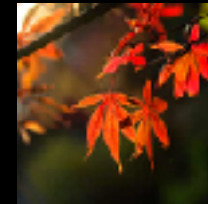


About Me: 题叶(jiyinyiyong)



前端工程师, 函数式语言爱好者

TypeScript, 单页面, 管理系统/报表.

2014 年开始接触 React, 用 CoffeeScript 写 React,
接触到 Webpack, Redux, Immutable

2016 年主要在 Vue, 手机端, 挖掘 ClojureScript

2017 上海徐汇, 积梦智能, 创业公司, React 开发, MobX, Immer.

Virtual DOM 展开的 一些有意思的探索

Virtual DOM 展开的...

(一些 ClojureScript 的探索 (Lisp 方言))

React China

2014-10 跟参加 WiredCraft @fraser 的活动, 搭建论坛, 开微信群

2015 公司参与组织过一些 React 相关的活动

2016 论坛托付给 Strikingly 运行

2017 中文文档上线, 印记中文

2017-11 跟上海的朋友组织了 React Meetup 上海

拆分全国的微信群

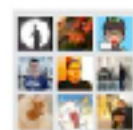
react-china.org



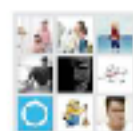
< react



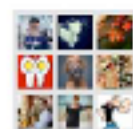
群聊



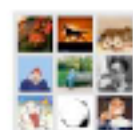
React Meetup 上海



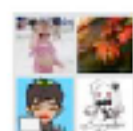
React Native 技术交流群



ReactJS 2群



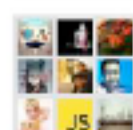
ReactJS 中文



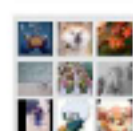
React 中文站线上活动



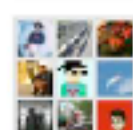
React 广州



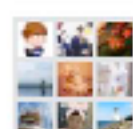
React 北京



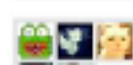
React 成都



React-杭州分会



React 深圳



JS 中国开发者大会(458)

Wishes

希望大家有更多的交流, 线下活动.

函数式编程能更多传播开来.

Goals

压缩开发成本.

Very quick feedback Loop

减少重复劳动.

发挥社区的规模效应...

§ ClojureScript

React patterns

if Expression

Immutability

pure function

...algebra effects?



symbol type mismatch

@Sander_Spies

Following



React - bringing ideas from OCaml to JS since 2013.

4:01 PM - 3 Aug 2018

15 Retweets 92 Likes



15



92



ClojureScript Features

Everything is expression

Persistent data structure

Emphasis on functional programming styles

React bindings: Om, Reagent, Rum...



clojurescript site:https://reactjs.org/blog



All

News

Books

Videos

Images

More

Settings

Tools

About 8 results (0.27 seconds)

Community Round-up #15 - React Blog

<https://reactjs.org/blog/2014/02/05/community-roundup-15.html>

Feb 5, 2014 - It allows you to define efficient React components using nothing but plain **ClojureScript** functions and data, that describe your UI using a ...

You visited this page on 8/7/18.

Community Round-up #12 - React Blog

<https://reactjs.org/blog/2013/12/23/community-roundup-12.html> ▼

Dec 23, 2013 - David Nolen announced Om, a thin wrapper on-top of React in **ClojureScript**. It stands out by only using immutable data structures. This unlocks ...

You visited this page on 8/7/18.

Community Round-up #6 - React Blog

reactjs.org/blog/2013/08/05/community-roundup-6.html ▼

Aug 5, 2013 - I built a somewhat similar framework a year ago WebFUI aimed at **ClojureScript**. My framework also uses global event delegates, a global ...

You visited this page on 8/7/18.

One Year of Open-Source React - React Blog

<https://reactjs.org/blog/2014/05/29/one-year-of-open-source-react.html> ▼



David Nolen

@swannodette

Following



Chrome extensions, React Native integrations, CommonJS processing - the cool stuff coming from the ClojureScript community is stunning :)

6:44 AM - 20 Oct 2015

12 Retweets 39 Likes



3



12



39



3 year exploring ClojureScript

2015 年底开始写 Clojure(Script),

2016 开发 Respo 类库(模仿 deku 开发的 Virtual DOM)

...增加 Cumulo, calcit-editor 代码

2017 帮助 shadow-cljs 推广,

...用来开发各种工具类的页面

It's a Lisp 🤯

```
[...]
```

```
(->> tasks  
  (sort by-latest-task)  
  (map  
    (fn [task]  
      [(:id task)  
       (div  
         {}  
         (comp-time (:done-time task))  
         (= 8 nil)  
         (<> (:text task))))]))]))]))]))]))]))]))]))))
```

pp/main.cljs



calcit-editor (括号编辑器)

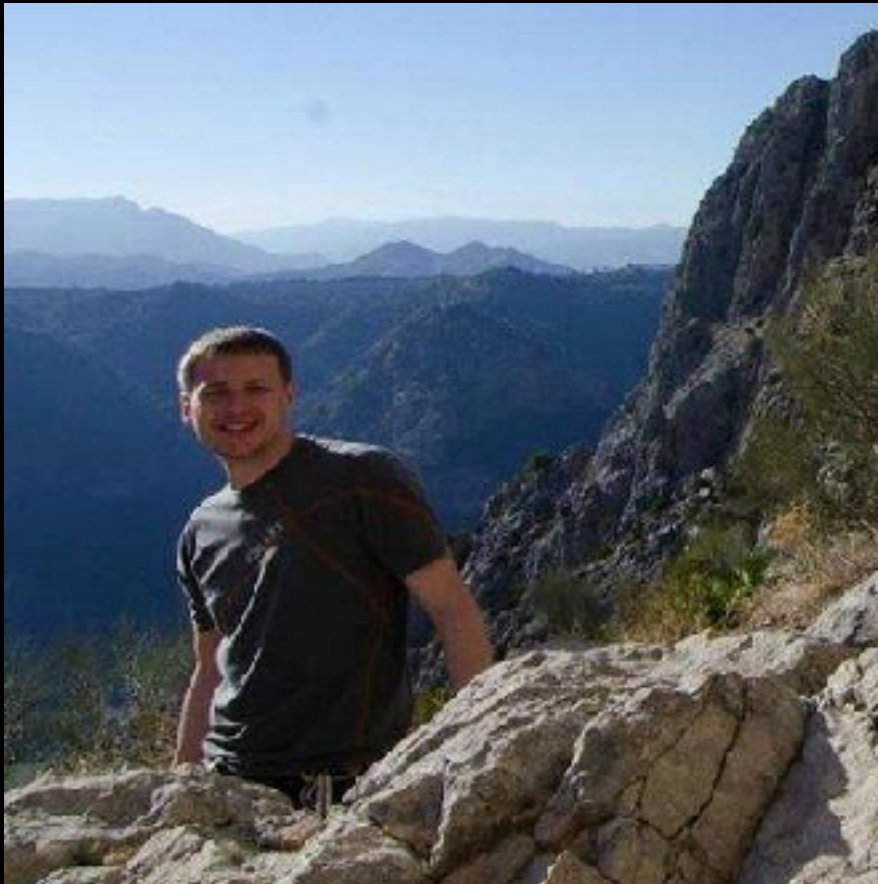
Clojure: a totally different world

Clojure, Lein,
Google Closure Compiler,
externs/shim,
npm-deps,
classpath, JVM,
Lisp, Macro,
Maven, Clojars,
...

```
(defn dodgy-map []  
  {:1 :2 :3})  
  
$ lein ring server-headless  
Exception in thread "main" java.lang.RuntimeException: Map literal must contain an e  
ber of forms, compiling:(one_man_wiki/views.clj:19)  
    at clojure.lang.Compiler.load(Compiler.java:6958)  
    at clojure.lang.RT.loadResourceScript(RT.java:359)  
    at clojure.lang.RT.loadResourceScript(RT.java:350)  
    at clojure.lang.RT.load(RT.java:429)  
    at clojure.lang.RT.load(RT.java:400)  
    at clojure.core$load$fn4890.invoke(core.clj:5415)  
    at clojure.core$load.doInvoke(core.clj:5414)  
    at clojure.lang.RestFn.invoke(RestFn.java:408)  
    at clojure.core$load_one.invoke(core.clj:5227)  
    at clojure.core$load_lib.doInvoke(core.clj:5264)  
    at clojure.lang.RestFn.applyTo(RestFn.java:142)  
    at clojure.core$apply.invoke(core.clj:603)  
    at clojure.core$load_libs.doInvoke(core.clj:5298)  
    at clojure.lang.RestFn.applyTo(RestFn.java:137)  
    at clojure.core$apply.invoke(core.clj:603)  
    at clojure.core$require.doInvoke(core.clj:5381)  
    at clojure.lang.RestFn.invoke(RestFn.java:457)  
    at one_man_wiki.handler$eval1564$loading4784$auto1565.invoke(handler.clj:1)  
    at one_man_wiki.handler$eval1564.invoke(handler.clj:1)  
    at clojure.lang.Compiler.eval(Compiler.java:6511)  
    at clojure.lang.Compiler.eval(Compiler.java:6501)  
    at clojure.lang.Compiler.load(Compiler.java:6952)  
    at clojure.lang.RT.loadResourceScript(RT.java:359)  
    at clojure.lang.RT.loadResourceScript(RT.java:350)  
    at clojure.lang.RT.load(RT.java:429)  
    at clojure.lang.RT.load(RT.java:400)  
    at clojure.core$load$fn4890.invoke(core.clj:5415)  
    at clojure.core$load.doInvoke(core.clj:5414)  
    at clojure.lang.RestFn.invoke(RestFn.java:408)  
    at clojure.core$load_one.invoke(core.clj:5227)  
    at clojure.core$load_lib.doInvoke(core.clj:5264)
```

thheller/shadow-cljs

ClojureScript compilation made easy!



Real benefits

Everything is an expression,

Very natural immutable data and built-in functions,

Functional programming and macros,

Friendly English community.

Respo: Virtual DOM Library

模仿 Deku 和 virtual-dom 设计,
基于 ClojureScript, 整个使用不可变数据.
只实现了渲染, 没有生命周期, 副作用很少,
自己控制渲染的细节和代码风格, 做试验.

<http://respo.site>

```
(div {:class-name "demo"  
      :style {:font-size 16  
              :font-family "Josefin Sans"}  
      :on-click (fn [event dispatch! mutate!]  
                  (dispatch! :action data)))})
```

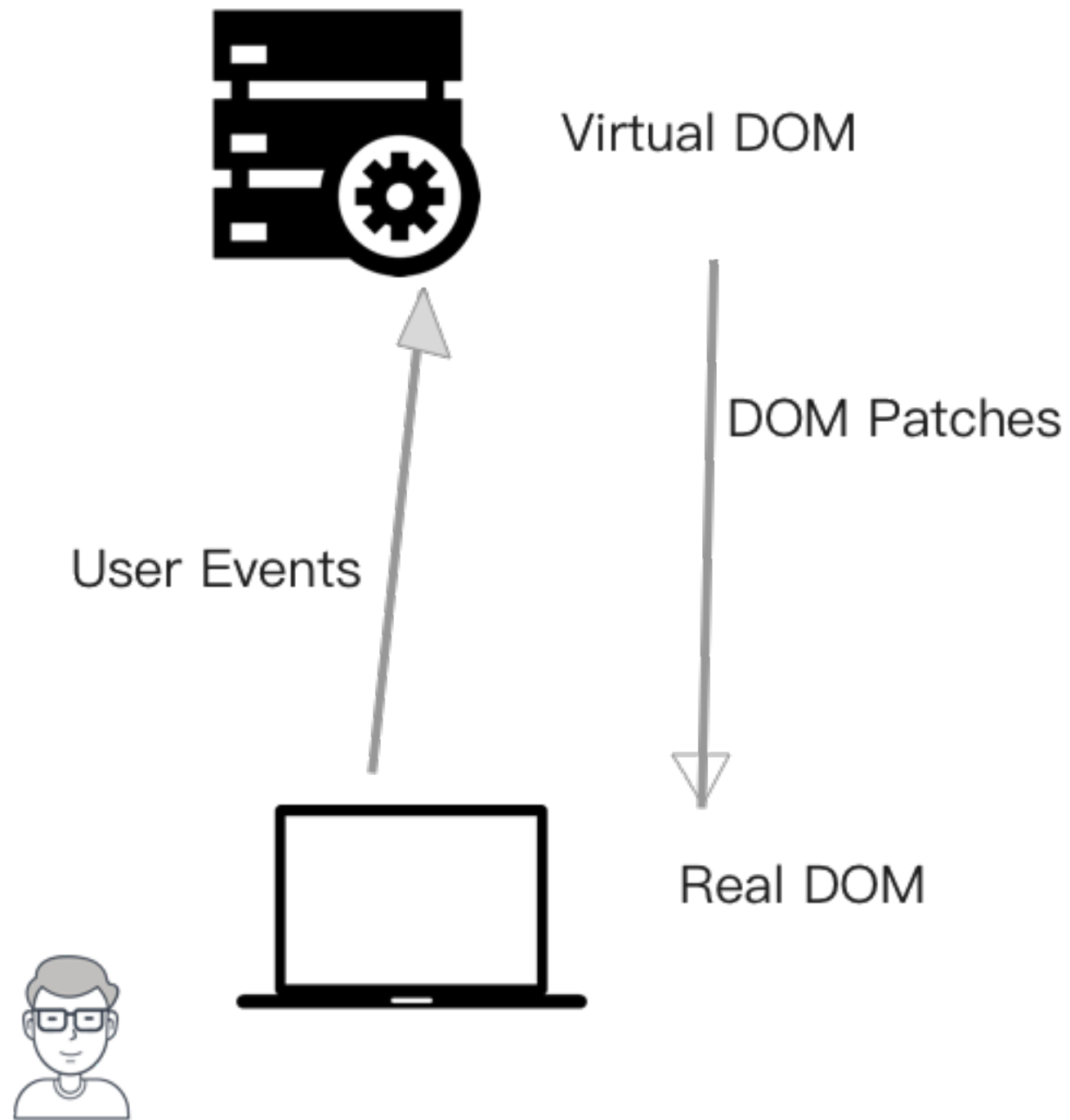
Other ideas

在服务端进行 DOM Diff


```

[[0
  [0 2 0]
  {:name :div,
   :coord [:comp-container 0 :comp-todolist 2 50 :comp-task],
   :attrs (),
   :style ([:display :flex] [:padding "4px 0px"]),
   :event #{} ,
   :children ([0
     {:name :pre,
      :coord [:comp-container 0 :comp-todolist 2 50 :comp-task 0 :comp-inspect],
      :attrs ([:inner-text "Task: Map/3"]),
      :style ([:background-color "hsl(240,100%,0%)"]
               [:border-radius "4px"]
               [:color :white]
               [:padding "2px 6px"]
               [:position :absolute]
               [:white-space :normal]),
      :event #{:click},
      :children (),
      :svg? false}]
    [1
     {:name :button,
      :coord [:comp-container 0 :comp-todolist 2 50 :comp-task 1],
      :attrs (),
      :style ([:background-color "hsl(200,80%,70%)"]
               [:border-radius "4px"]
               [:padding "2px 6px"]
               [:position :absolute]
               [:white-space :normal])}]]])

```

Drawbacks/Benefits

反应缓慢, 体验不好. 服务端性能开销非常大.
客户端没有逻辑, 更加安全. 瘦客户端.

§ Respo States Tree

Local states lost after HMR

可以通过 react-hot-loader 解决

热替换前强行复制存储状态, 热替换后恢复,

Component states in a tree

ClojureScript 社区有全局状态的传统.

把状态存储成一棵树, 存放在全局, 这样组件状态在热替换中不丢弃.

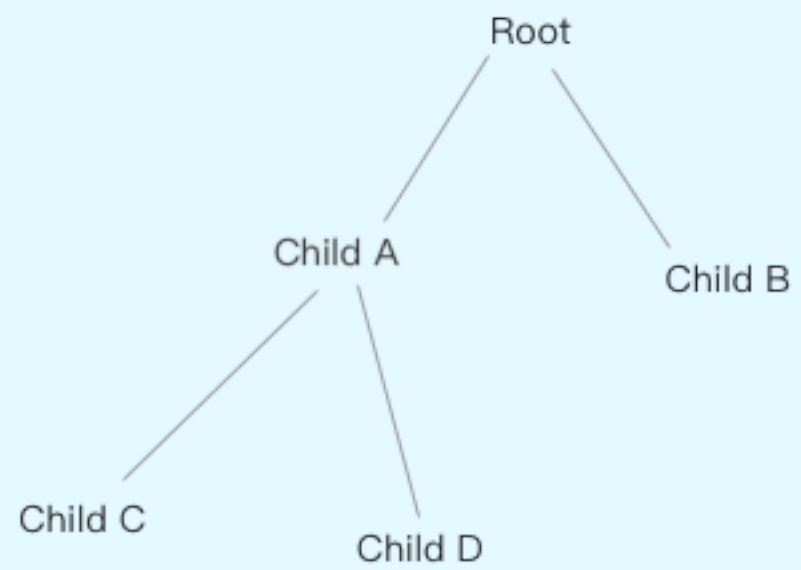
需要在代码当中做一些处理, 借助 macro 来进行达成.



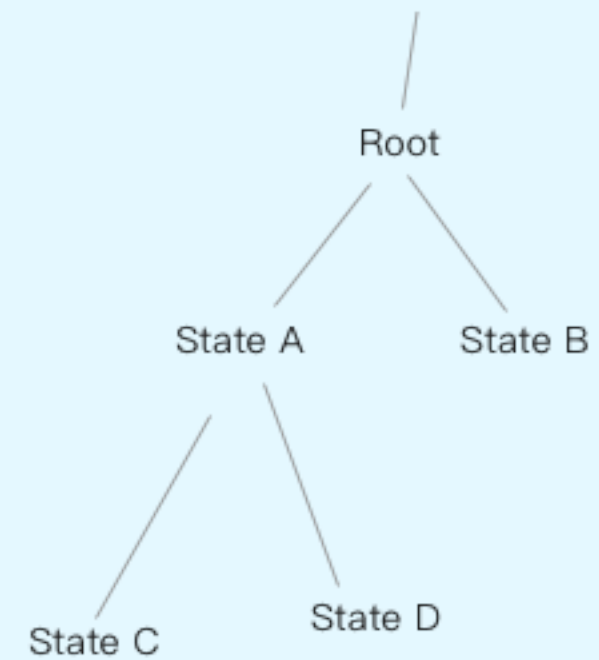
Store

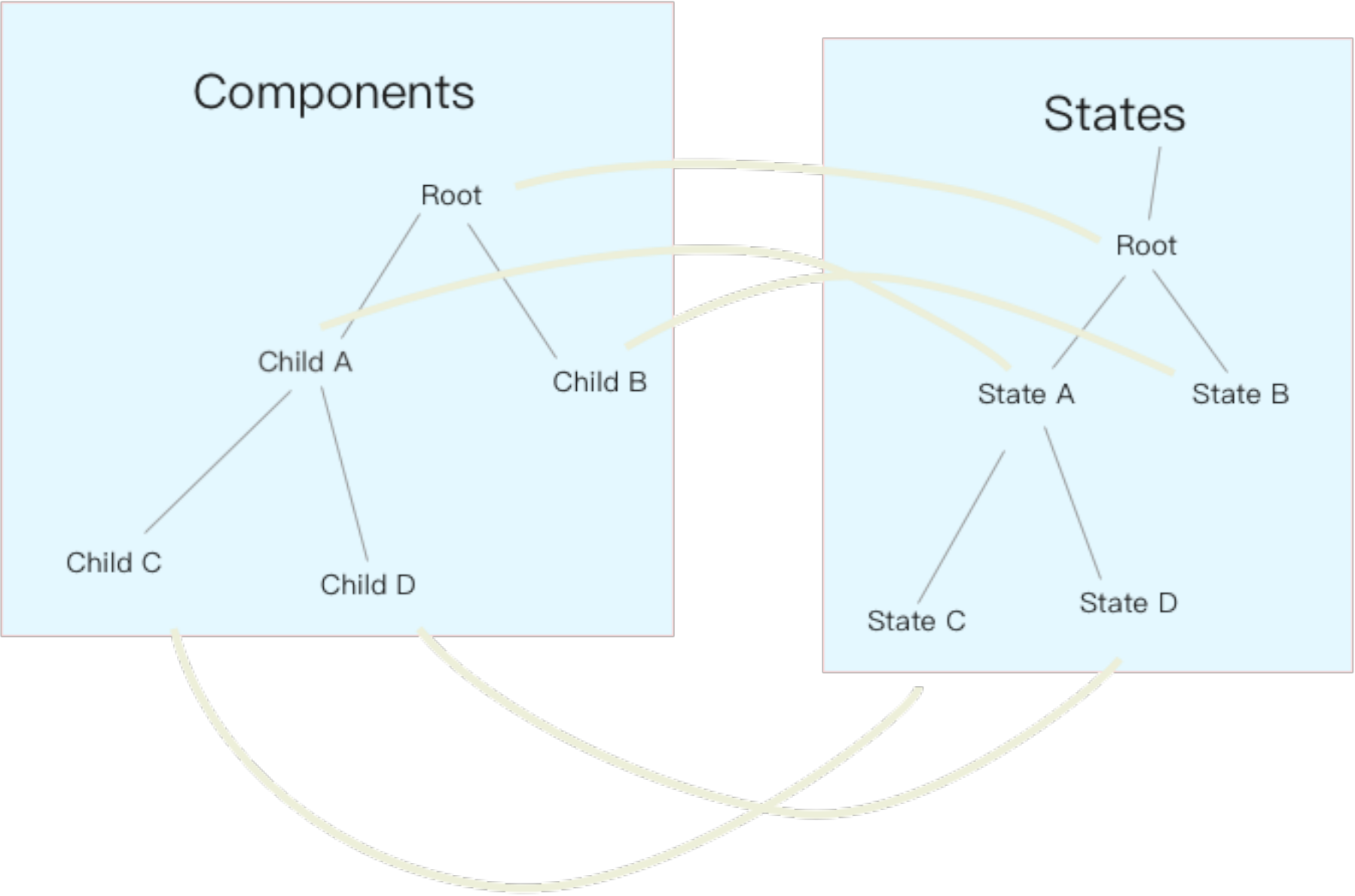
States tree

Components



States






```
(defonce store (atom {:states {}}))
```

```
(cursor-> :sub-cursor comp-demo states p1 p2)  
(assoc (comp-demo (:sub-cursor states) p1 p2)  
       :cursor (conj %cursor :sub-cursor))
```

```
(def initial-state {:count 1})

(defn on-click [%cursor state]
  (fn [e dispatch! mutate!]
    ; (mutate! (update state :count inc))
    (dispatch! :states [%cursor (update state :count inc)])))

(def comp-demo
  (create-comp
    (fn [states]
      (fn [%cursor]
        (let [state (or (:data states) initial-state)]
          (create-element :div {}
            (div {:inner-text (str (:count 1)), :style nil})
            (comp-space 8 0)
            (button {:inner-text "inc"
                      :on-click (on-click %cursor state)}))))))))
```




9



16



205

Working(9) +

```
{:text nil, :show? false}
```

standup meeting notes



索引晓风书屋



https://purelyfunctional.tv/guide/clojure-collections/?utm_source=dlvr.it&utm_



Wood 折叠已完成的信息



Woodenlist 居中布局



重新制作 tiye.me



Drawbacks

状态和组件独立, 不会随着组件重建而重置, 需要手动维护.

需要额外的代码标记状态树. 引入了 Macros.

Benefits

全局的状态, 跟热替换配合得很好.

状态树可以打印出来, 方便调试.

一些可以做某些跨组件的状态操作.

§ Realtime Apps(Cumulo)

Cumulo

在服务端渲染客户端需要的数据

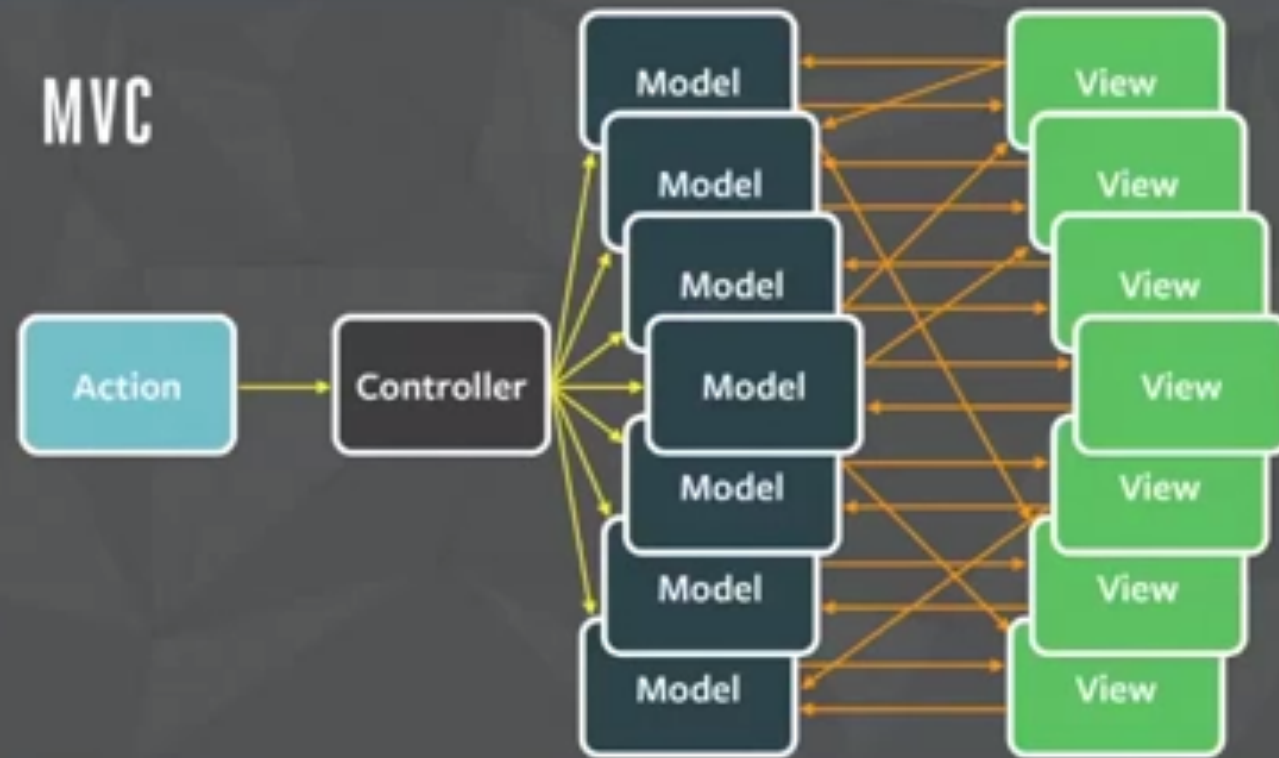
在服务端做数据的 Diff, 针对每个连接

客户端接收到 Patch 更新本地数据

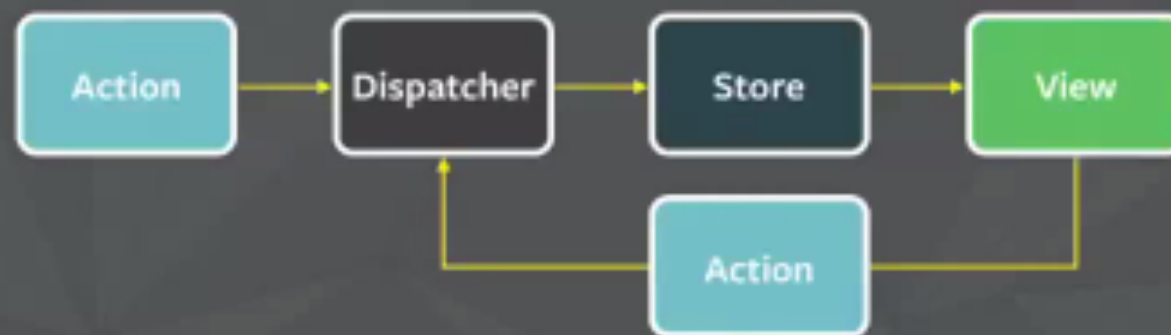
5

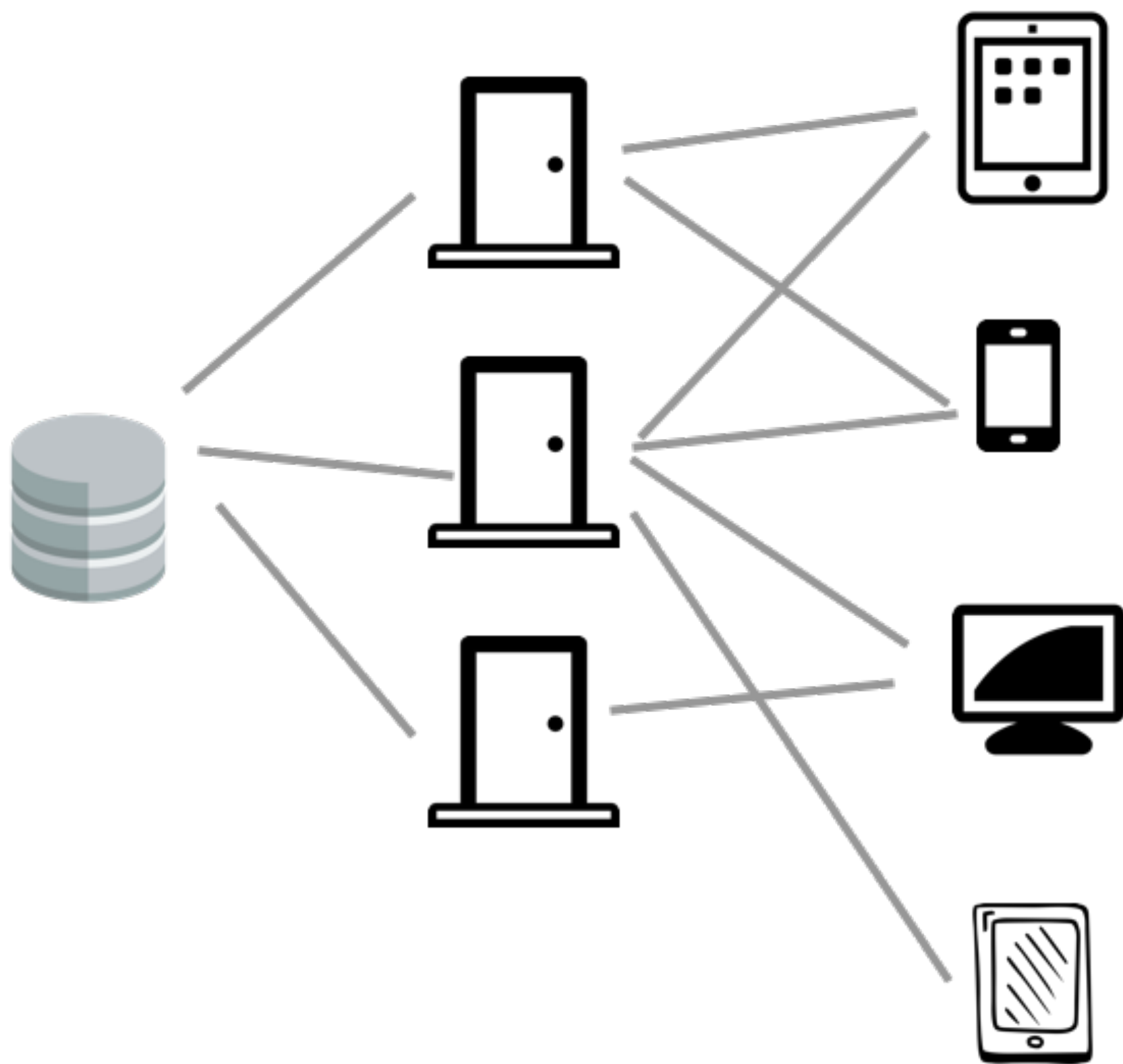


MVC

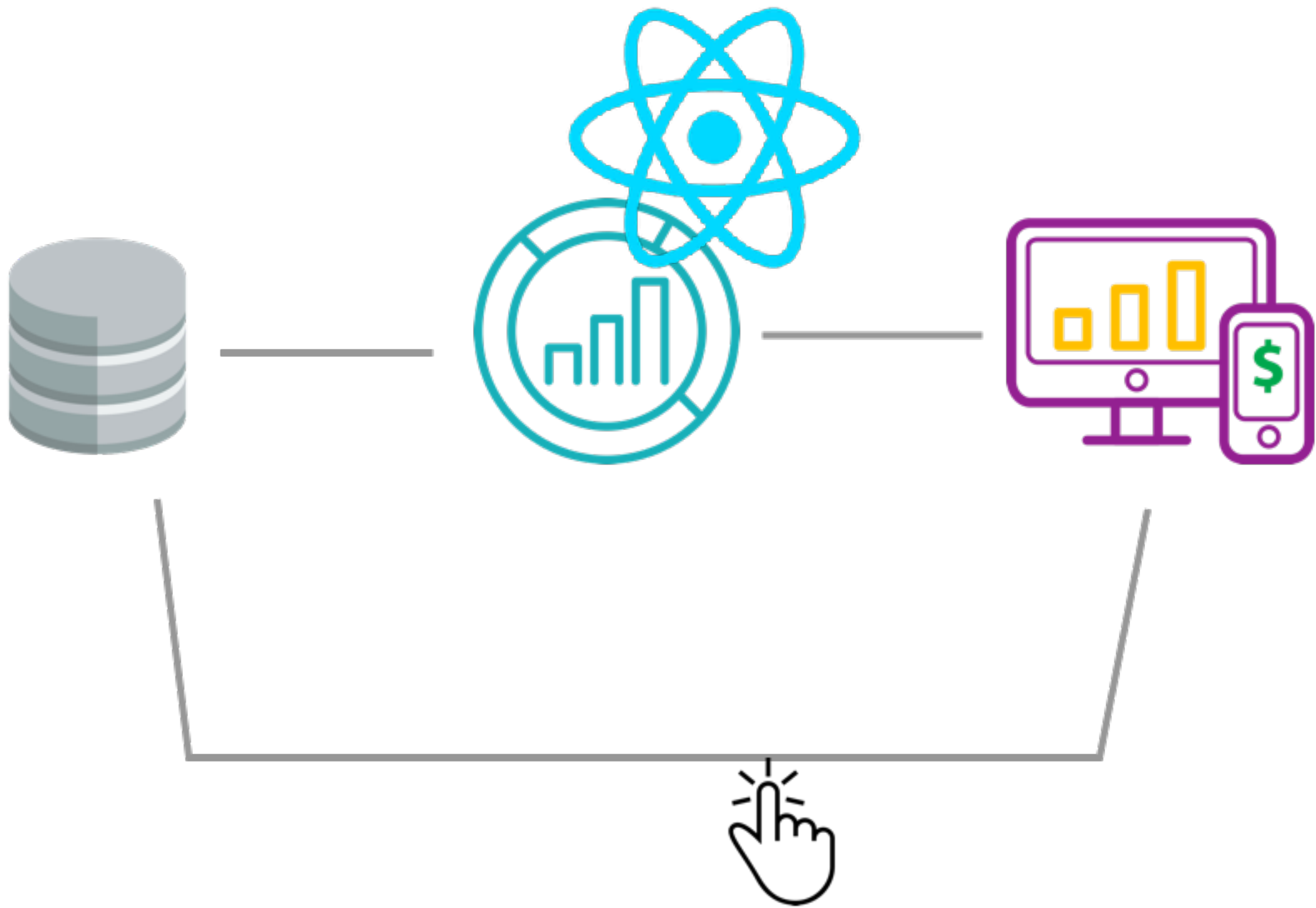


FLUX









Cumulo Workflow

Wood.topix.im

repo.topix.im/copyboard

table-two.topix.im

Drawbacks

性能很差, 不方便多机器扩展.

数据在内存里存放, 体积控制不好, 也不方便持久化.

Benefits

开发方便, 不用在服务端和客户端有很多重复工作.

能在后端开发完成热替换, 简化工作量.

类似于单页面的开发方式, 可以快速作出原型.

Don't be too slow

对数据封装, 做类似 `shouldComponentUpdate` 的缓存优化.

通过不可变数据减少冗余的计算.

但是这用同时限制了多机器的扩展...

What I got

基于 ClojureScript 的开发环境,
结合 Respo 和 Cumulo
支持前端热替换, 后端热替换

DEMO of Cumulo

Other ideas

在 Virtual DOM Diff 的基础上生成补间动画.

在线编辑器开发网页.

any-questions?))))))