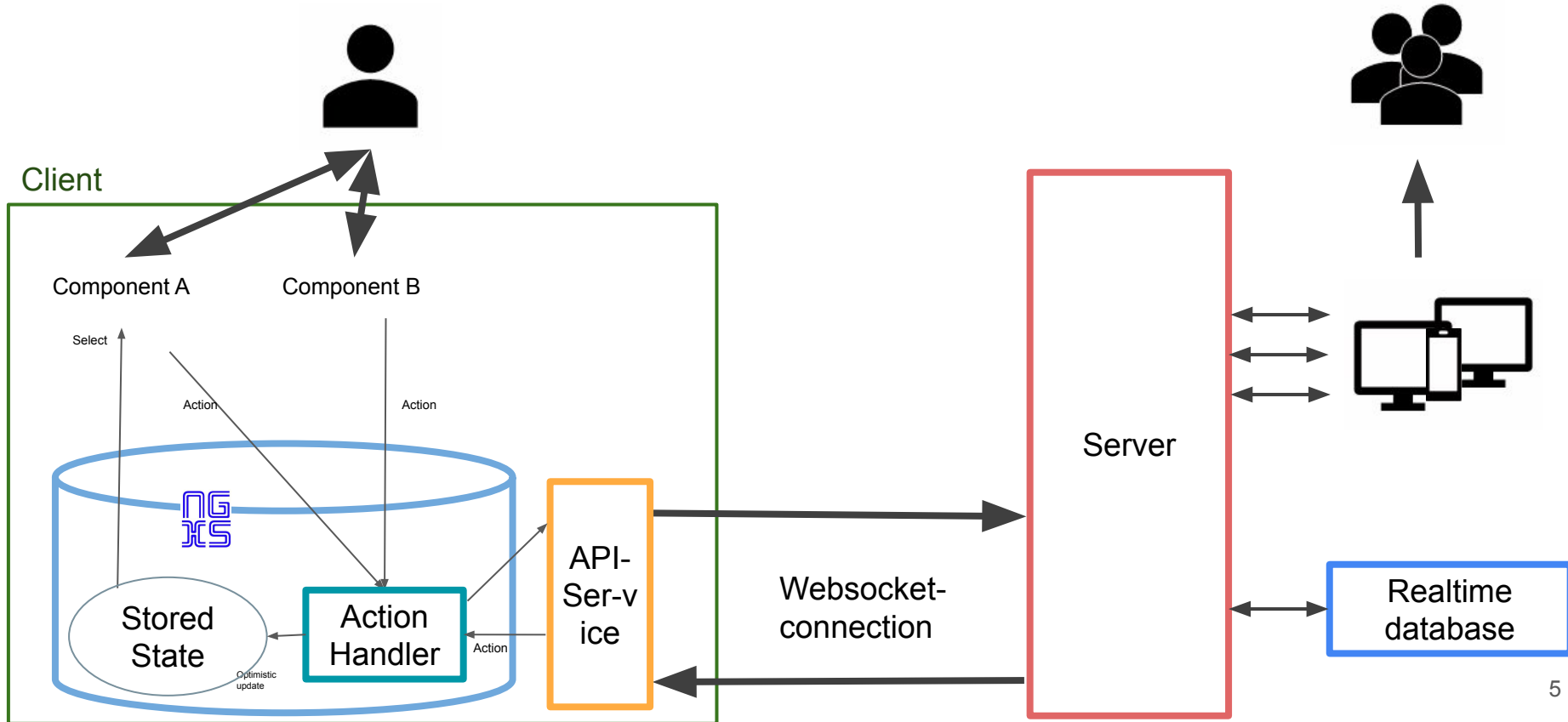


Low latency state management across multiple devices



Optimistic update algorithm

1. Action **A** mit optimistic update wird dispatched
 - a. Request wird an backend gesendet
 - b. State wird in **S** zwischengespeichert (via reference => kein doppelter Speicherplatz)
 - c. State wird synchron geupdated (via patch => keine Änderung in **S**)
 - d. Alle Actions werden solange deferred, bis eine positive Antwort der Action **A** zurückkam
 - e. Wenn negative Antwort von Action **A** zurückkommt, wird
 - i. eine Fehlermeldung angezeigt
 - f. die State Änderung von **A** rückgängig gemacht ([der State wird auf **S** gepatcht](#))
 - g. die deffernten Actions ausgeführt (priority queue -> backend bevor frontend)
- - SELECTs geben nicht immer den korrekten State zurück
- - eine Action mit optimistic update blockiert alle anderen Actions
- + Reihenfolge der Actions wird garantiert
- + State, auf dem eine Action ausgeführt wird, wird garantiert
- -> man muss bei der Kommunikation mit dem Server (zumindestens für actions) auf einer [request-response](#) Kommunikation aufbauen, in der response ist die ursprüngliche Action mit enthalten
- -> es ist bspw. möglich, einen Zähler synchronisiert auf allen clients zu verändern
- -> actionHandler werden zwischen client und server geshared