Fachhochschule Nordwestschweiz

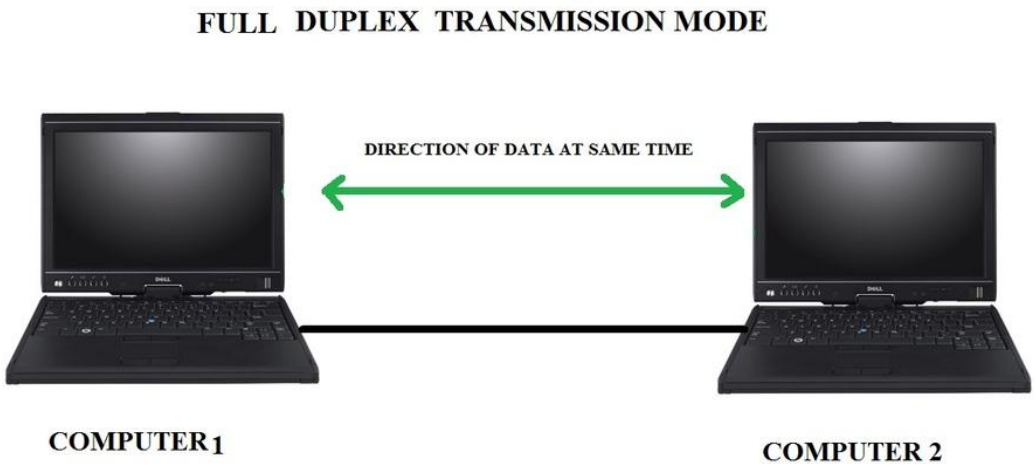
Websocket-Protokoll

Inhaltsverzeichnis

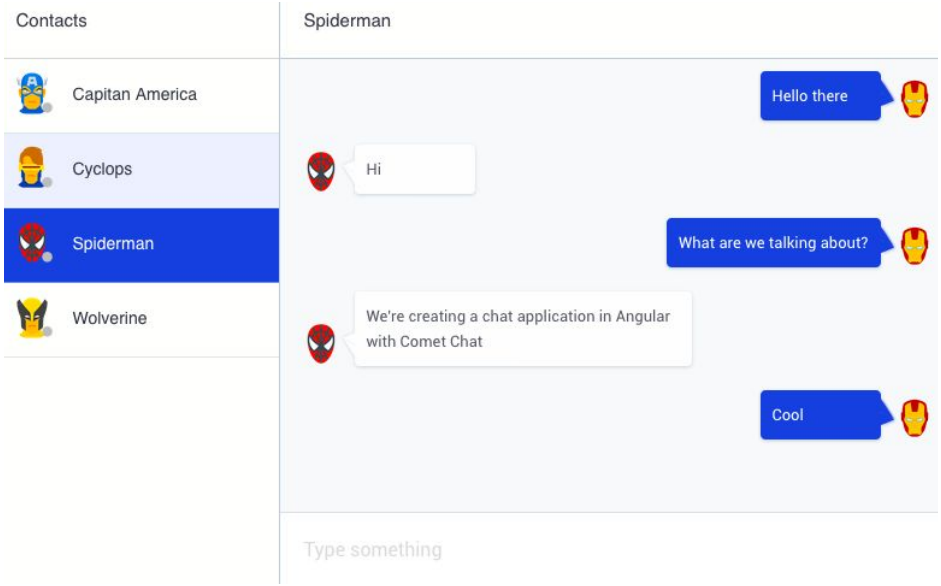
- Einführung Websocket
- Asynchrone Programmierung
- Code Snippets
- Demo Chat

Was ist Websocket?

- Netzwerkprotokoll mit Full-Duplex Kommunikation
- Auf TCP basierende Kommunikation
- Client-seitige Initialisierung



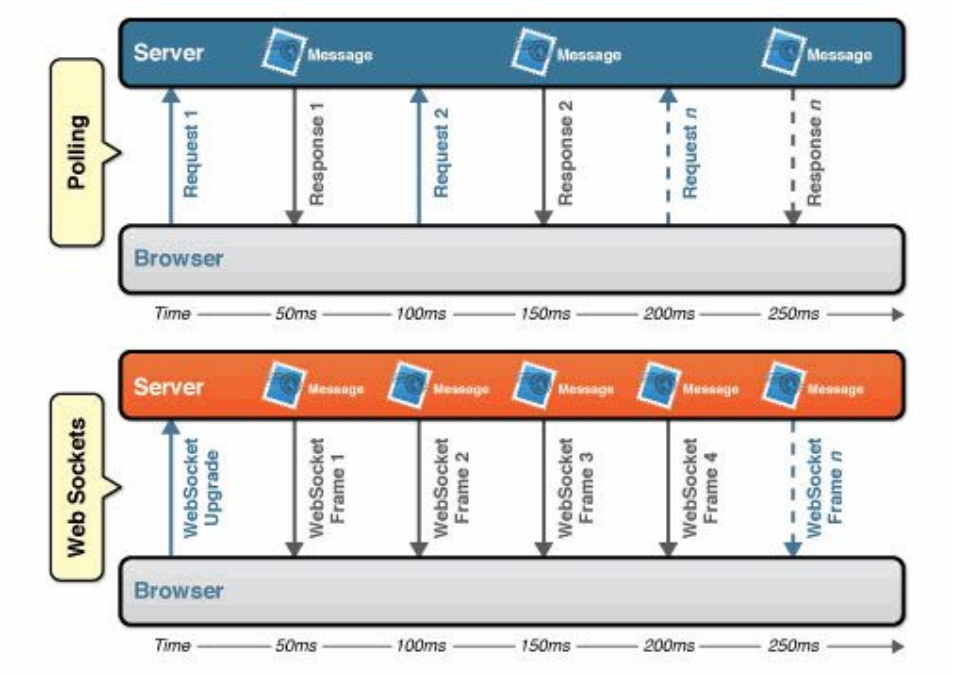
Wann werden Websockets verwendet?



- Schnelle Verbindungsaufbau
  - News-Ticker, Chat-Applikationen etc.
- Echtzeit Darstellung von Informationen

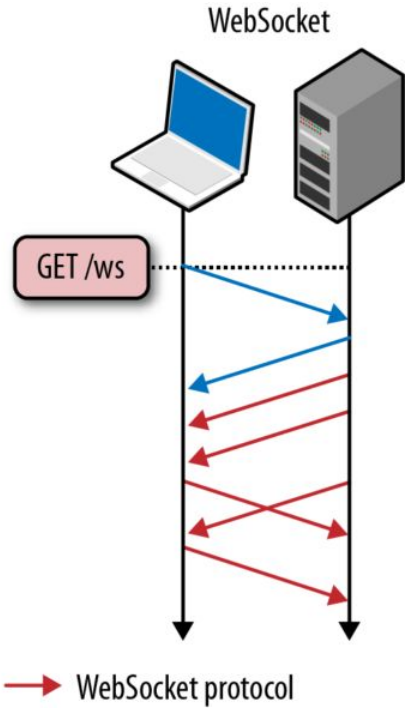
Vorteile

- Schnellere Ladezeiten
- Nur eine einzige TCP Verbindung nötig
- Ersetzt long-polling
- Minimale Overhead bei verschickten Nachrichten



"Websocket Protokoll" FHNW FS20

Nachteile

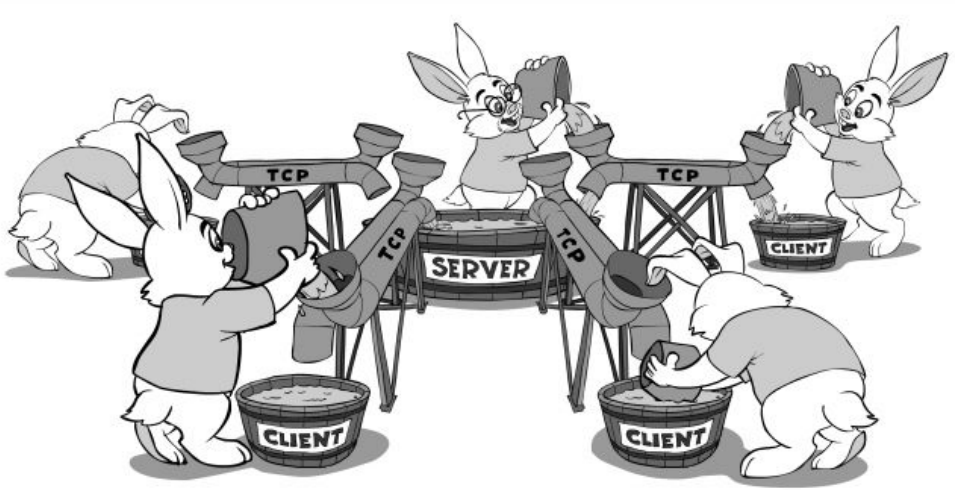


- Server-seitige Events erkennen gelöschte Clients nicht, bis eine Nachricht versendet wird
- Schwerer eine zuverlässige Kommunikation zu erreichen (v.a. für Mobile Users)

Welches Protokoll wird verwendet?

- TCP/UDP als Low-Level Protokoll
- Internet Socket (TCP/IP, UDP/IP)

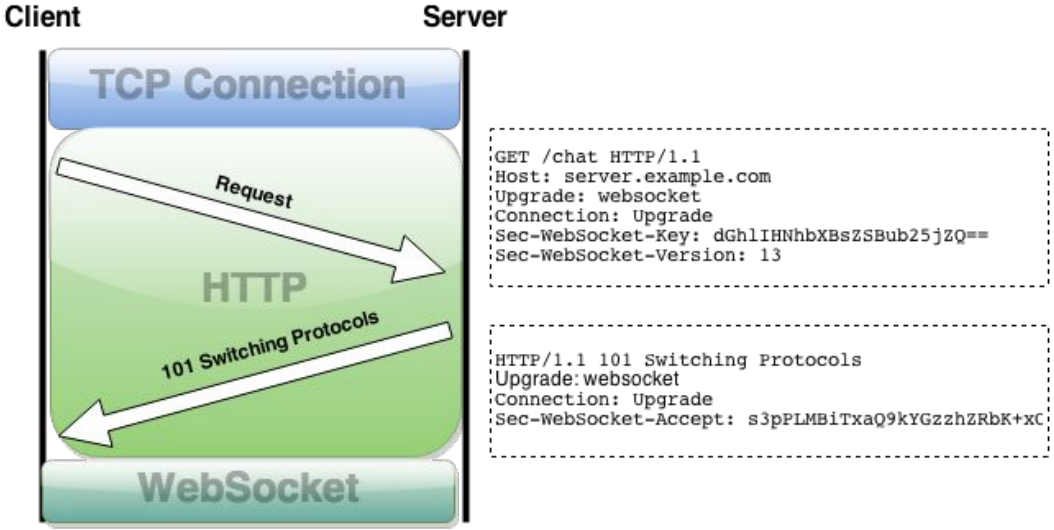
Websocket als Internetprotokoll



- Full-Duplex Kommunikation zwischen Client und Server
- Gleichzeitiges Senden von Daten möglich
  → bidirektionale Kommunikation
- Gleichwertige Kommunikationspartner

Websocket Verbindungsaufbau

- Upgrade des HTTP-Protokolls, benötigt TCP als Low-Level-Protokoll.
- Client startet immer den Upgrade Request (Opening Handshake).



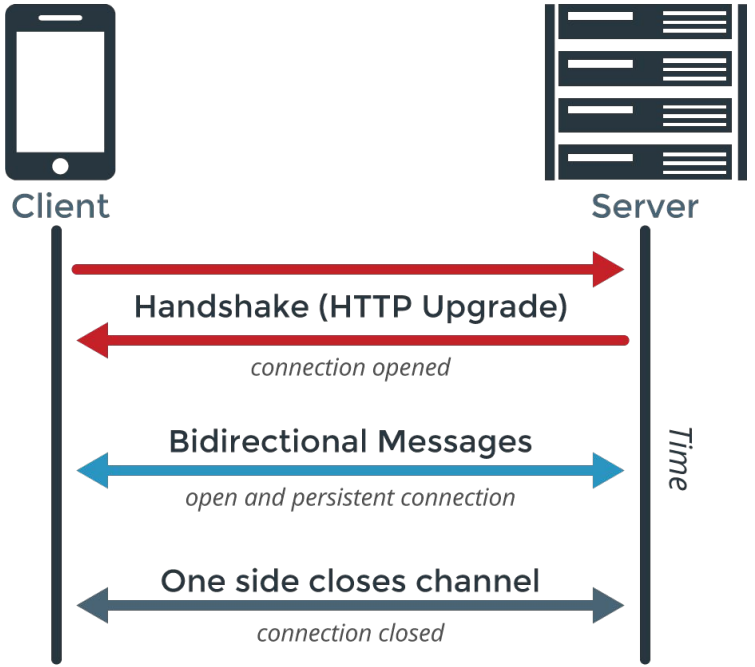
URI-Schema

- Zwei URI-Schema definiert
  - ws: unverschlüsselte Verbindung
  - wss: verschlüsselte Verbindung
- WSS-Protokolle laufen auf SSL/TLS über eine TCP-Verbindung
- WS-Port per default 80
- WSS-Port per default 443

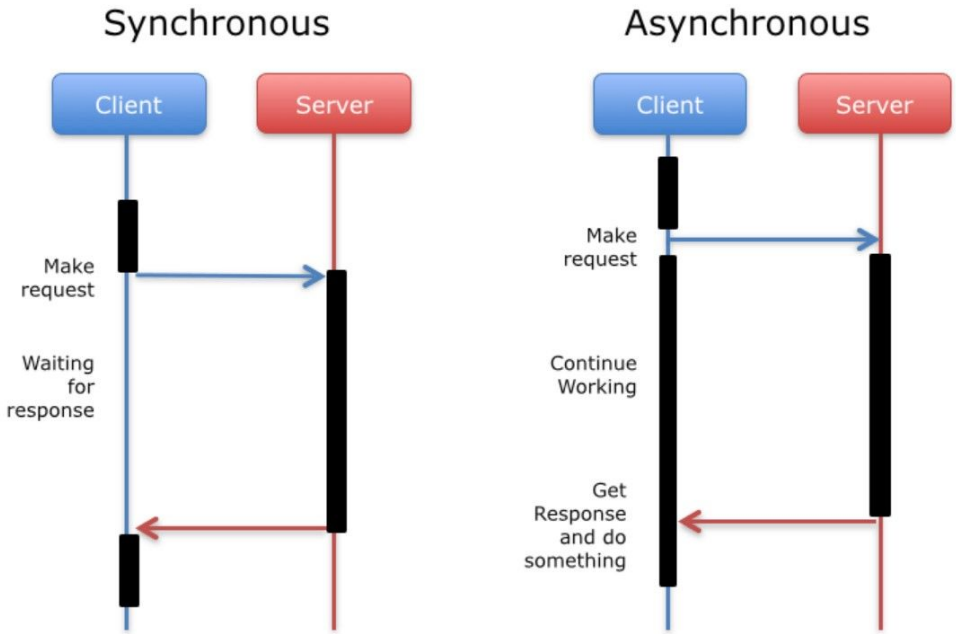
ws://localhost:8080/example

Websocket Kommunikation

- Client startet Kommunikation mit einem “Opening Handshake” (Update-Request)
- Server bestätigt mit Response
- Datenübertragung mittels CRUD-Operationen
- TCP-Verbindung bleibt nach Datenübertragung bestehen
- Verbindung mit Close-Frame schließen (TCP-Verbindung trennen)

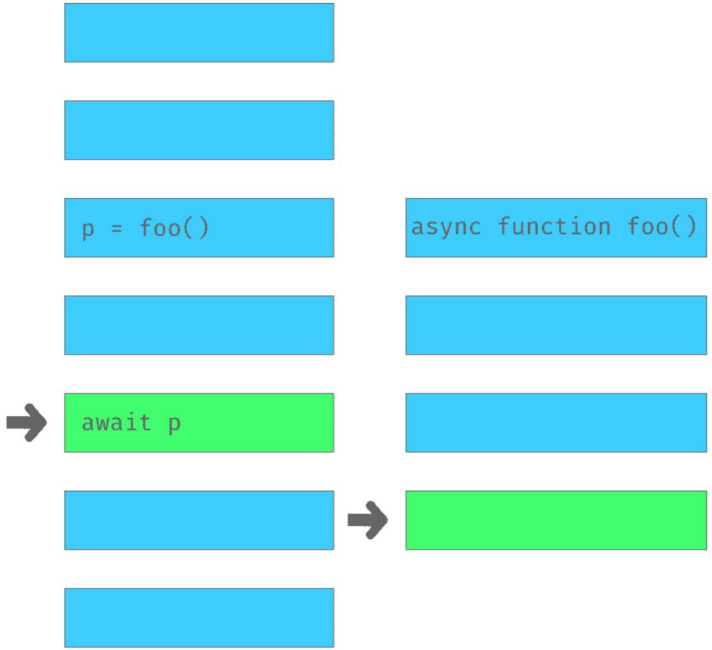


Asynchrone Programmierung



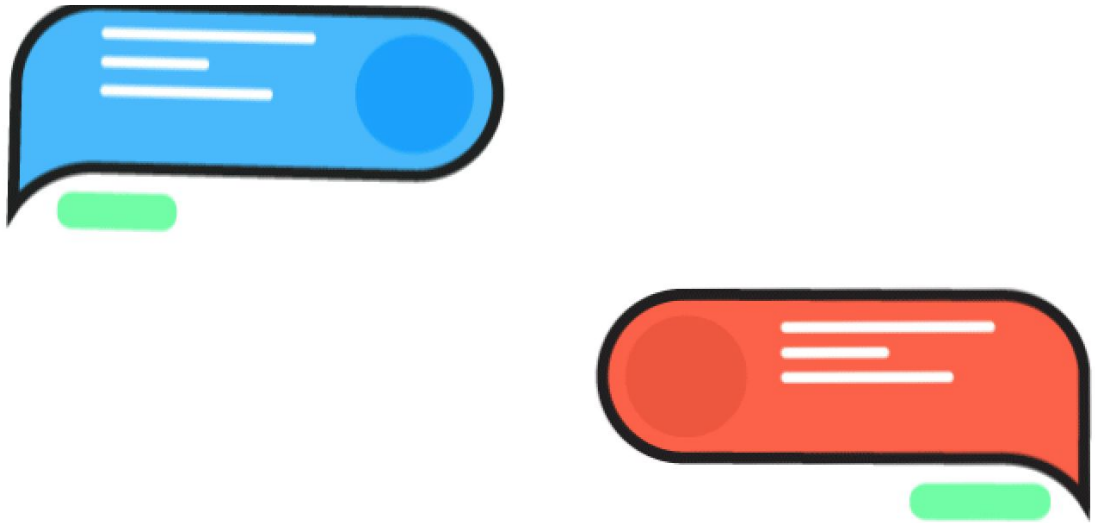
- Erlaubt dem Code andere Aufgaben zu erledigen, während man auf die Antwort einer Ressource abwartet
- In Python nennen wir asynchrone Funktionen «Coroutine»
- Brauchen Keywörter async und await

Async in Websockets



- Asynchrone Websocket-Verbindung
- Client und Server senden Daten zur jeder Zeit (gleichzeitig möglich) ohne Einschränkung

Demo Chat-Applikation



Referenzen

- https://sookocheff.com/post/networking/how-do-websockets-work/
- https://de.wikipedia.org/wiki/WebSocket
- https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/
- https://medium.com/velotio-perspectives/an-introduction-to-asynchronous-programming-in-python-af0189a88bbb