

# Quantifying React Best Practices

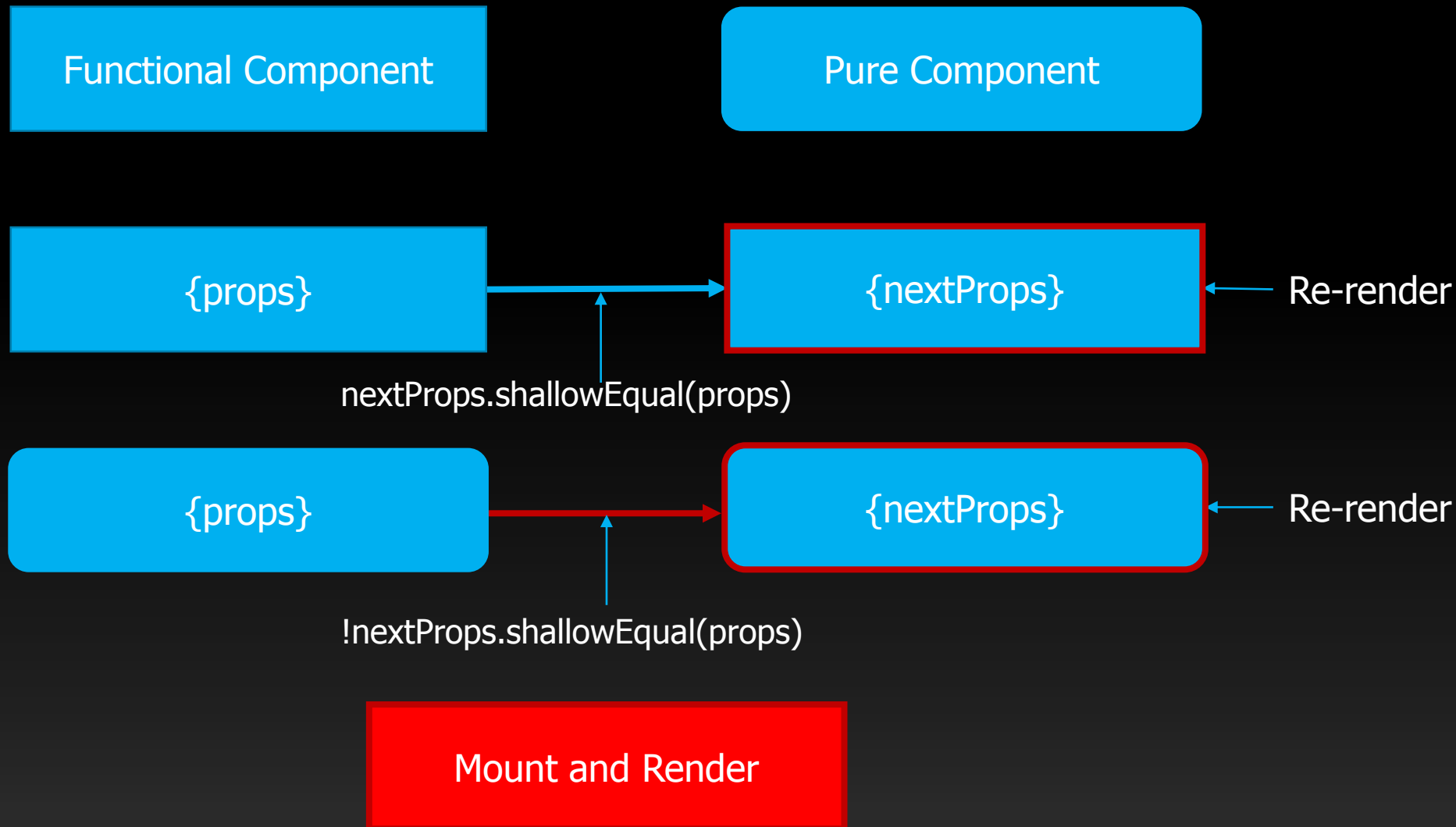
Louis Antonelli and Jackson Hamburger



# Why?

- React fundamentals supersede last minute optimizations
- Excessive rendering is the cause of the vast majority of react performance issues

# Legend



# The “key” Prop

❌ ▶ Warning: Each child in an array or iterator should have a unique "key" prop. Check the render method of `ShiftTest`.  
See <https://fb.me/react-warning-keys> for more information.  
in PureLabel (created by ShiftTest)  
in ShiftTest

Me: I'm not really sure what the 'key' does. I should really learn about it

Me to me: Just add it blindly and move on



# The “key” Prop – Prepending to an Array

*Pure Components*

## Key

name : 1 – key : 1

name : 2 – key : 2

name : 3 – key : 3

name : 4 – key : 4

name : 5 – key : 5

## No Key

name : 1

name : 2

name : 3

name : 4

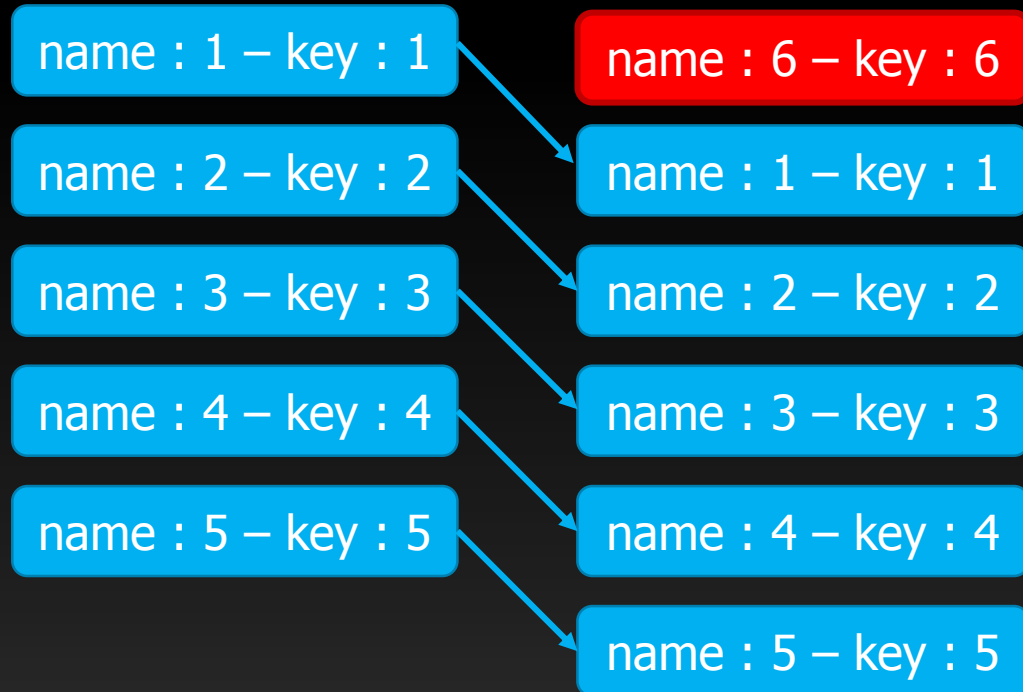
name : 5

# The “key” Prop – Prepending to an Array

*Pure Components*

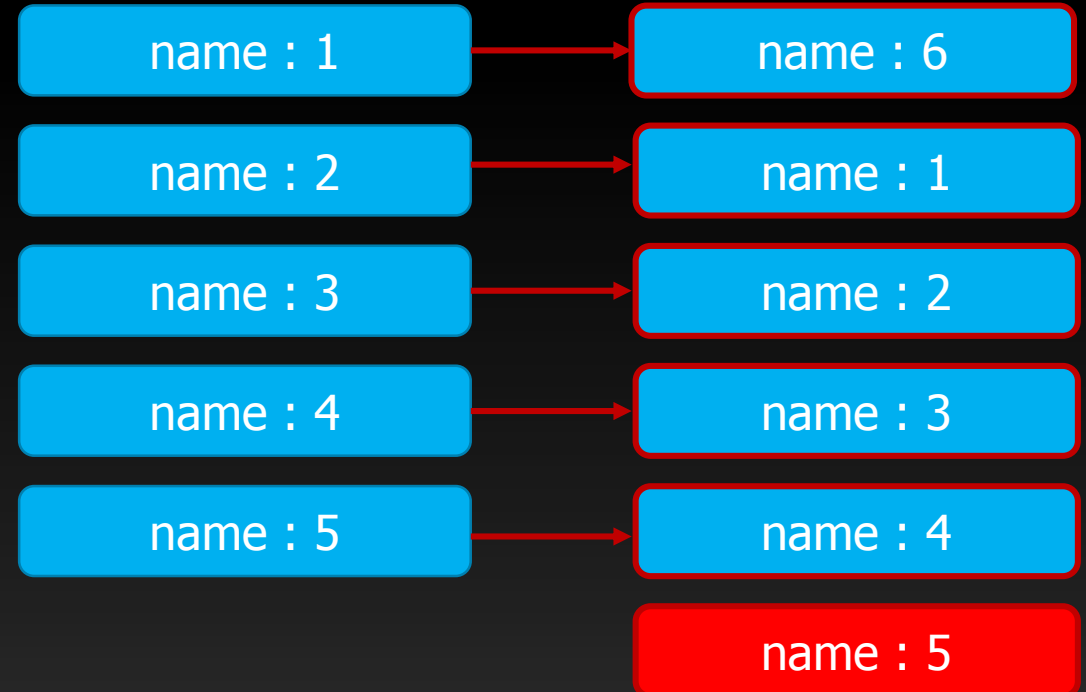
## Key

~10.3 milliseconds



## No Key

~60.3 milliseconds



# The “key” Prop – Prepending to an Array

## *Functional Components*

### Key

name : 1 – key : 1

name : 2 – key : 2

name : 3 – key : 3

name : 4 – key : 4

name : 5 – key : 5

### No Key

name : 1

name : 2

name : 3

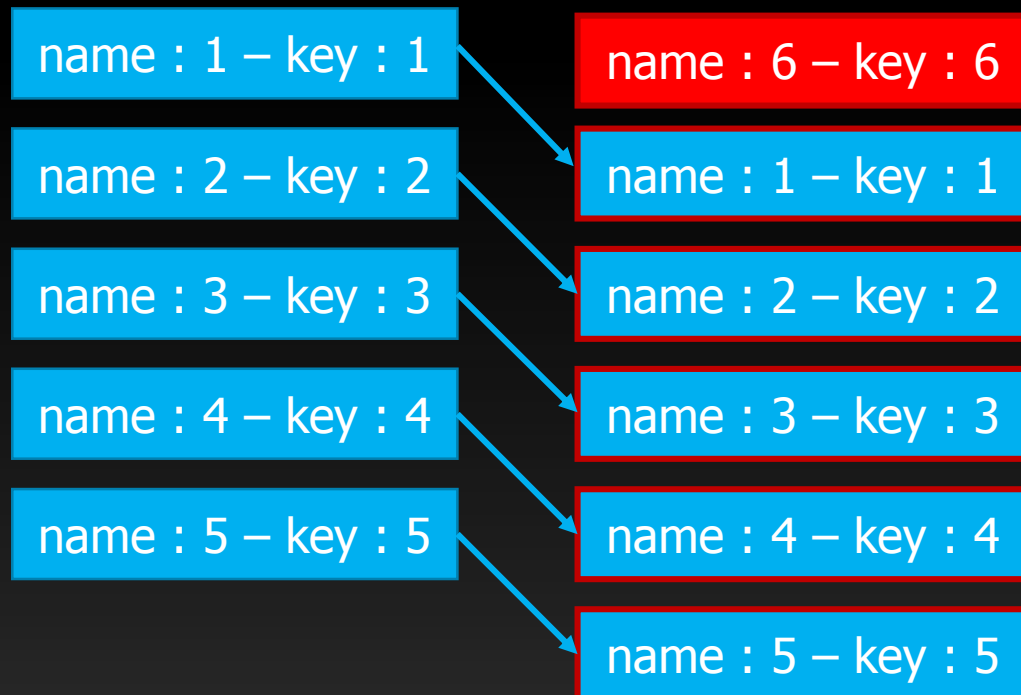
name : 4

name : 5

# The “key” Prop – Prepending to an Array

## *Functional Components*

**Key**  
~60.1 ms



**No Key**  
~60.3 ms





# The “key” Prop – Swapping Array elements

## *Pure Components*

### Key

name : 1 – key : 1

name : 2 – key : 2

name : 3 – key : 3

name : 4 – key : 4

name : 5 – key : 5

### No Key

name : 1

name : 2

name : 3

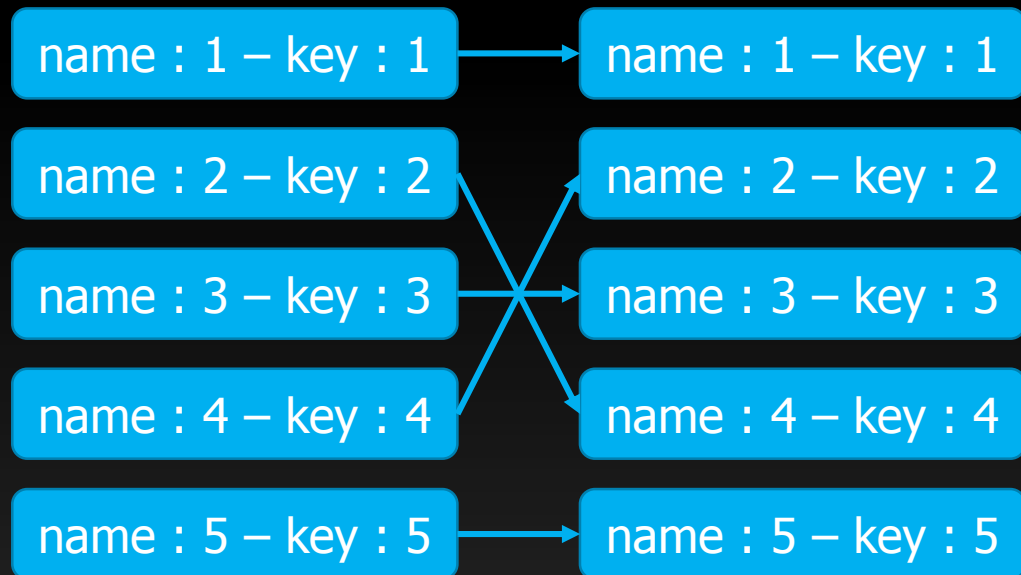
name : 4

name : 5

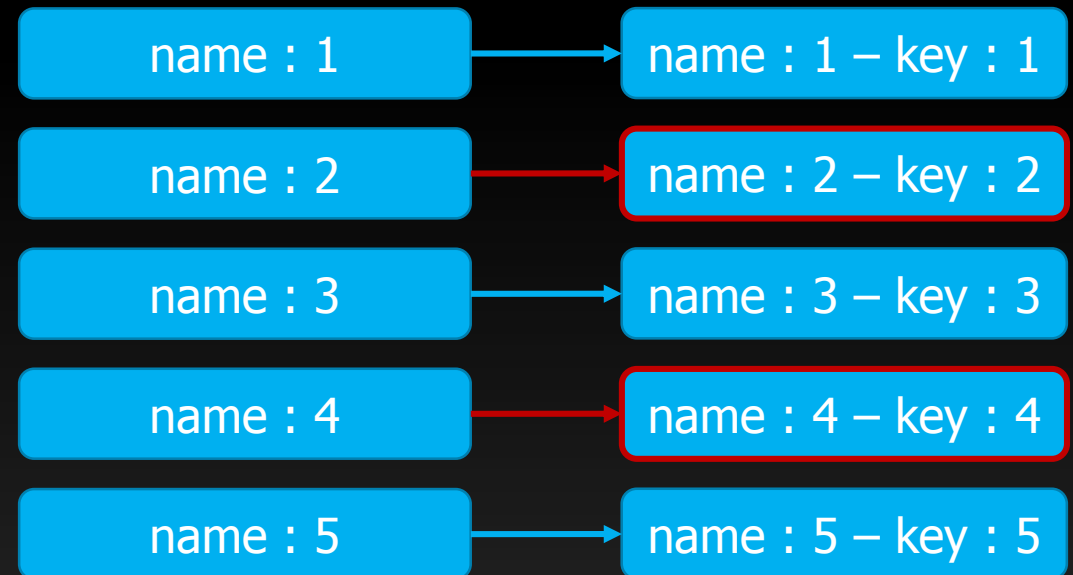
# The “key” Prop – Swapping Array elements

*Pure Components*

**Key**  
~1.3 ms



**No Key**  
~19.9 ms



# The “key” Prop – Swapping Array elements

## *Functional Components*

### Key

name : 1 – key : 1

name : 2 – key : 2

name : 3 – key : 3

name : 4 – key : 4

name : 5 – key : 5

### No Key

name : 1

name : 2

name : 3

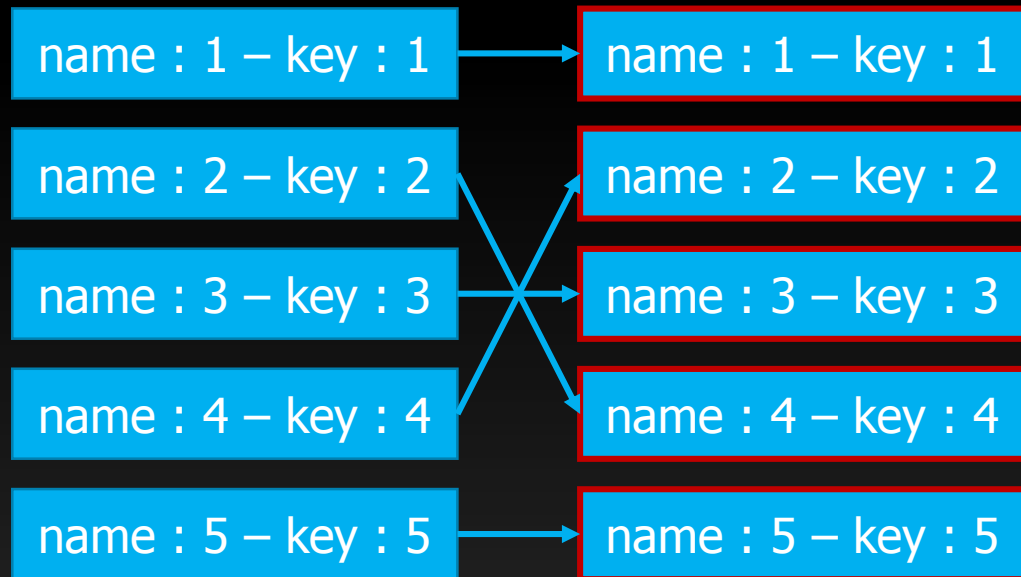
name : 4

name : 5

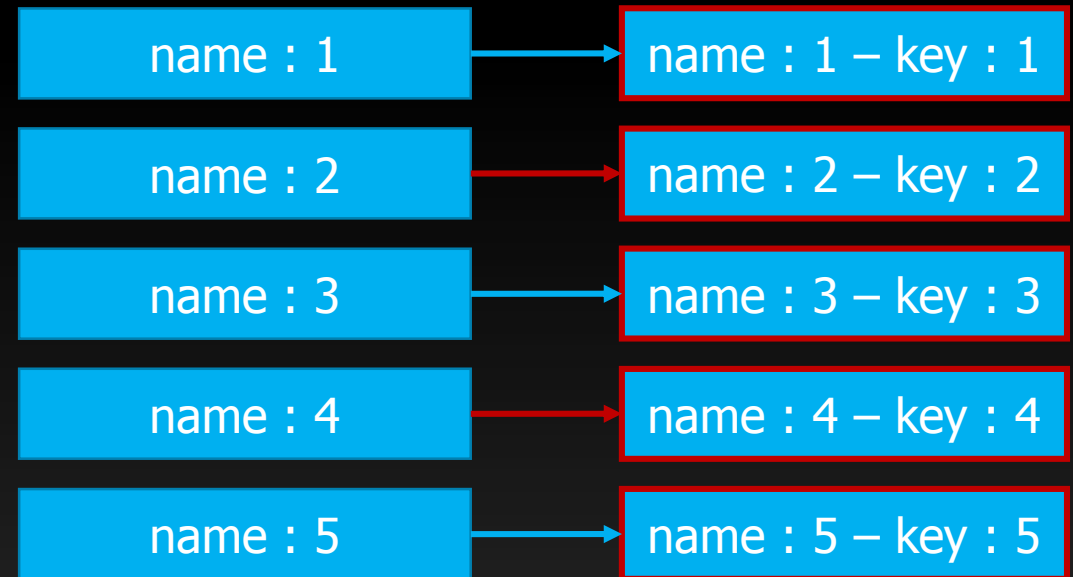
# The “key” Prop – Swapping Array elements

## *Functional Components*

**Key**  
~50.4 ms



**No Key**  
~50.0 ms



# Takeaways

- Keyed Functional components will always render at least as many times as Unkeyed Pure Components in an array
- You get a warning for keys but functional components are more damaging
- Important given the prevalence of lists and list mutation in apps
- Know when your components render!