**Dependency Array**

The useEffect dependency array in React controls when an effect should re-run by specifying dependencies.

**Purpose**

Informs React when to re-execute an effect based on changes to specified dependencies (state variables or props).

**Default Behavior**

Without a dependency array, the effect runs after every render, which is usually undesirable.

**Types of Dependency Arrays**

Multiple Dependencies

Effect runs on initial render and whenever any specified dependencies (e.g., X, Y, Z) change.

Empty Array

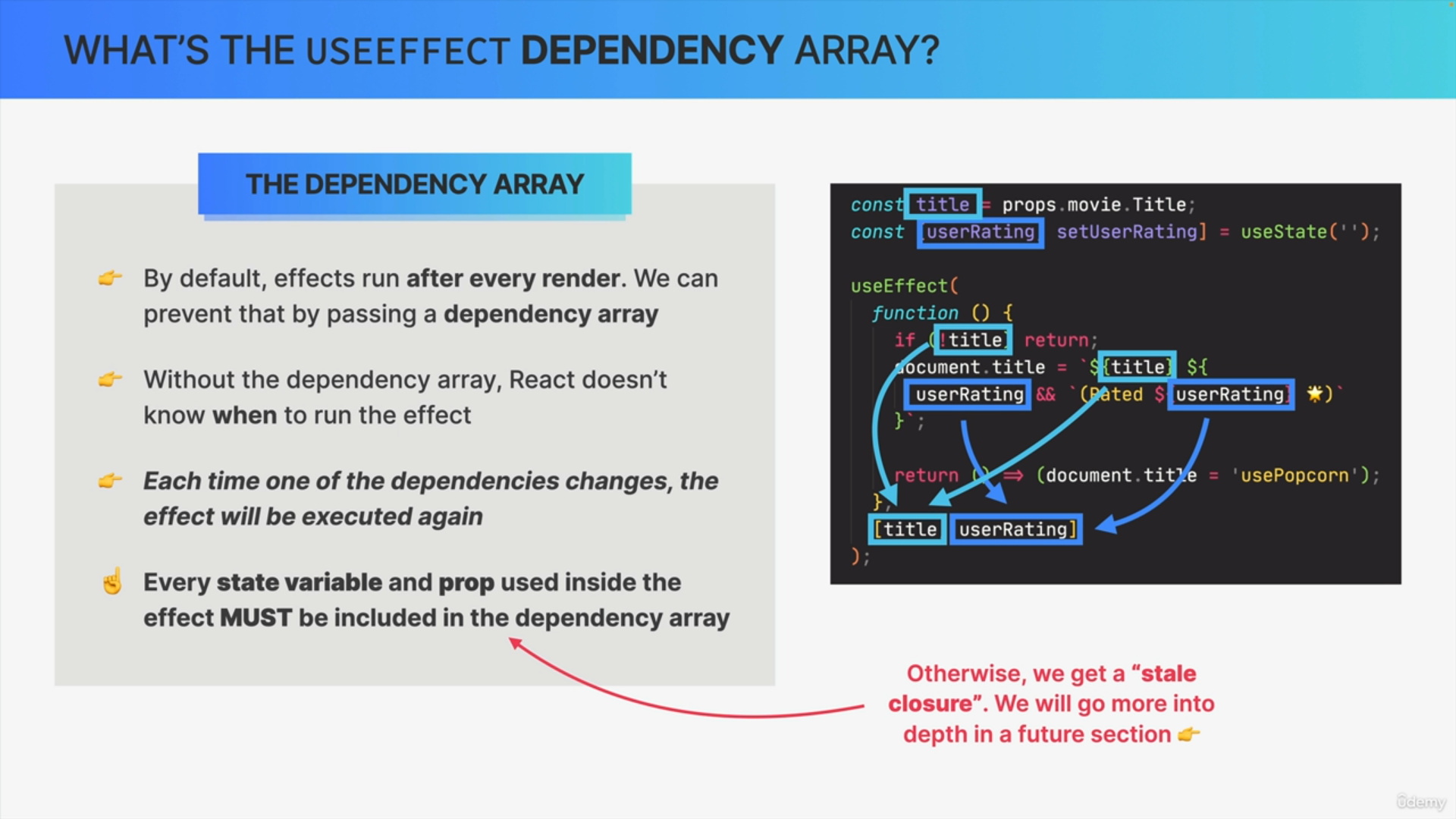
Effect runs only once, on mount, as it has no dependencies.

No Array

Effect runs after every render, effectively synchronizing with every state and prop in the component, which is not recommended.

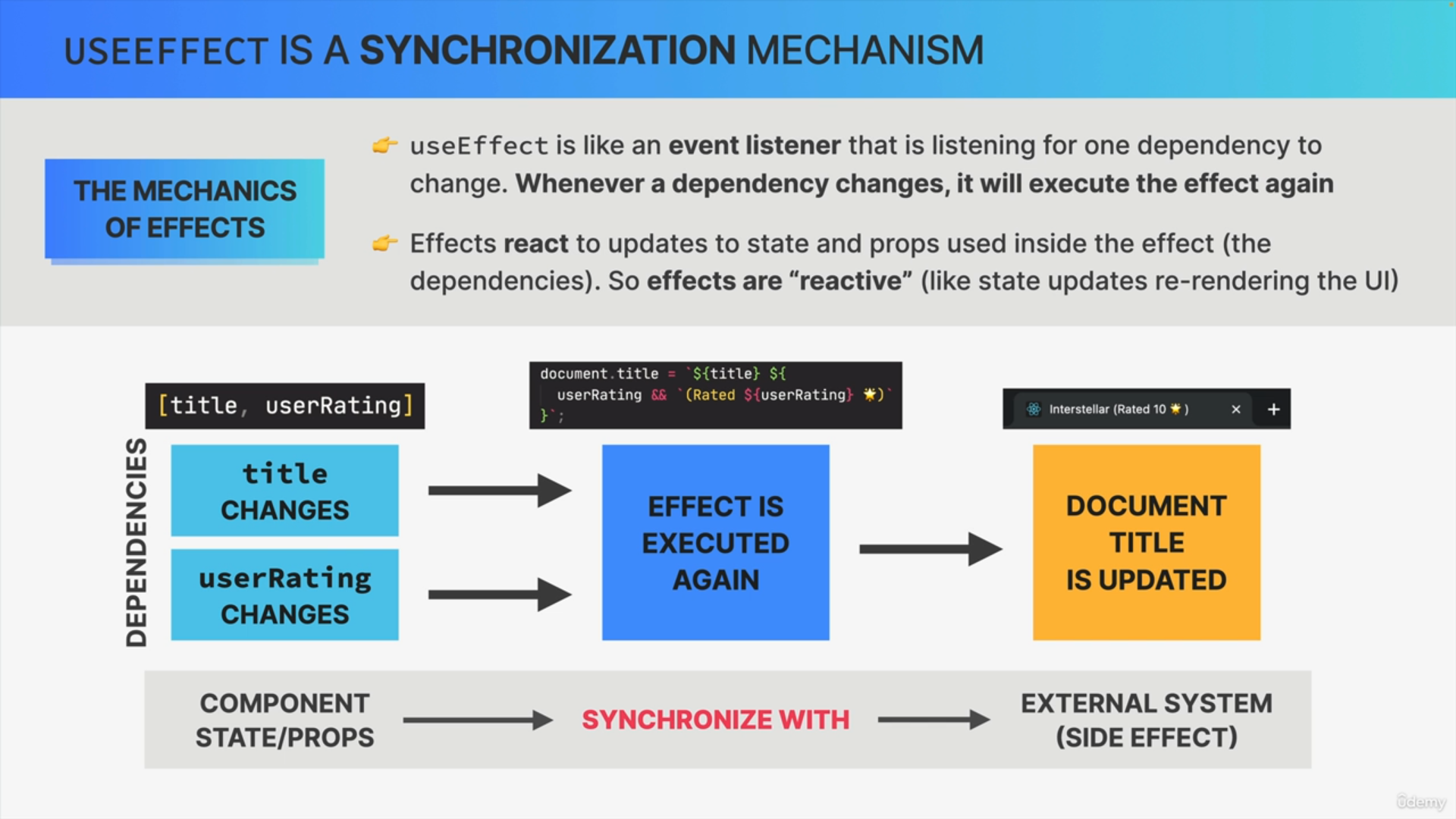
**Execution Timing**

* Effects run asynchronously after the component has been painted to the screen to avoid blocking the UI update process.
* If an effect sets state, it triggers an additional render for the UI to reflect the changes.



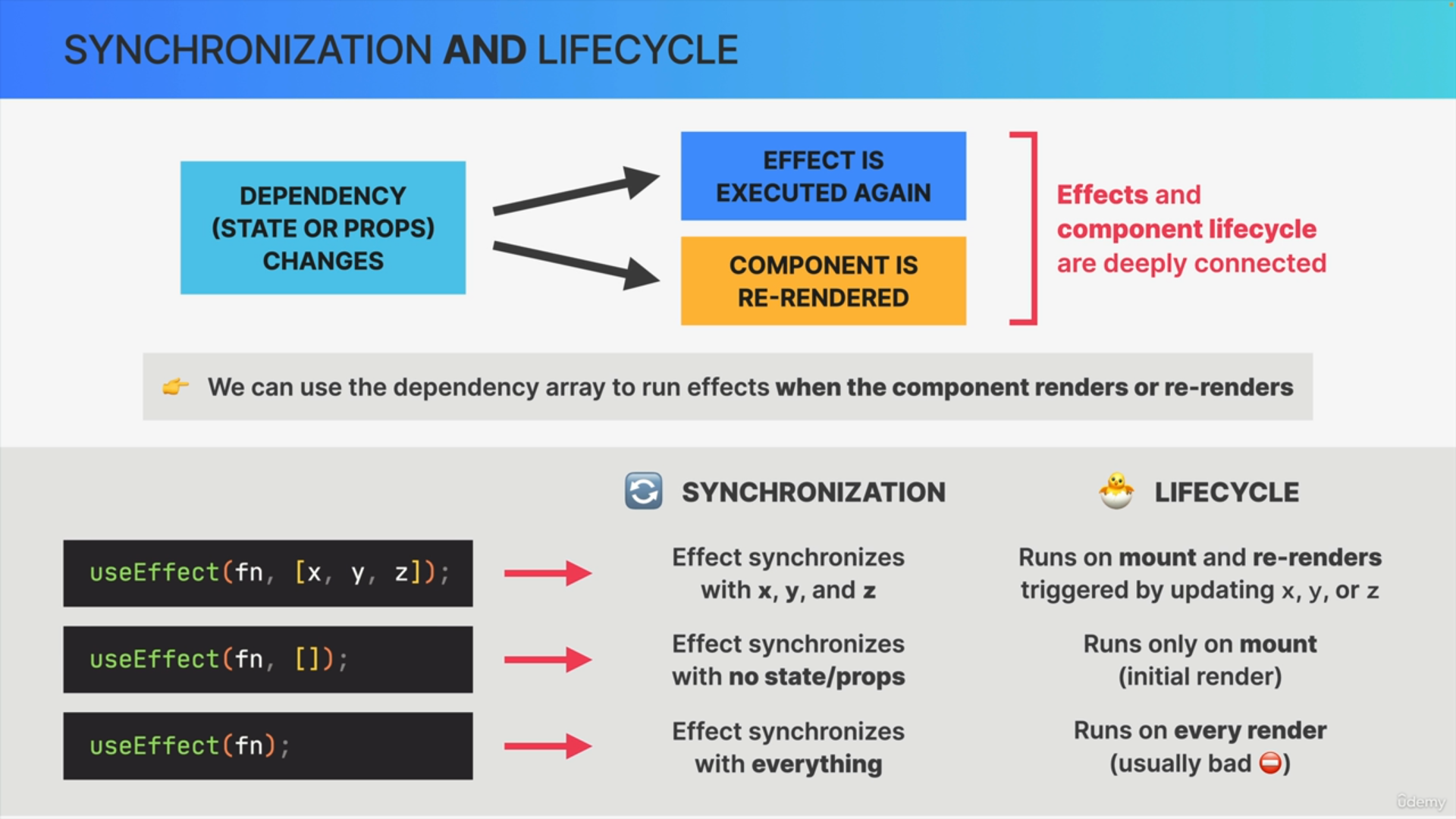
**Effects as Synchronization**

* Effects synchronize the component with external systems, such as updating the document title based on component state or props.
* This synchronization helps keep the component reactive to state and prop changes.



**Lifecycle Connection**

Dependency changes trigger re-renders, and effects are executed accordingly, aligning them with the component's lifecycle.



**Layout Effects**

* A special type of effect that runs before the browser paints the new screen, usually discouraged for general use.

Understanding the useEffect dependency array helps manage when effects are executed, ensuring efficient and synchronized updates in React components.

