GWD-C
GFSG

Freek Dijkstra, SARA
Richard Hughes-Jones, DANTE
Gregory B. Newby, Arctic Region Supercomputing Center
Joel Replogle, Open Grid Forum

August 2011

Procedure for Registration of Subnamespace Identifiers in the URN:OGF Hierarchy

Status of This Document

Community Practice (CP)

Copyright Notice

Abstract

URNs in the OGF namespace take the form
`urn:ogf:<snid>:<subnamespace-specific-string>`.

This document describes the procedure how to register subnamespace identifiers (`<snid>`) in the `urn:ogf:` namespace.

Contents

1 introduction

Uniform Resource Names (URNs) are persistent, globally unique identifiers [RFC 2141, RFC 3406].

An identifier labels a resource, which facilitates unambiguous identification of that resource. Any resource can be named, including a work, an instance of a work, an entry in an ontology or the ontology as a whole.

1.1 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC 2119].

1.2 Globally Uniqueness of URNs

[RFC 3406] stipulates:

- A URN MUST NOT be assigned to more than one resource.
- A URN MUST NOT be re-assigned to a different resource.
- A single resource MAY have more than one URN assigned to it for different purposes.

For the purpose of this document, we will call an identifier *unique* if adheres to the first two requirements above. So a unique identifier identifies only one resource, although one resource does not have to be represented by a single identifier.

Any organisation that assigns identifiers to its resources (the *assigning organisation*) must adhere to the above requirements. This ensures that the identifiers assigned by this organisation are unique. However, this only ensures that the identifier is locally unique, but does not yet guarantee that the identifier is globally unique.

To ensure global uniqueness of identifiers, each local identifier is prepended with a *prefix* that in itself is a unique identifier of the organisation that assigns the local identifiers. In the context of URNs, this prefix is often referred to as a *namespace identifier*. The group of all identifiers that start with a certain prefix is called a *namespace*. A second purpose of a prefix is to provide a cue for the type of resource being identified.

For example, a working group in the OGF may define an ontology with three terms, identified by their names *Grid*, *Cloud*, and *Cluster*. A different organisation may use the same terms, but with different meaning. To prevent ambiguity when using these terms, the working group

may register the namespace identifier `urn:ogf:example-wg`, and use the identifiers `urn:ogf:example-wg:grid`, `urn:ogf:example-wg:cloud` and `urn:ogf:example-wg:cluster`. When these identifiers are used, it is unambiguous which concept is referred to.

This document describes the procedure to register a namespace identifier within the `urn:ogf:` namespace.

1.3 Persistency of URNs

The goal of URNs is to provide persistent identification of resources. An identifier must remain valid and unmodified from its creation to well beyond the lifespan of the resource it identifies. (The identifier may still be used for archived information, like monitoring data, even after the resource itself is disbanded.) This persistency is the responsibility of the organisation maintaining the URN namespace as well as the delegate organisation(s) that assign the identifier.

The requirement that a URN MUST NOT be re-assigned only partly ensures longevity. An identifier MUST NOT change name, and (where applicable) validation and resolution procedures SHOULD still yield results, even after the following events:

- the resource ceases to exist;
- the properties of the resource change;
- a new version of the resource is created;
- the namespace organisation ceases to exist;
- the organisation that assigned the identifier ceases to exist; or
- the organisation that assigned the identifier changes its name.

For example, the Open Grid Forum (OGF) emerged from the merger of the Global Grid Forum (GGF) and Enterprise Grid Alliance (EGA) in 2006. Namespaces created before that time still have `ggf` in their prefix, and this MUST NOT change, as not to break the persistency of the identifier.

2 Selecting a Namespace

This document describes how to register a prefix in the `urn:ogf:` namespace. OGF working groups or interested third parties that wish to assign identifiers must first determine if the `urn:ogf:` namespace is the most suitable namespace.

Potential namespaces include:

- `urn:ogf:` namespace, as described in this document.
- `urn:` namespace, as maintained by the IETF and IANA [RFC 3406].
- `http://schemas.ogf.org/` namespace [GFD-C.84] (a location identifier that also serves as a resource identifier).
- A namespace in the handle system [RFC 3650].
- Non-prefixed namespaces, such as UUIDs [X.667] or hash values.

URNs in the `urn:ogf:` namespace may be suitable for identifying individual grid resources.

URNs in the `urn:` namespace may be suitable for identifying resources other then grid resources.

A URL may be suitable as identifier for ontologies and schemata, as the URL can double as the location where the normative schemata definition can be found. URLs are not suitable to identify individual resources, as the same resource may be available from multiple locations, or may not be accessible via a protocol with an associated URL scheme, or the resource may outlive the lifetime of a given scheme (e.g. a resource which is now accessible with the HTTP scheme may later be accessible with the HTTPS scheme).

Systems that like to integrate with the handle system for resource resolution of digital resources should consider using the (non-URN) identifiers defined in the handle system.

A non-prefixed system may be a suitable identifier namespace if a centrally maintained registry is considered a drawback.

DNS-based identifiers are generally NOT RECOMMENDED for persistent identifiers.

3 Canonical Syntax of URN:OGF identifiers

The canonical syntax of URNs in the `urn:ogf:` namespace is specified in sections 2.4 and 2.11 of [RFC-dijkstra-urn-ogf]. The sections of this RFC are replicated below.

```
OGF-URN = "urn:ogf:" SNID ":" SUBNAMESPACE-SPECIFIC-STRING
```

where `<SNID>` is a unique subnamespace identifier, and `<SUBNAMESPACE-SPECIFIC-STRING>` is a unique local identifier within the namespace with the prefix `"urn:ogf:" SNID ":"`.

Clearly, each `SNID ":" SUBNAMESPACE-SPECIFIC-STRING` MUST be unique.

`<SNID>` has the same syntax as a namespace identifier `<NID>` as defined in [RFC 2141]:

```
SNID = ( ALPHA / DIGIT ) *31( ALPHA / DIGIT / "-" )
```

<ALPHA> and <DIGIT> are defined in Appendix B of [RFC 5234].

Both "urn:ogf:" and the subnamespace identifiers are case insensitive.

Section 4 of this document describes the procedure how subnamespace identifiers are assigned by the OGF.

The syntax of <SUBNAMESPACE-SPECIFIC-STRING> is dependent on the <SNID>. [RFC-dijkstra-urn-ogf] does not pose any additional restrictions to the <SUBNAMESPACE-SPECIFIC-STRING> other than what is defined in the <NSS> syntax as defined by [RFC 2141]:

```
SUBNAMESPACE-SPECIFIC-STRING = 1*<URN chars>
```

<URN chars> is defined in Section 2.2 of [RFC 2141].

Documents defining a subnamespace identifier SHOULD specify further syntactic restrictions in <SUBNAMESPACE-SPECIFIC-STRING>. It is RECOMMENDED that these documents forbid the assignment of URNs containing characters in the <reserved> set ("%", "/", "?" and "#") as defined in [RFC 2141]. This is in accordance with section 2.2 of [RFC 3986].

For forward compatibility (e.g. updates in the URN syntax [RFC2141bis]), it is RECOMMENDED that software implementations that don't validate specific subnamespace-specific strings, validate the syntax according to the generic rules for validating URIs, as defined in [RFC 3986]. URIs may contain all characters defined in <URN chars>, including the characters in <reserved> (albeit they have a special meaning), as well the characters "&" and "~".

4 Procedure for Registering a Namespace Identifier

The Open Grid Forum delegates parts of its namespace urn:ogf: by assigning subnamespace identifiers (SNIDs). The formal application for a SNID is made via publication of an Grid Forum Document (GFD). The GFD SHOULD be an Informational Document, and MUST be published according to the normal procedure as described in [GFD-C.152] or its successor.

A GFD that registers a namespace MUST at least include the following sections:

- The prefix to register
- The version of the specification (usually starting with version 1)
- Syntax of URNs in the registered namespace
- Rules for lexical equivalence of two URNs in the namespace

- Procedure for assignment of identifiers (and possible further subdelegation of the registered namespace)
- Identifier immutability considerations
- Intended use and application area for URNs in the namespace

The template defined in section 6 MAY be used as (part of) an GFD that registers a namespace.

The Open Grid Forum maintains a registry of namespace identifiers in the `urn:ogf:` namespace at the URL `http://www.ogf.org/urn/`.

The technical director of the OGF MUST be notified of any (proposed) use of a namespace identifier, by contacting urn@ogf.org. The technical director of the OGF may, at his discretion "reserve" a subnamespace identifier for a limited period of time prior to approval of an Informational GFD document describing the namespace.

This document does not describe the resolution procedure of any namespace conflicts.

5 Review Criteria for SNID Proposals

The editor of the GFD series SHOULD make the following checks before assigning a subnamespace identifier (SNID):

- Is there a grid working document that includes all required sections (as listed above)?
- Does the intended use fall within scope of the OGF?
- Is the application technically sound?
  - Is the SNID well-chosen? Does it still make sense in 10 years if the working group is disbanded?
  - Is the syntax unambiguous? Is the allowed character set defined? What is the maximum length of a URN?
  - If subdelegation within a SNID is allowed: how is uniqueness and immutability ensured for third parties (who may be less versed in the process of assignment of *globally unique* and *persistent* identifiers).
  - If percentage-escaping is allowed, is a normalization function defined for the lexical equivalence?
  - How is immutability ensured? Does the URN contain attributes? Can these attributes change within the lifetime of the identifier (see also section 1.3)? If so,

what measures are in place to ensure immutability?

6 Template for Registering a Namespace Identifier

In this section, vertical solid bars at the left hand side indicate text given as illustrative examples.

Begin of Template

1. Registration

1.1. Namespace Identifier

The subnamespace identifier.

▌ Example: "`urn:ogf:example:`"

Consideration: This namespace is likely to exist for a prolonged period of time. It MUST NOT include volatile data, such as a working group name (`example-wg`), although a name of a protocol created by the working group is acceptable. `urn:ogf:` is a formal namespace, subnamespace identifiers MUST NOT start with `x-`.

1.2. Document Version

Registration version number: starting with 1, incrementing by 1 with each new version of the GFD.
Registration date: date of submission of the GFD.

Consideration: The version number only increases for approved GFD documents, not for each revision of the working draft.

1.3. Declared Registrant

The name and address of the namespace organisation and/or contact person that is responsible for making the registration.

This section may be omitted if the registrant is the same as the authors of the GFD.

2. Syntax

2.1. Syntactic Structure

This section should outline any structural features of identifiers in this namespace. It must describe the syntax of all current and future URNs within this namespace that are considered valid. This description may be used to introduce terminology used in other sections.

> Example:
>
> ```
> EXAMPLE-URN = "urn:ogf:" SNID ":" TYPE ":" YEAR ":" DOMAIN
>          ":" INDEX *1(QUERY) *1(FRAGMENT)
> SNID = "example"
> TYPE = ALPHA *15( ALPHA / DIGIT / "-" )
> YEAR = 4DIGIT ; Year in Gregorian calendar
> DOMAIN = LDH-LABEL *( "." LDH-LABEL ) ; domain name.
> LDH-LABEL = *63( ALPHA / DIGIT / "-" ) ; part of a domain name.
> INDEX = 1*DIGIT ; Identifier of a historic supercomputer
> QUERY = "?" 1*<URN chars>
> FRAGMENT = "#" 1*<URN chars>
> ```
>
> The total length of a <EXAMPLE-URN> MUST NOT exceed 255 bytes. If a recipient receives a <EXAMPLE-URN> of longer length, it MUST be discarded as invalid.
>
> Example-URNs that are assigned to a resource MUST set <YEAR> to the year of assignment. <DOMAIN> MUST be a fully qualified domain name. The domain name MAY not contain a record in the DNS systems, but the domain MUST be under control of the assigning organisation at the time of the assignment.

Consideration: The specification may use the augmented BFR format [RFC 5234].

This section may be combined with section 2.2 or 2.3.

2.2. Reserved Use

This section should outline which of valid URNs MAY be assigned, and which URNs MUST NOT (yet) be assigned, but are reserved for future use.

> Example: For current assignments, <TYPE> MUST be either "supercomputer" or "cluster":
>
> ```
> TYPE = ("cluster" / "supercomputer")
> ```
>
> The <QUERY> and <FRAGMENT> part MUST NOT be included in assigned URNs, until its

use is documented by a revision of this document.

Considerations: [RFC 2141] does not allow query or fragment identifiers, but the they may be allowed in the future. It is recommended to explicitly note if query or fragment identifiers are allowed in the namespace.

This section may be combined with section 2.1 or 2.3.

2.3. Encoding

This section should specify the encoding or decoding algorithms. This is particularly applicable in the case (a part of) a URN maps to a external naming system.

This section MUST specify if percent-escaping as defined in [RFC 2141] is allowed, and if so, what the binary code represents. This section MUST specify if the `<reserved>` characters as defined in `<URN chars>` in [RFC 2141] are allowed (it is RECOMMENDED that they are NOT allowed).

Example: `<DOMAIN>` MAY be an internationalized domain name. All labels of the domain must be written in classic LDH (letter, digit, hyphen) format. International labels must be converted to an A-label (starting with `xn-`) [RFC 5890].

Percentage-escaped strings MUST NOT be used in `<EXAMPLE-URN>`. It is RECOMMENDED that for display purposes, protocols that exchange URNs include an attribute of the URN that is the human readable name which may include Unicode characters. The definition of such attribute is out of scope of this specification.

Example where a URN contains a reference to a external naming system:

- The `<LOCODE>` part of the URN MUST map to a valid UN/LOCODE string [LOCODE], with the `" "` (space) in the UN/LOCODE mapped to a `"-"` in the `<LOCODE>` part of the URN. Since no `"-"` can occur in UN/LOCODE identifiers, the inverse mapping is also one-to-one.

Example where a URN may contain percentage-escaped characters:

- The `<NAME>` part of the URN MAY contain percent-escaped characters (`"%" <hex> <hex>`) as described in [RFC 2141]. The percentage-decoded `<NAME>` MUST be a valid UTF-8 encoded byte sequence.
- Assigning organisations SHOULD only assign `<NAME>` whose Unicode code points are NFKD-normalised according to the Unicode 5 specification)[UAX#15] and are allowed by the rules stipulated in [RFC 5892] (no upper case or control code points).
- Receiving software SHOULD accept URNs that are valid UTF-8 encoded, even if it

contains disallowed code points. Receiving software SHOULD NOT accept URNs that are not valid UTF-8 encoded.

- Receiving software SHOULD NOT display the percentage-decoded URN if it contains disallowed code points. This prevents ambiguity if these URNs are copied and pasted by a user.

This section may be combined with section 2.1 or 2.2.

2.4. Rules for Lexical Equivalence

This section should list the algorithm for determining lexical equivalence between two identifiers in the underlying namespace.

Example: URNs are lexical equivalent if and only if they are byte-equivalent after case normalisation.

Example where a URN may contain percentage-escaped characters:

Interpretation or normalisation of percentage-escaped characters must not take place.

Considerations: [RFC 2141, RFC-dijkstra-urn-ogf] specify that the `"urn:ogf:"` part and the `<SNID>` are case insensitive and that hex-encoding in a percentage-escaped character is case insensitive. The lexical equivalence algorithm defined in this section MUST comply with these requirements. Typical normalisation functions to consider are case normalisation, percentage-encoding using UTF-8, diacritical normalisation using canonical (de)composition (e.g. NFC and NFD in Unicode [UAX#15]), and ligature normalisation using compatibility (de)composition (e.g. NFKC and NFKD in Unicode [UAX#15]). If Unicode normalisation is to occur, the specification SHOULD list a specific Unicode version (e.g. Unicode 2.0 or Unicode 6.0) since this function may change between versions (e.g. there is no pre-composed character for Y-caron in Unicode version 6, but this may be created in a future version).

It is recommended to list a few examples.

Example: The following two Example-URNs are equivalent to each other:

1- `urn:ogf:example:cluster:2009:example.net:42`
2- `URN:OGF:Example:CLUSTER:2009:EXAMPLE.net:42`

3. Services

3.1. Validation mechanism

A URN namespace may provide mechanisms for "validating" a URN – i.e., determining whether a given string is currently an assigned URN or a non-assigned URN. For example, even if a telephone number-based URN namespace was created, it is not clear that all possible telephone numbers would immediately become "valid" URNs.

Where applicable, this section should describe if the operation of validation servers is an open process, or that it is subject to some authoritative delegation procedure.

Considerations: If such mechanism exists, this section may describe this mechanism, or may refer to a document that does describe it. This section may contain requirement how a recipient of a malformed URN should act: e.g. accept the URN as-is, attempt to normalise it, or reject it.

> Example: The Example-URN namespace has a Dynamic Delegation Discovery System (DDDS) [RFC 3402]. The DDDS is a lookup service where applications can verify the validity of a given Example-URN.
>
> The DDDS for the Example-URN namespace uses DNS as the lookup database. For persistency reasons, the first lookup is done within the specially registered grid.example.net domain, which services as a persistent entry point, even if the domain names that are part of Example-URNs are no longer in use.
>
> The Application Unique String is the full Example-URN, and the First Well Known Rule (as per [RFC 3402]) is:
>
> ```
> /urn:ogf:example:([a-z0-9\-]+):([0-9]+):([a-z0-9\-\.]+):([0-9]+)/
> \2.\3.\1.grid.example.net/i
> ```
>
> A lookup client SHOULD apply the above First Well Know Rule to the full Example-URN, and do a DNS lookup for a NAPTR record for the resulting DNS record. The result is a rule set, which MUST be applied to the full Example-URN as described in [RFC 3402].
>
> A URN is considered valid if a record with a NAPTR record is found that terminates the lookup sequence (thus with the S, A, U or P flag set) within at most 8 iterations.

This section is optional. It may be combined with section 3.2.

3.2. Process for identifier resolution

A URN namespace may provide mechanisms for "resolving" a URN – i.e., determining attributes of the resources that is described, such as metadata or the location(s) where the resource can be retrieved. A resolver may also be used to point to related URNs – i.e. URNs that refer to the server resource, or URNs that refer to a more general or more specific part of the resource.

Where applicable, this section should describe if the operation of resolution servers is an open process, or that it is subject to some authoritative delegation procedure.

Considerations: If such mechanism exists, this section may describe this mechanism, or may refer to a document that does describe it. This section may contain an estimate of the volatility of the resource attributes and a reasonable caching time for these attributes, or it could dictate that any resolution mechanism contains caching time-outs.

> Example: The URN-Example namespace does currently not define a Resolution Discovery System (RDS), but clients MAY translate an Example-URN to a identifier in the handle systems[RFC 3650] using a yet-to-be-defined mapping, and use the handle system to query information about the Example-URN.

This section is optional. It may be combined with section 3.1.

4. Namespace Considerations

4.1. Scope

This section should address the type of resource to be identified in the proposed namespace.

It is recommended that a namespace is limited in scope. For example, a namespace claiming to deal in "computers" must have a global scope and address all computers, which is unlikely. On the other hand, a namespace claiming to deal with only top500 supercomputers is more reasonable. It is expected that more than one namespace may serve the same "functional" purpose.

This section should also make it clear which version of the resource is identified. For example, the namespace could deal with a general resource (e.g. "the weather in Seattle"), a particular manifestation or version (e.g. "the weather in Seattle on May 1st, 2011") or a dynamic manifestation (e.g. "the current weather in Seattle").

> Example: The following resource types for bibliographic records exist:
>
> **work** e.g. Jane Austin's "Gone with the Wind", or the

> **expression** e.g. the Swedish translation of the work
>
> **manifestation** e.g. a hard cover of the expression
>
> **item** e.g. an actual book on a shelf
>
> The Example-URN namespace only deals with manifestations, but not with works, expressions or individual items.

> Example: The Example-URN namespace only deals with top500 supercomputers and equally large scale grid clusters. The namespace is intended for supercomputers and clusters for scientific use, though there are no technical or organisational limitations for other use (e.g. clusters at large corporations or military).
>
> Each Example-URN refers to the hardware of the supercomputer or cluster in its latest operational state. An assigning organisation MAY assign a new identifier if a significant hardware update is made. (In which case, any *status* attribute of the old identifier SHOULD be set to *decommissioned*.)
>
> Clients that process Example-URNs SHOULD consider attributes such as the size of the cluster or supercomputer to be dynamic, and SHOULD regularly query the assigning organisation for changes in these properties.

This section may be combined with sections 4.2 and/or 4.3.

4.2. Identifier uniqueness considerations

This section should address the requirement that URN identifiers be assigned uniquely – they are assigned to at most one resource, and are not reassigned.

> Example: An Example-URN always refers to the latest version (latest operational state) of a resource. Attributes of Example-URNs are considered dynamic. Example-URNs can not refer to earlier states of a cluster of supercomputer, unless the assigning organisation deliberately created a different URN for each version.
>
> Domain names are volatile for the duration of persistent URNs. The `<YEAR>` part in the Example-URN makes sure that the assigning organisations never re-use the same identifier, even if the domain is transferred to a different assigning organisation. This assumes that an assigning organisation SHOULD retain control of a domain name for one year after the last identifier was assigned

Considerations: non-reassignment does not prevent all situations where a single URN refers to multiple resources. Another risk is that a single URN indadvertedly refers to multiple versions of the same resource, where each version has different properties. A common solution to this potential problem is to either limiting the scope of the URN namespace (e.g. a URN

always refers to the latest version, and it is not possible to refer to an earlier version), or adding a version part to the URN syntax.

This section may be combined with sections 4.1 and/or 4.3.

4.3. Exposition of Structure

This section should include any other considerations dealing with the interpretation the of URNs in the namespace, in addition to the syntactic checks that are described before.

Considerations: If part of the structure is opaque in meaning (the meaning is not exposed), this should be noted.

> Example:
>
> - The `<INDEX>` part of an Example-URN is opaque; clients MUST NOT infer any attributes about the resource being identified from this index.
> - Clients MAY infer the assigning organisation from the combination of `<DOMAIN>` and `<YEAR>` (but SHOULD NOT infer that from the `<DOMAIN>` part alone).

This section may be combined with sections 4.1 and/or 4.2.

5. Community Considerations

5.1. Process of identifier assignment

This section should detail the mechanisms and/or authorities for assigning URNs to resources. It should make clear whether assignment is completely open, or if limited, how to become an assigner of identifiers, and/or get one assigned by existing assignment authorities.

Answers could include, but are not limited to:

- assignment is completely open, following a particular algorithm
- assignment is delegated to authorities recognised by a particular organisation
- identifiers are assigned sequentially by some automate process
- assignment is handled on a per-identifier basis by community consensus (e.g. by publication of a community reviewed Grid Forum Document.)

> Example: Assignment of Example-URN is open to all interested parties. The requirement to become an Assigning Organisation is to be the administrative owner of a fully qualified domain name (FQDN). The administrative owner picks a domain or (sub)domain thereof (e.g. the administrative owner of `example.net` may choose `example.net` or a subdomain

> such as `sc.example.net`). The assigning organisation is free to choose any `<INDEX>`, as long as it is syntactically correct, and the resulting URN has never been assigned before.
>
> If third parties should be able to verify the validity of Example-URN and retrieve basic attributes, the assigning organisation MUST register the combination of the `<DOMAIN>` and `<YEAR>` with the Grid Lookup Organisation that provides the lookup service at grid.example.net.

5.2. Identifier immutability considerations

A URN scheme MUST be designed such that even in rare events, the identifiers never need to be changed. This section should include the considerations why this given syntax of the URN is most likely to remain stable.

The desire to change an URN may occur when the URN contains attributes which have changed. The easiest way to avoid this is to avoid the inclusion of volatile or dynamic attributes in the URN syntax.

> Example: Example-URNs are assigned with the `<YEAR>` and `<DOMAIN>` as it exist *at the time of assignment*. When the year changes or if the domain changes, *the URN* MUST NOT *change*.
>
> The `<INDEX>` has a very limited set of allowed characters (only digits), to prevent assigning organisations from encoding (potentially volatile) attributes in the URN.

Considerations: While the following events rarely occur, their occurrence may still be more frequent than the anticipated lifetime of the resource identifier. Thus, they should be considered in this section.

- The prefix contains the name of assigning organisation, and the assigning organisation merges, ceases to exists or changes name.
- The identifier contains certain attributes of the resource, and these attributes change.
- The resource changes.
- Seemingly stable properties (such as a location) may change name (e.g. countries may be split, merged or change name).

The section may contain a realistic estimate of the maximum lifetime of the resource[1], a

[1] Among URN experts, it is uncommon to make estimates of the lifetime of *persistent* identifiers. Theoretically, they should last indefinitely. It is the believe of the authors that a good estimate of the maximum lifetime of a resource identifier (e.g. 50 years, 2000 years, ...) helps reviewers evaluate the SNID registration.

maximum lifetime of the resource *identifier*[2], and thus the minimum time that the resource identifier MUST NOT change. This section SHOULD explain the organisational requirements how this minimum time is ensured.

6. Examples

This section may list a few example identifiers.

> Example: The following four URNs are syntactically valid Example-URNs: These examples are informative only. They may not actually exist:
>
> 1- `urn:ogf:example:cluster:2009:example.net:42`
> 2- `urn:ogf:example:cluster:2010:example.net:00784505476220484608`
> 3- `urn:ogf:example:supercomputer:2011:colossus.bletchleypark.org.uk:1`
> 4- `urn:ogf:example:supercomputer:2011:xn-tcklt4eq6h7d1cbe.example.org:1`
>
> (Punycode `tcklt4eq6h7d1cbe` translates to スーパーコンピュータ, Japanese for *supercomputer*.)
>
> The following four URNs are syntactically **invalid** Example-URNs:
>
> 5- ~~`urn:ogf:example:supercomputer:2011:colossus.bletchleypark.org.uk:mark1`~~
> 6- ~~`urn:ogf:example:supercomputer:2011:colossus.bletchleypark.org.uk`~~
> 7- ~~`urn:ogf:example:supercomputer:1949:colossus.bletchleypark.org.uk:1`~~
> 8- ~~`urn:ogf:example:supercomputer:2000:%e3%82%b9%e3%83%bc%e3%83%91%e3%83%bc%e3%82%b3%e3%83%b3%e3%83%94%e3%83%a5%e3%83%bc%e3%82%bf.example.org:1`~~
>
> URN 5 has invalid characters in the `<INDEX>`; URN 6 has no `<INDEX>`; URN 7 has an invalid `<YEAR>` (the domain name certainly did not exist in 1949); URN 8 has a `<LDH-LABEL>` in percentage encoded UTF-8, instead of the required Punycode.

7. Relevant Ancillary Documentation

This section should list any GFDs, RFCs, standards, or other published documentation that defines or explains all or part of the namespace structure.

This section is optional and may be replaced with in-line bibliography references.

[2]the **maximum** lifetime of a resource **identifier** is usually significantly longer than the **average** lifetime of a resource identifier, and also significantly longer than the maximum lifetime of a resource itself.

8. Changes from Previous Versions / Prior Usage

This section SHOULD include changes to previous versions of the document, or list prior (undocumented) use of the namespace.

This section is optional.

End of Template

7 Security Considerations

There are no additional security considerations other than those normally associated with the use and resolution of URNs in general.

Implementors are recommended to check the OGF registry and documentation at `http://www.ogf.org/urn/` before assuming that a given identifier is valid or has a certain meaning.

8 Glossary

Assigning Organisation A namespace organisation that assigns unique identifiers to resources.

Cool URI A URL (!) that is persistent and is used as both a locator and an identifier.

Fragment Identifier The part of a URN that follows a pound sign ("#").

Globally Unique Identifier An identifier which identifies at most one resource worldwide. A global identifier typically consists of a prefix and a local identifier.

Identifier A label (string or byte sequence) that identifies a resource.

Locally Unique Identifier An identifier assigned by an assigning organisation, that identifies at most than one resource.

Namespace A set of all possible identifiers with the same namespace identifier.

Namespace Identifier A unique identifier of a namespace organisation, often used as a prefix.

Namespace Organisation An organisation maintaining a namespace. A namespace organisation may delegate part of their namespace to another namespace organisation.

Prefix A namespace identifier, which is prepended to a local identifier to form a globally unique identifier.

Question Identifier The part of a URN that follows a question mark ("?"), but excluding the Fragment Identifier.

Subnamespace Identifier A word that identifies a part of a a larger namespace. The string "urn:ogf:" followed by a subnamespace identifier forms the prefix of a namespace that is subject of the registration procedure as described in this document.

Unique Identifier An identifier that identifies at most one resource, and is never re-assigned.

URI A string that adheres to the URI syntax [RFC 3986], such as a URL or URN.

URL The location where a resource can be found.

URN A globally unique identifier that adheres to the URN syntax described in [RFC 2141].

9 Contributors

Freek Dijkstra (Editor)
SARA
Science Park 121
1098 XG Amsterdam
The Netherlands
Email: Freek.Dijkstra@sara.nl

Richard Hughes-Jones
DANTE
City House
126-130 Hills Road
Cambridge CB2 1PQ
United Kingdom
Email: Richard.Hughes-Jones@dante.net

Gregory B. Newby
Arctic Region Supercomputing Center
P.O. Box 756020
Fairbanks, Alaska 99775
USA
Email: gbnewby@alaska.edu

Joel Replogle (Corresponding Author)
Open Grid Forum Office
P.O. Box 2326

Joliet, Illinois 60434
USA
Email: replogle@ogf.org

10 Acknowledgments

This document is based on the best practice of delegation in the URN hierarchy, as described in RFC 3406.

Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

Disclaimer

This document and the information contained herein is provided on an "As Is" basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

Full Copyright Notice

References

Normative References

[GFD-C.152] Charlie Catlett, Cees de Laat, David Martin, Gregory B. Newby and Dane Skow. Open Grid Forum Document Process and Requirements. GFD-C.152 (Community Practice), 24 June 2009. URL `http://www.ogf.org/documents/GFD.152.pdf`.

[RFC 2119] Scott Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119 (Best Current Practice), March 1997. URL `http://tools.ietf.org/html/rfc2119`.

[RFC 2141] Ryan Moats. URN Syntax. RFC 2141 (Standards Track), May 1997. URL `http://tools.ietf.org/html/rfc214`.

[RFC 3406] Leslie L. Daigle, Dirk-Willem van Gulik, Renato Iannella and Patrik Fältström. Uniform Resource Names (URN) Namespace Definition Mechanisms. RFC 3406 (Best Current Practice), October 2002. URL `http://tools.ietf.org/html/rfc3406`.

[RFC 3986] Tim Berners-Lee, Roy T. Fielding, and Larry Masinter. Uniform Resource Identifier (URI): Generic Syntax RFC 3986 (Standards Track), January 2005. URL `http://tools.ietf.org/html/rfc3986`.

[RFC 5234] Dave Crocker and Paul Overell. Augmented BNF for Syntax Specifications: ABNF. RFC 5234 (Standards Track), January 2008. URL `http://tools.ietf.org/html/rfc5234`.

[RFC-dijkstra-urn-ogf] Freek Dijkstra and Richard Hughes-Jones. A URN Namespace for the Open Grid Forum (OGF). RFC-dijkstra-urn-ogf (Informational), July 2011. URL `http://tools.ietf.org/html/draft-dijkstra-urn-ogf`.

Informative References

[GFD-C.84] Michel Drescher and Ali Anjomshoaa. Standardised Namespaces for XML infosets in OGF. GFD-C.084 (Community Practice), 31 October 2006. URL `http://www.ogf.org/documents/GFD.84.pdf`.

[LOCODE] United Nations Economic Commission for Europe (UNECE). United Nations Code for Trade and Transport Locations. UN/LOCODE 2010-2, December 2010. URL `http://www.unece.org/cefact/locode/`.

[RFC2141bis] Alfred Hoenes (Editor). Uniform Resource Name (URN) Syntax. draft-ietf-urnbis-rfc2141bis-urn (Standards Track, if approved), November 2010. URL `http://tools.ietf.org/html/draft-ietf-urnbis-rfc2141bis-urn`.

[RFC 3402] Michael Mealling. Dynamic Delegation Discovery System (DDDS) - Part Two: The Algorithm. RFC 3402 (Standards Track), October 2002. URL `http://tools.ietf.org/html/rfc3402`.

[RFC 3650] Sam Sun, Larry Lannom and Brian Boesch. Handle System Overview. RFC 3650 (Informational), August 2002. URL `http://tools.ietf.org/html/rfc3305`.

[RFC 5890] John C. Klensin. Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework. RFC 5890 (Standards Track), August 2010. URL `http://tools.ietf.org/html/rfc5890`.

[RFC 5892] Patrik Fältström (editor). The Unicode Code Points and Internationalized Domain Names for Applications (IDNA). RFC 5892 (Standards Track), August 2010. URL `http://tools.ietf.org/html/rfc5892`.

[UAX#15] Mark Davis and Ken Whistler. Unicode Normalization Forms. Unicode 6.0, Unicode Standard Annex #15, September 2010. URL `http://www.unicode.org/reports/tr15/`.

[X.667] Information technology - Open Systems Interconnection. Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 object identifier components. ITU-T Recommendation X.667 / ISO/IEC 9834-8, August 2008. URL `http://www.itu.int/rec/T-REC-X.667`.