

Exposé zur Abschlussarbeit am LG Kooperative Systeme der FernUniversität in Hagen

Informatik Bachelor of Science

Ricardo Stolzlechner

9463470

# YReduxSocket

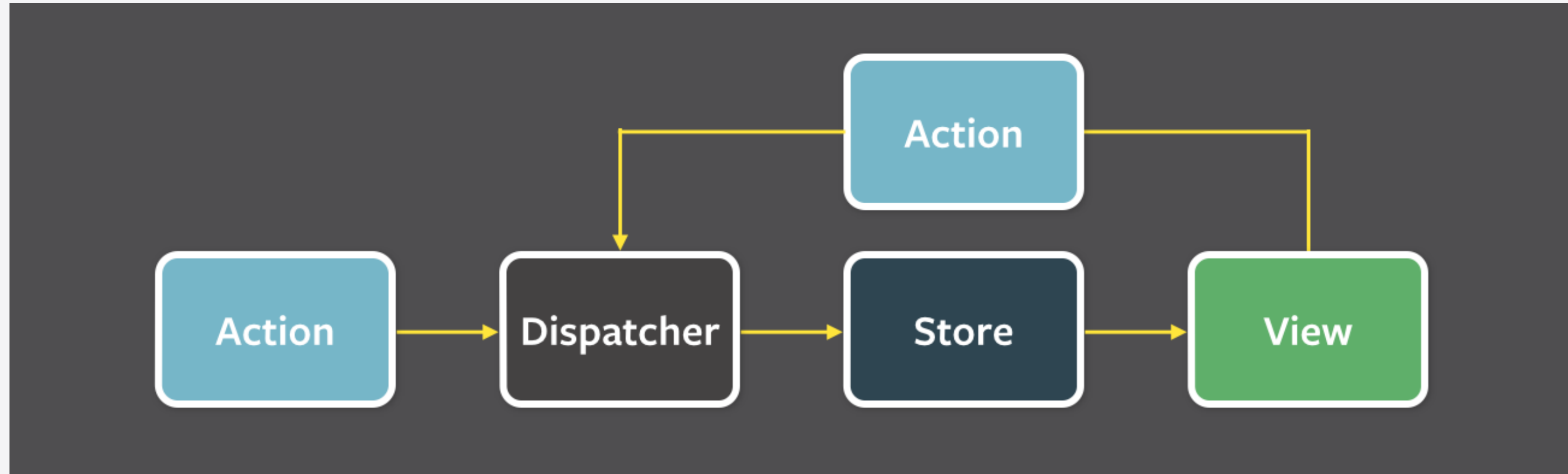
Optimierung der Entwicklung von Webanwendungen mit  
zentralem Datenstore und WebSocket Kommunikation

# Problemstellung

- Webanwendungen mit Redux Store und WebSocket Kommunikation
- Aufwand für neuer Funktionalität bzgl. Datenanpassung minimieren
- YReduxSocket
- Aufgabenstellung
  - Generische Konzeption und Spezifikation
  - Recherche und Vergleich von Alternativen
  - Implementierung und Tests

# Flux pattern [Far17, S. 32]

- CRUD Operationen immer nur in eine Richtung
- State nicht direkt manipulieren (senden einer Action)
- Nach Aktualisierung sollte View neu rendern



[Abb. 2] Quelle: <https://github.com/facebookarchive/flux/blob/main/examples/flux-concepts/flux-simple-f8-diagram-with-client-action-1300w.png> (abgerufen am: 02.10.23)

# Redux basierter Datenstore [Far17, S. 35-38]

## → Store

- Single source of truth
- subscribe (Benachrichtigung bei Updates)
- dispatch (Aufforderung zur State Aktualisierung)

## → Reducer

- pure function
- zwei Argumente: vorherigen State und Action
- Returnwert: neue State

## → Action

- Datenstruktur: type und payload

# Redux und WebSocket: herkömmlicher Ansatz

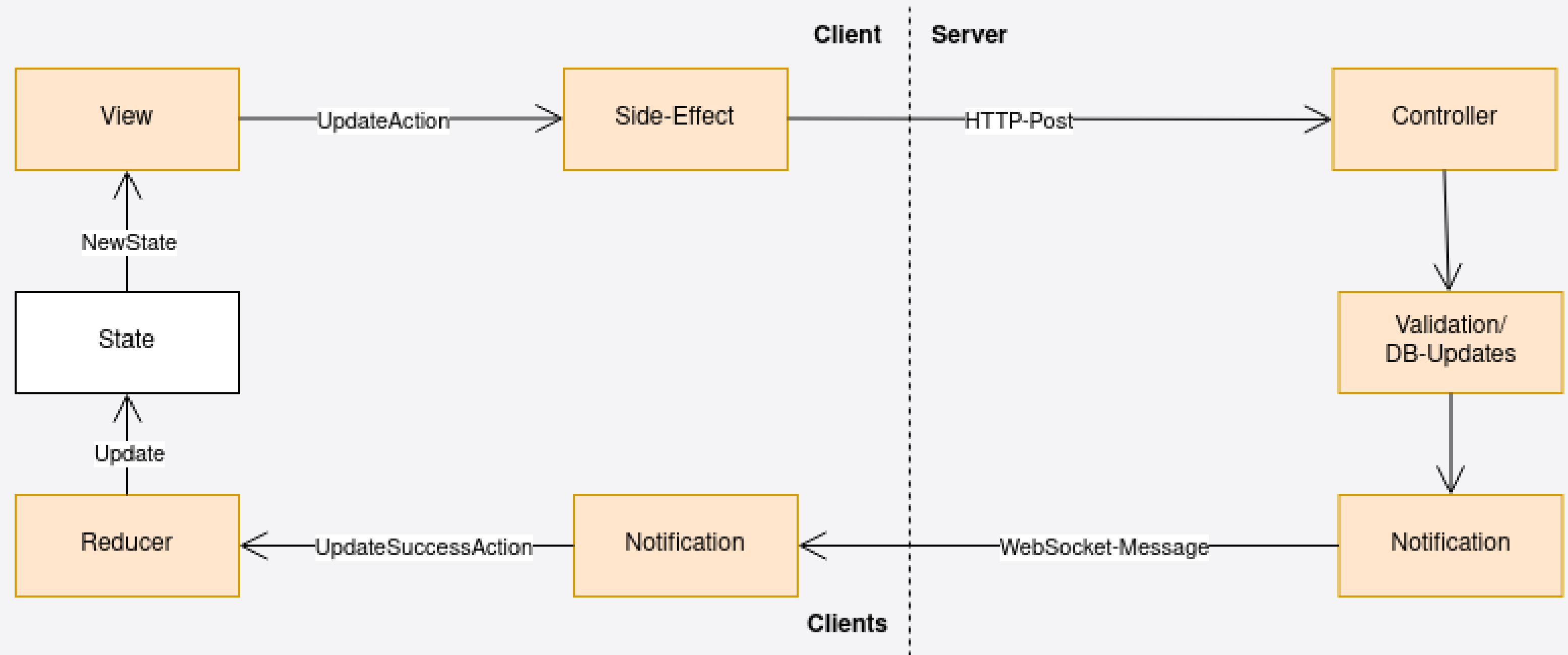


Abb 3. Redux und WebSocket: herkömmlicher Ansatz

# YReduxSocket

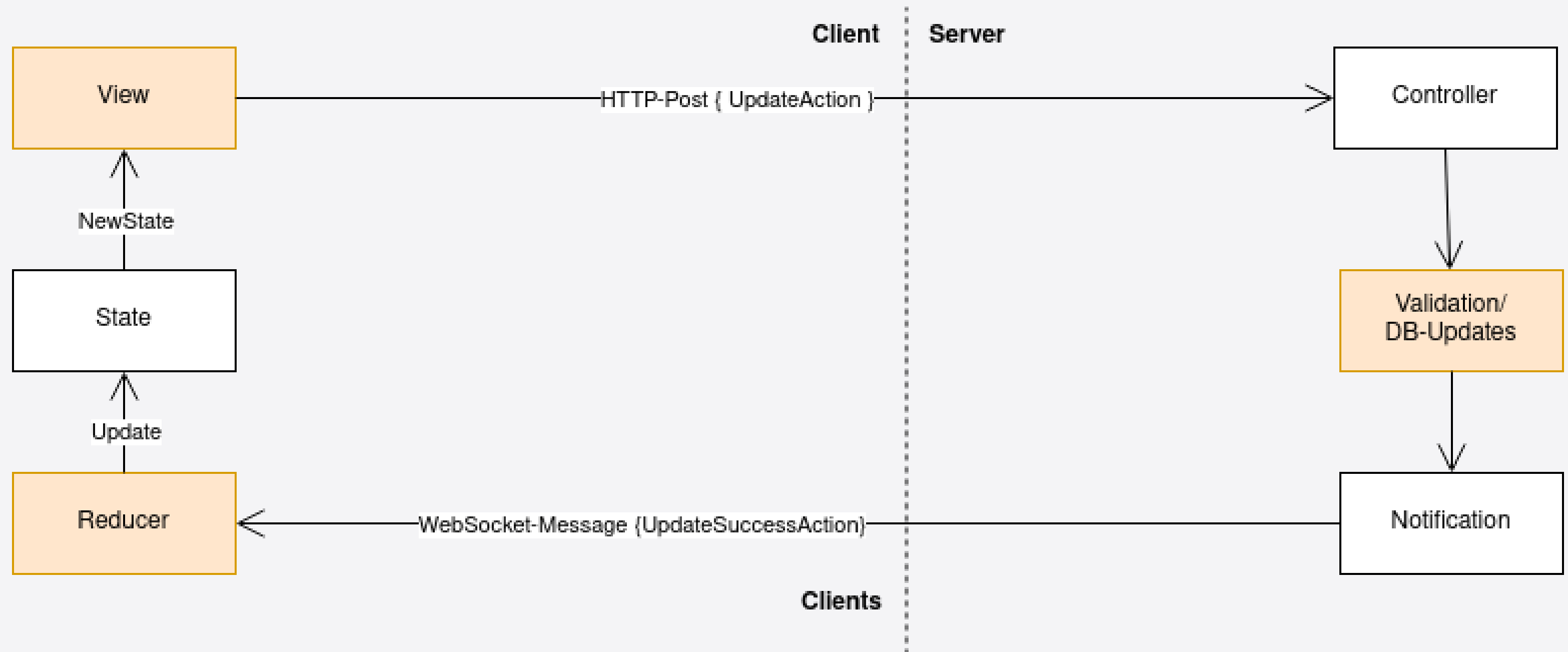


Abb 4. YReduxSocket



# Konsistenzverhalten

- Redux Store: alle Komponenten haben konsistente Daten (am selben Client)
- Wirkliche Wahrheit liegt aber an Server Datenbank
  - R-W und W-W Konflikte: eventual consistency [TS08, S. 319 ff.]
- Stärkere Konsistenz notwendig: Algorithmus von Marijn Haverbeke [Hav15]
  - Actions mit Versionsnummern
  - Fall A: Versionsnummern stimmen überein
  - Fall B: Versionsnummern stimmen nicht überein

# Offene Fragen

→ Haben Sie noch Fragen?

→ Meine Fragen

→ Vorgaben bzgl. Latex (book, Zeilenabstand, Titelblatt, ...)

→ Vorgaben bzgl. Präsentation (Layout, Zitationen)

→ Anmeldung beim Prüfungsamt

→ Termine (Versionen der Abreit, Kolloquium)

→ Englisch

→ Zitationen Blog-Posts, Framework Dokumentationen



# Quellenangaben

→ [Abb. 1] Quelle: <https://www.pexels.com/photo/blur-business-close-up-code-270557/> (abgerufen am: 30.09.23)

→ [Far 17] Oren Farhi. „Adding State Management with ngrx/store“. en. In: Reactive Programming with Angular and ngrx: Learn to Harness the Power of Reactive Programming with RxJS and ngrx Extensions. Hrsg. von Oren Farhi. Berkeley, CA: Apress, 2017, S. 31–49. isbn: 978-1-4842-2620-9. doi: 10.1007/978-1-4842-2620-9\_3. url: [https://doi.org/10.1007/978-1-4842-2620-9\\_3](https://doi.org/10.1007/978-1-4842-2620-9_3) (besucht am 20. 09. 2023)

→ [Abb. 2] Quelle: <https://github.com/facebookarchive/flux/blob/main/examples/flux-concepts/flux-simple-f8-diagram-with-client-action-1300w.png> (abgerufen am: 02.10.23)

→ [TS08] Andrew S. Tanenbaum und Maarten van Steen. Verteilte Systeme: Prinzipien und Paradigmen. ger. 2., aktualisierte Auflage. it-informatik. München Harlow Amsterdam Madrid Boston San Francisco Don Mills Mexico City Sydney: Pearson, 2008. isbn: 978-3-8273-7293-2.

→ [Hav15] Marijn Haverbeke. Collaborative Editing in ProseMirror. en. Okt. 2015. url: <https://marijnhaverbeke.nl/blog/collaborative-editing.html> (besucht am 04. 10. 2023).

**Vielen Dank für Ihre  
Aufmerksamkeit**