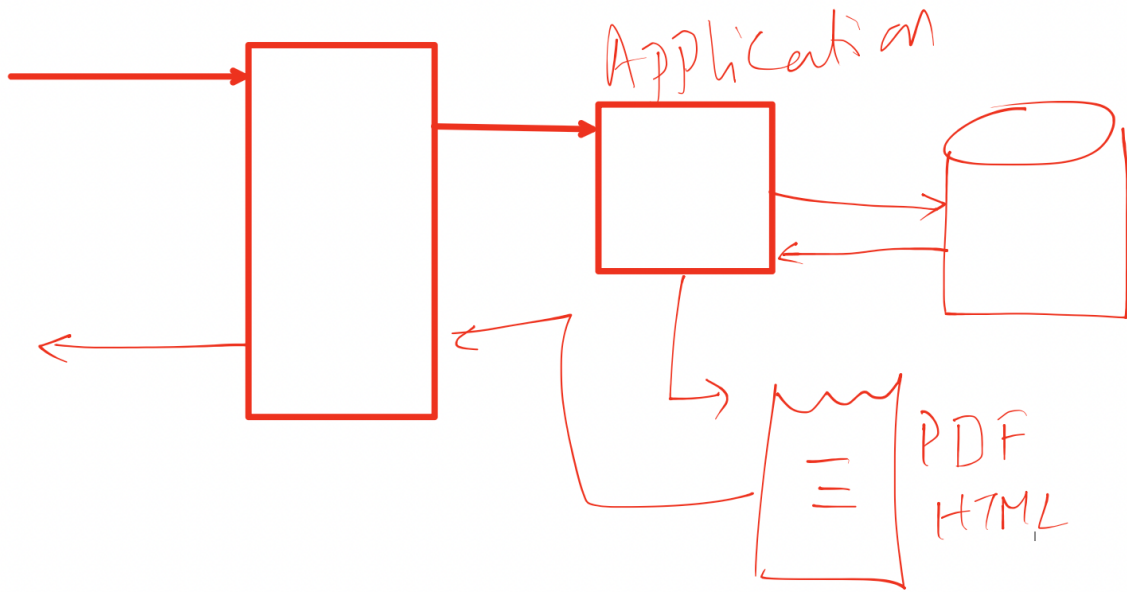
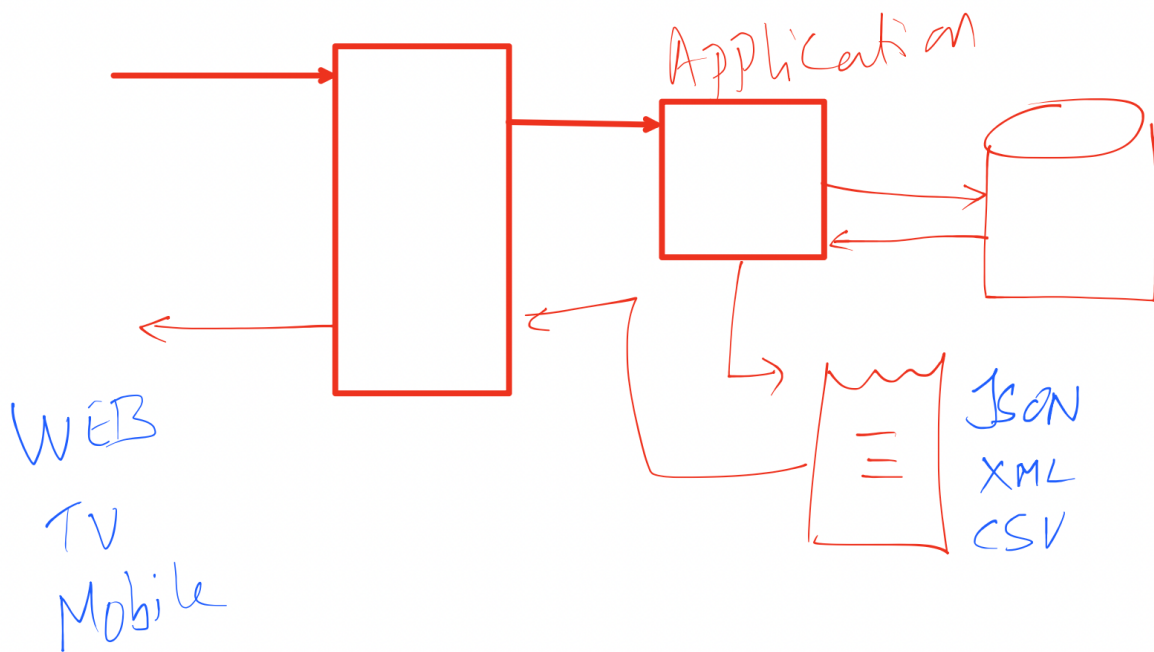


SSR



CSR



```

<ul>
  <li>Duke</li>
  <li>Villanova</li>
</ul>

```

DOM

→ VDOM

tag: ul

{
 tag: 'li'
 index: 0
 children: Duke
 }
 {
 tag: 'li'
 index: 1
 children: Villanova
 }

```

<ul>
  <li>Duke</li>
  <li>Villanova</li>
</ul>

```

DOM

VDOM Copy

tag: ul

{
 tag: 'li'
 index: 0 ✓
 children: ~~Duke~~ Connecticut
 }
 {
 tag: 'li'
 index: 1 ✓
 children: Villanova ~~Duke~~
 }

```

<ul>
  <li>Connecticut</li>
  <li>Duke</li>
  <li>Villanova</li>
</ul>

```

append

tag: ul
 {
 tag: 'li' ✓
 index: 0
 children: Duke
 }
 {
 tag: 'li'
 index: 1 ✓
 children: Villanova
 }

← diff alg →

tag: li index: 2 Villanova

```

export default class CustomerList extends Component {
  state = {
    // ...
  }

  deleteCustomer(id) {
    let custs = this.state.customers.filter(c => c.id !== id);
    // this.state.customers = custs; // update the state, reconciliation won't happen
    // update the state
    this.setState({
      customers: custs
    });

    console.log(this.state.customers.length);
  }

  render() {
    return (
      <div>
        <Filter />
        {
          this.state.customers.map(customer => <CustomerRow
            deleteEvent={id => this.deleteCustomer(id)}
            customer={customer} key={customer.id} />)
        }
      </div>
    );
  }
}

```

```

export default class CustomerRow extends Component {
  deleteCustomerRow(id) {
    console.log("<CustomerRow> id", id);
    this.props.deleteEvent(id); // passed from CustomerList -Parent
  }

  render() {
    // destructuring
    let {id, name, played_by} = this.props.customer;
    return (
      <div>
        {name} as {played_by}
        &nbsp;
        <button type='button' onClick={() => this.deleteCustomerRow(id)}>Delete</button>
      </div>
    );
  }
}

```

```
export default class CustomerList extends Component {  
  state = {  
  }  
  
  componentDidMount() {  
    this.state.complete = this.state.customers; // copy  
  }  
  filterCustomers(txt) {  
    let custs = this.state.complete.filter(c => c.name.indexOf(txt) >= 0);  
    this.setState({  
      customers: custs  
    })  
  }  
}  
  
render() {  
  return (  
    <div>  
      <Filter filterEvent={txt => this.filterCustomers(txt)} />  
    </div>  
  )  
}
```

```
export default function Filter(props) {  
  return (  
    <div>  
      <input type='text'  
        placeholder='search by name'  
        onChange={(evt) => props.filterEvent(evt.target.value)} />  
    </div>  
  )  
}
```

2

1