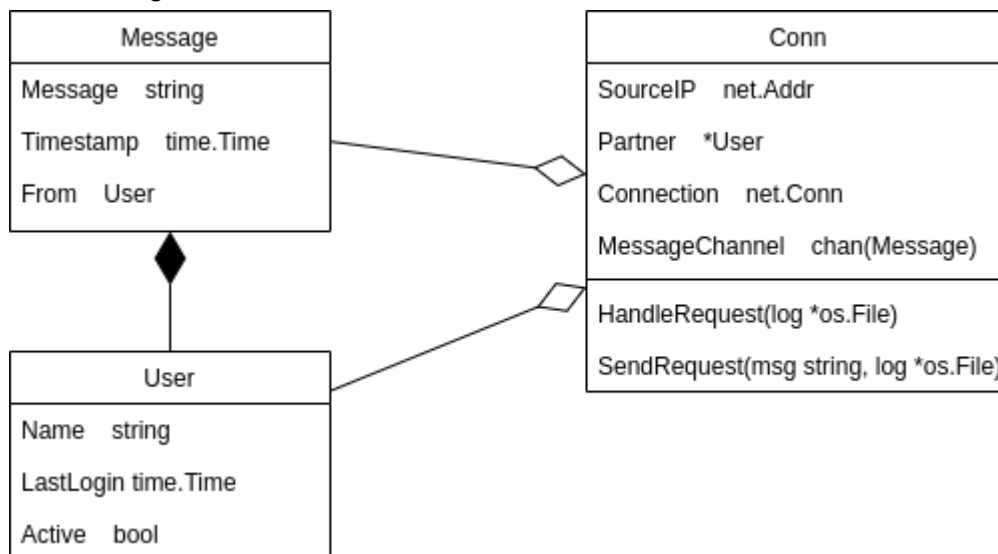


SchardtChat

Architektur

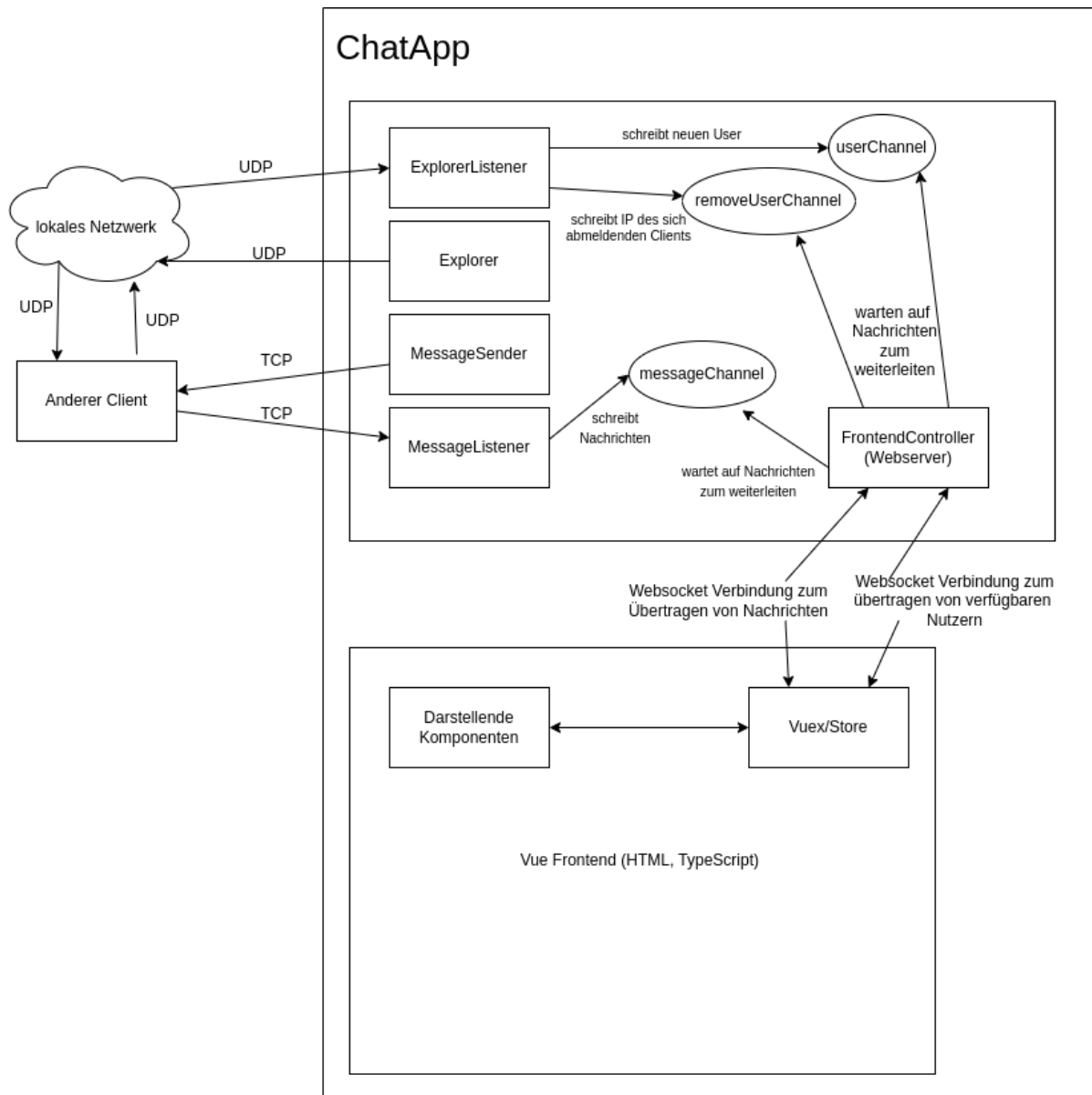
Das Backend ist in Golang verfasst und liefert ein JavaScript-Frontend aus. Diese kommunizieren über Websockets in beide Richtungen. Es wurde versucht, Multithreading in eine Art "Microservice-Architektur" zu verwandeln, zwischen den einzelnen Prozessen herrscht eine lose Kopplung über eine Thread-Sichere Datenstruktur, die den Zugriff auf die Ressource eigenständig reguliert. Die Anwendung besteht aus drei Hauptprozessen, zwei von ihnen geben ihre gesammelten Daten über zwei Websockets direkt an das Frontend weiter, um sie darzustellen. Innerhalb der empfangenden Prozesse wird je Nachricht ein Thread gestartet, um diese zu verarbeiten, damit der Hauptprozess nicht blockiert wird. Es laufen also mindestens zehn Threads gleichzeitig (ExplorerListener, Explorer, MessageSender, MessageListener, und innerhalb des FrontendController: je 2 Threads pro Channel. Übersichtlicher auf der nächsten Seite dargestellt.

Klassendiagramm:



verwendete Libraries:

- golang standard libraries (verschiedene)
- github.com/gorilla/websocket
- github.com/kelseyhightower/envconfig
- github.com/seanfoley/ipaddress-go
- VueJS
- TypeScript



Sequenzdiagramm auf der nächsten Seite.

Sequenzdiagramm:

