Building SLICES Data Management Infrastructure
Design and Implementation Tasks

WP7

SLICES-PP All-Hands Meeting

5 July 2023

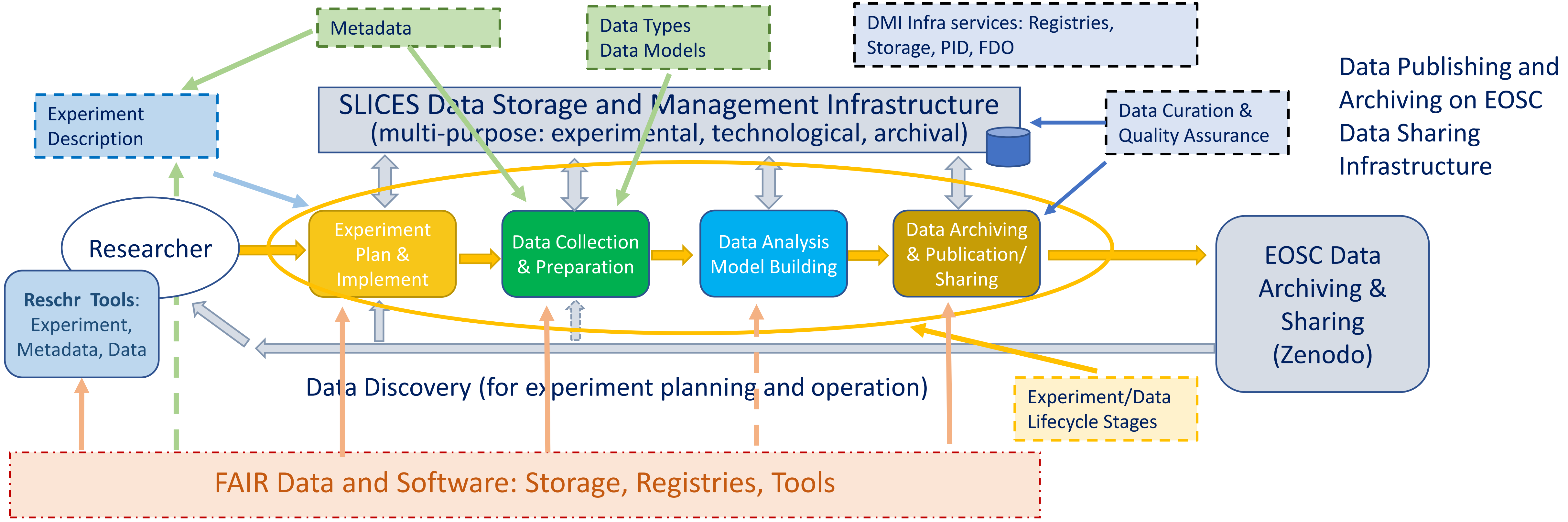
Goal and Topics to discuss

- Goal: Start complex of coordinated activities to build SLICES Data Management Infrastructure and Data and Metadata services
  - That can be used by all SLICES users/researchers, and
    - potentially provided as a service to cooperating RIs and projects
  - Define development (and implementation) streams with involvement of majority partners
- WP7 context and activities: AH meeting and Oulu meeting discussion
- FAIR: From declaration to infrastructure services
- SLICES Data Management Infrastructure (DMI): Components, Actors, Users
- SLICES Initiative on Experimental Research Reproducibility and Automation
- Assessing new/emerging technologies: Research Object (RO), EOSC Catalog, FAIR Data Object (FDO), PID, Machine Actionable DMP (maDMP)

SLICES to provide the Robust Data Infrastructure for Experiment/Data Driven Research

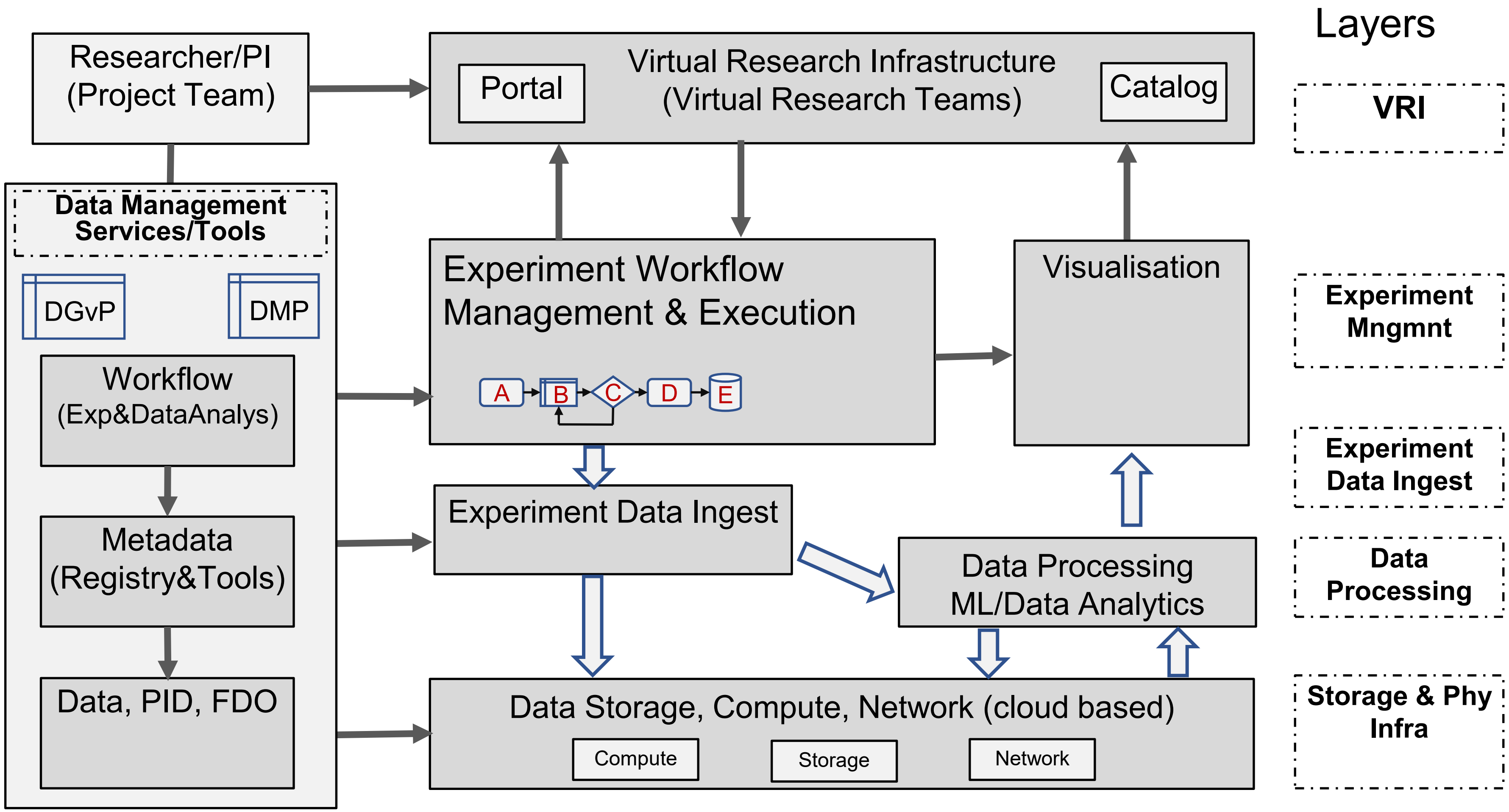
- **Experimental data are big, distributed, domain specific, serving specific communities**
  - **Require effective models and infrastructure services for Research Data Management and secure data sharing**
- Support the whole data lifecycle
  - Connected to research/experiment lifecycle or workflow
- Distributed data storage and experimental data(set) repositories
  - Supporting recognized data interoperability standards (data formats and metadata)
  - Eventually certified: RDA endorsed Maturity and certification practice
  - **Interoperability and integration with EOSC as Federated data infrastructure**
- Data management and data curation and quality assurance
  - FAIR data principles and SLICES metadata profiles (interoperable with EOSC)
- Linked data and data discovery using semantic search and knowledge graph
  - PID (Persistent IDentifier) and FDO (FAIR Digital Object) infrastructure (interoperable with EOSC)
- (Trusted) Data exchange and secure transfer protocols

SLICES Experimental Data Lifecycle Model and Dataflow


Metadata
Data Types
Data Models
DMI Infra services: Registries, Storage, PID, FDO
Data Publishing and Archiving on EOSC Data Sharing Infrastructure
Experiment Description
SLICES Data Storage and Management Infrastructure
(multi-purpose: experimental, technological, archival)
Data Curation & Quality Assurance
Researcher
Experiment Plan & Implement
Data Collection & Preparation
Data Analysis Model Building
Data Archiving & Publication/ Sharing
EOSC Data Archiving & Sharing (Zenodo)
Reschr Tools: Experiment, Metadata, Data
Data Discovery (for experiment planning and operation)
Experiment/Data Lifecycle Stages
FAIR Data and Software: Storage, Registries, Tools

- **Each Data Lifecycle stage** – experiment, data collection, data analysis, and finally data archiving, works with own **data set,** which must be **linked**.
  - All data sets need to be stored and possibly re-used in later processes.
- Many experiments and research require already existing datasets that will be available in SLICES data repositories or can be obtained/discovered in EOSC data repositories

Experimental Data Management Infrastructure



Researcher/PI
(Project Team)

Data Management Services/Tools

DGvP

DMP

Workflow
(Exp&DataAnalys)

Metadata
(Registry&Tools)

Data, PID, FDO

Virtual Research Infrastructure
(Virtual Research Teams)

Portal

Catalog

Experiment Workflow
Management & Execution

A B C D E

Visualisation

Experiment Data Ingest

Data Processing
ML/Data Analytics

Data Storage, Compute, Network (cloud based)

Compute

Storage

Network

Layers

VRI

Experiment
Mngmnt

Experiment
Data Ingest

Data
Processing

Storage & Phy
Infra

DMI layers

- Virtual RI and Researcher Portal
- Experiment Workflow Management
- Data Ingest
- Data Processing
- Storage & Physical Infrastructure

DGvP – Data Governance Policy DMP – Data Management Plan

slicesPP

Experimental Research Reproducibility as a Service

- **Experiment as a Research Object (RO)**
  - Identified with unique ID and containing smart metadata (for discovery and FAIR compliance)
  - Complying with the FDO/SFDO metadata schema
  - RO Registry: Local and integrated with EOSC
- Containing **full experiment (infrastructure) setup**
  - Components/nodes, parametrized infrastructure description and deployment sequence
  - Automation of deployment wit tools: Ansible, Terraform, shell script, others
- **Experiment description and orchestration/workflow**
  - Jupyter Notebook or Galaxy
  - Interactive Experiment configuration and management (web console and CLI)
- **Input/test data**
- **Data storage and preprocessing**
  - Data ingest link and API
  - Data model and interoperable/standard data format
  - FAIR by design: primarily metadata management
- **Measurement points and monitoring**

FAIR Data Principles: Metadata Management (GO FAIR recommendations)

Findable:

- F1 (meta)data are assigned a **globally unique and persistent identifier**;
- F2 data are **described with rich metadata**;
- F3 metadata clearly and explicitly include the **identifier** of the data it describes;
- F4 (meta)data are **registered or indexed** in a searchable resource;

Interoperable:

- I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- I2. (meta)data use vocabularies that follow FAIR principles;
- I3. (meta)data include qualified references to other (meta)data;

Accessible:

- A1 (meta)data are retrievable by their identifier using a standardized communications protocol;
  - A1.1 the protocol is open, free, and universally implementable;
  - A1.2 the protocol allows for an authentication and authorization procedure, where necessary;
- A2 metadata are accessible, even when the data are no longer available;

Reusable:

- R1 meta(data) are richly described with a plurality of accurate and relevant attributes;
- R1.1 (meta)data are released with a clear and accessible data usage license;
- R1.2 (meta)data are associated with detailed provenance;
- R1.3 (meta)data meet domain-relevant community standards;

FAIR from the technical point of view – Required Infrastructure functionality

- Findable
  - Metadata and PDI – infrastructure and tools
  - Registries and handles resolution, API
  - Policies and SLA
- Accessible
  - Repositories and data storage: infrastructure and management
  - Policy and access control: infrastructure and API management
  - Data access protocols
  - Usage Policy and Sovereignty
  - Data protection, compliance, privacy and GDPR
- Interoperable
  - Standard data formats
  - Metadata Registries and API
  - FAIR maturity level and certification
- Reusable
  - Data provenance and lineage
  - Preservation
  - Metadata, PID and API – linked or embedded into datasets

Require comprehensive **data infrastructure** to support

- **Data Storage and Registries**
- Data publication
- Data discovery
- Linked data and data lineage (provenance)
- Multiple datasets access for analysis

Scientific Objects for Metadata definition

- **Data models**: storage, databases, metadata
- **Experiment**
  - Orchestration; configuration; equipment: DUT, test generators, measurement; data storage; data models/metadata
- **Dataflow**: Stages, transformations, lineage/provenance, data models
- **Workflow**: Stages, Operations/conditions, workstations

Discussion at SLICES Summer School – 15 June 2023

Discussed topics:

1. General aspects of building SLICES Data Management Infrastructure (DMI) for Experimental Research and alignment with the SLICES Blueprint Architecture. Compliance with FAIR data principles.

2. Experiment description and metadata definition for reproducibility and sharing

3. Inventory of data produced in SLICES. Definition of data models and metadata

4. Data Management and metadata management tools for researchers

Goal:

- The definition of Requirements to the DMI as part of SLICES Blueprint Architecture and identification of task to implement DMI and necessary data practices .

Outcome:

- Identified Design and Implementation Tasks
  - (1) SLICES Data Management Infrastructure (DMI) (2) General Metadata definition and management (3) Experiment description and metadata (4) SLICES Blueprint Architecture (5) Metadata Management tools

SLICES Summer School 13-15 June 2023, Oulu, Finland

SLICES Data Management infrastructure services for Experimental Research Reproducibility

https://drive.google.com/drive/folders/1mfoZs3OXOx_Klhy1r6-YVXIW_4MtadFh?usp=sharing

- SLICES Initiative on Experimental Research Automation and Reproducibility
    - Reproducible Experimental Research as a Service
- Elements of the Experimental Research Reproducibility
    - Data types produced in SLICES
- FAIR data principles and Metadata Management
- (Prospective) SLICES Data Management Infrastructure

slicesPP

(1) SLICES Data Management Infrastructure (DMI)

Requirements and services

- SLICES must **create and maintain its own Data Management Infrastructure** that will include central data storage and connected data storage nodes operated by partners and big experimental facilities.
- DMI must include all necessary **services to support the whole experimental data lifecycle** and also include reference datasets such as required for experiment execution, or ML/AI algorithms training.
- SLICES DMI may use where possible **external data storage, registries, metadata services or data discovery services**, which should be **federated** with the SLICES DMI.
- DMI must implement **federated access control principles and allow integration and federation with EOSC and EGI** data management infrastructure and services, in particular for data sharing, publication, and access.

(1) SLICES Data Management Infrastructure (DMI) - Tasks

Tasks

- Task 1. Define who will host and operate the SLICES central data storage and an initial set of storage nodes. Decisions should be made on how to federate distributed storage nodes and provide transparent access to the whole DMI. – All, Exec Board

- Task 2. Assess options for establishing Metadata Registry to serve SLICES data management purposes, Identify what services should be deployed in the SLICES DMI and which services can be used from external providers/operators in EOSC or EGI infrastructure? Who can host and operate necessary metadata services? - UCLAN

(2) General Metadata definition and management

Requirements and services
Effective and consistent metadata management is the foundation of the FAIR data principles implementation.

All data are defined by the data models, metadata, data formats and data types. Metadata are defined as part of the data model.

- For SLICES as RI for experimental studies in digital technologies and ICT, metadata includes three main areas:
  - **General services description**: metadata profiles and metadata will be used for publishing SLICES services in EOSC Catalog and own services catalog
  - **Description of data collected, produced and handled in SLICES-RI** that include experimental data, staged/processed data, archival data, publications, reports, activities and management data. Additional data categorization is required.
  - **Experiment description** that must include all necessary information required for experiment reproducibility and deployment
- Two other categories of metadata that may be required to support SLICES include:
  - **Infrastructure descriptions** that are required for infrastructure management and monitoring (network devices, network traffic, status and events). This type of metadata are well supported by existing network management and service management standards (SNMP MIB-II and related, DMTF CIM and CIMI, TMForum SID)
  - **Metadata for data processing and lineage**, in particular, for data used in ML and AI processes
- Defining metadata often includes defining **metadata namespaces** that will create and basis for unique metadata elements identification and consequently discovery, sharing and integration.

(2) General Metadata definition and management - Tasks

Tasks

- Task 3. Assess existing metadata format and provide recommendations for metadata formats for SLICES services and resources, data produced in SLICES experiments. - UCLAN
  - Make a survey among SLICES partners
- Task 4. Assess existing metadata format for infrastructure description and management and their relevance for SLICES use cases. - UvA
- Task 5. Assess existing metadata format and provide recommendations for metadata formats for experiment description (see additional information below). – UvA, UCLAN, TUM

(3) Experiment description and metadata

Requirements and services

Consistent/full experiment description and corresponding metadata must ensure experiment/experimental research reproducibility and FAIR data sharing.

- The following data types and metadata are considered essential for consistent experiment description:
  - Experiment abstract model with parameters, input variables and variables under test (as it is known at the beginning)
  - Experiment setup/infrastructure, including network equipment, VMs/containers
  - Configuration of all infrastructure components, deployment sequence (presumably Ansible playbook, Terraform plan, or Jupyter Notebook)
  - Test generators, measurement equipment and sensors (and corresponding infrastructure points)
  - Environment description
  - Experiment workflow
  - Data ingest process, data preprocessing and assessment
  - APIs for experiment setup, monitoring, and data collection
- Data models and metadata must be defined for all types of data describing the experiment.
- For some well established experiments, data models maybe defined for the specific data storage and database type, such as data lakes, SQL database, kye-value, document based, or triple storage (for semantic data).
- Experiment as a Research Object must be assigned a unique identifier and experiment/object type, optionally registered schema and domain namespace.

(3) Experiment description and metadata - Tasks

Tasks

- Task 6. Make an inventory of all data required for full experiment reproducibility (with the target for portability), including infrastructure, environment, variables, used data formats or data models. Such inventory should document a current practice, which further will be used for more formal definitions of data models and metadata. - TUM
- Task 7. Investigate what and how essential metadata can be extracted from the Ansible playbook or Jupyter Notebook describing experiment setup and orchestration. - UVA

(4) SLICES Blueprint Architecture

Requirements and services

SLICES Blueprint defines a basic/core/instant infrastructure setup that can be used for running experiments on 5G/6G and related networking technologies.

- To achieve effective composability and flexible customization of the SLICES experimental setups, the following data and metadata should be defined (similar for the general experiment setup as described above):
  - Services deployed in the Blueprint and corresponding APIs
  - Input and output data or signals
  - Configuration of all elements, including RAN (RU – Radio Units, UE – User Equipment), core 5G network, dedicated network (VPN or VPC), network switches, servers
  - Computational nodes/instances type and configuration: hardware (AMD/Intel, RAM, OS, firmware)
  - Operational environment that may need to be documented for experiments
  - Infrastructure design patterns or templates (in the form of Ansible playbooks or Terraform plans)
  - Monitoring or measurement point and corresponding API
  - Experiment specific or other data collected in the infrastructure

(4) SLICES Blueprint Architecture - Tasks

Tasks

- Task 8. Make an inventory of all data used or required for the Blueprint infrastructure description, deployment and operation as a part of the experiment setup, as described above. Such inventory should document a current practice, which further will be used for more formal definitions of data models and metadata. - INRIA

(5) Metadata Management tools

Requirements and services

- SLICES must provide metadata management tools that would help researchers and data managers/data stewards to create, combine, transform/map and publish metadata in a consistent way and with minimum efforts and maximum automation.
- Metadata tools should support the creation of metadata when publishing research results, papers, datasets, reports.

Tasks

- Task 9. Assess existing metadata registries and metadata management tools for different categories of metadata and use cases/scenarios and provide recommendations to project partners. UCLAN

How/Where to start all this?

- Bottom-up approach: Start from pilot use cases
  - Pilot #1 – POS by TUM: Testbed and experiments description
  - Pilot #2 – Metadata Registry: Pilot implementation – UCLAN?
  - Pilot #3 – Metadata Tools: For one of testbeds; For classes of Scientific Objects – UCLAN? UvA
  - Pilot #4 – Experimental data storage with external access (and EOSC integration) - TBD

Approach – How we can do it (in Agile style)

- Learn from existing implementations and services, use Open Source platforms
  - Solutions implemented in EOSC
  - Metadata management in Data Warehouses: DW Data Objects Business, Operational, Technical
- Develop architecture vision and plan
- Implement based on selected OS platform (beneficially OpenStack)

Popular open-source data catalog tools

- 6 most popular open-source data catalog tools in 2023 https://atlan.com/open-source-data-catalog-tools/
- Apache Atlas - https://github.com/apache/atlas
  - Metadata classification
  - Metadata types and instances
  - Search and Lineage
  - Security and Data Masking
- Amundsen Lyft - https://github.com/amundsen-io/amundsen
- LinkedIn DataHub - https://github.com/linkedin/datahub/
- Netflix Metacat - https://github.com/Netflix/metacat
- OpenMetadata - https://open-metadata.org/
- Open Data Discovery - https://github.com/opendatadiscovery

EOSC Metadata Schema and Crosswalk Registry (MSCR)

https://faircore4eosc.eu/eosc-core-components/metadata-schema-and-crosswalk-registry-mscr

- The EOSC MSCR supports registering schemas/crosswalks hosted elsewhere as well hosting them in the repository
- It offers basic data management support: PIDs, metadata, versioning and provenance information.
- Supports a GUI for visually creating crosswalks between metadata schemas
  Provides an API and guidelines for organisations to register and maintain metadata schemas and crosswalks
- When registering metadata schema users are able to provide detailed data-type information for fields and attributes using the DTR
- Provides a (meta-)data interoperability service that facilitates conversion between metadata schemas
- The metadata schema and crosswalk registration process and governance is aligned with the EOSC Provider and Resource onboarding process (currently operated by EOSC Future)
- The MSCR will be integrated with all relevant EOSC-Core services: AAI, monitoring and helpdesk

FAIR Evaluator

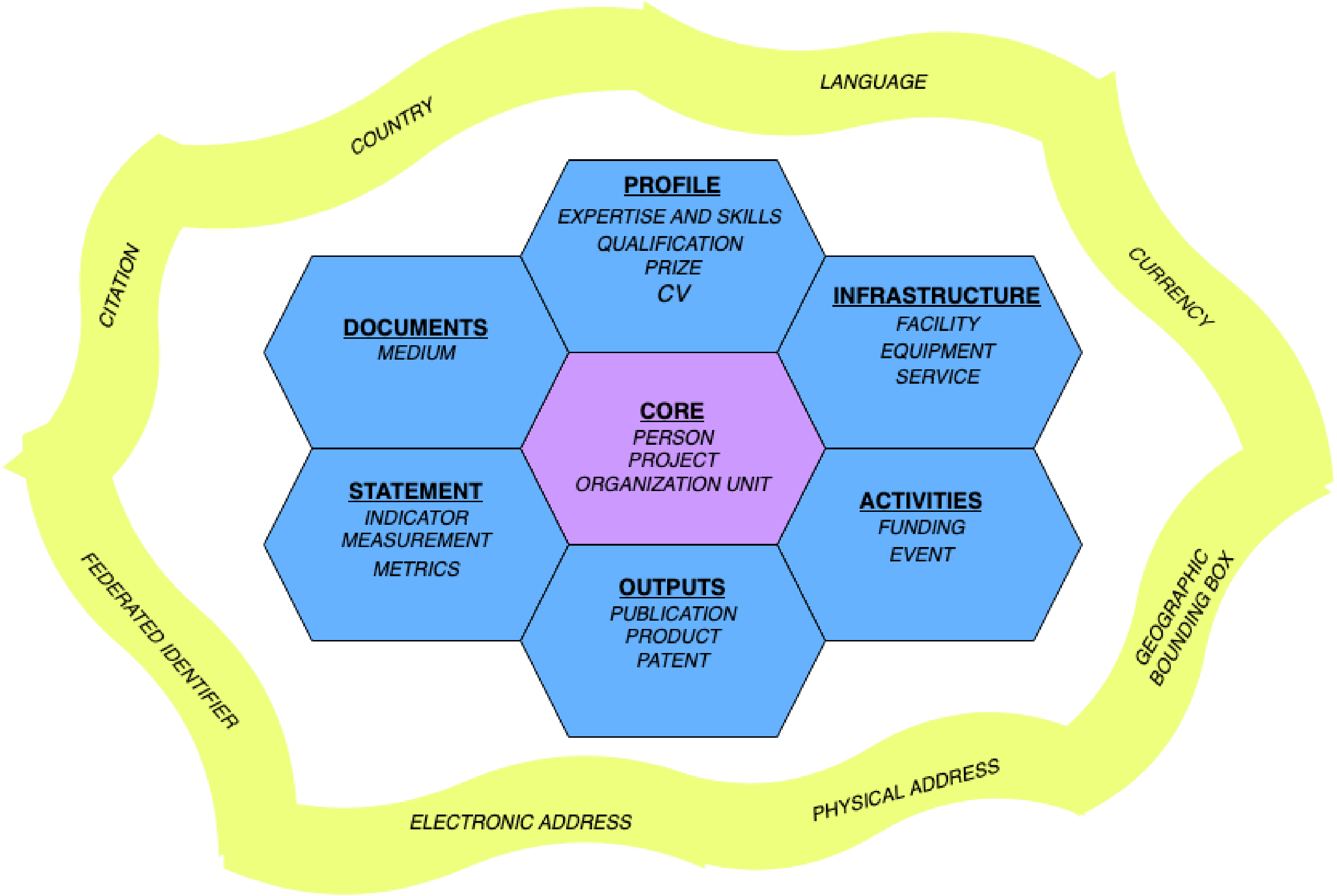
https://github.com/EOSC-synergy/FAIR_eva/blob/main/docs/index.md

```
# Metadata terms to find the resource identifier
identifier_term = ['identifier']

# Metadata terms to check richness (generic). These terms should be
included [term, qualifier]. None means no qualifier
terms_quali_generic = [['contributor',None],
['date', None],
['description', None],
['identifier', None],
['publisher', None],
['rights', None],
['title', None],
['subject', None]]

# Metadata terms to check richness (disciplinar). These terms should be
included [term, qualifier]
terms_quali_disciplinar = [['contributor', None],
['date', None],
['description', None],
['identifier', None],
['publisher', None],
['rights', None],
['title', None],
['subject', None]]
```

```
# Metadata terms that defines accessibility
terms_access = [['access', "], ['rights', "]]

# Metadata terms wich includes controlled vocabularies. More controlled
vocabularies can be imlpemented in plugins
terms_cv = [['coverage', 'spatial'], ['subject', 'lcsh']]

# List of data formats that are standard for the community
supported_data_formats = [".txt", ".pdf", ".csv", ".nc", ".doc", ".xls", ".zip",
".rar", ".tar", ".png", ".jpg"]

# Metadata terms that defines links or relation with authors, contributors
(preferebly in ORCID format)
terms_qualified_references = ['contributor']

# Metadata terms that defines links or relation with other resources,
(preferebly in ORCID format, URIs or persistent identifiers)
terms_relations = ['relation']

# Metadata terms that defines the license type
terms_license = [['license', ", "]]
```

CERIF (Common European Research Information Format)

https://github.com/EuroCRIS/CERIF-Core



CERIF covers the domain of research information with a focus on the administrative and organizational aspects. It aims to provide machine-processable representation of research information and cover situations where research information crosses borders between organizations (or between organizations and people).

Still to investigate

- CERIF-DataModel – the data model itself, including its full documentation
- CERIF-Vocabularies
- CERIF-TG-Tools – tools that support work with the CERIF model

Pilot #1 with TUM on Metadata definition for POS experiments

https://gallenmu.github.io/pos-artifacts//web/2020-10-07_23-22-39_868017.html

EXPERIMENT SETUP

The task of this script is the initialization and preparation of the experiment execution. It is executed on the management host.

- Experiment script

GLOBAL AND LOOP PARAMETERS

List of parameters that were used for this instance of the experiment.

- Global parameters
- Loop parameters

LOAD GENERATOR

The task of this node is the setup and execution of the load generator creating the load for the device under test.

- Local parameters
- Setup script
- Measurement script

DEVICE UNDER TEST

The task of this node is the setup and execution of the investigated packet processing device.

- Local parameters
- Setup script
- Measurement script

EVALUATION

The evaluation script that plots the results.

- Evaluation script call

PUBLICATION

The publication script that created this website.

- Publication script call

Types of data in ICT Experiment operation

- **Variables**
- **Configuration (equipment, DUT)**
- Test data
- Orchestration and workflow
- Measurement data (data model, metadata)
- Storage
- Environment?

Example: Global parameters

```
{
"dut_egress_if": "ens6f1",
"dut_egress_ip": "10.0.0.21",
"dut_egress_mac": "52:54:00:80:0a:21",
"dut_ingress_if": "ens6f0",
"dut_ingress_ip": "10.0.1.22",
"dut_ingress_mac": "52:54:00:80:0a:22",
"loadgen_egress_dev": 0,
"loadgen_egress_if": "ens5",
"loadgen_egress_ip": "10.0.1.23",
"loadgen_egress_mac": "52:54:00:78:0a:20",
"loadgen_enable_ip_sw_chksum_calc": 0,
"loadgen_ingress_dev": 1,
"loadgen_ingress_if": "ens4",
"loadgen_ingress_ip": "10.0.0.20",
"loadgen_ingress_mac": "52:54:00:78:0a:23",
"moongen_dir": "/root/moongen",
"moongen_repo": "https://github.com/gallenmu/MoonGen",
"moongen_repo_commit": "e56bb072d8892e3d5c288dec5aa8e0540cc17eb8"
}
```

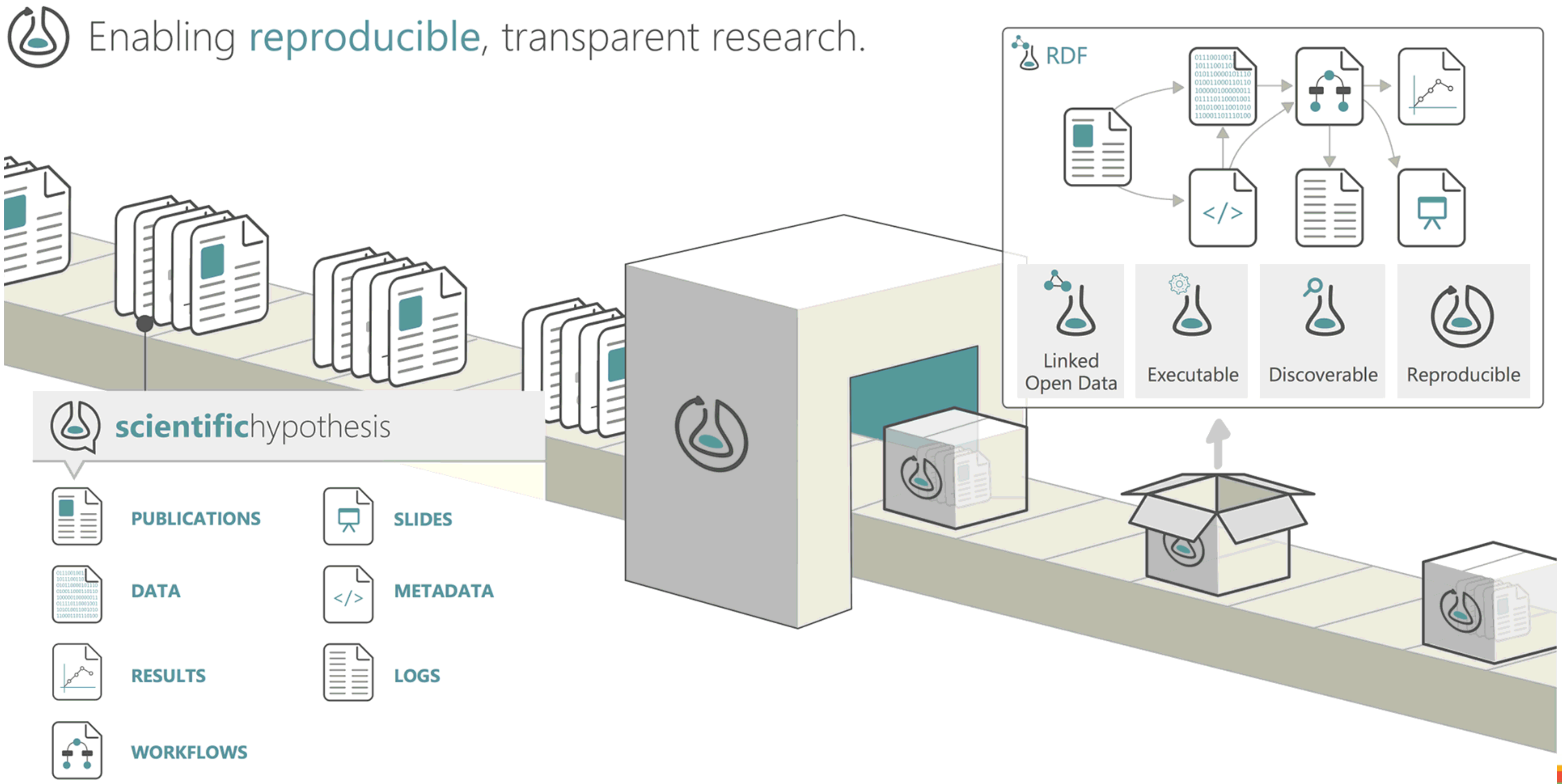
Example: Loop parameters

```
{
"pkt_rate": [
100000,
200000,
300000,
400000,
500000,
600000,
700000,
800000,
900000,
1000000,
1100000,
1200000,
1300000,
1400000,
1500000,
1600000,
1700000,
1800000,
1900000,
2000000
],
"pkt_sz": [
64,
128,
256,
512,
1024,
1280,
1500
]
}
```

New/emerging technologies to consider

- Research Object (RO)
- EOSC Catalog
- FAIR Data Object (FDO)
- PID
- Machine Actionable DMP (maDMP)

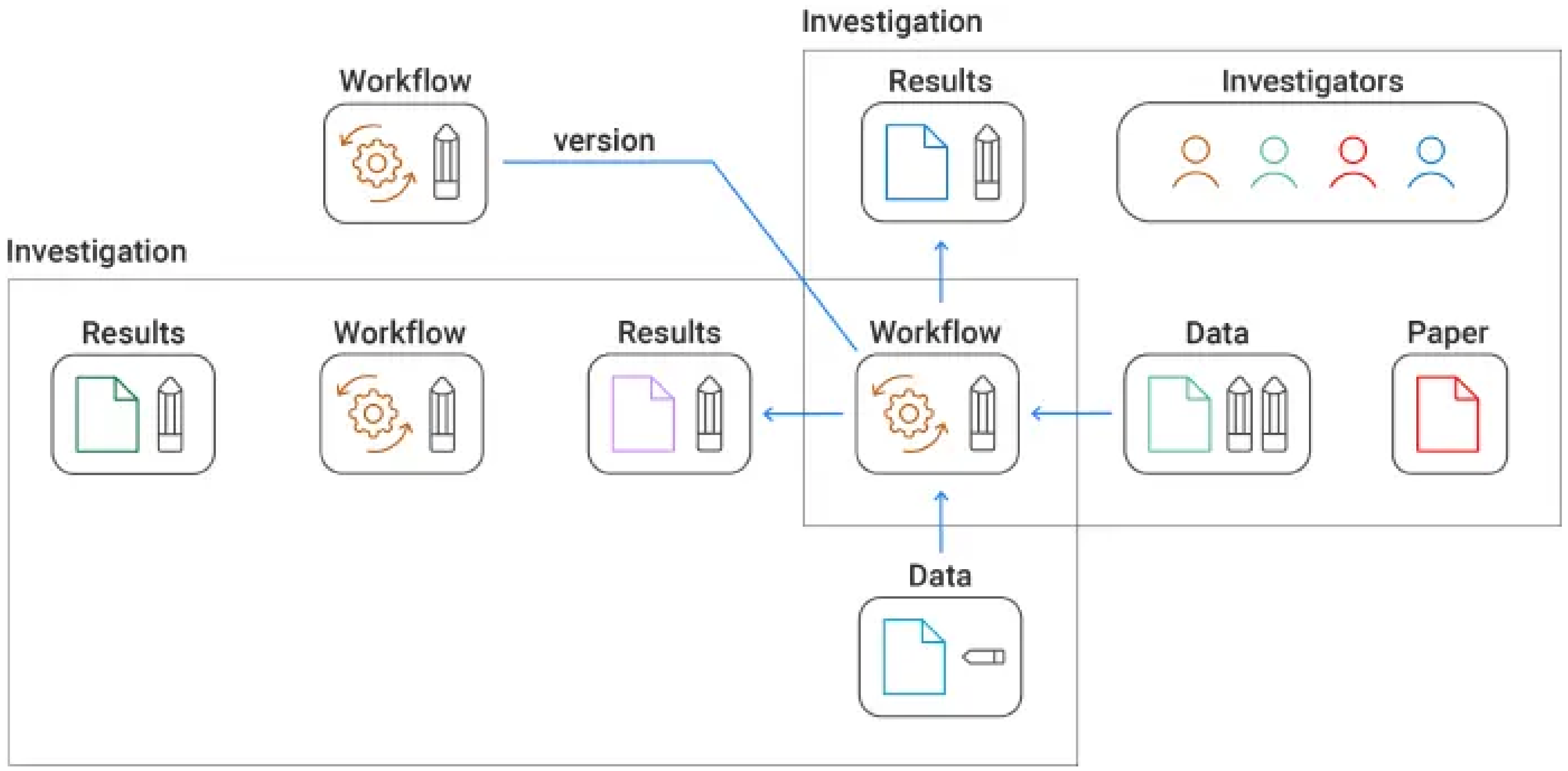
To Investigate: Research Object (2020?) - https://www.researchobject.org/ (https://www.reliance-project.eu/)



Research Object

- Multidimensional digital objects that encapsulate essential information about experiments and investigations to facilitate their reusability, reproducibility and better understanding.
- A research object can aggregate an arbitrary number of heterogeneous resources, which can be internal or external (linked by reference) to the research object location, such as
  - the data used or the results produced in an experiment study,
  - the (computational) methods employed to produce and analyse that data, and
  - the people involved in the investigation.
  - Additionally, the resources in the research objects can be organised within folders (a special type of resource), to facilitate their inspection.

RO Structure and Inter-relations



Actions

- Agreement by partners on Tasks assignment
- Provide design information for the Blueprint Design Workshop

Additional Information

- WP7 Context
- Metadata in Data Warehouses
- SZTAKI Cloud Reference Architectures

WP7: Data Management and Ethics Requirements

Work Done – February – April 2023 and identified tasks

- **D7.1 Data Management Plan (M6, UCLAN, R, PU) The data management plan including analysis of data produced by SLICES-RI, data governance principles and policy for data protection, sharing and reuse. - Submitted**
- WP7 coordination calls on 24 March 2023 and 5 April 2023 + Discussion at SLICES-PP AH meeting – To develop short-medium term WP7 activities plan
  - Longer term: Based on submitted D7.1 DMP, prepare plan on implementing of DMP and DM Office for SLICES-RI (PA/UCLAN)
- Identified Tasks
  - Prepare a Whitepaper with the vision how SLICES-RI can organise own data management infrastructure.
  - Specify requirements to data/metadata management services and to data management infrastructure to support experimental research
    - Specify metadata profiles for selected pilot experimental setups
  - Prepare suggestions for the technical aspects and infrastructure services to address current trends in research data management: machine-actionable DMP, FAIR maturity, data management for AI
  - Plan a joint SLICES and Chameleon workshop on Experimental research reproducibility

Discussion at AH meeting 13 April 2023

- Where to store data: Organisation/project/experiment or RI
  - How to share responsibility and service? SLICES at least should maintain metadata registry!
  - Can SLICES provide data storage in 1-2 yrs for external parties?
- Do/Should we enforce experiments (and researchers) to share data?
  - Open and shared data are started with ensuring data
  - FAIR and sharing is wider than Experiment reproducibility
- Can we identify 2-3 pilots for RDM in SLICES?
  - With real experiments/facilities: (1) TUM POS!! (TUM, UvA, MI, UCLAN?) (4) Define User services? (UCLAN?) (3) CONVERGE (2) ?
  - Experience to be shared between SLICES partners and general recommendations to be produced
  - Align with US CHAMELEON ? – Proposal/plan by SLICES
- What are the first steps we can do: Define metadata profiles, training, etc?
  - Revisit use cases defined for SLICES-RI and identify key requirements and technical functionalities
  - Extracting metadata from Jupyter Notebooks, Ansible templates, experimental equipment/environment

Development (and implementation) streams

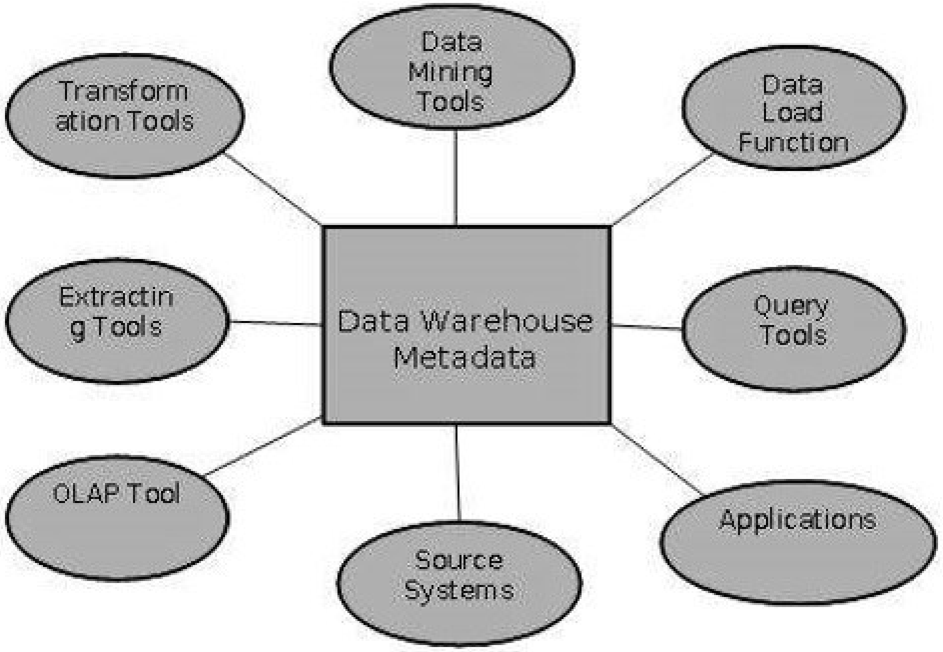
- (0) FAIR data principles: From words to infrastructure services
- (1) Data management and Data Management Plan (DMP) management
- (2) Data and metadata management services and infrastructure
- (3) Metadata definition and management
- (4) Developer and researcher tools:
  - + (primarily) metadata management - because it should be integrated into the experiment management platform, and
  - + (secondary) data management - because at the initial stage we can use existing services, f.e. by EOSC or EGI

Metadata in Data Warehouses

Metadata can hold all kinds of information about DW data
https://www.softwaretestinghelp.com/metadata-in-data-warehouse-etl/

https://www.infolibcorp.com/metadata-management/data-warehouse

- Source for any extracted data.
- Use of that DW data.
- Any kind of data and its values.
- Features of data.
- Transformation logic for extracted data.
- DW tables and their attributes.
- DW objects
- Timestamps



Typical Data Warehouse Metadata

https://www.infolibcorp.com/metadata-management/data-warehouse

- **Metadata**
- Source system and target system table mappings.
- Destination schema.
- Routines and scripts scheduling, mode of running.
- Routine version and location.
- Automated extract tool settings, if applicable.
- Quality reconciliation and control routines.
- **Dimension Table Management Metadata**
- Dimensional models.
- Dimensional schema info.
- Slowly changing dimension policies for each incoming descriptive attribute.
- Surrogate key assignment rule and counters.
- **Transformation and Metadata**
- Business specifications of the transformation rules.
- Source to destination mappings.
- Location of the transformation Scripts.
- Staging area schema.
- Load scripts, schedule and location.
- **Metadata for Job Control and Audit Logs as well as Documentation**
- Data lineage and audit records (where exactly did this record come from and when?).
- Data Extraction, transform, loading run time logs, success summaries, and time stamps.
- Data transform software version numbers.
- Security settings for extract files, extract software, and extract metadata.
- Security settings for data transmission (i.e., passwords, certificates).
- Data staging area archive logs and recovery procedures.
- Data staging archive security settings.
- **DBMS Metadata**
- DBMS system table contents.
- Partition settings.
- Indexes.
- DBMS-level security privileges and grants.
- View definitions.
- Stored procedures and SQL administrative scripts.
- Backup procedures and backup security.
- **Business Metadata**
- Business names and descriptions for columns, tables, groupings, and so on.
- Canned query and report definitions.
- Default join specification tool settings.
- Bi tools metadata.
- Reporting tools metadata.
- Where used reports for data elements, tables, views, and reports.

Types of metadata collected

- **#1) Backroom Metadata:** Directs the DBAs (or) the end-users on extract, clean and load processes.
- **#2) Front room Metadata:** Directs the end-users to work with BI tools and reports.
- **#3) Process Metadata:** This stores ETL process metadata such as the number of rows loaded, rejected, processed and time taken to load into a DW system, etc. This information can also be accessible to the end-users.
- **#4) Data Lineage:** This stores the logical transformation for each source system element to the DW target element.
- **#5) Business Definitions:** The context for DW tables has been derived from the business definitions. Every attribute in a table is associated with a business definition. Hence these should be stored as metadata (or) any other document for future reference. Both the end-users and the ETL team depend on these business definitions.
- **#6) Technical Definitions:** Technical definitions are exclusively used in the data staging area more than the business definitions. The main purpose is to reduce the ambiguity while creating staging tables and to reuse any existing tables. Technical definitions will store the details of each staging table such as its location and structure.
- **#7) Business Metadata:** Data will be stored in business terms for the benefit of end-users/analysts/managers/ any users. Business metadata is proxy to the source system data i.e. no data manipulations will be done on it. It can be derived from any business documents and business rules.
- **#8) Technical Metadata:** This will store technical data such as tables attributes, their data types, size, primary key attributes, foreign key attributes, and any indexes. This is more structured when compared to business metadata.
- **#9) Operational Metadata:** As we know the data into the DW system is sourced from many operational systems with diverse data types and fields. DW extracts transform such data into the unique type and load all this data into the system.
- **#10) Source System Information:**
- **#11) ETL Job Metadata:** ETL job metadata is very important as it stores the details of all the jobs to be processed in the schedule, to load the ETL system.
- **#12) Transformation Metadata:** Transformation metadata stores all ETL process-related construction information. Every single manipulation of data in the ETL process is known as data transformation.

#11) ETL Job Metadata

- **#11) ETL Job Metadata:** ETL job metadata is very important as it stores the details of all the jobs to be processed in the schedule, to load the ETL system.
- **This metadata stores the following information:**
- **Job Name:** ETL job name.
- **Job Purpose:** The purpose of running the job.
- **Source Tables/Files:** It provides the names and location of all tables and files from which the data is being sourced by this ETL job. This can have more than one table (or) file name.
- **Target Tables/Files:** It provides the names and location of all tables and files to which the data is being transformed by this ETL job. This can have more than one table (or) file name.
- **Rejected Data:** It provides the names and location of all the tables and files from which the intended source data has not been loaded into the target.
- **Pre Processes:** It provides the jobs (or) script names on which the current job is dependent. It means those have to be successfully executed before running the current job.
- **Post Processes:** It provides the jobs (or) script names that should be run immediately after the current job to complete the process.
- **Frequency:** It provides information on how frequently the job should be executed i.e. daily, weekly (or) monthly.

#12) Transformation Metadata

- **#12) Transformation Metadata:** Transformation metadata stores all ETL process-related construction information. Every single manipulation of data in the ETL process is known as data transformation.
- Any set of functions, stored procedures, cursors, variables and loops in the ETL process can be considered as transformations. But such transformations cannot be documented separately as metadata.
- The entire ETL process is built up with data transformations. Few transformations in ETL can be predefined and used across the DW system. ETL developers spend their time in building (or) re-processing all the data transformations. Reusing the predefined transformations during the ETL process development will speed up the work.
- **Read through the below data transformations that you can find in ETL:**
- **Source Data Extractions:** This involves data transformations to read from source system data such as a SQL Select query (or) FTP (or) reading XML/mainframe data.
- **Surrogate Key Generators:** The new sequence number that should be generated for every database table row is stored as metadata.
- **Lookups:** Lookups can be formed with all the IN statements, inner joins, and outer joins. These are mainly used to hold the surrogate keys from all the respective dimension tables while loading a fact.
- **Filters:** Filters are recommended to sort out the data that should be extracted, loaded and rejected in the ETL process. Filtering the data in the early stages of the ETL system is a good practice. Filters are applied depending on the business rules (or) constraints.
- **Aggregates:** Depending on the level of data granularity, the metadata related to aggregate functions can be used such as sum, count, average, etc.
- **Update Strategies:** These are the rules applied to a record while updating the data. If there is any modification in the existing data, then this will indicate if a record should be added, deleted (or) updated.
- **Target Loader:** Target loader will store the details of the database, table names and column names into which the data should be loaded through the ETL process. Moreover, this will also store the details of bulk load utility if any, that is performed while loading data into the ETL system.
- Every transformation can be named distinctively with a brief note about its purpose.

Metadata example?

- SRC_&lt;name of the table&gt;
  SEQ_&lt;surrogate key column name&gt;
  LKP_&lt;Name of the table referred&gt;
  FIL_&lt;Purpose&gt;
  AGG_&lt;Purpose&gt;
  UPD_&lt;Update type&gt;_&lt;Name of table&gt;
  TRG_&lt;Name of table&gt;

FAIR Core for EOSC -

1. EOSC Research Discovery Graph (RDGraph) to deliver advanced Discovery tools across EOSC resources and communities;
2. EOSC PID Graph (PIDGraph) to improve the way of interlinking research entities across domains and data sources on the basis of persistent identifiers (PIDs);
3. EOSC Metadata Schema and Crosswalk Registry (MSCR) to support publishing, Discovery and access of metadata schemas and provide functions to operationalize metadata conversions by combining crosswalks;
4. EOSC Data Type Registry (DTR) to provide user friendly APIs for metadata imports and access to different data types and metadata mappings;
5. EOSC PID Meta Resolver (PIDMR) to offer users a single PID resolving API in which any kind of PID can be resolved through a single, scalable PID resolving infrastructure;
6. EOSC Compliance Assessment Toolkit (CAT) to support the EOSC PID policy compliance and implementation;
7. EOSC Research Activity Identifier Service (RAiD) to mint PIDs for research projects, allowing to manage and track project related activities;
8. EOSC Research Software APIs and Connectors (RSAC) to ensure the long-term preservation of research software in different disciplines;
9. EOSC Software Heritage Mirror (SWHM) to equip EOSC with a mirror of the Software Heritage universal source code archive.

Metadata Standards Crosswalk

- https://www.getty.edu/research/publications/electronic_publications/intrometadata/crosswalks.html

SZTAKI Cloud Reference Architectures

https://science-cloud.hu/en/reference-architectures

- VSCode development environment
- RStudio development environment
- Quantum
- MiCADO
- OpenVPN
- MongoDB cluster
- MariaDB cluster
- Slurm cluster
- Kafka cluster
- Horovod cluster: Horovod is a distributed deep learning framework for TensorFlow, Keras, Pytorch and Apache MXNet.
- JupyterLab development environment
- Kubernetes cluster
- Apache Spark Cluster (Python stack with Jupyter notebook and PySpark)
- DataAvenue
- Flowbster - Autodock Vina
- Setting up a Docker-Swarm cluster
- Launching Apache Hadoop cluster
- Launching Occopus cloud orchestrator
- gUSE (Grid and Cloud User Support Environment)

Provided as Ansible or Terraform templates for OpenStack

SZTAKI Cloud Reference Architectures - Technologies

https://git.sztaki.hu/science-cloud/reference-architectures

- OpenStack
- Terraform
- Ansible
- Apache Hadoop
- Jupyter Notebook
- R Studio
- VS Studio