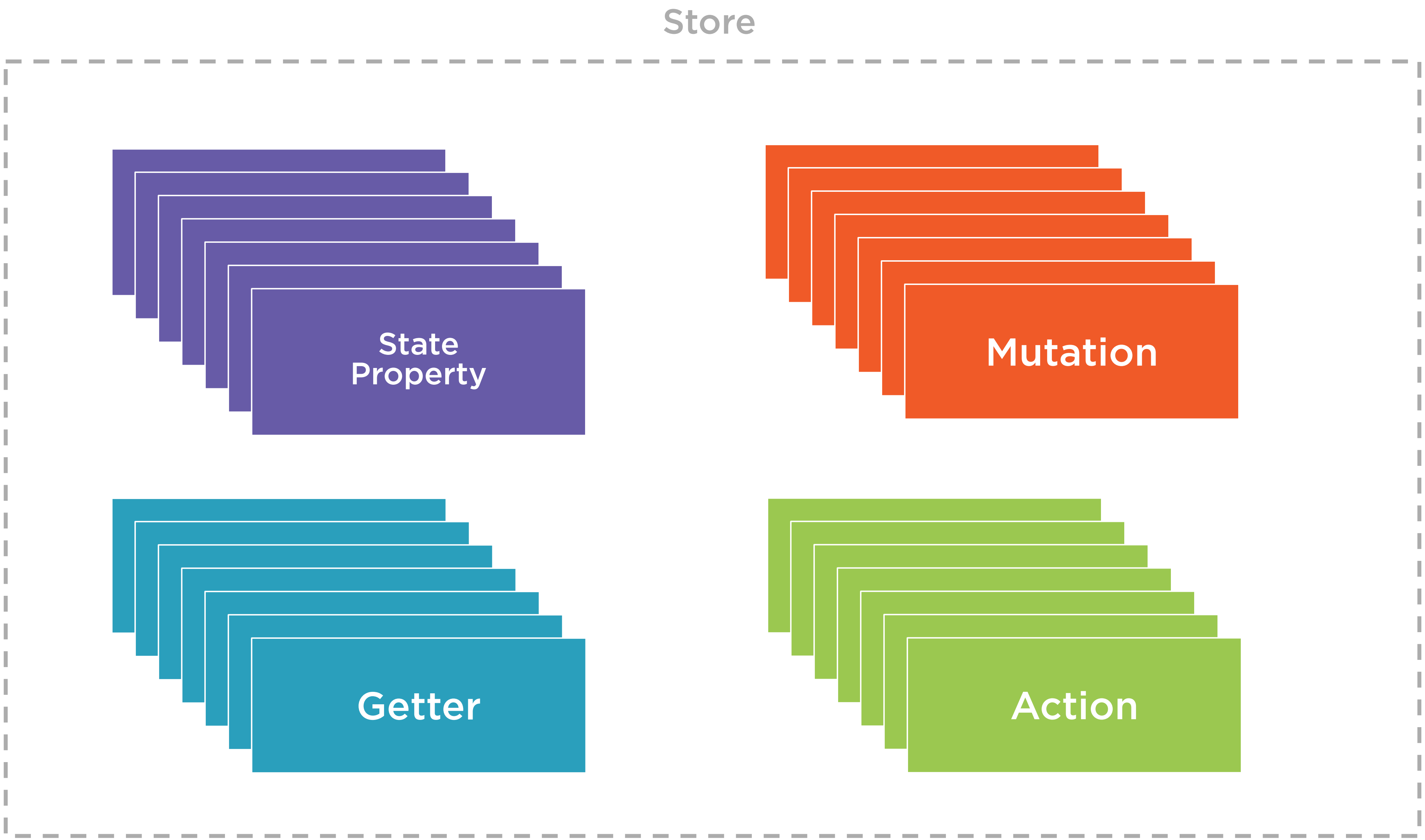
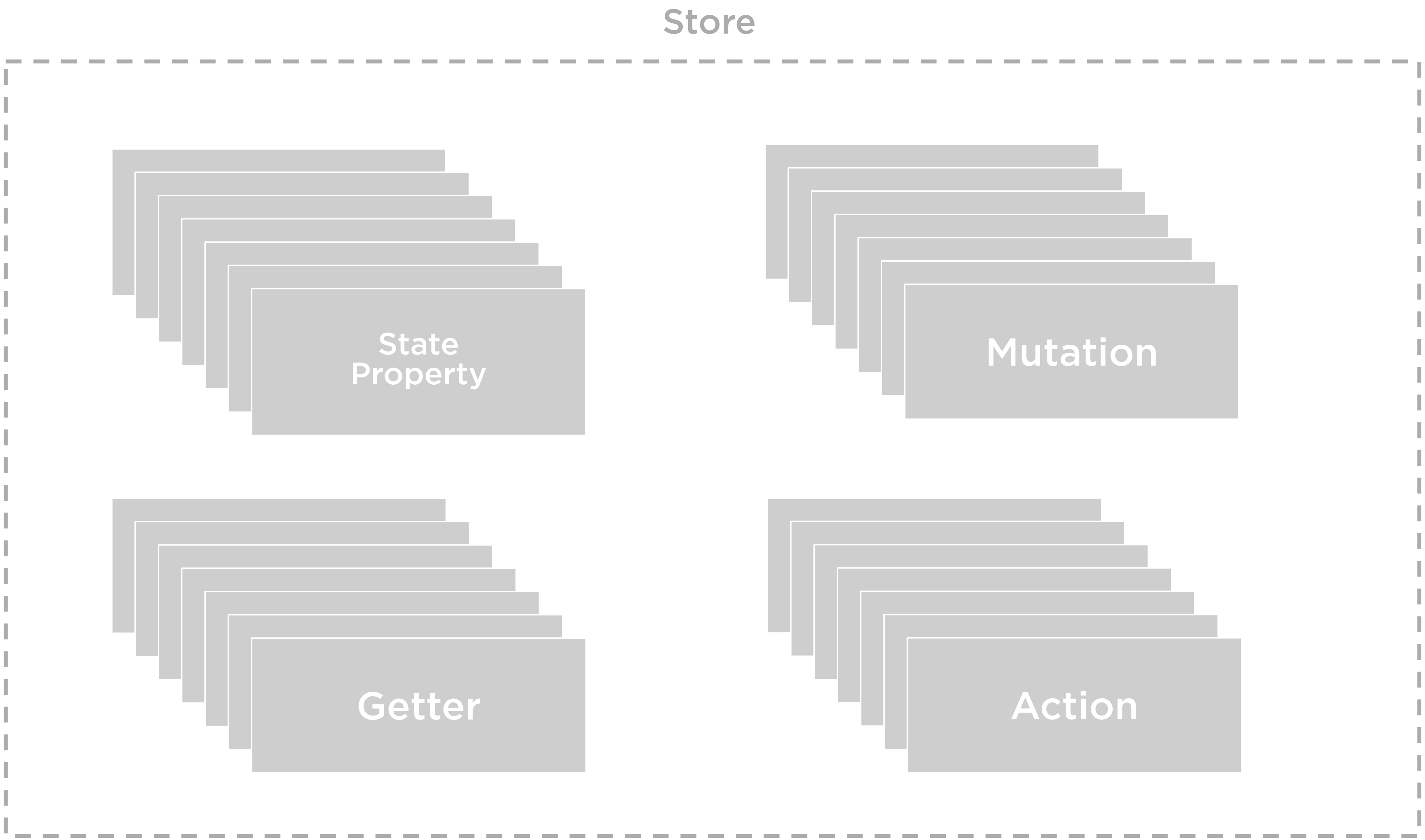
Organizing a Vuex Store With Modules

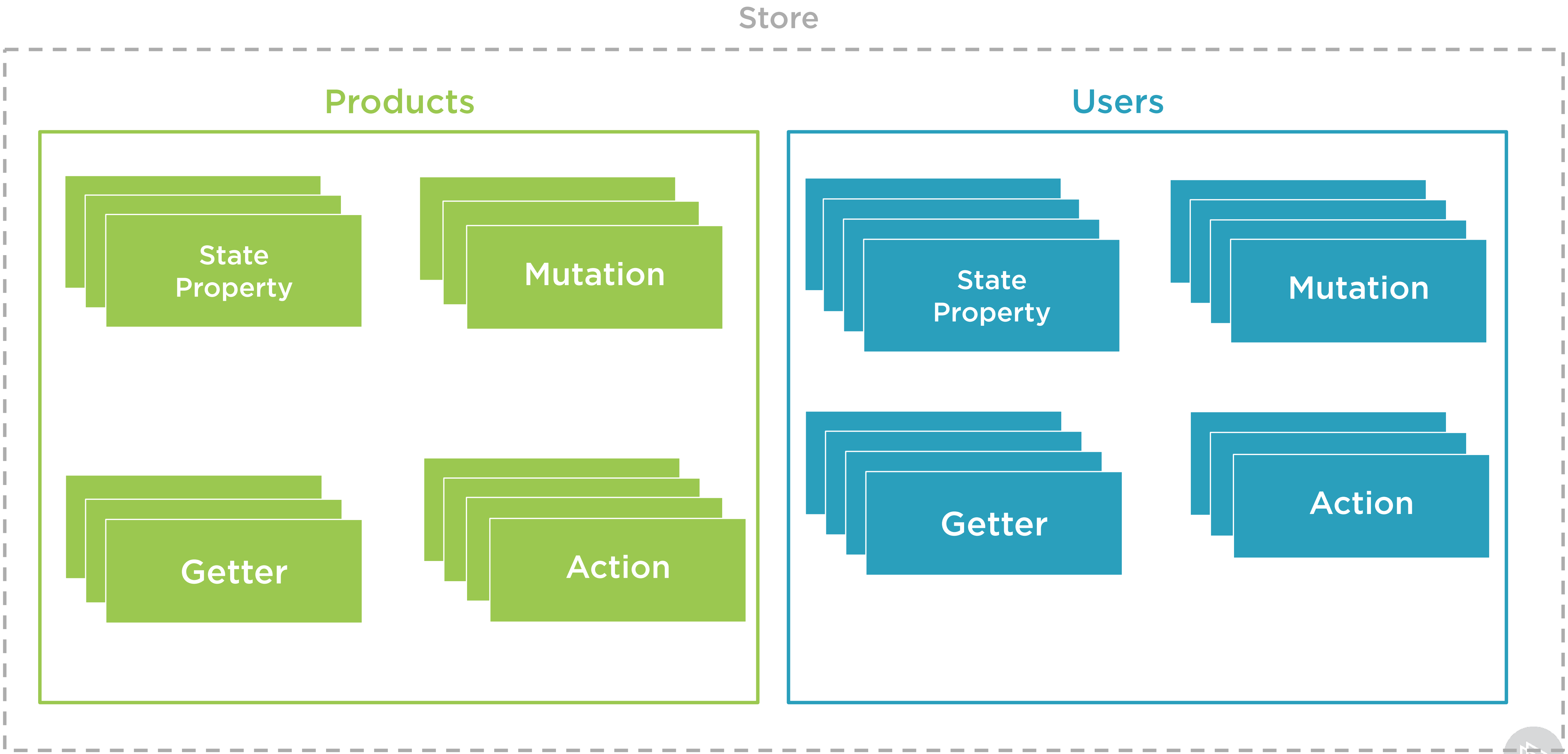
Jim Cooper
SOFTWARE ENGINEER

@jimthecoop   jcoop.io


Store
State Property
Mutation
Getter
Action

What are Vuex Modules?


Store
State Property
Mutation
Getter
Action



Store

Products

State Property

Mutation

Getter

Action

Users

State Property

Mutation

Getter

Action

Creating Vuex Modules

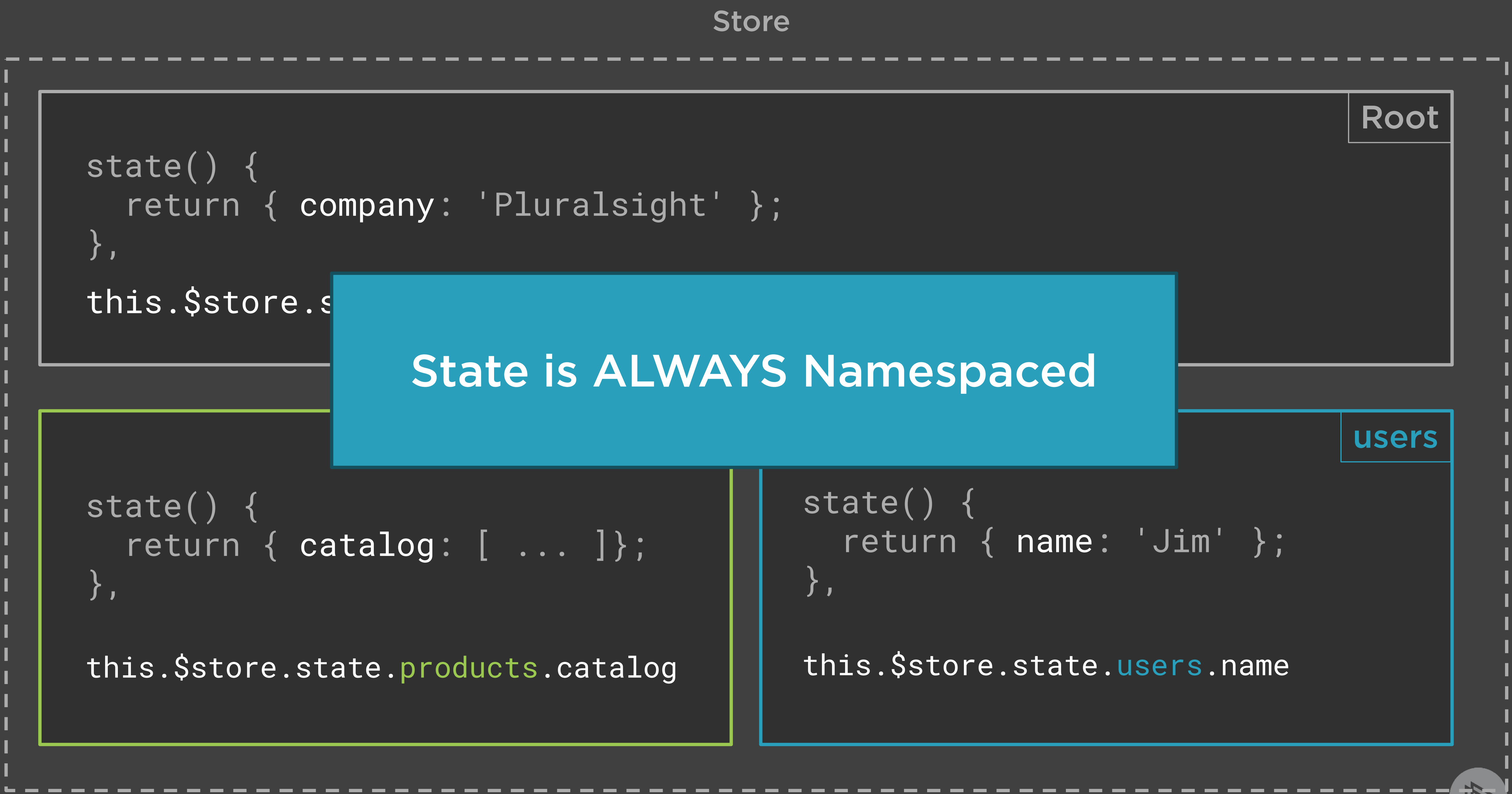
store/index.js

```
import usersModule from './test-module’;

export default new Vuex.Store({
  modules: {
    users: usersModule,
  },
  state() {   return { ... };  },
  mutations: { ... },
  actions: { … },
  getters: { … }
})
```

users-module.js

```
export default {
  state() {
    return { … };
  },
  mutations: { … },
  actions: { … },
  getters: { … },
}
```

State is ALWAYS Namespaced

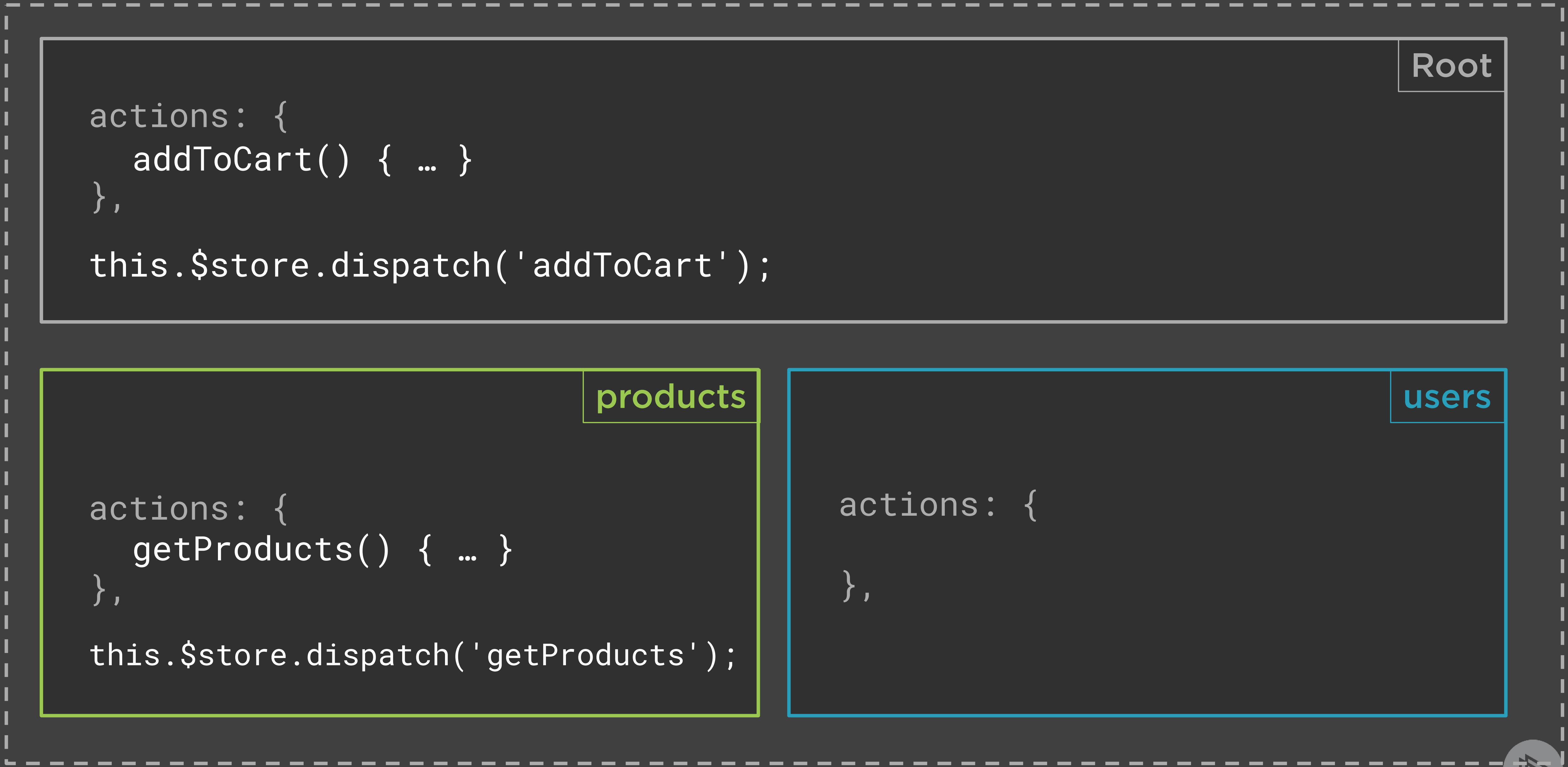


```
state() {
  return { company: 'Pluralsight' };
},

this.$store.s
```

```
state() {
  return { catalog: [ ... ]};
},

this.$store.state.products.catalog
```

```
state() {
  return { name: 'Jim' };
},

this.$store.state.users.name
```

Store



Root

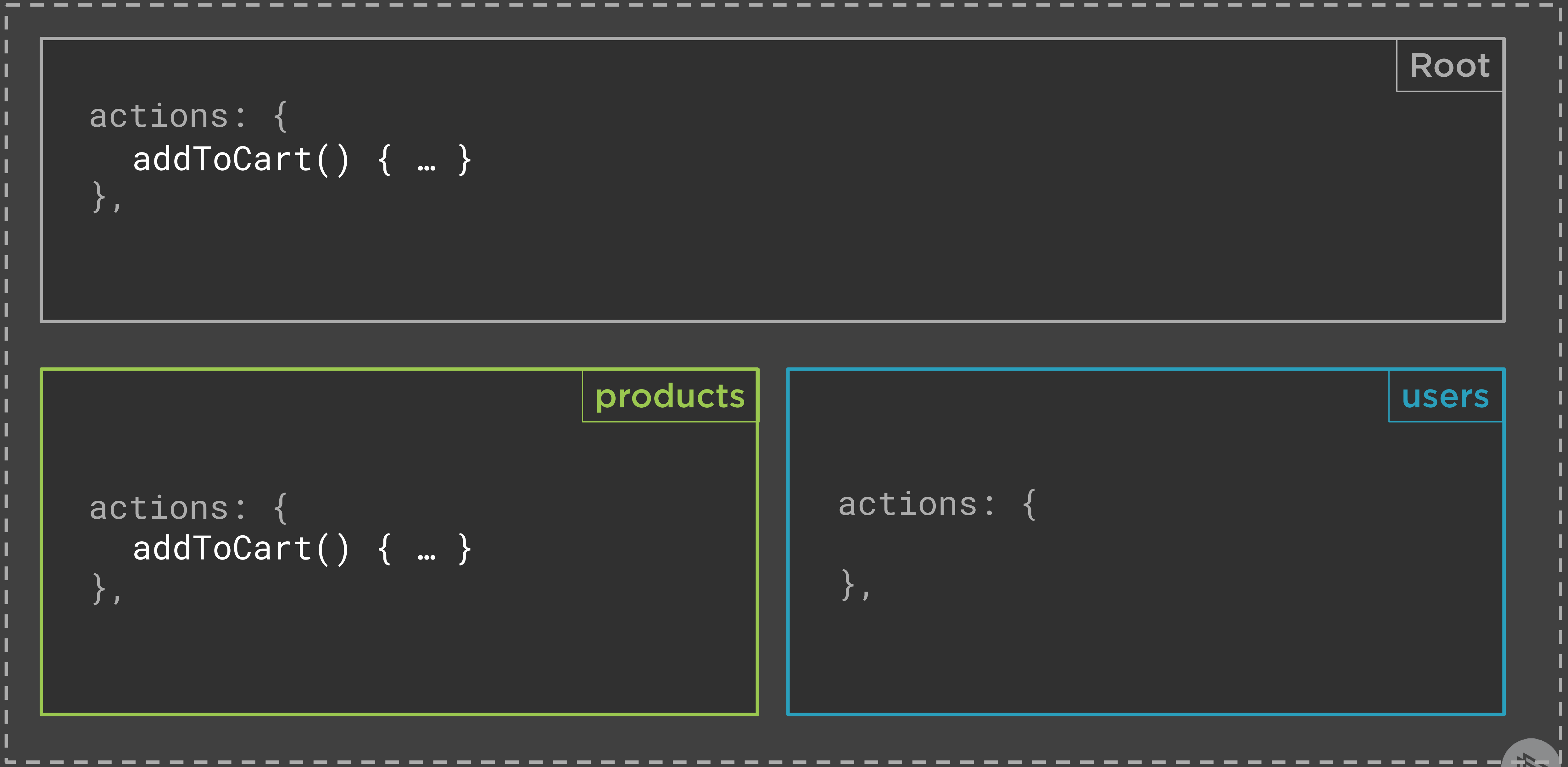
```
actions: {
  addToCart() { … }
},

this.$store.dispatch('addToCart');
```

products

```
actions: {
  getProducts() { … }
},

this.$store.dispatch('getProducts');
```

users

```
actions: {

},
```



Store

Root

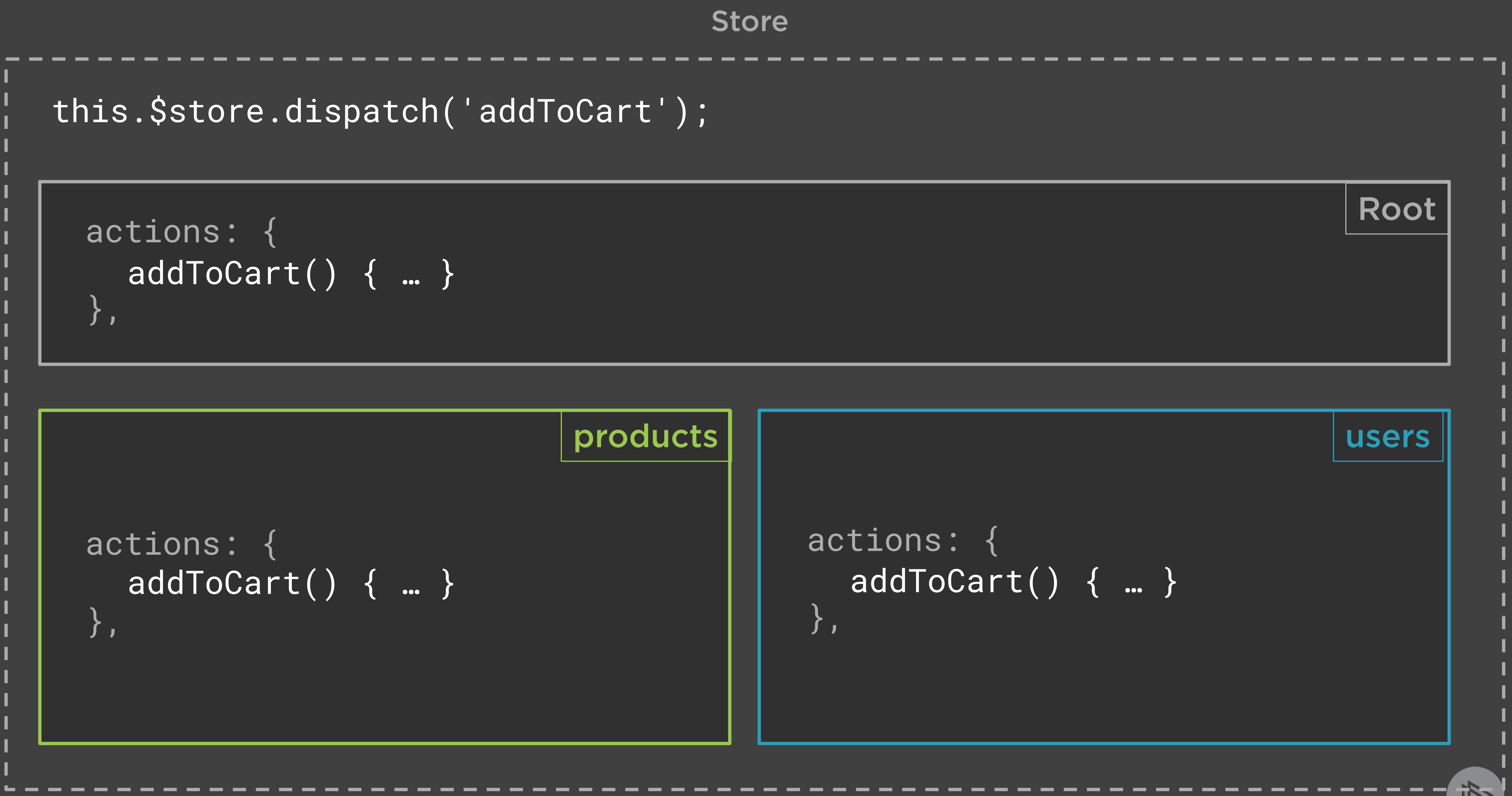
```
actions: {
  addToCart() { … }
},
```

products

```
actions: {
  addToCart() { … }
},
```

users

```
actions: {

},
```



Store

```
this.$store.dispatch('addToCart');
```

Root

```
actions: {
  addToCart() { … }
},
```

products

```
actions: {
  addToCart() { … }
},
```

users

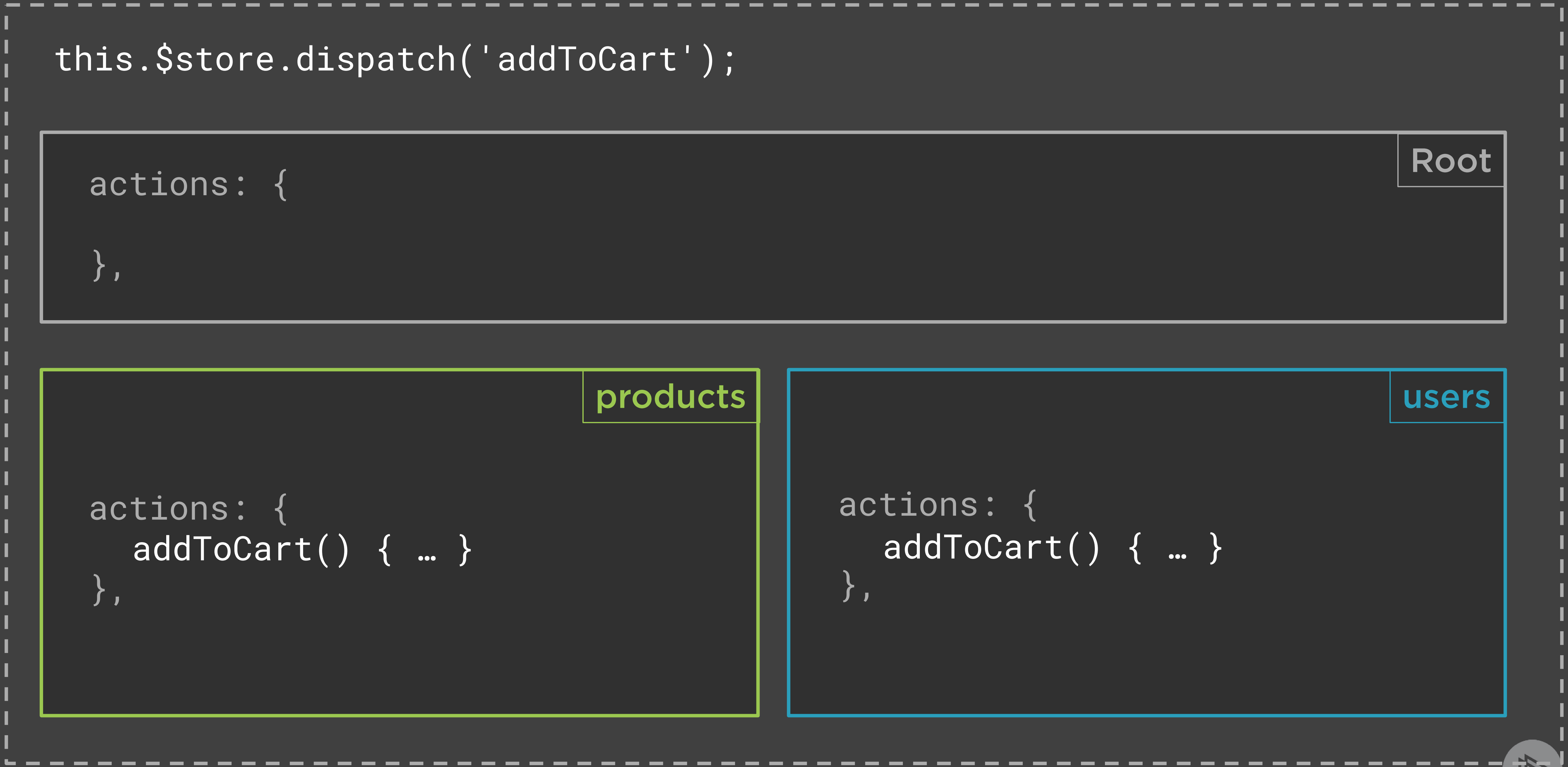
```
actions: {
  addToCart() { … }
},
```

Forcing a Module to be Namespaced

products.js

```
export default {
  namespaced: true,
  state() {
    return { ... };
  },
  mutations: { ... },
  actions: { ... },
  getters: { ... },
}
```

Store



```
this.$store.dispatch('addToCart');
```

Root

```
actions: {

},
```

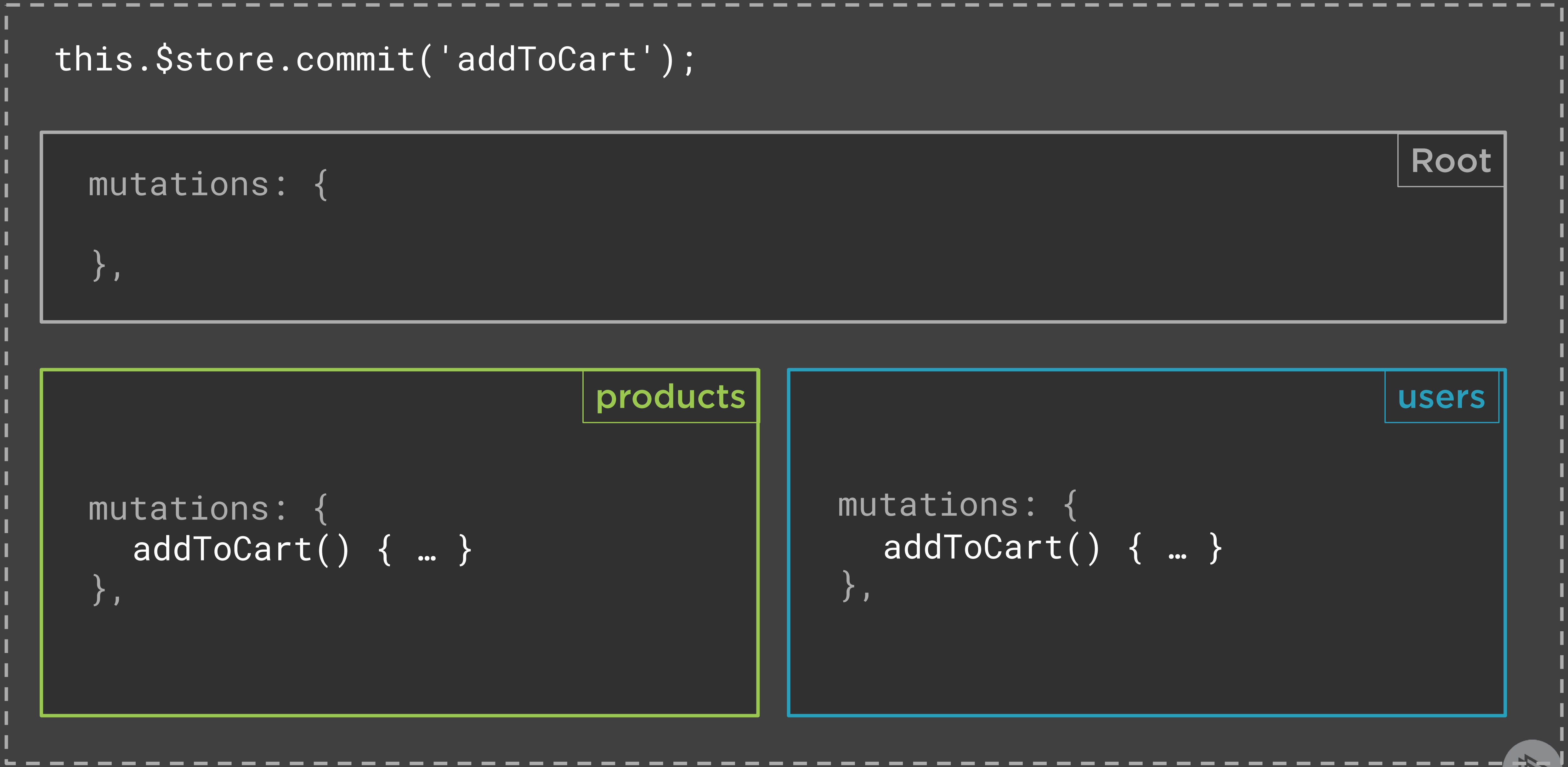
products

```
actions: {
  addToCart() { … }
},
```

users

```
actions: {
  addToCart() { … }
},
```

Store



```
this.$store.commit('addToCart');
```

Root

```
mutations: {

},
```

products

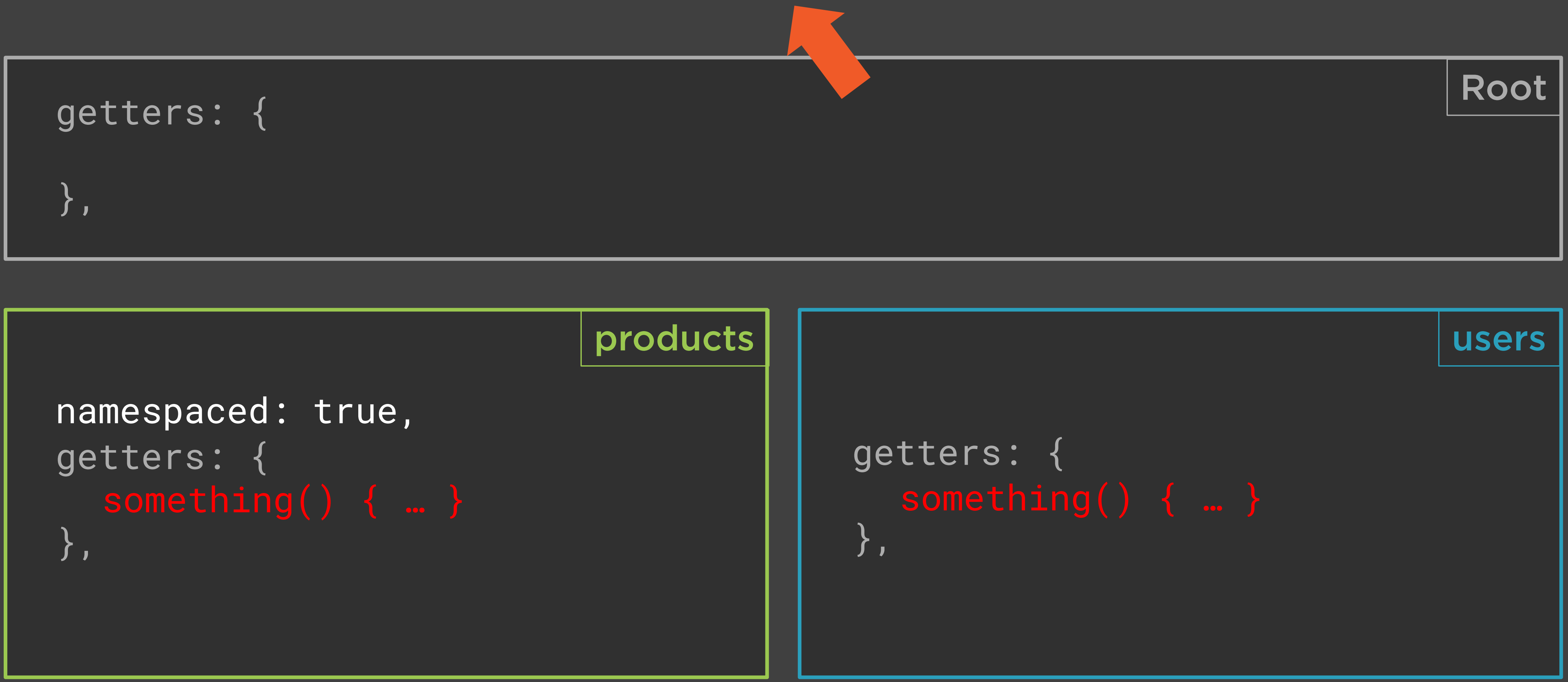
```
mutations: {
  addToCart() { … }
},
```

users

```
mutations: {
  addToCart() { … }
},
```

Store

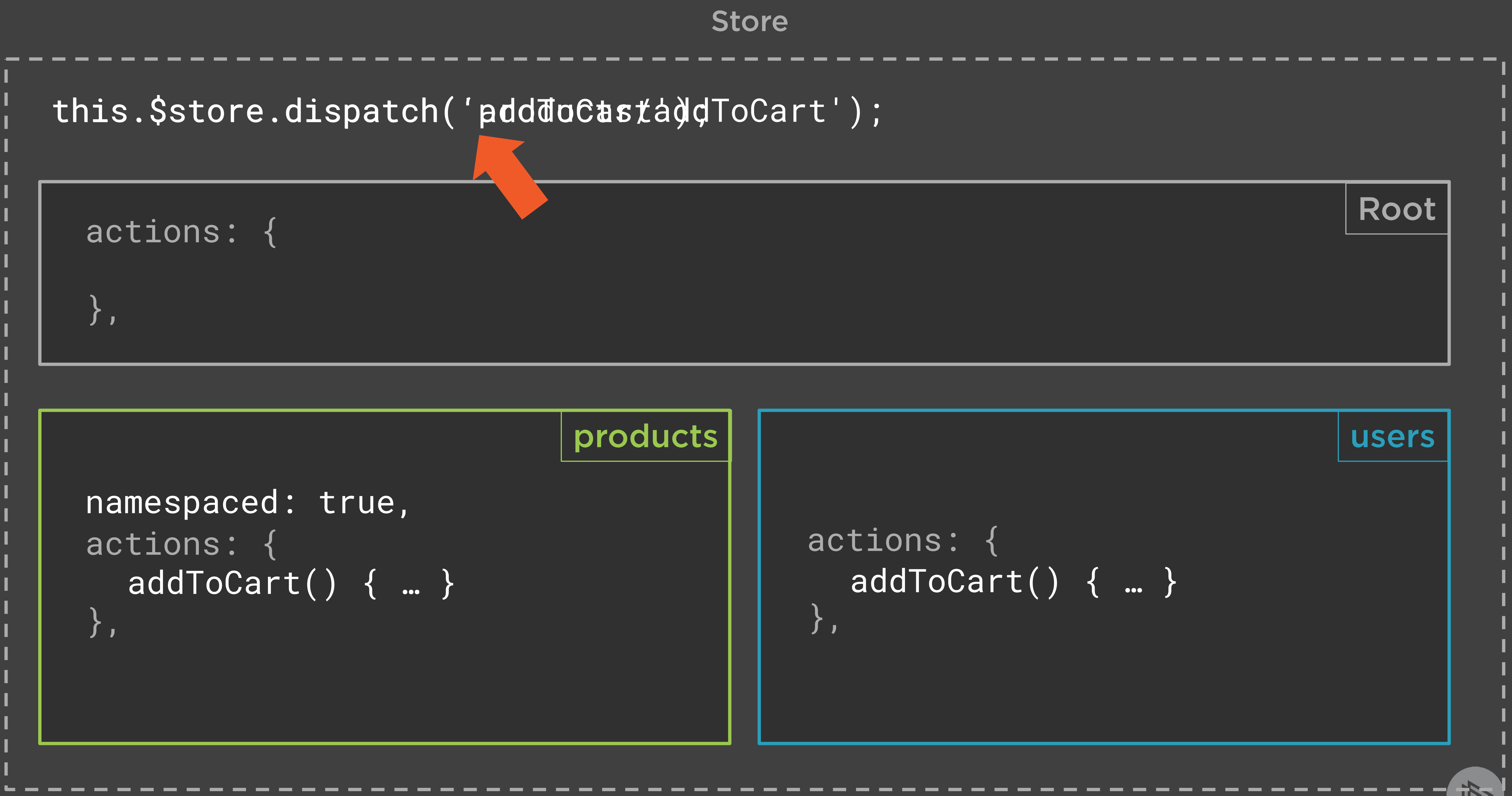
```
let products = this.$store.getters['products/something'];
```



```
getters: {

},
```

```
namespaced: true,
getters: {
  something() { … }
},
```

```
getters: {
  something() { … }
},
```



Store

```
this.$store.dispatch('products/addToCart');
```

Root

```
actions: {

},
```

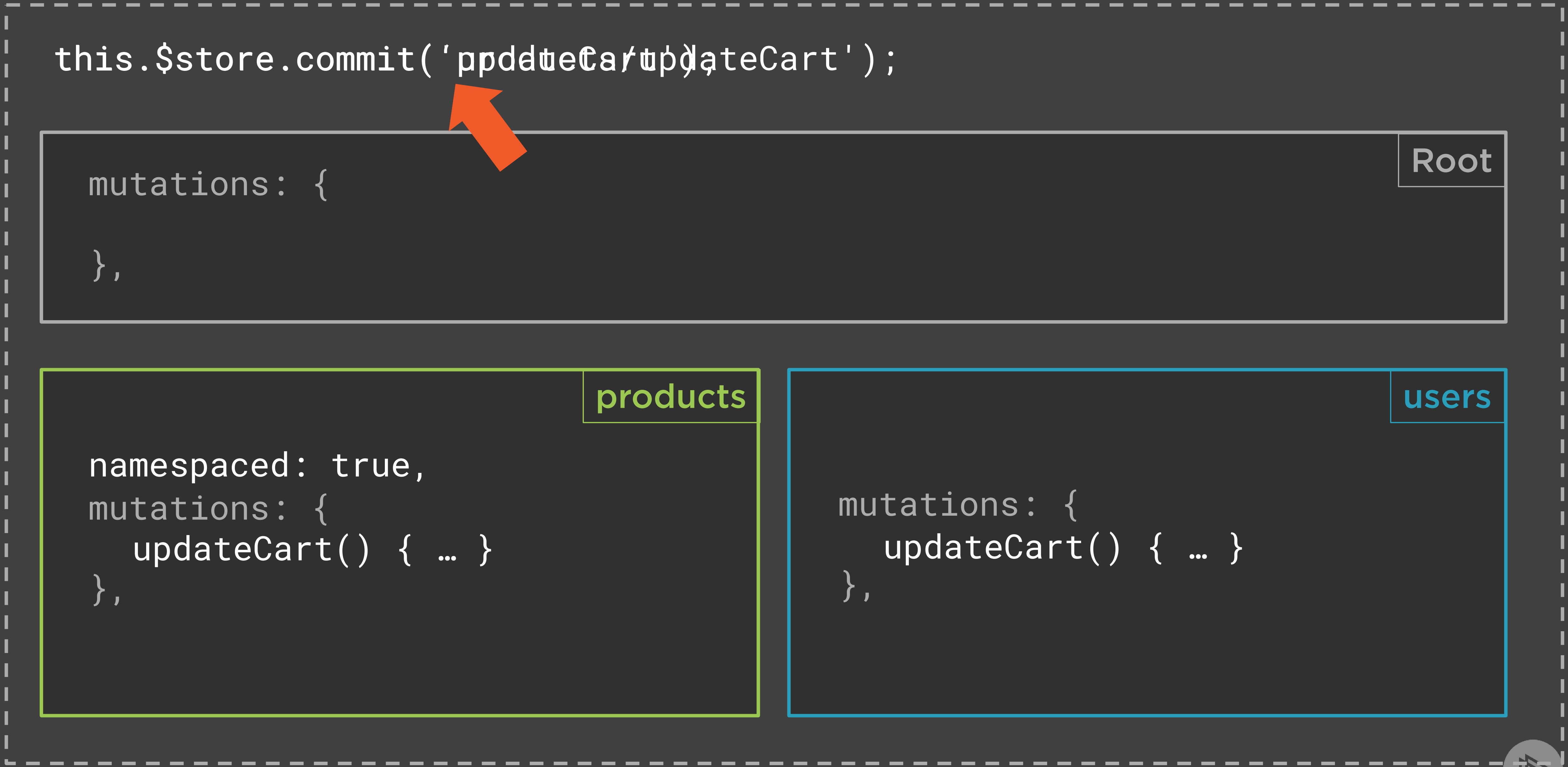
products

```
namespaced: true,
actions: {
  addToCart() { … }
},
```

users

```
actions: {
  addToCart() { … }
},
```

Store



```
this.$store.commit('updateCart');
this.$store.commit('products/updateCart');
```

Root

```
mutations: {

},
```

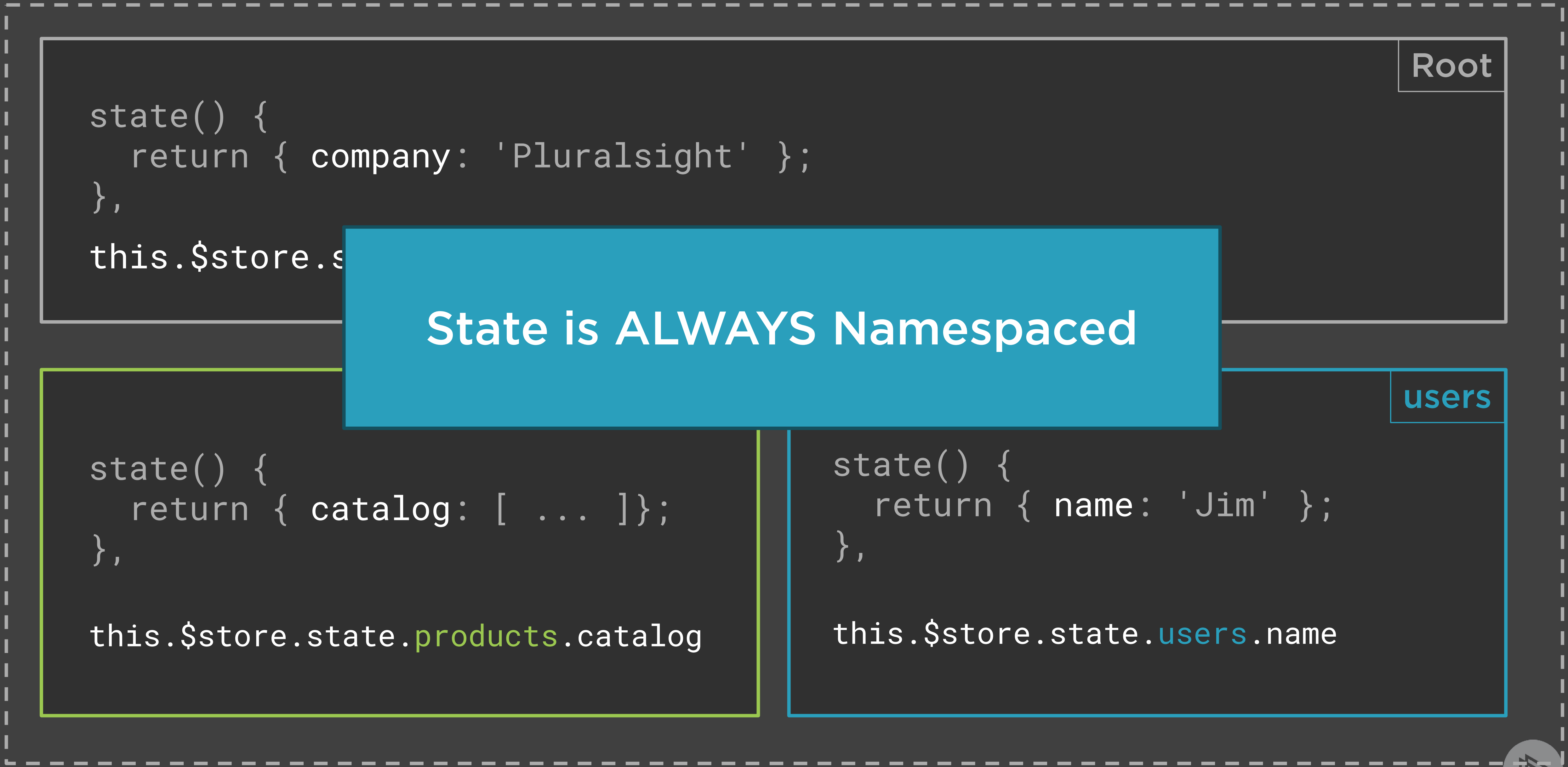
products

```
namespaced: true,
mutations: {
  updateCart() { … }
},
```

users

```
mutations: {
  updateCart() { … }
},
```

Store



Root

```
state() {
  return { company: 'Pluralsight' };
},

this.$store.s
```

State is ALWAYS Namespaced

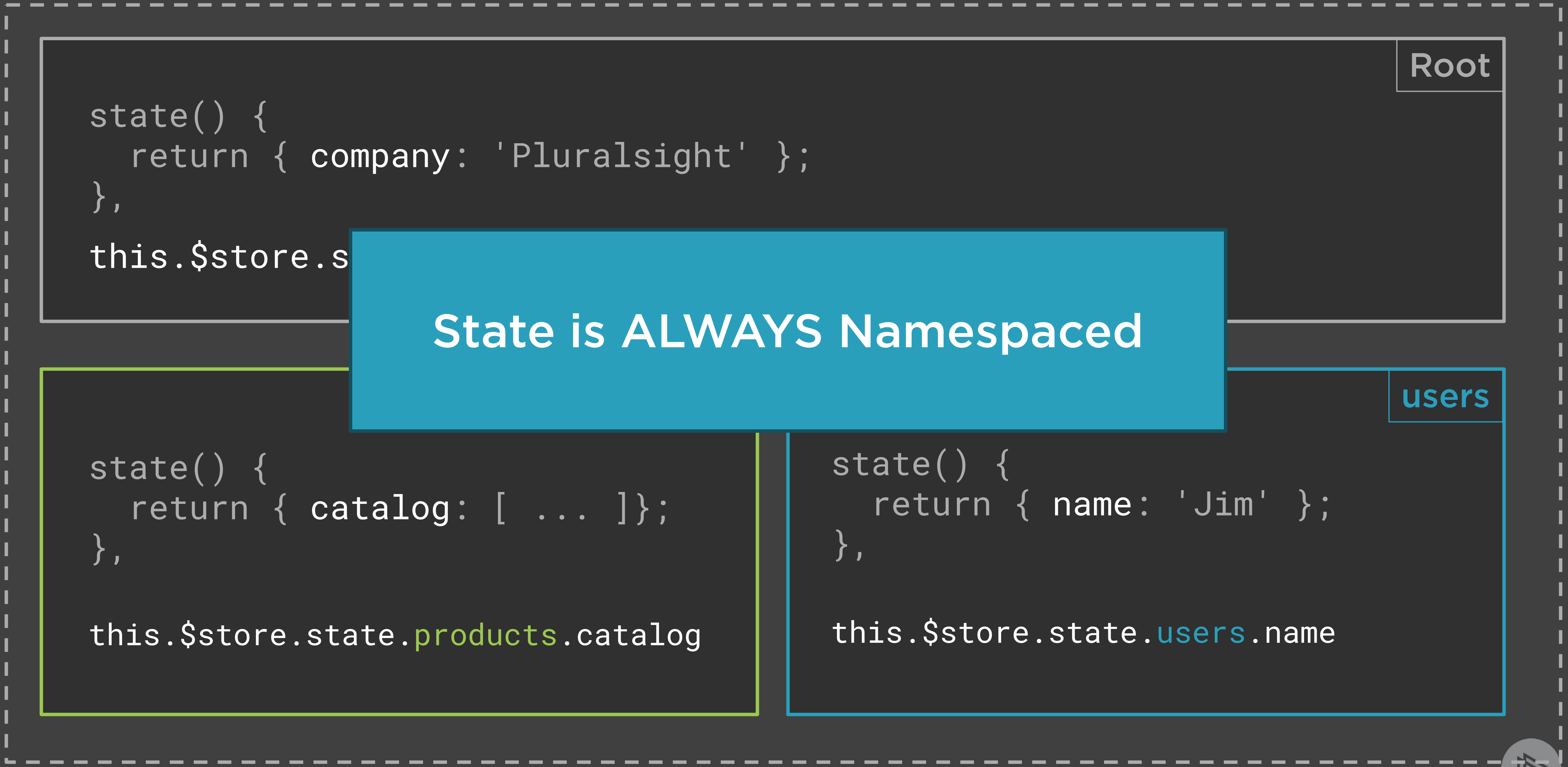
```
state() {
  return { catalog: [ ... ]};
},

this.$store.state.products.catalog
```

users

```
state() {
  return { name: 'Jim' };
},

this.$store.state.users.name
```

Let’s review

Store



Root

```
state() {
  return { company: 'Pluralsight' };
},

this.$store.s
```

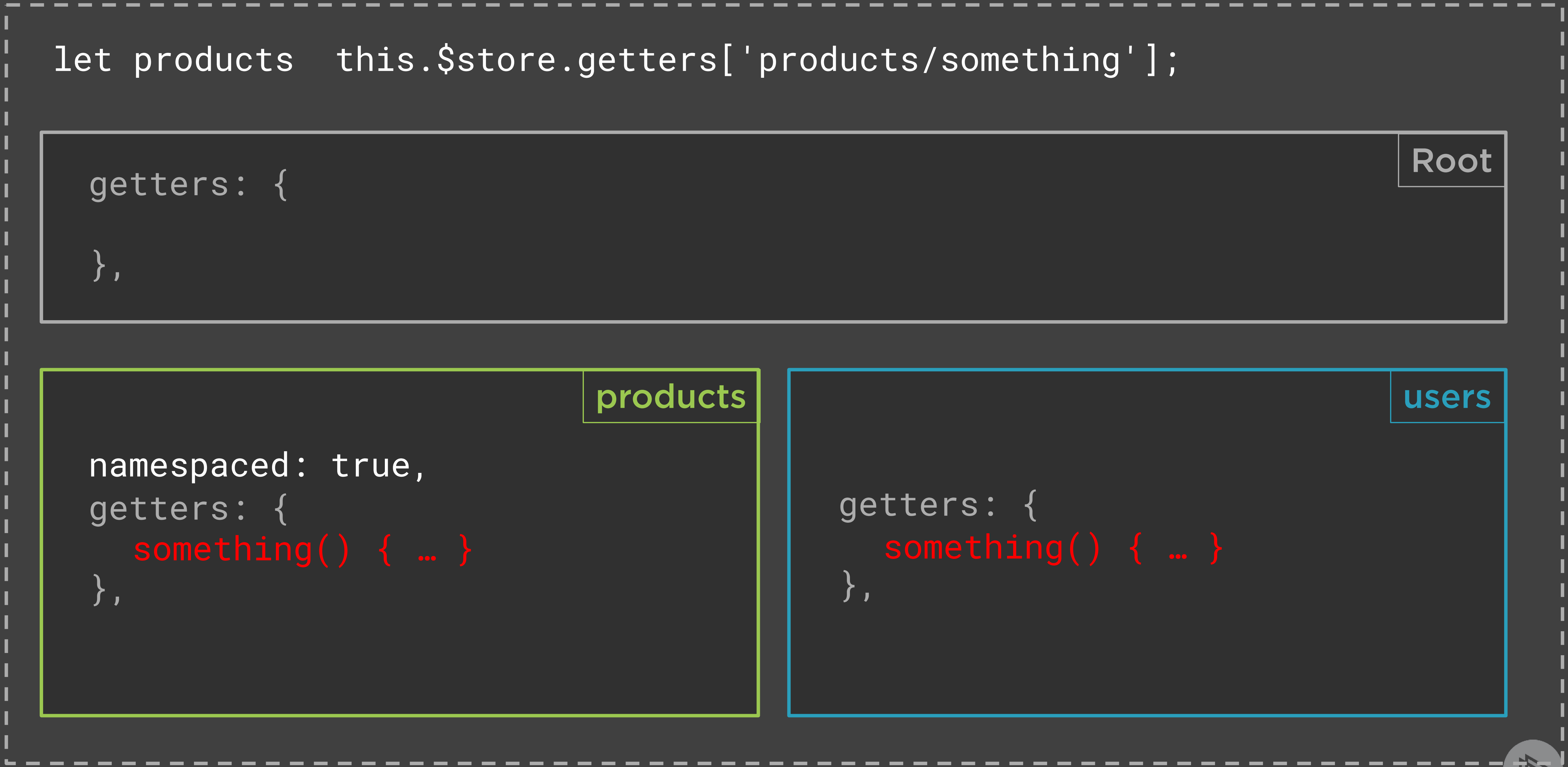
```
state() {
  return { catalog: [ ... ]};
},

this.$store.state.products.catalog
```

users

```
state() {
  return { name: 'Jim' };
},

this.$store.state.users.name
```

State is ALWAYS Namespaced

Store



```
let products  this.$store.getters['products/something'];
```

Root

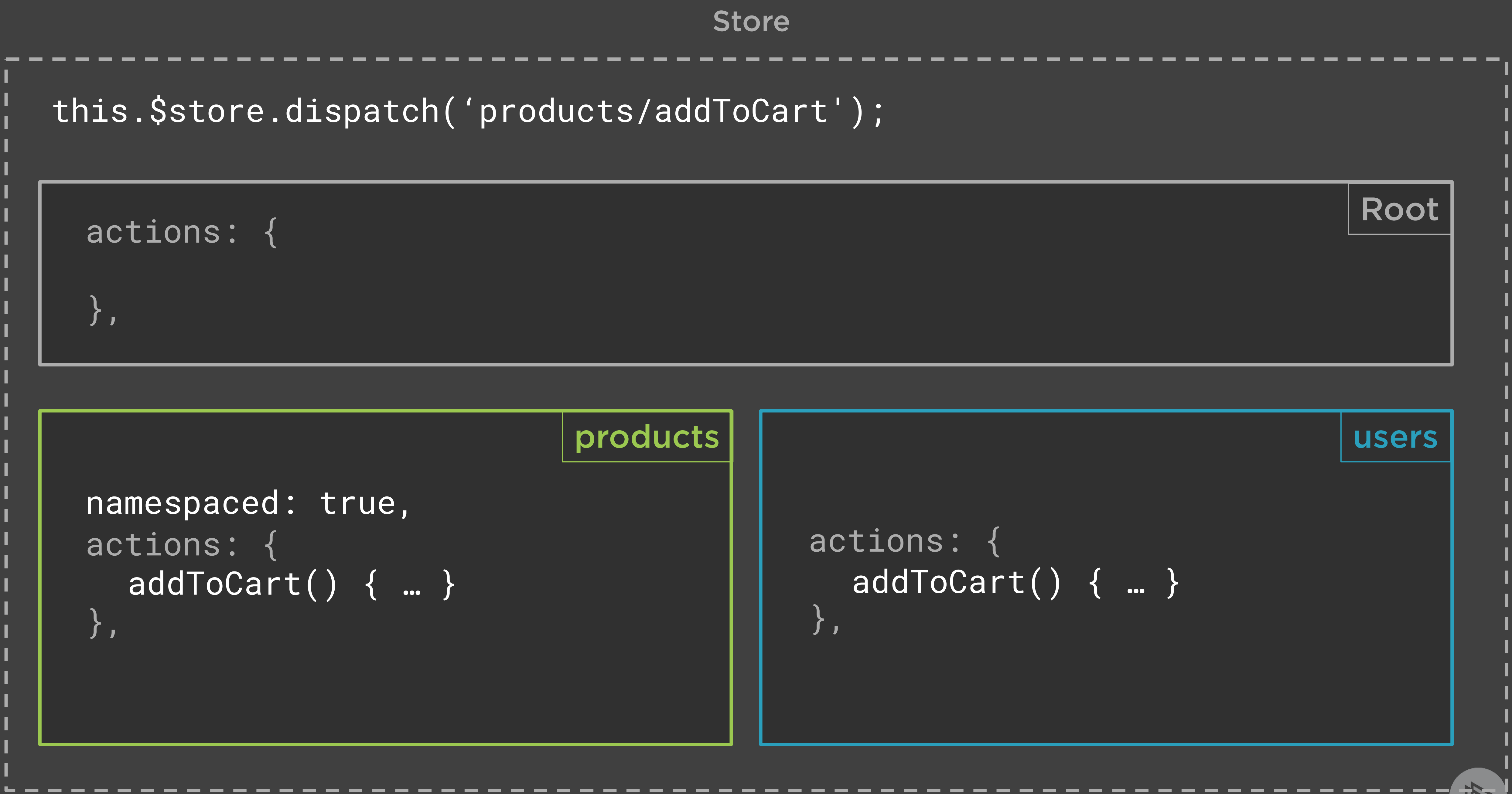
```
getters: {

},
```

products

```
namespaced: true,
getters: {
  something() { … }
},
```

users

```
getters: {
  something() { … }
},
```



Store

```
this.$store.dispatch(‘products/addToCart');
```

Root

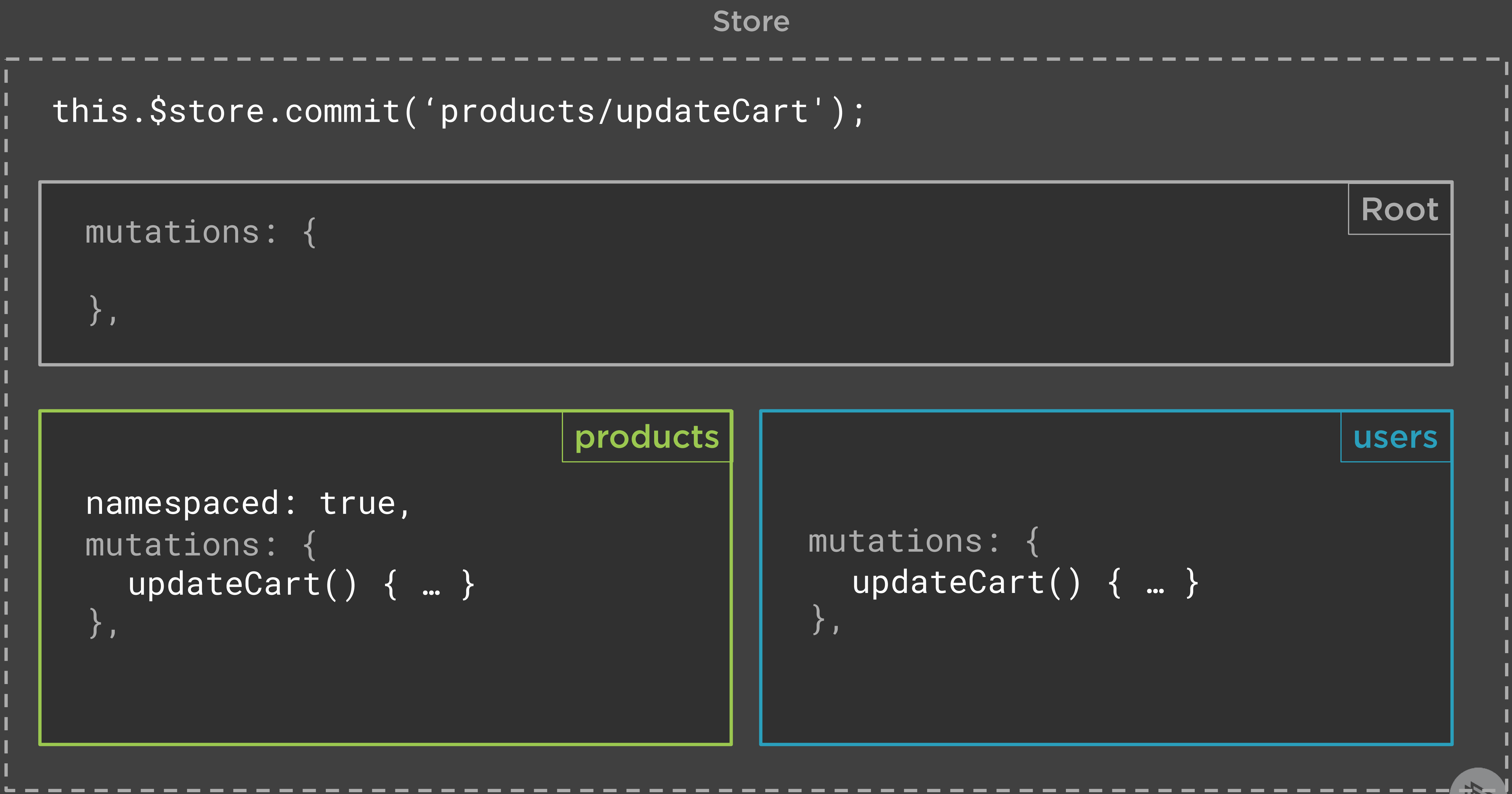
```
actions: {

},
```

products

```
namespaced: true,
actions: {
  addToCart() { … }
},
```

users

```
actions: {
  addToCart() { … }
},
```



Store

```
this.$store.commit(‘products/updateCart');
```

Root

```
mutations: {

},
```

products

```
namespaced: true,
mutations: {
  updateCart() { … }
},
```

users

```
mutations: {
  updateCart() { … }
},
```

Now, let’s go learn about Vuex’s mapping functions!