

Skill Training

React JS

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Schedule for React JS Training

Day1 : State Management : Class Components

Day2 : State Management : Functional Components, Hooks

Day3 : Http Client Programming

Day4 : Working with Forms, Validations, Services

Day5 : Redux

Day6 : Unit Testing

Day7 : Routing – SPA in React JS

Day4

Working with Forms



Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Index – Day4

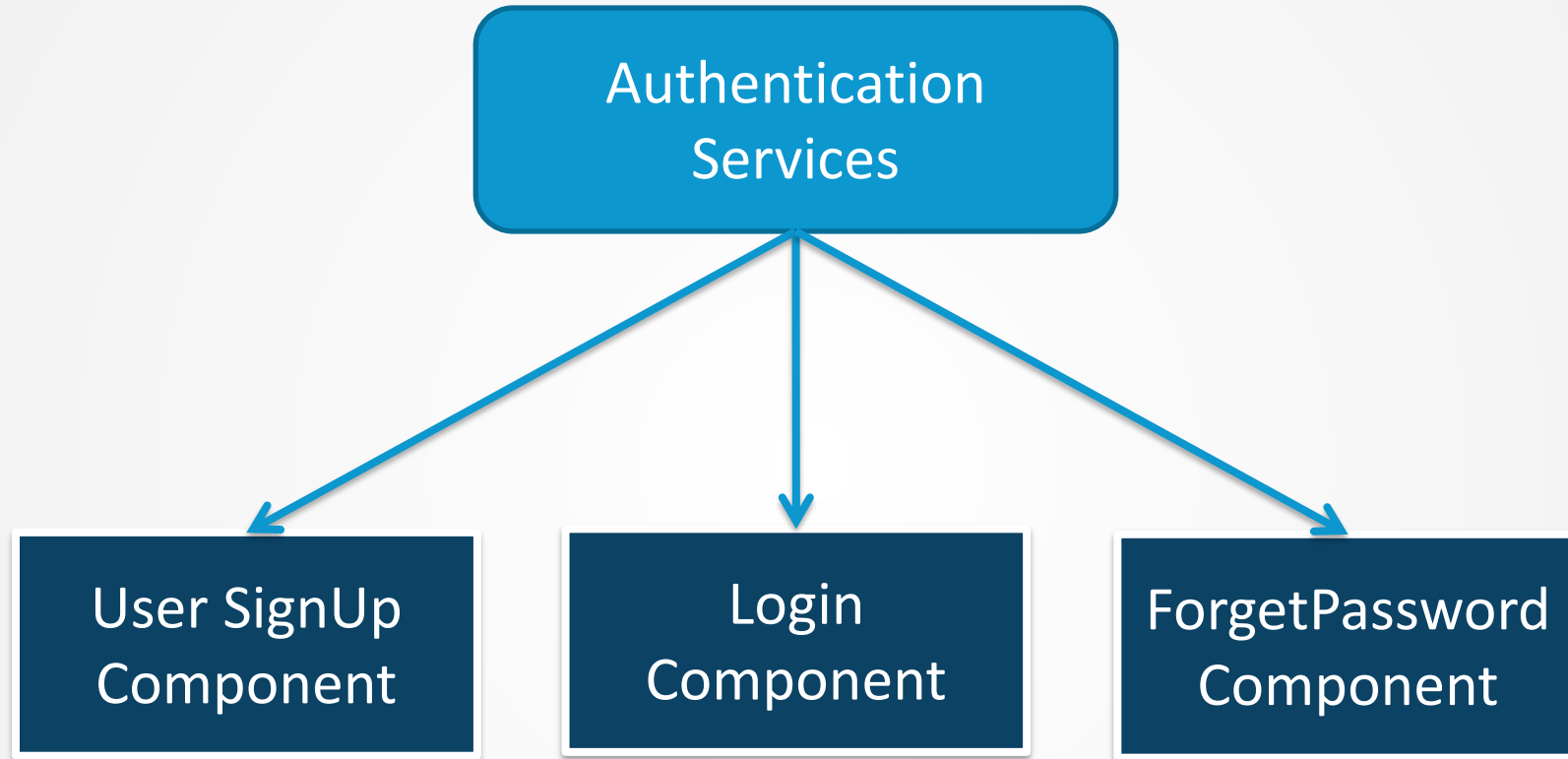
1. Reusable Services objects in React
2. Controlled and Uncontrolled components
3. React Forms
4. Custom Validations using React Forms

Reusable Services objects in React

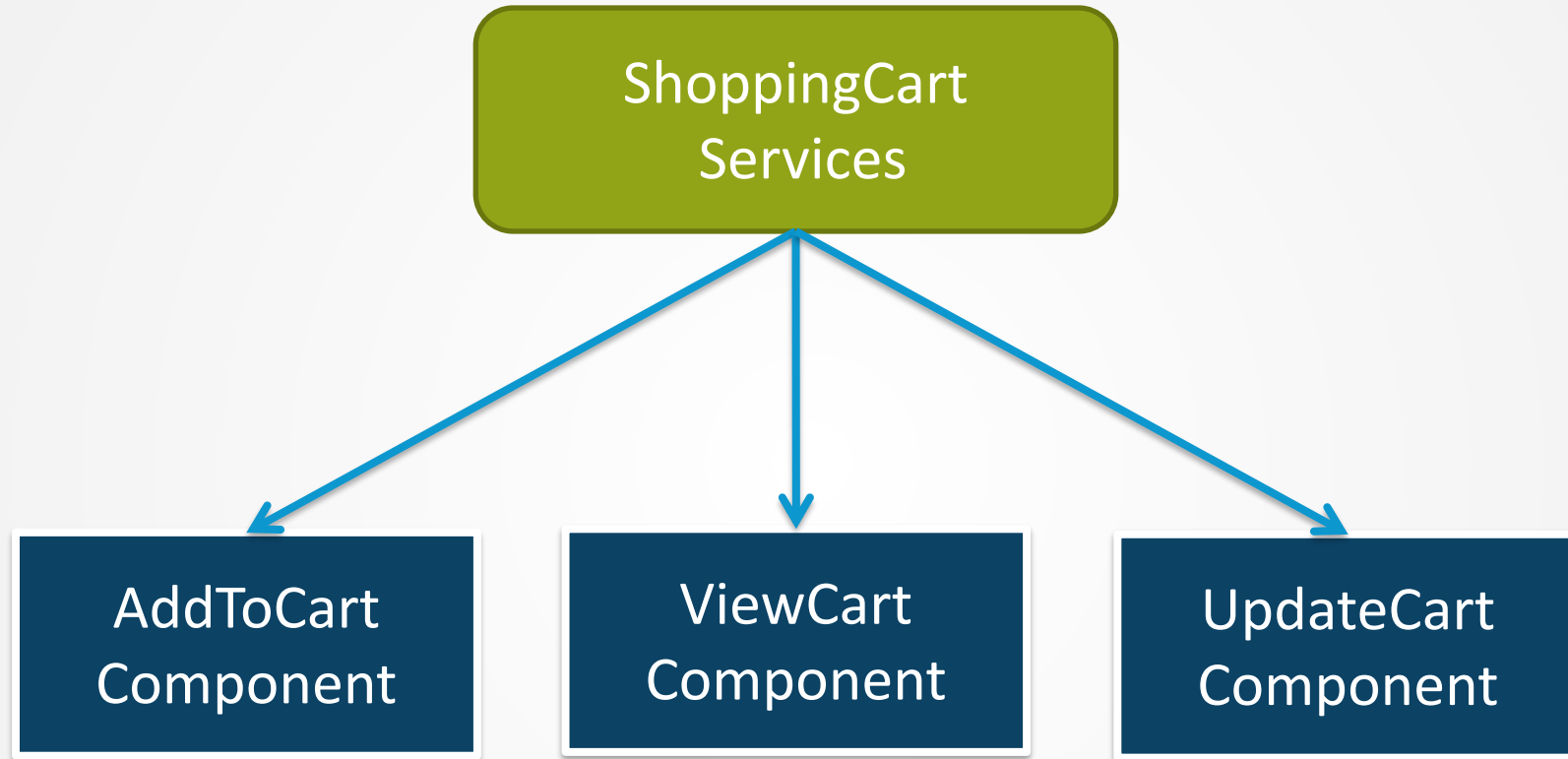
Services in React JS

- Service is just a group of helper functions that handle some operations particularly and can be reused across application.
- Eg:
 - Data Service that includes getData, updateData, deleteData, createData.....
 - Authentication Service that includes signing in, signing out, getting user info ...

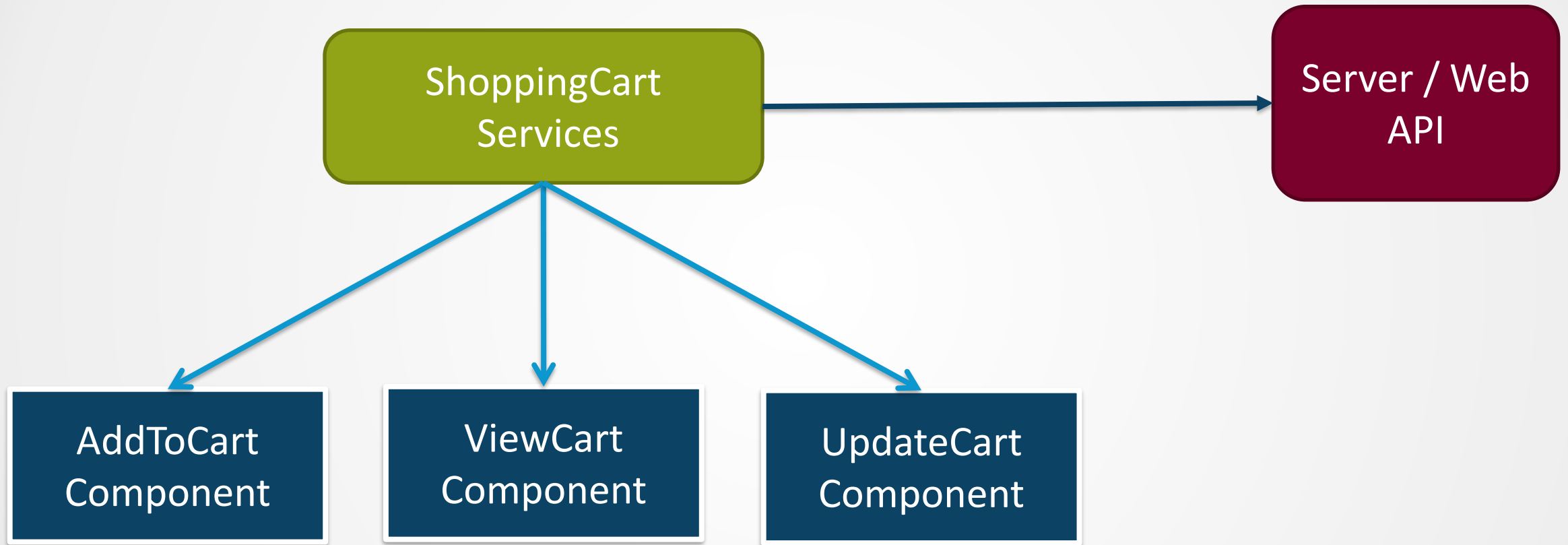
Services in React JS



Services in React JS



Services in React JS



Services in React JS

```
let dataSericeObj = {};
```

DataService.js

```
dataSericeObj.getAllDepts = function()  
{  
    // code to perform the operation  
};  
  
....  
  
export default dataSericeObj;
```

App.js

```
import dataSericeObj from "../DataService";  
  
function Demo() {  
    dataSericeObj .getAllDepts();  
    ....  
}  
  
export default Demo;
```

Practice Hands-Ons

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Controlled vs Uncontrolled Components

Controlled vs Uncontrolled Components

- A Controlled Component is one that takes its current value through props and notifies changes through callbacks like `onChange`.
 - A parent component "controls" it by handling the callback and managing its own state and passing the new values as props to the controlled component.
-
- A Uncontrolled Component is one that stores its own state internally, and you query the DOM using a `ref` to find its current value when you need it. This is a bit more like traditional HTML.

Controlled vs Uncontrolled Components

- In most cases, we recommend using controlled components to implement forms.
- In a controlled component, form data is handled by a React component.
- The alternative is uncontrolled components, where form data is handled by the DOM itself.
- To write an uncontrolled component, instead of writing an event handler for every state update, you can use a ref to get form values from the DOM.

Controlled vs Uncontrolled Components

```
import {useRef} from 'react';
```

```
const inputRef = useRef(null);
```

```
<input ref={inputRef} type="text" value={uname} />
```

```
inputRef.current.focus();
```

React Forms

Working with Forms in React

- Just like in HTML, React uses forms to allow users to interact with the web page.
- You can add a form with React like any other html element
- This will work as normal, the form will submit and the page will refresh. But this is generally not what we want to happen in React.
- We want to prevent this default behavior and let React control the form.

onSubmit event

```
const handleSubmit = (event) =>
{
  event.preventDefault();
  alert(`Form is submitted by user`);
}
```

```
<form onSubmit={handleSubmit}>
  .....
  <input type="submit" />
</form>
```

Practice Hands-Ons

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com



Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

