# Open Decentralized Digital Object Identifier Systems

## 1. URL, URN, and URI Basics

**URL: Uniform Resource Locators** are pointers the internet location where a digital resource. Anyone can buy a URL from a Registrar and can then edit its Domain Name Service (DNS) record to point anywhere he wishes. The DNS is a Resolution Discovery Service (RDS) consisting of a distributed system of servers that associate the human-readable domain name (such as https://foobar.com) with an IP Address (for example 66.151.147.142). URLs can include parameters and paths that reach into the server attached to destination IP address. Note the URLs can be also used to point to resources addressable using other protocols such as mailto:// and FTP://, in addition to HTTP.

**URN: Uniform Resource Names** use various schema called name spaces to given digital resources identifying names. For example, URN:ISBN:0-545-01022-5 is an unique number identifying a specific book using the International Standard Book Number name space. URNs do not point to locations where the resource might be found.

**URI: Uniform Resource Identifiers** include both URLs and URNs as subtypes. In addition, URIs can describe a digital resource using various XML namespaces. For example, URIs can include tagged metadata regarding the author, title, publisher, edition, etc. of a book using a schema defined by a standards organization. URIs may or may not include persistent names, or a location where a resource can be found.

## 2. What are DOIs?

**DOI: Digital Object Identifiers** are a specific type of URI using a name space for URNs defined and administered by the International DOI Foundation. DOI URNs have the following format:

urn:doi:10.NNNN/SSS.SS...

where 10 indicates this is the DOI namespace, NNNN takes values of 1000 to 9999 and indicates one of up to 9000 **Registrants** the DOI Foundation has authorized to create and assign DOIs and the suffix /SSS.SS... can be any set of numbers assigned by the registrant to an object. Assigning registrants a unique number (NNNN) prevents accidental collisions between URNs created by different registrants.

DOIs serve three main purposes:

- To give a digital object such as a book, scientific paper, video, or other creative work a unique and permanent identifier.
- To describe the object using an XML metadata schema identified in the DOI Record. This is optional for URIs more generally, but is a fundamental purpose of DOIs.

- To point to a current location of the object. Unlike URLs, DOIs do not use a DNS-based Resolution Discovery Service (RDS). Instead, individual registrants provide access to DOI metadata which may include a current location in the form of a URL as a service.

More generally, the intention is to uniquely identify digital objects, describe them in a way the facilitates search, discovery, and indexing, connect the identifier to a location where the object itself can be found, and establish a common record of scientific work.

## 3. Problems with the DOI System

The main use-case for DOIs is facilitating access and organization of research products, including working papers, published articles, data sets, and books. Creating a clear and coherent record of research activity to further scientific advance is without doubt a worthwhile goal. Unfortunately, the DOI system fails in this both from an organization, and architectural standpoint.

- The DOI System is controlled by the International DOI Foundation, a consortium largely comprised of commercial publishers, commercial data providers, national research agencies, and scholarly and profession organizations. Its explicit goal is to safeguard intellectual property rights. The mission of restricting access to scientific work to those who can afford it at best presents a conflict of interest with maximizing scientific advance, and the broad use of knowledge. DOIs point discovery of relevant work based on keyword and other metadata elements to official pay-walled versions and away from open access work, or alternative versions that can be accessed freely.
- The DOI Foundation controls who is allowed to become a registrant. The right to create and assign DOIs is conferred by the existing registrants, who may have their own ideas about what legitimate publisher or content creator should look like. This also creates significant conflict of interest.
- Becoming registrant is expensive, several thousand dollars per year if an organization is luck enough to get approved. The marginal cost of issuing a DOI is in the range of $1. Again, this tends to exclude non-profit, open access, and non-commercial publishers.
- DOI Records are not *per se* in the public domain. Paywalls to access them are permitted. This means the not only are research works behind paywalls, but search and discovery can be controlled by DOI registrants.
- DOI Records are not necessarily persistent. If a title or object is purchased or transferred, its new owner may change the metadata in the DOI or stop maintaining it completely. Registrants in good standing or supposed to keep these records, and make them searchable under some terms, but this is difficult to check or enforce, and frequently not the case.
- From an architectural standpoint, DOIs are poorly designed as a way to maintaining a credible record of research works. Although each DOI number is unique as quasi-persistent, the metadata, and the actually digital object any embedded URL points to are mutable.
    - By policy, metadata can be altered by the registrant that controls the DOI, provided that a new timestamp is applied with each update. Thus, the “unique and persistent” DOI num-

ber is not associated uniquely, or persistently, with any set of descriptive metadata. Authors could be added or deleted, abstracts changed, dates of publication moved backwards or forwards. There is no way for a user to know if the metadata served by a registrant is the same as it was when the paper was first cited by an author, much less that it is any way genuine. The timestamp requirement is unreliable since its inclusion and value is entirely under the control of the registrant.

- Even if the metadata was faithfully maintained, there is no way to verify that the document, or other object that the URL or other locator points to, bears any relationship to the DOI's metadata, or is the same document that was archived when the DOI was created. Even if the registrant is behaving faithfully, the content connected to the URL in question may be under someone else's control.
- Registrants may delete the DOIs of work that does not suit the current political climate, or that was produced by unfavored researcher. Abstracts, and other data may be altered to cover-up mistakes. Registrants my choose to stop maintaining or providing access to DOIs purely for commercial reasons as the economic environment changes over the years.

In short, the DOI system is propitiatory, expensive, centralized, mutable, censorable, and unreliable, and not auditable. It is of no value in establishing intellectual priory, or any aspect of the integrity of the academic record.

# 4. An NFT-based DOI and URI System

Geeq has built an NFT system as part of the base-level protocol which addresses all the problems above, and creates new functionality. At a high level Geeq's NFTs work as follows:

- NFTs are fixed, immutable, records that are kept in the ledgers of a public blockchain.
- An NFT Record includes five key elements:
  - Metadata which can follow any schema or name space. Tagged DOI information is chosen at the time that the NFT is minted, and can never be altered thereafter.
  - A hash of a digital object (a PDF of a journal article, for example). Hash functions are publicly available algorithms that map digital files of any size in a 32 byte digest which is sometime as the objects "fingerprint". This is because, a given file will always "hash" to the same digest, but if even a single bit of a file is changed, the hash value will be completely different. Thus, the hash of a file uniquely identifies its contents down the last one or zero.
  - A public key. The public key is half of a public/private key pair. Public keys are connected to real world entities (publishers, organizations, and even individuals) through the Public Key Infrastructure (PKI) which is also the basics of SSL certificates, HTTPS and the system that allows browsers to tell you are connected to your bank, or to a fake Phishing site.
  - A digital signature using the secret private key part of the PPK pair described above. Signing any set of data with a private key makes it possible verify that none of the data has

been changed. In our case, signing the NFT binds the metadata, the NFT object hash, and the public key together in a cryptographically provable way. Any change in the data is immediately detectable.

- Each NFT has a unique Identifying Record Number (IRN) which equal to the hash of the four elements above. This server two purposes. First, it is used to identify ownership of specific NFTs, described in detail below. Second, it serves as the essential part of the URN of an NFT (equivalent to the "SSS.SS... part of DOIs).

- NFTs are produced using NFT Mint accounts on the blockchain that can be created by anyone, from large commercial publishers, to individual content creator. There are no gatekeepers or central authority, and no censorship is possible. Users can find the identity of the NFT issuer using the public key and the PKI, and decide for themselves if the data in an NFT of credible or of value.

- Ownership of NFTs is attached to standard coin accounts what might be owned by the author of a work, the owner of the NFT mint issued it, a scholarly or non-profit organization that helps maintain the scientific record, a commercial publisher, or the current copyright owner. Transfer of NFTs is straight forward, and inexpensive (on the order of .1¢).

- Ownership is signified the presence of "NFT Asset Subgroups" in a given coin record. These subgroups have two essential parts:

  - The NFT's IRN, which is the hash of the NFT data. This uniquely points to a specific NFT in the ledger.

  - A 32 byte metadata field that can be altered at will by an NFTs owner. Like a DNS Record, this mutable data fields can  can point to an IPv4 address (using only four bytes), or an IPv6 address (using sixteen bytes), or an ASCII URL of up to 32 ASCII characters. Alternatively, it could contain a domain and directory (such as https://as.vanderbilt.edu/economics) which would have the IRN of an NFT appended to form a full URL to exact document. For example:

https://as.vanderbilt.edu/economics/d72ba69914318667a318e06a081bf646d80b56c2e5ac2416c49d8763e08471bd

    Users could automatically hash the document being served at this URL to verify that it matches the hash in the address, which is also the hash that is locked immutably in the NFT.

- The cost of minting an NFT is less than 1¢, and the cost of maintaining an NFT record with 1000 bytes of metadata is on the order of 1¢ per year. NFTs can be endowed at the time of creation with funds to pay ledger rent for as long as desired, or can have funds added as needed. Users do not need to interact with the blockchain otherwise to maintain the existence of the NFT record. Alternatively, a society could maintain thousands of NFTs for its members for a few tens of dollars per year.

- The ledger that holds NFTs is public and decentralized. It can be searched and indexed by anyone. Any alteration of NFT records is detectable, and results in the audit and expulsion of offending node that failed to maintain the records correctly.

- Even if an NFT was deleted, a transaction proving its creation would still exist in blockchain, although not in the current ledger. The time of creation of an NFT is also provable noting the block number in which this transaction exists.

# 5. Conclusion

Using Geeq NFTs for DOIs, URIs, and to create URNs solves many problems.

# 6. Metadata

Metadata subgroups are datagrams that form the core of Geeq data services. They have the following four element format:

| **Metadata Subgroup Format Table** | |
|---|---|
| **Data Element Type** | **Description** |
| Count<br>(one byte) | User selectable value for the number of fixed length metadata elements in a metadata subgroup instance. |
| Fixed length metadata elements<br>(varies parametrically by chain instance) | Fixed metadata elements are byte strings of an exact length that is set by parameter in each chain instance. |
| Count<br>(one byte) | User selectable value for the number of variable length metadata subgroups in a metadata subgroup instance. |
| Variable length metadata subgroups<br>(variable byte counts) | Variable length metadata subgroups are byte strings with a user selectable length. |

Variable length metadata subgroups have the following two-element format:

| **Variable Length Metadata Subgroup Format Table** | |
|---|---|
| **Data Element Type** | **Description** |
| Count<br>(two bytes) | User selectable value for the number of bytes in the variable length metadata element to follow. |
| Variable length metadata element<br>(byte count = Count) | Variable length metadata element. |

Each chain instance has four selectable parameters that govern the limits of metadata subgroups:

| **Parameter Name** | **Description** |
|---|---|

| | |
|---|---|
| METADATA_FIXED_BYTES | Byte count of fixed length metadata elements: The exact required byte-count for each fixed length metadata element in a chain instance. Rendered as two bytes, allowing between 1 and 65536 total bytes, with a default of 64 bytes. |
| METADATA_FIXED_NUM | Maximum number of fixed length metadata elements: The maximum number of fixed length metadata elements for any given metadata subgroup in a chain instance. Rendered as one byte, allowing between 0 and 255 elements, with a default of 5. |
| METADATA_VAR_BYTES | Maximum byte-count of variable length metadata elements: The maximum byte-count for each variable length metadata element for all variable length metadata subgroups in a chain instance. Rendered as two bytes, allowing between 1 and 65536 total bytes, with a default of 1024. |
| METADATA_VAR_NUM | Maximum number of variable length metadata elements: The maximum number of variable length metadata elements for any given variable length metadata subgroup in a chain instance. Rendered as one byte, allowing between 0 and 255 elements, with a default of 5. |

Geeq metadata subgroups could accommodate DOI in a number of different ways. Using fixed length metadata is probably not the best approach. Even identifying URI schema as a separate item would not be advisable since the length varies. One possible use would be to have one short fixed metadata field, say 4 bytes, that flags an NFT as containing a DOI, URI, or other more general use case.

Variable length metadata subgroups are a better choice for this application. Major elements such as title, author/s and journal publication data, might be assigned to three distinct subgroups with appropriate XML tags. Alternatively, using one variable metadata field would be the simplest way to hold DOI or URI information. Parsing of the metadata using XML or namespace conventions would apply to the entire field and would not rely on serializing elements into separate fields. This would also create a uniform standard that the one and only metadata field contained all the URI/DOI, information and metadata and included information required for correct parsing.

# 7. DOI NFT Format

The Metadata Subgroup contains all the information and metadata that is required for DOIs. In turn, the Metadata subgroup is one of two main elements of an Attestation subgroup. The other is an attestation Hash, which in this case is a hash of the digital object that is described by the DOI metadata, and tokenized in this NFT.

NFTs are created by NFT Mints that can be created by anyone. This includes Journals, scholarly organizations, archives, and commercial publishers, which already use the DOI system, but also independent authors. organizations, companies and anyone else. There is no centralization or control, and censorship possible.

DOI's claim authority and credibly from the vetting of issuers, the high cost of becoming an issuer, and of issuing individual DOIs. In theory, users should verify that a DOI was created by an authoritative registrant, perhaps by using the resolvers provided by the publisher that created it, or simply believing that if the DOI pointed to metadata available through doi.org, that its authenticity has be preferred. In practice, users do not pay much attention. doi.org becomes a trusted information agent. This is problematic because doi.org and its registrants have many interests, including commercial ones, that are distinct and in conflict with those of users, researchers, and even science itself.

Geeq's approach is to require issuers to cryptographically sign NFTs and the metadata they include. The authority and credibility of these NFTs relies on the provable identity of the owner of the private key that signed the NFT. The corresponding public key is part of the NFT Mint's data, and the connection of this public key of a real world entity can be found through the existing Public Key Infrastructure (PKI). If a credible publisher or other agent claims the public key, then its credibly is inherited by the signed NFT. Even if an unknown agent signs an NFT, the existence of the record is still informative.

Metadata subgroups as described above are the core of Geeq's attestation subgroups and have the following format:

| **Attestation Subgroup Format Table** | |
|---|---|
| **Data Type** | **Description** |
| Attestation hash (32 bytes) | Attestation hash, which is the hash of a digital object such as an image, document, form, deed, or stock certificate being identified and described in a DIO NFT. |
| Metadata subgroup (variable byte counts) | Metadata subgroup conforming the current metadata parameter allowances. |
| Public key (32 bytes) | Public key that is the counterpart of the private key that signs the attestation. |
| Signature (64 bytes) | Signature that can be verified using the public key above. |

In turn, attestation subgroups are the core of Geeq's attestation, counterparty, and NFT payloads which are the component groups that go into application layer blocks.Minted NFTs becomes the main element of permanently immutable records in Geeq's application layer ledger in the following form:

| **NFT Application Layer Record Format Table** | |
|---|---|
| **Data Type** | **Description** |
| Record Type Identifier = APP_NFT (1 byte) | Identifies this record in the Application Layer Ledger as being an NFT Record. |

| | |
|---|---|
| Attestation subgroup (variable byte counts) | A single attestation subgroup. |
| Application layer account record number (32 bytes) | We use the convention that Hash(Attestation subgroup) = identifying record number of each NFT Application Layer Record. |

# 8. Ownership and Control

NFTs are owned and controlled by coin asset account records, and is indicated by the presence of an NFT Asset subgroup in the record:

| **NFT Asset Subgroup** | |
|---|---|
| **Data Type** | **Description** |
| Boolean type indicator (1 byte) | NFT Flag:<br>0 : if this subgroup refers to an ordinary NFT record.<br>1 : if this subgroup refers to a Counterparty or Stackable record. |
| Metadata element (32 bytes) | A fixed 32-byte Metadata field to facilitate certain use-cases for NFTs. |
| Application layer account record number (32 bytes) | The identifying record number the application layer NFT record that is controlled by this subgroup. |

- Boolean: Equal to 0, and indicating that the DOI NFT follow transfer and protocol logic of ordinary NFS and not those of the Geeqs counterparty and stackable NFTs.

- Metadata: This a mutable, 32-byte metadata field that can be altered at will by an NFTs owner. Like a DNS Record, it can point to an IPv4 address (using only four bytes), IPv6 address (using sixteen bytes) or am URL of up to 32 ASCII characters. Alternatively, it could contain only the root URL (such as https://as.vanderbilt.edu/economics), and that the document itself is named using its NFT application layer record, which is the hash of the NFT being referenced: Hash(Attestation subgroup). Thus, appending the hash rendered in hex would complete the URL to exact document:

https://as.vanderbilt.edu/economics/d72ba69914318667a318e06a081bf646d80b56c2e5ac2416c49d8763e08471bd

  Changing metadata in the asset subgroup would then move the pointer to a URL, Since the hex version of a hash is not easily human-readable or transcribable, the full address would probably need to be parsed from the NFT Asset Subgroup.

- Application layer account record number: As we say above, this equals Hash(Attestation subgroup) Thus, the content of the signed NFT is uniquely identified by its record number. The

chances that NFT subgroups would hash to the same value are vanishingly small. in the event that they do, Geeq's protocol prevent the duplication. In this event, de-conflicting the colliding NFT Subgroup is trivially easy to accomplish.

## 9. Custody

Three custody models seem most reasonable. First, a publisher or sponsoring organization (a department hosting a working paper series, for example) might hold all the NFTs it creates in its own coin account. This would relieve authors of any responsibility, and would make this service effectively invisible to authors. Second, custody could be transferred to authors who would then both control, and be responsible for maintaining, their DOI/NFT Records. Third, NFTs might be transferred to a third party such a library consortia, or profession organization.

The cost of minting NFTs is less than 1¢, and the cost of maintaining an NFT record with 1000 bytes of metadata is on the order of 1¢ per year. NFTs can be endowed at the time of creation with funds to pay ledger rent for as long as desired, or can have funds added as needed. Users do not need to interact with the blockchain otherwise to maintain the existence of the NFT record. Alternatively, a society could maintain thousands of NFTs for its members for a few tens of dollars per year.