

Building a Functional Frontend with Clojurescript & Re-frame



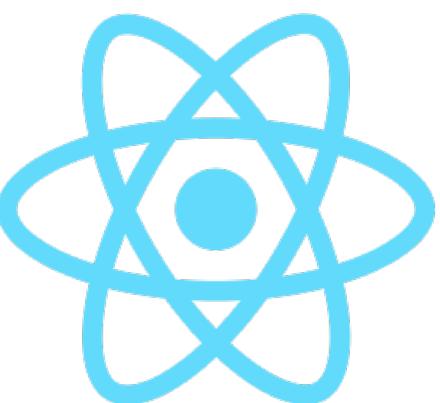
.:: Boyan Bonev ::.

js.talks();

WHERE IS ~~WALDO~~ RADO?



REASON

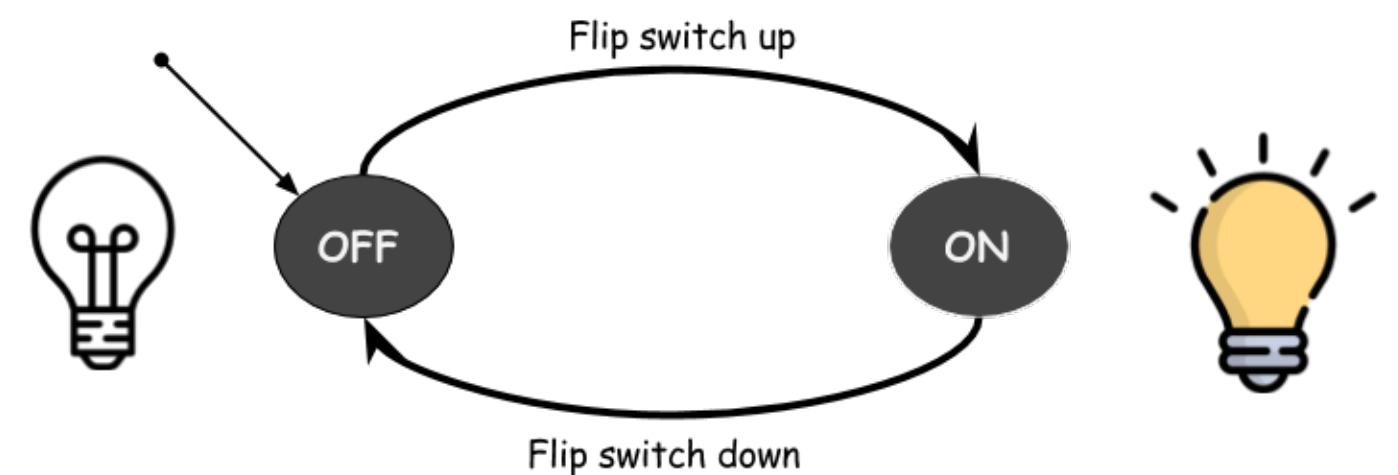


node
JS



cljs

Y λ



JS

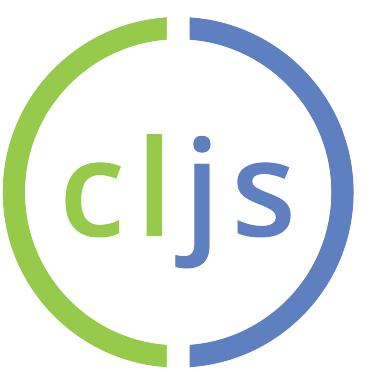
reframe

The Clojure logo, which consists of three overlapping circles in blue, green, and white.

V

TS





THE ROAD SO FAR

1

!

Act I
The Road So Far

2

?

Interlude
A Tale of Clojurescript

3

?

Act II
Building Functional

A BRIEF HISTORY OF ME JS

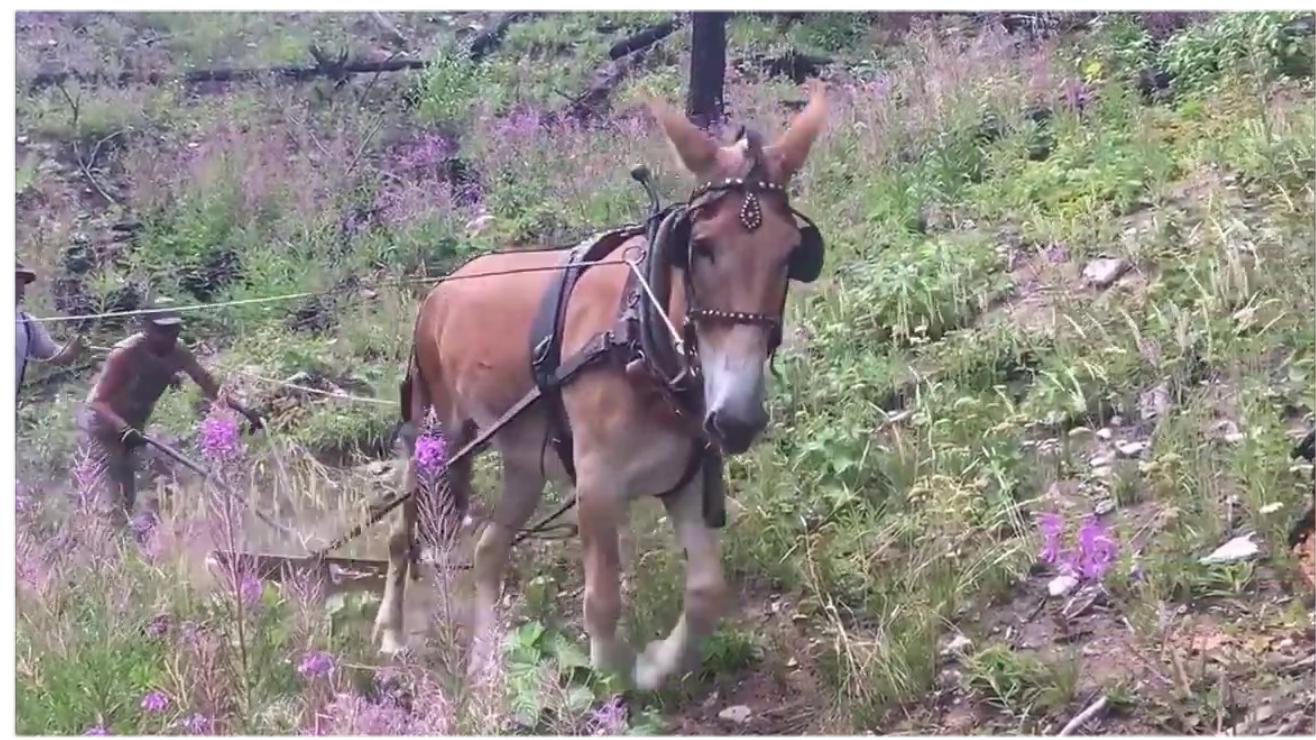


The jQuery Ages

Everybody is pulling their fair share, but at the end of the day it still feels as if you are a mule.

The Enlightenment

Rays of light appear here and there.



Agricultural Revolution

We're starting to do better, but we can't even guess what's ahead!



**2018/11/17,
after lunch**

Everybody's driving a Ferrari!

**“Anything invented after you're
35 is against the natural order
of things”**

ADOPTING TECH AT DIFFERENT AGES

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.

SOON:

SITUATION:

THERE ARE

COMPETING

STANDARDS.

SITUATION:

THERE ARE

COMPETING

STANDARDS.

— Douglas Adams

JS

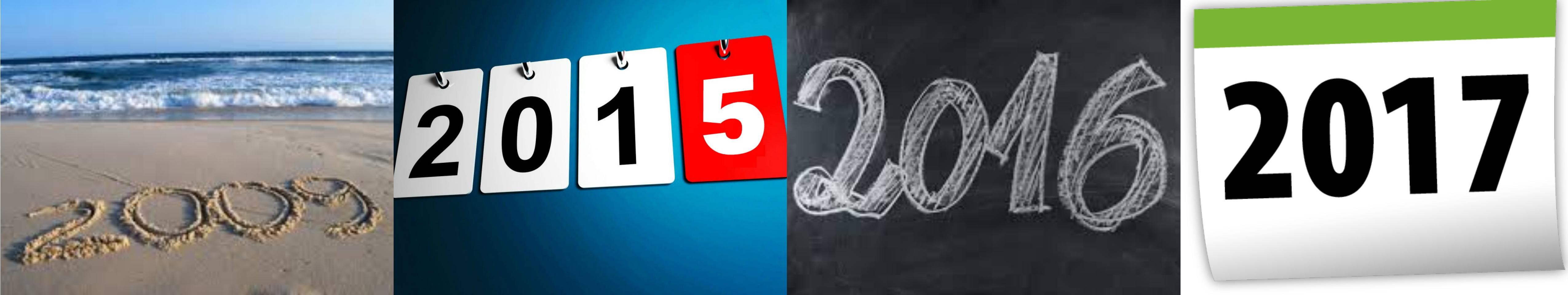


TS



Dart

THE EVOLUTION OF NaN



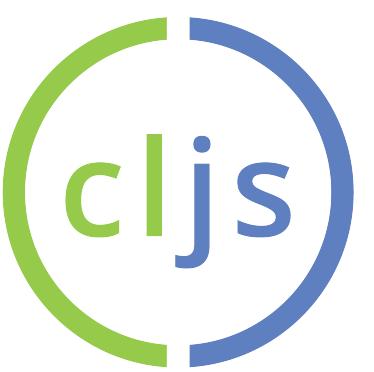
A TIMELINE OF TWO LANGUAGES

2007

2011
will be
awesome.



One Year
Later



A TALE OF CLOJURESCRIPT

1

=

Act I
The Road So Far

2

!

Interlude
A Tale of Clojurescript

3

?

Act II
Building Functional



CLOJURE

- **A LISP**
- **Dynamic language**
- **Immutable data structures**
- **Develop around data**
- **Very small core**
- **Emphasis on concurrency**
- **Designed to be a hosted language**

```
(ns hello-world.core)
```

```
(defn -main
  []
  (println "Hello, World!"))
```

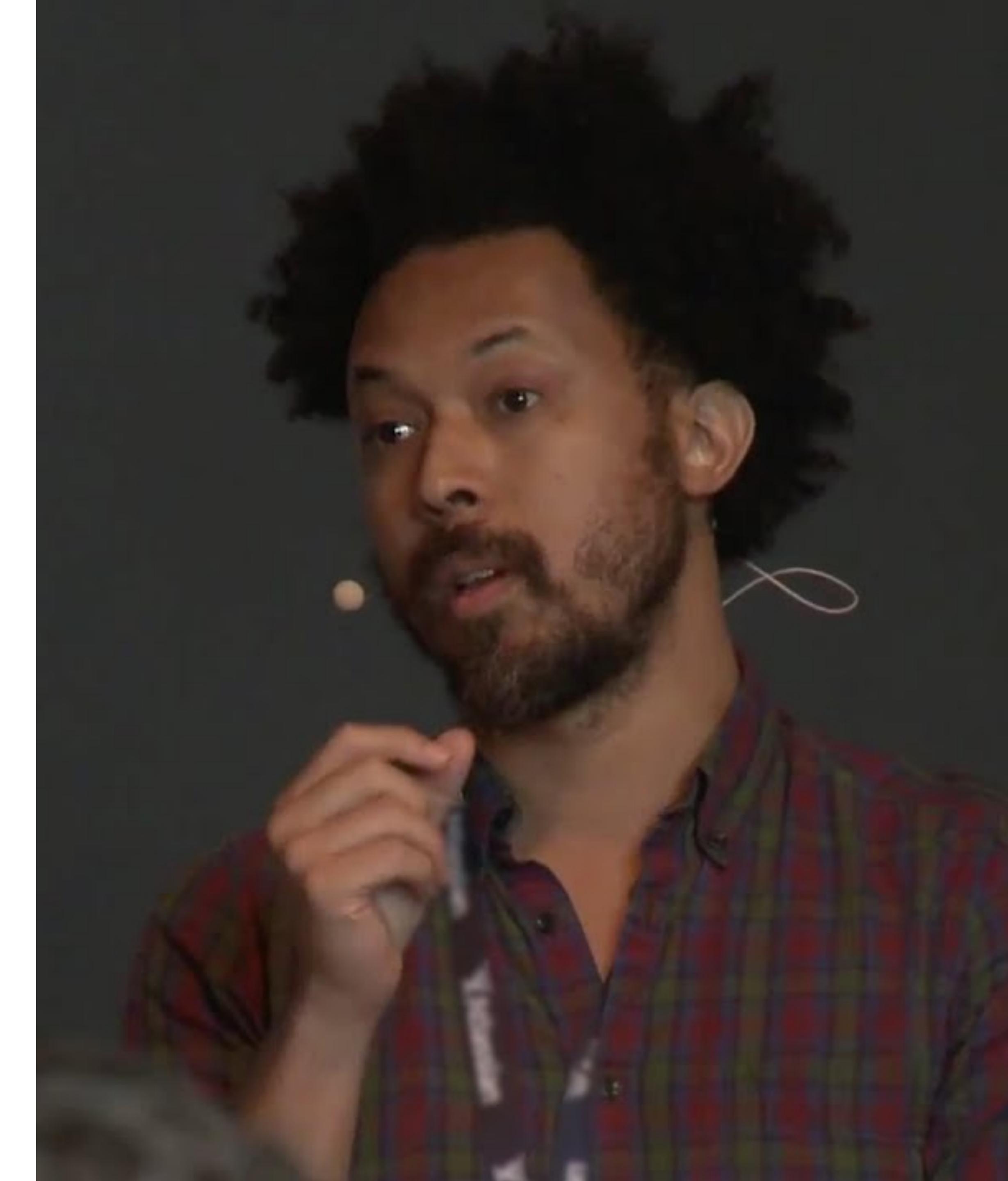


CLOJURESCRIPT

- It's Clojure
- Immutable
- Amazing interop story
- Mature and modern tooling
- Active community
- Cljs everywhere

```
(ns hello-world.core)
```

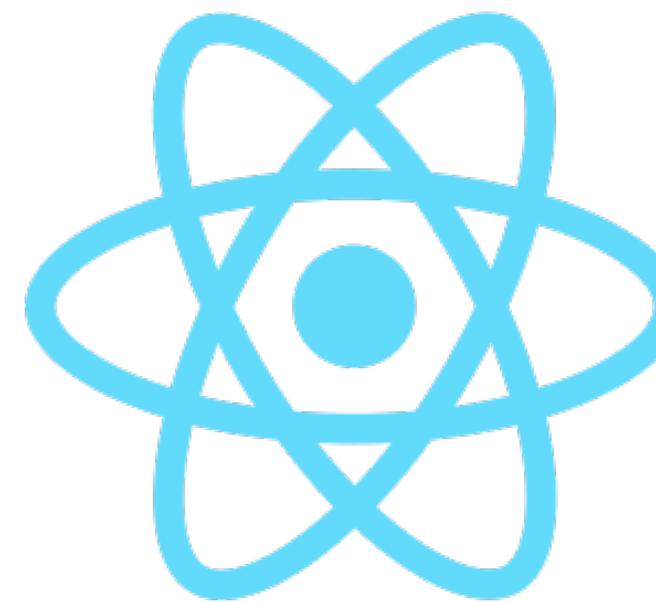
```
(println "Hello world!")
```



CLJS + REACT = <3 <3 <3



Reagent



```
(defn hello []
  [:div "Hello, world!"])

(defn ^:export main []
  (reagent/render [hello]
    (.getElementById js/document "app")))
```

```
ReactDOM.render(
  <div>Hello, world!</div>,
  document.getElementById('app')
);
```

IT ONLY GETS BETTER

- The Tooling actually intimidating
- Figwheel is civilization, but Java is scary
- Shadow-cljs to the rescue
- Error messages are awesome



cljs

Compile Warning src/flappy_bird_demo/core.cljs

Use of undeclared Var flappy-bird-demo.core/This

```
50
51 (defn curr-pillar-pos [cur-time {:keys [pos-x start-time] }]
52   (translate pos-x horiz-vel (- cur-time start-time)))
53
54
55 This is cool
^--- Use of undeclared Var flappy-bird-demo.core/This
56
57 (defn in-pillar? [{:keys [cur-x]}]
58   (and (>= (+ flappy-x flappy-width)
59             cur-x)
60         (< flappy-x (+ cur-x pillar-width))))
```

Please see line 55 of file src/flappy_bird_demo/core.cljs

Use of undeclared Var flappy-bird-demo.core/is at line 55, column 6 in file src/flappy_bird_demo/core.cljs

Use of undeclared Var flappy-bird-demo.core/cool at line 55, column 9 in file src/flappy_bird_demo/core.cljs

Elements Console Sources Network Performance Memory Application Security Audits

1 3

top Filter All levels Group similar

[Figwheel REPL] Session ID: a98183c4-362f-4c1c-b5ee-0336d9a6877c

console.js:203

console.js:203

Summary: CREATED 0 RUN 6 DISPOSED 0 NOT-RUN 0

Ignore 1 unchanged layer 2 sub

ACTIVITY	LAYER	DIFFS
▶ [:todos]	R	(3)
▼ [:sorted-todos]	R	(2)
▼ [4 ▶ {:id 4, :title "abc", :done true}][6 ▶ {:id 6, :title "abc", :done false}][7 ▶ {:id 7, :title "add", :done true}][8 ▶ {:id 8, :title "abc", :done true}][9 ▶ {:id 9, :title "abc", :done false}][10 ▶ {:id 10, :title "def", :done false}]	(6)	
▶ [:visible-todos]	R	(3)

- Debugging is a breeze
- Shell repl, browser console repl
- Advanced state debuggers
- Developing at scale works

BUILDING FUNCTIONAL

1

=

Act I
The Road So Far

2

=

Interlude
A Tale of Clojurescript

3

?

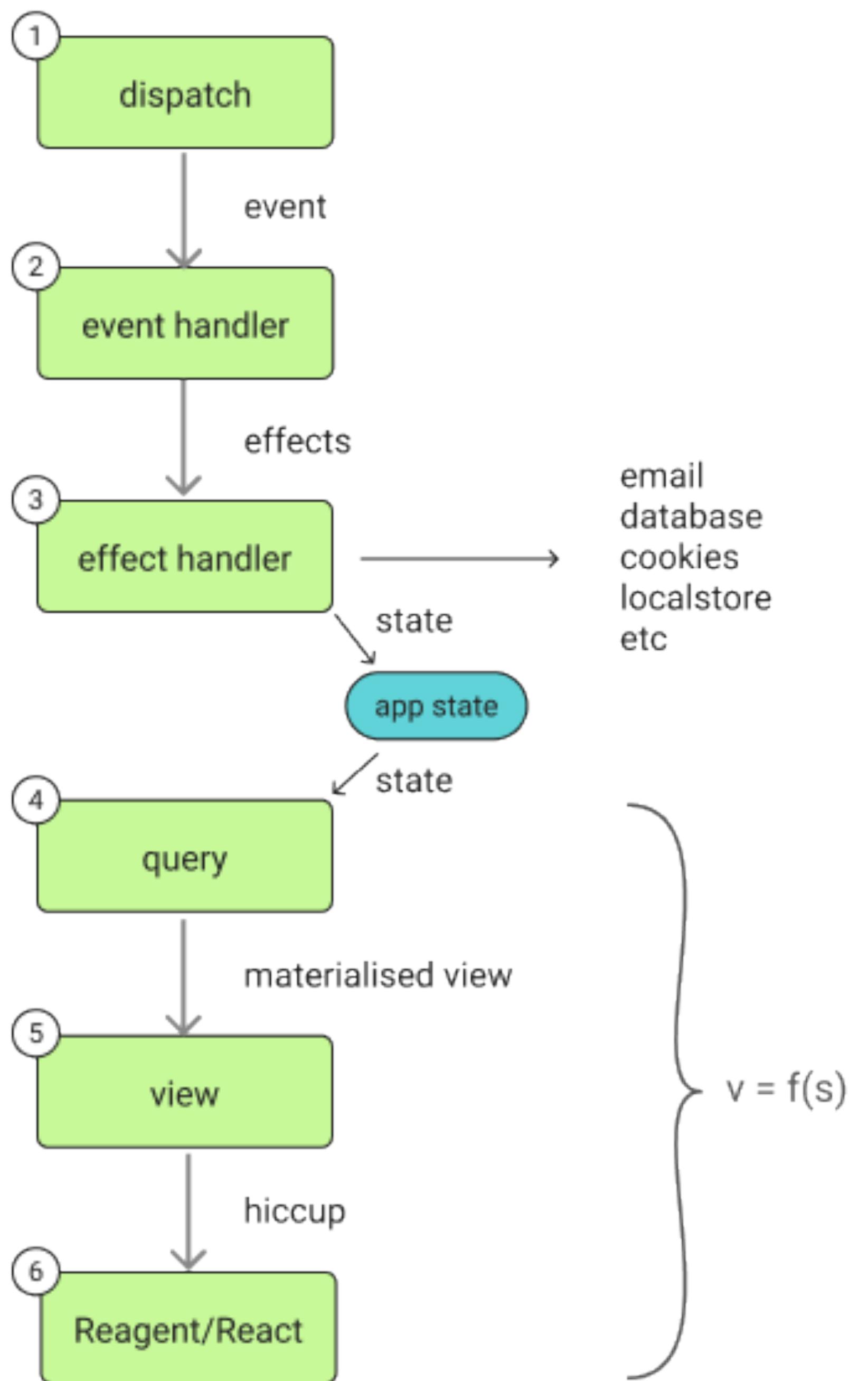
Act II
Building Functional



- **Frontend matters**
- **Frontend is in flux**
- **Simple, Complicated, Complex, Chaotic**
- **Tests are good**
- **Frontend is often hard to do right**

WE GET RELIGIOUS HERE

ENTER RE-FRAME



- Building complex SPA's on the frontend
- Loop of ideas - Elm/Om/Redux
- Practical, instead of preachy
- Amazingly documented
- Used in SPAs of size

DDATWD

- Model around events
- Oh, btw, events are data hth
- State as a perpetually derived value
- Designing Data Intensive Applications

```
(def <sub (comp deref re-frame.core/subscribe))
(def >evt re-frame.core/dispatch)

(defn toggle-all-checkbox []
  [:span
   [:input {:type "checkbox"
            :checked (<sub [:todos/all-complete?])
            :on-change #(>evt [:todos/toggle-all])}]
   [:label {:for "toggle-all"} "Mark all as complete"]])
```

```
(reg-event-db
  :delete-todo
  todo-interceptors
  (fn [todos [_ id]]
    (dissoc todos id)))

(reg-event-db
  :clear-completed
  todo-interceptors
  (fn [todos _]
    (let [done-ids (->> (vals todos)
                           (filter :done)
                           (map :id))]
      (reduce dissoc todos done-ids)))) ;; delete todos which are done
```

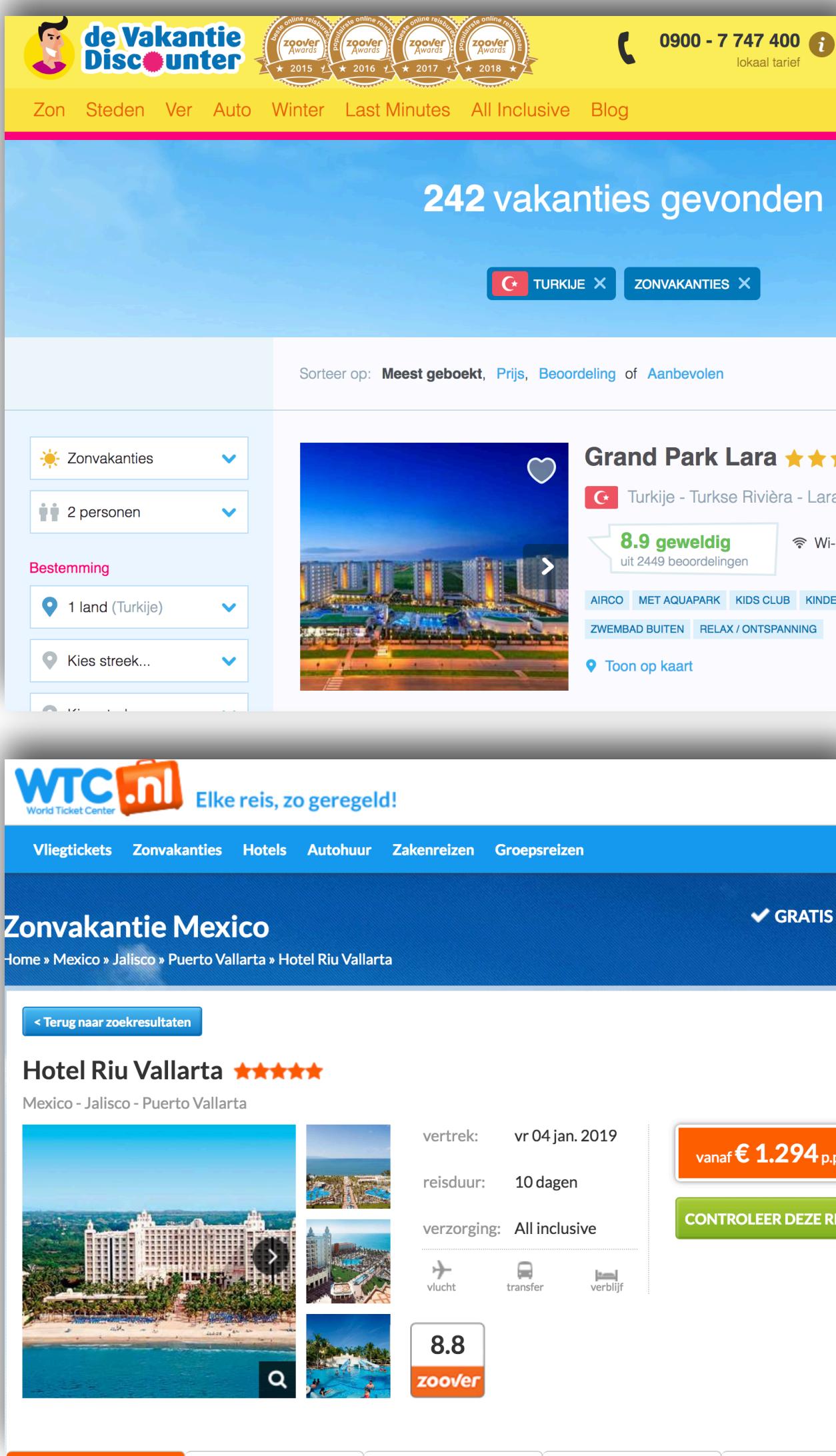
TRY TO DO A DEMO

- Click here
- Do demo
- Click here if fail

Improve your

The shortest distance from idea to exec

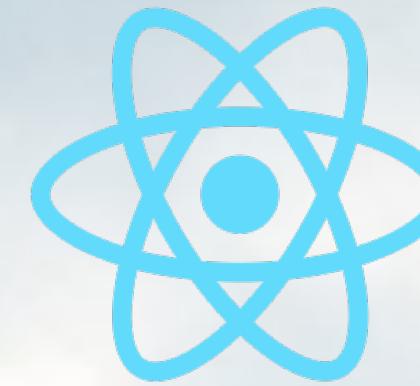
Automate your development process quickly
and at any scale.

[Start Building for Free](#)[Explore](#)

Experience report



REASON



The Future



“Everybody wants to change
the world, but nobody wants
to change the toilet paper
roll.”

NEWS
IS COMING

— Probably not Aristotle

THE
END

BOYAN@GAIDA.AI