

 <p>pilin persistent identifier linking infrastructure</p>	<p>web: <a href="http://resolver.net.au/hdl/102.100.272/ON8J991QH">http://resolver.net.au/hdl/102.100.272/ON8J991QH</a></p> <p>email: <a href="mailto:policy@pilin.net.au">policy@pilin.net.au</a></p>
---	--

Version History			
Version	Date	Status & changes	Expression identifiers
V1.0	2007-12-20	Release	PILIN/ 8KVRC7PQH hdl:102.100.272/ 8KVRC7PQH

## PILIN Ontology Summary

To cite the *latest* version of this work use <http://resolver.net.au/hdl/102.100.272/T9G74WJQH>  
To cite *this* version of this work, use <http://resolver.net.au/hdl/102.100.272/8KVRC7PQH>

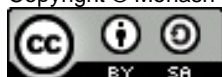
This document is a simple summary of the *PILIN Ontology for Identifiers and Identifier Services*. It is intended to introduce readers to the concepts defined by the project on identifiers and the services operating on them, and which are used throughout the documentation produced by the project. The summary is intended for introductory purposes only: readers are referred to the ontology proper for more formal and comprehensive definitions.

The PILIN **Identifier Systems ontology** is used to model identifiers and associated concepts, as well as the systems used to manage identifiers (**identifier systems**). The ontology distinguishes between entities, actions, and qualities. **Entities** are the things involved in the ontology; **actions** are the processes carried out on entities (and corresponding to *services* in implementations); and **qualities** are desirable properties of entities, which actions often make true. The ontology also defines **relations** between entities. An individual identifier system can be modelled using concepts from the ontology, with an **identifier system model**.

### 1 Entities

- Users and Managers are modelled as **parties**: people and groups that participate in activities. **Authorities** are parties **responsible** for other entities: they include system managers and people responsible for setting policies.
- **Policies** are sets of rules about entities, which can be enforced through systems.
- A **name** is an association of a **label** (a symbol) with a context that the label is in. **Contexts** define how the label is to be made sense of. There is only one instance of a given label in any one context. Contexts can impose policies on their labels.

Copyright © Monash University




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/>

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.

- Typical policies for contexts are: a **label format** policy (what labels are allowed in the context); an **access** policy (applying to parties' authorisation to carry out particular actions on the label); and an **association** policy (on how the label is used in an identifier).
- **Identifiers** are an association of a name with a single **thing**. The name is said to **identify** the thing.
- The range of things that can be identified is defined through an **information model**. The range of things that *may* be identified is defined through an association policy.
- Concrete and abstract contexts define concrete and abstract names, which in turn define concrete and abstract identifiers.
- Contexts have identifiers of their own.
- Labels may be mapped to other labels through an **encoding scheme**.
- **Actions** are triggered by parties, through a system; actions may produce results, which may be entities or changes in quality. Actions are subject to authorisation.
- Systems have a **curation boundary** defined by who is authorised to manage entities through the system. An entity crosses the curation boundary is crossed when access to it is granted to outside parties.

Example:

- **Parties:** The University of Hard Knocks; Joe Bloggs; the IT Department
- **Authorities:** The University of Hard Knocks, responsible for the local PURL identifier system.
- **Label:** 8323; cat; document1.pdf; 
- **Context:** University of Hard Knocks Purl Server Context (defined by the university's PURL system); URL (defined by DNS and the Internet as a concrete network); Employees of BHP; French Literature
- **Name:** (University of Hard Knocks Purl Server Context, report/312); (Australian Mobile Phone Numbers, 0482321234); (Employees of BHP, Joe Bloggs)
- **Policies: Label Format:** "All labels must be four digits long, and start with 8"
- **Policies: Access:** "Information on the label 8323 can only be looked up by staff members"
- **Policies: Association:** "All labels in this context are only used to identify SIM cards" (i.e. this is a context for mobile phone numbers)
- **Identifiers:** ((Australian Mobile Phone Numbers, 0482321234), "my phone"); ((University of Hard Knocks Purl Server Context, report/312), "the report on global warming from last week")
- **Identifiers: Contexts:** (<http://purl.org/uni-hardknocks/>, "University of Hard Knocks Purl Server Context")

## 2 Relations

- Two identifiers are **equivalent** if they identify the same thing at the same time.
- Two identifiers are **synonymous** if some authority claims that they are equivalent. (i.e. we trust the authority that they are equivalent, rather than confirm it for ourselves.)
- An identifier is an **alias** of a target identifier if the identifiers are synonymous, and the alias is managed to be dependent on the target.
- A **preferred** identifier is one out of a set of synonyms that an authority privileges and guarantees the persistence of.
- Two names are the **same** only if they contain the same label in the same context.
- Entities have one or more **representations** in order to communicate them to an audience. Labels are represented through encoding schemes appropriate to the medium of communication; e.g. URL-encoding is necessary for labels occurring inside URLs. Names, contexts, and identifiers are all **represented** through a single label. Representations of names can combine a context identifier with a label in the one representation. Two representations can look different, because of different encodings, but still represent the same name.
- 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.
- 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.
- Some entities are **concrete**, meaning that they are managed by a specific identifier system. Entities that are not concrete are **abstract**. In particular, an identifier management system defines a single concrete context specific to it. Abstract contexts are defined instead by their purpose and owner.
- Concrete contexts **realise** abstract contexts, if the concrete context's identifiers correspond to the abstract context's identifiers, and the two contexts' policies are consistent. Correspondingly, a concrete identifier can realise an abstract identifier. Realisation is necessary because abstract identifiers cannot appear in computer systems, by definition.
- Two concrete identifiers are **homologues** if they are equivalent and realise the same abstract identifier with the same label. Homologues are a simple way of realising a single abstract identifier in multiple systems.

### Example:

- **Equivalent Identifiers:** ((Employees of BHP Staff Numbers, 8336), "my cousin Fred"), ((Human Names, Fred Q. Bloggs), "my cousin Fred")
- **Synonymous Identifiers:** "Taiwan Province of China" and "Taiwan" according to the authority of the People's Republic of China (but not according to the Republic of China)
- **Canonical Identifier:** ISBN 0195306090 (as opposed to "Shirk, S. 2007. China: Fragile Superpower" or NLA:an40693053), according to the International ISBN Agency

- **Same Name:** (Names of soccer players, Pelé) = (Names of soccer players, Pelé) (Note that there are several soccer players named Pelé, but the name is the same regardless of what it refers to)
- **Not Same Name:** (Names of soccer players, Pelé) ≠ (Names of asteroids, Pelé); (Handle Server 102.100.272, Pelé) ≠ (PURL Server purl.nla.gov, Pelé) (even if both refer to the same thing)
- **Representation:** *identifier* ((University of Hard Knocks Purl Server Context, report/312), "the report on global warming from last week") → <http://purl.org/uni-hardknocks/report/312>; *name* (Handle Server 102.100.272, XYZZY) → <hdl:102.100.272/XYZZY>; *label* Pelé → Pelé (URL encoding scheme), .-.. .-.. .-.. (Morse Code encoding scheme), Pele (ASCII)
- **Context: Concrete:** University of Hard Knocks Purl Server Context (defined by the university's PURL system); URL (defined by DNS and the Internet as a concrete network); Australian Mobile Phone Numbers (defined by the mobile telephony system)
- **Context: Abstract:** University of Hard Knocks Library; Employees of BHP; French Literature
- **Homologues:** ((Handle Server 102.100.272, Pelé), "Pelé"), ((PURL Server purl.nla.gov, Pelé), "Pelé")—or as representations, <hdl:102.100.272/Pelé>, <http://purl.nla.gov/Pelé> . Both realise the same abstract identifier ((Names of soccer players, Pelé), "Pelé"), are equivalent, and have the same label.

### 3 Qualities

- A thing is **unique** if there exists only one of the thing within a given scope.
- An identifier is **universal** if it is the unique identifier identifying a thing within a given context. (If the context is "all known naming systems", this is not a realistic expectation, which is why various alternate strategies are in place to emulate it—particularly *preferred* identifiers. If the context is "single identifier management system", the expectation is often realised.)
- An entity or quality is **persistent** if it is managed and maintained for a defined period. (This does not have to be forever, so the emphasis is on uninterrupted maintenance rather than duration.)
- An entity or quality is **accountable** to a party if that party can access well-maintained information on its previous and current responsible authorities. Accountability is realised through **accountability data**, such as **authority metadata**.
- An entity is **trusted** by a party if that party is confident that their use of the entity meets certain expectations. Meeting expectations about system performance (e.g. uptime, load handling) makes the entity **reliable**.
- An entity is **nameable** if it may be treated as a name (i.e. it has a *representation*).
- Other qualities are defined with respect to the actions realising them. These include **registered**, **actionable**, **resolvable**, **reserved**, **published**, **citable**, **verified**, **verifiable**.

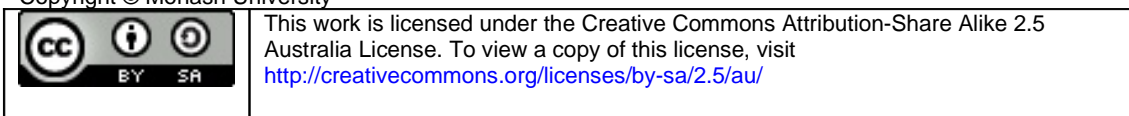
## 4 Actions

- **Curatorial** actions take place in order to manage entities—that is, to realise or maintain desirable qualities of the entities. They occur within the curation boundary of a system: only parties authorised to manage entities can undertake such actions.
- All other actions, which can be triggered outside the curation boundary, are **non-curatorial**. **Act** or **Action** is a cover term for such actions, and identifiers which can be operated on by non-curatorial actions are **actionable**.
- To **Cite** a thing is to *communicate* a representation of a thing to an audience. Identifiers can be cited, as can identifier actions (e.g. service calls). Depending on what the representation is embedded in, the thing is **citable** in different ways, e.g. **Web-Citable**, or **Print-Citable**.
- To **Create** a thing is to bring it into being.
- To **Register** a thing is to make it be maintained and managed in a system. Entities can be created without being registered, but well-behaved identifier systems require registration.
- To **Deregister** a thing is to delete it from a system; it is the opposite of Register, rather than Create.
- To **Reserve** a thing is to assign it a temporary or 'in use' status; it is used to mark an identifier object in a system as not yet fully populated.
- To **Identify** a thing is to associate it with a name, which creates an identifier.
- To **Publish** a thing is to enable access to it outside the curatorial boundary, through a defined non-curatorial action. (This is usually understood to be *Resolve*.) E.g. an identifier may be accessed through curatorial actions, such as *Update*; but this does not mean the identifier is published. An identifier may also be resolvable by members of the IT department, while it is being prepared for release; but it is only considered *published* once users outside the department are also allowed to resolve it.
- To **Act** on an entity is to perform a non-curatorial action on an entity. In particular, to **Resolve** an identifier is to get information on how to access the thing identified (**resolution data**). It does not mean actually providing such access (**Retrieve**), which is typically the responsibility of an external system. Resolve is the main non-curatorial action used with identifiers. For instance hdl:102.100.272/XYZ may be resolved to http://www.example.com/a.pdf through a Handle Resolver. But accessing the latter URL and downloading the PDF at that location is a distinct action (enabled by www.example.com rather than the 102.100.272 Handle server).
- An identifier is **Resolvable** if it can be resolved, and **Web-Resolvable** if it can be resolved to information usable directly on the web (e.g. resolvable to a URL). Registering **resolution data** is the most common way of registering **association data**: i.e. a representation on an identifier system of what thing is being identified. It is not the only way of registering association data, and some things can be identified without being resolvable: e.g. XML namespaces, vocabulary terms.
- An identifier can have multiple instances of resolution data, all of them providing access to the same thing according to the system's information model. Such identifiers can have **multiple resolution**, in which all instances

of resolution data are returned to the requester. Multiple resolution typically feeds into an *appropriate copy* selection process, which determines which resolution data is the best to use for the request.

- For instance, an identifier for a document can have multiple URLs as resolution data. This is allowed by the information model, because the thing identified is the abstract document, and not a particular instance of the document on a server. Any of the URLs counts as information on how to access the document, and all of them can be returned to the requester as resolutions.
- To **Query** an entity is to obtain selected information about the entity from a system. It is a curatorial action, and end users are typically interested in resolution instead, which is specific to resolution data.
- To **Verify** an entity is to confirm that a value for an entity, managed in a system, is what it should be. Verify applies to particular qualities, such as *accountable* and *resolvable*. Entities are **verifiable**, if the Verify action is possible for that entity and quality; they are **verified** if the Verify action has actually taken place. The usual target of verification is the resolvability of an identifier: verification confirms that the identifier resolves to something, and moreover that it resolves to the correct thing.

Copyright © Monash University



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.