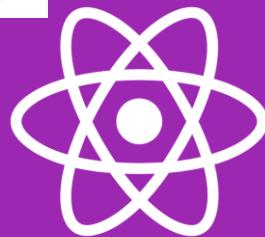


JS



+



MANAGE STATE

Redux

Christina Kayastha
Senior Software Engineer
Vistaprint, Cimpress
@christikaes

HELLO WORLD!

My name is Christina (:

Christina Kayastha

Senior Software Engineer

Vistaprint, Boston, MA

@christikaes

I'm all about:

- Icecream
- Community Events
- Bleeding Edge Technology



1

USER INTERFACE



2

EXPERIENCE DESIGN



3

VIRTUAL/AUGMENTED REALITY



4

BUILD FOR MOBILE



5

MANAGE STATE

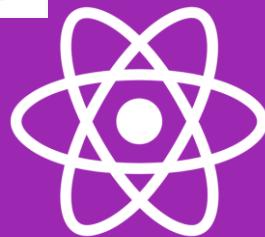


6

AUTH & DATABASE



JS



+



MANAGE STATE

Redux

Christina Kayastha
Senior Software Engineer
Vistaprint, Cimpress
@christikaes



Redux

WHAT is it ?

HOW can I use it ?

WHY should I care ?



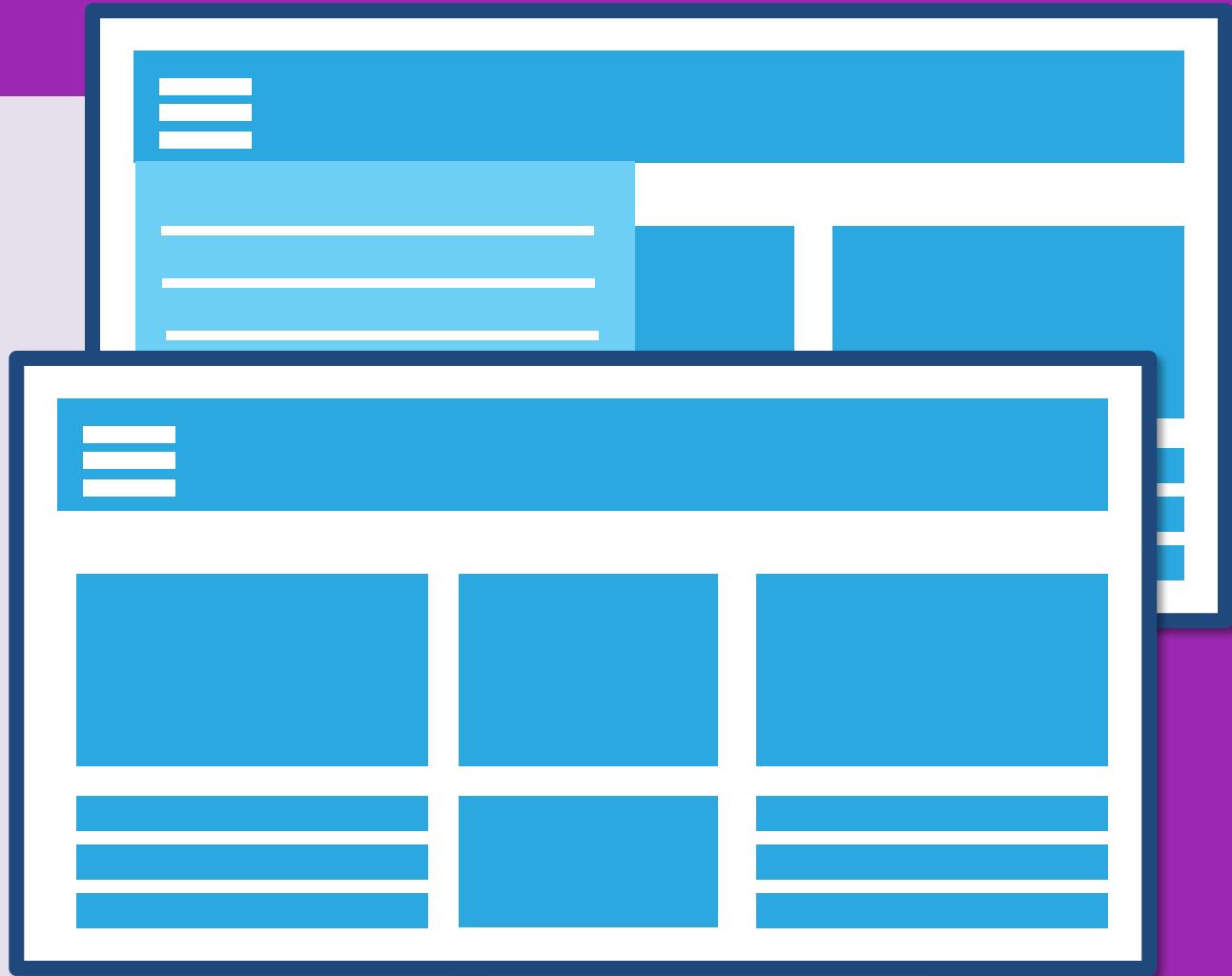
What is
STATE ?



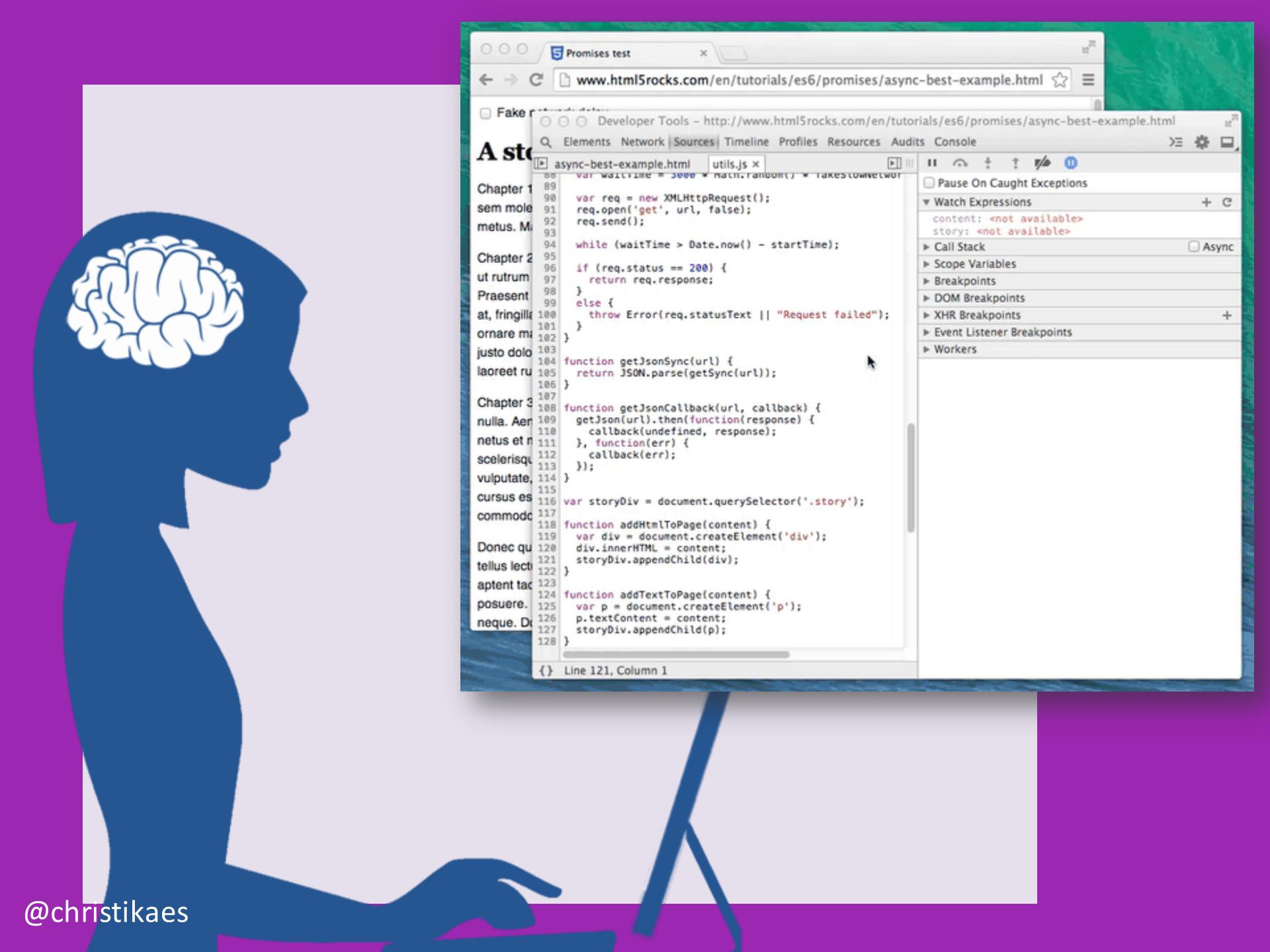
State is what an application knows about the user, interactions, and other pieces of global information



Why should
I care about
MANAGING STATE ?



@christikaes



A screenshot of a web browser window titled "Promises test" showing a developer tools interface. The main content area displays a portion of the file "utils.js" from the URL "www.html5rocks.com/en/tutorials/es6/promises/async-best-example.html". The code is a mix of placeholder text ("Chapter 1", "Chapter 2", "Chapter 3") and actual JavaScript. The developer tools sidebar on the right shows the "Sources" tab selected, with the "Breakpoints" section expanded. A mouse cursor is visible over the code editor.

```
var waitTime = 3000 * Math.random() * Math.random();
```

```
Chapter 1
90 var req = new XMLHttpRequest();
91 req.open('get', url, false);
92 req.send();
93
94 while (waitTime > Date.now() - startTime);
```

```
Chapter 2
95 if (req.status == 200) {
96   return req.response;
97 } else {
98   throw Error(req.statusText || "Request failed");
99 }
100 }
101 
```

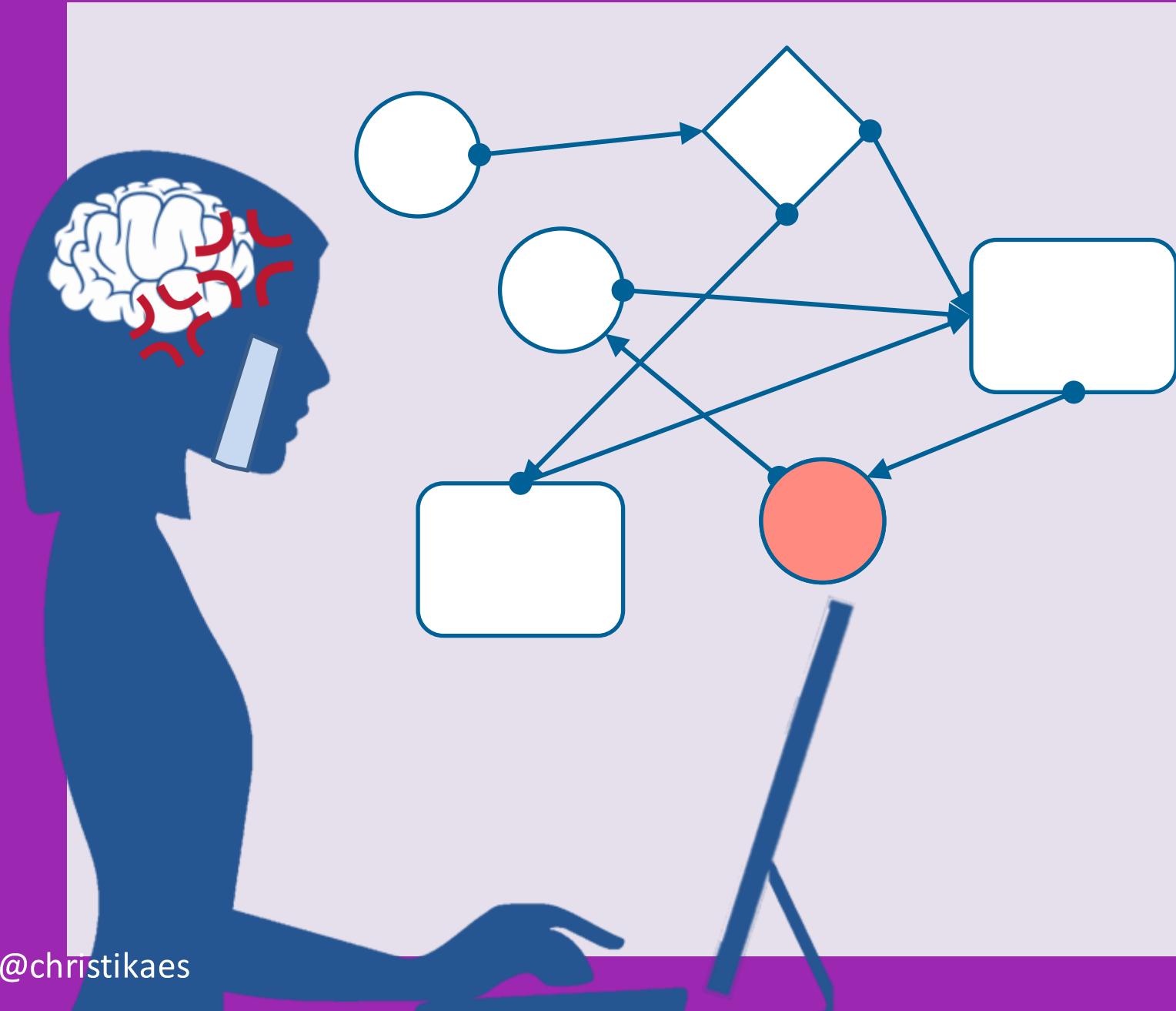
```
justo dolor laoreet ru
102 }
103 function getJsonSync(url) {
104   return JSON.parse(getSync(url));
105 }
106 
```

```
Chapter 3
107 function getJsonCallback(url, callback) {
108   getJson(url).then(function(response) {
109     callback(undefined, response);
110   }, function(err) {
111     callback(err);
112   });
113 }
114 
```

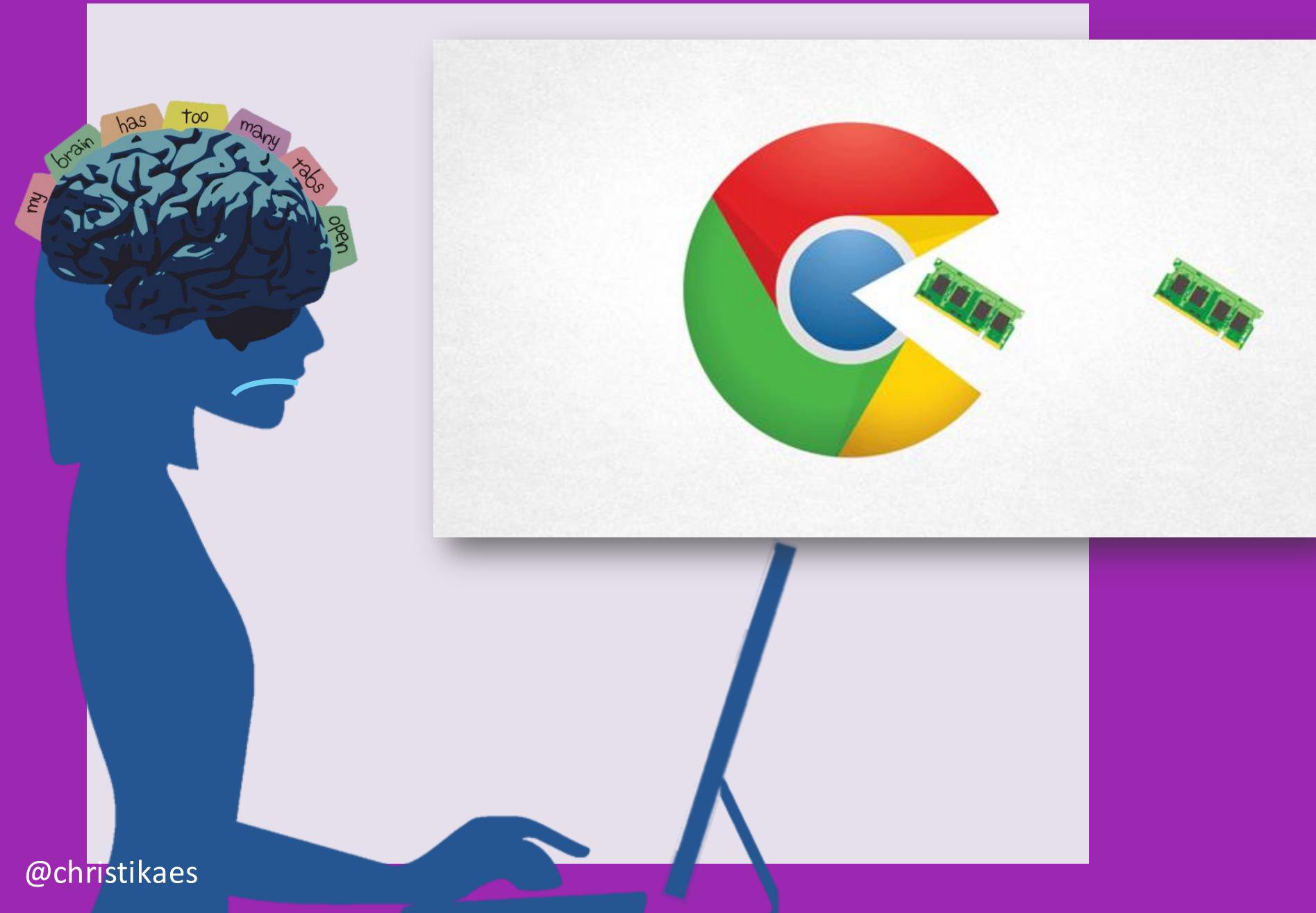
```
cursus es
115 commod
116 var storyDiv = document.querySelector('.story');
```

```
Donec qu
117 function addHtmlToPage(content) {
118   var div = document.createElement('div');
119   div.innerHTML = content;
120   storyDiv.appendChild(div);
121 }
122 
```

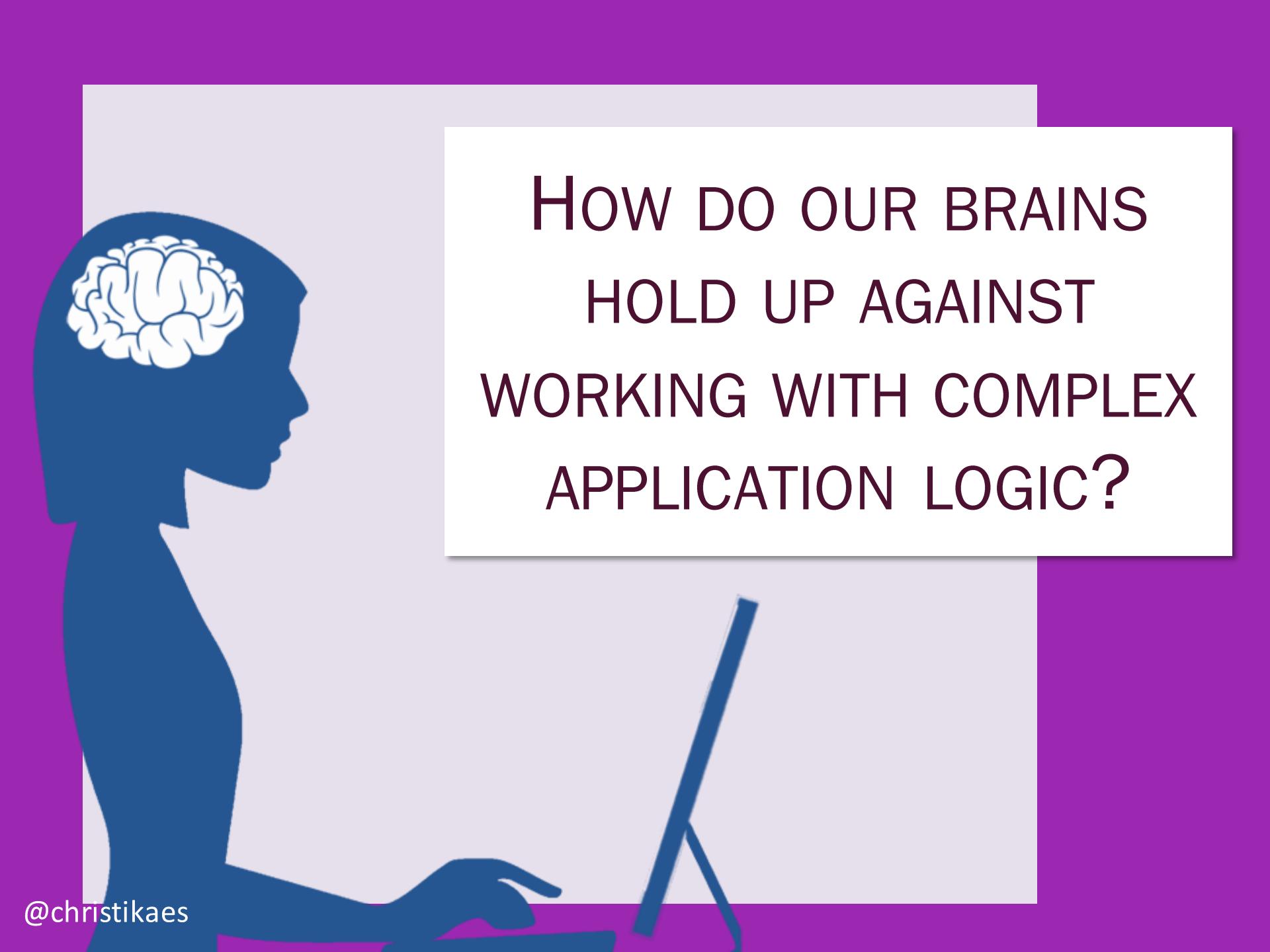
```
tellus lect
123 function addTextToPage(content) {
124   var p = document.createElement('p');
125   p.textContent = content;
126   storyDiv.appendChild(p);
127 }
128 }
```



@christikaes



@christikaes

A large silhouette of a person's head and shoulders is positioned on the left side of the slide. The head is turned slightly to the right, showing a white brain inside. The person is sitting at a desk, with their hands visible on a keyboard. A computer monitor is on the desk, displaying a white slide with the main text.

How do our brains hold up against working with complex application logic?

STAND BACK



I'M GOING TO TRY SCIENCE

WHAT PARTS OF THE BRAIN DO PROGRAMMERS ACTIVATE?

Understanding Understanding Source Code with Functional Magnetic Resonance Imaging

Janet Siegmund^π, Christian Kästner^ω, Sven Apel^π, Chris Parnin^β, Anja Bethmann^θ,
Thomas Leich^δ, Gunter Saake^σ, and André Brechmann^θ

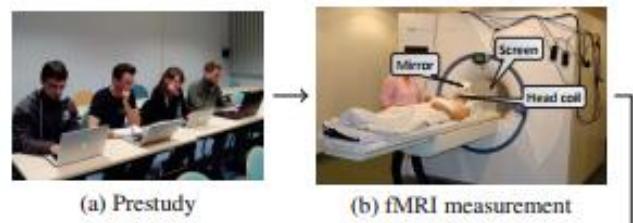
^πUniversity of Passau, Germany ^ωCarnegie Mellon University, USA

^βGeorgia Institute of Technology, USA ^θLeibniz Inst. for Neurobiology Magdeburg, Germany

^δMetop Research Institute, Magdeburg, Germany ^σUniversity of Magdeburg, Germany

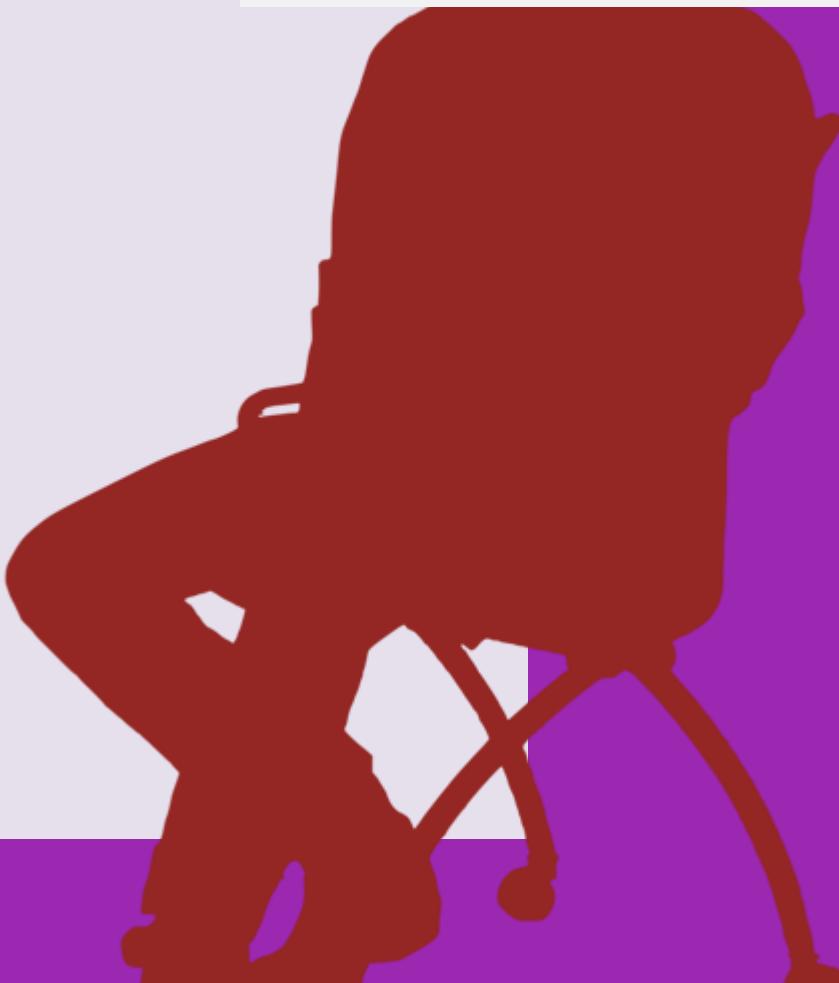
ABSTRACT

Program comprehension is an important cognitive process that inherently eludes direct measurement. Thus, researchers are struggling with providing suitable programming languages, tools, or coding conventions to support developers in their everyday work. In this paper, we explore whether *functional magnetic resonance imaging (fMRI)*, which is well established in cognitive neuroscience, is feasible to soundly measure program comprehension. In a controlled experiment, we observed 17 participants inside an fMRI scanner while they were comprehending short source-code snippets, which we contrasted with locating syntax errors. We found a clear, distinct activation pattern of five brain regions, which are related to working memory, attention, and language processing—all processes that fit well to our understanding of program comprehension. Our results encourage us and, hopefully, other researchers to use fMRI in future studies to measure program comprehension and, in particular, to compare it to other cognitive processes.

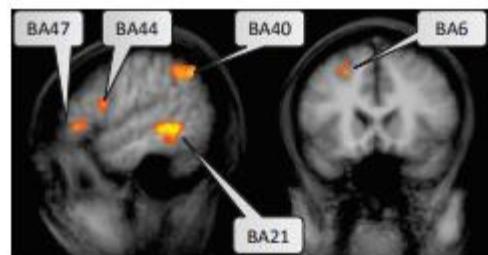
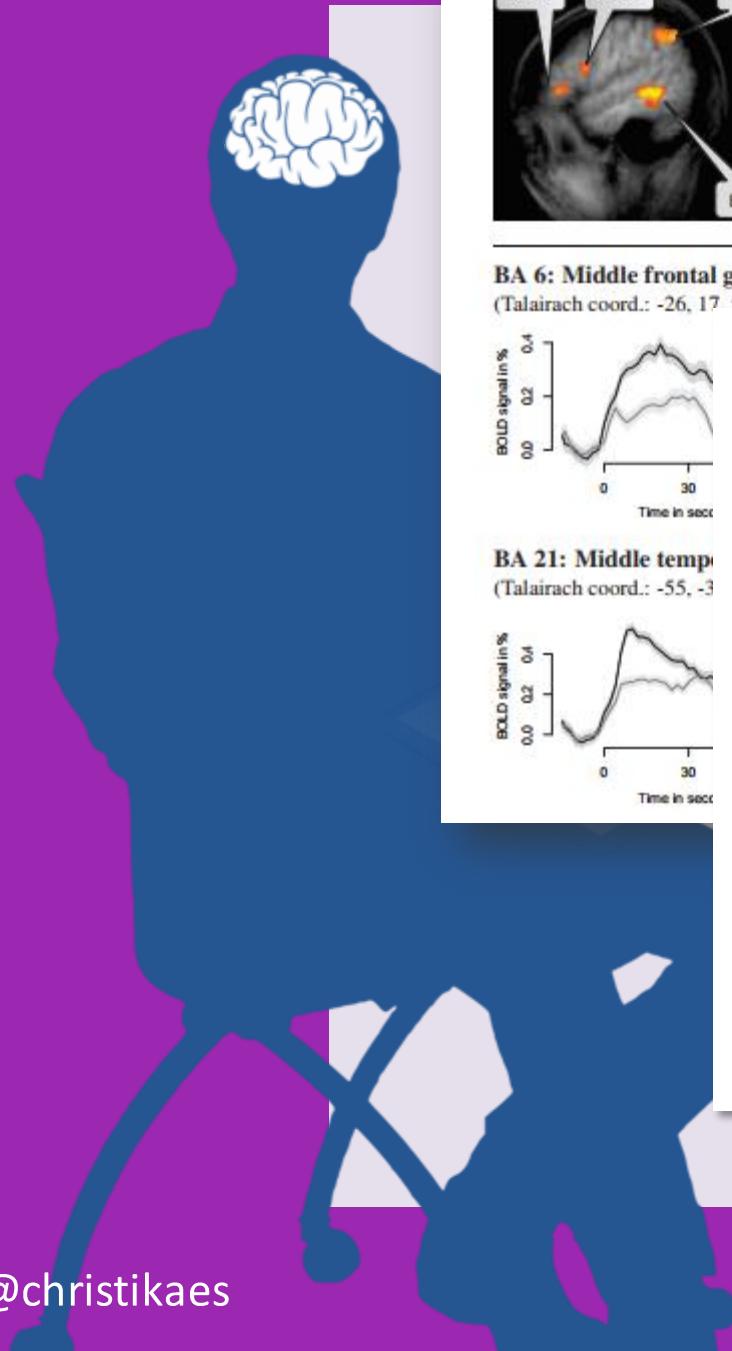




@christikaes

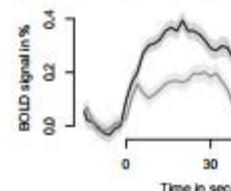


@christikaes



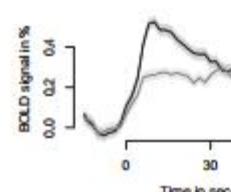
BA 6: Middle frontal gyrus

(Talairach coord.: -26, 17, 52; cluster size: 1270)



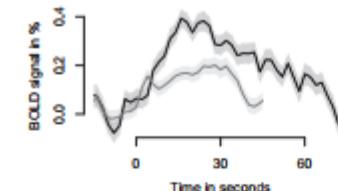
BA 21: Middle tempo

(Talairach coord.: -55, -3)



BA 40: Inferior parietal lobule

(Talairach coord.: -51, -49, 41; cluster size: 3368)



Working memory
Verbal/numeric
Problem solving

BA 44: Inferior frontal gyrus

(Talairach coord.: -50, 11, 16; cluster size: 698)

ATTENTION WORKING MEMORY SEMANTIC MEMORY LANGUAGE

WHEN DO WE USE THE MOST ATTENTION/WORKING MEMORY?

CHI 2004 | Late Breaking Results Paper

24-29 April | Vienna, Austria

2011 19th IEEE International Conference on Program Comprehension

CHI 2007 Proceedings • Tasks

April 28-May 3, 2007 • San Jose, CA, USA

Understanding and Developing Models for Detecting and Differentiating Breakpoints during Interactive Tasks

Shamsi T. Iqbal and Brian P. Bailey

Department of Computer Science

University of Illinois

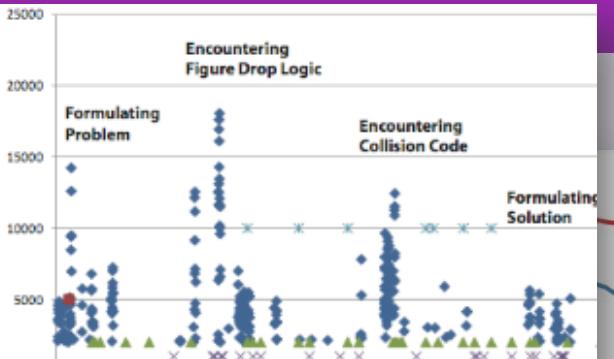
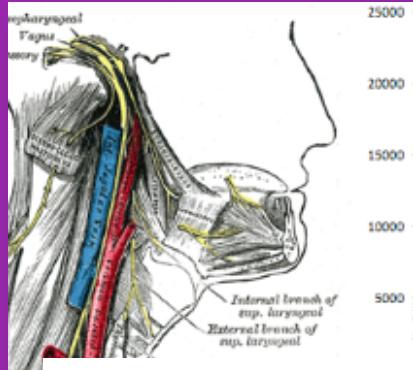
Urbana, IL 61801 USA

{siqbal, bpbailey}@cs.uiuc.edu

ABSTRACT

The ability to detect and differentiate breakpoints during task execution is critical for enabling defer-to-breakpoint policies within interruption management. In this work, we examine the feasibility of building statistical models that can detect and differentiate three granularities (types) of perceptually meaningful breakpoints during task execution, without having to recognize the underlying tasks. We collected ecological samples of task execution data, and

One common method for detecting breakpoints is to match users' ongoing interaction to specifications of tasks defined a priori [4]. Although this allows breakpoints to be easily detected within tasks that are fairly prescribed, it is much more difficult to leverage these types of static specifications to detect breakpoints within tasks that have highly variable interaction, i.e., *free-form* tasks, yet these are by far the most common type of computing task performed [8]. This



COMPREHENDING DATA FLOW AND CONTROL FLOW IN CODE



COMPLEX APPLICATION LOGIC IS TAXING ON OUR HUMAN BRAINS!



WE LIVE IN A WORLD OF
INCREASINGLY COMPLEX
FRONTEND APPLICATIONS



HOW CAN WE MANAGE INCREASINGLY COMPLEX FRONTEND APPLICATIONS



REDUX TO MANAGE INCREASINGLY COMPLEX FRONTEND APPLICATIONS





Managing State is makes
our app much easier to
Learn and Debug!



What does
REDUX
mean anyway?

Redux?

Brought back, revived

ReactFlux



Redux is a predictable state container for web applications



HOW do I use Redux?

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

`github.com/christinakayastha
/StudioReduxExample`

HELLO WORLD⁺
LET'S GET STARTED!

**REDUX IS A
PREDICTABLE
STATE CONTAINER
FOR JAVASCRIPT APPS**

`(state, action) => newState`

ACTION

REDUCER

STORE



`(state, action) => newState`

ACTION

REDUCER

STORE

ALL UPDATES TO THE APPLICATION
STATE ARE CAUSED BY **ACTIONS**

JS OBJECT THAT DESCRIBES WHAT HAPPENED?

```
{  
  type: "ADD_TODO",  
  payload: {  
    text: "Learn Redux!"  
  }  
}
```

BEST PRACTICE: USE ACTION CREATORS TO GENERATE ACTIONS

```
addTodoAction = () =>  
  return {  
    type: "ADD_TODO",  
    payload: {  
      text: "Learn Redux!"  
    }  
  }
```

`(state, action) => newState`

ACTION

REDUCER

STORE

ALL UPDATES TO THE APPLICATION
STATE ARE CAUSED BY **ACTIONS**

`(state, action) => newState`

ACTION

REDUCER

STORE

THE **REDUCER** TAKES THE
CURRENT STATE AND AN ACTION,
AND RETURNS A NEW STATE

DESIGN THE STATE SHAPE

```
state = [  
  {  
    text: "Learn Redux"  
  }  
  ...  
]
```

YOUR ENTIRE STATE IN 1 JS BLOB

A FUNCTION THAT DESCRIBES: WHAT'S THE NEW STATE?

```
reducer = (state, action) => {  
  switch(action.type) {  
    case "ADD_TODO":  
      return [  
        ...state,  
        action.payload  
      ];  
    default: return state;  
  }  
}
```

BEST PRACTICE: USE COMBINEREDUCERS

```
reducer = combineReducers({  
  reducer1,  
  reducer2,  
  ...  
})
```

`(state, action) => newState`

ACTION

REDUCER

STORE

THE **REDUCER** TAKES THE
CURRENT STATE AND AN ACTION,
AND RETURNS A NEW STATE

`(state, action) => newState`

ACTION

REDUCER

STORE

THE **STORE** HOLDS THE STATE OF THE APPLICATION, IT CALLS THE REDUCER WHEN ACTIONS ARE DISPATCHED

CREATE A STORE WITH A REDUCER

```
store = Redux.createStore(reducer)
```

GET STATE THROUGH STORE

```
store.getState()
```

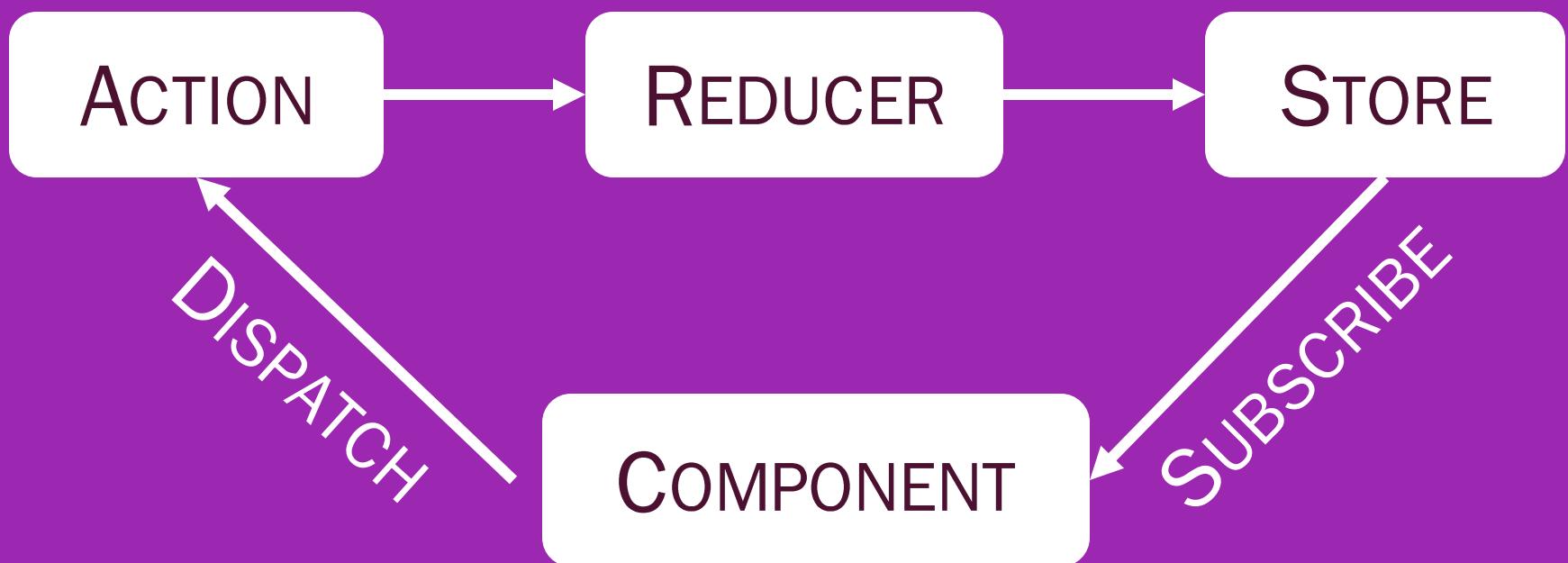
DISPATCH ACTIONS THROUGH STORE

```
store.dispatch(addTodoAction())
```

SUBSCRIBE TO CHANGES ON STORE

```
store.subscribe(() => {...})
```

$(\text{state}, \text{ action}) \Rightarrow \text{newState}$



HELLO WORLD⁺

LET'S GET STARTED!

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

The screenshot shows a browser window with the title "Redux TodoMVC example" and the URL "zalmoxisus.github.io/redux-devtools-extension/examples/todomvc/". The page content is about "1. With Redux" and "1.1 Basic store". It includes a code snippet for creating a Redux store and a note about the optional `preloadedState` argument. Below the code, there's a section about ESLint configuration. On the left, the Redux DevTools extension is visible, showing a state tree with nodes like "state", "todos", and "ADD_TODO". The "Todos" node has a value of 1, and the "ADD_TODO" action has a "text" field of "Add extension". The bottom part of the screenshot shows the browser's address bar and some UI elements.

1. With Redux

1.1 Basic store

For a basic Redux store simply add:

```
const store = createStore(  
  reducer, /* preloadedState, */  
  + window.__REDUX_DEVTOOLS_EXTENSION__ && window.__REDUX_DEVTOOLS_EXTENSION__()  
);
```

Note that `preloadedState` argument is optional in Redux' `createStore`.

For universal ("isomorphic") apps, prefix it with `typeof window !== 'undefined' &&`.

In case ESLint is configured to not allow using the underscore dangle, wrap it like so:

```
+ /* eslint-disable no-underscore-dangle */  
const store = createStore(  
  reducer, /* preloadedState, */  
  window.__REDUX_DEVTOOLS_EXTENSION__ && window.__REDUX_DEVTOOLS_EXTENSION__()  
);
```

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

UNDO/REDO REDUCER ENHancers!

A REDUCER ENHANCER IS
SIMPLY A
HIGHER ORDER REDUCER

redux undo/redo

npm v0.6.1 build passing dependencies up to date code style standard tips \$0.00/week

simple undo/redo functionality for redux state containers

Clicked: 1 times    

Protip: You can use the [redux-undo-boilerplate](#) to quickly get started with redux-undo.

Note: Make sure to update your programs to the [latest History API](#).

Installation

```
npm install --save redux-undo
```

UNDO/REDO REDUCER ENHancers!

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

SYNCING STATE REDUX MIDDLEWARE!

MIDDLEWARE IS AN
EXTENSION POINT
BETWEEN THE
ACTION DISPATCH AND
THE REDUCER

redux-localstorage-simple public

Save and load Redux state to and from LocalStorage. Supports Immutable.js data structures.

Installation

```
npm install --save redux-localstorage-simple
```

Usage Example (ES6 code)

```
import { applyMiddleware, createStore } from "redux"
import reducer from "./reducer"

// Import the necessary methods for saving and loading
import { save, load } from "redux-localstorage-simple"
```

SYNCING STATE REDUX MIDDLEWARE!

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

HELLO REAL WORLD
LESSONS LEARNED

REDUX (WITH REACT) AT VISTAPRINT

Search Design

Enter a keyword

See your info
all designs

Personalize it

Industry ▾

Art & Entertainment

Automotive & Tra
(10)

Beauty & Spa (11)

Business Services

Construction, Rep
Improvement (62)

Finance & Insuran

Health & Social Si

Law, Public Safet
(18)

Retail & Sales (14)

Using Your Photo
(20)

+ Show More

Personal & Family

Styles & Themes ▾

- + NO MORE COMPLEX PROPS TREE
- + SEPARATION OF CONCERNS
 - ADDING REDUX WAS A REWRITE
- + DEBUGGING AND ONBOARDING
 - DEBUGGING GENERATED STATE
 - LEARNING CURVE
- + UNIT TESTING/TDD

REDUX (WITH ANGULAR) AT PARKABLER

- + LOGIC IS CONSOLIDATED
 - ADDING REDUX WAS A REWRITE
- + SEPARATION OF CONCERNS
- + RAPID ITERATION
- + UNIT TESTING/TDD
- + DEBUGGING
- + DATA SYNC

HELLO REAL WORLD
LESSONS LEARNED

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?

HELLO WORLD

REDUX DEV TOOLS

UNDO/REDO

SYNCING STATES

HELLO REAL WORLD

WHAT'S NEXT?



To use Redux, just remember the 3 parts:

- 1) Store:** the entire state of your app
- 2) Action:** description of what happened
- 3) Reducer:** a function that creates a new modified store based on the action



WHAT is Angular Redux?



Angular Redux is an npm
package that provides a
series of Redux bindings
for Angular



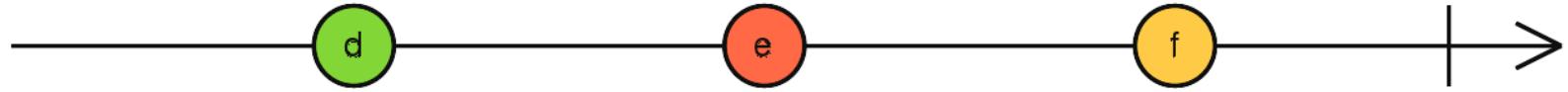
HOW do I use Angular Redux?

QUICK REVIEW

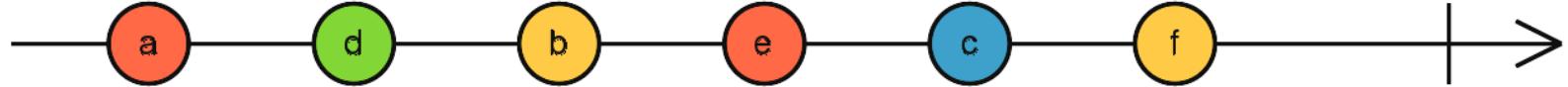
Observables

The Observable object represents a push based collection

The Observer and Observable interfaces provide a generalized mechanism for push-based notification, also known as the observer design pattern. The Observable object represents the object that sends notifications (the provider); the Observer object represents the class that receives them (the observer).



merge



```
1 const sourceOne = Rx.Observable.create(observer => {
2   observer.onNext(1);
3   observer.onNext(2);
4   observer.onNext(3);
5 })
6 sourceOne.subscribe(val => console.log('SourceOne:', val));
7

8
9 const sourceTwo = Rx.Observable.interval(2000);
10 sourceTwo.subscribe(val => console.log('SourceTwo:', val));
```

```
1▼ const {Component} = ng.core;
2▼ const {bootstrap} = ng.platform.browser;
3
4▼ @Component({
5    selector: 'my-app',
6    template: `
7        <section>
8            <h1>{{number$ | async}}</h1>
9        </section>
10    `
11})
12▼ class AppComponent {
13▼     constructor(){
14        this.number$ = Rx.Observable.interval(1000);
15    }
16}
17
18bootstrap(AppComponent);
19
```



HOW do I use Angular Redux?

@angular-redux/store

Angular bindings for [Redux](#).

For Angular 1 see [ng-redux](#)

[chat](#) [on gitter](#) [build](#) [passing](#) [npm](#) [v6.5.7](#) [downloads](#) [29k/month](#)

What is Redux?

Redux is a popular approach to managing state in applications. It emphasises:

- A single, immutable data store.
- One-way data flow.
- An approach to change based on pure functions and a stream of actions.

You can find lots of excellent documentation here: [Redux](#).

What is @angular-redux?

We provide a set of npm packages that help you integrate your redux store into your Angular 2+ application. This helps you by bridging the gap with some of Angular's advanced features, including:



To use Angular Redux:

- 1) npm install @angular-redux/store
- 2) Import/Setup your app
- 3) Store, Action, Reducer



HOW do I use React Redux?

React Redux

Official React bindings for [Redux](#).

Performant and flexible.

[build](#) [passing](#) [npm](#) [v5.0.6](#) [downloads](#) [3M/month](#) [slack](#) [redux@reactiflux](#)

Installation

React Redux requires [React 0.14 or later](#).

```
npm install --save react-redux
```

This assumes that you're using [npm](#) package manager with a module bundler like [Webpack](#) or [Browserify](#) to consume [CommonJS modules](#).

If you don't yet use [npm](#) or a modern module bundler, and would rather prefer a single-file [UMD](#) build that makes [ReactRedux](#) available as a global object, you can grab a pre-built version from [cdnjs](#). We *don't* recommend this approach for any serious application, as most of the libraries complementary to Redux are only available on [npm](#).

React Native

As of React Native 0.18, React Redux 5.x should work with React Native. If you have any issues with React Redux 5.x on React Native, run `npm ls react` and make sure you don't have a duplicate React installation in your `node_modules`. We recommend that you use `npm@3.x` which is better at avoiding these kinds of issues.

If you are on an older version of React Native, you'll need to keep using [React Redux 3.x branch and documentation](#) because of [this problem](#).

Documentation



To use React Redux:

- 1) npm install react-redux
- 2) Import/Setup your app
- 3) Store, Action, Reducer



WHY should I use Redux?



Redux is awesome because:

- Redux to manage your state
- Redux tools work out of the box
- Works with RxJS Observable
- ...



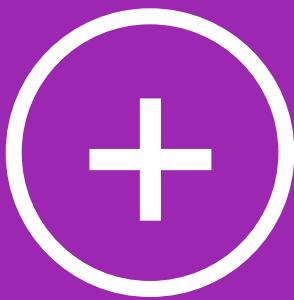
Redux

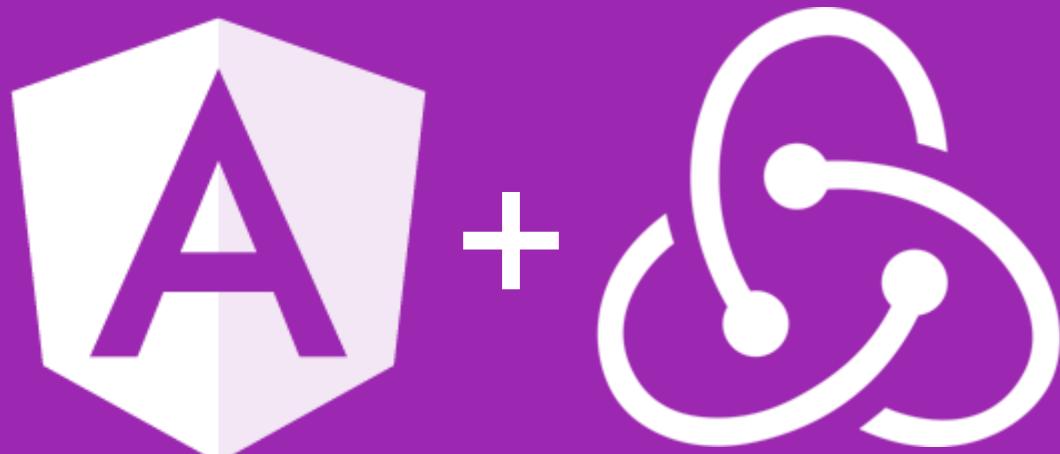
WHAT is it ?

HOW can I use it ?

WHY should I care ?

THANK YOU!





MANAGE STATE

Angular Redux

Christina Kayastha
Senior Software Engineer
Vistaprint, Cimpress
@christikaes