FOR THE REVIEW ONLY, **MAY CONTAIN MISTAKES!** (dispatch [:event-key1]) coeffects **VIEW** 6-domino cascade reg-cofx :cofx-key1 D1 Event Dispatch DOM Local Storage view2 element view1 element An event is sent when something happens the user clicks a button, or a websocket receives a new message. D2 Event Handling React In response to an event, an application must VDOM router decide what action to take. This is known as finite-state-machine event handling. render activate D6 changed subscriptions :event-keyn D3 Effect Handling views later in time after :event-key1 :event-key1 Reagent (views) The descriptions of effects are realised (actioned). defn view1 defn view2 defn view3 D4 Query on goog.async.nextTick (subscribe (subscribe (subscribe [:b]) [:sub-key3]) [:sub-key2]) your pure function your pure function re-frame :db cofx your pure function Domino 4 is about extracting data from "app state", and providing it in the right format for events view functions (which are Domino 5). reaction on queries result reg-event-fx reg-event-fx **QUERY** changes :event-key1 :event-keyn Domino 5 is one or more view functions (aka F8 Reagent components) that compute the UI subscriptions F2 DOM that should be displayed to the user. :coeffects :after

D6 DOM

You don't write Domino 6 - it is handled for you by Reagent/React.

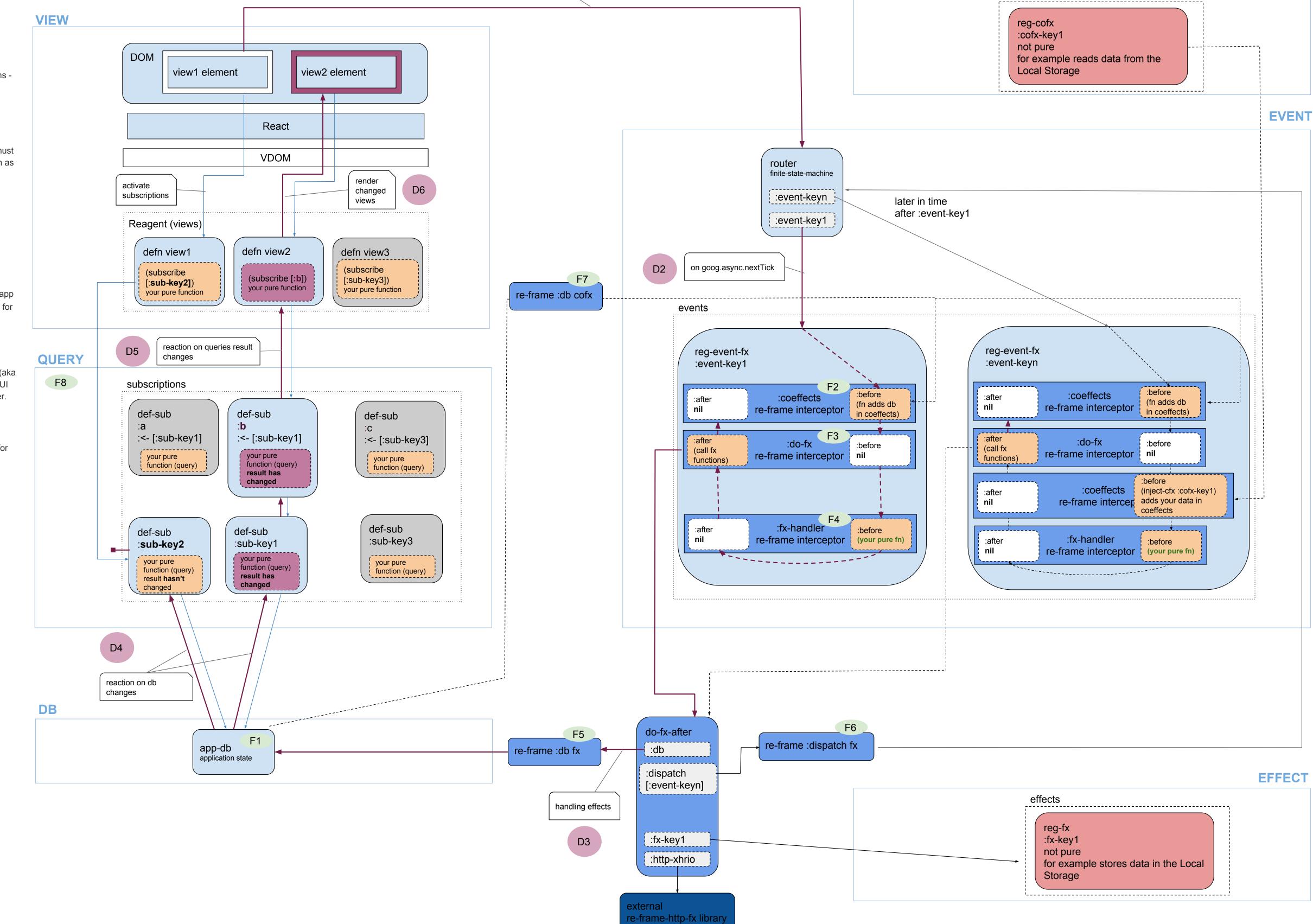
view3 not in the DOM, so it's not active, and subscribtion :sub-key3 not active too

Subscriptions :a and :c are not subscribed anywhere, so they are not active too.

Result of calculation :sub-key2 hasn't changed, so view1 doesn't recalculate and view1 element doesn't re-render.

domino

re-frame internal



RE-FRISK

```
https://github.com/flexsurfer/re-frisk
 ▼ {
:handlers ▼ {
               :initialize-db ▶ {"3 interceptors"}
                               :clock?-db ▼ {
                                         "3 interceptors" ▼ (
                                                          ▶ {:coeffects :before} F2
                                                          ▶ {:fx-handler :before} F4
                              :timer-db > {"4 interceptors"}
                              :time-color > {"4 interceptors"}
                        ▶ {:dispatch-n} F6
                        ▶ {:deregister-event-handler}
                        {:dispatch}
                        ▶ {:dispatch-later}
                  :cofx ▼ (
                         ▶ {:db} F7
F1 :app-db ▼ {
                 :timer Mon Dec 26 2016 15:36:07 GMT+0300 (Russia TZ 2 Standard Time)
                 :clock? true
```

COEFFECT