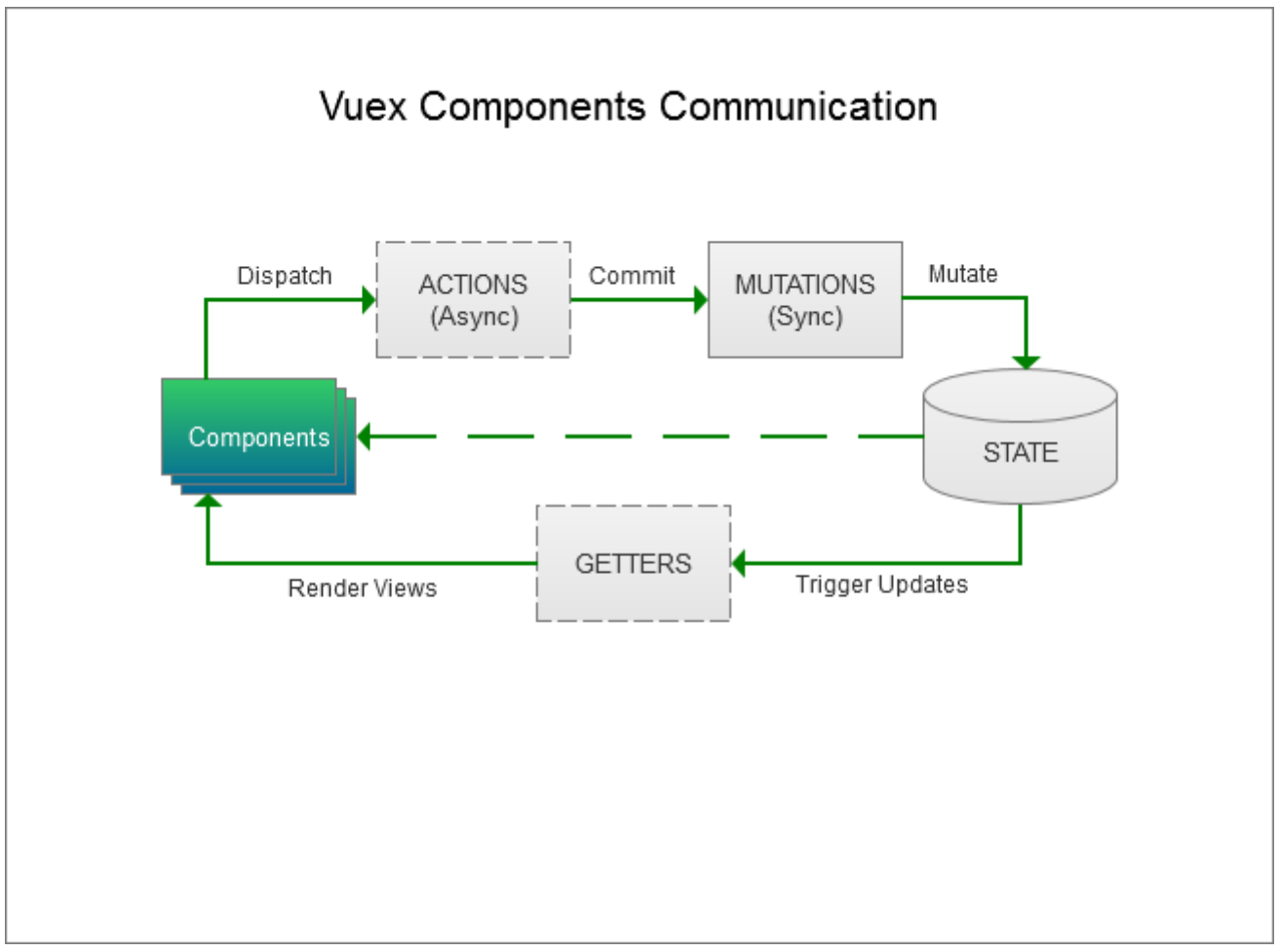
Vuex

Status	Not started
Assign	Vue.js

#Vuex, Vue프레임 워크의 상태관리



Store.js 스토어 셋팅

```
import Vue from "vue";
import Vuex from "vuex";

Vue.use(Vuex);

export default new Vuex.Store({
  state: {
    username: ""
  },
  mutations: {
    GET_USER_NAME(state, username) {
      state.username = username
    }
  },
  actions: {
    GET_USER_NAME({commit, {username}}) {
        return store.commit('GET_USER_NAME', username)
    }
  },
  getters: {
    getUserName: (state) => state.username
  },
})
```

State

- data라고 생각하자

- 중앙에서 관리하는 모든 상태 정보이다.
- react의 store라고 볼 수 있다.
- 각 컴포넌트는 Vuex에서 state를 가져와 사용
- Mutations에 의해 변경됨
- State가 변화하면 해당 state를 공유하는 여러 컴포넌트의 DOM은 알아서 렌더링

```
new Vue({
  created: function() {
    console.log(this.$store.state.username)
  }
})
```

Mutations

- State의 변경(생성, 수정, 삭제)을 담당
- 반드시 동기적 로직만
- Action의 commit(_)에 의해 호출
- 첫 번째 인자로 항상 state를 받음

Actions

- 비동기 로직 처리 가능
- 컴포넌트에서 dispatch(_)에 의해 호출
  - State변경과 관련된 일이라면 Mutations를 commit(_)으로 호출
  - Actions도 state를 조작할 수 없는 건 아니지만,
- 첫 번째 인자로 context 객체를 받음
- state변경을 제외한 다양한 일을 해야하기 때문에 context객체로 할 수 있는 일이 많다.

```
new Vue({
  methods: {
    getUserName() {
      const username = {
        username: '세경'
      }
      this.$store.dispatch('GET_USER_NAME', username)
    }
  }
})
```

Getters

- State의 상태를 기반하는 계산 값을 반환한다.

```
import { mapGetters } from 'vuex'
new Vue({
  computed: {
    ...mapGetters(['getUserName']),
  }
})
```