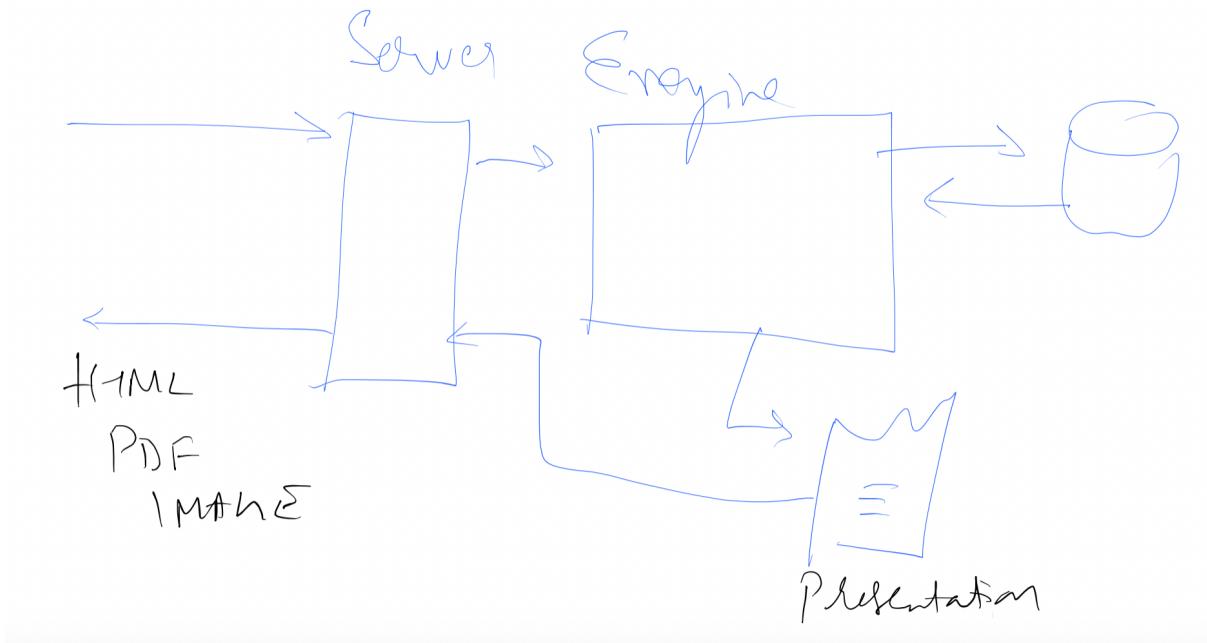
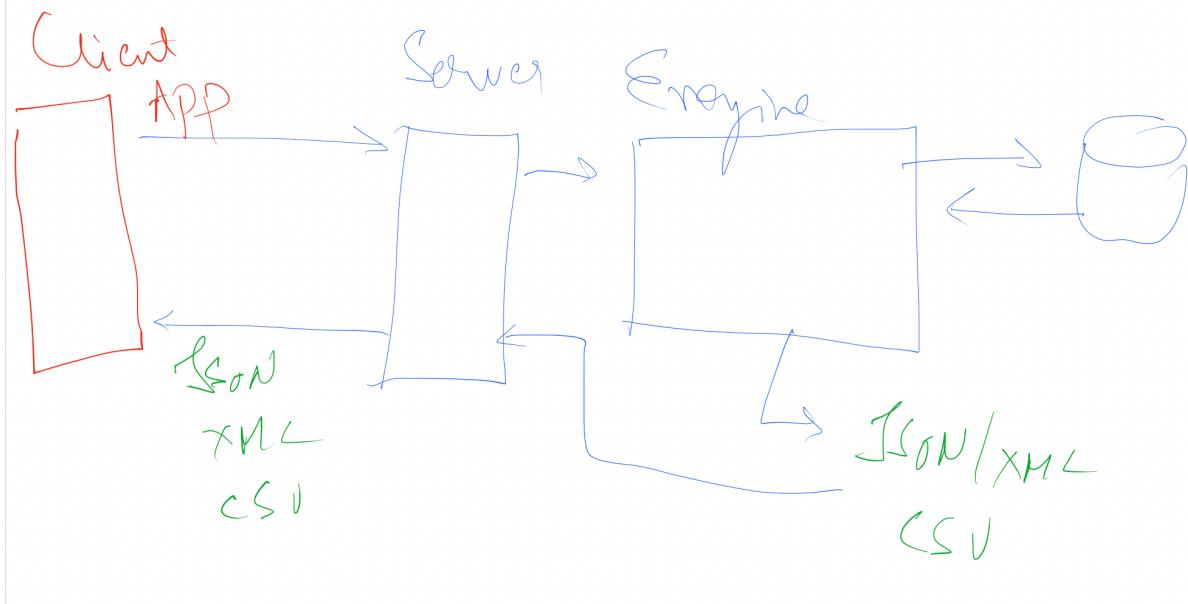


Server Side Rendering



Client Side Rendering



```

<ul class="people_list">
  {{#each people}}
    <li>{{this}}</li>
  {{/each}}
</ul>

```

```

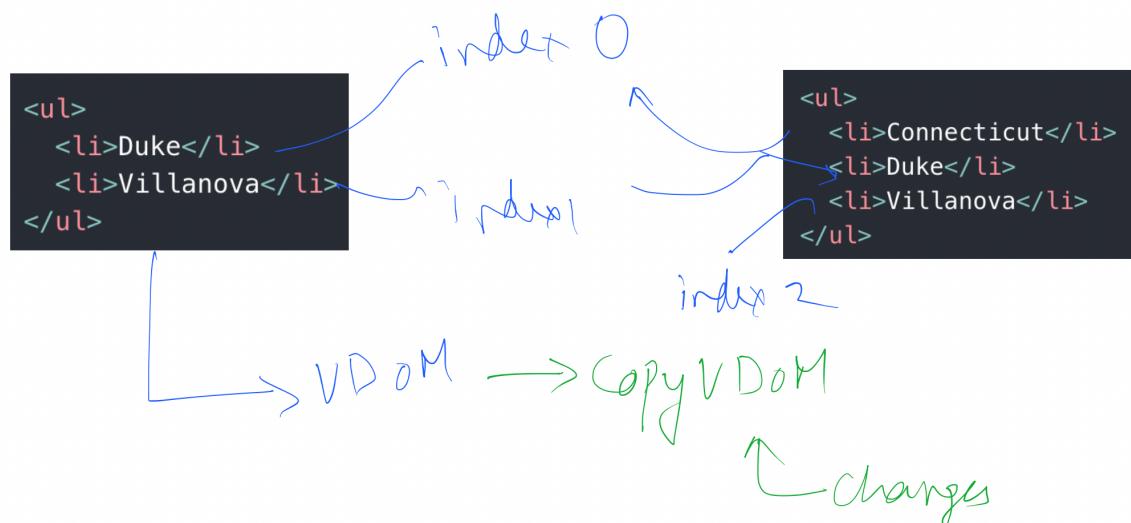
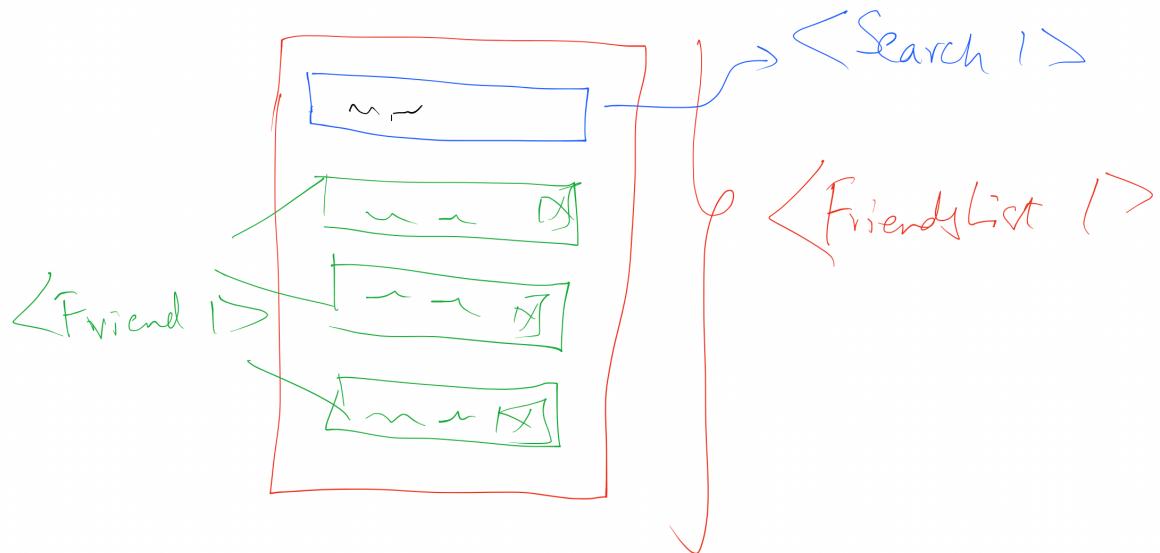
{
  people: [
    "Yehuda Katz",
    "Alan Johnson",
    "Charles Jolley",
  ],
}

```

```

<ul class="people_list">
  <li>Yehuda Katz</li>
  <li>Alan Johnson</li>
  <li>Charles Jolley</li>
</ul>

```



index.js

```

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App />);

function App() {
  return (
    <div className="App">
      <h1>Friends List</h1>
      <FriendsList />
    </div>
  );
}

export default class FriendsList extends Component {
  state = {
    friends
  }

  render() {
    return <div className="list">
      {
        this.state.friends.map(f => <Friend friend={f} key={f.id}/>)
      }
    </div>
  }
}

export default class Friend extends Component {
  render() {
    let {id, firstName, lastName} = this.props.friend;
    return <div className="row">
      {firstName} {lastName} &nbsp;
      <button type="button">&times;</button>
    </div>
  }
}

// deleteCustomer(id) {
//   let frds = this.state.friends.filter(f => f.id !== id);
//   //this.state.friends = frds; // avoid this, state changes but reconciliation won't happen
//   //async method to update state and reconcile
//   this.setState({
//     friends : frds
//   })
// }

render() {
  return <div className="list">
    {
      this.state.friends.map(f => <Friend
        delEvent = {(id) => this.deleteCustomer(id)}
        friend={f}
        key={f.id}/>)
    }
  </div>
}

```

Friend.js

```

export default class Friend extends Component {
  render() {
    let {id, firstName, lastName} = this.props.friend;
    return <div className="row">
      {firstName} {lastName} &nbsp;
      <button type="button" onClick={() => this.props.delEvent(id)}>&times;</button>
    </div>
  }
}

```