

DSA Chat

Benjamin Daniels, Zarko Dragojevic & Micha Schena

Technologien pro Aufgabe

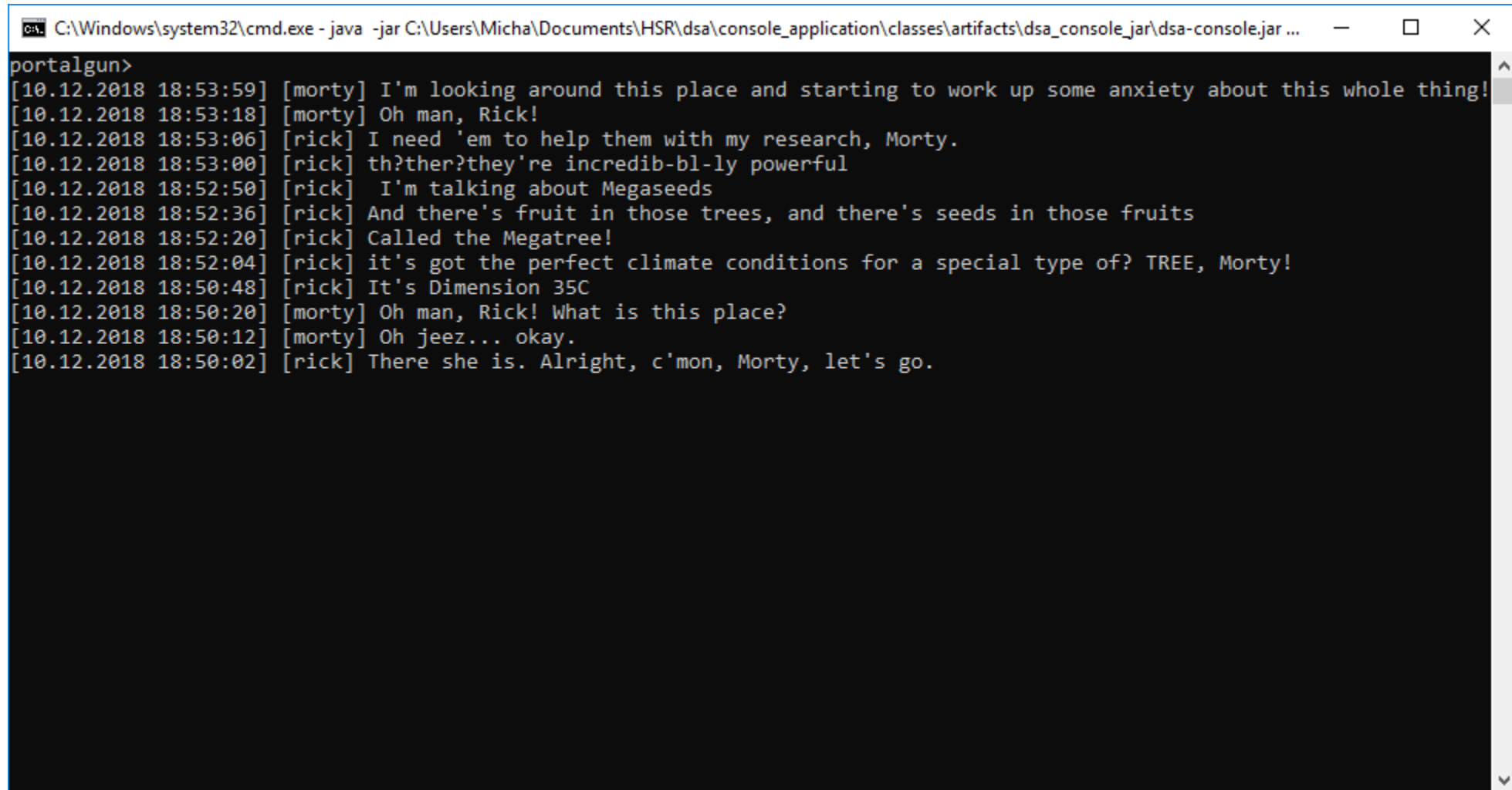
- Verteilte Chat-Applikation:

TomP2P

- Eingeschriebener Brief:

Smart Contract auf ETH + WEB3.js + Metamask Plugin

Zwischenstand



```
C:\Windows\system32\cmd.exe - java -jar C:\Users\Micha\Documents\HSR\dsa\console_application\classes\artifacts\dsa_console_jar\dsa-console.jar ...  
portalgun>  
[10.12.2018 18:53:59] [morty] I'm looking around this place and starting to work up some anxiety about this whole thing!  
[10.12.2018 18:53:18] [morty] Oh man, Rick!  
[10.12.2018 18:53:06] [rick] I need 'em to help them with my research, Morty.  
[10.12.2018 18:53:00] [rick] th?ther?they're incredib-bl-ly powerful  
[10.12.2018 18:52:50] [rick] I'm talking about Megaseeds  
[10.12.2018 18:52:36] [rick] And there's fruit in those trees, and there's seeds in those fruits  
[10.12.2018 18:52:20] [rick] Called the Megatree!  
[10.12.2018 18:52:04] [rick] it's got the perfect climate conditions for a special type of? TREE, Morty!  
[10.12.2018 18:50:48] [rick] It's Dimension 35C  
[10.12.2018 18:50:20] [morty] Oh man, Rick! What is this place?  
[10.12.2018 18:50:12] [morty] Oh jeez... okay.  
[10.12.2018 18:50:02] [rick] There she is. Alright, c'mon, Morty, let's go.
```

Architektur

- Backend:

Spring Boot REST API + WebSockets

- Frontend:

Angular mit Material Design + SockJS + StompJS + WEB3.js

Architektur TomP2P

- Bootstrap mit **peerId = Number160.createHash(name);**
- Userinfos (Name und Wallet) werden an Position **peerId** gespeichert:
peer.put(peerId).domainKey(USER_NAME_KEY).data(new Data(name))
- Userliste über **PeerBean**:
peer.peer().peerBean().peerMap().all()
- Messaging über **SendDirect**:
peer.peer().sendDirect(receiverAddress).object(message)

Architektur Ethereum Smart Contract

- WEB3.js
- Metamask
- Smart Contract mit Solidity auf Ethereum Blockchain

```

contract NotaryService {
    struct Letter {
        address sender;
        address receiver;
        string message;
        bool isRead;
        bool isAccepted;
        bool isValue;
    }

    mapping(bytes32 => Letter) letters;
    bytes32[] letterIds;

    function getLetterIds() public constant returns (bytes32[]) {
        return letterIds;
    }

    function getLetter(bytes32 id) public constant returns (address sender, address receiver, string message, bool isRead, bool isAccepted) {
        return (letters[id].sender, letters[id].receiver, letters[id].message, letters[id].isRead, letters[id].isAccepted);
    }

    function createLetter(bytes32 id, address receiver, string message) public {
        require(!letters[id].isValue);
        letters[id] = Letter(msg.sender, receiver, message, false, false, true);
        letterIds.push(id);
    }

    function acceptLetter(bytes32 id, bool isAccepted) public {
        require(letters[id].receiver == msg.sender && !letters[id].isRead);
        letters[id].isRead = true;
        letters[id].isAccepted = isAccepted;
    }
}

```

Demonstration

Users

Contacts



Rick

wallet

2018-12-10 19:47:14



Morty

wallet

2018-12-10 19:47:14

Users



Karl

wallet

2018-12-10 19:47:14

Invite



Tom

wallet

2018-12-10 19:47:14

Invite

Rooms

Create Room

Rooms

DSA Room
Rick, Morty

Garage
Morty

Room DSA Room



beth

yeah

2018-12-10 19:44:28



Morty

yeah

2018-12-10 19:44:50



Rick

whatsuup

2018-12-10 19:45:06



Morty

rooms

2018-12-10 19:45:25

Type your message

Fragen?