



„Sieht aus wie Flutter, ist es aber nicht.“

Die Architektur eines Dart Web Frameworks.

Kilian Schulte
@schultek_dev



Jasper



Kilian Schulte



@schultek



@schultek_dev

Was ist Jaspr?



Jaspr ist ein Dart
Framework zum bauen
von Websiten.

MOBILAPP

Home Blog Contact Us

Power up your business with a Flutter mobile app.

Grow your business with cross-platform apps crafted by experts.

[Get in touch](#)

Follow us: [In](#) [X](#) [R](#) [S](#) [Y](#) [G](#) [F](#)

[Join our community](#)

flutteristas

Home Flutteristas Conference Flutteristas Map Code of Conduct Privacy Policy

Who we are?

The Flutteristas is a world-wide community of people whose gender identity is either female or non-binary and have an interest in the Flutter SDK. ❤️

[Apply now](#)

Home Articles Courses Projects Contact

Articles

Here you can find all my articles on various topics related to software development and technology.

Aug 21, 2024 Article

Comprehensive Guide to Testing Riverpod Providers

Guide to testing Riverpod providers in Flutter: ProviderContainer, utilities, and best practices for robust state...

[Read article >](#)

Aug 15, 2024 Article

Taming the Beast: Mastering Forms in Flutter

A comprehensive guide to handling forms in Flutter, including validation, state management, and best practices

[Read article >](#)

Aug 10, 2024 Article

Testing Permission Handling in Flutter: A Comprehensive Guide

Learn how to effectively test permission handling in Flutter applications, ensuring robust and reliable functionality

[Read article >](#)

Home Articles Courses Projects © 2024. All rights reserved.

JasprPad New Pad Reset Format Download Project Weather Api Tutorial Samples

main.dart api.dart styles.css Run

```

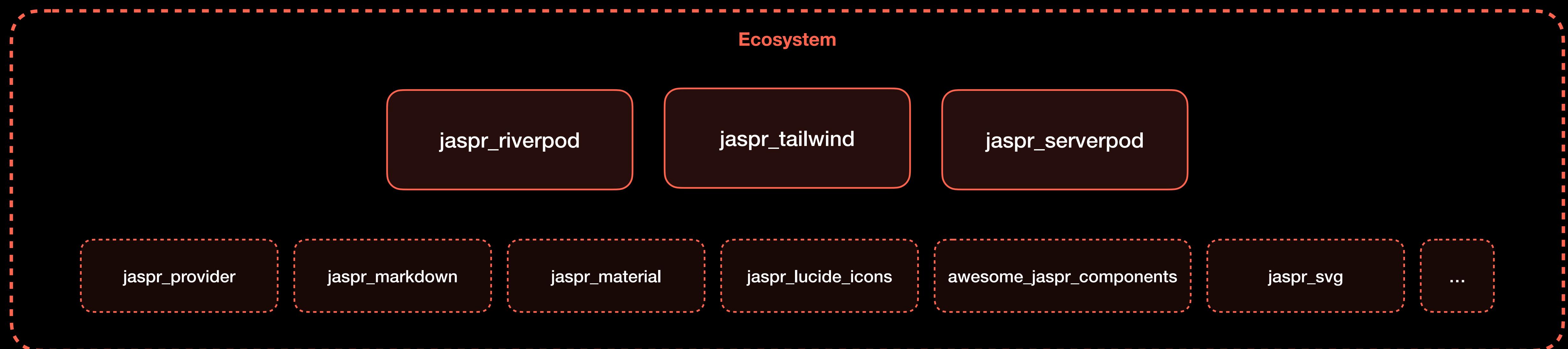
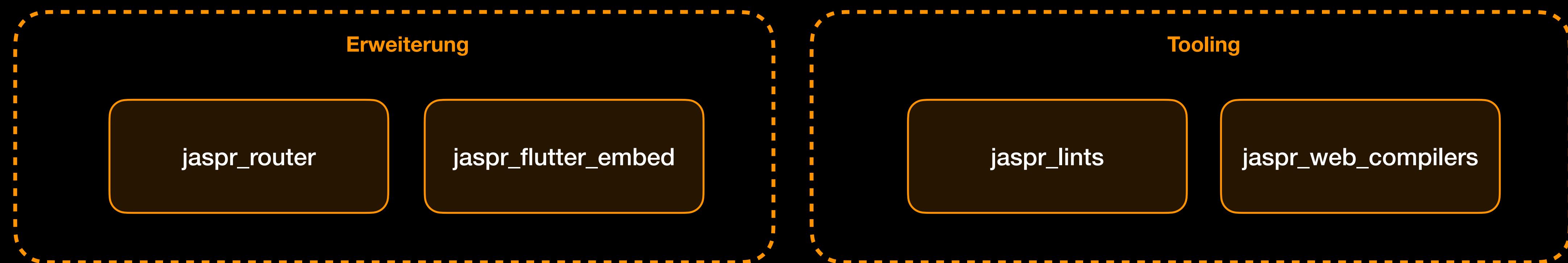
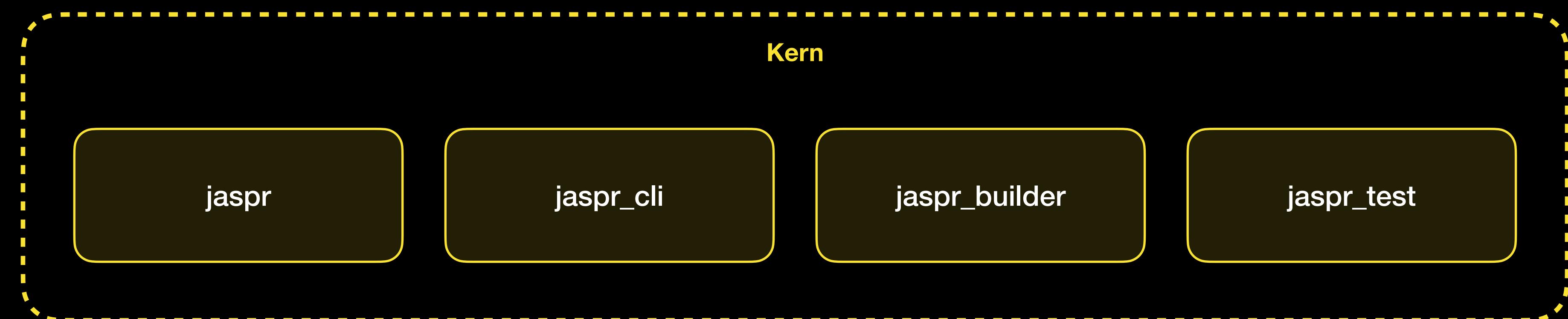
1 import 'package:jaspr/jaspr.dart';
2 import 'api.dart';
3 void main() {
4   runApp(App());
5 }
6 class App extends StatefulWidget {
7   const App({super.key});
8   @override
9   State<App> createState() => _AppState();
10 }
11 class _AppState extends State<App> {
12   final WeatherApi api = WeatherApi();
13   late Future<CurrentWeather> weatherFuture;
14   @override
15   void initState() {
16     super.initState();
17     updateWeather('Munich');
18   }
19   void updateWeather(String location) {
20     weatherFuture = api.getWeather(location);
21   }
22   @override
23   Iterable<Component> build(BuildContext context) sync* {
24     yield SearchBar(
25       controller: TextEditingController(),
26       onSearch: (query) {
27         setState(() {
28           weatherFuture = api.getWeather(query);
29         });
30       },
31     );
32   }
33 }
```

Enter Location Search 10.1° Munich, Germany

Jaspr Docs Join the Community Send feedback



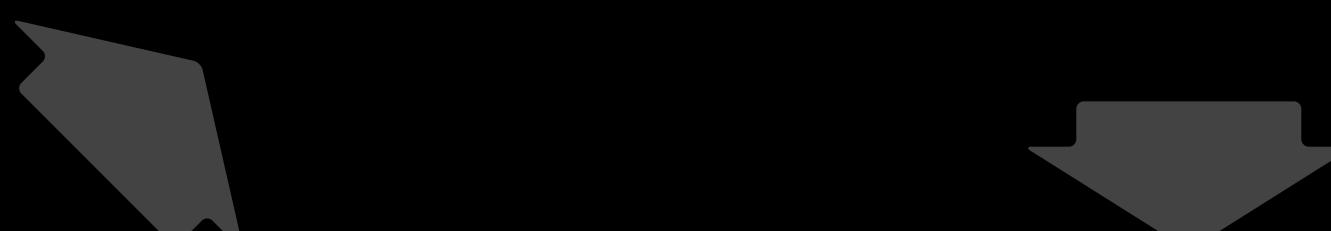
Wie ist Jaspr aufgebaut?



```
class FooComponent extends StatelessWidget {  
  const FooComponent({super.key});  
  
  @override  
  Iterable<Widget> build(BuildContext context) sync* {  
    yield div([  
      p([text('Hello World')]),  
    ]);  
    yield img(src: '/images/logo.png');  
  }  
}
```

```
class FooComponent extends StatelessWidget {  
  const FooComponent({super.key});  
  
  @override  
  Iterable<Widget> build(BuildContext context) sync* {  
    yield div([  
      p([text('Hello World')]),  
    ]);  
    yield img(src: '/images/logo.png');  
  }  
}
```

```
Node a = document.createElement('p');  
a.text = 'Hello World';  
Node b = document.createElement('div');  
b.append(a);  
Node c = document.createElement('img');  
c.setAttribute('src', '/images/logo.png');
```



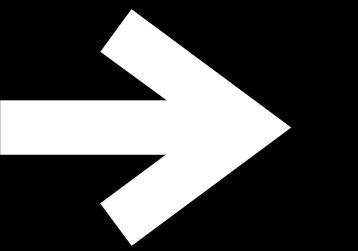
```
<div>  
  <p>Hello World</p>  
</div>  

```

Flutters Rendering System

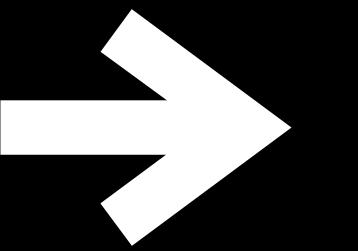
Widget

Konfiguriert ein UI
element.



Element

Repräsentiert ein UI
element.



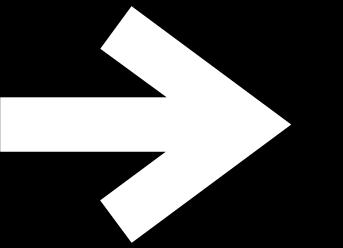
RenderObject

Rendert UI

Jasprs UI Rendering System

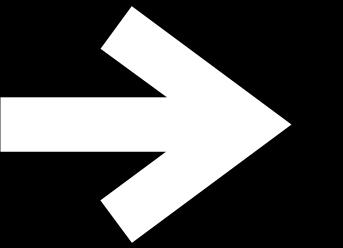
Component

Konfiguriert ein UI element.



Element

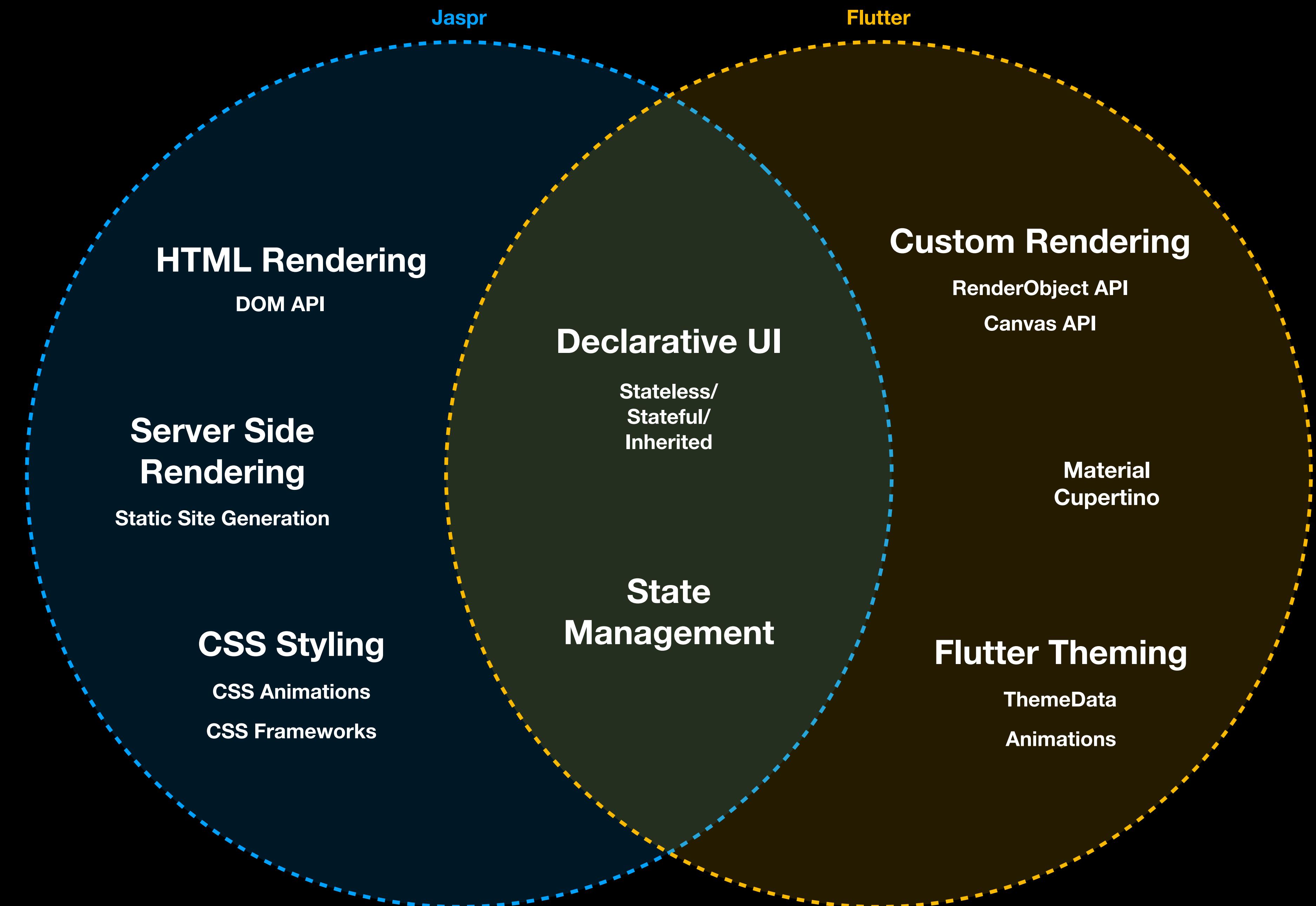
Repräsentiert ein UI element.



DOM Node

Rendert UI

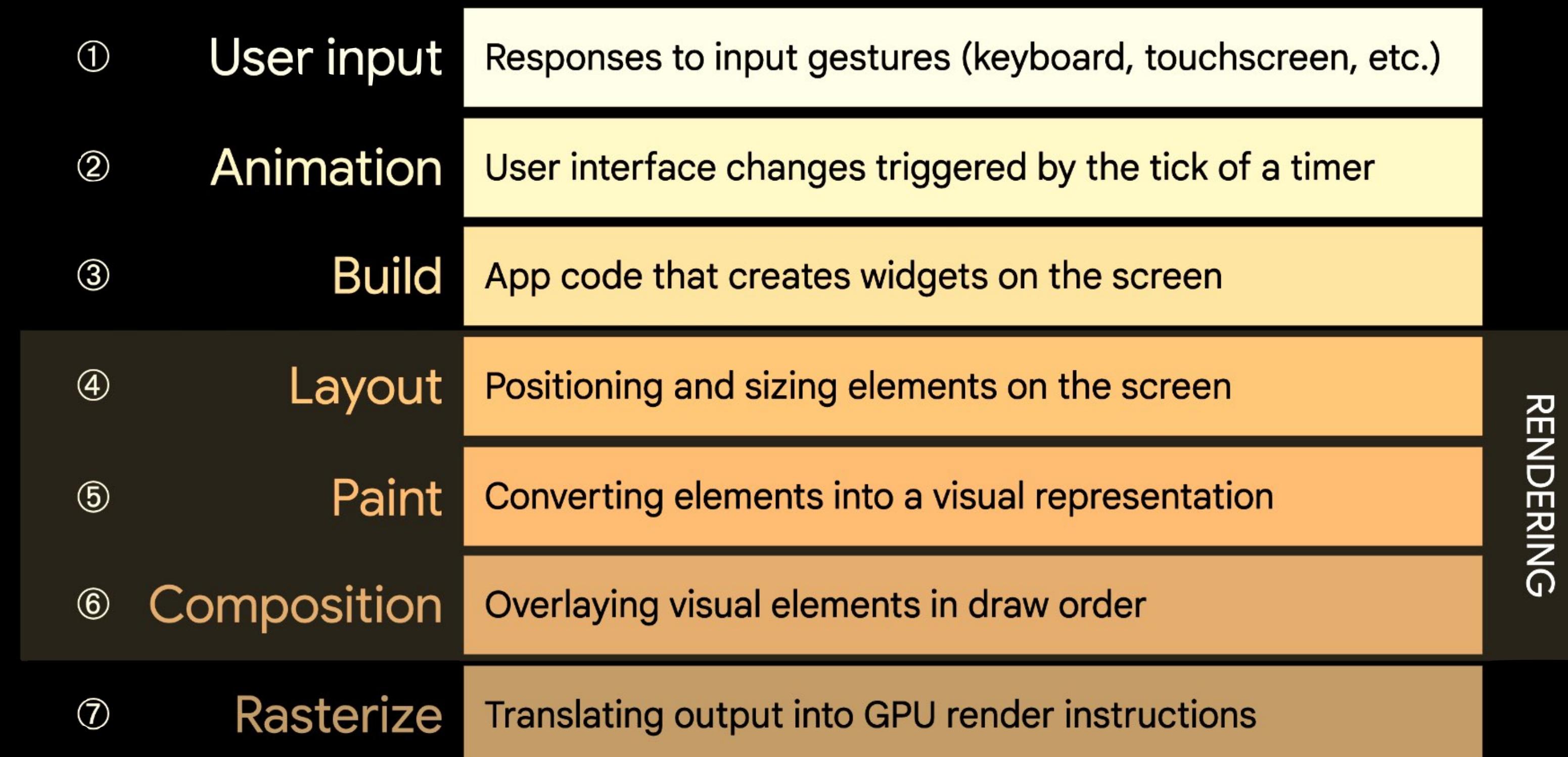




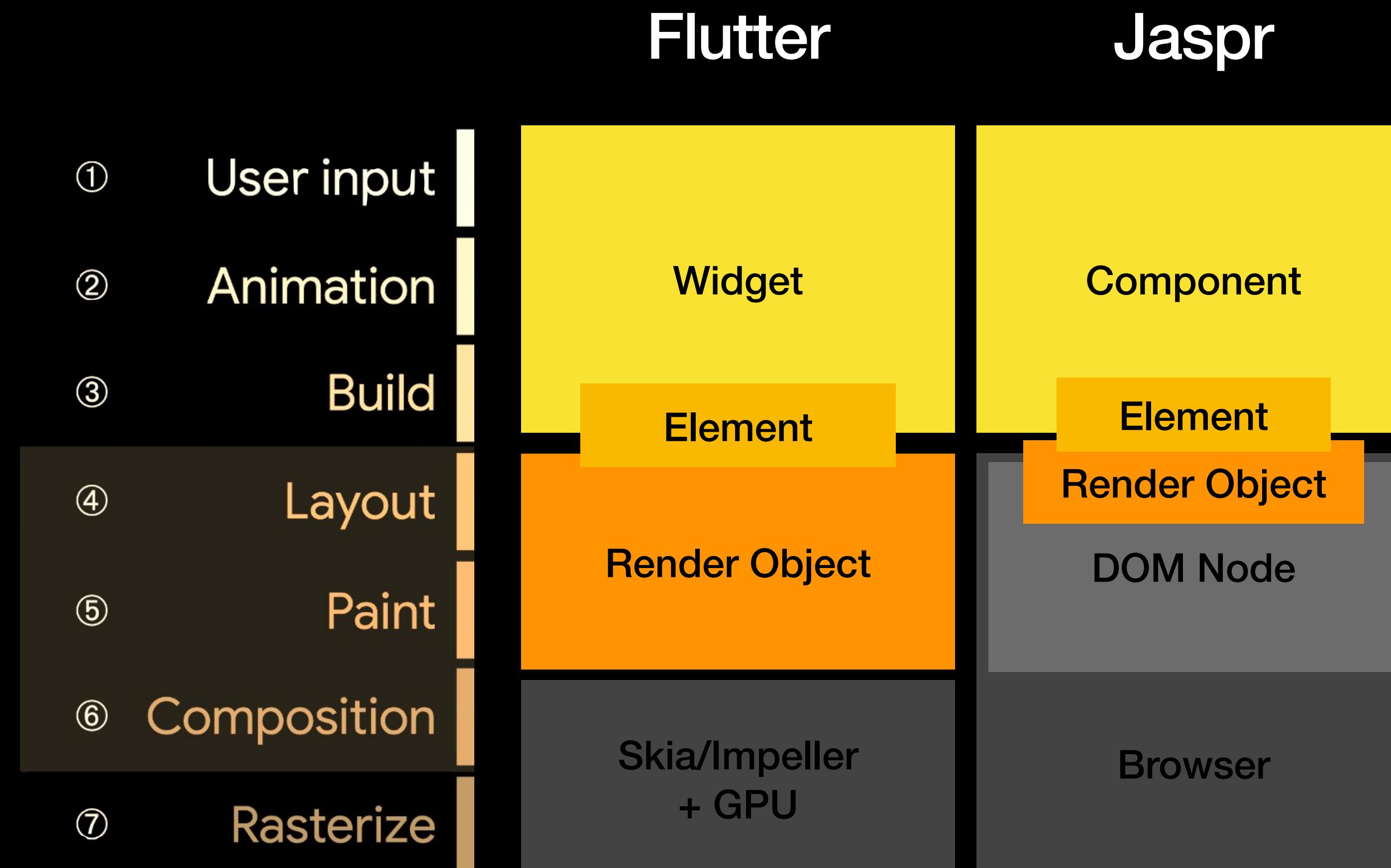


Wie funktioniert Rendering
in Jaspr?

Rendering Pipeline

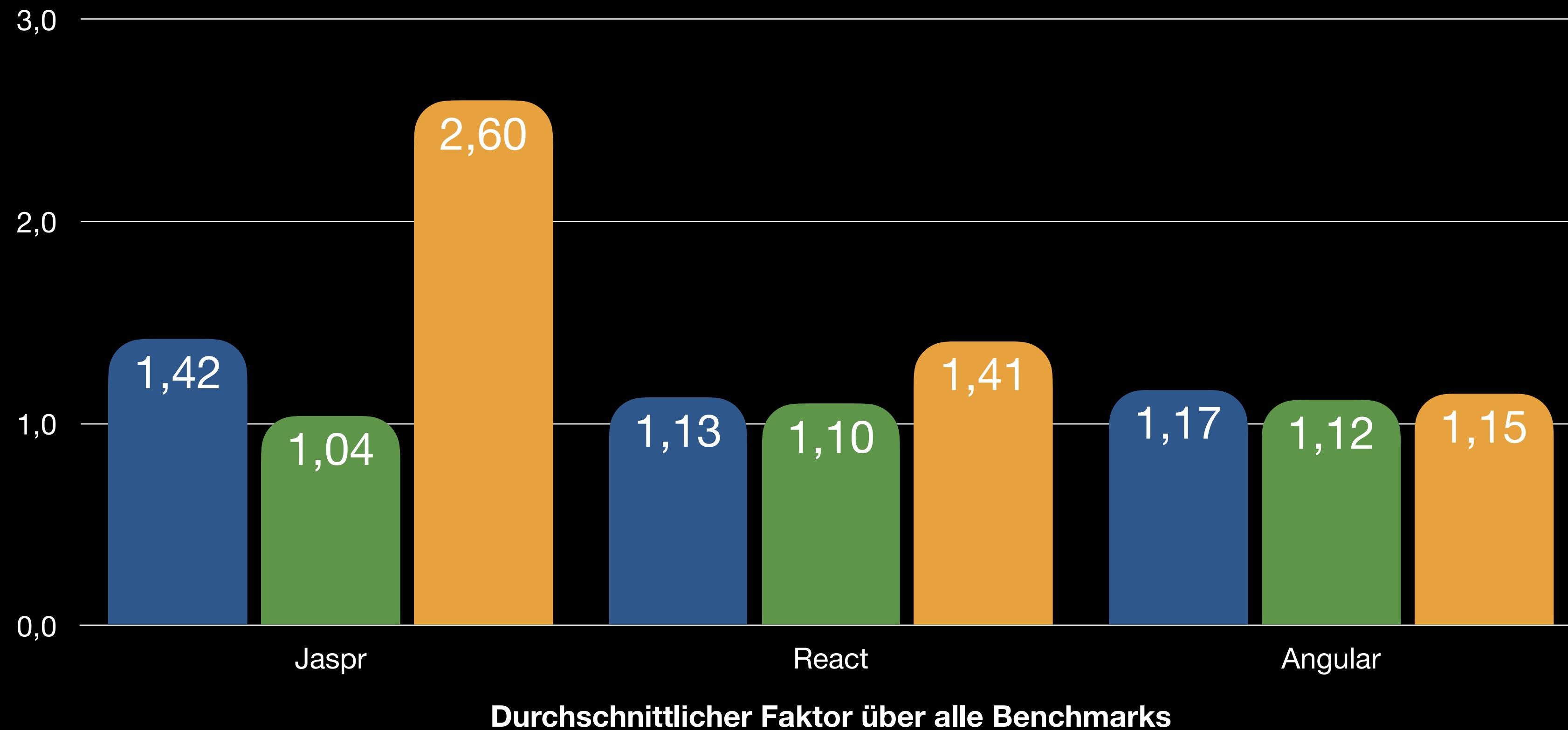


Rendering Pipeline



Performance?

<https://jaspr-benchmarks.web.app>



1.0 == Überall am schnellsten, 1.5 == 50% langsamer als der Schnellste

Allgemeine Performance:

~20% langsamer als R / A

Zeit die aufs Rendering fällt:

~5% schneller als R / A

Zeit die auf Javascript fällt:

~45% / ~55% langsamer als R / A



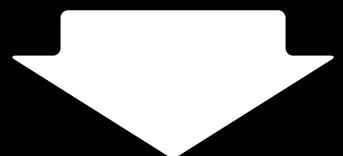
Wie geht eigentlich
Server Side Rendering?

```
runApp (FooComponent);
```

Jaspr

Pre-rendering

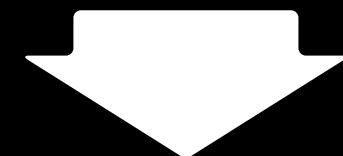
im
server



HTML Markup
(als String)

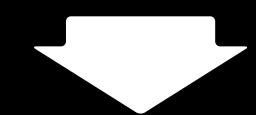
Rendering

im
client



DOM Nodes

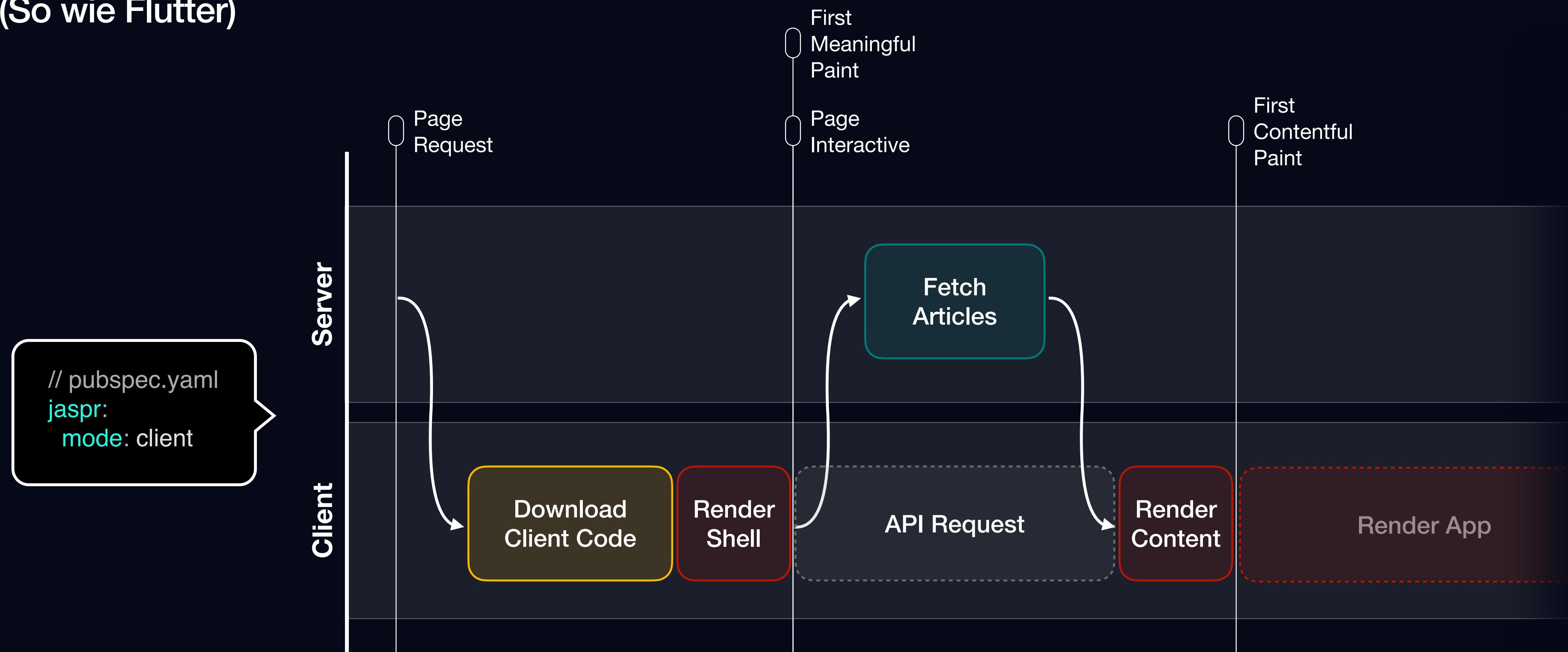
Browser



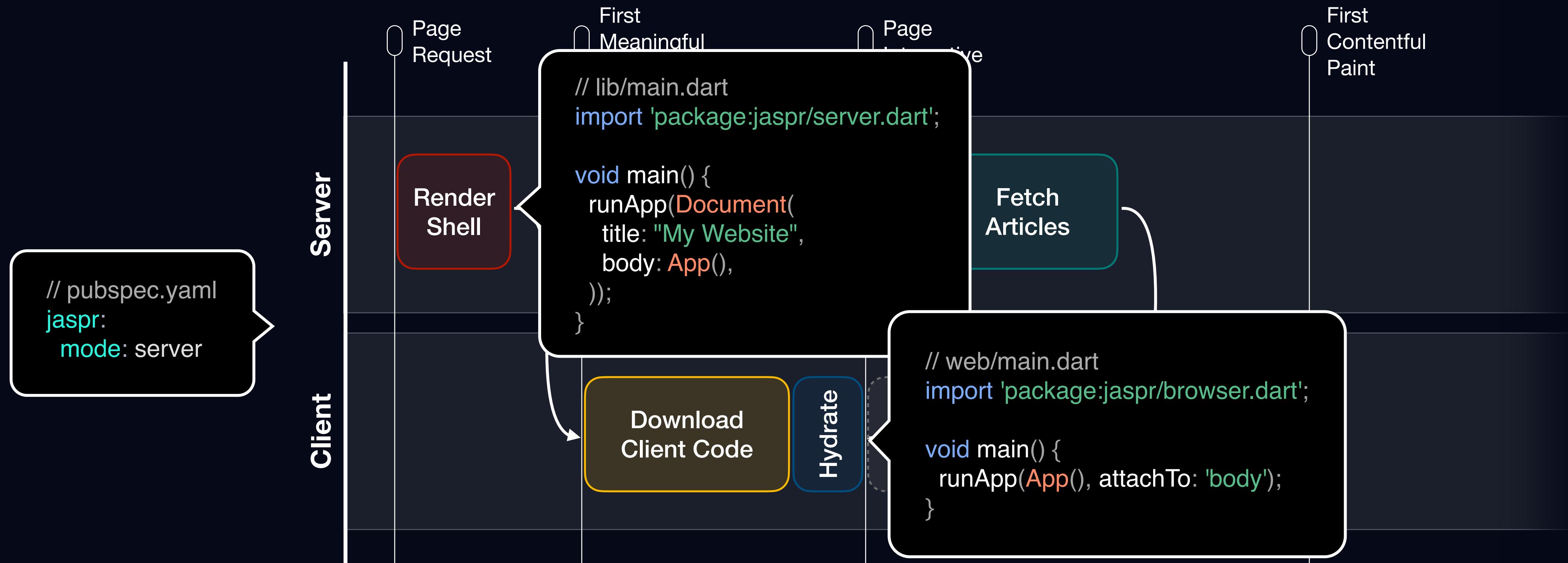
HTML Elements

Client Side Rendering

(So wie Flutter)



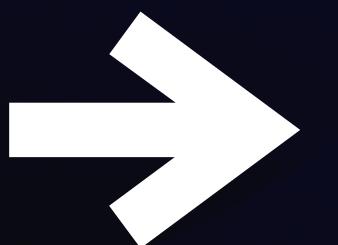
Server Side Rendering



Jasprs Rendering System

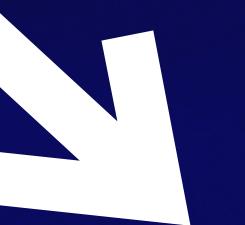
Component

Konfiguriert ein UI element.



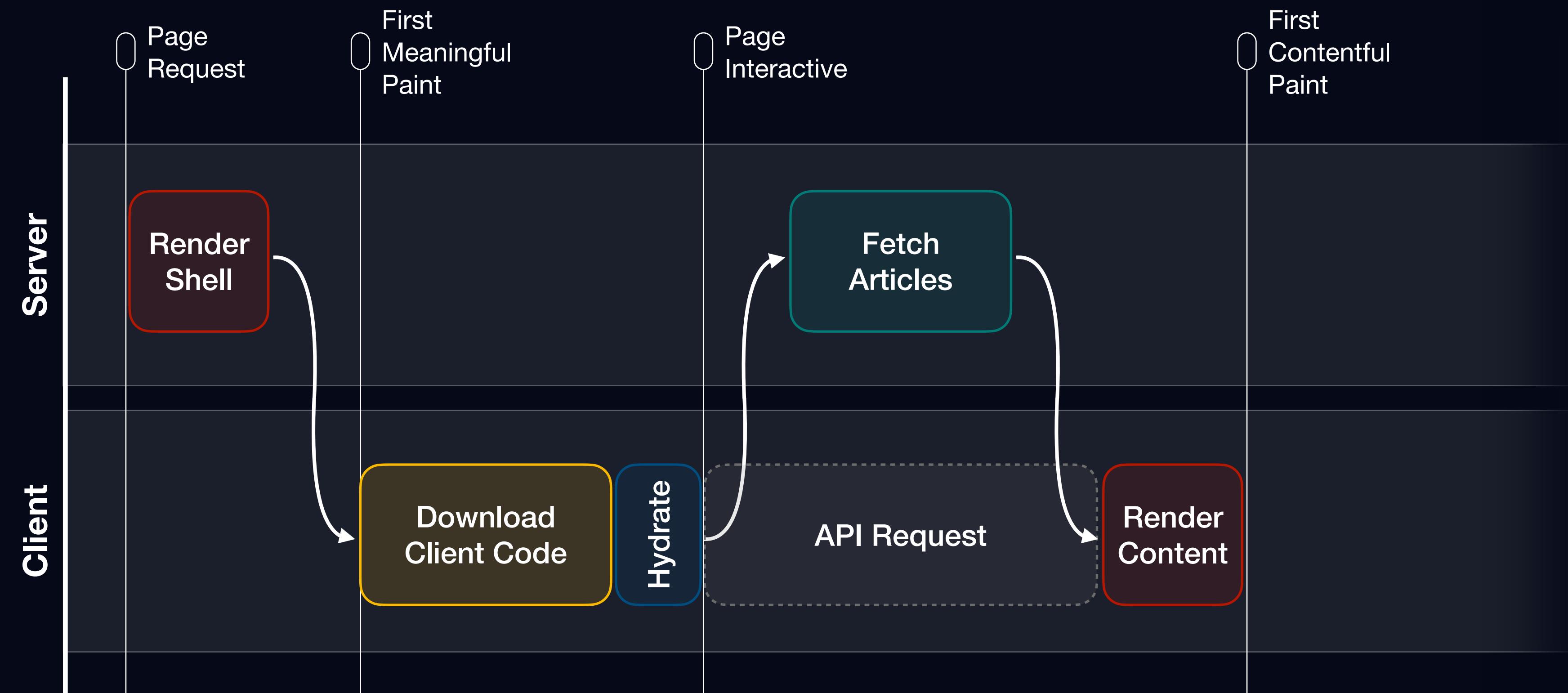
Element

Repräsentiert ein UI element.



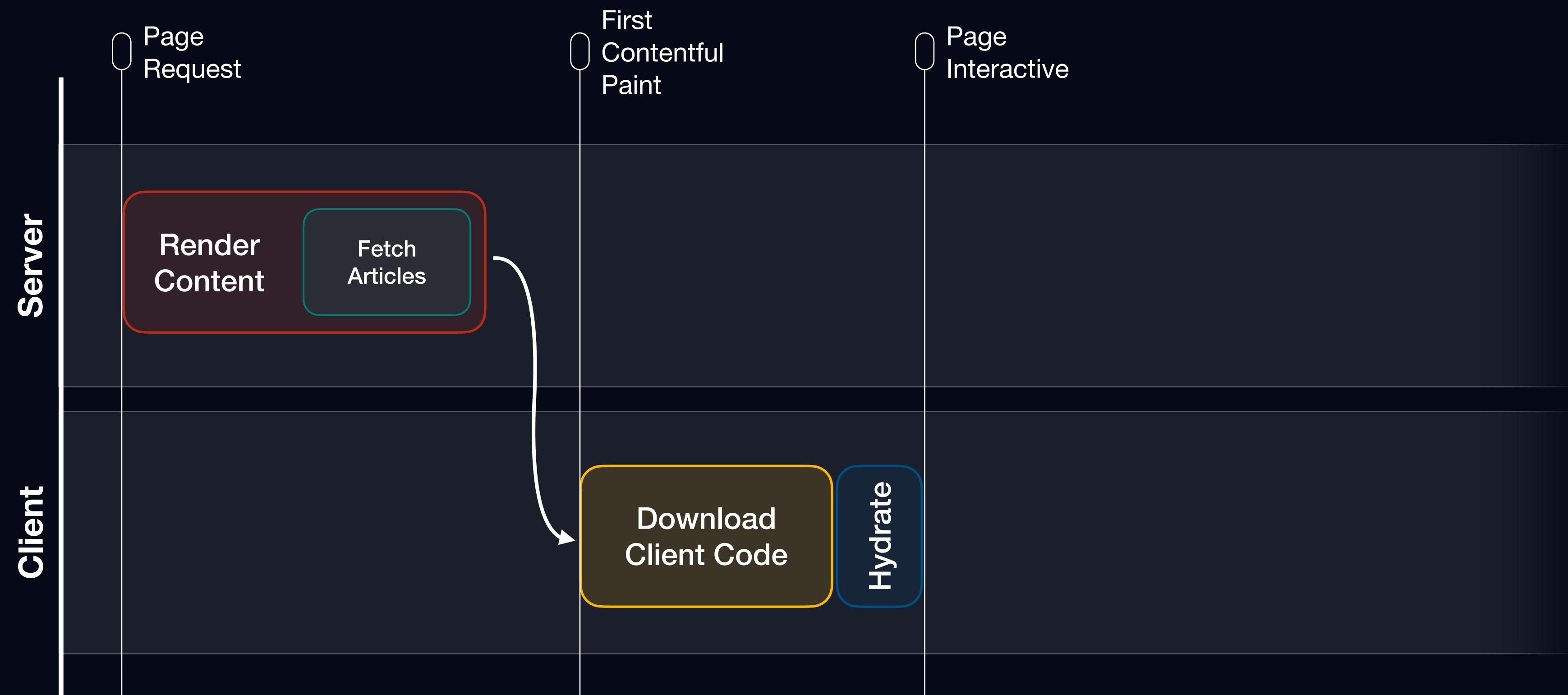
Server Side Rendering

(Die schlechte Art)



Server Side Rendering

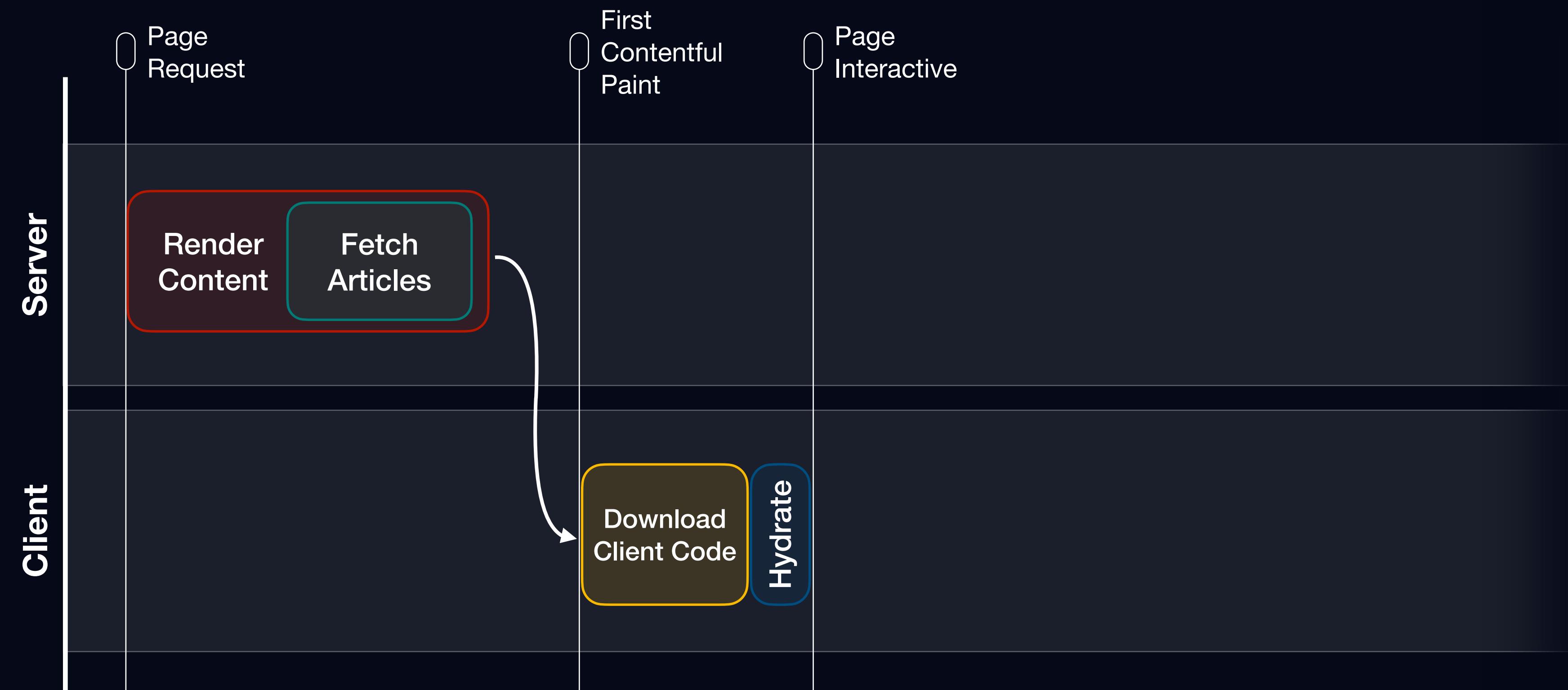
(Die gute Art)



AsyncComponent

```
class ArticlesList extends AsyncStatelessComponent {  
  const ArticlesList({super.key});  
  
  Stream<Component> build(BuildContext context) async* {  
    var articles = await fetchArticlesFromDatabase();  
  
    yield ul([  
      for (var article in articles)  
        li([  
          ArticleEntry(article: article),  
        ]),  
    ]);  
  }  
}
```

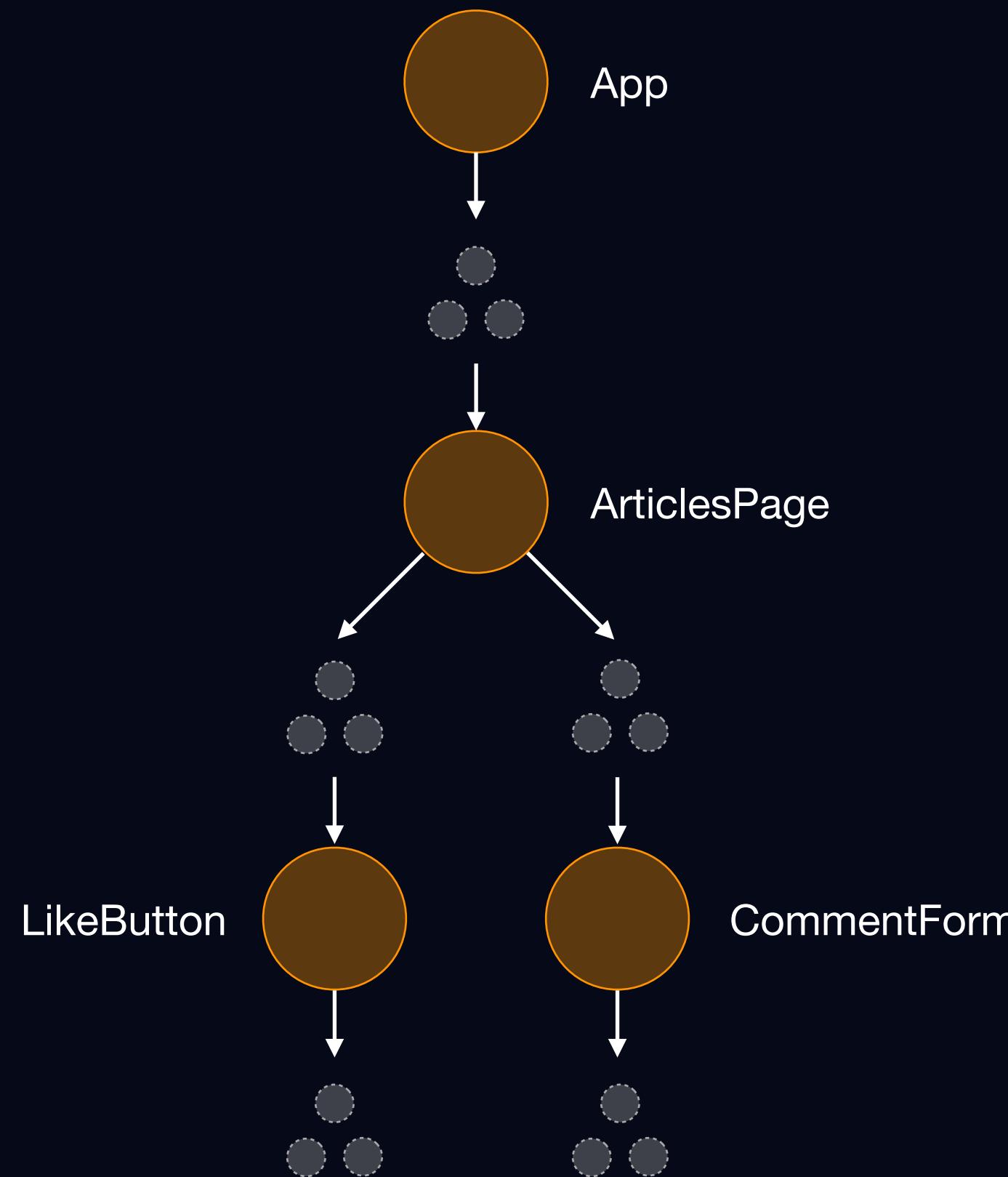
Server Side Rendering



```
// lib/main.dart
```

```
void main() {
  runApp(Document(
    title: "My Website",
    body: App(),
  ));
}
```

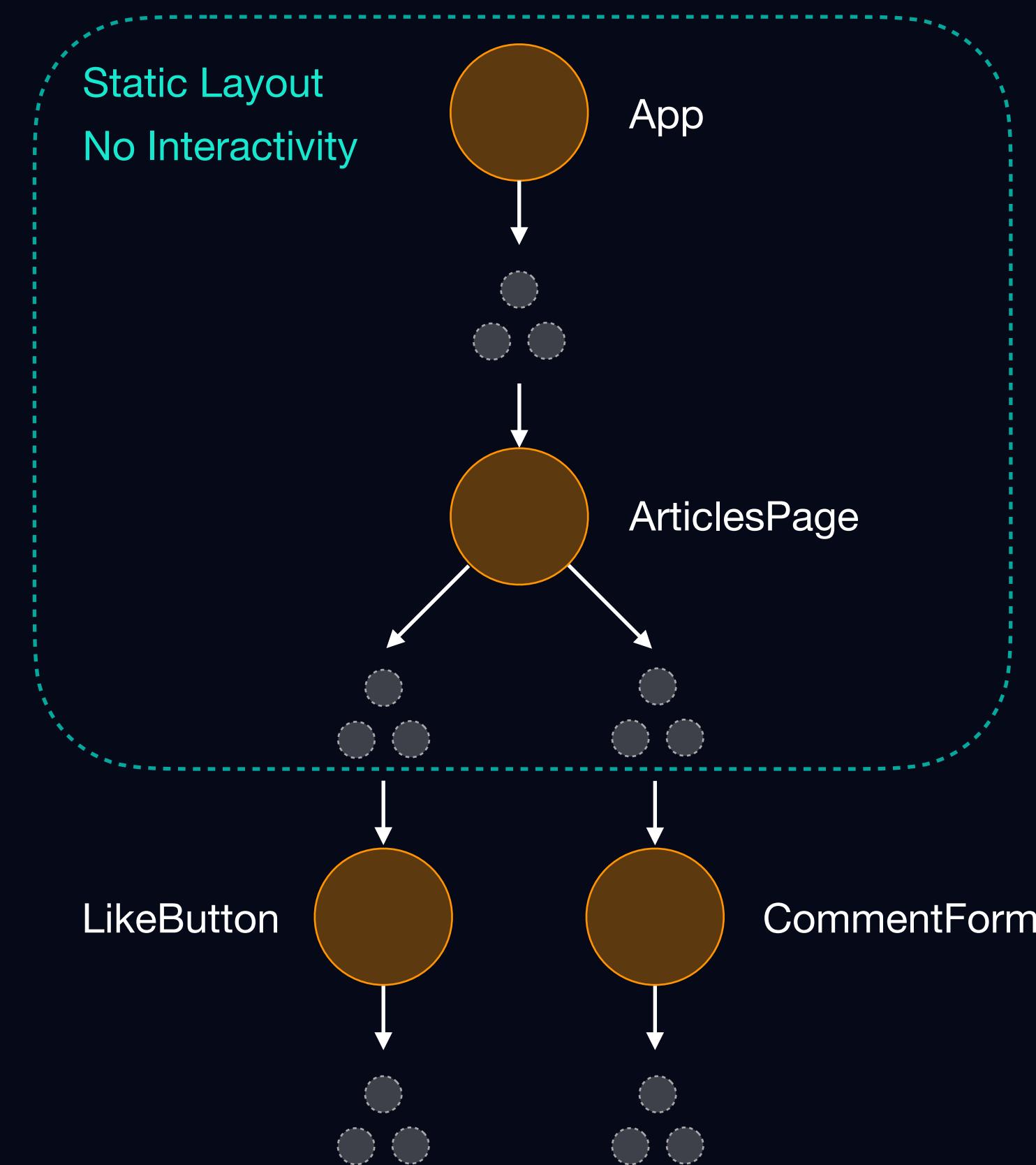
Server



Client

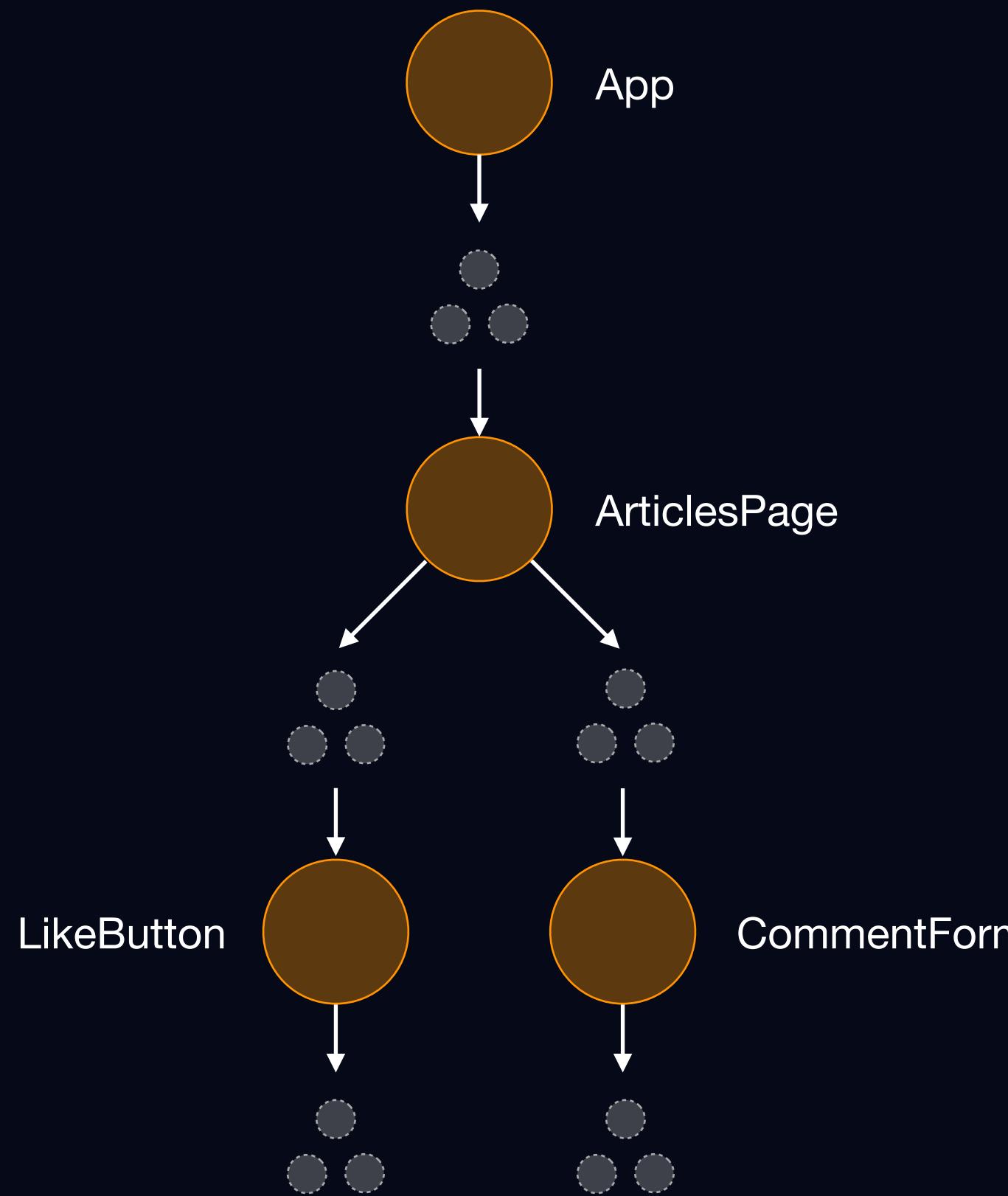
```
// web/main.dart
```

```
void main() {
  runApp(App(), attachTo: 'body');
}
```



```
// lib/main.dart
```

```
void main() {
  runApp(Document(
    title: "My Website",
    body: App(),
  ));
}
```



Server

Client

```
// lib/components/like_button.dart
```

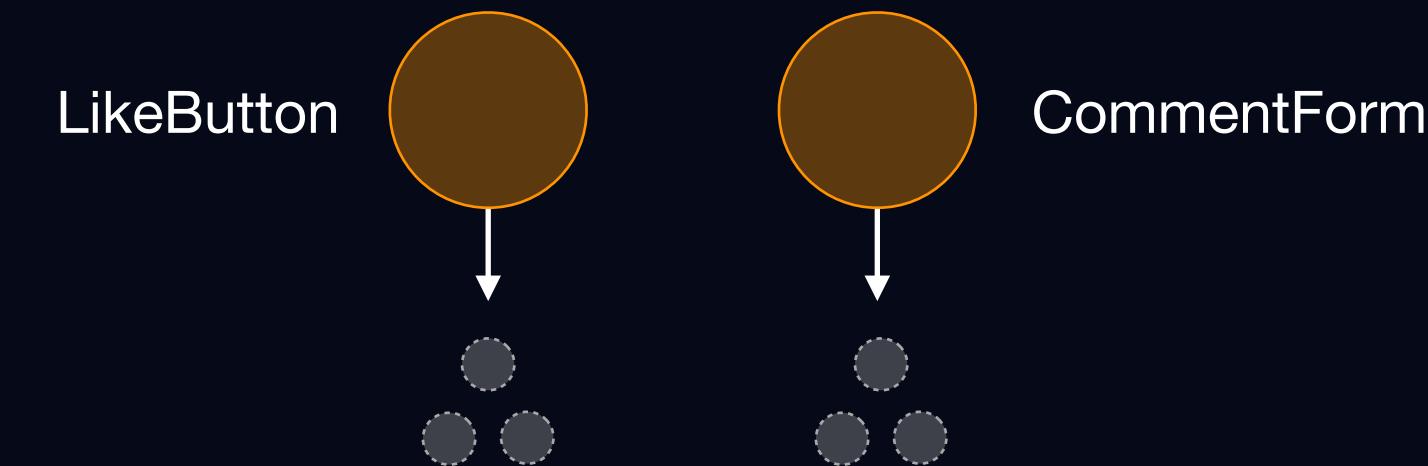
```
@client
class LikeButton extends StatelessWidget {
  const LikeButton({super.key});

  /* ... */
}
```

```
// lib/components/comment_form.dart
```

```
@client
class CommentForm extends StatelessWidget {
  const CommentForm({super.key});

  /* ... */
}
```



```
// lib/components/like_button.dart

{@client
class LikeButton extends StatelessWidget {
  const LikeButton({super.key});

  /* ... */
}}
```

Generiert

```
// web/components/like_button.client.dart (generated)

void main() {
  runApp(
    LikeButton(),
    id: 'components/like_button',
  );
}
```

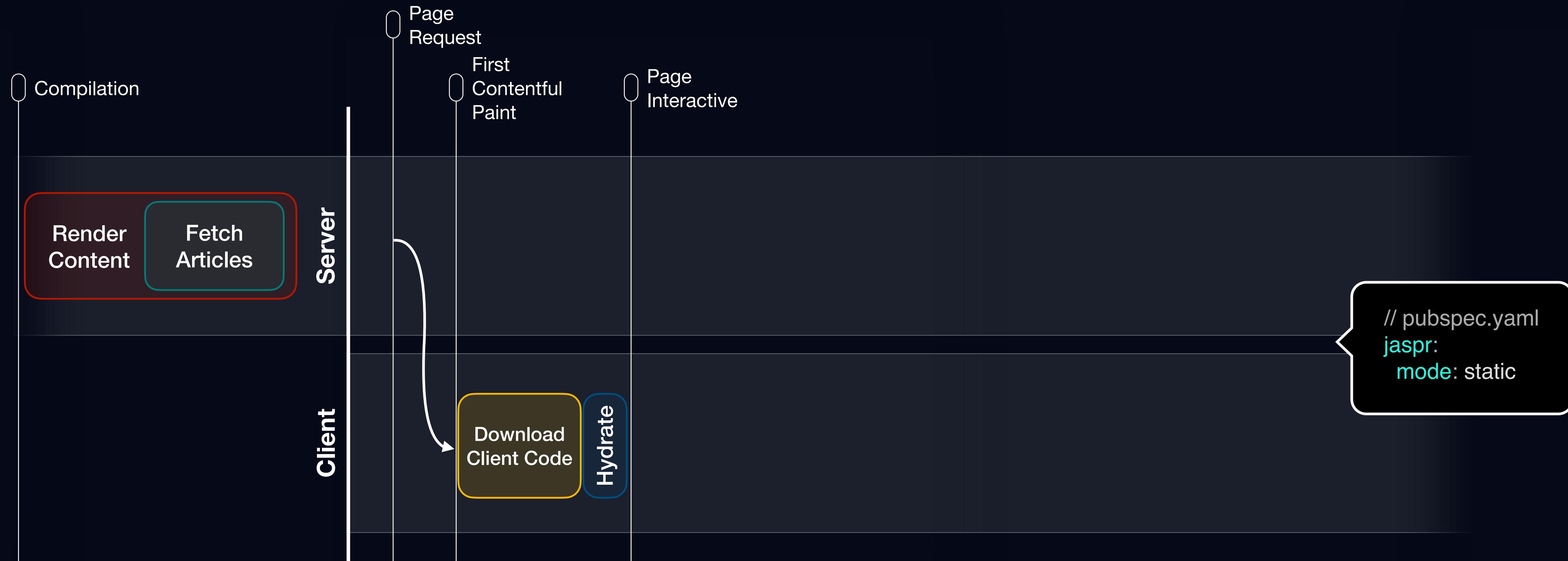
Pre-rendered

```
<script src="components/like_button.client.dart.js"></script>

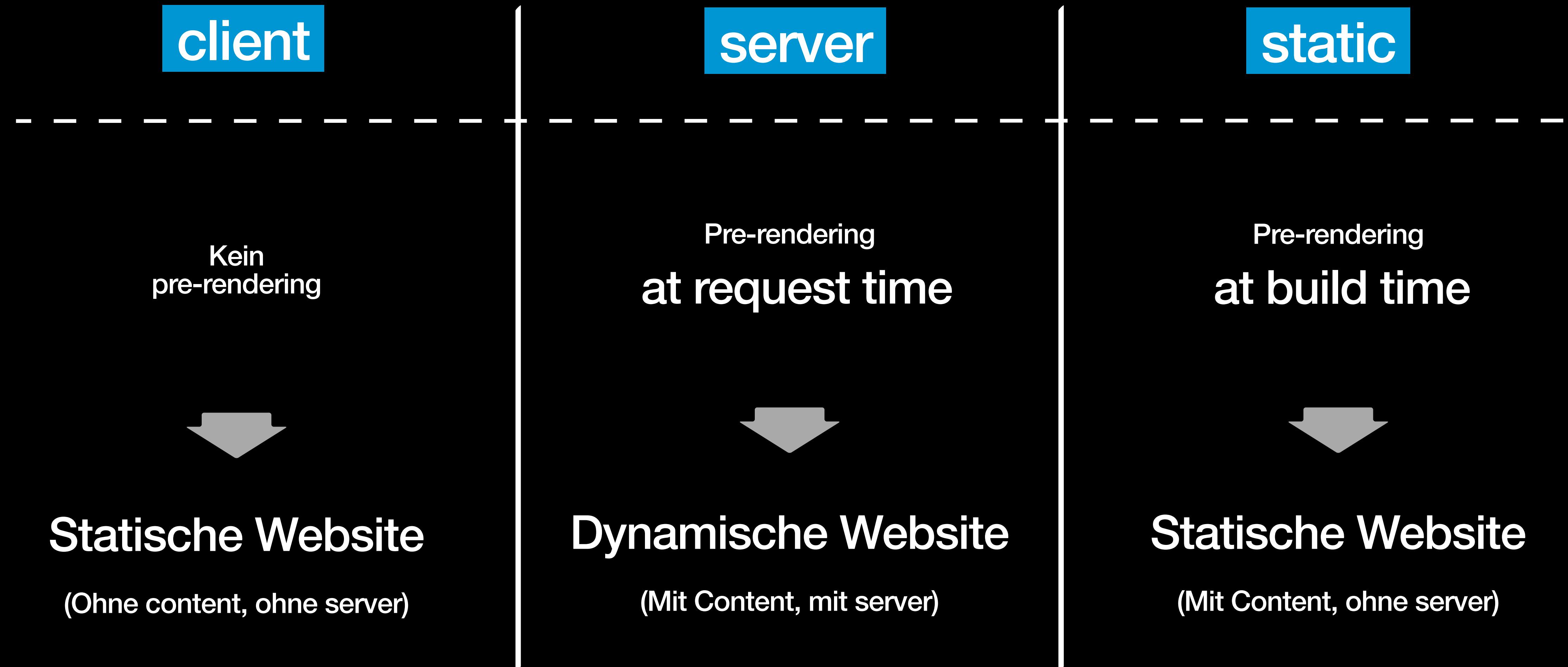
<!--$component/like_button-->
<button>
  Like Me
</button>
<!--/$component/like_button-->
```

Kompiliert

Static Site Generation



Jasprs Rendering Modes



Mehr zu entdecken!

Deployment

State
Management

Custom
Backends

Testing

Flutter
Embedding

Javascript
Integrations

Routing

CLI

CSS Dart
Bindings

SEO

Flutter
plugins



/schultek/jaspr



/schultek_dev

