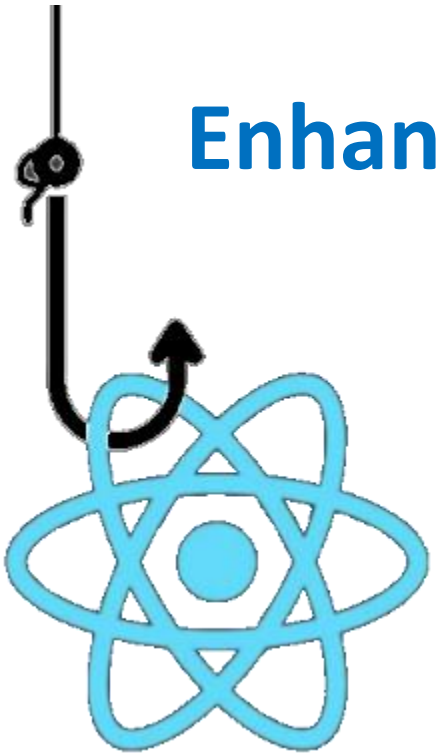


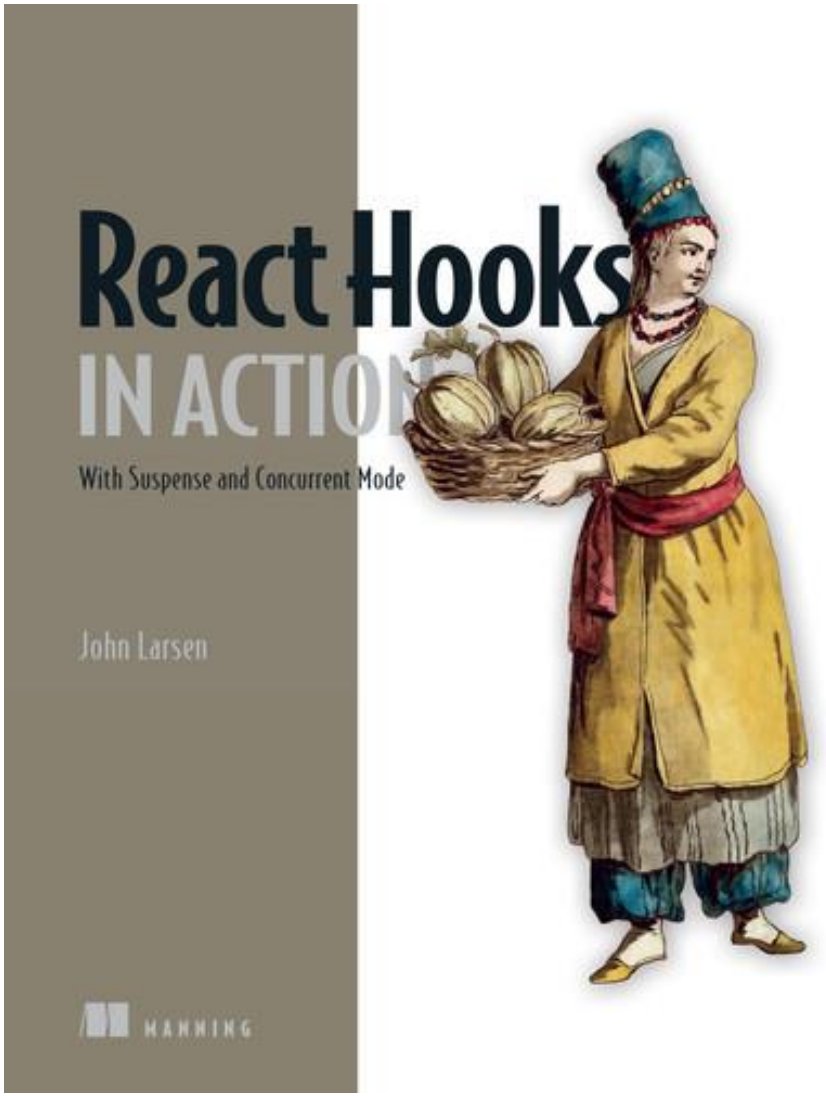
Enhance the Performance of React Apps

Outline

1. useMemo
2. useCallback
3. useTransition
4. <Suspend> Component

Aim = improve the user experience of our apps, making them feel more responsive

Slides are based on



https://learning.oreilly.com/library/view/react-hooks-in/9781617297632/

Developing a web application and interface to receive mission requests for an Earth Observation Satellite from satellite operators

- Dr. Abdelkarim Erradi is an expert in Web Application design, development, deployment and scaling including many years of industrial experience
- He delivered Web and Mobile development courses to both undergraduate and graduate CSE students
- Relevant publications:

- A. Erradi, W. Iqbal, A. Mahmood and A. Bouguettaya, "Web Application Resource Requirements Estimation Based on the Workload Latent Features," in *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 1638-1649, 1 Nov.-Dec. 2021, doi: 10.1109/TSC.2019.2918776. https://ieeexplore.ieee.org/abstract/document/8723190
- M. Abdullah, W. Iqbal, A. Erradi, Unsupervised learning approach for web application auto-decomposition into microservices, Journal of Systems and Software, Volume 151, 2019. https://www.sciencedirect.com/science/article/abs/pii/S0164121219300408

Configuration management approaches and tool for software product line and software product

- Dr. Abdelkarim Erradi is an expert in Web Application design, development, deployment and scaling including many years of industrial experience
- He delivered Web and Mobile development courses to both undergraduate and graduate CSE students
- Relevant publications:

- A. Erradi, W. Iqbal, A. Mahmood and A. Bouguettaya, "Web Application Resource Requirements Estimation Based on the Workload Latent Features," in *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 1638-1649, 1 Nov.-Dec. 2021, doi: 10.1109/TSC.2019.2918776. https://ieeexplore.ieee.org/abstract/document/8723190
- M. Abdullah, W. Iqbal, A. Erradi, Unsupervised learning approach for web application auto-decomposition into microservices, Journal of Systems and Software, Volume 151, 2019. https://www.sciencedirect.com/science/article/abs/pii/S0164121219300408

useMemo

- Avoid unnecessarily rerunning of expensive computations by wrapping them in the useMemo hook
    - Pass useMemo the expensive function you want to memorize and control re-execution with a dependency array

```
const value = useMemo(
  () => expensiveFn(dep1, dep2), [dep1, dep2]
);
```

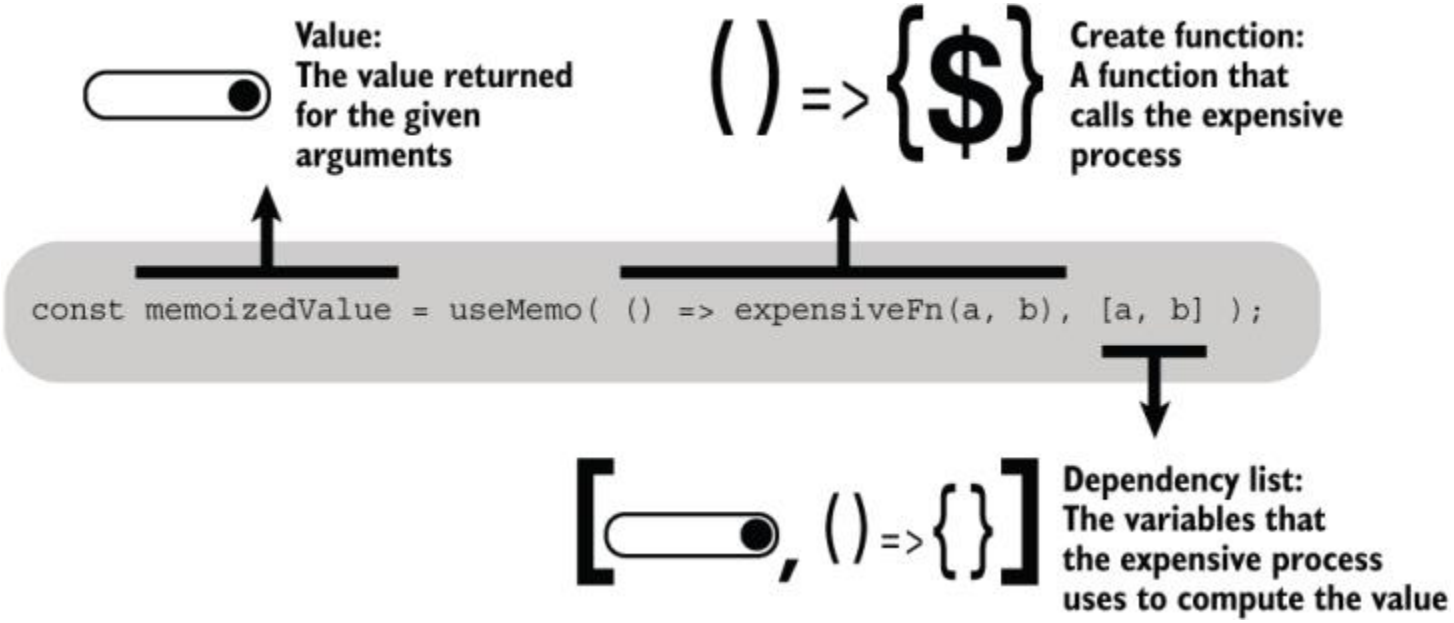
- Use the **memo** hook to avoid unnecessary re-rending of components
    - Help us avoid unnecessary and wasteful work to protect the user from sluggish UI updates

Memoizing

- By calling the function inside the useMemo hook, we ask React to
  - store a value computed by the function for a given set of arguments
  - If we call the function inside useMemo again, using the same arguments as the previous call, it should return the stored value
  - If we pass different arguments, it will use the function to compute a new value and update its store before returning the new value
- The process of storing a result for a given set of arguments is called memoizing

Memoizing expensive function calls with useMemo

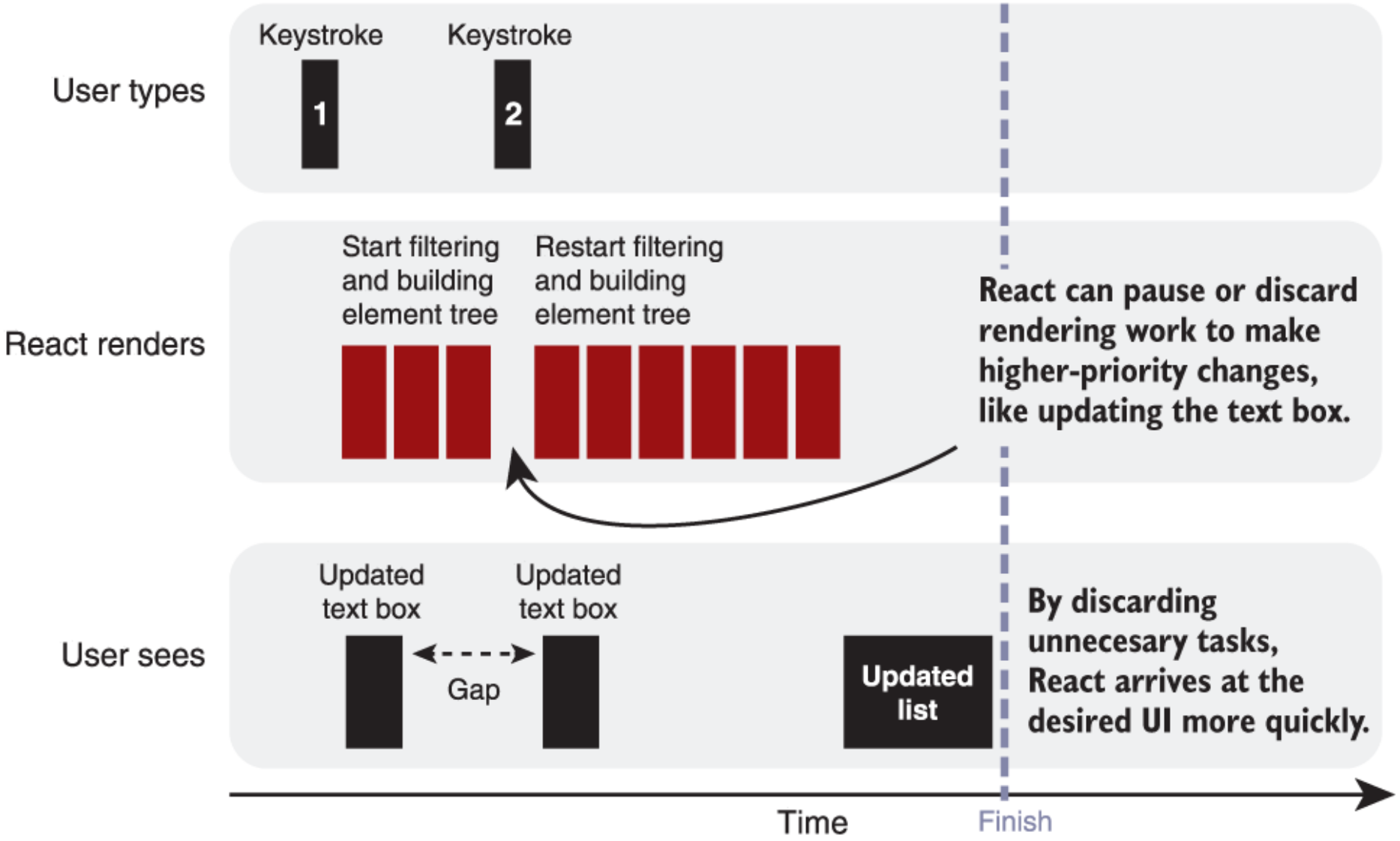
- Wrap expensive functions inside a useMemo hook and control re-execution with a dependency array
  - If the variables in the dependency array don't change from one call to the next, useMemo returns its stored result for the expensive function



useTransition

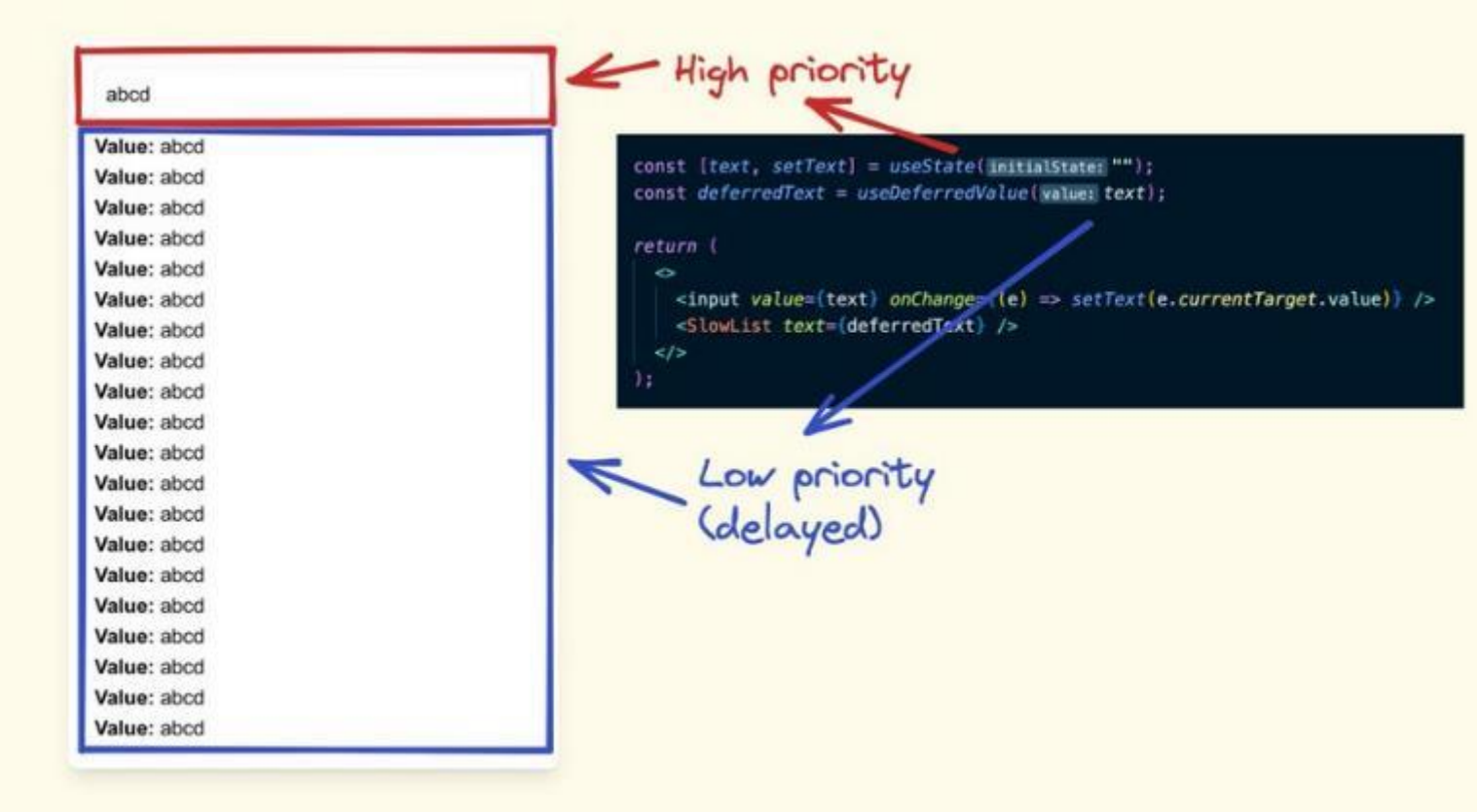
- Use useTransition to request React to delay (i.e., **lower the priority)** of updating parts of DOM after state changes to make sure it first responds to user interactions
    - E.g., when filtering a list, React can pause rendering of the filtered list to make sure the text that the user is typing appears in the textbox
- Reconciling and committing changes to one part of the component tree can be paused or abandoned to make sure components with higher priority are updated first
    - Enables React can pause longer-running updates to quickly react to user interactions to deliver smoother UI

useTransition



useDeferredValue

- useDeferredValue lets you split one piece of state into two: a **high priority version**, and a low priority one
  - In this example, the input updates immediately, and the SlowList renders whenever it can.



useDeferredValue

- useTransition() wraps the state updating code, whereas useDeferredValue() wraps a value affected by the state change
- If you have access to the **state updating code** and have some state updates that should be treated with a lower priority, it makes sense to use useTransition(). Use useDeferredValue() if you don't have that access
- UseTransition() and useDeferredValue() should not be used to wrap up all of your state updates or values.
    - Only use if a component **can't be optimized any other way**. Other performance enhancements, such as lazy loading, pagination, and performing work in worker threads or on the back end, should always be considered.

Suspend Component

- Suspense component allows rendering **fallback content** to indicate that a component is waiting for something, like loading data needed to build the UI

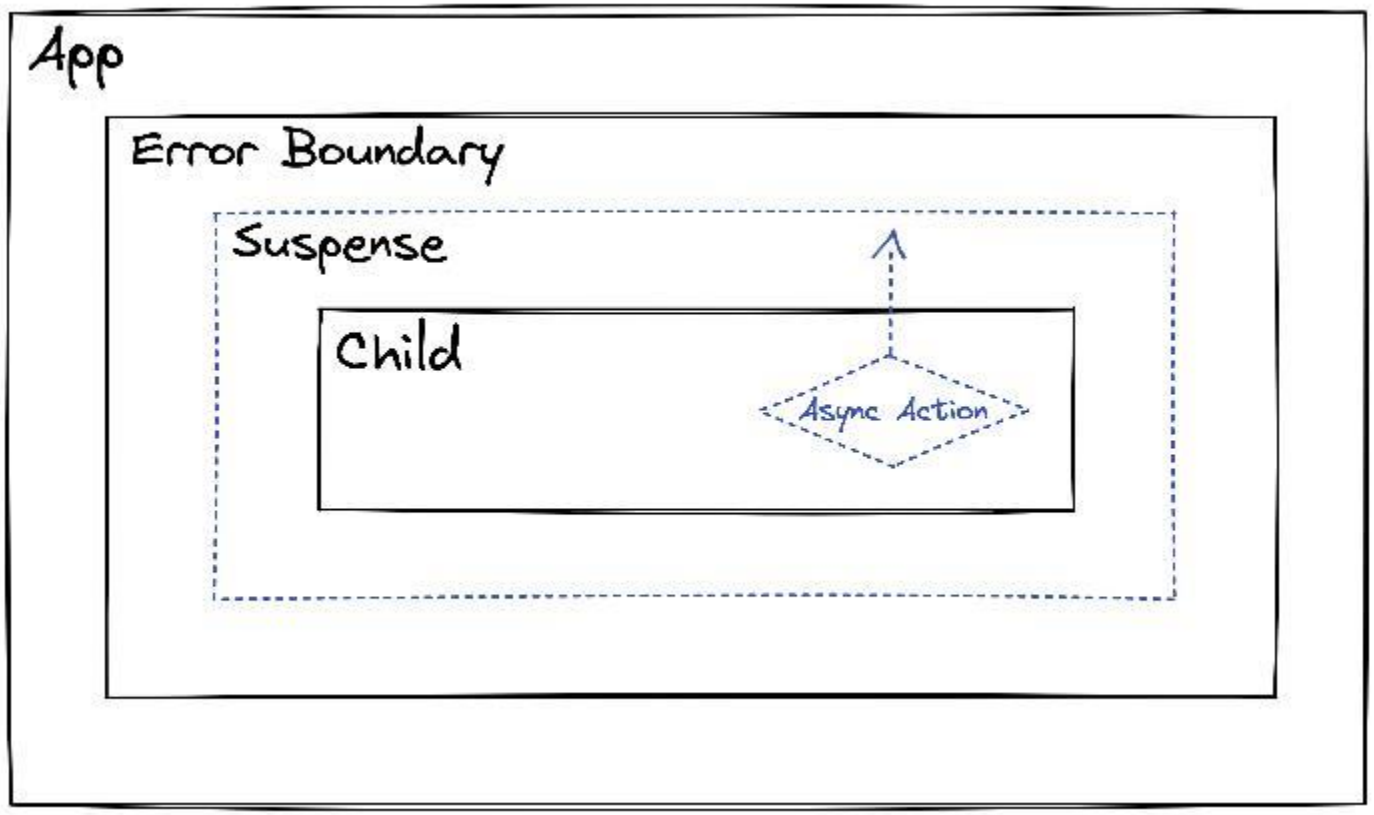
```
<Suspense fallback={<MySpinner />}>
   <MyFirstComponent />
   <MySecondComponent />
</Suspense>
```

- Suspense allows showing fallbacks either for individual components or groups of components

  - Suspense simplifies managing loading states in React components and provides a better way of showing the user a loading screen

Suspend Component

- Suspend component wraps a component, which is loading the data from some data source, and it will show a **fallback** until the data fetching is complete



React Query

- React Query is a library for fetching, caching, synchronizing and updating server state in a React app
- https://tanstack.com/query/v4/docs/overview
- `npm i @tanstack/react-query`

Summary

- **useMemo:** avoid rerunning expensive computations
- **useTransition**: React can pause longer-running updates to quickly react to user interactions
- **Suspense** component allows rendering **fallback content** to indicate that a component is waiting for something, like loading data

Resources

- Thinking in React

https://reactjs.org/docs/thinking-in-react.html

- React Router

https://reactrouter.com/

- Useful list of resources

https://github.com/enaqx/awesome-react