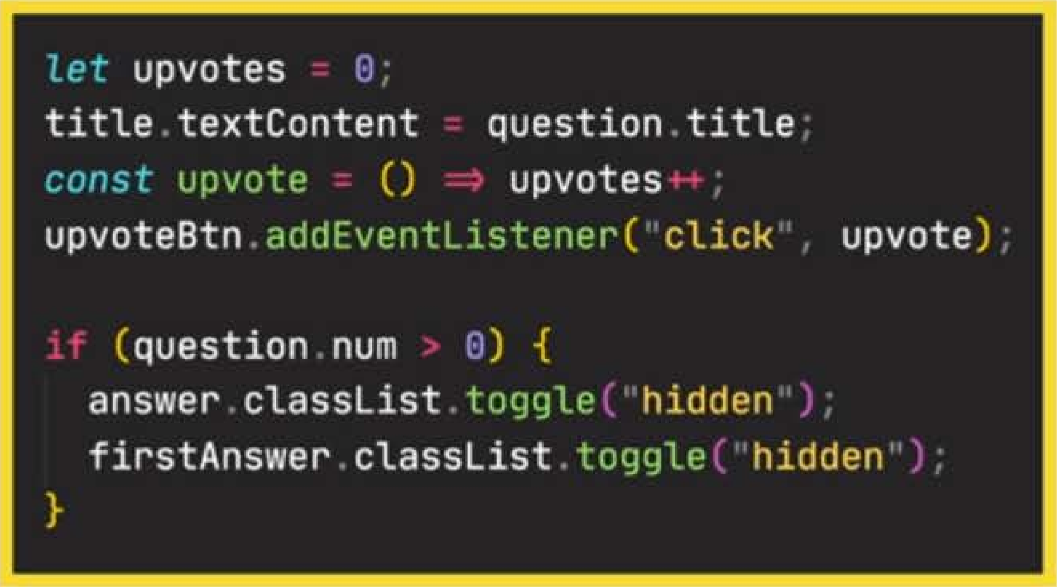
SEPARATION OF CONCERNS?

SEPARATION OF CONCERNS?

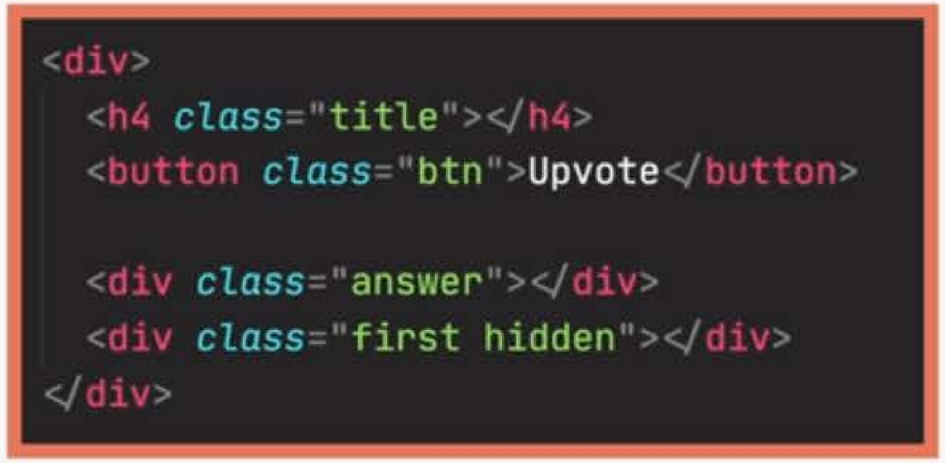
Rise of interactive SPAs

SEPARATION OF CONCERNS?

JS



```javascript
let upvotes = 0;
title.textContent = question.title;
const upvote = () => upvotes++;
upvoteBtn.addEventListener("click", upvote);

if (question.num > 0) {
  answer.classList.toggle("hidden");
  firstAnswer.classList.toggle("hidden");
}
```



```html
<div>
  <h4 class="title"></h4>
  <button class="btn">Upvote</button>

  <div class="answer"></div>
  <div class="first hidden"></div>
</div>
```

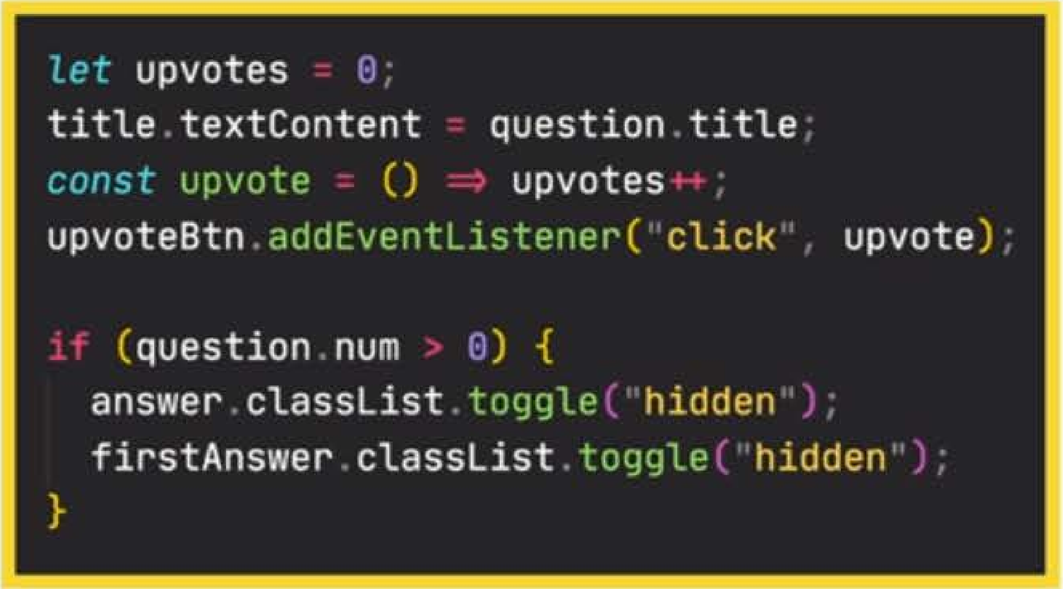
HTML

Rise of interactive SPAs

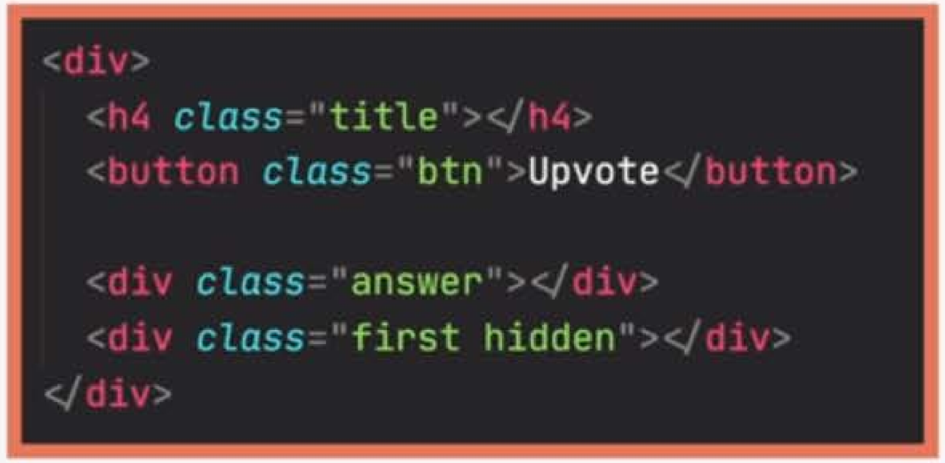
SEPARATION OF CONCERNS?

ONE *TECHNOLOGY* PER FILE

JS



```javascript
let upvotes = 0;
title.textContent = question.title;
const upvote = () => upvotes++;
upvoteBtn.addEventListener("click", upvote);

if (question.num > 0) {
  answer.classList.toggle("hidden");
  firstAnswer.classList.toggle("hidden");
}
```



```html
<div>
  <h4 class="title"></h4>
  <button class="btn">Upvote</button>

  <div class="answer"></div>
  <div class="first hidden"></div>
</div>
```

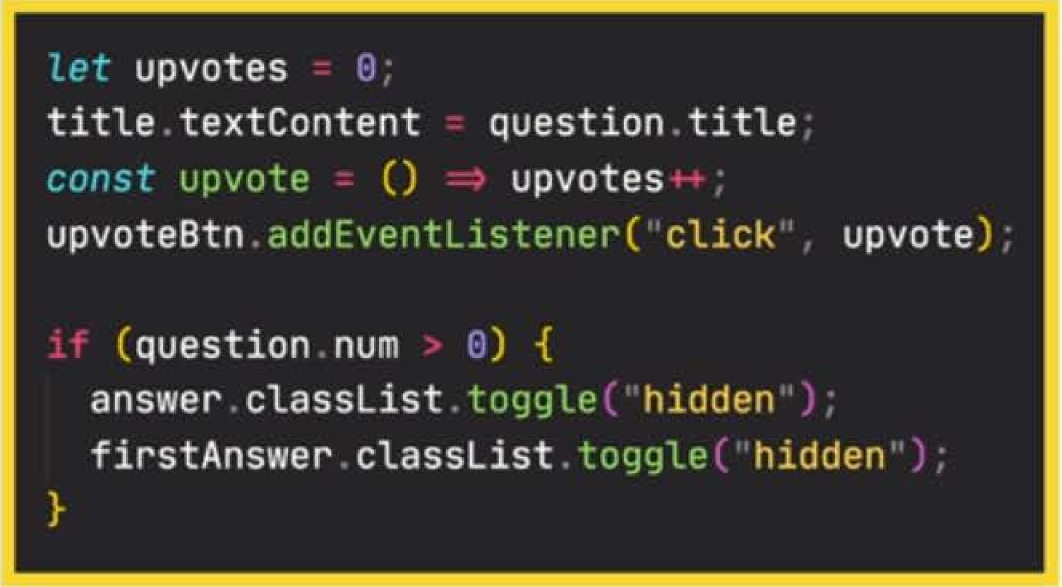
HTML

Rise of interactive SPAs

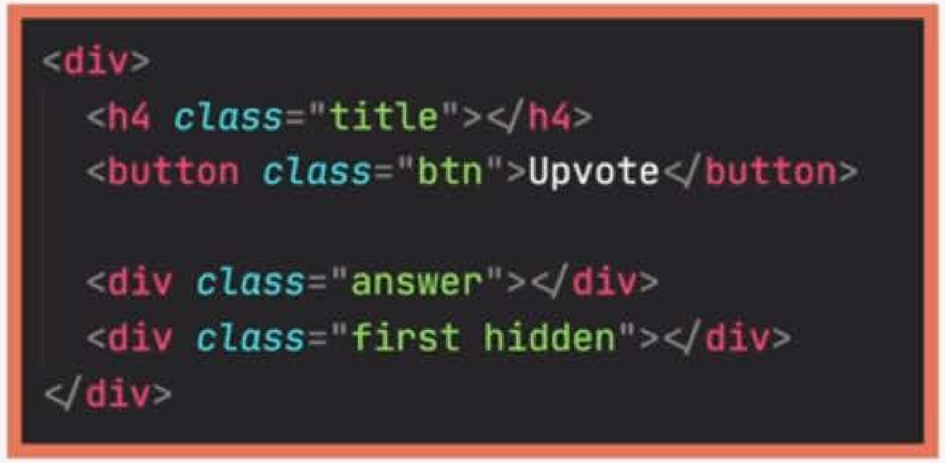
SEPARATION OF CONCERNS?

ONE *TECHNOLOGY* PER FILE

"Traditional" **separation of concerns**



```javascript
let upvotes = 0;
title.textContent = question.title;
const upvote = () => upvotes++;
upvoteBtn.addEventListener("click", upvote);

if (question.num > 0) {
  answer.classList.toggle("hidden");
  firstAnswer.classList.toggle("hidden");
}
```

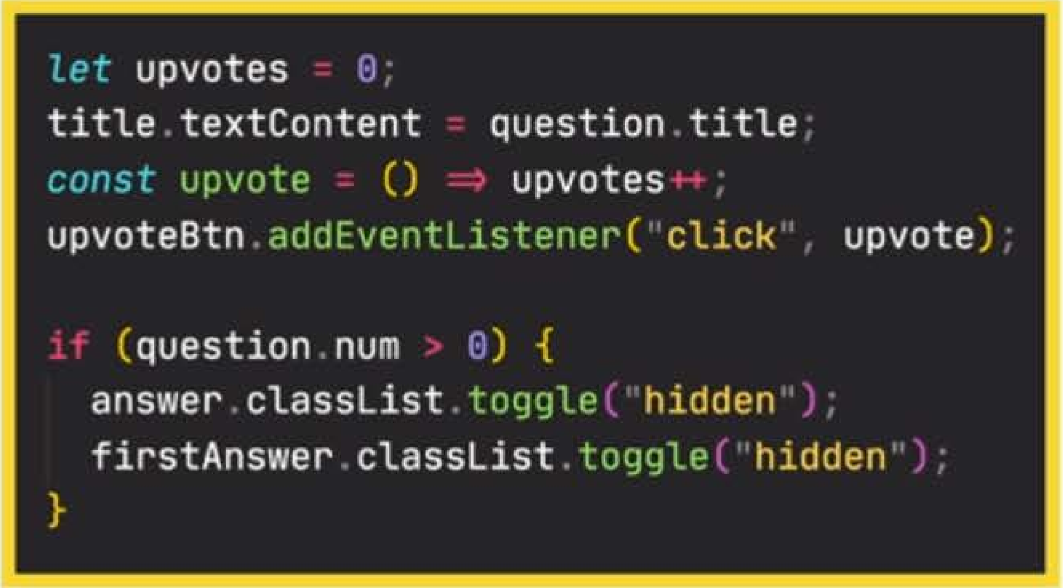


```html
<div>
  <h4 class="title"></h4>
  <button class="btn">Upvote</button>

  <div class="answer"></div>
  <div class="first hidden"></div>
</div>
```

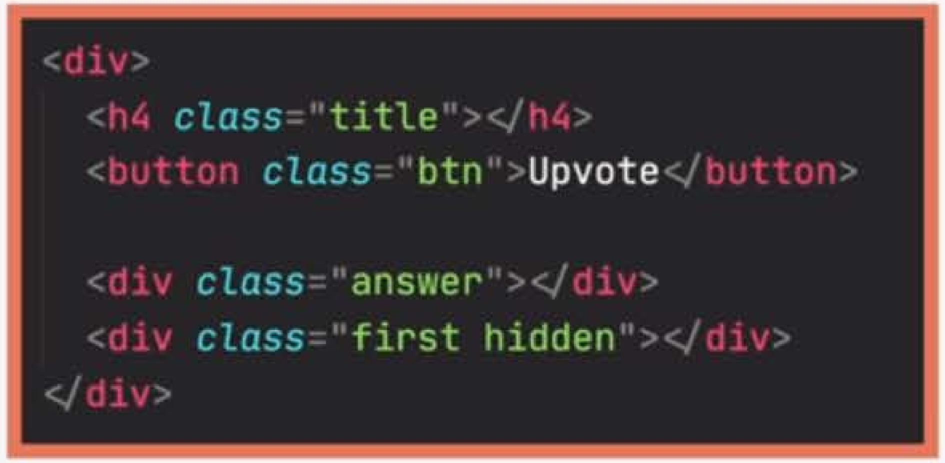
Rise of interactive SPAs

SEPARATION OF CONCERNS?

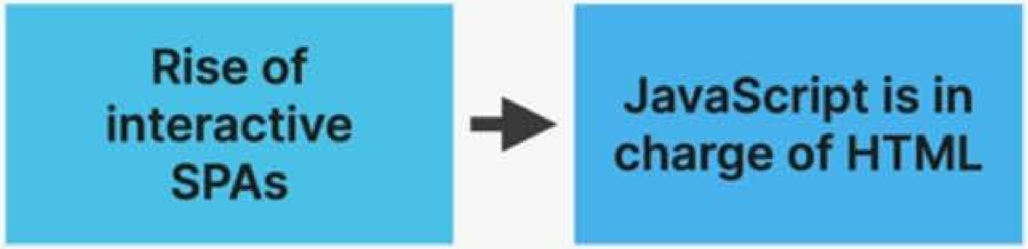
ONE *TECHNOLOGY* PER FILE ← “Traditional” **separation of concerns**



```javascript
let upvotes = 0;
title.textContent = question.title;
const upvote = () => upvotes++;
upvoteBtn.addEventListener("click", upvote);

if (question.num > 0) {
  answer.classList.toggle("hidden");
  firstAnswer.classList.toggle("hidden");
}
```



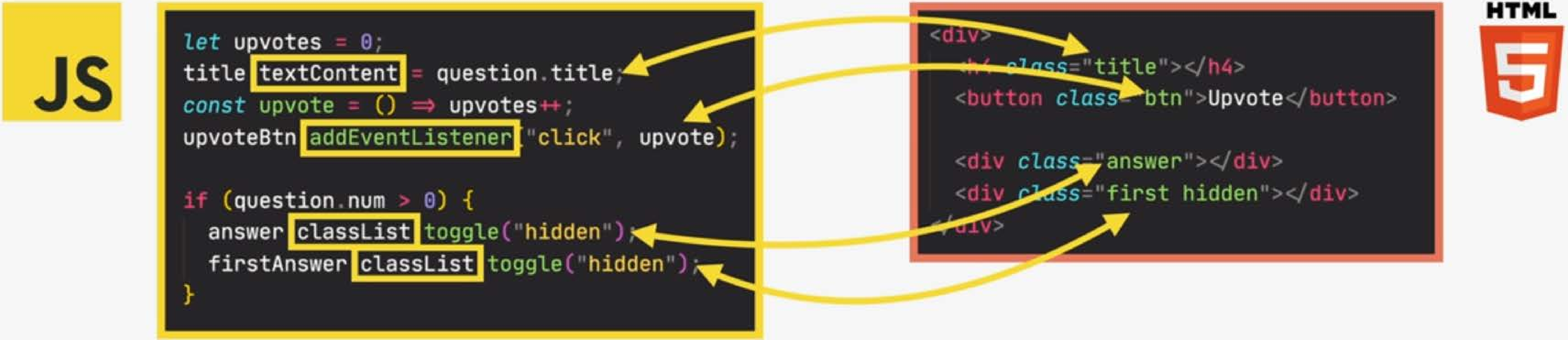
```html
<div>
  <h4 class="title"></h4>
  <button class="btn">Upvote</button>

  <div class="answer"></div>
  <div class="first hidden"></div>
</div>
```



Rise of interactive SPAs → JavaScript is in charge of HTML

SEPARATION OF CONCERNS?

ONE *TECHNOLOGY* PER FILE ← "Traditional" **separation of concerns**



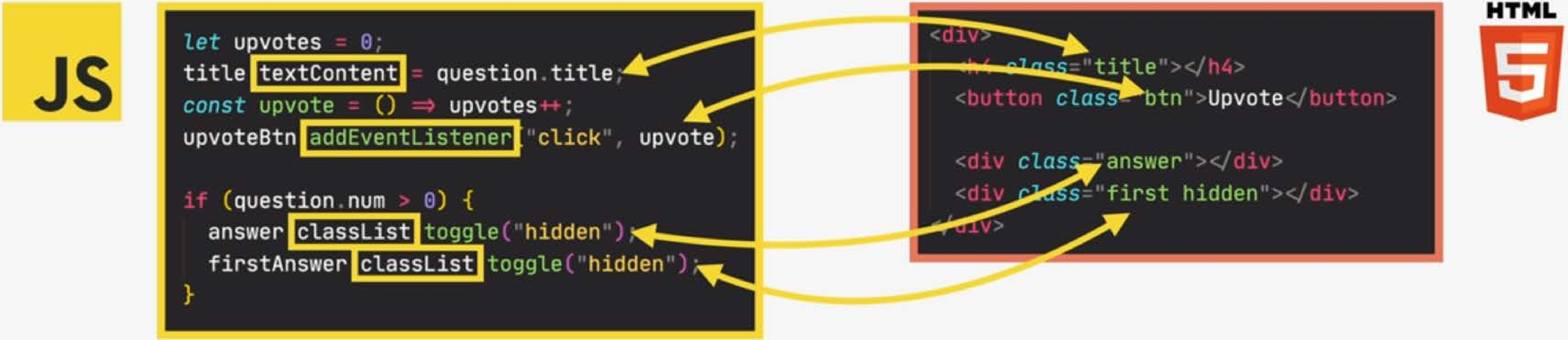
```javascript
let upvotes = 0;
title.textContent = question.title;
const upvote = () => upvotes++;
upvoteBtn.addEventListener("click", upvote);

if (question.num > 0) {
  answer.classList.toggle("hidden");
  firstAnswer.classList.toggle("hidden");
}
```

```html
<div>
  <h4 class="title"></h4>
  <button class="btn">Upvote</button>

  <div class="answer"></div>
  <div class="first hidden"></div>
</div>
```

Rise of interactive SPAs → **JavaScript is in charge of HTML**

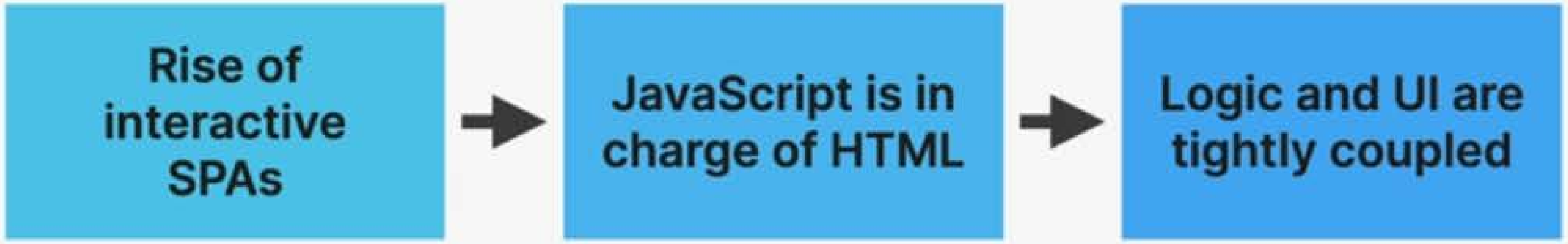
SEPARATION OF CONCERNS?

ONE *TECHNOLOGY* PER FILE ← “Traditional” **separation of concerns**



```
let upvotes = 0;
title.textContent = question.title;
const upvote = () => upvotes++;
upvoteBtn.addEventListener("click", upvote);

if (question.num > 0) {
  answer.classList.toggle("hidden");
  firstAnswer.classList.toggle("hidden");
}
```

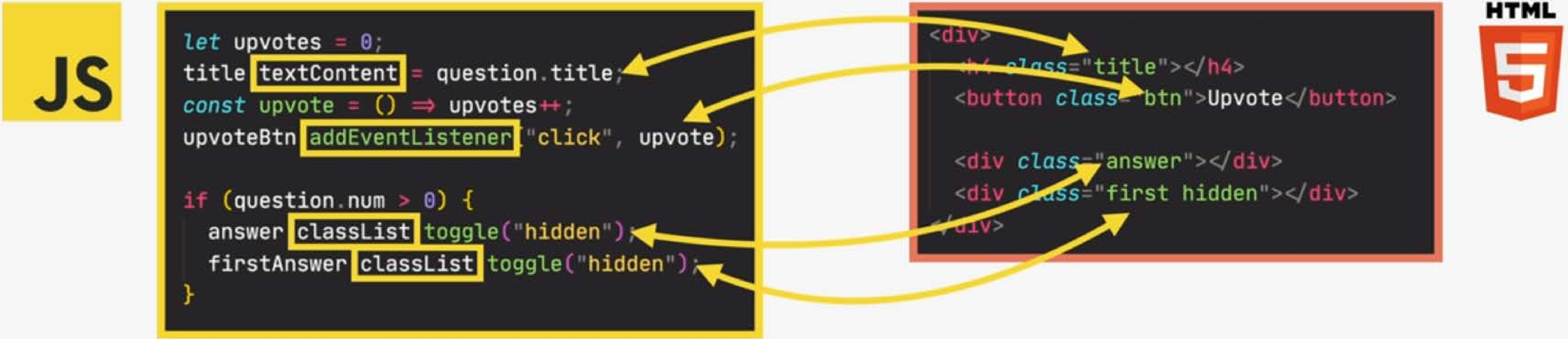
```
<div>
  <h4 class="title"></h4>
  <button class="btn">Upvote</button>

  <div class="answer"></div>
  <div class="first hidden"></div>
</div>
```



Rise of interactive SPAs → JavaScript is in charge of HTML → Logic and UI are tightly coupled

SEPARATION OF CONCERNS?

ONE *TECHNOLOGY* PER FILE ← "Traditional" **separation of concerns**



```javascript
let upvotes = 0;
title.textContent = question.title;
const upvote = () => upvotes++;
upvoteBtn.addEventListener("click", upvote);

if (question.num > 0) {
  answer.classList.toggle("hidden");
  firstAnswer.classList.toggle("hidden");
}
```

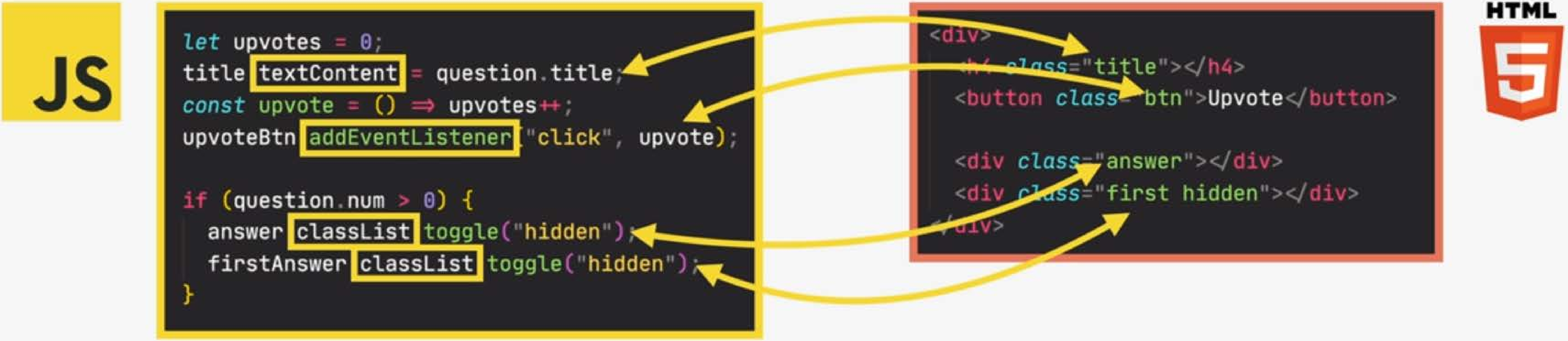
```html
<div>
  <h4 class="title"></h4>
  <button class="btn">Upvote</button>

  <div class="answer"></div>
  <div class="first hidden"></div>
</div>
```

Rise of interactive SPAs → **JavaScript is in charge of HTML** → **Logic and UI are tightly coupled** → **Why keep them separated?**

SEPARATION OF CONCERNS?

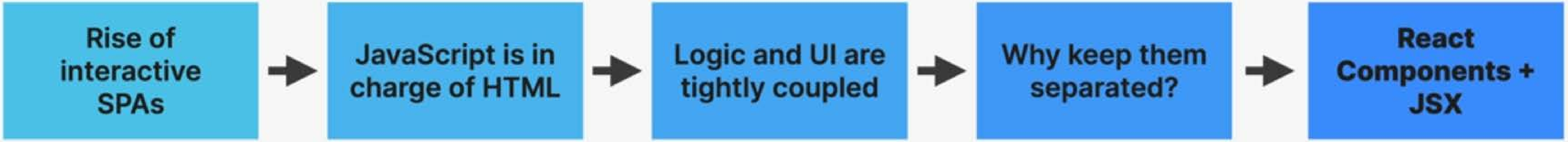
ONE *TECHNOLOGY* PER FILE

"Traditional" **separation of concerns**



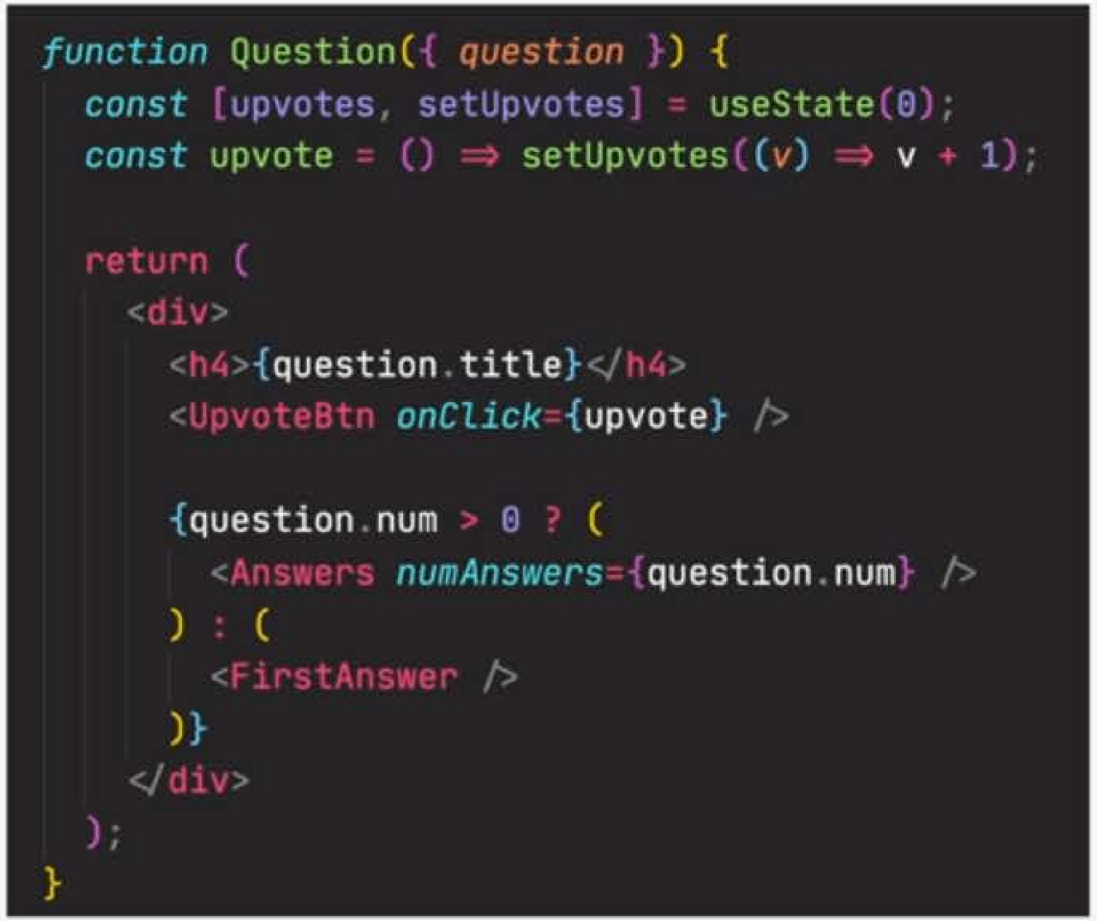
```
let upvotes = 0;
title.textContent = question.title;
const upvote = () => upvotes++;
upvoteBtn.addEventListener("click", upvote);

if (question.num > 0) {
  answer.classList.toggle("hidden");
  firstAnswer.classList.toggle("hidden");
}
```

```
<div>
  <h4 class="title"></h4>
  <button class="btn">Upvote</button>

  <div class="answer"></div>
  <div class="first hidden"></div>
</div>
```

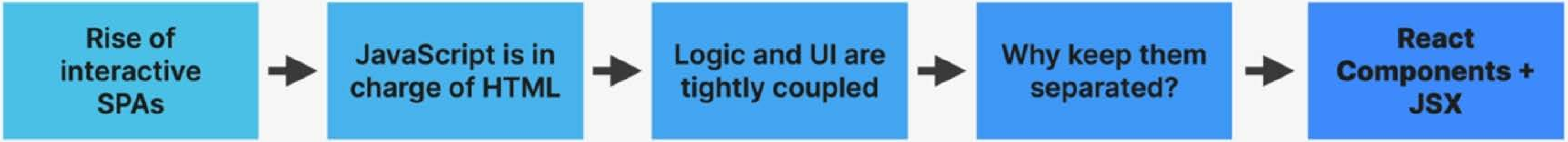


Rise of interactive SPAs → JavaScript is in charge of HTML → Logic and UI are tightly coupled → Why keep them separated? → React Components + JSX

SEPARATION OF CONCERNS?

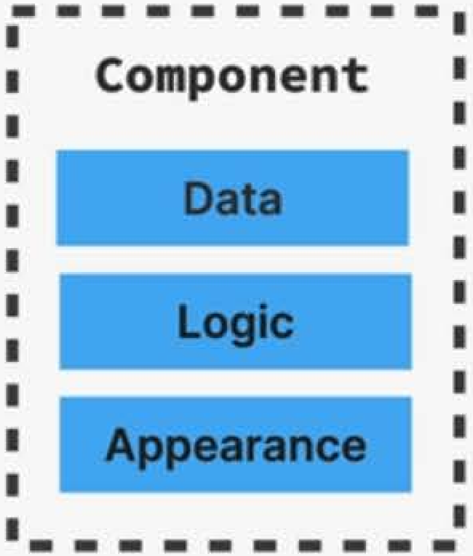


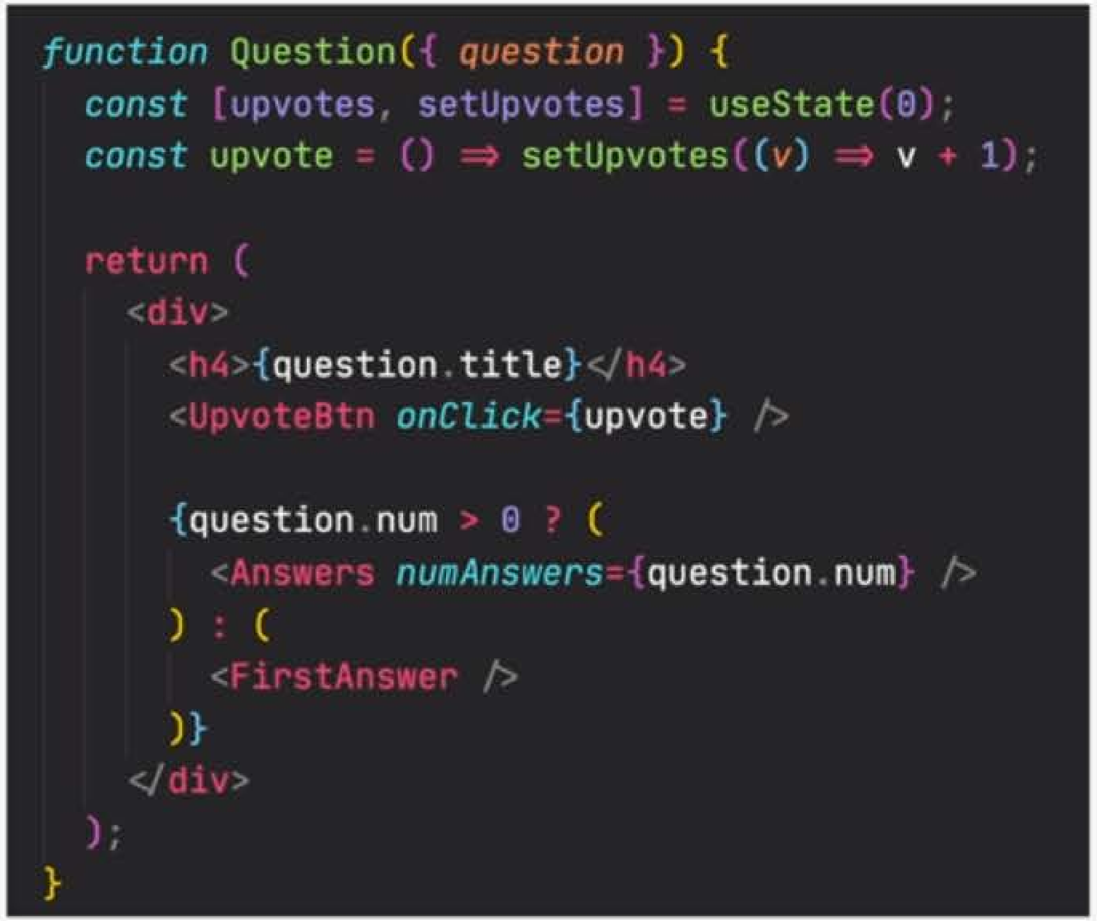
```
function Question({ question }) {
  const [upvotes, setUpvotes] = useState(0);
  const upvote = () => setUpvotes((v) => v + 1);

  return (
    <div>
      <h4>{question.title}</h4>
      <UpvoteBtn onClick={upvote} />

      {question.num > 0 ? (
        <Answers numAnswers={question.num} />
      ) : (
        <FirstAnswer />
      )}
    </div>
  );
}
```



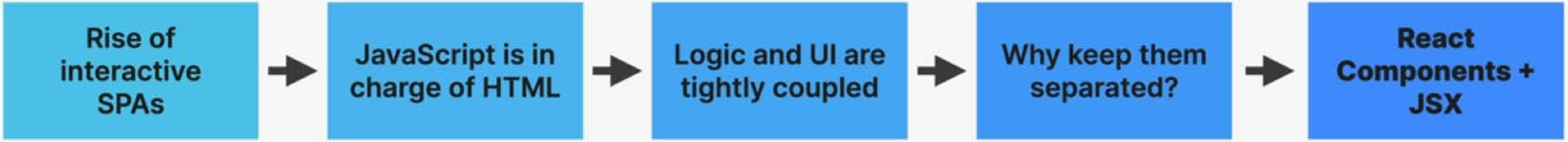
Rise of interactive SPAs → JavaScript is in charge of HTML → Logic and UI are tightly coupled → Why keep them separated? → React Components + JSX

SEPARATION OF CONCERNS?



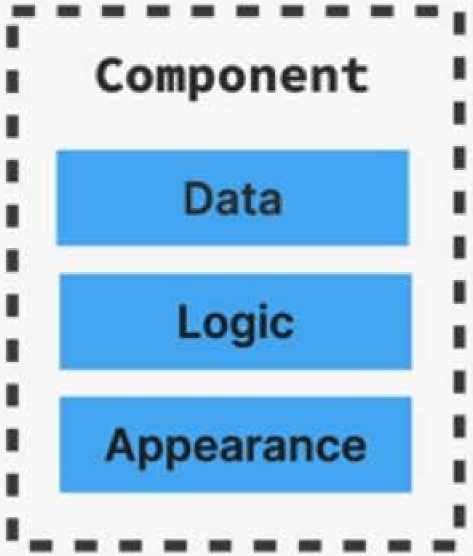


```
function Question({ question }) {
  const [upvotes, setUpvotes] = useState(0);
  const upvote = () => setUpvotes((v) => v + 1);

  return (
    <div>
      <h4>{question.title}</h4>
      <UpvoteBtn onClick={upvote} />

      {question.num > 0 ? (
        <Answers numAnswers={question.num} />
      ) : (
        <FirstAnswer />
      )}
    </div>
  );
}
```



Rise of interactive SPAs → JavaScript is in charge of HTML → Logic and UI are tightly coupled → Why keep them separated? → React Components + JSX

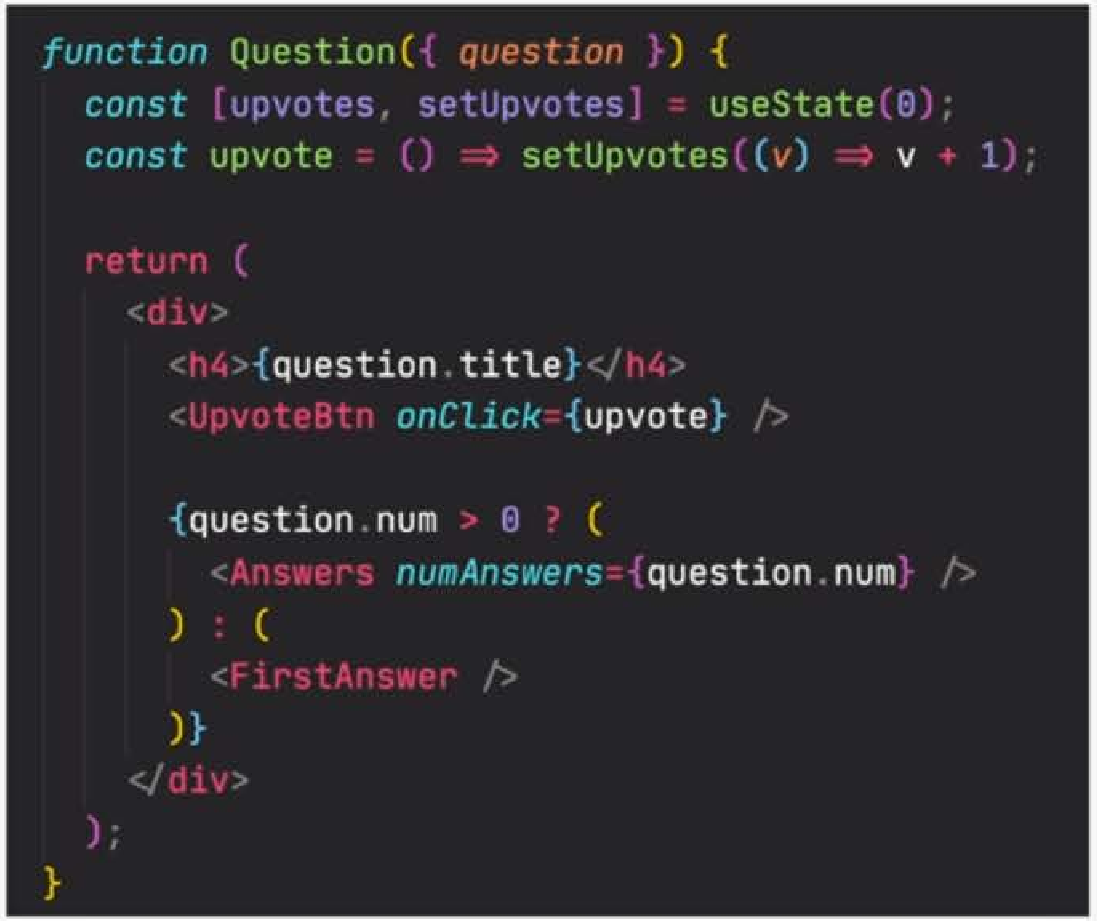
SEPARATION OF CONCERNS?


Component

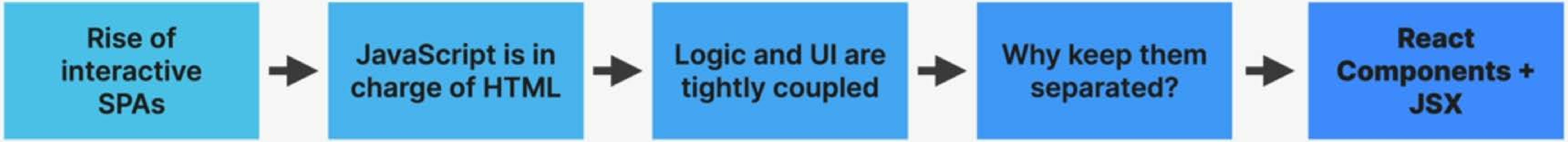
Data

Logic

Appearance


```
function Question({ question }) {
  const [upvotes, setUpvotes] = useState(0);
  const upvote = () => setUpvotes((v) => v + 1);

  return (
    <div>
      <h4>{question.title}</h4>
      <UpvoteBtn onClick={upvote} />

      {question.num > 0 ? (
        <Answers numAnswers={question.num} />
      ) : (
        <FirstAnswer />
      )}
    </div>
  );
}
```

Fundamental reason for **components**


Rise of interactive SPAs

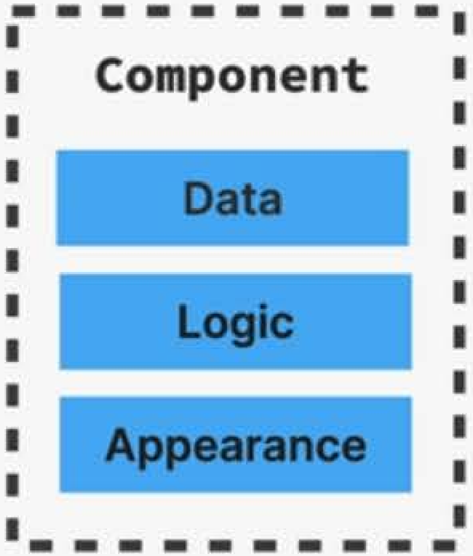
JavaScript is in charge of HTML

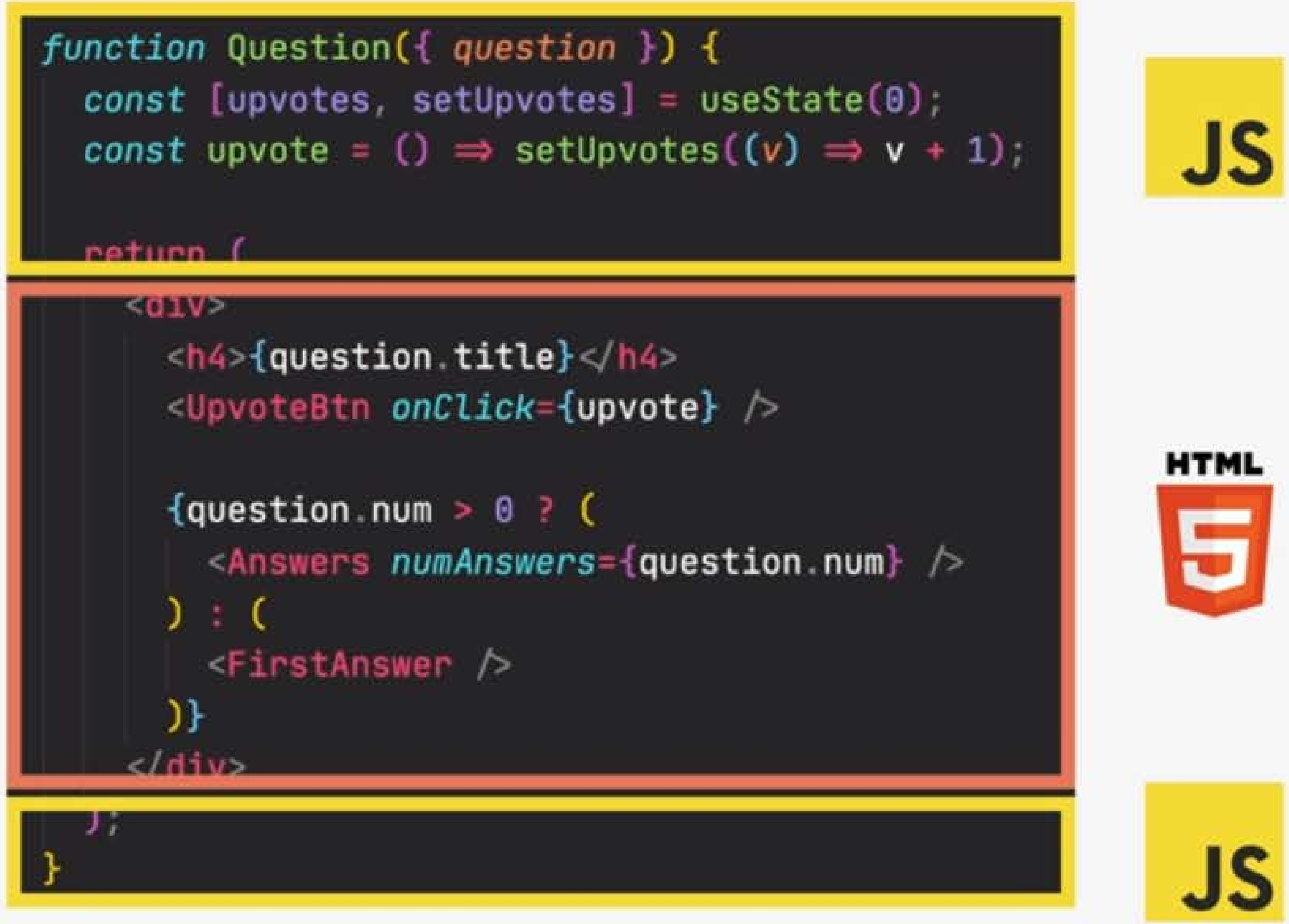
Logic and UI are tightly coupled

Why keep them separated?

React Components + JSX

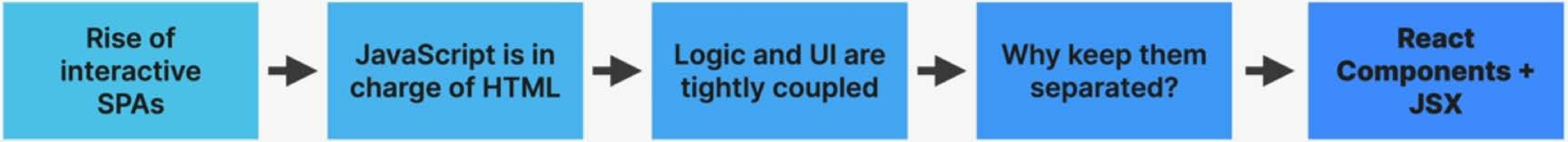
SEPARATION OF CONCERNS?



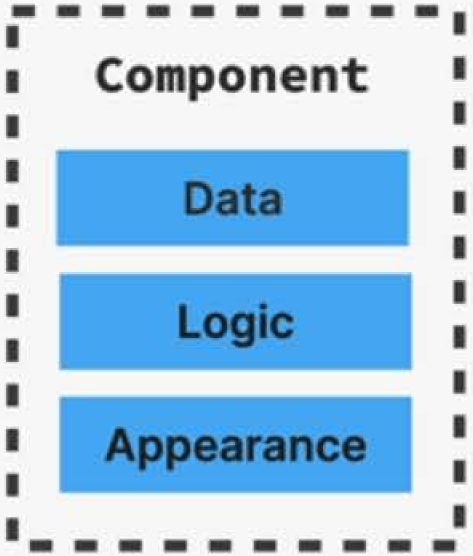


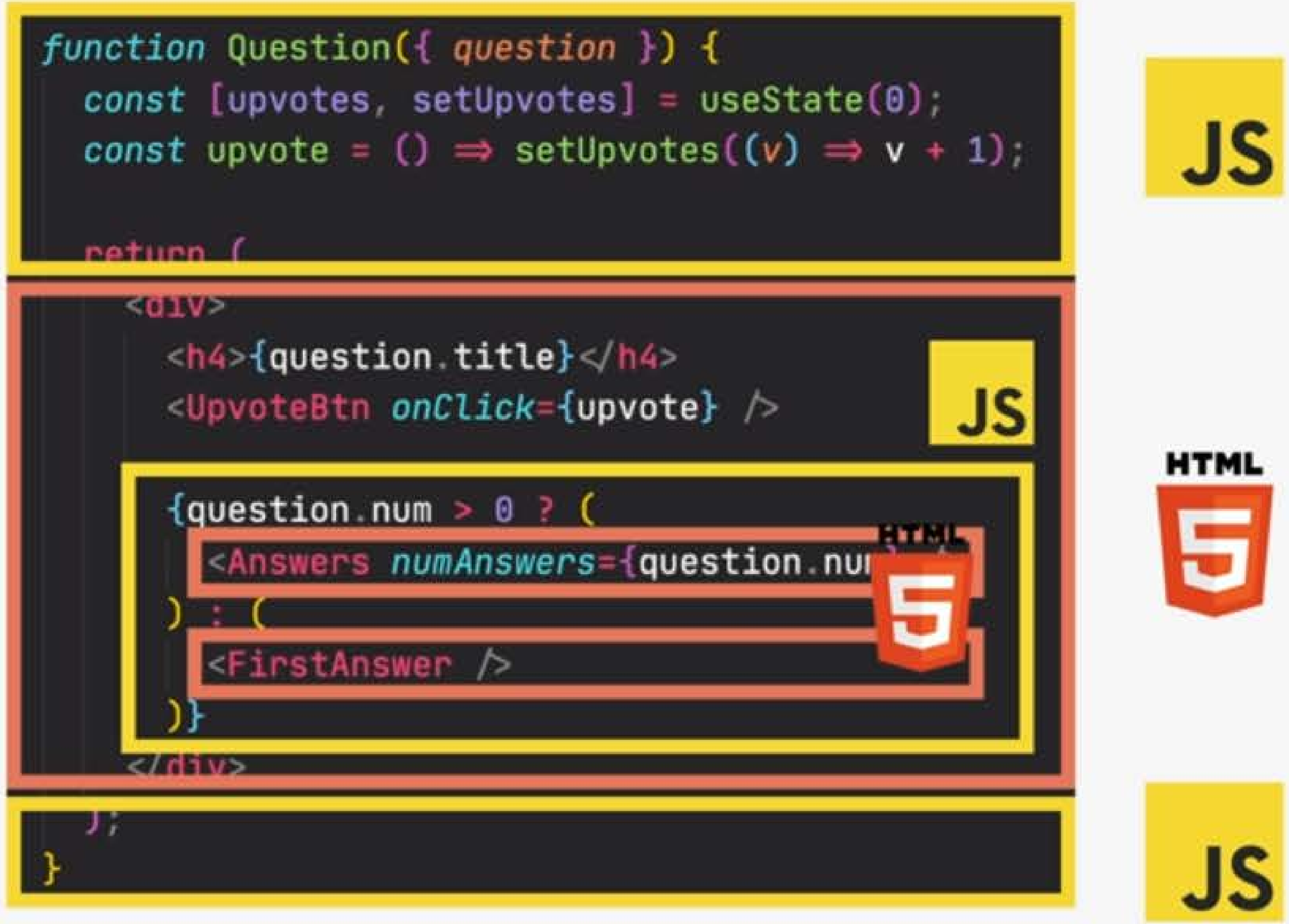
```
function Question({ question }) {
  const [upvotes, setUpvotes] = useState(0);
  const upvote = () ⇒ setUpvotes((v) ⇒ v + 1);

  return (
    <div>
      <h4>{question.title}</h4>
      <UpvoteBtn onClick={upvote} />

      {question.num > 0 ? (
        <Answers numAnswers={question.num} />
      ) : (
        <FirstAnswer />
      )}
    </div>
  );
}
```

Fundamental reason for **components**



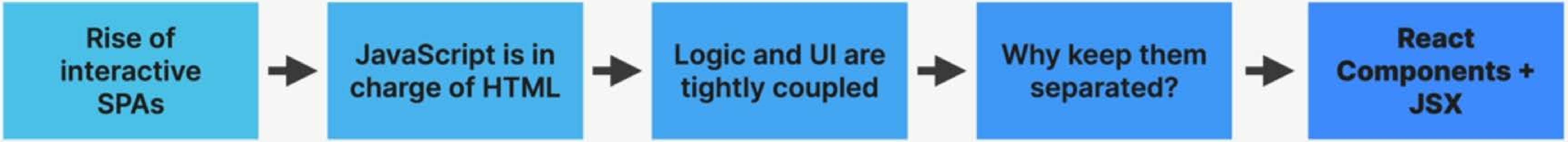
SEPARATION OF CONCERNS?



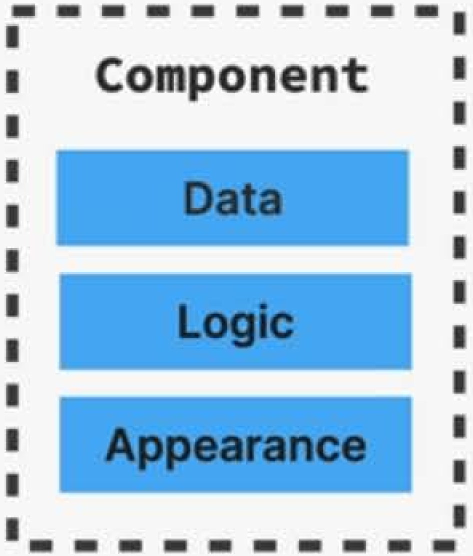


```
function Question({ question }) {
  const [upvotes, setUpvotes] = useState(0);
  const upvote = () => setUpvotes((v) => v + 1);

  return (
    <div>
      <h4>{question.title}</h4>
      <UpvoteBtn onClick={upvote} />

      {question.num > 0 ? (
        <Answers numAnswers={question.nu
      ) : (
        <FirstAnswer />
      )}
    </div>
  );
}
```

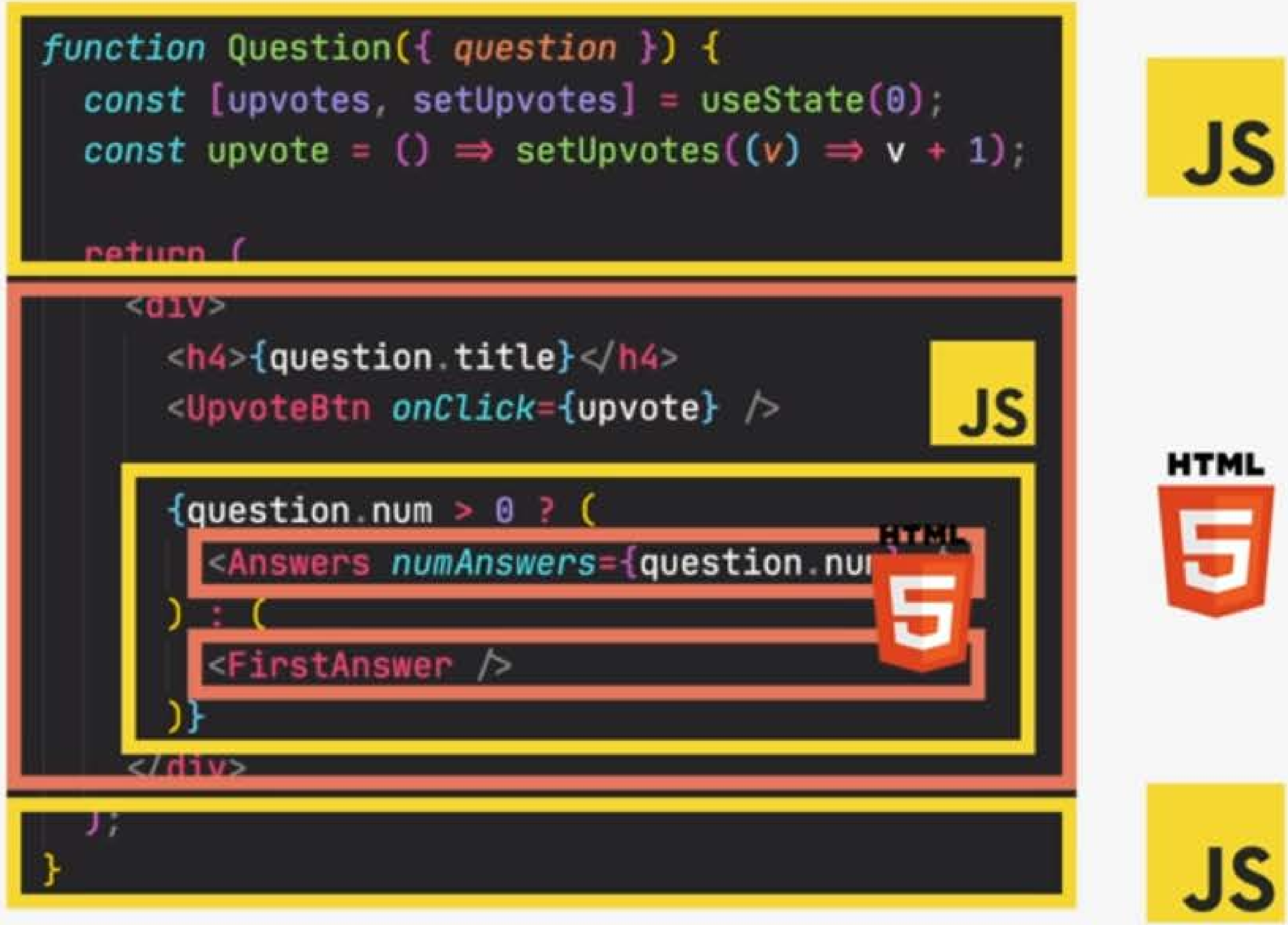
Fundamental reason for **components**



SEPARATION OF CONCERNS?

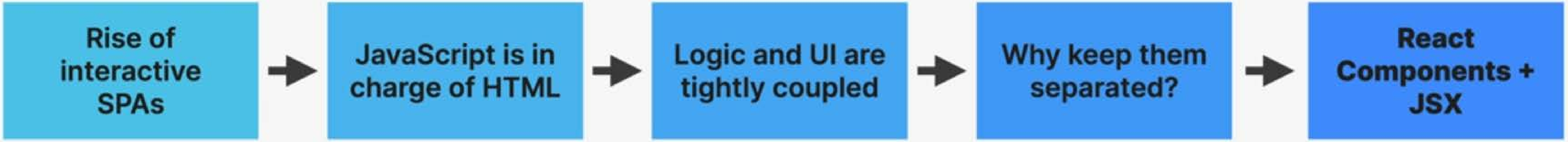


HTML and JS are *colocated*



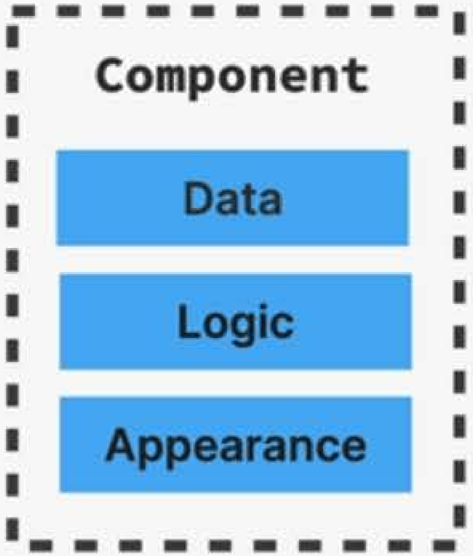
```
function Question({ question }) {
  const [upvotes, setUpvotes] = useState(0);
  const upvote = () => setUpvotes((v) => v + 1);

  return (
    <div>
      <h4>{question.title}</h4>
      <UpvoteBtn onClick={upvote} />

      {question.num > 0 ? (
        <Answers numAnswers={question.nu
      ) : (
        <FirstAnswer />
      )}
    </div>
  );
}
```

Fundamental reason for **components**

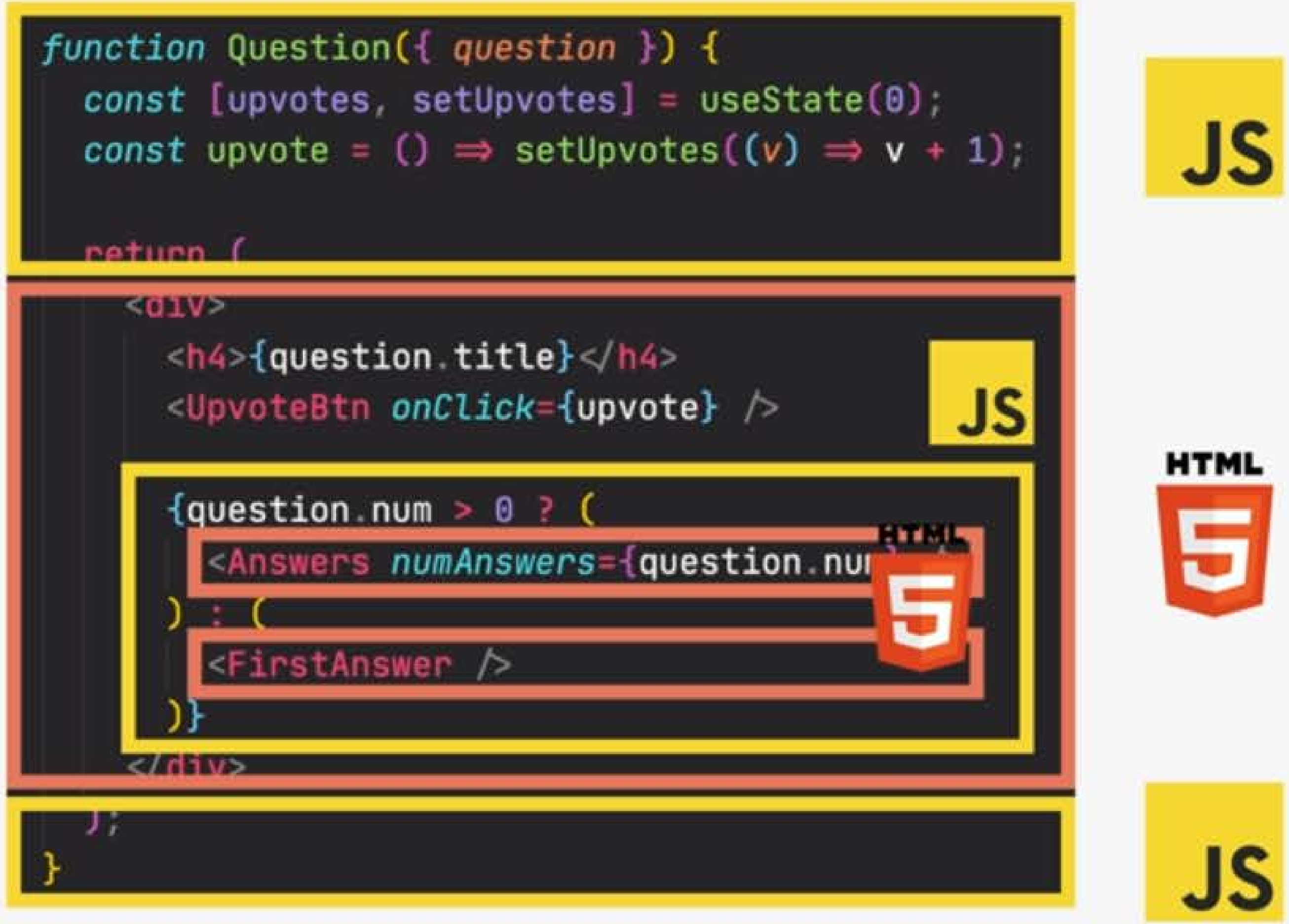


SEPARATION OF CONCERNS?

ONE *COMPONENT* PER FILE

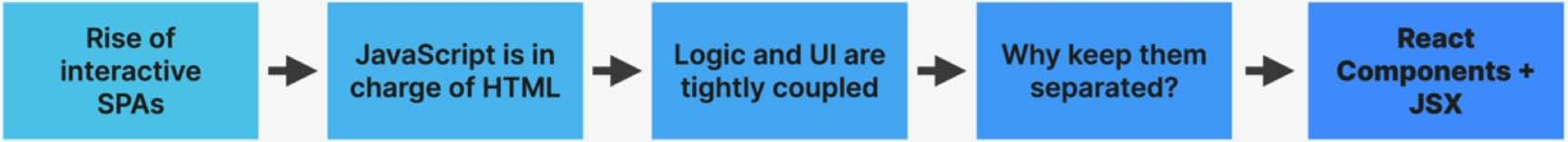


HTML and JS are ***colocated***



```
function Question({ question }) {
  const [upvotes, setUpvotes] = useState(0);
  const upvote = () => setUpvotes((v) => v + 1);

  return (
    <div>
      <h4>{question.title}</h4>
      <UpvoteBtn onClick={upvote} />

      {question.num > 0 ? (
        <Answers numAnswers={question.nu
      ) : (
        <FirstAnswer />
      )}
    </div>
  );
}
```

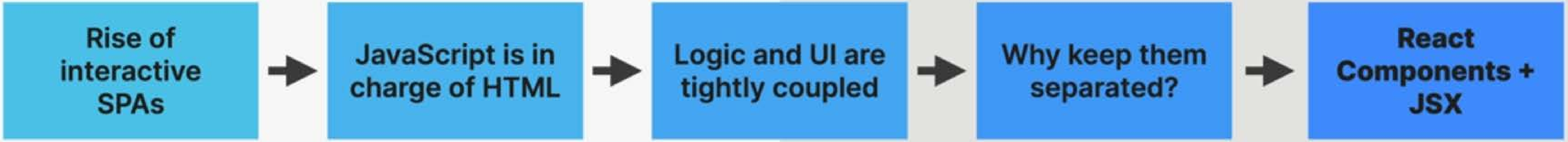
Fundamental reason for **components**



SEPARATION OF CONCERNS!

ONE *TECHNOLOGY* PER FILE

ONE *COMPONENT* PER FILE

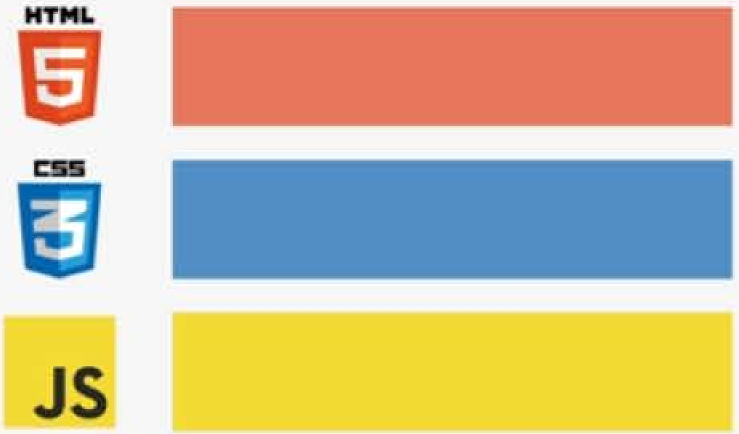


Rise of interactive SPAs → JavaScript is in charge of HTML → Logic and UI are tightly coupled → Why keep them separated? → React Components + JSX

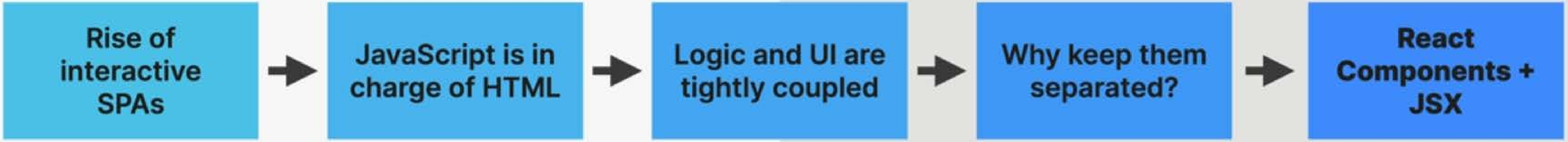
SEPARATION OF CONCERNS!

ONE *TECHNOLOGY* PER FILE

“Traditional” **separation of concerns**

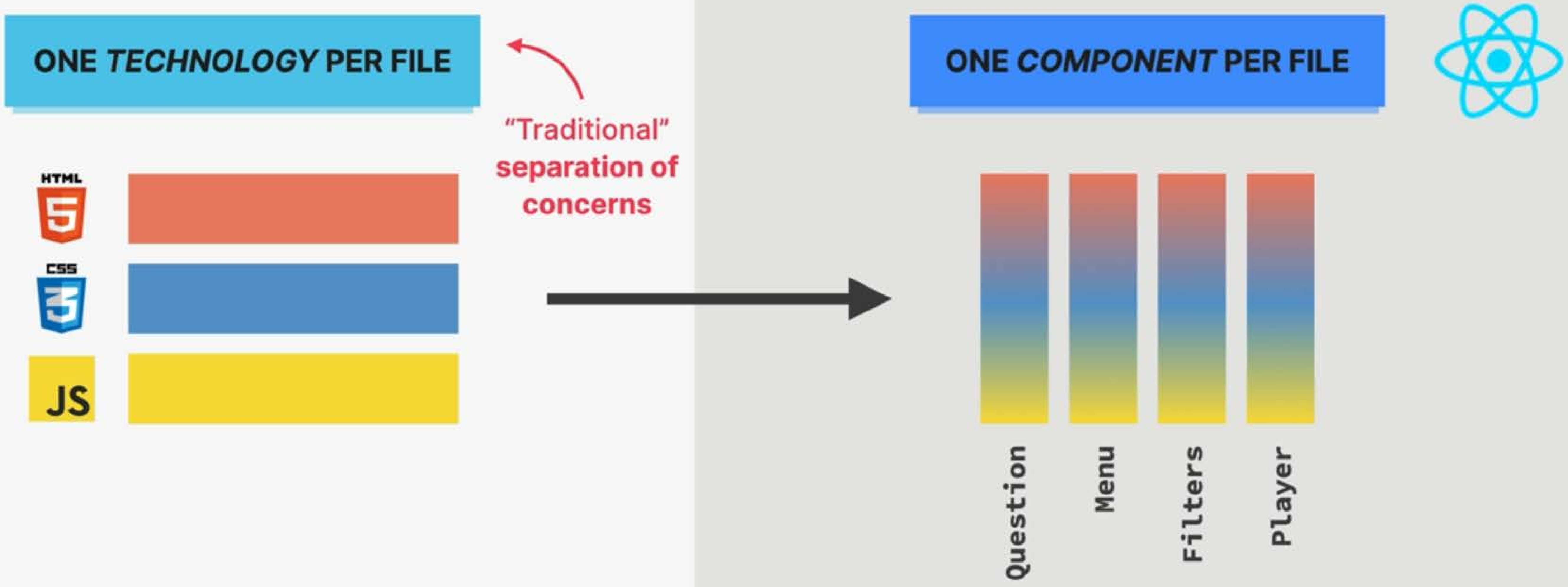


ONE *COMPONENT* PER FILE



Rise of interactive SPAs → JavaScript is in charge of HTML → Logic and UI are tightly coupled → Why keep them separated? → React Components + JSX

SEPARATION OF CONCERNS!



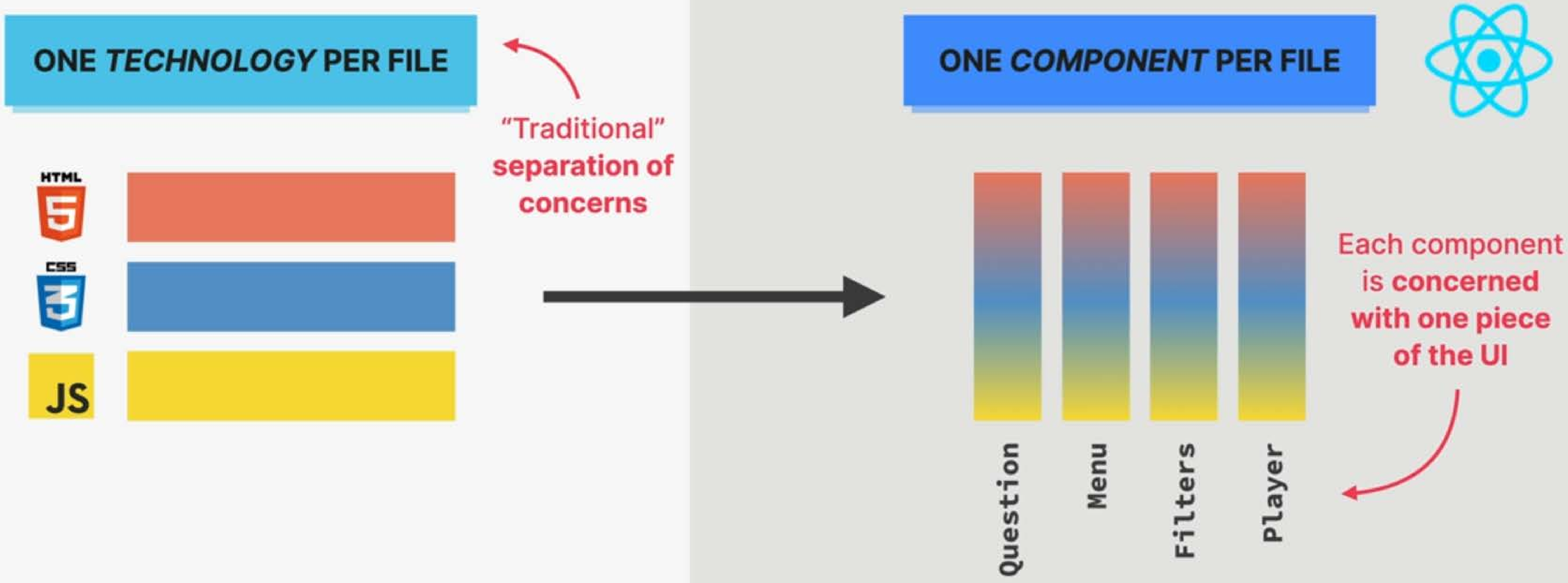
ONE *TECHNOLOGY* PER FILE

"Traditional" **separation of concerns**

ONE *COMPONENT* PER FILE

Rise of interactive SPAs → JavaScript is in charge of HTML → Logic and UI are tightly coupled → Why keep them separated? → **React Components + JSX**

SEPARATION OF CONCERNS!

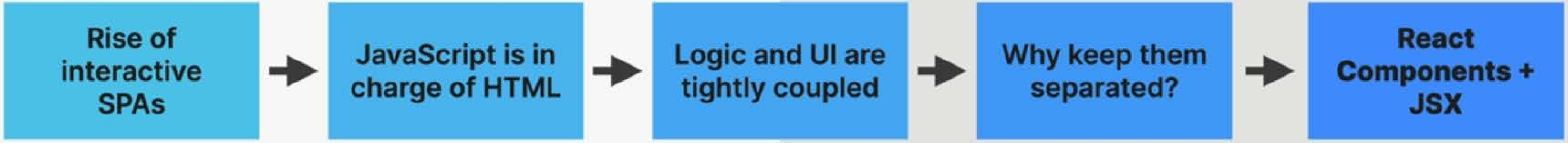


ONE *TECHNOLOGY* PER FILE

"Traditional" **separation of concerns**

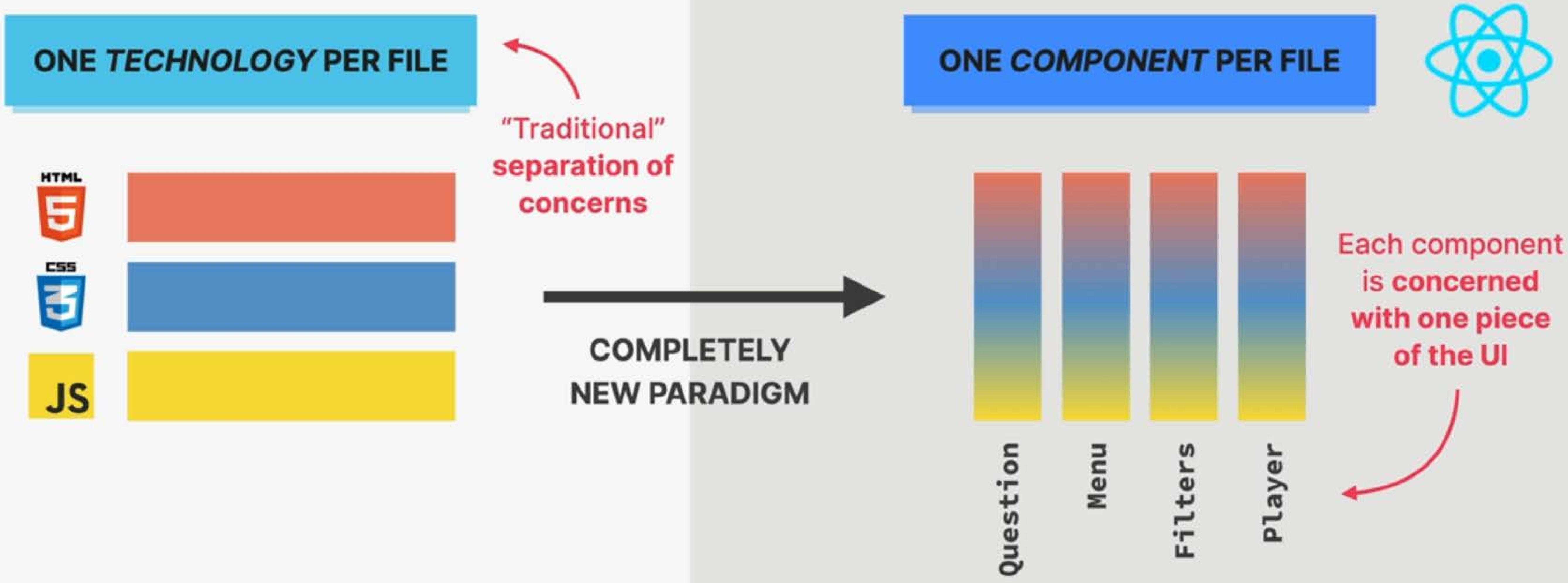
ONE *COMPONENT* PER FILE

Each component **is concerned with one piece of the UI**



Rise of interactive SPAs → **JavaScript is in charge of HTML** → **Logic and UI are tightly coupled** → **Why keep them separated?** → **React Components + JSX**

SEPARATION OF CONCERNS!



ONE *TECHNOLOGY* PER FILE

“Traditional” **separation of concerns**

COMPLETELY NEW PARADIGM

ONE *COMPONENT* PER FILE

Each component **is concerned with one piece of the UI**

Rise of interactive SPAs → **JavaScript is in charge of HTML** → **Logic and UI are tightly coupled** → **Why keep them separated?** → **React Components + JSX**