

# OParl Schnittstellen-Spezifikation (Entwurf)

OParl Team - <http://oparl.de/>

## Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Einleitung</b>  | <b>3</b> |
| 1.1      | Motivationen für den standardisierten Datenzugriff . . . . .                 | 4        |
| 1.2      | Funktionsumfang der OParl-Schnittstelle . . . . .                            | 4        |
| 1.3      | Status . . . . .   | 5        |
| 1.4      | Was ist OParl? . . . . .   | 5        |
| 1.5      | Zielsetzung von OParl . . . . .  | 5        |
| 1.6      | Werdegang von OParl 1.0 . . . . .  | 5        |
| 1.7      | Zukunft von OParl . . . . .  | 6        |
| 1.8      | Nomenklatur der Spezifikation und Satzkonventionen . . . . .                 | 6        |
| 1.8.1    | MÜSSEN, SOLLEN und KÖNNEN bzw. ZWINGEND, EMPFOHLEN<br>und OPTIONAL . . . . . | 6        |
| 1.8.2    | Besondere Hervorhebungen und Satzkonventionen . . . . .                      | 7        |
| 1.9      | Initiatoren . . . . .  | 7        |
| 1.10     | Unterstützer . . . . .   | 7        |
| 1.11     | Autoren . . . . .  | 7        |
| <b>2</b> | <b>Architektur</b>   | <b>7</b> |
| 2.1      | Überblick . . . . .  | 7        |
| 2.2      | Parlamentarisches Informationssystem . . . . .                               | 7        |
| 2.3      | Server . . . . .   | 8        |
| 2.4      | API . . . . .  | 8        |
| 2.5      | Client . . . . .   | 8        |
| 2.6      | Cache . . . . .  | 9        |
| 2.7      | Nutzerin oder Nutzer . . . . .   | 9        |
| <b>3</b> | <b>Nutzungsszenarien</b>   | <b>9</b> |
| 3.0.1    | Szenario 1: Mobile Client-Anwendung . . . . .                                | 9        |
| 3.0.2    | Szenario 2: Integration in Web-Portal . . . . .                              | 10       |
| 3.0.3    | Szenario 3: Meta-Suche . . . . .   | 10       |
| 3.0.4    | Szenario 4: Forschungsprojekt Themen- und Sprachanalyse . . . . .            | 10       |

|          |  |           |
|----------|--|-----------|
| <b>4</b> | <b>Prinzipien und Funktionen der API</b>             | <b>10</b> |
| 4.1      | Designprinzipien . . . . .                           | 11        |
| 4.1.1    | Allgemein . . . . .                                  | 11        |
| 4.1.2    | RESTful . . . . .                                    | 11        |
| 4.1.3    | Selbstbeschreibungsfähigkeit . . . . .               | 11        |
| 4.1.4    | Erweiterbarkeit . . . . .                            | 11        |
| 4.1.5    | Browseability/Verlinkung . . . . .                   | 11        |
| 4.2      | Zukunftssicherheit . . . . .                         | 11        |
| 4.3      | HTTP und HTTPS . . . . .                             | 11        |
| 4.4      | Serialisierung mittels JSON-LD und JSONP . . . . .   | 11        |
| 4.4.1    | JSON . . . . .                                       | 11        |
| 4.4.2    | JSON-LD . . . . .                                    | 11        |
| 4.4.3    | JSONP . . . . .                                      | 12        |
| 4.5      | Benannte und anonyme Objekte . . . . .               | 12        |
| 4.5.1    | Benannte Objekte . . . . .                           | 12        |
| 4.5.2    | Anonyme Objekte (Blank Nodes) . . . . .              | 12        |
| 4.5.3    | Empfehlungen für langlebige IRIs/URIs/URLs . . . . . | 12        |
| 4.5.4    | Empfehlungen für eindeutige URLs . . . . .           | 12        |
| 4.6      | Objektlisten . . . . .                               | 12        |
| 4.7      | Feeds . . . . .                                      | 13        |
| 4.7.1    | Neue Objekte . . . . .                               | 13        |
| 4.7.2    | Geänderte Objekte . . . . .                          | 13        |
| 4.7.3    | Entfernte Objekte . . . . .                          | 13        |
| 4.8      | Dokumentenabruf . . . . .                            | 13        |
| 4.9      | Ausnahmebehandlung . . . . .                         | 13        |
| <b>5</b> | <b>Datenmodell</b>                                   | <b>14</b> |
| 5.1      | Übergreifende Aspekte . . . . .                      | 14        |
| 5.1.1    | Eindeutige Identifizierung von Objekten . . . . .    | 14        |
| 5.1.2    | Objekteigenschaften . . . . .                        | 14        |
| 5.1.3    | Zu den Beziehungen . . . . .                         | 14        |
| 5.2      | Körperschaft ( <i>body</i> ) . . . . .               | 15        |
| 5.2.1    | Eindeutige Identifizierung . . . . .                 | 15        |
| 5.2.2    | Eigenschaften . . . . .                              | 15        |
| 5.2.3    | Beziehungen . . . . .                                | 16        |
| 5.2.4    | Beispiel . . . . .                                   | 16        |
| 5.3      | Gremium ( <i>committee</i> ) . . . . .               | 16        |
| 5.3.1    | Eigenschaften . . . . .                              | 17        |
| 5.3.2    | Beziehungen . . . . .                                | 17        |
| 5.3.3    | Beispiel . . . . .                                   | 17        |

|          |  |           |
|----------|--|-----------|
| 5.4      | Person ( <i>person</i> )                 | 17        |
| 5.4.1    | Eigenschaften                            | 17        |
| 5.4.2    | Beziehungen                              | 18        |
| 5.4.3    | Beispiel                                 | 18        |
| 5.5      | Organisation ( <b>organisation</b> )     | 19        |
| 5.5.1    | Eigenschaften                            | 19        |
| 5.5.2    | Beziehungen                              | 20        |
| 5.5.3    | Beispiel                                 | 20        |
| 5.6      | Sitzung ( <b>meeting</b> )               | 20        |
| 5.6.1    | Eigenschaften                            | 20        |
| 5.6.2    | Beziehungen                              | 21        |
| 5.6.3    | Beispiel                                 | 21        |
| 5.7      | Tagesordnungspunkt ( <b>agendaitem</b> ) | 21        |
| 5.7.1    | Eigenschaften                            | 22        |
| 5.7.2    | Beziehungen                              | 23        |
| 5.7.3    | Beispiel                                 | 23        |
| 5.8      | Drucksache ( <b>paper</b> )              | 23        |
| 5.8.1    | Eigenschaften                            | 24        |
| 5.8.2    | Beziehungen                              | 24        |
| 5.8.3    | Beispiel                                 | 25        |
| 5.9      | Dokument ( <b>document</b> )             | 25        |
| 5.9.1    | Eigenschaften                            | 25        |
| 5.9.2    | Beziehungen                              | 26        |
| 5.10     | Ort ( <b>location</b> )                  | 27        |
| 5.10.1   | Eigenschaften                            | 27        |
| 5.10.2   | Beziehungen                              | 27        |
| <b>6</b> | <b>Fußnoten</b>                          | <b>27</b> |
| <b>7</b> | <b>Glossar</b>                           | <b>28</b> |

Lizenz: Creative Commons CC-BY-SA

## 1 Einleitung

Dieses Dokument wird bei seiner Fertigstellung die Spezifikation des OParl Schnittstellen-Standards für parlamentarische Informationssysteme (Ratsinformationssysteme, RIS) darstellen. Es dient damit als Grundlage für die Implementierung von OParl-konformen Server- und Clientanwendungen.

## 1.1 Motivationen für den standardisierten Datenzugriff

Die Gründe, warum Betreiber von parlamentarischen Informationssystemen den Zugriff darauf über eine standardisierte Schnittstelle ermöglichen sollten, können vielfältig sein.

Ein zentrales Argument ist die Verpflichtung der Parlamente gegenüber der Bevölkerung, diese über die Fortschritte der parlamentarischen Arbeit zu informieren und auf dem Laufenden zu halten. Ein erster Schritt, der Bevölkerung Einblicke in die Arbeit und Zugriff auf Dokumente zu gewähren, ist vielerorts in den letzten Jahren durch Einführung von Ratsinformationssystemen mit anonymem, lesenden Zugriff über das World Wide Web gemacht worden.

Die damit eingeschlagene Richtung konsequent weiter zu gehen, bedeutet, die Daten der parlamentarischen Informationssystemen gänzlich offen zu legen, sofern die Inhalte es erlauben. Es bedeutet, die Daten und Inhalte so universell weiterverwendbar und so barrierearm wie möglich anzubieten, dass jegliche weitere Verwendung durch Dritte technisch möglich ist. Der seit einiger Zeit etablierte Begriff für dieses Prinzip heißt “Open Data”.

Das Interesse an parlamentarischen Informationen und an Anwendungen, die diese nutzbar und auswertbar machen, ist offensichtlich vorhanden. Die Entwickler der alternativen Ratsinformationssysteme wie Frankfurt Gestalten[14], Offenes Köln[15] oder der OpenRuhr:RIS-Instanzen[16] wissen zu berichten, wie viel Interesse den Projekten gerade aus Orten entgegen gebracht wird, in denen derartige Systeme noch nicht verfügbar sind.

Die Anwendungsmöglichkeiten für parlamentarische Informationen, wenn sie über eine Schnittstelle schnell und einfach abgerufen werden können, sind vielfältig. Beispiele könnten sein:

- Apps für den Abruf auf mobilen Endgeräten
- Möglichkeiten zur Wiedergabe für Nutzerinnen und Nutzer mit Beeinträchtigung des Sehvermögens
- Alternative und erweiterte Suchmöglichkeiten in Inhalten
- Auswertung und Analyse von Themen, Inhalten, Sprache etc.
- Benachrichtigungsfunktionen beim Erscheinen bestimmte Inhalte

Die Standardisierung dieses Zugriffs über die Grenzen einzelner Systeme hinweg erlaubt zudem, diese Entwicklungen grenzüberschreitend zu denken. Damit steigt nicht nur die potenzielle Nutzerschaft einzelner Entwicklungen. Auch das Potenzial für Kooperationen zwischen Anwendungsentwicklern wächst.

Darüber hinaus sind auch Motivationen innerhalb von Organisationen und Körperschaften erkennbar. So sollen parlamentarische Informationssysteme vielerorts in verschiedenste Prozesse und heterogene Systemlandschaften integriert werden. Durch eine einheitliche Schnittstelle bieten sich effiziente Möglichkeiten zur Integration der Daten in anderen Systeme, wie beispielsweise Web-Portale.

## 1.2 Funktionsumfang der OParl-Schnittstelle

Die vorliegende Spezifikation soll eine Webservice-Schnittstelle definieren, die den anonymen und lesenden Zugriff auf öffentliche Inhalte aus Parlamentarischen Informationssystemen ermöglicht. Die Zugriffe erfolgen über das Hypertext Transfer Protocol (HTTP). Daten werden als JSON oder als JSONP ausgeliefert.

Die Spezifikation wird obligatorische Bestandteile (MUSS) und optionale Bestandteile (KANN) haben. Der tatsächliche Funktionsumfang kann daher zwischen den Implementierungen variieren.

### 1.3 Status

Die Spezifikation befindet sich in Arbeit. Das Dokument enthält entsprechend viele Ungenauigkeiten und Hinweise auf offene Fragestellungen.

### 1.4 Was ist OParl?

### 1.5 Zielsetzung von OParl

- Nutzen für Kommunen, Bürger, politische Parteien
- Nutzen für Anbieter von RIS-Pflegesoftware
- Nutzen für Anbieter von RIS-Darstellungssoftware
- Nutzen für Open Data Initiativen
- Nutzen für die Wissenschaft
- Linked Data erwähnen

### 1.6 Werdegang von OParl 1.0

Stichpunkte:

- 17. und 18. November 2012: Die Open Knowledge Foundation Deutschland veranstaltet in den Räumen der Heinrich-Böll-Stiftung in Berlin einen Workshop für Entwickler von Anwendungen, die einen gesellschaftlichen Nutzen bringen sollen. Hier ist VITAKO, die Bundes-Arbeitsgemeinschaft der Kommunalen IT-Dienstleister, als Sponsor engagiert. Die Geschäftsführerin, Dr. Marianne Wulff, ist persönlich vor Ort. Auch das Projekt Offenes Köln wird in einem Vortrag von Marian Steinbach präsentiert. Es kommt zum Austausch über die Frage, wie das Prinzip der offenen Ratsinformationen effektiv auf weitere Kommunen ausgeweitet werden könnte.
- 6. Dezember 2012: Anhörung im Landtag NRW in Düsseldorf zu einer Open-Data-Strategie der Landesregierung, wo Jens Klessmann und Marian Steinbach als Sachverständige gehört werden. Danach Gespräch über Möglichkeiten der Standardisierung offener Ratsinformationssysteme.
- Dezember 2012: Dr. Marianne Wulff, Jens Klessmann und Marian Steinbach beginnen mit der Abstimmung über einen Workshop mit Vertreterinnen und Vertretern von Kommunen, kommunalen IT-Dienstleistern, RIS-Anbietern und Zivilgesellschaft. Ziel: Die Bereitschaft zur Zusammenarbeit an einem gemeinsamen Standard ermitteln. Unterdessen beginnt Marian Steinbach mit der Formulierung eines Standard-Entwurfs als Diskussionsgrundlage. Der Entwurf wird von Beginn an öffentlich auf GitHub.com bereit gestellt.
- 17. April 2013: Insgesamt 30 Teilnehmer versammeln sich in Köln, um sich in einem ersten Treffen über Ziele und Chancen einer Standardisierung für offene Ratsinformationen auszutauschen. Als Ergebnis wird ein großes Interesse an der weiteren Zusammenarbeit auf Basis des vorliegenden Standardentwurfs festgestellt. Als Termin für die Fertigstellung der ersten Version der Spezifikation wird der 30. Juni 2013 festgelegt. Die Initiatoren präsentieren den Anwesenden hier erstmals den Namen "OParl", der künftig als Marke für die Bemühungen der Gruppe stehen soll.
- 22. Januar 2014: Nachdem sich die verteilte Zusammenarbeit am Standard-Entwurf seit April 2013 als nicht zielführend erwiesen hat, laden Jens Klessmann und Marian Steinbach und VITAKO zu einem eintägigen OParl-Workshop in Bielefeld ein. Das Ziel ist, die Spezifikation so weit wie möglich voran zu treiben und eine gute Basis für die baldige Fertigstellung zu legen.

## 1.7 Zukunft von OParl

- Verfeinerung, Lücken schliessen
- Globalisierung
- Erweiterung über die kommunale Ebene hinaus (Land, Bund)
- Vereinheitlichung von Kategorien (Drucksachentypen, Arten von Gremien)
- Erweiterung von Personendaten, z.B. mit Social Media URLs
- Mehr Abfragekriterien
- Suchfunktionen (Volltextsuche)
- Abstimmungsverhalten und maschinenlesbare Protokolle

## 1.8 Nomenklatur der Spezifikation und Satzkonventionen

### 1.8.1 MÜSSEN, SOLLEN und KÖNNEN bzw. ZWINGEND, EMPFOHLEN und OPTIONAL

Dieses Spezifikationsdokument nutzt die Modalverben müssen, können und sollen in einer Art und Weise, die bestimmte Anforderungen möglichst unmissverständlich in drei verschiedene Abstufung einteilen lässt. Um ihre normative Bedeutung zu unterstreichen, werden diese Wörter grundsätzlich in Großbuchstaben gesetzt.

Diese Konvention ist angelehnt an die Definitionen der Begriffe MUST, SHOULD und MAY (bzw. MUST NOT, SHOULD NOT und MAY NOT) aus RFC2119.<sup>1</sup>

Die Bedeutung im Einzelnen:

**MÜSSEN/MUSS bzw. ZWINGEND:** Die Erfüllung einer Anforderung, die explizit vom Modalverb MÜSSEN bzw. MUSS Gebrauch macht, ist zwingend erforderlich.

Die Entsprechung in RFC2119 lautet “MUST”, “REQUIRED” oder “SHALL”.

**NICHT DÜRFEN/DARF NICHT:** Dieses Stichwort kennzeichnet ein absolutes Verbot.

Die Entsprechung in RFC2119 lautet “MUST NOT” oder “SHALL NOT”.

**SOLLEN/SOLL bzw. EMPFOHLEN:** Mit dem Wort SOLLEN bzw. SOLL sind empfohlene Anforderungen gekennzeichnet, die von jeder Implementierung erfüllt werden sollen. Eine Nichterfüllung ist als Nachteil zu verstehen, beispielsweise weil die Nutzerfreundlichkeit dadurch Einbußen erleidet, und sollte daher sorgfältig abgewogen werden.

Die Entsprechung in RFC2119 lautet “SHOULD” oder “RECOMMENDED”.

**NICHT SOLLEN/SOLL NICHT bzw. NICHT EMPFOHLEN:** Diese Formulierung wird verwendet, wenn unter gewissen Umständen Gründe existieren können, die ein bestimmtes Verhalten akzeptabel oder sogar nützlich erscheinen lassen, jedoch die Auswirkung des Verhaltens vor einer entsprechenden Implementierung verstanden und abgewogen werden sollen.

Die Entsprechung in RFC2119 lautet “SHOULD NOT” oder “NOT RECOMMENDED”.

**DÜRFEN/DARF bzw. OPTIONAL:** Mit dem Wort DÜRFEN bzw. DARF oder OPTIONAL sind optionale Bestandteile gekennzeichnet. Ein Anbieter könnte sich entscheiden, den entsprechenden Bestandteil aufgrund besonderer Kundenanforderungen zu unterstützen, während andere diesen Bestandteil ignorieren könnten. Implementierer von Clients oder Servern DÜRFEN in solchen Fällen NICHT davon ausgehen, dass der jeweilige Kommunikationspartner den entsprechenden, optionalen Anteil unterstützt.

Die Entsprechung in RFC2119 lautet “MAY” oder “OPTIONAL”.

---

<sup>1</sup>RFC2119 <http://tools.ietf.org/html/rfc2119>

### 1.8.2 Besondere Hervorhebungen und Satzkonventionen

## 1.9 Initiatoren

## 1.10 Unterstützer

## 1.11 Autoren

An diesem Dokument haben mitgewirkt:

Felix Ebert, Jan Erhardt, Jens Klessmann, Andreas Kuckartz, Babett Schalitz, Marian Steinbach, Thomas Tursics, Jakob Voss

# 2 Architektur

In diesem Abschnitt werden grundlegenden Konzepte, die von OPaL abgedeckt werden, erläutert. Die Erläuterungen sind nicht im engeren Sinne Teil der Spezifikation, sondern dienen dazu, die Anwendungsbereiche von OPaL und die Funktionen einer OPaL-konformen API verständlicher und konkreter beschreiben zu können.

Da die Architektur auf der generellen Architektur des World Wide Web (WWW) aufbaut, sind einzelne Konzepte direkt den Begriffen der Architekturbeschreibung des W3-Konsortiums entlehnt.<sup>2</sup>

## 2.1 Überblick

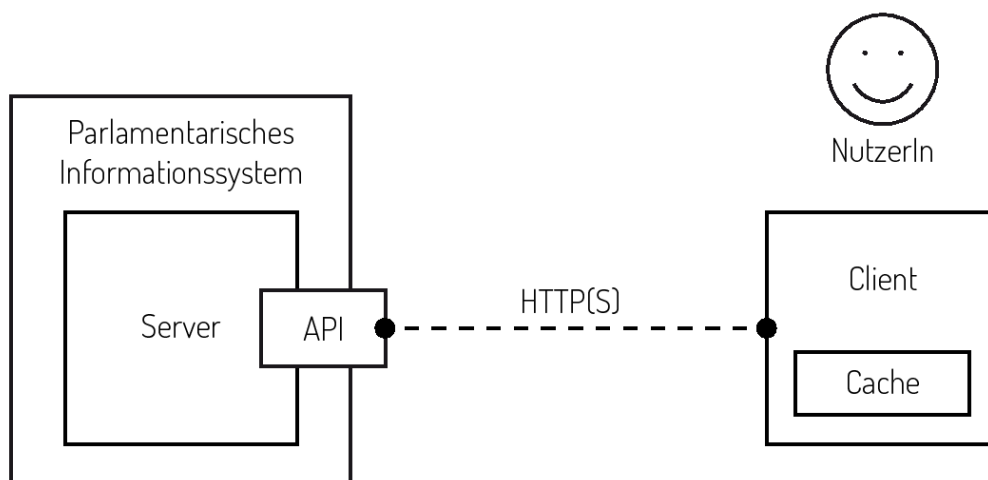


Figure 1: Architekturdiagramm

## 2.2 Parlamentarisches Informationssystem

Parlamentarische Informationssysteme sind Software-Systeme, die von verschiedensten Körperschaften eingesetzt werden, um die Zusammenarbeit von Parlamenten zu organisieren, zu dokumentieren und öffentlich nachvollziehbar zu machen.

<sup>2</sup>Architecture of the World Wide Web, Volume One. <http://www.w3.org/TR/webarch/>

Im kommunalen Umfeld in Deutschland, wo das Parlament je nach Art der Kommune häufig als Stadtrat oder Gemeinderat bezeichnet wird, hat sich für diese Art von Informationssystem auch der Begriff “Ratsinformationssystem” (kurz “RIS”) etabliert.

Parlamentarische Informationssysteme sind jedoch nicht auf die kommunale Ebene begrenzt. Ähnliche Systeme werden auch auf Ebene z.B. von Landkreisen, Regierungsbezirken und diversen Zweckverbänden eingesetzt.

Diese Systeme unterstützen in der Regel mehrere der folgenden Funktionen:

- Das Erzeugen, Bearbeiten und Darstellen von Sitzungen und deren Tagesordnung
- Das Erzeugen und Abrufen von Sitzungsprotokollen
- Das Erzeugen, Bearbeiten und Anzeigen von Drucksachen
- Das Erzeugen, Bearbeiten und Anzeigen von Gremien und deren Mitgliedern

Funktionen, die die Eingabe und Bearbeitung von Daten betreffen, sind in der Regel einem geschlossenen Nutzerkreis vorbehalten. Die Darstellung und der Abruf von Informationen und Dokumenten hingegen ist in vielen Fällen für die Öffentlichkeit freigegeben.

Die OParl Spezifikation beschreibt eine Schnittstelle, die den maschinellen, lesenden Zugriff auf derartige Informationen ermöglicht.

## 2.3 Server

Der Server im Sinne dieser Spezifikation ist ein Software-Dienst, der auf einem mit dem Internet verbundenen Rechnersystem läuft. Dieser Dienst ist eine spezielle Form eines WWW- bzw. HTTP(S)-Servers. Entsprechend beantwortet der Server HTTP-Anfragen, die an ihn auf einem bestimmten TCP-Port gestellt werden.

Der Server ist als Bestandteil des parlamentarischen Informationssystems zu verstehen. Der Betrieb des Servers steht damit üblicherweise in der Verantwortung desjenigen, der das parlamentarische Informationssystem betreibt.

Von einem Server, der die OParl-Spezifikation erfüllt, wird erwartet, dass er bestimmte parlamentarische Informationen in einem bestimmten Format zur Verfügung stellt und auf bestimmte Anfragen von so genannten Clients über die OParl API entsprechend dieser Spezifikation reagiert.

## 2.4 API

Der Begriff API steht in diesem Dokument für die Webservice-Schnittstelle, die der Server anbietet. Die Schnittstelle basiert auf dem HTTP-Protokoll. Mittels HTTPS ist wahlweise auch die verschlüsselte Nutzung der API möglich, sofern Server dies unterstützt.

Die API steht im Mittelpunkt dieser Spezifikation. Server und Clients sind als Kommunikationspartner zu verstehen, die über das Internet als Kommunikationskanal mit einander kommunizieren können. Die API-Spezifikation stellt dabei die nötige Grammatik und das Vokabular bereit, anhand dessen eine sinnvolle Kommunikation erfolgen kann.

## 2.5 Client

Der Begriff “Client” steht für eine Software, die über die OParl API mit dem Server kommuniziert. Da die API auf dem HTTP-Protokoll aufbaut, handelt es sich bei dem Client um eine spezielle Form eines HTTP-Clients.



## 2.6 Cache

Ein Cache ist ein Speicher, der einem Client dazu dienen kann, von einem Server abgerufene Informationen längerfristig vorzuhalten. Dies kann beispielsweise dazu dienen, mehrfache Anfragen der selben Informationen zu vermeiden, wodurch sowohl Ressourcen auf Seite des Servers geschohnt als auch die Nutzung von Netzwerkbandbreite reduziert werden kann. Die Nutzung eines Cache kann auch zur Verbesserung der Nutzerfreundlichkeit eines Clients beitragen, indem Wartezeiten zur Bereitstellung einer Ressource verkürzt werden.

## 2.7 Nutzerin oder Nutzer

Mit einer Nutzerin oder einem Nutzer ist in diesem Fall eine natürliche Person gemeint, die mittels eines OParl-Clients auf parlamentarische Informationen zugreift.

# 3 Nutzungsszenarien

Die nachfolgenden Nutzungsszenarien dienen dazu, die Architektur und die Anwendungsmöglichkeiten anhand konkreter Beispiele zu verdeutlichen. Sie erheben keinen Anspruch auf Vollständigkeit.

### 3.0.1 Szenario 1: Mobile Client-Anwendung

Eine **Client**-Anwendung für mobile Endgeräte wie SmartPhones und Tablets, nachfolgend “App” genannt, könnte das Ziel verfolgen, Nutzern unterwegs sowie abseits vom Desktop-PC bestmöglichen Lesezugriff auf Dokumente aus Ratsinformationssystemen (RIS) zu bieten. Die möglichen Kontexte und Nutzungsmotivationen sind vielfältig:

- Teilnehmer einer Sitzung greifen während der Sitzung auf die Einladung dieser Sitzung und die zur Tagesordnung der Sitzung gehörenden Drucksachen zu, außerdem auf die Protokolle vorheriger Sitzungen.
- Eine Redakteurin der Lokalpresse geht unterwegs die Themen der nächsten Sitzungen bestimmter Gremien, für die sie sich besonders interessiert, durch.
- Eine Gruppe von Studierenden erkundet zusammen mit ihrem Dozenten die lokalpolitischen Aktivitäten des Viertels rund um ihre Hochschule. Dazu nutzen sie die GPS-Lokalisierung ihrer Smartphones in Verbindung mit den Geodaten, die an vielen Drucksachen des lokalen RIS zu finden sind. Direkt vor Ort an einer Baustelle öffnen sie Beschlüsse, Pläne und Eingaben aus dem Planfeststellungsverfahren, die dieser Baustelle voran gegangen sind.

Zur Realisierung derartiger Szenarien können die Fähigkeiten von OParl-kompatiblen Servern mit den besonderen Eigenschaften der mobilen Endgeräte verknüpft werden.

Smartphones und Tablets verfügen beispielsweise, je nach Aufenthaltsort, über sehr unterschiedlich gute Internetanbindung. In einem Büro oder zuhause können Nutzer über ein WLAN Daten mit hoher Bandbreite austauschen, in Mobilfunknetzen vor allem außerhalb der Ballungsgebiete jedoch sinken die Bandbreiten deutlich. Einige Tablets werden sogar ohne Möglichkeit zur Mobilfunk-Datenübertragung genutzt. In solchen Fällen kann ein **Cache** auf dem Endgerät dazu dienen, Inhalte vorzuhalten, die dann auch bei langsamer oder fehlender Internetverbindung zur Verfügung stehen. Sobald dann wieder eine Verbindung mit hoher Bandbreite bereit steht, kann die App im Hintergrund, entweder über die **Feeds** der OParl API oder über den einzelnen Abruf von Objekten, die gecachten Inhalte aktualisieren.

Eine Stärke eines mobilen Clients ist auch die Möglichkeit der Personalisierung, also der Anpassung auf die Bedürfnisse und Interessen der Nutzerin oder des Nutzers. Es wäre

beispielsweise denkbar, dass eine Nutzerin die Ratsinformationssysteme, für die sie sich interessiert, dauerhaft in der App einrichtet und eine Favoritenliste der Gremien, die ihre bevorzugten Themengebiete behandeln, hinterlegt. Die App könnte aufgrund dieser Favoritenliste eigenständig über die API nach neuen Sitzungsterminen, Tagesordnungspunkten, Drucksachen und Dokumente suchen. Taucht dabei ein neues Objekt auf, wird die Nutzerin darüber benachrichtigt. Sie kann dann beispielsweise entscheiden, Dokumente direkt zu öffnen oder für den späteren Offline-Zugriff zu speichern.

Einem derartigen Szenario kommt das Graph-orientierte Datenmodell der OParl API entgegen. Ausgehend von einer Sitzung eines bestimmten Gremiums beispielsweise ist es damit einfach möglich, die in Verbindung stehenden Mitglieder des Gremiums, Teilnehmer der Sitzung, Tagesordnungspunkte der Sitzung oder Drucksachen zu den Tagesordnungspunkten und letztlich Dokumente zu Drucksachen und Sitzung abzurufen.

Für die Nutzer einer mobilen Client-Anwendung könnte es sich als besonders hilfreich erweisen, wenn Dokumente auf dem Server in verschiedenen Formaten zur Verfügung gestellt werden. Denn nicht jedes Endgerät mit kleinem Bildschirm bietet eine nutzerfreundliche Möglichkeit, beispielsweise Dokumente im weit verbreiteten PDF-Format darzustellen. Hier könnte schon der Entwickler der mobilen App Mechanismen vorsehen, die, sofern vorhanden, besser geeignete Formate wie z.B. HTML abrufen.

Neben dem kleinen Display kann für einige mobile Endgeräte auch die im Vergleich zu einem zeitgemäßen Desktop-PC geringere CPU-Leistung eine Einschränkung darstellen. Solchen Geräten kommt es besonders entgegen, wenn der Server zu allen Dokumenten auch den reinen Textinhalt abrufbar macht, der dann beispielsweise für eine Volltextsuche auf dem Endgerät indexiert werden kann. So wiederum kann auf dem Client eine Suchfunktion realisiert werden, welche die OParl-API selbst nicht zur Verfügung stellt.

Eine solche Suchfunktion kann auch über die reine Volltextsuche hinaus gehen und über die Suche mittels Text- oder Spracheingabe hinaus gehen. Denn ein Client könnte von einem **Server**-System, das Drucksachen mit Geoinformationen anbietet, diese abrufen und räumlich indexieren. Anhand der Position des Geräts, die mittels GPS genau bestimmt werden kann, könnte so der lokale Cache nach Objekten in der Umgebung durchsucht werden. Das Ergebnis könnte auf einer Karte dargestellt oder in einer Ergebnisliste angezeigt werden, die nach Distanz zum Objekt sortiert werden könnte.

### **3.0.2 Szenario 2: Integration in Web-Portal**

### **3.0.3 Szenario 3: Meta-Suche**

### **3.0.4 Szenario 4: Forschungsprojekt Themen- und Sprachanalyse**

## **4 Prinzipien und Funktionen der API**

(In diesem Kapitel werden die Zugriffsmethoden der OParl-konformen Schnittstelle beschrieben. Hierzu gehören alle chapter-Dateien, deren Nummerierung mit der Ziffer 6 beginnt.)

Stichpunkte:

- Grundlage für den Zugriff auf die Schnittstelle ist das Hypertext Transfer Protocol (HTTP).
- Optional gzip Encoding und andere Kodierungen, wenn Client und Server dies unterstützen
- Das Protokoll ist zustandslos
- Authentifizierung wird nicht benötigt.

## 4.1 Designprinzipien

### 4.1.1 Allgemein

- Ausgerichtet am Status Quo
- Einige Möglichkeiten weisen in die Zukunft (z.B. Geodaten, Volltexte, Vereinheitlichung der Kategorien-Nomenklatur)
- Ziel: Schnelle Implementierbarkeit in vielen bestehenden RIS

### 4.1.2 RESTful

### 4.1.3 Selbstbeschreibungsfähigkeit

### 4.1.4 Erweiterbarkeit

### 4.1.5 Browseability/Verlinkung

## 4.2 Zukunftssicherheit

(In diesem Abschnitt wird die Zielsetzung beschrieben, zukünftige Versionen der OParl Spezifikation auf dieser Version aufbauen zu lassen. Damit soll nach Möglichkeit Abwärtskompatibilität erreicht werden, so dass z.B. OParl-1.0-Clients mit einem 1.1-Server kommunizieren können.

Ein Garantie wird es jedoch nicht geben.)

## 4.3 HTTP und HTTPS

(Hier geht es um die Verwendung von HTTP im Sinne der OParl-Spezifikation.

- HTTPS kann optional an Stelle von HTTP eingesetzt werden.

Siehe auch <https://github.com/OParl/specs/issues/90>)

## 4.4 Serialisierung mittels JSON-LD und JSONP

### 4.4.1 JSON

- Siehe RFC4627
- Generelle Terminologie übernehmen (JSON-Objekt, Array, String, Number, true/false, null)

### 4.4.2 JSON-LD

- JSON-LD: <http://www.w3.org/TR/json-ld/>
- Einschränkungen von OParl gegenüber JSON-LD
- Schlüssel in einem JSON-LD-Objekt müssen einzigartig sein.
- Unterscheidung von Groß- und Kleinschreibung
- Benannte Objekte (URL als Schlüssel)
- Anonyme Objekte (Blank Nodes)

- Mime Type application/ld+json
- Kompakte Form mit Verwendung externer @context-URL als SOLL-Anforderung
- Verweis auf [http://www.bmi.bund.de/SharedDocs/Downloads/DE/Themen/OED\\_Verwaltung/ModerneVerwa](http://www.bmi.bund.de/SharedDocs/Downloads/DE/Themen/OED_Verwaltung/ModerneVerwa)
- Siehe <https://github.com/OParl/specs/issues/77>
- Siehe <https://github.com/OParl/specs/issues/10>

#### 4.4.3 JSONP

- TODO: Spezifikation finden/verlinken. (RFC gibt es nicht)
- <https://github.com/OParl/specs/issues/67>
- Zeichenvorrat für callback-Parameter beschränken auf [a-zA-Z0-9] aus Sicherheitsgründen

### 4.5 Benannte und anonyme Objekte

#### 4.5.1 Benannte Objekte

- IRIs (RFC3987) als eindeutige Kennung

#### 4.5.2 Anonyme Objekte (Blank Nodes)

#### 4.5.3 Empfehlungen für langlebige IRIs/URIs/URLs

- Hinweise und evtl. Auszüge aus
- <http://www.w3.org/Provider/Style/URI.html>
- <https://joinup.ec.europa.eu/sites/default/files/D7.1.3%20-%20Study%20on%20persistent%20URIs.pdf>

#### 4.5.4 Empfehlungen für eindeutige URLs

- Vermeidung von Duplicate Content durch Fehlkonfiguration
- z.B.: verschiedene CNAMEs mit der selben IP-Adresse
- z.B.: Aufruf über <http://www.example.com> und <http://example.com>
- z.B.: Direkter Aufruf über IP-Adresse <http://1.2.3.4/>
- URL-Parameter in definierter Reihenfolge verwenden
- Groß- und Kleinschreibung unterscheiden

### 4.6 Objektlisten

(Hier geht es darum, wie Listen von Objekten ausgegeben werden.

- Ausgabe von URLs
- Stabile Sortierung (Sets vs. Lists)
- Paginierung
- Einschränkung auf Datumsbereiche

Dazu:

- <https://github.com/OParl/specs/issues/66>
- <https://github.com/OParl/specs/issues/30> )

## 4.7 Feeds

- Warum Feeds?
- Möglichst ressourcenschonendes Auffinden neuer und geänderter Objekte
- Cache-Invalidierung, Entfernung von gelöschten/depublizierten Objekten
- <https://github.com/OParl/specs/issues/19>

### 4.7.1 Neue Objekte

### 4.7.2 Geänderte Objekte

### 4.7.3 Entfernte Objekte

- Liste von Objekt-URLs

## 4.8 Dokumentenabruf

- HTTP GET Methode MUSS unterstützt werden
- HEAD-Methode MUSS unterstützt werden
- HTTP Last-Modified Header sowie Conditional GET sind zu unterstützen

## 4.9 Ausnahmebehandlung

(Diskussion hierzu unter <https://github.com/OParl/specs/issues/89>) Liste reservierter URL-Parameter \_\_\_\_\_

Die in dieser Liste enthaltenen Zeichenketten haben eine reservierte Bedeutung und stehen bei Implementierungen eines OParl-Servers nicht mehr für die freie Verwendung in URLs zur Verfügung.

**callback:** Mit diesem Parameter wird die JSONP-Ausgabe aktiviert. Mehr dazu im Abschnitt **JSONP**.

**startdate:** Parameter für die Einschränkung einer Abfrage anhand eines Datums bzw. einer Zeitangabe.

**enddate:** Parameter für die Einschränkung einer Abfrage anhand eines Datums bzw. einer Zeitangabe.

- (Parameter für Datums-/Zeitbereichsfilter)

## 5 Datenmodell

Das Datenmodell definiert die Objekttypen bzw. die Klassen, auf die über die Schnittstelle zugegriffen werden kann.

Die Hinweise auf die Praxis in bestehenden Ratsinformationssystemen beziehen sich auf nach außen, bei Nutzung der jeweiligen Weboberflächen, feststellbare Eigenschaften. Dabei wird vereinzelt und beispielhaft auf die folgenden Systeme Bezug genommen:

- Stadt Köln [2] - Plattform: Somacos SessionNet [3]
- Bezirksverwaltung Berlin Mitte [4] - Plattform: ALLRIS [5]
- Stadt Rösrath [6] - Plattform der Firma PROVOX [7]
- Stadt Euskirchen [8] - Plattform: SD.NET RIM 4 [9]
- Stadt Bonn - BoRis [10]

### 5.1 Übergreifende Aspekte

#### 5.1.1 Eindeutige Identifizierung von Objekten

Sämtliche Objekte, die über die Schnittstelle geladen werden können, sollen anhand einer einzigen Objekteigenschaft eindeutig identifizierbar sein. Die Objekteigenschaft, mit der dies erreicht wird, wird hier im folgenden - unabhängig vom tatsächlichen Namen der Eigenschaft - als "Schlüssel" bezeichnet.

Eindeutigkeit meint hier eine Einzigartigkeit innerhalb des Informationssystems und für den jeweiligen Objekttyp. Das bedeutet: zwei von einander unabhängige Ratsinformationssysteme für verschiedene Körperschaften dürfen sich überlappende Schlüssel nutzen. Innerhalb eines Systems dürfen zwei Objekte unterschiedlichen Typs (beispielsweise eine Person und ein Gremium) den selben Schlüssel nutzen. Jedoch MÜSSEN zwei Objekte des selben Typs innerhalb des selben Systems grundsätzlich verschiedene Schlüssel haben.

Schlüssel-Eigenschaften werden grundsätzlich als Unicode-Zeichenkette übergeben. Sie können daher gleichermaßen aus numerischen wie alphanumerischen Werten befüllt werden.

Es wird grundsätzlich vorausgesetzt, dass Schlüssel unveränderlich sind. Ändert sich der Schlüssel eines Objekts nach der Erzeugung, werden Nutzer der Schnittstelle annehmen, dass es sich nicht mehr um das selbe Objekt handelt.

#### 5.1.2 Objekteigenschaften

Die nachfolgend beschriebenen Eigenschaften der Objekttypen sind, wenn nicht anders angegeben, verpflichtend. Das bedeutet: Bei jedem von der Schnittstelle ausgelieferten Objekt muss diese Eigenschaften definiert sein. Optionale Eigenschaften sind entsprechend gekennzeichnet.

Eigenschaften werden deutschsprachig und englischsprachig benannt. Die deutschsprachige Benennung dient der bestmöglichen Verständlichkeit im Kontext dieses Dokuments, während die Schnittstelle aus Gründen der Zugänglichkeit für möglichst viele Entwickler mit englischsprachigen Begriffen arbeiten soll.

#### 5.1.3 Zu den Beziehungen

Bei der Beschreibung von Beziehungen zwischen Objekten wird zu diesem Zeitpunkt nicht berücksichtigt, ob eine Beziehung zwischen zwei Objekten A und B am Objekt A oder am Objekt B definiert wird. So spielt es bislang keine Rolle, ob einem Gremium mehrere Personen zugeordnet werden oder einer Person mehrere Gremien zugewiesen werden. Das Augenmerk liegt hier nur auf der Tatsache, welche Beziehung existieren können und was diese Beziehungen aussagen sollen.

## 5.2 Körperschaft (*body*)

Die Körperschaft erlaubt es, den Betreiber bzw. Eigentümer des Informationssystems wie zum Beispiel einen Landkreis, eine bestimmte Gemeinde oder einen bestimmten Stadtbezirk in Form eines Datenobjekts abzubilden.

Viele RIS werden nur genau eine Instanz dieses Typs „beherbergen“. Einige Systeme werden jedoch für mehrere Mandanten betrieben, wobei die Mandanten verschiedene Körperschaften repräsentieren (z.B. „Verbandsgemeinde Ulmen“ und „Stadt Ulmen“.)

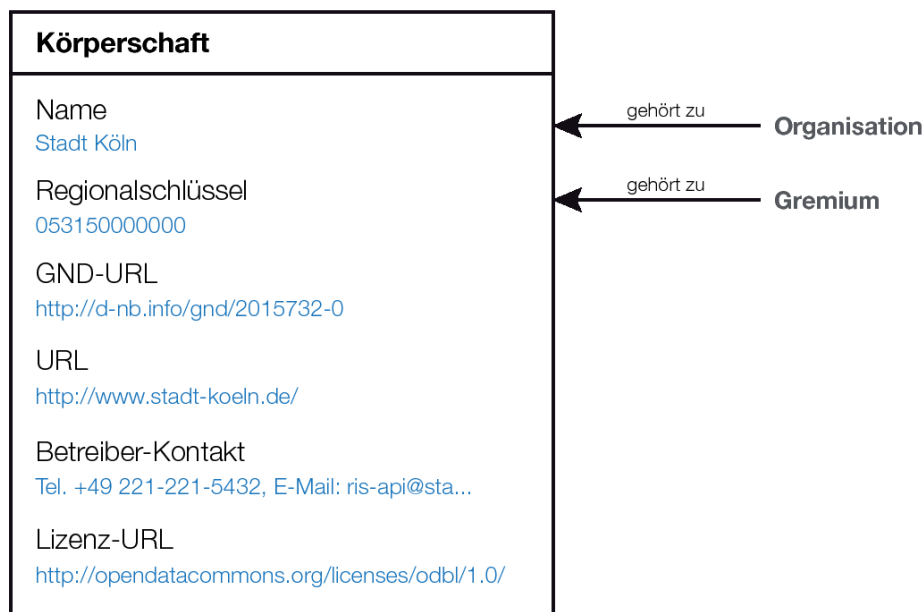


Figure 2: Objekttyp Körperschaft

### 5.2.1 Eindeutige Identifizierung

Die Körperschaft hat eine innerhalb des Systems eindeutige ID.

Darüber hinaus werden verschiedene Möglichkeiten geboten, die Körperschaft semantisch zu repräsentieren.

Handelt es sich beim Betreiber des Systems um eine Gebietskörperschaft (Landkreis, Kommune etc.), soll für die eindeutige Identifizierung der Regionalschlüssel[1] verwendet werden.

Darüber hinaus soll zusätzlich, sofern vorhanden, die eindeutige Kennung der Körperschaft aus der GND[12] verwendet werden.

Als dritte Möglichkeit, die Körperschaft zu identifizieren, kann eine aussagekräftige URL, unter der weitere Informationen zur Körperschaft zu finden sind, genannt werden.

Sämtliche hier genannten Methoden zur Identifizierung können kombiniert werden.

### 5.2.2 Eigenschaften

**Schlüssel (id)** Zur eindeutigen Identifizierung der Körperschaft im System

**Name (name)** Der Name der Körperschaft, z.B. „Stadt Köln“

**Regionalschlüssel (regionalschlüssel)** *Optional.* Regionalschlüssel der Gebietskörperschaft, z.B. „05315000000“. Muss grundsätzlich 12-stellig angegeben werden.

**GND URL (gnd\_url)** *Optional*. URL des Eintrags in der GND, z.B. “http://d-nb.info/gnd/2015732-0”

**URL (url)** *Optional*. URL der Homepage oder einer vergleichbaren Seite mit Informationen über die Körperschaft, z.B. “http://www.stadt-koeln.de/”

**Lizenz (license\_url)** *Optional*. URL der Lizenz, unter der die Daten, die über die API abgerufen werden können, stehen.

**Betreiber-Kontakt (operator\_contact)** *Optional*. Kontaktinformationen für die direkte Kontaktaufnahme zum Betreiber der API.

### 5.2.3 Beziehungen

- Objekte vom Typ “Organisation” sind zwingend genau einer Körperschaft zugeordnet. So wird beispielsweise eine SPD in Köln von einer SPD in Leverkusen unterschieden.
- Objekte vom Typ “Gremium” sind zwingend genau einer Körperschaft zugeordnet. Damit wird der “Rat” einer bestimmten Kommune von den gleichnamigen Gremien anderer Kommunen abgegrenzt.

### 5.2.4 Beispiel

```
{
  "id": "1",
  "name": "Stadt Köln",
  "regionalschlüssel": "053150000000",
  "gnd_url": "http://d-nb.info/gnd/2015732-0",
  "url": "http://www.stadt-koeln.de/",
  "operator_contact": "Tel. +49 221-221-5432, E-Mail: ris-api@stadt-koeln.de",
  "license_url": "http://opendatacommons.org/licenses/odbl/1.0/"
}
```

## 5.3 Gremium (*committee*)

Das Gremium ist ein Personenkreis, üblicherweise von gewählten und/oder ernannten Mitgliedern. Beispiele hierfür sind der Stadtrat, Kreisrat, Gemeinderat, Ausschüsse und Bezirksvertretungen. Gremien halten Sitzungen ab, zu denen die Gremien-Mitglieder eingeladen werden.

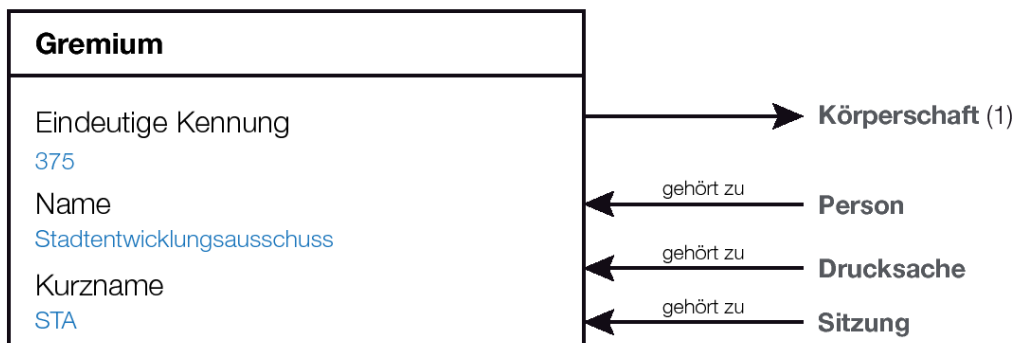


Figure 3: Objekttyp Gremium



### 5.3.1 Eigenschaften

**Schlüssel (id)** Zur eindeutigen Identifizierung des Gremiums im Kontext einer bestimmten Körperschaft. In der Praxis kommen sowohl numerische IDs als auch Namenskürzel (Beispiel: “STA” für den Stadtentwicklungsausschuss) vor. Beides sollte hier Verwendung finden können.

**Name (name)** Der Name des Gremiums. Beispiele: “Rat”, “Hauptausschuss”, “Bezirksvertretung 1 (Innenstadt)”

**Kurzname (short\_name)** *Optional.* Eine zur Anzeige bestimmte, kürzere Form des Namens.

**Zuletzt geändert (last\_modified)** Datum und Uhrzeit der letzten Änderung

### 5.3.2 Beziehungen

- Objekte vom Typ “Person” referenzieren auf Gremien, um die Mitgliedschaft/Zugehörigkeit einer Person im/zum Gremium zu kennzeichnen. Diese Beziehung ist datiert. Das bedeutet, sie hat einen Anfangszeitpunkt und ggf. einen Endzeitpunkt.
- Objekte vom Typ “Drucksache” verweisen auf Gremien. Beispielsweise wird eine Anfrage oder ein Antrag dem Rat und/oder einer bestimmten Bezirksvertretung zugeordnet. Details zu dieser Beziehung werden unter “Drucksache” erläutert.
- Das Gremium verweist auf die Körperschaft, zu der das Gremium gehört.

### 5.3.3 Beispiel

```
{
  "id": "7",
  "name": "Finanzausschuss",
  "short_name": "FA",
  "body": "1",
  "last_modified": "2012-08-16T14:05:27+02:00"
}
```

## 5.4 Person (*person*)

Jede natürliche Person, die Mitglied eines Gremiums ist, ist als Person im Datenmodell eindeutig identifizierbar.

### 5.4.1 Eigenschaften

**Schlüssel (id)** Zur eindeutigen Identifizierung sollte jede Person eine Kennung besitzen, die keinen Änderungen unterworfen ist und aus diesem Grund nicht mit dem Namen in Verbindung stehen sollte. Viele RIS nutzen rein numerische Kennungen.

**Vorname (first\_name)** Der Vorname der Person.

**Nachname (last\_name)** Der Nachname der Person.

**Titel (academic\_title)** *Optional.* Akademische Titel wie “Dr.” und “Prof. Dr.”

**Geschlecht (sex)** *Optional.* Weiblich (Wert F für *female*), männlich (Wert M für *male*), anderes (Wert 0 für *others*)

**Beruf (profession)** *Optional.* Z.B. “Rechtsanwalt”

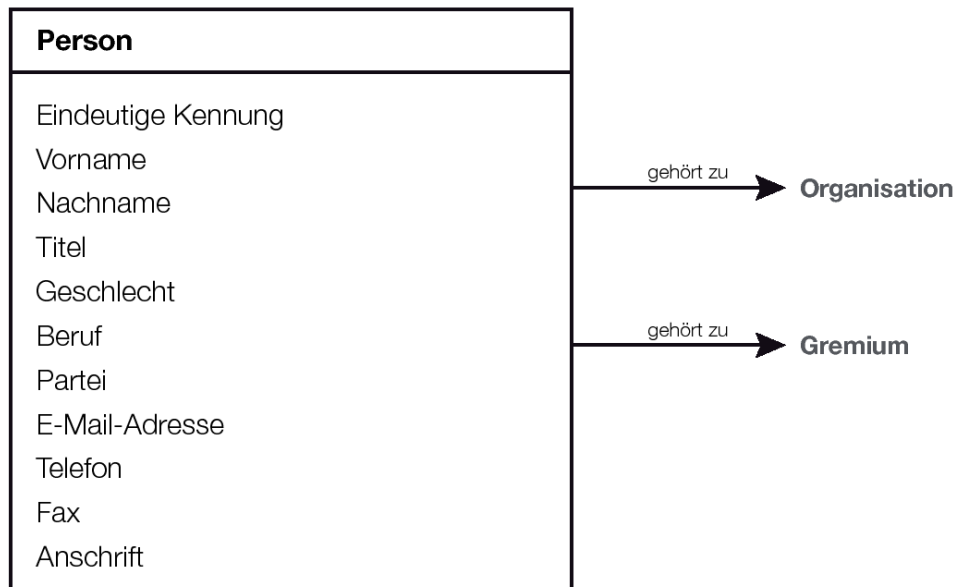


Figure 4: Objekttyp Person

**E-Mail-Adresse (email)** *Optional.*

**Telefon (phone)** *Optional.*

**Fax (fax)** *Optional.*

**Anschrift (address)** *Optional.* Straße und Hausnummer, Postleitzahl und Ort

**Zuletzt geändert (last\_modified)** Datum und Uhrzeit der letzten Änderung

### Anmerkungen

- Das System von Euskirchen scheint Vor- und Nachname (evtl. einschl. Titel) in einem gemeinsamen Feld “Name” zu führen. Ob das System hier technisch differenziert, ist unklar. Falls einzelne Systeme den angezeigten Namen nur als ganzes speichern, sollte dies für den Standard übernommen werden, da es für die meisten Anwendungen ausreichen sollte.
- Das System PROVOX unterscheidet zwischen privaten und geschäftlichen Anschriften.

### 5.4.2 Beziehungen

- Objekte vom Typ “Person” können einer Organisation, z.B. einer Fraktion, zugeordnet werden. Diese Beziehung ist datiert.
- Objekte vom Typ “Person” können einem oder mehreren Gremien zugewiesen werden, um die Mitgliedschaft in diesem Gremium darzustellen. Diese Beziehungen sind ebenfalls datiert.

### 5.4.3 Beispiel

```

{
  "id": "1000",
  "first_name": "Max",

```

```

    "last_name": "Mustermann",
    "academic_title": "Dr.",
    "sex": "M",
    "profession": "Rechtsanwalt",
    "email": "max@mustermann.de",
    "phone": "+4977777",
    "fax": "+4988888",
    "address": "Musterstraße 5, 11111 Musterort",
    "last_modified": "2012-08-16T14:05:27+02:00",
    "organisations": [
      {
        "id": "2000",
        "start": "2011-03-01",
        "end": "2013-02-28"
      },
      {
        "id": "2001",
        "start": "2013-03-01"
      }
    ],
    "committees": [
      {
        "id": "7",
        "start": "2013-01-01"
      }
    ]
  }
}

```

## 5.5 Organisation (organisation)

Organisationen sind üblicherweise Parteien bzw. Fraktionen, denen die Personen angehören können.

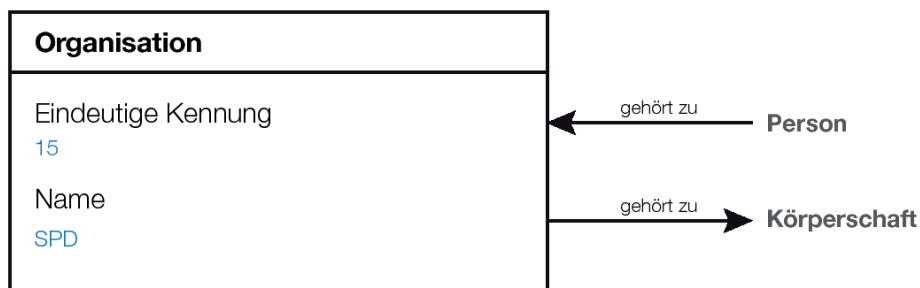


Figure 5: Objekttyp Organisation

### 5.5.1 Eigenschaften

**Schlüssel (id)** Zur eindeutigen Kennzeichnung einer Organisation innerhalb des Systems

**Name (name)** Der gebräuchliche Name der Organisation, z.B. “SPD” oder “DIE LINKE”.

**Zuletzt geändert (last\_modified)** Datum und Uhrzeit der letzten Änderung

### Anmerkungen

- Unklar ist bislang, ob Organisationen in der Praxis eher Fraktionen (“SPD-Fraktion im Kölner Rat”, “SPD-Fraktion in Köln-Innenstadt”) abbilden oder ob eher Ortsverbände von Parteien (“SPD Köln”) gemeint sein werden. Einblicke, wie gängige Systeme dies handhaben, sollten evtl. gesammelt und berücksichtigt werden.

### 5.5.2 Beziehungen

- Jede Organisationen gehört zu einer Körperschaft.
- Personen können Organisationen angehören (*datiert*).

### 5.5.3 Beispiel

```
{
  "id": "15",
  "name": "SPD",
  "body": "1",
  "last_modified": "2012-08-16T14:05:27+02:00"
}
```

## 5.6 Sitzung (meeting)

Eine Sitzung ist die Versammlung der Mitglieder eines Gremiums oder mehrerer Gremien zu einem bestimmten Zeitpunkt an einem bestimmten Ort.

Die geladenen Teilnehmer der Sitzung sind jeweils als „Person“ in entsprechender Form referenziert. Verschiedene Dokumente (Einladung, Ergebnis- und Wortprotokoll, sonstige Anlagen) können referenziert werden.

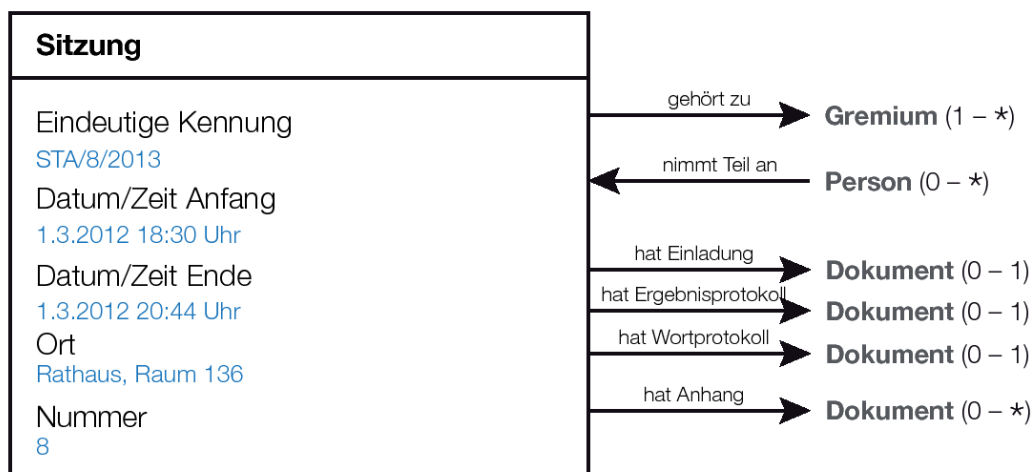


Figure 6: Objekttyp Sitzung

### 5.6.1 Eigenschaften

**Schlüssel (id)** Zur eindeutigen Identifizierung der Sitzung innerhalb des Systems. In der Praxis wird ein solcher Schlüssel entweder durch eine numerische ID gebildet oder durch Kombination mehrerer Merkmale wie dem Kürzel des Gremiums, der laufenden Nummer der Sitzung in einem Jahr und der Jahreszahl (z.B. “BV1/0034/2012”).

**Nummer (sequence\_number)** *Optional*. Laufende Nummer der Sitzung, üblicherweise innerhalb der Wahlperiode mit 1 beginnend. In der Praxis wird dadurch z.B. die "2. Sitzung des Rats" gekennzeichnet. Ist dieses Feld gesetzt, MUSS ein numerischer Wert enthalten sein.

**Anfang (start)** Datum und ggf. Uhrzeit des Anfangszeitpunkts der Sitzung

**Ende (end)** *Optional*. Datum und Uhrzeit vom Ende der Sitzung

**Ort (address)** *Optional*. Textliche Information zum Ort der Sitzung, z.B. "Rathaus, Raum 136".

**Zuletzt geändert (last\_modified)** Datum und Uhrzeit der letzten Änderung

### 5.6.2 Beziehungen

- Sitzungen sind mindestens einem Gremium zugeordnet
- Einer Sitzung sind Personen zugeordnet, um die Teilnahme an der Sitzung auszudrücken.
- Dokumente können vom Typ "Sitzung" *optional* zu mehreren Zwecken referenziert werden:
  - Zum Verweis auf die Einladung zur Sitzung
  - Zum Verweis auf das Ergebnisprotokoll zur Sitzung
  - Zum Verweis auf das Wortprotokoll zur Sitzung
- Weiterhin können Sitzungen beliebige weitere Dokumente, die keine eigenständigen Drucksachen sind, referenzieren. Dabei handelt es sich dann um nicht weiter spezifizierte Anlagen.

### 5.6.3 Beispiel

```
{
  "id": "3271",
  "identifier": "STA/0034/2012",
  "start": "2013-01-04T08:00:00+01:00",
  "end": "2013-01-04T12:00:00+01:00",
  "address": "Rathaus, Raum 136",
  "sequence_number": 1,
  "committees": ["STA"],
  "people": ["1000", "1001"],
  "invitation": "0001/2013",
  "result_minutes": "0002/2013",
  "verbatim_minutes": "0003/2013",
  "attachments": [
    "0004/2013",
    "0005/2013"
  ],
  "last_modified": "2012-01-08T14:05:27+01:00"
}
```

## 5.7 Tagesordnungspunkt (agendaitem)

Der Tagesordnungspunkt wird für eine bestimmte Sitzung angelegt, erhält eine (innerhalb dieser Sitzung eindeutige) Nummer und einen Titel (Betreff). Nach der Sitzung wird dem Tagesordnungspunkt außerdem ein Ergebnis angehängt. Unter Umständen kann dem Tagesordnungspunkt ein bestimmter Beschlusstext beigelegt sein.

Überlicherweise haben Sitzungen mehrere Tagesordnungspunkte.

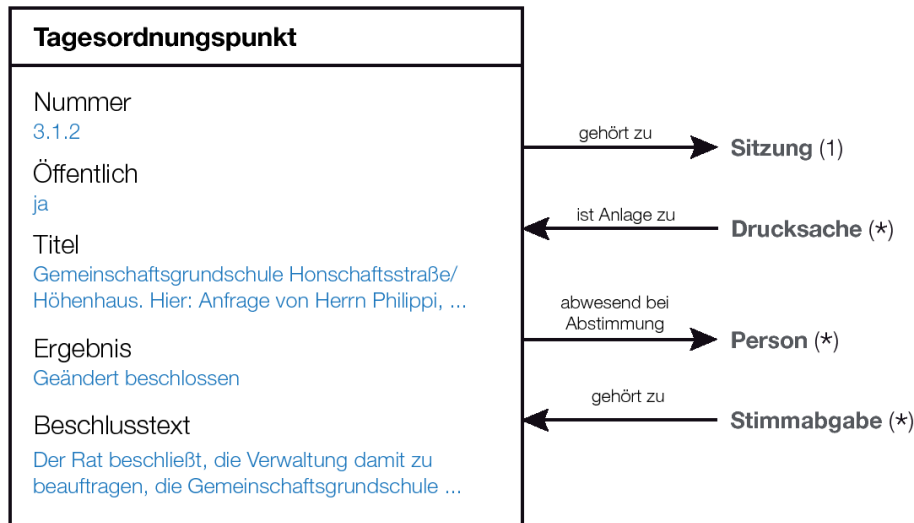


Figure 7: Objekttyp Tagesordnungspunkt

### 5.7.1 Eigenschaften

**Nummer (identifier)** Beispiel: “1.2.3”. Diese Nummer gibt an, in welcher Reihenfolge die Tagesordnungspunkte einer Sitzung normalerweise behandelt werden. Im Kontext einer Sitzung ist diese Nummer eindeutig.

**Öffentlich (public)** Kennzeichnet, ob der Tagesordnungspunkt in öffentlicher Sitzung behandelt wird. Kann die Werte **true** (öffentlich) oder **false** annehmen.

**Titel (title)** Das Thema des Tagesordnungspunktes

**Ergebnis (result)** *Optional*. Kategorische Information darüber, welches Ergebnis die Beratung des Tagesordnungspunktes gebracht hat. In der Praxis sind hier Kategorien wie “Unverändert beschlossen”, “Geändert beschlossen”, “Endgültig abgelehnt”, “Zur Kenntnis genommen”, “Ohne Votum in nachfolgende Gremien überwiesen” und weitere zu erwarten.

**Ergebnis Details (result\_details)** *Optional*. Ermöglicht die Angabe zusätzlicher Textinformationen zum Ergebnis, zum Beispiel im Fall der Verweisung an ein anderes Gremium die Angabe, an welches Gremium verwiesen wurde.

**Beschlusstext (resolution\_text)** *Optional*. Falls in diesem Tagesordnungspunkt ein Beschluss gefasst wurde, kann der Text hier hinterlegt werden. Das ist besonders dann in der Praxis relevant, wenn der gefasste Beschluss (z.B. durch Änderungsantrag) von der Beschlussvorlage abweicht.

**Zuletzt geändert (last\_modified)** Datum und Uhrzeit der letzten Änderung

#### Anmerkungen

- Einige Systeme vergeben zu Tagesordnungspunkten intern unveränderliche, numerische IDs. Es ist unklar, ob es zusätzlichen Nutzen bringt, derartige IDs, neben den Nummern, in den Standard zu übernehmen. Dies würde vermutlich nur Sinn ergeben, wenn es als Pflichtfeld gelten könnte.
- Teil der Beratungen über einheitliche Nomenklatur im Standard sollte sein, eine Vereinheitlichung der Werte für die Eigenschaft **result** zu diskutieren.

### 5.7.2 Beziehungen

- Jeder Tagesordnungspunkt gehört zu genau einer Sitzung.
- Der Tagesordnungspunkt kann auf eine Drucksache verweisen, die im Rahmen dieses Tagesordnungspunkt beraten werden soll.
- Es können Personen referenziert werden, die während der Abstimmung zu diesem Tagesordnungspunkt *nicht* anwesend waren.

### 5.7.3 Beispiel

```
{  
  "meeting": "3271",  
  "identifier": "3.1.2",  
  "public": true,  
  "title": "Gemeinschaftsgrundschule Hornschaftsstraße/Höhenhaus. Hier: Anfrage von Herrn Phi  
  "result": "Geändert beschlossen",  
  "resolution_text": "Der Beschluss weicht wie folgt vom Antrag ab: ...",  
  "people_absent": ["1002", "1003"],  
  "last_modified": "2012-08-16T14:05:27+02:00"  
}
```

## 5.8 Drucksache (paper)

Eine Drucksache bildet Mitteilungen, Antworten auf Anfragen, Beschlussvorlagen, Anfragen, Anträge und weitere Vorlagen ab. Jede Drucksache erhält eine eindeutige Kennung.

Die Drucksache hat im Informationsmodell eine hervorgehobene Bedeutung. Im Fall eines Antrags kann mit einer einzigen Drucksache ein über Monate oder Jahre dauernder politischer Entscheidungsprozess verbunden sein. In dem Zusammenhang entstehen üblicherweise weitere Drucksachen.

Drucksachen spielen in der schriftlichen wie mündlichen Kommunikation eine besondere Rolle, da in vielen Texten auf bestimmte Drucksachen Bezug genommen wird. Hierbei kommen in Ratsinformationssystemen unveränderliche Kennungen der Drucksachen zum Einsatz.

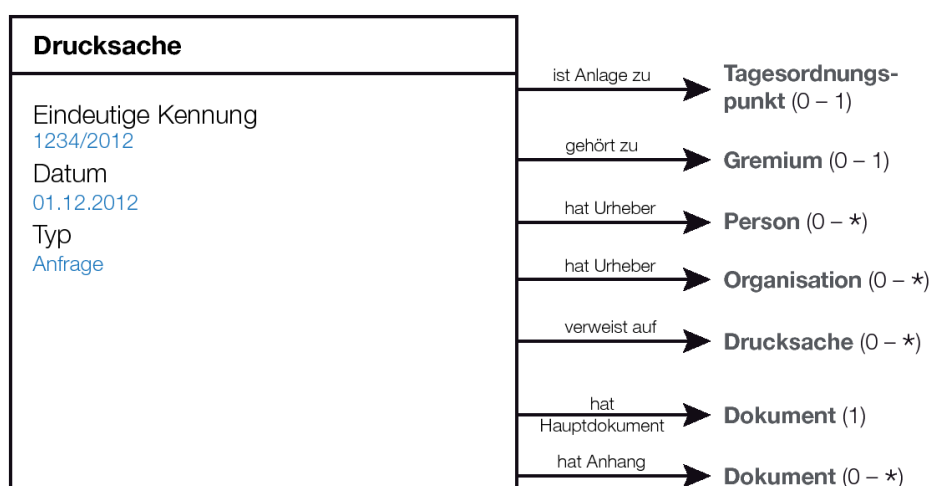


Figure 8: Objekttyp Drucksache

Jede Drucksache ist über die Eigenschaft “Typ” als eine der folgenden Arten von Drucksachen gekennzeichnet:

- **Beschlussvorlage:** Entscheidungsvorschlag der Verwaltung
- **Antrag:** Entscheidungsvorschlag einer Fraktionen bzw. mehrerer Fraktionen oder einer/mehrerer Einzelperson/en
- **Anfrage:** Frage(n) einer oder mehrerer Fraktion oder Einzelpersonen an die Verwaltung
- **Mitteilung/Stellungnahme der Verwaltung:** Eine Information der Verwaltung an einzelne oder mehrere Gremien. Darunter fallen nicht Beantwortungen von Anfragen.
- **Beantwortung einer Anfrage:** Antwort der Verwaltung auf (mündliche oder schriftliche) Anfragen

### 5.8.1 Eigenschaften

**Schlüssel (id)** Die Kennung einer Drucksache muss für die jeweilige Körperschaft eindeutig sein. Sie kann sowohl Ziffern als auch Buchstaben enthalten. Einige Systeme (z.B. Köln) verwenden besondere Trennzeichen wie “/”, um eine Jahreszahl von einer laufenden Nummer abzutrennen. Weiterhin werden mancherorts führende Nullen verwendet.

**Datum (date)** Datum der Veröffentlichung

**Typ (type)** Art der Drucksache (Erläuterung siehe oben)

**Zuletzt geändert (last\_modified)** Datum und Uhrzeit der letzten Änderung

### 5.8.2 Beziehungen

- Es muss genau ein **Hauptdokument** (Objekttyp “Dokument”) referenziert werden.
- Es können beliebig viele weitere Dokumente referenziert werden, die als nachgeordnete **Anlagen** zur Drucksache verstanden werden.
- Die Drucksache ist beliebig vielen Gremien zuzuordnen, in denen diese beraten wird.
- Drucksachen können **Urhebern** zugewiesen werden. Im Fall von Mitteilungen der Verwaltung ist dies oft der Oberbürgermeister. Bei Anträgen oder Anfragen können Organisationen oder Einzelpersonen referenziert werden. Es können stets mehrere Urheber verknüpft werden.
- Es können beliebig viele **Orte** (siehe Objekttyp “Ort”) referenziert werden, die im Inhalt der Drucksache behandelt werden. Beispiel: Beschlussvorlage zur Freigabe von Mitteln für die Sanierung eines Sportplatzes, wobei der Ort die Lage des Sportplatzes genau beschreibt.
- Drucksachen können auf andere Drucksachen referenzieren. Diese Verweise können verschiedene semantische Beziehungen ausdrücken. So kann eine Drucksache auf eine übergeordnete oder eine oder mehrere untergeordnete Drucksachen verweisen. Beim Drucksachen-Typ “Beantwortung einer Anfrage” ist die Drucksache zu referenzieren, die die ursprüngliche **Anfrage** beinhaltet. Denkbar sind auch Verweise auf frühere Drucksachen zum selben Thema. Zu klären ist, wie die verschiedenen möglichen Beziehungen formell ausgedrückt werden.
- Drucksachen können zu beliebig vielen Tagesordnungspunkten in Beziehung stehen, um die **Beratungsfolge** einer Drucksache abzubilden. Hierbei kann die Beziehung jeweils mit einer Zuständigkeit versehen sein, die noch näher zu bestimmen ist (TODO).



### 5.8.3 Beispiel

```
{
  "id": "1234/2012",
  "date": "2013-01-04",
  "type": "Beantwortung einer Anfrage",
  "related_papers": [
    "0768/2012"
  ],
  "main_document": "3000.pdf",
  "attachments": [
    "3002.pdf",
    "3003.pdf"
  ],
  "locations": [
    {
      "description": "Theodor-Heuss-Ring 1",
      "lat": 7.148,
      "lon": 50.023
    }
  ],
  "committees": ["STA"],
  "creators": [
    {
      "typ": "Organisation",
      "id": "2000"
    },
    {
      "typ": "Person",
      "id": "1000"
    }
  ],
  "consultations": [
    {
      "meeting": "3271",
      "agendaitem": "3.1.2",
      "role": "Federführende Beratung"
    }
  ],
  "last_modified": "2013-01-08T12:05:27+01:00"
}
```

## 5.9 Dokument (document)

Ein Dokument hält die Metadaten einer Datei vor, beispielsweise einer PDF-Datei, eines RTF- oder Word-Dokuments.

Wird von einem Word-Dokument eine PDF-Ableitung hinterlegt, ist diese Ableitung ebenfalls ein Dokument. Um zu zeigen, dass es sich um eine Ableitung handelt, verweist dieses auf das Original als “Master”.

Im Unterschied zur Drucksache benötigt das Dokument keine nutzerfreundliche Kennung.

### 5.9.1 Eigenschaften

**Schlüssel (id)** Unveränderliche Kennung

**Name (name)** Dateiname, z.B. “12345.pdf”

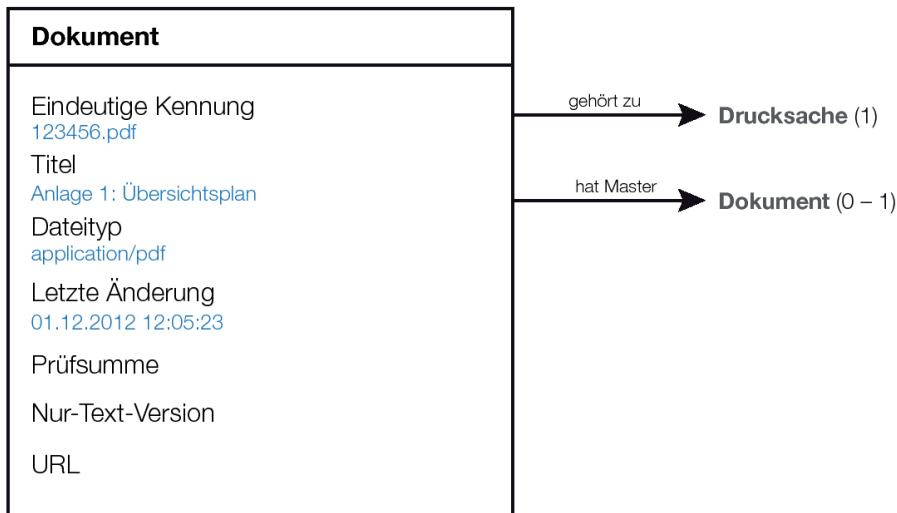


Figure 9: Objekttyp Dokument

**Dateityp (mime\_type)** Mime-Typ des Inhalts, z.B. “application/pdf”

**Veröffentlichungsdatum (date)** Datum des Tages, an dem das Dokument ins System eingestellt wurde

**Änderungsdatum und -uhrzeit (last\_modified)** Datum und Uhrzeit der letzten Änderung des Dokuments

**Prüfsumme (sha1\_checksum)** SHA1-Prüfsumme des Dokumenteninhalts

**URL (url)** URL zum Abruf der Daten dieses Dokuments mittels HTTP GET-Aufruf

**Nur-Text-Version (text)** Reine Text-Wiedergabe des Dokumenteninhalts, sofern es sich nicht um eine reine Abbildung handelt.

### 5.9.2 Beziehungen

- Dokumente gehören zwingend zu einer **Drucksache**, optional auch zu mehreren. Ein Dokument kann entweder als Hauptdokument einer Drucksache oder als Anlage eingestuft sein.
- Ein Dokument kann auf ein anderes Dokument referenzieren, wenn es von dem anderen Dokument abstammt. So ist es möglich, von einem abgeleiteten Dokument zu seinem Dokumenten-Master zu gelangen (Beispiel: von einem PDF-Dokument zum OpenOffice-Original).

```

{
  "id": "3000",
  "name": "3000.pdf",
  "mime_type": "application/pdf",
  "date": "2013-01-04T07:54:13+01:00",
  "last_modified": "2013-01-04T07:54:13+01:00",
  "sha1_checksum": "da39a3ee5e6b4b0d3255bfef95601890afd80709",
  "url": "http://ris.beispielstadt.de/api/documents/3000.pdf",
  "text": "Der Übersichtsplan zeigt alle Ebenen des ...",
  "master": "2099"
}
  
```

## 5.10 Ort (location)

Dieser Objekttyp dient dazu, einen Ortsbezug einer Drucksache formal abzubilden. Ortangaben können sowohl aus Textinformationen bestehen (beispielsweise der Name einer Straße/eines Platzes oder eine genaue Adresse) als auch aus Geodaten.

OParl sieht die Angabe von Geodaten in Anlehnung an die GeoJSON-Spezifikation [13] vor. Die GeoJSON-Spezifikation erlaubt die Abbildung von vielen unterschiedlichen Geometrien wie Punkten, Pfaden und Polygonen. Während GeoJSON zu jedem Geodaten-Objekt auch die Speicherung zusätzlicher Metadaten ermöglicht, beschränkt sich OParl lediglich auf das **geometry**-Attribut in GeoJSON. Sämtliche Geo-Koordinatenangaben werden in OParl im WGS-84-System [11] erwartet.

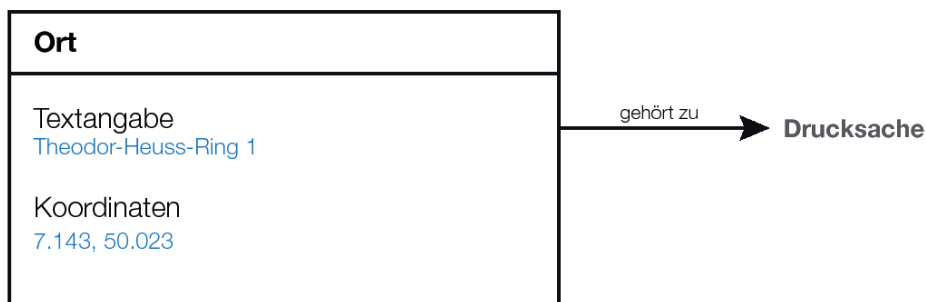


Figure 10: Objekttyp Ort

### 5.10.1 Eigenschaften

**Textangabe (description)** *Optional*. Textliche Beschreibung eines Orts, z.B. in Form einer Adresse

**Koordinaten (geometry)** *Optional*. GeoJSON geometry Objekt

**Zuletzt geändert (last\_modified)** Datum und Uhrzeit der letzten Änderung

### 5.10.2 Beziehungen

- Orte können mit Drucksachen in Verbindung stehen.

```
{
  "description": "Honschaftsstraße 312, 51061 Köln",
  "geometry": {
    "type": "Point",
    "coordinates": [7.03291, 50.98249]
  },
  "last_modified": "2013-02-14T14:05:27+01:00"
}
```

## 6 Fußnoten

[1]: Siehe [de.wikipedia.org/wiki/Amtlicher\\_Gemeindeschlüssel](http://de.wikipedia.org/wiki/Amtlicher_Gemeindeschlüssel)

[2]: Ratsinformationssystem der Stadt Köln, <http://ratsinformation.stadt-koeln.de/>

[3]: Firma Somacos, [SessionNet Produktinformation](#)

- [4]: Ratsinformationssystem der Bezirksverwaltung Berlin Mitte, <http://www.berlin.de/...>
- [5]: CC e-gov GmbH, [ALLRIS Produktionformationen](#)
- [6]: Ratsinformationssystem der Stadt Rösrath, <http://212.227.97.55/...>
- [7]: [Firma PROVOX](#)
- [8]: Ratsinformationssystem der Stadt Euskirchen, <https://sitzungsdienst.euskirchen.de/>
- [9]: Firma Sternberg, [SD.NET RIM Produktionformationen](#)
- [10]: BoRis, Ratsinformationssystem der Stadt Bonn (Eigenentwicklung). <http://www2.bonn.de/...>
- [11]: World Geodetic System 1984 (EPSG:4326), wird unter anderem auch vom Global Positioning System (GPS) verwendet.
- [12]: Gemeinsame Normdatei: [de.wikipedia.org/wiki/Gemeinsame\\_Normdatei](http://de.wikipedia.org/wiki/Gemeinsame_Normdatei)
- [13]: GeoJSON [www.geojson.org](http://www.geojson.org)
- [14]: Frankfurt Gestalten [www.geojson.org](http://www.geojson.org)
- [15]: Offenes Köln [offeneskoeln.de](http://offeneskoeln.de)
- [16]: OpenRuhr:RIS [openruhr.de/openruhrris](http://openruhr.de/openruhrris)

## 7 Glossar

**JSON-LD** JSON for Linked Data

**RIS** Ratsinformationssystem

**WGS 84** World Geodetic System 1984. Ein weltweites Referenzsystem für die Interpretation von Geokoordinaten-Angaben.