Dynamic Data Life-Cycle in EUDAT

Daan Broeder
TLA/MPI for Psycholinguistics

EUDAT

Panta Rhei

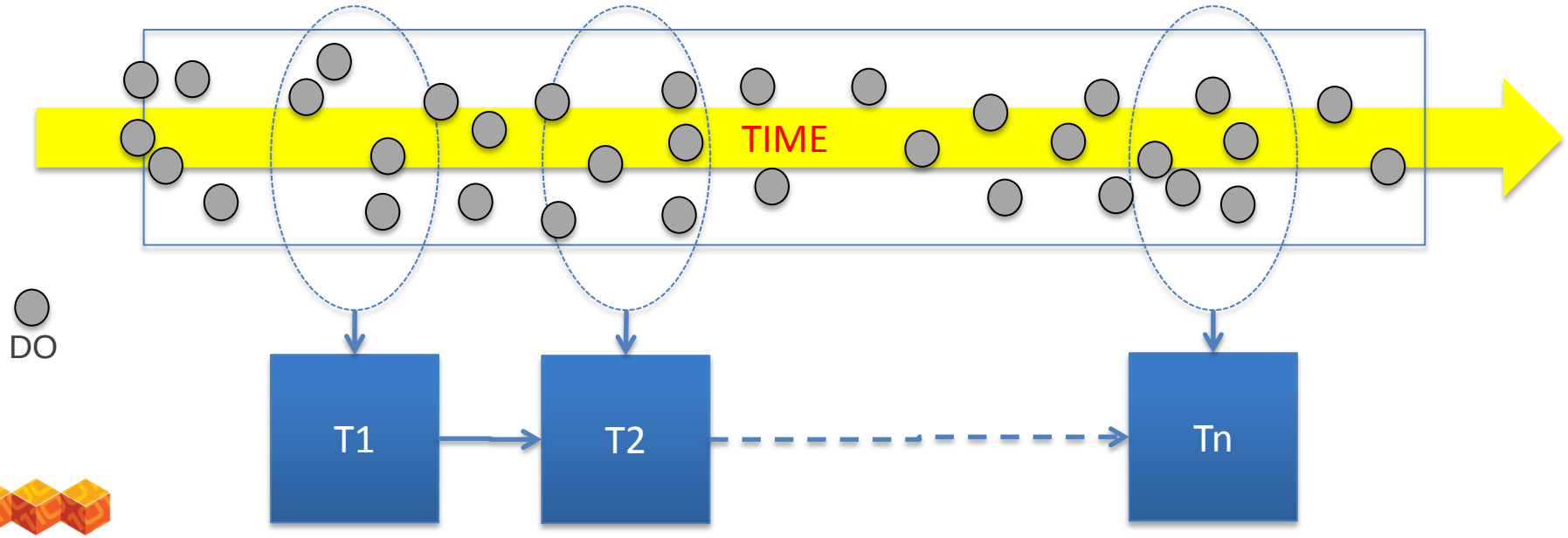
- Isn’t all data dynamic? Maybe… all data is static
- But data management concepts, procedures and efforts have been mostly focusing on the static parts, finding stable references, archiving, etc.
- Dynamic Data is usually accommodated as managing a series of versions
  - Although that seems sufficient
  - It is not always the most efficient

‘Converging’ Dynamic Data

- Use cases:
  - (Multi-dimensional) signal with gaps that get filled in an unpredictable manner
  - Incomplete collection of crowd sourced tasks that …
  - Incomplete linguistic corpus of recordings that …
  - Incomplete collection of survey results that …
- In all cases it is needed to access, process, describe, cite the (sometimes) still incomplete data

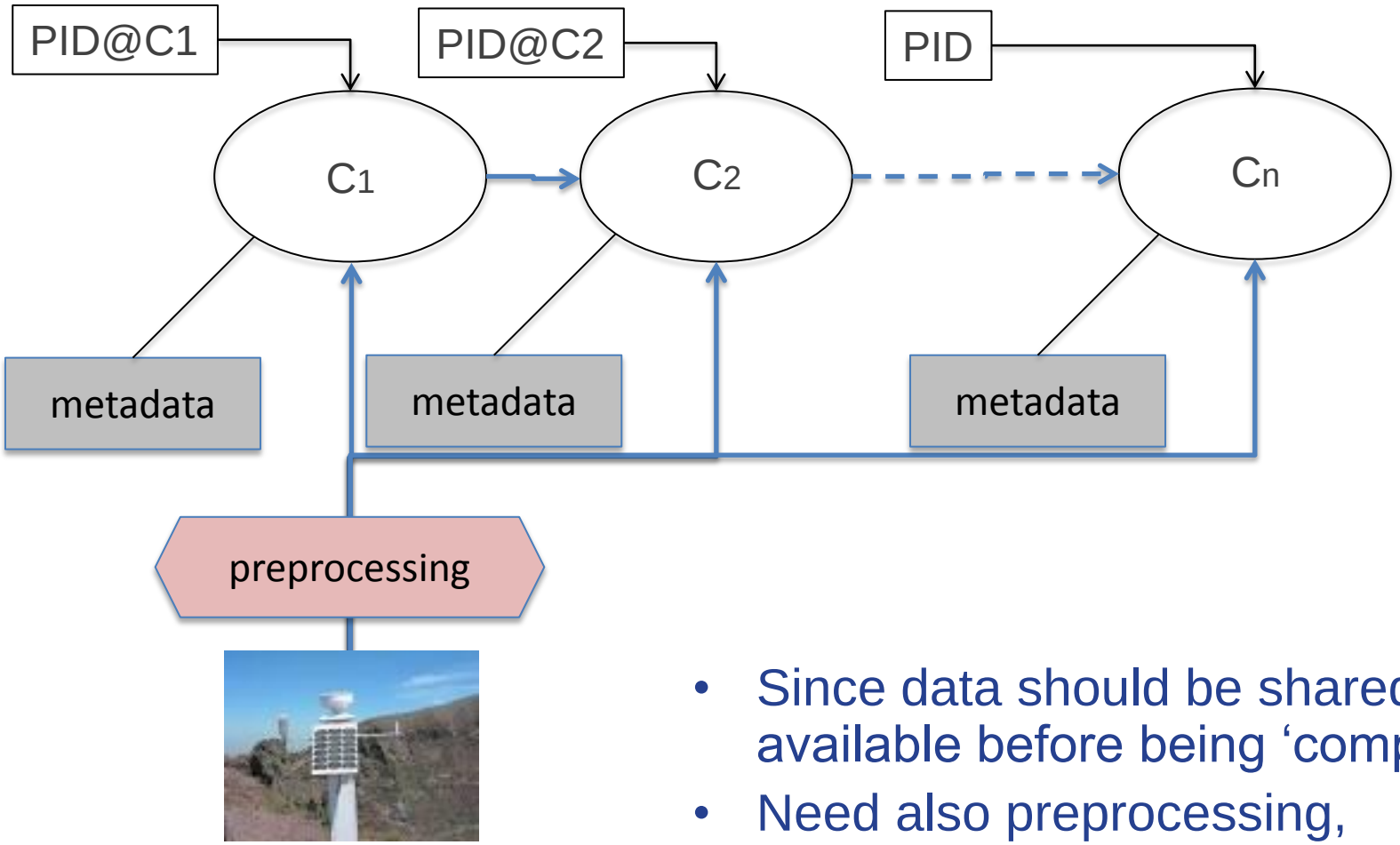
Dynamic Data and Versioning

Describe Dynamic Data as a series of versions; snapshots of a changing DO collection, sampled equidistant or non-equidistant.



EUDAT

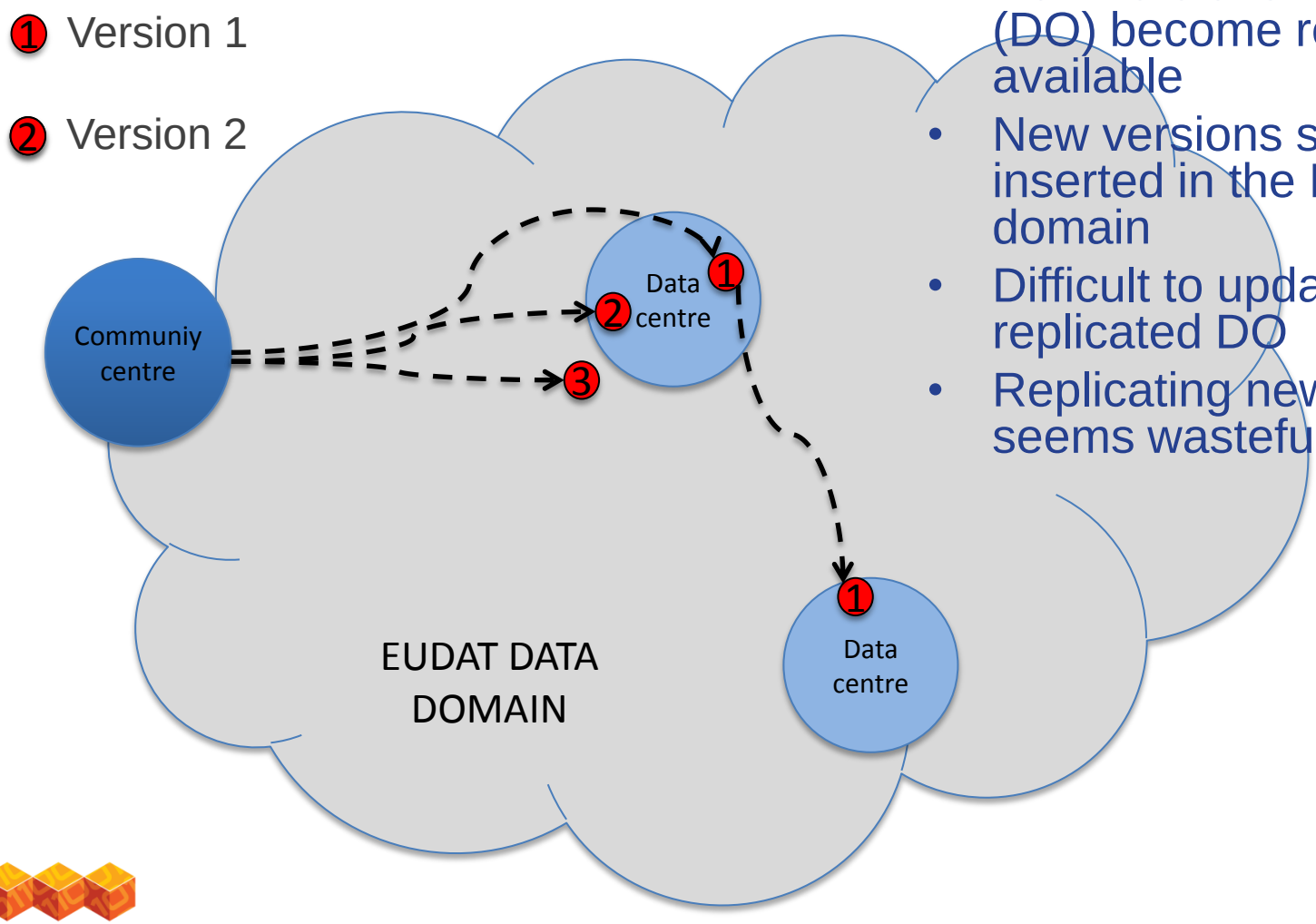
Dynamic Data Life Cycle



- Since data should be shared and available before being ‘complete’
- Need also preprocessing, metadata, registration, …

EUDAT

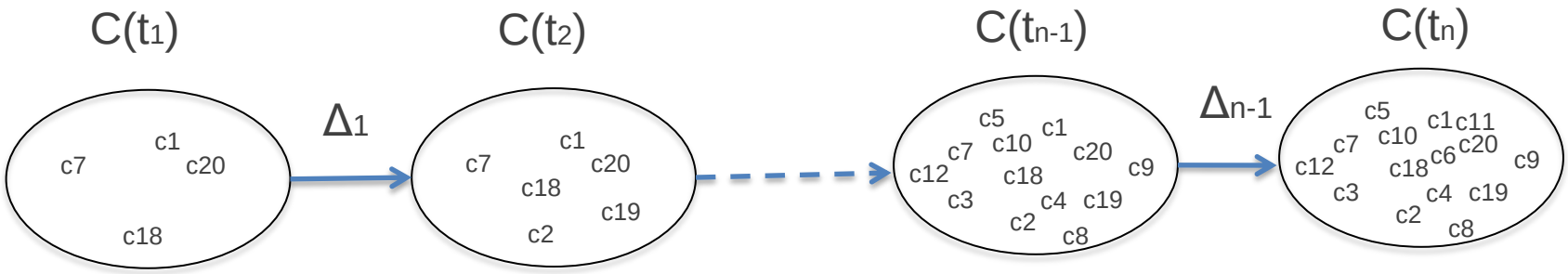
The problem with Dynamic Data and Replication



- New versions Digital Object (DO) become regularly available
- New versions should be inserted in the EUDAT data domain
- Difficult to update an already replicated DO
- Replicating new versions seems wasteful

Keeping track of Dynamic Data

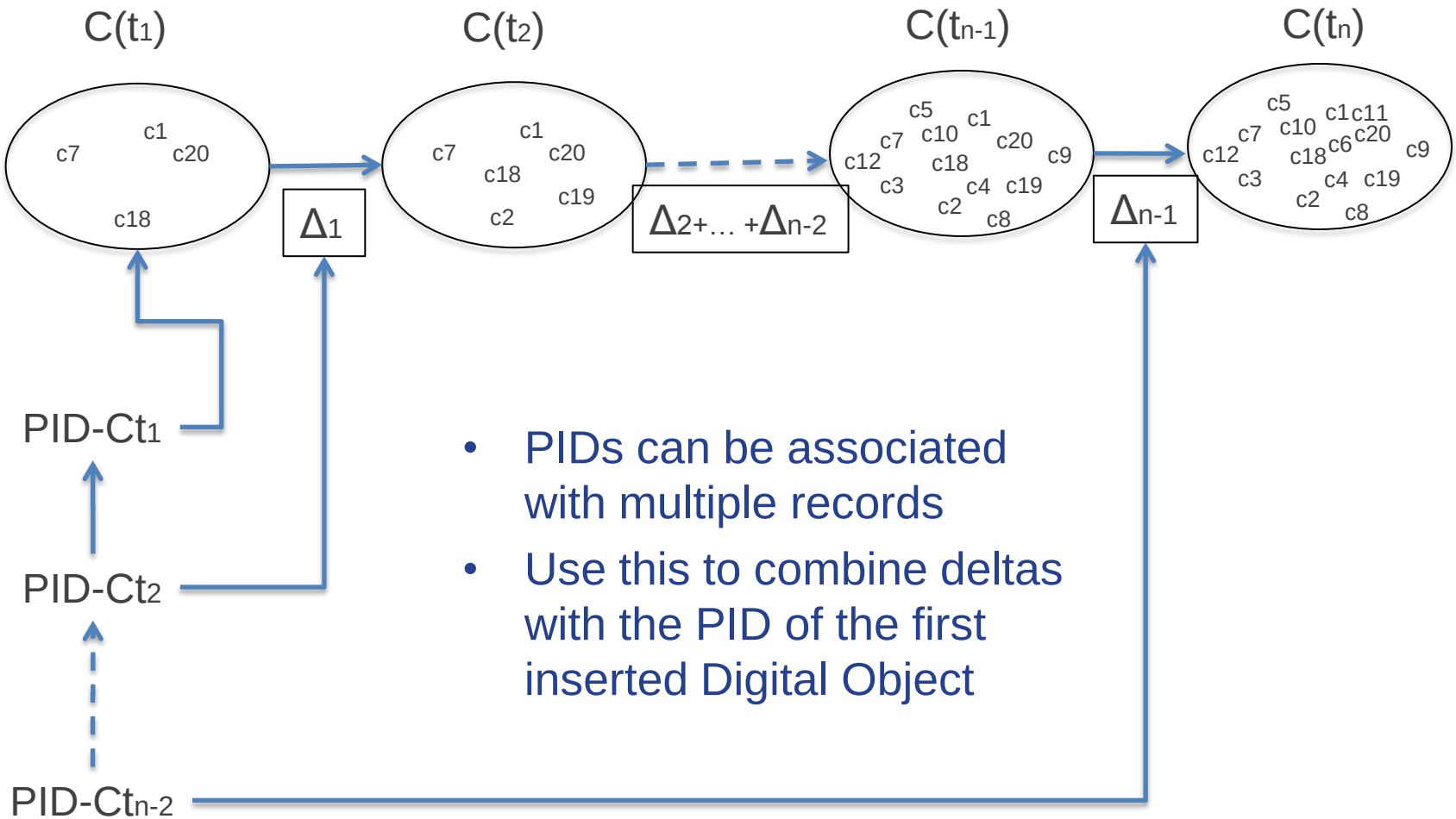
- Keep track of the relations between the Dynamic Data fragments or elements



- Depending on your data management administration use:
  - In the PID registration process
  - In the metadata and provenance data
  - Virtual collection registries

EUDAT

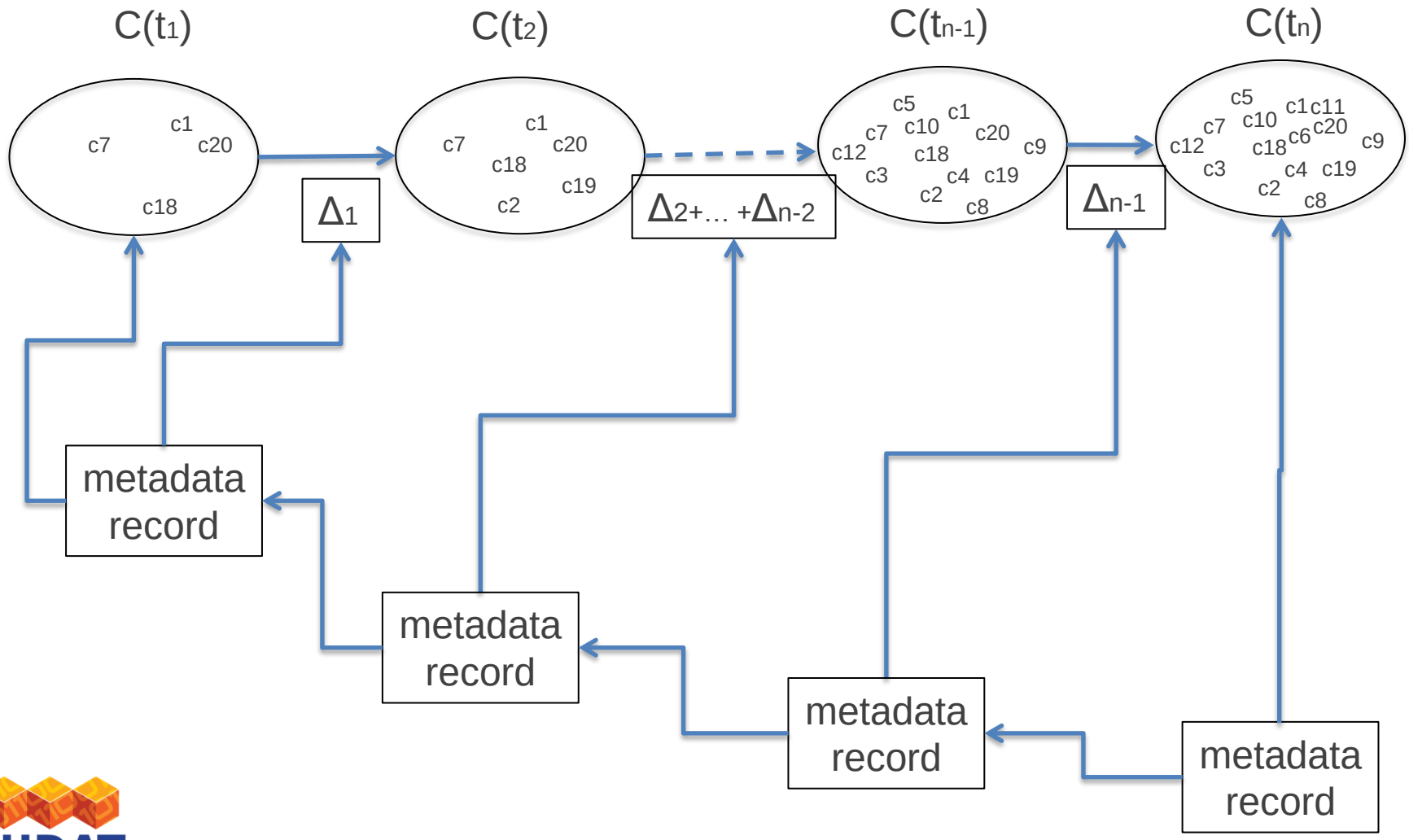
PIDs



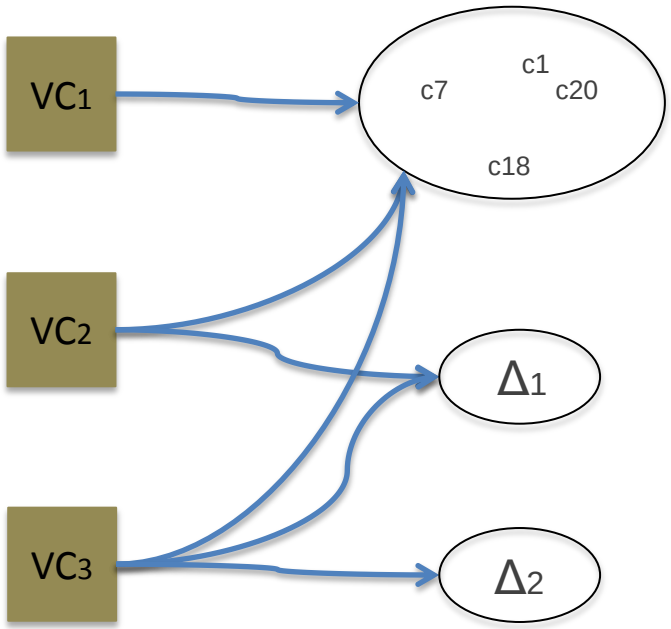
- PIDs can be associated with multiple records
- Use this to combine deltas with the PID of the first inserted Digital Object

EUDAT

Using metadata



Use of Virtual Collections



- Introduce new objects or a registry for defining $C(t_n)$ using virtual collections VC_1, VC_2, …, VC_n
- $C(t_3) = C(t1) + \Delta_1 + \Delta_2$
- Looks more efficient

- However a PID for the DD version needs to refer to its VC

EUDAT

So what to do?

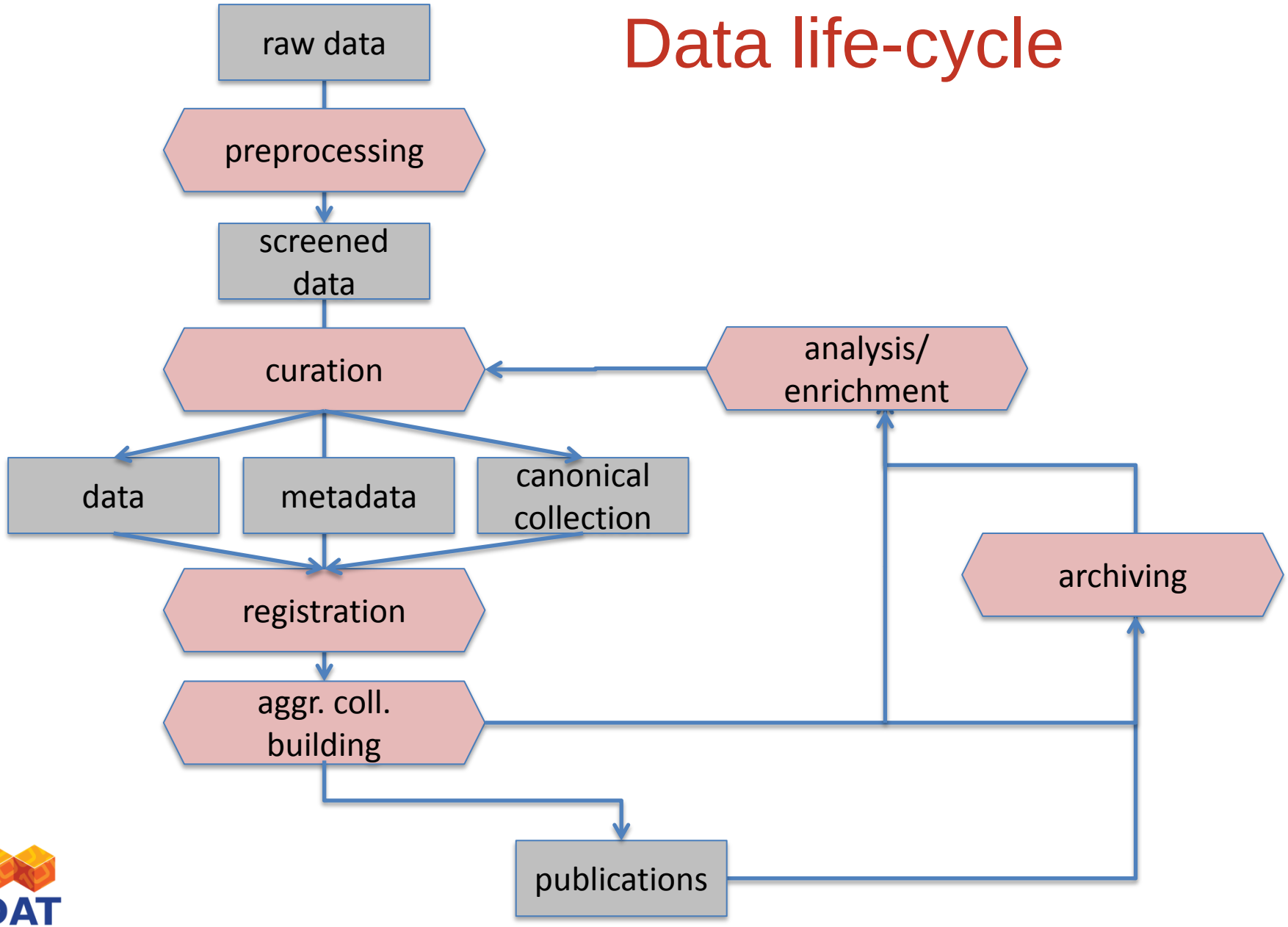
- Dynamic Data can be adequately viewed as a series of versions.
  - How the versions should be encoded as collections only or as collections + deltas depends also on other priorities
- For PIDs we want the PID to refer to the last stable version, if there is one, if not just pick one
- For metadata, new versions require new metadata
- With big data we want efficient modeling for intermediate versions using deltas.

EUDAT

THANK YOU

EUDAT

Data life-cycle



EUDAT