**Scoring of candidate**

**Permanent Unique Identifiers (PUIDs)**

**in the context of the Global Information System (GLIS)**

Please score candidates as follows: **1**=Poor/Not supported, **2**=Good, **3**=Best

Note:

* The GBIF Secretariat does not have strong experience with ARK so does not comment on the suitability of ARK identifiers for all requirements.
* Our comments on DOI are based on experience with DataCite only.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Requirement** | **ARK** | **DOI** | **LSID** | **Comments** |
| 1 | **Uniqueness** |  | 3 | 3 | Provide equal uniqueness |
| 2 | **Permanence** |  | 3 | 2 | In the GBIF community, we don’t find LSID gaining the same traction as DOI |
| 3 | **Opacity** |  | 3 | 3 | The person minting the identifier can decide the opacity of DOI and LSID. |
| 4 | **Resolvability** |  | 3 | 1 | All use central resolution mechanisms. LSID resolvers are not widespread, and are not backed by similar resources as DOI networks. |
| 5 | **Discoverability** |  | 3 | 1 | DOIs are increasingly being used by sites and turned into hyperlinks and used in citation graphs. None are natively supported by Google – i.e. Google do not recognize and preprend a resolver to formulate an actionable URL. |
| 6 | **Security** |  | 1 | 1 | DOI and LSID require open access |
| 7 | **Scalability** |  | 3 | 2 | In terms of ease of adopting, DOI is far easier than LSID which requires access to DNS servers, while DOI through DataCite is a very basic API to develop against.  In terms of write throughput, LSID does not require registration of each ID in a central DB (only the authority) so minting IDs can be done at the speed of the authority.  Write throughput on DOI is limited to the speed the central authority can consume at, as we understand the DataCite architecture is a central master, with many replicated read-only slaves (please verify with DataCite if this is still accurate). |
| 8 | **Interoperability** |  | 3 | 3 | Interoperability can be considered at two levels i) resolution mechanism, and ii) resolution payload (metadata). DOIs are stronger on resolution interoperability being just a REST call. DOI is a generic object identifier, which can be used for anything (HTML page, document, binary file). It is simply a byte sequence to identify something. But metadata schemas vary across DOI authorities, so interoperability at metadata level requires understanding of metadata profiles and schemas.  LSID resolution is more complex but provides RDF based metadata so has potential to enable a greater interoperability as RDF based technologies can be used to link across domains (assuming shared vocabularies). But, in practice we don’t believe this is used anywhere. |
| 9 | **Compatibility** |  | 3 | 1 | DOI is being increasingly used, so we anticipate greater compatibility in time. Currently LSID and DOI are simply byte sequences. |
| 10 | **Content negotiation** |  | 2 | 3 | LSID uses content negotiation to support RDF based metadata. DOI has no such requirements but we believe could be implemented without breaking the specification. |
| 11 | **Accepted standard** |  | 3 | 1 | Use of LSIDs appears to be falling off. Before this group approached us, it had been some years since we had heard any mention of them.  DOI is increasingly being adopted, for citation purposes primarily. |
| 12 | **Acquisition and maintenance costs** |  | *3* | 3 | Costs are minimal (<€2,000 per year for DOI). Possibly more costs involved in maintaining an LSID system due to running a resolver |
| 13 | **Acceptance by publishers** |  | 3 | 1 | DOI is increasingly accepted mainly due to the fact that they are familiar due to citation of papers and there is strong interest in using them for data papers: data publishers wish to be cited in a similar manner. |
| 14 | **Popularity** |  | 3 | 1 | DOI increasing.  No new LSID adopters seen for some years |
| 15 | **Availability of tools** |  | 2 | 1 | Increasingly for DOI – in particular, citation graphs such as Thomson Reuters: webofknowledge.com . LSIDs not so well supported with tools |
| 16 | **Resolution service and multiple resolution** |  | 3 | 1 | DataCite supports multiple endpoints for redundancy purposes. We have not experienced any outages in the central resolution  LSID resolvers typically go offline and unnoticed for days |
| 17 | **Framework design** |  | 3 | 3 | DOI is basic and robust |
| 18 | **Metadata** |  | 3 | 2 | DataCite kernel appears stable and satisfies all the requirements we have had.  LSID is flexible, and allows for any RDF |
| 19 | **Relations** |  | 3 | 1 | DataCite have been reactive |
| 20 | **Identification of fragments** |  |  |  | Not used |