

# Manage Application State with Vuex

---

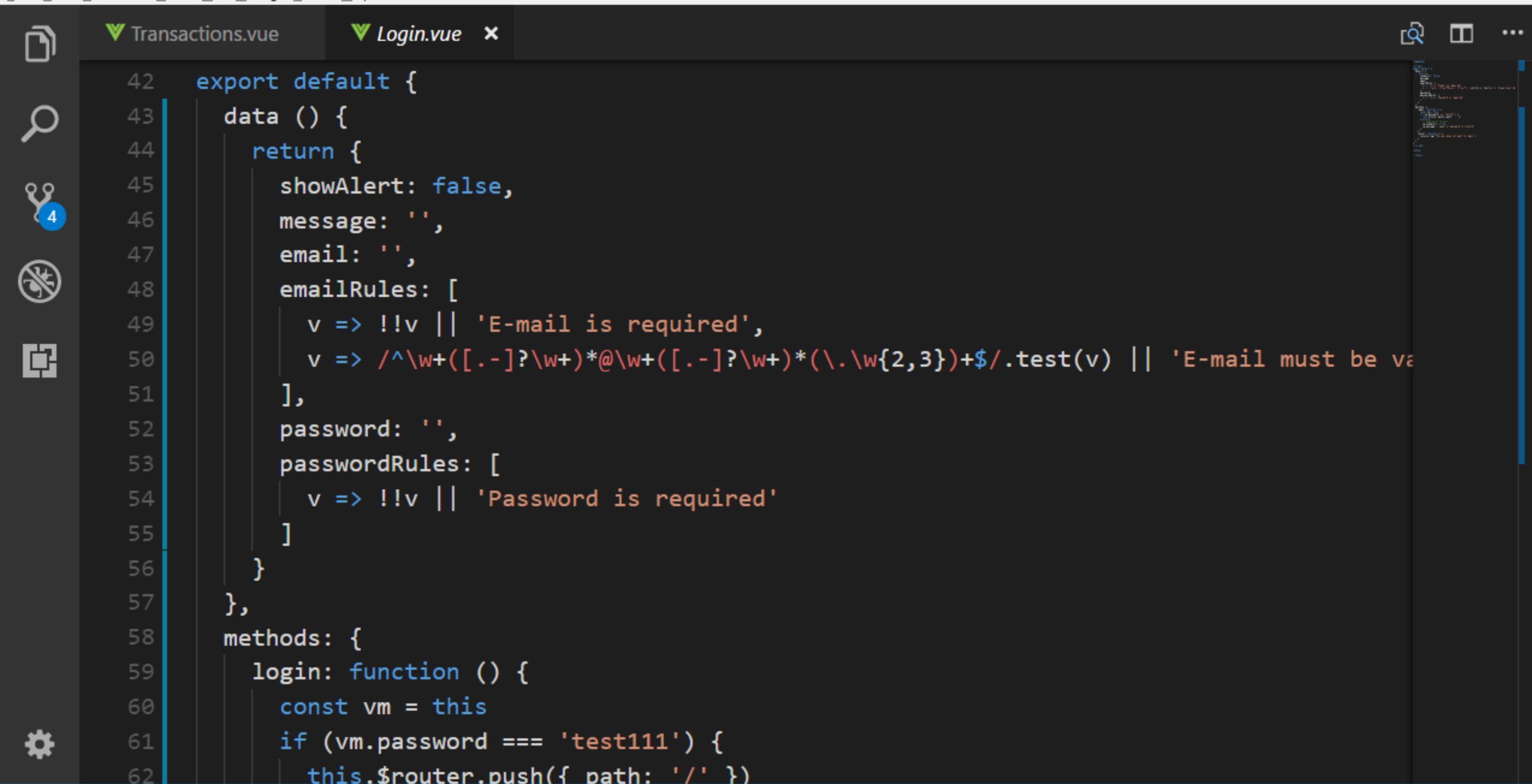


**Mark Scott**

PLURALSIGHT AUTHOR

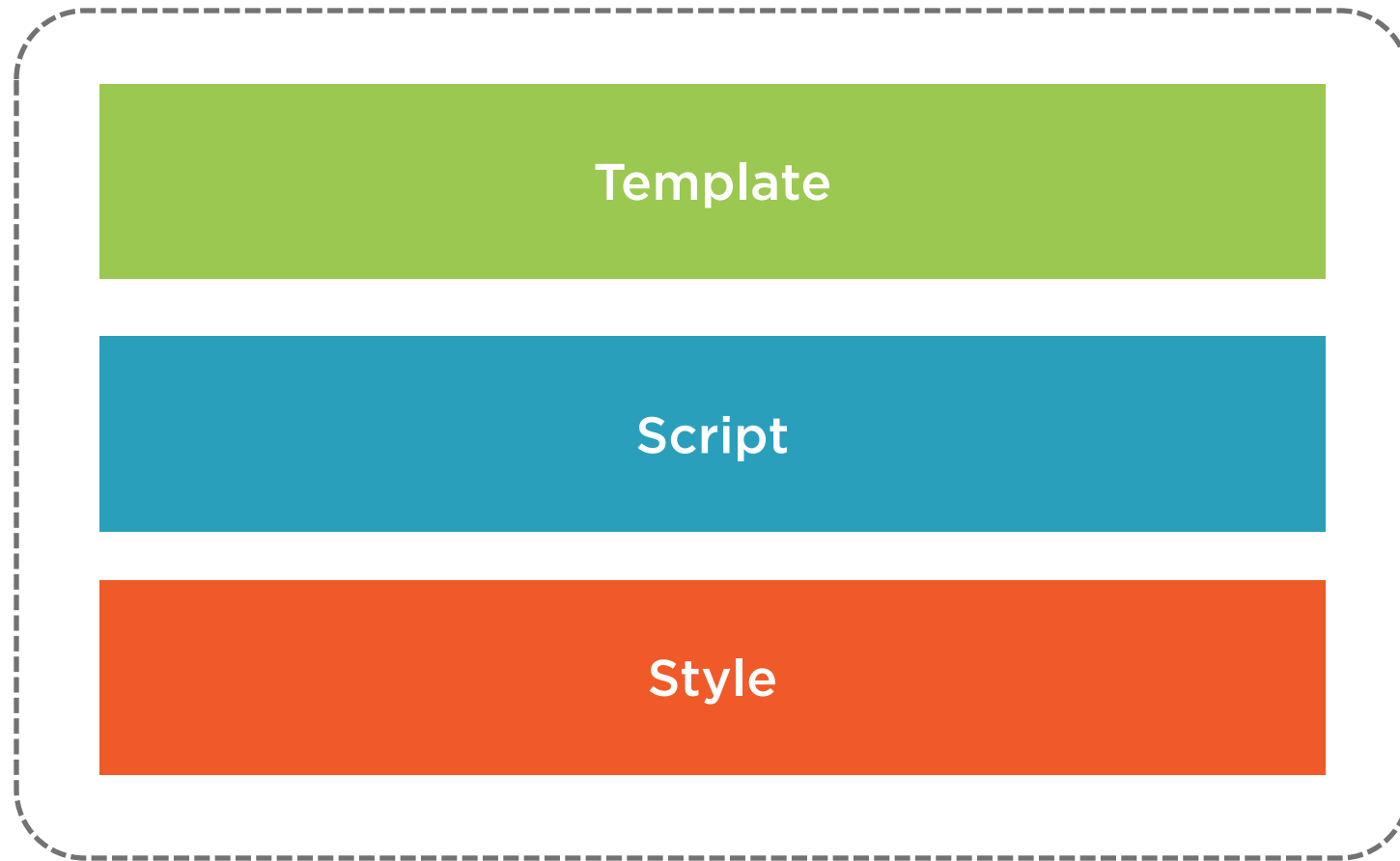
@tripletdad99



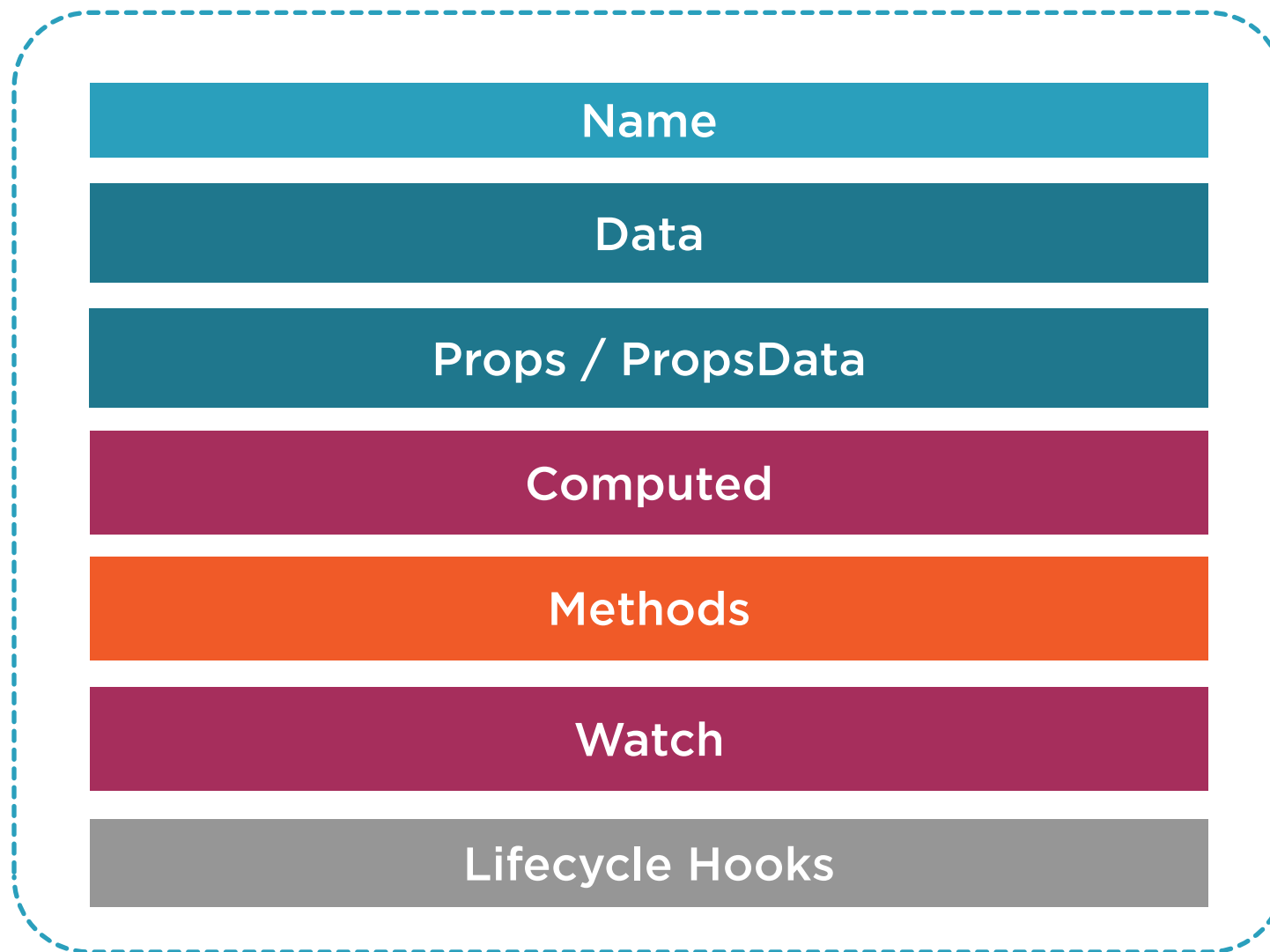


```
42 export default {
43   data () {
44     return {
45       showAlert: false,
46       message: '',
47       email: '',
48       emailRules: [
49         v => !!v || 'E-mail is required',
50         v => /^\\w+([.-]?\\w+)*@\\w+([.-]?\\w+)*\\.\\w{2,3}+$/\\.test(v) || 'E-mail must be va
51       ],
52       password: '',
53       passwordRules: [
54         v => !!v || 'Password is required'
55       ]
56     }
57   },
58   methods: {
59     login: function () {
60       const vm = this
61       if (vm.password === 'test111') {
62         this.$router.push({ path: '/' })
```

# Anatomy of a Single File Component



# Script Block



# Example Data Function

```
<script>

export default {
  data () {
    return {
      user: { first: 'Some', last: 'Body' },
      isActive: true
    }
  }
}

</script>
```

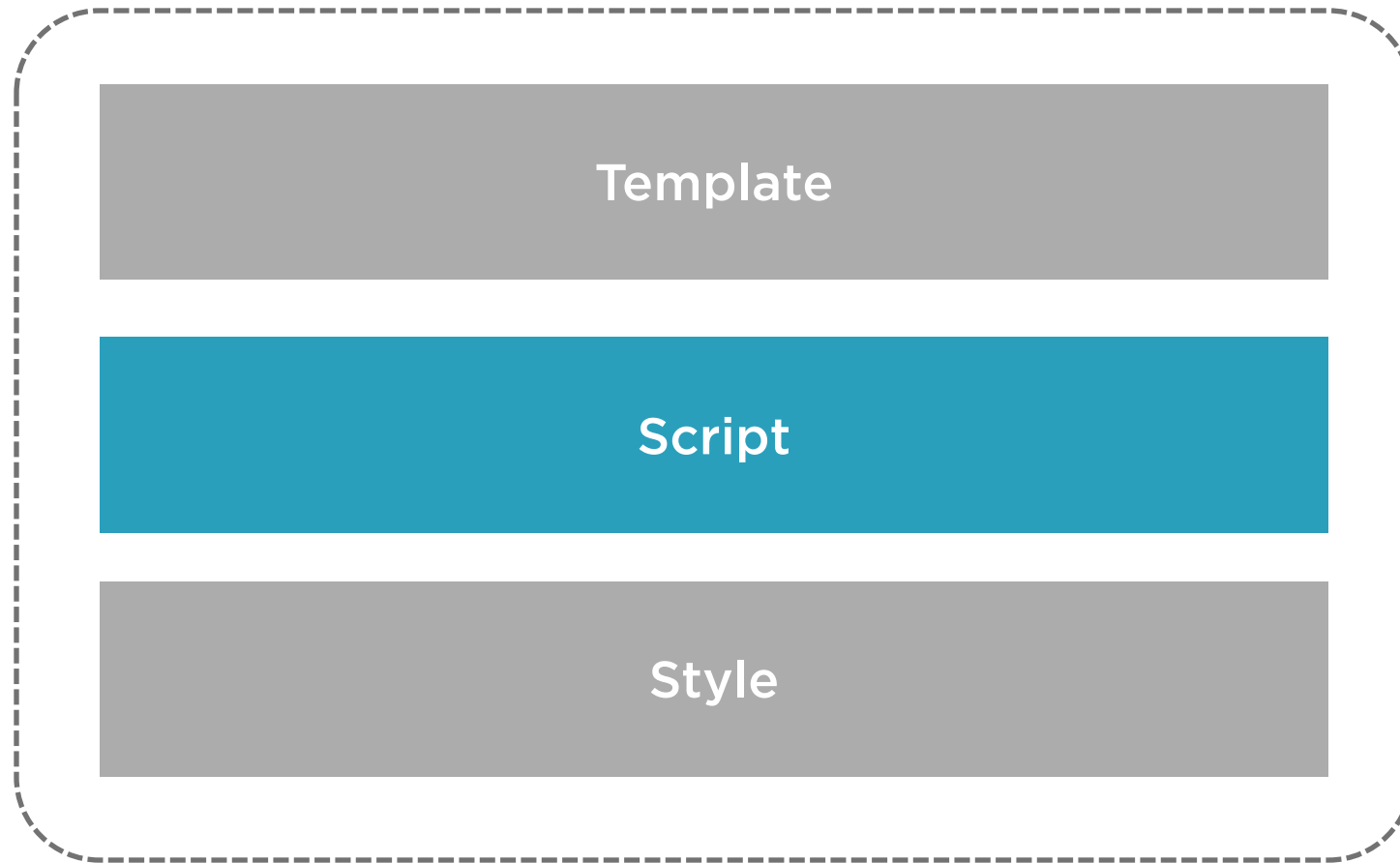


# Example Data Function

```
<script>
export default {
  data () => ({
    user: { first: 'Some', last: 'Body' },
    isActive: true
  })
}
</script>
```



# Local State Management



# What is Vuex?

---





“Vuex is a state management pattern & library for Vue.js applications. It serves as a centralized store for all the components in an application...”

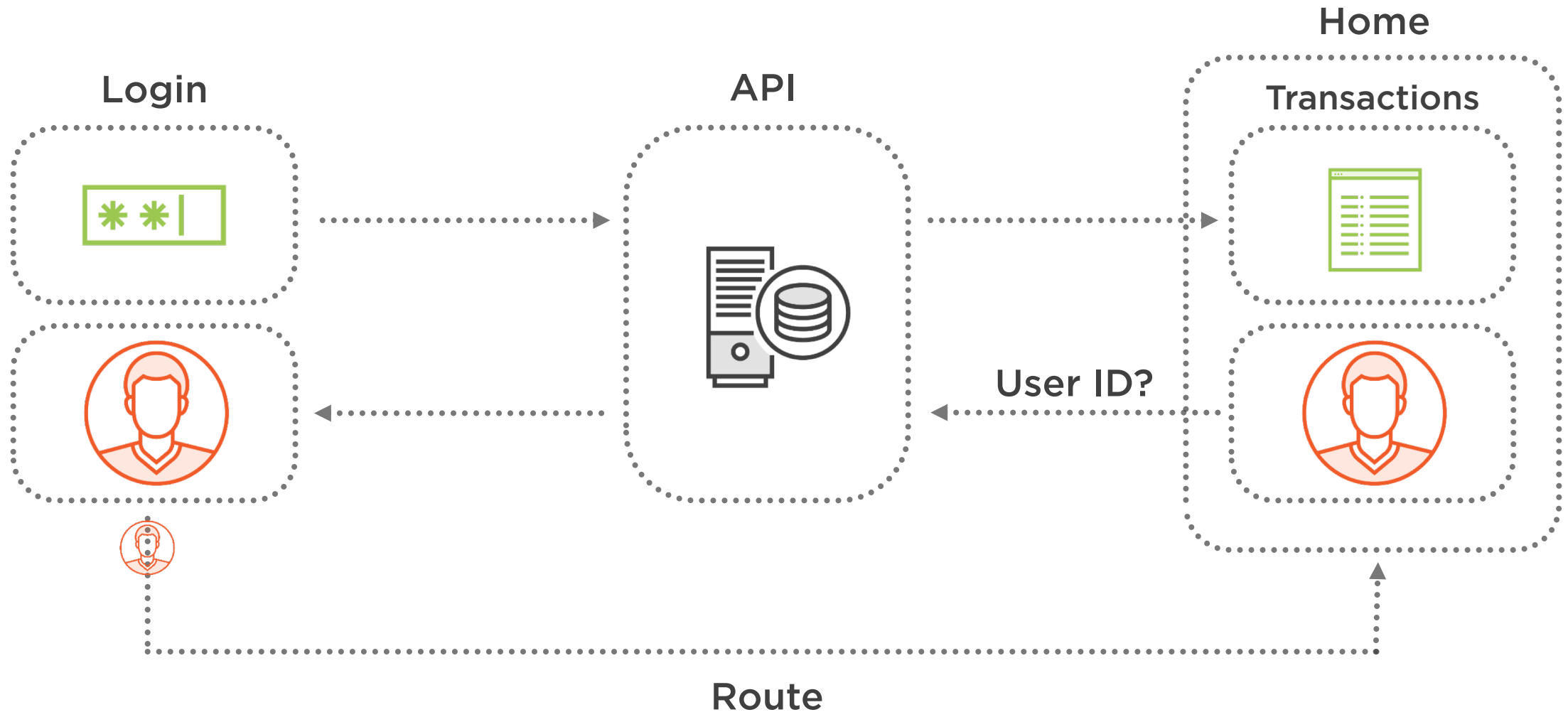
[vuex.vuejs.org/en/intro.html](https://vuex.vuejs.org/en/intro.html)



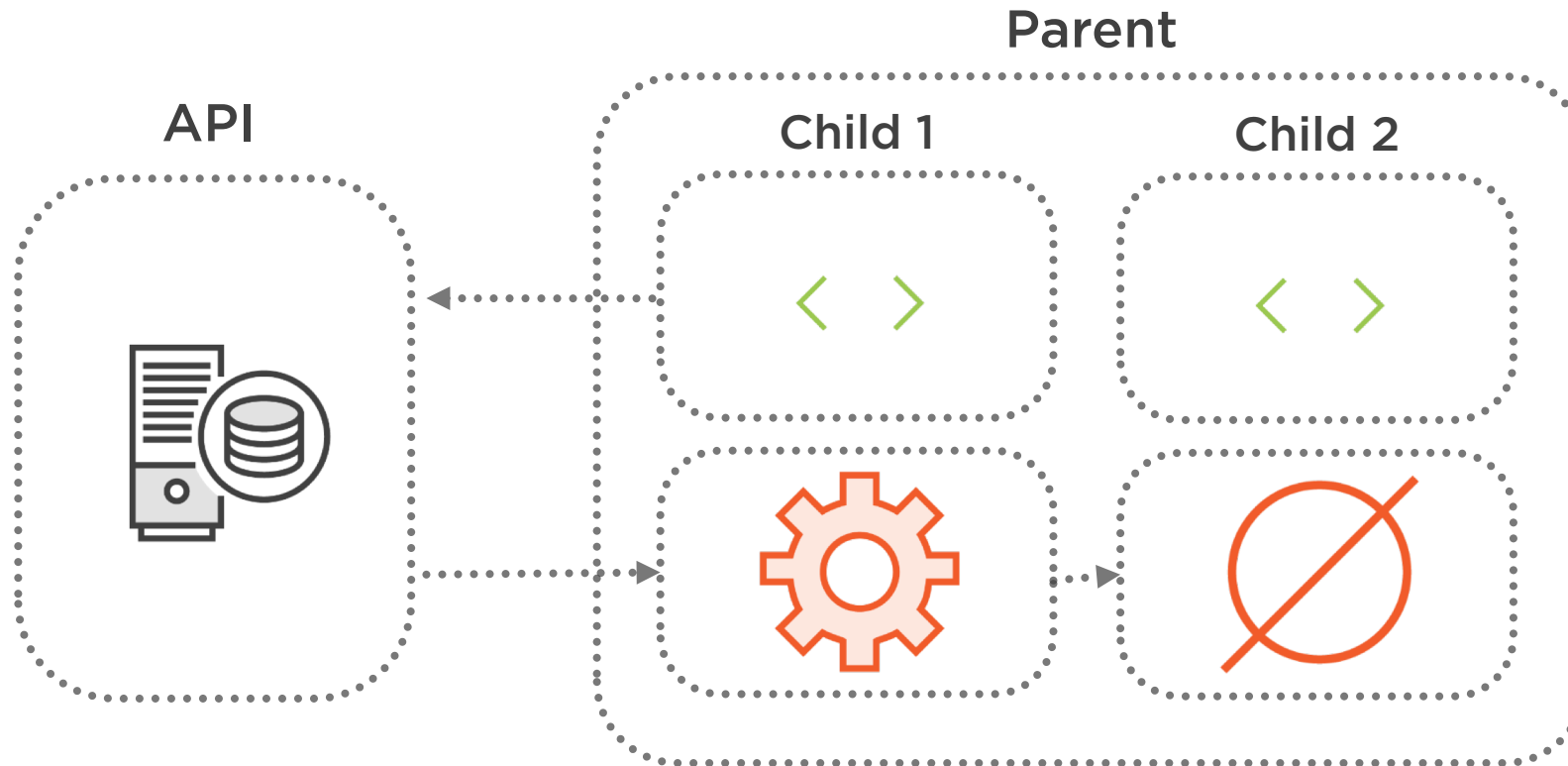
Do we really need a  
centralized store?



# Without a Central Store



# Child Components & Parameters



Do we really ~~need~~ want a  
centralized store?

**YES**



# Vuex Store Objects

---



# Vuex Store Objects

State

Actions

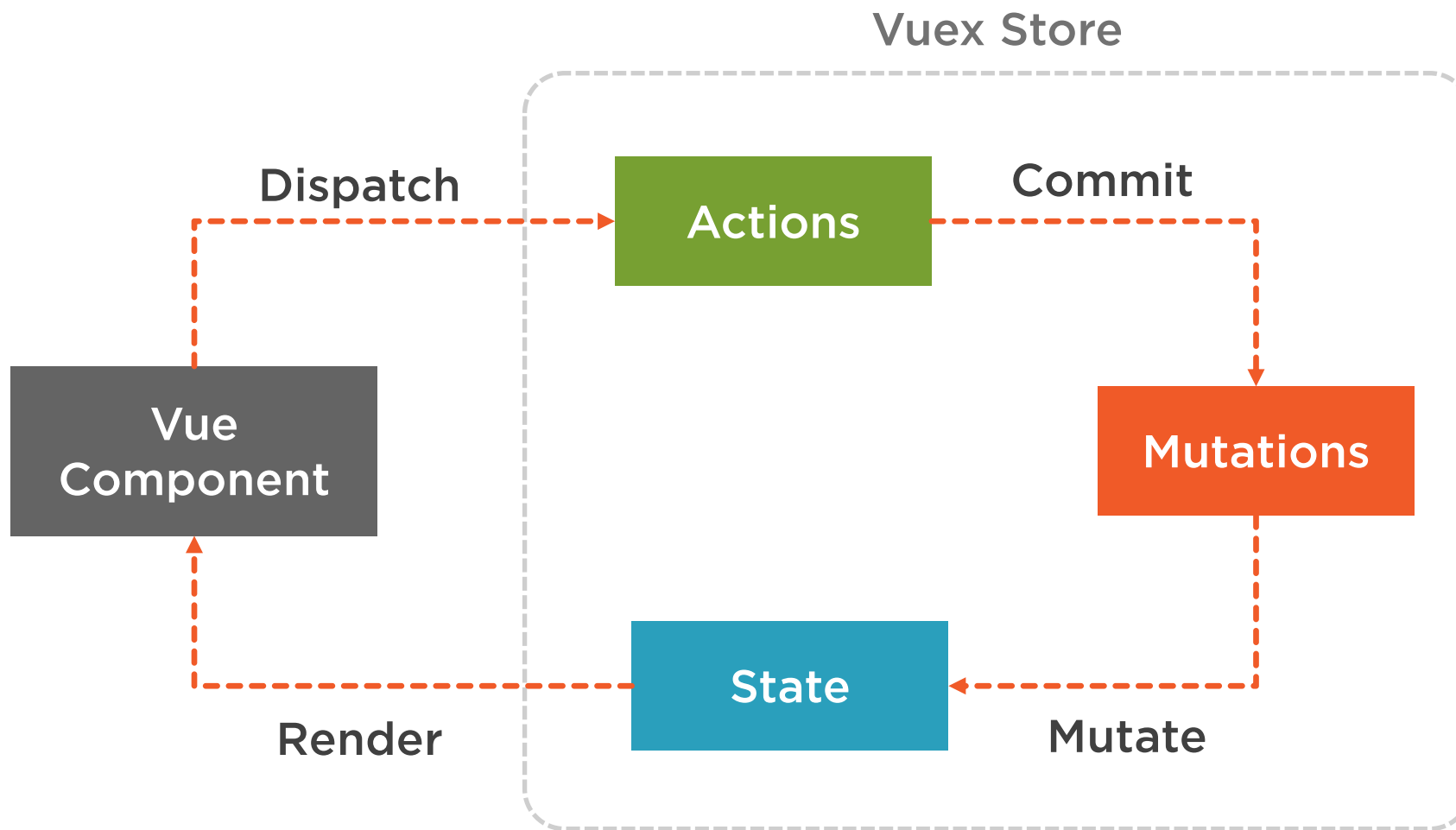
Mutations

Getters

Modules



# Vuex Workflow





# State

---



# Vuex Store: state

State

Actions

Mutations

Getters

Modules



# Vuex Store: state example

```
const store = new Vuex.Store({  
  state: {  
    user: null,  
    isAuthorized: false,  
    roles: []  
  }  
})
```



```
function initialState () {  
    return {  
        user: null,  
        isAuthorized: false,  
        roles: []  
    }  
}  
  
const state = initialState()  
  
export default {  
    state,  
    // ...  
}
```

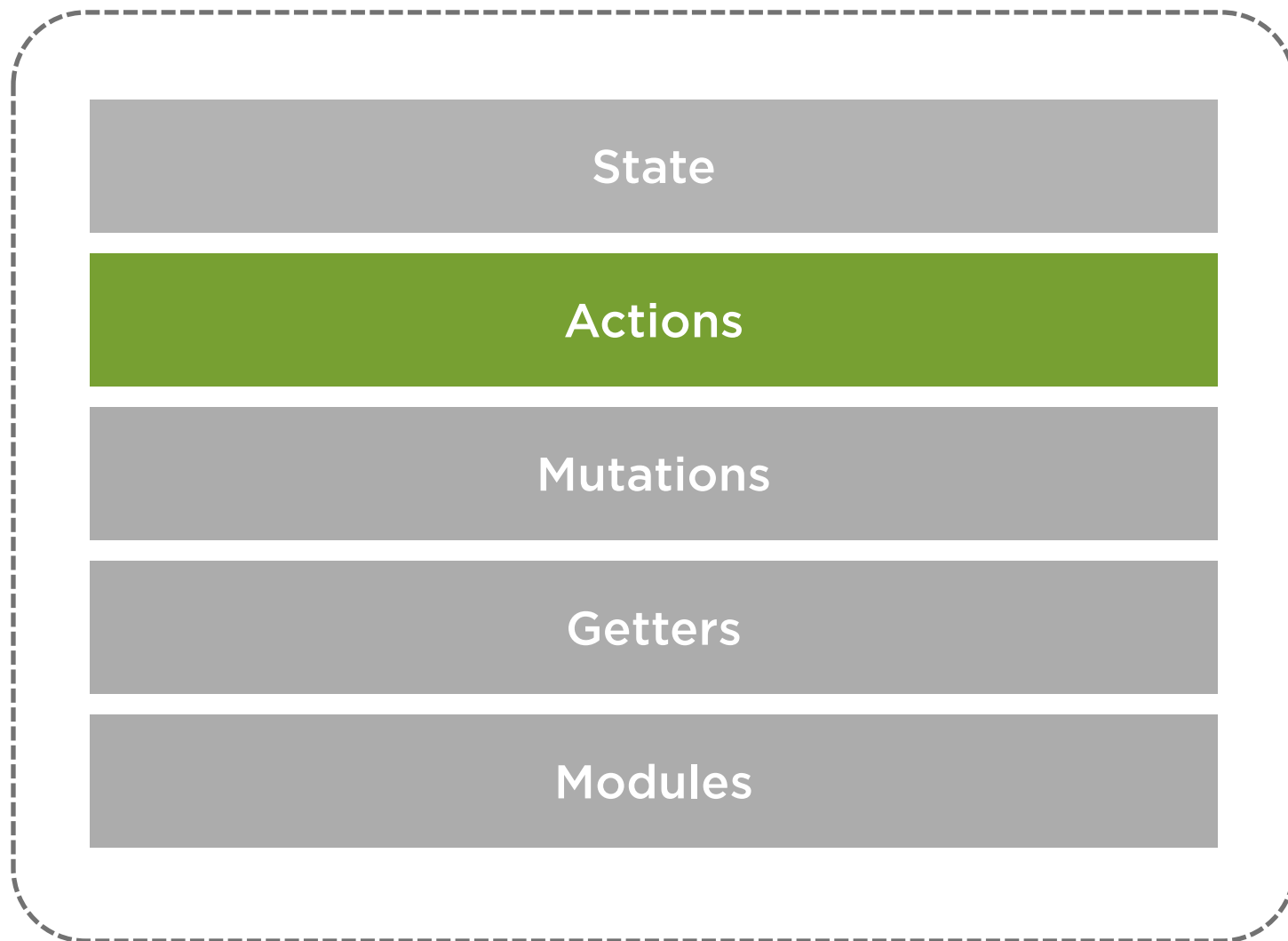


# Actions

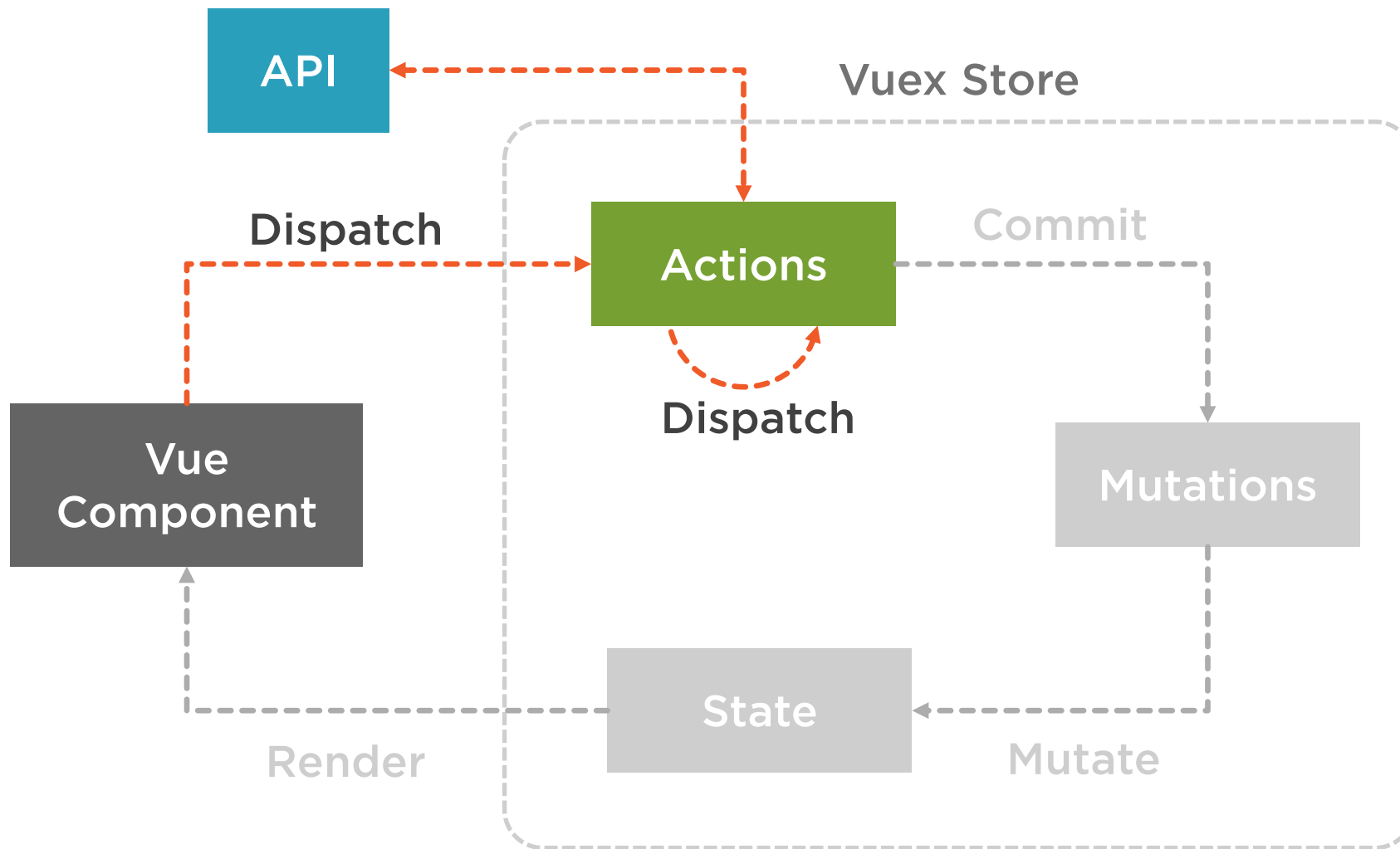
---



# Vuex Store: actions



# Vuex Workflow



# Vuex Store: action example

```
const store = new Vuex.Store({  
  state: { ... },  
  actions: {  
    loginUser (context, user) {  
      // Login Code Goes Here...  
      let isAuthorized = true  
      context.commit('loginUser', isAuthorized )  
    }  
  }  
})
```





# Vuex Store: action example

```
const store = new Vuex.Store({  
  state: { ... },  
  actions: {  
    loginUser ({ commit }, user) {  
      // Login Code Goes Here...  
      let isAuthenticated = true  
      commit('loginUser', isAuthenticated )  
    }  
  }  
})
```

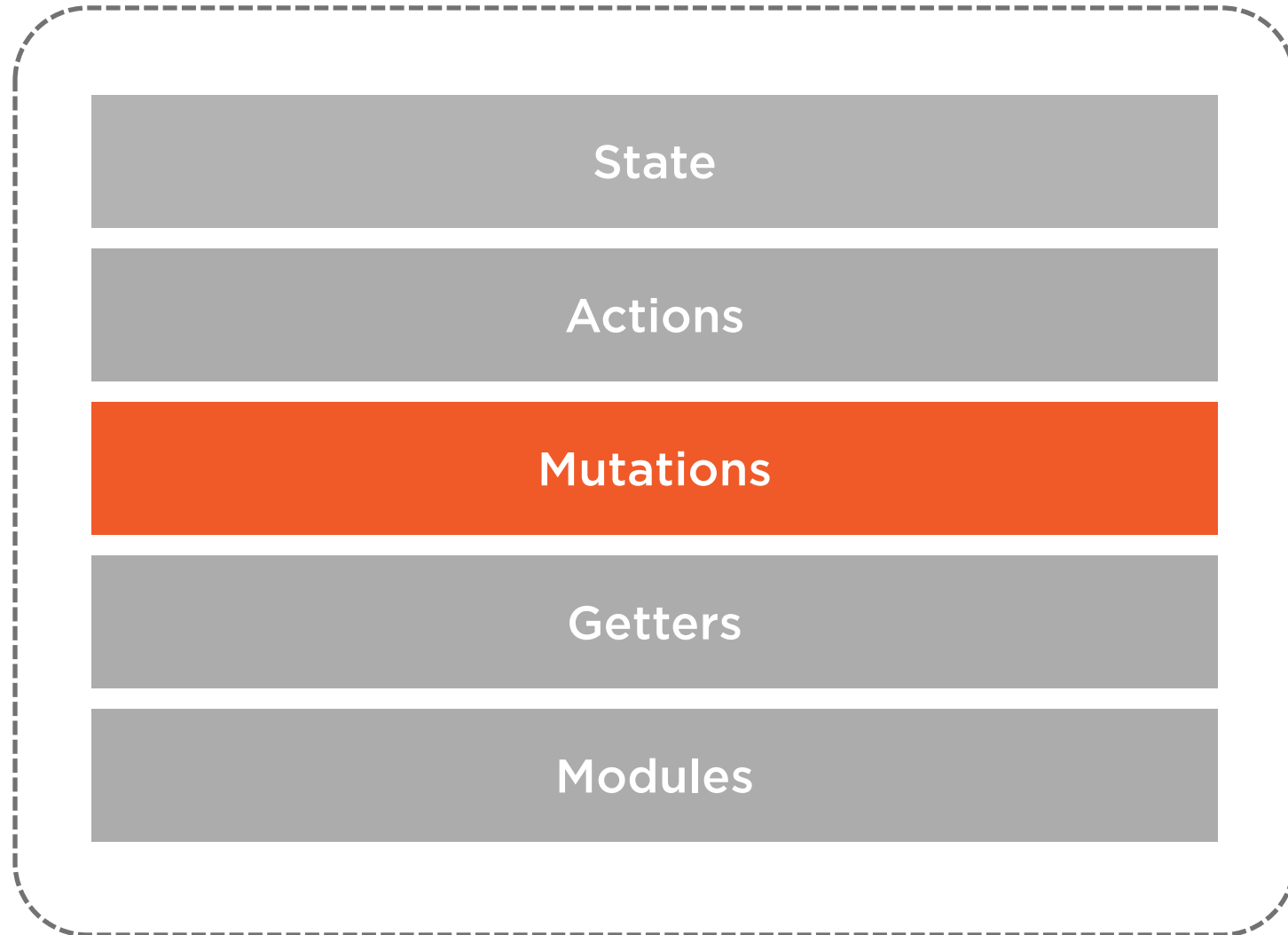


# Mutations

---



# Vuex Store: mutations

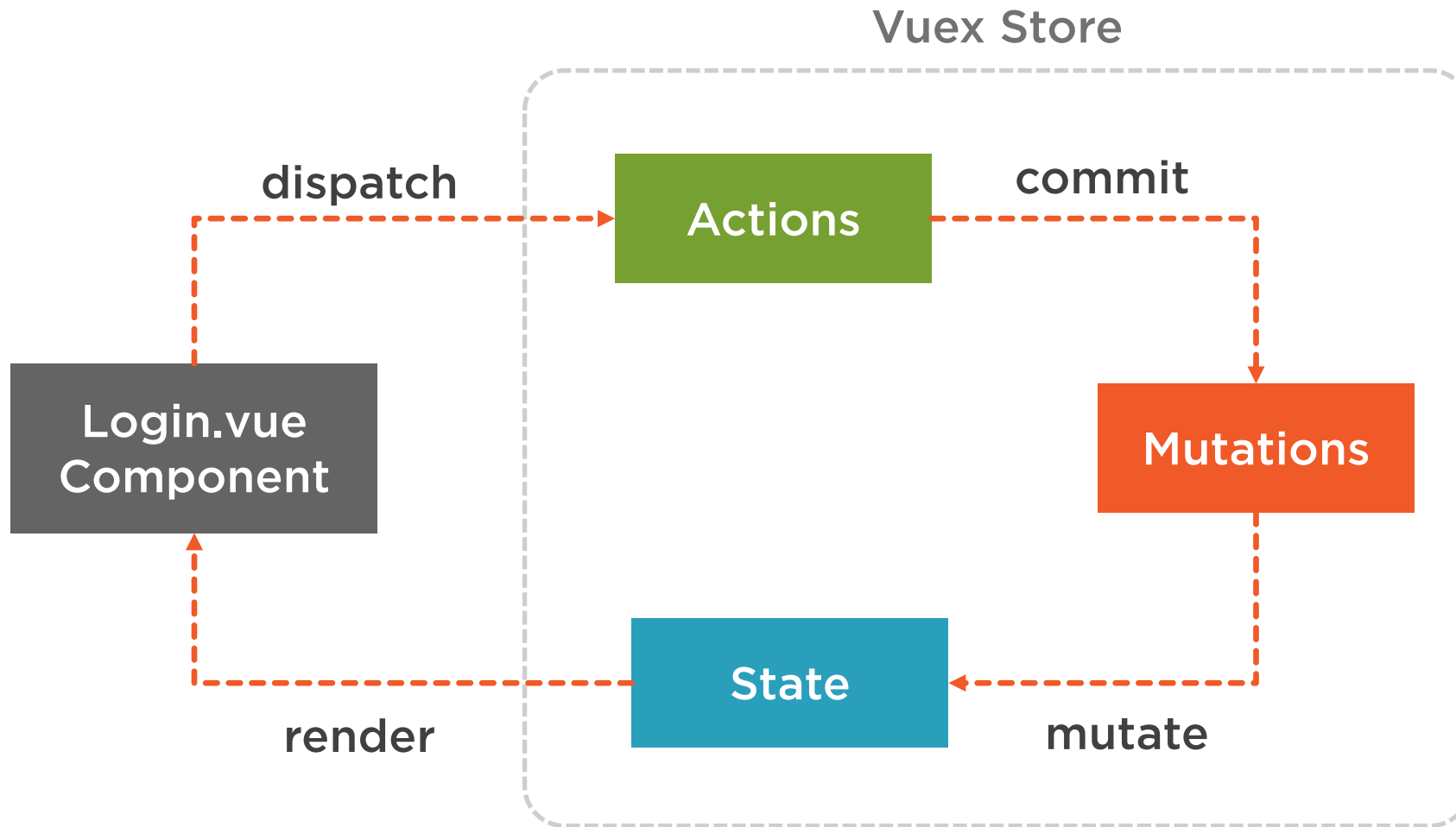


# Vuex Store: mutation example

```
const store = new Vuex.Store({  
  state: { ... },  
  actions: { loginUser (context, user) { ... } },  
  mutations: {  
    loginUser (state, isAuthorized) {  
      state.isAuthorized = isAuthorized  
    }  
  })
```



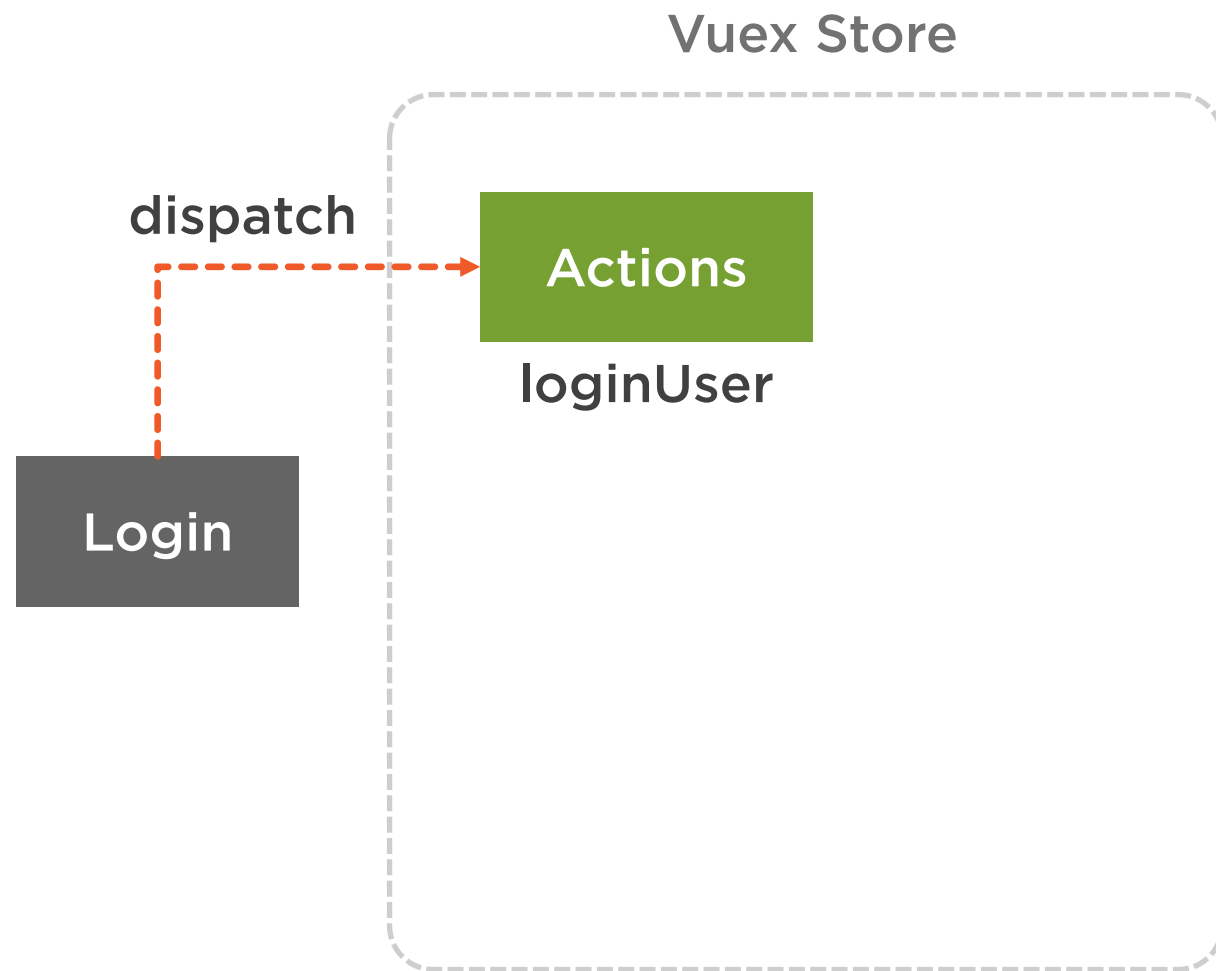
# Login Example



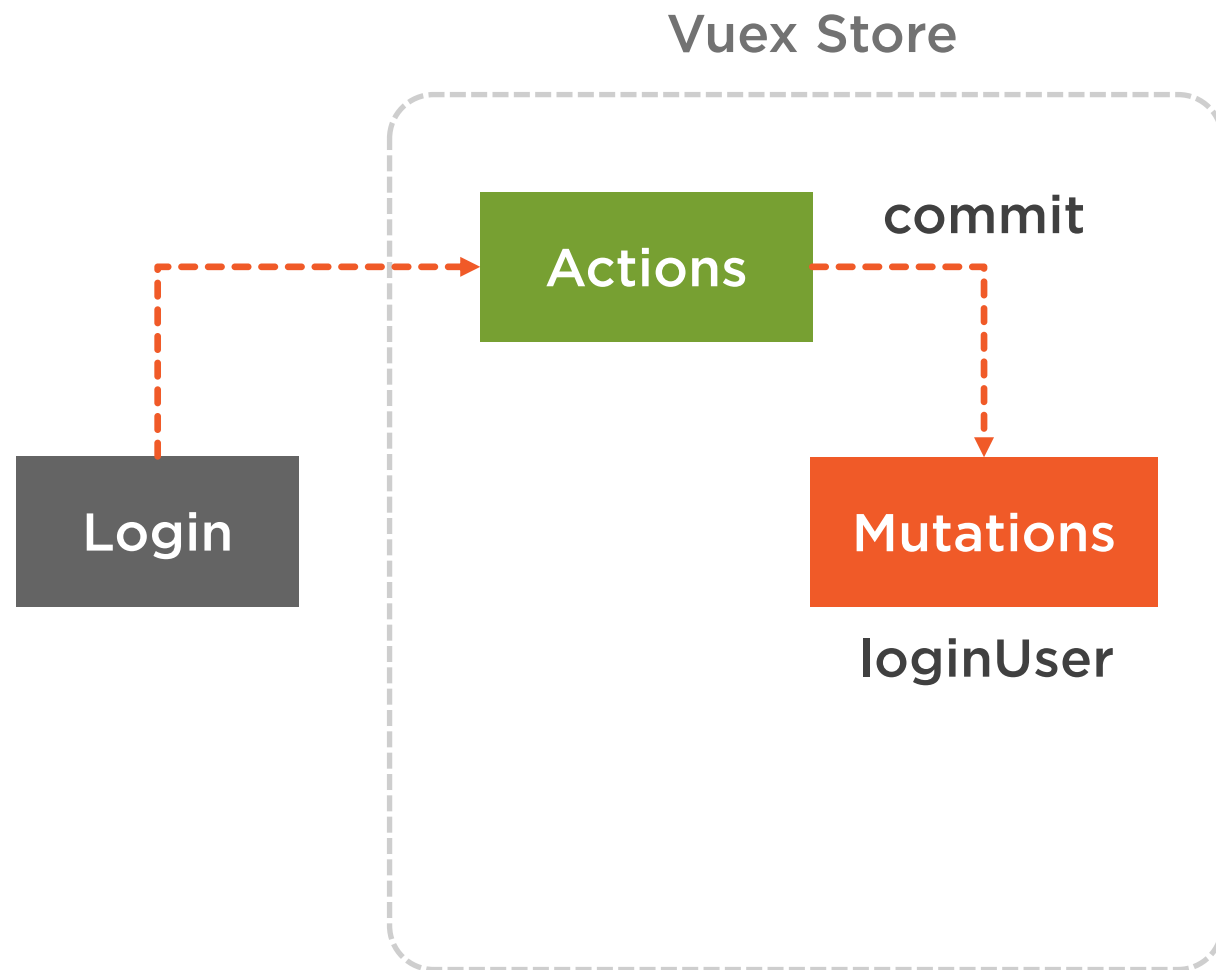
```
<script>
export default {
  data: () => ({
    user: { ... }
  }),

  methods: {
    login: function () {
      this.$store
        .dispatch('loginUser')
    },

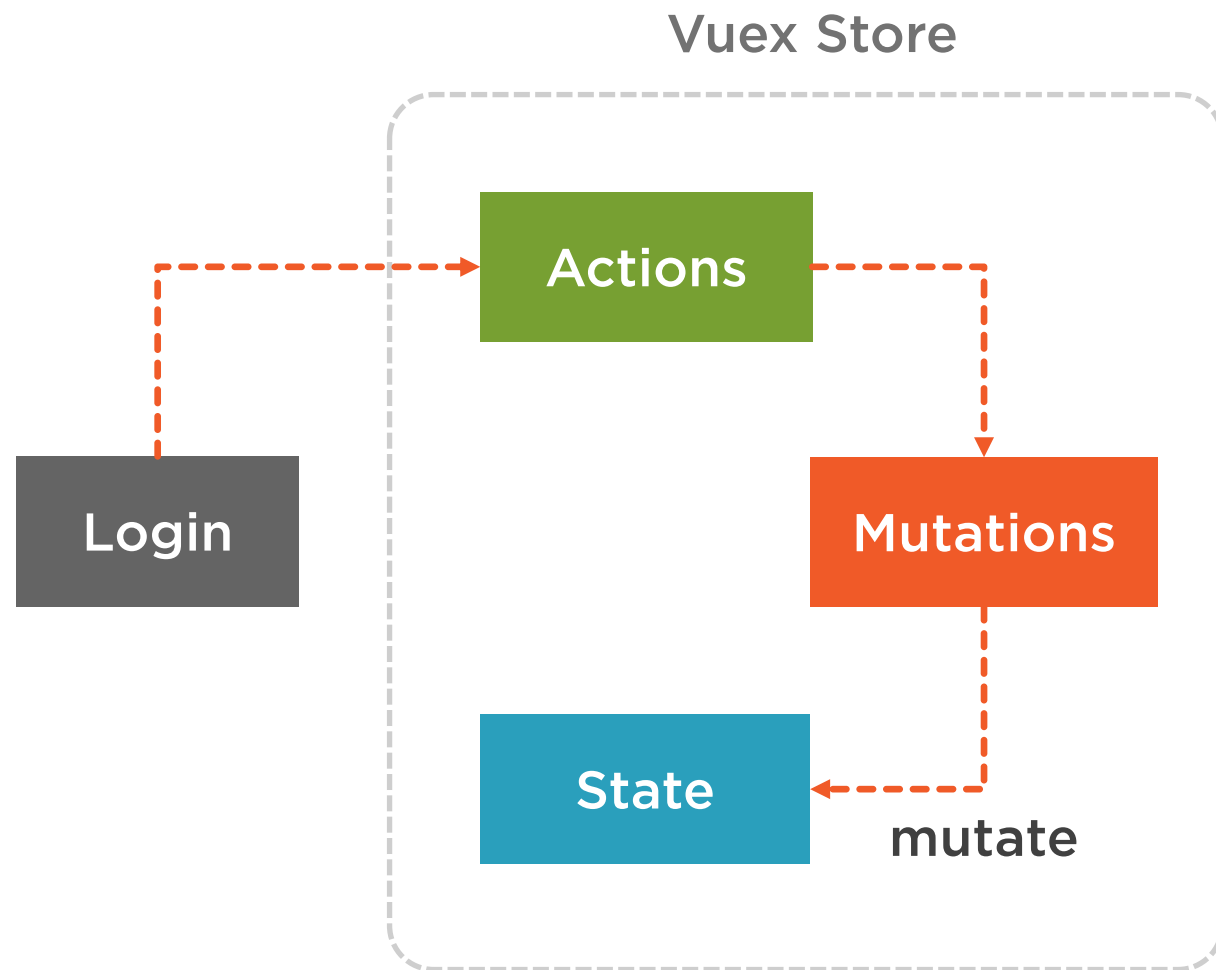
    // ...
  }
}
</script>
```



```
actions: {  
  loginUser (context) {  
    // Login logic here ...  
    context.commit('loginUser')  
  }  
}
```



```
mutations: {  
  loginUser (state) {  
    state.isLoggedIn = true  
  }  
}
```





```
// Vuex Store
```

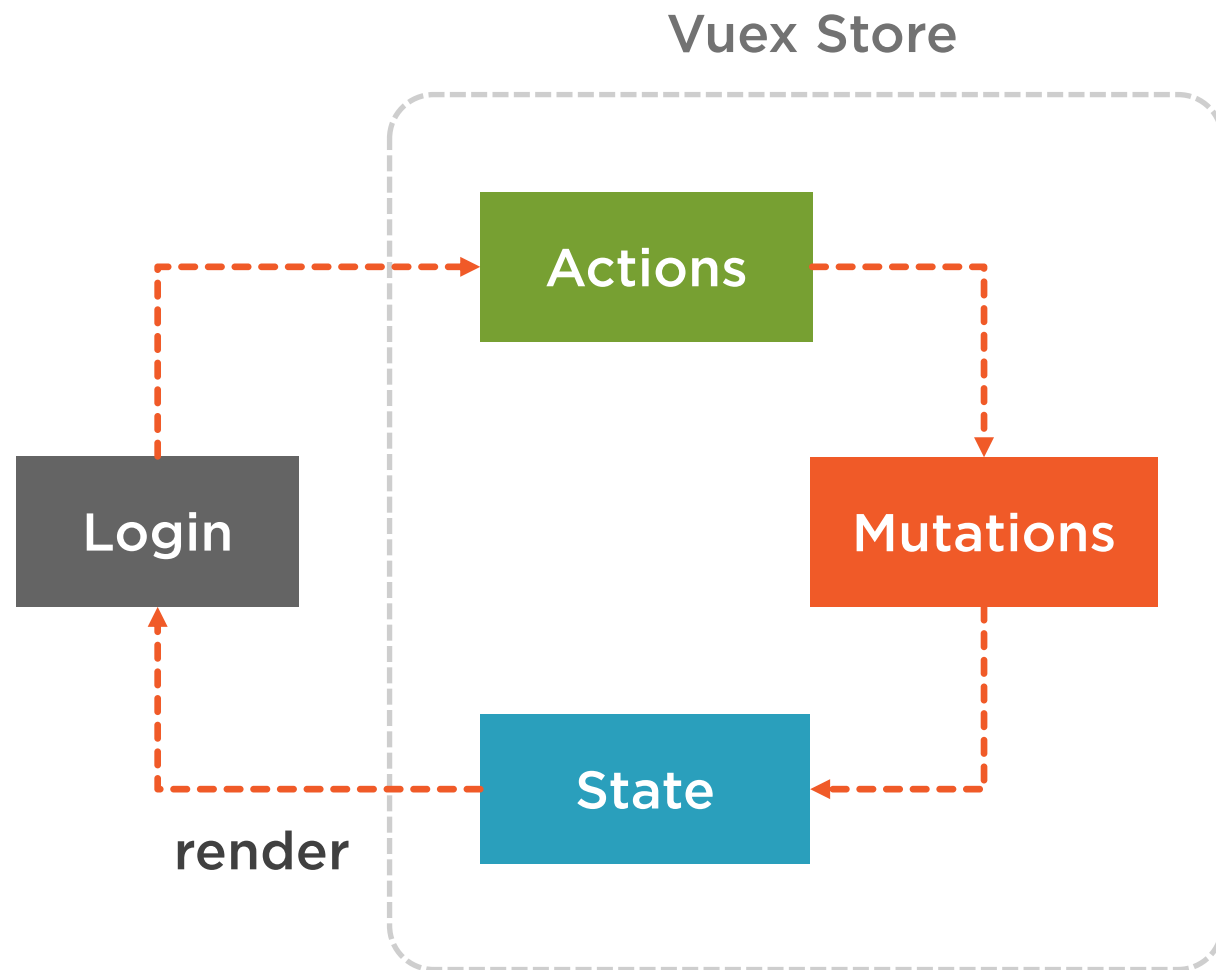
```
state: {  
  isLoggedIn: true  
}
```

```
// Login.vue Component Reacts
```

```
<template>
```

```
  <div v-if="isLoggedIn">  
    // Template markup here...  
  </div>
```

```
</template>
```

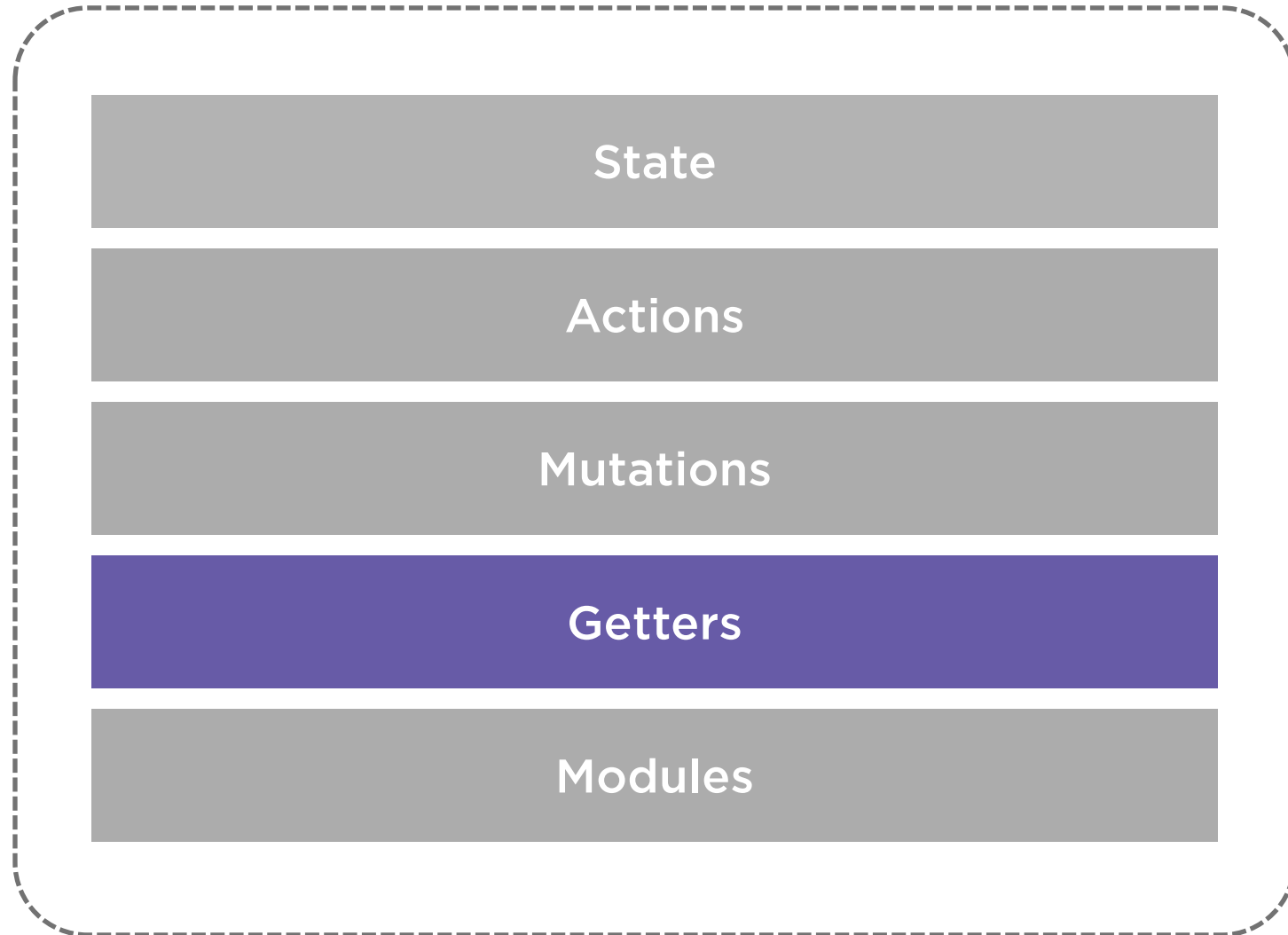


# Getters

---



# Vuex Store: getters



# Vuex Store: getters example

```
const store = new Vuex.Store({  
  state: {  
    firstName: 'Test',  
    lastName: 'User'  
  },  
  getters: { ... },  
  actions: { ... },  
  mutations: { ... }  
})
```



# Vuex Store: getters example

```
const store = new Vuex.Store({  
  state: { ... },  
  getters: {  
    fullName: state => {  
      return state.firstName + ' ' + state.lastName  
    }  
  },  
  actions: { ... },  
  mutations: { ... }  
})
```

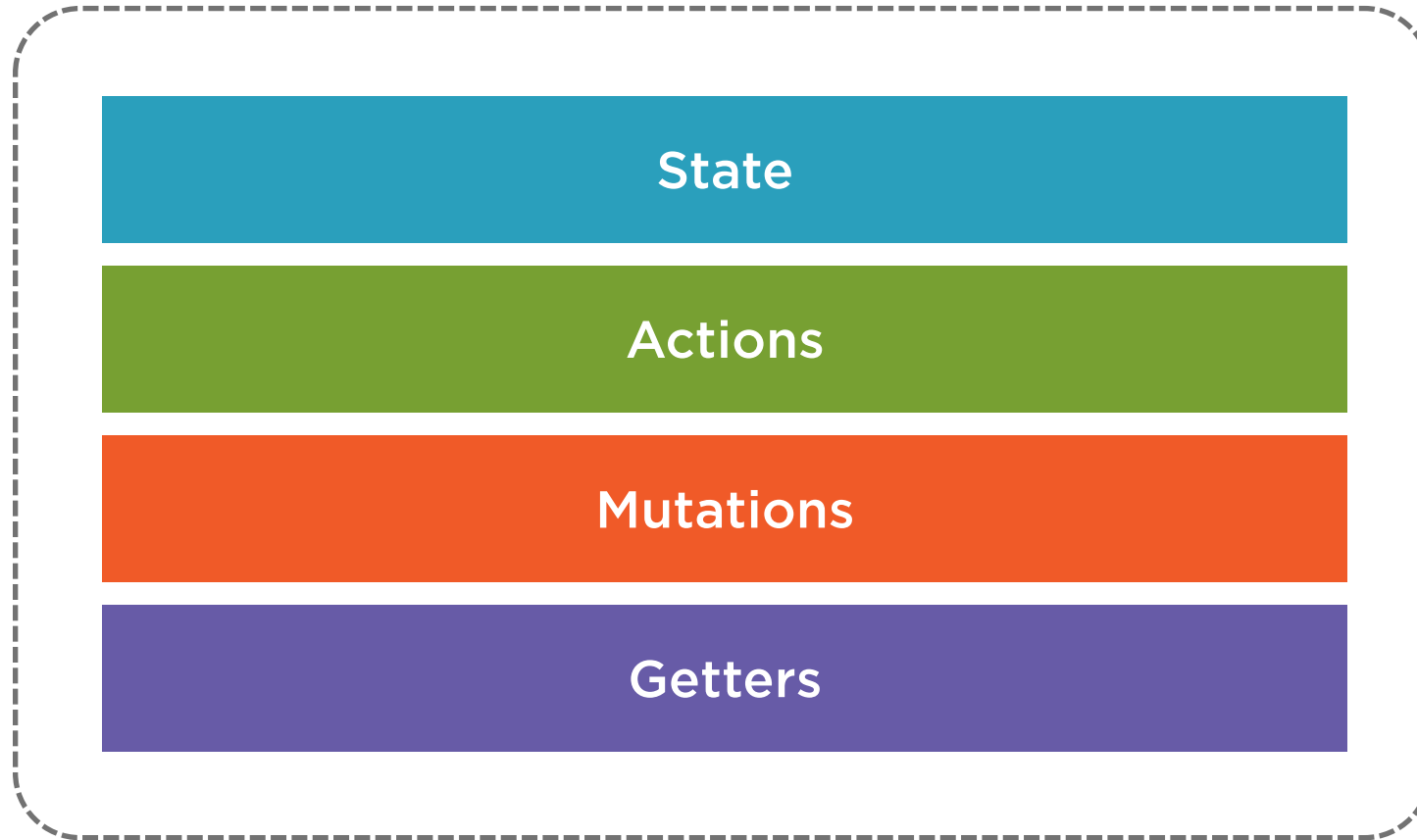


# Modules

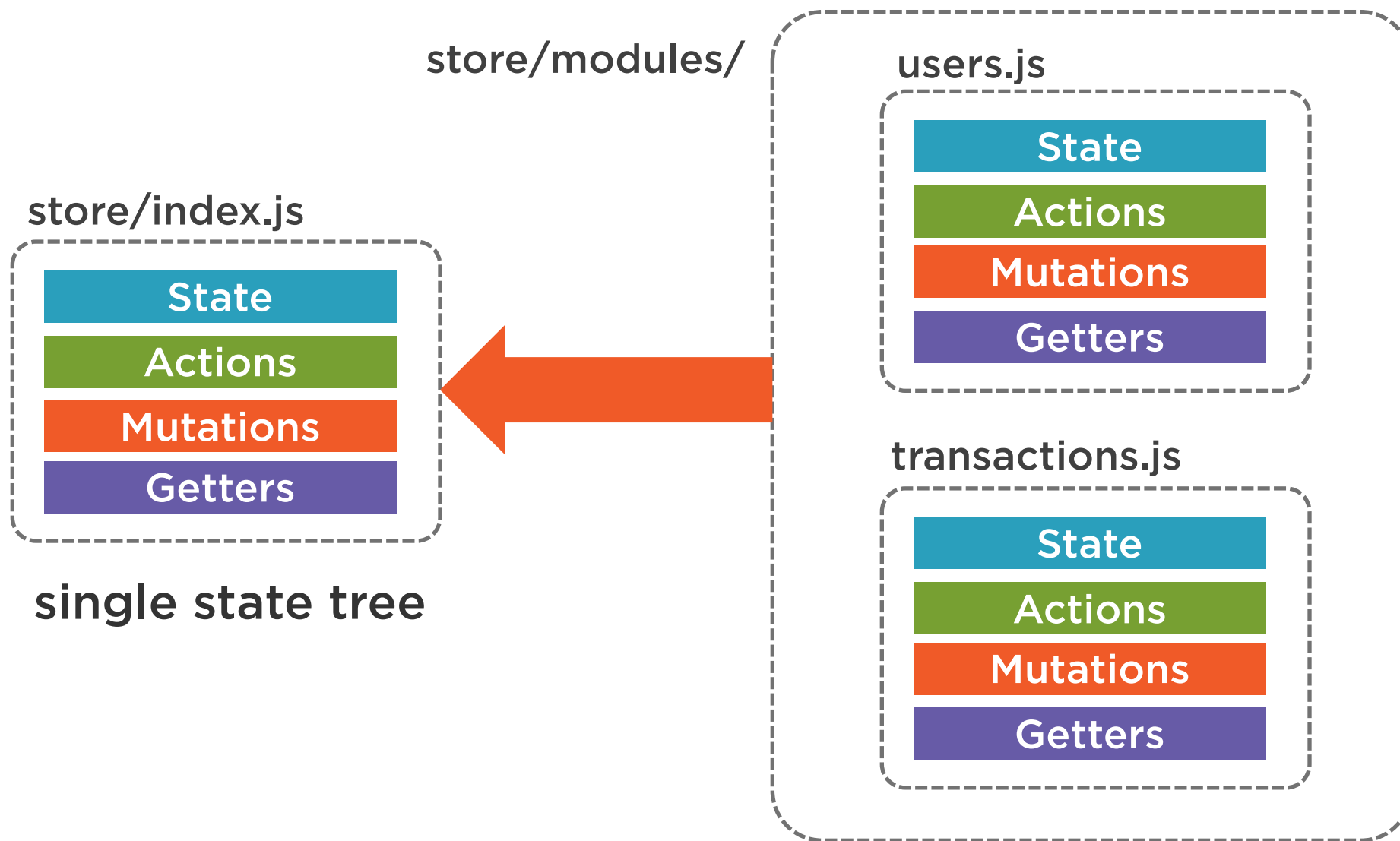
---



# Vuex Store – One Big Object



# Vuex Store – Modules





[vuex.vuejs.org](https://vuex.vuejs.org)



# Define Our Store

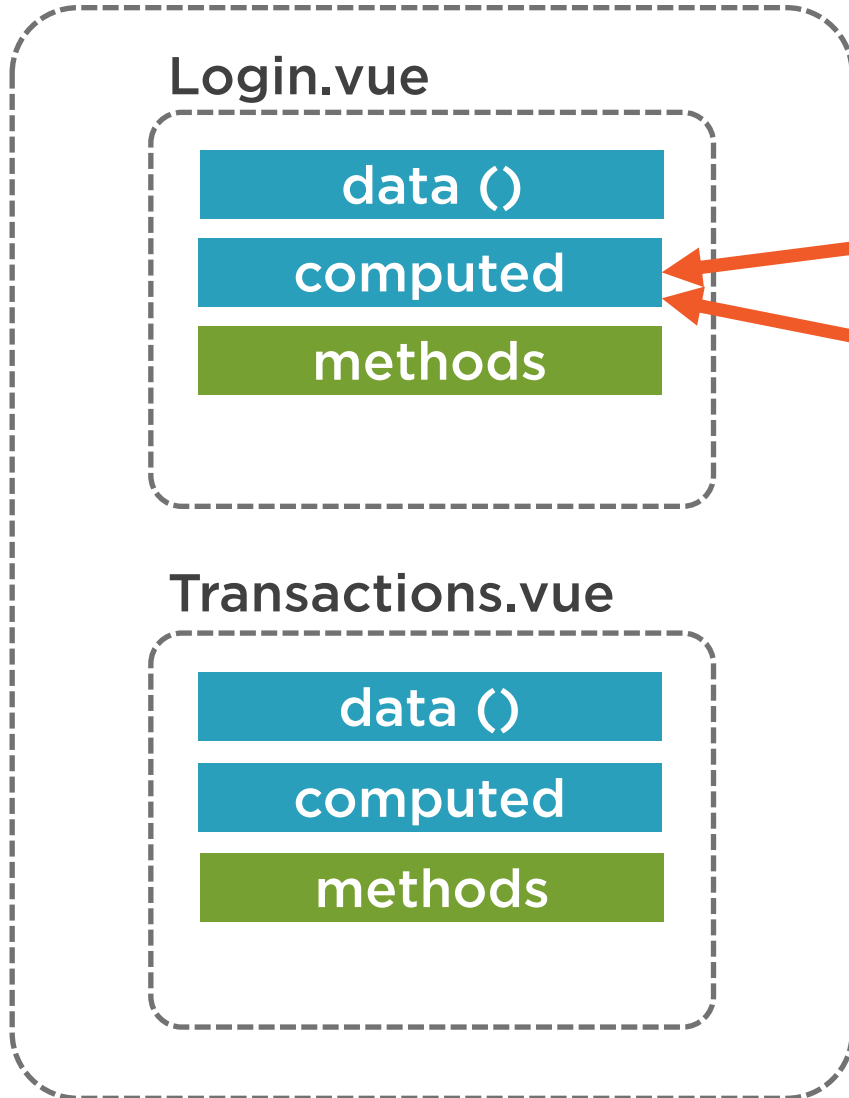
---



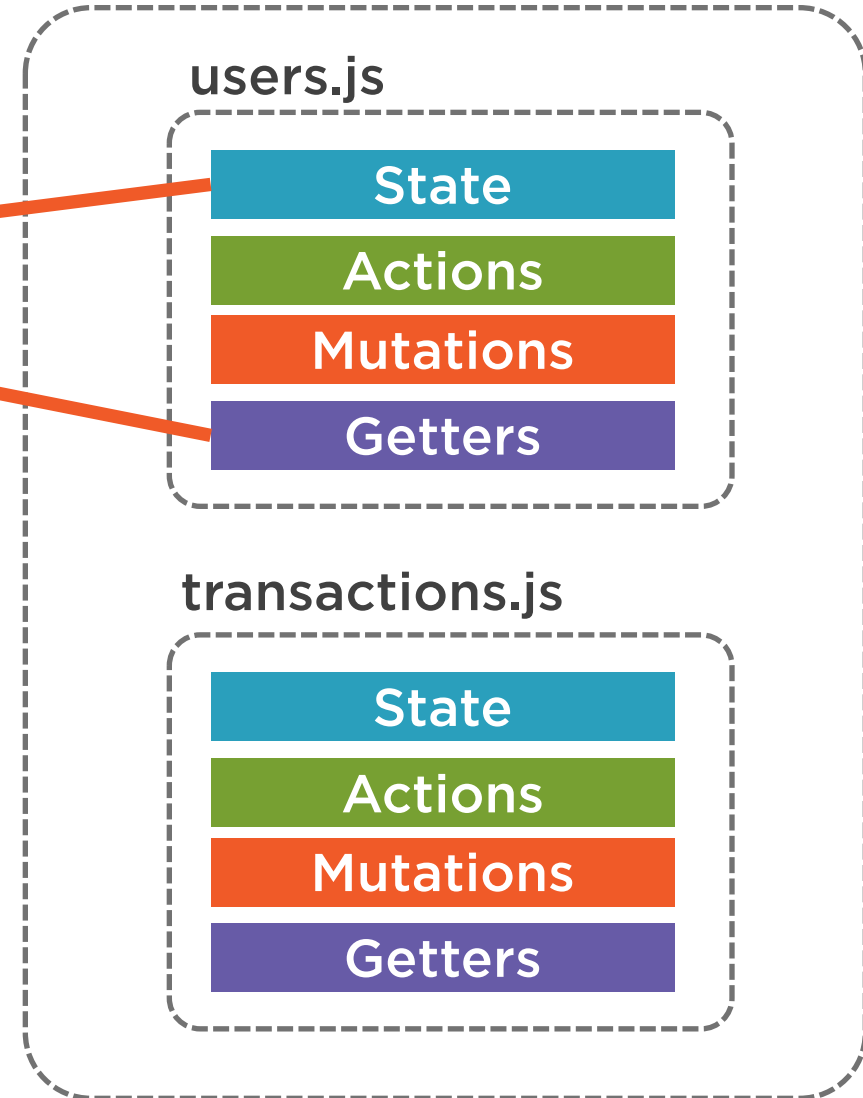
Does everything have to go  
into the Vuex state store?



# Vue Components



# Vuex Store



# Demo



## Modules

- User
- Transactions

## Review code

- State
- Actions
- Mutations

## Modify Login.vue



# Demo



## Modify Login.vue

- Use Vuex store vs. local data
- Access logged in user globally



## Next Steps



**Do you have the Vue dev tools installed?**

### **Getting data from the API**

- Introduce Axios
- Call into the API from our Vuex store

