Counter Application Data Flow

1. Introduction

This document provides a detailed Data Flow Diagram (DFD) representation of the Counter React application.

It explains the process of state management, user interaction, and UI updates.

2. System Overview

The React application consists of the following key components:

- App.js: The main file that initializes the counter state and handles UI updates.
- useState Hook: Manages the counter state.
- UI Components: Buttons for incrementing and decrementing the counter.

3. Data Flow Diagram (DFD)

Level 0 (Context Diagram)

At the highest level, the system consists of external users interacting with a simple counter UI.

External Entities:

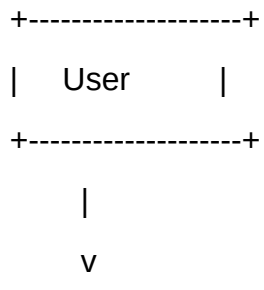
- User: Clicks buttons to increment or decrement the counter.

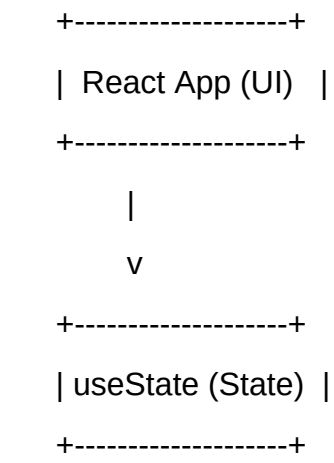
Processes:

- React Application: Updates the state and re-renders the UI.

Data Stores:

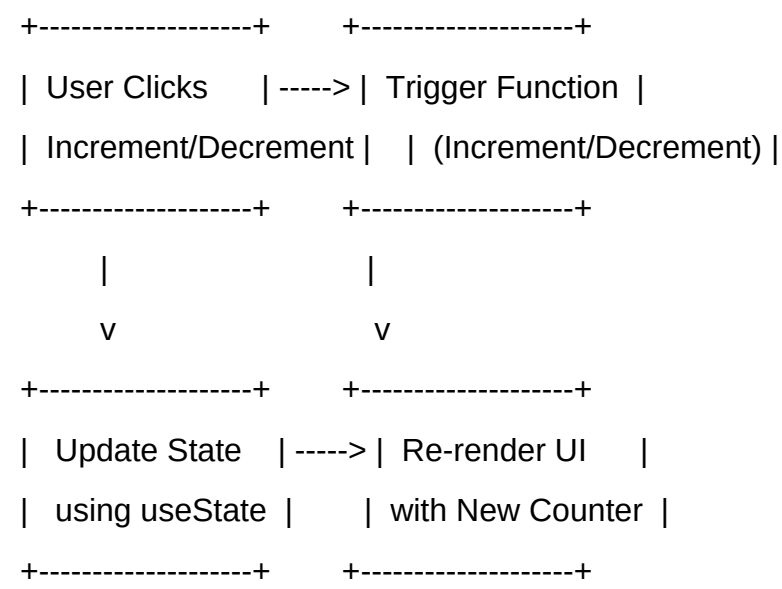
- useState Hook: Stores the current counter value.





Level 1 DFD (State Update Process)

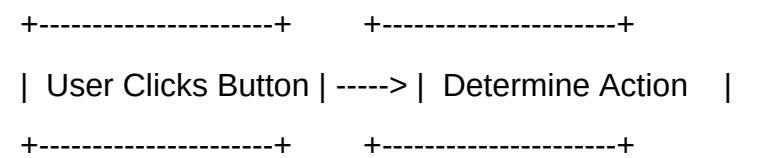
This level breaks down how state is updated when the user interacts with the counter.

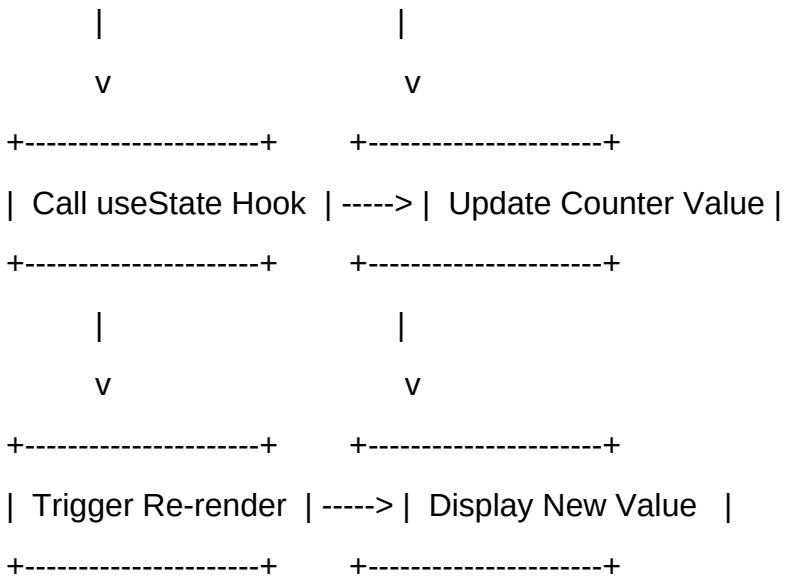


Process Explanation:

1. The user clicks an increment or decrement button.
2. The corresponding function modifies the state using `useState`.
3. React detects the state change and re-renders the UI with the updated counter value.

Level 2 DFD (Detailed Counter Management)





4. Explanation of Data Flow

1. The user interacts with the counter UI by clicking buttons.

2. The event handlers trigger functions that modify the counter state.

3. The `useState` hook updates the state, triggering a re-render.

4. The UI updates dynamically to display the new counter value.

5. Conclusion

This document outlines the detailed data flow in a React-based counter application.

The application effectively manages state updates and dynamic UI rendering using `useState`.