Programmierung eines

Pseudo-Torrentnetzwerkes

Rechnerkommunikation und Middleware

Entwurfsarbeit der Studiengänge I-16/ I-17

Fachbereich Automatisierung und Informatik

Hochschule Harz

**Autoren:** Dirk Neumann (24160), Dominik Viererbe (25401)

**Dozent:** Prof. Dr. O. Drögehorn

**Veranstaltung/Semester:** Rechnerkommunikation und Middleware  
Wintersemester 2019/2020

I Inhaltsverzeichnis

[1 Einleitung 5](#_Toc32911141)

[2 Aufgabenstellung 5](#_Toc32911142)

[2.1 Initiale Aufgabenstellung 5](#_Toc32911143)

[2.2 Modifikationen und Abgrenzung der Aufgabenstellung 6](#_Toc32911144)

[3 Architekturentwurf 7](#_Toc32911145)

[3.1 Logische Architektur 7](#_Toc32911146)

[3.2 Technische Architektur 8](#_Toc32911147)

[4 Implementierung 9](#_Toc32911148)

[4.1 Zeugs 9](#_Toc32911149)

[5 Diskussion der Ergebnisse 9](#_Toc32911150)

[6 Fazit 9](#_Toc32911151)

[7 Ausblick 10](#_Toc32911152)

[8 Quellen 11](#_Toc32911153)

[8.1 Literaturquellen 11](#_Toc32911154)

[8.2 Bildquellen 11](#_Toc32911155)

[9 Anhang 12](#_Toc32911156)

[9.1 Glossar 12](#_Toc32911157)

[9.2 Verwendete Programme 12](#_Toc32911158)

[9.3 Projektdateien 12](#_Toc32911159)

II Abbildungsverzeichnis

[Abbildung 4: Logische Architektur des PTN 7](#_Toc32911134)

[Abbildung 5: Technische Architektur des Prototypen 8](#_Toc32911135)

III Tabellenverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

IV Abkürzungsverzeichnis

|  |  |
| --- | --- |
| Abkürzung | Bedeutung |
| xxx | 3 x hintereinander |

V Versionsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Autor | Version | Datum | Änderungen |
| Dirk Neumann | 0.1 | 12.02.2020 | Dokument erstellt |

# Einleitung

Diese Entwurfsarbeit wird im Zuge der Lehrveranstaltung „Rechnerkommunikation und Middleware“ durchgeführt. Die Studierenden sollen strukturierte Entwicklungsarbeit, mit dem Fokus auf Implementierung und Dokumentation, anhand einer selbst gewählten Aufgabenstellung erlernen und vertiefen.

Mit zunehmendem Einzug des Internets in alle Gesellschafsbereiche und einer steigenden Vernetzung von Computern und Netzwerken, hat sich der Umgang mit Daten stark verändert. Dies ist nicht zuletzt auch dem Fortschreiten von Datenanalysemethoden geschuldet welche den Nutzen und das Gefahrenpotential von Daten stark erhöht haben. Dementgegen standen und stehen immer neu Entwicklungen im Bereich von Datenbanken, Sicherheitsmechanismen und Softwareparadigmen wie z.B. Webservices oder Cloud Computing. Eine der inzwischen nicht mehr neusten aber noch immer weit verbreiteten Technologien ist das BitTorrent-Protokoll, welcher das dezentralisierte und redundante sowie, im Zusammenhang mit verschiedenen Sicherheitsmaßnahmen, sichere Abspeichern und -abrufen von Daten ermöglicht. Der Aspekt der Sicherheit ist dabei je nach Implementierung durch andere Sicherheitsziele wie z.B. Integrität, Verschlüsselung, Authentizität oder Anonymität geprägt.

# Aufgabenstellung

Die Datenhaltung wie sie in einem solchen Rechnernetzwerk mit BitTorrent-Protokoll realisiert wird, ist eine praktische Anwendung der in der Vorlesung „Rechnerkommunikation und Middleware“ vermittelten Inhalte, darunter Datenserialisierung und –deserialisierung, Rechnerkommunikation und Kommunikationsprotokolle. Diese Inhalte sollen an einem praktischen Beispiel erlernt werden, indem ein ähnliches Rechnernetz aufgebaut wird. Dazu sollen die entsprechend notwendigen Komponenten identifiziert, beschrieben und implementiert werden und als Gesamtsystem getestet werden.

## Initiale Aufgabenstellung

Im Kontext der Lehrveranstaltung wird der zu realisierende Umfang des Projekts wie folgt definiert:

* Erstellen Sie ein Torrentnetzwerk mit Java und C#, die via CORBA kommunizieren.
* Das Netzwerk besteht aus einem Client, einem Trackerserver und mehreren Torrentservern.
* Das Netzwerk soll es ermöglichen Dateien zentral hochzuladen und dezentral herunterzuladen.
* Client:
* Der Nutzer besitzt einen Client. An diesem kann er eine Datei hochladen, die dann an den Trackerserver gesendet wird.
* Der Nutzer kann mit Hilfe des Clients den Datenbestand am Trackerserver anfragen und zu einer gewünschten Datei die Torrentserver ausfindig machen, welche die Fragmente der gewünschten Datei gespeichert haben.
* Der Client lädt diese Fragmente von den Torrentserver herunter und setzt diese dann selbstständig zur Zieldatei zusammen, welche der Nutzer unter einem gewünschten Pfad abgelegen kann.
* Trackerserver:
* Der Trackerserver nimmt Dateien vom Client an und teilt diese in mehrere Fragmente fest konfigurierter Größe auf.  
  Diese werden vom Trackerserver publiziert woraufhin sich die Torrentserver diese Dateifragmente selbstständig herunterladen.
* Der Trackerserver überwacht zentral über MQTT welcher Torrentserver welche Fragmente besitzt.
* Fragt ein Nutzer eine Datei an, so stellt der Trackerserver die nötigen Informationen zur Verfügung, um die ursprüngliche Datei wieder zusammenzusetzen.
* Torrentserver
* Der Torrentserver abonniert die Informationskanäle, welche der Trackerserver für diese zur Verfügung stellt.
* Werden neue Datenfragment vom Trackerserver ausgeschrieben, so entscheiden die Torrentserver nach einem zu definierenden Protokoll, ob sie dieses Fragment herunterladen und abspeichern.
* Der Trackerserver vermerkt diese Entscheidung.
* Auf Anfrage stellt der Torrentserver die geforderten Datenfragmente zur Verfügung.
* Die Abgabe enthält:
* Quellcode mit Quellcodedokumentation
* Gesamtdokumentation
* UML und andere Diagramme
* Entwurfsentscheidungen
* Arbeitsaufteilung aus welcher Bearbeitungsverantwortlichkeiten hervorgehen

Wie aus der Aufgabenstellung entnommen werden kann, handelt es sich bei dem Projekt nur um ein BitTorrent ähnliches Netzwerk. Dieses Pseudotorrentnetzwerk (PTN) besitzt einige, aber nicht alle Merkmale des BitTorrents, was das Projekt zum einen etwas leichter und damit innerhalb des Zeitrahmens realisierbar macht und zum anderen die Möglichkeit mit sich bringt Technologie wie MQTT zu verwenden, welche ebenfalls Vorlesungsinhalt war.

## Modifikationen und Abgrenzung der Aufgabenstellung

Die Aufgabenstellung wurde in Abstimmung mit dem betreuenden Professor im Nachhinein noch einmal angepasst. Die Anforderung, dass innerhalb des PTN mit CORBA kommuniziert werden soll wurde verworfen und durch die Anforderung ersetzt die Kommunikation stattdessen mit gRPC zu realisieren. Diese Entscheidung wurde erst am Ende der Analyse- und Konzeptionsphase getroffen, da der aktuellste Stand von CORBA und dessen Dokumentation aus dem Jahr XXX stammt und mit vielen aktuellen Techniken, darunter XXX nicht mehr einwandfrei kompatibel ist. Um die in der Konzeptionsphase bereits gewählten Kommunikationsschnittstellen beibehalten zu können, wurde eine Technologie gewählt, die CORBA sehr ähnlich ist. gRPC ist eine von Google entwickelte Technologie, die das gleiche Prinzip wie CORBA verfolgt und es ermöglicht an einer zentralen Stelle die Schnittstellen mittels derer die Teilnehmer kommunizieren zu definieren und diese Schnittstellen dann in einer von vielen unterstützten Programmiersprachen an den benötigten Orten implementieren zu können.

XXX Abgrenzung??? Nach erfolgreichem Anlegen einer finalen Aufgabenstellung konnte daraus ein lauffähiger Prototyp implementiert werden, wobei hier an verschiedenen Stellen abgegrenzt werden muss, was bestimmte Begriffe in verschiedenen Kontexten bedeuten.

# Architekturentwurf

Im Folgenden wird die Architektur, des zu implementierenden Gesamtsystems beschrieben. Dazu wird erst eine Unterteilung in verschiedene Aufgaben vorgenommen, welche dann im nächsten Schritt umgesetzt werden sollen.

## Logische Architektur

Der Nutzer kann mittels eines grafischen Interfaces mit dem PTN interagierten. Dieses wird vom PTN zur Verfügung gestellt und ist Teil des Projekts, gehört logisch aber nicht zum PTN. Der Nutzer kann über dieses Interface, wie in Abbildung 4 zu sehen, Daten in das PTN hochladen und Daten die er zuvor hochgeladen hat, oder auf die er Zugriff hat herunterlade. Das PTN verwaltet sich intern selbst. Es regelt welcher Teil der Daten wo abgespeichert wird und wie und ob ein Nutzer auf Daten zugreifen darf.

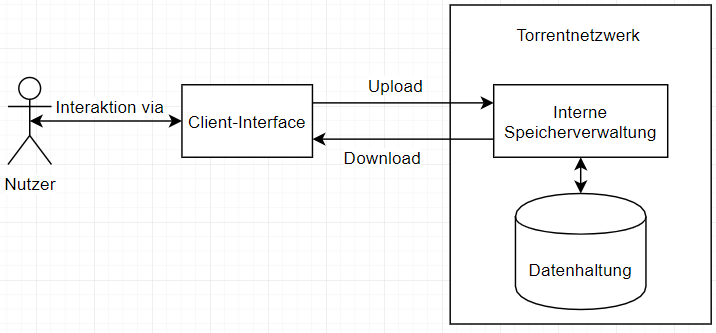


Abbildung : Logische Architektur des PTN

## Technische Architektur

Lädt ein Nutzer eine Datei in das PTN hoch, wird dieser vom Nutzerinterface zunächst in einzelne Datenpakete zerlegt, welche dann nach und nach hochgeladen werden. Der in Abbildung 4 als „Interne Speicherverwaltung“ bezeichnete Vorgang wird in Abbildung 5 weiter verfeinert. Beim Daten-in-das- PTN -Laden wird über einen Trackerserver organisiert, der die einzelnen Datenpakete ausschreibt. Die im Netzwerk befindlichen Torrentserver entscheiden ob sie das ausgeschriebene Datenpaket annehmen wollen oder nicht. Dies entscheidet der Torrentserver anhand der aktuellen Verteilung von Daten im Netzwerk. Möchte ein Nutzer eine Datei aus dem PTN herunterladen, so kann er sich an jeden beliebigen Torrentserver wenden, welcher dann die für die angeforderte Datei relevanten Datenpakete im PTN lokalisiert und diese Informationen zur Verfügung stellt. Im Anschluss lädt das Nutzer-Interface die restlichen benötigten Datenpakete von den entsprechenden Torrentservern herunter und fügt die Datei wieder zusammen. Der Nutzer kann dann die wieder zusammengefügte Datei abspeichern.

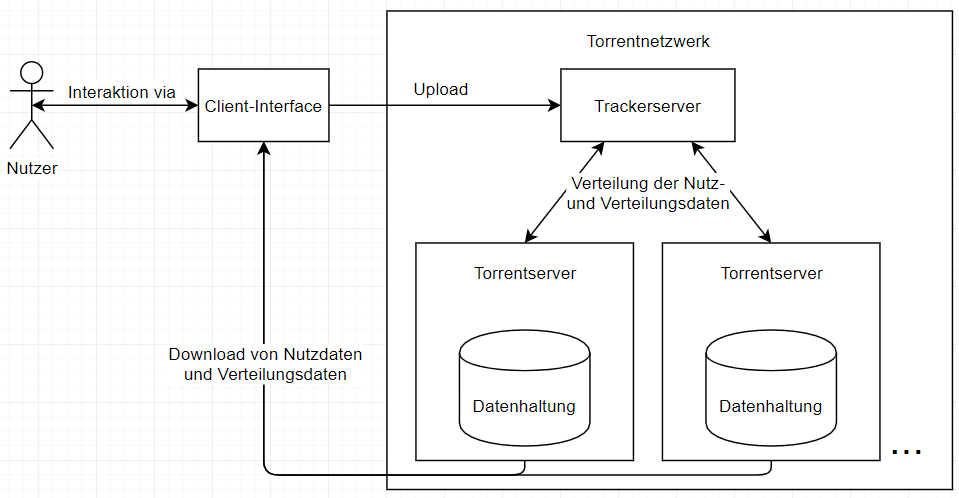


Abbildung : Technische Architektur des Prototypen

# Implementierung

## Zeugs

asdasd

# Diskussion der Ergebnisse

Im Rahmen der Entwurfsarbeit ist es gelungen, ein funktionsfähigen Prototypen zu implementieren. Dieser weist allerdings noch Mängel auf, die in dem bestehenden Kontext nicht mehr gelöst werden können/konnten. XXX ist eine große Schwäche, weil XXX. Auch XXX ist kritisch. XXX ist mit ausreichender Qualität gelöst worden, sollte aber XXX.

Zusätzlich zu den eben beschriebenen Softwaremängeln, treten auch bei dem Testaufbau Probleme auf. So fand ein Test des PTN nur im Rahmen XXX statt. Dies ist jedoch nicht gut, weil XXX. Des Weiteren wurde XXX. Auch nicht beachtet wurde, dass XXX.

# Fazit

Im Kontext dieser Entwurfsarbeit wurden der Aspekt der Kommunikation von mehreren Computerprogrammen untereinander genauer betrachtet. Dabei wurden die Protokolle zur Kommunikation entsprechend der Anforderungen selbst entworfen, implementiert und getestet. Dadurch konnte ein tieferes Verständnis über Rechnerkommunikation erworben werden und vor allem mit den Technologien gRPC und MQTT angewandt werden. Projektes wurde die Thematik Rückkehr des Wolfes nach Deutschland genauer betrachtet. Zugleich wurde mit der Thematik „Torrentnetzwerke“ ein besonderes Augenmerk auf die Punkte der Datensicherheit, darunter Redundanz, Integrität und Anonymität gelegt. Diese sind vor allem im Kontext der zunehmenden Digitalisierung von Daten und Prozessen besonders wichtig. Zudem beinhaltet die Thematik „Torrentnetzwerk“ auch Prinzipien von verteilten Systemen, welche auch in anderen zukunftsweisenden Themen wie Grid, Cloud und Edge Computing umgesetzt werden. Aufgrund einer Technologierecherche vor Beginn der Implementierungsphase, konnte Aufwand eingespart werden der eine Technologieänderung von CORBA zu gRPC mitten in der Implementierungsphase bedeutet hätte. Ab Beginn der Implementierungsphase wurde inkrementell gearbeitet und der dabei entstehende Stand mit jedem Inkrement mittels Modultests getestet. Ein abschließender Systemtest wurde am Ende der Implementierungsphase angeschlossen.

Mit Ablauf der Projektlaufzeit ist damit ein funktionsfähiger Prototyp eines abgewandelten Torrentnetzwerkes unter Berücksichtigung der zu Beginn aufgestellten Anforderungen konzipiert, realisiert und getestet. Die Tests haben das Funktionieren des Systems bestätigt. Für einen professionellen Einsatz ist das entwickelte System jedoch nicht geeignet. Zum einen fehlen noch weitere Sicherheitsfunktionen wie Verschlüsselung und Authentifikation, zum anderen wurde noch keine Belastungstests durchgeführt, um die Verfügbarkeit des Systems zu überprüfen und gegebenenfalls zu verbessern. Das selbstimplementierte System würde zudem den permanenten Betrieb mehrerer Server als Torrent- und Trackerserver erfordern sowie eingetragene Domains oder statische IP-Adressen, u die Funktionalität durchgehend zu gewährleisten. Dieser Kostenfaktor stünde nicht in Relation zum erbrachten nutzen und ist daher abzulehnen.

Als grundlegende Erkundung dieses Gebiets ist das Projekt dennoch erfolgreich gewesen. Es konnte gezeigt werden, dass es möglich ist in kurzer Zeit ein solches System mit Schnittstellenbeschreibungen zu entwerfen, zu implementieren und zu testen. Ob und wie diese Erkenntnis in Zukunft genutzt werden wird ist vorerst nicht ersichtlich, weitere Möglichkeiten was und wie verbessert werden kann wird im Ausblick dargestellt.

# Ausblick

Nach erfolgreicher Konzeption und Realisierung eines prototypischen Systems kann diese Grundlagenerkundung genutzt werden um ein potenziell marktreifes Produkt weiter zu entwickeln. Dabei ist vordergründig das Eliminieren des Trackerserver das Ziel das verfolgt werden muss, um das implementierte PTN als normales Torrentnetzwerk nennen zu können. Mittels eines dezentralen Hochladevorgangs von Nutzdaten könnte dieses Ziel weiter verfolgt werden. Des Weiteren müsste mit Abschaffung eines Trackerservers auch der regelmäßige Datenaustausch zwischen den Torrentservern gewährleistet sein, was zusätzliche Kommunikationsprotokolle erfordert. Ein Fall der bisher ebenfalls nicht betrachtet wurde ist die dynamische Veränderung der Anzahl von Torrentservern im Netzwerk. Fällt ein Torrentserver aus dem Netzwerk weg, müssen die Datenpakete die dieser mit sich führte neu verteilt werden ohne z.B. die Redundanz zu gefährden. Kommt ein neuer Server in das Netzwerk hinzu, müssen Pakete von den anderen Servern auf diesen neuen umverteilt werden

, die Unterscheidung zwischen Trackerserver und Torrentserver das Hardwarekomponenten und der Bildklassifizierungsalgorithmus als Punkte mit dem größten Ausbaupotenzial zu nennen. Im Bereich der Hardware ist eine hochauflösende Kamera mit großem Aufnahmewinkel, eines feste oder unabhängige Stromversorgung, sowie Witterungs- und Diebstahlschutz zu nennen. Im Bereich der Bildklassifizierung könnten bessere Ergebnisse mit einer lokalen, einer drittanbieterunabhängigen und/ oder einer selbst auf einen bestimmten Bereich trainierten künstlichen Intelligenz erzielt werden.

Werden diese Erweiterungen berücksichtigt oder ähnliche Verbesserungen eingeführt, die die Funktionalität des Gesamtsystems stabilisieren oder erweitern, kann ein solches Produkt die anfangs gesetzte Motivation als neues Projektziel erreichen und so einen Beitrag zur nachhaltigen Entwicklung der Pflanzen- und Tierwelt leisten. Zudem skaliert ein solches Projekt auch global, denn jedes System für sich ist unabhängig. Kumulierte man die Daten einzelner Systeme ließe sich zudem ein Beitrag zum Wolfsmonitoring und –management leiten.

Darüber hinaus ist die verwendete Technologie nicht nur auf Wölfe anwendbar. So wie Schafe vor Wölfen geschützt werden können, könnten auch Hühner für Füchsen oder Mardern geschützt werden. Genauso kann mit dieser Technologie nicht nur die Populationsentwicklung von Wölfen überwacht werden, sondern auch die anderer gefährdeter oder vom Aussterben bedrohter Tierarten. Damit sind tiererkennende Überwachungs- und Abschreckungssysteme umfangreich anwendbar und bergen ein großes Potenzial in Bezug auf wirtschaftliche als auch nachhaltige Projekte.

# Quellen

## Literaturquellen

1. Deutschland. Sachsen / Landesamt für Umwelt, Landwirtschaft und Geologie [Herausgeber/in], *Erprobung und Bewertung von Schutzmaßnahmen für Nutztiere vor dem Wolf, insbesondere der Einsatz von Herdenschutzhunden und Elektronetzen.* Dresden: Sachsen / Landesamt für Umwelt, Landwirtschaft und Geologie, 2014 [Online]. Verfügbar: https://www.publikationen.sachsen.de/bdb/artikel/22053/documents/30077. [Zugriff: 14.04.2019].

## Bildquellen

[WB1]

https://images.ecosia.org/1SgdlOP4r4trCziiTyogcgvsFE=/0x390/smart/http%3A%2F%2Ffootage.framepool.com%2Fshotimg%2Fqf%2F616256141-wolf-ungarn-stamm-pflanze-raubtier.jpg [29.07.2019]

# Anhang

## Glossar

|  |  |
| --- | --- |
| Wort | Bedeutung |
| Adafruit | Open-Source-Hardware Unternehmen |

## Verwendete Programme

[PG1] XnViewMP: https://www.xnview.com/de/xnviewmp/

[Zugriff: 28.07.2019]

[PG2] PuTTY https://www.putty.org/

[Zugriff: 30.08.2019]

[PG3] Etcher https://www.balena.io/etcher/

[Zugriff: 30.08.2019]

[PG4] Testprogramm Modultest https://github.com/NeumannDirk/Wolfsabschreckung

[Zugriff: 30.08.2019]

## Projektdateien

Link: <https://github.com/NeumannDirk/Wolfsabschreckung>