Distributed Systems Advanced & Blockchain

Challenge Task

Lukas Lätsch
Rolf Furrer

11. Dezember 2018

dsl.hsr.ch

Inhalt

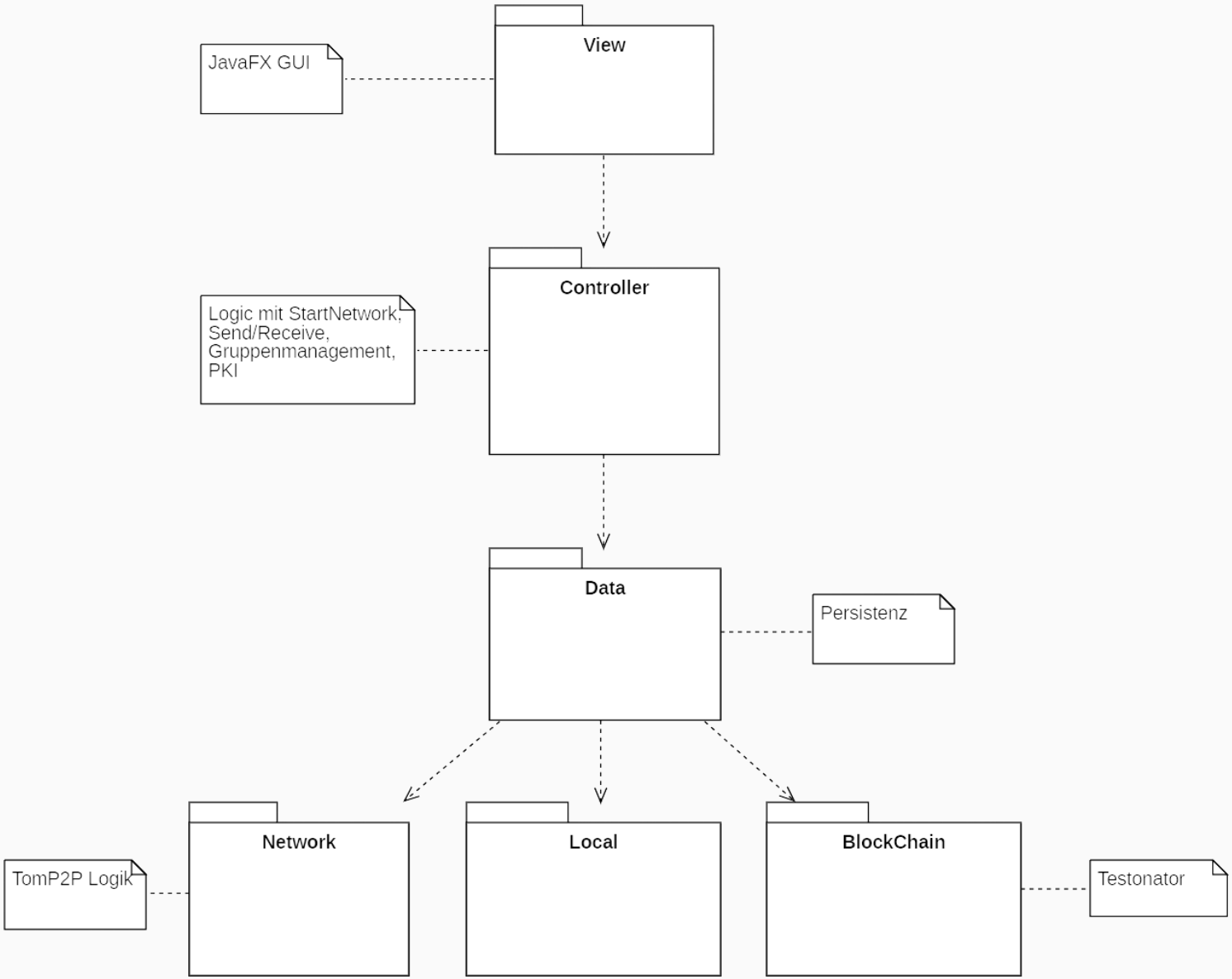
Übersicht

Verwendete Technologien

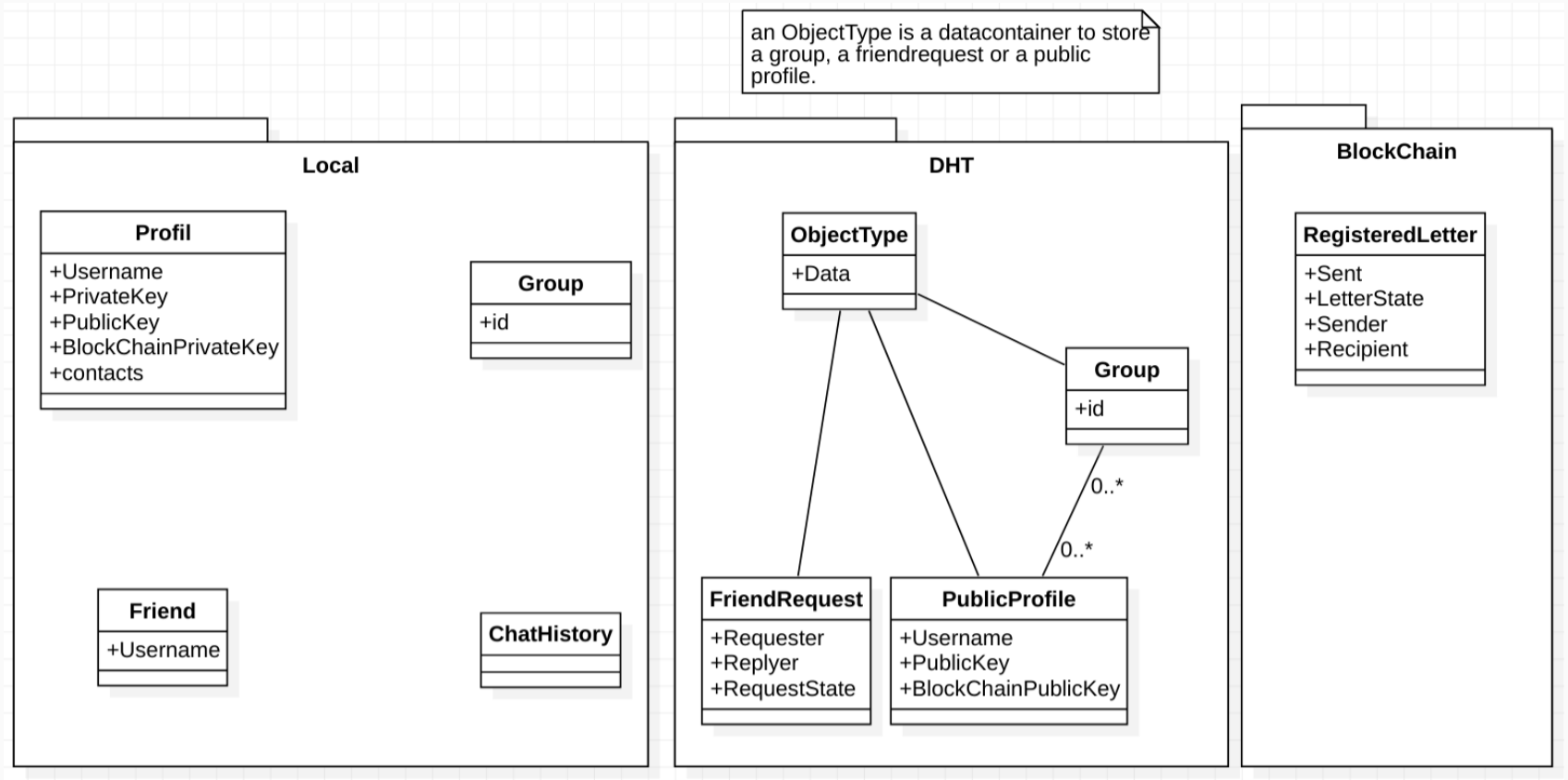
- Java 8
- Java FX
- TomP2P
- Testonator, mit Rinkeby-Testnet
- Gitlab CI
- JUnit 5

Inhalt

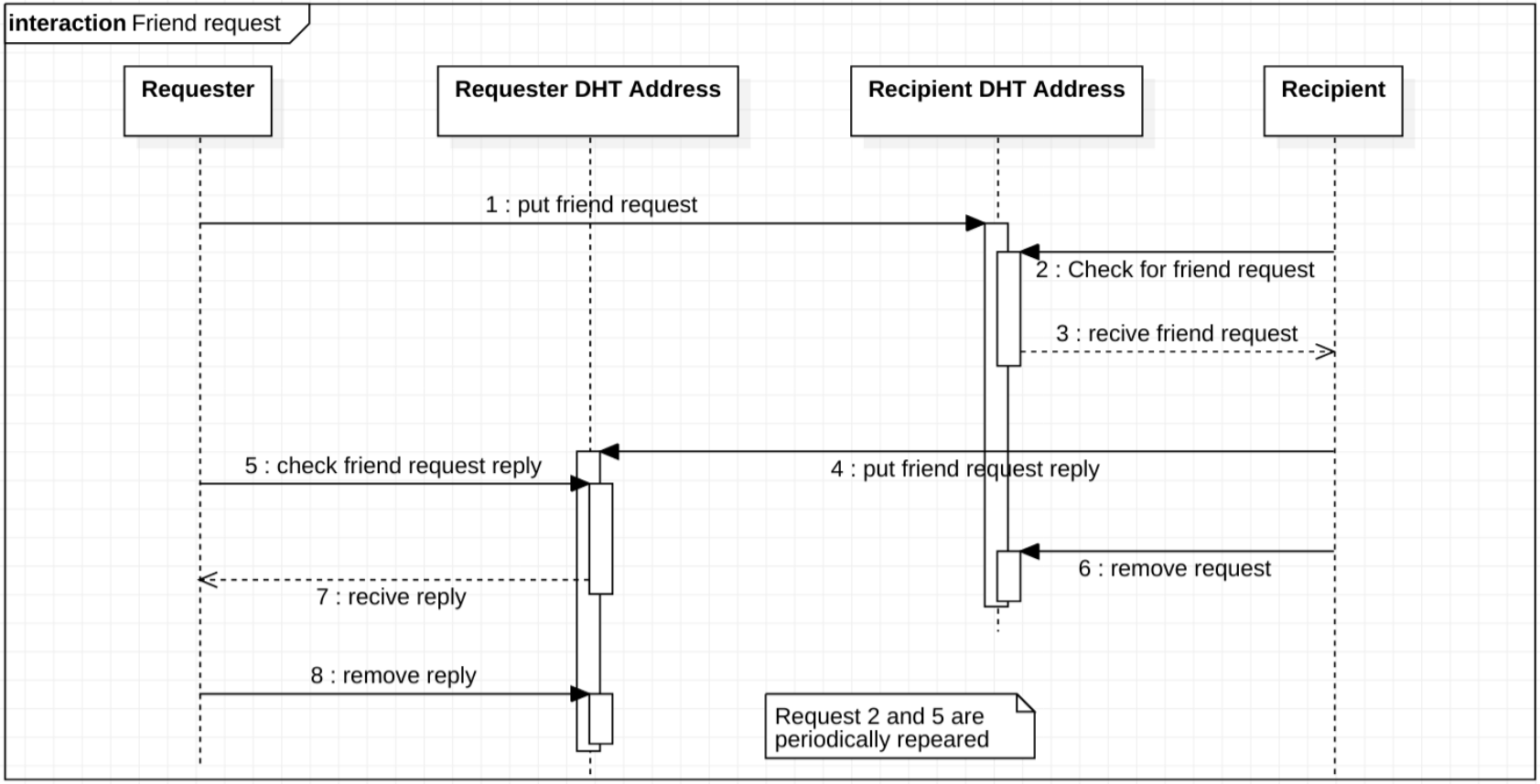
Architektur



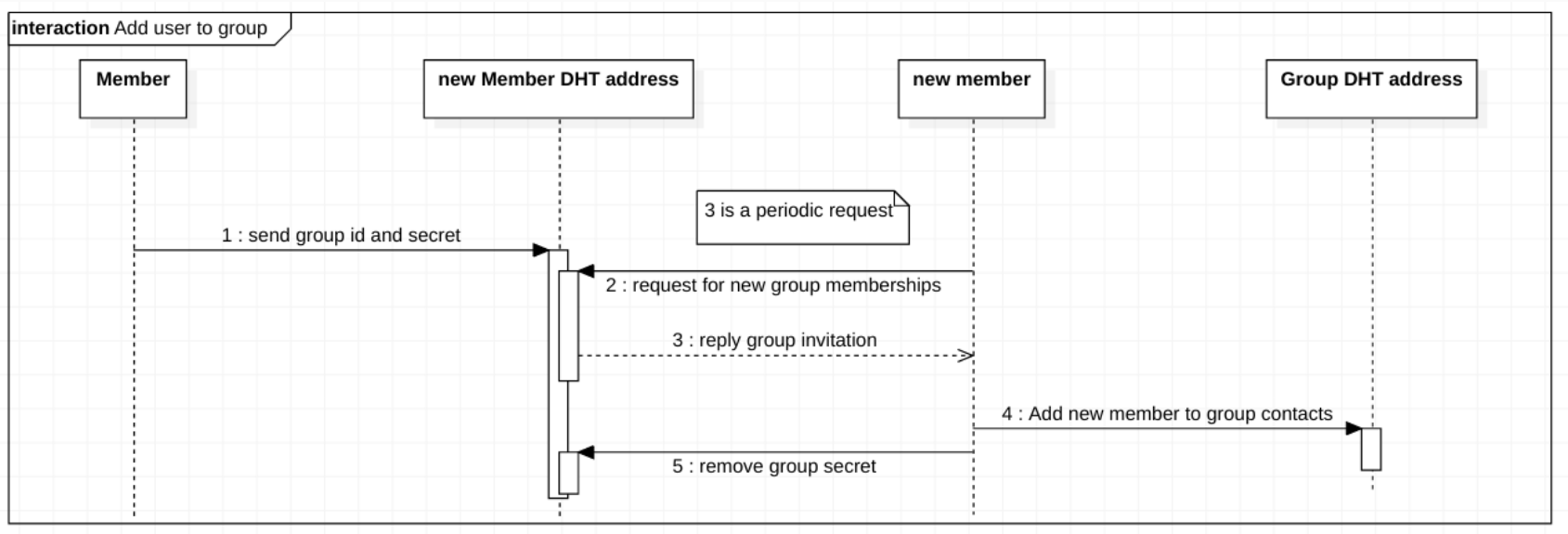
Datenlokation



Freundschaftsanfrage



Gruppen-Einladung



Inhalt

Verschlüsselung

- Lokal Daten AES verschlüsselt
- DHT direkt Chat Daten RSA verschlüsselt
- DHT Gruppen Chat Daten AES verschlüsselt

TomP2P

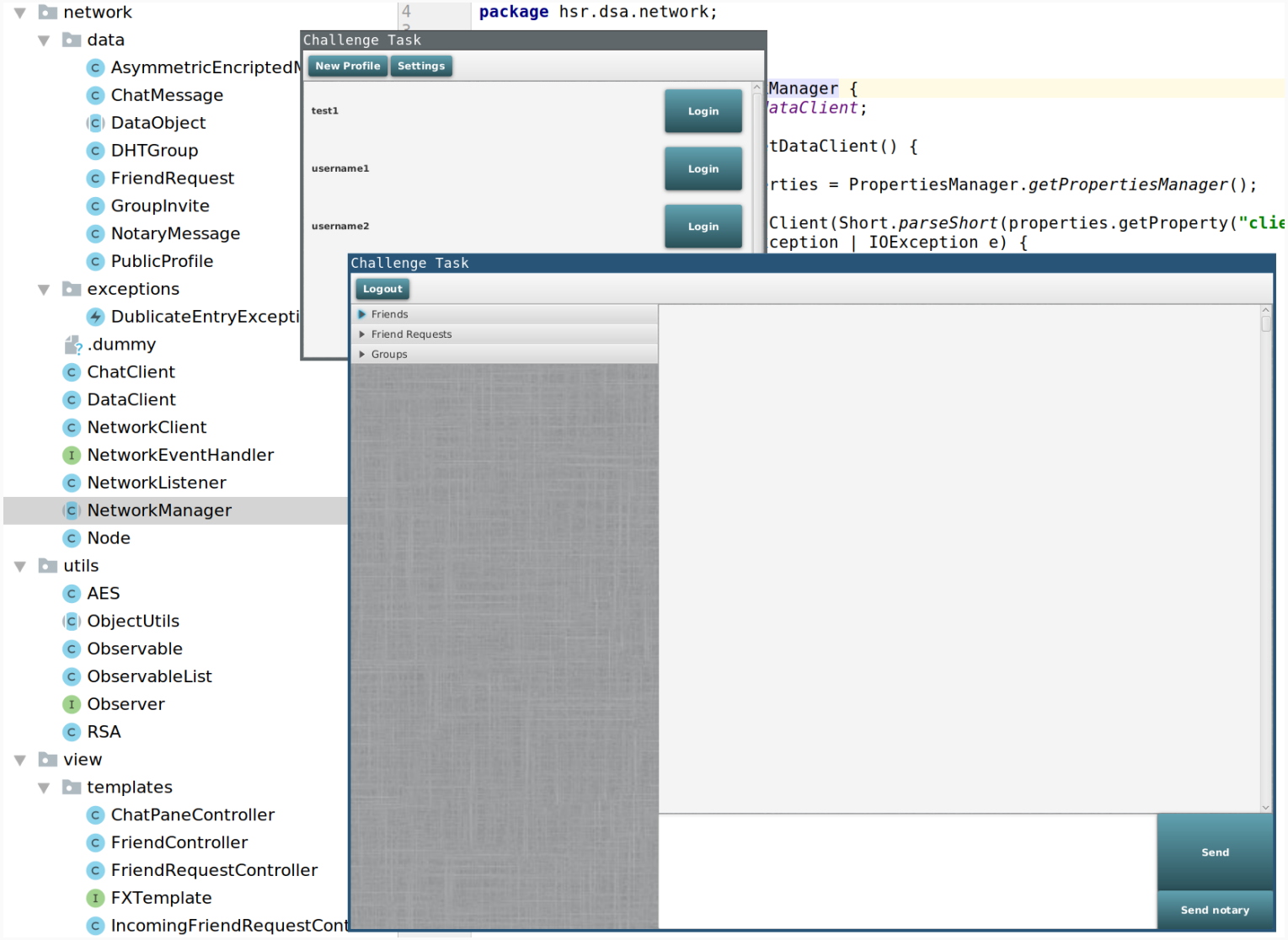
- Tomp2p 5.0-Beta8
- Asynchrone Kommunikation
- Kommunikation mit UI über Observer pattern

Testonator

```
pragma solidity ^0.5.00;

contract Notary {
    mapping (address => mapping (bytes32 => NotaryTransaction)) stamps;

    struct NotaryTransaction {
        address originator;
        address recipient;
        uint256 uploaded;
        uint256 received;
        uint256 acknowledged;
        bool accepted;
    }

    function upload(bytes32 hash, address recipient) public {
        stamps[recipient][hash] = NotaryTransaction(msg.sender, recipient, block.timestamp, 0, 0, false);
    }

    function received(bytes32 hash) public {
        NotaryTransaction memory t = stamps[msg.sender][hash];
        if(t.recipient == msg.sender && t.uploaded > 0 && t.received == 0) {
            t.received = block.timestamp;
            stamps[msg.sender][hash] = t;
        }
    }

    function acknowledge(bytes32 hash, bool accept) public {
        NotaryTransaction memory t = stamps[msg.sender][hash];
        if(t.recipient == msg.sender && t.uploaded > 0 && t.received > 0 && t.acknowledged == 0 ) {
            t.acknowledged = block.timestamp;
            t.accepted = accept;
            stamps[msg.sender][hash] = t;
        }
    }

    function verifyUpload(address recipient, bytes32 hash) public view returns (uint256) {
        return stamps[recipient][hash].uploaded;
    }

    function verifyReceived(address recipient, bytes32 hash) public view returns (uint256) {
        return stamps[recipient][hash].received;
    }

    function verifyAcknowledged(address recipient, bytes32 hash) public view returns (uint256) {
        return stamps[recipient][hash].acknowledged;
    }

    function isAccepted(address recipient, bytes32 hash) public view returns (bool) {
        return stamps[recipient][hash].accepted;
    }
}
```

Inhalt

Demo


network
data
AsymmetricEncriptedN
ChatMessage
DataObject
DHTGroup
FriendRequest
GroupInvite
NotaryMessage
PublicProfile
exceptions
DublicateEntryExcepti
.dummy
ChatClient
DataClient
NetworkClient
NetworkEventHandler
NetworkListener
NetworkManager
Node
utils
AES
ObjectUtils
Observable
ObservableList
Observer
RSA
view
templates
ChatPaneController
FriendController
FriendRequestController
FXTemplate
IncomingFriendRequestCont
package hsr.dsa.network;
Manager {
ataClient;
tDataClient() {
rties = PropertiesManager.getPropertiesManager();
Client(Short.parseShort(properties.getProperty("clie
ception | IOException e) {
Challenge Task
New Profile
Settings
test1
Login
username1
Login
username2
Login
Challenge Task
Logout
Friends
Friend Requests
Groups
Send
Send notary

Inhalt

Fragen


DHT COLLISION & SERVICES
COMPLETE AUTO REPAIR · PAINTING & BODY WORK
288-8856
61 STOCKTON AVE.
60 - 62
ESTIMATES