

VUEX

State management in Vue JS

About ME

- A programmer who love open source
- A backend developer working with PHP (Laravel), Go
- A frontend developer working with Vue and React
- Write some blog post at Medium when I am free
- Working remote developer as Go and Js @ss123h

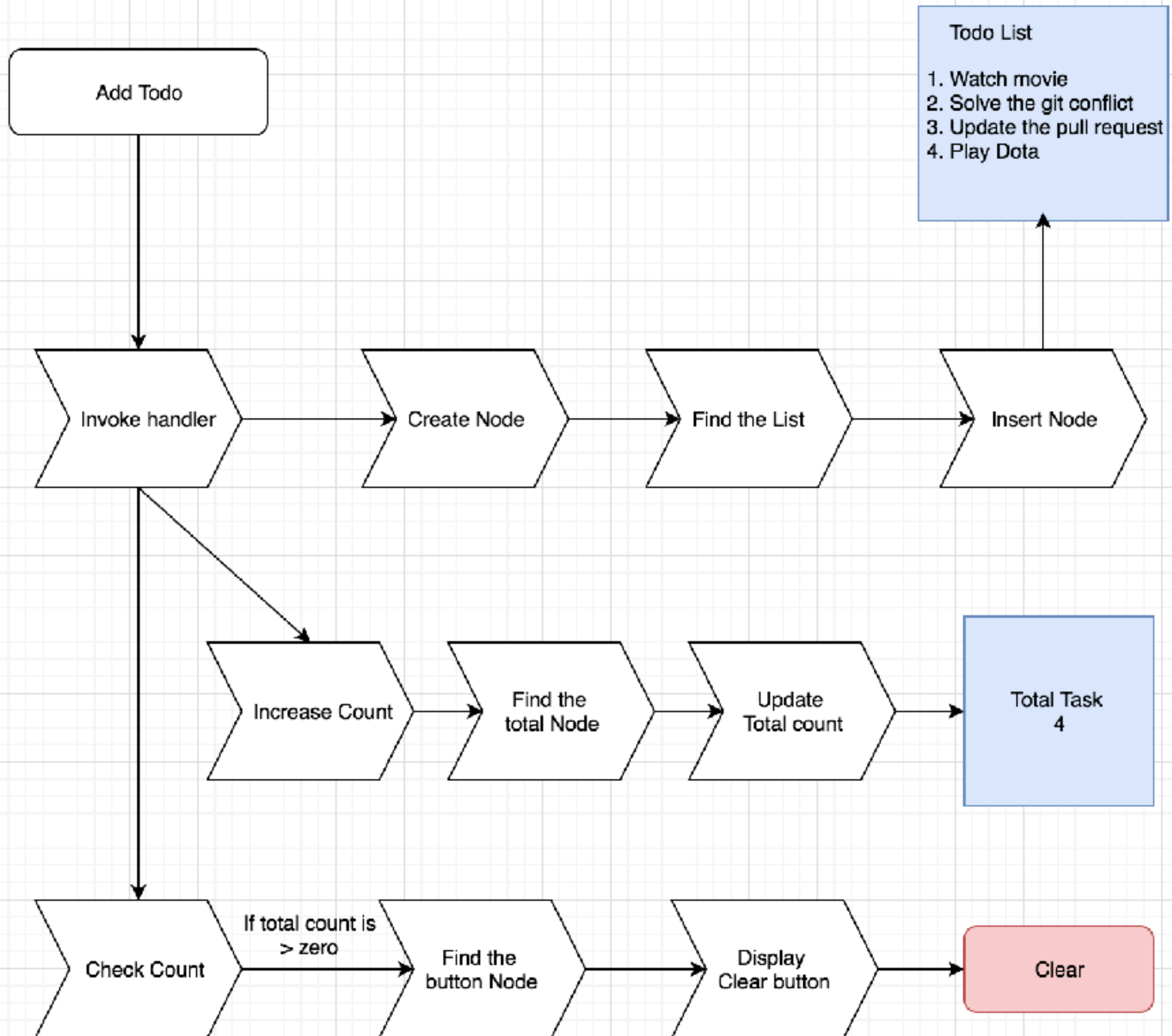
Topics

- Imperative vs declarative
- Vue is declarative
- Three types of way to share component state
- VUEX basic and core concept
- When to use local state in VUEX app
- VUEX features and advance pattern
- When single source of truth is not true anymore
- Q & A

Back to old days
imperative way

Imperative ways

“We instruct things “HOW” to do”

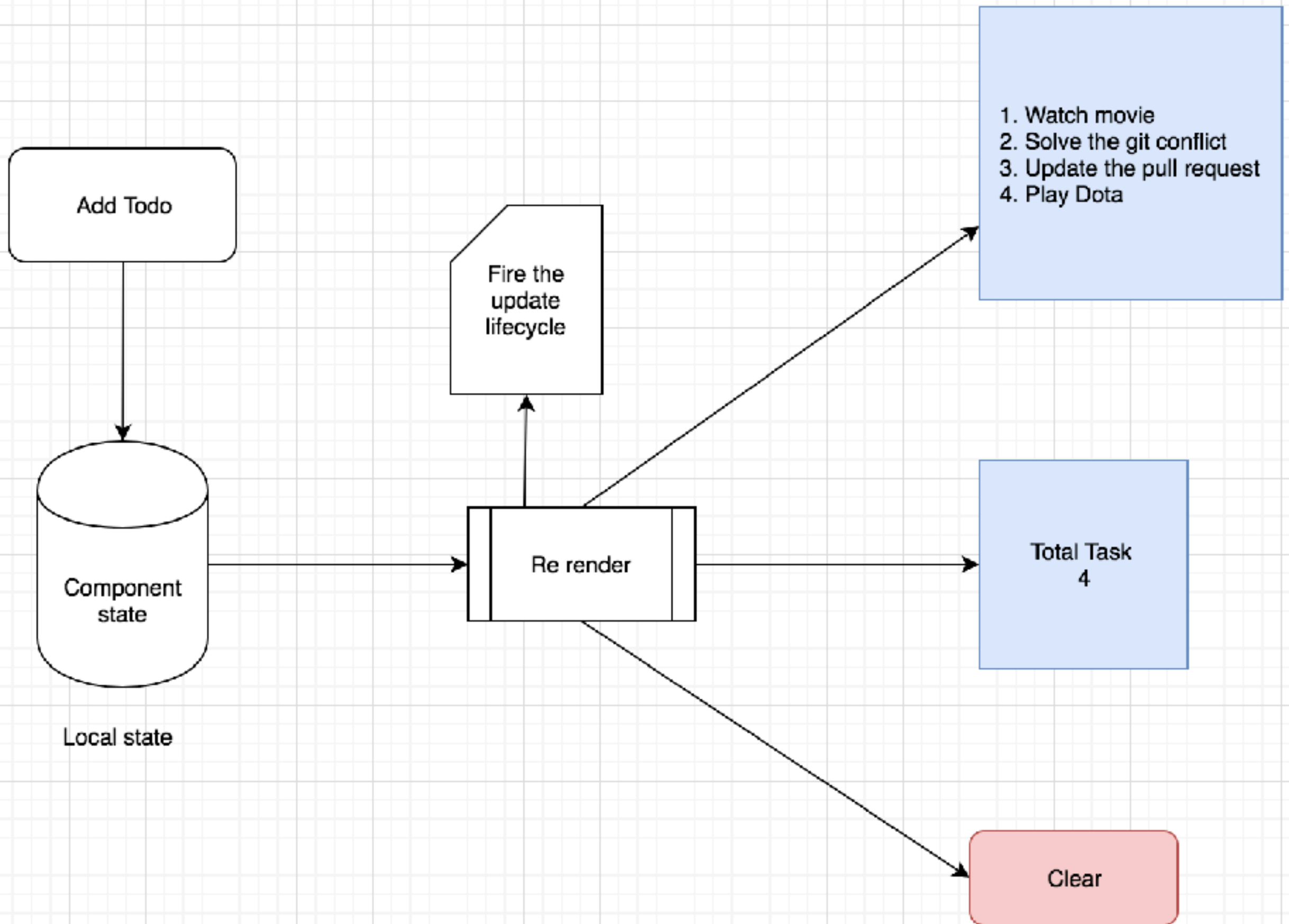


Declarative ways

“We instruct things “WHAT” to do”

```
<template>
  <div id="todo">
    <ul>
      <li v-for="item in items" :key="item.id">
        {{ item.message }}
      </li>
    </ul>
    <div>
      <input type="text" v-model="input" />
      <button v-on:click="submit">Submit</button>
    </div>
  </div>
</template>
```

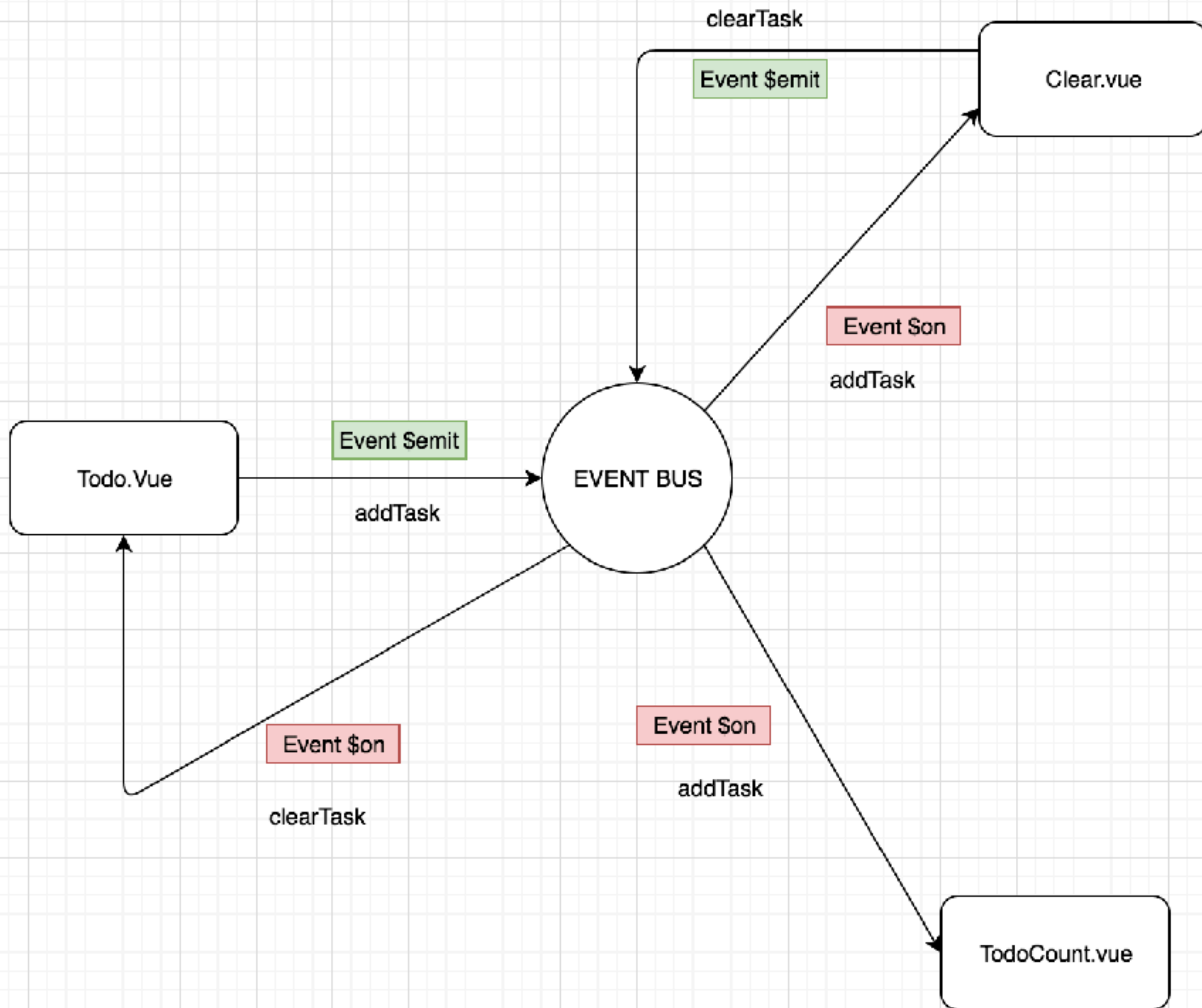
```
<script>
export default {
  name: "Todo",
  props: {
    msg: String
  },
  data: function() {
    return {
      input: '',
      items: [{ message: "so hai" }]
    };
  },
  methods: {
    submit: function () {
      this.items.push({ message: this.input })
      this.input = ""
    }
  }
};
</script>
```



Sharing State without VUEX

Event Bus

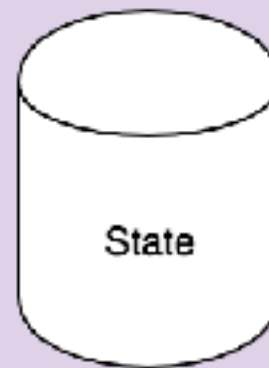
- Use event to share data by communicating between components
- More actions, you need to maintain more



Root state

- Store all the data in root components
- Pass those data as children
- To update data from children , define root method and pass props
- You have to pass all of props to children without needing it

App Root Component



State

All of the methods to update
state define here

App Root Component

pass
as
prop

Footer.vue

pass
as
prop

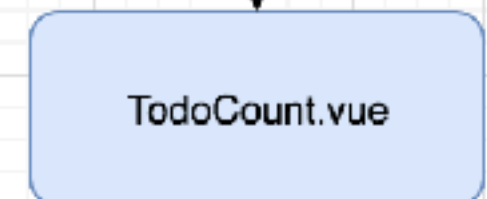
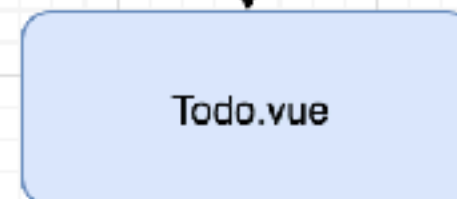
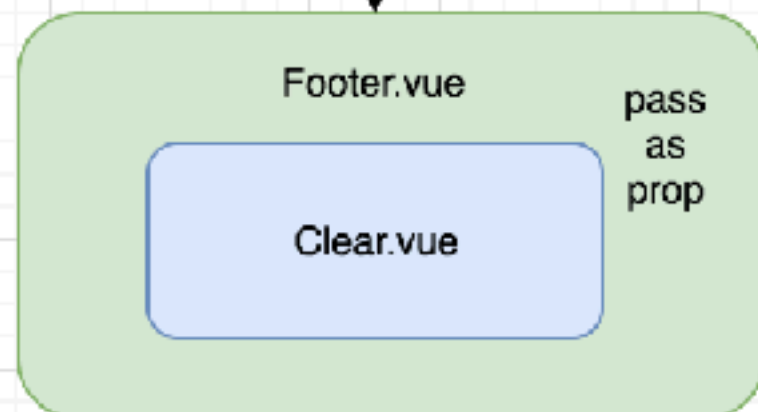
Clear.vue

pass
as
prop

Todo.vue

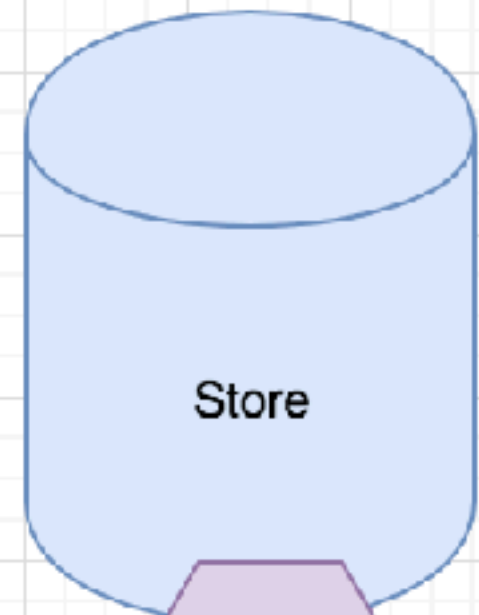
pass
as
prop

TodoCount.vue

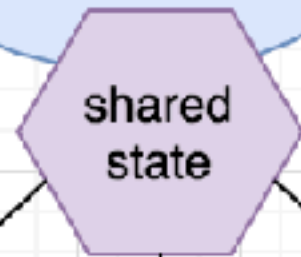


Manual Source of Truth

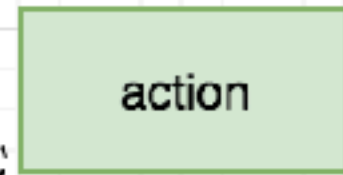
- Yeah now you can use single source of truth without VUEX



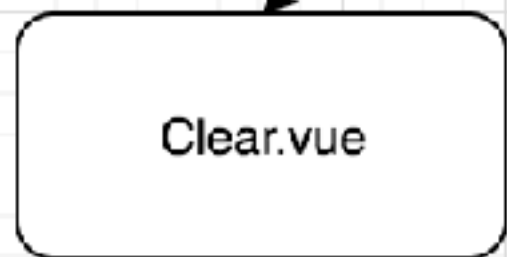
Store



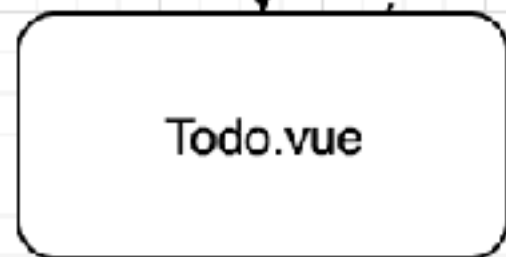
shared
state



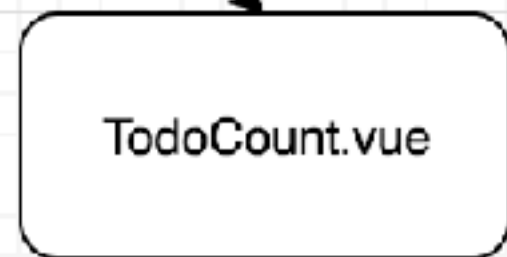
action



Clear.vue



Todo.vue



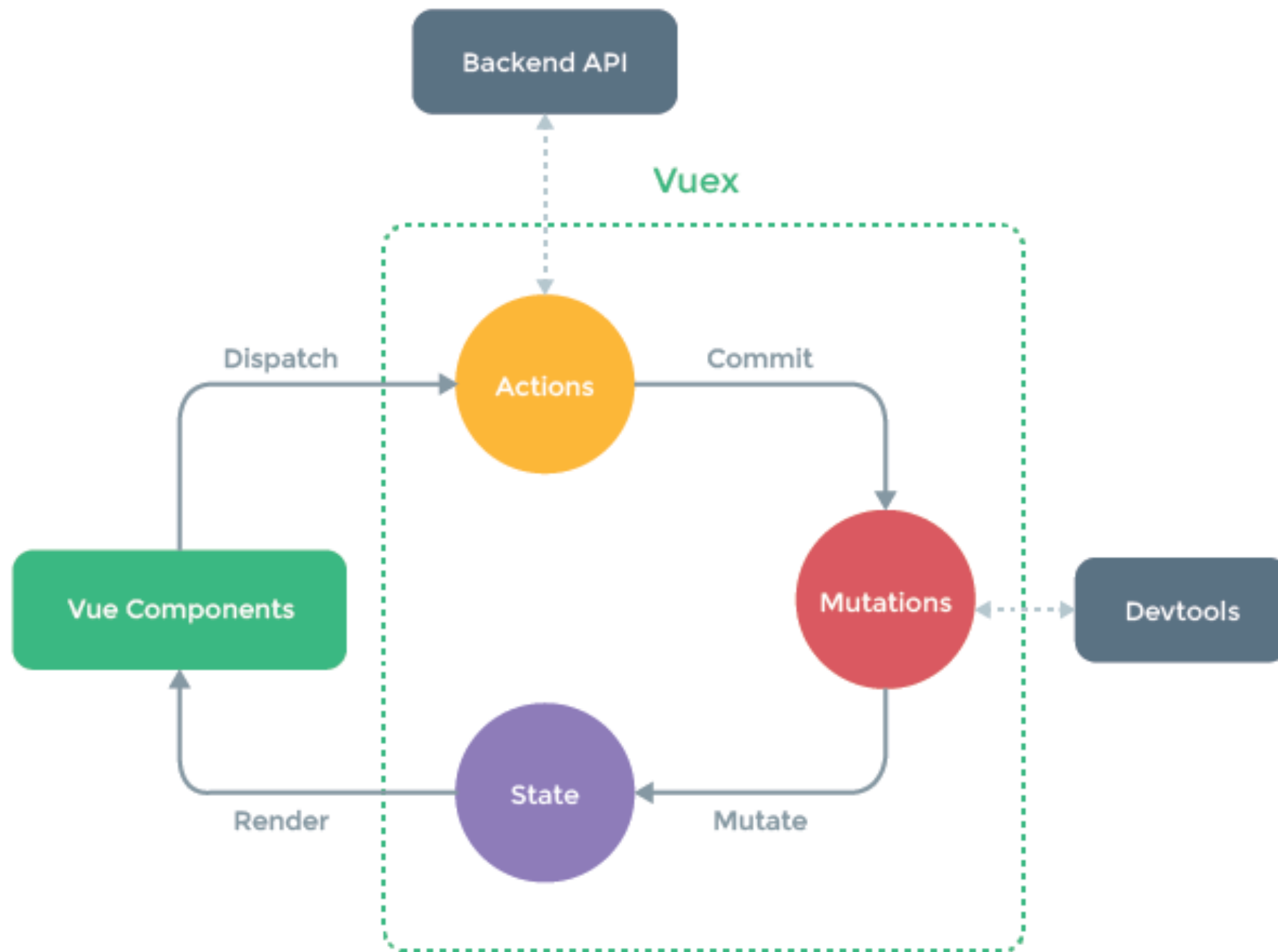
TodoCount.vue



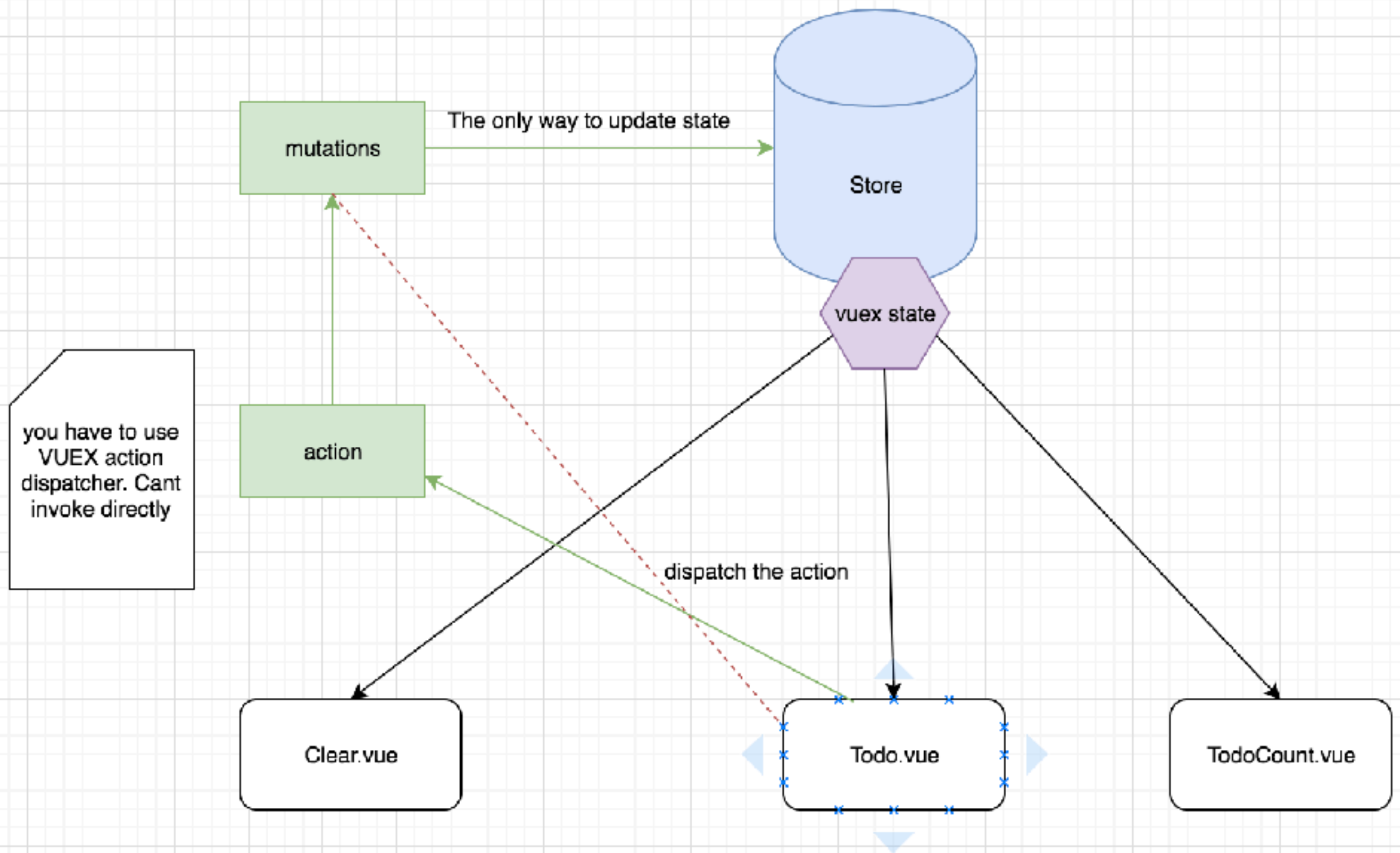
VUEX

CORE CONCEPT

- State
- Mutations
- Actions



VUEX



“You don’t have to use VUEX always.”

Local state is needed sometimes.

When to use local state
inside VUEX app ?

- Use in form elements
- When other components doesn't depend on the component' state , you are free to use local state
- Use in reusable UI components

Other Feature

- You may think that VUEX is only useful for state management
- It is useful in time traveller debugging
- You can snapshot current state send to server to reproduce the bug
- You can log all of the mutations event

Advance VUEX pattern

- mapState
- getter
- mapMutations
- mapActions
- Last but not least , Module pattern


```
|— index.html
|— main.js
|— api
|   └─ ... # abstractions for making API requests
|— components
|   └─ App.vue
|   └─ ...
└─ store
    └─ index.js          # where we assemble modules and export the store
    └─ actions.js       # root actions
    └─ mutations.js     # root mutations
    └─ modules
        └─ cart.js      # cart module
        └─ products.js  # products module
```

When single source of
truth is not true anymore

- The main principle of state management is source must be truth
- If there is duplicated entity, the source is not true anymore

```
const store = {
  topItems : [
    {
      id: 1,
      title: "IphoneX",
      price: 1200000
    },
    {
      id: 2,
      title: "Samsung",
      price: 1000000
    }
  ],
  myProducts : [
    {
      id: 1,
      title: "IphoneX",
      price: 1200000
    },
    {
      id: 2,
      title: "Samsung",
      price: 1000000
    },
    {
      id: 3,
      title: "Xiaomi",
      price: 500000
    }
  ],
  TrendingItems: [
    {
      id: 3,
      title: "Xiaomi",
      price: 500000
    },
    {
      id: 4,
      title: "Xiaomi",
      price: 500000
    }
  ]
},
```

```
const store = {
  topItems : [
    {
      id: 1,
    },
    {
      id: 2,
    }
  ],
  myProducts : [
    {
      id: 1,
    },
    {
      id: 2,
    },
    {
      id: 3,
    }
  ],
  TrendingItems: [
    {
      id: 3,
    },
    {
      id: 4,
    }
  ],
  products: [
    {
      id: 1,
      title: "IphoneX",
      price: 1200000
    },
    {
      id: 2,
      title: "Samsung",
      price: 1000000
    },
  ],
}
```

- Use normalizr library at Github for complex JSON to define entity
- <https://redux.js.org/recipes/structuring-reducers/normalizing-state-shape>

“VUEX is like a glass, you will know when you need it.”

Thanks You
Q & A