

Pflichtenheft: Software Challenge



Stand: 31. Mai 2009 17:09 Uhr
Version: 1.0a

Autoren:

Christian Wulf
Florian Fittkau
Marcel Jackwerth
Raphael Randschau

Inhaltsverzeichnis

1. Zielsetzung	1
1.1. Musskriterien	1
1.2. Sollkriterien	1
1.3. Kannkriterien	1
1.4. Abgrenzungskriterien	1
2. Produkteinsatz	2
2.1. Anwendungsbereich	2
2.2. Zielgruppen	2
2.3. Betriebsbedingungen	2
3. Produktumgebung	3
3.1. Software	3
3.2. Hardware	3
3.3. Orgware	3
4. Produktübersicht	4
5. Akteure	6
6. Produktfunktionen	7
6.1. Anwendungsfälle aus Benutzersicht	7
6.1.1. Veröffentlichen von lokalen Dateien	8
6.1.2. Alle veröffentlichten Daten eines Peers anzeigen	9
6.1.3. Eine Backup-Datei verteilen	10
6.1.4. Eine veröffentlichte Datei herunterladen	11
6.1.5. Speicherplatz für andere freigeben	12
6.1.6. Identifikation mit Benutzername und Passwort	13
6.1.7. Backup-Datei löschen	14
6.1.8. Veröffentlichte Dateien suchen	15
6.2. Anwendungsfälle aus Systemsicht	16
6.2.1. Verbindung herstellen	16
6.2.2. Verbindung trennen	17
6.2.3. Nachricht validieren	18
6.2.4. Peerliste aktualisieren	19
6.2.5. Verteilung überprüfen	20
6.2.6. Daten verschlüsseln	21
6.2.7. Daten entschlüsseln	22
6.2.8. Daten komprimieren	23
6.2.9. Daten dekomprimieren	24
6.2.10. Daten teilen	25
6.2.11. Daten zusammenfügen	26
6.2.12. Backup-Daten verteilen	27

7. Produktdaten	28
7.1. Peerliste und Nodes	28
7.2. Veröffentlichte Dateien	28
7.3. Sicherungsdateien	29
8. Produktleistungen	30
9. Benutzeroberfläche	31
9.1. Dialogstruktur	31
9.2. Bildschirmlayout	32
9.3. GUI-Entwurfsvorschläge	32
10. Protokoll	33
10.1. Beschreibung	33
10.1.1. status-Nachricht	33
10.1.2. get-Nachricht	33
10.1.3. store-Nachricht	34
10.1.4. find-Nachricht	34
10.1.5. meta-Nachricht	34
10.1.6. listBackups-Nachricht	34
10.1.7. deleteBackups-Nachricht	35
10.1.8. checkParts-Nachricht	35
10.2. Zulässige Nachrichtensequenzen	36
11. Qualitätsanforderung	38
12. Testszenarien	39
12.1. Share-Szenarien	39
12.2. Backup-Szenarien	39
13. Entwicklungsumgebung	40
13.1. Software	40
13.2. Hardware	40
13.3. Orgware	40
A. Anhang	41
A.1. XML Nachrichten-Schemata	41
Glossary	46
Acronyms	47

1. Zielsetzung

Software Challenge ist ein System, dass...

1.1. Musskriterien

- Muss

1.2. Sollkriterien

- xx

1.3. Kannkriterien

- xx

1.4. Abgrenzungskriterien

- xx

2. Produkteinsatz

2.1. Anwendungsbereich

Die hier beschriebene Software kann in allen Bereichen eingesetzt werden um von einem beliebigen Rechner aus Daten auf anderen Rechnern verteilt zu speichern. Ebenfalls ist es möglich von einem anderen Rechner aus, auf dem BaseTorrent installiert ist, auf diese Daten zuzugreifen und, sofern man möchte, anderen Teilnehmern vom BaseTorrentNetzwerk (BTN) zu gestatten auf diese zuzugreifen.

2.2. Zielgruppen

Hinsichtlich der Zielgruppe gibt es fast keine Einschränkungen für BaseTorrent. Da die GUI intuitiv zu gestalten ist, kann jeder die Software bedienen, der grundlegende Kenntnisse im Umgang mit Computerprogrammen besitzt. Der Benutzer muss lediglich dafür sorgen, dass sein PC während der Ausführung von BaseTorrent mit einem lokalen Netzwerk bzw. dem Internet verbunden ist, wenn er Daten in einem dieser Netzwerke speichern oder aus dem Netzwerk kopieren möchte.

Falls keine weiteren Sprachen integriert werden, ist das Verständnis der deutschen Sprache erforderlich.

2.3. Betriebsbedingungen

Durch die sorgfältige Planung des Systems ergeben sich folgende wichtige Kriterien:

- Betriebsdauer: unbegrenzt
- Das System ist wartungsfrei.
- Die Verbindung mit dem Internet/Lokalen Netzwerk darf während der Ausführung nicht getrennt werden.

3. Produktumgebung

3.1. Software

- Java Runtime Environment (mindestens Version 1.6 Update 13),
- beliebiges Betriebssystem, das Java unterstützt.

3.2. Hardware

- Rechner mit Netzwerkverbindung.

3.3. Orgware

- Gewährleistung der Internetanbindung bzw. des LAN-Zugangs.

4. Produktübersicht

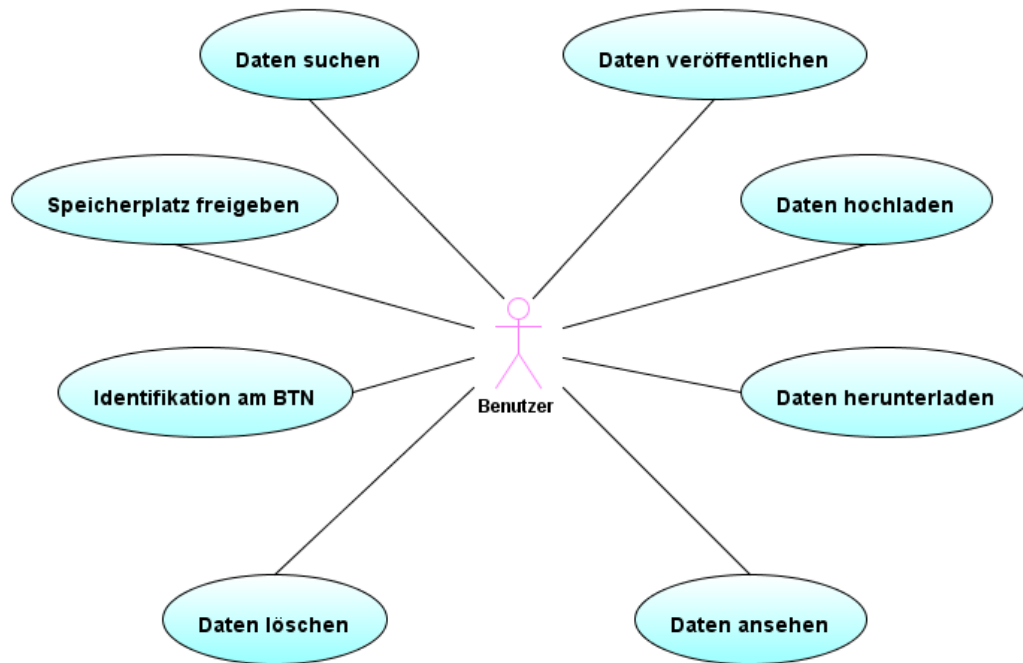


Abbildung 1: Übersicht über die Anwendungsfälle aus der Benutzersicht (s. auch Kapitel 6.1).

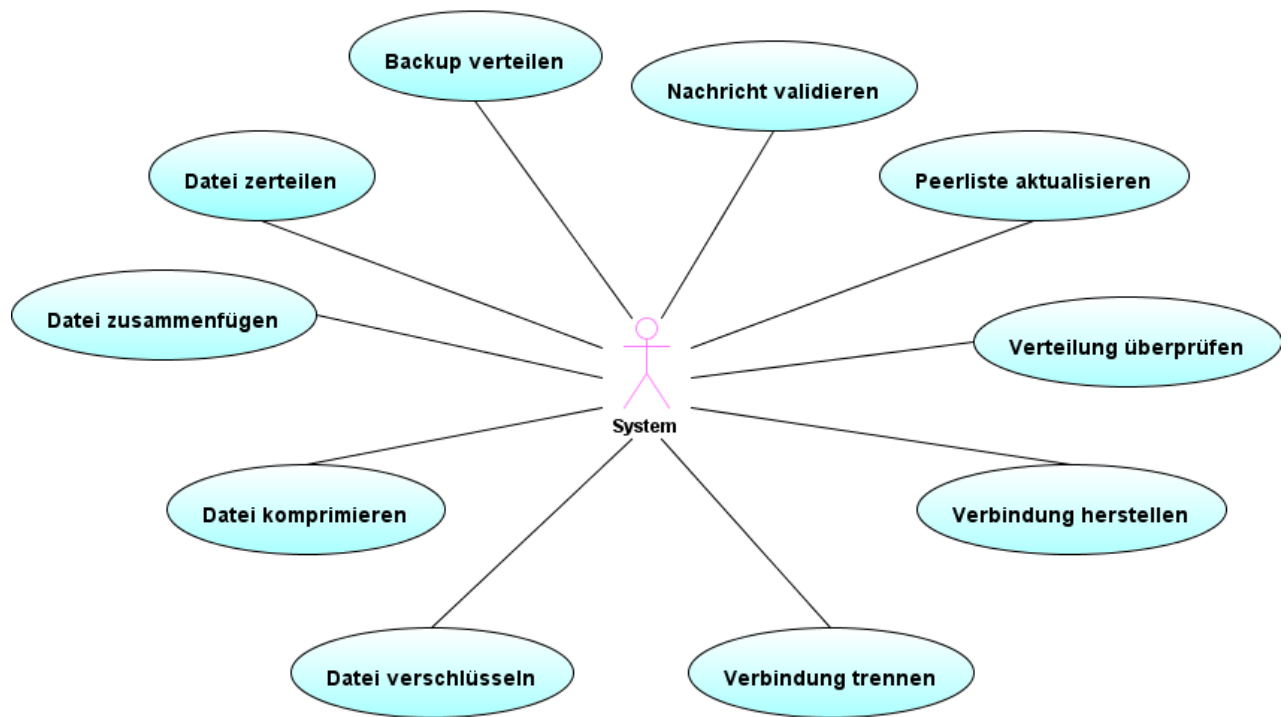


Abbildung 2: Übersicht über die Anwendungsfälle aus der Systemsicht (s. auch Kapitel 6.2).

5. Akteure

Die nachfolgenden Anwendungsfälle beziehen sich auf den folgenden Akteur:

Akteur	Beschreibung	Zugehörige Anwendungsszenarios
Benutzer	Empfängt Backup-Dateiteile, kann Dateien veröffentlichen, verschickt eine Liste von Dateien, kann Dateien löschen und suchen und lädt Dateien hoch.	<ul style="list-style-type: none"> • Eine eigene, lokale Datei veröffentlichen • Alle veröffentlichten Daten eines Peers anzeigen • Eine Backup-Datei verteilen • Eine veröffentlichte Datei eines Peers herunterladen • Speicherplatz für andere freigeben • Identifikation mit Benutzernamen und Passwort • Eine eigene Backup-Datei aus dem BTN löschen • Eine veröffentlichte Datei anhand von Stichwörtern suchen
System	Führt die Aktionen des Benutzers auf unterster Ebene aus, und verwaltet Nachrichten von anderen Peers und verarbeitet Daten .	<ul style="list-style-type: none"> • Verbindung herstellen • Verbindung trennen • Nachricht validieren • Peerliste aktualisieren • Verteilung überprüfen • Daten verschlüsseln • Daten entschlüsseln • Daten komprimieren • Daten dekomprimieren • Daten teilen • Daten zusammenfügen • Backup-Daten verteilen

6. Produktfunktionen

6.1. Anwendungsfälle aus Benutzersicht

Nachfolgend werden die folgenden Anwendungsfälle detailliert beschrieben:

- Eine eigene, lokale Datei veröffentlichen (Alternativ: mehrere Dateien)
- Alle veröffentlichten Daten eines Peers anzeigen (Alternativ: alle Peers)
- Eine Backup-Datei verteilen (Alternativ: mehrere Backup-Dateien)
- Eine veröffentlichte Datei eines Peers herunterladen (Alternativ: Backup mit Passwort)
- Speicherplatz für andere freigeben
- Identifikation mit Benutzername und Passwort
- Eine eigene Backup-Datei aus dem BTN löschen
- Eine veröffentlichte Datei anhand von Stichwörtern suchen

6.1.1. Veröffentlichen von lokalen Dateien

Use Case Nummer	V-1
Use Case Name	Eigene, lokale Datei veröffentlichen
Initiierender Akteur	Benutzer
Weitere Akteure	-
Kurzbeschreibung	Der Benutzer wählt eine eigene, lokale Datei aus und markiert sie als „veröffentlicht“.
Vorbedingung	Die BaseTorrentApplikation (BTA) ist mit dem BTN verbunden.
Nachbedingung	Die Datei wurde veröffentlicht und kann von allen Peers angesehen werden.
Funktionalität des Use Cases	Ablauf: <ol style="list-style-type: none"> 1. Der Benutzer wählt die eigene, lokale Datei aus 2. Der Benutzer markiert die Datei als „veröffentlicht“
Alternativen	zu 1) Der Benutzer wählt mehrere Dateien/Ordner aus zu 2) Abbruch der Auswahl
Ausnahmen	-
Benutzte Use Cases	-

6.1.2. Alle veröffentlichten Daten eines Peers anzeigen

Use Case Nummer	V-2
Use Case Name	Alle Veröffentlichten Daten eines Peers anzeigen
Initiierender Akteur	Benutzer
Weitere Akteure	-
Kurzbeschreibung	Der Benutzer lässt sich alle veröffentlichten Daten eines Peers anzeigen
Vorbedingung	Die BTA ist mit dem BTN verbunden.
Nachbedingung	Die BTA zeigt alle Daten eines ausgewählten Peers an.
Funktionalität des Use Cases	<p>Ablauf:</p> <ol style="list-style-type: none"> 1. Der Benutzer wählt einen Peer aus der aktuellen Peerliste aus. 2. Der Benutzer wählt „Alle veröffentlichten Daten anzeigen“ aus. 3. Die BTA zeigt alle Daten des ausgewählten Peers an.
Alternativen	<p>zu 1) Der Benutzer wählt mehrere Dateien/Ordner aus.</p> <p>zu 2) Abbruch der Auswahl.</p>
Ausnahmen	-
Benutzte Use Cases	-

6.1.3. Eine Backup-Datei verteilen

Use Case Nummer	V-3
Use Case Name	Eine Backup-Datei verteilen
Initiierender Akteur	Benutzer
Weitere Akteure	-
Kurzbeschreibung	Der Benutzer wählt eine eigene, lokale Datei aus und markiert sie als „gesichert“.
Vorbedingung	Die BTA ist mit dem BTN verbunden.
Nachbedingung	Die Datei wurde gesichert und kann nur von dem Benutzer angesehen werden.
Funktionalität des Use Cases	Ablauf: <ol style="list-style-type: none"> 1. Der Benutzer wählt die eigene, lokale Datei aus. 2. Der Benutzer markiert die Datei als „gesichert“.
Alternativen	zu 1) Der Benutzer wählt mehrere Dateien/Ordner aus. zu 2) Abbruch der Auswahl.
Ausnahmen	-
Benutzte Use Cases	-

6.1.4. Eine veröffentlichte Datei herunterladen

Use Case Nummer	V-4
Use Case Name	Eine veröffentlichte Datei herunterladen
Initiierender Akteur	Benutzer
Weitere Akteure	-
Kurzbeschreibung	Der Benutzer lädt eine veröffentlichte Datei eines Peers herunter.
Vorbedingung	Die BTA ist mit dem BTN verbunden. Die Liste der öffentlichen Dateien eines Peers wird angezeigt.
Nachbedingung	Die BTA zeigt die heruntergeladene Datei an.
Funktionalität des Use Cases	<p>Ablauf:</p> <ol style="list-style-type: none"> 1. Der Benutzer wählt eine veröffentlichte Datei eines Peers aus der Liste. 2. Der Benutzer lädt diese Datei herunter.
Alternativen	<p>zu 1) Der Benutzer lädt eine von ihm gesicherte Datei herunter, nachdem das richtige Passwort eingegeben wurde.</p>
Ausnahmen	-
Benutzte Use Cases	-

6.1.5. Speicherplatz für andere freigeben

Use Case Nummer	V-5
Use Case Name	Speicherplatz für andere freigeben
Initiierender Akteur	Benutzer
Weitere Akteure	-
Kurzbeschreibung	Der Benutzer wählt einen eigenen, lokalen Ordner und stellt die maximale Größe ein.
Vorbedingung	
Nachbedingung	Ein eigener Ordner ist für die empfangenen Backup-Dateien anderer mit einer maximalen Größe bereit.
Funktionalität des Use Cases	Ablauf: 1. Der Benutzer wählt den eigenen, lokalen Ordner aus. 2. Der Benutzer legt eine maximale Größe fest.
Alternativen	zu 2) Abbruch der Auswahl.
Ausnahmen	-
Benutzte Use Cases	-

6.1.6. Identifikation mit Benutzernamen und Passwort

Use Case Nummer	V-6
Use Case Name	Identifikation mit Benutzernamen und Passwort
Initiierender Akteur	Benutzer
Weitere Akteure	-
Kurzbeschreibung	Der Benutzer identifiziert sich mit einem Benutzernamen und Passwort.
Vorbedingung	Die BTA ist mit dem BTN verbunden.
Nachbedingung	Der Benutzer ist identifiziert.
Funktionalität des Use Cases	Ablauf: 1. Der Benutzer gibt Benutzernamen und Passwort ein.
Alternativen	zu 1) Es erfolgt keine Eingabe seitens des Benutzers.
Ausnahmen	-
Benutzte Use Cases	-

6.1.7. Backup-Datei löschen

Use Case Nummer	V-7
Use Case Name	Eigene Backup-Datei löschen
Initiierender Akteur	Benutzer
Weitere Akteure	-
Kurzbeschreibung	Der Benutzer löscht eine seiner Backup-Dateien aus dem BTN.
Vorbedingung	Die BTA ist mit dem BTN verbunden.
Nachbedingung	Die eigene Backup-Datei wurde aus dem BTN gelöscht.
Funktionalität des Use Cases	<p>Ablauf:</p> <ol style="list-style-type: none"> 1. Der Benutzer lässt sich die Liste seiner Backup-Dateien anzeigen. 2. Der Benutzer markiert eine Datei aus der Liste. 3. Der Benutzer klickt auf „Backup-Datei löschen“.
Alternativen	zu 2) Der Benutzer bricht ab.
Ausnahmen	-
Benutzte Use Cases	-

6.1.8. Veröffentlichte Dateien suchen

Use Case Nummer	V-8
Use Case Name	Veröffentlichte Dateien anhand von Stichwörtern suchen
Initiierender Akteur	Benutzer
Weitere Akteure	-
Kurzbeschreibung	Der Benutzer gibt Stichwörter ein und sucht damit nach veröffentlichten Dateien.
Vorbedingung	Die BTA ist mit dem BTN verbunden.
Nachbedingung	Die Liste der veröffentlichten Dateien, die zu den Stichwörtern passen, wird angezeigt.
Funktionalität des Use Cases	Ablauf: 1. Der Benutzer gibt Stichwörter ein. 2. Die BTA zeigt alle veröffentlichten Daten mit den eingegebenen Stichwörtern an.
Alternativen	-
Ausnahmen	-
Benutzte Use Cases	-

6.2. Anwendungsfälle aus Systemsicht

Nachfolgend werden die folgenden Anwendungsfälle detailliert beschrieben:

- Verbindung herstellen
- Verbindung trennen
- Nachricht validieren
- Peerliste aktualisieren
- Verteilung überprüfen
- Daten verschlüsseln
- Daten entschlüsseln
- Daten komprimieren
- Daten dekomprimieren
- Daten teilen
- Daten zusammenfügen
- Backup-Daten verteilen

6.2.1. Verbindung herstellen

Use Case Nummer	S-1
Use Case Name	Eine Verbindung mit dem BTN herstellen
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System stellt eine Verbindung zum BTN her.
Vorbedingung	Es besteht keine Verbindung zum BTN . Es ist zu mindestens die Adresse von einem Peer im BTN bekannt.
Nachbedingung	Es besteht eine Verbindung zum BTN .
Funktionalität des Use Cases	Ablauf: 1. Das System verbindet sich mit dem bekannten Peer .
Alternativen	-
Ausnahmen	Es konnte keine Verbindung hergestellt werden.
Benutzte Use Cases	-

6.2.2. Verbindung trennen

Use Case Nummer	S-2
Use Case Name	Die Verbindung mit dem BTN trennen
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System trennt die Verbindung zum BTN.
Vorbedingung	Es besteht eine Verbindung zum BTN.
Nachbedingung	Es besteht keine Verbindung zum BTN.
Funktionalität des Use Cases	Ablauf: 1. Das System sendet, empfängt und antwortet auf keine Nachrichten mehr.
Alternativen	-
Ausnahmen	-
Benutzte Use Cases	-

6.2.3. Nachricht validieren

Use Case Nummer	S-3
Use Case Name	Eine eingehende Nachricht validieren
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System prüft, ob eine eingehende Nachricht im korrekten Format vorliegt.
Vorbedingung	Das System kennt eine Spezifikation, anhand der es die Korrektheit einer Nachricht nachweisen kann.
Nachbedingung	Die Nachricht wurde als korrekt oder nicht korrekt identifiziert.
Funktionalität des Use Cases	<p>Ablauf:</p> <ol style="list-style-type: none"> 1. Das System prüft, ob die eingehende XML-Nachricht einem vorgegebenen XML-Schema entspricht. 2. Die Nachricht wird als korrekt identifiziert.
Alternativen	zu 2) die Nachricht wurde als nicht korrekt identifiziert.
Ausnahmen	-
Benutzte Use Cases	-

6.2.4. Peerliste aktualisieren

Use Case Nummer	S-4
Use Case Name	Peerliste aktualisieren
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System prüft, welche Nodes in der Peerliste online sind und welche offline sind.
Vorbedingung	Es besteht eine Verbindung zum BTN
Nachbedingung	Das System hält eine aktuelle Peerliste.
Funktionalität des Use Cases	<p>Ablauf:</p> <ol style="list-style-type: none"> 1. Jede Stunde verschickt das System seine Peerliste an alle Peers. 2. Das System kriegt Peerlisten von jedem Peer zurück und setzt das Online-Flag dieses Peers auf true. 3. Das System fügt die fremden Informationen in seine Peerliste ein.
Alternativen	<p>zu 2) das System kriegt von einem Knoten keine Peerliste und setzt das Online-Flag des Peers in der Peerliste auf false.</p>
Ausnahmen	-
Benutzte Use Cases	-

6.2.5. Verteilung überprüfen

Use Case Nummer	S-5
Use Case Name	Die Verteilung von Dateiteilen überprüfen
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System prüft, ob die Dateiteile ausreichend im Netz verteilt sind.
Vorbedingung	Das System hat Informationen über die Dateiteile und eine aktuelle Peerliste. Anforderungen an die Anzahl der Verteilung der Dateiteile.
Nachbedingung	Die Verteilung wurde als ausreichend oder nicht ausreichend gekennzeichnet.
Funktionalität des Use Cases	<p>Ablauf:</p> <ol style="list-style-type: none"> 1. Das System schickt an alle Peers eine Anfrage, ob sie den Dateiteil x haben. 2. Das System überprüft, ob die Anzahl der Dateiteile x im System den Anforderungen genügt. 3. Das System bewertet die Verteilung als ausreichend.
Alternativen	zu 3) das System bewertet die Verteilung als nicht ausreichend.
Ausnahmen	-
Benutzte Use Cases	-

6.2.6. Daten verschlüsseln

Use Case Nummer	S-6
Use Case Name	Daten mit einem Passwort verschlüsseln
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System verschlüsselt die Daten mit einem Passwort
Vorbedingung	Daten, die verschlüsselt werden soll. Das System besitzt Zugriffsrechte für die Daten. Es ist genügend Speicherplatz vorhanden.
Nachbedingung	Die Daten sind verschlüsselt.
Funktionalität des Use Cases	Ablauf: <ol style="list-style-type: none"> 1. Das System liest die Daten ein. 2. Das System verschlüsselt die Daten mit Hilfe eines Verschlüsselungsalgorithmus. 3. Das System legt die verschlüsselten Daten ab.
Alternativen	-
Ausnahmen	Es ist nicht genügend Speicherplatz vorhanden.
Benutzte Use Cases	-

6.2.7. Daten entschlüsseln

Use Case Nummer	S-7
Use Case Name	Daten mit einem Passwort entschlüsseln
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System entschlüsselt die Daten mit einem Passwort.
Vorbedingung	Daten, die entschlüsselt werden soll und verschlüsselt sind. Das System besitzt Zugriffsrechte für die Daten. Es ist genügend Speicherplatz vorhanden. Ein Passwort ist vorhanden.
Nachbedingung	Die Daten sind entschlüsselt.
Funktionalität des Use Cases	<p>Ablauf:</p> <ol style="list-style-type: none"> 1. Das System liest die Daten ein. 2. Das System entschlüsselt die Daten mit Hilfe eines Verschlüsselungsalgorithmus und dem Passwort. 3. Das System legt die entschlüsselten Daten ab.
Alternativen	zu 3) die Daten wurden nicht erfolgreich entschlüsselt, weil das Passwort falsch war.
Ausnahmen	Es ist nicht genügend Speicherplatz vorhanden.
Benutzte Use Cases	-

6.2.8. Daten komprimieren

Use Case Nummer	S-8
Use Case Name	Daten komprimieren
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System komprimiert die Daten.
Vorbedingung	Daten, die komprimiert werden soll. Das System besitzt Zugriffsrechte für die Daten. Es ist genügend Speicherplatz vorhanden.
Nachbedingung	Die Daten sind komprimiert.
Funktionalität des Use Cases	Ablauf: <ol style="list-style-type: none"> 1. Das System liest die Daten ein. 2. Das System komprimiert die Daten mit Hilfe eines Komprimierungsalgorithmuses. 3. Das System legt die komprimierten Daten ab.
Alternativen	-
Ausnahmen	Es ist nicht genügend Speicherplatz vorhanden.
Benutzte Use Cases	-

6.2.9. Daten dekomprimieren

Use Case Nummer	S-9
Use Case Name	Daten dekomprimieren
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System dekomprimiert die Daten.
Vorbedingung	Daten, die dekomprimiert sind. Das System besitzt Zugriffsrechte für die Daten. Es ist genügend Speicherplatz vorhanden.
Nachbedingung	Die Daten sind dekomprimiert.
Funktionalität des Use Cases	Ablauf: <ol style="list-style-type: none"> 1. Das System liest die Daten ein. 2. Das System dekomprimiert die Daten. 3. Das System legt die dekomprimierten Daten ab.
Alternativen	-
Ausnahmen	Es ist nicht genügend Speicherplatz vorhanden.
Benutzte Use Cases	-

6.2.10. Daten teilen

Use Case Nummer	S-10
Use Case Name	Daten teilen in mehrere kleinere Dateiteile
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System teilt die Daten in mehrere kleinere Dateiteile.
Vorbedingung	Daten, die geteilt werden sollen. Das System besitzt Zugriffsrechte für die Daten. Es ist genügend Speicherplatz vorhanden.
Nachbedingung	Die Daten sind geteilt in mehrere kleinere Dateiteile.
Funktionalität des Use Cases	<p>Ablauf:</p> <ol style="list-style-type: none"> 1. Das System komprimiert die Daten und teilt dabei die Daten in mehrere kleinere Dateiteile. 2. Das System legt die Dateiteile ab.
Alternativen	-
Ausnahmen	Es ist nicht genügend Speicherplatz vorhanden.
Benutzte Use Cases	-

6.2.11. Daten zusammenfügen

Use Case Nummer	S-11
Use Case Name	Dateiteile zusammenfügen zu den originalen Daten
Initiierender Akteur	System
Weitere Akteure	-
Kurzbeschreibung	Das System fügt die Dateiteile zu den originalen Daten zusammen.
Vorbedingung	Dateiteile, die zusammengefügt werden sollen und alle Dateiteile zu den originalen Daten . Das System besitzt Zugriffsrechte für die Dateiteile. Es ist genügend Speicherplatz vorhanden.
Nachbedingung	Die originalen Daten stehen wieder zur Verfügung.
Funktionalität des Use Cases	<p>Ablauf:</p> <ol style="list-style-type: none"> 1. Das System dekomprimiert die Daten aus den Dateiteilen. 2. Das System legt die Daten ab.
Alternativen	-
Ausnahmen	Es ist nicht genügend Speicherplatz vorhanden.
Benutzte Use Cases	-

6.2.12. Backup-Daten verteilen

Use Case Nummer	S-12
Use Case Name	Backup-Daten verteilen
Initiierender Akteur	System
Weitere Akteure	Peer
Kurzbeschreibung	Das System verteilt eine Backup-Datei redundant an mehrere Peers.
Vorbedingung	Es besteht eine Verbindung zum BTN, die zu verteilenden Daten sind komprimiert, verschlüsselt und aufgeteilt worden und die zugehörigen Meta-Daten sind verschlüsselt worden.
Nachbedingung	Die Backup-Daten wurden vollständig gemäß der Redundanzqualität verteilt.
Funktionalität des Use Cases	<p>Ablauf:</p> <ol style="list-style-type: none"> 1. Das System öffnet einen TCP-Port. 2. Das System sendet solange sukzessiv Speicher-Anfragen (10.1.3) (mit dem geöffneten TCP-Port) pro BaseTorrent file part (BTFP) von den zu verteilenden Backup-Daten gemäß der Redundanzqualität an verschiedene Peers aus der eigenen Peerliste, bis alle BTFP erfolgreich versandt wurden. 3. Jeder Peer, der einen BTFP speichern kann, lädt einen solchen über den TCP-Port herunter. 4. Bei kompletter, erfolgreicher Übertragung schließt das System den geöffneten TCP-Port.
Alternativen	-
Ausnahmen	<p>zu 2.) Das System kann keine Verbindung zum Peer herstellen: Das Online-Flag (6.2.4) wird auf false gesetzt und die Speicher-Anfrage wird an einen anderer Peer gesendet.</p> <p>zu 2.) Das System konnte nicht alle BTFP gemäß der Redundanzqualität versenden: Das System benachrichtigt Benutzer.</p>
Benutzte Use Cases	-

7. Produktdaten

Es werden nachfolgend alle Daten und Datenstrukturen angegeben, die die BTA speichert und benutzt:

7.1. Peerliste und Nodes

Die BTA hält eine Liste (die sogenannte Peerliste) mit allen Nodes im BTN, wobei jede Node mit folgenden Informationen gespeichert wird:

- IP-Adresse
- UDP-Port
- Zeitstempel der letzten Aktivität des Nodes
- Gesamter Speicherplatz
- Freier Speicherplatz
- Online-Flag 6.2.4

Die empfangenen Dateiteile werden lokal mit einem zusätzlichen Informationsteil abgelegt, der wie folgt aussieht:

7.2. Veröffentlichte Dateien

- Hash über die Meta-Daten und die Daten
- Meta-Daten:
 - Titel
 - Schlüsselwörter
 - Benutzername des Autors
 - Teil x von y
 - Datum und Uhrzeit der letzten Änderung
 - Gesamt-Hash

7.3. Sicherungsdateien

- Benutzername des Autors
- Benutzername des Autors (verschlüsselt)
- Hash über die verschlüsselten Meta-Daten und die verschlüsselten Daten
- Meta-Daten (verschlüsselt):
 - Titel
 - Teil x von y
 - Datum und Uhrzeit der letzten Änderung
 - Gesamt-Hash
 - Part-Passwort

8. Produktleistungen

Ausfallsicherheit: Nach der manuellen Eingabe eines Peers versucht die BaseTorrentApplikation unmittelbar den Peer zu kontaktieren und bei Erfolg die Nodes des Peers abzufragen, um bei Ausfall des initialen Peers weiterhin am BaseTorrent-Netzwerk teilnehmen zu können.

Korrekte Dateiübertragung: Das korrekte Übertragen eines Dateiteils wird durch TCP/IP übernommen. Hierdurch ist eine komplette und korrekte Übertragung eines Dateiteils bereits sichergestellt.

Verschlüsselung: Eine Sicherungsdatei wird verschlüsselt im BTN verteilt. So wird erreicht, dass kein Unbefugter Zugriff auf die Daten hat.

Komprimierung: Die Dateien und die Zusatzinformationen zu diesen werden komprimiert und danach erst übertragen. Auf diese Weise ist eine möglichst ressourcenschonende Belastung des Netzwerkes gewährleistet.

Aufteilen der Dateien: Durch das Aufteilen von einer Datei in mehrere kleinere Dateiteile erreicht das Produkt eine höhere Geschwindigkeit bei der Verteilung der Dateien. Da ein Peer dadurch von mehreren Peers gleichzeitig die gleiche Datei aber andere Dateiteile herunterladen kann. Diese Funktion bietet auch eine effizientere Möglichkeit der Verteilung der Sicherungsdateien.

9. Benutzeroberfläche

In diesem Kapitel wird die Benutzeroberfläche der BaseTorrentApplikation sowie ein entsprechender Entwurfsvorschlag vorgestellt.

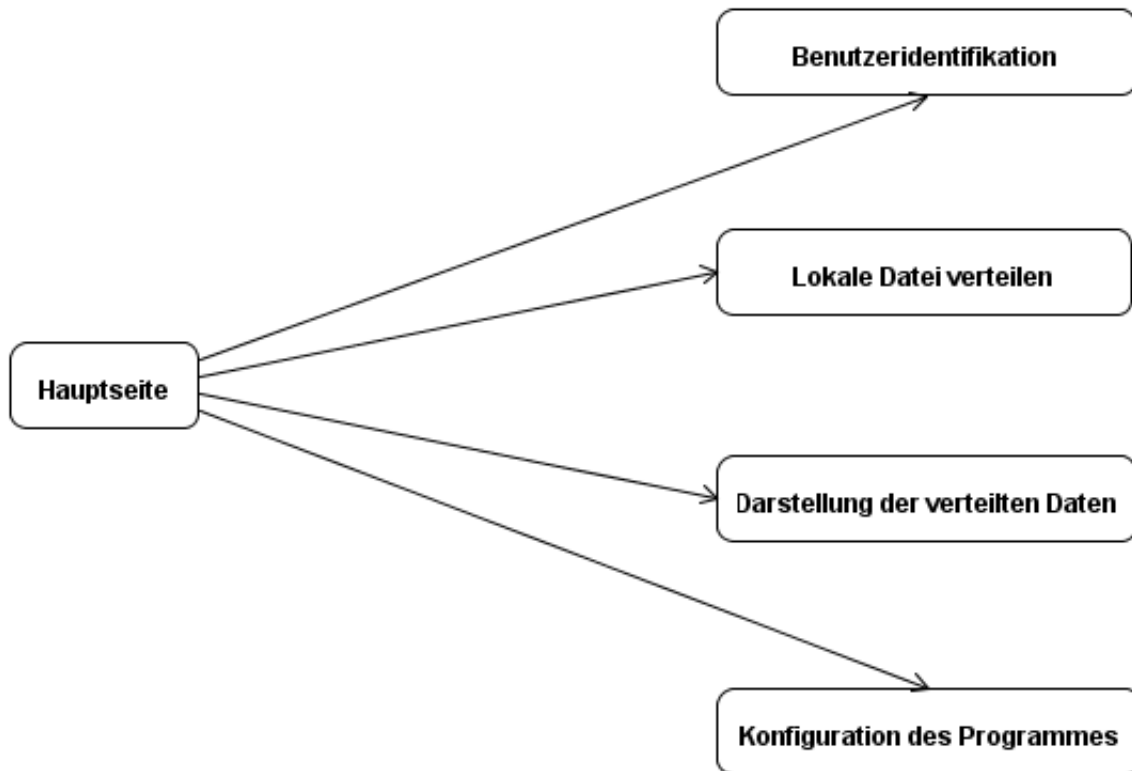


Abbildung 3: Die Benutzeroberfläche (schematisch).

9.1. Dialogstruktur

Für die Benutzerführung ist die nachfolgende Dialogstruktur vorgesehen, die sich aus den nachfolgenden Komponenten zusammensetzt:

- 1.) Einem Dialog zum Anmelden an das System. Eine Abmeldung ist nicht vorgesehen. Eingabe ist die Benutzerkennung sowie ein Kennwort.
- 2.) Eine Dialogbox wird verwendet für die Konfiguration von BaseTorrent, d.h. der Eingabe eines Verzeichnisses, der beim Ablegen von verteilten Dateien und Ordnern verwendet wird. Zudem wird eine maximale Speichergröße als auch ein zusätzliches Verzeichnis hinterlegt. Beides wird herangezogen um anderen Benutzern auf dem vorliegenden Rechner Speicherkapazität zum Verteilen von Dateien im Netzwerk bereitzustellen.

- 3.) Die Darstellung von bereits veröffentlichten Dateien und Ordnern sowie, sofern der Benutzer sich gemäß 1.) angemeldet hat, auch der benutzerbezogenen Daten. Diese können in zwei separaten, wenn auch weitestgehend identisch aufgebauten Komponenten dargestellt werden.
- 4.) Eine Dialogbox zeigt den Fortschritt des Speicherns von verteilten Daten (der sogenannte *Download*) an.
- 5.) Eine weitere Dialogbox zeigt ein Eingabefeld an, um nach Dateien mit einem bestimmten Schlüsselwort zu suchen.

9.2. Bildschirmlayout

Um eine intuitive Benutzerführung zu gewährleisten wird ein Layout entsprechend der Programmierrichtlinien für die Gestaltung der Benutzeroberflächen von Desktop-Applikationen angestrebt¹. Dies impliziert insbesondere eine einheitliche Bezeichnung von Schaltflächen und Menüpunkten. Eine Umsetzung mittels Java Swing wird angestrebt.

9.3. GUI-Entwurfsvorschläge

Nachfolgend stellen wir einen Entwurf der Benutzeroberfläche dar.

¹Etwas *Microsoft Inductive User Interface Guidelines*, <http://msdn.microsoft.com/en-us/library/ms997506.aspx>.

10. Protokoll

10.1. Beschreibung

Das Protokoll von BaseTorrent basiert auf Nachrichtenübermittlung mit UDP und Datenübermittlung mit TCP.

Bei Verlust von UDP-Nachrichten wird nach einer einstellbaren Wartezeit erneut gesendet.

Die Suchnachrichten werden gestaffelt versendet, d.h. aus der Liste der Nodes werden der Reihe nach eine gewisse Anzahl an Nodes angeschrieben, um das Netzwerk nicht unnötig zu belasten.

10.1.1. status-Nachricht

Beschreibung:	Informiert die bekannten Peers aus der eigenen Peerliste, dass man noch mit dem BTN verbunden ist.
Netzwerkprotokoll:	UDP
Parameter:	aktuelle Peerliste
Verhalten Peer:	<ol style="list-style-type: none"> 1. Prüfe die Liste auf Aktualität 2. Sende die eigene aktualisierte Peerliste

10.1.2. get-Nachricht

Beschreibung:	Daten-BTFP-Anfrage an Peer X, um Daten von X auf den BTA-Computer zu kopieren.
Netzwerkprotokoll:	UDP
Parameter:	BTFP-Hash, ein eigener TCP-Port
Verhalten Peer:	<ol style="list-style-type: none"> 1. Lädt die Daten, wenn sie existieren, über den TCP-Port hoch

10.1.3. store-Nachricht

Beschreibung:	Frage an, wer die eigenen Daten aufnehmen kann.
Netzwerkprotokoll:	UDP
Parameter:	Part-Hash, Metadaten, TCP-Port, Dateigröße
Verhalten Peer:	<ol style="list-style-type: none"> 1. Prüfe auf Vorhandensein 2. Prüfe freien Speicherplatz 3. Wenn beides vorhanden, dann Datei herunterladen

10.1.4. find-Nachricht

Beschreibung:	Anfrage an Peer, ob er Datei x hat.
Netzwerkprotokoll:	UDP
Parameter:	Liste mit Schlüsselwörtern
Verhalten Peer:	<ol style="list-style-type: none"> 1. Prüfe auf Vorhandensein der Datei. 2. Wenn Datei vorhanden, dann antwortet mit den Metadaten

10.1.5. meta-Nachricht

Beschreibung:	Antwort, ob Peer eine Datei x hat.
Netzwerkprotokoll:	UDP
Parameter:	Metadaten
Verhalten Peer:	<ol style="list-style-type: none"> 1. Antwortet mit den Metadaten

10.1.6. listBackups-Nachricht

Beschreibung:	Anfrage ob Peer BackUps vom Autor y hat.
Netzwerkprotokoll:	UDP
Parameter:	Autor
Verhalten Peer:	<ol style="list-style-type: none"> 1. Wenn Datei vorhanden, dann "Meta Antwort"

10.1.7. deleteBackups-Nachricht

Beschreibung:	Löschen von Backups.
Netzwerkprotokoll:	UDP
Parameter:	BTFTP-Hash, Part-Passwort
Verhalten Peer:	<ol style="list-style-type: none"> 1. Prüfe auf Vorhandensein der Datei. 2. Prüfe auf Übereinstimmung von Autor und durch Part-Passwort entschlüsseltem Autor 3. Wenn Bedingungen erfüllt, dann Datei löschen

10.1.8. checkParts-Nachricht

Beschreibung:	Fragt nach, wie oft ein gewisser Part noch im Netz vorhanden ist.
Netzwerkprotokoll:	UDP
Parameter:	Liste von Part-Hashes
Verhalten Peer:	<ol style="list-style-type: none"> 1. Prüfe auf Vorhandensein der Datei 2. Wenn Datei vorhanden, dann "Meta Antwort"

10.2. Zulässige Nachrichtensequenzen

Die folgenden Sequenzdiagramme zeigen die zulässigen interessanten Nachrichtensequenzen auf (bei den anderen ist dies trivial):

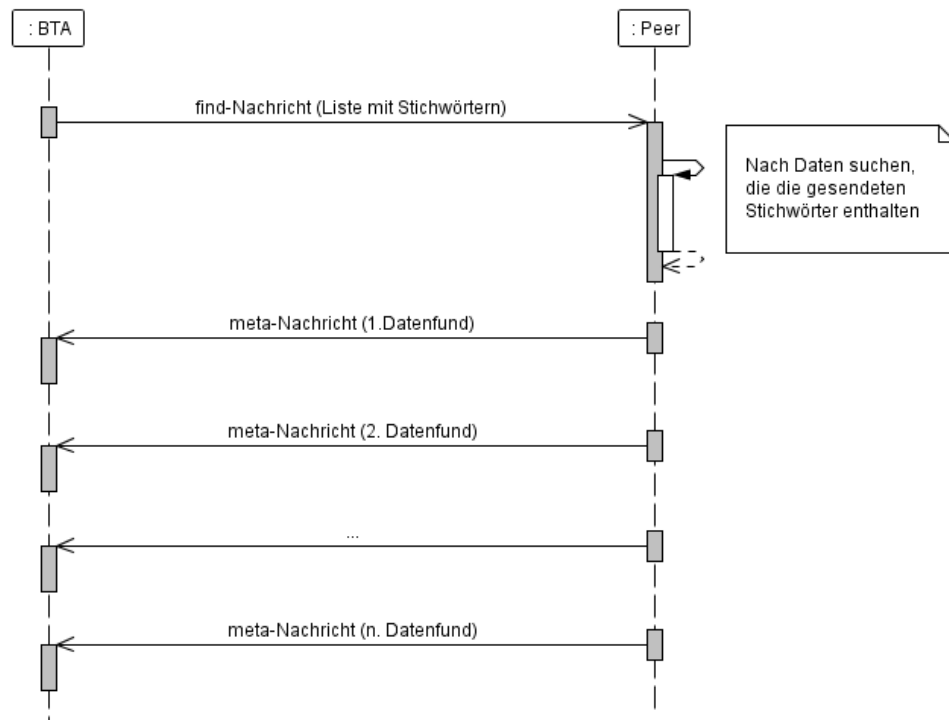


Abbildung 4: Beispiel für eine zulässige Suchanfrage

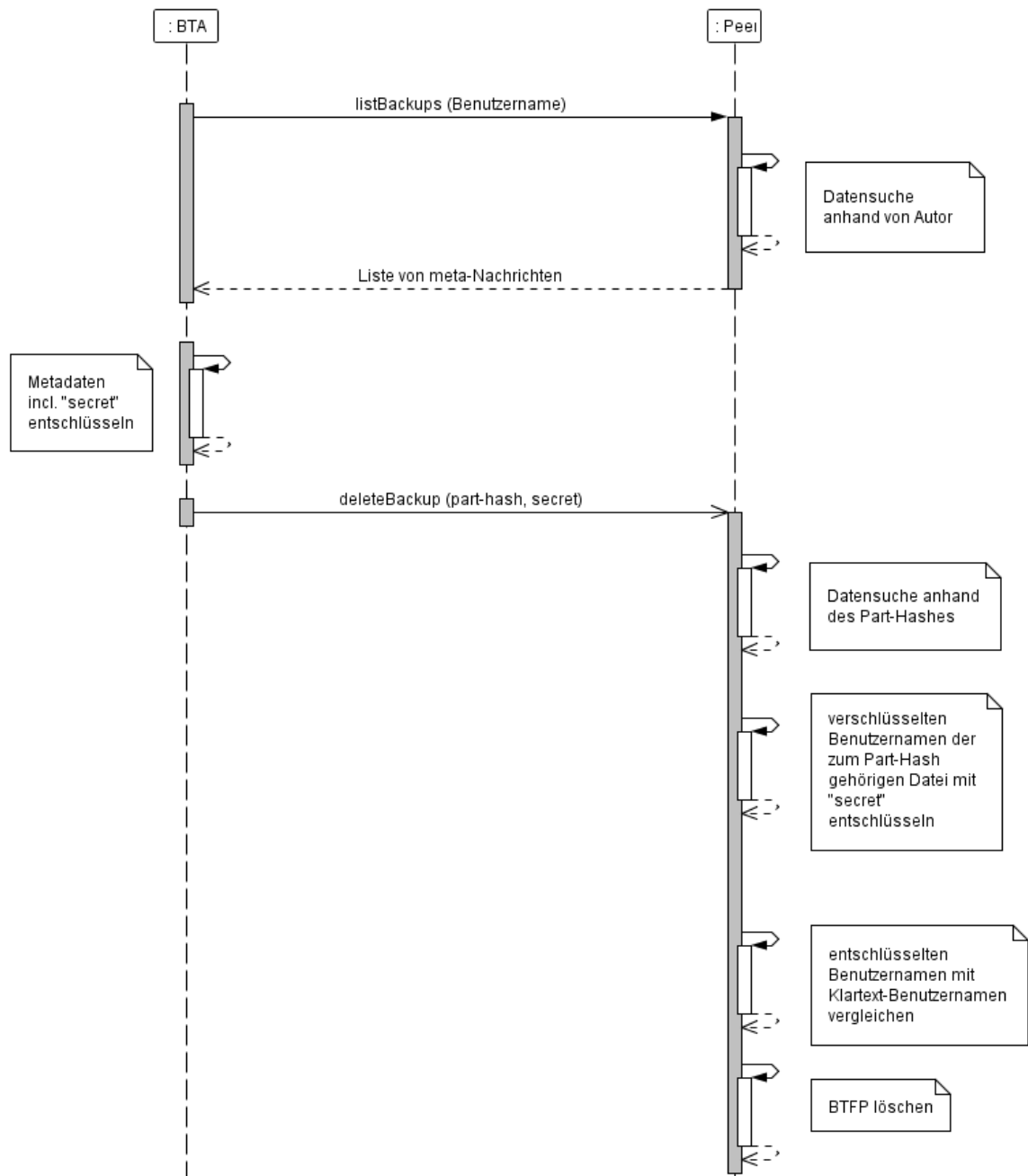


Abbildung 5: Beispiel für eine zulässige Löschanfrage

11. Qualitätsanforderung

	sehr wichtig	wichtig	weniger wichtig	unwichtig
Robustheit	x			
Zuverlässigkeit	x			
Benutzerfreundlichkeit		x		
Effizienz		x		
Portierbarkeit		x		
Kompatibilität		x		

12. Testszzenarien

Die folgenden Szenarien werden besonders ausgiebig getestet, um optimale Produktqualität zu erreichen:

12.1. Share-Szenarien

- Benutzer X veröffentlicht eine Datei z, anschließend holt sich Benutzer Y die Veröffentlichungsliste von Benutzer X und kontrolliert, dass die neue Datei z vorhanden ist (*erfolgreiche Veröffentlichung*).
- Benutzer X entfernt eine veröffentlichte Datei z aus seiner Veröffentlichungsliste, ruft anschließend seine eigene Veröffentlichungsliste ab und kontrolliert, dass die Datei z nicht in der Liste angezeigt wird (*korrekte Share-Löschung*).

12.2. Backup-Szenarien

- Der Benutzer verteilt eine Backup-Datei, ruft anschließend seine eigenen Backup-Dateien ab und kontrolliert, ob die neue Backup-Dateien (häufig genug) im BTN vorhanden ist (*erfolgreiche Backup-Verteilung*).
- Der Benutzer löscht eine Backup-Datei, ruft anschließend seine eigenen Backup-Dateien ab und kontrolliert, dass die gelöschte Datei nicht mehr im BTN vorhanden ist (*erfolgreiche Backup-Löschung*).
- Der Benutzer ruft seine eigenen Backup-Dateien ab und schlägt beim Versuch, eine seiner Backup-Dateien mit einem falschen Passwort zu löschen, fehl (*Backup-Sicherheit*).
- Benutzer X verteilt eine Backup-Datei z (u.a. an Benutzer Y), Benutzer Y trennt sich vom BTN, Benutzer X löscht lokal die Backup-Datei z, ruft anschließend seine eigenen Backup-Dateien ab und kontrolliert, ob die neue Backup-Datei z weiterhin vollständig im BTN verfügbar ist (*Ausfallsicherheit*).
- Der Benutzer verteilt zwei Backup-Dateien mit demselben Namen, ruft anschließend seine eigenen Backup-Dateien ab und kontrolliert, ob die beiden neuen Backup-Dateien im BTN verfügbar sind (*korrekte Backup-Verteilung*).
- Der Benutzer verteilt eine Backup-Datei z. Benutzer Y ruft die veröffentlichten Dateien von Benutzer X ab und kontrolliert, dass die Backup-Datei z von Benutzer X nicht angezeigt wird (*Backup-Sichtbarkeit*).

13. Entwicklungsumgebung

13.1. Software

- Plattform
 - Java,
 - Subversion.
- Werkzeuge
 - Eclipse,
 - OpenOffice,
 - Adobe Acrobat Reader,
 - L^AT_EX

13.2. Hardware

- Mindestens vier Personal Computer mit Internetzugang.

13.3. Orgware

A. Anhang

A.1. XML Nachrichten-Schemata

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <xs:schema elementFormDefault="qualified" version="1.0"
3   targetNamespace="http://se.informatik.uni-kiel.de/BASETorrent"
4   xmlns:tns="http://se.informatik.uni-kiel.de/BASETorrent"
5   xmlns:xs="http://www.w3.org/2001/XMLSchema">
6
7   <xs:element name="deleteBackup" type="tns:deleteBackupMessage" />
8
9   <xs:element name="find" type="tns:findMessage" />
10
11  <xs:element name="get" type="tns:getMessage" />
12
13  <xs:element name="listBackups" type="tns:listBackupsMessage" />
14
15  <xs:element name="meta" type="tns:metaMessage" />
16
17  <xs:element name="status" type="tns:statusMessage" />
18
19  <xs:element name="store" type="tns:storeMessage" />
20
21  <!-- vllt reicht auch ein Element fuer Beides, so ists uebersichtlicher -->
22  <xs:element name="checkPartsReq" type="tns:checkParts" />
23  <xs:element name="checkPartsResp" type="tns:checkParts" />
24
25  <!-- einziges nicht-Nachricht-Wurzelement -->
26  <!-- zu ueberschlsselnde und in encryptedMeta einzupackende Daten -->
27  <xs:element name="backupMeta" type="tns:backupMeta" />
28
29  <!-- Wrapper üfr Nachrichten, z.B. falls timestamping/TTL/etc ügewünscht -->
30  <xs:complexType name="message" abstract="true">
31    <xs:sequence>
32      <xs:element name="node" type="tns:node" />
33      <xs:element name="reqId" type="xs:string" />
34    </xs:sequence>
35  </xs:complexType>
36
37  <!-- Alles was mit einem konkreten Part zu tun hat -->
38  <xs:complexType name="partMessage" abstract="true">
39    <xs:complexContent>
40      <xs:extension base="tns:message">
41        <xs:sequence>
42          <xs:element name="partHash" type="xs:base64Binary" />
43        </xs:sequence>
44      </xs:extension>
45    </xs:complexContent>
46  </xs:complexType>
47
48  <!-- Part runterladen -> ServerSocket an tcpPort offen -->

```

```

49  <xs:complexType name="getMessage">
50    <xs:complexContent>
51      <xs:extension base="tns:partMessage">
52        <xs:sequence>
53          <xs:element name="tcpPort" type="tns:portType" />
54        </xs:sequence>
55      </xs:extension>
56    </xs:complexContent>
57  </xs:complexType>
58
59  <!-- Part hochladen, sowohl Backup als auch Share -->
60  <xs:complexType name="storeMessage">
61    <xs:complexContent>
62      <xs:extension base="tns:partMessage">
63        <xs:sequence>
64          <xs:choice>
65            <xs:element name="encryptedMeta"
66              type="tns:encryptedMeta" />
67            <xs:element name="shareMeta"
68              type="tns:shareMeta" />
69          </xs:choice>
70          <xs:element name="tcpPort" type="tns:portType" />
71          <xs:element name="size" type="xs:unsignedInt" />
72        </xs:sequence>
73      </xs:extension>
74    </xs:complexContent>
75  </xs:complexType>
76
77  <!-- öLschen eines Backupparts -> öVerffentlichen des Challenge-Secrets -->
78  <xs:complexType name="deleteBackupMessage">
79    <xs:complexContent>
80      <xs:extension base="tns:partMessage">
81        <xs:sequence>
82          <xs:element name="secret" type="xs:string" />
83        </xs:sequence>
84      </xs:extension>
85    </xs:complexContent>
86  </xs:complexType>
87
88  <!-- Parts finden per üöSchlsselwrter -->
89  <xs:complexType name="findMessage">
90    <xs:complexContent>
91      <xs:extension base="tns:message">
92        <xs:sequence>
93          <xs:element name="keywords" type="tns:keywordsType" />
94        </xs:sequence>
95      </xs:extension>
96    </xs:complexContent>
97  </xs:complexType>
98
99  <!-- Eigene Backups auflisten -->
100  <xs:complexType name="listBackupsMessage">

```

```

101     <xs:complexContent>
102       <xs:extension base="tns:message">
103         <xs:sequence>
104           <xs:element name="author" type="xs:string" />
105         </xs:sequence>
106       </xs:extension>
107     </xs:complexContent>
108   </xs:complexType>
109
110   <!-- Antwort auf Partinfoanfragen mit Metadaten -->
111   <xs:complexType name="metaMessage">
112     <xs:complexContent>
113       <xs:extension base="tns:partMessage">
114         <xs:sequence>
115           <xs:choice>
116             <xs:element name="encryptedMeta"
117               type="tns:encryptedMeta" />
118             <xs:element name="shareMeta"
119               type="tns:shareMeta" />
120           </xs:choice>
121         </xs:sequence>
122       </xs:extension>
123     </xs:complexContent>
124   </xs:complexType>
125
126   <!-- Alle Parts haben Author im Klartext -->
127   <xs:complexType name="metaData" abstract="true">
128     <xs:sequence>
129       <xs:element name="author" type="xs:string" />
130     </xs:sequence>
131   </xs:complexType>
132
133   <!-- Alles, was eine Datei beschreibt -->
134   <xs:complexType name="fileMeta">
135     <xs:complexContent>
136       <xs:extension base="tns:metaData">
137         <xs:sequence>
138           <xs:element name="title" type="xs:string" />
139           <xs:element name="partNumber"
140             type="xs:positiveInteger" />
141           <xs:element name="partCount"
142             type="xs:positiveInteger" />
143           <xs:element name="timeStamp" type="xs:dateTime" />
144           <xs:element name="fileHash" type="xs:base64Binary" />
145         </xs:sequence>
146       </xs:extension>
147     </xs:complexContent>
148   </xs:complexType>
149
150   <!-- Infos über überffentlichte Parts -->
151   <xs:complexType name="shareMeta">
152     <xs:complexContent>

```

```

153     <xs:extension base="tns:fileMeta">
154         <xs:sequence>
155             <xs:element name="keywords" type="tns:keywordsType" />
156         </xs:sequence>
157     </xs:extension>
158 </xs:complexContent>
159 </xs:complexType>
160
161 <!-- Infos über Backups -->
162 <xs:complexType name="backupMeta">
163     <xs:complexContent>
164         <xs:extension base="tns:fileMeta">
165             <xs:sequence>
166                 <xs:element name="secret" type="xs:string" />
167             </xs:sequence>
168         </xs:extension>
169     </xs:complexContent>
170 </xs:complexType>
171
172 <!-- Alles üverschlüsselt -->
173 <xs:complexType name="encryptedMeta">
174     <xs:complexContent>
175         <xs:extension base="tns:metaData">
176             <xs:sequence>
177                 <!-- delete-Challenge -->
178                 <xs:element name="encAuthor" type="xs:base64Binary" />
179                 <!-- eigentliche Metadaten -->
180                 <xs:element name="encMeta" type="xs:base64Binary" />
181             </xs:sequence>
182         </xs:extension>
183     </xs:complexContent>
184 </xs:complexType>
185
186 <xs:simpleType name="keywordsType">
187     <xs:list itemType="xs:string" />
188 </xs:simpleType>
189
190 <!-- Hello , Listupdate , Ping/Pong -->
191 <xs:complexType name="statusMessage">
192     <xs:complexContent>
193         <xs:extension base="tns:message">
194             <xs:sequence>
195                 <xs:element name="nodeList">
196                     <xs:complexType>
197                         <xs:sequence>
198                             <xs:element name="node"
199                                 type="tns:nodeInfo" minOccurs="0" maxOccurs="unbounded" />
200                         </xs:sequence>
201                     </xs:complexType>
202                 </xs:element>
203             </xs:sequence>
204         </xs:extension>

```

```

205     </xs:complexContent>
206 </xs:complexType>
207
208 <!-- Nodeidentifikation -->
209 <xs:complexType name="node">
210     <xs:sequence>
211         <xs:element name="ip" type="xs:string" />
212         <xs:element name="port" type="tns:portType" />
213     </xs:sequence>
214 </xs:complexType>
215
216 <!-- voller Nodedatensatz -->
217 <xs:complexType name="nodeInfo">
218     <xs:complexContent>
219         <xs:extension base="tns:node">
220             <xs:sequence>
221                 <xs:element name="freeMb" type="xs:int" />
222                 <xs:element name="totalMb" type="xs:int" />
223                 <xs:element name="lastActivity" type="xs:int" /> <!-- Sekunden seit letztem K
224                 <!-- eher UNIX-TS? sonst üsste ja im Sekundentakt aktualisiert werden... -->
225             </xs:sequence>
226         </xs:extension>
227     </xs:complexContent>
228 </xs:complexType>
229
230 <xs:simpleType name="portType">
231     <xs:restriction base="xs:positiveInteger">
232         <xs:maxInclusive value="65535" />
233     </xs:restriction>
234 </xs:simpleType>
235
236 <!-- Sender polled seine Liste von Parts, Empfaenger antwortet mit Schnittmenge -->
237 <xs:complexType name="checkParts">
238     <xs:complexContent>
239         <xs:extension base="tns:message">
240             <xs:sequence>
241                 <xs:element name="partList">
242                     <xs:complexType>
243                         <xs:sequence>
244                             <xs:element name="partHash"
245                                 type="xs:base64Binary" minOccurs="1" maxOccurs="unbounded" />
246                         </xs:sequence>
247                     </xs:complexType>
248                 </xs:element>
249             </xs:sequence>
250         </xs:extension>
251     </xs:complexContent>
252 </xs:complexType>
253
254
255 </xs:schema>

```


Glossar

Backup

Sicherungskopie einer Datei oder eines ganzen Ordners.

BaseTorrent file

Eine Datei (oder ein Ordner), die auf irgendeine Art und Weise im BTN bekannt ist.

BaseTorrent file part

Ein *Daten*-Teil bzw. Stück einer BTF.

BaseTorrentApplikation

Das Programm, welches den Zugang zum BTN ermöglicht und Benutzereingaben verarbeitet.

BaseTorrentNetzwerk

Peer2Peer Netzwerk aus Rechnern in verschiedenen Netzwerken.

Daten

Dateien sowie ganze Ordner, die wiederum Ordner und Dateien enthalten können.

Hash

Eine Zahlen und Buchstabenfolge, die aus einer Datei zur Identifikation und Validierung gebildet wird.

IP

Eine Adresse, die zur Identifikation im Netzwerk verwendet wird.

Meta-Daten

Informationsdaten über bestimmte *Daten*, z.B. "Teil x von y" oder der Dateiname.

Node

Im Allgemeinen eine andere Bezeichnung für einen Peer, die in den Vordergrund stellt, dass andere Peers über diesen Peer erreichbar sind.

Orgware

Rahmenbedingungen bei IT-Projekten, die nicht unter Hardware oder Software fallen.

Part-Password

Bei *Backups* wird neben dem Autorennamen im Klartext dieser zusätzlich nochmal in verschlüsselter Form gespeichert, um den Ersteller eindeutig zu identifizieren. Dieses Passwort wird Part-Password genannt und ist Teil der verschlüsselten *Meta-Daten*, aus denen nur der Inhaber das Passwort rekonstruieren kann.

Peer

Ein Rechner im Netzwerk, der sowohl als Client als auch als Server fungiert.

Peerliste

Die Liste von allen Nodes im BTN.

Port

Ein Tor, um den richtigen Dienst auf dem Zielrechner anzusprechen.

Redundanzqualität

Die Redundanzqualität k ist der Grad bzw. die Anzahl der Verteilungen eines BTFP im BTN. $k = \min(\lceil 3 \cdot \log |Peerliste| \rceil, |Peerliste|)$, d.h. ein fester Wert, damit alle Benutzer gleichberechtigt verteilen.

TCP

Ein Protokoll zur Nachrichtenübertragung in einem Netzwerk; speziell zur Übertragung von großen Datenmengen geeignet.

Tracker

Ein Server-Programm, dass den Clients Informationen über die im BaseTorrent-Netzwerk verteilten Dateien liefert, insbesondere den Aufenthaltsort der einzelnen Teile einer konkreten Datei.

UDP

Ein Protokoll zur Nachrichtenübertragung in einem Netzwerk.

XML

Eine spezielles Dateiformat mit `<tag></tag>` Struktur.

Acronyms

BTA

BaseTorrentApplikation.

BTf

BaseTorrent file.

BTfP

BaseTorrent file part.

BTN

BaseTorrentNetzwerk.