
# PID Information Types WG + x

## 3rd RDA collaboration meeting, KIT, June 11, 2015

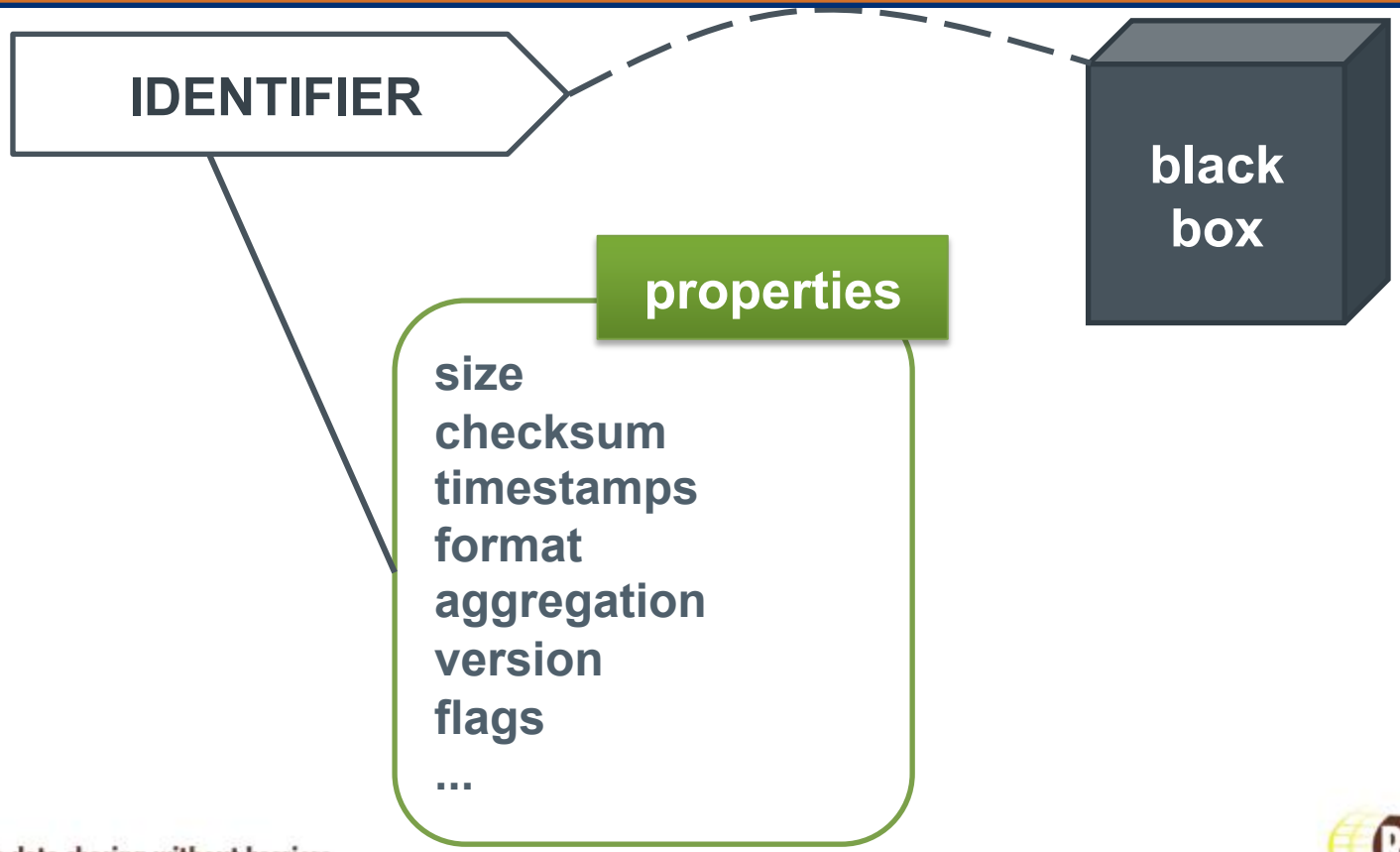
Tobias Weigel, Tim DiLauro, Stephan Kindermann

research data sharing without barriers

rd-alliance.org

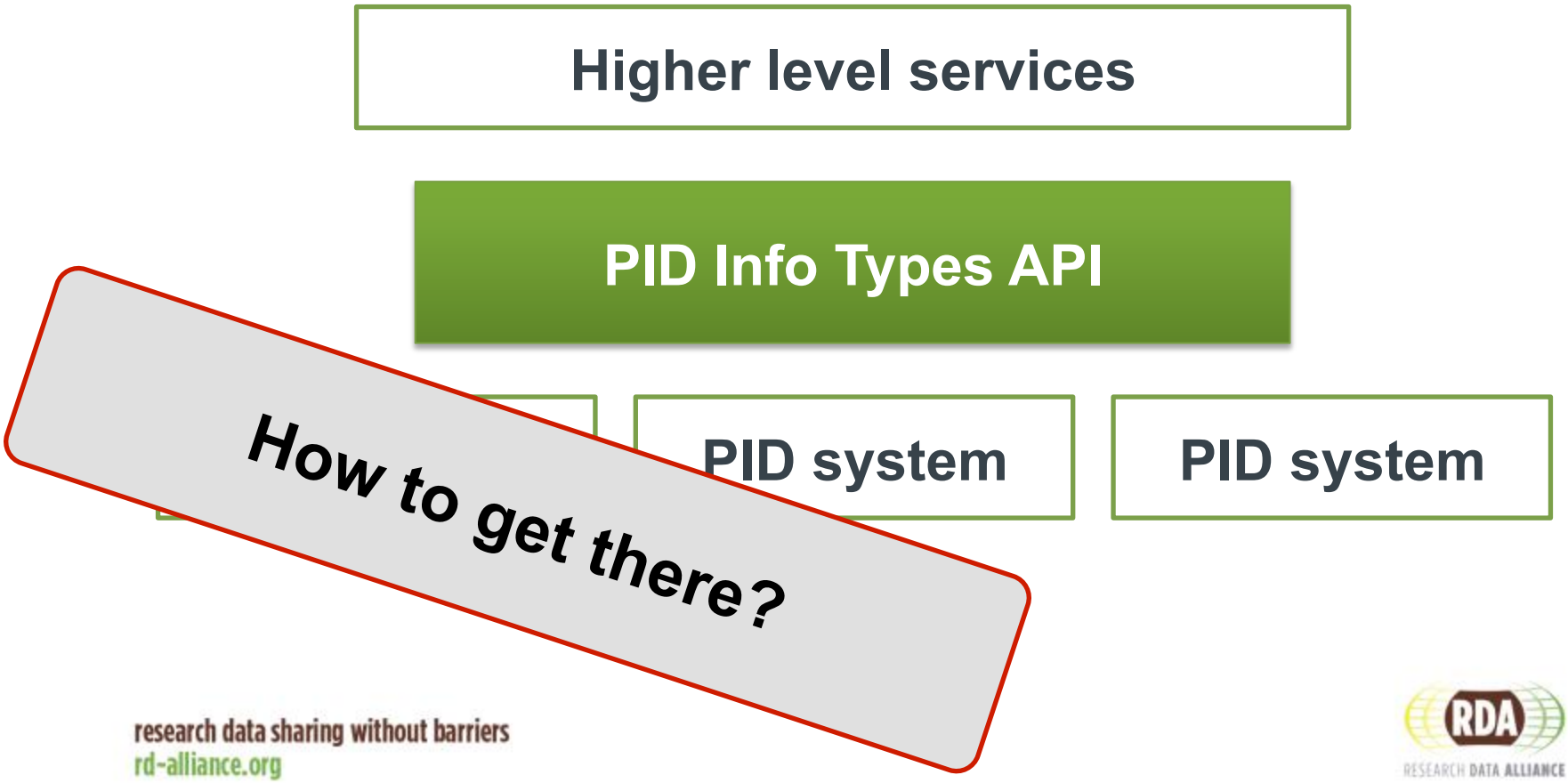
# Types for information directly associated with PIDs

**A Persistent Identifier is a long-lasting ID represented by a string that uniquely points to a DO and that is intended to be persistently resolvable to access meaningful, current state information about the identified DO.** ***(from DFT wiki)***



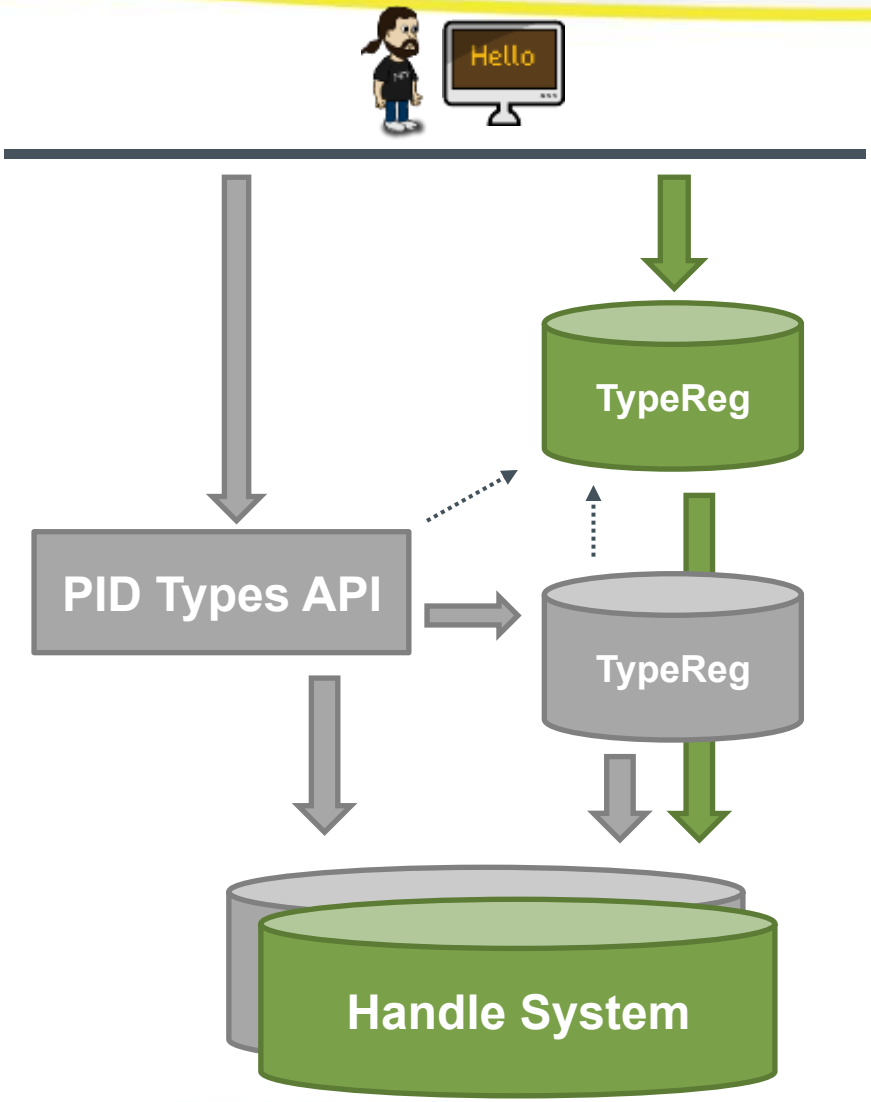
# Something needs to reach across PID systems

The PID Information Types API serves two purposes:
Facilitating **typing** and enabling **interoperability** across PID Systems.

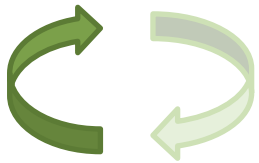


# Typing and Type Registries






- Two usage scenarios for TypeReg:
  - **Typing of data entities**



  - **Typing of PID record value fields**
- Reference to data type in properties record

# Example type list from the final report – to be continued...?



| Name | Range | Identifier | Flags |
|---|---|---|---|
| **Type: Citation Information (EXAMPLE namespace) 11314.2/d5396a97c316a0eaca055846ba4233ac** | | | |
| Title | STRING | 11314.2/07841c3f84cbe0d4ff8687d0028c2622 | |
| Creator | STRING | 11314.2/31810b2c24913929bb5e0d4d949de9f7 | |
| Publication date | DATE | 11314.2/daed5901fbbe2570ee95c4009c739de2 | |
| Language | STRING | 11314.2/56211d62153b3500ce3b16cf86d6b403 | optional |
| License | STRING | 11314.2/2f305c8320611911a9926bb58dfad8c9 | optional |
| **Type: System level access information (EXAMPLE namespace) 11314.2/09d35f22e48b60284029ba51c17e2944** | | | |
| Creation date | DATE | 11314.2/6b3e1230d1b68965e290b16a43d2f46d | |
| Deletion date | DATE | 11314.2/7e78be9736ad7f6bb5fb31218821eba5 | optional |
| Permissions | STRING | 11314.2/d057258f7b406fd9aad5a3893aba8208 | optional |
| Checksum | STRING | 11314.2/56bb4d16b75ae50015b3ed634bbb519f | |
| Object size (in bytes) | STRING | 11314.2/0006e2b8e2f6e1ecce836e593bed38ae | |
| **Type: Aggregation information (EXAMPLE namespace) 11314.2/699d487eff50c2e10982f4b85ed053a9** | | | |
| Parent object identifier | IDENTIFIER | 11314.2/f9e66e5f64ba3179d8f1e64138c69e04 | optional |
| Child object identifier | IDENTIFIER | 11314.2/f8db9e3b5f97aa8168fbd59788476375 | optional |
| **Type: Versioning information (EXAMPLE namespace) 11314.2/6b507d787dd06e4eb8f23b5bb56ae8bb** | | | |
| Predecessor identifier | IDENTIFIER | 11314.2/467d9ba30e2d9879fd9d483f319e462c | optional |
| Successor identifier | IDENTIFIER | 11314.2/fc78024cb9dac0b0a80ed631ea650d4b | optional |
| **Type: Preliminary example for EUDAT core information (EUDAT namespace) 11314.2/5f45666fc8689e3565728ca512c1b5e7** | | | |
| Checksum | STRING | see above | |
| Format | STRING | 11314.2/1a4f53a28b72d4bf4f8fdda7a2089595 | |
| Data identifier | IDENTIFIER | 11314.2/24dd85c4a3d39fb0d7e83a510a5041c6 | |
| Metadata identifier | IDENTIFIER | 11314.2/58a44100d2bcd1a34fb87eb87bc6f701 | |
| Repository of Record | IDENTIFIER | 11314.2/5546b0166091d9ae869f081f5548f3fc | |
| Mutability flag | BOOLEAN | 11314.2/7c81e954eaead6a2f772abd83986d3e9 | |
| Landing page address | URL | 11314.2/66af2639d388977e81b85f6413df1e2c | |
| Date of deposition | DATE | 11314.2/35837218f18dcc54a2d32e0fb30fa7fb | |

## Brief overview of the actual API methods

- The API focuses on reading and making sense of typed PID record information.
- There are interfaces via Java and HTTP.

**GET /peek/{identifier}**

**GET /property/{identifier}**

**GET /type/{identifier}**

**GET /pid/{identifier}?...**

- Conformance information i... ...ven

**Needs further refinement...**
PIT v2.0?

## How to get the PIT output

- Demonstrator at RZG and documentation:
  http://smw-rda.esc.rzg.mpg.de/PitApiGui/
  http://smw-rda.esc.rzg.mpg.de/apidocs/
- Prototype source code available via git:
  `git clone git://redmine.dkrz.de/rdapit.git`
- Final overview report available from the RDA websites:
  https://rd-alliance.org/groups/pid-information-types-wg.html

- More formal outcome package in the loop.
- Licenses: CC0 / simple BSD

## A bit of reflection...

- Even with very simple information, each use case favors a different set of types
- There is no single set of types fitting all cases – we have to live with that in practice and look towards the Type Registries to help us
- Community processes must define types from practical adoption

## How does the PID Information Types effort continue?

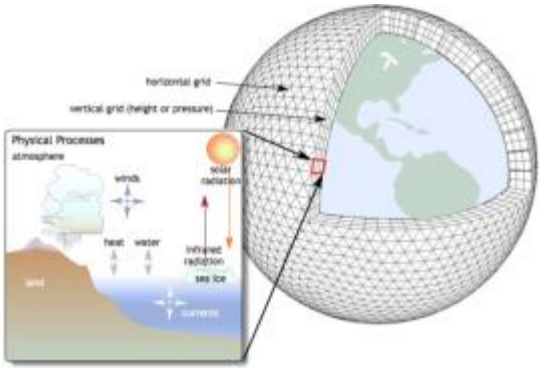
- The API is a prototype that has to see further refinement further practical adoption
- DKRZ follows through with future plans in the context of an international data infrastructure (ESGF) and EUDAT
  - This will also shed more light on essential types
- Interest was also stated by e.g. Deep Carbon Observatory and the Materials Genome Initiative

# The Earth System Grid Federation and CMIP6

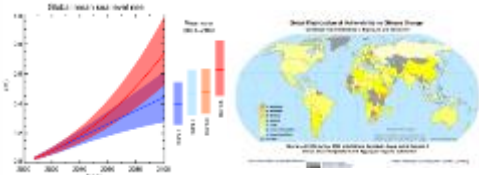
- CMIP6: worldwide coordinated climate simulations (>28 modeling centers, >40 models, CMIP5)
- ESGF data federation: worldwide distributed e-infrastructure for climate data distribution

End users:

- Climate modeling community
- Climate impact community
- Interdisciplinary




http://www.cmmap.org/learn/modeling/whatIs2.html



**Data preparation and ingest** → **Data distribution** → **Downstream user tools**

research data sharing without barriers
rd-alliance.org

# Getting the results into practice poses new challenges...

- Questions asked:
  - What information do we put in the PID records?
  - How do we combine it with catalog information?
  - How do we structure the records?
  - How do we keep the records sane – what are useful processes?
- ESGF governance process
  - CMIP6 data files from core nodes will receive PIDs.
  - Prefixes not clear yet – possibly via EPIC/DONA
- Development possibilities:
  - Further refinement of PIT API
  - Downstream tools that make use of PITs and the Type Registrsy

# PID Collections WG – Case Statement is ready!

- Proposed co-chairs:
  Bridget Almas, Tobias Weigel, Tom Zastrow

- Challenge: No cross-community approach for building and managing collections
- Acknowledge collections as primary objects
- Key deliverables:
  - Collection models – communities need different models, including fragments; also for citation, deep references
  - API and demonstrator – work with collection metadata, CRUD for items
- Kick-off in Paris – first task is to gather use cases
- Need to talk to IGs!

# PIT and Collections in the Data Fabric

- PID Types and collections are part of the Data Fabric – Components

- We need further detailed models that explain the interaction – who will build them?
  - Need to become much more concrete
  - Feedback through practice!

# Take-home messages

- Political consensus in a community/infrastructure is crucial – challenge too big for single institution
- Keep It Simple & Stupid – also in the future

- Practice in EUDAT and ESGF
- PID Collections WG

- Thank you for your attention.