HOCHSCHULE
MITTWEIDA
University of Applied Sciences

MASTERARBEIT

Herr
Timo Burkhardt

Mechanismen zur Definition, Hinterlegung, Abruf und Versionierung von Credential-Schemata

Mittweida, 2023

Fakultät **Angewandte Computer- und Biowissenschaften**

MASTERARBEIT

Mechanismen zur Definition, Hinterlegung, Abruf und Versionierung von Credential-Schemata

Autor:
Timo Burkhardt

Studiengang:
Blockchain & Distributed Ledger Technologies

Seminargruppe:
BC19w1-M

Erstprüfer:
Prof. Dr.-Ing. Andreas Ittner

Zweitprüfer:
M.Sc. Christoph Menzer

Einreichung:
Mittweida, 2023

Verteidigung/Bewertung:
Mittweida, 2023

Faculty of **Applied Computer Sciences and Biosciences**

MASTER THESIS

Mechanisms for defining, depositing, retrieving and versioning of Credential Schemes

Author:
Timo Burkhardt

Course of Study:
Applied Computer Science

Seminar Group:
BC19w1-M

First Examiner:
Prof. Dr.-Ing. Andreas Ittner

Second Examiner:
M.Sc. Christoph Menzer

Submission:
Mittweida, 2023

Defense/Evaluation:
Mittweida, 2023

Bibliografische Beschreibung:

Burkhardt, Timo:
Mechanismen zur Definition, Hinterlegung, Abruf und Versionierung von Credential-Schemata. – 2023. – 59 S.
Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Masterarbeit, 2023.

Referat:

In der aktuellen digitalen Ära gewinnt das Konzept der Self-Sovereign-Identity (SSI) zunehmend an Bedeutung, da es Nutzern ermöglicht, Kontrolle über ihre eigene Identität und die damit verbundenen Informationen auszuüben. Um die Informationsqualität von Credentials in diesem Kontext zu gewährleisten, ist der Einsatz von Schemata notwendig. Die Mechanismen Definition, Hinterlegung, Abruf und Versionierung des Schemaprozesses werden analysiert und bewertet, um eine effektive Strukturierung und Interpretierbarkeit der Information innerhalb der Credentials zu ermöglichen. Dabei wird ein Überblick über die Technologielandschaft geboten und ein umfassendes Verständnis der Mechanismen sowie deren Auswirkungen auf die Informationsqualität von Credentials und die Interoperabilität von SSI-Technologien vermittelt. Die Arbeit untersucht auch die Rolle von Schemata im Ökosystem von SSI durch den Trust-over-IP-Stack, die Beziehung zwischen Schema und Information sowie die gegenseitige Abhängigkeit von Qualität. Die Credential-Formate AnonCreds und Verifiable-Credentials-Data-Model (VCDM) werden verglichen. Bestehende Credential-Type-Definitionen, editorbasierte Plattformansätze und Overlay-Strukturen in Verwendung mit dem VCDM werden analysiert. Die Arbeit betrachtet schließlich polymorphe Schemata, das Semantic-Web und Möglichkeiten zur Verbesserung semantischer Zuordnung von Credentials und der Interpretierbarkeit von Informationen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Quelltextverzeichnis

Abkürzungsverzeichnis

CAS . . . . . Content-Addressable-Storage

CERD_DEF . . . . . Credential-Definition

CF . . . . . Credential-Framework

CLR . . . . . Comprehensive-Learner-Record

DID . . . . . Decentralized-Identifier

DIF . . . . . Decentralized-Identity-Foundation

DRI . . . . . Decentralized-Resource-Identifier

EF . . . . . Ecosystem-Framework

IPFS . . . . . Inter-Planetary-File-System

IQ . . . . . Informationsqualität

IRI . . . . . Internationalized-Resource-Identifier

LD . . . . . Linked-Data

OCA . . . . . Overlays-Capture-Architecture

OWA . . . . . Open-World-Assumption

OWL . . . . . Web-Ontology-Language

PII . . . . . Persönlich identifizierbare Informationen

RDF . . . . . Resource-Description-Framework

RDFS . . . . . Resource-Description-Framework-Schema

SAID . . . . . Self-Addressing-Identifier

SHACL . . . . . Shapes-Constraint-Language

SOyA . . . . . Semantic-Overlay-Architecture

SSI . . . . . Self-Sovereign-Identity

ToIP . . . . . Trust-over-IP

VC . . . . . Verifiable Credential

VCDM . . . . . Verifiable-Credentials-Data-Model

VDR . . . . . Verifiable-Data-Registry

VP . . . . . Verifiable Presentation

W3C . . . . . World-Wide-Web-Consortium

1 Einleitung

In der aktuellen digitalen Ära wurden Identitätssysteme stetig weiterentwickelt, um den wachsenden Anforderungen von Nutzern und Organisationen gerecht zu werden. Die Evolution von Modellen zur Abbildung von Identität durchlief verschiedene Phasen, beginnend mit zentralisierten Ansätzen über nutzerzentrierte und föderierte Modelle bis hin zu selbstsouveränen Konzepten (vgl. [Str+21]). Self-Sovereign-Identity (SSI) repräsentiert dabei einen signifikanten Wendepunkt, indem sie Nutzern ermöglicht, Kontrolle über die eigene Identität und attestierten Informationen in Verbindung mit ihrer Entität eigenständig zu verwalten. Im Kontext selbstsouveräner Identitäten SSI ist die Flexibilität verifizierbarer Informationen von entscheidender Bedeutung, da sie die Souveränität der Nutzer unterstützt und ihnen erlaubt, Informationen im Digitalen zu attestieren und zu verifizieren. Die resultierende größere Freiheit in der Informationsgestaltung und im Informationsvolumen erfordert eine gewisse Form der Strukturierung, um unkontrollierte Wucherung zu verhindern. Um eine effiziente Verarbeitung sowie eine breite Anwendbarkeit der Information in verschiedenen Kontexten und Domänen sicherzustellen, ist der Einsatz von Schemata notwendig. Diese verhindern die Ansammlung unstrukturierter Informationen und fördern die Qualität und Aussagekraft der Informationen für deren Nutzung.

Als Träger der Information fungiert im Digitalen ein digitaler Nachweis, der als Verifiable Credential bezeichnet wird. Ein Verifiable Credential wird durch ein Credential-Schema strukturiert. Das Schema bildet die Vorlage für das Anlegen der Information. Den Nutzern und Autoren ist Zugang zu dieser strukturiert vorliegenden Information, in Form der Schemata, und zu den an das Schema gebundenen Prozess zu gewähren. Um die Erstellung und Verwendung von Schemata zu erleichtern und zu gliedern, bedarf es definierter Mechanismen. Die Systematisierung des Schemaprozesses soll die Anwendung und Nutzbarkeit von Credentials sowie die Qualität und Interpretierbarkeit der Credential-Schemata steigern.

Ein Ziel dieser Arbeit ist es, Ansätze zur Erreichung einer möglichst hohen Informationsqualität von Credentials durch den Einsatz von Schemata zu untersuchen. Die Mechanismen Definition, Hinterlegung, Abruf und Versionierung des Schemaprozesses werden analysiert und bewertet, in der Absicht eine effektive Strukturierung und Interpretierbarkeit der Information innerhalb der Credentials zu gewährleisten. Die Ergebnisse dieser Arbeit sollen dazu beitragen, die Mechanisierung und damit die Systematisierung des Schemaprozesses in digitalen Vertrauensräumen zu schaffen und zu verbessern, und somit die Akzeptanz und Nutzbarkeit von Verifiable Credentials zu erhöhen. Die Arbeit soll einen Überblick der Technologielandschaft und ein umfassendes Verständnis der Mechanismen zur Definition, Hinterlegung, Abruf und Versionierung von Credential-Schemata und deren Auswirkungen auf die Informationsqualität von Credentials sowie die Interoperabilität von SSI-Technologien vermitteln.

Die Einordnung des Credential-Schemas im Komplex von SSI und dessen Ökosystem erfolgt über eine Klassifizierung durch den Trust-over-IP (ToIP)-Stack. Die Betrachtung dieser Vertrauensinfrastruktur konzentriert sich neben der Auseinandersetzung mit grundlegenden Komponenten und Akteuren besonders auf die Rolle oder Funktion, die das Schema darin übernimmt, und auf die Stellen, an denen es zum Einsatz kommt. In diesem Zusammenhang stellt sich die Frage, in welchem Verhältnis Vertrauensentscheidungen und Schemata stehen, die auf verifizierbaren Informationen basieren und

einer Zuordnung zu einem bestimmten Kontext vorausgehen. Der Schemaprozess wird in Bezug auf die beteiligten Akteure, deren Funktionen und die an sie gestellten Anforderungen analysiert. Die Beziehung zwischen Schema und Information wird näher untersucht, insbesondere im Hinblick auf ihre Qualität und gegenseitige Abhängigkeit. Dabei stellt sich die Frage, inwiefern sie sich gegenseitig bedingen und ob ihre Qualität in Abhängigkeit voneinander verbessert werden kann. Das Schema fungiert als strukturierende Vorlage der Information und wird durch den Kontext sowie die Definition auf syntaktischer, semantischer und pragmatischer Ebene bestimmt. In einer Literaturrecherche werden Informationsqualitätsdimensionen und zugehörige Metriken identifiziert, die der Entwicklung eines Bewertungsrahmens für die Mechanismen des Schemaprozesses dienen. Die Credential-Formate, welche das Datenmodell und den Gestaltungsrahmen für Schemata festlegen, werden anhand der Formate AnonCreds und des Verifiable-Credentials-Data-Model (VCDM) verglichen. Diese Gegenüberstellung zielt darauf ab, die unterschiedlichen Ansätze der Formate hinsichtlich der Informationsrepräsentation und der jeweiligen Mechanismen der Schemaprozesse zu analysieren. Basierend auf dem VCDM wird der Schemaprozess detaillierter untersucht. Dabei werden bestehende Credential-Type-Definitionen, editorbasierte Plattformansätze und Overlay-Strukturen ausgerichtet auf die vier Mechanismen analysiert und verglichen. In diesem Zusammenhang finden polymorphe Schemata, das Semantic-Web sowie Möglichkeiten der Verbesserung semantischer Zuordnung von Credentials und der Interpretierbarkeit von Information Betrachtung.

2 Grundlagen

Im vorliegenden Kapitel werden die zentralen Begriffe und Konzepte erläutert, welche im Rahmen der Arbeit relevant sind. Zentrales Konzept stellen die Self-Sovereign-Identity-Ökosysteme und ihre digitalen Nachweise dar.

2.1 Self-Sovereign-Identity (SSI)

Das Paradigma der Self-Sovereign-Identity (SSI) stellt eine innovative Identitätsmanagement-Methode dar, welche auf den zehn Prinzipien Allens basiert. Der Fokus liegt hierbei darauf, dem Endnutzer eine autonome Kontrolle seiner Identität zu gewährleisten und somit die Datensouveränität des Individuums zu fördern. Nachfolgend die zehn Prinzipien nach Allen [All16]:

1. Eigenständige Existenz: Ein Individuum muss unabhängig von digitalen Identifikatoren eine eigenständige Existenz haben, die nur als Teilmenge digital repräsentiert wird.
2. Kontrolle: Individuen müssen die Kontrolle über ihre Identität haben und in der Lage sein, sie ohne Einflussnahme Dritter zu aktualisieren, zu verbergen oder darauf zu verweisen.
3. Zugang: Individuen müssen Zugang zu allen Daten haben, die sich auf ihre Identität beziehen, und in der Lage sein, diese Daten bei Bedarf abzurufen.
4. Transparenz: Jegliche Systeme und Algorithmen, die für digitale Identitäten eingesetzt werden, müssen transparent sein, dies beinhaltet deren Funktionsweise als auch deren Wartung.
5. Persistenz: Identitäten sollten langfristig sein und im Idealfall für immer bestehen oder zumindest so lange, wie es der Wunsch des Individuums ist.
6. Portabilität: Identitätsbezogene Informationen und Dienste sollten transportabel sein und nicht an ein bestimmtes Ökosystem gebunden sein.
7. Interoperabilität: Identitäten sollten global und plattformübergreifend nutzbar sein, ohne dass der Nutzer die Kontrolle über seine Identität verliert, sowohl über Staats- als auch über Systemgrenzen hinweg.
8. Zustimmung: Die Verwendung einer Identität muss durch das Einverständnis des Identitätsbesitzers bestätigt werden.
9. Minimierung: Die Datenerhebung sollte so gestaltet sein, dass nur das erforderliche Minimum an Informationen offengelegt werden muss, das für die jeweilige Aufgabe notwendig ist.
10. Schutz: Die Rechte der Nutzer müssen gewahrt werden und über den Interessen der Identitätsnetzwerksanbietern stehen.

Für die SSI werden Technologien genutzt, um den dezentralisierten Austausch von Identitätsdaten, Identifikatoren, Beziehungen, sicheren Nachrichten und digitalen Nachweisen zu ermöglichen. SSI ist noch am Anfang der Entwicklung. SSI strebt an, sowohl die Authentifizierung als auch die Autorisierung von Entitäten ortsungebunden, ohne Kenntnisnahme von Dritten und in Echtzeit zu ermöglichen. Authentifizierung bedeutet, dass die Identität einer Person oder eines Objekts durch die Überprüfung vorliegender Aussagen zu dieser Identität nachgewiesen wird. Diesen Nachweis kann das Gegenüber authentifizieren, was es ihm ermöglicht, eine Autorisierung für eine bestimmte Handlung oder Ressource zu erteilen. SSI umfasst verschiedene Komponenten und Mechanismen, die zusammen agieren sollen mit dem Ziel, eine interoperable und vertrauenswürdige Identitätsinfrastruktur bereitzustellen.

2.2 Digitaler Nachweis

Verifiable Credential (VC) Ein Nachweis dient dazu, eine spezifische Behauptung zu belegen. Der digitale Nachweis wird in dieser Arbeit, basierend auf dem etablierten VCDM, als Verifiable Credential (VC) bezeichnet. Das VC ist ein Datencontainer, der die Information trägt, die zur Authentifizierung der entitätsbezogenen Behauptung benötigt wird. Ein VC wird von einem Issuer für eine Entität, den Holder, ausgestellt.

Identity-Wallet Identity-Wallets sind Softwarearchitekturen, die es Benutzern ermöglichen, ihre digitalen Identitätsdaten zu verwalten und damit in SSI-Ökosystemen zu interagieren. Diese werden üblicherweise als Apps oder Softwareprogramme bereitgestellt. Laut Preukschat setzen sich Identity-Wallets aus verschiedenen Komponenten zusammen, darunter Agents und Secure-Storages [PR21, S. 216]. Secure-Storage, der in anderen Quellen auch als Wallet bezeichnet wird (vgl. [Win23, §18]), dient der sicheren Aufbewahrung von Identitätsinformationen wie VCs und kryptografischen Schlüsseln. Agenten hingegen sind Softwaremodule, die die Kommunikation und Interaktion mit anderen Agenten und Systemen im SSI-Ökosystem übernehmen [Win23, §18]. Sie führen Aufgaben wie den Versand und Empfang von Nachrichten, die Verwaltung kryptografischer Funktionen, die Pflege von Kontakten und das Routing von Nachrichten zwischen den beteiligten Akteuren durch [PR21, S. 217].

Claim Die Behauptung, die ein VC transportiert, bezeichnet man als Claim. Ein VC kann mehrere Claims oder nur einen Claim enthalten. Ein Claim bezieht sich auf eine Person oder ein Objekt, über das eine Aussage getroffen wird. Die Aussagekraft eines Claims hängt von verschiedenen Faktoren ab, etwa ob die Attestierung des Issuers als glaubwürdig eingestuft werden kann oder nicht. Beziehungen zwischen dem Objekt, seinen Eigenschaften und Werten dienen als Ausdrucksmittel für eine Behauptung (vgl. [SLC23, §3.1]). Folgend werden Beispiele für Claims gelistet:

- Die Meldebehörde, der Issuer, attestiert, dass diese Person, der Holder, derzeit an der angegebenen Adresse wohnhaft und dies ihre offizielle Adresse ist.
- Die Hochschule, der Issuer, attestiert, dass diese Person, der Holder, einen Master-Abschluss in Informatik von der Hochschule Mittweida erhalten hat.

Attribut Ein Claim setzt sich aus einem oder mehreren Attributen zusammen. Im obigen Beispiel besteht der Claim aus den Attributen „Bildungseinrichtung: Hochschule Mittweida", „Abschluss: Master“ und „Name der Absolventin: Maxine Musterfrau“ . Laut Duden ist ein Attribut eine charakteristische Eigenschaft oder ein Merkmal, das einer Person oder Sache zugeordnet wird (vgl. [Duda]). Innerhalb von VCs stellt das Attribut die kleinstmögliche Einheit einer Information dar, dargestellt als Datensatz-Paar besteht es aus einer Eigenschaft und ihrem zugehörigen Wert. Beispielsweise das Datensatz-Paar mit der Eigenschaft „Körpergröße“ und dem dazugehörigen Wert „170 cm“.

Verifiable Presentation (VP) Eine Verifiable Presentation (VP) ist eine Ableitung aus einem Credential. Sie wird vom Holder aus dem VC für die Verifikation mit dem Verifier erstellt. Der Verifier kann anhand der VP die Herkunft und Integrität der Behauptung überprüfen. Der Verifier entscheidet, ob er dem Claim, den ein Issuer einem Holder attestiert, Glauben schenkt. Ist dies der Fall folgt auf den Informationsaustausch eine Handlung, wie Zutritt zu einem Gebäude.

Schema Ein Credential-Schema dient als Vorlage für ein VC. Es gibt die Struktur und Attribute eines bestimmten Credential-Types vor, wie ein Mitarbeiter-Credential, der die Attribute Vornamen, Nachname, Adresse und Arbeitgeber beinhaltet. Die Verwendung von Credential-Schemata ermöglicht es, einheitliche VCs eines bestimmten Typs auszustellen und somit die Kommunikation und Verarbeitung von Daten durch definierte Strukturen zu erleichtern.

Decentralized Identifier (DID) Ein Decentralized-Identifier (DID) repräsentiert eine neuartige Kategorie von Identifikatoren, die auf URL-Basis konzipiert sind und einer Entität zugeordnet werden. DIDs zeichnen sich durch ihre Verifizierbarkeit und Selbstsouveränität aus, wodurch die zugehörige Entität die Kontrolle über den Identifikator behält, ohne auf eine zentrale Vergabestelle angewiesen zu sein (vgl. [Men22; SLC23, 4, §2]). Die DID wird über die Spezifikation [Spo+22] dokumentiert. DIDs können in VCs eingesetzt werden, um einer Entität einen entsprechenden Claim zuzuweisen.

Datenformat Für VCs können verschiedene Datenformate eingesetzt werden, abhängig vom verwendeten Credential-Format. JSON[1] ist ein weit verbreitetes Datenformat, das von vielen Credential-Formaten unterstützt wird und deshalb maßgeblich in vorliegender Arbeit betrachtet wird. Attribute werden als Datenpaare dargestellt. Das JSON-Datenformat wird durch die ECMA[2] spezifiziert, wie in [ECM17] beschrieben. JSON-LD[3] JSON-LD ist eine Erweiterung des JSON-Datenformats. Es ermöglicht die Repräsentation von Resource-Description-Framework (RDF)-Triples, die aus einem Subjekt, Prädikat und Objekt bestehen. Wie Abbildung 2.1 zu entnehmen ist, lässt sich folgender Claim ableiten: Der Holder, repräsentiert durch *did:example:b3...23*, trägt den Vornamen *John*, da dem Identifikator *did:example:b3...23* der Vorname *John* durch das Prädikat *schema:givenName* zugeordnet wird.

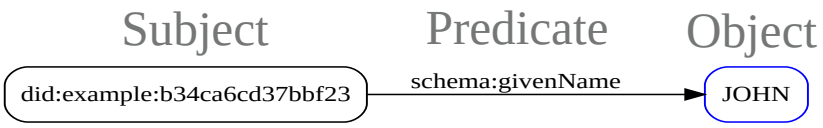

Abbildung 2.1: Datenrepräsentation eines RDF-Triples [4]

Die Definition von JSON-LD erfolgt durch die W3C-Spezifikationen [Gre+20a; Lon+20; Dav+20]. Verschiedene Datenformate wie JSON-LD und N-Quads, als Erweiterung des Formats Turtle, können verwendet werden, um Triples darzustellen (vgl. Quelltext 2.1 und 2.2).

```
{
 "@context": "https://schema.org/",
 "@id": "did:example:b34ca6cd37bbf23",
 "givenName": "JOHN"
}
```

Quelltext 2.1: JSON-LD: Attribute givenName

```
@prefix schema: <https://schema.org/>.
<did:example:b34ca6cd37bbf23>
schema:givenName "JOHN" .
```

Quelltext 2.2: N-Quads: Attribute givenName

[1] JavaScript Object Notation (JSON)
[2] European Computer Manufacturers Association(ECMA)
[3] JavaScript Object Notation for Linked Data (JSON-LD)
[4] Die Visualisierungen in Abbildung 2.1 und alle folgenden Graph-Visualisierungen wurden mithilfe des Tools `https://semantechs.co.uk/turtle-editor-viewer/` erstellt.

Spezielle Algorithmen wie Expanded und Compacted werden bei der Verwendung von JSON-LD eingesetzt, um eine Abwärtskompatibilität mit JSON und dessen Lesbarkeit zu gewährleisten. JSON-LD ermöglicht die Erstellung von Linked-Data-Verknüpfungen, die die Basis des Semantic-Web bilden [Gre+20a; Kel23].

Vokabular World-Wide-Web-Consortium (W3C) definiert Vokabular im Kontext des Semantic-Web als Konzepte und Beziehungen, die zur Beschreibung und Repräsentation eines bestimmten Interessenbereichs verwendet werden (vgl. [W3C19]). Vokabularien werden verwendet, um Begriffe und Terminologien eines bestimmten Kontextes zu definieren, also zu klassifizieren, mögliche Beziehungen zu charakterisieren und bestehende Limitierungen für deren Verwendung festzulegen. Ein kontrolliertes Vokabular stellt in der Regel eine Liste von Fachbegriffen und deren Bezeichnungen aus einer bestimmten Domäne, beispielsweise Krankheitsbilder: Diabetes, Hepatitis oder Krebs, dar [Hop20, S. 115].

Semantic Web Das Semantic-Web, ein von der W3C unter der Feder von Tim Berners-Lee seit 2001 entwickeltes Technologie-Stack, zielt darauf ab, ein „Web of Data“ zu etablieren (vgl. [BH01]). Durch die Erweiterung des bestehenden Webs um semantische Kontextinformationen sollen, wie in [W3C15] dargelegt, Maschinen befähigt werden, effektivere und vertrauenswürdige Interaktionen im Netzwerk selbsttätig durchzuführen. Der Technologie-Stack des Semantic-Webs gründet auf der Nutzung verknüpfter Daten und wird durch Technologien wie RDF, SPARQ[5], Web-Ontology-Language (OWL) und SKOS[6] ermöglicht. Diese Technologien erlauben es, Daten mit Kontext zu hinterlegen und sie somit semantisch im Web zu repräsentieren, Vokabularien und Ontologien zu entwickeln und Regeln für den Umgang mit Daten festzulegen (vgl. [W3C15]).

Ontologie Ontologien und Vokabularien lassen sich sprachlich nicht immer eindeutig voneinander abgrenzen. Bei komplexeren semantischen Modellen spricht man in der Regel von einer Ontologie, während bei einfacheren Strukturen der Begriff Vokabular verwendet wird (vgl. [Hop20; W3C19, S. 115]). Ontologien basieren auf einer einvernehmlichen, formalen Konzeptualisierung und bestehen aus einem semantischen Netz von Begriffen und deren Beziehungen, welches deren inhärente Logik abbilden kann [Hop20; SS09, S. 118 f., 21]. Im Zusammenhang mit Ontologien und dem Semantic-Web wird häufig auf die Open-World-Assumption (OWA) verwiesen mit dem Ansatz fehlende Informationen als unbekannt und nicht als negativ zu interpretieren. Dies ermöglicht eine erweiterbare Wissensrepräsentation [Kee13, S. 1567]. Ontologien im Semantic-Web werden häufig durch Anwendung von RDFS[7] [DRB14] und OWL [IIP12] erstellt. Ein Beispiel für eine Ontologie ist die Gene-Ontology, die eine strukturierte und standardisierte Repräsentation von Genen und Genprodukten in verschiedenen Organismen ermöglicht, um Informationen aus unterschiedlichen wissenschaftlichen Disziplinen zu integrieren und damit vergleichbar zu machen [BN20, S. 108].

Um das Prädikat in RDF-Triples einheitlich zu definieren, werden etablierte Vokabulare und Ontologien wie schema.org, Dublin-Core, GoodRelations oder FOAF[8] verwendet. Diese domänenspezifischen Wissensrepräsentationen bieten unter IRIs[9] Definitionen und Hierarchiestrukturen unterschiedlichen Detaillierungsgrades für Begriffe und deren Kontext an, beispielsweise ist der Begriff

[5] SPARQL Protocol and RDF Query Language (SPARQL)
[6] Simple Knowledge Organization System (SKOS)
[7] Resource-Description-Framework-Schema (RDFS)
[8] Friend of a Friend (FOAF)
[9] Internationalized-Resource-Identifier (IRI) nach (RFC 3987 [DS05])

givenname einer Person in *schema.org* zugeordnet. Menschen können diese Definitionen aufrufen und die Bedeutung interpretieren, während Maschinen RDF-Triples durch die eindeutigen IRIs Objekte einem Subjekt zuordnen können. Ein Literal in RDF ist ein Element, das zur Darstellung von Datentypen verwendet wird. Es besteht aus einer lexikalischen Form, einer Datentyp-IRI, die den Abbildungsmechanismus von der lexikalischen Form zum Literalwert bestimmt und gegebenenfalls einen Sprach-Tag beinhaltet (vgl. [RDM14, §3.3]).

3 Das Schema im digitalen Ökosystem

Der Austausch von VCs und VPs findet innerhalb eines digitalen Identitätsökosystems statt, wobei eine stabile Vertrauensinfrastruktur für die Funktion dieses Systems entscheidend ist. Um Vertrauen im digitalen Raum zu ermöglichen und abzubilden, ist die Implementierung und Etablierung umfassender Strukturen erforderlich.

3.1 Trust-over-IP (ToIP)

Trust-over-IP (ToIP) ist ein Modell, das ein digitales Identitäts-Ökosystem beschreibt. Genauer ist es der Versuch, über die Bau- und Bestandteile eines Systems das Vertrauensproblem zu lösen. Im SSI-Kontext erfährt dieses generelle Modell eine hohe Gewichtung.

Der Gesamtkontext für ein vertrauenswürdiges digitales Ökosystem wurde von der Trust-over-IP (ToIP)-Foundation in der Architektur ToIP-Stacks definiert, dargestellt mittels eines Schichtenmodells. Der ToIP-Stack stellt neben dem PKI[10] und analogen Ansätzen einen Lösungsvorschlag für das brisante Problem mangelhafter Vertrauensinfrastrukturen dar. Die Schichten sollen, wie der TCP/IP-Stack für das Internet, durch Standardisierung von Protokollen das Problem der Interoperabilität in der Kommunikation innerhalb von Vertrauensinfrastrukturen lösen. Die Protokolle sollen den Informationsaustausch und die Zusammenarbeit in den Vertrauensinfrastrukturen ermöglichen. Der ToIP-Stack fußt auf den Prinzipien der SSI, die von Christopher Allen postuliert wurden (vgl. [All16]).

Die ToIP-Foundation wird von der Linux Foundation unter der Rechtsstruktur der Joint-Development-Foundation betrieben. Zum Lenkungsgremium[11] zählen Unternehmen wie Avast, CVS Health, Accenture, esatus und weitere sowie die Regierung von British Columbia. Die ToIP-Stack-Architektur dient als Rahmenwerk, um ein verallgemeinertes Verständnis für die unterschiedlichen Aspekte der Vertrauensbildung, der Prozesse und Rollen in einem digitalen Ökosystem zu schaffen (vgl. [RS21b]). Dafür werden kryptografisches Vertrauen auf der Maschinenebene mit menschlichem Vertrauen auf der geschäftlichen, rechtlichen und sozialen Ebene kombiniert ([PR21, S. 39]; vgl. [RS21a; RP21; BCO22]). Gegliedert wird die Architektur zum einen vertikal in zwei Spalten, eine technologische und eine gesellschaftliche, und zum anderen horizontal in vier durchgängige Schichten, Layer 1-4 (vgl. Abbildung 3.1).

Layer 1 Utility: Die unterste Schicht Layer 1 stellt den sogenannten Root-of-Trust dar, auf dem die nachfolgenden Schichten aufbauen. Kernkomponente der Schicht ist eine Verifiable-Data-Registry (VDR) (Utility in der Abbildung 3.1), auf der relevante Informationen wie öffentliche Schlüssel oder Credential-Schemata für die Vertrauensinfrastruktur abgelegt und überprüft werden können. Welche Eigenschaften, z.B. der Dezentralisierungsgrad, eine VDR für den Einsatz in einer Vertrauensinfrastruktur aufweisen muss, ist von den Regularien, dem Utility-Framework, die durch die Governing-Authority aufgestellt werden, abhängig. Ebenso, ob mehrere VDRs in einem Ökosystem eingesetzt werden. Als VDR kann eine Blockchain, DLT[12], Content-Addressable-Storage (CAS) wie

[10] Public-Key-Infrastruktur (PKI)
[11] `https://trustoverip.org/about/members/`
[12] Distributed-Ledger-Technologie (DLT)

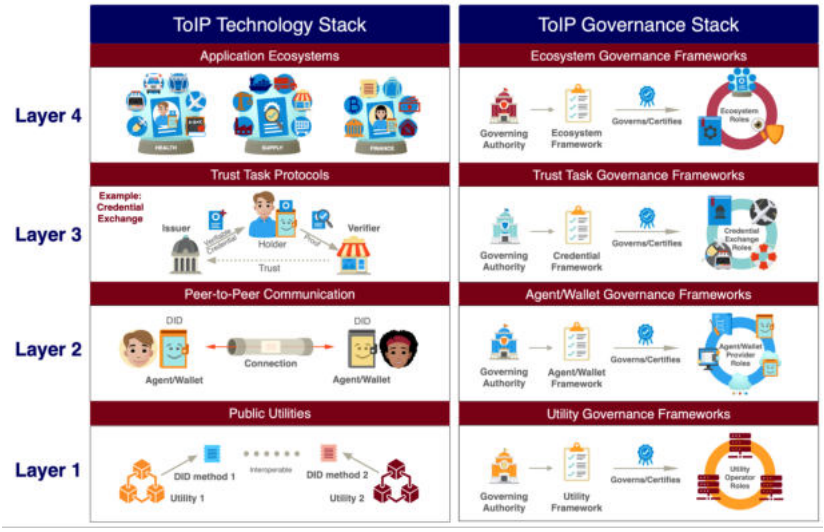

Abbildung 3.1: ToIP-Stack [RS21b]

Inter-Planetary-File-System (IPFS), oder eine Versionskontrollsoftware wie GIT, eingesetzt werden (vgl. [Cap+22, S. 16]; [PR21, 279–281]). Die Governing-Authority legt darüber hinaus Rollen und deren Rechte fest, die sie in Verbindung mit der VDR einnehmen.

Layer 2 Agent/Wallet: In der zweiten Schicht werden benötigte Aspekte wie Sicherheits-, Privatsphäre-, Datenschutz- und Interoperabilitätsstandards für digitale Wallets sowie digitale Agenten und deren Kommunikation untereinander festgelegt. Kernkomponenten der Schicht sind digitale Wallets und Agenten, die von Dingen, Personen oder Organisationen benötigt werden, um Credentials über ein standardisiertes P2P[13]-Protokoll wie DIDComm zu akzeptieren, zu speichern und auszutauschen (vgl. [RS21b]).

Layer 3 Credential: Die dritte Schicht stellt den Übergang von technischem Vertrauen zu menschlichem Vertrauen dar und wird als Identity-Data-Exchange bezeichnet [PR21, S. 104]. Zentrales Konzept dieser Schicht ist das Trust-Triangle, Vertrauensdreieck, das die Rollen Issuer, Holder und Verifier über ihre Vertrauensbezüge in den Kontext von SSI einbettet (vgl. [SLC23]). Die Regelungen für diese Schicht werden im Credential-Framework (CF) festgelegt. Das Trust-Triangle ist auf physische und digitale Vertrauensbeziehungen anwendbar, was die digitale Transformation von physischen, papierbasierten, Nachweisen erleichtert ([PR21, S. 24]). Vertrauensmedium stellt dabei das Credential und seine Präsentation dar (siehe Abschnitt 5). Ein Schema kann für das Ausstellen eines Credentials genutzt werden, um eine Struktur, die in Layer 4 für eine Domäne vorgegeben werden kann, zu definieren.

Layer 4 Ecosystem: In der vierten Schicht des ToIP-Stacks werden die Rahmenbedingungen definiert, die für die Steuerung eines SSI-Ökosystems benötigt werden und sie wird als Application-Ecosystem bezeichnet. Diese Regelungen werden über ein Ecosystem-Framework (EF) veröffentlicht, das rechtliches, organisatorisches und geschäftliches Verhalten festlegt. Ob die Regelungen zu organisatorischen und technischen Systemen in den unteren Schichten entsprechend implementiert wurden, kann durch Zertifizierung der Governance-Authority, z.B. Regierungen, angezeigt werden. Ein EF kann zu unterschiedlichen Implementierungen unterer Schichten kompatibel sein. Die Einstu-

[13] Peer-to-Peer (P2P)

fung in LOA[14] (vgl. [RS21b]), Stufen eins bis vier, legt das Vertrauensniveau fest, das einer Entität oder einem Credential attestiert wird. In einem EF kann definiert werden, durch welche Bedingungen und anhand welches Schemas ein Credential von wem ausgestellt werden darf. Die Informationen welche Entität, unter welchen Voraussetzungen, oder welches Schema den Regularien eines Trust-Frameworks, wie eIDAS[15], PCTF[16] oder UKDIATF[17], entspricht, können in einem entsprechenden Trust-Registry festgehalten werden. In der aktuellen Version 1.0 des Trust-Registry-Protocols ist das Schemata-Management nicht spezifiziert, dies soll in der Version 2.0 erfolgen (vgl. [OR22]). Darin sollen die einem domänenspezifischen Trust-Framework zugrunde liegenden Schemata hinterlegt oder Quellen aufgelistet werden, in welcher VDR die Schemata abgelegt wurden (vgl. [O D22]).

Schema im ToIP Um deutlich den Fokus auf das Schema im ToIP-Stack zu richten und seine Wirkung zu veranschaulichen, wurde Abbildung 3.2 erstellt. Das Schema wird in der ersten Schicht in einer VDR hinterlegt. Anhand der Definition eines Schemas kann in der dritten Schicht ein Credential ausgestellt werden. Die Regularien an die Credential-Handhabung werden im CF definiert. Welche Implikationen mit dem Schema einhergehen, soll in der vierten Schicht mithilfe einer Trust-Registry, die einem EF zugrunde liegt, definiert werden. Die Unterschiede in Hinblick auf die Credentials und deren Schemata von CF und EF beziehen sich auf den jeweiligen Layer. EF regelt die Anforderungen einer spezifischen Domäne und deren aufbauenden Anwendungsfällen. CF regelt die Anforderungen, die an den Credential und seine Präsentation als Informationsmedium von den Akteuren Issuer, Holder und Verifier für eine Vertrauensbeziehung benötigt werden.

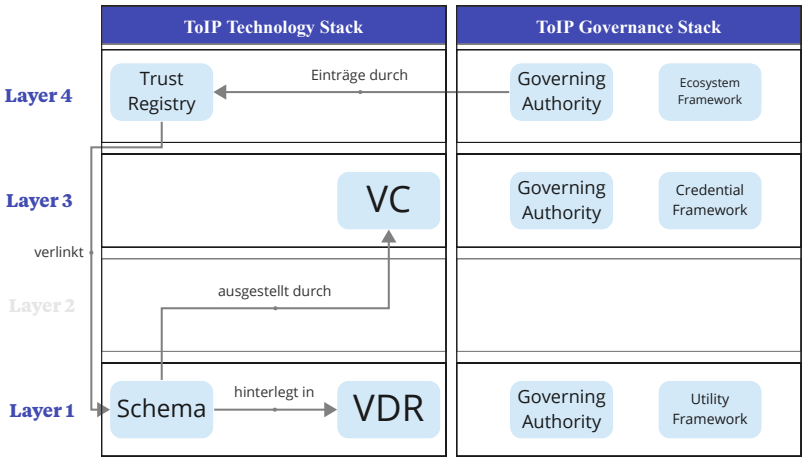

Abbildung 3.2: ToIP-Stack: Fokus auf den Schemaprozess

3.2 Vertrauen und Kontext

Vertrauen manifestiert sich als subjektive Überzeugung eines Menschen, diese basiert auf der Erwartung, dass einer Person oder einem Objekt gegenübergestellte Anforderungen erfüllt werden. Diese Erwartungen können sowohl Handlungen als auch Eigenschaften einer Person oder eines Objekts umfassen. Gemäß der Duden-Definition handelt es sich bei Vertrauen um ein „festes Überzeugtsein

[14] Levels of Assurance (LOA)
[15] Electronic Identification, Authentication and Trust Services (eIDAS)
[16] The Pan-Canadian Trust Framework ™ (PCTF)
[17] UK Digital Identity and Attributes Trust Framework (UKDIATF)

von der Verlässlichkeit und Zuverlässigkeit einer Person oder Sache “[Dudb]. Vertrauen ist in Beziehungen zwischen dem Vertrauenden (A) und dem Vertrauensobjekt (B) verankert, siehe Abbildung 3.3. Das Vertrauen das A B entgegenbringt ist in erster Linie von Kontext abhängig. Der Kontext legt den Rahmen für die Vertrauensentscheidung fest.

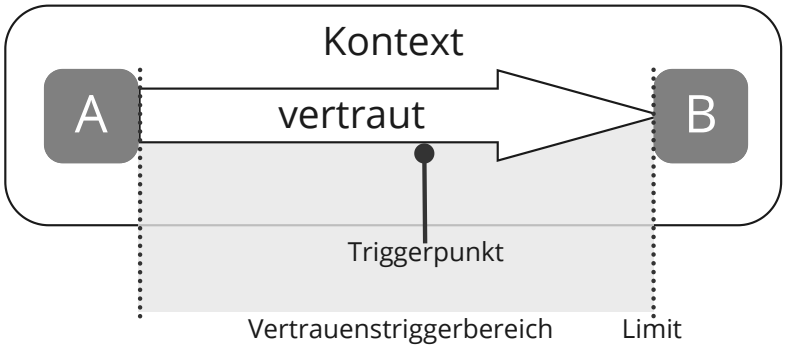

Abbildung 3.3: Vertrauensentscheidung [RS21a, S. 44]

Beispielsweise, wie von Perry [Sco23] beschrieben, könnte ein Erziehungsberechtigter einem Jugendlichen unter Umständen nicht zutrauen, bei sämtlichen anfallenden Haushaltsaufgaben unterstützend tätig zu sein. Jedoch könnte die Betreuung eines Kleinkindes als vertrauenswürdig erachtet werden, vorausgesetzt der Jugendliche hat einen Babysitterkurs beim Deutschen Roten Kreuz absolviert und ein Elternteil hat die Kinderbetreuung des Jugendlichen ohne Zwischenfälle überwacht. Im Kontext der Unterstützung bei der Instandsetzung des Familienfahrzeugs sind hingegen andere Indikatoren erforderlich, um das Vertrauen in den Jugendlichen für diesen spezifischen Kontext zu gewährleisten wie Erfahrung und eine Ausbildung zum KFZ-Mechatroniker. Vertrauen wird demnach nicht in allen Entscheidungssituationen generiert, sondern jeder Kontext wird individuell evaluiert.

Kontext hat eine begrenzte Auflösung. Ob in einem spezifischem Kontext Vertrauen geschenkt wird, hängt von dem Triggerbereich und dem darin liegenden Triggerpunkt des Vertrauenden ab. Der Triggerbereich wird von den Vorerfahrungen, den entsprechenden Fähigkeiten, dem Wohlwollen, der Intergrität und der Neigung des Vertrauensgebers zum Vertrauensobjekt aufgespannt. Ob der Triggerpunkt erreicht wird, um eine positive Vertrauensentscheidung zu fällen, ist abhängig vom wahrgenommenen Risiko und der Risikobereitschaft in der Beziehung (vgl. [MDS95, S. 715 f.]).

Die erforderliche Menge an Information, die für das Fällen einer Vertrauensentscheidung notwendig ist, ist endlich. Sie hat ein Limit. Innerhalb des Vertrauenstriggerbereichs liegt die Schwelle, an der das Treffen einer Vertrauensentscheidung möglich ist. Das Fällen einer Entscheidung ist dann auch notwendig. Mehr Informationsgewinn überwiegt das Nutzen der Entscheidung nicht. In digitalen Identitätsökosystemen liegt der Fokus auf dem Austausch von Informationen und der Schaffung von Vertrauen. Dieser Austausch wird durch die Bereitstellung von Kontext unterstützt. Das Credential fungiert als Träger von Informationen und dient als Austauschmedium. Schema, der zugehörige Credential-Schemaprozess und die damit verbundenen Mechanismen bilden eine Vorlage für die Struktur und den Inhalt dieser Informationen.

4 Anforderungserhebung

In diesem Kapitel werden die Anforderungen für den Credential-Schemaprozess herausgestellt und mithilfe von Informationsqualitätskriterien konkretisiert. Dies soll einen Vergleich verschiedener Ansätze und eine Bewertung der Mechanismen des Schemaprozesses, insbesondere im Hinblick auf Schemata, die sich auf Linked-Data beziehen, ermöglichen. Um die Anforderungen für den Schemaprozess zu präzisieren, werden im Folgenden Use-Cases und beteiligte Akteure des Prozesses aufgestellt und definiert.

4.1 Use Cases

In Abbildung 4.1 werden die Mechanismen und Akteure des Schemaprozesses dargestellt und in der Folge erläutert.

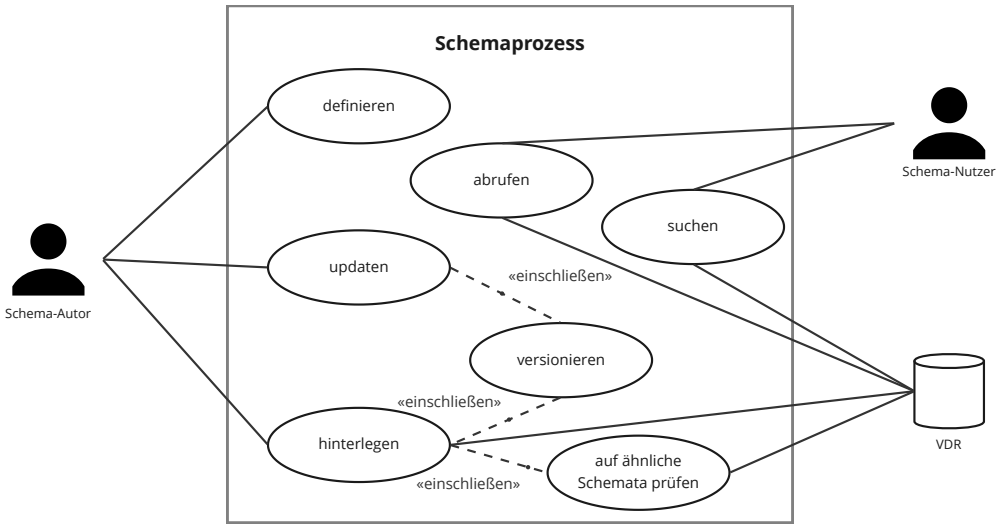

Abbildung 4.1: Use-Case-Diagramm des Schemaprozesses

- Schema-Autor: Ist Akteur im Schemaprozess, erstellt Schemata anhand von Parametern, aktualisiert Schemata und hinterlegt Schemata.
- Schema-Nutzer: Ist Akteur des Schemaprozesses, sucht nach passenden Schemata und ruft Schemata ab.
- Schema-Editor: Softwarekomponente über die der Schema-Autor Schemata definiert.
- VDR: Ist Speicherort für Schemata. Liefert die Schemata-Daten und die Funktion der Hinterlegung.
- Definieren: Durch Erstellen von Typ und Attributen wird das Schema modelliert.
- Hinterlegen: Schema wird in einer VDR abgelegt und auf ähnliche Schemata geprüft.
- Updaten: Neuerstellen eines Schemas durch Updaten bestehender Version.
- Suchen: Potentieller Schema-Nutzer sucht nach passendem Schema anhand von Metadaten.
- Abrufen: Auf Schema mittels eines Identifikators zugreifen.

Schema-Nutzer Als Schema-Nutzer werden alle Akteure bezeichnet, die das Datenschema in einem nachfolgenden Prozess verwenden wie Governing-Authority, Issuer, Holder, und Verifier (siehe Abschnitt 3.1 und folgender).

VDR In der VDR werden die Schemata hinterlegt. Diese Schemata können für weitere Prozessschritte, beispielsweise den Ausstellungsprozess von VCs, von der VDR abgerufen werden (siehe Abschnitt 3.1).

Schema-Editor Über den Schema-Editor können Datenschemata erstellt werden. Der Funktionsumfang des Schema-Editors hängt sowohl von der Zielgruppe als auch vom Format der Credentials ab. Schemata, die etablierte Standards abbilden, oder individuelle Schemata, die von einer kleinen Nutzergruppe genutzt werden, resultieren jeweils in unterschiedlichen Anforderungen an den Schemaprozess.

Schema-Autor Der Schema-Autor ist für die Erstellung von Schemata und deren Verwaltung zuständig. Die Rolle des Schema-Autors kann prinzipiell von jeder physischen oder juristischen Person eingenommen werden. Üblicherweise tritt als Schema-Autor eine Standardisierungsorganisation wie IATA[18], DCSA[19] oder XÖV[20] auf, um eigene Standards durch ein Schema zu etablieren.

Credential-Prozesskette Die Credential-Prozesskette gliedert den Ablauf der Ausstellung und Verifikation von VC und zugehörigem VP. Vor dem Ausstellungsprozess veröffentlicht der Schema-Autor ein Schema in einer VDR, das anschließend der Issuer verwendet, um ein Credential gemäß der Vorlage auszustellen. In Abbildung A.1 wird die generalisierte Credential-Prozesskette durch ein sequenzielles Diagramm dargestellt. Die Anwendung eines Schemas in der Credential-Prozesskette, ob obligatorisch oder optional, hängt sowohl vom gewählten Credential-Format als auch vom jeweiligen Anwendungsfall ab. Der Issuer bestimmt und attestiert die in einem VC enthaltenen Daten. Die Schema-Nutzer beziehen sich auf das Schema, um eine Referenz für den Informationsaustausch eines definierten Anwendungsfalls zu ermöglichen. Der Holder nutzt das Schema als Referenz für das erwartete VC, während der Verifier das Schema verwendet, um eine Anfrage für eine VP zu erstellen und die empfangene VP bzw. enthaltene VCs mit dem Schema abzugleichen.

Definition als Mechanismus: Die Mechanismen zur Definition von Credential-Schemata sollen die Erstellung von Schemata systematisieren und konsistent gestalten. Dies umfasst die Bereitstellung einer definierten Vorlage für Credential-Schemata, die Festlegung der inhaltlichen Struktur von Credentials sowie die Einbindung von Kontextinformationen für das Schema und das Credential. Dadurch sollen Schema-Autoren in der Lage sein, präzise syntaktische und semantische Schemata für VCs zu erstellen.

Hinterlegung als Mechanismus: Die Hinterlegung von Credential-Schemata in einer VDR soll durch einen Mechanismus realisiert werden, der es dem Schema-Autor ermöglicht, die Speicherung und Veröffentlichung von Informationen durchzuführen, die für die Verwendung und Verarbeitung von Schemata für VCs erforderlich sind.

[18] International Air Transport Association (IATA)
[19] Digital Container Shipping Association (DCSA)
[20] XML in der öffentlichen Verwaltung (XÖV)

Abruf als Mechanismus: Ein Mechanismus zum Abruf von Credential-Schemata aus einem VDR soll Schema-Nutzern die Möglichkeit bieten, Schemata für Anwendungsfälle abzurufen. Dabei sollen sowohl Schema-Nutzer als auch -Autoren in der Lage sein, nach bestehenden Schemata zu suchen.

Versionierung als Mechanismus: Die Versionierung von Credential-Schemata soll mithilfe eines Mechanismus realisiert werden, der es ermöglicht, Änderungen und Aktualisierungen im Laufe der Zeit nachzuverfolgen und bereitzustellen. Schema-Nutzer sollen dadurch Änderungen am Schema erkennen können und die Möglichkeit haben, auf ältere Versionen zurückzugreifen, während Schema-Autoren die Möglichkeit erhalten, Schemata weiterzuentwickeln.

Domänenspezifisches Anwendungsszenario Im Tauchsport gibt es zwei große Organisationen, PADI[21] und Scuba-Schools-International, die allgemeine Standards für diese Domäne festlegen. PADI und Scuba-Schools-International würden im SSI-Kontext als Governing-Authority bzw. Schema-Author auftreten oder ein externes Schema als domänenkonform deklarieren. Schemata für Tauchzertifikate wie Open-Water-Diver, Advanced-Open-Water-Diver oder Enriched-Air-Nitrox-Diver würden von den Schema-Autoren veröffentlicht. Nach Akkreditierung durch eine Organisation erhielte eine Tauchschule die Kompetenz, nach einem der Organisation zugeordneten Schema Credentials auszustellen. Ein Tauchschüler erhält nach erfolgreichem Abschluss eines Tauchkurses ein Credential, das von der Tauchschule auf seinen Namen ausgestellt wird. Durch Vorlage des Credentials bei einer von der Organisation anerkannten Tauchschule erhält der Taucher Zugang zu Dienstleistungen und Ausrüstung. Ohne die Struktur und das Fachwissen einer Standardisierungsorganisation wie PAID und Scuba-Schools-International wäre ein weltweiter Konsens über das Ausbildungsniveau und die Fähigkeiten von Tauchern nicht möglich, das schafft Sicherheit für Tauchschulen und Taucher.

Im Kontext von SSI kann diese Standardisierung durch die Definition domänenspezifischer Schemata erreicht werden. Aktuell sind PADI und Scuba-Schools-International zentralisierte Organisationsstrukturen. Vertrauensbindung zwischen den Parteien erfolgt über analoge Strukturen. Um domänenspezifische Strukturen über ein SSI-Ökosystem abzubilden, werden Ansätze wie das ToIP entwickelt.

4.2 Information

Der Kontext, in dem eine Vertrauensentscheidung durch VC und/oder VP getroffen wird, hängt von den zum Kontext zuortenbaren Claims ab. Um die Wirksamkeit dieser Claims in verschiedenen Kontexten zu erhöhen, müssen sie vom Vertrauensgeber im jeweiligen Kontext als Information verstanden werden. Credential-Schemata dienen als Vorlage für Credentials, die als Träger dieser Informationen fungieren. In Abbildung 4.2 wird Information mit der Fähigkeit des Empfängers gleichgesetzt, Daten als Handlungsinstruktion des Senders zu interpretieren, in Anlehnung an [Hei96, S. 14].

Laut Weber sind Informationen dann mit Wissen gleichzusetzen, wenn sie miteinander vernetzt werden bzw. interpretiert werden können [WK20, S. 8]. Eine Information wird von Sender zu Empfänger übermittelt. Für das Verständnis dieser Information ist ein Kontext, auf den sich Sender und Empfänger gemeinsam beziehen, notwendig (vgl. [WC66; Hoi54, S. 207 f., 95 f.]). Je weniger Kontext Sender und Empfänger als Basis für einen Informationsaustausch teilen, desto mehr Kontext muss

[21] Professional Association of Diving Instructors (PADI)

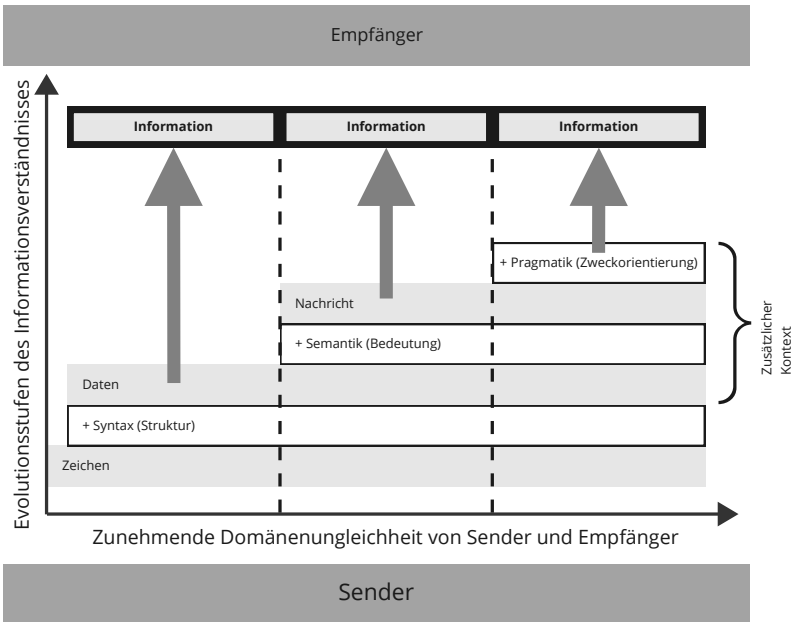

Abbildung 4.2: Information und ihr empfängerbezogener Kontext (in Anlehnung an [Klo11])

mit dem Informationsmedium an den Empfänger übermittelt werden. Die Bedeutung einer Objektrepräsentation steckt nicht in der Repräsentation selbst [EHR15, S. 42]. Die Herausforderung wird gesteigert, wenn die Information nicht an einen spezifischen Empfänger adressiert ist, sondern von einer unspezifischen Empfängergruppe verstanden werden soll. In diesem Fall ist es erforderlich, dass die Information explizit vorliegt oder aus dem Kontext abgeleitet werden kann, wie von Hoope in [EHR15, S. 42] beschrieben. Diese Handlungsinstruktion, also Wissen das zum Erreichen eines Zweckes genutzt wird, muss bei zunehmender Domänenungleichheit von Sender und Empfänger durch zusätzliche Kontextdaten dem Empfänger mit überliefert werden.

Auf der syntaktischen Ebene werden die verschiedenen Zeichen durch formale Strukturen, insbesondere grammatische Regeln, miteinander verknüpft und integriert. Durch diesen Prozess entstehen Daten, die einer konsistenten Syntax folgen, und mit technischen Hilfsmitteln verarbeitet werden können (vgl. [Krc15; WK13, S. 13, 12]). Zusätzlicher Kontext kann durch die Addition von Semantik bereitgestellt werden. Daten werden dadurch zur Nachricht. Ist der Empfänger in der Lage die Bedeutung der Nachricht zu interpretieren, versteht er die Information. Die Verarbeitung der Nachricht durch den Empfänger aktiviert dessen Bewusstsein. Jedoch führt das Verständnis der Nachricht nicht automatisch zu einer Handlung beim Empfänger, da die Nachricht für ihn möglicherweise nicht relevant ist [WK20, S. 7].

Die pragmatische Ebene bezieht sich auf das handlungsleitende Interesse des Senders, um aus einer Nachricht eine Information zu machen, die zur Erfüllung einer bestimmten Aufgabe führt [WK13; Krc15, S. 554, 13]. Dabei spielt das Wissen eine Rolle, da es ermöglicht, Aufgaben in größere Zusammenhänge einzuordnen. Das Interesse ergibt sich aus einem Situationsbezug, bei dem zur Erreichung eines bestimmten Zwecks eine oder mehrere Aufgaben innerhalb einer gewissen Zeit auszuführen sind [Klo11, S. 12]. Die Information unterstützt oder ermöglicht die Aufgabenbearbeitung, indem sie direkt zu einem bestimmten Arbeitsergebnis führt oder zu Handlungen anregt, die zur Aufgabenerfüllung führen. Eine erfolgreiche Kommunikation auf der pragmatischen Ebene erfordert

somit das Verständnis des Empfängers für den Zusammenhang zwischen der Information und der Aufgabe, die es zu erfüllen gilt. „Eine Information kann nur dann entstehen, wenn sie sprachlich kodiert vorliegt und wenn sich ein menschlicher Gedankenträger ihrer bewusst ist“ [WK13, S. 556].

4.3 Informationsqualität (IQ)

Im folgenden Abschnitt werden dem Schemaprozess in Anlehnung an die IQs-Dimensionen der DGIQ[22] IQs-Kriterien, die sogenannten IQs-Dimensionen, zugeordnet, um die Anforderungen an den Schemaprozess zu spezifizieren (vgl. [Roh+21]). Die IQs-Dimensionen sind eine Definitionsmethode für den Begriff Informationsqualität. Informationen werden in 15 IQs-Dimensionen und 4 Kategorien eingeteilt, um ihre Qualität zu bewerten (vgl. [HGM21, S. 25 f.]). In Abbildung 4.3 wurde die Definitionsmethode auf den Schemaprozess projiziert und die spezifischen Akteure Schema-Nutzer/Autor, VDR und Schema eingeführt und den entsprechenden Akteuren zugeordnet.

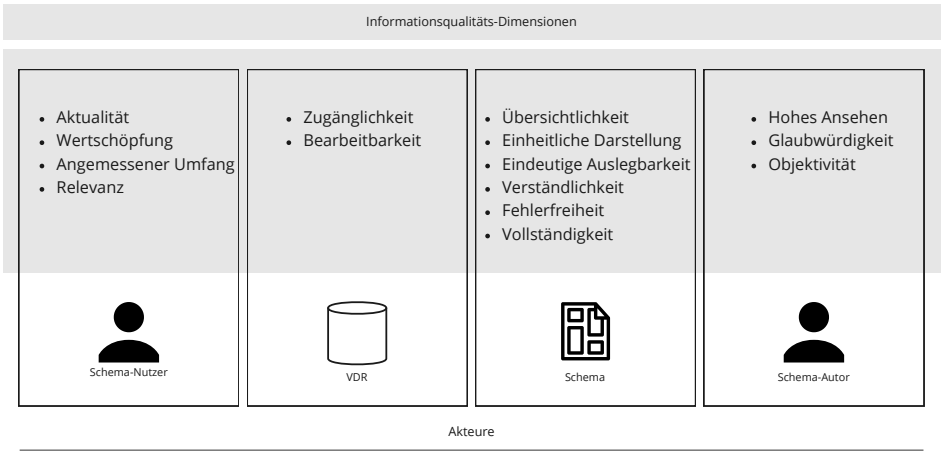

Abbildung 4.3: IQs-Dimensionen den Schema-Akteuren zugeordnet

Schema-Nutzer Die IQs-Dimensionen Aktualität, Wertschöpfung, angemessener Umfang und Relevanz sind von den individuellen Schema-Nutzern zu bewerten. Ob ein Schema für den Ausstellungsprozess eine Wertschöpfung darstellt oder über den angemessenen Umfang verfügt, muss von den Schema-Nutzern bewertet werden. Diese Bewertungsbefähigung kann nur in unregulierten Bereichen stattfinden, bestehen Regularien sind die Entscheidungen des Schema-Nutzers daran gebunden. Regulierungen können entweder von staatlicher oder ökosystemischer Seite vorgeschrieben werden.

VDR Der VDR werden die IQs-Dimensionen Zugänglichkeit und Bearbeitbarkeit zugeschrieben. Die Bearbeitbarkeit bezieht sich im Zusammenhang mit der VDR auf die Kenntlichmachung jeglicher Bearbeitung eines in der VDR hinterlegten Schemas, z.B. über die Erhöhung der Versionsnummer, und den Verbleib der Vorgängerversion des Schemas. Die VDR muss für die weitere Verwendung den Akteuren den Zugang und das dauerhafte Abrufen von Schemata gewährleisten.

[22] Deutsche Gesellschaft für Informations- und Datenqualität (DGIQ)

Schema Das Schema definiert die strukturelle Verifizierbarkeit eines Credentials, indem überprüft wird, ob das Credential vollständig und fehlerfrei ausgefüllt wurde. Das Schema soll eine klare und eindeutige Darstellung aufweisen, die eine einfache Interpretation ermöglicht. Die einheitliche Darstellung ergibt sich aus dem Credential-Format, der Definition von Schemata sowie der Gestaltung der Benutzeroberfläche bei der Ausstellung und Nutzung von Credentials.

Schema-Autor Ob ein Schema Verwendung findet, ist maßgeblich vom Schema-Autor abhängig. Zeichnet sich der Autor durch ein hohes Ansehen und einen hohen Bekanntheitsgrad aus, findet das Schema weite Verbreitung. Um die Verständlichkeit und Nachvollziehbarkeit und damit die Glaubwürdigkeit eines Schemas für die Schema-Nutzer zu steigern, bedarf es einer angemessenen Dokumentation wie einen Regelsteckbrief (vgl. [WK20, S. 122 f.]). Zum Beispiel bildet der Schema-Autor quantifizierbare Ausprägungen der Attribute wie Tagestemperatur in ℃ oder nicht objektive wie heiß und kalt ab.

Metriken für die IQs-Dimensionen In einer Literaturrecherche wurden die zugeordneten IQs-Dimensionen überarbeitet und entsprechenden Metriken zugeordnet. Die initialen Suchbegriffe *(schema OR template) AND ("Data Quality")* wurden später präzisiert zu *(("Data schema") OR ("Data template")) AND ("Linked Data Quality")*, was zu 82 Ergebnissen aus der Datenbank Google Scholar führte. Nach Entfernung von Duplikaten und einer ersten Sichtung wurden 41 Veröffentlichungen als relevant eingestuft. Die ausgewählten Veröffentlichungen umfassen [Can+23; NBL22; Bat+22; Iss+21; Zav+15; Bat+15; Wil+16; Beh+14; BS14]. Zusätzlich wurden die Veröffentlichungen [All16; W3C13; NR00] für die Metrikenanalyse betrachtet. Zusätzliche Klassifizierungen von Informationsqualitätsdimensionen wurden in den Veröffentlichungen [ZJL21; NBL22] festgelegt. In Tabelle B.1 wurden die relevanten IQs-Dimensionen und zugehörigen Metriken zur Beurteilung der IQ von Schemata dargestellt und durch die erhobenen Daten aus der Literaturrecherche gespeist. Nicht alle zuvor aufgestellten Dimensionen wurden in die Tabelle integriert, sondern stattdessen jenen Dimensionen zugeordnet, die den Bewertungsrahmen besser repräsentieren. Im Hinblick auf die einheitliche Auslesbarkeit und Fehlerfreiheit wurde die semantische Genauigkeit berücksichtigt. Sicherheit wurde in Bezug auf Glaubwürdigkeit und Bearbeitbarkeit betrachtet. Leistung wurde im Kontext der Zugänglichkeit bewertet, während Minimalismus dem angemessenen Umfang zugeordnet wurde. Interoperabilität, Lizenz und Verknüpfung wurden in Verbindung mit eindeutiger Auslegbarkeit und Verständlichkeit assoziiert. Syntaktische Genauigkeit und offene Formate wurden der einheitlichen Darstellung zugeschrieben. Die einheitliche Darstellung umfasst die Darstellungselemente des Credential-Formats, der Vokabulare und des Credential-Schemas, welche mit zur Übersichtlichkeit beitragen.

Minimalismus In dieser Arbeit wurde die IQ-Dimension des Minimalismus als Indikator für die Qualität von Schemata eingeführt und durch die Integration der Metriken M2 und M3 für das SSI spezifisch erweitert (siehe Tabelle 4.1). Datenminimalismus wird erreicht, wenn eine Realität mit einem Minimum an Information verkörpert wird (vgl. [Bat+15; BS14]).

Die Datenminimierung in Bezug auf die Schemadefinition hängt sowohl von den technischen Möglichkeiten des Credential-Formats als auch von der für einen bestimmten Anwendungsfall erforderlichen Datenerhebung ab, die durch das Credential-Schema vorgegeben werden. Der Autor eines Schemas ist verantwortlich für das Schaffen der Grundlage des Minimalismus, wobei die Einhaltung rechtli-

Tabelle 4.1: Metriken des Minimalismus

IQ-Dimension	Abk.	Metrik	Typ	Akteur
Minimalismus	M1	Ist die Datenerhebung auf das Minimum der jeweiligen Anwendung reduziert (Angemessener Umfang) [WS96; NR00; All16; Roh+21]	QL	Schema
	M2	Verwendung von Methoden zur selektiven Offenlegung von Daten (Selective-Disclosure) [All16]	QN	Format
	M3	Verwendung von Methoden zur binären Präsentation von Daten (Predicate-Proofs) [All16]	QN	Format

cher Vorgaben wie der DSGVO[23] erforderlich ist. Der Nutzer des Schemas wendet es an, während der Endnutzer vom Minimalismus profitiert, da er nur die wesentlichen Teile seiner Daten mit dem Verifizierer teilen muss. Allerdings stellt der Endnutzer das schwächste Glied in der Prozesskette dar (siehe Abbildung A.1 im Anhang).

Der Grad der Datenminimierung im Verifizierungsprozess wird von verschiedenen Faktoren beeinflusst, wie z.B. der Art des Anwendungsfalls und den technologischen Möglichkeiten zur Offenlegung von Informationen. Um den Grundsatz der minimalen Datenoffenlegung zu erfüllen, wie ihn Allen beschrieben hat, können Predicate-Proofs oder Selective-Disclosure verwendet werden. Alternativ kann auch die Verwendung von Atomic Credentials in Erwägung gezogen werden, welche Verifizierungsnachweise darstellen, die nur aus einem Attribut bestehen. Predicate-Proofs ermöglichen es, binäre Zustände aus einem VC kryptografisch abzuleiten und dies als VP zu verwenden, zum Beispiel ob eine Person über 18 Jahre alt ist, kann binär mit *true* oder *false* dargestellt werden. Selective-Disclosure ermöglicht es, dem Holder durch kryptografische Verfahren nur ausgewählte Attribute eines VC an den Verifier in einer VP zu übermitteln, z.B. verfügen AnonCreds über diese Funktionalität (vgl. Abschnitt 5.1).

Minimalismus ist insbesondere relevant bei Persönlich identifizierbare Informationen (PII). Unter PII versteht man Informationen, die dazu genutzt werden können, eine Person zu identifizieren oder ihre Identität offenzulegen (vgl. [SLC23, §7.8]). Schema-Autoren müssen genannte Aspekte berücksichtigen, um ein minimalistisches Schema zu erstellen (siehe Tabelle 4.1). Die Bewertung, ob ein Schema den angemessenen Umfang erfüllt, muss spezifisch von den Schema-Nutzern für deren Anwendungsfälle in unregulierten Bereichen erfolgen und kann nicht allgemeingültig bestimmt werden. Durch optionale Attribute kann der Ermessensspielraum des Schema-Nutzers auf Kosten der einheitlichen Darstellung erhöht werden.

Fehlerfreiheit Die IQ-Dimension der Fehlerfreiheit resultiert aus dem Vergleich verschiedener Informationen und deren jeweiliger Genauigkeit. Der Begriff Genauigkeit wird von der ISO 8000:130[ISO16] Norm als Indikator definiert, der die Exaktheit der erfassten Daten mit dem Datenursprung, der Realität, angibt. Mit anderen Worten: Daten sind fehlerfrei, wenn sie im Rahmen ihrer Näherung, der Genauigkeit, mit der Realität übereinstimmen (vlg. [WK20; HGM21, S. 86, 55 f.]). Im Rahmen des Ausstellungsprozesses beglaubigt der Aussteller die Claims über eine Entität. Als Medium für diese Form der Beglaubigung wird das VC verwendet. Die Fehlerfreiheit bezieht sich auf die Übereinstim-

[23] Datenschutz-Grundverordnung (DSGVO)

mung der im VC erfassten Daten mit seinem Schema und der Realität. Faktoren für die Fehlerfreiheit in Bezug auf das Schema sind dabei die Einhaltung des Umfangs, der Struktur und die Genauigkeit der Validierung der Attributsausprägungen.

Ein Bewertungsrahmen für den Schemaprozess und seine Mechanismen wurde mithilfe der IQs-Dimensionen erstellt, auf dessen Grundlage im nachfolgenden Kapitel die Bewertungen der zu untersuchenden Credential-Formate vorgenommen werden.

5 Credential-Formate

Ein Credential-Format ist das Datenmodell eines Credentials. Es legt die Struktur der gespeicherten Information fest. Darüber hinaus spezifiziert es die eingesetzten kryptografischen Signaturverfahren, die für die Beweisführung des Credentials eingesetzt werden, und die Mechanismen für das Informationsmanagement von VCs und deren VPs. Derzeit wird an über 10 verschiedenen Implementierungen und deren Spezifikationen von Formaten für Credentials gearbeitet, wie in der Credential-Comparison-Matrix [Kud+22] dargestellt. Eine im Jahr 2022 durch die Decentralized-Identity-Foundation (DIF) durchgeführte Umfrage [Dec22a] ergab, dass die am weitesten verbreiteten Formate JSON-JWT, LD-Proofs und AnonCreds waren. Nachfolgend werden diese Credential-Formate beschrieben besonders im Hinblick auf die Anwendung eines Schemas und die damit verbundenen Mechanismen.

5.1 AnonCreds und Hyperledger Indy

AnonCreds Für den Ausstellungsprozess eines Credentials setzt AnonCreds das Verwenden eines Schemas voraus. Das Schema muss in der dem Ökosystem zugehörigen VDR hinterlegt sein, um das Ausstellen eines VCs zu ermöglichen (vgl. [Cur+23, §7]). Die Camenisch-Lysyanskaya Signaturen, die in den AnonCreds verwendet werden, nutzen ein Commit-and-Reveal-Pattern für den Informationsaustausch. Für die Verifikation des Commit-and-Reveal-Patterns wird eine Credential-Definition (CERD_DEF) für die Überprüfung der Datenintegrität benötigt. Genauer, für die Ausstellung nutzt der Issuer ein in der VDR hinterlegtes Schema. Aus diesem Schema wird für jedes Credential eine individuelle CERD_DEF durch den Issuer erstellt. Die CERD_DEF wird anschließend in der VDR hinterlegt, dies ermöglicht die spezifische Zuweisung vom Credential zum Issuer. Die Claims werden beim Ausstellungsprozess verschlüsselt in das Credential geschrieben. Für den Verifikationsprozess nutzt der Verifier entsprechendes Schema und die daraus abgeleitete CERD_DEF, um ein Presentation-Request an den Holder zu stellen. Der Holder übermittelt die angeforderten Informationen an den Verifier. Anhand der Informationen aus der VDR und der vom Holder übermittelten, kann der Verifier die Datenintegrität des Credentials überprüfen (vgl. Abbildung A.1). Das AnonCreds-Schema ist eine JSON-Datenstruktur, welche die folgenden Eigenschaften beinhaltet (siehe Quelltext 5.1):

```
{
        "id": "https://www.did.example/schema.json",
        "name": "Example schema",
        "version": "0.0.1",
        "attr_names": ["name", "age", "vmax"]
}
```

Quelltext 5.1: Datenstruktur des AnonCreds-Schemas

Die Attribute des Schemas werden dabei über den *attr_names* Array von Strings definiert. Mit dieser Datenstruktur werden die Attributsbezeichner der Claims definiert, aber nicht die Datentypen der Attributswerte. Claims werden über ein JSON-Datensatz-Paar gebildet. Eine neue Version eines Schemas kann über die Eigenschaft *version* erstellt und hinterlegt werden. Die entsprechende

Erhöhung der Versionsnummer zeigt die Neuerung an. Die AnonCreds-Spezifikation spricht Empfehlungen, aber keine bindenden Vorgaben für die Hinterlegung, den Abruf und die Versionierung von Schemata aus. Wer ein Schema in die entsprechende VDR schreiben darf, hängt von den Regularien der jeweiligen VDR ab. Je nach Regulierungen erhalten nur spezielle Rollen im Ökosystem das Recht, Schemata auf die VDR zu schreiben. Für AnonCreds wurde vor der Version 1.0, veröffentlicht 2022, die Hyperledger-Indy Blockchain als VDR vorausgesetzt. Für nachfolgende Versionen können auch andere VDRs verwendet werden (vgl. [Cur+23]).

Hyperledger-Indy Hyperledger-Indy ist eine Blockchain, die von AnonCreds als VDR genutzt werden kann, um entsprechende Daten abzulegen. Über die Hyperledger-Indy kann die Struktur der ersten Schicht des ToIP-Stacks abgebildet werden. Eine bekannte Hyperledger-Indy-Implementierung ist das Sovrin-Mainnet, das als public permissioned Blockchain aufgesetzt wurde. Die Blockchain ist öffentlich (public) einsehbar. Die Rollen des Netzwerkes sind an spezifische Regularien, das Trust-Framework, gebunden (permissioned). Eine entsprechende Rollenübernahme setzt Bewerbung und Akzeptanz durch das Netzwerk voraus. Für die Umsetzung der Regularien im Netzwerk sind die Sovrin-Foundation sowie die Rollen Steward und Endorser zuständig. Stewards fungieren als Validator-Nodes des Netzwerkes, sie sind für das Erlangen eines Konsenses im Netzwerk über die Transaktionen in die Blockchain verantwortlich. Sie können ebenfalls die Rolle des Endorsers und des Transaction-Authors einnehmen. Ziel des Sovrin-Netzwerkes ist es ein möglichst breites Spektrum an individuellen Stewards zu erhalten, die eine ausgewogene Repräsentation diverser Rechtsgebiete, Branchen, Größen und Strukturen widerspiegeln sollen (vgl. [Sov]). Zu den Stewards[24] gehören Unternehmen[25] wie dhiway, everynm, monokee und SICPA. Der Endorser hat die Aufgabe, Transaktionen in das Netzwerk zu propagieren. Der Endorser kann auch für Dritte, Transaction-Authors, Transaktionen in die Blockchain schreiben. Voraussetzung dafür ist ein Transaction-Author-Agreement zwischen Endorser und Transaction-Author. Schemata können über den Endorser in die Blockchain geschrieben werden.

Sovrin will das Schreiben in die Blockchain auch als nicht autorisierten Prozess ermöglichen. Der Zweck der Taskforce Public-Write-Access für den öffentlichen Schreibzugang besteht darin, die geschäftlichen und technischen Anforderungen für das Sovrin-Ledger zu entwickeln, um den Übergang vom autorisierten Schreibzugang zum öffentlichen Schreibzugang zu schaffen und so einen weiteren Schritt zu einem deregulierenden (permissionless) Netzwerk zu schaffen (vgl. [Sov20]).

Eine weitere Hyperledger-Indy-Implementierung ist das IDunion-Netzwerk. Im Gegensatz zum Sovrin-MainNet fokussiert sich das IDunion-Netzwerk als rechtliche Gesellschaft nur auf den europäischen Raum, seinen öffentlichen wie privatwirtschaftlichen Sektor (vgl. [IDu]). Das IDunion-Netzwerk ist ein föderiertes Ökosystem, durch Regularien und spezifische Rollen werden Governance-Strukturen etabliert.

Durch eine Blockchain-Transaktion kann ein Schema in Hyperledger-Indy hinterlegt werden. Die Transaktion muss die von AnonCreds definierten Schemaeigenschaften enthalten. [26] (siehe Quelltext 5.2).

```
{
 "txn": {
```

[24] `https://sovrin.org/stewards/`
[25] 8 der 42 Sovrin-Stewards sind auch im Lenkungsgremium der ToIP-Foundation.
[26] `https://indyscan.io/tx/SOVRIN_MAINNET/domain/140373`

```
        "data": {
                "data": {
                        "attr_names": [
                        "Start Date",
                        "Type",
                        "Department",
                        "Gender",
                        "Position",
                        "Birth Date",
                        "Biometric",
                        "Issue Time",
                        "Employee No",
                        "Document Id",
                        "Country",
                        "Issuer",
                        "Organisation",
                        "Full Name"
                        ],
                        "name": "Employee ID",
                        "version": "1.0"
                        }
                },
                "metadata": {
                        "digest": "683b...37c58",
                        "endorser": "3hz...4Hjt",
                        "from": "JSVdVn5x7UE4tcqAP9qv1x",
                        "payloadDigest": "b9dbcf...1f6d282f",
                        "reqId": 1652162084660801500,
                        "taaAcceptance": {
                                "mechanism": "service_agreement",
                                "taaDigest": "8cee5d7...c92879641f",
                                "time": 1652140800
                        }
                },
                "protocolVersion": 2,
                "type": "101",
                "typeName": "SCHEMA"
        },
        "txnMetadata": {
                "seqNo": 140373,
                "txnId": "JSVdVn5x7UE4tcqAP9qv1x:2:Employee ID:1.0",
                "txnTime": "2022-05-10T05:54:45.000Z"
        }
}
```


Quelltext 5.2: Beispiel eines AnonCreds-Schemas, hinterlegt im Sovrin-Mainnet

Der Mechanismus für die Schema-Hinterlegung in die Hyperledger-Indy kann durch eine Transaktion, z.B. durch die Indy-CLI[27], instruiert und via API[28]-Call an die Hyperledger-Indy übermittelt werden. Das Abrufen von Schemata kann über den *GET_SCHEMA* Befehl eingeleitet werden (vgl. [Hyp20]). Die Transaktionen sind in visualisierter Form über einen Blockexplorer öffentlich einsehbar (vgl. [Mi19]). Im Quelltext 5.2 sind die abrufbaren Informationen zu einem Schema, welches in der Blockchain veröffentlicht wurde, zu finden:

[27] command-line interface (CLI)
[28] Application Programming Interface (API)

- DID des Schema-Authors
- Name (Bezeichner) des Schemas und Version
- Attribute des Schemas
- Veröffentlichungsdatum

Die Relevanz eines Schemas innerhalb der Blockchain kann über die Anzahl der CERD_DEF-Transaktionen, im Sovrin-Netzwerk als CLAIM_DEF bezeichnet, ermittelt werden. Die Aktualität eines Schemas kann anhand der Transaktionshistorie überprüft und die Autorenschaft über die DID des Autors nachgewiesen werden.

5.2 Verifiable Credentials Data Model (VCDM)

Das von der W3C spezifizierte VCDM [SLC23] stellt ein Credential-Format dar. Es definiert die Datenstruktur für ein VC und dessen VP. Diese bestehen aus je drei Hauptkomponenten: Metadaten, Claims und Proofs; wobei der Claim der VP dem VC entspricht. Die VCDM-Spezifikation führt ihr Datenmodell mithilfe des Serialisierungsformats JSON und seiner Erweiterung JSON-LD ein. Alternativ können auch Datenformate wie beispielsweise XML[29], YAML[30] oder CBOR[31] für das VCDM Abbilden des Datenmodells eingesetzt werden [SLC23, §6]. Im Folgenden sind die obligatorischen Credential-Eigenschaften sowie eine Auswahl an Optionen aufgelistet.

- @context (obligatorisch): Die Integration von JSON-LD-Kontextdateien erfordert die Verwendung eines JSON-Arrays, bei dem das erste Element die URL `https://www.w3.org/ns/credentials/v2` aufweisen muss.
- id (optional): Kann zur Zuordnung des VCs zu einem Objekt wie einer Person, einem Produkt oder einer Organisation verwendet werden. Es wird empfohlen, eine URL als Identifikator zu verwenden.
- type (obligatorisch): Ein JSON-Array, bei dem das erste Element das *VerifiableCredential* ist. Weitere Elemente im Array sollen den spezifischen VC-Typ definieren wie PermanentResidentCard, BankAccountCredential, VaccinationCertificate.
- issuer (obligatorisch): Bezeichnet den Aussteller des VC und muss entweder eine URL oder ein Objekt mit einer *id*-Eigenschaft aufweisen.
- validFrom (obligatorisch): Gibt den zeitlichen Beginn der Gültigkeit des VC an. Der Wert muss einem String gemäß der XML-Schema-Date-Time [Dav+12, §3.3.7] entsprechen.
- validUntil (optional): Gibt das zeitlichen Ende der Gültigkeit des VC an. Der Wert muss einem String gemäß der XML-Schema-Date-Time [Dav+12, §3.3.7] entsprechen.
- credentialSubject (obligatorisch): Ein JSON-Objekt, das die Claims über ein oder mehrere Subjekte repräsentiert.
- CredentialSchema (optional): Ein JSON-Objekt, das ein Schema für das VC angibt.
- Evidence (optional): Ein JSON-Objekt, das zusätzliche Informationen für eine positive Vertrauensentscheidung bereitstellt wie Verlinkungen zu weiteren Dokumenten.
- termsOfUse (optional): Ein JSON-Objekt, das die Nutzungsbedingungen fasst, die mit dem VC und/oder der VP einhergehen.

[29] Extensible Markup Language(XML)
[30] Yet Another Markup Language (YAML)
[31] Concise Binary Object Representation (CBOR)

- proof (obligatorisch): Ein JSON-Objekt, das die verwendete Signatur für den Nachweis der Datenintegrität und der Autorenschaft enthält, muss die *type*-Eigenschaft des verwendeten Signaturtyps angeben.

Alle Eigenschaften, die weder das CredentialSubject, welches die Claims definiert, noch den Proof betreffen, können als Metadaten betrachtet werden. Ergänzend zu geführter Auflistung wird ein Credential des Typs PermanentResidentCard exemplarisch im Quelltext 5.3 dargestellt.

```
{
        "@context": [
        "https://www.w3.org/ns/credentials/v2",
        "https://w3id.org/citizenship/v1"
        ],
        "id": "https://issuer.oidp.uscis.gov/credentials/83627465",
        "type": ["VerifiableCredential", "PermanentResidentCard"],
        "issuer": "did:example:28394728934792387",
        "validFrom": "2022-12-03T12:19:52Z",
        "validUntil": "2029-12-03T12:19:52Z",
        "credentialSubject": {
                "id": "did:example:b34ca6cd37bbf23",
                "type": ["PermanentResident", "Person"],
                "givenName": "JOHN",
                "familyName": "SMITH",
                "gender": "Male",
                "image": "...kJggg==",
                "residentSince": "2015-01-01",
                "lprCategory": "C09",
                "lprNumber": "999-999-999",
                "commuterClassification": "C1",
                "birthCountry": "Bahamas",
                "birthDate": "1958-07-17"
        },
        "credentialSchema": {
                "id": "https://example.org/examples/Resident.json",
                "type": "CredentialSchema2022"
        },
        "proof": {
                "type": "Ed25519Signature2018",
                "created": "2020-01-30T03:32:15Z",
                "jws": "eyJhbGciOiJFZERTQSIsI...wRG2fNmAx60Vi4Ag",
                "proofPurpose": "assertionMethod",
                "verificationMethod": "did:example:28394728934792387#keys-7f83he7s8"
        }
}
```

Quelltext 5.3: VCDM: Credential des Typs PermanentResidentCard (in Anlehnung [LS20])

Credential-Type Das VCDM verfolgt bei der Erstellung von Credential-Types den Ansatz der OWA. VCs sollen als Linked Data definiert, maschinenlesbar und nicht von einer einzigen zentralen Organisation registriert oder verwaltet werden (vgl. [Cha+19, §10]). Die Erstellung eines eigenen Credential-Types bedarf der Erweiterung des etablierten VCDM. Für die Definition des Typs sind folgende Prozessschritte zu bearbeiten, die Chadwick in den VC-Implementation-Guidelines 1.0 [Cha+19] erläutert.

1. Design the data model.
2. Create a new JSON-LD context.
3. Select a publishing location.
4. Use the new JSON-LD context when issuing new credentials.
   [Cha+19, §10.1]

Bei der Entwicklung eines Credential-Types werden Inhalt und das zu verwendende Vokabular definiert, um den entsprechenden Anwendungsfall abzudecken. Im Abschnitt 6.3.1 wird auf die Erstellung und die Vokabularien eingegangen. Die Auswahl einer VDR, Punkt 3 bei Chadwick, und Punkt 4, Verwendung des erstellten JSON-LD-Kontextes, werden in den Abschnitten zu VDR 5.2 und Proof 5.2 weiter erläutert.

Eine JSON-LD-Kontextdatei definiert Terminologien für einen Credential-Type, auf die sich alle entsprechenden Credentials beziehen. Die Kontextinformationen könnten direkt in die JSON-LD-Datei eingebettet werden, doch die Trennung in eine separate Kontextdatei, abrufbar über die angegebene URL, ermöglicht eine schlankere Repräsentation, verbesserte Lesbarkeit der VCs und das Caching von VC-Types über ihre JSON-LD-Kontextdateien. Im dargestellten Ausschnitt der JSON-LD-Kontextdatei (Quelltext 5.4) für den PermanentResidentCard Credential-Type wird der Kontext über *@context* definiert. Die *@version* gibt an, dass die JSON-LD-Version 1.1 verwendet wird. Mit *@protect* können Warnungen vor Umbenennungen von Terminologien im Credential durch den JSON-LD-Prozessor generiert werden, um die Konsistenz eben dieser zu gewährleisten.

```
{
 "@context": {
        "@version": 1.1,
        "@protected": true,
        ...
        "PermanentResident": {
                "@id": "https://w3id.org/citizenship#PermanentResident",
                "@context": {
                        "@version": 1.1,
                        "@protected": true,

                        "id": "@id",
                        "type": "@type",

                        "ctzn": "https://w3id.org/citizenship#",
                        "schema": "http://schema.org/",
                        "xsd": "http://www.w3.org/2001/XMLSchema#",

                        "birthCountry": "ctzn:birthCountry",
                        "birthDate": {"@id": "schema:birthDate", "@type": "xsd:dateTime"},
                        "commuterClassification": "ctzn:commuterClassification",
                        "familyName": "schema:familyName",
                        ...
        }
```

Quelltext 5.4: Ausschnitt der JSON-LD-Kontextdatei des Credential-Types PermanentResidentCard [LS20]

Am Beispiel von *birthDate* wird die Bildung von Attributen verdeutlicht, wobei die Verwendung von Vokabularien und Datentypen hervorgehoben wird. Die Eigenschaft *birthDate* wird durch das externe Vokabular Schema.org definiert und mittels *@id* sowie *schema:birthDate* der IRI `https://`

`schema.org/birthDate` zugeordnet. Diese Zuordnung ermöglicht eine kompakte IRI-Darstellung, wie *schema:birthDate*, anstelle der vollständigen IRI `https://schema.org/birthDate`. Der Datentyp des Literals wird durch die Verwendung von *@type* und dem XMLSchema, repräsentiert durch *xsd:dateTime*, bestimmt. Für weitere Informationen zum JSON-LD-Kontextdatei sei auf Kellogg [Gre+20a; Kel23, §3.1, §6] verwiesen. Die Definition des Credential-Types (Quelltext 5.3) erfolgt durch die JSON-LD-Kontextdatei (Quelltext 5.4). Die Verknüpfung der IRI im *@context* des VC weist den Attributen den Kontext aus der JSON-LD-Kontextdatei zu. Dadurch wird die semantische Genauigkeit von maschinenlesbaren Daten verbessert und die Informationsqualität des Credentials erhöht.

Credential-Schema Bei Credential-Schemata des VCDMs können zwei Varianten unterschieden werden: Datenvalidierungsschemata und Datenkodierungsschemata. Erstere dienen zur Überprüfung der Übereinstimmung von Struktur und Inhalt eines Credentials durch ein entsprechendes Schema, während letztere die Umwandlung des Inhalts eines VC in ein alternatives Repräsentationsformat, beispielsweise ein binäres Format für Zero-Knowledge-Proofs[32], leisten (vgl. [SLC23, §5.4]). Datenvalidierungsschemata können die strukturelle Integrität von Credentials validieren. Dabei können sie entweder einzelne Komponenten wie Claims oder das gesamte Credential abdecken. Wie in Abbildung 5.1 veranschaulicht, erstellt der Schema-Autor Schemata zur Nutzung durch Schema-Nutzer. Dabei kann der Autor selbst auch als Nutzer agieren und umgekehrt. Der Issuer erstellt ein Credential basierend auf der Struktur des VC-Typ-Schemas, signiert ihn und übergibt das VC an den Holder. Für die Transformation des VCs in eine VP kann ein Datenkodierungsschema eingesetzt werden. Für eine VP besteht die Möglichkeit, mehrere VCs zu einer einzigen VP zusammenzufassen, die als sogenannter Compounded-Proof bezeichnet wird. Im Allgemeinen signiert der Holder bei der Bildung einer VP die Präsentation und übermittelt diese anschließend an den Verifier. Der Verifier kann anhand des veröffentlichten Schemas den Inhalt der erhaltenen VP auf die strukturelle Integrität abgleichen. Anschließend steht er vor der Vertrauensentscheidung.

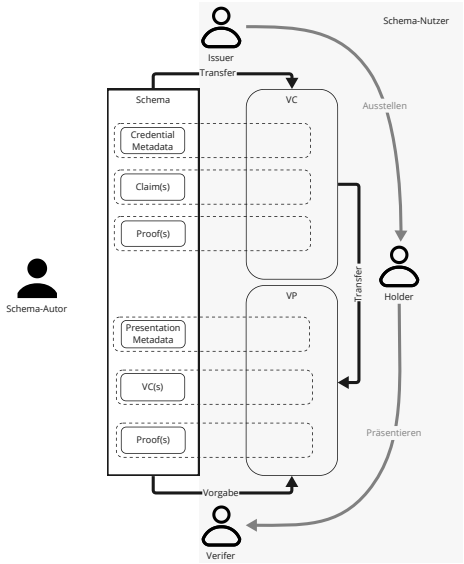

Abbildung 5.1: Nutzung des Schemas im Credential-Handlingprozess

[32] Null-Wissens-Nachweise

Zur Validierung der Struktur des Credentials können Technologien wie JSON-Schema, Shapes-Constraint-Language (SHACL) oder ShEx[33] verwendet werden. SHACL und ShEx wurden speziell für die Validierung von RDF-Datenstrukturen entwickelt, während JSON-Schema für die von JSON-Daten konzipiert ist. Bei Verwendung von JSON-LD können SHACL und ShEx zur Validierung der Terminologie, definiert in der JSON-LD-Kontextdatei, und der Datenstruktur des Credential-Types eingesetzt werden. Im Gegensatz kann JSON-Schema den über JSON-LD übertragenen Kontext und dessen Auflösung in RDF-Triples nicht validieren. Im Falle von der Spezifikation VC-JSON-Schema kommt JSON-Schema für die Überprüfung der Credential-Struktur zum Einsatz.

VC-JSON-Schema Die Spezifikation VC-Extension-Registry fungiert als Informationsquelle für alle bekannten Erweiterungen des VCDMs sowie deren zugehörigen Spezifikationen und befindet sich momentan noch in einem inoffiziellen Stadium. Als Validierungsmethode für das Schema wird die Methode *CredentialSchema2022* aufgeführt (vgl. [SLC23]). Festgelegt wird das CredentialSchema2022 in der VC-JSON-Schema-2022-Spezifikation. Sie wurde von der W3C-Credentials-Community-Group entwickelt und basiert auf einer Initiative des Unternehmens Workday, welches die Vorläuferspezifikation bereitgestellt hat (vlg. [Ham+19]). Die Definition eines Credential-Schemas unter Anwendung der VC-JSON-Schema-2022 Spezifikation orientiert sich an folgenden Grundsätzen:

> - A schema is versionable and it can evolve via new versions over time.
> - A schema is available for any issuer to use in a Credential and any holder or verifier of that Credential read.
> - A schema always guarantees the structure of a credential. A schema can apply to all or specific parts of a credential.
>
> [CS22, §4.2]

In der Spezifikation besteht ein Schema aus zwei Komponenten: Metadaten, die Kontextinformationen zum Schema bereitstellen und Daten, die zur Validierung eines Credentials verwendet werden. Um die Metadaten des Schemas zu definieren, müssen die folgend aufgelisteten Eigenschaften als *Daten-String* beschrieben werden:

1. type: URI[34] der Spezifikation (siehe Quelltext 5.5)
2. version: Semantische Version des Schemas.
3. id: Eindeutiger Identifikator des Schemas, empfohlen wird eine DID, die als URI fungiert.
4. name: Für Menschen einfach interpretierbarer Bezeichner des Schemas.
5. author: Schema-Autor, empfohlen wird eine DID zu verwenden.
6. authored: Erstellungsdatum in Form eines RFC330[35]-Timestamps.

Metadaten sollen die Auffindbarkeit und die Glaubwürdigkeit des Schemas verbessern und somit die IQ. Der Quelltext 5.5 zeigt, ergänzend zu obiger Liste, die von der Spezifikation festgelegten Metadaten.

```
{
        "type": "https://w3c-ccg.github.io/vc-json-schemas/",
        "version": "1.0",
        "id": "06e126d1-fa44-4882-a243-1e326fbe21db",
```

[33] Shape Expressions (ShEx)
[34] Uniform Resource Identifier
[35] https://www.rfc-editor.org/rfc/rfc3339

```
                "name": "Email",
                "author": "did:example:MDP8AsFhHzhwUvGNuYkX7T",
                "authored": "2021-01-01T00:00:00+00:00"}
}
```

Quelltext 5.5: CredentialSchema2022: Metadata [CS22]

Über das JSON-Object *schema* werden die Attributseigenschaften des Schemas für das VC angegeben. Die einheitliche Darstellung des erstellten Schemas, im Hinblick auf die Anforderungen gemäß der Spezifikation, kann mithilfe des JSON-Schemas, Credential-Schema-2.0[36], überprüft werden.

Die Spezifikation versioniert Schemata mittels des Modell-Revision-Ansatzes, einer Form der semantischen Versionierung. Versionsnummern werden in der Form *Modell.Revision* angegeben. Änderungen mit Auswirkungen auf die Abwärtskompatibilität werden durch Erhöhung der Modell-Ziffer, Änderungen ohne Auswirkungen durch Erhöhung der Revisions-Ziffer signalisiert (siehe Quelltext 5.6). Für die Vertrauenswürdigkeit des Schemas und dessen Autor wird als Authentifizierungsmethode Data-Integrity [DM22a] oder JOSE[37] [IET16] vorgeschlagen, aber nicht weiter ausgeführt. Um ein Schema als konsistente Vorlage für ein VC verwenden zu können, setzen die Autoren dessen Datenintegrität und Versionierung voraus. Eine Weiterentwicklung eines Schemas muss über die Versionseigenschaft in den Metadaten und über den Identifikator sichtbar gemacht werden, um die Interoperabilität bei der Nutzung zu verbessern. Ein anforderungsentsprechender Identifikator für ein Schema kann über eine DID realisiert werden (siehe Quelltext 5.6).

```
did:example:MDP8A...X7T/06...-1b;version=1.0
```

Quelltext 5.6: DID: Syntaktisches Beispiel eines Schemaidentifikators

Dies kann durch einen eindeutigen Identifikator erreicht werden, der das Schema über verschiedene Speichermedien hinweg referenziert. Die Referenzierung über unterschiedliche Datenspeicher soll zu einer verbesserten Verfügbarkeit der Schemata beitragen. Als Datenspeicher für Schemata kann eine beliebige VDR verwendet werden. Faktoren wie Speicherort, Urheberschaft, verwendeten Identifikatoren und Versionen beeinflussen entscheidend die Akzeptanz von Schemata durch die Schema-Nutzer. Der Prozess und die Kriterien, die zur Akzeptanz von Schemata führen, bleiben in der Spezifikation unbehandelt. Die Autoren der VC-JSON-Schema-2022-Spezifikation stellen einen Ansatz vor, der durch definierte Metadaten und den Einsatz von DIDs ermöglicht, ein Schema in verschiedenen Ökosystemen anzuwenden und zu versionieren.

Proof Ein Proof ist ein kryptografisches Verfahren zur Sicherstellung der Datenintegrität des VCs und zur Bestimmung der Urheberschaft des Proof-Erstellers. Im VCDM können diverse Signaturverfahren wie Ed25519Signature2018 oder RsaSignature2018, für die Erstellung eines Proofs eingesetzt werden. Diese Verfahren lassen sich grob in zwei Kategorien unterteilen: Linked-Data (LD)-Proofs und JSON-Web-Token.

Linked-Data-Proofs (LDPs) Bei LDPs werden Metadaten und Claims im Credential durch die Auflösung mit den verlinkten Kontextdateien, wie im Beispiel PermanentResidentCard wie URLs `https://www.w3.org/ns/credentials/v2` und `https://w3id.org/citizenship/v1`, expan-

[36] `https://github.com/w3c-ccg/vc-json-schemas/blob/main/docs/schema/v2/schema.json`
[37] Javascript Object Signing and Encryption (JOSE)

diert. In einem weiteren Schritt werden die JSON-LD-Daten in RDF normalisiert, speziell in N-Quad-Strings. Hierfür wird der Normalisierungsalgorithmus URDNA2015[38] verwendet (vgl. [DM22b]). Anschließend werden die normalisierten Strings gehasht und die daraus resultierenden Hashes signiert. Bei diesem Proof werden somit auch die Inhalte der Kontextdateien in die Signatur miteinbezogen. Informationen, die über eine URL im VC verlinkt sind und nicht in RDF umgewandelt werden können, unterliegen dem Risiko einer nicht nachprüfbaren Änderung und besitzen keine durch den Proof gesicherte Datenintegrität. Die im Zusammenhang mit dem VCDM verwendeten LDPs sind im Verzeichnis der Linked-Data-Cryptographic-Suite-Registry aufgeführt [MDO20]. Dazu gehören unter anderem BBS+[39][TO23] und JWS[40] für Data-Integrity-Proofs [OM22].

JSON Web Token (JWT) Das JWT[41]-Proof-Format, welches JSON als Datenstruktur nutzt, wird in der Spezifikation [SJ23] für die Verwendung im Zusammenhang mit dem VCDM beschrieben. JWT wird durch den RFC7519 [JBS15] Standard festgelegt. JWT wird in einer Vielzahl von Anwendungen, insbesondere im Bereich von OpenID-Connect und OAuth 2.0, eingesetzt. Die IANA[42] hat für JWTs ein Vokabular festgelegt, welches bestimmte Terminologien für den Einsatz mit dem VCDM definiert. Beispielsweise müssen Anpassungen wie die Umbenennung von *id* zu *jti* oder *issuer* zu *iss* vorgenommen werden (vgl. [SLC23; Bra+22, §2.2]). Um eine VCDM-kompatible JSON-Kodierung zu erzielen, muss die Eigenschaft *@context* angegeben, aber nicht verarbeitet werden. JSON-LD-Kontextdaten können für die Informationsrepräsentation mit JWTs verwendet werden, besitzen aber keine durch den Proof gesicherte Datenintegrität.

Verifiable Data Registry (VDR) Im Gegensatz zu AnonCreds, das spezifische VDRs wie Hyperledger-Indy nutzt, schreibt das VCDM keine bestimmte VDR vor. Im Vertrauensökosystem (siehe Kapitel 3) kann festgelegt werden, welche VDR für das Hinterlegen der Credential-Schemata und der JSON-LD-Kontextdateien eingesetzt werden soll. Sporny und Chadwick betonen die Wichtigkeit der Zugänglichkeit von JSON-LD-Kontextdateien für Credential-Types, um damit die Nutzbarkeit sicherzustellen. Sie schlagen verschiedene Strategien vor, wie die Nutzung von content-addressed-URLs, das Zusammenfassen von JSON-LD-Kontextdateien und die Integration dieser in Identity-Wallets (durch fest codierte @context-Werte) oder das aktive Caching von Kontextdateien mithilfe des Identity-Wallets (vgl. [SLC23; Cha+19, §5.3, §10]). Das Caching und die feste Kodierung der @context-Werte bieten für den Nutzer den Vorteil, dass für gespeicherte Daten keine Verbindung zu Drittanbietern hergestellt werden muss, um mit den entsprechenden Credential-Types zu interagieren. Dadurch wird die Datenverarbeitung unabhängiger und effizienter gestaltet. Durch Content-addressable-Identifikatoren wird ein Hash des zu verlinkenden Inhalts erzeugt, der anschließend in der URL integriert wird. Beispiele sind u.a. IPFS [Pro] und Hashlinks [SR21]. Bei Daten, die nicht durch einen JSON-LD-Kontext repräsentiert werden oder für die ein Proof über JWTs gebildet wird, kann die Datenintegrität der URL im VC beispielsweise durch die Verwendung eines Hashlinks [SR21] überprüft und damit die IQ-Metrik S3 B.1 für die angegebenen URL erfüllt werden. Beim Einbinden externer Quellen im VC müssen der Schema-Autor oder der Issuer entscheiden, ob und in welcher Form Datenintegrität geschaffen wird.

[38] Universal RDF Dataset Normalization Algorithm 2015 (URDNA2015)
[39] Boneh-Boyen-Short (BBS)
[40] JSON-Web-Signature (JWS)
[41] JSON Web Token (JWT)
[42] Die Internet-Assigned-Numbers-Authority (IANA) ist eine Organisation, die für die Koordination und Verwaltung globaler Internetressourcen zuständig ist, einschließlich der Zuweisung von IP-Adressen, Domain-Namen und Protokoll-Identifikatoren [Inta].

5.3 Vergleich der Credential-Formate

In Tabelle 5.1 werden die erläuterten Credential-Formate miteinander verglichen, Kriterien bilden dafür relevante IQs-Dimensionen.

Tabelle 5.1: Übersicht: Credential-Formate

			Credential-Format				
IQ-Dimension	**Abk.**	**Eigenschaften**	JSON-LD + JWS	JSON-LD + BBS	JSON-JWT	JSON-ZKP-CL (AnonCreds) mit Indy	**Kommentar**
Offene Formate	O1	Datenformat	JSON-LD		JSON		JSON ist ein offenes, selbstbeschreibendes Format (erfüllt O1 und I6).
Genauigkeit	Syn	Strukturelle Disambiguität / CERD_DEF erforderlich?	Ja		Nein	Ja	[Yil+22]
	Sem	Semantische Eindeutigkeit	Ja*		Nein / Ja*	Nein	* = Wenn definiertes Vokabular via @context eingebunden ist [Yil+22; You21].
Abrufbarkeit (Availability)	A	Verwendbar, wenn die Quelle der Anhänge offline sind?	Nein*		Ja	Nein	* = Nicht möglich ohne caching von JSON-LD-Kontextdateien [Yil+22].
Lizenz	L1, L2	Möglichkeit der Integration einer Nutzerbedingung im Credential-Format	Ja*			Nein	* = Via der Credential-Eigenschaft TermsOfUse
Interoperabilität	I3	Primäre Terminologiequelle für die Definition von Credential-Types	Nein, OWA		Ja, IANA	Nein/JA*, On-Ledger	* = VDR in der das Schema hinterlegt werden soll, das den Credenital-Type definiert.
Sicherheit	S3	Kann die Integrität des Datenschemas überprüft werden?	Nein / Ja*			Ja	* = Ist von der eingesetzten Technologie abhängig.
Minimalismus	M2	Selective Disclosure	Nein	Ja	Nein	Ja	[Yil+22; You21]
	M3	Predicate-Proofs	Nein	Ja*	Nein	Ja	Noch nicht implementiert [Yil+22].

Minimalismus Um die Informationsqualität zu steigern, ist es für den Autor eines Schemas unerlässlich, das Qualitätskriterium des Minimalismus zu berücksichtigen, wie in Abschnitt 4.3 bereits dargelegt. Besondere Sensibilität ist dabei im Umgang mit PIIs geboten, um rechtliche Aspekte und die Privatsphäre zu achten (vlg. [Kno19], Abschnitt 6.2.2). Zur Umsetzung der Klassifizierung von PII kann die BIT[43] genutzt werden (vgl. [KW20]). Mithilfe dieser standardisierten Klassifizierung können Agents innerhalb eines Schemas Funktionen bereitstellen, welche den Holder auf potenzielle Risiken hinweisen und ihn dazu auffordern, zu entscheiden, ob er tatsächlich bereit ist, seine PII mit dem Verifier zu teilen (vgl. [Sak+14, §17]). Beim Erstellen von Schema-Definitionen für JSON-LD-JWS und JSON-JWT ist zu beachten, dass keine Funktion zur selektiven Offenlegung vorhanden ist. Eine solche Funktion kann im Anwendungsfall allerdings notwendig sein, um den Grundsatz des Datenminimalismus während des Verifizierungsprozesses zu wahren (siehe IQ-Metriken in Tabellen 5.1 M2 und M3).

Sowohl AnonCreds als auch das VCDM basieren auf JSON. Die Untersuchung der Credential-Formate zeigt mitunter, dass das VCDM auch andere, in RDF serialisierbare Formate, unterstützt. Aufgrund seiner Kompatibilität liegt der Schwerpunkt der Arbeit auf JSON-LD.

Möglichkeiten der Informationsrepräsentation Das VCDM ermöglicht die Erstellung von Kontexten auf der semantischen und teilweise auf der pragmatischen Ebene. Durch das *CredentialSchema* kann ein Schema im VC festgelegt werden, das die syntaktischen Ebene des entsprechenden

[43] Blinding Identity Taxonomy (BIT)

Credential-Types bestimmt. Eine dem Credential-Type entsprechende JSON-LD-Kontextdatei ermöglicht die Definition dem VC-Types entsprechender Terminologien, während der pragmatische Kontext im VC durch die Verwendung von *TermsOfUse* hinterlegt werden kann. Die IQ-Metrik I5, welche mehrere Serialisierungsformate berücksichtigt, kann beim Credential-Format VCDM auf die JSON-LD-Repräsentation reduziert werden, da JSON-LD vorrangig als Serialisierungsformat für Credentials verwendet wird.

Die Bildung eines Schemas über das Credential-Format AnonCreds beschränkt sich derzeit auf die syntaktische Ebene, wobei nur die Attributsbezeichner im String *attr_names* angegeben werden können. Um Definitionen auf der semantischen Ebene zu ermöglichen, wurde das Konzept des Rich-Schemas vorgestellt, das die Verwendung von JSON-LD-Kontext mit AnonCreds erlaubt [Zun21]. Die Veröffentlichung im Jahr 2019 mündete jedoch nicht in der Implementierung von Rich-Schema [KB19]. Im Vergleich zu AnonCreds bietet das VCDM eine umfangreichere Palette an Ansätzen zur Integration von Kontexten in VCs und schafft dadurch mehr Gestaltungsmöglichkeiten für die zugrundeliegenden Schemata.

Der SSI-Technologie-Entwickler Evernym[ST18] beschreibt die Entstehung von Schemata als einen natürlichen Prozess, der sich aus den spezifischen Anwendungsfällen in bestimmten Bereichen ergibt (vgl. auch [Bel21]). Im Kontext der SSI gibt es beispielhafte Konzepte für die Entwicklung von Schemata aus Anwendungsfällen wie Open-Badges (siehe Abschnitt 6.1) und Bonifi, ehemals CU-Ledger (vgl. [SSN22, S. 14]).

Die Möglichkeit der Transformation zwischen den Credential-Formaten schafft eine höhere Interoperabilität und Nutzbarkeit der Credentials. Eine Umsetzung in Realanwendungen fand bis jetzt noch nicht statt. Prototypisch wurde die Transformation von AnonCreds und VCDM-Credentials anhand einer JSON-LD-Kontextdatei[44] und eines Python-Scripts demonstriert. Dabei wurden AnonCreds VCs in das konforme VCDM-Format umgewandelt und wieder zurückkonvertiert, ebenso wie AnonCreds-Präsentationen in ein konformes VCDM-Format VPs (vlg. [WS22]).

Angesichts der Beschränkungen von AnonCreds bei der Abbildung der semantischen Ebene, der Bestrebungen von Rich-Schemas sowie der potentiellen Transformation von AnonCreds zu VCDM-Credentials, wird in dem folgenden Kapitel das VCDM mit Fokus auf Datenvalidierungsschemata untersucht. Die Datenkodierungsschemata werden in dieser Arbeit nicht weiter behandelt.

[44] `https://andrewwhitehead.github.io/anoncreds-w3c-mapping/schema.json`

6 Credential-Schemaprozess

Basierend auf dem vorherigen Kapitel konzentriert sich die Untersuchung möglicher Credential-Schemaprozess-Ansätze in diesem Kapitel auf das Credential-Format VCDM. Es werden bestehende Credential-Types Definitionen und Schemaprozesse beschrieben und anhand der in den Anforderungen erhobenen Mechanismen analysiert.

6.1 Definierte Credential-Types

In diesem Abschnitt erfolgt eine Beschreibung bereits existierender Credential-Types und ihrer zugehörigen Vokabularien und Schemata. Diese dienen in Abschnitt 6.3 als Grundlage der Analyse der zu untersuchenden Mechanismen.

Open Badges Open-Badges stellen ein digitales Nachweissystem für den Bildungssektor dar, welches weltweit eingesetzt wird und von der Mozilla- und MacArthur Foundation 2010 ins Leben gerufen wurde [1Ed23]. Im Jahr 2022 wurden laut des 2022-Badge-Count-Reports [1Ed22] 74,7 Millionen Open-Badges ausgestellt und es gibt etwa 521.000 verschiedene Abzeichen, die für die Attestierung von Bildungsleistungen einer Person verwendet werden können. Im Zuge des Übergangs von Version 2 zu Version 3 der Open-Badges-Spezifikation, vom Konsortium 1EdTech erstellt, werden die Badges im Credential-Format VCDM, Version 1.1 umgesetzt, d.h. sie werden über den Credential-Type OpenBadgeCredential definiert (vgl. [Nat+23b, §2.2]). Eine der Spezifikationen von 1EdTech ist der 1EdTech-JSON-Schema-Validator-2019. Er beschreibt wie Credential-Schemata für Open-Badges in einem VC angegeben und validiert werden [And22]. Der JSON-LD-Kontext wird in der Open-Badges-Spezifikation, Version 3, definiert und ist über die URL `https://purl.imsglobal.org/spec/ob/v3p0/context.json` im JSON-Format abrufbar. Durch die Integration einer weiteren JSON-LD-Kontext-Datei namens Extensions[45] kann das Open-Badge-Credential um Linked-Data-Beschreibungen erweitert werden. Mögliche Erweiterungen sind der genannte 1EdTech-JSON-Schema-Validator-2019, der 1EdTechCredentialRefresh-Service und die 1EdTechRevocationList. Ein weiterer Credential-Type, der von 1EdTech spezifiziert wurde, ist der Comprehensive-Learner-Record (CLR). Dies liefert einen umfassenden Lernerfolgsbericht, der über die Kapazität von Open-Badges hinausgeht. Der CLR [Nat+23a] liegt derzeit in Version 2 vor. Open-Badges sind visuelle Anerkennungen, die im Internet geteilt werden können, während der CLR dazu dient, detailliertere Informationen zu Lernergebnissen und Lernweg zu erfassen. Der CLR-Standard erlaubt es, Lernleistungen und Beurteilungsergebnisse innerhalb spezifischer Programme oder Lernpfade zu kontextualisieren und ergänzt damit die Kapazität von Open-Badges (vgl. [1Ed]).

Traceability Vocabulary Das Traceability-Vocabulary ist ein Vokabular für VCs innerhalb der Logistikdomäne, es fasst Informationen zu Lieferketten und Rückverfolgbarkeit. Diese Informationen geben Rückschluss auf unter anderem Herkunftsland, chemische und mechanische Eigenschaften sowie andere Attribute von Produkten und Materialien. Der Einsatz von VCs dient der Bestimmung der Legitimität von Organisationen im globalen Handel sowie der des Status der beschriebenen Pro-

[45] `https://purl.imsglobal.org/spec/ob/v3p0/extensions.json`

dukte und Materialien (vgl. [Ori+23]). Sämtliche VCs des Traceability-Vokabulars beziehen sich auf eine JSON-LD-Kontextdatei[46]. Das Traceability-Vokabular umfasst in der Version 0.1 gegenwärtig 62 Credential-Types[47].

Die Software zur Repräsentation von Credential-Types im Bereich der Rückverfolgbarkeit wird von der Arbeitsgruppe W3C-Credentials-Community-Group entwickelt und ist der W3C-Software- und Dokumentlizenz[48] sowie den damit verbundenen Regelungen unterworfen. Die Verwendung dieser Architektur ermöglicht die Definition von Credential-Types innerhalb eines spezifischen Anwendungsbereichs durch die Kombination von Credential-Schemata mit einer JSON-LD-Kontextdatei und zugehörigen webbasierten Dokumentation.

Neben dem Traceability-Vocabulary hat die W3C-Credentials-Community-Group andere Vokabularien experimentellen Charakters erstellt, darunter das Citizenship-Vocabulary [LS20], welches VCs in der Domäne von Residenz- und Staatsangehörigkeitsinformationen beschreibt. Darin sind Attribute wie Vor- und Nachname, Geburtsdatum und Staatsangehörigkeit enthalten, die zur Identitätsbestimmung eines Bürgers genutzt werden können. Ein weiteres Vokabular ist das Vaccination-Certificate-Vocabulary [LSP21], welches ein Vokabular für VCs im Zusammenhang mit Impfzertifikaten für medizinische Zwecke bereitstellt.

In Abbildung 6.1 wird der Schemaprozess gemäß dem in den Anforderungen definierten Use-Case-Diagramm dargestellt.

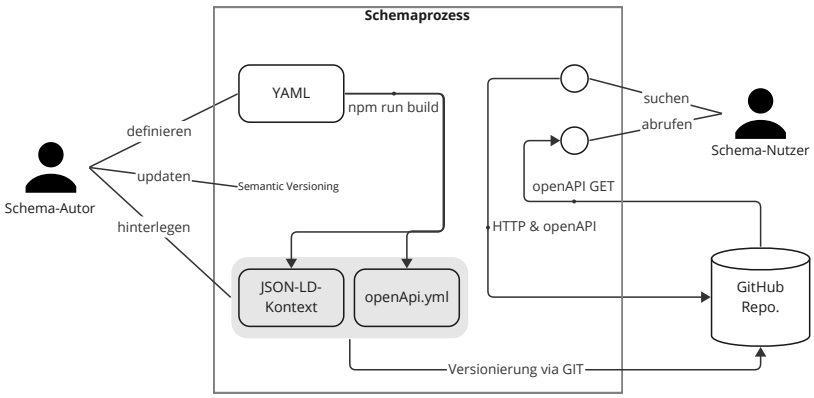

Abbildung 6.1: Use-Case-Diagramm: Adaption des Schemaprozesses für das Traceability-Vocabulary

Definition: Durch das Erstellen einer YAML-Datei, welche die Claims (Attribute) definiert, und einer weiteren YAML-Datei, die den Credential-Type bestimmt und mit der zuvor erstellten YAML-Datei verknüpft wird, können die Credential-Types und ihre Attribute definiert werden.

[46] https://w3c-ccg.github.io/traceability-vocab/contexts/traceability-v1.jsonld
[47] https://github.com/w3c-ccg/traceability-vocab/tree/main/docs/openapi/components/schemas/credentials
[48] https://github.com/w3c-ccg/traceability-vocab/blob/main/LICENSE.md

Hinterlegung: Das Traceability-Vocabulary nutzt die GitHub-Repositiory als VDR, Schemata sind dort gespeichert. Durch die Ausführung des Befehls *npm build* im entsprechenden JavaScript-Repository werden aus vordefinierten YAML-Dateien verschiedene Ausgabeformate erzeugt: eine JSON-LD-Kontextdatei, OpenAPI[49]-Daten im YAML-Datenformat und eine HTML-Datei.

Abruf: Die Credential-Schemata können mittels der OpenAPI aus der betreffenden GitHub-Repository bezogen werden. Für die Validierung des Credentials, die Prüfung der Übereinstimmung mit dem zugehörigen Schema, kann der CredentialSchema-Typ OpenApiSpecificationValidator2022 verwendet werden (siehe [Ste22]). Zur Durchführung des Abgleichs zwischen der YAML-Datei und dem Credential kann das Werkzeug ajv-cli[50] und der Befehl *ajv validate -s ./schema-credential.yaml -d ./credential.json –strict=false* eingesetzt werden.

Versionierung: Über GitHub erfolgt die Versionierung der Codebase auf der die Traceability-Vocabulary-Spezifiaktion aufbaut. Die Spezifikation selbst wird über semantische Versionierung verwaltet (vgl. [Pre22]). Eine direkte Integration von Datenintegritätsprüfungen ist nicht implementiert.

6.2 Plattformbasierte Ansätze des Schemaprozesses

In diesem Abschnitt werden plattformbasierte Ansätze des Schemaprozesses beschrieben und mit den Anforderungen des Credential-Schemaprozesses in Bezug gesetzt.

6.2.1 Editorgestützter Schemaprozess

Die Unternehmen Affinidi[51] und Serto[52] bieten Plattformen für den Schemaprozess an. Diese verfügen jeweils über einen Schema-Editor. Die Funktion der beiden Plattformen weisen eine starke Simularität auf und werden daher gemeinsam beschrieben. Über eine browserbasierte grafische Benutzeroberfläche können sowohl bestehende Schemata gesucht als auch neue Schemata erstellt werden. Um ein Schema zu erstellen, muss ein Benutzerkonto bei der entsprechenden Plattform vorhanden sein. Der Schema-Editor leitet den Schema-Autor grafisch durch den Erstellungsprozess. Die konzipierten Schemata sind ausschließlich für die Verwendung des Credential-Formats VCDM ausgerichtet. Bei der Definition des Schemas werden die Metadaten in Form von Name, Version und Beschreibung angelegt. Das Schema kann wahlweise im öffentlichen Register der Plattformen gelistet werden. Bei Serto lässt sich optional ein URL-Slug sowie ein Icon hinzufügen. Bei der Erstellung von Attributen im Schema-Editor werden der Attributsbezeichner und der entsprechende Datentyp deklariert und optional eine Beschreibung hinzugefügt. Die unterstützten Datentypen umfassen DID, Text, Boolean, Number, Datetime, Date, URI und Nest-Attributes. Diese Datentypen werden mithilfe von JSON-LD *@type* definiert und in Bezug zum Vokabular von Schema.org gesetzt. Attribute können als Pflichtfelder des Schemas gekennzeichnet werden. Diese Angaben werden durch die Required-Function in die JSON-Schema-Datei übertragen, um das Ausfüllen des Attributs bei der Validierung des VCs sicherzustellen. Der Schema-Editor bei Serto lässt eine manuelle Bearbeitung der

[49] Die OpenAPI-Spezifikation ist ein standardisiertes Verfahren zur Dokumentation von HTTP-basierten APIs, früher als Swagger-API bekannt (vgl. [Dar+21]).
[50] `https://github.com/ajv-validator/ajv-cli`
[51] `https://ui.schema.affinidi.com/schemas`
[52] Serto ist ein Teil des ConsenSys-Unternehmens. `https://schemas.serto.id/`

JSON-Dateien zu. Unter Anwendung des JSON-LD-Context-Plus-Schema-Playgrounds[53] können sowohl der Kontext als auch die JSON-Schema-Datei übersichtlich bearbeitet werden. Die Schema-Editoren generieren anhand der Angaben des Schema-Autors eine JSON-LD-Kontextdatei sowie eine JSON-Schema-Datei. Nach ihrer Generierung sind Schemata nicht mehr editierbar. Stattdessen können sie durch semantische Versionierung seitens des Schema-Autors oder durch Forking von anderen Autoren weiterentwickelt werden. Die Schemata sind öffentlich über ihre URL zugänglich und verfügen über keine Zugangsbeschränkungen für die Schema-Nutzer. Existiert der Name des Schemas bei Affinidi bereits, wird die Revisionsnummer inkrementell erhöht; ist der Name des Credential-Types verschieden, wird ein neues Schema auf der Basis des bestehenden Schemas erstellt (vgl. [Aff22]). Eine Version kann nicht mehr als 999 Revisionen erfahren, alle Versionen und Revisionen müssen sequentiell sein. Die JavaScript-Bibliothek VC-Schema-Tools stellt eine Klasse namens VCSchema bereit, die mit einem JSON-LD-Context-Plus-Schema instanziiert werden kann. Die Klasse bietet Funktionen zur Validierung des VCs, zur Generierung eines JSON-LD-Kontexts und eines JSON-Schemas sowie zum Zugriff auf verschiedene andere Dienstprogramme. Die Hauptfunktion der Bibliothek besteht darin, die *credentialSchema*-Eigenschaft zu überprüfen, die das JSON-Schema für die VCs enthält, um diese entsprechend zu validieren. Wenn keine *credentialSchema*-Eigenschaft gefunden wird, wird das VC anhand eines Fallback-Schemas überprüft, das grundlegende Eigenschaften wie Typ, Aussteller, Ausstellungsdatum usw. abgleicht. Die auf den Plattformen erstellten Schemata können gemäß des VCDM in das Credential integriert werden, wie in Quelltext 6.1 dargestellt.

```
"@context": [
  "https://www.w3.org/2018/credentials/v1",
  "https://beta.api.schemas.serto.id/v1/public/employee/1.2/ld-context.json"
        ],
        ...
"credentialSchema": {
  "id": "https://beta.api.schemas.serto.id/v1/public/employee/1.2/json-schema.json",
  "type": "JsonSchemaValidator2018"
```

Quelltext 6.1: Definition eines Credentials durch ein Serto-Schema

Die Affinidi-Plattform stellt API-Endpunkte bereit, die mithilfe einer OpenAPI[54] spezifiziert sind. Über diese API-Endpunkte können Benutzer Schemata über einen Post-Befehl erstellen und über einen Get-Befehl Schemata abrufen.

6.2.2 Datenmodelle auf Basis von Overlay-Strukturen

Overlay-basierte Datenmodelle bilden einen weiteren Ansatz für die Abbildung des Schemaprozesses. Besonderer Fokus liegt bei diesen Modellen auf der semantischen Ebene. Die dargelegten Modelle Overlays-Capture-Architecture (OCA) und Semantic-Overlay-Architecture (SOyA) sind generischer Art und nicht primär für die Erstellung von Credential-Schemata nach VCDM konzipiert. Dennoch werden sie in die Betrachtung einbezogen, um einen potentiellen Vorzug der Overlay-Struktur herauszuarbeiten und zu prüfen.

[53] https://schemas.serto.id/playground
[54] https://affinidi-schema-manager.prod.affinity-project.org/api-docs/

Laut Allen ist die Interoperabilität zwischen verschiedenen Anwendungen und Organisationen für die SSI von entscheidender Bedeutung. Um diese zu gewährleisten, müssen SSI-Anwendungen in der Lage sein, eine Vielzahl von Credentials und zugehörigen Claims zu verarbeiten und in Beziehung zu setzen, dies kann sich je nach individuellem Kontext des Anwendungsfalls unterscheiden. Die Flexibilität spielt eine entscheidende Rolle, um Credentials in verschiedenen digitalen Ökosystemen und über Staatsgrenzen hinweg für Vertrauensentscheidungen nutzen zu können. Konventionelle digitale Identitätssysteme, wie GOV.UK Verify, sind oft auf bestimmte Zwecke oder feste Attributsätze beschränkt, was ihre Flexibilität einschränkt (vgl. [Win23, §3]). In diesem Zusammenhang könnten polymorphe Datenschemata, die sich an unterschiedliche Anforderungen und Kontexte anpassen und damit die Flexibilität und Autonomie der Benutzer in verschiedenen Anwendungsfällen gewährleisten, von entscheidender Bedeutung sein. Gebunden zum einen an die Frage, wie und ob sich polymorphe Datenschemata für Credential-Schemata einsetzen lassen. Zum anderen, ob die Nutzung solcher Schemata zu einer erhöhten Fallzahl an Verwendungseinsatz der Credentials und deren Präsentationen führt sowie generell zu einer besseren Ausschöpfung des Potenzials von SSI-Ökosystemen beiträgt.

Overlays Capture Architecture (OCA) Die Human Colossus Foundation hat ein Datenmodell für die Harmonisierung von Daten entwickelt, das als OCA bezeichnet wird. Das Datenmodell wird im Good-Health-Pass-Interoperability-Blueprint für die Verwendung mit Credentials empfohlen (vgl. [Tru21, S. 159 – 161]). Die Datenstruktur von OCA setzt sich aus OCA-Bundles zusammen. Ein Bundle besteht aus der Basisinformation, der Capture-Base, und den zugehörigen, in beliebiger Anzahl vorliegenden Overlays. Durch das Hinzufügen neu definierter Overlays können verschiedene Parteien einer Capture-Base zusätzlichen Kontext verleihen. Ein Bundle bildet ein Schema. Die Capture-Base stellt folglich die Minimalform eines Schemas dar, das über folgende Eigenschaften charakterisiert wird:

- Type: Bezeichnet den Typ des Objektes.
- Classification: Dient zur Zuweisung von Klassifikationstandards wie GICS[55].
- Attributes: Die Attribute bestehen aus Paaren von Schlüssel und Wert, wobei der Schlüssel den Attributsnamen und der Wert den Attributtyp repräsentiert. Der Attributsname ermöglicht eine eindeutige Identifikation des Attributs innerhalb des OCA, während der Attributtyp dessen Syntax und Sortierung definiert.
- Flagged attributes: Über diesen Marker kann ein Attribut als PII klassifiziert werden (vgl. Abschnitt 5.3).

Die Datenstruktur der Capture-Base erlaubt die Verwendung der Datentypen Text, Numeric, Reference, Boolean, Binary, DateTime und Array. Der spezielle Datentyp Reference ist ein Self-Addressing-Identifier (SAID), der durch die Implementierung des SAID-Protokolls erzeugt wird, wie im Punkt 6.2.2 beschrieben.

Overlays werden durch die folgenden Eigenschaften definiert:

- Capture-Base: Wird über Referenzierung durch den Datentyp Reference der entsprechenden Capture-Base zugewiesen.
- Type: Gibt an welcher Typ von Overlay beschrieben wird.
- Dem Overlay entsprechende Attribute.

[55] Global Industry Classification Standard (GICS)

Overlays erfüllen spezifische semantische Aufgaben, indem sie Kontextinformationen bereitstellen, die beispielsweise für eine bestimmte Nationalsprache in einem Information-Overlay definiert sind. Zusammen liefern Overlays Kontext für das Datenobjekt in der Capture-Base. Overlays sind deterministisch mit ihrer Capture-Base verbunden und voneinander entkoppelt. Es existieren vier Kategorien von Overlays, welche im Folgenden aufgeführt sind:

- Semantik
- Eingaben
- Transformation
- Präsentation

Die Presentation-Overlay-Kategorie dient der Bereitstellung von Kontextinformationen für die Präsentation der Capture-Base. Innerhalb dieser Kategorien haben Overlays jeweils spezifische Aufgaben, so gibt zum Beispiel das Form-Overlay die Gestaltung des Eingabeformulars oder das Sensitive-Overlay die der Kennzeichnung von PIIs (siehe Abbildung 6.2) vor.

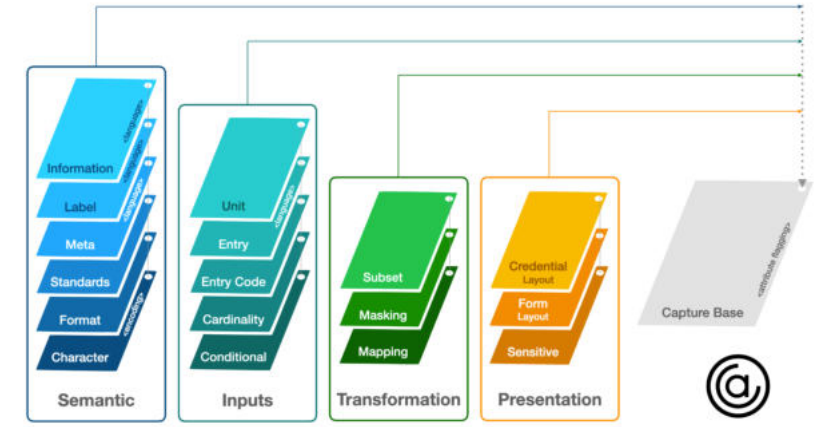

Abbildung 6.2: Übersicht der Overlay-Typen der OCA [Hum23a]

Die Human Colossus Foundation entwickelt Software-Komponenten unter der EURL-1.2-Lizenz[56] für die verschiedenen Prozessschritte, die den Lebenszyklus der OCA abbilden, nämlich Capture, Validation, Transformation und Presentation.

Über OCA kann ein Schema für ein VC definiert werden, in dem die Eigenschaften des VCs in den Overlays durch ihren Attributsbezeichner und deren Datentyp aufgelistet werden. Sollte einer Partei ein sekundärer Anwendungsfall vorliegen, der nicht durch den Schema-Autor abgedeckt wurde, so kann dessen Fehlen durch das Hinzufügen eigener Overlays zur vorhandenen Capture-Base behoben werden.

In Abbildung 6.3 wird der Schemaprozess gemäß dem in den Anforderungen definierten Use-Case-Diagramm dargestellt.

Definition: Über OCA ist die Darstellung einer JSON-LD-Kontextdatei nicht möglich. Daher wird der Fokus für die Definition auf das Erstellen eines Credential-Schemas gelegt. Um ein Schema zu definieren, kann eine XLS-Datei unter Verwendung des Tabellenwerkzeugs Excel bearbeitet werden, in der die Capture-Base und die zugehörigen Overlays definiert werden. Anschließend kann die

[56] `https://github.com/THCLab/licensing`

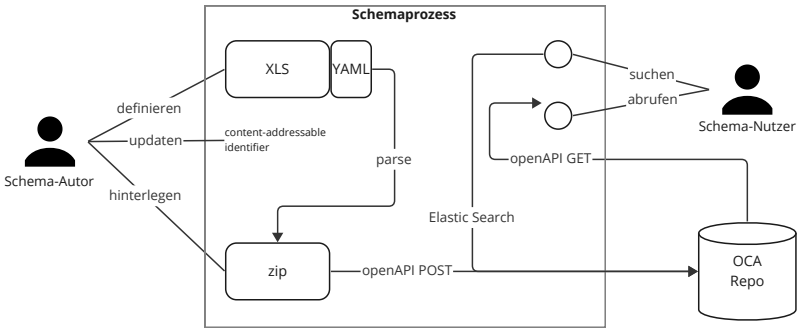

Abbildung 6.3: Use-Case-Diagramm: Adaption des Schemaprozesses für die OCA

XLS-Datei mit Hilfe des OCA-Parsers in ein OCA-Bundle konvertiert und innerhalb einer ZIP-Datei gespeichert werden (vgl. [Hum23c]). Um die Credential- und Form-Layouts zu erstellen, müssen die entsprechenden Daten in YAML definiert werden. Diese YAML-Dateien können dann einem OCA-Bundle hinzugefügt werden. Für diesen Vorgang kann der OCA-Browser verwendet werden, welcher das Erstellen von OCA-Bundles über ein Web-Interface unterstützt (vgl. [Hum23b]). Der OCA-Browser ermöglicht die Vorschau des durch das OCA-Bundle generierten Formulars und die Überprüfung der Datenintegrität mit den SAIDs.

Hinterlegung: Ein OCA-Bundle kann nicht nur mithilfe des OCA-Browsers in das OCA-Repository hochgeladen werden, sondern auch über einen OpenAPI-definierten Endpunkt (vgl. [Hum23d]).

Abruf: Bundles können über den API-Endpunkt aus dem OCA-Repository abgerufen werden. Dafür sollte neben der Repository eine Elastic-Search-Funktion bereitgestellt werden, die eine gezielte Suche nach entsprechenden Schemata zulässt. Der OCA-Validator ermöglicht die Validierung von Daten anhand eines Bundles. Der OCA-Validator sichert dabei die Datenintegrität des Bundles in Bezug auf die angegebenen SAIDs (vgl. [Hum23f]). Die Validierung eines Datensatzes erfolgt entsprechend den Umfangs- und Eingaberegeln des Bundles. Zum gegenwärtigen Zeitpunkt ist die Transformation von Datensätzen durch OCA noch nicht implementiert (vgl. [Hum23e]).

Versionierung: Die Bundles werden mithilfe des SAID-Identifikators versioniert, der content-adressierbar ist und sich je nach Inhalt ändert. SAIDs heben sich von anderen Identifikatoren durch das Einfügen des SAIDs in die zu adressierende Datei über ein Derivationsprotokoll ab. SAID verwendet den Blake-Hash-Algorithmus zur Bildung des Identifikators (vgl. [Smi22]).

Semantic Overlay Architecture (SOyA) Aufbauend auf dem Konzept von OCA hat die Initiative OwnYourData das Datenmodell SOyA entwickelt, das im Gegensatz aber auf semantischen Webtechnologien basiert. SOyA ist unter der MIT-Lizenz[57] verfügbar und dient in Verbindung mit den Semantic-Containern als Plattform zur Erstellung und Veröffentlichung von Datenmodellen. SOyA ist ein auf dem RDF basierendes Datenmodell, das darauf abzielt, die Einstiegshürde für Entwickler zu senken und die Tool-Unterstützung für RDF zu verbessern. Kernfunktionen von SOyA umfassen den Datenaustausch und die Dateninteroperabilität. Ziel des Datenmodells ist es, die Herausforderungen des Datenaustauschs in Bezug auf unterschiedliche Kodierungen, Syntaxen und Semantiken zu lösen und offene Standards effektiv zu nutzen, um damit die Dateninteroperabilität zu verbessern (vgl.

[57] `https://github.com/OwnYourData/soya/blob/main/LICENSE`

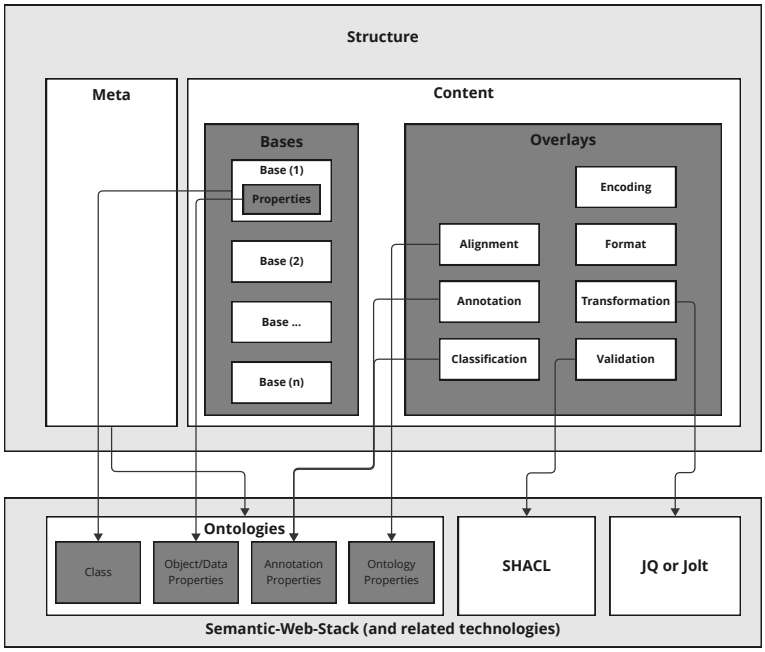

Abbildung 6.4: SOyA Komponenten [CFG23]

[Own22b]). Die Plattform bietet Funktionen zur Erstellung, Veröffentlichung, Validierung und Transformation von Daten an (siehe Abbildung 6.4). SOyA besteht aus den folgenden Hauptkomponenten zur Beschreibung und Verwaltung von Datenmodellen (vgl. [CFG23, §2]):

- Strukturen: Beschreibung eines Datenmodells mithilfe von Bases und Overlays sowie zusätzlichen Meta-Attributen.
- Semantic-Web-Stack: Integration etablierter Technologien zur Handhabung von Dateninstanzen und Datenmodellen. Dazu gehören SHACL zur Validierung von Daten und Werkzeuge wie jq und jolt zur Transformation und Speicherung von Daten in einem Semantic-Container.
- Ontologien: OWL2[58]-konforme Repräsentation von Datenmodellen, die automatisch aus der Structure generiert werden können.

In Abbildung 6.5 wird der Schemaprozess gemäß dem in den Anforderungen definierten Use-Case-Diagramm dargestellt.

Definition: Mithilfe von SOyA kann ein Credential-Type definiert werden, indem eine Basisstruktur mit den entsprechenden Properties (Attributen) erstellt wird. Um externe Vokabulare der Basis zuzuordnen, kann ein Alignment-Overlay definiert werden. Die Erstellung von Basisstruktur und Overlays erfolgt im YAML-Datenformat. Durch den CLI-Befehl *soya init* werden diese in JSON-LD-Dateien konvertiert, die der Terminologie der JSON-LD-Kontextdatei `https://ns.ownyourdata.eu/soya/soya-context.json` entsprechen. Die Konvertierung gewährleistet eine einheitliche Darstellung der JSON-LD-Dateien. Ein Alignment-Overlay oder die Basisstruktur kann als JSON-LD-Kontextdatei für einen Credential-Type verwendet werden. Ein Credential-Schema wird mithilfe eines ValidationOverlays, das eine Validierung nach SHACL spezifiziert, erstellt. Über das TransformationOverlay kann die Datenstruktur des Credentials definiert werden, in die die Daten aus der SOyA-Structure transformiert werden sollen. Die Transformation kann auch in andere Credential-Formate erfolgen.

[58]Web Ontology Language 2 (OWL2)

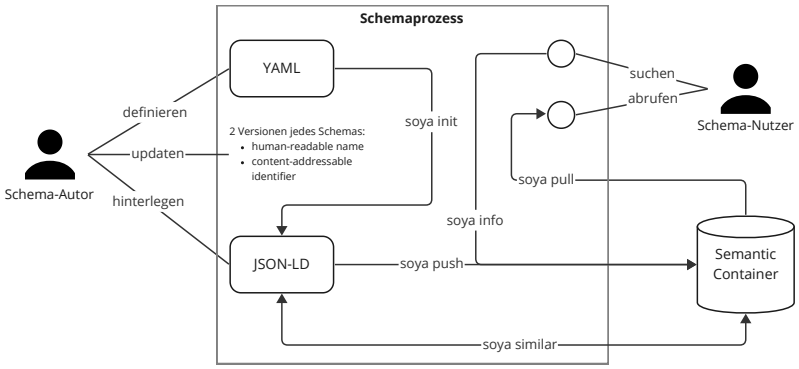

Abbildung 6.5: Use-Case-Diagramm: Adaption des Schemaprozesses für die SOyA

Die acSOYA-Structure, die die Claims eines Credentials definiert, wird über die Angaben *credentialSubject* für die Credential-Struktur und *credentialSchema* für die Spezifikation realisiert. Hierbei können Tools wie jq oder jolt verwendet werden.

Hinterlegung: Für die Datenhinterlegung verwendet SOyA Container, eigens entwickelte Docker-Images *oydeu/soya-base*. Dieses Docker-Image basiert auf dem Image des semantischen Containers *semcon/sc-base* (vgl. [Own22a]). Semantic-Containers weisen eine standardisierte Infrastruktur für die Datenbereitstellung auf. Datenanbieter können so Daten effizient bereitstellen, ohne die Kontrolle über deren Verwendung und Monetarisierung zu verlieren. Datenverbraucher wiederum können auf effizient und gut verwaltete Daten zugreifen und diese integrieren. (vgl. [Ver19]). Die erstellte acSOYA-Structure kann mit dem CLI-Befehl *soya similar* mit den Inhalten eines Semantic-Containers verglichen werden. Die Hinterlegung eines Schemas kann mithilfe des CLI-Befehls *soya push* erfolgen.

Abruf: In einem Semantic-Container hinterlegte Schemata können über den CLI-Befehl *soya pull* abgerufen werden. Mit dem CLI-Befehl *soya info* können Bases und deren Overlays im Semantic-Container abgefragt werden. Die Suche nach Schemata kann über eine API-Abfrage erfolgen. Der Befehl *soya form* leitet auf Basis von JSON-Forms[59] die automatische Generierung eines Formulars für die Dateneingabe ein, welches unter anderem in verschiedenen Nationalsprachen definiert werden kann. Mithilfe des Befehls *soya acquire* können JSON-Dateien in JSON-LD-Dateien transformiert werden, die dieselben Attribute wie die zugrunde liegende Base aufweisen. Bei SOyA werden Attribute auch als Properties bezeichnet. Der Befehl *soya validate* initiiert den Abgleich einer JSON-LD-Datei mit einem Validation-Overlay. Die Eingabe *soya transform* wandelt eine JSON-LD-Datei entsprechend eines Transformation-Overlays um. Semantic-Container können zur Bereitstellung eines SPARQL-Endpunktes erweitert werden. Der SPARQL-Endpunkt[60] wird durch ein zusätzliches Docker-Image bereitgestellt, das auf dem RML[61] basiert und für die Daten eines Semantic-Containers genutzt wird (vgl. [Own20]).

[59] `https://jsonforms.io/`

[60] Ein SPARQL-Endpunkt ist eine spezielle Art von HTTP-Server, der HTTP-Anfragen und -Antworten für SPARQL-Protokoll-Operationen verarbeitet [Lee+13, §1.2].

[61] RDF Mapping Language (RML)

Versionierung: Zur Sicherstellung der Datenintegrität und Adressierung von Datenobjekten nutzt SOyA den Decentralized-Resource-Identifier (DRI) als Hashlink. DRI verwendet das Multihash-Konfigurationsformat SHA256 und die Multibase-Kodierung BASE58-BTC, wie in der Hashlink-Spezifikation [SR21, §4] beschrieben. Ein Schema und seine Versionshistorie sowie zugehörige Overlays können beispielsweise über die URL `https://soya.data-container.net/Person/info` abgerufen werden. Bei Veröffentlichung einer neuen Version eines Schemas erhält diese den Index 0 und ist unter der URL `https://soya.data-container.net/Person/` verfügbar. Ältere Versionen werden in aufsteigender Reihenfolge neu indexiert. Ein Schema kann über seinen Hashlink, der einen eindeutigen Identifikator darstellt, abgerufen werden wie zQmTprHHbNsZiR4WGimgJSw3ATBH86qrbf92ipgyZAGHpXM.

6.3 Mechanismen des Credential-Schemaprozesses

Die in den Anforderungen, Kapitel 4, aufgestellten Mechanismen bilden und strukturieren den Schemaprozess. Im Folgenden werden unter dem Gesichtspunkt der Mechanismen die vorgestellten Ansätze miteinander verglichen und anschließend einer Bewertung unterzogen. In Tabelle 6.1 sind die untersuchten Ansätze aufgelistet und ihren jeweiligen JSON-LD-Kontextdateien, Credential-Schemata sowie der zugehörigen Schema-Dokumentation zugeordnet.

Tabelle 6.1: Überblick der untersuchten Ansätze

	Credential-Type					**Editor**	**Overlay**		
	Traceability Voc. v0.1	OpenBadges v3.0	Comprehensive Learner Record v2	Citizenship Voc. v0.3	Vaccination Certificate Voc. v0.1	Serto & Affinidi	OCA	SOyA	**Kommentar**
JSON-LD-Kontextdatei	traceability -v1.jsonld	ob/v3p0/ context-3.0.1.json	clr/v2p0/ context.json	citizenship-v1 .jsonld	vaccination-vocab/ context/v1/index.json	Ja*	.	Ja*	* = je nach erstelltem Schema
Credential-Schema (Typ)	OpenApi Specification-Validator2022	1EdTechJsonSchemaValidator2019		-	-	JSON-Schema*	-	SHACL*	* = nicht in der VC Specifications Directory gelistet
Schema-Dokumentation (Vorlage)	ReSpec Doc.	1EdTech Doc.* auf Basis von ReSpec		ReSpec Documentation		via Plattform*	-	-	* = individueller Aufbau

6.3.1 Definition

Der Mechanismus der Definition von Credential-Schemata soll die Erstellung von Schemata systematisieren und konsistent gestalten. Dies umfasst die Bereitstellung einer definierten Vorlage für Credential-Schemata, die Festlegung der inhaltlichen Struktur von Credentials sowie die Einbindung von Kontextinformationen für das Schema und das Credential. Dadurch sollen Schema-Autoren in der Lage sein, präzise syntaktische und semantische Schemata für VCs zu erstellen. Die Definition eines Credential-Types von hoher Informationsqualität verlangt die Erstellung dreier Dateien: einer JASON-LD-Kontextdatei, eines Credential-Schemas und einer Schema-Dokumentations-Datei. Ein Credential-Type wird durch eine JSON-LD-Kontextdatei definiert, welche die Terminologie und semantische Genauigkeit des Credentials festlegt. Zusätzlich kann ein Credential-Schema zur Definition der syntaktischen Genauigkeit verwendet werden. Für die menschenlesbare Beschreibung des Credentials kann eine Schema-Dokumentation definiert werden.

Schema-Dokumentation In Tabelle 6.1 sind die Vorlagen für die verwendeten menschenlesbaren Beschreibungen entsprechend der IQ-Metrik I2 für die Schema-Definitionen der untersuchten Credential-Types aufgeführt. Das verwendete JSON-LD-Datenformat ist menschenlesbar, eine zusätzliche Schema-Dokumentation erhöht die Lesbarkeit. Um die IQ-Dimension einheitliche Darstel-

lung im Stil von Regelsteckbriefvorlagen zu gewährleisten, wird die Repräsentation des Credential-Types mittels HTML-Seiten unter Verwendung der Vorlagen von DIF Spec-Up[62] oder W3C ReSpec-Dokumentation[63] empfohlen.

JSON-LD-Kontextdatei Die Definition einer JSON-LD-Kontextdatei von hoher Informationsqualität verlangt die interoperable Definition der Terminologien des Credential-Types. Für das Wiederverwenden bestehender Vokabularien (IQs-Metrik I1) empfiehlt Steele eine progressionsbasierte Vorgehensweise, die von allgemeinen Begriffsdefinitionen zu spezifischeren Definitionen übergeht [Ori+23, §1.3.1]. Gemäß [Gre+20b] führt die W3C-JSON-LD-Working-Group eine Recommended-Context-Datei, die eine Liste von Präfixen und den dazugehörigen generischen Vokabularien enthält, auf die für die allgemeine Begriffsdefinition Bezug genommen werden kann. Die DIF hat ein Schema-Verzeichnis erstellt, das DIF-Schema-Directory [DJM22], welches verschiedenen Themengebieten definierte Vokabularien zuordnet. Dieses Verzeichnis wurde von der Arbeitsgruppe Claims-and-Credentials erstellt. Ein Beispiel daraus ist das Vokabular von Schema.org, welches den Themengebieten Person, Organisation und Gesellschaft zugeordnet ist. Das Schema-Verzeichnis dient als Orientierung für eine einheitliche Verwendung von Vokabularien.

In Tabelle 6.2 wird ein Vergleich zwischen den verschiedenen in Abschnitt 6.1 beschriebenen definierten VCs und der DIF-Schema-Directory durchgeführt. Das Comprehensive-Learner-Record bezieht sich auf das Vokabular von Open-Badges und wird daher in den folgenden Tabellen nicht separat aufgeführt.

Tabelle 6.2: Vergleich zwischen verwendeten Vokabularien (in Anlehnung an [DJM22])

		DIF Schema Directory	Traceability Vocabulary v0.1 [64]	Open Badges v3.0[65]	Citizenship Vocabulary v0.3[66]	Vaccination Certificate Vocabulary v0.1 [67]	Affiniti & Serto	SOyA	
Terminologie	**Vertikal**				**Vocabular**				**Beschreibung**
Person	Allgemein	Schema.org	Schema.org	Schema.org	Schema.org	Schema.org	-	-	Menschen, Individuen
Gesellschaft	Allgemein	Schema.org	Schema.org	-	-	-	-	-	Jede Form einer eingetragenen Körperschaft.
Organisation	Allgemein	Schema.org	Schema.org	-	-	-	-	-	Jede allgemeine Art von Organisation.
Organisation	Supply Chain	GS1[68]	GS1	-	-	-	-	-	Jede Organisation oder jedes Unternehmen, das in der Lieferkette tätig ist.
Organisation	Healthcare	HL7 FHIR[69]	-	-	-	-	-	-	Jedes Unternehmen, jede Institution, jede Gruppe im Gesundheitswesen usw.
Datatype	Allgemein		Schema.org & xsd[70]	xsd	xsd	xsd	Schema.org	xsd	

Es wurde festgestellt, dass sich das Vokabular von Schema.org als Vokabular für allgemeine Terminologien wie Person und Organisation eignet, während XMLSchema dies für die Definition von Datentypen leistet. Auch Steele und Kellogg empfehlen, Schema.org als Standardvokabular zu

[62] https://identity.foundation/spec-up/
[63] https://respec.org/docs/
[64] https://w3id.org/traceability/v1
[65] https://purl.imsglobal.org/spec/ob/v3p0/schema/json/ob_v3p0_getopenbadgecredentialsresponse_schema.json
[66] https://w3id.org/citizenship/v1
[67] https://w3id.org/vaccination/v1
[68] https://www.gs1.org/voc/
[69] https://hl7.org/fhir/
[70] https://www.w3.org/2001/XMLSchema

verwenden [Ori+23; Gre+20b, §1.3.1]. Die DIF-Schema-Directory befindet sich noch in der Entwicklungsphase, Vorschläge für Vokabularien zu einem bestimmten Themenkomplex können eingereicht werden, um eine breitere Abdeckung der Realität mit spezifischem Vokabular zu erreichen [Dec22b].

Existiert kein geeigneter Begriff in Schema.org oder kein passendes Vokabular in der DIF-Schema-Directory für eine Anwendung, kann der Schema-Autor nachfolgend auf die LOV[71]-Plattform zurückgreifen, um nach geeigneten Vokabular-Optionen zu suchen. Wenn in der Domäne, für die ein Credential definiert werden soll, bereits ein spezifisches Vokabular wie IATA[72] vorhanden ist, müssen anwendungsspezifische Bewertungen vorgenommen werden. Die LOV-Plattform ist ein Online-Katalog, der es ermöglicht, nach passenden Vokabularien für die Beschreibung von Linked-Data zu suchen. Zurzeit werden im LOV-Katalog 782 Vokabularien gelistet, die insgesamt rund 76.000 Begriffe umfassen und von der Ontology-Engineering-Group, die an der Universidad Politécnica de Madrid angesiedelt ist, verwaltet werden (vgl. [Ont23]). Zur Erfüllung der IQs-Metrik I1, Wiederverwendung bestehender Vokabularien (siehe B.1), wird im Semantic-Web empfohlen, von der Entwicklung eigener Begriffsdefinitionen abzusehen und stattdessen ausschließlich in Fällen gründlicher und erfolgloser Suche nach passenden Definitionen eine eigene zu erstellen. Die mehrfache Nutzung eines Attributs, beschrieben durch ein kontrolliertes Vokabular wie das Attribut *givenName* definiert durch `schema.org/givenName`, kann zu einer klareren Identifizierung dieses Attributs und damit wiederum zur Förderung der Interoperabilität auf Attributs-Ebene führen (vgl. [Cap+22, S. 17]). Die Untersuchung zur Bestimmung des Abdeckungsgrads der vorhandenen Credential-Types-Definitionen durch die Wiederverwendung bereits bestehender Vokabulare ist in Tabelle 6.3 gefasst, um die IQ-Metrik I1 anzugeben.

Tabelle 6.3: Vergleich: Abdeckung der Attribute durch definierte Vokabularien

	Credential-Type			
	Traceability Voc. v0.0	Open Badges v3.0	Citizenship Voc. v0.3	Vaccination Certificate Voc. v0.1
Ges.-Zahl. individueller Attribute	956	80	18	26
Definiert durch eigenes Vokabular	16%	33%	17%	54%
schema.org	33%	68%	83%	46%
UN/CEFACT[73]	30%			
List of Darwin Core terms[74]	11%			
GS1	4%			
Restlich definierte Attribute	5,33%			

Durch die Verwendung von schema.org als Referenz, durch RDF-Predicate, werden in den untersuchten Credential-Types-Definitionen mindestens 33% der Attribute definiert, was dazu beiträgt, einen gemeinsamen Kontext für Sender und Empfänger der Information aus unterschiedlichen Domänen zu schaffen.

[71] Linked Open Vocabularies (LOV)
[72] `https://github.com/IATA-Cargo/ONE-Record/tree/master/working_draft/API/json-ld`
[73] `https://service.unece.org/trade/uncefact/vocabulary/uncefact/`
[74] `https://dwc.tdwg.org/list/`

Basierend auf Tabelle 6.2 wird in Tabelle 6.4 die Terminologie für Personen von Schema.org in einen Vergleich mit den Credential-Types gestellt, um zu prüfen, ob die untersuchten Credential-Types dieselben Begrifflichkeiten verwenden und damit auf syntaktischer Ebene Interoperabilität zwischen ihnen herrscht.

Tabelle 6.4: Vergleich der verwendeten Begrifflichkeiten

Terminologie	**Credential-Type**			
schema.org/Person	Traceability Voc. v0.1	Open Badges v3.0	Citizenship Voc. v0.3	Vaccination Certificate Voc. v0.1
		Attributsbezeichner		
birthDate	-	-	birthDate	birthDate
birthPlace	-	-	-	-
address	address	address (eigene URI)	-	-
email	email	email	-	-
familyName	lastName & familyName	familyName	familyName	familyName
gender	-	-	gender	gender
givenName	firstName & givenName	givenName	givenName	givenName
honorificPrefix	-	honorificPrefix	-	-
honorificSuffix	-	honorificSuffix	-	-
image	image	image	image	image

Der Vergleich zeigt, dass die Bezeichnungen der Attribute, mit Ausnahme von *lastName* und *firstName* bei unterschiedlichen Credential-Types des Traceability-Vokabulars, anstelle von *familyName* und *givenName*, der Wiederverwendbarkeit von Terminologie in Übereinstimmung mit der IQs-Metrik I3 gerecht werden. Folglich ist eine syntaktische Interoperabilität gewährleistet. Bei der Konzeption einer JSON-LD-Kontextdatei ist es von zentraler Bedeutung, die innerhalb eines Credential-Types verwendete Terminologie sorgfältig zu gestalten. Dies kann durch den Einsatz von Vokabularen und deren Begrifflichkeiten erreicht werden, wobei ein Ansatz verfolgt wird, der von allgemeinen zu spezifischen Terminologien übergeht. Durch diese Herangehensweise wird eine nahtlose Integration der semantischen Ebene mit den Technologien des Semantic Web im Rahmen der JSON-LD-Kontextdatei sichergestellt.

Erstellen von Credential-Types über visuelle Editoren Die editorbasierten Plattformansätze von Affinidi und Serto, welche die geringste Einstiegshürde für das Definieren von Credential-Types über einen visuellen Editor aufweisen, werden nachfolgend untersucht. Die von den Editoren erstellten JSON-LD-Kontextdateien dienen auf semantischer Ebene ausschließlich zur Definition von Datentypen gemäß dem Schema.org-Vokabular und der Zuweisung von URLs zu entsprechender Plattform und deren Schema. Dies begünstigt die Bildung von Synonymen bei den Attributsbezeichnern der Schemata, denen entsprechende Schema-Suffixe zugeordnet werden, wie am Bezeichner *email* untersucht. Einige Beispiele für solche URL-Suffixe, die als *schema-id* für *email* repräsentiert werden, sind:

- https://schema-manager.prod.affinity-project.org/EventEligibilityV1-0.jsonld#email

- https://schema-manager.prod.affinity-project.org/EducationCertificateV1-0.jsonld#email
- https://schema.affinidi.com/EmployeeIDV1-4.jsonld#email

Im Quelltext 6.2 wird die Repräsentation des E-Mail-Attributs, definiert durch Serto oder Affinidi, dargestellt. Hier verweist das RDF-Predicate eines Attributs eines Schemas auf die zugehörige URL, die wiederum auf die entsprechende Plattform und *schema-id* verweist.

```
"https://plattform+schema-id.jsonld#email": {
        "@type": "http://schema.org/Text",
        "@value": "Max.Muster@mail.com"
}
```

Quelltext 6.2: Repräsentation des Attributsbezeichners *email* über Serto und Affinidi

In Tabelle 6.4 wird gezeigt, wie das Attribut *email* in den JSON-LD-Kontextdateien des Traceability-Vokabulars und denen von Open-Badges definiert ist. Beide Vokabulare beziehen sich auf *Schema.org/email* und schaffen dadurch einen gemeinsamen terminologischen Bezugspunkt, der die Interoperabilität fördert. Die erstellten RDF-Predicates für das Attribut *email* basieren auf den JSON-LD-Kontextdateien und entsprechen *schema.org/email*. Alle über die beiden Vokabulare erstellten Credential-Attribute sind über die genannte URL definiert und ihre Bedeutung wird einheitlich und eindeutig über diese als Bezugspunkt festgelegt.

Für eine weiterführende Analyse der Funktionen der Schema-Editoren wurden exemplarisch die definierten Credential-Types Vaccination-Certificate und Citizenship mithilfe der Editoren von Serto und Affinidi erstellt. Um die erzeugten Schemata zu überprüfen, wurden Online-Tools[75] eingesetzt, um eine einfache Replizierbarkeit der Versuche zu gewährleisten.

Mittels der Schema-Dokumentationen der definierten VCs wurden die Schemata und deren Attribute in den Editoren erstellt. Für die Validierung wurden die bereitgestellten Beispiele aus den jeweiligen Dokumentationen herangezogen, wie in [LSP21; LS20, §1.2, §1.2] beschrieben. Die Eingabedaten für die RDF-Auflösung wurden durch das Einfügen der entsprechenden JSON-LD-Kontext-URLs modifiziert . Beispielsweise wurde die URL `https://w3id.org/vaccination/v1` durch `https://schema.affinidi.com/PermanentResidentCardV1-0.jsonld` ersetzt.

Die Ergebnisse zeigen, dass mittels visueller Editoren valide JSON-Schema-Dateien erzeugt werden können. Bei der Auflösung der Credentials mithilfe der erzeugten JSON-LD-Kontextdateien traten jedoch Fehlermeldungen auf. Eine konsistente Auflösung des JSON-LD-Kontexts in Bezug auf die definierten Credentials konnte nicht sichergestellt werden, wie in Tabelle 6.5 dargestellt.

[75] Die erstellten JSON-Schema-Dateien wurden mit `https://json-schema.hyperjump.io/` validiert und die Auflösung des JSON-LD-Kontextes mit `https://json-ld.org/playground` geprüft.
[76] `https://w3c-ccg.github.io/citizenship-vocab/contexts/citizenship-v1.jsonld`
[77] `https://schemas.serto.id/schema/permanent-resident-card`
[78] `https://ui.schema.affinidi.com/schemas/PermanentResidentCardV1-0`
[79] `https://w3c-ccg.github.io/vaccination-vocab/context/v1/index.json`
[80] `https://schemas.serto.id/schema/vaccination-certificate-vc`
[81] `https://ui.schema.affinidi.com/schemas/VaccinationCertificateV1-0`

Tabelle 6.5: Schema-Editoren: Analyse der erstellten Schemata

Vorlage	Editor	Valides Credential-Schema	Credential Kontext auflösbar	Kommentar
Citizenship Vocabulary v0.3[76]	Serto[77]	Ja	Nein*	* = Systematische Überdefinition von credentialSubject mit dem VCDM
	Affinidi[78]	Ja	Ja	
Vaccination Certificate Vocabulary v0.1[79]	Serto[80]	Ja	Nein*	** = Überdefinition von type mit der Definition type des VCDM
	Affinidi[81]	Ja	Nein**	

Wenn der JSON-LD-Kontext nicht aufgelöst werden kann, ist es nicht möglich, die Daten im VC durch zusätzlichen Kontext anzureichern. Infolgedessen können keine RDF-Triples generiert werden, die die maschinelle Interpretierbarkeit des Kontextes ermöglichen. Infolge der nicht auflösbaren Attribute kann bei den LD-Proofs keine Signatur gebildet werden und deren Claims sind nicht verifizierbar, vgl. Abschnitt 5.2.

Die VCDM-Kontextdatei definiert das Credential als solches, *credentialSubject* ist eine darüber definierte Eigenschaft. Bei der Erstellung von JSON-LD-Kontextdateien durch Serto wird die über die VCDM-Kontextdatei festgelegte Eigenschaft *credentialSubject* ebenfalls in jeder generierten Kontextdatei definiert. Da alle VCs den Kontext von VCDM verwenden müssen, führt dies zu einer Überdefinition des *credentialSubject*. Diese Überdefinition verursacht Konflikte, da die JSON-LD-Kontextdatei von VCDM die *@protect*-Eigenschaft verwendet, um die Neuzuordnung von nachfolgenden Definitionen zu verhindern. Dies kann bei der Anwendung der *Expanded*-Funktion auf die JSON-LD-Daten der Credentials zu Problemen bei der Auflösung führen.

```
{
        "@context": {
                "w3ccred": "https://www.w3.org/2018/credentials#",
                ...
                "credentialSubject": {
                        "@id": "w3ccred:credentialSubject",
                        "@context": {
                                ...
```

Quelltext 6.3: Serto: Systemantische Überdefinition des *credentialSubject*

Die Kontextdatei des VCDM-V2 in der zweiten Version definiert alle Attribut-Präfixe, die nicht durch eine spezifische IRI definiert sind, durch *@vocab* als `https://www.w3.org/ns/credentials/issuer-dependent#`. Dadurch kann ein Credential ohne eine zugehörige JSON-LD-Kontextdatei, die dessen Type definiert, in RDF-Triples aufgelöst werden, und LD-Proofs für dessen Verifikation können generiert werden. Ohne eine Definition der Terminologie durch die Kontextdatei entstehen bei Attributsbezeichnern für bestimmte Attribute unter derselben URL Homonyme[82]. Im Allgemeinen wird durch die Verwendung der Kontextdatei des VCDM-V2 ein einheitlicher Zustand der RDF-Predicates der Attribute, die nicht in einer zusätzlichen JSON-LD-Kontextdatei definiert sind, geschaffen. Dies erleichtert die Abfrage von Attributen, ist beispielsweise das Attribut `http://schema.org/givenName` unauffindbar, kann nach `https://www.w3.org/ns/credentials/`

[82] Ein Homonym beschreibt verschiedene Bedeutungen eines Begriffs, die dieselbe Bezeichnung tragen, wie zum Beispiel Bank als Geldinstitut oder als Parkbank.

`issuer-dependent#givenName` gesucht werden (vgl. Abbildungen 6.6 und 6.7). In den Abbildungen wird der PermanentResidentCard-Credential ohne und mit Definition des Citizenship-Vocabulary dargestellt und der Unterschied in den RDF-Predicates der beiden Credentials verdeutlicht.

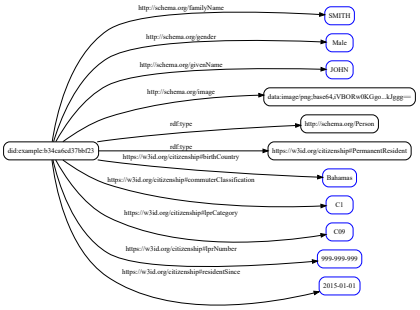

Abbildung 6.6: Auflösung des Credentials mit Citizenship-Vocabulary

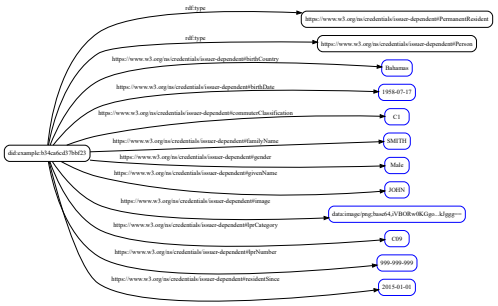

Abbildung 6.7: Auflösung des Credentials ohne Citizenship-Vocabulary

Im Vergleich zu den in Abschnitt 6.1 vorgestellten Credential-Types, die speziell für die Definition von Credentials entwickelt und im VC-Specifications-Directory aufgeführt sind, haben die von SOyA erstellten Dateien einen anderen Aufbau. Insbesondere die Auflösung von JSON-LD in RDF verdeutlicht dies, vor allem hinsichtlich der Integration von Vokabularien. In den Quelltexten 5.4, 6.4 und 6.5 wird das Attribute *birthDate* definiert.

SOyA, 6.4, definiert ihre Terminologie über die JSON-LD-Eigenschaft *@graph*. Beim Erstellen eines Credentials auf Basis einer SOyA-Kontextdatei muss die Integration der Eigenschaft *@graph* in die Struktur des Credentials beachtet werden. Um eine Auflösung in RDF-Triples anzuschließen, muss *@graph* wie in Quelltext 6.6 dargestellt integriert werden.

```
{
"@context": {
 "@import": ".../soya-context.json",
 "@version": 1.1,
 "@base": ".../zQm..G81GVisk(DRI)/"
 },
 "@graph": [
        ...
   {
   "@id": "birthDate",
   "@type": "owl:DatatypeProperty",
   "domain": "PermanentResident",
   "range": "xsd:date"
   },
    ...
```

Quelltext 6.4: SOyA: Attribute definition

```
{
"@context": {
"@version": 1.1,
 "@protected": true,
    ...
 "VaccineRecipient": {
  "@id": "https://w3id.org/vaccination#VaccineRecipient",
       "@context": {
       "@version": 1.1,
       "@protected": true,
       "id": "@id",
       "type": "@type",
   "birthDate": {
      "@id": "http://schema.org/birthDate",
      "@type": "http://www.w3.org/2001/XMLSchema#dateTime"
   ...
```

Quelltext 6.5: Vaccination-Vocab.: Attribute definition

Die Attribute des Credentials werden mithilfe von *@graph* der JSON-LD-Kontextdatei zugewiesen, die über SOyA erstellt wurde. Die dadurch entstehende Abweichung von der VCDM-Datenstruktur kann mittels einer separaten durch ein Transformation-Overlay erzeugten Kontextdatei ausgeglichen werden. Dies bedeutet jedoch mehr Komplexität und zusätzlichen Implementierungsaufwand für den Schema-Autor.

```
{
        "@context": [
        "https://www.w3.org/2018/credentials/v1",
        {
                "@version": 1.1,
                "@vocab": "https://soya.data-container.net/PermanentResidentCard/"
        }
        ],
        "id": "https://issuer.oidp.uscis.gov/credentials/83627465",
        "type": ["VerifiableCredential", "PermanentResidentCard"],
        "issuer": "did:example:28394728934792387",
        "identifier": "83627465",
        "name": "Permanent Resident Card",
        "description": "Government of Example Permanent Resident Card.",
        "issuanceDate": "2019-12-03T12:19:52Z",
        "expirationDate": "2029-12-03T12:19:52Z",
        "credentialSubject": {
                "@graph": [
                {
                        "@type": "PermanentResidentCard",
                        "id": "did:example:b34ca6cd37bbf23",
                        "type": [
                        "PermanentResident",
                        "Person"
                        ],
                ...
```

Quelltext 6.6: SOyA: Permanent-Resident-Card

Beim Abruf von Claims während der Verifikation, siehe A.1, wird im Presentation-Exchange [Dan+22a, §11] JSONPath für das Auslesen der Attribute verwendet. Die Abfrage des Attributs *birthDate* im Permanent-Resident-Card-Credential mittels JSONPath lautet *$.credentialSubject.birthDate*. Für ein Permanent-Resident-Card-Credential, das durch SOyA definiert ist, müsste die Abfrage hingegen wie folgt lauten: *$.credentialSubject.@graph[0].birthDate*. Die Divergenz in der Credentialstruktur erfordert bei Credential-Types, welche mittels SOyA definiert sind, während des Verifikationsprozesses Berücksichtigung, um eine adäquate Funktionalität der Abfrage sicherzustellen.

Die Software, die das Traceability-Vocabulary erzeugt, wie in Abschnitt 6.1 beschrieben, generiert eine JSON-LD-Kontextdatei basierend auf den Credential-Schemata und Definitionen des verwendeten externen Vokabulars. Die Definition der Credential-Types richtet sich nach der Spezifikation des VCDM.

Credential-Layout Die DIF hat zwei Spezifikationen veröffentlicht, um eine einheitliche Darstellung von Credentials zu ermöglichen, das Credential-Manifest und Wallet-Rendering. Beide Spezifikationen wurden von denselben Autoren verfasst und befinden sich noch im Entwicklungsstadium. Das Credential-Manifest [Dan+23] beschreibt ein Datenmodell, das ein Subjekt an einen Issuer

bereitstellen muss, sowie die Darstellungs- und Stilpräferenzen des Issuers, deren Wahl dem Ausstellungsprozess eines Credentials vorgelagert ist. Das Identity-Wallet des Holders kann dadurch den Ausstellungsprozess verstehen und aushandeln. Die Spezifikation Wallet-Rendering [Dan+22b] legt eine Reihe an Stil- und Datendarstellungshinweisen für Credentials fest, um eine einheitliche Darstellung in den Identity-Wallets sicherzustellen. Dies dient dem Zweck, UI[83]-Elemente im Zusammenhang mit Entitäten und Daten konsistent darzustellen. Die Wallet-Rendering-Spezifikation gibt eine einheitliche Darstellung von Credentials vor, die auch im Credential-Manifest genutzt wird. Hierzu gehören UI-Elemente wie das Logo des Issuers und die Hintergrundfarbe des Credentials. Ziel der Wallet-Rendering-Spezifikation ist die Verbesserung der Lesbarkeit von Credentials für den Menschen. Die Absicht einer einheitlichen Darstellung von Credentials ergibt einen sinnvollen Anwendungsfall für einen Overlay-Typ. Durch die Definition eines entsprechenden Transformation-Overlays würde dieses in die durch die Spezifikationen vorgegebene Form gebracht werden, um eine bestimmte Darstellung des Credentials mittels Overlay-Strukturen zu ermöglichen.

Credential-Schema Das Credential-Schema gewährleistet auf Grundlage der JSON-LD-Kontextdatei die syntaktische Genauigkeit und sichert damit die semantische Genauigkeit des Credential-Types. Es spezifiziert den Inhalt des Credential-Types, indem es die erforderlichen Attribute, deren Umfang, Struktur und Eingaberegeln definiert. Dabei ist es wichtig, dass das Schema die Struktur des Credentials so gestaltet, dass die JSON-LD-Kontextdatei erfolgreich aufgelöst werden kann. Dies ermöglicht den Schema-Nutzern, definierte Credentials zu verwenden. In diesem Zusammenhang dient das Credential-Schema dem Issuer bei der Ausstellung von VCs, während der Verifier das Credential-Schema als Kontrollinstanz für die Überprüfung der VP nutzen kann und abgleicht, ob die VP gemäß dem Credential-Schema strukturiert ist. Ein Credential-Schema kann entweder die Definition des gesamten Credentials oder nur Teile davon abdecken. In den untersuchten Ansätzen wird JSON-Schema für die Definition von Credential-Schemata eingesetzt, mit Ausnahme von SOyA, das SHACL verwendet. Da die JSON-LD-Kontextdatei im VCDM den Schwerpunkt der Schema-Definition darstellt, wird das Credential-Schema in dieser Arbeit nicht weiter untersucht.

Die Affinidi-Plattform ermöglicht es Schema-Autoren, Schemata zu erstellen, und Schema-Nutzern, diese Schemata zu verwenden. Dabei kann ein visueller Editor zur Erstellung von Schemata verwendet werden, bei dessen Verwendung müssen Abstriche bei der semantischen Genauigkeit in Kauf genommen werden. Die über diese Plattform generierten JSON-LD-Kontextdateien repräsentieren Literale, die durch die Datentypen des Vokabulars von Schema.org definiert sind. Eine eindeutige semantische Repräsentation der Attribute des Credentials wird nicht erreicht, da es keine automatisierte Funktion für die Zuweisung zu externen Vokabularien gibt, um die IQ-Metrik I1 zu erfüllen, welche die Zuordnung von Datentypen durch Schema.org überschreitet. Im gegenwärtigen Entwicklungsstadium der Editoren ist es notwendig, die JSON-LD-Kontextdatei manuell zu erstellen. Dies kann beispielsweise durch manuelles Bearbeiten oder durch den strukturierten Einsatz von Werkzeugen wie SOyA, JSON-LD-Context-Plus-Schema-Playground von Serto oder dem Traceability-Vocabulary-Ansatz geschehen, um eine semantisch präzise Terminologie für einen Credential-Type zu definieren.

Overlay-Struktur und VCDM OCA ist aufgrund der fehlenden Integration von Semantic-Web-Technologien nicht für die Verwendung von JSON-LD-basierten Credentials geeignet. Da eine Ontologie basierend auf dem Semantic-Web fehlt, muss das OCA-Datenmodell für jede Anwender-

[83] User Interface

software implementiert werden, um die Kontextinformationen aus den Overlays interpretierbar zu machen, was der Interoperabilität entgegenwirkt. SOyA hingegen bietet durch die Definition seiner Overlay-Strukturen, die mittels einer Ontologie auf Basis von OWL und RDFS gebildet werden, die Möglichkeit der Repräsentation und Interoperabilität durch Semantic-Web-Technologien. Wie in den Abschnitten 6.2.2 und 6.2.2 beschrieben, bringt das Erstellen und Nutzen von Overlay-Strukturen einen zusätzlichen Aufwand mit sich. Diese Strukturen ermöglichen jedoch eine höhere Flexibilität bei der Abbildung von Kontextinformationen eines Schemas. Der durch die Overlay-Strukturen geschaffene Kontext setzt die Integration der Strukturen in das eigene System voraus, ansonsten ist der zusätzliche Kontext nicht interpretierbar und strukturbedingte Vorteile bleiben ungenutzt. Die Maschinenlesbarkeit des Credential-Formats VCDM wird durch das Semantic-Web ermöglicht, Overlay-Strukturen sind für deren Interpretierbarkeit nicht ausschlaggebend.

SOyA kann durch das Transformation-Overlay in Kombination mit einer SOyA-Base verschiedene Credential-Formate für ein Schema nutzbar machen. Für die menschliche Lesbarkeit bieten die Overlay-Ansätze Form-Overlays an, die Eingabeformulare grafisch eindeutig darstellen. Overlays könnten zur Abbildung von Credential-Manifests und Wallet-Renderings eingesetzt werden, um eine einheitliche Struktur für die Darstellung und Übermittlung von Credentials bereitzustellen. Dabei dient die Overlay-Basis als Bezugspunkt für unterschiedliche Schema-Autoren. Die Overlays des Schemas können unabhängig und domänenspezifisch bearbeitet werden.

Die Bewertung von Overlay-Strukturen hängt von verschiedenen Faktoren ab, wie der Frage, ob die Lesbarkeit für Mensch oder Maschine im Vordergrund steht. Overlay-Strukturen erlauben die Definition von visuellen Elementen innerhalb der Overlays. Insbesondere bietet OCA durch ein Credential-Layout- und ein Form-Overlay umfangreiche Funktionen hinsichtlich der grafischen Repräsentation. JSON-LD bietet ebenfalls Wege Attributen zusätzlichen Kontext zu verleihen. Eine Möglichkeit ist der Einsatz von *@language* für die Verwendung von Sprachkennzeichnungen, wie in [Gre+20a, §4.2.4] beschrieben. Eine andere sind die Mehrfachbeschreibungen von Attributen, die durch die Verwendung der *label*-Eigenschaft realisiert werden (vgl. [Gre+20a, §4.4]). Eine eindeutigere Verständlichkeit eines Attributs erhöht seine Nutzbarkeit in unterschiedlichen Domänen. In Kombination mit der *label*-Eigenschaft kann die *@nest*-Eigenschaft in JSON-LD genutzt werden, um verschachtelte Beschreibungen zu organisieren und die Struktur der Kontextinformation übersichtlicher und leichter verständlich zu gestalten. Zur Erhöhung der semantischen Genauigkeit der *label*-Struktur kann das SKOS-Vokabular eingesetzt werden (vgl. [AS09, §5]).

Polymorphes Schema Die Modularität, die Overlay-Strukturen bieten, kann nicht ausschließlich über die JSON-LD-Kontextdatei abgebildet werden. Dies betrifft zum Beispiel die Bezeichnungen von Attributen in verschiedenen Nationalsprachen. Die Datenharmonisierung von Eingabedaten kann durch ein Transformation-Overlay erreicht werden. Die angesprochenen Punkte zeigen, dass die Informationsrepräsentation technisch durch JSON-LD abgedeckt werden kann. Es bedarf jedoch einer Management-Ebene, die Zusammenarbeit organisiert und den Prozess vorgibt. Die Nutzung der vorgestellten Overlay-Struktur ermöglicht eine systematische Modularisierung von Information verschiedener Parteien gebunden an eine gemeinsamen Basis, indem Werkzeuge für die einheitliche Umsetzung der Datenmodelle entwickelt werden. Vorab steht die organisatorische Einigung auf ein einheitliches Vorgehen bei der Kontextbearbeitung. Entweder wird gemeinsam eine JSON-LD-Kontextdatei entwickelt oder der Basis wird durch Overlays individuell Kontextinformation hinzugefügt. Die Flexibilität und Modularität, die Overlay-Strukturen bieten, ermöglichen die Schaffung eines polymorphen Schemas, wie in Abbildung 6.8 veranschaulicht.

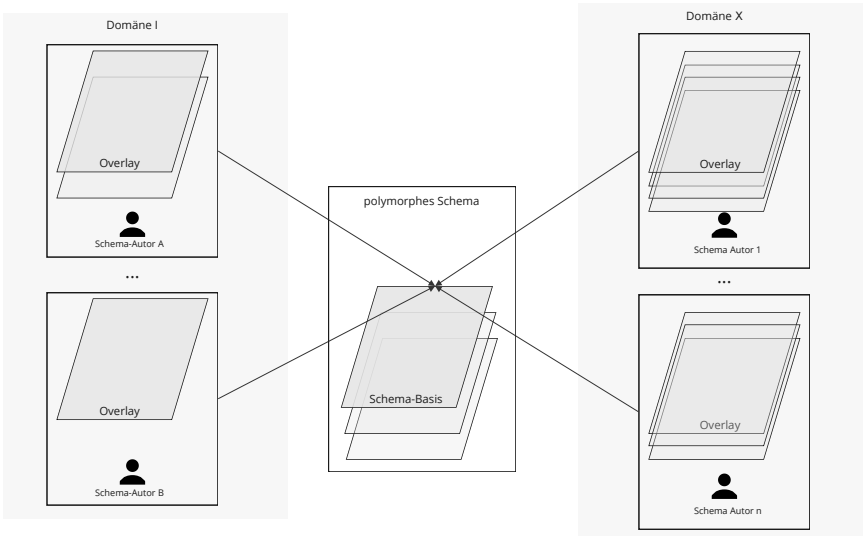

Abbildung 6.8: Overlay-Struktur: Eine Basis - Mehrere Autoren

Ausgehend von einer Schema-Basis, welche die Attribute des Schemas definiert, können Schema-Autoren durch Hinzufügen von Overlays die Anwendbarkeit für ihren spezifischen Anwendungsfall und ihrer Domäne entsprechend abdecken. Die Umsetzung eines solchen multilateralen Erstellungs- und Bearbeitungsprozesses verlangt den Schema-Autoren Disziplin ab. Der Bezug zur bestehenden und gemeinsamen Basis ist zu wahren und das Vorhandensein von erforderlichen Overlays vor dem Hinzufügen eigener zu prüfen, die entweder durch Anreizsysteme geschaffen oder durch organisatorische Strukturen vorgegeben werden muss. Bislang existieren weder Anreizsysteme noch organisatorische Strukturen in Form von Standards, die den Einsatz von Overlay-Strukturen begünstigen oder vorschreiben. Die Komplexität des Overlay-Systems stellt bei der Umsetzung von Overlay-Strukturen im Zusammenhang mit dem VCDM eine große Einstiegshürde für Schema-Autoren und Schema-Nutzer dar. Im Gegensatz zu Overlay-Strukturen verfolgt der Ansatz des Traceability-Vocabularys eine monolithische Kontextstrategie, in der die Schemata der Credentials ihre Credential-Types definieren. Die Entwicklung dieses Entwurfs erfolgt gemeinschaftlich und transparent über GitHub, um die verschiedenen Aspekte der Traceability durch Credentials abzubilden.

6.3.2 Hinterlegung

Der Mechanismus der Hinterlegung von Schemata in VDR zielt auf eine effektive Verwaltung von Credential-Types ab. Schema-Autoren können die Ergebnisse des vorhergehenden Definitionsmechanismus, JSON-LD-Kontexte, Credential-Schemata und entsprechende Dokumentationen, hinterlegen. Dabei sind aus den Prinzipien nach Allen [All16] Persistenz und Portabilität zu berücksichtigen, um eine dezentrale und interoperable Infrastruktur zu schaffen, siehe Abschnitt 2.1. Zusätzliche Kriterien sind die Kosten der Hinterlegung, die Leistungsfähigkeit der VDR beschrieben durch die IQ-Metriken P1 bis P3, Verfügbarkeit und Bearbeitbarkeit der hinterlegten Schemata. In welcher Art von VDR, als Speicherort, Schemata im Laufe eines Schemaprozesses hinterlegt werden, wäre auf einer organisatorischen Ebene, im Rahmen eines Schema-Managements, zu regeln. Im gegenwärtigen Stadium des ToIP-Stacks sind der Schemaprozess und das Schema-Management in groben Zügen beschrieben, wie dem Abschnitt 3.1 zu entnehmen ist. Die Einstufung, welche Systeme als VDR verwendet werden können, muss von der Government-Authority des jeweiligen Ökosystems festgelegt werden, um das Vertrauen im Ökosystem aufrechtzuerhalten. Das Schema-Management

muss die Infrastruktur berücksichtigen, wobei entweder Verlinkungen von Schemata zu anderen Ökosystemen oder das Schema selbst im Ökosystem hinterlegt werden können, falls externe VDRs nicht den eigenen Anforderungen entsprechen. Das Schema-Management soll laut ToIP-Stack über Trust-Registries abgebildet werden. Während des Bearbeitungs- und Einsatzzyklus können Schemata an unterschiedlichen Speicherorten abgelegt werden. Dabei ist es möglich, dass ein Schema gleichzeitig in mehreren VDRs gespeichert ist. Die simultane Speicherung unterstützt die Erfüllung der Anforderungen verschiedener Ökosysteme bei der systemübergreifenden Verwendung eines Schemas. In Abschnitt 3.1 werden potenzielle VDRs wie Blockchains, CAS und Versionskontrollsoftware genannt, und in Tabelle 6.6 verschiedene Ausprägungen von VDRs gelistet und anschließend ausgewertet. Die Auswahl der gelisteten VDR Ausprägungen ist bis auf IPFS auf ihre Anwendung in den untersuchten Ansätzen zurückzuführen.

Tabelle 6.6: Vergleich: Hinterlegung

	VDR-Typ						
	CAS	Git-Repo.	Blockchain		Docker-Container Basiert		**Kommentar**
VDR Ausprägung	IPFS	Github	cheqd	Dock.io	OCA-Container	Semantic-Container	
genutzt von	-	Traceability-Vocab.	Serto*	Affinidi*	OCA	SOyA	*= angedacht
JSON-LD-Kontextdatei	x	x	x*	-	-	x	*= in Umsetzung
Credential-Schema	x	x	x	x	x	x	
Schema-Doku.*	x	x	-	-	-	-	*= HTML-Datei
Datenintegrität via	CID[84]*	Git	Blockchain		SAID*	DRI*	*= Identifikator
Dezentralisiert	x	-	x			-*	*= je nach Implementierung
Bearbeitbarkeit	-	hohe	-	-	-	-	
Auf ähnlich Schemata prüfen	-	-	-	-	-	soya similar	
Fallen Kosten je Hinterlegung an?	Nein*	Nein	x		Nein*	Nein*	*= Kosten der eigenen Infrastruktur

Bei der Entwicklung von Caching-Strategien für Identity-Wallets bietet sich ein monolithischer Ansatz an, der die Terminologie eines Vertikals mithilfe einer einzigen JSON-LD-Kontextdatei abbildet, wie im Traceability-Vokabular angewendet. Um die Offline-Verfügbarkeit von LD-Proofs für Credentials des Traceability-Vokabulars sicherzustellen, genügt es, lediglich zwei Dateien, die VCDM-JSON-LD- und Traceability-JSON-LD-Kontextdatei, fest im Identity-Wallet zu codieren oder auf dem jeweiligen Gerät durch Caching zu hinterlegen. IPFS kann zur Speicherung von JSON-LD-Kontextdateien, Credential-Schemata und Schema-Dokumentationen eingesetzt werden. Dank der Peer-to-Peer-Architektur von IPFS entstehen Vorteile gegenüber konventionellen Ansätzen hinsichtlich Persistenz, da der Datenaustausch unabhängig vom Standort und durch ein dezentrales System ermöglicht wird. In Offline-Situationen können Schemata zwischen Identity-Wallets ausgetauscht werden, sofern diese über eine integrierte IPFS-Knoteninstanz und den Transport der Schemata mittels WebSockets verfügen (vgl. [Intc]). Ohne eine IPFS-Knoteninstanz erfolgt der Abruf von Schemata über ein HTTP-Gateway [Intb]. Um die Verfügbarkeit der Schemata zu gewährleisten, muss eine IPFS-Knoteninstanz die Informationen bereitstellen. Dieser Ansatz steht im Gegensatz zu einer Blockchain, bei der die gesamte Datenhistorie des Systems über das Netzwerk gespeichert und bereitgestellt wird.

Cheqd und Dock.io sind public Blockchains, die im Gegensatz zur Hyperledger-Indy das Hinterlegen von Schemata des Credential-Formates VCDM unterstützen. Cheqd ist ein in London ansässiges Unternehmen, das im Jahr 2021 gegründet wurde und an einem Blockchain-basierten Zahlungsmechanismus für vertrauenswürdige Daten arbeitet. Seit Ende 2021 hat Cheqd Partnerschaften mit

[84] Content Identifier (CID)

verschiedenen SSI-Unternehmen wie DanubeTech, Spherity, esatus, Serto und Evernym geschlossen, wie in [che22] berichtet wird. Serto plant, seine Schemata in das Cheqd-Netzwerk zu integrieren. Dies in der Absicht dem Nutzer die Möglichkeit der einfachen Integration seiner Schemata in einer weiteren VDR zu bieten (siehe Abschnitt 6.2.1). Cheqd nutzt die Cosmos-SDK-Blockchain [Edw21]. Die Cheqd-Blockchain bietet die Möglichkeit, Credential-Formate wie AnonCreds zu hinterlegen und zukünftig ist auch die Hinterlegung VCDM-konformer Credential-Types geplant. Die Funktionalität für LD-Proofs mit dem Veramo-SDK soll im Jahr 2023 implementiert werden, wie in [Pow23] beschrieben wird. Dock.io basiert auf der Substrate-SDK-Blockchain [doc22a] und ermöglicht das Hinterlegen von Credential-Schemata [doc22b]. Mithilfe der Plattform Affinidi erstellte Schemata können in die Dock.io-Plattform integriert werden, indem deren Template-Funktion verwendet wird [doc23].

Die Auswahl eines geeigneten Speicherorts für die Hinterlegung von Schemata hängt von verschiedenen Faktoren wie dem Entwicklungsstadium und einer eventuellen regulierenden Funktion des Schemas oder der zuständigen Governance-Autorität ab. In der frühen Entwicklungsphase, wenn der Bearbeitungsaufwand des Schemas wie die Zahl der daran beteiligten Parteien hoch ist, sind für eine domänenunabhängige Hinterlegung Git-Repository-Hosting-Dienste wie GitHub in Verbindung mit Zenodo[85] empfehlenswert. GitHub gestattet eine automatisierte Versionierung, Kommunikation und Dokumentation über die GitHub-Infrastruktur (vgl. Traceability-Vocabulary). Abhängig von der Art bzw. dem Grad der Regulierung des Anwendungsbereichs eines Schemas ergeben sich unterschiedliche Anforderungen hinsichtlich der Hinterlegung. In unregulierten Bereichen liegt der Fokus auf übergreifender Zusammenarbeit und Bearbeitbarkeit des Objekts wie beim Traceability-Vocabulary, weshalb GitHub als Ablageort ideal ist. Bestehen regulatorische Anforderungen, wie bei der Europäischen Blockchain Service Infrastructure (EBSI), stehen hingegen Datenintegrität und Unveränderlichkeit des Schemas im Vordergrund. Im Rahmen der EBSI wird eine Trusted-Schema-Registry (TSR) als Verkörperung eines Ökosystem-basierten Schema-Managments entwickelt (vgl. [Eur]).

Da Schemata in unterschiedlichen VDRs innerhalb verschiedener SSI-Ökosysteme hinterlegt werden können und die Wahl des jeweiligen Speicherortes unter dem Einfluss zahlreicher Faktoren steht, sind Persistenz und Portabilität der Schemata für den Mechanismus der Hinterlegung von entscheidender Bedeutung. In Zusammenhang mit Persistenz und Portabilität eines Schemas stehen dessen Metadaten und Identifikatoren. Beide Faktoren können durch die Verwendung eines Content-Addressable-Identifiers, wie beispielsweise des DRIs oder SAIDs, verbessert werden. Der Identifier wird auf Basis des Schemainhalts generiert, die Anwendung einer Hashfunktion fördert dabei die Erkennung von Inhaltsänderungen während der Übertragung an einen alternativen Speicherort. Spezifische Provenienz-Metadaten sind beschreibende Daten, die die Beziehung zur Ableitung eines Datenobjekts angeben (vgl. [W3C10]). Metadaten dieser Art enthalten Informationen über das Datenobjekt, wie die Datenhistorie, das Erstellungsdatum, die Autoren und eingesetzte Methoden. Durch die Bereitstellung dieser Informationen tragen Provenienz-Metadaten zur Persistenz der Schemata bei, indem sie das Verständnis der Daten nachvollziehbar machen, zum Beispiel lässt sich über sie die Methode, die für die Validierung des Schemas eingesetzt werden soll, ableiten. Bei Datenaustausch unterstützen sie zudem die Portabilität, indem sie die Integration und Nutzung der Schemadaten in und zwischen verschiedenen Systemen und Anwendungen durch die enthaltenen Provenienz-Informationen erleichtern.

[85] `https://zenodo.org/`

6.3.3 Abruf

Der Mechanismus des Abrufs von Credential-Schemata aus einer VDR soll im Schemaprozess sowohl Schema-Autoren als auch -Nutzern ermöglichen, nach bestehenden Schemata zu suchen und diese für ihre Anwendungsfälle abzurufen. Der Zugriff auf die Schemata sollte für alle Schema-Nutzer ohne Zugangsbeschränkungen sein. Der Mechanismus Abruf bündelt die Funktionen der Suche nach geeigneten Schemata, das Abrufen dieser Credential-Schemata für die Ausstellung von VCs, Bereitstellung von JSON-LD-Kontextdateien für die RDF-Auflösung sowie die einer URL, über die die Schema-Dokumentation abrufbar ist. In Tabelle 6.7 sind die untersuchten Ansätze aufgeführt und hinsichtlich des Abrufmechanismus bewertet. Die Ansätze von Open-Badges, CLR, Citizenship und Vaccination definieren lediglich einen einzigen Credential-Type. Das CredentialSchema2022 wird in der Tabelle nicht gelistet, da in der Spezifikation nur Vorschläge unterbreitet werden, aber keine konkreten Implementierungsvorgaben zum Abruf dokumentiert sind. Die untersuchten Ansätze, die APIs einsetzen, basieren durchgängig auf der OpenAPI-Spezifikation, welche eine einheitliche Dokumentation der Funktionalität der APIs erbringt. Bei der Metrik A1, dem Dereferenzieren einer auf einen Credential-Type verweisenden URL, ist es wichtig, darauf zu achten, dass standardmäßig der MIME[86]-Typ *application/ld+json* zurückgegeben wird (vgl. [Gre+20a, §6]). Dadurch können JSON-LD-Prozessoren den Kontext in Form der JSON-LD-Kontextdatei verarbeiten. Fordert ein User-Agent jedoch HTML als Schema-Dokumentation an, sollte menschenlesbarer Text bereitgestellt werden, der die Bedeutung der verwendeten Begriffe und deren Zuordnung, also die Terminologie des Credential-Types, erläutert (vgl. [Cha+19, §10.1]).

Die Nutzung einer DID als Identifikator für ein Credential-Schema kann die Anforderungen der Informationsqualitätsmetrik A6, Zugänglichkeit von Metadaten, erfüllen, indem sie über die bloße Verfügbarkeit von Daten hinaus gewährleistet wird. Ein Beispiel für eine Lösung, die Schemata gemäß den Informationsqualitätsmetriken A6 und A7 zugänglich macht, ist Zenodo. Dieses Online-Repository verwendet DOIs[87], um die Auffindbarkeit und Verfügbarkeit von Metadaten sicherzustellen. GitHub-Repositories und Zenodo können miteinander verknüpft werden, sodass Credential-Types einfach über eine DOI und ein zusätzliches Repository abgebildet werden können.

Tabelle 6.7: Vergleich: Mechanismen des Abrufs

		Credential-Type					Editor		Overlay	
Abk.	**IQ-Metrik**	Traceability Voc. v0.1	OpenBadges v3.0	Comprehensive Learner Record v2	Citizenship Voc. v0.3	Vaccination Certificate Voc. v0.1	Serto	Affinidi	OCA	SOyA
A1	Möglichkeit der Dereferenzierung von Ressourcen	HTTP-URIs							-	HTTP-URIs
A2	Bereitstellung eines RDF-Exports + Credential-Schema-Exports	Data Dump via GitHub und API-GET	Data Dump via GitHub	-	Data Dump via GitHub		-		Data-Dump via API-GET	Data Dump via API-GET und CLI
A3	Bereitstellung eines öffentlichen SPARQL-Endpunktes	-								Erweiterbar um SPARQL-Endpunkt
A4	VDR is API-fähig	Ja[88]		-				Ja[89]	Ja[90]	Ja[91]
A5	Bereitstellung der Daten durch HTTPS-GET-Methode	Ja							Ja, von Implementierung abhängig	
UI-Explorer		über Schema-Doku.	Nein, nur Abbildung eines einzelnen Credential-Type				über die Plattform		-	
Schema Suchfunktion		-					-	via API-GET	via API-GET und Elastic-Search	-

[86] Multipurpose Internet Mail Extensions (MIME)
[87] Digital Object Identifier (DOI)
[88] `https://w3c-ccg.github.io/traceability-vocab/openapi/`
[89] `https://affinidi-schema-manager.prod.affinity-project.org/api-docs/`
[90] `https://repository.oca.argo.colossi.network/`
[91] `https://api-docs.ownyourdata.eu/soya/`

Über die Plattformansätze von Serto und Affinidi können Schemata mithilfe des UI-Explorers in visualisierter Form und damit benutzerfreundlich dargestellt werden. Im Vergleich erfüllt SOyA die meisten IQ-Metriken, die im Zusammenhang mit dem Mechanismus Abruf stehen, und stellt den besten Ansatz für den Abruf von Linked-Data dar.

6.3.4 Versionierung

Der Mechanismus der Versionierung soll im Schemaprozess gewährleisten, dass Schema-Nutzer im Laufe der Zeit Änderungen nachverfolgen und auf ältere Versionen zurückgreifen können, während Schema-Autoren Schemata weiterentwickeln können. In Tabelle 6.8 sind die untersuchten Ansätze gelistet. Die Versionierung der JSON-LD-Kontextdateien bei den beschriebenen Credential-Type-Definitionen erfolgt durch Semantic-Versioning in der URL, wie in Tabelle 6.1 dargestellt. 1EdTech weist die präziseste Versionierung auf, da es durch das *MAJOR.MINOR.PATCH*-Schema, nach der Semantic-Version 2.0 (vgl. [Pre22]), definiert ist. SOyA versioniert die erstellten JSON-LD-Dateien, Kontext und Credential-Schema, über die DRI und den Indexer der SOyA-Base, siehe Abschnitt 6.2.2. In den Spezifikationen von OpenApiSpecificationValidator2022 und EdTechJsonSchemaValidator2019 wird die Versionierung von Credential-Schemata nicht thematisiert. Bei CredentialSchema2022 erfolgt die Versionierung über das Attribut *version* durch *Modell.Revision* innerhalb der Metadaten des Schemas und in der DID des Schemas, wie im Abschnitt 5.2 beschrieben. Die Schemata des Traceability-Vocabularys, Open-Badges, Citizenship und Vaccination-Certificate werden über das GitHub-Repository versioniert. OCA versioniert die erstellten Credential-Schemata über den SAID, siehe Abschnitt 6.2.2. Bei Serto und Affinidi werden die Credential-Schemata über die URL und die Metadaten versioniert. Für die Gewährleistung der Datenintegrität schlägt CredentialSchema2022 die Verwendung von JOSE-Signaturen vor, während SOyA und OCA jene durch Hashlinks sicherstellen.

Tabelle 6.8: Vergleich: Mechanismen der Versionierung

<table>
<tr><th></th><th></th><th colspan="5">Credential-Type</th><th colspan="2">Editor</th><th colspan="2">Overlay</th><th></th></tr>
<tr><th></th><th></th><th>Traceability Voc. v0.1</th><th>OpenBadges v3.0</th><th>Comprehensive Learner Record v2</th><th>Citizenship Voc. v0.3</th><th>Vaccination Certificate Voc. v0.1</th><th>Serto & Affinidi</th><th>OCA</th><th>SOyA</th><th>Credential Schema2022</th></tr>
<tr><td>JSON-LD-Kontextdatei</td><td colspan="5">Semantic-Versioning (SemVer) über die URL</td><td></td><td>-</td><td rowspan="2">Über Indexer der SOyA-Base und Fixierung der Versionen durch DRIs.</td><td>-</td></tr>
<tr><td>Credential-Schema (Typ)</td><td>-</td><td colspan="3">SemVer und Metadaten</td><td>-</td><td>-</td><td>SemVer und Metadaten</td><td>Fixierung der Versionen durch SAIDs.</td><td>SemVer und Metadaten</td></tr>
<tr><td rowspan="2">Schema- Dokumentation (Vorlage)</td><td>Deprecation Policy</td><td colspan="7">-</td></tr>
<tr><td>Migration Guides</td><td colspan="7">-</td></tr>
<tr><td>Versionskontrollsystem</td><td colspan="2">GitHub</td><td>-</td><td colspan="2">GitHub</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Changelog</td><td rowspan="2">via GitHub: JSON-LD-Kontext, CredentialSchema.yml</td><td rowspan="2">via GitHub: JSON-LD-Kontext, CredentialSchema.json</td><td>-</td><td rowspan="2" colspan="2">via GitHub: JSON-LD-Kontext</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Schema-historie</td><td>-</td><td colspan="2">-</td><td colspan="2">SOyA-Base Indexer</td><td>-</td></tr>
</table>

Um die Versionierung von JSON-LD-Dateien mithilfe von Metadaten sicherzustellen und die Erfüllung der IQ-Metriken T2, T3 und T5 zu gewährleisten, kann das Vokabular PROV-O[92] herangezogen werden. PROV-O ist eine Ontologie, die dazu dient, Provenienzinformationen konsistent und standardisiert zu beschreiben, erfassen und auszutauschen (vgl. [Tim+13]). Für die Berücksichtigung von Metadaten im Credential-Schema wird empfohlen, die im CredentialSchema2022 festgelegten Attribute für Metadaten zu nutzen, siehe Abschnitt 5.2.

Der Mechanismus der Versionierung wird bei SOyA als eigenständige Lösung am besten umgesetzt. Dahingehend, dass der Schema-Autor ein Schema weiterentwickeln kann und SOyA die Versionierung automatisiert durchführt. Über die SOyA-Base Info erhält der Schema-Nutzer eine Übersicht

[92] Provenance Ontology (PROV-O)

der Schemahistorie, wobei die verschiedenen Versionen über die DRI eindeutig identifizierbar sind. Auf diese Weise kann auch die Datenintegrität des Schemas gemäß IQ-Metrik S3 überprüft werden. Über die DRI kann das Schema im ausgestellten Credential eindeutig angegeben und bei der Verifikation für die Überprüfung der Informationsvollständigkeit genutzt werden. Um die Datenintegrität außerhalb des Credentials weiter zu erhöhen und den Anforderungen der IQ-Metrik S1 gerecht zu werden, ist es notwendig, vor der Veröffentlichung des Schemas Signaturen über das Schema zu bilden. Um Vertrauen in die Signatur aufzubauen, ist es im Kontext eines SSI-Ökosystems wichtig, ein Schema-Management zu etablieren, das die Autorenschaft des Signierenden identifizierbar und bewertbar macht.

7 Fazit

In dieser Arbeit wurde der Schemaprozess mit den Mechanismen Definition, Hinterlegung, Abruf und Versionierung von Credential-Schemata untersucht. Ausgehend von Kapitel 3 wurden Schemata im Ökosystem gemäß dem ToIP-Stack eingeordnet. Es wurde festgestellt, dass für die Umsetzung des Schemaprozesses als Teil des Schema-Managements Trust-Registries entwickelt werden müssen. Des Weiteren wurde der Zusammenhang zwischen Vertrauensentscheidungen, Kontext und Schema hergestellt. In Kapitel 4 wird der Schemaprozess in Akteure und Mechanismen untergliedert und systematisiert, wobei die Abhängigkeiten zwischen Akteuren, Funktionen und Schema dargestellt werden. Ein Bewertungsrahmen für die Mechanismen des Schemaprozesses wurde entwickelt, der Schemata und Informationsqualität in Beziehung setzt und daraus Metriken ableitet. Kapitel 5 untersucht den Schemaprozess der Credential-Formate AnonCreds und VCDM und analysiert die Darstellung von Informationen auf semantischen, syntaktischen und pragmatischen Ebenen. Dabei werden technische Proof-Verfahren und Datenminimalismus durch Selective Disclosure oder Predicated Proofs betrachtet. Es zeigt sich, dass VCDM im Vergleich zu AnonCreds grundlegende Strukturen für die Abbildung von semantischer und pragmatischer Ebene bietet, wobei die semantische Ebene über das Semantic-Web adressiert wird. Diese Informations-Ebenen erweitern die Anforderungen an den Funktions- und Leistungsumfang des Schemas und der Fokus wird in Kapitel 6 auf das VCDM gelegt. In Kapitel 6 wird der Schemaprozess untersucht, wobei bestehende Credential-Type-Definitionen, Ansätze zur Definition von Vertikalen und Overlay-Strukturen durch die vier Mechanismen analysiert werden. Das frühe Entwicklungsstadium des VCDM und der gesamten SSI-Welt wird hervorgehoben, wobei die Verknüpfung von Vokabular und semantischem Web über Kontextdateien als entscheidend identifiziert wird. Die Definition, als komplexester und wichtigster Mechanismus, legt den Grundstein für die Informationsqualität der Credentials. Für diesen Prozessschritt sind Domänenwissen und Kenntnisse im Semantic-Web des Schema-Autors erforderlich, während nachfolgende Mechanismen automatisiert werden können. Es wird angemerkt, dass es noch keine visuellen Editoren für die benutzerfreundliche Erstellung von Schemata gibt, ohne semantische Zuordnung einzubüßen. Die Anforderungen der Persistenz und Portabilität an das Schema werden betrachtet, und organisatorische Aspekte als notwendig für die Implementierung in einem SSI-Ökosystem mit funktionierender Vertrauensinfrastruktur erachtet.

Das VCDM ermöglicht die Erstellung polymorpher Credential-Schemata, insbesondere durch die Entkopplung von Credential-Schema und JSON-LD-Kontextdatei, die den Credential-Type definiert. Diese Trennung erlaubt die Entwicklung verschiedener Schemata für einen Credential-Type, um eine höhere Granularität der Anwendungsfälle abzudecken, wie beim Traceability-Vocabulary. Die Grenzen des VCDM sowie die Notwendigkeit von Overlay-Strukturen müssen noch in realen Anwendungen untersucht werden. Aktuell ist unklar, ob das VCDM und zugehörige Schemata an die Grenzen der Kontextdarstellung stoßen und ob Overlay-Strukturen eine Vertrauensentscheidung positiv beeinflussen können. Angesichts des derzeit schwer zu prognostizierenden Implementierungsaufwands und potentieller Interoperabilitätsprobleme durch eine Erweiterung des VCDM mit Overlay-Strukturen ist es ratsam, sich an die bestehende VCDM-Spezifikation zu halten. Insbesondere durch die Verwendung der JSON-LD-Kontextdatei für das maschinelle Verständnis kann eine hohe Informationsqualität des Credentials erreicht werden. Bei den Overlay-Strukturen ist die Interoperabilität auf das Wissen oder die Implementierung beschränkt, wodurch die Flexibilität, die eine Overlay-Struktur ermöglicht, erst gegeben wäre. Da SOyA für die Definition ihrer Overlay-Struktur

auf das Semantic-Web setzt, ist die Interoperabilität für das VCDM besser als bei OCA, die ihre Overlay-Struktur ohne ontologiebasierte Modelle wie OWL und RDFS definieren. Die Theorie des Semantic-Webs versucht seit Anfang der 2000er Jahre, Semantik in breiter Anwendung zu etablieren. Herausforderungen dabei sind Komplexität und Implementierungsaufwand. Diese Hürden sollten für den Schema-Autor durch passende Werkzeuge und Anreizsysteme minimiert werden, um das Definieren von Credential-Types mit hoher Informationsqualität einer breiten Autorenschaft zugänglich zu machen. Primär sollten unregulierte Anwendungsfälle analysiert werden, da regulierte Anwendungsfälle wie Führerschein und Personalausweis aufgrund ihrer hoheitlichen Bedeutung spezifischen Bedingungen unterliegen. Der Kontextbezug wird durch die staatliche Autorität hergestellt und entsteht eher weniger durch den Informationsaustausch zwischen Holder und Verifier.

Insgesamt diente diese Arbeit einer Bestandsaufnahme des Schemas im Komplex von SSI-Ökosystemen und der Untersuchung des Schemaprozesses. Der Fokus hat sich im Laufe der Erarbeitung des Schemaprozesses deutlich auf den Mechanismus der Definition gerichtet. Eine Erfassung der Thematik um Schema und SSI gestaltet sich jedoch aufgrund des frühen Entwicklungsstadiums der Technologien, der mangelnden realen Anwendungen und der begrenzten Quellenlage konkret zum Titel der Arbeit als schwierig. Allein das Fassen von Grundlagen oder grundlegenden Begriffen wie Information oder Vertrauen erwies sich als schwer, aufgrund der Multidimensionalität dieser Begrifflichkeiten. Die Category-Theory stellt neben den untersuchten Ansätzen der Overlay-Strukturen einen weiteren Ansatz für die semantische Interpretation von Information durch Credentials dar, aufgrund ihrer Umfänglichkeit blieb sie in dieser Arbeit unbehandelt. Der aufgestellte Bewertungsrahmen an Informationsqualitätsmetriken (IQ-Metriken) kann als Evaluationsmethode für weitere Untersuchungen von Schemaprozessen angewandt werden.

Die Steigerung der semantischen Interoperabilität durch die Einigung auf gemeinsame Schemata in verschiedenen Vertrauensdomänen und interdisziplinären Bereichen stellt eine bedeutende Herausforderung dar. Die Erschließung neuer Wertschöpfungspotenziale durch SSI-Ökosysteme kann erreicht werden, wenn sich der Ausstellungs- und der Verifikationsprozess des Credentials in Echtzeit und unabhängig vom Standort durchführen lassen. Eine (Teil-)Automatisierung dieser Prozesse könnte dies leisten. Wird eine Maschine als Akteur eingeführt und versteht die Information, die durch Credentials repräsentiert wird, kann sie auf Basis von vordefinierten Regeln, die von einem menschlichen Entscheidungsträger festgelegt wurden, autonome Prozesse in SSI-Ökosystemen steuern. Ein zentraler Forschungsgegenstand bei der Umsetzung solcher Vorgänge ist die Abbildung der pragmatischen Ebene.

Das Paradox, das die SSI-Prinzipien implizieren, indem sie einerseits den Fokus auf die Wahrung der Privatsphäre und auf Datenminimalismus bei Interaktionen legen, und andererseits die Notwendigkeit betonen, hochsensible personenbezogene Daten semantisch eindeutig interpretierbar zu gestalten, bedarf weiterer Erforschung. Zukünftige Untersuchungen sollten sich darauf konzentrieren, inwieweit persönliche Daten mithilfe von Semantic-Web-Technologien repräsentiert werden sollten und ob Reibungen im System möglicherweise einen Schutz des Nutzers bieten können. Die digitalen Prozesse ermöglichen zum jetzigen Zeitpunkt keine vollständige (Identitäts-)Verifikation, was zu zusätzlichen kostenverursachenden Prozessen wie Postident-Verfahren bei der Eröffnung eines Onlinebankkontos führt. Diese Umstände dienen als Hürde oder Schutz vor einer inflationären Verwendung oder gar einem Abgabezwang persönlicher Daten. Eine extrem hohe Datenqualität, die die Verwendung der Information in verschiedenen Kontexten zulässt, steigert den Wert der Information korrelativ. Die semantische Aufbereitung hochwertiger, eindeutig personenbezogener Daten optimiert die Nutzbar-

keit der Daten, oft in guter Absicht, während gleichzeitig das Risiko des Missbrauchs steigt. Die Diskussionen und Positionen innerhalb der DIF zu *Actors, Objects, and Linked Dat* sind in diesem Zusammenhang von großer Bedeutung (vgl. [Har22; ODo22]).

Ein Faktor für die semantische Genauigkeit von Schemata und deren Attributen ist die effiziente Nutzung bestehender Vokabulare und deren Terminologien. Hierbei steht das Motto „Nicht mehr Vokabular schaffen, sondern die Wiederverwertung bestehender Vokabulare fördern“ im Vordergrund. Die zentrale Aufgabe von Schemata besteht darin, den Umfang und Inhalt der Information von Credentials zu bestimmen, um klar definierte Credential-Typen in SSI-Ökosystemen zu etablieren. Die Definition der Attribute sollte durch den Einsatz von Semantic-Web-Technologien erfolgen, um eine unabhängige, semantische Repräsentation der Attribute vom Schema zu gewährleisten. Dies ermöglicht die Verwendung von Claims und deren Attributen in einem weitgefassteren Kontext, der über die strukturgebundene Begrenzung eines Schemas hinausgeht. Insgesamt zeigt die Untersuchung in der Arbeit, dass die semantische Interpretierbarkeit von Daten in domänenspezifischen Kontexten eine entscheidende Rolle spielt, um SSI effektiv als Baustein für eine Data-Driven-Economy zu nutzen.

Anhang A: Credential-Prozesskette

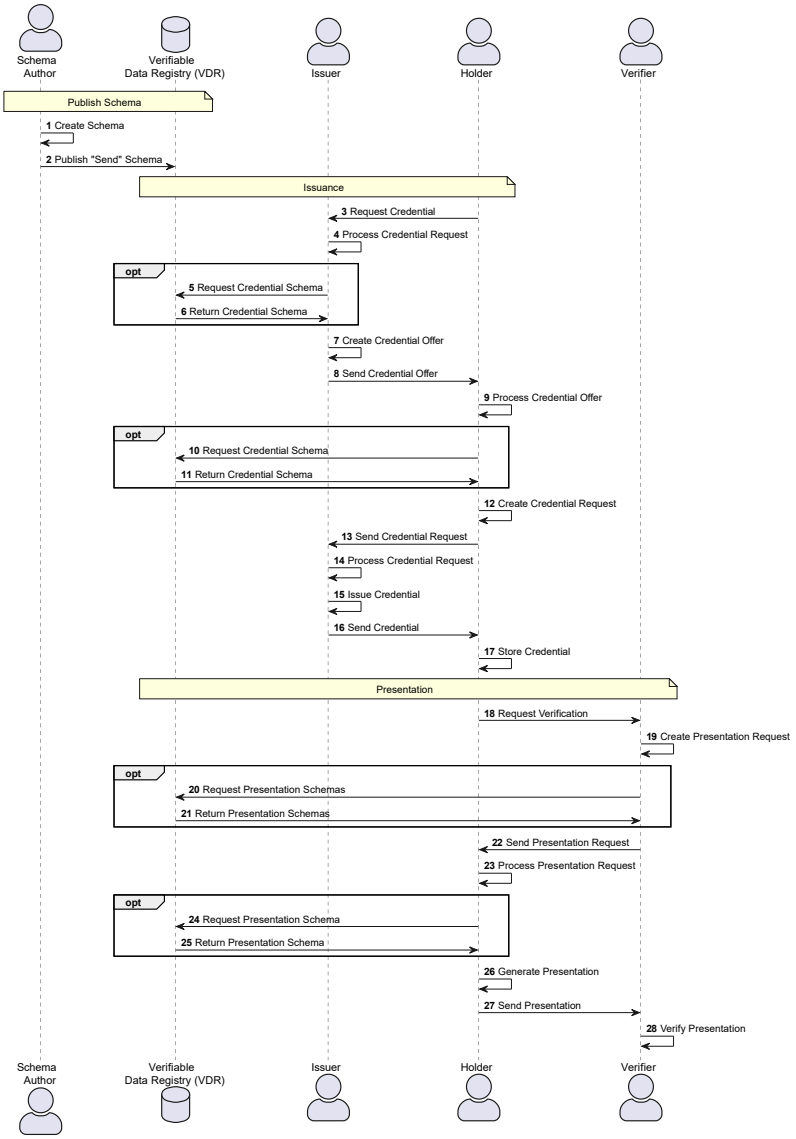

Abbildung A.1: Credential-Prozesskette (in Anlehnung an [Cur+23])

Anhang B: IQs-Dimensionen

Tabelle B.1: IQs-Dimension und Metriken

IQs-Dimension	Abk.	Metrik	Typ	Akteur
Lizenz (Licensing)	L1	Vorhanden in maschinenlesbare Form [Can+23; Zav+15; Bat+15; Wil+16]	QN	Format & Schema & VDR
	L2	Vorhanden in menschlich lesbarer Form [Zav+15]	QN	Format & Schema & VDR
	L3	Angabe, ob der Datensatz unter derselben Lizenz steht wie das Original [Zav+15]	QN	VDR
	L4	Rechtliche Wiederverwendbarkeit [Bat+15; Bat+22]	QN	Schema
Abrufbarkeit (Availability)	A1	Möglichkeit der Dereferenzierung von Ressourcen [Can+23; Zav+15; Bat+15]	QN	VDR
	A2	Bereitstellung eines RDF-Exports [Zav+15]	QN	VDR
	A3	Bereitstellung eines öffentlichen SPARQL-Endpunktes [Zav+15]	QN	VDR
	A4	VDR ist API-fähig [Wil+16]	QN	VDR
	A5	Bereitstellung der Daten durch https-GET-Methode [Bat+22]	QN	VDR
	A6	Metadaten sind zugänglich, auch wenn die Daten nicht mehr verfügbar sind. [Wil+16]	QN	VDR
	A7	Metadaten sind anhand ihrer Kennung über ein standardisiertes Kommunikationsprotokoll abrufbar. [Wil+16]	QN	VDR
Verknüpfung (Linkability)	V1	Verlinkung mehrere Vokabulare zur Beschreibung derselben Ressourcen [Can+23; Bat+22; Iss+21]	QN	Schema
	V2	Links zu externen Datenquellen sind vorhanden und gültig. [Can+23; Zav+15; W3C13; Bat+22; Wil+16; Iss+21]	QN	Schema
Interoperabilität	I1	Wiederverwendung bestehender Vokabulare [Can+23; Zav+15; Bat+15; Beh+14]	QN	Schema
	I2	Veröffentlichung von menschen- und maschinenlesbaren Beschreibungen. [Beh+14]	QN	VDR & Schema

Tabelle B.1: IQs-Dimension und Metriken

IQs-Dimension	Abk.	Metrik	Typ	Akteur
	I3	Wiederverwendung bestehender Begriffe [Can+23; Zav+15; Bat+15]	QN	Schema
	I4	Vermeidung von blank-nodes und RDF-Reification [Can+23]	QN	Schema
	I5	Bereitstellung von mehreren Serialisierungsformaten [Can+23]	QN	VDR
	I6	Verwendung von selbstbeschreibenden Formaten [Zav+15; W3C13]	QN	Format & Schema
Offene Formate	O1	Verwendung eines offenen Formates [Zav+15; W3C13; Wil+16]	QN	Format & Schema
Minimalismus	M1	Datenerhebung ist auf das Minimum der jeweiligen Anwendung reduziert (Angemessener Umfang) [All16]	QL	Author & Nutzer
	M2	Verwendung von Methoden zur selektiven Offenlegung von Daten (Selective-Disclosure) [All16]	QN	Format
	M3	Verwendung von Methoden zur binären Präsentation von Daten (Predicate-Proofs) [All16]	QN	Format
Leistung (Performance)	P1	geringe Latenz [Zav+15]	QN	VDR
	P2	hoher Durchsatz [Zav+15]	QN	VDR
	P3	Skalierbarkeit einer Datenquelle [Zav+15]	QN	VDR
Sicherheit (Security)	S1	Verwendung digitaler Signaturen [Zav+15]	QN	Format & Schema
	S2	Zugangsbeschränkung [NBL22]	QN	VDR
	S3	Kann die Integrität der Daten überprüft werden (Überprüfbarkeit) [NR00; BS14]	QN	VDR & Schema
Syntaktische Genauigkeit (Syntactic Accuracy)	Syn1	Anteil der syntaktisch falschen Triple [Beh+14]	QN	VC & Schema
	Syn2	Anteil von Triples mit unzulässigen Zuweisungen von Datentypen zu Literalen [Beh+14]	QN	VC & Schema
	Syn4	Anteil der Instanzen, die undefinierte Klassen/Eigenschaften verwenden [Beh+14]	QN	VC & Schema
	Syn5	Anteil der Instanzen, die Mitglieder von disjunkten Klassen sind [Beh+14]	QN	VC & Schema

Tabelle B.1: IQs-Dimension und Metriken

IQs-Dimension	Abk.	Metrik	Typ	Akteur
	Syn6	Anteil der Triple mit unsachgemäßer Verwendung von Vokabular [Beh+14]	QN	VC & Schema
Semantische Genauigkeit (Semantic Accuracy)	Sem1	Syntaktische Gültigkeit von RDF-Dokumenten [Can+23]	QN	VC & Schema
	Sem2	Syntaktische Gültigkeit von Literalen [Can+23]	QN	VC & Schema
	Sem3	Syntaktische Gültigkeit von Triples [Can+23]	QN	VC& Schema
	Sem4	Prüfung auf doppelte Entitäten [Can+23]	QN	VC & Schema
	Sem5	keine Ausreißer [Zav+15]	QN	VC & Schema
	Sem6	keine ungenauen Werte [Zav+15]	QN	VC & Schema
	Sem7	keine ungenauen Anmerkungen, Beschriftungen oder Klassifizierungen [Zav+15]	QN	VC & Schema
	Sem8	keine missbräuchliche Verwendung von Eigenschaften [Zav+15]	QN	VC & Schema
	Sem9	Erkennung von gültigen Regeln [Zav+15]	QN	VC & Schema
	Sem10	Anteil der Triple mit fehlenden Objekten [Beh+14]	QN	VC & Schema
	Sem11	Anteil von Triples mit Objekten außerhalb des Bereichs [Beh+14]	QN	VC & Schema
	Sem12	Anteil der Triples mit falsch geschriebenem Datenwert [Beh+14]	QN	VC & Schema
	Sem13	Anteil der Entitäten ohne Entsprechung in der realen Welt [Beh+14]	QN	VC & Schema
Aktualität (timeliness)	T1	Aktualitätshäufigkeit des Schemas [Can+23]	QN	VDR & Schema
	T2	Angabe des Gültigkeitszeitraums des Schemas [Can+23]	QN	VDR& Schema
	T3	Angabe des Änderungsdatums des Schemas [Can+23]	QN	VDR & Schema
	T4	Aktualität der Datensätze basierend auf Währung und Volatilität [Zav+15]	QN	VDR & Schema

Tabelle B.1: IQs-Dimension und Metriken

IQs-Dimension	Abk.	Metrik	Typ	Akteur
	T5	Aktualität von Datensätzen basierend auf ihrer Datenquelle [Zav+15]	QN	VDR & Schema

Literaturverzeichnis

[1Ed] 1EdTech. *CLR FAQ | IMS Global Learning Consortium*. URL: http://www.imsglobal.org/clr/faq#clrIntro (besucht am 11. 04. 2023).

[1Ed22] 1EdTech. *Badge Count 2022 - Badge Count 2022*. 2022. URL: https://content.1edtech.org/badge-count-2022/findings (besucht am 21. 02. 2023).

[1Ed23] 1EdTech Consortium Inc. *Open Badges History | IMS Open Badges*. 2023. URL: https://openbadges.org/about/history (besucht am 16. 02. 2023).

[Aff22] Affinidi. *Create Custom Verifiable Credentials with Affinidi's Schema Manager*. Medium. 15. März 2022. URL: https://academy.affinidi.com/create-custom-verifiable-credentials-with-affinidis-schema-manager-86149b2d49d6 (besucht am 19. 09. 2022).

[All16] Christopher Allen. *The Path to Self-Sovereign Identity*. Mit einem Komm. von Shannon Appelcline u. a. 2016. URL: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html (besucht am 12. 01. 2023).

[And22] Andy Miller. *1EdTech Verifiable Credential Refresh Service | IMS Global Learning Consortium*. 1EdTech, 2022. URL: https://www.imsglobal.org/spec/vccr/v1p0 (besucht am 03. 04. 2023).

[AS09] Alistair Miles und Sean Bechhofer. *SKOS Simple Knowledge Organization System Reference*. 2009. URL: https://www.w3.org/TR/skos-reference/#L1629 (besucht am 25. 04. 2023).

[Bat+15] Carlo Batini u. a. „From Data Quality to Big Data Quality". In: *Journal of Database Management* 26 (1. Jan. 2015), S. 60–82. DOI: 10.4018/JDM.2015010103.

[Bat+22] Dominique Batista u. a. „Machine Actionable Metadata Models". In: *Scientific Data* 9.1 (30. Sep. 2022), S. 592. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01707-6. URL: https://www.nature.com/articles/s41597-022-01707-6 (besucht am 12. 12. 2022).

[BCO22] Daniel Bachenheimer, Wejing Chu und Darrell O' Donnel. *Trust over IP (ToIP) Technology Architecture Specification*. Hrsg. von Trust Over IP Foundation. 2022. URL: https://trustoverip.org/wp-content/uploads/ToIP-Technical-Architecture-Specification-V1.0-PR1-2022-11-14.pdf (besucht am 31. 01. 2023).

[Beh+14] Behshid Behkamal u. a. „A Metrics-Driven Approach for Quality Assessment of Linked Open Data". In: *Journal of theoretical and applied electronic commerce research* 9 (1. Mai 2014), S. 64–79. DOI: 10.4067/S0718-18762014000200006.

[Bel21] Doug Belshaw. *Good Things Happen Slowly, Bad Things Happen Fast*. Medium. 30. Nov. 2021. URL: https://blog.weareopen.coop/good-things-happen-slowly-bad-things-happen-fast-2fd894cbd4df (besucht am 16. 03. 2023).

[BH01] Tim Berners-Lee und James Hendler. „The Semantic Web". In: *Nature* 410 (1. Mai 2001), S. 1023–4. DOI: 10.1038/35074206.

[BN20] Andreas Blumauer und Helmut Nagy. *The Knowledge Graph Cookbook: Recipes That Work*. 1st edition. Wien: edition mono/monochrom, 2020. 256 S. ISBN: 978-3-902796-70-7.

[Bra+22] John Bradley u. a. *Iana - JSON Web Token (JWT)*. 2022. URL: https://www.iana.org/assignments/jwt/jwt.xhtml (besucht am 01. 12. 2022).

[BS14] Maria Batista und Ana Carolina Salgado. „Information Quality Measurement in Data Integration Schemas." In: 2014, S. 61–72.

[Can+23] Gustavo Candela u. a. „Shape Expression Approach for Assessing the Quality of Linked Open Data in Libraries". In: *Semantic Web* 14.2 (1. Jan. 2023), S. 159–179. ISSN: 1570-0844. DOI: 10.3233/SW-210441. URL: https://content.iospress.com/articles/semantic-web/sw210441.

[Cap+22] Steve Capell u. a. *White Paper eDATA Verifiable Credentials for Cross White Paper eDATA Verifiable Credentials for Cross Border Trade*. Hrsg. von UN E CE – UN / CE F A CT. 2022. URL: https://unece.org/sites/default/files/2022-07/WhitePaper_VerifiableCredentials-CBT.pdf (besucht am 08. 12. 2022).

[CFG23] Christoph Fabianek, Fajar J. Ekaputra und Gabriel Unterholzer. *Semantic Overlay Architecture*. 2023. URL: https://ownyourdata.github.io/soya/ (besucht am 19. 01. 2023).

[Cha+19] David Chadwick u. a. *Verifiable Credentials Implementation Guidelines 1.0*. 2019. URL: https://w3c.github.io/vc-imp-guide/ (besucht am 10. 11. 2022).

[che22] cheqd press. *Cheqd Announces Multiple Partnerships with Market-Leading Self-Sovereign Identity Vendors Ahead of Mainnet Launch*. 11. Dez. 2022. URL: https://cheqd.io/blog/cheqd-announces-multiple-partnerships-with-market-leading-self-sovereign-identity-vendors-ahead-of-mainnet-launch (besucht am 11. 12. 2022).

[CS22] Gabe Cohen und Orie Steele. *Verifiable Credentials JSON Schema Specification*. 2022. URL: https://w3c-ccg.github.io/vc-json-schemas/ (besucht am 23. 11. 2022).

[Cur+23] Stephen Curran u. a. *AnonCreds Specification*. 2023. URL: https://hyperledger.github.io/anoncreds-spec/ (besucht am 10. 01. 2023).

[Dan+22a] Daniel Buchner u. a. *DIF Credential Manifest*. Spezification. Decentralized Identity Foundation, 2022. URL: https://identity.foundation/credential-manifest/ (besucht am 06. 02. 2023).

[Dan+22b] Daniel Buchner u. a. *DIF Wallet Rendering*. Spezification. Decentralized Identity Foundation, 2022. URL: https://identity.foundation/wallet-rendering/ (besucht am 06. 02. 2023).

[Dan+23] Daniel Buchner u. a. *DIF Presentation Exchange*. 2023. URL: https://identity.foundation/presentation-exchange/ (besucht am 14. 02. 2023).

[Dar+21] Darrel Miller u. a. *OpenAPI Specification v3.1.0 | Introduction, Definitions, & More*. 2021. URL: https://spec.openapis.org/oas/latest.html#openapi-specification (besucht am 17. 04. 2023).

[Dav+12] David Peterson u. a. *W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes*. 2012. URL: https://www.w3.org/TR/xmlschema11-2/#dateTime (besucht am 25. 03. 2023).

[Dav+20] Dave Longley u. a. *JSON-LD 1.1 Framing*. 2020. URL: https://w3c.github.io/json-ld-framing/ (besucht am 20. 01. 2022).

[Dec22a] Decentralized Identity Foundation. „DIF Interop Survey Results“. 2022. URL: https://docs.google.com/presentation/d/1lRxa49ZJjrSiWx-4ldLAwWvin5wukenEIXjK5Rj9l24 (besucht am 02. 02. 2023).

[Dec22b] Decentralized Identity Foundation. *GitHub - Decentralized-Identity/Schema-Directory: A Work Item of the Claims and Credentials WG at DIF*. 27. Okt. 2022. URL: https://github.com/decentralized-identity/schema-directory (besucht am 15. 02. 2023).

[DJM22] Daniel Buchner, Jack Couch und Martin Riedel. *Schema Directory*. 2022. URL: https://identity.foundation/schema-directory/ (besucht am 22. 02. 2023).

[DM22a] Dave Longley und Manu Sporny. *Data Integrity 1.0*. 2022. URL: https://www.w3.org/community/reports/credentials/CG-FINAL-data-integrity-20220722/ (besucht am 03. 02. 2023).

[DM22b] Dave Longley und Manu Sporny. *RDF Dataset Canonicalization*. 2022. URL: https://w3c-ccg.github.io/rdf-dataset-canonicalization/spec/ (besucht am 20. 01. 2022).

[doc22a] dock.io. *Dock Blockchain*. 2022. URL: https://docs.dock.io/intro-to-dock/dock-blockchain (besucht am 24. 04. 2023).

[doc22b] dock.io. *Dock SDK: Schemas*. 2022. URL: https://docknetwork.github.io/sdk/tutorials/concepts_blobs_schemas.html (besucht am 22. 09. 2022).

[doc23] dock.io. *Custom Credential Template Builder Guide*. 2023. URL: https://www.dock.io/guides/custom-credential-template-builder?ref=blog.dock.io (besucht am 22. 04. 2023).

[DRB14] Dan Brickley, R.V. Guha und Brian McBride. *RDF Schema 1.1*. 2014. URL: https://www.w3.org/TR/rdf-schema/ (besucht am 22. 03. 2023).

[DS05] Martin J. Dürst und Michel Suignard. *Internationalized Resource Identifiers (IRIs)*. Request for Comments RFC 3987. Internet Engineering Task Force, Jan. 2005. 46 S. DOI: 10.17487/RFC3987. URL: https://datatracker.ietf.org/doc/rfc3987 (besucht am 24. 03. 2023).

[Duda] Duden. *Attribut Rechtschreibung, Bedeutung, Definition, Herkunft | Duden*. URL: https://www.duden.de/rechtschreibung/Attribut (besucht am 02. 03. 2023).

[Dudb] Duden. *Vertrauen Rechtschreibung, Bedeutung, Definition, Herkunft | Duden*. URL: https://www.duden.de/rechtschreibung/Vertrauen (besucht am 19. 03. 2023).

[ECM17] ECMA. *ECMA-404 The JSON Data Interchange Syntax*. 2017. URL: https://www.ecma-international.org/publications-and-standards/standards/ecma-404/ (besucht am 28. 11. 2022).

[Edw21] Fraser Edwards. *Why Cheqd Has Joined the Cosmos*. cheqd. 19. Aug. 2021. URL: https://cheqd.io/blog/why-cheqd-has-joined-the-cosmos/ (besucht am 24. 04. 2023).

[EHR15] Börteçin Ege, Bernhard Humm und Anatol Reibold, Hrsg. *Corporate Semantic Web: Wie semantische Anwendungen in Unternehmen Nutzen stiften*. X.media.press. Berlin, Heidelberg: Springer, 2015. ISBN: 978-3-642-54885-7. DOI: 10.1007/978-3-642-54886-4. URL: https://link.springer.com/10.1007/978-3-642-54886-4.

[Eur] European Blockchain Services Infrastructure. *Trusted Schemas Registry API v2 | EBSI Developers Hub*. URL: https://api-conformance.ebsi.eu/docs/apis/trusted-schemas-registry/v2 (besucht am 21. 04. 2023).

[Gre+20a] Gregg Kellogg u. a. *Json-Ld-Rc/Context.Jsonld at Main · W3c/Json-Ld-Rc · GitHub*. 2020. URL: https://github.com/w3c/json-ld-rc/blob/main/context.jsonld (besucht am 15. 02. 2023).

[Gre+20b] Gregg Kellogg u. a. *RDFa Core Initial Context, JSON-LD Recommended Context*. 2020. URL: https://w3c.github.io/json-ld-rc/context.jsonld (besucht am 15. 02. 2023).

[Ham+19] Bjoern Hamel u. a. *Workday-Credential-Schemas.Pdf*. 2019. URL: https://lists.w3.org/Archives/Public/public-credentials/2019Oct/att-0008/workday-credential-schemas.pdf (besucht am 03. 12. 2022).

[Har22] Daniel Hardman. *Actors, Objects, and Linked Data*. Medium. 18. Nov. 2022. URL: https://daniel-hardman.medium.com/actors-objects-and-linked-data-7f60701af9bd (besucht am 28. 11. 2022).

[Hei96] Heinz Hübner. *Unternehmung Und Unvollkommene Information*. 1996. ISBN: 978-3-322-98938-3. URL: https://link.springer.com/book/10.1007/978-3-322-98938-3.

[HGM21] Knut Hildebrand, Marcus Gebauer und Michael Mielke, Hrsg. *Daten- und Informationsqualität: die Grundlage der Digitalisierung*. 5., erweiterte und aktualisierte Auflage. Wiesbaden [Heidelberg]: Springer Vieweg, 2021. 487 S. ISBN: 978-3-658-30991-6.

[Hoi54] Harry Hoijer, Hrsg. *Language in Culture: Conference on the Interrelations of Language and Other Aspects of Culture*. Comparative Studies of Cultures and Civilizations. Chicago: University of Chicago Press, 1954. 286 S.

[Hop20] Thomas Hoppe. *Semantische Suche: Grundlagen Und Methoden Semantischer Suche von Textdokumenten*. Springer eBook Collection. Wiesbaden: Springer Vieweg, 2020. ISBN: 978-3-658-30427-0. DOI: 10.1007/978-3-658-30427-0.

[Hum23a] Human Colossus Foundation. *Introduction | Overlays Capture Architecture*. 2023. URL: https://oca.colossi.network/guide/introduction.html (besucht am 21. 01. 2023).

[Hum23b] Human Colossus Foundation. *OCA Browser | Overlays Capture Architecture*. 2023. URL: https://oca.colossi.network/ecosystem/oca-browser.html (besucht am 17. 04. 2023).

[Hum23c] Human Colossus Foundation. *OCA Parser | Overlays Capture Architecture*. 2023. URL: https://oca.colossi.network/ecosystem/oca-parser.html (besucht am 17. 04. 2023).

[Hum23d] Human Colossus Foundation. *OCA Repository | Overlays Capture Architecture*. 2023. URL: https://oca.colossi.network/ecosystem/oca-repository.html (besucht am 17. 04. 2023).

[Hum23e] Human Colossus Foundation. *OCA Transformer | Overlays Capture Architecture*. 2023. URL: https://oca.colossi.network/ecosystem/oca-transformer.html (besucht am 17. 04. 2023).

[Hum23f] Human Colossus Foundation. *OCA Validator | Overlays Capture Architecture*. 2023. URL: https://oca.colossi.network/ecosystem/oca-validator.html (besucht am 17. 04. 2023).

[Hyp20] Hyperledger Indy. *Indy SDK for Node.Js — Hyperledger Indy SDK Documentation*. 2020. URL: https://hyperledger-indy.readthedocs.io/projects/sdk/en/latest/wrappers/nodejs/README.html?highlight=schema#getschema-poolhandle-wh-submitterdid-id-options-gt-schema (besucht am 15. 01. 2023).

[IDu] IDunion. *Projekt – IDunion*. URL: https://idunion.org/projekt/ (besucht am 12. 01. 2023).

[IET16] IETF. *Javascript Object Signing and Encryption (Jose)*. 2016. URL: https://datatracker.ietf.org/wg/jose/charter/ (besucht am 03. 02. 2023).

[IIP12] Ivan Herman, Ian Horrocks und Peter F. Patel-Schneider. *OWL 2 Web Ontology Language Document Overview (Second Edition)*. 2012. URL: https://www.w3.org/TR/owl2-overview/ (besucht am 22. 03. 2023).

[Inta] Internet Assigned Numbers Authority. *Internet Assigned Numbers Authority*. URL: https://www.iana.org/ (besucht am 25. 03. 2023).

[Intb] InterPlanetary File System. *Best Practices | IPFS Docs*. URL: https://docs.ipfs.tech/how-to/gateway-best-practices/#selecting-a-gateway-type-to-use (besucht am 22. 04. 2023).

[Intc] InterPlanetary File System. *Exchange Files between Nodes | IPFS Docs*. URL: https://docs.ipfs.tech/how-to/exchange-files-between-nodes/ (besucht am 22. 04. 2023).

[ISO16] ISO. *ISO 8000-130:2016, Data Quality — Part 130: Master Data: Exchange of Characteristic Data: Accuracy*. Geneva, Switzerland, 2016. URL: https://www.iso.org/obp/ui/#iso:std:iso:8000:-130:ed-1:v1:en (besucht am 21. 01. 2023).

[Iss+21] Subhi Issa u. a. „Knowledge Graph Completeness: A Systematic Literature Review“. In: *IEEE Access* 9 (2021), S. 31322–31339. DOI: 10.1109/access.2021.3056622.

[JBS15] Michael Jones, John Bradley und Nat Sakimura. *JSON Web Token (JWT)*. Request for Comments RFC 7519. Internet Engineering Task Force, Mai 2015. 30 S. DOI: 10.17487/RFC7519. URL: https://datatracker.ietf.org/doc/rfc7519 (besucht am 25. 03. 2023).

[KB19] Ken Ebert und Brent Zundel. *RFC 0250: Rich Schema Objects*. Aries-RFCS: Hyperledger, 2019. URL: https://github.com/hyperledger/aries-rfcs/blob/482d6e642d3d2377f4d0276ec6bf828afaa6f928/concepts/0250-rich-schemas/README.md (besucht am 01. 03. 2023).

[Kee13] C. Maria Keet. „Open World Assumption“. In: *Encyclopedia of Systems Biology*. Springer, New York, NY, 2013, S. 1567. DOI: 10.1007/978-1-4419-9863-7âĂŮ734. URL: https://link.springer.com/referenceworkentry/10.1007/978-1-4419-9863-7_734.

[Kel23] Gregg Kellogg. *JSON-LD Best Practices*. 2023. URL: https://w3c.github.io/json-ld-bp/ (besucht am 10. 02. 2023).

[Klo11] Michael Klotz. *Konzeption Des Persönlichen Informationsmanagements*. 26. Juni 2011. URL: https://www.researchgate.net/publication/254460443_Konzeption_des_personlichen_Informationsmanagements.

[Kno19] Paul Knowles. *Overlays Data Capture Architecture (ODCA): Providing a Standardized Global Solution for Data...* Medium. 8. Mai 2019. URL: https://medium.com/@paul.knowles_52509/overlays-data-capture-architecture-odca-providing-a-standardized-global-solution-for-data-caeb1679137a (besucht am 18. 02. 2022).

[Krc15] Helmut Krcmar. *Informationsmanagement*. Berlin, Heidelberg: Springer, 2015. ISBN: 978-3-662-45862-4. DOI: 10.1007/978-3-662-45863-1. URL: https://link.springer.com/10.1007/978-3-662-45863-1 (besucht am 19. 02. 2023).

[Kud+22] Andre Kudra u. a. *Credential Comparison Matrix*. 2022. URL: https://docs.google.com/spreadsheets/d/1Z4cYfjbbE-rABcfC-xab8miocKLomivYMUFibOh9BVo/edit?usp=embed_facebook (besucht am 28. 11. 2022).

[KW20] Paul Knowles und John Wunderlich. *Blinding Identity Taxonomy*. 2020. URL: https://docs.kantarainitiative.org/Blinding-Identity-Taxonomy-Report-Version-1.0.pdf (besucht am 22. 02. 2022).

[Lee+13] Lee Feigenbaum u. a. *SPARQL 1.1 Protocol*. W3C, 2013. URL: https://www.w3.org/TR/sparql11-protocol/ (besucht am 17. 04. 2023).

[Lon+20] Dave Longley u. a. *JSON-LD 1.1 Processing Algorithms and API*. 2020. URL: https://w3c.github.io/json-ld-api/ (besucht am 18. 12. 2022).

[LS20] Dave Longley und Manu Sporny. *Citizenship Vocabulary v0.3*. 0.3. 2020. URL: https://w3c-ccg.github.io/citizenship-vocab/ (besucht am 18. 10. 2022).

[LSP21] Tobias Looker, Orie Steele und Michael Prorock. *Vaccination Certificate Vocabulary v0.1*. 2021. URL: https://w3c-ccg.github.io/vaccination-vocab/ (besucht am 18. 10. 2022).

[MDO20] Manu Sporny, Drummond Reed und Orie Steele. *Linked Data Cryptographic Suite Registry*. 29. Dez. 2020. URL: https://w3c-ccg.github.io/ld-cryptosuite-registry/ (besucht am 29. 12. 2022).

[MDS95] Roger C. Mayer, James H. Davis und F. David Schoorman. „An Integrative Model of Organizational Trust“. In: *The Academy of Management Review* 20.3 (1995), S. 709–734. ISSN: 0363-7425. DOI: 10.2307/258792. JSTOR: 258792. URL: https://www.jstor.org/stable/258792 (besucht am 19. 03. 2023).

[Men22] Christoph Menzer. „Interoperabilität Zwischen DID-Methoden, Wallets, Agents Und Verifiable-Credentials“. HS Mittweida, 2022.

[Mi19] Michael Boyd und infominer33. *Hyperledger Indy — Hyperledger Indy 1.0 Documentation*. 2019. URL: https://hyperledger-indy.readthedocs.io/en/latest/ (besucht am 12. 01. 2023).

[Nat+23a] Nate Otto u. a. *Comprehensive Learner Record Standard Version 2.0 | IMS Global Learning Consortium*. 2023. URL: https://www.imsglobal.org/spec/clr/v2p0 (besucht am 22. 03. 2023).

[Nat+23b] Nate Otto u. a. *Open Badges Specification | IMS Global Learning Consortium*. Spezification. 1EdTech, 2023. URL: https://imsglobal.org/spec/ob/v3p0/ (besucht am 14. 02. 2023).

[NBL22] Aparna Nayak, Bojan Božić und Luca Longo. „Linked Data Quality Assessment: A Survey“. In: *Web Services – ICWS 2021*. Hrsg. von Chengzhong Xu u. a. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, S. 63–76. ISBN: 978-3-030-96140-4. DOI: 10.1007/978-3-030-96140-4_5.

[NR00] Felix Naumann und Claudia Rolker. „Assessment Methods for Information Quality Criteria“. In: *IQ* (1. Okt. 2000), S. 148–162. DOI: 10.18452/2441.

[O D22] Darrell O' Donnel. *Trust Registry Protocol v2 - Loose Capture*. 2022. URL: https://wiki.trustoverip.org/display/HOME/Trust+Registry+Protocol+v2+-+Loose+Capture (besucht am 12. 01. 2023).

[ODo22] Darrell O'Donnell. *Darrell O'Donnell on LinkedIn: Actors, Objects, and Linked Data*. 2022. URL: https://www.linkedin.com/posts/darrellodonnell_actors-objects-and-linked-data-activity-7001926140764307456-mxjZ (besucht am 28. 11. 2022).

[OM22] Orie Steele und Michael Jones. *JSON Web Signatures for Data Integrity Proofs*. 2022. URL: https://w3c.github.io/vc-jws-2020/ (besucht am 25. 03. 2023).

[Ont23] Ontology Engineering Group (OEG). *Linked Open Vocabularies (LOV)*. 2023. URL: https://lov.linkeddata.es/dataset/lov (besucht am 15. 03. 2023).

[OR22] Darrell O' Donnel und Drummond Reed. *ToIP Trust Registry Protocol V1 Specification*. Unter Mitarb. von Antti Kettunen, Daniel Bachenheimer und Eric Drury. 16. Nov. 2022. URL: https://github.com/trustoverip/tswg-trust-registry-tf/blob/fa6951c77a973944170c0f2706732bdc2ea8872a/docs/ToIP%20Trust%20Registry%20V1%20Specification.md (besucht am 12. 01. 2023).

[Ori+23] Orie Steele u. a. *Traceability Vocabulary v0.1*. 2023. URL: https://w3c-ccg.github.io/traceability-vocab/ (besucht am 26. 02. 2023).

[Own20] OwnYourData. *Semcon/Sc-Sparql - Docker Image | Docker Hub*. 2020. URL: https://hub.docker.com/r/semcon/sc-sparql/ (besucht am 17. 04. 2023).

[Own22a] OwnYourData. *GitHub - Sem-Con/Sc-Base: Base Container for the Semantic Container Infrastructure.* 2022. URL: https://github.com/sem-con/sc-base (besucht am 17. 04. 2023).

[Own22b] OwnYourData. *Semantic Overlay Architecture*. 2022. URL: https://www.ownyourdata.eu/en/semantic-overlay-architecture/ (besucht am 17. 02. 2023).

[Pow23] Ross Power. *Our Product Vision for 2023 at Cheqd*. Medium. 6. Feb. 2023. URL: https://blog.cheqd.io/our-product-vision-for-2023-at-cheqd-70167722a92f (besucht am 13. 02. 2023).

[PR21] Alexander Preukschat und Drummond Reed. *Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials*. Shelter Island, NY: Manning Publications Co., 2021. ISBN: 978-1-61729-659-8.

[Pre22] Tom Preston-Werner. *Semantic Versioning 2.0.0*. 15. Sep. 2022. URL: https://semver.org/ (besucht am 15. 01. 2023).

[Pro] Protocol Labs. *IPFS Powers the Distributed Web*. URL: https://ipfs.tech/ (besucht am 29. 03. 2023).

[RDM14] Richard Cyganiak, David Wood und Markus Lanthaler. *RDF 1.1 Concepts and Abstract Syntax*. 2014. URL: https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/ (besucht am 07. 04. 2023).

[Roh+21] Jan P. Rohweder u. a. „Informationsqualität – Definitionen, Dimensionen und Begriffe“. In: *Daten- und Informationsqualität: Die Grundlage der Digitalisierung*. Hrsg. von Knut Hildebrand, Marcus Gebauer und Michael Mielke. Wiesbaden: Springer Fachmedien, 2021, S. 23–43. ISBN: 978-3-658-30991-6. DOI: 10.1007/978-3-658-30991-6_2.

[RP21] Drummond Reed und Scott Perry. *ToIP Governance Architecture Specification: Version 1.0*. Hrsg. von Trust Over IP Foundation. 2021. URL: https://trustoverip.org/wp-content/uploads/ToIP-Governance-Architecture-Specification-V1.0-2021-12-21.pdf (besucht am 10. 01. 2023).

[RS21a] Drummond Reed und Victor Syntez. *Design Principles for the Trust over IP Stack: Version 1.0*. Hrsg. von Trust Over IP Foundation. 2021. URL: https://trustoverip.org/wp-content/uploads/Design-Principles-for-the-ToIP-Stack-V1.0-2022-11-17.pdf.

[RS21b] Drummond Reed und Victor Syntez. *Introduction to Trust over IP: Version 2.0*. Unter Mitarb. von Trust Over IP Foundation. 2021. URL: https://trustoverip.org/wp-content/uploads/Introduction-to-ToIP-V2.0-2021-11-17.pdf (besucht am 25. 01. 2023).

[Sak+14] Nat Sakimura u. a. *Final: OpenID Connect Core 1.0 Incorporating Errata Set 1*. 2014. URL: https://openid.net/specs/openid-connect-core-1_0.html (besucht am 01. 12. 2022).

[Sco23] Scott S. Perry. *How Do Humans Trust?* Trust Over IP. 15. März 2023. URL: https://trustoverip.org/blog/2023/03/15/how-do-humans-trust/ (besucht am 21. 03. 2023).

[SJ23] Orie Steele und Michael B. Jones. *Securing Verifiable Credentials Using JSON Web Tokens*. 2023. URL: https://w3c.github.io/vc-jwt/ (besucht am 13. 01. 2022).

[SLC23] Manu Sporny, Dave Longley und David Chadwick. *Verifiable Credentials Data Model v2.0*. Unter Mitarb. von Orie Steele, Gabe Cohen und Oliver Terbu. 2023. URL: https://w3c.github.io/vc-data-model/ (besucht am 25. 02. 2023).

[Smi22] Samuel M. Smith. *Self-Addressing IDentifier (SAID)*. Internet Draft draft-ssmith-said-02. Internet Engineering Task Force, 31. Mai 2022. 11 S. URL: https://datatracker.ietf.org/doc/draft-ssmith-said (besucht am 13. 04. 2023).

[Sov] Sovrin. *Write to the Sovrin Public Ledger*. Sovrin Foundation. URL: https://sovrin.org/issue-credentials/ (besucht am 12. 01. 2023).

[Sov20] Sovrin. *Sovrin Foundation Announces New Task Forces Focused on Token and Public Write Access*. Sovrin Foundation. 29. Juli 2020. URL: https://sovrin.org/sovrin-foundation-announces-new-task-forces-focused-on-token-and-public-write-access/ (besucht am 12. 01. 2023).

[Spo+22] Manu Sporny u. a. *Decentralized Identifiers (DIDs) v1.0*. 2022. URL: https://www.w3.org/TR/did-core/ (besucht am 19. 12. 2022).

[SR21] Manu Sporny und Leonard Rosenthol. *Cryptographic Hyperlinks*. Internet Draft draft-sporny-hashlink-07. Internet Engineering Task Force, 2. Mai 2021. 11 S. URL: https://datatracker.ietf.org/doc/draft-sporny-hashlink (besucht am 29. 03. 2023).

[SS09] Steffen Staab und Rudi Studer, Hrsg. *Handbook on Ontologies*. Berlin, Heidelberg: Springer, 2009. ISBN: 978-3-540-92673-3. DOI: 10.1007/978-3-540-92673-3. URL: http://link.springer.com/10.1007/978-3-540-92673-3 (besucht am 22. 03. 2023).

[SSN22] Abylay Satybaldy, Anushka Subedi und Mariusz Nowostawski. „A Framework for Online Document Verification Using Self-Sovereign Identity Technology“. In: *Sensors* 22.21 (21 Jan. 2022), S. 8408. ISSN: 1424-8220. DOI: 10.3390/s22218408. URL: https://www.mdpi.com/1424-8220/22/21/8408 (besucht am 29. 12. 2022).

[ST18] SSI Meetup und Tyler Ruff, director. *Verifiable Credentials 101 for SSI with Tyler Ruff - Decentralized Digital Identity*. 18. Sep. 2018. URL: https://www.youtube.com/watch?v=6O_iJnhIh5o (besucht am 29. 03. 2023).

[Ste22] Orie Steele. *Verifiable Credential Schemas with Open API Specification*. 2022. URL: https://transmute-industries.github.io/vc-credential-schema-open-api-specification/#dfn-openapispecificationvalidator2022 (besucht am 08. 12. 2022).

[Str+21] Dr Jens Strüker u. a. „Grundlagen, Anwendungen und Potenziale portabler digitaler Identitäten“. In: (2021), S. 52.

[Tim+13] Timothy Lebo u. a. *PROV-O: The PROV Ontology*. W3C, 2013. URL: https://www.w3.org/TR/prov-o/ (besucht am 08. 02. 2023).

[TO23] Tobias Looker und Orie Steele. *BBS Cryptosuite V2023*. Specification. 2023. URL: https://w3c-ccg.github.io/vc-di-bbs/ (besucht am 25. 03. 2023).

[Tru21] Trust Over IP Foundation. *Good Health Pass Interoperability Blueprint*. 2021. URL: https://goodhealthpass.org//blueprint (besucht am 12. 06. 2022).

[Ver19] Verein zur Förderung der selbstständigen Nutzung von Daten. *White Paper Semantic Container*. 2019. URL: https://www.ownyourdata.eu/wp-content/uploads/2019/06/WhitePaper_Jun19.pdf (besucht am 03. 08. 2022).

[W3C10] W3C. *What Is Provenance - XG Provenance Wiki*. 2010. URL: https://www.w3.org/2005/Incubator/prov/wiki/What_Is_Provenance (besucht am 21. 04. 2023).

[W3C13] W3C. *5 Star Linked Data - Government Linked Data (GLD) Working Group Wiki*. 2013. URL: https://www.w3.org/2011/gld/wiki/5_Star_Linked_Data.

[W3C15] W3C. *Semantic Web - W3C*. 2015. URL: https://www.w3.org/standards/semanticweb/ (besucht am 21. 03. 2023).

[W3C19] W3C. *Ontologies*. 19. Juni 2019. URL: https://www.w3.org/standards/semanticweb/ontology.html (besucht am 04. 01. 2023).

[WC66] Benjamin Lee Whorf und Stuart Chase. *Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf*. Hrsg. von John B. Carroll. M.I.T. Paperback Series. Cambridge: M.I.T. Press, 1966. 278 S.

[Wil+16] Michael Wilkinson u. a. „The FAIR Guiding Principles for Scientific Data Management and Stewardship“. In: *Scientific Data* 3.1 (15. März 2016), S. 160018–160018. DOI: 10.1038/sdata.2016.18. pmid: 26978244.

[Win23] Phillip J. Windley. *Learning Digital Identity: Design, Deploy, and Manage Identity Architectures*. First edition. Sebastopol, CA: O'Reilly Media, Inc., 2023. ISBN: 978-1-09-811766-5.

[WK13] Kristin Weber und Michael Klotz. „Informations- Und Datenmanagement“. In: 2. Juli 2013, S. 553–610. ISBN: 978-3-446-43557-5. DOI: 10.3139/9783446436220.012.

[WK20] Kristin Weber und Christiana Klingenberg. *Data Governance: Der Leitfaden für die Praxis*. 1. Aufl. Hanser Fachbuchverlag, 2020. 343 S. ISBN: 978-3-446-46674-6.

[WS22] Andrew Whitehead und Stephen Curran. *AnonCreds to W3C Format Verifiable Credential and Presentation Converter*. 2022. URL: https://github.com/andrewwhitehead/anoncreds-w3c-mapping (besucht am 01. 03. 2023).

[WS96] Richard Y. Wang und Diane M. Strong. „Beyond Accuracy: What Data Quality Means to Data Consumers“. In: *Journal of Management Information Systems* 12.4 (1. März 1996), S. 5–33. ISSN: 0742-1222. DOI: 10.1080/07421222.1996.11518099. URL: https://doi.org/10.1080/07421222.1996.11518099 (besucht am 30. 01. 2023).

[Yil+22] Hakan Yildiz u. a. „A Tutorial on the Interoperability of Self-sovereign Identities“. In: (9. Aug. 2022). DOI: 10.36227/techrxiv.20430825.v1. URL: https://www.techrxiv.org/articles/preprint/A_Tutorial_on_the_Interoperability_of_Self-sovereign_Identities/20430825/1. preprint.

[You21] Kaliya Young. *Verifiable-Credentials-Flavors-Explained-Infographic.Pdf*. 2021. URL: https://www.lfph.io/wp-content/uploads/2021/04/Verifiable-Credentials-Flavors-Explained-Infographic.pdf (besucht am 01. 12. 2021).

[Zav+15] Amrapali Zaveri u. a. „Quality Assessment for Linked Data: A Survey: A Systematic Literature Review and Conceptual Framework“. In: *Semantic Web* 7.1 (17. März 2015). Hrsg. von Pascal Hitzler, S. 63–93. ISSN: 22104968, 15700844. DOI: 10.3233/SW-150175. URL: https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/SW-150175.

[ZJL21] Lina Zhang, Dongwon Jeong und Sukhoon Lee. „Data Quality Management in the Internet of Things“. In: *Sensors* 21.17 (2021), S. 5834. DOI: 10.3390/s21175834.

[Zun21] Brent Zundel. *Why the Verifiable Credentials Community Should Converge on BBS+*. Evernym. 24. März 2021. URL: https://www.evernym.com/blog/bbs-verifiable-credentials/ (besucht am 01. 03. 2023).

Eidesstattliche Erklärung

Hiermit versichere ich – Timo Burkhardt – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Mittweida, 25. April 2023 — Timo Burkhardt

Ort, Datum — Timo Burkhardt