

DSA-Testat

Ether-Chat

Sandro Blatter, Matthias Dunkel,
Raffael Vögeli



Ablauf

- Genutzte Technologien
- Architektur
- Schwierigkeiten
- Demo

Genutzte Technologien

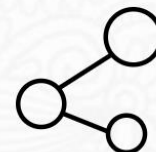
- Vue.js
- Web RTC
- Web3js
- Node JS

Vue.js



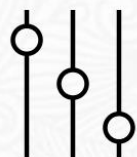
EASE OF USE

Clear syntax, live preview



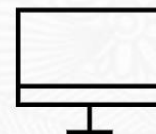
MODULARITY

Reuse slides and styles



INTERACTIVITY

Embed forms and live demos



PORTABILITY

Play in any browser

Web RTC

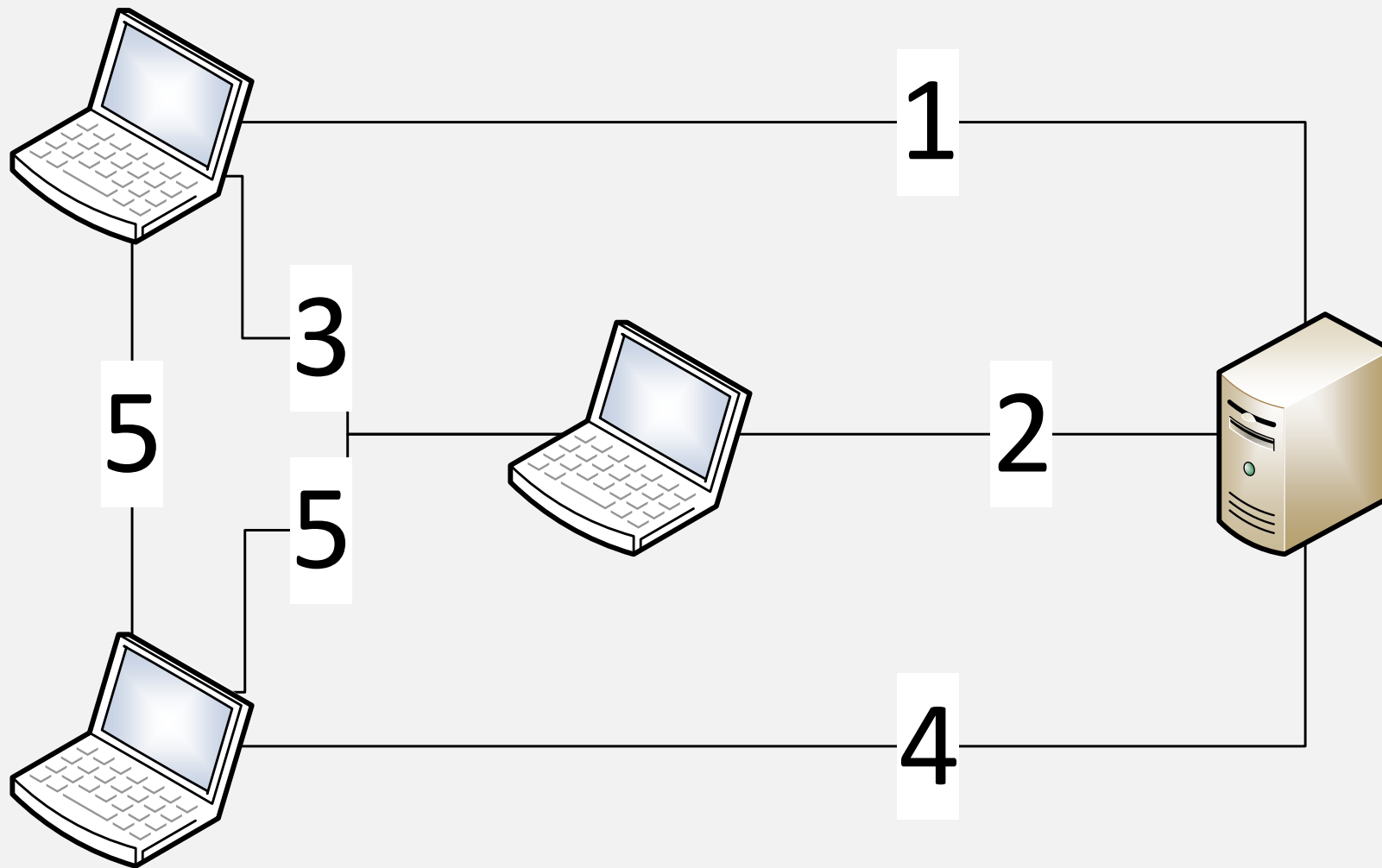
- Wieso WebRTC?
 - Keine Plugins
 - Peer to Peer
 - Gute aktuelle Dokumentation
 - Kombination mit JavaScript

Web3js

- API Client ⇔ Smart Contract
- Smart Contract gespeichert via
 - remix.ethereum.org
- Verwalten der Konten mit Metamask 
- Aufruf von Funktionen im Smart Contract

```
1  pragma solidity ^0.4.23;
2
3  contract notaryContract {
4      mapping (address => mapping(bytes32 => uint)) messages;
5
6      function store(address sender, bytes32 hash) public {
7          messages[sender][hash] = block.timestamp;
8      }
9
10     function verify(address sender, bytes32 hash) public constant returns (uint) {
11         return messages[sender][hash];
12     }
13 }
```

Architektur



Schwierigkeiten

- Web3js
 - Welche Version Beta oder 1.x
 - Zusammenarbeit MetaMask mit Web3js
- Interpretation der Anforderungen
 - Welche Art der Bestätigung?
 - Welche Technologien dürfen verwendet werden?

Demonstration

