

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.



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

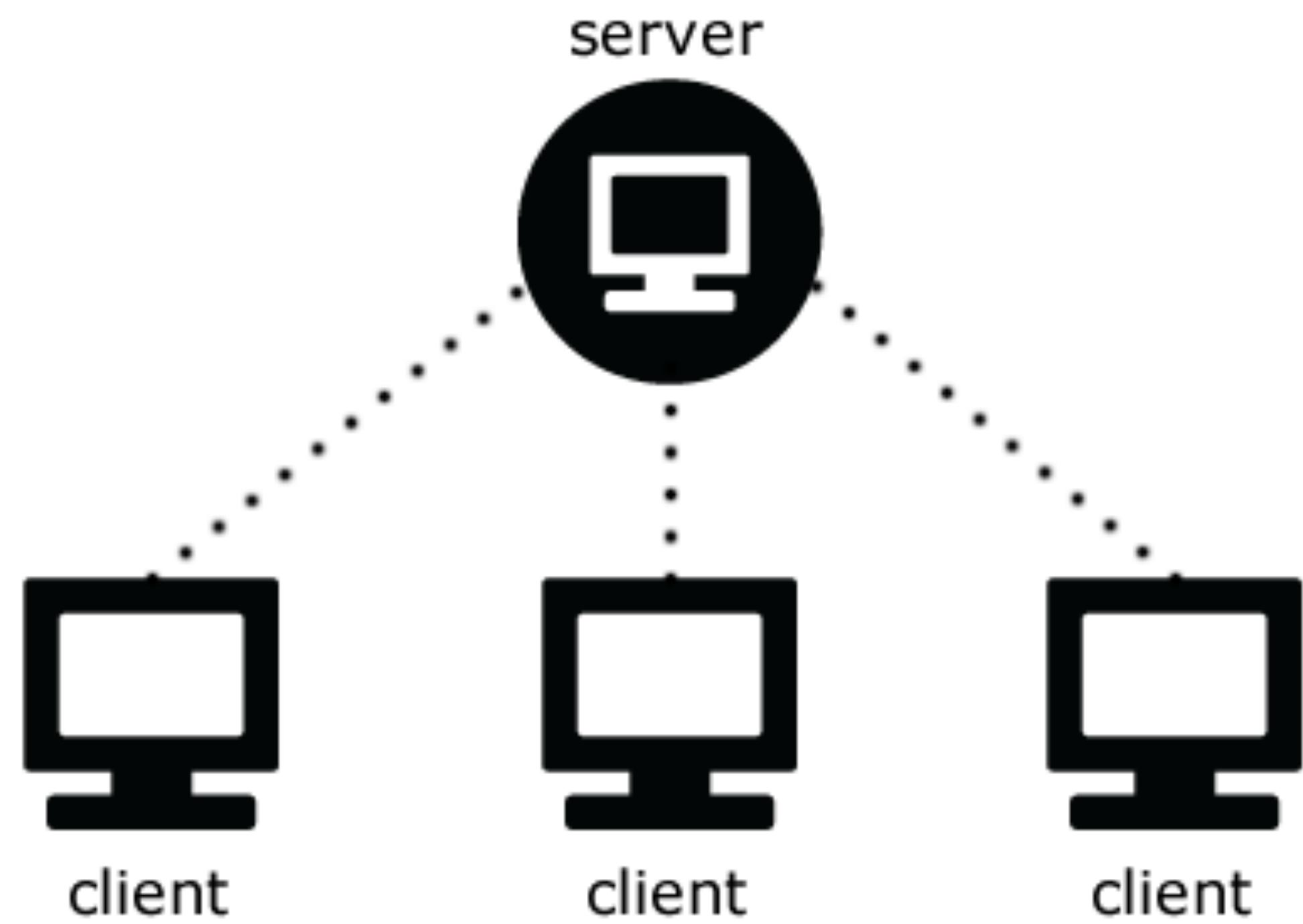


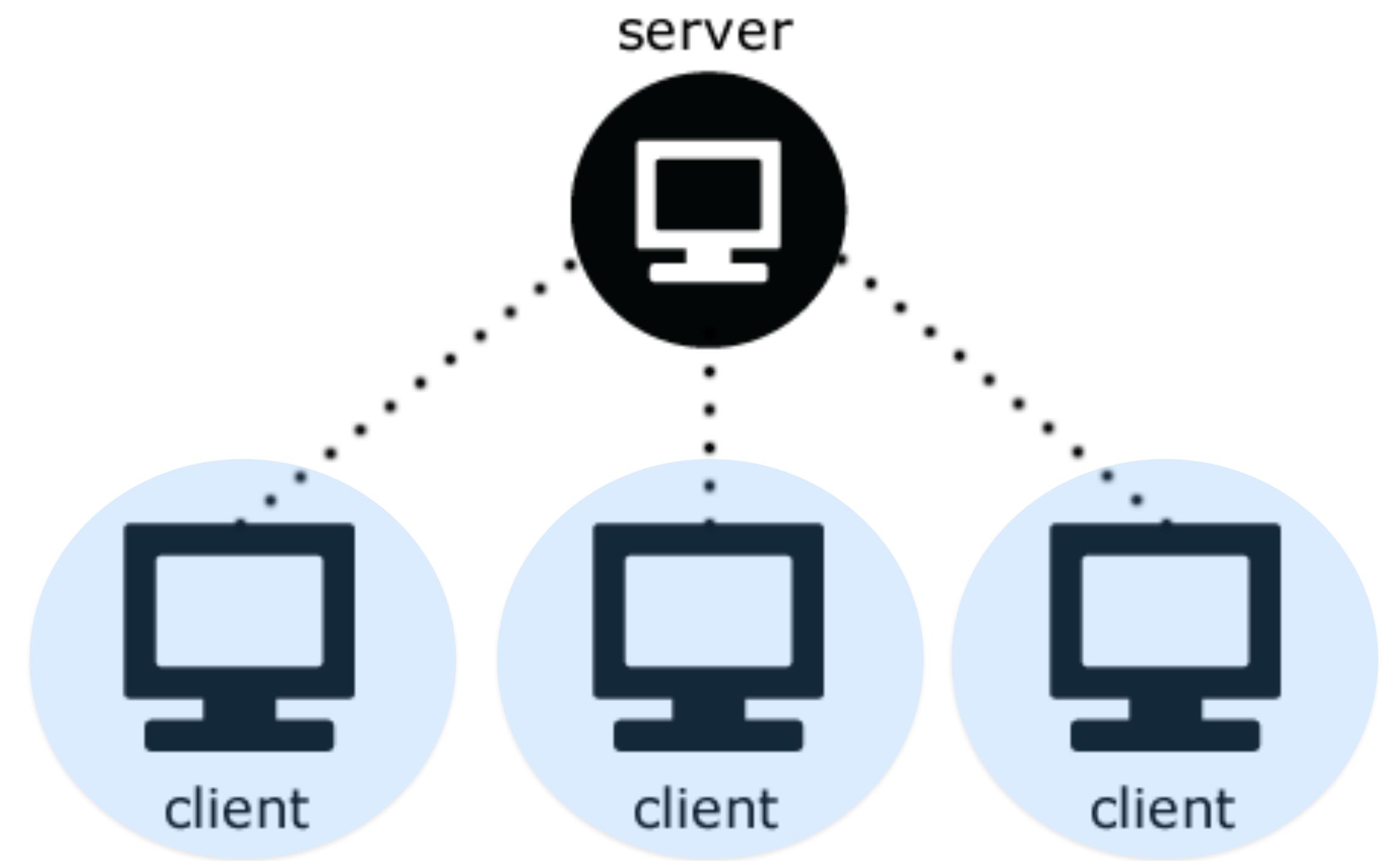
THE INTERNET IS A SERIES OF APPLICATIONS

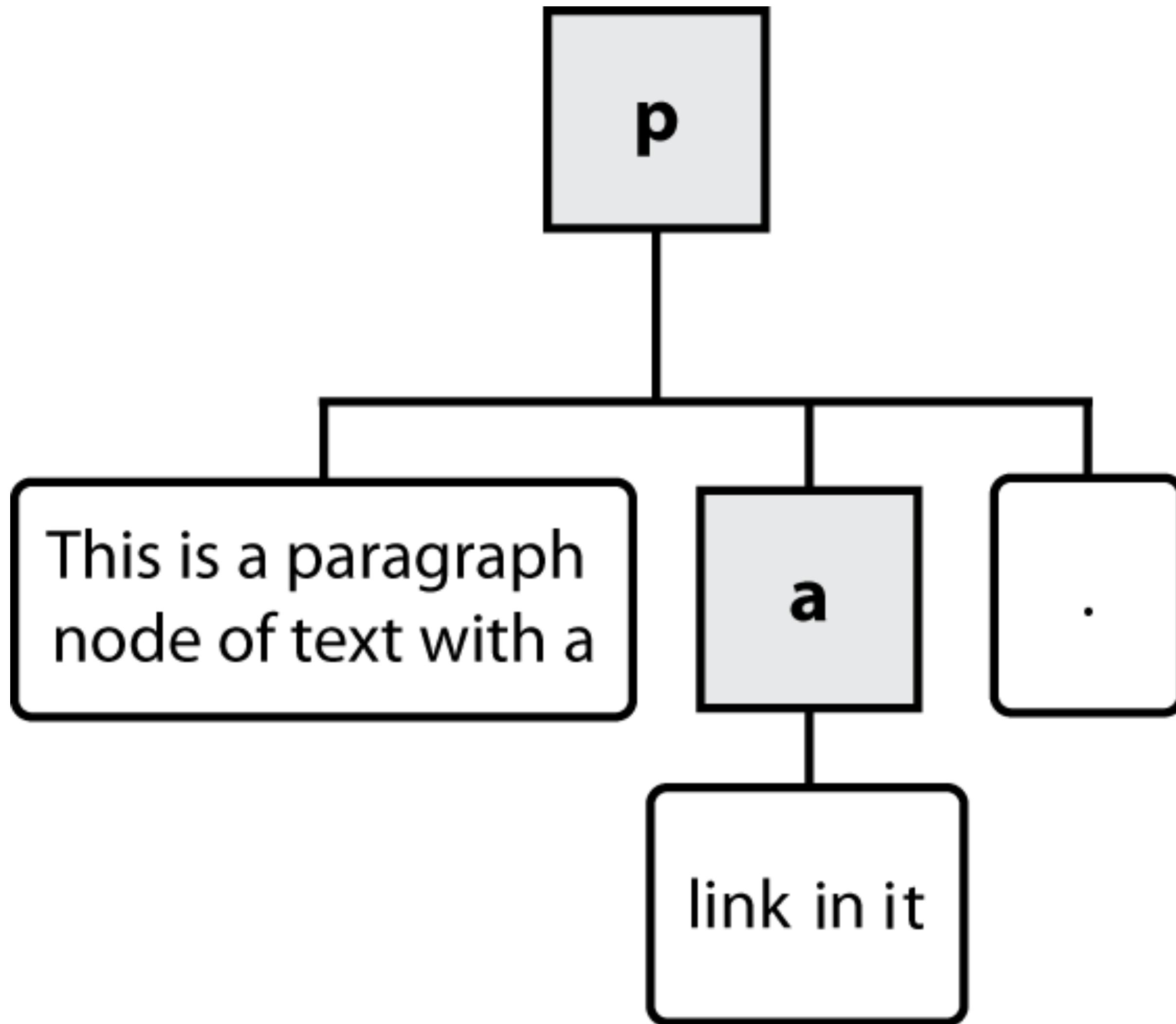
- True interactive applications like Gmail and Google Maps arrive.
- Applications do not refresh; facilitate tons of user events, hardware.
- The browser becomes a maturing application platform.
- “Software”

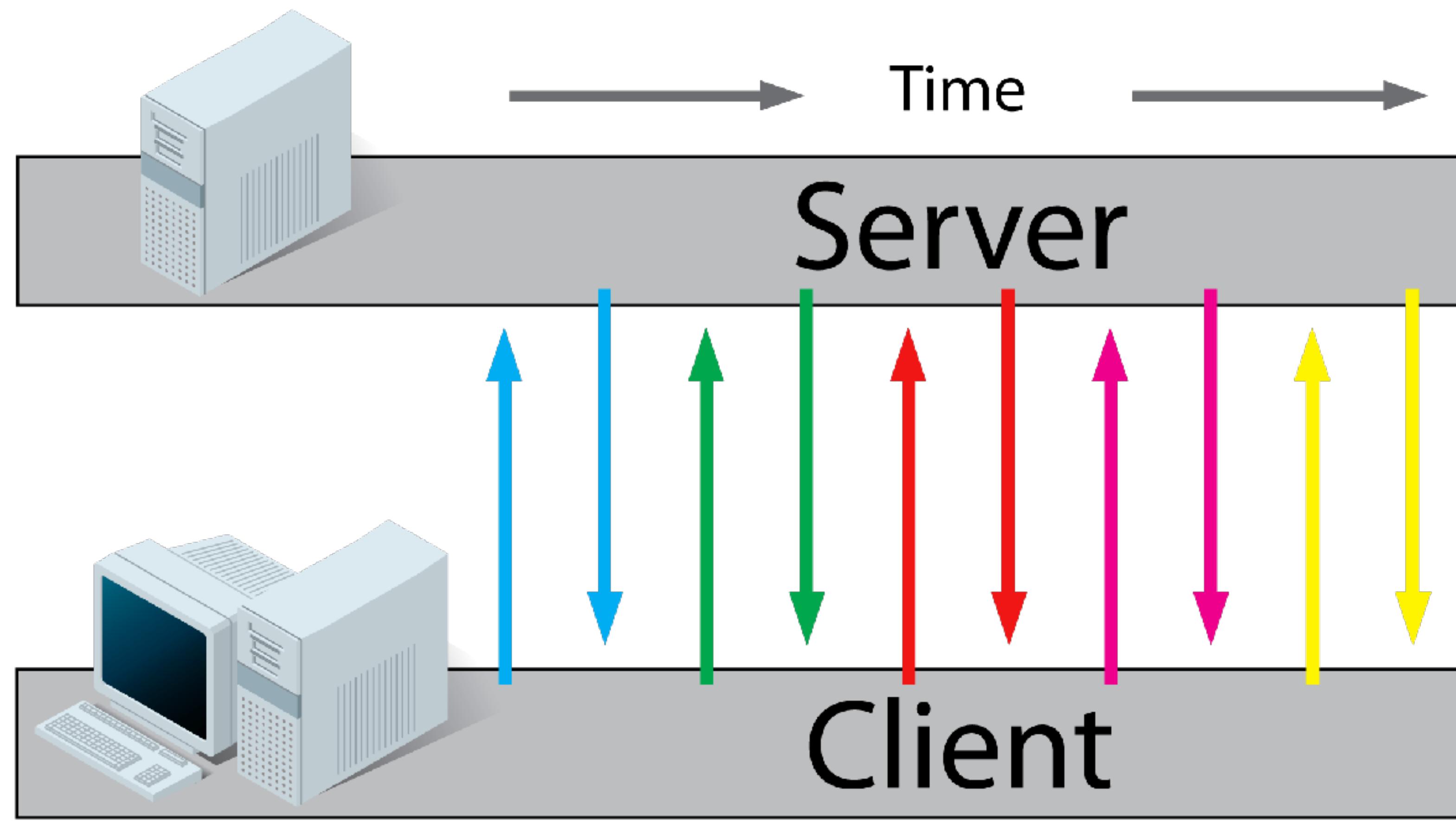
DRIVEN BY JAVASCRIPT

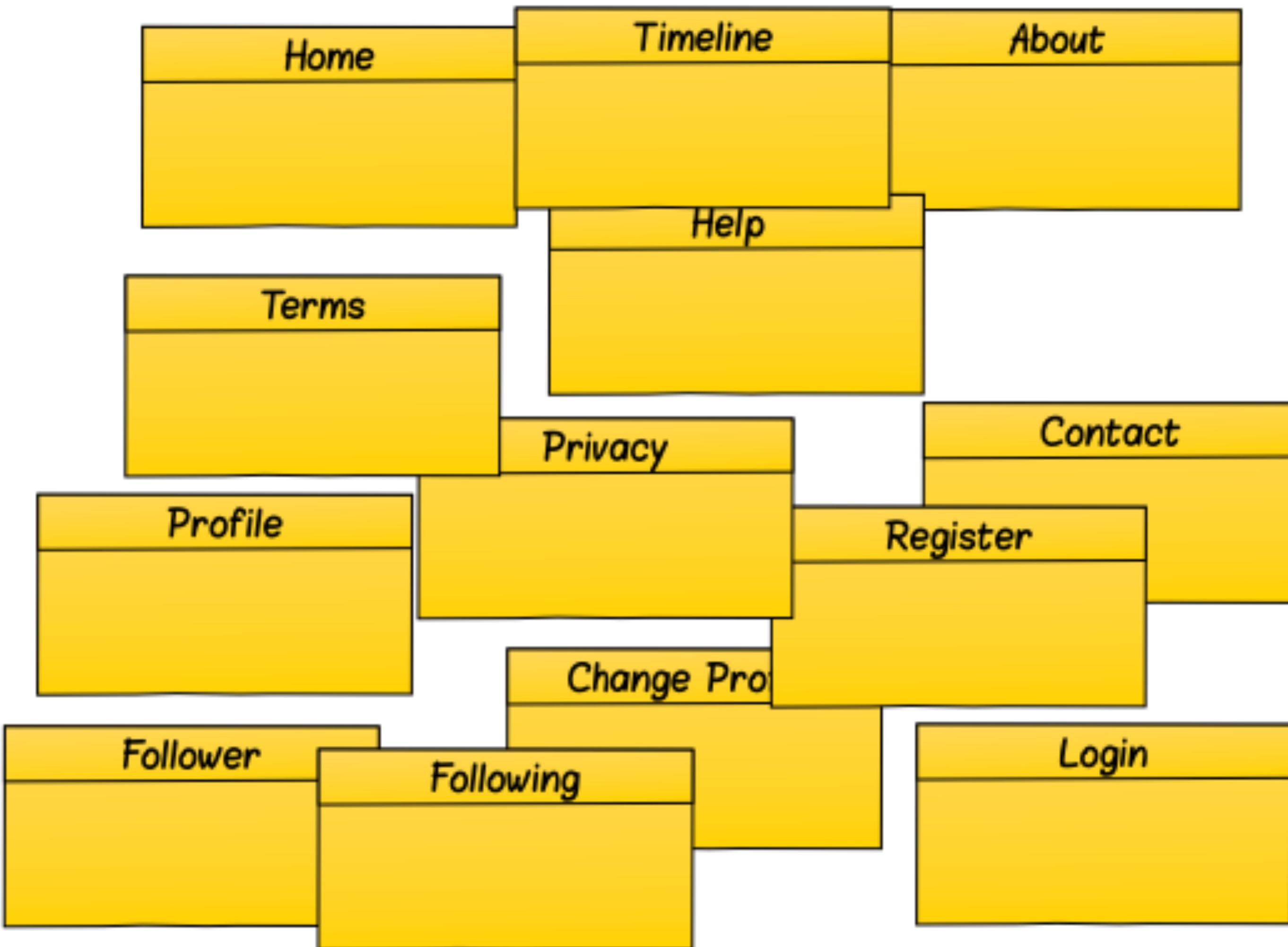
The little language that could

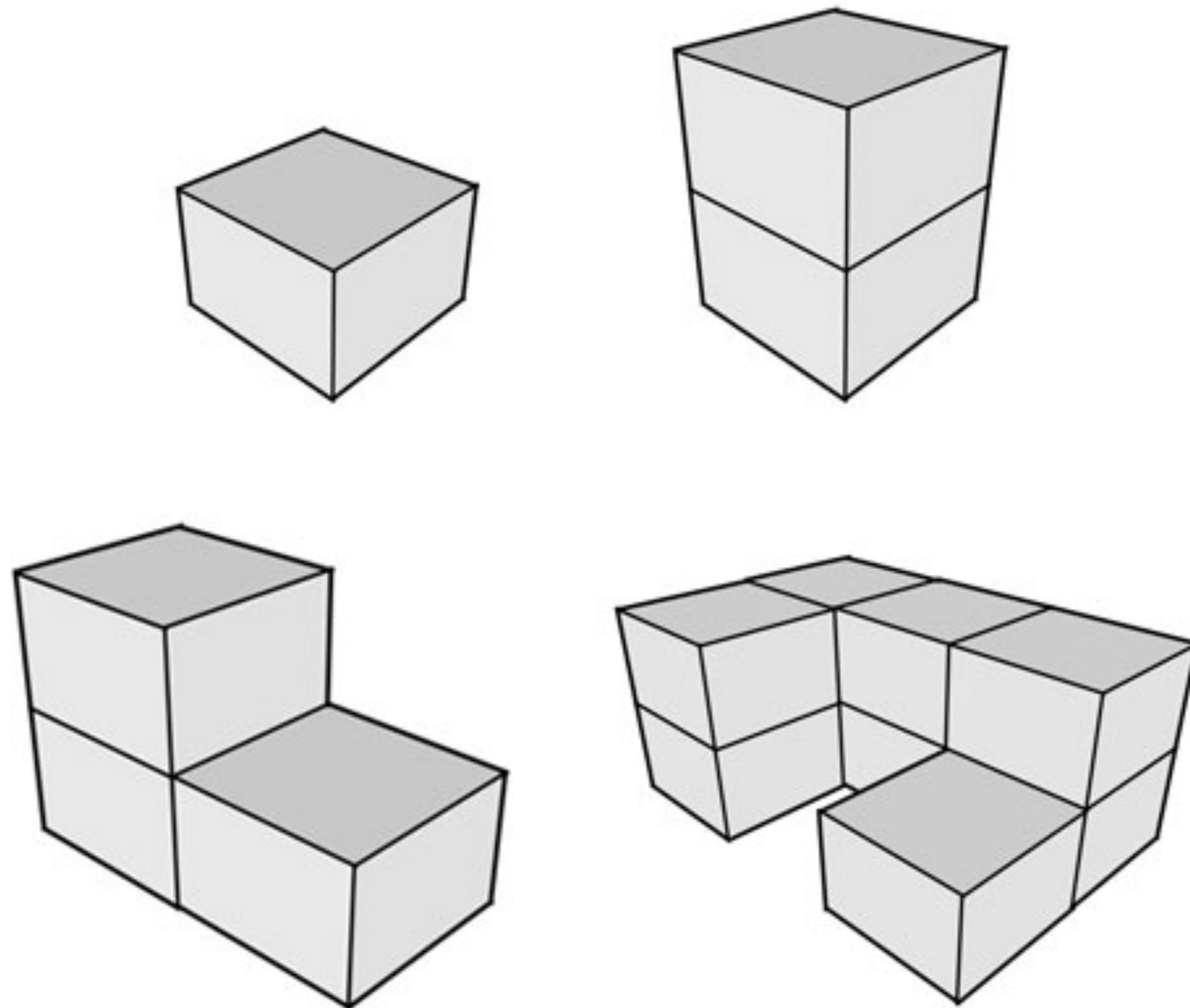












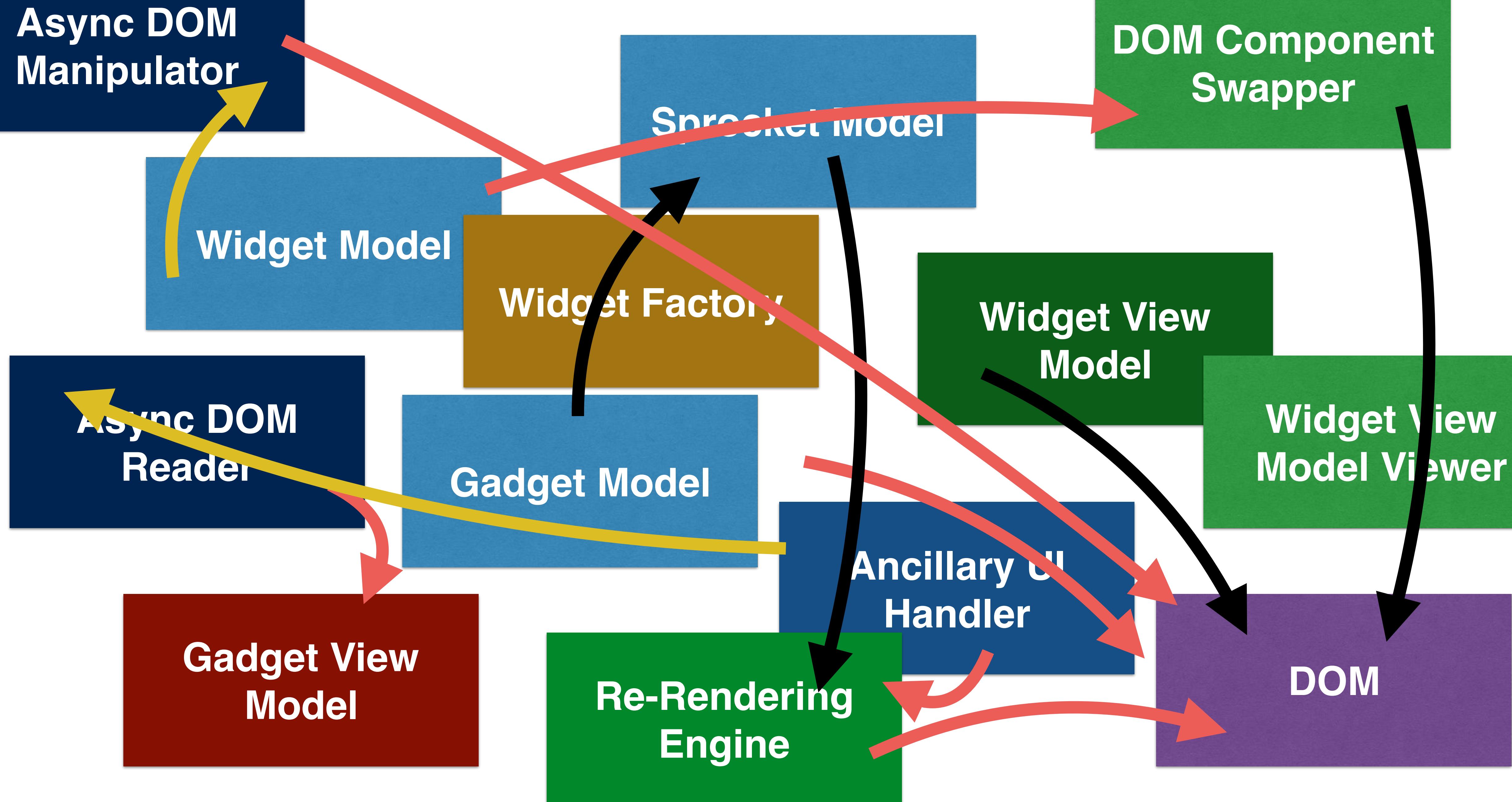
SERIOUS SOFTWARE IS SERIOUS

- 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



**"WAT THE (*@#^\$ EVEN IS
ASYNC_SUB_RENDERER_COMPONENT
_FACTORY_SYNC_CHECK???"**

—DEVE LOPER JR.

"UM... I'M NOT SURE ANYMORE"

—COWBOY CODY



BACKBONE.JS

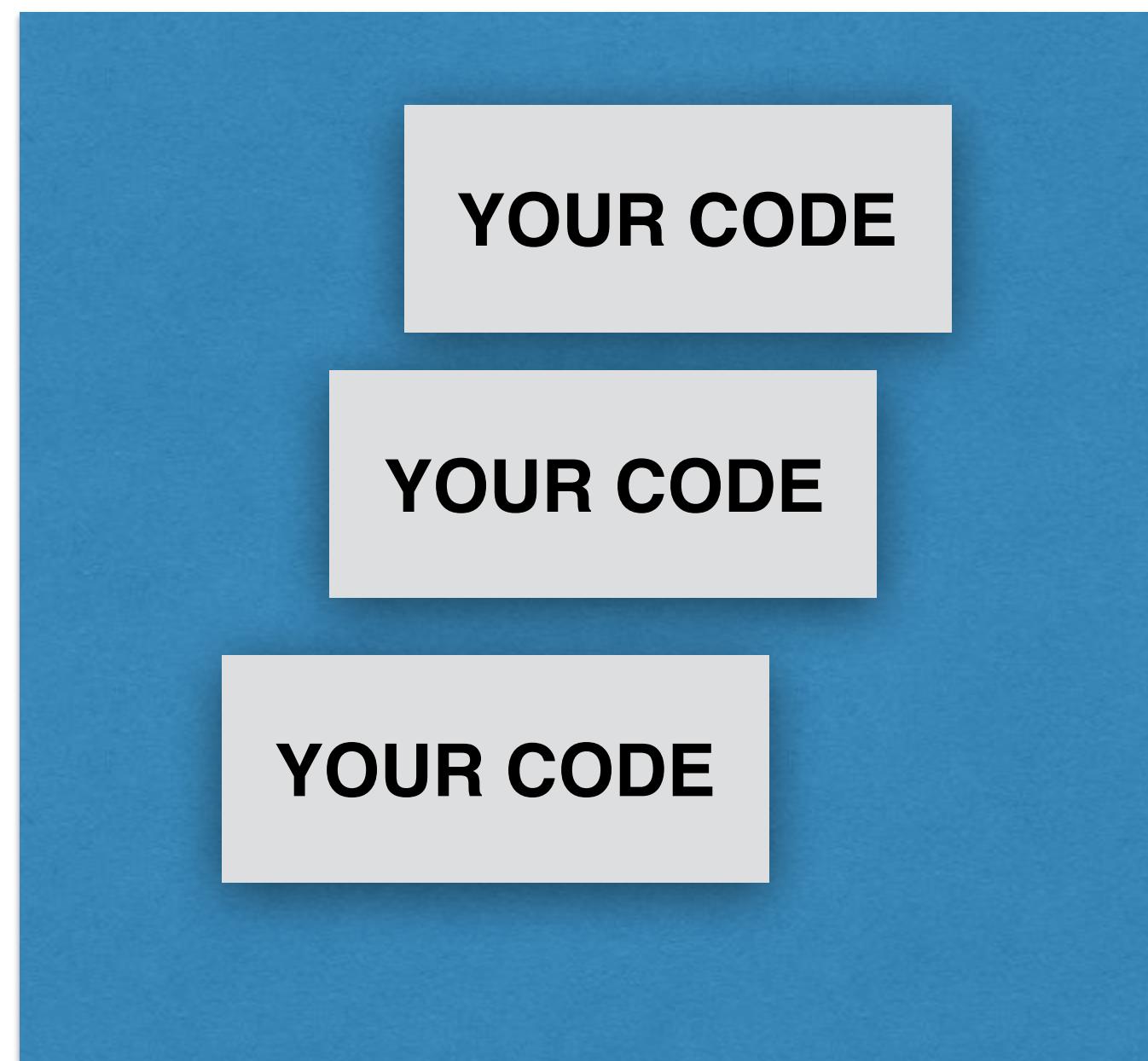


APPLICATION FRAMEWORK

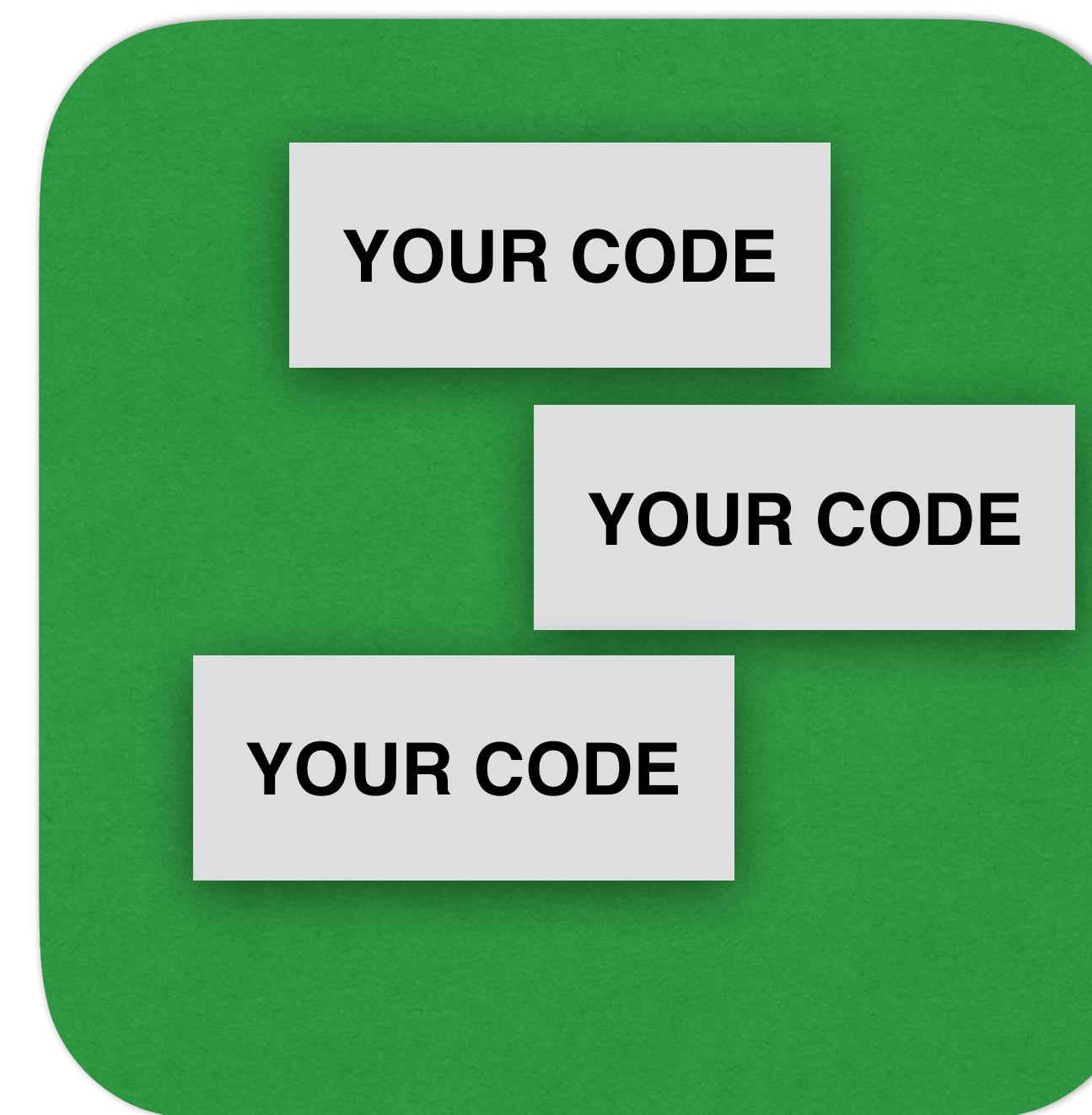
- Provides generic functionality and environment.
- Establishes reusable patterns and architecture.
- Allows focus on product-specific code.
- Suggests solutions to platform's major problems.

Framey™

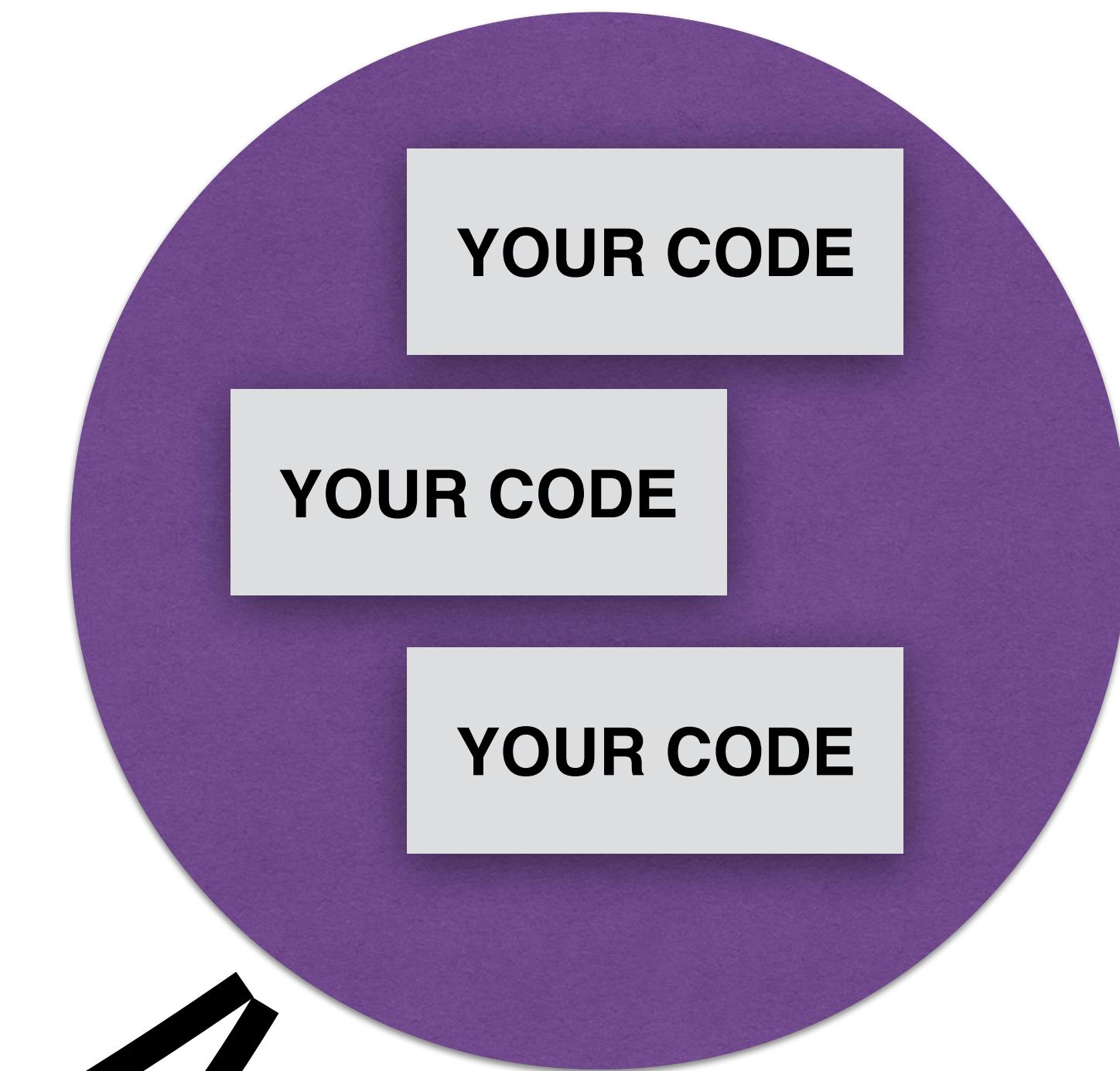
Foo Pieces



Bar Pieces



Baz Pieces



"WE USE FRAMEY™. CAN YOU MAKE A PANEL SHOWING THE USER'S FAVORITE WIDGETS?"

—N. G. NEER

"SURE, THAT'S JUST A FOO WIRED TO A BAZ. GIMME A COUPLE HOURS."

—DEVE LOPER JR.

**SO EVERYTHING IS OKAY
FOREVER, RIGHT?**

NOPE!

- **(Some) Problems we face**

- Portability & reusability ("um, I know Framey™... how do I make Foos in Boxy™? What's that, Boxy™ doesn't have Foos!?)")
- Performance
- Managing state and asynchronicity

PORABILITY & REUSABILITY

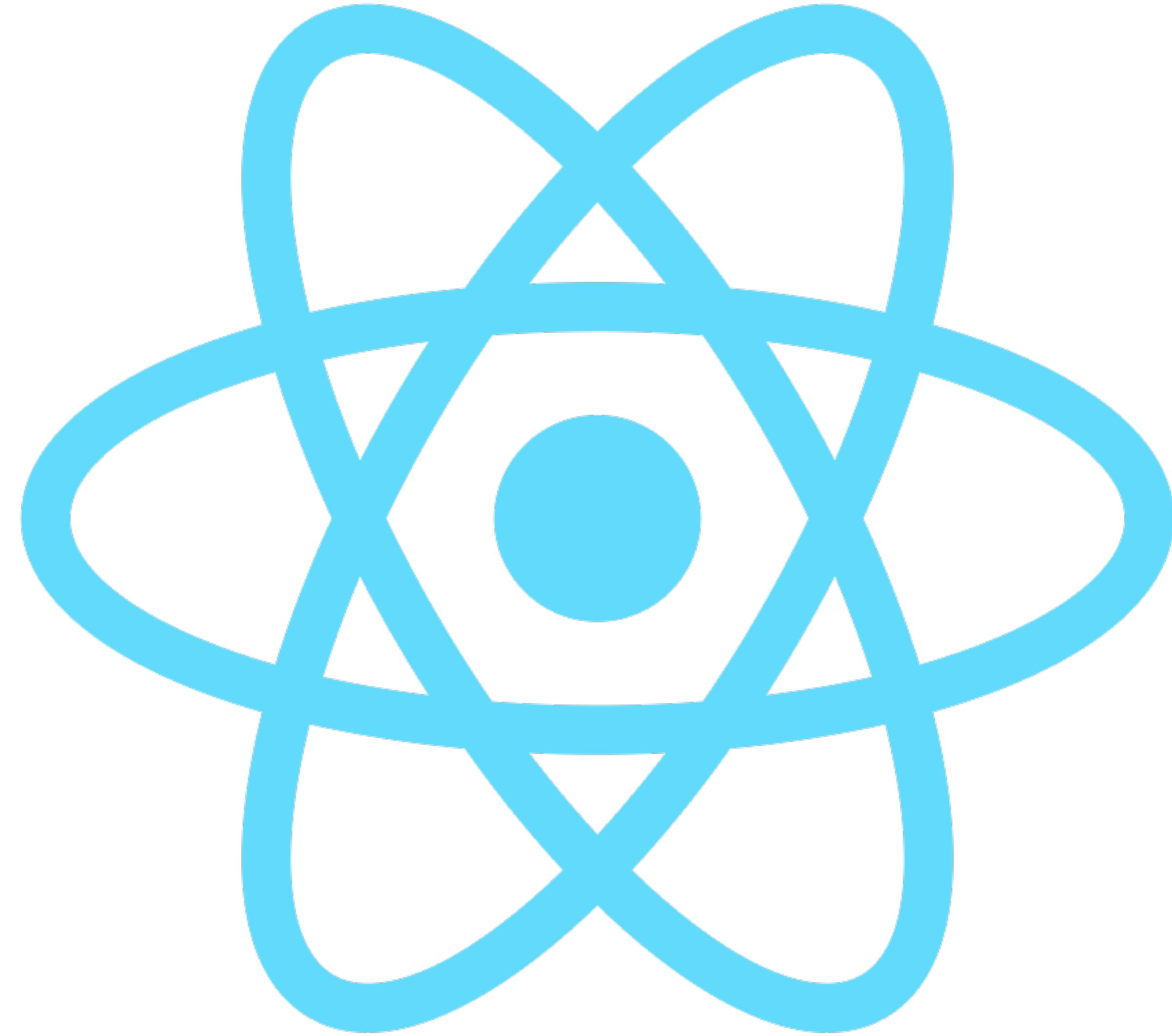
- **Code written within a specific framework often has to conform to that framework and/or use domain-specific languages that aren't portable outside that framework**
- **Difficult to move and re-use existing code when technologies change**
- **Here in internet-town, technologies change a lot, and quickly!**

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

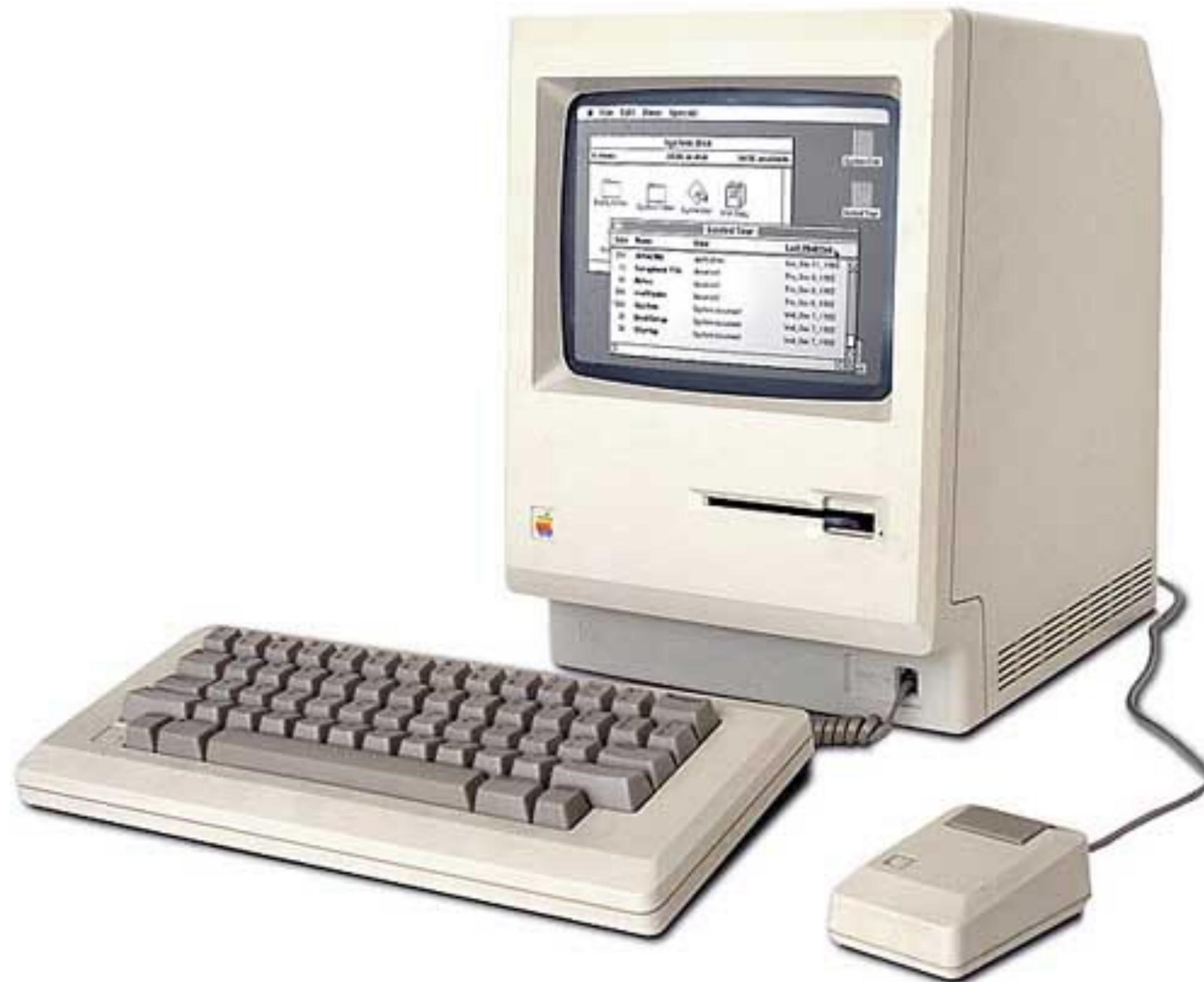
- **Most web frameworks up until recently have been built on the MVC pattern**
- **The introduction of shared mutable state makes the state of your application difficult to predict**
- **Existing patterns of data-binding in the DOM can make reacting to a change difficult to predict**



REACT

- **No data-binding, no MVC: instead, data flows in one direction**
- **Don't touch the DOM - declaratively describe your UI.**
- **Reactive rendering (If state changes, UI is entirely re-rendered to reflect the new state)**

COMPONENTS



COMPONENTS

- **Self-contained, concern-specific building blocks**
- **Components:**
 - are reusable
 - are composable
 - lead to separation of concerns
- **Allows a “Divide and conquer” approach to development**

REACT

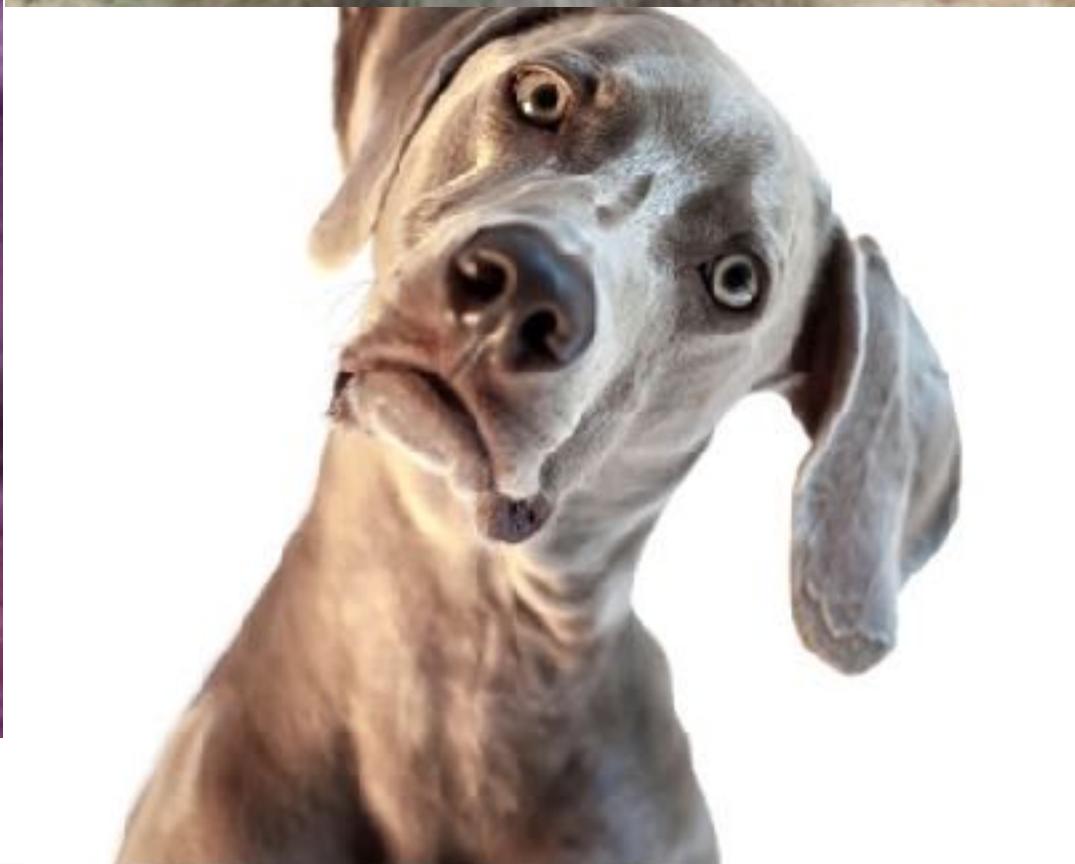
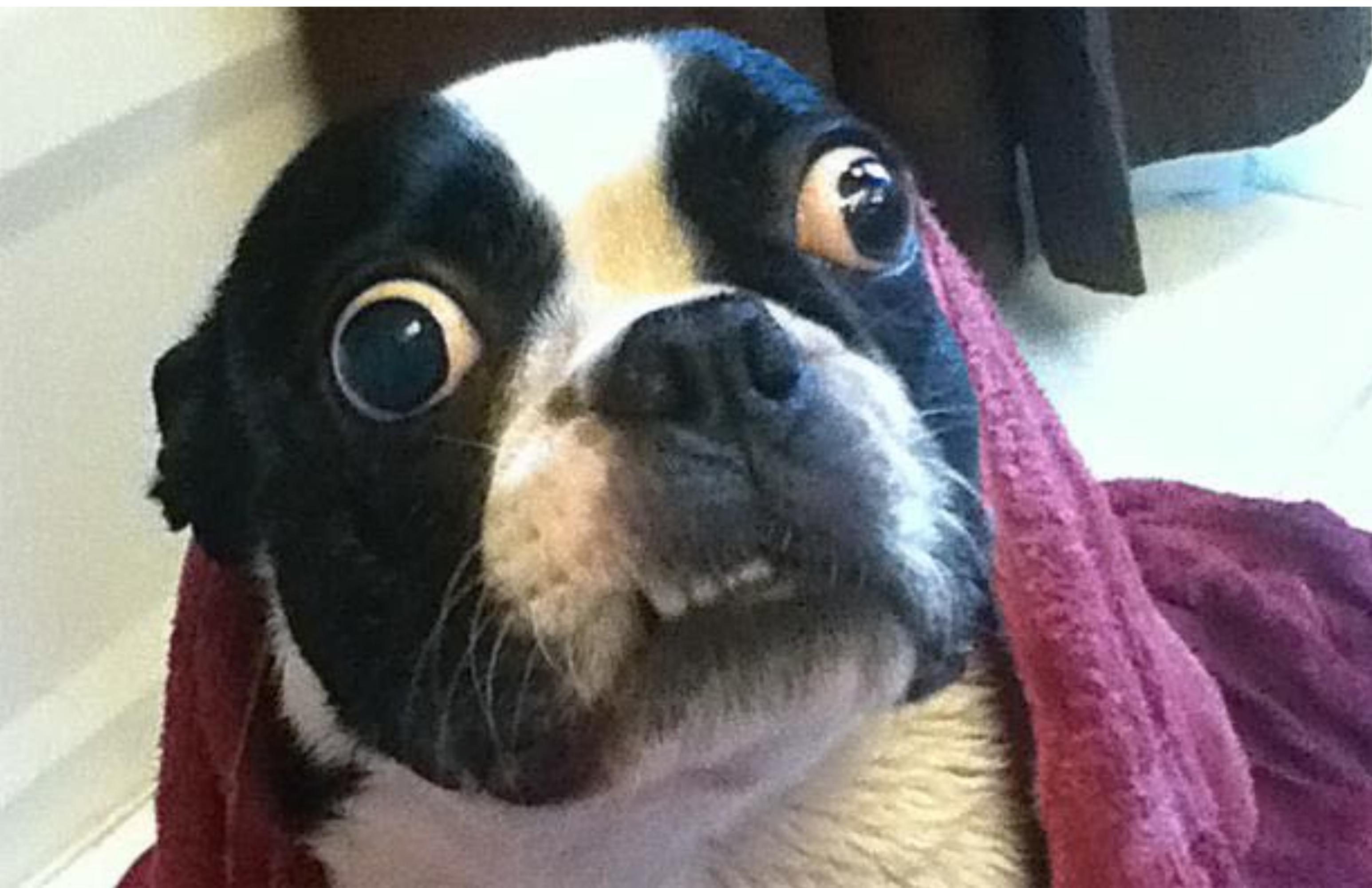
- ~~An Engine~~ **A library** for building composable user interfaces using JavaScript

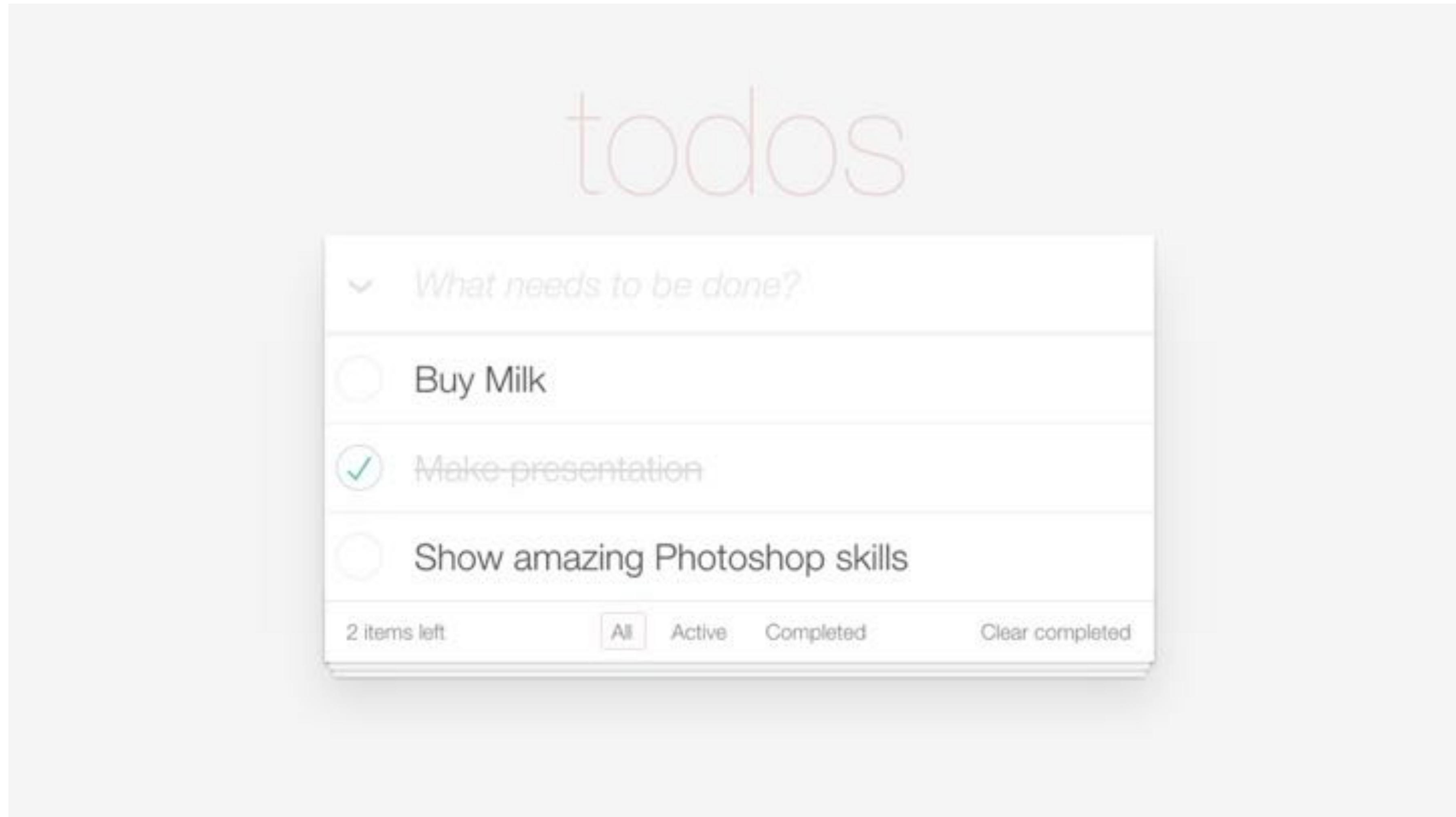
REACT

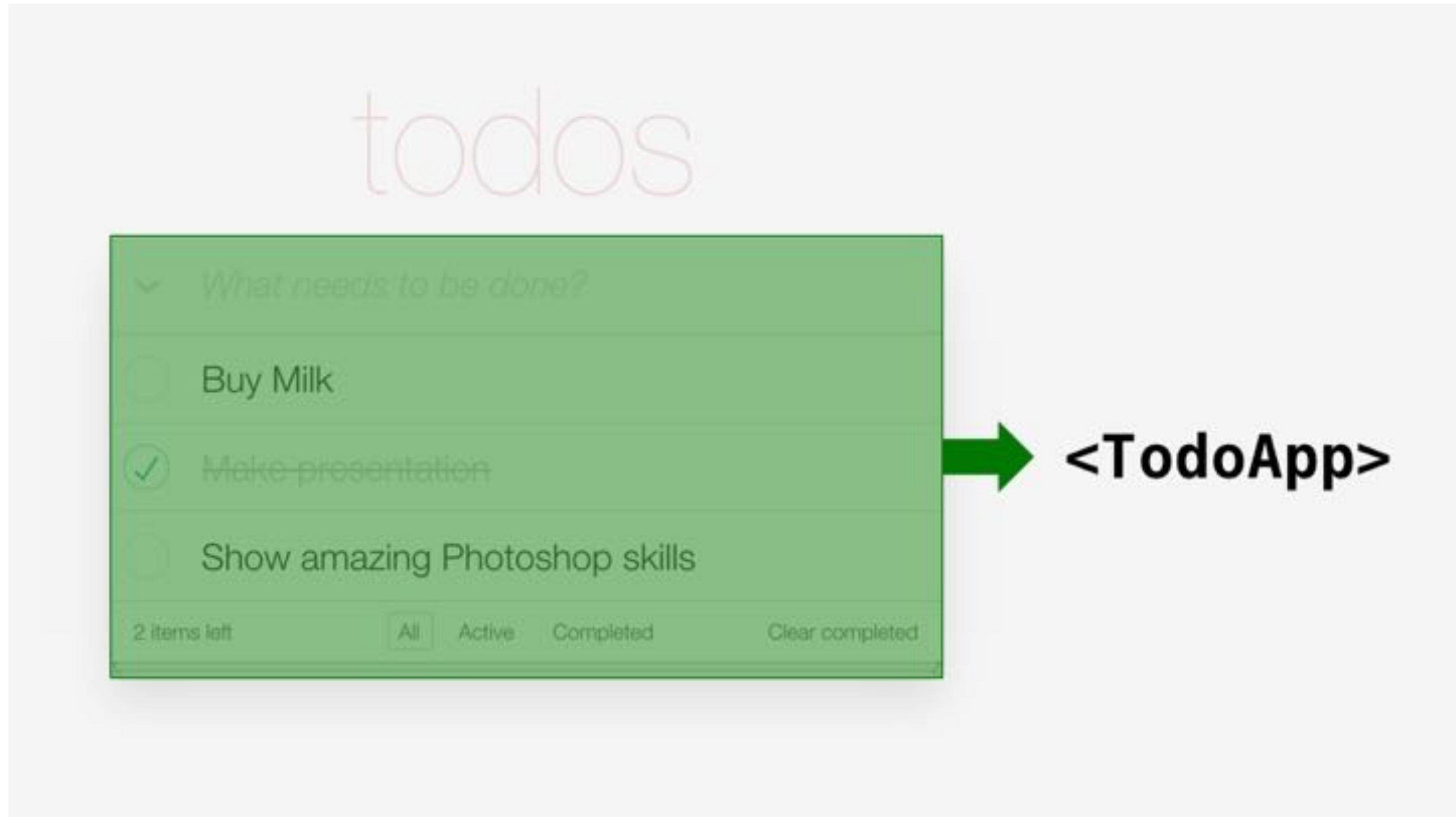
- ~~An Engine~~
~~A library~~ for building composable user interfaces using JavaScript
- Declarative, Component-Based approach

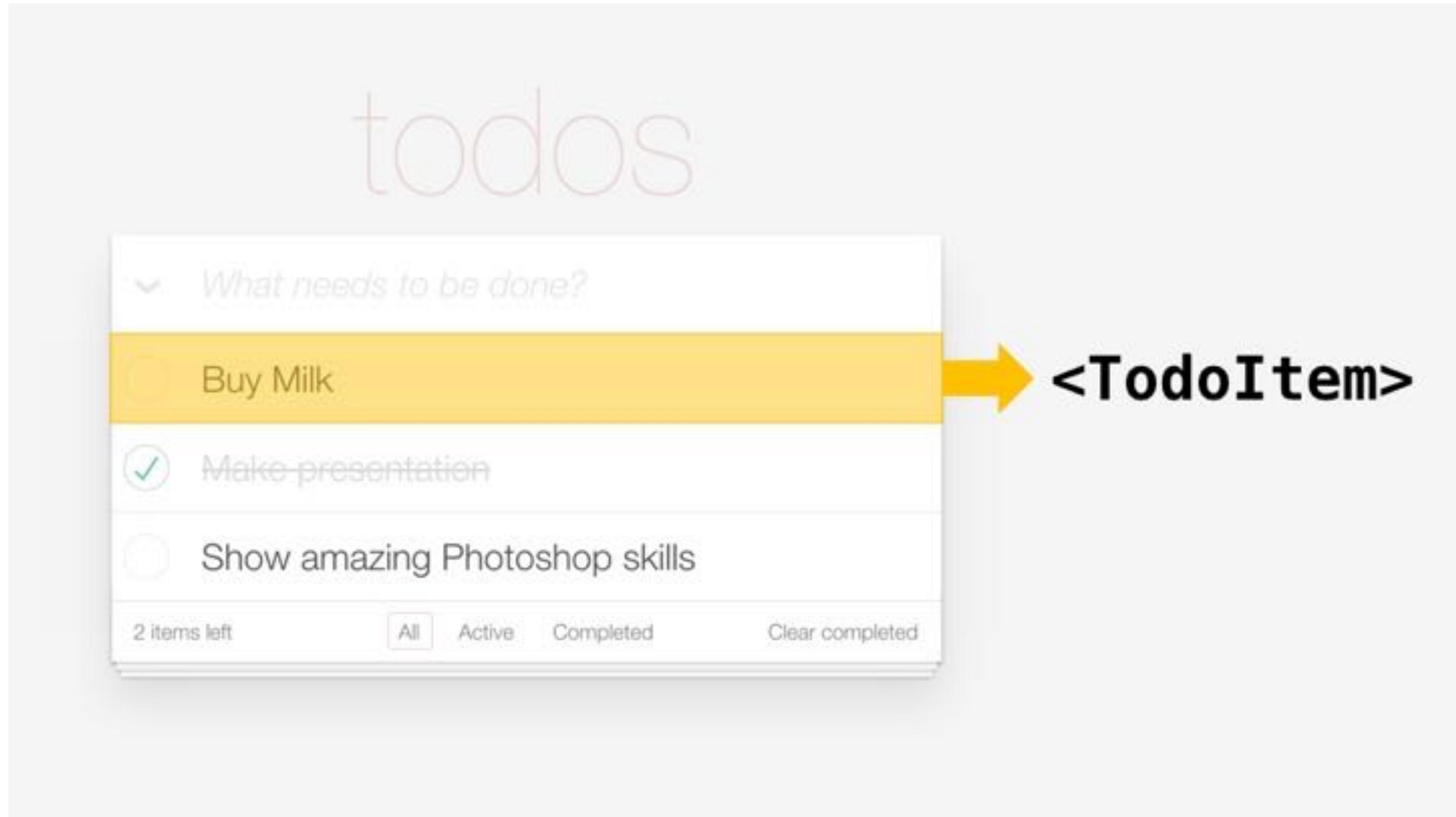
EXAMPLE

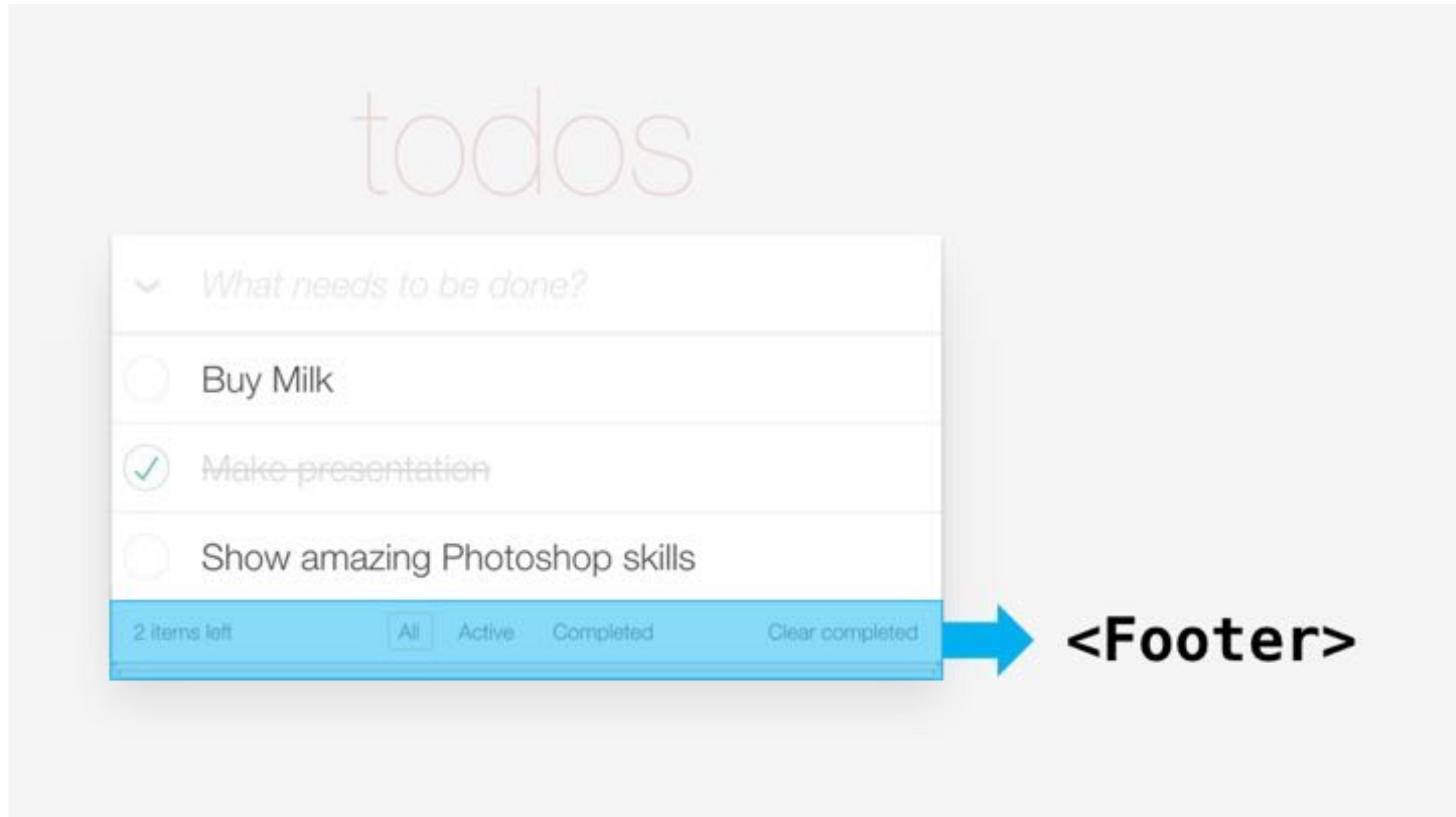
JSX











PROPS & STATE

- **Props are a component's configuration. They are received from above and immutable as far as the component receiving them is concerned.**
- **State starts with a default value defined in the component's constructor and then suffers from mutations in time (mostly generated from user events). A component manages its own state internally.**
- **Every time the props or state changes, the component re-renders.**