Bring FAIR DOs into Light
Towards a generic tool set to make FAIR DOs visible and tangible for users

Thomas Jejkal, et al.
Section Overview

- Design
- Instantiate
- Consume
- Gaps
Section Goals

Learn about good and “not so good” DataTypes and Profiles.

Learn how to use the FAIR DO Cookbook for DataType and Profile design.

Gain awareness of which gaps currently exist and planned activities.
FAIR DO-Design
Example: Big Mac Index

How it works
Purchasing-power parity implies that exchange rates are determined by the value of goods that currencies can buy.

- Differences in local prices – in our case, for Big Macs – can suggest what the exchange rate should be.

Using burgernomics, we can estimate how much one currency is under- or over-valued relative to another.

**Raw index** /GDP-adjusted

**Big Mac exchange rate**

- $5 \text{ } 1:4 \text{ } 20 \text{ yuan}

- 38% under-valued

**Actual exchange rate**

- $1 = 6.4 \text{ yuan}
# FAIR DO-Design

Example: Big Mac Index

## Data

<table>
<thead>
<tr>
<th>date</th>
<th>iso_a3</th>
<th>currency_code</th>
<th>name</th>
<th>local_price</th>
<th>dollar_ex</th>
<th>dollar_price</th>
<th>usd_raw</th>
<th>eur_raw</th>
<th>gbp_raw</th>
<th>jpy_raw</th>
<th>cny_raw</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000-04-01</td>
<td>ARG</td>
<td>ARS</td>
<td>Argentina</td>
<td>2.5</td>
<td>1</td>
<td>2.5</td>
<td>-0.00398</td>
<td>0.05007</td>
<td>-0.16722</td>
<td>-0.09864</td>
<td>1.09091</td>
</tr>
<tr>
<td>2000-04-01</td>
<td>AUS</td>
<td>AUD</td>
<td>Australia</td>
<td>2.59</td>
<td>1.68</td>
<td>1.54166666666667</td>
<td>-0.38579</td>
<td>-0.35246</td>
<td>-0.48645</td>
<td>-0.44416</td>
<td>0.28939</td>
</tr>
<tr>
<td>2000-04-01</td>
<td>BRA</td>
<td>BRL</td>
<td>Brazil</td>
<td>2.95</td>
<td>1.79</td>
<td>1.64804469273743</td>
<td>-0.34341</td>
<td>-0.30778</td>
<td>-0.45102</td>
<td>-0.40581</td>
<td>0.37836</td>
</tr>
<tr>
<td>2000-04-01</td>
<td>CAN</td>
<td>CAD</td>
<td>Canada</td>
<td>2.85</td>
<td>1.47</td>
<td>1.93877551020408</td>
<td>-0.22758</td>
<td>-0.18566</td>
<td>-0.35417</td>
<td>-0.30099</td>
<td>0.62152</td>
</tr>
<tr>
<td>2000-04-01</td>
<td>CHE</td>
<td>CHF</td>
<td>Switzerland</td>
<td>5.9</td>
<td>1.7</td>
<td>3.47058823529412</td>
<td>0.3827</td>
<td>0.45774</td>
<td>0.15609</td>
<td>0.2513</td>
<td>1.90267</td>
</tr>
<tr>
<td>2000-04-01</td>
<td>CHL</td>
<td>CLP</td>
<td>Chile</td>
<td>1260</td>
<td>514</td>
<td>2.45136186770428</td>
<td>-0.02336</td>
<td>0.02964</td>
<td>-0.18342</td>
<td>-0.11618</td>
<td>1.05023</td>
</tr>
</tbody>
</table>

[https://t1p.de/fdo-data](https://t1p.de/fdo-data)
FAIR DO-Design
Example: Big Mac Index

Metadata

<table>
<thead>
<tr>
<th>variable</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Date of observation</td>
</tr>
<tr>
<td>iso_a3</td>
<td>Three-character [ISO 3166-1 country code][ISO 3166-1]</td>
</tr>
<tr>
<td>currency_code</td>
<td>Three-character [ISO 4217 currency code][ISO 4217]</td>
</tr>
<tr>
<td>name</td>
<td>Country name</td>
</tr>
<tr>
<td>local_price</td>
<td>Price of a Big Mac in the local currency</td>
</tr>
<tr>
<td>dollar_ex</td>
<td>Local currency units per dollar</td>
</tr>
<tr>
<td>dollar_price</td>
<td>Price of a Big Mac in dollars</td>
</tr>
<tr>
<td>USD_raw</td>
<td>Raw index, relative to the US dollar</td>
</tr>
<tr>
<td>EUR_raw</td>
<td>Raw index, relative to the Euro</td>
</tr>
<tr>
<td>GBP_raw</td>
<td>Raw index, relative to the British pound</td>
</tr>
<tr>
<td>JPY_raw</td>
<td>Raw index, relative to the Japanese yen</td>
</tr>
<tr>
<td>CNY_raw</td>
<td>Raw index, relative to the Chinese yuan</td>
</tr>
</tbody>
</table>

https://t1p.de/fdo-data
**FAIR DO-Design**
**Solution 1 - The easy one**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>SimpleProfile</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://github.com/rfordata%5B">https://github.com/rfordata[</a>...]</td>
</tr>
<tr>
<td>MetadataLocation</td>
<td><a href="https://github.com/rfordata%5B">https://github.com/rfordata[</a>...]</td>
</tr>
</tbody>
</table>
FAIR DO-Design
Solution 1 - The easy one

• Easily applicable
• Potential increase of findability
• Generic profile and data types

Key         Value
Profile     SimpleProfile
DigitalObjectType  CSVFile
DigitalObjectLocation  https://github.com/rfordata[...]
MetadataLocation  https://github.com/rfordata[...]

• Very limited overall machine-actionability
• Hard to reuse due to lack of machine-readable metadata
• Human evaluation required
FAIR DO-Design Solution 2 - Frictionless Data-based

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>SimpleProfile</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>FrictionlessDataPackage</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://repo/fric_data_pkg.zip">https://repo/fric_data_pkg.zip</a></td>
</tr>
</tbody>
</table>

https://specs.frictionlessdata.io/
FAIR DO-Design
Solution 2 - Frictionless Data-based

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>SimpleProfile</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>FrictionlessDataPackage</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://repo/fric_data_pkg.zip">https://repo/fric_data_pkg.zip</a></td>
</tr>
</tbody>
</table>

- Easily applicable
- Potential increase of findability and reuse
- Generic profile and data types
- Machine-readable data and metadata

- Republishing required
- Limited machine-actionability on FAIR DO-level
- Full download required
- Format knowledge required

https://specs.frictionlessdata.io/
# FAIR DO-Design

## Solution 3 - FAIR DO times two

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>SimpleProfile</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://github.com/rfordata%5B...%5D">https://github.com/rforda...</a></td>
</tr>
<tr>
<td>MetadataLocation</td>
<td>Metadata-FAIR-DO</td>
</tr>
</tbody>
</table>
### FAIR DO-Design

#### Solution 3 - FAIR DO times two

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>SimpleProfile</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://github.com/rfordata">https://github.com/rfordata</a> [...]</td>
</tr>
<tr>
<td>MetadataLocation</td>
<td>Metadata-FAIR-DO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>SimpleProfile</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVTableDefinition</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="http://repo/CSVTableDefinition.json">http://repo/CSVTableDefinition.json</a></td>
</tr>
</tbody>
</table>
### FAIR DO-Design
**Solution 3 - FAIR DO times two**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>SimpleProfile</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://github.com/rfordata%5B">https://github.com/rfordata[</a>...]</td>
</tr>
<tr>
<td>MetadataLocation</td>
<td>Metadata-FAIR-DO</td>
</tr>
</tbody>
</table>

- Enrichment of existing (legacy) data
- Potential increase of findability and reuse
- Generic profile and data types
- Machine-actionable via linked metadata
- Republishing of metadata required
- Requires metadata download and format knowledge
- Increased complexity
# FAIR DO-Design Solution 4 - All-in-one

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Profile4MachineActionableCSV</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://github.com/rfordata">https://github.com/rfordata</a> [...]</td>
</tr>
<tr>
<td>CSVTableDef</td>
<td>JSON-Object</td>
</tr>
</tbody>
</table>
**FAIR DO-Design Solution 4 - All-in-one**

- Easily applicable
- Potential increase of findability and reuse
- Specific but reusable profile and data types
- Machine-actionability on PID-level

- Huge entry barrier as profile and complex data type are required
- Unfolds full potential only if broadly applied
FAIR DO-Design
- The Good, the Bad, and the Ugly -

https://www.menti.com/al9tppotna6g
FAIR DO Cookbook

Features
• Collection of recipes on common FAIR DO tasks
• Focus on concrete FAIR DO implementation
• Shows good practices and alternate paths

Target Audience
• Developers
• FAIR DO creators and designers

Status
• Release

https://t1p.de/fdo-cb-about
# The FAIR DO Cookbook and DataTypes

## PID

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Profile4MachineActionableCSV</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td>[<a href="https://github.com/rfordata%5B">https://github.com/rfordata[</a>...]]</td>
</tr>
<tr>
<td>CSVTableDef</td>
<td>JSON-Object</td>
</tr>
</tbody>
</table>

[https://t1p.de/fdo-cb-dt](https://t1p.de/fdo-cb-dt)
# The FAIR DO Cookbook and DataTypes

## PID

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Profile4MachineActionableCSV</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://github.com/rfordata">https://github.com/rfordata</a> [...]</td>
</tr>
<tr>
<td>CSVTableDef</td>
<td>JSON-Object</td>
</tr>
</tbody>
</table>

## Ingredients

- Data Type Registry (DTR), e.g. [http://dtr-test.pidconsortium.eu/](http://dtr-test.pidconsortium.eu/)
- Registered user account at Data Type Registry
- Web Browser
The FAIR DO Cookbook and DataTypes

XML Table:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Profile4MachineActionableCSV</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://github.com/rfordata%5B%E2%80%A6">https://github.com/rfordata[…</a>]</td>
</tr>
<tr>
<td>CSVTableDef</td>
<td>JSON-Object</td>
</tr>
</tbody>
</table>

Ingredients:
- Data Type Registry (DTR), e.g. [http://dtr-test.pidconsortium.eu/](http://dtr-test.pidconsortium.eu/)
- Registered user account at Data Type Registry
- Web Browser

Work Steps:
1. Search data type registry for suitable data type
2. Describe/model envisioned data type (offline)
   2.1. Decide on Data Type's level of detail
   2.2. DTR Inquiry
3. Register data type(s) in data type registry
   3.1. Register new Child-Data Type(s)
   3.2. Register extended Child-Data Type(s)
   3.3. Register main Data Type
Decide on a DataType’s Level of Detail

What’s the purpose of the new DataType

Uniquely identify resource with implicit characteristics
- Effort
- Reusability
- Validation

Uniquely identify resource and certain properties
- Effort
- Reusability
- Validation

Uniquely identify resource and its structure
- Effort
- Reusability
- Validation

Examples:
- „CSVFile“
- „SimpleProfile“
- „CSVTableDef“
DTR Inquiry

CSV Dialect Descriptor

```json
{
  "properties":
    {
      "csvddfVersion":
        {
        },
      "delimiter":
        {
        },
      "doubleQuote":
        {
        },
      "lineTerminator":
        {
        },
      "nullSequence":
        {
        },
      "quoteChar":
        {
        },
      "escapeChar":
        {
        },
      "skipInitialSpace":
        {
        },
      "header":
        {
        },
      "commentChar":
        {
        },
      "caseSensitiveHeader":
        {
        }
    }
}
```

CSV Field Description

```json
{
  "fields":
    {
      "type": "array",
      "minItems": 1
    }
}
```

```json
{
  "items":
    {
      "title": "Table Schema Field",
      "type": "object",
      "oneOf":
        {
          "0":
            {
              "type": "object",
              "title": "String Field",
              "description": "The field contains strings, that is, sequences of characters.",
              "required":
                {
                }
            }
        }
    }
}
```

```json
{
  "properties":
    {
      "name":
        {
        },
      "title":
        {
        },
      "description":
        {
        },
      "example":
        {
        },
      "type":
        {
        },
      "format":
        {
        },
      "constraints":
        {
        },
      "rdfType":
        {
        },
      "examples":
        {
          "0":
            {
              "name": "name",
              "type": "string"
            },
          "1":
            {
              "name": "name",
              "type": "string",
              "format": "email"
            },
          "2":
            {
              "name": "name",
              "type": "string",
              "constraints": {"minLength": 3, "maxLength": 35}
            }
        }
    }
}
```
## DTR Inquiry

### CSV Dialect Descriptor

- `properties`
- `csvddfVersion`
- `delimiter`
- `doubleQuote`
- `lineTerminator`
- `nullSequence`
- `quoteChar`
- `escapeChar`
- `skipInitialSpace`
- `header`
- `commentChar`
- `caseSensitiveHeader`

### DTR Inquiry Result (Reuse, Extend, Create)

<table>
<thead>
<tr>
<th>Property</th>
<th>DTR Inquiry Result (Reuse, Extend, Create)</th>
</tr>
</thead>
<tbody>
<tr>
<td>csvddfVersion</td>
<td></td>
</tr>
<tr>
<td>delimiter</td>
<td></td>
</tr>
<tr>
<td>doubleQuote</td>
<td></td>
</tr>
<tr>
<td>lineTerminator</td>
<td></td>
</tr>
<tr>
<td>nullSequence</td>
<td></td>
</tr>
<tr>
<td>quoteChar</td>
<td></td>
</tr>
<tr>
<td>escapeChar</td>
<td></td>
</tr>
<tr>
<td>skipInitialSpace</td>
<td></td>
</tr>
<tr>
<td>header</td>
<td></td>
</tr>
<tr>
<td>commentChar</td>
<td></td>
</tr>
<tr>
<td>caseSensitiveHeader</td>
<td></td>
</tr>
</tbody>
</table>
## Register Data Types

<table>
<thead>
<tr>
<th>Property</th>
<th>DTR Inquiry Result (Reuse, Extend, Create)</th>
</tr>
</thead>
<tbody>
<tr>
<td>csvddfVersion</td>
<td>Reuse version-number (21.T11148/ac9849005793b63ac780)</td>
</tr>
<tr>
<td>delimiter</td>
<td>Create</td>
</tr>
<tr>
<td>doubleQuote</td>
<td>Create</td>
</tr>
<tr>
<td>lineTerminator</td>
<td>Create</td>
</tr>
<tr>
<td>nullSequence</td>
<td>Create</td>
</tr>
<tr>
<td>quoteChar</td>
<td>Create</td>
</tr>
<tr>
<td>escapeChar</td>
<td>Create</td>
</tr>
<tr>
<td>skipInitialSpace</td>
<td>Create</td>
</tr>
<tr>
<td>header</td>
<td>Create</td>
</tr>
<tr>
<td>commentChar</td>
<td>Create</td>
</tr>
<tr>
<td>caseSensitiveHeader</td>
<td>Create</td>
</tr>
</tbody>
</table>

- **Create DataTypes** *(Extend/Create)*
- Work from bottom to top
- Document DataType PIDs in table

<table>
<thead>
<tr>
<th>Property</th>
<th>DTR Inquiry Result (Reuse, Extend, Create)</th>
</tr>
</thead>
<tbody>
<tr>
<td>csvddfVersion</td>
<td>Reuse version-number (21.T11148/ac9849005793b63ac780)</td>
</tr>
<tr>
<td>delimiter</td>
<td>Create (21.T11148/f1627ce853823247507b)</td>
</tr>
<tr>
<td>doubleQuote</td>
<td>Create (21.T11148/f1627ce853823247507b)</td>
</tr>
<tr>
<td>lineTerminator</td>
<td>Create (21.T11148/ab56fee853823247507b)</td>
</tr>
<tr>
<td>nullSequence</td>
<td>Create (21.T11148/f1622135467223456249)</td>
</tr>
<tr>
<td>quoteChar</td>
<td>Create (21.T11148/231c28758724856f703e)</td>
</tr>
<tr>
<td>escapeChar</td>
<td>Create (21.T11148/8608328253c27f5e1477)</td>
</tr>
<tr>
<td>skipInitialSpace</td>
<td>Create (21.T11148/734172af52685270c83)</td>
</tr>
<tr>
<td>header</td>
<td>Create (21.T11148/8c7227658143875e03)</td>
</tr>
<tr>
<td>commentChar</td>
<td>Create (21.T11148/18f2786724c555306b27)</td>
</tr>
<tr>
<td>caseSensitiveHeader</td>
<td>Create (21.T11148/87253e3041f5786c72862)</td>
</tr>
</tbody>
</table>

- **Create top-level DataType**
## The FAIR DO Cookbook and Profiles

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Profile4MachineActionableCSV</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://github.com/rfordata">https://github.com/rfordata</a> [...]</td>
</tr>
<tr>
<td>CSVTableDef</td>
<td>JSON-Object</td>
</tr>
</tbody>
</table>

![PID](https://t1p.de/fdo-cb-kip)
The FAIR DO Cookbook and Profiles

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Profile4MachineActionableCSV</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://github.com/rfordata%5B">https://github.com/rfordata[</a>...]</td>
</tr>
<tr>
<td>CSVTableDef</td>
<td>JSON-Object</td>
</tr>
</tbody>
</table>

Ingredients

- Data Type Registry (DTR), e.g. http://dtr-test.pidconsortium.eu/
- Registered user account at Data Type Registry
- Web Browser
The FAIR DO Cookbook and Profiles

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Profile4MachineActionableCSV</td>
</tr>
<tr>
<td>DigitalObjectType</td>
<td>CSVFile</td>
</tr>
<tr>
<td>DigitalObjectLocation</td>
<td><a href="https://github.com/rfordata">https://github.com/rfordata</a> [...]</td>
</tr>
<tr>
<td>CSVTableDef</td>
<td>JSON-Object</td>
</tr>
</tbody>
</table>

Ingredients

- Data Type Registry (DTR), e.g. [http://dtr-test.pidconsortium.eu/](http://dtr-test.pidconsortium.eu/)
- Registered user account at Data Type Registry
- Web Browser

Work Steps

1. Search data type registry for suitable KIP
2. Describe/model envisioned KIP
   2.1. Call to mind basic KIP principles
   2.2. Decide on KIP contents
3. Register new KIP in Data Type Registry

KIP = Kernel Information Profile
Decide on a Profile

Is there a KIP which can be reused directly?

Yes, fully reusable → Reuse identified KIP

Yes, partly reusable → Extend identified KIP and add reference under „Applicable Standards or Recommendations“

No → Create new KIP
Decide on a Profile

Is there a KIP which can be reused directly?

Yes, fully reusable

Profile

DigitalObjectType

DigitalObjectLocation

CSVTableDef

No

Reuse identified KIP

Create new KIP

Standards or Recommendations
Register Profile

1. Identify and create all missing DataTypes, just like before.
2. Copy existing/create new profile.
3. In case of copy, refer to original profile.
4. Name it.
5. Add all attributes and their type information (must be unique!).
6. Save it and become a gardener.
Section Summary

Learn about good and „not so good“ DataTypes and Profiles.

• There is no clear „good“, „bad“, or „ugly“.
• Effort strongly depends on what you want to achieve.
• Will improve with a growing number of DataTypes and Profiles.

Learn how to use the FAIR DO Cookbook for DataType and Profile design.

• Comprehensive collection of good practices with clear focus on single implementation can lower entry barriers.
• More of such guidelines required?

Gain awareness of which gaps currently exist and planned activities.

• It can be a long way from the idea to a new FAIR DO.
• DataType Registry requires a major overhaul.
• FAIRCORE4EOSC and alternative implementation at KIT.
Section Goals

Understand different approaches of instantiating FAIR DOs.

Learn how to use available tools for FAIR DO creation.

Get informed about planned activities.
Two Sides of a Similar Medal

+ Native FAIR DOs with authoritative contents
+ Supports vision of one-for-all interface
+ Built-in support for extended operations

- Requires infrastructure providers to implement DOIP or operate proxy
- Pure DOIP is TCP/IP-based, i.e., very low-level
- Seems to be poorly adopted

DOIP = Digital Object Interface Protocol
Two Sides of a Similar Medal

+ Non-invasive application of FAIR DOs on top of existing infrastructures
+ Accessible via PID resolver
+ May accumulate metadata from different sources

~ Location preferably (open) Web-resolvable, HTTP-based API endpoint

- PID record non-authoritative source, regular validation required
- No safety net while creating FAIR DOs
- Plain HTTP calls required
The Typed PID Maker

+ Supports use of different PID resolvers
+ Includes validation of PID records
+ Supports caching of DataTypes and Profiles
+ Allows PID customization
+ Allows fine-grained authorization
+ Supports indexing and search

- Another wrapper
- Standalone service that requires to be operated
- Still, plain HTTP calls required

thomas.jejkal@kit.edu
The Typed PID Maker – Create a FAIR DO

curl --location --request POST 'http://typed-pid-maker-instance/api/v1/pit/pid/' \
--header 'Content-Type: application/json' \
--data-raw \\

Further Endpoints:

GET http://typed-pid-maker-instance/api/v1/pit/pid/{pid}
PUT http://typed-pid-maker-instance/api/v1/pit/pid/{pid}
GET http://typed-pid-maker-instance/api/v1/pit/known-pid
POST http://typed-pid-maker-instance/api/v1/search

https://t1p.de/tpidm-web
https://t1p.de/tpidm-source
FAIR DO Builder

Features
• Easy-to-use Web UI
• Form-based FAIR DO creation including basic validation
• Visual linking of FAIR DOs

Target Audience
• Developers
• FAIR DO creators and designers

Status
• Early access (single profile)

https://t1p.de/fdo-builder
FAIR DO Builder

Features
• Easy-to-use Web UI
• Form-based FAIR DO creation including basic validation
• Visual linking of FAIR DOs

Target Audience
• Developers
• FAIR DO creators and designers

Status
• Early access (single profile)

Future Plans
• Integrate multi-Profile support
• Integrate dynamic DataType support
• Improve user experience, e.g., detailed error messages
• Support bulk-creation of FAIR DOs
iDORIS
Integrated Data Type and Operations Registry with Inheritance System

Features
• Registry for DataTypes, Profiles, and Operations
• Inheritance support for DataTypes and Profiles
• Support for DataType-Operation matching

Target Audience
• Developers
• Infrastructure providers

Status
• Conceptual phase (tdb. Q4 2024)
Section Summary

Understand different approaches of instantiating FAIR DOs.
• **Learned about FAIR DOs via DOIP and PID resolver.**
• **Typed PID Maker for additional functionality.**
• **(Many) other approaches exists, harmonization and mapping required.**
• **For production use automation/integration required.**

Learn how to use available tools for FAIR DO creation.
• **Low-level APIs available for automation/integration, in-depth knowledge required.**
• **Introduced FAIR DO Builder as easy-to-use UI for humans.**
• **Further tooling and flexibility required.**

Get informed about planned activities.
• **Improvements and extensions for existing tooling.**
• **iDORIS to improve DataType/Profile creation and adding Operation support.**
• **Ideas and contributions are always welcome!**
Section Goals

Learn how to possibly consume FAIR DOs depending on the target audience.

Future plans and collection of ideas.
FAIR DO Search

Features
• Easy-to-use Web UI
• Runs on top of Typed PID Maker
• Enhanced search via Elasticsearch
• Facet support for FAIR DO filtering

Target Audience
• Scientific users

Status
• Early access (single profile, static facets)

https://t1p.de/fdo-search
FAIR DO Search

Features
• Easy-to-use Web UI
• Runs on top of Typed PID Maker
• Enhanced search via Elasticsearch
• Facet support for FAIR DO filtering

Target Audience
• Scientific users

Status
• Early access (single profile, static facets)

Future Plans
• Customization of facets and result view
• Integration of multi-Profile support
• Allow better reusability

DEMO
https://demo.datamanager.kit.edu/elastic-search-fdo.html
FAIR-DOscope

Features
• Visualization of and navigation via PID records
• DataType-driven rendering of key/values
• Visualization of FAIR DO relationships

Target Audience
• Developers
• FAIR DO creators and designers

Status
• Released (v1.1.0)

https://t1p.de/fdoscope
**FAIR-DOscope**

**Features**
- Visualization of PID records
- DataType-driven rendering of key/values
- Visualization of FAIR DO relationships

**Target Audience**
- Developers
- FAIR DO creators and designers

**Status**
- Released (v1.1.0)

**Future Plans**
- Dynamic resolution of DataTypes/Profiles
- Improved graph visualization
- Improved integration with other tools
PID-Component

Features
• Web component to render FAIR DOs
• Seamless integration into any Web page
• Extensible rendering, e.g. ORCiDs

Target Audience
• Platform developers (integration)
• Scientific users

Status
• Released (v0.0.11)

https://t1p.de/pid-com
PID-Component

Features
• Web component
• Seamless integration
• Extensible rendering

Target Audience
• Platform developers (integration)
• Scientific users

Status
• Released (v0.0.11)

Features
• Web component to render FAIR DOs
• Seamless integration into any Web page
• Extensible rendering, e.g. ORCiDs

Future Plans
• Additional renderers (ROR, IGSN, etc.)
• Any idea?

https://www.menti.com/al9573c43ce3
Section Summary

Learn how to possibly consume FAIR DOs depending on the target audience.

- FAIR DO Search allows search in huge FAIR DO collections.
- FAIR-DOscope enables fast inspection of created FAIR DOs.
- PID-Component for seamless integration into existing Web UIs.
- Move towards reusability and integrability.

Future plans and collection of ideas.

- Focus mainly on flexibility and customization.
- Ideas and contributions are always welcome!
Section Goals

Identification of other missing tools/components.

Clarification of open questions.

Summary and closing of session.
Ideas for Missing Tools/Components

- **Design**
  - DataType Designer UI

- **Instantiate**
  - Metadata2Record Mapper

- **Consume**
  - FAIR DO Finder

- **Others**
  - Maintenance tools
Summary

Got to know the 🍔-index.

Got 4 ideas on how to represent CSV data as FAIR DO.

Completed 2 Mentimeter.

Learned about 7 tools.

Gave 90 minutes of grateful attention.
Acknowledgements

• The Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the National Research Data Infrastructure – NFDI 38/1 – project number 460247524
• The Joint Laboratory Model and Data driven Materials Characterization (JL MDMC), a cross-centre platform of the Helmholtz Association
• NFFA-Europe-Pilot (EU H2020 – n. 101007417)
• The research program “Engineering Digital Futures” of the Helmholtz Association of German Research Centers
• The Helmholtz Metadata Collaboration Platform
Image References

Slide 1, FAIR DO, Schultes, E., & Wittenburg, P. (2019). FAIR Principles and Digital Objects: Accelerating Convergence on a Data Infrastructure., 10.1007/978-3-030-23584-0_1
Slide 16, Good, Bad, Ugly, https://i.pinimg.com/originals/38/75/71/3875711647b09bfb8d6e9b2da1c7838f.png
Slide 29, Gardener, https://i.redd.it/wra9pru80t171.jpg
Slide 38, Construction, https://media.istockphoto.com/id/185244309/de/foto/arbeiten-in-bearbeitung.jpg?s=612x612&w=0&k=20&c=eaP8_iYyCDOAuXn8niT-Dp3arHnl-kgZPGBt8wXQnGQ=
Slide 41, Search, https://bibliotecnica.upc.edu/sites/default/files/actualitat/11986_sgb_googledata.png
Slide 50, Questions, https://images.squarespacecdn.com/content/v1/5ea237e587e03021f9e8cc2/1591321482139-5N4EC0SIATZQYDTX8I3G/Question-mark.jpg