CERIF-CRIS Reference Implementation (CC-REFIM) and CERIF compatibility testing system

PROJECT CHARTER

Date: 02/10/2015
Version: 1.2
Prepared by: Valerie Brasse, Nikos Houssos, Jan Dvorak, Miguel-Angel Sicilia

Date	Version	Author	Comments
11/02/2013		Brasse, Valerie (IS4RI)	Document initialised.
11/03/2013	0.1	Brasse, Valerie (IS4RI)	First elements / notes.
09/05/2013	0.2	Nikos Houssos Valerie Brasse Jan Dvorak Miguel-Angel Sicilia	First draft based on discussions, input via email and Skype calls
15/11/2013	0.3	Nikos Houssos Valerie Brasse Jan Dvorak Miguel-Angel Sicilia	Updated according to feedback in the Bonn CERIF TG/Arch TG joint session
24/01/2014	1.0	Valerie Brasse	Version for contracts
13/02/2014	1.1	Valerie Brasse	Addition of CASRAI task (Profile identification and specification)
02/10/2015	1.2	Valerie Brasse	CASRAI tasks moved to IS4RI Revised timeline

PROJECT CHARTER

Project Name	CERIF-CRIS Reference Implementation (CC-REFIM) and CERIF compatibility testing system
Project Sponsor(s)	
Proposed Start Date	01 February 2014
Proposed End Date	09 June 2016

1. Project Background and Rationale

- euroCRIS, as promoter of CERIF and CERIF-based CRIS, must prove the feasibility and demonstrate the use of CERIF-CRIS.
- euroCRIS, as standard body, must ensure the quality and homogeneity of use of CERIF by systems claiming to be CERIF-compliant.
- Today, there is a growing number of CRIS implementations (in-house development and products), enough to justify the need for this quality/compliance validation service.
- Initially, CERIF-based implementation was supported by euroCRIS through provision of SQL scripts, but today more products such as CERIF-XML files and Toolkit, RDF generated from CERIF, and specific vocabularies, are available and should be showcased / should illustrate given use-cases.

2. Project Goals

- Provide a reference system able to reliably check and confirm CERIF compatibility of (a) CERIF-CRIS software platforms (e.g. PURE, CONVERIS, Symplectic Elements, Research in View, etc.) and (b) individual operational CERIF-CRIS systems currently in production, for example institutional, national or international CRIS (e.g. OpenAIREplus, EuroRIs-Net+); for example, candidates for checking with Scenario (b) are all systems that are registered on the DRIS.
- Develop an open system that will be available for any interested party to study and understand CERIF down to the implementation level.

3. Scope

- CERIF compatibility and interoperation can be defined at different levels, for example the Point-to-point, LOD-based and Distributed DB facets defined and discussed previously (January 2013) within the Board. Creating a full compliance suite for all levels is a huge task that can be tackled only incrementally. The current proposal concerns a first phase, which is feasible for implementation within a short timeframe.

- This proposal aims to tackle compatibility regarding the Point-to-Point facet, concerning the point-to-point transfer of CRIS data in CERIF-XML format. In particular, it concerns the ability of a CRIS system to provide information in standard CERIF-XML through a basic REST API.

- The fundamental components of the envisioned CERIF compatibility testing architecture are the following:

  1. **The CERIF CRIS Reference Implementation (CC-REFIM)**, which comprises the following parts (as depicted in Figure 1):

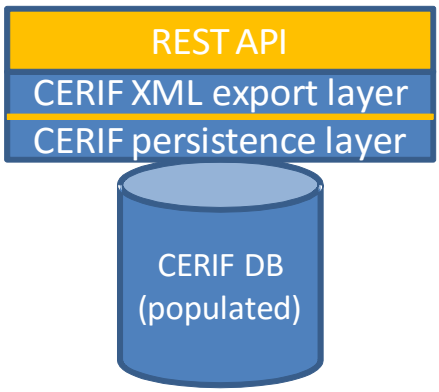

Figure 1. CC_REFIM architecture.

a. **The CERIF database**. A native CERIF database, fully populated with fabricated (but realistic-looking) data that covers the entire CERIF model. The entire data of this reference CERIF database is available for download in CERIF XML by anyone, including CRIS providers.
b. **The REST API wrapper**. A layer over the CERIF database that exposes its contents over a REST API in CERIF XML format. This functional element shall provide also write operations on the CERIF database (Create/Update/Delete) with CERIF XML data as input – however this might not be absolutely necessary for a first release of the CC-REFIM and the compatibility testing suite.
c. **The CERIF Export functional element**. This element handles the transformation of CERIF database data to standard CERIF XML.
d. Another functional element for general persistence functions over the relational database is foreseen to be useful both to the REST API wrapper and the CERIF Export element – however, this depends on the implementation technologies and architecture.

2. **The Compatibility Checker module**, which performs queries against the system under test, retrieves the results and validates them, using, when appropriate, also information that it retrieves from the CC-REFIM.
3. **The CRIS system under test**, which can fall into two categories: (a) an operational CRIS system currently in production (e.g. institutional, national or international CRIS), (b) a specific testing installation of a particular CRIS software platform / product made available by the CRIS platform provider for the purpose of testing by the euroCRIS CERIF Compatibility Testing. CERIF-compatible systems of both categories must be able to expose the data in the CRIS system in standard CERIF XML through a basic REST API. Systems of category (b) must be able, in addition to that, to import data in CERIF XML – at this phase this import will not be done through a standard API, i.e. will be performed by the platform provider in a way transparent to the euroCRIS CERIF Compatibility Testing System.

- Two compatibility testing scenarios (a) and (b) are envisaged, as depicted in Figure 2 and Figure 3.

- Scenario (a) aims to test CERIF compatibility of CRIS software platforms and in particular their ability to (i) to ingest data in standard CERIF XML and transform it to their local internal format and (ii) reply to basic REST API queries in CERIF XML.

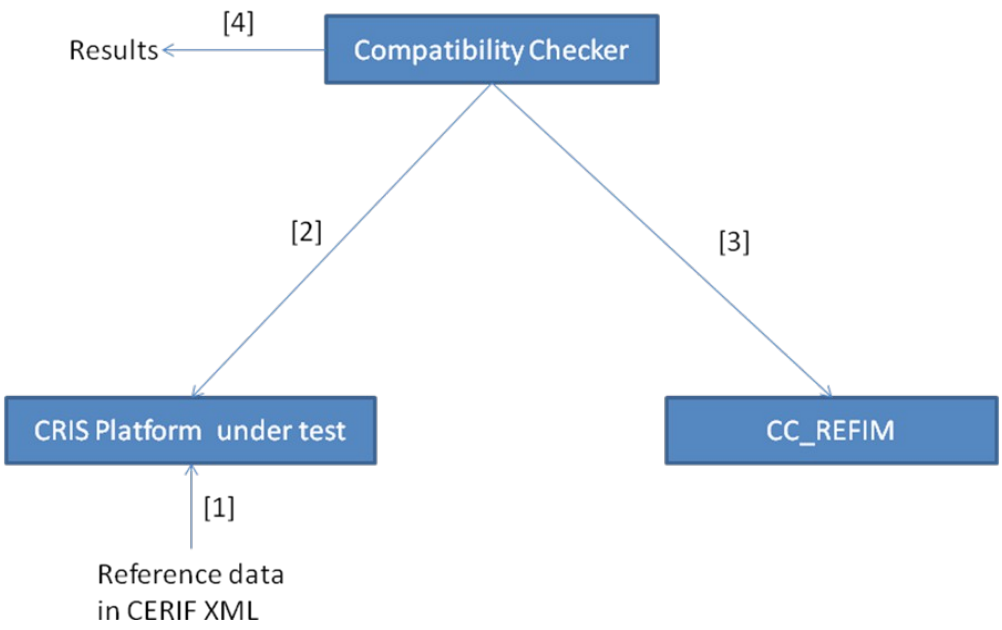

Figure 2. Scenario (a)

The basic flow of information for Scenario (a) is the following:

[1] The data of the reference CERIF database in downloaded in CERIF XML by the provider of the CRIS platform under test and is imported into the CRIS platform. The data import is triggered by the CRIS platform provider and is transparent to the CERIF compatibility testing software. After the completion of the import, the system under test should contain the full data in the CC-REFIM database in its internal format, which might not necessarily be the CERIF native relational database format. Before any compatibility testing activity begins, the CRIS platform provider confirms to the CERIF Compatibility Testing System operator the fact that the import of the reference CERIF XML data into the system under test has been completed and provides all the technical information necessary for the CERIF Compatibility Testing System to connect to the system under test.

[2] The Compatibility Checker performs a series of queries on the CRIS system under test, retrieves the results and stores them locally.

[3] The Compatibility Checker performs a series of queries on the CC-REFIM system, retrieves the results and stores them locally.

[4] The Compatibility Checker compares the results retrieved in steps [2] and [3]. If the results match, the Compatibility Checker confirms that the system under test is compatible; otherwise, it produces a detailed list of matching errors that is returned to the CRIS platform provider.

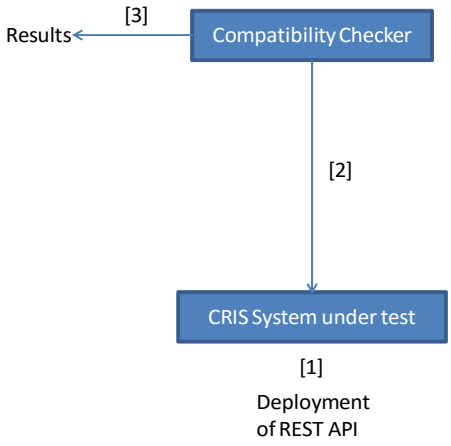

Figure 3. Scenario (b)

- Scenario (b) aims to test CERIF compatibility of operational CRIS system currently in production (e.g. institutional, national or international CRIS) and in particular their ability to expose the data in the CRIS system as standard CERIF XML via a basic REST API.

  The basic flow of information for Scenario (b) is the following:

  i. Before any compatibility testing activity begins, the CRIS platform provider confirms to the CERIF Compatibility Testing System operator the fact that the running CRIS system is ready for compatibility testing and provides all the technical information necessary for the CERIF Compatibility Testing System to connect to the system under test.
  ii. The Compatibility Checker performs a series of REST API calls on the CRIS system under test, retrieves the results and stores them locally.
  iii. The Compatibility Checker performs a series of compatibility checks on the results, including standard CERIF XML schema compliance. The compatibility checks might include a human inspection component for publicly available CRIS systems (e.g. to check that the returned information corresponds to the data published in the running CRIS system). If the compatibility checks are successful, the Compatibility Checker confirms that the system under test is compatible; otherwise, it produces a detailed list of matching errors that is returned to the CRIS system manager.

- The REST API should be minimal, so that it can be feasible for quick implementation by CRIS providers. It cannot be a full-fledged CRUD REST API, since this would raise the bar for vendors to implement it. Ideally, three major operations must be supported:

  a. Get the identifiers of all instances of a particular type of entity (e.g. get a list of all projects – however only the identifiers are returned for each project). The identifier returned for each entity instance must be adequate to allow retrieval of the full information about the particular project.
     Format of call: http://api.eurocris.org/{Entity}/
  b. Get all information about a particular instance of a particular type of entity (e.g. get information about project with id: 123456789). The information returned must contain the full record of the entity instance, including plain data fields, multi-lingual fields and 1st level of linked entities (in embedded mode).
     Format of call: http://api.eurocris.org/get/{Entity}/{id}
  c. Get information about all CERIF entities (core and link entities) supported by the CRIS system (i.e. definition of the CERIF subset supported by the CRIS system)
     Format of call: http://api.eurocris.org/entities

  One further operation is desirable but optional (is foreseen for a future API version after completion of the first version of CC-REFIM):

d. Return all information matching a search query about a specific entity type. The information returned for each matching entity instance must contain only the identifiers (e.g. get a list of all projects matching the term "nanotechnology"). The identifier returned for each entity instance must be adequate to allow retrieval of the full information about the particular project.
Format of call: http://api.eurocris.org/search/{Entity}/{value} or http://api.eurocris.org/search/{Entity}?{value}

Some notes on the format of calls:

i. If no verb/operation is mentioned, default is "get".
ii. If no id is specified, default is "all".
iii. It would be very "developer-friendly" to use as entity names the logical (instead of physical) names of entities specified in the CERIF data model specification, ideally with the first letter non-capitalised (e.g. project instead of cfProj, person instead of cfPers).

Examples of calls:
http://api.eurocris.org/project/123456789 *(get information of project with id 123456789)*
http://api.eurocris.org/project *(get list of identifiers of all projects in the CRIS system)*

4. Deliverables / Tasks

Tasks:

1. Setup of an operational CERIF reference database with remote access for development purposes
2. CERIF reference database population with fabricated, but realistic-looking data covering the entire CERIF data model (estimated: 1 person-month).
3. Development of the CERIF XML export function / CERIF persistence layer
4. Development of the REST API wrapper.
5. Development of the Compatibility Checker.
6. Description of CERIF-compliance criteria, procedures and queries for validation.
7. Setup of a production, high availability environment for running the operational CC_REFIM and compatibility checker.
8. Profile identification and specification

Deliverables:

1. Software components produced by Tasks 3, 4, 5
2. Data created through Task 2
3. Documentation from Task 6
4. Development/test and production environment for software