



Hochschule Karlsruhe  
Technik und Wirtschaft  
UNIVERSITY OF APPLIED SCIENCES

# **Analyse einer Referenzimplementierung des ActivityPub Standards mit Blick auf die Sicherheit**

Bachelor Thesis von

Armin Kunkel

an der Fakultät für Informatik und Wirtschaftsinformatik  
Fachrichtung Verteilte Systeme (VSYS)

Erstgutachter: Prof. Dr. rer. nat. Christian Zirpins  
Betreuer: M. Sc. Robert Schäfer

01. Dezember 2018 – 31. März 2019

Hochschule Karlsruhe Technik und Wirtschaft  
Fakultät für Informatik und Wirtschaftsinformatik  
Moltkestr. 30  
76133 Karlsruhe

---

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Änderungen entnommen wurde.

**Karlsruhe, 26. Dezember 2018**

.....  
(Armin Kunkel)



# **Zusammenfassung**

Deutsche Zusammenfassung



# Inhaltsverzeichnis

<b>Zusammenfassung</b>	<b>i</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	2
<b>2. Grundlagen zur sicheren Umsetzung dezentraler sozialer Netzwerke</b>	<b>3</b>
2.1. ActivityPub Standard . . . . .	3
2.1.1. Bestandteile des Protokolls . . . . .	4
2.1.2. Zugehörige Standards und Komponenten . . . . .	4
2.2. Client-zu-Server Kommunikation . . . . .	5
2.2.1. Client Teil . . . . .	5
2.2.2. Server Teil . . . . .	6
2.3. Server-zu-Server Kommunikation . . . . .	6
2.4. Authentifizierung und Datenintegrität . . . . .	7
<b>3. Analyse bestehender Ansätze</b>	<b>9</b>
3.1. Vergleich bestehender Implementierungen . . . . .	9
3.2. Verwandte Protokolle . . . . .	9
<b>4. Entwurf einer Lösung für sichere Server-zu-Server Interaktion mit ActivityPub</b>	<b>11</b>
4.1. Entwurfsentscheidung . . . . .	11
4.2. Technische Architektur . . . . .	11
<b>5. Implementierung eines ActivityPub Prototyps</b>	<b>13</b>
5.1. Server-zu-Server Protokoll . . . . .	13
<b>6. Evaluation</b>	<b>15</b>
6.1. Anwendungsbeispiel . . . . .	15
6.2. (Performanzmessungen) . . . . .	15
6.3. Diskussion von Vor- und Nachteilen der Lösung . . . . .	15
<b>7. Zusammenfassung und Ausblick</b>	<b>17</b>
<b>Literatur</b>	<b>19</b>
<b>A. Anhang</b>	<b>21</b>
A.1. First Appendix Section . . . . .	21






# 1. Einleitung

Zentralisierung von Daten und das Vertrauen auf einzelne Instanzen ist heutzutage allgegenwärtig. Im Internet gibt es eine Fülle an sozialen Netzwerken wie Facebook, Twitter, Google+, Instagram oder Pinterest die zumeist das Recht an den Daten, die Sie von ihren Nutzer bekommen, behalten und nicht zuletzt diese Daten auch verkaufen für zB. Werbezwecke. Für Kriminelle sowie Geheimdienste ist es außerdem leichter an die gesamte Datenbank zu kommen, da bei zentralisierten Netzwerken meist auch eine zentrale Datenbank, bzw. ein zentraler Zugriffspunkt, vorhanden ist. Wenn der Zugriff auf diesen/-diese erreicht ist kann die ganze Datenbank ausgelesen werden.

Was bedeutet nun dezentral? In der Politik bedeutet Dezentralisierung „die Übertragung zentralstaatlicher Aufgaben auf subnationale oder subsidiäre Ebene(n)“[10]. In der Energiewirtschaft spricht man von einer „dezentralen Stromerzeugung“, wenn der Strom an den Stellen wo er verbraucht auch erzeugt wird. Ein Beispiel hierfür wäre ein Wasserkraftwerk, dass den Strom für die umgebenen Dörfer oder Städte liefert[9]. In der Informatik versteht man unter Dezentralisierung das verteilen von u.A. Daten über mehrere unabhängige Server hinweg.

Durch die Verteilung der zentralstaatliche Aufgaben auf subnationale oder subsidiäre Ebene wird die zentralstaatliche Ebene entlastet und hat somit mehr Ressourcen für andere Aufgaben. Beim Errichten von Wasserkraftwerken nahe an Dörfern und Städten entfällt der Transport des Stroms und somit verringert sich der Verlust beim Transport über das Stromnetz. Das Dezentralisieren von sozialen Netzwerken bring verschiedene Vorteile mit sich. Ein Vorteil ist die Skalierbarkeit bei dezentralen sozialen Netzwerken. Durch das hinzufügen weiterer Instanzen können dem Netzwerk neue Ressourcen bereitgestellt werden. Ein weiterer Vorteil liegt darin, dass keine zentrale Kontrollinstanz vorhanden ist welche korumpiert werden kann, sei es durch Kriminelle, wirtschaftliche Interessenverteter oder staatliche Autoritäten. 

ActivityPub besteht aus zwei Teilen. Dem Client-zu-Server Protokoll und dem förderierten Server-zu-Server Protokoll. Durch die ActivityPub W3C Empfehlung Anfang 2018 und die Social Web Working Group Arbeitsgruppe des W3C sowie weiteren, wird die Verbreitung des Protokolls vorangebracht.

Der Netzwerkeffekt ist ein aus der Volkswirtschaftslehre stammender Begriff. Übertragen bedeutet er soviel wie: „Je mehr Netzwerke den ActivityPub Standard implementieren, desto höher ist der Nutzen für ein einzelnes Netzwerk und im Endeffekt für die einzelnen Nutzer“.



Bibliothek gehen (Bibliothek gehen): Netzwerkeffekt, Dezentralisierung 3 mal

Um die Arbeit erfolgreich abzuschließen soll eine Implementierung des verteilten Server-zu-Server Protokolls als Prototyp und eventuell ein Konzept zum sicheren Einsatz des Protokolls vorliegen. Darüber hinaus soll die Arbeit einen Überblick über die Grundlagen des Protokolls geben und wie die relevanten Bestandteile zu verstehen sind. In diesem Zusammenhang bedeutet „sicher“, das sichere Authentifizieren vom Client gegenüber dem Server und zusätzlich der Server untereinander, sowie das Sicherstellen der manipulationsfreien Übertragung der Inhalte.

Der ActivityPub[**activityPub**] Standard wurde am 23 Januar 2018 von der W3C empfohlen und von einer Arbeitsgruppe des W3C, der Social Web Working Group (SWWG)[1, 2], entwickelt. Diese Gruppe war vom 21. Juli 2014 bis zum 13 Februar 2018 aktiv[1] und entwickelte unter anderem ActivityPub, ActivityStreams Core[4], ActivityStreams Vocab[3]. Die SWWG war eine Arbeitsgruppe des W3C mit dem Ziel neue Protokolle, Vokabulare und API's zu definieren für den Zugriff auf soziale Inhalte der sogenannten Open Web Platform[8].

### 1.1. Motivation

Bei zentralen sozialen Netzwerken, z.B. Facebook, besitzt die Kontrolle über die Daten das Unternehmen. Durch die Dezentralisierung bleiben die Daten bei den einzelnen Instanzen des sozialen Netzwerkes. Die Datenbanken bei zentralen Netzwerken können zwar verteilt sein und auch das Netzwerk könnte dezentral angelegt sein, trotzdem gehören die Daten einem Unternehmen. Außerdem können bei zentralen sozialen Netzwerken nicht ohne weiteres Inhalte mit anderen Netzwerken ausgetauscht werden. Mit einem Protokoll wie ActivityPub wird das Verteilen sowie der Zugriff auf Inhalte und der Austausch von Aktivitäten über mehrere soziale Netzwerke hinweg ermöglicht.


Protokolle wie „Diaspora Federation und OStatus“ bieten den Zugriff auf dezentral gespeicherte soziale Inhalte und ermöglichen das veröffentlichen von sozialen Inhalten. Der im März letzten Jahres vom W3C empfohlene Standard namens „ActivityPub“ wurde auf Basis des Wissens im Umgang mit dem OStatus und Pump.io Protokoll entwickelt[7]. ActivityPub ist wie „OStatus“ ein Protokoll für dezentrale soziale Netzwerke und wird in dieser Bachelor Arbeit untersucht.




## 2. Grundlagen zur sicheren Umsetzung dezentraler sozialer Netzwerke

„ActivityStreams 2.0“ beinhaltet Modelle für Aktoren, Aktivitäten, Intransitiven Aktivitäten, Objekte, Links, Sammlungen, Natürliche Sprachwerte (Strings) und für Internationalisierung. Das Kernvokabular von ActivityStreams 2.0 wird durch ActivityStreams Vocab erweitert. Dazu gehören verschiedene Aktivitätstypen wie z.B. „Accept“, „Add“, „Remove“, „Delete“ und „Create“<sup>1</sup>, um Aktorentypen wie „Person“, „Application“ und „Group“<sup>2</sup> sowie um verschiedenste Objekttypen wie „Article“, „Event“, „Note“ und „Relationship“<sup>3</sup>.

Sammlungen beschreiben!

Wusste eben nicht wie ich das anders machen soll außer als Fußnote. Wie könnte ich ein fliehe Weblink“gestalten??? Weil alles hier nochmal aufzulisten wäre denke ich nicht gut.. 

Meinst du mit Übersichtsdiagramm, UseCase Diagramme mit z.B. Client to Server Interaktionen als Package und senden von Aktivitäten an die Outbox sowie empfangen von der Inbox als Usecases??? usw.. 

### 2.1. ActivityPub Standard

ActivityPub definiert zwei Protokollschichten, sowie Konzepte, Sammlungen und Interaktionen für dezentrale soziale Netzwerke. Eine Protokollschicht ist das Client-zu-Server Protokoll (Social API), um Clients den Zugriff auf einen Server zu ermöglichen sowie zum entgegennehmen von Anfragen. Die zweite Protokollschicht besteht aus dem föderierten Server-zu-Server Protokoll (Federation Protocol), welches den einzelnen Instanzen von dezentralen sozialen Netzwerken den Austausch von Inhalten untereinander gestattet. ActivityPub setzt auf bereits bestehende Empfehlungen des WWW Consortium (W3C) auf, welche teilweise auch von der SWWG entwickelt wurden wie z.B. ActivityStreams Core und ActivityStreams Vocab.

Auch andere Technologien wie JSON-LD werden benutzt um die Erweiterbarkeit zu gewährleisten. Über neue Ontologien (Vokabulare) können weitere syntaktische Definitionen und semantische Beschreibungen zu den bestehenden hinzugefügt werden. Diese

<sup>1</sup>activity-types <https://www.w3.org/TR/activitystreams-vocabulary/>

<sup>2</sup>actor-types <https://www.w3.org/TR/activitystreams-vocabulary/>

<sup>3</sup>object-types <https://www.w3.org/TR/activitystreams-vocabulary/>

Vokabulare können im Kontext des JSON-LD Objektes, angegeben werden.

Stimmt das mit den Ontologien???

Beispiel Context Abbildung mit erweitertem Context???



### 2.1.1. Bestandteile des Protokolls

Das Client-zu-Server sowie förderierte Server-zu-Server Protokoll können unabhängig voneinander implementiert werden. Ersteres besteht aus einem Client und Server Teil.

In ActivityPub werden Benutzer als „Aktoren“(actors) dargestellt. Diese können nicht nur Personen, sondern auch Applikationen, Organisationen, Gruppen und Services sein[5]. Jedes Aktoren Objekt muss eine „Inbox“ und „Outbox“, welche geordnete Sammlungen sein müssen, sowie eine ID und ein Typ besitzen[**activityPubActor**]. Die ID muss global einzigartig sein. Dies kann garantiert werden durch eine Domänen und Protokoll bezogene URI oder IRI wie zum Beispiel „https://example.org/users/alice“oder „https://example.org/alice/posts/2“. Die Typ Eigenschaft kann variieren zwischen fünf oben genannten.



---

Listing 2.1: Beispiel Aktoren Objekt

---

### 2.1.2. Zugehörige Standards und Komponenten

ActivityPub benutzt die ActivityStreams Daten Syntax und das Vokabular. Zusätzlich kann ein Sicherheitsvokabular benutzt werden. Am 22 April 2016 hat die „W3C Community Group“ einen Entwurfsl<sup>1</sup> herausgebracht. Durch diesen wird neue Syntax und Semantik definiert um Internet basierten Applikationen das Verschlüsseln, Entschlüsseln sowie digitale signieren und verifizieren von verlinkten Daten (Linked Data) zu ermöglichen. Es enthält auch Vokabeln für die Erstellung und Verwaltung einer dezentralen Public-Key-Infrastruktur über das Internet.”[6]. Ein Anwendungsfall ist das holen des öffentlichen Schlüssels eines Nutzers, über dessen Aktoren Objekt, um eine von Nutzer gesendete Nachricht zu verifizieren.

JSON-LD ist eine Erweiterung des JSON Formates um verlinkte Daten zu Repräsentieren. JSON an sich, ist ein Format welches im Web häufig Anwendung findet um Daten auszutauschen. Im Kern sind ActivityStreams 2.0 auch JSON-LD Objekte. Der ActivityStreams 2.0 Kontext definiert verschiedene Klassen und Eigenschaften, von denen nicht alle benutzt werden. Typische Klassen sind „Activity“, „Link“ und „OrderedCollection“. Ein Beispiel ActivityStreams 2.0 Objekt sieht wie folgt aus:

---

Listing 2.2: Beispiel ActivityStreams 2.0 Objekt

---



Über ein Profil ist es möglich, einem „Media-Type“ zusätzliche Semantik hinzuzufügen <https://tools.ietf.org/html/rfc6906>.

Es ist auch möglich ein Profil anstatt eines Aktoren Objektes anzugeben.

Die letzten beiden Absätze gehören zur Frage..

Mir ist nicht ganz klar was mit dem Profil gemeint ist.. laut dem RFC am Ende des dritten Absatzes, wird das verwendet um neue Semantik hinzuzufügen.. Bloß wird dafür doch schon der JSON-LD Context benutzt "@context": "https://www.w3.org/ns/activitystreams"?



## 2.2. Client-zu-Server Kommunikation

Der Client interagiert über zwei Schnittstellen mit dem Server. Er kann sich per HTTP GET Anfrage auf seine eigene „Inbox“ die neusten an ihn adressierten Inhalte holen und über eine HTTP POST Anfrage auf seine „Outbox“ neue Inhalte erstellen.

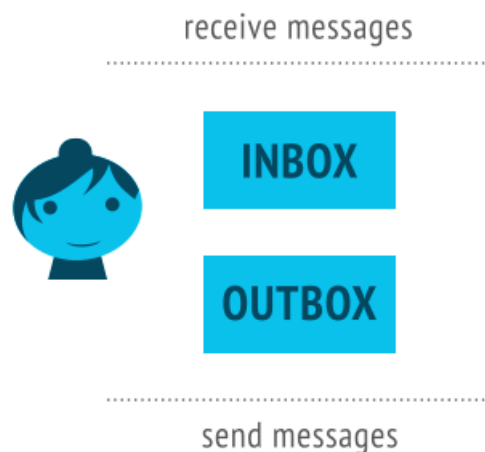





Abbildung 2.1.: Interaktionen des Client mit dem Server

Um neue Aktivitäten an seine Folgenden, direkt an einzelne Personen, sichtbar innerhalb der Domain oder unangemeldet für alle zugänglich zu veröffentlichen, kann der Client eine HTTP POST Anfrage an seine „Outbox“ senden mit einer Aktivität oder einem ActivityStreams 2.0 Objekt als Inhalt. Der Client muss mehrere Aufgaben, wie in 2.2.1 beschrieben, übernehmen.

### 2.2.1. Client Teil

- Auffinden der URL über das Aktoren Objekt
- Serialisieren der Daten in eine Aktivität oder ActivityStreams 2.0 Objekt
- Adressieren von Aktivitäten oder ActivityStreams 2.0 Objekten

- Eine HTTP POST Anfrage mit „Content-Type: application/ld+json: profile=“https://w3.org/ns/activitystreams“ oder „Content-Type: application/activity+json: profile=“https://w3.org/ns/activitystreams“ an seine „Outbox“ absetzen können 
- Bei den Aktivitätstypen CREATE, UPDATE, DELETE, FOLLOW, ADD, REMOVE, LIKE, BLOCK, UNDO wird die „object“ Eigenschaft bereitgestellt 
- Die „target“ Eigenschaft wird bei den Aktivitätstypen ADD und REMOVE bereitgestellt 
- Anfragen müssen mit den Zugangsdaten des Nutzer, zu dem die „Outbox“ gehört, authentifiziert werden

Die beiden Eigenschaften „target“ und „object“ müssen bei verschiedensten Aktivitätstypen bereitgestellt werden. Zum Beispiel muss die „target“ Eigenschaft bei den Aktivitätstypen ADD und REMOVE angegeben werden.

Der Client Teil des Client-zu-Server Protokolls beinhaltet die oben gelisteten Aufgaben. Zusammengefasst muss sich der Client um das Entdecken von Endpunkten - Inbox/Outbox - über Aktorenobjekte, das Adressieren von Aktivitäten und Serialisieren von Daten zu Aktivitäten oder ActivityStreams 2.0 Objekten sowie um das setzen der entsprechenden Kopfzeilen (Mittlerer Teil der obigen Auflistung) und Absenden der HTTP POST Anfrage kümmern. Außerdem muss bei HTTP GET Anfragen an den Server die „ACCEPT: application/ld+json“ Kopfzeile gesetzt sein.

Wenn ein Nutzer z.B. ein „Post“ erstellt, wird der „Post“ entweder zu einem ActivityStreams 2.0 Objekt oder direkt zu einer Aktivität serialisiert. Anschließend wird das serialisierte Objekt adressiert. Dies geschieht über das beschaffen der „Inboxen“ über die Aktoren Objekte der Empfänger. Sobald die Endpunkte bekannt sind, können diese in das „to“ Feld in dem Objekt geschrieben werden. Zum Schluß wird eine authentifizierte HTTP POST Anfrage, wie in Element 4 der obigen Liste beschrieben, abgesetzt.

### 2.2.2. Server Teil

Zu den Aufgaben des Serverteils gehören das Annehmen von HTTP POST Anfragen auf die „Outbox“ eines Nutzers und das verifizieren ob der Nutzer berechtigt ist diese Anfrage zu tätigen. Weiter werden die „Inbox“ Endpunkte bereitgestellt um authentifizierten Nutzern den Zugriff auf die neusten Inhalte zu geben. Der Server Teil des Client-zu-Server Protokolls bezieht sich nur auf eine Domäne. Für das verteilen von Aktivitäten an andere Server muss zusätzlich das förderierte Server-zu-Server Protokoll implementiert werden.

## 2.3. Server-zu-Server Kommunikation

Bei der Server-zu-Server Kommunikation bestehen die Hauptaufgaben im annehmen und zustellen von HTTP POST Anfragen auf die „Inboxen“ und „Outboxen“ der Nutzer und

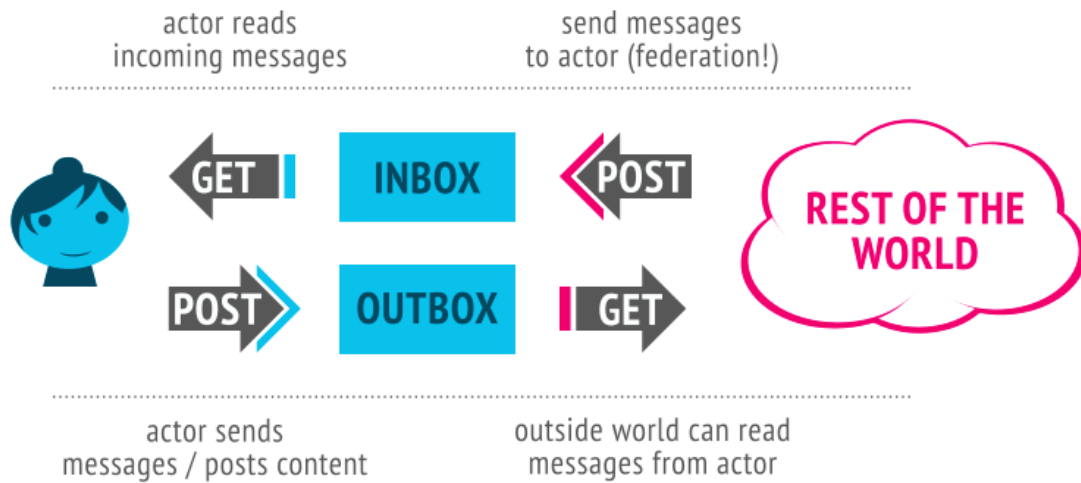


Abbildung 2.2.: Schnittstellen des ActivityPub Protokolls

im Weiterleiten von Aktivitäten die der Server nicht zustellen kann. Empfängt dieser eine Aktivität welche nicht an die zugehörige Domäne gerichtet ist, wird die Aktivität an den zur Domäne gehörigen Server gesendet. Dazu kommt das Bereitstellen der „Outbox“ Sammlungen.



## 2.4. Authentifizierung und Datenintegrität

Für die Authentifizierung und zum sichern der Datenintegrität definiert der Standard keine Mechanismen. Es gibt allerdings „Best Practices“ für die Umsetzung dieser Anforderungen.

Zum einen werden bei der Client-zu-Server Authentifizierung „OAuth 2.0“ Tokens benutzt, zum anderen auf der Server Seite „HTTP“ oder „Linked Data Signatures“ zur Sicherstellung der Datenintegrität.

Datenintegrität

OAuth 2.0

HTTP Signatures



Um sicherzustellen das HTTP Anfragen beim Transport nicht verändert wurden, können HTTP Signatures verwendet werden.

Linked Data Signatures

Wenn ein Objekt nicht nur vom Client zum Server gesendet, sondern auch zwischen Servern untereinander weitergeleitet werden soll wird zum Sicherstellen der Datenintegrität ein anderes Verfahren benötigt als HTTP Signatures. Die „Best Practices“ empfehlen

für solche Fälle „Linked Data Signatures“. Der größte Unterschied zwischen HTTP Signaturen und „Linked Data Signatures“ besteht darin, welche Daten zum Erstellen der Signatur verwendet werden. Bei HTTP Signaturen sind es die Kopfzeilen. Mit „Linked Data Signatures“ kann auch das Objekt selbst, also der Payload einer HTTP Anfrage, anstatt nur die Kopfzeilen, zum signieren verwendet werden.



## **3. Analyse bestehender Ansätze**

### **3.1. Vergleich bestehender Implementierungen**

### **3.2. Verwandte Protokolle**

- OStatus
- Diaspora Federation



## **4. Entwurf einer Lösung für sichere Server-zu-Server Interaktion mit ActivityPub**

### **4.1. Entwurfsentscheidung**

### **4.2. Technische Architektur**



## **5. Implementierung eines ActivityPub Prototyps**

### **5.1. Server-zu-Server Protokoll**



## **6. Evaluation**

### **6.1. Anwendungsbeispiel**

### **6.2. (Performanzmessungen)**

### **6.3. Diskussion von Vor- und Nachteilen der Lösung**





## **7. Zusammenfassung und Ausblick**



# Literatur

- [1] W3 Consortium. *Social Web Working Group*. Date accessed: 12.12.2018. 2018. URL: <https://www.w3.org/Social/WG>.
- [2] W3 Consortium. *W3C launches push for social web application interoperability*. Date accessed: 21.07.2014. 2014. URL: <https://www.w3.org/blog/news/archives/3958>.
- [3] Evan Prodromou James M Snell. *Activity Vocabulary*. Date accessed: 23.05.2017. 2017. URL: <https://www.w3.org/TR/activitystreams-vocabulary/>.
- [4] Evan Prodromou James M Snell. *ActivityStreams 2.0*. Date accessed: 23.05.2017. 2017. URL: <https://www.w3.org/TR/activitystreams-core/>.
- [5] Evan Prodromou James M Snell. *ActivityStreams 2.0*. Date accessed: 23.05.2017. 2017. URL: <https://www.w3.org/TR/activitystreams-core/>.
- [6] Inc. Manu Sporny Digital Bazaar. *The Security Vocabulary*. Date accessed: 22.04.2016. 2016. URL: <https://web-payments.org/vocabs/security>.
- [7] W3C. *ActivityPub Acknowledgements*. Date accessed: 23.01.2018. 2018. URL: <https://www.w3.org/TR/activitypub/#acknowledgements>.
- [8] W3C. *Social Web Working Group Charter*. Date accessed: 17.12.2018. URL: <https://www.w3.org/2013/socialweb/social-wg-charter>.
- [9] Wikipedia. *Dezentrale Stromerzeugung*. Date accessed: 06.11.2018. 2018. URL: [https://de.wikipedia.org/wiki/Dezentrale\\_Stromerzeugung](https://de.wikipedia.org/wiki/Dezentrale_Stromerzeugung).
- [10] Wikipedia. *Dezentralisierung (Politik)*. Date accessed: 10.08.2018. 2018. URL: [https://de.wikipedia.org/wiki/Dezentralisierung\\_\(Politik\)](https://de.wikipedia.org/wiki/Dezentralisierung_(Politik)).



# A. Anhang

## A.1. First Appendix Section

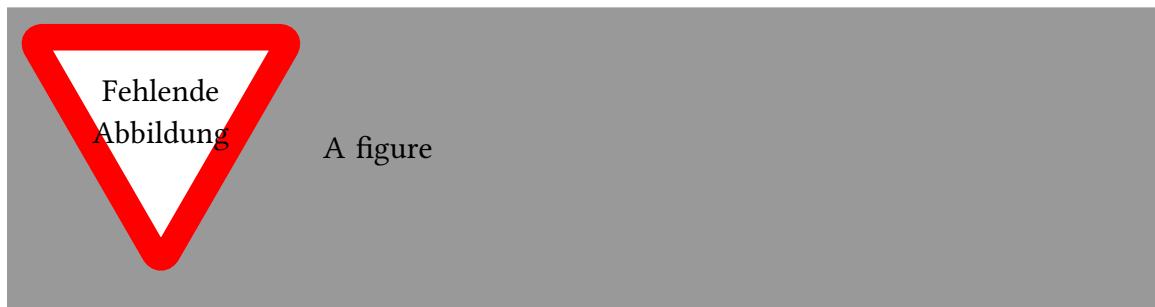


Abbildung A.1.: A figure

...