B2SAFE metadata management

version 1.2 by Claudio Cacciari, Robert Verkerk, Adil Hasan, Elena Erastova

Introduction

The B2SAFE service provides a set of functions for long term bit stream data preservation:

- Replication of objects (data and/or metadata) across multiple sites geographically distributed
- Data integrity check of the copies
- Data world-wide unequivocally identification

Each function is implemented through one or more service operations.
Other benefits of these functions are the possibility to easily cite such data through persistent and reliable references and the possibility to choose the geographical distribution in order to have the data closer to the users or to other services.

The data are uploaded to the B2SAFE storage resources through multiple interfaces and protocols. The latest version of the B2SAFE software does not distinguish between data and metadata.
Metadata is a form of documentation that uses terms or statements to describe data, it can contain fields that provide standardized structured information and many metadata standards[1] exist across a broad range of disciplines and applications[2].
The current document aims to describe a method to support metadata management capability[3] through the B2SAFE service.

Requirements

The metadata can be associated to the data in three different ways:

1. They could be embedded in the data, for example like a text header of a binary file.
2. They can be uploaded as a separate object.
3. They can be linked to the data, but not uploaded with them.

---

[1] http://www.jiscdigitalmedia.ac.uk/guide/putting-things-in-order-links-to-metadata-schemas-and-related-standards

[2] http://researchguides.library.yorku.ca/content.php?pid=382352&sid=3873543

[3] a possible definition of capability: the combination of tools, processes, skills, behaviour and organisation that delivers a specified outcome (http://www.strategyand.pwc.com/global/home/what-we-think/multimedia/video/mm-video_display/what-is-a-capability)

The B2SAFE service should be able to store the metadata and their relation with the data, covering all the three cases, according to the input received by WP4[4]. In fact features like B2SAFE metadata harvesting, indexing and discovery are requested by the communities and thus means to expose explicitly the metadata and their relations to external services and users should be implemented.
Moreover, it would be important to preserve the data structure according to the data provider[5] wish, which means, for example, that the service should not force it to create a package containing the data, if this is not a request of the data provider itself.

Context

It can be useful to specify some aspects of the context before to go into the details of the proposed solution. The data and metadata objects can be related each other with different multiplicity: there could be *n* metadata objects pointing to a single data object or vice versa. They can be uploaded at different points in time and they can change independently or even disappear. The distinction between data and metadata themselves can be ambiguous because what a data provider calls metadata, could be interpreted as data by a data consumer. Finally there are different types of metadata and they could be managed differently according to their types (the three main categories are "descriptive metadata", "structural metadata" and "administrative metadata", but they can be named in a different way, like in the DCC Digital Curation Manual[6]).
Now imagine to replicate a collection (including data and metadata objects) and to want to keep track of the relations of data and metadata, for example storing this piece of information into an external registry. This implies to update such registry each time the collection change. Therefore it could happens that, when even just a single change affect an object (e.g. it is replicated, it is deleted, ...), multiple entries of that information registry have to be updated.
The complexity of this scenario can be modeled defining multiple levels for the support of the metadata management.

- **basic level**: we make really few assumptions. The users are not constrained in terms of formats or packages, they can upload set of data and metadata objects separately and change them later. The users get the maximum flexibility and the minimum set of guarantees in terms of coherence of the information.
- **intermediate level**: there are some constraints. The users must provide packages in order to allow B2SAFE to keep consistent the information about data and metadata relations

[4] https://confluence.csc.fi/display/EUDAT2/Metadata+in+the+CDI
[5] *data provider* here is a synonym of *data producer,* intended like described in "Reference model for an open archival information system (OAIS)", chapter 2.1 (http://public.ccsds.org/publications/archive/650x0m2.pdf). The same reference is valid for *data manager* and *data consumer.*
[6] http://www.dcc.ac.uk/sites/default/files/documents/resource/curation-manual/chapters/metadata/metadata.pdf

and/or they must agree on a metadata format to allow B2SAFE to support the indexing of the metadata objects.

- **top level**: the users abide by all the constraints in terms of package and metadata formats and they got the maximum set of guarantees about the consistency of the collections and all the features in terms of metadata indexing and discovery.

The following table should clarify the expected features of the proposed solution in relation with the different level of support of the metadata management.

features support levels	metadata identification and manifest validation	metadata relations consistency	descriptive metadata indexing	domain metadata indexing	constraints
basic level					just give me a manifest
intermediate level A					give me a package and a manifest
intermediate level B					give me a manifest and descriptive metadata format that I can understand
intermediate level C					give me a manifest and descriptive and domain metadata formats that I can understand
top level					give me all

The next chapters will describe a proposal which aims to implement the basic level of support as a preliminary step for the other levels. This progressive approach will allow the B2SAFE team to plan properly the development and to provide, at least a minimal set of functions, in a reasonable time.

Proposed solution

There are many initiatives that have already investigated this topic, most part of them originated from the library world. A popular way to archive and exchange data seems to exploit the BagIt packaging format[7], which has been mixed with the Research Object Bundle[8] to define an (almost) interoperable hybrid called Research Object BagIt archive[9]. It is interesting to note that free open source tools are already available to manage all these formats.
However, the usage of a package format conflicts with the preservation of the data structure, because it would be necessary to move the collections within a predefined root directory inside the package. The extraction of the data from the package after the upload could be a workaround, but it would be quite expensive in term of computational resources. Besides it would mean to store the data in a structure which differs from that (the package) uploaded by the user and this is not what we want.
Therefore, this approach cannot be adopted as the general solution, but as an opportunity that the service can offer to the data providers which are happy to store their data as packages for their own convenience.

The manifest

If the metadata are not associated to the data because encapsulated in the same package, then the service must offer another way and it can be the definition of *a manifest file* which *contains the links between data and metadata*. The manifest format can exploit the abundance of metadata standards already defined. Among them the *METS schema[10] seems interesting*. According to the National Library of Australia, METS can be used as a means of transmitting a representation of an object (physical or digital or partially digital) from one system to another. It can:

- Fully describe the object and its components.
- Encode the metadata needed to aid its preservation and future access.
- Represent the physical and/or logical structure of highly complex objects.
- Represent collections of objects, even where these objects are not stored in the same repository.
- Support a range of submission and dissemination scenarios.
- Deliver representations of an object appropriate to the scenario by using a protocol such as OpenURL to request the required parameters.[11]

---

[7] https://en.wikipedia.org/wiki/BagIt
[8] https://researchobject.github.io/specifications/bundle/
[9] https://github.com/ResearchObject/bagit-ro
[10] http://www.loc.gov/standards/mets/
[11] http://www.dlib.org/dlib/march08/pearce/03pearce.html

It can also be a bridge between different protocols and formats, because in principle it is possible to map part of the METS information into a Dublin Core[12] description, useful to expose the metadata through OAI-PMH[13] or to use it to export the data into a Archival Information Package [14], or to include in the METS document itself some PREMIS elements as shown in the "ECHO Dep Generic METS Profile for Preservation and Digital Repository Interoperability"[15].
The manifest file is provided by the user who upload the data and its content is under her responsibility. The B2SAFE team could provide tools to simplify its generation, like a template.
The objective of the manifest file is to allow the B2SAFE service to keep track of the metadata in the simplest way possible, but this does not prevent the definition of more complex and structured services on top of or next to B2SAFE. Therefore, in the next paragraphs we will distinguish between the core functions which should be provided by B2SAFE and others which can be integrated and combined with B2SAFE, but are decoupled from it, thus supporting a modular approach.

Metadata core functions

The core functions of the B2SAFE metadata management are:

- **metadata ingestion**, which means that the metadata are clearly identified. This identification will not happen synchronously with the upload of the objects (data and/or metadata), but in a second step. *The manifest file can be uploaded in advance or just after the data/metadata ingestion*. In this way the current upload channel based on GridFTP, which does not provide means to identify explicitly the metadata objects, can still be supported.
- **metadata replication**: the metadata are replicated according to a replication policy which is agreed at EUDAT level. Initially the policy will be based on the agreement reached within the Technical Committee to support only the replication of data and metadata coupled together (a manifest file provides such link) if both are available[16]. In this way we expect to decrease the effort to keep track of the relations between data and metadata. *The manifest document will be replicated together with the data set and the related metadata without any change*. In fact, *it is a document who describes data and metadata independently from the data and metadata hosting repository*.
- **metadata retrieval**: users and services should be able to retrieve the metadata objects in the same way they do for the data objects. Therefore, in order to identify the metadata, they need to be able to understand the manifest file.

[12] http://www.dublincore.org/documents/dces/
[13] http://www.openarchives.org/pmh/
[14] https://en.wikipedia.org/wiki/Open_Archival_Information_System
[15] http://www.loc.gov/standards/mets/profiles/00000015.xml
[16] https://confluence.csc.fi/display/EUDAT2/Metadata+in+the+CDI

- **metadata update**: the users can change the manifest document after its ingestion. B2SAFE will validate it and replicate it, if required.

The implementation of the core functions implies the definition of B2SAFE workflows, which we can foresee very similar to those already available for data objects, so the required effort to implement them seems reasonable. However, it is necessary to pay attention to the desired behavior in case of missing objects or errors in the manifest document.

- The manifest document is missing: when? Since the manifest is uploaded asynchronously, B2SAFE cannot determine if the manifest is really missing or just delayed. But that does not matter. The approach based on the aforementioned functions allows B2SAFE to consider the objects as they are all data objects until it receives the proper manifest. For example, it can replicate all the objects and later add the manifest to the ingestion node and the replica nodes. This is possible since we are assuming that *the manifest document does not trigger any action targeting the data and metadata objects*.
- The manifest is available, but some objects listed in the document are missing. This means that the manifest is inconsistent. B2SAFE can validate the manifest and discover such inconsistency: the upload of the manifest should trigger automatically the validation and provide back the result in some way. For example, it can rename the manifest introducing a tag “valid” (or “not valid”) as a suffix to the name of the object. If the manifest is not consistent then it should be a valid reason to stop any replication, otherwise we will propagate the inconsistencies. If the manifest is added or changed later and the objects are already replicated, then there could be two options:
    - the replicas are deleted.
    - the manifest is replicated and tagged “not valid” so that anyone accessing the replicas is aware of the inconsistency.

  However, it is possible that the manifest is not really inconsistent, but just that the upload of some objects is delayed as regards to the manifest. In this case B2SAFE can offer two options:
    - the manifest validation is triggered each time a new object is uploaded into the same collection/directory. This is an expensive behavior and should be limited only to scenarios where it is required to trigger immediately new actions on the uploaded data.
    - the manifest validation is triggered periodically with a timeout. The frequency and the timeout should be set according to the specific scenario.

So far we have assumed implicitly the belonging of a manifest to a data (and metadata) set. However, B2SAFE needs an explicit way to associate the manifest and the related collection(s) otherwise it cannot support the aforementioned features. The manifest is conventionally placed in the same directory of the objects, then to put it in relation with data and metadata is easy if the

collection is structured as a single flat directory, but what about the scenario with a tree structure with multiple sub-directories? The manifest should be placed in the root collection. Hence one manifest for each uploaded collection could be enough if the collection is managed (read "replicated") as a whole by the service. Which is not true anymore if we admit the possibility to replicate subsets of the uploaded collection. In this case the B2SAFE can offer two options:

- the service will search the parent collections up to the root to find the manifest and then it will add it to the replicated subset with minor changes to reflect the fact that only a subset of the main collection is replicated. This is a flexible approach, but it adds some overhead to the replication.
- the replication of sub-collections is forbidden by default. Easier, but less flexible.

Note that so far there has not been any need to mention the multiplicity of the metadata objects in relation to the data objects or vice versa, because the described approach is agnostic in respect with it.

Metadata additional functions

The metadata additional functions are implemented on top of the core functions, hence they rely on them, in particular on the availability of the manifest file. Currently not all the additional functions can be clearly defined because there is an ongoing discussion within EUDAT about them. We will try to list here just some examples, which are based on the aforementioned requirements.

- **Metadata-data relation registration**: the PID service is candidate to keep track of the link(s) between a metadata object(s) and the related data object(s) in the PID record. It is not clear how these pieces of information can be recorded in the PID service and keep updated. In principle the B2SAFE service could add this information when it triggers the PID registration of an object, but then, to keep it aligned and consistent with the status of the related objects in a synchronous way would be really difficult. If we accept that the coherence of the information recorded in the B2HANDLE is not granted in real time, but only consolidated over time, then we can mitigate the complexity of the process and It would be possible to implement a separate mechanism to pull such information from the B2SAFE service and push it to the B2HANDLE.
- **Metadata harvesting and indexing**: it is a function already implemented by the B2FIND service. However in order to harvest the metadata stored behind the B2SAFE, it would be necessary to publish them via the OAI-PMH protocol[17]. B2SAFE team can work with the B2FIND team to share the effort of the development of a OAI-PMH compliant interface. If the B2FIND service should result not suitable for this purpose, then an other tool can be adopted (elasticsearch, solr, etc.).

[17] https://www.openarchives.org/pmh/

- **Metadata publishing**: within EUDAT there is the proposal to implement a new service called Metadata Store, which should be available only on a subset of the EUDAT CDI nodes and publish the metadata in a user friendly way. In order to support this service the B2SAFE should provide the metadata retrieval function as described in the previous chapter. Even push mechanisms could be implemented if required, but it is too early to decide it.

Manifest for data objects (exotic features)

It is important to note that, even if the manifest document has been defined to support metadata management, it can be (ab)used for further scopes.

- It could be uploaded to describe the collection even if the collections does not contains metadata objects. In fact the manifest itself is a metadata object and can contain descriptive and administrative metadata.
- It could be used to define virtual collections because it can describe a collection of objects stored in different locations (iRODS paths with different root directories).
- There could be multiple manifest documents, each one providing a different view of the same collection.
- In the (distant) future B2SAFE could even base the replication mechanism on the manifest, not on the iRODS path.
- The manifest could be associated to a PID and the user could then get all the PIDs of a collection using the information of the manifest itself.
- The manifest could be hierarchical, so it would be possible to build collections of collections linking together manifest documents.
- It is possible to imagine an index of collections based on the manifest content, which is different from the metadata index.