

# rdrop - Projekt Dokumentation

Lars Zocher, Simon Stiegler, Philipp Eichwald

14. Mai 2023

## 1 Einleitung

In der modernen Welt ist der Austausch von Daten ein immer wichtiger werdender Bestandteil des alltäglichen Lebens. Dennoch bieten die meisten Anbieter für das Übertragen von Daten keine Möglichkeit, die Daten auf direktem Wege zwischen den Nutzern auszutauschen. In den meisten Fällen müssen die Daten zunächst auf einem Server zwischengespeichert werden, damit die empfangende Partei diese Daten herunterladen kann.

Dieser Umweg ist nicht nur Zeitaufwändig, sondern auch ein Sicherheitsrisiko, da die Anbieter, die die Daten zwischenspeichern, Zugriff auf die Daten erhalten [1]. Eine Alternative dazu bietet AirDrop. AirDrop ist ein Dienst der Firma Apple der es ermöglicht Daten direkt zwischen zwei Geräten zu übertragen. Jedoch besitzt AirDrop den entscheidenden Nachteil, dass sich die Geräte in unmittelbarer Nähe zueinander befinden [2].

Alle bisherigen Lösungen benötigen mindestens einen Server, um die Kommunikation zwischen beiden Parteien aufzubauen. Eine wirklich einfache Lösung für die gesicherte direkte Übertragung von Daten zwischen zwei Parteien existiert bisher nicht.

Ziel dieses Projekts ist es, eben dieses Problem zu lösen. Dafür wird eine Anwendung entwickelt, die es ermöglicht ohne die Verwendung eines Server Daten

auf direktem Wege zwischen zwei Parteien auszutauschen. Die Anwendung soll dabei auf allen gängigen Betriebssystemen lauffähig sein. Insbesondere wird ein Fokus auf die unkomplizierte Bedienung der Anwendung gelegt, um so auch unerfahrenen Nutzern die Möglichkeit zu bieten, Daten sicher und schnell auszutauschen.

Das folgende Dokument beinhaltet die Dokumentation des Projekts. Dabei werden zunächst die Anforderungen an die Anwendung definiert. Anschließend wird die Architektur der Anwendung erklärt. Schlussendlich wird die Umsetzung und Lösung des Projekts kritisch reflektiert und ein Ausblick auf mögliche Erweiterungen gegeben.

## 2 Anforderungen

Im folgenden sind die Anforderungen an die Software definiert. Die Anforderungen werden dabei in funktionale und nicht funktionale Anforderungen unterteilt.

### Funktionale Anforderungen

**Anforderung 1: Lauffähigkeit** Die Software soll auf allen gängigen Betriebssystemen lauffähig sein.

**Anforderung 2: Übertragung** Die Software soll es ermöglichen, ohne die Verwendung eines Servers, Daten direkt zwischen zwei Parteien zu übertragen.

**Anforderung 3: Sicherheit** Die gesamte Kommunikation zwischen den Parteien soll verschlüsselt erfolgen.

**Anforderung 4: Dateien** Die Software soll die Möglichkeit bieten, Dateien zwischen den Parteien auszutauschen. Diese sollen automatisch aus dem eingelesen und abgespeichert werden.

## Nicht-Funktionale Anforderungen

**Anforderung 1: Benutzerfreundlichkeit** Die Software soll eine leicht verständliche Oberfläche bieten. Zu jedem Zeitpunkt muss klar sein, welche Aktionen möglich sind. Es werden keine überflüssigen Informationen angezeigt.

## 3 Architektur

Insgesamt ist die Softwarelösung in die drei Module Benutzeroberfläche, Kommunikation und Datei-I/O aufgeteilt. Grund dafür ist, dass insbesondere die Module Kommunikation und Datei-I/O auch unabhängig voneinander arbeiten können.

So wird die Kohäsion der einzelnen Teile erhöht und die Kopplung verringert. Im folgenden werden die einzelnen Teile genauer erklärt.

### 3.1 Benutzeroberfläche

architektur und so

### 3.2 Kommunikation

Das Modul Kommunikation ist für den Verbindungsaufbau und den Nachrichtenaustausch zwischen den Parteien zuständig. Nach außen soll es eine stark abstrahierte Schnittstelle bieten, die Fehlbenutzung vermeidet und die Verwendung maximal vereinfacht.

Das Modul soll in die Untermodule *Client* und *Protokoll* aufgeteilt werden. Das Modul *Client* bietet dabei die Schnittstellen für das Senden und Empfangen von Nachrichten. Mithilfe des *Protokoll* Moduls wird die Möglichkeit geboten die Verbindung zweier *Client* Instanzen aufzubauen und zu verschlüsseln.

Insgesamt sollen drei Umsetzungen eines Clients implementiert werden. Darunter fallen zwei Basisimplementierungen, die die Kommunikation über TCP und

UDP ermöglichen. Die dritte Implementation baut auf einer der Basisimplementierungen auf und verschlüsselt die entsprechende Kommunikation.

Für jeden Client besteht jeweils die Möglichkeit den Sender- und Empfängerteil in zwei separate Objekte aufzuteilen. Somit wird sichergestellt, dass Anwender die Kommunikation auf mehrere Threads aufteilen können.

Das Protokoll verwendet die implementierten Clients und sorgt für einen reibungslosen Übergang zwischen den verschiedenen Implementationen. Dies ist notwendig, da zunächst nur eine UDP-Kommunikation aufgebaut werden kann, da das TCP-Protokoll eine zeitliche Synchronisation benötigt. Diese Synchronisation kann nur über eine bereits existierende Verbindung erfolgen.

Um die Verbindung während des Aufbaus der TCP-Verbindung abzusichern ist der erste Schritt nachdem UDP-Verbindungsaufbau die Verschlüsselung der Verbindung. Anschließend erfolgt ein Versuch die Zeiten zu synchronisieren und eine TCP-Verbindung aufzubauen. Bei Erfolg wird die Kommunikation in eine verschlüsselte TCP-Verbindung überführt, andernfalls wird die bereits bestehende UDP-Verbindung weiter verwendet.

Benutzern der Schnittstelle ist offen gestellt welche Art der Kommunikation sie verwenden möchten. Das Protokoll kann zu jedem Zeitpunkt in einen aktiven Client überführt werden, über den dann entsprechender Datenaustausch möglich ist.

### **3.3 Datei-I/O**

architektur und so

## **4 Umsetzung**

### **4.1 Benutzeroberfläche**

umsetzung und so

## 4.2 Kommunikation

umsetzung und so

## 4.3 Datei-I/O

umsetzung und so

## 5 Reflexion

reflexion und so

## Literatur

- [1] Bundesamt für Sicherheit in der Informationstechnik, *Cloud: Risiken und Sicherheitstipps*. Adresse: [https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Cloud-Computing-Sicherheitstipps/Cloud-Risiken-und-Sicherheitstipps/cloud-risiken-und-sicherheitstipps\\_node.html](https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Cloud-Computing-Sicherheitstipps/Cloud-Risiken-und-Sicherheitstipps/cloud-risiken-und-sicherheitstipps_node.html).
- [2] Apple Inc., *So verwendest du AirDrop auf einem iPhone oder iPad*. Adresse: <https://support.apple.com/de-de/HT204144>.