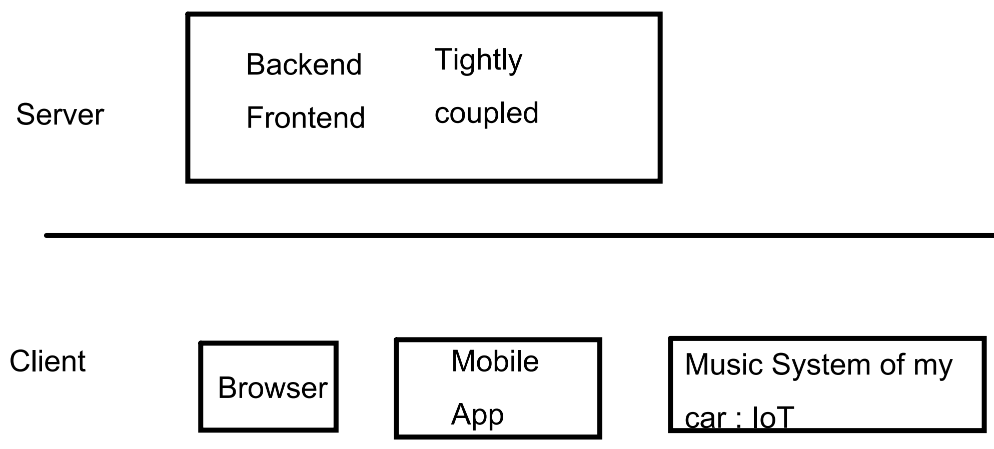
ReactJS : JS lib for building UI

Dedicated,focus, expert lib to create Frontend app

Traditionally



Decouple the backend -& frontend

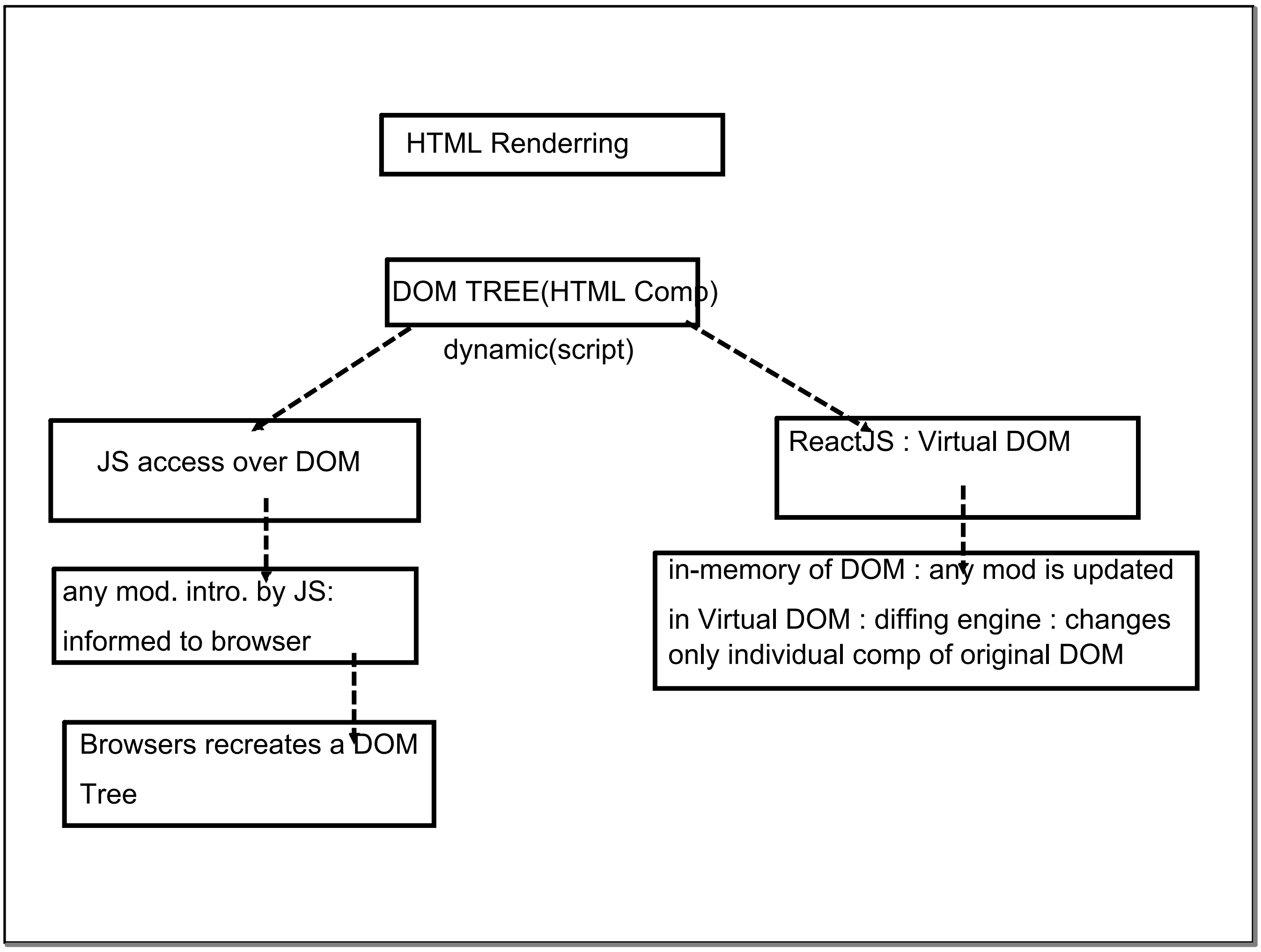
Performance enhancements

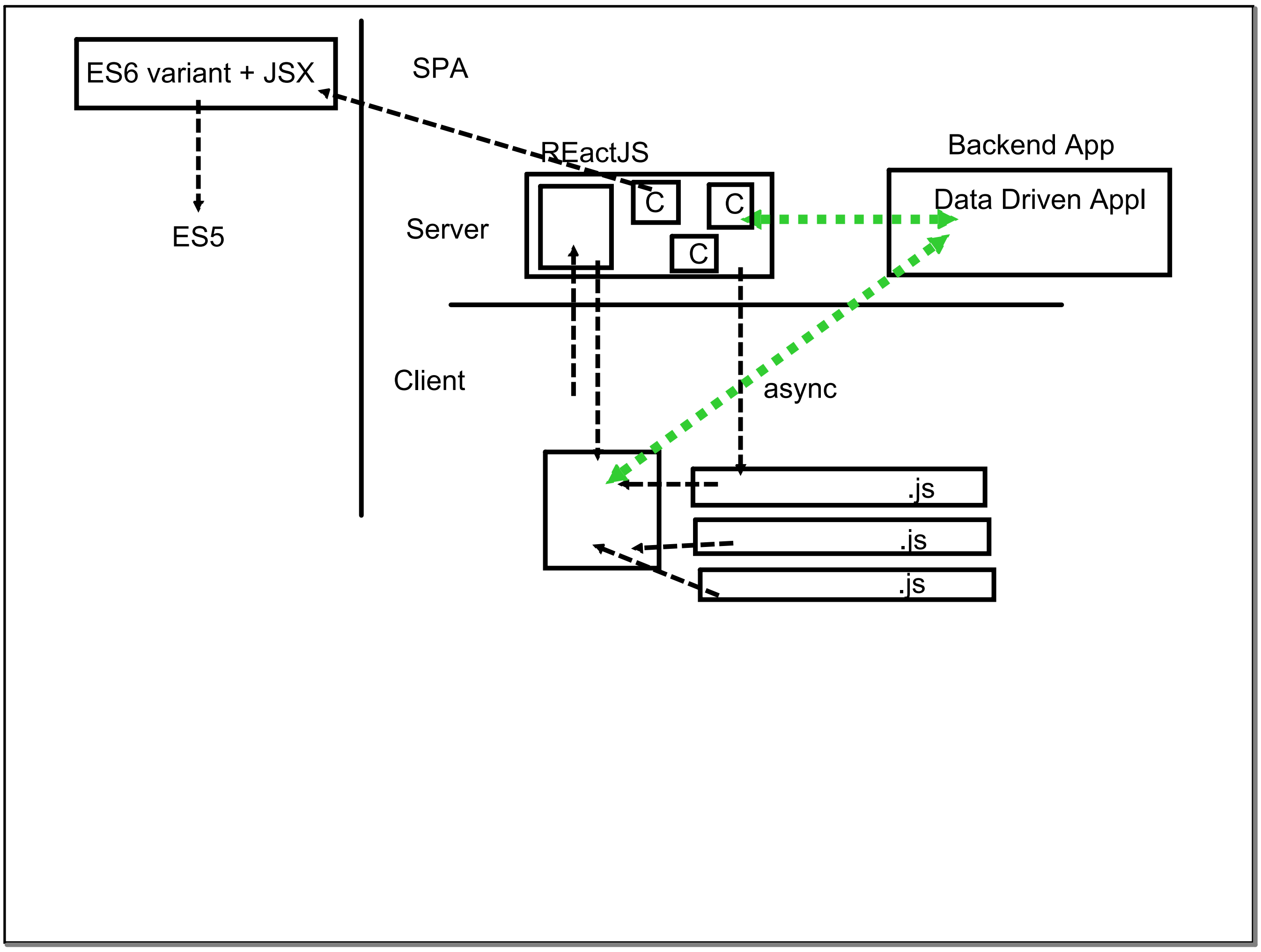
Angular : MVC architecture

ReactJS : V

build large app : data changes frequently with time

DOM Tree : Recreats the DOM tree on every change : Rerenderring


HTML Renderring
DOM TREE(HTML Comp)
dynamic(script)
JS access over DOM
any mod. intro. by JS: informed to browser
Browsers recreates a DOM Tree
ReactJS : Virtual DOM
in-memory of DOM : any mod is updated in Virtual DOM : diffing engine : changes only individual comp of original DOM


ES6 variant + JSX
ES5
SPA
REactJS
C
C
C
Backend App
Data Driven Appl
Server
Client
async
.js
.js
.js

Resource Management

complete React JS library is in single file

react.js(vesrion)

Additional lib :

react-dom.js :(virtual DOM)

JS

NodeJS : npm tool (node package manager)

installing new frameworks,api, cli, tool

structure, config

npm initiate any JS

Manual :

Download all lib/api

structure code

add and define config

Auto : Tool by Facebook

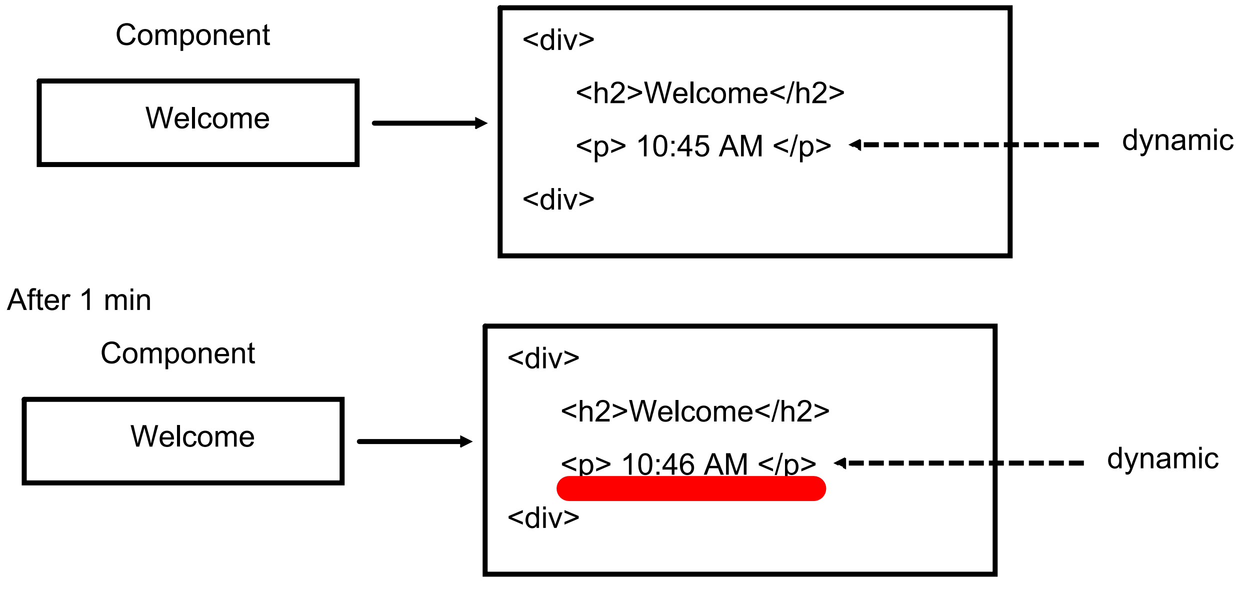
create-react-app (cli)

```
>npm i -g <tool-name>
>npm i -g create-react-app
```

Component based programming : create reusable HTML components (JS logic)

# custom HTML tag

# Javascript function : generates output every time it is invoked (managed & modelled by Virtual DOM)



Components :

1. Traditional JS : functions
2. ES6 std + JSX (create a class)

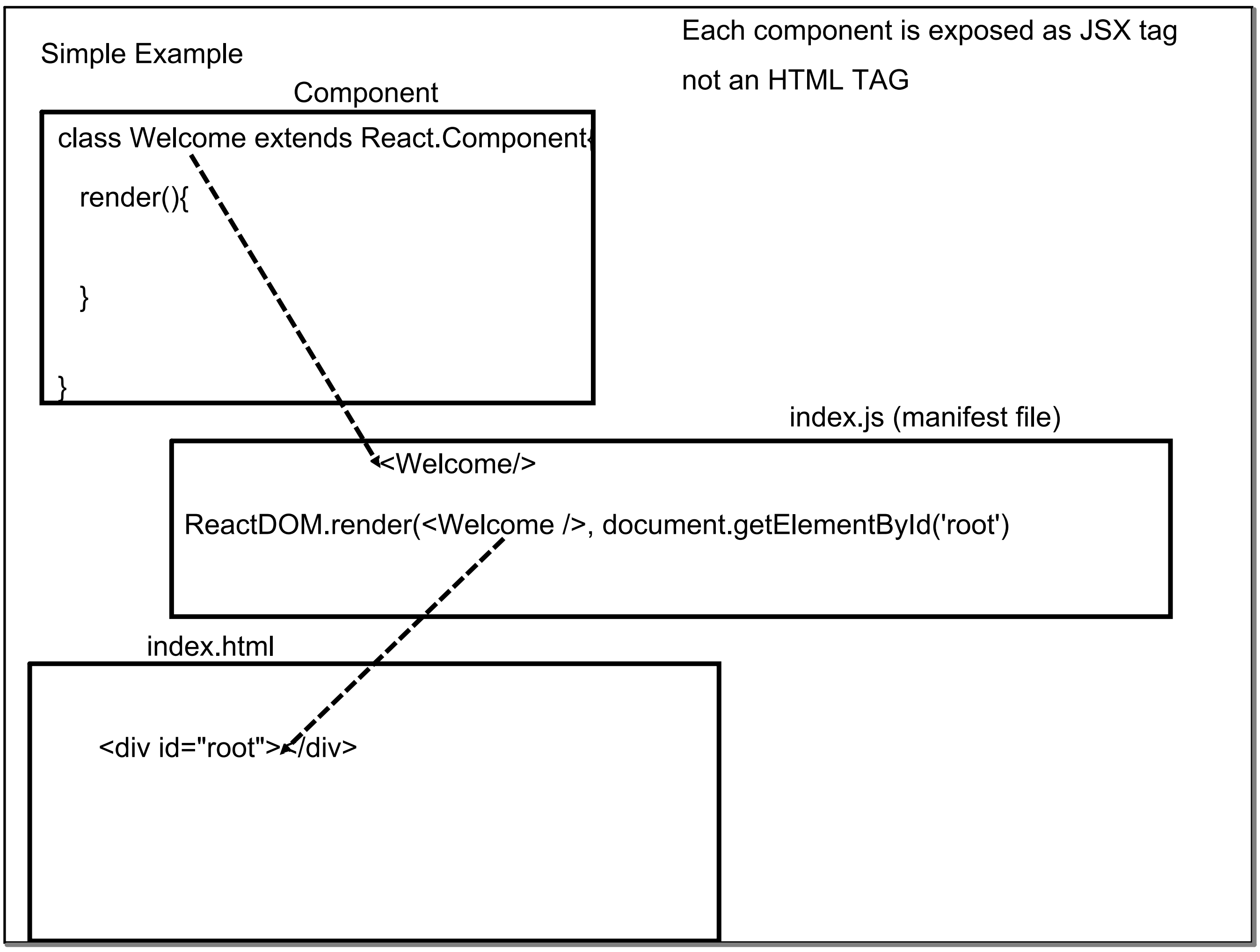
Create a React Application

```
>create-react-app <project-name>
```

create a JS class and inherit class React.Component

In package.json : add new lib

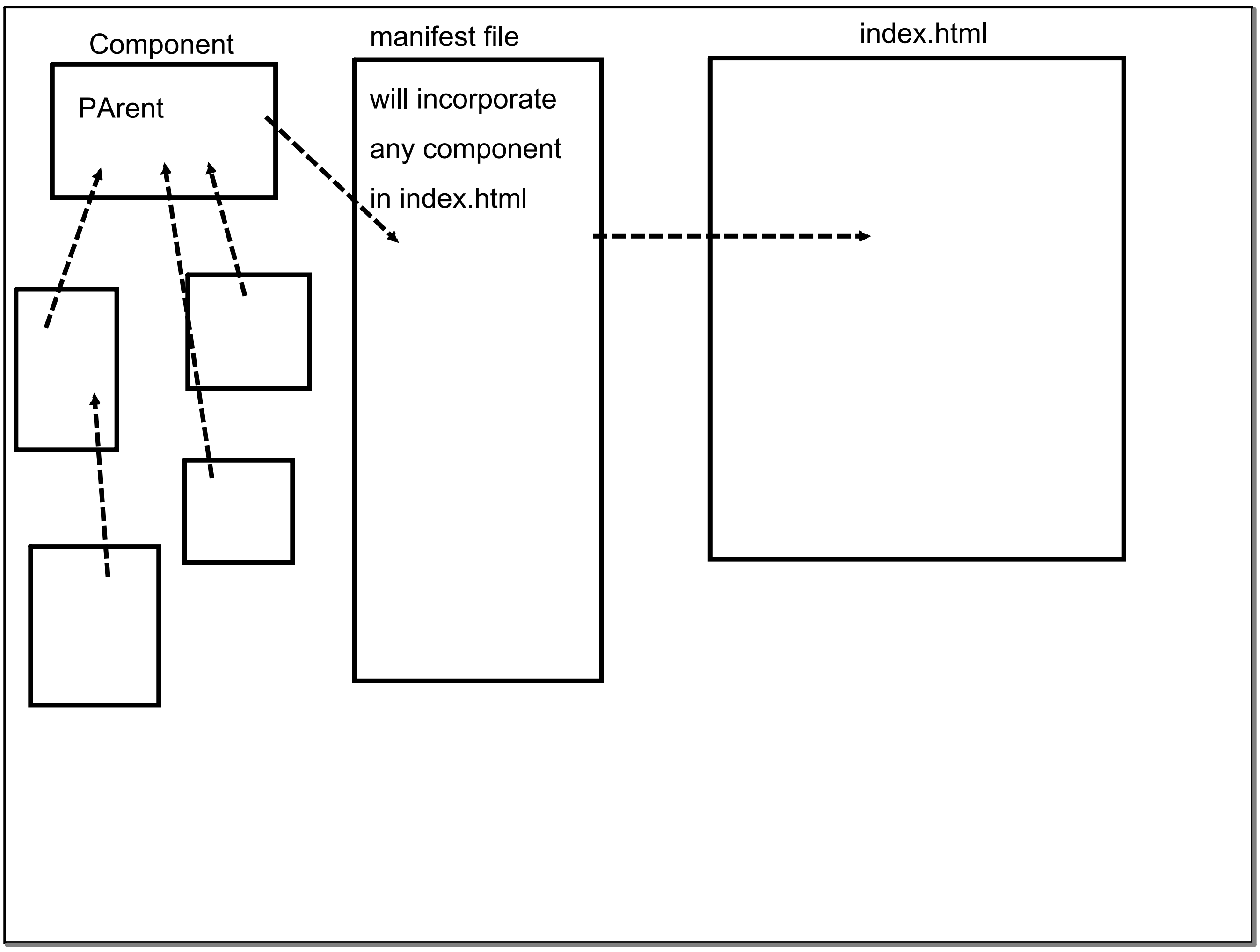
```
>npm install
```

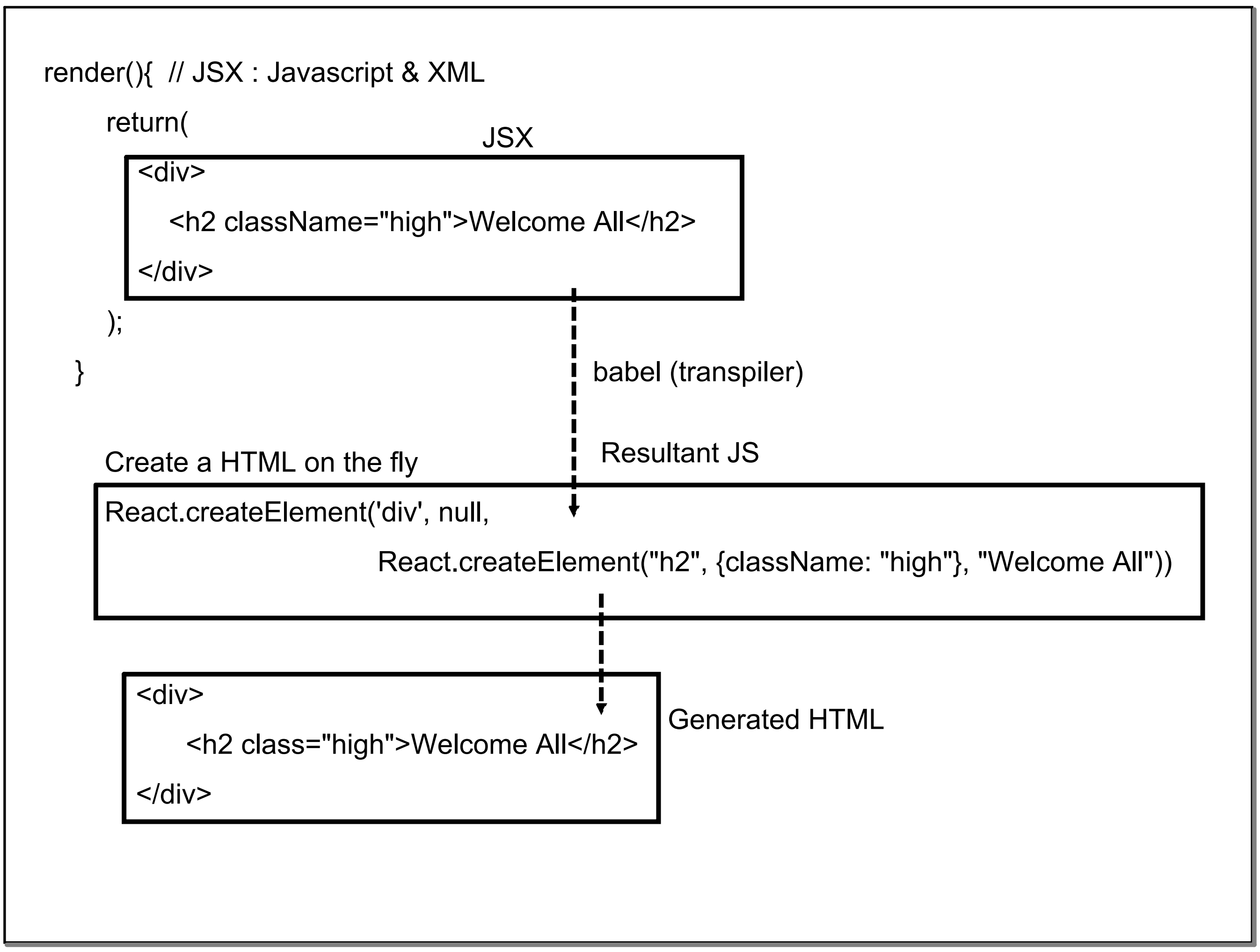


```
class Welcome extends React.Component{
   render(){

   }

}
```

```
<Welcome/>

ReactDOM.render(<Welcome />, document.getElementById('root')
```

```
<div id="root"></div>
```


Component
PArent
manifest file
will incorporate
any component
in index.html
index.html



```
render(){  // JSX : Javascript & XML
      return(
        <div>
           <h2 className="high">Welcome All</h2>
        </div>
      );
   }
```

```
React.createElement('div', null,
                              React.createElement("h2", {className: "high"}, "Welcome All"))
```

```
<div>
     <h2 class="high">Welcome All</h2>
</div>
```

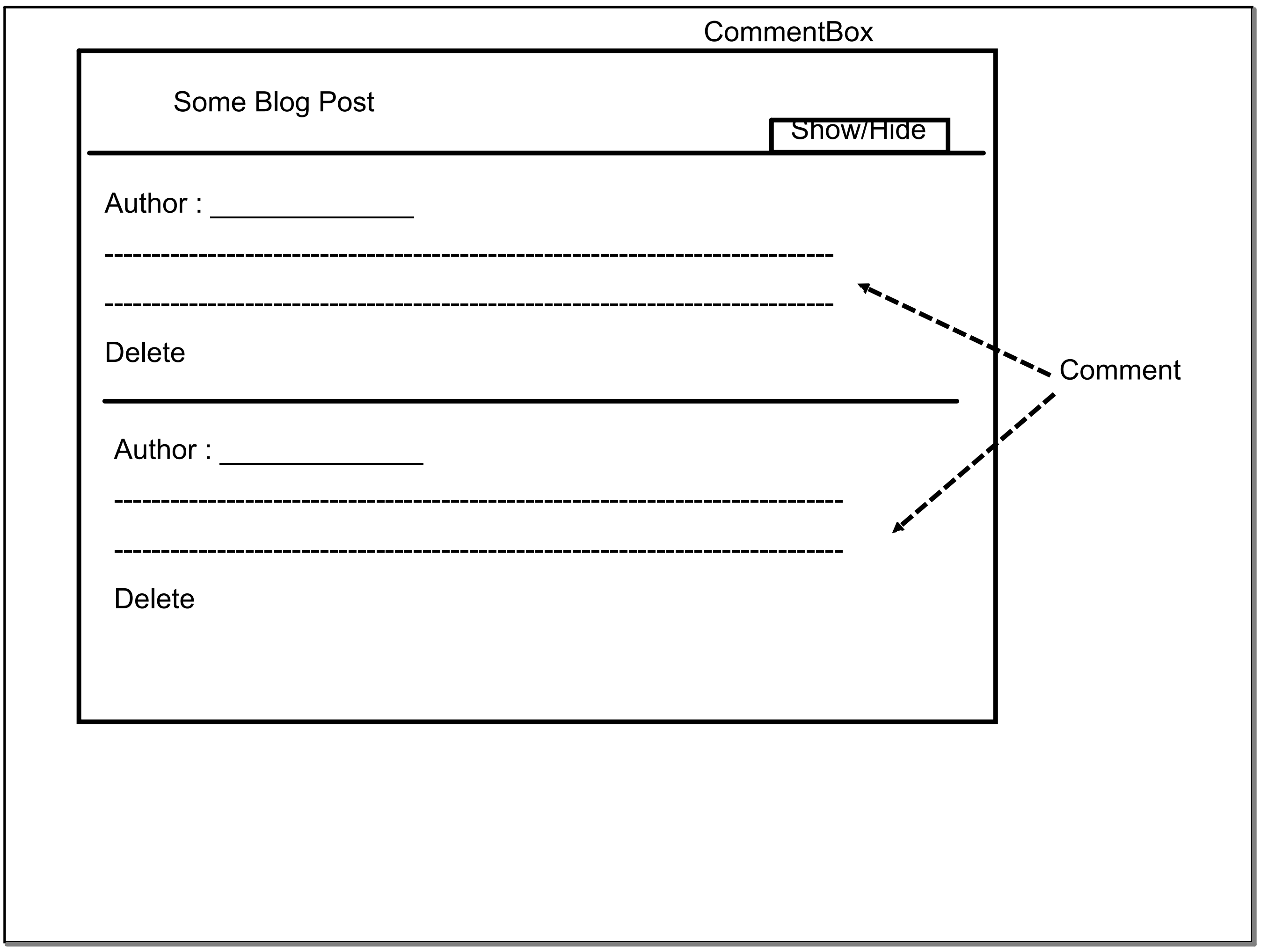
Commenting Engine Use Case

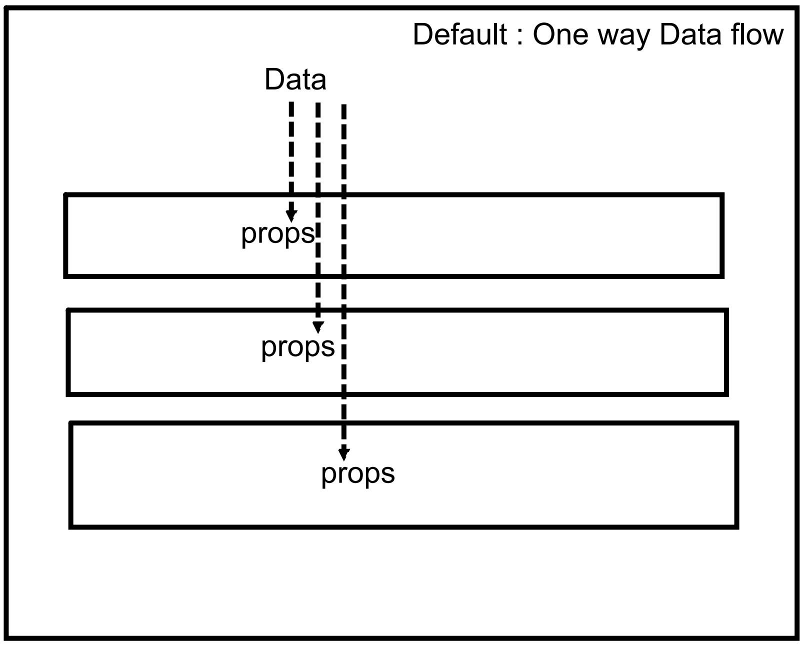
==> Allow visitors to post comments on a blog post

# able to show / hide the comments

# able to add new comments

# delete comments


CommentBox
Some Blog Post
Show/Hide
Author : ______________
Delete
Author : ______________
Delete
Comment



Each React Component has inbuilt object that hold the props (values passed from parent as attributes)

Child/Any Component can define default values for prop

# How to handle the events
# state

Manipulate DOM

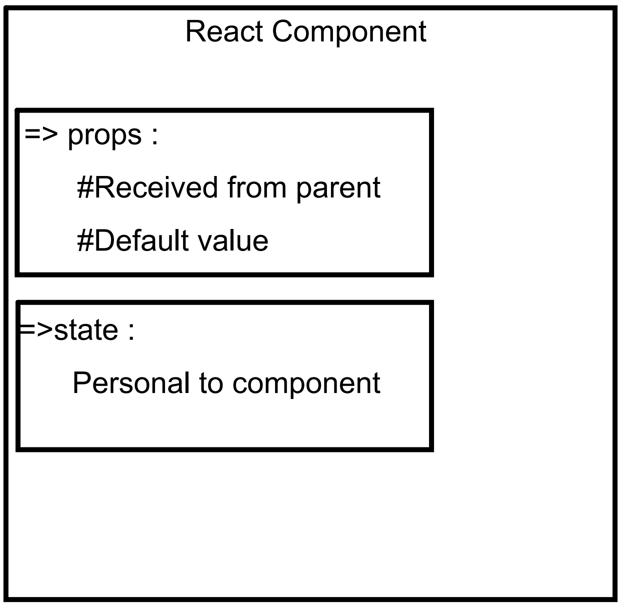
1. Traditional : Direct/Manually DOM Manipulation,(Plain, Jquery)

```
    $('.show_hide_btn').on('click', function(){
        $('.comment-list').show();
    })
    $('.show_hide_btn').on('click', function(){
        $('.comment-list').hide();
    })
```

Complete DOM Tree will be re-renderer

2. Indirect DOM Manipulation:

ReactJS : modify component state object in response to user event and let React handle updates to DOM (Virtual DOM + Diffing Engine)



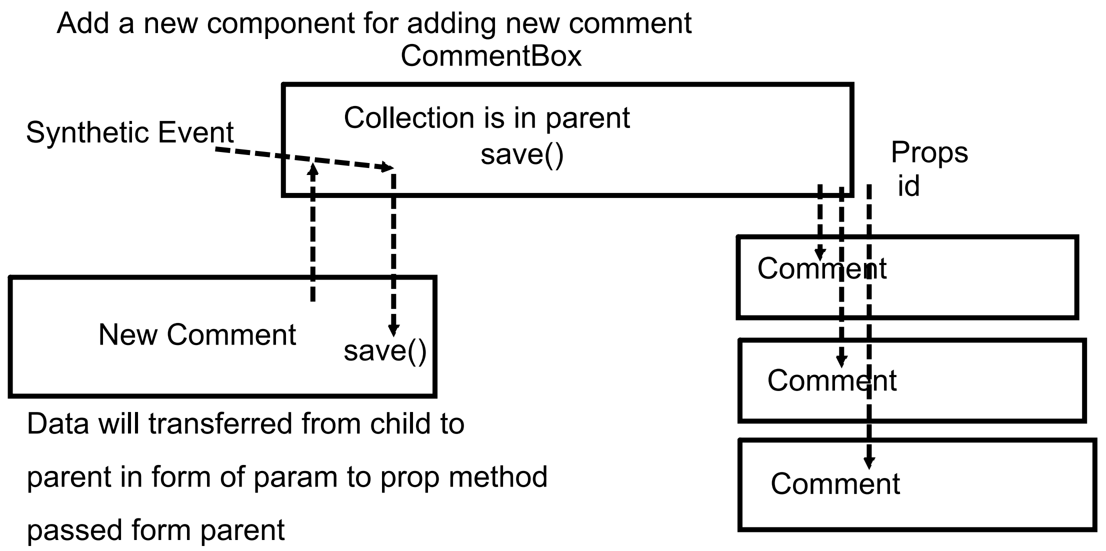
We cannot manually call render method for UI change

React handles the call of render method

When REACT calls render method

# when there is a change in:

1. state object
2. props



Data will transferred from child to parent in form of param to prop method passed form parent ref : attribute that asks for a callback function : function gets called on render

: function will recieve that component as parameter

Making a React App talk with backend server

# Assume a backend server exposing REST endpoint

# All interaction needs to take place async (AJAX)

# need the support JQuery

1. add CDN Link of Jquery in index.html
2. Add Jquery Lib to React

React Component Lifecycle (Hooks/phases)

Hooks methods : callback methods

get called auto as per the phases

Need a provision to continously

poll for new comments from other users

Timer facility will allow to schedule any activity

constructor

componentWillMount() only once

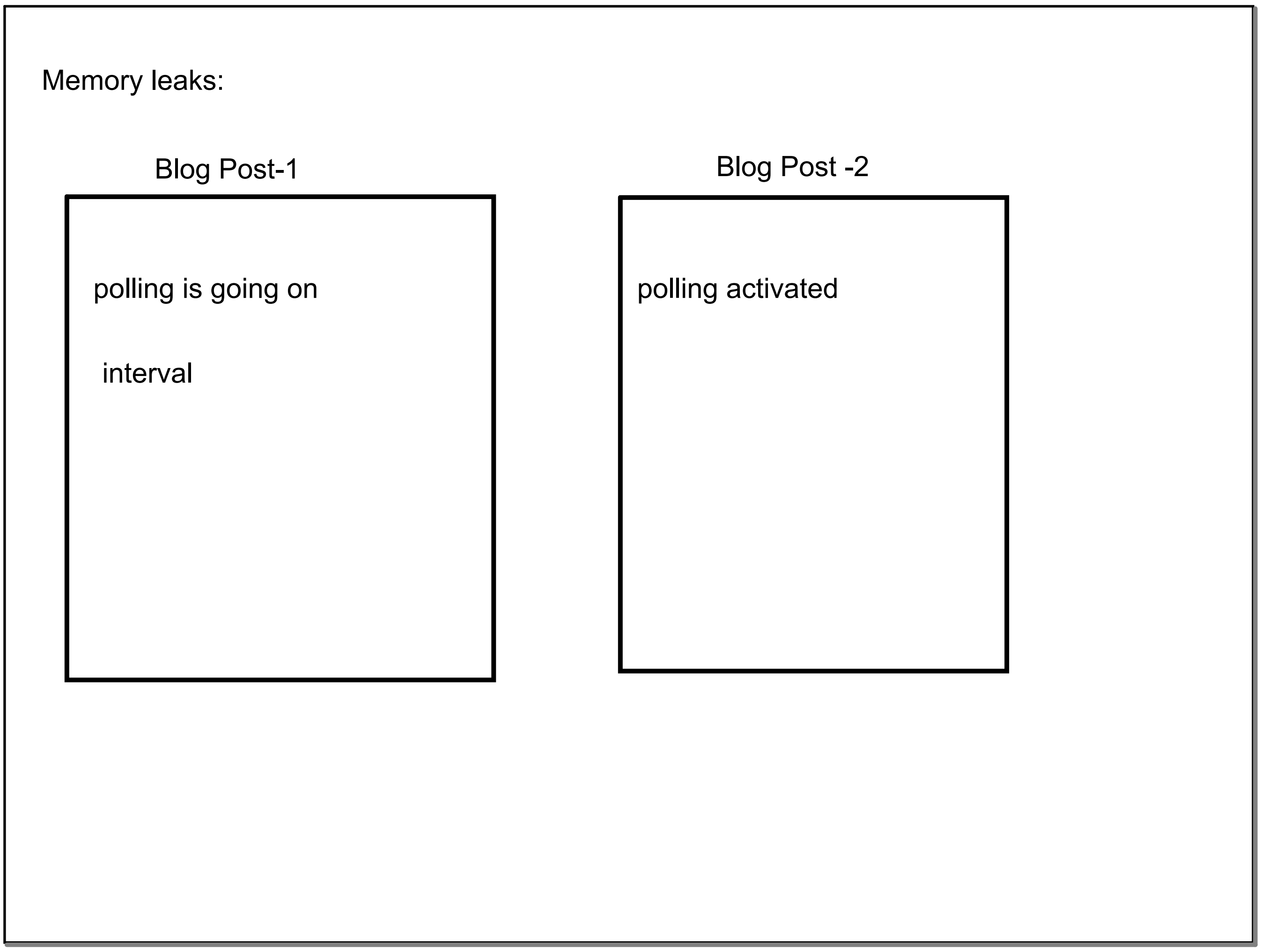
render

Schedule the call for _fetchComments()

componentDidMount()

componentWillUnMount()

whenever a component is about to be removed from DOM Tree


Memory leaks:
Blog Post-1
polling is going on
interval
Blog Post -2
polling activated