

# UI Development with React

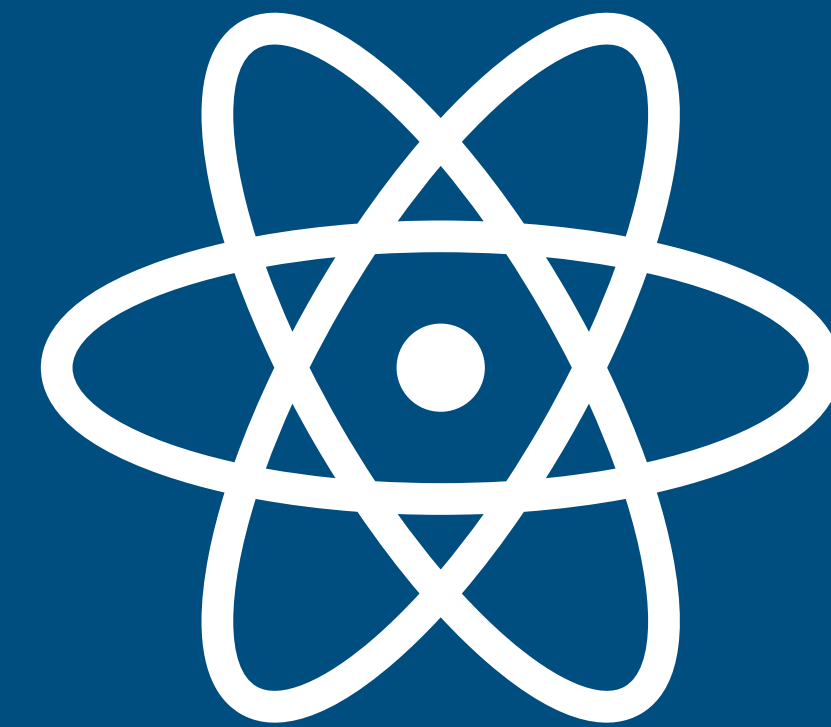
 @pavanpodila

Senior Director, Technology

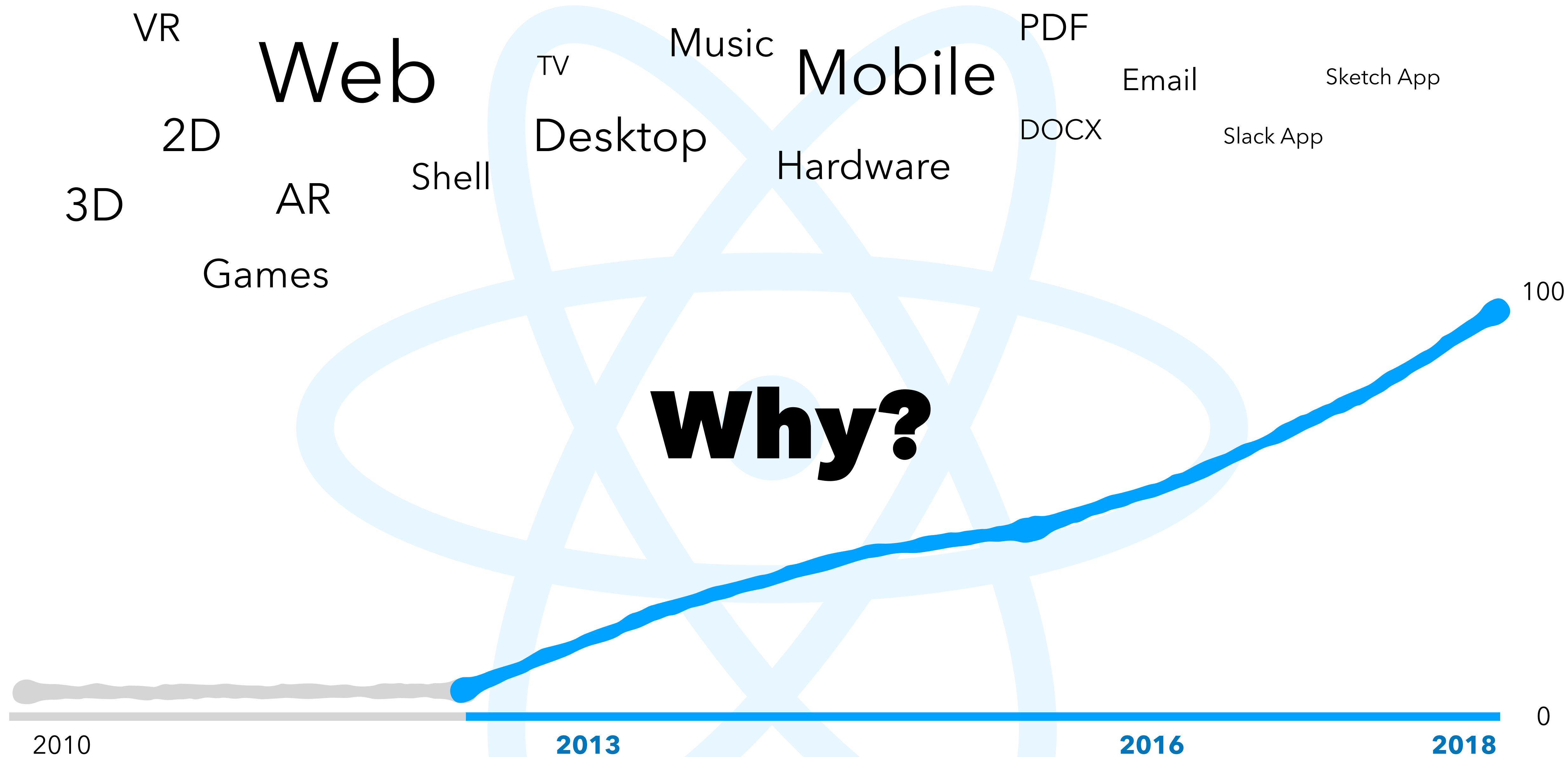
**PUBLICIS.SAPIENT**



# Philosophy



<Code />



```
const tick = new Howl({
  src: [tickSound],
});

class TickSound extends React.Component {

  render() {
    tick.play();

    return null;
  }
}

export function AnalogFace2() {
  const strapColor = '#ff3987';

  return (
    <section style={{
      flex: 1, display: 'flex',
      justifyContent: 'center',
      alignSelf: 'stretch', alignItems: 'center',
      background: 'linear-gradient(#eee, #aaa)'
    }}>
      <WatchShell strapColor={strapColor} rimColor={'#940038'} faceColor={'#111'}
        buttonColor={'#777'}>
        {
          ({ config, time }) => {
            let hr = time.hours();
            const hours = hr >= 12 ? hr - 12 : hr;
            const minutes = time.minutes();
            const width = config.width / 2, config.height / 2 - config.rim
```

# Text



<!DOCTYPE html>

<html>

<head>

<title>Example</title>

<link rel="stylesheet" href="style"

</head>

<body>

Behavior

JavaScript

XML

*Style*

CSS

<a href="/">Header</a>

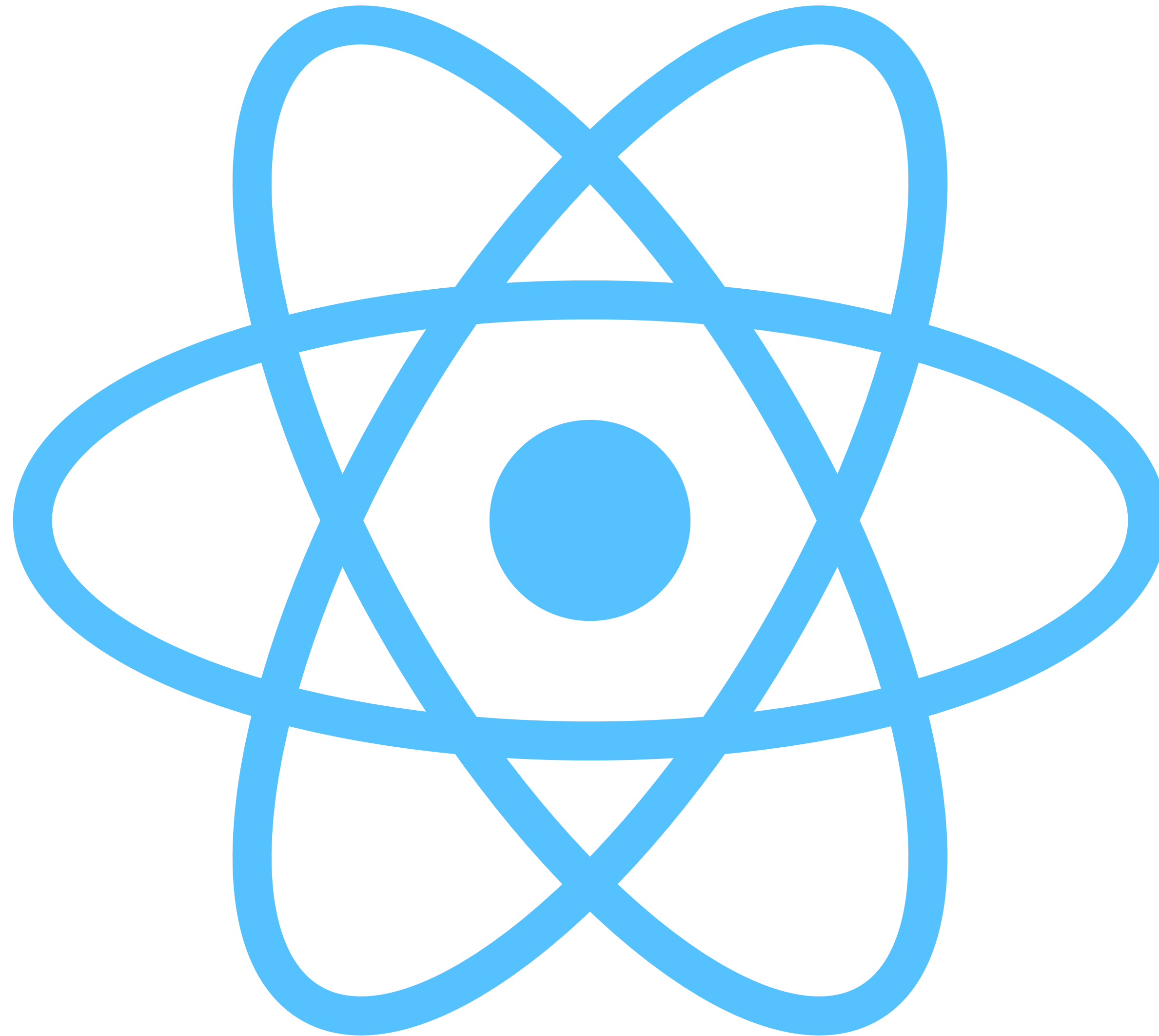
</h1>

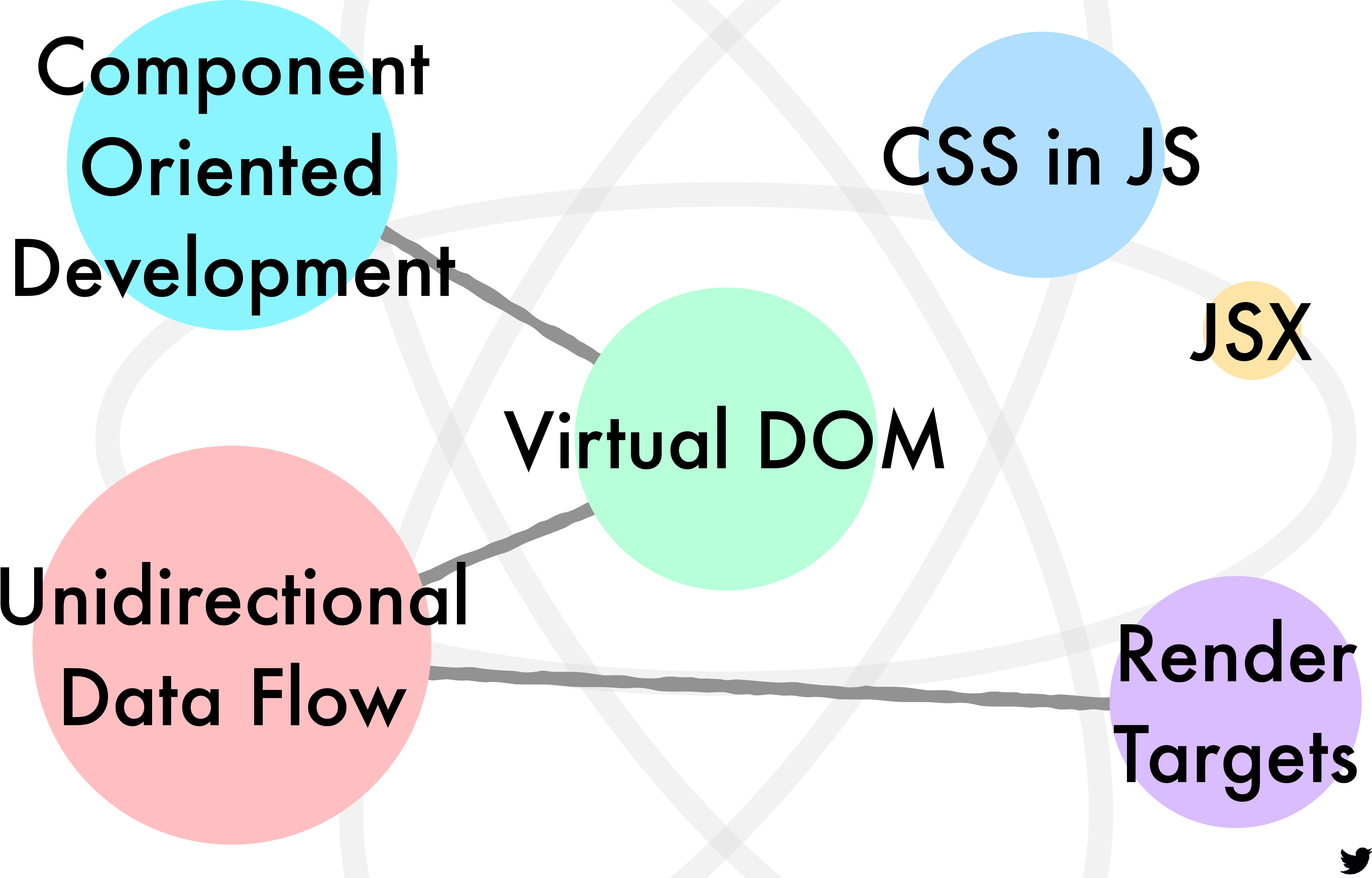
# React

XML

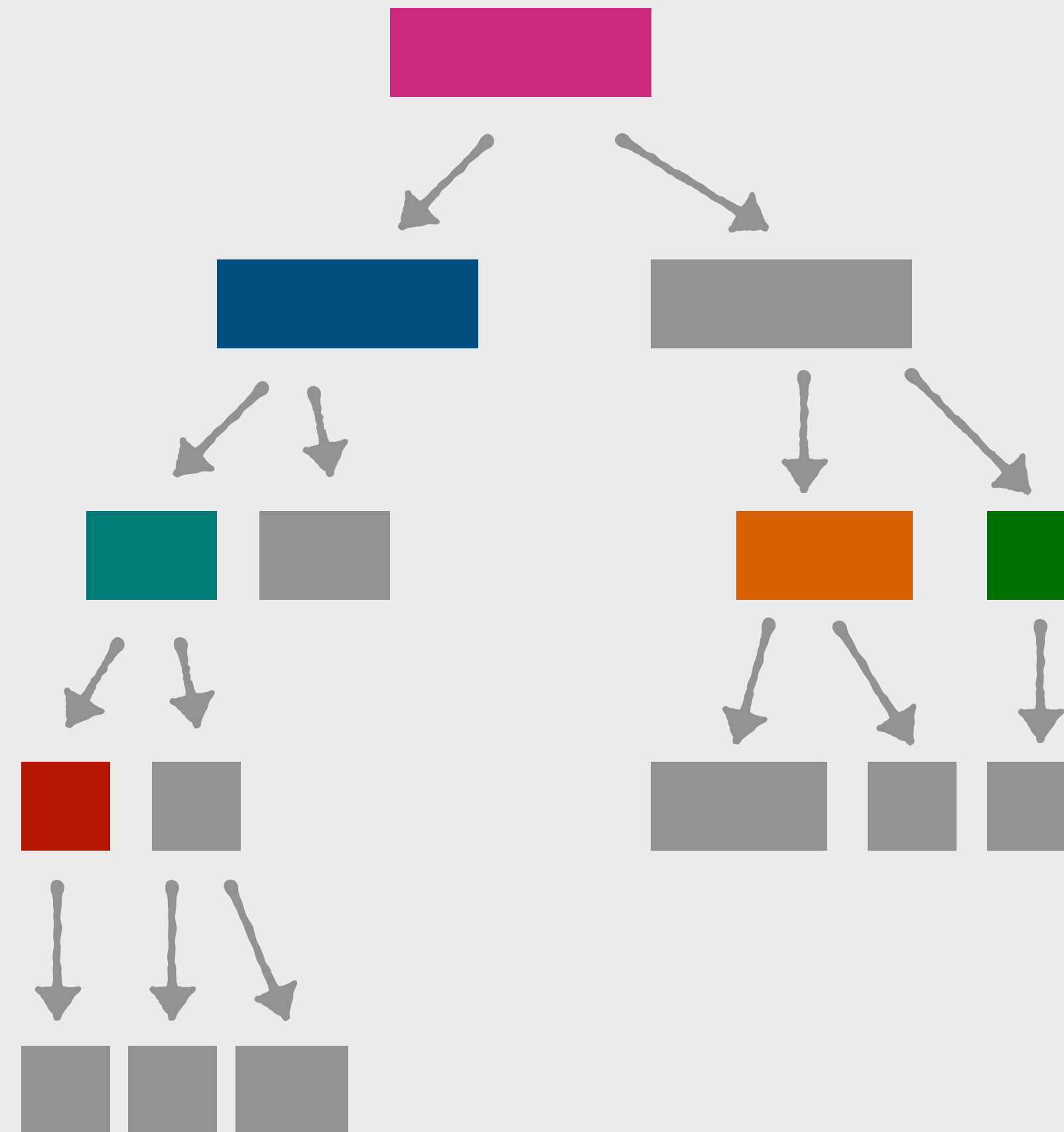
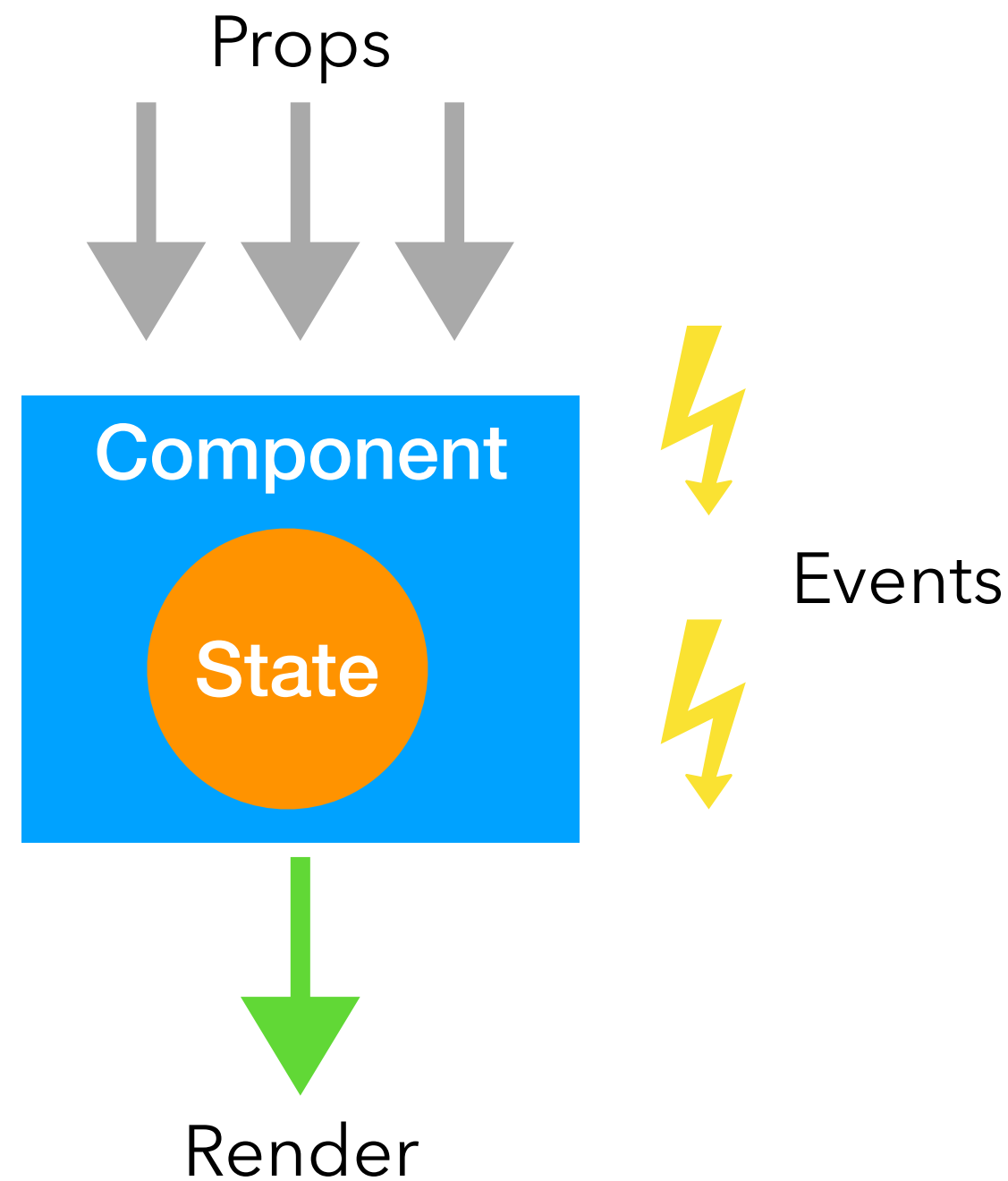
JavaScript

CSS





# Component Oriented Development



Higher Order Component

Render Prop

Compound Component

Controlled Component

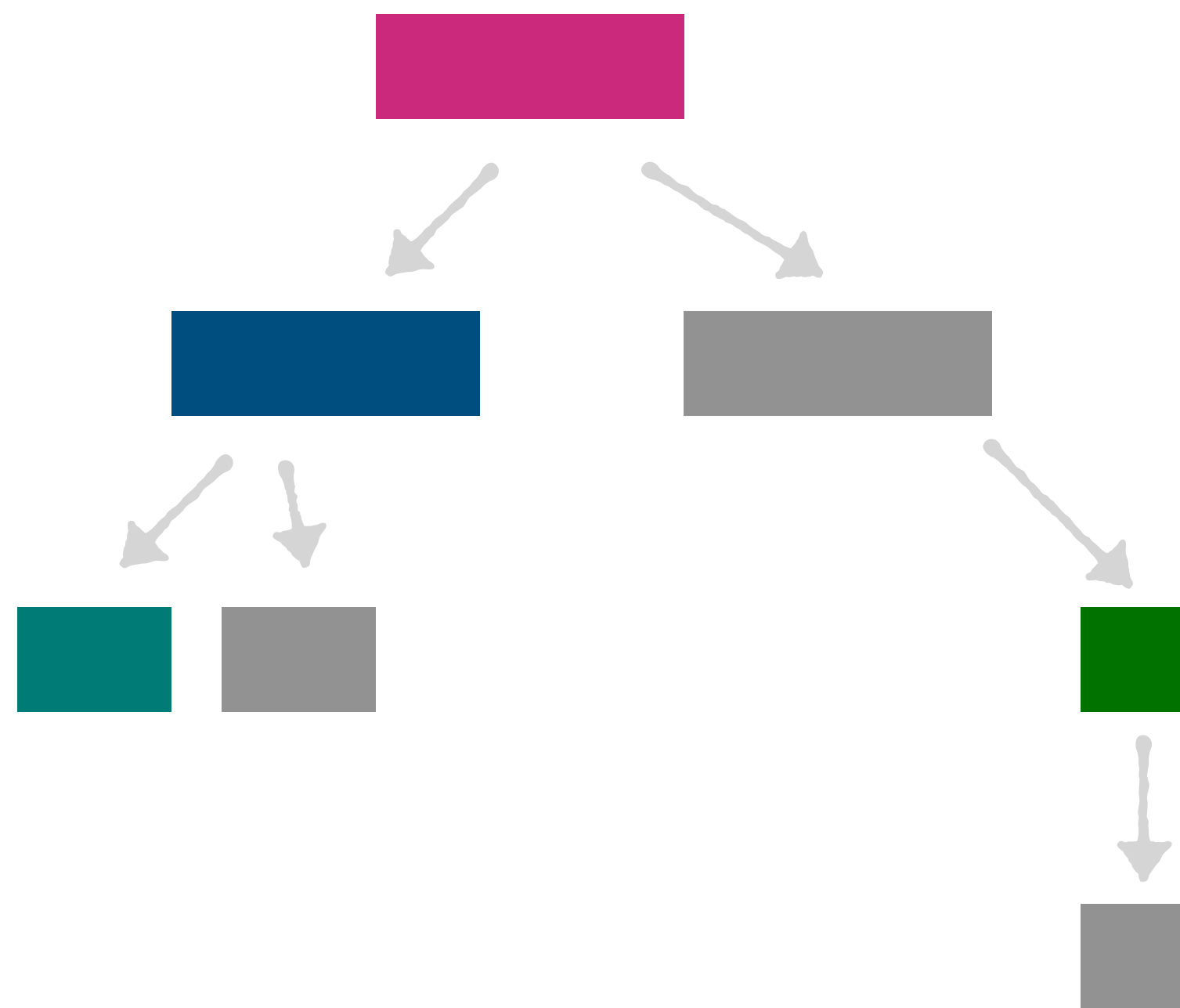
Domain / Service Component

Container / Presentation

Provider / Consumer



# Virtual DOM

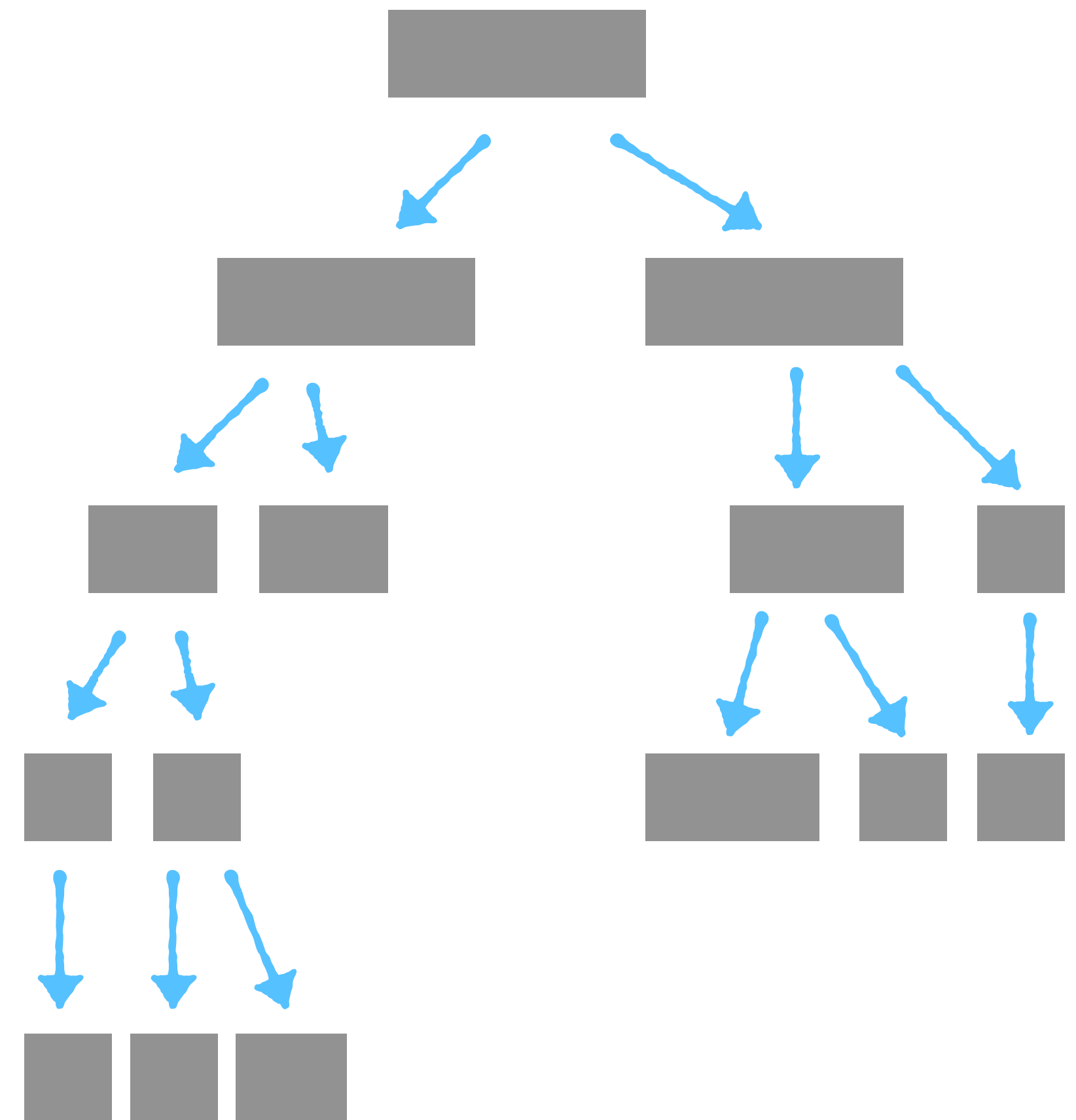


Component Tree

diff + patch

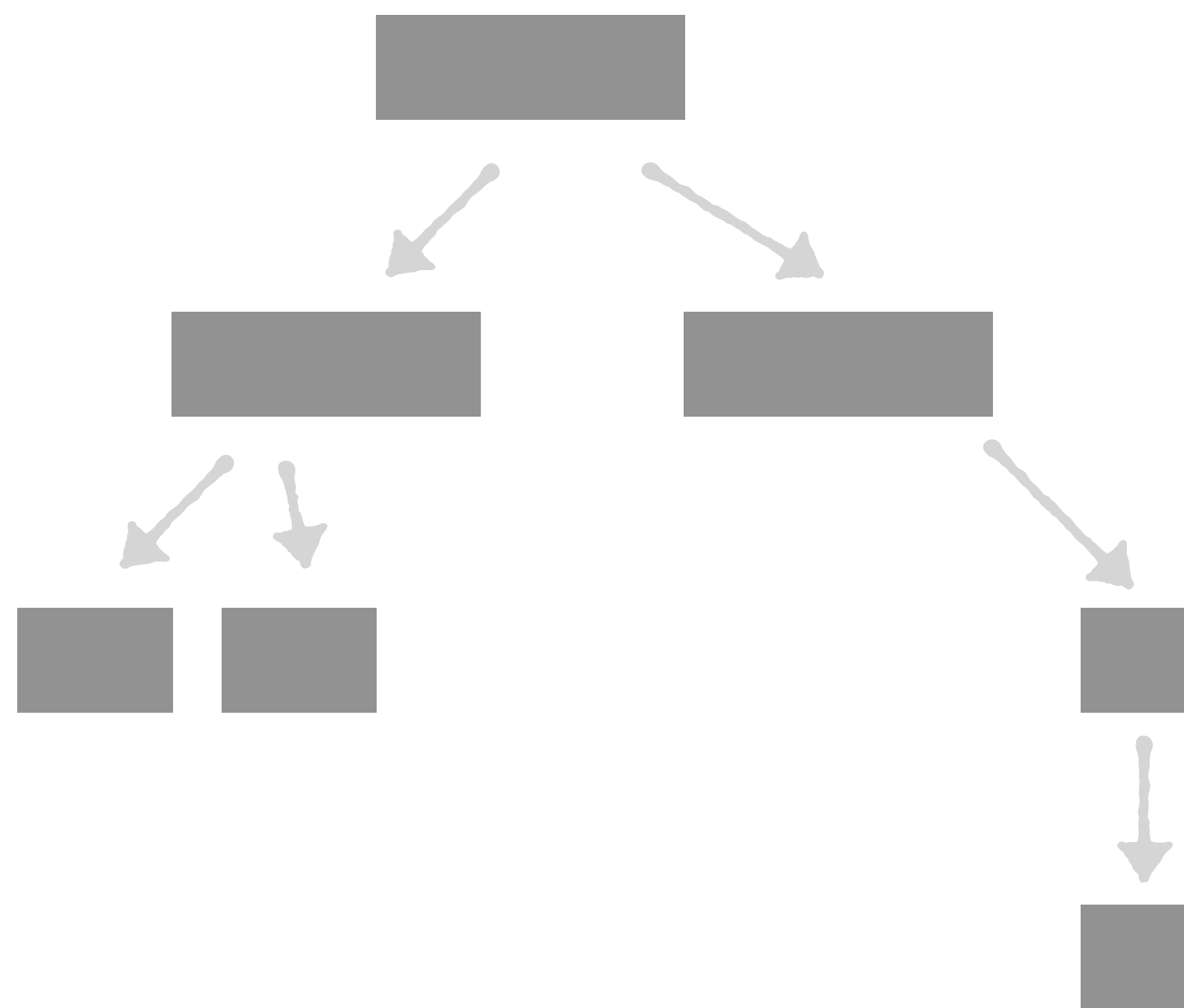


reconciliation



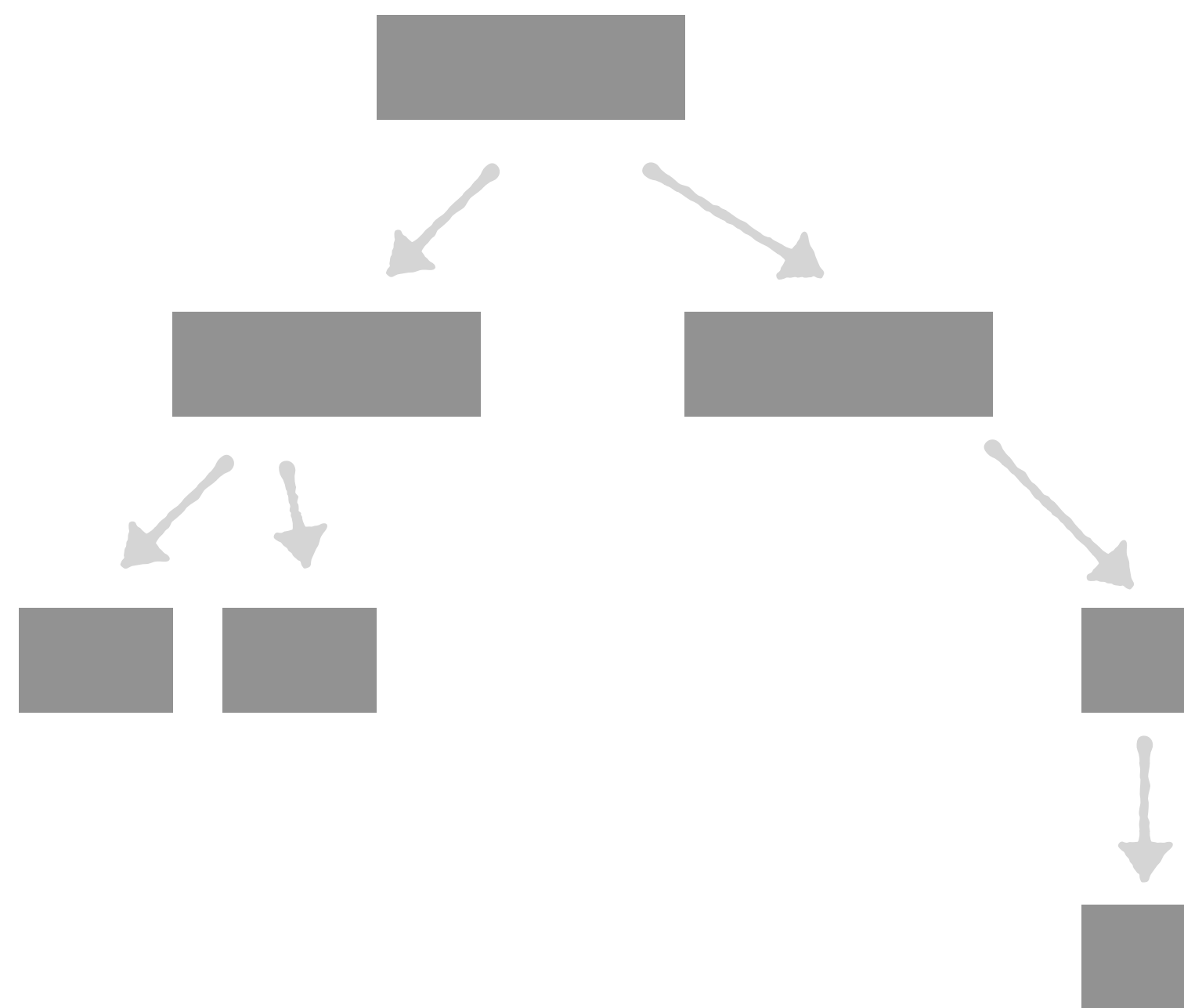
HTML DOM

# Virtual DOM

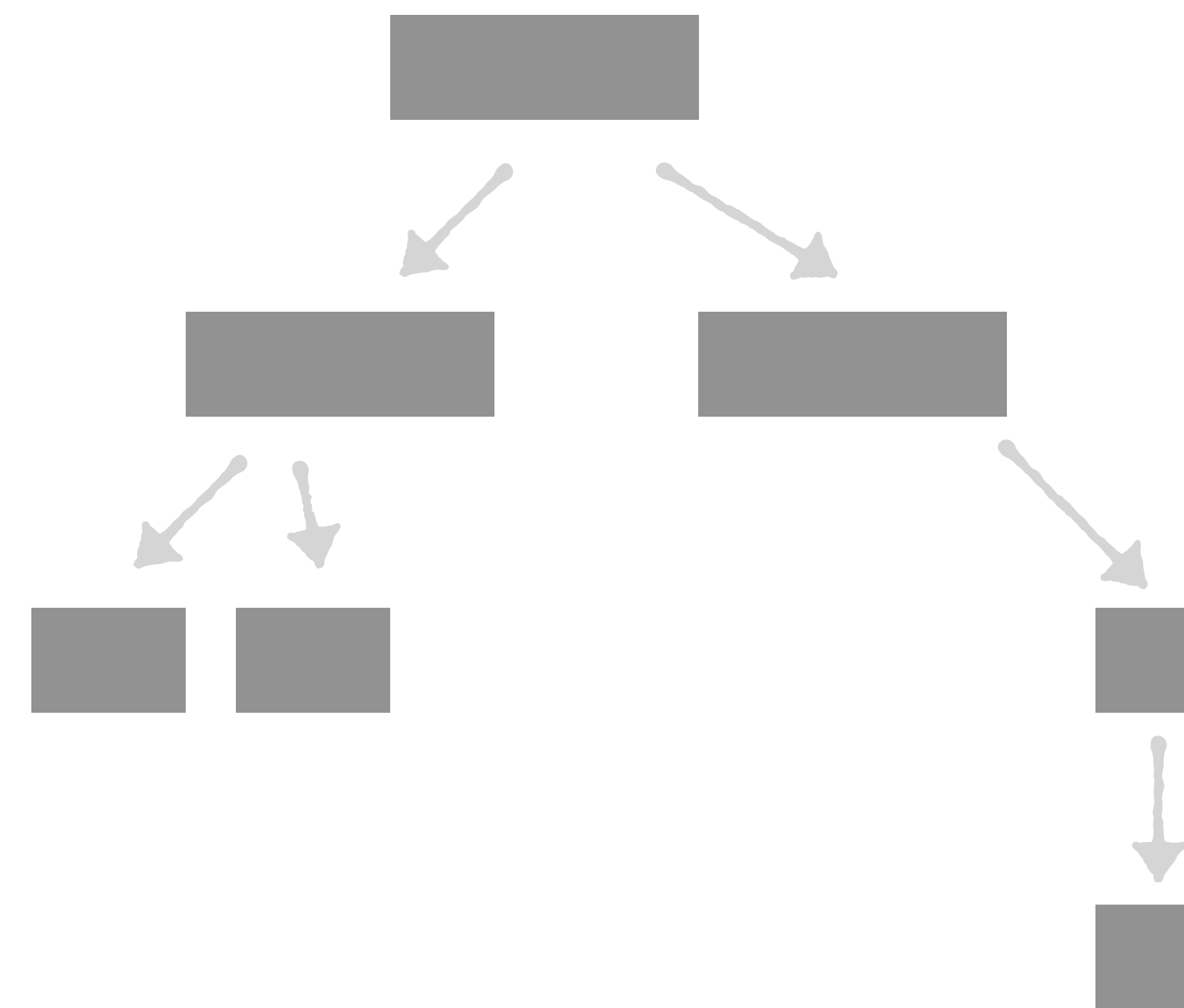


Render #1

# Virtual DOM

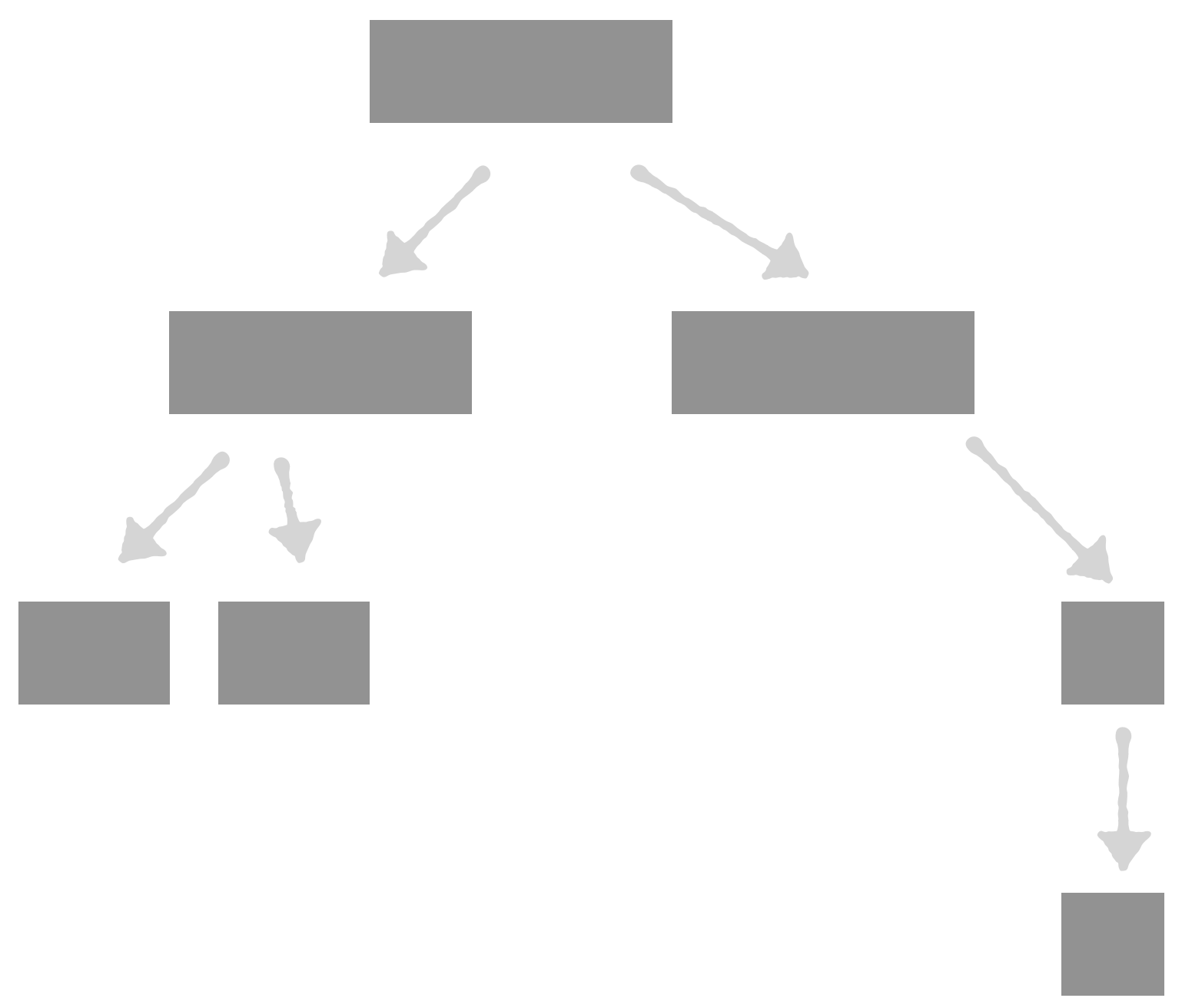


Render #1

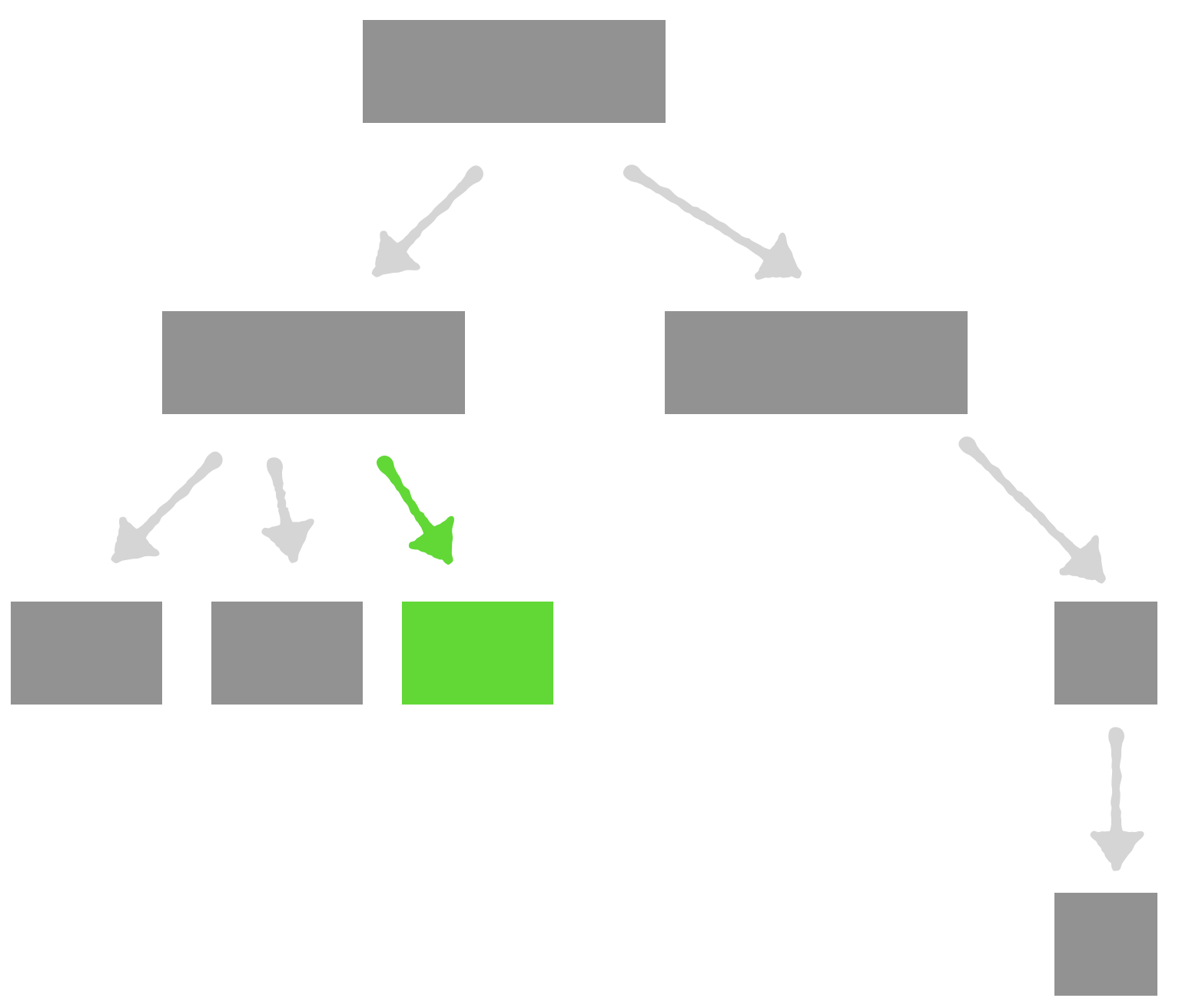


Render #2

# Virtual DOM

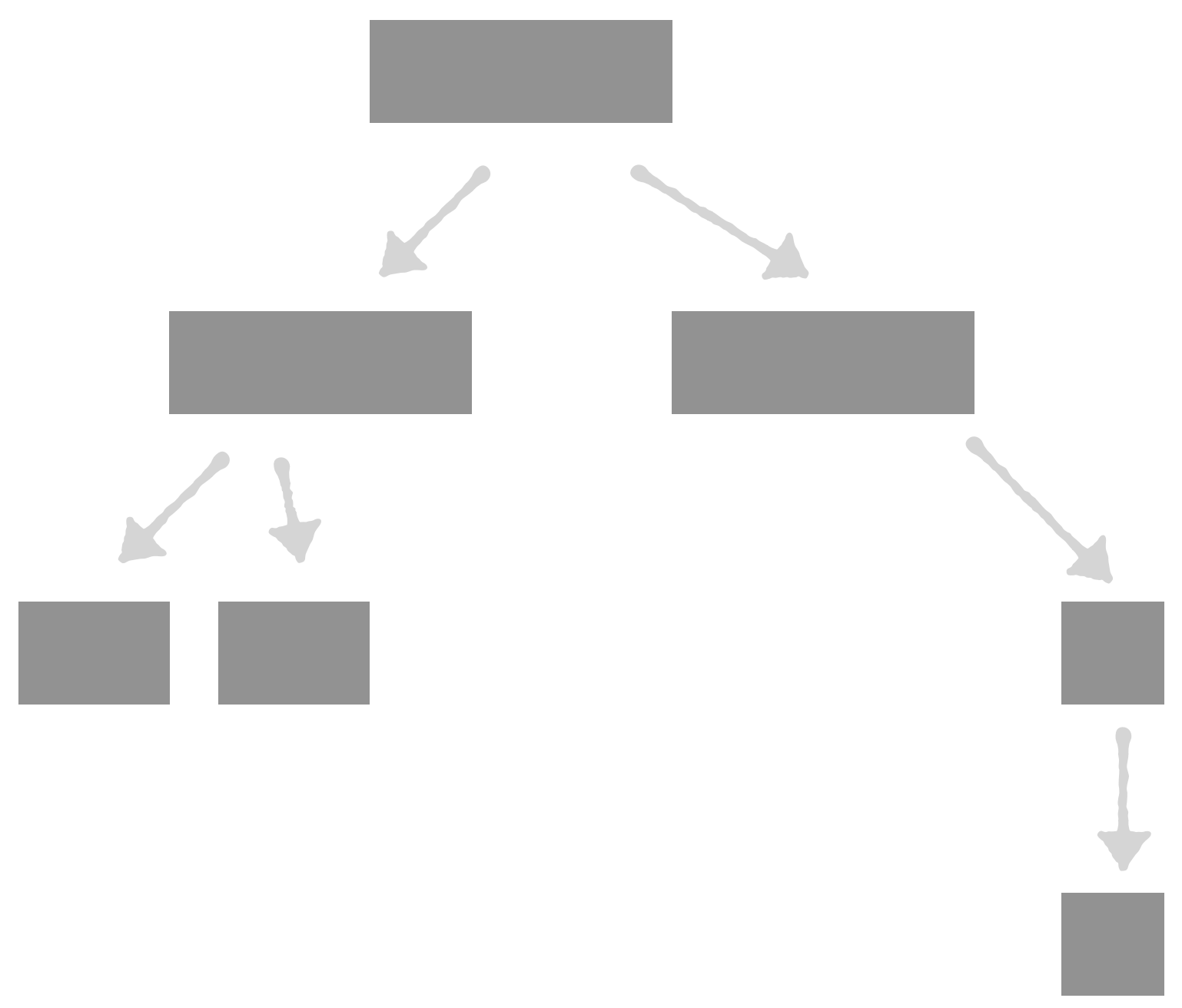


Render #1

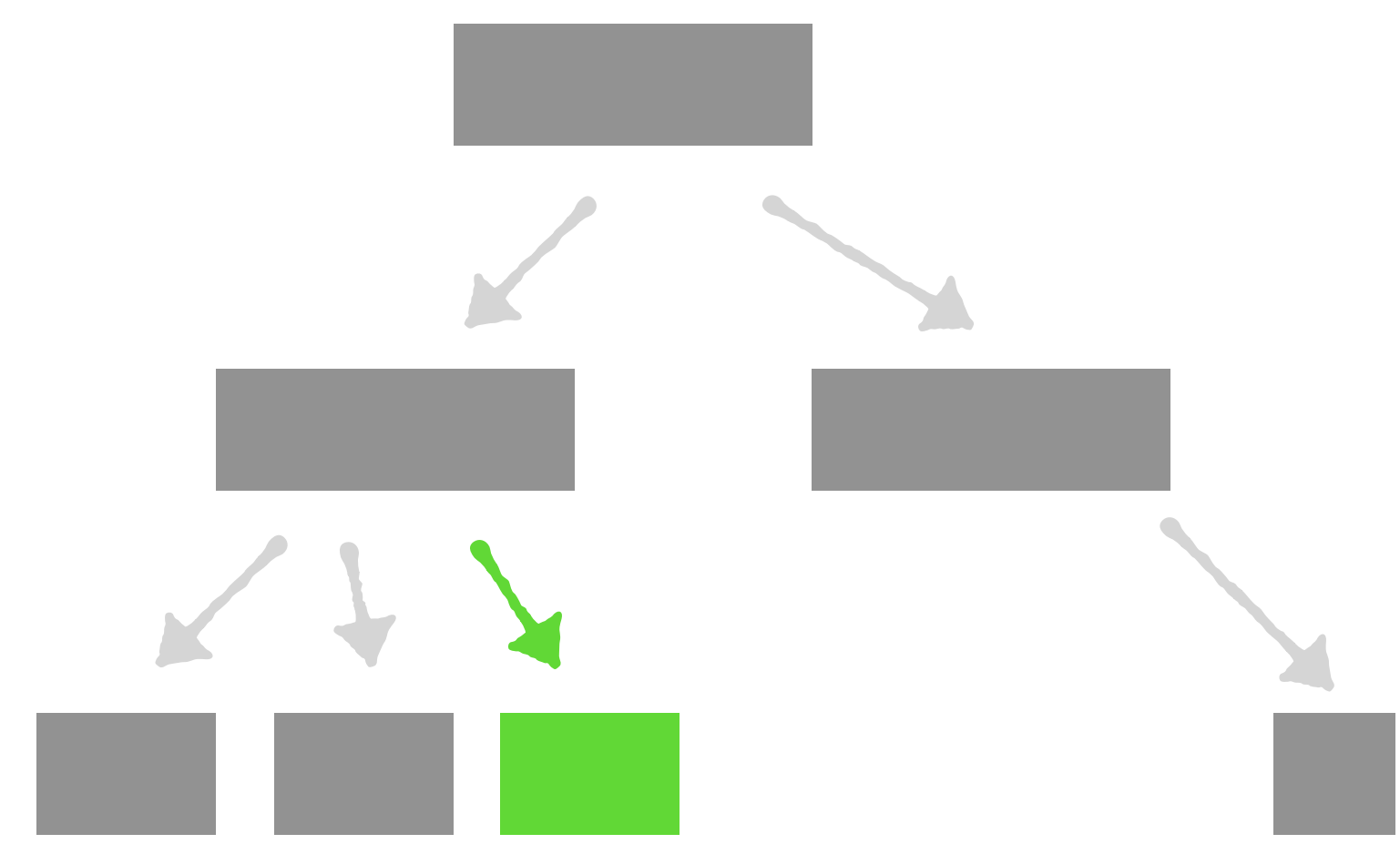


Render #2

# Virtual DOM



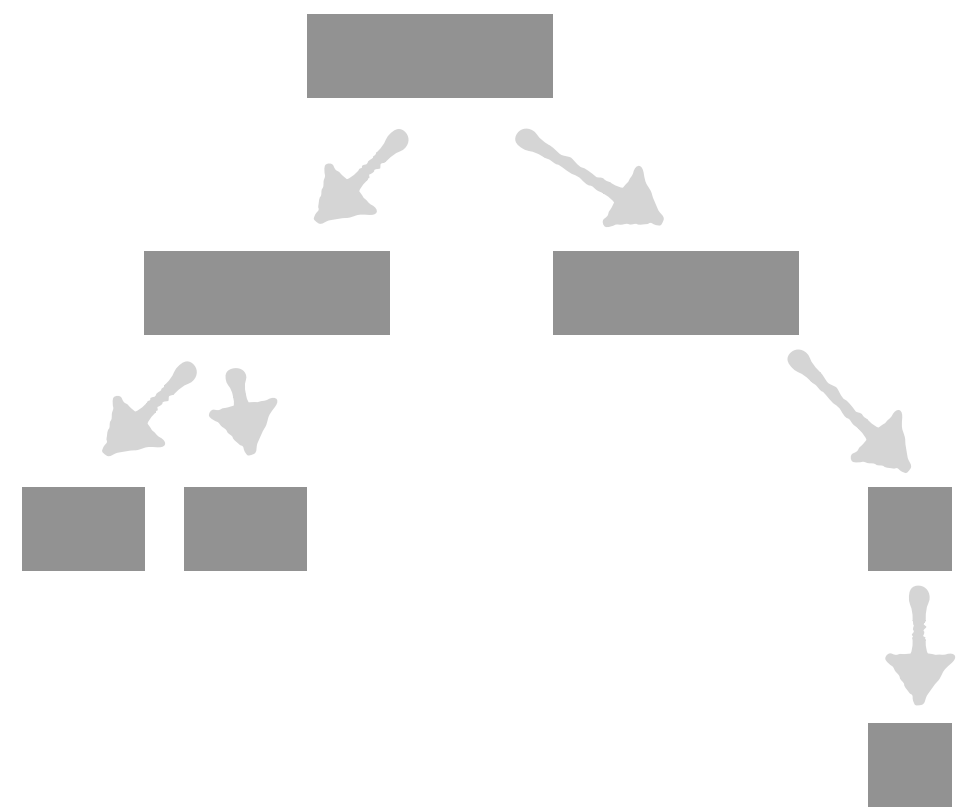
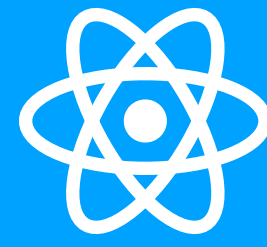
Render #1



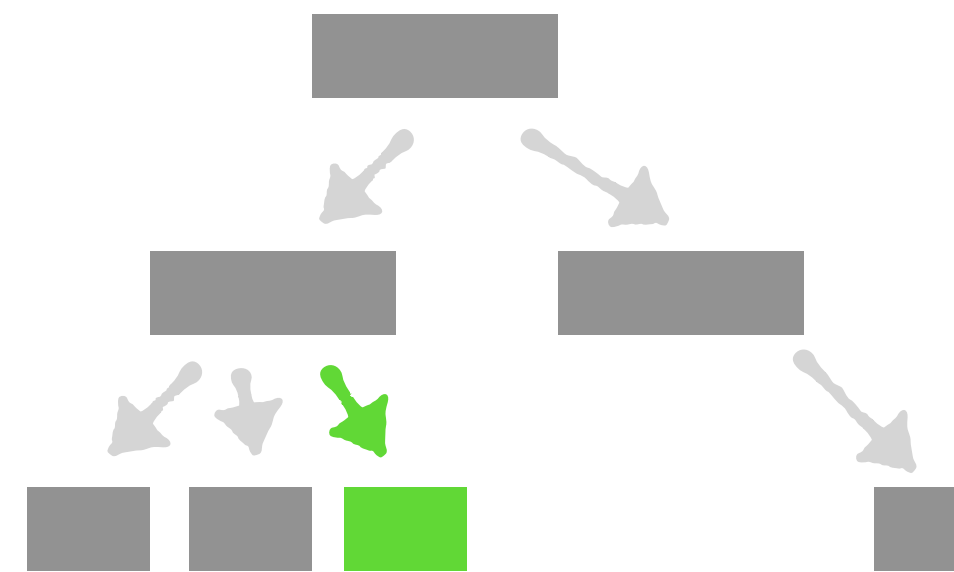
Render #2



# Virtual DOM

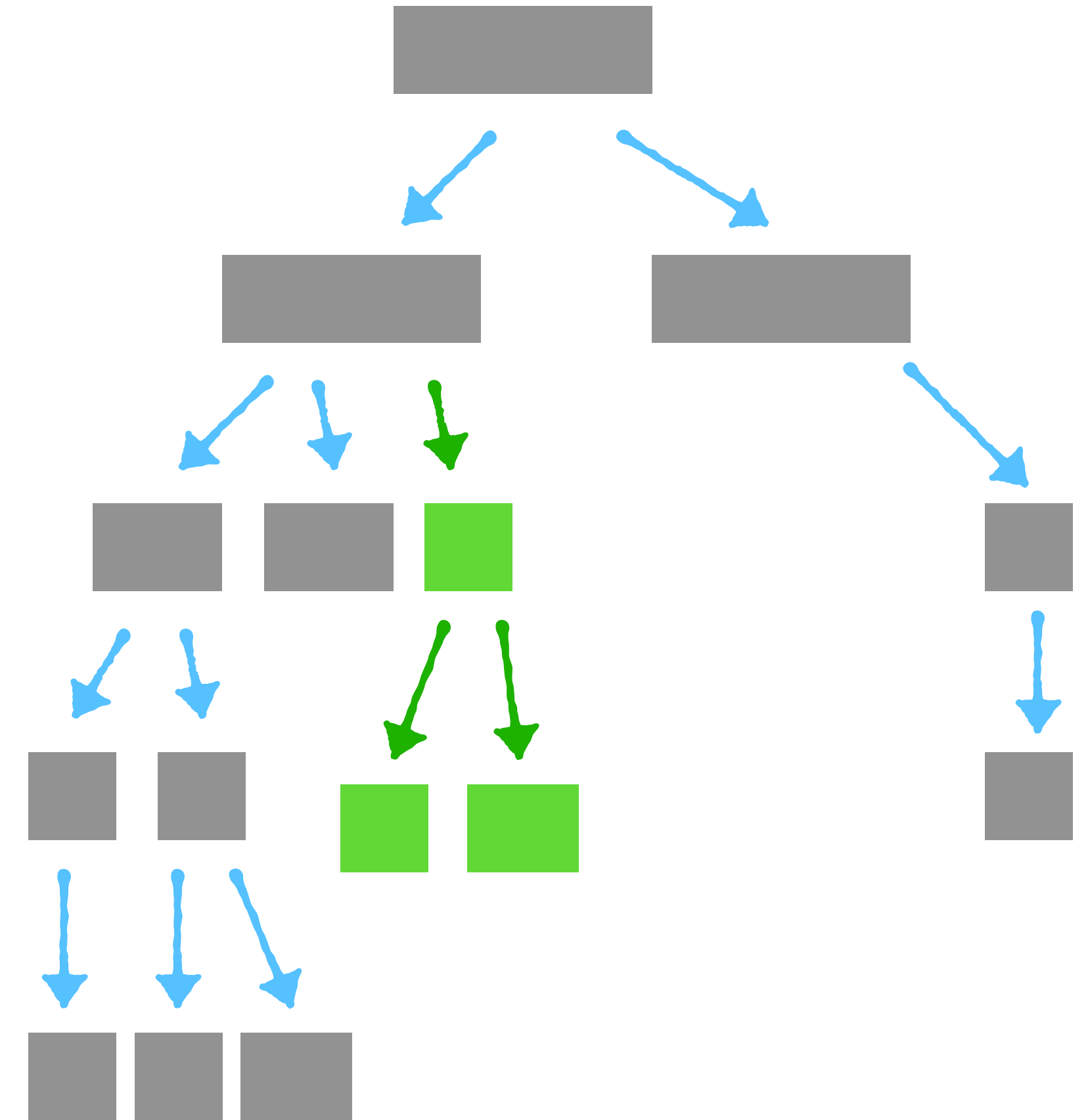


Render #1



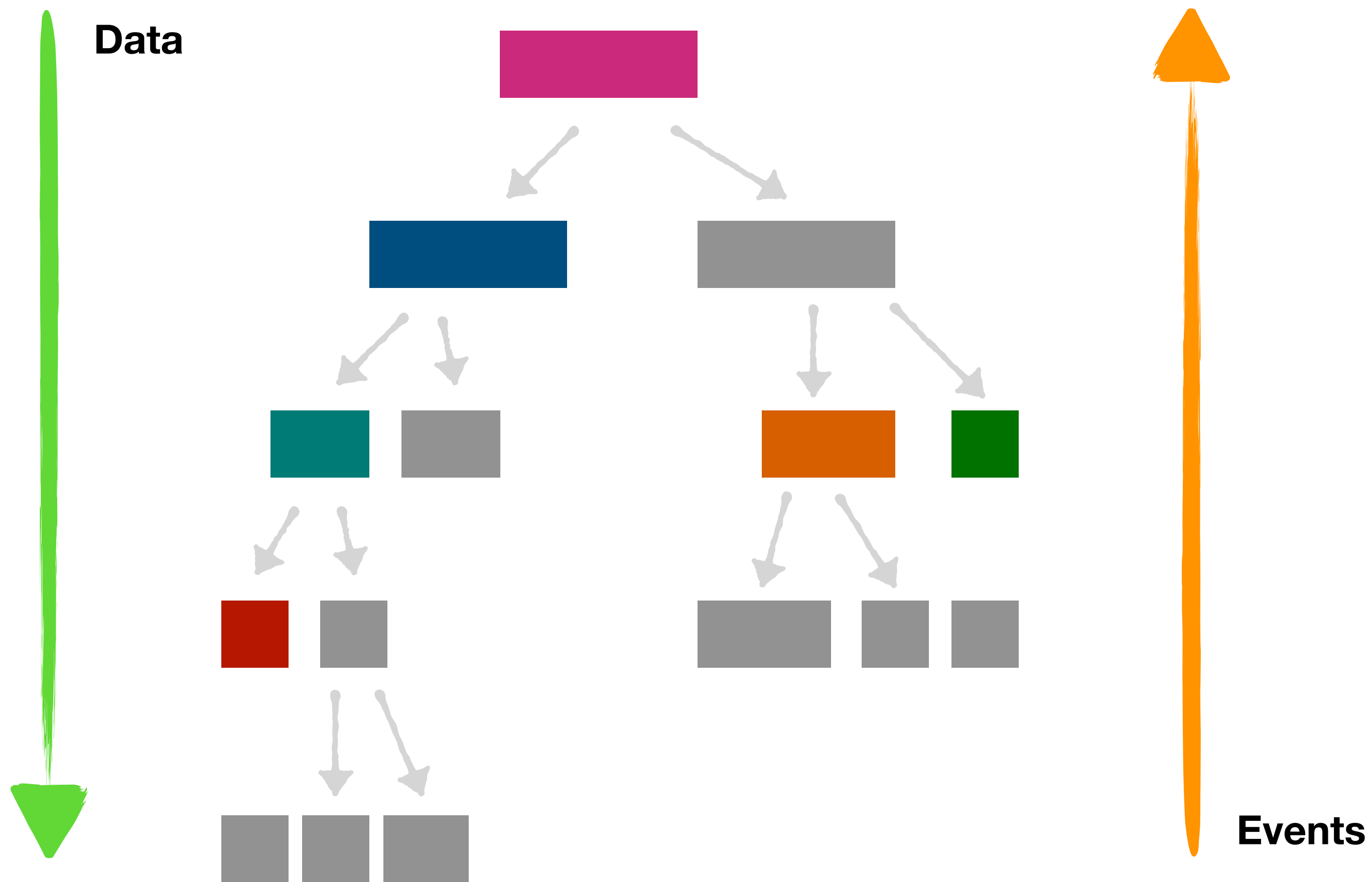
Render #2

reconciliation



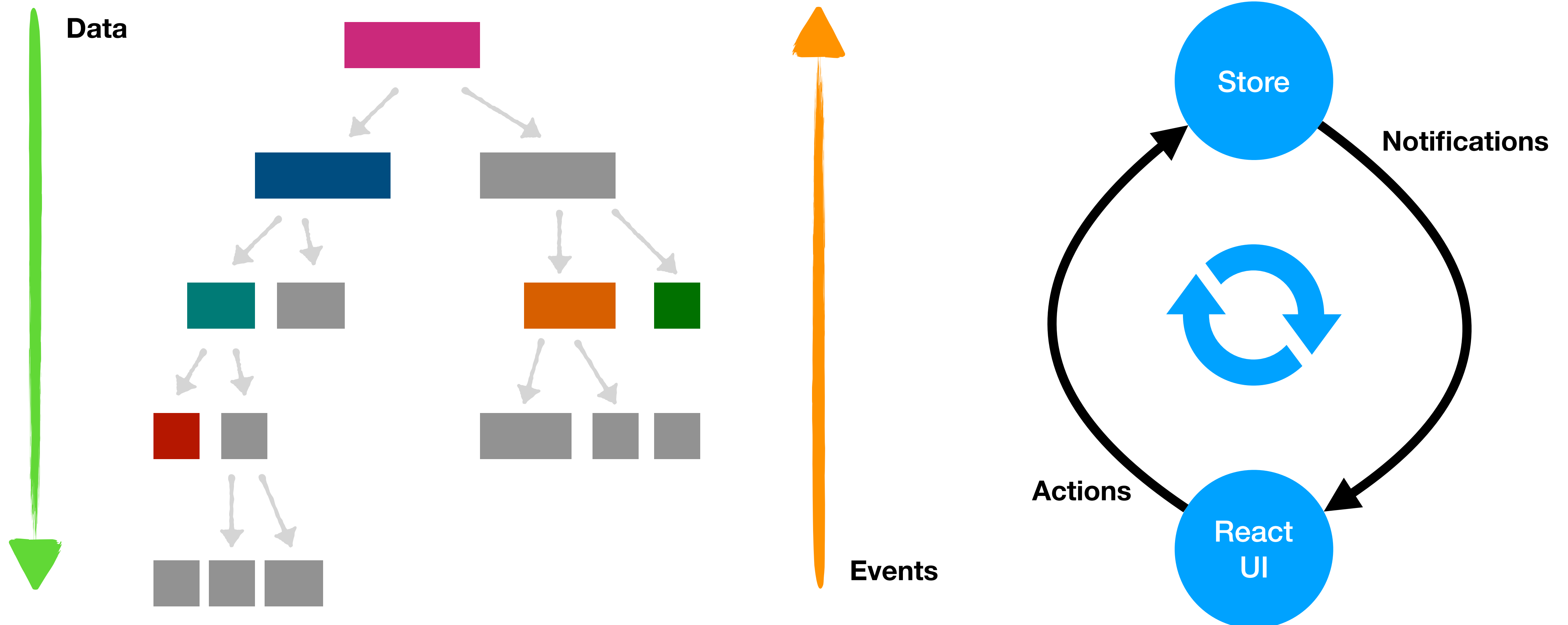
HTML DOM

# Unidirectional Data Flow



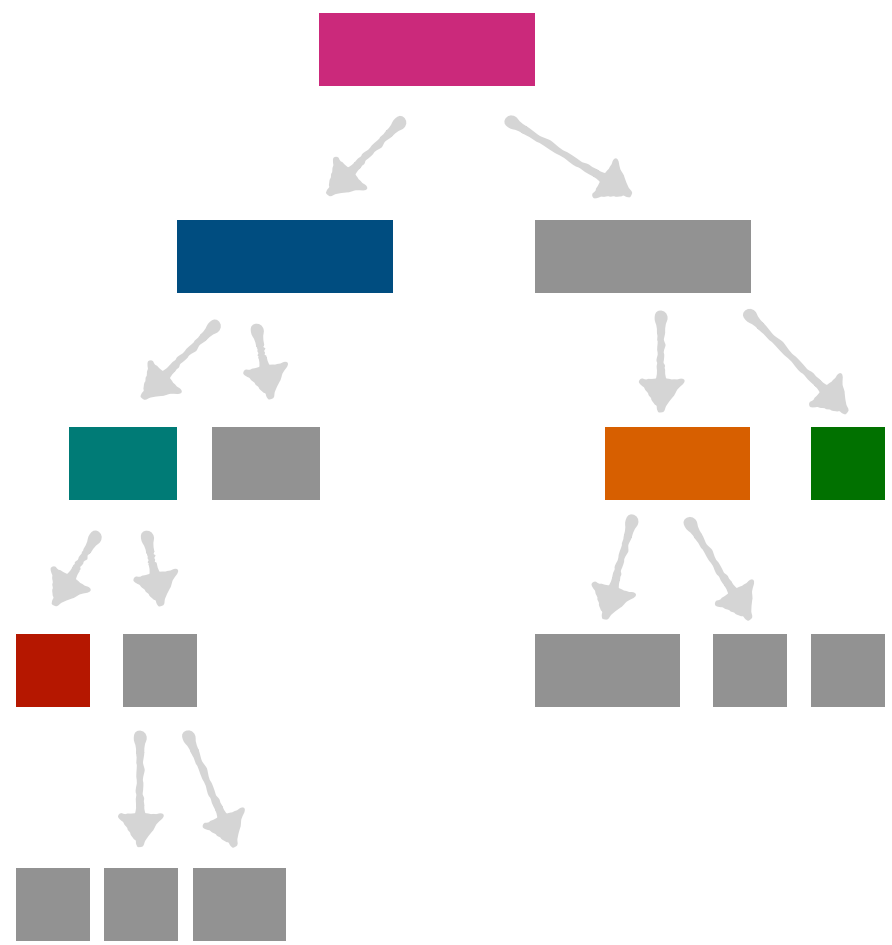
Data down, Events up

# Unidirectional Data Flow

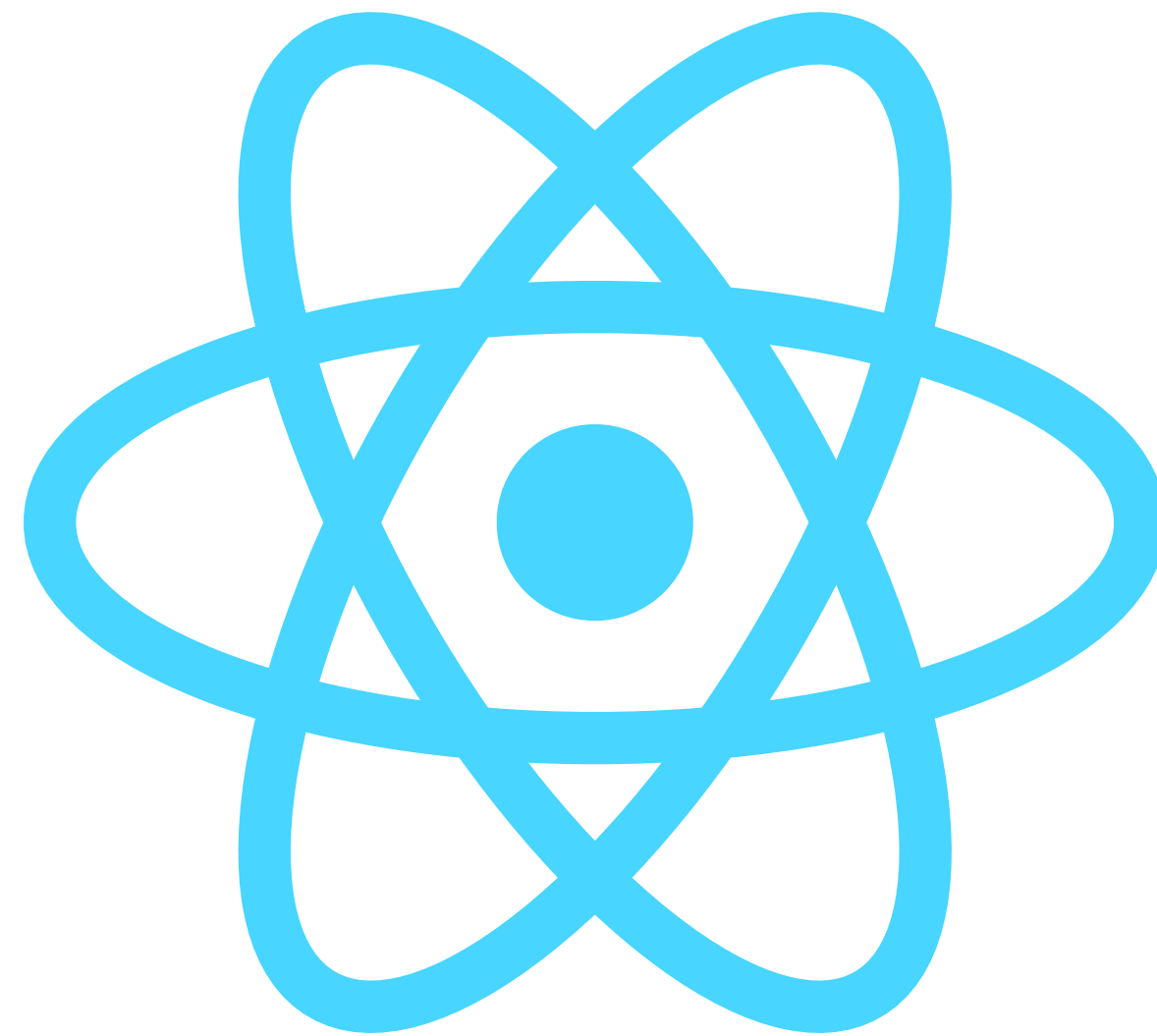


Data down, Events up

# Render Targets



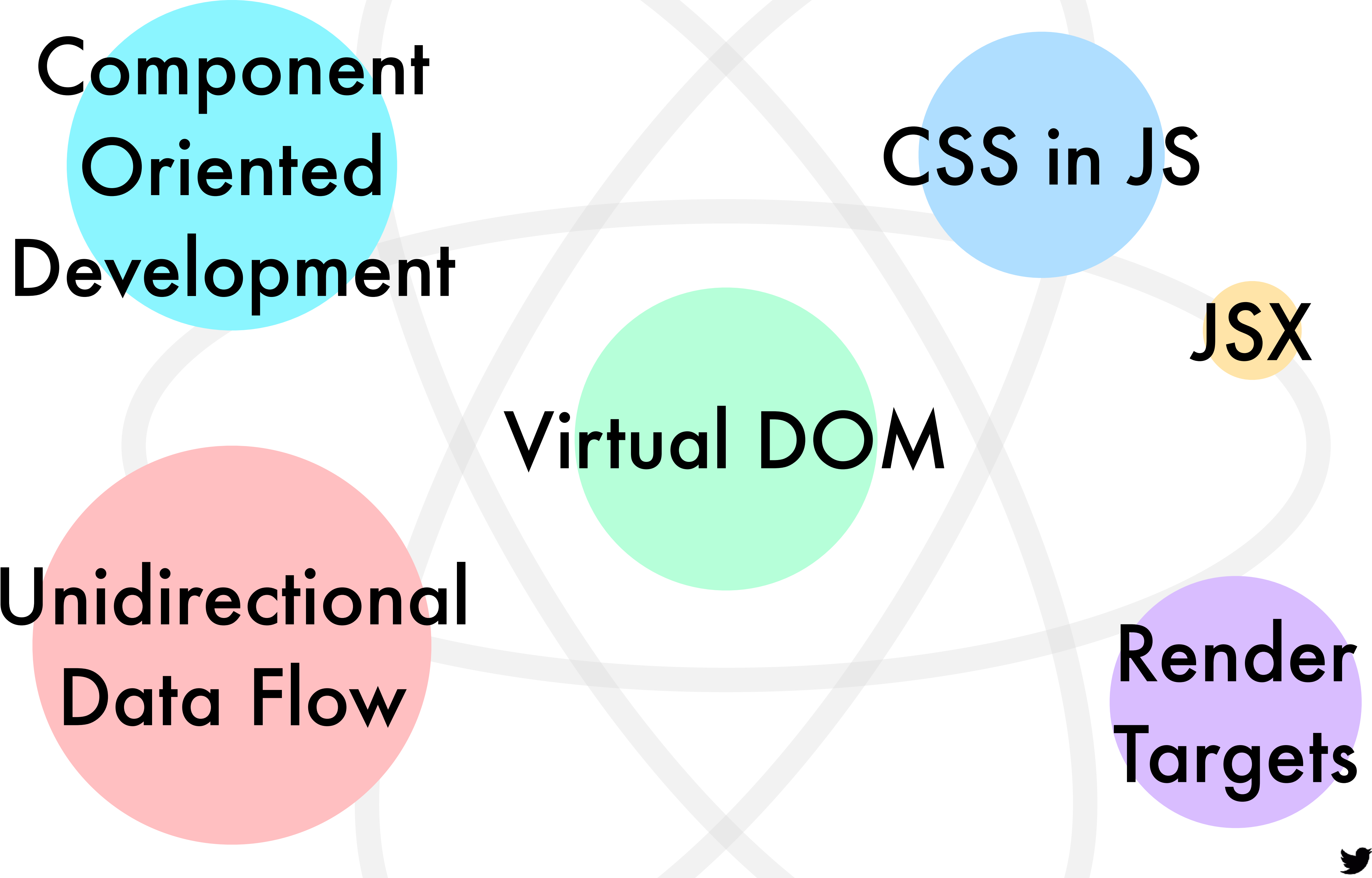
Virtual DOM



render



- HTML DOM
- Mobile (React Native)
- VR
- 3D
- Canvas
- Server Side Rendering
- Hardware
- Desktop
- Word
- Email
- PDF
- Sketch

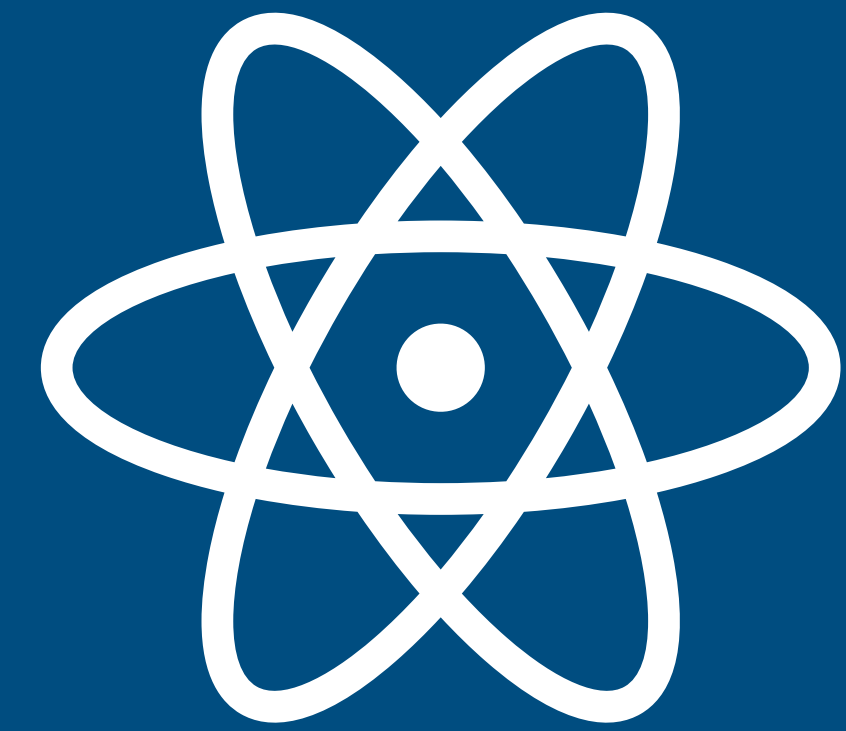




The React logo, a blue stylized atom with a central circle and three intersecting elliptical orbits, is positioned on the left side of the slide.

# UI Description Language

“Learn once,  
Write anywhere”



<Code />



## Practices

JavaScript, TypeScript

CSS, Styled Components, Glamorous

Component Driven Development, SOLID, DDD,  
Functional Programming

Patterns: HOC, Render Props, Producer-Consumer,  
Container-Presentation, Controlled,  
Domain/Service Components

## State Management

setState, React Context

Redux +  
Middlewares(redux-observable, redux-saga)

MobX  
(mobx, mobx-react, mobx-react-devtools)

## Tools

create-react-app

Jest Prettier

VS Code

WebStorm

React DevTools

## Libraries

React Router

React Loadable

Material UI

Ant Design

```
import React from 'react';
import { Clock } from '../core/components/Clock';
import { clockStore } from './clock.store';

export function ClockPage() {
  const store = clockStore;

  return (
    <div className="mt-5">
      <Clock color={'white'} format={'h:mm:ss'} store={store} />
      <Clock
        format={'MMM DD, YYYY'}
        className="display-4"
        color={'lightgray'}
        store={store}
      />
    </div>
  );
}
```

<Code with="react"/>

# Explore...

<https://reactjs.org/>

Awesome React  
Awesome React Talks

<https://theuidev.com>





