Publications Office of the European Union

Initiative for digital transformation in the Metadata and Reference Data Sector of the Publications Office of the European Union

Asset Publication Lifecycle Enterprise Architecture

Disclaimer

This report was prepared for the Publications Office of the European Union by Infeurope.

Document metadata

Reference	WP 1.1: Asset Publication Lifecycle Enterprise Architecture
Corporate Author	Publications Office of the European Union
Author	Eugeniu Costetchi
Reviewers	Denis Dechandon and Willem Van Gemert
Contractor	Infeurope S.A.
Framework contract	10688/35368
Work package	WP 1.1
Delivery date	25 September 2020
Suggested readers	sector management, business and technical staff, stakeholders, enterprise architects, software developers, project partners

Abstract

Public procurement is undergoing a digital transformation. The EU supports the rethinking of the public procurement process with digital technologies in mind. This goes beyond simply moving to electronic tools; it rethinks various pre-award and post-award phases. The aim is to make them simpler for businesses to participate in and for the public sector to manage. It also allows for the integration of data-based approaches at various stages of the procurement process.

With digital tools, public spending should become more transparent, evidence-oriented, optimised, streamlined and integrated with market conditions. This puts eProcurement at the heart of other changes introduced to public procurement in new EU directives.

Given the increasing importance of data standards for eProcurement, a number of initiatives driven by the public sector, the industry and academia have been kick started in the recent years. Some have grown organically, while others are the result of standardisation work.

In this context, the Publications Office of the European Union aims to develop an eProcurement ontology.

The objective of the eProcurement ontology is to act as this common standard on the conceptual level, based on consensus of the main stakeholders and designed to encompass the major requirements of the eProcurement process in conformance with the Directives and Regulations.

This document provides a working definition of what is the architectural stance and the design decisions that shall be adopted for the eProcurement formal ontology along with the specifications how to generate comprising components.

Contents

1 Introduction

This document provides a working definition of what is the architectural stance and the design decisions that shall be adopted for the reference asset management lifecycle materialised as the publishing workflow used by the Standardisation Unit (SU) at the Publications Office of the European Union (PO).

In this document we (a) establish the baseline architecture, supported by the strategic and motivational information; (b) develop a target architecture guiding the transitional processes of implementing new technologies. This constitutes a natural evolution in response to changing mission needs defined by the SU management, and also takes into consideration the strategic directions proposed by PO and European Commission (EC) and European Parliament (EP).

1.1 Background considerations

Given the increasing importance of data standards for the European institutions, a number of initiatives driven by the public sector, the industry and academia have been kick started in the recent years. Some have grown organically, while others are the result of standardisation work. The vocabularies and the semantics that they are introducing and the technologies that they are using all differ. These differences hamper data interoperability and thus its reuse by them or by the wider public. This creates the need for a common data standard for publishing public reference data and models, hence allowing data from different sources to be easily accessed and linked, and consequently reused.

PSI directive [8] across the EU is calling for open, unobstructed access to public data in order to improve transparency and to boost innovation via the reuse of public data. The reference data maintained and published by the PO have been identified as data with a high-reuse potential [2]. Therefore, making this data available in machine-readable formats, following the data as a service paradigm, is required in order to maximise its reuse.

In this context the Publications Office of the European Union maintains and publishes an ever growing number of reference data assets vital in the context of inter-institutional information exchange. With regards to reference data, PO provides an ever growing number of services to the main institutional stakeholders and with the aim to extend them to a broader public; enabling active or passive participation in the reference data life cycle, standardisation and harmonisation.

1.2 EU trajectory towards semantic and linked data

European institutions set sail to adopt Semantic Web and Linked Data technologies as part of the vision to become data centric e-government bodies [cite find directives].

Many of the legacy systems in the institutions use XML data format for exchange and document formats governed by the XSD schemas. The aim is to evolve so that the existing and the new systems are capable to operate with semantic data representations using RDF [cite], OWL [cite], SHACL [cite] and other representations and serialised at least in RDF/XML [cite], Turtle [cite] and JSON-LD [cite] formats.

For this reason, the PO is already publishing the data in RDF format for over a decade using Cellar repository [cite]. And the SU, in particular, is committed to publish and disseminate the reference data in semantic formats. Next we describe the state of affairs of the SU to describe the context of the current work.

1.3 Target audience

The target audience for this document comprises the following groups of stakeholders:

- Management of the SU
- Enterprise architects and data governance specialists
- Documentalists involved in the reference data lifecycle
- Technical staff in charge of operating the workflow components
- Developers in charge of the workflow implementation

1.4 Document scope

This document aims to support SU in the transition towards the semantic technologies focusing on the architecture of the publishing workflow. The central use case is to support the asset management lifecycle presented in Section (below). It includes managing the incoming requests, editing the reference assets in VocBench3 system, then exporting the RDF data and passing them as input to a set of processes that validate, assess, transform, package and finally publish the assets in Cellar, the main dissemination platform.

This document will provide a motivational, business and application account of the workflow. Each of these accounts is limited strictly to the success scenario of the above mentioned use case and does not include possible extensions and variations which may be.

2 Publication workflow digital transformation

2.1 State of play

The SU publishes reference data in several formats, most important being XML, XSD and RDF/XML. On the technological side, the SU currently employs a legacy custom-built system for controlling and executing, in part, the asset management lifecycle operations (legacy workflow system). The system was developed using a mixture of XSLT technology, Perl and Bash scripting languages. The system was developed to execute a wide variety of conversions and transformations based on XML source files into various other formats including human readable documents.

The source data representation (XML in this case) has the primary role to serve as the only source of truth, and additionally, maintaining non-redundancy and rich expressivity. All other data forms and representations are secondary and are generated by transformation and conversion processes from the source representation.

One peculiarity of the legacy system setup is that the editing of the asset content is performed using Microsoft Excel. This is done by transformation of the content from XML representation into Excel stylesheets, which are edited by the SU documentalists, and then the stylesheets are converted back into XML form. This way a circular transformation is achieved which also serves as an integrity checking and validation mechanism. In addition XSD schema definitions are used to validate the XML source representation.

The legacy workflow system uses the file system for data persistence. In addition, this functionality is aided by a version controlling system, SVN [cite], to trace the temporal evolution of data.

Some steps in the legacy workflow are automated. The automation is based on cron tasks and SVN hooks that, upon changes in the source XML or Excel files trigger a set of conversion mechanisms. Some other steps require manual triggering and eventually parameterization intervention. The execution of the automated steps

often requires assistance of technical staff or IT skills above average which represents an impediment for the non technical documentalists and a hindrance for the IT staff.

Moreover, the maintenance of this system is burdened by a technical debt that accumulated over time, because the system evolved organically based on ever flowing requests.

2.2 Towards semantic technology workflow

SU's mission regarding the technological evolution is to migrate towards Semantic Web and Linked Data technologies and representations. The maintenance of the reference data is currently done based on XML source representation and the desired transition is towards RDF based representation. For that purpose Excel and XML sources are no longer suitable and a dedicated editor is necessary.

To solve this issue, SU took the development flagship of the VocBench3[cite] system - a web-based, multilingual, vocabulary editing tool based on the SKOS [cite] model, which is built on top of RDF/S standard. Later, VocBench3 was developed to support authoring of RDFS and OWL vocabularies.

Switching to RDF-based sources and adoption of VocBench3 system implies a technological and business process disruption. The main reason being the legacy workflow system, which operates with XML based sources only and does not support RDF sources. RDF representation being only a by-product derived from XML.

VocBench3 naturally adopted a persistence based on triple stores, which are NoSQL database systems implementing the directed graph data model instead of the hierarchical or relational model of data. The relational data model is mentioned here because the Excel worksheets are based on the tabular data organisation; the hierarchical data model is mentioned because the XML is fundamentally a hierarchically organised data structure; and each of them is only partially compatible with the graph paradigm present in semantic data models.

Migration towards a new workflow that integrated VocBench3 thus requires reconciliation between file-system and database approaches to persistence. Also a paradigmatic transition to graph based data representation, from the hierarchical models of source representation, and tabular models used for source authoring is necessary.

The legacy workflow system is also lacking in validation, structural analysis (fingerprinting) and content comparison capabilities (calculating the difference between

two versions of an asset). A transition would imply development of at least these new capabilities in order to maintain the current business processes.

3 Architecture building blocks

3.1 Methodology

In this document we take an enterprise architecture perspective and aim to provide several architecture aspects which are necessary and sufficient to describe the publishing process.

In developing this architecture, we adopt parts of the TOGAF [22], which is a framework for enterprise architecture that provides an approach for designing, planning, implementing, and governing an enterprise information technology architecture.

For the architecture representation, we adopt ArchiMate language [23], which is an open and independent enterprise architecture modeling language to support the description, analysis and visualization of architecture within and across business domains in an unambiguous way.

The motivation view is developed by implementing the following steps. First, TOGAF & ArchiMate were chosen as frameworks. Then some interviews were conducted with the SU management, having as a goal to elicit who are the main stakeholders and their motivations. The interview notes have been distilled and organised in ArchiMate diagrams (using EnterpriseArchitect Tool for drawing the models). Finally this report was written presenting the resulting motivation structure.

3.2 ArchiMate elements

Table 1: Overview of the relevant motivation elements [23]

Element	Definition	Notation
Stakeholder	Represents the role of an individual, team, or organization (or classes thereof) that represents their interests in the effects of the architecture.	

Table 1 continued from previous page

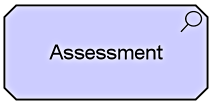
Driver	Represents an external or internal condition that motivates an organization to define its goals and implement the changes necessary to achieve them.	
Assessment	Represents the result of an analysis of the state of affairs of the enterprise with respect to some driver.	
Goal	Represents a high-level statement of intent, direction, or desired end state for an organization and its stakeholders.	

Table 2: Overview of the relevant business layer elements [23]

Element	**Definition**	**Notation**	
Business actor	Represents a business entity that is capable of performing behavior.		
Business role	Represents the responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event.		
Business collaboration	Represents an aggregate of two or more business internal active structure elements that work together to perform collective behavior.		
Business interface	Represents a point of access where a business service is made available to the environment.		
Business process	Represents a sequence of business behaviors that achieves a specific result such as a defined set of products or business services.		

Table 2 continued from previous page

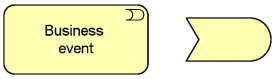
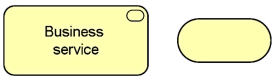
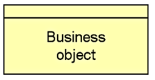
Business function	Represents a collection of business behavior based on a chosen set of criteria (typically required business resources and/or competencies), closely aligned to an organization, but not necessarily explicitly governed by the organization.	
Business event	Represents an organizational state change.	
Business service	Represents explicitly defined behavior that a business role, business actor, or business collaboration exposes to its environment.	
Business object	Represents a concept used within a particular business domain.	
Representation	Represents a perceptible form of the information carried by a business object.	

Table 3: Overview of the relevant application layer elements [23]

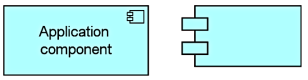
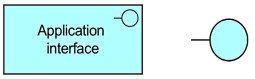
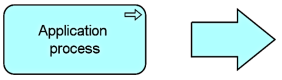
Element	**Definition**	**Notation**
Application component	Represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable.	
Application interface	Represents a point of access where application services are made available to a user, another application component, or a node.	
Application function	Represents automated behavior that can be performed by an application component.	
Application process	Represents a sequence of application behaviors that achieves a specific result.	

Table 3 continued from previous page

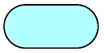
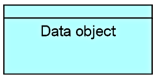
Application event	Represents an application state change.		
Application service	Represents an explicitly defined exposed application behavior.		
Data object	Represents data structured for automated processing.		

Table 4: Overview of the relevant technology layer elements [23]

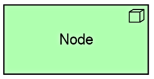
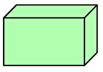
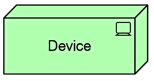
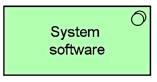
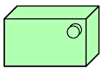
Element	**Definition**	**Notation**	
Node	Represents a computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources.		
Device	Represents a physical IT resource upon which system software and artifacts may be stored or deployed for execution.		
System software	Represents software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.		
Technology interface	Represents a point of access where technology services offered by a node can be accessed.		
Communication network	Represents a set of structures that connects nodes for transmission, routing, and reception of data.		
Technology service	Represents an explicitly defined exposed technology behavior.		
Artifact	Represents a piece of data that is used or produced in a software development process, or by deployment and operation of an IT system.		

Table 5: Overview of the ArchiMate relationships [23]

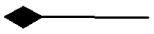
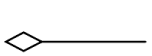
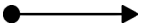
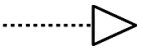
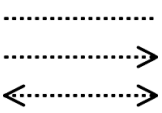
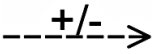
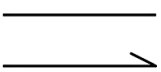
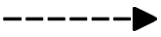
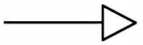
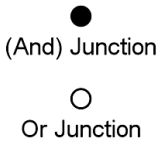
Element	Definition	Notation
	Structural Relationships	
Composition	Represents that an element consists of one or more other concepts.	
Aggregation	Represents that an element combines one or more other concepts.	
Assignment	Represents the allocation of responsibility, performance of behavior, storage, or execution.	
Realization	Represents that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.	
	Dependency Relationships	
Serving	Represents that an element provides its functionality to another element.	
Access	Represents the ability of behavior and active structure elements to observe or act upon passive structure elements.	
Influence	Represents that an element affects the implementation or achievement of some motivation element.	
Association	Represents an unspecified relationship, or one that is not represented by another ArchiMate relationship.	
	Dynamic Relationships	
Triggering	Represents a temporal or causal relationship between elements.	
Flow	Represents transfer from one element to another.	
	Other Relationships	
Specialization	Represents that an element is a particular kind of another element.	

Table 5 continued from previous page

Junction	Used to connect relationships of the same type.	

3.3 Architecture views

4 Motivation architecture

This section presents the motivation and goal structure of the Standardisation Unit. This will help determine the scope and rationale for the current architecture specification. The analysis of the motivation structure was not conducted in depth in order to constitute a decision making tool for the management but rather was aimed at accounting for the context of the publishing workflow which is the final goal of this architecture.

Nonetheless, this motivation view helps address the questions why a demand is meaningful, model crucial drivers and root causes behind the demand, actual goals and related outcomes, as well as concrete requirements for further development. In short it answers the questions to WHOM, WHY and WHAT.

4.1 Overall motivation structure

The structure of motivations, in ArchiMate, is hierarchically organised in several layers. For simplicity, we have chosen to use the top four layers: *stakeholders, drivers, assessments and goals*; leaving out the *outcomes*, *principles* and *requirements*. Figure 1 depicts the organisation of the motivation architecture. The structure starts at the top with enumerating the stakeholders, who are individuals, teams or organisations that represent their interests in the effects of the architecture [23].

Stakeholders have associated interests, concerns or drivers, which represent internal or external conditions that motivate an organisation to define goals [23].

Assessments represent results of analysis of the state of affairs with respect to some driver. They reveal strengths and weaknesses, opportunities or threats to an area of interest [23].

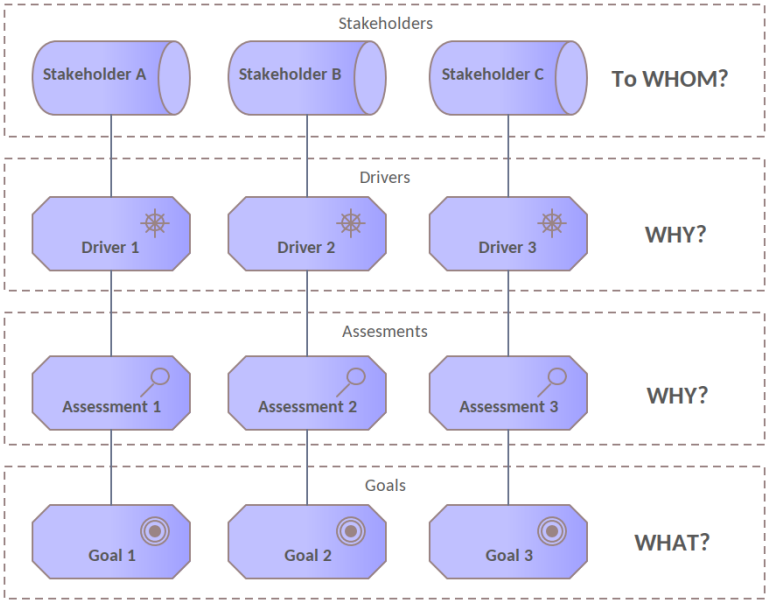


Figure 1: The layered motivation structure

Assessments are associated with goals, which represent a high level statement of intent, direction to desired end state for an organisation and its stakeholders [23].

Next we present the SU motivation structure spread over several sections.

4.2 Stakeholders and their roles

The Standardisation Unit involves multiple stakeholders. We can enumerate them but the list will be long and outside the scope of this exercise. Instead, we highlight the most important ones and in addition we group them based on the role they play in interaction with SU. In Figure 2 the roles are depicted as aggregate stakeholders in a grouping frame in the middle of the figure. Above the roles are placed the most important external stakeholders while below are enumerated the stakeholders from the PO.

The most important external stakeholders are: European Commission together with the Secretary General and all of the Commission's Directorates, the Inter-institutional Metadata and Formats committee (IMFC), the EuroVoc Committee (Group interinstitutional Lex (GIL)-subgroup EuroVoc), EU member states repre-

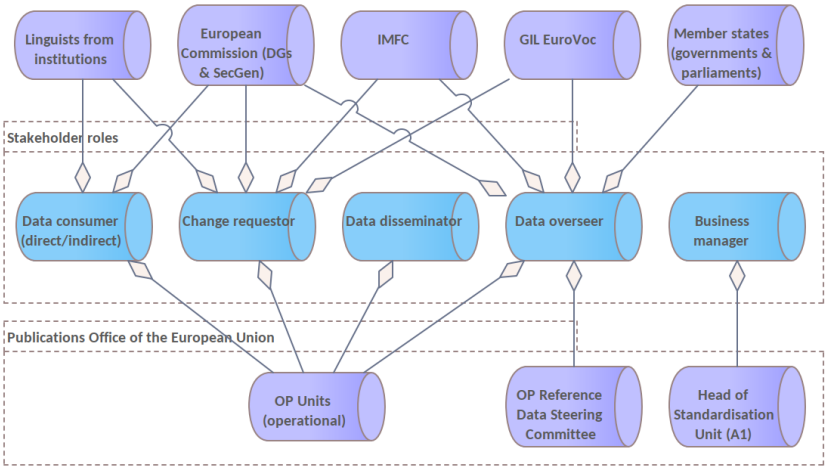


Figure 2: The layered motivation structure

sented by their governments and parliaments, and Linguists from different institutions and with a particular interest to the IATE project.

In the Figure 2 the OP stakeholders are placed in organisation bounding context, as the SU is a part of the PO and so its sibling units are not entirely external but members of the same organisation. The stakeholders within PO are the various units that use the reference data (e.g. Cellar team, OP Portal team, EurLex team, and others). A Reference Data Steering Committee is planned to be formed in the near future in order to coordinate and harmonise the published reference data. And finally the Head of Standardisation Unit who is in charge of running the enterprise.

In order to easier account for the stakeholders drivers, interests and goals, we grouped them based on their roles in interaction with SU. In Figure (above) the roles are depicted as aggregate stakeholders in a grouping frame in the middle of the figure. The roles are: data consumers, data requestors, data disseminators, data overseers and business managers. Next we describe each of the roles and then briefly enumerate the stakeholders.

The *consumers* of assets are the users that directly engage with the published assets (direct consumer) or the users of applications and services that are making use of the published assets (indirect consumer).

The *change requesters* are the agents that need and therefore requires particular content to be available as reference data.

The *data disseminators* are services and platforms where the reference assets are published for broad public consumption.

The *data overseers* are the agents that ensure that the content satisfies business needs is harmonised, coherent and complete. It is also responsible for the content correctness and harmonization among multiple stakeholders and its usefulness in broader context of application. Usually the role of data overseers is played by the standardization committees, steering committees and data stewards at large.

4.3 Drivers: primary

We have identified four *primary drivers*, three *secondary drivers* and two *internal efficiency drivers*. The distinction between primary and secondary drivers is based on whether the driver is shared between the external and internal stakeholders (in this case the business management). Figure 3 depicts the main stakeholders and their concerns, where the business manager, in this case head of the SU, has the same primary concerns as the main stakeholder roles.

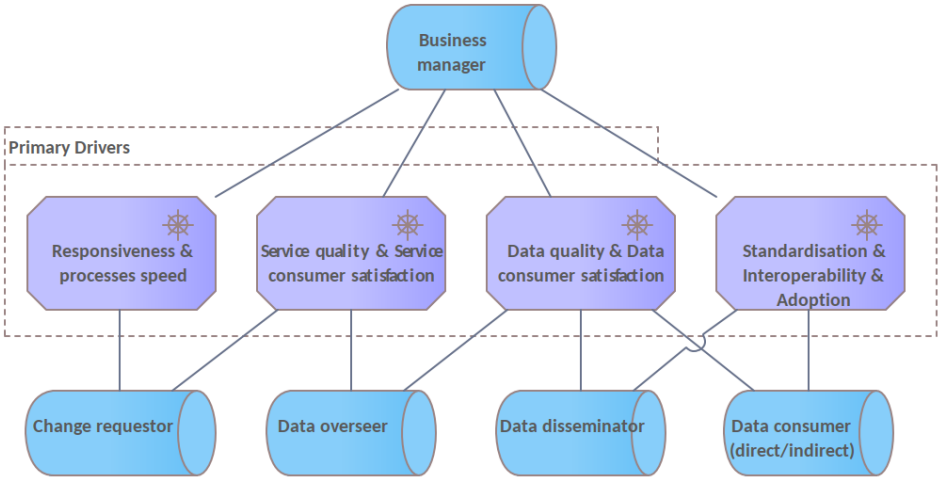


Figure 3: Primary drivers, motivating both, the internal and external stakeholders

For change requesters, the interaction *responsiveness and the speed of the asset life-cycle process* is of primary concern. The sooner the requests are processed and analysed the sooner they can be implemented, processed and published. The goal

of the SU is to reach the swiftness of publishing overnight change requests, as compared to the current situation when four major publications are scheduled per year allowing also a few urgent ones in between.

The *quality of service* provided by the SU at large and the service consumer satisfaction is a direct concern for the change requesters and data overseers as primary users various SU services.

The *quality of data* is of special interest for the data overseers as they are directly responsible for this aspect and implicitly of the data consumer satisfaction. The data quality here has a wide meaning covering aspects of formal, semantic and conceptual correctness while also being timely and up to date with the business. Besides the data overseers, the data users are also interested in high quality reference data. The data disseminators are indirectly affected by the quality of the data they distribute and share this interest to a lesser degree.

The last of the primary drivers is the *standardisation, interoperability and adoption*, which is a major concern for the data disseminators and the data users. This driver covers the adoption of widely used meta-models, formally well defined models representing shared conceptualisation of major bodies and organisations, usage and implementation of national and international standards proposed by the standardisation bodies (e.g. ISO, W3C, OMG). These standards refer not only to aspects of data representation, but also to protocols, exchange schemas, validation mechanisms and other tools facilitating systemic interoperability.

4.4 Drivers: secondary

The secondary drivers are those that are important to either external stakeholders ro the internal ones alone.

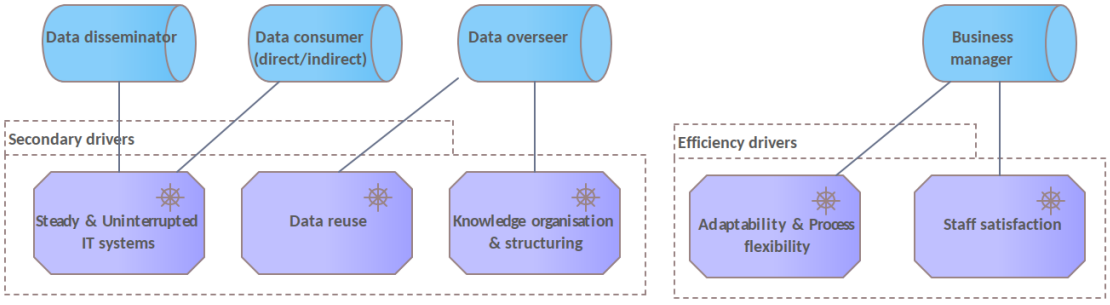


Figure 4: Secondary drivers, motivating either internal or external stakeholders

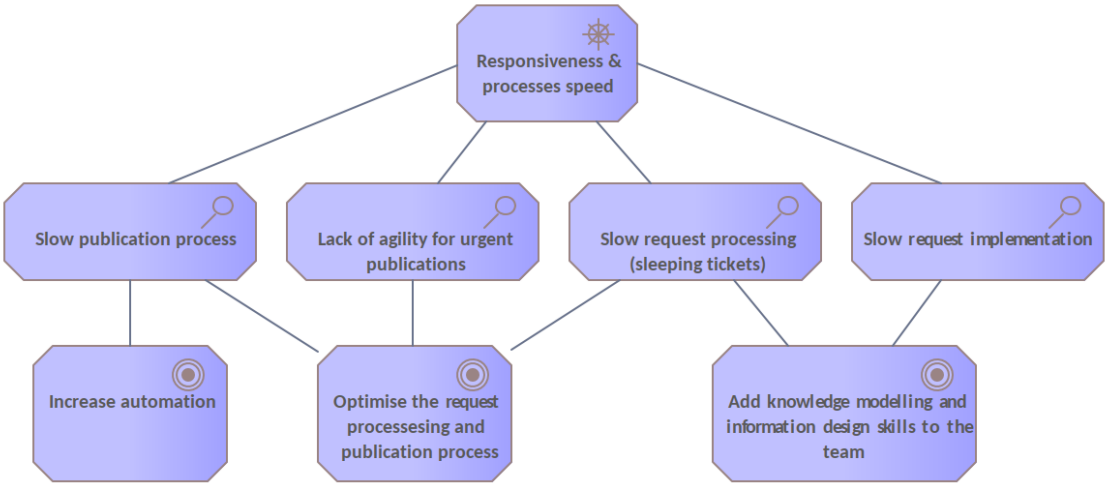


Figure 5: The assessment of the responsiveness and processing speed driver

4.5 Assessment: Responsiveness and processing speed

4.6 Assessment: Data quality

4.7 Assessment: Service quality

5 Business use cases

This section presents the core business use cases that have been identified in discussions with the SU team members. These use cases have been structured in the light of the new asset lifecycle process, and not the current one, even though they are

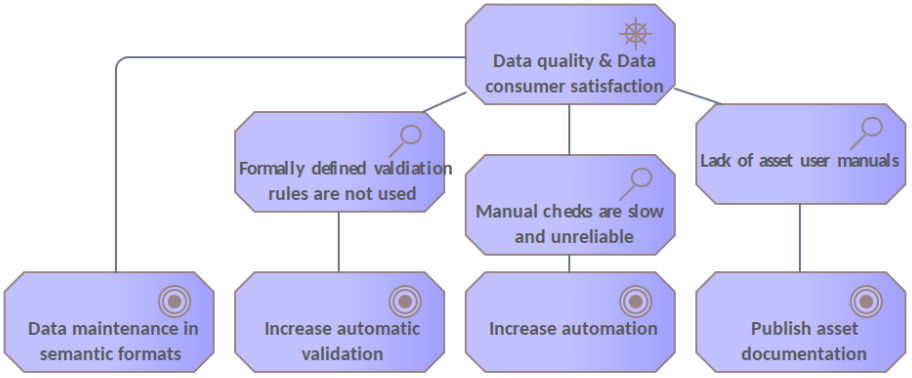


Figure 6: The assessment of data quality and data consumer satisfaction

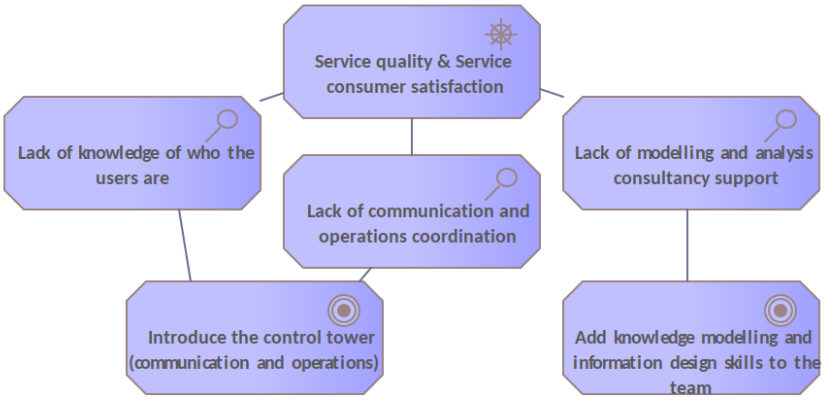


Figure 7: The assessment of service quality and service consumer satisfaction

heavily inspired by the current asset lifecycle process. Section 6 will present the designs of the current and the new asset lifecycle processes illustrating differences and commonalities. The designed processed are a derived from the use case descriptions following below.

5.1 UC1: Evolution: Register request for content change

Trigger: A request for content change is received in the functional mailbox

Success guarantee: A complete and clear change request case is created and scheduled for approval.

Main success scenario:

1. The client requests a change of one or multiple concepts in an asset
2. The request manager creates a new change request case and acknowledges the client
3. The asset manager analyses the request (in terms of business needs and in term of data management implications) and summarises the case
4. Request manager informs the client of the case summary
5. The request manager proposes the case for discussion in the next meeting of the team steering committee

Extensions:

4a The request is incomplete or unclear:

- 4a1 The asset manager formulates the information needs
- 4a2 The request manager collects the needed details and clarifications from the client
- 4a3 Return to step 3

4b The request is complex and needs deeper conceptual analysis and modelling/design:

- 4b1 The asset manager presents the case to the knowledge modelling expert
- 4b2 The knowledge modelling expert analyses the case and proposes a solution
- 4b3 Return to step 3

5.2 UC2: Evolution: Register request for content change

Trigger: A team steering committee meeting takes place

Preconditions: A case is in the meeting agenda

Success guarantee: The case is rejected or approved for implementation

Main success scenario:

1. Any time between the case is proposed for discussion and the meeting, committee members may assess the open cases and add business, technical or implementation related comments.
2. During the meeting, the asset manager presents the case.
3. The steering committee members discuss, comment the case
4. The steering committee approves the case for implementation
5. Asset manager schedules the case for implementation

Extensions:

4a The case is rejected:

- 4a1 The steering committee reject the case along with a justification
- 4a2 The request manager informs the client and provides recommendations

4b The case needs additional input:

- 4b1 The steering committee reject the case along with a request for action, information or agreement to an alternative proposal
- 4b2 The request manager informs the user and requests additional actions, information or agreement to an alternative proposal
- 4b3 The request manager register request for content change (UC1)

5.3 UC3: Implementation: Implement request for content change

Trigger: A case is scheduled for implementation

Preconditions: The case is approved for implementation

Success guarantee: The case is implemented and validated, while the data are exported and stored in the common repository

Main success scenario:

1. The request manager schedules a case for implementation

2. Data authoring officer reads the case and executes the content authoring accordingly
3. Data authoring officer automatically or assisted by the data processing officer
   (a) exports the asset from the authoring tool and
   (b) runs the SHACL validation for conceptual and structural issues and
   (c) runs the difference calculation between exported content and the previous release export and
   (d) runs the fingerprint calculation for the exported content
   (e) the data and reports are stored in the common repository
4. Data authoring officer checks
   (a) (verification) that the diff report calculated between the previous release export corresponds to the implemented change request[1].
   (b) (validation) that no structural anomalies are present in the fingerprint and validation reports
5. Repeat steps 1 - 4 until all cases are implemented for the asset
6. Data authoring officer informs the quality assurance officer of the successful implementation of all cases.

Extensions:

2a Translations are necessary:

   2a1 Additionally, data authoring officer manages the necessary translations and proof-reading (process described elsewhere: export selected data for translators, send to the translation unit, import updated data containing the translations, validate and proofread the translations)

4a Implementation or data is invalid:

   4a1 Data authoring officer collects and documents all the issues

   4a2 Return to step 2

[1]This is to validate that the export reflects change request case for the change request ticket, keeping the editors on the safe side. If all is good, then this is the final diff.

5.4 UC4: Validation: Validate the request for change

Trigger: An implementation is scheduled for validation (data available in SRC-AP format along the assessment reports)

Preconditions: All the cases are implemented for an asset and no further updates are foreseen

Success guarantee: The data are validated by a second pair of eyes and marked as fit for publication

Main success scenario:

1. Data authoring officer provides the data and validation reports in the common repository
2. Quality assurance office verifies that the diff report corresponds to case requirements
3. Quality assurance office checks the fingerprint and validation reports for semantic or structural anomalies.
4. Quality assurance office accepts the implementation and the data and informs the asset manager
5. Asset manager marks the asset as fit for publication.

Extensions:

4a Data quality issues are detected:

- 4a1 The quality assurance officer identifies and documents issues in the validation and fingerprint reports and informs the data authoring officer what the issues are and eventually explains how to fix them.
- 4a2 Data authoring officer implements the request for change (UC3)

4b Implementation issues are detected:

- 4b1 The quality assurance officer identifies and documents issues in the diff report and informs the data authoring officer what the issues are and eventually explains how to fix them.
- 4b2 Data authoring officer implements the request for change (UC3)

5.5 UC5: Release: Prepare the publication content

Trigger: Asset release is requested

Preconditions:

- The data is conceptually and formally validated (SRC-AP) and its content is fit to be published
- Code freeze is declared, no more changes are foreseen

Success guarantee: The data are available in standard (and where requested additional) forms and formats.

Main success scenario:

1. Asset manager requests a release
2. Data processing officer start the transformation processes from SRC-AP form into
   (a) Target forms: SKOS-AP-EU/Act/Core
   (b) Target formats: RDF/XML, Turtle, JSON-LD
3. Data processing officer runs the fingerprinting and SHACL validation for structural issues and confirms the transformation went well[2].
4. Data processing officer start the transformation processes from SRC-AP/SKOS-AP-EU into required additional forms and formats such as CAT-XML, XSD, Genericode, Excel/CSV, MarcXML, GeoJSON etc.
5. Data processing officer places the generated output into the common repository, along with the validation reports, and informs the asset manager and the publication officer

Extensions:

3a The validation reports reveal content related issues:
   3a1 The data processing officer identifies and documents the issues and reports them to the quality assessment officer

[2]This process is automatic and has the purpose of ensuring the transformation process passed correctly, keeping the data processing officers on the safe side.

3a2 Data authoring officer implements the request for change (UC3)

3b The validation reports reveal data related issues:

3b1 The data processing officer identifies and documents the issues and informs the quality assessment officer

3b2 The data processing officer fixes the issues due to the transformation process

3b3 Return to step 2

5.6 UC6: Publish: Publish a reference data asset

Trigger: A publication of selected assets is requested

Preconditions:

1. The selected assets, validated and converted into all the necessary forms and formats, are available in the common repository
2. Asset user manual is available in the common repository
3. Format user manuals are available in the common repository
4. Asset metadata, both content-related and technical, are available in the common repository

Success guarantee: The updated assets are accessible on the selected dissemination platforms and the broad public is informed about the new publication

Main success scenario:

1. The scheduled publication due date occurs
2. The publication officer generates the release notes from the diff-report that summarises what has changed (in more details than the impact assessment).
3. The publication officer generates the publication summary and impact assessment report (having sections customised for each major stakeholder) that presents an overview of the main content changes and if structural changes are included.

4. Asset manager checks the release notes and the impact assessment (that they reflect the change request cases)
5. Request manager sends the publication summary and impact assessment reports to the stakeholders, to inform them of upcoming changes and collect any pre-publication feedback.
6. Publication officer runs the packaging process for each asset (parallel to the impact assessment process) resulting in
   (a) Generation of additional technical metadata (DCAT, METS, IMMC, etc.)
   (b) Generation of packages (ZIP or other) for selected dissemination platforms (Cellar, ODP, Bartoc, Joinup, etc.) that contain all the necessary content, documentation and metadata
7. Publication officer tests the integrity/fitness of the generated packages by using the validation mechanisms offered by the dissemination platforms (validators or test dissemination environments)
8. Request manager receives the implicit acceptance of the impact assessment from the stakeholders (that is, no objections are raised during the established deadline) and informs the publication officer that the assets can be uploaded to the dissemination platform(s).
9. Publication officer publishes the packages to the dissemination platform, tests that the assets are accessible and informs the asset manager that publication is completed with success.
10. Request manager informs the broad public (including stakeholders) that the publication is complete.

5.7 UC7: Publish: Publish a model asset

Trigger: A publication of selected assets is requested

Preconditions:

- The selected assets, which were approved and converted into the standard forms and formats, are available in the common repository
- Asset user manual is available in the common repository

- Formats/representation user manuals are available in the common repository
- Asset metadata, both content-related and technical in the common repository

Success guarantee: The assets are accessible to broad public on the selected dissemination platforms

Main success scenario:

1. The scheduled publication due date occurs
2. The publication officer generates automatically the release notes, which summarises the content of the publication.
3. Request manager sends the publication summary to inform them of upcoming changes and collect any pre-publication feedback.
4. Publication officer runs the packaging process for each asset resulting in
5. Publication officer runs the packaging process for each asset (parallel to the impact assessment process) resulting in
   (a) Generation of additional technical metadata (DCAT, METS, IMMC, etc.)
   (b) Generation of packages (ZIP or other) for selected dissemination platforms (Cellar, IMMC, ODP, Wikidata, Bartoc, Joinup, etc.) that contain all the necessary content, documentation and metadata
6. Publication officer tests the integrity/fitness of the generated packages by using the validation mechanisms offered by the dissemination platforms (validators or test dissemination environments)
7. Publication officer publishes the packages to the dissemination platform, tests that the assets are accessible and informs the asset manager that publication is completed with success
8. Request manager informs the broad public (including stakeholders) that the publication is complete

Extensions:

6a Packages are rejected by the dissemination system:

6a1 The publication officer contacts the support team of the dissemination system and resolves the issue.

6a2 In case the generated package is incorrect, the publication officer corrects the generation processes.

6 Business architecture

This section covers in its extent the business architecture. The focus falls almost entirely on the bottom layer of the business architecture structure (see Figure 8) describing the internal processes, events and roles answering the questions who shall do what and when.

We address here both, the current organisation and the new organisation of the asset lifecycle process. First we establish a baseline representing the current setup and, second, how the new processes will look like in the light of digital transformations moving towards goals identified in the motivation structure (Section 4).

Next we explain the general idea of how to the business architecture is structured, which serves as an interpretation framework for the succeeding diagrams.

6.1 Prototypical business structure

Following the metaphor of layers presented in the motivation view, we decided to explain the organisation of business structure in terms of layers as well. Figure 8 depicts three layers with the most important elements of the business structure.

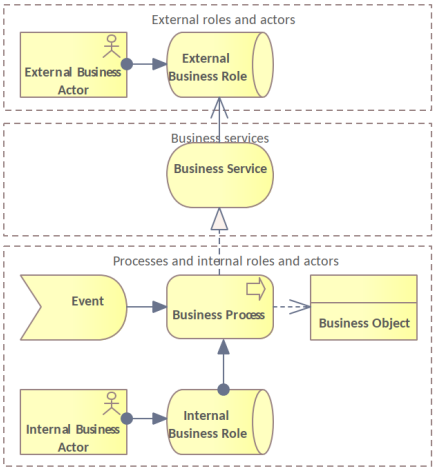


Figure 8: The prototypical business structure view

The topmost layer accounts for the external players or *actors*, which represent a business entity that is capable of performing behaviour, and *roles*, which represent skills and responsibilities for performing specific behaviour, and to which an actor can be assigned [23].

The middle layer represents the *services* that are offered by the organisation to the external players. A business service represents explicitly defined behaviour that a business role, business actor, or business collaboration exposes to its environment [23].

The lower layers accounts for the internal organisation in terms of *events*, *roles*, *processes* and *objects*. The business process represents a sequence of business behaviours that achieves a specific result such as a defined set of products or business services. The business event represents an organizational state change; while a business object represents a (passive) concept used within a particular business domain.

6.2 Current asset lifecycle stages

The current asset lifecycle process is organised in six stages: *inception (or evolution)*, *implementation*, *pre-release*, *release*, *publication* and *consumption*. Each of the stages represents a business sub-process. Figure 9 depicts the order in which stages flow and indicate that each stage process accesses a data asset, the central artefact in the diagram.

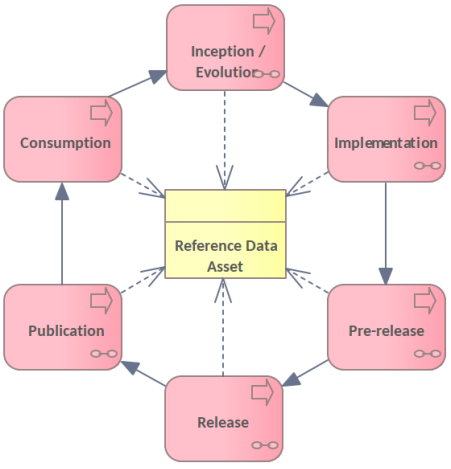


Figure 9: The current asset lifecycle stages

The *inception* stage means that a request arrives for creating and publishing a new data asset and it is being dealt with by the team. The *evolution* stage is similar only that the request is one for change of an existent data asset. There is no difference in the way these two requests are being treated and so the stage name is a conflation of the two. This stage also includes negotiating the request back with the client and then finally deciding and planning its implementation and publication.

The *implementation* stage deals with actually changing, authoring, converting (Excel to XML and back) and verifying the client request.

Pre-release marks that the content has been implemented accordingly and can be validated by a second pair of eyes implementing the *four eyes principle* implemented in SU. This verification and validation is performed by checking the validation reports and by comparing the difference between the current and the previous version of the asset conveyed in a diff report.

In the *release* stage the validated content is placed in a dedicated location of the common repository which indicated that the content is fit for publication.

The *publication* stage deals with packaging the content and disseminating it to the selected data disseminators, Cellar being the most important one. During this stage a set of announcements and communications ensure that the main stakeholders and the broad public are aware of the published new version of the asset.

Consumption stage is the one that happens outside the SU borders. It is the clients who use the data and then in the process come up with additional request for either changing existent assets or adding and publishing for new ones.

6.3 Actors and roles

This section describes the identified actors and roles relevant to the asset lifecycle process. Figure 10 depicts their relations.

Asset manager (a mix of *operational and business data steward*) is primarily responsible for data content, context, and associated business rules. This role is characterised by the full responsibility for the asset quality, enforcing policies and data governance processes, and ensuring asset fitness (both content and metadata) to the business needs. In the Standardisation Unit, this role is also responsible for high level interaction with the main stakeholders and important clients.

Team steering committee (also known as the team meetings) is a body composed of business, technical and analytical roles whose main purpose is to provide executive and operational guidance validating the business requests and assessing both the data management and the broader impact, determining the implementation priority, and promoting data governance and standardisation practices.

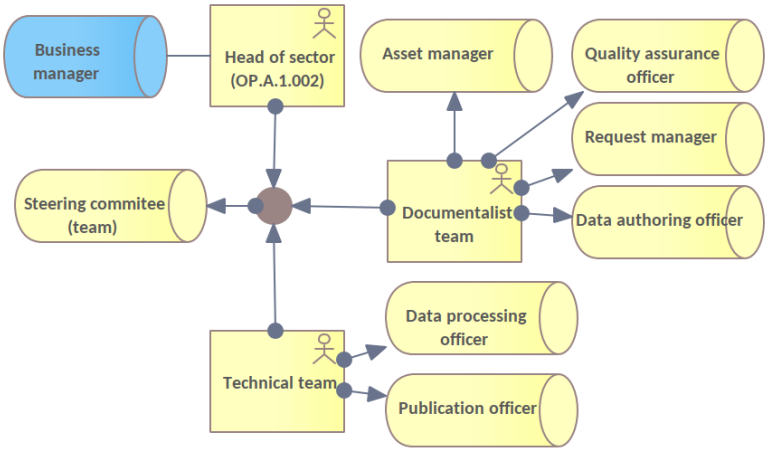


Figure 10: The internal roles in metadata and reference data sector

Request manager is the interface with the client collecting the change requests, assessing the business needs and translating them into data management requirements all being summarised and documented case by case.

Data authoring officer is responsible for editing data in a content management system implementing the cases prepared by the request manager. Quality assurance officers validate that the content implementation is correct from a technical and from a business point of view.

Data processing officer is a technical role that is responsible for preparing the assets for publication. The responsibilities include but are not limited to data storage, manipulation, automatic transformation, and generation of validation and assessment reports.

Publication officer is a technical role responsible for packaging and disseminating the assets to the specialised platforms

Stakeholder steering committee is a body representing the main clients and stake-

holders ensuring data and models harmonisation, alignment of the data management practices and adoption of international standards.

Client (*change requester* and *data user*) is a generic external role, who on one side consumes the data and services provided by the Standardisation Unit and on the other side demands publication of new assets or modification of the existing ones.

Data disseminator is an external role providing the Standardisation Unit with reliable data dissemination capabilities, which are meant to make the assets available for the clients.

The external roles and stakeholders have already been addressed in the motivations structure depicted in Figure 2. Each of these roles has a correspondent element in the business model and will be employed accordingly.

6.4 Current asset lifecycle overview

This section assembles the asset lifecycle process stages and the main internal roles together in an overview diagram depicted in Figure 11. It indicated what roles are assigned to which processes, along with cyclical depiction of the process sequence. Next we comment on the involvement of each role in the asset lifecycle process. All the lifecycle stages are internal to teh SU except for the last one, consumption, which takes place at the client premisses.

In the inception/evolution stage, the request manager is responsible for creating, documenting and ensuring descriptive completeness for requests arriving from clients to change existent assess or to create of new ones. These requests are managed as individual or, sometimes, interdependent cases. This role serves as the primary interface with the third parties. For this reason, in the publication stage, this role communicates with the stakeholders and broad public about the asset changes when it is published.

The asset manager is in charge of analysing and summarising the request case in the inception/evolution stage. This role intervenes in the pre-release stage to acknowledge that the case has been implemented and the asset is fit for publication; and in the publication stage to check the impact assessment and the release notes before they are used in the communication with external parties.

The team steering committee is involved in the initial stage only. After a new change request case is created, the team steering committee decides whether the case shall

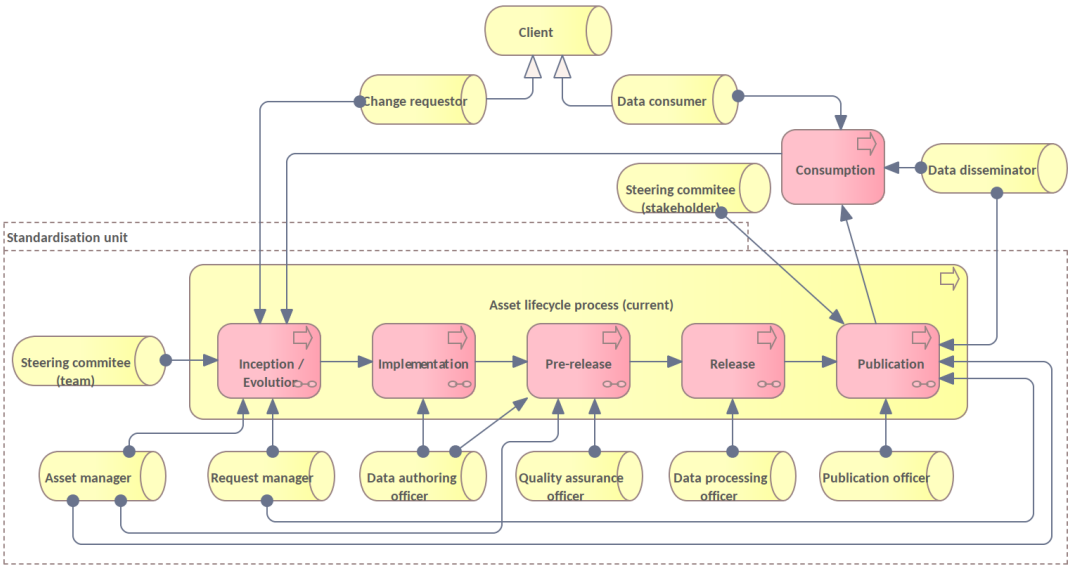


Figure 11: The current asset lifecycle stages and roles

be further processed; and if so, then the decision is about when and how.

The data authoring officer is responsible for the case implementation and, in the pre-release stage, verifying and validating its own work as the "first pair of eyes" (of the "two pairs of eyes" principle). The "second pair of eyes" verifying and validating the case implementation is enacted vy the quality assurance officer in the same pre-release stage.

Once the case is marked as fit for publication, in the release stage, it is placed automatically or by intervention of the data processing officer in a region of the common repository tagged for "release". If are any technical issues are encountered, the the data processing officers intervenes and fixes them.

In the publication stage, the publication officer generates the release artefacts, the release notes, packages the assets and disseminates them to the dissemination partners. External steering committees, such as IMMC metadata sub-committee, GIL EuroVoc an others are asked for final feedback two weeks in advance before the final dissemination.

Next, we turn to discuss the asset lifecycle stages in more detail as elicited from the technical and business teams of the SU. These descriptions are not covering the

ultimate details of the process but aim at describing the important building blocks of the current stages.

6.5 Current inception and evolution stage

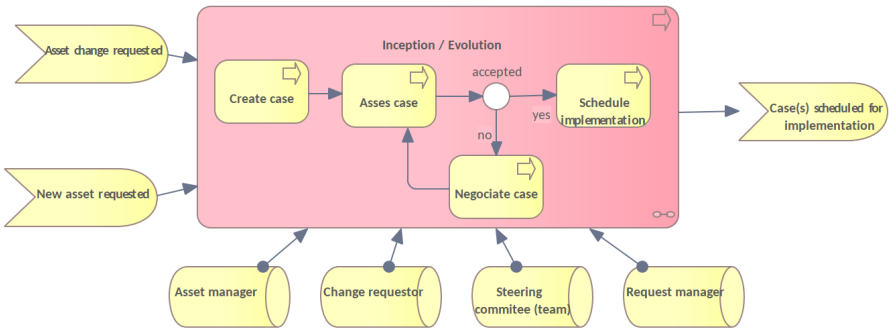


Figure 12: The current process for the inception and evolution stage

The process, depicted in Figure 12, starts when a request from a client arrives to either update a data asset or create a new one. This case, after being recorded, analysed and summarised, is asses by the asset manager and then discussed in the team meeting.

In case the case is accepted, then it is scheduled for implementation. Otherwise the case enters a so called negotiation stage, due to one of two things being the case: either the case is incomplete and more information is required from the client, either the case is unacceptable and a rejection is provided to the client with an explanation why or eventually with counter proposals.

The client communication is mainly carried out by email or telephone conversations, whereas the cases are managed using Jira ticket management system [3].

The process ends when the case is rejected or when it is scheduled for implementation.

6.6 Current implementation stage

Figure 13 depicts the current implementation process. It starts, when the case is queued for implementation. The data authoring officer modifies the asset content according to the instructions provided in the case description. The editing takes

place in an Microsoft Excel [14] workbook which represents an interface to the asset content. Excel is the main editing tool. Once the changes are complete, the workbook is committed into SVN repository [1], which triggers and automatic conversion of the Excel workbook into an XML form [16]. The XML is considered the primary asset source (structured with CAT XSD scheme). It is further converted back into Excel form, this way entering a conversion loop which also serves as validation mechanism ($XML \rightarrow Excel \rightarrow XML \rightarrow Excel$).

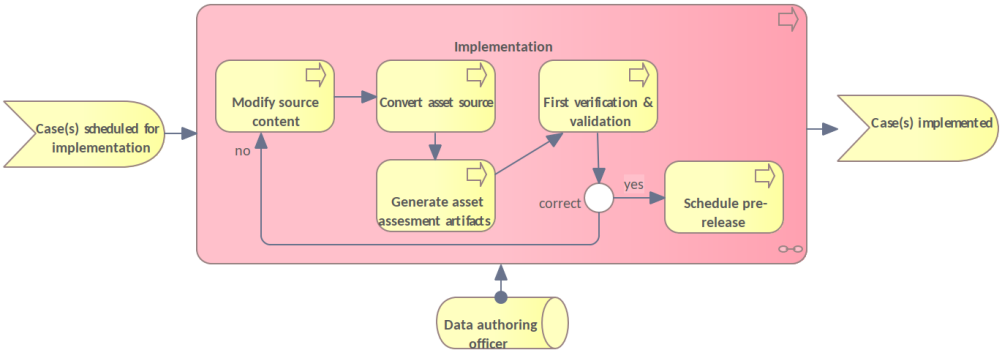


Figure 13: The current process for the implementation stage

Once the asset is converted into XML form, it becomes possible through a set of Perl scripts and XSLT style-sheets [11] to automatically generate assessment artefacts such as the diff report and schema validation report. The diff report indicates what changes have been done to the content between the previous and the latest version, while the validation report contains violations, if any, of XML structural constraints.

The editor then verifies the diff report to ensure the case implementation completeness and that the asset can be tagged for pre-release. Otherwise the content is being edited again.

The process end with the asset being marked for pre-release, which means that the case implementation is complete.

6.7 Current pre-release stage

The pre-release stage is depicted in Figure 14. Once the case is marked as implemented, and the assessment artefacts were generated after the conversion into

XML form, the second verification and validation can be performed by the quality assurance officer.

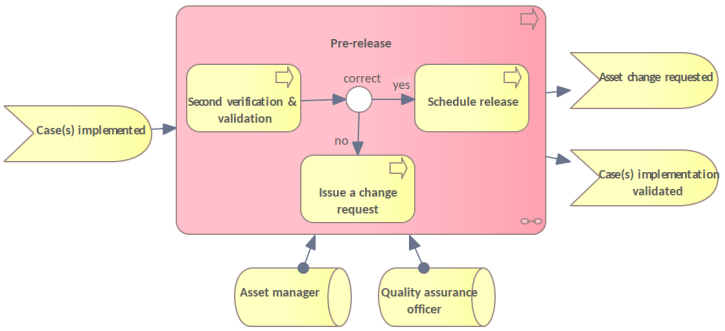


Figure 14: The current process for the pre-release stage

In case issues are identified in the case implementation, the quality assurance officer created a new request for changes and the case returns to the implementation stage, otherwise the case implementation is validated and the asset is marked as ready for release.

This is process where mostly manual steps are taken. Some minor content transformation can take place such as RDF content prettyification and an additional validation of record identifiers in the XML file. These these operations, however, are merely technicalities and do no have business relevance. Therefore they are omitted in the process diagram from Figure 14.

6.8 Current release stage

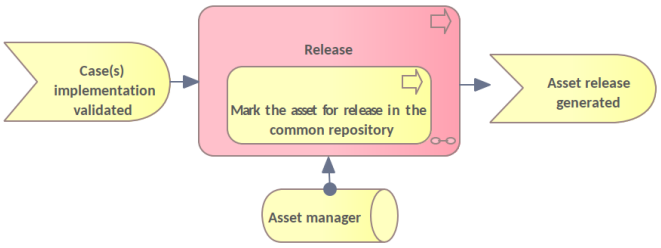


Figure 15: The current process for the release stage

The release stage is a symbolic step where an asset is simply marked as being in the release stage after it has been verified and validated by four pairs of eyes and confirmed that all the cases had been correctly implemented and that the asset is fit for release in the subsequent publication. This si depicted in Figure 15.

The release stage is realised by copying the updated version of the asset into a special area of the common repository and marked with the "release" tag.

6.9 Current publication stage

The current publication stage is depicted in Figure 16. It is a wide process that involves almost all the multiples roles.

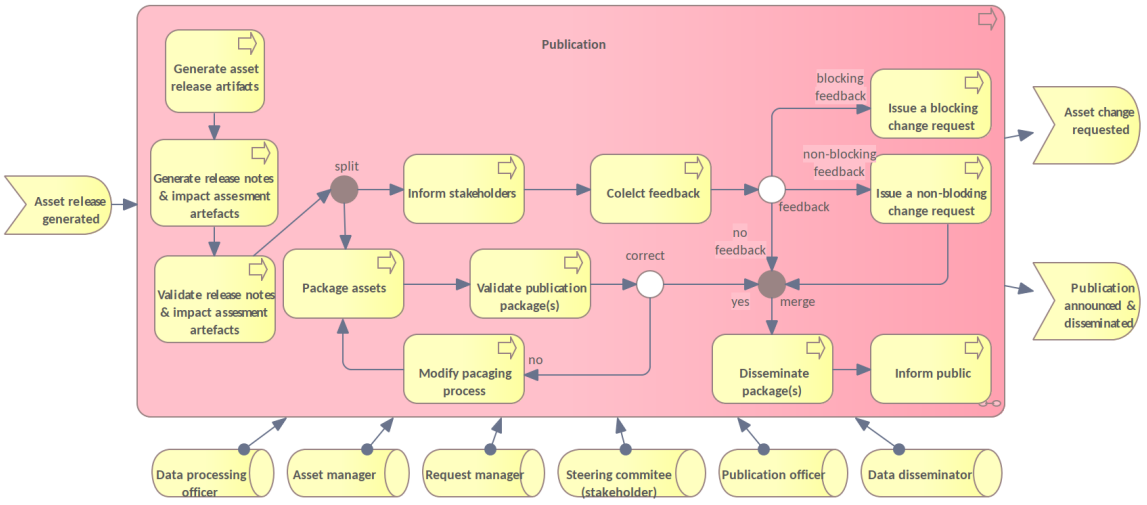


Figure 16: The current process for the publication stage

The publication process starts several weeks before the scheduled publication date, when a code freeze is announced and all the pre-selected assets are marked for release. It commences by generating from the CAT-XML source a selection of forms and formats to ease asset consumption by various clients. Mostly XSD, CAT-XML, SKOS-AP-EU forms are generated that are serialised in XML, Turtle and JSON formats. In addition some assets are also prepared in Genericode, Excel/CSV, MarcXML, GeoJSON and other forms.

Next, the publication notes are prepared together with a few different impact assessments specially prepared for targeted stakeholders. These communication artefacts

are checked for correctness and sent to the corresponding stakeholders. After a predefined number of days (usually two weeks) the feedback is collected, if any. Reception of no feedback is considered as a tacit acceptance of the current publication and the process can continue. Seldom, blocking or non-blocking feedback is received which leads to creation of new change requests, and depending on the situation, last minute changes are executed. Or, if the feedback is non-blocking, the intervention is scheduled or the next publication.

In parallel a packaging process is executed during which, the assets are prepared for dissemination. They are assembled in packages accompanied by their content-related and technical metadata, user manuals, format documentation and, of course, the asset itself expressed in all pre-generated forms and formats, the release artefacts. The packages are verified whether the target dissemination platforms accept them.

Finally the packages are disseminated and the successful publication is announced to the broad public. The dissemination is done primarily through Cellar[9] although other dissemination channels are also employed, among which Wikidata[24], Publications Office Open Data Portal (ODP), Bartoc [13], JoinUp platform [10].

This section completes the detailed presentation of the asset lifecycle process as it is today.

The digital transformation currently undertaken by SU management, that is in part targeted by this architecture, has impact on the asset lifecycle business, application and technical architectures. The next sections will describe how the new asset lifecycle architecture is envisaged.

6.10 New asset lifecycle overview

The structure of the new asset lifecycle process is depicted in Figure 17. In this new architecture, we aim to propose incremental changes, as to cause least disruptions to the team and the ongoing operations.

This process is similar, in its structure and stage names, to the current one: six stages circularly linked. The only change is the replacement of pre-release stage a stage called validation. There are however significant changes in the structure of three stages: implementation, validation and release, while the inception/evolution stage is identical to the current one, and the publication stage are is almost entirely unchanged. These changes are addressed in detail in the next sections.

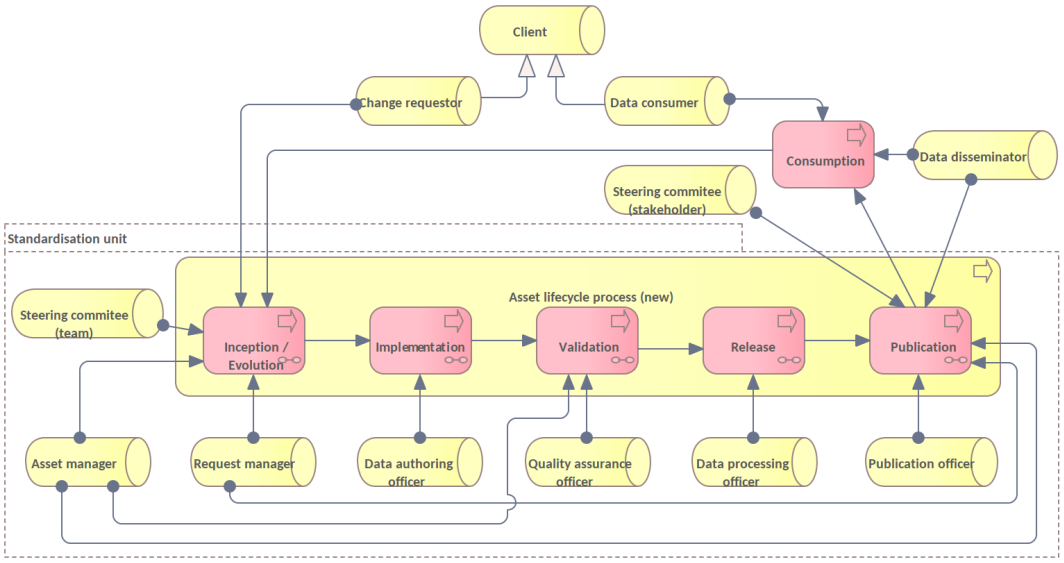


Figure 17: The current asset lifecycle stages and roles

There are no new roles in the new process, but there is a difference in the role allocation and involvement at different stages of the process as compared to the current one.

As mentioned above, the implementation/evolution stage is identical to the current one. So after the case has been registered and processed the implementation is scheduled. How it happens in the new workflow is presented in the next section.

6.11 New implementation stage

The new implementation process is depicted in Figure 19. The main digital transformation at this stage is adoption of RDF asset source, and the switch from Microsoft Excel as the content editor to VocBench3 [20, 21] semantic web editor.

The overall process looks very similar to the current one, but the employed technology for the source editing: VocBench3 and SKOS model(s) [15], render an entirely different editor experience when modifying the source content to implement the request case.

When the editing is complete, the content is exported from VocBench into the common repository and a set of asset assessment artefacts are generated. These

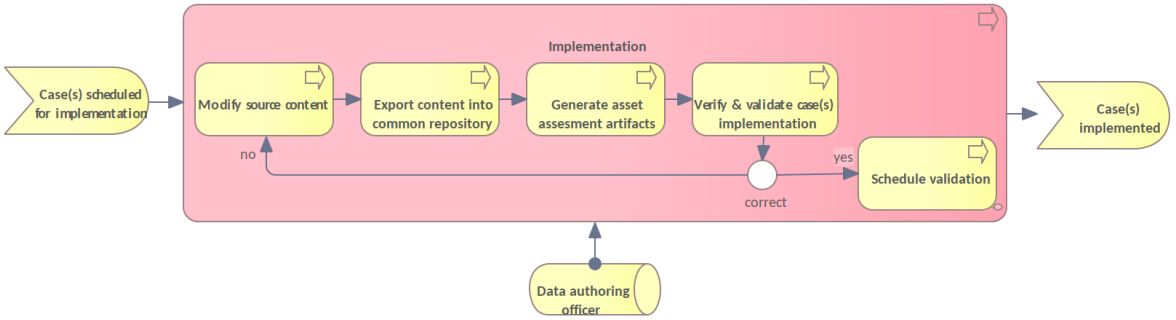


Figure 18: The new process for the implementation stage

artefacts are: the asset diff report, the fingerprint report and the validation report (based on SHACL [12] validation rules). These reports are no longer implemented based on the XSLT technology [11], because the underlying source is the new RDF representation [25] and no longer (CAT-)XML. More technical details are addressed in the application architecture in Section 7.5.

The editor (data authoring officer) then verifies that the case is implemented correctly, and if so a validation, by the quality assurance office is scheduled, otherwise, if some issues are spotted more editing is performed in VocBench3.

6.12 New validation stage

The validation stage is new in the asset lifecycle process. Here the correctness assessment is separated into two steps: verification and validation. This is a continuation of the current conception of how the case implementation is being assessed. The verification is primarily focused on the business aspects of the case implementation; whereas, the validation, deals with the technical and more formal aspects of the content structure and consistency. As the new technology allows for semantic accounts, then validation extends to deal with semantics as well.

In case errors are detected in the implementation, the quality assurance officer issues a change request and the case goes back into the implementation stage. Otherwise, the case implementation is considered correct and from now on a release can be scheduled. This is performed by the asset manager.

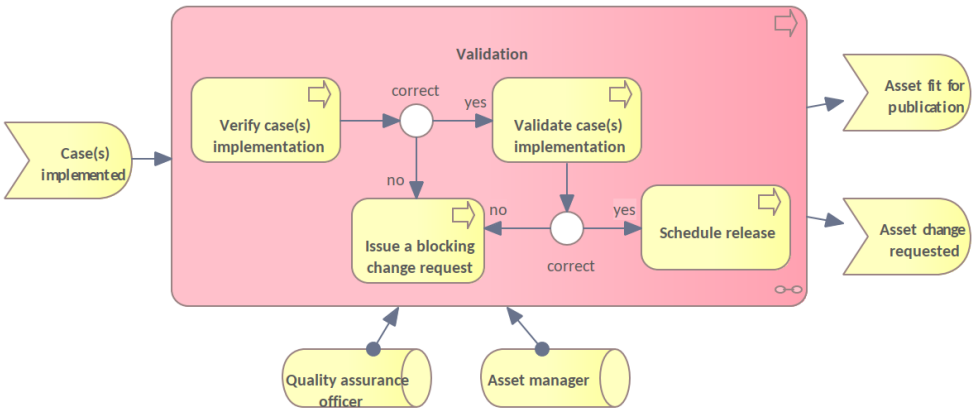


Figure 19: The new process for the validation stage

6.13 New release stage

The new release stage, depicted in Figure 20, differs considerably from the current one. This stage is no longer one of simply marking the asset implementation as fit for publication but deals with all the data transformations and preparation of artefacts necessary in the publication stage.

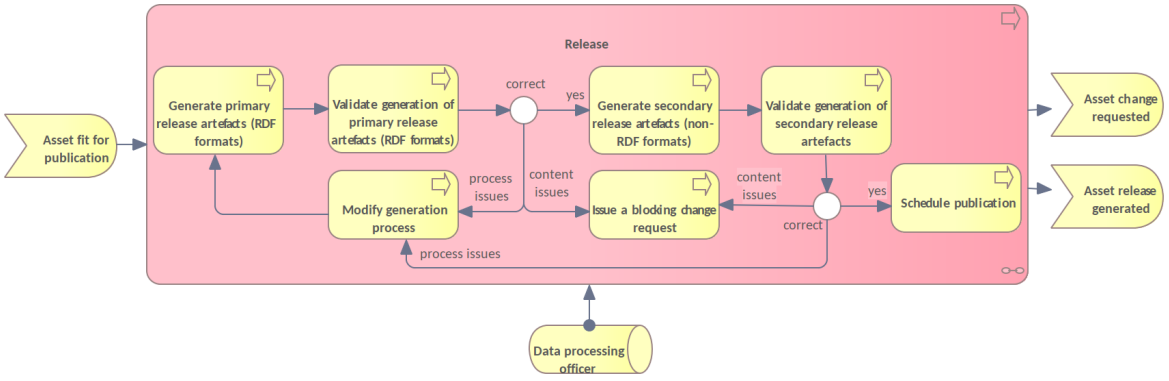


Figure 20: The new process for the release stage

The release starts when all the change request cases, in the scope of the scheduled publication, have been implemented and validated. Then, the primary release artefacts are generated from the RDF source expressed in SRC-AP form [6]. The main artefacts are the source in SKOS-AP-EU form [7], a special extension of the

SKOS-AP-EU which is necessary for publishing the content in Cellar because of the backwards compatibility issues (called SKOS-AP-EU-Act), and the SKOS core [15] representation. These artefacts are also generated in RDF/XML [17, 4], Turtle [5] and JSON-LD [18, 19] formats.

The transformation process is accompanied by a set of validation processes meant to safeguard the and ensure that the transformations are executed correctly. The validation reports are checked by the data processing officer. In case process related issues are detected, then most likely the processes contain bugs and need to be fixed. In case content related issues is spotted then a new change request is created and the process goes back to the implementation stage.

After the primary release artefacts are created, then secondary release artefacts are created. All of them are non-RDF forms and formats that currently are produced for clients and stakeholders and must continue so. An automated validation of the generated assets, to the extent possible, secures and ensures quality of the output. Like in the case of the primary artefacts, if the errors are detected then depending on their nature, either the data transformation process must be updated or a content change request is issued and the process returns back to the implementation stage.

The transformation technology employed here needs to provide ETL[3]/ELT[4]-like capabilities for both: RDF and for data formats. An extended discussion about the application capabilities is covered in Section 7.7.

At the end of this stage, all the important asset transformations and data conversions must be complete and the necessary forms and formats consumed by target clients shall be available for publication. Finally the assets are copied into the special place of the common repository available to the publication process.

6.14 New publication stage

The new publication stage process is depicted in Figure 21. This process is very similar to the one currently performed described in Section 6.9.

The main difference in this process is the starting point. If currently it starts by

[3]Extract, transform, load (ETL) is the general procedure of copying data from one or more sources into a destination system which represents the data differently from the source(s) or in a different context than the source(s).

[4]Extract, load, transform (ELT) is a variant of ETL where the extracted data is loaded into the target system first.

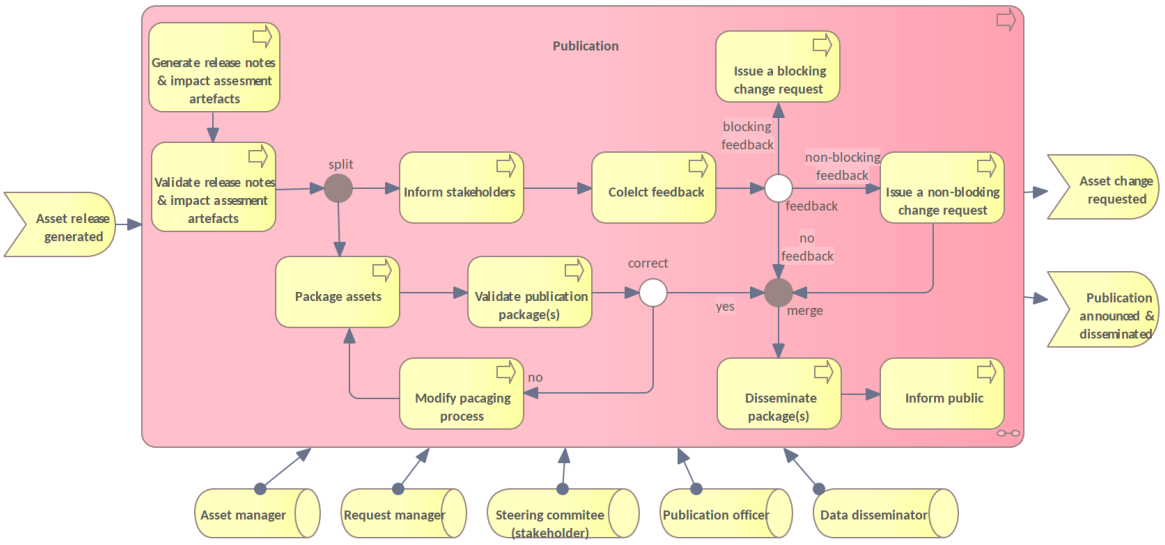


Figure 21: The new process for the release stage

transforming and converting the asset source into all forms and formats demanded by the clients, then the new process starts from the premise that the entire publication content has already been generated. What is left to be dealt with, at this stage, is the packaging and distribution of the publication packages to the data disseminators. And of course the same communication procedure with the stakeholders and the broad public.

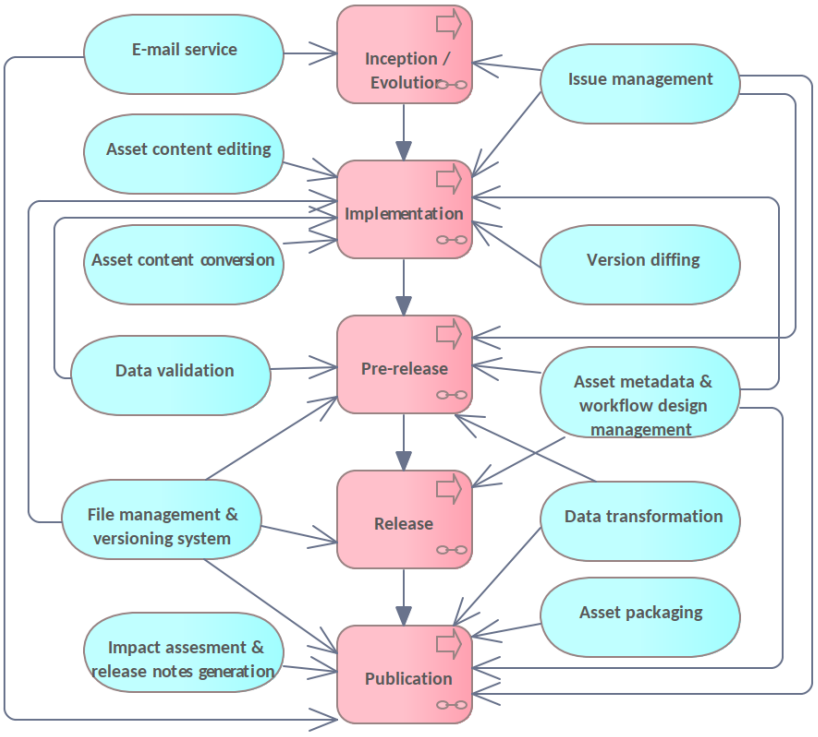


Figure 22: The application services that serve the current asset lifecycle

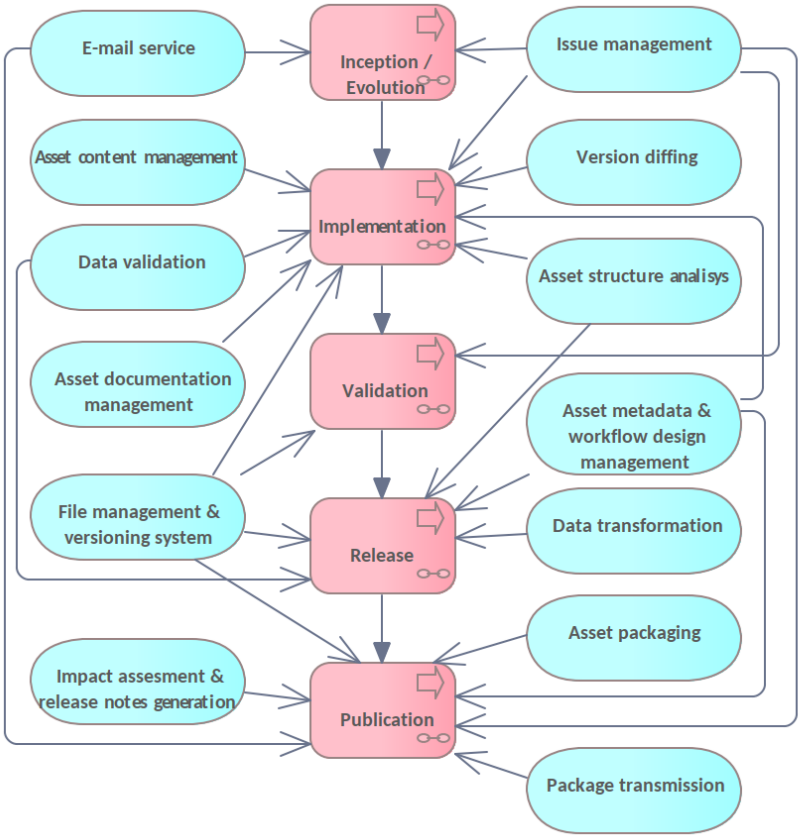


Figure 23: The application services that serve the new asset lifecycle

7 Application architecture

7.1 Prototypical application structure

7.2 Current application service architecture

7.3 New application service architecture

7.4 Inception and evolution services and components

7.5 Implementation services and components

7.6 Pre-release and validation services and components

7.7 Release services and components

7.8 Publication services and components

8 Technical architecture

9 Prototypical technology structure

10 Current technology architecture

11 New technology architecture

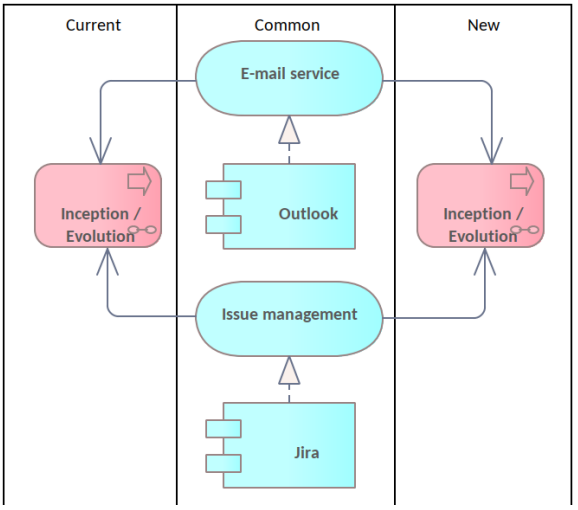


Figure 24: The application services and components that serve the current and new inception and evolution stage

leave a lasting impression. Do this, for example, by highlighting key points in your analysis or findings.

Summarizing your thoughts and conveying the larger implications of your study. The conclusion is an opportunity to succinctly answer the "so what?" question by placing the study within the context of past research about the topic you've investigated.

Demonstrating the importance of your ideas. Don't be shy. The conclusion offers you a chance to elaborate on the significance of your findings.

Introducing possible new or expanded ways of thinking about the research problem. This does not refer to introducing new information [which should be avoided], but to offer new insight and creative approaches for framing/contextualizing the research problem based on the results of your study.

Move towards becoming a data centric organisation Adopt a data governance strategy; implement data steward and data custodian roles; training and clear communication in place

Extend the enterprise architecture thinking: fully assess and document motivations and develop a strategy (i.e. include EC vision, EU Directives, data governance initiative, etc.); asses and document in detail who are the main classes of clients and stakeholders; fully assess and document the currently provided services and those

needed or requested by clients but not yet implemented; asses and document the technological capabilities and resources necessary to fulfill the service provisions; develop an implementation strategy to move from current state of affairs sto the desired one.

Adopt/implement a monitoring system (technical and business); further extend it with measurements of performance indicators

Further develop missing services and and increase integration and harmonisation of the employed micro services into a holistic reference data management system

Asses the current state of the reference data assets; their structure and content; implement/define clear model for each dataset

Clean up the datasets to provide internal consistency

12.1 Data stewards and custodians

Data stewards and custodians In Data Governance groups, responsibilities for data management are increasingly divided between the business process owners and information technology (IT) departments. Two functional titles commonly used for these roles are Data Steward and Data Custodian.

Data Stewards are commonly responsible for data content, context, and associated business rules. Data Custodians are responsible for the safe custody, transport, storage of the data and implementation of business rules. Simply put, Data Stewards are responsible for what is stored in a data field, while Data Custodians are responsible for the technical environment and database structure.

The data custodians are the agents that ensure that the content is coherent and complete. It is also responsible for the content correctness and harmonization among multiple stakeholders and its usefulness in broader context of application.

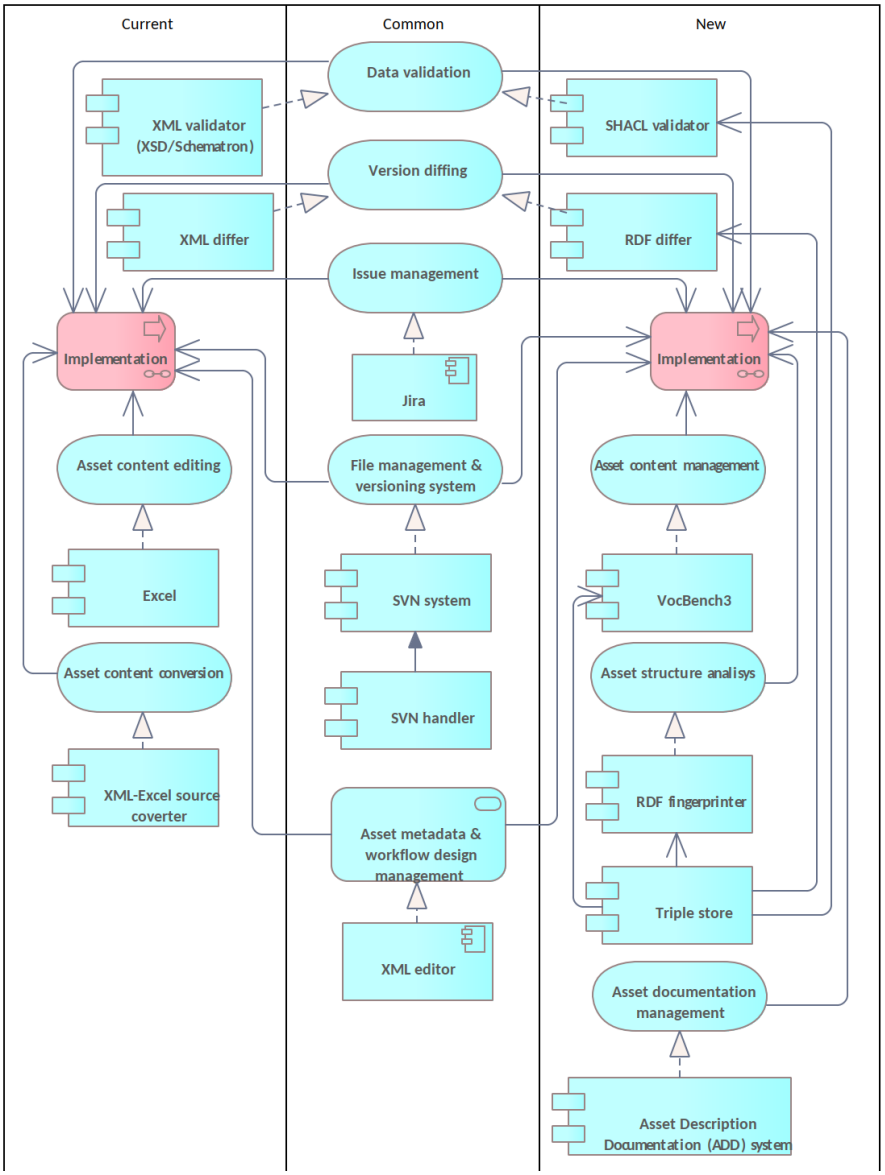


Figure 25: The application services and components that serve the current and new implementation stage

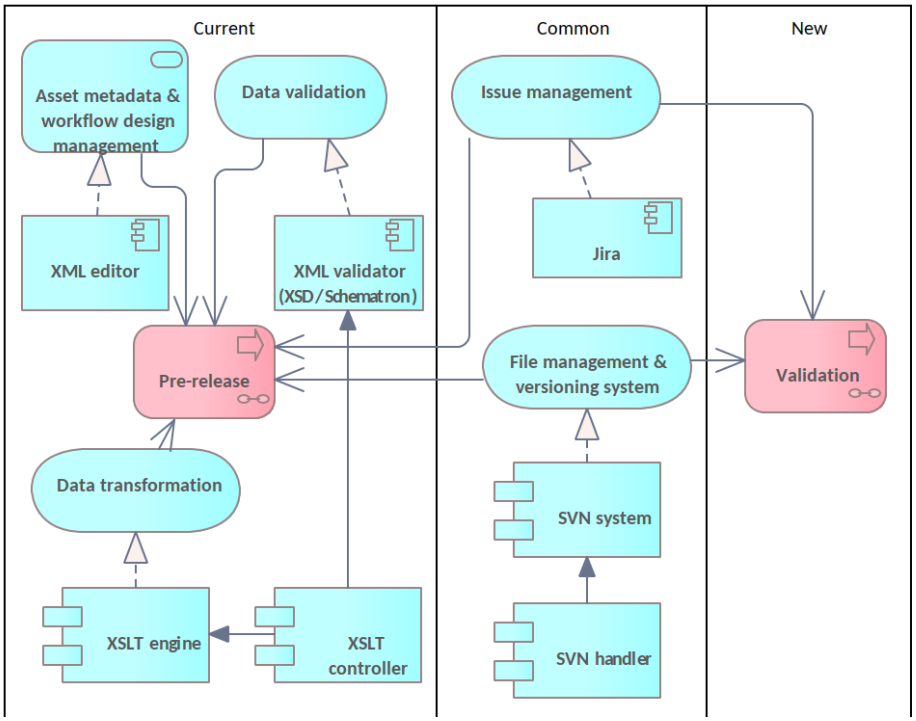


Figure 26: The application services and components that serve the current pre-release and the new validation stages

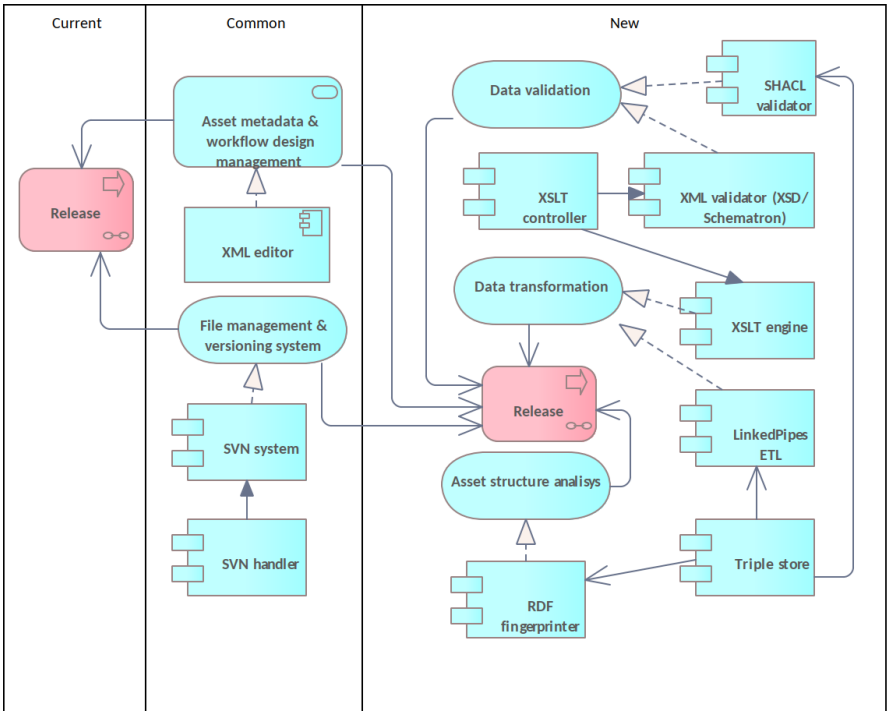


Figure 27: The application services and components that serve the current and new release stage

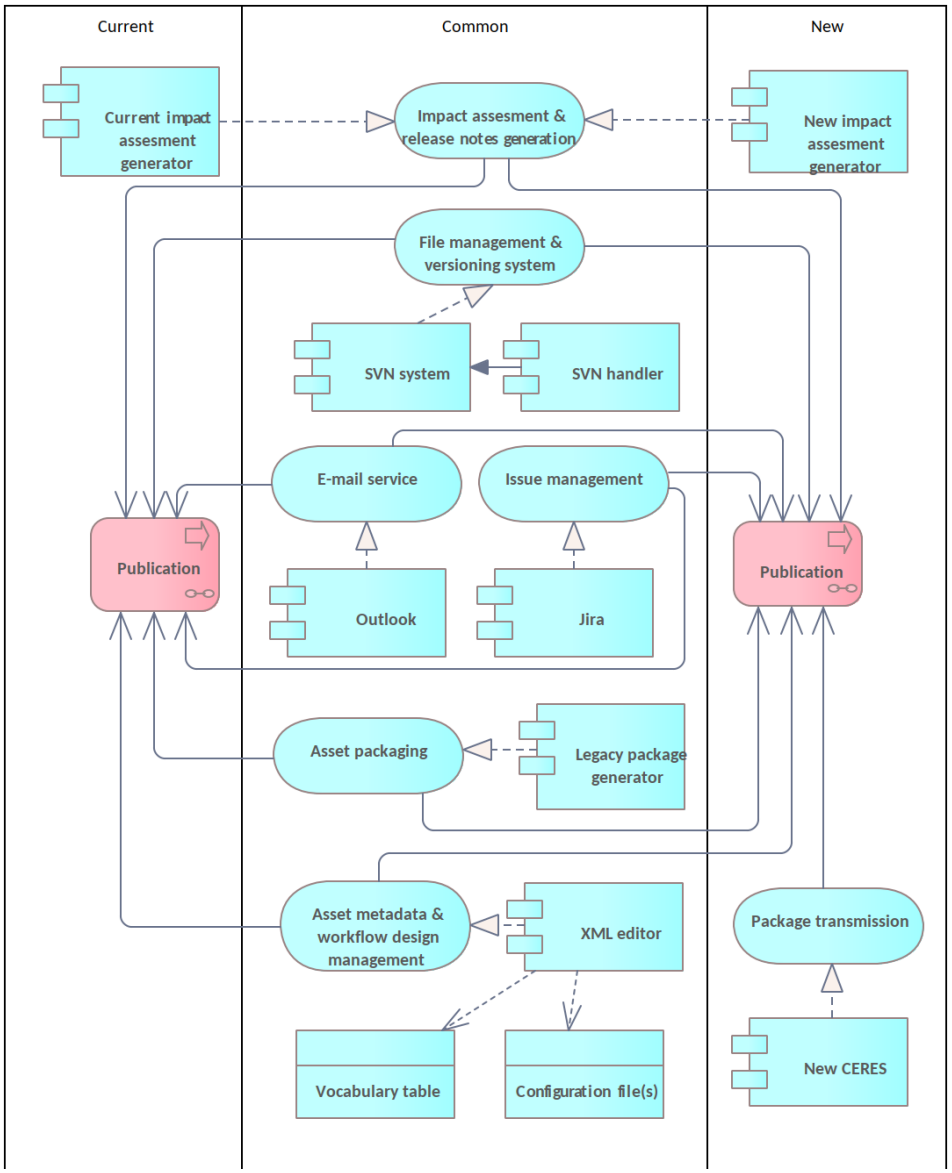


Figure 28: The application services and components that serve the current and new publication stage

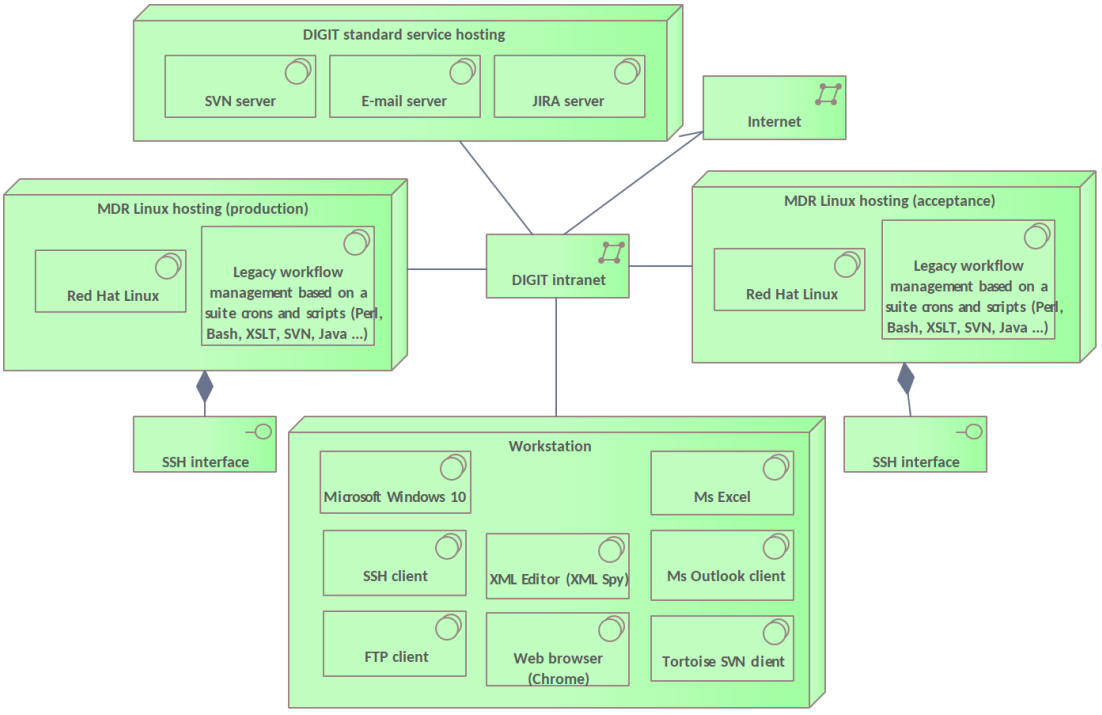


Figure 29: The technology structure that supports the current asset lifecycle

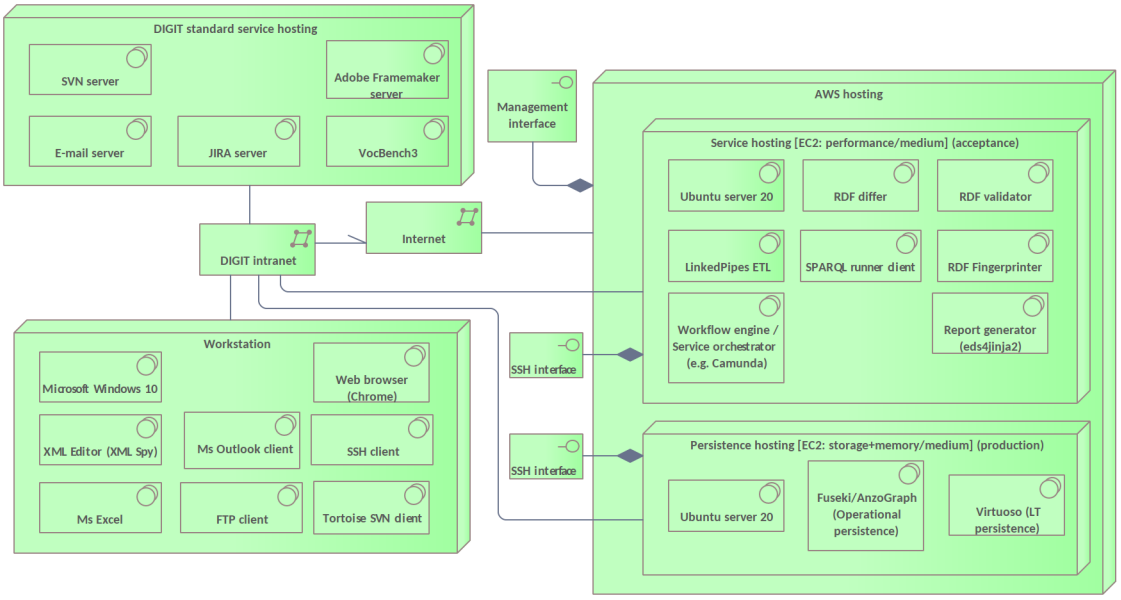


Figure 30: The technology structure that supports the new asset lifecycle

Bibliography

[1] Apache Software Foundation. Apache Subversion, 2000. URL `https://www.office.com/`.

[2] P. Archer, L. Bargiotti, M. D. Keyzer, S. Goedertier, N. Loutas, and F. V. Geel. Report on high-value datasets from eu institutions. Deliverable SC17DI06692, European Commission, 2014.

[3] Atlassian. Jira Software, 2002. URL `https://www.atlassian.com/software/jira`.

[4] D. Beckett. RDF/xml syntax specification (revised). W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/.

[5] G. Carothers and E. Prud'hommeaux. RDF 1.1 turtle. W3C recommendation, W3C, Feb. 2014. http://www.w3.org/TR/2014/REC-turtle-20140225/.

[6] E. Costetchi. SRC-AP-VB3: an application profile for metadata assets maintained in VocBench3. Recommendation, Publications Office of the European Union, September 2019.

[7] E. Costetchi and J. Delahousse. SKOS-AP-EU: an application profile for publishing SKOS-based metadata assets. Recommendation, Publications Office of the European Union, September 2019.

[8] European Parliament and the Council. Directive 2013/37/EU of the European Parliament and of the Council of 26 June 2013 amending Directive 2003/98/EC on the re-use of public sector information Text with EEA relevance. *OJ*, L 175: 1–8, 2013. CELEX:32013L0037.

[9] E. Francesconi, M. W. Küster, P. Gratz, and S. Thelen. The ontology-based approach of the publications office of the eu for document accessibility and open data services. In *International Conference on Electronic Government and the Information Systems Perspective*, pages 29–39. Springer, 2015.

[10] G. Hillenius. Free and open source software across the eu. *IFOSS L. Rev.*, 5: 153, 2013.

[11] M. Kay. XSL transformations (XSLT) version 3.0. W3C recommendation, W3C, June 2017. https://www.w3.org/TR/2017/REC-xslt-30-20170608/.

[12] H. Knublauch and D. Kontokostas. Shapes constraint language (SHACL). W3C recommendation, W3C, July 2017. https://www.w3.org/TR/2017/REC-shacl-20170720/.

[13] A. Ledl and J. Voß. Describing knowledge organization systems in bartoc and jskos. *e-prints in library and informations science*, 2016.

[14] Microsoft. Excel 2013, 2013. URL `https://www.office.com/`.

[15] A. Miles and S. Bechhofer. SKOS simple knowledge organization system reference. W3C recommendation, W3C, Aug. 2009. http://www.w3.org/TR/2009/REC-skos-reference-20090818/.

[16] J. Paoli, F. Yergeau, M. Sperberg-McQueen, T. Bray, E. Maler, and J. Cowan. Extensible markup language (XML) 1.1 (second edition). W3C recommendation, W3C, Aug. 2006. http://www.w3.org/TR/2006/REC-xml11-20060816/.

[17] G. Schreiber and F. Gandon. RDF 1.1 XML syntax. W3C recommendation, W3C, Feb. 2014. http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/.

[18] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, and N. Lindström. Json-ld 1.0. *W3C recommendation*, 16:41, 2014.

[19] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, P.-A. Champin, and N. Lindström. JSON-LD 1.1–a JSON-based serialization for Linked Data. W3C recommendation, W3C, 2020. https://www.w3.org/TR/json-ld11/.

[20] A. Stellato, A. Turbati, M. Fiorelli, T. Lorenzetti, E. Costetchi, C. Laaboudi, W. Van Gemert, and J. Keizer. Towards vocbench 3: pushing collaborative development of thesauri and ontologies further beyond. In *17th European Networked Knowledge Organization Systems Workshop, NKOS 2017*, volume 1937, pages 39–52. CEUR-WS, 2017.

[21] A. Stellato, M. Fiorelli, A. Turbati, T. Lorenzetti, W. van Gemert, D. Dechandon, C. Laaboudi-Spoiden, A. Gerencsér, A. Waniart, E. Costetchi, et al. Vocbench 3: A collaborative semantic web editor for ontologies, thesauri and lexicons. *Semantic Web*, (Preprint):1–27, 2020.

[22] The Open Group. *The TOGAF Standard, Version 9.2*. The Open Group, 2018.

[23] The Open Group. *ArchiMate 3.1 Specification*. The Open Group, 2019.

[24] D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[25] D. Wood, R. Cyganiak, and M. Lanthaler. RDF 1.1 concepts and abstract syntax. W3C recommendation, W3C, Feb. 2014. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.