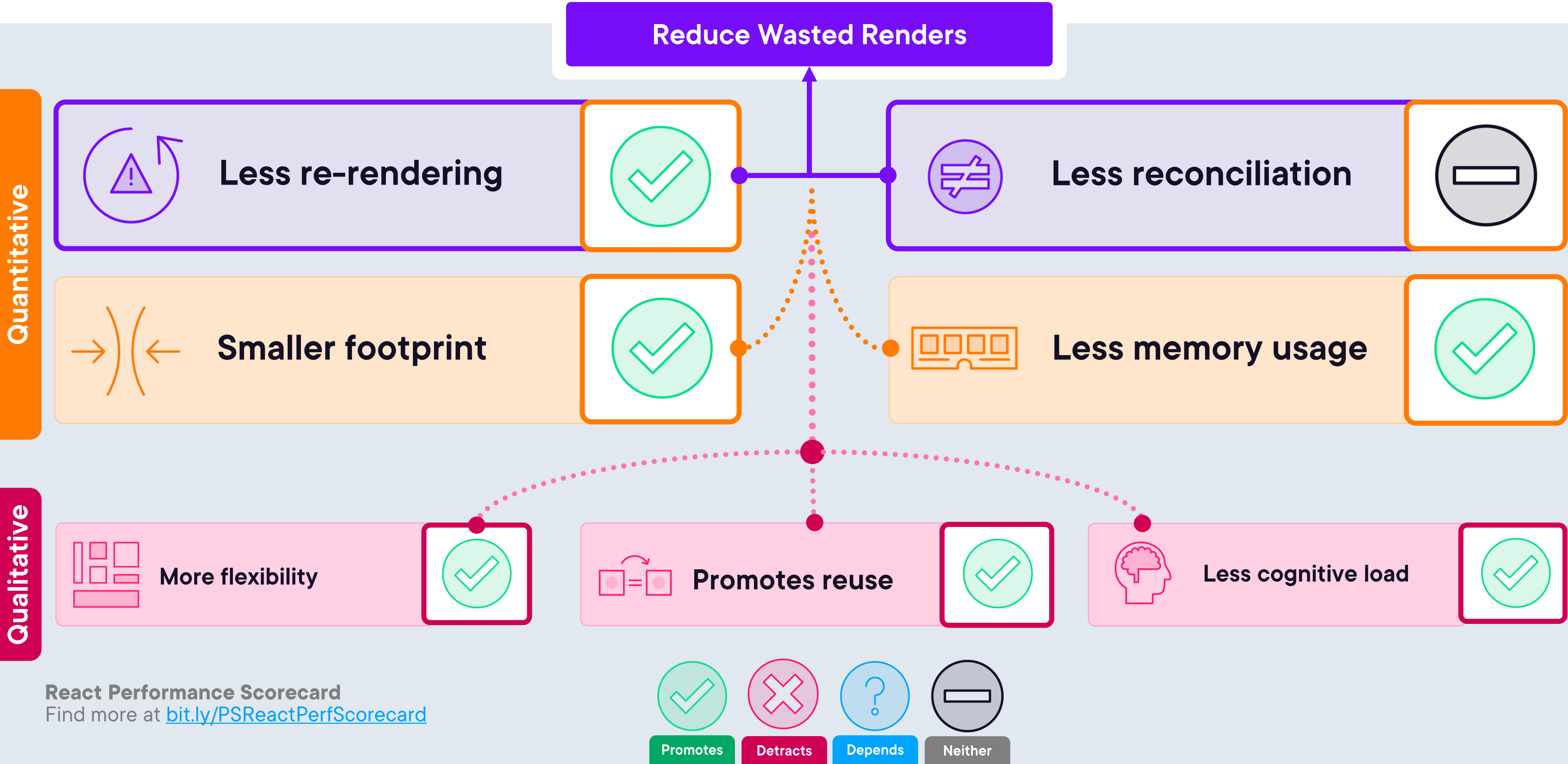
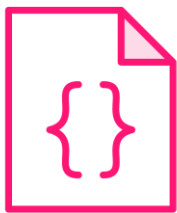


Component Slots

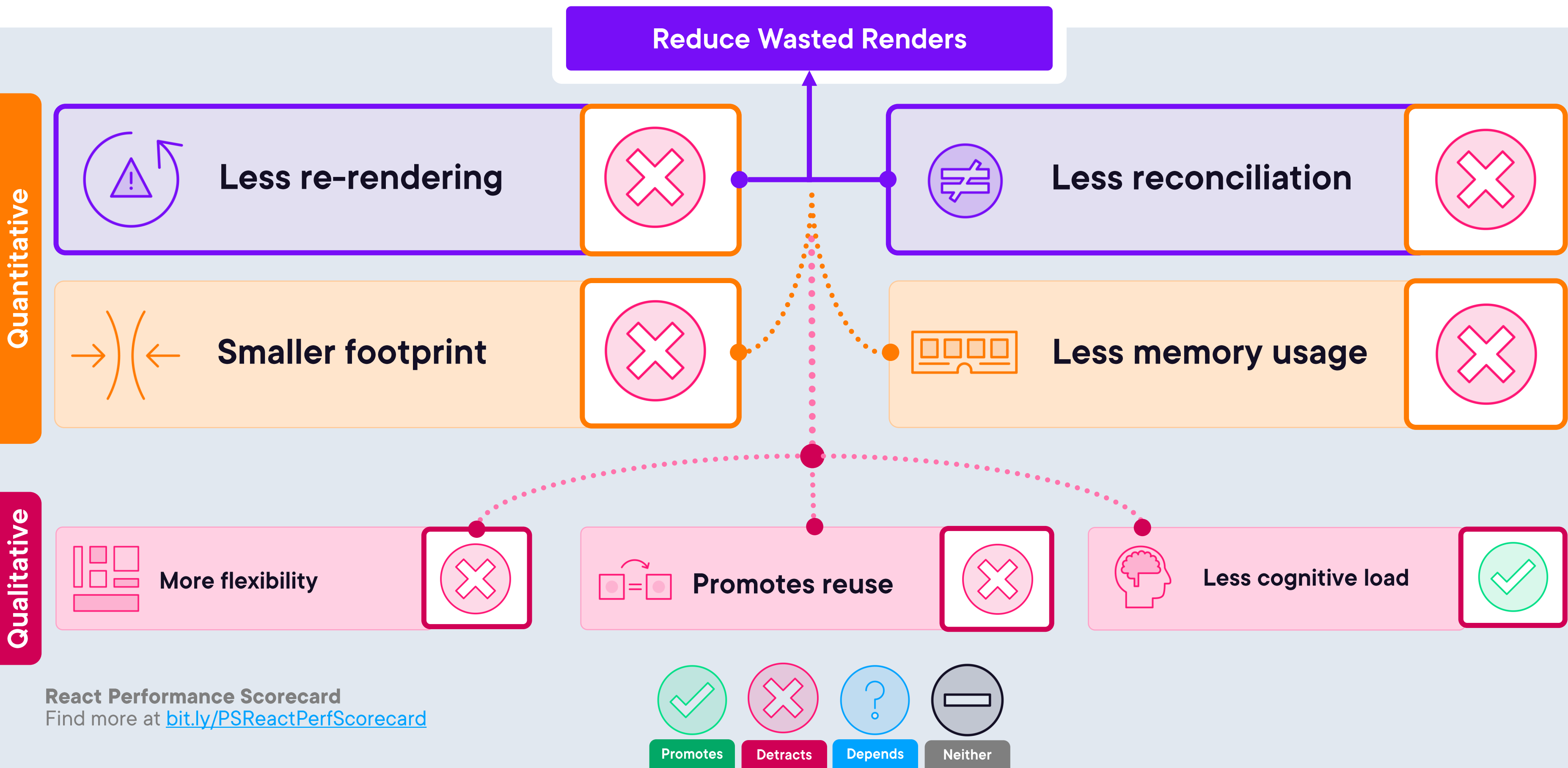
Using props to substitute nested component trees and change rendering ownership





Inline Style Objects

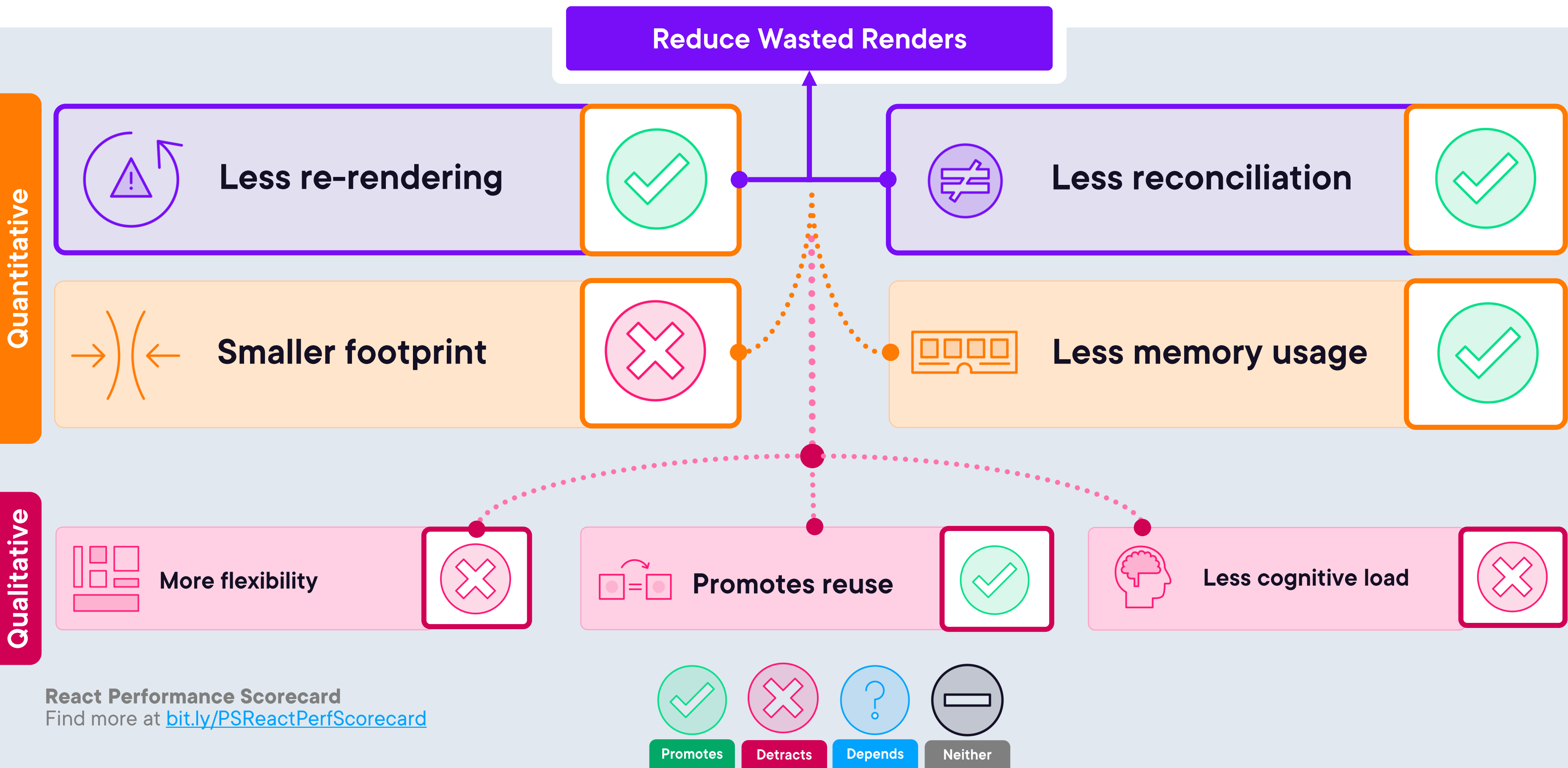
Passing styles via the `style` prop and inline or const objects

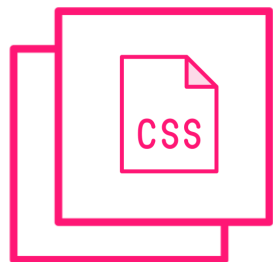




External Styles

Importing or linking to external stylesheet CSS files, including using pre- or post-processors





CSS Modules

Component Design Scorecard

Importing scoped CSS files that transform and hash class selectors to static classNames

Reduce Wasted Renders

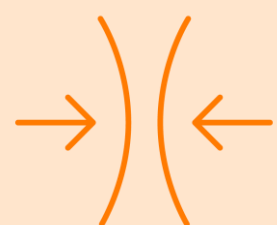
Quantitative



Less re-rendering



Less reconciliation



Smaller footprint



Less memory usage



Qualitative



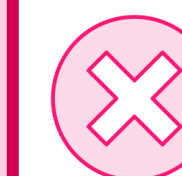
More flexibility



Promotes reuse



Less cognitive load



React Performance Scorecard
Find more at bit.ly/PSReactPerfScorecard



Promotes



Detracts



Depends



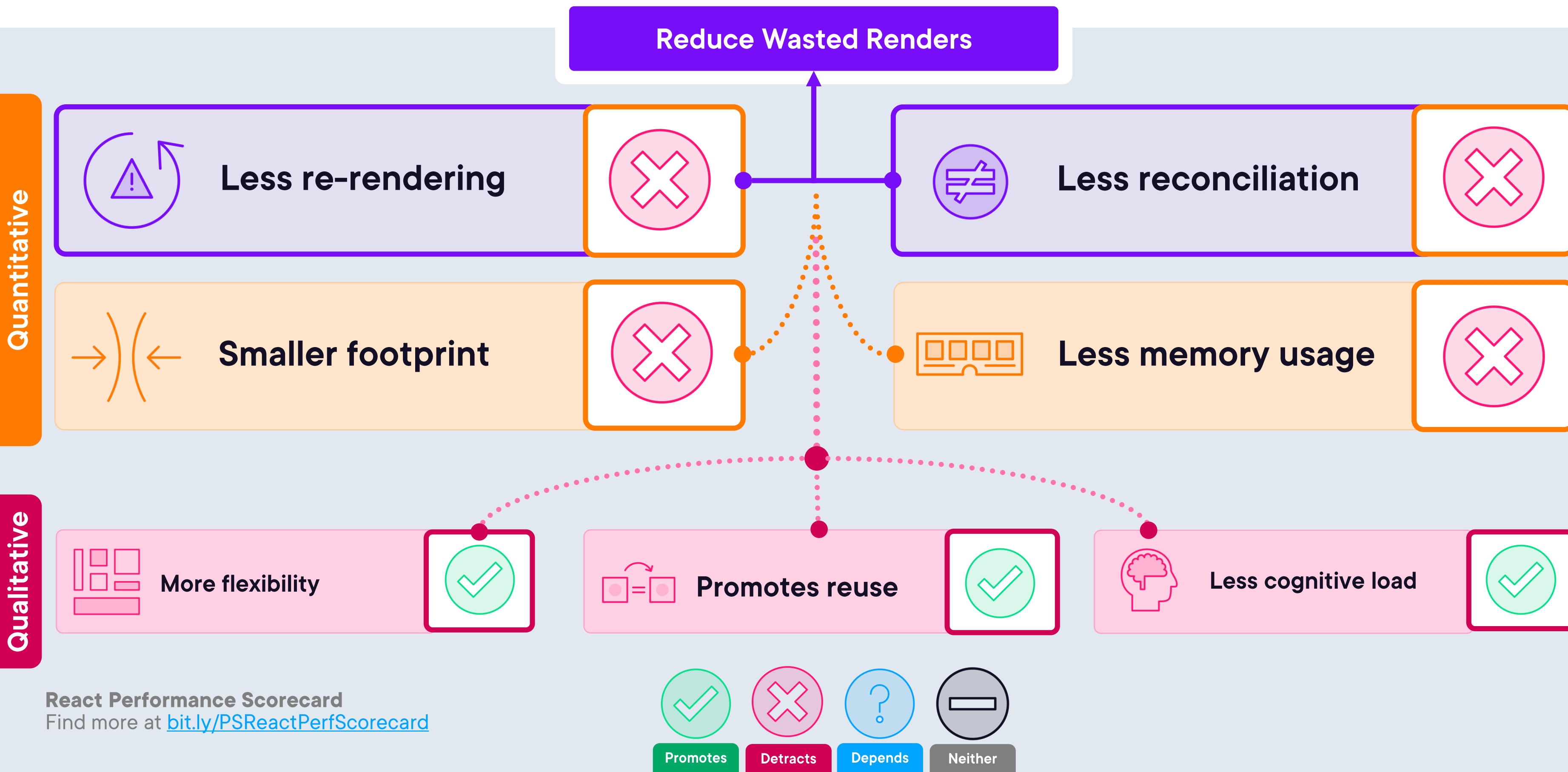
Neither

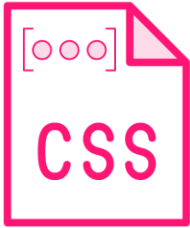


CSS-in-JS

Component Design Scorecard

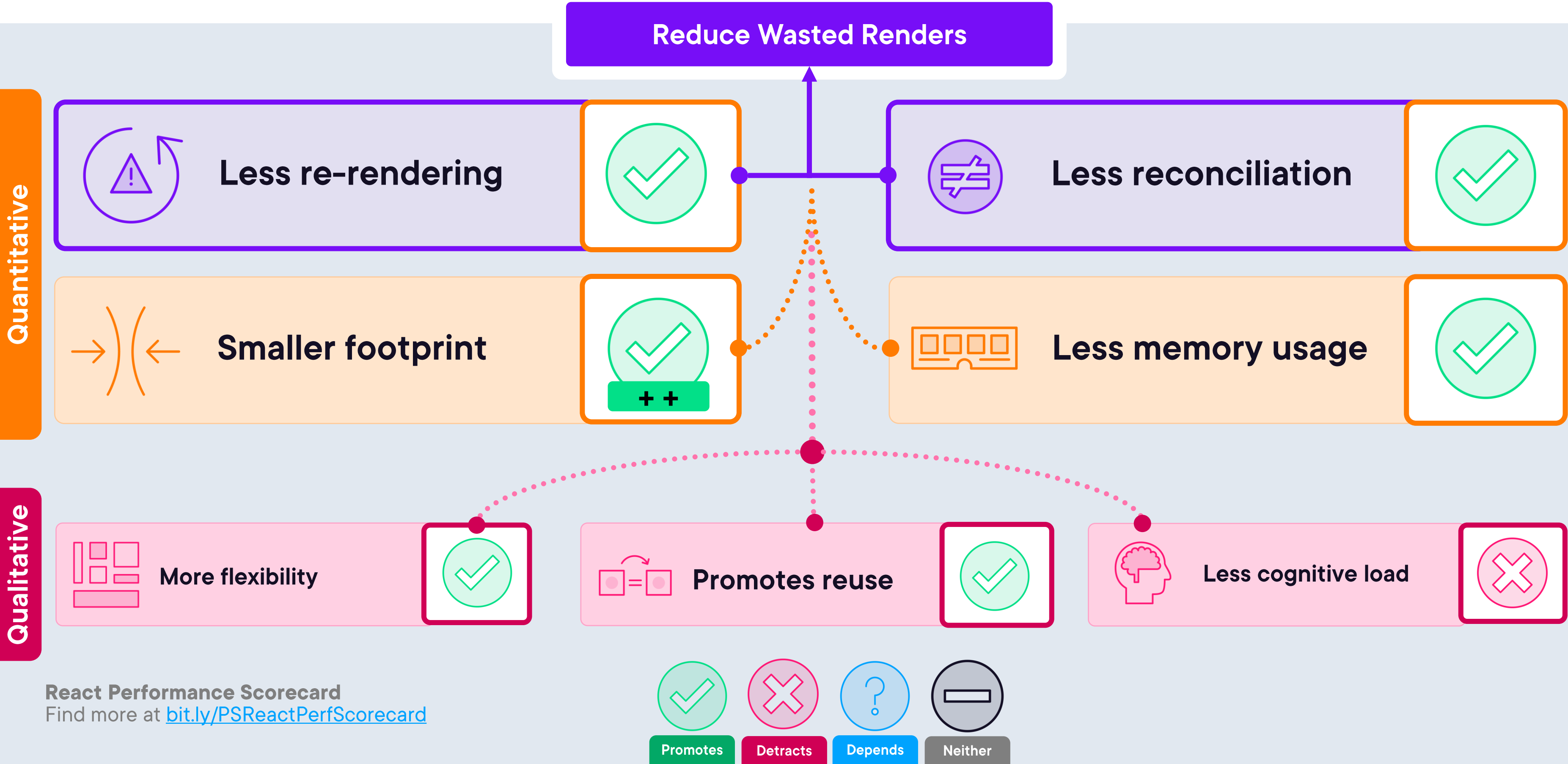
Importing scoped CSS files that transform and hash class selectors to static classNames





Atomic CSS

Atomic classes that can be composed into complex component UIs

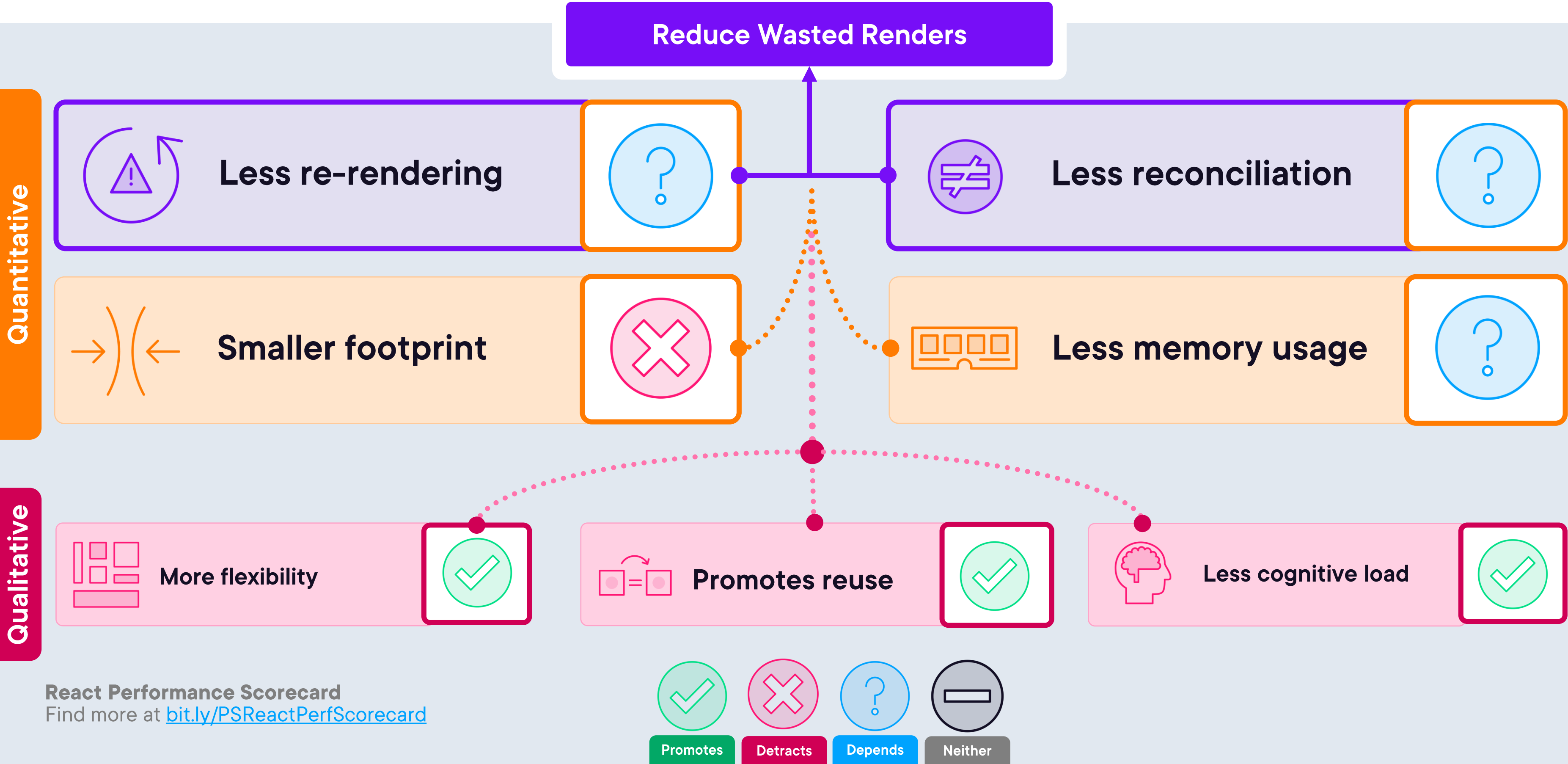


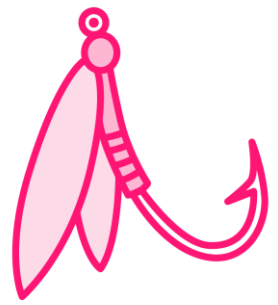


CSS Frameworks

Component Design Scorecard

Shared design systems or CSS frameworks that offer flexible styling approaches

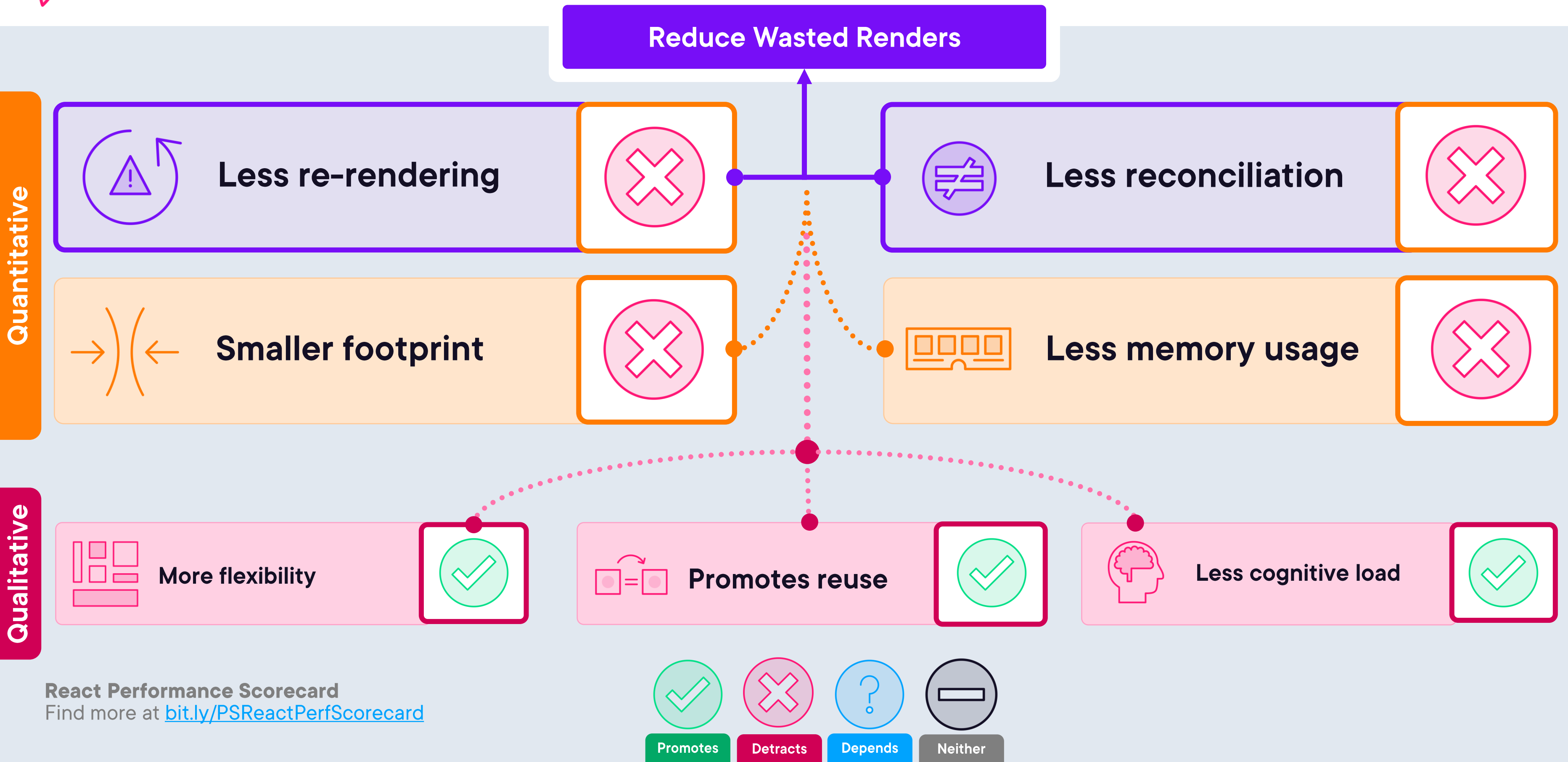


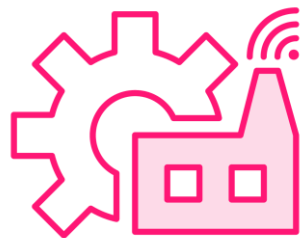


Custom Hooks

Component Design Scorecard

Reducing hook usage in components by encapsulating them into custom hooks

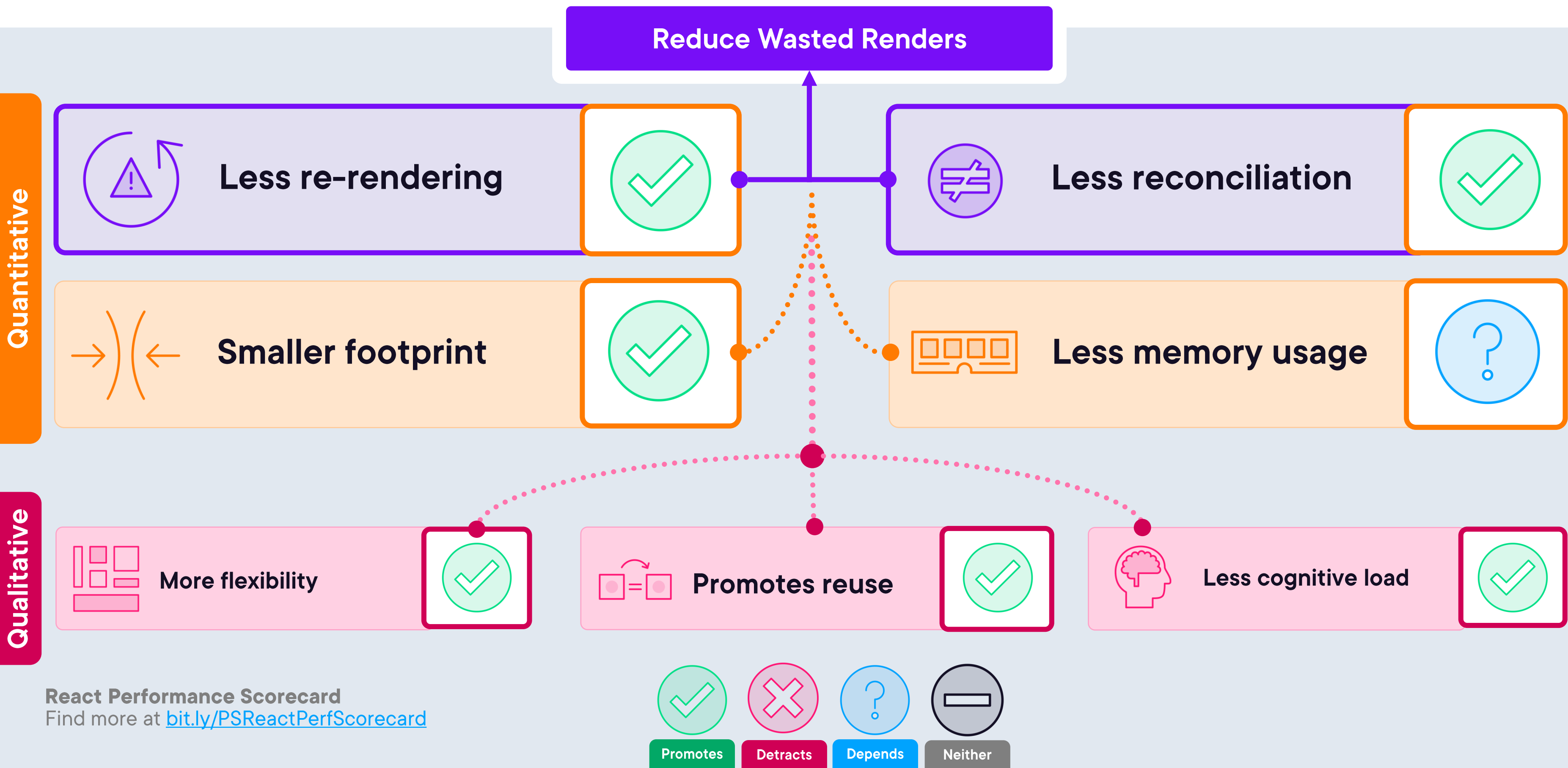




State Factories

Component Design Scorecard

Generating specialized hooks
to share state or logic without Context

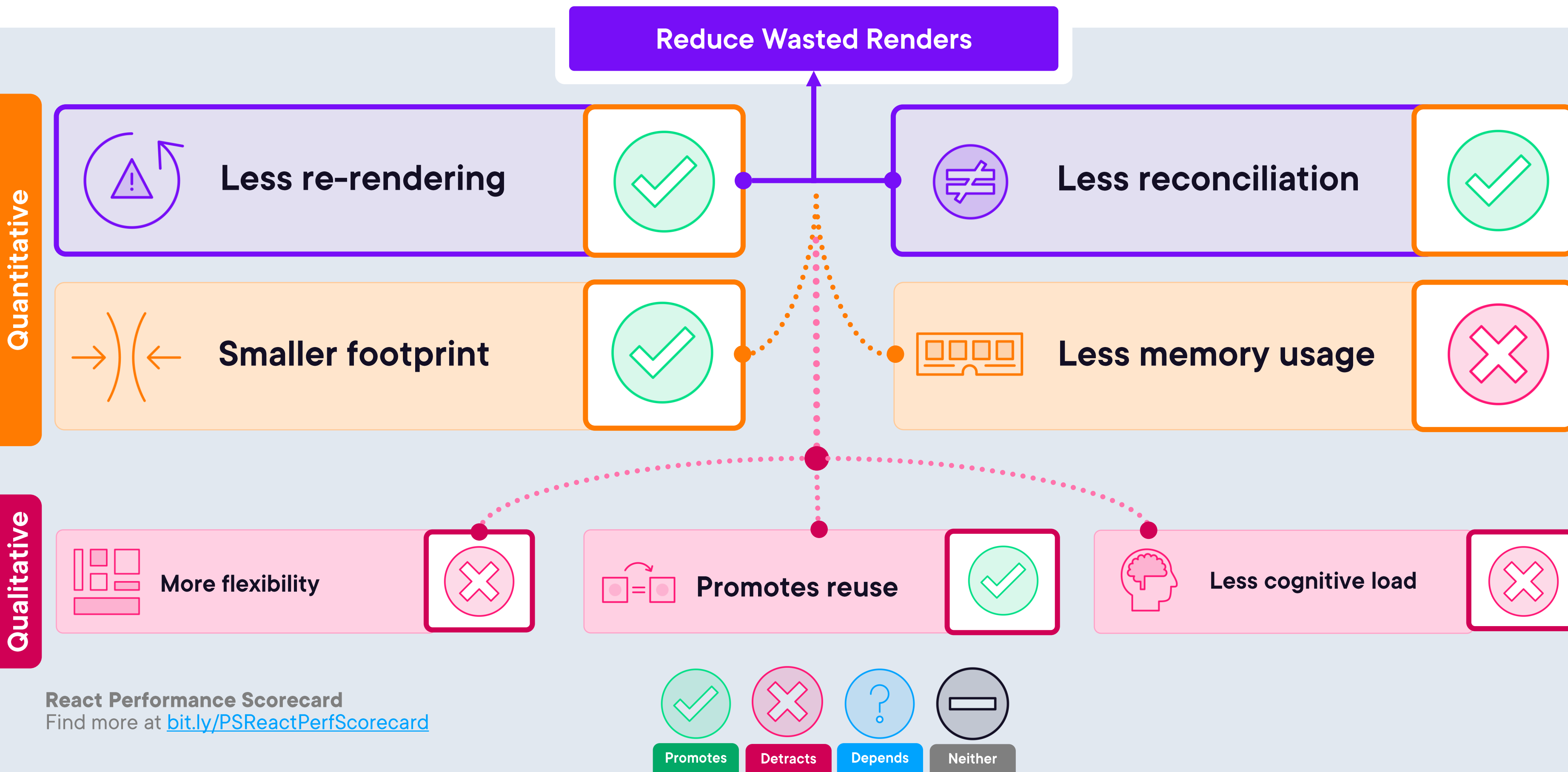




Query Hooks

Component Design Scorecard

Specialized hooks that handle data fetching with caching and error handling





Render Props

Component Design Scorecard

Sandbox parts of your component tree by limiting updates that could affect a parent component

Reduce Wasted Renders

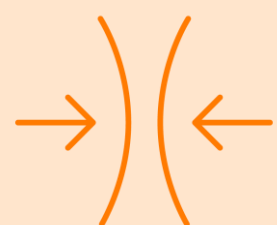
Quantitative



Less re-rendering



Less reconciliation



Smaller footprint



Less memory usage



Qualitative



More flexibility



Promotes reuse



Less cognitive load



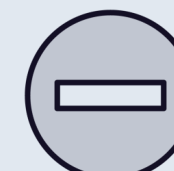
Promotes



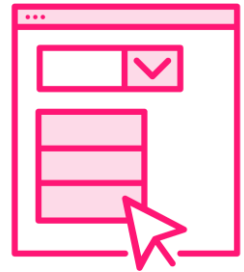
Detracts



Depends

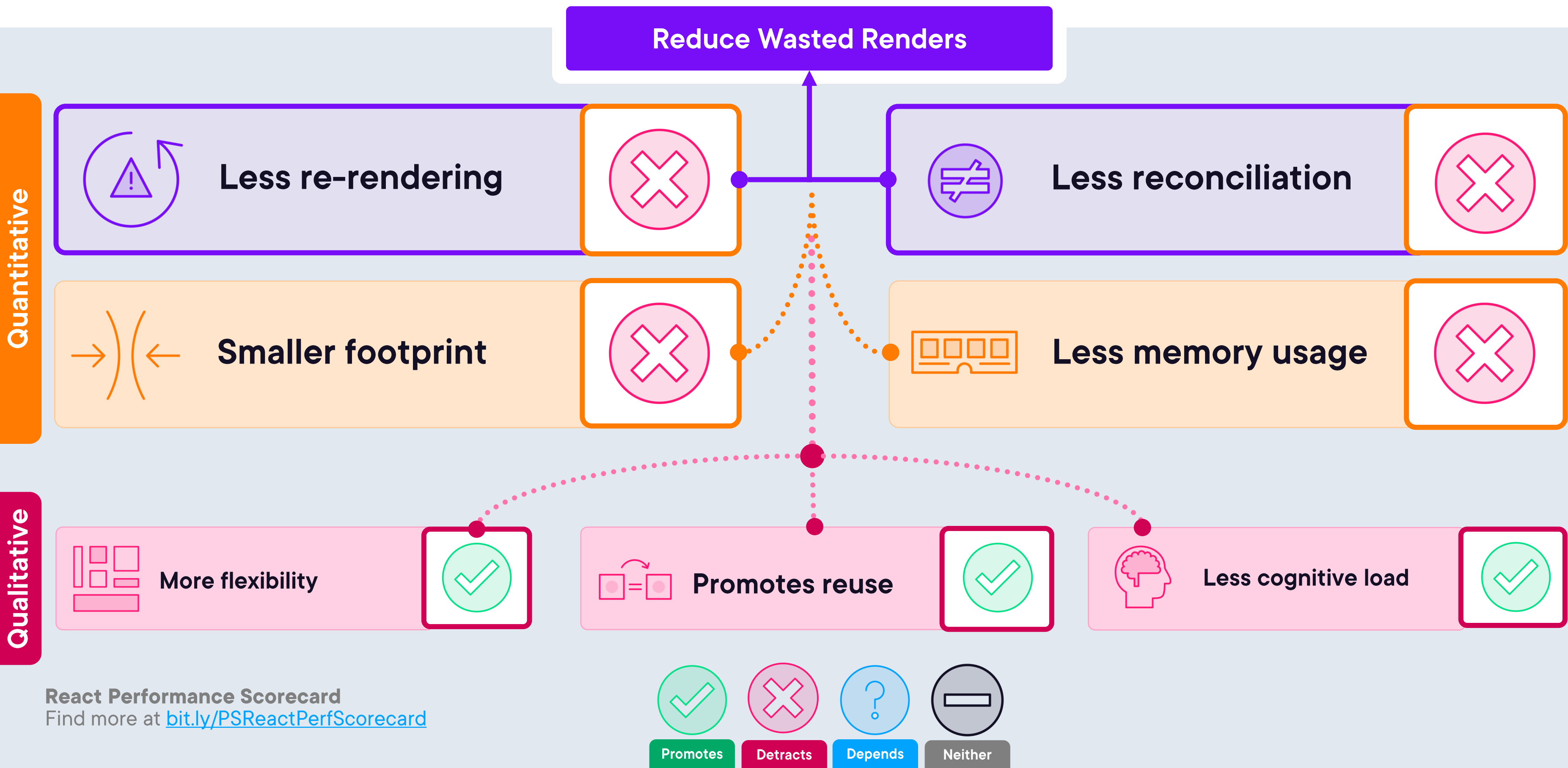


Neither



Controlled Components

Creating a predictable data flow by having React manage the state of the DOM

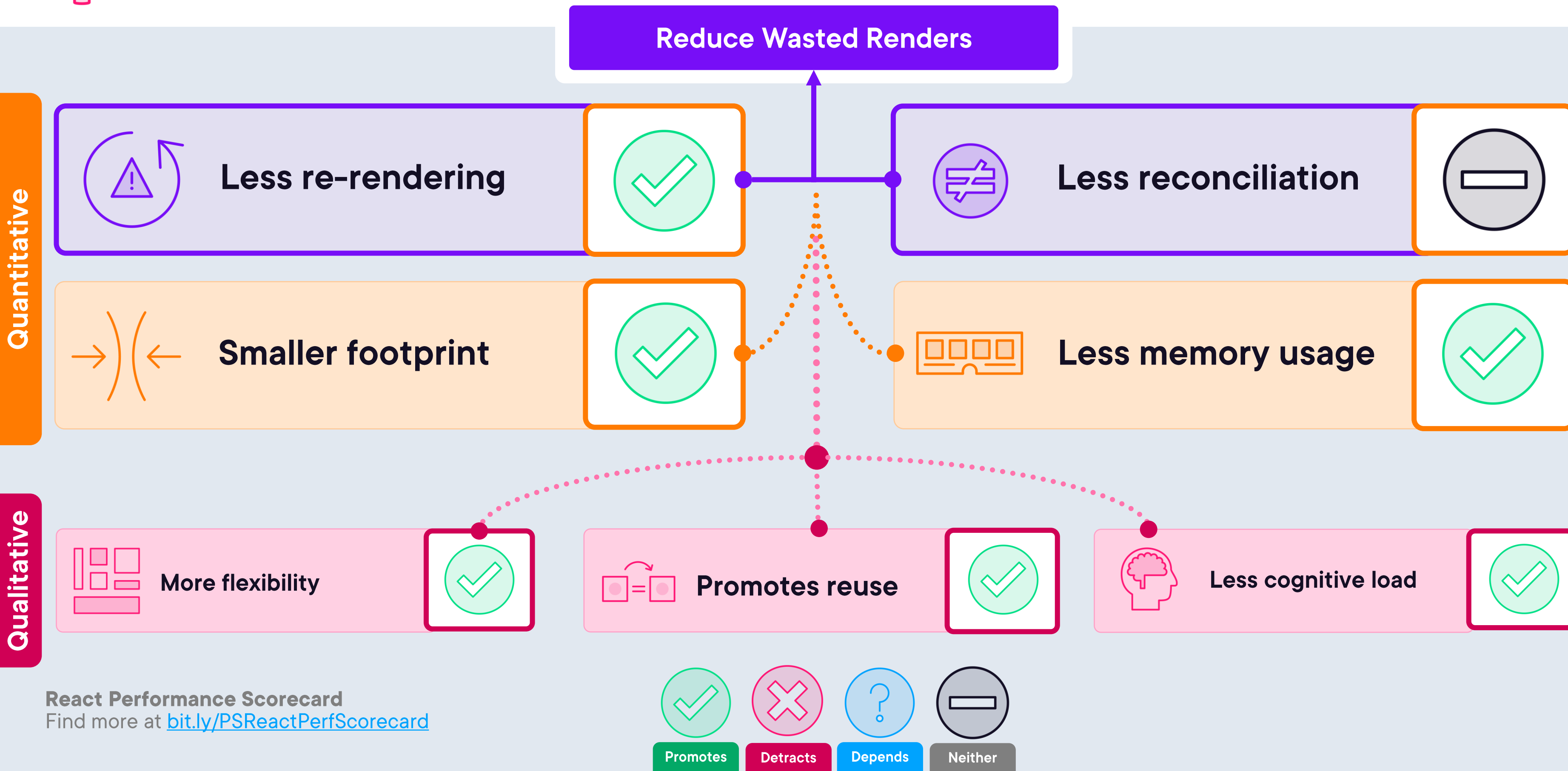




Flux Reducers

Component Design Scorecard

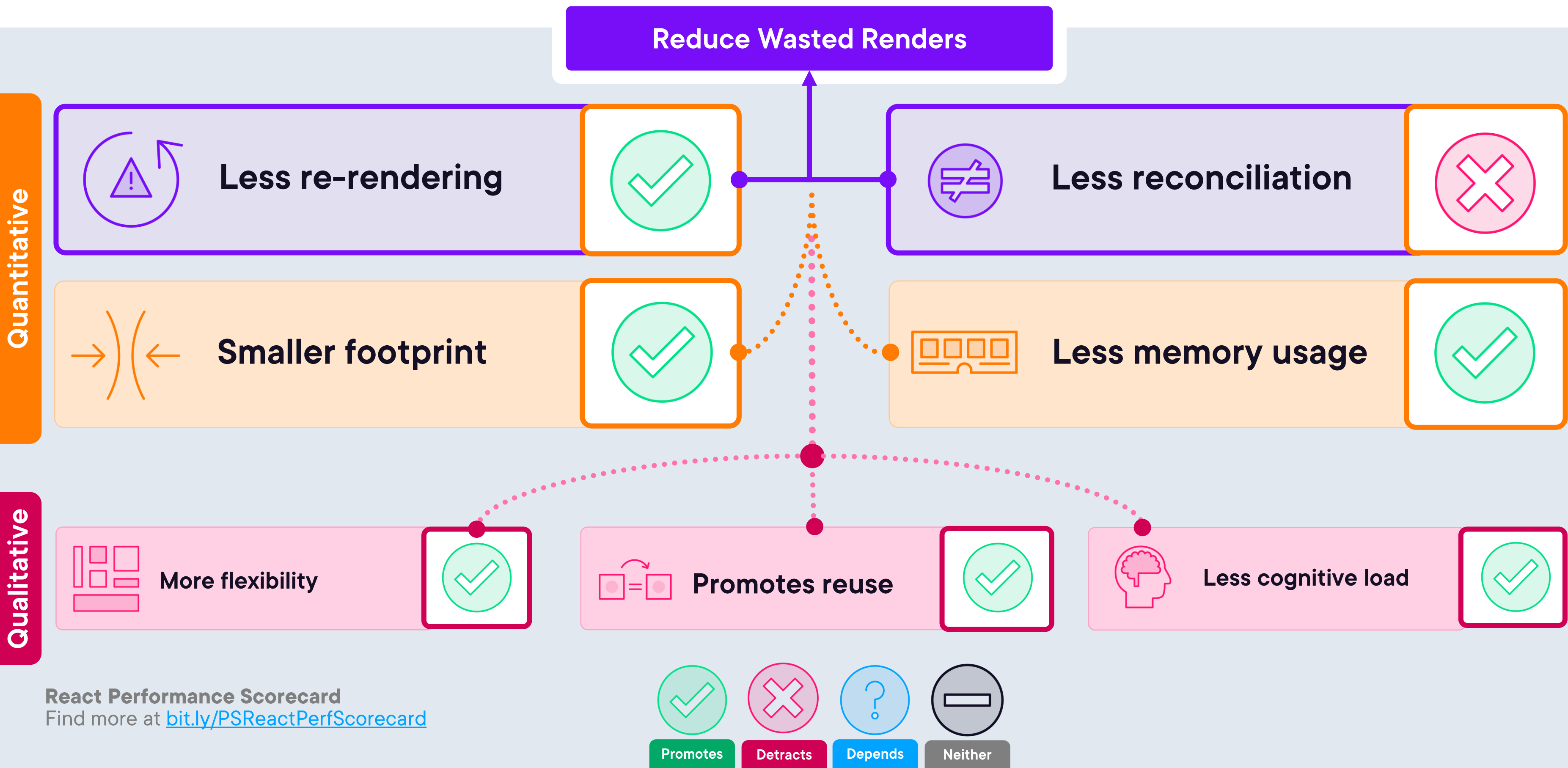
Consolidate complex local state using the Flux data flow pattern

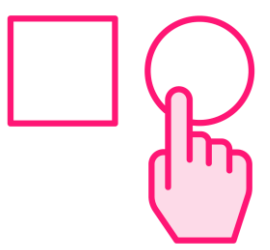




State Management Solutions

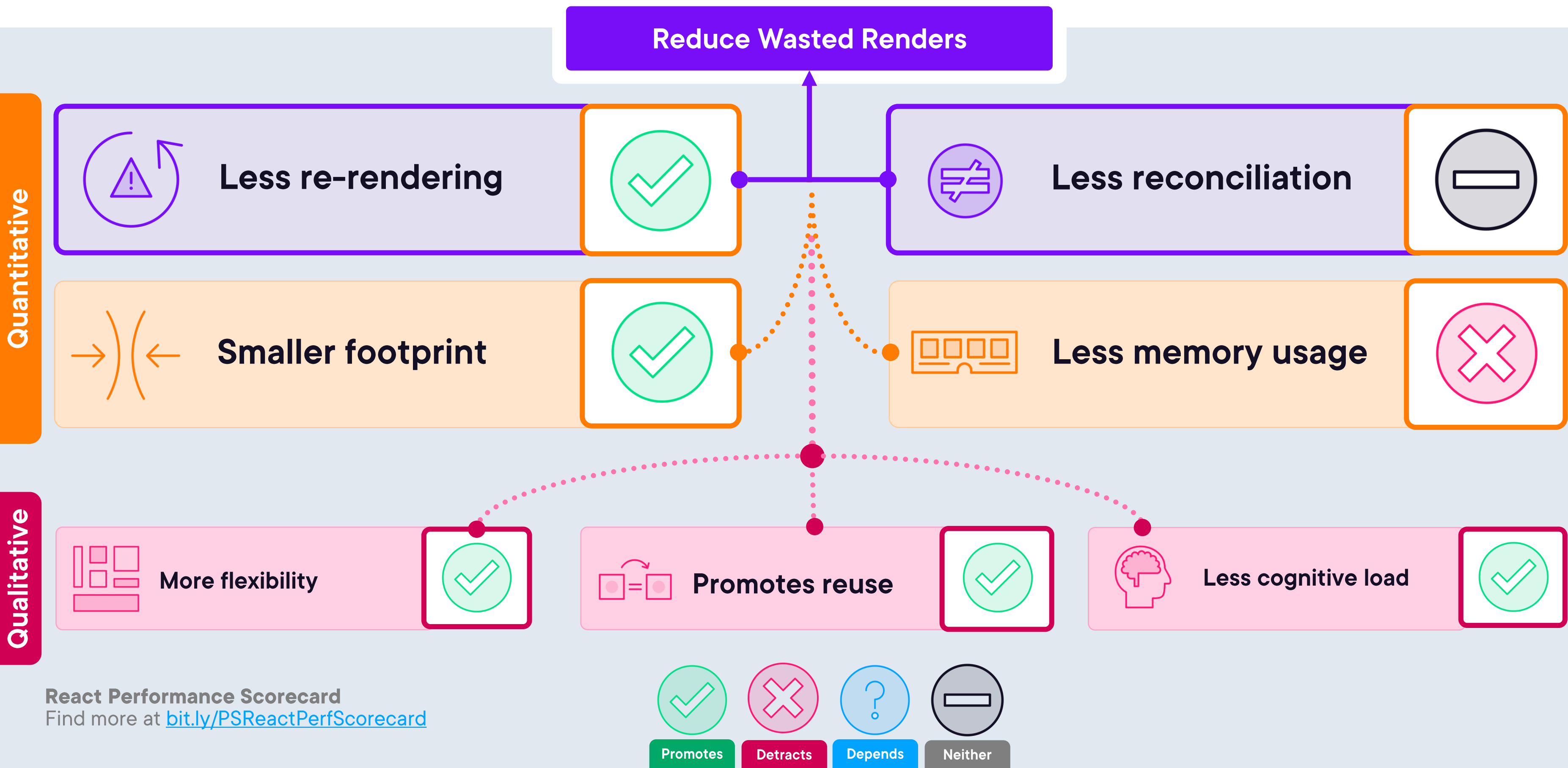
Share data across an app to help simplify complex UIs

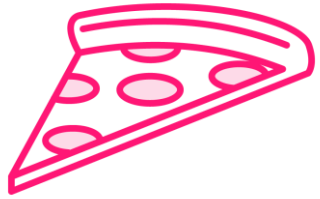




State Selectors

A pure function that can transform global state for local usage

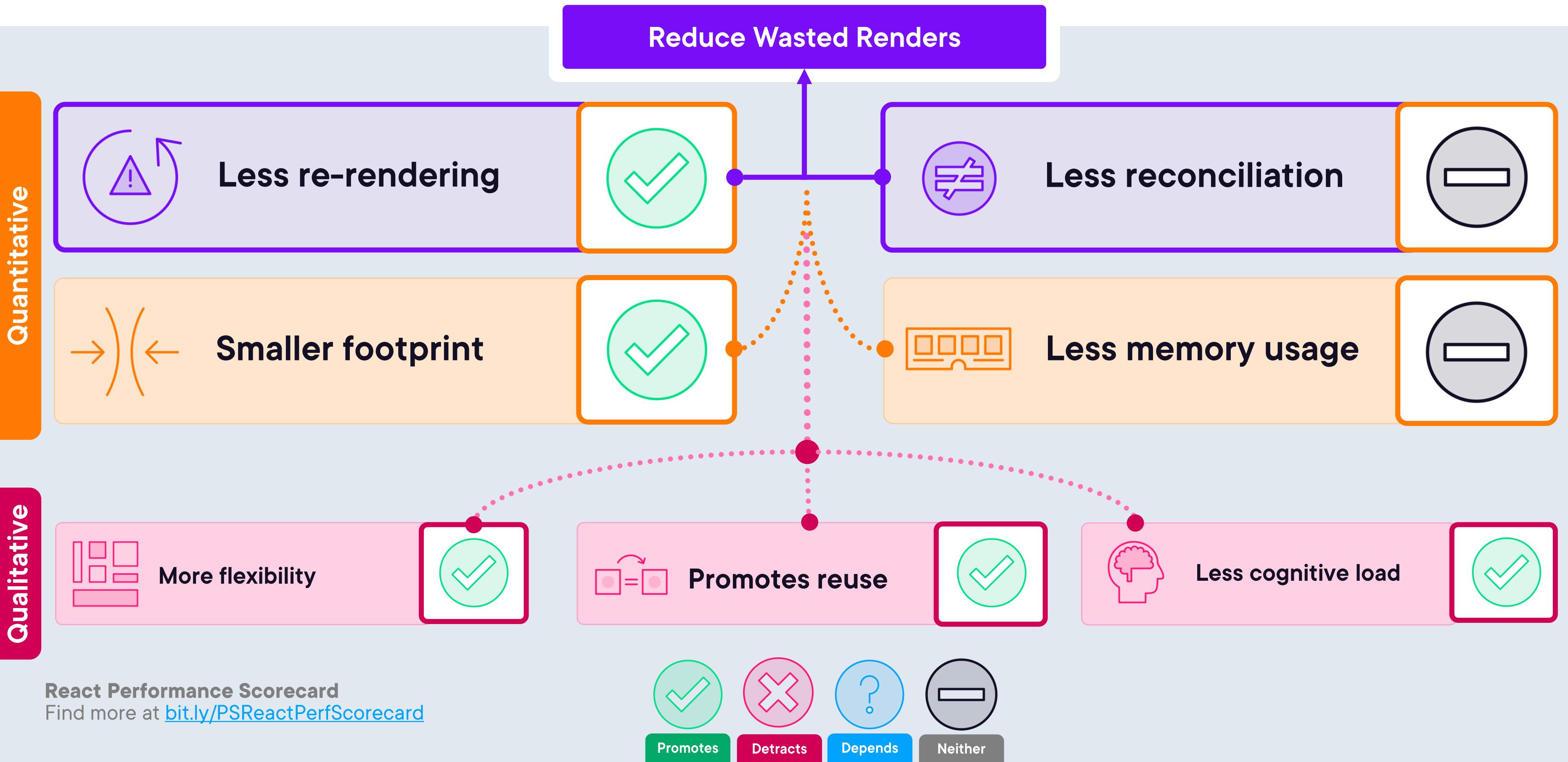


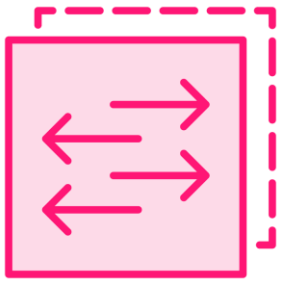


State Slicing

Component Design Scorecard

Structure data by tailoring it to reduce data lookup and transformation costs





Immutable State Helpers

Make working with immutable state easier and more consistent by "drafting" state

