REACT.JS

A JavaScript library for building user interfaces

THE INTERNET

World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
: Pointers to the world's online information, subjects , W3 servers, etc.

Help
: on the browser you are using

Software Products
: A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)

Technical
: Details of protocols, formats, program internals etc

Bibliography
: Paper documentation on W3 and references.

People
: A list of some people involved in the project.

History
: A summary of the history of the project.

How can I help ?
: If you would like to support the web..

Getting code
: Getting the code by anonymous FTP , etc.

CAUTION!

anyone
wanna
chill
tn?

Welcome to my
Homepage!

THE INTERNET IS A SERIES OF DOCUMENTS

- Static HTML with inline styling (gradually CSS)
- Some images
- Interactivity mostly meant linking to other pages
- “Brochureware”

BUZZWORD

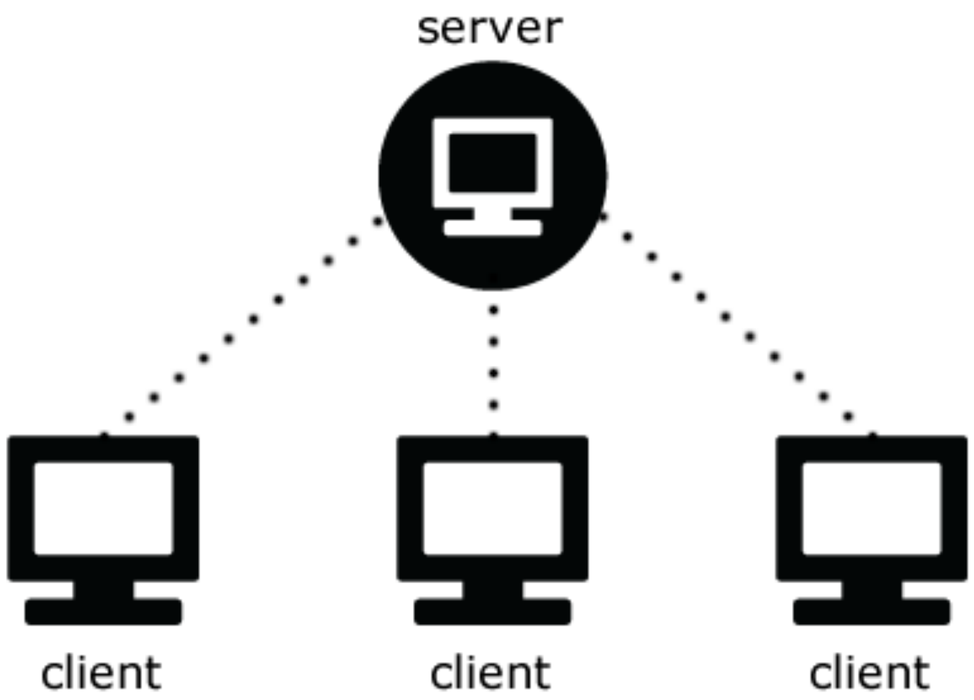
WEB 2.0

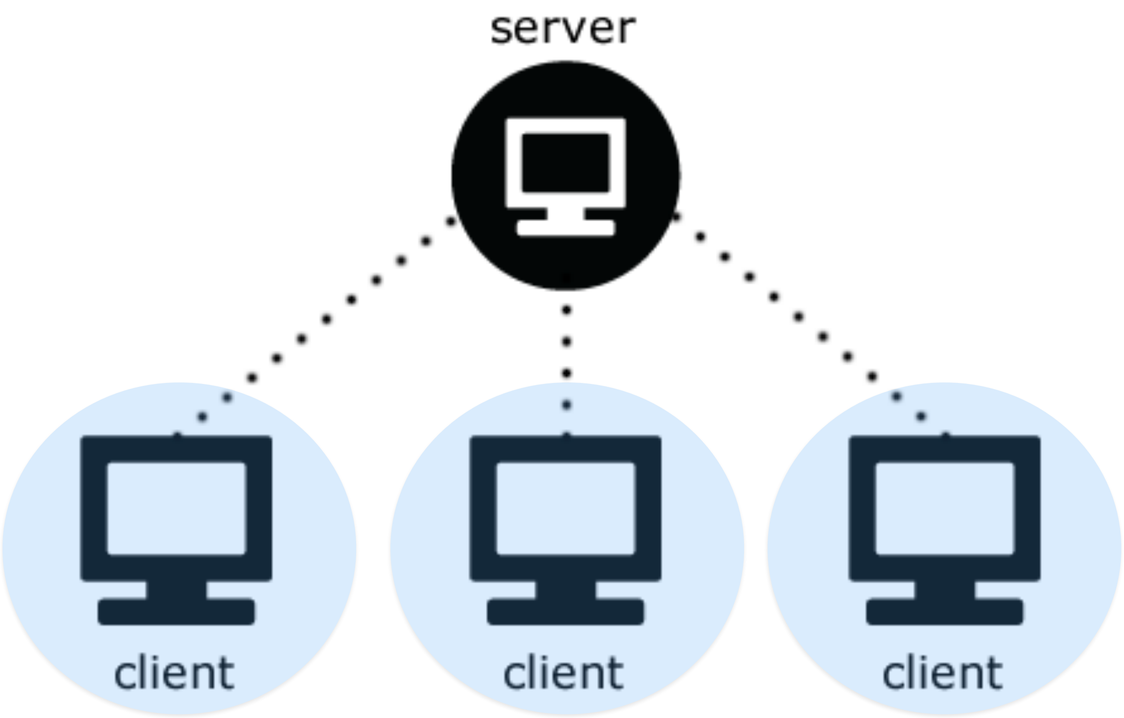
THE INTERNET IS A SERIES OF APPLICATIONS

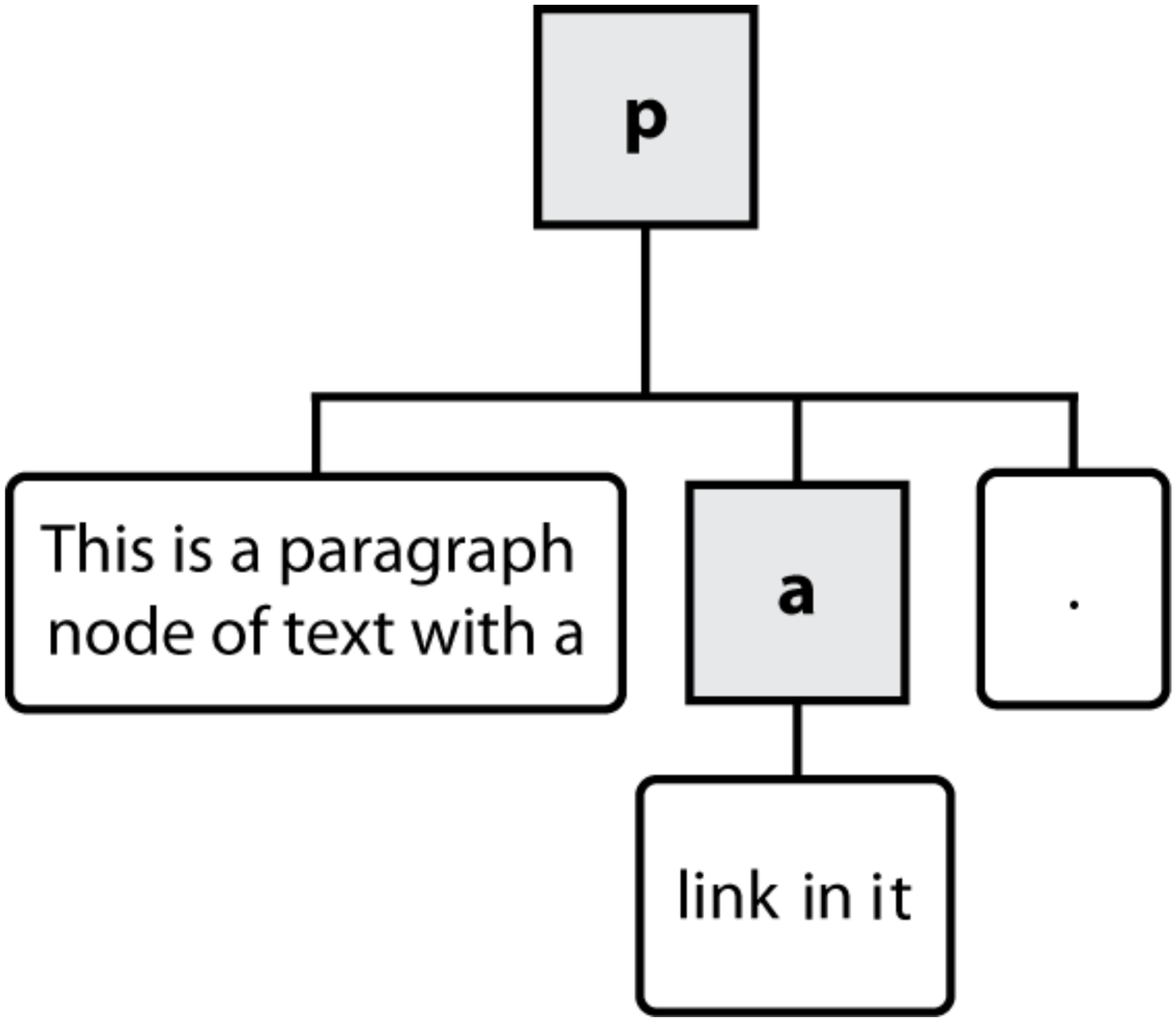
- True interactive applications like Gmail and Google Maps arrive.
- Applications do not refresh; facilitate tons of user events, hardware.
- The browser becomes a maturing application platform.
- “Software”

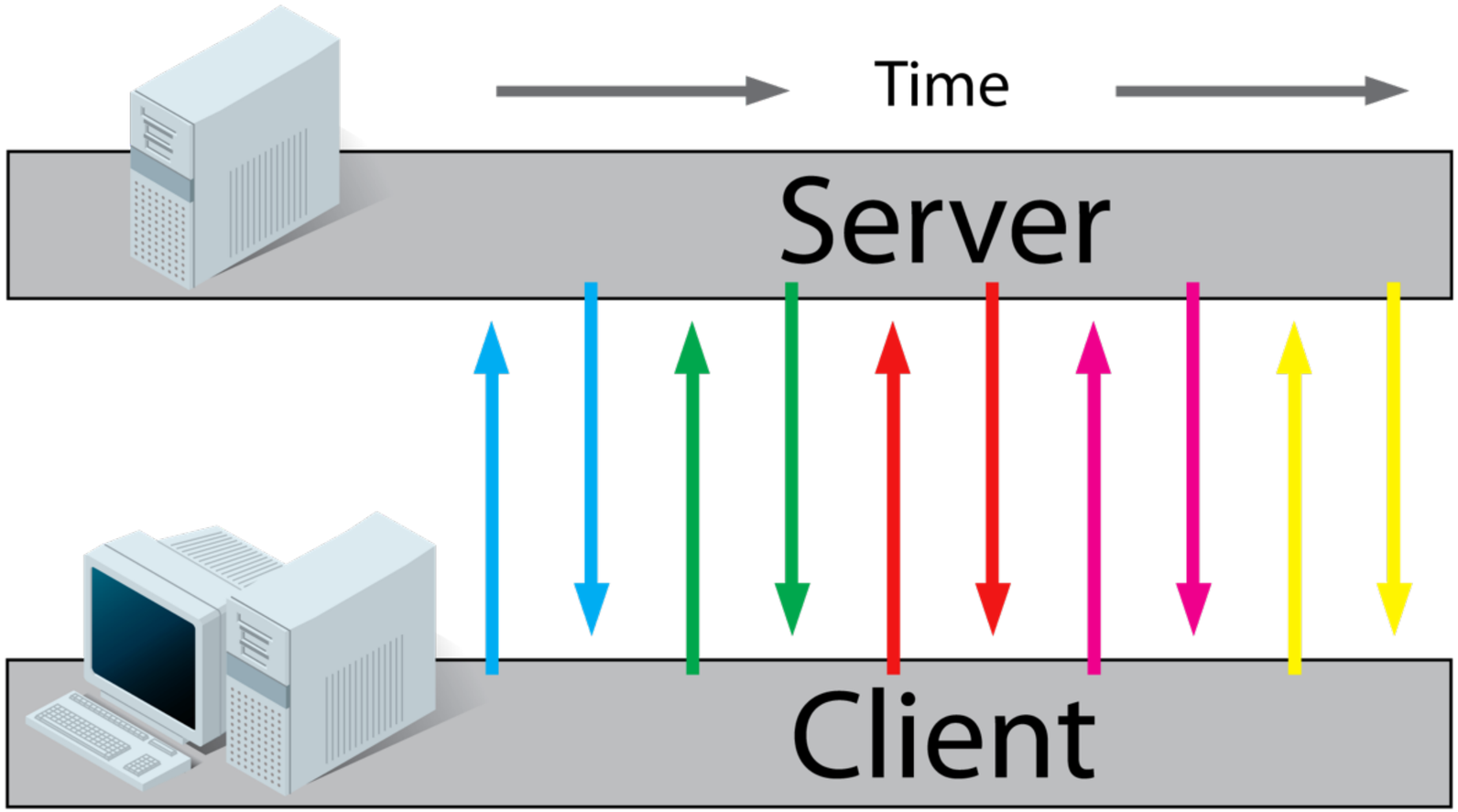
DRIVEN BY JAVASCRIPT

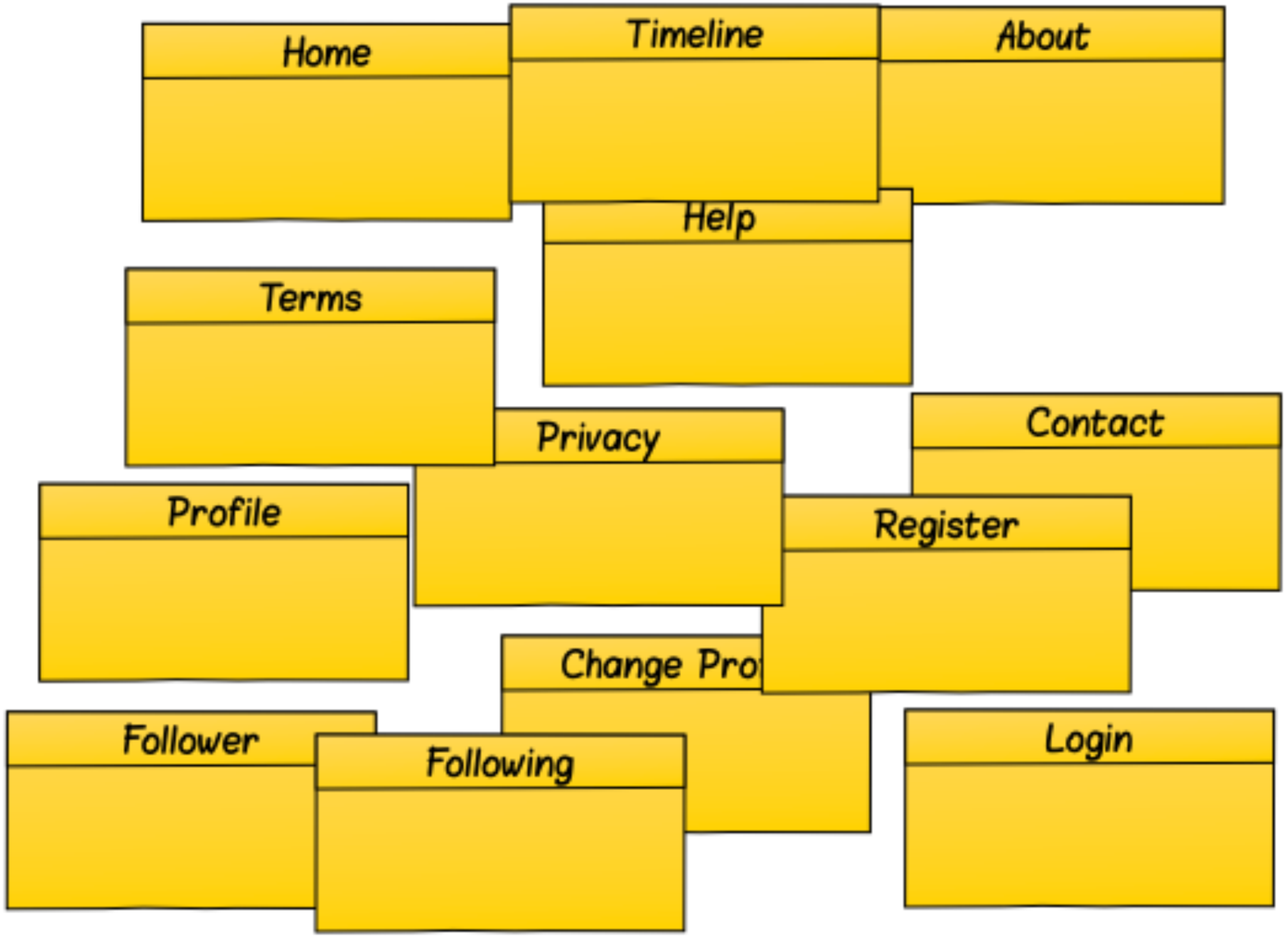
The little language that could

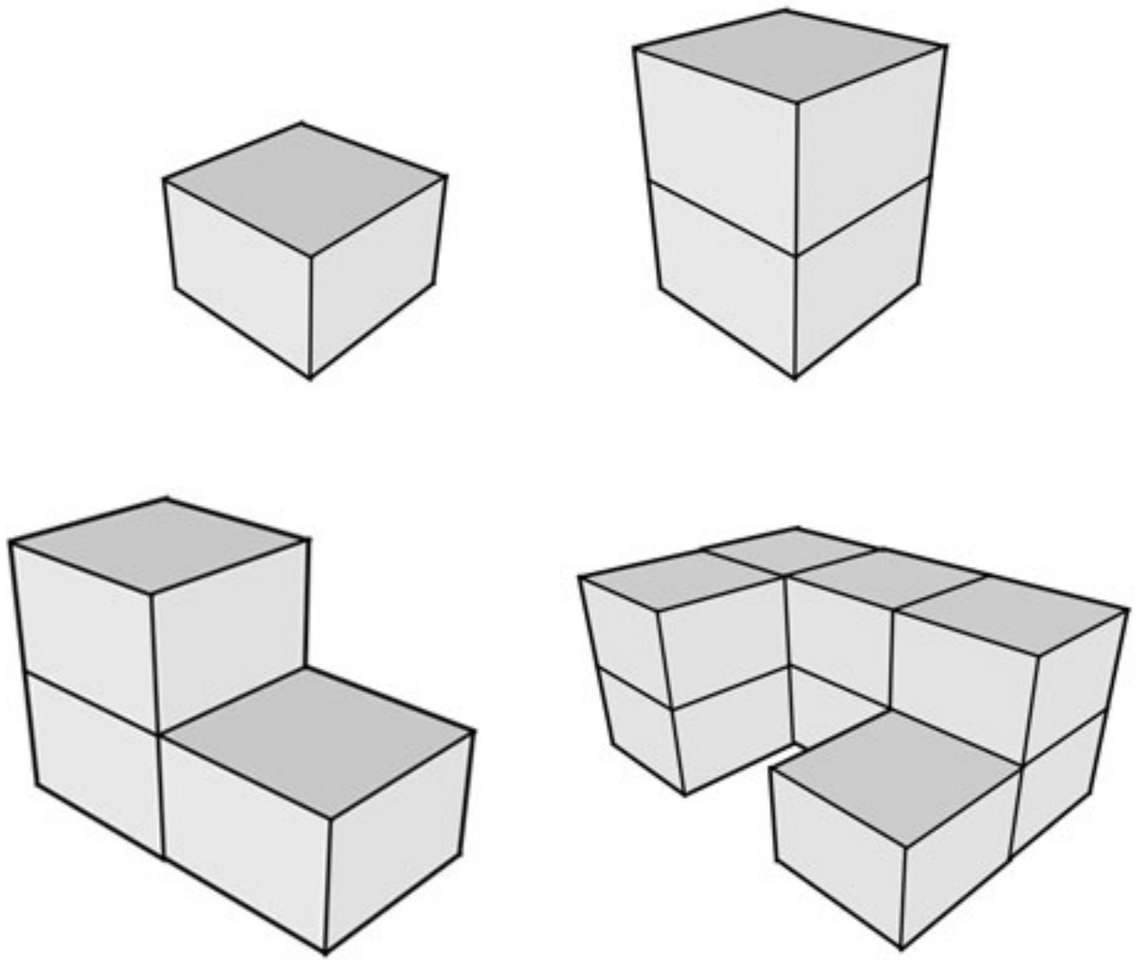

server
client
client
client


server
client
client
client


p
This is a paragraph node of text with a
a
.
link in it


Time
Server
Client


Home
Timeline
About
Help
Terms
Privacy
Contact
Profile
Register
Change Pro
Follower
Following
Login



SERIOUS SOFTWARE IS SERIOUS

- Must be easily understood.
- Must be easily maintained.
- Must scale.
- Must work.

“Let’s put the day’s hotels on the DOM node”

Cowboy Coder, 2010

ember

BACKBONE.JS

APPLICATION FRAMEWORK

- Provides generic functionality and environment.
- Establishes reusable patterns and architecture.
- Allows focus on product-specific code.
- Suggests solutions to platform's major problems.

SO EVERYTHING IS OKAY FOREVER, RIGHT?

NOPE!

- **Problems we face**
  - Portability & reusability
  - Performance
  - Managing state and a-synchronicity

PORTABILITY & REUSABILITY

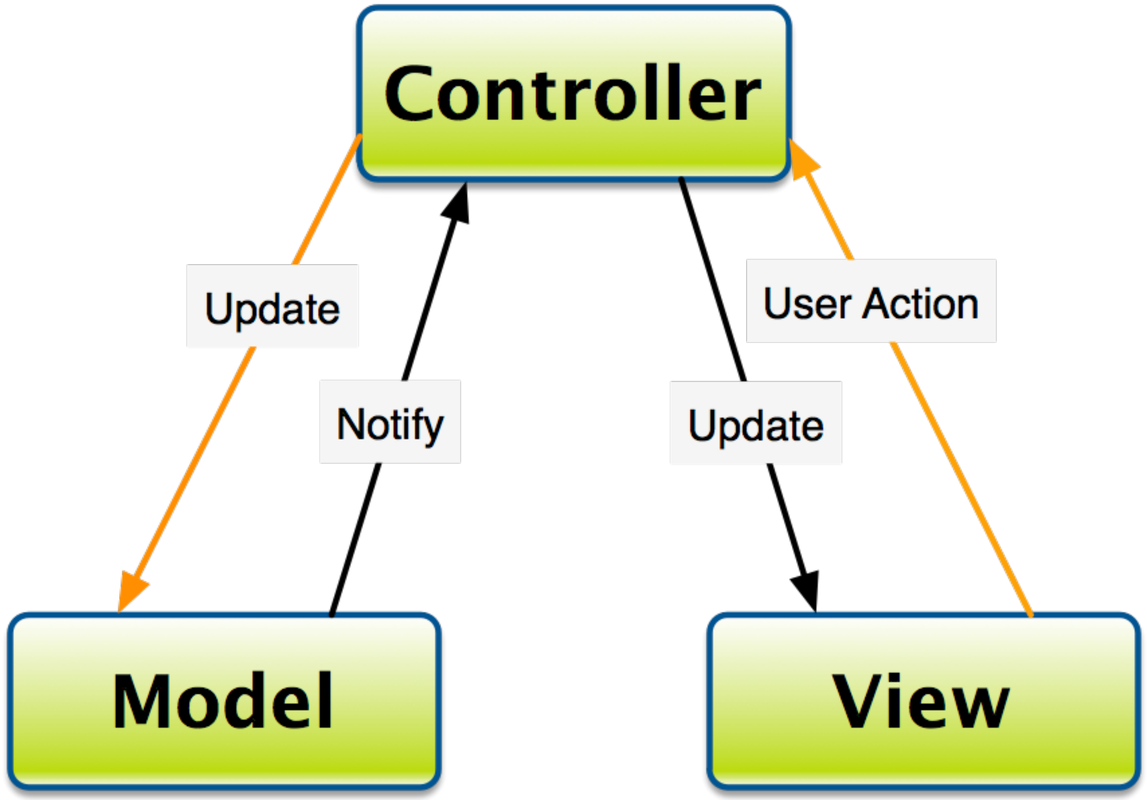
- Code written within a specific framework often has to conform to that framework and/or use domain-specific languages that aren't portable outside that framework
- Difficult to move and re-use existing code when technologies change
- Here in internet-town, technologies change a lot, and quickly!

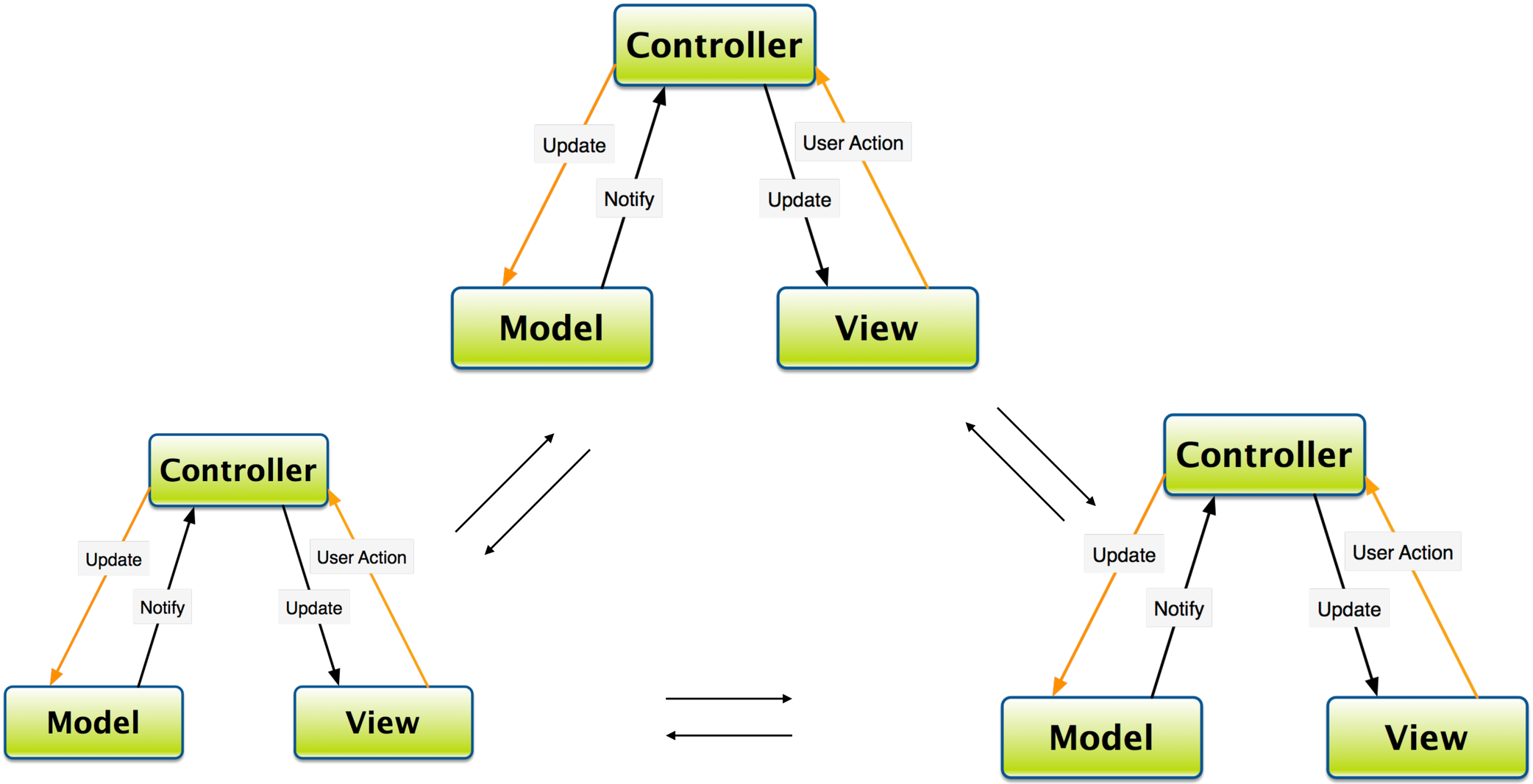
PERFORMANCE

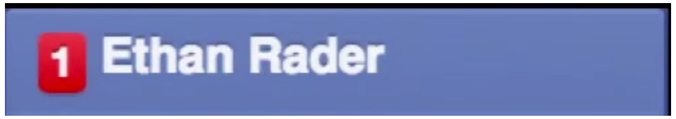
- Technically speaking, the DOM never imagined that web apps would get this big
- Consequently, mutating the DOM is an expensive operation - each interaction requires things like re-draws, stylesheet recalculation, etc
- User interfaces for web apps need to change the DOM, sometimes a lot

STATE

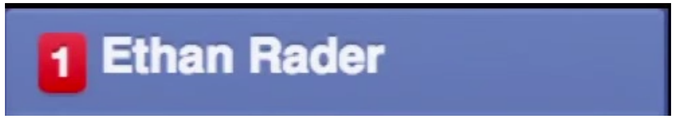
- Most web frameworks up until recently have been built on the MVC pattern
- The introduction of shared mutable state makes the state of your application difficult to predict
- Existing patterns of data-binding in the DOM can make reacting to a change difficult to predict


Controller
Update
Notify
User Action
Update
Model
View


Controller
Update
Notify
User Action
Update
Model
View
Controller
Update
Notify
User Action
Update
Model
View
Controller
Update
Notify
User Action
Update
Model
View

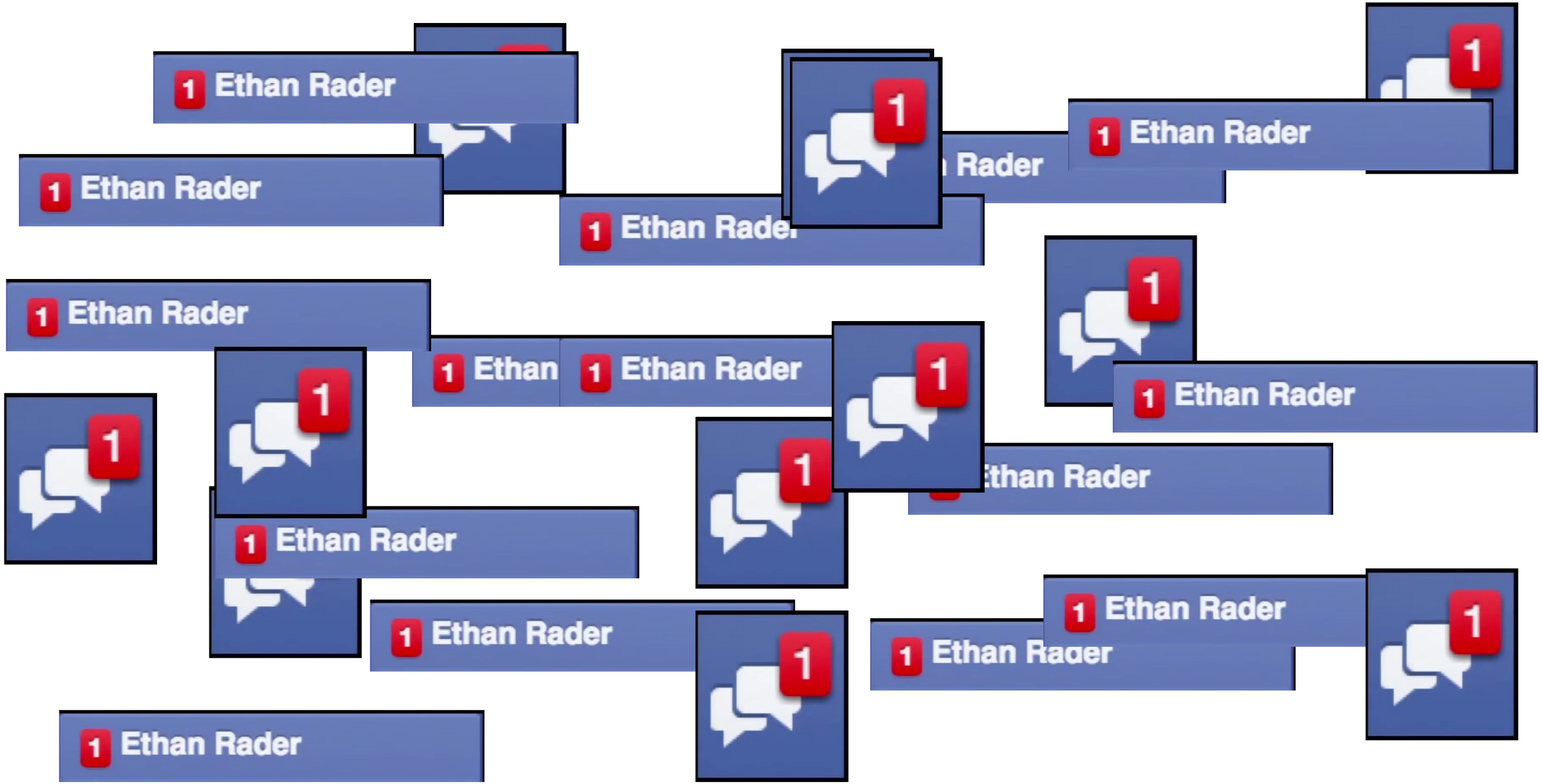


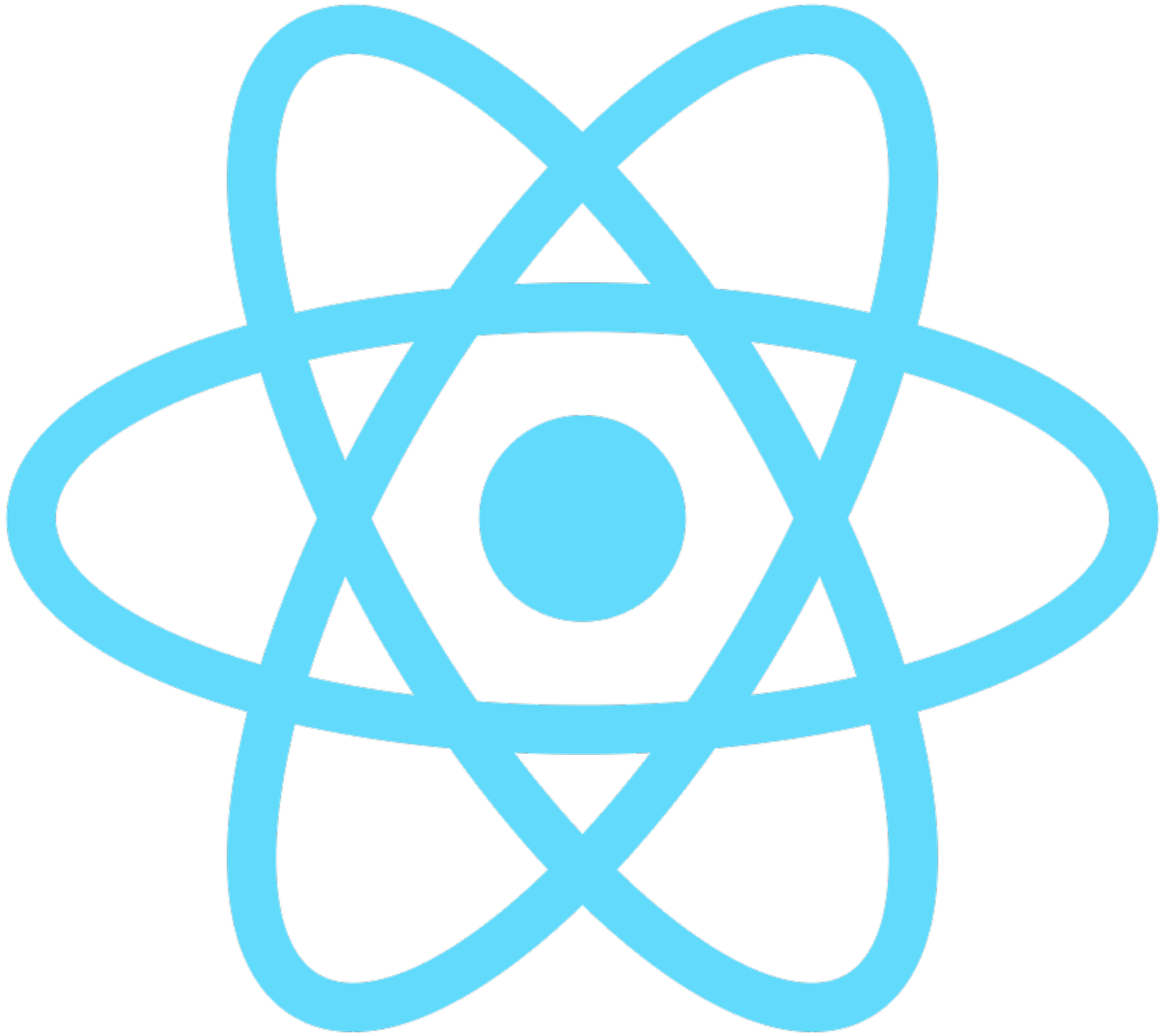

```
var unreadMessageCounts = {};
function onNewMessage(msg) {
  var unreadCount = ++unreadMessageCounts[msg.threadId];
  $('#messageCount' + msg.threadId).text(unreadCount);
}
```



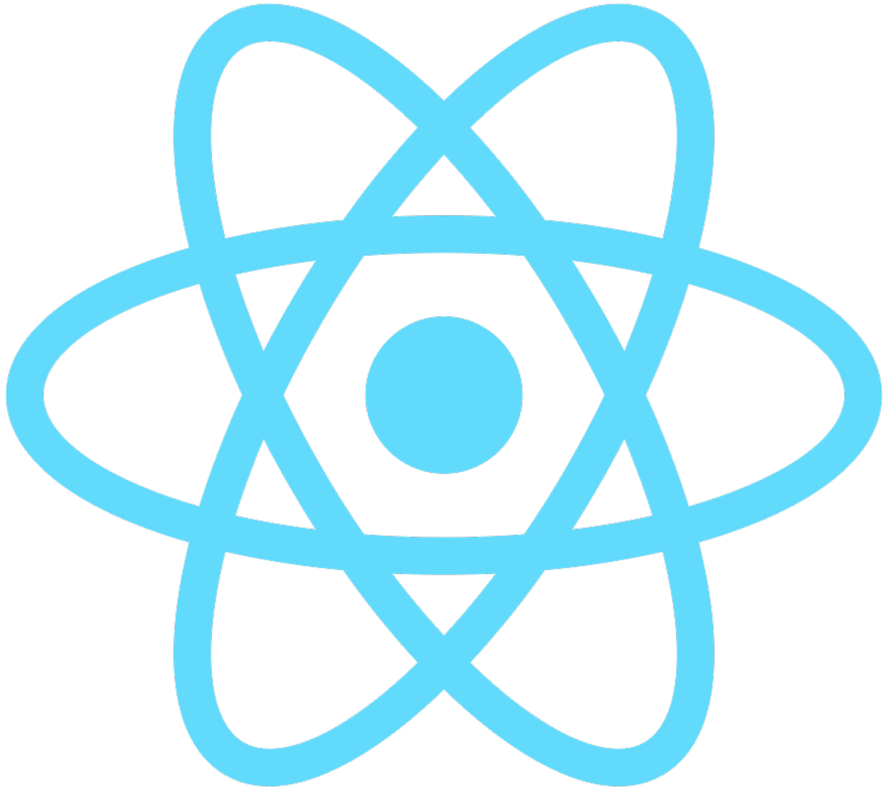
```
var unreadMessageCounts = {};
var unreadThreadCount = 0;

function onNewMessage(msg) {
  var unreadCount = ++unreadMessageCounts[msg.threadId];
  if (unreadCount === 1) {
    unreadThreadCount++;
    $('#unreadThreads').text(unreadThreadCount);
  }
  $('#messageCount' + msg.threadId).text(unreadCount);
}
```


1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader
1 Ethan Rader



REACT

- No data-binding: instead, data flows in one direction
- Just like a 90's era web page…only just for your client!
- state => view
- Minimize DOM manipulation by rendering to a Virtual DOM before rendering to the Real DOM

GETTING STARTED WITH REACT