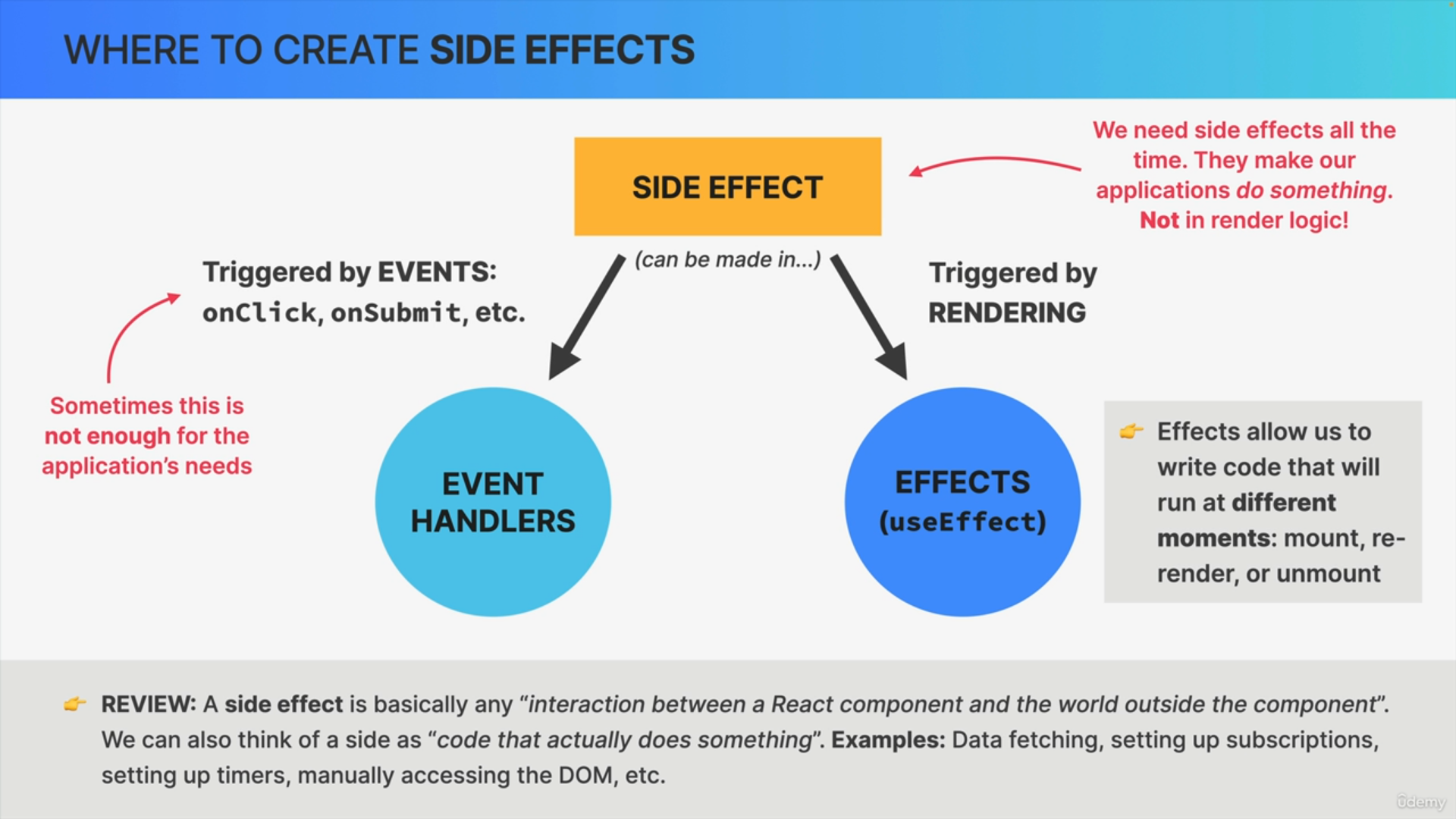
**Side Effects vs. Event Handlers**

* Event Handlers: Triggered by user actions or events (e.g., clicking a button). They are used to handle immediate interactions and should be preferred for side effects whenever possible.
* Effects (useEffect Hook): Used to run code automatically at different times during a component's lifecycle. This includes fetching data after the component mounts, not just in response to user actions.

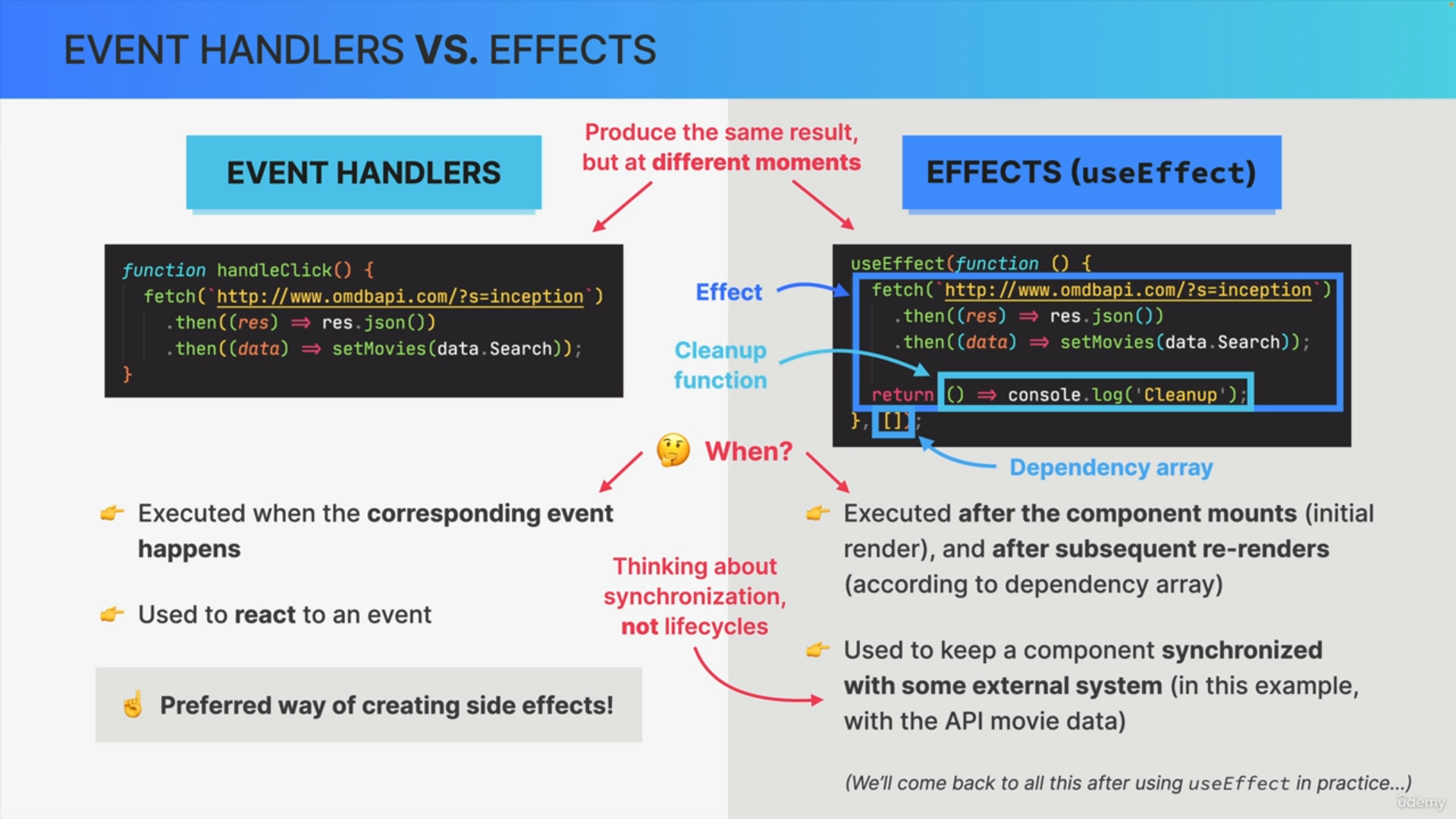


**UseEffect Hook Parts**

* Effect Code: The code that performs the side effect.
* Dependency Array: Specifies when the effect should re-run, based on changes to its dependencies.
* Cleanup Function: Optional function returned by the effect, used to clean up resources before the component re-renders or unmounts.

**Synchronization vs. Lifecycle**

* Effects help keep the component in sync with external systems (like APIs), while event handlers respond to user interface events.
* The goal is to use effects to synchronize with external systems and event handlers for direct user interactions.



In summary, use event handlers for user-triggered side effects and the useEffect hook for side effects related to the component lifecycle.