

Trainingskurs Arbeiten mit PIDs, Teil2

RDA-DE Trainings Workshop 2017

Ulrich Schwardmann

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen
(GWDG)

Am Fassberg, 37077 Göttingen
ulrich.schwardmann [at] gwdg.de

08 June 2017, Dresden

Content

- 1 Einführung
- 2 PID Information Types
- 3 Warum Datentypen registrieren ?
- 4 Wie wird eine Data Type Registry genutzt?
- 5 ePIC DTR
- 6 Beispiele für PID-InfoType Instanzen
- 7 Zugriff auf PIDpe-HandleView-InfoTypes
- 8 Prüfung einer PID-InfoType-Instanz
- 9 Was fehlt noch ?

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-InfoType-
Instanz

Einführung PID Information Types

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Info3/45-
Instanz

Motivation

Diese Session zeigt einfache und fortgeschrittene Anwendungsbeispiele für PID Information Types (PITs) und der ePIC Data Type Registry (ePIC DTR)

Das Ziel ist zu verstehen, wie **PITs verwendet** werden.

Diese Session verwendet die folgenden Dienste:

- **ePIC PID Data Type Registry (PID-Resolution):**
 - die ePIC DTR zur Auflösung von PITs
 - wird gleichzeitig auch zur Verwaltung der PITs genutzt

URL:

<http://dtr-test.pidconsortium.eu/>

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Inf4/45-
Instanz

PID Information Types

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

**PID
Information
Types**

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-InfoType
Instanz

PID Information Types, Datenaustausch und Reusability

Arbeiten mit
PIDs

Ulrich
Schwardmann

- PID InfoTypes sind zusätzliche *Metadaten*
- **ähnlich wie bei Mime-Types** werden sie schon vor dem Datenzugriff ausgewertet
 - aber Mime-Types reichen für Daten nicht aus
 - Zum Beispiel braucht man bei Tabellen Metadaten über jede Spalte
- sie werden daher direkt verknüpft mit der PID
 - für Handle-PIDs werden sie direkt in die Handle-DB geschrieben
- ähnlich wie Mime-Types ist eine Standardisierung notwendig

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Info
67/45
Instanz

PID Information Types, Datenaustausch und Reusability

Arbeiten mit
PIDs

Ulrich
Schwardmann

- die Überprüfung der Einhaltung des Standards ist aber wegen der Variabilität komplexer
- Beispiele:
 - Datenintegrität: Checksum, Größe, ...
 - Datenverfügbarkeit: Embargos, Löschung, ...
 - Ort-Zeit-Angaben: GeoLocs, Entstehung, Veränderung, ...
 - Provenance: Creator, DC5, Versionen, Kopien, ...

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Inf7/45-
Instanz

PID Information Types Geschichte

- Handle Types bei CNRI - 0.Type/SomeType
 - Gute Idee, geringe Anwendbarkeit
- Sloan Grant: 2012 -2014
 - Generic Registry System nutzt Type Registry als Usecase
- NSF Grant: 2013 -2014
 - Support für Type Registry

RDA hat sich bereits in den ersten Plenaries mit Types und PID Information Types beschäftigt:

- WG *Data Type Registry (DTR)*, Ziele:
 - Spezifikation einer DTR
 - Prototypische Implementierung einer DTR
- WG *PID Information Types*, Ziele:
 - allgemeine Beschreibung derartiger Metadaten
 - Spezifikation einer PIT-API
- Adaptionen:
 - verschiedene DTRs existieren, u.a. die ePIC DTR
 - PIT API ist prototypisch implementiert

ePIC das Handle System stellt mit v8 eine andere API bereit

Warum Datentypen registrieren ?

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

**Warum
Datentypen
registrieren ?**

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Instanz
9 / 45

Wie also sollte ein PID Info Type aussehen?

Namen sind keine hinreichende Charakteristik für InfoTypes

- Identifikation sollte unabhängig vom Namen sein
 - verschiedene Schemata für den gleichen Namen sollten möglich sein
 - umgekehrt sollten für das gleiche Schema auch mehrere Namen erlaubt sein

InfoTypes brauchen Schemata zur syntaktischen Überprüfung

- InfoTypes sollen für automatische Prozesse wie MimeTypes funktionieren:
 - schon vor dem Zugriff auf die Daten sollen wichtige Informationen über die Daten vorliegen
 - dies erfordert eine hohe Zuverlässigkeit der InfoType-Inhalte

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Instanz
10/45

Wie also sollte ein PID Info Type aussehen?

- verwende deshalb eine **Data Type Registry**,
 - Eine Low-Level Infrastruktur mit breiter Anwendbarkeit um Type Records zur Verfügung zu stellen,
 - die Definitionen der InfoType Charakteristiken enthält,
 - InfoTypes durch PIDs **identifiziert**,
 - InfoTypes durch Name und Beschreibung **interpretierbar** macht,
 - und idealerweise ein **Schema** für die InfoTypes angibt.
 - Die ein gemeinsames Datenmodell verlangt und
 - Ausdrücke validiert zur Interoperabilität zwischen Registries.
 - mit API für Maschinennutzung und UI für Menschen

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-InfoType
Instanz

Wie wird eine Data Type Registry genutzt?

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

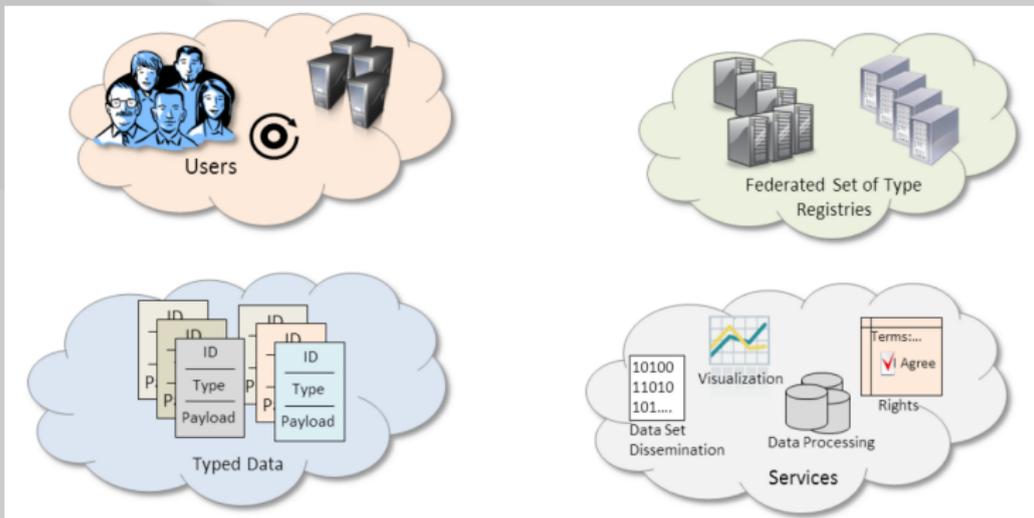
ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

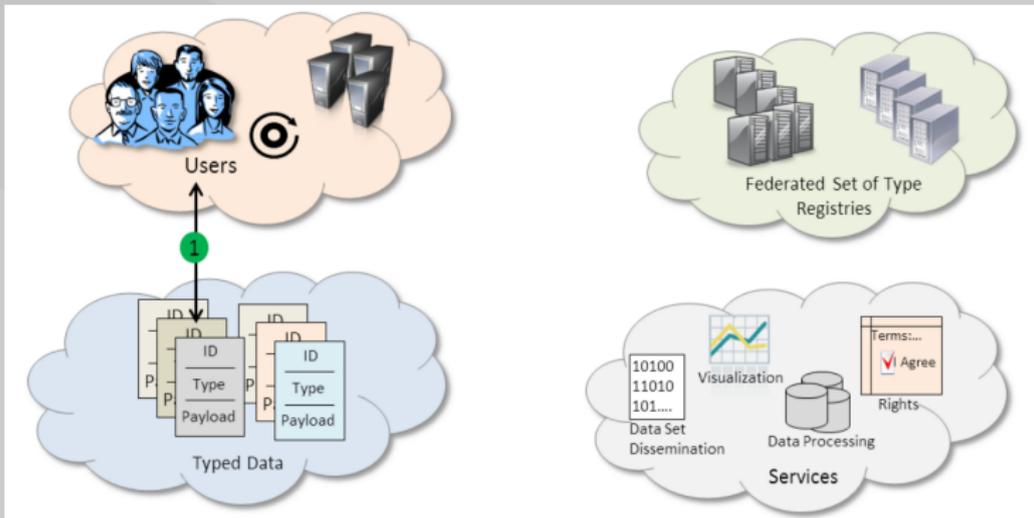
Prüfung einer
PID-Instanz

DTR Process Use Case



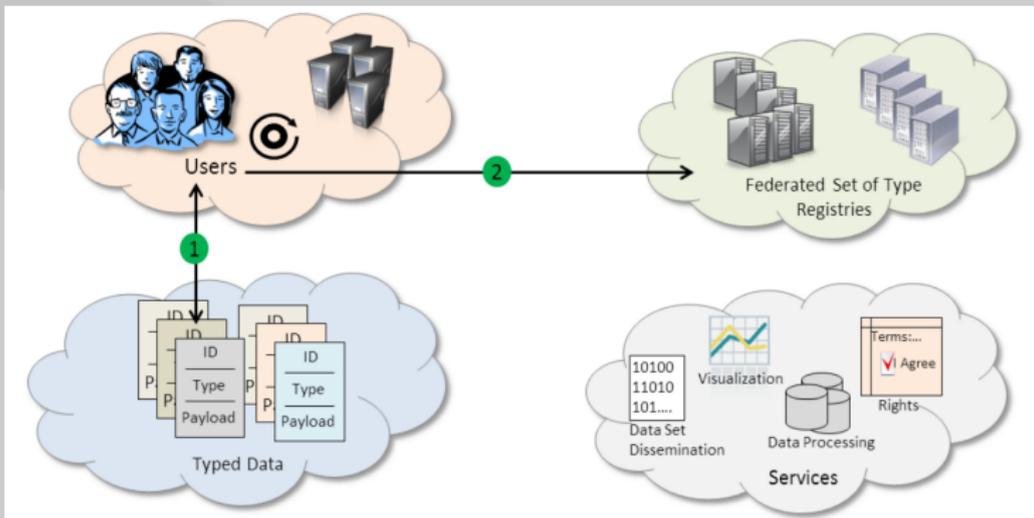
- 1 Client (process or people) encounters unknown data type.
- 2 Resolved to Type Registry.
- 3 Response includes type definitions, relationships, properties, and possibly service pointers. Response can be used locally for processing, or, optionally
- 4 typed data or reference to typed data can be sent to

DTR Process Use Case



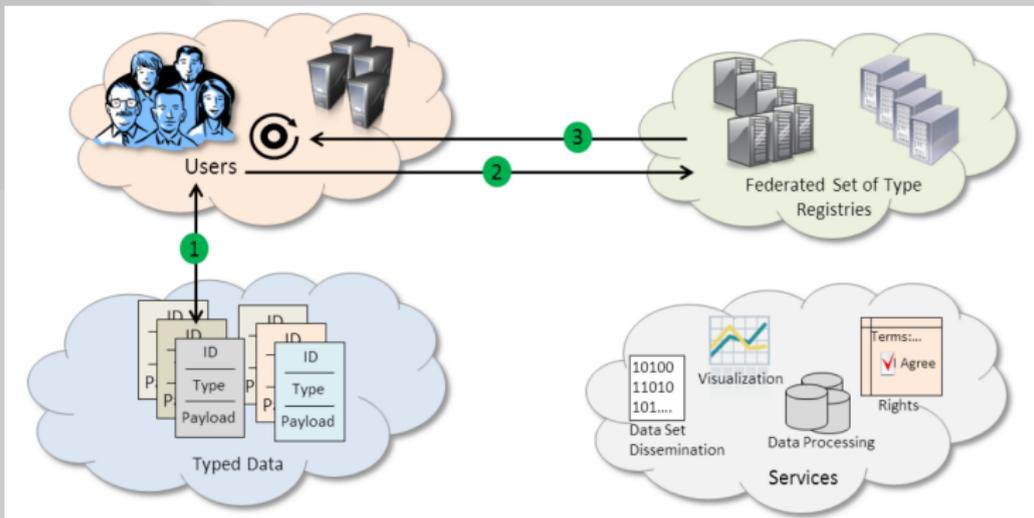
- 1 Client (process or people) encounters unknown data type.
- 2 Resolved to Type Registry.
- 3 Response includes type definitions, relationships, properties, and possibly service pointers. Response can be used locally for processing, or, optionally
- 4 typed data or reference to typed data can be sent to

DTR Process Use Case



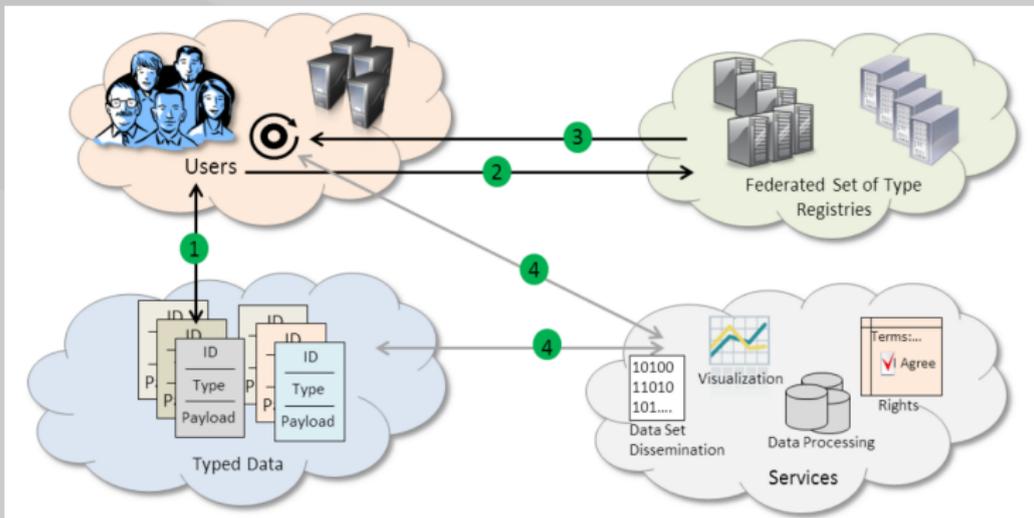
- 1 Client (process or people) encounters unknown data type.
- 2 Resolved to Type Registry.
- 3 Response includes type definitions, relationships, properties, and possibly service pointers. Response can be used locally for processing, or, optionally
- 4 typed data or reference to typed data can be sent to

DTR Process Use Case



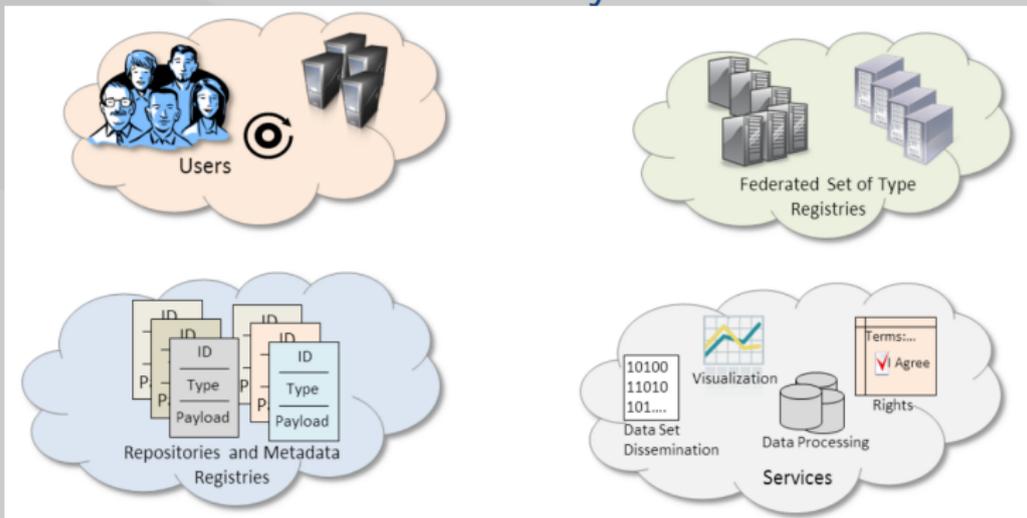
- 1 Client (process or people) encounters unknown data type.
- 2 Resolved to Type Registry.
- 3 Response includes type definitions, relationships, properties, and possibly service pointers. Response can be used locally for processing, or, optionally
- 4 typed data or reference to typed data can be sent to

DTR Process Use Case



- 1 Client (process or people) encounters unknown data type.
- 2 Resolved to Type Registry.
- 3 Response includes type definitions, relationships, properties, and possibly service pointers. Response can be used locally for processing, or, optionally
- 4 typed data or reference to typed data can be sent to ePICS service provider.

DTR Discovery Use Case



Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

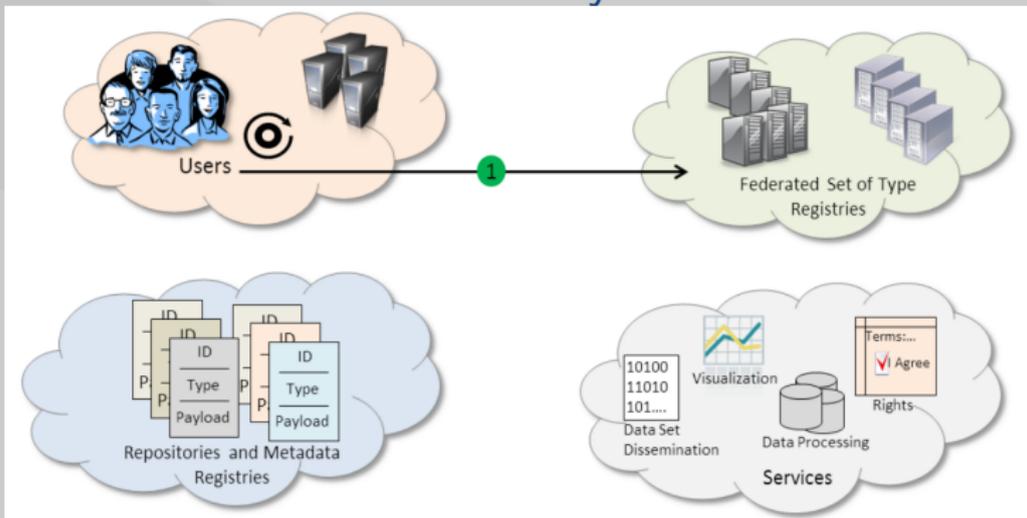
Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-InfoType
Instanz

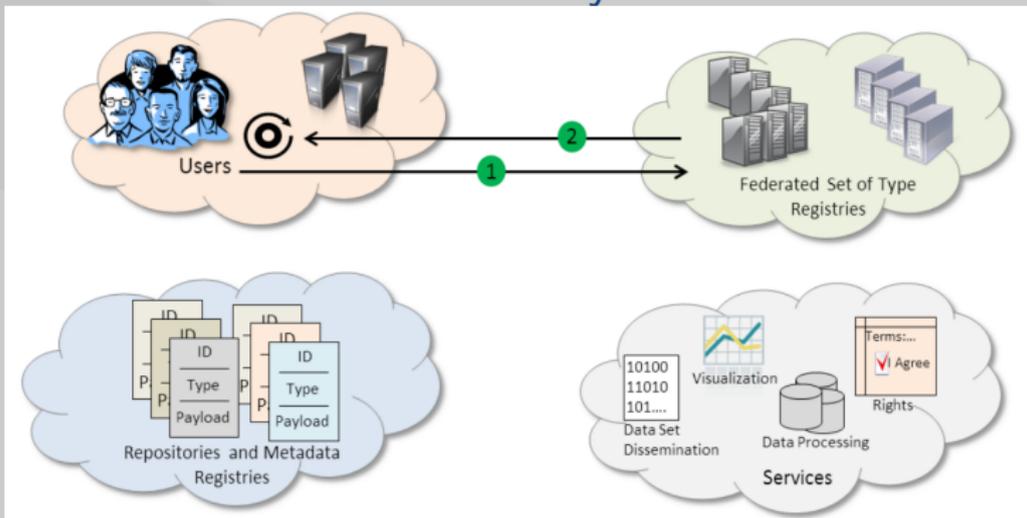
- 1 Clients (process or people) look for types that match their criteria for data processing
- 2 Type Registry returns matching types.
- 3 Clients look up for data sets matching those types.
- 4 Appropriate typed data is returned.
- 5 Client can process this data. (Example: format conversion)

DTR Discovery Use Case



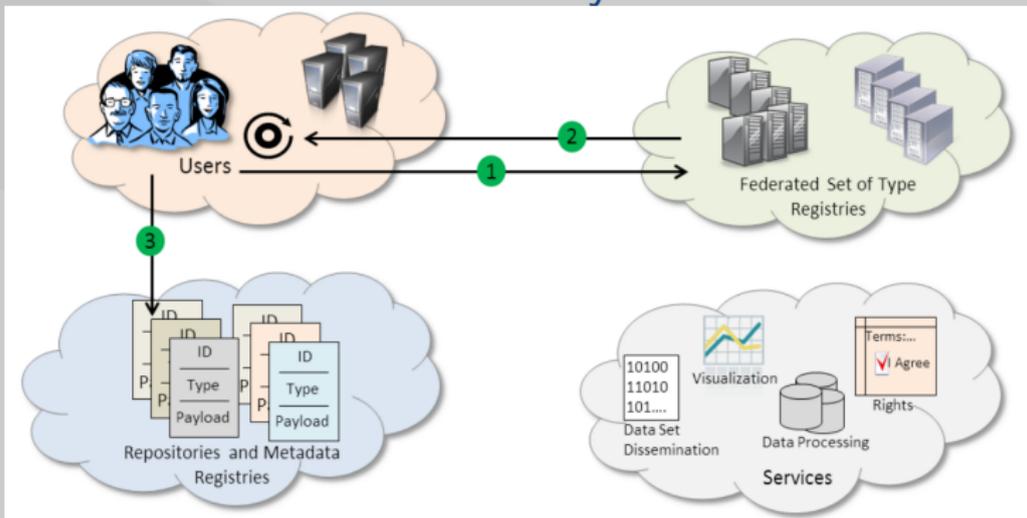
- 1 Clients (process or people) look for types that match their criteria for data processing
- 2 Type Registry returns matching types.
- 3 Clients look up for data sets matching those types.
- 4 Appropriate typed data is returned.
- 5 Client can process this data. (Example: format conversion)

DTR Discovery Use Case



- 1 Clients (process or people) look for types that match their criteria for data processing
- 2 Type Registry returns matching types.
- 3 Clients look up for data sets matching those types.
- 4 Appropriate typed data is returned.
- 5 Client can process this data. (Example: format conversion)

DTR Discovery Use Case



Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

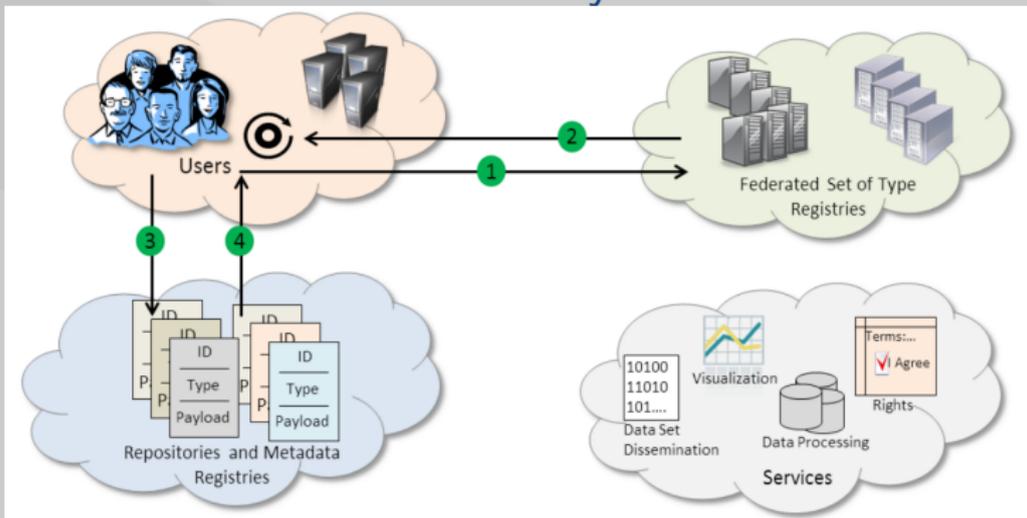
Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PID-
HandleView-
InfoTypes

Prüfung einer
PID-Instanz

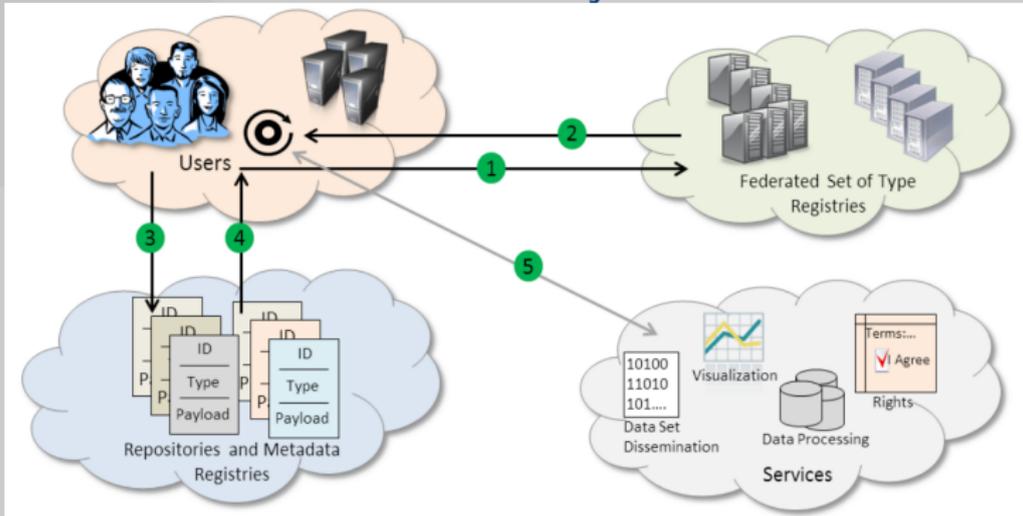
- 1 Clients (process or people) look for types that match their criteria for data processing
- 2 Type Registry returns matching types.
- 3 Clients look up for data sets matching those types.
- 4 Appropriate typed data is returned.
- 5 Clients can process this data. (Example: format conversion)

DTR Discovery Use Case



- 1 Clients (process or people) look for types that match their criteria for data processing
- 2 Type Registry returns matching types.
- 3 Clients look up for data sets matching those types.
- 4 Appropriate typed data is returned.

DTR Discovery Use Case



- 1 Clients (process or people) look for types that match their criteria for data processing
- 2 Type Registry returns matching types.
- 3 Clients look up for data sets matching those types.
- 4 Appropriate typed data is returned.
- 5 Client can process this data. (Example: format conversion)

Die ePIC DTR

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Instanz
24 / 45

Die ePIC DTR Startseite

Beispiele unter: <http://dtr.pidconsortium.eu/>

- PID InfoType Zustände sind:
 - *in preparation (21.T11148), candidate, approved, deprecated (21.11104)*

Startbildschirm der PID InfoType Registry for *preparation*:

<http://dtr-test.pidconsortium.eu/>

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information

Document Repository - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

Document Repository x +

vm11.pid.gwdg.de:8081/#urls/intro.html Suchen

Document Repository Introduction All Types Sign in

Search Search

A few things to note before you get started...

Administration

You can administer the repository by accessing <http://localhost:8080/admin.html>. By default, you can administer the repository using username 'admin' and password 'changeit'. To change these defaults edit the file `repository/data/password.dct` and restart the repository.

The administration page allows you to configure metadata schemas, declare authentication and access control policies, enable federation, etc.

You can add users and groups using the main repository page (link below). Authenticate as admin and create user and group records as needed.

By default the repository is configured to store documents and associate metadata with those documents.

You have been temporarily allotted a Handle prefix 21.T11148 that will be used by the repository to allot persistent identifiers to registered digital objects.

Repository Access

You and other users can access the repository at <http://localhost:8080/> or using DOIIP at port 8900. Depending on the access controls set in the administration page, you and other users may need to authenticate prior to searching, retrieving, or registering any data or metadata.

ePIC DTR Abfragen: PID-InfoTypes

http://dtr-test.pidconsortium.eu/
#objects/?query=type:"PID-InfoType"

Arbeiten mit
PIDs

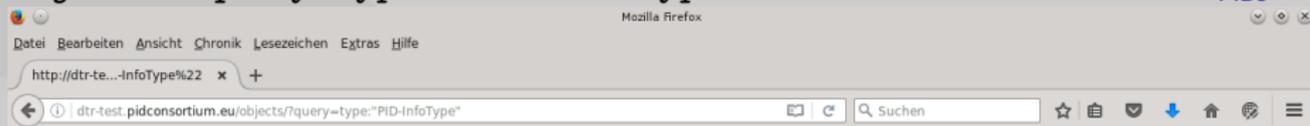
The screenshot shows a Mozilla Firefox browser window displaying the Document Repository website. The address bar contains the URL: `dtr-test.pidconsortium.eu/#objects/?query=type:"PID-InfoType"`. The page header includes the site name "Document Repository", navigation links like "Introduction", "All", and "Types", and a user profile for "UlrichSchwardmann" with a "Sign out" button. A search bar is present with the text "Search" and a "Create" dropdown. Below the search bar, a pagination control shows "1 to 10 of 72 results" with a "JSON View" button. The main content area lists several PID-InfoTypes with their descriptions:

- previous-version** [Type: PID-InfoType]
pointer to previous version of digital object by identifier
- next-version** [Type: PID-InfoType]
pointer to next version of digital object by identifier
- ORCID-nameIdentifier-DataCite** [Type: PID-InfoType]
Uniquely identifies an individual or legal entity, according to ORCID schema.
- Other-nameIdentifier-DataCite** [Type: PID-InfoType]
Uniquely identifies an individual or legal entity, according to an arbitrary schema.
- ISNI-nameIdentifier-DataCite** [Type: PID-InfoType]
Uniquely identifies an individual or legal entity, according to ISNI schema.
- time-period-open** [Type: PID-InfoType]
describes a time period between given begin and end date-times. Allow a missing begin or end entry, because it does not allow abbreviated form. A missing begin or end entry in the array means every time before end-time or after begin-time. An empty time period (no date-times specified) represents every time.
- false** [Type: PID-InfoType]
Type Name: caseSensitiveHeader-CSVDDF
indicates that case in the header is meaningful. For example, columns CAT and Cat should not be equated. Default = false
- \r\n** [Type: PID-InfoType]
Type Name: lineTerminator-CSVDDF
specifies the character sequence which should terminate rows. Default = '\r\n'

ePIC DTR Abfragen: PID-InfoTypes as JSON

http://dtr-test.pidconsortium.eu/objects/?query=type:"PID-InfoType"

Arbeiten mit
PIDs



```
{
  "pageNum": 0,
  "pageSize": 0,
  "size": 72,
  "results": [
    {
      "id": "21.T11148/47a0dabaef514c3553c9",
      "type": "PID-InfoType",
      "content": {
        "representationsAndSemantics": [
          {
            "subSchemaRelation": "denyAdditionalProperties",
            "expression": "",
            "allowAbbreviatedForm": "Yes",
            "value": ""
          }
        ],
        "name": "previous-version",
        "provenance": {
          "creationDate": "2016-07-28T14:23:41.674Z",
          "lastModificationDate": "2016-11-17T17:22:04.128Z",
          "contributors": [
            {
              "identifiedUsing": "Text",
              "name": "Ulrich Schwarldmann",
              "details": "GWDG"
            }
          ]
        },
        "validationSchema": "{\n\"definitions\": {\n\"21.T11148_38330bcc6a40ca85e5b4\": {\n\"anyOf\": [\n{\n\"anyOf\": [\n{\n\"required\": [\n\"PMCID-Identif...
      }
    ]
  }
}
```


ePIC DTR Abfragen: PID-InfoType latitude API

http://dtr-test.pidconsortium.eu/
#objects/21.T11148/5fcccdf1d079c4a85c9

Arbeiten mit
PIDs

The screenshot shows a Mozilla Firefox browser window displaying the Document Repository website. The address bar shows the URL: `dtr-test.pidconsortium.eu/#objects/21.T11148/5fcccdf1d079c4a85c9`. The page title is "Document Repository". The navigation bar includes "Introduction", "All", "Types", and a "Sign out" button for "UlrichSchwardmann".

The main content area displays the details for the PID object "latitude". The object type is "PID-InfoType". There are buttons for "Edit Object", "Digital Object View", "JSON View", "Versions View", "Share Object", and "Clone Object".

The "identifier" field contains the value: `21.T11148/5fcccdf1d079c4a85c9`.

The "Type Name" field contains the value: `latitude`.

The "Description" field contains the text: `latitude with degrees in decimal fraction or sexagesimal minutes and seconds (see ISO 19111) and leading [+|-] or ending [N|S] from 0 to 90 degrees.`

Under the heading "Applicable Standards or Recommendations", there is a table with the following data:

| Nature of Applicability | Standard Name * | Issued By * | Details |
|-------------------------|---|-------------|---------|
| depends | 19111 <small>Type ID or standard</small> | ISO | |

JSON-Struktur von PID-InfoTypes

- `name` : der Name des Typs
- `description` : die Beschreibung des Typs
- `identifizier` : der PID des Typs
- `provenance` : Herkunft, Erzeuger, Zeiten etc.
- `properties` : die verwendeten Untertypen
 - `representationsAndSemantics` : die Eigenschaften verwendeten Untertypen
- `representationsAndSemantics` : die Eigenschaften des Typs
- `validationSchema` : das Schema des Typs
- `standards` : für den Typ relevante Standards

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Instanz
30/45

Was sind PID BasicInfoTypes?

Struktur

- `name` : der Name des Typs
- `description` : die Beschreibung des Typs
- `identifizier` : PID des Typs
- `provenance` : Herkunft, Erzeuger, Zeiten etc.
- `validationSchema` : das Schema des Typs
- `standards` : für den Typ relevante Standards
- `dataType` : Datentyp: integer, string
- `regexp` : regular expression (bei String)
- `flavour` : der regexp, i.e. ecma-262-RegExp
- `enum` : Aufzählung erlaubter Werte
- `default` : Angabe eines Defaults, wenn kein Wert angegeben
- `restrict` : Beschränkungen wie `minimum`, `maxLength`, ...

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Instanz
32/45

Beispiele für PIDs mit PID-InfoType (Instanzen)

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Instanz
33 / 45

Wie sieht eine PID mit InfoType aus?

■ erzeugen:

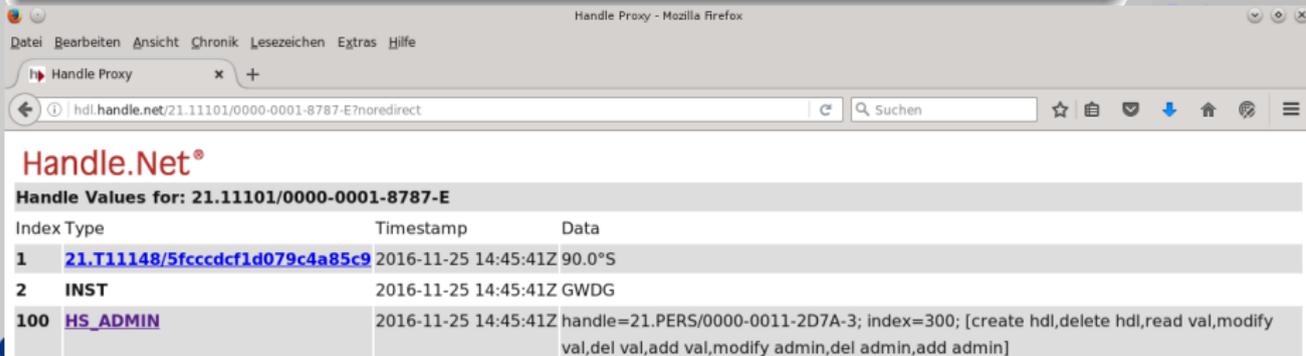
Code

```
curl -u 'user:pwd' -H 'Accept:application/json'  
-H 'Content-Type:application/json' -X POST --data  
' [{"type": "21.T11148/5fccccdf1d079c4a85c9",  
"parsed_data": "90.0°S"} ]' http://pid.gwdg.de/handles/21.11101/
```

■ anzeigen über Handle Proxy:

Code

```
http://hdl.handle.net/21.11101/0000-0001-8787-E?noredirect
```



| Index | Type | Timestamp | Data |
|-------|--|----------------------|--|
| 1 | 21.T11148/5fccccdf1d079c4a85c9 | 2016-11-25 14:45:41Z | 90.0°S |
| 2 | INST | 2016-11-25 14:45:41Z | GWDG |
| 100 | HS_ADMIN | 2016-11-25 14:45:41Z | handle=21.PERS/0000-0011-2D7A-3; index=300; [create hdl,delete hdl,read val,modify val,del val,add val,modify admin,del admin,add admin] |

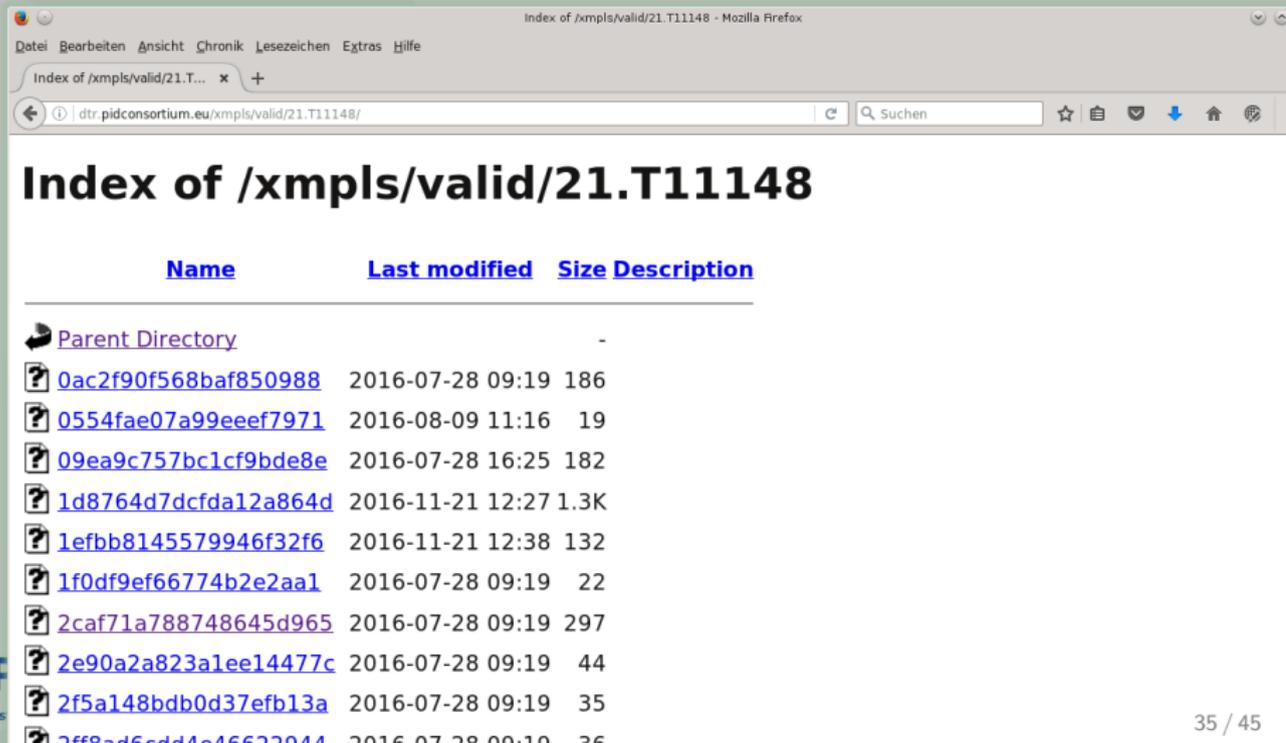
Was sind valide und nicht valide Instanzen?

Code

```
http://dtr.pidconsortium.eu/  
xmpls/valid/21.T11148/
```

Arbeiten mit
PIDs

Ulrich
Schwardmann



The screenshot shows a web browser window with the address bar containing the URL `dtr.pidconsortium.eu/xmpls/valid/21.T11148/`. The page title is "Index of /xmpls/valid/21.T11148". Below the title, there is a table with columns for "Name", "Last modified", "Size", and "Description". The table lists several XML instance files, each with a unique identifier, a timestamp, and a size. The first entry is "Parent Directory" with a size of "-".

| Name | Last modified | Size | Description |
|--------------------------------------|------------------|------|-------------|
| Parent Directory | - | - | - |
| 0ac2f90f568baf850988 | 2016-07-28 09:19 | 186 | |
| 0554fae07a99eeef7971 | 2016-08-09 11:16 | 19 | |
| 09ea9c757bc1cf9bde8e | 2016-07-28 16:25 | 182 | |
| 1d8764d7dcfda12a864d | 2016-11-21 12:27 | 1.3K | |
| 1efbb8145579946f32f6 | 2016-11-21 12:38 | 132 | |
| 1f0df9ef66774b2e2aa1 | 2016-07-28 09:19 | 22 | |
| 2caf71a788748645d965 | 2016-07-28 09:19 | 297 | |
| 2e90a2a823a1ee14477c | 2016-07-28 09:19 | 44 | |
| 2f5a148bdb0d37efb13a | 2016-07-28 09:19 | 35 | |
| 2ff8ad6edd4a46622044 | 2016-07-28 09:19 | 36 | |

Was sind valide und nicht valide Instanzen?

Code

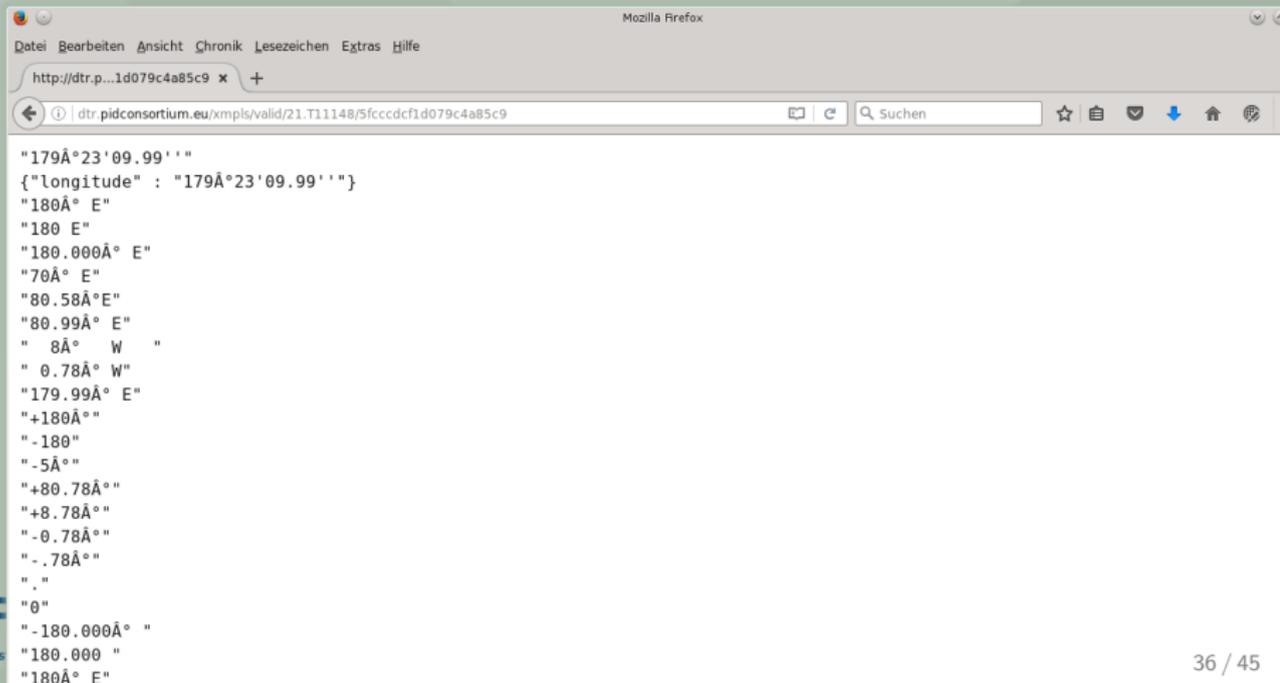
```
wget -O - http://dtr.pidconsortium.eu/  
xmpls/invalid/21.T11148/5fccdcf1d079c4a85c9
```

Arbeiten mit
PIDs

Ulrich
Schwardmann

im Browser evt. Artefakte bei der Unicode-Darstellung

Einführung



The screenshot shows a Mozilla Firefox browser window with the address bar containing the URL `http://dtr.pidconsortium.eu/xmpls/valid/21.T11148/5fccdcf1d079c4a85c9`. The page content displays a JSON object with several fields, each containing a string with various Unicode artifacts and control characters. The artifacts include characters like `179Å°`, `180Å°`, `180.000Å°`, `70Å°`, `80.58Å°`, `80.99Å°`, `8Å°`, `0.78Å°`, `179.99Å°`, `+180Å°`, `-180°`, `-5Å°`, `+80.78Å°`, `+8.78Å°`, `-0.78Å°`, `-.78Å°`, `"`, `°`, `-180.000Å°`, `180.000`, and `180Å°`.

```
"179Å°23'09.99'"  
{ "longitude" : "179Å°23'09.99'" }  
"180Å° E"  
"180 E"  
"180.000Å° E"  
"70Å° E"  
"80.58Å°E"  
"80.99Å° E"  
" 8Å° W "  
" 0.78Å° W"  
"179.99Å° E"  
"+180Å°"  
"-180"  
"-5Å°"  
"+80.78Å°"  
"+8.78Å°"  
"-0.78Å°"  
"-.78Å°"  
"  
"°"  
"-180.000Å° "  
"180.000 "  
"180Å° E"
```


Prüfung einer PID-InfoType-Instanz

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-InfoType-
Instanz

hole den JSON Inhalt einer PID

über Proxy mit REST API: `hdl.handle.net/api/handles/`

Code

```
wget -o /dev/null -O - http://hdl.handle.net/api/handles/  
21.11101/0000-0001-8787-E | python -m json.tool
```

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID



```
uschwar1: bash - Konsole  
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe  
uschwar1@pcscw:~> wget -o /dev/null -O - http://hdl.handle.net/api/handles/21.11101/0000-0001-8787-E|python -m json.tool  
{  
  "handle": "21.11101/0000-0001-8787-E",  
  "responseCode": 1,  
  "values": [  
    {  
      "data": {  
        "format": "string",  
        "value": "90.0\u00b0S"  
      },  
      "index": 1,  
      "timestamp": "2016-11-25T14:45:41Z",  
      "ttl": 86400,  
      "type": "21.T11148/5fcccdc f1d079c4a85c9"  
    },  
    {  
      "data": {  
        "format": "string",  
        "value": "GWDC"  
      },  
      "index": 2,  
      "timestamp": "2016-11-25T14:45:41Z",  
      "ttl": 86400,  
      "type": "INST"  
    },  
    {  
      "data": {  
        "format": "admin"  
      },  
      "index": 3,  
      "timestamp": "2016-11-25T14:45:41Z",  
      "ttl": 86400,  
      "type": "INST"  
    }  
  ]  
}
```

hole den Inhalt eines Typs innerhalb der PID

Arbeiten mit
PIDs

Ulrich
Schwardmann

Code

```
wget -o /dev/null -O - http://hdl.handle.net/api/handles/  
21.11101/0000-0001-8787-E | python -c "import sys, json;  
print json.load(sys.stdin)['values'][0]['data']['value']"
```



```
uschwar1 : bash - Konsole  
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe  
uschwar1@pcscw:~> wget -o /dev/null -O - http://hdl.handle.net/api/handles/21.11101/0000-0001-8787-E|python -c "import sys, json ; print json.load(sys.stdin)['values'][0]['data']['value']"  
90.0°S  
uschwar1@pcscw:~> █  
uschwar1 : bash
```

Einführung

PID
Information
Types

Warum
Datentypen
registrieren?

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PID-
HandleView-
InfoTypes

Prüfung einer
PID-InfoType-
Instanz

nutze die Python Library *PITclient.py*

wget <https://gitlab.gwdg.de/uschwar1/PIT-DTR-Type-Verification/raw/master/PITclient.py>
Die Python Bibliothek enthält eine Reihe nützlicher Funktionen zum Bearbeiten von Werten in PIDs und PID InfoTypes.

- Beispiele (Auswahl):
 - Type-Instanzen in einer PID
 - get a list of all types in this PID
 - contains this PID a type with TypeID?
 - get complete content in PID for given type
 - get value for type in PID
 - validates value in PID against schema of type
 - Content of type definitions
 - get the content in DTR of type
 - get name/description/schema of a type

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-InfoType-
Instanz

Wie wird der Inhalt eines PID-InfoType geprüft

die Funktion `typeInPidValidatesAgainstSchema` in `PITclient.py` validiert eine Typ-Instanz in einer PID gegen das Type-Schema

Arbeiten mit
PIDs

Ulrich
Schwardmann

Einführung

```
PIT_schema_extraction : bash - Konsole
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
Python 2.7.6 (default, Nov 21 2013, 15:55:38) [GCC] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys, json, PITclient
>>> pid = PITclient.PID("21.11101/0000-0001-8787-E")
>>> tyid = PITclient.typeID("21.T11148/5fccdcfd1d079c4a85c9")
>>> pid.getTypeContentInPID(tyid)
{'u'index': 1, u'ttl': 86400, u'type': u'21.T11148/5fccdcfd1d079c4a85c9', u'timestamp': u'2016-11-25T14:45:41Z', u'
data': {u'value': u'90.0\xb0S', u'format': u'string'}}
>>> pid.getTypeValueInPID(tyid)
u'90.0\xb0S'
>>> tyid.getNameOfTypeID()
u'latitude'
>>> tyid.getDescriptionOfTypeID()
u'latitude with degrees in decimal fraction or sexagesimal minutes and seconds (see ISO 19111) and leading [+|-] o
r ending [N|S] from 0 to 90 degrees.'
>>> pid.typeInPidValidatesAgainstSchema(tyid)
True
>>>
```

PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-InfoType-
Instanz

Was fehlt noch ?

Arbeiten mit
PIDs

Ulrich
Schwardmann

- Wie lege ich einen PID-InfoType an ?
- Kann ich auch Typen anderer DTRs nutzen ?
- Wie erfahre ich, welche Typen schon in einer DTR definiert sind?
- Wie geschieht die automatische Schema-Generierung bei ePIC ?
- Welche Teile der PID-InfoType-Definition werden dabei genutzt ?

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Instanz
44/45

Vielen Dank

Arbeiten mit
PIDs

Ulrich
Schwardmann

Questions ???

Kontakt@ePIC:

- support@pidconsortium.eu

Kontakt@GWDG:

- **Ulrich Schwardmann**

T: 0551 201-1542, E: ulrich.schwardmann@gwdg.de

Einführung

PID
Information
Types

Warum
Datentypen
registrieren ?

Wie wird eine
Data Type
Registry
genutzt?

ePIC DTR

Beispiele für
PID-InfoType
Instanzen

Zugriff auf
PIDpe-
HandleView-
InfoTypes

Prüfung einer
PID-Instanz
45/45