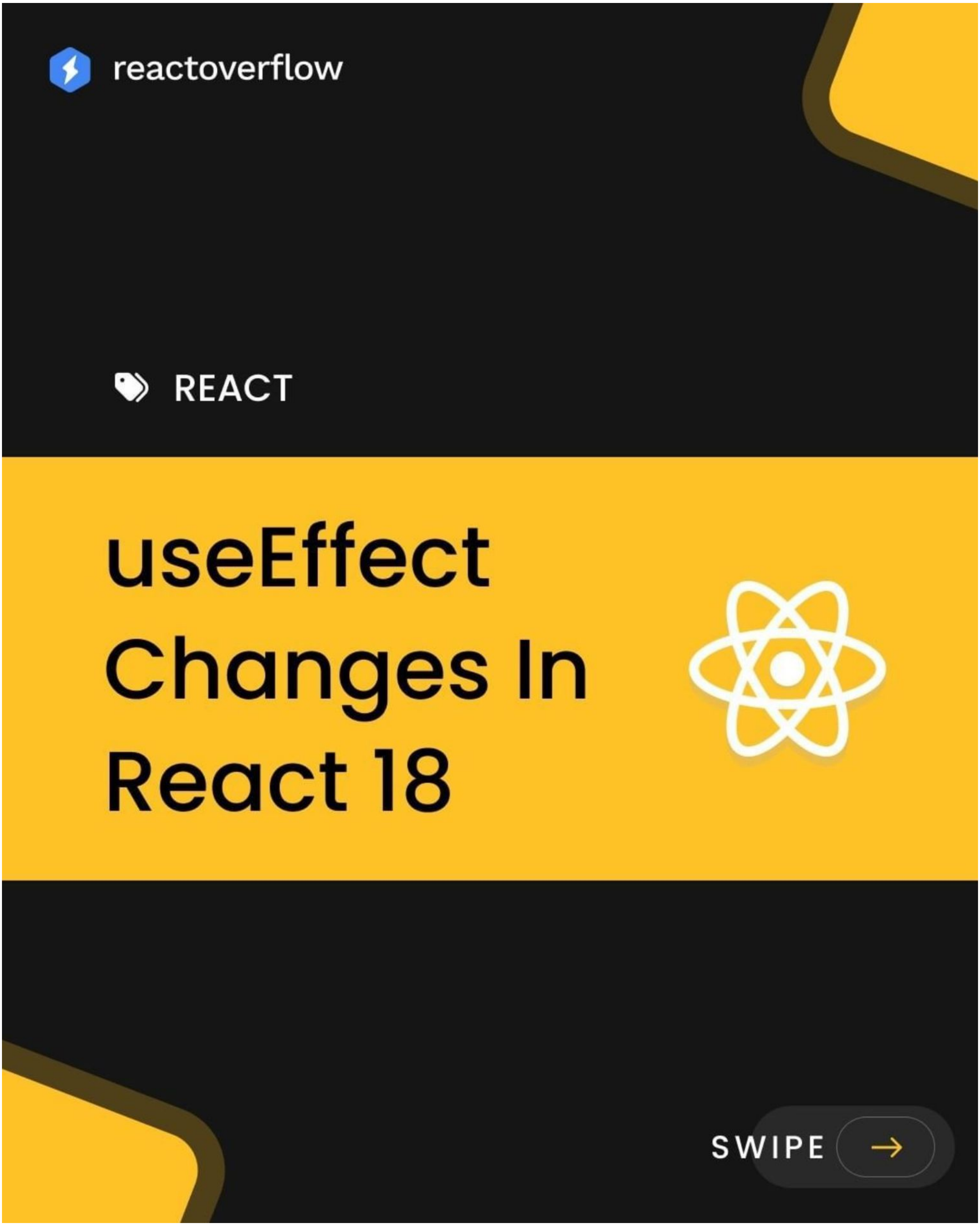
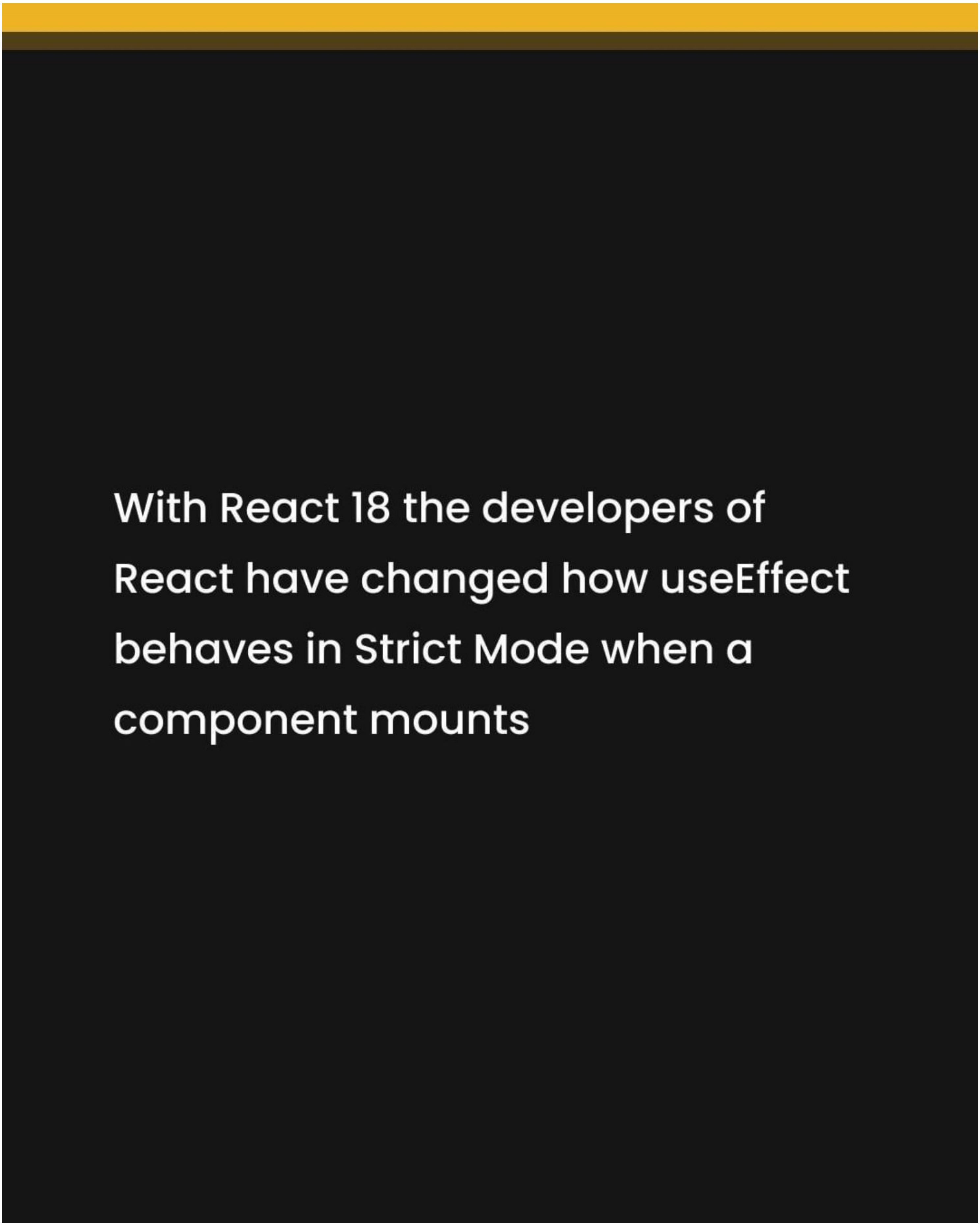


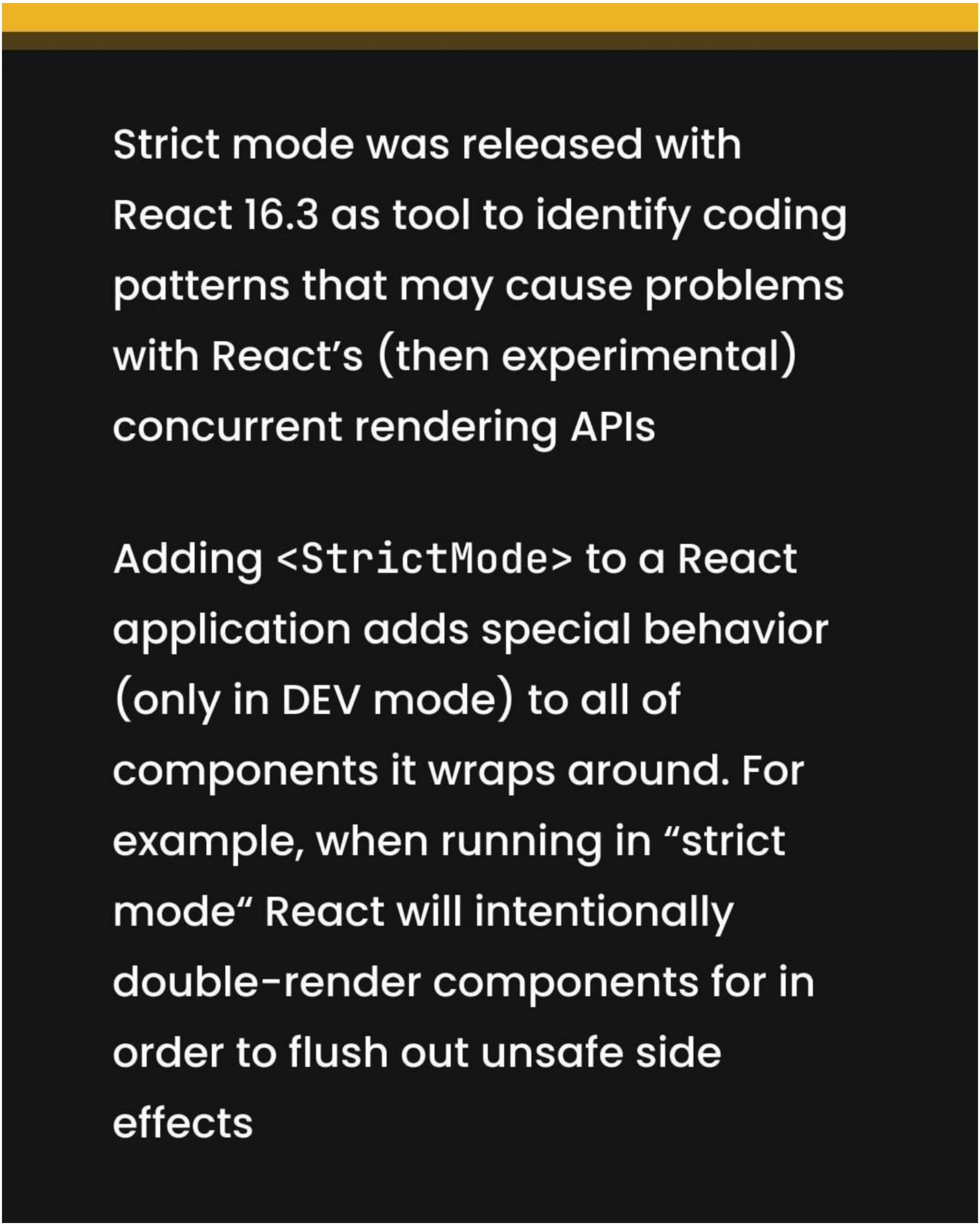
reactoverflow

REACT

useEffect Changes In React 18

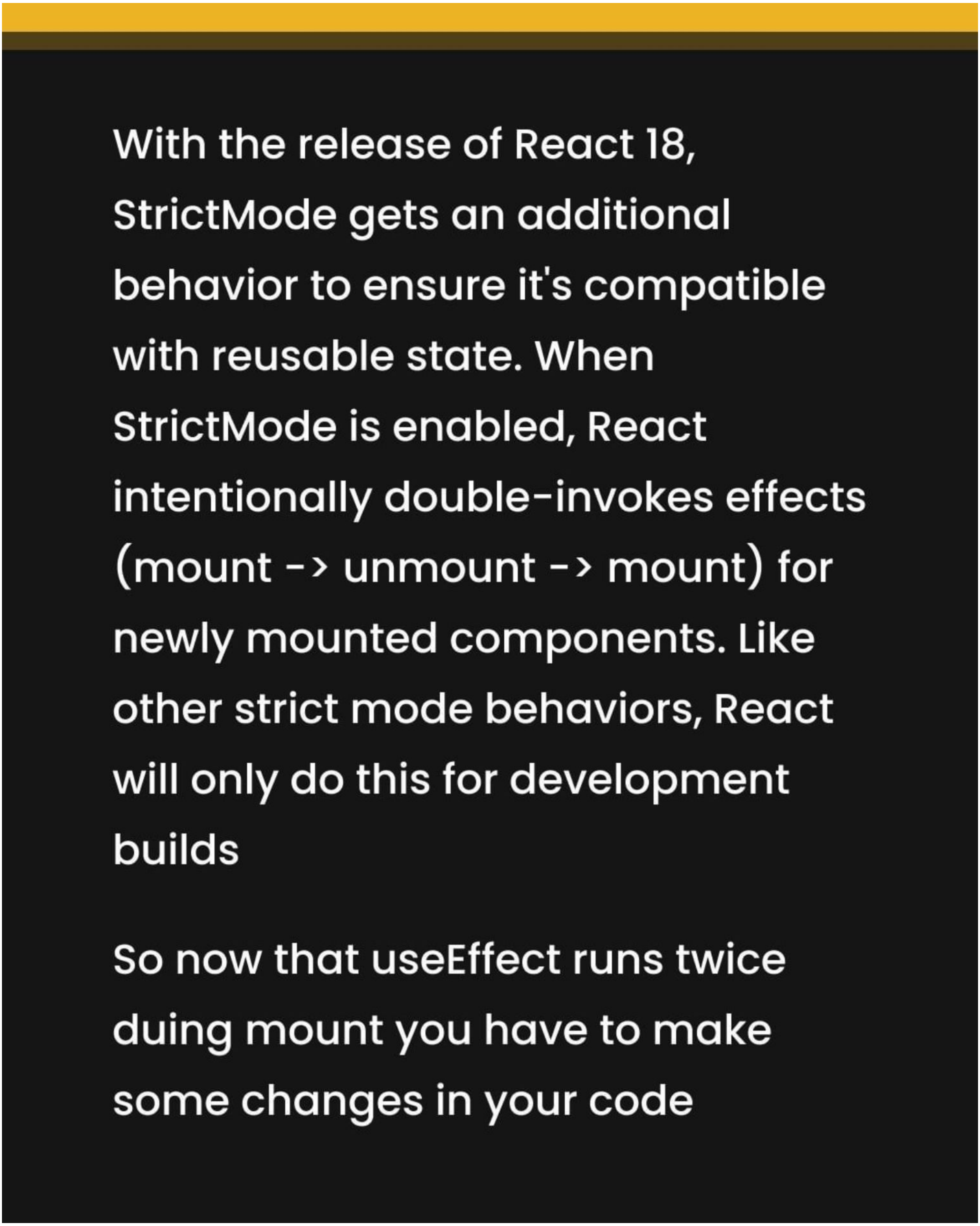


With React 18 the developers of React have changed how useEffect behaves in Strict Mode when a component mounts



Strict mode was released with React 16.3 as tool to identify coding patterns that may cause problems with React’s (then experimental) concurrent rendering APIs

Adding `<StrictMode>` to a React application adds special behavior (only in DEV mode) to all of components it wraps around. For example, when running in “strict mode“ React will intentionally double-render components for in order to flush out unsafe side effects

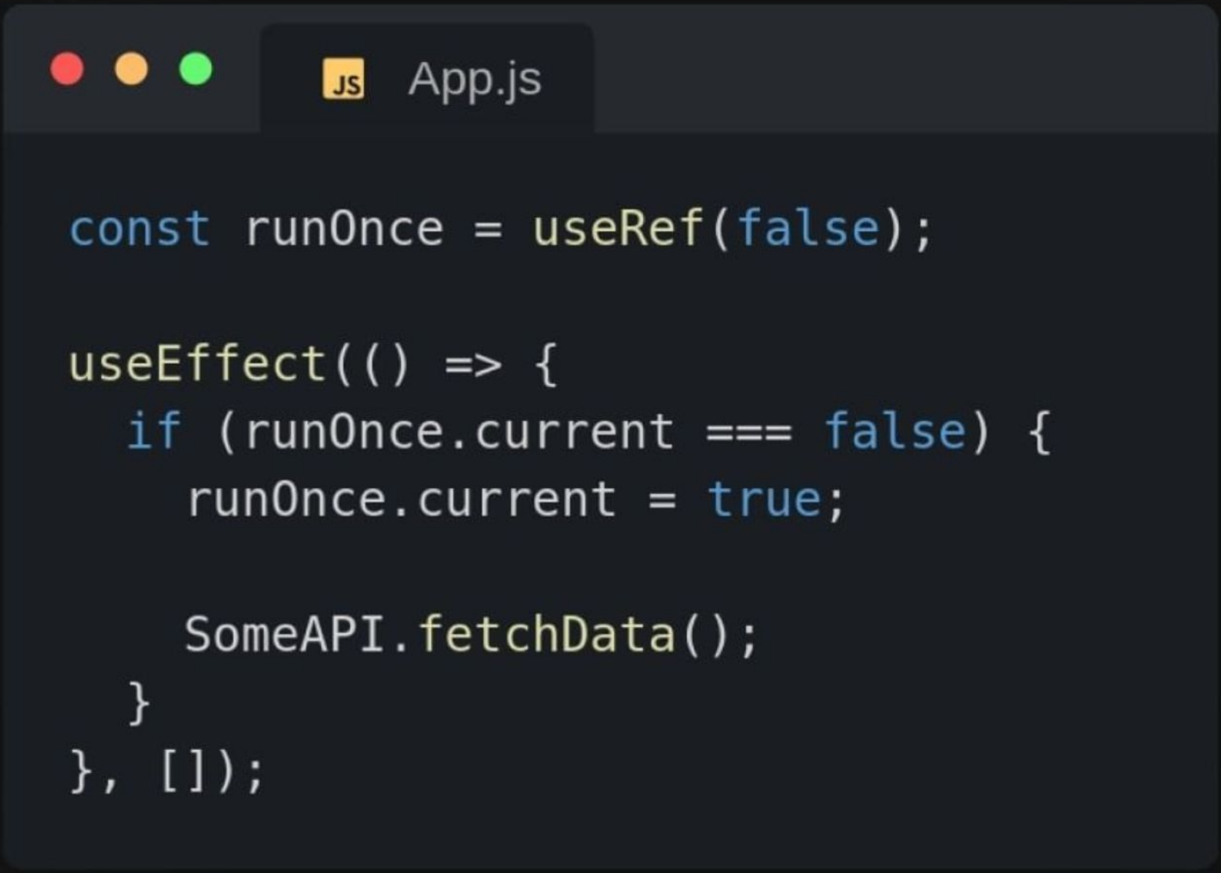


With the release of React 18, StrictMode gets an additional behavior to ensure it's compatible with reusable state. When StrictMode is enabled, React intentionally double-invokes effects (mount -> unmount -> mount) for newly mounted components. Like other strict mode behaviors, React will only do this for development builds

So now that useEffect runs twice duing mount you have to make some changes in your code

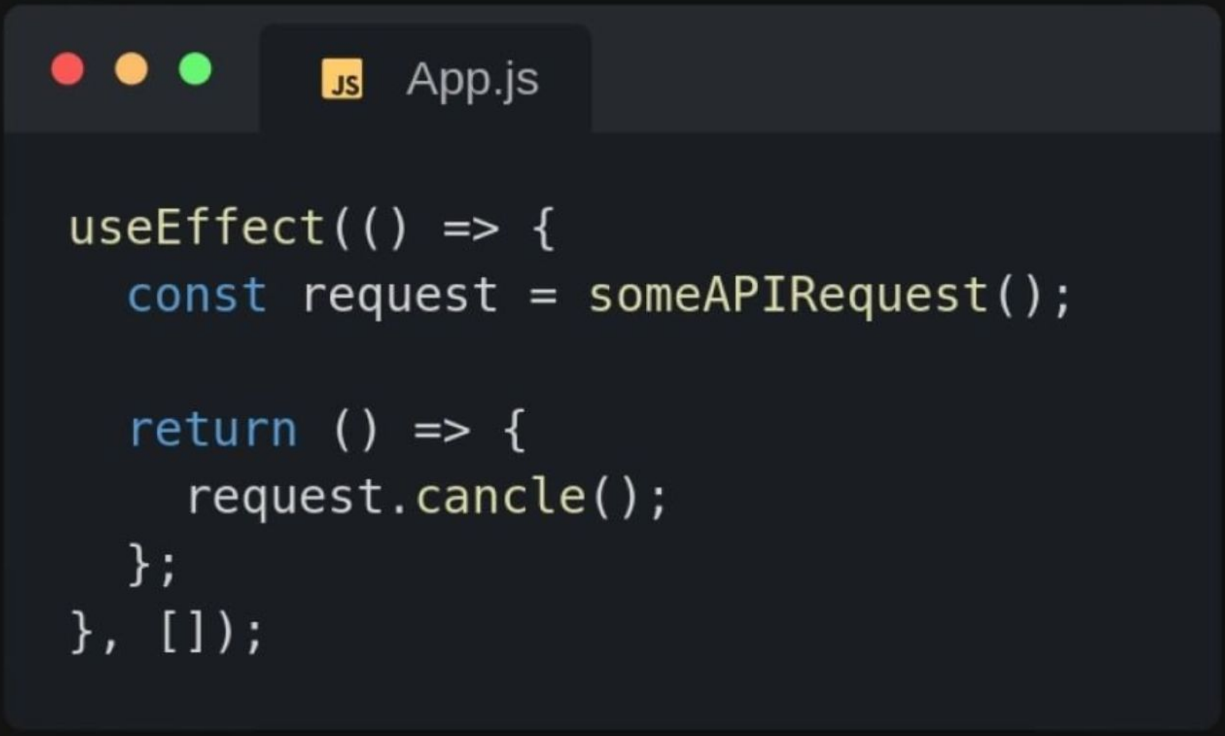
If the useEffect has dependencies does not require any changes

If you are using useEffect for on mount effect then you need to utilize useRef hook

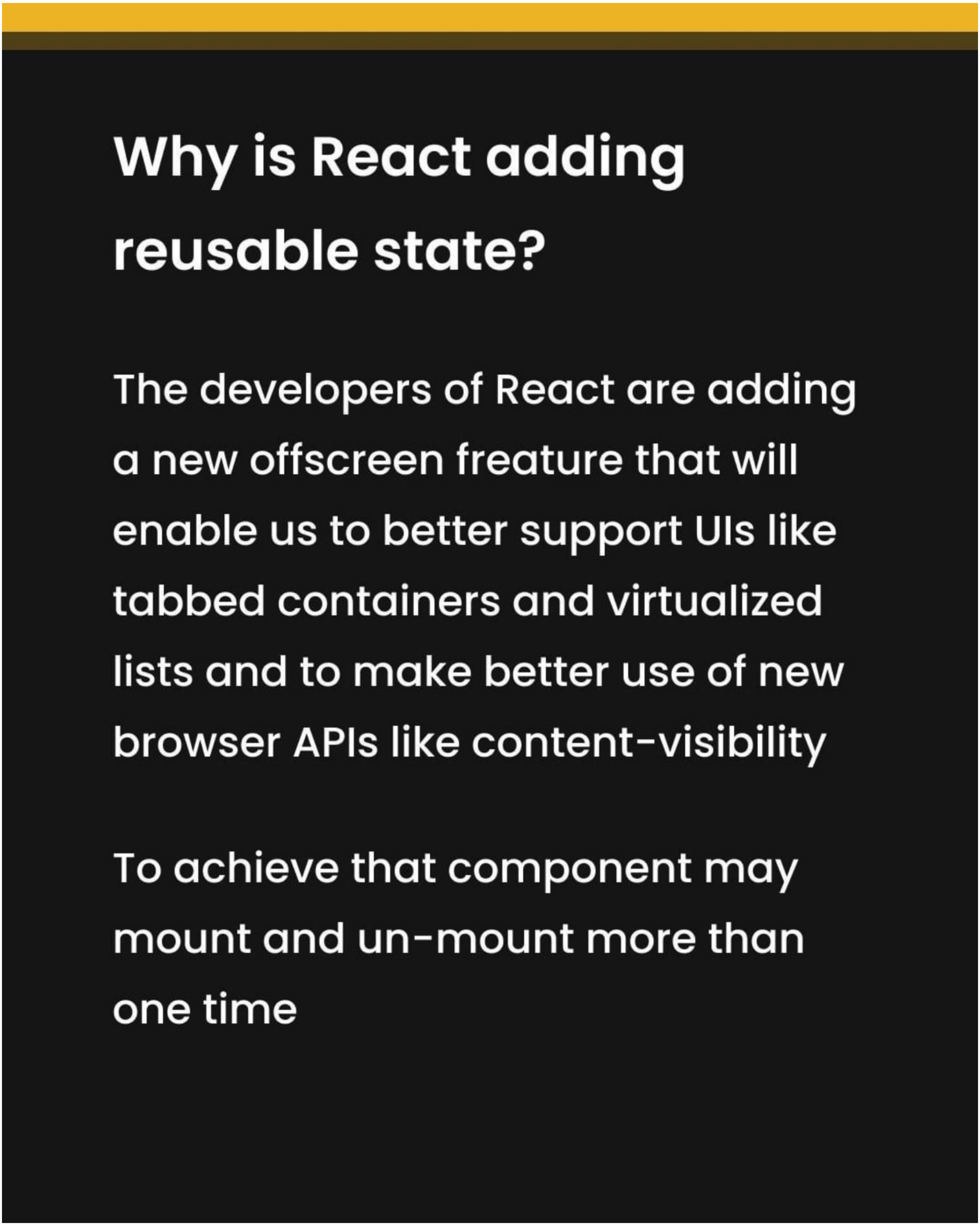


```
const runOnce = useRef(false);

useEffect(() => {
  if (runOnce.current === false) {
    runOnce.current = true;

    SomeAPI.fetchData();
  }
}, []);
```

If you are using useEffect for cleanup on un-mount then as long as the setup function is inside the same useEffect it should be fine



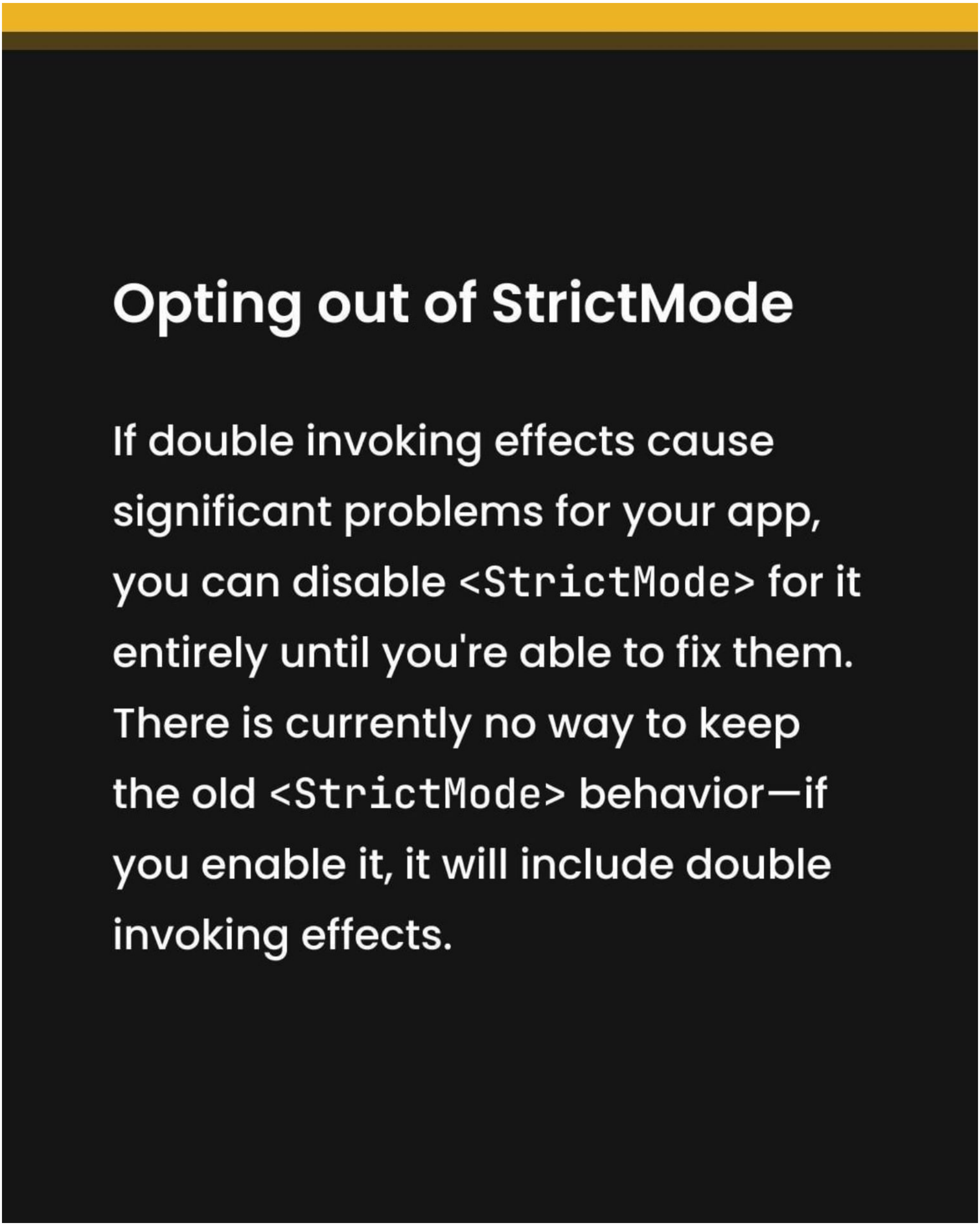
```
useEffect(() => {
  const request = someAPIRequest();

  return () => {
    request.cancle();
  };
}, []);
```



Why is React adding reusable state?

The developers of React are adding a new offscreen freature that will enable us to better support UIs like tabbed containers and virtualized lists and to make better use of new browser APIs like content-visibility

To achieve that component may mount and un-mount more than one time



Opting out of StrictMode

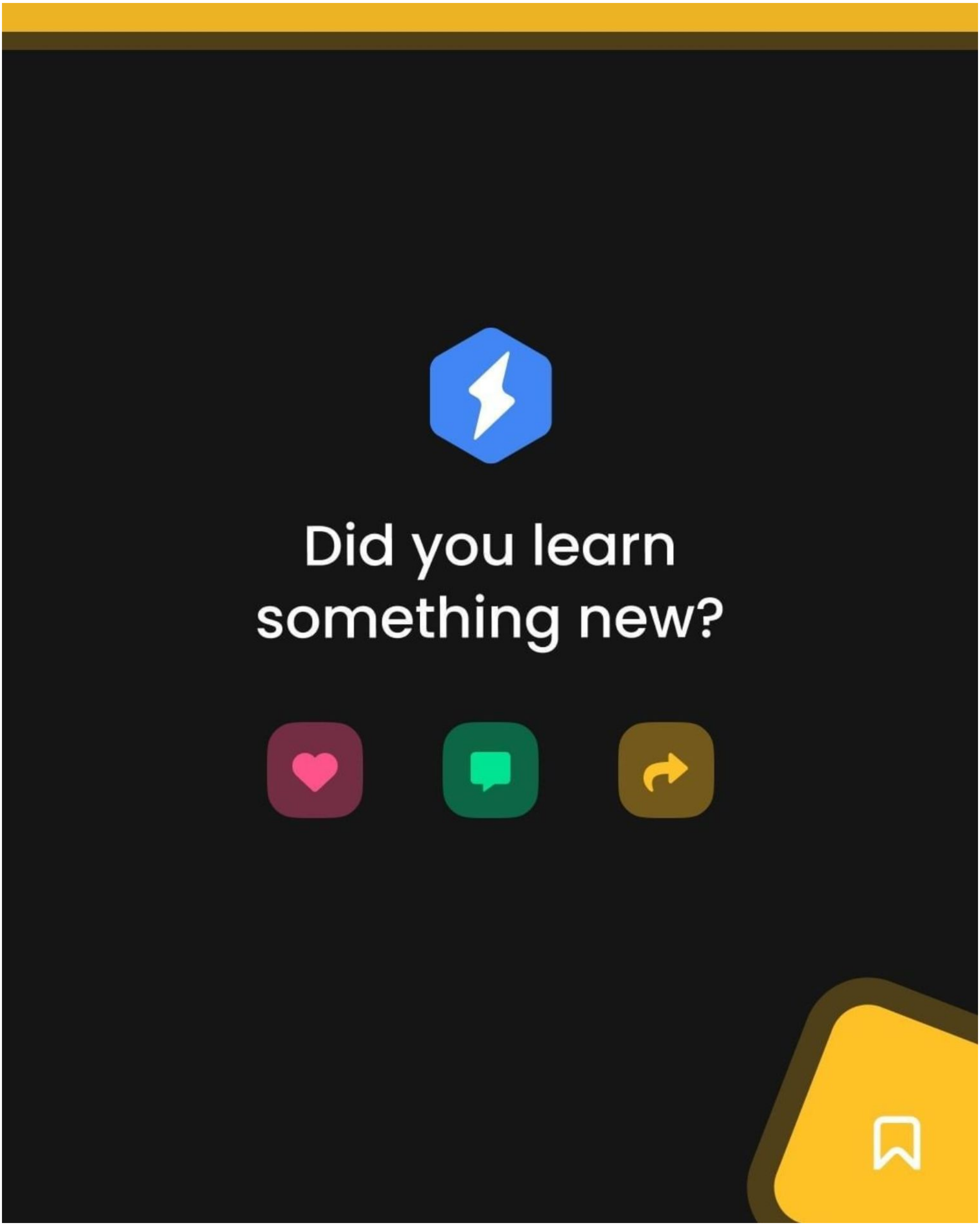
If double invoking effects cause significant problems for your app, you can disable `<StrictMode>` for it entirely until you're able to fix them. There is currently no way to keep the old `<StrictMode>` behavior—if you enable it, it will include double invoking effects.

THANK YOU

FOLLOW FOR MORE SUCH INTERESTING POST

LinkedIn

http://www.linkedin.com/in/depaksharma/



Did you learn something new?