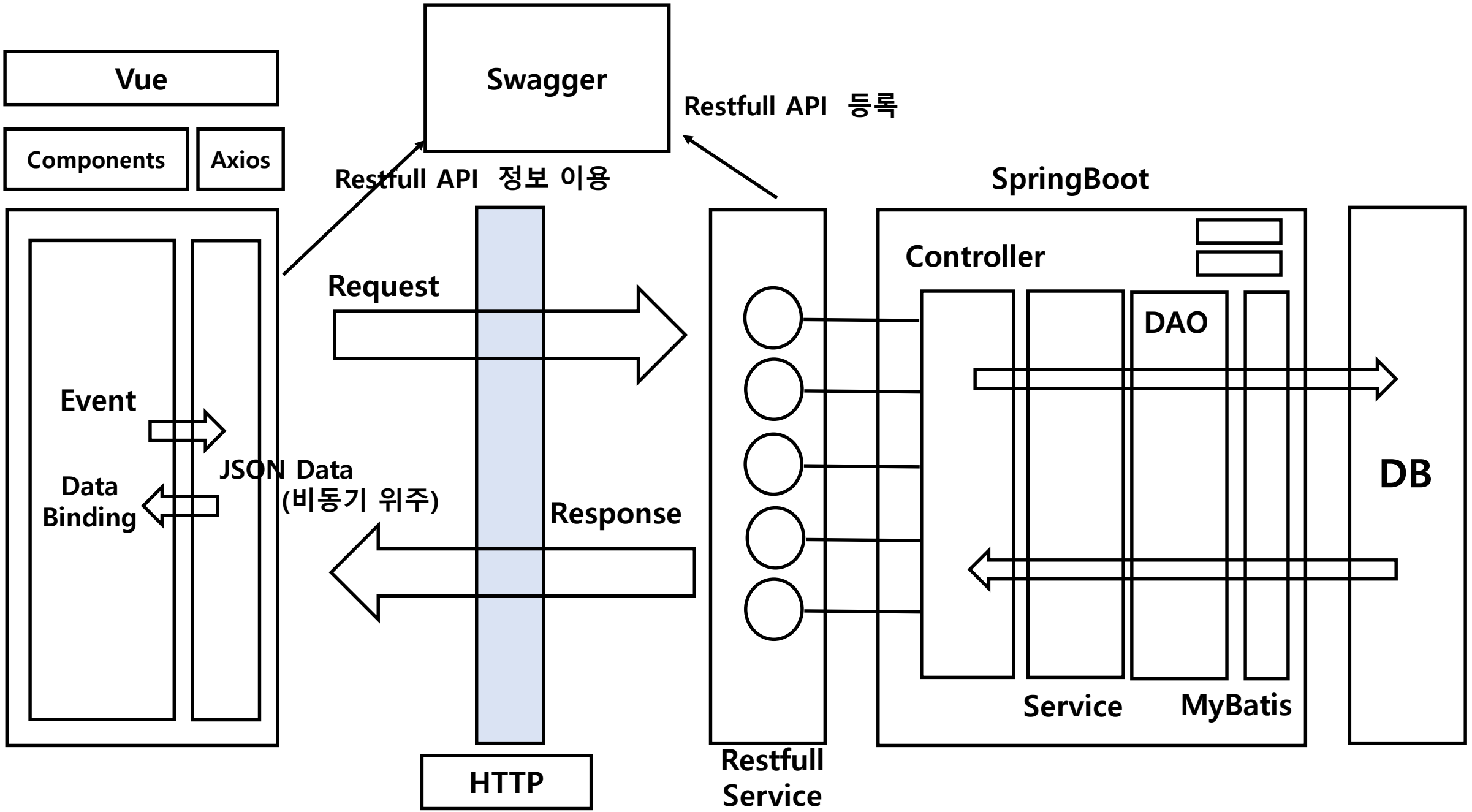

Vue
Components
Axios
Swagger
Restfull API 등록
Restfull API 정보 이용
SpringBoot
Request
Event
Data Binding
JSON Data
(비동기 위주)
Response
HTTP
Restfull Service
Controller
DAO
Service
MyBatis
DB

Single File Component (SFC)

<template>

</ template>

<script>
export deault{

}
</script>

<style scoped>

</style>

```
<template><div><h1> {{ msg }} </h1><div</template>
```

```
<script>
export default {
 components: {},     /* 자식 컴포넌트 등록 */
 props: [],          /* 부모와 자식간 데이터 통신용 프로퍼티 */
 data() {            /* 컴포넌트 옵션  *
  return {
   msg: '한 컴포넌트의 데이터'
  }
 },
 watch: {},          /* 비동기나 지속성 데이터용 */
 computed: {},       /* {{ }}나 bind에서 사용할 속성  */
 methods: {
 },
                     /* 라이프 싸이클 관려 */
 created() {},  mounted() {},  updated() {},  destroyed() {}
}
</script>
```

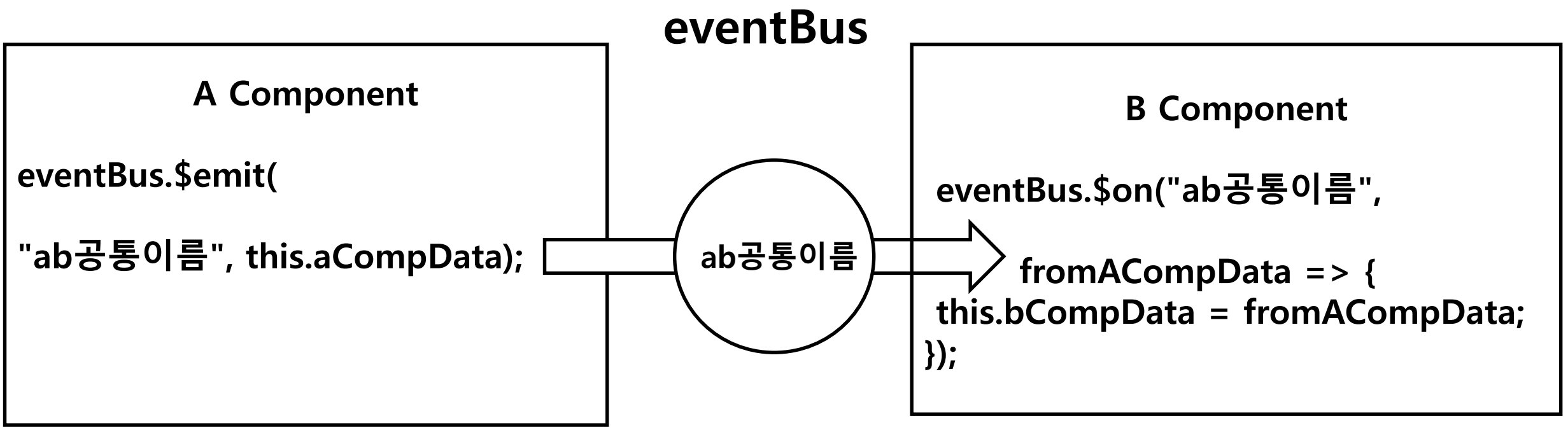
<style scoped> </style>

Sibling Component간 데이터 통신

comp/eventbus.js

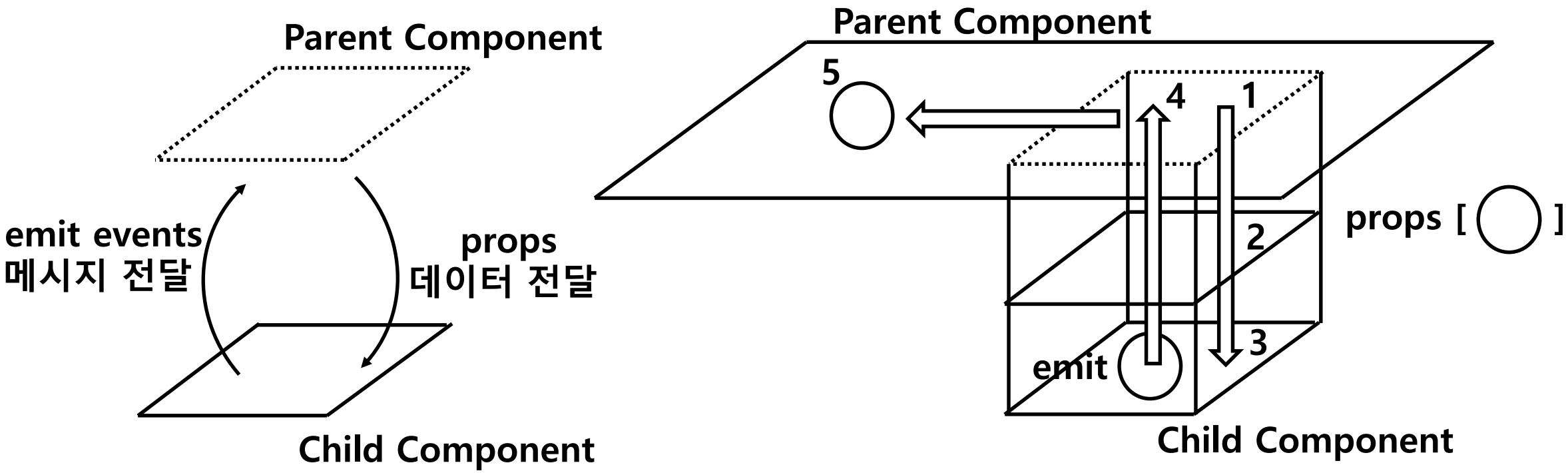
```
export const eventBus = new Vue();
```

```
import { eventBus } from "~/comp/eventbus";
```

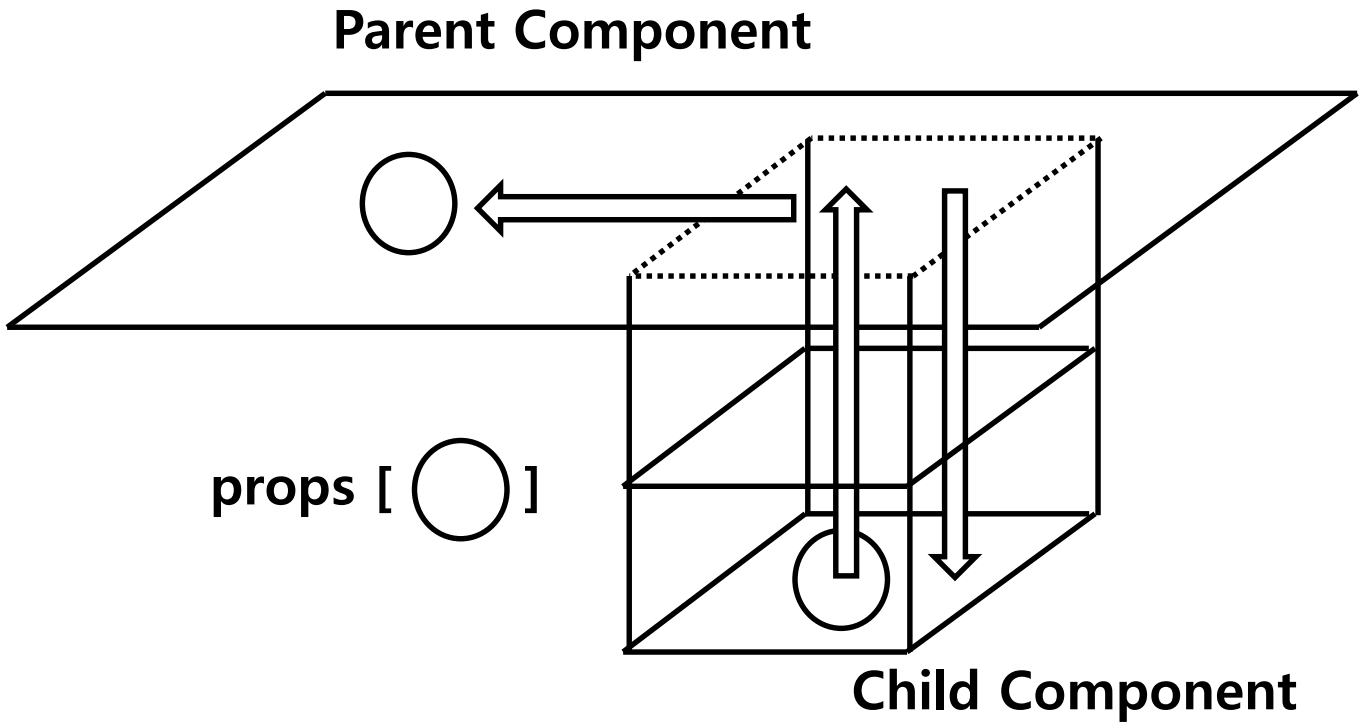


부모 컴포넌트는 props를 통해 자식에게 데이터를 전달,

자식 컴포넌트는 events를 통해 부모에게 메시지를 보냅니다



형제간 데이터 전송에도 사용할 수 있습니다.

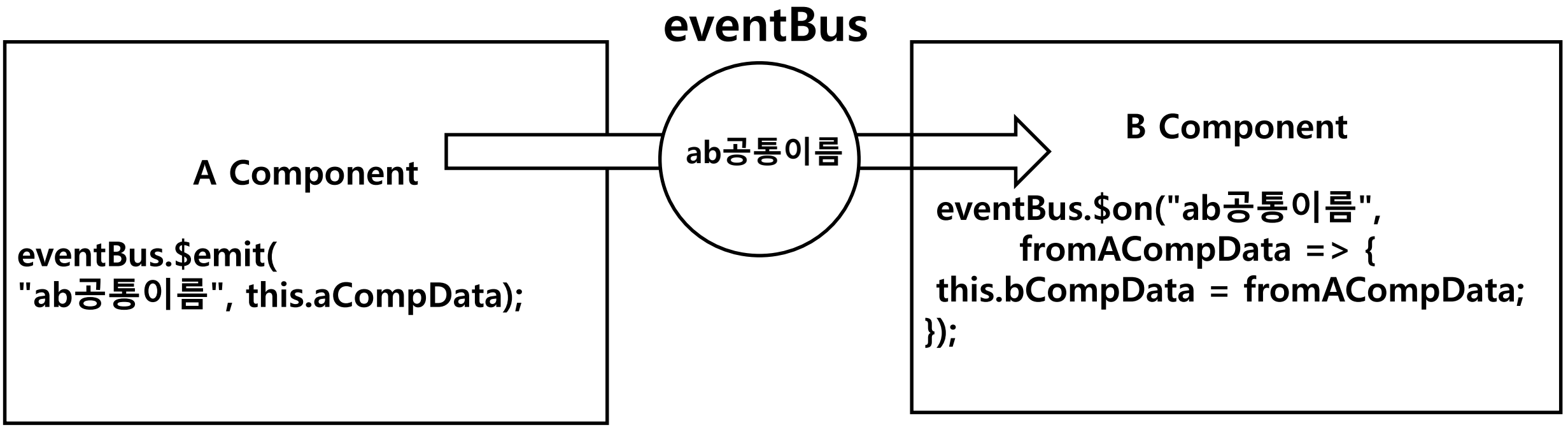

Parent Component
props [ ]
Child Component

Sibling Component간 데이터 통신

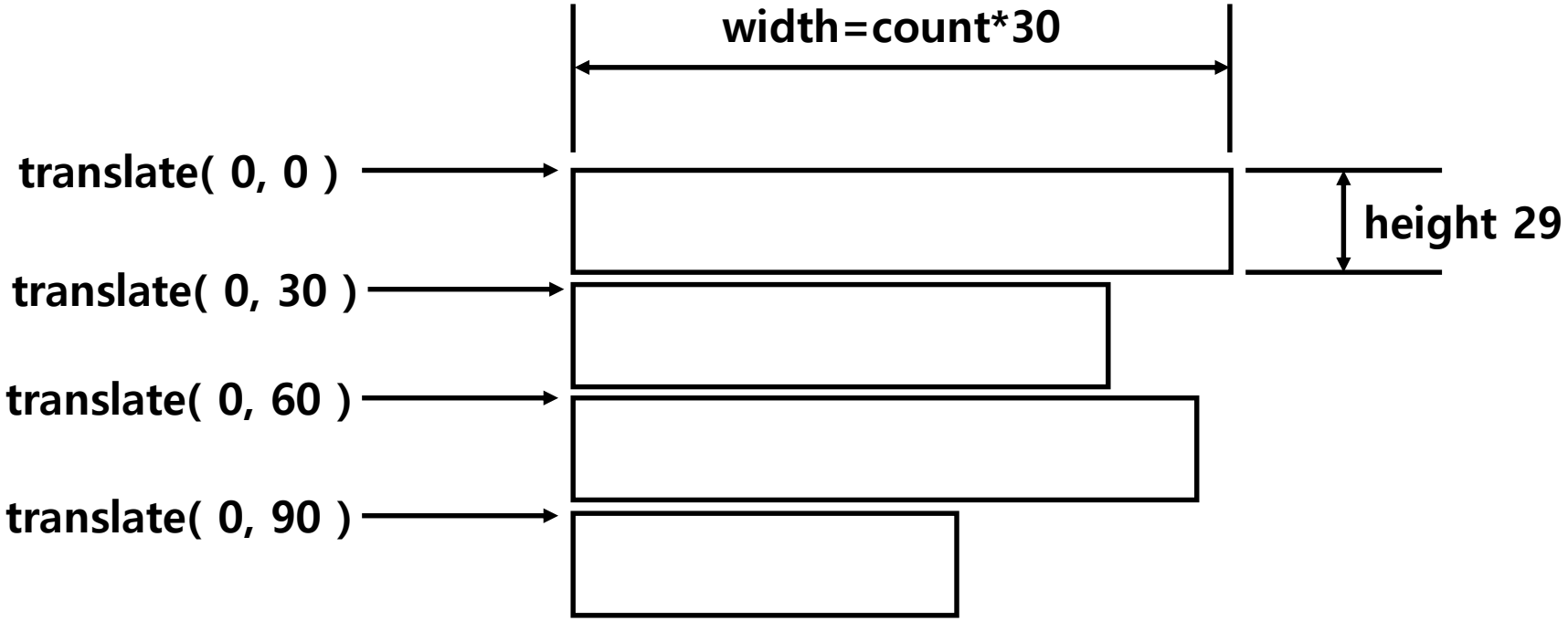
comp/eventbus.js

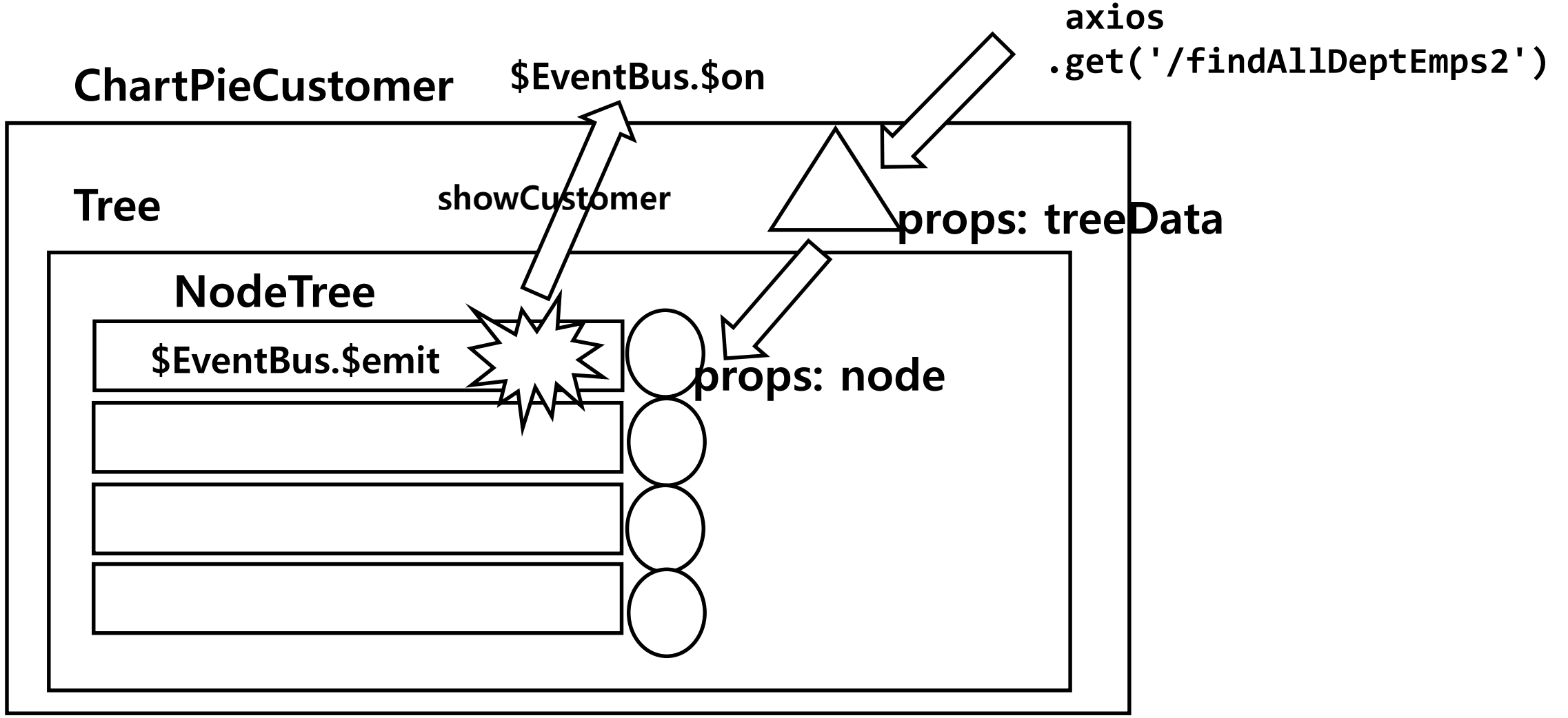
```
export const eventBus = new Vue();
```

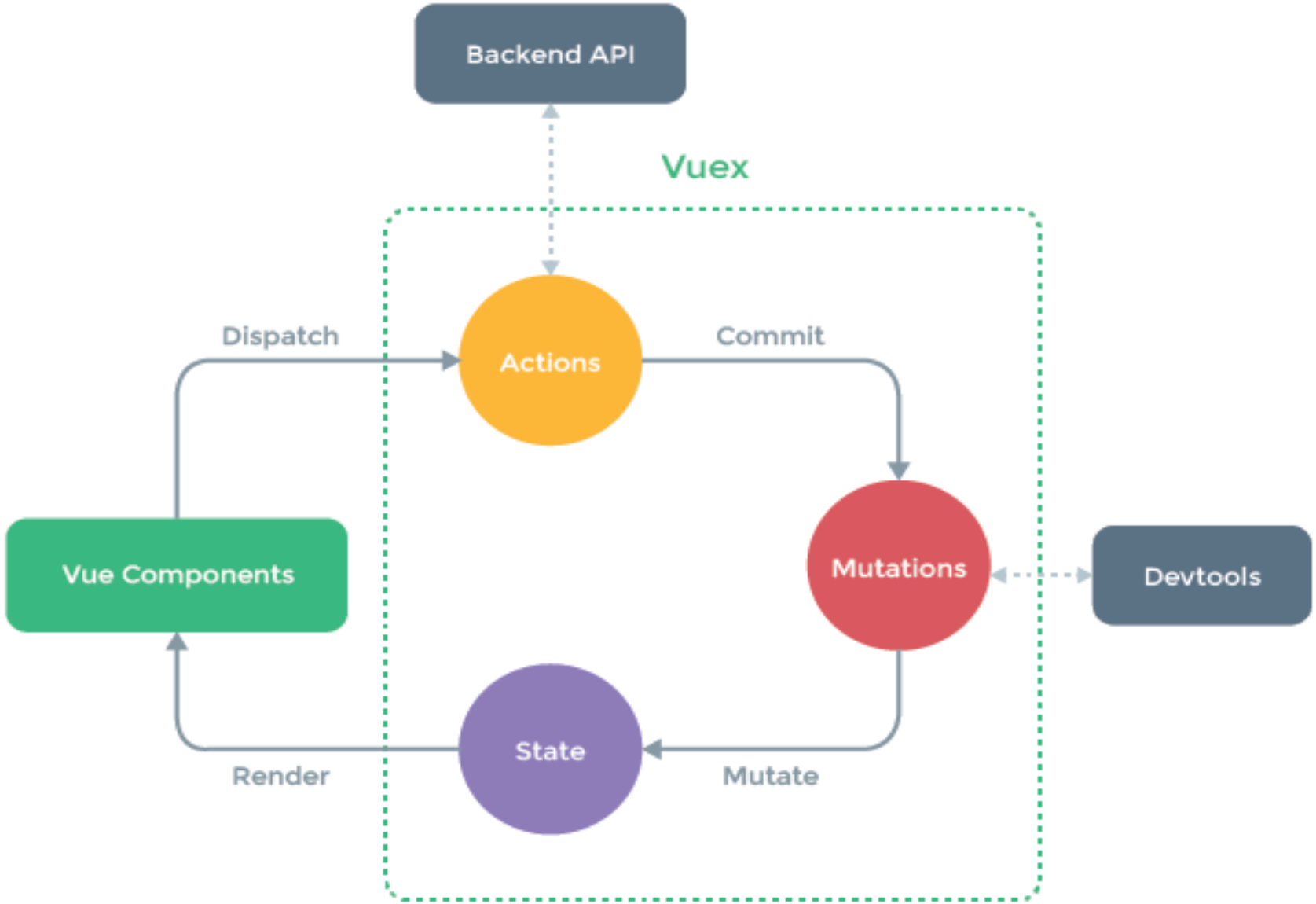
```
import { eventBus } from "~/comp/eventbus";
```



```
<script>
export default {
 components: {},     /* 자식 컴포넌트 등록 */
 props: [],          /* 부모와 자식간 데이터 통신용 프로퍼티 */
 data() {            /* 컴포넌트 옵션  *
  return {
   msg: '한 컴포넌트의 데이터'
  }
 },
 watch: {},          /* 비동기나 지속성 데이터용 */
 computed: {},       /* {{ }}나 bind에서 사용할 속성  */
 methods: {
 },
                     /* 라이프 싸이클 관려 */
 created() {},  mounted() {},  updated() {},  destroyed() {}
}
</script>
```


width=count*30
translate(0, 0)
translate(0, 30)
translate(0, 60)
translate(0, 90)
height 29

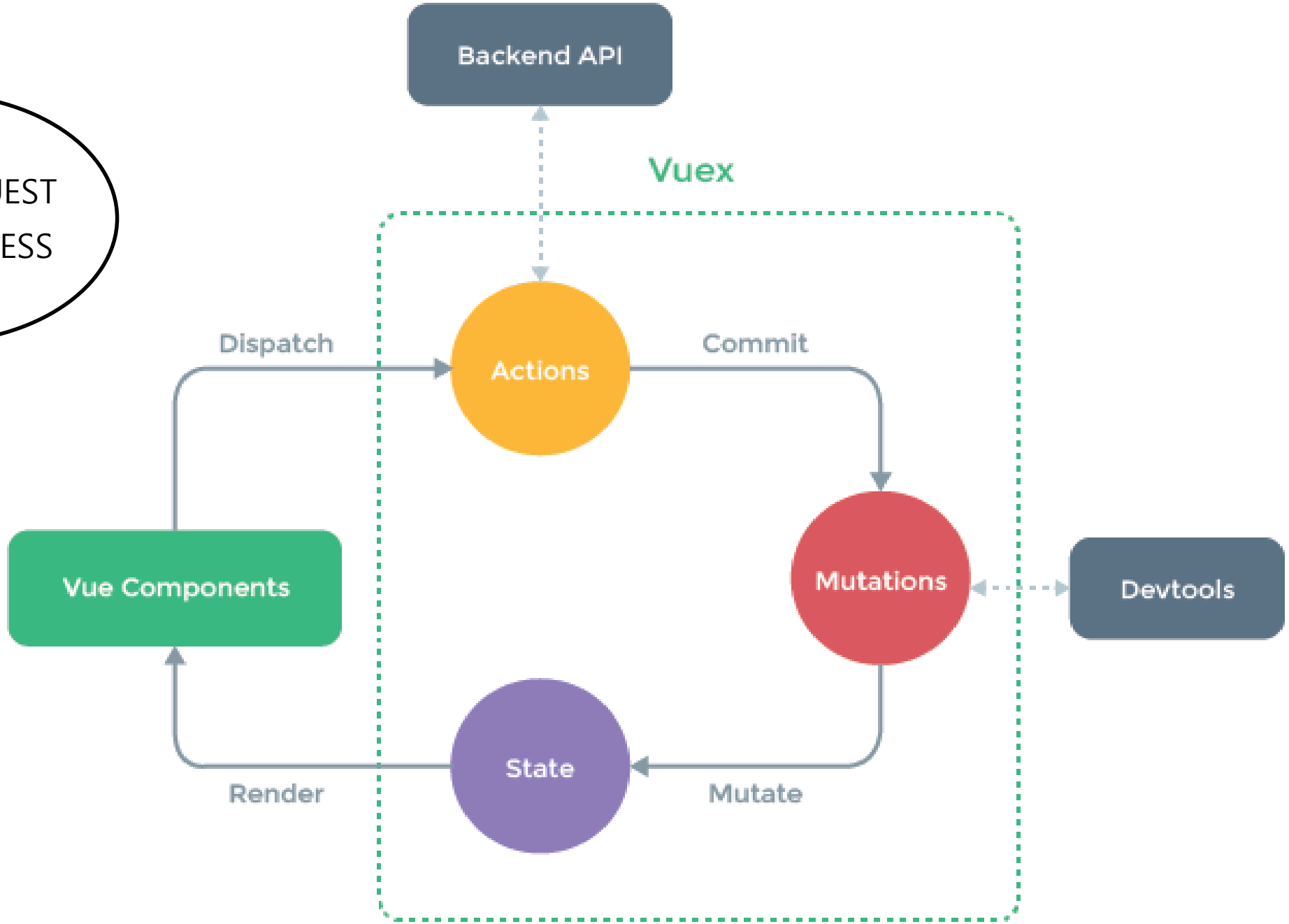

axios
.get('/findAllDeptEmps2')
ChartPieCustomer
$EventBus.$on
Tree
showCustomer
props: treeData
NodeTree
$EventBus.$emit
props: node

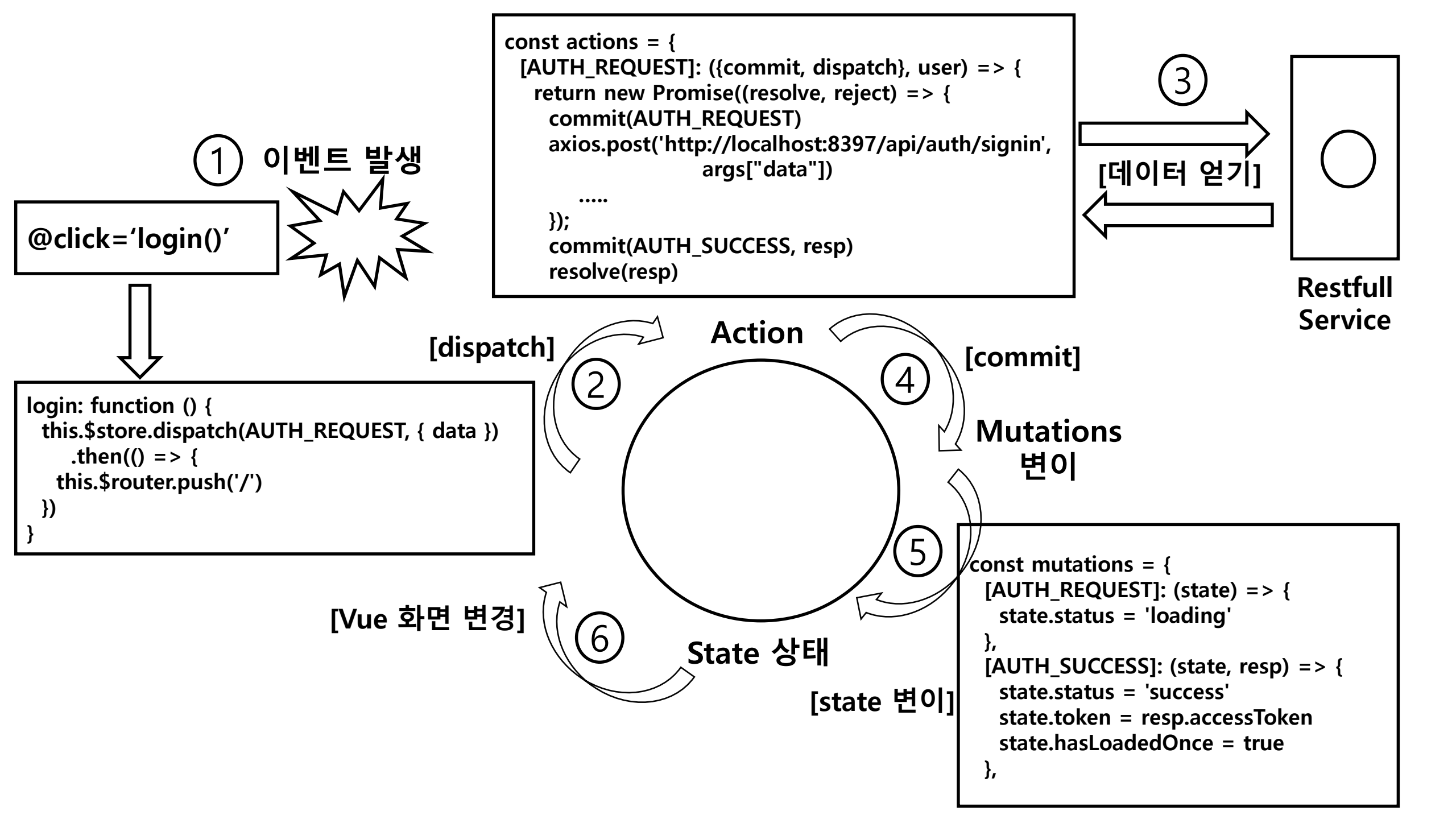

Backend API
Vuex
Dispatch
Actions
Commit
Vue Components
Mutations
Devtools
State
Render
Mutate

Store

AUTH_REQUEST

AUTH_SUCCESS




const actions = {
[AUTH_REQUEST]: ({commit, dispatch}, user) => {
return new Promise((resolve, reject) => {
commit(AUTH_REQUEST)
axios.post('http://localhost:8397/api/auth/signin',
args["data"])
.....
});
commit(AUTH_SUCCESS, resp)
resolve(resp)
③
[데이터 얻기]
Restfull Service
① 이벤트 발생
@click='login()'
login: function () {
this.$store.dispatch(AUTH_REQUEST, { data })
.then(() => {
this.$router.push('/')
})
}
[dispatch]
Action
②
④
[commit]
Mutations
변이
const mutations = {
[AUTH_REQUEST]: (state) => {
state.status = 'loading'
},
[AUTH_SUCCESS]: (state, resp) => {
state.status = 'success'
state.token = resp.accessToken
state.hasLoadedOnce = true
},
⑤
[state 변이]
State 상태
⑥
[Vue 화면 변경]