Use Cases

Data life cycle management in Earth system modelling deals with large numbers and volumes of data and evolving dissemination, annotation and archival processes. E-Science infrastructures can benefit from early assignment of PIDs much prior to archival and formal publication stages, causing a rise in the number of PIDs assigned and complexity of attached information.

When new data is available, an e-Infrastructure must make old data obsolete without breaking user expectations or external workflows.

Understanding PIDs

How should the concept of persistent identification be understood in view of data that may not persist? How to retain the original purpose of persistent identification? What are essential requirements in PID-related processes? How do we organize large numbers of interrelated PIDs, deal with object disappearance and persistent context information? What is the role of PID infrastructures in such scenarios?

Actionable PID Collections

The base concept of abstract data types is fundamental to structured programming and can be transferred to PID scenarios. Actionable PID collections structure once isolated identifiers and enable complex operations on them beyond the conventional lookup.

Versioned data can form a collection with its own identity and agents can reliably navigate to both old and new data.

Layering

A layered architecture with distinct roles and interfaces can enable modularity, interoperability and extensibility. Distinct layers encapsulate the fundamental concepts on persistency, typing and structures such as collections and typed links.

A versioned dataset is described by structural elements which rely on typing. Persistency of the structure is retained beyond the data’s life time.

Usage Policies

Daily operations in scientific data management exhibit patterns of behaviour and action that can be classified. If PIDs are involved, these patterns must be clearly defined so that interoperability and long-term interpretability are ensured. What are essential requirements and emerging principles to uphold in such processes, and how do they apply to the e-Infrastructures in use?

Solid versioning calls for a reliable process that is best established as a central principle at all entry points to object access and modification.