SIDE-EFFECTS

RESULTS MAY VARY

Yulia Startsev (@ioctapteb)

Partial::Conf 2017

moz://a

Internet for people, not profit —

Mozilla Corporation (US) | https://www.mozilla.org/en-US/

z://a Firefox Pocket Internet Health Technology About Us Get Involved Download Fir

Console Debugger { } Style Editor Performance Memory Network Storage

⌘+P to search

ExtensionContent.jsm × common-bundle.bc3cd58a7ff2.js

```
1  /* This Source Code Form is subject to the terms of the Mozilla Public
2   * License, v. 2.0. If a copy of the MPL was not distributed with this
3   * file, You can obtain one at http://mozilla.org/MPL/2.0/. */
4
5  "use strict";
6
7  this.EXPORTED_SYMBOLS = ["ExtensionContent"];
8
9  /* globals ExtensionContent */
10
11 const {classes: Cc, interfaces: Ci, utils: Cu, results: Cr} = Components;
12
13 Cu.import("resource://gre/modules/Services.jsm");
14 Cu.import("resource://gre/modules/XPCOMUtils.jsm");
15
16 XPCOMUtils.defineLazyModuleGetters(this, {
17   LanguageDetector: "resource:///modules/translation/LanguageDetector.jsm",
18   MessageChannel: "resource://gre/modules/MessageChannel.jsm",
19   Schemas: "resource://gre/modules/Schemas.jsm",
20   TelemetryStopwatch: "resource://gre/modules/TelemetryStopwatch.jsm",
21   WebNavigationFrames: "resource://gre/modules/WebNavigationFrames.jsm",
22 });
23
24 XPCOMUtils.defineLazyServiceGetter(this, "styleSheetService",
25                                    "@mozilla.org/content/style-sheet-service;1",
```

lytics.com
anager.com

Outline View

{ }

Watch Expressions
Add Watch Ex
Breakpoints
No Breakp
Call Stack
Not Paus
Scopes
Not Paus

Backbone.js • TodoMVC

x-dev.tools/debugger-exan

Search

todos

What needs to be done?

Double-click to edit a todo

Written by Addy Osmani

Part of TodoMVC

Debugger

localhost:8000/?firefox-tab=child1/tab1

Search

⌘ P to search

Sources | Outline

- firefox-dev.tools
  - debugger-examples/examples/todor
    - bower_components
    - js
      - collections
      - models
        - todo.js
      - routers
      - views
      - app.js

todo.js × | todos.js

```
1  /*global Backbone */
2  var app = app || {};
3
4  (function () {
5      'use strict';
6
7      // Todo Model
8      // ----------
9
10     // Our basic **Todo** model has `title`, `order`,
11     app.Todo = Backbone.Model.extend({
12         // Default attributes for the todo
13         // and ensure that each todo created has `tit
14         defaults: {
15             title: '',
16             completed: false
17         },
18
19         To    e    `compl    d` state of this todo
20         gl    fu    ion ()
21             this.s    ({
22                 co    ed: !    s.get('completed')
23             });
24
25     });
26 })();
27
```

Watch Expressions

Add Watch Expre

Breakpoints

No Breakpoin

Call Stack

Scopes

Event Listeners

TARGET

Debugger

localhost:8000/?firefox-tab=child1/tab1

⌘ P to search

todo.js × todos.js +

Sources | Outline

- firefox-dev.tools
  - debugger-examples/examples/todor
    - bower_components
    - js
      - collections
      - models
        - todo.js
      - routers
      - views
      - app.js

```
1  /*global Backbone */
2  var app = app || {};
3
4  (function () {
5      'use strict';
6
7      // Todo Model
8      // ----------
9
10     // Our basic **Todo** mode
11     app.Todo = Backbone.Model.
12         // Default attributes
13         // and ensure that eac
14         defaults: {
15             title: '',
16             completed: false
17         },
18
19         // Toggle the `complet
20         toggle: function () {
21             this.save({
22                 completed: !th
23             });
24         }
25     });
26 })();
27
```

{ } C

Debugger

localhost:8000/?firefox-tab=child1/tab1

⌘ P to search

backbone.js ×

Sources | Outline

- firefox-dev.tools
  - debugger-examples/examples/todon
    - bower_components
      - backbone
        - backbone.js
      - backbone.localStorage
      - jquery
      - underscore
    - js

```
1  //     Backbone.js 1.1.2
2
3  //     (c) 2010-2014 Jeremy Ashkenas, DocumentCloud and In
4  //     Backbone may be freely distributed under the MIT li
5  //     For all details and documentation:
6  //     http://backbonejs.org
7
8  (function(root, factory) {
9
10   // Set up Backbone appropriately for the environment. St
11   if (typeof define === 'function' && define.amd) {
12     define(['underscore', 'jquery', 'exports'], function(_
13       // Export global even in AMD case in case this scrip
14       // others that may still expect a global Backbone.
15       root.Backbone = factory(root, exports, _, $);
16     });
17
18   // Next for Node.js or CommonJS. jQuery may not be neede
19   } else if (typeof exports !== 'undefined') {
20     var _ = require('underscore');
21     factory(root, exports, _);
22
23   // Finally, as a browser global.
24   } else {
25     root.Backbone = factory(root, {}, root._, (root.jQuery
26   }
27
28 }(this, function(root, Backbone, _, $) {
29
30   // Initial Setup
31   // -------------
32
33   // Save the previous value of the `Backbone` variable, s
34   // restored later on, if `noConflict` is used.
35   var previousBackbone = root.Backbone;
36
37   // Create local references to array methods we'll want t
38   var array = [];
39   var push = array.push;
40   var slice = array.slice;
41   var splice = array.splice;
42
43   // Current version of the library. Keep in sync with `pa
44   Backbone.VERSION = '1.1.2';
45
46   // For Backbone's purposes, jQuery, Zepto, Ender, or My
47   // the `$` variable.
48   Backbone.$ = $;
49
50   // Runs Backbone.js in *noConflict* mode, returning the
51   // to its previous owner. Returns a reference to this Ba
52   Backbone.noConflict = function() {
53     root.Backbone = previousBackbone;
```

{ } C

Debugger

localhost:8000/?firefox-tab=child1/tab1

⌘ P to search

Sources | Outline

- firefox-dev.tools
  - debugger-examples/examples/todor
    - bower_components
    - js
      - collections
      - models
        - todo.js
      - routers
      - views
      - app.js

todo.js | todos.js

```
/*global Backbone */
var app = app || {};

(function () {
    'use strict';

    // Todo Model
    // ----------

    // Our basic **Todo** mode
    app.Todo = Backbone.Model.
        // Default attributes
        // and ensure that eac
        defaults: {
            title: '',
            completed: false
        },

        // Toggle the `complet
        toggle: function () {
            this.save({
                completed: !th
            });
        }
    });
})();
```

{ } C

Debugger

localhost:8000/?firefox-tab=child1/tab1

⌘ P to search

Sources | Outline

- firefox-dev.tools
  - debugger-examples/examples/todon
    - bower_components
      - backbone
        - backbone.js
      - backbone.localStorage
      - jquery
      - underscore
    - js

backbone.js

```
//     Backbone.js 1.1.2

//     (c) 2010-2014 Jeremy Ashkenas, DocumentCloud and In
//     Backbone may be freely distributed under the MIT li
//     For all details and documentation:
//     http://backbonejs.org

(function(root, factory) {

  // Set up Backbone appropriately for the environment. Sta
  if (typeof define === 'function' && define.amd) {
    define(['underscore', 'jquery', 'exports'], function(_
      // Export global even in AMD case in case this script
      // others that may still expect a global Backbone.
      root.Backbone = factory(root, exports, _, $);
    });

  // Next for Node.js or CommonJS. jQuery may not be needed
  } else if (typeof exports !== 'undefined') {
    var _ = require('underscore');
```

{ } C

Inspector | Console | Debugger | { } Style Editor | Performance | Memory | Netwo

⌘ P to search

- localhost:8000
  - assets/build
- webpack://

Sources | Outline

debugger.js

```
(function webpackUniversalModuleDefinition(root, factory)
  if(typeof exports === 'object' && typeof module === 'obj
    module.exports = factory();
  else if(typeof define === 'function' && define.amd)
    define([], factory);
  else {
    var a = factory();
    for(var i in a) (typeof exports === 'object' ? exports
  }
})(this, function() {
return /******/ (function(modules) { // webpackBootstrap
/******/  // The module cache
/******/  var installedModules = {};
/******/
/******/  // The require function
/******/  function __webpack_require__(moduleId) {
/******/
/******/    // Check if module is in cache
/******/    if(installedModules[moduleId]) {
/******/      return installedModules[moduleId].exports;
/******/    }
/******/    // Create a new module (and put it into the ca
/******/    var module = installedModules[moduleId] = {
/******/      i: moduleId,
/******/      l: false,
/******/      exports: {}
/******/    };
/******/
```

Debugger

localhost:8000/?firefox-tab=child1/tab1

⌘ P to search

Sources | Outline

- firefox-dev.tools
  - debugger-examples/examples/todor
    - bower_components
    - js
      - collections
      - models
        - todo.js
      - routers
      - views
    - app.js

todo.js | todos.js

```
1  /*global Backbone */
2  var app = app || {};
3
4  (function () {
5      'use strict';
6
7      // Todo Model
8      // ----------
9
10     // Our basic **Todo** mode
11     app.Todo = Backbone.Model.
12         // Default attributes
```

Debugger

localhost:8000/?firefox-tab=child1/tab1

⌘ P to search

Sources | Outline

- firefox-dev.tools
  - debugger-examples/examples/todon
    - bower_components
      - backbone
        - backbone.js
      - backbone.localStorage
      - jquery

backbone.js

```
1  //     Backbone.js 1.1.2
2
3  //     (c) 2010-2014 Jeremy Ashkenas, DocumentCloud and Inv
4  //     Backbone may be freely distributed under the MIT lic
5  //     For all details and documentation:
6  //     http://backbonejs.org
7
8  (function(root, factory) {
9
10   // Set up Backbone appropriately for the environment. Sta
11   if (typeof define === 'function' && define.amd) {
12     define(['underscore', 'jquery', 'exports'], function(_
13       // Export global even in AMD case in case this script
```

Developer Tools - undefined

Console | Debugger | Memory

⌘ P to search

- chrome://
  - content
- jar://
- localhost:8000
- resource://
- webpack://

Sources | Outline

debugger.js

```
1  (function webpackUniversalModuleDefinition(ro
2    if(typeof exports === 'object' && typeof mo
3      module.exports = factory();
4    else if(typeof define === 'function' && def
5      define([], factory);
6    else {
7      var a = factory();
8      for(var i in a) (typeof exports === 'obje
9    }
10 })(this, function() {
11 return /******/ (function(modules) { // webpa
12 /******/  // The module cache
13 /******/  var installedModules = {};
14 /******/
15 /******/  // The require function
16 /******/  function __webpack_require__(module
17 /******/
```

Watch Expressions

Add Watch Expression

Breakpoints

No Breakpoints

Call Stack

Scopes

Memory | Netwo

Sources | Outline

```
11 return /******/ (function(modules) { // webpackBootstrap
12 /******/  // The module cache
13 /******/  var installedModules = {};
14 /******/
15 /******/  // The require function
16 /******/  function __webpack_require__(moduleId) {
17 /******/
18 /******/    // Check if module is in cache
19 /******/    if(installedModules[moduleId]) {
20 /******/      return installedModules[moduleId].exports;
21 /******/    }
22 /******/    // Create a new module (and put it into the ca
23 /******/    var module = installedModules[moduleId] = {
24 /******/      i: moduleId,
25 /******/      l: false,
26 /******/      exports: {}
27 /******/    };
28 /******/
```

Javascript:

```
function foo(x) {
  return x.a;
}
```

Emitter:

```
main:
00000: getarg 0
00003: dup
00004: callprop "a"
00009: swap
00010: call-ignores-rv
00013: pop
00014: retrval
```

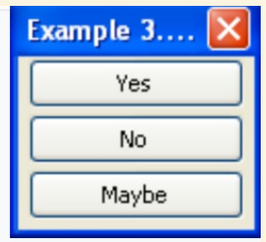
THE UI

```
17 # Side-Effects
18 ### Results may vary
19
```

ONCE...

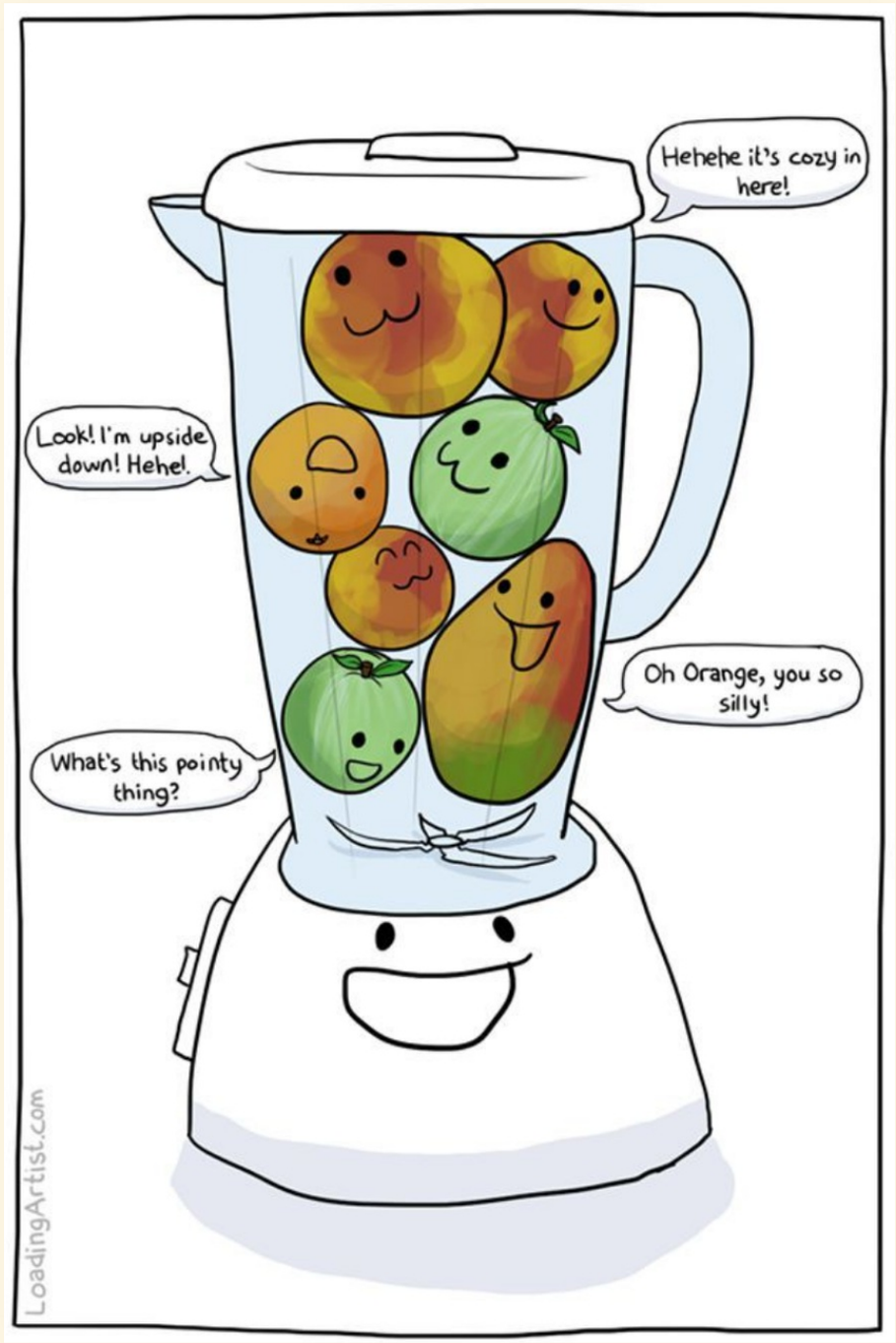
THERE WAS ONLY XUL

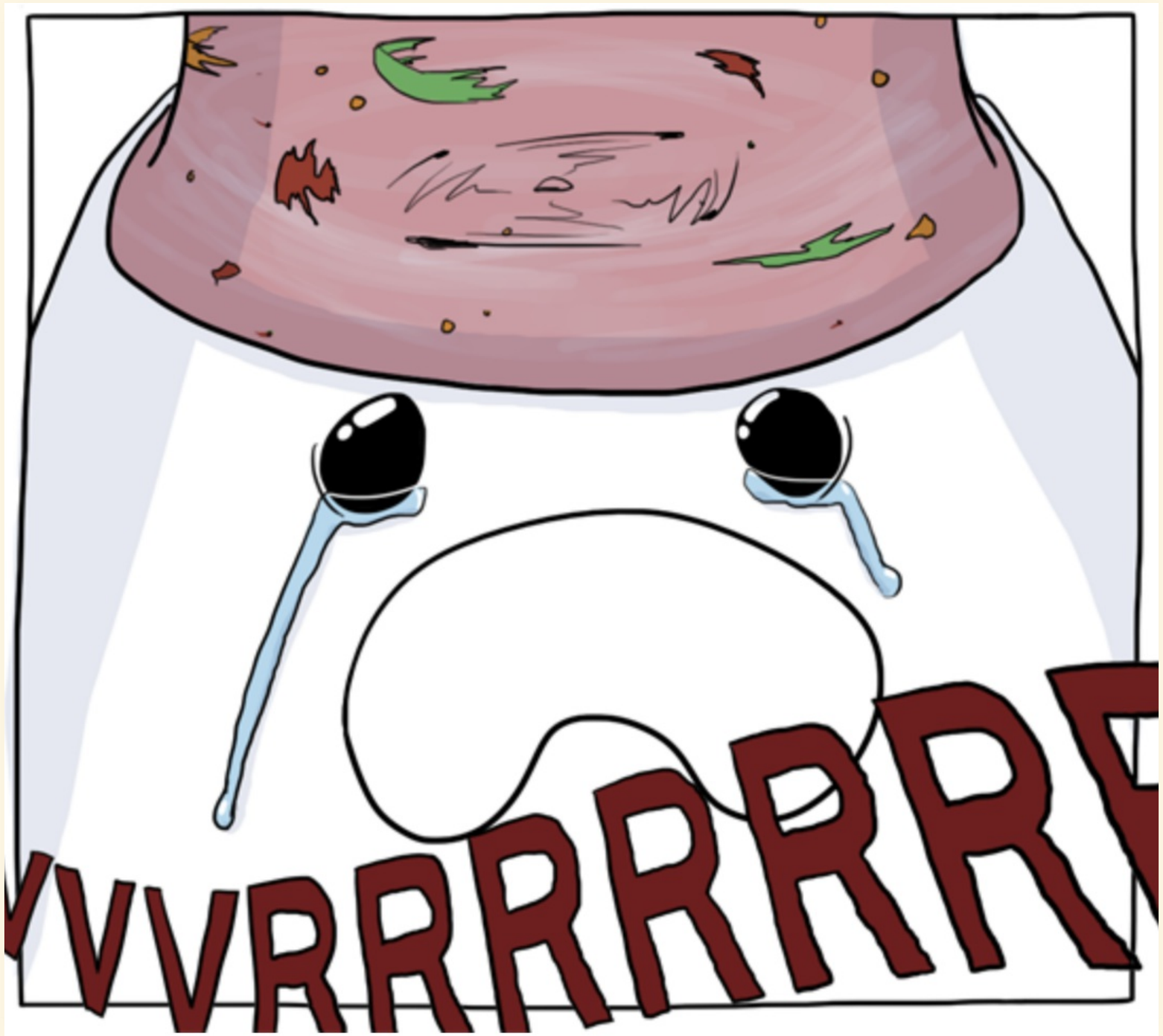
```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>

<window id="vbox example" title="Example 3...."
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <vbox>
    <button id="yes1" label="Yes"/>
    <button id="no1" label="No"/>
    <button id="maybe1" label="Maybe"/>
  </vbox>
</window>
```



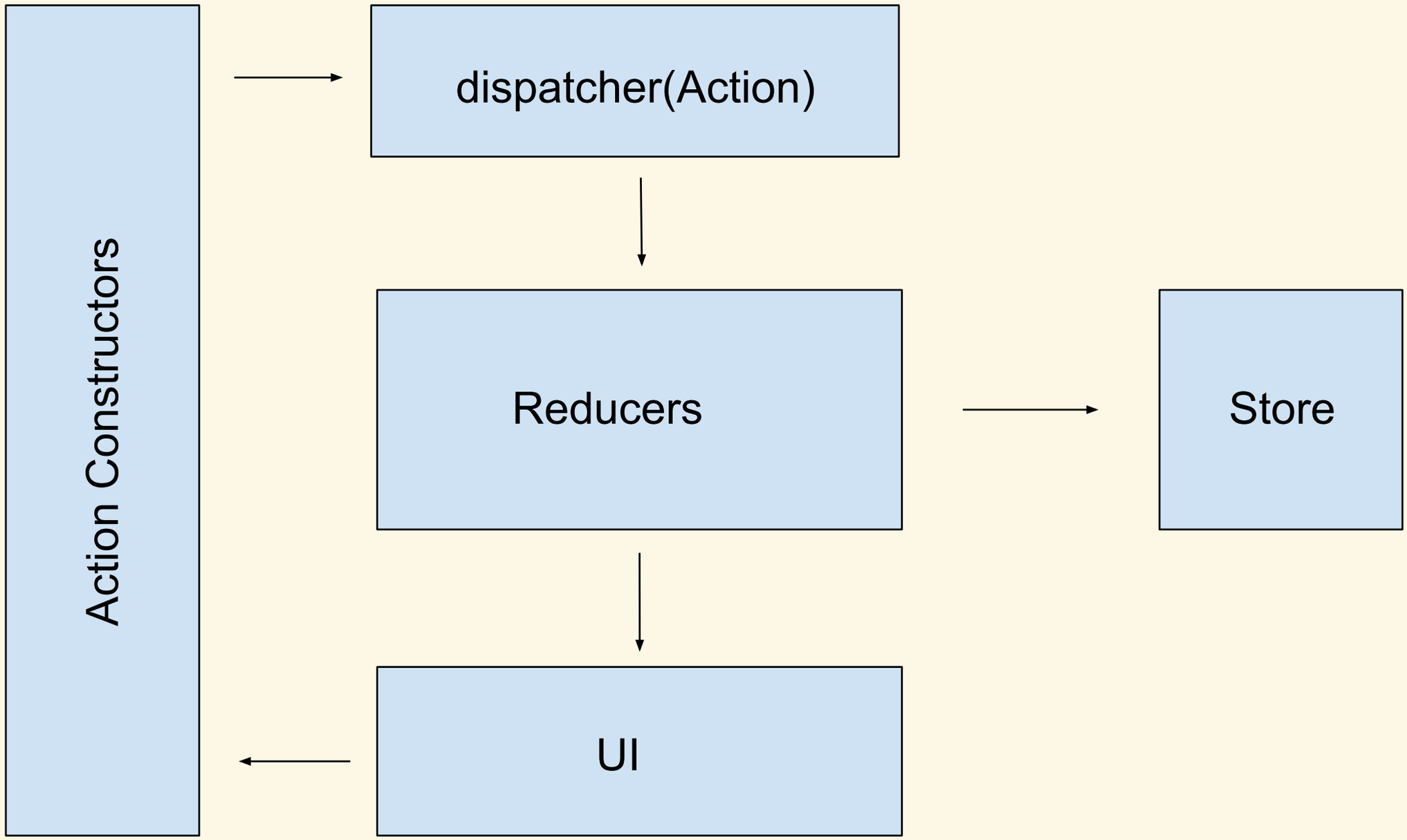
https://en.wikipedia.org/wiki/XUL

THE JAVASCRIPT BLENDER




VVVRRRRRRR

FLUX



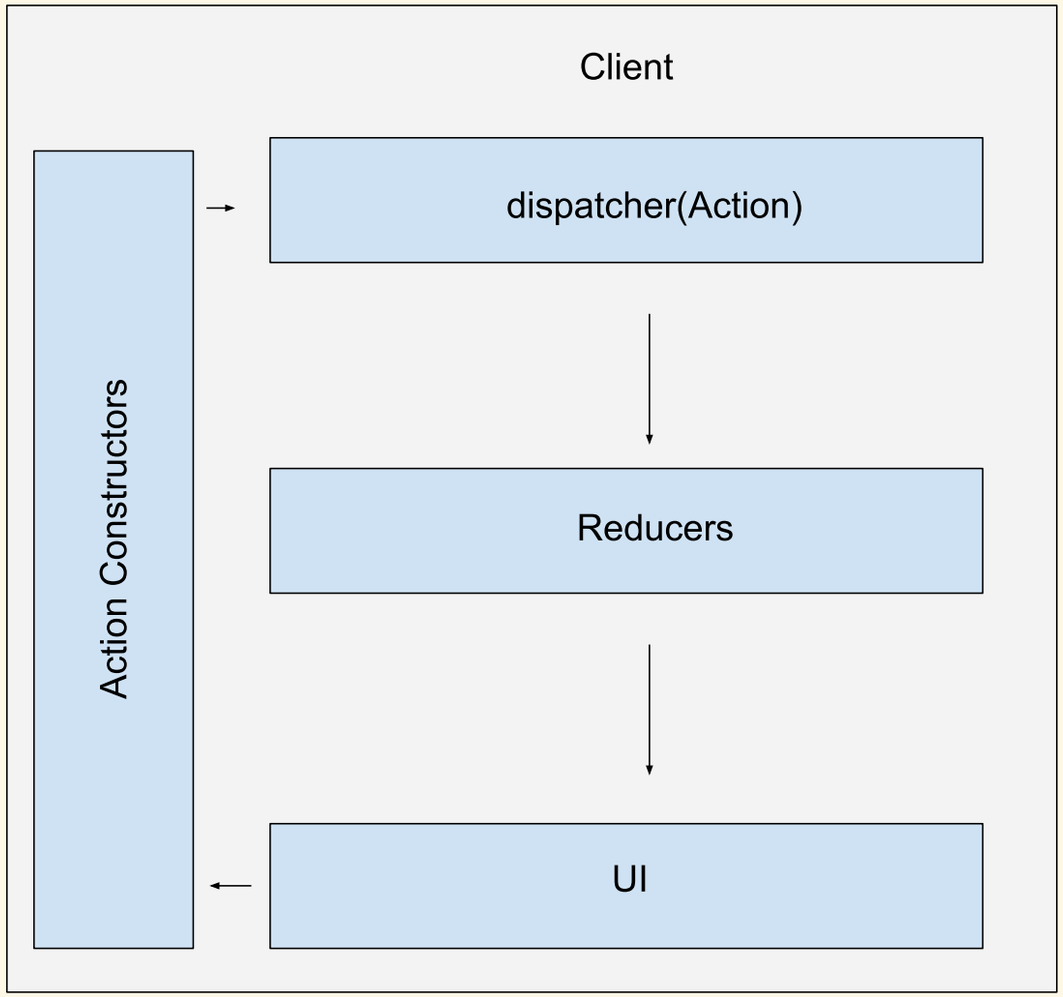
ACTION CREATOR

```
function basicAction() {
  return ({ dispatch }) => {
    dispatch({ type: "BASIC_ACTION", message: "hi" });
}
```

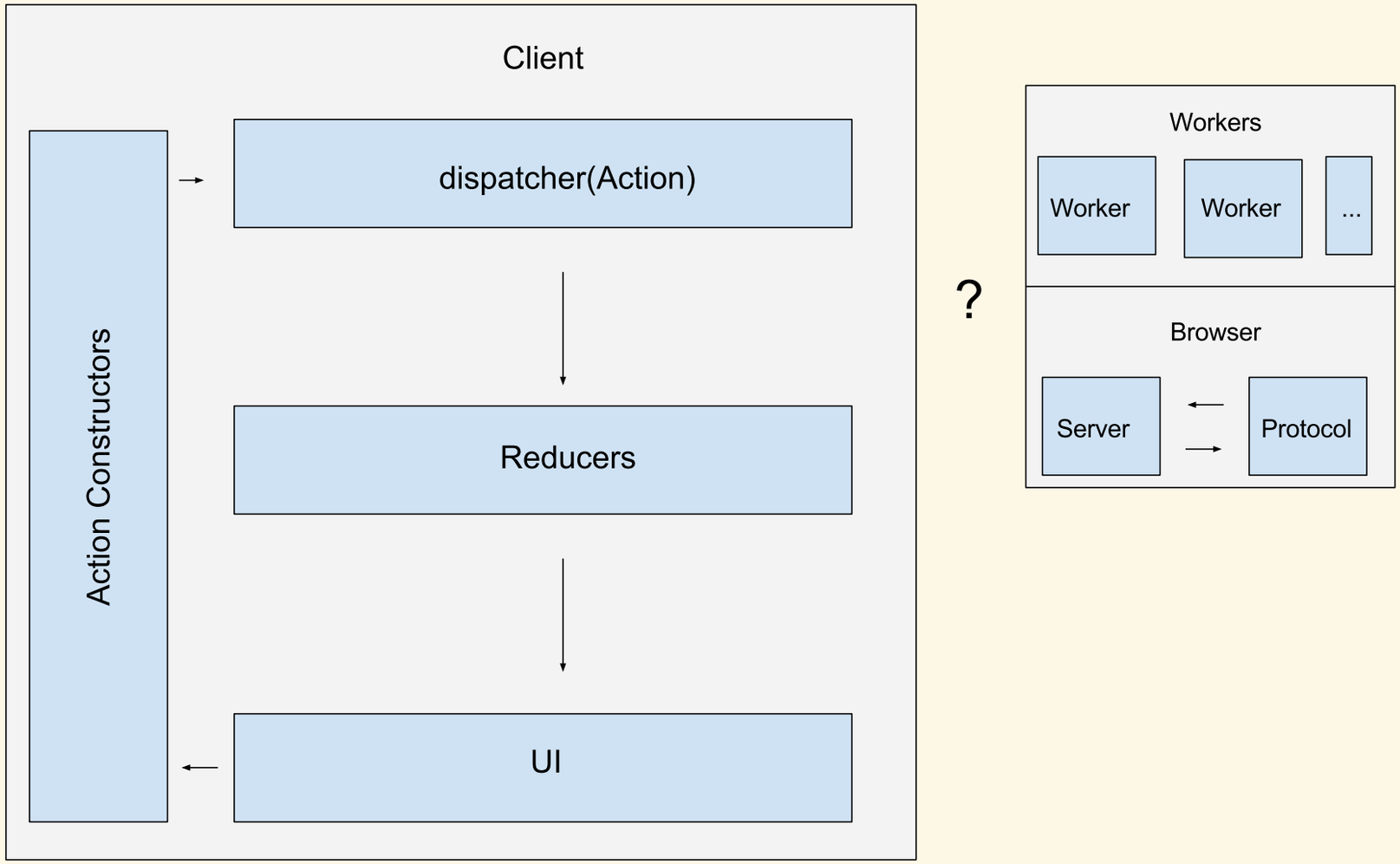
REDUCER

```
function update(state, action) {
  // action = { type: "BASIC_ACTION", message: "hi" }
  switch (action.type) {
    case "BASIC_ACTION":
      return { message: action.message };
    default:
      return state;
  }
}
```

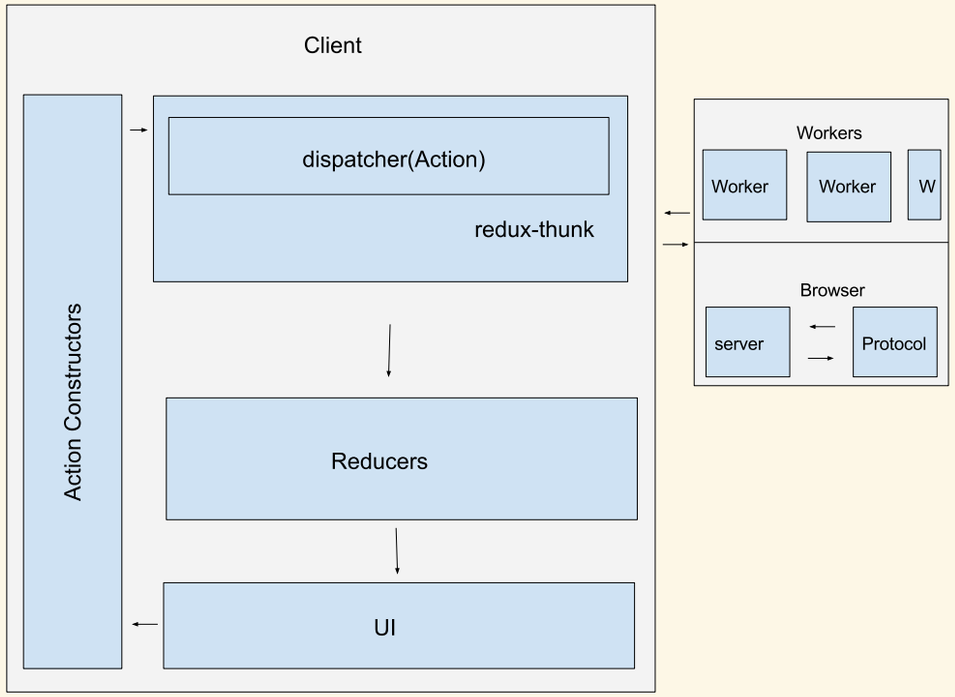
ASYNC DATA FLOW

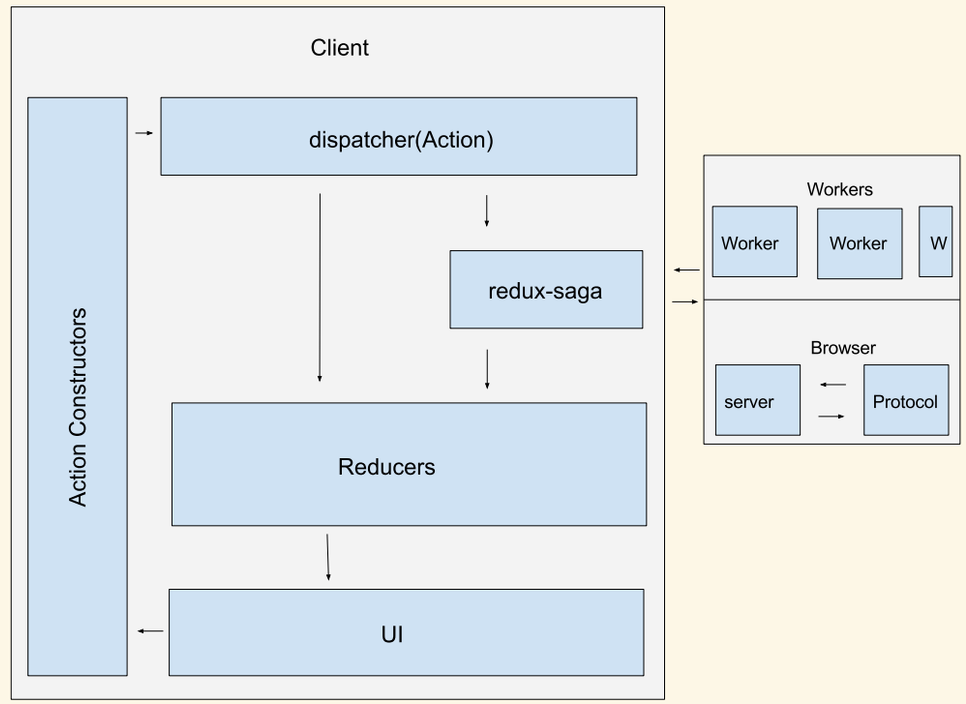


ASYNC DATA FLOW



TWO STRATEGIES





REDUX-THUNK

REDUX-SAGA

INTERLUDE:

UNDERSTANDING ASYNC METHODS IN JAVASCRIPT

- promise
- async / await
- generators

SIMPLE PROMISE

```
const asyncedFn = new Promise((resolve, reject) => {
  // ...
  if (success) {
    resolve(response);
  } else {
    reject("Error!");
  }
});
```

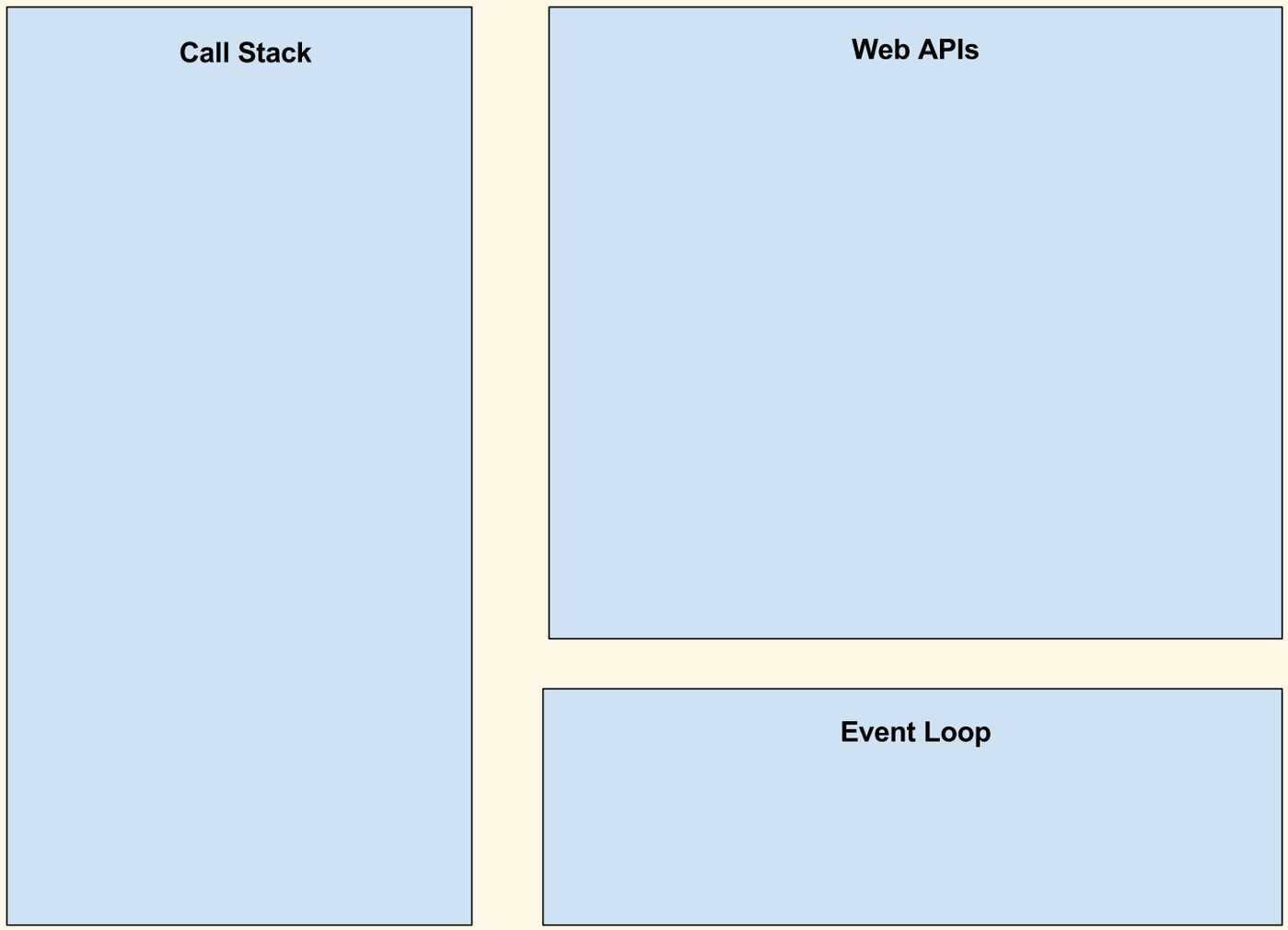
USING THE PROMISE

```
asyncFn
  .then(result => {
    return result;
  })
  .catch(err => {
    throw new Error(err);
  });
```

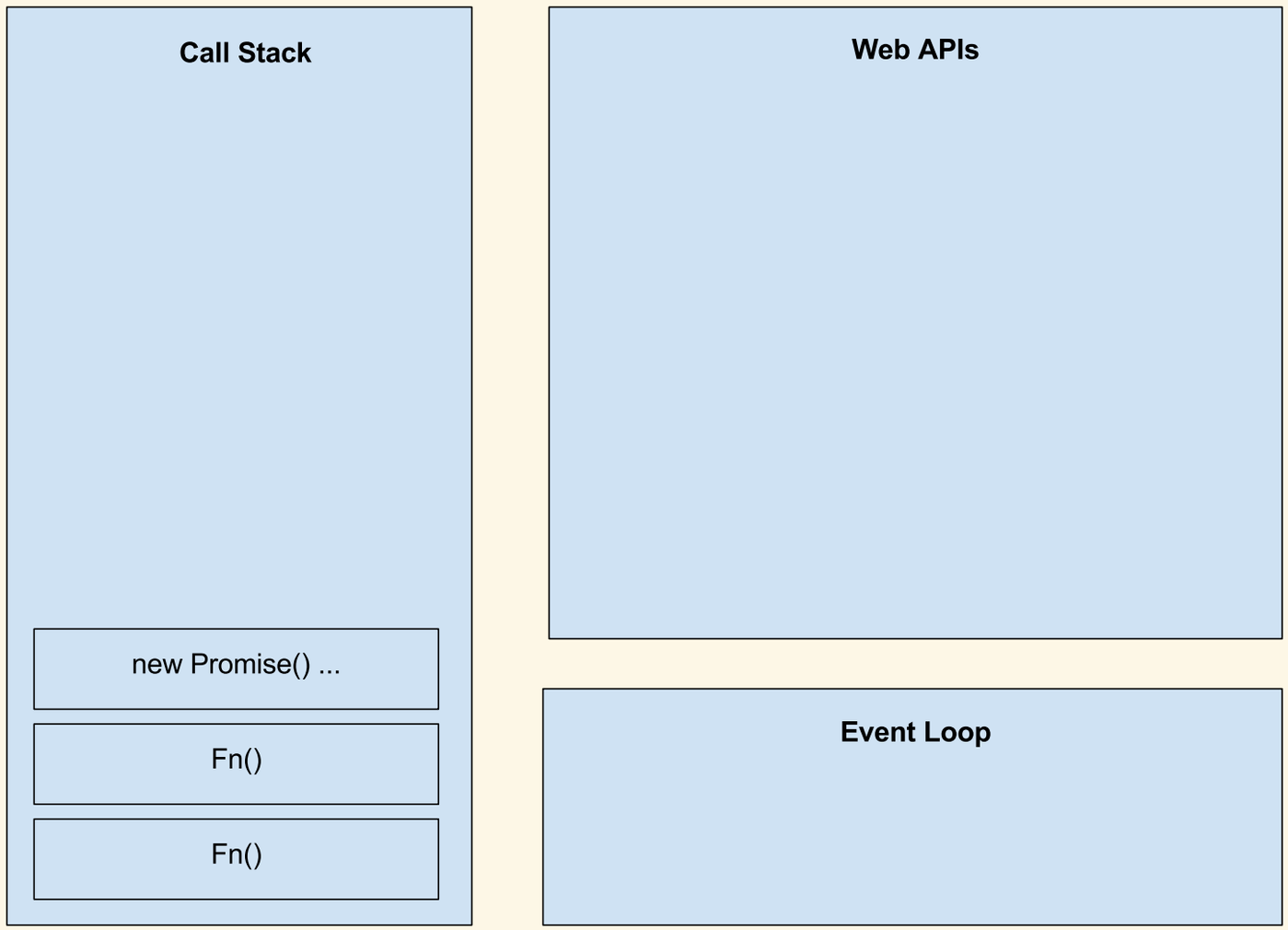
USING THE PROMISE

```
const initialState = doSomething1();

let globalVar;
asyncFn
  .then(result => globalVar = result)

doSomething2(global);
```

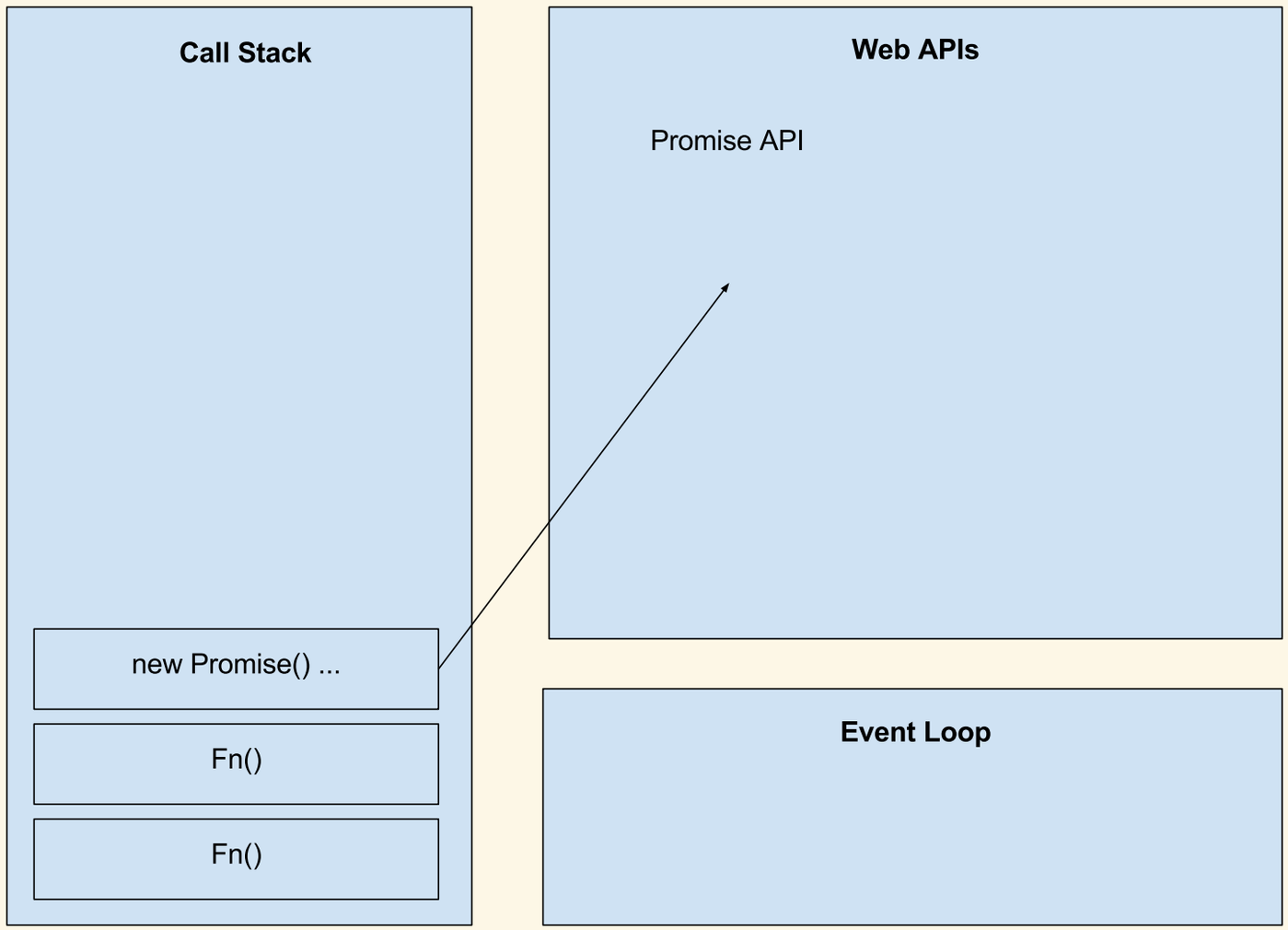
PROMISE AND CALLSTACK



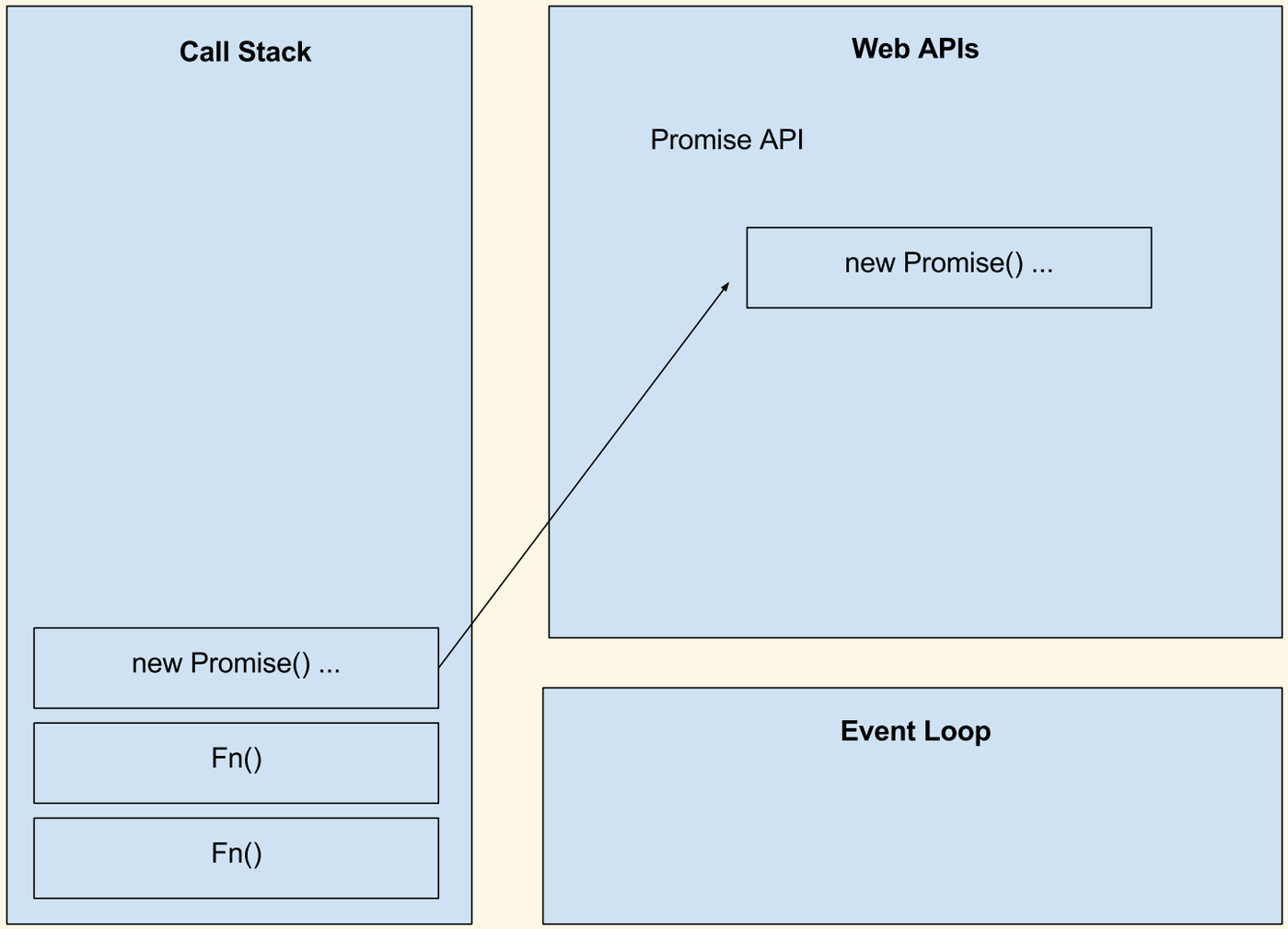
PROMISE AND CALLSTACK



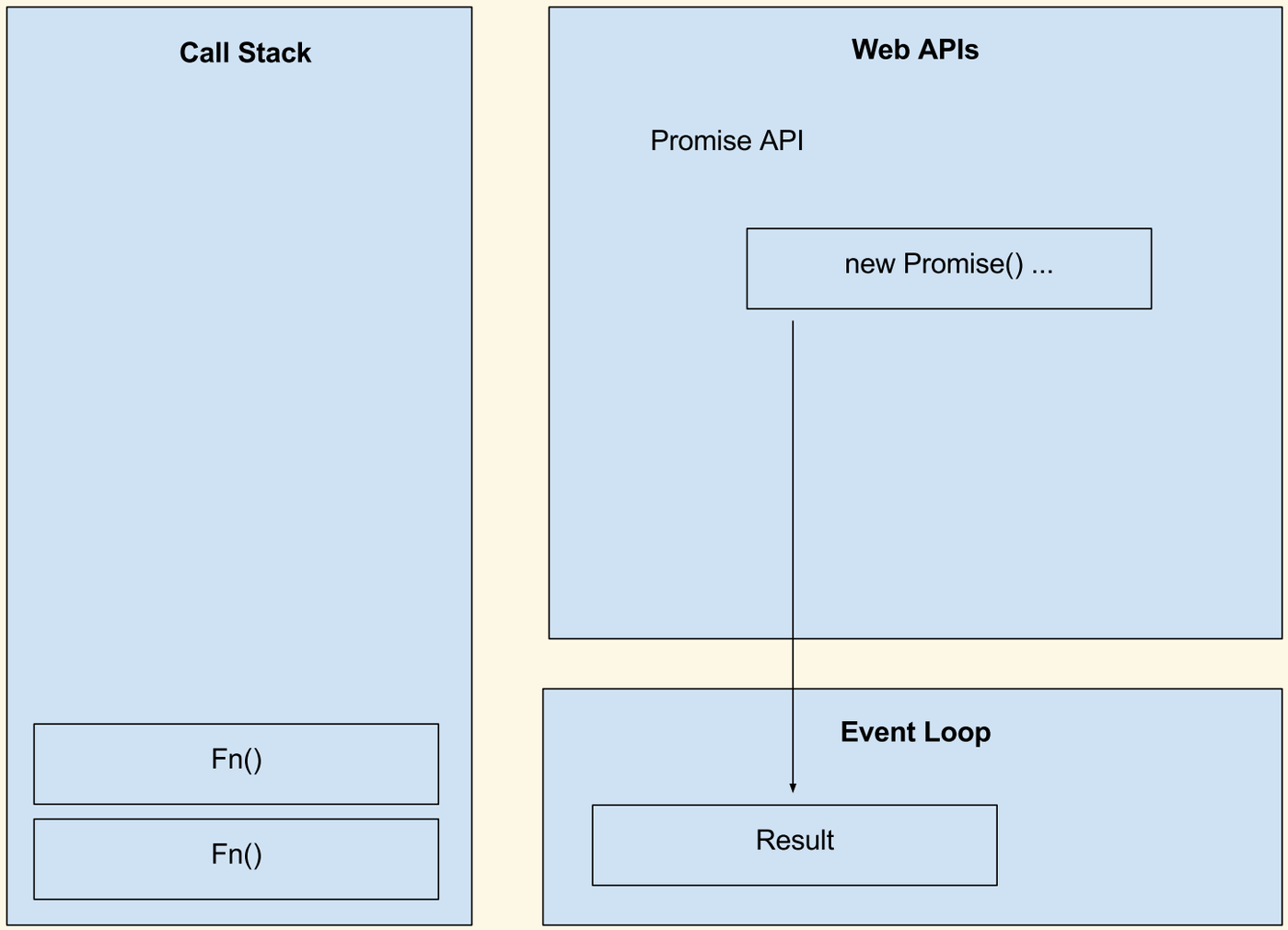
PROMISE AND CALLSTACK



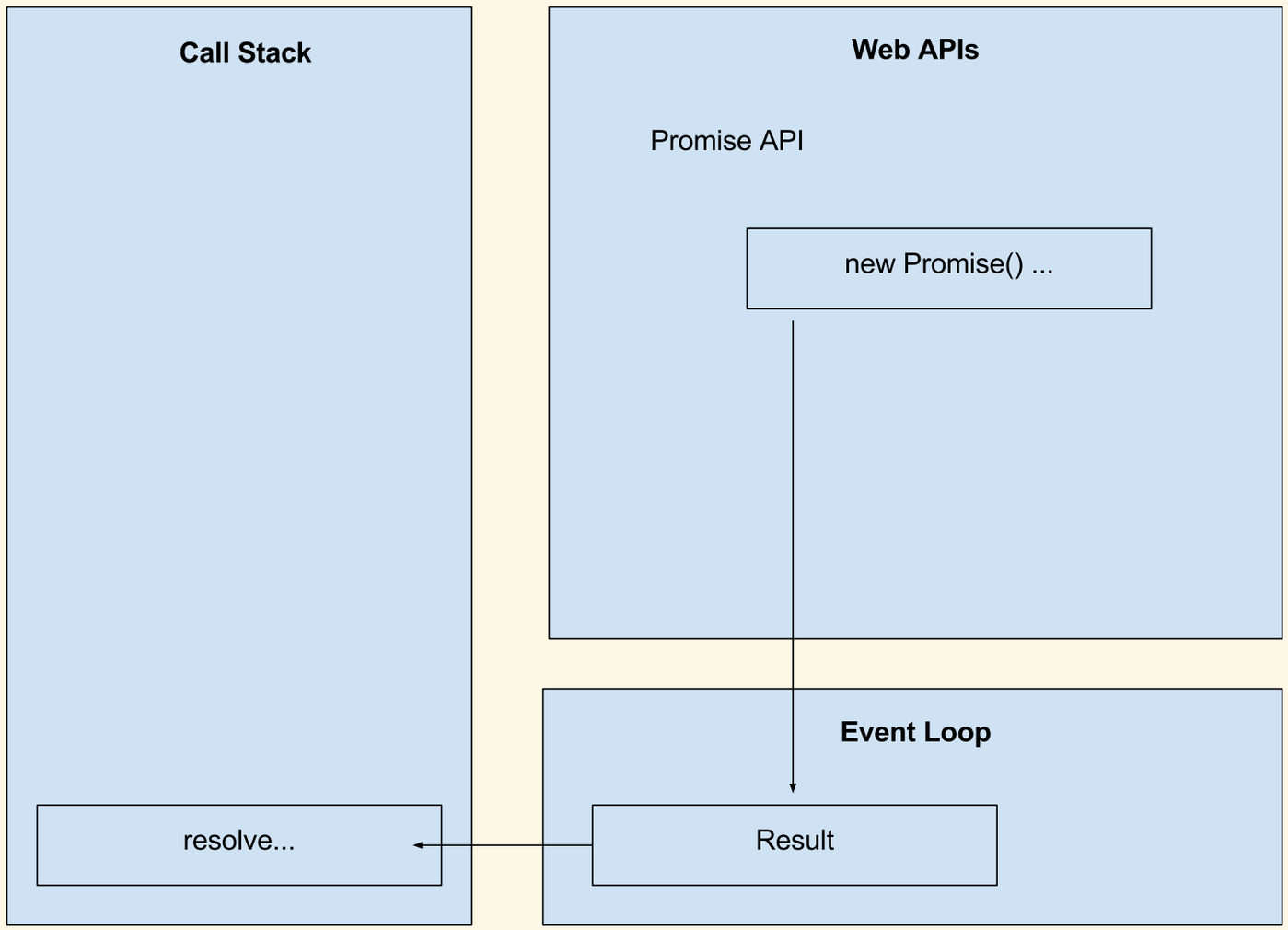
PROMISE AND CALLSTACK



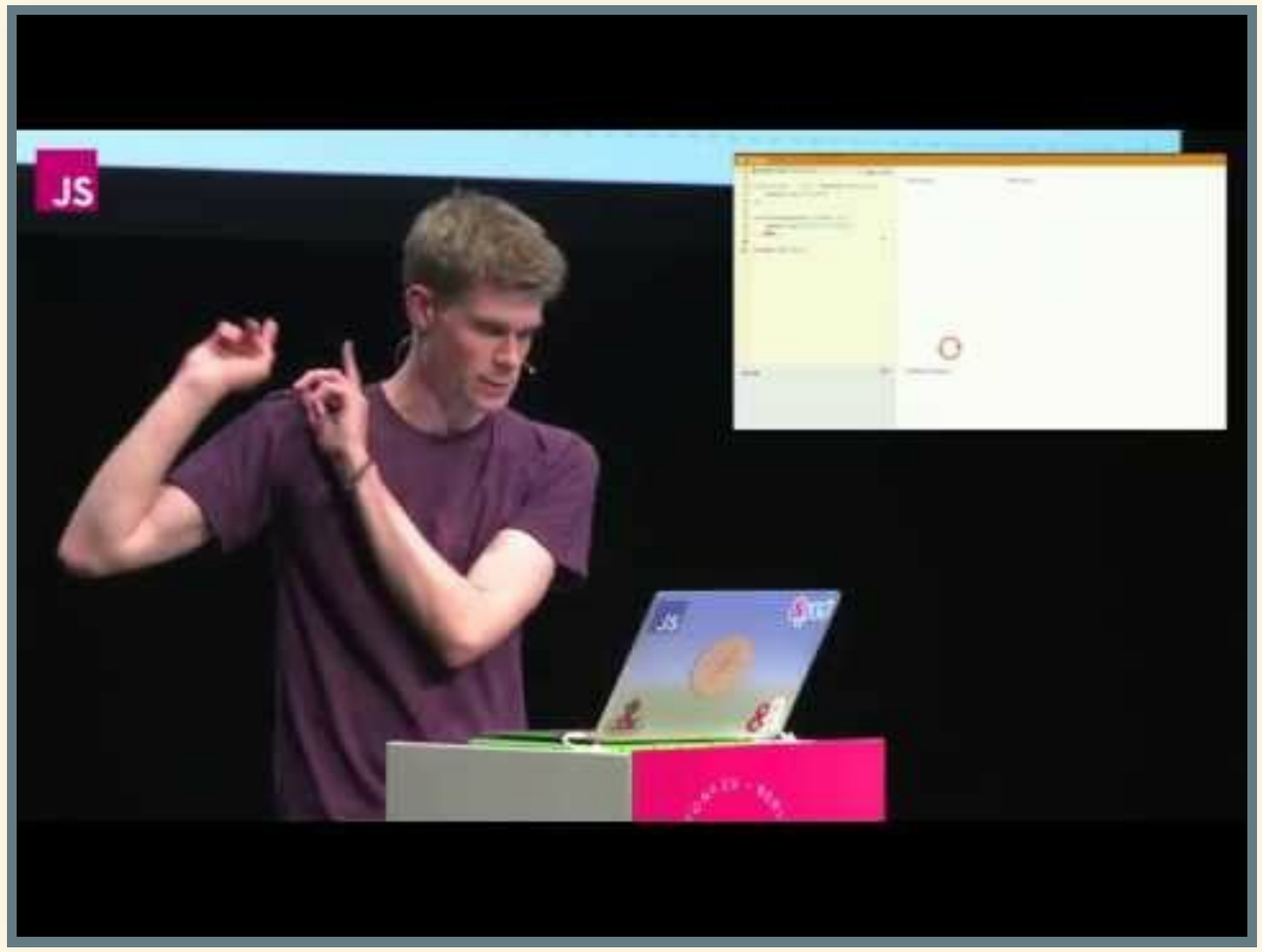
PROMISE AND CALLSTACK



PROMISE AND CALLSTACK



PHILIP ROBERTS: WHAT THE HECK IS THE EVENT LOOP ANYWAY?



USING THE PROMISE

```
const initialState = doSomething1();

let globalVar;
asyncFn
  .then(result =>
    doSomething2(result)
  )
```

USING THE PROMISE

```
const initialState = doSomething1();

let globalVar;
asyncFn
  .then(result =>
    doSomething2(result)
    asyncFn2
      .then(result =>
        // ...
      )
    // ...
  )
```

PROMISES AS ASYNC / AWAIT

```
async function something() {
  const initialState = doSomething1();
  const result = await asyncFn(initialState);
  const finalState = doSomething2(result);
  return finalState;
}
```

UNDERSTANDING GENERATORS

```
function* beuysGenerator() {
  yield "ja ja ja";
  yield "ne ne ne";
}

const jaNein = beuysGenerator();

jaNein.next() // { value: "ja ja ja", done: false }
jaNein.next() // { value: "ne ne ne", done: false }
jaNein.next() // { value: undefined, done: true }
```

Generators are functions that can be paused and resumed (think cooperative multitasking or coroutines), which enables a variety of applications.

http://exploringjs.com/es6/ch_generators.html

BODIL STOKKE - THE MIRACLE OF GENERATORS

REDUX-THUNK

README.md

Redux Thunk

Thunk middleware for Redux.


```
npm install --save redux-thunk
```

Note on 2.x Update

Most tutorials today assume Redux Thunk 1.x so you might run into an issue when running their code with 2.x.

If you use Redux Thunk 2.x in CommonJS environment, don't forget to add `.default` to your import:

```
- var ReduxThunk = require('redux-thunk')
+ var ReduxThunk = require('redux-thunk').default
```

If you used ES modules, you're already all good:

```
import ReduxThunk from 'redux-thunk' // no changes here 😀
```

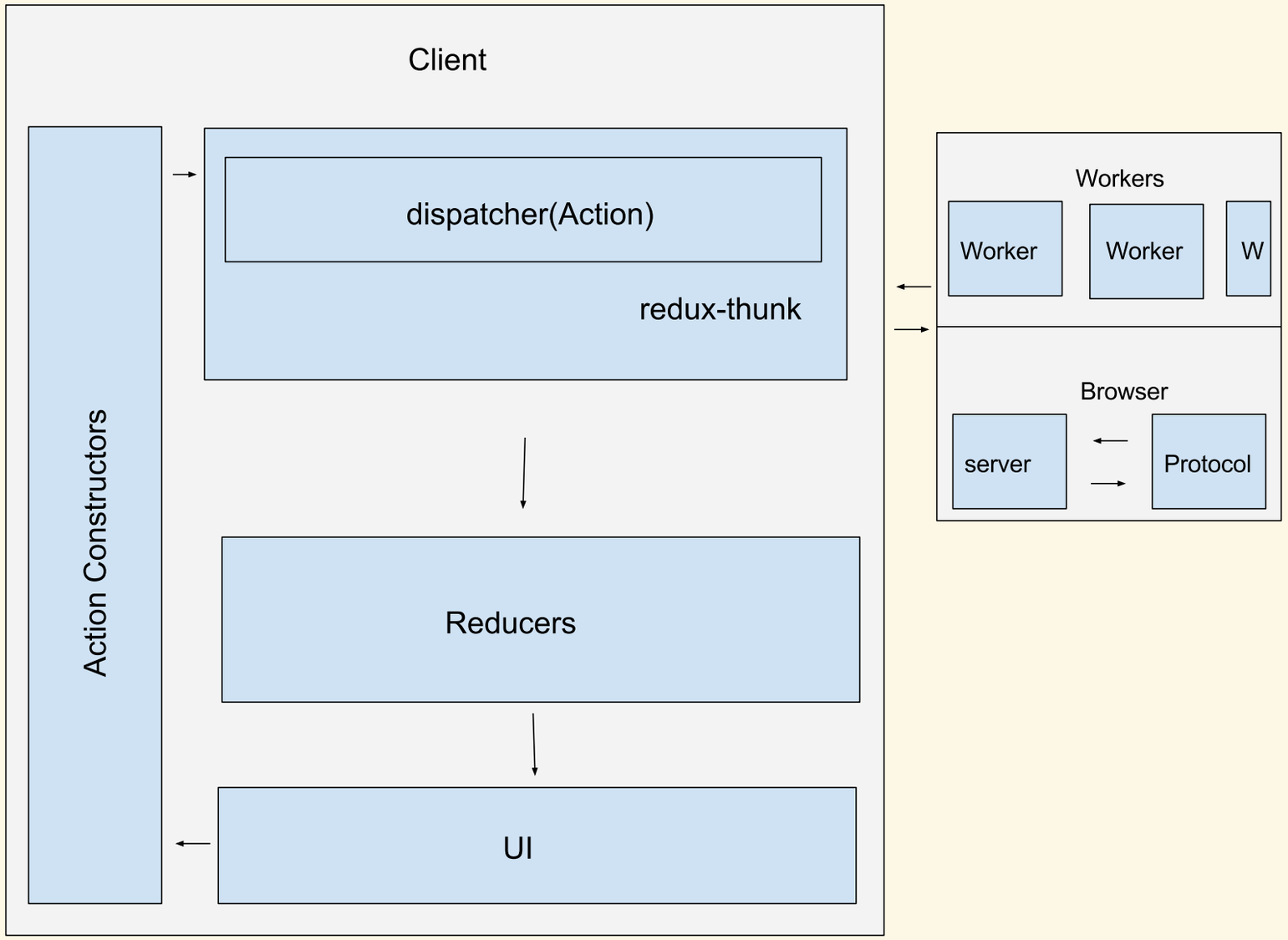
Additionally, since 2.x, we also support a UMD build:

```
var ReduxThunk = window.ReduxThunk.default
```

As you can see, it also requires `.default` at the end.

Why Do I Need This?

THUNK DATAFLOW



ACTION CREATOR

```
function basicAction() {
  return ({ dispatch }) => {
    dispatch({
      type: "BASIC_ACTION",
      message: "hi",
      [PROMISE]: async funtion() {
        return await asyncOperation(); // { message: `hi ${user}`
      }
    });
}
```

REDUCER

```
function update(state, action) {
  switch (action.type) {
    case "BASIC_ACTION":
      if (action.status === "start") {
        // { status: "start", message: "hi", ... }
        return { message: action.message };
      }
      if (action.status === "done") {
        // { status: "done", message: "hi", value: { message: "hi
        return { message: action.value.message };
      }
      return state;
    default:
      return state;
  }
}
```

REDUCER: ANOTHER WAY TO LOOK AT IT

```
function update(state, action) {
  switch (action.type) {
    case "BASIC_ACTION":
      switch (action.status) {
        case "start":
          // { status: "start", message: "hi", ... }
          return { message: action.message };
        case "done":
          // { status: "done", message: "hi", value: { message: "
          return { message: action.value.message };
        default:
          return state;
    default:
      return state;
  }
}
```

REDUX-SAGA

README.md

redux-saga

`redux-saga` is a library that aims to make application side effects (i.e. asynchronous things like data fetching and impure things like accessing the browser cache) easier to manage, more efficient to execute, simple to test, and better at handling failures.

The mental model is that a saga is like a separate thread in your application that's solely responsible for side effects. `redux-saga` is a redux middleware, which means this thread can be started, paused and cancelled from the main application with normal redux actions, it has access to the full redux application state and it can dispatch redux actions as well.

It uses an ES6 feature called Generators to make those asynchronous flows easy to read, write and test. *(if you're not familiar with them here are some introductory links)* By doing so, these asynchronous flows look like your standard synchronous JavaScript code. (kind of like `async` / `await`, but generators have a few more awesome features we need)

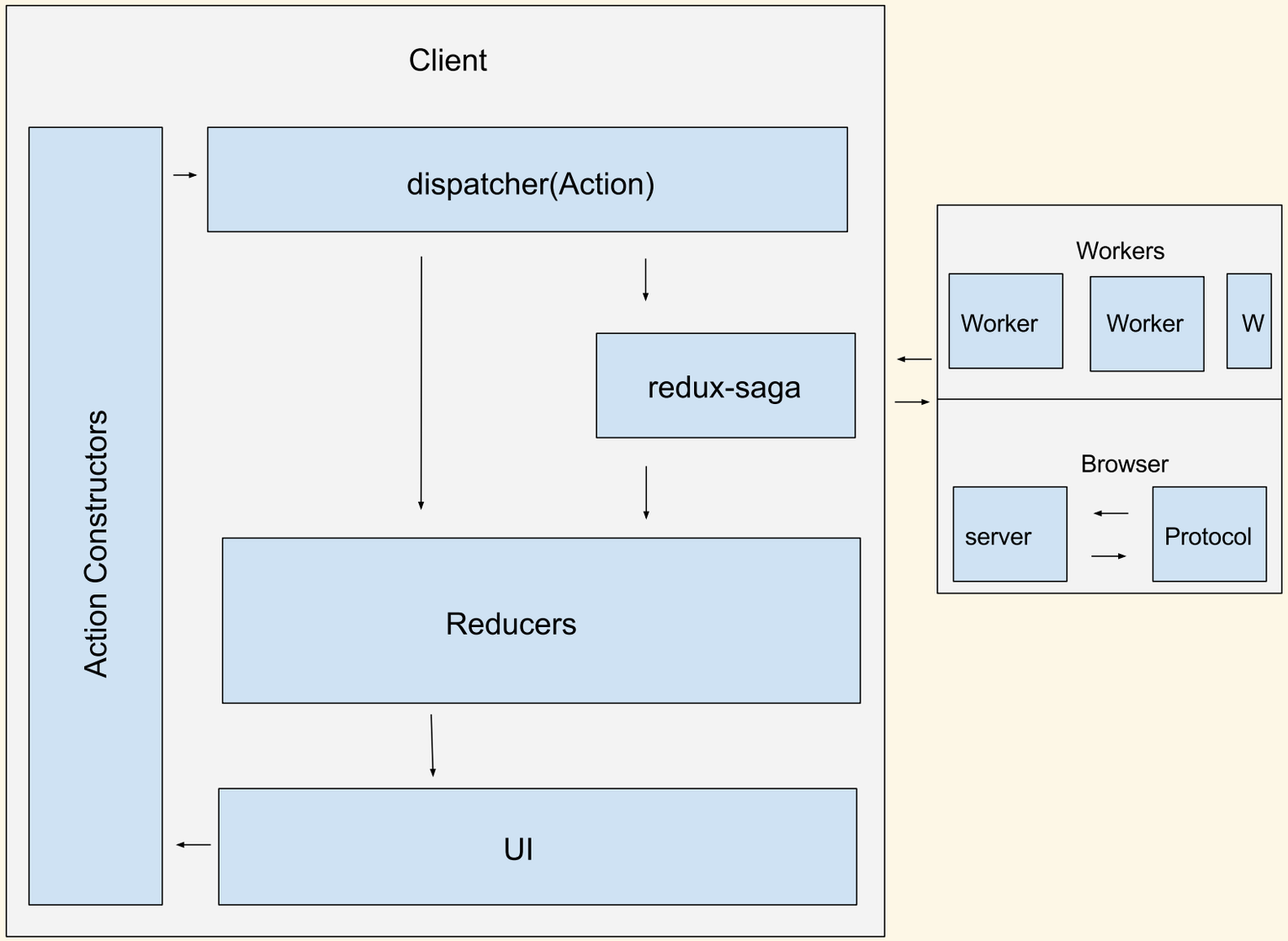
You might've used `redux-thunk` before to handle your data fetching. Contrary to redux thunk, you don't end up in callback hell, you can test your asynchronous flows easily and your actions stay pure.

Getting started

A long life transaction is a saga if it can be written as a sequence of transactions that can be interleaved with other transactions

-- Hector Garcia-Molina & Kenneth Salem

SAGA DATA-FLOW



ACTION CREATOR

```
function basicAction() {
  return ({ dispatch }) => {
    dispatch({ type: "BASIC_ACTION", message: "hi" });
}
```

SAGA

```
function* watchBasicAction() {
  yield takeEvery('BASIC_ACTION', basicSaga)
}

function* basicSaga() {
  const message = await asyncOperation(); // { message: `hi ${use
  yield put({ type: "SUCCESSFUL_ACTION", message });
}
```

THUNK WRAPPED DISPATCH

```
dispatch({
  type: "BASIC_ACTION",
  message: "hi",
  [PROMISE]: async funtion() {
    return await asyncOperation(); // { message: `hi ${user}` }
  }
});
```

REDUCER

```
function update(state, action) {
  switch (action.type) {
    case "BASIC_ACTION":
      // action = { message: "hi", ... }
      return { message: action.message };
    case "SUCCESSFUL_ACTION":
      // action = { message: "hi user", ... }
      return { message: action.message };
    default:
      return state;
  }
}
```

THE PROBLEM

SPOOKY ACTION AT A DISTANCE

CONSIDER THE BUG

opening a loaded source causes the source to blink

```
server event "new source"
|-> new source
    |-> dispatch "ADD_SOURCE"
    |-> maybe add source map
    |    |-> load sourceMap
    |    |-> add original sources
    |-> maybe select source
    |    |-> load Source Text
    |    |-> re-evaluate breakpoints
    |    |-> dispatch addBreakpoints
    |-> maybe add breakpoints
         |-> load Source Text
         |-> dispatch selectSource
```

LOAD SOURCE TEXT

- is used in two places
- is costly

THE SOURCE OF THE BUG

```
function newSource(source) {
  return async ({ dispatch, getState }) => {
    dispatch({ type: "ADD_SOURCE", source });
    selectedSource(getState(), dispatch, source);
    await checkBreakpoints(getState(), dispatch, source);
    dispatch(loadSourceMap(source));
  };
}
```

```
function selectSource(sourceId}) {
  return ({ dispatch, getState }) => {
    // ...
    if (source.isSelected) {
      await dispatch(loadSourceText(source));
      // ...
    }
  };
}
```

```
async function checkBreakpoints(state, dispatch, source) {
  // ...
  if (breakpoints.length) {
    await dispatch(loadSourceText(source));
    // ...
  }
}
```

SOLUTION 1

USING THUNKS

```
function newSource(source) {
  return async ({ dispatch, getState }) => {
    dispatch({ type: "ADD_SOURCE", source });
    await selectedSource(getState(), dispatch, source);
    checkBreakpoints(getState(), dispatch, source);
    dispatch(loadSourceMap(source)); // an async operation
  };
}
```

SOLUTION 2

USING SAGAS

```
function* watchNewSource() {
  yield takeEvery('NEW_SOURCE', newSource)
}

function* newSource(source) {
  yield put({ type: "ADD_SOURCE", source });
  const proc1 = yield fork(loadSourceMap, source));
  const proc2 = yield fork(selectSource, source));
  const proc3 = yield fork(checkBreakpoints, source));
  yield join([ proc1, proc2, proc3 ]);
  yield put({ type: "SOURCE_ADDED, source });
}
```

BLOCKING

```
take
take.maybe
put.resolve
call
all
join
cancel
actionChannel
```

NON-BLOCKING

```
takeEvery
put
fork
spawn
```

SAGA: BLOCKING PUT

```
// thunk
await dispatch(loadSourceText(source));

// saga
yield put.resolve({ type: "LOAD_SOURCE_TEXT", source });
```

SAGA: CHANNELS

```
yield takeEvery('BASIC_ACTION', basicSaga)

// vs

const requestChan = yield actionChannel('LOAD_SOURCE_TEXT')
```

```
function* watchLoadSourceText() {
  const requestChan = yield actionChannel('LOAD_SOURCE_TEXT')
  while (true) {
    const { payload: { source } } = yield take(requestChan)
    if (source.text) {
      yield source;
    } else {
      yield call(loadSourceText, source)
    }
  }
}
```

```
function* loadSourceText(source) {
  const data = yield call(loadSourceTextContents, source);
  yield put.resolve({ type: "SOURCE_TEXT_AVAILABLE", source, ...d
  yield call(setSymbols, source.id);
  yield call(setEmptyLines, source.id);
}
```

DOING MORE WITH SAGAS

```
function* watchLoadSourceText() {
  const requestChan = yield actionChannel('LOAD_SOURCE_TEXT')
  while (true) {
    const { payload: { source } } = yield take(requestChan)
    if (source.text) {
      yield source;
    } else {
      // lets do something else here...
    }
  }
}
```

ROLL FORWARD ON AN ERROR

```
// retry and error handling
function* loadSourceTextRepeat(source) {
  for(let i = 0; i < 5; i++) {
    try {
      yield call(loadSourceText, source)
    } catch (err) {
      yield call(delay, 1000); // try again in a bit
    }
  }
  yield put({ type: "LOAD_TEXT_ERROR", source });
}
```

DO SOMETHING ELSE ENTIRELY

```
// retry and error handling
function* streamSource(source) {
  try {
    yield call(loadSourceText, source)
  } catch (err) {
    yield put({ type: "DELETE_SOURCE", source }); //re-enter the
  }
}
```

THUNK REDUCER

```
function update(state, action) {
  switch (action.type) {
    case "LOAD_SOURCE_TEXT":
      if (status === "error") {
        // ... we would have to handle rollback here
      }
      return state;

    default:
      return state;
  }
}
```

SAGA REDUCERS

```
function update(state, action) {
  switch (action.type) {
    case "LOAD_SOURCE_TEXT":
      // ...

    case "SOURCE_TEXT_AVAILABLE":
      // ...

    case "SOURCE_ERROR":
      // ...

    case "DELETE_SOURCE":
      // ...

    case "RESET_TO_LOADING":
      //
```

OBSERVATIONS

- simple promises are sufficient in most cases
- thunks are useful if you need eager updates and error handling on the reducer
- sagas are useful when working with complex scheduling tasks

CRITICISMS OF BOTH THUNKS AND SAGAS

- they are too powerfull and do too much
- the same can be achieved with simple promises
- thunks lead to duplicate async logic
- sagas and generators are difficult to understand

POSITIVE ASPECTS OF THUNKS

- easy to start using
- work for a large number of cases
- good for eager updates when an action reflects an api call

POSITIVE ASPECTS OF SAGAS

- excellent for testing
- handle complex scheduling very well
- allows fine grained control over what happens in the system

WHAT DID WE DO IN THE END?

We are sticking with thunks... for the time being

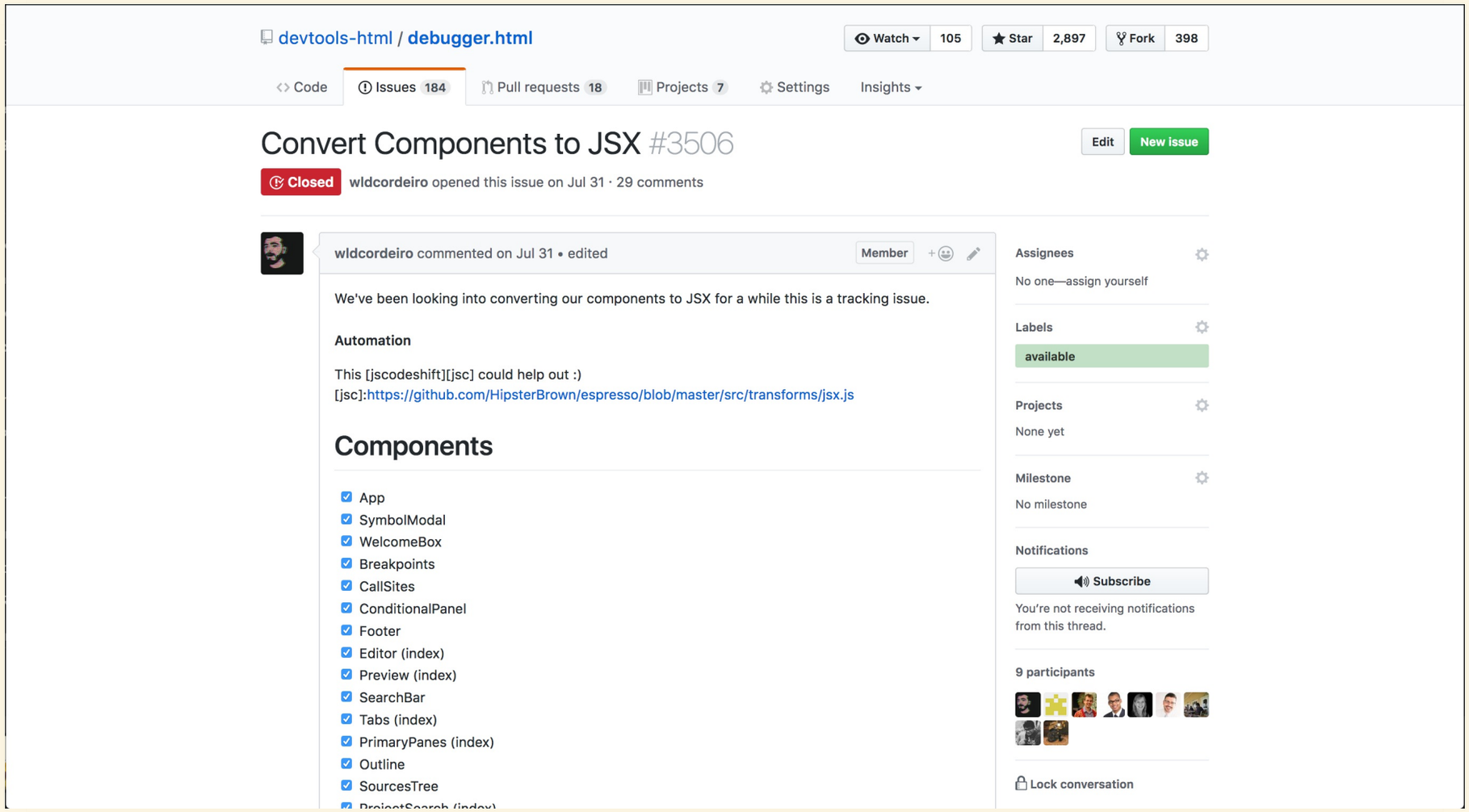
Reasons:

- while difficult to maintain, the community is familiar with them
- it is not yet the right time to do a large scale refactoring

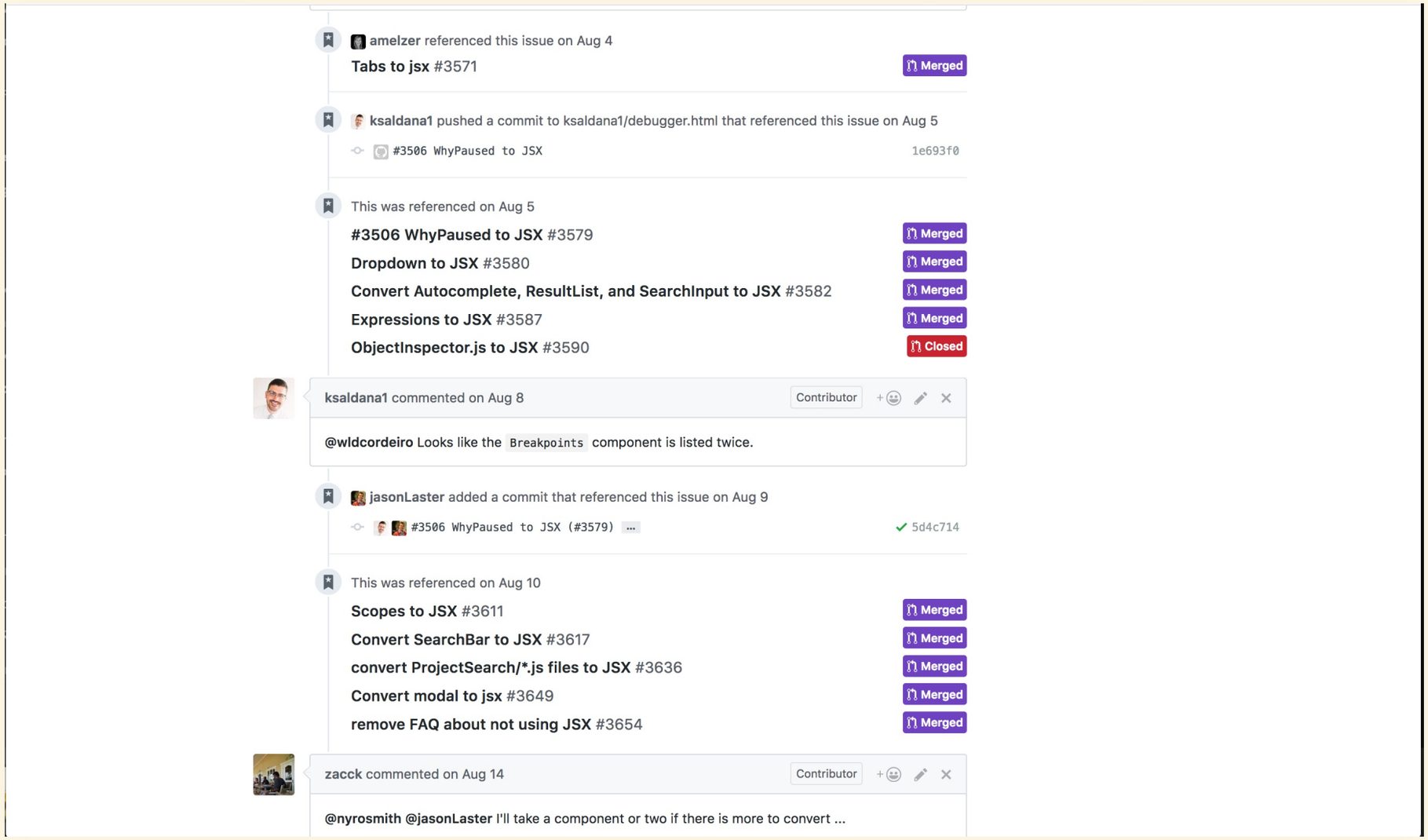
GETTING THE COMMUNITY READY FOR A CHANGE

- discussed with core community members
- received feedback from the maintainers of redux-saga
- experimented on a small scale

GOING FORWARD AS A COMMUNITY



GOING FORWARD AS A COMMUNITY



SO...

PERHAPS YOU ARE INTERESTED

http://github.com/devtools-html

THANK YOU!

http://github.com/devtools-html

@ioctaptceb