



Abschlussbericht Testbed "Persistent Identifiers"

Autor:
Martina Trognitz

Version 1.0
29. Januar 2014

Koordination



Förderung

DFG

Autor: Martina Trognitz

Titel: Abschlussbericht Testbed "Persistent Identifiers"

Sprache: Deutsch

URL: <<http://www.ianus-fdz.de/projects/ergebnisse/wiki>>

Zitierhinweis:

Martina Trognitz (2013) Abschlussbericht Testbed "Persistent Identifiers". [Version 1.0] Hrsg. IANUS.

Online unter: <<http://www.ianus-fdz.de/projects/ergebnisse/wiki>>

Kontakt:

ianus-fdz@dainst.de

www.ianus-fdz.de

Lizenz:



Dieses Werk bzw. Inhalt ist lizenziert unter einer Creative Commons – Namensnennung – nicht-kommerziell – Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz.

<http://creativecommons.org/licenses/by-nc-sa/3.0/de/>

Version	Autoren	Datum	Beschreibung
1.0	Martina Trognitz	29.01.2014	Erste vollständige Fassung

Inhaltsverzeichnis

1	Spezifikation des Testbeds	4
1.1	Zu adressierende Fragen	4
2	Was ist ein PID?	5
3	Wie funktioniert ein PID?	5
4	Worauf wird ein PID angewendet?	6
4.1	Dokumente	7
4.2	Konzepte	7
4.3	Datensammlung	7
5	Eigenschaften von PIDs	7
5.1	Auflösbarkeit	8
5.2	Flexibilität	9
5.3	Metadaten	9
5.4	Opazität und Semantik	10
5.5	Interoperabilität	10
5.6	Granularität	10
5.7	Versionierung	10
5.8	Zuverlässigkeit und Vertrauenswürdigkeit	11
5.9	Autorität	11
5.10	Kosten	11
6	Vorhandene PID-Systeme	11
6.1	Übersicht der PID-Systeme	12
6.1.1	URI	12
6.1.2	URL	14
6.1.3	URN	15
6.1.4	PURL	17
6.1.5	NBN	18
6.1.6	Handle (HS)	20
6.1.7	DOI	22
6.1.8	ARK	24
6.1.9	XRI	26
6.1.10	OpenURL	27
6.1.11	Zusammenfassung	28
6.2	Wo wird welches PID-System verwendet?	28
6.2.1	DANS - EDNA	29
6.2.2	Archaeology Data Service	30
6.2.3	GESIS	30
6.2.4	MPI - The Language Archive	31
6.2.5	PANGAEA	31
6.2.6	tDAR	31
6.2.7	OpenContext	32
6.2.8	Digitales Archiv NRW	32
6.3	Eingrenzung der Auswahl für IANUS	32

7	Relevante Einrichtungen und Dienste	32
7.1	Für URN:NBN	32
7.1.1	Deutsche Nationalbibliothek	33
7.2	Für Handle (HS)	33
7.2.1	Global Handle Registry	33
7.2.2	EPIC	33
7.2.3	DARIAH-DE	34
7.3	Für DOI	34
7.3.1	DataCite	34
7.3.2	da ra	34
7.3.3	Citation Formatter	35
7.3.4	Content Negotiation	35
7.4	Dienste für ARK	35
7.4.1	California Digital Library	35
7.4.2	EZID	35
7.5	Weitere Dienste	36
7.5.1	N2T	36
7.5.2	Browser Plugins	36
8	Empfehlungen für IANUS	37
8.1	Welches PID-System für IANUS geeignet ist	37
8.1.1	Das Suffix	37
8.2	Praktische Anwendungshinweise	38
9	Weitere Quellen	40
10	Abkürzungsverzeichnis	41
A	Granularität	42
B	Landing Pages	47
C	PID Service DARIAH-DE - Handle System	48
D	Testumgebung von da ra - DOIs	53
E	Testumgebung von EZID - ARKs	57

1 Spezifikation des Testbeds

Dieses Testbed behandelt die Vergabe von permanenten Identifikatoren (PIDs), um archivierte Dateien und Dokumente nachhaltig zitieren zu können.

Ziel dieses Testbeds ist es Informationen zu den aktuell verfügbaren Systemen für PIDs zu sammeln. Die Informationen werden so aufbereitet, dass schnell ersichtlich wird, worin die Vor- und Nachteile eines bestimmten Systems liegen. Dabei wird besonders Rücksicht auf die Fragestellungen und Anforderungen von IANUS genommen, da schlussendlich ein geeignetes Schema zur Vergabe von PIDs für IANUS gefunden werden soll.

Auf die Funktionsweise von PIDs wird eingegangen und allgemeine Fragen, aber auch IANUS-spezifische Fragen werden behandelt. Die Gedankenanstöße und Hinweise aus dem zweiten Arbeitsgruppentreffen in dem *Arbeitspaket 2 – Interoperabilität* am 16.10.2012 in Berlin werden in dieses Testbed einbezogen.

Wichtig für IANUS ist außerdem die Frage zur technischen und praktischen Umsetzung. Dabei muss geklärt werden, ob ein bestehender Dienst in Anspruch genommen werden kann oder IANUS selbst zum Anbieter eines Dienstes werden sollte.

1.1 Zu adressierende Fragen

- Was ist ein PID?
- Wie funktioniert ein PID?
- Worauf wird ein PID angewendet?
- Welche Eigenschaften hat ein PID?
- Welche PID-Systeme gibt es?
- Wo wird welches PID-System verwendet?

Für IANUS von Interesse sind spezielle Fragen zu der Funktionsweise und dem Verhalten von PIDs:

- Welche Voraussetzungen müssen für die Vergabe einer PID erfüllt sein?
- Wann wird ein PID vergeben?
- Was passiert, wenn ein Objekt schon einen PID hat?
- Wie granular sollen PIDs vergeben werden?
- Wie soll mit hierarchischen Strukturen umgegangen werden?
- Sollen die PIDs eine semantische Struktur aufweisen oder durch Opazität jegliche semantische Information verbergen?
- Wie soll mit verschiedenen Versionen eines Objektes umgegangen werden? In welchen Fällen tritt Versionierung auf?
- Soll das PID-System Prüfsummen (*checksums*) und Prüfwerte verwenden?

- Welche Metadaten und welches Metadatenschema sollen zusammen mit dem PID-System verwendet werden?

Zusätzlich müssen für eine technische Umsetzung in IANUS weitere Fragen berücksichtigt werden:

- Welche Dienste gibt es?
- Welche Registrierungsagenturen und Anbieter gibt es?
- Welche Kosten entstehen?

Die Beantwortung dieser Fragen, dient der Formulierung von Empfehlungen, welche eine Antwort auf die zentralen Fragestellungen „*Welches PID-System soll IANUS verwenden?*“ und „*Wie soll die Anwendung des PID-Systems praktisch umgesetzt werden?*“ geben.

2 Was ist ein PID?

PID steht für *Persistent Identifier*, also persistenter Identifikator. Ein Identifikator ist eine einzigartige Zeichenkette, die einem Etwas zugeordnet werden kann, um dieses Etwas eindeutig zu identifizieren oder zu referenzieren. Eine Signatur ist beispielsweise ein Identifikator für Bücher in Bibliotheken.

Ein persistenter Identifikator ist ein Identifikator, der dauerhaft zugewiesen wird und für immer mit dem Zugewiesenen Etwas zusammenhängen wird. Kein anderes Etwas wird diesen Identifikator bekommen. Beispielsweise ist die ISBN-Nummer ein persistenter Identifikator für Bücher.

Das Etwas, im folgenden als Objekt bezeichnet, „kann alles sein worüber gesprochen werden kann, insbesondere alles, was unterschieden und mit einem Identifier versehen werden kann. Es können also z.B. statische oder dynamische Objekte, Dokumente oder Prozeduren und auch aggregierte Objekte oder Teilobjekte identifiziert werden.“¹

3 Wie funktioniert ein PID?

Die Funktionsweise eines PIDs ähnelt der einer Signatur. Ähnlich wie ein bestimmtes Buch in einer Bibliothek eine eindeutige Signatur zugewiesen bekommt, bekommt auch ein digitales Objekt in einem Dateisystem einen PID zugewiesen.

Mittels dem PID kann dieses Objekt später referenziert und somit zitiert werden. Möchte ein Nutzer auf das zitierte Objekt zugreifen, so kann er dies mittels des angegebenen PIDs tun. Der Nutzer gibt den PID bei einem geeigneten PID-Resolver ein, der den PID auflöst und dem Nutzer die Adresse des Speicherorts mitteilt oder ihn gleich an den Speicherort weiterleitet. Von dem Speicherort erhält der Nutzer die Repräsentation des gefragten Objektes. Dieser Vorgang wird in Abbildung 1 veranschaulicht.

Den Resolver kann man sich im Prinzip wie eine große Tabelle vorstellen, in der jedem vergebenen PID eine gültige Adresse zugeordnet ist. Weiterhin können in dieser Tabelle noch zusätzliche Metadaten zu dem PID abgelegt sein. Durch

¹Nestor (2009), S. 3

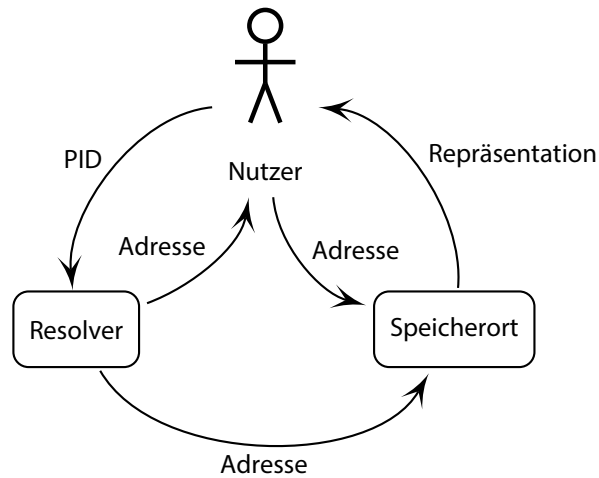


Abbildung 1: Funktionsweise der Auflösung eines PIDs.

diesen Mechanismus wird der eigentliche Speicherort von dem Objekt getrennt, womit einer der häufigsten Ursachen bei kaputten URLs (sogenannte *broken Links*) vorgebeugt wird. Denn oft geht nicht das gespeicherte Objekt verloren, sondern die URL-Adresse ändert sich, durch Umbenennungen oder Umstrukturierungen, wodurch eine bereits in einer Referenz angegebene URL-Adresse ungültig wird.

Durch die Verwendung eines PID-Systems kann jede Änderung des Speicherortes durch die Institution, die die digitalen Objekte bereit stellt, eingetragen werden. Der PID ändert sich dadurch nicht. Lediglich der Verweis auf den Ort, wo das zugehörige Objekt zu finden ist, wird ein anderer. Theoretisch könnte man auch mehrere Speicherorte eintragen, wenn das Objekt als Kopie an mehreren Orten abgelegt wurde.

Der Nutzer, der PIDs verwendet und nachschlägt, bemerkt nichts von dem Vorgang, sondern bekommt üblicherweise die Repräsentation des digitalen Objektes angezeigt. Wie die Repräsentation aussieht, hängt von dem Objekt ab. Es kann sich beispielsweise um eine Tabelle, ein Bild oder auch eine Webseite mit mehreren Dateianhängen handeln. Außerdem können, je nach Zugangsrechten, bestimmte Informationen ausgeblendet sein.

Vergleichen lässt sich die Auflösung von PIDs auch mit dem Auflösen von Literaturangaben. Diese können in einem Bibliothekskatalog nachgeschlagen werden, wo anschließend der Standort des physischen Exemplars angegeben wird.

4 Worauf wird ein PID angewendet?

Prinzipiell kann ein PID auf jedes beliebige Objekt, egal ob es nur digital oder auch real existiert, angewendet werden. Üblicherweise wird es sich jedoch um digitale Ressourcen handeln, zumal physische Ressourcen, wie gedruckte Publikationen oder museale Objekte, im Normalfall eine ISBN- oder eine Inventarnummer bekommen.

Digitale Ressourcen können grob in zwei Kategorien unterteilt werden: Dokumente und Konzepte. Zusammenhängende Dokumente oder Konzepte bilden eine Datensammlung.

4.1 Dokumente

Ein Dokument bildet eine in sich geschlossene Einheit, die aus Text, Tabellen, Bildern oder deren verschiedenen Kombinationen miteinander bestehen kann. Ein Dokument stellt üblicherweise auch eine eigene Datei dar. Beispiele hierfür sind: ein elektronischer Zeitschriftenartikel, ein Foto oder eine Präsentation.

An einem Dokument wird meist nur in einem bestimmten zeitlichen Rahmen und für einen bestimmten Zweck gearbeitet, weshalb dieses auch irgendwann als abgeschlossen betrachtet werden kann und nicht damit gerechnet werden muss, dass häufige Änderungen daran vorgenommen werden.

Beispielsweise wird an einem zur Veröffentlichung vorbereiteten elektronischen Artikel nichts mehr verändert werden. Auch bereits vorgeführte Präsentationen können als abgeschlossen gelten. Wird eine gegebene Präsentation für eine andere Vorführung wiederverwendet und angepasst, so handelt es sich nicht mehr um das ursprüngliche Dokument, sondern um eine neue Version, welche als ein eigenes neues Dokument betrachtet werden muss. Ein weiteres Beispiel stellen Fotos dar, die in einer neuen Version bearbeitet wurden.

Wird einem Dokument ein PID zugewiesen, so muss eine neue Version dieses Dokumentes auch einen eigenen neuen Identifikator bekommen. Hinweise auf andere Versionen können in den Metadaten untergebracht werden.

4.2 Konzepte

Ein Konzept unterscheidet sich von einem Dokument dadurch, dass der Inhalt regelmäßigen Änderungen ausgesetzt ist. Beispielsweise handelt es sich bei einer Webseite zu dem Thema „*Aktuelle Nachrichten aus den Altertumswissenschaften*“ um ein Konzept. Ein Besucher dieser Seite erwartet dort täglich neue Nachrichten zu den Altertumswissenschaften. Möchte der Besucher einem Dritten über diese Seite berichten, so kann er den PID dieses Konzeptes zur Weitergabe verwenden.

Wenn er aber auf eine einzelne Nachrichtenmeldung dieser Seite verweisen möchte, so wäre dies möglich, wenn diese jeweils als Dokument betrachtet werden und einen eigenen Identifikator bekommen, der anschließend zur Referenzierung verwendet werden kann.

4.3 Datensammlung

Eine thematisch zusammenhängende Sammlung von Dokumenten oder Konzepten wird als Datensammlung bezeichnet. Auch eine Datensammlung kann einen eigenen PID zugewiesen bekommen. Die Bestandteile einer Datensammlung können, müssen aber nicht, eigene Identifikatoren besitzen.

5 Eigenschaften von PIDs

Welche Funktionalitäten und Eigenschaften ein persistenter Identifikator aufweist, hängt davon ab, welches System verwendet wird, wie die Spezifikationen des Systems technisch umgesetzt werden und welche zusätzlichen Eigenschaften implementiert werden.

Es gibt jedoch zwei wichtige Eigenschaften, die sich aus der Definition ergeben:

- Globale Eindeutigkeit
- Unbegrenzte Existenzzeit

Ein PID, der nicht global eindeutig und einzigartig ist, kann nicht eindeutig einem Objekt zugeordnet werden und wäre somit ambig. Global meint in diesem Zusammenhang einen bestimmten Kontext, der durch das verwendete PID-System und die Registrierungsagentur definiert wird. Diese Eigenschaft ist bei allen hier berücksichtigten PID-Systemen gegeben

Die Lebensspanne eines persistenten Identifikators ist idealerweise unbegrenzt, was impliziert, dass der Identifikator weder gelöscht noch einem anderen Objekt zugewiesen werden kann. Ein persistenter Identifikator kann das zu identifizierende Objekt bei Weitem überleben.

Die Persistenz hängt vor allem von der Registrierungsagentur und deren Verpflichtungen (*commitment*) und *nicht* unbedingt von dem verwendeten PID-System ab. Man könnte auch von verschiedenen Persistenzstufen sprechen, da in manchen Fachbereichen beispielsweise schon 10 Jahre Bestandszeit ausreichen, während gerade in den Altertumswissenschaften eine deutlich längere Lebensdauer erwartet wird.

Weitere unterschiedlich stark ausgeprägte Eigenschaften und Stichworte, die im Zusammenhang von PIDs berücksichtigt werden, sind:

- Auflösbarkeit
- Flexibilität
- Metadaten
- Opazität und Semantik
- Interoperabilität
- Granularität
- Versionierung
- Zuverlässigkeit und Vertrauenswürdigkeit
- Autorität
- Kosten

Insbesondere ist zu betonen, dass ein PID eines Systems nicht unbedingt die gleichen Eigenschaften wie ein PID desselben Systems eines anderen Anbieters oder Datengebers besitzt. Dies hängt damit zusammen, dass verschiedene technische Aspekte unterschiedlich gelöst werden oder auch gar nicht implementiert werden.

5.1 Auflösbarkeit

Mit Auflösbarkeit (*resolvability*) ist das Verknüpfen eines PIDs mit dem Speicherort des Objektes gemeint. Allein die Identifizierung eines Objektes reicht nicht aus, wenn nicht angegeben werden kann, wo sich dieses befindet und wie man es erreicht, um darauf zuzugreifen.

Wie die Auflösung realisiert wird, hängt von dem PID-System ab. Aktuell lösen aber alle PID-Systeme auf eine URL-Adresse auf, die auf den Speicherort des Objektes zeigt. Dabei obliegt es den Datenanbietern die hinterlegten URLs in dem PID-System bei Bedarf zu aktualisieren.

Für den Nutzer besonders komfortabel sind PIDs, die nicht nur auflösbar, sondern auch direkt anklickbar, also (*actionable*) sind. Dies ist im Moment nur der Fall, wenn der PID ein URL ist oder als URL angegeben wird. Letzteres kann man als „URL-fizierung“ (*URL-ify*) bezeichnen, wobei üblicherweise ein Resolver-Dienst für das verwendete PID-System adressiert wird. Beispielsweise kann der DOI *10.1000/182* auch als <http://doi.org/10.1000/182> angegeben werden.

Die URL-fizierung ist aber problematisch, da nicht gewährleistet ist, dass der URL des Resolver-Dienstes über die Zeit bestehen bleibt oder gar, dass das Adressieren per URLs nicht irgendwann durch einen anderen Mechanismus ersetzt wird.

Eine Alternative bilden Browser-Plugins, die es dem Nutzer erlauben einen PID direkt in die Adresszeile einzugeben oder in einem Dokument anzuklicken. Allerdings müssen diese aktiv vom Nutzer installiert werden und es ist nicht gewährleistet, dass sie auf jedem System funktionieren.

Wünschenswert wäre es, dass Browser nativ mit PIDs umgehen können. Um dies zu erreichen bräuchte man aber einen zentralen Auflösungsdienst (analog zu den DNS-Servern, die aktuell Domains auf die entsprechenden physikalischen Adressen auflösen), was aber angesichts der Menge an PID-Systemen und deren vergleichsweise noch geringen Verbreitung noch eine ganze Weile dauern wird.

5.2 Flexibilität

Ein PID-System sollte so flexibel sein, dass die unterschiedlichsten digitalen Objekte und Granularitätsebenen referenziert werden können. Darauf muss vor allem in den Metadaten Rücksicht genommen werden. Auch die Umsetzung individueller (und sinnvoller) Wünsche des Datengebers oder der Nutzer sollte ermöglicht werden.

5.3 Metadaten

Üblicherweise beinhaltet ein PID nicht nur den Verweis auf ein bestimmtes Objekt, sondern auch weitere Informationen, die sogenannten Metadaten, die Aufschluss über den Autor, das Format oder den Inhalt geben können.

Mit Hilfe der Metadaten kann auch überprüft werden, ob das Objekt auf das ein PID verweist, auch wirklich das richtige ist. Idealerweise sollten die Metadaten einfach und übersichtlich abrufbar sein, am besten noch bevor das eigentliche Objekt angezeigt wird. Ein wichtiger Grund hierfür ist zum Beispiel, dass dem Nutzer angezeigt wird, in welchem Dateiformat das Objekt vorliegt, so dass dieser sehen kann, ob er die Datei überhaupt auf seinem Gerät öffnen kann.

Metadaten bieten dem Nutzer nicht nur die Möglichkeit ein Objekt zu identifizieren, sondern auch zu überprüfen, dass das zurückgegebene Objekt auch wirklich das Objekt ist, das abgefragt wurde. Beispielsweise könnten in den Metadaten hinterlegte Prüfsummen dem Nutzer erlauben, das Objekt auf Unversehrtheit zu überprüfen.

5.4 Opazität und Semantik

Opazität bezieht sich auf den Teil eines PIDs, der das zu identifizierende Objekt benennt, also den Namen. Ein opaker Name gibt keinen Aufschluss über das Objekt. Im Gegensatz dazu stehen sprechende Namen, die einen ersten Eindruck über den Inhalt des Objektes vermitteln.

Sprechende Namen haben den Vorteil, dass sie leichter zu merken sind. Allerdings bergen sprechende Namen Probleme, wie etwa Nomenklaturänderungen, Bedeutungsänderungen oder auch die ambige Definition von Kurzformen.

Besser eignen sich nicht zu lange opake Namen, die einer bestimmten Syntax folgen. Um sicherzustellen, dass der Name korrekt eingegeben wurde, könnten Prüfziffern verwendet werden, wie es auch bei ISBN-Nummern der Fall ist. Damit können Tippfehler abgefangen werden.

5.5 Interoperabilität

Interoperabilität bezeichnet die Zusammenarbeit verschiedener Systeme oder Techniken. In diesem Sinne sollte zum einen das jeweils verwendete Metadatenschema auf andere Schemata abbildbar sein und zum anderen wäre es wünschenswert, dass die verschiedenen PID-Systeme untereinander auflösbar sind. In diesem Zusammenhang ist es auch wichtig, dass die verschiedenen Resolver-Dienste mit den verschiedenen PID-Systemen umgehen können und bei Bedarf auf das jeweilige andere System verweisen können.

5.6 Granularität

Granularität meint, wie detailliert Objekte identifiziert werden sollen. Beispielsweise kann einer Datensammlung lediglich ein übergeordneter PID zugewiesen werden oder aber jede einzelne Datei in der Datensammlung bekommt ihren eigenen PID.

Wird jeder einzelnen Datei ein Identifikator zugewiesen, so kann die Anzahl an einzelnen PIDs sehr schnell sehr groß werden, was ein wichtiger Kostenfaktor sein kann. Andererseits kann die Vergabe von Identifikatoren pro Datensammlung in sehr großen Projektarchiven problematisch werden, wenn eine bestimmte Datei identifiziert werden soll, die hierarchisch sehr tief „vergraben“ ist.

Ein Faktor, der bei der Granularität eine entscheidende Rolle spielt, ist die Zitierbarkeit oder Zitierweise eines Objektes. Beispielsweise wird ein Buch üblicherweise als ein eigenes zitierbares Werk betrachtet, während ein einzelnes Kapitel oder gar eine einzelne Seite mittels der Referenz auf das übergeordnete Buch zitiert werden.

Wird nicht standardmäßig jeder einzelnen Datei ein eigener PID zugewiesen, so kann eigentlich nur der Datengeber selbst am besten entscheiden welches Objekt einen Identifikator braucht. Ergänzend könnte man anbieten, gegebenenfalls PIDs auf Wunsch von Datennutzern zu vergeben („PID on Demand“). In Anhang A wurde ein Fallbeispiel mit unterschiedlichen Granularitätsebenen und deren Implikationen durchgespielt.

5.7 Versionierung

Da persistente Identifikatoren unveränderlich sind, sollte das zu identifizierende Objekt nicht nachträglich geändert werden. Wenn beispielsweise eine Publikation mittels eines PIDs auf ein Objekt verweist, dieses Objekt jedoch nachträglich

verändert wird, kann ein Leser unter Umständen einen Argumentationsvorgang nicht korrekt nachvollziehen, da das Objekt nicht mehr das gleiche ist, auf das der Autor sich ursprünglich bezog.

Dementsprechend muss jeder neuen Version eines Objektes ein eigener neuer PID zugewiesen werden. Allerdings sollte das PID-System die Möglichkeit bieten, die verschiedenen Versionen miteinander zu verknüpfen, um den Nutzer auf neuere oder ältere Versionen hinweisen zu können. Eine Ausnahme bilden PIDs für Konzepte, wie zum Beispiel der DOI 10.1000/182, welcher immer auf die aktuellste Version des DOI Handbuchs verweist.

5.8 Zuverlässigkeit und Vertrauenswürdigkeit

Ein PID-System muss zuverlässig funktionieren und rund um die Uhr verfügbar sein. Nutzer werden sehr schnell von der Verwendung eines Systems absehen, das selten erreichbar ist, langsam reagiert oder gar falsche Auflösungen tätigt.

Die Vertrauenswürdigkeit in ein PID-System wird insbesondere durch Transparenz und Nachvollziehbarkeit gewährleistet. Außerdem muss dem Nutzer die Möglichkeit gegeben werden zu überprüfen, ob er zum richtigen Objekt geleitet wurde und ob dieses nicht manipuliert wurde.

Allgemeine Metadaten über das Objekt, wie Autor, Titel oder Datum, die dem Nutzer beim Aufruf eines PIDs auf einer sogenannten Landing Page angezeigt werden, stellen einen ersten Schritt in diese Richtung dar. Ideal wären aber weitere Metadatenfelder, zur Hinterlegung von Prüfsummen, mit denen der Nutzer bei Bedarf die Integrität des Objektes überprüfen lassen kann.

5.9 Autorität

Ein Faktor, der auch für die Vertrauenswürdigkeit des PID-Systems eine große Rolle spielt, ist die Autorität (*authority*) des Systems und dessen bereitstellenden Anbietern.

Es gilt zu bedenken welche Verbreitung das PID-System findet und welchen Stand es in dem gewünschten Anwendungsgebiet hat. Eine aktive Community dahinter und ein Anbieter, der seine Verpflichtungen (*commitment*) klar darstellt und sich daran hält steigern die Glaubwürdigkeit. Ein längeres Bestehen, übergeordnete nationale oder internationale Einrichtungen und Zertifikate können die Autorität unterstreichen.

5.10 Kosten

Bei der Wahl eines PID-Systems müssen die möglichen Kosten für Wartung und Entwicklung des Systems berücksichtigt werden. Auch sollte überprüft werden, ob für die Vergabe einzelner PIDs Gebühren erhoben werden. Vor der Wahl eines bestimmten PID-Systems muss die Verteilung der Kosten und eine eventuelle Refinanzierung geklärt werden.

6 Vorhandene PID-Systeme

Es gibt eine Vielzahl an PID-Systemen, die sich nur in feinen Details voneinander unterscheiden und teilweise Untermengen oder Teilmengen voneinander sind.

Auch wenn es sich nicht immer um offiziell registrierte Schemata handelt, können alle PID-Systeme als URIs bezeichnet werden. In Abbildung 2 wird veranschaulicht, wie die verschiedenen hier vorgestellten PID-Systeme sich zueinander verhalten. Der Pfeil von der Menge der Handles und DOIs hin zu der Menge der URLs meint, dass diese durch URL-fizierung zu ebensolchen umgewandelt werden können. Das gleiche gilt auch für die Menge der XRIs und der NBNs.

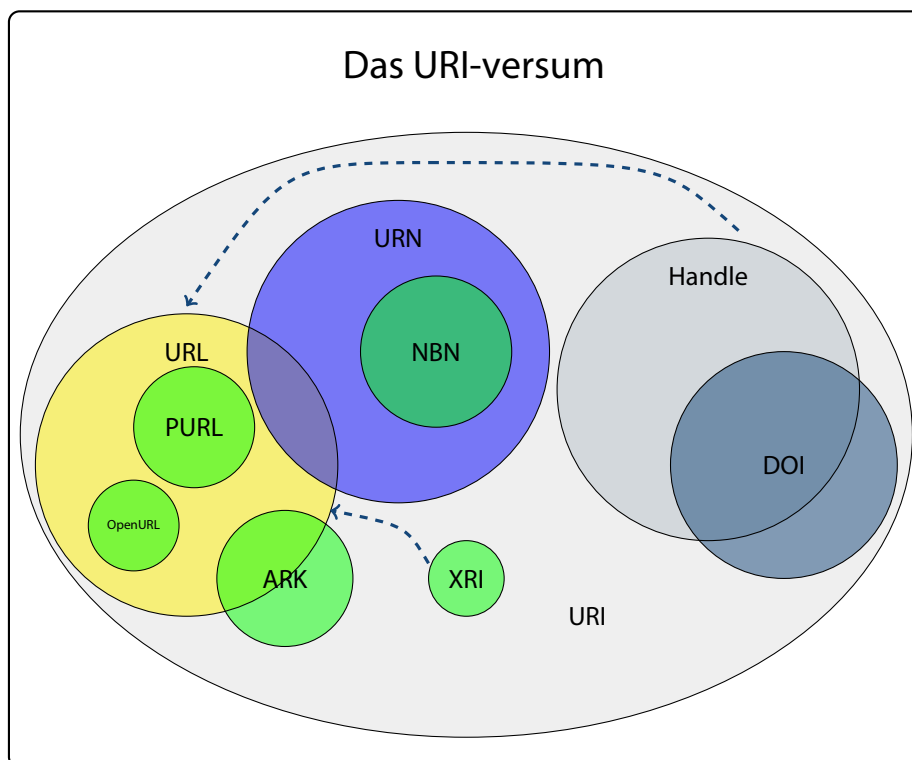


Abbildung 2: Veranschaulichung der Abhängigkeit der verschiedenen PID-Systemen untereinander.

6.1 Übersicht der PID-Systeme

Für dieses Testbed wurden nur PID-Systeme berücksichtigt, die bereits länger etabliert sind und durch ihre Nutzerzahlen eine gewisse Relevanz erreicht haben.

6.1.1 URI

Uniform Resource Identifier
 (einheitlicher Bezeichner für Ressourcen)
 Seit 1994
 Entwickelt von W3C und IETF

Ein URI ist ein Identifikator, der aus einer Reihe von Zeichen besteht, die einer bestimmten festgelegten Syntax folgen. Mit einem URI können einheitlich Ressourcen

mittels eines separat definierten erweiterbaren Satzes von Schemas identifiziert werden. Wie diese Identifizierung erfolgt, übertragen oder ermöglicht wird, hängt von der Spezifikation des angegebenen Schemas ab.

Ein URI ist ein übergreifender Begriff für alle Möglichkeiten, um ein Ressource im Internet zu identifizieren und kann entweder als ein Name, als ein „locator“ (wo etwas zu finden ist) oder als beides klassifiziert werden.

Syntax

URI = <Schema>:<schemenspezifischer Teil>[?<Anfrage>]

Ein URI besteht immer aus der Angabe des verwendeten Schemas gefolgt von einem Doppelpunkt und anschließend dem schemenspezifischen Teil. Optional kann durch Anhängen eines Fragezeichens noch eine bestimmte Anfrage (*query*) gemacht werden.

Es gibt eine Reihe offizieller Schemata, die bei der IANA registriert sind², wie etwa *http*, *info*, *mailto* oder *urn*. Außerdem werden weitere etablierte Schemata verwendet, die aber nicht registriert sind.

Das Schema gibt an womit ein URI aufgelöst wird. Beispielsweise werden bei URIs, die das HTTP-Schema verwenden, die Domains immer mit DNS aufgelöst. Zudem bestimmt das Schema darüber, wie der schemenspezifische Teil aussieht, wie die folgenden Beispiele zeigen.

Beispiele

```

http://example.org/pfad/zur/datei.txt
  \--/ \-----/ \-----/
   |      |              |
   |      Authority      Pfad
   |      \-----/
   |      |
Schema    spezieller Teil
  
```

```

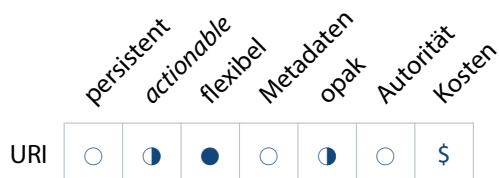
mailto:ianus-fdz@dainst.de?Subject=PIDs
 \--/ \-----/ \-----/ \-----/
  |      |          |          |
  |      userinfo   hostname   query
  |      \-----/
  |      |
Schema    spezieller Teil
  
```

```

urn:ISBN:0262531283
 \--/ \--/ \-----/
  |      |          |
  |      Namensraum spez. Teil
  |      \-----/
  |      |
Schema  spezieller Teil
  
```

²<http://www.iana.org/assignments/uri-schemes>

Eigenschaften



Die Persistenz eines URIs hängt von verschiedenen Faktoren ab, wie beispielsweise von dem Erzeuger und dessen Engagement für die Aufrechterhaltung der Ressource. Abhängig von dem verwendeten Schema sind URIs direkt anklickbar, also *actionable*. Worauf ein URI angewendet wird, ist nicht näher spezifiziert, weshalb eine hohe Flexibilität gegeben ist. Wie Metadaten hinterlegt werden sollen, ist nicht definiert. Da beliebige Domainnamen, Kontextnamen und auch beschreibende Objektamen verwendet werden dürfen, sind URIs nicht immer opak. Domainnamen können auch die Komponente bilden, in der die Autorität, also die dahinter stehende Institution, abgebildet wird.

URIs sind an sich kostenlos, können jedoch in Abhängigkeit von dem Schema und von einer eventuellen Registrierungsagentur Kosten nach sich ziehen.

Verbreitung

URIs gehören zum Internet-Standard.

Spezifikation und Quellen

<http://www.ietf.org/rfc/rfc3986.txt>

6.1.2 URL

Uniform Resource Locator
(einheitlicher Quellenanzeiger)
Seit 1994
Entwickelt von W3C und IETF

Der Begriff URL bezieht sich auf eine Untermenge der URIs und bezeichnet URIs, die nicht nur eine Ressource identifizieren, sondern auch ein Mittel zur Lokalisierung der Ressource bereitstellen, indem der primäre Zugriffsmechanismus beschrieben wird.

Ein URL adressiert also eine bestimmte Ressource, was vergleichbar mit der postalischen Adresse einer Person ist.

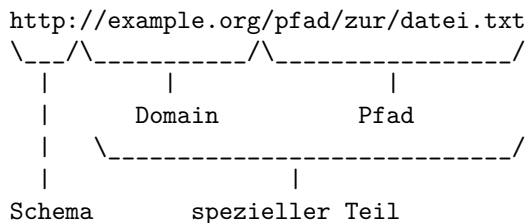
Syntax

URL = <schema>:<schemenspezifischer Teil>

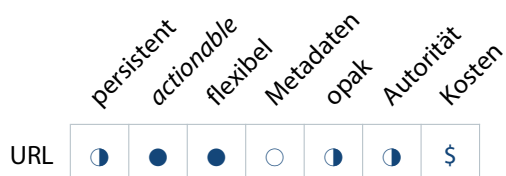
Das Schema oder Protokoll beschreibt, welcher Zugriffsmechanismus zu verwenden ist, um die Verbindung zur Ressource aufzubauen. Meist wird *http* oder *https* verwendet.

Der schemenspezifische Teil kann aus der Domain oder dem Host, die angeben wohin verbunden wird, und einem Pfad, der angibt wonach gesucht wird, bestehen. Aus der Domain kann die dahinter stehende Autorität abgelesen werden.

Beispiel



Eigenschaften



Wenn Ressourcen den Speicherort wechseln, können die URLs ungültig werden, also in gewisser Weise verrotten (*rot*), weshalb sie nur persistent sind, wenn die erzeugende Instanz dahinter auch dafür sorgt. Zudem ist es möglich, dass aufgegebene URLs wiederverwendet werden, wodurch in einem ungünstigen Fall ein alter URL auf eine völlig andere Datei als die ursprüngliche zeigen kann.

Eingebettete URLs sind in vielen Medien direkt anklickbar und somit auch *actionable*. Sie haben zudem die gleiche Flexibilität wie URLs. Eine zusätzliche Speicherung von Metadaten ist nicht vorgesehen.

URLs sind nur opak, wenn von sprechenden Domainnamen und Pfadbestandteilen abgesehen wird. Die Domainnamen können, wie bei den URLs, die Komponente bilden, mit welcher die dahinter stehende Institution, abgebildet wird.

Bis auf die Kosten für die Domain sind URLs kostenlos.

Verbreitung

URLs gehören zum Internet-Standard.

Spezifikation und Quellen

<http://tools.ietf.org/html/rfc1738>

6.1.3 URN

Uniform Resource Name
(einheitlicher Name für Ressourcen)

Seit 1997

Entwickelt von W3C und IETF

Der Begriff URN bezieht sich auf URIs, die das urn-Schema verwenden, um global einzigartig und persistent zu bleiben, auch wenn die Ressource nicht mehr verfügbar ist. Andere URIs, welche die Eigenschaften eines Namens besitzen, also lediglich Objekte benennen, können auch als URN bezeichnet werden.

URNs können nicht auf den Ort des referenzierten Objektes verweisen, sind also ortsunabhängig. Beispielsweise identifiziert eine URN mittels der ISBN-Nummer ein bestimmtes Buch, aber nicht den Ort an dem es sich befindet.

Syntax

URN = urn:<Namensraum>:<spezieller Teil>

Ein URN wird durch das Kürzel *urn*, gefolgt von einem Doppelpunkt, begonnen. Anschließend folgt die Angabe eines Namensraums (*namespace*), in der Form eines Namensraum Identifikators (*NID, namespace identifier*). Offizielle Namensräume sind bei der IANA³ registriert und erlauben es, andere Schemata als Untermengen von URNs zu integrieren.

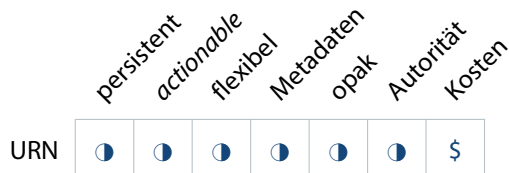
Die Angabe des Namensraumes wird durch einen Doppelpunkt von dem speziellen Teil getrennt. Die Syntax des speziellen Teils wird durch den angegebenen Namensraum näher definiert.

Beispiele

urn:ISBN:0262531283

Schema Namensraum spez. Teil

Eigenschaften



Auch URNs sind nicht unbedingt persistent. Allerdings sind manche Namensräume weiter spezifiziert und institutionalisiert, so dass eine Persistenz so gut wie sicher gegeben ist. Ein Beispiel hierfür ist *urn:nbn*, das auf Seite 18 näher beschrieben wird. Die übrigen Eigenschaften hängen ebenfalls von dem verwendeten Namensraum ab.

Verbreitung

URNs gehören zum Internet-Standard.

Spezifikation und Quellen

<http://www.w3.org/2001/tag/doc/URNsAndRegistries-50>

<http://tools.ietf.org/html/rfc2141>

³<http://www.iana.org/assignments/urn-namespaces>

6.1.4 PURL

Persistent Uniform Resource Locator

(persistenter einheitlicher Quellenanzeiger)

Ab 1995

Entwickelt von OCLC

Ein PURL ist ein URL. Allerdings zeigt ein PURL nicht direkt auf den Ort einer Ressource, sondern verweist zunächst auf einen zwischengeschalteten Linkresolver, also auf eine Art Auflösungsdienst (*resolution service*). Dieser Dienst verknüpft den PURL mit dem eigentlichen URL, der auf den Ort der Ressource zeigt, und leitet den Nutzer weiter. Dadurch kann auf das Objekt referenziert werden, ohne den Speicherort zu berücksichtigen. Wird ein digitales Objekt an einem anderen Ort abgelegt, so muss lediglich der URL in dem Linkresolver angepasst werden.

PURLs wurden so gestaltet, dass sie sich leicht in URNs umwandeln lassen, weshalb sie manchmal auch als Zwischenlösung vor der verbreiteten Verwendung von URNs bezeichnet wurden. Sie können von jedem erzeugt werden, der eine Domain besitzt. Zur Pflege der PURLs werden zusätzlich Metadaten für jedes Objekt gespeichert.

Zu Beachten ist, dass die Abkürzung PURL auch für personalisierter einheitlicher Quellenanzeiger (*personalized uniform resource locator*) stehen kann.

Syntax

PURL = <Schema>:<schemenspezifischer Teil>

PURL = http://<Host>/<Name>

Da ein PURL ein URL ist, entspricht die Syntax der eines URLs. Als Schema wird häufig *http* verwendet, gefolgt von der Angabe der Domain, welche durch einen Schrägstrich von dem Namen des digitalen Objektes getrennt ist.

Das zweite Beispiel zeigt die Umwandlung eines PURL in ein URN.

Beispiele

```
http://purl.oclc.org/OCLC/PURL/FAQ
```

```
\_--/ \-----/ \-----/
|      |             |
Schema Domain       Name
```

```
http://purl.oclc.org/keith/home
```

```
\_--/ \-----/ \-----/
|      |             |
Schema Host         Name
```

Umgeformt in ein URN:

```
URN:/org/oclc/purl/keith/home
```

```
\_--/ \-----/ \-----/
|      |             |
Schema Naming authority Name
```

Eigenschaften

	persistent	actionable	flexibel	Metadaten	opak	Autorität	Kosten
PURL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

PURLs werden persistent, wenn die Instanz dahinter dafür sorgt. Da ein PURL auch ein URL ist, ist dieser direkt anklickbar und somit *actionable*. Da sprechende Domain-, Kontext- und Objektamen verwendet werden können, sind PURLs nicht immer opak.

Der zentrale PURL-Dienst des OCLCs ist an sich kostenlos und es besteht die Möglichkeit, selbst zum Anbieter mittels der zur Verfügung gestellten Software zu werden.

Metadaten, die über die Informationen zu dem PURL (URL und Registrant) hinausgehen, sind nicht zwingend erforderlich und können auch nur mittels extra angehängter Datei hinzugefügt werden.

Verbreitung

PURLs sind eher im Nordamerikanischen Raum verbreitet und finden vor allem im Bibliotheksbereich, im Regierungsbereich und teilweise in Biologischen Fächern Anwendung. Da es keine zentrale Registrierungsstelle gibt, können keine genauen Zahlen genannt werden. Statistiken von 2007 nennen 669.733 registrierte PURLs, die insgesamt 378.307.173 mal aufgelöst wurden.⁴

Spezifikation und Quellen

<http://www.purl.org/docs/index.html>
http://purl.oclc.org/docs/long_intro.html
<http://www.oclc.org/research/activities/purl.html>
<http://bibpurl.oclc.org/faq.html>

6.1.5 NBN

National Bibliographic Numbers
(nationale Bibliographienummern)
Seit 1998 produktiv; seit 2001 bei der IANA registriert
Von der Nationalbibliothek in Finnland entwickelt

NBN ist ein Namensraumidentifikator für URNs. NBNs stellen also eine Untermenge der URNs dar, weshalb sie auch als URN:NBN bezeichnet werden. Wie der Name schon verrät, sind NBNs speziell für die Bedürfnisse von Nationalbibliotheken entwickelt worden und im Fokus steht die Benennung von gedruckten und elektronischen Publikationen. Außerdem ist die Verwendung von URNs nur für Nationalbibliotheken und den von ihnen autorisierten Institutionen gedacht.

⁴<http://www.mpi.nl/dam-ir/meetings/Persistent%20identifiers.pdf>, Folie Nr. 7

Global werden NBNs bei der Library of Congress registriert. Allerdings gibt es für die einzelnen Länder, die mittels ISO Länderkennungen gekennzeichnet werden, Unternehmensräume (sub-namespaces). Die Registrierung für die Unternehmensräume wird von der jeweiligen Nationalbibliothek übernommen, welche auch über weitere Bestandteile der Syntax entscheidet und gegebenenfalls weitere untergeordnete Namensräume vergibt.

Da jede Nationalbibliothek ihre eigenen Entscheidungen treffen kann, gibt es keine Anwendungen, die für alle nationalen Unternehmensräume verwendet werden können. Daher gibt es auch keinen zentralen Resolver für die NBNs. Stattdessen muss man den jeweiligen Dienst der zuständigen Nationalbibliothek verwenden.

Syntax

NBN = urn:nbn:<ISO 3166 Länderkennung>-<zugewiesener NBN-String>

NBN = urn:nbn:<ISO 3166 Länderkennung>:

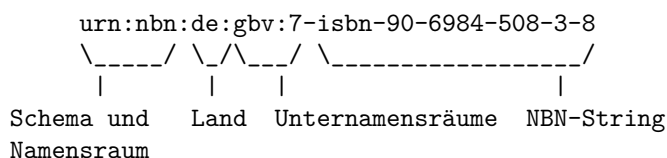
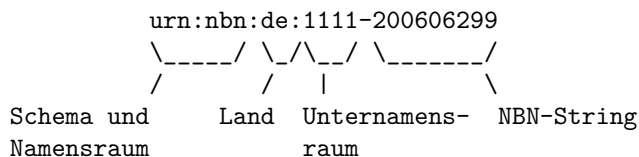
<Unternehmensraum>-<zugewiesener NBN-String>

NBN = urn:nbn:<non-ISO 3166 Präfix>-<zugewiesener NBN-String>

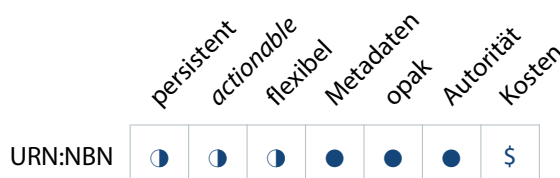
Eine NBN wird durch das URN-Schema *urn:nbn* gekennzeichnet. Darauf folgt ein Doppelpunkt und eine ISO 3166 Länderkennung. Ausnahmen können auch ein beliebiges anderes Präfix enthalten. Auf die Länderkennung folgt, durch einen Bindestrich getrennt, der von der registrierenden Einrichtung zugewiesene NBN-String, also der Name des Objektes.

Die jeweiligen Nationalbibliotheken können Unternehmensräume vergeben, die direkt hinter die Länderkennung nach einem Doppelpunkt, eingetragen werden.

Beispiele



Eigenschaften



Die Zentralisierung der Vergabe von URN:NBNs sorgt dafür, dass diese auch eine höhere Lebensdauer als etwa URLs haben. Allerdings werden in Deutschland nicht für alle Objektarten NBNs vergeben, weshalb die Flexibilität eingeschränkt ist.

NBNs wurden opak entworfen und es müssen immer Metadaten zusätzlich abgelegt werden. Sie können direkt anklickbar gemacht werden, wenn sie als URL notiert werden. Üblicherweise wird dafür die Adresse des Auflösungsdienstes der NBN vorangestellt, wie zum Beispiel:

<http://nbn-resolving.de/urn:nbn:de:gbv:7-isbn-90-6984-508-3-8>

Es besteht allerdings die Gefahr, dass die Adresse des Auflösungsdienstes sich ändert und damit die gesamte URL-Adresse ungültig wird, obwohl die NBN noch existiert. Der URN-Service der deutschen Nationalbibliothek ist kostenlos, weshalb nur mit geringen Kosten für die Aufbereitung gerechnet werden muss. Die zentralisierte Vergabe von URN:NBNs durch die jeweiligen Nationalbibliotheken gewährleistet eine hohe Autorität hinter dem System.

Verbreitung

URN:NBNs werden im Bibliotheksbereich verwendet und scheinen dort recht weit verbreitet zu sein. Mitte 2011 waren es knapp über 4 Millionen registrierte NBNs.⁵

Spezifikation und Quellen

<http://tools.ietf.org/html/rfc3188>

<http://www.dnb.de/urnservice>

<http://www.persistent-identifier.de/>

Policy für die Vergabe von URNs im Namensraum urn:nbn:de:

<http://nbn-resolving.de/urn:nbn:de:101-2012121200>

6.1.6 Handle (HS)

Seit 1994

Entwickelt von CNRI und anfangs auch mit DARPA

Im November 2003 als RFC veröffentlicht

Das Handle System ist ein Softwarepaket mit Protokollen, einem Namensraum und der Implementierung einer Referenzsoftware, um Identifikatoren zu verwalten, diese zu identifizieren und aufzulösen. Ein Handle ist ein eindeutiger Name für digitale Objekte und wird von dafür autorisierten Anbietern (*naming authorities*) vergeben. Da das Handle System eigene Protokolle und einen eigenen *root server* verwendet, um Handles aufzulösen, ist es unabhängig von aktuellen Diensten, wie etwa DNS. Es wurde so konzipiert, dass die Identifikatoren mit beliebigen Systemen aufgelöst werden können, unter anderem auch als URLs.

Obwohl es sich um ein einzelnes System handelt, ist es physisch auf unterschiedliche lokale Anbieter verteilt, weshalb es zuverlässig jederzeit erreichbar ist, sich aber auch in diversen Funktionsdetails unterscheiden kann.

Das Handle System stellt die Basis von DOIs dar, die im nächsten Abschnitt beschrieben werden.

⁵http://www.knowledge-exchange.info/Admin/Public/DWSDownload.aspx?File=%2fFiles%2fFiler%2fdownloads%2fPersID%2fWorkshop+POID%2fURN-experience_kett.pdf, Folie 12

Syntax

Handle = <Präfix>/<Suffix>

Handle = <Präfix>/<fg>-<Kürzel>-<Num1>-<Num2>-<Num3>-<Prüfziffer>

Das Präfix besteht aus Ziffern, welche die Handle Naming Authority kodieren. Das Suffix, der eigentliche Name des Handles wird in Abhängigkeit der Spezifikationen des verwendeten Anbieters gestaltet.

Beispielsweise setzt sich das Suffix der Handles, die von EPIC vergeben werden, aus einer zweistelligen Kennzeichnung (einer sogenannten *flag* (fg)), dem Kürzel der verantwortlichen Institution, drei Ziffernblöcken und einer Prüfziffer zusammen. Letztere dient der Vermeidung von Eingabefehlern.

Beispiele

10013/epic.10286.d001

```

\___/ \_____ /
 |      |
Präfix  Suffix
  
```

11858/00-ZZZZ-0000-0001-6D1D-0

```

\___/ \_/ \___/ \___/ \___/ \___/ \
 |      | | | | | | |
 |      fg Kürzel num1 num2 num3 Prüfziffer
 |      \_____ /
 |
Präfix  Suffix
  
```

Eigenschaften



Auch Handles können durch entsprechende Plugins oder Umwandlungen in URLs direkt anklickbar gemacht werden. Sie können vollkommen flexibel auf jedes Objekt angewendet werden. Welche Metadaten hinterlegt werden müssen und wie opak der Name ist, hängen von der Registrierungsagentur ab, da hier Spielraum besteht. Die Software ist frei. Es wird lediglich eine kleine jährliche Gebühr (\$50) von dem globalen Handle Service erhoben, wenn man selbst als Registrierungsagentur auftreten möchte.⁶

Verbreitung

Die genaue Zahl von registrierten Handles lässt sich nicht ermitteln, da es tausende dafür eingerichtete Dienste gibt, die sich auf 70 Länder und sechs Kontinente

⁶<http://hdl.handle.net/4263537/5029>

verteilen. Mittlerweile gibt es mehr als 200.000 registrierte Präfixe und allein für das DOI-Präfix sind über 84 Millionen Handles registriert. Die der CNRI bekannten Handle-Server lösen pro Monat zwischen 75 und 100 Millionen Anfragen auf.⁷

Spezifikation und Quellen

<http://www.handle.net/>

Gebühren: RFC 3650: <http://www.handle.net/rfc/rfc3650.html>

RFC 3651: <http://www.handle.net/rfc/rfc3651.html>

RFC 3652: <http://www.handle.net/rfc/rfc3652.html>

6.1.7 DOI

Digital Object Identifier System
(digitales Objektidentifikatorensystem)

Seit 1996, öffentlich seit 1998

Entwickelt von Association of American Publishers, teilweise mit CNRI

Koordiniert von IDF

Seit 2010: ANSI/NISO Z39.84-2005 (R2010) Syntax

Seit 2012: ISO 26324:2012

Das DOI System, das von der IDF verwaltet und kontrolliert wird, basiert auf dem Handle System. Im DOI System werden DOI Namen zu den zugehörigen Objekten aufgelöst. Der DOI Name ist permanent mit dem digitalen Objekt verbunden, wobei darauf Rücksicht genommen wird, dass Informationen über das Objekt (Metadaten) oder der Speicherort des Objekts im Laufe der Zeit geändert werden können.

Metadaten, die mit einem DOI Namen verknüpft sind, können einem festgelegten Metadaten-schemata entsprechen, was eine bessere Interoperabilität ermöglicht. Beispielsweise stellt DataCite so ein Metadaten-schemata zur Verfügung. DataCite ist eine globale Registrierungsagentur für DOIs im Bereich der Wissenschaft.

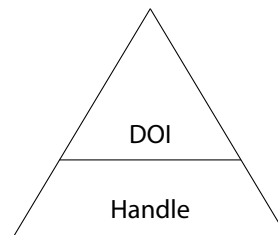


Abbildung 3: DOI setzt auf dem Handle System auf.

Syntax

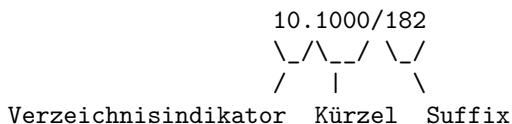
DOI = 10.<Naming Authority>/<Suffix>

Ein DOI besteht, wie ein Handle auch, aus einem Präfix und einem Suffix. Der Präfix setzt sich aus einem Verzeichnisindikator (*directory indicator*), aktuell immer „10.“, und dem meist vierstelligen Code der Registrierungsagentur (*naming authority*) zusammen.

Das Suffix besteht aus einer Zeichenkette, kann theoretisch beliebig lang sein und jedes beliebige UTF8-Zeichen enthalten.

⁷<http://www.handle.net/factsheet.html>

Beispiel



Eigenschaften



Da DOIs auf dem Handle System basieren, sind die Eigenschaften im Prinzip gleich. DOIs zeichnen sich lediglich durch ihre Zusatzfunktionen, ihre zentrale Strukturierung mit der IDF an der Spitze und strikteren Spezifikationen aus.

DOIs dürfen nur von Registrierungsagenturen, die der IDF angehören, vergeben werden. Seit Beginn 2013 sind DOIs im Bereich Wissenschaft und Forschung in Deutschland kostenfrei.⁸ Lediglich im Bereich von Verlagen und anderen kommerziellen Dienstleistern muss mit einer allgemeinen Gebühr und einer weiteren Gebühr für jeden vergebenen DOI gerechnet werden, weshalb die Kosten sehr schnell ansteigen können. Es darf nicht vergessen werden, dass die IDF kein gemeinnütziger Verein oder eine Forschungseinrichtung ist, sondern eine Firma, die für ihren Fortbestand auch von den Einnahmen abhängig ist.

Auch DOIs können durch URL-fizierung *actionable* gemacht werden.

Verbreitung

DOIs haben ihren Ursprung im Verlagswesen und sind dort entsprechend weit verbreitet. Die Nutzerzahlen steigen stetig und inzwischen ist das System auch in anderen Domänen verbreitet.

Es gibt über 84 Millionen vergebene DOIs und mehr als 5.000 Registrierungsagenturen mit insgesamt mehr als 215.000 Präfixen. Jährlich werden mehr als eine Milliarde Auflösungen angefragt.⁹

Spezifikation und Quellen

<http://www.doi.org/>

<http://www.doi.org/hb.html>

http://www.niso.org/apps/group_public/project/details.php?project_id=62

http://www.iso.org/iso/catalogue_detail?csnumber=43506

⁸<http://www.tib-hannover.de/de/die-tib/aktuelles/aktuelles/id/362/>

⁹<http://www.doi.org/factsheets/DOIKeyFacts.html>

6.1.8 ARK

Archival Resource Key

(Archivalischer Ressourcen-Schlüssel)

Seit 2001

Entwickelt von John Kunze und R.P.C. Rogers an der US National Library of Medicine; wird von der California Digital Library gepflegt

ARKs sind URLs, die so gestaltet wurden, dass ein langfristiger Zugang zu digitalen Objekten ermöglicht wird. Der Grundgedanke von ARKs ist, dass die Persistenz allein eine Angelegenheit des Dienstleisters ist, und nicht dem Objekt innewohnt oder diesem durch eine spezielle Benennungssyntax übertragen wird. Der Nutzer wird auf die Dienste und Verpflichtungen des Dienstleisters hingewiesen.¹⁰

Ein ARK verlinkt den Nutzer zu drei Sachen: dem Objekt (*Access Service*), dessen Metadaten (*Description Service*) und der Erklärung der Verpflichtungen des aktuellen Anbieters (*Policy Service*). Diese verschiedenen Aspekte können durch Anhängen von keinem, einem oder zwei Fragezeichen (sog. *qualifier*) abgerufen werden. Allerdings führen die Fragezeichen in Browsern häufig zu Problemen, weshalb nicht alle ARK-Systeme dieses Feature umgesetzt haben.

Syntax

ARK = `ark:/<NAAN>/<Name>[qualifier] [?|??]`

ARK = `[http://<NMAH>/]ark:/<NAAN>/<Name>[qualifier] [?|??]`

Ein ARK besteht mindestens aus der Nummer der Namensgebenden Institution (NAAN, *Name Assigning Authority Number*) und dem Namen des zu identifizierenden Objektes, welche durch ein „/“ voneinander getrennt sind. Optional kann hinter dem Namen noch ein sogenannter *qualifier* folgen, womit ein bestimmter Teil eines Objektes, etwa ein bestimmtes Bild, aufgerufen werden kann. An das Ende des ARKs können ein oder zwei Fragezeichen gestellt werden, um den *Description Service* oder den *Policy Service* aufzurufen.

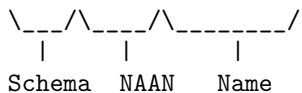
Die Liste der Nummern der Namensgebenden Institution kann auf http://www.cdlib.org/services/uc3/naan_table.html gefunden werden.

Ein ARK kann auch als URL angegeben werden. Dazu wird das *http*-Schema zusammen mit der Namenszuordnenden Autorität (NMAH, *Name Mapping Authority*) vorangestellt. Laut Spezifikation ist der Teil mit der NMAH zeitlich begrenzt, entbehrlich und ersetzbar, weshalb diese Variante besser nicht als Angabe in einer Referenz verwendet werden sollte.

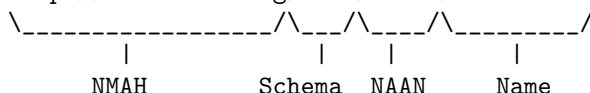
¹⁰Kunze (2003)

Beispiele

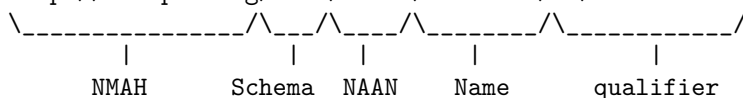
ark:/13030/ft4w10060w



http://ark.cdlib.org/ark:/13030/ft4w10060w



http://example.org/ark:/12025/654xz321/s3/f8.05v.tiff



Eigenschaften

persistent
actionable
flexibel
Metadaten
opak
Autorität
Kosten



ARKs können flexibel vergeben werden und ermöglichen durch die Angabe von *qualifiern* einen bequemen Umgang mit granularen Objekten und Objektbestandteilen.

Metadaten können gesondert vom Objekt durch anhängen von einem Fragezeichen an den ARK aufgerufen werden. ARKs sind laut Definition opak.

Jede stabile Institution kann kostenlos eine Nummer (NAAN) erhalten und ARKs vergeben und registrieren.

Verbreitung

ARKs wurden an der California Digital Library entwickelt und sind dementsprechend auch eher im Bibliotheksbereich und dort speziell im wissenschaftlichen oder universitären Bereich zu finden. Aktuell sind 191 Institutionen mit einer NAAN registriert.¹¹

Dadurch, dass die französische Nationalbibliothek ARKs verwendet, sind sie ebenfalls innerhalb Frankreichs weit verbreitet. Genauere Zahlen zu der Menge an registrierten ARKs konnten nicht gefunden werden.

Spezifikation und Quellen

<http://www.cdlib.org/inside/diglib/ark>

<https://wiki.ucop.edu/display/Curation/ARK>

http://www.cdlib.org/services/uc3/docs/jak_ARKs_Berlin_2012.pdf

<https://wiki.ucop.edu/download/attachments/16744455/arkcdl.pdf>

¹¹http://www.cdlib.org/uc3/naan_table.html

6.1.9 XRI

Extensible Resource Identifier

(erweiterbarer Ressourcenidentifikator)

Seit 2003

Entwickelt von OASIS XRI Technical Committee

XRI ist ein System zur Identifizierung und Auflösung von Objekten, unabhängig von ihrem physischen Speicherort oder bestimmten Protokollen. XRIs eignen sich besonders gut für Querverweise. Man kann XRIs als ein DNS für URIs betrachten. XRIs sind mit URIs und IRIs kompatibel.¹²

Syntax

XRI = xri://<Autorität>/<Pfad>[?<Anfrage>] [\#<Fragment>]

Ein in der URI-Normalform notierter XRI besteht aus der Angabe einer Autorität und eines Pfades. Die Autorität entspricht dabei grob dem NMAH-Teil von einem ARK. Der Pfad entspricht der Syntax von URI-Pfaden, wo jede Strukturebene durch einen „/“ voneinander getrennt ist.

Mit dem Fragezeichen kann eine Anfrage an den XRI angefügt werden. Mit einem Doppelkreuz werden Fragmente eingeleitet. Die Syntax für Anfragen und Fragmente entspricht der von IRIs. Zusätzlich gibt es eine Reihe von reservierten Zeichen, die bestimmte Zwecke erfüllen.

Die Syntaxspezifizierung sieht sowohl persistente als auch austauschbare Segmente in einem XRI vor.

Eigenschaften

	persistent	actionable	flexibel	Metadaten	opak	Autorität	Kosten
XRI	●	●	●	○	●	●	\$

XRIs sind nur persistent, wenn man ausschließlich persistente Bestandteile verwendet. Auch XRIs können URL-fiziert werden, indem die Adresse eines XRI-Resolvers vorangestellt wird. Sie können vollkommen flexibel vergeben werden.

Über den Umgang mit Metadaten ist in den Spezifikationen nichts angegeben, weshalb diese auch nicht von dem System gefordert werden.

Opak sind XRIs nur, wenn auch opake Bestandteile für die Autorität und den Pfad verwendet werden. Die Autorität des Systems hängt von dem Anbieter ab.

Durch ihre Gestaltung, die Querverweise und Weiterverwendung von Informationsteilen ermöglicht, sind XRIs interoperabel.

¹²Der Internationalized Resource Identifier (internationaler Ressourcenidentifikator) verwendet im Gegensatz zu einem URI jedes Zeichen aus dem Universal Character Set. Siehe: <http://www.ietf.org/rfc/rfc3987>

Verbreitung

Die Nutzerbasis von XRIs ist relativ beschränkt und auch deren Aktivität hat ihren Höhepunkt (2008) lange überschritten. Außerdem wurden die Spezifikationen von XRI 2.0 durch OASIS abgelehnt.¹³

Spezifikation und Quellen

https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xri
<http://docs.oasis-open.org/xri/xri-syntax/2.0/specs/cs01/xri-syntax-V2.0-cs.html>
<http://equalsdrummond.name/2008/09/06/xri-in-a-nutshell/>

6.1.10 OpenURL

Seit 1999, Herbert Van de Sompel, Universität Ghent
Seit 2006 von OCLC entwickelt
Jetzt ein NISO Standard (Z39.88-2004)

Es handelt sich bei OpenURL *nicht* um einen persistenten Identifikator im eigentlichen Sinne, sondern eher um ein Transportprotokoll für Metadaten (*metadata transport protocol*).

Der Grundgedanke ist, dass Nutzer mittels Links zu den für sie geeigneten Ressourcen geleitet werden sollen. Dabei wird die Auswahl mittels der transportierten Metadaten getroffen.

Syntax

OpenURL = <URL>?<Anfrage> [&<Anfrage>]

Ein OpenURL ist im Prinzip ein normaler einfacher URL mit einer Anfrage (*query*), deren Anfang durch ein vorangestelltes Fragezeichen gekennzeichnet wird. Die Anfrage kann mehrteilig sein indem die einzelnen Teile durch ein Et-Zeichen (&) getrennt werden.

Beispiele

OpenURL, Version 0.1:

```
http://resolver.example.edu/cgi?genre=book&isbn=0836218310
|                                     |
|-----|                             |-----|
|                                     |
einfacher URL                         Anfrage
```

Es wird hierbei nach einem Buch mit der ISBN 0836218310 gefragt.

Die Anfrage wird bei OpenURL, Version 1.0, deutlich länger:

```
?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:book&rft.isbn=0836218310
|-----| / |-----| / |-----| /
|                                     |                                     |
URL-Version                         individuelle Metadaten                     spezielles Objekt
```

¹³<https://lists.oasis-open.org/archives/xri/200806/msg00001.html>

Verbreitung

Da es sich bei OpenURLs nicht um PIDs im eigentlichen Sinne handelt, wird deren Verbreitung an dieser Stelle nicht von angegeben.

Spezifikation und Quellen

<http://www.exlibrisgroup.com/category/sfxopenurl>
<http://listserv.oclc.org/scripts/wa.exe?Ao=OPENURL>
<http://library.caltech.edu/openurl/Bibliography.htm>
http://www.niso.org/apps/group_public/project/details.php?project_id=82
 Demo: <http://www.ukoln.ac.uk/distributed-systems/openurl/>

6.1.11 Zusammenfassung

Eine Übersicht der Eigenschaften von PID-Systemen ist in Tabelle 1 zusammengefasst dargestellt.

	persistent	actionable	flexibel	Metadaten	opak	Autorität	Kosten
URI	○	◐	●	○	◐	○	\$
URL	◐	●	●	○	◐	◐	\$
URN	◐	◐	◐	◐	◐	◐	\$
PURL	◐	●	●	○	◐	◐	\$
URN:NBN	◐	◐	◐	●	●	●	\$
Handle	◐	◐	●	●	●	●	\$
DOI	◐	◐	●	●	●	●	\$\$\$
ARK	◐	◐	●	●	●	●	\$
XRI	◐	◐	●	○	◐	◐	\$

Tabelle 1: Eigenschaften der vorgestellten PID-Systeme.

6.2 Wo wird welches PID-System verwendet?

Eine Rolle bei der Entscheidung für ein PID-System spielt auch, welche Systeme von ähnlichen Einrichtungen verwendet werden. In der unten aufgeführten Tabelle 2 sind alle Einrichtungen aufgeführt, die von IANUS, teilweise auch im Rahmen von Expertengesprächen, näher untersucht wurden.

In der Tabelle werden URIs, URLs und URNs nicht berücksichtigt, da sie zu allgemein sind und auch nicht von den Einrichtungen als PID-Systeme verwendet

werden. OpenURLs werden nicht aufgeführt, da es sich dabei nicht um ein PID-System im eigentlichen Sinne handelt. Einrichtungen, die ein eigenes System verwenden, wurden in der Tabelle nicht berücksichtigt.

	DANS - EDNA	Archaeology Data Service	GESIS	MPI - The Language Archive	PANGAEA	tDAR	OpenContext	Digitales Archiv NRW
PURL								
URN:NBN	●							●
Handle (HS)				●				
DOI		●	●		●	●	●	
ARK							●	
XRI								

Tabelle 2: Verwendung von PID-Systemen.

Es wird deutlich, dass DOIs am häufigsten verwendet werden, da sie von fünf verschiedenen Einrichtungen genutzt werden. NBNs und ARKs werden jeweils von zwei Einrichtungen verwendet, während sich eine Einrichtung für die Verwendung des Handle Systems entschieden hat. Die Dominanz von DOIs spiegelt sich auch in der Anzahl der weltweit registrierten DOIs im Vergleich zu den anderen PID-Systemen wider.

Im Folgenden werden die in Tabelle 2 dargestellten Einrichtungen und ihr jeweils gewähltes PID-System näher beschrieben.

6.2.1 DANS - EDNA

Das e-depot voor de Nederlandse archeologie (EDNA) verwendet URN:NBNs. Es gehört dem DANS (Data Archiving and Networked Services) an, welches wiederum Teil der niederländischen Akademie der Wissenschaften (KNAW) und der niederländischen Organisation für Forschung (NWO) ist. Die Gründe für die Entscheidung dafür waren, dass sie kostenlos sind, in unbegrenzter Anzahl vergeben werden können, eigenständig von vergänglichen Infrastrukturen sind und DANS genügend Kontrolle über den Namensraum hat.

Die NBNs von DANS haben die Form: `urn:nbn:nl:ui:13-<NBN-String>`, wobei der NBN-String sich aus zwei, durch einen Bindestrich voneinander getrennten, dreistelligen Ziffern- und Buchstabenblöcken zusammensetzt.

Eine Suchanfrage mit „`urn:nbn:nl:ui:13`“ auf der Seite von DANS¹⁴ ergab 25.783 Treffer, was schließen lässt, dass ebenso viele URN:NBNs für DANS registriert sind,

¹⁴<https://easy.dans.knaw.nl/ui/home> aufgerufen am 17.9.2013.

wovon ein Teil auch EDNA zuzuschreiben ist. EDNA vergibt PIDs für jede Publikation und für jedes Projektarchiv, verwendet also eine niedrige Granularität.

6.2.2 Archaeology Data Service

Bei dem Archaeology Data Service (ADS) in York (GB) werden DOIs mit dem Präfix 10.5284 und einem Suffix mit einer zufälligen siebenstelligen Zahlenkombination verwendet. Partner für die Registrierung von DOIs ist die British Library, die Mitglied von DataCite ist. Für das ADS sind 21.167 DOIs registriert, wie eine Anfrage auf DataCite ergab.¹⁵

Dabei bekommen beim ADS sogenannte *Collections*¹⁶ und sogenannte *Graue Literatur* jeweils eigene PIDs, während die dazugehörigen Dateien lediglich auf der *Landing Page* gelistet werden.

Die Entscheidung für DOIs wurde vom ADS aufgrund der Bereitstellung derselben durch die British Library gefällt, wodurch den DOIs ein gutes Maß an Autorität gegeben wird. Außerdem bietet die British Library eine nachhaltige Infrastruktur für die Verwendung von DOIs und es bestand dort bereits eine Gemeinschaft von anderen namhaften Institutionen, die bereits DOIs verwendeten.¹⁷

6.2.3 GESIS

Das Datenarchiv für Sozialwissenschaften innerhalb von dem GESIS –Leibniz-Institut für Sozialwissenschaften verwendet DOIs mit dem Präfix 10.4232, die von da|ra registriert werden. Da|ra ist Mitglied von DataCite und fungiert als Registrierungsagentur für alle Sozial- und Wirtschaftswissenschaften.

Einer der wichtigsten Gründe bei der Entscheidung von GESIS für DOIs war, dass diese schon im Rahmen der Zitierung von Zeitschriftenartikeln für die Forscher bekannt waren und somit die Bereitschaft diese für das Zitieren von digitalen Datensätzen zu verwenden, möglicherweise höher wäre. Weiterhin bot die Mitgliedschaft bei DataCite die Chance, einer bereits bestehenden starken Gemeinschaft zur Verwendung von DOIs beizutreten. Die verschiedenen Anforderungen von DataCite führen außerdem zu einer hohen Qualität der DOIs, wenn auch etwas mehr Arbeit erforderlich ist.¹⁸

Die Registrierungsagentur da|ra hat bisher 8.177 DOIs registriert,¹⁹ wovon 5.915 allein GESIS zugeschrieben werden können.²⁰ Bei GESIS bekommen hauptsächlich Studien²¹ eigene PIDs. Einzelnde dazugehörige Dateien sind über die *Landing Page* aufrufbar. Neue Versionen alter Studien erhalten jeweils einen neuen PID.

¹⁵http://stats.datacite.org/?fq=allocator_facet%3A%22BL++The+British+Library%22&q=#tab-datacentres aufgerufen am 17.9.2013.

¹⁶z.B. doi:10.5284/1000186

¹⁷Aus einer E-Mail vom 18. September 2013.

¹⁸Aus einer E-Mail vom 23. September 2013

¹⁹<http://stats.datacite.org/?q=> aufgerufen am 17.9.2013.

²⁰http://stats.datacite.org/?fq=allocator_facet%3A%22GESIS+-GESIS+-Leibniz+Institute+for+the+Social+Sciences%22&q=#tab-datacentres aufgerufen am 17.9.2013

²¹z.B. doi:10.4232/1.10028

6.2.4 MPI - The Language Archive

Das Spracharchiv des Max Planck Instituts für Psycholinguistik (MPI - The Language Archive) verwendet das Handle System, da es sich um ein bewährtes und skalierbares System handelt, das nicht von den Risiken eines Geschäftsmodells, wie das der IDF, abhängt. Insbesondere wegen der hohen Kosten für DOIs und da das Spracharchiv eine sehr hohe Granularität und entsprechend viele PIDs benötigt, fiel die Wahl auf das Handle System, das lediglich eine geringe jährliche Gebühr für das Präfix verlangt.²²

Das von dem Spracharchiv verwendete Präfix lautet 1839. Die Menge an registrierten PIDs konnte nicht ermittelt werden, jedoch wird jede Datei mit einem solchen versehen.²³

6.2.5 PANGAEA

PANGAEA ist ein Publikationsagent und Archiv für Daten der Erd- und Umweltwissenschaften, das seit 2009 DOIs verwendet. Das Vorgängersystem war sRef (Society Reference Catalogue). DOIs von PANGAEA haben die Form 10.1594/PANGAEA.X, wobei das 'X' eine Ziffernfolge repräsentiert. Das Suffix entspricht dabei auch der internen ID, die jedem Datensatz von dem Archivsystem in PANGAEA zugewiesen wird. PANGAEA begründet die Entscheidung für DOIs damit, dass dieses PID-System im Verlagswesen sehr bekannt und etabliert ist und auf eine hohe Akzeptanz bei den Nutzern stößt.

Für PANGAEA wurden bisher 682.902 DOIs registriert, wobei dabei auch Partnerinstitutionen zu berücksichtigen sind.²⁴

PANGAEA vergibt einen Identifikator pro Datensatz²⁵, der zumeist aus einzelnen Messdaten oder Messreihen und den zugehörigen Metadaten besteht. Für Publikationen oder Sammlungen, die sich auf mehrere Datensätze beziehen, können weitere Identifikatoren vergeben werden.²⁶

6.2.6 tDAR

The Digital Archaeological Record (tDAR) verwendet hauptsächlich einen eigenen Identifikator (TDAR ID), jedoch werden teilweise auch DOIs vergeben, die über Digital Antiquity als Betreiber von tDAR registriert werden. Digital Antiquity hat das DOI-Präfix 10.6067 und ist Teil der California Digital Library.

Für Digital Antiquity wurden bereits 25.091 DOIs registriert, wobei nicht klar ist, wie viele davon tDAR zuzuordnen sind.²⁷

²²Aus einer E-Mail vom 19. September 2013.

²³z.B. hdl:1839/00-0000-0000-000F-E34E-A

²⁴ <http://stats.datacite.org/?q=#tab-allocator> aufgerufen am 17.9.2013. Durch Klicken auf *TIB - German National Library of Science and Technology > TIB.PANGAEA > PANGAEA - Publishing Net... > 10.1594* bekommt man eine Liste der beteiligten Einrichtungen.

²⁵z.B. doi:10.1594/PANGAEA.747338

²⁶z.B. doi:10.1594/PANGAEA.611088

²⁷ http://stats.datacite.org/?fq=allocator_facet%3A%22CDL+-+California+Digital+Library%22&q=#tab-datacentres aufgerufen am 17.9.2013.

6.2.7 OpenContext

OpenContext hat von Anfang an ARKs verwendet, die von EZID registriert wurden. Inzwischen werden auch DOIs verwendet. Dabei bekommt jedes individuelle Objekt ein ARK zugewiesen, während DOIs für größere Datensammlungen oder -sets vergeben werden, da sie für OpenContext nicht kostenlos zur Verfügung stehen und den Entwicklern dort nicht ganz klar ist, wie sich das DOI-System bei einer großen Anzahl von einzelnen PIDs verhält.²⁸

Die DOIs von OpenContext haben das Präfix 10.6078 und werden von der UC Berkeley registriert, die wiederum Mitglied bei der California Digital Library ist. Bisher wurden 82 DOIs mit dem Präfix registriert.²⁹

6.2.8 Digitales Archiv NRW

Das Digitale Archiv NRW (DA-NRW) verwendet URN:NBNs als PID-System. Dabei bekommt jedes Archivinformationspaket (archival information package, AIP) anfangs eine NBN zugewiesen, die aber erst dann bei der Deutschen Nationalbibliothek registriert wird, wenn auch ein Präsentationsinformationspaket (presentation information package, PIP) erzeugt wird. Der Umfang eines AIPs hängt von dem Datengeber ab. NBNs von DA-NRW beginnen mit urn:nbn:de:danrw.

Die Grundlagen der Entscheidung von DA-NRW für URN:NBNs sind eine Auswahl von Kriterien von Nestor³⁰ und wohl auch das Mitwirken von mehreren Universitäts- und Landesbibliotheken in dem Projekt.

6.3 Eingrenzung der Auswahl für IANUS

Aus der Übersicht der PID-Systeme und deren in Tabelle 2 gezeigten Verwendung in ähnlichen Einrichtungen lässt sich eine engere Auswahl der für IANUS in Frage kommenden PID-Systeme erstellen, die hier nach Anzahl der verwendenden Einrichtungen aufgelistet ist:

- DOI
- URN:NBN
- ARK
- Handle (HS)

In dem nächsten Abschnitt werden relevante Dienste für diese vier PID-Systeme untersucht, deren Leistungen für IANUS zukünftig relevant sein könnten.

7 Relevante Einrichtungen und Dienste

7.1 Für URN:NBN

²⁸Aus einer E-Mail vom 22. August 2013.

²⁹http://stats.datacite.org/?fq=allocator_facet%3A%22CDL+++California+Digital+Library%22&q=#tab-datacentres aufgerufen am 17.9.2013.

³⁰Thaller (2013), S. 46-47

7.1.1 Deutsche Nationalbibliothek

Um URN:NBNs in Deutschland verwenden zu können, muss man sich an die Deutsche Nationalbibliothek wenden, die dafür zuständig ist. Diese bietet den URN-Service³¹ an, wo alle notwendigen Informationen bereitgestellt werden.

Um als eigenständige Registrierungsagentur für NBNs aufzutreten, wird ein eigener Unternehmensraum benötigt, der bei der Deutschen Nationalbibliothek beantragt wird. Es müssen dazu einige Grundvoraussetzungen erfüllt werden, die in der „Policy für die Vergabe von URNs im Namensraum urn:nbn:de“³² festgelegt sind.

Wichtigste Kriterien hierfür sind die Langzeitarchivierung zertifiziert nach dem Data Seal of Approval, eine ausreichend große öffentliche Einrichtung, die regelmäßig in großer Stückzahl publiziert, und geeignete Objektarten.

Geeignete Objekte sollten inhaltlich eine geschlossene Einheit bilden und archivierbar sein. In der Policy der DNB werden hier vor allem klassische Publikationsformen genannt, wie Monografien, Zeitschriftenartikel und archivierte Webseiten. Insbesondere für Rohdaten wird erwähnt, dass diese nur eingeschränkt geeignet sind und es noch keinen konkreten Anwendungsfall gibt.

Der Resolverdienst der DNB kann auf <http://nbn-resolving.org/> aufgerufen werden und löst auch Schweizer NBNs auf. Außerdem kann der Resolver mit anderen europäischen NBNs, sowie mit DOIs, Handles und ARKs umgehen.

7.2 Für Handle (HS)

7.2.1 Global Handle Registry

Zur Registrierung von Handles mit einem eigenen Präfix, muss das Präfix bei der Global Handle Registry³³ beantragt und registriert werden. Es fällt dabei ein jährlicher Beitrag von 50 an. Eine Liste der bereits registrierten Präfixe und der dahinterstehenden Einrichtungen ist nicht einsehbar, wodurch die Autorität für spätere Nutzer verschleiert wird, was eventuell zu Einbußen in der Vertrauenswürdigkeit der Registrierungsagentur führt.

Wenn man sich nicht für eine bestehende Einrichtung (mit einem bestehenden Präfix) entscheidet, kann man auch ein eigenes Handle-System aufsetzen, muss dieses aber auch eigenständig warten.

7.2.2 EPIC

Das 2009 gegründete European Persistent Identifier Consortium (EPIC)³⁴ bietet auf dem Handle System basierende PID-Dienste für die europäische Forschungsgemeinschaft an.

Dazu soll auch ein gut dokumentiertes Softwarepaket gehören, das Registrierungsagenturen zur Verfügung gestellt wird. Aktuell wird aber noch daran gearbeitet.³⁵ Für den Dienst von EPIC gibt es bereits eine API, während ein Demo-System noch in Planung ist.

³¹<http://www.dnb.de/urnservice>

³²<http://nbn-resolving.de/urn:nbn:de:101-2012121200>

³³<http://www.handle.net/introduction.html>

³⁴<http://www.pidconsortium.eu>

³⁵http://www.pidconsortium.eu/index.php?page=service_for_providers

7.2.3 DARIAH-DE

Mitglied bei EPIC ist auch das Projekt DARIAH-DE, das in Zusammenarbeit mit der GWDG den PID Service von EPIC verwendet.³⁶ IANUS bekam von DARIAH-DE die Möglichkeit diesen Service zu verwenden und Handles zu registrieren. In diesem Fall handelt es sich nicht um eine Testumgebung, sondern um regulär registrierte PIDs. Ausführliches dazu ist in Anhang C zu finden.

7.3 Für DOI

Da DOIs von der International DOI Foundation verwaltet werden, dürfen diese auch nur von Mitgliedern der IDF vergeben werden, weshalb man entweder einen entsprechenden Dienst in Anspruch nehmen oder selbst Mitglied werden muss. Aktuell gibt es neun Registrierungsagenturen, die Mitglied der IDF sind.³⁷

Die einzelnen Agenturen haben inhaltliche Schwerpunkte oder regionalen Bezug und haben teilweise selbst wiederum kleinere Registrierungsagenturen als Mitglieder.

7.3.1 DataCite

DataCite³⁸ ist eine 2009 in London gegründete Registrierungsagentur, deren Ziele es sind den Zugriff auf Forschungsdaten zu erleichtern, das Ansehen der Forschungsrohdaten zu steigern und die Langzeitarchivierung dieser Daten zu unterstützen.³⁹ DataCite ist Mitglied der IDF und nimmt an EPIC teil.⁴⁰ Aktuell ist die Technische Informationsbibliothek in Hannover die bevollmächtigte Einrichtung von DataCite.

Unter den Mitgliedern von DataCite befinden sich vier deutsche Einrichtungen, welche die Fachbereiche Technik und Naturwissenschaften, Medizin, Sozial- und Wirtschaftswissenschaften unter sich aufteilen. Diese Einrichtungen haben auch beschlossen, dass DOIs ab 2013 für akademische Einrichtungen kostenlos bleiben.⁴¹

Zu den verschiedenen Informationsangeboten und Ressourcen von DataCite gehört auch ein Statistik-Tool, das es ermöglicht, Nutzerstatistiken für bestimmte DOI-Präfixe zu ermitteln.⁴²

7.3.2 da|ra

Die Registrierungsagentur für Sozial- und Wirtschaftsdaten da|ra⁴³ ist Mitglied von DataCite und wird gemeinsam von GESIS und der ZBW betrieben.

Da die Agentur ihren Sitz in Berlin hat, bot sich für IANUS die Gelegenheit, vor Ort ein intensives Gespräch zur Klärung von Fragen zu führen. Dabei wurde IANUS auch eine Testumgebung zur Verfügung gestellt. In Anhang D ist ein Testbericht darüber zu finden.

³⁶<https://de.dariah.eu/web/guest/pid-service>

³⁷http://www.doi.org/registration_agencies.html aufgerufen am 18.9.2013

³⁸<http://www.datacite.org/>

³⁹<http://www.datacite.org/faqs#background>

⁴⁰www.doi.org/topics/EPIC_DataCite_March2012.pdf

⁴¹<http://www.tib-hannover.de/de/die-tib/aktuelles/aktuelles/id/362/>

⁴²<http://stats.datacite.org/>

⁴³<http://www.da-ra.de/>

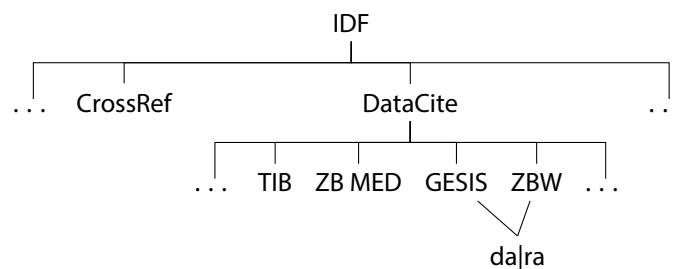


Abbildung 4: Position von da|ra innerhalb der Hierarchie der IDF.

7.3.3 Citation Formatter

Ein praktisches Tool für die spätere Arbeit mit bereits registrierten DOIs ist der *Citation Formatter*⁴⁴, der für einen bestimmten DOI eine Zitierung in über 100 verschiedenen Zitierstilen ausgeben kann.

7.3.4 Content Negotiation

Um dem Nutzer die Möglichkeit zu bieten, individuell die Repräsentationsform eines mit einer DOI registrierten Objektes zu wählen, kann *content negotiation* (Inhaltsvereinbarung) verwendet werden. Auf diese Weise kann der Nutzer beispielsweise wählen, ob ein Objekt oder dessen Metadaten als Webseite, PDF-Dokument oder zip-Datei angezeigt werden sollen.

Zusammen mit CrossRef bietet DataCite Content Negotiation⁴⁵ an. Für die einzelnen Repräsentationsformen muss aber jeweils der passende Speicherort (also der URL) angegeben werden.

7.4 Dienste für ARK

7.4.1 California Digital Library

Um ARKs vergeben zu können, muss eine NAAN bei der California Digital Library⁴⁶ beantragt werden. Bis auf die Voraussetzung, dass die Einrichtung stabil sein sollte, werden offiziell keine weiteren Vorgaben gemacht.⁴⁷

Es steht einem anschließend auch frei, eine eigene Instanz der ARK-Software zu programmieren oder bereits bestehende Dienste anderer Einrichtungen zur Verwaltung der ARKs zu verwenden.

7.4.2 EZID

Die CDL bietet den Dienst EZID⁴⁸ zur Registrierung und Verwaltung von ARKs an. Mit diesem Dienst könnten auch DOIs verwaltet werden. Es gibt eine API und zu

⁴⁴<http://crosscite.org/citeproc/>

⁴⁵<http://crosscite.org/cn/>

⁴⁶<http://www.cdlib.org>

⁴⁷<https://confluence.ucop.edu/display/Curation/ARK>

⁴⁸<http://nzt.net/ezid/>

Testzwecken wird eine Testumgebung zur Verfügung gestellt.⁴⁹ In Anhang E ist der entsprechende Testbericht zu finden.

7.5 Weitere Dienste

7.5.1 N2T

N2T (*Name to thing*) ist ein Resolver für persistente Identifikatoren, der von einem Konsortium von Organisationen des kulturellen Gedächtnisses (cultural memory organizations) betrieben wird.⁵⁰

Es handelt sich um einen (mehrfach gespiegelten) Webserver, der einfache HTTP redirects ausführt. Er dient dazu persistente Identifikatoren auf den Ort ihres aktuellen Anbieters aufzulösen. N2T kann mit allen Schemata umgehen, die als URL ausgedrückt werden können, wie etwa Handle, DOI, URN, PURL, ARK, usw.

Dieser Dienst kann zur URL-fizierung verwendet werden, indem die Adresse von N2T vor den PID gestellt wird, wie zum Beispiel:

```
http://n2t.info/doi:10.1000/182  
oder  
http://n2t.info/ark:/13030/ft4w10060w
```

Der Dienst wird auf http://www.n2t.info/d/n2t_vision.html genauer beschrieben. Auch an dieser Stelle sei erneut darauf hingewiesen, dass URL-fizierung nicht zu empfehlen ist, wie schon in Abschnitt 5.1 diskutiert wurde.

7.5.2 Browser Plugins

PIDs können komfortabel wie normale Links aufgelöst werden, wenn der Browser entsprechende Plugins hat. In dem Plugin ist hinterlegt, welcher Resolverdienst verwendet wird und das Programm erkennt dann selbstständig ob und welche PIDs in einem Dokument auftreten. Die PIDs können dann wie normale URL-Links angeklickt werden.

Aktuell gibt es ein Plugin zur Auflösung von DOIs für Google Chrome.⁵¹ Für Firefox existiert ein Plugin, das von der CNRI entwickelt wurde, und mit Handles und DOIs umgehen kann.⁵² CNRI stellt ebenfalls den Handle URI Resolver für Microsoft Internet Explorer (Versionen 3 bis 7) zur Verfügung.⁵³ Wie das Plugin für Firefox, kann dieser Resolver mit Handles und DOIs umgehen.

PANGAEA hat ein kleines Programm namens PanDOI entwickelt, das DOIs, Handles, URNs und sRef IDs auflösen kann. Es handelt sich dabei um ein Desktop-Programm für Windows Vista und höher. Für Mac OS X gibt es ebenfalls eine Version.⁵⁴

Für die übrigen PID-Systeme und Browser gibt es momentan keine entsprechenden Zusatzfunktionen.

⁴⁹<http://n2t.net/ezid/demo>

⁵⁰www.n2t.info

⁵¹<https://chrome.google.com/webstore/detail/doi-resolver/goanbaknlbojfglcepinankooabfakbpg?hl=en-GB>

⁵²http://www.handle.net/hs-tools/extensions/firefox_hdlclient.html

⁵³<http://www.handle.net/resolver/>

⁵⁴<http://doi.pangaea.de/10.1594/PANGAEA.759947>

8 Empfehlungen für IANUS

Nachdem die verschiedenen Aspekte und Fragen zu PID-Systemen besprochen wurden, können an dieser Stelle Empfehlungen gemacht und die zwei in der Spezifikation des Testbeds gestellten Fragen beantwortet werden.

8.1 Welches PID-System für IANUS geeignet ist

Aus der in Abschnitt 6.3 getroffenen Auswahl von PID-Systemen lässt sich anhand der besprochenen Dienste, ausgeführten Tests und der Verbreitung der jeweiligen Systeme eine klare Empfehlung für IANUS aussprechen.

URN:NBNs eignen sich nicht für die Verwendung im zukünftigen Forschungsdatenzentrum, da die DNB in ihrer Policy darauf hinweist, dass Rohdaten bisher noch nicht mit NBNs registriert wurden und auch nur Ausnahmefälle darstellen sollen. Diese werden aber voraussichtlich den Hauptbestandteil der Daten in IANUS ausmachen.

Gegen die Verwendung von ARKs spricht die recht geringe Verbreitung und der Umstand, dass ein Ansprechpartner dafür aktuell nur in Californien gefunden werden kann, obwohl der EZID-Dienst gut zu benutzen wäre.

Die Verwendung des Handle Systems würde theoretisch einen großen Funktionsumfang bieten, jedoch erfordert dies einen hohen zusätzlichen Aufwand, da viele Zusatzfunktionen selbst programmiert werden müssten. Ein Vergleich von dem Service der GWDG (Anhang C) mit den Diensten von daJra oder EZID (Anhänge D und E) verdeutlicht diesen Umstand. Es würden zusätzliche langfristige Kosten für die Programmierung und Wartung des Systems entstehen.

Dementsprechend kann für IANUS das DOI-System empfohlen werden. Hierfür spricht vor allem die weite Verbreitung und der inzwischen relativ hohe Bekanntheitsgrad, wie es in Abschnitt 6.2 dargelegt wurde. Auch andere Einrichtungen, die Forschungsdaten langfristig archivieren und bereitstellen, verwenden bereits DOIs und haben hier Erfahrungen gesammelt.

Inzwischen sind DOIs im Verlagswesen so verbreitet, dass auch Altertumswissenschaftler schon in Berührung mit diesem System gekommen sein dürften. Ein bereits etabliertes System fördert das Vertrauen des zukünftigen Nutzers. Glücklicherweise spielen die Kosten für dieses System zumindest in Deutschland aktuell keine Rolle.

Hinzu kommt, dass durch die hierarchische Struktur der Agenturen mit dem IDF an der Spitze, nicht jeder beliebige Nutzer DOIs registrieren kann und auch tatsächlich alle registrierten DOIs über die DOI-Resolver auflösbar sind. Mit dem von daJra bereitgestellten Dienst gibt es außerdem einen kompetenten und flexiblen Anbieter als Registrierungsagentur für IANUS.

8.1.1 Das Suffix

Wenn DOIs verwendet werden, wird das Präfix die Form „10.xxxxx“ haben. Die fünfstelligen Ziffer wird von der Registrierungsagentur vorgegeben. Das Suffix kann von IANUS selbst gestaltet werden. Es sollte keine semantischen Informationen preisgeben und keine Sonderzeichen oder Umlaute verwenden.

An dem Ende des Suffixes sollte eine Prüfziffer stehen, die durch einen Bindestrich von dem vorhergehenden Teil getrennt wird. Die Prüfziffer soll mit einem

Algorithmus aus dem Präfix und dem Suffix (Prüfziffer ausgenommen) errechnet werden, wie es etwa auch bei ISBN- oder IBAN-Nummern üblich ist.

Das Suffix könnte zweigeteilt sein und aus einem Kürzel für einen Datengeber und der Zeichenkombination für den Datensatz bestehen. Beides mit einem Punkt voneinander getrennt. Das Kürzel für den Datengeber sollte nicht länger als drei Zeichen sein und darf nur Zahlen oder Buchstaben enthalten. Ein Name darf nicht direkt daraus ablesbar sein, da man davon ausgehen muss, dass Einrichtungen ihren Namen ändern oder aufgelöst werden können.

Die Zeichenkombination für den Datensatz könnte einer ID entsprechen, die in dem Archivsystem von IANUS verwendet wird und auch die Landing Page benennt. Es darf sich um eine beliebige Buchstaben- und Zahlenkombination handeln, die idealerweise acht Zeichen lang ist.

Berücksichtigt man diese Vorschläge, würde ein DOI von IANUS folgende Form haben: „10.xxxxx/xxx.xxxxx-x“

8.2 Praktische Anwendungshinweise

Generell soll jede Datensammlung, die über IANUS archiviert und bereitgestellt wird, einen eigenen Identifikator erhalten, der zugewiesen wird, sobald die Daten übergeben wurden. Die Registrierung des Identifikators soll aber erst erfolgen, wenn die Daten für die Archivierung und Bereitstellung vorbereitet wurden und der Datengeber seine Zustimmung gegeben hat. Dadurch besteht die Möglichkeit noch eventuelle Verbesserungen an der Datensammlung durchzuführen. Werden Veränderungen nach der Registrierung durchgeführt, so muss in der Regel eine neue Version mit einem neuen PID registriert werden. Ausnahmen bilden hierbei Konzepte.

Für die Granularität sind die in Anhang A formulierten Gedanken zu berücksichtigen. Das Beantragen von PIDs für alle einzelnen Objekte oder Dateien, die Teil einer größeren Datensammlung sind, die bereits einen PID besitzt, könnte sich nur eingeschränkt als praktikabel erweisen. Es scheint aber sinnvoll, dem Nutzer eine Möglichkeit zur Verfügung zu stellen, mit der er einzelne neue PIDs beantragen kann. Wichtig ist auf jeden Fall, dass für jedes digitale Objekt der passende Zitierhinweis in der Weboberfläche gut sichtbar platziert wird.

Eine Landing Page wird dringend empfohlen. Aus Bibliotheksaktalogen ist dieses Konzept hinreichend bekannt und eine übersichtliche Darstellung der Informationen zu dem aufgerufenen Objekt ermöglicht es dem Nutzer zu überprüfen, ob das Objekt den Erwartungen entspricht. Auch können hier der Zitierhinweis, Hinweise auf andere Versionen oder verwandte Objekte und weitere Hinweise platziert werden. Gegebenfalls kann die *Landing Page* in eine *Tombstone Page* umfunktioniert werden. Mehr dazu in Anhang B.

Die Metadaten sollten generell uneingeschränkt sichtbar sein. Einzige Ausnahmen bilden hierbei personenbezogene Daten und, wenn notwendig, auch geografische Angaben. Teil der Metadaten können auch Prüfsummen sein, um dem Nutzer eine weitere Möglichkeit zur Überprüfung des Objektes zu geben. Prüfsummen eignen sich aber nur für in sich abgeschlossene einzelne Dateien.

Zukünftige Nutzer sollten die Möglichkeit haben, defekte PIDs zu melden und bei Fehlern eine informative Fehlermeldung zu erhalten. Dies stärkt die Vertrauenswürdigkeit und sichert die Qualität des Dienstes. Ideal wäre es, wenn IANUS anhand der Prüfziffer den PID auf Richtigkeit überprüfen kann und dem Nutzer

mitteilen kann, falls ein Zahlendreher bei der Eingabe erfolgte, anstatt einfach eine Fehlermeldung zurückzugeben.

In der Praxis könnte sich die Verwendung eines Datumsfeldes als nützlich erweisen, in das ein Ablaufdatum oder ein Datum zur Wiedervorlage eingetragen werden kann. Auf diese Weise könnte das System eigenständig auf Metadatensätze hinweisen, bei denen etwa die URL oder die Zugriffsrechte überarbeitet werden müssen.

Letztendlich ist es wichtig zu wissen, dass PIDs lediglich eine Hilfestellung zur Referenzierung darstellen. Die Verwendung von PIDs nimmt einem nicht die Arbeit ab, die referenzierten Objekte zu warten und zugänglich zu machen.

9 Weitere Quellen

Nestor, Persistent Identifier: <http://www.langzeitarchivierung.de/Subsites/nestor/DE/Standardisierung/PI.html>

SURF: <http://wiki.surf.nl/display/PersistentIdentifier/EN>

Deutsche Nationalbibliothek, Persistent Identifier: http://www.dnb.de/DE/Standardisierung/PI/pi_node.html

TWR, PIDs: <http://metadaten-twr.org/2010/10/13/persistent-identifiers-an-overview/>

DANS: <http://www.dans.knaw.nl/en/content/categorieen/diensten/persistent-identifiers>

DCC, PIDs: <http://www.dcc.ac.uk/resources/briefing-papers/introduction-curation/persistent-identifiers>

Knowledge Exchange: <http://www.knowledge-exchange.info/Default.aspx?ID=332>

PANGAEA: http://wiki.pangaea.de/wiki/Persistent_identifier

Diskussion um ORCID: <http://www.offene-wissenschaft.de/tag/pangaea/>

ARIADNE: <http://www.ariadne.ac.uk/issue56/tonkin>

EPIC, PIDs, Kalman: hdl.handle.net/11858/00-ZZZZ-0000-0001-6D1D-0

Hans-Werner Hilse –Jochen Kothe (2006) Implementing Persistent Identifiers. Overview of concepts, guidelines and recommendations.

Kunze (2003) –John A. Kunze (2003) Towards Electronic Persistence Using ARK Identifiers.

Nestor (2009) –Niklaus Bütikofer (2009) Kriterienkatalog zur Prüfung der Vertrauenswürdigkeit von PI-Systemen. Hrsg. von der nestor Arbeitsgruppe Standards für Metadaten, Transfer von Objekten in digitale Langzeitarchive und Objektzugriff. (urn:nbn:de:0008-20080710140)

Thaller (2013) –Manfred Thaller (2013) Das Digitale Archiv NRW in der Praxis.

10 Abkürzungsverzeichnis

ADS	-	Archaeology Data Service
API	-	Application Programming Interface
ARK	-	Archival Resource Key
CDL	-	California Digital Library
CNRI	-	Corporation for National Research Initiatives
DA-NRW	-	Digitales Archiv NRW
DANS	-	Data Archiving and Networked Services
DARPA	-	Defense Advanced Research Projects Agency
DNB	-	Deutsche Nationalbibliothek
DNS	-	Domain Name System
DOI	-	Digital Object Identifier
EDNA	-	e-Depot voor de Nederlandse Archeologie
EPIC	-	European Persistent Identifier Consortium
ERC	-	Electronic Resource Citation
HTTP	-	Hypertext Transfer Protocol
IANA	-	Internet Assigned Numbers Authority
IDF	-	International DOI Foundation
IETF	-	Internet Engineering Task Force
MPI	-	Max Planck Institut
NAAN	-	Name Assigning Authority Number
NBN	-	National Bibliographic Numbers
NMAH	-	Name Mapping Authority
OCLC	-	Online Computer Library Center
PID	-	Persistent Identifier (persistenter/ permanenter Identifikator)
PURL	-	Persistent URL
RFC	-	Request for Comments
tDAR	-	The Digital Archaeological Record
TIB	-	Technische Informationsbibliothek
URI	-	Uniform Resource Identifier
URL	-	Uniform Resource Locator
URN	-	Uniform Resource Name
W3C	-	World Wide Web Consortiums
XML	-	eXtensible Markup Language
XRI	-	eXtensible Resource Identifier
ZBW	-	Deutsche Zentralbibliothek für Wirtschaftswissenschaften

A Granularität

In Abschnitt 5.6 wurde kurz vorgestellt, welche Überlegungen bei der Festlegung der Granularitätsstufe für die Vergabe von persistenten Identifikatoren gemacht werden müssen. Welche Granularitätsstufen in den verschiedenen verwandten Einrichtungen zur Anwendung kommen, ist aus dem Abschnitt 6.2 ersichtlich.

Welche Konsequenzen die Wahl der Granularitätsstufe nach sich zieht, soll hier anhand der Datenbestände aus zwei Forschungsprojekten, Testdatensammlung 002 und 008, veranschaulicht werden. Die Testdatensammlungen wurden unverändert von den Datengebern übernommen und nicht für die Archivierung vorbereitet. Sie vermitteln aber einen guten Eindruck über den Aufwand des Ingest-Prozesses. Die Datensammlungen wurden mit dem Programm WinDirStat analysiert.

Testdatensammlung 002 Die Testdatensammlung 002 hat eine Größe von 109,3GB und beinhaltet Projektdateien über die Faustinathermen in Milet.

Auf 2.161 Verzeichnisse verteilen sich insgesamt 27.791 einzelne Dateien.

In der Baumkarte in Abbildung 5 ist zu erkennen, wie die Sammlung strukturiert ist. Unterschiedliche Farben stehen für unterschiedliche Dateiformate. Die Ordnerstruktur ist in den kissenartigen Strukturen zu erkennen.

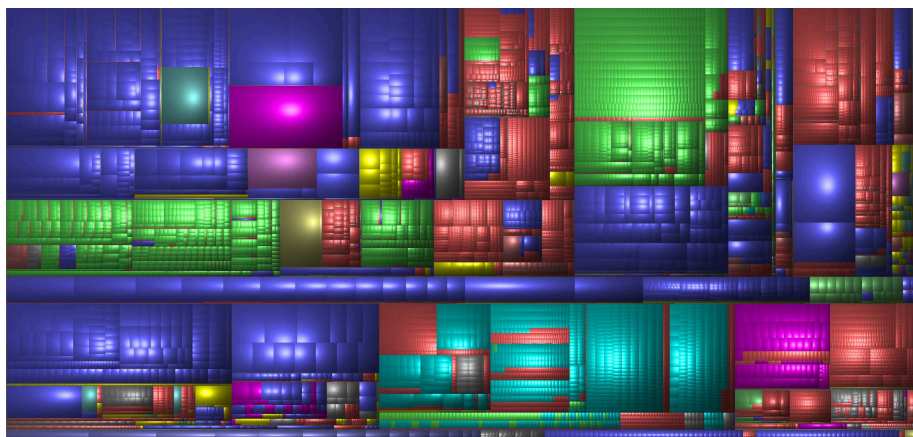


Abbildung 5: Baumkarte für Testdatensammlung 002 - Milet.

Die Baumkarte verdeutlicht, dass die Dateistruktur recht heterogen ist. Ein Ordner enthält mehrere unterschiedliche Dateitypen und auch die Anzahl der einzelnen Dateien in den Ordnern variiert sehr stark.

Testdatensammlung 008 Die Testdatensammlung 008 hat eine Größe von 117,7GB mit insgesamt 3.129 Dateien, die sich auf 160 Verzeichnisse verteilen.

Inhalt der Testdaten ist die bronzezeitliche Tempelanlage von Aleppo in Syrien. Es handelt sich hauptsächlich um 3D-Daten.

Die Baumkarte für diese Datensammlung ist in Abbildung 6 zu sehen. Im Gegensatz zur Testdatensammlung 002 in der Baumkarte in Abbildung 5 ist hier eine klare Struktur zu erkennen. Ein Ordner enthält maximal zwei verschiedene Dateiformate und die Anzahl der einzelnen Dateien in den Ordnern ist ähnlich.

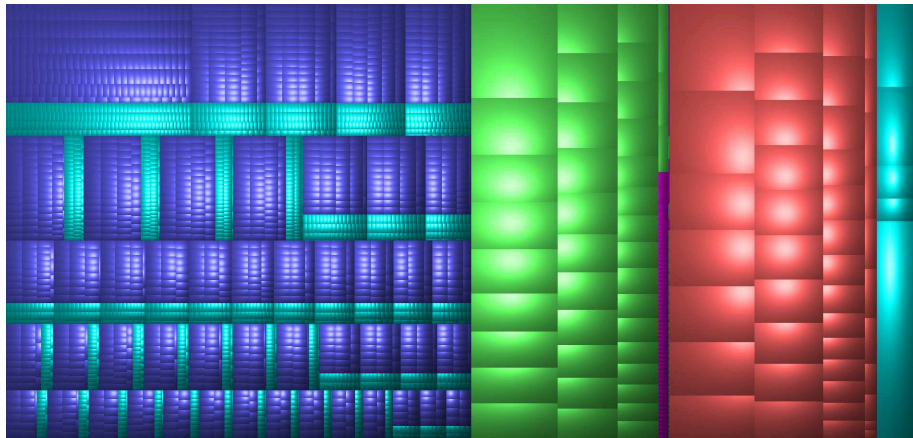


Abbildung 6: Baumkarte für Testdatensammlung 008 - Aleppo.

Höchste Granularität Würde man für die Testdatensammlung eine sehr hohe Granularität wählen und jeder einzelnen Datei einen PID geben, so würden für die Testdatensammlung 008 insgesamt 3.129 PIDs benötigt und für Testdatensammlung 002 sogar 27.791. In der Summe wären das 30.920 PIDs.

Vorteil hierbei wäre, dass die PIDs automatisch für jede einzelne Datei zugewiesen werden könnten. Allerdings stünde dann jeder einzelne Identifikator für sich und Beziehungen oder Gruppierungen von einzelnen Objekten können nur erkannt werden, wenn sie explizit in den Metadaten angegeben werden. Eventuell müssten für die Gruppierungen auch wieder eigene PIDs vergeben werden. Für jeden PID ist jeweils auch ein Metadatensatz erforderlich, was einen hohen Kuratierungsaufwand nach sich zieht.

Außerdem müsste im Sinne der guten Praxis für jeden Identifikator eine eigene Landing Page erstellt (oder automatisch erzeugt) werden, womit man allein für die zwei Testdatensammlungen mindestens 30920 eigene Seiten bräuchte.

Niedrigste Granularität Wählt man für die Testdatensammlungen die geringste Granularitätsstufe, so werden nur zwei persistente Identifikatoren vergeben, die sich jeweils auf die komplette Datensammlung beziehen. Auch diese könnten automatisch vergeben werden.

Ein zukünftiger Nutzer könnte jedoch vor dem Problem stehen, dass beispielsweise der Verweis auf eine bestimmte Datei lediglich mit der Angabe des PIDs von Testdatensammlung 002 gemacht wird. Der Nutzer würde mit einer Liste von 27.791 Dateien konfrontiert werden, aus der er die passende Datei herausuchen muss.

Die Nutzbarkeit der Daten für einen Nutzer, der nicht mit den Inhalten einer Datensammlung vertraut ist, wird erheblich erschwert, wenn thematisch divergente Objekte mit einem einzigen Identifikator zusammengefasst werden. Beispielsweise zeigen die Namen der Unterordner „Bauforschung“, „Geophysik“ oder „iDAIfield“ exemplarisch solch eine Heterogenität. Diese Heterogenität verdeutlicht zudem die Notwendigkeit für eine ausführliche Dokumentation der Datenablagestruktur.

Für die Testdatensammlung 008 wäre ein einzelner Identifikator für die gesamten Sammlung kein Problem, da diese übersichtlich genug ist.

Die niedrigste Granularitätsstufe wird auch von verwandten Einrichtungen gewählt, wie es aus Abschnitt 6.2 etwa für GESIS, ADS und PANGAEA hervorgeht.

Ein Mittelweg In der Praxis muss für IANUS ein Mittelweg gefunden werden, da manche Projektarchive zu groß sind, um lediglich mit einem einzelnen Identifikator auszukommen, während für andere Datensätze ein Identifikator ausreichen könnte.

Beispielsweise kann für Testdatensammlung 008 ein übergeordneter PID vergeben werden, der weitere PIDs enthält. Diese weiteren, untergeordneten PIDs könnten im Fall von Datensammlung 008 die einzelnen Tempelbestandteile, wie etwa Reliefblöcke referenzieren, die selbst wiederum Datensammlungen aus prozessierten Daten und Rohdaten enthalten.

In Abbildung 7 ist die Ordnerstruktur von Testdatensammlung 008 dargestellt. Dabei wird zwischen der Tempelanlage und einzelnen Reliefblöcken unterschieden. Die Tempelanlage ist unterteilt in *Punktwolken* und *Mesh*, wobei letzteres noch Teilbereiche und die Gesamtanlage unterscheidet.

Die Reliefblöcke liegen als Rohdaten und als prozessierte Daten vor. Die Rohdaten sind jeweils noch nach Format für jeden einzelnen Block gruppiert.

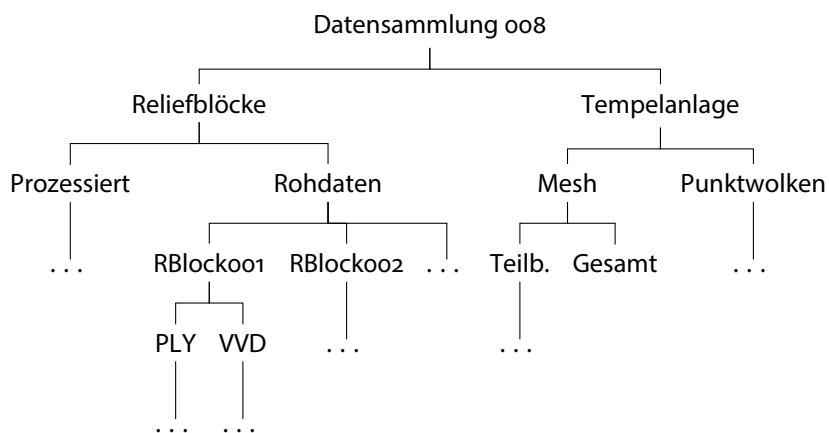


Abbildung 7: Ordnerstruktur Testdatensammlung 008 - Aleppo.

Eine mögliche PID-Zuweisung ist in Abbildung 8 skizziert, wobei blauer Text einen PID markiert. Dabei werden die Rohdaten und die prozessierten Daten jeweils zusammengefasst und dem entsprechenden Reliefblock zugewiesen.

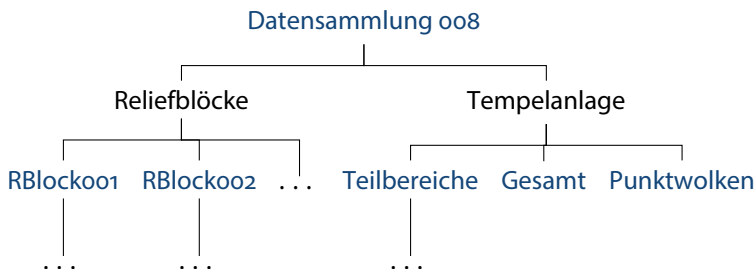


Abbildung 8: Vergabe von PIDs für Testdatensammlung 008 - Aleppo.

Da die zweite Ebene nur aus den Teilbäumen *Reliefblöcke* und *Tempelanlage* besteht, muss hierfür nicht jeweils extra ein eigener PID angelegt werden. Die Referenzierung kann in diesem Fall mit Hilfe des übergeordneten Identifikators und zusätzlichem Hinweis auf den gewünschten Teilbaum unkompliziert bewerkstelligt werden. Soll aber ein bestimmter Reliefblock oder eine bestimmte Datei davon adressiert werden, so kann hierfür wieder ein eigener Identifikator verwendet werden. In den Abbildungen 9 und 10 sind mögliche Referenzierungen für ein Objekt aus der Darstellung in Abbildung 8 angegeben.

Der in Abbildung 8 angegebene Vorschlag zur Vergabe von PIDs stellt lediglich eine mögliche Form vor. Theoretisch sind viele weitere Varianten möglich, auch die Vergabe von nur einem Identifikator für die gesamte Testdatensammlung wäre praktikabel. In dem letzten Fall könnte beispielsweise mit dem Zitierhinweis aus Abbildung 11 auf ein bestimmtes Objekt in der Datensammlung 008 referenziert werden.

Zitierhinweis: Arne Weiser (2013) Reliefblock 001 der Tempelanlage von Aleppo, Syrien. (PID-System:PID von *RBlock001*)

Abbildung 9: Angabe, wenn das Objekt einen eigenen PID hat.

Zitierhinweis: Arne Weiser (2013) Reliefblock 001 der Tempelanlage von Aleppo, Syrien. (PID-System:PID von *RBlock001*) *In:* Arne Weiser (2013) Virtual Archaeology. Die Entwicklung eines Instrumentariums (toolbox) für die Einbindung und Analyse verschiedener 3D-Daten zur Beantwortung archäologischer und geoarchäologischer Fragestellungen. Beispielhaft durchgeführt an den Datensätzen der bronzezeitlichen Tempelanlage Aleppo, Syrien. (PID-System:PID von *Datensammlung 008*)

Abbildung 10: Alternative Angabe, wenn das Objekt einen eigenen PID hat.

Zitierhinweis: Arne Weiser (2013) Reliefblock 001 der Tempelanlage von Aleppo, Syrien. *In:* Arne Weiser (2013) Virtual Archaeology. Die Entwicklung eines Instrumentariums (toolbox) für die Einbindung und Analyse verschiedener 3D-Daten zur Beantwortung archäologischer und geoarchäologischer Fragestellungen. Beispielhaft durchgeführt an den Datensätzen der bronzezeitlichen Tempelanlage Aleppo, Syrien. (PID-System:PID von *Datensammlung 008*)

Abbildung 11: Angabe, wenn das Objekt keinen eigenen PID hat.

Im Fall von Testdatensammlung 002 ist die Lage etwas komplizierter, da die Datenstruktur nur von dem Datengeber vollständig verstanden wird. Hier müsste der Datengeber noch Zuarbeit leisten und die Ordnerstruktur in einer Dokumentation ausführlich beschreiben oder manuell bestimmen, wie PIDs vergeben werden sollen. In der aktuell vorliegenden Form würde ein einzelner Identifikator nicht ausreichen.

Empfehlungen In der Praxis sollte minimal pro Projektarchiv oder Datensammlung ein PID vergeben werden. Diese Grenze darf nicht unterschritten werden.

Als Maximum ist die Vergabe von PIDs für jede einzelne Datei zu betrachten. Eine Überschreitung dieser Grenze, beispielsweise durch Zuweisung von PIDs für Dateiteile, sollte nur im begründeten Ausnahmefall erfolgen.

Generell gilt zu beachten, dass ein PID vergeben wird, um auf ein Objekt referenzieren zu können. Üblicherweise reicht ein PID für eine übersichtlich strukturierte Datensammlung aus. Mit Hilfe von Zitierhinweisen auf der *Landing Page* kann die richtige Referenzierung für Einzeldateien in einem Datensatz erfolgen. Auf einer *Landing Page* können auch Verweise auf verwandte oder zugehörige Ressourcen gemacht werden.

Für besonders große Projektarchive muss die Vergabe von zusätzlichen PIDs in jedem Fall manuell, am besten nach Rücksprache mit dem Datenproduzenten, erfolgen, um zu entscheiden welche Objekte in welcher Granularität einen PID erhalten sollen.

Wenn nicht alle Dateien automatisch einen eigenen PID bekommen, sollte die Möglichkeit geschaffen werden, dass Datennutzer zu einem späteren Zeitpunkt bei Zugriff auf einzelne Ressourcen, wie etwa ein Bild, die noch keinen eigenen PID besitzen, für diese einen neuen PID beantragen können (*PID-on-Demand*). Damit dies realisiert werden kann, muss es einen Mechanismus geben, der überprüft, ob ein PID schon existiert oder überhaupt für das gewünschte Objekt vergeben werden darf. Außerdem müssen die relevanten Metadaten zur Verfügung stehen oder im System nachgetragen werden.

Die Vorbereitungen und Entscheidungen für die Vergabe von PIDs müssen schon während des Ingest-Prozesses vorgenommen werden, um ein wiederholtes Bearbeiten der Daten zu vermeiden.

B Landing Pages

Auf einer *Landing Page* werden Informationen zusammengefasst, die sich mit einem bestimmten Objekt befassen. Beispielsweise ist die Seite, die nach dem Klick auf ein bestimmtes Suchergebnis in einem digitalen Bibliothekskatalog angezeigt wird, eine Landing Page. Sie liefert mehr Informationen zu einer Publikation, als die Kurzinformation in der Übersicht der Suchergebnisse.

Es gilt als gute Praxis, für alle mit einer PID registrierten Objekte eine eigene Landing Page bereitzustellen. Tatsächlich wird das auch von vielen Datenzentren praktiziert, wie beispielsweise die Seiten von ADS, GESIS oder PANGAEA zeigen.

Die Informationen, die eine Landing Page umfasst, sollten mindestens folgende Angaben umfassen:

- Volle Zitation der Daten
- Zugriffsmöglichkeiten auf die Daten und eventuell Informationen über Zugriffsbeschränkungen
- Zugehörige Metadaten
- Informationen über Software oder Kontext, um die Daten zu öffnen, zu lesen und zu interpretieren.
- Informationen zu anderen Versionen der Daten oder Hinweise auf verwandte Daten

Bei DataCite ist eine Landing Page unter bestimmten Gegebenheiten verpflichtend:⁵⁵

- Es wird spezielle Software benötigt, um die Daten zu öffnen
- Die Daten sind zugriffsbeschränkt

Auch wenn die Daten zugriffsbeschränkt sind, sollte die Landing Page trotzdem öffentlich einsehbar sein.

Wird das digitale Objekt gelöscht oder ist nicht mehr verfügbar, kann die Landing Page in eine sogenannte *Tombstone Page* umfunktioniert werden, die den Nutzer über den aktuellen Verbleib des Objektes aufklärt.

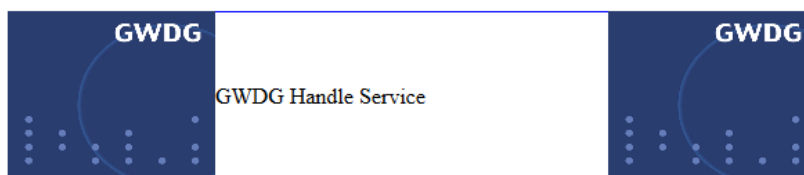
⁵⁵http://www.datacite.org/sites/default/files/Business_Models_Principles_v1.0.pdf

C PID Service DARIAH-DE - Handle System

Von dem Projekt DARIAH-DE hat IANUS einen Zugang für den Handle Service bekommen. Es handelt sich dabei allerdings nicht um einen Testzugang, sondern alle damit erstellten Handles werden auch tatsächlich registriert und können nicht mehr gelöscht werden.

Für dieses System wurden zwei unterschiedliche digitale Objekte ausgewählt. Das erste Objekt ist ein PDF-Dokument, das ein Poster über das IANUS-Projekt darstellt.⁵⁶ Das zweite Objekt ist als ein Konzept zu verstehen, da es sich um eine Webseite zu einem Workshop⁵⁷ handelt und sich die Inhalte vor, während und kurz nach der Durchführung noch ändern.

Der Service, der von der GWDG bereitgestellt wird, ist mit dem URL <http://handle.gwdg.de:8080/pidservice/> aufrufbar und die Oberfläche ist in Abbildung 12 dargestellt. Es gibt auch eine RESTful API, auf der die Weboberfläche basiert.



GWDG PID Handle Service

View Handle

PID / Handle:

jump to proxy view of data

jump via proxy to resource

jump directly to metadata

view XML instead of HTML (no effect for 'jump')

Enter PID (Handle) above and

Find Handle

URL:

Checksum:

Checksum is lower case hex: 'md5:32digits' or 'sha1:40digits'

Filesize: (in bytes)

Title:

Author(s):

Publication date: (see below)

Abbildung 12: Oberfläche des Handle Services von DARIAH-DE und GWDG.

⁵⁶http://www.ianus-fdz.de/attachments/download/473/IANUS%20Plakat%20A1_web.pdf

⁵⁷<http://www.ianus-fdz.de/lizenzworkshop>

Der Service bietet folgende Möglichkeiten:

- View Handle
 - Damit lässt sich ein Handle auflösen. Wenn „*jump via proxy to resource*“ ausgewählt ist, wird gleich zum Speicherort des Objektes aufgelöst. Ansonsten werden zunächst die verfügbaren Metadaten angezeigt.
- Find Handle
 - Bietet die Möglichkeit Handles anhand von Metadaten zu finden, allerdings sind die Suchmöglichkeiten sehr eingeschränkt und spezifisch. Beispielsweise besteht keine Möglichkeit nach Stichworten zu suchen. Die Felder URL, Checksum, Filesize und Title sind nur dann nützlich, wenn diese Informationen zu einem Objekt bereits bekannt sind.
- Create Basic Handle
 - Einen einfachen Handle ohne weitere Metadaten erstellen.
- Create Verbose Handle
 - Einen Handle mit Metadaten und weiteren Optionen erstellen. Dies sollte die bevorzugte Funktion sein.
- Edit Existing Handle
 - Einen vorhandenen Handle verändern, um beispielsweise den Speicherort zu aktualisieren.
- Check Login
 - Die eigenen Login-Daten ansehen.

Zunächst wurde mit der Funktion *Create Basic Handle* ein einfacher Handle für die Webseite zu einem Workshop erstellt. Dazu muss lediglich die URL-Adresse des gewünschten Objektes eingegeben werden, wie Abbildung 13 zeigt. Die URL-Adresse ist auch das einzige verpflichtende Feld in diesem System. Alle weiteren Informationen sind optional.

Create Basic Handle

URL:

Suffix: (user defined, optional)

Confirm in XML: (default: HTML)

Enter URL above and

Abbildung 13: Informationen zum Erstellen eines einfachen Handles.

Theoretisch kann ein eigenes Suffix für den Handle eingetragen werden, das mit in den Namen des Handles übernommen wird. Allerdings ist es nicht ratsam, zu viele semantische Informationen in den Namen aufzunehmen, weshalb der PID-Service auch empfiehlt, das Feld leer zu lassen.

Nach der Eingabe des URLs wird der Handle erstellt und die vorhandenen Details dazu werden angezeigt. In diesem Fall hat die Webseite den Handle *hdl:11858/00-1780-0000-0022-DBE5-C* bekommen, der sofort funktionsfähig ist und auch schon über den Resolver von CNRI⁵⁸ aufrufbar ist.

In Abbildung 14 ist die Rückmeldung vom System nach der Registrierung zu sehen.

Der Name des Handles folgt den Vorgaben von EPIC, inklusive der Prüfziffer an der letzten Stelle. Wird nach dem Handle gesucht und dabei ein Zahlendreher eingebaut, könnte anhand der Prüfziffer festgestellt werden, dass die Eingabe fehlerhaft war. Diese Funktionalität ist aber in dem getesteten System nicht umgesetzt, weshalb lediglich eine Fehlermeldung erscheint, dass der Handle nicht vergeben wurde.

Die eingegebene URL-Adresse wird nicht überprüft, was es zum einen ermöglicht, dass Handles für Objekte vergeben werden können, die noch nicht existieren, aber zum anderen müssen die Handles mindestens einmal aufgelöst werden, um zu prüfen, ob der eingegebene URL korrekt ist.

Ein bereits bestehender Handle kann nachträglich bearbeitet werden. Dies ist wichtig, da bei der Änderung des Speicherortes der URL entsprechend angepasst werden muss. Zur Bearbeitung des Handles gelangt man über die Suchfunktion, wenn die Metadaten des eingegebenen Handles angezeigt werden. Alle vorhandenen Felder können bearbeitet werden. Mittels dieser Funktion können die Informationen eines einfachen Handles auf den Umfang ergänzt werden, der mit der Funktion *Create Verbose Handle* zur Verfügung gestellt wird.

Wird versucht ein Handle für ein Objekt anzulegen, das bereits mit einem PID versehen ist, gibt das System eine Fehlermeldung aus, die hier in Abbildung 15 wiedergegeben ist.

Dieser Mechanismus funktioniert aber nicht zuverlässig. Tests ergaben, dass Objekte mit einem bestehenden DOI⁵⁹ oder Handle⁶⁰ nicht mit der Fehlermeldung abgefangen werden und tatsächlich einen weiteren Handle erhalten könnten.

Mit der Funktion *Create Verbose Handle* können Handles mit zusätzlichen Informationen angelegt werden. In Abbildung 16 sind die verfügbaren Felder abgebildet.

In diesem Fall wurden bereits die Informationen für die Vergabe eines Handles für ein PDF-Dokument eingetragen. Das Dokument hat den Handle *hdl: 11858/00-1780-0000-0022-DBE6-A* erhalten.

Die Prüfsumme, die entweder als „*md5*“ oder „*sha1*“ eingegeben werden kann, wurde für dieses Beispiel nicht angegeben. Die Prüfsumme kann nur ein einziges Mal eingetragen werden und anschließend nicht mehr verändert werden. Auch ein Ablaufdatum wurde nicht gesetzt. Mit dem Ablaufdatum kann angegeben werden,

New PID Handle created:

[11858/00-1780-0000-0022-DBE5-C](#)

User details

User: 1780
Institute: 1780
Contact: email:
Can create PID: yes
Can update PID: yes

[\[Back to input form\]](#)

Abbildung 14: Bestätigung nach Erstellung eines einfachen Handles.

⁵⁸<http://hdl.handle.net/>

⁵⁹Im Test: doi:10.4232/1.10028

⁶⁰Im Test: hdl:11858/00-ZZZZ-0000-0001-6D1D-0

HTTP Status 403 - Another Handle / PID already points here: 11858/00-1780-0000-0022-DBE5-C

type Status report

message [Another Handle / PID already points here: 11858/00-1780-0000-0022-DBE5-C](#)

description Access to the specified resource ([Another Handle / PID already points here: 11858/00-1780-0000-0022-DBE5-C](#)) has been forbidden.

Apache Tomcat/5.5.27

Abbildung 15: Fehlermeldung, wenn das Objekt bereits einen PID besitzt.

Create Verbose Handle

URL:	<input type="text" value="http://www.ianus-fdz.de/attachments/download/473/IANUS%20Plakat%20A1_web.pdf"/>
Filesize:	<input type="text" value="322768"/> (in bytes)
Checksum:	<input type="text"/> (optional, see below)
Checksum is write-once, lower case hex: 'md5:32digits' or 'sha1:40digits'	
Title:	<input type="text" value="Plakat - IANUS Forschungsdatenzentrum Archäologie & Altertumswissenschaften"/>
Author(s):	<input type="text" value="IANUS"/>
Publication date:	<input type="text" value="2013"/> (see below)
Expiry date:	<input type="text"/> (optional, see below)
Dates can be in format yyyy, yyyy-mm-dd or yyyy-mm-dd hh:mm:ss	
Metadata URL:	<input type="text" value="http://www.ianus-fdz.de/documents/66"/> (optional)
Suffix:	<input type="text"/> (user defined, optional)
Confirm in XML:	<input type="checkbox"/> (default: HTML)
Enter data above and	<input type="button" value="Create Verbose Handle"/>

Abbildung 16: Einen vollständigen Handle erzeugen.

bis zu welchem Zeitpunkt die Informationen und vor allem der URL aktualisiert werden.

In das Feld „Metadata URL“ kann eine URL-Adresse eingegeben werden, die auf eine Landing Page oder eine Datei mit weiteren möglichen Metadaten über das Objekt verweist. Es werden dafür keine weiteren Vorgaben gemacht, weshalb der Nutzer frei wählen kann, welche zusätzlichen Informationen zu dem Objekt gespeichert werden sollen. In diesem Testfall führt die URL-Adresse auf eine Landing Page, von der aus das PDF-Dokument aufgerufen werden kann.

Fazit Der PID-Service von der GWDG und DARIAH-DE bietet die grundlegendsten Funktionen zur Vergabe von Handles. Sehr gut ist, dass bei der Auflösung von Handles standardmäßig zunächst die zugehörigen Metadaten angezeigt werden und die Weiterleitung zu dem eigentlichen Objekt nur dann erfolgt, wenn dies explizit gewünscht ist. Das ist von Vorteil, da der Nutzer auf diese Weise die Möglichkeit

hat, zunächst die vorhandenen Informationen zu dem Objekt zu überprüfen und so auch sehen kann, worum es sich bei dem Objekt handelt und in welchem Format dieses vorliegt.

Die Funktionen dieses Services sind aber nicht völlig ausgereift. Beispielsweise funktioniert die Überprüfung des URLs auf einen bereits bestehenden PID nicht zuverlässig. Auch wird bei der Auflösung von Handles keine Rücksicht auf die Prüfziffer genommen, womit es aber möglich wäre den Nutzer auf eventuelle Eingabefehler bei dem Handle hinzuweisen.

Die einzige verpflichtende Information ist die URL-Adresse, was aber zu wenig für ein zitierfähiges Objekt ist. Ein URL allein lässt auch keine zuverlässige Identifizierung des Objektes zu, was das Vertrauen in das System nicht unbedingt fördert.

Aufgefallen ist außerdem, dass die Eingabefelder nicht konsistent beschriftet sind. Beispielsweise ist das Feld „*Metadata URL*“ bei *Create Verbose Handle* in der Bearbeitungsfunktion nur mit „*Metadata*“ beschriftet.

Auch die Nutzeroberfläche könnte nutzerfreundlicher gestaltet werden, indem beispielsweise Hinweise zum Ausfüllen der Felder direkt als Tooltip bei den Feldern hinterlegt werden. Die Fehlermeldungen müssten teilweise informativer gestaltet werden und vor allem fehlen Schaltflächen, um wieder zu dem Eingabeformular oder dem zuletzt bearbeiteten Handle zu gelangen.

Wünschenswert wäre auch eine Option, um ein Objekt mit der gegebenen Prüfsumme zu vergleichen, und für den Bereitsteller der Daten wäre es praktisch, wenn nach Ablauf des Ablaufdatums eine Erinnerung versendet wird, um gegebenenfalls den URL anzupassen oder andere Schritte einzuleiten. Eine Überprüfung der URLs auf Funktionsfähigkeit wäre ebenfalls hilfreich.

Letztendlich hat dieser PID-Service eine gute Grundlage, bietet jedoch noch erheblichen Verbesserungs- und Erweiterungsspielraum.

D Testumgebung von da|ra - DOIs

Die Registrierungsagentur für Sozial- und Wirtschaftsdaten da|ra ist Mitglied von DataCite und wird von GESIS und der ZBW betrieben. Sie registriert DOIs.

Für IANUS wurde eine Testumgebung zur Verfügung gestellt, die auf <http://dara-test.gesis.org:8084/dara/mydara> zu erreichen ist. Zusätzlich gibt es im Produktsystem eine API.

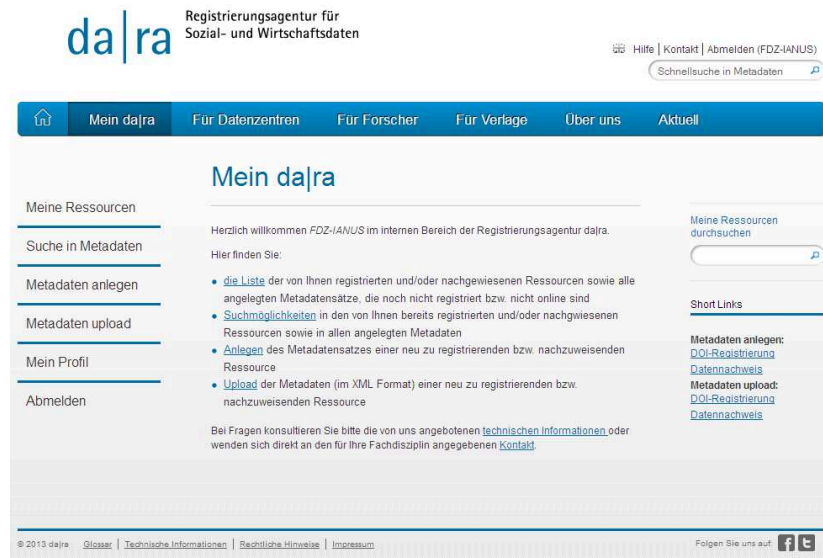


Abbildung 17: Oberfläche von da|ra.

Die Oberfläche des Testsystems ist in Abbildung 17 wiedergegeben. Die Funktionen des Systems umfassen eine Übersicht der eigenen registrierten Objekte, eine Suchfunktion in den Metadaten, das Anlegen von Metadaten für zu registrierende Objekte und den Upload von Metadaten im XML-Format für ein zu registrierendes Objekt.

Die Suchfunktion (Abbildung 18) erlaubt es entweder nach Stichworten im gesamten Metadaten-satz oder nach Stichworten in bestimmten Feldern des Metadaten-satzes zu suchen. Es kann auch nach bereits vergebenen DOIs gesucht werden. Allerdings eignet sich nur das Feld für die einfache Suche nach Stichworten, um Objekte zu

Abbildung 18: Suchmaske von da|ra.

finden, die nicht genau bekannt sind. Die übrigen Felder dienen eher der Suche nach einem spezifischen Objekt.

Da|ra unterscheidet für das Anlegen von Metadaten zwischen einem zu registrierenden Objekt und einem nachzuweisenden Objekt. Der Unterschied besteht darin, dass ein zu registrierendes Objekt einen DOI bekommen soll und dementsprechende Informationen eingetragen werden müssen. Soll ein Objekt nicht registriert werden, so bietet da|ra den Nachweis desselben an, um einem zukünftigen Nutzer Metadaten zur Verfügung zu stellen, die die Auffindbarkeit und die Identifikation eines nicht-registrierten Objektes erleichtern. Für dieses Testbed ist aber die DOI-Registrierung relevant.

Beide Methoden verwenden in der Eingabemaske die gleichen Metadatenfelder. Lediglich für die Registrierung eines DOIs gibt es noch einen darauf bezogenen Abschnitt. Das Metadatenschema von da|ra setzt auf die Vorgaben von DataCite auf und ist unter der URL-Adresse <http://www.da-ra.de/de/fuer-datenzentren/daten-registrieren/doi-und-metadaten/> abrufbar.

Hilfestellungen bei der Verwendung des Dienstes und beim Ausfüllen der Metadatenfelder erhält man als Nutzer über Tooltips, die hinter blauen Informationszeichen hinterlegt sind. Außerdem können auf den Seiten von da|ra ausführliche Dokumente zu den relevanten Themen gefunden werden.

Für die Registrierung eines DOIs müssen mindestens acht gekennzeichnete Pflichtfelder ausgefüllt werden. Fehlen diese oder Teile davon, weist das System darauf hin. Die Eingabemaske ist teilweise in Abbildung 19 zu sehen.

Wird eine noch nicht existierende oder falsche URL-Adresse für das Objekt angegeben, kann der DOI noch nicht registriert werden.⁶¹ Der Dienst meldet auch, dass die Registrierung fehlgeschlagen ist, spezifiziert aber nicht genau die Fehlerursache. Der URL sollte dabei idealerweise auf eine *Landing Page* verweisen.

Beim Anlegen eines Metadatensatzes für ein Objekt kann gewählt werden, ob der DOI gleich registriert werden soll, oder zu einem späteren Zeitpunkt. Auf diese Weise kann beispielsweise schon ein DOI für ein später zu veröffentlichendes Objekt reserviert werden.

Das Suffix für den anzulegenden DOI kann entweder von dem System automatisch erzeugt werden oder manuell eingetragen werden. Es ist auch möglich mehrere URLs anzugeben, mit denen das Objekt zu finden sein wird. Bei der Registrierung wird zunächst der erste Eintrag verwendet. Nachdem ein Metadatensatz erfolgreich angelegt und ein zugehöriger DOI registriert wurde, erscheint eine Erfolgsmeldung (Abbildung 20).

In dem Testsystem erhalten die angelegten DOIs das Präfix 10.5072, das innerhalb von DataCite als Testpräfix gehandhabt wird und die in unregelmäßigen Abständen gelöscht werden.

Innerhalb von da|ra werden die zu registrierenden Objekte als Studien bezeichnet, was der Entstehungsgeschichte als Registrierungsagentur für sozial- und wirtschaftswissenschaftliche Daten, die hauptsächlich als Studien vorliegen, geschuldet ist. Auch einige Metadatenfelder sind speziell auf dieses Fachgebiet abgestimmt.

Nachdem ein Objekt angelegt wurde, kann es nachträglich weiterbearbeitet werden. Neben den bereits beim Anlegen zur Verfügung stehenden Feldern, können hier auch weitere Informationen zur Datenerhebung und zur Beschreibung des Objekts eingetragen werden, wie Abbildung 21 zeigt.

⁶¹Dies wurde nur an einem Beispiel getestet und muss nicht unbedingt dem üblichen Verhalten des Dienstes entsprechen.

1. 1. Typ der Ressource* ⓘ

Datensatz ▾

1. 2. Identifizier der Ressource* ⓘ

Meine ID und Version
 Meine ID verwenden
 Automatische Generierung

ID Ressource*

Version*

1. 3. Titel* ⓘ

1. 4. Primärforscher* ⓘ

Person hinzufügen
 Institution hinzufügen

1. 5. DOI* ⓘ

Meinen DOI-Suffix verwenden
 Automatische Generierung

DOI Präfix* / DOI Suffix*

10.5072 /

URL*

URL 1 (URL, zu der der DOI-Name aufgelöst wird)

URL hinzufügen

1. 6. Sprache der Ressource* ⓘ

Abbildung 19: Ausschnitt aus der Eingabemaske zur Registrierung von DOIs.

Außerdem können weitere Identifikatoren (wenn das Objekt beispielsweise schon einen Handle erhalten hatte) und verbundene Identifikatoren hinzugefügt werden. Mit verbundenen Identifikatoren sind verwandte Objekte, wie etwa andere Versionen oder auch Publikationen gemeint. In dem Reiter für Publikationen können zugehörige Publikationen sogar strukturiert aufgelistet werden.

Der Dienst von da|ra ist aktuell kostenlos, wobei das in der Policy⁶² für die Zukunft nicht explizit ausgeschlossen wird.

Fazit Dass da|ra bereits eine breite Anwendung findet, ist der ausgereiften Nutzeroberfläche und der Menge an zusätzlich verfügbaren Informationsmaterialien deutlich anzumerken. Lediglich zusätzliche Beispiele für das Eintragen von Metadaten in bestimmte Felder (wie es z.B. bei EZID gelöst wurde) wären noch hilfreich.

⁶²Abschnitt 6 in: <http://www.da-ra.de/de/ueber-uns/da-ra-policy/policy/> [Version 2.2]



Abbildung 20: Erfolgreich registrierter DOI in da|ra.

Es fällt auf, dass der Dienst auf die Sozial- und Wirtschaftswissenschaften spezialisiert ist. Sollte IANUS den Dienst benutzen wollen, müssten die Metadatenfelder an den entsprechenden Stellen etwas angepasst und ergänzt werden. Ein zusätzliches Feld für Checksums von abgeschlossenen Dokumenten wäre wünschenswert.

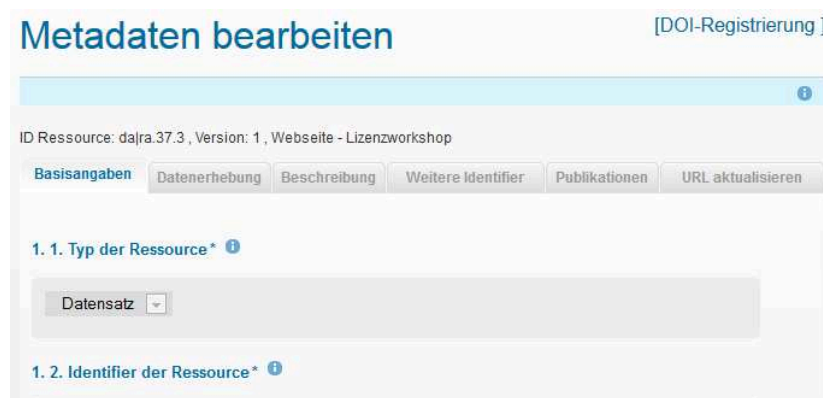


Abbildung 21: Bearbeiten von Metadatensätzen.

E Testumgebung von EZID - ARKs

Die California Digital Library bietet EZID als Dienst zur Registrierung und Verwaltung von ARKs an. Der Dienst kann auch mit DOIs umgehen und hat ebenfalls eine API. Die Kosten für den Dienst betragen für eine Forschungseinrichtung 1500 jährlich, um bis zu eine Million ARKs zu vergeben.⁶³

Für den Dienst wird eine Testumgebung auf <http://nzt.net/ezid/demo> zur Verfügung gestellt, die hier ebenfalls mit dem PDF-Dokument und der Webseite aus den vorangegangenen Testfällen getestet wird. Die mit dem Demo-System erzeugten PIDs sind zwei Wochen lang gültig.

Die Oberfläche von EZID ist in Abbildung 22 dargestellt. Bei dem Aufruf der Seite wird standardmäßig die einfache Oberfläche angezeigt. Die erweiterte Oberfläche wird weiter unten besprochen.

The screenshot shows the EZID website interface for creating a demo identifier. At the top left is the EZID logo with the tagline 'Long-term identifiers made easy'. On the top right are links for 'Login' and 'Contact Us'. Below the logo is a navigation menu with 'Home', 'Manage IDs', 'Create IDs', 'Lookup ID', and 'Demo' (which is highlighted). Under the navigation menu are links for 'Simple' and 'Advanced'. The main content area is titled 'Create Demo Identifier' and is divided into two steps. Step 1, 'Choose an identifier type', has two radio button options: 'ARK (ark:/99999/fk4)' (selected) and 'DOI (doi:10.5072/FK2)'. Step 2, 'Describe the identified object', has four text input fields labeled 'Location (URL):', 'Who:', 'What:', and 'When:'. Below these fields is a 'Create' button. To the right of the form is a box titled 'About Demo Identifiers' containing text: 'Demo identifiers, indicated on EZID screens by yellow highlighting, are temporary identifiers for use in testing the EZID service. They last for two weeks.' Below the form, there is a note: 'Clicking "Create" will create the identifier and make it public. To defer making it public, use the Advanced form.' followed by a question mark icon. Below that is another note: 'Once created, an identifier cannot be renamed or deleted.' At the bottom of the page, there is a footer with the text: 'EZID is a service of the University of California Curation Center of the California Digital Library', 'Copyright © 2009-2013 The Regents of the University of California', a logo, and links for 'About Us', 'Help', and 'Privacy Policy'.

Abbildung 22: Oberfläche von EZID.

Für die einfachste Form eines ARKs wird lediglich die URL-Adresse, ein Autor, ein Titel und ein Veröffentlichungsdatum benötigt. Diese Felder entsprechen dem Dublin Kernel und werden in EZID als Electronic Resource Citation (ERC) bezeichnet. Wird der ARK mit der einfachen Oberfläche erstellt, ist er sofort öffentlich und für jeden sichtbar.

⁶³<http://nzt.net/ezid/home/pricing>

Der Dienst bietet Hilfestellungen in Form von Tooltips an. Die Tooltips erscheinen, wenn der Mauszeiger über eines der Felder ruht und sind inhaltlich auf das entsprechende Feld abgestimmt. Zusätzlich zeigt ein Klick auf die eingblendeten Fragezeichen weitere ausführlichere Hilfetexte an. In Abbildung 23 ist gerade der Tooltip für das Feld „Who“ eingeblendet und in der unteren rechten Ecke ist ein Fragezeichen zu sehen, dessen dahinterliegende Hilfetext eine Erklärung zu dem nebenstehenden Satz liefert.

The screenshot shows the 'Create Demo Identifier' form in EZID. At the top, there are links for 'Simple' and 'Advanced'. The form is divided into two steps: 'Step 1: Choose an identifier type' and 'Step 2: Describe the identified object'. In Step 1, 'ARK (ark:/99999/fk4)' is selected. In Step 2, there are input fields for 'Location (URL)', 'Who', 'What', and 'When', followed by a 'Create' button. A tooltip is displayed over the 'Who' field, providing instructions on how to format names. Below the form, a note states: 'Clicking "Create" will create the identifier and make it public. To defer making it public, use the Advanced form.' A question mark icon is also visible.

Simple | [Advanced](#)

Create Demo Identifier

Step 1: Choose an identifier type

ARK ([ark:/99999/fk4](#))

DOI ([doi:10.5072/FK2](#))

Step 2: Describe the identified object

Location (URL):

Who:

What:

When:

Clicking "Create" will create the identifier and make it public. To defer making it public, use the [Advanced form](#). ?

Who
 The name of an entity (person, organization, or service) responsible for creating the content or making it available. For an article, this could be an author. Put name parts in "sort-friendly" order, such as:
 van Gogh, Vincent,
 Hu Jintao
 Gilbert, William, Sir,,; Sullivan, Arthur, Sir,,
 Separate multiple names with ",". Append one or more final commas (",") to indicate that one or more internal commas can be used as inversion points to recover natural word order (if different from sort-friendly word order).

Abbildung 23: Tooltips in EZID, hier für das Feld *Who*.

Nachdem der PID erstellt wurde, erscheint eine Bestätigungsseite, wie in Abbildung 24 dargestellt. Darauf sind die verfügbaren Informationen zu dem PID zusammengefasst. Außerdem erstellt EZID einen QR-Code, der den Link auf den ARK codiert, der unter *Get link* hinterlegt ist. Der Link ist die URL-fizierte Form des ARKs. In diesem Beispiel hat die Webseite des Workshops den ARK [ark:/99999/fk4qv5r9q](#) erhalten. Die URL-fizierte Form (und somit auch der Link) ist <http://n2t.net/ark:/99999/fk4qv5r9q>

Die Eingaben können mit der Editierfunktion korrigiert und ergänzt werden. Im Editiermodus besteht die Möglichkeit das Zitierprofil zu ändern, wodurch mehr Metadatenfelder zur Verfügung gestellt werden. Allerdings werden, bis auf die URL-Adresse, keine der bereits eingegebenen Informationen in die analogen Felder übernommen, weshalb das Zitierprofil anfangs festgelegt und später nicht mehr geändert werden sollte.

Wenn ein PID in der einfachen Oberfläche erstellt wird, ist es möglich komplett leere PIDs anzulegen, da das System, wenn kein URL eingegeben wurde, eine eigene URL-Adresse einträgt. Wird aber ein URL eingegeben, wird dessen Syntax geprüft und eine Meldung ausgegeben, wenn die Eingabe syntaktisch nicht einem URL entspricht. Die URL-Adresse wird aber nicht auf Funktionsfähigkeit überprüft und es wird auch nicht geprüft, ob ein PID mit dem eingegebenen URL bereits vorhanden ist. Dementsprechend ist es auch möglich einen Identifikator für ein Objekt anzulegen, das noch nicht online vorhanden ist.

Identifier Details

Identifier: [ark:/99999/rk4q5r9q](https://nbn-resolving.org/urn:nbn:de:hbz:5:1-63864-p0001-9) [Get link](#)

About the identified object

Location (URL): <http://www.ianus-fdz.de/lizenzworkshop>
Who: IANUS; DARIA-DE
What: Webseite - Lizenzworkshop
When: 2013

[Edit](#)



About the identifier



Owner: anonymous
Group: anonymous
Co-owners: none
Created: 10/17/2013 04:18 AM
Last modified: 10/17/2013 04:18 AM
Status: public 
Allow harvesting/indexing? Yes 

Abbildung 24: Bestätigung nach Erstellung eines ARKs.

Die erweiterte Oberfläche von EZID (Abbildung 25) bietet zusätzlich die Option, ein eigenes Suffix einzutragen, wobei dies von den Betreibern nicht empfohlen wird. Ebenfalls kann entschieden werden, ob der PID öffentlich und indizierbar gemacht werden soll oder nicht. Bevor die Metadaten eingetragen werden, kann ein Zitierprofil ausgewählt werden. Neben dem hauseigenen ERC-Profil stehen noch *DataCite* und *Dublin Core* zur Verfügung.

Dublin Core beinhaltet zusätzlich zu den Feldern von ERC noch die Felder „*Publisher*“ (Herausgeber) und „*Type*“ (Typ der Ressource). Wird das Zitierprofil *Dublin Core* gewählt, kann, wie auch mit dem ERC-Profil, ein komplett leerer Identifikator angelegt werden, da auch hier der Fall toleriert wird, wenn kein URL eingegeben wurde.

Das Zitierprofil *DataCite* ist wesentlich umfangreicher und das Metadatenschema ist auf <http://schema.datacite.org/> dokumentiert. In Abbildung 26 ist ein Ausschnitt der in EZID angezeigten Felder zu sehen. Erforderliche Felder sind gekennzeichnet und werden vom System auch auf eventuell fehlende Eingaben überprüft. Zu den Pflichtangaben gehören der Ersteller, der Titel, der Herausgeber und das Veröffentlichungsjahr. Felder, die mehrfach benötigt werden, können bei Bedarf mit den Plus-Buttons hinzugefügt werden.

ARKs, die mit dem *DataCite*-Profil erstellt wurden, können im Moment nicht über die grafische Weboberfläche bearbeitet werden, sondern nur über die API.

Fazit Der Dienst EZID ist sehr nutzerfreundlich und mit den vielen Hilfen kommen auch wenig erfahrene Nutzer gut zurecht. Problematisch ist, dass leere PIDs mit dem ERC- und *Dublin Core*-Profil erstellt werden können. Zumindest eine URL und ein paar zugehörige Metadaten müssten verlangt werden.

Es fehlt eine Suchfunktion, die Identifier anhand ihrer Metadaten findet.

[Simple](#) | [Advanced](#)

Create Demo Identifier

Step 1: Choose an identifier type

- ARK ([ark:/99999/fk4](#))
- DOI ([doi:10.5072/FK2](#))

Step 2: Optional: Enter a custom remainder (suffix) ?

Recommended: Leave blank

Step 3: Decide

- Make this identifier public? Yes No ?
- Allow harvesting/indexing? Yes No ?

Step 4: Describe the identified object using the ERC citation profile: ?

Location (URL):

Who:

What:

When:

Once created, an identifier cannot be renamed or deleted.

About Demo Identifiers

Demo identifiers, indicated on EZID screens by yellow highlighting, are temporary identifiers for use in testing the EZID service. They last for two weeks.

Abbildung 25: Erweiterte Oberfläche von EZID.

Step 4: Describe the identified object using the DataCite **citation profile:** ?

Location (URL):

Creator * ? - +

Name: *

Name Identifier:

Identifier Scheme:

Scheme URI:

Title * ? - +

Title: *

Type:
 Main title Alternative title Subtitle Translated title

Publisher: * ?

Publication Year: * ?

Resource Type: ?
Dataset ▼

Resource Type Description: ?

Abstract ? - +

Descriptive information:

Type:
Abstract ▼

Abbildung 26: Ausschnitt der DataCite-Felder in EZID.