

# Redux



Redux

# Redux

# Redux

- A predictable state container pattern

# Redux

- A predictable state container pattern
- First implemented as a JavaScript library, frequently used with React

# Redux

- A predictable state container pattern
- First implemented as a JavaScript library, frequently used with React
- Based on Facebook's Flux architecture

**Store**

**Reducers**

**Actions**

## **Store**

A single source of truth through which to  
read and interact with the current state

**Reducers**

**Actions**

**Store**

**Reducers**

**Actions**

**Reducers**

**Store**

**Actions**

Information representing how we  
want to change the store's state

**Reducers**

**Store**

**Actions**

```
graph TD; Store[Store] --> Reducers[Reducers]; Actions[Actions] --> Reducers
```

**Reducers**

**Actions**

**Store**

Pure functions that take the previous state  
and an action, and return the next state

**Store**

**Reducers**

**Actions**

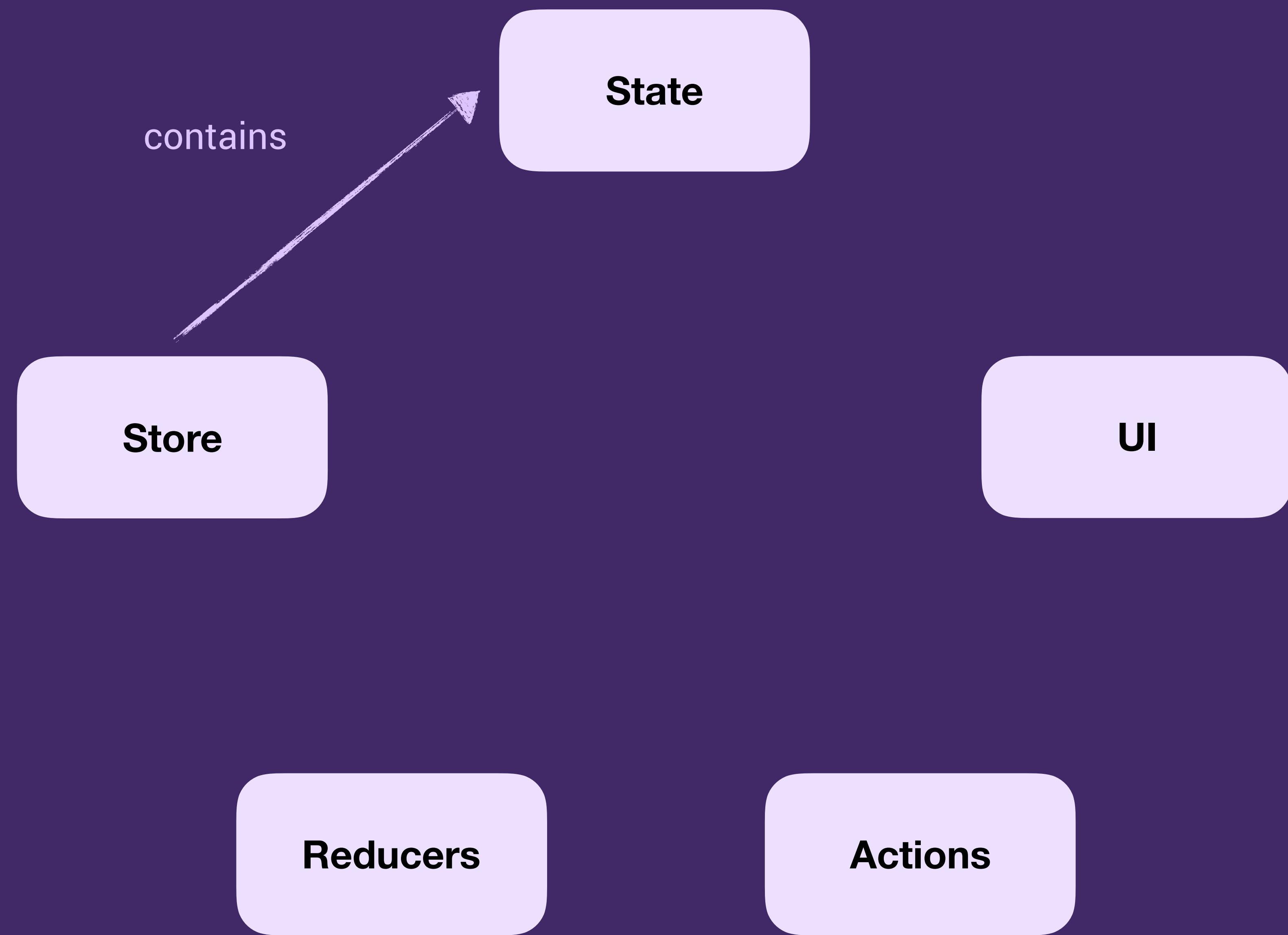
**State**

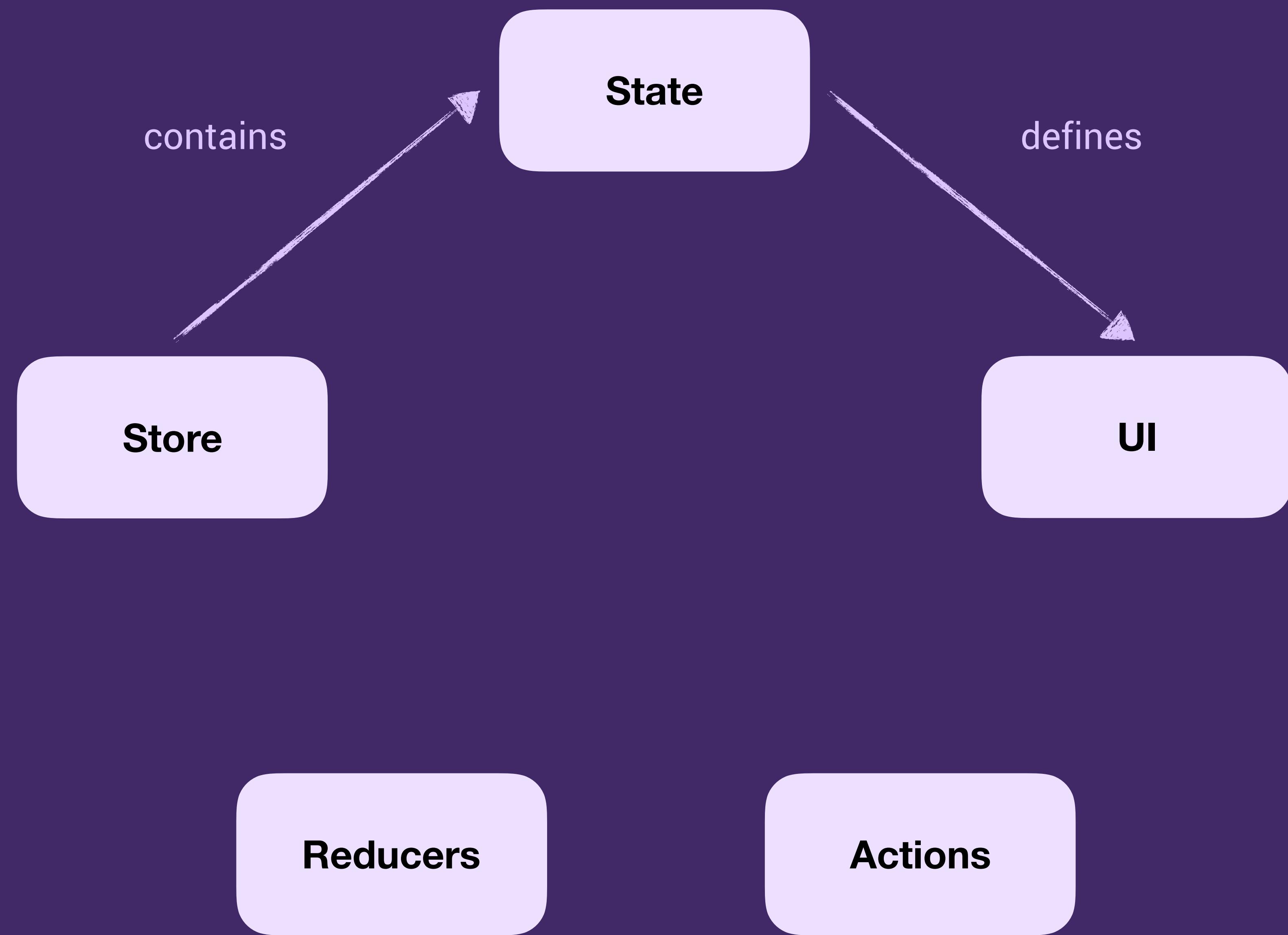
**Store**

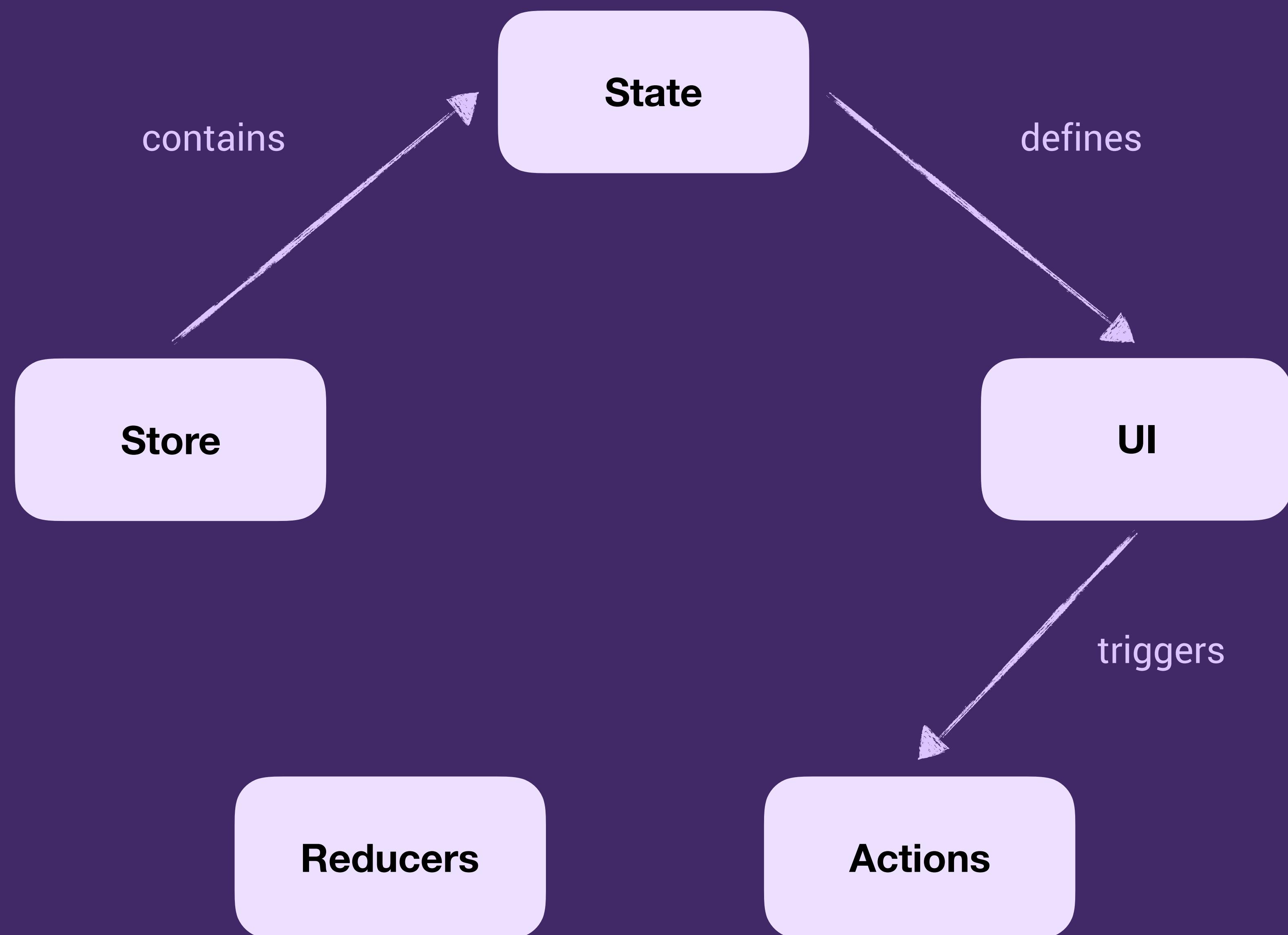
**UI**

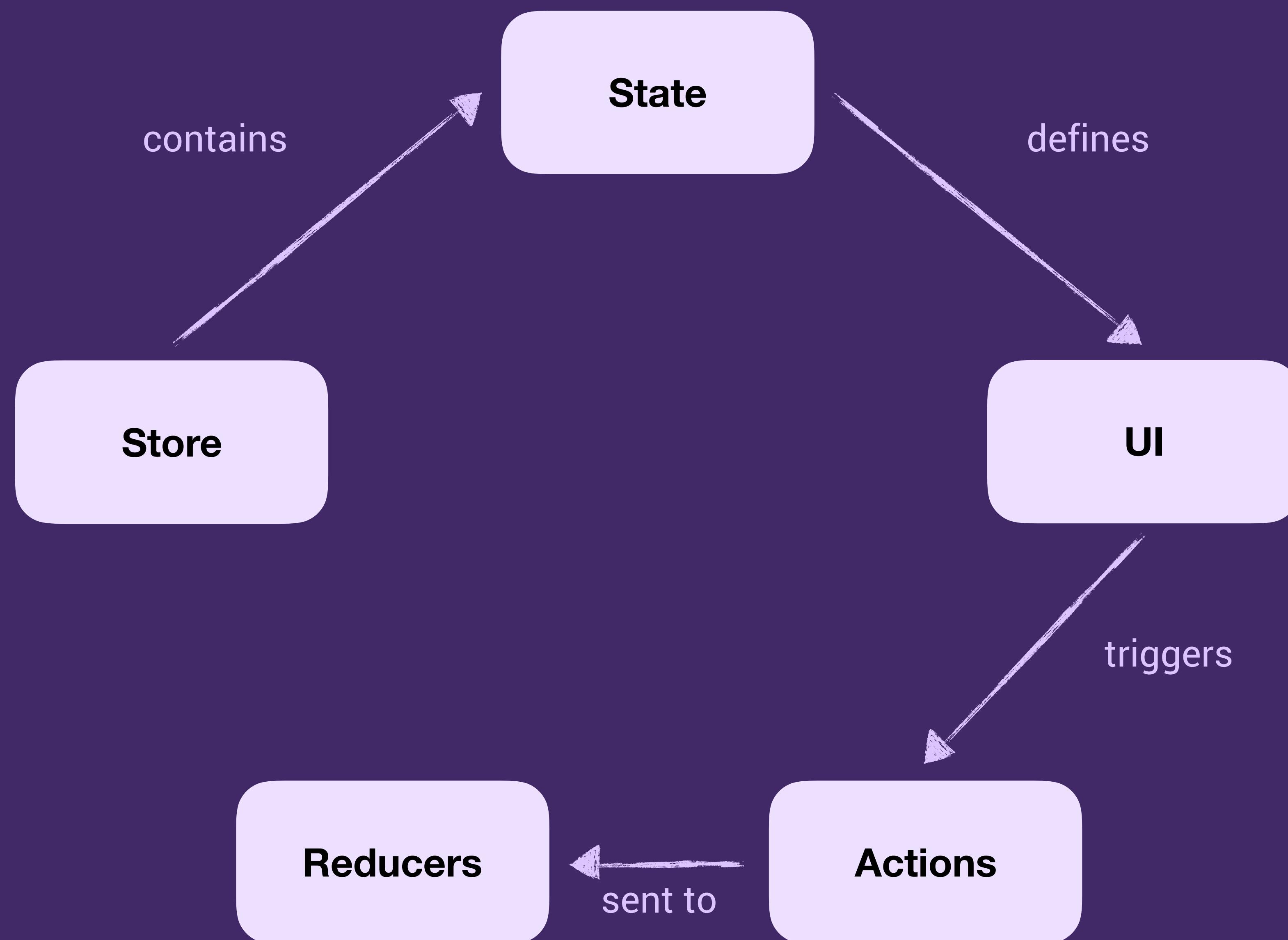
**Reducers**

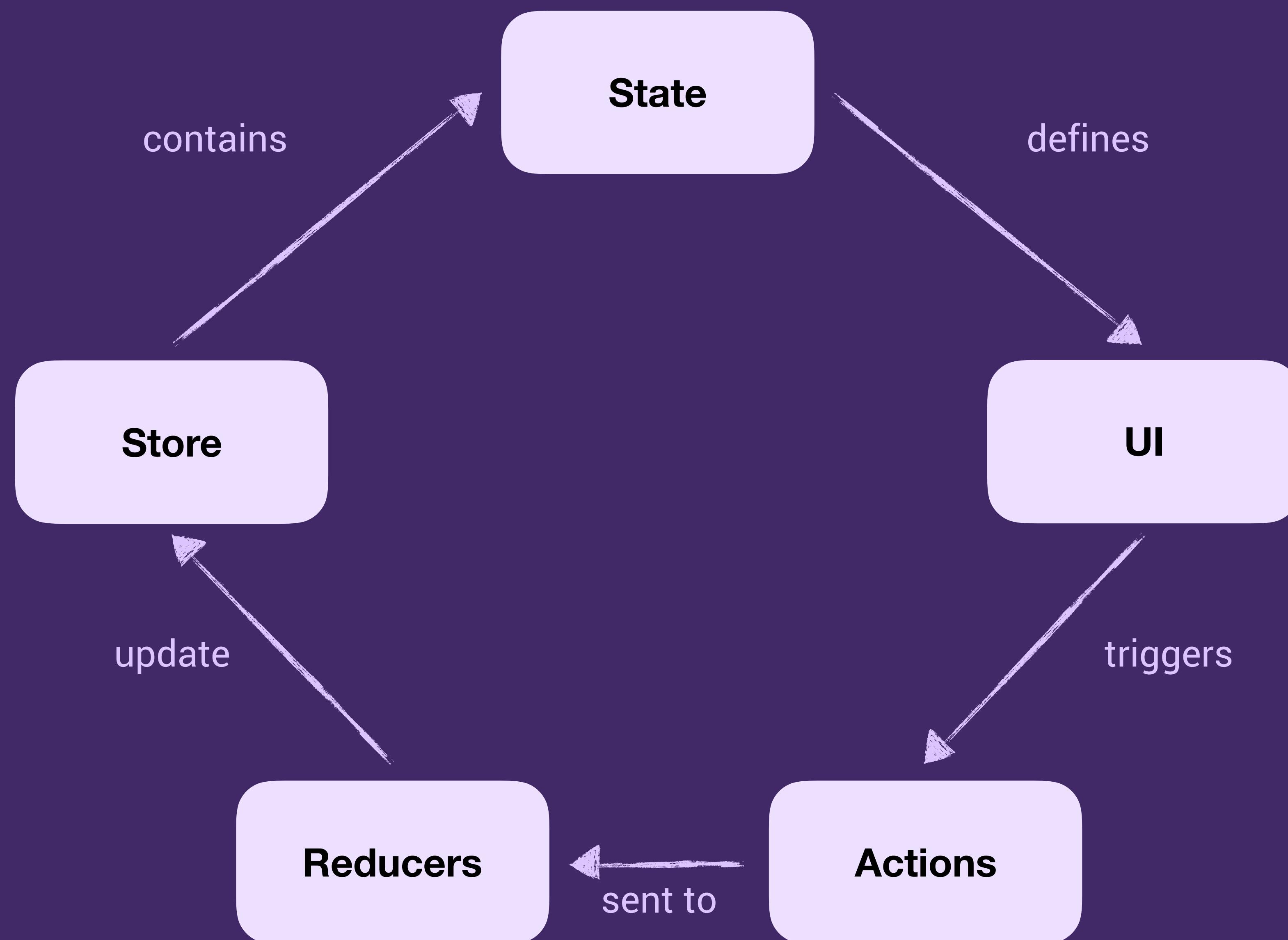
**Actions**



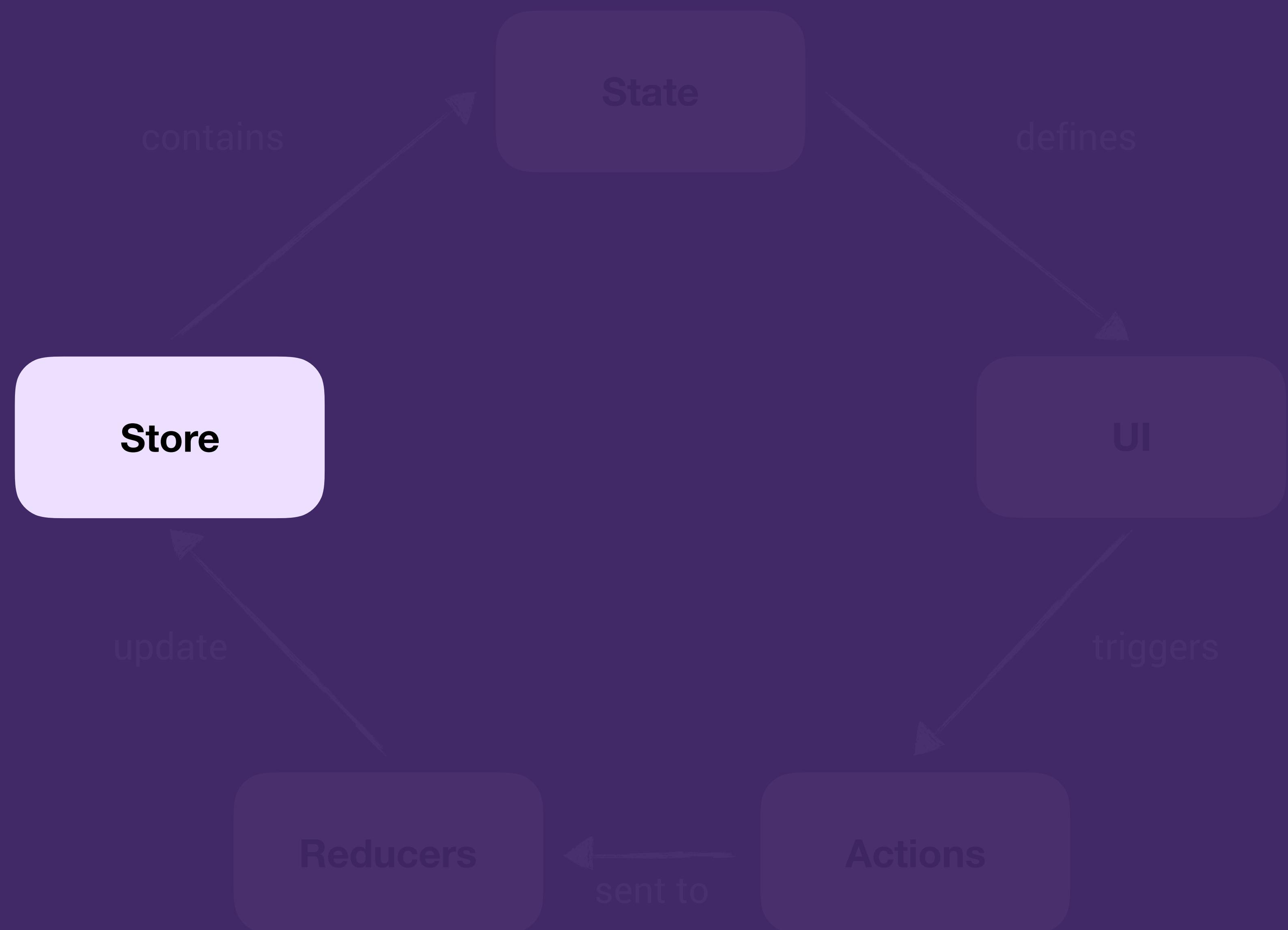


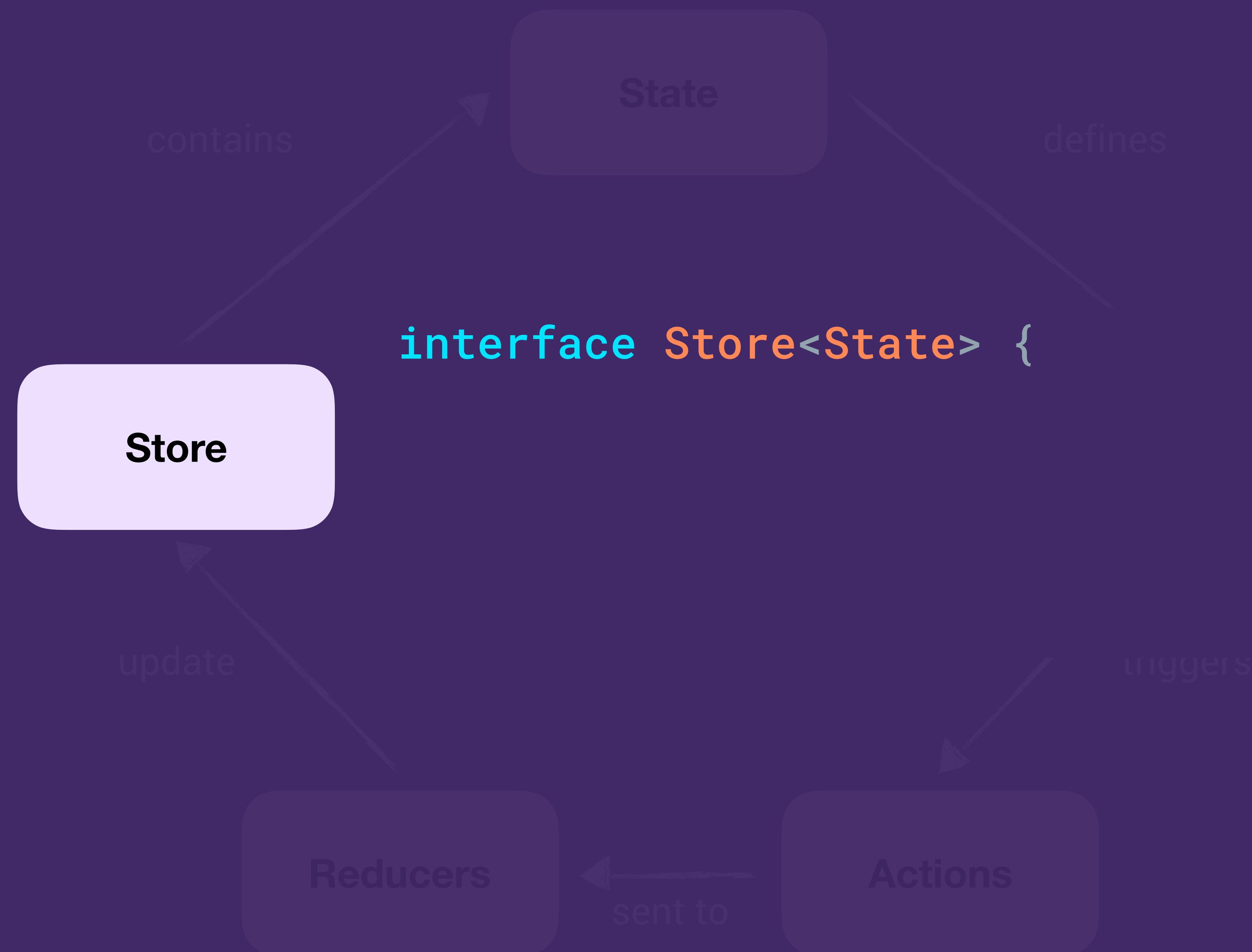


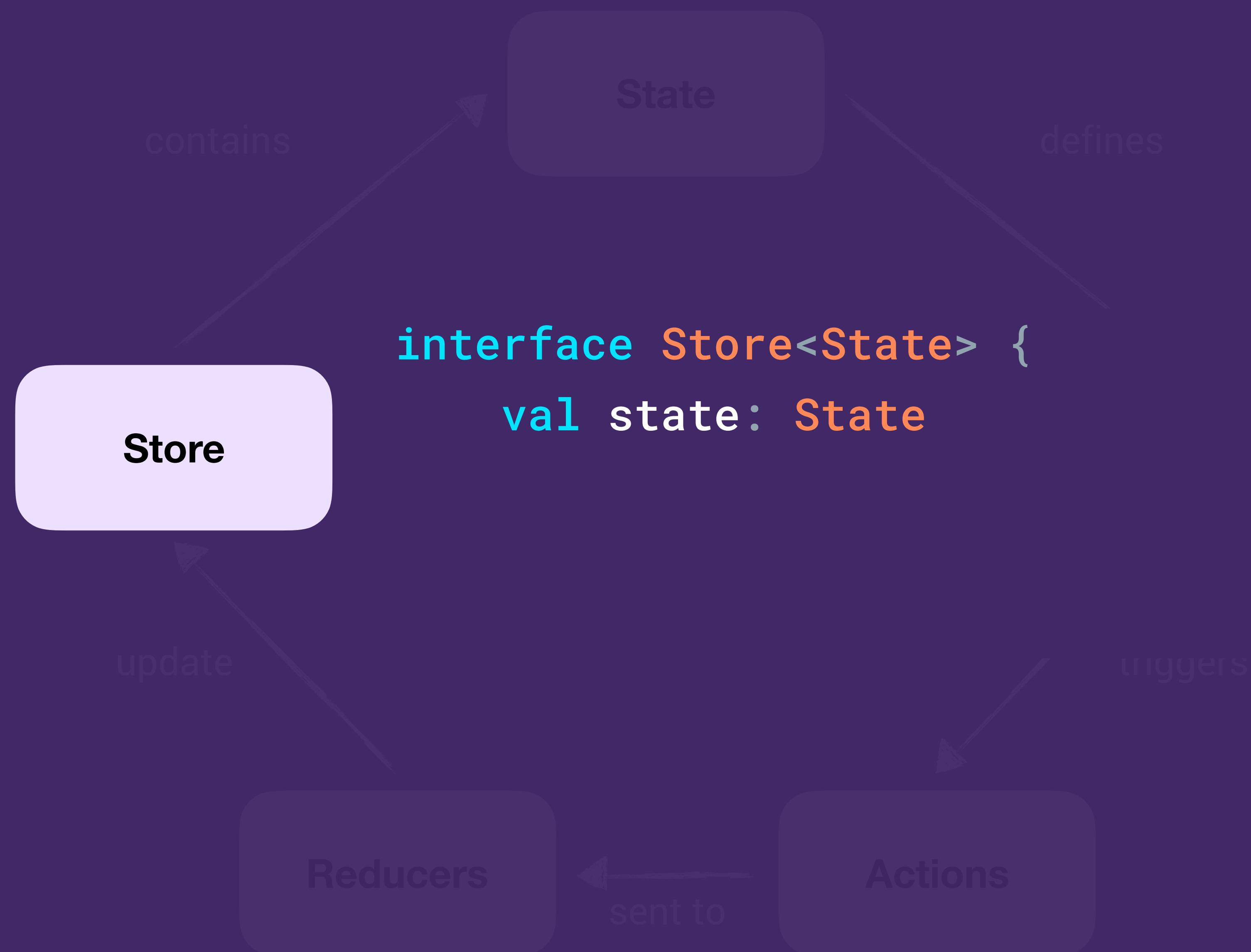


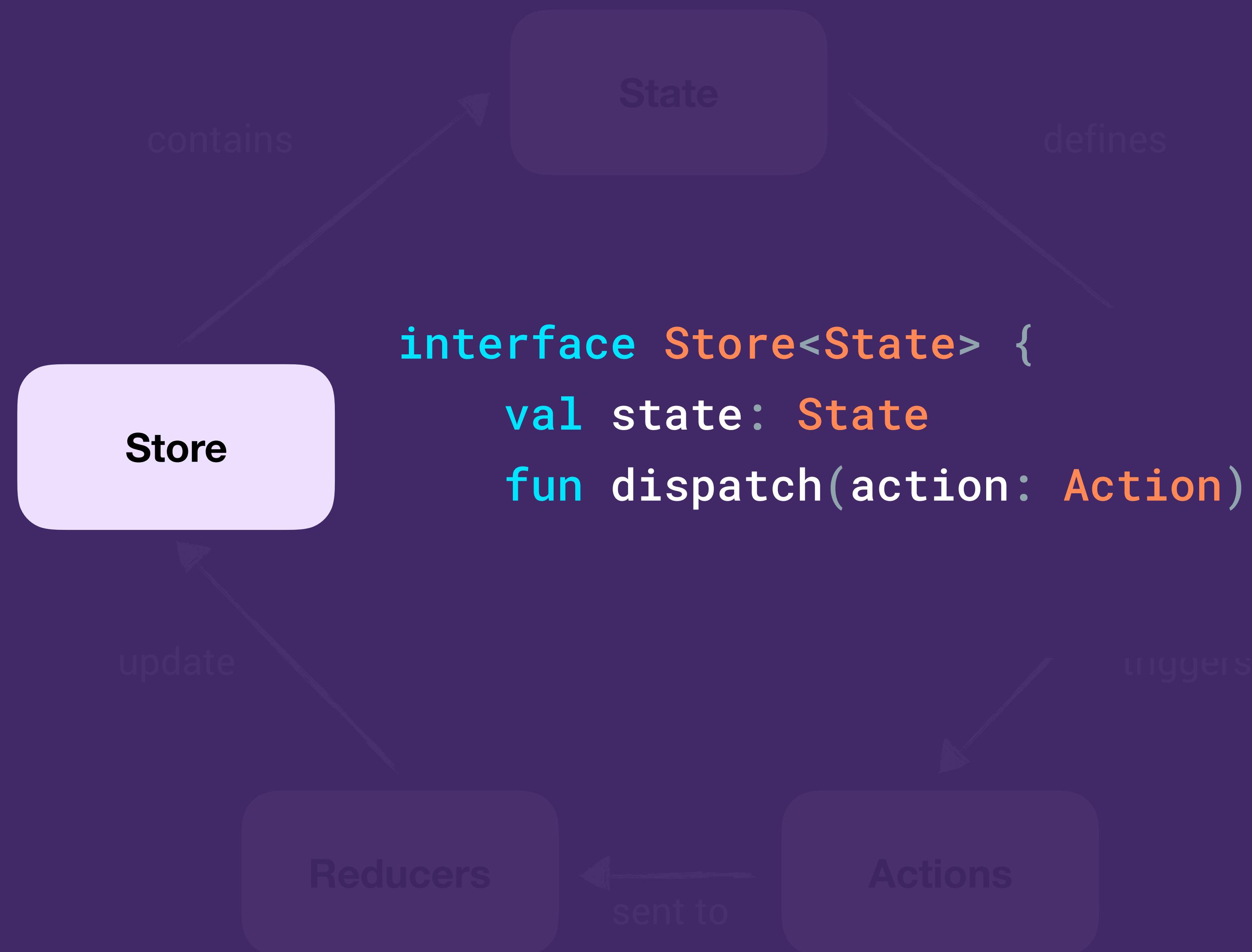


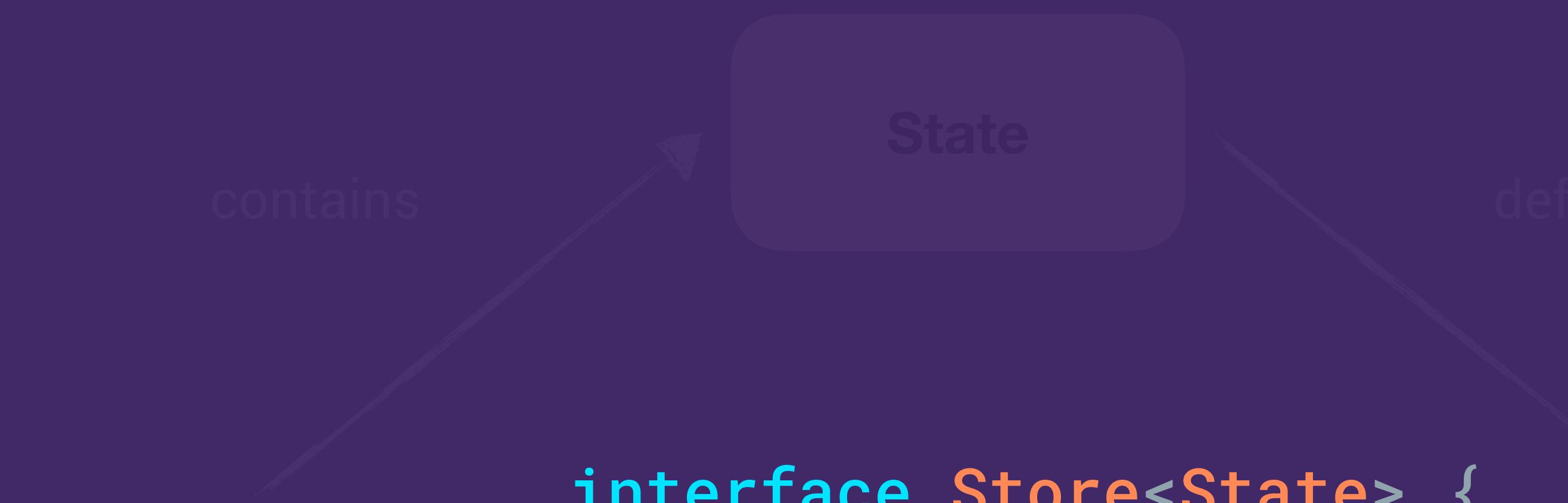




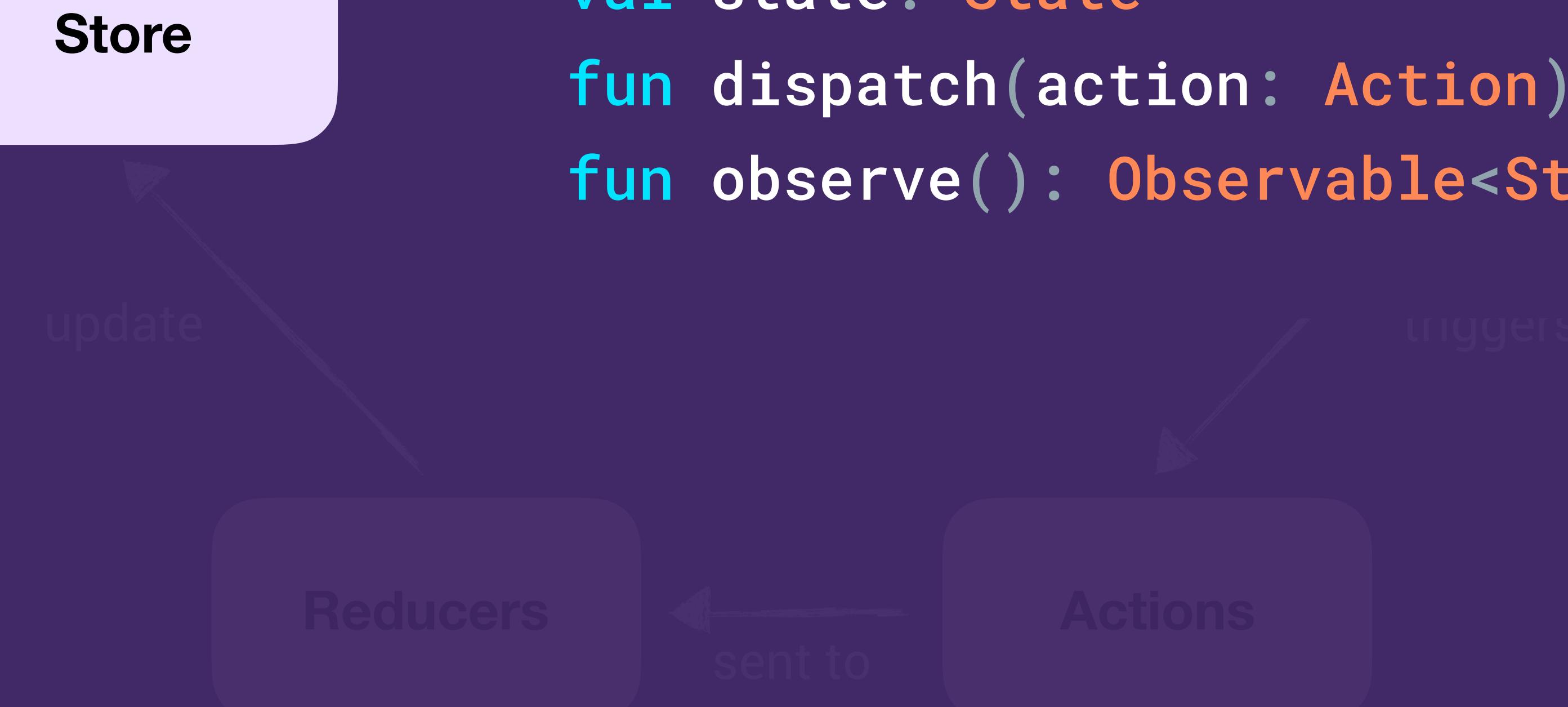








```
interface Store<State> {  
    val state: State  
    fun dispatch(action: Action)  
    fun observe(): Observable<State>
```



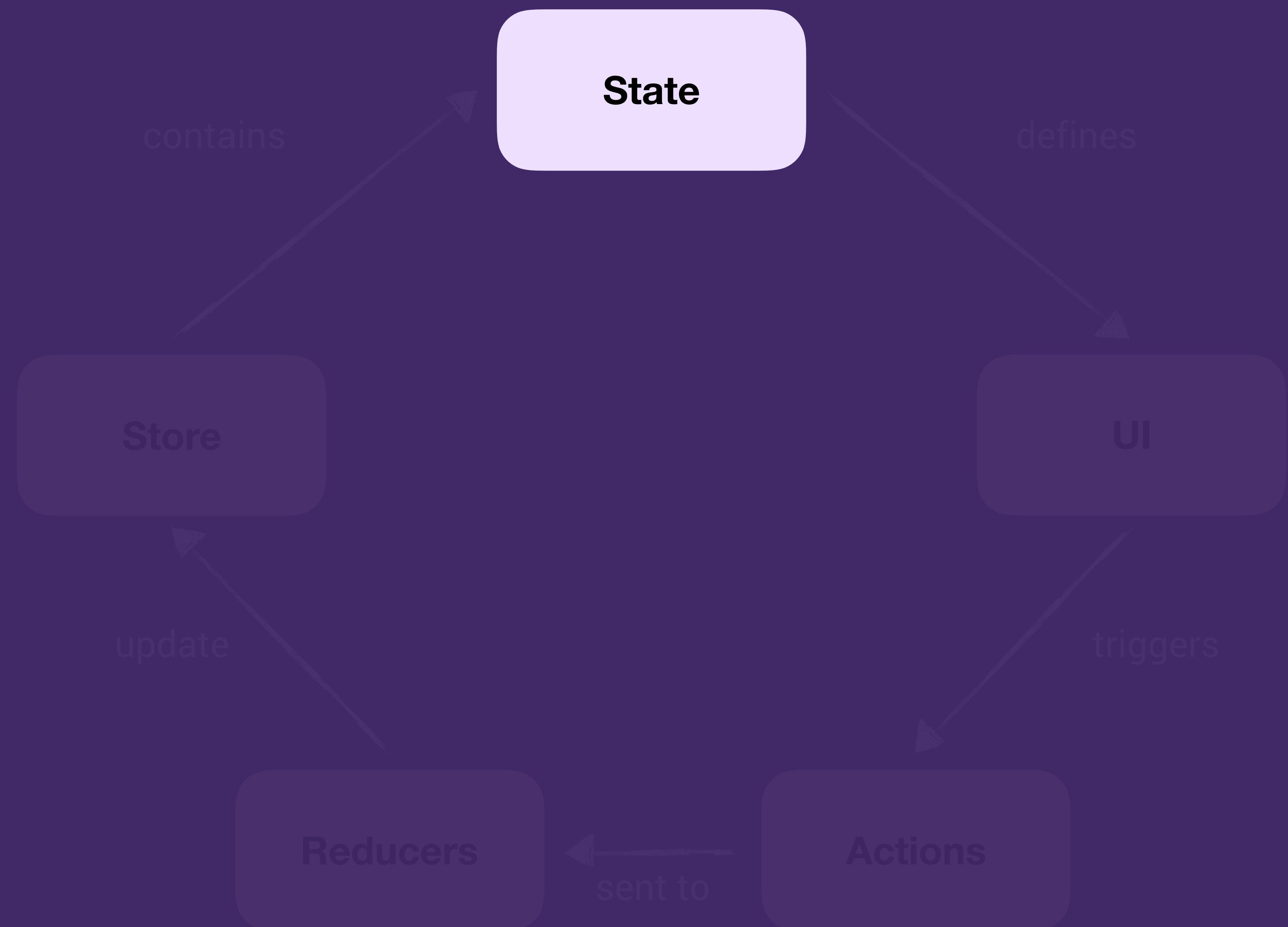


```
interface Store<State> {  
    val state: State  
    fun dispatch(action: Action)  
    fun observe(): Observable<State>
```

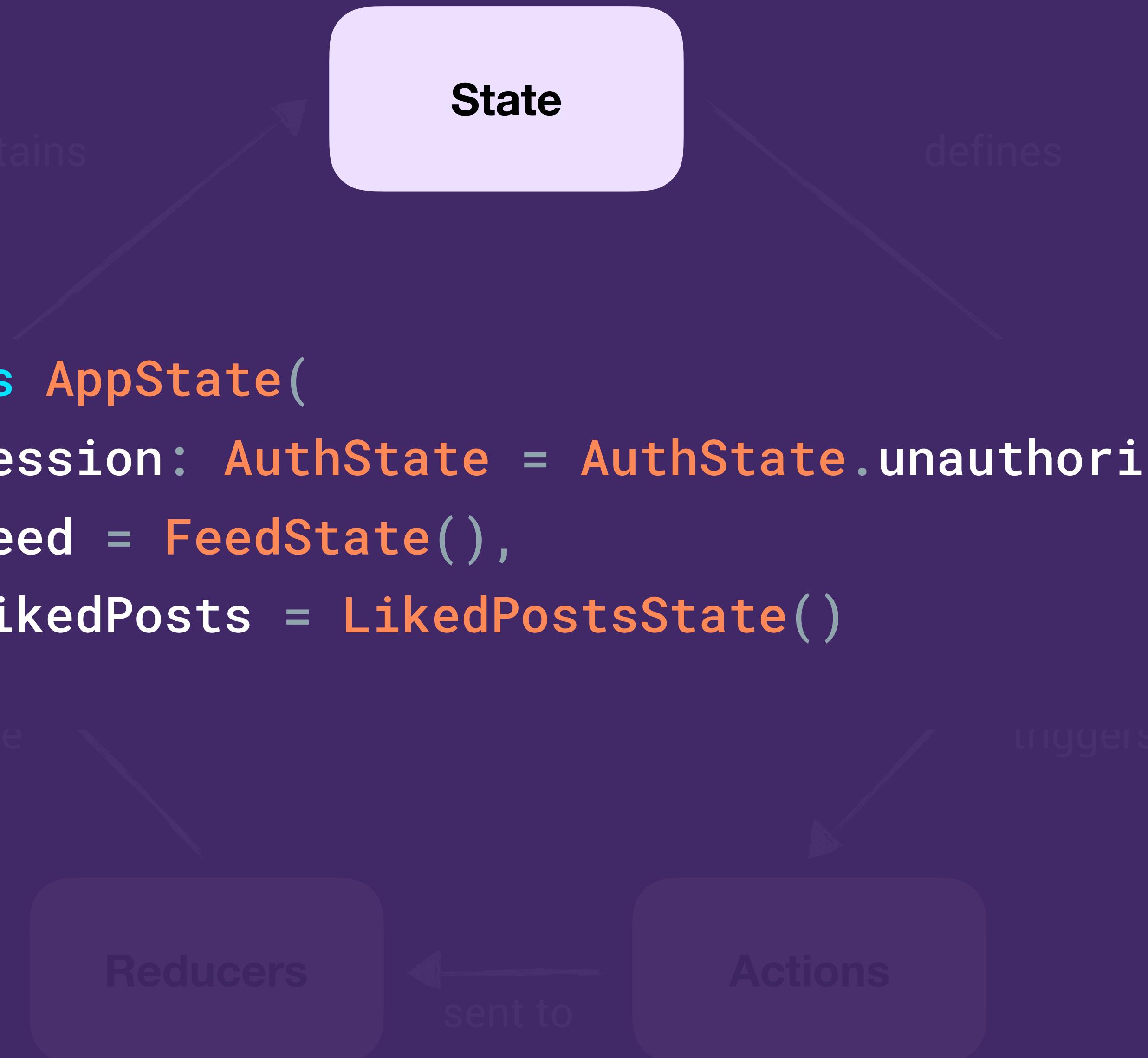
}

sent to

triggers



```
data class AppState(  
    val session: AuthState = AuthState.unauthorised,  
    val feed = FeedState(),  
    val likedPosts = LikedPostsState()  
)
```



## State

```
data class AppState(  
    val session: AuthState = AuthState.unauthorised,  
    val feed = FeedState(),  
    val likedPosts = LikedPostsState()  
)
```

```
AppState.copy(session = AuthState.authorised(token = newUserToken))
```

Reducers

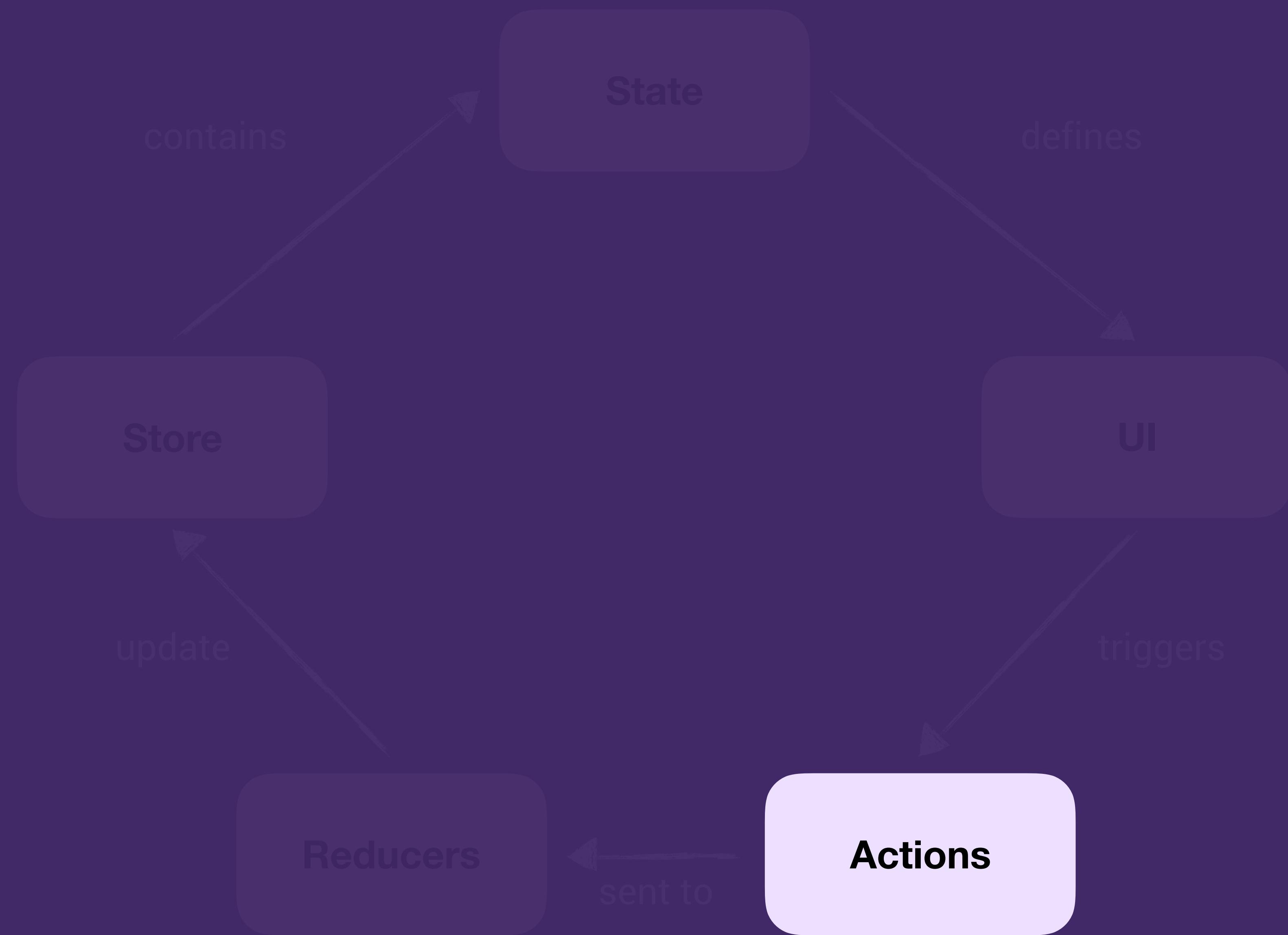
Actions

update

triggers

sent to





```
data class CreatePostAction(val text: String)
```

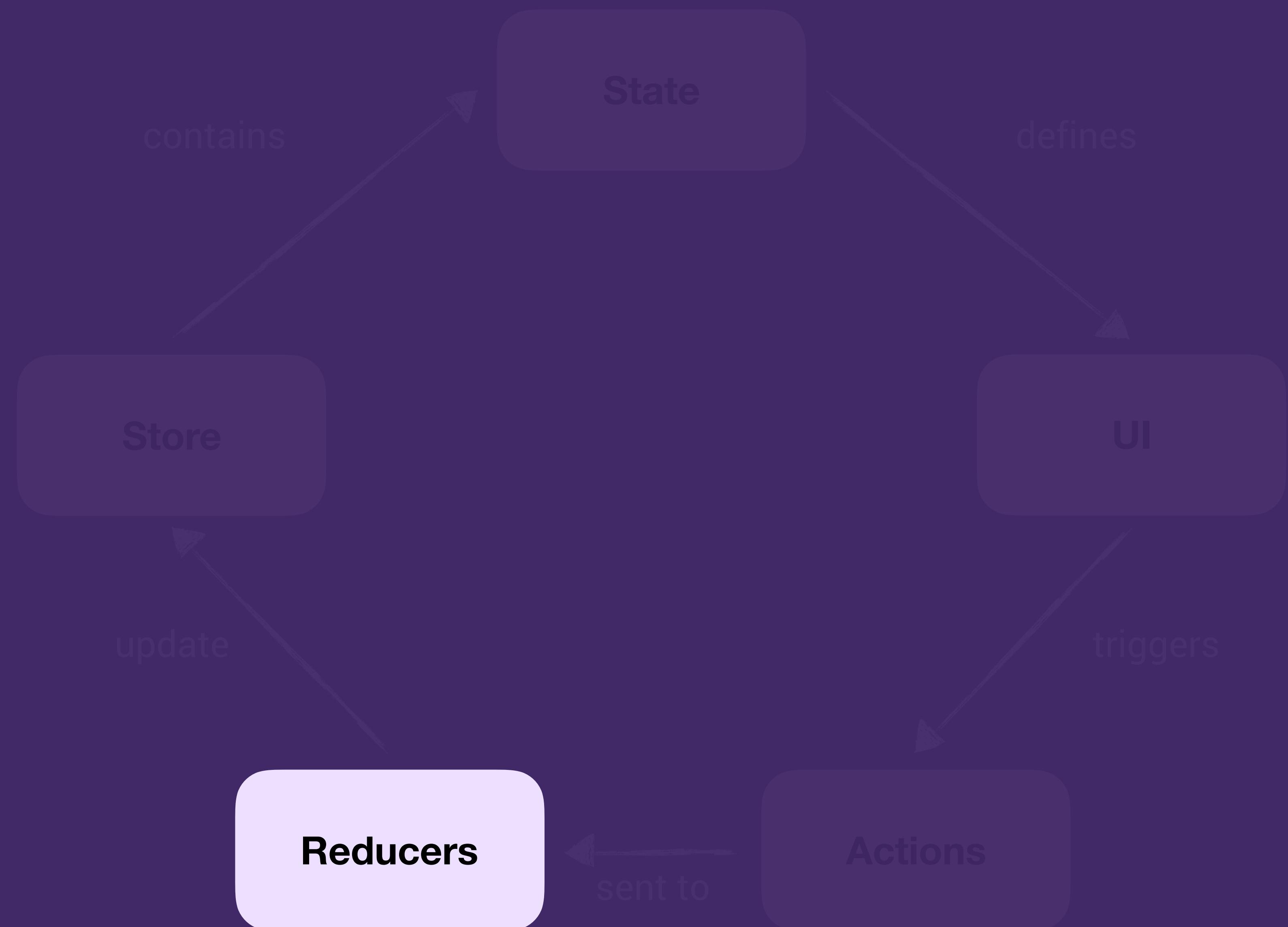


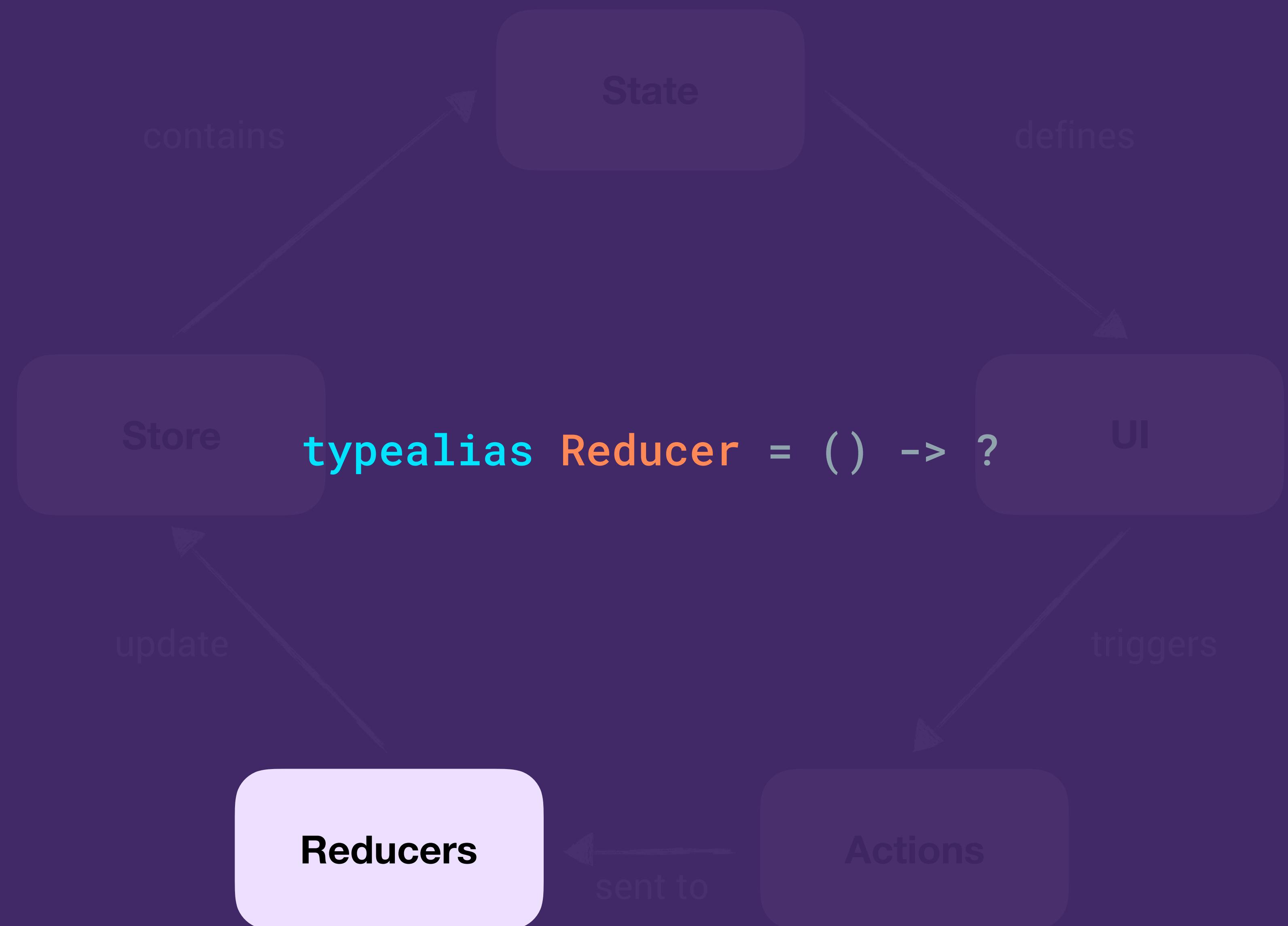
```
graph TD; State[State] -- contains --> CreatePostAction[data class CreatePostAction(val text: String)]; State -- contains --> DeletePostAction[data class DeletePostAction(val id: String)]; Actions[Actions] -- triggers --> CreatePostAction; Actions -- triggers --> DeletePostAction; CreatePostAction -- update --> Reducers[Reducers]; DeletePostAction -- update --> Reducers; Reducers -- sent to --> Actions;
```

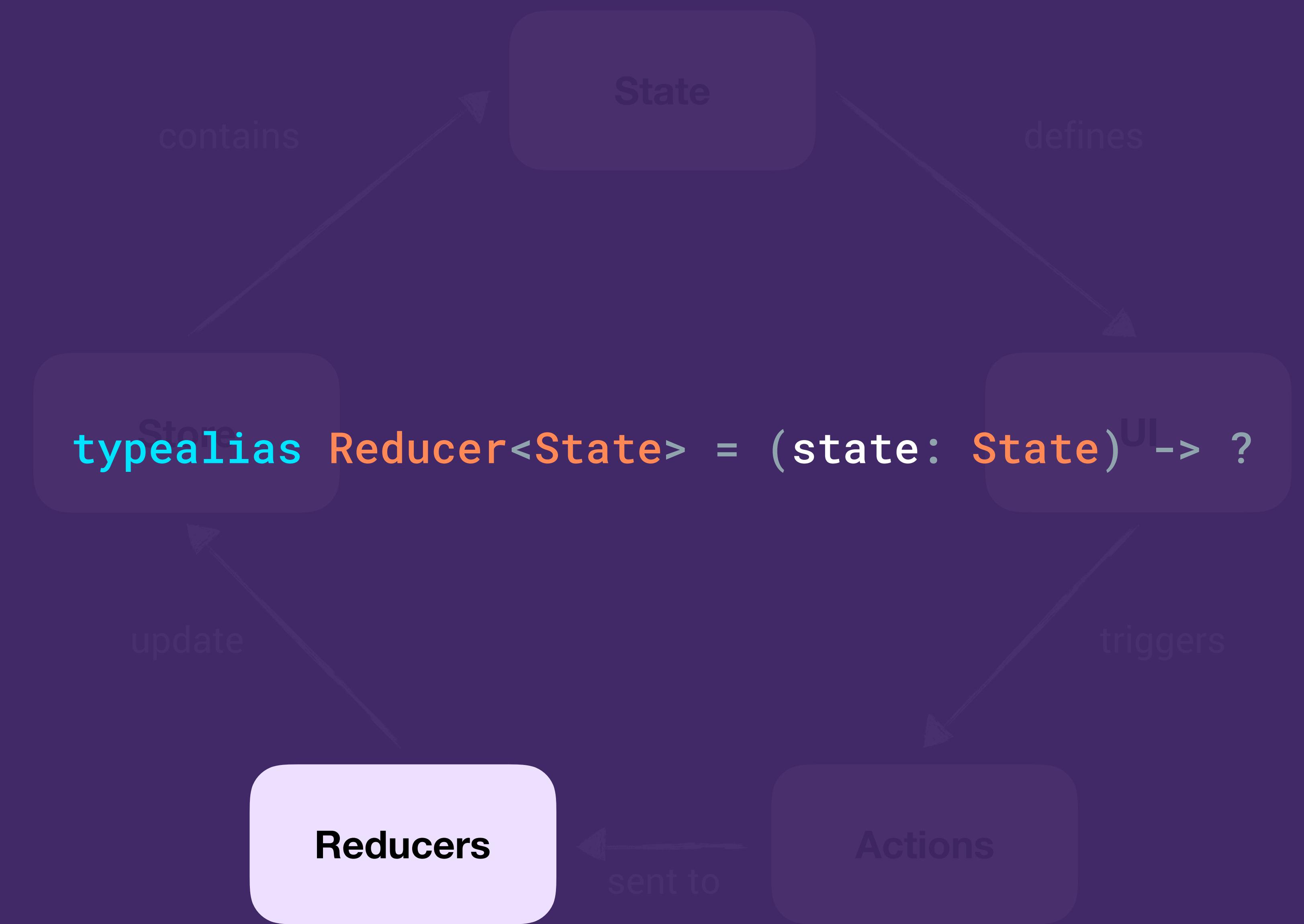
`data class CreatePostAction(val text: String)`  
`data class DeletePostAction(val id: String)`

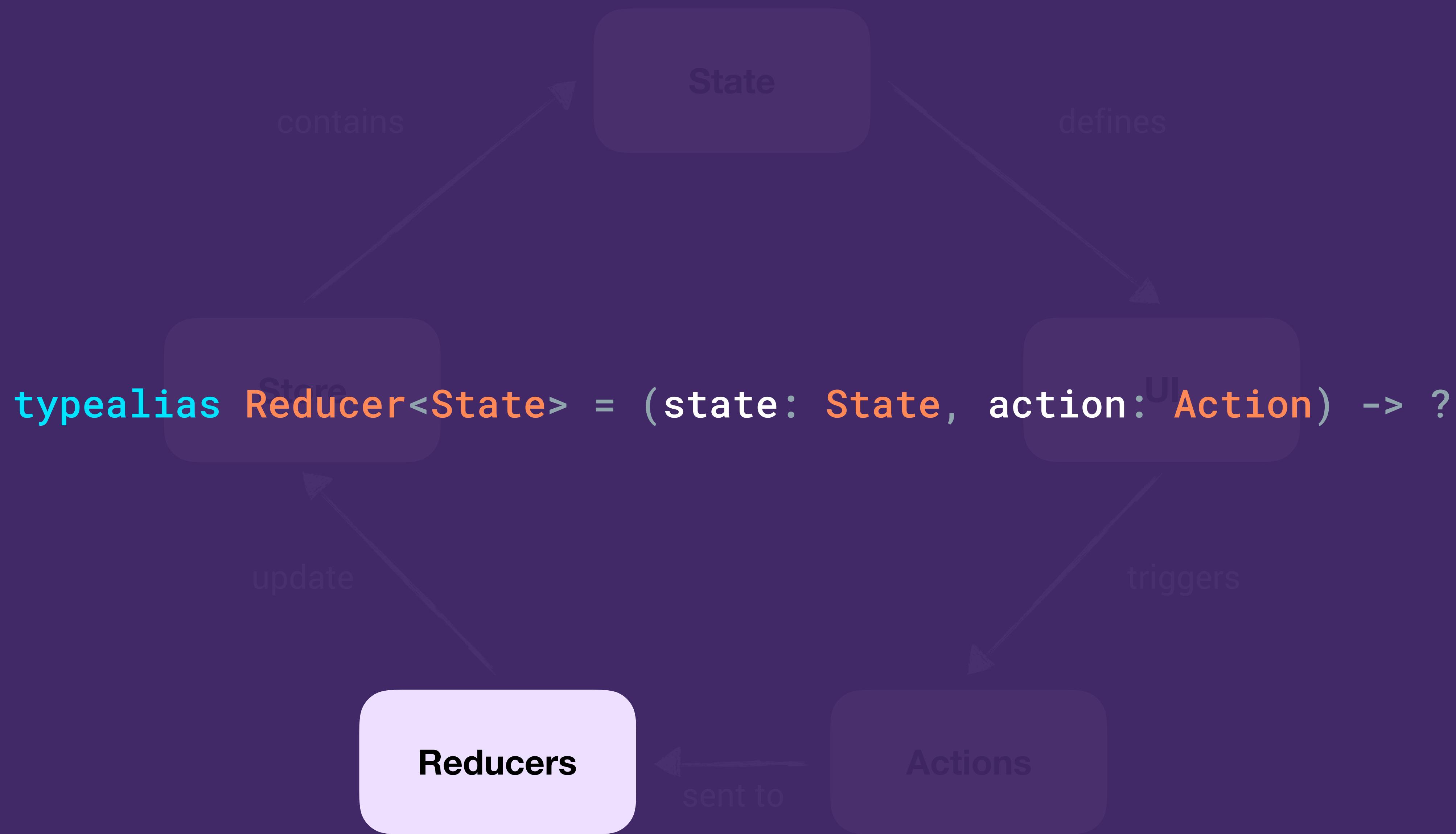
```
graph TD; State[State] -- contains --> Actions[Actions]; Actions -- triggers --> Reducers[Reducers]; Reducers -- update --> State; Reducers -- sent to --> Actions
```

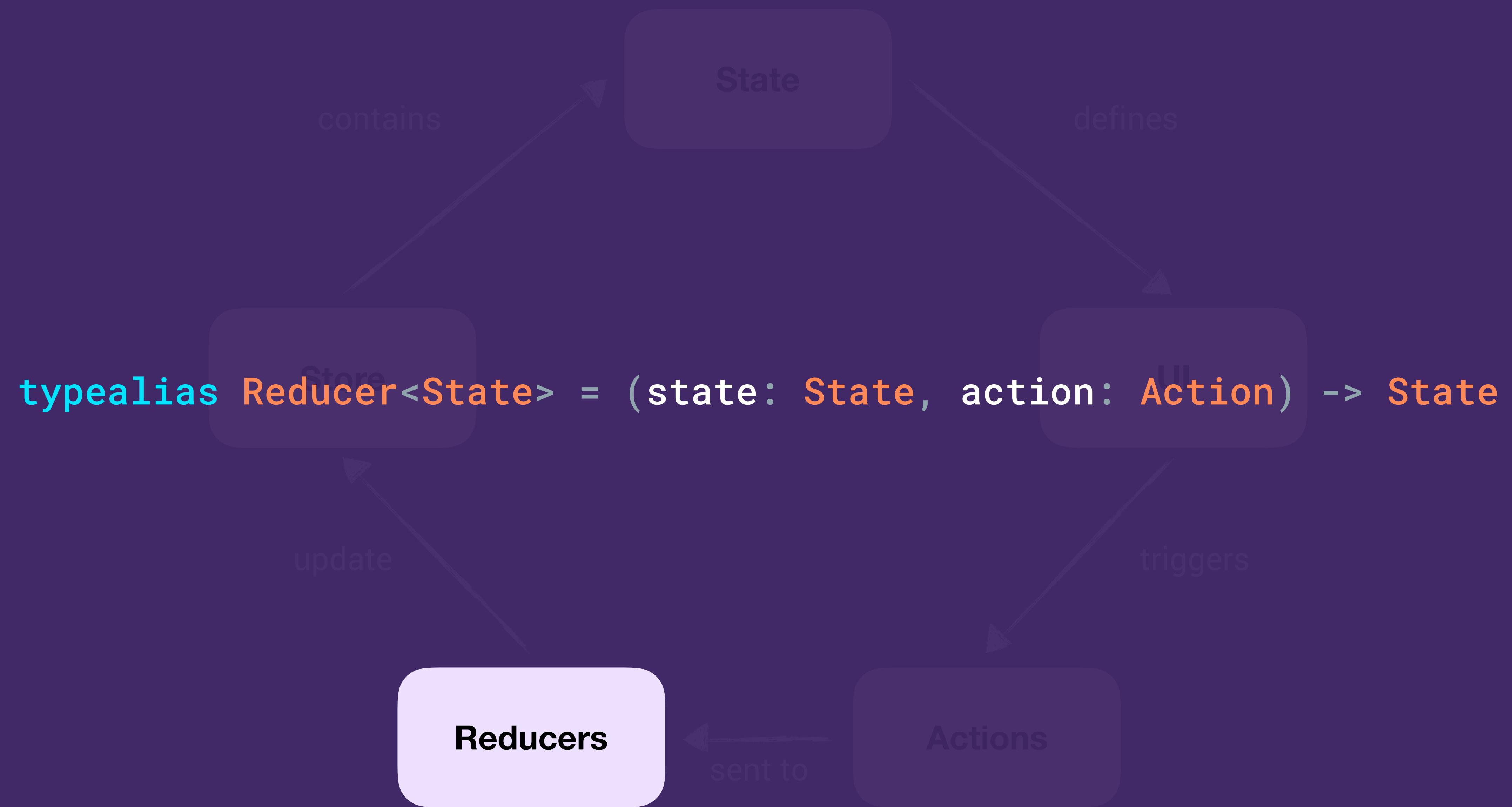
`data class CreatePostAction(val text: String)`  
`data class DeletePostAction(val id: String)`  
`data class LikePostAction(val id: String)`

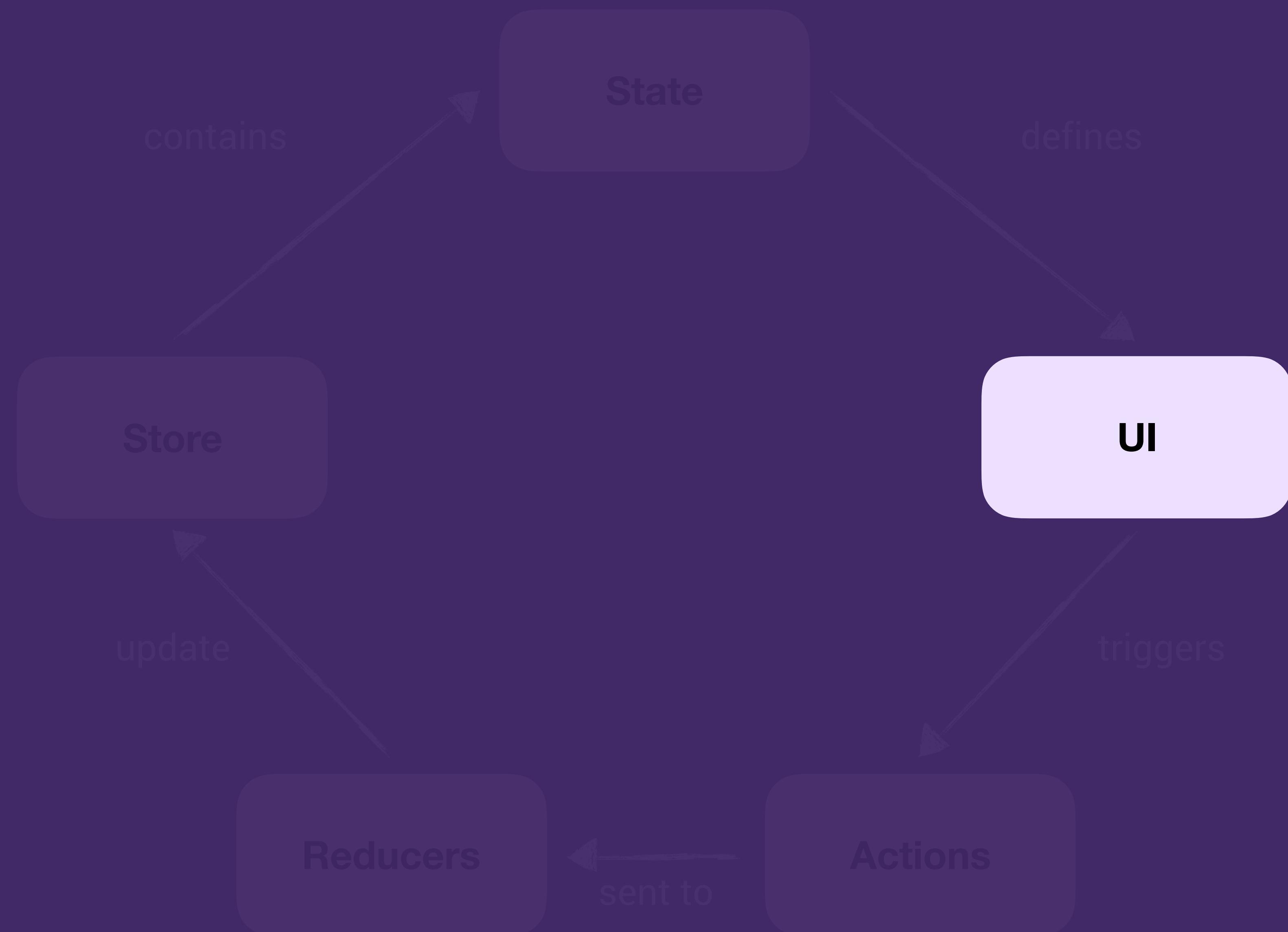














Initial State



Initial State

`StartLoginAction(username="Anon", password="password")`



Initial State

`StartLoginAction(username="Anon", password="password")`

`CompleteLoginAction(secretAuthToken="12345678")`



Initial State

`StartLoginAction(username="Anon", password="password")`

`CompleteLoginAction(secretAuthToken="12345678")`

`FetchPostsAction()`



Initial State

`StartLoginAction(username="Anon", password="password")`

`CompleteLoginAction(secretAuthToken="12345678")`

`FetchPostsAction()`

`CompletePostsAction(posts=[Post(id="1", text="Hello world")])`





Initial State

`StartLoginAction(username="Anon", password="password")`

`CompleteLoginAction(secretAuthToken="12345678")`

`FetchPostsAction()`

`CompletePostsAction(posts=[Post(id="1", text="Hello world")])`

`LikePostAction(id="1")`

Initial State

`StartLoginAction(username="Anon", password="password")`

`CompleteLoginAction(secretAuthToken="12345678")`

`FetchPostsAction()`

`CompletePostsAction(posts=[Post(id="1", text="Hello world")])`

`LikePostAction(id="1")`

`LikePostFailureAction()`

```
[  
  {  
    type: "StartLoginAction",  
    payload: {  
      username: "Anon",  
      password: "password"  
    }  
  },  
  {  
    type: "CompleteLoginAction",  
    payload: {  
      secretAuthToken: "12345678"  
    }  
  },  
  {  
    type: "FetchPostsAction"  
  },  
  {  
    type: "CompletePostsAction",  
    payload: {  
      posts: [  
        {  
          id: "1",  
          text: "Hello world"  
        }  
      ]  
    }  
  },  
  {  
    type: "LikePostAction",  
    payload: {  
      postID: "1",  
      userID: "1"  
    }  
  }]
```

Initial State

`StartLoginAction(username="Anon", password="password")`

`CompleteLoginAction(secretAuthToken="12345678")`

`FetchPostsAction()`

`CompletePostsAction(posts=[Post(id="1", text="Hello world")])`

`LikePostAction(id="1")`

`LikePostFailureAction()`

Initial State

`StartLoginAction(username="Anon", password="password")`

`CompleteLoginAction(secretAuthToken="12345678")`

`FetchPostsAction()`

`CompletePostsAction(posts=[Post(id="1", text="Hello world")])`

`LikePostAction(id="1")`

`LikePostFailureAction()`

Initial State

`StartLoginAction(username="Anon", password="password")`

`CompleteLoginAction(secretAuthToken="12345678")`

`FetchPostsAction()`

`CompletePostsAction(posts=[Post(id="1", text="Hello world")])`

`LikePostAction(id="1")`

`LikePostFailureAction()`

