

---

Workgroup: The Interpeer Project  
Published: 14 March 2025  
Author: J. Finkhaeuser  
*Interpeer*

# PIEs - Proposals for Interpeer Enhancement

---

## Abstract

This document specifies a document identifier format and creation process that does not require centralized management, yet is sufficiently unambiguous to be interoperable across loosely organized participating entities.

It was created for use in the Interpeer Project. This document is itself published using this identification system, i.e. it adheres to [[PIE.f92f09.00](#)].

## About This Document

This document is a standard PIE and so adheres to the publishing process and naming described in [[PIE.f92f09.00](#)].

- This version is published at [PIE.f92f09.00](#)
- The latest version can be found at [PIE.f92f09.00-00](#)

## Contributing

Responsibility for this document lies with The Interpeer Project.

Source code for it can be found at <https://codeberg.org/interpeer/PIE.f92f09.00>.

Additional coordination and discussion occurs on a mailing list:

- Address: [interpeer@lists.interpeer.org](mailto:interpeer@lists.interpeer.org)
- [Archive](#)
- [Membership management](#)

---

## Table of Contents

|   |    |
|---|----|
| 1. Introduction                           | 3  |
| 2. Conventions and Definitions            | 3  |
| 2.1. Terminology                          | 4  |
| 3. Proposals for Interpeer Enhancement    | 6  |
| 3.1. Identification Scheme                | 6  |
| 3.1.1. Basic Syntax Definitions           | 6  |
| 3.1.2. Identifier Component Separators    | 7  |
| 3.1.3. PIE Prefix                         | 7  |
| 3.1.4. PIE Authority                      | 8  |
| 3.1.5. PIE Versions                       | 9  |
| 3.1.6. Full Syntax                        | 12 |
| 4. Other Identifiers                      | 12 |
| 5. Registries                             | 13 |
| 6. Persistence                            | 13 |
| 7. Referencing PIEs                       | 15 |
| 7.1. Referencing Registry PIEs            | 15 |
| 7.2. Referencing any PIEs from drafts     | 15 |
| 7.3. Referencing PIEs from published PIEs | 15 |
| 8. Standardization Process                | 15 |
| 8.1. Adoption Process                     | 16 |
| 9. References                             | 16 |
| 9.1. Normative References                 | 16 |
| 9.2. Informative References               | 17 |
| Acknowledgments                           | 18 |
| Copyright Notice                          | 18 |
| Index                                     | 18 |
| Author's Address                          | 20 |

## 1. Introduction

There is no interoperability without standards. If the Interpeer Project is to fulfil its [\[MISSION\]](#), it needs to produce public interest artefacts for its research results, which include standards and related documents.

There are two main paths open to the project: either get specifications passed through standards organizations (SDOs) that also fulfill the project's open access requirements, or to produce its own repository for such documents.

Standards organizations are focused on industry needs, whereas the project is focused on earlier technology research and development; such technology is not yet under the purview of any SDO. This requires that the project publishes its own standards for the time being. A similar approach is taken by a number of other efforts.

There is a lot to be learned from document identification systems used in either SDOs or those other projects. Nevertheless, none offer the three key criteria which the Interpeer Project is aiming for:

**De-Centralization:** A truly human-centric project should not exercise centralized control over what is or isn't relevant to it. Even a federated approach may not be appropriate here.

**Simplicity:** It should not be prohibitively difficult to generate new document identifiers.

**Clarity:** Identifiers should identify a document with as little ambiguity as is best for the type of document.

Note that the primary use case for these identifiers lies in identifying standards documents. As a result of this, language in this document will refer to standardization efforts and processes. It is eminently possible to use the same approach for other documents, however.

This document specifies how these identifiers are constructed, but also places obligations on participants in processes that use these identifiers.

## 2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [\[RFC2119\]](#) [\[RFC8174\]](#) when, and only when, they appear in all capitals, as shown here.

Where applicable, this document uses plural pronouns according to inclusive language guidelines from [\[IPARCH.NIST.IR.8366\]](#).

## 2.1. Terminology

The motivation for PIEs is to provide for better inclusivity than other standardization processes offer, which requires that some commonly used terms are better defined within this context. The below definitions aim for sufficient clarity in the full understanding that they may not cover every edge case.

**Open:** A lot of debate has been spent on defining the word "Open" as it refers to either "Open Source" or "Open Standards". The major point of contention is equitable access both to the results of the standardization process, as well as to its participation. We define "Open" in the context of this document as a) being publicly accessible free of charge, and b) adhering to a standardization process that is accessible to anyone free of charge.

**Participating Entity:** A participating entity is any individual or organization (natural or legal person) participating in an open standardization process as described here. The definition also includes groups of such entities with no specific legal status. This includes any entity that states it adheres to this process, whether they actively pursue their participation or not.

If an entity neither actively participates in the process, nor publicly commits to adhering to this standard, it is not a participating entity under this definition.

Examples of participating entities include the Interpeer Project itself, its supporting organization Interpeer gUG, contributors to Open Source projects under the stewardship of either, etc.

Entities that are not participating, but nonetheless may present practical hurdles, such as an Internet access provider, cannot be considered a participating entity.

**Free of Charge:** "Free of Charge" in this context means that the participating entities **MUST NOT** request direct or indirect fees for access to the document, use of its contents, or the participation in the standardization process.

**Indirect Fee:** An indirect fee in this context is any cost incurred when attempting to participate in the standardization process that is necessary to pay in order to effectively participate.

Examples of this include e.g. a requirement to travel to a conference where in-person voting on standards adoption may occur.

Participating entities **MUST** make reasonable effort to remove such indirect fees if they are discovered.

**Reasonable Effort:** Reasonable effort describes any effort that the entity can reasonably be requested to undertake without hindering its own purpose unduly.

This is understood to be a somewhat vague definition, and bears a case-by-case examination if disputed.

An example of "reasonable effort" may involve removing accessibility barriers, which can be as simple as providing textual information whenever non-textual information is produced that provides the same quality of information as the non-textual parts. This will permit folk relying on e.g. screen readers to participate in the process.

However, an entity cannot reasonably be expected to expend the majority of their time and financial budget on such pursuits. Judging "reasonable effort" therefore includes balancing effort against available resources.

This does not, however, absolve participating entities from their responsibility to improve their own internal processes to reduce indirect fees over time.

Additionally, the purpose of PIEs is to identify specific kinds of documents, typically referred to as "standards" or "standard documents". The following definitions may therefore be useful for disambiguation as well.

**Document:** A document refers to any kind of textual information in any format.

Documents may exist in multiple equivalent bit stream encodings; the term document then refers to the collection of all equivalent documents.

Documents are considered equivalent if they contain the same information. For further disambiguation on this topic, see [Section 6](#).

**Specification:** The term specification refers to a technical document that defines how a system is to be used. This can define protocols or APIs, but also processes. They may include a list of requirements, capacity, operating principles, and any other specifics that are necessary to either build a compatible system or interact with it (or both).

**Standard:** A standard is any kind of document that is considered sufficiently complete to serve its purpose, such as serving as a specification.

In order to qualify as a standard, a participating entity **MUST** openly publish it and define a "standard PIE" identifier for it. This identifier **MUST** be included prominently in the document.

**Draft:** Drafts refer to documents that are in earlier stages of development than standards. They **MAY** be openly published by a participating entity.

If drafts are published, they **MUST** have a "draft PIE" identifier defined for it. This identifier **MUST** be included prominently in the document.

**Registry:** A registry is a special kind of standard that is a specification of well-defined items. A registry PIE **SHOULD NOT** include this specification itself, but **SHOULD** refer to other standards defining the item.

See also [Section 5](#).

### 3. Proposals for Interpeer Enhancement

It is common, though not ubiquitous, to refer to standards published in SDOs and Open Source projects by terms that make adoption appear optional. Examples include the Internet Engineering Task Force's ([IETF]) "Request for Comments (RFC)", "Fediverse Enhancement Proposals ([FEP])" or "Python Enhancement Proposal ([PEP])".

This is sensible, as neither of these sources produce standards that have any power to be (legally) binding in themselves. They are binding only insofar that if one wishes to implement any of those standards, interoperability can only be achieved if the standard is followed.

According to this reasoning, the name "Proposal for Interpeer Enhancement" is chosen to describe standards published according to this document. This name can be abbreviated to "PIE", which is both delicious, sounds better when pronounced than "IEP" (from a possible "Interpeer Enhancement Proposal"), and is less likely to be confused with other "Enhancement Proposal" abbreviations.

This document is the first ever PIE.

#### 3.1. Identification Scheme

PIE identifiers consist of several components. These specify minimal information about the PIE such that it can be more easily referenced. The following sections outline the components of PIE identifiers.

The syntax for PIE identifiers is specified as [RFC5234]-compliant ABNF definitions.

##### 3.1.1. Basic Syntax Definitions

PIE identifier components may refer to one of the following syntax definitions. PIE identifiers typically also use only lower case ASCII characters.

```
digit    = %x 3 0 - 3 9 ; digits 0 - 9
alpha    = %x 6 1 - 7 A ; alphabetic, lower case characters 'a' to 'z'
upper_alpha = %x 4 1 - 5 A ; alphabetic, upper case characters 'A' to 'Z'
```

These are combined into the following character sets:

```
alphanumeric = digit / alpha
hexadecimal  = digit / "a" / "b" / "c" / "d" / "e" / "f"
```

Sequences of such character sets may be referred to as strings of the character set:

```
digit-string = 1digit
alpha-string  = 1alpha
alphanumeric-string = 1alphanumeric
hexadecimal-string = 1hexadecimal
```

### 3.1.2. Identifier Component Separators

Components of PIE identifiers **MUST** be separated. The choice of separator is specified in the section describing the component before the separator.

Possible separators are:

```
dash    = "-"
dot     = "."
```

### 3.1.3. PIE Prefix

All PIE identifiers start with an upper case ASCII string of at least three characters. This **MUST** be followed by a dot separator.

This prefix identifies the scope the documents refer to, such as e.g. the Interpeer Project itself.

Prefixes **SHOULD** be as short as possible without losing the ability to be used for disambiguation. It is likely that three characters are sufficient for some time here.

*Motivation:* Prefixing PIE identifiers with an easy to detect string helps disambiguation, both when reading and when parsing identifiers.

The separator in this case serves the dual purpose of visually separating the next component as well as being required by some document generation tooling.

prefix = 3\*upper\_alpha

Scopes need to be well-defined, but they are not owned. Any participating entity can publish documents in any scope. They serve an informational purpose only.

In order to qualify as well-defined, definitions of scopes **MUST** be published as PIEs under the "PIE" prefix. They **SHOULD** be included in a PIE prefix registry, and **MAY** be included in the PIE prefix registry ([[PIE.f92f09.01](#)]) maintained by the Interpeer Project.

#### 3.1.3.1. Initial PIE Prefix Registry

The prefix "PIE" is reserved for the scope of the Interpeer Project.

This scope includes:

- PIEs of any kind relating to PIEs and their usage.
- PIEs of any kind relating to R&D projects maintained by the Interpeer Project.
- PIEs of any kind relating to additions or extensions to either the PIEs or the R&D projects included in the previous point.

### 3.1.4. PIE Authority

Each participating entity is considered an authority for the PIEs under their control. This suggests that using namespaces identifying the entity are a sensible precaution to avoid conflicting identifiers. However, once namespaces are part of the schema, this raises the spectre of having to deal with conflicts in namespace assignment.

The following scheme is used to deconflict namespaces:

1. Participating entities **MUST** decide on a stable, unchanging byte sequence as their namespace input. No specific requirements are placed on this byte sequence; however, participating entities **MUST** ensure that it is likely to be globally unique. This is the *namespace input sequence*.
2. A cryptographic hash of the *namespace input sequence* **MUST** be generated using the SHA3-224 algorithm ([SHA3]).
3. The hash **MUST** be converted to a hexadecimal representation.
4. At least six (6) of the rightmost hexadecimal characters **MUST** be chosen as the authority identifier, and so the effective PIE namespace.
  - If another authority is known to exist with the same identifier, the rightmost seven (7) characters **MUST** be picked, and so forth until clashes are avoided.

The process of generating the *namespace input sequence* may be fully or It's possible to randomize the *namespace input sequence*, but this is not a requirement. Randomization merely helps in avoiding authority clashes right from the start.

Methods for arriving at a *namespace input sequence* are not defined, but **MAY** include one of the following:

- Generating a UUID ([RFC9562]).
- Using a reverse domain name registered to the entity, such as e.g. "org.interpeer".
- etc.

authority = 6\*hexadecimal

The authority **MUST** be further followed by a dot separator.

*Motivation:* The approach is inspired by generating identifiers from known metadata.

For example, the [FEP] process generates a hash from the document title. However, common complaints about that process is that it effectively fixes the title, or risks generating new identifiers. It is, however, effective at avoiding collisions.

The alternative of requiring an authority to sanction namespaces (e.g. in the form of an official registry of sorts) is in conflict with the purpose of increased inclusivity. Anyone should be able to become a PIE authority.

By requiring that the input sequence is stable and likely to be globally unique, collisions can effectively be avoided. At the same time, PIEs themselves are not identified based on their own contents in a way that becomes difficult to manage.

The use of a cryptographic hash function yields a maximum likelihood of avoiding namespace clashes, and the lack of additional requirements for the *namespace input sequence*, including the lack of requirement for its publication further ensures that collisions are highly unlikely.

#### 3.1.4.1. Initial Authority Registry

The Interpeer Project itself picks "org.interpeer" as its *namespace input sequence*. The resulting SHA3-224 hash, truncated to the last six characters is "f92f09".

```
import hashlib
algo = hashlib.sha3_224()

input_sequence = b'org.interpeer'
algo.update(input_sequence)
digest = algo.hexdigest()
print(digest[-6:])
```

#### 3.1.5. PIE Versions

Instead of treating draft PIEs and standard PIEs with highly diverging processes, all PIEs are considered PIEs whether published as standards or not. Their only distinction lies in the versioning scheme to apply to them.

PIE versions consist of two components which **MUST** be separated by a dash. The first component, the "PIE specifier" identifies the specific PIE, whether standard or draft. The second component, the "PIE revision" identifies the specific version of that PIE.

```
PIE-specifier = PIE-draft-specifier / PIE-standard-specifier
PIE-version = PIE-specifier [ dash PIE-revision ]
```

Note that the revision **MAY** be omitted, but only when referencing a document (see [Section 7](#)). When publishing a document, the full version with the revision part **MUST** be used. When the revision is omitted, also the separating dash **MUST** be omitted.

##### 3.1.5.1. PIE Revision

Even published standards are living documents. While some kind of persistence is encouraged (see [Section 6](#)), it must be possible to bring standards up-to-date.

Standards such as e.g. [\[SEMVER\]](#) exist to address versioning of software, but experience in SDOs suggests that a simpler scheme is typically sufficient.

The simplest workable scheme is to use incrementing integer values as versions, where any larger integer indicates that the document so versioned supersedes any document with a lower integer.

We define the revision as positive integer values, starting from zero.

```
PIE-revision = digit-string
```

Note that gaps in numbering **MUST** be avoided. That is, in order to create a new revision, the previous revision is simply incremented by one.

PIE revisions **MUST** be interpreted by their integer value. That is, the PIE revisions "0" and "00" are semantically equivalent.

*Motivation:* Experience with other SDOs identification schemes suggests that versioning is helpful, but should be kept to a minimum of complexity. A single incrementing integer version seems to fulfil both requirements.

### 3.1.5.2. Draft PIE Specifiers

There are no semantic requirements imposed on PIE specifiers for drafts. This section only defines syntactical requirements. These exist for ease of parsing, disambiguation from specifiers for standard PIEs, and for use in domains where not every possible character set or character is well supported.

Therefore, specifiers for drafts **MUST** only consist of alphanumeric characters and the dash symbol, and **MUST** start with an alphabetic character.

```
PIE-draft-specifier = alpha 1 * (alphanumeric | dash)
```

*Motivation:* The naming for drafts is really none of this document's concern.

The requirement to start with an alphabetic character disambiguates draft specifiers from standard specifiers.

The further limiting to alphanumeric characters and dashes permits for adopting a naming and versioning scheme comparable to that of Internet-Drafts [\[IDs\]](#).

Additionally, the character set is safe to use in URLs much like the remainder of the PIE identifier character set.

### 3.1.5.3. Standard PIE Specifiers

PIEs for standards must not only be identifiable as such, but also require stable identifiers that may be referenced at will. Standard PIE specifiers may also be referred to as PIE numbers; the meaning of both is equivalent.

PIEs adopt much the same approach as many other SDOs and projects, and simply identify standard PIEs by number. The format is the same as the revision:

```
PIE-standard-specifier = digit-string
```

Just as with revisions, standard PIE specifiers **MUST** be interpreted by their integer value, meaning "0" and "00" are semantically equivalent.

Unlike revisions, there is no specific requirement to start counting PIE standard specifiers at zero, or to avoid numbering gaps. However, participating entities need to follow some kind of process or risk confusion.

It is entirely possible for each authority to define their own process for PIEs under any prefix. This approach may not work for any prefix, however.

1. Any specification of a prefix ((prefix)) **SHOULD** specify how standard PIE specifiers under this prefix are to be produced.
2. If the specification of a prefix does not include a scheme describing how standard PIE specifiers are to be produced, participating entities **MUST** produce a standard PIE specifying how they determine standard PIE specifiers.

Either **MAY** be as simple as referring to other such processes.

Motivation: There is very little reason to not to require simple incrementing PIE numbers, but it may make identification of groups of related PIEs easier to diverge from this. Therefore, it is not made a requirement.

However, lack of centralized organization demands that some rules on a scheme to be used be published.

#### 3.1.5.3.1. Standard PIE Specifiers for the "PIE" Prefix

For the "PIE" prefix, standard PIE specifiers are produced in much the same way as PIE revisions are: starting from zero, and incrementing the previously published specifier by one.

The only exception to this rule is that under the Interpeer Project's authority "f92f09", there exist reserved number spaces. Therefore, the above is better phrased as "starting from a zero offset within their number space", where the default number space also starts from zero.

Under the combined prefix and authority "PIE.f92f09":

- PIE numbered from "00" to "99" (inclusive) are to be used for PIEs relating to PIEs.
- Any PIE number from "100" and above refers to other PIEs within the scope identified by the "PIE" prefix.

Other participating entities in the "PIE" prefix **SHOULD** use simple incrementing integers starting from zero. If they wish to assign number spaces, they **MUST** publish a standard PIE describing those spaces.

Note that number space definitions are, in a sense, registry documents in that they register which number space is intended for what purpose. As in the above example, however, a full document may be more effort than the exercise is worth.

This document therefore places no requirement on publishing number spaces in a separate registry PIE, but authorities are encouraged to consider whether the subdivisions are likely to expand or change. If they are expected to, a separate registry PIE is likely the right choice and **SHOULD** be used.

*Motivation:* Numbers identifying standards documents are simple, provide little ambiguity, and are unlikely to fall prey to spelling mistakes.

### 3.1.6. Full Syntax

The full syntax for PIE identifiers results in the following specification:

```

prefix          = 3 * upper_alpha
authority       = 6 * hexadecimal

PIE-draft-specifier = alpha 1 * (alphanumeric | dash)
PIE-standard-specifier = digit-string

PIE-revision    = digit-string

PIE-identifier  = prefix dot
                  authority dot
                  ( PIE-draft-specifier /
                    PIE-standard-specifier )
                  [ dash PIE-revision ]

```

## 4. Other Identifiers

The specification of other identifiers for PIEs is outside the scope of this document. Such identifiers might refer to Archival Resource Keys ([[ARK](#)]) or Digital Object Identifiers ([[DOI](#)]), but is not limited to those examples.

The only requirement made here is that the persistence criteria for PIEs and their respective other identifiers **MUST** be compatible (see [Section 6](#)).

In particular, this document cannot specify that other identifiers must always refer to the same PIE, because other identifiers may not have such requirements. Therefore "compatible persistence criteria" as specified above implies that those other identifiers may need to be used in a compatible way rather than being intrinsically compatible.

## 5. Registries

Several standards are fundamentally extensible, and define a scheme by which extensions may be identified. This usually involves a process of registering extension identifiers together with a brief description of the extension with an organization in charge of maintaining such registries.

PIEs are intended to be inclusive, so placing an organization in charge of sanctioning "official" extensions is counter to their intent. Therefore, any registries related to standard PIEs **MUST** themselves be PIE documents, and so undergo the same process as any other PIE.

PIEs **MAY** include a registry section, which then **MUST** be clearly marked as a registry by the use of a section header that contains the word "registry". PIEs **SHOULD**, however, refer to other PIEs as registries.

PIEs that are solely registries **SHOULD** not contain significantly more text than is necessary to explain the purpose of the registry, and descriptions for registry entries. They **MUST NOT** provide any other standardization, including terminology.

Rules for maintaining registries are lightweight.

- Registry entries **MUST** be identified in some fashion, and the same identifier **MUST NOT** be used for multiple entries.
- Registry entries **MUST** include a PIE identifier to refer to for the full specification of the entry.
- When a registry entry is no longer in use, registry PIE documents **MUST NOT** remove it, and **MUST** instead mark it as obsolete. This is to avoid the same entry name being used again, which could lead to confusion.

## 6. Persistence

As has been mentioned in [Section 3.1.5.3](#), there are good reasons to keep standards preserved for archival purposes. At the same time, they may need to be updated for practical reasons. Meanwhile, standards drafts are expected to undergo revision, but should still be referenceable.

The essay "Persistence Statements: Describing Digital Stickiness" ([\[PERSISTENCE-VOCAB\]](#)) describes these conflicts in detail, and arrives at a vocabulary by which to describe the "stickiness" or persistence of digital artefacts.

In particular, it proposes the following vocabulary for content variance:

**frozen:** The bit stream representing previously recorded content will not change.

**keeping:** Previously recorded content will not change, but character, compression, and markup encodings may change during a format migration, and high-priority security concerns will be acted upon (e.g., software virus decontamination, security patching).

**fixing:** Previously recorded content may be corrected at any time, in addition to any change under "keeping".

**rising:** Previously recorded content may be improved at any time, for example, with better metadata (datasets), new features (software), or new insights (pre- and post-prints). This encompasses any change under "fixing".

**molting:** Previously recorded content may be entirely overwritten at any time with content that preserves thematic continuity. For example, an organization's homepage may be completely reworked while continuing to be its homepage, and a weather or financial service page may reflect dramatic changes in conditions several times a day.

This terminology mostly suffices to be readily applied to PIEs. More specifically:

- Standard PIEs **MUST** be maintained at least at the "fixing" level of persistence, and **MAY** be promoted to "keeping" level.

*Motivation:* The motivation behind this is that it is not necessary to run an archive proper for publishing standards (i.e. "frozen" persistence), but the next highest level of persistence is desirable for reference.

That being said, it should be possible to correct e.g. spelling mistakes, or update URLs to references at any time if the original URL is no longer resolvable. This suggests a level between "keeping" and "fixing" is appropriate.

- Draft PIEs do not have adhere to any persistence requirements.

*Motivation:* If authors choose to version drafts according to any scheme, all but the latest draft revision **MAY** be promoted to "fixing" persistence level.

This would help preserve the history of draft evolution.

With regards to availability, the document suggests the following terms:

**finite:** Availability is expected to end on or around a given date (e.g., limited support for software versions not marked "long term stable") or trigger event (e.g., single-use link).

**indefinite:** The provider has no particular commitment to the object.

**lifetime:** The object is expected to be available as long as the provider exists.

**subinfinite:** Due to succession arrangements, the object is expected to be available beyond the provider organization's lifetime.

In light of these terms, the Interpeer Project commits to "lifetime" availability of PIEs under the "PIE" prefix, and will conduct best effort succession arrangements.

## 7. Referencing PIEs

As PIE identifiers can exist in versioned and unversioned forms, care must be taken how to reference them.

It is generally recommended to refer to PIEs by their versioned identifier. This is to avoid ambiguity in the case of revisions and extensions to a document. However, there are types of PIEs where this is not particularly useful, such as when referring to e.g. registries ([Section 5](#)).

When referencing an unversioned PIE identifier, the logical thing to assume is that its authors were referring to the latest version of that PIE at the time the reference was made.

However, as identifiers also permit cross-referencing on e.g. the World Wide Web, it is entirely possible that a locator for such an unversioned reference may forward to a newer version, published later.

Therefore, referencing unversioned PIEs effectively means referencing the PIE in the version that was current at the time the reference was made, or any later version.

### 7.1. Referencing Registry PIEs

Registry PIEs **SHOULD** be referred to by their unversioned identifier, as their content is expected to be updated independently of any material that references them.

Requirements for registries include not redacting entries, but marking them obsolete instead. This should decrease the likelihood that an unversioned reference no longer contains the referenced information.

### 7.2. Referencing any PIEs from drafts

As drafts are expected to undergo change, no particular care needs to be taken to refer to other PIEs by their versioned identifier. However, nonetheless also draft PIEs **SHOULD** refer to other pies by their versioned identifier to avoid ambiguity.

### 7.3. Referencing PIEs from published PIEs

Published PIEs **MUST** refer to other pies by their versioned identifier with the exception of:

- This document [[PIE.f92f09.00](#)].
- Any registry PIE (see [Section 7.1](#)).

## 8. Standardization Process

So far, this document has laid out how PIEs are to be identified and referenced.

The terminology section ([Section 2.1](#)) definitions that imply requirements of any PIE compatible standardization process: in summary, they **MUST NOT** deliberately exclude anyone from becoming a participating entity. Furthermore, best effort **MUST** be expended by participating entities to remove unindented hurdles if any are identified.

In pursuit of this, attention has been paid to allow participating entities to largely act independently of each other. This also means that it is not in the interest of PIEs to prescribe any specific standardization process here.

However, a few requirements exist beyond those imposed by the terms. In order to be eligible to be called PIEs, documents **MUST** do the following.

1. Reference [[PIE.f92f09.00](#)] as normative.
2. Adhere to the identification scheme and other requirements of [[PIE.f92f09.00](#)].
3. Be openly accessible and licensed as outlined in [Section 2.1](#).
4. Conform to the persistence criteria from [Section 6](#) according to its type.

Furthermore, the process the participating entity institutes to contribute to any PIEs must similarly be open as defined in [Section 2.1](#).

Examples of such processes might include voting on message boards or mailing lists. Participating entities **SHOULD** make records of such processes available in lifetime keeping form, such as in the form of e.g. mailing list archives.

They could also take the form of e.g. maintaining source code repositories for which pull/merge requests are accepted, etc.

*Motivation:* Strive for clarity in identification of PIEs and strive for inclusivity by not specifying any specific process, but outlining requirements instead.

## 8.1. Adoption Process

While any participating entity may produce documents under this scheme at will, it may be desirable to have them formally adopted by the Interpeer Project or an affiliated entity such as a working group.

This document does not currently propose any specific process for this. It is however expected that a more formal process will evolve over time, to be included or referenced in future revisions of this PIE.

## 9. References

### 9.1. Normative References

[[IPARCH.NIST.IR.8366](#)]

Miller, K., Alderman, D., Carnahan, L., Chen, L., Foti, J., Goldstein, B., Hogan, M., Marshall, J., Reczek, K. K., Rioux, N., Theofanos, M. F., Wollman, D., and National Institute of Standards and Technology (U.S.), "Guidance for NIST staff on using inclusive language in documentary standards", NIST IR (Interagency/Internal Reports) 8366, DOI 10.6028/NIST.IR.8366, 29 April 2021, <<https://specs.interpeer.org/archive/NIST.IR.8366.pdf>>. Archived copy; this document has been withdrawn by NIST due to [Executive Order \(E.O.\) 14151](#) of the President of the United States of America.

**[MISSION]** Finkhaeuser, J., "Mission Statement of the Interpeer Project", June 2022, <<https://interpeer.org/about/mission-statement-of-the-the-interpeer-project/>>.

**[PIE.f92f09.00]** Finkhaeuser, J., "PIEs - Proposals for Interpeer Enhancement", PIE PIE.f92f09.00-00, 14 March 2025, <<https://specs.interpeer.org/PIE.f92f09.00/PIE.f92f09.00-00>>.

**[RFC2119]** Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.

**[RFC8174]** Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

**[SHA3]** "SHA-3 standard :: permutation-based hash and extendable-output functions", National Institute of Standards and Technology (U.S.), DOI 10.6028/nist.fips.202, 2015, <<https://doi.org/10.6028/nist.fips.202>>.

## 9.2. Informative References

**[ARK]** "Archival Resource Key", n.d., <<https://arks.org/>>.

**[DOI]** "Digital Object Identifier", n.d., <<https://www.doi.org/>>.

**[FEP]** "Fediverse Enhancement Proposal", n.d., <<https://codeberg.org/fediverse/fep>>.

**[IDs]** "Internet-Drafts", n.d., <<https://www.ietf.org/participate/ids/>>.

**[IETF]** "Internet Engineering Task Force", n.d., <<https://ietf.org/>>.

**[PEP]** "Python Enhancement Proposal", n.d., <<https://peps.python.org/>>.

**[PERSISTENCE-VOCAB]** Kunze, J., Calvert, S., DeBarry, J., Hanlon, M., Janée, G., and S. Sweat, "Persistence Statements: Describing Digital Stickiness", Ubiquity Press, Ltd., Data Science Journal vol. 16, DOI 10.5334/dsj-2017-039, August 2017, <<https://doi.org/10.5334/dsj-2017-039>>.

**[PIE.f92f09.01]** Finkhaeuser, J., "Prefix Registry for PIEs", PIE PIE.f92f09.01-00, 14 March 2025, <<https://specs.interpeer.org/PIE.f92f09.01/PIE.f92f09.01-00>>.

- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/rfc/rfc5234>>.
- [RFC9562] Davis, K., Peabody, B., and P. Leach, "Universally Unique Identifiers (UUIDs)", RFC 9562, DOI 10.17487/RFC9562, May 2024, <<https://www.rfc-editor.org/rfc/rfc9562>>.
- [SEMVER] "Semantic Versioning 2.0.0", n.d., <<https://semver.org/>>.

## Acknowledgments

The authors wish to acknowledge the work that has gone into defining other open standards processes.

Some feedback on this document has been received from parties wishing to stay anonymous at this time. Their inputs are greatly appreciated!

## Copyright Notice

Copyright (C) the document authors.

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

## Index

[A](#) [D](#) [F](#) [I](#) [K](#) [L](#) [M](#) [O](#) [P](#) [R](#) [S](#)

### A

authority [Section 3.1.4, Paragraph 1](#)

### D

document [Section 2.1](#)

draft [Section 2.1](#)

draft specifier

draft PIE specifier; PIE draft specifier [Section 3.1.5.2, Paragraph 1](#)

### F

finite availability

finite [Section 6](#)

fixing persistence

fixing [Section 6](#)

free of charge [Section 2.1](#)

frozen persistence

frozen [Section 6](#)

## I

indefinite availability

indefinite [Section 6, Paragraph 9](#)

indirect fee [Section 2.1](#)

## K

keeping persistence

keeping [Section 6](#)

## L

lifetime availability

lifetime [Section 6, Paragraph 10](#)

## M

molting persistence

molting [Section 6](#)

## O

open [Section 2.1](#)

## P

participating entity [Section 2.1](#)

prefix [Section 3.1.3, Paragraph 2](#)

## R

reasonable effort [Section 2.1](#)

registry [Section 2.1](#)

revision

PIE revision [Section 3.1.5.1, Paragraph 4](#)

rising persistence

rising [Section 6](#)

## S

scope [Section 3.1.3, Paragraph 2](#)

specification [Section 2.1](#)

standard [Section 2.1](#)

standard specifier

PIE standard specifier; standard PIE specifier; PIE number [Section 3.1.5.3, Paragraph 1](#)

subinfinite availability

subinfinite *Section 6, Paragraph 11*

## Author's Address

**Jens Finkhäuser**

Interpeer gUG (haftungsbeschränkt)

Email: [ietf@interpeer.org](mailto:ietf@interpeer.org)

URI: <https://interpeer.org/>