

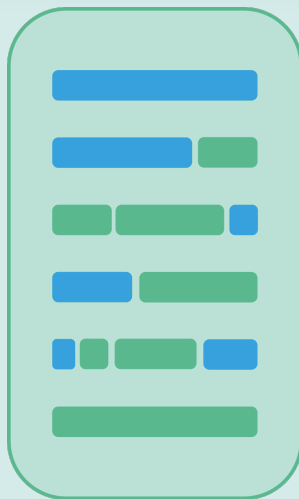
React

wo soll das noch hinführen?

Eine Diskussion

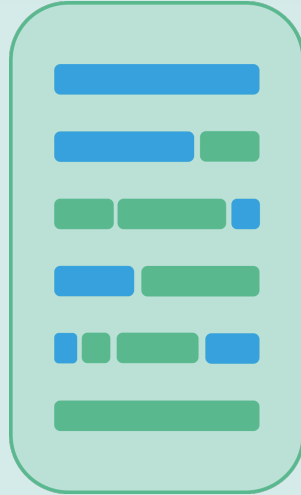
Sebastian Springer | Nils Hartmann | und ihr

Vergangenheit: JavaScript-Schnipsel-Architektur



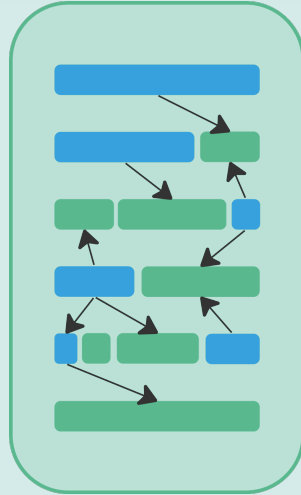
Typische Technik: Backend-Sprache, Template-Sprache, JavaScript, jQuery

Vergangenheit: JavaScript-Schnipsel-Architektur



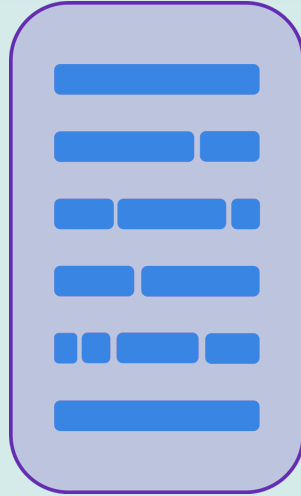
Gut: JavaScript nur für Interaktion

Vergangenheit: Schnipsel-Architektur



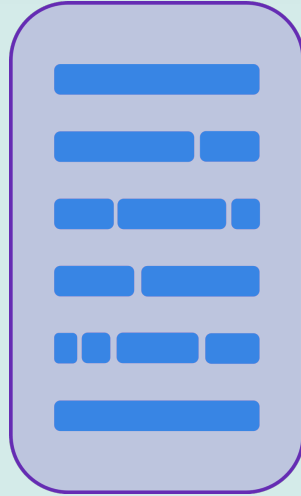
Schlecht: Schnell unübersichtlich, unwartbar, kaputt

Gegenwart: Single-Page-Anwendung



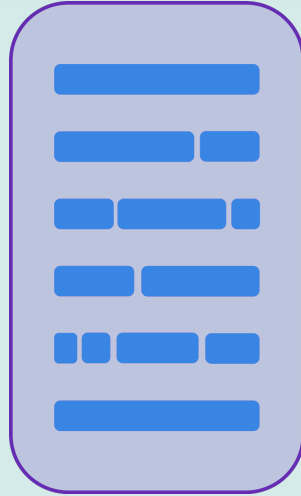
Alles in JavaScript

Gegenwart: Single-Page-Anwendung



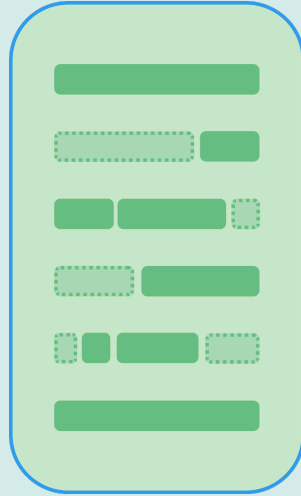
Gut: Klare Verantwortlichkeit (UI ist Client-Sache)

Gegenwart: Single-Page-Anwendung



Schlecht: JavaScript auch für statische UI (Performance!)

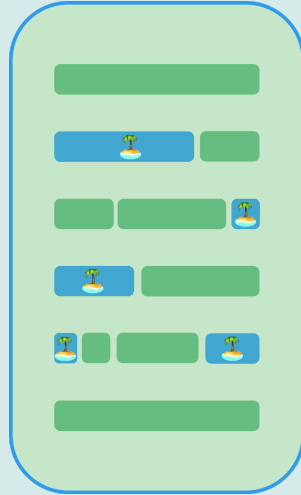
Fullstack-Anwendungen



Ebenfalls ganz in JavaScript

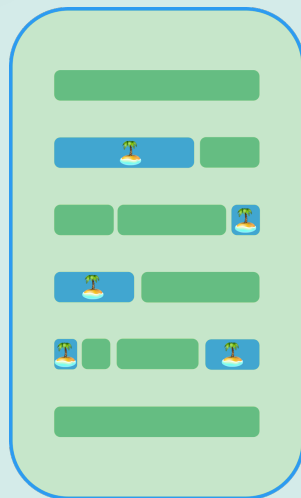
Laufen auf dem Server

Fullstack-Anwendungen



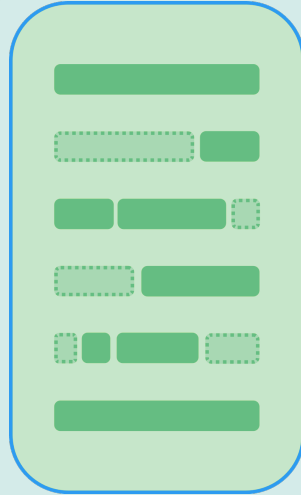
Gut: JavaScript **zur Laufzeit** nur für interaktive Teile im Client ("Islands")

Fullstack-Anwendungen



Schlecht: ?

Fullstack-Anwendungen



Zurück zur Schnipsel-Architektur

Aber: Auftrennung in Server/Client-Teil übernimmt aber das Framework für uns

...aber wo soll das alles hinführen?



Was bedeutet das für die Zukunft von SPAs?

Sind Fullstack-Frameworks überhaupt stabil?

oder doch lieber Vanilla React?

wie seht ihr das?

braucht ihr das?