



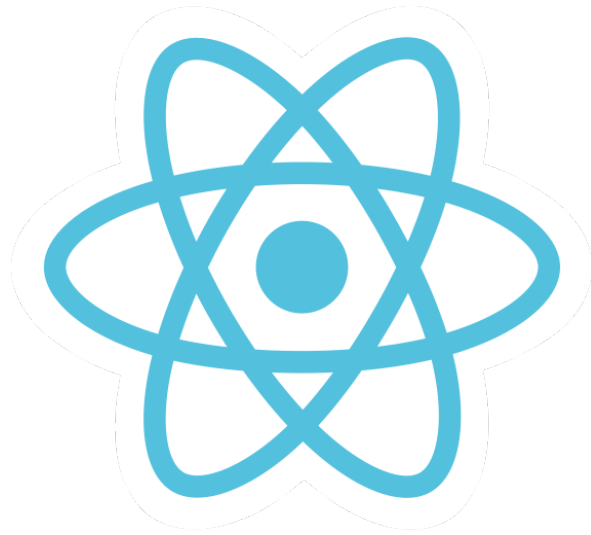
Intro to Vue.js

The Progressive
JavaScript Framework

Tom Meagher
GitHub @tmm

What is Vue.js?

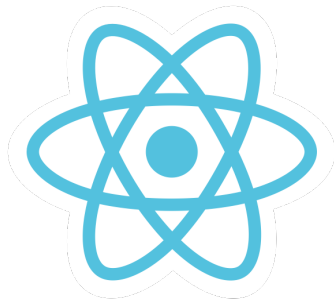
- Vue (pronounced /vjuː/, like **view**)
- **Progressive framework** for building user interfaces
- Incrementally adoptable
- Core library is focused only on the **view layer**



VS



- utilize a virtual DOM
- provide reactive and composable view components
- maintain focus in the core library, with concerns such as routing and global state management handled by companion libraries



```
render () {  
  let { items } = this.props  
  
  let children  
  if (items.length > 0) {  
    children = (  
      <ul>  
        {items.map(item =>  
          <li key={item.id}>{item.name}</li>  
        )}  
      </ul>  
    )  
  } else {  
    children = <p>No items found.</p>  
  }  
  
  return (  
    <div className='list-container'>  
      {children}  
    </div>  
  )  
}
```



```
<template>  
  <div class="list-container">  
    <ul v-if="items.length">  
      <li v-for="item in  
items" :key="item.id">  
        {{ item.name }}  
      </li>  
    </ul>  
    <p v-else>No items found.</p>  
  </div>  
</template>
```

facebook/react

A declarative, efficient, and flexible JavaScript library for building user interfaces.

Updated 4 hours ago

● JavaScript

★ 64.5k

vuejs/vue

A progressive, incrementally-adoptable JavaScript framework for building UI on the web.

javascript

vue

framework

frontend

Updated 2 hours ago

● JavaScript

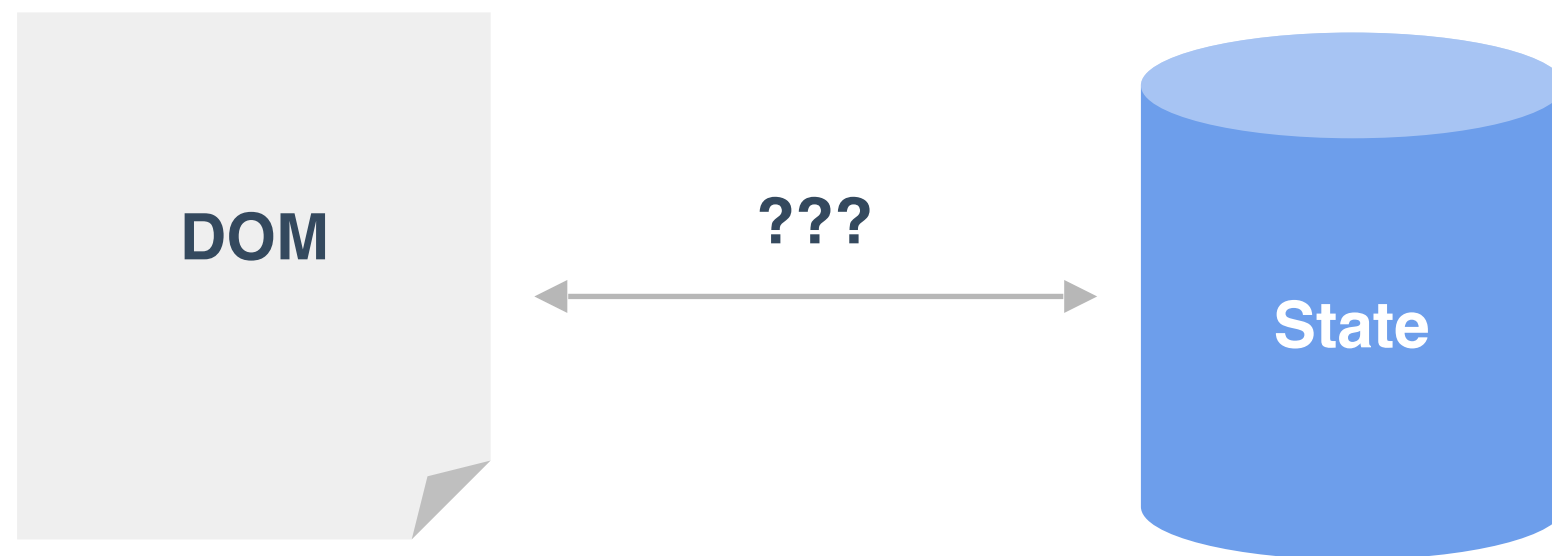
★ 50.5k

Core Concepts

**1. Declarative & Reactive
Rendering**

**2. Component Oriented
Architecture**

Declarative & Reactive Rendering



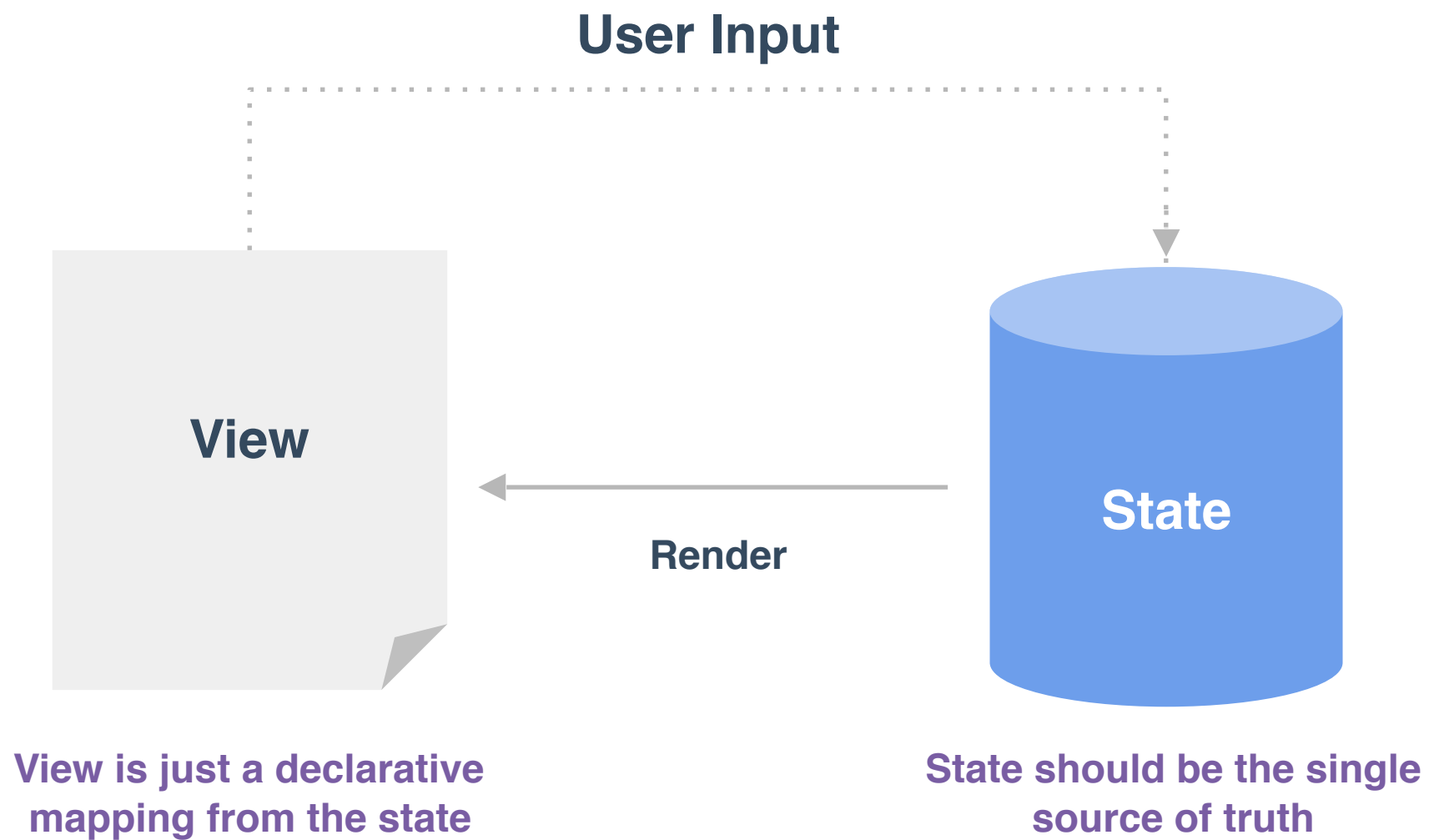


DOM

Problems with the DOM

- Re-rendering entire chunks of DOM is expensive and disruptive
- Imperatively keeping the DOM in sync with the state is tedious and error-prone

Declarative & Reactive Rendering



HTML

```
<div id="demo">  
  <h1>{{msg}}</h1>  
</div>
```

JavaScript

```
let vm = new Vue({  
  el: '#demo',  
  data: {  
    msg: 'Hello Vue.js!'  
  }  
})
```

Vue instance

Template

JavaScript

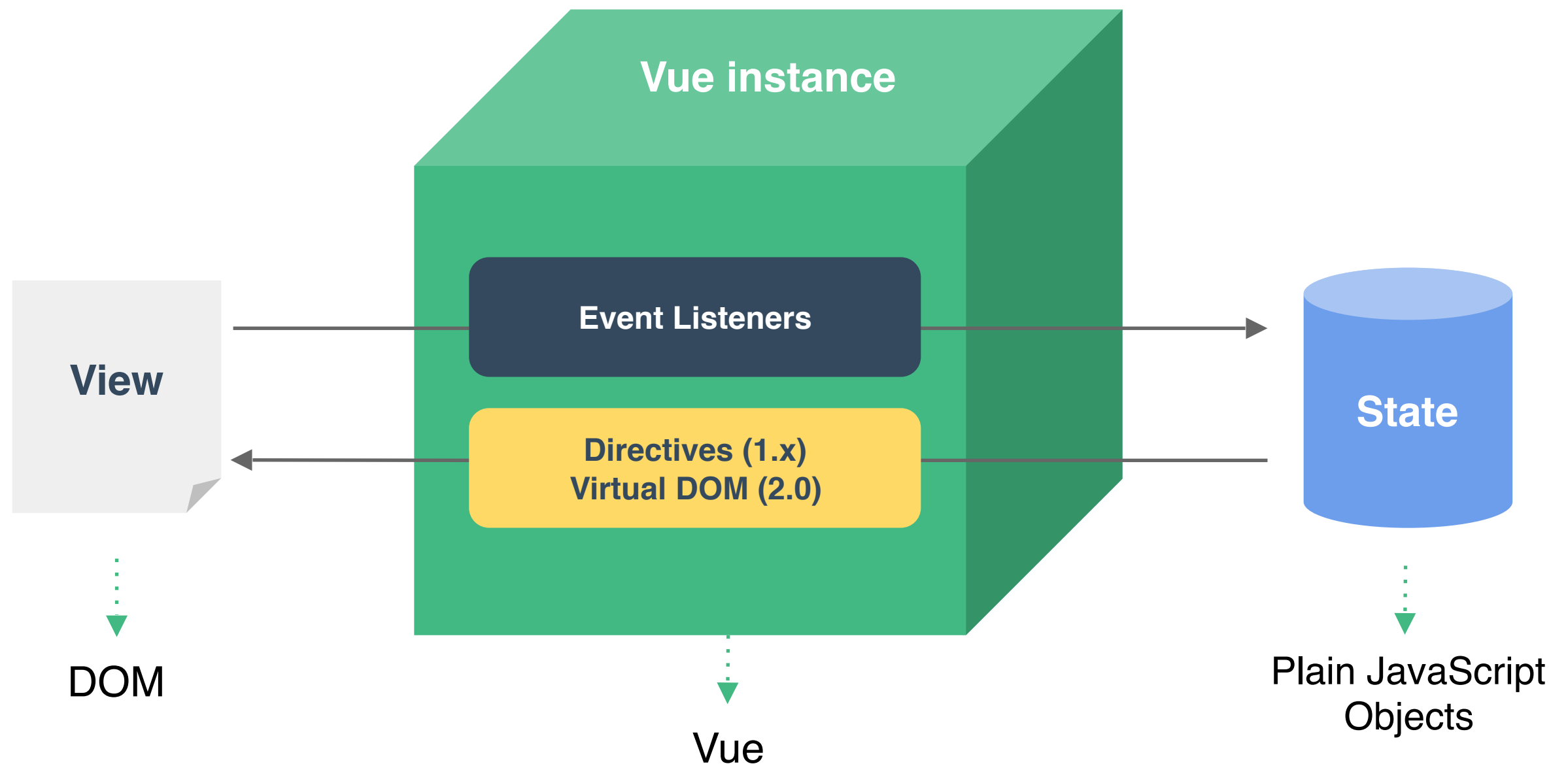
```
<div id="demo">  
  <h1>{{msg}}</h1>  
</div>
```

```
let vm = new Vue({  
  el: '#demo',  
  data: {  
    msg: 'Hello Vue.js!'  
  }  
})
```

DOM Mounting
point

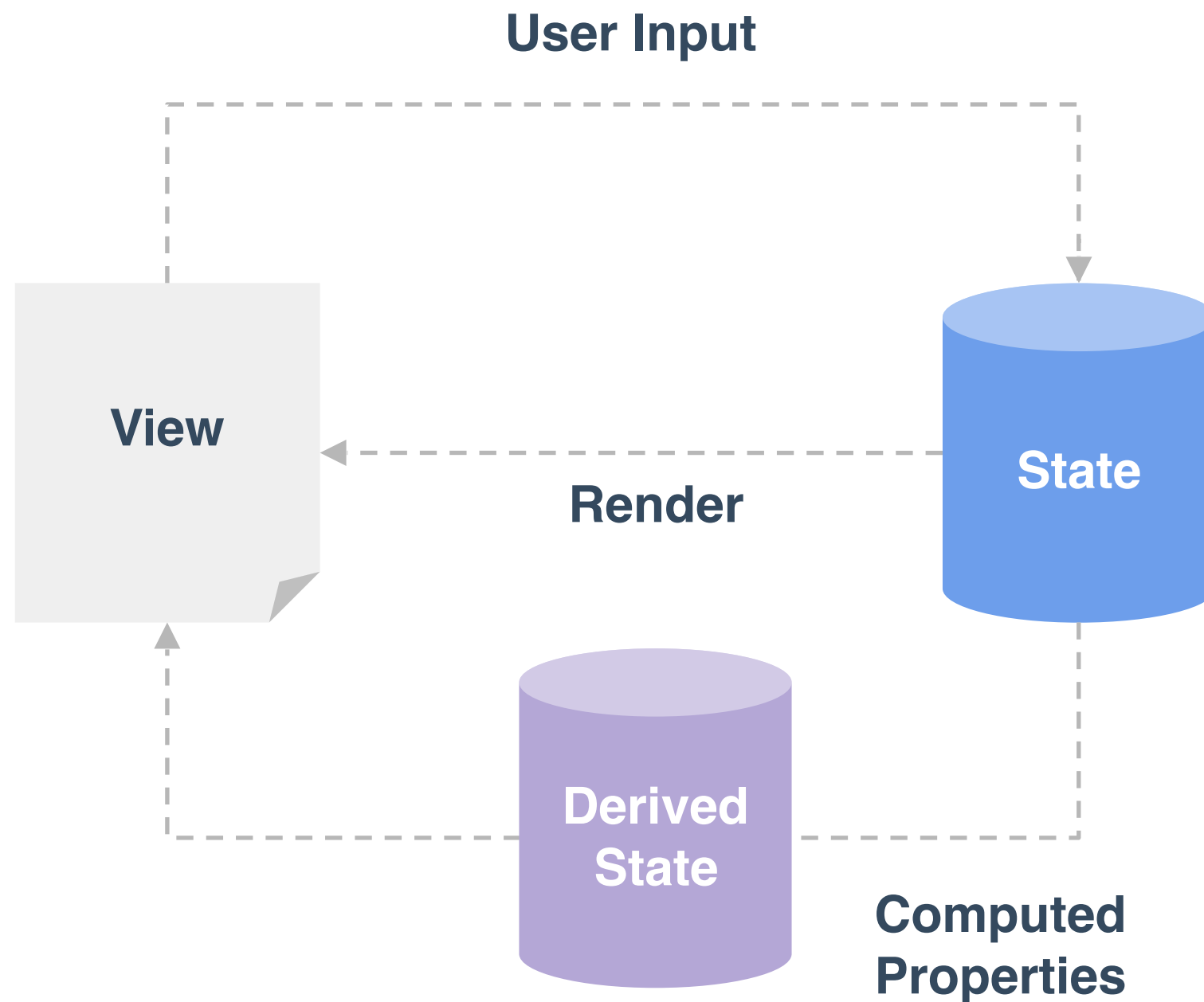
Dynamic Bindings

State



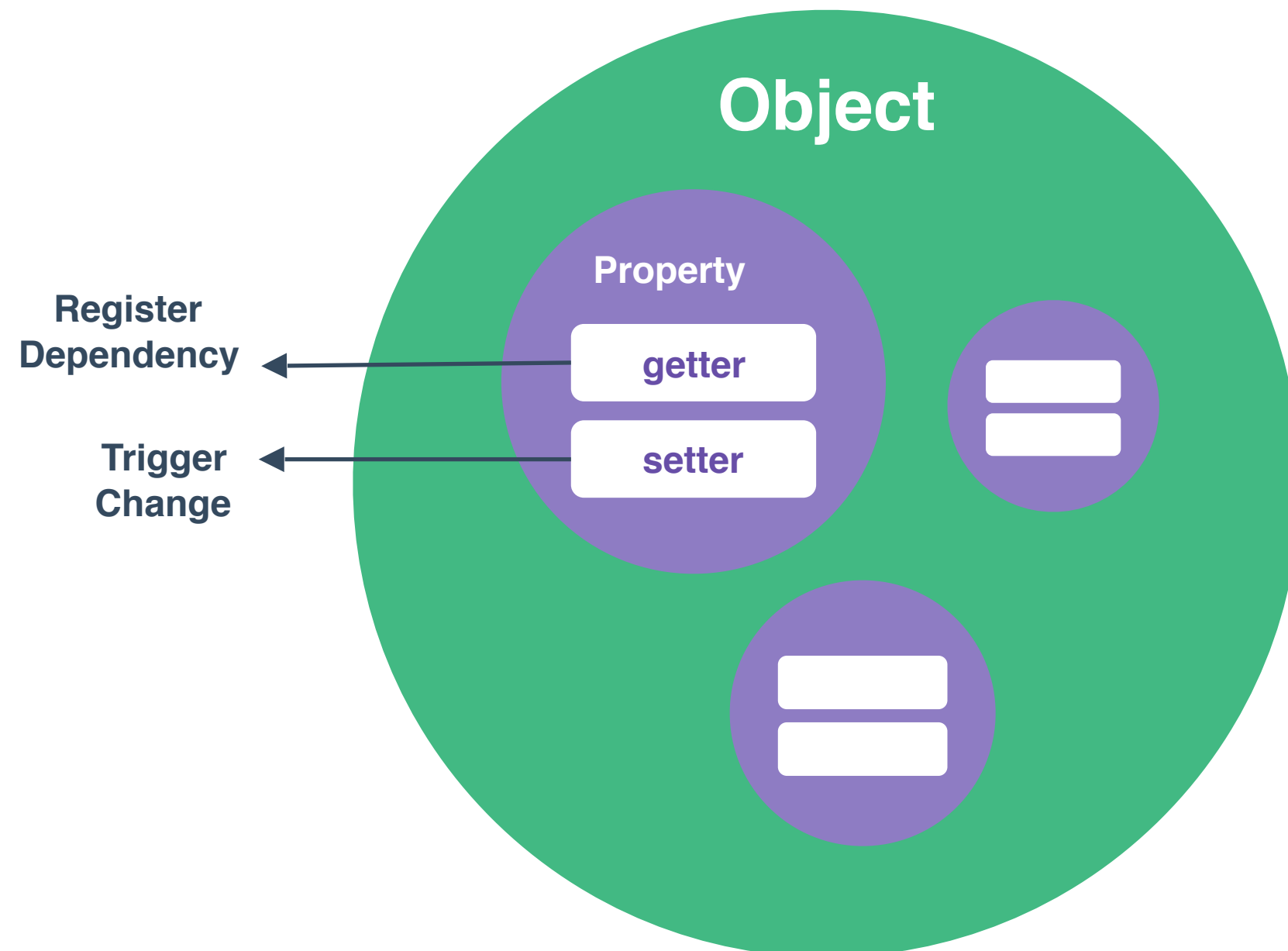
Todo List Demo

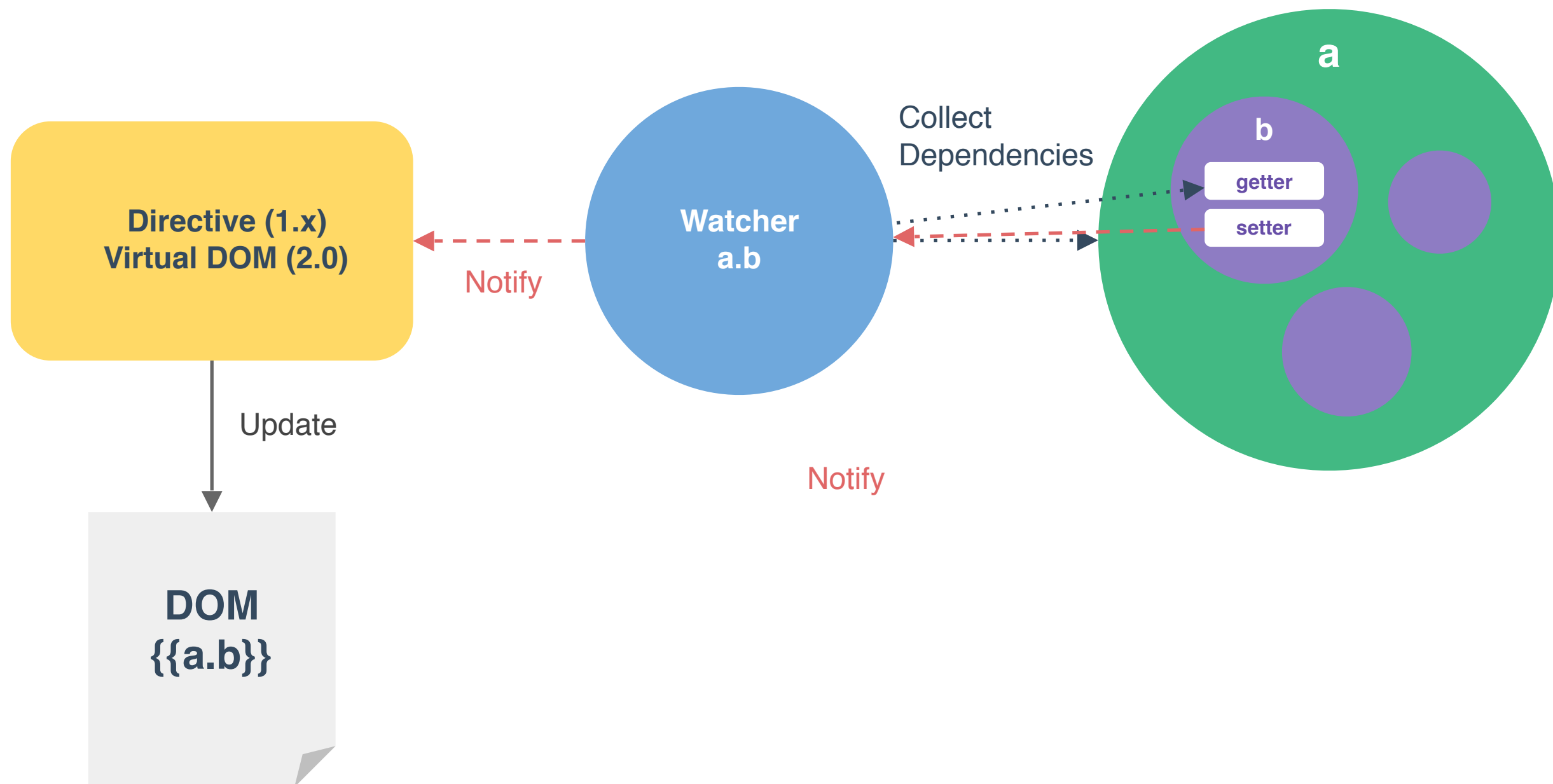
Derived State with Computed Properties



How does it work,
really?

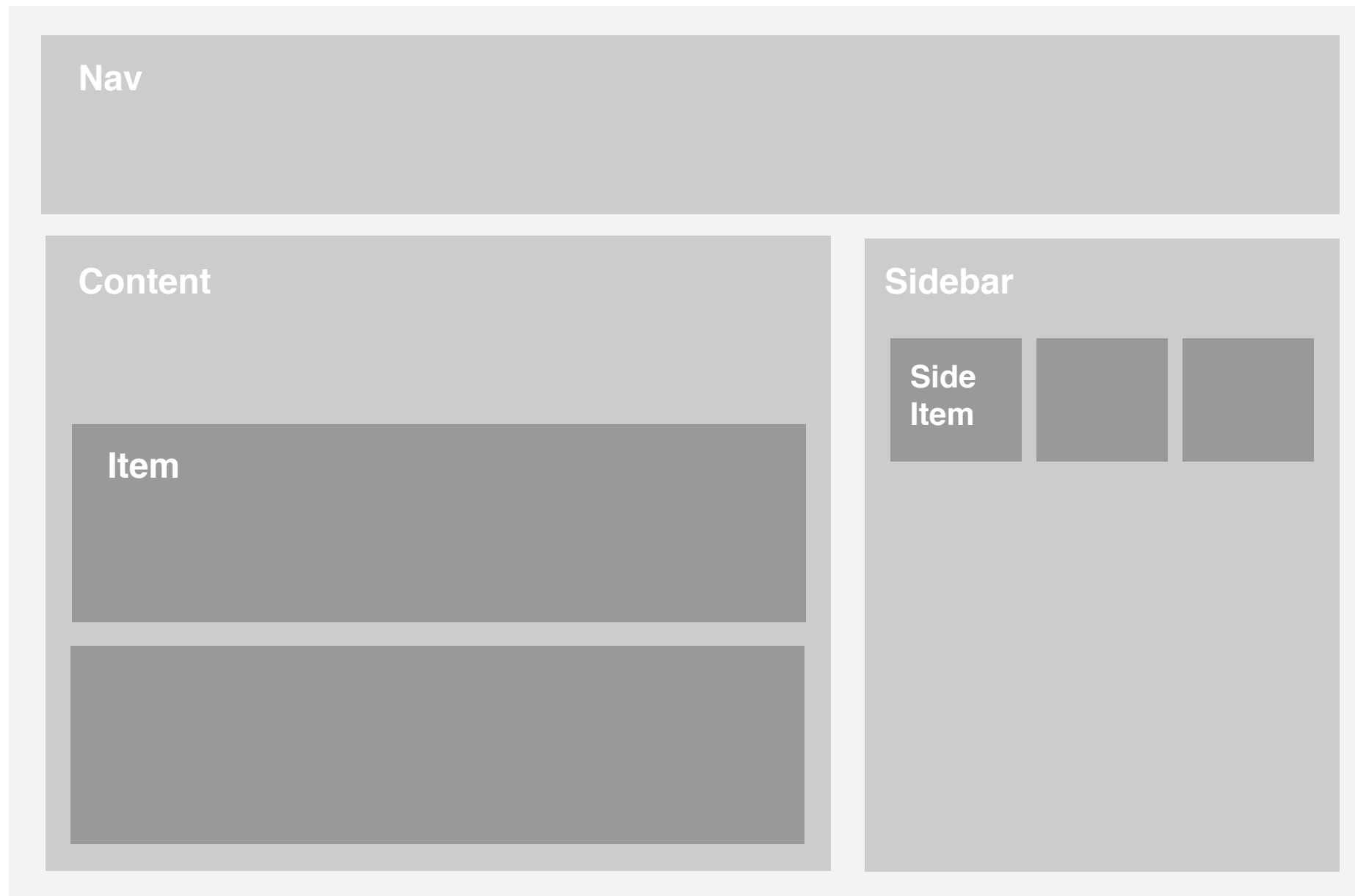
Observe POJO mutations with ES5 Object.defineProperty



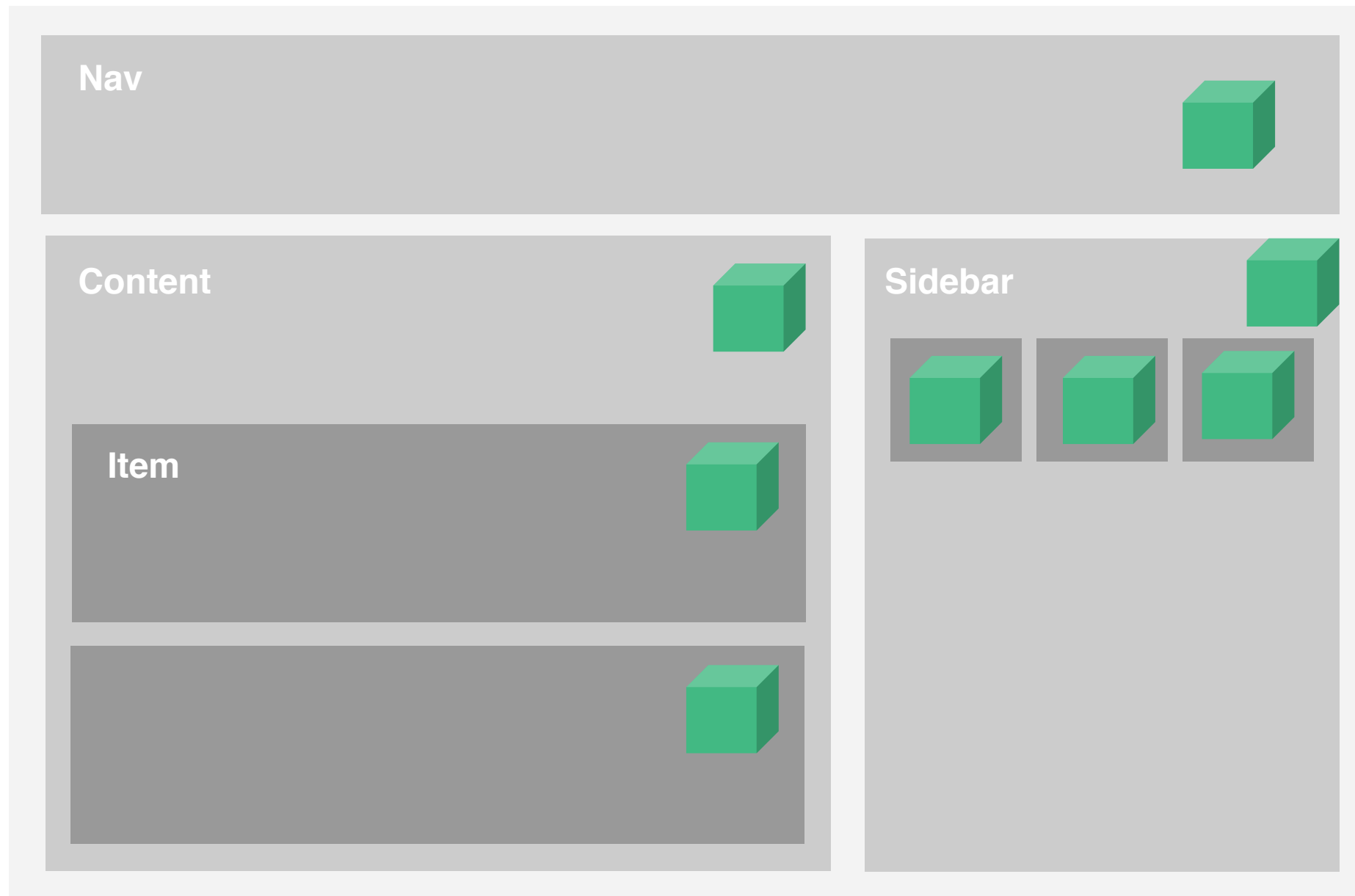


Component Oriented

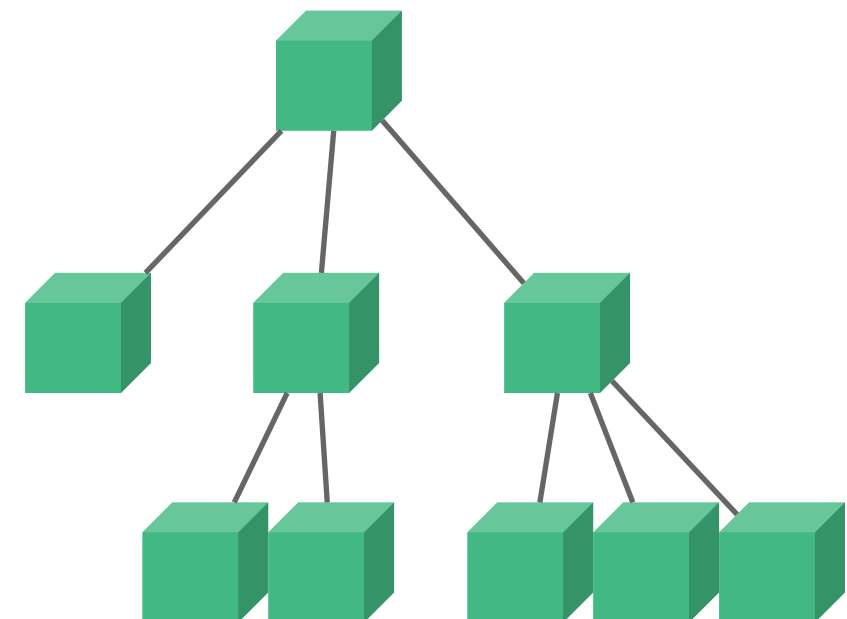
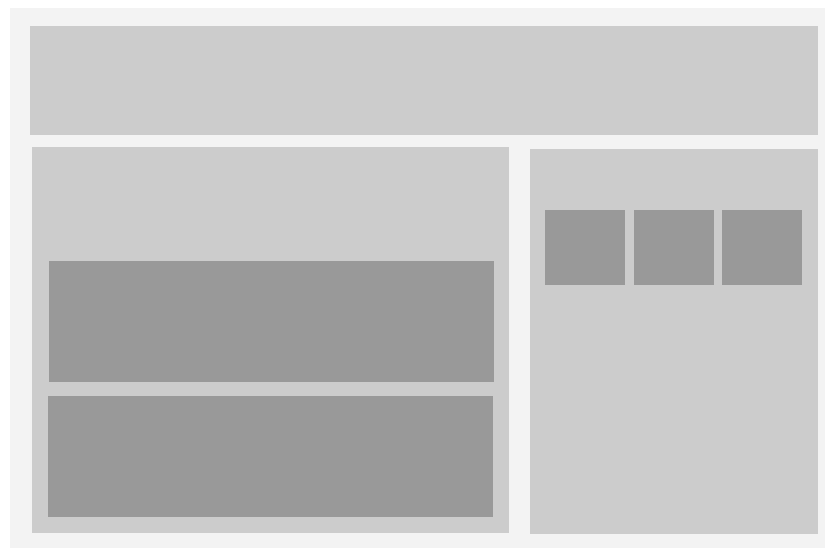
Most App UIs can be broken
down into components



Every component is responsible
for managing a piece of DOM



The entire UI can be
abstracted into a tree of
components



Registering a global component

```
Vue.component( 'my-component',  
{  
  props: [ 'msg' ],  
  template: '<p>{{msg}}</p>'  
})
```

Using the component

```
<my-component  
msg="Hello!">  
</my-component>
```



my-component's template will be inserted into the container element

```
<p>{{msg}}</p>
```

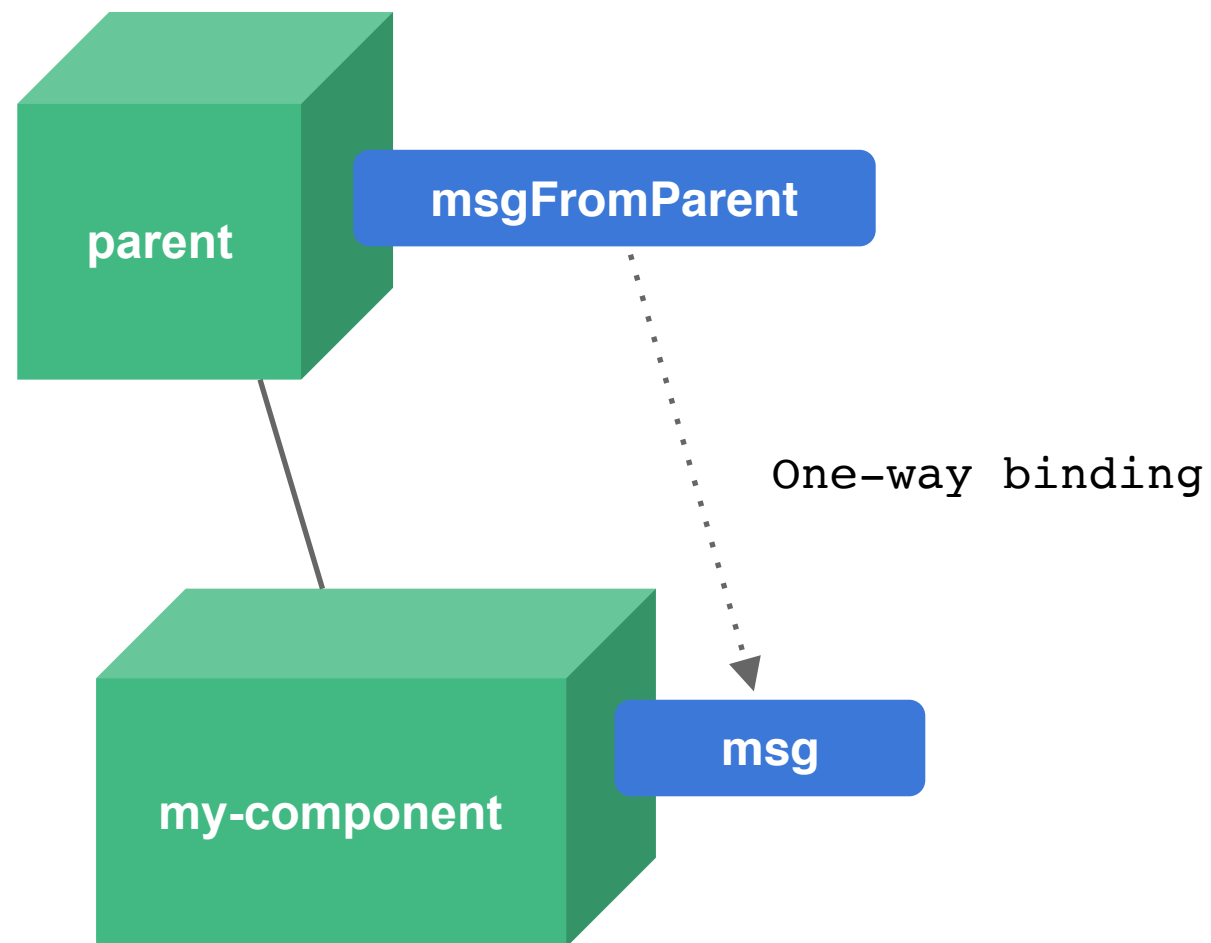


`msg` will be passed in as data state.

```
<p>Hello!</p>
```

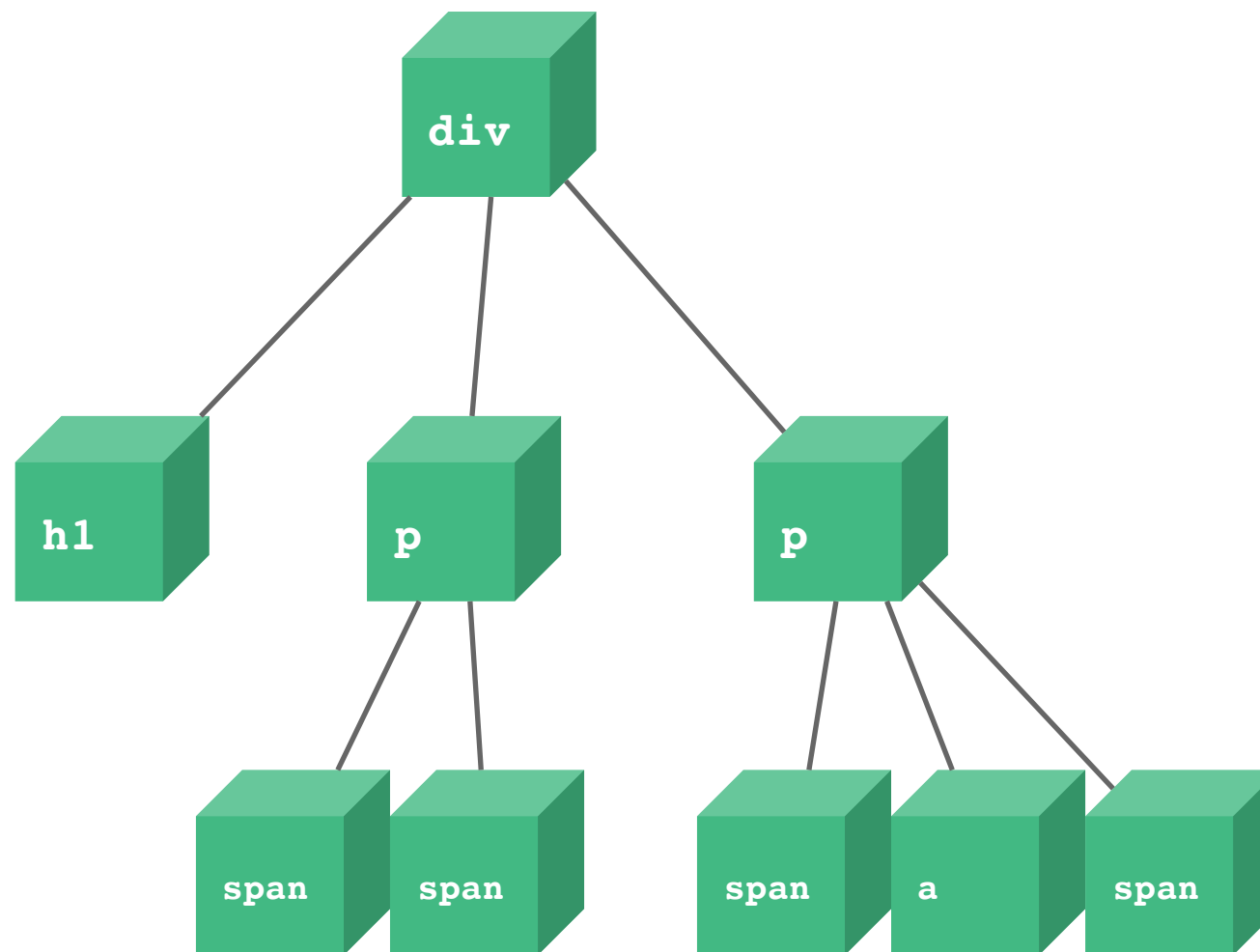
Data-passing with props

```
<my-component :msg="msgFromParent"></my-component>
```



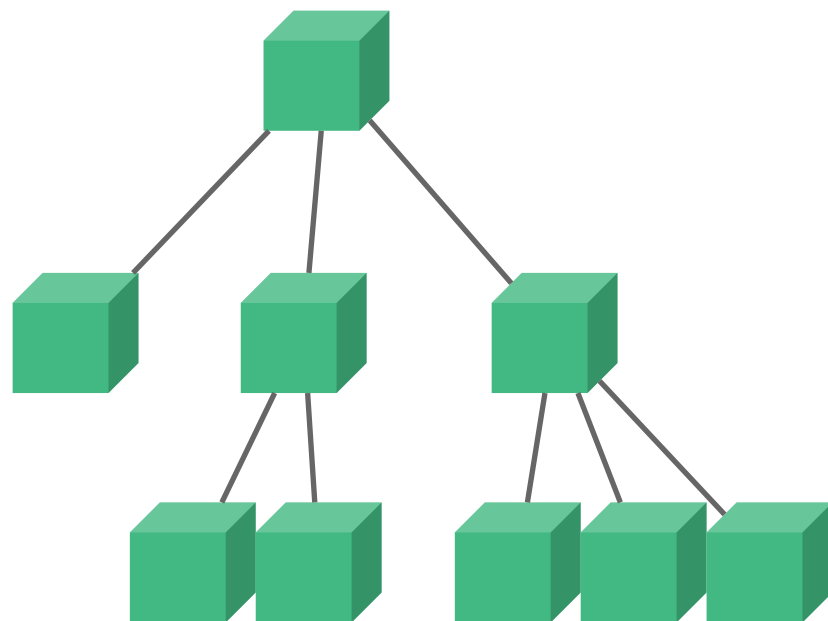
2.0: Virtual DOM

Virtual DOM Tree



Virtual DOM initial render

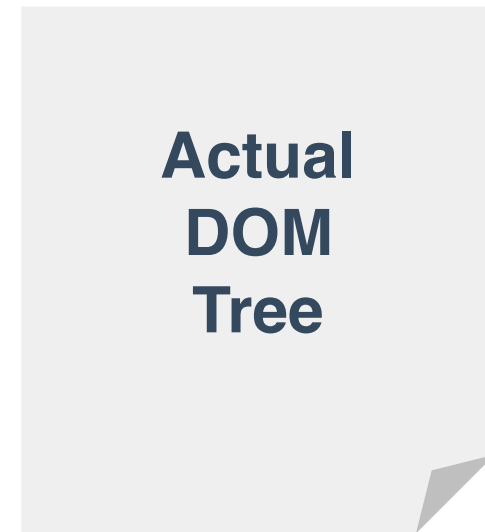
Virtual DOM Tree



create

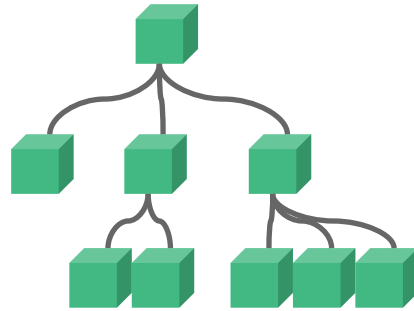


**Actual
DOM
Tree**

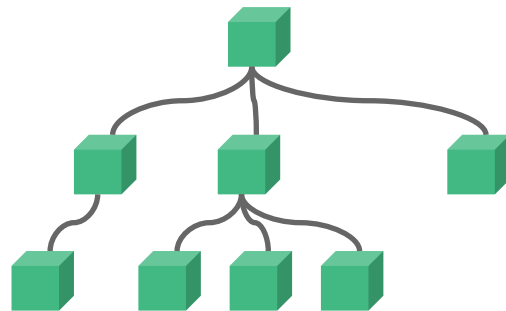


Virtual DOM update

Old Virtual DOM Tree



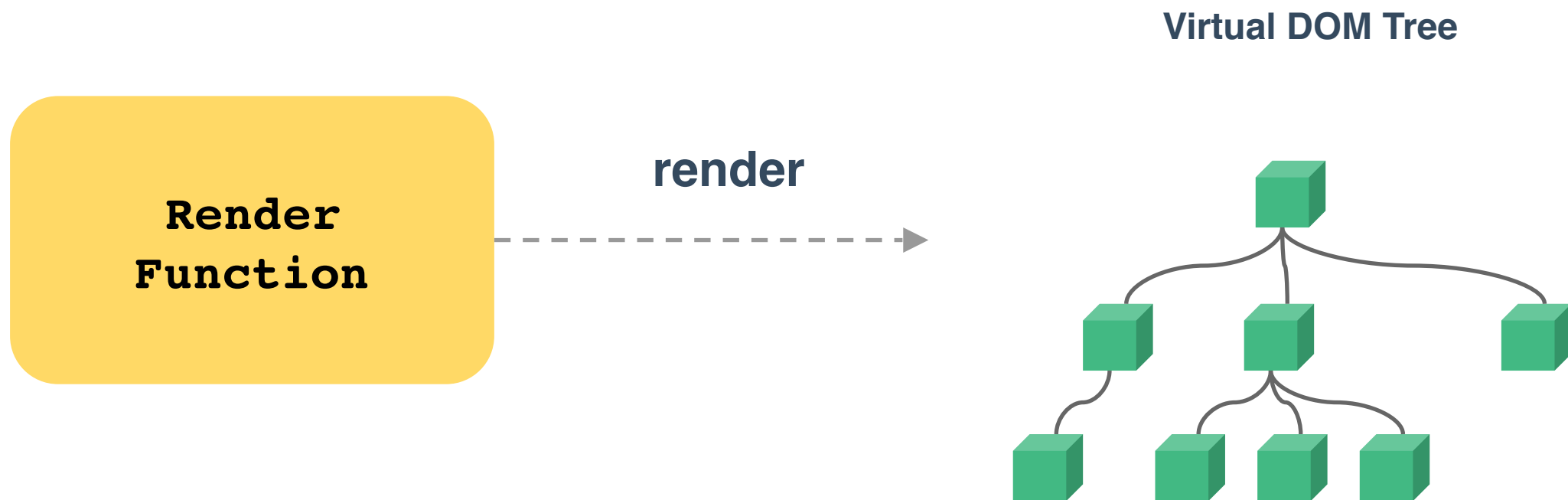
New Virtual DOM Tree



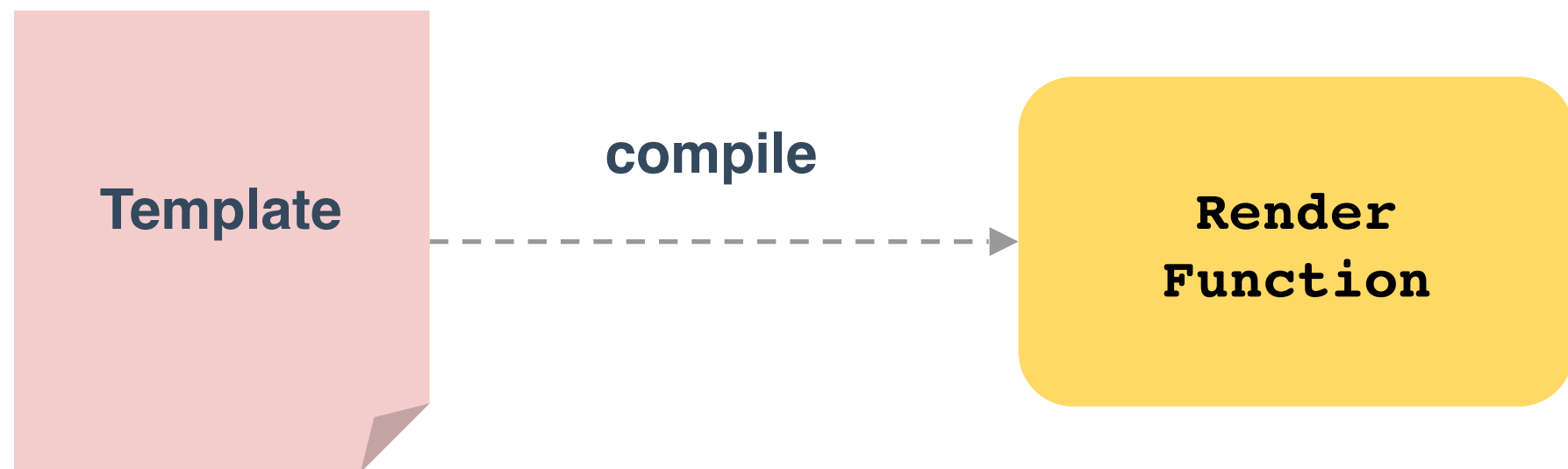
diff + patch

Actual
DOM
Tree

Render functions: Functions that return Virtual DOM trees



Template Compilation



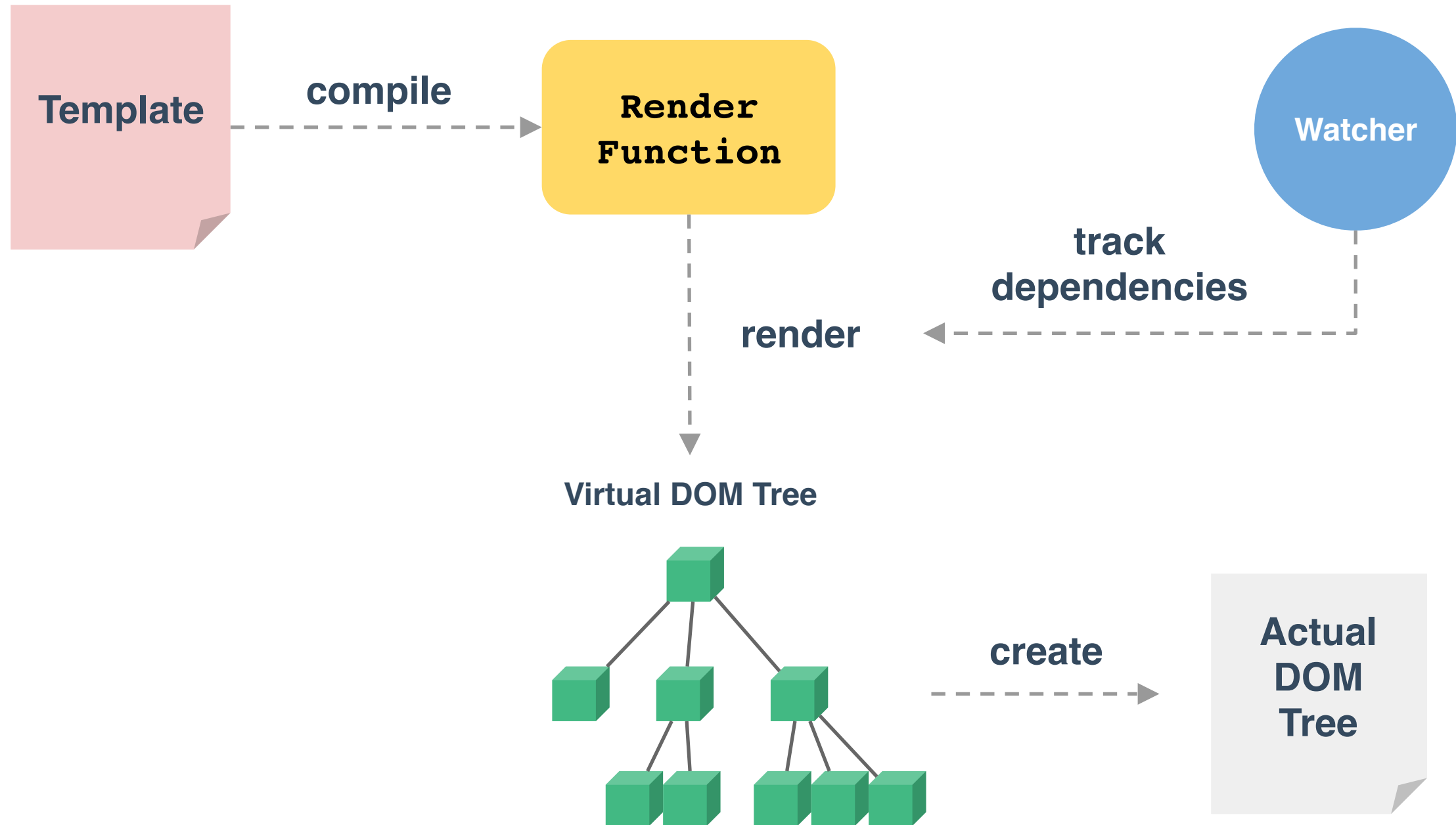
Template Compilation

```
<div id="demo">  
  <h1>{{msg}}</h1>  
</div>
```

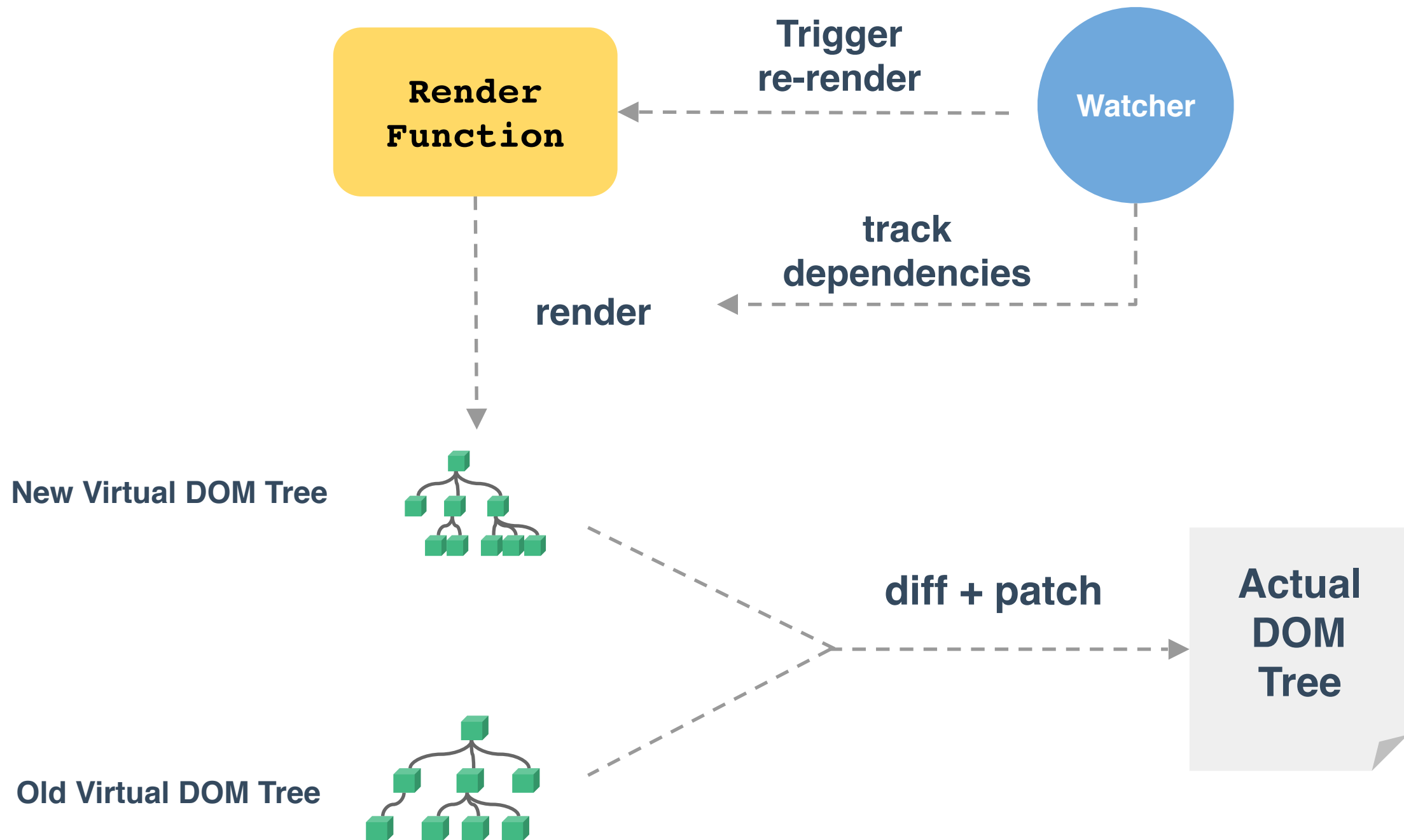


```
render (createElement) {  
  return createElement(  
    'div',  
    { attrs: { id: 'demo' } },  
    [createElement('h1', this.msg)]  
  )  
}
```

The full picture: initial render



The full picture: updates



Wrap Up

Summary

- View-layer Centric library
- Incrementally adoptable
- Declaritive & Reactive Rendering
 - Virtual DOM
 - Single source of truth state
- Component Oriented Architecture

& Beyond

- Directives, transitions, ...
- SPA routing (vue router)
- Data fetching (axios, superagent)
- State management (vuex)



Questions

tom@meagher.co

Tom Meagher
GitHub @tmm

References

- **Vue.js @ Laracon**, <https://docs.google.com/presentation/d/16MpK3I2LZz47QdLg3uMNkCC3PqmM0znXF3-FdCEpics/>
- **The Progressive Framework**, <http://slides.com/evanyou/progressive-javascript>