Set to read-only 2025-04-07 19:10  
Changes/comments will be integrated to markdown/html docs.

# Documentation

**Initial Draft** for a comprehensive pid4cat documentation.

This document matches the file “future-docs.md” in commit [fd9d88d](https://github.com/nfdi4cat/pid4cat-model/commit/fd9d88d7e9dc03b565079261613ff0e8f6d7db66)

* Initial author: David Linke
* Created: 2025-03-14
* Updated: 2025-03-25
* Contributors: MirjamS (anonymous), Preston, Nikolaos … (add your name)

# Overview

*chapter will be converted to overview.md after review*

## What is a Persistent Identifier and what is pid4cat?

A Persistent Identifier (PID) is a long-lasting reference to a digital resource that remains valid regardless of changes in the resource’s location or ownership.

pid4cat is a persistent identifier service designed specifically for the needs of the catalysis research community. The pid4cat service builds upon existing PID infrastructure ([Handle System](https://www.handle.net/)) while providing enhanced capabilities for metadata management and namespace delegation. It enables partner organizations to manage their own sub-namespaces while maintaining a consistent approach to identifier creation and metadata management. It consists of three main components:

1. **pid4cat-model**: A data model for metadata of persistent identifiers
2. **pid4cat API**: A REST API for creating, updating, and retrieving PIDs and their associated metadata
3. **pid4cat hosting**: Operation of a production handle system with an officially registered permanent handle-prefix 21.11165 and of an analogous test system with the demo prefix 21.T11978.

## Design goals

For the PID system:

* Provide PIDs for NFDI4Cat and its tools & services
* Low cost per PID
* Managed and operated by one main authority (HLRS)
* Sub ID-spaces for sub name assigning authorities (SNAA)
* Allow each SNAA to create IDs according to self-defined patterns (with some constraints on allowed characters)
* Custom REST-API access with authorization for PID-management by SNAAs
* Use of a single common handle prefix
* Open tools & open data (CC-0 for PID metadata, MIT or Apache-2-license for code)

For the PID metadata model:

* Predominantly metadata about the PID (status, changes, contact)
* Low post-creation update work for metadata
* Minimum information about the resource
  + What is identified? (selection among few types)
  + How to access the resource? (for example which other media type than html can be requested)
* Relations to other PIDs (reuse relations from DataCite [metadata schema](https://datacite-metadata-schema.readthedocs.io/))
* linked-data conform to facilitate PID-graph integration/creation

For the pid4cat pattern:

* Recognizable as pid4cat
* Short meaningless identifier for the namespaces of SNAAs.
* Best-practice guidelines for identifier patterns to help the SNAAs.
* Within the prefix space, only the combination of identifier for the namespace and SNAA-created ID must be unique to allow offline (uncoordinated & de-centralized) minting of identifiers.

## Handle System Basics

Every handle consists of two parts: its naming authority, otherwise known as its prefix, and a unique local name under the naming authority, otherwise known as its suffix:

<Handle> ::= <Handle Naming Authority> "/" <Handle Local Name>

The naming authority and local name are separated by the ASCII character “/”.

21.zzzzz/4cat/638s-k9dx  
\\_\_\_\_\_\_/\\_\_\_\_\_\_\_\_\_\_\_\_\_/  
 | |  
 prefix suffix

The collection of local names under a naming authority defines the local handle namespace for that naming authority. Any local name must be unique under its local namespace. The uniqueness of a naming authority and a local name under that authority ensures that any handle is globally unique within the context of the Handle System.

The naming authority for pid4cat handles is HLRS.

Ref.: https://datatracker.ietf.org/doc/html/rfc3650

## Identifier pattern

The handle-suffix part has to fulfill different roles in pid4cat handles. First, it should allow the minting of PIDs by different institutions. This requires that a part of the PID suffix is used as an identifier for an organization using a sub-PID-namespace. Second, the suffix must contain an identifier part that makes it unique in each sub-PID-namespace. Third, the suffix should make pid4cat handles easily distinguishable from other handles (for humans) [ref?].

The following scheme was selected for pid4cat handles:

* <handle-prefix> / <branding-suffix> / <ns-suffix> / <id-suffix>

21.zzzzz/4cat/xxx/723-hk-09  
 \\_\_\_/\\_\_/\\_\_\_\_\_\_\_\_/  
 | | |  
 branding | id-suffix  
 suffix ns-suffix

As a branding label we use 4cat as already done for the dataverse data repository. Next comes an identifier ns-suffix for the namespace of sub-authorities associated with the PID-subspace followed by the ID-suffix part. Regarding uniqueness, it is important to note that only the combination of ns-suffix and id-suffix must be unique. Due to this, all SNAAs are independent in the IDs they generate. The same id-suffix may be present in different sub namespaces. The different parts are separated by the ASCII character “/”.

Rules for the <ns-suffix>:

* To identify the sub-authority, opaque (meaningless) IDs should be used. It is suggested to keep the sub-authority identifier short (3 chars) and use only the crockford-32 alphabet. This gives 32\*\*3 = 32768 combinations.
* Sub-namespaces have to be requested at HLRS.
* The ns-suffixes are generated randomly and should have no meaning (for example they should not match abbreviations of institutions). [ref.]

Rules for the <id-suffix>:

* The local-IDs are managed by the responsible SNAA. It may contain letters [A-Za-z], numbers [0-9] and symbols [./] as part of the identifier.
* Dashes in this part are ignored.
* If an optional checksum is included, it must be calculated including the <ns-suffix> and ignoring dashes.

Suggested checksum calculations:

* **The ISO 7064 Mod 97, 10 algorithm.** It evaluates the whole identifier as an integer which is valid if the number modulo 97 is 1. It has two check digits and is easy to compute. It is for example used by [ROR](https://ror.org/) and [DataCite](https://datacite.org/).
* **The ISO 7064 Mod 37, 36 algorithm.** It uses one alphanumeric check digit and the identifier itself may also be alphanumeric (digit or char). Hence, it is well suited to detect errors in alphanumeric identification numbers. This algorithm is for example used by the [Global Release Identifier](https://en.wikipedia.org/wiki/Global_Release_Identifier).

Implementation of these algorithms are available in many languages (for example in [python-stdnum](https://arthurdejong.org/python-stdnum/doc/1.20/stdnum.iso7064)).

Open question:

* Should the ID-suffix be stored as case sensitive string? While this would not be required for numeric, hex or base32 encoded IDs, case sensitivity would be essential to support also base64 encoded ID-suffixes.
  + Proposal (David): design & program for case-sensitive IDs. My reasoning is that we should not artificially add a significant constraint that will be impossible to remove later. Case-insensitivity may be specified by individual SNAA for their sub-namespaces but should not be enforced for all created identifiers.

## Use case: PIDs for samples, devices etc.

pid4cat identifiers can be used for various resources in the catalysis research domain:

| Category | Definition | Description of use |
| --- | --- | --- |
| COLLECTION | [voc4cat:0005012](https://w3id.org/nfdi4cat/voc4cat_0005012) | A collection is a group of resources and/or other collections |
| SAMPLE | [voc4cat:0005013](https://w3id.org/nfdi4cat/voc4cat_0005013) | A representative part of a material of interest on which observations are made. |
| MATERIAL | [voc4cat:0005014](https://w3id.org/nfdi4cat/voc4cat_0005014) | A material used in the research process (except samples). |
| DEVICE | [voc4cat:0005015](https://w3id.org/nfdi4cat/voc4cat_0005015) | A physical device used in a research or manufacturing process. |
| DATA\_OBJECT | [voc4cat:0005016](https://w3id.org/nfdi4cat/voc4cat_0005016) | A collection of data available for access or download. A data object might be a data file, a data set, a data collection. |
| DATA\_SERVICE | [voc4cat:0005017](https://w3id.org/nfdi4cat/voc4cat_0005017) | An organized system of operations that provide data processing functions or access to datasets. |

The categories are linked to SKOS concepts defined in the [Voc4Cat](https://nfdi4cat.github.io/voc4cat/) vocabulary. Such mapping to machine readable terminologies facilitate the use of pid4cat PIDs in linked-data applications and are key to fully realize FAIR data.

## Use case: repo4cat

[Repo4cat](https://repository.nfdi4cat.org/) is the data sharing portal created and operated by NFDI4Cat. It is built upon [Dataverse](https://dataverse.org/). Repo4cat makes use of pid4cat handles to persistently identify the stored resources.

Example handle PID for data in the production instance of [repo4cat](https://repository.nfdi4cat.org/):

* [https://hdl.handle.net/21.11165/4cat/638s-k9dx](https://hdl.handle.net/21.11165/4cat/638s-k9dx?noredirect)
* The shoulder “4cat” is also used as “PID-namespace” for repo4cat PIDs.
* The <ns-suffix> is not present in repo4cat PIDs.

Currently the PIDs minted by repo4cat lack the rich metadata that are attached to other pid4cat PIDs due to limitations of Dataverse.

## PID resolution methods

pid4cat identifiers can be resolved using multiple methods:

1. **Direct Handle System resolution**:

https://hdl.handle.net/<prefix>/4cat/<ns-suffix>/<id-suffix>

1. **pid4cat resolver**:

<https://pid.nfdi4cat.org/><prefix>/4cat/<ns-suffix>/<id-suffix>

[https://handle.nfdi4cat.org/](https://pid.nfdi4cat.org/)<prefix>/4cat/<ns-suffix>/<id-suffix>

1. **DOI proxy**:

The DOI resolver also works with other handles than the DOI 10.\*-handles.

https://doi.org/<prefix>/4cat/<ns-suffix>/<id-suffix>

## Logos & display guidelines

The pid4cat identifier should be displayed consistently across all platforms to ensure recognition and proper use.

### Display Format

When displaying pid4cat identifiers in text:

* Use the full handle identifier including prefix, branding suffix, namespace suffix, and local identifier part
* Do not abbreviate any part of the identifier
* Maintain the case sensitivity of the identifier

Example: 21.11165/4cat/ABC/SAMPLE-2023-001

When creating hyperlinks use the handle.net resolver URL:

* https://hdl.handle.net/21.11165/4cat/ABC/SAMPLE-23-001

For using pid4cat identifiers in linked-data applications we plan to register pid4cat as the *preferable prefix* for https://pid.nfdi4cat.org/21.11165/4cat/ in [Bioregistry](https://bioregistry.io/). This allows to use the CURIE (compact URI) notation:

* pid4cat:ABC/SAMPLE-23-001

### Logo Usage

When referencing pid4cat in applications, websites, or publications, the following logo guidelines apply:

* The official pid4cat logo should be used when possible
* Maintain the aspect ratio of the logo
* Do not modify or alter the logo
* Ensure sufficient clear space around the logo
* Use the full-color version on light backgrounds and the inverted version on dark backgrounds

The official pid4cat logos can be downloaded from [URL to be provided].

## Best Practices

### Local Identifier Design

SNAAs are encouraged to follow these best practices when designing their local identifier scheme:

1. **Consistency**: Adopt a consistent pattern across all identifiers within your namespace
2. **Documentation**: Document your identifier pattern for future reference
3. **Avoid semantics**: Minimize semantic content that might become outdated.
   * Semantic identifiers contain meaningful information (e.g., JoeMiller-Sample-LiMg-001)
   * Opaque identifiers have no inherent meaning (e.g., x7z-9q2)
4. **Validation**: Implement validation to ensure identifiers conform to your pattern
5. **Checksums**: Consider incorporating checksums for error detection
6. **Future-proofing**: Design for scalability and longevity

If the resources that you want to assign identifiers to, have already identifiers in a local system (for example an electronic lab notebook), it is strongly suggested to reuse these locally unique identifiers for the identifier part in the pid4cat identifier. Often these are just integer numbers or type-number combinations.

If you design a new scheme, consider how the IDs are used. For example, whenever reading and typing of the ID by humans is important, consider using character sets (alphabets) designed for this purpose and/or using a checksum.

* **Base32-encoded sequential number with checksum**:
* to be added, ...

For pure machine use and if global uniqueness is important consider using

* **UUID4-based identifiers** (unordered):
* 40944082-40ed-4f66-9e8f-66fbcd3908a9
* **UUID7-based identifiers** (time-ordered):
* 0195c559-4b8a-7201-a7ab-f1a5d06687e0

See also [pid4cat suffix design](#bookmark=id.dl2tkqu921o7) section.

### Landing Pages

Landing pages should:

1. **Be persistent**: Remain accessible for as long as the identifier exists
2. **Provide metadata**: Display key metadata about the resource
3. **Offer access information**: Provide direct access to the resource or clear instructions for obtaining access
4. **Include citation information**: Show how the resource should be cited
5. **Show provenance**: Include information about the origin and history of the resource
6. **Be human-readable**: Present information in a user-friendly format
7. **Be machine-readable**: Include structured metadata (e.g., JSON-LD, RDFa)

### Metadata Quality

To maintain high-quality metadata:

1. **Completeness**: Provide as many relevant metadata elements as possible
2. **Accuracy**: Ensure metadata correctly describes the resource
3. **Timeliness**: Update metadata when the resource changes
4. **Relationships**: Document relationships to other resources

### Integrating with Local Systems

Recommendations for integrating pid4cat with local systems:

1. **Automated workflows**: Implement workflows that automatically create identifiers at appropriate points in research data lifecycles
2. **Metadata harvesting**: Set up processes to harvest metadata from local systems for pid4cat
3. **User interfaces**: Provide user-friendly interfaces for creating and managing identifiers
4. **Authentication integration**: Integrate local authentication systems with pid4cat API access
5. **Monitoring**: Implement monitoring to ensure identifiers remain resolvable

### Security and Access Control

The SNAAs ensure the security and access control of the data and must secure it with sufficient technical and organisational measures.

#### API Key Management

1. Keep your API key secure
2. Rotate keys periodically (recommended every 90 days)
3. Use separate keys for different environments (development, production)
4. Revoke compromised keys immediately

#### Access Policies

1. Only authorized personnel should have access to create and modify PIDs
2. Implement internal approval workflows for PID creation
3. Maintain an audit log of all PID operations
4. Review access permissions quarterly

## Roadmap

The development and expansion of pid4cat is planned in several phases:

### Phase 1: Infrastructure Establishment (Completed)

* Handle prefix registration
* Basic API implementation
* Core metadata model development
* Initial documentation

### Phase 2: Service Expansion (Current)

* Enhance API capabilities
* Improve metadata model
* Integration with related services
* Expansion of user and developer documentation
* Onboarding of initial SNAAs

### Phase 3: Community Growth (Upcoming)

* Outreach to potential SNAAs
* Development of additional tools and services
* Integration with the broader PID ecosystem
* Community building activities

### Phase 4: Sustainability (Future)

* Establish a sustainable funding model
* Plan long-term preservation
* Evaluation and improvement of governance model

## History

The pid4cat service was developed as part of the NFDI4Cat consortium, a project funded by the German Research Foundation (DFG) to establish a research data infrastructure for catalysis research (see about.md TODO make this a link).

Key milestones in the development of pid4cat:

* **2023**: Initial conceptualization of pid4cat
* **2024**: Launch of the production handle server
* **2025**: Launch of pid4cat service; Onboarding of initial SNAAs

The development of pid4cat has been guided by the needs of the catalysis research community and best practices in PID management.

# Getting Started

## Registration

To use pid4cat services, organizations must apply to become a Sub-Name Assigning Authority (SNAA). The application process involves:

1. Submitting an application form to the pid4cat core organization
2. Agreeing to the pid4cat policies (detailed in section Policies [TODO add internal link])
3. Demonstrating the capability to maintain the identified resources
4. Designating a technical and administrative contact

Upon approval, the organization will receive:

* A unique namespace suffix (<ns-suffix>)
* API credentials for accessing the pid4cat API
* Documentation and support for the integration into SNAA-operated services.

## Creating identifiers

Identifiers can be created through the pid4cat API. Partner organizations (SNAAs) have two options:

1. **Generated identifiers**: Identifiers generated by the pid4cat API
2. **Provided identifiers**: Custom identifiers created according to local naming conventions

To create Custom identifier:

POST /api/v2/… TODO add endpoint

PUT <https://api.nfdi4cat.org/><version>?customid=xxx-xxx (future)

With a payload containing:

* The local identifier (optional for opaque identifiers)
* The URL of the resource
* Required metadata according to the pid4cat-model

For detailed API documentation, see the [REST APIs](#bookmark=id.2fz758bocsya) section.

## Updating identifiers

Identifiers and their associated metadata can be updated through the pid4cat API:

PUT /api/v2/… TODO add endpoint

Updates may include:

* Changes of the target URL
* Additions or modifications to metadata
* Changes in access control

Note that the identifier itself cannot be changed once created. Only its associated metadata and target URL can be modified.

## Resolving identifiers

pid4cat identifiers can be resolved using:

1. **Direct resolution**: https://hdl.handle.net/<prefix>/4cat/<ns-suffix>/<local-id>
2. **pid4cat resolver**: https://pid.nfdi4cat.org/<prefix>/4cat/<ns-suffix>/<local-id>

Resolution will typically redirect to the landing page for the identified resource, which should be maintained by the responsible SNAA.

# Metadata

*chapter will be converted to metadata.md after review*

For pid4cat handles metadata are stored directly in the handle record in contrast to the DOI-system where metadata are managed by the DOI-providers like DataCite in separate data systems.

The metadata for pid4cat mainly include data about the PID and its relation to other PIDs. The relation information should relate the new PID to relevant existing PIDs. pid4cat allows, but does not mandate, that PIDs are updated with new relations that come into existence only later, after the PID assignment to a resource.

The metadata schema for pid4cat is defined as a Linked Data Modeling Language (LinkML) schema. From this LinkML schema various representations can be created.

The schema is documented under [overview](about:blank).

The Python package derived from the schema is described in the section [Tools and Services](#bookmark=id.4vyym0show5p) below.

## Core Metadata Elements

pid4cat requires a set of core metadata elements for each identifier:

| Element | Description | Required |
| --- | --- | --- |
| EMAIL | Contact email of PID curator in responsible SNAA | Yes |
| STATUS | PID status information | Yes |
| SCHEMA\_VER | The version of the pid4cat metadata model | Yes |
| METADATA\_LICENSE | SPDX license specifier “CC0-1.0” | Yes |
| RESOURCE | Information about the resource and its representation (json) | Yes |
| RELATED | Information about related identifiers and the type of relation (json) | No |
| CHANGES | Structured change log for the PID record (json) | Yes |

These metadata elements are stored in the handle records of pid4cat PIDs.

## Metadata Schema

The pid4cat metadata schema is defined using LinkML, which allows for:

* Clear definition of metadata elements
* Validation of metadata values
* Generation of various schema representations (JSON Schema, SHACL, etc.)
* Integration with semantic web technologies

The complete schema is available in the [pid4cat-model repository](https://github.com/nfdi4cat/pid4cat-model).

## Metadata Validation

All metadata submitted to pid4cat is validated against the pid4cat-model schema. This ensures:

* Required fields are present
* Field values conform to expected formats
* Structural integrity of the metadata

Validation failures result in rejection of the current operation (create or update), with detailed error messages to assist in their correction.

## Metadata Access

pid4cat metadata can be accessed using the following methods:

* Direct API access
* Handle record retrieval
* Metadata dumps (for bulk analysis)

All metadata is provided under a CC0 1.0 Universal Public Domain Dedication license, enabling unrestricted reuse.

# REST APIs

*chapter will be converted to rest\_apis.md after review*

HLRS API (create, read, update)

## pid4cat API

The pid4cat API provides a RESTful interface for creating, reading, and updating pid4cat identifiers and their associated metadata.

### Base URL

main : https://api.nfdi4cat.org/<version>

test : https://api.nfdi4cat.org/testpid/<version>

### Authentication

The API uses API key authentication. Each SNAA is issued a unique API key that must be included in all requests:

Authorization: Basic <api-key>

Authorization: Bearer <api-key> (future)

### Endpoints

The endpoints are described in the auto-generated [API documentation](https://api.nfdi4cat.org/testpid/v2/index.html).

A demo showing how to access all available API endpoints from Python is provided in the [Jupyter](https://jupyter.org/) notebook [examples/api-v2-demo.ipynb](https://github.com/nfdi4cat/pid4cat-model/blob/main/examples/api-v2-demo.ipynb).

## Handle.net API

In addition to the pid4cat API, identifiers and their metadata can also be retrieved through the API of the [HDL.NET Proxy Server System](https://www.handle.net/proxy_servlet.html).

### Base URL

https://hdl.handle.net/api/handles/

### Request full handle metadata

GET https://hdl.handle.net/api/handles/<prefix>/4cat/<ns-suffix>/<id-suffix>

Response:

{  
 "responseCode": 1,  
 "handle": "<prefix>/4cat/<ns-suffix>/<id-suffix>",  
 "values": [  
 {  
 "index": 1,  
 "type": "URL",  
 "data": {  
 "format": "string",  
 "value": "https://example.org/resource"  
 },  
 "ttl": 86400,  
 "timestamp": "2023-01-01T00:00:00Z"  
 },  
 **//** **Additional** **handle** **values** **containing** **metadata**  
 ]  
}

Tip: To request a tabular HTML view of all data in the pid4cat record you can attach the query parameter noredirect to the PID URL, for example

* <https://hdl.handle.net/21.T11978/test/C9K6-C6K9?noredirect>

### Limitations

The Handle.net API provides read-only access to pid4cat identifiers. To create or update identifiers, the pid4cat API must be used.

# Tools and Services

*chapter will be converted to tools.md after review*

Python package based on LinkML pid4cat-model.

## pid4cat Python package

pid4cat is a Python package for:

* Interacting with the pid4cat API to create, read and update pid4cat identifiers and their metadata
* Validating pid4cat metadata against the pid4cat-model schema

## pid4cat API

A REST API server that provides the endpoints to create, read and update pid4cat identifiers. It handles authentication of the SNAAs and SNAA-related validation.

The source code of the software is not yet available as open source but there are plans to release it in the future.

For usage instructions, see [pid4cat API](#bookmark=id.b633srfn5pjz).

### Features

* Sub-namespace management
* Create, read, update operation on single pid4cat handles
* Bulk operations (future)
* Reporting (future)
* Monitoring (future)

### Access

The pid4cat-admin tool is available only to the core organization and authorized administrators.

Integration with NFDI Base4IAM is ongoing in order to use NFDI identification and authorisation services for access control to the pid4cat API gateway.

## pid4cat in Django

A Django application [django-pid](https://gitlab.com/opensourcelab/scientificdata/pid/django-pid) to integrate pid4cat in django-based ELNs/LIMS solutions is in development (pre-alpha state).

## Schema representations

Through LinkML generators we provide other schema representations that are automatically derived from the pid4cat-model. These are available in the repository under [/project](https://github.com/nfdi4cat/pid4cat-model/tree/main/project), and are, for example JSON-schema, SHACL shapes, SQL.

## Integration Libraries

Through LinkML generators, pid4cat provides integration code for some programming languages other than Python, see [generated model artifacts](https://github.com/nfdi4cat/pid4cat-model/tree/main/project).

# Policies

## Governance

pid4cat is governed by a two-tiered structure:

1. **pid4cat Core Organization**: Responsible for the overall governance, technical operation, and sustainability of the pid4cat service
2. **Sub-Name Assigning Authorities (SNAAs)**: Partner organizations responsible for managing identifiers within their assigned sub-namespaces

## Governance Principles

1. **Transparency**: All policies, procedures, and decisions are documented and publicly available
2. **Inclusivity**: The governance process includes input from all stakeholders
3. **Sustainability**: Governance structures are designed to ensure the long-term viability of the service
4. **Accountability**: Clear responsibilities and reporting mechanisms are established
5. **Community-driven**: Governance is responsive to community needs and priorities

## Registration as SNAA

To become a Sub-Name Assigning Authority (SNAA), an organization must:

1. Submit an application to the Core Organization (HLRS)
2. Demonstrate the capability to maintain identified resources
3. Agree to the pid4cat policies
4. Designate technical and administrative contacts

## Identifier Persistence

pid4cat is committed to the persistence of identifiers. Once created, an identifier should remain resolvable indefinitely. This implies:

1. **No deletion**: Identifiers must not be deleted, even if the identified resource is no longer available
2. **Tombstones**: If a resource becomes unavailable, a tombstone page should be provided with information about the resource and its status
3. **Responsibility**: SNAAs are responsible for maintaining the resolution of identifiers they create

In exceptional circumstances, identifiers may be deleted or transferred, but only with the approval of the pid4cat Core Organization.

## Metadata

Metadata associated with identifiers can be updated to reflect changes in the resource or to correct errors.

1. **Metadata Quality**: SNAAs must provide accurate and complete metadata
2. **Metadata Updates**: Metadata may be updated to correct errors or reflect changes in the resource
3. **Core Organization Rights**: The Core Organization reserves the right to update metadata to ensure policy compliance
4. **Provenance**: All metadata updates are recorded with provenance information
5. **Metadata License**: All metadata is made available under CC0 1.0 Universal Public Domain Dedication

## Access Policy

1. **Landing Pages**: All identifiers must resolve to a landing page or point directly to the resource.
2. **Resource Access**: If a resource has access restrictions, a landing page must provide clear information about how to access the resource.
3. **Metadata Access**: Metadata associated with identifiers should be openly accessible.

## Resource Access

pid4cat does not require identified resources to be openly accessible. However:

1. **Landing page requirement**: All identifiers must resolve to a landing page with basic metadata about the resource or point directly to the resource.
2. **Access information**: If access to the resource is restricted, the landing page must provide information about the restrictions and if/how access can be obtained.
3. **Metadata accessibility**: Metadata associated with identifiers should be openly accessible, even if the resource itself is not

## SNAA Responsibilities

Each SNAA is responsible for:

1. **Identifier Management**: Creating and maintaining identifiers within their sub-namespace
2. **Resource Maintenance**: Ensuring identified resources remain accessible or providing appropriate tombstone pages
3. **Metadata Accuracy**: Providing and maintaining accurate metadata
4. **Policy Compliance**: Adhering to all pid4cat policies
5. **Contact Information**: Maintaining up-to-date technical and administrative contacts
6. **Sustainability**: Contributing to the long-term sustainability of pid4cat

## Licensing

pid4cat adopts the following licensing policies:

1. **Service licensing**: The pid4cat service is operated under [specific license]
2. **Metadata licensing**:
   * Metadata submitted to pid4cat is made available under the CC0 1.0 Universal Public Domain Dedication
   * This allows for unrestricted reuse of metadata, promoting interoperability and discovery
3. **Resource licensing**:
   * pid4cat does not mandate specific licenses for identified resources
   * SNAAs should include clear license information in the metadata
   * Recommended licenses include Creative Commons licenses, Open Data Commons licenses, and other open licenses

## Sustainability

To ensure the long-term sustainability of the pid4cat service:

1. **Succession planning**: The pid4cat Core Organization has established succession plans to ensure continuity of service
2. **SNAA succession**: SNAAs must designate a successor organization or return control of their sub-namespace to the Core Organization if they can no longer fulfill their responsibilities
3. **Financial model**: [Details about the financial model supporting pid4cat]
4. **Data preservation**: All identifier metadata is regularly backed up and preserved in multiple locations

# Open Data

*chapter will be converted to opendata.md after review*

We plan to make a PID metadata dump available as FAIR data in the future.

## PID Metadata Dump

As part of our commitment to FAIR data principles, pid4cat will provide regular dumps of PID metadata. These dumps will:

* Be available in multiple formats (JSON, RDF, CSV)
* Include all non-sensitive metadata fields
* Be licensed under CC0 1.0 Universal Public Domain Dedication
* Include comprehensive documentation
* Be assigned their own pid4cat identifiers

### Availability

The metadata dumps will be available through:

1. Direct download from the pid4cat website
2. API access for programmatic retrieval
3. Integration with metadata aggregators

### Usage

The metadata dumps can be used for:

* Research on PID usage patterns
* Integration with discovery services
* Network analysis of PID relationships
* Development of value-added services

## PID Graph Integration

pid4cat is designed to integrate with the broader PID Graph ecosystem. The PID Graph connects different types of persistent identifiers, creating a rich network of scientific and scholarly entities.

pid4cat contributes to this ecosystem by:

1. Including relationships to other PIDs in its metadata
2. Using standard relationship types
3. Exposing metadata in machine-readable formats
4. Supporting linked data principles

## Analytics and Reporting

The open data approach enables various analytics and reporting functions:

1. **Usage statistics**: Tracking resolution requests and metadata access
2. **Growth metrics**: Monitoring the growth of pid4cat identifiers over time
3. **Connectivity analysis**: Analyzing relationships between pid4cat identifiers and other PIDs
4. **Community impact**: Assessing the impact of pid4cat on the research community

These analytics will be made available through regular reports and an interactive dashboard.