



European Data
Grant agreement number: RI-283304

B2SAFE Repository Package – Design document

Author(s)	Stéphane Coutin (CINES)
Status	Ready for review
Version	1.0.5
Date	9 jan 2015

Abstract: Repository package aims to deliver an easy to deploy interface enabling some of the most commonly used data repository to benefit from EUDAT B2SAFE service. This document presents the design of the different components.

Document identifier: EUDAT-WP2-EUDAT-OA	
Deliverable lead	
Related Work package	
Author(s)	Stéphane Coutin (CINES)
Contributor(s)	
Due date of deliverable	
Actual submission date	
Reviewed by	
Approved by	
Dissemination level	PUBLIC
Website	www.eudat.eu
Call	FP7-INFRA-2011-1.2.2
Project Number	283304
Instrument	CP-CSA
Start date of Project	01/10/2011
Duration	36 months

Disclaimer: The content of the document herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the document is believed to be accurate, the author(s) or any other participant in the EUDAT Consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the EUDAT Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the EUDAT Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

CONTENTS

1	GENERAL DESIGN	4
1.1	Objective	4
1.2	Architecture	4
1.3	Implementation	5
2	MODULE RP_CORE	5
2.1	RP_CORE API	5
2.2	Parameters	5
3	B2SAFE SERVER SIDE COMPONENTS	6
3.1	Data organization and users	6
3.2	Configuration	6
3.3	RP_INGEST features	6
3.3.1	Data object replication	6
3.3.2	Delete a data object	8
4	THE B2SAFE_RP_PILOT PROGRAM	8
4.1	Command and result text files	8
4.1.1	Command files	8
4.1.2	Result files	9
4.2	Execution modes	9
4.2.1	Interactive mode	9
4.2.2	Batch mode	10
4.3	Replication features	10
4.3.1	Replicate a set of data objects	10
4.3.2	Delete a set of data objects	10
4.4	Access features	10
4.4.1	Retrieve a set of Data Objects by replica full path	10
4.5	Control features	10
4.5.1	Test connection	10
4.5.2	Test handling of DO file	10
5	REPOSITORY INTERFACE COMPONENTS	10
5.1	Data stored on a file system	10
5.2	RP_DSPACE	10
5.3	FEDORA	11
5.4	ePRINTS	11

1 General design

1.1 Objective

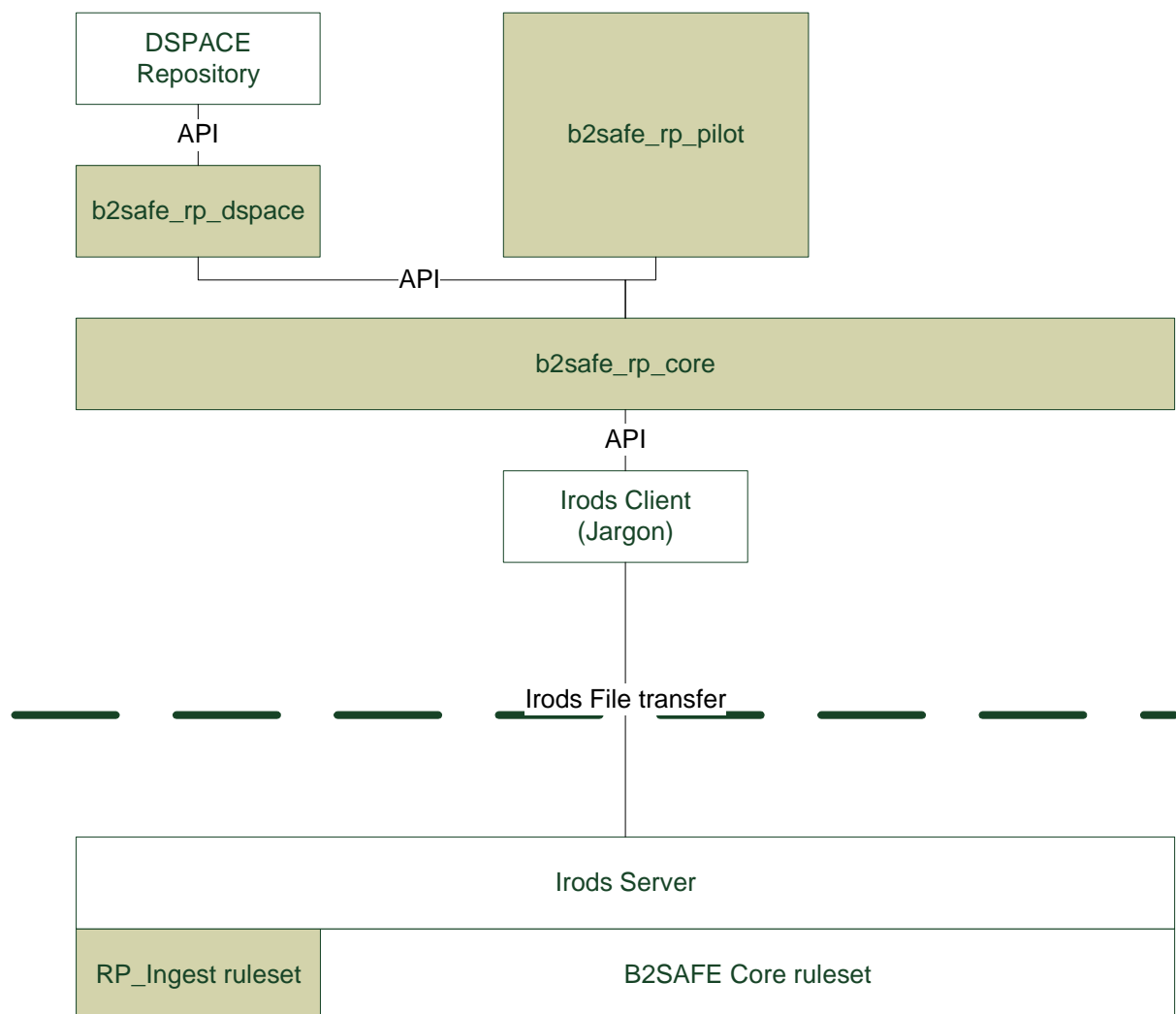
The aim of this development package is to enable communities running a data repository built with a popular repository solution to benefit from the EUDAT B2SAFE service.

To do so, special packages (named connectors) will be developed with which popular data repository solutions (e.g. FedoraCommons, DSpace, ePrints, ..) can easily be configured to interface with B2SAFE service.

1.2 Architecture

The connectors are interfaces between repositories and EUDAT CDI, more specifically the B2SAFE service. In order to keep the complexity under control, it is crucial to factor into a common engine all the cross repositories features and leave only a limited set of repository specific components.

The result of this is that the connector architecture will have a logical 3 layers architecture:



We need also to plan for different interfaces to EUDAT CDI : today we could use direct iRods access to B2SAFE or B2STAGE gridftp door, in the future we could use HTTP API currently under development.

1.3 Implementation

In the initial version we are making some implementation choices in order to build an operational version while dealing with the available level of resources (human resources and platforms). So we will aim to keep the design simple and modular.

The choices are the following:

- The repository side components (b2safe_rp_core, b2safe_rp_dspace and b2safe_rp_pilot) are developed in java.
- B2safe_rp_core and b2safe_rp_dspace are libraries (.jar files) and will offer a documented API
- B2safe_rp_pilot is an executable and will use b2safe_rp_core library.
- On the B2SAFE side, the functionalities are delivered by a set of irods rules, combining B2SAFE core plus additional rules for ingestion (rp_ingest)

Note that all source files are available on github at :

<https://github.com/EUDAT-B2SAFE/B2SAFE-repository-package>

2 Module rp_core

2.1 RP_CORE API

The module exposes an API which must be used by all the repository specific modules. The API is high level and is dealing with application level logic. It aims to hide the technological stack used to exchange with the server side service and to remain stable.

The API is documented on a javadoc format and published as github pages at: <http://eudat-b2safe.github.io/B2SAFE-repository-package/>

A typical usage of the API is as follows:

- Instantiate a DataSet object
- For every data object to be replicated, create an occurrence of DataObject with associated list of AVUMetaData (those 2 classes offer getters and setters methods)
- Call the methods replicateOneDO or replicateAllRequestedDO to launch the replication in B2SAFE.
- The methods will return the result as DataObjects with updated properties. You can get the data object status to check if replication has been successful and get the EUDAT PID.

An example of a program using the API is the b2safe_rp_pilot program (see later in this document).

2.2 Parameters

The connector is configurable to deal with the different environments.

Parameters are stored in the file config.properties. An example file is available on github

3 B2SAFE Server side components

On the B2SAFE side, the application is implemented as a set of iRods rules coming in addition to the rules developed in the B2SAFE-core module.

The additional rules are essentially dealing with the fact that repository package deals with the initial ingestion of a data object in B2SAFE service.

The plan is to, as soon as possible, include the rp_ingest rules into the B2SAFE-core module.

3.1 Data organization and users

On B2SAFE, we have a dedicated space for each data project starting from a dedicated root collection (example: /MyZone/dataproject1). We define one user for the project.

The user has the following access rights to this collection and sub-collections:

- Read/write during the ingestion
- Read only when the data object is archived (this can be deactivated)

We have also a generic user for the node. This user will own all the data objects once archived.

3.2 Configuration

A minimum set of parameters must be set up in addition to the parameters required by B2SAFE-core.

They are configured on the top of the rp_ingest.re file and drive the way the archived data ACL are managed.

3.3 RP_INGEST features

3.3.1 Data object replication

3.3.1.1 Attributes associated with each Data Object

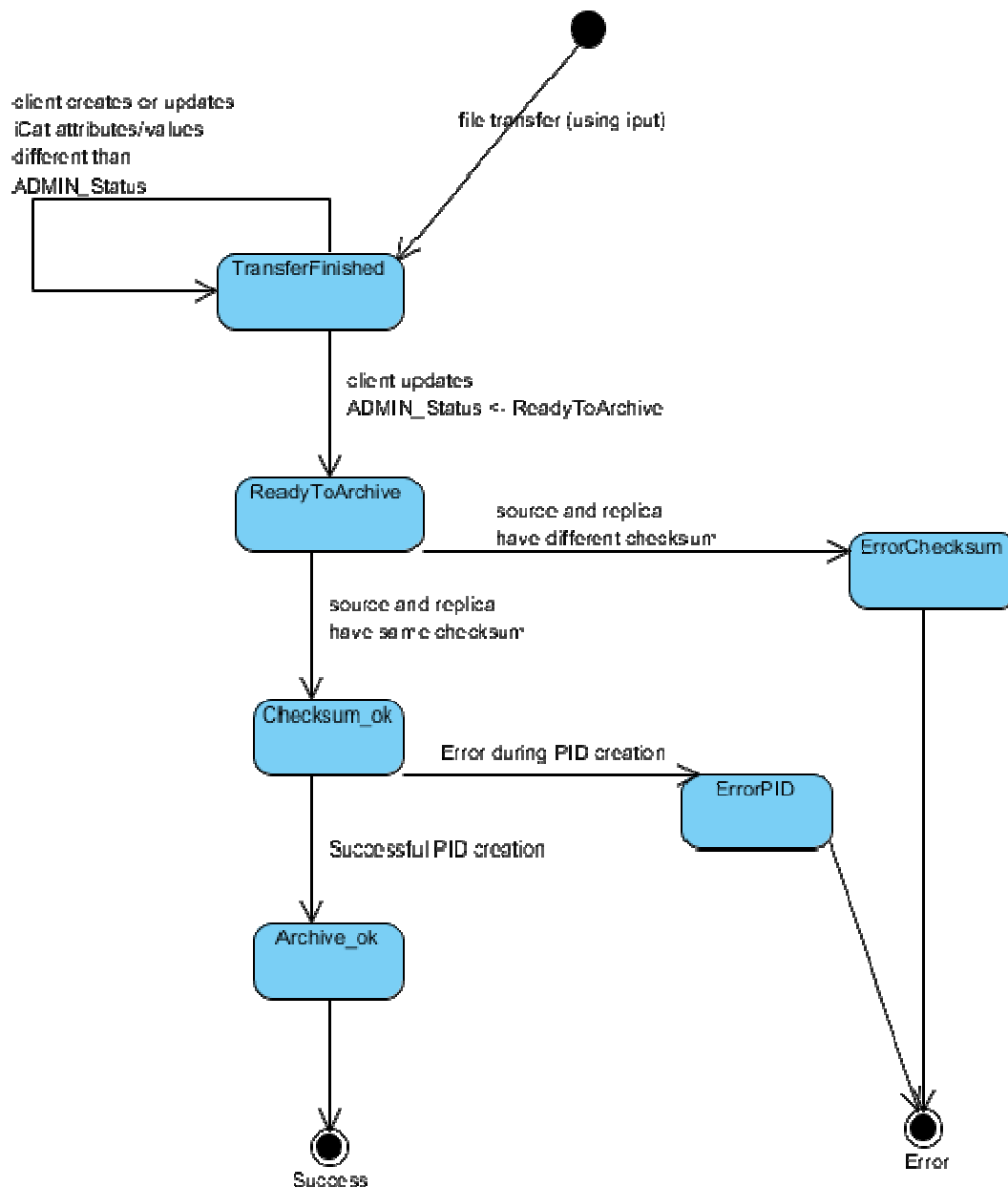
A number of attributes are associated to each DO in iCAT. They are used by the B2SAFE ruleset.

Attribute	Description / Value	Unit
EUDAT/ROR	URL for the repository PID handle of DO	
PID	Handle PID of the DO replica in B2SAFE.	
resource_id	The ID of the storage resource as defined in the EUDAT RCT tool	
INFO_TimeOfStart	The time stamp of the beginning of the transfer.	
INFO_TimeOfTransfer	The time stamp of the transfer finishes.	
OTHER_From	Source repository for the data object	
OTHER_original_checksum	the checksum provided by the user	
INFO_Checksum	MD5 checksum calculated once the transfer has been completed	
ADMIN_Status	The status in the ingestion workflow.	

	See next chapter for possible values.	
--	---------------------------------------	--

3.3.1.2 Replication transaction

The Data Object (DO) is received in a deposit space. This kicks off the workflow described on the ADMIN_Status transition state diagram copied below



In Case of success, depending on the parameters, the file might become read only.

3.3.1.3 Handle record

The handle record is created with the following values. They are standard in EUDAT B2SAFE

Field	Value
URL	URL to access this copy of the object on B2SAFE
10320/LOC	List of references to other copies of the object.
ROR	This is the Record Of Reference if available, usually the PID for the object in the repository
CHECKSUM	MD5 checksum

3.3.2 Delete a data object

This feature should be limited to administrator. Its initial objective is to facilitate the testing.

When a data object is deleted on B2SAFE, a trigger kicks off the EUDATePIDremoveForce rule.

Note that this rule doesn't look for the object reference in other replicas.

4 The b2safe_rp_pilot program

The b2safe_rp_pilot program aims to deliver the repository packages features on a repository independent way.

4.1 Command and result text files

4.1.1 Command files

Most of the features are controlled from text files providing the information about Data Objects to replicate or to retrieve.

The format is the same for all the files used in b2safe_rp_pilot and is described below. The path and name of the command files are defined in the properties file.

File format is text, one line per Data Object with ';' as field separator.

The first line is a header including field names. It must be in the file. The line is

fileName;localFilePath;remoteDirPath;ror;eudatPid;

The list of fields is in the table below:

filename	Data Object file name.
localFilePath	Path to read Data Object file on the repository (shared space)
remoteDirPath	Collection to use to store the Data Object on the EUDAT replication area, must end with '/' char.
Ror	ROR (Record Of Reference) : if any, this is the PID on the repository side
eudatPid	EUDAT PID for the replica (copied from the .DSS file)

Not all the fields are mandatory.

Here is an example:


```
fileName;localFilePath;remoteDirPath;ror;eudatPid;replicaLaunchDate;replicaEndDate;adminStatus;
test1.txt;C:\\tmp\\test1.txt;testcoll/sub1/id1
test2.txt;C:\\tmp\\test2.txt;testcoll/sub2/id2
test9.txt;C:\\tmp\\test9.txt;testcoll/sub1/id9
```

4.1.2 Result files

Those text files are used to provide output for the feature when relevant.

The application uses one file to log the results for all the operations (file path is on the 'replicationResultFile' parameter), and a file to log the result of all the operations resulting in error (file path is on the 'operationErrorResultFile' parameter).

The format is the same for all the result files used in b2safe_rp_pilot and is described below. The path and name of the result files are defined in the properties file.

File format is text, one line per Data Object with ';' as field separator.

The first line is a header including field names. It must be in the file. The line is

```
Operation;Status;StatusMessage;LaunchDate;EndDate;FileName;LocalFilePath;remoteDirPath;ror;eudat
Pid;
```

The list of fields is in the table below:

Operation	Type of operation, values can be REPLICATE, RETRIEVE, DELETE
Status	Values can be SUCCESS or ERROR
StatusMessage	A message providing details about the status, for example a error message.
LaunchDate	Start date for operation
EndDate	End date for operation
filename	Data Object file name.
localFilePath	Path to read Data Object file on the repository (shared space)
remoteDirPath	Collection to use to store the Data Object on the EUDAT replication area, must end with '/' char.
Ror	ROR (Record Of Reference) : if any, this is the PID on the repository side
eudatPid	EUDAT PID for the replica (copied from the .DSS file)

Not all the fields are mandatory.

4.2 Execution modes

The application can be executed in interactive mode or in batch mode. This is controlled by the parameter PILOT_EXEC_MODE.

4.2.1 Interactive mode

The interactive mode opens a menu on the console. Main features can be launched from the menu.

4.2.2 Batch mode

The current command line doesn't accept any argument. It runs the 'Replicates a set of objects' feature once.

4.3 Replication features

4.3.1 Replicate a set of data objects

This is the basic replication of a set of data objects from the repository into the EUDAT CDI.

It uses the *localIngestFileList* file as the input and writes results in the standard result files.

4.3.2 Delete a set of data objects

In this release the deletion is based on the replica object full path.

It uses the *localDeleteFileList* file as the input and writes results in the standard result files.

4.4 Access features

4.4.1 Retrieve a set of Data Objects by replica full path

In this release the retrieval is based on the replica object full path.

It uses the *localRetrieveFileList* file as the input and writes results in the standard result files.

4.5 Control features

They are available on console mode only.

4.5.1 Test connection

This is required to check initial setup and later on for any support issue.

The application check if the connection is already established and if not tries to connect. It displays the result.

4.5.2 Test handling of DO file

This opens the 'ToReplicate' text file as defined in the parameters and writes the values to the screen. There is no connection to B2SAFE.

5 Repository interface components

5.1 Data stored on a file system

This is not a proper repository, but it is possible to use the connector application to replicate data from a file system into EUDAT B2SAFE. We will also use it to perform some tests.

This works with the *b2safe_rp_pilot* program, the minimum being to produce a *RepoPackToReplicate* file. The file could be created manually or via a script, and should be copied on the dedicated directory.

5.2 RP_DSPACE

The *rp_dspace* module, developed by CUNI, is available on github at : <https://github.com/ufal/lr-b2safe-dspace> (see readme.md for details)

Note that at this point in time, this module uses a specific version of the *b2safe_rp_core* module (in fact, it uses a lower level interface which shouldn't be use as it doesn't include some of the business logic). The upgrade is not complex and will be done shortly, at least before anybody else than CUNI uses it.

5.3 FEDORA

To be implemented in a later release.

5.4 ePRINTS

To be implemented in a later release.