ReactJS.

The Component model

Topics

- **Component State.**
  - **Basis for dynamic, interactive UI.**
- **The Virtual DOM.**
- **Data Flow patterns.**
- **Lifecycle methods.**

Component DATA

- **Two sources of data for a component:**
  1. **Props - Passed in to a component; Immutable; an object (t**his.props**).**
  2. *State* **- Managed internally by the component; Mutable; an object (**this.state**)**
     - ***** The basis for dynamic and interactive Uis *****

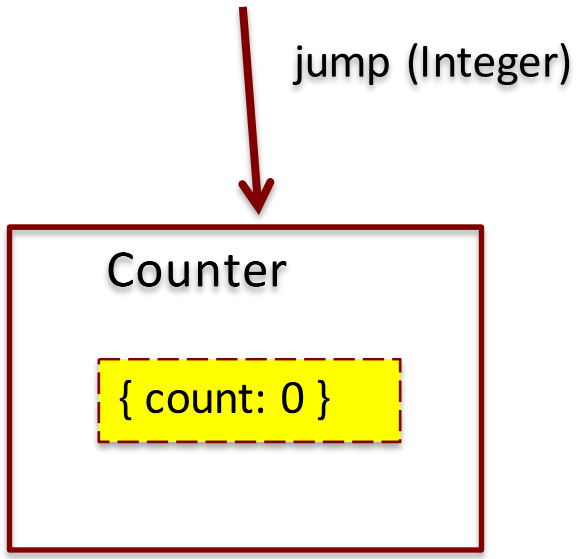
- **Props-related features:**
  - **Default values.**
  - **Type-checking.**

- **State-related features:**
  - **Initialization.**
  - **Mutation - the** setState() **method.**
    - **Performs a merge operation, not an overwrite.**
    - ***** Automatically causes component to re-render. *****

Component State - Example

- **The Counter component.**
- **Ref.** samples/06_state.js

- **Coding features:**
  1. **Custom function/method, e.g. incrementCount().**
  2. **Static class property, e.g. defaultProps.**
  3. **Class instance property, e,g. state.**



React’s event system.

- **Cross-browser support.**
- **Event handlers receive** SyntheticEvent **– a cross-browser wrapper for the browser's native event.**
- **Event naming convention slightly different from native:**

React	Native
onClick	onclick
onChange	onchange
onSubmit	onsubmit

- See https://reactjs.org/docs/events.html for full details,

Automatic Re-rendering

- **EX.: The Counter component.**

  User clicks ‘increment’ button

  → onClick event handler (incrementCounter) executed

  → state is changed (setState())

  → render() method executed

Modifying the DOM

- **DOM – an internal data structure; mirrors the state of the UI; always in sync.**
- **Traditional performance best practice:**
  1. **Minimize access to the DOM.**
  2. **Avoid expensive DOM operations.**
  3. **Update elements offline, then reinsert into the DOM.**
  4. **Avoid changing layouts in Javascript.**

- **Should the developer be responsible for low-level DOM optimization? Probably not.**
  – **React provides a** *Virtual DOM* **to shield developer from these concerns.**

The Virtual DOM

- **Consequence: Re-render everything on every update.**
  - **Sounds expensive!**

- **How?**
  1. **Create a lightweight, efficient form of the DOM – the Virtual DOM.**
  2. **Perform *diff* operation between it and the previous (virtual) UI state.**
  3. **Compute the minimal set of changes to apply to (real) DOM.**
  4. **Batch execute all updates to real DOM.**

- **Benefits:**
  - **a) Clean – Clean, descriptive programming model.**
  - **b) Fast - Optimized DOM updates and reflows.**

Automatic Re-rendering (detail)

- **EX.: The Counter component.**

User clicks ‘increment’ button

→ onClick event handler (incrementCounter) executed

→ state is changed (setState())

→ render() method executed

→ The Virtual DOM has changed

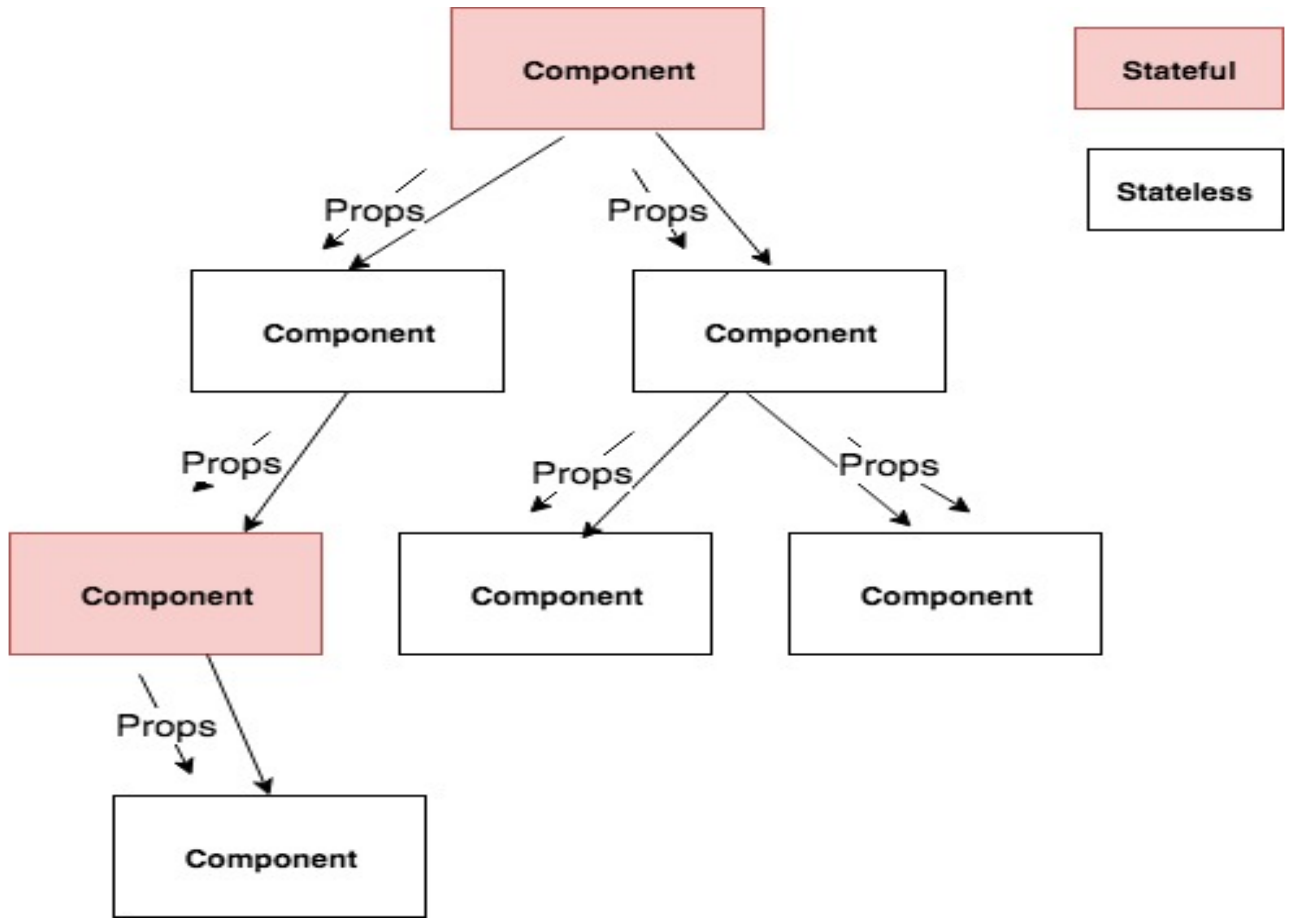
→ React diffs the changes (between the current and previous Virtual DOM)

→ React batch updates the Real DOM

Topics

- **Component State.** ✓
- **The Virtual DOM.** ✓
- **Data Flow patterns.**
- **Lifecycle methods.**

Unidirectional data flow



Unidirectional data flow

- **In a React app, data flows uni-directionally ONLY.**
  - **Most other SPA frameworks use two-way data binding.**

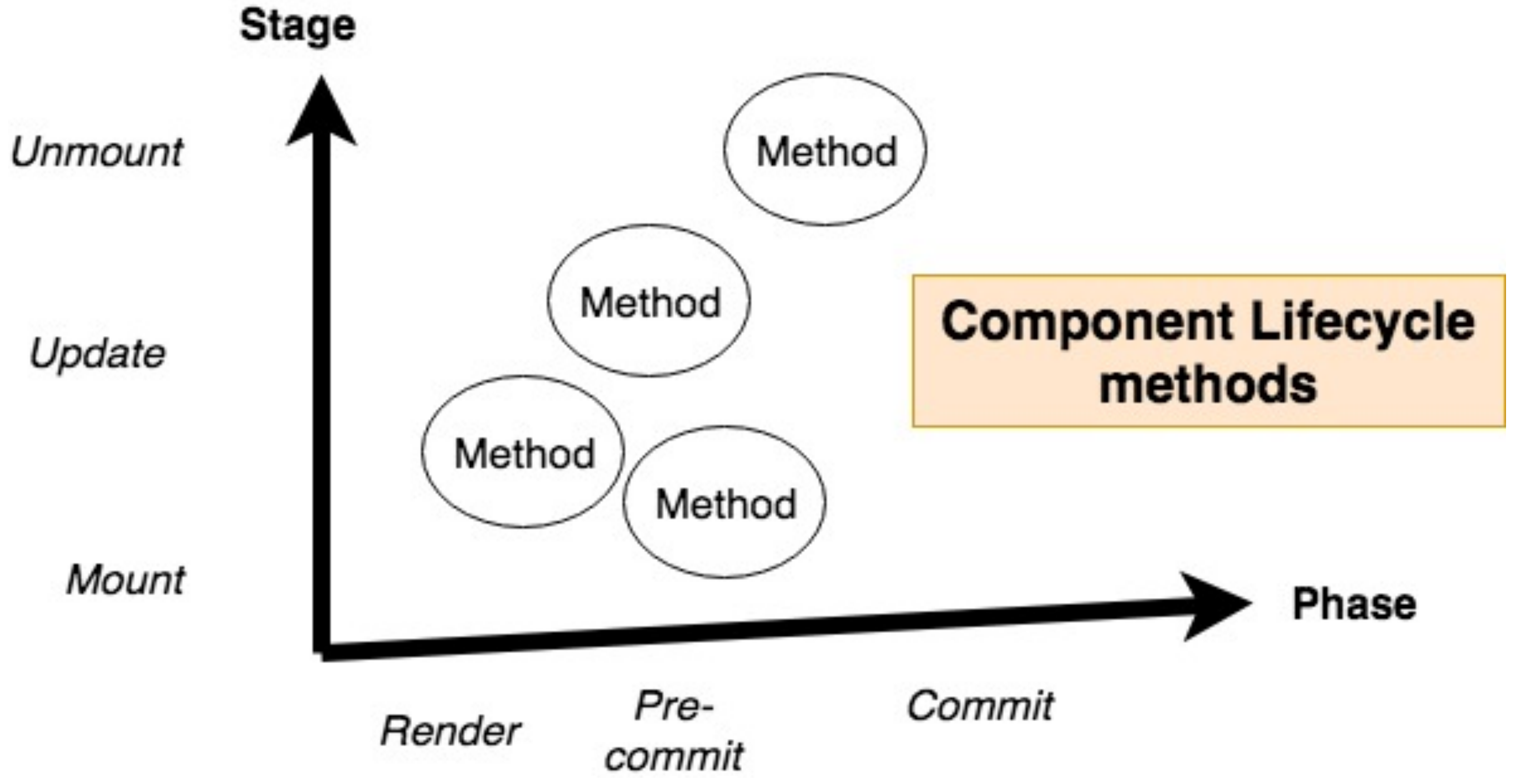
- **In a multi-component app, a common pattern is:**
  - **A small subset (maybe only 1) of components will be statefull – the rest are stateless.**

- **Statefull components:**
  - **Call *setState*() to update its state.**
  - **Re-renders itself.**
  - **Pass updated (and unchanged) props to subordinate components.**
  - *React guarantees subordinate components are re-rendered with updated prop values.*

Topics

- **Component State.** ✓
- **The Virtual DOM.** ✓
- **Data Flow patterns.** ✓
- **Lifecycle methods.**

Component *Lifecycle methods*



Component *Lifecycle methods*

- **Methods invoked by React at specific times in a component's lifecycle (Most are optional).**

- **Lifecycle stages:**
  1. **Mounting (Initialization).**
  2. **Update.**
     a) **New props.**
     b) **setState();.**
     c) **forceUpdate.**
  3. **Un-mounting.**

- **Phases:**
  - **Render phase.**
  - **Pre-commit phase (Pre DOM update).**
  - **Commit phase (Post DOM update).**

The Lifecycle methods

1. shouldComponentUpdate**() – returns boolean – can cause a component to skip re-rendering.**

2. getDerivedStateFromProps**() - when a component state object is computed from its prop values.**

3. componentDidUpdate**() – executed after a rendering has updated real DOM, e.g. perform real DOM manipulation, set up external subscription, cause side-effect.**

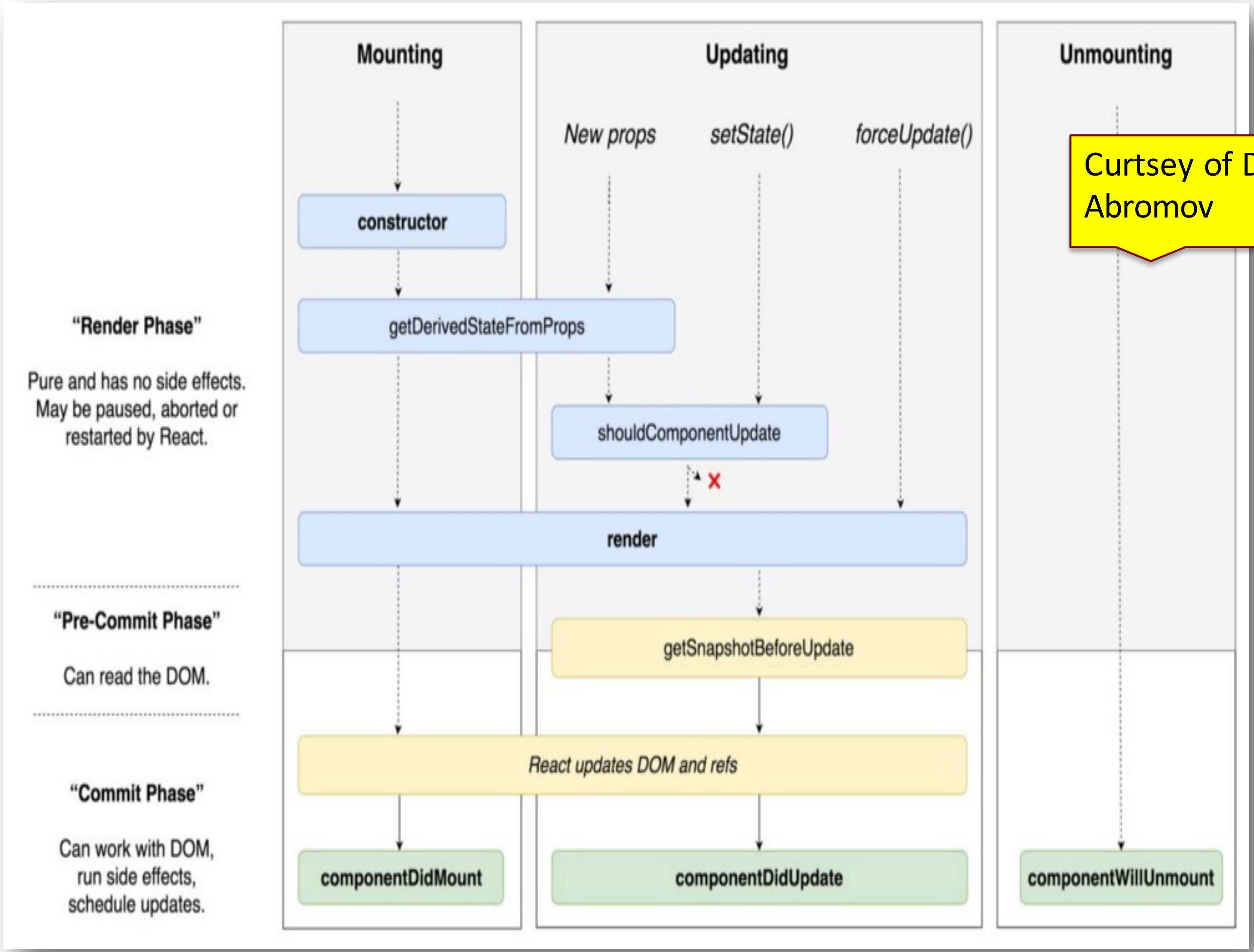
4. componentDidMount**() – executed after component has mounted (see later)**

5. **componentWillUnmount(); executed before a component is about to unmount; Perform cleanup operations, e.g. remove external subscription.**

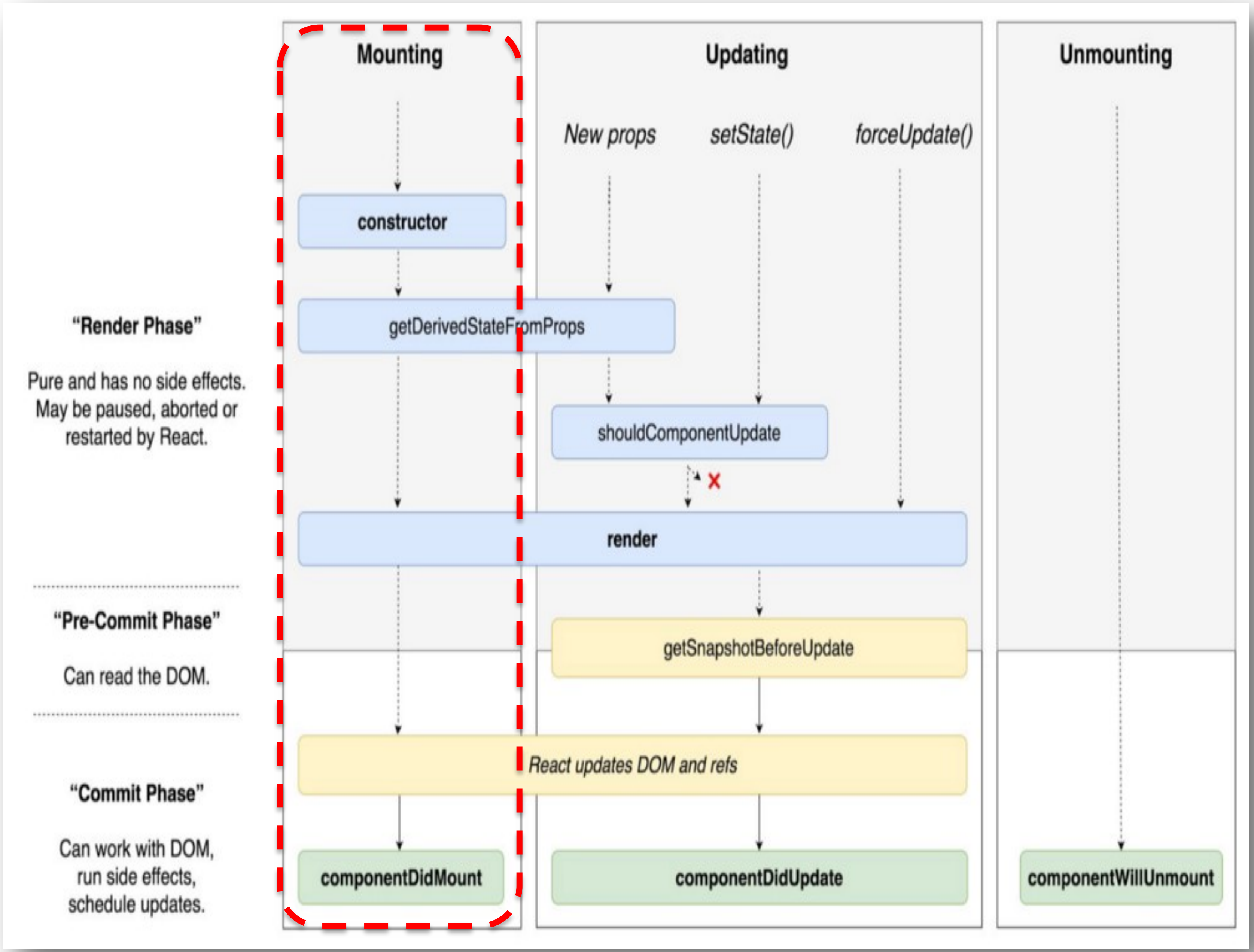
The Lifecycle methods

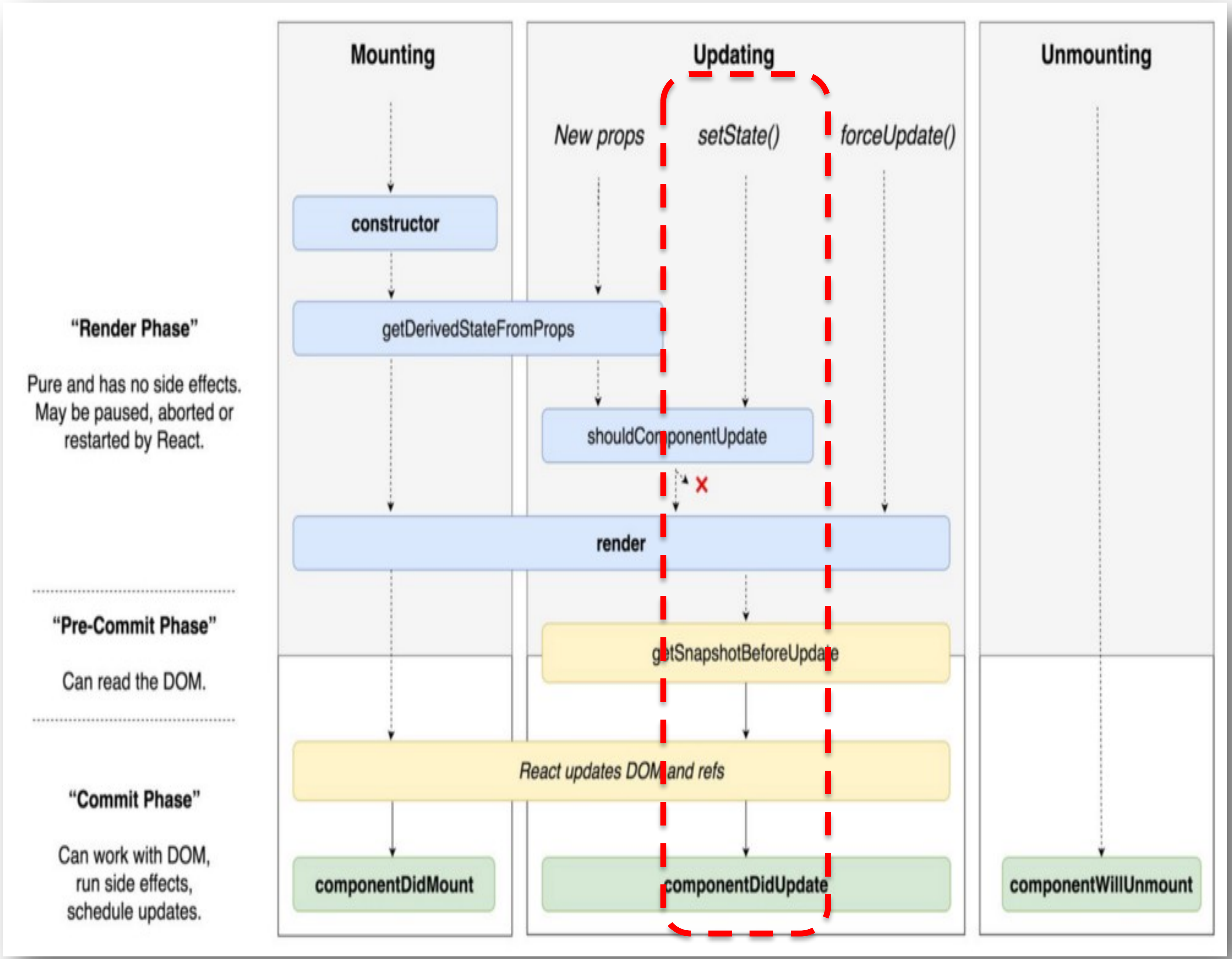
- *shouldComponentUpdate()* ✻✻
- *getDerivedStateFromProps().*
- *render().* ✻✻
- *componentDidUpdate()*
- *componentDidMount()* ✻✻
- *componentWillUnmount***()**

✻✻. used most frequently

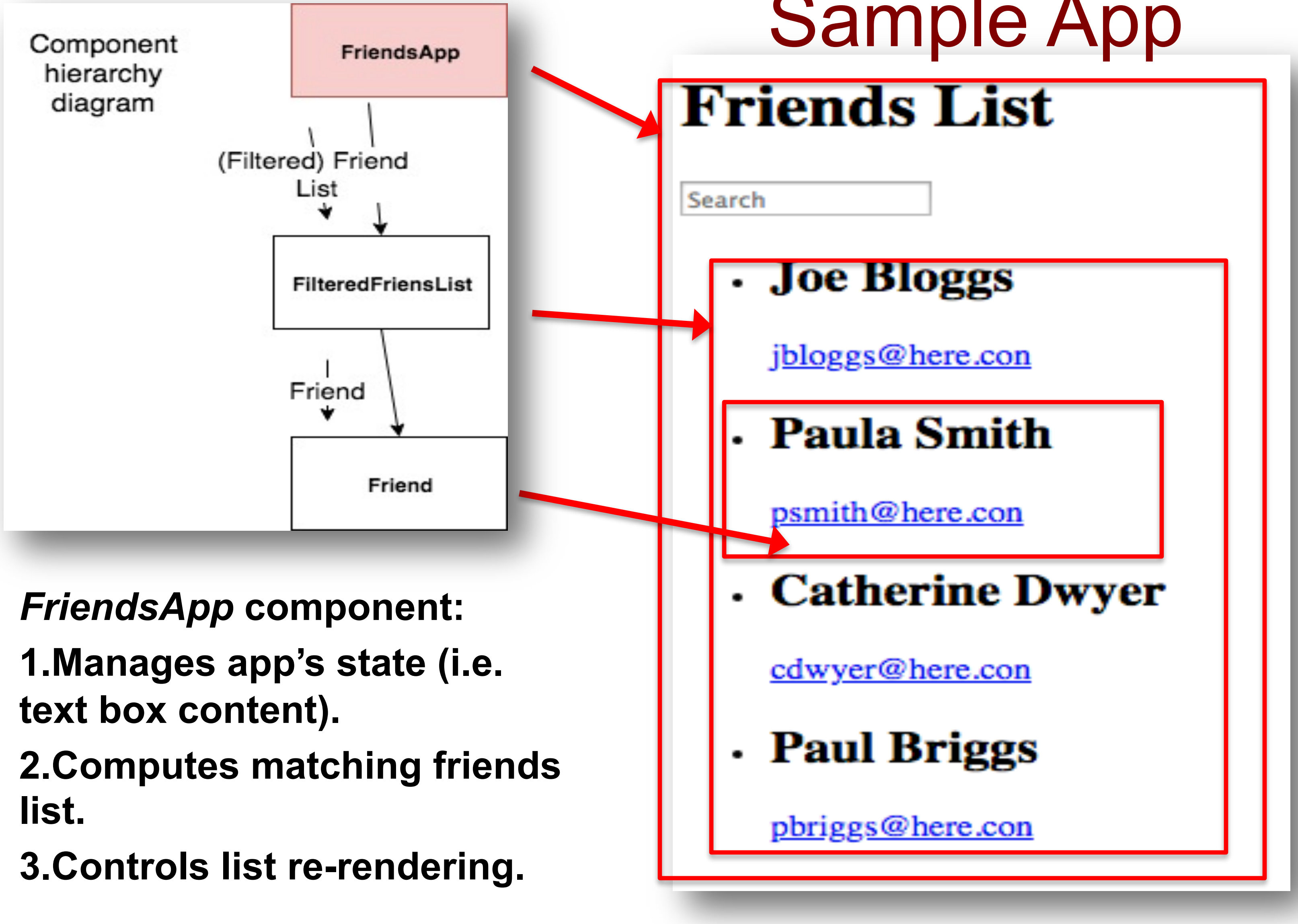


Curtsey of Dan Abromov


Mounting
Updating
Unmounting
New props
setState()
forceUpdate()
constructor
getDerivedStateFromProps
shouldComponentUpdate
render
getSnapshotBeforeUpdate
React updates DOM and refs
componentDidMount
componentDidUpdate
componentWillUnmount
"Render Phase"
Pure and has no side effects. May be paused, aborted or restarted by React.
"Pre-Commit Phase"
Can read the DOM.
"Commit Phase"
Can work with DOM, run side effects, schedule updates.


Mounting
Updating
Unmounting
New props
setState()
forceUpdate()
constructor
getDerivedStateFromProps
shouldComponentUpdate
render
getSnapshotBeforeUpdate
React updates DOM and refs
componentDidMount
componentDidUpdate
componentWillUnmount
"Render Phase"
Pure and has no side effects. May be paused, aborted or restarted by React.
"Pre-Commit Phase"
Can read the DOM.
"Commit Phase"
Can work with DOM, run side effects, schedule updates.

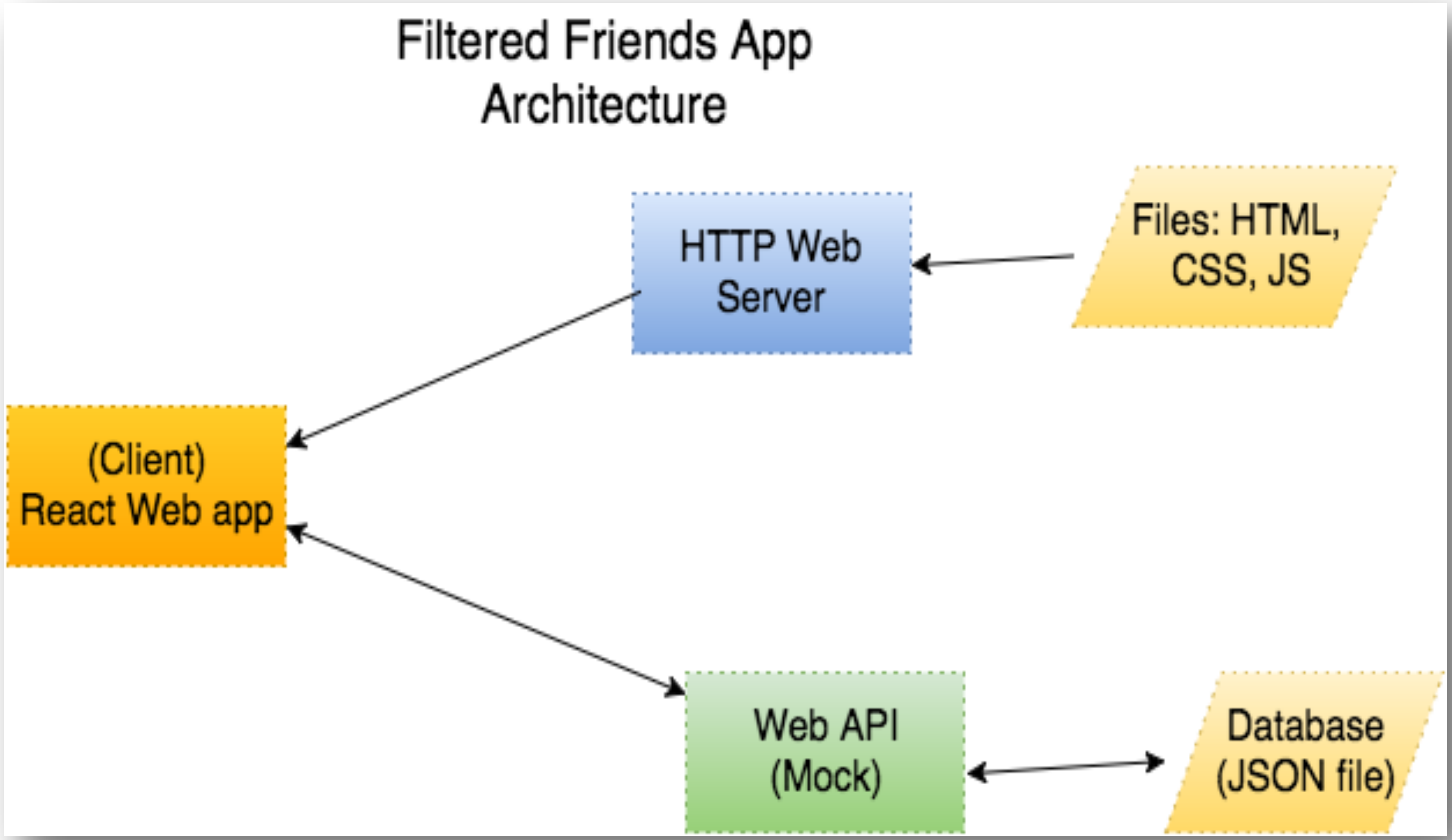
Sample App



***FriendsApp* component:**

1. **Manages app's state (i.e. text box content).**
2. **Computes matching friends list.**
3. **Controls list re-rendering.**

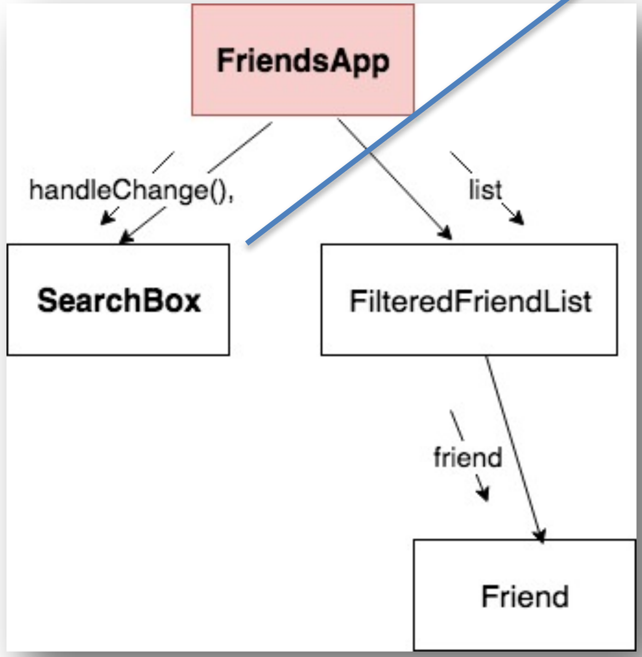
Sample App – Architecture..

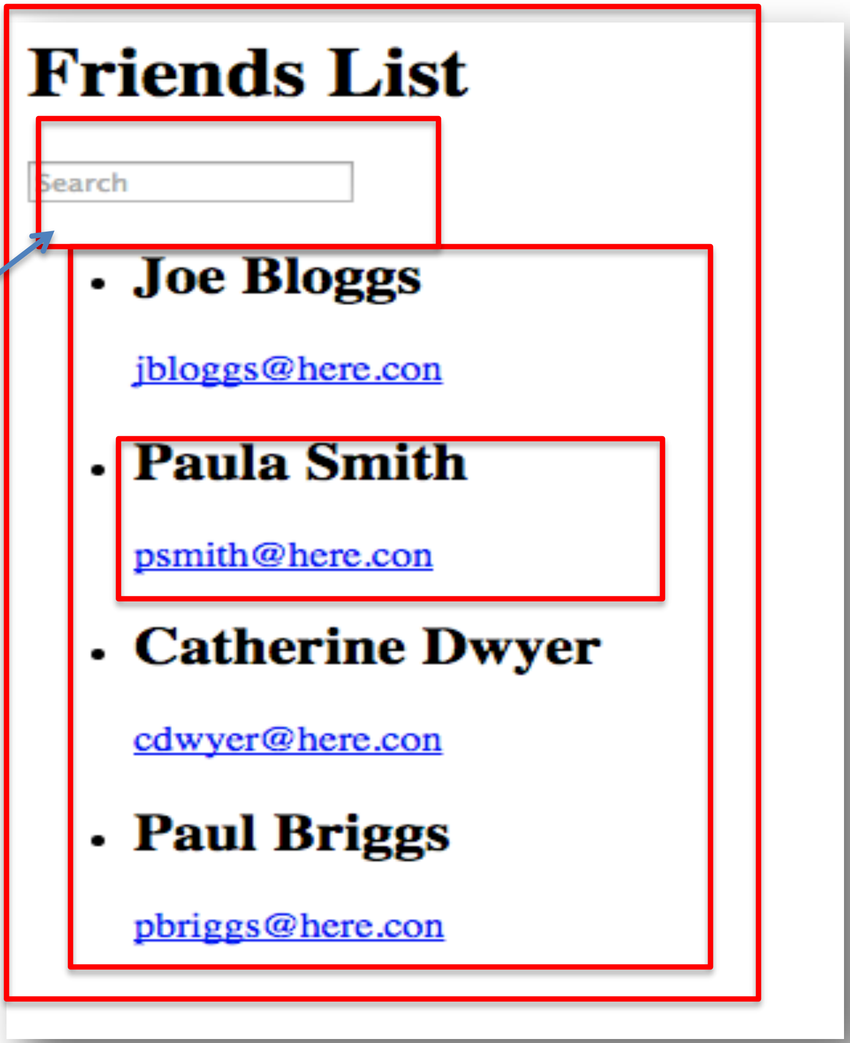


DEMO

Inverse data flow

- **What if a component's state is effected by an event in a subordinate component?**
- **Solution: The** inverse data flow pattern.

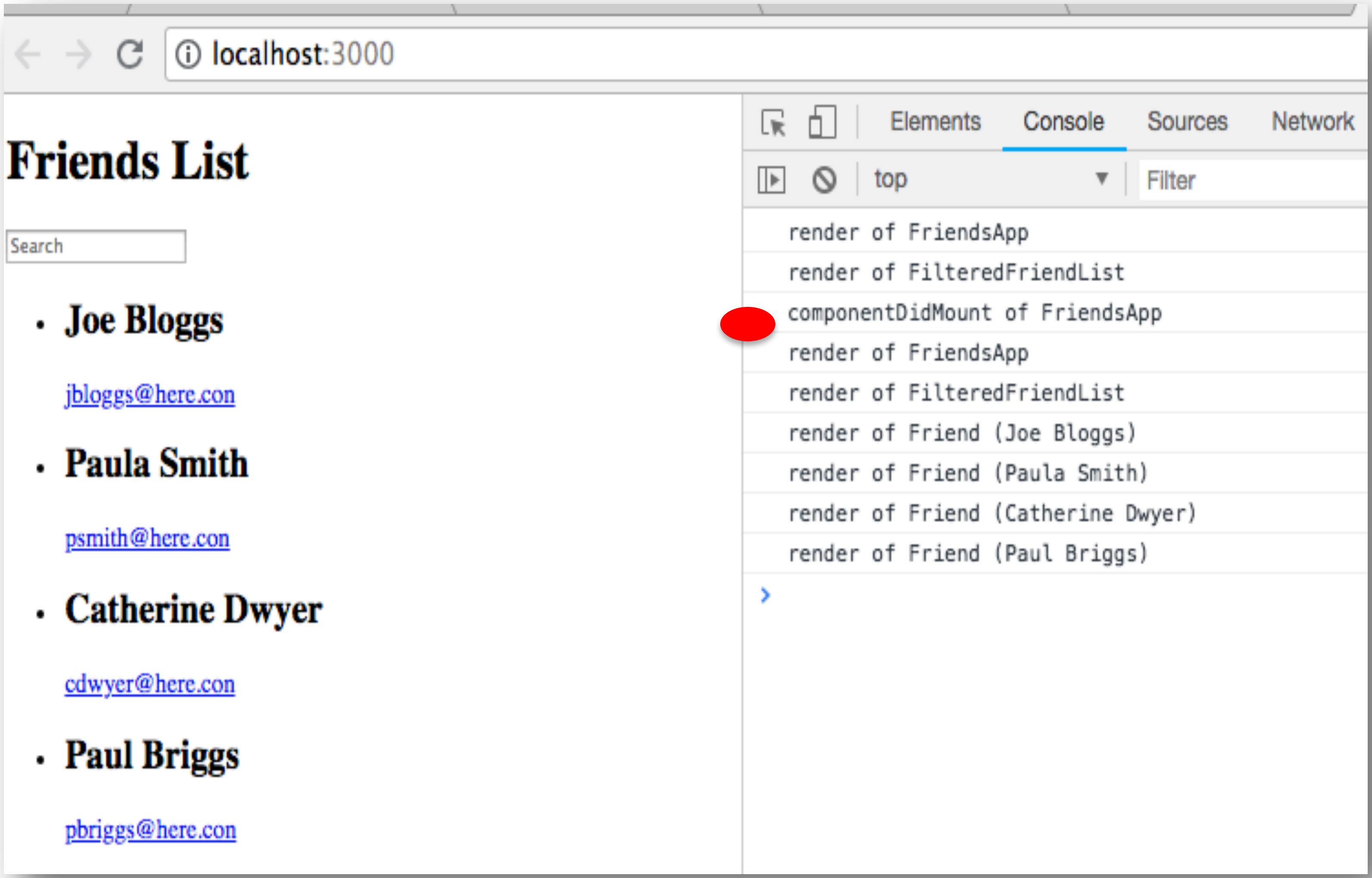




- **Joe Bloggs**
  jbloggs@here.con
- **Paula Smith**
  psmith@here.con
- **Catherine Dwyer**
  cdwyer@here.con
- **Paul Briggs**
  pbriggs@here.con

. . . . back to Lifecycle methods . . . .

Sample App – Execution trail (Mounting & setState)



Friends List

Search

- Joe Bloggs

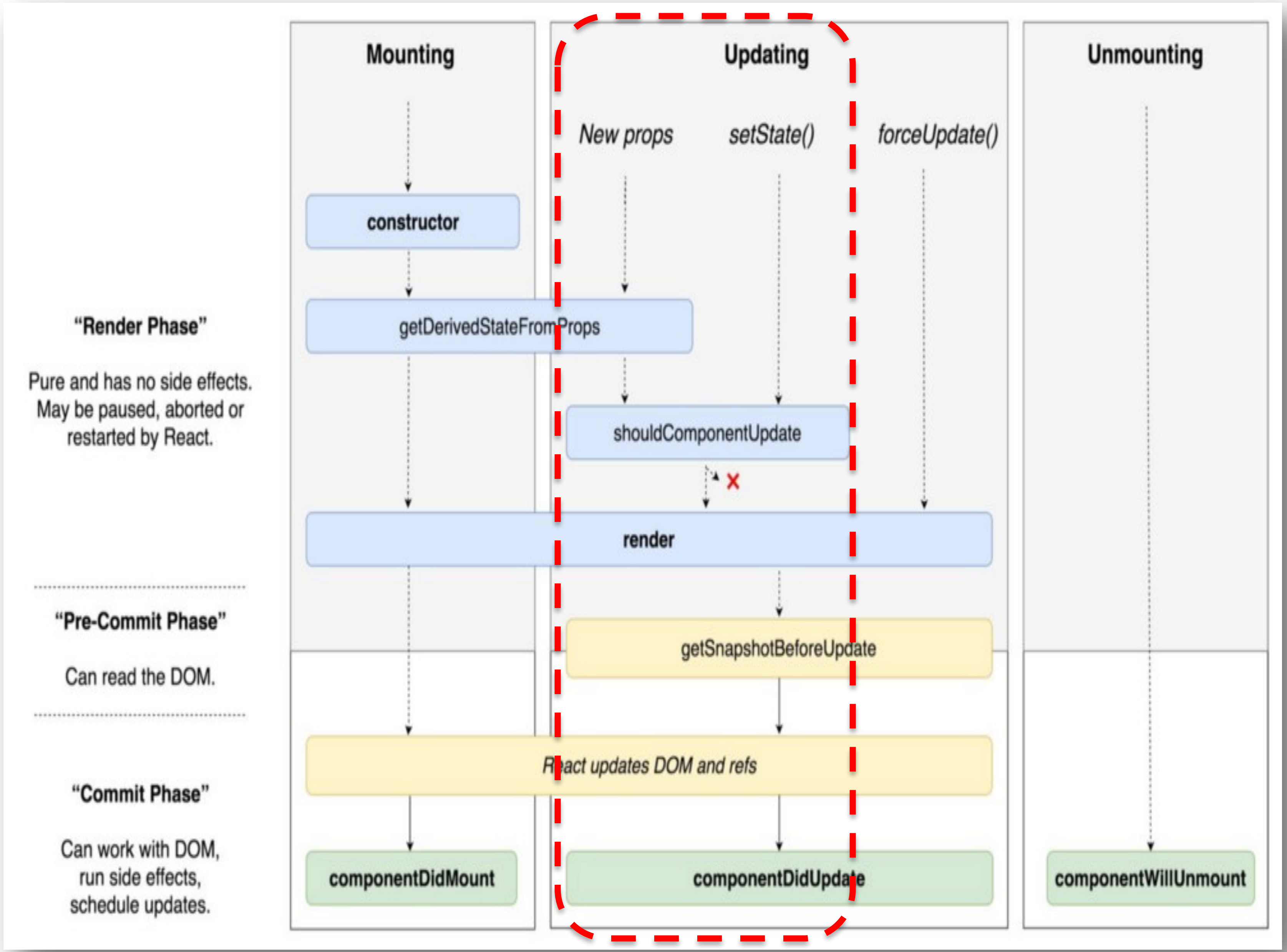
  jbloggs@here.con
- Paula Smith

  psmith@here.con
- Catherine Dwyer

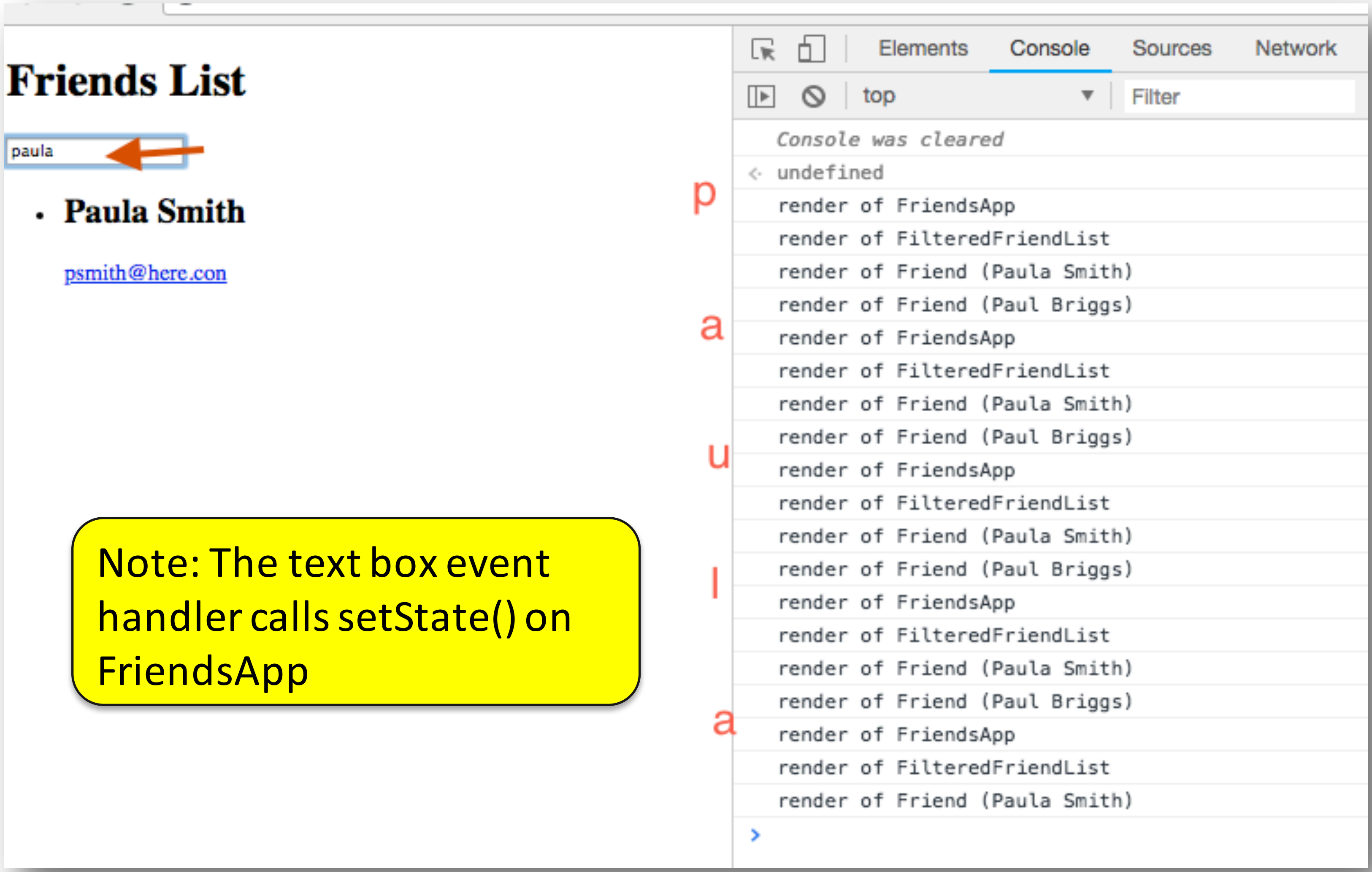
  cdwyer@here.con
- Paul Briggs

  pbriggs@here.con

```
render of FriendsApp
render of FilteredFriendList
componentDidMount of FriendsApp
render of FriendsApp
render of FilteredFriendList
render of Friend (Joe Bloggs)
render of Friend (Paula Smith)
render of Friend (Catherine Dwyer)
render of Friend (Paul Briggs)
```


Mounting
Updating
Unmounting
New props
setState()
forceUpdate()
constructor
getDerivedStateFromProps
shouldComponentUpdate
render
getSnapshotBeforeUpdate
React updates DOM and refs
componentDidMount
componentDidUpdate
componentWillUnmount
"Render Phase"
Pure and has no side effects. May be paused, aborted or restarted by React.
"Pre-Commit Phase"
Can read the DOM.
"Commit Phase"
Can work with DOM, run side effects, schedule updates.

Sample App – Execution trail (Update on new props & setState)..



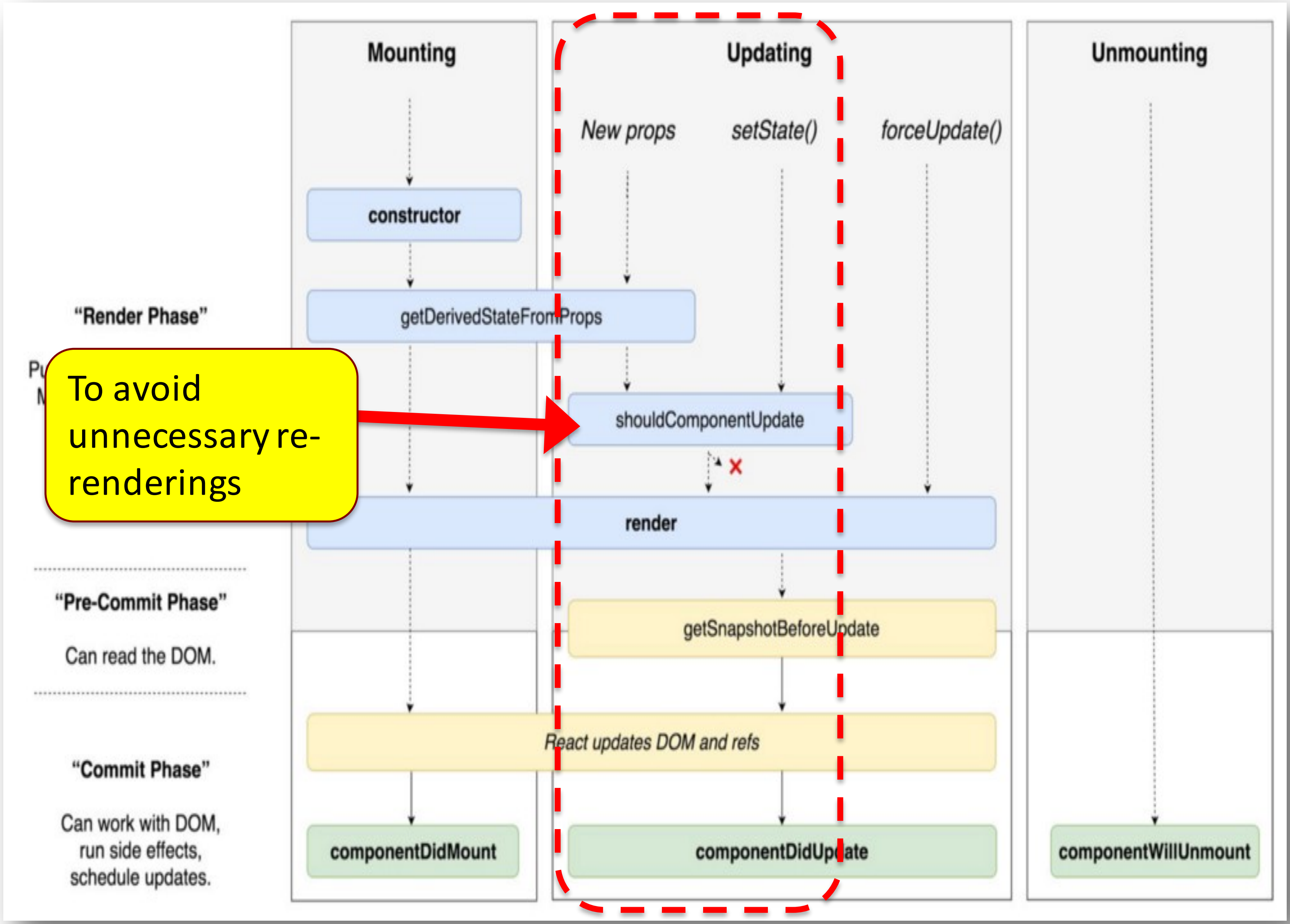
```
Console was cleared
< undefined
render of FriendsApp
render of FilteredFriendList
render of Friend (Paula Smith)
render of Friend (Paul Briggs)
render of FriendsApp
render of FilteredFriendList
render of Friend (Paula Smith)
render of Friend (Paul Briggs)
render of FriendsApp
render of FilteredFriendList
render of Friend (Paula Smith)
render of Friend (Paul Briggs)
render of FriendsApp
render of FilteredFriendList
render of Friend (Paula Smith)
render of Friend (Paul Briggs)
render of FriendsApp
render of FilteredFriendList
render of Friend (Paula Smith)
>
```

Note: The text box event handler calls setState() on FriendsApp

Unidirectional data flow & **Re-rendering**

- **What happens when user types in text box?**

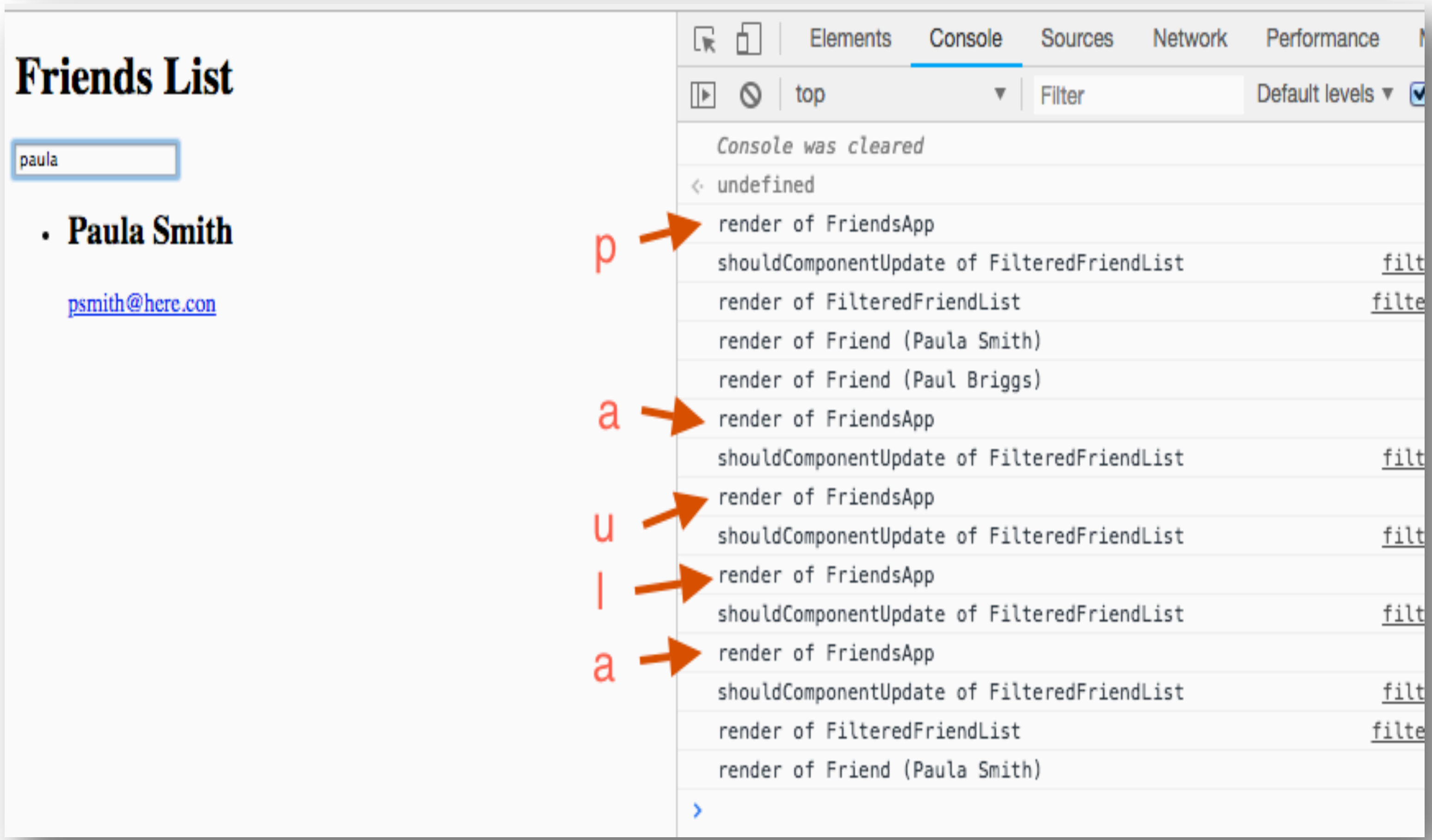
  User types a character in text box
  - → *onChange* ***event handler executes***
    - → ***Handler calls setState() (****FriendsApp* ***component)***
      - → ***React calls*** *FriendsApp* ***render() method***
      - → ***React calls render() method of children (****FilteredFriendList****) with new prop values***
      - → ***React calls render() method of*** *FilteredFriendList* ***children.***
      - → ***(Pre-commit phase) React re-computes the new Virtual DOM***
      - → ***React diffs the new and previous Virtual DOMs***
      - → ***(Commit phase) React batch updates the Real DOM***
      - → ***Browser repaints screen***



To avoid unnecessary re-renderings

Sample App – Execution trail (Update on new props & setState)..

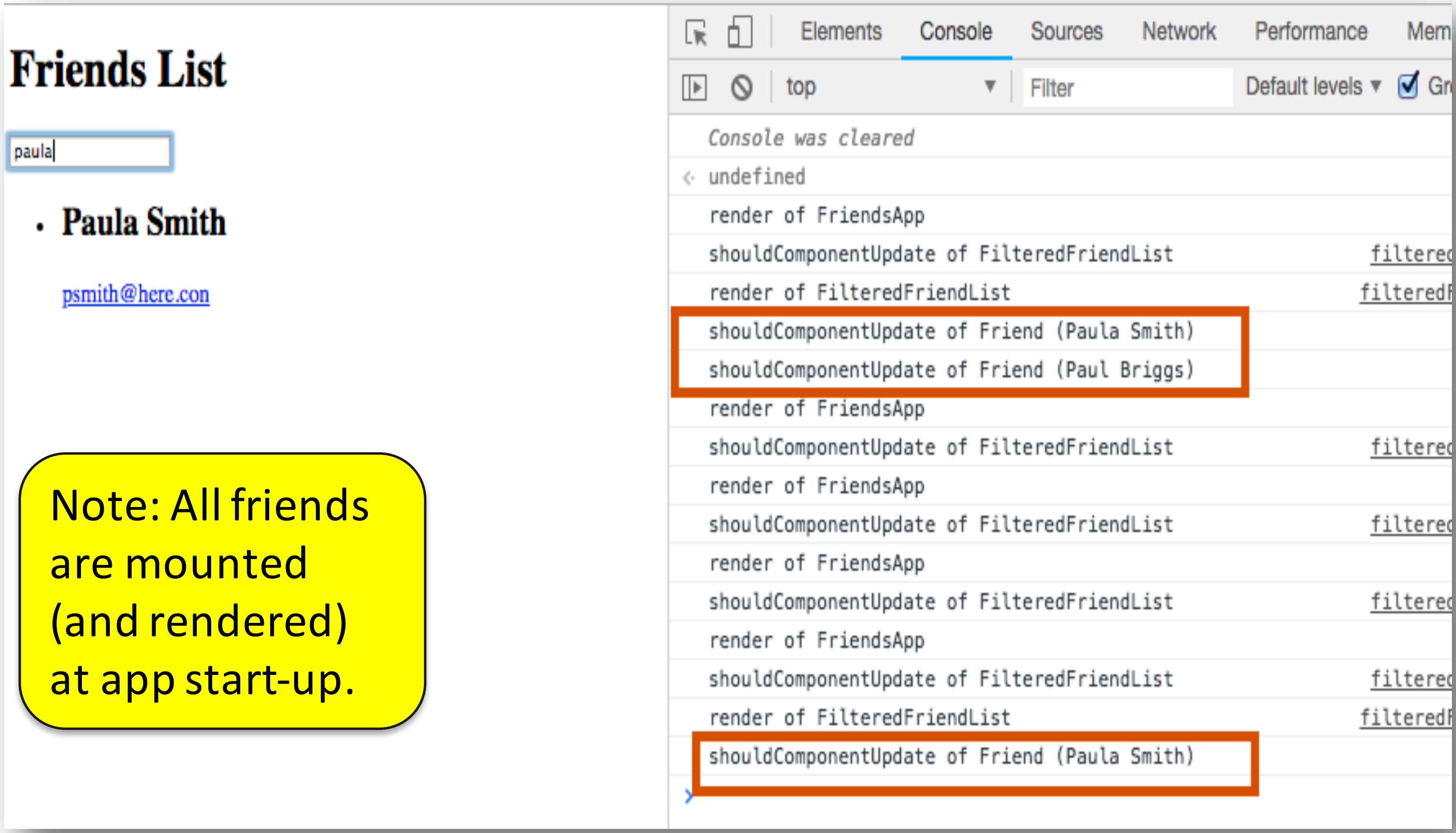
FilteredFriendsList should NOT re-render if the the length of array prop (of matching friends) has not changed



```
Console was cleared
undefined
render of FriendsApp
shouldComponentUpdate of FilteredFriendList                filt
render of FilteredFriendList                              filte
render of Friend (Paula Smith)
render of Friend (Paul Briggs)
render of FriendsApp
shouldComponentUpdate of FilteredFriendList                filt
render of FriendsApp
shouldComponentUpdate of FilteredFriendList                filt
render of FriendsApp
shouldComponentUpdate of FilteredFriendList                filt
render of FriendsApp
shouldComponentUpdate of FilteredFriendList                filt
render of FilteredFriendList                              filte
render of Friend (Paula Smith)
```

Sample App – Execution trail (Update on new props & setState)..

Friend should NOT re-render once it is mounted



Friends List

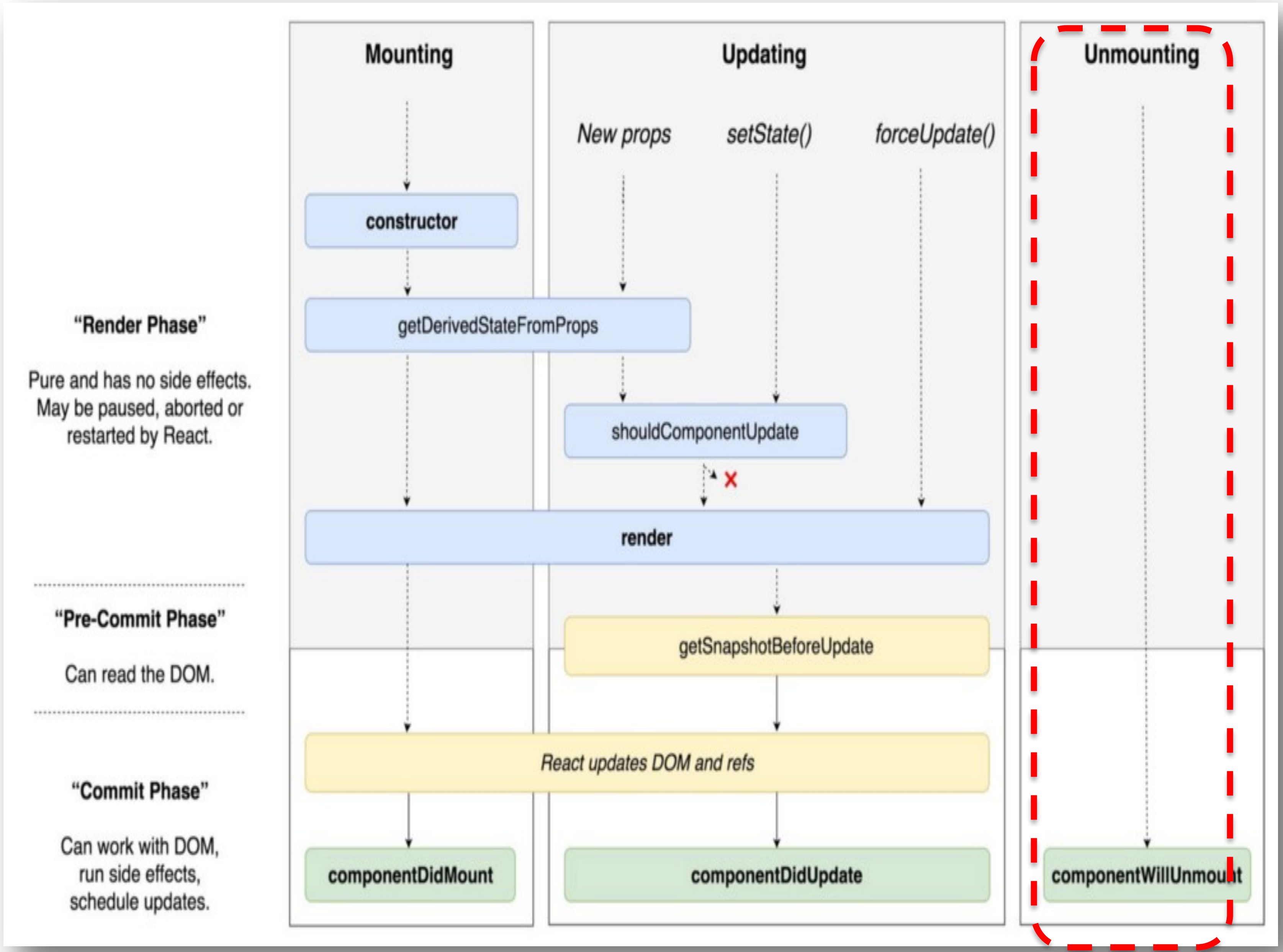
paula

- Paula Smith

  psmith@here.con

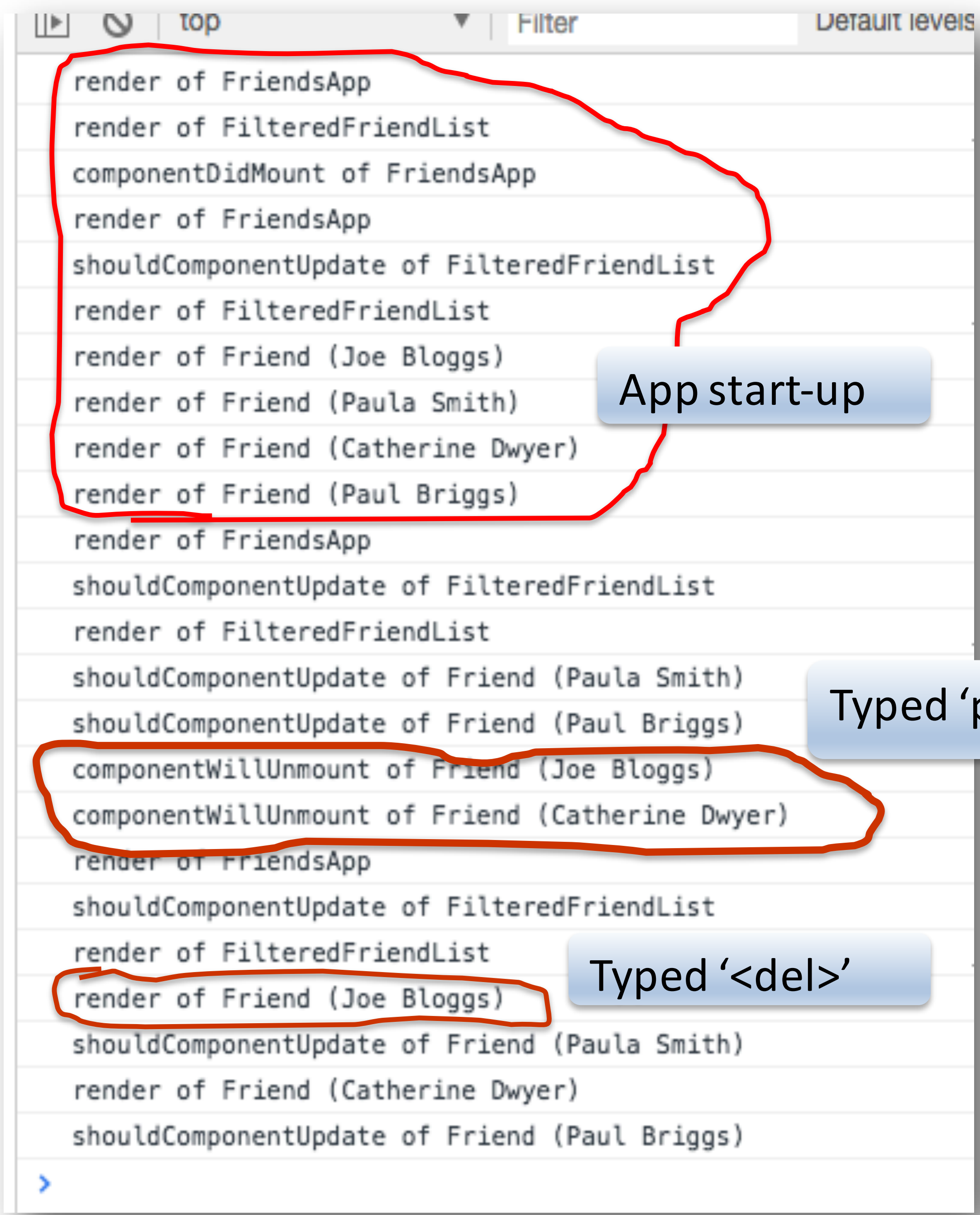
```
Console was cleared
undefined
render of FriendsApp
shouldComponentUpdate of FilteredFriendList                  filtered
render of FilteredFriendList                                filteredF
shouldComponentUpdate of Friend (Paula Smith)
shouldComponentUpdate of Friend (Paul Briggs)
render of FriendsApp
shouldComponentUpdate of FilteredFriendList                  filtered
render of FriendsApp
shouldComponentUpdate of FilteredFriendList                  filtered
render of FriendsApp
shouldComponentUpdate of FilteredFriendList                  filtered
render of FriendsApp
shouldComponentUpdate of FilteredFriendList                  filtered
render of FilteredFriendList                                filteredF
shouldComponentUpdate of Friend (Paula Smith)
```

Note: All friends are mounted (and rendered) at app start-up.



"Render Phase"
Pure and has no side effects. May be paused, aborted or restarted by React.
"Pre-Commit Phase"
Can read the DOM.
"Commit Phase"
Can work with DOM, run side effects, schedule updates.
Mounting
Updating
Unmounting
New props
setState()
forceUpdate()
constructor
getDerivedStateFromProps
shouldComponentUpdate
render
getSnapshotBeforeUpdate
React updates DOM and refs
componentDidMount
componentDidUpdate
componentWillUnmount

Sample App – Execution trail (Un-mounting)..



```
render of FriendsApp
render of FilteredFriendList
componentDidMount of FriendsApp
render of FriendsApp
shouldComponentUpdate of FilteredFriendList
render of FilteredFriendList
render of Friend (Joe Bloggs)
render of Friend (Paula Smith)
render of Friend (Catherine Dwyer)
render of Friend (Paul Briggs)
render of FriendsApp
shouldComponentUpdate of FilteredFriendList
render of FilteredFriendList
shouldComponentUpdate of Friend (Paula Smith)
shouldComponentUpdate of Friend (Paul Briggs)
componentWillUnmount of Friend (Joe Bloggs)
componentWillUnmount of Friend (Catherine Dwyer)
render of FriendsApp
shouldComponentUpdate of FilteredFriendList
render of FilteredFriendList
render of Friend (Joe Bloggs)
shouldComponentUpdate of Friend (Paula Smith)
render of Friend (Catherine Dwyer)
shouldComponentUpdate of Friend (Paul Briggs)
>
```

Summary

- **For interactive apps we record the user's input/interaction in component(s) state object.**
  - **The interaction may cause UI changes – dynamic app.**

- **React achieves DOM update performance improvements by managing an intermediate data structure, the Virtual DOM.**

- **Data only flows downward through the component hierarchy – this aids debugging.**

- **A component's life-span includes stages, from** mounting **to** un-mounting, **and phases, including** render, pre-commit and post-commit**.**
  - **We can hook logic in to the life span at prescribed times using lifecycle methods.**