Issues of Data Organisation

Peter Wittenburg
Max Planck Compute & Data Facility

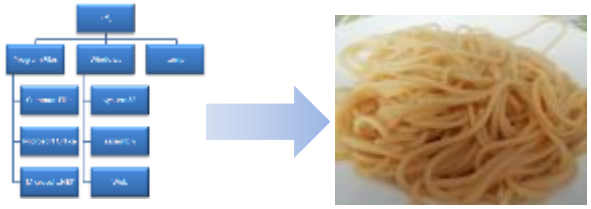
Dynamic data world

- Sciences/Societies are Changing & Data is the Oil.
- Are in an Exploratory Phase & Let 1000 Flowers Blossom.
- Consolidation Phase is needed & Reduction of solution Space.

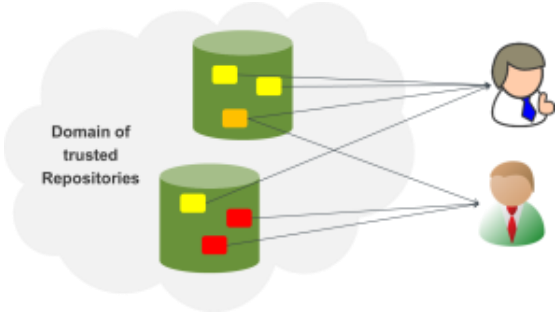
- Can Harmonization of Data Organization Help?

Basic aspects when talking about data

- Volume, Variety, Velocity, etc.
- From simple to complex structures (it‘s the multiple relations)



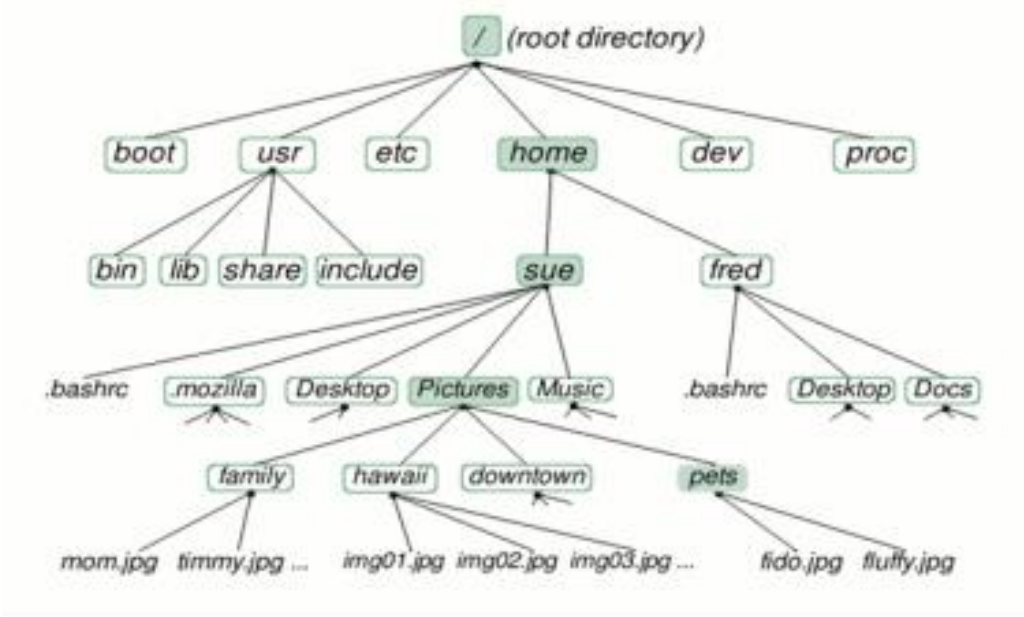
- Re-use/re-combination of data in different contexts by unknown experts



- Trust and Acknowledgement Problem

Need to estimate usage and requirements in 10 years! Building infrastructures takes time! Imagine agents (humans/machines) using profiles to find and correlate useful data!

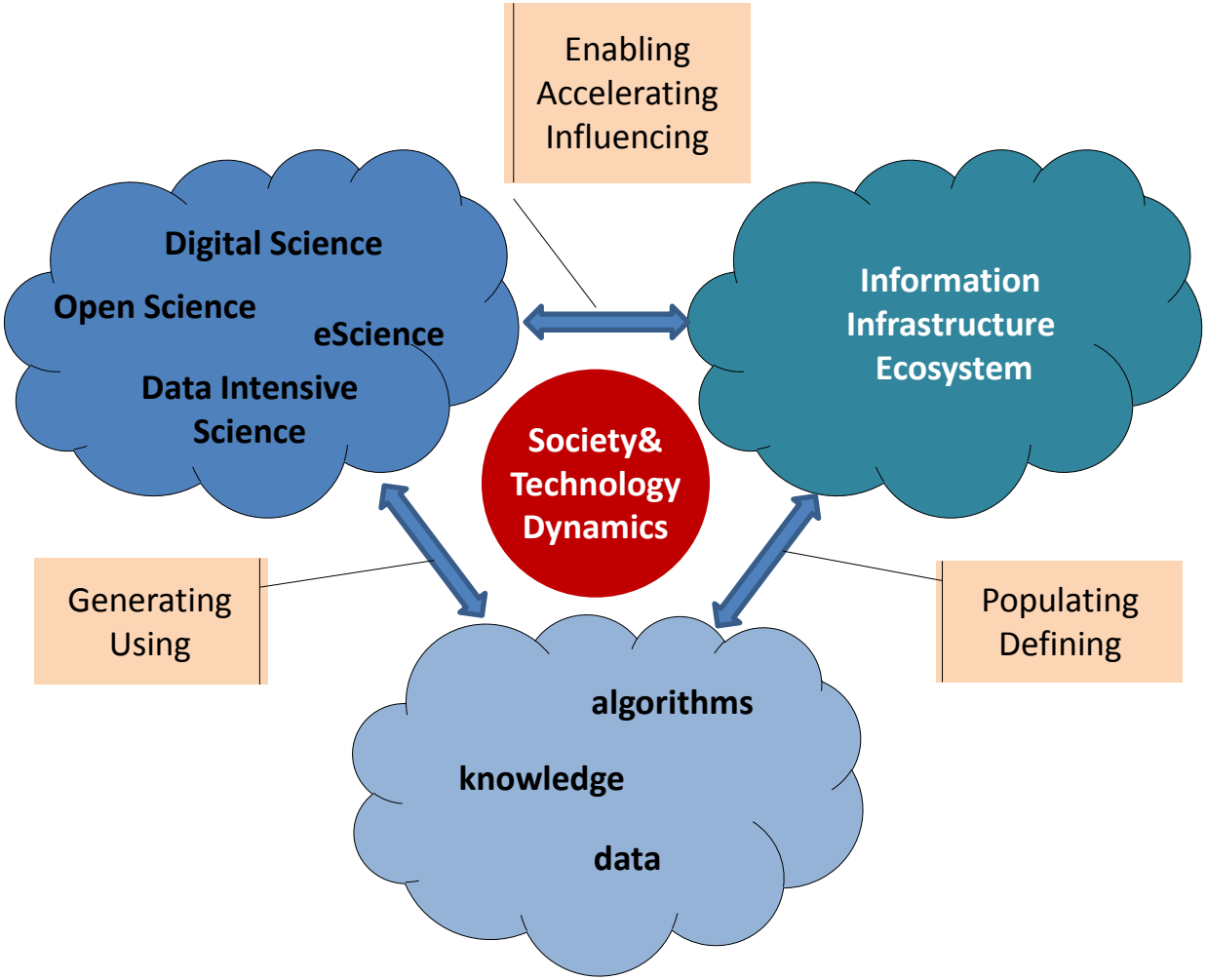
Old methods don‘t work any longer



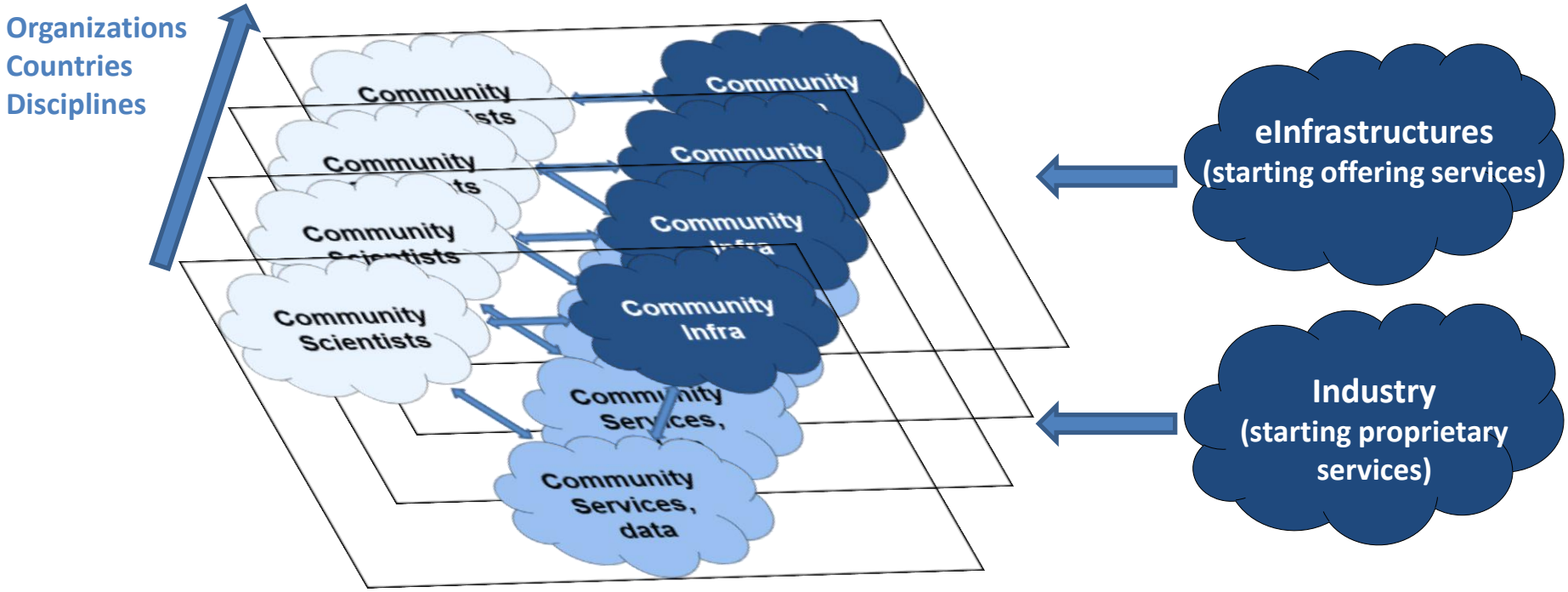
- too many files
- context can‘t be stored in names
- relations can‘t be stored in directory paths

- spreadsheets will be forgotton after x months
- take care: databases encapsulate and many don‘t have an XML export
- etc.

Need for infrastructures



But ending up in silos & fragmentation



- **ESFRI: much awareness raising in Europe, lots of young people trained, much testing of variety of approaches, identifying gaps in service landscape, etc.**
- **eInfra: starting to change towards service orientation, need more stable services, need clarification of costs**

Solution Space is huge – costs are huge!

Hampering investments!

Results from interviews

- ~120 Interviews/Interactions
- 3 Workshops with Leading Scientists (RDA EU, US)

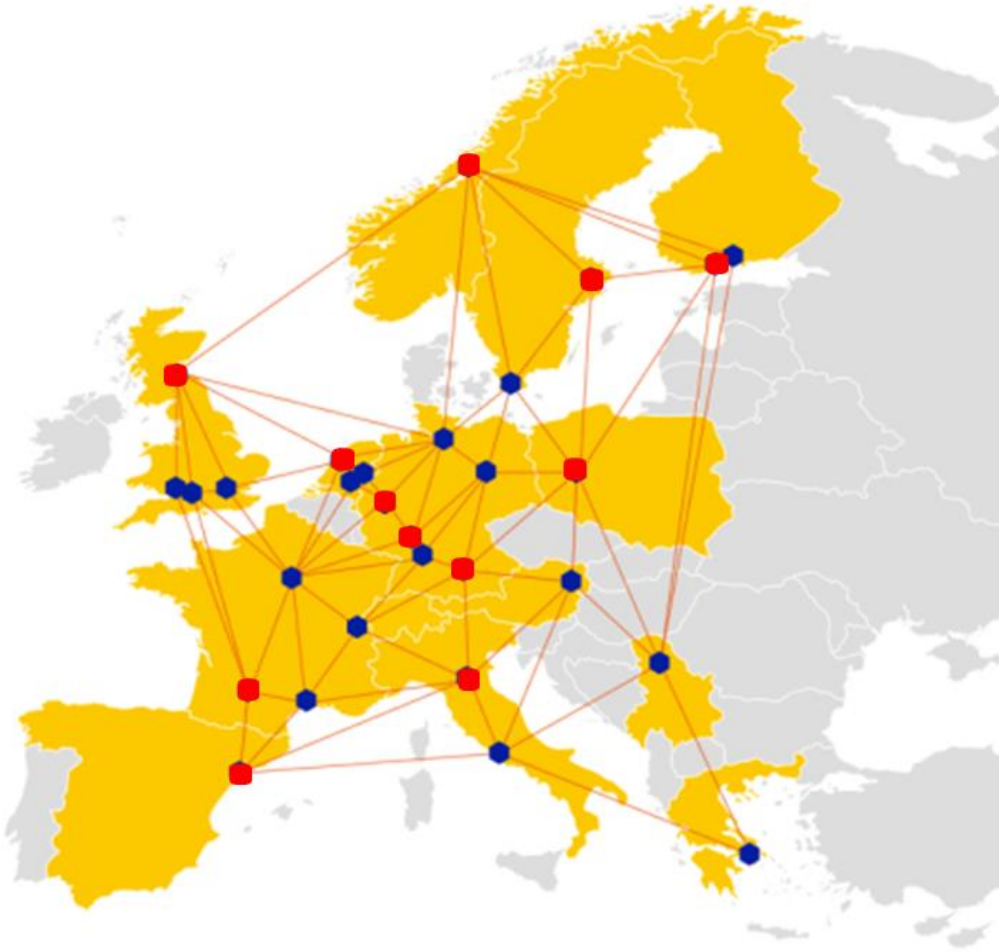
- still many obstacles to Open Data
- lack mechanisms of trust and acknowledgements
- trend towards trustful centres – still lacking offers for all
- there are positive project examples etc. but ...
  - too much manual work or via ad hoc scripts
  - hardly usage of automated workflows and lack of reproducibility
  - DM and DP not efficient and too expensive
    (Biologist for 75% of his time data manager)
  - federating data incl. virtual information much too expensive

Results from interviews

- pressure towards DI research is high, but only some departments are fit for the challenges
- DI research is only available for Power-Institutes

- Senior Researchers: can't continue like this!
  - need to move towards proper data organization and automated workflows is evident
  - but changes now are risky:
    - lack of trained experts,
    - lack of guidelines and support

Federating data is too costly!



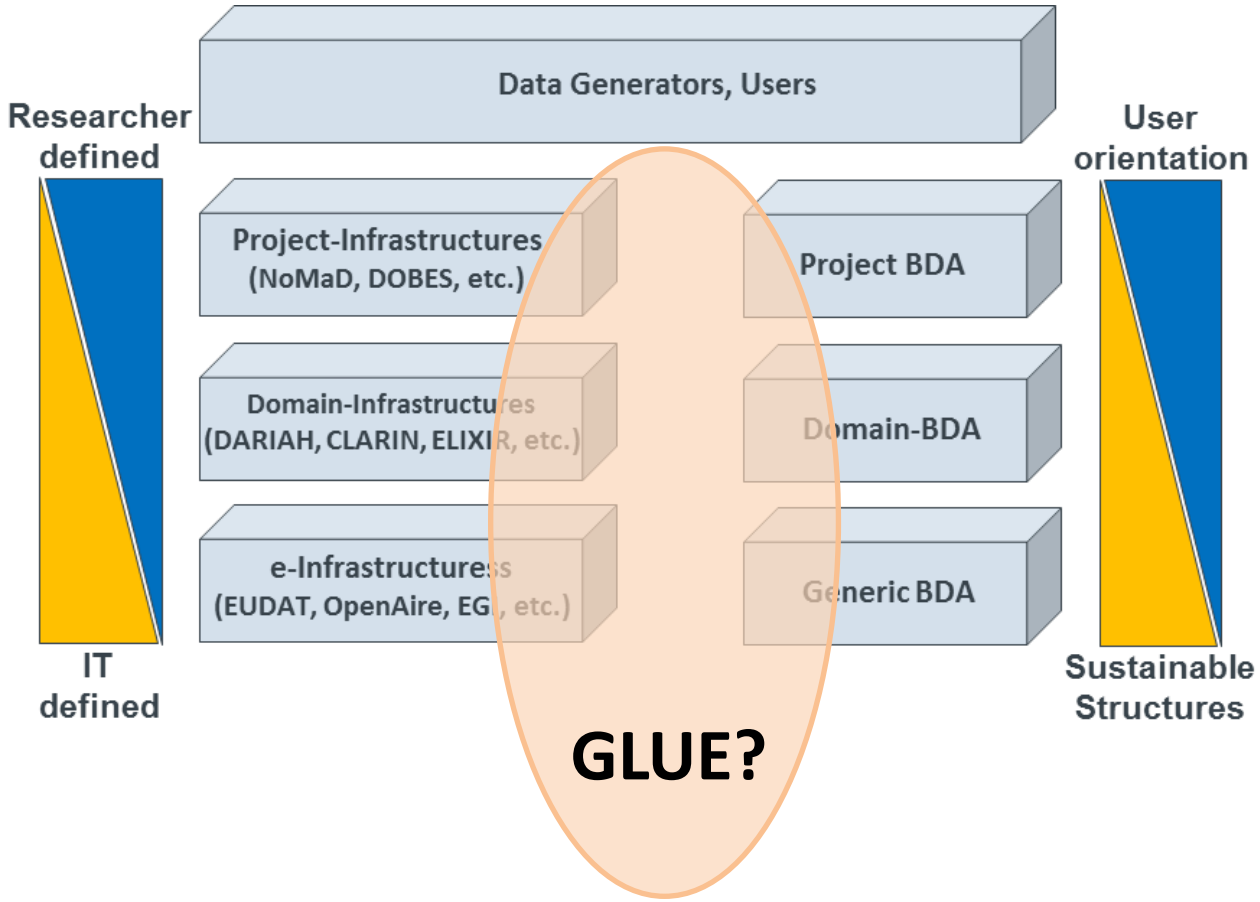
- Replicating data on physical level (files, clouds, databases) is doable
- But what about all sorts of metadata (keywords, annotations, relations, rights, etc.)
- Too complicated due to a lack of agreements

Also requests from funders

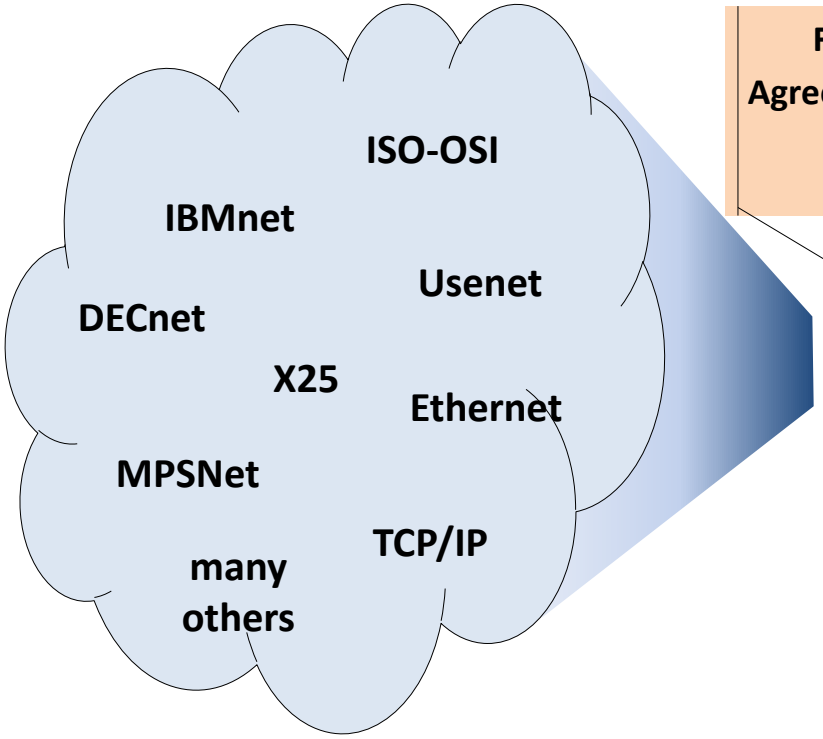
- G8/FAIR/FORCE11/etc. – data should be
  - searchable -> create useful metadata
  - accessible -> deposit in trusted repository and use PIDs
  - interpretable -> create metadata, register schema and semantics
  - re-usable -> provide contextual metadata
  - persistent -> provide persistent repositories

Need urgent actions to improve – but how?

What can we do????



There are similarities

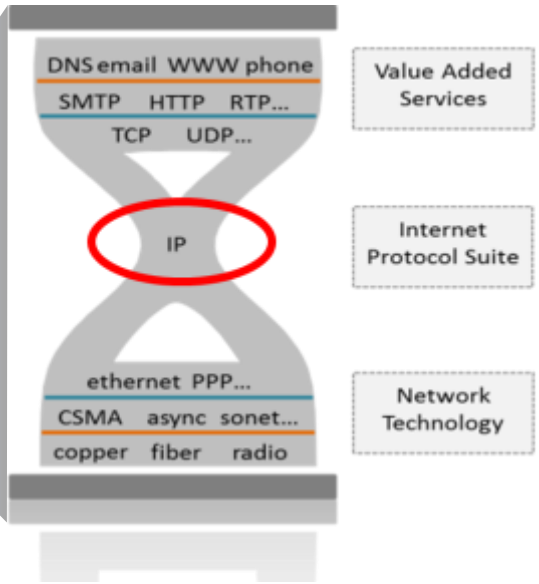


Finding the right level

Agreeing on one standard as a community process

IETF

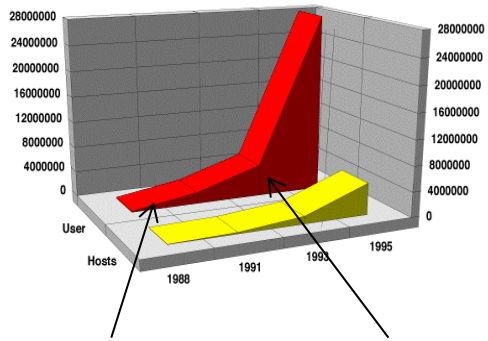
TCP/IP



Opened new area

- **New industries**
- **New businesses**
- **New jobs**

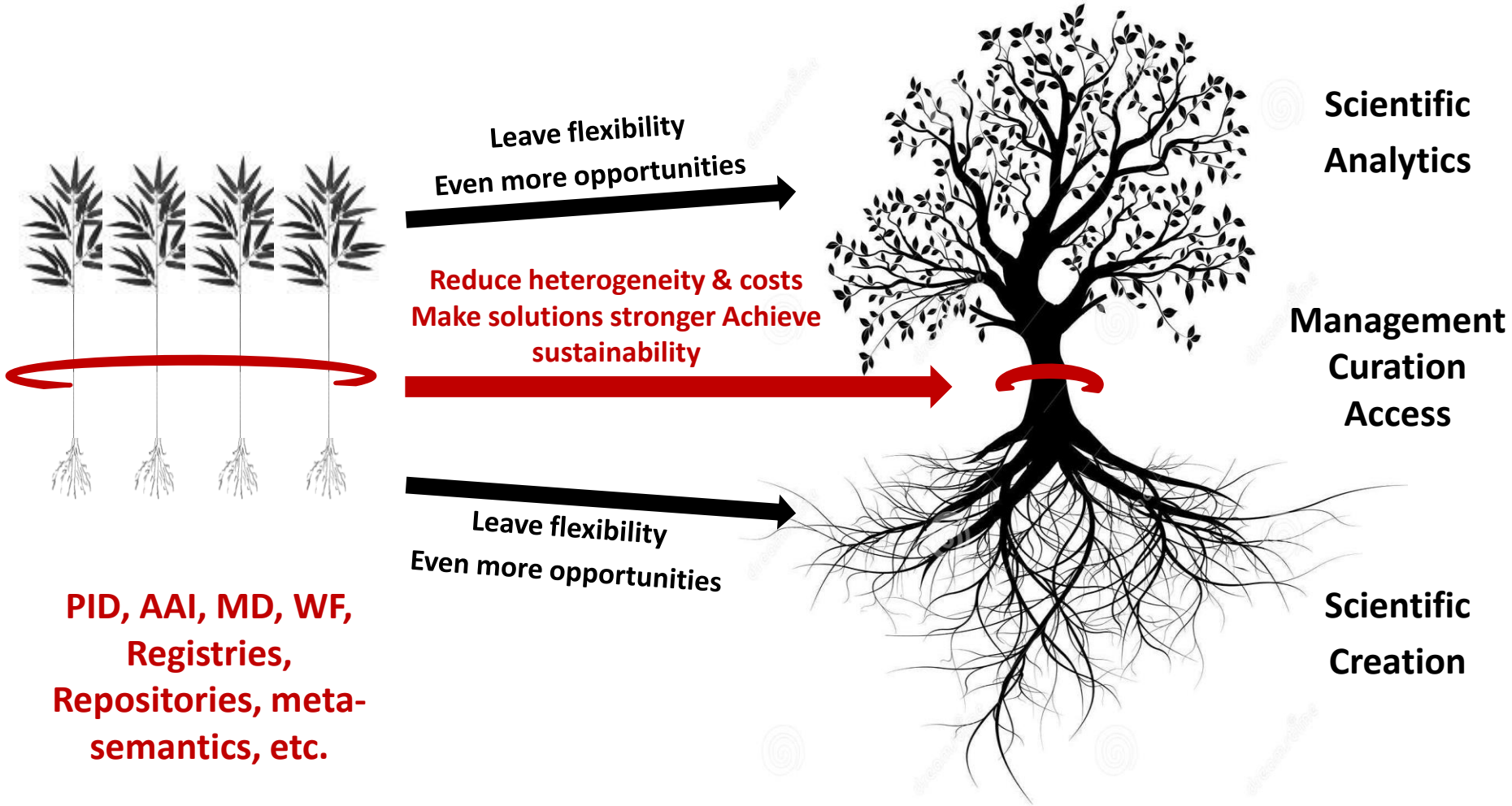
- **X*10 suggestions**
- **Hampering openness, innovation, investments, collaboration**
- **Little job creation**



Global TCP/IP

Mosaic Browser

Identifying COmmon COmponents



Consolidation Phase is needed to Reduce the solution space

What is data modelling in CS sense

- Widely influenced by the database community for many years

Conceptual Schema	Entities, Attributes, Relationships, Integrity Rules
Logical Schema	Tables, Colums, OO classes, XLS, etc.
Physical Schema	Storage, Channels, parallelisation, etc.

Nicely harmonized for rDBMS etc
But there is another part of data reality.
Often databases are just used as containers for fast operation.

Kinds of Digital Data

- So many different file-types with DD (no proper classification)
  - time series data
  - derived data
  - text data (whatever structure)
  - assertions (triples)
  - graphs (whatever structure)
  - „metadata“
  - programs (some could see them as data)
  - databases (as containers)
  - etc.
- some containers use proprietary formats, i.e. reading them is technology dependent

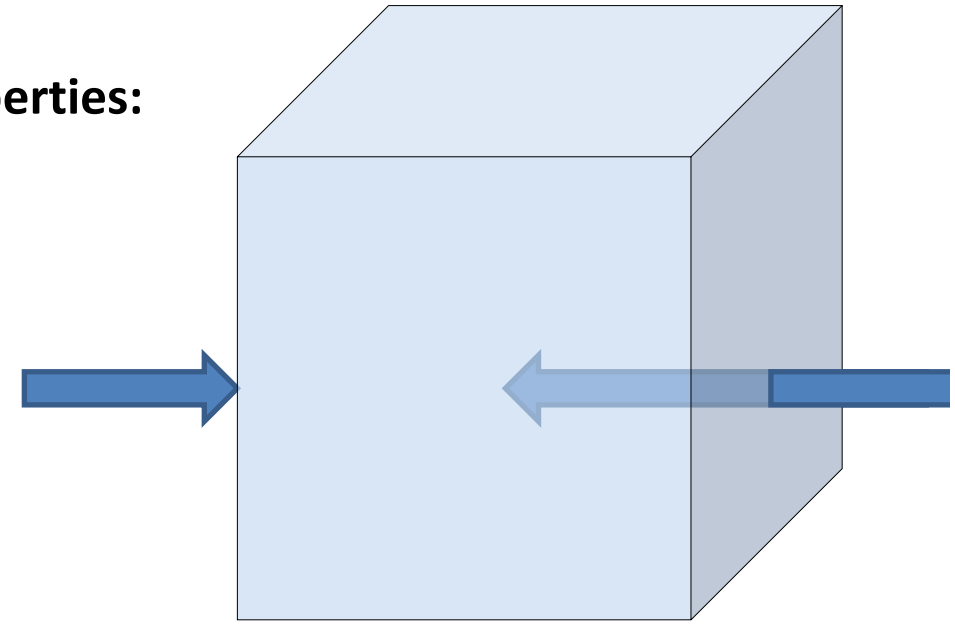
Our/my traditional view

reality: are still manipulating a lot at file level

External Properties:

- PID
- size
- owner
- location
- checksum
- type
- rights
- relations
- etc.

management
finding
accessing
curation



Internal Properties:

- syntax/format
- semantics
- encoding

interpretation
processing
curation

Will NoSQL DB change world?

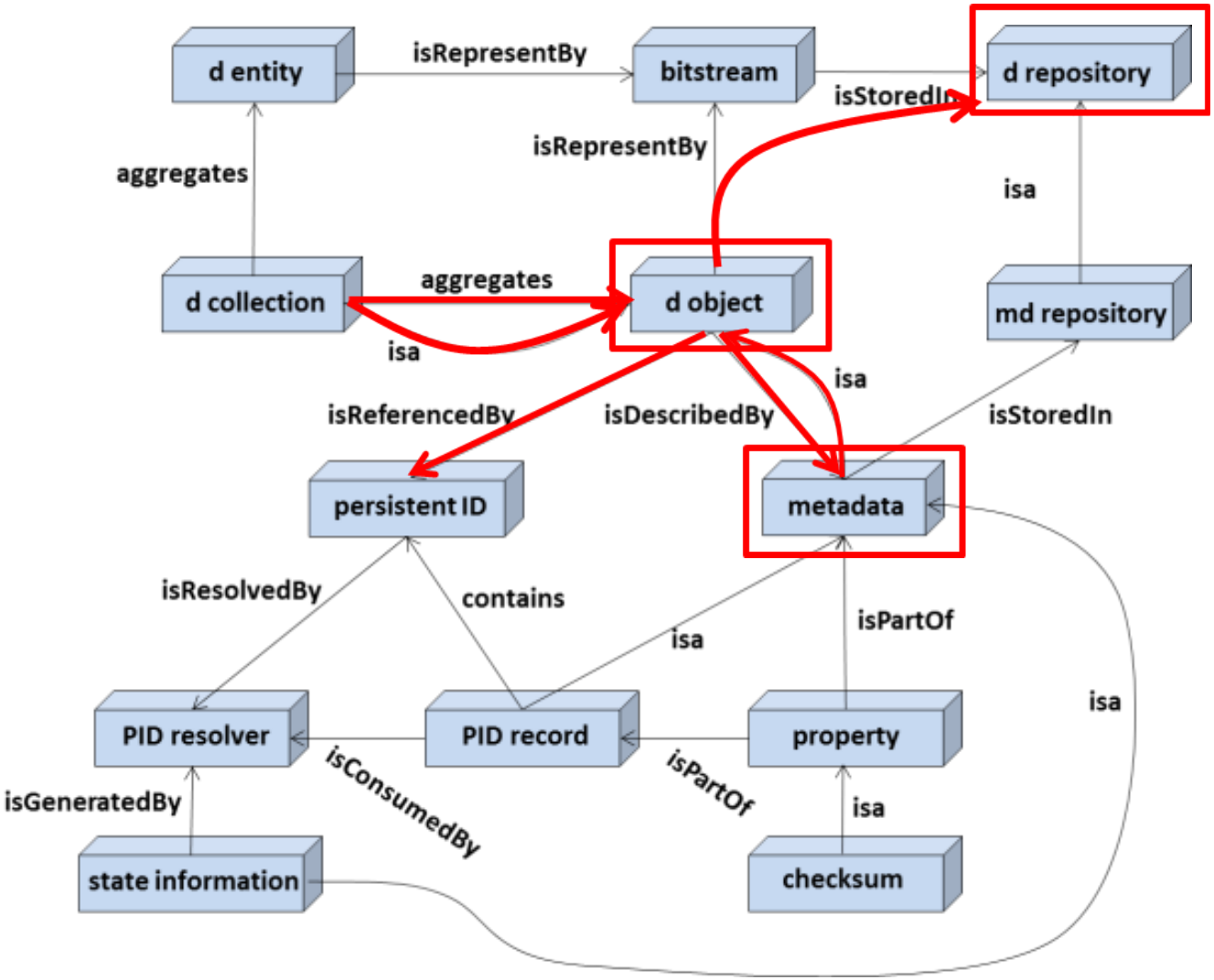
- this is all new technology to support big volumes, aggregates, clusters & distribution
  - key-value db
  - document db
  - column family db
  - graph db
  - array db (for multivariable time series data)
- many of the dbs are opensource, so we can access content independent of technology
- many open questions to me

Aspects making federating data hard

- easy to locate ONE instance of a file in a directory path or a cloud object
- but ...
  - where are the instances (copies)?
  - where is the metadata?
    (how to interpret content in case of headers)
  - where to find its PID if it has one?
  - where to find its access permissions?
  - where to find its relations (context beyond dir system)?
  - how to extract information from scripts?
  - etc.

Nothing has been agreed upon, everyone does it differently!

Notion of a Digital Object – DFT Model

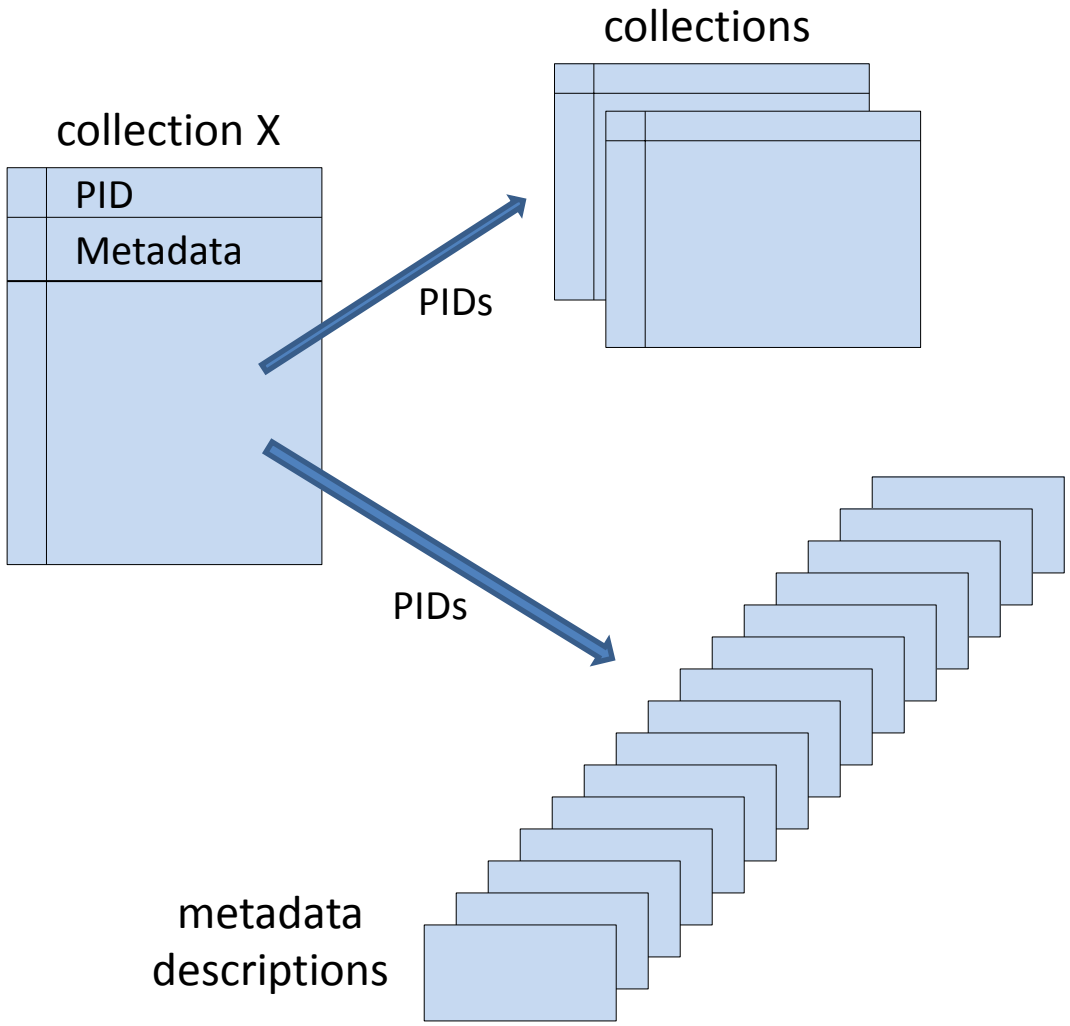


basic messages are congruent with FAIR principles

- need a method to identify digital content independent of its type (realization?), etc.
- otherwise no reference possible which would be fatal
- granularity is a domain decision

- Robert Kahn
- Janis Kallinikos
- Fedora Commons
- DOI Documents
- etc.

Nature of (virtual) collections



collection has

- a PID
- some metadata
- a huge amount of PIDs pointing to collections and metadata descriptions (and/or data PIDs)

Typical Access Pattern

Scientists, Data Curators,
End Users, Applications

Enabling
Technologies

Discovery

Access
(ref. resolution,
protocols, AAI)

Interpretation

Reuse

metadata domain

PID

requires to look into the object and thus use content and contextual information

Typical Management Pattern

Data Managers
Data Scientists

Enabling Technologies

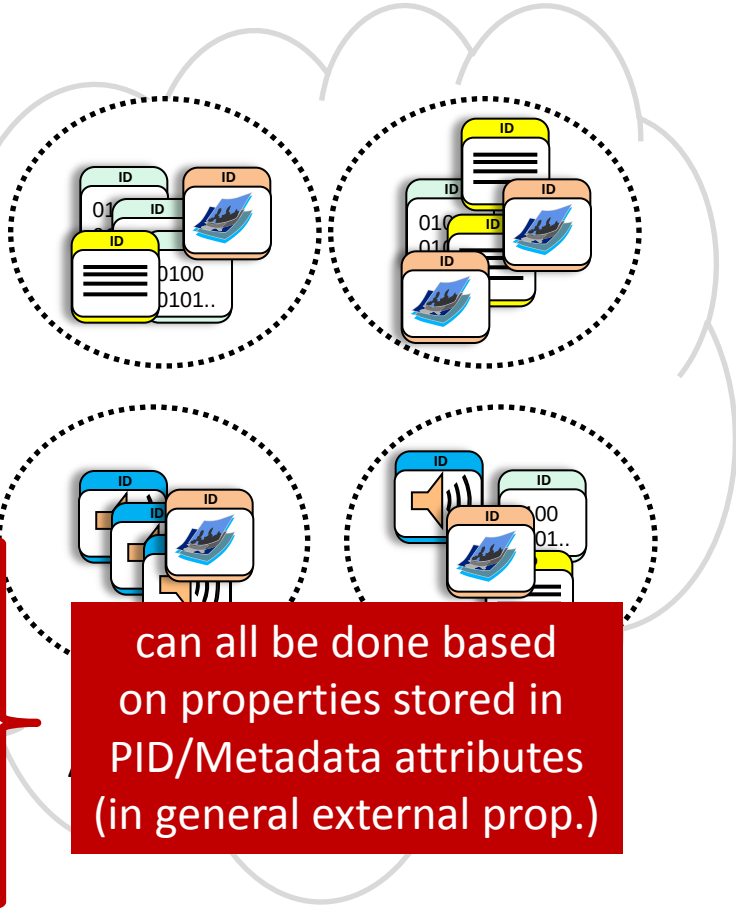
Collections + Properties

Access (ref. resolution, protocols, AAI)

formalized policies workflow engine

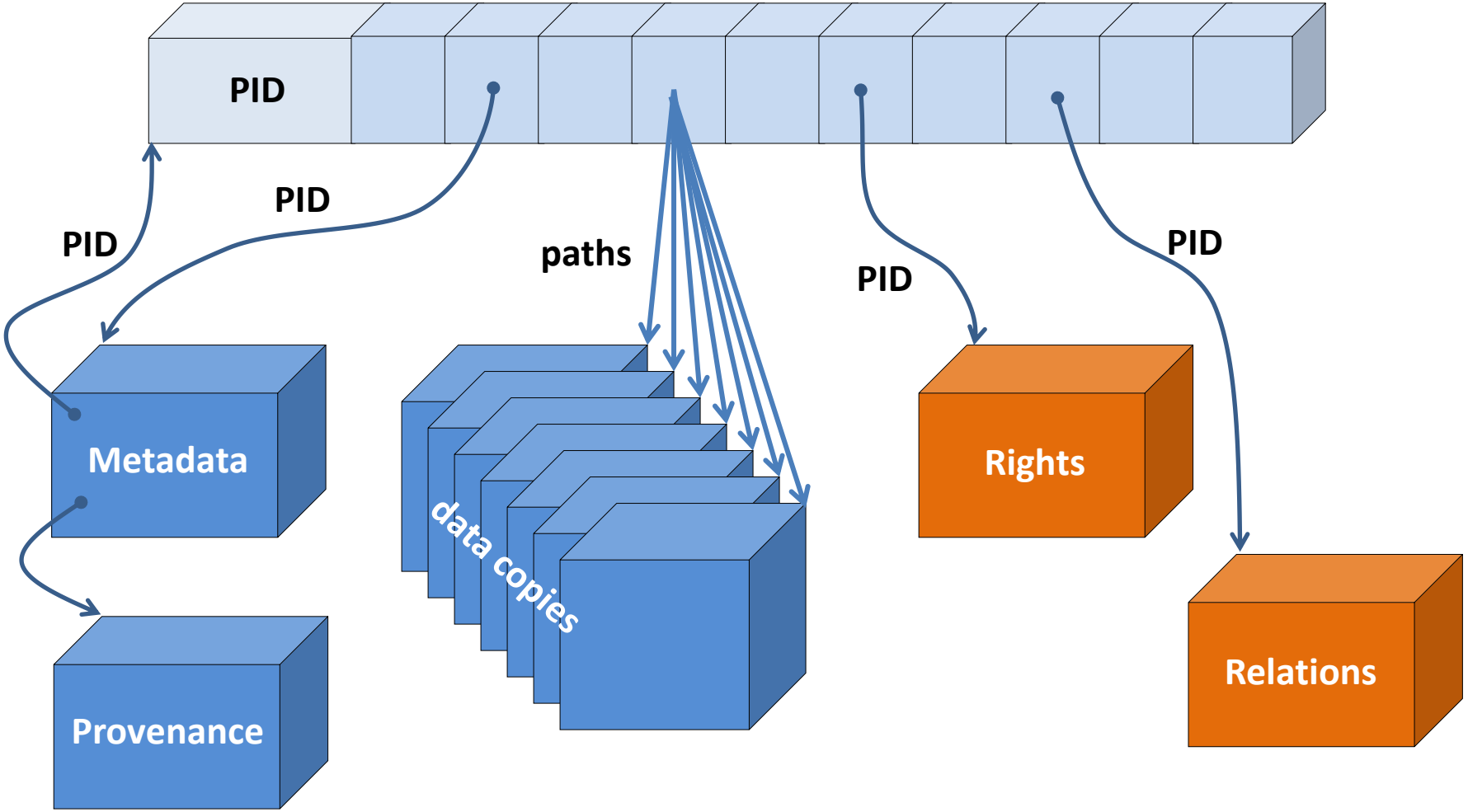
Assessment

PID



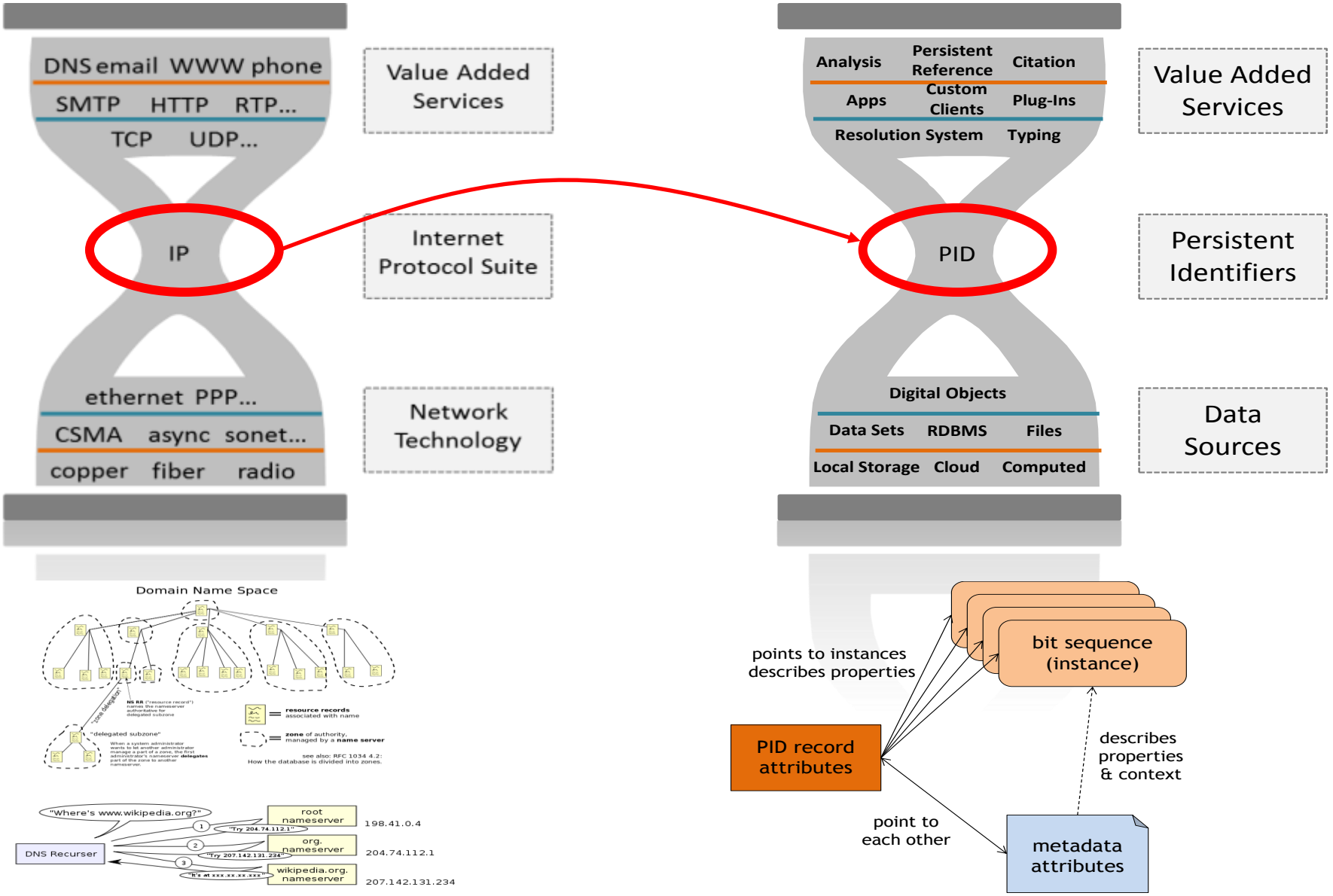
can all be done based on properties stored in PID/Metadata attributes (in general external prop.)

How to bind all this – PID centered model

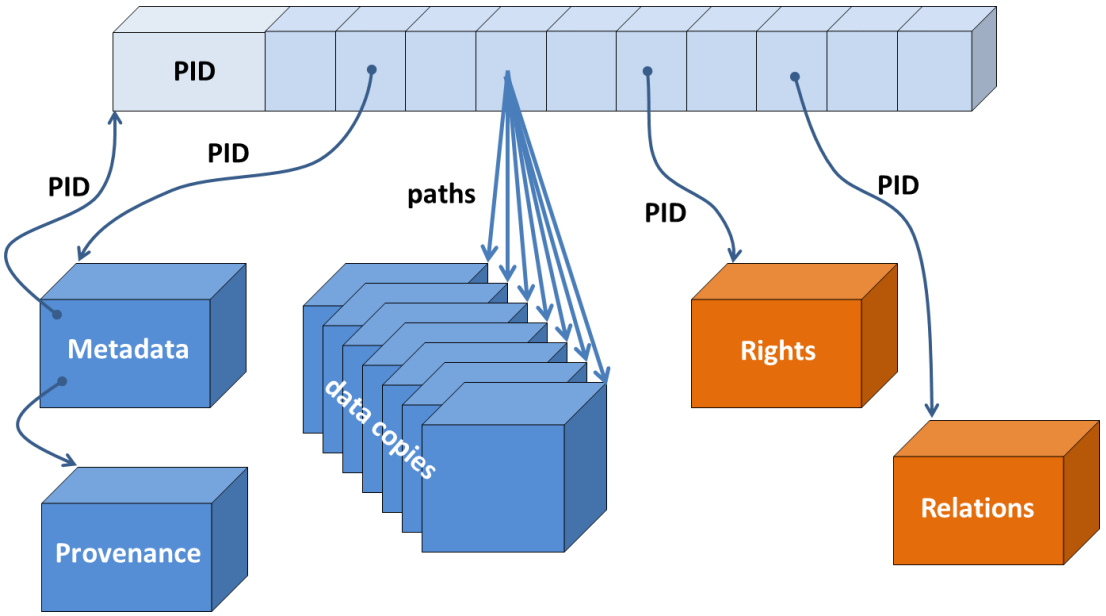


Can we rely on persistence and availability of PID Records?
Is this all performant enough?

Role of Persistent Identifier



Goodies of such a data organization



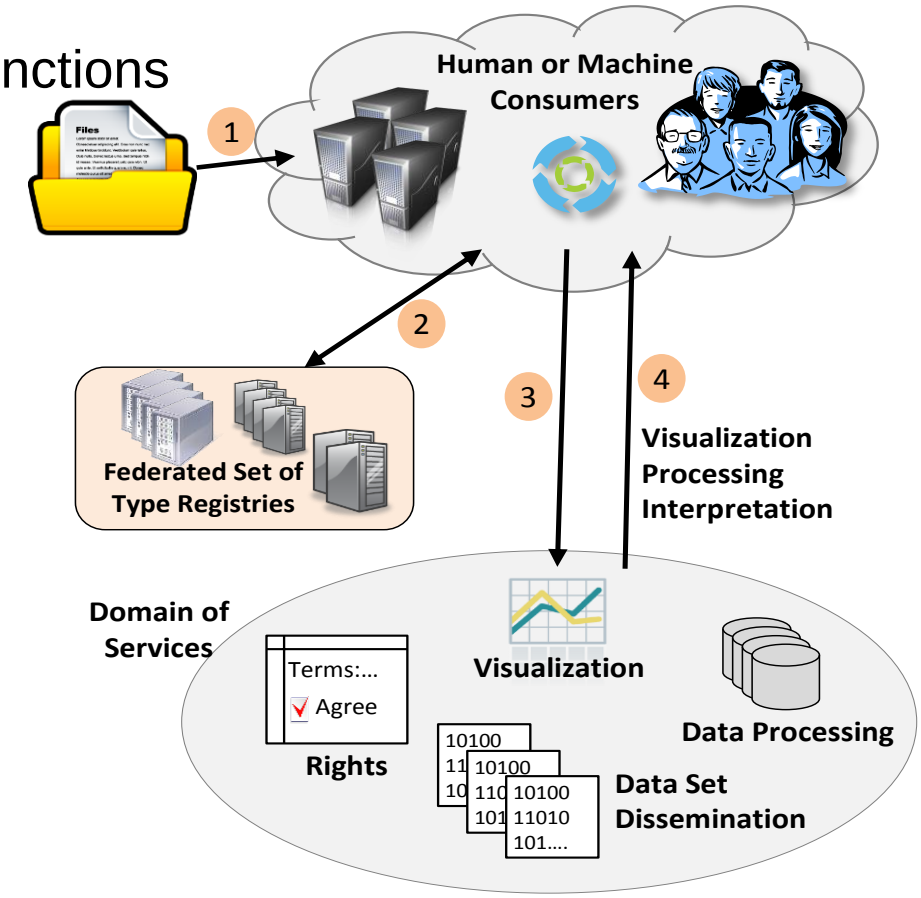
other points of interest:

- pointers to schemas
- checksum
- RoR flag
- etc.

- PID system is global
- just need the DO‘s PID to find all related information
- all is not embedded in ONE repository and thus independent of instances etc.
- two access ways are supported since metadata includes PID
- could be extended to versions and presentations
- in general a simple system
- but?

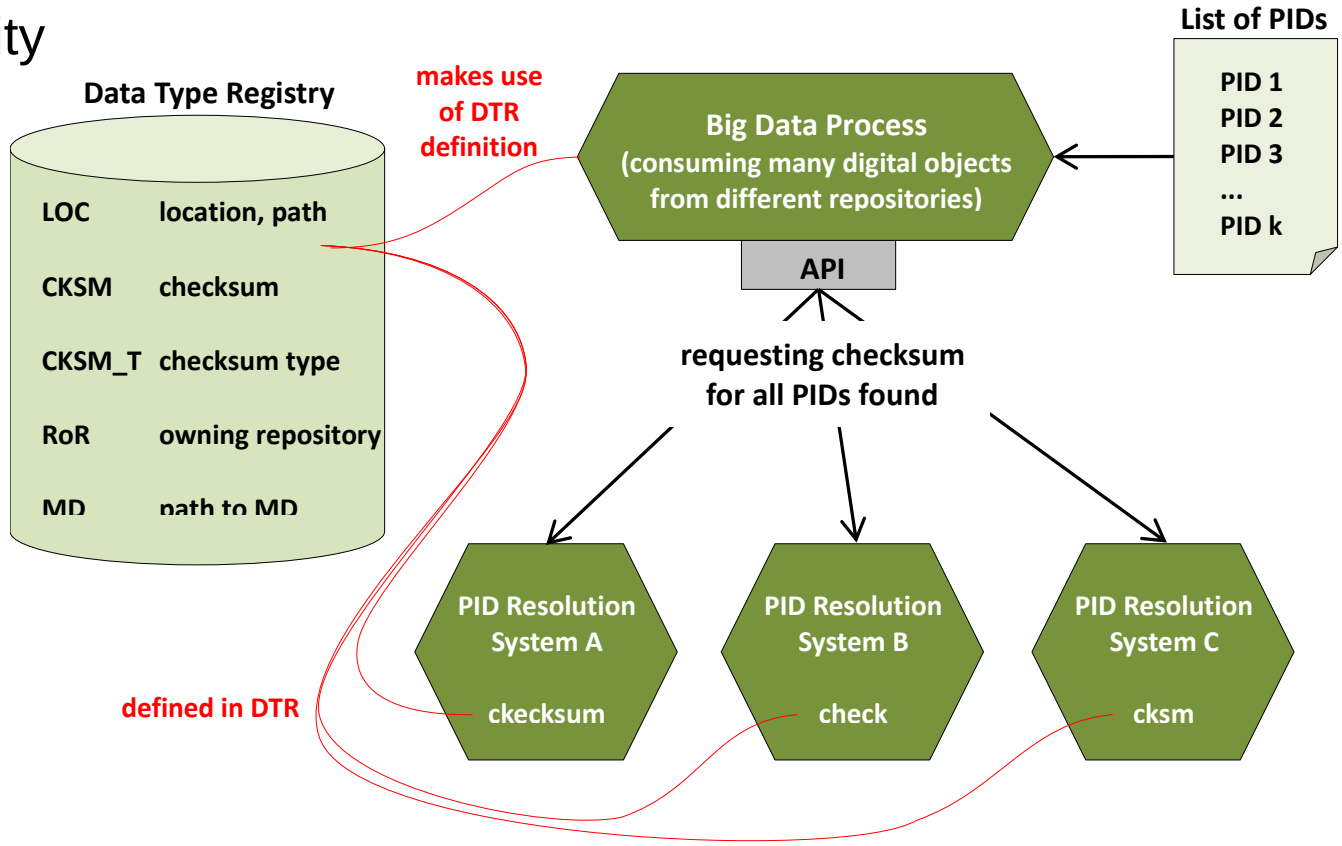
Requires information typing

- result: a registry for data types
- Linking structure/semantics with functions
- you get an unknown file,
  pull it on DTR and content is being visualized
- You find a tag and know how to interpret
- no free lunch: someone needs to register and define type
- PIT Demo already working with DTR
- Various sciences make use of it



Information typing allows generic API

- result: a generic API and a set of basic attributes
- a PID Record is like a Passport (Number, Photo, Exp-Date, etc.)
- if all PID Service-Provider agree on one API and talk the same language (registered terms) SW development will become easy
- Climate community using it together with DTR
- EPIC will adapt its API



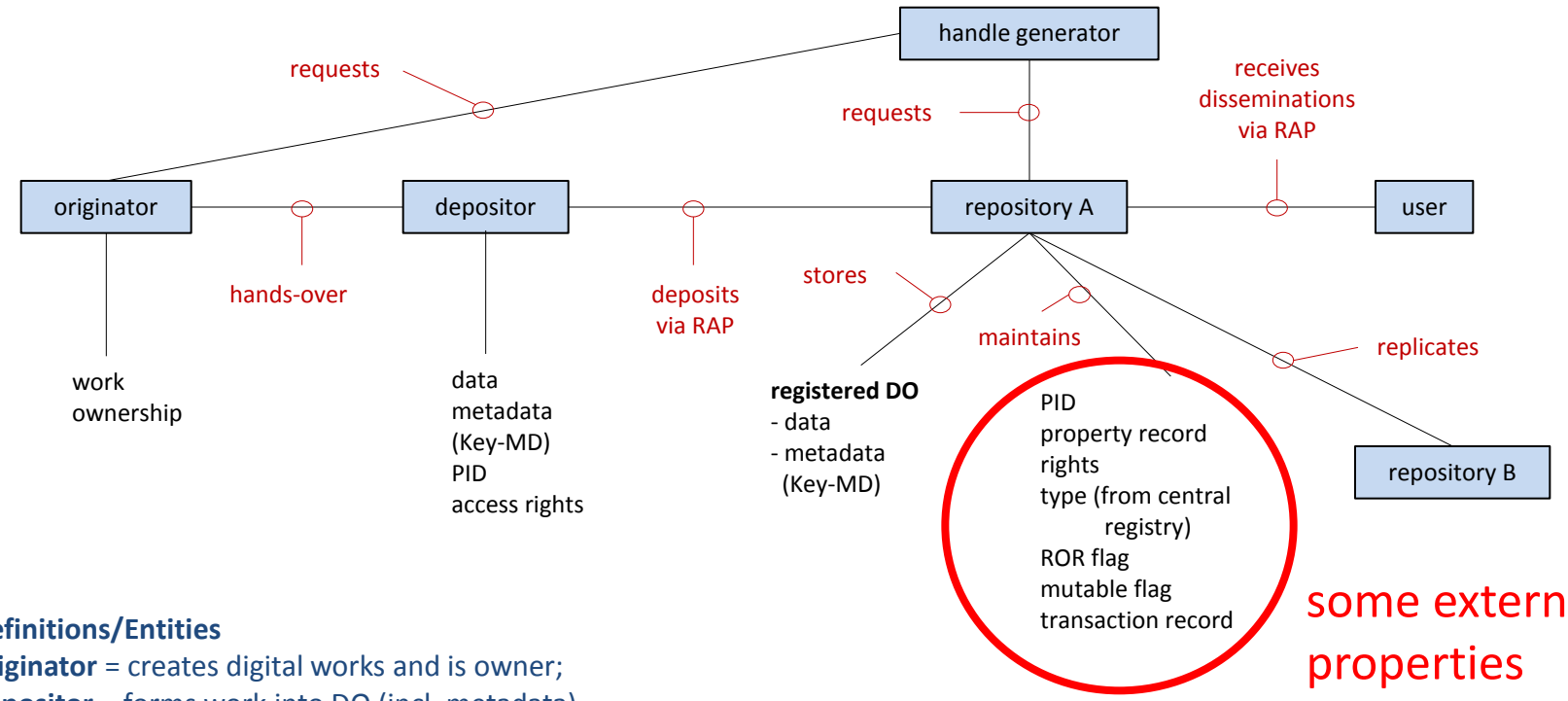
Does a DO always stand for a file?

- no - DO can be many different types off entitites
- DO could be a file or a collection of files/collections
- DO could be a query for a database
- DO could include an assertion etc.

What is the deal?

- repository needs to assure that the user always get‘s the same content!

Kahn&Wilensky Organisation

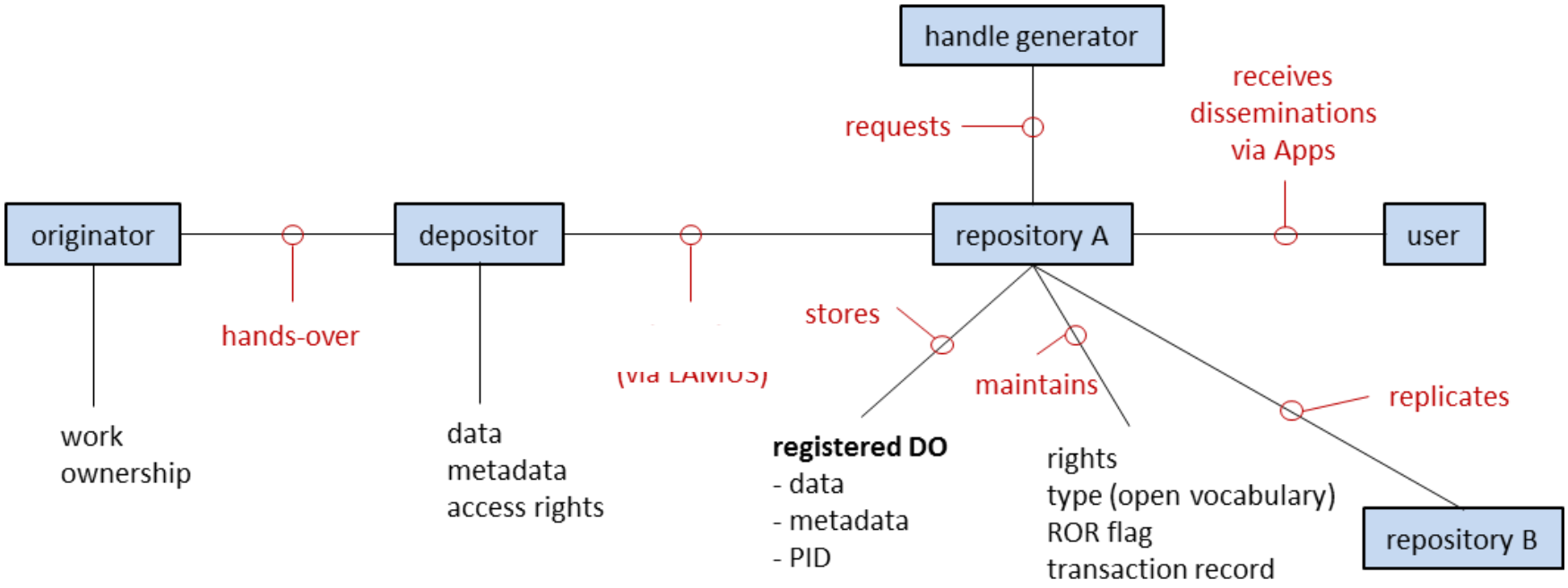


some external properties

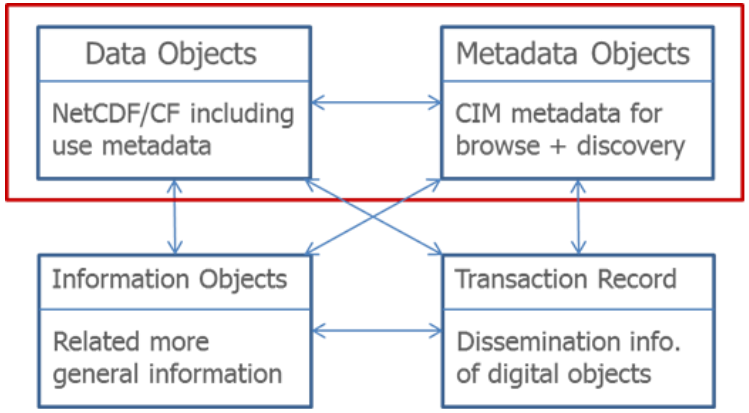
Definitions/Entities
originator = creates digital works and is owner;
depositor = forms work into DO (incl. metadata),
digital object (DO) = instance of an abstract data type;
registered DOs are such DOs with a Handle;
repository (Rep) = network accessible storage to store DOs;
RAP (Rep access protocol) = simple access protocol
Dissemination = is the data stream a user receives
ROR (repository of record) = the repository where data was stored first;
Meta-Objects (MO) = are objects with properties
mutable DOs = some DOs can be modified
property record = contains various info about DO
type = data of DOs have a type
transaction record = all disseminations of a DO

- from Kahn & Wilensky paper on Digital Objects from 2006 as basis for interactions
- worked extremely well

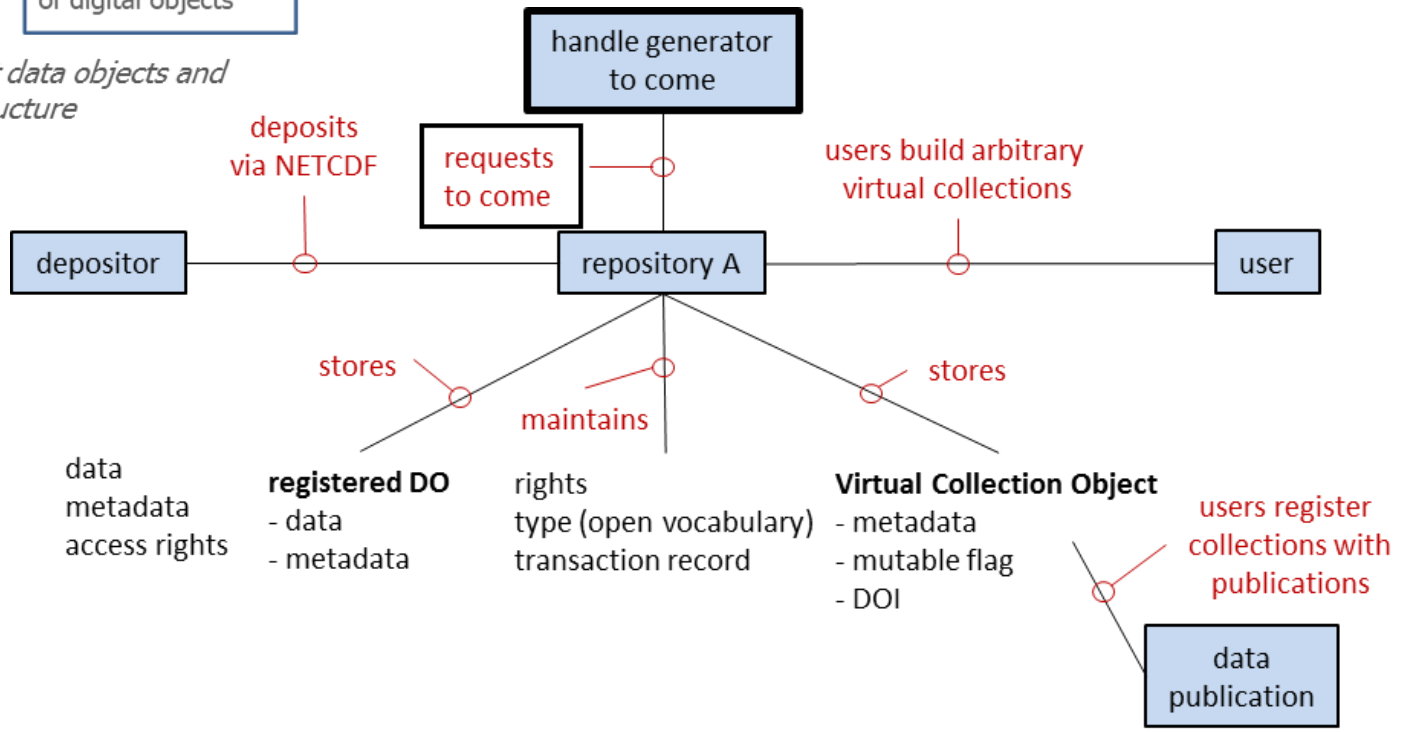
Typical organisation in CLARIN



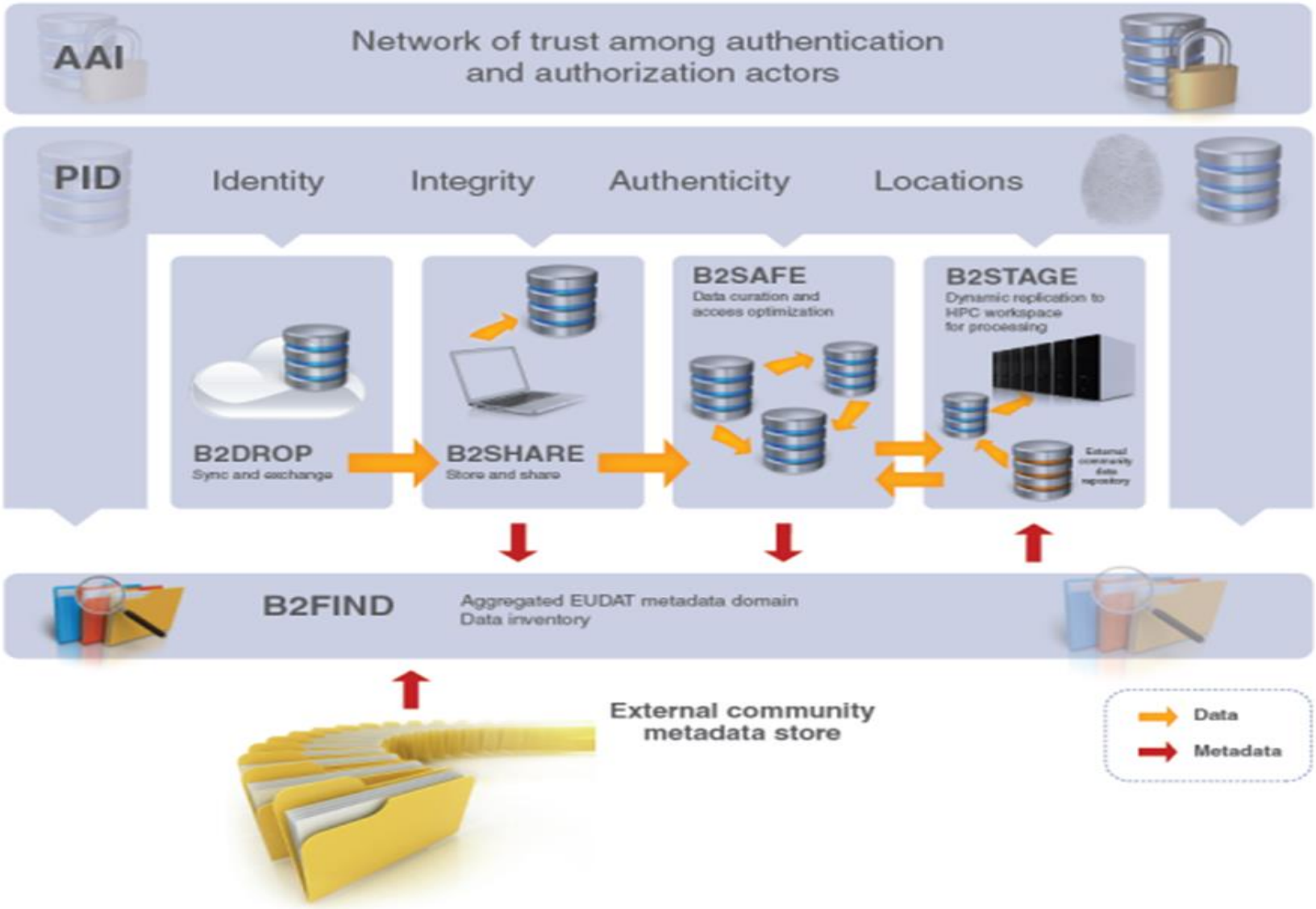
Typical organisation in ENES



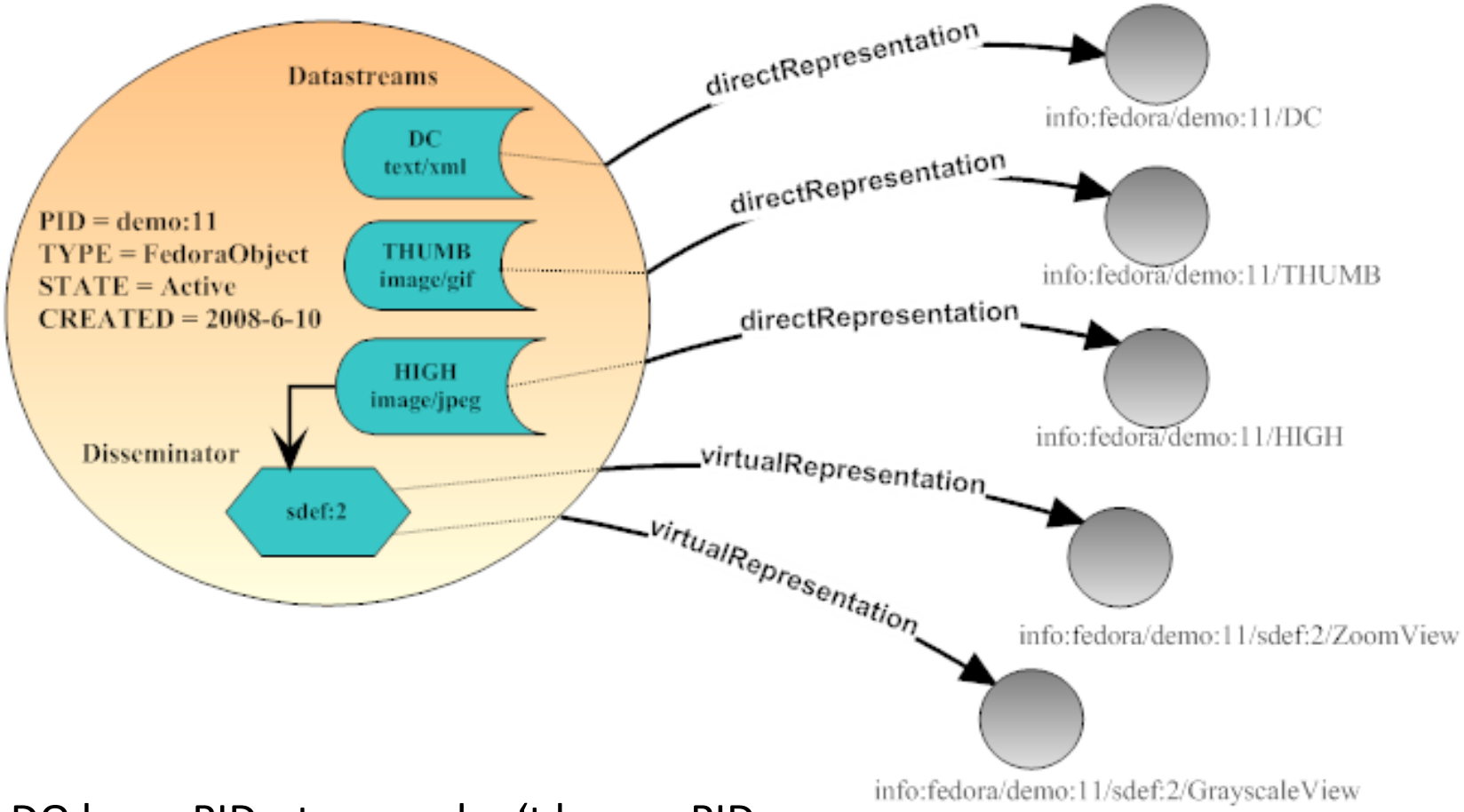
Identification of distinct data objects and P2P infrastructure



Data organisation in EUDAT

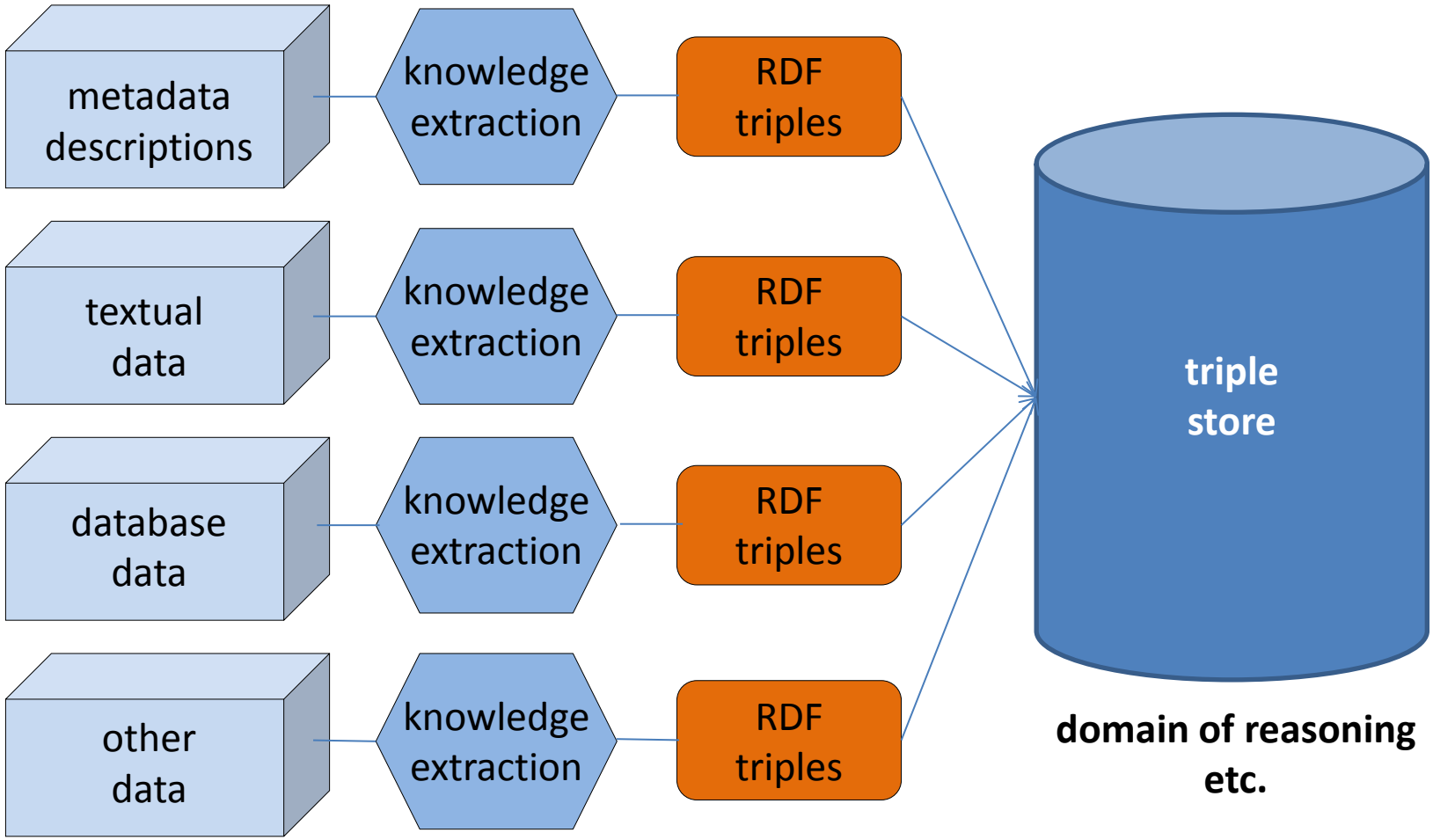


FEDORA Object Model



- DO has a PID, streams don‘t have a PID
- binding is done within the object
- some information is lacking such as existence of copies, rights record, etc.
- could be inserted but ...

Semantic Web /OLD



What is a DO in the domain of assertions – obviously any assertion needs to be identified. Which is persistent and citable store?

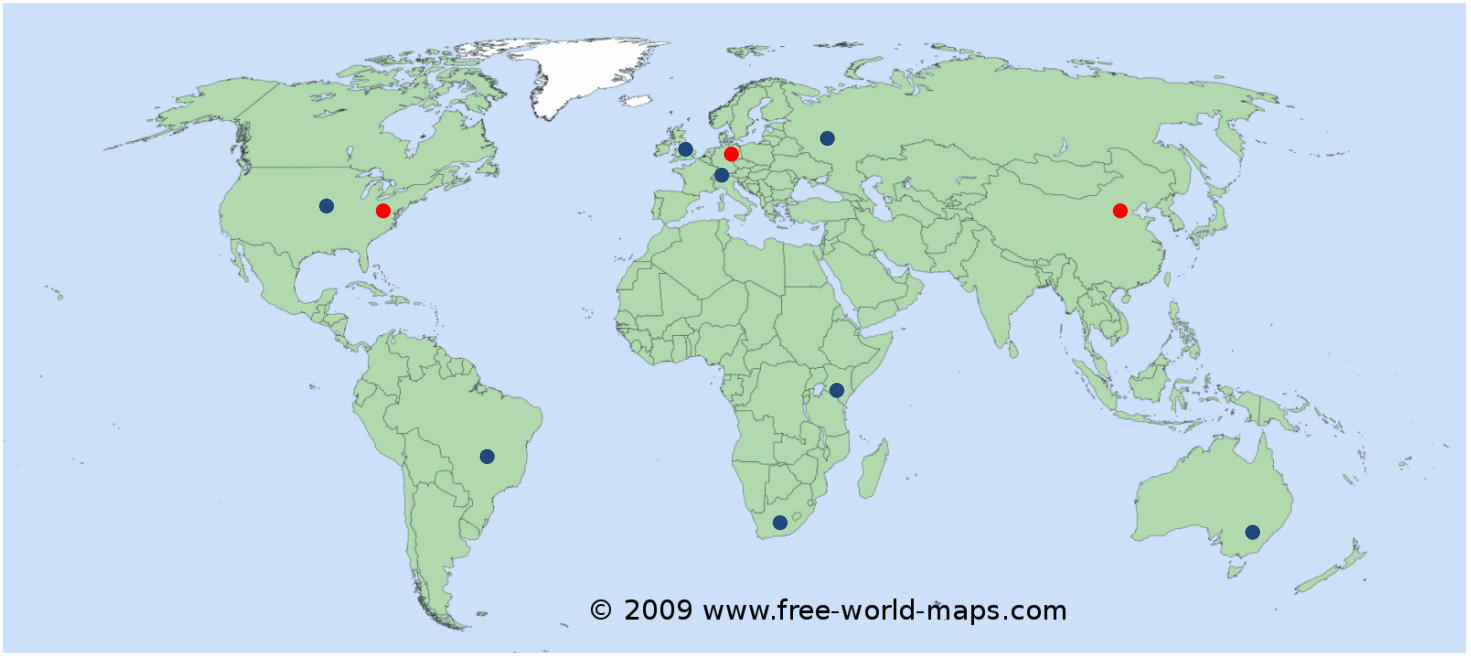
PID Registration & Resolving Systems

- it‘s all about trust
  - do we trust that PID (records) will survive?
    - finally it‘s about trust in a bunch of people and organisations
    - trust on stability of specs
  - do we trust that data will survive?
    - not per se – dependent on repository and policies applied
    - a policy could state that data can be deleted after 10 years
    - in ideal case a flag would be inserted into PID record
  - do we trust in reliability, availability and performance (resolution & registration) of a world-wide service?
  - do we trust that there will be services on top?

Types of Identifiers

- much out there – just a few to be mentioned
  - **domain IDs** in specific registries, databases, etc.
  - **BAR codes** for all sorts of things
  - **ORCID** for authors to correct for spelling variations etc.
  - **cool URIs** finally remain places, no attributes
  - **IP addresses** to be meant for routing & finding nodes in network
  - **AWK** interesting ideas for DOs, but no wide support
  - **Handles** interesting ideas for DOs and wide support
- some identifiers are just numbers
- some identifiers are designed to respond with relevant properties such as multiple locations, checksum for checks, etc. which can be administered by the record owner

Worldwide Handle Services



- HS now governed by International DONA Board acting under the umbrella of the International Telecom Union (ITU)
- currently a redundant system of MPAs in operation (one at GWDG)
- more such MPAs will come – probably in all major countries
- they act as registration authorities for centres offering services such as DOI, EPIC, CrossRef, etc.
- HS is ready to serve everyone

Recommendations

- adhere to the basic **DFT data organisation**
- participate in a **domain of registered data and metadata** to which we can refer and which we can cite
- use **Handles/DOI** where useful
- participate in a **simple binding strategy** so that our machines can find all information related to a DO
- make sure that **metadata** is accessible
- store your data in **trustworthy repositories** and take care that these are audited by DSA/WDS
- make use of generic **APIs** in your software where possible
- register your **syntax and semantics**
- don‘t rely on **encapsulated formats**
- in case of DBs make sure that **queries get a PID**

Vielen Dank für die Aufmerksamkeit.