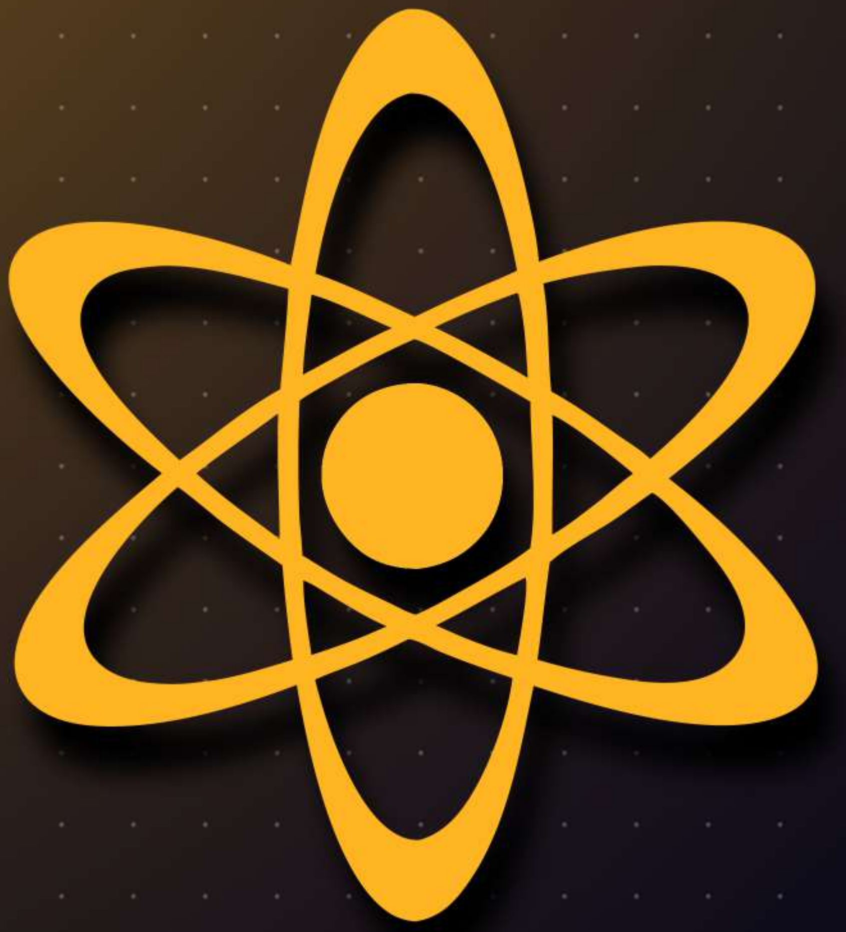
What Is Redux Thunk And Why is It Used?

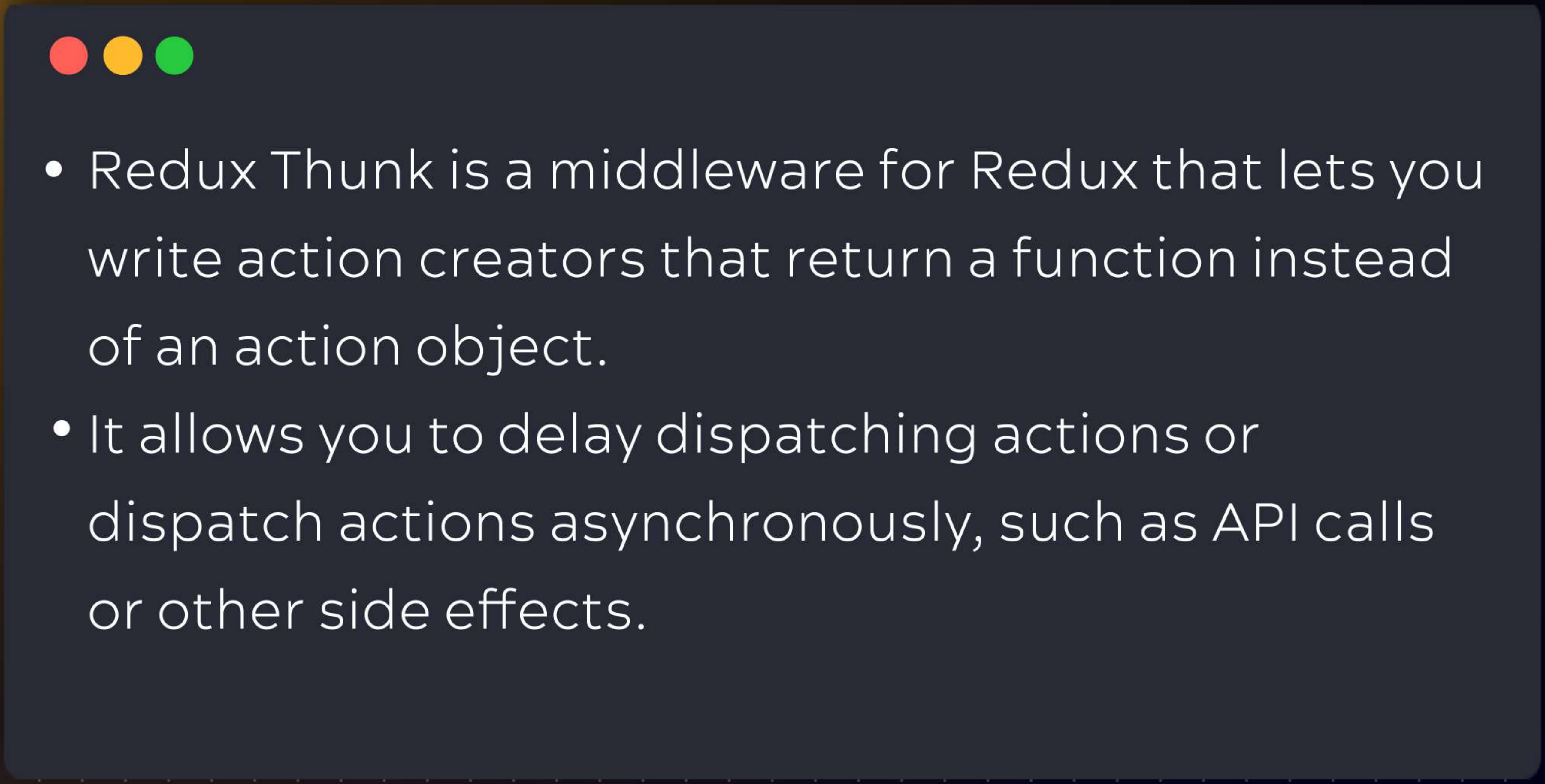


SUMANTH M

Frontend developer

1

What is Redux Thunk?

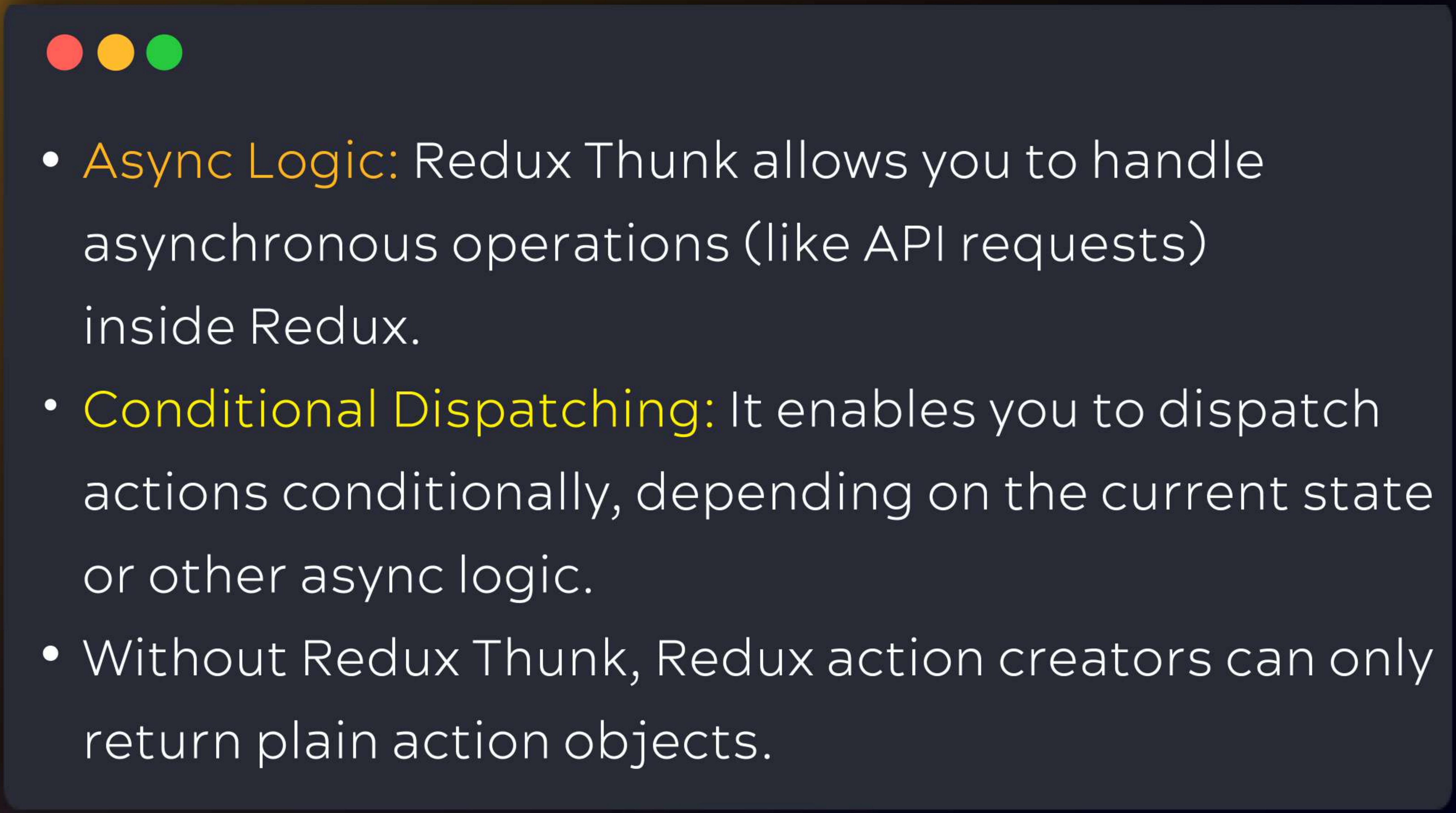


- Redux Thunk is a middleware for Redux that lets you write action creators that return a function instead of an action object.
- It allows you to delay dispatching actions or dispatch actions asynchronously, such as API calls or other side effects.

SUMANTH M

Frontend developer

2 Why Use Redux Thunk?

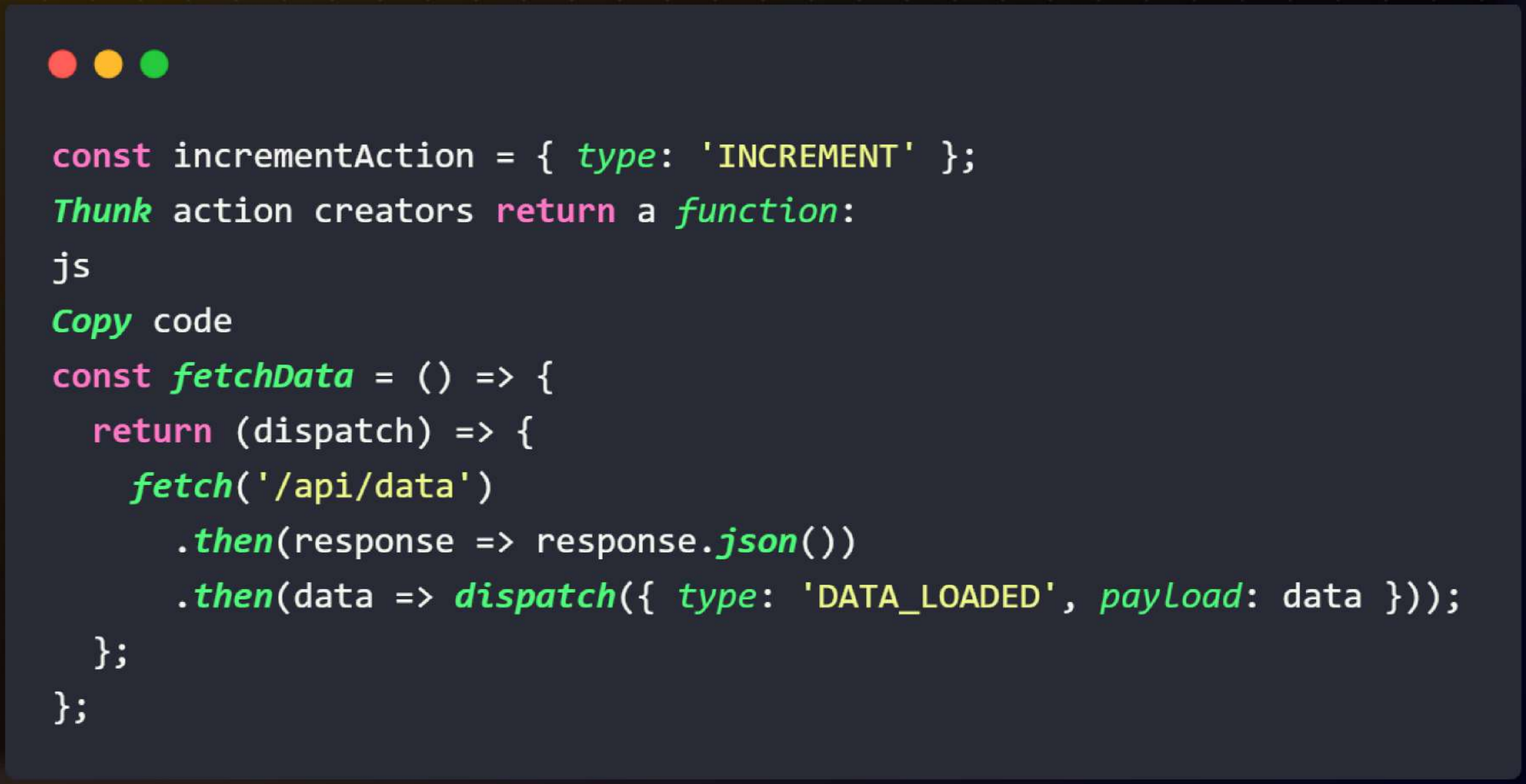


- Async Logic: Redux Thunk allows you to handle asynchronous operations (like API requests) inside Redux.
- Conditional Dispatching: It enables you to dispatch actions conditionally, depending on the current state or other async logic.
- Without Redux Thunk, Redux action creators can only return plain action objects.

SUMANTH M
Frontend developer

3

How Does Redux Thunk Work?



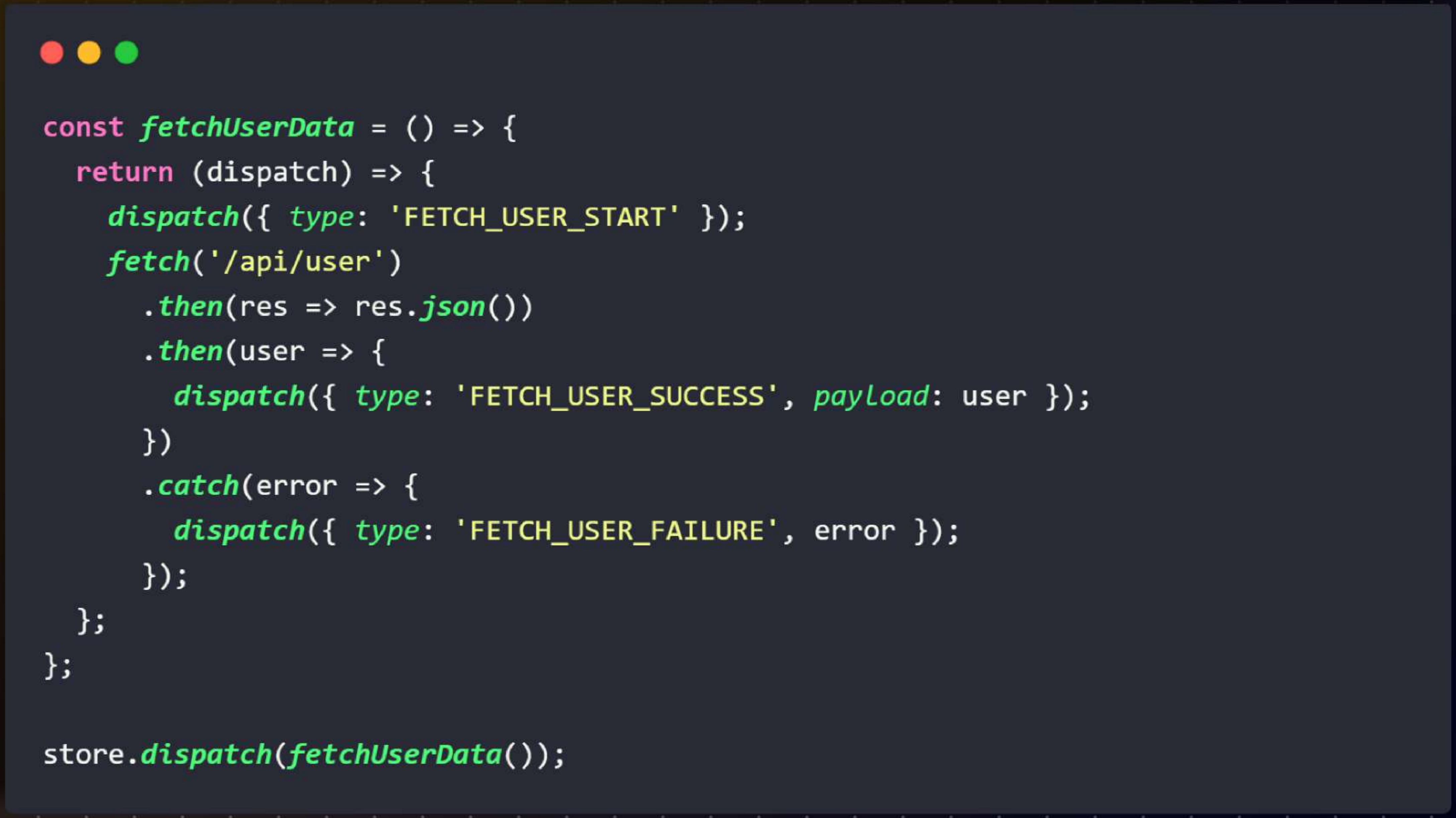
```
const incrementAction = { type: 'INCREMENT' };
Thunk action creators return a function:
js
Copy code
const fetchData = () => {
  return (dispatch) => {
    fetch('/api/data')
      .then(response => response.json())
      .then(data => dispatch({ type: 'DATA_LOADED', payload: data }));
  };
};
```

Thunk is just a function that can contain async logic and dispatch actions after async tasks (e.g., after an API call). Normal action creators return plain objects:

SUMANTH M
Frontend developer

4

Example:
Redux Thunk in Action



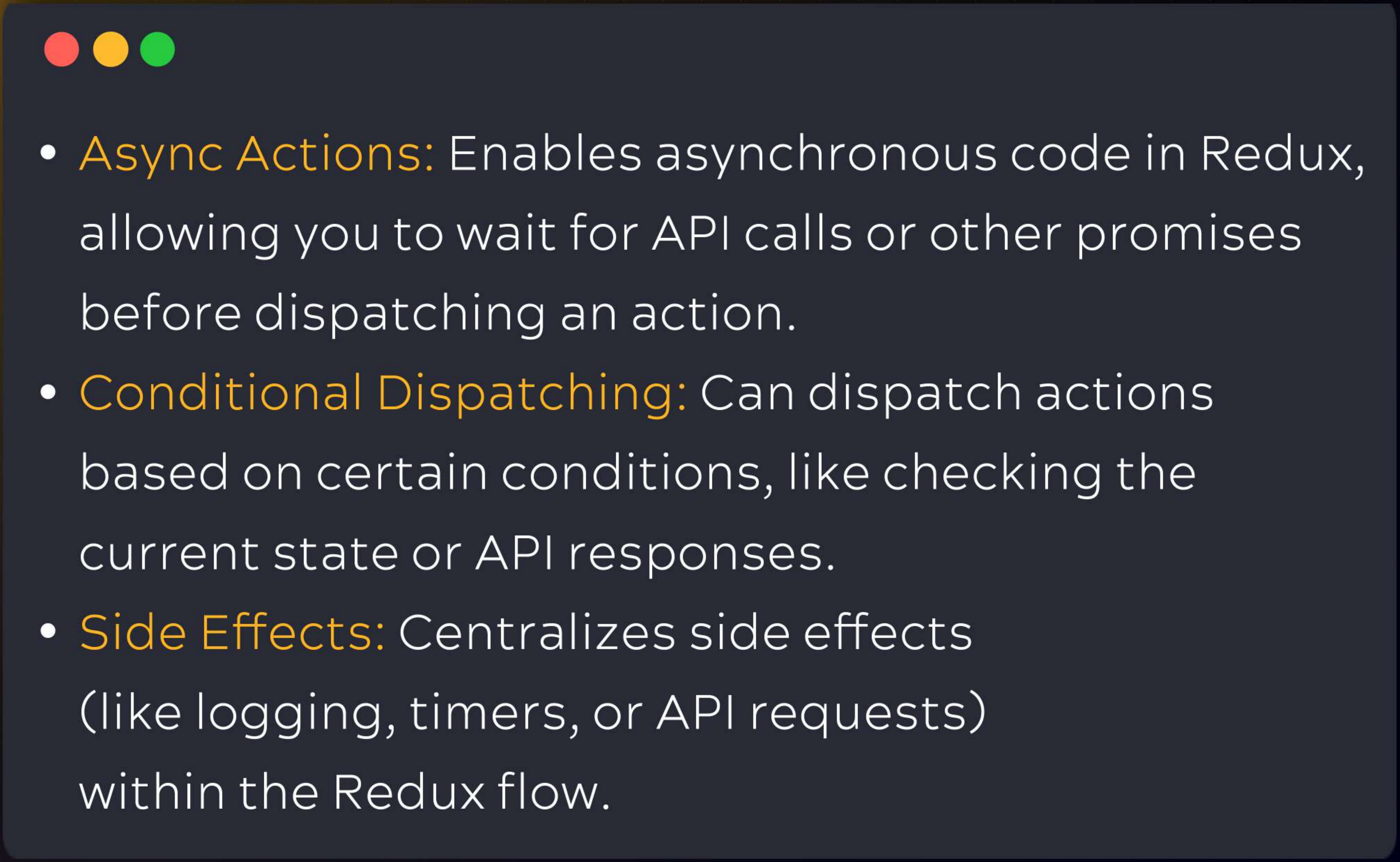
```
const fetchUserData = () => {
  return (dispatch) => {
    dispatch({ type: 'FETCH_USER_START' });
    fetch('/api/user')
      .then(res => res.json())
      .then(user => {
        dispatch({ type: 'FETCH_USER_SUCCESS', payload: user });
      })
      .catch(error => {
        dispatch({ type: 'FETCH_USER_FAILURE', error });
      });
  };
};

store.dispatch(fetchUserData());
```

Scenario: Fetching user data from an API.

Key Point: Redux Thunk lets you handle async logic like API calls while dispatching different actions at different stages

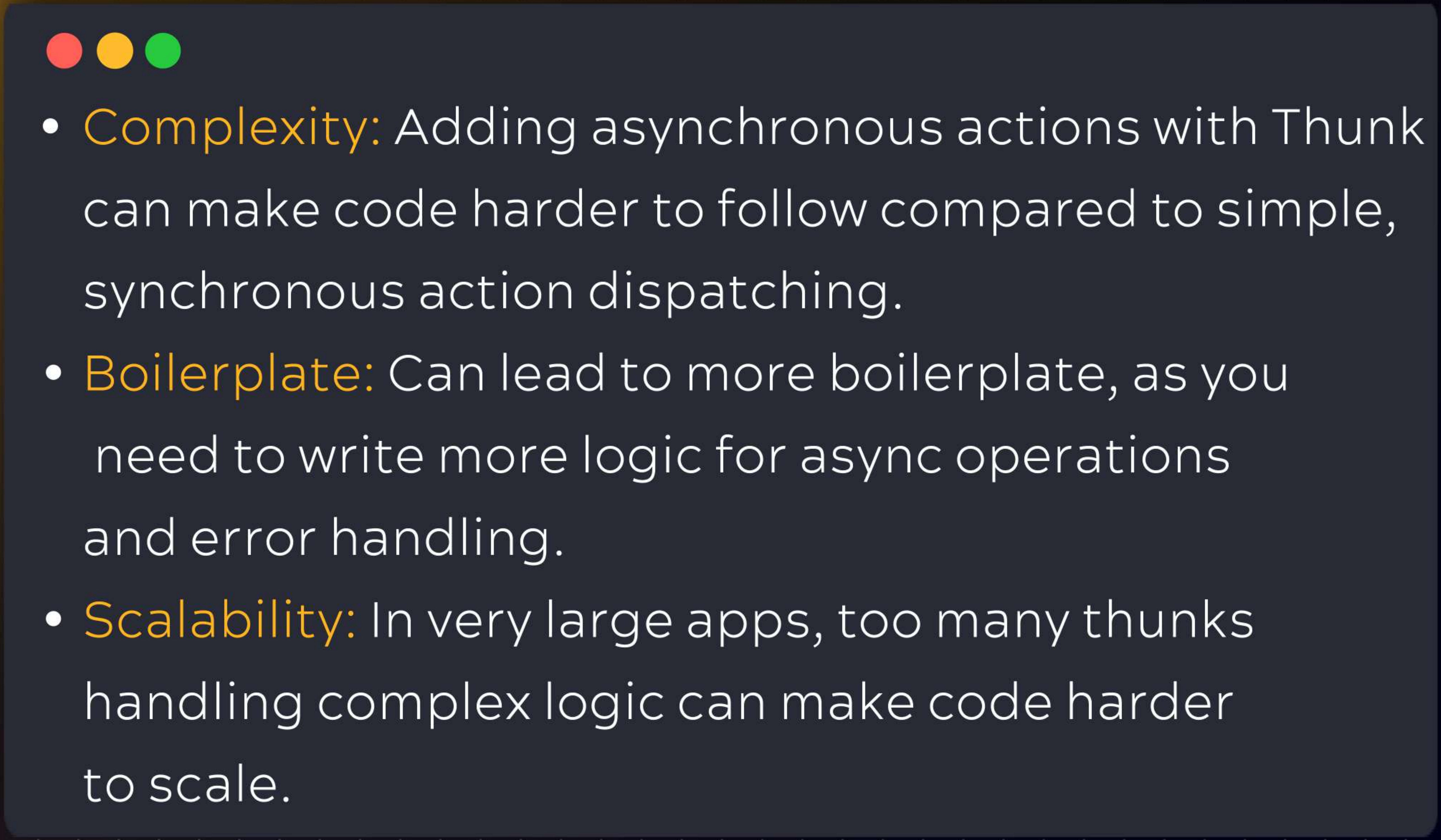
5 Advantages of Redux Thunk



- Async Actions: Enables asynchronous code in Redux, allowing you to wait for API calls or other promises before dispatching an action.
- Conditional Dispatching: Can dispatch actions based on certain conditions, like checking the current state or API responses.
- Side Effects: Centralizes side effects (like logging, timers, or API requests) within the Redux flow.

6

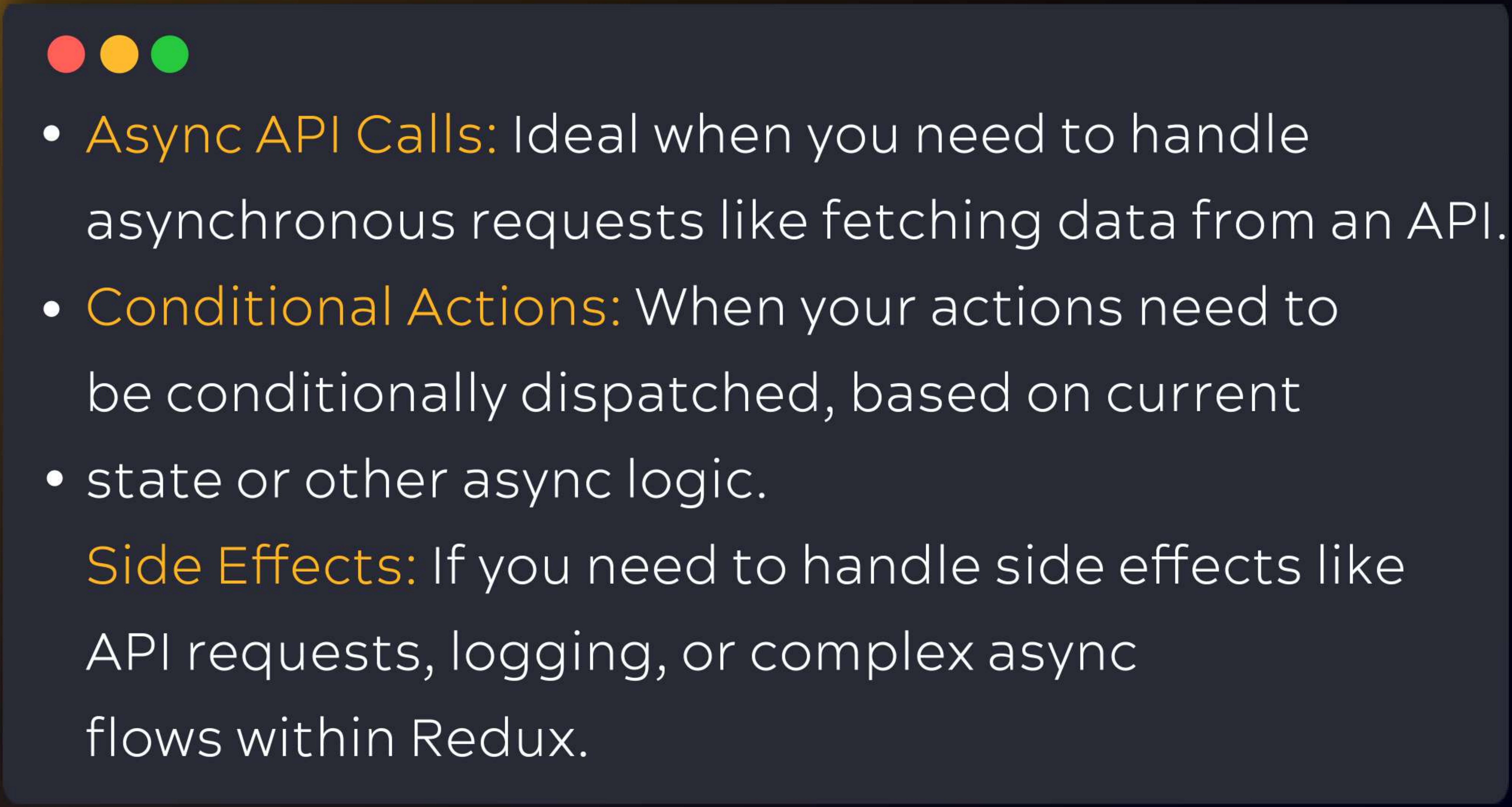
Disadvantages/ Limitations of Redux Thunk



- Complexity: Adding asynchronous actions with Thunk can make code harder to follow compared to simple, synchronous action dispatching.
- Boilerplate: Can lead to more boilerplate, as you need to write more logic for async operations and error handling.
- Scalability: In very large apps, too many thunks handling complex logic can make code harder to scale.

SUMANTH M
Frontend developer

When to Use Redux Thunk



- Async API Calls: Ideal when you need to handle asynchronous requests like fetching data from an API.
- Conditional Actions: When your actions need to be conditionally dispatched, based on current
- state or other async logic.

  Side Effects: If you need to handle side effects like API requests, logging, or complex async flows within Redux.

SUMANTH M

Frontend developer

React Js

8

When NOT to Use Redux Thunk

Simple, Synchronous Logic: If your app doesn’t require asynchronous actions or side effects, Redux Thunk may not be necessary.

Alternatives: If your app requires more complex async flows, tools like Redux-Saga or React Query might be better suited for handling side effects in a more structured way.

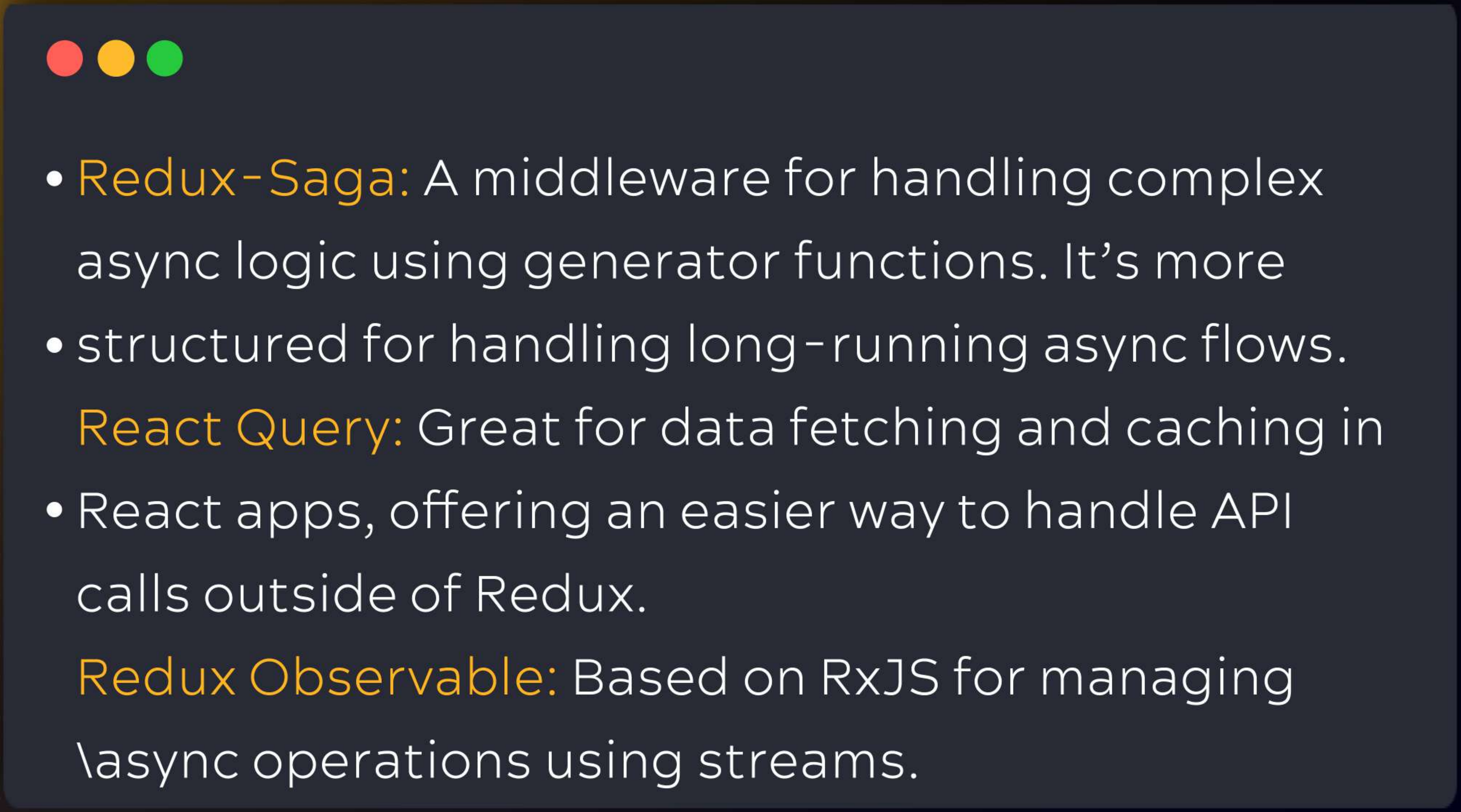
SUMANTH M

Frontend developer

9

React Js

Alternatives to Redux



- Redux-Saga: A middleware for handling complex async logic using generator functions. It’s more
- structured for handling long-running async flows. React Query: Great for data fetching and caching in
- React apps, offering an easier way to handle API calls outside of Redux. Redux Observable: Based on RxJS for managing \async operations using streams.

SUMANTH M

Frontend developer

Conclusion

Redux Thunk is a powerful middleware that allows Redux to handle asynchronous actions like API calls.

It's a great tool for apps that require async logic, but for larger projects, more advanced middleware like Redux-Saga may be worth considering.

Question: Have you used Redux Thunk in your projects? Let's discuss how it has helped with async operations in your apps!

SUMANTH M
Frontend developer