ePIC - Persistent Identifiers for eResearch

Ulrich Schwardmann

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG)

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

30 August 2016, Garching

ePIC
Persistent Identifiers for eResearch

ePIC

Persistent Identifiers for eResearch

Ulrich Schwardmann

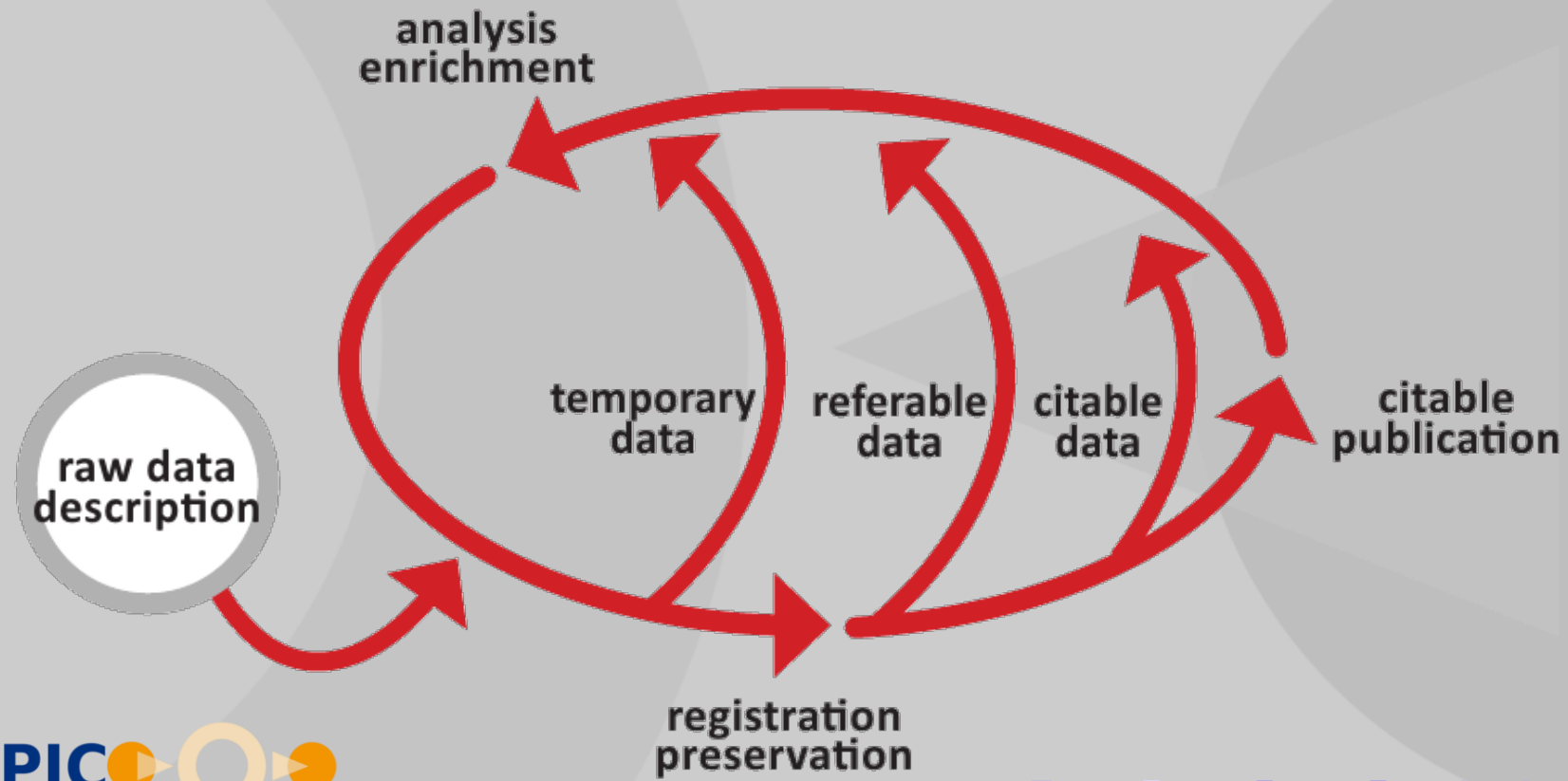
Luce (2008): *... eResearch refers to the development of, and the support for, advanced information and computational technologies to* **enhance** *all phases of research processes. ...*

ePIC provides **enhanced** Persistent Identifier
in order to **accelerate** the Research Life Cycle

The Research Data Life Cycle

data intensive research is highly collaborative

- scientists share data already in an early state
- ad hoc techniques for sharing are often prohibitive
- reliable references can accellerate the Research Life Cycle



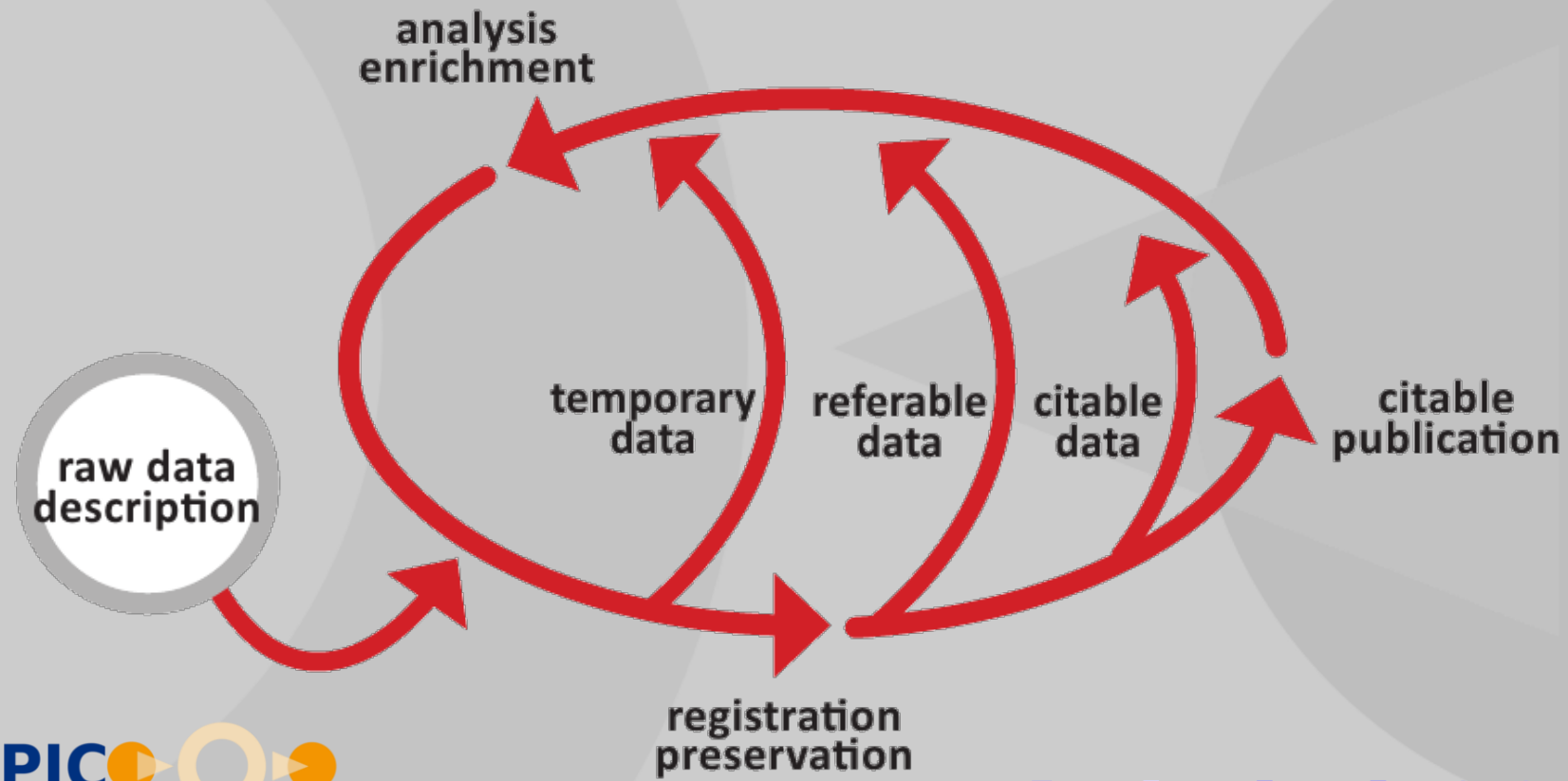
Ulrich
Schwardmann

ePIC
Persistent Identifiers for eResearch

The Research Data Life Cycle

data intensive research is highly collaborative

- scientists share data already in an early state
- ad hoc techniques for sharing are often prohibitive
- reliable references can accellerate the Research Life Cycle



Ulrich
Schwardmann

ePIC
Persistent Identifiers for eResearch

The ePIC Members

build a network of currently six strong scientific service providers signed a contract

- to ensure a reliable and persistent identifier infrastructure
- devoted to the needs of the research community at large.

Mayor focus: the referability of data

- with finer granularity and
- for sharing during the research process

Ulrich Schwardmann

The ePIC Members

- signed a contract to ensure
  - high availabitity for the PID services and
  - a long term perspective at organisational level
- have agreed on a Quality of Service Level,
- have implemented a redundancy scheme
- share the same service, the same API
- and the same framework of policies

Ulrich Schwardmann

The ePIC Members

- ePIC has minted already more then 30 million PIDs
- uses currently about 30 prefixes
- two have more than 8 million, three more than 2 million PIDs
- ePIC takes an important part in the global Handle System
- GWDG is on behalf of ePIC a Multi Primary Adminstistrator (MPA) in **DONA**
- GWDG provides on behalf of ePIC a **DONA MPA GHR** and a **Handle Proxy Server**
- ePIC is interested to run a data type registry as part of **Cordra**

Quality of Service

- Conditions of Operation
  - user management, privacy protection and secrecy
- incident management and monitoring
- support system with agreed responsabilties
- certification of ePIC PID services
- several policies for PID minting and update agreed
  - others are still under discussion
- quality of resolution
  - audits can be requested

Sharing Data in Research

Ulrich Schwardmann

- data sharing of early results requires
  - a reliable framework of trust
  - transparent and standardized policies
  - registration for referable data
    - strong coupling between data and metadata
    - stable references
  - but also a review procedure to delete data

PIDs can be the pivot to fulfil these requirements

Flexilibity

Ulrich Schwardmann

- ePIC PIDs can be enhanced with policies for sharing data for
  - stable references
  - transparent embargo and deletion rules
  - transparent provenance
  - direct access to data and metadata
- flexible rules for data registration granularity
- performance of resolution and minting
  - depending on needs for speed and policy

Types

Ulrich Schwardmann

- are additional metadata stored in the PID database
- intended to be directly accessible independent of any redirection
- typical cases are
  - checksum
  - mime type (incl. version)
  - embargo time
  - expiration date
  - add. metadata file
  - basic Dublin Core

Granularity

digital objects shared with other scientists for investigation have *finer granularity*

- use cases are
  - single experiments
  - simulation output and/or parameter sets
  - single files, tables, pictures, single scanned pages or video/audio sequences
  - snapshots of sensor outputs (dynamic data)
  - software and software versions
- in some cases these sets of digital objects are highly structured
  - and accessible by parameterized services
  - here also *templates* or *fragment identifiers* can be a solution

Templates or Fragment Identifier

rules for strings appended to the PID (see IETF RFC 6570)

- often used to address *service functions* operating on digital objects
- the template implementation in the handle system is simply a rewrite rule
- delimiter and replacement is configurable at prefix level
- *example*
  - delimiter is `@`, which is replaced by `?`
    `11858/00-ZZZZ-0000-0001-CCD1-4@aaa=bbb&ccc=ddd`
  - translates into:
    `http://wwwuser.gwdg.de/~tkalman/downloads/formtest.php?aaa=bbb&ccc=ddd`
- be **careful**: fragment identifier are much **less persistent** then the PIDs itself
- the rewrite rule can be much more complex:
  - replace semantic string elements like URLs by other strings
  - use delimiter strings instead of characters
  - ...

Ulrich Schwardmann

Variation in the Needs for Close Metadata

properties of digital objects and there needs for close metadata

	checksum	checksum fixed	metadata pointer	author	basic Dublin Core	mime type & version	creation date	expiration date	embargoes	versioning	templates & services	fine data granularity
edition	*		*		*					*	*	
object of edition	*			*			*		*		*	*
electronic lab notebook	*		*		*					*	*	
notebook entry		*	*	*					*			*
experiment output	*			*		*	*	*	*		*	*
image library			*		*							
image	*			*		*	*	*		*	*	*
repository entry	*			*	*	*		*		*	*	

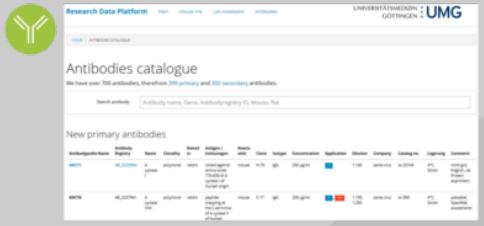
Ulrich Schwardmann

CRC1002: Modulatory Units in Heart Failure

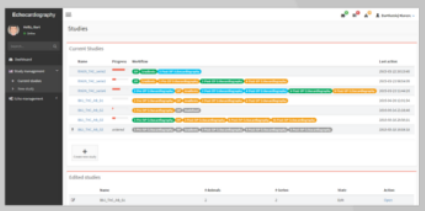
Collaborative Research Centre funded by: *Deutsche Forschungsgemeinschaft*

Ulrich Schwardmann

- template development
- based on existing standards



- Adaptation to PSI-MI/MIAPAR
- Antibody Registry/Antibodypedia



- automatic EPIC PIDs for datasets
- links to Biological databases

- PubMed standard and Indexing Number (PMID)
- PubMed Central referencing number (PMCID)
- Journal Digital Object Identifier (DOI)

- Unique name (EPIC PID)/URL

ePIC
Persistent Identifiers for eResearch

What is the general problem?

Typing is used for preparing the data processing of a DO

- well known: mime types
- others applications could cover:
  - availability and access conditions,
  - provenance,
  - processing preconditions or
  - data integrity.

Mayor goal is to make DOs machine actionable depending on their types

Examples of types in certain contexts

	checksum	checksum fixed	metadata pointer	author	basic Dublin Core	mime type & version	creation date	expiration date	embargoes	versioning	templates & services	fine data granularity
edition	*		*		*					*	*	
object of edition	*			*			*		*		*	*
electronic lab notebook	*		*		*					*	*	
notebook entry		*	*	*					*			*
experiment output	*			*		*	*	*	*		*	*
image library			*		*							
image	*			*		*	*	*		*	*	*
repository entry	*			*	*	*		*		*	*	

Ulrich Schwardmann

What is the proposed solution?

- RDA proposed a general framework for data typing:
  - the registration of data types and
  - the concept of information types used together with PIDs
- RDA working groups (DTR,PIT) provided
  - An interface description for the access to information types (PIT)
  - and CORDRA as a data type registry software is available (DTR)
- some registries for data types are already implemented,
  - where each data types is represented by a PID

What was still missing?

Type conformity and verification

- Type information has to be particularly reliable, because
  - the functionality of the data services is dependent on a correct preprocessing.
  - this kind of metadata is interpreted by automated services
  - therefore it is necessary to avoid each precondition of human interpretation.
- **Schemas** need to be part of the information type description
  - and have to be defined in a clearly determined and reproducible way.
  - only an automatic process can guarantee this.

Hierarchies of Types

- Information types are often refering to simpler types:
  - a geolocation contains longitude, given in sexagesimal or decimal form.
  - citation information contains an author, perhaps given by an ID in a certain ID system
- they are eventually based on very basic types
  - determined by regular expressions or other restrictions

Suggestion:

- Information types are recursively built out of a finite combination of
  - information types and
  - such basic information types
- possible advantage: reuse of schemas

Schemas, how to generate and maintain them?

Ulrich Schwardmann

- The common practice:
  - describe metadata generally and also more formally
  - manually build a schema according to that description
  - exploitation of the hierarchy is only a part of the manual process
  - consequences:
    - inconsistencies between the metadata description and the schema
    - a lot of manual work for schema derivation and adaption

Schemas, how to generate and maintain them?

Ulrich Schwardmann

- automatic schema derivation (in JSON):
  - exactly describe the information type dependencies in the type description in the DTR
    - enable as much flexibility in the JSON framework as possible
  - exploit the hierarchy in an automated process
  - basic information types have a (simple) schema as leaves in the graph
- storage of the schema in the type description of the DTR

Validation of instances

- all necessary information for validation can be found in the PID and the DTR
  - and can be retrieved via the REST API of the Handle System or the DTR

Suggestion:

- provide client function calls like backslash`typeIsValid(PID,typeID)` backslash`backslash`
- provide client libraries even for all kinds of queries about PIDs and types in the handle system or DTR

Ulrich Schwardmann

Ulrich Schwardmann

Ulrich Schwardmann

Ulrich
Schwardmann