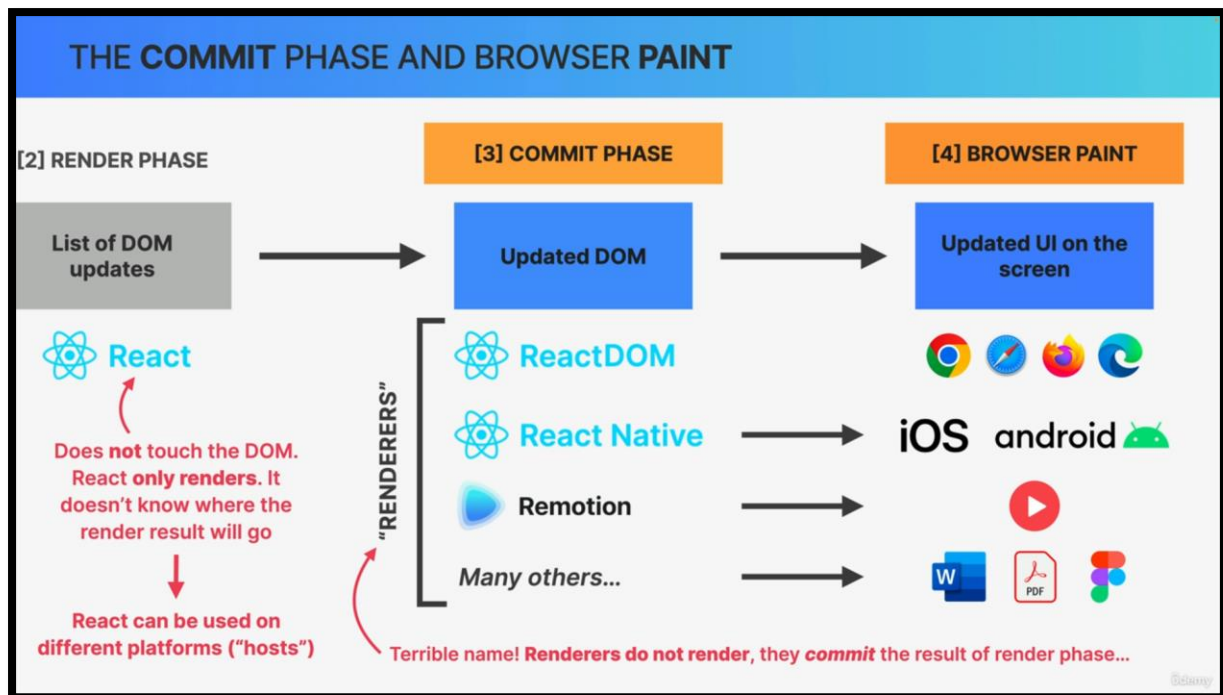


1. The commit phase is where React finally writes to the DOM. So, it inserts, deletes and updates DOM elements. React flushes updates to the DOM in this phase. React goes through the effects list that was created during rendering and applies them one by one to the actual DOM elements that were in the already existing DOM tree.
2. Writing to the DOM happens all in one go. So, the commit phase is synchronous unlike the rendering phase, which can be paused. So, committing cannot be interrupted. This is necessary so that the DOM never shows partial results which ensures that the UI always stays consistent.
3. In fact, that's the whole point of dividing the entire process into the render phase and the commit phase in the first place. It's so that rendering can be paused, resumed, and discarded and the results of all that rendering can then be flushed to the DOM in one go.
4. Once the commit phase is completed, the work in progress fiber tree becomes the current tree for the next render cycle. That's because, fiber trees are never discarded and never recreated from scratch. Instead, they are reused in order to save precious rendering time.

With that, we close up the commit phase.



The browser will then notice that the DOM has been changed and as a result, it will repaint the screen whenever it has some idle time. So, this is where these DOM updates are finally made visible to the user in the form of an updated user interface.

The browser paint phase is performed by whatever browser the user is using and the render phase is obviously performed by the React Library.

But what about the commit phase? We would think that it's also done by React, right?

But, actually, that's not true. It's actually a separate library that writes to the DOM and it's called React DOM.

React itself does never touch the DOM and it actually has no idea where the result of the render phase will actually be committed and painted.

So, React only does the render phase but not the commit phase.

RECAP: PUTTING IT ALL TOGETHER

