

	<p>persistent identifier linking infrastructure</p>	<p>web: http://resolver.net.au/hdl/102.100.272/0N8J991QH email: policy@pilin.net.au</p>
---	---	---

PILIN Glossary

Change Log

Version	Date	Status & Changes	Expression identifiers
V1.0	2007-12-20	RELEASE	PILIN/ SLGCW8JQH hdl:102.100.272/SLGCW8JQH

To cite the latest version of this work use <http://resolver.net.au/hdl/102.100.272/HHYMV8JQH>

To cite this version of this work, use <http://resolver.net.au/hdl/102.100.272/SLGCW8JQH>

Starred entries do not appear in the PILIN Ontology for Identifiers and Identifier Services.

Abstract

An **entity** is **abstract** if it is not **concrete**—that is, if it is not managed by some **identifier management system**. It is used to model the relation between concrete entities by representing what those entities have in common. The abstract entity is then said to be **realised** by the concrete entities.

In particular, an **abstract context** is a **context** defined by who is responsible for it, what purpose it serves, and what **policies** it requires, rather than how it is implemented. It is typically specific to an organisation. For instance, National Library **identifiers** belong to the National Library Abstract Context; they may be realised by one or more concrete contexts, such as the National Library **Handle namespace** and the National Library PURL namespace.


Accountable

A **component** is **accountable** if up-to-date information is accessible to outside **parties** on who has managed the component how in the past (**provenance**), and currently. This information is called **authority metadata**. If the component is accountable, parties can work out who has been responsible for the component in the past, and can query any changes to the component in the past. They can also work out who is currently responsible for the component, and query them if they find anything wrong or broken with the component.

Act

See **Actionable**.

Copyright © Monash University

	<p>This work is licensed under the Creative Commons Attribution-Share Alike 2.5 Australia License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/2.5/au/</p>
---	--

This work was created as part of the PILIN project. The PILIN project is funded by the Australian Commonwealth Department of Education, Science and Training, (**DEST**) under the Systemic Infrastructure Initiative (**SII**) as part of the Commonwealth Government's Backing Australia's Ability – An Innovation Action Plan for the Future (**BAA**) under the ARROW Project.

Action

An **action** is a process which can be applied to an **entity** to produce a result. Actions are triggered by **parties**, through a system (such as an **identifier management system**). **Services** are a type of action.

Actionable

An **entity** is **actionable** if some **action** can be applied to the entity. The action must not involve **curation**. (The cover term for such non0-curatorial actions is **act**.) For instance, an **identifier** is actionable if there exists a **service** that returns some metadata about the **thing identified**. (So "actionable" is not limited to **resolvable**.) But an identifier is not actionable just because there exists a service to update the identifier **resolution**: that is not what an end-user would expect to do with the identifier.

Administrator

See **Curation Boundary**.

*Aggregate Object

A **digital object** is **aggregate** if it is a **digital object** put together out of multiple preexisting **digital objects**, which are the constituents of the aggregate. For example an online course could be an aggregate object, put together based on several preexisting course modules. Aggregate objects force a policy decision on identifier managers: should the aggregate object have its own **published identifier**, and conversely, should the constituent objects have their identifiers published as well.

Alias

An **identifier** is an **alias** of a target identifier if they are **synonymous**, but the alias is managed and actioned to be dependent on its target. For example if hdl:102.100.272/XYZ is an alias of hdl:102.100.272/12672, then both identifiers are synonymous, **identifying** the same **thing**, according to the manager of the identifier management system. But hdl:102.100.272/XYZ is not managed independently of hdl:102.100.272/12672; instead, any changes in the **resolution** of hdl:102.100.272/12672 are reflected in the resolution of hdl:102.100.272/XYZ. Similarly, any requests to act on hdl:102.100.272/XYZ are treated by the identifier management system as requests to act on hdl:102.100.272/12672.

*Allocate

A **component** is **allocated** if it is transferred from one **authority** to another. For example, if I take over management of your **identifier**, then I have been allocated your identifier.

*Appropriate Copy

An **appropriate copy service** selects for delivery one out of several instances of a resource, according to user and other contextual parameters. If an **identifier** has **multiple resolution**, an appropriate copy service can select to resolve to the most appropriate instance of the **thing identified**, out of the instances nominated by each possible **resolution**.

*Arbitrary

An **identifier** is **arbitrary** if there is no direct relationship between the **name** and any relevant facts about the **thing identified**. An identifier is **arbitrary by policy**, if there is a policy to ignore any relationship between the name and the thing identified in using the identifier.

*Archival Lifespan

An **archival lifespan** is the lifespan for a **digital object**, during which either the object or information about the object is maintained, without the object necessarily remaining in regular active use. For example, once a digital object is archived to tape, its active lifespan is over, but its archival lifespan continues until the tapes are destroyed. An **identifier** can continue to point to a **thing** throughout its archival lifespan: even if the thing is no longer online, the identifier can provide access to information on how to retrieve the digital object from storage. Similarly, a **service** can continue to act on an identifier even if the digital object identified is not online, though it will not necessarily be able to retrieve the object. The archival lifespan varies by domain; planning for **persistent** identifiers typically quotes a lifespan of 25 years.

Associate

See **Identify**

Association data

Association data presents the association between the **name** and the **thing** identified in an **identifier**, in a form that can be **registered**. For **digital objects**, association data is often a retrieval key for the object from a **data source** (e.g. a **locator**). More generally, **resolving** an identifier returns some form of the association data for the identifier that can be used to gain access to the **thing** identified.

An identifier can have association data which does not constitute a **resolution**. This is important when the identifier **identifies** something which is not a thing retrievable online; e.g. an offline resource, or an abstract concept. In that case, association data is used as **discriminant metadata**: the association data is information about the thing identified, which can be used to distinguish (discriminate) the thing from other candidates for identification. So a dictionary definition of "love" counts as discriminant metadata (it can be used to distinguish the emotion of love from hate or anger). It is therefore association data, even though it does not meaningfully lead to resolution (a way of accessing a thing).

Association policy

An **association policy** is a **policy** deciding the range of **things** that can be **identified** by a **name**: the association of a name to a thing gives an **identifier**. Association policies are specific to the **context** that name is in, and make it possible for the context to match the purpose of interpreting identifiers.

Authority

An **authority** for a **component** is a **party responsible** for maintaining the component. This includes that the component has accurate content. For example, **identifiers**, **policies**, and **identifier management systems** all have authority.

*Authority Metadata

See **Accountable**.

*Autonomous

Identifier management for a range of **authorities** is **autonomous** if each authority **exclusively manages** their **identifiers** through their own **identifier management system**, but all the systems are **hosted** on their behalf by a central **party**.

*Centralised

Identifier management for a range of authorities is **centralised** if all authorities manage their **identifiers** through a common **identifier management system**, **hosted** on their behalf by a central **party**.

Cite

An **entity** is **cited** if its **representation** is communicated to an audience through some medium. The entity is **citable** if it can be cited. For example, citing the **identifier** ("Handle server 102.100.272", "XYZ"), "PILIN policy on citation") means coming up with an appropriate representation of the identifier (e.g. hdl:102.100.272/XYZ), and embedding that representation in a PDF. An entity is **Web-Citable** if the representation can be embedded in a document on the web, **Print-Citable** if it can be embedded in a print document, and **Speech-Citable** if it can be embedded in spoken text (e.g. read out over the phone). An identifier in particular should be usable in non-digital environments.

Component

A **component** is a cover term for anything that is modelled through the PILIN **identifier ontology**, and which can potentially be managed through an **identifier management system**. Components include **entities**, **actions**, **qualities**, attributes, and associations.

*Compound Digital Object

See **Aggregate Object**.

Concrete

An **entity** is **concrete** if it is managed by an **identifier management system**. Concrete entities are contrasted with **abstract** entities, which they **realise**. For example, if the National Library has **identifiers** on the PURL **namespace** "<http://purl.nla.gov.au/id>", then that namespace is a concrete **context**, realising the abstract context "identifiers in the National Library". The **identifier management system context** is a type of concrete **context**.

Contains

A **context (enclosing context)** **contains** another context (**subcontext**), if all **labels** in the enclosing context are also in the subcontext, and all **policies** enforced by the subcontext are also enforced by the enclosing context.

Context

A **context** differentiates **labels** used for distinct purposes and with different authorities. The combination of a a label and a context for the label gives a **name**, and the same label can be used in different contexts to give different names; any label is necessarily **unique** in its context. Contexts impose **policies** on the labels in the context, including **association policy**: so the context of a name determines how the name is interpreted as an **identifier**. Contexts can also impose policies on what labels are allowed in the context (label format policy); and policies on who can perform what **actions** on a name in that context (access policy).

Contexts may be identified by one or more **context identifiers**. The identifier for a context has a name in a context of context identifiers. There is a defined context instance of "known naming systems", preventing infinite recursion of contexts.

*Core Service

A **core service** in the context of **persistent identifiers** is a **service** targeted directly at the maintenance of persistent identifiers. Core services include **Register, Update, Delete, and Resolve**. Core services are distinguished from **value-added services**, which are enabled or enhanced by core services, but are not primarily intended to maintain identifiers.

Create

A **thing** is **created** if it is brought into being. Things can be created without being **registered**; for instance, geographical coordinates can be created for use as an **identifier**, without those coordinates being registered explicitly in an **identifier management system**.

*Curation

Curation describes a range of activities and processes done to **create**, manage, maintain, and validate a **component**. All such activities are referred to as

curatorial. Curation is undertaken by the managers of a component, and is not accessible to end-users of the component. Curation of a component is undertaken in preparation of **publishing** the component; if the component is already published, curation can lead to a new event of publishing (e.g. a new release).

*Curation boundary

The **curation boundary** is a boundary defined by access to actions for **curation**, and is used to model **publishing**. A **digital object** within the curation boundary is only accessed through curation actions triggered by authorised parties (termed **administrators**); this means that the object is not accessible to end users. The object is not considered published, even if there is a large number of authenticated users authorised to do curation actions. Curation of the object is undertaken with the aim of improving it to the point where it is ready for publishing.

An object is moved outside the curation boundary once it reaches that point: this consists of providing access to the object through an action that does not involve curation. Such actions are the actions that end-users use on the object, so this constitutes publishing the object to end-users.

The curation boundary can be crossed without necessarily providing public access to the digital object, and in fact without necessarily providing direct access (**resolution**) to the object: releasing metadata about the object through a **service** still counts as crossing the curation boundary.

Data source

A **data source** is a tool for the storage and management of data by a **party**. The data source exposes **services** allowing access to that data by other parties.

*Deduplication

A process to ensure that only one instance of a thing is managed by a system, including a **data source**. **Universal** identifiers may be used to ensure deduplication: if there is only one identifier per thing, then the identifier will not permit differentiation between two instances of the same thing.

Deregister

To deregister an entity is to delete a registered entity from an identifier management system. Deregister is the opposite of Register, rather than Create.

*Devolved

Management of a range of **identifier management systems** is **devolved** if each system is **exclusively managed** by its own **authority**, with its own **hosting** of system infrastructure. There is no coordination at all between the management systems.

*Digital Object

A **digital object** is a **thing** that may be transmitted through electronic networks, and that can be stored and managed on a **data source**.

*Discriminant metadata

See **Association Data**

Enclosing Context

See **Contains**

Encoding Scheme

A mapping of **labels** to labels, preferably one-to-one. An encoding scheme may be appropriate to **representing** labels in a particular **medium**.

Entity

An **entity** is any **thing** used to define a core concept in an **identifier management system**. Instances of entities may be stored and managed in the identifier management system.

Equivalent

Two **identifiers** are **equivalent** if they both **identify** the same **thing**.

*Expression

See **Version**

*Exclusive management

A **party** is said to **exclusively manage** a **component** if they are the only party authorised to manage it.

*Federated

Identifier management for a range of **authorities** is **federated** if each authority **hosts** their own **identifier management system**, but there is **shared management** of the **identifiers** by all members of the federation.

*FRBR

The **Functional Requirements for Bibliographical Records** is an **influential information model** for bibliographical things, which has also been applied to digital objects.

*Global

In some usage, a **component** is referred to as **global** if it is usable outside a private or proprietary domain. In the context of **digital objects**, it refers to the component being accessible through a readily available protocol such as HTTP. A component can be used globally but still be subject to authorisation.

*Handle system

The **Handle** system is one of a number of **identifier management systems** available for online **identifiers**. It is a robust and flexible system, which allows different kinds of **resolution** and metadata to be associated with identifiers.

Homologue

Two **concrete identifiers** are **homologues** if they **realise** the same **abstract identifier**. Homologues are an easy way for an **authority** to manage multiple **identifier management systems**: the **labels** and the **things identified** remain the same, and only the **concrete contexts** differ. For example, <http://purl.oclc.org/NET/rfc/4321> and <http://www.ietf.org/rfc/rfc4321.txt> both identify RFC document 4321, and can be seen as the same label, "4321", in a PURL and a static URL context. They therefore both realise the abstract identifier "4321 in the RFC numbering scheme".

*Hosting

An institution **hosts** an **identifier management system** if the institution as a corporate **party** is the **authority** for the system. **Own hosting** requires that the institution provides the infrastructure to make the identifier management system functional and **trustworthy**. An institution may instead choose **external hosting** for the identifier management system: in that case the institution retains responsibility for managing the **digital objects** in the management system, but delegates the infrastructure for the system to another party.

Identifier

An **identifier** is the association of a **name** with a **thing**. A name may only be associated with one thing at any time, and the name is said to **identify** the thing.

Identifier Management System

An **identifier management system** is a collection of definitions, **information models**, **policies**, and **data sources**, used to manage **identifiers**. An identifier management system has a **curation boundary** associated with it, and an **authority** acting as its owner.

Identifier Management System Context

Deploying an **identifier management system** defines an **identifier management system context**: this is a single **concrete context** specific to that deployment. The purposes, **labels**, **policies** and **authorities** of the

identifier management system context are defined with reference to the system itself.

Identifier ontology

An **identifier ontology** is an overall model on how to describe an **identifier management system** generically. It uses **entities, qualities, actions**, and the relations between these. The PILIN project has devised its own Identifier Ontology, on which much of this glossary is based.

Identifier System Model

An **identifier system model** is the representation of a specific **identifier management system**, using concepts drawn from an **identifier ontology**.

Identify

A **thing** is **identified** by a **name** if the name and the thing together form an **identifier**. The name is **associated** with the thing identified, although that is not the only form of association possible in an **identifier management system**. (For instance, **authority metadata** is also associated with a name.)

Information Model

An **information model** is a model of **things** in a domain, their properties, and the relations between them. The choice of what things to **identify** in an **identifier management system** is informed by an information model.

*Interoperable

A **component** is **interoperable** if an action can operate on the component from outside the **curation boundary** of the **identifier management system**. The action must follow a well-defined interface, which is known outside the curation boundary. If a component is not interoperable, then only the identifier management system's own infrastructure can be used to operate on it. If the action uses a publicly documented interface through an open protocol such as Web **services**, it is interoperable.

Label

A **label** is a symbol that can potentially be used as a **name**. In online **identifier management systems**, labels are typically strings.

*Locator

A **locator** is a string giving the location of a **digital object** in a **data source**, and can be used as a retrieval key to gain access to the object. A URL is an example of a locator, although not all http: URIs are locators. A locator is specific to a data source, and cannot be used to access an instance of the digital object in a different data source. A locator can be used as an **identifier**; but it will usually not be **persistent**. Persistent identifiers often **resolve** to the current locator(s)

of the **thing identified**. This uncouples the persistent identification of a resource from the current retrieval key for the resource.

*Loose

An **identifier** is **loose** if the **thing** it identifies can change over time. For instance, the identifier "latest version of the PILIN citation policy" points to a single well-defined thing; but the content of that thing can change over time as new versions of the policy are released. What stays constant for a loose identifier is the role that the thing identified fulfils (e.g. "latest", "local"), rather than its content. Identifiers which are not loose are **rigid**.

Manage

An **identifier management system manages** an **entity**, if it is used to record and update **representations** of the entity and its attributes, which **parties** can then consult.

*Manifestation

See **Presentation**

*Meaningful

An **identifier** is **meaningful** if there is a direct relationship between the **name** and a relevant fact about the **thing identified**.

Medium

A vehicle through which a message is transmitted from a sender to a receiver.

*Mint

In some usage, the term **mint** is used to refer to the entire process of **creating**, **registering**, and **publishing** an **identifier** through an **identifier management system**.

*Multiple Resolution

An **identifier** has **multiple resolution** if the **association data** returned for a request for **resolution** can be used to access the **thing** identified from more than one location. In typical **identifier management systems**, this means that the identifier can be resolved to more than one URL. Multiple resolution is only possible if the thing identified is not a specific instance of a **digital object** at a given location, but an abstraction (e.g. "any digital object with this content"). Any of the **locators** returned by the identifier allows a valid resolution of the identifier, and allows the user the choice of which locator to access.

Name

A **name** is the association of a **label** with a **context**. The label must comply with any policy requirements that the context makes for the association to be valid. The same label paired with a different context gives a different name.

*Namespace

*Namespace

A **namespace** is a **concrete context** for names, used to disambiguate **labels**. It is typically included in the **representation** of names as a prefix. The namespace is typically the same as the **identifier management system context**.

*Naming Authority

A **naming authority** is an **authority** over a system for managing **names**. An **identifier management system** manages names as part of managing identifiers; and an identifier management system defines its own **concrete context** for those names. Therefore "naming authority" is often used to refer to the concrete context of an identifier system.

*Obfuscated

An **identifier** is **obfuscated** if it is **meaningful**, but the meaningfulness of the identifier cannot be inferred by inspection. The meaningfulness of the identifier can only be inferred if one is aware of the process through which the **name** has been generated.

*Opaque

An **identifier** is **opaque** if it may be **meaningful**, but the meaningfulness of the identifier cannot be inferred by inspection. Any meaningfulness of the identifier can only be inferred if one is aware of the process through which the **name** has been generated.

Party

A **party** is a person or a group which can act as an **authority** over a **component**, and which can participate in various processes, including managing or using identifiers.

*Patch

An **identifier** is **patched** if its **name** is changed, for whatever reason. The old name in the identifier is deprecated in favour of the new. Patching means that all **cited** instances of the identifier must be updated to the new name to maintain identifier functionality: this is increasingly unrealistic the more widely the identifier has been **published** and used.

Persistent

A **component** is **persistent** if it is managed and maintained for a defined timespan. Maintaining the component includes ensuring that its **published**

content (such as its **association data**) is valid at all times. The timespan for persistence need not be indefinite. Persistence can apply to other **qualities**; e.g. persistence of actionability, of accountability, of association (i.e. the association between **name** and **thing identified** in an **identifier**), of functionality (i.e. the type of thing being identified remaining the same). Normally when an identifier is called persistent, persistence of association is meant.

Policy

A **policy** is a set of rules regarding **components**: they describe states of affairs that should be true of those components. Policies are set by an **authority**. Policies may be enforced in an **identifier management system** for the components it maintains: this means that the systems ensure that the rules defined in the policy are true of the **entities, actions** and **qualities** maintained through the system. ***Policy Domain**

A **policy domain** is the scope over which a **policy** can be enforced. For instance a policy may have its domain restricted to a certain format of **labels**, or a certain type of authorised user. Policies set by an institution have their policy domain set by default to the members and property of the institution.

Preferred Identifier

An **identifier** is **preferred** according to an **authority**, if that authority guarantees its **persistence** over other **equivalent** identifiers. The authority recommends that the preferred identifier should be **cited**, rather than any other equivalents. For example, a **digital object** in a repository may be identified by a title, a URL **locator**, and a PURL. The repository manager guarantees persistence only for the PURL. Therefore the PURL is the preferred identifier for the object, according to the repository manager.

*Presentation

A **presentation** of a **digital object** is an abstraction fixing both the content and the appearance of the digital object. Two instances belong to the same presentation if they have the same content and the same appearance; they belong to different presentations if they have different appearances, even if they have the same content. Presentations may differ by file format, schema, formatting, branding, and so forth. **Manifestations** in the **FRBR** model are a type of presentation.

Preservation

Preservation is the activity of ensuring that a **thing** remain accessible and valid. Preservation is a **curatorial** (i.e. **curation**-driven) activity. Preservation of digital objects is a separate activity from maintaining **persistent identifiers** for objects, undertaken by different **parties**. A **digital object** must remain preserved over a defined lifespan in order to have a valid resolvable persistent identifier.

*Provenance

Provenance is the history of how a thing has been managed over time. Data documenting the provenance of a **digital object** are part of the **authority metadata** for that object, and can be used to establish **accountability** for any changes in the object.

Publish

A **component** is **published** when access to it is enabled through at least one non-**curatorial action** for a given user profile. Publishing a component is modelled through the notion of a **curation boundary**. The activity of publishing a **digital object** is distinct from the activity of publishing an **identifier** for the object: an object is typically published through a resolvable identifier, but the two publishing events need not coincide.

Quality

A **quality** of an **entity** defines how an entity should behave in an **identifier management system**. Identifier management systems are profiled to realise certain desirable qualities for their **identifiers**, such as persistence, accountability, and resolvability.

Realises

A **concrete entity realises** an **abstract** entity if the two entities are **equivalent** in some way, and the concrete entity can be used to fulfill requirements made of the abstract entity. For example, an abstract **identifier** does identify a **thing** and is citable; but it is not **resolvable** or **accountable**. A concrete identifier **synonymous** with the abstract identifies the same **thing** (because it is synonymous), and is **citable**; so it fulfils the same requirements. But it is also resolvable and accountable, so it is used in actual systems instead of the abstract identifier, as its realisation. (A concrete identifier must have the same **label** as the abstract identifier it **realises**.)

Register

A **party registers** a **component** if they cause it to be **registered** in an **identifier management system**, and the component is not already registered there. Registering is a **curatorial** action.

Registered

A **component** is **registered** if it is maintained (i.e. stored or represented) and managed in an **identifier management system**.

*Rehoming

An **identifier management system** and its components are **rehomed** if the **authority** over the system (and potentially its physical location) is transferred from one **party** to another. A rehoming plan is necessary to ensure persistence of **identifiers** past changes in identifier management. As long as an identifier system is not dependent on **locators** as identifiers or identifier **resolvers**,

rehosting does not affect the actionability of identifiers. E.g. if a Handle server is rehomed from Melbourne to Monash, the **context identifier** need not change, so the Handles transferred need not change. However if a URL server is rehomed from Melbourne to Monash, the context identifier (domain name) will typically need to change.

The ability to cope with rehosting is a difference between DNS-based and non-DNS-based identifier schemes. **Resolver services** still have Web-Resolvable locators, but non-DNS schemes typically have **centralised** resolver services to address this problem. For instance, calls to a Handle resolver service at <http://handle.unimelb.edu.au> will no longer persist if the resolver service is rehomed to Monash; but using a centralised resolver service like <http://hdl.handle.net> mitigates this risk.

*Reliable

A **component** is **reliable** if the component remains continuously **actionable** for a defined range of actions, over a non-trivial timespan. For instance, an **identifier** is reliable if it remains resolvable without interruption for the duration specified in a service level agreement. This requires adequate IT infrastructure to be provided, including support for access, performance, and backups. Reliability is an important component of **trustworthiness**.

Most actions on identifiers involve both a **resolver service**, provided by the **identifier management system**, and an external **service** using the **resolution** data; e.g. a content delivery system. The identifier should be considered reliable so long as the resolver service is reliable, even if the external service is not. For example, if the identifier resolves correctly to a **digital object** on a repository, but the repository is down, the failure in reliability is the repository's, and not the identifier management system's. It is the data manager's responsibility to mitigate that risk, e.g. by mirroring the object and allowing resolution to mirrored copies.

Representation

An **element** can have one or more **representations**, which can be communicated to an audience through some **encoding scheme**. Communicating a representation of an element is **citing** the element. The **representation** of an element is a single symbol, whatever the internal structure of the element. For example, we have defined an **identifier** as an association of a **label**, a **context**, and a **thing identified**; but the representation of an identifier is a single symbol, presenting the **name** through a combination of encodings of the **label** and the context (the latter often optional). So (("National Library Names", "XYZ"), "PILIN citation policy") can have the single representation "hdl:102.100.272/XYZ".

Reserved

An **element** is **reserved** in an **identifier management system** if it has been marked as having a "temporary" or "in use" status. A reserved element cannot yet be published. Typically a reserved element may not undergo **curatorial** actions either until it is **allocated** to an identifier manager; for example, a set of **labels** may be reserved for use in **identifiers**, but are not actually used in identifiers until they are allocated to some identifier manager.

Resolve

An **identifier** is **resolved** by providing information on how to access the **thing** it identifies. This information is the **resolution** of the identifier: it is the output of the Resolve action. The resolution must be consumable by other processes, for instance a Content Delivery process. An identifier is **Internet-Resolvable** if the information on how to access the thing identified can be requested and consumed through a well-defined Internet application protocol, and **Web-Resolvable** if that protocol is a defined web application layer protocol. An example of the latter is a Web **service** query on a Handle, with its request in HTTP GET and returning a URL.

Resolution in general operates on **association data, registered** with the identifier in an **identifier management system**. Resolution is a non-**curatorial** action: a resolvable identifier is **actionable**.

*Resolver

A **resolver** is a system which provides **resolution of identifiers**; that is to say, the resolver **resolves** identifiers, returning information based on **association data**. Resolvers are typically **identifier management systems**.

Responsible

A **party** is **responsible** for a **thing** if they are committed to ensuring the maintenance and accuracy of the thing. This makes that party the **authority** for the thing.

Retrieve

To obtain or gain access to a representation of a **thing** through its **identifier**. Retrieve presupposes **resolve**: it acts on the resolution data for the identifier.

*Rigid

An **identifier** is **rigid** if it **identifies** exactly the same **thing** at all times, with the same content (where applicable). For example, an identifier for Release 1.0 of a document is rigid, so long as that release is frozen, and its content does not change. Identifiers which are not rigid are **loose**.

*Service

A **service** is an **action** operating on a **component** through some defined protocol for requests and responses, and hosted by a computer system.

*Shared management

A **party** has **shared management** of a **component** if they are not the only part authorised to manage it. Shared management requires infrastructure to coordinate between the different managers of the component, to prevent inconsistency.

Subcontext

See Contains.

Synonymous

Two **identifiers** are **synonymous** if an **authority** claims that they are **equivalent**.

Thing

A **thing** is what anything that can be talked about is; in particular, it includes whatever can be **identified** with an **identifier**.

*Transferable

An **identifier** is **transferable** from one **identifier management system** to another if an **equivalent** identifier can be **registered** in the new system with the same functionality as the original. This means the new identifier must **identify** the same **thing** as the old, and allow the same range of **actions** to operate on it.

*Transparent

An **identifier** is semantically **transparent** if it is **meaningful**, and the meaningfulness of the identifier can be inferred by inspection.

*Trust Boundary

A **trust boundary** delimits a set of **parties, services, data sources** and systems which may be involved in processes without need of authorisation. If a party, service, or system is outside another party, service, or system's trust boundaries, then the two may only participate in the same process after appropriate authorisation is established.

Trustworthy

A **component** is **trustworthy** if an end user can be confident that their use of the component satisfies certain expectations. Those expectations typically include **reliability**, accuracy, and **accountability**.

*Typed

An **identifier** is **typed** if the kind of **thing** it is allowed to **identify** is fixed by an **authority**, through a **policy** implemented in an **identifier management system**. Typing an identifier ensures **persistence** of functionality (the identifier will interact in similar ways with action whatever it currently identifies), and is a weak mechanism for ensuring persistence of association.

Unique

A **component** is **unique** if there exists one and only one instance of the component within a given scope. For example, if a **label** is associated with a **context** in a **name**, the label must be unique in the context. If a **name** identifies a **thing** in an **identifier**, the thing must be unique in the identifier.

*Universal

An **identifier** is **universal** if it is **unique** in a **context** in **identifying a thing**: there are no **equivalent** identifiers in that context to choose from. A universal identifier allows all actions operating on a thing to **interoperate** using the same identifier, and makes **deduplication** possible. Universality is impossible over the context of all known naming systems, but various strategies attempt to emulate it, including **preferred identifiers**, and **services** mapping between **synonymous** identifiers. Within the context of a single identifier management system instance, on the other hand, identifiers are often universal.

*Value-Added Service

A **value-added service** in the context of **persistent identifiers** is a **service** enabled or enhanced through the use of persistent identifiers. It is distinct from a **core service**, which is targeted directly at the maintenance of persistent identifiers. Value-added services lie outside the domain of **identifier systems**, and are not typically hosted by identifier systems; they instead consume the core services provided by identifier systems.

Verify

A **party verifies** a **quality** value for an **entity** in an **identifier management system**, if they confirm that the value of the quality reflects a true fact about the world. The usual target of verification is that an **identifier** will **resolve** to valid **association data**. Qualities may be **verifiable** (verification is feasible), and **verified** (verification has taken place successfully).

*Version

A **version** of a **digital object** is an abstraction fixing the content but not the appearance of the digital object. Two instances belong to the same version if they have the same content; they belong to different version if they have different content, but are still seen to be underlyingly the same **thing**. Versions may include revisions, transformations, translations, and so forth. **Expressions** in the **FRBR** model are a type of version.

*Work

A **work** is an abstraction in the **FRBR** model representing a distinct result of intellectual endeavour. Works may have different **expressions**, **manifestations** and instances ("items"), which are nevertheless considered to represent the same intellectual endeavour.