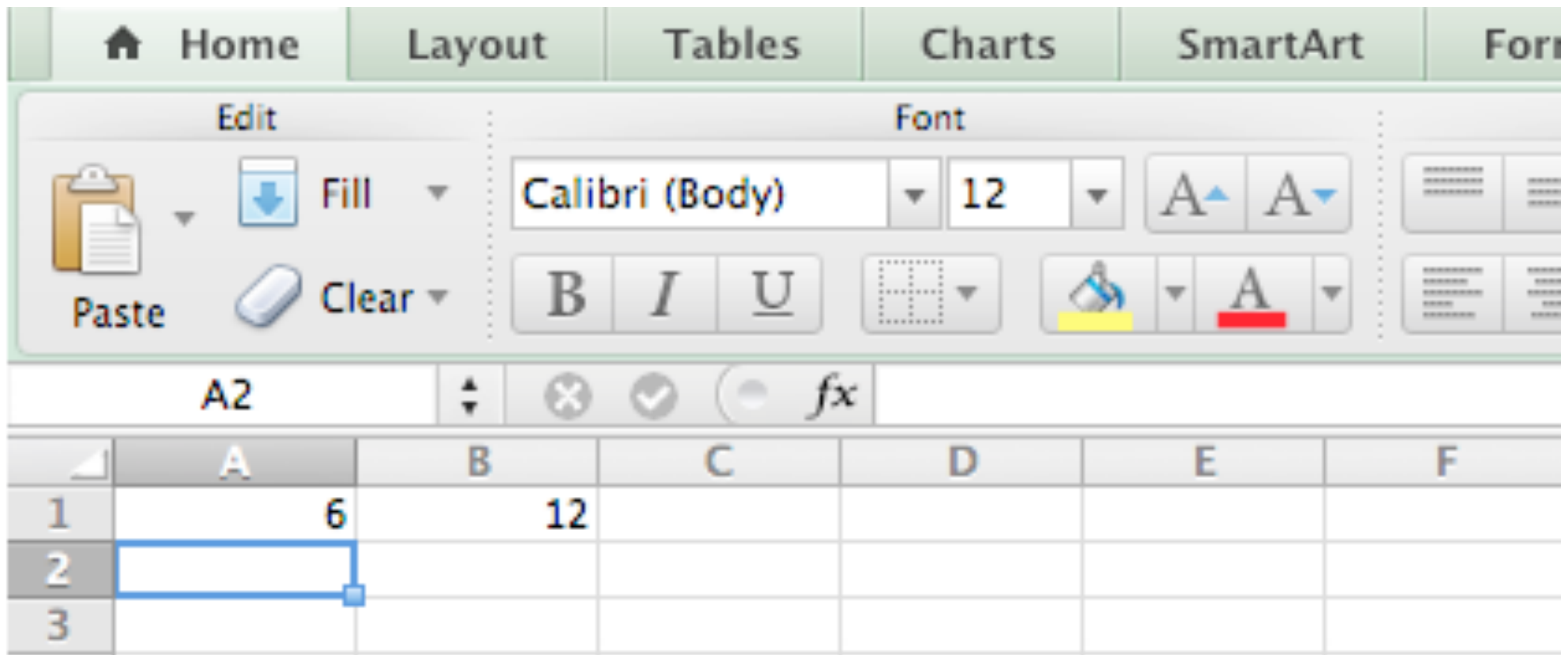
Ou avez vous vu ça ?



Ou avez vous vu ça ?



Ou avez vous vu ça ?



Pourquoi la programmation réactive ?

- ▶ Gestion d'évènements et de l'asynchrone
- ▶ Faible latence (contraintes sur les temps de réponse)
- ▶ Flux de données importants (et rapides).
- ▶ Tolérance aux fautes

Exemples

À vous

Pourquoi la prog. réactive sur le Web ?

The screenshot displays a comprehensive web-based ticketing system interface. On the left, a sidebar provides navigation options including 'IMs', 'My Tickets' (12), 'Tickets I Follow' (34), 'My Teams' (9), 'Unassigned Tickets' (132), 'Tickets by User' (5), and 'All Tickets' (132). The 'All Tickets' section is further categorized by user: Harrison Newman (132), Tyler Weston (132), Ellis Glover (132), David Benson (132), and Demi Carroll (132). Below this, 'SECONDARY TICKETS' are listed: 'Mine On Hold' (26) and 'All On Hold' (324). The main content area features a search bar ('Search Tickets or everything') and filters for 'ORDER BY: URGENCY', 'GROUP BY: NONE', and 'TICKET VIEW: DEFAULT'. A table of tickets is shown, with columns for 'Status', 'Agent', 'Department', and 'Flags'. The tickets listed include: 1. 'Count records per hour within a time span' (#34528) by Nelson Manning (BitDefender,CEO) with a status of 1 and 'Last reply: 25m ago'. 2. 'Problems using windows 7 built-in virtual wifi hotspot' (#34528) by Sophie Norton (BitDefender,CEO) with a status of 2, 'SLA: 25m 4h33m', and 'Last reply: 25m ago'. 3. 'JIRA REST API to get work log - "You do not have the permission to see the specified issue"' (#34528) by Sophie Norton (BitDefender,CEO) with a status of 3, 'SLA Failed', and 'Last reply: 25m ago'. 4. 'Count records per hour within a time span' (#34528) by Nelson Manning with a status of 4, 'Last reply: 25m ago'. 5. 'How could manage the communication between a WCF thread and another thread?' (#34528) by Nelson Manning with a status of 5, 'Last reply: 25m ago'. The right sidebar contains a 'TICKETS' section with 'MERGE' and 'SPLIT' options, a 'BitBucket' icon, a 'STATUS: AWAITING AGENT' dropdown, and sections for 'TICKET PROPERTIES' (LANGUAGE: English, PRODUCT: DeskPRO Cloud, BRAND: DeskPro Cloud), 'PARTICIPANTS', and 'BILLING'.

Les bibliothèques Javascript

[Guide](#) [API](#) [Examples](#) [Blog](#) [Community](#) ▼



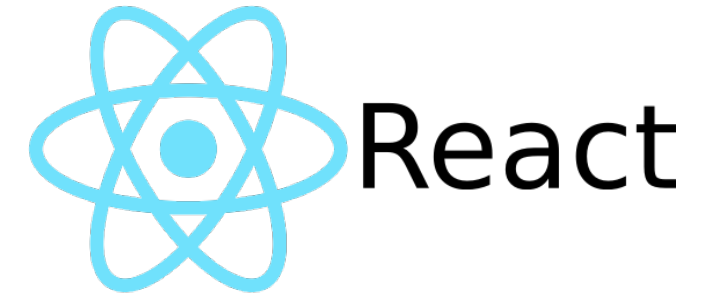
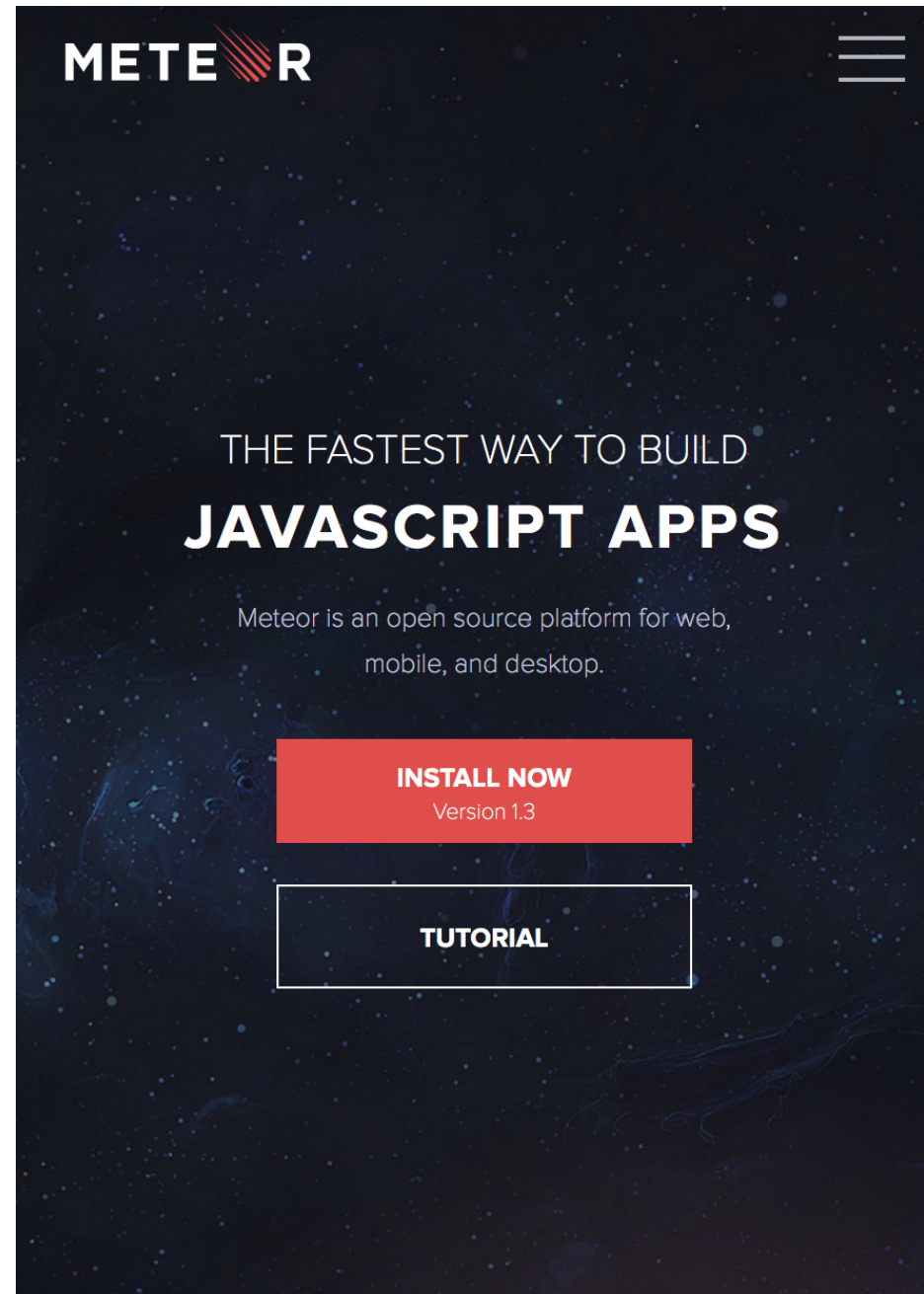
Vue.js

Reactive Components for Modern Web Interfaces

Install v1.0.24

[Follow @vuejs](#) [Star](#) 18,631 [Support Vue.js](#)

[中文](#) | [日本語](#) | [Italiano](#)



Plan

- ▶ Introduction
- ▶ **Les principes de la programmation réactive**
- ▶ Exemples
- ▶ Programmation réactive et Web
- ▶ React

Les principes de base

- ▶ Responsive,
- ▶ Résilient,
- ▶ Élastique,
- ▶ Orienté message

Responsive

- ▶ Réponse en temps voulu, si possible
- ▶ Temps de réponses rapides et fiables (limites hautes)

Résilient

- ▶ Résiste à l'échec
- ▶ Principes :
Réplication, conteneurs, isolement, délégation
- ▶ On fait en sorte qu'un échec n'impacte qu'un seul composant

Élastique

Le système reste réactif en cas de variation de la charge de travail.

- ▶ Pas de point central
- ▶ Pas de goulot
- ▶ Distribution des entrées entre composants

Message Driven

- ▶ Passage de messages asynchrones
-> Couplage faible, isolation
- ▶ Pas de blocage, les composants consomment les ressources quand ils peuvent

Plan

- ▶ Introduction
- ▶ Les principes de la programmation réactive
- ▶ **En pratique les transformations de flux**
- ▶ Programmation réactive et Web
- ▶ React

Un concept important : l'immuabilité

Objet immuable (Immutable object)

- ▶ Objet dont l'état ne peut pas être modifié après sa création
- ▶ Opposé d'objet variable

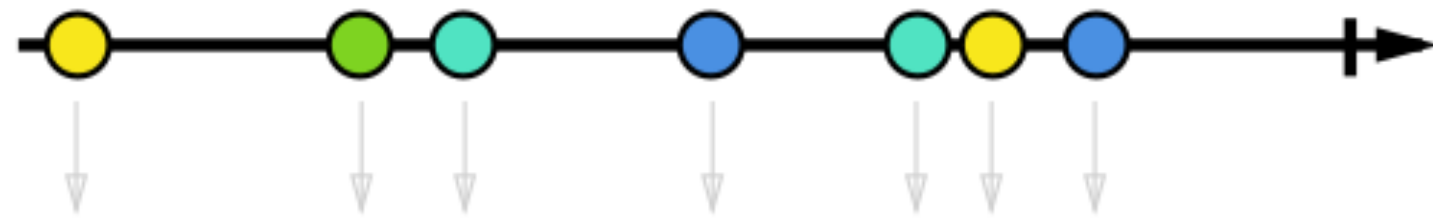
Facilite la prog. purement fonctionnelle (pratique pour plein de choses, évite les effets de bords, facilite le undo)

Une seule source de “vérité”

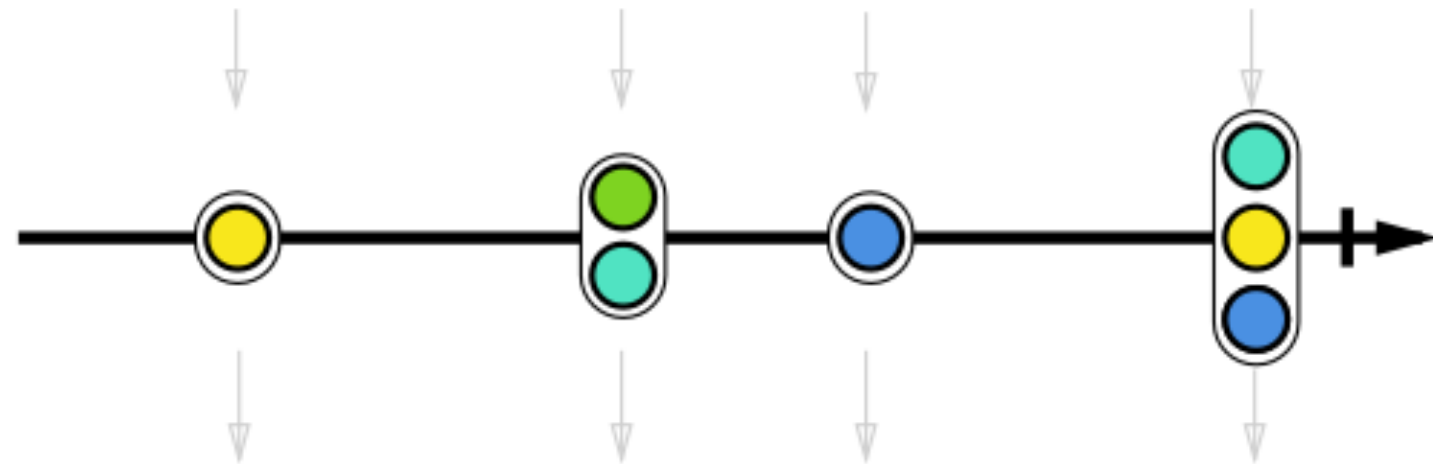
Facilite le caching

Mais ce n'est pas forcément assez : <https://codewords.recurse.com/issues/six/immutability-is-not-enough>

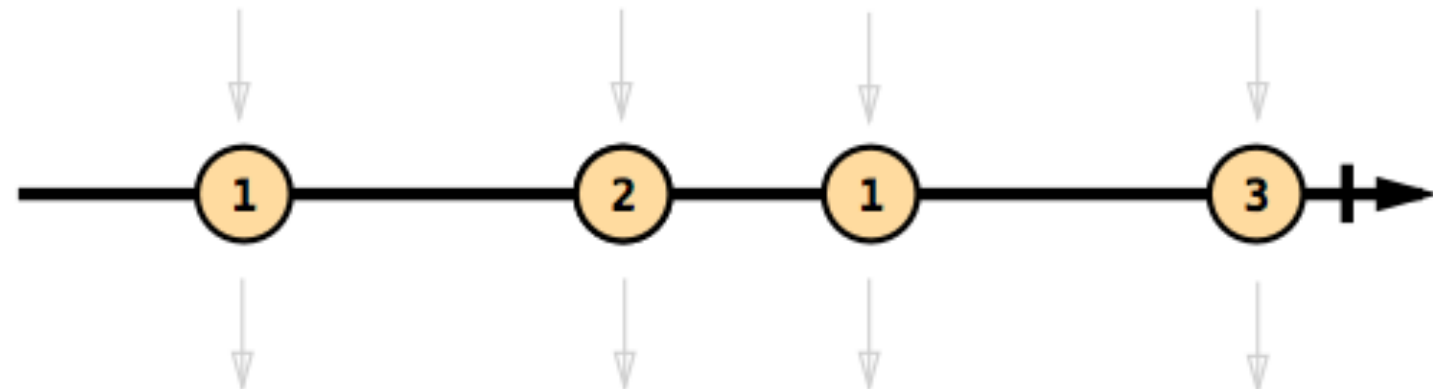
Click stream



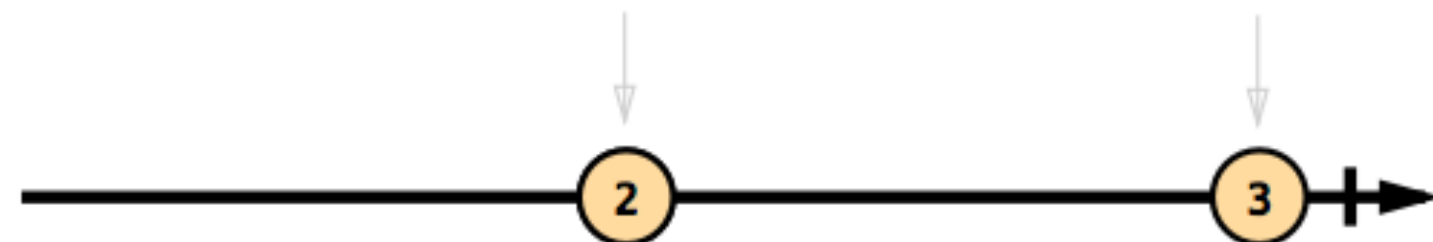
`buffer(clickStream.throttle(250ms))`



`map('get length of list')`



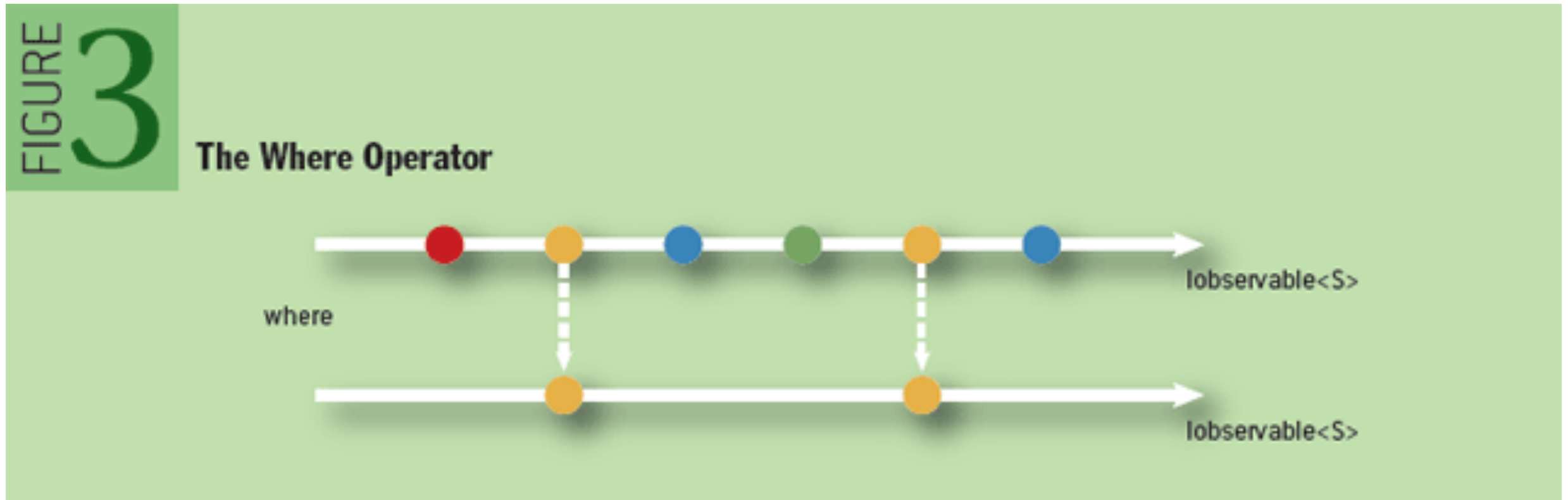
`filter(x >= 2)`



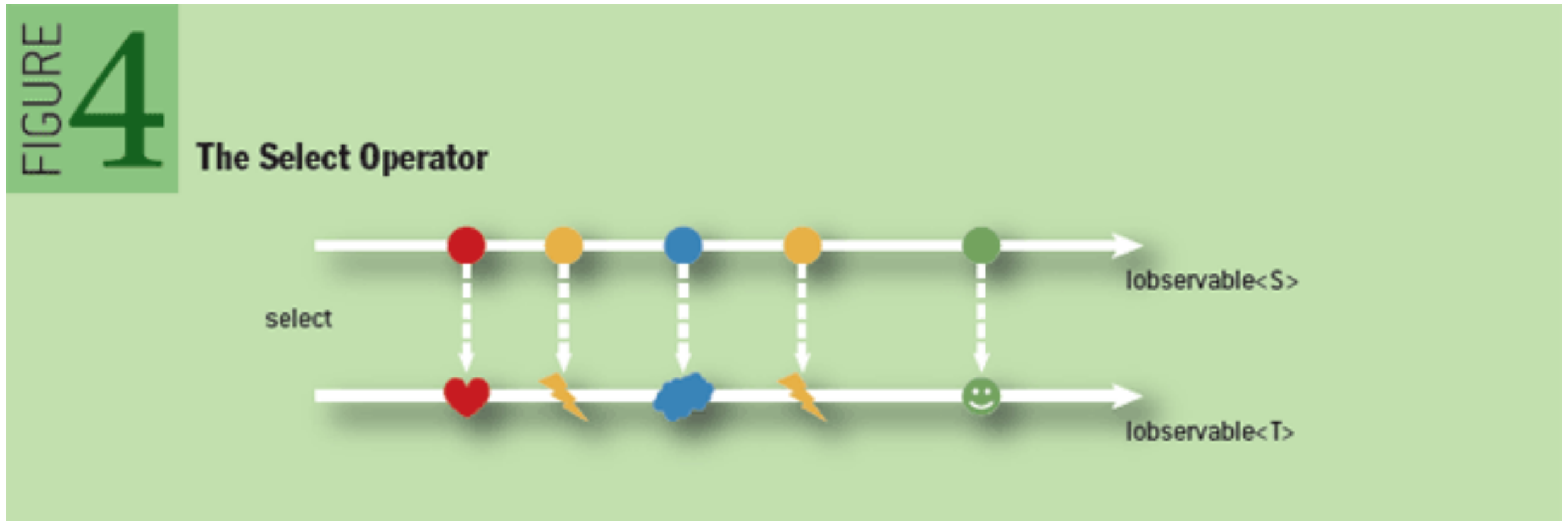
Multiple clicks stream

Un exemple de
transformation

Where



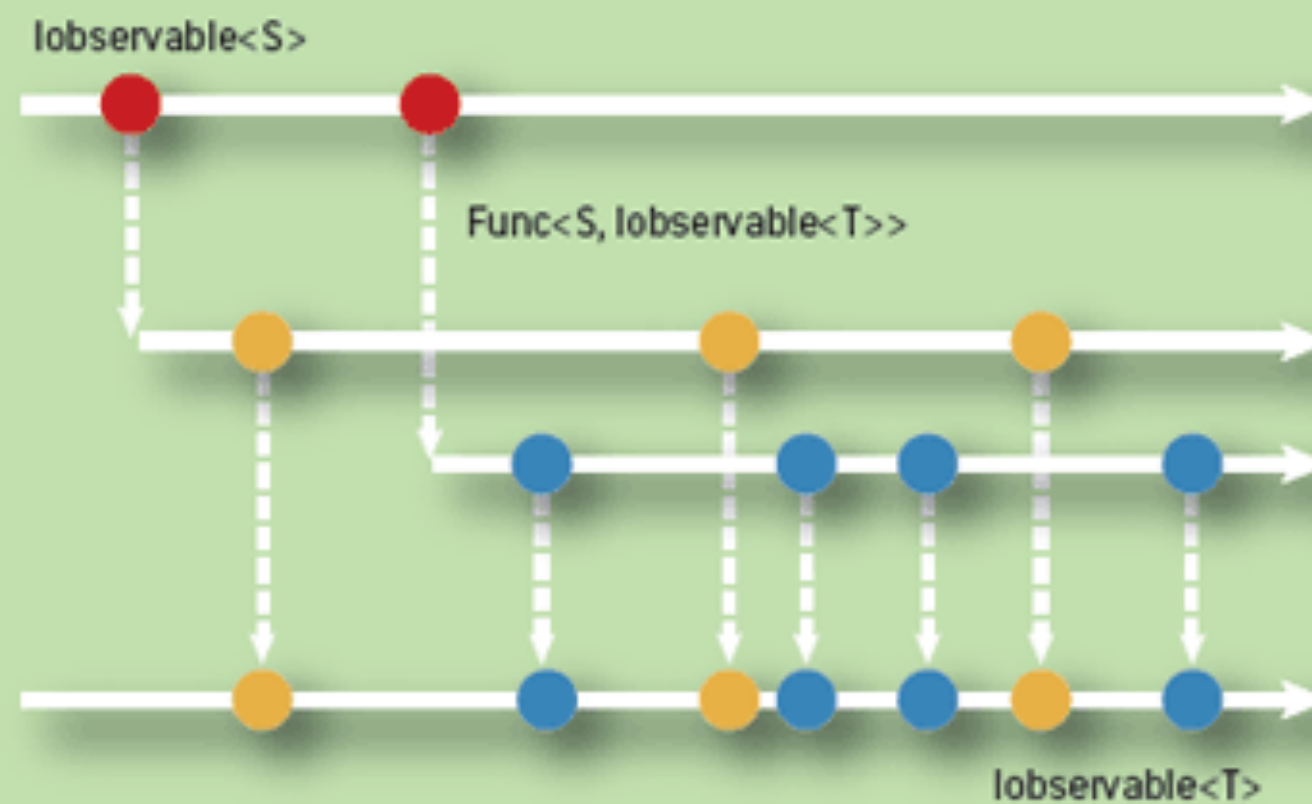
Select



SelectMany : plusieurs flux

FIGURE 5

The SelectMany Operator



Throttle

FIGURE 9

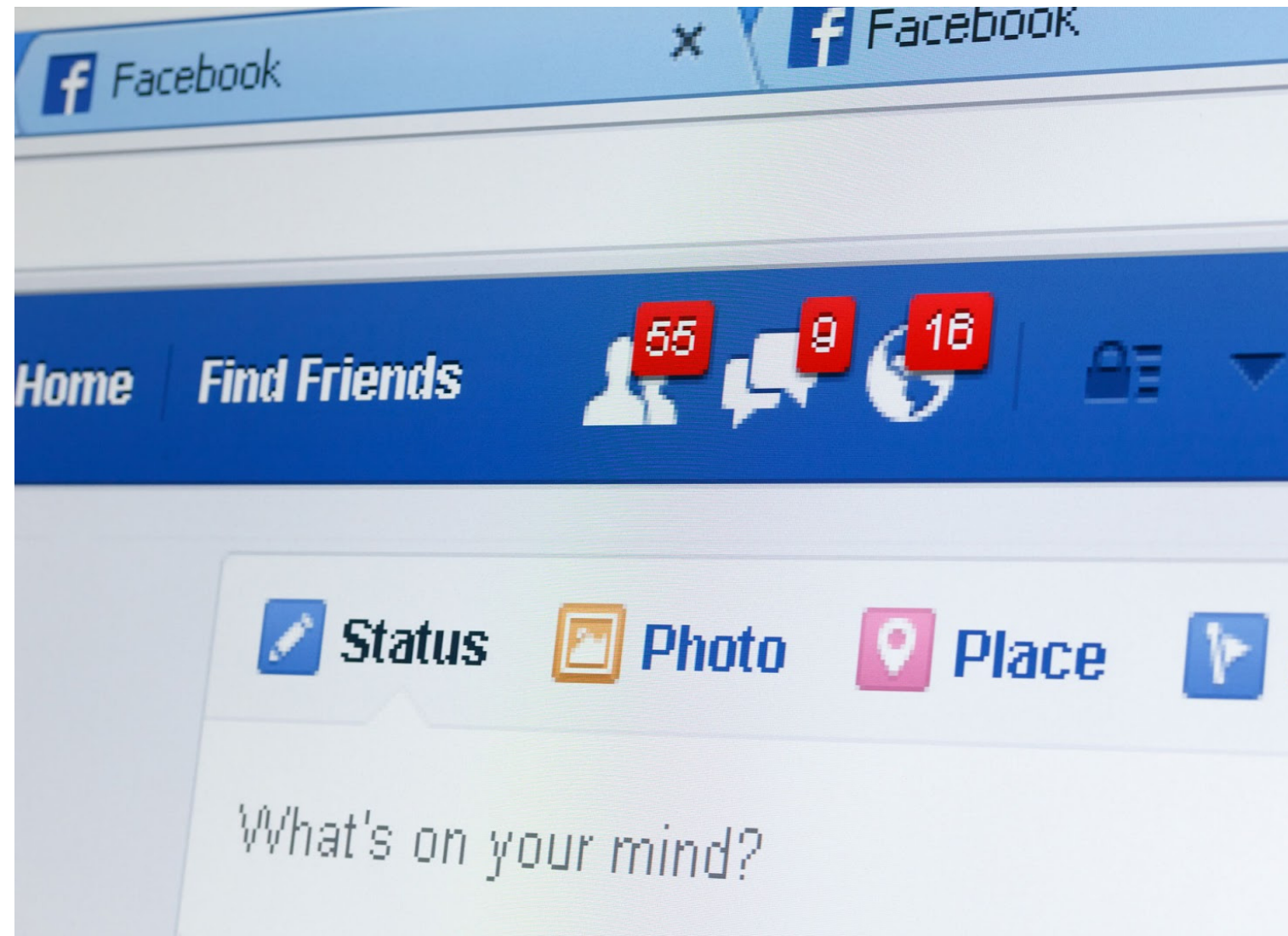
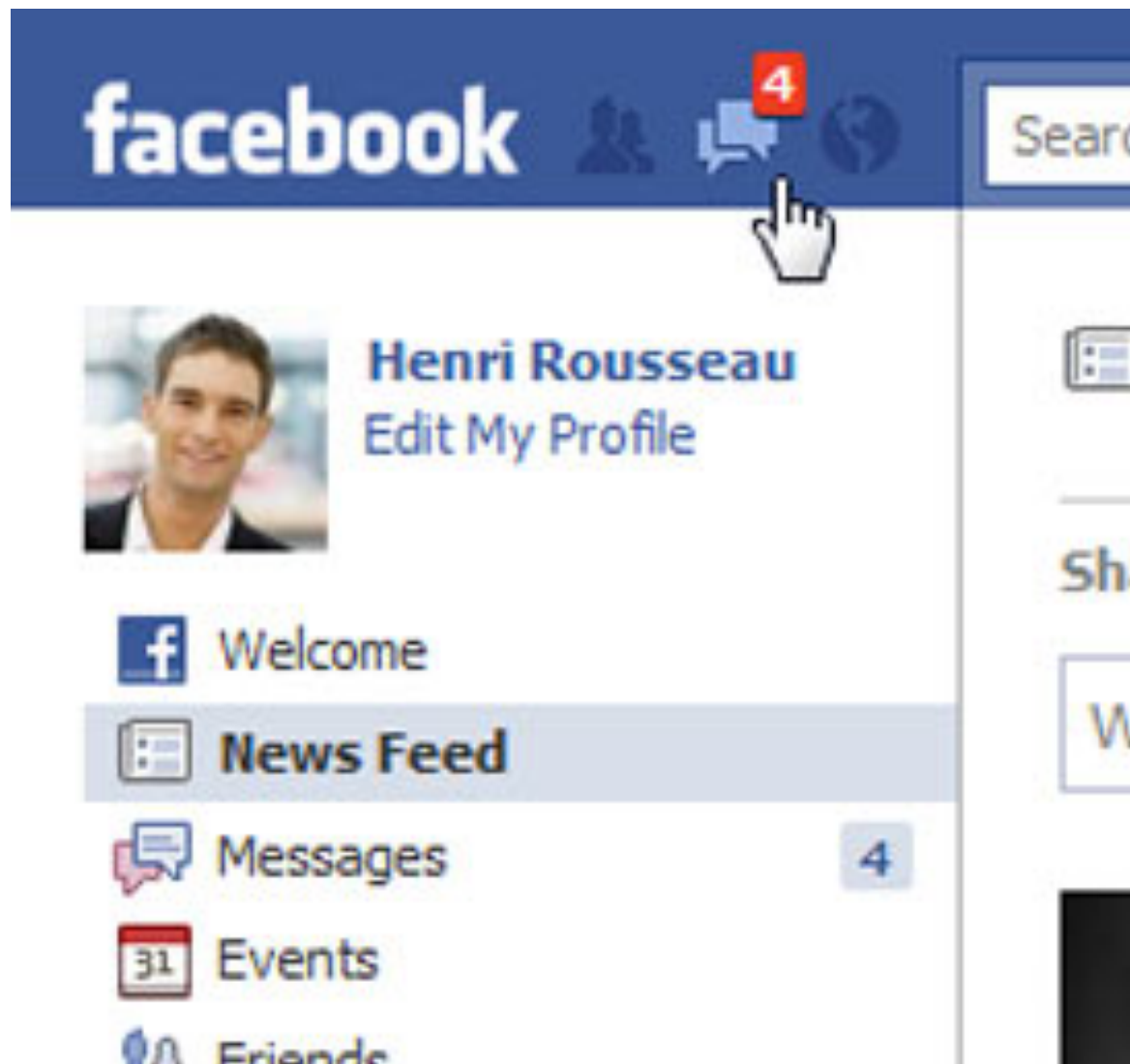
The Throttle Operator



Plan

- ▶ Introduction
- ▶ Les principes de la programmation réactive
- ▶ Exemples
- ▶ **Programmation réactive et Web**
- ▶ React

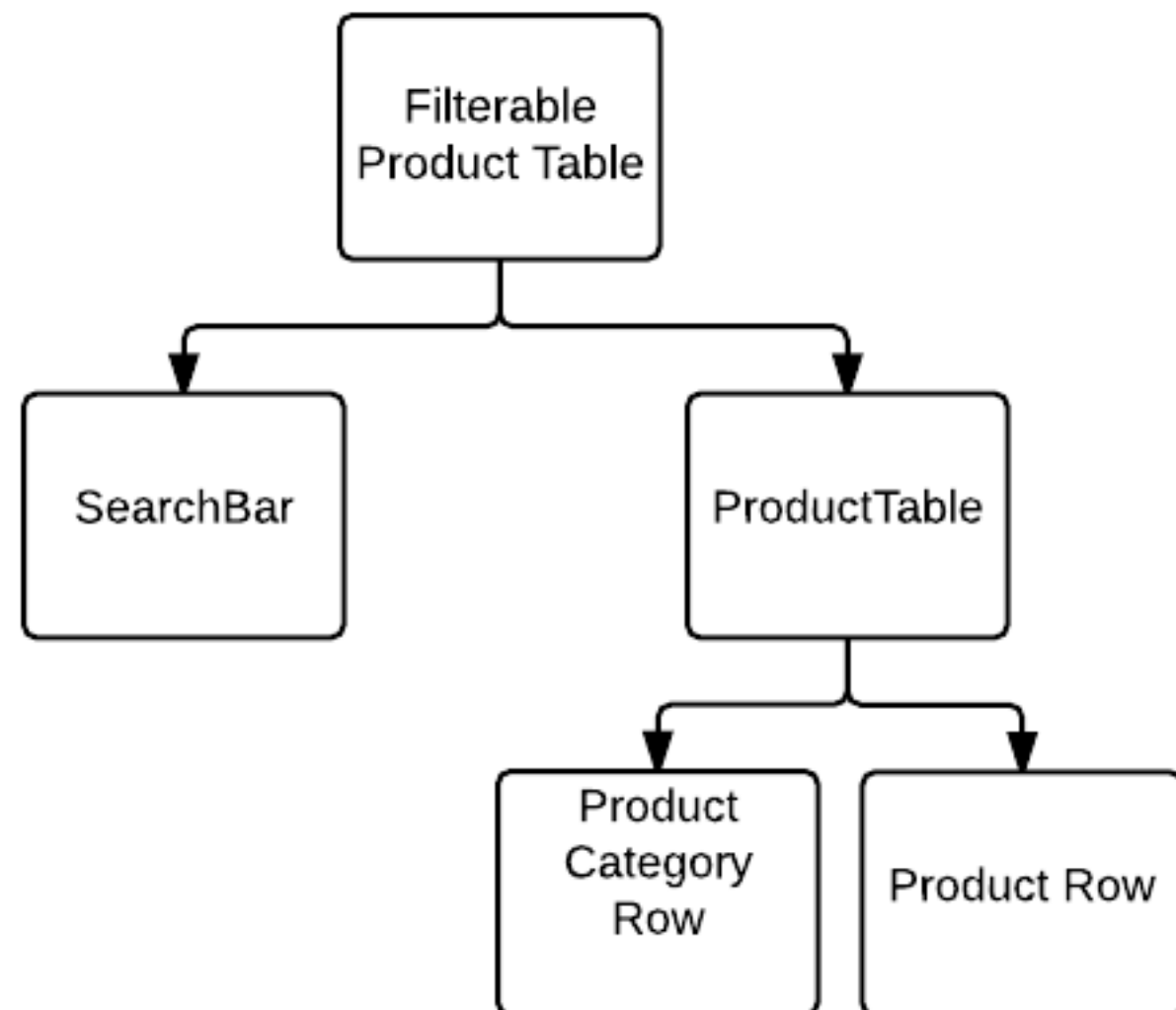
Pourquoi React ?



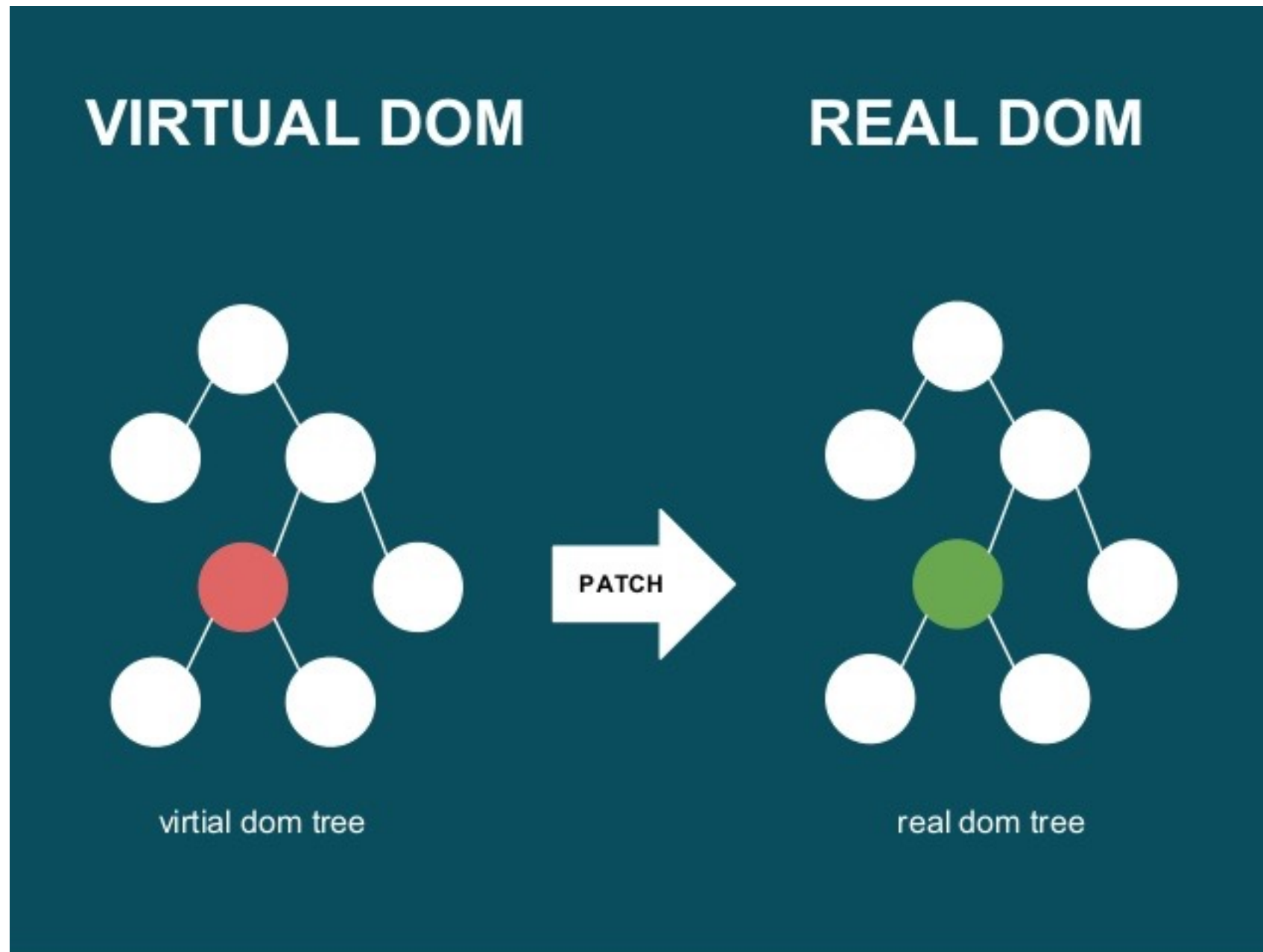
Des composants

☐ Only show products in stock

Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

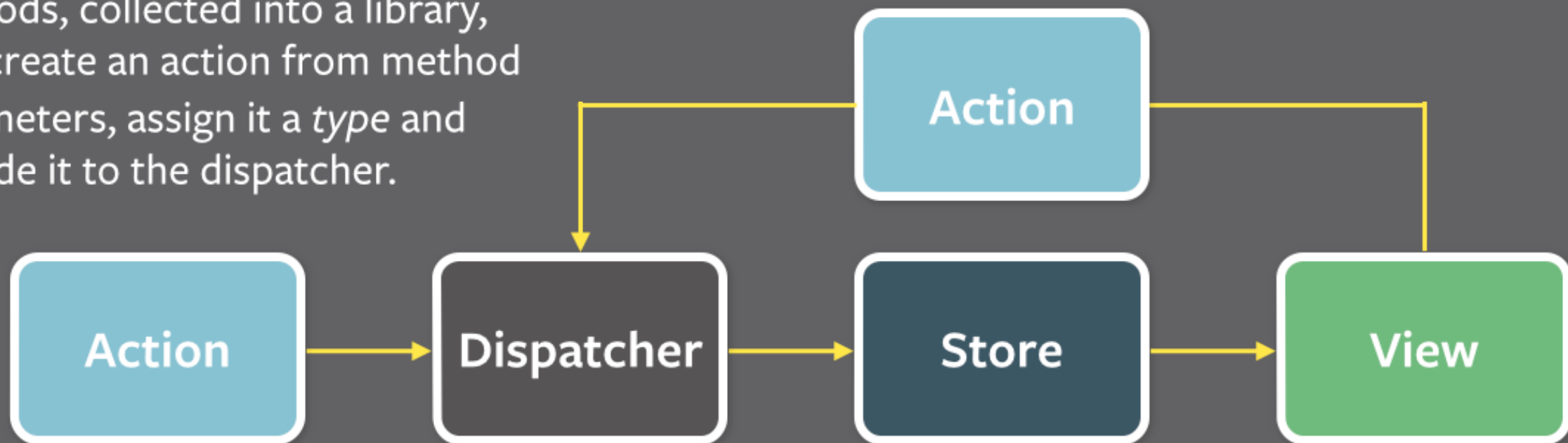


Un DOM Virtuel



Le principe : un flux unidirectionnel

Action creators are helper methods, collected into a library, that create an action from method parameters, assign it a *type* and provide it to the dispatcher.



Every action is sent to all stores via the *callbacks* the stores register with the dispatcher.

After stores update themselves in response to an action, they emit a *change* event.

Special views called *controller-views*, listen for *change* events, retrieve the new data from the stores and provide the new data to the entire tree of their child views.

Deux façons de gérer les données

- ▶ Données qui changent (mutable) :
on utilise un état (state)
- ▶ Données qui ne changent pas (immutable) :
on utilise des propriétés (props)
- ▶ On essaie de minimiser les données qui changent
quitte à refaire des calculs

Un exemple d'application React + Backbone



Instagram

Search

Get the app

Log in



jr



Follow

JR Artist until i find a real job :) Snapchat : JRARTISTE presse.louvre.fr/jr-au-louvre/

3,160 posts

827k followers

774 following

