

Bachelorarbeit:

Verteiltes Veranstaltungsmanagement mit einer mobilen Webanwendung

Christian Meter

Institut für Informatik
Heinrich-Heine-Universität Düsseldorf

10.10.2013

Problembeschreibung

- 1 Veranstaltung
- 3500 Teilnehmer
- 89 ehrenamtliche Helfer, zunächst deutschlandweit verteilt
- Verschiedene Aufgaben
 - ① Anmeldungen bearbeiten
 - ② Helfer verwalten
 - ③ Statistiken erstellen und auswerten
 - ④ ...

Was wird benötigt?

Was wird benötigt?

MySQL

PHP5

HTML5

Was wird benötigt?

MySQL

Einfach

PHP5

Kompatibel

HTML5

Mobil

Was wird benötigt?

MySQL

Einfach

PHP5

Kompatibel

HTML5

Mobil



Meißner App

Grundfunktionen

Datenbanken

Veranstaltung

Benutzer

Grundfunktionen

Datenbanken

Veranstaltung

Geschlecht

Anreise Uhrzeit

:

Benutzer

Grundfunktionen

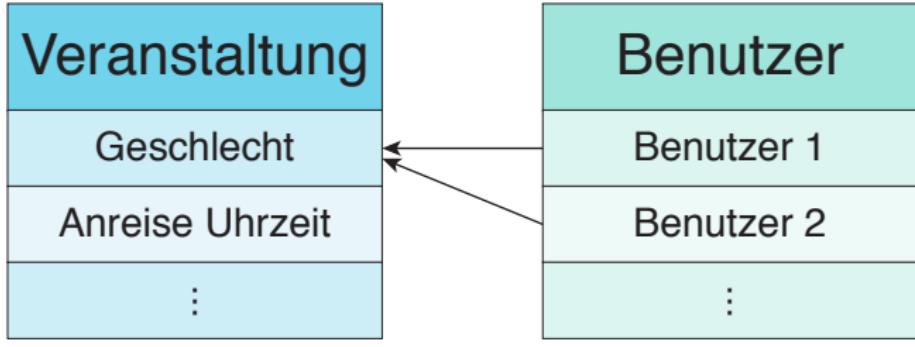
Datenbanken

Veranstaltung
Geschlecht
Anreise Uhrzeit
:

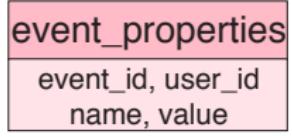
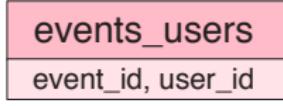
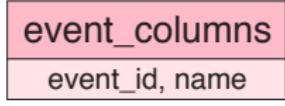
Benutzer
Benutzer 1
Benutzer 2
:

Grundfunktionen

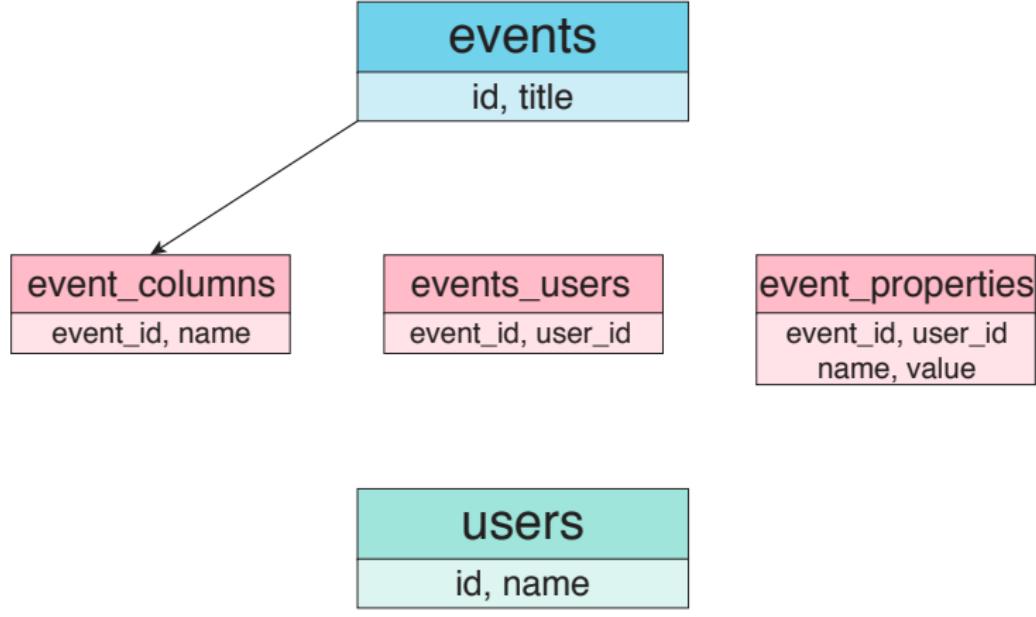
Datenbanken



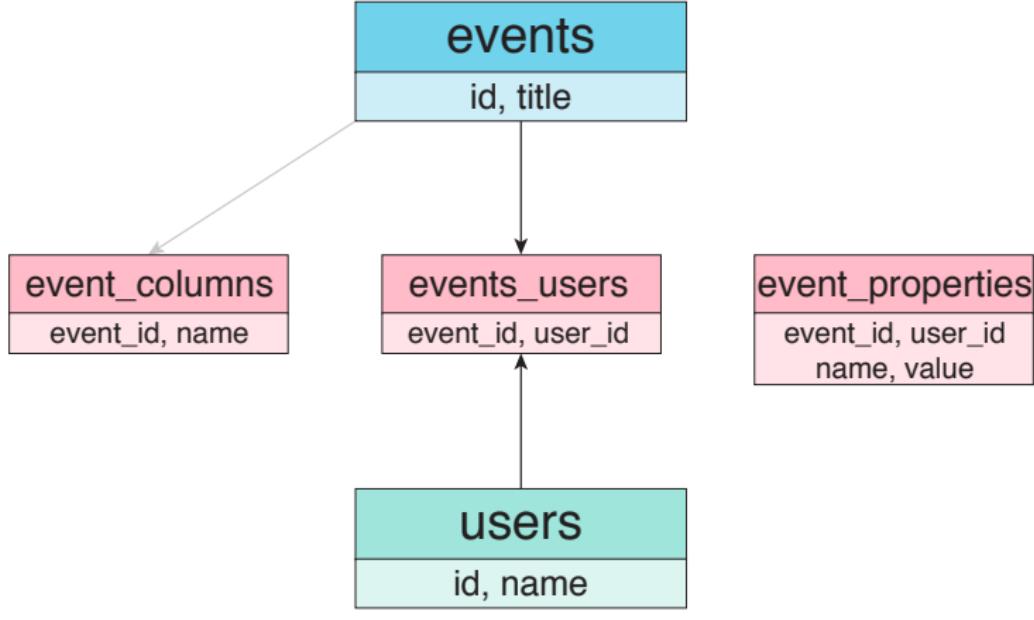
Datenbanken genauer betrachtet



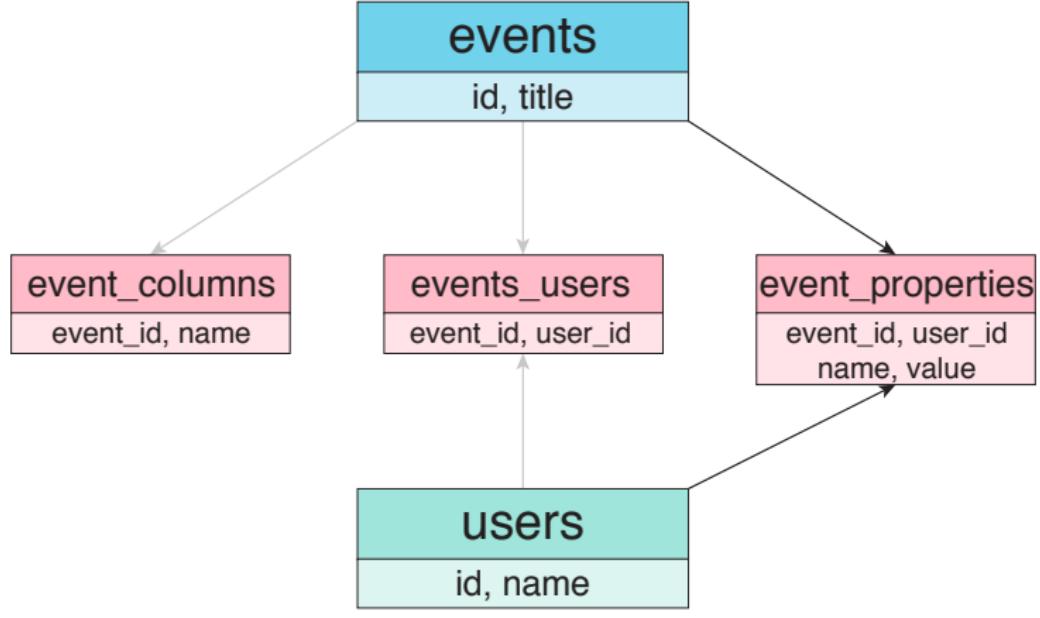
Datenbanken genauer betrachtet



Datenbanken genauer betrachtet



Datenbanken genauer betrachtet



Statistiken

- Tabellarische und grafische Auswertung
- Selbstständige Visualisierung!

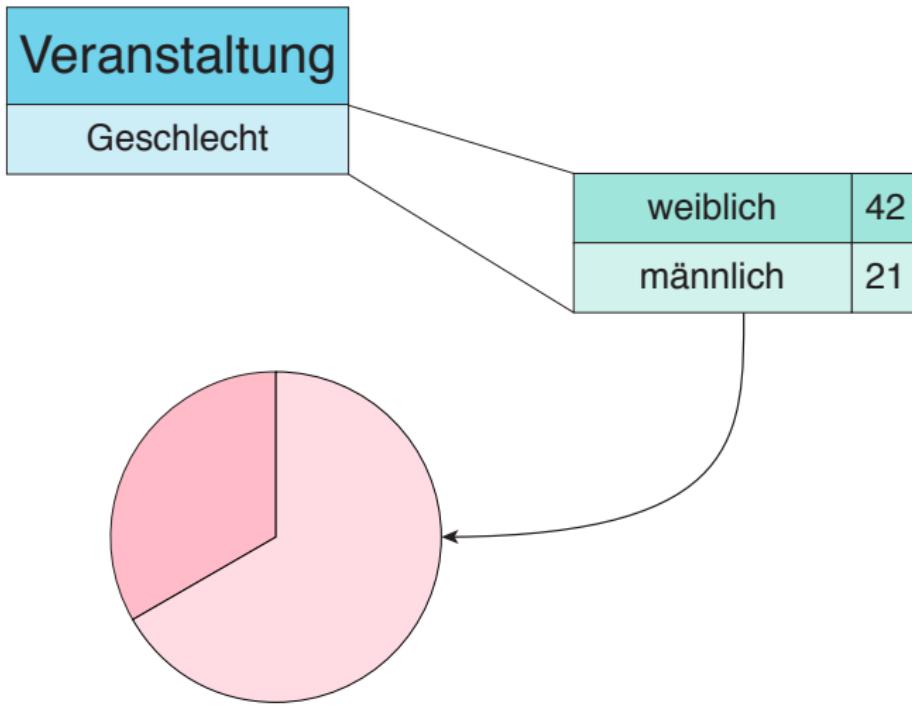
Statistiken

Veranstaltung
Geschlecht

Statistiken

Veranstaltung	Geschlecht	
	weiblich	42
	männlich	21

Statistiken

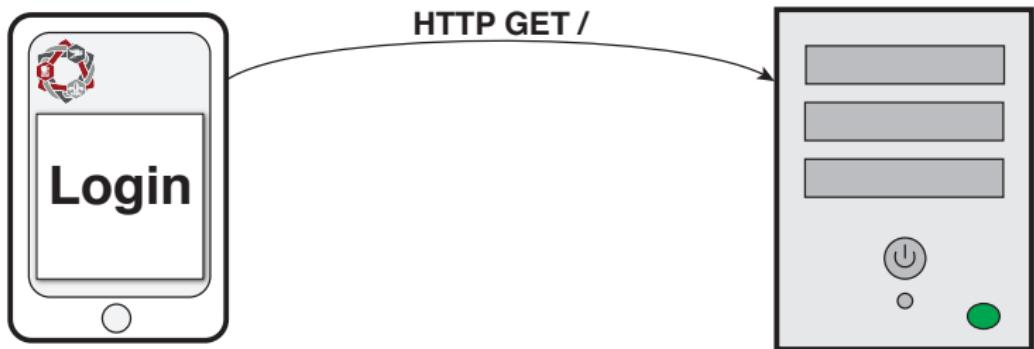


WebSockets



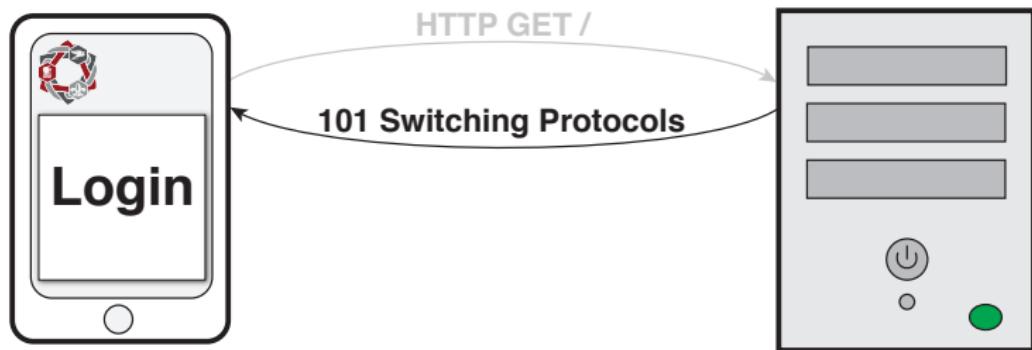
- TCP-basiertes Netzwerkprotokoll
- bidirektionale Verbindung zwischen Client und Webserver

WebSockets



- TCP-basiertes Netzwerkprotokoll
- bidirektionale Verbindung zwischen Client und Webserver

WebSockets



- TCP-basiertes Netzwerkprotokoll
- bidirektionale Verbindung zwischen Client und Webserver

WebSockets



- TCP-basiertes Netzwerkprotokoll
- bidirektionale Verbindung zwischen Client und Webserver

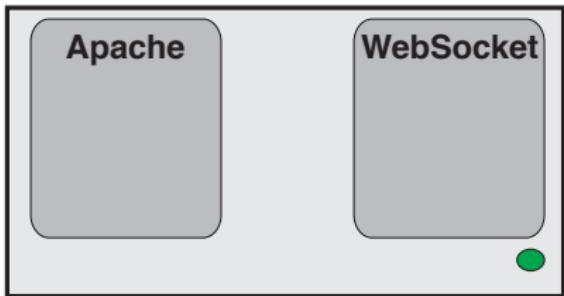
Authentifizierung beim WebSocket Server

Public Key Kryptographie



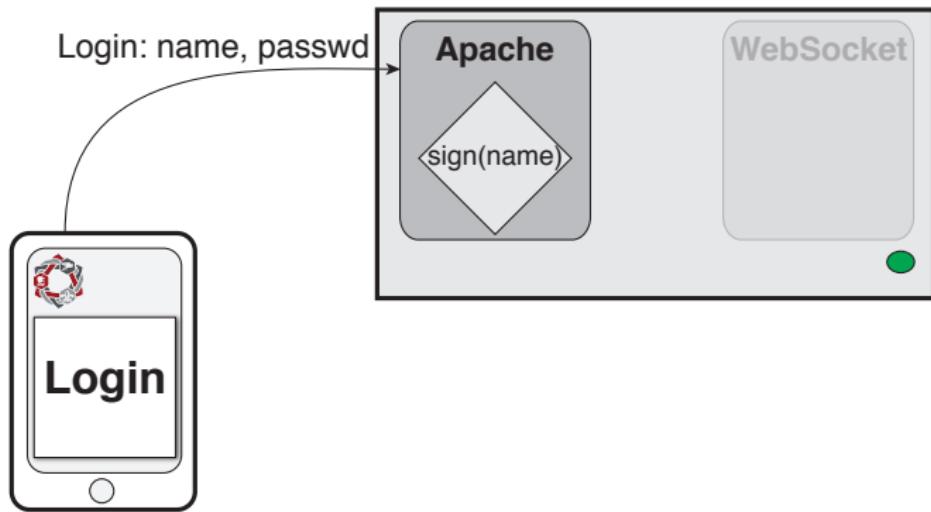
Authentifizierung beim WebSocket Server

Public Key Kryptographie



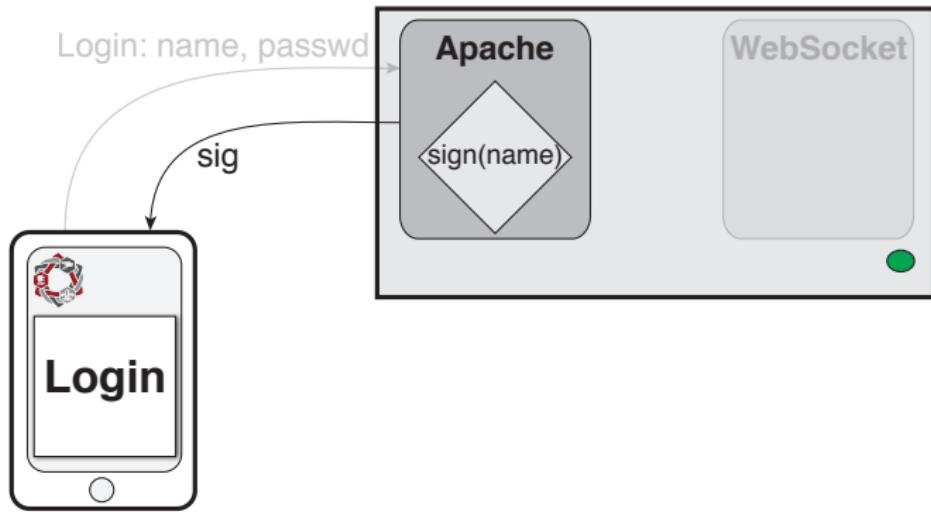
Authentifizierung beim WebSocket Server

Public Key Kryptographie



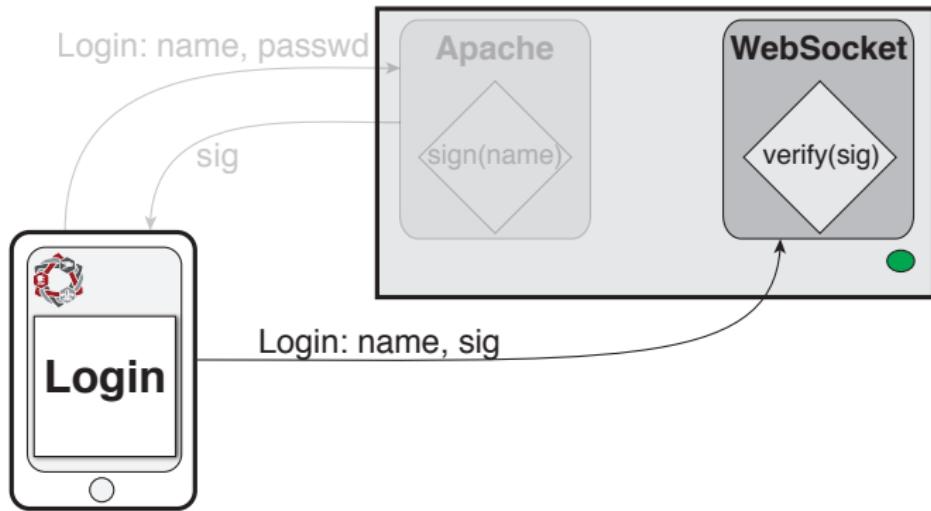
Authentifizierung beim WebSocket Server

Public Key Kryptographie



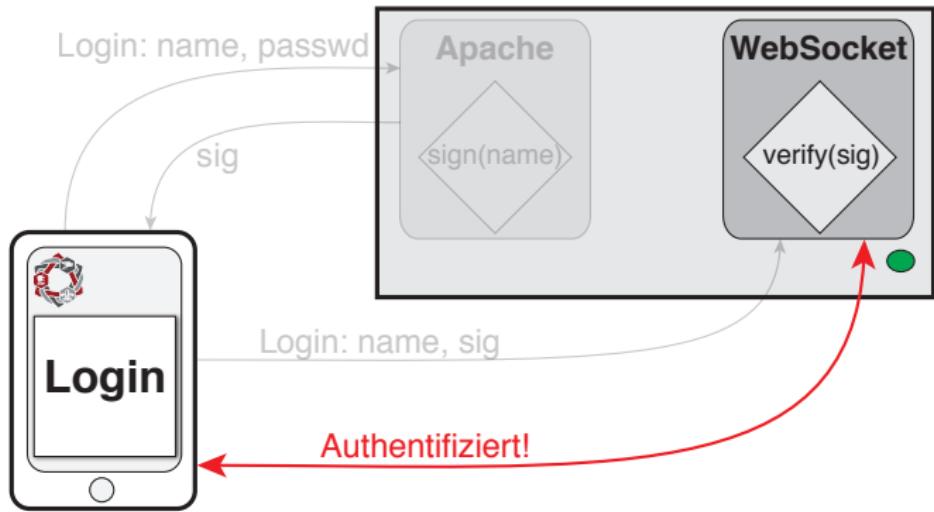
Authentifizierung beim WebSocket Server

Public Key Kryptographie



Authentifizierung beim WebSocket Server

Public Key Kryptographie



Publish / Subscribe

- Client abonniert eine Veranstaltung oder eine Seite
 ⇒ Ermöglicht automatische Updatebenachrichtigung, Chats, usw.
- Client und Server müssen nichts voneinander wissen
- Publish funktioniert ohne JavaScript!
 - Direkte Kommunikation zwischen Apache und WebSocket Server

Publish / Subscribe

- Client abonniert eine Veranstaltung oder eine Seite
 - ⇒ Ermöglicht automatische Updatebenachrichtigung, Chats, usw.
- Client und Server müssen nichts voneinander wissen
- Publish funktioniert ohne JavaScript!
 - Direkte Kommunikation zwischen Apache und WebSocket Server

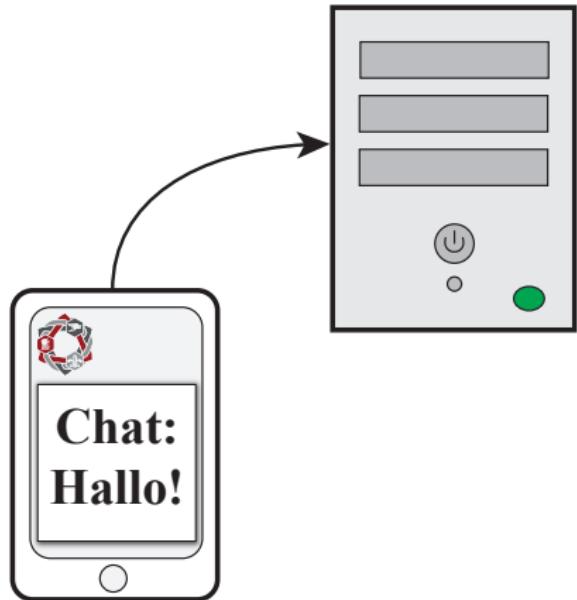
Wird hier genutzt für:

- ① Benachrichtigung bei Änderung einer Veranstaltung
- ② Chats
- ③ Geolokalisierung

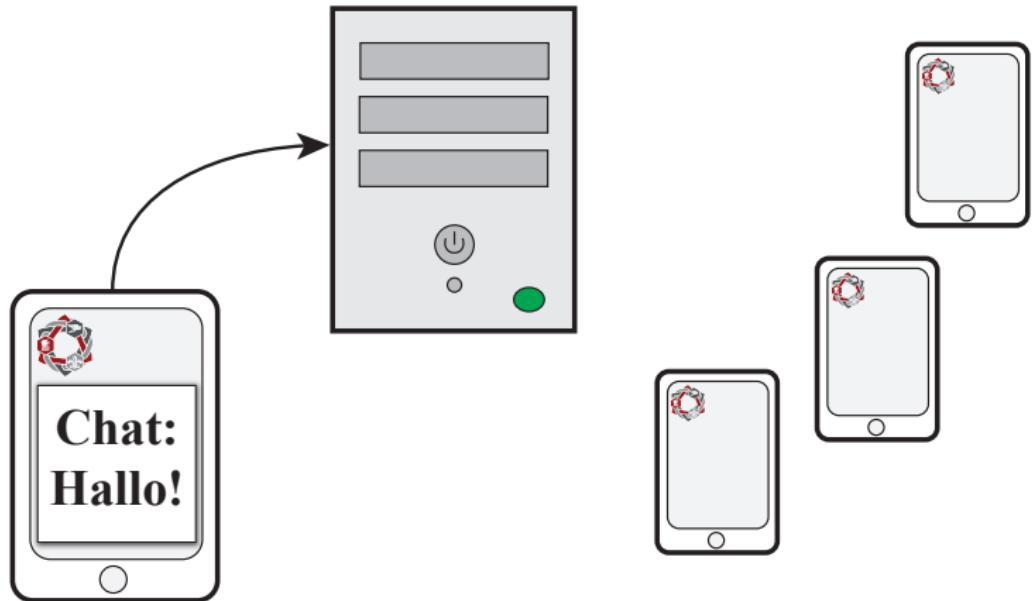
Chats



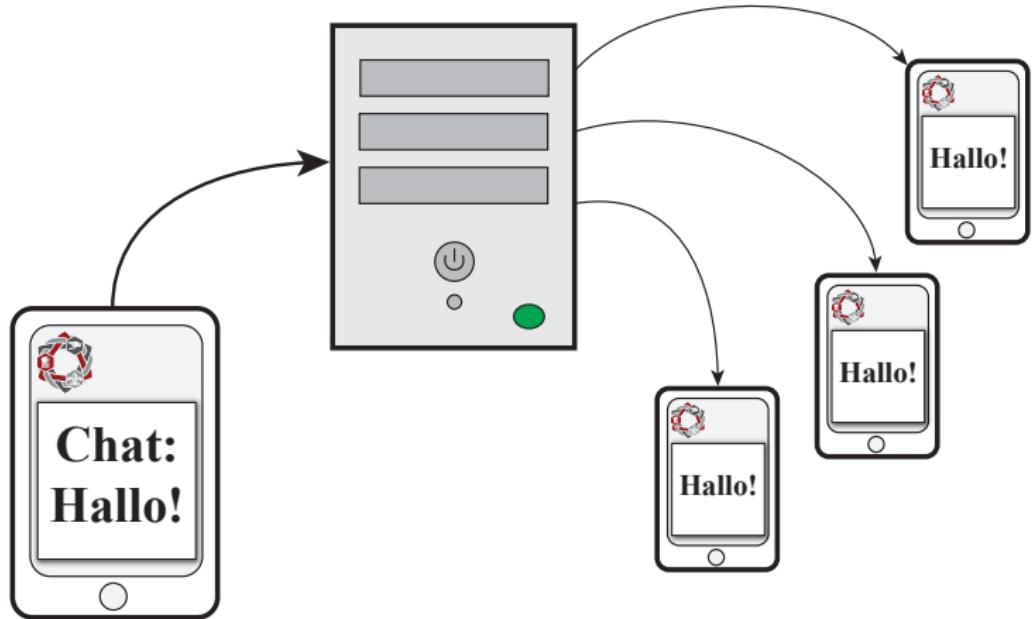
Chats



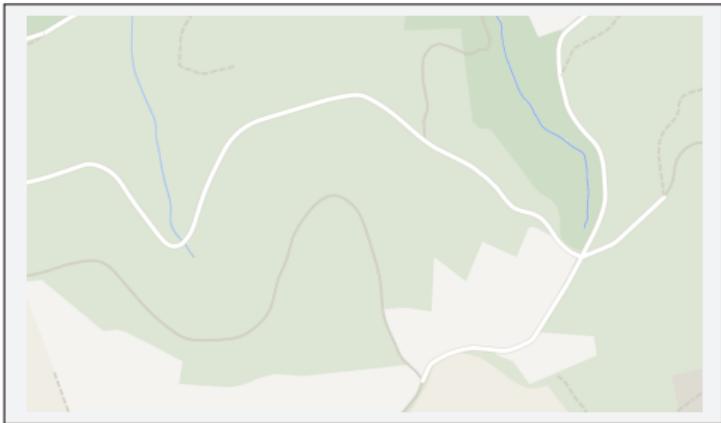
Chats



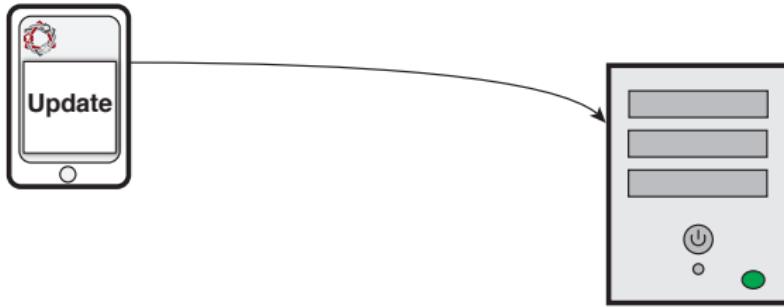
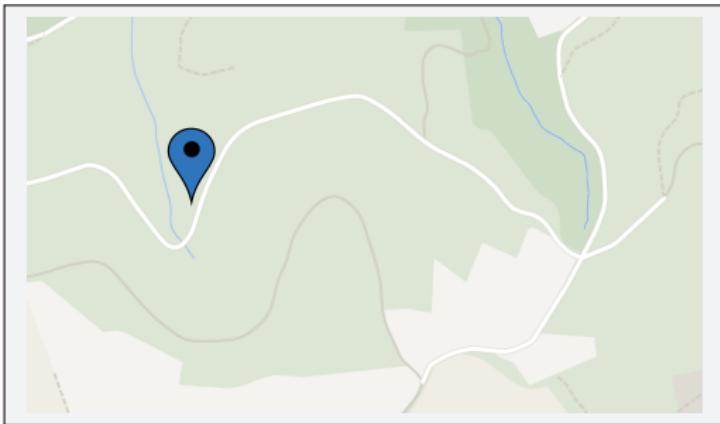
Chats



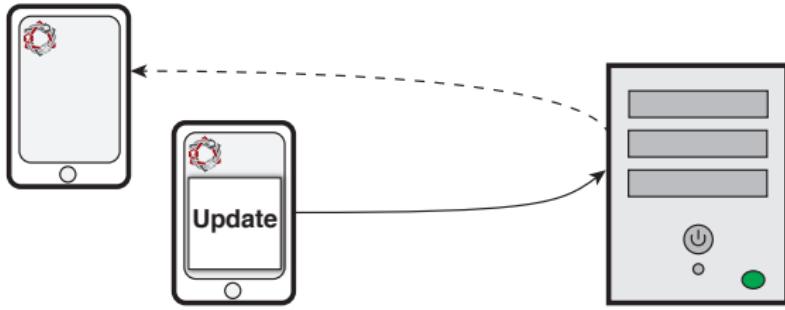
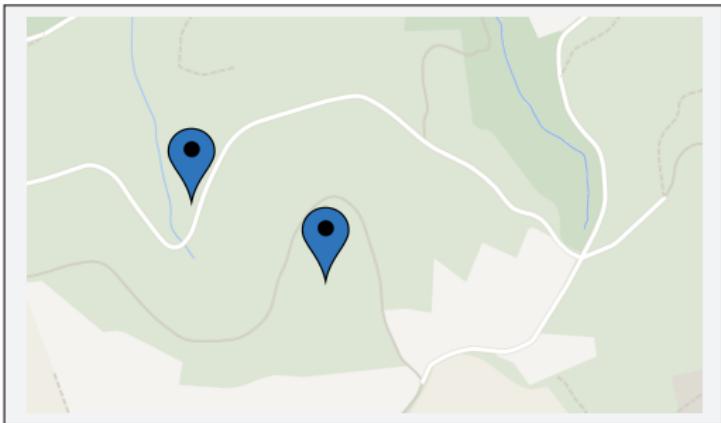
Geolocations



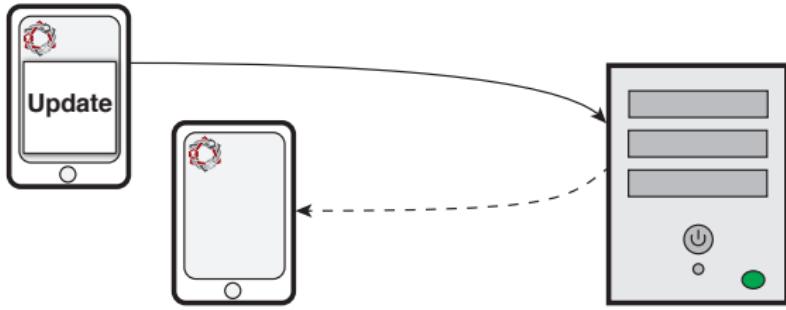
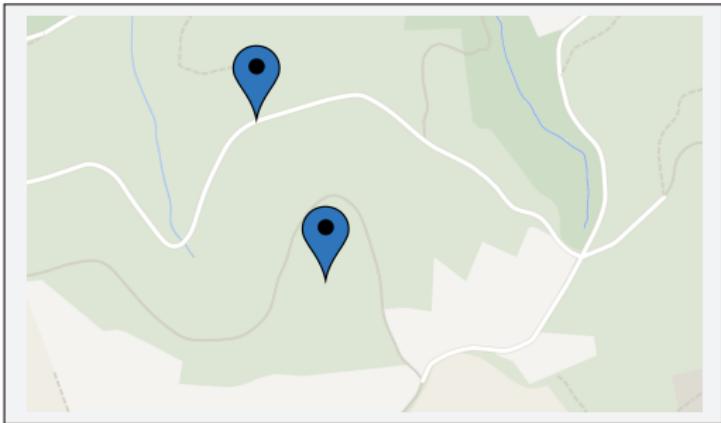
Geolocations



Geolocations



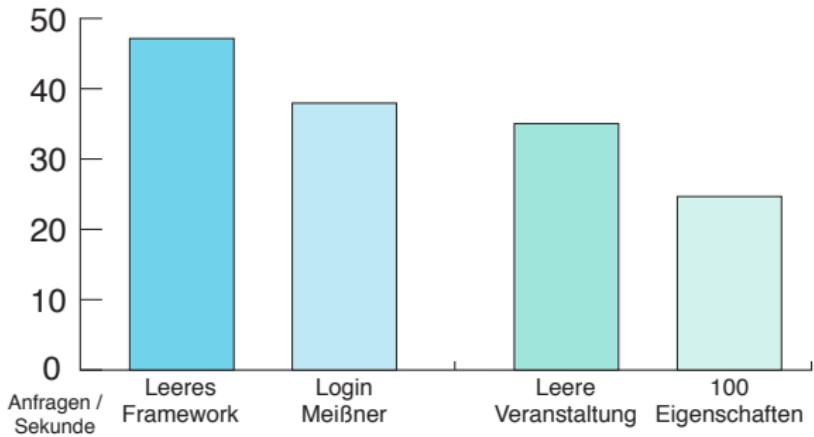
Geolocations



Offline Cache

- Automatisches Cachen von Bildern, Skripten, Seiten, ...
- Bei Änderung der Datenbank: Update des Caches notwendig
 - ⇒ Deutlich schnellere Bedienung
 - ⇒ Unempfindlicher für Funknetzabbrüche

Benchmark WebServer



- Apache Bench: 1000 Verbindungen, immer 10 gleichzeitig
- Praxis: Weniger Anfragen nötig durch Offline Cache

WebSocket Server

Netzwerkauslastung

- WebSockets: nur 2 Bytes Overhead!
- Normale HTTP Anfragen (Polling o.Ä.): 700-800 Bytes Overhead
- Besonders relevant bei hoher Anzahl von Clients

WebSocket Server

Netzwerkauslastung

- WebSockets: nur 2 Bytes Overhead!
- Normale HTTP Anfragen (Polling o.Ä.): 700-800 Bytes Overhead
- Besonders relevant bei hoher Anzahl von Clients, bspw. 1.000 Clients:

Polling	WebSockets	Einheit
800.000	2.000	$\frac{\text{Bytes}}{\text{Sekunde}}$
6.400.000	16.000	$\frac{\text{Bits}}{\text{Sekunde}}$
6,104 :-)	0,015 :-)	$\frac{\text{MBit}}{\text{Sekunde}}$

⇒ **Ersparnis von 400% Traffic**

Besonderheiten der Webanwendung

- Eigenständig lauffähige Webanwendung
- Echtzeitaktualisierung
- Unterstützung von mobilen Geräten
 - Offline Cache!
- Automatisches Installationsskript für debianbasierte Systeme
- Für die Nutzung im Freien ausgelegt
- Modular aufgebaut
- Ähnliche Produkte kosten mehrere hundert Dollar und sind meistens nur als Erweiterung für WordPress verfügbar

Demo

- Zeit die Anwendung zu testen!

Noch Fragen?

