

The background of the entire image is a light gray color, populated with a variety of dark gray file icons. These icons include document pages, music notes, video cameras, dollar signs, and percentage symbols, all scattered across the frame. The central text is positioned over this pattern.


rdrop.


A filesharing service by Philipp, Simon and Lars

rdrop.

Oberfläche

rdrop.

IPv4 Address
91.4.17.173 

IPv6 Address
2003:df:f736:4600:bcb2:7870:263d:3d29:62334 


Partner IP Address


Port
2000









Transfer

Connect →

rdrop.

 **Upload File**

 Close

	Super Secret File.docx 1.2 MB	Pending	 
	Other File.txt 8.2 MB	Pending	 
	Cool Song.mp3 4.4 MB	50%	

rdrop.

Aufbau



Figma

Design



Nextjs

Framework



Tauri

Applikation und Deployment

rdrop.

Was ist Tauri?

Installation

Self Updater

Windows, MacOS, Linux

App Signing

Native Notifikationen

Plugin System

Fenster Management

Rust Funktionen aufrufen

rdrop.

Druid

The fltk-rs crate

Tauri

iced

Warum Tauri?

relm

gtk-rs

Slint

egui

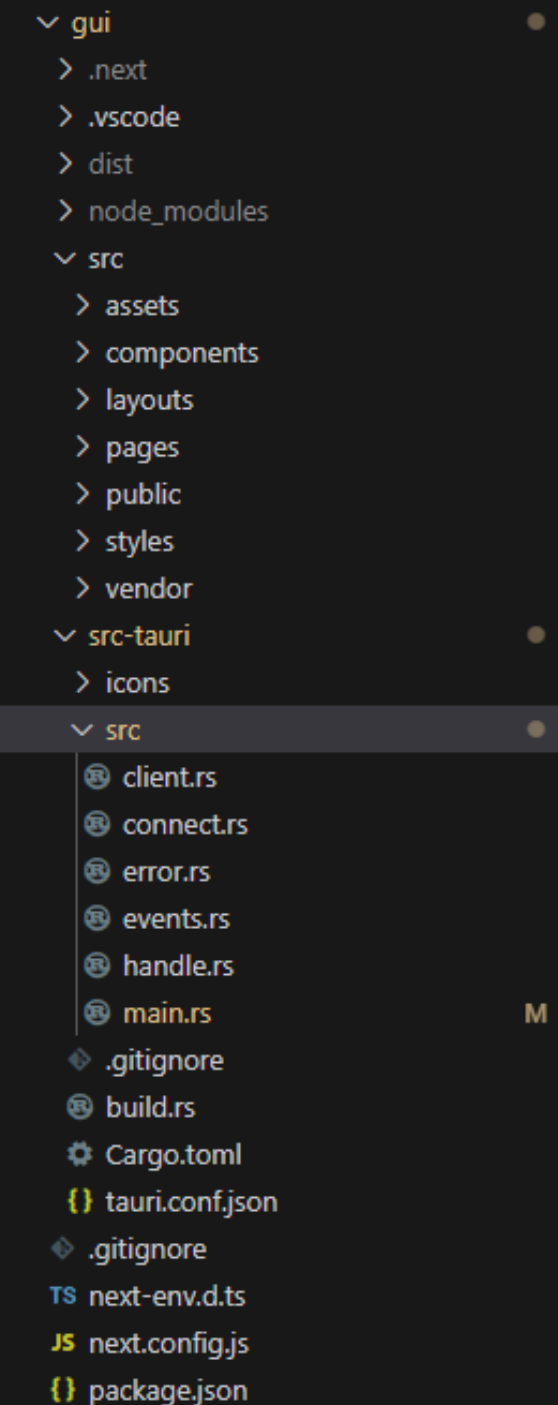
The Azul framework

rdrop.

Die Struktur

src – Frontend (HTML, CSS, JS)

src-tauri – Rust code, Window Handling



rdrop.

Rust im Frontend aufrufen

```
#[tauri::command]
pub fn offer_file(app_state: State<AppState>, path: String) -> Result<(), ClientError> {
    println!("[EVENT] offer_file");
    let mut unlocked_state: MutexGuard<Current> = (*app_state).0.lock()?;

    match unlocked_state.deref_mut() {
        &mut Current::ConnectedUdp(ref mut client: &mut Client<EncryptedWriter<...>, ...>) => client.offer_file(path),
        &mut Current::ConnectedTcp(ref mut client: &mut Client<EncryptedWriter<...>, ...>) => client.offer_file(path),
        _ => Err(ClientError::new(ClientErrorKind::WrongState)),
    }
}
```

```
const handleUpload = async () => {
    const selected = await open({
        multiple: true
    });

    if(!selected) return;
    selected.forEach((file) => {
        invoke('offer_file', { path: file });
    });
};
```

rdrop.

Events

```
useTauriEvent('tauri://file-drop', (event) => {  
  if (event.payload.length > 0) {  
    event.payload.forEach((file) => {  
      invoke('offer_file', { path: file });  
    });  
  }  
  
  setHover(false);  
});
```

update

blur

window-created

file-drop

file-drop-cancelled

close-requested

move

resize

theme-change

menu

scale-change

rdrop.

Events

```
useTauriEvent('app://update-status', (event) => {  
  |   setConncectionStatus(event?.payload);  
});
```

```
pub fn send_connect_status(handle: &AppHandle<Wry>, status: &str, description: &str) -> Result<(), ClientError> {  
  |   handle &AppHandle  
  |   |   .emit_all(event: "app://update-status", payload: Status { status: status.into(), description: description.into(), error: false })  
  |   |   ?;  
  |   |   Ok(())  
  }  
}
```

rdrop.

Kommunikation

Datenübertragung über das Internet

rdrop.

Verbindungsaufbau

1. Öffnen eines **UDP-Sockets**
2. In einem Intervall **Eröffnungspakete** senden
3. Bei Erhalt eines Eröffnungspakets wird die **Verbindung aktiv**

rdrop.

Verschlüsselung

1. **Rollen** festlegen durch Austausch von zufälligen Zahlen
2. Öffentliche **Schlüssel** austauschen

XChaCha20 stream cipher

Poly1305 message authentication

rdrop.

Übergang zu TCP

P2P Synchronisation

1. **Zeitstempel** erstellen
2. Paket **versenden**
3. **Warten** auf Antwort
4. Rountrip **Delay** berechnen
5. **Zeitdifferenz** Schätzung erstellen
6. **Median** der Schätzungen verwenden

NTP Synchronisation

1. Anfrage an **NTP Server** senden
2. **Zeitdifferenz** ermitteln

rdrop.

Übergang zu TCP

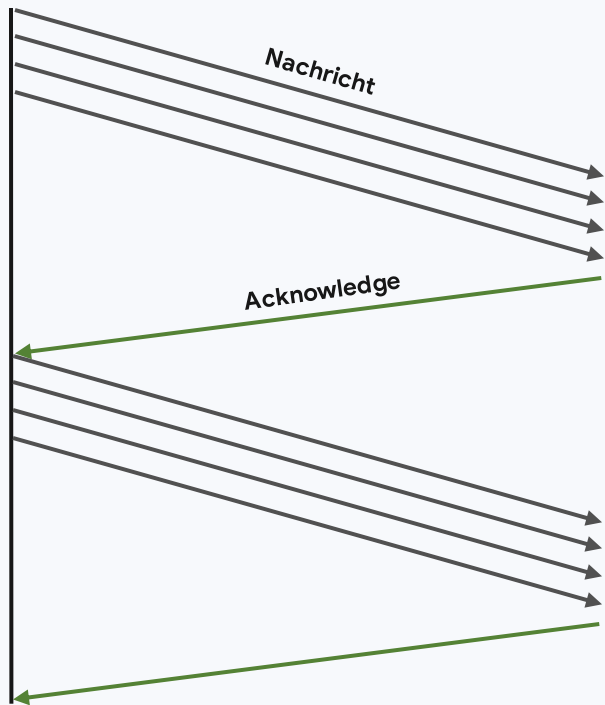
1. **Uhrzeit** für Verbindungsaufbau festlegen
2. Verbindungsversuch **starten**

Problematisch bei Asynchronem **Netzwerk-Jitter**

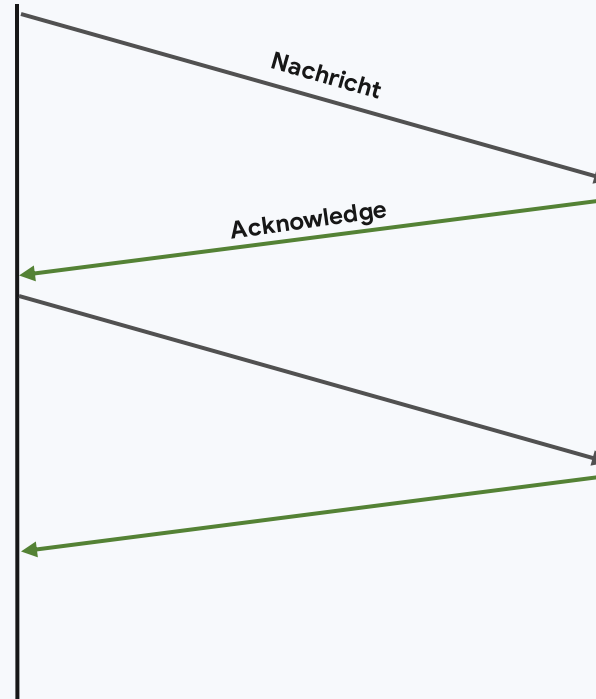
rdrop.

Protokolle auf Basis von UDP

Sliding-Window



Send-And-Wait



rdrop.

File I/O

Dateiaufspaltung, Hashing und Nachrichtenkodierung

rdrop.

I/O – Modul

- Erstellt und kodiert Nachrichten zwischen Sender und Empfänger
- Angebot / Bestellung Dateien
- Zerteilt Dateien in kleinere Pakete
- Fügt Pakete wieder zu einem File zusammen

rdrop.

I/O – Modul

- Bibliothek (lib)
- Modularisiert:
 - Error
 - Hash
 - Offer
 - Order
 - File
 - General

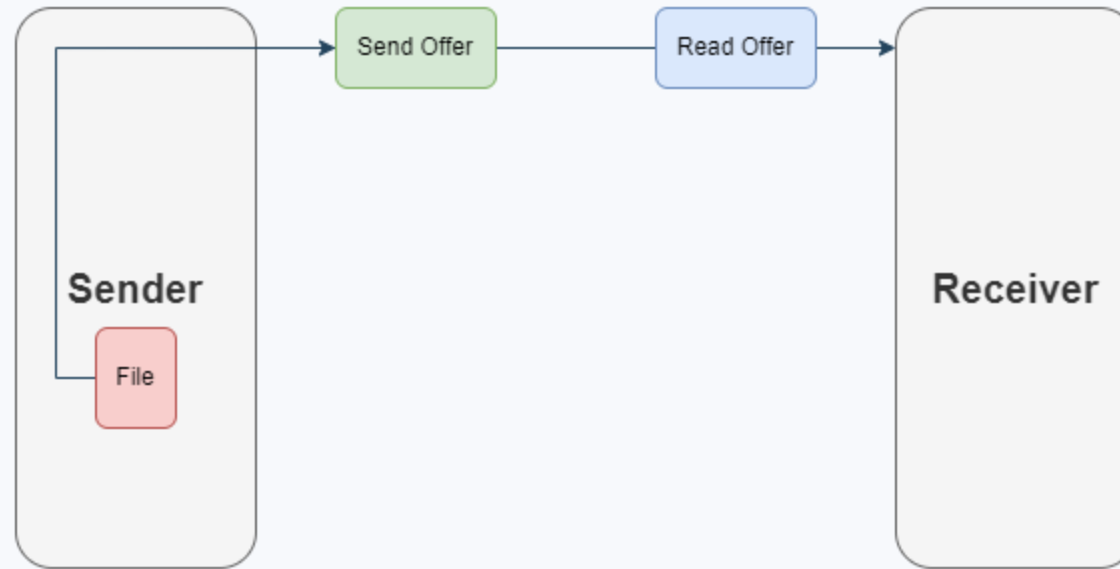
rdrop.

Vorgehensweise



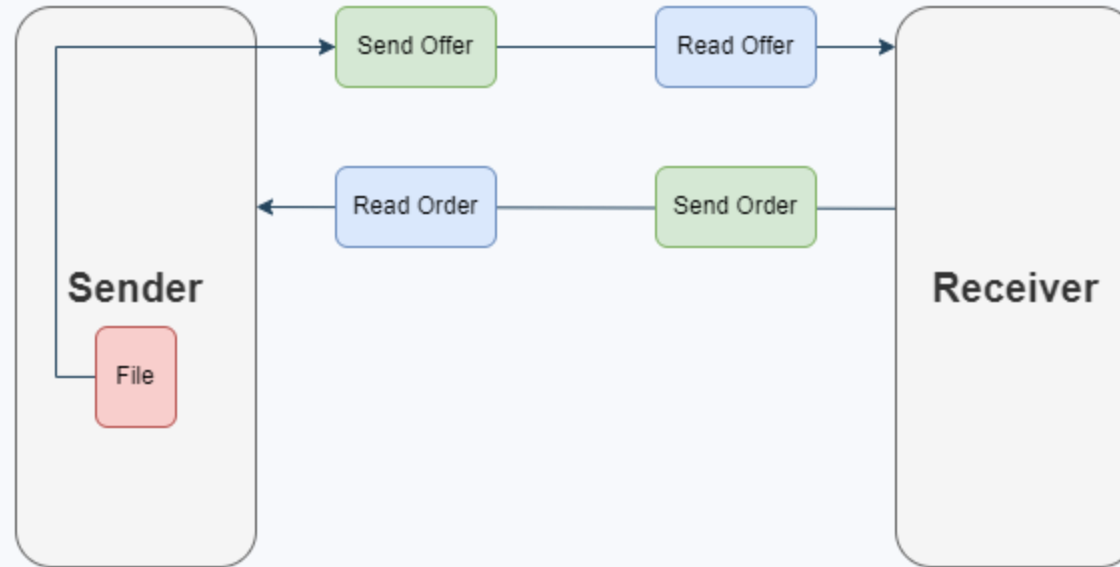
rdrop.

Vorgehensweise - Offer



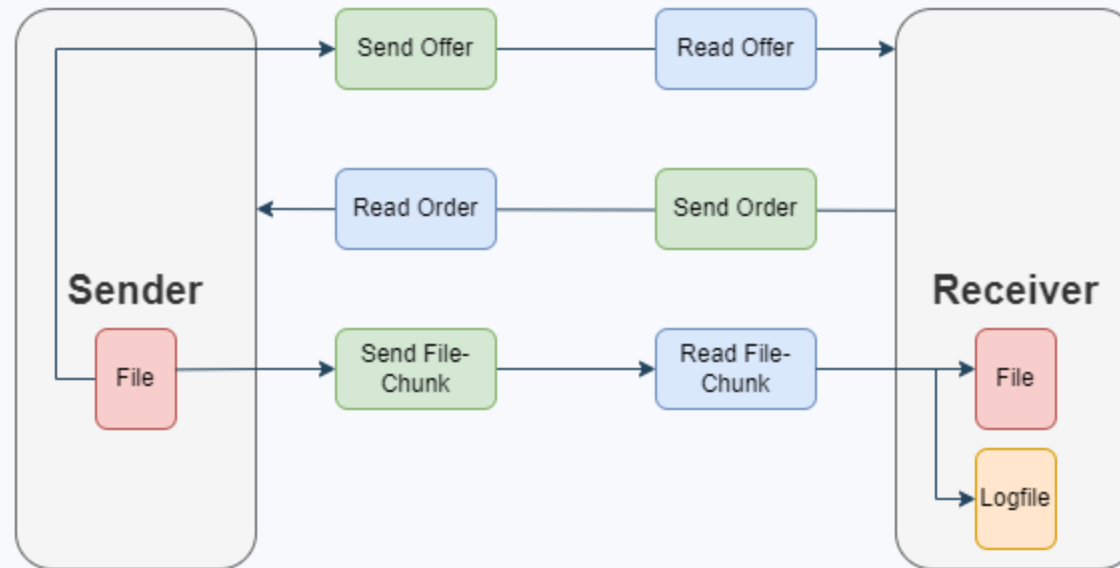
rdrop.

Vorgehensweise - Order



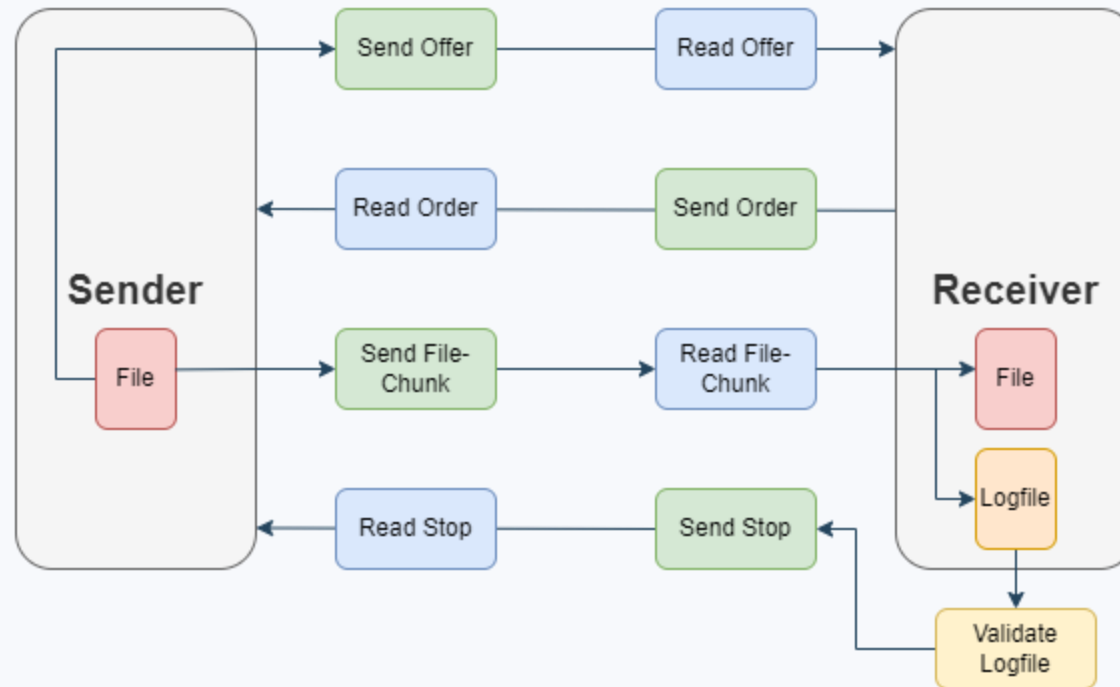
rdrop.

Vorgehensweise - File-Data



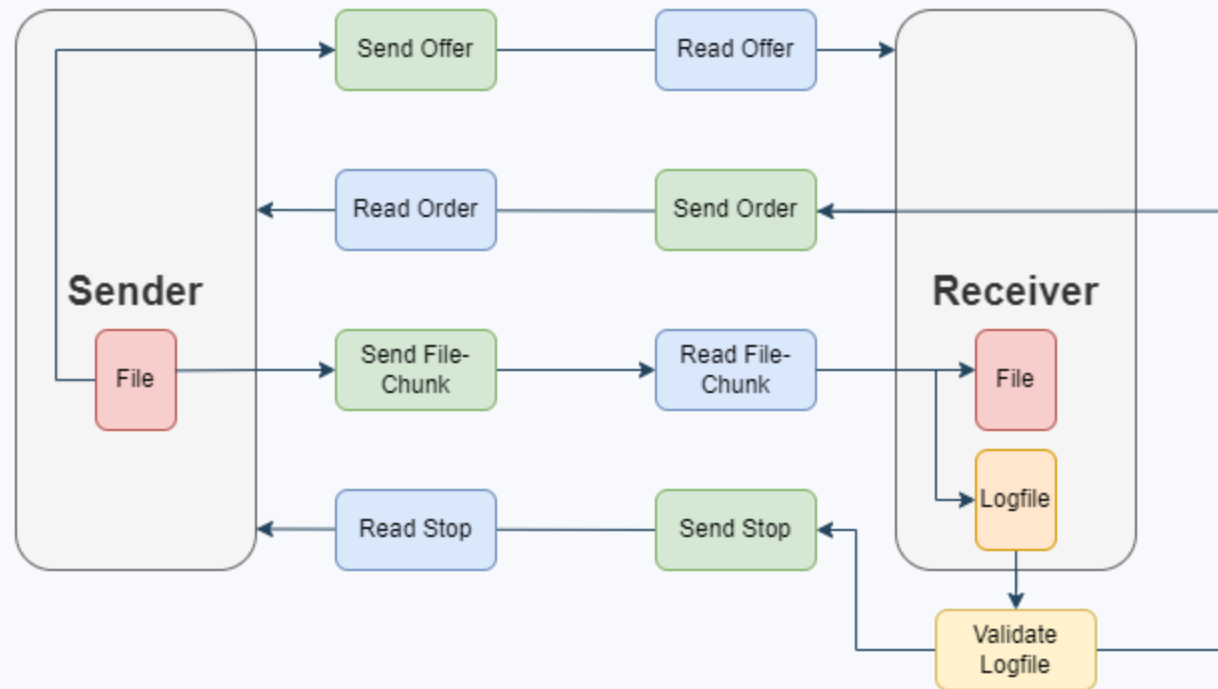
rdrop.

Vorgehensweise - Stop



rdrop.

Vorgehensweise - Reorder



rdrop.

Kodierung

- Vec<u8> (Byte-Vector)
- Write!-Makro
- Regular Expressions
- Offer
 - (1)[Name] - [Größe] - [Hash-Algorithmus] - [File-Hash]
 - \[(.+)\] - \[(\d+)\] - \[(SHA256|SHA512|MD5|SIPHASH24)\] - \([(0-9a-fA-F)+)\]
- Order
 - (2)[Chunk-Size] - [Hash-Algorithmus] - [File-Hash] - [Name] - [Start] - [End] - [Chunk-Hash-Algorithmus]
 - \[(\d+)\]\s\s\[(SHA256|SHA512|MD5|SIPHASH24)\]\s\s\[([a-zA-Z0-9]+)\]\s\s\[(.+)\]\s\s\[(\d+)\]\s\s\[(\d+)\]\s\s\[(SHA256|SHA512|MD5|SIPHASH24)\]\)?

rdrop.

Kodierung

- Daten
 - [Header][Datenpaket]
 - 1. Byte: 0
 - 2. Byte: Header-Länge
 - 3. Byte: Header-Feld-Größen: WXXY YZZZ
 - 4. Byte – max 151. Byte: Header-Informationen
- User-Hash, Paketgröße, File-Hash, Paketgesamtanzahl, Paketnummer, Pakethash

rdrop.

Kodierung

- Stop

- (3)[File-Hash]
- \([a-zA-Z0-9]+\)

- Logfile

- [Timestamp] - [UserHash] - [FileHash-Algorithmus] - [FileHash] - [Paketnummer] - [Maximale Paketnummer] - [Paketgröße bytes] - [PaketHash-Algorithmus] - [PaketHash]
- \((\d{2}\.\d{2}\.\d{4} \- \d{2}:\d{2}:\d{2}\.\d{3})\)[\t\f\v]*-\[\t\f\v]*\([a-zA-Z0-9]+\)[\t\f\v]*-\[\t\f\v]*\[(SHA256|SHA512|MD5|SIPHASH24)\)[\t\f\v]*-\[\t\f\v]*\([a-zA-Z0-9]+\)[\t\f\v]*-\[\t\f\v]*\((\d+)\)[\t\f\v]*-\[\t\f\v]*\((\d+)\)[\t\f\v]*-\[\t\f\v]*\((\d+) bytes\)[\t\f\v]*(-[\t\f\v]*\[(SHA256|SHA512|MD5|SIPHASH24)\)[\t\f\v]*-\[\t\f\v]*\([a-zA-Z0-9]+\))?

rdrop.

Ausblick - Weiterentwicklung

- Paketgröße variabel
- Buffersize variabel
- Hashalgorithmus variabel
- Userhash nicht nur Platzhalter