Shanoir NG – Import

Software Design Description

Table des matières

[1 Microservice context 1](#_Toc486500170)

[1.1 Microservice presentation 1](#_Toc486500171)

[1.2 Main functionalities 2](#_Toc486500172)

[1.3 Application users 2](#_Toc486500173)

[2 Functional architecture 2](#_Toc486500174)

[2.1 Imports 2](#_Toc486500175)

[2.1.1 Import from DICOM CD/DVD 2](#_Toc486500176)

[2.1.2 Import from PACS 2](#_Toc486500177)

[2.1.3 Import from ShanoirUploader 2](#_Toc486500178)

[2.1.4 Import processed dataset (NifTi) 3](#_Toc486500179)

[2.1.5 Import from files 3](#_Toc486500180)

[2.2 Steps 3](#_Toc486500181)

[2.2.1 Choose modality 3](#_Toc486500182)

[2.2.2 Upload files 3](#_Toc486500183)

[2.2.3 Select series 3](#_Toc486500184)

[2.2.4 Select clinical context 3](#_Toc486500185)

[2.2.5 Describe dataset 3](#_Toc486500186)

[2.2.6 Select/create data processing 3](#_Toc486500187)

[2.2.7 Anonymization 3](#_Toc486500188)

[2.2.8 Nifti conversion 4](#_Toc486500189)

[2.2.9 Select settings for research study 4](#_Toc486500190)

[3 Software/technical Architecture 4](#_Toc486500191)

[3.1 Anonymization 4](#_Toc486500192)

# Microservice context

## Microservice presentation

This microservice is a part of the Shanoir-NG application and is used to import data.

## Main functionalities

Shanoir NG main functionalities are:

* Import from DICOM CD/DVD
* Import from PACS
* Import from ShanoirUploader
* Import processed dataset (NifTi)
* Import from files

## Application users

Target population is:

* Administrators
* Experts
* Users

# Functional architecture

## Imports

### Import from DICOM CD/DVD

Steps of this import are:

* Choose modality
* Upload dataset (DICOM zip)
* Select series
* Select clinical context
* Anonymization
* Nifti conversion
* Select settings for research study

### Import from PACS

Steps of this import are:

* Choose modality
* Define DICOM query and import data
* Select clinical context
* Anonymization
* Nifti conversion
* Select settings for research study

### Import from ShanoirUploader

Steps of this import are:

* Choose modality
* Upload files
* Anonymization
* Select series
* Nifti conversion

### Import processed dataset (NifTi)

Steps of this import are:

* Upload files
* Describe dataset
* Select/create data processing
* Nifti conversion

### Import from files

Import from files allows to upload:

* Processed dataset (NifTi)
* Extra data
* Spectroscopy data

Steps to import processed dataset are:

* Upload files
* Describe dataset
* Select/create data processing
* Nifti conversion

Steps to import extra data are:

* Upload files
* Select clinical context

Steps to import spectroscopy data are:

* Upload files
* Select clinical context

## Steps

### Choose modality

### Upload files

### Select series

### Select clinical context

### Describe dataset

### Select/create data processing

### Anonymization

### Nifti conversion

### Select settings for research study

# Software/technical Architecture

## Anonymization

### Anonymized fields :

See anonymization.xlsx document

### Prerequisite

Zip files should be unzipped.

Check if zip files are unzipped during anonymization step or before.

### Steps

* Anonymization
* Send email
* Create folders

### Technical Specification of Shanoir Old Anonymization

During the import, the user have to click on the button “import Dicom Data” displayed by the xhtml page importFromSCP.xhtml. This action triggers the method processFullImportFromNetwork() of the java class ContextualImportSupport. processFullImportFromNetwork() perform many actions to prepare data import in shanoir PACS, described in figure 1.

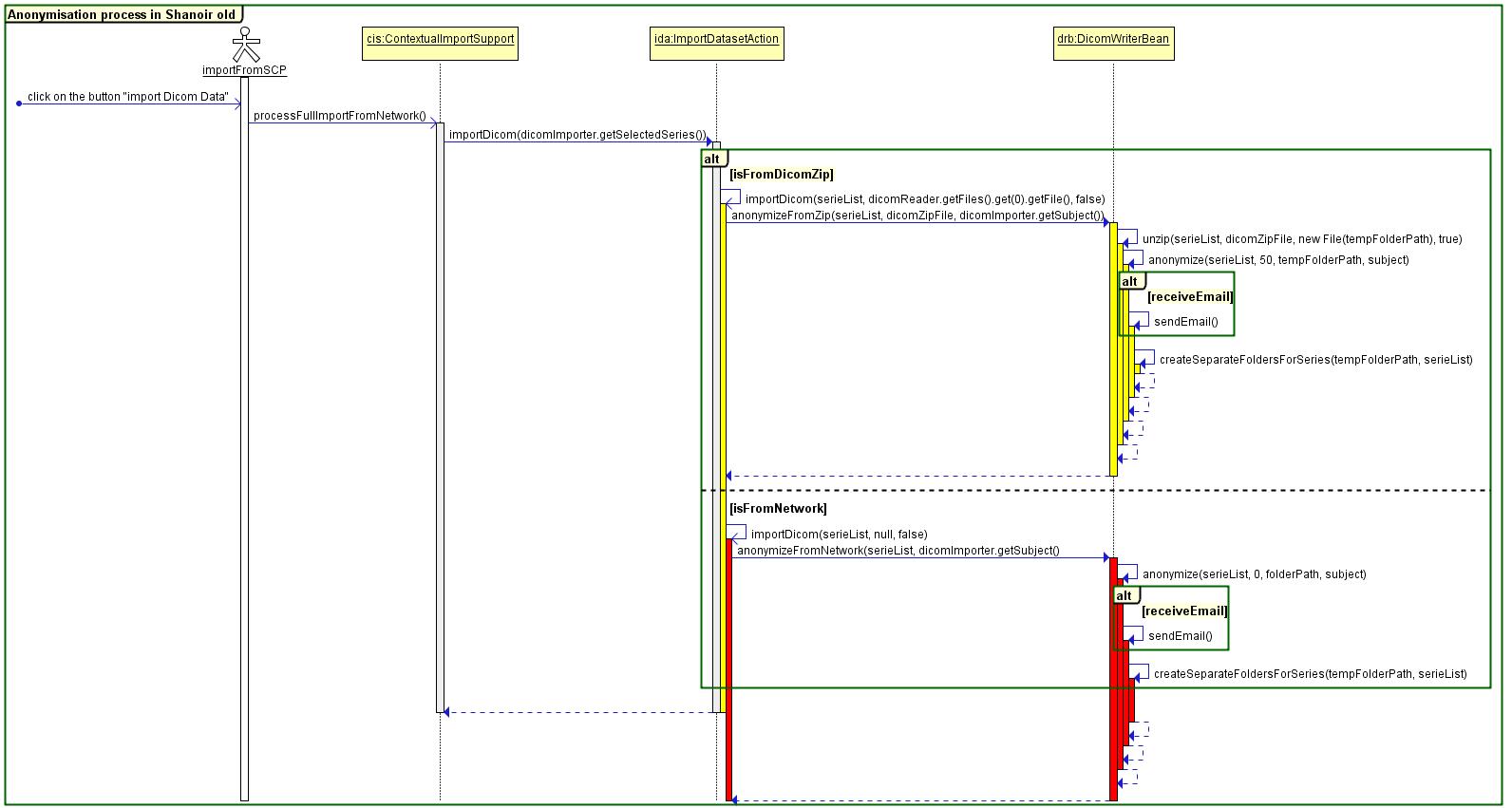


Figure 1. Anonymisation process in Shanoir old

Data anonymization is one of the actions that should be done before storing DICOMs in shanoir. The method anonymize(…) of the class DicomWriterBean.java is responsible of the anonymization process. This process is described by figure 2.

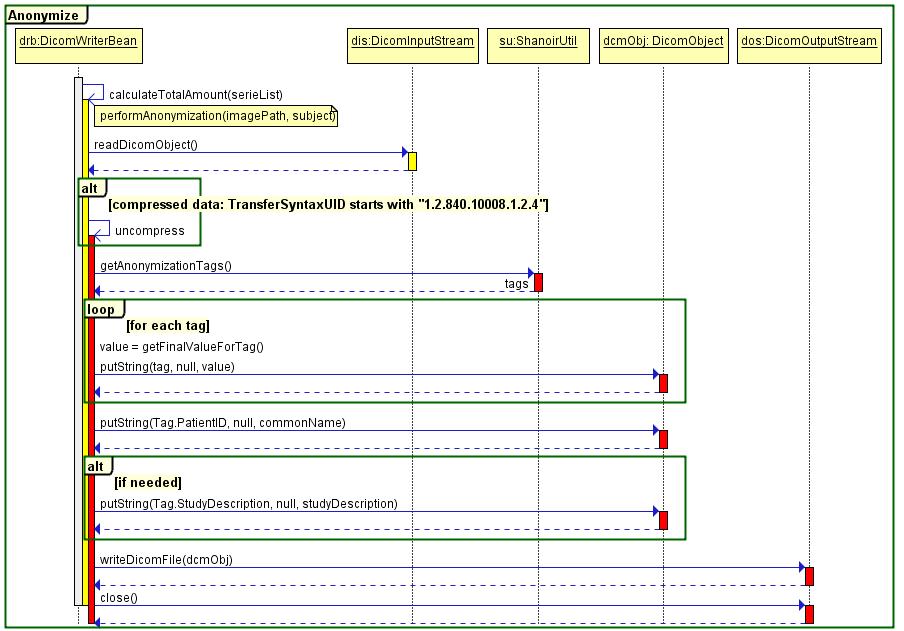


Figure 2. The anonymize method

The final value that replaces the nominative data in the DICOM tag depends on the tag anonymized. Some tags are simply deleted from the DICOM file, others are replaced by other values as described by figure 3.

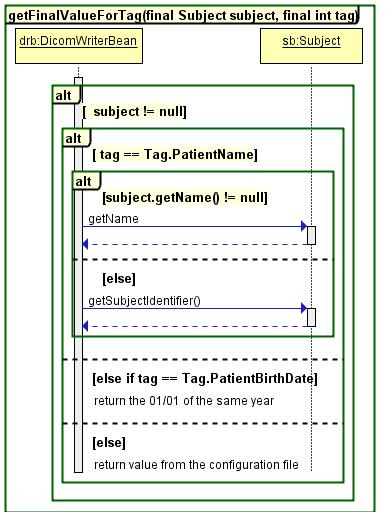


Figure3. getFinalValueForTag method: gets anonymized tag value

The list of tags to be anonymized is stored is the property file shanoir.propeties.

### Technical Specification of Shanoir NG Anonymization V1

This specification is inspired from shanoir old one. We keep the same conception design used in shanoir old and we add specific cases to be conform to Basic Profile recommendations (see figure 4).

In this case, the anonymization service reads the content of the anonymization.xlsx file, which contains the list of the tags to anonymize and the operation to do to anonymize each tag. The anonymization service stores this information in a hashpap as illustrated in figure 5. Then, for each element in the hashmap, the anonymization service anonymizes the tag based on its basic profile value (see figure 6).

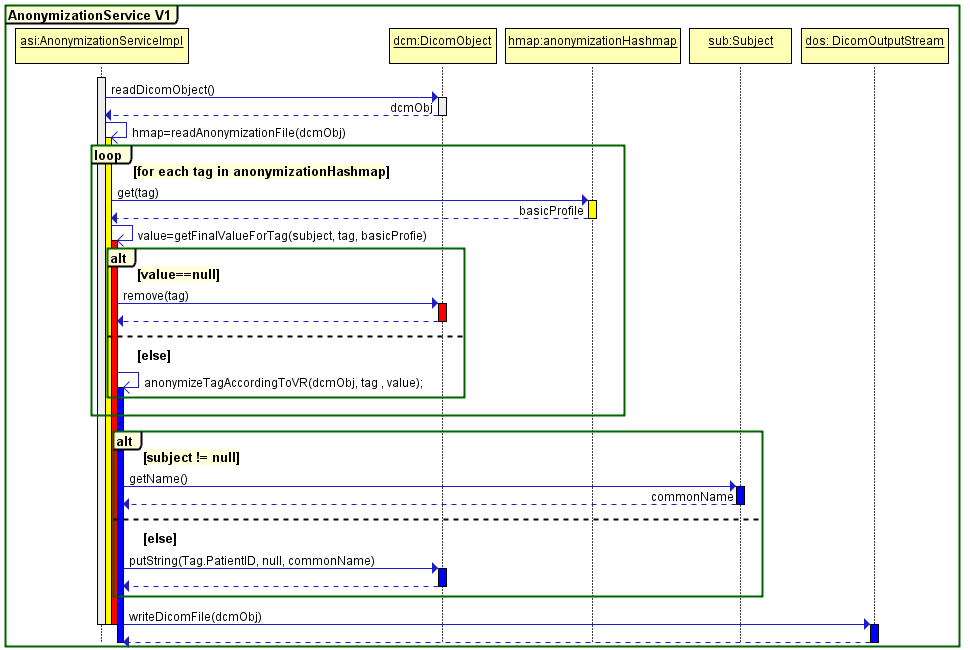


Figure 4. Anonymisation process in Shanoir NG V1

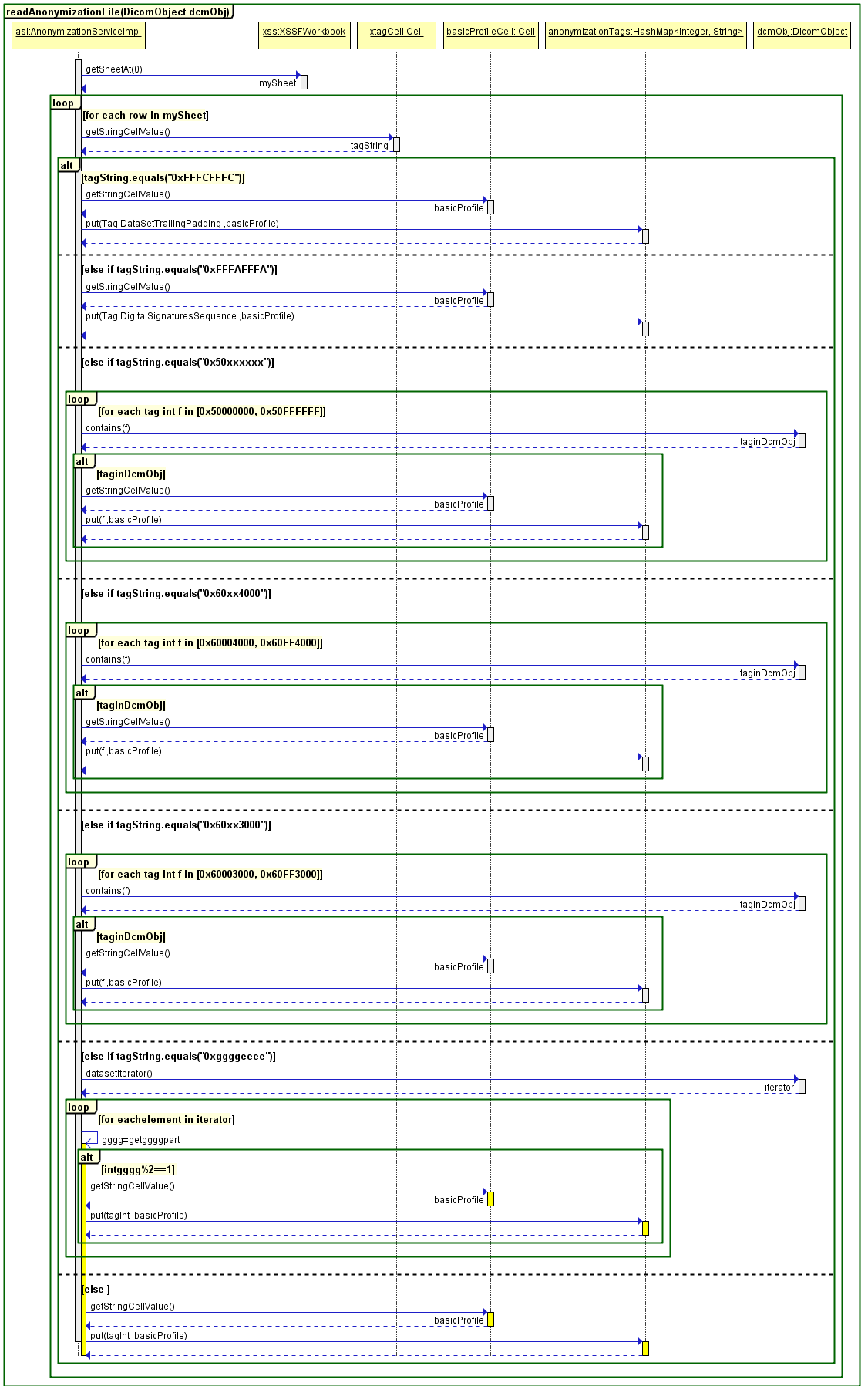


Figure 5. Read anonymization xml file and create the anonymization hashmap

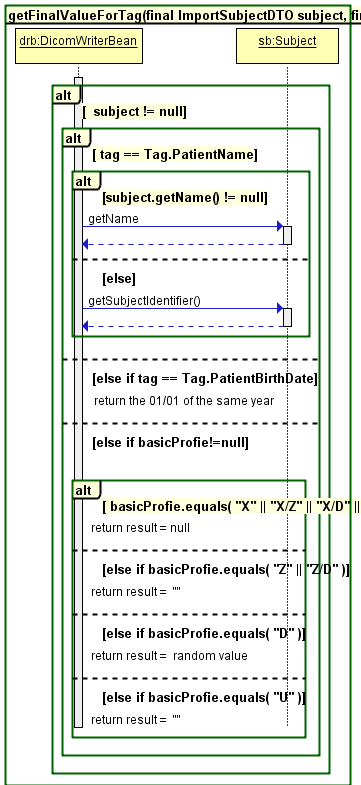


Figure 6. getFinalValueForTag method ameliorated: including basic profile concept

The drawback of the above design is that the dicomObject created from the dicom image is solicited too many times which many have a negative consequence on the anonymization process performances.

### Technical Specification of Shanoir NG Anonymization V2

The idea behind this conception is to iterate over the dicom tags in the image instead of iterating over the dicom tags to anonymize specified by the DICOM norm (figure 7).

The first step is to read the dicom object and to create three dicom tag hashmaps: dicomPrivateTags, dicomPublicTags and dicomBigHexaTags as illustrated by figure 8. Then, the anonymization service will read the anonymization file to create one HashMap<Integer, String> containing the list of DICOM tags to anonymize and one HashMap<String, String> containing specific tags to anonymize that cannot be converted directly into Integer (figure 9).

The anonymization service will iterate over dicomPrivateTags, dicomPublicTags and dicomBigHexaTags hashmaps and anonymize only the tags specified in the anonymization file.

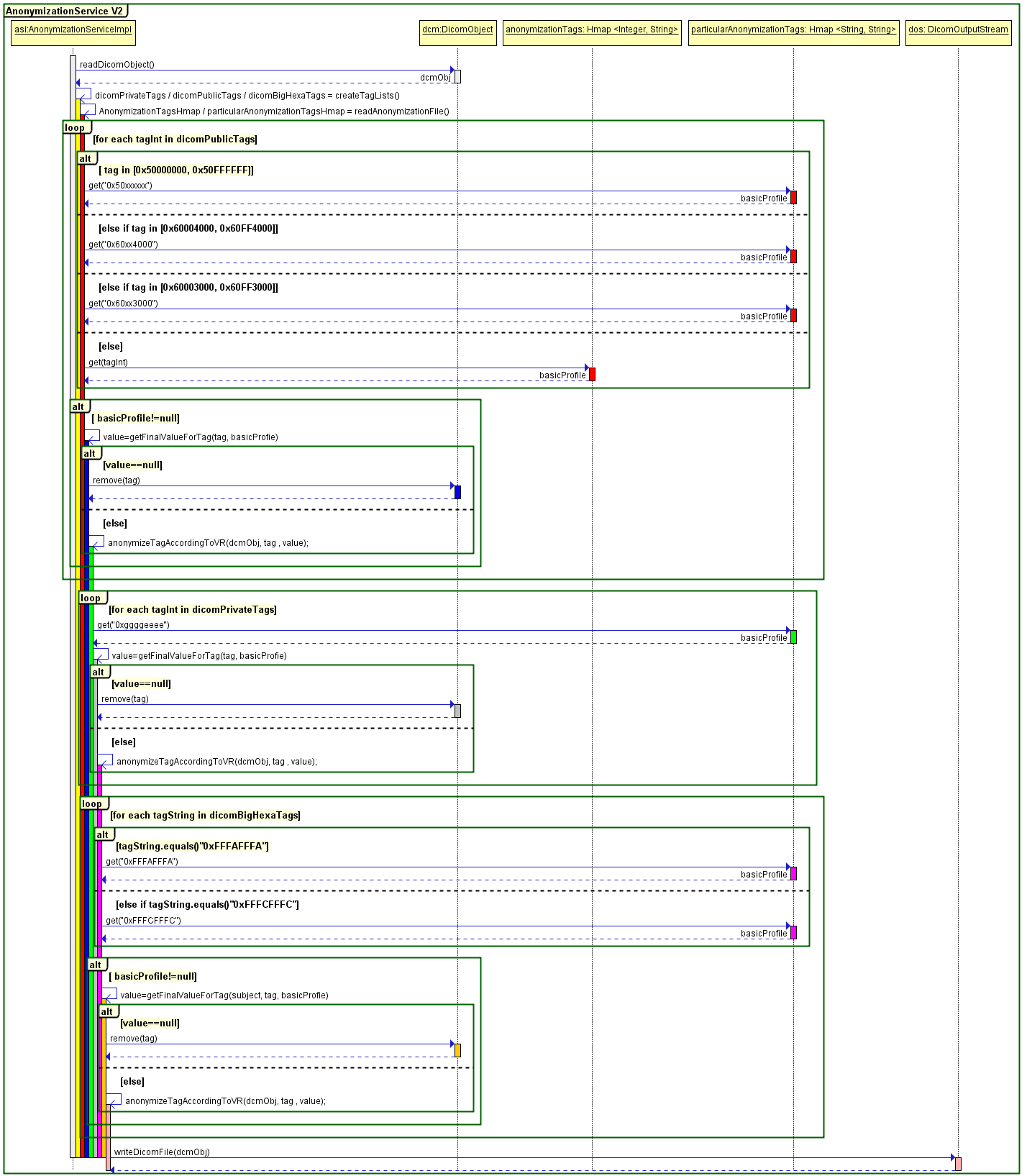


Figure 7. Anonymisation process in Shanoir NG V2



Figure 8. Read DicomObject and create a set of hashmaps

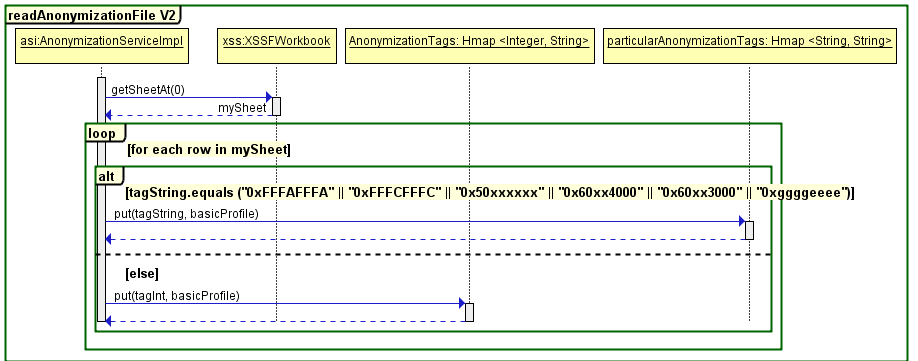


Figure 9. Read anonymization file

### Technical Specification of Shanoir NG Anonymization V3

The objective of this conception is to create a hashmap of tags that are present in the dicom image and need to be anonymized. Thus, the anonymization service reads the xml anonymization file and creates hashmaps as depicted by figure 9. Then, reads the dicom image and store only the dicoms to be anonymized in an hashmap<Integer, String> as illustrated by figure 11. Finally, it anonymizes the tags in the hashmap based on theire basic profile as depicted by figure 10.

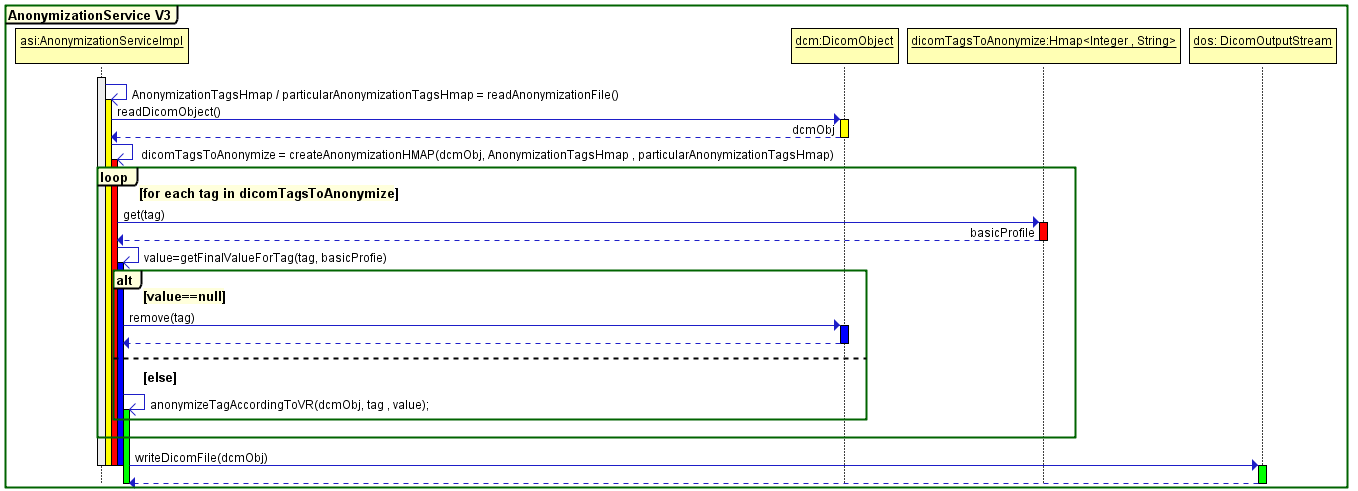


Figure 10. Anonymisation process in Shanoir NG V3

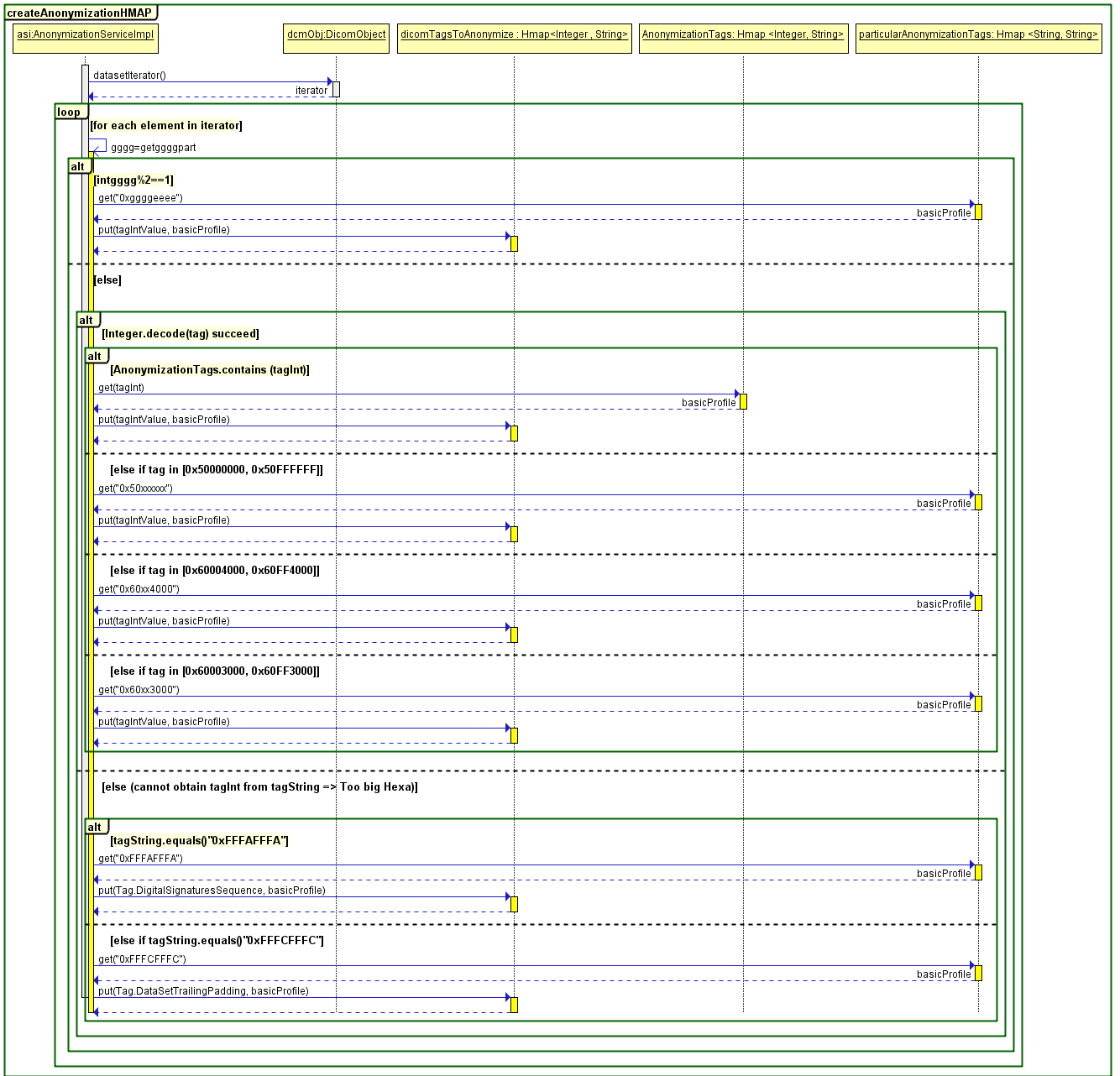


Figure 11. Read DicomObject and create a hashmap of tags to anonymise

### Email

Send email

Define email content and when it is sent.

### New folders

TODO