



## Earth Observation Image Librarian ARCHITECTURE CONCEPT

Prepared By

\_\_\_\_\_  
Vlad Manilici (DFD)

\_\_\_\_\_  
2013-04-04

\_\_\_\_\_  
Daniela Espinoza-Molina (IMF)

\_\_\_\_\_  
Date

Reviewed By

\_\_\_\_\_  
Mihai Datcu (IMF)

\_\_\_\_\_  
Date

Released By

\_\_\_\_\_  
Dietmar Preuß (DFD)

\_\_\_\_\_  
Date

© **Restriction of Disclosure:** All rights reserved. No part of this document may be reproduced, in any form or by any means, without permission in written form by the EOLib Project Manager of the German Aerospace Center (DLR).

## DOCUMENT CHANGE RECORD

Issue	Date	Where	Change
1.0	2012-11-27		First release
1.1	2012-03-05		<u>Answers to review questions risen during the preliminary design review meeting.</u>
		Figure 1	The Architecture Baseline diagram was improved in order to explain some previously unclear points: <ul style="list-style-type: none"> <li>• The Semantic Annotation component is explicitly shown within KDD</li> <li>• Product Selection for Reprocessing part of OT (shown explicitly)</li> <li>• A DIMS Order Handling component is now shown in order to explain how an Epitome order is passed from the Web Server (GDAS) to OPG (DP's request)</li> <li>• The pick-up point for deliveries was added</li> <li>• Processing and Reprocessing were merged</li> <li>• Epitome Generation only accesses LTA</li> </ul>
		§3.7	MI1: Relevance metrics Added example and referred the reader to the upcoming detailed design (footnote).
		§4.4	MI2: Similarity measures Refer the reader to the upcoming detailed design (footnote).
		§4.4	MI3: CBIR Explained in the text.
		§4.8	MI4: System Evaluation System evaluation will be detailed during the design phase.
		§5.2	MI5: Patch / Tile size Answered as proposed in the final disposition.
		§5.7	MI6: EOLib Ontology / Taxonomy (1) Replaced "ontology" with "taxonomy".
		§6.2	MI7: EOLib Ontology / Taxonomy (2) Replaced "ontology" with "taxonomy".
		§7.1 Figure 1	MI8: EOLib interface (3) Subject to upcoming system design; partly addressed in Figure 1.
		§2.4	MI-C01: Operator role and privileges (2) UIF is an automatic, internal interface: explained.
		§3	MI-C02: GDAS meaning Explained on first occurrence and in the corresponding chapter.
		Figure 1	MI-C03: LTA component (3) Added missing interface designation for LTA-EOWEB.
		§3.4	MI-C04: Online User Services link More details added, both in Fig. 1 and in the text.
		§3.4 Figure 1	MI-C05: Semantic catalogue upload Added to Figure 1
		§3.4 Figure 1	MI-C06: Online User Services components (2) Added to Figure 1.
		§3.6 Figure 2	MI-C07: Epitome Browser Updated figure; explained in text that the browser is separately downloadable.
		§3.7	MI-C08: EOLib interface (1) No other connections for the moment; Yet, the architecture is not frozen.
		§3.7 Figure 1 Figure 2	MI-C09: EOLib interface (2) Added Tiling to the Arch. Baseline Fig.
		§4.2	MI-C091: Data mining functions Added short overview and refer the reader to the DB TN document.
		§1.1	MI-C092: GIS data processing (2) Added explanation in text.
		§2.2	MI-C1.1: LTA component (1) Long-time archival (LTA) vs. speedy, ad-hoc access (DM-DB) explained. Added explanation of the incremental update of DM-DB from LTA for consistency.
		§2.2	MI-C1.2: LTA component (2) No direct link from DMI to the EOLib interface: LTA is the long term, "master" repository.
		§2.2	MI-C1.3: Bulk Product Selection component Add descriptions for all components shown in the Arch. Baseline.



**Earth Observation Image Librarian  
Architecture Concept**

ID EOLIB-TN-DLR-3100  
Issue 1.1  
Date 2013-04-04  
Page 3 of 28

Issue	Date	Where	Change
		§2.2	MI-C2: Online User Services components (1) 1. EOLib functionality now shown in the Online User Services (orange) 2. A transaction-like locking mechanism from DM-DB to LTA was added.
		§2.2	MI-C3: Epitome Access Explained: Epitome browser is downloaded separately; Epitome accessed off-line.
		§2.2	MI-C4: EOLib interface role (1) Explained
		§2.2 Figure 1	MI-C5: Eolib Interface role (2) Tiling is now an explicit functionality in the Arch. Baseline. Annotation to be explained later.
		§2.2	MI-C6: GIS data processing (1) GIS data processing (geo-features) explained
		§2.3	MI-C7: Data mining functions Explained the reason for stored procedures.
		§2.3 Figure 1	MI-C8: KDD positioning Updated text/figure to show that KDD is writing to DM-DB.
		§2.3 §7.2	MI-C9: Operator role and privileges (1) Roles explained; privileges subject to design document.



## TABLE OF CONTENTS

<b>1. Introduction .....</b>	<b>6</b>
1.1 Purpose.....	6
1.2 EOLib .....	6
1.3 Document Structure .....	6
<b>2. EOLib Architecture Overview .....</b>	<b>7</b>
2.1 EOLib Baseline Architecture .....	7
2.2 Payload Ground Segment Components Overview .....	8
2.3 EOLib System Components Overview.....	8
2.4 Interfaces Overview .....	9
2.5 System Actors Overview .....	9
<b>3. Payload Ground Segment Components .....</b>	<b>10</b>
3.1 Processing.....	10
3.2 <i>Operating Tool</i> .....	10
3.3 Long Term Archive.....	10
3.4 Online User Services .....	11
3.5 DIMS Order Handling.....	11
3.6 Delivery .....	11
3.7 EOLib Interface .....	11
<b>4. EOLib Core Components.....</b>	<b>12</b>
4.1 Data Model Generation .....	13
4.2 Data Mining Database.....	14
4.3 Query Engine .....	14
4.4 Content-Based Image Retrieval .....	15
4.5 Visual Data Mining .....	15
4.6 Knowledge Discovery in Databases .....	15
4.7 Epitome Generation .....	16
4.8 System Evaluation .....	16
<b>5. Data Structures.....</b>	<b>17</b>
5.1 Earth Observation Products.....	17
5.2 Tiles.....	17
5.3 High-Resolution Quick-Looks.....	17
5.4 Basic Features .....	18
5.5 GIS Data.....	18
5.6 Annotations .....	18
5.7 Semantic Catalogue.....	18
5.8 Epitome .....	18
<b>6. Interfaces .....</b>	<b>19</b>
6.1 User interfaces .....	19
6.2 System Interfaces .....	19
<b>7. End-to-End Functionality.....</b>	<b>22</b>
7.1 Internal Use Cases.....	22
7.2 User Roles.....	22
7.3 Use Cases with Human Interaction.....	22
<b>Appendix A Abbreviations .....</b>	<b>26</b>
<b>Appendix B Definitions.....</b>	<b>27</b>



**Appendix C   References ..... 28**

**LIST OF FIGURES**

Figure 1 EOLib Baseline Architecture .....7  
Figure 2 EOLib Core Components .....12  
Figure 3 EOLib Core Component Functionality and Dependencies .....13  
Figure 4 Processing Workflow .....23  
Figure 5 Epitome Generation Workflow.....24

## 1. Introduction

### 1.1 Purpose

EOLib is an *Image Information Mining system* for *Earth Observation* (EO) data. It processes, extracts and accesses the content of EO products. It stores higher-level abstractions of these products as semantic annotations. EOLib offers information mining services on the original corpus of EO products based on the EO content, associated metadata, semantic annotations and information from external GIS sources.

EOLib is integrated with the *Payload Ground Segment* (PGS) existing at DLR's premises at Oberpfaffenhofen. EOLib offers data model generation, visual data mining and knowledge discovery in databases (including semantic annotation), queries, and *Epitome* production functionalities.

The purpose of this document is to present the architecture concept of the EOLib system. This is a live, evolving document in an iterative incremental design process, and it constitutes the baseline of the final *EOLib System Architectural Design* document.

The scope of this document is to introduce the EOLib modular and scalable architecture in order to allow further detailed design and implementation steps and also to ease maintenance and version based evolution of the system.

### 1.2 EOLib

EOLib aims at *long-term data exploitation* by enhancing the PGS *long-term data preservation* architecture with new *Data Mining* (DM) and *Knowledge Discovery in Database* (KDD) elements. Its architecture allows its evolution, migration and scalability, as well as supports its integration in distinct environments like the TerraSAR-X Payload Ground Segment or for use with the Sentinels' PGS. It is also the infrastructure for further research and development of DM and KDD concepts, algorithms and systems.

EOLib fulfils the following high-level project goals:

- operates in real time (relative to the operator/user reaction and also relative to the EO data acquisition rate)
- operates on very large data volumes,
- extends today's functions of the PGS with new interactive operation concepts, including advanced signal processing, statistical and machine learning, pattern recognition, etc. implemented in Data Mining and Knowledge Discovery components,
- integrates the novel components in the PGS architecture.

### 1.3 Document Structure

This document is organized as follows:

- [Chapter 2](#) presents an overview of EOLib architecture. The architecture overview develops the base line of the global architecture. It presents the main components of the Payload Ground Segment, the new EOLib components, their main interfaces, integration points, and also the principal data and information items which are transmitted or processed in the system.
- [Chapter 3](#) lists the Payload Ground Segment components affected by EOLib.
- [Chapter 4](#) details the EOLib core components. EOLib system components are presented using the static model view and the dependencies and decomposition view.
- [Chapter 5](#) describes the used data structures.
- [Chapter 6](#) describes the interfaces between current PGS and new EOLib components.
- [Chapter 7](#) presents the dynamic behaviour of the system.
- Appendix A** lists the abbreviations used in this document
- Appendix B** defines common terms used in EOLib.
- Appendix C** lists document and citation references.

## 2. EOLib Architecture Overview

We present in this chapter a complete overview of the EOLib system architecture. This document will be followed by the more detailed *EOLib Architecture Document*. The design of the individual components and interfaces will be described in the *EOLib Design Document*.

For an overview of the EOLib functionality, please see the *EOLib Vision* [1], *EOLib Scenarios* [2], the *EOLib Requirements Specification* [3] and the *EOLib Algorithms Specification* [4].

### 2.1 EOLib Baseline Architecture

EOLib consists of several independent systems that communicate with each other via established interfaces in a service-oriented architecture. Each of these systems may integrate one or several components which provide specific functionality to the system. We use in this document the terms system and component interchangeable (stricter definitions to be expected in the *EOLib Architecture Document*). New for EOLib are several query, DM and KDD components and the necessary interfaces and extensions of the PGS.

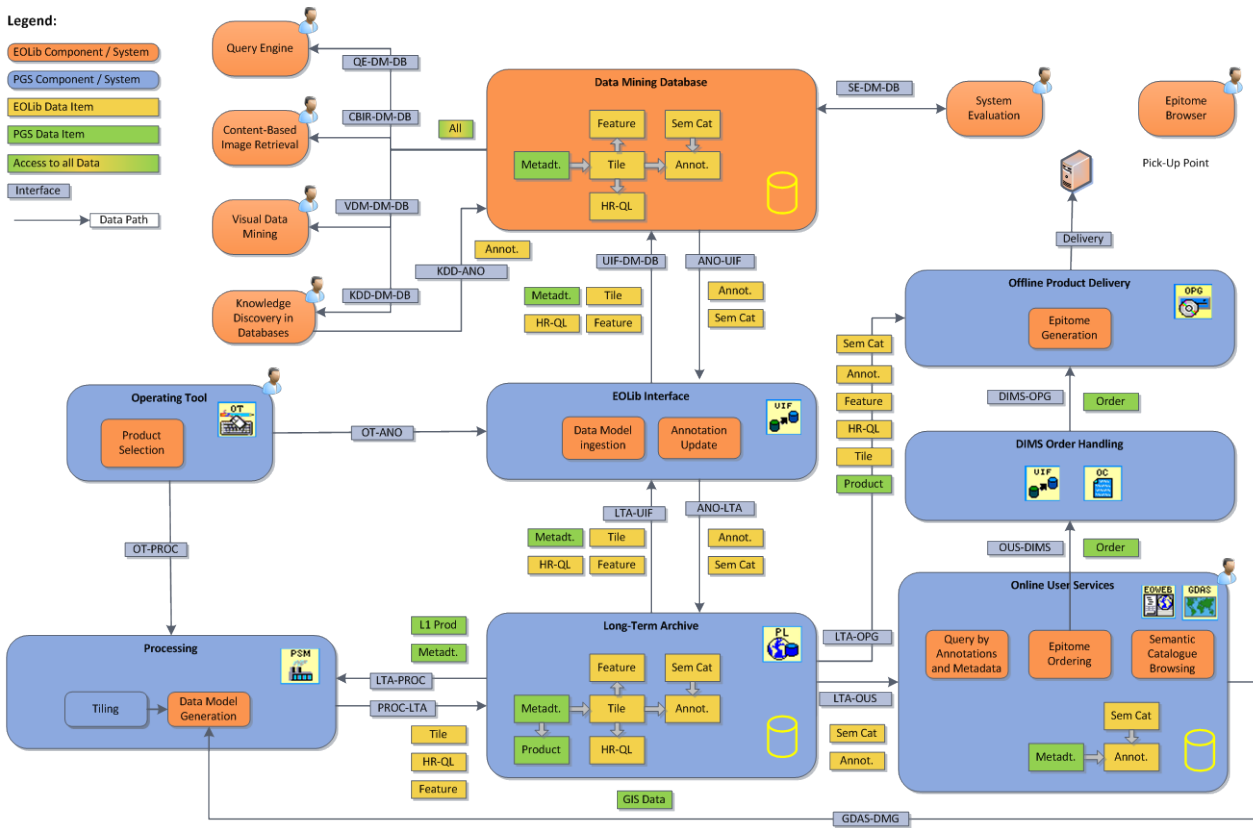


Figure 1 EOLib Baseline Architecture

The *Baseline Architecture* presented in Figure 1 shows the EOLib components and their interactions. Systems/components are shown as rectangles with rounded corners, blue for current PGS and orange for new EOLib. Interfaces are displayed as arrows pointing to the direction of the information flow. Control flow is not always shown in this diagram. Information items transmitted via an interface are displayed near the interface arrow as rectangles (yellow for EOLib, green for PGS). Components with user interfaces have an icon *user* shown in the upper right corner. Primary-foreign key relations between data items are shown as wider grey arrows.

The rest of this chapter briefly describes the Payload Ground Segment (blue domain) and the EOLib segment (orange domain), and presents an overview of the interfaces used within the system.

## 2.2 Payload Ground Segment Components Overview

DLR's payload ground segment consists of several subsystems such as data acquisition, data processing, data and information management, librarian and archiving. EOLib is designed to integrate mainly with the payload ground segment's subsystem *Data and Information Management System* (DIMS). PGS is shown in the *Baseline Architecture* in blue (components) and green (data items).

The product ingestion chain will be logically augmented with a PSM-based *Processing* component containing the following functionalities:

- *Tiling* splits EO data into tiles following projection, grid placement/size and resolution requirements.
- Generation of a *High-Resolution Quick-Look* for each tile.
- *Data Model Generation* computes for each tile several *features*.

Input comes from LTA (EO products) or GDAS (GIS data). The *Processing* results are stored in the Long-Term Archive (LTA). *Processing* is controlled through the *Operating Tool* (OT).

The *Long-Term Archive* provides persistent storage and access to EOLib Data Items. Data from the processing workflow will be uploaded over the *EOLib Interface* to the *Data Mining Database* and EOLib Data Items will be downloaded and stored in the Long-Term Archive together with the L1b products. The LTA is based on DIMS' Product Library (PL). LTA is the master repository for EOLib. An additional high-speed database (DM-DB) is required for performance reasons. It shall be possible to rebuild the DM-DB from data in LTA. Moreover, a mechanism to ensure the consistency of LTA against DM-DB data will be implemented.

*The Online User Services, based on PGS librarians Earth Observation on the Web (EOWEB) and Geospatial Data Access System (GDAS), will be enhanced by additional EOLib components for browsing a specific part of the Data Mining Database, the semantic catalogue, for queries by metadata and annotations and for ordering Epitomes (see Figure 1).*

The *Delivery* (based on PGS *Online-Product Generation*) needs to be enhanced in order to support the delivery of EOLib Epitomes [5]. *Epitome Delivery* is commanded through the *Online User Services*.

The *EOLib interface* (based on PGS User Interface Framework UIF) provides the connection between the PGS long-term archive (LTA) and the EOLib database (DM-DB).

PGS components are presented in Chapter 3.

## 2.3 EOLib System Components Overview

The EOLib system components contain most of the novel and innovative EOLib functionality. They are shown in the *Baseline Architecture* in orange (components) and yellow (data items). EOLib system modules are either independent components or functionality integrated into existing PGS components. They are presented in detail in Chapter 4.

The *Data Mining Database* provides high-speed storage and some data mining functionality whose performance (both processing and retrieval) requires a database-close implementation.

The following components provide front-end functionality to the user and operator:

- *Query Engine (QE)*: queries based on metadata and semantic labels, and image time series.
- *Content-Based Image Retrieval (CBIR)*: query by example.
- *Visual Data Mining (VDM)*: structured browsing facility for large amounts of image data,
- *Knowledge Discovery in Databases (KDD)*: adds semantic annotation to EO products,
- *System Evaluation (SE)*: supports the evaluation of the main processing units of EOLib, the associated interfaces and the GUI for operation and quality monitoring of the system.



The Query Engine, Content-based Image Retrieval and Visual Data Mining components read the information from the databases, while the System Evaluation and Knowledge Discovery in Databases components read and write information from/to the database.

The *Epitome Generation* is integrated with the PGS delivery system.

*Data Model Generation* adds EOLib specific information to the EO products processed during Ingestion and Reprocessing.

*Data Model Ingestion* and *Annotation Update* allow UIF to route data from Product Library (PL) to DM-DB and vice versa.

## 2.4 Interfaces Overview

The EOLib architecture aims at providing unambiguous, structured interfaces whose semantics do not necessarily depend on a specific underlying technology. All interfaces have been identified in Figure 1 by grey boxes. The following types of interfaces have been identified: user and internal interfaces. Although DIMS has external interfaces, they are not specific and directly relevant to EOLib.

*User interfaces* allow for the direct operator or user interaction with EOLib. They are shown in the diagram as a human symbol near the affected component. These will be presented in detail in Chapter 6.

*Internal interfaces* are to connect the systems and their system components to each other. For example, interface LTA-PROC interfaces LTA and the Processing facility. Details on all internal interfaces are described in chapter 6.

## 2.5 System Actors Overview

EOLib use cases are initiated by several actors, human and automatic [2]. Human-machine interfaces are shown in Figure 1 using a person-icon. Human actors are the end-user, the information mining operator, the system operator and the system engineer.

Not described in the Scenarios document [2] are the non-human actors, including the ingestion process for EO data and several time-based triggers. These will be specified in the EOLib design document.

### 3. Payload Ground Segment Components

EOLib is designed to be integrated with parts of the Payload Ground Segment, specifically with its Data and Information Management System (DIMS), with processing control functions (PSM) and processors (TMSP), and an interface with Geospatial Data Access System (GDAS, part of the DLR user services). Partially, they will be used as is, partially they will be enhanced.

In the following, we explain the current components of PGS.

#### 3.1 Processing

The *Processing* component provides the EOLib-specific processing functionality (tiling, basic feature extraction etc.) for EO products. A set of L1B products is selected in the *Operating Tool (OT)* and communicated to the *Processing* component through the OT-PROC interface. Each product is fetched from *LTA*, tiled and the tiles subjected to *Data Model Generation*. The L1B product together with the *DMG* output (*Tiles, High-Resolution Quick-Looks, Basic Features*) is stored back in the *LTA*.

It is assumed that L1B products are already present in the *LTA*. If not, processing shall be conducted in advance (not part of EOLib).

#### 3.2 Operating Tool

The DIMS *Operating Tool (OT)* “is designed to centrally provide a harmonized control of the distributed subsystems within DIMS” [6]. The EOLib operator uses the OT interface to select a set of products for bulk processing and annotation for EOLib. Other control flows towards DIMS components as performed with the OT, which are not specific or directly relevant to EOLib are not described in this document.

#### 3.3 Long Term Archive

The *Long-Term Archive (LTA)* [7] provides long-term persistent storage and access to EOLib Data Items. Additionally to the existing EO product and metadata storage, tiles, high-resolution quick-looks, basic features, annotation data and a semantic catalogue shall be stored. The *LTA* is the master repository of EOLib data<sup>1</sup>.

The *LTA* receives EOLib input from the *Processing* component and outputs data to the *EOLib Interface* (further linked to the *DM-DB*) and to the *Epitome Generation*. Some other interfaces may be added in the design phase.

The DIMS Long-Term Archive [8] provides the following persistence functionalities for EO products:

- Archival; organization in collections,
- Product/metadata queries and retrieval,
- Version control,
- Event triggering,
- Replication,
- Access control,
- Data structure management.

The Long-Term Archive is implemented by the PL DIMS component.

*LTA* already stores EO products and metadata that will be used by EOLib. EOLib extends the EO product data model in PL with several additional data items, including tiles, high-resolution quick-looks, features, annotations and a semantic catalogue.

---

<sup>1</sup> Please see [3]:

- *REQ-129 Long term data preservation*: The system’s long-term archive shall comply with the consistency requirements of long-term data preservation.

### 3.4 Online User Services

The *Online User Services* (EOWEB [9] and GDAS) will be enhanced by additional EOLib components for:

- browsing a specific part of the Data Mining Database, the *semantic catalogue*;
- performing queries based on metadata and *semantic annotations*;
- ordering *Epitomes*.

The semantic catalogue data is uploaded to the local EOWEB/GDAS catalogue from the DIMS LTA. GDAS is the source of GIS data for EOLib.

### 3.5 DIMS Order Handling

*DIMS Order Handling* represents a set of components for the control/data path between the *Online User Services* and the *Offline Product Delivery* components. Epitome delivery orders are routed through this part of the system (containing at least a DIMS UIF and an OC component).

### 3.6 Delivery

The delivery function is based on the DIMS service *Online-Product Generation* (OPG) [10] and needs to be enhanced in order to support the delivery of EOLib Epitomes.

An epitome is an EO product delivery augmented with EOLib data items as follows:

- EO product and metadata (as in a normal product delivery),
- Tile parameters and high-resolution quick looks,
- Basic features,
- Annotations

The Epitome data is read from LTA. A tool for browsing and classification (*Epitome browser*) can be downloaded separately from a special Web site.

### 3.7 EOLib Interface

This component is based on DIMS' UIF and provides the connection between the new EOLib components and the current PGS components. It routes the products obtained from PL to Data Mining Database and vice versa.

## 4. EOLib Core Components

In this chapter, we describe in detail the novel EOLib system components. An overview of the novel EOLib components is given in Figure 2 (*Core Components*) while a more detailed view including functionalities is shown in Figure 3 (*Core Component Functionality and Dependencies*).

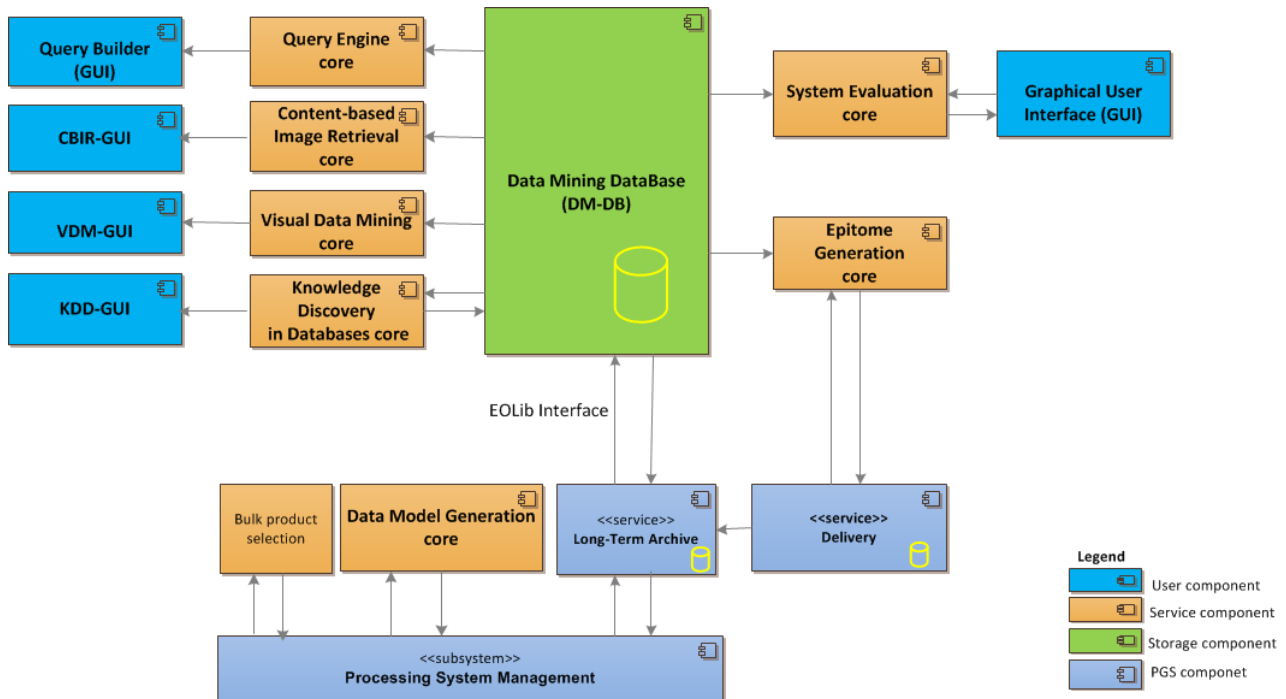


Figure 2 EOLib Core Components

Figure 2 (*Core Components*) presents the EOLib system configuration with a selection of core components and interfaces. In contrast to Figure 1 (*Baseline Architecture*), some PGS components are not included. Moreover, some components were elaborated to a finer detail level (e.g., *Query Builder* in *Core* and *GUI*). Rectangles stand for components while arrows represent core interfaces and point in the direction of data flows.

We see in the *Core Components* diagram that *Data Model Generation* (DMG) is controlled by the *Processing System Management* (PSM). All EOLib components rely on the *Data Mining DataBase* (DM-DB). The data model generation is connected to DM-DB through the Long-Term Archive and the EOLib interface. EOLib interface implements data model ingestion and annotation update functions. The components oriented to the users or operators are composed of the graphical interface and the core. For example, the *Query Engine* is composed of the *Query Builder* and *Query Engine Core*. The Epitome is generated in the delivery service.

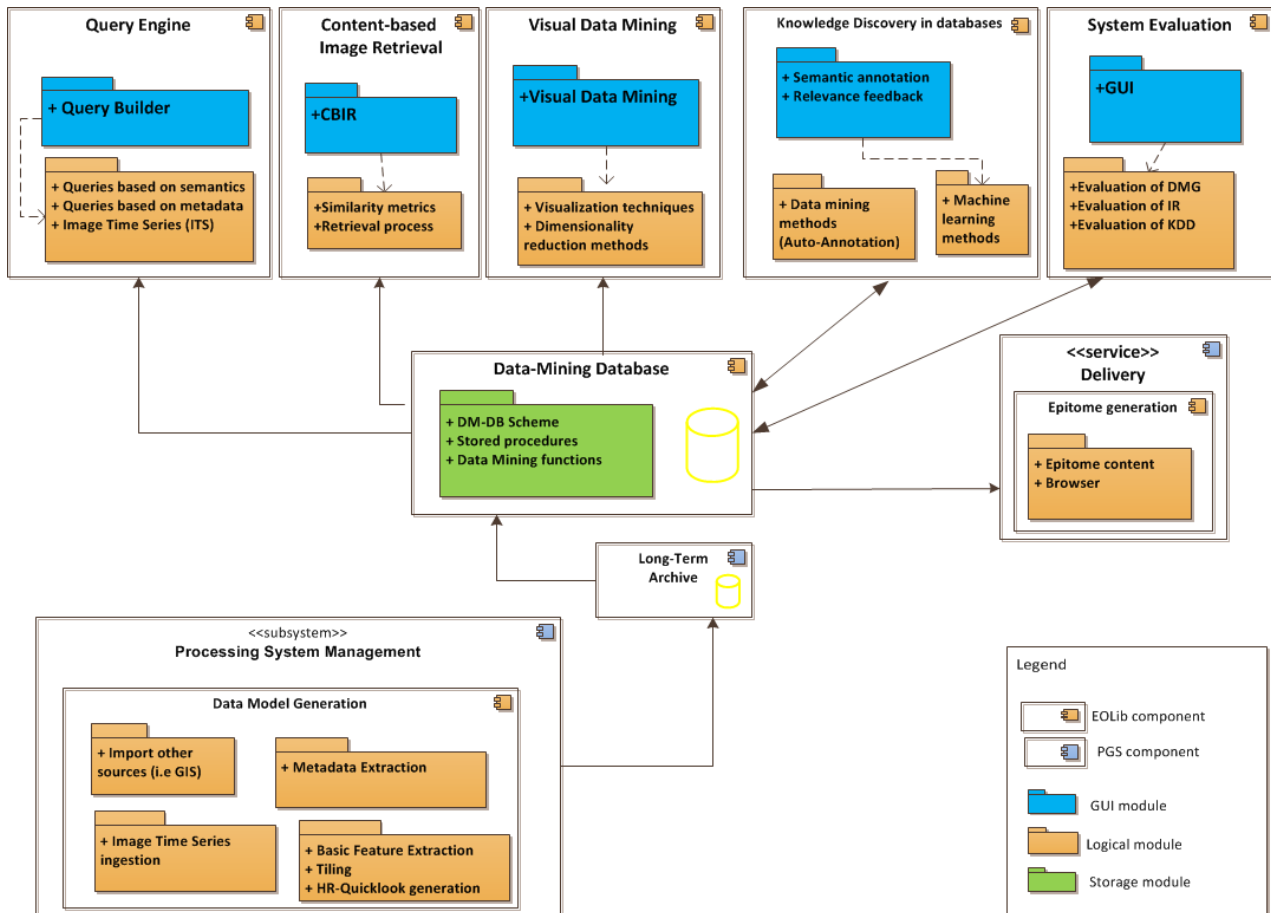


Figure 3 EOLib Core Component Functionality and Dependencies

Figure 3 (Core Component Functionality and Dependencies) shows the functional decomposition of the EOLib core components together with their dependencies. Components are shown as rectangles, functional sub-components as folders. Functions relate to algorithms described in [4]. Dashed arrows represent *used-by* relationships. Full arrows mean as in Figure 2 interfaces and point to the direction of the data flow.

In the following, we describe the core components of EOLib and their sub-components.

#### 4.1 Data Model Generation

The Data Model Generation (DMG) component (see Figure 1 and Figure 2) consists of

- The data model generation core, which processes the EO products and stores it in the Long-Term Archive.

The Data Model Generation component is a processing chain that produces EOLib data items from an EO product and its metadata. Figure 3 depicts the most important operations:

- metadata extraction,
- basic feature extraction (BFE),
- tiling the image content at multiple resolutions,
- high-resolution quick-look (HR-QL) generation,
- ingestion of image time-series (ITS) ,
- import other sources as for example GIS information.

The output, consisting of: metadata, tiles, high-resolution quick-looks and extracted features is saved into the Long-Term Archive. The data is later transferred to the DM-DB via the EOLib interface.

The main output of the DMG consists of the tiles, metadata, basic features and HR-QLs called EO data items.

## 4.2 Data Mining Database

The Data-Mining Database (DM-DB) component manages data handling, storage, administration and some of the processing for the entire EOLib components [11]. It is a new development for EOLib and consists of:

- Data
- Database schema
- Stored procedures (supporting data mining operations e.g. clustering)

The following information is stored: metadata, image parameters, tile parameters, feature vectors, semantic labels. In addition, some data mining functions are created inside the database scheme.

The baseline architecture (see Figure 1) together with the static model (see Figure 2) illustrates the interaction with the other components:

- DMG provides the input data to be stored into the Data Mining Database and the Long-Term Archive.
- The EOLib components (Query Engine, Content-Based Image Retrieval, Visual Data Mining, Knowledge Discovery in Databases, and System Evaluation) access the information and update certain aspects (e.g., annotations).

## 4.3 Query Engine

The Query Engine (QE) component (see Figure 1 and Figure 2) consists of

- A GUI called Query Builder which interacts with the user and receives the user's queries and
- The Core (Query Engine Core) which accepts the queries from the GUI and uses them to get the data from the DM-DB.

The Query Engine allows the user to search for EOLib/PGS data including Terra-SAR-X EO products. It is a new development for EOLib. The following types of queries are supported:

- Query based on metadata (QbM). The user can query the image archive using standard metadata as for example coordinates system, type or product, acquisition time, etc. (attributes stored in metadata table).
- Query based on semantic annotations (QbS). The user can enter a simple label in the form a text or select from the available labels in the semantic catalogue to perform the query. This kind of query can be improved by using ontologies. It is worth to mention that those labels are pre-defined labels previously obtained as results of the semantic annotation.
- Query by multi-temporal parameters for image time series (ITS). Query by multi-tempo parameters involves Image Time Series components and analysis.

The query reads part of the data model stored into the data mining database (DM-DB) and the output is a list of data items like patches or images. Results are sorted by a relevance metric<sup>1</sup> (e.g., compression-based distance).

---

<sup>1</sup> The *relevance metric* will be defined in the detailed system design.

#### 4.4 Content-Based Image Retrieval

The Content-based Image Retrieval (CBIR) component (see Figure 1 and Figure 2) consists of

- A GUI which interacts with the user and receives the image patch selected by the user and
- The Core (Query Engine Core) which accepts the queries from the GUI and uses them to get the data from the DM-DB.

The Content-based Image Retrieval is based on the concept of query by example (QbE), where the user is able to present to the system a query image, and retrieve images which are similar in content. The semantic criteria are not explicitly formulated by the user but chosen through examples.

The input of CBIR is part of the data model and the output is a list of patches ordered according their similarity. Figure 3 describes the functionality of CBIR:

- similarity measurement<sup>1</sup> and
- retrieval process.

As result of this component is a collection of visual quick-looks displayed in a ranked order on the screen.

#### 4.5 Visual Data Mining

The Visual Data Mining (VDM) component (see Figure 1 and Figure 2) consists of

- A GUI which allows the user to navigate in the image archive and
- The Core (VDM Core) which gets the data from the DM-DB and adapts them in order to display to the user through the GUI.

The Visual Data Mining provides visualization techniques for very large, highly complex image sets (in our case, tile HR-QLs). These are shown in 3D arrangements that allow for easy interactive exploration and analysis. The technique can also be applied to non-visual data sets.

The analyst can navigate in a multi-modal 3D arrangement and interactively search and refine relevance criteria to explore patterns of interest in the data.

As it is depicted in Figure 3, Visual Data Mining consists of the following functionality:

- visualization techniques and
- dimensionality reduction methods.

As result of this component is a collection of visual quick-looks displayed in a 3D arrangement on the screen.

#### 4.6 Knowledge Discovery in Databases

The Knowledge Discovery in Databases (KDD) component (see Figure 1 and Figure 2) consists of

- A GUI which interacts with the user and receives the user's input and
- The Core (KDD Core) which accepts the user's requirements from the GUI and uses them to get the data from the DM-DB.

The Knowledge Discovery in Databases involves several processes of discovering useful information (*knowledge*) from a set of data while relying on data mining methods.

The EOLib KDD module accesses the data stored in the data mining database and uses some the available stored procedures.

Figure 3 shows the functionality provided by the EOLib KDD module:

- The graphical interface allows the user an interactive annotation of the image content. It also provides relevance feedback of the results.

---

<sup>1</sup> The *similarity measures* will be defined in the detailed system design.

- The *semantic annotation* is based on machine learning methods, which take the operator feedback as training data and provides the prediction of the results. After training, the whole data corpus can be annotated (*automated annotation*).

#### 4.7 Epitome Generation

The epitome is generated in the Delivery PGS component and it is composed of metadata, high-resolution quick-looks, basic features and all the annotations. The epitome generation component reads the information from the Data Mining Database and LTA and transfers it to delivery service.

#### 4.8 System Evaluation

The System Evaluation component (see Figure 1 and Figure 2) consists of

- A GUI which interacts with the operator and receives the operator's input and
- The Core (system evaluation core) which accepts the inputs from the GUI and uses them to get and to process the data from the DM-DB.

The System Evaluation component collects information and statistical data about the activity of the EOLib components. Quality metrics and the evaluation procedure are defined according to the component to be evaluated. The system evaluation requires pre-defined test data sets.

This component is also in charge of the quality evaluation of individual EOLib components as well as the validation of the complete system.

Figure 3 shows the functionality of the system evaluation component.

- Evaluation of data model generation.
- Evaluation of Image Retrieval which involves the query engine and CBIR evaluations. The precision of IR results shall be verified using quantitative metrics as for example precision and recall, false alarms, etc.
- Evaluation of Knowledge Discovery in Databases.

The complete description of the system evaluation component will be presented in Detailed Design Technical Note.



## 5. Data Structures

This chapter presents a conceptual overview of the PGS data structures enhanced with EOLib data. The EOLib data items (with the exception of the semantic catalogue) relate to a given product and are attached to every EO product that is relevant for EOLib. Most PGS/EOLib data items are stored persistently both in the LTA and the DM-DB and transferred over the various system interfaces. For clarity, please recall the definitions of the data structure and data model [11].

Let us start by looking at the LTA as a list of EO product metadata and associated items. Each metadata set is related to an EO product and a set of tiles (part of the product). Each tile has an associated HR-QL, a set of features (one feature vector for each extraction algorithm) and another set of annotations. Annotations are chosen from the semantic catalogue.

### 5.1 Earth Observation Products

EO products (TSX and TDX products for the time being) are stored in the LTA together with the corresponding metadata. The data structure is extended for EOLib to contain tiles, HR-QLs, basic features and annotations, discussed below.

Products are delivered as an accepted directory and file structure [12]. This is extended for EOLib to contain the additional data that makes up the *epitome* of a specific EO product.

### 5.2 Tiles

The EO image is divided into several tiles. A *tile* is a part of the image, which can contain one or several objects depending on its size.

The tile size and geometry is addressed in the Algorithms Technical Note [4]. A tile set is aligned with one world-wide grid<sup>1</sup>. Tiles are square, adjacent, not overlapping and have the same size in pixels. Tiles have similar geographical sizes. A second set of tiles is cut in such a way that it overlaps with the borders of the first. Additional tile sets are created in different resolutions for resolution pyramids.

A candidate grid for the tiling algorithm is defined by the UTM standard.

A synonym for tiles is patches. We prefer the word tiles in the EOLib context. Patches stems from the IIM science, whereas tiles originates from common GIS usage.

### 5.3 High-Resolution Quick-Looks

For visualisation purposes, high-resolution quick-looks are created for each tile. A better resolution than in the normal product-level quick-looks is needed due to the smaller tile size.

---

<sup>1</sup> Please see [3]:

- *REQ-58* Projection of tiles: Tiles shall fit the same projection or projections that are close within a defined error range. Tiles shall have the same orientation.
- *REQ-86* Resolution pyramids of tiles: Image tiles should be created at different resolutions levels, thus building a resolution pyramid. The length of the tile border is invariant.
- *REQ-186* Tiling: Images shall be cut in square tiles. Tile size shall be configurable/parametric. It shall depend on resolution and is if appropriate adapted for multi-sensor and multi-scale data sets. Ground projection of tiles shall be such that they are superimposable within a close resolution range (up to factor 4).
- *REQ-151* GIS text data: GIS text data shall be reprojected and split according to the tiling scheme. Text data shall be used for analysis and optionally for gazetteering of tiles.

## 5.4 Basic Features

The basic features are relevant information for characterizing an image as for example mean and variance values, texture, shape, etc. The features are known as descriptors of the image content and their information depend on the method. Please see [4] for more details.

## 5.5 GIS Data

The GIS data (geographic features) at the Online User Interface is used for overlay purposes by the GDAS component. However, within the data model generation it could enhance the available information for searches and annotation. Available GIS data could enrich the Data Mining Database.

## 5.6 Annotations

The content of the tile is described with semantic annotations e.g., forest or agriculture.

## 5.7 Semantic Catalogue

The semantic catalogue is taxonomy of semantic labels together with the ontological relations between them. For simplicity, EOLib will use a taxonomy in a tree representation with a few levels.

## 5.8 Epitome

The Epitome is a summary of the EO product information content as presented as actionable information for information mining in individual EO products. The epitome is a result of the BFE and semantic annotation and may be delivered with the standard EO product or as a distinct product component. It is intended to be used for the individual product content inspection. Epitome access is done offline, on the user's PC/notebook, using the Epitome Browser (a separate program).

The distribution of the epitome is organized in three main folder containing:

- Standard L1b product: It is the standard TerraSAR-X L1b product.
- Features: The Epitome is composed of metadata, high-resolution quick-looks, basic features and the semantic annotations per patch. Basic features are included for the purpose of allowing the users to run their own classification algorithms.
- Tool: It contains a *Browser* which is *client* software with the following functionality
  1. EO product (image) visualization, including GIS layer,
  2. Browsing metadata and semantic labels,
  3. Interactive learning/classification (operating on tiles).

A complete description of the Epitome together with its content is presented in detail in [5].

## 6. Interfaces

An overview of the EOLib interfaces is given in Figure 1, *Baseline Architecture* (interfaces shown in grey). We classify the interfaces as external, user and internal interfaces. External interfaces connect EOLib to systems outside it e.g., an EO satellite acquisition chain. Internal interfaces connect EOLib and PGS components with each other. User interfaces allow the user or the operator to interact with EOLib.

This chapter describes the most important interfaces in EOLib at the conceptual level.

### 6.1 User interfaces

#### Query Builder

This interface will be described in detail in a later project phase.

#### Content-Based Image Retrieval

This interface will be described in detail in a later project phase.

#### Visual Data Mining

This interface will be described in detail in a later project phase.

#### Knowledge Discovery in Databases

This interface will be described in detail in a later project phase.

#### System Evaluation

This interface will be described in detail in a later project phase.

#### Online User Services

The Online User Services (EOWEB and GDAS) allow the user/operator to:

- Browse the semantic catalogue
- Place orders for epitomes

#### Operating Tool Interface

The Operating Tool (OT) allows the operator to command bulk processing of EO products

#### Delivery Interface

Epitomes are delivered to end-users either on DVDs or on an ftp pick-up point.

### 6.2 System Interfaces

*System interfaces* connect EOLib system parts with each other. They interact by command-and-control messages and by transferring data. The *Baseline Architecture* (Figure 1) shows the direction in which data items are transferred. Control flow is not shown in this diagram.

The following data items are transferred over the internal interfaces, as shown in the *Baseline Architecture*:

- *Metadata*: describing a specific product,
- *Product*: for now, an L1B TSX Earth Observation product,
- *Tile*: fixed-size (in pixels), related to a specific product,
- *High-resolution quick look*: of a tile,
- *Feature*: abstract descriptor of a tile, used in data mining, KDD and for queries,
- *Annotation*: describe the information found in a tile,
- *Semantic Catalogue*: contains the taxonomy learned and used by EOLib,

Command-and-control<sup>1</sup> interfaces and messages are not shown in the diagram.

#### **OT-PROC**

This interface allows the operator to select a range of products for bulk processing and initiate the operation in the *Processing* component.

#### **PROC-LTA**

This interface allows the *Processing* component to save processing results (*Tiles*, *High-Resolution Quick Looks* and *Features*) back into the LTA.

#### **LTA-PROC**

This interface allows the *Processing* component to access L1B products and metadata in the LTA.

#### **ANO-LTA**

The *Annotation* component uses this interface to store *Semantic Annotations* back into the LTA.

#### **GDAS-DMG**

Geo-data in PGS are managed by the GDAS component. This interface allows the data model generation access to the required geo-data.

#### **LTA-UIF**

This interface facilitates the upload of the PGS/EOLib data from the large, secure but sometimes slow DIMS LTA component to the faster and dynamic data mining database. The upload process is managed by a UIF instance sitting between the LTA and the DM-DB. UIF operates automatically without user interaction.

A mechanism to keep DM-DB synchronized with LTA will be implemented e.g., incremental uploads during bulk processing.

#### **ANO-LTA**

This interface transfers *annotations* and the *semantic catalogue* from the DM-DB back to the LTA.

#### **UIF-DM-DB**

This interface serves the same function as *LTA-UIF* (i.e., it relays data one step further). Even though the function is the same, the interface and protocol may differ.

#### **ANO-UIF**

This interface transfers *annotations* and the *semantic catalogue* from the DM-DB back to the LTA.

#### **LTA-OUS**

Online User Services receive over this interface a subset of the LTA database. For instance, product metadata and quick looks are transferred this way. The data is needed for performance reasons, similar to the DM-DB. We extend this interface to additionally transfer the *Semantic Catalogue* and product *Annotations* thus enabling functionality like *Query by Annotations and Metadata* or *Semantic Catalogue Browsing*.

#### **OUS-DIMS**

This interface transmits *Epitome* orders to the DIMS order handling.

#### **DIMS-OPG**

This interface transmits *Epitome* orders to the *Offline Product Delivery*.

#### **LTA-OPG**

The *LTA-OPG* interface provides access to the required data in LTA for the *Epitome Generation* and *delivery* component.

---

<sup>1</sup> Examples of command-and-control messages are status updates within PGS, for instance to EOWEB when a product/epitome was delivered.

---

**QE-DM-DB**

This interface will be described in detail in a later project phase.

**CBIR-DM-DB**

This interface will be described in detail in a later project phase.

**VDM-DM-DB**

This interface will be described in detail in a later project phase.

**KDD-DM-DB**

This interface will be described in detail in a later project phase.

**SE-DM-DB**

This interface will be described in detail in a later project phase.

## 7. End-to-End Functionality

This chapter describes command-and-control chains which implement the major EOLib use cases [2]. The chains span in most cases over several PGS/EOLib components. Command-and control chains use the data flow presented in Figure 1 *Baseline Architecture*.

### 7.1 Internal Use Cases

#### Upload DM-DB

Update DM-DB from LTA with metadata, tiles, HR-QLs and basic features. This workflow is driven by the *EOLib interface* and involves the *LTA* and *DM-DB*.

#### Update LTA

Update LTA from DM-DB with the semantic catalogue and annotations. This workflow is driven by the *EOLib interface* and involves the *DM-DB* and *LTA*.

#### Upload Online User Services

The semantic catalogue annotations are uploaded from LTA to EOWEB [2] [13].

#### Automated Annotation

To be described in a later project phase.

### 7.2 User Roles

#### User

The *User* [2] interacts with EOLib with the goal of finding relevant EO product data. He uses several EOLib functionalities targeted to the end-user, including the query engine, the content-based image retrieval or the visual data mining.

#### Information Mining Operator

According to the EOLib Scenarios Document [2], the Information Mining Operator “*will train the system with new data to explore and assign semantic annotations to the data or to refine them in iterative training sessions*”.

#### Operator

The System Operator is responsible for monitoring the system’s operation.

#### Administrator

The Administrator is responsible for deployment and configuration of the system.

### 7.3 Use Cases with Human Interaction

#### Processing

Product processing is triggered by an operator request for processing a batch of L1B products. This interaction is done via the *Online User Services*. The *Online User Services* then request the batch reprocessing from the *Reprocessing* component. The processing workflow is controlled by a PSM instance.

Prior processing of low-level L0 products to L1B with the desired parameters is not part of the EOLib workflow.

The following steps are executed during the processing workflow:

- Tiling
- Data Model Generation:
  - Creation of HR-QLs for each tile,

- Basic feature extraction for each tile,
- Additional GIS data (points of interest with coordinates) is read from the Online User Services and added to the metadata,
- Saving tiles, HR-QLs and basic features in LTA additionally to the *classic* product data structure.

The processed product together with the generated data model is stored in the LTA.

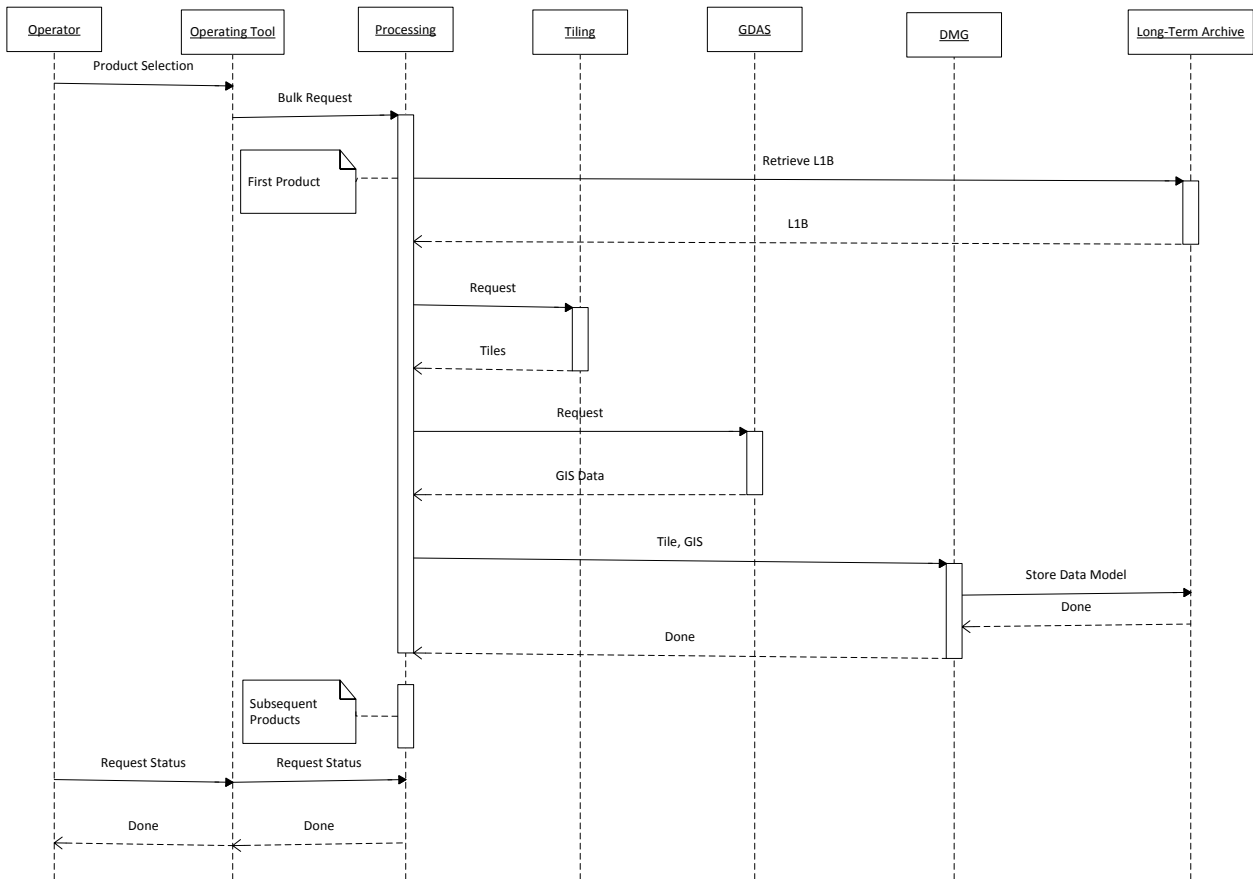


Figure 4 Processing Workflow

Figure 4 *Processing Workflow* details the steps in the reprocessing workflow.

**Order Epitome**

Users can order Epitome products using the *Online User Services*. The *Epitome Generation* is then requested from the *Delivery* component. *Delivery* retrieves the required data (parts of the Epitome) from the *LTA* and *DM-DB*. The Epitome is delivered to the user on a data medium or a pick-up point in the Internet.

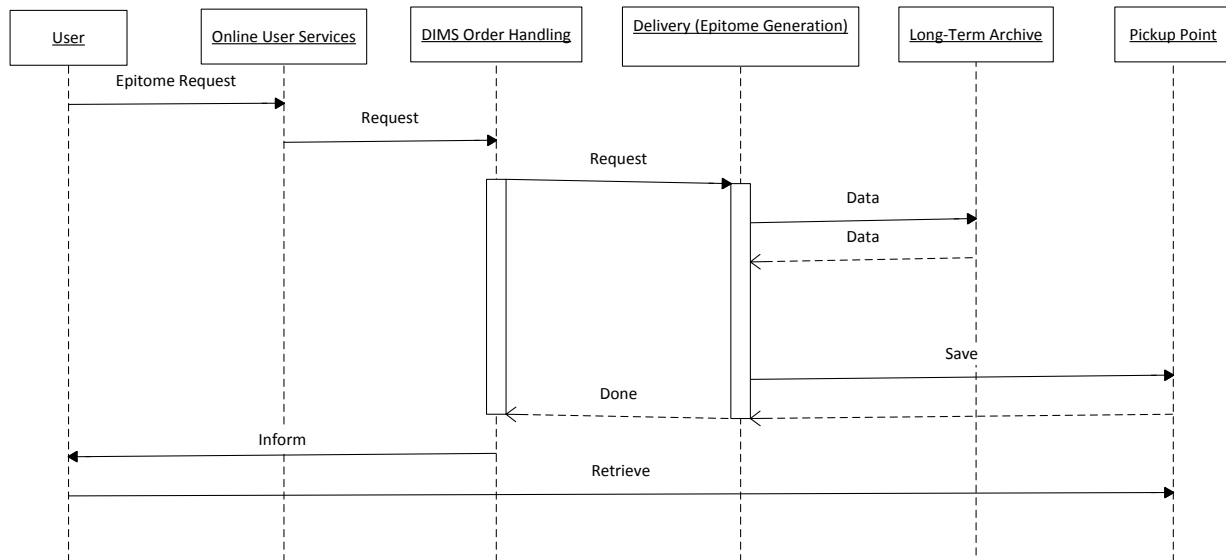


Figure 5 Epitome Generation Workflow

Figure 4 *Epitome Generation Workflow* shows the processing sequence of the Epitome.

### Browse Semantic Catalogue

The use case allows for an interactive session of the user with the system. The user is presented a visual/textual representation of the semantic catalogue.

### Search by Semantic Annotations

The use case allows for an interactive session of the user with the system. It involves the *Query Engine* and the *DM-DB* components. The implementation of this use case will be detailed in a later project phase.

### Search by Metadata

The use case allows for an interactive session of the user with the system. The implementation of this use case will be detailed in a later project phase.

### Content-Based Image Retrieval

The use case allows for an interactive session of the user with the system. It involves the *Content-Based Image Retrieval* and the *DM-DB* components. The implementation of this use case will be detailed in a later project phase.

### Visual Data Mining

The use case allows for an interactive session of the user with the system. It involves the *Visual Data Mining* and the *DM-DB* components. The implementation of this use case will be detailed in a later project phase.

### Knowledge Discovery in Databases

The use case allows for an interactive session of the user with the system. It involves the *Knowledge Discovery in Databases* and the *DM-DB* components. The implementation of this use case will be detailed in a later project phase.





### **Command System Evaluation**

The implementation of this use case will be detailed in a later project phase.

## Appendix A Abbreviations

The following abbreviations are an extract of the EOLib Glossary [14] and are here to ease the reading of the present document.

<i>DBMS</i>	Database Management System
<i>CBIR</i>	Content-Based Image Retrieval
<i>CIDL</i>	Configuration Item Data List
<i>DIMS</i>	Data Information Management System
<i>DMG</i>	Data Model Generation
<i>DM</i>	Data Mining
<i>DM-DB</i>	Data Mining Database
<i>EO</i>	Earth Observation
<i>EOLib</i>	Earth Observation Library
<i>ftp</i>	File Transfer Protocol
<i>GDAS</i>	Geospatial Data Access System
<i>GIS</i>	Geographic Information System
<i>GUI</i>	Graphical User Interface
<i>HMI</i>	Human Machine Interface
<i>HR</i>	High-Resolution
<i>IIM</i>	Image Information Mining
<i>ITS</i>	Image Time Series
<i>KDD</i>	Knowledge Discovery in Databases
<i>LTA</i>	Long-Term Archive
<i>PFE</i>	Primitive Feature Extraction
<i>PGS</i>	Payload Ground Segment
<i>PL</i>	Product Library
<i>PSM</i>	Processing System Management
<i>QL</i>	Quick-Look
<i>R&amp;D</i>	Research and Development
<i>SoW</i>	Statement of Work
<i>SRR</i>	System Requirements Review
<i>TMSP</i>	TerraSAR-X Multi-Mode SAR Processor
<i>UIF</i>	User Interface Framework

## Appendix B Definitions

*Data structure* = defines the EOLib/PGS persistent data items and the relationships between them.

*Data model* = the set of EOLib/PGS data items which describe an EO product.

*Stored procedure* = In a database management system (DBMS), a stored procedure is a set of Structured Query Language (SQL) statements with an assigned name that's stored in the database in compiled form so that it can be shared by a number of programs. The use of stored procedures can be helpful in controlling access to data (end-users may enter or change data but do not write procedures), preserving data integrity (information is entered in a consistent manner), and improving productivity (statements in a stored procedure only need to be written one time)

## Appendix C References

- [1] M. Datcu, "EOLib Vision — EOLIB-PRE-DLR-1400-3," DLR, Weßling, 2011.
- [2] M. Winkler, M. Datcu, K. Molch, "Earth Observation Image Librarian — Scenarios — EOLIB-TN-DLR-2100," DLR, Weßling, 2012.
- [3] V. Manilici, M. Datcu, M. Winkler, G. Schwarz, "Earth Observation Image Librarian — Requirements Specification — EOLIB-RBD-DLR-3200," DLR, Weßling, 2012.
- [4] D. Espinoza-Molina, "Earth Observation Image Librarian — Algorithm Specification," DLR, Weßling, 2012.
- [5] C. O. Dumitru, "Earth Observation Image Librarian — Epitome Technical Note," DLR, Weßling, 2012.
- [6] C. Reck, "DIMS Operating Tool — Service Manual," DLR, Weßling, 2012.
- [7] S. Kiemle, N. Weiland, "Product Library — Service Manual," DLR, Weßling, 2012.
- [8] E. Mikusch, S. Kiemle, M. Böttcher, M. Wolfmüller, A.-K. Schröder-Lanz, C. Reck, D. Dietrich, S. Kröger, "Data Information and Management System — Design Document — DIMS-DLR-GEN-DD-5032," DLR, Weßling, 2010.
- [9] D. Dietrich, P. Harms, M. Schmenkel, "EOWEB User Services — Design Document — DIMS-WERUM-UIS-DD-5029," DLR, Weßling, 2011.
- [10] Werum, "Online/Offline Product Generation — User Manual," Werum, Lüneburg, 2010.
- [11] D. Espinoza-Molina, "Earth Observation Image Librarian — Reference Database Design," DLR, Weßling, 2012.
- [12] M. Wolfmüller, "Payload Ground Segment — DIMS Configuration for TerraSAR-X," DLR, Weßling, 2012.
- [13] A.-K. Schröder-Lenz, S. Kiemle, "Product Library — Design Document — DIMS-DLR-PL-DD-5033," DLR, Weßling, 2004.
- [14] C. Reck, "Earth Observation Image Librarian — Glossary — EOLIB-LST-DLR-0005," DLR, Weßling, 2012.
- [15] Werum, "DIMS Processing System Management — Integrator's Manual and ICD," Werum, Lüneburg.