



DEV ACADEMY
TE KURA HANGARAU
O AOTEAROA

Introduction to React





Preface

- Lotta change, today!
- Be gentle with yourself (take breaks)
- Ask questions



Agenda for Today

- Server-Side Rendering vs Single Page App
- Compiling & Bundling
- Components and JSX
- Functional Programming



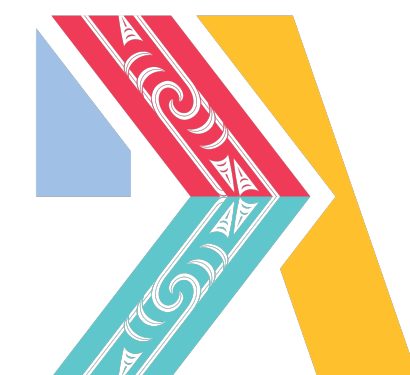
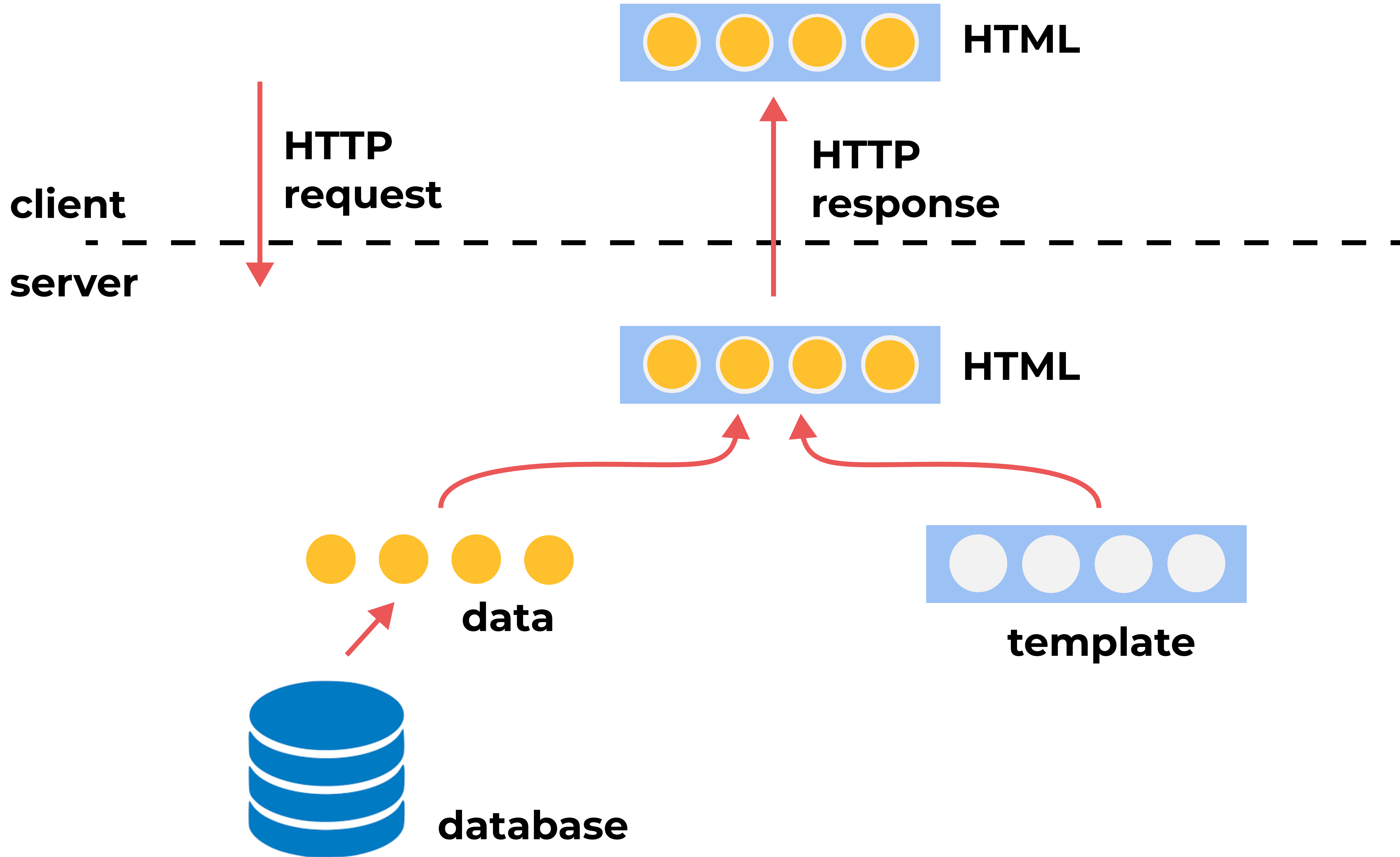
SSR

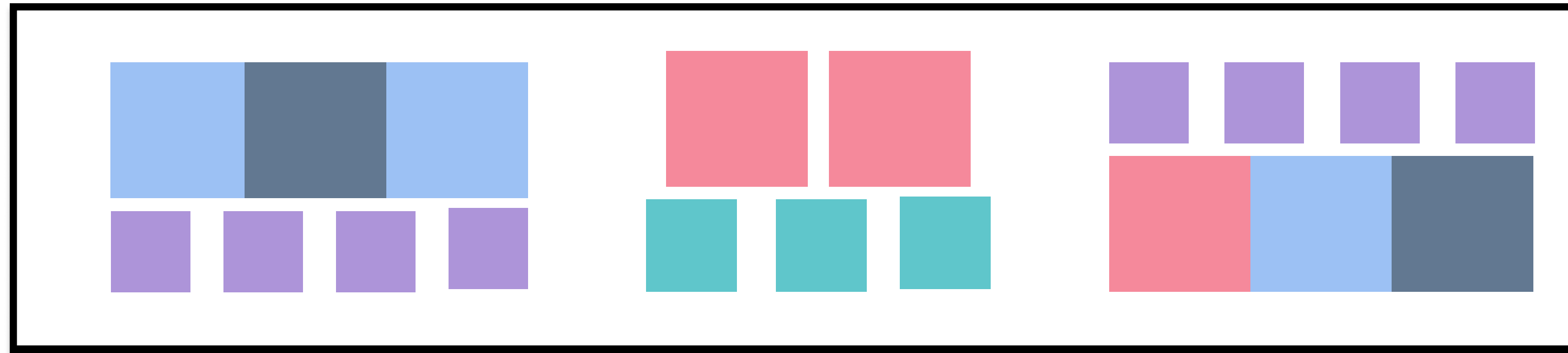
- Static
- Rendering happens on the server
- Routes handled by express

SPA

- Dynamic
- Rendering happens in the client
- Routes will be handled by React Router (not today, though)







React.js

**HTTP
request**

client

server



Components





Compiling & bundling our code

- Previously, the JS we wrote was the JS that went to the browser
- Now: TypeScript and JSX
- Now: Bundling, which reduces network requests, affects CSS and images, too
- Vite
 - Includes "hot reloading"

Now, the code that eventually ships to the browser in production will not look like the code we write.





JSX

- Javascript Syntax eXtension
- Write HTML-*like* markup in Javascript
- Allows us to embed markup easily and mix it with JS
- Single curly braces to signify JS

```
1  {artworks.map(art => <li>{art.name}</li>)}
```




Components

- Components are the basic building blocks of React
 - A component is a function that accepts "props" and returns markup
 - Always start with a capital letter
- Compose multiple components together to build UI
- To call the component function and render its results, include it like an HTML Element

```
`<Widget></Widget>`
```

- To pass props to the component function, write them like HTML attributes

```
`<Widget colour="red"></Widget>`
```



Writing a component

```
1  function Greeting(props) {  
2  
3    return (  
4      <p>Hello, {props.yourName}</p>  
5    )  
6  }
```

Using a component

```
1  function GreetingContainer() {  
2  
3    return (  
4      <h1>Welcome</h1>  
5      <Greeting yourName="world" />  
6    )  
7  }
```

Result in the browser

```
1  <h1>Welcome</h1>  
2  <p>Hello, world.</p>
```



Functional programming

- Focus on values, rather than effects
 - What goes in and what comes out?
 - Spread operators are extra handy
- Compare to DOM manipulation (blueprint vs recipe)



Review

- What is one way that our work in React will differ from the apps we built in phase 1?