D2.2.3 System Architecture & Specification v2.0

DURAARK

FP7 – ICT – Digital Preservation
Grant agreement No.: 600908

Date: 2014-01-31
Version 1.0
Document id. : duraark/2014/D.2.2.3/v1.0

Grant agreement number	:	600908
Project acronym	:	DURAARK
Project full title	:	Durable Architectural Knowledge
Project's website	:	www.duraark.eu
Partners	:	LUH – Gottfried Wilhelm Leibniz Universitaet Hannover (Coordinator) [DE] UBO – Rheinische Friedrich-Wilhelms-Universitaet Bonn [DE] FhA – Fraunhofer Austria Research GmbH [AT] TUE – Technische Universiteit Eindhoven [NL] CITA – Kunstakademiets Arkitektskole [DK] LTU – Lulea Tekniska Universitet [SE] Catenda – Catenda AS [NO]
Project instrument	:	EU FP7 Collaborative Project
Project thematic priority	:	Information and Communication Technologies (ICT) Digital Preservation
Project start date	:	2013-02-01
Project duration	:	36 months
Document number	:	duraark/2014/D.2.2.3
Title of document	:	D2.2.3 System Architecture & Specification v2.0
Deliverable type	:	Report
Contractual date of delivery	:	2014-01-31
Actual date of delivery	:	2014-01-31
Lead beneficiary	:	FhA
Author(s)	:	Jakob Beetz <J.Beetz@tue.nl> (TUE) René Berndt <rene.berndt@vc.fraunhofer.at> (FhA) Stefan Dietze <dietze@l3s.de> (LUH) Dag Fjeld Edvardsen <dag.fjeld.edvardsen@catenda.no> (Catenda) Ujwal Gadiraju <gadiraju@l3s.de> (LUH) Michelle Lindlar <michelle.lindlar@tib.uni-hannover.de> (LUH) Sebastian Ochmann <ochmann@cs.uni-bonn.de> (UBO) Martin Tamke <martin.tamke@kadk.dk> (CITA) Richard Vock <vock@cs.uni-bonn.de> (UBO)

Responsible editor(s)	:	René Berndt <rene.berndt@vc.fraunhofer.at> (FhA) Eva Eggeling <eva.eggeling@vc.fraunhofer.at> (FhA)
Quality assessor(s)	:	Stefan Dietze <dietze@l3s.de> (LUH) Michelle Lindlar <michelle.lindlar@tib.uni-hannover.de> (LUH) Raoul Wessel <wesselr@cs.uni-bonn.de> (UBO)
Approval of this deliverable	:	Stefan Dietze <dietze@L3S.de> (LUH) – Project Coordinator Marco Fisichella <fisichella@L3S.de> (LUH) – Project Manager
Distribution	:	Public
Keywords list	:	System Architecture

Executive Summary

This deliverable presents the second iteration on the system architecture of the DURAARK framework. It describes the philosophy, decisions, constraints, justifications, significant elements, and any other overarching aspects of the system that shape the design and implementation. It complements and refines the deliverable D2.2.2 System Architecture & Specification v1.0 from month 6.

Table of Contents

1 Architectural goals and philosophy 6
2 Use-cases revisited 9
3 Decisions, constraints, and justifications 13
3.1 Quality and organization of measured data 13
3.2 Quality and organization of BIM data 14
3.3 Use of Semantic Web and Linked Data standards 15
3.4 Digital preservation system: Rosetta 17
3.5 Search and Retrieval with PROBADO3D 19
3.6 Strategic decision on file formats 20
4 System architecture 23
4.1 Geometric Enrichment 26
4.2 Semantic Enrichment 37
4.3 Data preservation 49
4.4 Search & Retrieval: PROBADO3D 60
5 Conclusion 64
References 65


DURAARK
DURABLE
ARCHITECTURAL
KNOWLEDGE

1 Architectural goals and philosophy

The DURAARK system architecture is based on a component-based approach. An individual software component is a software package, a web service, a web resource, or a module that encapsulates a set of related functions (or data). All system processes are placed into separate components so that all of the data and functions inside each component are semantically related (just as with the contents of classes). Because of this principle, components are designed to be modular and cohesive. The component based approach allows the reuse of the developed components for the various use-cases described in D2.2.1. They can easily be rearranged to adopt to new usage scenarios, which will be identified during the project's duration.

Figure 1 illustrates an overview of the individual parts of the project that lead to the preservation of architectural information. The three main activities can be distinguished:

- **Geometric enrichment of data** which allows the inclusion of surveyed data in the form of point-clouds as additional as-built information beside existing explicit building information models or as the main form of geometric representation where such explicit models are not available at the time of archival.
- **Semantic enrichment of data** which allows the addition of information from various information sources to the building information models themselves as well as to the meta-data of the archival information package (AIP).
- **Preservation of data** which ensures that the information is available and useable for the designated community. It addresses the aspects of bit preservation, logical preservation and semantic preservation, which were described in detail in D2.2.1.

The functional and non-functional requirements have been described in deliverable D2.2.1. In the course of this document the current state on the system architecture will be described, which will be the basis for the first prototype in month 18.

Section 2 aligns the use-cases from D2.2.1 with the components to be developed within DURAARK. Based on the curation lifecycle model the use-cases are categorized by their role within preservation and how the components contribute to the various steps within the use-cases.

Based on the use-cases and requirements from D2.2.1 the decision, constraints and justi-

fications are described in section 3. These will serve as guidelines for defining architecturally significant parts of the system.

The system architecture is described in section 4 providing a detailed description about the various components and their interaction among each others.

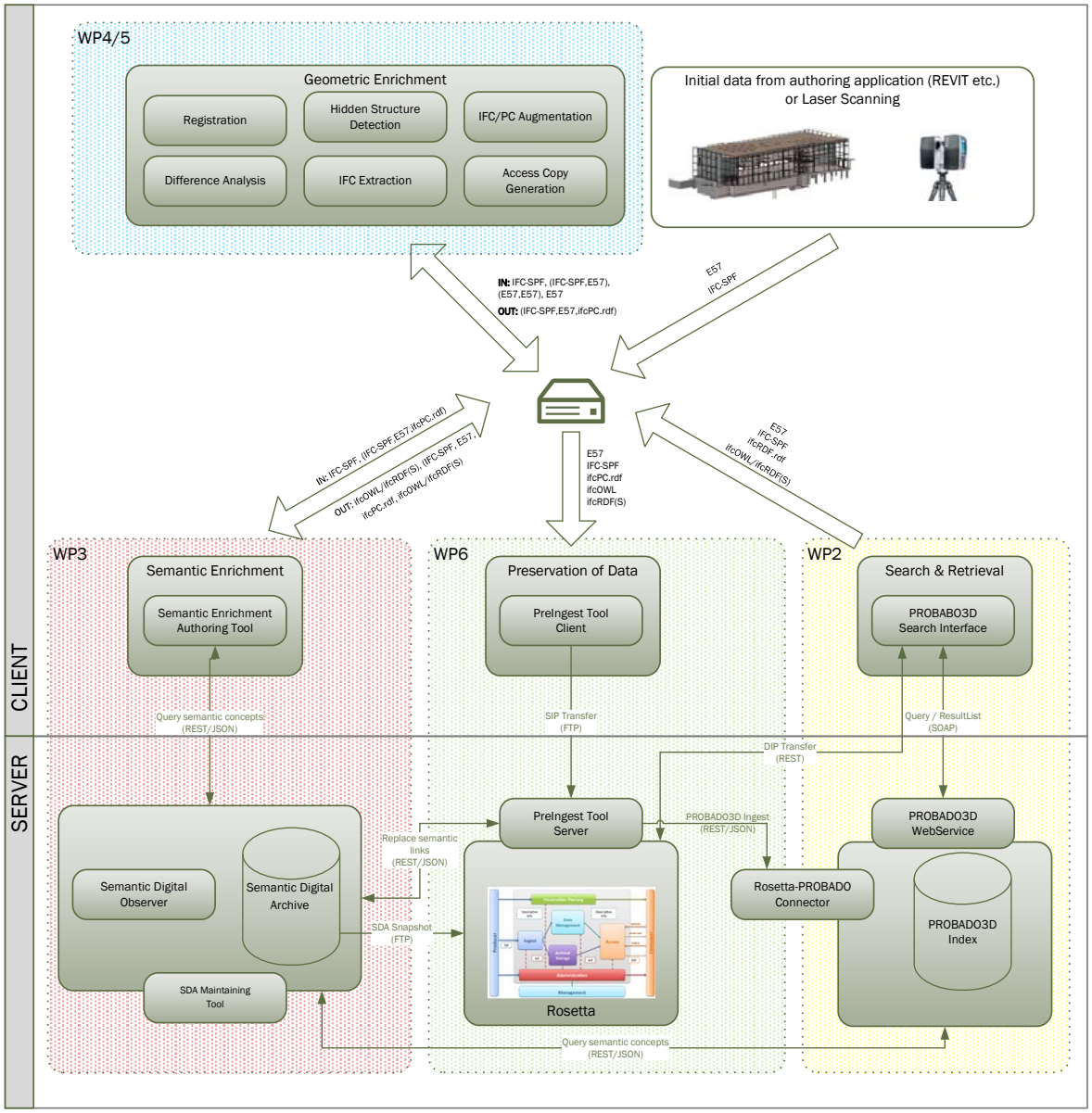

Figure 1: Systems overview of DURAARK.

2 Use-cases revisited

Long-term archival of data is a multifaceted task. In D2.2.1, the three basic layers of a digital preservation were mentioned: the bit preservation, the logical preservation and the semantic preservation layer. In maintaining the accessability and understandability of an object over time, all three layers have to be taken into consideration. D6.6.1 describes the state-of-the-art approaches currently in place for each of those layers.

Additionally to the above mentioned three basic layers of preservation it should be noted that preservation activities cannot be conducted without knowledge of the context in which they were created and the use or re-use they are intended for. Pennock developed the idea of digital curation as a life-cycle approach to both management and preservation of digital information in 2007, defining digital curation as "the active management and appraisal of digital information over its entire life cycle" [20]. Pennock carries on stating that this continuous task can only be fulfilled by maintaining active communication with the stakeholders of the data and by documenting their activities and needs.

The DCC (Digital Curation Centre) Curation Lifecycle Model[15] describes the lifecycle of an object from its conceptualization to its continuous use, disposal or re-use and transformation which leads to a new object. The model is a generic and high-level one, which is intended to be used with further reference models, frameworks or domain-specific tools and standards to take more granular approaches.

Regarding the role of stakeholders in the Curation Lifecycle Model, the identified DURAARK stakeholders described in D2.2.1 can be mapped to the three long-term archiving actors *producer*, *archive (OAIS)* and *consumer* in the following way:

Producer:

- Architects and Engineers
- Construction Companies
- Researchers and Lawyers
- Building Owners and Real Estate Managers
- Cultural Heritage Institutions
- Knowledge Base Providers

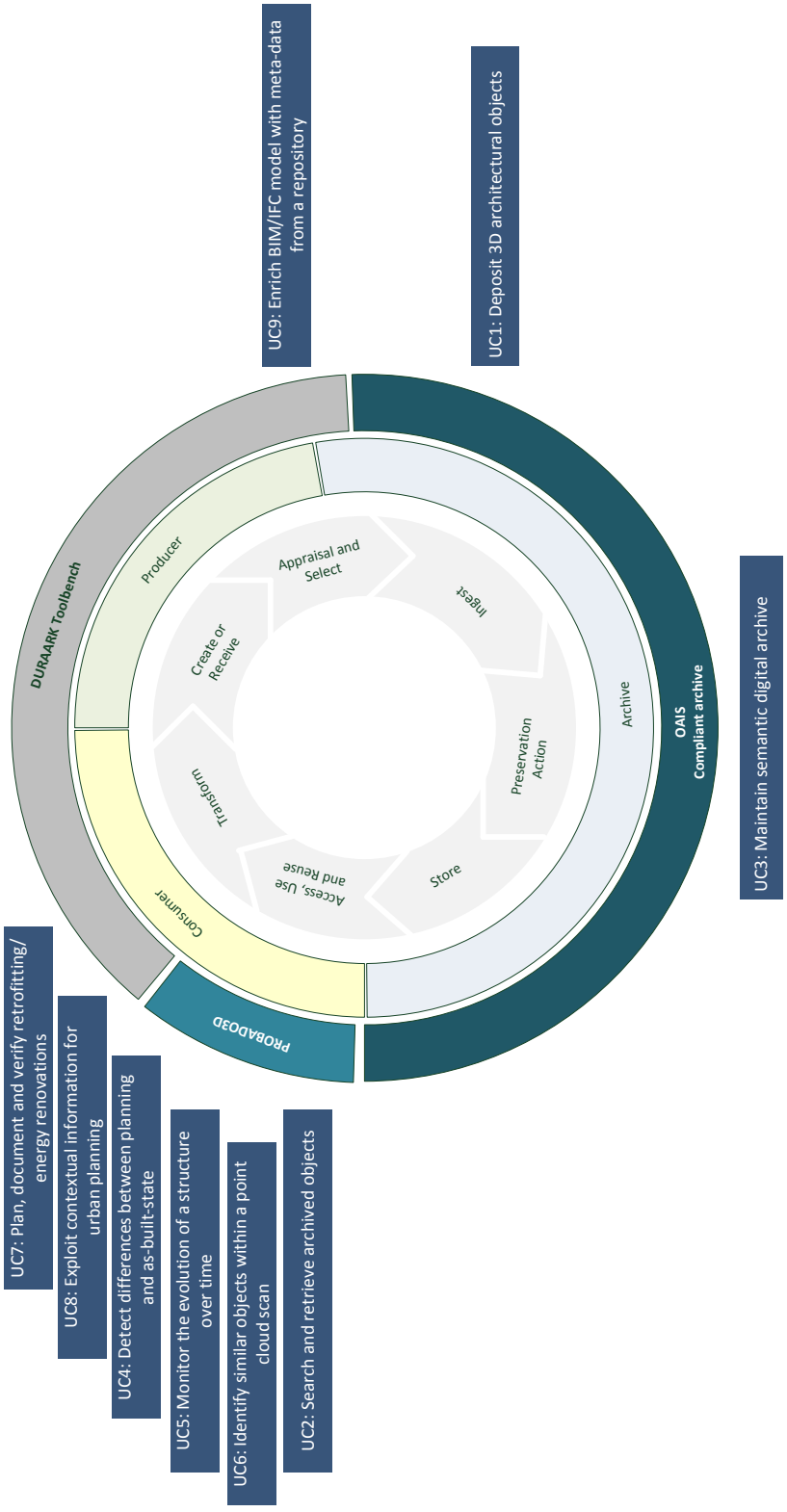

Figure 2: Preservation - Curation Lifecycle Model.

Archive (OAIS):

- Building Owners and Real Estate Managers
- Public Administrations / Public Planning / Policy Makers
- Cultural Heritage Institutions
- Knowledge Base Providers

Consumer:

- Architects and Engineers
- Construction Companies
- Researcher and Lawyers
- Building Owners and Real Estate Managers
- Public Administrations / Public Planning / Policy Makers
- Cultural Heritage Institutions

Figure 2 shows how those roles are involved in steps of the lifecycle model. Furthermore, the basis of the DCC lifecycle has been used to describe a domain specific view in the DURAARK context - positioning the relevant systems and the use-cases described in D2.2.1 along the lifecycle stages.

The approach of DURAARK is not to develop one monolithic system, but rather a set of tools, which can be assembled in various ways in order to fulfill the needs and requirements of the various use-cases. It enables also the use of 3rd party tools where necessary. As an example the used components for realizing the "UC4: Detect differences between planning and as-built state" are shown in figure 3.

The other curation use-cases (UC4-UC9) can be expressed in a similar way. A more generalised alignment between the components and the meta use-cases for geometric and semantic enrichment are described in section 4.

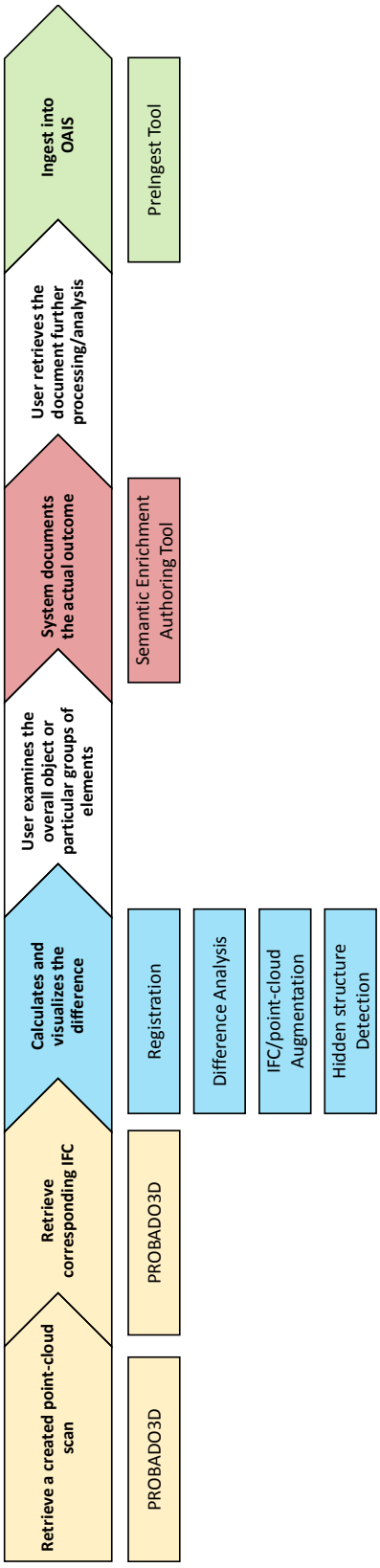

Figure 3: The first row shows the steps of UC4: "Detect differences between planning and as-built state". The boxes below show the DURAARK components which address the corresponding workflow step. The coloring of the components corresponds to the WP categorization of Figure 1.

3 Decisions, constraints, and justifications

This section lists the decisions that have been made regarding architectural approaches and the constraints which result from these decisions. These will serve as guidelines for defining architecturally significant parts of the system.

3.1 Quality and organization of measured data

The methods and tools developed in the course of the DURAARK project target usage scenarios from the Architecture, Engineering, Construction and Facility Management (AEC/FM) domain where a certain degree of precision and quality of computational results is mandatory. This imposes certain requirements on the input data and therefore the assumption is made that the measured point-cloud data is of a rather high quality (i.e. scans from high-quality laser scanning devices instead of consumer hardware). In general, each scan has to be (almost) spherical (that is, rays are cast from a point into all directions) and points have to be organized into rows and columns. Note that this is not a hard assumption as professional laser scanning devices capture the data organized in a fixed-size 2D-array.

Furthermore, in order to yield a satisfactory construction of a BIM model from point-cloud data, all relevant parts of the building have actually be "seen" by the scanner. Where small gaps in the data and scan shadows are a natural part of 3D scanning, good results cannot be guaranteed if large parts of the building are missing in the data. The same applies for parts of buildings that are not scanned at all.

For the format of point-cloud data, the E57 standard[1] has been selected. The main reasons for choosing this file format are that it is vendor-neutral, versatile and well-documented, as well as the availability of an open-source library (libE57[2]) for parsing it. The format can not only store the point data but also the scanner locations as well as images taken in the course of the scanning process. Scans from multiple locations may be contained in a single file.

The availability of the sensor locations can be a valuable hint for later processing and analysis of the point-cloud data. The images taken from the scanner's location are important for the computer vision methods that will be employed in WP5.

3.2 Quality and organization of BIM data

Information about buildings is captured in a wide variety of media that need to be preserved over long periods of time. These include text documents, images and technical drawings. Since the advent of Computer Aided Design (CAD) technologies crucial information is also captured increasingly in three dimensional representations of current (as-built) or intended (as-planned) states of a building. Where traditional 3D CAD models are limited to merely capture geometry, Building Information Models (BIM) describe buildings as an Object-Oriented (OO) assembly of components that also carry additional information such as the intended meaning ('load-bearing wall'), properties ('made of reinforced concrete') relations ('connected to floor') and behaviour ('buckles under load'). Each vendor of software tools implementing such Building Models employs an own, proprietary file format that is heavily depended on internal design decisions, software features etc.. The building industry however largely depends on collaboration. This also requires the interoperability of data among different trades and in heterogeneous software environments.

In the context of the DURAARK project, the Industry Foundation Classes (IFC) format has been chosen as the main carrier capturing Building Information Modeling data for the Long Term Digital Preservation. The format has become the industry standard in the building industry for exchange of building and construction data. As it is a neutral and open specification it is not controlled by a single vendor or group of vendors. The DURAARK consortium has good relations to buildingSMART group [1] (formerly the International Alliance for Interoperability, IAI). This organization governs the IFC data model and is continuously developing the format in order to facilitate interoperability in the Architecture, Engineering and Construction (AEC) industry. The IFC model specification is open and freely available. It is registered by ISO and is an official International Standard ISO 16739:2013[2]

It is assumed that the data captured in a Building Information Model is delivered with information that complies to the IFC model specification[3] at least on a geometrical level . In particular, files should be encoded as ISO 10303 part 21 files (referred to as SPF-STEP Physical File Format), which is also used in other engineering domains and their digital

[1] http://www.buildingsmart.org/bim
[2] http://www.buildingsmart.org/openbim
[3] http://buildingsmart-tech.org/specifications/ifc-overview/

preservation systems[25]. Since the initial formal requirements of IFC models are very lax (e.g. about 80% percent of all schema-level attributes are optional) additional constraints will have to be fulfilled to qualify an IFC identified for archival. Such constraints are expressed in so-called Model View Definitions (MVD). An MVD for the minimal requirements on IFC files is being defined in the context of the DURAARK project. This MVD referred to as "IFC/A" also imposes constraints on the meta-data extracted from the IFC model such as authorships, software versions involved in the production of the model, measures and units and other information that will be exposed for indexing and searching in the AIPs. Flawed, corrupted or damaged files are not part of data that is considered suitable for ingest to a long term archive.

Together with the CAD software market and the general technological progress in the construction industry the IFC model is constantly evolving. Hence the projected interoperability between software packages using the same IFC file is not always given. These incompatibility problems cannot be part of the considerations of the DURAARK project. It is assumed, that the current model revision ("IFC 2x3") will continue to be the predominant format for another few years before gradually being replaced by instance models adhering to the next release ("IFC 4"). However, since all model schema releases up to now have been created in backward compatible ways, future migrations for archival purposes are expected to be achievable with reasonable efforts.

More in-depth discussion on the use of the IFC data model and the SPF serialization formats can be found in the related reports D3.3.1, D6.6.1 and D7.1.1.

3.3 Use of Semantic Web and Linked Data standards

The extensive IFC model (see section 3.2) allows the modelling of semantically rich Building Information Model. However, even the currently over 700 class definitions, along with their few thousand attributes and properties are not enough to capture all information aspects in a uniform, interoperable way. Examples are local building regulations, domain-specific technical specifications of individual building products by vendors or organizational standards for data administration. To this end extension mechanisms are needed that allow the flexible implementation of these information requirements into data structures without creating new interoperability issues by e.g. requiring hardwired modifications of software tools. In the context of the DURAARK project such customized

data model extensions tailored to the individual needs of building projects must be stored along in the archive to allow comprehensive preservation.

Past and ongoing activities to harness Semantic Web technologies play an important role in the building and construction industry, including the architectural domain. In the past, a number of research efforts have aimed at providing manually curated, structured vocabularies of the various building-related engineering domains. Among them are the EU-projects eConstruct[27], IntelliGrid[13] and SWOP[9], as well as other national and international initiatives such as FUNSIEC[17]. The buildingSMART data dictionary (bsDD) has the ambition to be a central vocabulary repository that allows the parallel and integrated storage of different vocabularies such as the various classification systems (OMNICLASS Masterformat[4], UNICLASS [11], or SfB(-NL)[5]) which are widely adopted in the respective countries to structure building data. The bsDD also serves as the central repository to store meta-model extensions of IFCs - referred to as PSets - which are not part of the core model schema but are recognized as typical properties of common building component. A number of commercial domain-specific building product catalogs and conceptual structures have been established that are captured in proprietary data structures that are not yet exposed as Open Data, yet have gained the status of de facto industry standards. These include the international ETIM[6] classification along with its commercial implementation in the 2BA platform[7] for the description of electronic equipment in buildings, the Dutch Bouwconnect[8] platform, the German Heinze [9] product database and the CROW library for infrastructural objects[10]. Such structured vocabularies are often tightly integrated and oriented at local building regulation requirements and best practices and are often underlying structures for ordering higher-level datasets such as standardized text for tendering documents (German StLB[11], Dutch STABU[12], Finnish Haahtela[13] etc.). Even though their use and application in the context of the Semantic Web and Linked Open Data has been suggested repeatedly over a long period[5], the up-

[4]http://www.csinet.org/Home-Page-Category/Formats/MasterFormat.aspx
[5]http://nl-sfb.bk.tudelft.nl/
[6]http://e5.working.etim-international.com/
[7]http://www.2ba.nl/
[8]http://www.bouwconnect.nl/
[9]http://www.heinze.de/
[10]http://www.gww-ob.nl/
[11]http://www.stlb-bau-online.de/
[12]http://www.stabu.org/
[13]https://www.haahtela.fi/en/

take of harmonized structures is still in its infancy although internationally anticipated by large end-user communities.

The recent evolution and wide-spread adoption indicates that Linked Data is currently establishing itself as the de-facto standard for data integration on a web scale. Although certainly in harmony with Semantic Web research, work on Linked Data has focused on applying lighter technologies, e.g., RDF(S) and SPARQL, on devising simple, yet extensible and integrated vocabularies, and on establishing the fundamental means for exposing and interlinking data. Effectively, these technologies have simplified the integration of heterogeneous data sources to a certain extent, providing common languages for data representation and querying, as well as by borrowing Web standards for uniquely and globally identifying entities and transporting data. These various aspects make such technologies particularly appealing for capturing, sharing and integrating architectural knowledge within DURAARK, which is likely to be highly heterogeneous, provided by diverse actors, and yet needs to be effectively integrated on a global basis. Therefore, DURAARK meta-data will exploit Linked Data techniques and datasets to ensure a high level of interoperability and availability of generated data and meta-data. In particular, DURAARK will deploy existing state-of-the-art Linked Data storage and processing environments, which will be adopted and expanded to the needs of the architectural domain.

3.4 Digital preservation system: Rosetta

Within the project, the DURAARK system is to be used in conjunction with an existing OAIS compliant digital preservation system. As it is not within the scope of the DURAARK project to develop such a system, the chosen target system should be robust, extendable and well accepted within the preservation community. The system chosen for this task is Rosetta by Ex Libris[14]. The system has been included as an OAIS choice in other FP7 projects, such as PrestoPrime [15] and SCAPE [16]. Rosetta has been in productive use at TIB (LUH) since January 2012. TIB uses the system for their own data and furthermore hosts the consortially operated system for the two other German national subject libraries - the German National Library of Medicine (ZB MED) and the German

[14]http://www.exlibrisgroup.com/category/RosettaOverview
[15]`http://www.prestoprime.org/project/index.en.html`
[16]`http://www.scape-project.eu/`

National Library of Economics (ZBW). Together, the three libraries form the Goportis consortia, which conducted a digital preservation pilot study from 2009 through 2011. Within the pilot study several digital preservation systems were analyzed and a pilot implementation of Rosetta was tested against pre-defined criteria in a cooperatively operated digital preservation system. The Goportis consortium placed a high focus on the openness and the extendability of the system. Rosetta is a format agnostic system and is easily extendable for new workflows and materials through available APIs as well as its innate rule-based workflow engine and ability to support plugins. As a trustworthy OAIS compliant digital preservation system, it does, however, have a number of fundamental principles which hold true for all workflows and cannot be changed. As such, Rosetta implements the PREMIS data model[17] to capture the information flow as well as information about intellectual entities, rights, agents, objects and events. Within this context, an object is the smallest discrete unit of information. One or more objects make up an intellectual entity (IE), which is an intellectual unit for management and description of data in the digital preservation system. Furthermore, within the IE one or more objects may be grouped together as representations and objects may be further differentiated as files or bitstreams. The DURAARK system architecture needs to treat objects inline with the PREMIS data structure. The components should support output which can be mapped to the PREMIS entities - especially agents and events.

Rosetta will act as a "light archive", meaning the data is available to users in addition to being part of the content repository. The other option - a "dark archive" where access to the data is highly or completely restricted - has been dismissed, because of the need of an additional repository layer.

In the digital preservation workflow, the DURAARK system will function as a pre-ingest workbench which will prepare SIPs to be delivered to Rosetta (see figure 4). Pre-ingest functions are currently not covered within the OAIS, however, numerous sources like Beedham et al. [4], Kärberg [16], the UK Data Archive [28], Ruusalepp and Dobreva [23] have described the need to include pre-ingest processes in a preservation workflow. In an analysis of current research in digital preservation Strodl et. al have identified a "slow shift from addressing questions that help to fix problems in maintaining digital information over time to ensuring that the problem will not appear in its full complexity in the first place, reducing the need for specific ex-post fixing." [26]

[17] http://www.loc.gov/standards/premis/

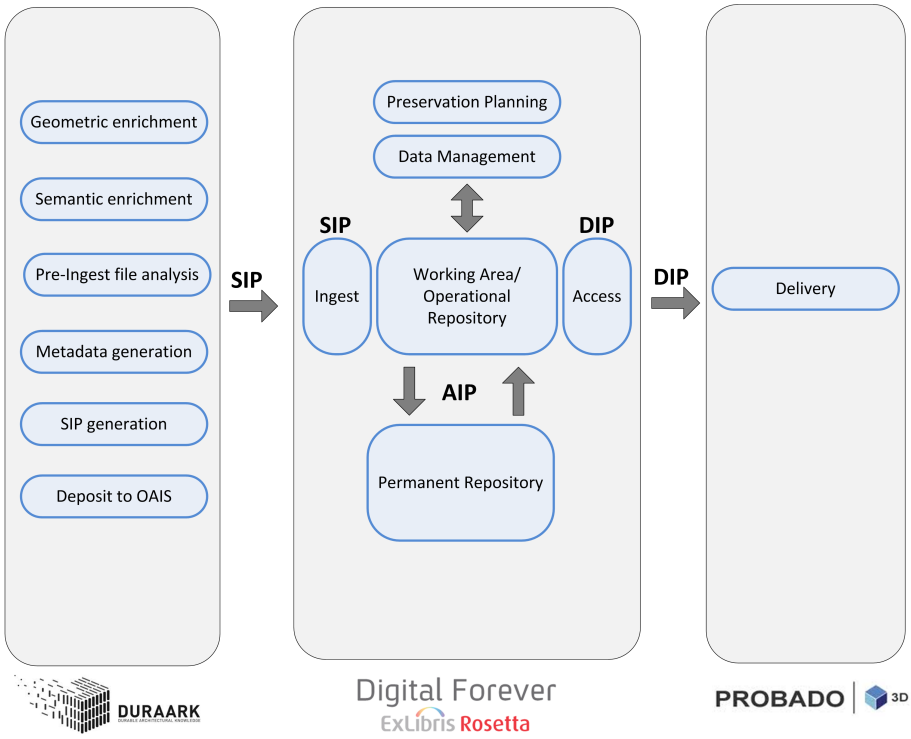

Figure 4: DURAARK, Rosetta and PROBADO in an OAIS compliant preservation workflow

This move from fixing problems to preventing them must involve the producer who holds contextual as well as technical domain knowledge. The various DURAARK system components are targeted towards different preservation issues of the object. The output aims to be a "preservation-ready SIP" approved by the producer to be included into an OAIS compliant system such as Rosetta.

3.5 Search and Retrieval with PROBADO3D

All use-cases belonging to the consumer side have the need for a search/browse facility in order to choose and retrieve the desired data. The requirements, how to formulate such a query, can differ depending on the use-case and stakeholder. Taken the use-cases UC7 "Plan, document and verify retrofitting/energy renovation" for example, the user - building owner/real estate manager, architect/engineer, construction company - may want to retrieve the archived model by simple specifying the address of the building. Another example is urban planning, where the user may want to perform the retrieval

based on a geospatial query, e.g. by selecting a rectangular area on a map. Also browsing the archived data by categories (e.g. using the Getty Art & Architecture Thesaurus[18]) and techniques like faceted search will improve the (re)use of archived data, e.g. when the user is looking for other buildings, which share a architectural concept.

Since Rosetta does neither provide sophisticated user interfaces nor any content-based retrieval techniques for architectural 3D data, it is intended to use PROBADO3D[7] as a basis for search & retrieval for the data preserved within Rosetta. PROBADO3D is one reference domain of the PROBADO project, which is a research effort to develop and operate advanced Digital Library support for non-textual documents[6]. PROBADO3D will be used as the interface for browsing and searching the archived data within Rosetta. This can either be done by using the PROBADO3D web pages or the various SOAP requests provided by the PROBADO3D service. PROBADO3D is especially tailored to the needs of the architectural domain and offers an established search & retrieval infrastructure (e.g. indexing, 3D PDF preview generation), which can be easily utilized for the various DURAARK workflows.

3.6 Strategic decision on file formats

3.6.1 Geometric enrichment

The geometric enrichment components developed in WP4/5 deal with the extraction of information from point-cloud and BIM data. The gained information may either complement the datasets (e.g. attributes attached to parts of the dataset) or interlink datasets (e.g. corresponding parts of a point-cloud and a BIM model). Therefore, file formats for storing point-clouds, BIM models, and any additional, derived information is needed.

The chosen file format for point-cloud data is E57 and the format for BIM models is IFC. The reasons for these decisions have been discussed in section 3.1 and 3.2

For linking datasets with additional information or with each other, a format based on the RDF model (Resource Description Framework[19]) will be developed. The RDF model basically specifies means to store relationships between entities in so-called *triples*.

[18] `http://www.getty.edu/research/tools/vocabularies/aat/`
[19] `http://www.w3.org/RDF/`

For example, when a set of points within a point-cloud is found to have corresponding entities within an IFC file of the same building using the geometric enrichment tools, this relationship may be stored in an additional RDF file which is subsequently stored in addition to the original datasets. This method of storing the derived information has several advantages:

- The original datasets are not modified. This is an important property with respect to long-term preservation because any modification of the original data may have unwanted implications.
- The standards used for storing the point-cloud and BIM datasets (E57, IFC) do not have to be extended or otherwise modified in order to store the additional information.
- If the additional information is missing or for any reason becomes invalid, the original datasets are not affected.
- Redundancy is avoided where possible. For instance, storing alignment information coming from the registration component only requires to store concise information about the transformation itself instead of storing a transformed copy of a dataset.
- The RDF standard offers an universal and extensible basis for storing different kinds of information. This allows future extensions where necessary or desired.

3.6.2 Semantic enrichment

As described in section 3.3, Building Information Models are often augmented with additional information not specified e.g. on the schema level of the IFC model. This process is referred to as 'semantic enrichment'. The increasing use of external, structured vocabularies in formats such as RDF published as Linked Data, also require novel preservation strategies for such distributed and networked data. To adress these needs, efforts in the DURAARK project are dedicated to specialized repositories that archive such external datasets in a Semantic Digital Archive (SDA). The main purpose of the SDA is the preservation of evolving external datasets to create a degree of self-containment and autonomy for the archival institutions using the DURAARK system.

Semantic enrichment developed and further explained in the deliverable of WP3 act on a number of different levels:

- pre-ingest semantic enrichment of IFC model instances using EXPRESS schema-conform referencing mechanisms
- pre-ingest semantic enrichment of RDF(S)/OWL representations of such model instances using RDF mechanisms
- semantic enrichment of building models from the outside by a curator

A more detailed description on the formats used for semantic enrichment is given in deliverable D3.3.1.

4 System architecture

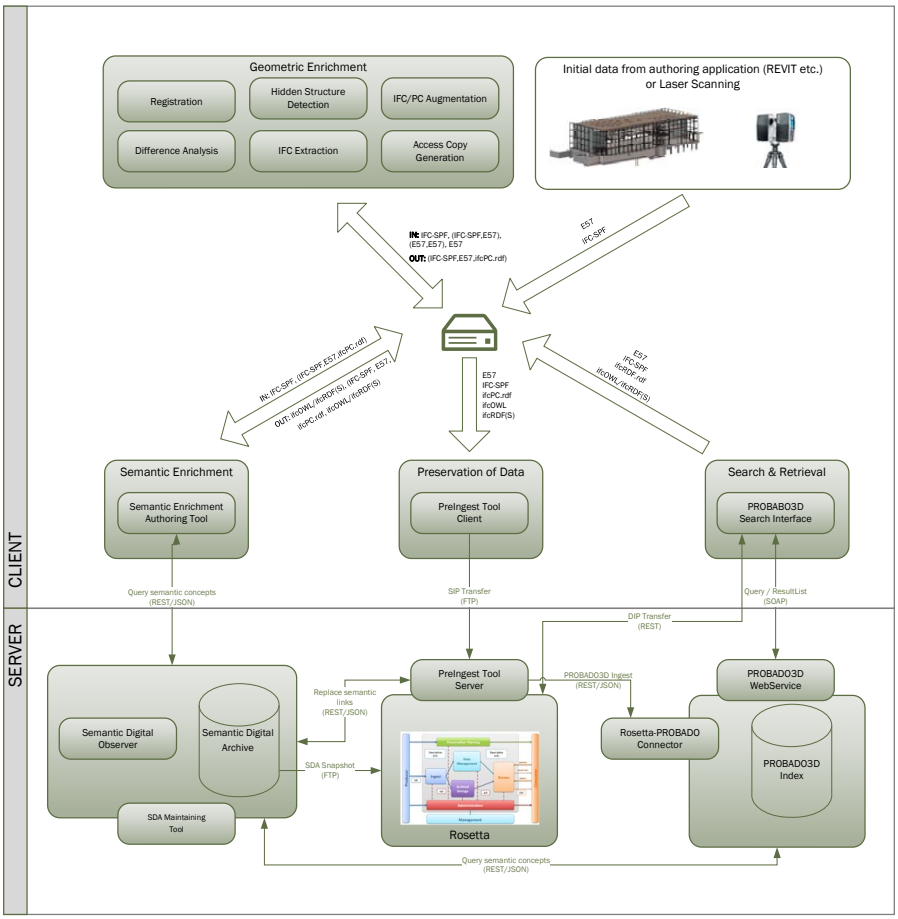

Figure 5: System Overview of the DURAARK system

Figure 5 gives an overview of the DURAARK components and the interfaces between them. One common aspect of the three main activities - geometric enrichment, semantic enrichment and preservation of data - is that they are data-centric from the client-side perspective, e.g. that they process the input file(s) and either modify or attach additional information to them. Each of these activities follow the "input, processing, output" pattern accepting a defined set of input and output files. Using this file-based approach for exchanging the data between the geometric enrichment, semantic enrichment, and preservation of data is the key property to enable an easy coupling of these components with other 3rd party tools in order to address different workflows.

Figure 6 shows the involved file formats within the DURAARK system. A detailed description can be found in sections 4.1, 4.2, 4.3, and 4.4.

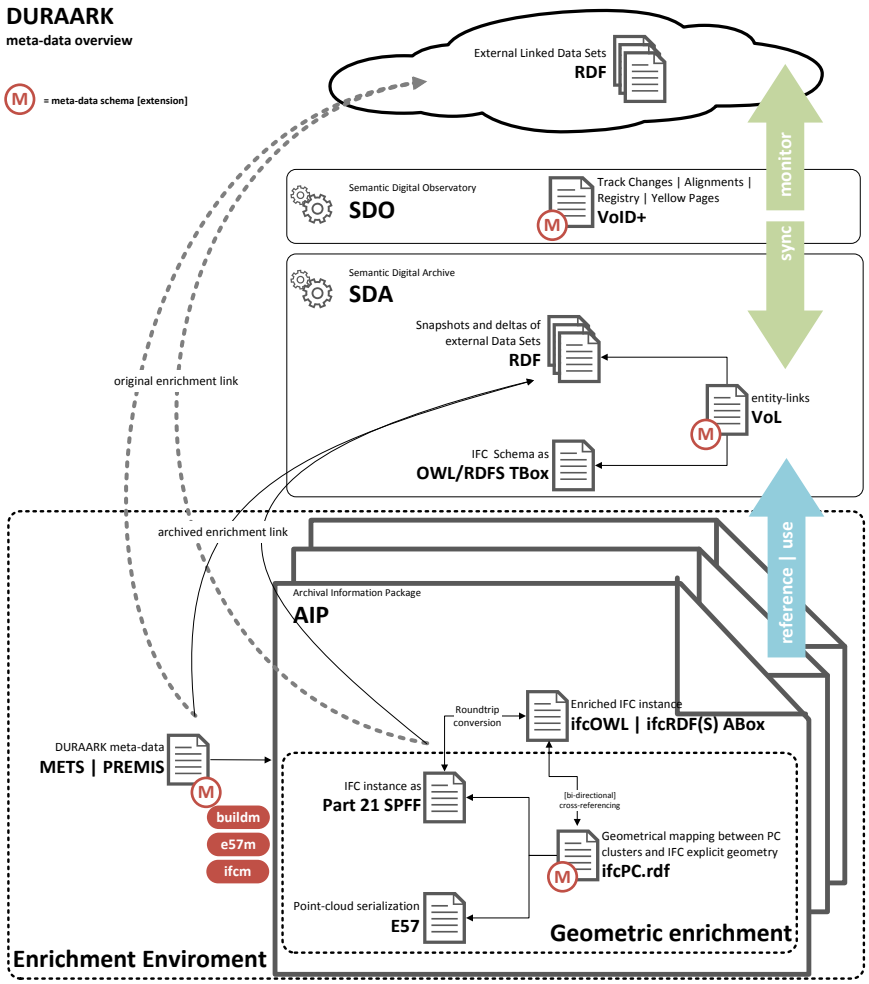

Figure 6: Overview of the involved data formats within the DURAARK system.

While the geometric enrichment client-components do their processing completely on the client, components for semantic enrichment, preservation of data, and search & retrieval require data exchange with their corresponding server-components. The communication within and between these subsystems follows the service-oriented architecture design paradigm (SOA). SOA is the aggregation of components that satisfy a business need and communicate between each other by exchanging structured messages following standardized APIs. It comprises components, services, and processes. Components are

binaries that have a designed interface (usually only one), and a service is a grouping of components (executable programs) to get the job done[3].

The semantic enrichment (SDA/SDO) and the preservation of data (PreIngest Tool Server/Rosetta) subsystem will be based on REST-based interfaces - where feasible - exposing either JSON or XML or both as message formats. Components involved in the exchange of large quantities of data (e.g. ingest of a point-cloud data) require more low-level and closer integration (e.g. using FTP). Communication between the PROBADO3D search client and the PROBADO3D service will be based on SOAP[8].

In the following sections the various subsystems (geometric enrichment, semantic enrichment, preservation of data, and search & retrieval) will be described in more detail.

4.1 Geometric Enrichment

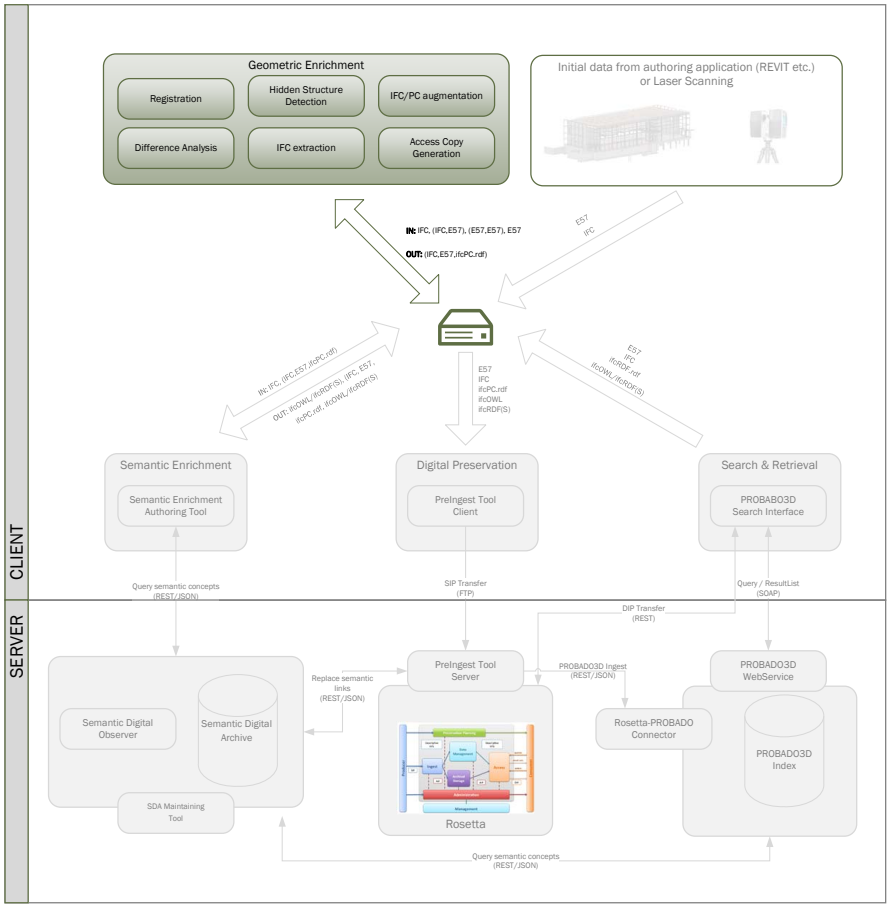

Figure 7: Components and interfaces related to the geometric enrichment.

This section will give a more detailed description of the geometric enrichment aspects of the DURAARK system along with a refined component list. Figure 7 highlights the components and interfaces related to geometric enrichment.

The algorithms developed within the geometric enrichment deal with the processing and analysis of point-cloud data as well as BIM models. They include algorithms for the registration (alignment) of two representations of the same building, the detection and extraction of differences between them as well as methods for access copy generation. Also, algorithms for the generation of IFC data from given point-clouds and the transfer of semantics from a given IFC model to a point-cloud will be included. These components (see section 4.1.3) will be compiled as libraries which may then be used in interactive or

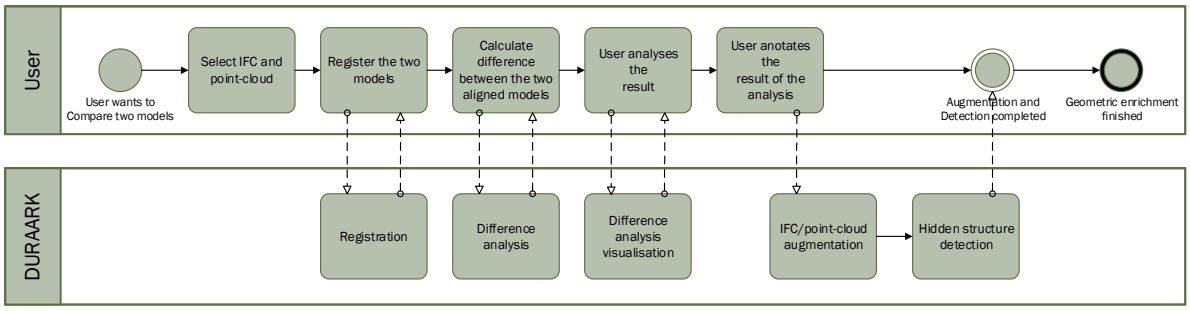

Figure 8: Workflow for geometric enrichment of a IFC model with the corresponding point-cloud scan.

automated tools. The advantage of this model is that the functions implemented in the libraries are reusable and not necessarily tied to a specific program.

4.1.1 Workflow geometric enrichment

A typical workflow for geometric enrichment is shown in Figure 8. The user selects the IFC representation and a corresponding point-cloud scan of an architectural model. The IFC model and the point-cloud scan can be either newly created data (e.g. by an authoring application like Autodesk Revit or a laser scanning) or already preserved data, which resides in the DPS. How IFC models and corresponding point-cloud scans are retrieved from the DPS is described in section 4.4. After the data has been selected, the IFC and the point-cloud are registered (aligned) to each other. The registered versions are now processed by the difference analysis. Using the IFC/point-cloud augmentation links regions within the point-clouds to their counter-parts within the IFC model. The format for this geometric enrichment is described in detail in section 4.1.2.

4.1.2 Data Formats

In the course of WP4 and WP5 algorithms are developed that geometrically enrich point-cloud data. This includes the semantic association of point-cloud subsets with BIM elements described in the corresponding IFC files. For example a set of points could be linked to a door element described as an IFC entity. In order to store this semantic

association and make it accessible, a well-defined and standardized representation format must be defined in the course of the DURAARK project.

This section proposes a description format based on the RDF specifications in order to externally link point-cloud data with IFC entities. A specific association defined in such a way consists of

- a reference to the point-cloud linked,
- a reference to the IFC file linked,
- specifications of subsets in the aforementioned point-cloud, and
- link definitions between defined point-cloud subsets and entities in the referenced IFC file.

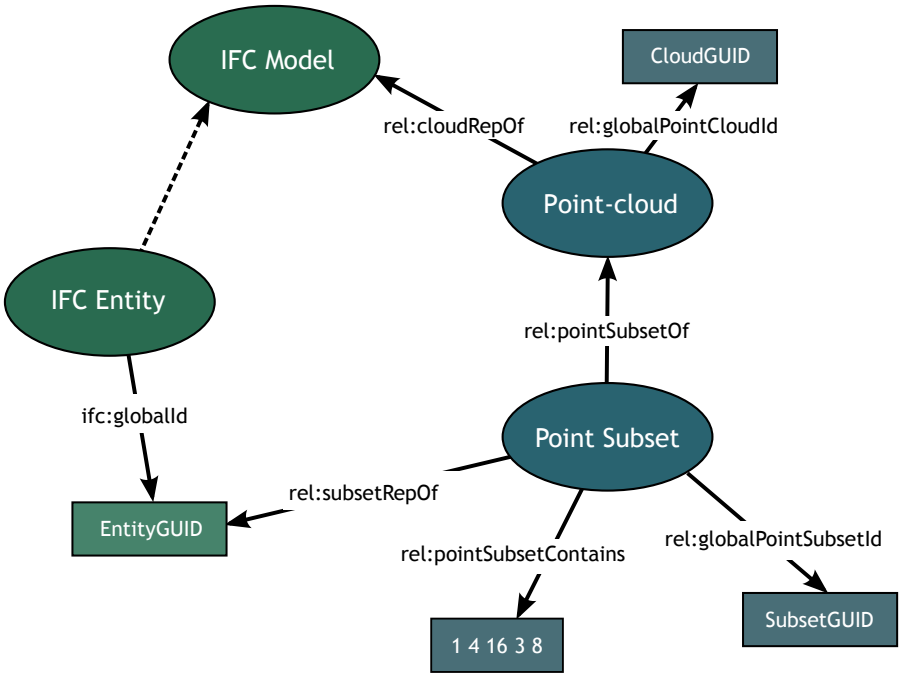

Figure 9: IFC/Point-cloud linking.

Figure 9 illustrates the proposed structure of the RDF-based entity linking.

One of the benefits of adhering to the RDF standard is that all specified references address entities in a globally unique way (i.e., despite being an association defined neither inside the IFC file nor inside the point-cloud file, it is possible to define a *specific* point-cloud subset and link it to a *specific* IFC entity across file or model boundaries).

Additionally, using the RDF provides a standardized and well defined way to represent this geometric enrichment in a self-documenting way and make it globally accessible through well defined interfaces.

Example file representations in the *RDF/XML* as well as *Turtle* format are shown in Listing 1 and Listing 2 and implement an association between an IFC file and a point-cloud where two non-disjoint subsets are linked to two distinct IFC entities.

Listing 1: IFC/pointcloud linking example in RDF/XML format

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rel="http://duraark.eu/rdf/relations#"
  xmlns:obj="http://duraark.eu/rdf/objects#"
  xmlns:types="http://duraark.eu/rdf/types#"
  xmlns:model="http://dsg.cs.hut.fi/model#"
  xmlns:ifc="http://dsg.cs.hut.fi/schema/IFC2X3#">

  <rdf:Description rdf:about="obj:subset1">
    <rel:globalPointSubsetId rdf:datatype="xsd:string">
      "236236-23623623-36236236-236236"
    </rel:globalPointSubsetId>
    <rel:pointSubsetOf rdf:resource="obj:cloud1"/>
    <rel:pointSubsetContains rdf:datatype="types:indexSet">
      "1 2 15 4 7 3"
    </rel:pointSubsetContains>

    <rel:subsetRepOf>
      <rdf:Description rdf:nodeID="entity1">
        <ifc:globalId rdf:datatype="xsd:string">
          "29afdfa5-45e1-44ec-bff7-851b5165e8ef"
        </ifc:globalId>
      </rdf:Description>
    </rel:subsetRepOf>
  </rdf:Description>

  <rdf:Description rdf:about="obj:subset2">
    <rel:globalPointSubsetId rdf:datatype="xsd:string">
      "683957-32850622-24857061-656779"
    </rel:globalPointSubsetId>
    <rel:pointSubsetOf rdf:resource="obj:cloud1"/>
    <rel:pointSubsetContains rdf:datatype="types:indexSet">
      "15 4 25 22"
    </rel:pointSubsetContains>
    <rel:subsetRepOf>
      <rdf:Description rdf:nodeID="entity2">
        <ifc:globalId rdf:resource="xsd:string">
          "f55eaf97-145e-4431-b2f3-69f9634f244b"
```

```
        </ifc:globalId>
      </rdf:Description>
    </rel:subsetRepOf>
  </rdf:Description>

  <rdf:Description rdf:about="obj:cloud1">
    <rel:globalPointSubsetId rdf:datatype="xsd:string">
      "224352-2303900-2249-99786"
    </rel:globalPointSubsetId>
    <rel:cloudRepOf rdf:resource="model:GUID_Ka_fpUXhROy_94UbUWXo7w"/>
  </rdf:Description>
</rdf:RDF>
```

Listing 2: IFC/pointcloud linking example in Turtle format

```
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#>    .
@prefix rel:   <http://duraark.eu/rdf/relations#>     .
@prefix obj:   <http://duraark.eu/rdf/objects#>       .
@prefix types: <http://duraark.eu/rdf/types#>         .
@prefix model: <http://dsg.cs.hut.fi/model#>          .
@prefix ifc:   <http://dsg.cs.hut.fi/schema/IFC2X3#> .

obj:subset1
  rel:globalPointSubsetId "236236-23623623-36236236-236236"^^xsd:string ;
  rel:pointSubsetOf obj:cloud1 ;
  rel:pointSubsetContains "1 2 15 4 7 3"^^types:indexSet ;
  rel:subsetRepOf [
    ifc:globalId "29afdfa5-45e1-44ec-bff7-851b5165e8ef"^^xsd:string
  ] .

obj:subset2
  rel:globalPointSubsetId "683957-32850622-24857061-656779"^^xsd:string ;
  rel:pointSubsetOf obj:cloud1 ;
  rel:pointSubsetContains "15 4 25 22"^^types:indexSet ;
  rel:subsetRepOf [
    ifc:globalId "f55eaf97-145e-4431-b2f3-69f9634f244b"^^xsd:string
  ] .

obj:cloud1
  rel:globalPointCloudId "224352-2303900-2249-99786"^^xsd:string ;
  rel::cloudRepOf model:GUID_Ka_fpUXhROy_94UbUWXo7w .
```

4.1.3 Components

This section describes the components dealing with geometric enrichment within the DURAARK system. All components for geometric enrichment are client-side components.

Registration component This component shall provide functions for registering (aligning) multiple point-cloud scans and IFC files to each other. It would be most meaningful to integrate this into an interactive GUI environment, offering the user an interactive preview and controls over the registration process.

- **INPUT** A pair of two (possibly roughly pre-aligned) representations of a building (point-cloud or IFC model).
- **OUTPUT** A transformation aligning both representations.

Difference analysis component This component shall provide functions for comparing two point-clouds or a point-cloud with a given IFC model or legacy CAD data or between two IFC models. This library could be used in an interactive GUI environment that allows the user to see differences between the models with appropriate highlighting of parts that differ in the available representations. It should also be possible to extract and store differences in an appropriate file format (for instance an enriched, annotated point-cloud storing the correspondences and differences and a report highlighting the minimal, maximal, and median deviations).

- **INPUT** Either two point-clouds or a point-cloud and an IFC model.
- **OUTPUT** Annotated point-clouds or a custom file format for storing differences (similar to textual "diff" tools).

Access Copy Creation This component shall provide functions for generating lightweight representations of point-clouds which are suitable for quick previewing and fast transfer of the datasets, e.g. over network channels. The obtained representations are not necessarily lossless but shall provide a good visual preview of the underlying data. This software component may be used in a "headless" executable which is suitable for batch-processing multiple input datasets, or it may be used as part of a graphical application which offers the user a preview of the generated representations.

- **INPUT** Point-clouds and – if available – a corresponding IFC model serving as additional information for compression.
- **OUTPUT** Either point-cloud data that has been decimated in a meaningful way or data in a custom compressed format.

IFC extraction component This component shall provide functions for creating IFC files from point-clouds in a (semi-)automatic manner. It may be used in an interactive GUI-environment, providing a multi-phase conversion workflow for creating an IFC model from given scans. The user should be able to intervene in each phase of the extraction in order to tweak or correct it where necessary while being supported by automatic algorithms where possible.

The scope of the extraction will be the coarse building structure consisting of floors, walls, doors and windows. This representation also enables to search for entities that can be described in a simple manner (for instance, to find all instances of a door of a given width and height).

- **INPUT** A registered point-cloud consisting of multiple scans (including the information about the scanner positions in the global coordinate system).
- **OUTPUT** An IFC model containing the extracted entities (floors, walls, doors, etc.).

IFC-based geometric point-cloud augmentation component The process of comparing a scan with an existing IFC opens up the possibility to connect semantically meaningful parts of the point-cloud with their corresponding IFC entities.

One possible approach might be: A decision which point belongs to which entity is made based on the distance between the point and a geometric representation of the entity. Semi-automatic feature detection will help identifying objects, e.g. walls that are positioned further off than a given threshold value. Some degree of manual intervention will be needed using a GUI.

- **INPUT** IFC model and matching point-clouds.
- **OUTPUT** Either an annotated point-cloud where each point is associated with

its IFC entity (if any) or an enriched IFC model where each entity is assigned its corresponding subset of points.

Hidden structure detection component This component will identify power sockets and outlets, as well as light switches and (if visible) distribution sockets and cavity sockets. The result will be a set of images with markups for the identified components (e.g. a light switch) and their probability. These markups are the input for shape grammars. The grammar represents the rules, according to which power and water lines are installed; the markups in the images are the terminal symbols. A 3D structure will be generated, which represents the known inputs (e.g. light switches, respectively terminal symbols) best. This component may be used in an interactive GUI-environment.

- **INPUT** Multiple separate point-clouds from different scanner positions (including the information about the scanner positions in the global coordinate system) and the images created during the scanning process.
- **OUTPUT** An IFC file containing the detected structures (power lines, etc.).

4.1.4 Technical aspects

The components for the geometric enrichment of datasets developed in WP4/5 will be implemented mainly using the C++11 programming language. The most important reasons for this decision are:

- C++ is a popular, general-purpose programming language which allows for high performance due to a high degree of control over the program.
- The relatively new C++11 standard offers powerful and modern extensions to the C++ language.
- The external software libraries that are used are written in C++ or C. Using the C++ language enables efficient integration of these libraries into the developed components.
- A software component written in C++ can also be integrated into programs which use the Microsoft .NET framework (for instance, using Managed C++) which will

be useful for connecting the developed components to other programs like Autodesk Revit using plugin interfaces.

External software libraries will be used for reading and writing the E57 and IFC formats as well as for working with point-cloud and mesh data internally. The following libraries have been chosen for these tasks:

- libE57[20] for input/output of point-cloud data,
- IfcOpenShell [21] for working with IFC files and conversion of the contained BIM models to mesh geometry,
- Point Cloud Library (PCL[22]) for working with point-cloud data once loaded into memory and performing operations on that data,
- OpenMesh [23] for working with mesh data internally.

In addition, the following libraries are used (most of them are dependencies of other libraries).

- Boost, a collection of utility libraries for C++,
- Eigen 3, a versatile linear algebra library,
- Flann, a library for approximate nearest neighbor searches,
- GLEW, an extension wrapper library for OpenGL,
- ICU, International Components for Unicode,
- Open CASCADE, used by IfcOpenShell to generate mesh geometry from IFC entities,
- OpenCL, a programming interface for parallel computations,
- OpenGL, a widely-used, cross-platform graphics library,
- pcshapes, a library for detecting primitive shapes in point-clouds (see [24]),
- Qt5, a cross-platform graphical user interface toolkit,

[20] http://www.libe57.org/
[21] http://ifcopenshell.org/
[22] http://pointclouds.org/
[23] http://www.openmesh.org/

- Xerces, a XML parser,
- Zlib, a compression library.

4.2 Semantic Enrichment

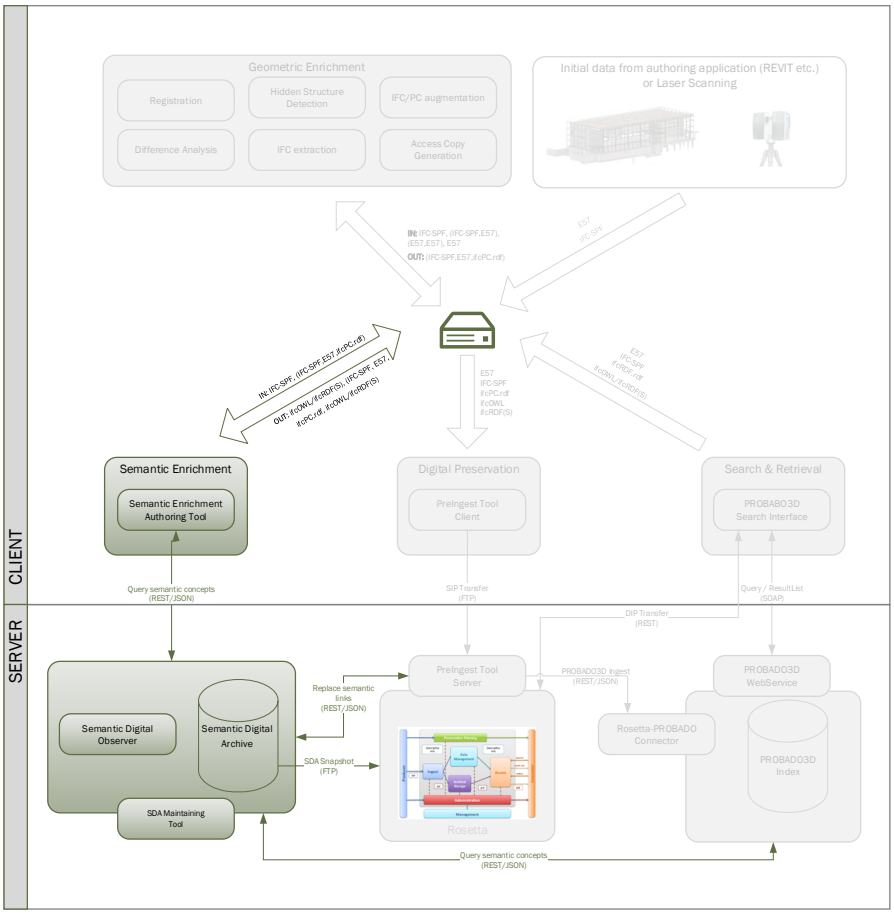

Figure 10: Components and interfaces related to the semantic enrichment.

This section will give a more detailed description of the semantic enrichment aspects of the DURAARK system along with a refined component list. Figure 10 highlights the components and interfaces related to semantic enrichment.

Within this subsystem a large repository and archive of vocabularies (Semantic Digital Archive, SDA) will be developed, which will complement BIM data and provides more expressive meta-data about built structures, their context, environment, usage and history. The Semantic Digital Observatory (SDO) is the component responsible for harvesting the Linked Open Data (LOD)[14] and organise updates to already existing datasets within the SDA.

Following the design decisions for exploiting Linked Data techniques and standards at the

meta-data level, the SDA will be based on graph-based RDF storage systems, providing SPARQL-based access for communication. While it is assumed that (a) large quantities of data have to be managed and (b) frequent queries need to be supported from other components, DURAARK will rely on established storage and SPARQL endpoint solutions, which will be selected according to the requirements elicited in early stages of the project. SPARQL endpoints are services that enable users to query RDF datasets via using the SPARQL query language. RDF triplestores, i.e. databases providing persistent storage and access to RDF graphs, usually provide SPARQL endpoints. All of the most widely used triplestores (e.g. Virtuoso, AllegroGraph, Joseki, Sesame/OpenRDF or Mulgara just to name a few), implement SPARQL endpoints (albeit somtimes with varying support for the features of the query language specification). In addition, while we anticipate large quantities of data to be processed over time, DURAARK will consider distributed RDF storage and processing solutions on the basis of Hadoop[24], exploiting the Hadoop cluster available at LUH (L3S).

4.2.1 Workflow

The SDA/SDO are involved in two different steps during the various workflows within DURAARK: data ingestion (link mirroring and curation) and delivery of DIP.

Data ingestion - link mirroring References to vocabularies from inside the IFC to external vocabularies are checked for their presence/up-to-dateness as mirrors in the SDA. If elements such as complete datasets or individual resources in a known dataset are missing from the current state of the SDA, the SDO is triggered to 'harvest' them from their original data sources interfaced with SPARQL endpoints. For every URI referencing external data inside an IFC instance file, a 'shadow'/'redirection' URI will be created pointing to the appropriate resource in the SDA. This is an automated process. Interaction takes place via web-services.

Data ingestion - curation the librarian/curator consults the SDA from his 'curation dashboard' for datasets and vocabularies for the semantic enrichment of the meta-data

[24]http://hadoop.apache.org/

(in the METS/PREMIS records) and 'instantiates' them: Sentiment towards a building, historic period, geo-references, architectural style. Interaction takes place via web services.

Figure 11 shows the interaction of the PreIngestTool with the SDA/SDO in order to archive and preserve external links within the IFC file. Figure 12 shows the typical curation workflow, where a term is looked up for existing concepts within the SDA, while Figure 13 shows the tasks necessary to enrich the IFC with data from the Linked Open Data cloud. Details on this can be found in deliverable D3.3.1.

Delivery of DIP During delivery of a DIP: original URIs from the IFC file are resolved via the mapping with the 'shadow' URIs (see ingest step 1). If needed, minimal 'shadow copies' of the vocabularies/RDF graphs are delivered as files alongside the DIP. Another possibility is a tool that manipulates URIs of the extracted copies of the IFC files from the DIP to point to the 'live' SDA. Original data stays untouched, a copy is being used.

4.2.2 Data Formats

In the context of the DURAARK project a number of different forms and notations of links between data items have to be considered. In general terms they can be categorized roughly in

internal links that are embedded into the main information carrier which stands in the focus of the preservation effort and point **to other internal or external resources**. In particular, references to external vocabularies that are used to semantically enrich the description of buildings beyond the IFC schema-level are considered here. A concrete example for such internal enrichment link is the reference to the specification of a concrete building product (the model number and configuration of a heating boiler). Thus, a self-sustaining archive of this building has to capture the IFC file itself as well as the snapshot of the external product data. Note though that such concrete building products will most likely be less frequent than other internal aspects.

external links that leave the main information items to be preserved unmodified but relate information resources **in external contexts**. The main examples for this

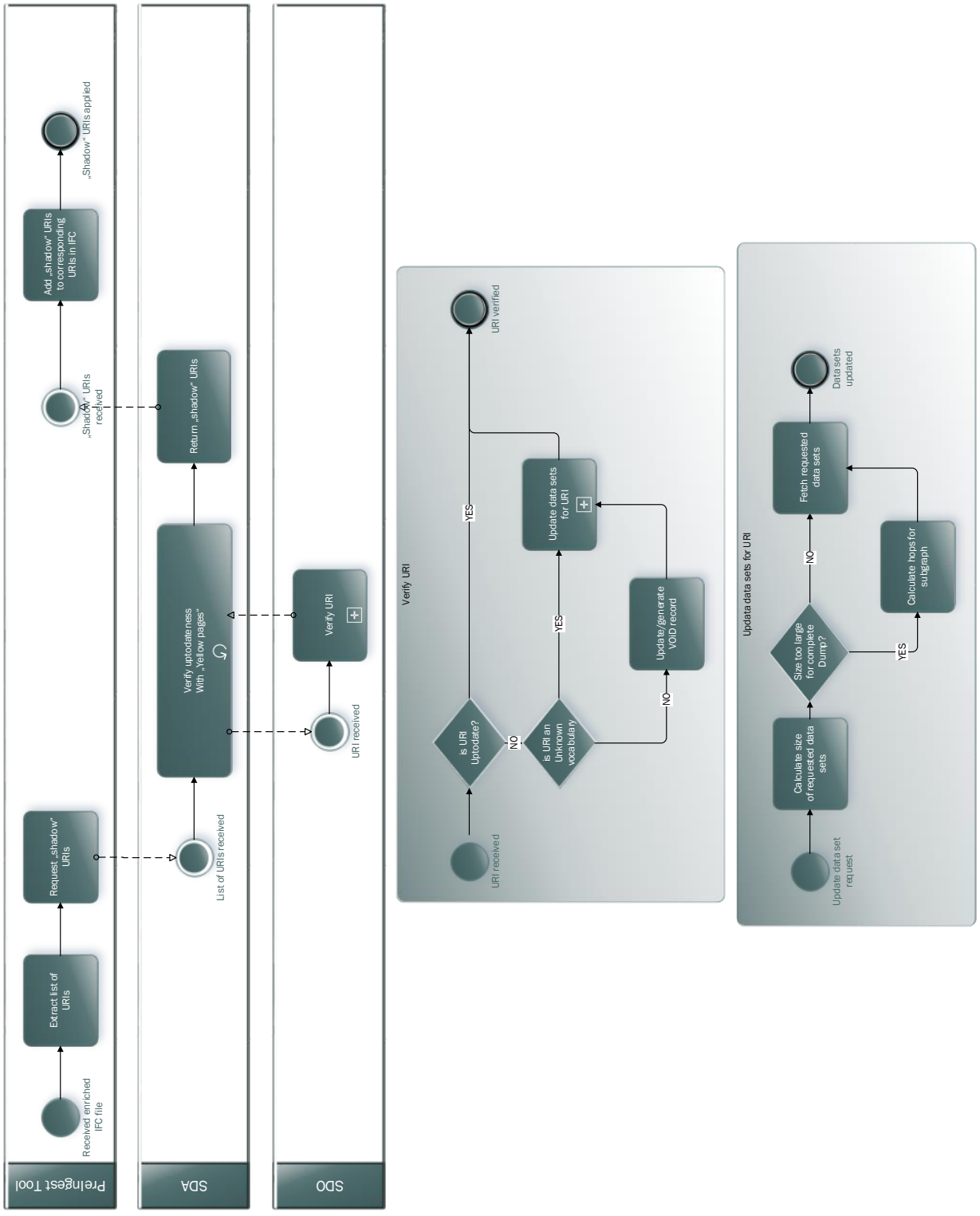

Figure 11: Interaction between the PreIngestTool and the SDA/SDO during ingestion.

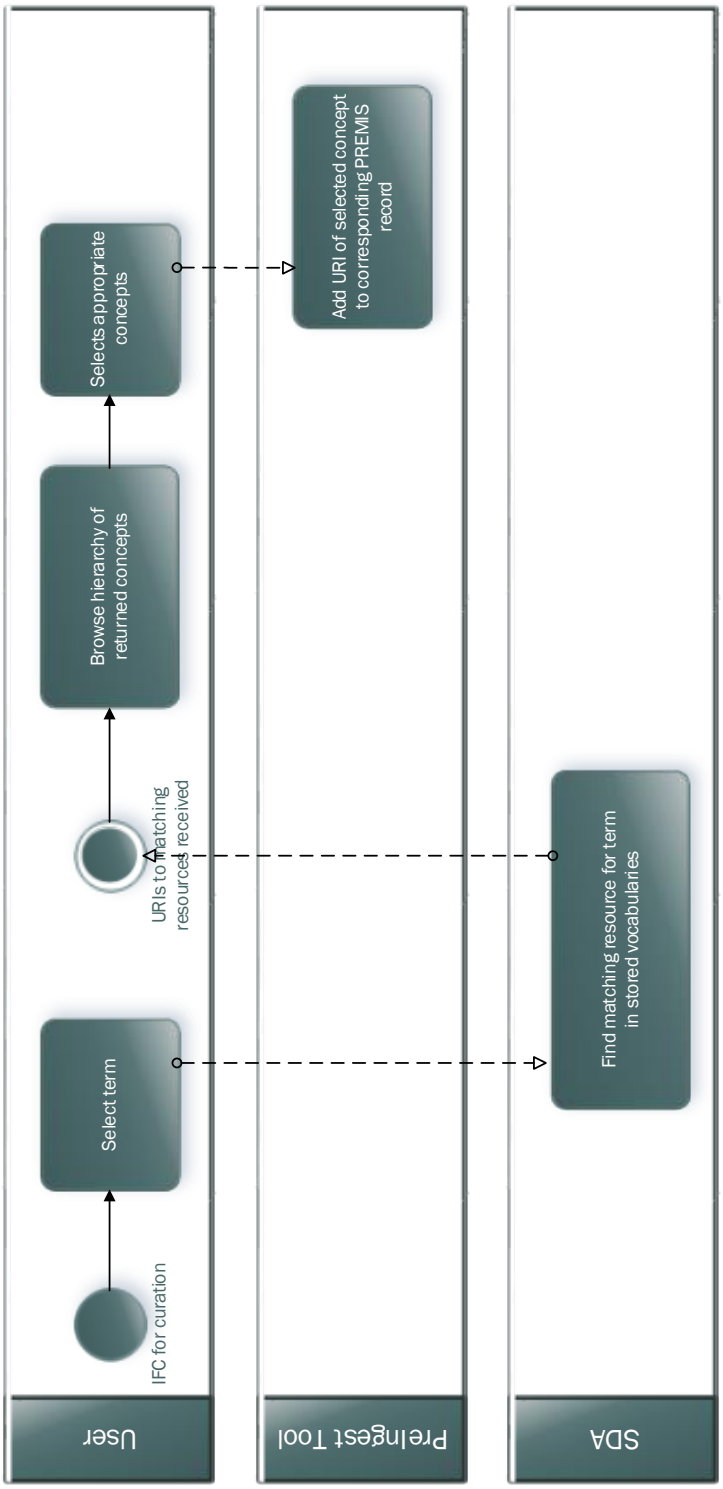

Figure 12: Typical curation workflow for looking up existing concepts.

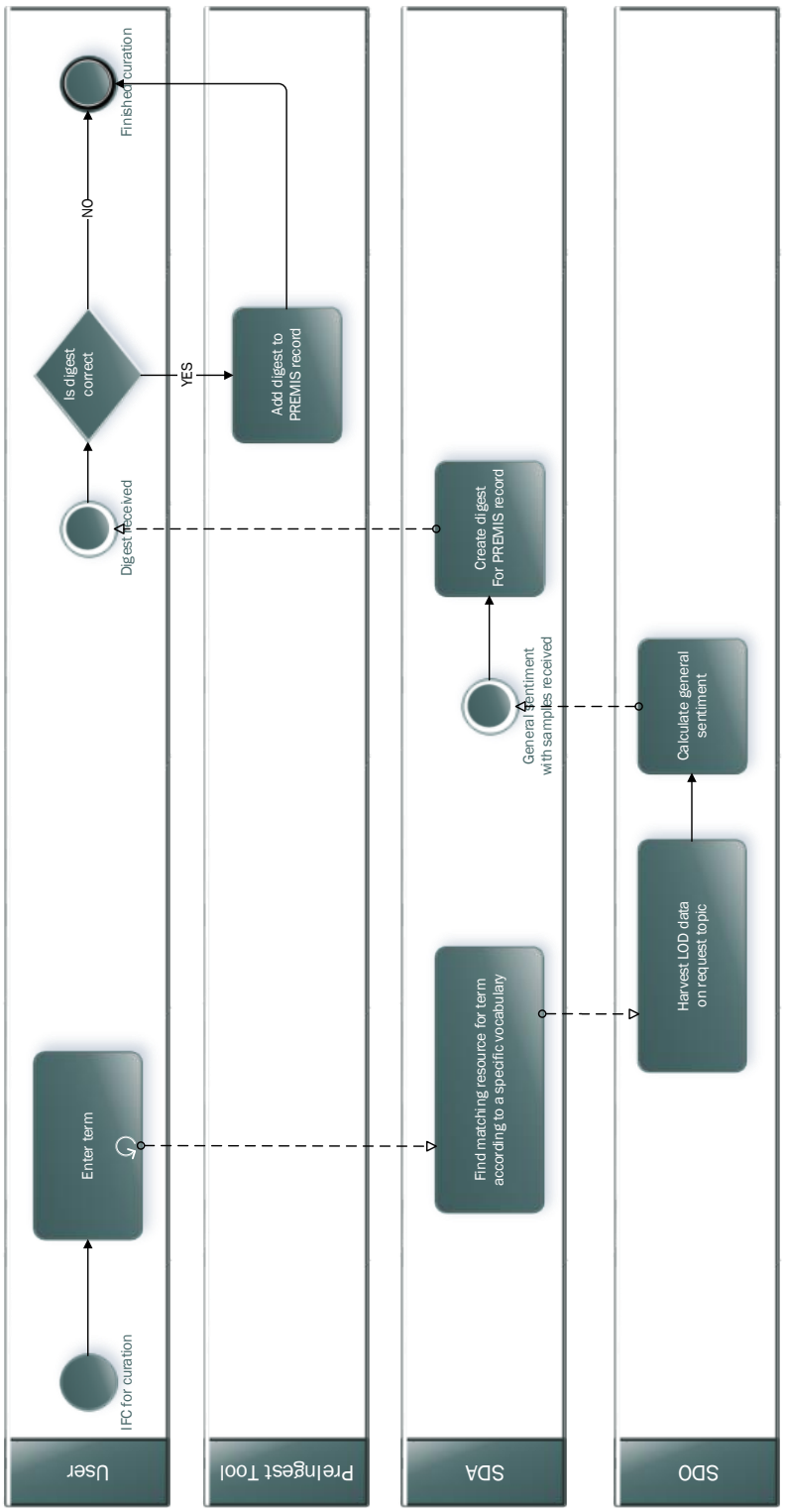

Figure 13: Typical curation workflow for enriching an IFC with LOD data.

category are meta-data links provided in archive description formats such as METS and PREMIS that enrich entity instances found in IFC files (and provided either as SPFF or as RDF serializations). A concrete example is the aggregate gross floor area of a building that might not be captured explicitly in an IFC files at the time of ingestion but is determined by the DURAARK ingest tool suite and captured in the archive meta-data description by pointing to an IfcBuilding instance found in the preserved file.

For both of these semantic link categories the original pointers, e.g. URIs or IRIs are redirected into the SDA using e.g. a URI strategy.

More detailed information about the different forms of semantic linking as well as concrete examples can be found the respective deliverables D.3.3.1 and D.3.3.2.

4.2.3 Components

Meta-data extraction from IFC model component A considerable number of information facets that are relevant to indexing and searching of information in an archival context is already available in the IFC model itself. In accordance to D3.3.1 this component will parse the relevant information in the prepared IFC model and map them into meta-data sets that are exposed to the archival system. This meta-data includes authorship, software tools involved in the creation of the data, building components and their manufacturers and the intended functions of spaces

- **INPUT** IFC model.
- **OUTPUT** Meta-data schema instance (RDF) enriched with information extracted from the IFC model.

Semantic enrichment (context) of BIM component This component allows the access to external datasets of architectural relevance (geodata, energy-efficiency policy, related transport, related infrastructure etc) and the creation of links between entities in these datasets and individual building elements, spaces or the building as a whole. This component focuses on non-engineering data that is publicly available, extracted from unstructured Web information or added by e.g. librarians or archive curators such as the

classification of architectural styles, social and historic context information and public perceptions (see Figure 14).

- **INPUT** IFC model.
- **OUTPUT** Meta-data schema instance (RDF) enriched with information extracted from the IFC model and from additional external resources stored / mirrored in the Semantic Digital Archive.

Semantic enrichment (technical) of BIM component This component allows the access to external datasets and the creation of links between entities in these datasets and individual building elements, spaces or the building as a whole. This component focuses on technical engineering data on high levels of detail. Examples of external datasets include classification systems, building regulations, product data bases, and concept repositories such as the buidlingSMART data dictionary bsDD. Snapshots of the content of the linked data will be added to the SDA. Resources available in the SDA can also be accessed directly. The component is either a stand-alone post-processing tool such as Catenda's 'Bimsync' or integrated into a modeling CAD application.

- **INPUT** IFC model with of-the-shelf information generated by modeling (CAD) packages.
- **OUTPUT** IFC model enriched with semantic information from external data sources.

SDA curation component
This component allows the preparation of datasets and their storage in an archive which provides self-containing snapshots of external vocabularies and datasets that can be shared by ingested archival packages.

- **INPUT** URI of external vocabulary or dataset.
- **OUTPUT** Self-containing snapshot of external vocabulary and dataset.
- **OUTPUT** Update in registry of Semantic Digital Archive.

Interlinking & clustering component This component is the main processing part for the semantic enrichment. In addition to the meta-data extracted from information

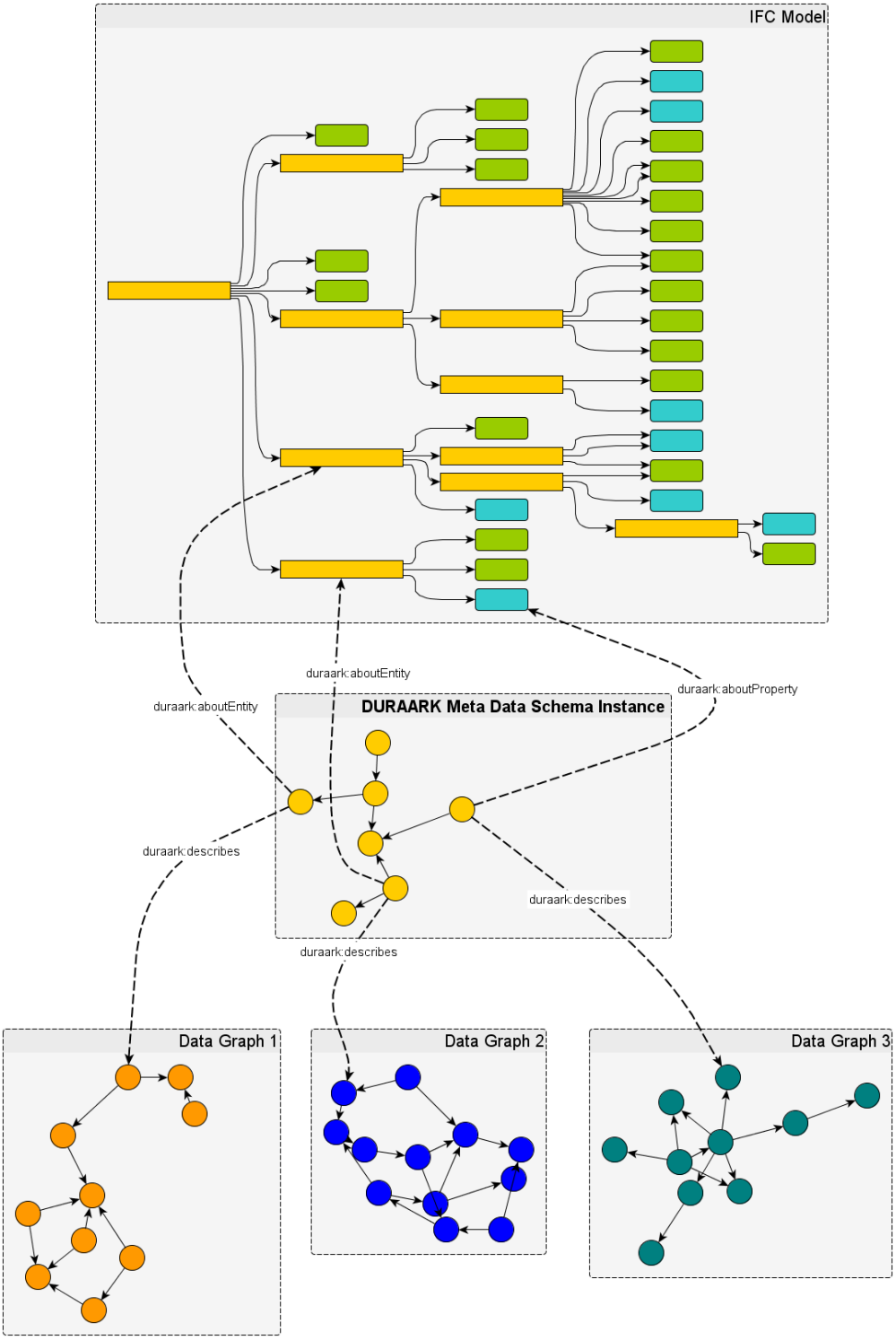

Figure 14: Meta-data schema as a hub for linking external information to IFC models.

explicitly present in the underlying IFC model, a number of aspects relevant to indexing is not available in the model itself but will be retrieved in a semi-automated way. It will deploy mechanisms which identify correlations between data and models and make these links explicit. Examples for such interlinking and clustering scenarios are:

- to create links between different structural models and architectural models which are submodels of the same building,
- different models which represent the same structure at different points in time,
- energy efficiency data and the building it relates to,
- two structures sharing similar geodata or similar correlated information where explicit links are required in order to ensure consistency.

As such, the main purpose of the interlinking and clustering component is to populate the SDA with richer, and more coherent data, i.e. to produce a more coherent *RDF graph of architectural models and related information*. With respect to enrichment and interlinking with external data, such information may include geographical context, related transport and infrastructural information, environmental data, historic information about the structure and its surroundings, public perceptions of structures and their evolutions and similarly relevant information that has to be attached to the mere engineering information covered in the BIM/IFC model by external experts.

In the context of the Semantic Web, a large number of datasets can be harnessed to add such information. These are currently under investigation as part of WP3 activities. The meta-data schema developed in WP 3 allows the creation of links. The software component allows the enrichment of an ingested IFC/A dataset with such information. While heterogeneity of datasets (at the schema and instance-level) is a major obstacle, state of the art schema matching[21][12] and entity linking techniques[19] will be deployed to detect and capture links between disparate datasets. In addition, feature-based clustering and machine learning techniques will be deployed to identify links between related models and data.

- **INPUT** IFC model.
- **OUTPUT** Meta-data schema instance (RDF) enriched with information extracted from the IFC model and from additional external resources stored / mirrored in the Semantic Digital Archive.

4.2.4 Technical aspects

The components for the semantic enrichment of datasets such as the Semantic Digital Archive itself, the pre- and post-processing tools concerning semantic the IFC model instances and the Semantic Digital Observatory (SDO) will mainly be developed in using the Java platform tools and libraries. The rationale behind this decisions is:

- Most Semantic Web related software libraries, commercial and open source are written in Java. Popular examples include Jena, OWLAPI, triple store servers, RDF(S)/OWL editors, visualization tools, XML processing tools etc.
- many IFC processing tools such as the bimserver.org framework (co-developed by one of the DURAARK partners and commercialized by another), JSDAI etc are available on the Java platform
- many digital preservation tools and libraries are based on Java

The following libraries and platforms have been chosen for the tasks in

RDF(S) and OWL processing A number of different libraries for general purpose RDF and OWL processing will be used. These include

- Apache Jena, a Semantic Web an Linked data libraries framework [25]
- OWL API: [26]

but will be depending on the individual requirements during the implementation phase.

Tripple stores which will be the core of the semantic digital archive to store, serve and query stored vocabularies used for semantic enrichment. These include

- Virtuoso[27]
- Sesame[28]

STEP and IFC tools that will be needed to validate ingested IFC models and extract meta-data from. These include

[25] `http://jena.apache.org/`
[26] `http://owlapi.sourceforge.net/`
[27] `http://virtuoso.openlinksw.com`
[28] `http://www.openrdf.org/`

- Bimserver.org[29], a framework including some efficient parsers, query- and other processing tools for IFC files and have been partly been developed at TUE
- JSDAI[30], an Open Source general purpose STEP data processing tool that allows early- and late-binding processing of instance files

Archival tools

- JHOVE[31] to wrap extraction and validation tools in standard conform interfaces that can be used with other archival implementations
- BagIt Library[32] to structure archival packages

[29] http://www.bimserver.org
[30] http://www.jsdai.net
[31] http://jhove.soureforge.net
[32] https://github.com/LibraryOfCongress/bagit-java

4.3 Data preservation

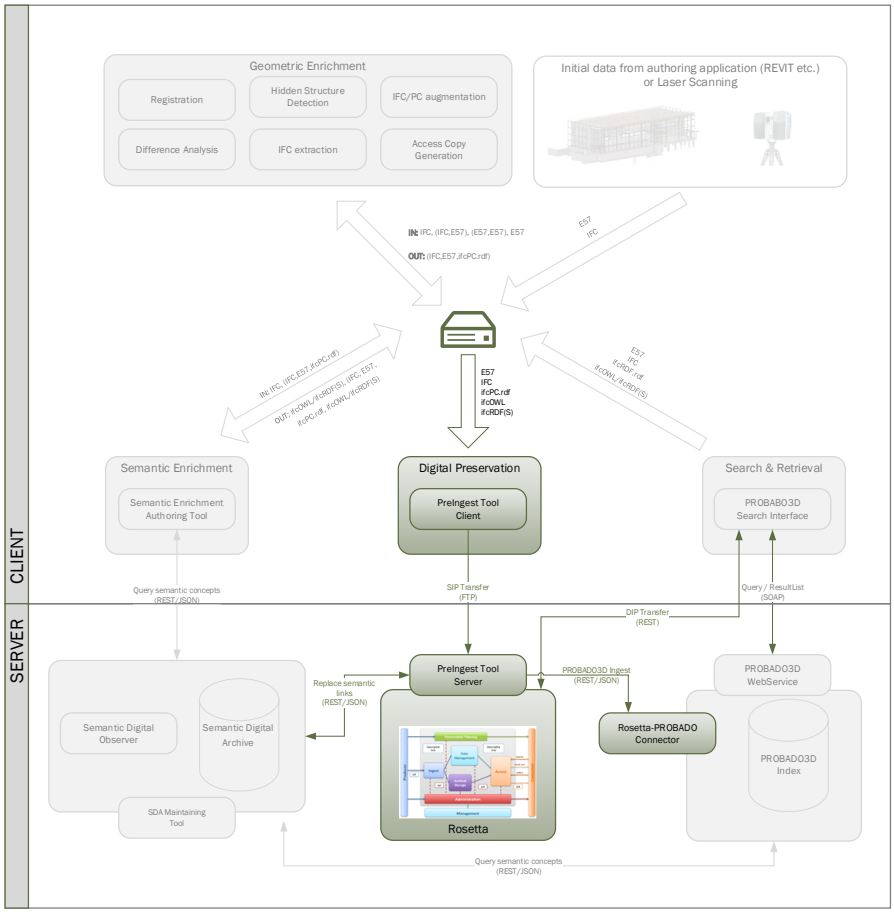

Figure 15: Components and interfaces related to the DURAARK DPS.

This section will give a more detailed description of the ingest aspects of the DURAARK digital preservation system along with a refined component list. Figure 15 highlights the components and interfaces related to the DURAARK data preservation.

4.3.1 Workflow data ingestion

The pre-ingest tool includes a component which allows the user to choose a target for SIP submission. The packages can either be submitted to the digital preservation system hosted at TIB or to an arbitrary (local) directory or sftp/ftp site chosen by the user. The proof of concept conducted within the DURAARK project will deposit the SIPs to the

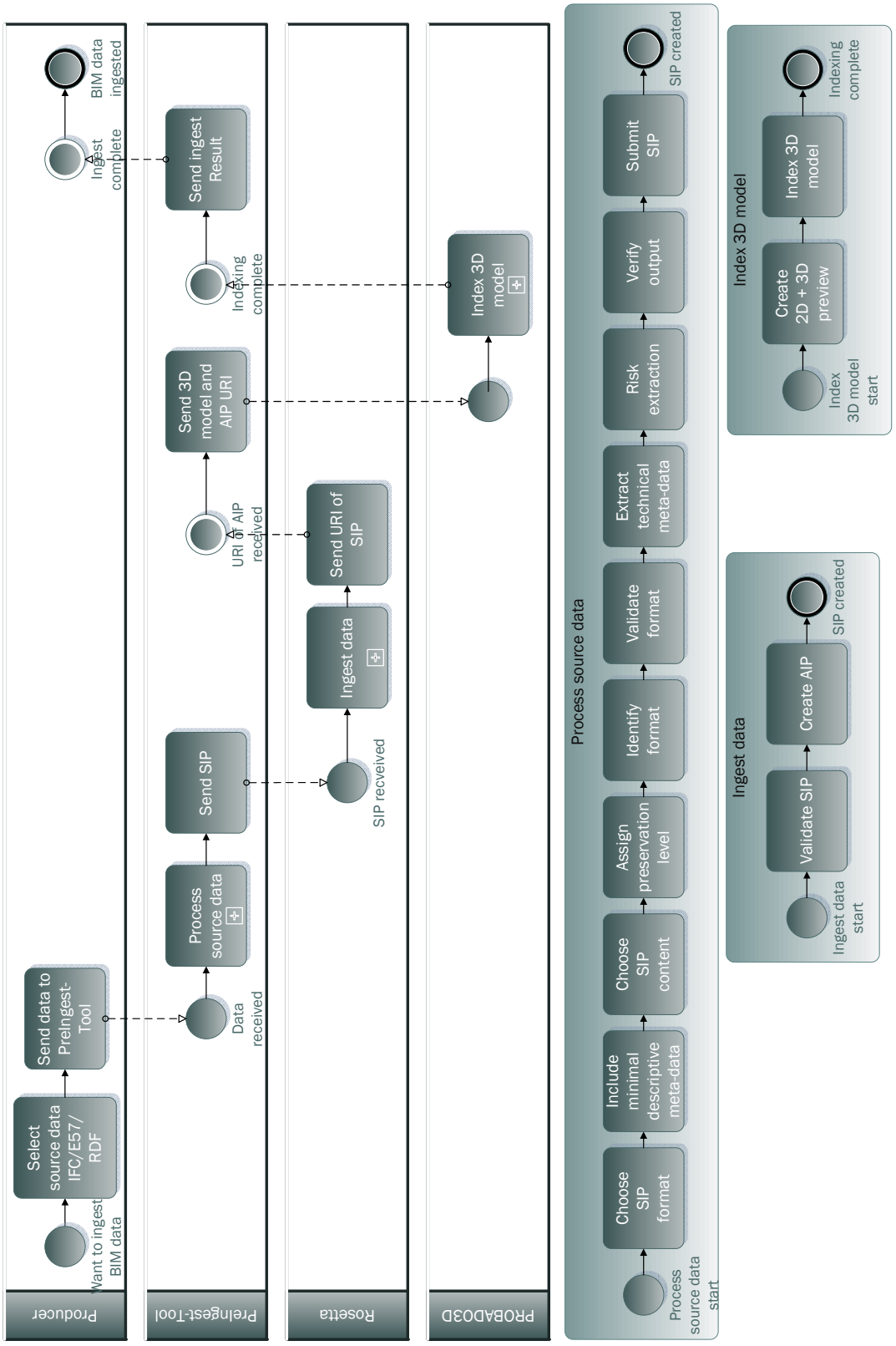

Figure 16: Ingest of architectural data into the DPS.

Rosetta system at TIB. The pre-ingest tool may transfer the SIPs to an ftp server which is used as an entry point for package submission into the digital preservation system hosted by TIB. Figure 16 shows the complete ingestion process.

The SIP submission to the digital preservation system Rosetta is possible via a submission application which uses the SDK provided by Ex Libris for communication with the Rosetta Deposit API. The API allows to deposit the SIP to the preservation system and to query the system for the status in the ingest process, i.e., whether the SIP could be successfully ingested and the URI of the SIP within the preservation system. For dissemination, the same URI can be used by an external discovery system, such as PROBADO3D, to retrieve the object in a DIP from the digital preservation system.

4.3.2 Data Formats

As discussed in section 3.6, IFC-SPF and E57 were chosen as producer input file formats. Further information generated by the geometric and semantic enrichment components will be available as ifcOWL and ifcPC.rdf information and will also be a part of the submission information package (SIP) to be submitted to an OAIS compliant digital preservation system.

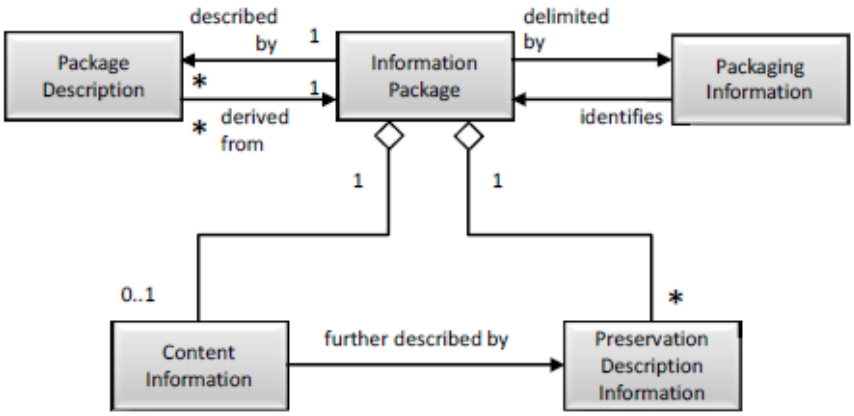

Figure 17: OAIS information package taxonomy [10]

A SIP is an information package which is to be submitted to an OAIS to be preserved for a designated community. A SIP contains content information, which consists of the data

object(s) and accompanying representation information, preservation description information. The OAIS defines preservation description information as information required for the preservation of content information, such as provenance, context or access rights information.

The OAIS explicitly recommends PREMIS as a standard for the submission of preservation meta-data. Within the PREMIS data model, the data object is referred to as an "intellectual entity". An intellectual entity (IE) is a single or combination of information objects which the archive shall consider a single intellectual unit. An intellectual entity may include other intellectual entities, e.g, an encyclopaedia can include books, a book can include an entry. One or more representations are possible for each intellectual entity[22].

Within the context of a DURAARK SIP the following holds true: A DURAARK SIP contains one or more intellectual entities. One intellectual entity is a capture of one architectural structure, i.e., a building or a part of a building, at a point in time. Within an intellectual entity one or more representations of an object are possible.

The DURAARK system will support two SIP output formats: SIPs to be delivered to the OAIS compliant digital preservation system "Ex Libris: Rosetta" and SIPs to be transferred to an arbitrary OAIS compliant system via a BagIt package.

Rosetta SIP The structure of a Rosetta SIP is shown in Figure 18. The dc.xml file contains basic descriptive information at a SIP level, e.g., person and organization who created the SIP. The content directory holds the information object. Each intellectual entity within the SIP is described in a METS xml file. Ex Libris has registered the METS profile used in Rosetta on the METS standards site.[33]

Descriptive information within the Rosetta METS file has to be provided in Dublin Core - both, the simple[34] and the qualified[35] Dublin Core schemas are supported. Preservation meta-data within the Rosetta METS file is captured in the Ex Libris proprietary DNX standard, which is based on PREMIS. Furthermore, the Rosetta METS allows for the

[33] http://www.loc.gov/standards/mets/profiles/00000038.xml
[34] `http://dublincore.org/schemas/xmls/qdc/2003/04/02/dc.xsd`
[35] `http://dublincore.org/schemas/xmls/qdc/2003/04/02/dcterms.xsd`

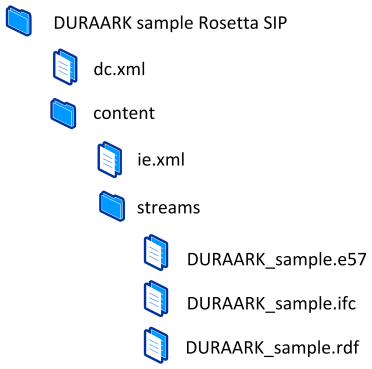

Figure 18: Sample Rosetta SIP

integration of other meta-data types, such as MARC, MODS, EAD, NISO or MIX. The data objects themselves are contained in the streams directory.

BagIt BagIt is a hierarchical file packaging format originally developed by the Library of Congress and the California Digital Library for the transfer and storage of digital content. The specification contains the objects as "payload" which is accompanied by optional "tags" describing them. The BagIt specification is currently available as a Internet Engineering Task Force (IETF) draft [18].

A "bag" consists of a number of required and optional directories and files. Figure 19 shows an exemplary BagIt SIP. The manifest description "bagit.txt" describes the BagIt version and the file encoding, the latter must be UTF-8 according to the current specification version 0.97. The "data" directory contains the payload, or the data objects themselves. BagIt makes no assumptions about the content of files - METS XML files, for example, are therefore included within the data directory.

For the contents of the data directory, each file's integrity should be documented with a checksum. In the example, "manifest-md5.txt" contains the filename and a checksum for every file in the data directory. The checksum can be calculated in any algorithm; the algorithm used must be stated in the filename (e.g., manifest-sha1.txt, manifest-crc.txt). "Tagmanifest-md5.txt" contains the filename and checksum for every "tag" file. In the example, the only tag file present is bagit-info.txt, which contains basic descriptive information about the SIP, e.g., person and organization who created the SIP.

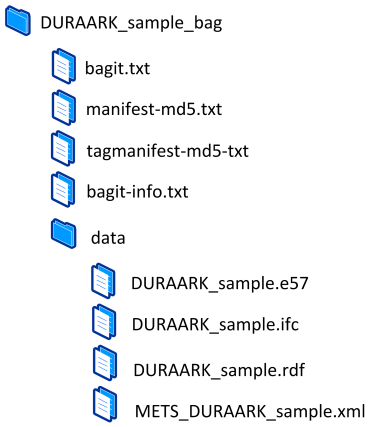

Figure 19: Sample BagIt Packet

4.3.3 Components

This section describes the preservation components within the DURAARK system. The "preservation" group of components describes a pre-ingest workbench, which analyzes and treats objects prior to their ingest into a digital preservation system.

The following components can be grouped into three categories: One category deals with the orchestration of the SIP - choosing the SIP output format, including a minimal set of meta-data, choosing the SIP content, assigning preservation levels to the content and verifying the output to be submitted to an OAIS compliant archive. Especially for complex SIPs containing several objects, the benefit of adding structural and contextual meta-data at an early stage in the lifecycle process can be seen as a step from object based archiving towards the archiving of digital codified knowledge [16].

The second category consists of components dealing with the interpretability and soundness of the objects at a logical level: file format identification, validation, technical meta-data extraction and risk extraction. While these steps may be included within the OAIS compliant system as part of the ingest and data management functions, pushing them to the pre-ingest stage stresses the producer awareness and involvement in qualitative aspects of the data. In conjunction with the components described in the geometric and semantic enrichment sections, the producer controlled preservation-ready SIP creation also forms a major step towards the aforementioned archiving of digital codified knowledge.

The last category is the integration between the OAIS system and the digital preservation system in form of the Rosetta-PROBADO connector. This component is highly specific to the DURAARK system architecture.

Choose SIP output format The preservation tasks in the pre-ingest workbench start with the user choosing the target SIP output format. To enable a wider adoption beyond the spectrum of the integration with Rosetta as an OAIS compliant digital preservation system, the DURAARK system will also support the optional output of the SIP as a BagIt package. BagIt is used in a number of digital library and digital preservation solutions, such as in the open source digital preservation solution Archivematica [36]. Depending on the option chosen, the system will prepare the SIP structure and move the files into the appropriate directories within the SIP.

- **INPUT** "BagIt" or "Rosetta SIP".
- **OUTPUT** Create SIP structure in form of directory structure and necessary files for meta-data or manifest descriptions.

Include minimal set of descriptive meta-data Within an OAIS a basic set of descriptive elements is needed to put the digital objects into context. If these elements are not already supplied with the data to be included in the SIP, the producer will be asked to provide the data at this stage of the process.

- **INPUT** "BagIt" or "Rosetta SIP".
- **OUTPUT** The minimal set of descriptive meta-data is captured in descriptive meta-data.

Choose SIP content This component should allow the user to choose which of the files originally uploaded or created as part of other DURAARK components, such as at the geometric or semantic enrichment stage, should be included in the OAIS to be submitted to the OAIS compliant digital preservation system.

- **INPUT** List of files currently intented for SIP inclusion.

[36] `https://www.archivematica.org/`

- **OUTPUT** List of files to be included in SIP mapped to structural meta-data. Files moved to respective directory in SIP structure.

Assign preservation level to files Within the project different representations of the same content were identified as possible candidates for a SIP: E57, IFC, ifcOWL and ifcPC.rdf. Additionally, it may be possible that a producer wants to store files of the same format family in different quality levels, e.g., to include an ifc.xml file or a compressed E57 file as dedicated access copies in the archive. Acces copies serve the main purpose of access by the consumer. As such, they may contain features such as compression which may not be considered suitable for preservation purposes and are therefore only kept in the archive for as long as necessary. They may be replaced by new access copies generated from the preservation master over the course of time. To be aware of the producer's intended preservation level and to treat the files accordingly within the archive, the content of the SIP should be labeled appropriately in structural meta-data.

- **INPUT** List of files to be included in SIP.
- **OUTPUT** Labels indicating preservation level ("Preservation Master" / "Access Copy") added to structural meta-data.

Identify format To be able to preserve digital objects on a logical level, knowledge about the file format is essential. Pushing file format identifcation to the pre-ingest level allows producer input and awareness of how the file is to be recognized within the archive. Common file format identification tools used in archives, such as the unix file utility, DROID or fido currently do not support E57 or IFC files. The "identify format" component uses an extended file format identifcation tool and captures information about the tool and outcome for each SIP content file in meta-data. Ideally, the file format information will include an ID associated with a global file format registry, such as the PRONOM Unique Idenfier (PUID).

- **INPUT** Files to be included in SIP.
- **OUTPUT** For each file in SIP: Identification tool used and file format information outcome added to administrative/preservation meta-data.

Validate formats Like file format identification, file format validation is a central aspect of the preservation of objects at a logical level. File format validation checks standard and schema conformity of objects. The output of validation components may be broken down into statements whether an object is "well-formed" and whether an object is "valid". Well-formedness refers to the low level syntax of an object. For example, XML files are considered well-formed when the object adheres to the syntax rules specified in the XML specification. These syntax rules, that a document has a single root element, define, that each element must have a closing tag and that elements are properly nested. The XML specification does not, however, pre-define a set of tags or attributes. In the case of XML this may be done via a schema. The conformity check against a schema determines whether an XML is "valid". Based on this, an object can only be "valid" when it is "well-formed". Lack of well-formedness and/or validity have an impact on preservation capabilites. Pushing file format validation to the pre-ingest level allows producer awareness and input to counteract possible problems in maintaining the renderability of the object, e.g., in form of re-starting the process with a valid file.

- **INPUT** Files to be included in SIP.
- **OUTPUT** For each file in SIP: Validation tool used and outcome "Well-formed and valid" or "Well-formed and not valid" or "Not well-formed and not valid" with error that triggered output added to administrative/preservation meta-data.

Extract technical meta-data Technical meta-data extraction addresses general or content type-specific information which is needed to maintain the ability to render the object or to describe the object's quality or behavior. Examples for general technical meta-data are the extraction of XMP meta-data if included or single information values like the creating application. Examples for content type-specific information are the encryption algorithm (if applied), object count and fonts for PDF or other textual formats, color mode (e.g., CMYK, RGB, grayscale, bitonal), bitdepth, image length and image width for raster images and sample rate, number of channels and essence encoding (e.g., PCM, mp3) for audio files. Making technical meta-data available at the pre-ingest level may allow the user to check quality factors about the file such as the presence of descriptive meta-data like authorship within the objects embedded meta-data.

- **INPUT** Files to be included in SIP.

- **OUTPUT** For each file in SIP: Extraction tool used and extracted technical meta-data in form of respective fields and values added to administrative/preservation meta-data.

Risk extraction - check file dependencies Risk extraction is similar to technical meta-data extraction. However, whereas technical meta-data extraction captures a lot of information as separate values, risk extraction addresses a single risk at greater detail. Such a risk is the file dependency on embedded or referenced files, such as textures either integrated as a binary blob (bmp, jpg, gif, png) or referenced via a URI. The risk extraction/file dependencies component will analyse the object for any dependencies and give further information about the objects embedded or referenced. Pushing risk extraction to the pre-ingest level enables the producer to be aware of possible preservation risks of the objects, i.e. dependency on external resources.

- **INPUT** Files to be included in SIP.
- **OUTPUT** Number of embedded or referenced objects, availability of embedded or referenced objects, type of embedded or referenced objects

Verify output After all pre-ingest components have been completed, the suggested SIP output should be verified by the producer. In the light of trustworthy digital preservation a verified and producer approved SIP content is of high relevance. The producer verification of the SIP is the last point in the process before the SIP is moved to the digital preservation system of the target chosen in the first pre-Ingest step.

- **INPUT** SIP structure and content.
- **OUTPUT** Finalized SIP structure and content.

Submission to target The DURAARK pre-ingest system supports two targets: the digital preservation system hosted at TIB which is based on Rosetta or an arbitrary (local) directory or ftp site the user has access to. The form of the SIP submission is determined by the choice made within the first component. The system should be able to collect SIPs and schedule jobs to deliver them to the integrated digital preservation system or to the respective directory at a definable point in time.

- **INPUT** Target and optional parameters.
- **OUTPUT** SIP submitted to target at predefined time or right away.

Connector PROBADO - Rosetta The connector between PROBADO and Rosetta is not directly part of the pre-ingest platform exposed to the user but a background component which runs during the submission of the objects to the digital preservation system Rosetta. It is further specified in section 4.4. To enable retrievability of the objects from the PROBADO platform, PROBADO needs to hold information about the ID of the objects within Rosetta as well as an index covering information to be made searchable from the PROBADO platform. The ID will enable PROBADO to generate a link to the Dissemination Information Package (DIP). The ID is ideally captured during the submission job and passed to PROBADO from there.

- **INPUT** SIP
- **OUTPUT** ID of object in Rosetta

4.3.4 Technical aspects

Besides Rosetta as the platform for the OAIS compliant digital preservation system the following libraries and platforms will be used

STEP and IFC tools that will be needed to validate ingested IFC models and extract meta-data from (see section 4.2.4 details).

Archival tools

- JHOVE to wrap extraction and validation tools in standard conform interfaces that can be used with other archival implmentations
- BagIt Library[37] to structure archival packages

[37] `https://github.com/LibraryOfCongress/bagit-java`

4.4 Search & Retrieval: PROBADO3D

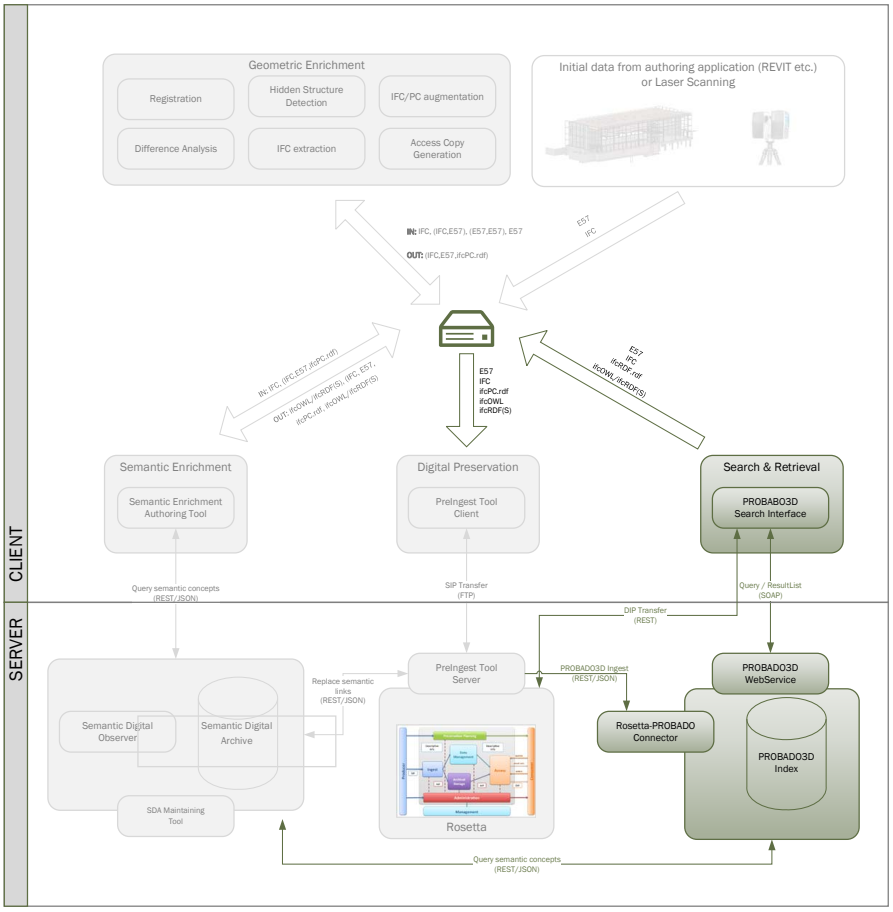

Figure 20: Components and interfaces related to Search & Retrieval

This section will give a detailed description of the search & retrieval aspects of the DURAARK system (see Figure 20).

4.4.1 Workflow search & retrieval

The interaction for searching/browsing the archived data is shown in Figure 21. The user formulates the query; either by entering free keywords or using a classification interface. The query is then sent to the PROBADO3D web-service, which returns a result list for the query from the available indexer. Every item in the result list contains a link to the archived data within Rosetta. If the user wants to access an item from the list of

results, he can use this link in order to transfer the data to his client system for further processing.

4.4.2 Components

PROBADO3D Search Interface PROBADO3D supports search in meta-data space, as well as in content-based space in 3D architectural data comprising models of buildings[7]. Currently the following query interfaces are implemented within PROBADO3D:

- searching in the textual meta-data of the objects (title, description, etc)
- browsing the repository content using different filters (category, contributor, etc.)
- upload of a model for a query-by-example
- graph-based search using a 2D interface for constructing room connectivity graph queries
- an interactive 3D modeling environment for formulating 3D queries

Within the DURAARK project existing search interfaces will be modified to address the needs of the use-cases, for example using the browse facilities of PROBADO3D for exploring the vocabularies of the SDA as shown in Figure 22 left. Figure 22 right shows the map-based search, which can be used for the urban context exploitation or selecting specific buildings for retro-fitting/energy renovation.

PROBADO3D IFC Indexer Due to PROBADO's original focus on legacy 3D CAD formats, PROBADO3D currently does not support indexing or preview generation (2D thumbnails, PDF 3D) for IFC files and point-cloud data. In order to address the search & retrieval requirements from the different use-cases, PROBADO3D needs to be enhanced to handle IFC and point-cloud data. Also the time-line aspects (e.g. that one IFC model has a number of associated point-cloud scan) needs to be addressed in the design of the indexer.

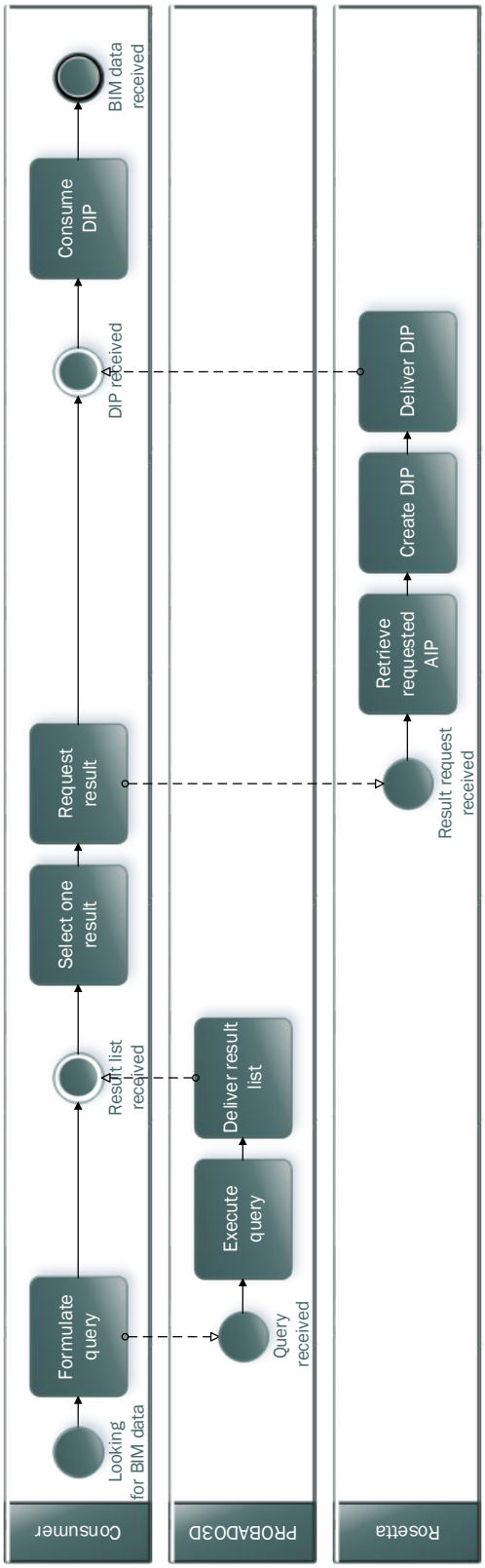

Figure 21: Search and retrieval of archived data.

Figure 22: (left) PROBADO3D interface for browsing categories (right) map-based search.

4.4.3 Technical aspects

Based on the constraints of PROBADO3D the server-side components will be implemented using the C# programming language. The client-side components - the PROBADO3D Search Interface - will be implemented using HTML (used for web-access).

In addition, the following libraries/frameworks are used:

- IfcOpenShell (`http://ifcopenshell.org/`) for working with IFC files and conversion of the contained BIM models to mesh geometry.
- WCF (Windows Communication Foundation), used for the SOAP web services.

5 Conclusion

In this deliverable the next iteration on the system architecture of DURAARK was presented. The constraints and decisions towards this architecture from D2.2.2 have been extended - among others - by the strategic decision on file formats. Based on the Curation Lifecycle Model the use-cases from D2.2.1 were categorized and aligned with the stakeholders and components from D2.2.2. The various data-formats for encoding and exchanging the information of the geometric and the semantic enrichment have been described. Finally, a detailed look at the ingest and retrieval process concerning the digital preservation system was presented.

The system architecture will continue to evolve during the course of this project. The next deliverable in month 18 will be the first prototype of the DURAARK system marking the next iteration in the DURAARK system architecture.

References

[1] ASTM International Technical Committee E57. http://www.astm.org/COMMITTEE/E57.htm.

[2] libE57: software tools for managing e57 files (ASTM e2807 standard). http://www.libe57.org/.

[3] Messaging patterns in service-oriented architecture, part 1. http://msdn.microsoft.com/en-us/library/aa480027.aspx, 2004.

[4] H. Beedham, J. Missen, M. Palmer, and R. Ruusalepp. Assessment of ukda and tna compliance with oais and mets standards. Technical report, JISC, 2005.

[5] J. Beetz and B. de Vries. Building product catalogues on the semantic web. *Proc." Managing IT for Tomorrow*, page 221–226, 2009.

[6] R. Berndt, I. Blümel, M. Clausen, D. Damm, J. Diet, D. W. Fellner, C. Fremerey, R. Klein, M. Scherer, T. Schreck, I. Sens, V. Thomas, and R. Wessel. The probado project – approach and lessons learned in building a digital library system for heterogeneous non-textual documents. In M. Lalmas and et al., editors, *Research and Advanced Technology for Digital Libraries, 14th European Conference ECDL. Proceedings ECDL 2010*, volume 6273, pages 376–383. Springer, 2010.

[7] R. Berndt, I. Blümel, H. Krottmaier, R. Wessel, and T. Schreck. Demonstration of user interfaces for querying in 3d architectural content in probado 3d. In *Research and Advanced Technology for Digital Libraries*, pages 491–492, 2009.

[8] R. Berndt, H. Krottmaier, S. Havemann, and T. Schreck. The probado-framework: Content-based queries for non-textual documents. In *ELPUB 2009: 13th International Conference on Electronic Publishing*, page 17, 2009.

[9] M. Böhms, P. Bonsma, M. Bourdeau, and A. S. Kazi. Semantic product modelling and configuration: challenges and opportunities. *ITcon Special Issue Next Generation Construction IT*, 14:507–525, 2009.

[10] CCSDS. Reference model for an open archival information system (oais) - magenta book. Technical report, 2012.

[11] M. Crawford, R. O'Leary, and J. Cann. *Uniclass: Unified Classification for the Construction Industry*. Riba Publications Limited, 1997.

[12] M. D'Aquin, A. Adamou, and S. Dietze. Assessing the educational linked data landscape. In *ACM Web Science 2013 (WebSci2013), Paris, France*. ACM, 2013.

[13] M. Dolenc, P. Katranuschkov, A. Gehre, K. Kurowski, and Z. Turk. The InteliGrid platform for virtual organisations interoperability. *ITcon*, 12:459–477, 2007.

[14] B. R. Florian and K. S. W. C. Martin, editors. *Linked Open Data: The Essentials - A Quick Start Guide for Decision Makers*. edition mono/monochrom, Vienna, Austria, 2012.

[15] S. Higgins. The dcc curation lifecycle model. *The International Journal of Digital Curatoin*, 3(1):134–140, 2008.

[16] T. Kärberg. Digital preservation of knowledge in the public sector: a pre-ingest tool. *Archival Science*, pages 1–13, 2013.

[17] C. Lima, C. F. Da Silva, C. Le Duc, and A. Zarli. A framework to support interoperability among semantic resources. In *Interoperability of Enterprise Software and Applications*, page 87–98. Springer, 2006.

[18] Network Working Group. The bagit file packaging format (v, October 2013. Internet-Draft.

[19] B. P. Nunes, S. Dietze, M. Casanova, R. Kawase, B. Fetahu, and W. Nejdl. Combining a co-occurrence-based and a semantic measure for entity linking. In *ESWC 2013 - 10th Extended Semantic Web Conference*, May 2013.

[20] M. Pennock. Digital curation: A life-cycle approach to managing and preserving usable digital infor. *Library & Archives Journal*, 2007.

[21] B. Pereira Nunes, A. Mera, M. Antonio Casanova, B. Fetahu, L. A. P. Paes Leme, and S. Dietze. Complex matching of rdf datatype properties. In *In Proceedings of 24th International Conference on Database and Expert Systems Applications*, 2013.

[22] PREMIS Editorial Committee. Premis data dictionary for preservation metadata, version 2.2. Technical report, 2012.

[23] R. Ruusalepp and M. Dobreva. Digital preservation services: State of the art analysis. Technical report, DC-NET, 2012.

[24] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226, June 2007.

[25] M. Smith. MIT FACADE Project Final Report. Technical report, August 2009.

[26] S. Strodl, P. Petrov, and A. Rauber. Research on digltal preservation within project co-funded by the european union in the ict programme. Technical report, Vienna University of Technology, May 2011.

[27] F. Tolman, M. Böhms, C. Lima, R. van Rees, J. Fleuren, and J. Stephens. eConstruct: expectations, solutions and results. *Electronic Journal Of Information Technology In Construction (ITcon)*, 6:175–197, 2001.

[28] UK Data Archive. *Preservation Policy*, 2012.