VRE infrastructure and Services – Intermediate Version

Workpackage	**5 VRE Infrastructure and Services Design and Development**	
Task		
Author (s)	Pedro Gonçalves	Terradue
	Serena Avolio	ACS
	Raul Palma	PSNC
	Simone Mantovani	MEEO
	José Manuel Gómez	ESI
Reviewer (s)	Raul Palma	PSNC
Approver (s)	Cristiano Silvagni	ESA
Authorizer	Mirko Albani	ESA
Document Identifier	EVER-EST DEL WP5-D5.10	
Dissemination Level	Public	
Status	Approved by the EC	
Version	1.1	
Date of Issue	14 December 2018	

Document Log

Date	Author	Changes	Version	Status
01/12/2016	Pedro Gonçalves	ToC	0.1	Draft
05/12/2016	Pedro Gonçalves, Serena Avolio, Raul Palma, Simone Mantovani, José Manuel Gómez	Updates from partner contribution	0.2	Draft
19/12/2016	Pedro Gonçalves	Update after final review	1.0	Draft to be approved by EC
14/12/2018	Cristiano Silvagni	Document status	1.1	Approved by the EC

Table of Contents

List of Figures

Definitions and Acronyms

Acronym	Description
CMS	Content Management System
COTS	Commercial Off-The-Shelf
EO	Earth Observation
GUI	Graphical User Interface
RODL	Research Object Digital Library
RO	Research Object
UI	User Interface
VC	Visual Components
VM	Virtual Machine
VRC	Virtual Research Community
VRE	Virtual Research Environment
WP	Work Package

Applicable Documents

Document ID	Document Title
EVER-EST DEL WP5-D5.2	Technical Note on Common Services
EVER-EST DEL WP5-D5.3	Technical Note on Digital Information and E-Collaboration Services
EVER-EST DEL WP5-D5.4	Technical Note on e-Research Applications Services
EVER-EST DEL WP5-D5.5	Technical Note on e-Learning Services

1 Introduction

1.1 Purpose of the document

The main purpose of this document is to list the software components selected for the consolidated design and the development of the EVER-EST Virtual research Environment (VRE).

1.2 Background

As detailed within the proposal, a consistent part of the infrastructure is based on modules and experience derived from previous Earth Science EU projects, including among others, SCIDIP-ES, GEOWOW and EarthServer. D5.1 provides details concerning the EVER-EST VRE architecture and initial set of corresponding software components. This document joins those elements and lists the software originating from the following tasks:

- Task 5.2 - Common services;
- Task 5.3 - Digital Information and E-Collaboration Services;
- Task 5.4 - E-Research Application Services;
- Task 5.5 - E-Learning application services.

1.3 Document Structure

This introductory chapter aims to provide key information to readers that do not belong to the EVER-EST technical team in order to provide the context and placement of this document in the overall WP5 activities. For a more general perspective the reading of D5.1 is recommended.

Chapter 2 provides a high-level overview of the EVER-EST VRE architecture focusing and lists the adopted software.

2 EVER-EST VRE Software

2.1 EVER-EST VRE Architecture

Figure 1 provides a high-level overview of the EVER-EST VRE architecture.

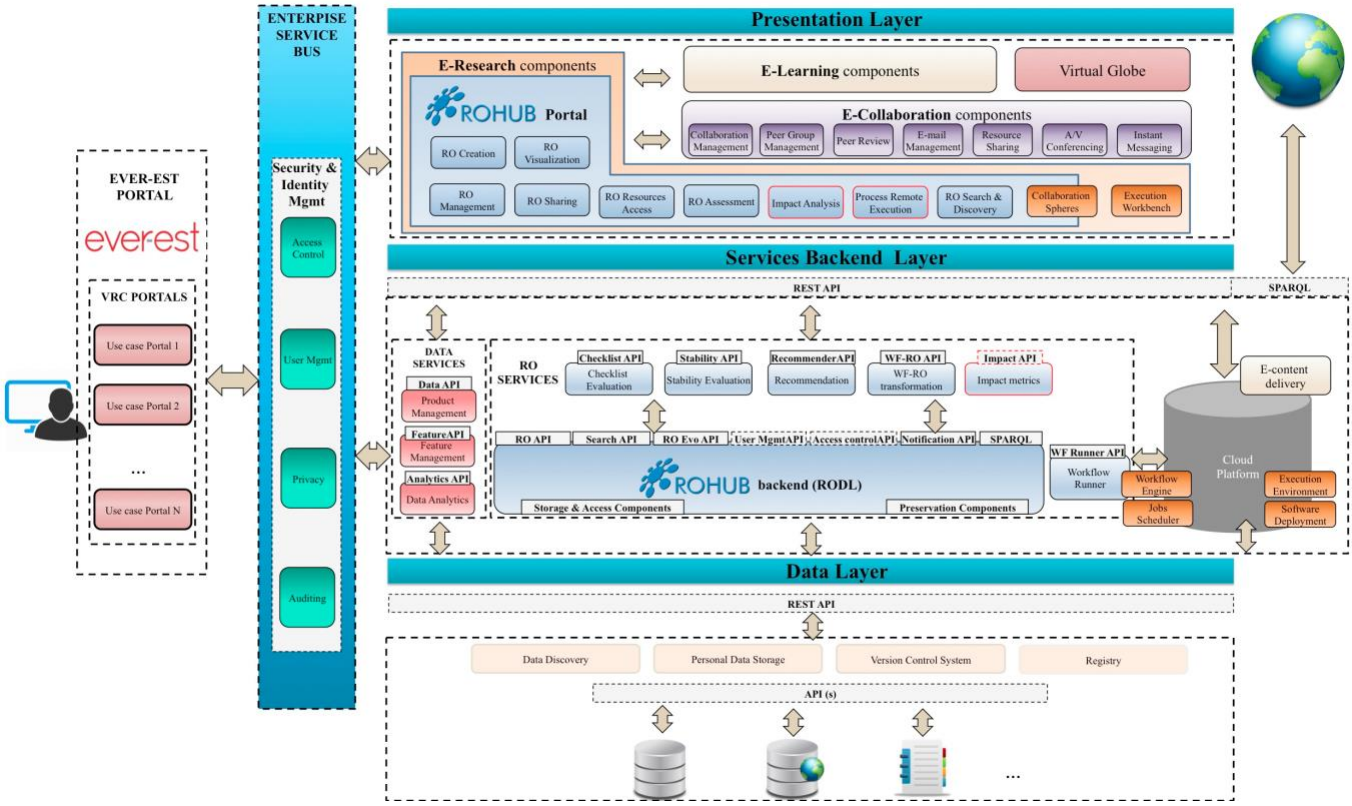

Figure 1 – EVER-EST VRE architecture

The architecture reflects the organization of functions across the various layers and with regards to the core element of the design the following can be identified:

1. **Presentation Layer** - on the top part of the architectural diagram - contains the core elements and technologies that shall guarantee the availability of those services and functions, which shall be directly used by the different communities. It includes the ROHub, the e-collaboration, e-learning and e-research services along with the mechanisms for Earth Science data discovery.
2. **Service Layer** - in the central part of the architectural diagram - provides both generic VRE services and Earth Science specific services. These components represent the reasoning engine of the e-infrastructure and actually orchestrate and manage the services available to the VRE final users.
3. **Data Layer** - bottom part of the design - references the data holdings made available to the Virtual Research Communities (VRCs): data is linked and proper means are provided, where feasible, to access it from the VRE. As a default setting, data will not be copied or duplicated, but will continue to reside on the provider's local servers unless it is directly retrieved by the user.

Included in the Presentation Layer is also the VRE portal that is the main access the components giving shortcut to each VRC user interface. While some services and functions can be seen as common and transversal to all the VRCs, a set of features and functions are specific for each of the communities and required an ad-hoc design.

The implementation of the EVER-EST VRE was divided in 4 main tasks:

- Task 5.2 - Common services;
- Task 5.3 - Digital Information and E-Collaboration Services;
- Task 5.4 - E-Research Application Services;
- Task 5.5 - E-Learning application services.

Each of these tasks focused on different software components and addressed development and configuration tasks at different levels. The following sections list the software used for the definition of the EVER-EST VRE according to those tasks.

2.2 Common services

The Common Services category represents the family of services, which are used by different elements of the EVER-EST infrastructure. They refer to the category of services that will guarantee the correct functioning between various infrastructure components and between the infrastructure and the final users.

The software components offering the common services are the WSO2 Identity Server[1], WSO2 Enterprise Service Bus[2], the WSO2 Data Analytic Server[3] and SeaFile[4].

The WSO2 Identity Server provides secure Identity Management for the EVER-EST Web Application services and APIs by managing identity and entitlements of the users. It is a commercial off-the-shelf (COTS) product that is configured and customized for the EVER-EST environment.

WSO2 Enterprise Service Bus provides, when it is needed, the bus for the messages exchanged between service provider and service consumer. It is a COTS configured and customized for the EVER-EST environment.

WSO2 Data Analytic Server offers data analytics functionalities. It is a COTS configured and customized for the EVER-EST environment.

SeaFile offers a repository for the personal data storage. Seafile guarantees file synchronization, version control, public link sharing, and desktop client and web API.

Detailed information on Common Services can be found in the [EVER-EST DEL WP5-D5.2] document.

2.3 Digital information and e-collaboration services

2.3.1 EVER-EST and VRC Portals

The EVER-EST portal and the VRC portals are developed using Django [5] (source code available at: https://github.com/django/django) and the Django-CMS [6] Content Management System technologies (source code available at: https://github.com/divio/django-cms).

[1] https://github.com/wso2/product-is

[2] https://github.com/wso2/product-esb

[3] https://github.com/wso2/product-das

[4] https://github.com/seafile

Information about the portals can be found in chapter 2 of the [EVER-EST DEL WP5-D5.3] document. The VRE portals source code is available at:

- https://github.com/ec-everest/vre/

Visual Components

The EVER-EST Visual Components (VCs) constitute the core of the EVER-EST Portal and VRC Portals. The VCs are developed as independent objects that could be added to a web page by adding just a few lines of code. The client side of the VC is developed using open web technologies (such as HTML5 standard, Javascript, jQuery, Bootstrap 3 Framework, and CSS 3).

The VCs currently under development are:

- Discovery Visual Component: allows datasets and data discovery via OpenSearch queries on OpenSearch enabled repositories;
- RO Manager Visual Component: allows RO creation, visualization and editing;
- Virtual Globe Visual Component: allows EO data visualization;
- Upload to Seafile Visual Component: allows files upload to Seafile within the VRC.

Information about the Digital Information Services and the Visual Components can be found in chapter 3 of the [EVER-EST DEL WP5-D5.3] document.

The VC source codes are available at:

- https://github.com/ec-everest/vre/

E-Collaboration Services

Django, the Content Management System (CMS) on which the EVER-EST Portal and the VRC Portals are based, allows the integration of most of the e-collaboration capabilities through third party plugins, while other e-collaboration services have been integrated as JavaScript extensions:

- Forum capabilities are provided by the django-machina [7] plugin (source code available at: https://github.com/ellmetha/django-machina);
- Internal messaging and notification capabilities are provided by the django-postman[8] plugin (source code available at: https://bitbucket.org/psam/django-postman/wiki/Home);
- Calendar/scheduling capabilities are provided by the django-todo[9] plugin (source code available at: https://github.com/shacker/django-todo);
- Chat capabilities are provided using EXtensible Messaging and Presence Protocol (XMPP)[10] standard, eJabberd[11] server side (source code available at: https://github.com/processone/ejabberd) and JavaScript XMPP Client (JSXC)[12] client side (source code available at: https://github.com/jsxc/jsxc).

[5] https://www.djangoproject.com/

[6] https://www.django-cms.org/en/

[7] https://django-machina.readthedocs.io/en/stable/

[8] http://django-postman.readthedocs.io/en/stable/

[9] https://github.com/shacker/django-todo

[10] http://xmpp.org/

[11] https://www.ejabberd.im/

[12] https://www.jsxc.org/

Information about the E-Collaboration Services can be found in chapter 4 of the [EVER-EST DEL WP5-D5.3] document.

2.4 e-research application services

2.4.1 RODL

Research Object Digital Library (RODL) is the main service provided by the ROHUB system. RODL offers research object storage and retrieval service which is the interface for storing and retrieving research objects and research object evolution service for creating snapshots and releases of the research objects.

https://github.com/rohub/rodl

2.4.2 ROHUB portal

ROHUB Portal provides User Interface (UI) for the RODL in the form of web page. ROHUB Portal implements a set of UI components that were designed to support user dealing with his research objects. ROHUB Portal source code was divided into two separate parts:

- Portal rest api, which provides data and user oriented functionality (login/logout/user session management):
  https://github.com/rohub/portal-restapi
- Portal web ui, which implements actual user interface visible components:
  https://github.com/rohub/portal-webui

2.4.3 Cloud platform

The EVER-EST VRCs will be able to access execution environments with capabilities enabling to specify, launch and terminate Virtual Machines (VMs). These capabilities are provided by Terradue Cloud Platform, a Hybrid Cloud infrastructure associating PSNC resources with a cloud controller managing scalable data management frameworks that are exploited by a Developer's application to deliver a Cloud Appliance. The software of the Cloud Platform is available at the OpenNebula GitHub - https://github.com/OpenNebula

The Terradue Cloud Controller uses the OpenNebula Sunstone GUI intended for both end users and administrators that simplifies the typical management operations in private and hybrid cloud infrastructures. This GUI allows to easily manage all resources and perform typical operations on them. This software is available here: https://github.com/OpenNebula/one

2.5 e-learning application services

The e-Learning Services for Earth Observation (EO) data are built using two core technologies: Data Cubes and Web Notebooks. The use of these technologies respond to the expectations of an easy integration of multi-source data and a capability to provide a complete analysis of the e-Learning modules in direct contact with the final user. The web context and the provision of services from both components allow participants to interactively execute the courses.

2.5.1 Web notebook jupyter

The Jupyter Notebook is an interactive computing environment that enables users to author notebook documents that include: live code, interactive widgets, plots, narrative text, equations, images and video. The Jupyter Notebook provides a complete and self-contained record of a computation that can be converted to various formats and shared with others. The Jupyter Project source code is available in their GitHub at: https://github.com/jupyter

2.5.2 Data Cubes

The EVER-EST Service Developer is provided with a VM containing the open source Australian Geoscience Data Cube (AGDC) software package. The Data Cube source code is available in specific fork in github at: https://github.com/data-cube/agdc-v2

2.5.3 Web Notebooks

The EVER-EST e-Learning Web notebooks are available at: https://github.com/ec-everest/e-learning-modules

This repository contains an evolving list of Web Notebooks that demonstrate their potential to fully support the Earth Observation data life cycle including examples to perform data access, data cleansing, exploration, and reproducibility. It will use information dissemination and data management tools that offer easy access to search and retrieval data repositories operations to allow extraction and distribution of single parameters or combined products. These Jupyter Notebooks take advantage of EO toolboxes (e.g. GDAL, SNAP), access data in HDFS, Docker data buckets and Data Cubes running on top of a Cloud based cluster.