

# Real-Time Communication und Notification APIs

Maria Husmann

[maria.husmann@gmail.com](mailto:maria.husmann@gmail.com)

WebZuerich

maria

All Threads

CHANNELS (14)

# announcements

# introduce-yourself

# jobs

# random

# therapy

# web-platform

DIRECT MESSAGES

slackbot

maria (you)

#announcements

☆ | 👤 290 | 📌 2 | Event- and other important group announcements

📞 ⚙️ 📅 🔍 Search @ ☆ ⋮ 🎁



Yesterday

👤 1



**hirsig** 10:39 PM ☆

A team including @hannes and me are bringing Powercoders to Zurich. We are still looking for a:

LEAD TEACHER for our Zurich Team

We are looking for a teacher who:

- \* has lot of experience teaching beginners frontend skills
- \* is well networked in the IT scene in Zurich
- \* is easy going and loves to work in a team
- \* is open towards refugees and their stories
- \* is available in fall (July/August 20%, September - December 60%)

And likes to do the following:

- \* improve the curriculum
- \* write a test to recruit participants
- \* conduct the test on-site

👤 🗨️ ➡️ ⋮

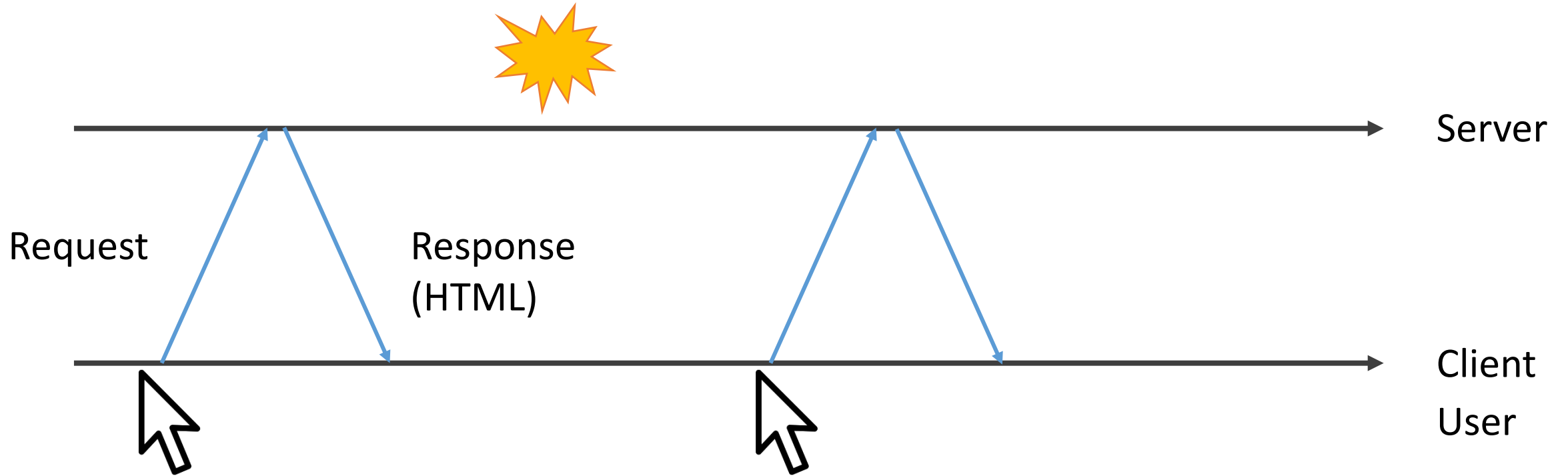
+ Message #announcements 😊

# Lernziele

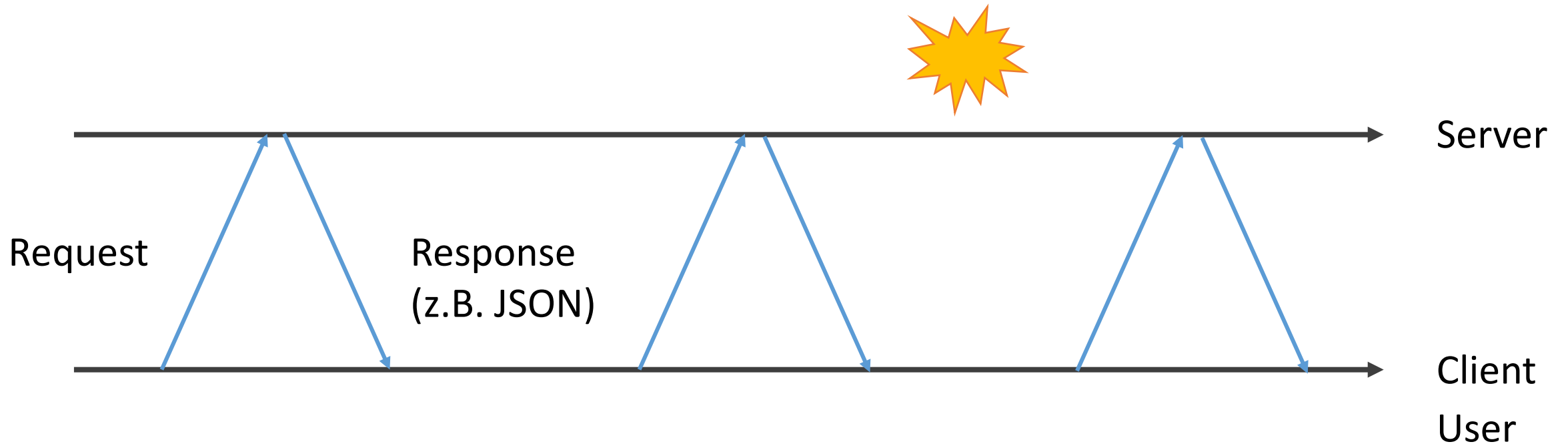
- Sie sind fähig, real-time Kommunikation zwischen Client und Server in einer einfachen Applikation mit Hilfe einer Bibliothek zu implementieren.
- Sie sind in der Lage, Notifications für eine Applikation zu gestalten und auszulösen in dem Sie die Web Notifications API benutzen.

# Real-Time Communication mit Websockets

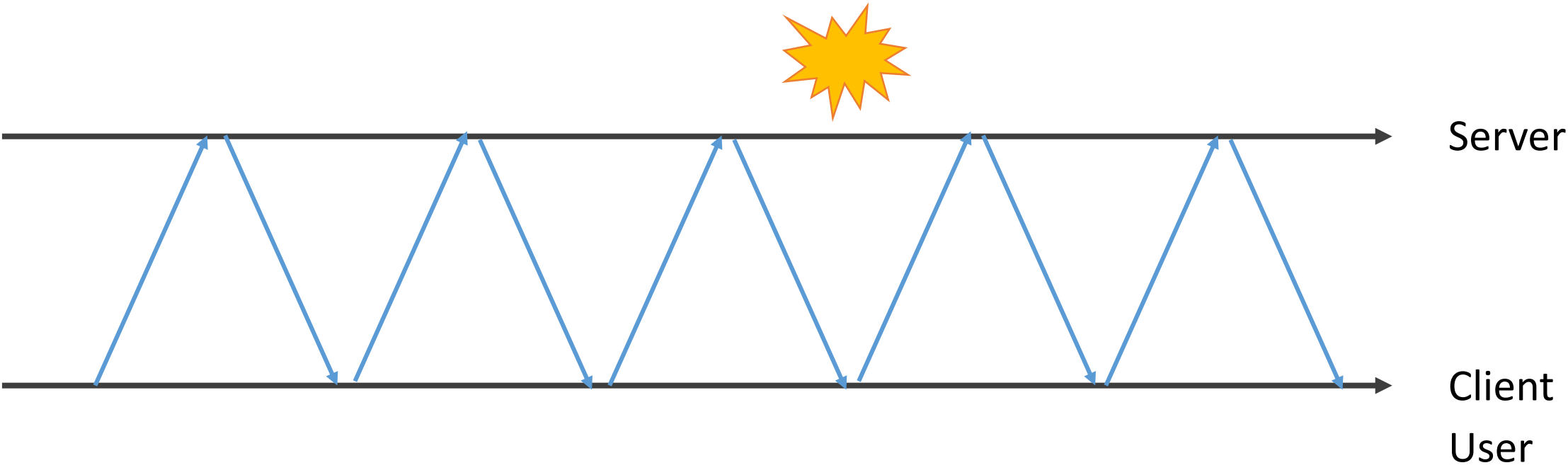
# Seite um Seite



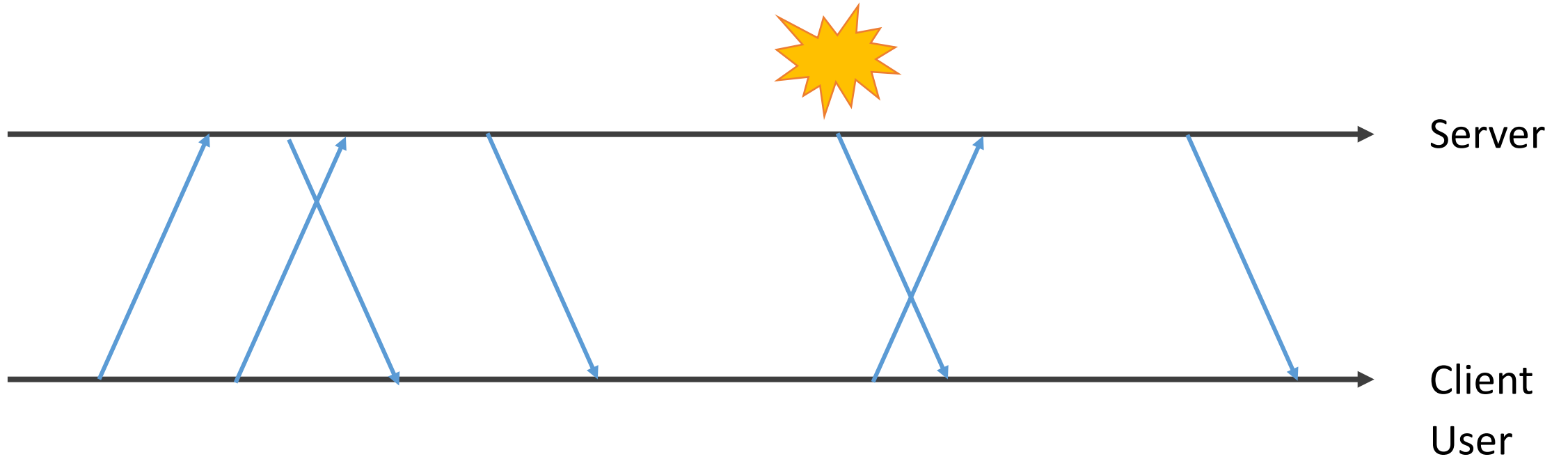
# AJAX



# AJAX Polling



# WebSockets





# WebSocket-Protokoll

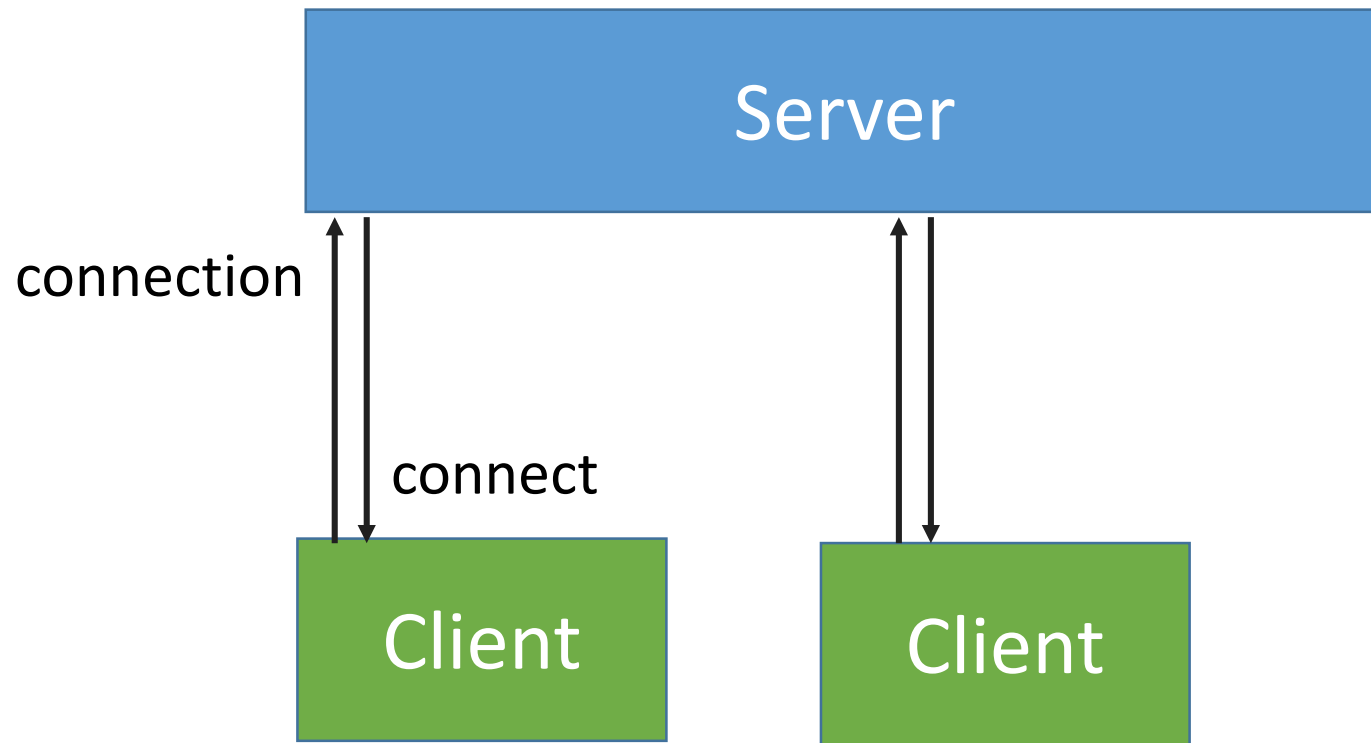
- Full-duplex Kommunikation über eine TCP-Verbindung
- *ws* und *wss* URI-Scheme: `ws://example.com:8042/ws/endpoint`
- HTTP-Handshake um die Verbindung aufzubauen

# Socket.io



- Bibliothek für real-time communication basierend auf WebSockets
- Unterstützt auch ältere Browser
- Abstraktionen wie Events
- Node.js-Server, JavaScript-Client

# Socket.io: Verbindungsaufbau



# Socket.io: Verbindungsaufbau im Server

```
let io = require('socket.io')(http);  
  
io.on("connection", socket => {  
    console.log(socket.id);  
});
```

# Socket.io: Verbindungsaufbau im Client

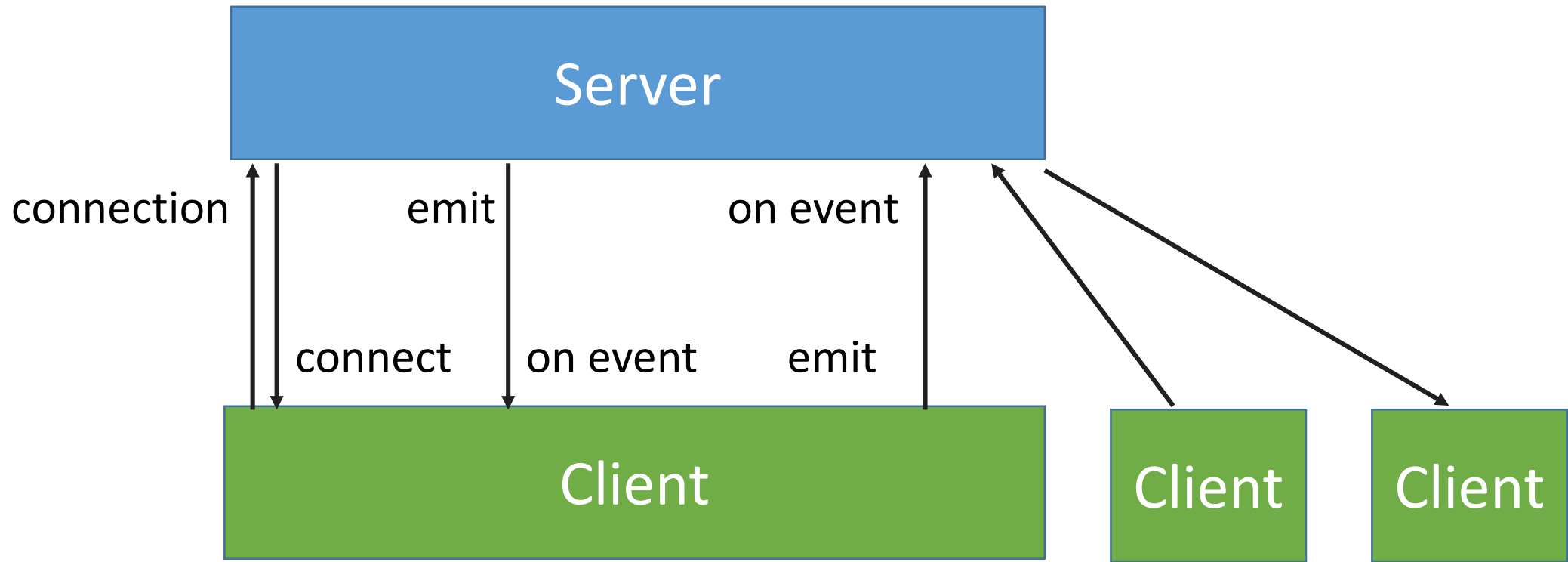
Im HTML

```
<script src="/socket.io/socket.io.js"></script>
```

Im JavaScript

```
window.socket = io();  
socket.on("connect", () => {  
    console.log(socket.id);  
});
```

# Socket.io: Kommunikation



# Socket.io: Events senden (Client und Server)

```
socket.emit("mein Event", {meine: "Daten"});
```

```
socket.emit("noch ein Event", "andere Daten");
```

# Socket.io: Events empfangen (Client und Server)

```
socket.on("mein Event", daten => {  
    console.log(daten);  
});
```



# Aufgabe 1.1

- Verbindung aufbauen
- Server schickt *nickname* an jeden neuen Client

```
names.getName ( ) ;
```

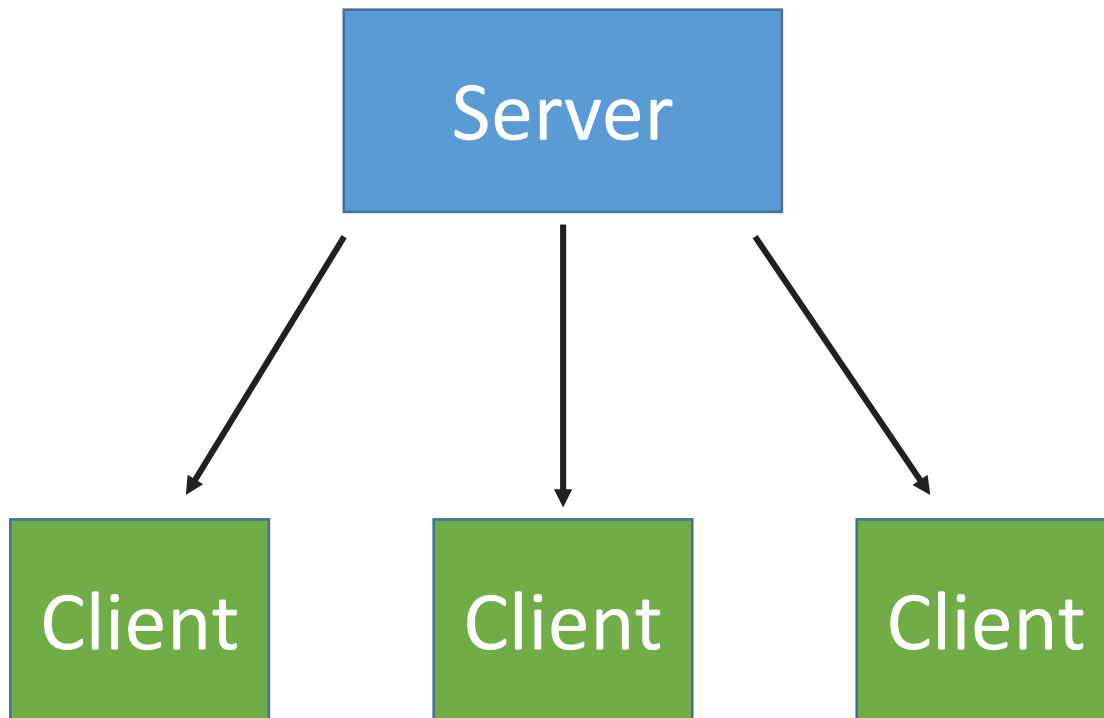
- Client zeigt Nickname an  
*setNickname ( name ) ;*

# Code-Setup

- Code-Skelet herunterladen von <https://github.com/mhusm/chat>
- Installieren mit yarn
- Node.js  $\geq$  v5.7
- Server starten mit *node index.js*
- Client verfügbar unter <http://localhost:8080/>
- Der Server-Code ist in *index.js*
- Der Client-Code ist im *public*-Ordner (*index.html* und *chat.js*)

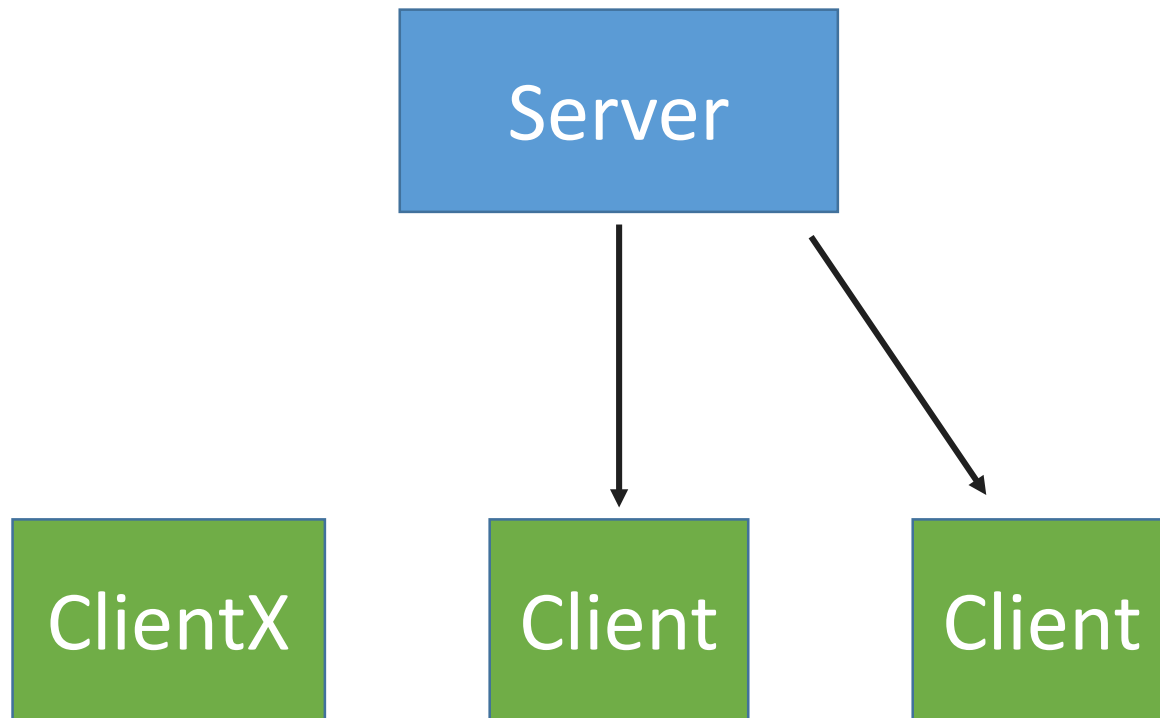
# Socket.io: Broadcasting

```
io.emit("mein Event", {meine: "Daten"});
```



# Socket.io: Broadcasting an alle ausser X

```
socketX.broadcast.emit("mein Event", {meine: "Daten"});
```



# Aufgabe 1.2

- Chat-Nachricht an alle anderen Teilnehmer senden

`let handleMessageInput = function...`

- Die Nachricht wird bei allen Teilnehmern angezeigt
- Der Sender wird auch angezeigt

`displayMessage(message, user) {`

# Socket.io: Verbindungsunterbrüche

```
socket.on("disconnect", reason => {  
    console.log(reason);  
});
```

# Aufgabe 1.3

- Eine Liste mit den Chat-Partnern anzeigen auf jedem Client
- Die Liste soll aktualisiert werden, wenn sich jemand verbindet oder abhängt

```
socket.nickname = nickname;
```

Server

```
getAllUsers()
```

```
displayUsers(other_users);
```

Client

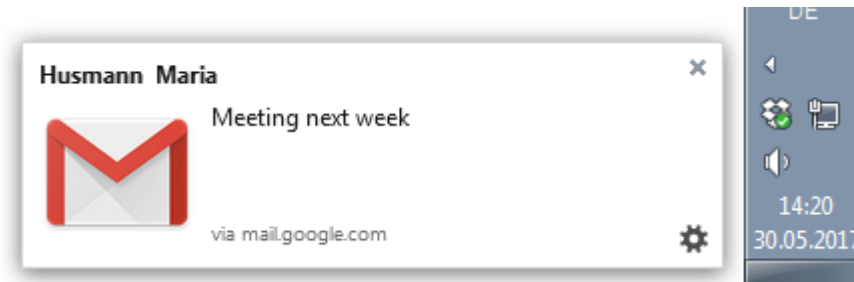
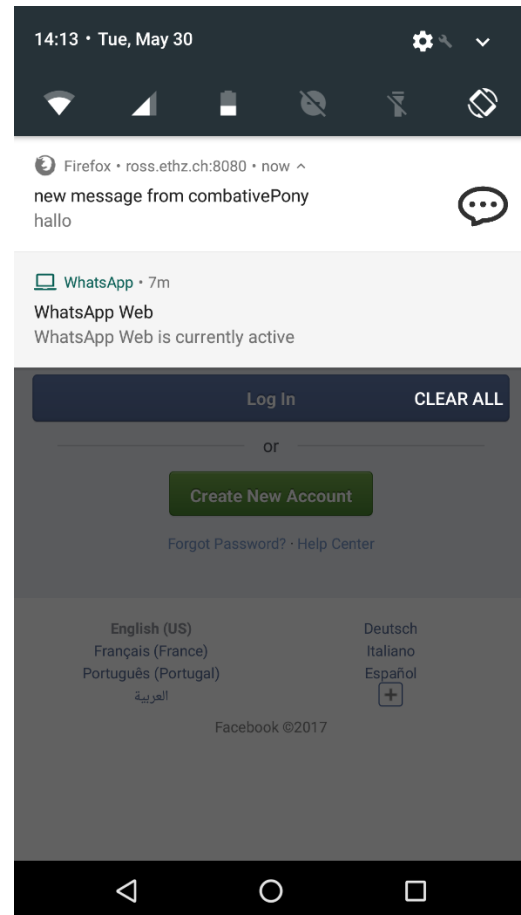
# Mögliche Erweiterungen

- Direkte Nachrichten senden
- Weitere Kanäle kreieren
- Chat History speichern
- Offline-Support

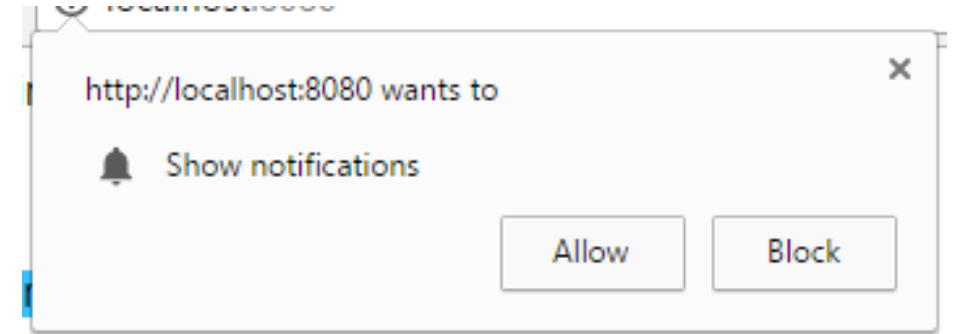
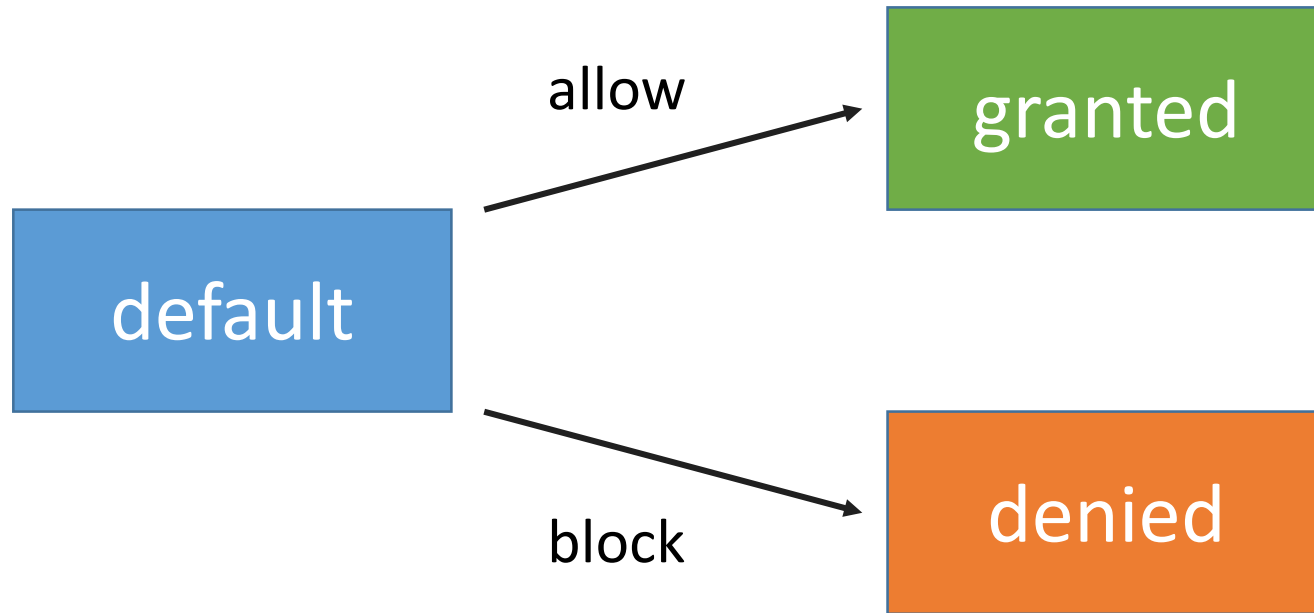


# Web Notifications API

# User Engagement mit Notifications



# Permission



Wann sollte man nach Permission fragen?

# Permission: Browser-Support & aktueller Status

```
if ( ! ( "Notification" in window ) ) {  
    console.warn( "This browser does not support  
notification" );  
}  
else if ( Notification.permission === "granted" ) {  
...  
}
```

# Permission: Benutzer anfragen

```
Notification.requestPermission(permission => {  
    console.log(permission);  
}) ;
```

# Aufgabe 2.1

- Aktuellen Zustand der Permission für Notifications abfragen
- Label für Notification mit aktuellem Zustand aktualisieren
- Falls der User noch nicht gefragt wurde, soll eine Klick auf das Label die Abfrage auslösen.

- Das Label soll farbig markiert werden

```
permissionSpan.classList.add( "active" );
```

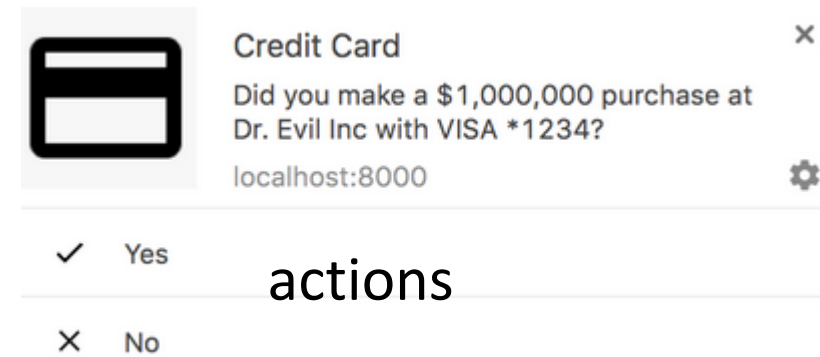
- Anschliessend das Label wieder korrekt aktualisieren

```
let enableNotifications = function ...
```

Wie könnte man Notifications im Chat nutzen?



# Gestaltung von Notifications



# Notifications auslösen

```
let options = {  
    body: "Mein Body text",  
    icon: "http://example.com/someImage.png"  
};  
let n = new Notification("Mein Titel", options);
```

## Aufgabe 2.2

- Eine Notification auslösen, sobald eine neue Nachricht ankommt.
- Als Icon kann z.B. das folgende verwendet werden oder ein eigenes  
<http://www.free-icons-download.net/images/chat-icon-68862.png>
- Was gehört in den Title und was in den Body?

Wann sollte man Notifications auslösen?

## Aufgabe 2.3

- Notifications nur auslösen, wenn das Chat-Tab nicht aktiv ist.
- Dazu kann man das Page Visibility API benutzen

```
if (document.hidden) {
```

# Interaktion mit Notifications

```
n.onclick = function () {  
    ...  
    this.close();  
};
```

# Aufgabe 2.4

- Ein Klick auf die Notification soll
  - die Applikation fokussieren  
`window.focus( ) ;`
  - die Notification schliessen

# Mögliche Erweiterungen

- Mehrere Notifications zusammenlegen
- Actions?



# Notifications: Browser-Support

## Web Notifications - LS

Method of alerting the user outside of a web page by displaying notifications (that do not require interaction by the user).

Global 37.1% + 5.4% = 42.5%  
unprefixed: 37.1%  
Switzerland 42.79% + 0.35% = 43.15%  
unprefixed: 42.79%

Current aligned Usage relative Date relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			56			9.3		4.4	
	14	52	57	10		10.2		4.4.4	
11	15	53	58	10.1	44	10.3	all	56	57
		54	59	TP	45				
		55	60		46				
		56	61						

Notes Known issues (3) Resources (10) Feedback

<sup>2</sup> Chrome for Android supports notifications via the [Push API](#) but not the Web Notifications API.

# Ausblick: Push API und ServiceWorker

- Notifications ohne, dass die Applikation im Browser geladen sein muss

# Zusammenfassung

- WebSockets mit Socket.io
  - Verbindungsaufbau im Client und Server
  - Events senden und empfangen
  - Broadcasting
  - Verbindungsunterbruch
- Web Notifications API
  - Permissions
  - Notifications gestalten und auslösen
  - Interaktion mit Notifications
  - User Experience?

# Ressourcen

- <https://developer.mozilla.org/en/docs/Web/API/notification>
- <https://socket.io/docs/#>
- <http://caniuse.com>