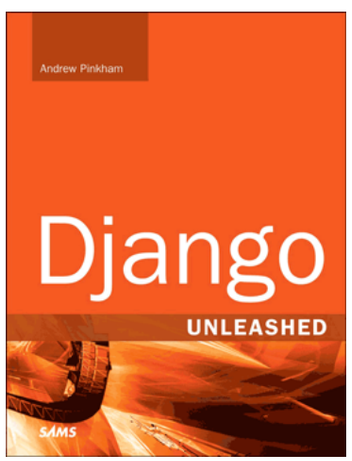
DJANGOCON US 2017

UNDERSTANDING JAVASCRIPT VIA REACT

$ whoami

ANDREW PINKHAM

- Author of Django Unleashed
- Founder of JamBon Software
- CertCycle



JAMBON software

CERTCYCLE

DJANGOCON US 2017

UNDERSTANDING JAVASCRIPT VIA REACT

TALK OUTLINE

- How did this happen?
- Brief History of the Web
- JavaScript (and HTML)
- JavaScript Frameworks (including React)
- React's Ecosystem
- React and Django
- The Future

XII·P·

HECESTHORRENDA
CARIBDIS

D

F

HVST

LOLOT

VAST

DVVANES TRÕDANES

DVVANES

NYGAVIK

DIA
HELGALA
TERRA NOBILIVM

HORVPISCIV
CAPITIBVS
VTVNTVR
LOCO LIGNO

STEK

ROOPEDVM

TASKED WITH:

- Learn JavaScript
- Learn the JavaScript Ecosystem
- Pick a Framework
- Build a dynamic frontend website to communicate with an API via REST (the Django bit)

I DID NOT UNDERSTAND THE PROBLEM

DJANGO

- I need to build a website. What should I use?
- Oh, you should use Django.

DJANGO

- I need to build a a software application that communicates with other computers over the HTTP protocol by providing HTML and CSS documents to those computers. What should I use?

- I recommend the Django framework, because it supplies all the code you need to communicate via HTTP, and it provides methods to rapidly produce HTML and package CSS for that HTML.

JAVASCRIPT

- I need to build a dynamic frontend. What should I use?
- Oh, you should use Angular/React/Vue/RxJS/CycleJS.

WHAT IS THE PROBLEM?

HTML

- HTML first appears in 1991
- Mosaic, Netscape Navigator, Internet Explorer
- HTML is intended to share documents: provides the Document Object Model (DOM), directed graph of nodes
- HTML provides the content and structure of a document
- Loosely analogous to Model (content) and View (structure)
- CSS is first proposed in 1994; spec released in Dec 1996

JAVASCRIPT

- JavaScript built in 10 days as "glue language" to cement partnership between Sun and Netscape against Microsoft in 1995

- Brendan Eich wants to put Scheme in the browser; politics give it Java-like syntax and the name (JavaScript). Prototypal inheritance is taken from Self.

- JavaScript is built to manipulate the DOM

- JavaScript spec changes over time

THE PROBLEMS

- JavaScript redefines known vocabulary and is better conceptualized as a functional language
- JavaScript is implemented in different browsers with slightly different features
- HTML's DOM is a directed graph with it's own hidden state and event handling; HTML differs across browsers
- HTML is both the state and display of the program

JAVASCRIPT QUIRKS

- A human/developer problem
- Read Douglas Crockford's "JavaScript: the Good Parts"
- Read Eric Elliott's "Programming JavaScript Applications"
- Read Kyle Simpson's JavaScript Series
- FrontEnd Masters and Egghead.io

JAVASCRIPT IN DIFFERENT BROWSERS

- JQuery
- Mootools

DIFFERENT LANGUAGE VERSIONS

- EcmaScript (ES) and TC39
- ES 3 or 5 in browsers
- ES 6 and 7... but you can't upgrade the browser!
- Write ES 6 or 7, but ship ES 5?
- Compiler JS to JS - "transpilation"
- Bundle namespaced JS into single file

TRANSPILERS

- Babel JS for ES 6 to ES5
- Bublé
- CoffeeScript and TypeScript

BUNDLES [AND PIPELINES]

- [Grunt or Gulp] and Browserify
- Webpack (for apps)
- Rollup (for libraries)

ENVIRONMENT TOOLS

- Package Managers
  - NPM
  - Yarn
  - PNPM
- Linters
  - JSLint and JSHint
  - ESLint

STARTING STACK

- Babel for ES 6 transpilation
- Webpack for bundling
- ESLint for linting
- Yarn for package management? NPM has fixed the issue? What about PNPM?

NOMENCLATURE

- Library: a codebase that you call from your code (`requests`)

```
def main():
    r = requests.get(
        'http://placekitten.com/300/300')
```

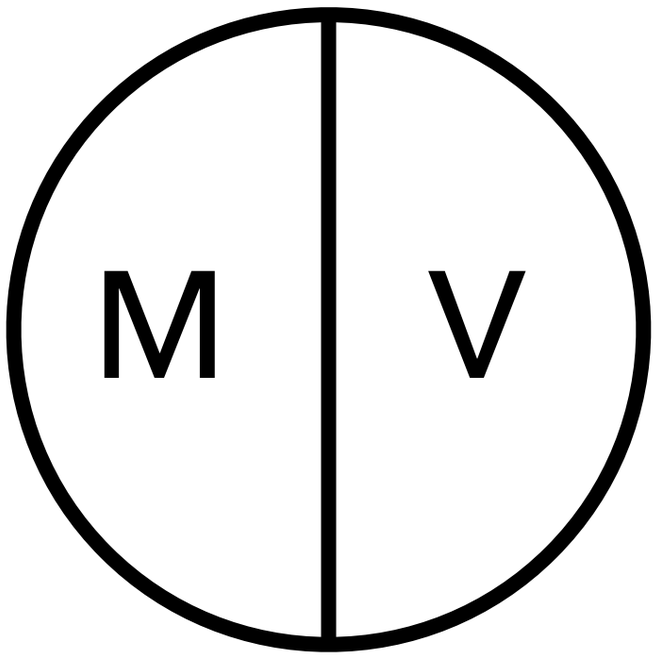
- Framework: a codebase that calls your code (`Django`)

```
def a_django_view(request):
    return render(request, 'template.html')
```

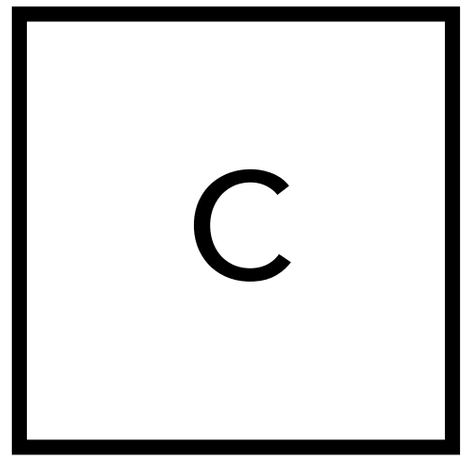
THIS IS YOUR REMINDER THAT THE DOM IS ACTUALLY A GIANT, MUTABLE, GLOBAL VARIABLE IN THE MIDDLE OF YOUR PROGRAM.

@polotek Marco Rogers

VANILLA JS



HTML

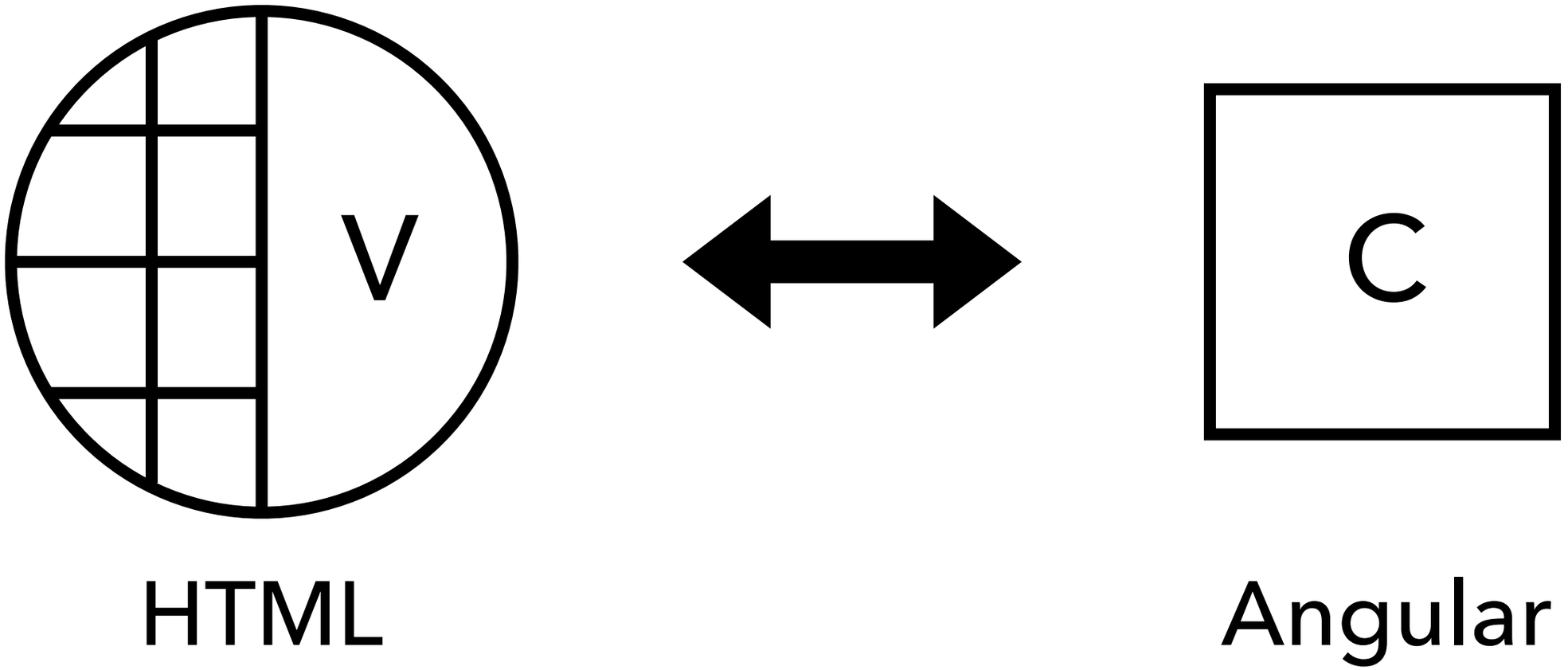


JavaScript

MODEL VIEW CONTROLLER

- Angular Framework
  - Referred to as MVC
  - JavaScript is a separate Controller
  - Annotate HTML with custom tags (directives) to split single variable (DOM) until smaller scopes
  - Two-way binding
  - Content is dynamic; structure of app still in HTML

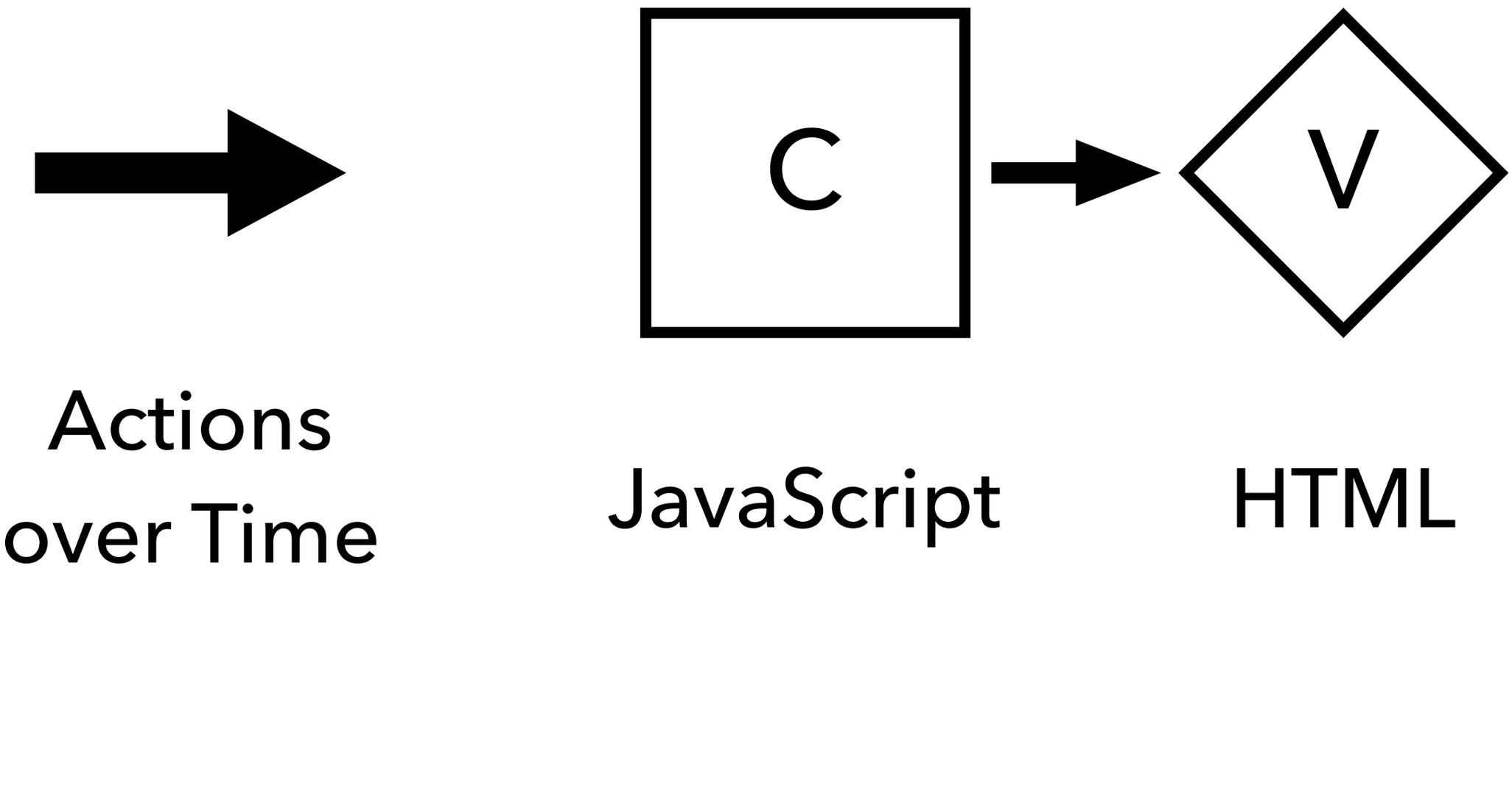
ANGULAR



REACTIVE PROGRAMMING

- RxJS and CycleJS
  - Streams or Observables
    - Inputs over time
    - Functions defined at beginning that operate over time

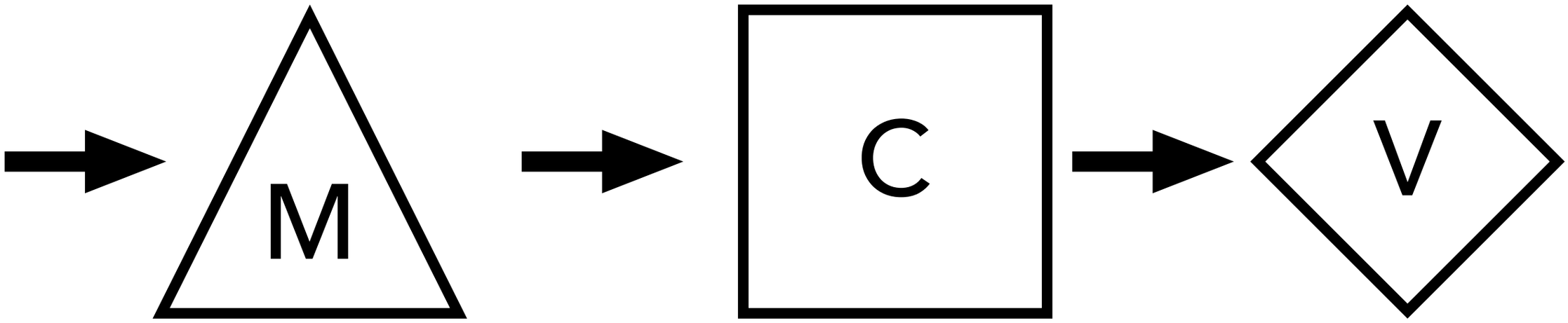
CYCLE AND REACTIVE JS



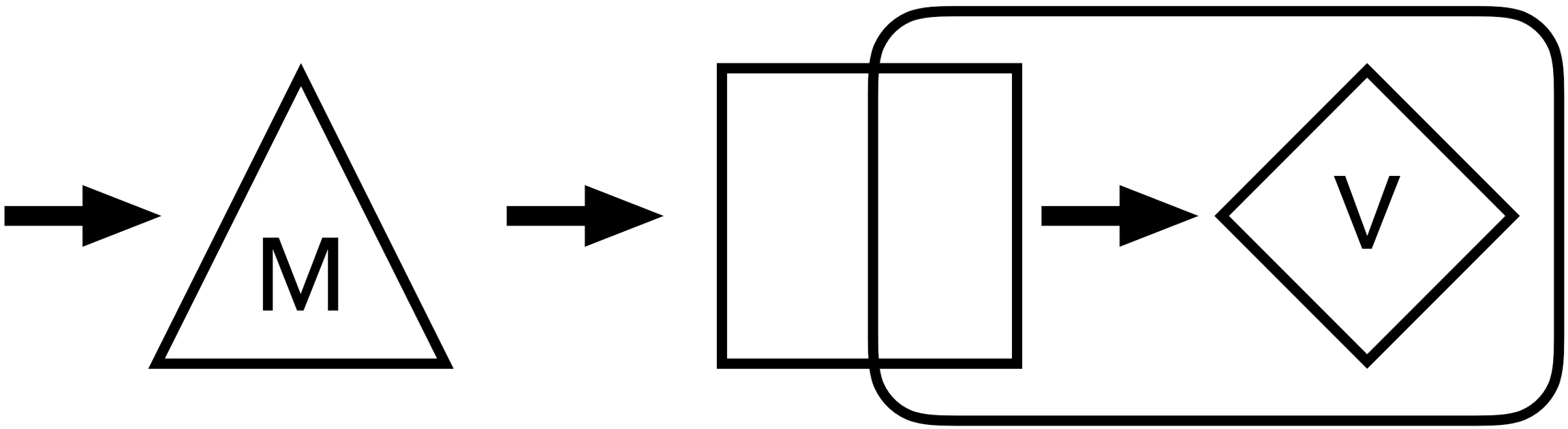
FACEBOOK'S REACT

- React Library
  - Said to be the View in an MVC app
  - Defines a "Flux Architecture"
- Virtual DOM
- Single direction of information

REACT

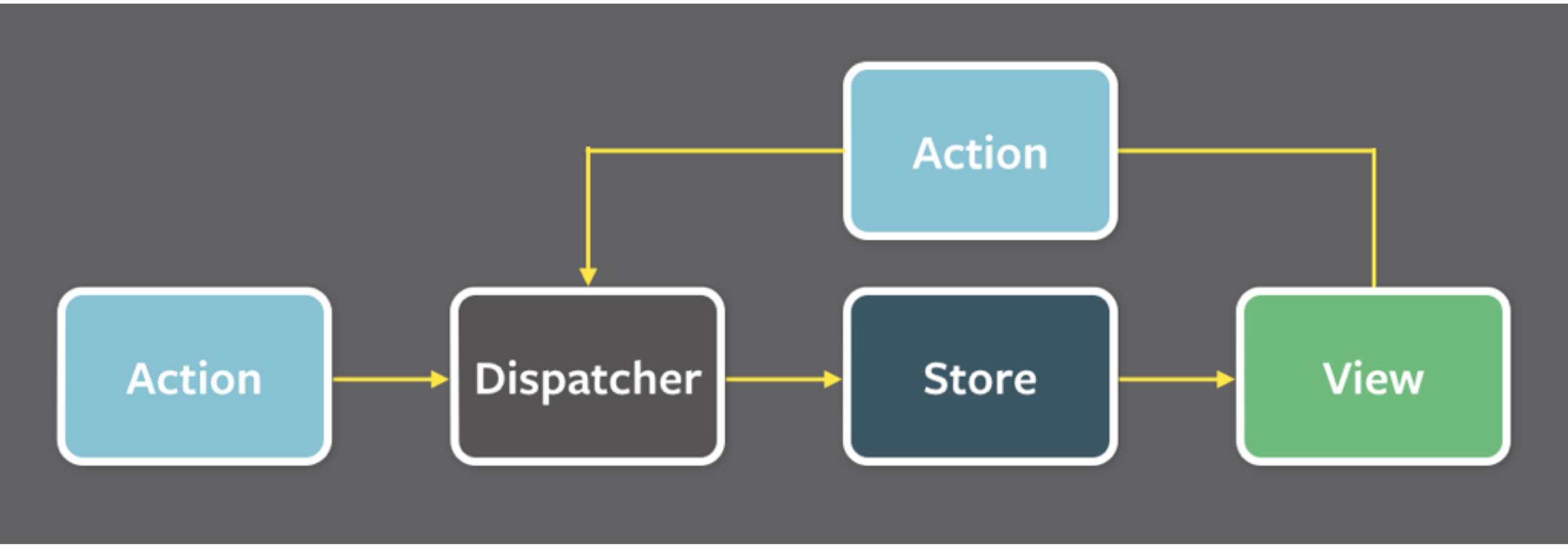


REACT



React

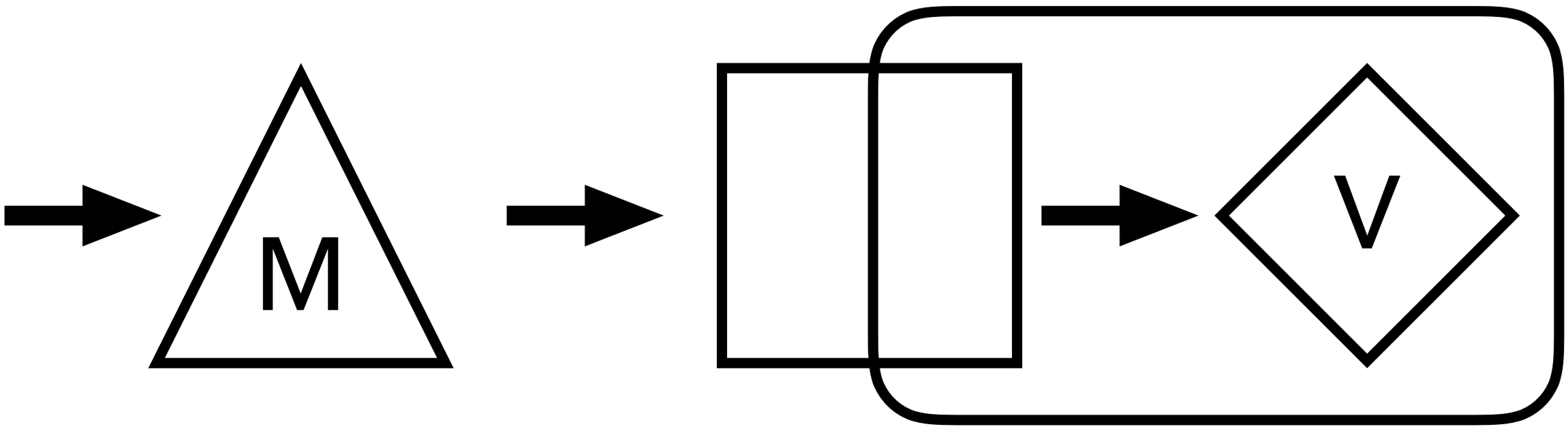
REACT



WHAT TO USE WITH REACT

- JSX - Syntactic sugar to declare HTML
- Redux & React-Redux (Reducer/Accumulator)

REACT



Redux

& React-Redux

React & JSX

WHAT TO USE WITH REACT

- Redux Thunk (Asynchronous Actions)
- Fetch Polyfill (XMLHttpRequest Replacement)
- Redux Forms (HTML form handling)
- Redux Logger

TESTING WITH REACT

- Browser vs Node
- Browser: Karma with Jasmine or Mocha
- Browser: Webdriver.io
- Node: Jest

WHO CONTROLS THE DOM?

- Django API
  - Easy
- Django Templates
  - but React owns the DOM
  - Precomputing React?
  - Webpack bundles? (for cache busting)
  - Watch Julien Phalip's talk about React and Django

JAVASCRIPT'S PROBLEM IS THE DOM

- Frameworks seek to provide a mental framework to allow developers to avoid the perils of modifying a global variable that also counts as the structure and display of the app
- Appears to me to be an increasing tendency to provide a single direction, instead of two-way bindings
- However, still good reasons to use two-way binding frameworks, as they may provide better solution for your interface

AVOID ALL THIS CONFIG

- create-react-app
- Yarn compatible, Webpack & Babel, ESLint, and Jest
- JSX and React
- Redux must be installed

WHAT'S COMING NEXT?

- Performance: InfernoJS and Svelte
- React Fiber
- But what of React patents and OSS?
- Vue JS?

@andrewsforge

http://andrewsforge.com

THANK YOU!