Introduction to React and the DOM

This presentation covers React and the Document Object Model (DOM).

Understand key concepts building efficient web interfaces.

BN **by bismis nomrita**

What is React?

JavaScript Library

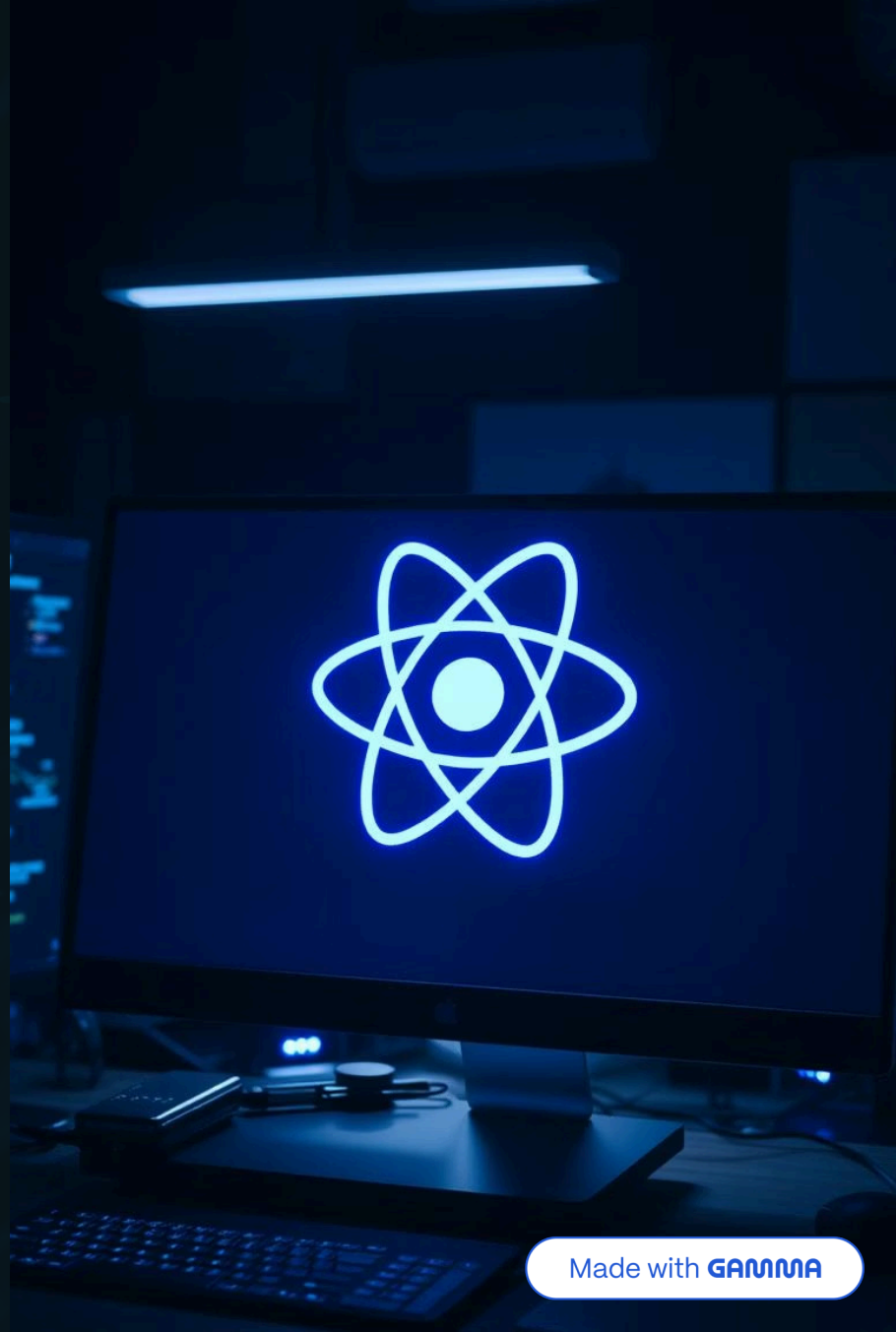
For building user interfaces with component-based architecture.

Backed by Facebook

Reliable and widely supported by the developer community.

Popular Usage

Used by Netflix, Instagram, Airbnb; 44% developer adoption (2023).



Understanding the DOM

Definition

HTML elements represented as a tree of nodes and objects.

Rendering

Browsers use DOM to construct and display web pages.

Drawbacks

Direct manipulation is inefficient and triggers full re-renders.

Virtual DOM

In-memory Representation

Lightweight copy of the real DOM for fast comparison.

Performance

Batches updates to minimize expensive DOM operations.

No Direct DOM Manipulation

Updates are applied via Virtual DOM for efficiency.

React's Reconciliation Process

Diffing

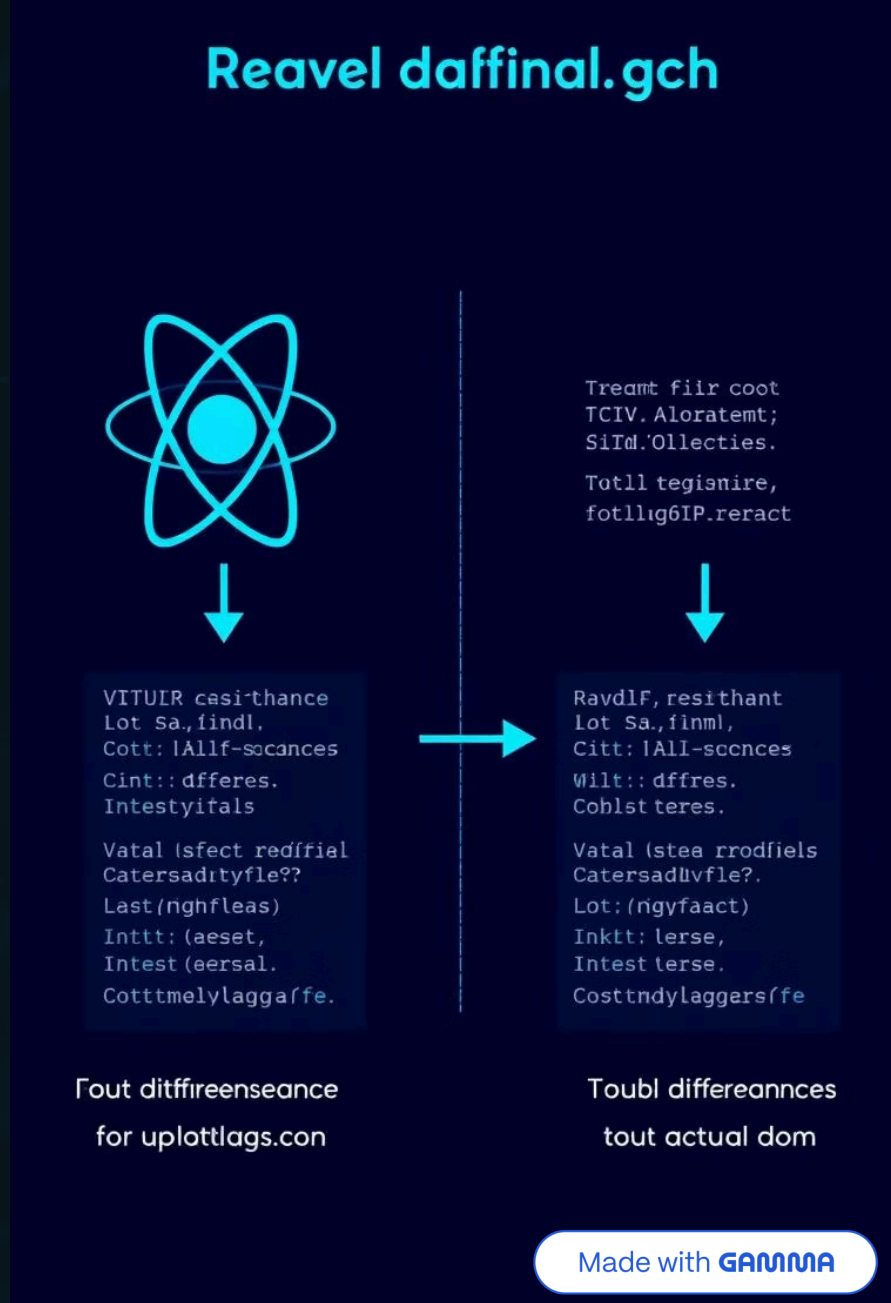
Compares new Virtual DOM with previous version.

Minimal Changes

Identifies only areas that need updates.

Fast Updates

Minimizes real DOM changes for better speed.



Benefits of the Virtual DOM

Enhanced Performance

Speeds up user interfaces by reducing costly DOM work.

Simplified Development

Allows developers to focus on UI logic without manual DOM updates.

Improved UX

Smoother interactions and quicker response times.

React Components

Reusable Blocks

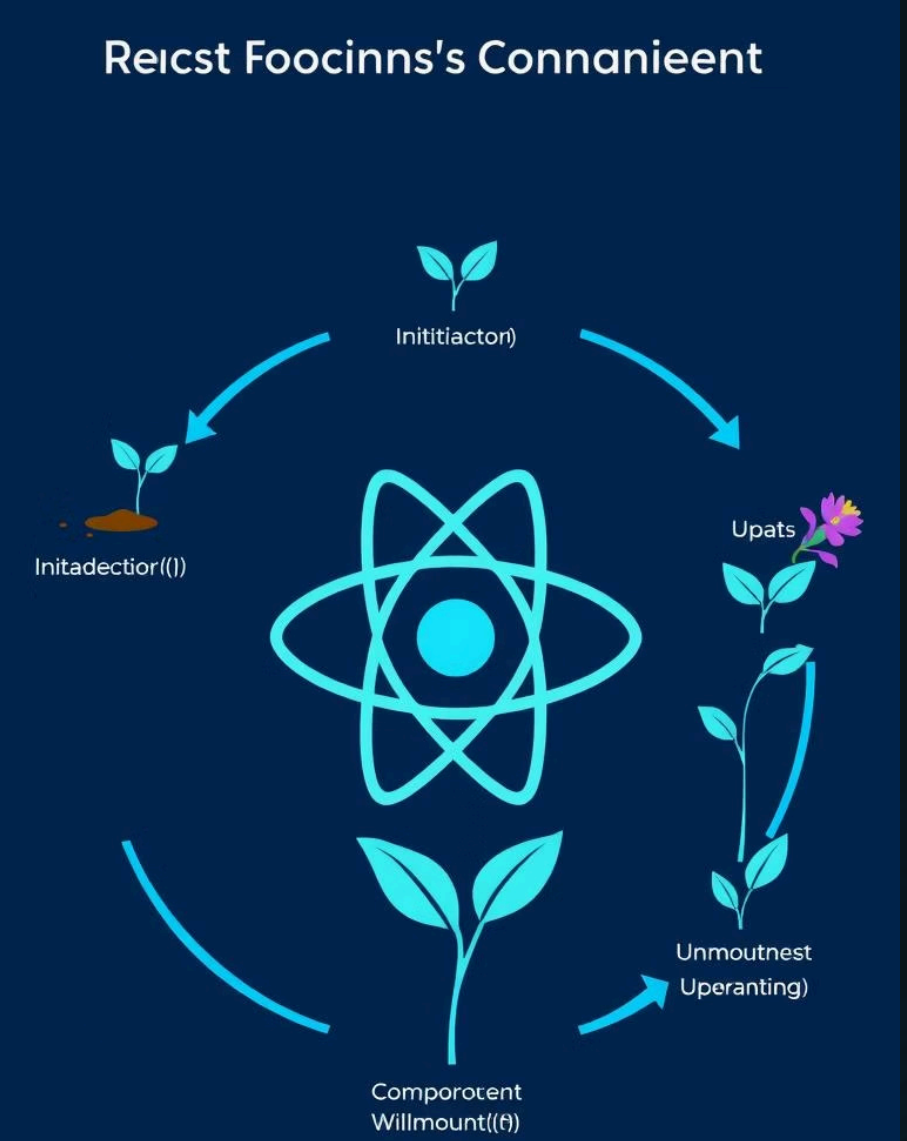
Components encapsulate UI code and behavior.

Stateful or Stateless

Manage their own state for dynamic interfaces.

Example Components

Button, InputField, Card serve as common UI parts.



Component Lifecycle

1

Mounting

Component creation and insertion into the DOM.

2

Updating

Handling changes via lifecycle methods.

3

Unmounting

Cleanup before component removal to save memory.

React and the Real DOM

Render to Virtual DOM

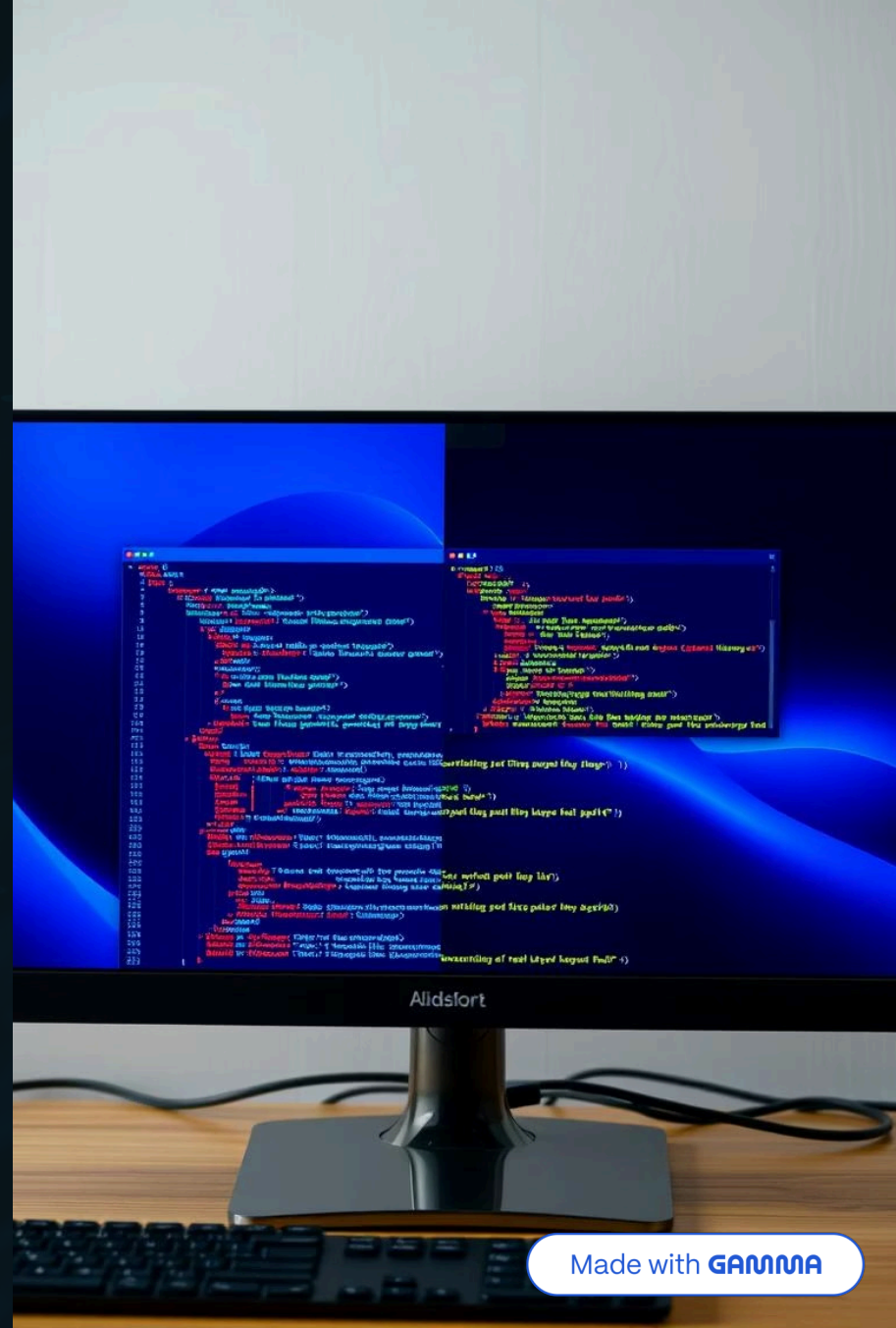
React components update an in-memory representation.

Compare and Update

Only changed parts update the real DOM efficiently.

Declarative UI

Developers write what UI should look like, React handles changes.



Conclusion

Efficient Updates

Virtual DOM optimizes rendering speed and performance.

Modular Code

Components enable reusable, manageable UI development.

Improved UX

Better performance leads to smoother user experiences.