Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1031–1040
DOI: 10.15439/2016F87
ACSIS, Vol. 8. ISSN 2300-5963

On Constructing Persistent Identifiers with Persistent Resolution Targets

Oliver Wannenwetsch
Gesellschaft für wissenschaftliche
Datenverarbeitung Göttingen (GWDG),
Göttingen, Germany
oliver.wannenwetsch@gwdg.de

Tim A. Majchrzak
Department of Information Systems
University of Agder,
Kristiansand, Norway
timam@uia.no

***Abstract*—Persistent Identifiers (PID) are the foundation referencing digital assets in scientific publications, books, and digital repositories. In its realization, PIDs contain metadata and resolving targets in form of URLs that point to data sets located on the network. In contrast to PIDs, the target URLs are typically changing over time; thus, PIDs need continuous maintenance – an effort that is increasing tremendously with the advancement of e-Science and the advent of the Internet-of-Things (IoT). Nowadays, billions of sensors and data sets are subject of PID assignment. This paper presents a new approach of embedding location independent targets into PIDs that allows the creation of maintenance-free PIDs using content-centric network technology and overlay networks. For proving the validity of the presented approach, the Handle PID System is used in conjunction with Magnet Link access information encoding, state-of-the-art decentralized data distribution with BitTorrent, and Named Data Networking (NDN) as location-independent data access technology for networks. Contrasting existing approaches, no green-field implementation of PID or major modifications of the Handle System is required to enable location-independent data dissemination with maintenance-free PIDs.**

***Index Terms*—Persistent Identifier; Information Centric Networks; Named Data Networking; Magnet Link; URN; Handle System; Digital Object Identifier; Overlay Network**

I. Introduction

The concept of *Persistent Identifier (PID)* is essential for referencing, citing and linking (digital) resources using a durable and reliable identifier. PIDs are used to ensure the long-term valid access to possibly moving digital resources that suffer from changing URLs and storage locations in networks. PIDs contain an adjustable target Uniform Resource Locator (URL). To reflect changing and volatile data locations, the target URL of a PID must be updated to the currently valid storage location. By employing organizational and technical measurements, different PID systems allow building, using and maintaining a long-term existing (digital) identifier that is backed by distributed systems, replication schemes, and policies. With these measurements in place, PID infrastructure operating organizations are able to offer PID systems that are resilient against failure, and even catastrophic scenarios. Ideally, the range of PID infrastructure resilience includes scheduled downtimes of server and networks, major infrastructure problems caused by power failures, and hardware and software problems, as well as ultra critical events of complete data center losses caused by fires, explosions or natural disasters.

Besides the measurements that protect PIDs on infrastructure level, PIDs have to be protected on the content-side as well. The content of the PID has to be intact and readable with given encoding schemes. Furthermore, the metadata have to be addressed with a given metadata scheme that explains the semantics of the metadata. Then, the content has to reflect the current state of the data object the PID is linking to. This includes the *up-to-dateness* of metadata sets stored in PIDs, which are often encoded as *key-value* pairs. Particularly important is the correctness of the PID metadata field `target URL`, which points to the digital object addressed by the PID for long-term access (c.f. Figure 1). Contrasting the infrastructure protection of PID, this content validation is not done by the PID infrastructure operating organization. It is a task of the data owners that registered the PID for their data or the subsequent organization that has the task of curating the data and its associated PIDs. Only those organizations have the necessary understanding on the data and its metadata to check and update PIDs for assuring long-term access. Moreover, they are aware of the current location of the data linked by PID. Thus, with regular control and adjustment of the target URLs to the current data location, well-established PID systems can guarantee persistency of identifiers physically, while data owner organizations have to accept the burden of regularly checking and updating metadata and target URLs. Only with collaboration PIDs are able to provide long-term access [1].

While PIDs solve the problem of changing data locations by constant efforts from PID infrastructure providers and data hosting organizations, the network research community has come up with numerous concepts for creating location-independent data access. In these concepts that access information for data attached to a network is not based on the data location, but it is based on the content of the data [2]. Hence, in the case of changing data location the access information remain stable. Two efforts that realize location-independent access is the state-of-the-art decentralized data distribution technology *BitTorrent* and *Named Data Networking (NDN)* as next-generation Internet technology. Although both technologies allow stable location-independent access to data, they do not provide persistent access like PID.

Our approach presented in this paper combines the very stable concept of the *Handle* PID system with the advantages of

978-83-60810-90-3/$25.00©2016, IEEE

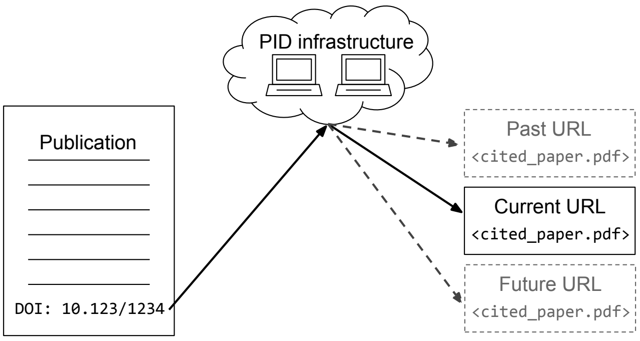

Figure 1: Adjustment of PID target URLs to reflect current data location.

location-independent access with its stable access information scheme. By this, we significantly lower the efforts of PID maintenance, as regular checks for data hosting organizations concerning locations and data availability will become obsolete. PIDs that are enabled for location-independent access remain valid as long as data is available online in BitTorrent or Named Data Networking (NDN) networks. To store the location-independent access information in existing Handle PID data structures, we extend the – yet not standardized – approach of *Magnet Link Schemes*, which is very successful in peer-to-peer communities [3]. In detail, our novel approach has the benefits of using an unmodified version of the Handle PID system that allows a practical implementation of our approach in existing Handle PID systems. Besides the advantages of interoperability, it does not come with a significant impact on PID resolution. For the location-independent access with BitTorrent and NDN only the small overhead of PID resolution is added.

Furthermore, we extend the Magnet Link scheme into the domain of NDN and provide a transport container format that includes besides the NDN data name also the necessary cryptographic access information for verifying data access authenticity. Thus, we can also improve Web browser-based NDN applications that use HTML-links for interconnecting NDN Web resources by the Magnet Link scheme. The latter has been originally designed for interconnecting non HTTP-resources in hypertext document contexts.

This paper is structured as follows. First, we present existing efforts in Section II. Second, we summarize the existing approaches of location-based data access through PID in Section III. Then, we line out state-of-the-art techniques for location-independent access in Section IV. In Section V, we introduce the Magnet Uniform Resource Identifier (URI) scheme format as container for storing location-independent access information. After that, we extend the Magnet URI scheme for the usage in the domain of content-centric networks and PID in Section VI. A proof-of-concept implementation is illustrated in Section VII together with an evaluation of performance in Section VIII that is combined with a discussion of the results. Finally, we draw a conclusion in Section X.

II. Related Work

To our knowledge, the concept of building care-free persistent identifier targets using content-centric technology has not been subject of extensive study. In the literature several related concepts can be identified.

The concept of bridging different content-centric network systems through a centralized Uniform Resource Name (URN) system has been initially drafted by Sollins in 2012 [4]. Her concept utilizes foundations of PID principles for creating an identification system for different Information Centric Networks (ICN) families and their related data objects that meets the requirements of scalability, longevity, evolvability, and security. Sollins's identification system abstracts different object naming schemes from ICN families such as Data Oriented Network Architecture (DONA) [5], Network of Information (NETINF) [6] and Publish-Subscribe Architecture (PURSUIT) [7]. Although, PID principles are used for location-independent data access, the publication by Sollins does not suggest access through an existing well-introduced PID system that is provided in our work, but rather uses a greed-field approach for location-independent data access.

The realization of complex secure naming schemes for content-centric data has been covered by Dannewitz et al. in 2010 [8]. They demand name persistency without incorporating the concept of PIDs. In their publication, Dannewitz et al. clarify that basic security functionality must be attached directly to the data and its naming scheme, because the identity of network locations cannot be used as a trust base for data authenticity. Our approach follows this principle for secure location-independent data access and facilitates directly attached PID security mechanisms. By this, location-independent access through PID is shifted into the requirements formulated by Dannewitz et al., and our approach enables authentic data access through PIDs.

In the context of semantic digital archives for archiving data of Personal Information Manager (PIM) applications, Haun and Nürnberger proposed a PID schema for accessing objects in file systems using an URN-like Magnet Links scheme [9]. They link the congruent attributes of the Magnet Link scheme to the attributes provided by some PID systems such as global uniqueness, persistence and scalability for the application in offline data archives serving data from archive medium such as file systems on Write Once Read Multiple (WORM) medium. In contrast, our approach relies on currently employed PID systems and incorporates location-independent data access in a distributed online environment using the full-featured Handle PID system.

III. Location-based PID Targets

Today's data dissemination is dominated by end-to-end connections, URLs, and DNS-backed domain names. When data is moved from one host to the other, it results in broken URLs and inaccessible content. To ease these problems, PIDs are used for long-term data access by providing a long-living identifier. When using a PID, the identifier is embedded into a medium such as scientific publications, books, or Web sites. To access the digital resources, *behind* the PID a resolution service is employed that uses the PID to provide a currently valid network location (target URL). Figure 1 illustrates how the target URL is adjusted to the current location when the data behind the PID is moved from one host to the other. Thus, PIDs reflect the current location of data and the identifier on the medium can remain unchanged.

The location-dependent data access through target URLs stored in PIDs also forms the chain of PID resolution. For this we have a look at the resolution chain presented in Figure 2. It depicts the fact that data access through PID with Hypertext Transport Protocol (HTTP) relies on different infrastructures and involves five levels from the PID resolution up to the data download. Even if the chain is shortened, e.g. by directly linking PID targets to IP-addresses instead of DNS-based host names, the problem remains identical: If PID-tagged data is moved from one host to the other, the PID access chain needs to be adjusted on one or even on multiple levels to provide valid PID resolution. If the adjustment is not done or partially incorrect, location-depended data access is impossible through PID. Defects can occur on every level. On level ❶ the PID HTTP resolution service can be temporary out of order. The target URL does not reflect the current location of the data in stage ❷. On level ❸ the Domain Name System (DNS) resolution can fail if a domain has been expired or DNS resolution fails due to misconfiguration. Level ❹ and ❺ are related to the network, but their functionality is also required for successful data access through PID. To detect broken PID resolution a check of every PID target and, thus, a successful resolution is necessary for judging the integrity. The check is done in many cases by evaluating the HTTP status codes like `404 - not found` that are provided by the data repositories software [10]. This can only be achieved by regularly checking *all* PIDs of an organization. This is very time consuming since typically robot or spider programs crawl all PIDs of a data owner. The crawling programs are programmed and operated by repository owners and not PID infrastructure providers. They provide an optional data quality assurance service.

The adjustment of target URLs has impact on different dimensions and is shared unevenly between the users – the PID operators and data repository owners. The costs and efforts behind URL adjustments have been accepted as part of the PID operation. They are considered *inevitable*, such as energy leaks in today's electrical grid infrastructure. The adjustment of target URLs is a shared effort on the side of data owners and dependent on the number of PIDs a data owner has registered.

It can be questioned whether the proliferation of e-Science already increases the effort necessary for PID maintenance. We thus have visualized statistics from DataCite (one of the largest PID infrastructure providers) in Figure 3. The assignment of new PIDs (*line*) massively increased, following a super-linear pattern [11]. An aggregation of the DataCite Statistics for successful DOI PID resolutions shows also a massive increase in PID-tagged data sets (*bar*) [12] [13]. With a massive increase of PID numbers, the efforts for maintaining PID targets will increase identically, as every assigned PID needs to be checked for validity to comply with PID infrastructure policy. It is not a question, whether the PID systems are scaling out sufficiently, but rather the data hosters are able to verify their location-dependent PIDs with a reasonable effort regularly. PIDs with location-independent targets decouple the growing number of PIDs from the efforts of maintaining PID target URLs.

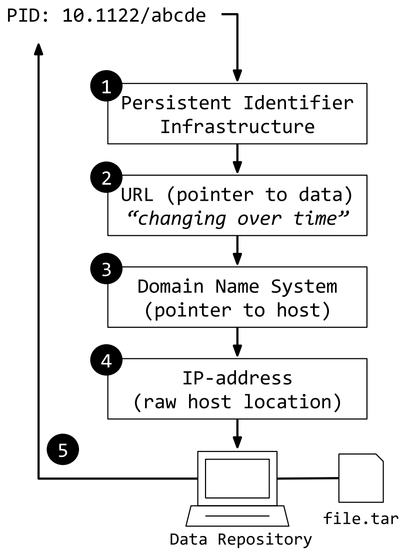
PID: 10.1122/abcde
1
Persistent Identifier Infrastructure
2
URL (pointer to data) "changing over time"
3
Domain Name System (pointer to host)
4
IP-address (raw host location)
5
file.tar
Data Repository

Figure 2: Data access through PID requires a working chain of services relying on unstable URLs.

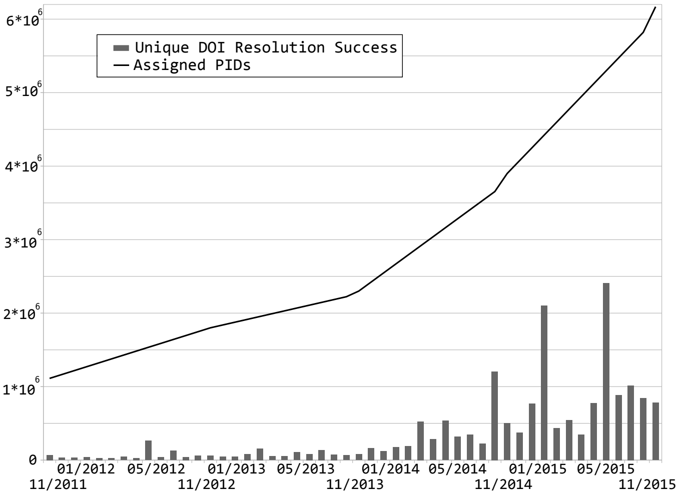

Figure 3: PID assignment and unique successful resolution for the DataCite DOI infrastructure between 11/2011 and 11/2015 based on data from [11] [12].

IV. Location-Independent Access

For location-independent data access, we propose two different techniques that allow access based on the content and not on the network location. We can thus show that our approach of location-independent persistent PID resolution targets works with various location-independent access technologies.

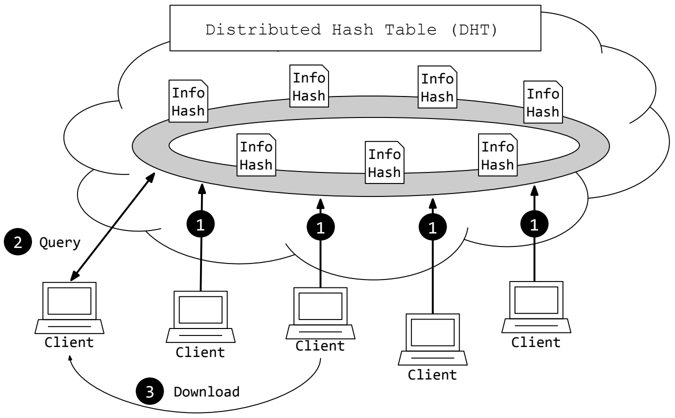

Figure 4: Accessing data from a DHT-controlled swarm using an infohash in BitTorrent.

We propose BitTorrent, a well-established location-independent access technology. It works on top of today's location-based networks with Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) [14]. In contrast to existing location-based data repositories that are subject of PID target resolution, BitTorrent technology uses a peer-to-peer approach supporting parallel downloads. With its latest features of Distributed Hash Table (DHT) and Peer Exchange (PEX), BitTorrent does not require central infrastructure to discover other network peers and localize files [15] [16]. Thus, data can be accessed with BitTorrent from every connected peer, as long as data is available online. For addressing data sets, BitTorrent uses *infohashes* that are computed as SHA-1 checksums on the content of the file. Every peer that possesses the infohash can download the data set from the BitTorrent *swarm* that consists of the peers offering the data set for download. The swarm arrangement and the *overlay network* for the specific file is computed for every download. In Figure 4, the BitTorrent file access is depicted. In step ❶, every node is sending its network address and the infohashes of the files ready for upload to the DHT. Then, a client can look up the infohash in the DHT (step ❷) to locate other peers that are able to serve the file that belongs to the infohash (or at least parts of the file). Through connecting to the peers in step ❸, the client is retrieving the file. This can be done simultaneously by parallel peer connection.

In contrast to BitTorrent, Named Data Networking (NDN) is a current research topic of location-independent data access using information-centric principles [17]. NDN is also featured in the location-independent PID approach presented in this paper to support a next-generation Internet technology. In NDN, data sets are enumerated through *Data Names* that form a hierarchical name space [17]. The working principle of NDN is shown in Figure 5. To access data from a client (step ❶) in the *Named Data* space, an interest data package is sent through the network. Based on the data name driven routing principles, the NDN network directs the interest through the network (step ❷ and ❸). If a NDN node is found that owns a named data set (step ❹), a data package is sent back along the interest path to reach the node, which stated the data request. Hence, NDN abstract from the network location and the data source remains opaque [17].

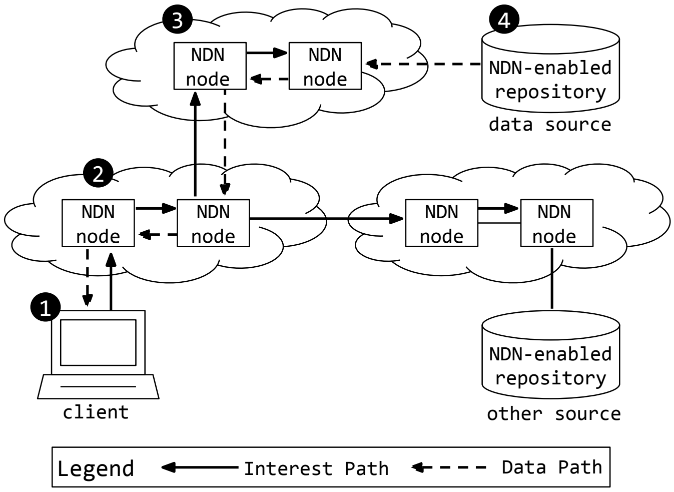

Figure 5: Accessing data using data names in NDN networks.

V. Magnet URI Scheme

For embedding location access information into PID, the Magnet URI scheme is used as transport container, which is a work-in-progress specification for *Magnet Links* [18]. Magnet Links can store extensive information on accessing resources in networks, like HTTP download URLs, mirror server information, or peer-to-peer access data. By applying this principle, Magnet Links are usable for describing digital resources and its content. But as a descriptive access format, Magnet Links need a storage medium to be present on. This could be a Web site with HTML content, an E-Mail message content, or – like in our use case – a Persistent Identifier. To provide persistent access to digital data encoded in Magnet Links for a time of years, or possibly decades, a medium is needed that provides these properties. Hence, persistency is not archived by the Magnet Links but provided by the PID System that ensures long living existence of access information through its infrastructure, policies and replication partnerships. If Magnet Links are stored on *perishable* media like Web sites, they do not provide any advantage over common URL access information. The conjunction with PID provides the additional benefit of long-term access from data hosted at suddenly moving data. Besides the encoding of access information, one design goal of Magnet Links is to integrate features made available by local utility programs seamlessly into the storage medium like a Web site, by following best practices among the Internet Engineering Task Force (IETF) specifications for URN [19]. Although its lack of specification, Magnet Links are supported by numerous peer-to-peer tools and are the de facto standard in large file sharing communities, such as BitTorrent [20]. In BitTorrent-enabled Magnet Links, access information for downloading files from a decentralized peer-to-peer infrastructure are stored together with optional metadata and suggested file names (c.f. Tab. I). To initiate a download with a Magnet Link from a Web browser, a pseudo-protocol handler for the Magnet Link URL format `magnet:?xt=urn:<System>:<Access Information>` is registered. It passes the information to a location independent download client.

Table I: Magnet URI scheme keys

Key	Name	Purpose
`as`	Acceptable Source	location dependent download URL
`dn`	Display Name	file name
`kt`	Keyword Topic	search key word
`tr`	Address Tracker	optional tracker information for BitTorrent
`xl`	Exact Length	size in bytes
`xt`	Exact Topic	location independent access information in URN-format

VI. Location-Independent Data Access through PID

For creating a persistent resolution target in a Handle PID that is based on the content of linked data set and not the network location like the target URL, we leverage the principles of location-independent data access with Magnet URI-encoded access information. In the first step in Subsection VI-A, we extend the Magnet URI scheme into the domain of NDN applications to support this cutting-edge data access technology. By this, we can encode all access information for BitTorrent and NDN as well as all systems listed in Table II into one uniform scheme.

Then, in the second step in Subsection VI-C, we propose an approach of embedding the Magnet Link URI scheme into PIDs of an unmodified Handle System. Starting from Subsection VI-E onward, we examine the PID resolution of PIDs with enabled location-independent access data.

A. Magnet URI Scheme Extension for NDN

Besides the already established usage of the Magnet URI scheme in peer-to-peer overlay networks that particularly allow BitTorrent location-independent data access, a next-generation access technology for supporting location-independent data access is integrated in our approach. We extend the Magnet URI schema into the domain of content-centric networks, enabling support for Named Data Network data access through Magnet Links. Thus, the Magnet URI schema is extended to store a NDN data name that identifies a digital object within a NDN network. With the data name, the NDN network can transport data from a source node holding the data back to the client node requesting the information through an interest [17]. The location of the data is not important, as long as it is attached to the network through a reachable NDN node. The data name is encoded through an extended `xt` key that holds besides the data name also a checksum of the data name to detect data corruption. The schema is `uid:ndn<DATANAME>.<CHECKSUM>`. Unlike current host-based networks that use a Public Key Infrastructure (PKI) to verify host identities through SSL certificates, NDN data cannot verified through a trustful data location. As a result, the Magnet Link also has to include the verification information needed to assure that the received content *is* the requested content. This is done by adding a cryptographic signature of the NDN content in a separate `<SIGNATURE>` field, which is part of the second `xt` key extension. Thus, verification needs to be done on content level using a public PKI, which is in the scope of current NDN research. To get the certificate needed to verify the data, the NDN access information for the certificate need to be added to the Magnet Link, too. To obtain the certificate with the public key, we propose the `xt` key `uid:ndnsec<SIGNATURE>.<CERT_DATANAME>` that allows the download of information for content verification. By this, access NDN access information can be encoded into a Magnet Link and also the genuineness of the obtained data can be verified through security information embedded into the Magnet Link. The extension we provide for the Magnet URI scheme supporting NDN is depicted in bold letters in Table II.

B. Embedding Magnet Links in Handle PIDs

For this integration of Magnet Links into the Handle PID System maximum compatibility is paramount, as data dissemination has a very slow change momentum, owed to billions of PID-tagged data sets. Hence, the usage of Magnet Links for location independent data access and its impact on the adaption in the Handle System is investigated. By design, Handle supports hierarchical data types, identified by UTF-8 named fields. The data itself is organized as indexed, typed key-value pairs that store sequences of octets, which are preceded by its length in a 4-byte unsigned integer. Like other PID systems, the Handle System provides a URL data type `0.TYPE/URL` [21]. Supplemental services such as PID HTTP resolvers, also known as Handle proxies, use the URL semantic to resolve PIDs into target URLs by using HTTP-forwarding with HTTP status code `303`.

Table II: Magnet Links Scheme Adopters (proposed schemes for Named Data Networking in bold font)

System	URN	Value
Gnutella2	`sha1`	file hash (SHA-1)
BitTorrent	`btih`	unique file identifier
Gnutella2	`tiger`	file hash (Tiger Tree Hash)
Kazaa	`kzhash`	file hash (proprietary)
NDN Access	**`ndn`**	**DataName and Checksum (SHA256)**
NDN Verification	**`ndnsec`**	**content signature & public key data name (NDN specs.)**

Unlike URLs, Magnet Links do not specify the data location but can be considered as URN-like data classification. Hence, the Handle PID data type `0.TYPE/URL` does not fit semantically for Magnet Links, because URL is a subset of URI [22]. As a result, an own data type `MAGNET` needs to be registered at a Handle PID server that should be capable of holding Magnet Links. To retrieve data from the PID via location-independent technology, a Magnet Link can be placed into the Handle. As Magnet Links fit into the UTF-8 encoding of Handle values, they can be placed without any further encoding.

For Instance, a valid DHT-enabled Magnet Link containing BitTorrent information for retrieving a file can be generated and stored in the MAGNET field of a Handle PID. Additionally, a Magnet Link-wrapped NDN Data Name can be placed into the MAGNET using URL escaping according to NDN name specifications [23].

C. Data Access Through PID with Persistent Resolution Targets

Handle PIDs that are equipped with a Magnet Link can be resolved like any other PIDs using the native Handle protocol. This can be done either by using that protocol on top of TCP or UDP, or via a HTTP-based proxy that answers resolution requests as a Web service. The *resolving* process for PIDs with persistent resolution targets works similar to the resolution process of location-based PIDs regarding the initial steps done within the Handle infrastructure. Hence, Figures 2 and 6 share the first initial step ❶, where the PID is resolved by the Handle infrastructure. In this step the Global Handle Registry (GHR) determines the Local Handle System (LHS), which is responsible for a specific local sub-namespace (Handle prefix). Then, the LHS looks up the requested PID in its database and returns the requested values to the client. For resolving using location-based data access, the value with the type 0.TYPE/URL is returned from the database and for resolving a with a persistent target location independent data access information in form of MAGNET is returned.

The new data access chain depicted in Figure 6 is different from the location-based data access using the target URL shown in Figure 2. With the Magnet Link of the PID acquired through the resolution process in step ❷ the data access is now handled by the overlay network in BitTorrent, or by the NDN network (step ❸). The PID resolving process is then a single redirection that leads to a starting download right after resolving, instead of multiple redirections using an entire chain of services for data access. These connections rely on peer connections based on IP-addresses for BitTorrent and node connections for NDN. Thus, the number of steps is reduced to three; also fewer layers and infrastructure are required for accessing the data. No central infrastructure is involved and the entire chain of location-based infrastructure is not needed anymore. The only requirements for data access is a running PID infrastructure and employing BitTorrent or NDN software for sharing the data online.

D. Creating and Resolving PIDs with Persistent Resolution Targets in Web Environments

For a convenient and smooth resolution of PIDs in the context of the World Wide Web, querying and resolving of Handle PIDs is realized by using a proxy service that accepts requests in HTTP(S) and resolves and maintains PID with the native protocol [21]. The Handle System maintainer, the Corporation for National Research Initiatives (CNRI), provides a Handle Proxy Servlet that offers HTTP-based resolving. The official Handle Proxy needs a small extension to resolve PIDs smoothly with HTTP into location-independent access information. This is done by changing the resolving mechanism from 0.TYPE/URL to MAGNET for the default value. However, non-modified Handle Proxies also work, when explicitly querying for MAGNET or using URL rewriting appending query parameters on the HTTP request.

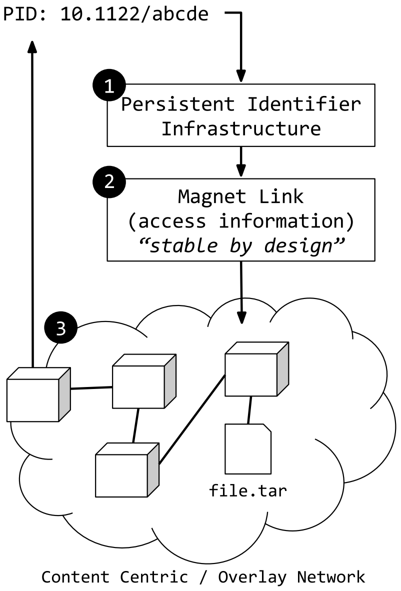

Figure 6: Location independent access through PID relies on stable content-based access information.

For offering a Web-based creation of PIDs that features a Magnet Link as persistent resolution target, a Web service is a reasonable choice. For every system that is offered for location-independent access technology, the Web service consumes either the native access information and encodes them into a Magnet Link or directly Magnet Links. Then, the Web service is registering a Handle PID at the LHS using the native Handle protocol and updates the Handle with a MAGNET containing the Magnet Link. In case of BitTorrent, the Web service consumes torrent files that contain along with the checksum and the name of the torrent also the infohash. These information are parsed and embedded into the Magnet Links as persistent PID resolution target using the current Magnet URI scheme description. For NDN data access, the Web service consumes the data name and the SHA-256 checksum of the data set. Furthermore, the Web service allows attaching NDN verification information to PID by extending the Magnet Link in the PID containing the cryptographic signature, as well as the data name to obtain the public certificate of the owner. The NDN access information are encoded into Magnet Link using our proposed Magnet URI scheme (c.f Subsection VI-A) in order to be embedded into the PID.

E. Data Access Through PID with Persistent Resolution Targets using a Web browser

Although, the Handle System can remain unmodified and the Handle proxy is only subject of optional minor modifications, the automatic resolution on HTTP clients provides solvable challenges. To provide a smooth resolution, a HTTP client like a Web browser needs to support multiple protocols through asking a protocol handler that determines the behaviour for resources outside HTTP sphere, like `mailto:` resources that are forwarded to the E-Mail program. For `magnet:` resources, the Magnet Link protocol handler is invoked [24]. Based on information stored in the Magnet Link value `Exact Topic (xt)`, the Magnet Link Handler selects the appropriate application and passes all information necessary for the download. Then the application is doing the heavy data download via the suggested access technology stored in the Magnet Link.

For invoking the process automatically in the Web browser, following steps are applied. When resolving HTTP-related URLs, the proxy-based resolver responds with HTTP status "`303 - See Other`". According to the HTTP standard RFC 7231, a `303` response to a `GET` request indicates that the server does have a representation of the target resource which can be transferred over HTTP [25]. In the case of normal location-based forwarding, the target `URL` field of the PID is resolved into a target server target URL, as the proxy server does not possess the data. In the case of PID holding location-independent access information, the `MAGNET` field is resolved into a forwarding to the overlay network of NDN name space. This HTTP-forwarding indicates that the Handle proxy server does not posses the data and it is not reachable via HTTP. Hence, the use of Magnet URIs in PID for HTTP-based resolution is in line with the HTTP standard and especially supports it with embedding additional information into the Magnet Link for a self descriptive data access.

VII. Implementation

A. Server Side

For verifying our approach, we set up an entire stack of software component for verification. On the server side, the stack consists of an unmodified Handle System (c.f. VII-A1) and a custom Web service called *PID-Burner* (c.f. VII-A2), which is able to create, update and resolve Magnet Link enabled PIDs based on our approach.

1) Local Handle System: The LHS that hosts the Handle PIDs under a specific prefix does not require any modifications in the source code to run our approach for storing and serving PIDs with Magnet Links. By default, the Handle System supports a list of preconfigured data types that are available as standard type set in every Handle System of a specific version. The list of pre-configures data types contains types for realizing typical PID scenarios with `EMAIL` and `URL` for location-based access. But also special scenarios like target URL forwarding based on users' geolocations. It should also contain a `URN` data type according to the Handle System documentation [26], but it – unfortunately – is not part of the preconfigured data types in the most recent version 8.1.0 [27].

As none of these preconfigured data types are suitable or working for storing Magnet Links, we use the well-designed extensible type system of the Handle System architecture to register a new data type `MAGNET`. By this, we only add a configuration item to the LHS that has no impact on the existing type system and runs on Handle legacy systems, too.

2) PID-Burner - Creating, Maintaining and Resolving of PIDs with Persistent Resolution Targets: The PID-Burner Web service allows creating, updating and resolving PIDs with Magnet Links embedded as persistent resolution targets. It is implemented from scratch, but incorporates libraries and frameworks for PID management, as well as BitTorrent libraries and initially created libraries for Handling NDN access information and processing Magnet Links with the extensions proposed. The PID-Burner engine is implemented in Python using the *Bottle Web framework* from creating a Representational State Transfer (REST) interface [28]. Besides the REST interface it offers a JavaScript-based user interface for the Web browser. As back-end for interacting with the Handle PID service the EPIC-API v2 Web service from European Persistent Identifier Consortium (EPIC) is incorporated to create and update PIDs [29]. For processing BitTorrent access information contained in torrent files, *libtorrent* Python bindings are used for extracting the necessary access information like the infohash, the file name and checksum [30]. NDN access information processing is done with a custom library, as well as the generation of Magnet Links.

For creating and updating PIDs with Magnet Links, the user can upload a torrent file that contains the BitTorrent access information. These torrent files can be created from original files in BitTorrent programs like Transmission [31]. NDN access information are uploaded as Java Script Object Notation (JSON) data structures and can store the data name and the checksum. The NDN data names has to be determined by the user depending on its NDN network topology. The checksum can be computed using any checksumming tool like OpenSSL. JSON is used for NDN access information encoding due to the lack of standardized access container formats in NDN that are comparable to torrent container files. The optional cryptographic verification information are attachable to the PID using NDN access information containing the cryptographic signature of the data and the NDN data name to retrieve the X.509 certificate of the data signer.

For resolving Magnet Links enabled PIDs with HTTP, the Web service implements a resolution functionality that is almost identical to the original Handle HTTP proxy by CNRI [32]. As described in Subsection VI-C, the resolution process does not depend on target URLs, but rather uses the PID value stored in the `MAGENT` data field of the PID. Hence, if a Web client asks the PID-Burner service for resolving, the Magnet Link with the access information is returned as HTTP status `303 - See Other` for existing PIDs. If the PID does not contain a Magnet Link, the resolution is done against the URL value of the PID and PID-Burner behaves identical to the Handle HTTP proxy service. The behaviour allows a maximum on compatibility towards the original Handle System also on HTTP resolution.

B. Client Side

End-users who are interested in using Magnet Link-enabled PIDs for data access need client software that is able to process access information for location-independent access. For BitTorrent-based access a client software is needed that is able to process Magnet Links. To simulate end-user environment we are using a Gnome 3.16.2 Desktop on Fedora 22 together with Chromium 47.0.2526.106 as Web browser running on Intel i5-2400 with 8GB RAM. As BitTorrent Software, we use an unmodified version of Transmission 2.92 [31] and use public BitTorrent infrastructure available to every Internet user (DHT bootstrap servers) for file download.

For NDN access, we provide an own Magnet Link adapter that process the NDN access information and passes them to the NDN download tools. This NDN Magnet Link adapter has been implemented in Python and is necessary to parse NDN Magnet Links the based on our proposes schema extension. For NDN data hosting and downloading, we use the experimental NDN Repo NG tool set that consists of NDN server and download applications for exchanging data over a NDN network [33]. The Repo NG tool set is running in a private testbed at Gesellschaft für wissenschaftliche Datenverarbeitung Göttingen mbH (GWDG) that consists of six NDN nodes.

VIII. Evaluation and Discussion

All approaches related to PID systems have to provide interoperability at its highest degree to comply with the slow changing momentum of the Handle Infrastructure.

First, the Handle System is a very large distributed infrastructure with shared responsibilities. The services consist of 1 000 servers in 75 countries, which are operated by hundreds of organisations. It currently holds over >100 million PIDs, owned by over 12 thousand registrants in 2015 [34] [35]. Hence, approaches that demand a fundamental change in the system have the challenges of convincing a large community. In contrast, our approach presented in the paper operates on-top of the infrastructure and has no impact on existing PID infrastructure. If LHS operators are interested in implementing PIDs with persistent resolution targets they just have to add a service that is able to resolve, maintain and update `Magnet` Links within PIDs as described in VII-A2.

Second, for data repository owners that use PID for registering their data and HTTP-based file distribution, our approach opens up new perspectives on data dissemination. With our approach they can combine the advantages of location-independent access, peer-to-peer networks and PID into a single concept. For this, they have to create access information for their existing files and provide a location-independent upload point such as a BitTorrent Upload server.

Although our concept offers many advantages, we have to investigate its performance. The evaluation has to be split into two parts; the first is the evaluation of the PID access. This is done by comparing the distribution of string lengths in PID target URLs against existing Magnet Link resources, checking whether Magnet Links will increase the size of PIDs. If Magnet Links increase the size, resolution performance will decrease. This can be explained with higher data transmission volumes and larger data sets that are to be handled by software stacks.

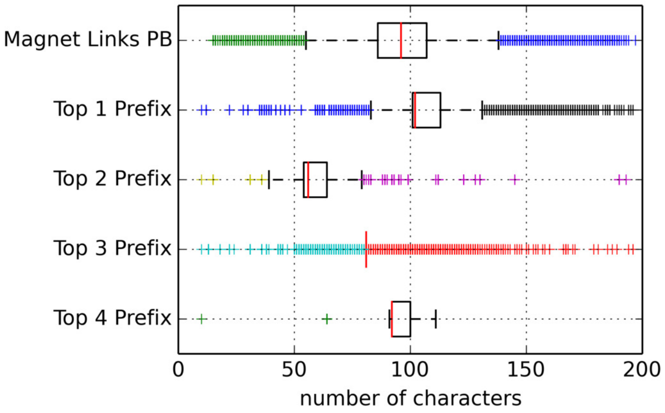

Figure 7: Distribution of string length for Pirate Bay Magnet Links and PID target URLs for frequently resolved Handle prefixes (95% data shown in plot, x-axis limited at 200 chars)

In Figure 7, the string length distribution for BitTorrent Magnet URLs aggregated from the *Pirate Bay* Web site is plotted as box-plot in row one. The Pirate Bay data set (row one) is chosen as a benchmark data set, as it forms one of the largest data accumulation for heterogeneous files exchanged by BitTorrent. For the string length analysis of the Pirate Bay data sets the tracker information have to be removed in order to provide clean analysis; furthermore, the tracker information are not necessary to perform a download using DHT techniques in BitTorrent. It consists of 1 643 194 Magnet Links in total that consist of BitTorrent access information in the form of infohashes and file names [20].

In comparison to the access information in Magnet Links, we now compare the string lengths of PIDs (row two to five). For this we use real-world data from network users who resolved PIDs at Handle Servers hosted at GWDG. The data set for PID resolving data consists of 1.294.668 target URLs in total for a time span between June 2014 and August 2014. The five Handle prefixes with the top-most resolution at the time span have been selected.

The box plots in Figure 7 show that real-world Magnet Link collections share a comparable string length distribution with Handle PID target URLs. To investigate the relation between PID size and the PID resolution performance, we measured the resolutions performance with PIDs of a defined size (c.f. Figure 8). The measurements were done at the LHS, hosting the Handle Prefix 11022 at GWDG with suppressed caching support to measure raw resolution times. It can be observed that the resolution time is slightly increasing for a number of milliseconds for extreme PID sizes with 32 768 characters. With the average PID size derived from the Pirate Bay data set of 97 characters (c.f. with the 2^7 bar in Figure 8), the impact of Magnet Link usage in PID is not perceivable to users. As a result, the PID replication and resolution can be expected similar to the existing target URL-based approach. Embedding Magnet Links in PIDs has no significant impact on the size

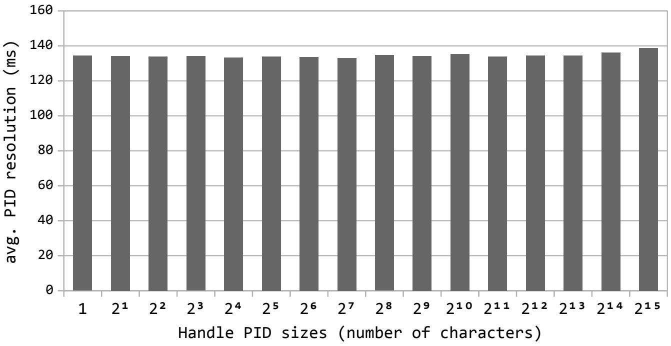

Figure 8: Average resolution time of Handle PIDs with different target URL lengths measured at the GWDG LHS for the Handle Prefix 11022.

of PIDs and underlines the practical usability of the concepts presented in this paper. Hence, Magnet Link enabled PIDs will not resolved into location-independent targets slower than current state-of-the-art Handle PIDs. The PID resolution time t_r can be assumed to be identical.

The second part of the evaluation is the data download speed provided from the location-independent data access. For BitTorrent and NDN the time for bootstrapping t_b the node and initiate a data download from the swarm is time intensive. Bootstrapping includes in the case of BitTorrent joining the DHT. The latter is very fast, although state-of-the-art DHT joining is accelerated through hard-coded bootstrapping servers. For NDN bootstrapping includes learning the node environments and the name routes. Different NDN bootstrapping algorithms are subject of current research [36].

After successful bootstrapping the advantages can speed up data transfer. If a peer is found that offer data access the download speed is at least comparable to existing location-based access that uses HTTP for download. If more than one peer is found, the bandwidth can be utilized to provide simultaneous data transfers, too. In this case the data volume v is split into n parts and the longest transfer $max()$ time of chunk is setting the overall transmission time as most time consuming partial transfer. Hence, the transfer duration d_{tr} of the first transfer through PID can be estimated as

$$d_{tr} = t_r + t_b + max\left(\frac{v_n}{v/sec}\right) \tag{1}$$

This parallelization results in higher download bandwidth and short transmission duration. Thus, for large download volumes the data access starting from PID resolution to download completion is faster. For small download volumes the completion time is longer in comparison to traditional data access through PID. Traditional location-based approaches using HTTP with no multi-source download capabilities have a transfer duration of

$$d_{tr} = t_r + \sum_{n=0}^{N} \frac{v_n}{v/sec} \tag{2}$$

where the duration is the sum of all partial volumes that are downloaded in serial.

Besides the performance aspects, it is observable that the PID usage of the PID infrastructure usage has an impact on the string length distribution of the target URLs. The top 4 prefix has a dense character distribution that is caused by the fixed pattern of the target URL, where only parts of the URL are varying. This is caused by the repository software that is managing the PIDs and uses IDs with similar length for ever data set. In contrast, the top four prefix shows a sparse distribution of target URL string length caused by almost non-systematic target PIDs. This distribution can be found by PID System operator that offer PID services to a large group of institutions like EPIC [29].

IX. Future Work

Despite our contribution to the PID efforts in the Handle System challenges provide open research questions for future work. The support for non-URL targets in HTTP-based Handle PID resolution could be moved into the existing Handle stack. By this, resources can be directly linked in the typical workflows that end-user facilitate in their Web browsers. Thus, location-independent access through PIDs in the Handle system could work with click, as simple as opening a PDF file. Fortunately, the Handle PID is very well designed and implemented by CNRI, and the source code is available.

A number of challenges arises from the usage of Magnet Links. The Magnet URI scheme originates in the file sharing community and has evolved in the past decade. Despite being a community effort, it is widely used by the most frequented file sharing search engines. For usage in the research data community, Magnet URI community contributions and drafts have to be collected and a standardisation effort needs to be started, e.g., as an initial IETF draft. The relation between Magnet URI and URN will facilitate the standardisation efforts and help to improve the reputation of Magnet Links.

Moreover, we propose using Magnet Links in the NDN community for encoding of full access information. Magnet Links could be one appropriate container format for transmitting NDN access information, which is closer to the original idea of object identification, based on URN, rather than location identification done with current NDN concepts based on URL. Similar approaches have been provided for other content-centric network like Open NetINF and proposed at IETF [37].

X. Conclusion

The integration of location independent data access in persistent identifiers is feasible without major modification of PID infrastructure. The Magnet URI scheme is suitable container for storing application independent access information inside Handle PIDs although it currently lacks IETF standardisation. As illustrated in our evaluation, their usage has no major implication on PID System operation and usage. Employing Magnet Link enables the creation of maintenance free PIDs, which do not require target URL adjustments and thus reduce the residual efforts on data repository owner sides. With the support of overlay network usage and NDN data access, *better* data dissemination is achievable with augmented resilience through multi-peer data hosting.

Acknowledgments

We like to thank Sven Bingert from the European Persistent Identifier Consortium for his support on the Handle evaluation infrastructure used for this paper. We acknowledge research funding by Deutsche Forschungsgemeinschaft (DFG) under grant SFB 963/2 "Astrophysical Flow Instabilities and Turbulence", projects INF.

References

[1] N. Paskin, "Digital Object Identifier (DOI) System," in *Encyclopedia of Library and Information Sciences*, 3rd ed. Boca Raton, FL: CRC Press, 2011, pp. 1586–1592.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Comm. Magazine*, vol. 50, no. 7, pp. 26–36, 2012. doi: 10.1109/MCOM.2012.6231276

[3] E. Van der Sar, "The Pirate Bay Tracker Shuts Down for Good," Nov. 2009. [Online]. Available: https://torrentfreak.com/the-pirate-bay-tracker-shuts-down-for-good-091117/

[4] K. Sollins, "Pervasive persistent identification for Information centric networking," in *Proc. of the Second Edition of the ICN Workshop on Information-centric Networking*. Helsinki, Finland: ACM, 2012. doi: 10.1145/2342488.2342490 pp. 1–6.

[5] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-oriented (and Beyond) Network Architecture," in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '07. New York, NY, USA: ACM, 2007. doi: 10.1145/1282380.1282402 pp. 181–192.

[6] C. Dannewitz, M. Herlich, and H. Karl, "OpenNetInf - prototyping an information-centric Network Architecture," in *Proceedings of the 37th IEEE Conference on Local Computer Networks Workshops 2012*, Clearwater, USA, Oct. 2012. doi: 10.1109/LCNW.2012.6424044 pp. 1061–1069.

[7] N. Fotiou, D. Trossen, and G. C. Polyzos, "Illustrating a publish-subscribe Internet architecture," *Telecommunication Systems*, vol. 51, no. 4, pp. 233–245, Dec. 2012. doi: 10.1007/s11235-011-9432-5

[8] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, "Secure Naming for a Network of Information," in *Proc. of IEEE Conference on Computer Communications INFOCOM*. San Diego, USA: IEEE, 2010. doi: 10.1109/INFCOMW.2010.5466661 pp. 1–6.

[9] S. Haun and A. Nürnberger, "Towards Persistent Identification of Resources in Personal Information Management," in *Proc. of the 3rd International Workshop on Semantic Digital Archives (SDA 2013)*, vol. 1091. Valetta, Malta: CEUR Workshop Proc., Sep. 2013, pp. 73–80.

[10] H.-W. Hilse and J. Kothe, *Implementing persistent identifiers: overview of concepts, guidelines and recommendations*. London: CERL, 2006.

[11] T. Cruse, "General Assembly 2016, moving DataCite forward," 2016. [Online]. Available: https://blog.datacite.org/general-assembly-2016/

[12] "DataCite Metadata Stats," Apr. 2016. [Online]. Available: http://stats.datacite.org/

[13] M. Fenner, "Digging into Metadata using R," Aug. 2015. [Online]. Available: https://blog.datacite.org/digging-into-data-using-r/

[14] B. Cohen, "The BitTorrent Protocol Specification - BEP 3," Oct. 2013. [Online]. Available: http://www.bittorrent.org/beps/bep_0003.html

[15] A. Loewenstern and A. Norberg, "The BitTorrent Protocol Specification - BEP 5," Mar. 2013. [Online]. Available: http://www.bittorrent.org/beps/bep_0005.html

[16] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *Peer-to-Peer Systems*. Berlin, Heidelberg: Springer, 2002, vol. 2429, pp. 53–65.

[17] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. Rome, Italy: ACM Press, Dec. 2009. doi: 10.1145/1658939.1658941 p. 1.

[18] G. Mohr, "Magnet URI - Draft Tech Overview/Spec," Jun. 2002. [Online]. Available: http://magnet-uri.sourceforge.net/magnet-draft-overview.txt

[19] K. Sollins and L. Masinter, "RFC 1737 - Functional Requirements for Uniform Resource Names," Dec. 1994. [Online]. Available: https://tools.ietf.org/html/rfc1737

[20] E. Van der Sar, "Download a Copy of The Pirate Bay, It's Only 90 MB," Feb. 2012. [Online]. Available: https://torrentfreak.com/download-a-copy-of-the-pirate-bay-its-only-90-mb-120209/

[21] S. X. Sun, S. Reilly, and B. Boesch, "RFC 3650 - Handle System Overview," 2003. [Online]. Available: https://tools.ietf.org/html/rfc3650

[22] T. Berners-Lee, R. Fielding, and L. Masinter, "RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax," 2005. [Online]. Available: https://tools.ietf.org/html/rfc3986

[23] Y. Yu, A. Afanasyev, Z. Zhu, and L. Zhang, "NDN Technical Memo: Naming Conventions - NDN, Technical Report NDN-0023, Revision 1," Jul. 2014. [Online]. Available: http://named-data.net/wp-content/uploads/2014/08/ndn-tr-22-ndn-memo-naming-conventions.pdf

[24] D. Thaler, T. Hansen, and T. Hardie, "RFC 7595 - Guidelines and Registration Procedures for URI Schemes," Jun. 2015. [Online]. Available: https://tools.ietf.org/html/rfc7595

[25] R. Fielding and J. Reschke, "RFC 7231 - Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content," Jun. 2014. [Online]. Available: http://tools.ietf.org/html/rfc7231#section-6.4.4

[26] CNRI, "4.9 Handle Value Line Format," in *HANDLE.NET (version 8.1) Technical Manual*, Nov. 2015, pp. 28–29. [Online]. Available: https://hdl.handle.net/20.1000/105

[27] ——, "Handle.Net software (HN_v8.1)," 2015. [Online]. Available: http://www.handle.net/download_hnr.html

[28] M. Hellkamp, "Bottle: Python Web Framework," Feb. 2016. [Online]. Available: http://bottlepy.org/docs/0.12/

[29] European Persistent Identifier Consortium, "pidconsortium/EPIC-API-v2," Mar. 2016. [Online]. Available: https://github.com/pidconsortium/EPIC-API-v2

[30] A. Norberg, "libtorrent python binding," 2015. [Online]. Available: http://www.rasterbar.com/products/libtorrent/python_binding.html

[31] Transmission Project, "Transmission," Mar. 2016. [Online]. Available: https://www.transmissionbt.com/

[32] CNRI, "Handle.Net Registry," 2015. [Online]. Available: https://www.handle.net/proxy_servlet.html

[33] A. Afanasyev, S. Chen, W. Shang, and J. Shi, "repo-ng: Next generation of NDN repository," Nov. 2015. [Online]. Available: https://github.com/named-data/repo-ng

[34] CNRI, "HDL® Identifier and Resolution Services," Oct. 2015. [Online]. Available: http://www.handle.net/factsheet.html

[35] International DOI Foundation, "DOI News - September 2014," Sep. 2014. [Online]. Available: http://www.doi.org/news/DOI_News_Sep14.pdf

[36] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: Named-data Link State Routing Protocol," in *Proc. of the 3rd ACM SIGCOMM Workshop on Information-centric Networking ICN*. New York, USA: ACM, 2013. doi: 10.1145/2491224.2491231 pp. 15–20.

[37] S. Farrell, C. Dannewitz, P. Hallam-Baker, D. Kutscher, and B. Ohlman, "RFC 6920 - Naming Things with Hashes," Apr. 2014. [Online].