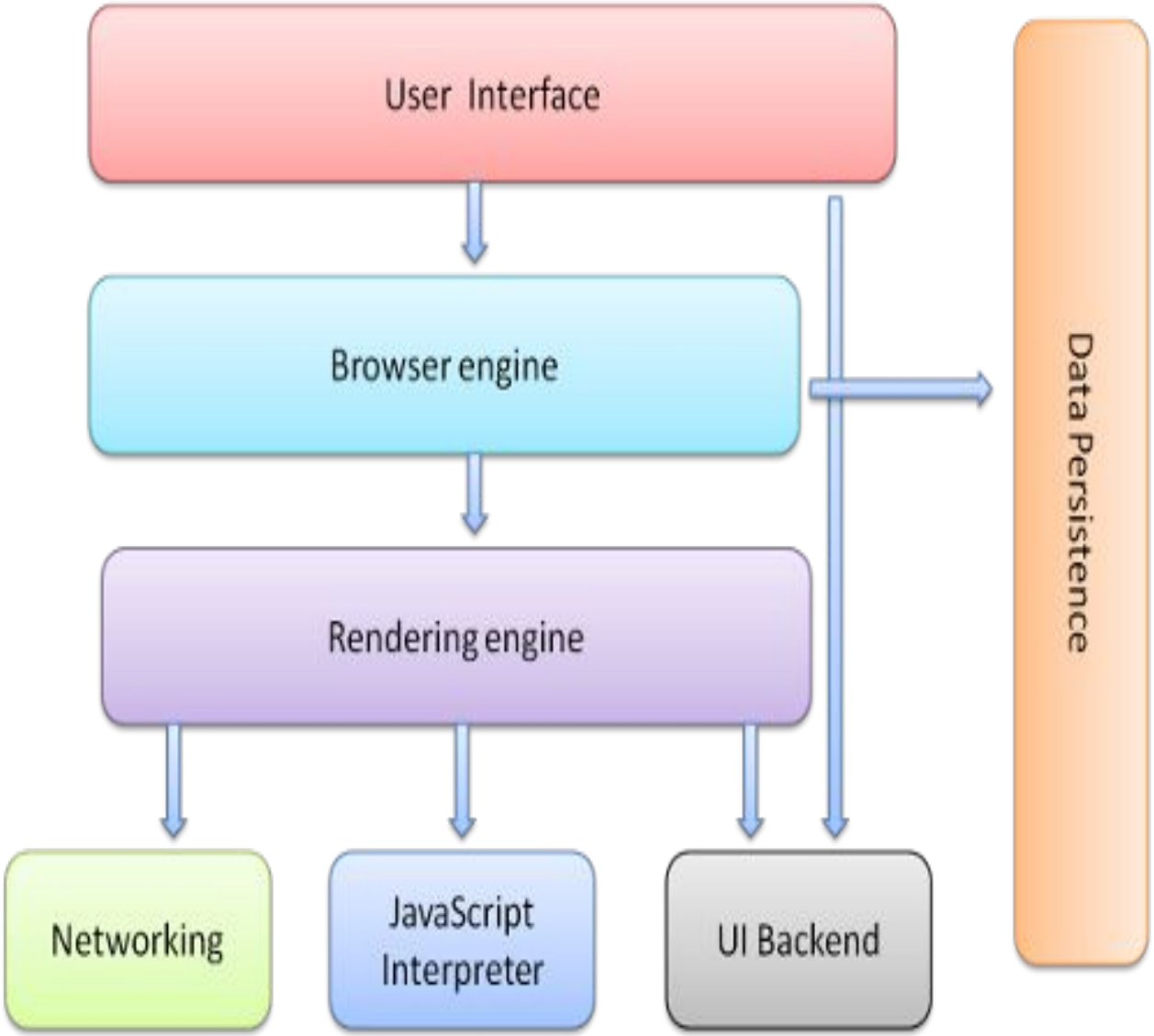
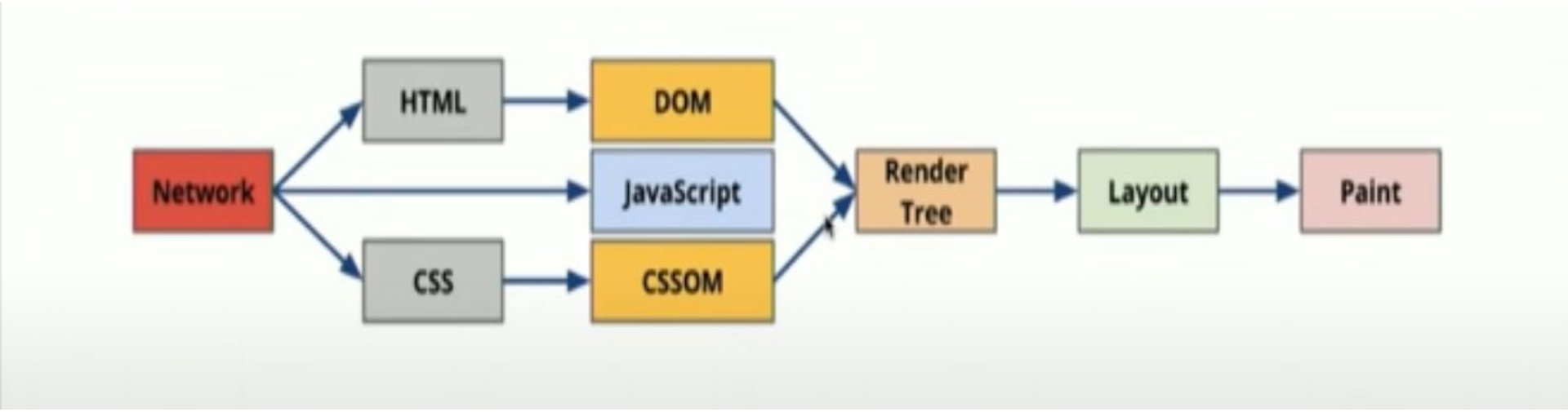
Browser Architecture



Critical Rendering Path



HTML Parsing

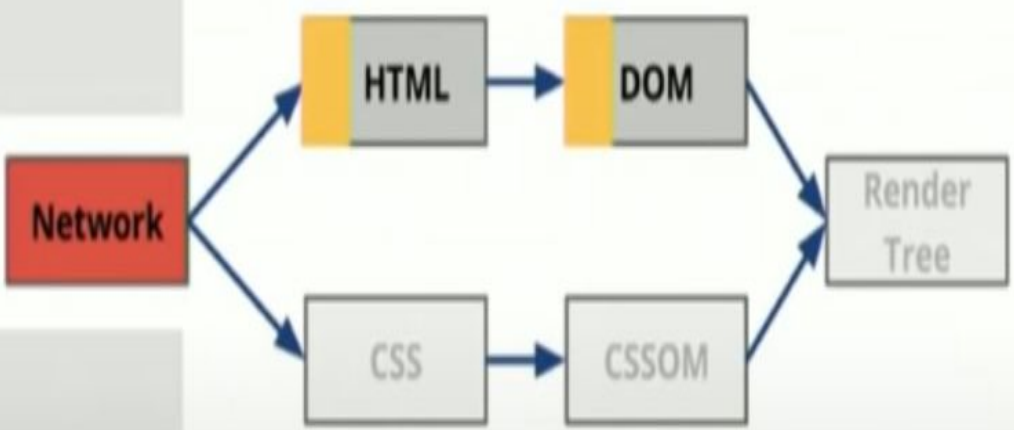
index.html

```
<!doctype html>
<meta charset=utf-8>
<title>Performance!</title>

<link href=styles.css rel=stylesheet />

<p>Hello <span>world!</span></p>
```

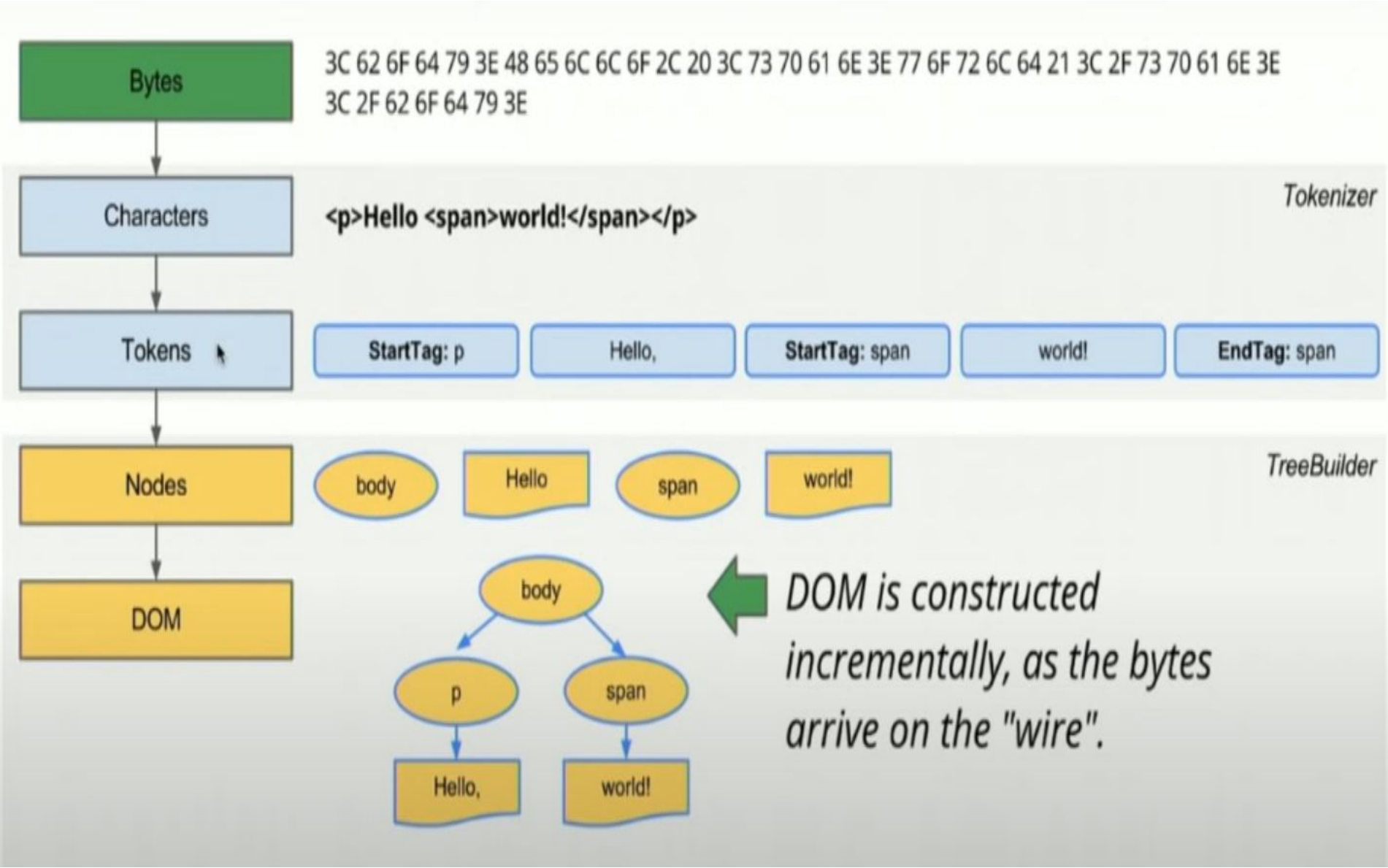
- first response packet with index.html bytes
- we have not discovered the CSS yet...



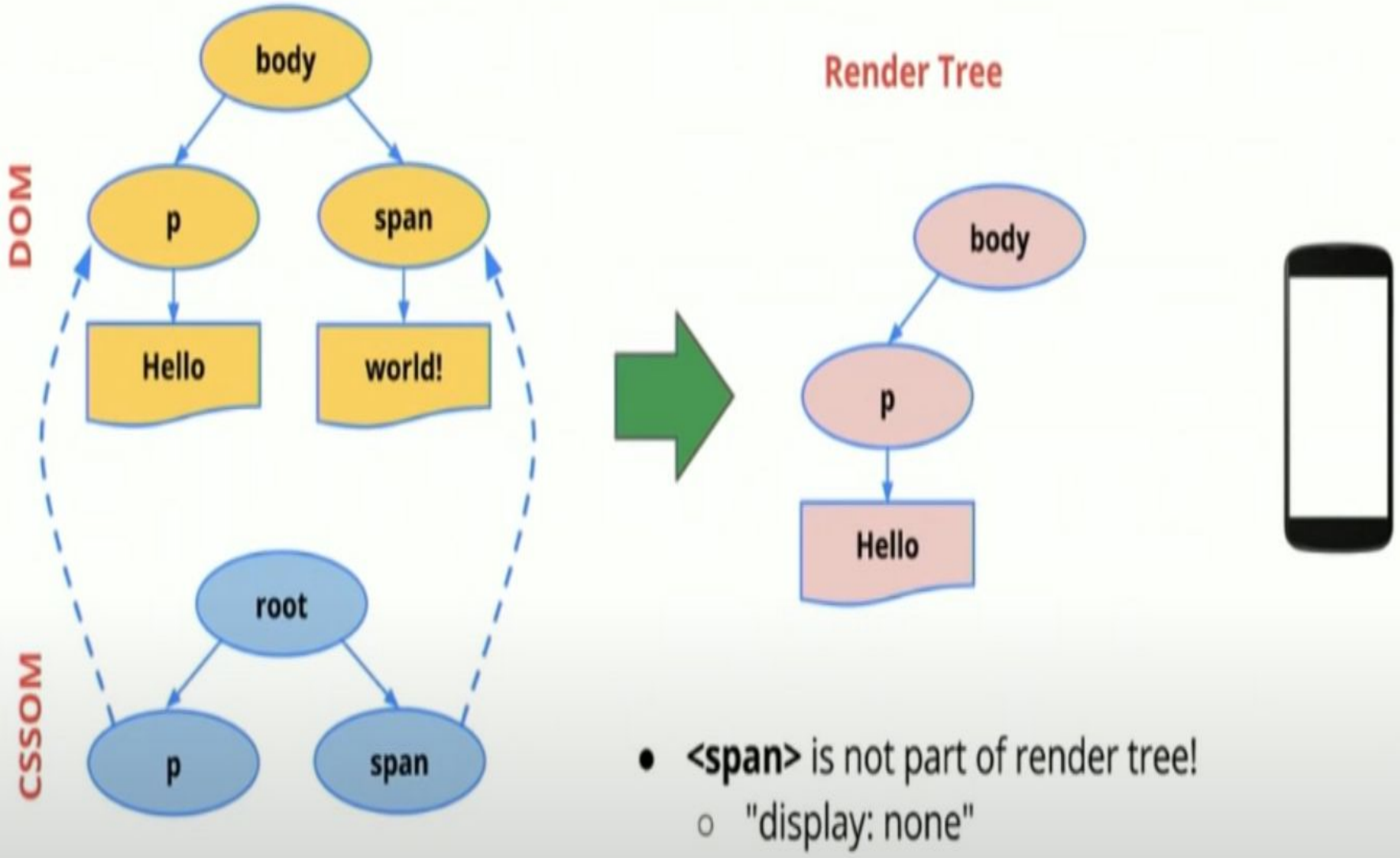
styles.css

```
p    { font-weight: bold; }
span { display: none; }
```

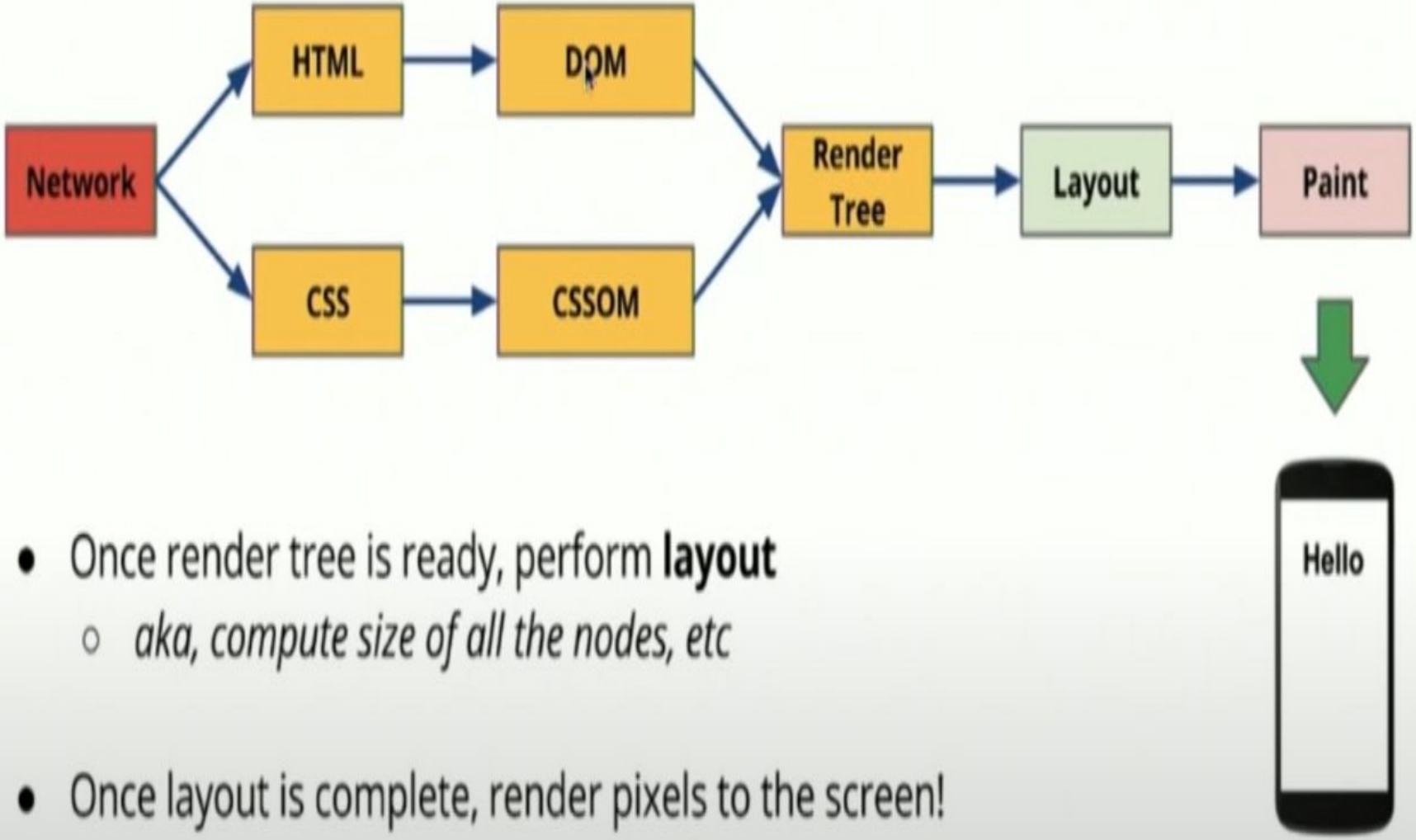
HTML Parsing



DOM + CSSOM = Render Tree



- **** is not part of render tree!
  - "display: none"



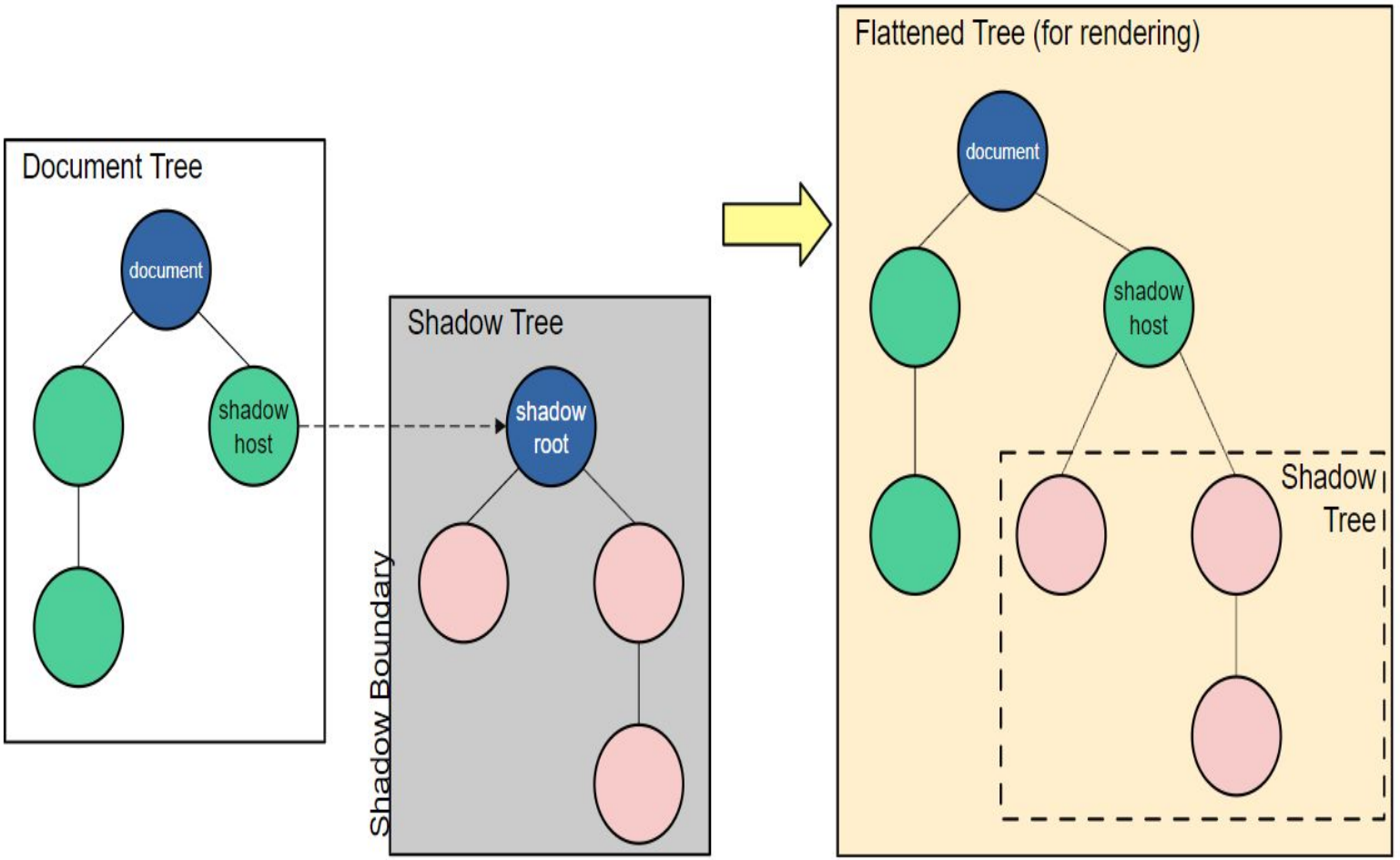
- Once render tree is ready, perform **layout**
  - *aka, compute size of all the nodes, etc*
- Once layout is complete, render pixels to the screen!

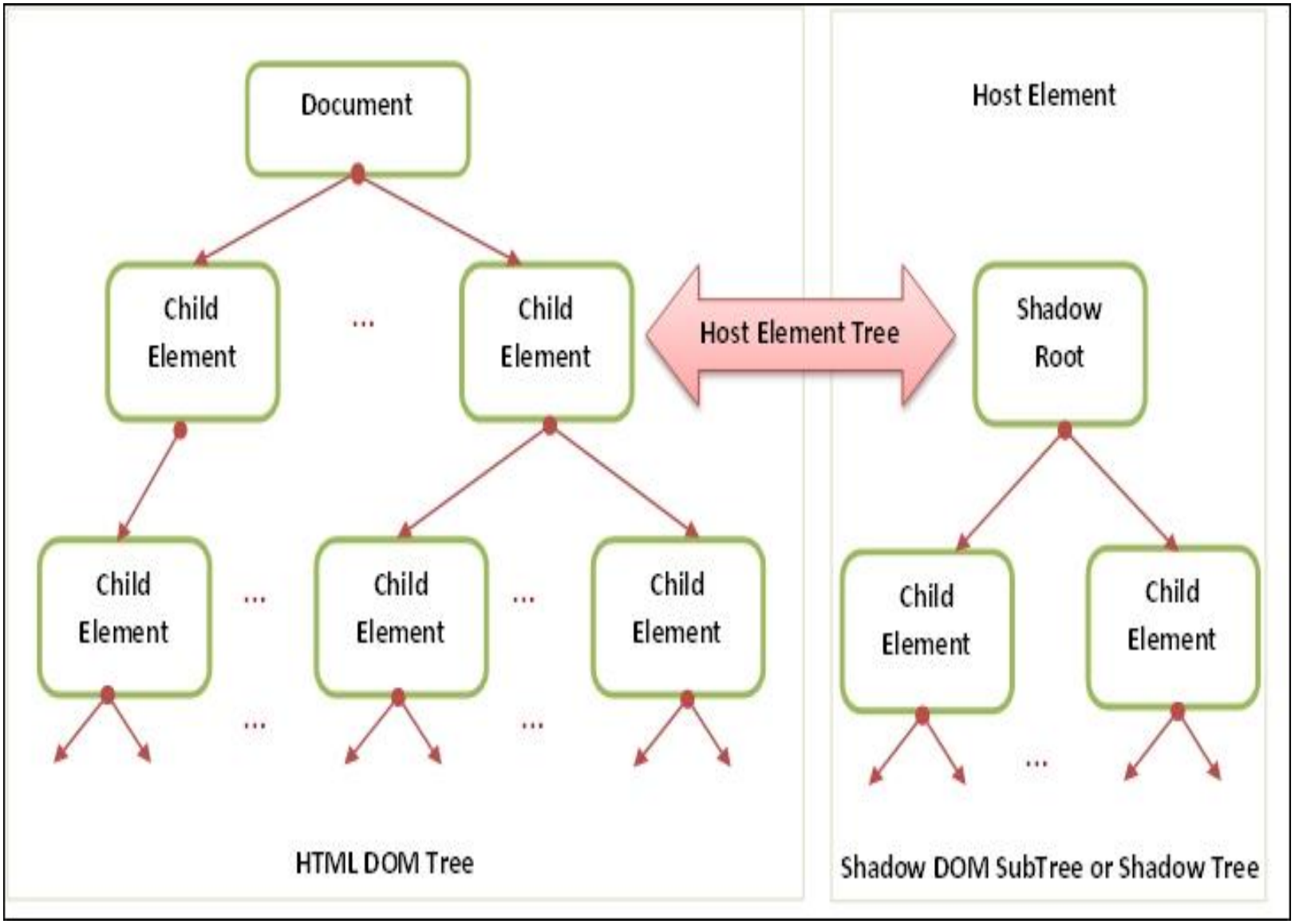
What is Shadow DOM?

- To keep the markup structure, style, and behavior hidden and separate from other code on the page so that different parts do not clash, and the code can be kept nice and clean.

- The Shadow DOM API is a key part of this, providing a way to attach a hidden separated DOM to an element

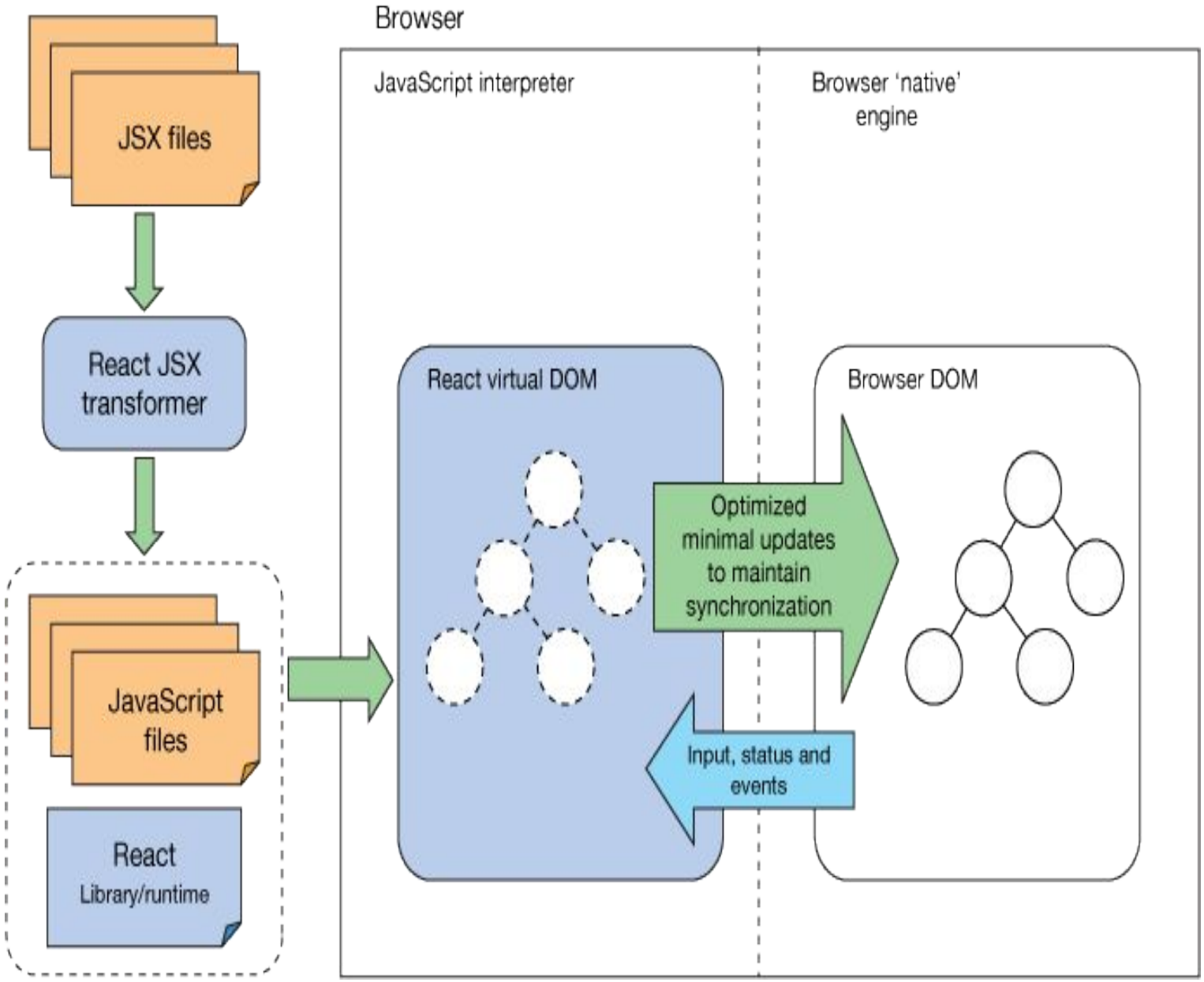
Shadow DOM allows hidden DOM trees to be attached to elements in the regular DOM tree — this shadow DOM tree starts with a shadow root, underneath which can be attached to any elements you want, in the same way as the normal DOM




Document
Host Element
Child Element
...
Child Element
Host Element Tree
Shadow Root
Child Element
...
Child Element
...
Child Element
Child Element
Child Element
...
...
...
...
HTML DOM Tree
Shadow DOM SubTree or Shadow Tree

What is Virtual DOM?

- The virtual DOM (VDOM) is a programming concept where an ideal, or “virtual”, representation of a UI is kept in memory and synced with the “real” DOM by a library such as ReactDOM. This process is called reconciliation.


Browser
JavaScript interpreter
Browser 'native' engine
JSX files
React JSX transformer
JavaScript files
React
Library/runtime
React virtual DOM
Browser DOM
Optimized minimal updates to maintain synchronization
Input, status and events