

Aufgabenbeschreibung

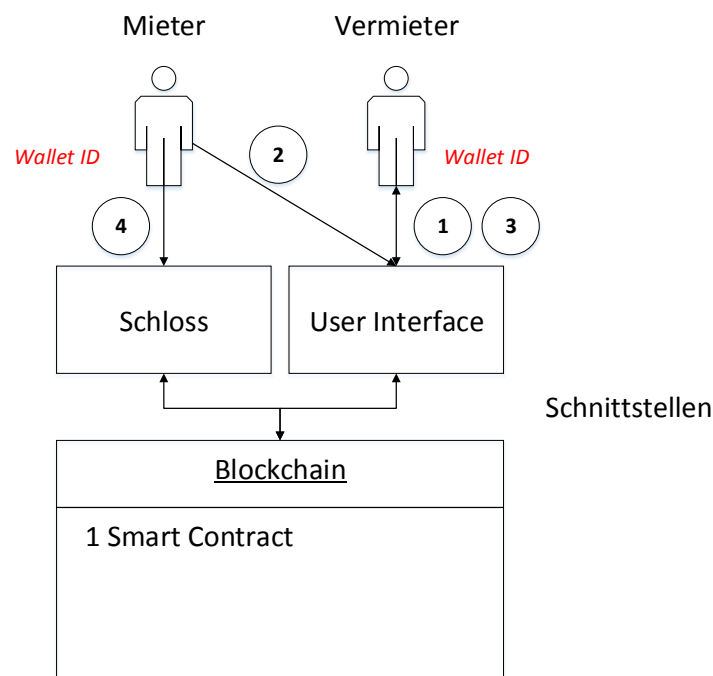
Gegeben (Minimum)

- 1 Zimmer (kann immer wieder vermietet werden)
- 1 Mieter (mietet ein Zimmer)
- 1 Vermieter (vermietet ein Zimmer)

Anmerkungen:

1. Es ist die Möglichkeit gegeben, während des gebuchten Zeitraums das Schloss mehrmals zu öffnen (Umsetzung mit Whisper).
2. Das Schloss soll als Hardware vorhanden sein.

Grafik



- 1 Vermieter legt Angebot über User Interface in Blockchain an (= Transaktion)
- 2 Mieter schaut, welche Angebote es gibt (Only Read) und entscheidet sich ggf.
- 3 Vermieter führt Buchung durch (nach Buchungsanfrage durch den Mieter, welche nicht vom Vermieter abgelehnt werden kann)
- 4 Mieter kann Schloss mit Smartphone öffnen

Anmerkungen:

1. Ein Smart Contract, welcher in einer Datenbank verwaltet wird, um die Daten immer auf den aktuellsten Stand zu halten.
2. In einem Smart Contract in der Blockchain wird immer das gleiche Booking durchgeführt.
3. Wie sieht die Kommunikation zwischen User Interface und Blockchain aus?
4. Ggf. im User Interface auswählen welche Technologie eingesetzt werden soll (Ethereum oder Hyperledger).

Aufbau eines Angebots (für ein Mietzimmer)

Variable	Datentyp
Angebot ID	Long
Tür ID	Long
Flag (besetzt / frei)	Boolean
Mietpreis (pro Nacht)	Long
Zeitraum (Check-in /Check-out)	Date (Verwendung von UNIX)
Adresse des Objekts	Text
Name des Vermieters	String
Mieter Wallet ID	
Vermieter Wallet ID → <i>Public Key</i>	
evtl. Zimmerbild	

Umzusetzende Punkte

1. *Smart Contract*
 - Ein oder mehrere Smart Contracts pro Angebot
 - Verwaltung von Public Keys (Ethereum-Seite) und überlegen wie diese in Block geschrieben werden
 - Evtl. Umsetzung mit dem Framework Truffle
 - Smart Contract ID referenziert immer auf Ursprungsblock
 - Welche Aktionen werden innerhalb des Smart Contracts ausgeführt?
2. *Whisper Messages Protokoll für Ethereum*
 - Skizze bereits vorhanden
 - Keine Transaktionen (und dadurch keine Kosten)
 - Integriertes Schloss (Hardware) → Zugriff über Blockchain
 - Smartphone sendet Token, welches zur Authentifizierung des Mieters durch den Vermieter dient
 - Whisper signiert und hinterlegt einen Public Key
3. *web3.js Client bei Ethereum Blockchain*
 - Umsetzung mit Angular und einem Node
 - Anbindung der Daten in der Blockchain allgemein
 - Kommunikation zwischen Blockchain und User Interface
4. *REST-Schnittstelle bei Hyperledger Blockchain*
 - Anbindung der Daten in der Blockchain allgemein
 - Kommunikation zwischen Blockchain und User Interface
5. *User Account für UI*
 - Eigene Verwaltung von Accounts, Geld und Keys
 - Plugin oder direkt über Ethereum bzw. Hyperledger Wallet
6. *User Interface*
 - Schickt Nachrichten an die Tür
 - Webinterface (grafische Benutzeroberfläche)
 - Frontend Darstellung / wie buche ich überhaupt?

Mögliche Meilensteine

1. Architektur festlegen (Skizze)
2. Schloss und User Interface Implementierungen & Implementierung und Integration der Komponenten in der Blockchain
3. Mieter und Vermieter mit Wallets einrichten
4. Testdurchführung des Minimums mit 1 Zimmer, 1 Mieter und 1 Vermieter
5. Evtl. Umsetzung für mehrere Zimmer / Mieter / Vermieter

Organisatorisches

- Pflege von Einzelaufgaben und Einzelnen Projektteilen übernimmt jede Kleingruppe selbst.
- Besprechung immer Dienstag 30-60 Minuten. Danach Arbeiten in Kleingruppen.

Kleingruppen:

- | | |
|--------------------------------|------------------------------------|
| ○ Hyperledger: | Deniz, Frank |
| ○ Tür mit Whisper: | Felix, Max, Peter |
| ○ Smart Contract mit Ethereum: | Anna, James |
| ○ Web3 Client: | Alex, Daniel, (Max), Michi, (Rene) |