

Data Flow on Steroids **– Das fullbody** **Workout für deine SPA**

Introductory and overview

Ihr Ansprechpartner

Dominik Deschner

Senior Software Developer

Tim Steiner

Senior Software Developer



deschner@medialesson.de | steiner@medialesson.de



+49 7231-133-258-0



Take aways

Datenfluss von
Webapplikationen
verstehen

Warum State-
Management?

Hands-On-
Erfahrung

Vor- und Nachteile
verschiedener
Ansätze

Agenda

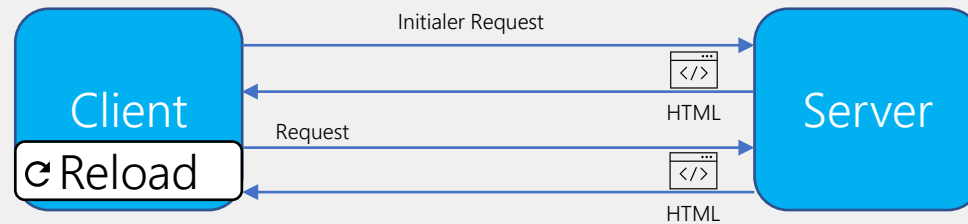
- Was ist State und wenn ja, wie viele?
- Szenario 1: PumpIT Studios Einführung
- Szenario 2: State Facades
- Szenario 3: CQRS mit NGXS

Was ist State?

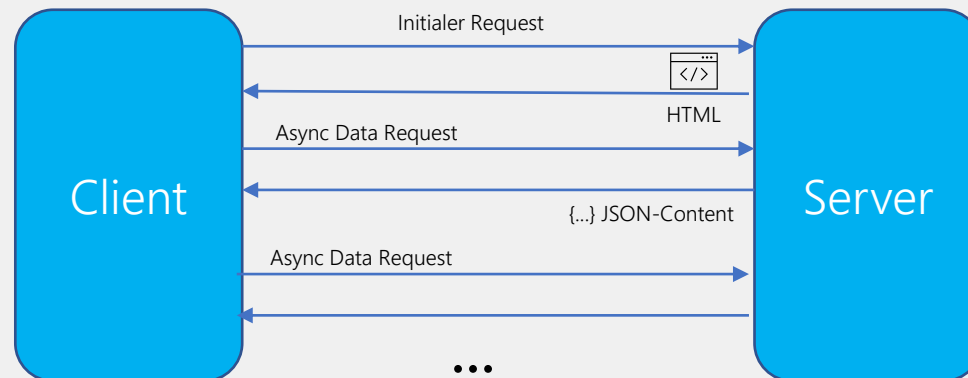
Einführung

Die Single Page Application

- Bei einer MPA wird der Datenfluss in der Regel serverseitig bearbeitet und es wird HTML ausgeliefert.



- Bei einer SPA wird der Datenfluss in der Regel im Browser bearbeitet.





Verkettete Änderungen

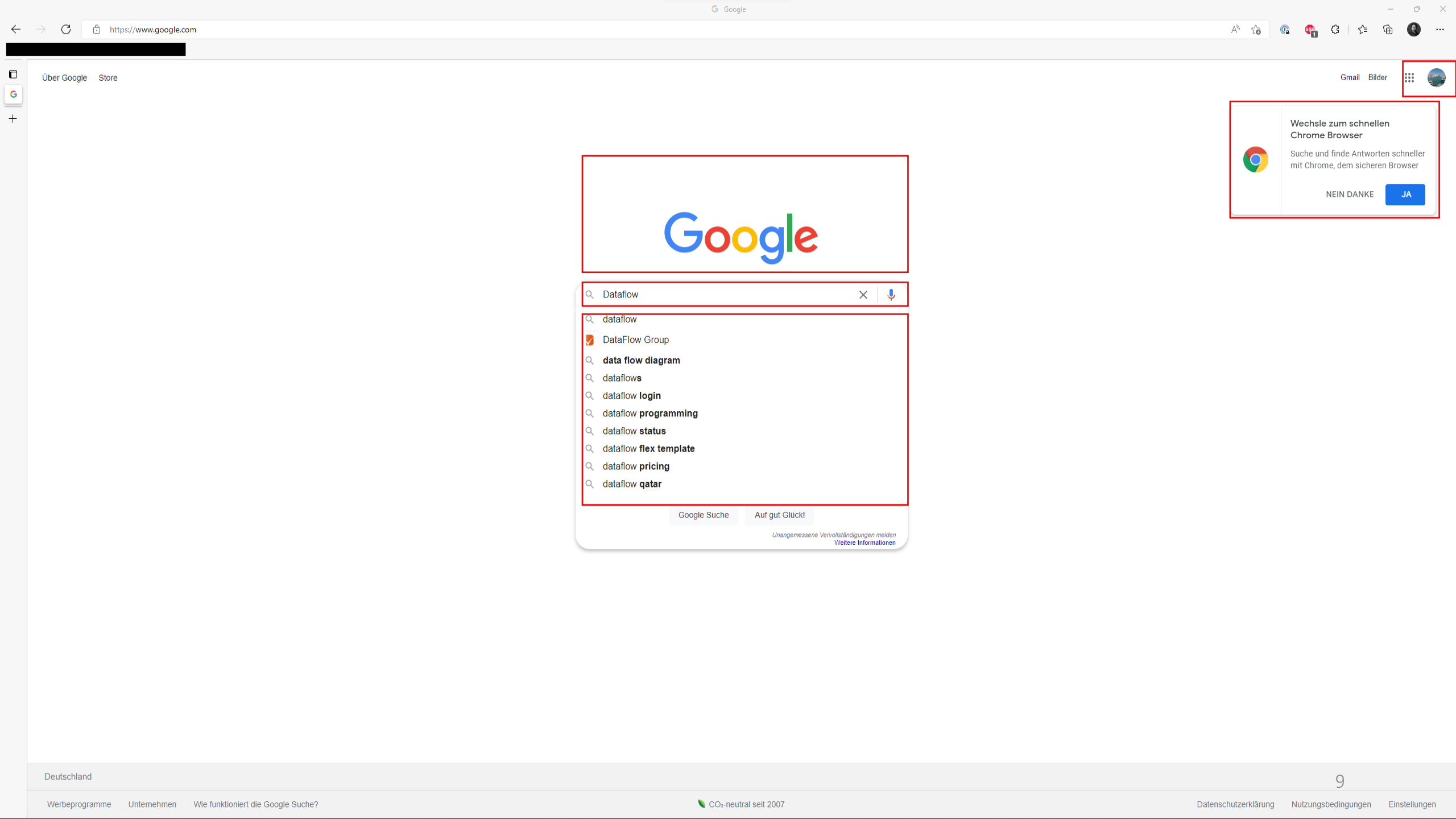
+



Asynchronität

Was versteht man unter Datenfluss und welche Daten gibt es?

- Der Datenfluss einer Webapplikation beschreibt den Fluss sämtlicher Daten zwischen Server und Client oder innerhalb des Clients.
- Welche Daten gibt es?
 - Inhaltliche Daten
 - z.B. Workout-Plan, Anleitungen zum Workout, Kalenderdaten, Übersicht der Studios, etc.
 - Zustandsdaten
 - z.B. Ist der Benutzer eingeloggt? Wie heißt der aktuelle Nutzer? In welchem Studio befindet sich der Nutzer aktuell?
 - > Generelle Zustandsdaten der aktuellen Seite, welche über verschiedene Komponenten oder Module geshared werden müssen
 - Ereignisse
 - Button-Klick, Texteingabe etc.



Google


☒

DataFlow Group

Google Suche

Auf gut Glück!

Unangemessene Vervollständigungen melden
[Weitere Informationen](#)



Wechsle zum schnellen
Chrome Browser

Suche und finde Antworten schneller
mit Chrome, dem sicheren Browser

NEIN DANKE

JA

Was ist State?

- Jede Applikation hat einen State, unabhängig davon ob dieser aktiv verwaltet wird oder nicht.
 - Aktuell aufgerufene URL
 - Benutzername
 - Selektiertes Webpage-Element
 - Tabellen-Filter, Queries
 - Session Historie
 - Etc.

Let's code

HandsOn-Introduction

<https://github.com/tim1993/dataflow-angular-samples>

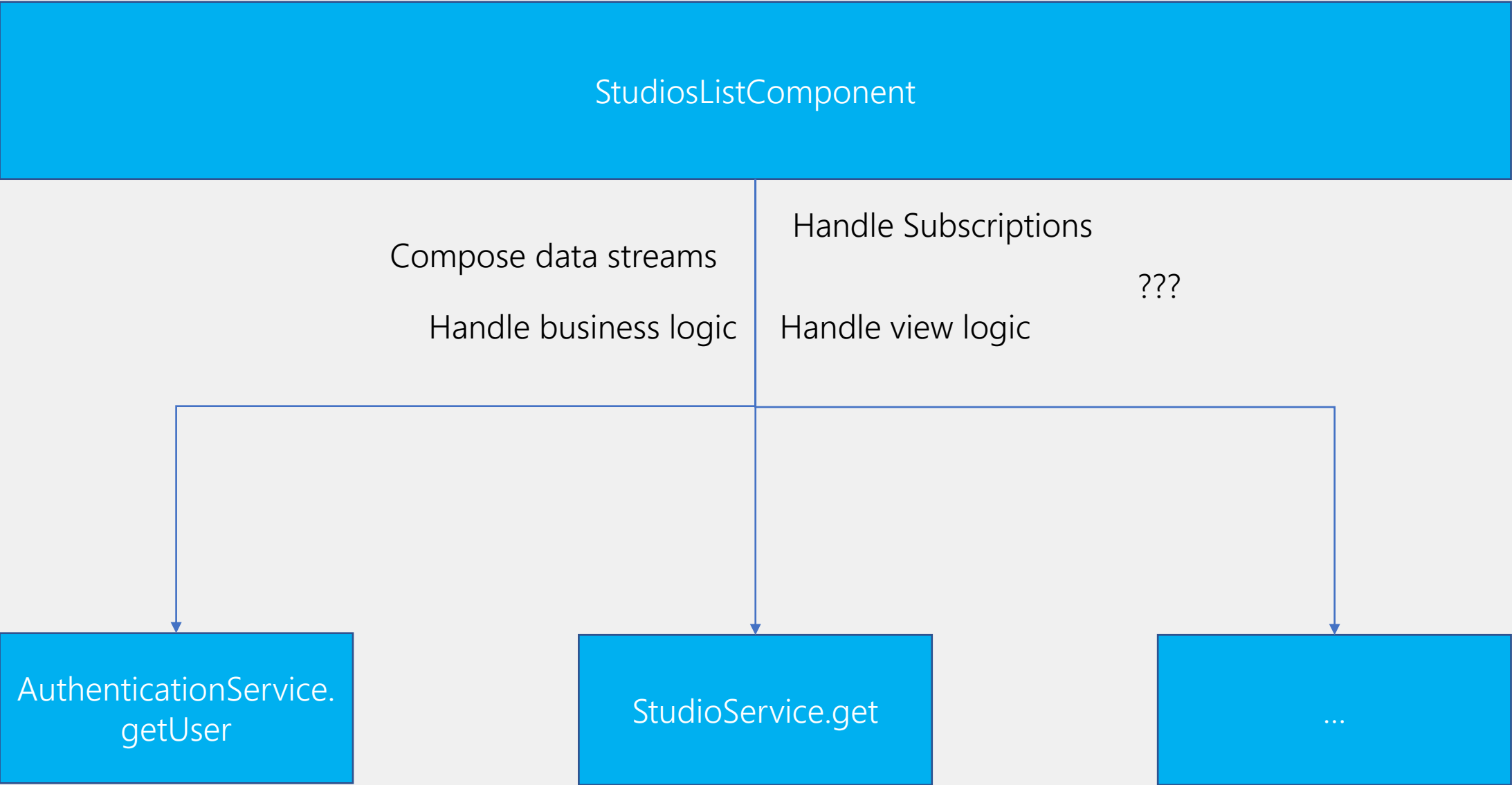
 45 Minuten

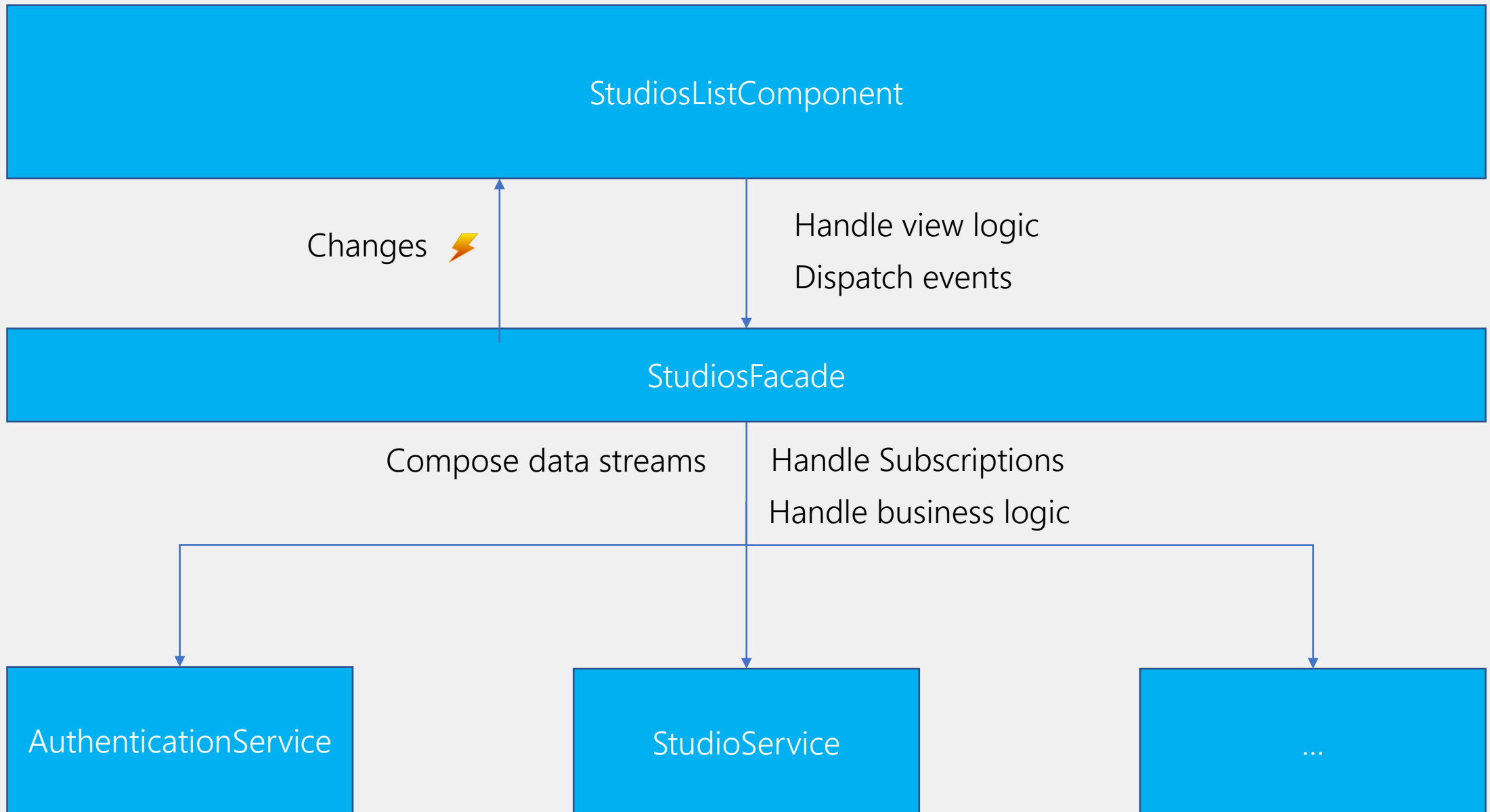
Symptome für chaotisches State Management

- Wenn wir den Zustand nicht verwalten kann man folgende Dinge beobachten:
 - Es werden immer wieder die selben Anfragen an das Backend geschickt, um die Daten überall abfragen zu können.
 - Es werden Services oder Input bzw. Output Bindings in Kombination mit Events verwendet um die Daten zu teilen.
- Mit zunehmender Anwendungsgröße wird das Chaos immer größer.

A photograph of a modern building facade constructed from numerous vertical wooden slats of varying heights and widths, creating a screen-like effect. The slats are made of light-colored wood and are arranged in a staggered pattern. A bright blue horizontal banner is superimposed over the middle of the image, containing the text 'Facades to the rescue' in white, bold, sans-serif font. Below the banner, the building's entrance is visible, featuring dark grey doors. The foreground is filled with tall, green grasses. To the left, a portion of a grey-sided house is visible, and to the right, a red brick house with white window frames is partially seen. The sky is clear and blue.

Facades to the rescue





Welche Probleme lösen Facades?

- Extrahiert einen Teil der Business Logik, Datenkomposition aus Komponente
- Reduziert Verantwortlichkeit der Komponente
- Verdichtet Business Logik in Facades

Let's code

HandsOn-Facades

 60 Minuten

Welche Herausforderungen begegnen uns?

- Gewisser Anteil an Boilerplate-Code
- Zugriff auf Zustand aus anderen Facades „ungewöhnlich“
- Aufteilung/Schnitt von Facades

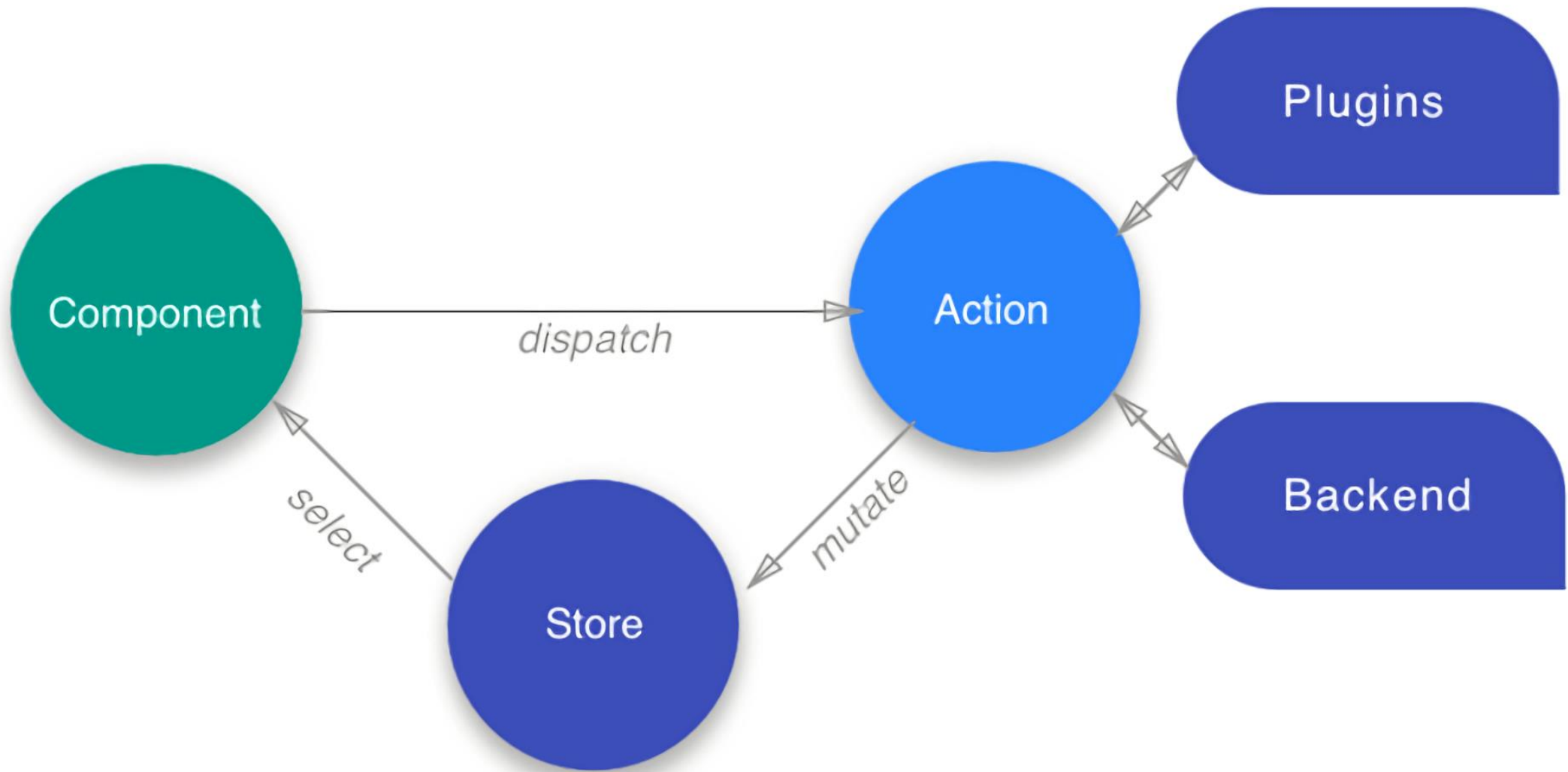
CQRS mit NGXS

Einführung

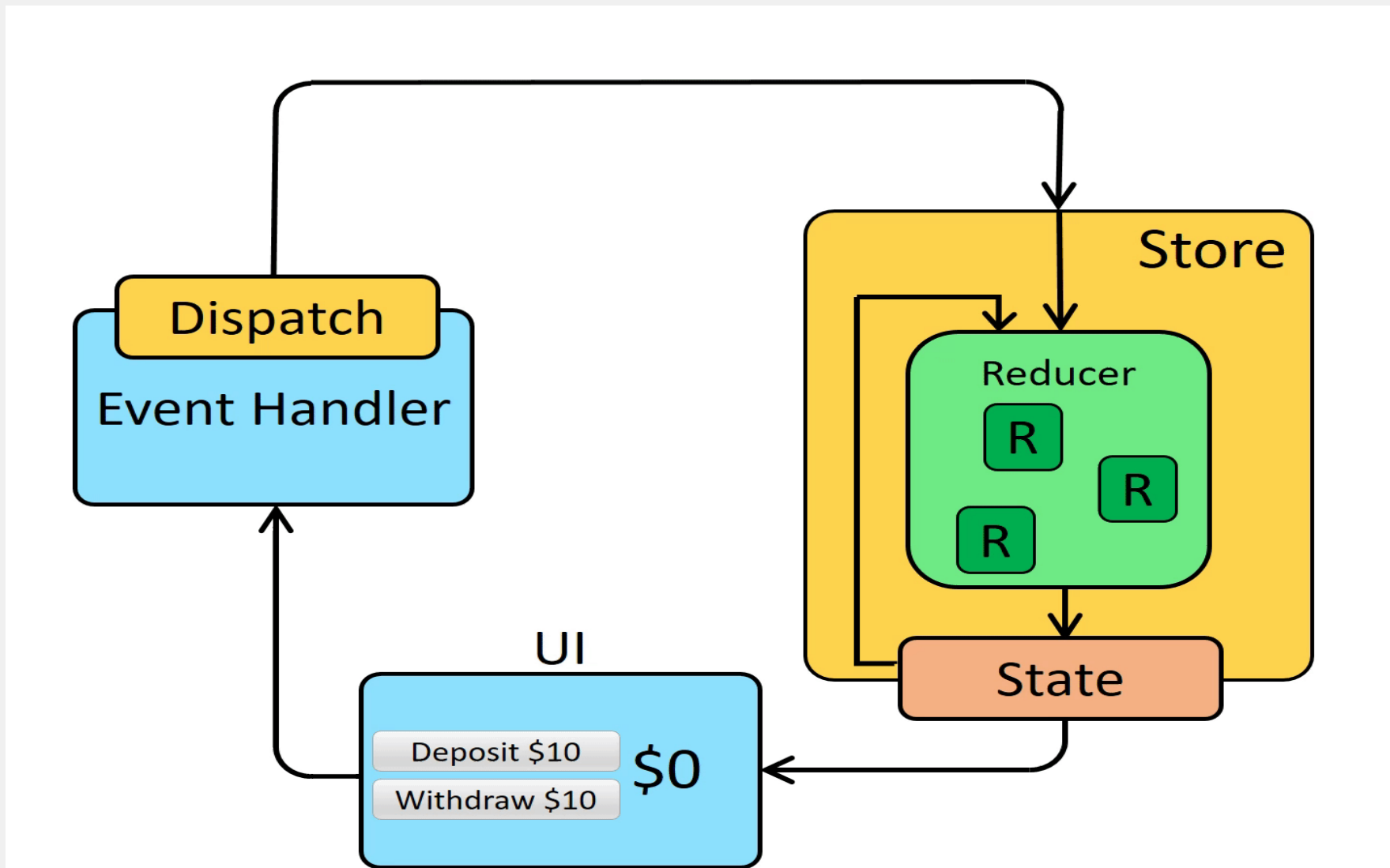
Was ist NGXS?

- State Management Library für Angular
 - CQRS mit Typescript Features und ohne Boilerplate
- Angular style
- Erleichtert die Interaktion mit Zustandsinstanzen
 - Immutability
 - Patching

Das Konzept



Das Redux Pattern



Let's code

HandsOn-NGXS



60 Minuten

Herausforderungen

- Die Einstiegshürde ist ggf. größer, da jeder Entwickler Knowhow über State-Management benötigt
- Was soll bzw. muss in den State und was nicht?
- Wie verändere ich den State und wo bekomme ich meine Daten her?
- Wie sollte der State strukturell aufgebaut sein?

Fazit

- State-Management lohnt sich
 - Sauberere Komponenten-Architektur
 - Mehr Flexibilität
- Beide Ansätze sind einsteigerfreundlich und skalieren mit den Anforderungen
- Unsere Empfehlung: Fangt mit Facades an...

Links

<https://github.com/tim1993/dataflow-angular-samples>