

The background of the cover features a dark blue field with a large, stylized white pie chart on the left. A yellow 3D wedge is pulled out from the pie chart. A white, jagged upward arrow starts from the bottom left and points towards the top right. In the bottom left corner, there is a white lightbulb icon with radiating lines. The text 'Innovating innovation' is written in a large, bold font, with 'In' in yellow and 'novating innovation' in white.

Innovating innovation

BIVEE
Environment
Prototype (v2.0)



EXECUTIVE SUMMARY

The objective of WP6 is to design and build the integrated BIVEE Environment, consisting of the BIVEE Platform prototype - also in the scope of WP6 - and of several BIVEE Application prototypes that are the outcome of WP3, WP4 and WP5. This deliverable is the prototype release of the BIVEE Platform v2.0, and the prototype integration of the Mission Control Room v1.0 (D3.21), of the Virtual Innovation Factory v1.0 (D4.21) and of the Production & Innovation Knowledge Repository (D5.1 + D5.2 + D5.3).

Section 1 gives a functional overview of the BIVEE Environment, and describes the achievements obtained since the previous release of the prototype.

Section 2 deals with the technical architecture of the BIVEE Environment and with integration of BIVEE Applications and of the BIVEE Platform.

Section 3 shows the *big picture* of BIVEE Environment use cases.

Section 4 describes each single module of the BIVEE Platform in detail, from the architectural and technical points of view.

Section 5 is a user guide providing actual screenshots of the BIVEE Platform and, to a minor extent, of the linked BIVEE Applications.

Section 6 lists references to external sources.

Finally, four annexes provide the technical *facts* (links to implementation, to source code, installation instructions, etc.) for the BIVEE Platform modules that are released as part of this deliverable.

DOCUMENT INFORMATION

IST Project Number	285746			Acronym	BIVEE
Full title	Business Innovation and Virtual Enterprise Environments				
Project URL	http://www.bivee.eu				
EU Project officer	Danuta Seredynska				
Deliverable	Number	D6.32	Title	BIVEE Environment prototype (v2.0)	
Work package	Number	WP6	Title	Architecture, IT Platforms federation and services	
Date of delivery	Errore. L'origine riferimento non è stata trovata.				
Status	Version 1.0, dated 30/09/2013			final x	
Nature	Report Demonstrator x Other				
Abstract (for dissemination)	This document provides technical details and user manual for the BIVEE Environment v2.0 prototype				
Keywords	BIVEE, prototype, manual, platform				
Authors (Partner)	Mauro Isaja (ENG), Maria Carla Mantuano (ENG), Fabrizio Smith (CNR-IASI), Nesat Efendioglu (BOC), Francisco Calle Moreno (ATOS)				
Responsible Author	Mauro Isaja		Email	mauro.isaja@eng.it	
	Partner	ENG	Phone	+39 05771887100	

This work is licensed under the Creative Commons Attribution 3.0 License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

This work is partially funded by EU under the grant of 285746.



TABLE OF CONTENTS

EXECUTIVE SUMMARY.....	2
DOCUMENT INFORMATION	3
TABLE OF CONTENTS.....	4
INDEX OF FIGURES.....	6
GLOSSARY OF TERMS	7
1 BIVEE ENVIRONMENT OVERVIEW	8
1.1 General Description	8
1.2 Achievements with respect to the previous release	9
2 BIVEE ENVIRONMENT ARCHITECTURE.....	10
2.1 BIVEE Platform integration	11
2.2 Mission Control Room integration.....	12
2.2.1 Integration between MCR and PIKR	12
2.2.2 Integration between MCR and VIF	15
2.2.3 Integration between MCR, BIVEE Service Platform and Third Parties	15
2.3 Virtual Innovation Factory integration.....	16
2.3.1 PIKR.....	16
2.3.2 MCR	17
2.3.3 BIVEE PLATFORM.....	17
2.4 Production and Innovation Knowledge Repository integration	18
3 BIVEE ENVIRONMENT USE CASES	20
4 BIVEE PLATFORM MODULES	21
4.1 Web Portal	21
4.2 Single-Sign-On	21
4.2.1 Introduction.....	21
4.2.2 Open technologies.....	22
4.2.3 Architecture of the SSO module	24
4.2.4 SSO Service sub-module	24
4.2.5 LSA Service sub-module.....	25
4.2.6 BIVEE standard SSO attributes and roles.....	26
4.3 Virtual Enterprise Management	27
4.4 Raw Data Handler	28
4.4.1 Introduction.....	29
4.4.2 RDH-DS	29
4.4.3 RDH-WS	32
4.4.4 RDH-FE	32

4.4.5 RDH-Sync	32
5 BIVEE PLATFORM USER GUIDE.....	33
5.1 Login to the Web Portal	33
5.2 BIVEE Platform menu	34
5.3 Menu entry: Home	35
5.4 Menu entry: Mission Control Room	36
5.5 Menu entry: Virtual Innovation Factory	37
5.5.1 Sub-menu entry: Dashboard	38
5.5.2 Sub-menu entry: New Idea	38
5.5.3 Sub-menu entry: Posted Ideas	39
5.5.4 Sub-menu entry: Challenges	39
5.5.5 Sub-menu entry: Observatory	40
5.5.6 Sub-menu entry: Store	40
5.5.7 Sub-menu entry: Whiteboards	41
5.5.8 Sub-menu entry: Admin	41
5.6 Menu entry: PIKR	42
5.7 Menu entry: KPI	43
5.8 Menu entry: Admin	45
5.8.1 Sub-menu entry: Virtual Enterprises	45
5.8.2 Sub-menu entry: RDH Synchronization	47
5.9 User and Enterprise management	48
5.10 Add a new Organization	53
5.11 Add a new User	53
6 REFERENCES.....	56
ANNEX A - WEB PORTAL FACTSHEET	57
ANNEX B - SINGLE-SIGN-ON FACTSHEET	59
SSO Service	60
SSO Client	61
ANNEX C - VIRTUAL ENTERPRISE MANAGEMENT FACTSHEET	62
ANNEX D - RAW DATA HANDLER FACTSHEET	70

INDEX OF FIGURES

Figure 1 - BIVEE Environment overview.....	8
Figure 2 - Example deployment of a BIVEE Environment instance.....	10
Figure 3 - BIVEE Platform API overview	11
Figure 4 – Synchronization of PIKR and MCR for Model Annotation.....	12
Figure 5 - Partner Search during VE Creation in MCR Modeller	13
Figure 6 - Document and Knowledge Resource Retrieval in MCR Assistant.....	14
Figure 7 - Synchronization of MCR and PIKR for KPI Visualisation.....	14
Figure 8 - Exposing Conceptual Models to VIF Whiteboard in order to take Comments Suggestions and Annotations (excerpt from [6])	15
Figure 9 - Integrated components from the VIF side	16
Figure 10 - PIKR services consumed by VIF	17
Figure 11 - MCR -VIF whiteboard integration	17
Figure 12 - Bivee Environment use case diagram	20
Figure 13 - CAS authentication process	23
Figure 14 - SSO login sequence	24
Figure 15 - Multiple LSA Services under a single SSO master	26
Figure 16 - BIVEE Environment federation.....	26
Figure 17 - Relationship of the RDH module with BIVEE components	28
Figure 18 - Example of Dimension in KPIOnto	29
Figure 19 - Example of how ontology concepts are mapped to database tables	30
Figure 20 - Example of how ontology concept instances are mapped to database records	30
Figure 21 - Example of DB schema involving two KPI <i>fact tables</i>	31
Figure 22 - DB schema of the RDH-DS Catalogue.....	31
Figure 23 - Web Portal home page	33
Figure 24 - Sign In button	33
Figure 25 - RDH-FE landing page	43
Figure 26 - RDH-FE: KPI data	43
Figure 27 - RDH-FE: KPI value sequence	44
Figure 28 - RDH-FE: insert KPI value.....	44
Figure 29 - VEM-FE landing page	45
Figure 30 - VEM-FE: VE Details dialog window	46
Figure 31 - VEM-FE: new VE dialog window	47
Figure 32 - Go To menu	48
Figure 33 - Control Panel menu	48
Figure 34 - Users and Organizations page.....	49
Figure 35 - Organizations basic search.....	50
Figure 36 - Organizations advanced search	50
Figure 37 - Users basic search	51
Figure 38 - Users advanced search.....	51
Figure 39 - Delete Organization	52
Figure 40 - Deactivate User	52
Figure 41 - Delete User.....	52
Figure 42 - Add Organization.....	53
Figure 43 - Add User	53
Figure 44 - New User: basic data.....	54
Figure 45 - User menu: Roles	54
Figure 46 - User menu: Organizations.....	55
Figure 47 - New User: Organization list.....	55

GLOSSARY OF TERMS

PIKR - Production and Innovation Knowledge Repository: the BIVEE Application which is responsible for maintaining the Knowledge Base of a given BIVEE Environment.

PIKR KB - PIKR Knowledge Base: the reference ontologies of a given BIVEE Environment, each composed by concepts (classes and attributes) and real-world entities (individuals of a given class).

KPI - Key Performance Indicator: numerical indicator used to express a performance measurement that is key to assess some process.

KPIOnto: specific reference ontology which defines KPIs and their related concepts (dimensions / levels) and entities (level members) in the scope of a given BIVEE Environment.

MCR - Mission Control Room: the BIVEE Application

VIF - Virtual Innovation Factory: the BIVEE Application

SSO - Single-Sign-On: general-purpose authentication pattern where a centralized *trusted* service performs user authentication on behalf of one or more *trusting* applications.

LSA - Local Security Authority: specific BIVEE Environment concept denoting any system, internal or external to the Environment, which provides a credential verification service to the centralized SSO system.

RDH - Raw Data Handler: core module of the BIVEE Platform dealing with the collection of KPI data.

RDH-DS - RDH Data Storage: sub-module of RDH in charge of the storage of *normalized* KPI data.

RDH-WS - RDH Web Service: sub-module of RDH exposing a web API for accessing stored KPI data.

RDH-FE - RDH Front-end: sub-module of RDH, integrated into the BIVEE Platform, implementing a web-based UI for navigating and managing stored KPI data.

VE - Virtual Enterprise: temporary and dynamic association of Enterprises for the achievement of common business goals.

VEM - Virtual Enterprise Management: core module of the BIVEE Platform maintaining a registry for Virtual Enterprises and their Enterprise members.

VEM-WS - VEM Web Service: sub-module of VEM exposing a web API for accessing the VE registry.

VEM-FE - VEM Front-end: sub-module of VEM, integrated into the BIVEE Platform, implementing a web-based UI for navigating and managing the VE registry.

1 BIVEE ENVIRONMENT OVERVIEW

1.1 General Description

The BIVEE Environment prototype is the first integrated release of the BIVEE Platform which includes the complete set of BIVEE Applications. More specifically, the following elements were integrated:

- BIVEE Platform v2.0: D6.32 - M24 issue (*this deliverable*)
- Mission Control Room (MCR) v1.0: D3.21 - M22 issue
- Virtual Innovation Factory (VIF) v1.0: D4.21 - M22 issue
- Production and Innovation Knowledge Repository (PIKR) v1.0: D5.1 - M8 issue; including:
 - Advanced Data Adaptors: D5.2 - M21 issue
 - Advanced Interoperability Services: D5.3 - M21 issue

The BIVEE Platform plays the role of integration enabler for BIVEE applications and services in several ways:

- Web portal and single-sign-on system
- Common back-end services (VEM-WS and RDH-WS)
- Unified storage for *raw* performance indicator data (RDH / Data Storage)

Additionally, the BIVEE Platform exposes its own front-end for data and environment administration.

Figure 1 shows how all the above mentioned elements are related. See also section 2 for an architectural view of the BIVEE Environment.

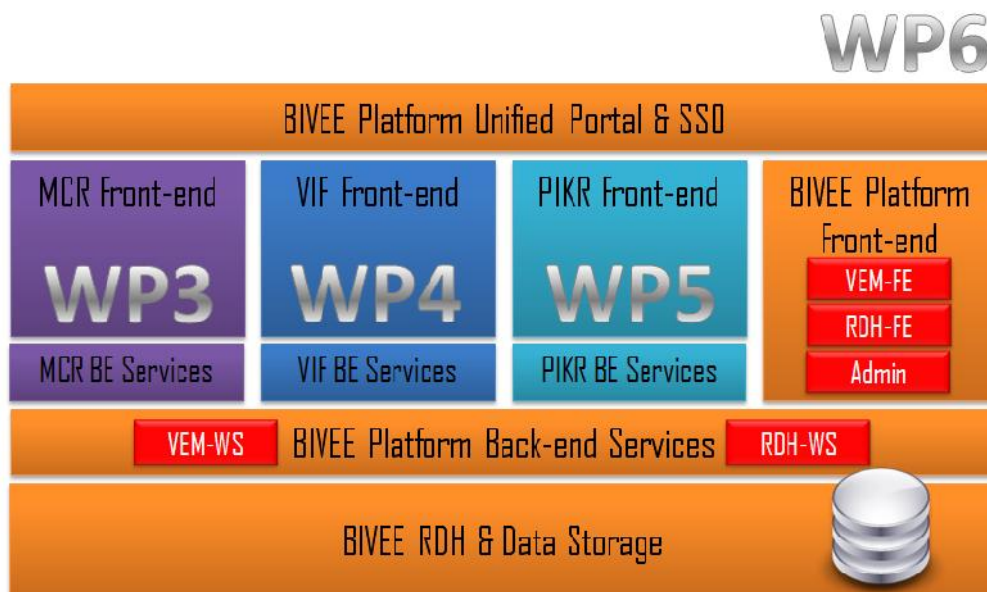


Figure 1 - BIVEE Environment overview

1.2 Achievements with respect to the previous release

This deliverable builds upon the M12 issue (D6.31), which released the first version of the BIVEE Platform with no integrated BIVEE Environment - at that time, the only BIVEE Application released being PIKR (D5.1 - M8).

In order to include MCR, VIF and PIKR, the Liferay web portal on which the BIVEE Platform is based (see section 4.1) was extended with a Single-Sign-On (SSO) implementation, providing the central authentication and attribute release services required in order to run several distinct web applications together in a seamless way. The SSO service was developed from a de-facto standard open source implementation (Jasig CAS), but innovative BIVEE functionality was also added that enable the federation of multiple BIVEE Environment instances into a single business ecosystem, without the need for a centralized user administration - see section 4.2 for details.

BIVEE Applications that were released as standalone applications by M22 - MCR and VIF - were embedded into the web portal. Their web front-ends were decomposed into functional modules which can be navigated through the common BIVEE Environment menu. Moreover, they were integrated with the SSO service. Finally, a first step towards look & feel harmonization was taken, by applying some common BIVEE style elements as defined in the portal stylesheet.

The PIKR web front-end, due to its nature of separate administration tool, is linked from the BIVEE Environment menu and is integrated with SSO, but still remains a standalone application.

A new component was added to the BIVEE Platform: the Virtual Enterprise Management (VEM) - see section 4.3. This component enables Virtual Enterprise (VE) management at the base BIVEE Platform level, and is also integrated with the SSO service, such that VE membership is *attached* as an attribute to authenticated users. The VEM comes with its own web UI, but also exposes a web API for BIVEE Applications to use (VEM-WS).

The existing Raw Data Handler (RDH) component, in charge of acquiring performance indicator data and of navigating the repository, was entirely redesigned in the scope of WP5 in order to support the D5.3 deliverable "Advanced Interoperability Services", released by M21; the new design was then implemented in the scope of this deliverable. More specifically, the Data Storage (RDH-DS) was re-engineered and the RDH-WS web API was extensively modified, enabling the integration between the end-user IT systems and the BIVEE semantic knowledge base that was the objective of D5.3. On top of that, an enhanced web UI for data navigation and administration was also developed.

2 BIVEE ENVIRONMENT ARCHITECTURE

The BIVEE Environment is an inherently distributed architecture: the BIVEE Platform is a web-based portal, assembling together an array of web-based BIVEE Applications which leverage different technologies and may run on different nodes of the hosting network.

The unifying elements are a centralized or federated (see section 4.2.3) authentication and attribute release system, a standard user profiling schema, a portal-like container, a set of Platform-level services and, to a lesser extent, a common look & feel across user interfaces.

Given this fabric, and the *stateless* nature of all back-end services in the BIVEE Environment - both Platform-level and Application-level - cloud deployment is a viable option: with a minimum of extra effort, any Application which is part of the Environment, as well as the Platform itself, can be virtualized and installed on high-availability clusters.

The following deployment diagram illustrates this design with an example where each subsystem runs on its own node. More complex, cloud-oriented setups are possible, with the introduction of system-level facilities like load-balancers and database clusters; however, these scenarios are out of the scope of this document.

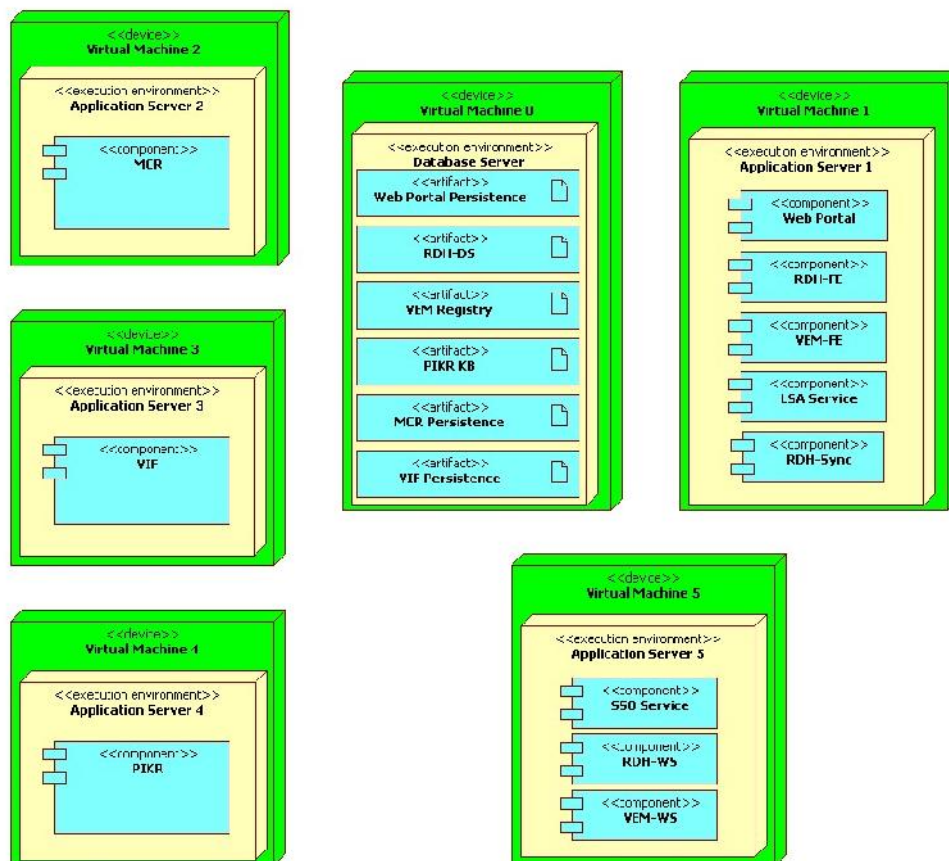


Figure 2 - Example deployment of a BIVEE Environment instance

In a distributed architecture such as the BIVEE Environment, where several different BIVEE Applications need to cooperate in a seamless way, integration issues are of paramount importance. The following subsections give an overview of how this objective was achieved, with a focus on cross-application communications.

2.1 BIVEE Platform integration

The BIVEE Platform exposes a set of APIs for BIVEE Applications to consume, and in turn consumes some APIs exposed by BIVEE Applications. The following component diagram identifies these endpoints.

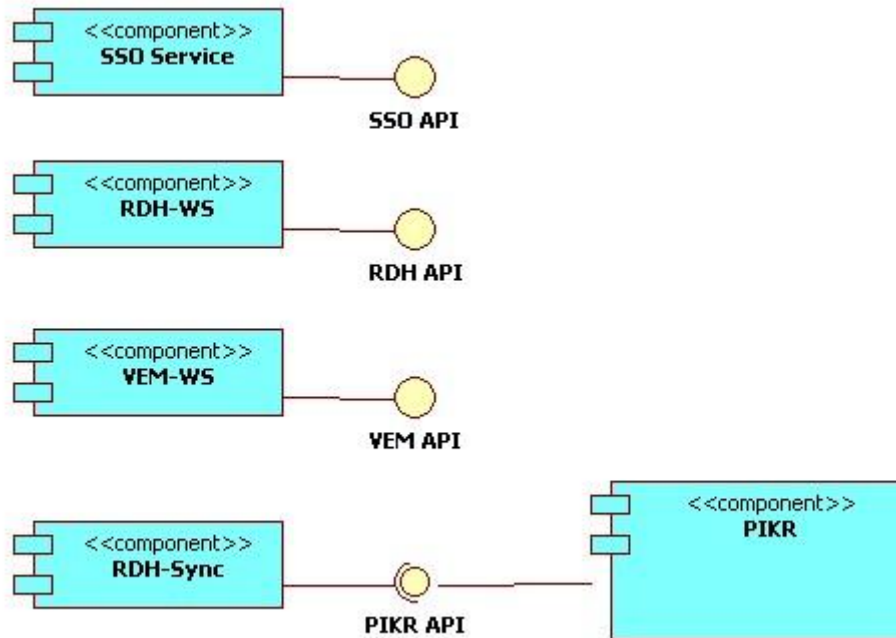


Figure 3 - BIVEE Platform API overview

The SSO API provides means for BIVEE Applications to verify user authentication tokens and to obtain relevant information about the current web user (user attributes). See section 4.2.4 for more details, and also section 4.2.6 for information about user attributes.

The RDH API enables BIVEE Applications to read and modify KPI data stored in RDH-WS. See section 4.4.3 for more details.

The VEM API enables BIVEE Applications to read and modify Virtual Enterprise data stored in VEM. See section 4.3 for more details.

Finally, the RDH-Sync sub-module depends on the PIKR API for reading the KPIOnto ontology. See the PIKR documentation[8] for more details.

2.2 Mission Control Room integration

The first prototype of MCR provides a set of web methods in the Component API and Semantic Modelling Kernel (SMK) in order to enable interactions of the other two BIVEE macro blocks ("VIF" and "PIKR"), the BIVEE Platform itself and the third parties with the MCR (see details in [6]).

Furthermore, during the ongoing implementation of the second prototype of MCR, within the task T3.2, integration scenarios in following three fields have been concretized in order to realize interaction with BIVEE macro blocks, BIVEE Platform and third parties;

- Integration between MCR and PIKR
- Integration between MCR and VIF
- Integration between MCR, BIVEE Platform and third parties

2.2.1 Integration between MCR and PIKR

Integration work between MCR and PIKR concentrates on following scenarios:

- Synchronization of the PIKR and the MCR for Model Annotation
- Partner Search during VE Creation in the MCR Modeller
- Document and Knowledge Resource Retrieval in the MCR Assistant
- Synchronization of the MCR and the PIKR for the KPI Visualisation

Synchronization of the PIKR and the MCR for Model Annotation

The idea in this scenario is based on the semantic lifting approach as described in [11] and [12] and its scenarios that have been detailed in [6]. The scenario aims to synchronize the domain ontologies as concept lists within the MCR environment for annotation of modelling constructs using domain concepts.

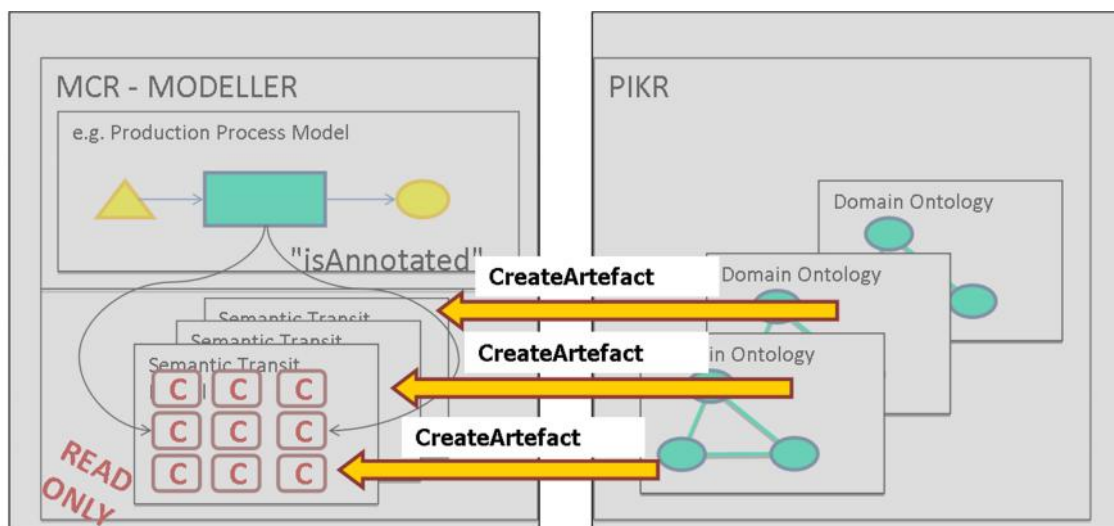


Figure 4 – Synchronization of PIKR and MCR for Model Annotation

In order to realize the idea, in ongoing implementation of MCR prototype V2.0 within the T3.2 a functionality has been developed, which runs every 24 hours (as a cron job) to compare and synchronize the bases and automatically creates a semantic transit model for each ontology. The semantic transit model includes related domain concepts in form of concepts (a modelling

construct specified for semantic transit model, see details in [11]). As depicted in Figure 4 the cron job basically utilize the services from PIKR in order to look up the domain ontologies and applies web method “createArtefact” from the MCR Component API in order to create semantic transit models with related concepts in model repository within the MCR.

Partner Search during VE Creation in the MCR Modeller

The idea is to use PIKR during VE design/creation to find and select appropriate partners for the definition of the VE for certain process/activities/product features based on the annotation with the concepts imported from PIKR as described in previous scenario.

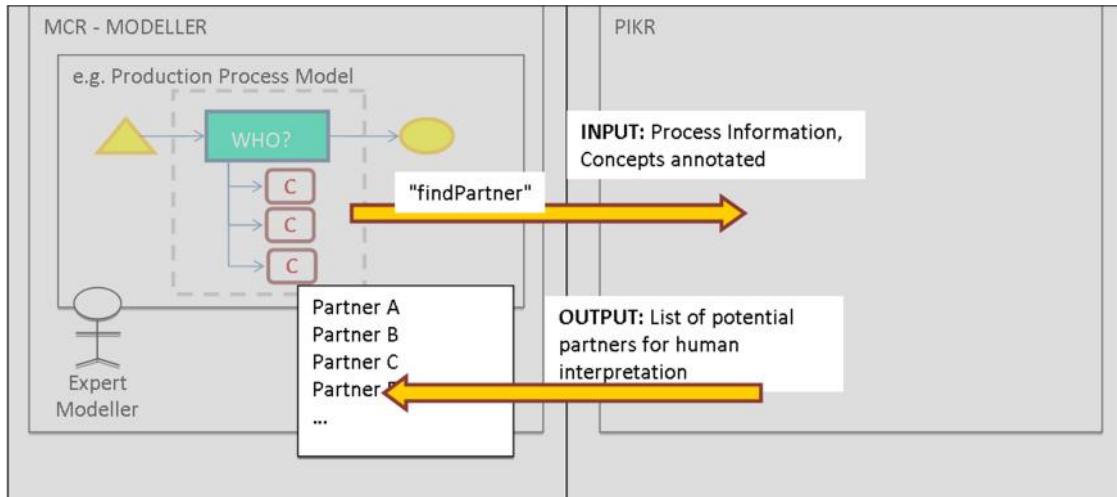


Figure 5 - Partner Search during VE Creation in MCR Modeller

For the realization of the proposed scenario, in the ongoing implementation of second prototype of MCR a specific method has been implemented which, as depicted in Figure 5, sends information about process/activity/product feature and annotated concepts to the PIKR service and displays returning proposed partners in a list within the MCR Modeller and allows users to select one partner from the list. Although the first interaction has been established in the context of this scenario, the scenario is still under development within the T3.2.

Document and Knowledge Resource Retrieval in the MCR Assistant

This scenario is set around the idea to make related documents/knowledge resources available to the user of the MCR Assistant based on the annotation of the flow object by using concepts imported from the PIKR. The documents/knowledge resources are not hard-linked but derived according to the annotation.

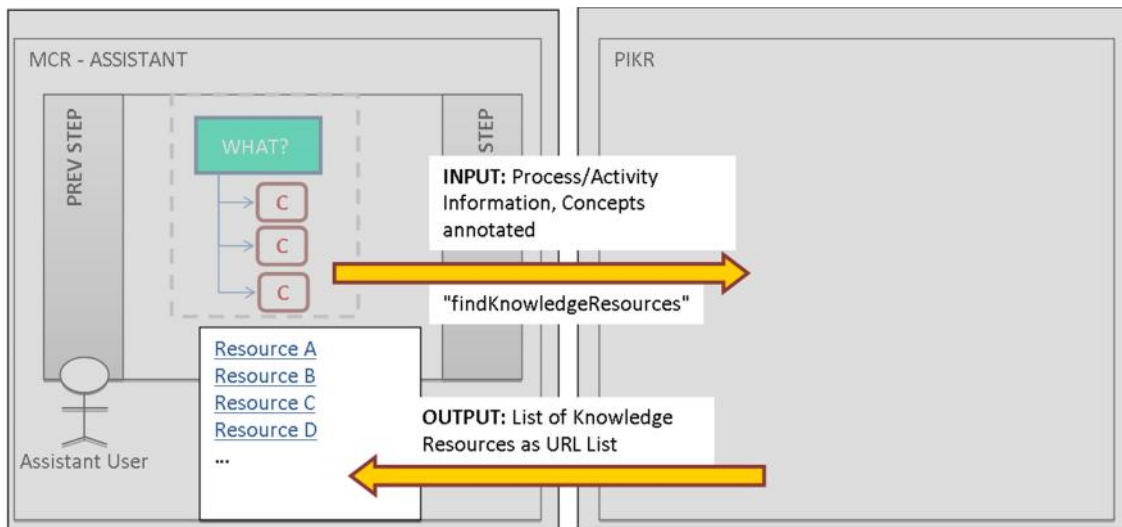


Figure 6 - Document and Knowledge Resource Retrieval in MCR Assistant

In order to realize the idea, in ongoing implementation of MCR prototype V2.0 within the Task T3.2 an appropriate approach has been specified which is currently being implemented.

According to approach, as depicted in Figure 6, the MCR sends information about process/activity and annotated concepts to PIKR service and receives related list of URLs of related documents/knowledge resources and displays them in the MCR Assistant.

Synchronization of the MCR and the PIKR for KPI Visualisation

The idea in this scenario is to feed the sensors related with KPIs in the MCR by data retrieved from PIKR. The MCR controls the KPI update and calculation frequency and then visualizes the KPI data on MCR Monitor according to predetermined parameters.

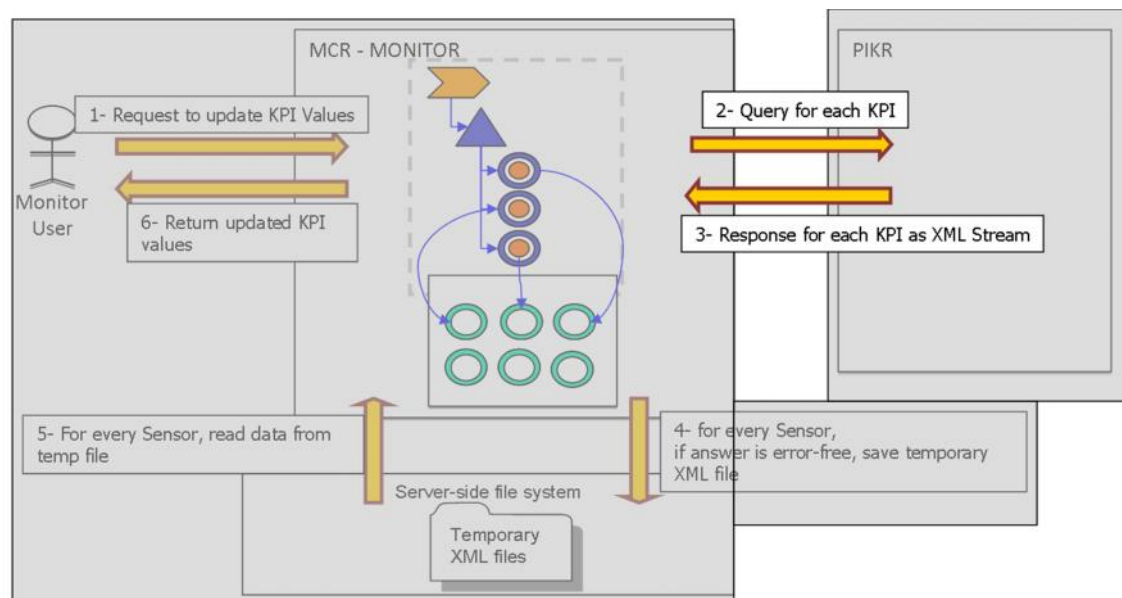


Figure 7 - Synchronization of MCR and PIKR for KPI Visualisation

In order to realize the idea, in addition to on going implementation of MCR prototype V2.0 within the Task T3.2, a method "updateKPI" has been implemented which serializes a query for each KPI in question to XML complying XML schema provided by the PIKR and saves the returning XML stream in XML file for further processing (as depicted in Figure 7).

2.2.2 Integration between MCR and VIF

Effort to establish integration between MCR and VIF is focused on the scenario defined in [6]. The idea is to expose conceptual models modelled with the MCR Modeller to the Whiteboard in VIF, open them for a discussion and get them back enriched with comments in order to improve VE.

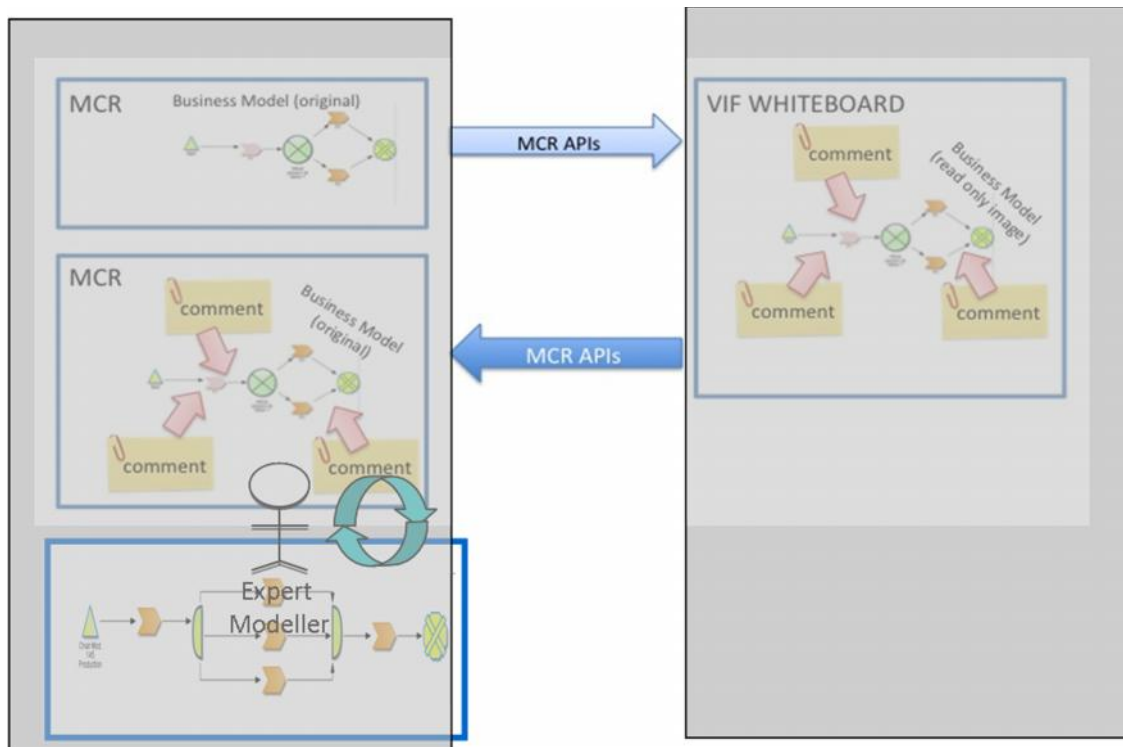


Figure 8 - Exposing Conceptual Models to VIF Whiteboard in order to take Comments Suggestions and Annotations (excerpt from [6])

In order to realize the idea, the approach specified in [6] has been taken as basis. In order to realize that approach the web methods GetTOC, GetModelNameById, GetModelIdByNameAndType, GetModelAsXML, GetModelImage are provided within the MCR Component API and they are currently in being tested in the WP4.

2.2.3 Integration between MCR, BIVEE Service Platform and Third Parties

Effort to establish integration between MCR, BIVEE Platform and third parties gathers around three subjects; (1) Implementation of web methods in the MCR Component API in order to allow interactions with MCR, (2) Single Sign On (SSO) integration and (3) look & feel.

Currently 56 web methods in the MCR Component API have been implemented. The SSO integration using CAS and style adaptations in order to have common look & feel are under development.

2.3 Virtual Innovation Factory integration

The Virtual Innovation Factory (VIF) is one of the components integrated within the BIVEE environment. This is done through the use of web services (consumed / exposed). The integrated components are

- PIKR
- MCR
- BIVEE platform services

This is the main picture of the integrated components:

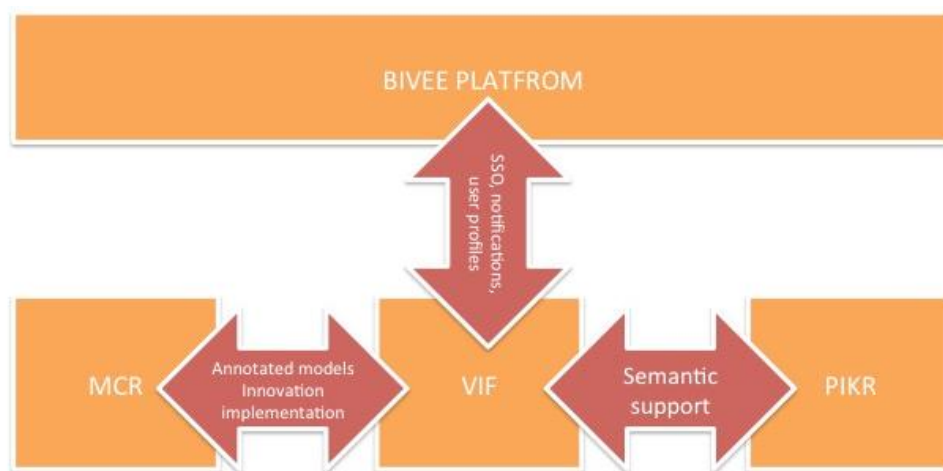


Figure 9 - Integrated components from the VIF side

The VIF also partially exposes its own database information using a CRUD system based on REST web services. There's also read-only specific services that are expose; those one needed to obtain the information in the KPI calculation processes.

2.3.1 PIKR

PIKR offer the VIF semantic support through SOAP based web services. While SOAP it is heavily used in other technological frameworks, it is of no common use in javascript environments. Thus, the VIF has implemented a component to enable such kind of connections.

Consumed services by the VIF from are:

- Element tags (Concepts from the PIKR)
- Document annotation
- Semantic search

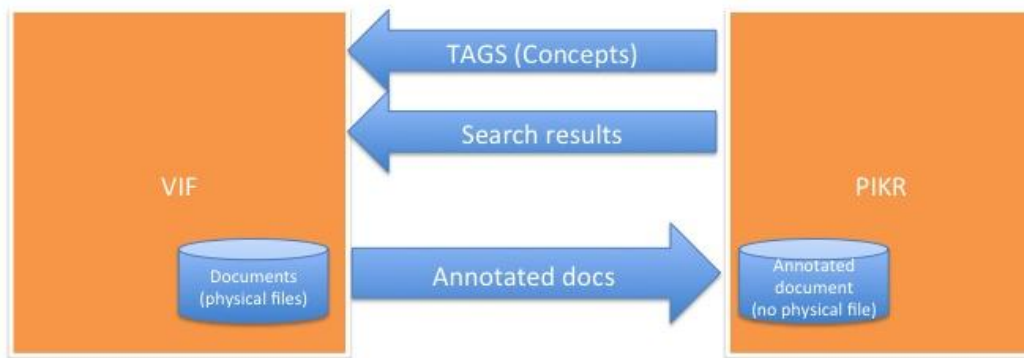


Figure 10 - PIKR services consumed by VIF

2.3.2 MCR

There are two kinds of integrations between MCR and VIF. First one, the already explained in the MCR section 2.2.2. between the MCR and the VIF whiteboard. This integration allows the models to be annotated – not semantically but with notes. This integration is started from the MCR and ends in it.

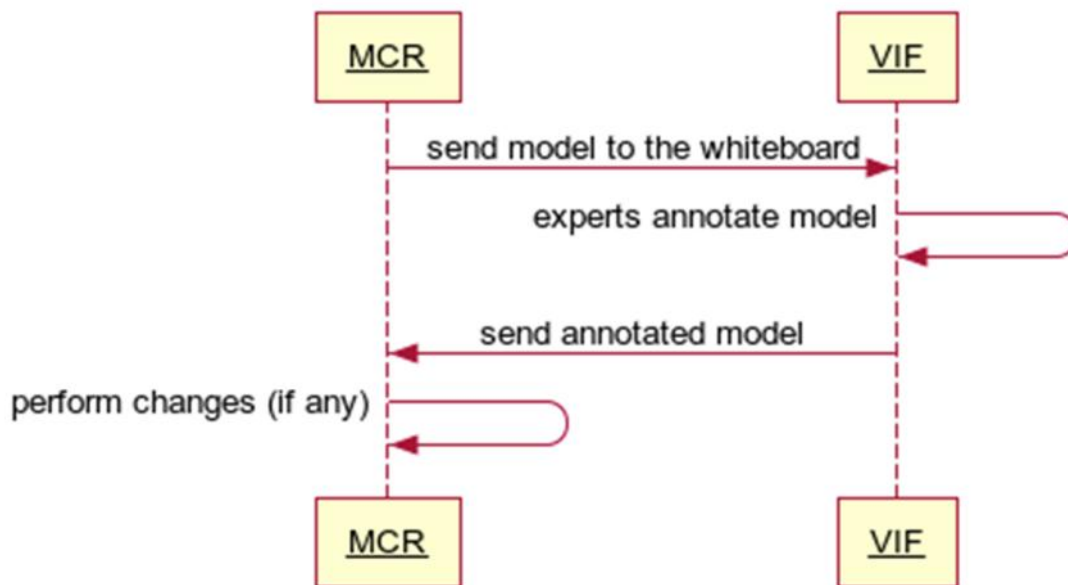


Figure 11 - MCR -VIF whiteboard integration

The second one – yet to be implemented – will happen when the Innovation Project is finished. This will be a web service based communication protocol allowing the Production Space to implement the outcomes of a concrete Innovation Project.

2.3.3 BIVEE PLATFORM

The VIF uses these services from the BIVEE platform:

- SSO
- User profiles (next release)
- Notifications (next release)

The SSO is performed using an specific module developed as part of the VIF. Again, as it happens with the SOAP based services, since they are not of common use, the CAS is not heavily expanded within the javascript community.

User profiles are to be extended by each component – VIF, MCR and PIKR – but the user and its general profile are common to each component. Therefore, the VIF will extend the profile of each user to meet specific needs such as:

- Roles and permissions within the VIF
- Detailed skills
- Etc.

2.4 Production and Innovation Knowledge Repository integration

The PIKR implements a collection of interfaces, jointly referred to as *PIKR Connector* (see Deliverable D5.2), realizing a bridge on the ontology-based structures defined in the PIKR and the data structures adopted by the other BIVEE components. In the application scenarios considered so far, the PIKR acts mainly as a back-end module, providing services to the BIVEE front-end realized by the VIF and MCR. Please refer to Deliverable D5.2 (Chapter 5 and 6) for a discussion regarding the approach followed for the ontology-based lifting of the informative resources pertaining to both the Business Innovation Space and the Value Production Space. The main API implemented by the Connector can be grouped as follows:

- *Accessing to the Domain Specific Ontologies*, in order to retrieve domain *concepts* and related information to be used for the annotation/tagging of resources.
- *Management of the Semantic Descriptors*, in order to operate on the PIKR Knowledge Base for storing/modifying/retrieving the semantic representation of informative resources; e.g., models created in the MCR, documents edited in the VIF.
- *Exporting fragments of the PIKR Knowledge base*, by a suitable conversion or transformation; e.g., retrieve the form for describing a given document encoded according to a data structure processable by the VIF, the list of the dimensions to be provided in order to compute a given KPI.

In addition, the reasoning facilities provided by the *PIKR Service Library* (see Deliverable D5.1) are made available through the interface exposed by the *Service Manager*. The main API that can be accessed by the VIF and the MCR through the Service Manager interface are:

- *Semantic search*, which allows exploiting semantics-enabled search services. The *search_request* is an expression containing an ontology-based feature vector describing the criteria for the selection of the resources of interest. The function returns a list of ranked results (in terms of resource URIs) by applying semantic similarity techniques.
- *Query evaluation*, which allows posing queries to the PIKR knowledge base, compiling queries into the underlining inference engine, and collecting the result of the evaluation. Besides the query service described in D5.1, a pure SPARQL end-point has been deployed.
- *Compliance verification*, which allows verifying the consistency between the semantic descriptors and the business rules both formalized and stored in the PIKR.
- *KPI definition management*, which supports the management of KPIOnto by allowing the introduction of new KPIs and their mathematical formulas, their update and their deletion.

These functions are capable to check if the new KPI is consistent with respect to all the other formulas, to avoid incoherencies and redundancies.

- *KPI reasoning for VE setup*, which allows for the extraction of those indicators that are in common among a set of enterprises, and that can be used to monitor and compare their performances. Such functionalities are useful during the setup phase, to determine the sets of information that have to be provided by enterprises in order to calculate the requested KPI or, on the other hand, to recognize which kind of indicators can be derived, given the available information at enterprise level.

Finally, the Semantic Data Handler (SDH) is the module of the PIKR that allows the other BIVEE components to access KPI data by providing a uniform view over them, independently of their original data sources (see Deliverable D5.3).

The main functionality is *Multidimensional Query*, which accepts a request for the computation of a certain KPI with respect to levels of some dimensions (e.g., ROI for the enterprise ACME, evaluated for the first semester of 2013, for the countries Italy and Spain). Terms are expressed through the VE common reference, that is the KPIOnto. The approach, described in details in D5.3, exploits reasoning over the KPIOnto in order to determine if the KPI specified in the request can be actually computed starting from the data in the RDH. If so, the request is translated into a Data Storage compliant query, and then passed to the RDH for retrieving the actual data. Otherwise, it is capable to rewrite the request by (1) performing drill-down of some dimensions, in order to obtain the required information by aggregation of lower level data (e.g., aggregation of KPI data about months January 2013,..., June 2013 to evaluate KPI data about the first semester 2013), or (2) by decomposing the KPI formula in its components (e.g., $A=B+C$), computing the required information for each of the component (e.g., B and C separately), and then evaluating the required value through the formula(e.g., A).

3 BIVEE ENVIRONMENT USE CASES

Figure 12 is a use case diagram showing the *pig picture* of the BIVEE Environment from the functional point of view. Only those use cases which are directly supported by the BIVEE Platform are described in detail in the User Guide (see section 5). For information on the others, which are related with specific BIVEE Applications, see the relevant documentation released as part of other BIVEE deliverables ([4], [6], [7]).

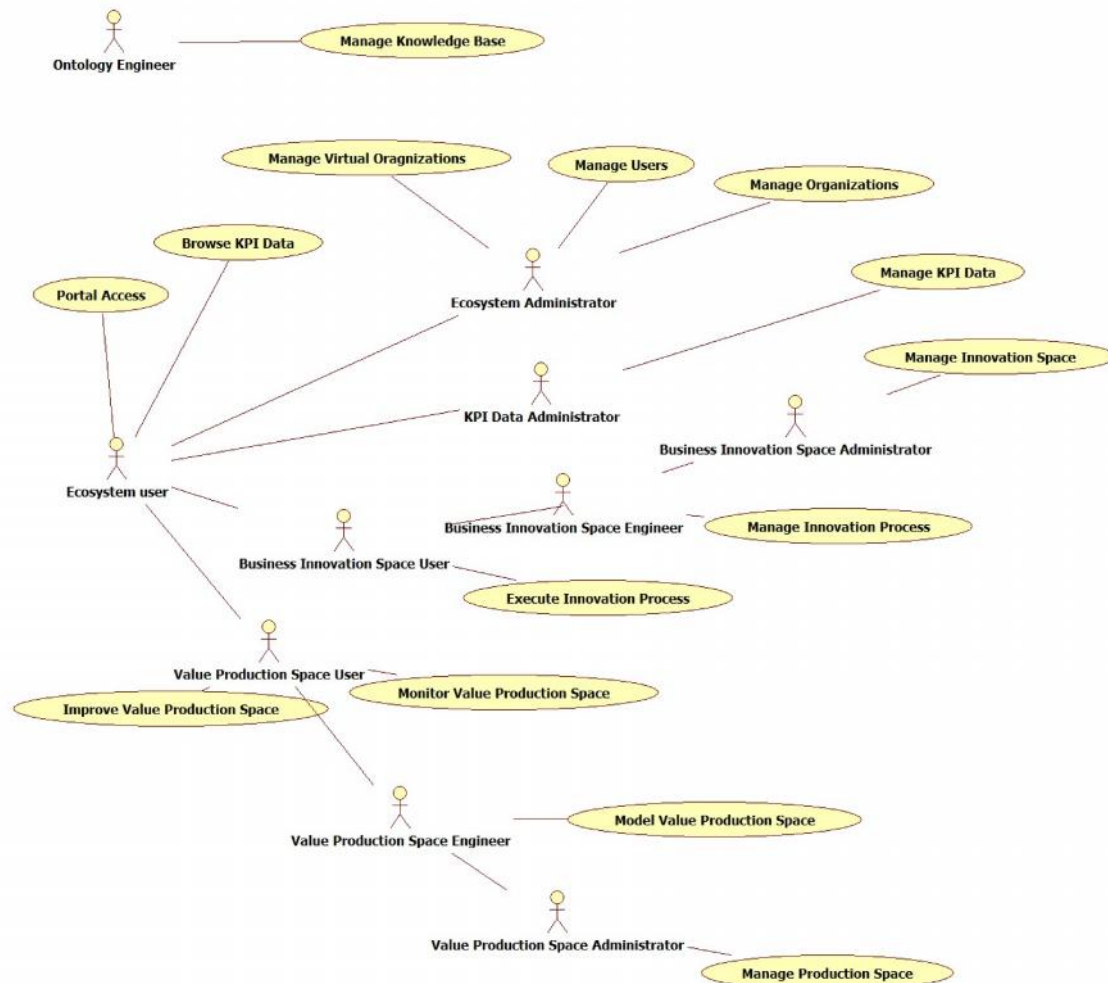


Figure 12 - Bivee Environment use case diagram

Regarding the diagram in Figure 12, it is worth noting that the Actor names and their relationships represent a *logical* responsibility network, and do not necessarily correspond to the physical user roles assigned by the BIVEE Platform's Single-Sign-On system - see section 4.2.6. In the current version of the BIVEE Environment, roles that are application-specific are still managed internally by the applications themselves - e.g., Value Production Space User, Value Production Space Engineer and Value Production Space Administrator are all roles belonging to the Mission Control Room application, which is in charge of mapping them to the users as appropriate.

4 BIVEE PLATFORM MODULES

4.1 Web Portal

The BIVEE Platform web portal module is based on the Liferay Community Edition (CE) product. Liferay CE is a JSR-168 compliant web portal, working on top of the Apache Tomcat web application server. Both Liferay CE and Apache Tomcat are free, open-source and Java-based software, and can run on any computing platform which supports the Java Runtime Environment.

In the BIVEE Platform, the base Liferay implementation is enriched with custom extensions. The most notable example of such extensions is the Liferay-based implementation of Single-Sign-On (see section 4.2), which enables BIVEE Applications to integrate with Liferay's authentication / authorization system. In particular, the BIVEE Platform inherits from Liferay the concepts of "user", "organization" and "role", and the functionalities for their management. In the BIVEE context, the mapping of users and roles is straightforward while organizations are mapped to Enterprises in the BIVEE business ecosystem.

BIVEE web applications that are compatible with the JSR-168 standard are directly integrated as *portlets* in the web portal. This is the case of the VEM and of the RDH front-end UIs.

Technical details on this module are given in Annex A - Web Portal Factsheet.

4.2 Single-Sign-On

4.2.1 Introduction

Generally speaking, Single-Sign-On (SSO) provides a central authentication point for users of complex, distributed, multi-tier web applications and services: users authenticate themselves once and then are granted access to the entire environment, without repeating annoying login operations. Furthermore, the overall security level of the environment is enhanced with respect to application-based sign-on, as user credentials remain invisible to applications.

There are two basic approaches at implementing SSO:

- Centralized – All users, and their credentials, are registered in a single repository. This approach is most effective when used to centralize the management of security policies, e.g., within a single company. Each organization tends to keep an internal centralized repository of users and groups, often provided by a LDAP¹ server, in order to control access to corporate resources.
- Federative – Users and credentials are stored in local systems, or *nodes*, which are grouped in a federation. Accessing one of the federated nodes grants also access to all the others. This approach is particularly useful when different organizations want to keep control over their own security policies.

In the BIVEE context, both approaches are feasible with different configurations of the SSO module - see section 4.2.5.

¹ Lightweight Directory Access Protocol

4.2.2 Open technologies

Several open technologies and standards have been developed in the last years to address authentication and authorization issues in distributed systems. Here we present two of them, Central Authentication Service (CAS) and Security Assertion Markup Language (SAML), as they are the foundation on which the BIVEE SSO solution is built.

Central Authentication Service

Central Authentication Service (CAS) is an open source authentication system, developed by Yale University (Jasig)², which manages all authentication requests by means of a centralized server.

This solution is characterized by the following features:

- Security
 - Passwords are never transmitted to applications
 - All *tickets* exchanged are encrypted
- Simple integration and portability
 - Several clients are available for free, including libraries for Java, Microsoft .Net, PHP, Perl, Apache, Liferay
- Open source codebase

The CAS authentication protocol is illustrated in Figure 13 and is briefly described here:

- When the user first reaches a web application, she is redirected to the CAS login page on a secure connection, where she enters her username and password.
- When user credentials are submitted, CAS verifies them:
 - If authentication fails, the user receives an error page.
 - If authentication is successful, the user is redirected again to the originating web application, this time carrying an authentication token; from this point on, CAS maintains an active user session to “remember” that this user has been authenticated.
- The web application receives again the same user request, but this time with an authentication token attached: before granting access, it validates the token with the CAS service:
 - If validation fails (the problem might be a session timeout, or a faked token), the user is redirected to the CAS login page.
 - If validation is successful, the user finally receives the requested resource.
- When a *different* web application is accessed by the same user, the handshake sequence is started again, but this time when the user reaches the CAS server she is recognized as authenticated (she owns an active session on CAS), so she does not go through the login form and is immediately directed back with a new authentication token. From the user point of view, this second handshake is totally transparent.

² <http://www.jasig.org/cas>

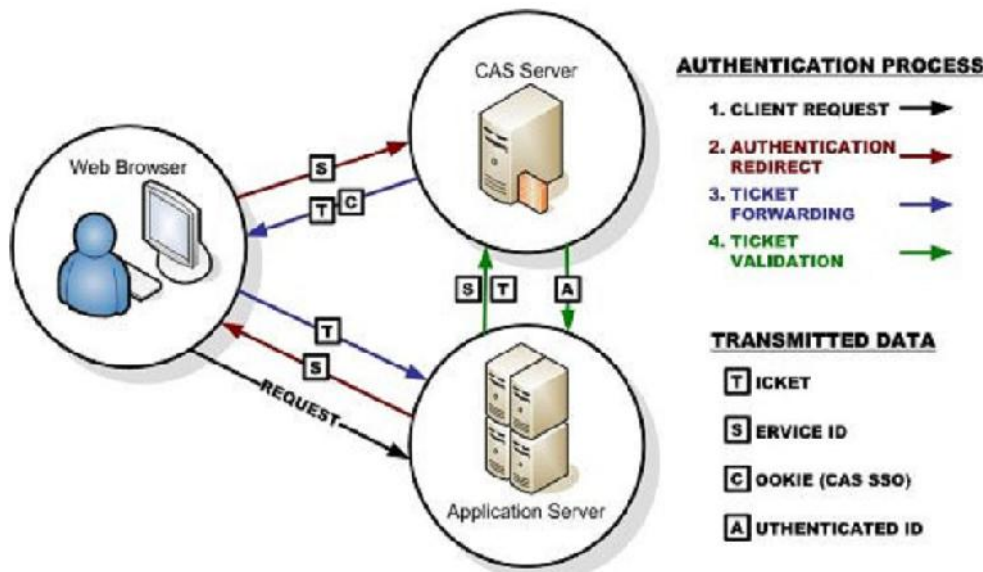


Figure 13 - CAS authentication process

Security Assertion Markup Language

Security Assertion Markup Language (SAML)³ is an open standard based on XML for the exchange of information between different entities during authentication and authorization operations. It was developed by SSTC (Security Services Technical Committee), which belongs to OASIS (Organization for the Advancement of Structured Information Standards). The purpose of SAML is to provide a standard XML syntax to represent security assertions made by an IT system. The SAML standard allows interoperability between different infrastructures / software / business partners, enabling the creation of an application platform based on a single shared authentication process.

SAML supports the interaction between the following entities:

- **Principal**: generic person who actively participates in the process of authentication or authorization, uniquely identifiable by the system using a profile that collects information about her.
- **Identity Provider**: also called SAML Authority or Asserting Party, keeps track of the credentials of the Principal and provides the associated information. In the most general sense, a SAML Authority is considered as an entity capable of responding to requests made through the SAML protocol.
- **Service Provider**: also called Relying Party, represents the entity that receives the information on a Principal released by the Identity Provider. SAML has mechanisms that allow the service provider to check the validity of this information.

SAML is based on the concept of “assertions”. An assertion is the reply to a question made by a Service Provider to a SAML Authority. The Service Provider requests and obtains an identity

³ SAML: <http://saml.xml.org/>

assertion from the Identity Provider. On the basis of this assertion, the Service Provider can make an access control decision - in other words it can decide whether or not to service a request from the connected Principal. Before delivering the identity assertion to the Service Provider, the Identity Provider may in turn request some information from the Principal - such as a user name and a password - in order to authenticate its identity. In SAML, one Identity Provider may provide SAML assertions to many Service Providers. Conversely, one Service Providers may rely on and trust assertions from many independent Identity Providers.

4.2.3 Architecture of the SSO module

The BIVEE SSO module is designed on top of CAS and SAML technologies.

BIVEE SSO introduces some new extension elements in the standard CAS architecture:

- The notion of VE and Enterprise as properties of the authenticated user, which in the BIVEE context implies the integration with the VEM component - see section 4.3.
- An attribute release feature: a SAML-based way of passing additional user information to SSO clients.
- A BIVEE model for "standard" user attributes and roles - see section 4.2.6.

This architecture is implemented in two sub-modules - the SSO Service and the LSA Service - which are described below from the functional point of view. See Annex B - Single-Sign-On Factsheet for detailed technical information.

4.2.4 SSO Service sub-module

The SSO Service is a CAS instance, customized to delegate user authentication and attribute release to a Local Security Authority (LSA) Service chosen from a registry. The target LSA Service is identified from the credentials presented by users, as demonstrated by the following sequence diagram.

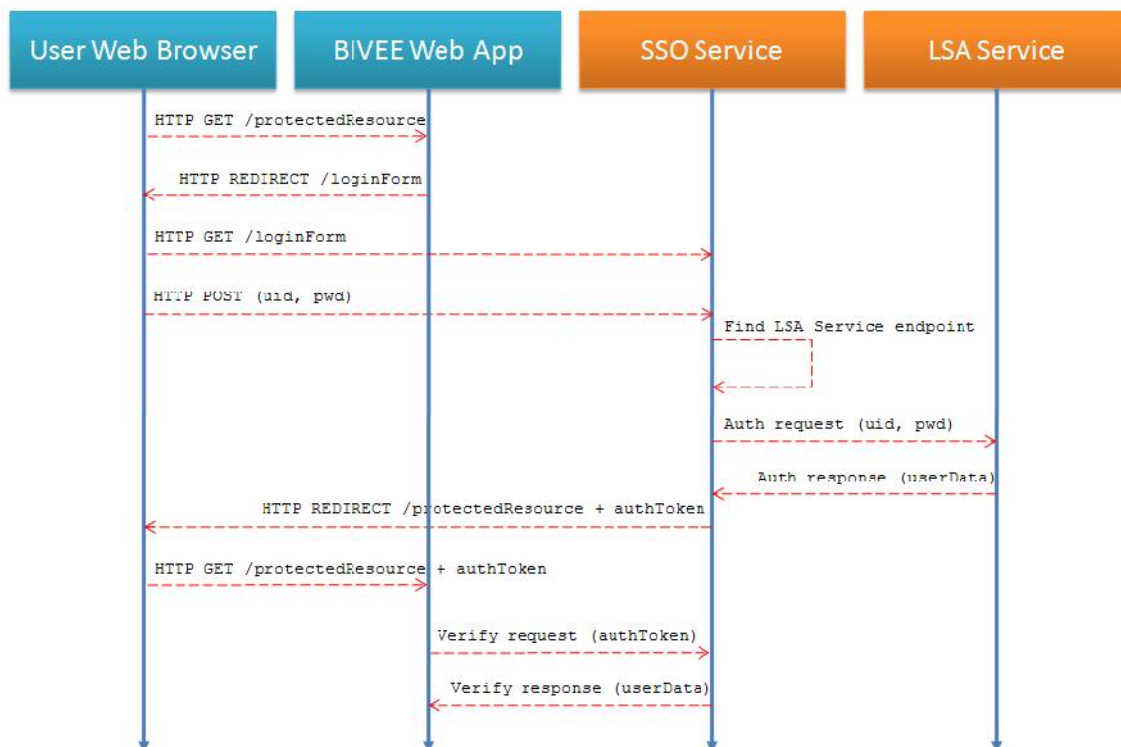


Figure 14 - SSO login sequence

4.2.5 LSA Service sub-module

All users in a BIVEE Ecosystem are registered with some Local Security Authority (LSA), which is an abstraction for a user repository exposed as a service.

The LSA Service sub-module is a two-faced object. On one side it exposes a public endpoint for user credential validation and user attribute release; on the other, it connects with some pluggable user data repository.

- The public endpoint is a RESTful service that accepts HTTP GET requests, issued on a secure channel, with username and password parameters; the response body is a SAML document stating the authenticated identity of the user and a set of standard attributes - see section 4.2.6 .
- The *user repository connector* is a plug-in module chosen from a library of off-the-shelf (e.g., for LDAP or JDBC-based integration) or custom-built adapters. As the LSA Service sub-module is binary compatible with the AuthenticationManager architecture of CAS, the entire library of connectors from the CAS standard distribution is available for reuse.

In order to retrieve Virtual Enterprise membership information and populate the attribute set of an authenticated user, the LSA Service also interacts with Virtual Enterprise Management - see section 4.3.

Typically, there is one single LSA Service instance in place for each BIVEE Platform installation. In this very common configuration, the LSA Service is a simple wrapper around Liferay's built-in user repository: Users, Enterprises and Roles are those defined in Liferay⁴, and the LSA Service added value comes from its capability of merging these data with information from the Virtual Enterprise Management registry, assigning Users to Virtual Enterprises as appropriate⁵. Regardless of this enhancement, the context is still that of the traditional, *centralized* approach to SSO - as described in the introduction.

However, the SSO module's distributed architecture also enables the *federative* paradigm. Having multiple LSA Service instances governed by one single SSO Service means that you can configure:

- one BIVEE Environment with user data stored and managed by several different authorities - like corporate users in LDAP or ActiveDirectory services - see Figure 15;
- multiple federated BIVEE Environments, each running its own Liferay web portal;
- a mix of the above - see Figure 16.

This wide range of possible SSO configurations reflects on the flexibility of the BIVEE Environment itself, which can be adapted to complex setups involving a variety of businesses and of business ecosystems.

⁴ Enterprise is the BIVEE equivalent of Organization in Liferay.

⁵ A User is *virtually* member to a given VE if at least one of the Enterprises she belongs to (as defined by Liferay's mappings) is actually member of that VE (as per VER's mappings).

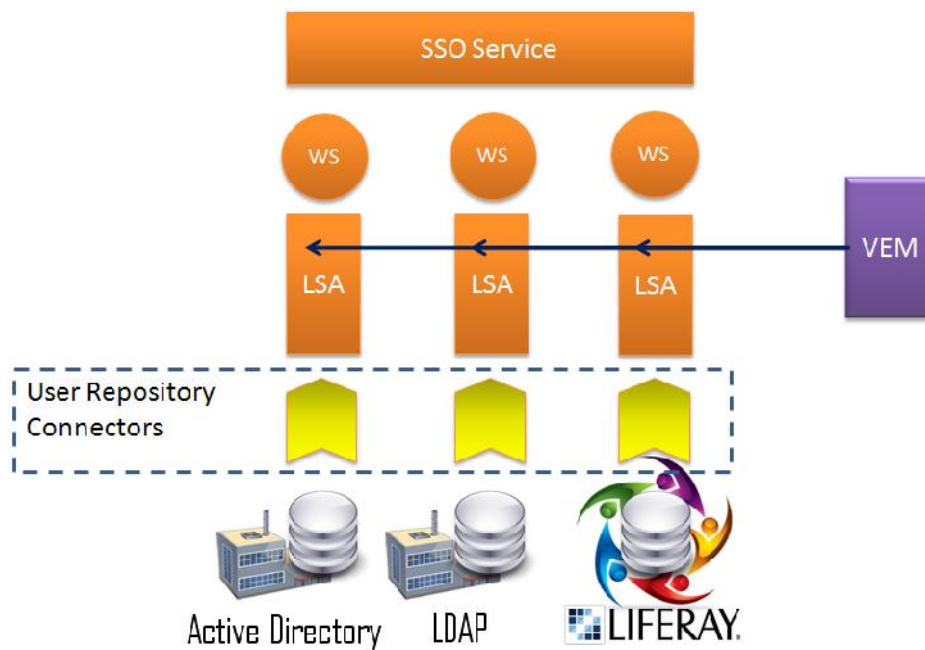


Figure 15 - Multiple LSA Services under a single SSO master

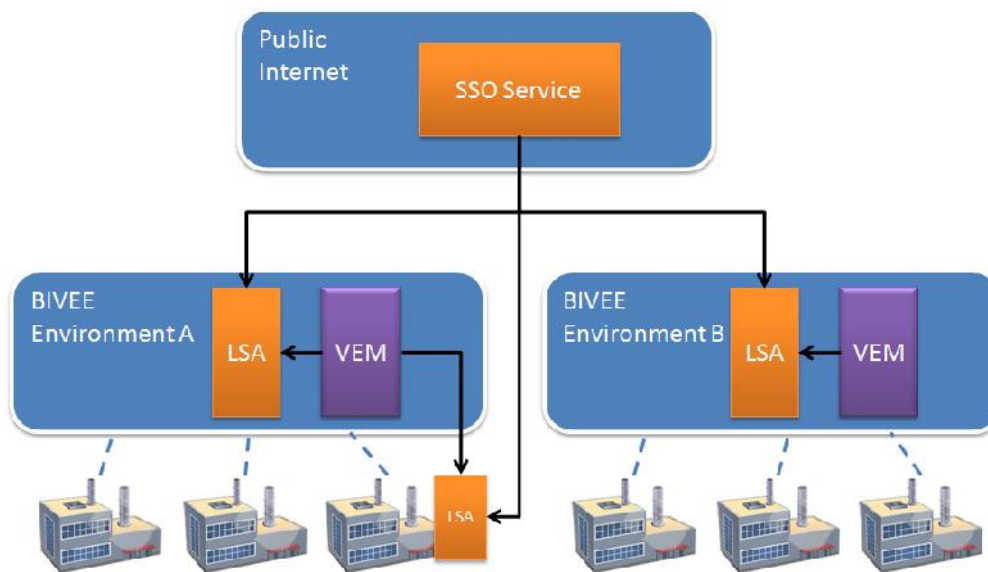


Figure 16 - BIVEE Environment federation

4.2.6 BIVEE standard SSO attributes and roles

In the BIVEE Environment, the SSO module is responsible not only for user authentication, but also defines a standard schema for user data. As previously mentioned (see section 4.2.3), the BIVEE implementation of SSO adds the *attribute release* feature, which means that BIVEE Applications will obtain a standard set of attributes *attached* to each authenticated user.

Attributes are basically a name-value pair, where the name is one of the following:

- **BIVEE_Name** (1 occurrence): Screen name of the user
- **BIVEE_Email** (0/1 occurrence): Email address of the user
- **BIVEE_Organization** (0-* occurrences): List of Enterprises the user belongs to
- **BIVEE_VirtualOrganization** (0-* occurrences): List of Virtual Enterprises (VEs) the user is member of
- **BIVEE_Role** (0-* occurrences) - List of BIVEE Roles assigned to the user

The unique ID of the user - i.e., the login name - is also exposed to applications, even if not as an attribute (this is standard information provided by the base SSO protocol).

BIVEE Applications can, and should, use the security-related attributes - Organization, Virtual Organization and Role - for applying their own user profiling logic. In the BIVEE Environment there is a standard set of Roles defined for this purpose:

- **BIVEE_User** - basic access to non-public areas of the BIVEE Platform
- **PIKR_User** - access to the PIKR application
- **VIF_User** - access to the VIF application
- **MCR_User** - access to the MCR application
- **BIVEE_KPI_Admin** - access to KPI data management capabilities
- **BIVEE_Ecosystem_Admin** - access to VEM and to web portal user administration
- **BIVEE_Admin** - no restrictions in the BIVEE Environment, full admin control on web portal

Note about PIKR_User, VIF_User and MCR_User: these are coarse-grained roles that enable a first-level user profiling - who can access which BIVEE Application. Each BIVEE Application may also add its own user profiling layer on top - e.g., enable / disable some of its features on the basis of user identity. The management of this kind of local profiling is in charge to each single application, and is not covered by this document.

Attributes are released to applications in SAML format - see section 4.2.2. Technical details of the attribute release feature, including an example of SAML, are given in Annex B - Single-Sign-On Factsheet.

4.3 Virtual Enterprise Management

Virtual Enterprise Management (VEM) is a simple module which enables the management of Virtual Enterprises (VE) in the BIVEE Environment. The operations supported are the following:

- Create, modify and delete VE entities in the BIVEE Environment
- Add and remove Enterprises as VE members

VEM is integrated with the BIVEE Environment at several levels:

- VEM web UI (VEM-FE) is part of the BIVEE Platform front-end
- VEM web API (VEM-WS) is part of the BIVEE Platform common services
- VEM is exploited by the BIVEE Platform's SSO module: the notion of VE membership is attached as an attribute to authenticated users; from there, it propagates to all BIVEE Applications

All supported operations can be performed both by BIVEE Platform administrators through the VEM-FE, and by BIVEE Applications using VEM-WS, which is implemented as a set of REST calls.

See the user manual (section 5.8) for screenshots and details about VEM-FE, and Annex C - Virtual Enterprise Management Factsheet for technical documentation on VEM-WS API.

4.4 Raw Data Handler

The Raw Data Handler (RDH) module, and mainly its Data Storage sub-module (RDH-DS), address the need of collecting Key Performance Indicator (KPI) data from BIVEE ecosystem members, and publish them to consumer applications - such as the Mission Control Room (MCR) - for monitoring and analysis.

The most relevant feature of RDH / RDH-DS is its integration with the Production and Innovation Knowledge Repository (PIKR), which defines a reference ontology for BIVEE KPIs (KPIOnto). By virtue of this integration, KPI data can be collected from heterogeneous sources - typically Enterprise IT systems belonging to different organizations - and still be presented and analyzed in a homogeneous way. This ontology-driven approach to data interoperability is commonly called *semantic lifting*.

Note that RDH / RDH-DS are just two elements of a complex, multi-layered solution (see Figure 17) having WP5 as the main contributor. More specifically, the prototype deliverable D5.3 "Advanced Interoperability Services" [8] (released by M21) had the objective of providing "... advanced services to achieve full interoperability among the different activities and services taking place in the business ecosystem, which may 'speak' different languages and are not natively able to exchange data and knowledge among each other" [9]. In the scope of the T5.3 task, RDH and RDH-DS were extensively redesigned in order to support the new *Data Interoperability Services* WP5 artefact. The complete reference documentation of this design is available in the D5.3 deliverable; in this D6.32 deliverable, where the actual prototype implementation of RDH and RDH-DS is done, we just provide some highlights.

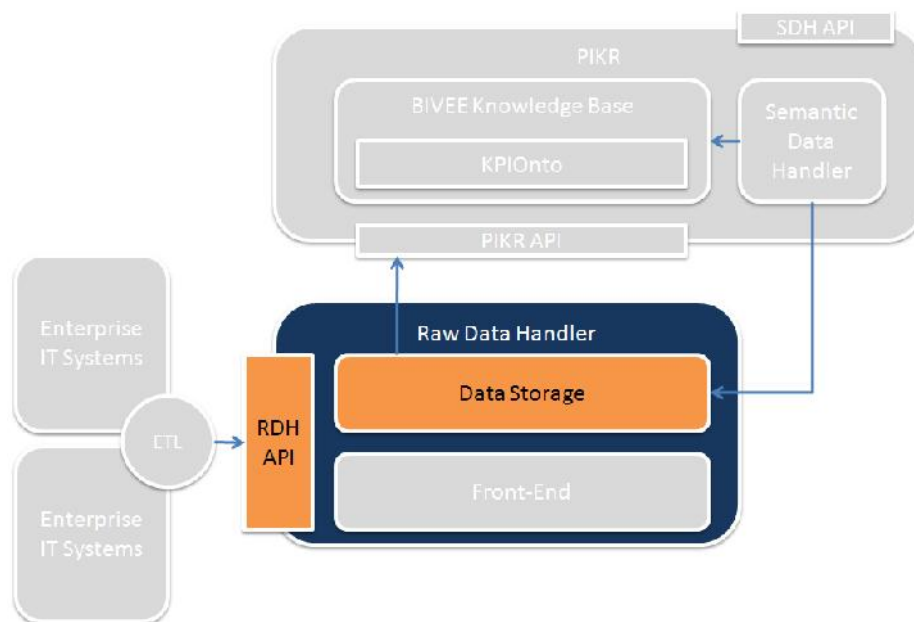


Figure 17 - Relationship of the RDH module with BIVEE components

4.4.1 Introduction

In KPIOnto, KPI definitions are instances of the Indicator ontology class. The other semantic concepts of relevance are the Dimension class and its sub-classes. These, together with their concrete instances (i.e., dimension members), represent a standardized, machine-readable model of the real world from which KPI data originates. Using this model as KPI meta-data, it is then possible to achieve a unified view of KPIs, regardless of their original source and structure. For instance, a KPI may be defined in KPIOnto as supporting OrganizationDimension (see Figure 18), which means that each single KPI value collected by the BIVEE system will have meta-data attached describing how that value relates with standard BIVEE entities like Enterprise, Department, ProjectTeam, etc.

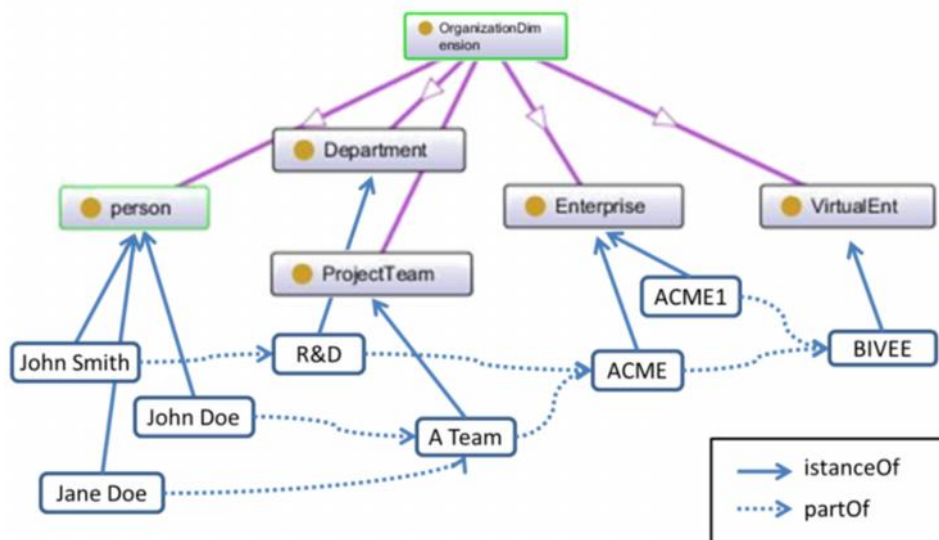


Figure 18 - Example of Dimension in KPIOnto

This use of meta-data is an example of *semantic annotation*. In the RDH context it is implemented by storing KPI data collected from heterogeneous systems in *ad-hoc* structures, which can be easily matched back with concepts of the reference ontology. This responsibility is on the Data Storage sub-module: RDH-DS.

4.4.2 RDH-DS

The RDH-DS sub-module is a relational database management system (RDBMS) with a specific schema in which KPI data is stored. This data schema is actually *auto-generated* by the RDH-Sync utility (see section 0) using KPIOnto as a blueprint, so that historical KPI data are kept in *normalized* form – that is, in a relational format that is highly compatible with the logical structure modelled by KPIOnto. The main value of data normalization is that application queries, expressed in KPIOnto terms, can be then translated by the Semantic Data Handler into equivalent RDH-DS queries (SQL) which will perform very fast.

Basically, KPIOnto classes derived from the Dimension class are translated into domain tables (see Figure 19), and are populated with records corresponding to concrete instances (see Figure 20); each Indicator instance becomes a *fact* table by its own, connected with the relevant domains (see Figure 21). The resulting schema is described in the RDH-DS Catalogue, and from there entities can be easily mapped back to KPIOnto concepts (see Figure 22).

The RDH-DS sub-module is a *private* back-end component, in that its functionality is made available *externally* (i.e., to applications which are not integrated into the BIVEE Environment) only by means of another module: RDH-WS, implementing the RDH web API layer (see section

4.4.3). Internally, RDH-DS is accessible through a direct, low-level channel: a Java Database Connectivity (JDBC) driver.

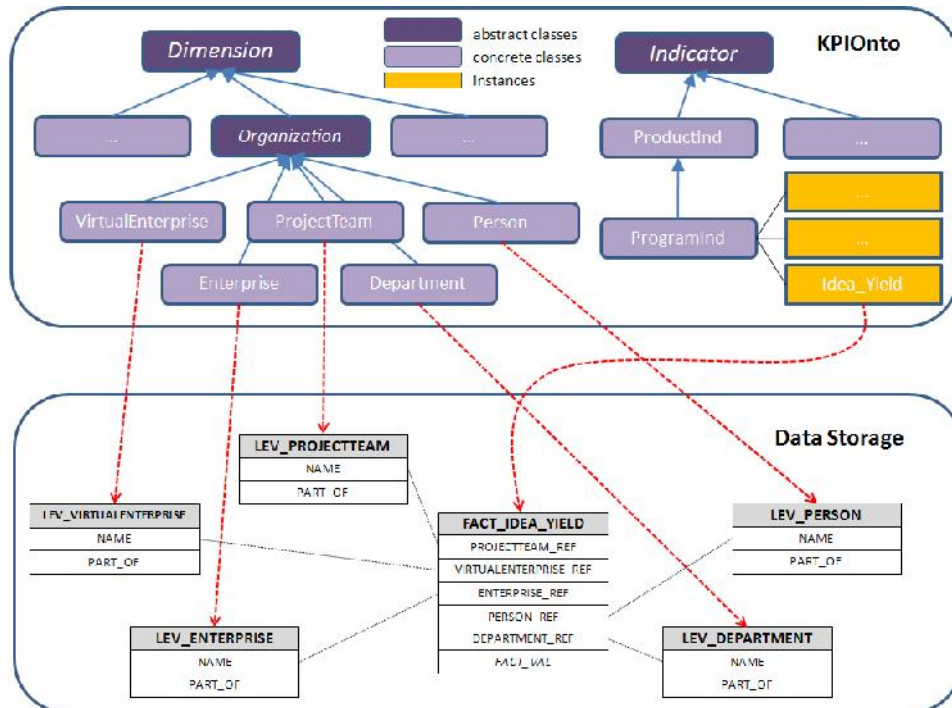


Figure 19 - Example of how ontology concepts are mapped to database tables

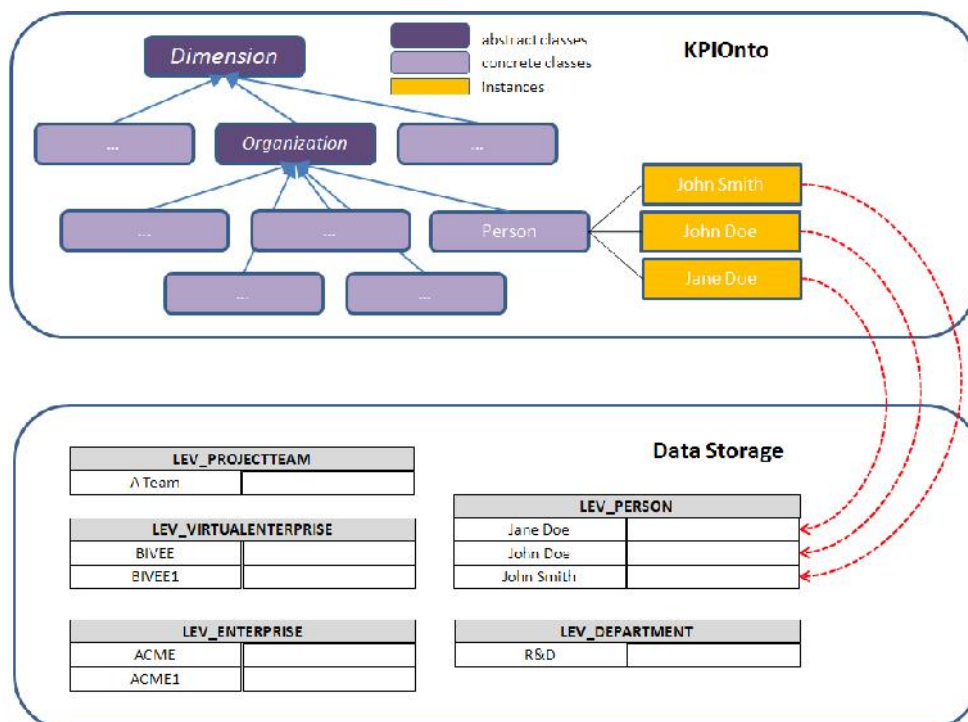


Figure 20 - Example of how ontology concept instances are mapped to database records

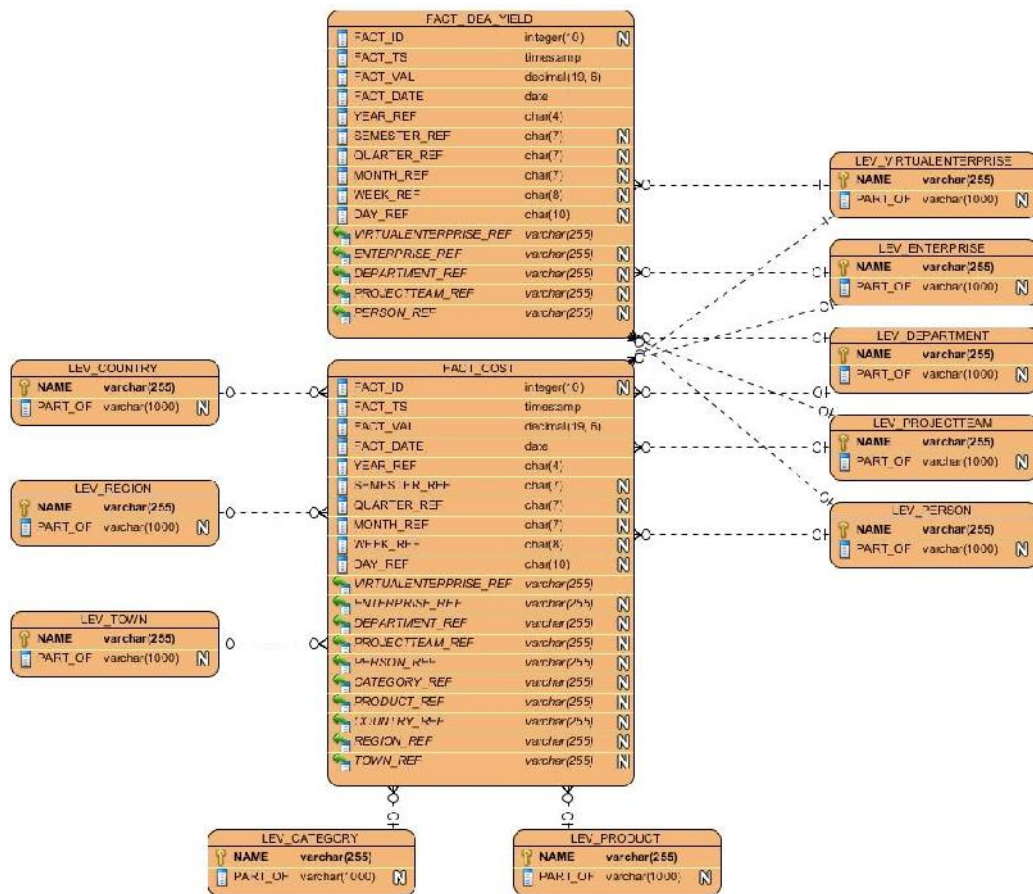


Figure 21 - Example of DB schema involving two KPI *fact tables*

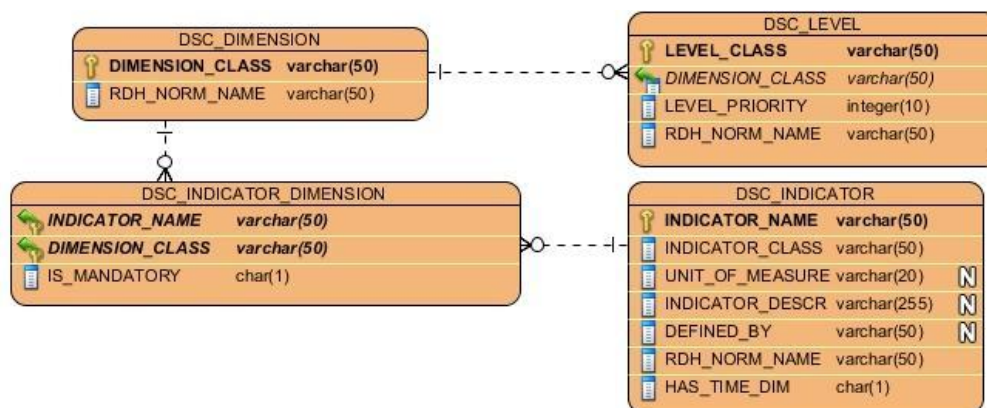


Figure 22 - DB schema of the RDH-DS Catalogue

4.4.3 RDH-WS

The RDH module exposes a web API (RDH-WS), used by external systems to load KPI data in RDH-DS and by RDH-FE to provide KPI data navigation and administration functionality to BIVEE Platform users.

RDH-WS is implemented as a HTTP/SOAP web service which exposes a set of operations:

- Retrieve all the existing KPI definitions
- Retrieve all the dimensions and dimension levels for a specific KPI definition
- Retrieve all the values stored for a specific KPI
- Load/update/delete values for a specific KPI

A formal, WSDL-based definition of this API is given in Annex D - Raw Data Handler Factsheet.

4.4.4 RDH-FE

The RDH module also provides a web-based front-end UI (RDH-FE), which is an integral part of the BIVEE Platform. It enables users to navigate and manage the KPI historical data maintained in RDH-DS. All the operations provided by RDH-WS (see 4.4.3) are exposed by RDH-FE as interactive functionalities.

See the user manual of this sub-module (section 0) for screenshots and details.

4.4.5 RDH-Sync

RDH-Sync is an utility program that keeps RDH-DS schema aligned with KPIOnto, leveraging on the PIKR web API to retrieve KPIOnto. It has to run once when the BIVEE Platform is installed, and then each time KPIOnto is modified.

RDH-Sync tries to avoid the disruption or the loss of stored data, whenever possible, but there are operations that may generate loss of historical data, or that may cause a data integrity violation problem.

In particular, loss of historical data may be caused by:

- the removal of an existing indicator (because of the dropping of the corresponding fact table)
- the removal of a dimension for an existing indicator (because of the removing of the corresponding columns from the fact table)
- the removal of a level for dimension (because of the removing of the corresponding column from all the fact table where the column were present).

Data integrity violation problems may occur when:

- adding a new dimension to an existing indicator
- removing an existing dimension instance

This sub-module is activated interactively by the user, by means of a menu entry in the RDH-FE web UI - see section 5.8.

5 BIVEE PLATFORM USER GUIDE

5.1 Login to the Web Portal

The user loads the Web Portal URL⁶ in her web browser: the Portal's home page is displayed. Only the "Home" entry is showing up in the first level of the Platform's menu, as the user is still navigating as a *guest* (anonymous user).



Figure 23 - Web Portal home page

To access the full set of functionalities available through the Web Portal, the user clicks on the "Sign In" button.



Figure 24 - Sign In button

⁶ Ask the URL of your BIVEE Platform installation to your IT contact

The user is redirected to the login form. The user types her username and password, as provided by the Web Portal administrator, in the corresponding fields of the login form. Then, she clicks on the “Confirm” button.

If the credentials are valid, the Web Portal home page is again displayed, this time with a set of menu options according to the user's profile - see section 5.3. For the sake of completeness, from now on we assume that the user holds full administrative rights, so that the entire set of menu options is available.

5.2 BIVEE Platform menu

The menu of the BIVEE Platform is structured on two levels. The first one lists *areas of interest* within the BIVEE Environment - e.g., a BIVEE Applications or a group of related functionalities in the BIVEE Platform. These top-level entries may have child entries as well, where multiple entry points or different functionalities exist. The second level of the menu is hidden by default: you can reach it by clicking on a top-level entry, which is then expanded showing all its children.

Individual menu entries, regardless of their level, are also selectively hidden to reflect the actual access rights of the current user. Profiling is done by user role - see section 4.2.6 for details on the standard roles supported in the BIVEE Environment.

The complete list of entries in the BIVEE Platform menu follows. For each of them, the list of enabling roles is also given - i.e., the user must hold at least one of them to have access.

- Home (*public*)
- Mission Control Room (**MCR_User, BIVEE_Ecosystem_Admin, BIVEE_Admin**)
- Virtual Innovation Factory (**VIR_User, BIVEE_Ecosystem_Admin, BIVEE_Admin**)
 - Dashboard (*same as top-level entry*)
 - New Idea (*same as top-level entry*)
 - Posted Ideas (*same as top-level entry*)
 - Challenges (*same as top-level entry*)
 - Observatory (*same as top-level entry*)
 - Store (*same as top-level entry*)
 - Whiteboard (*same as top-level entry*)
 - Admin (*same as top-level entry*)
- PIKR (**PIKR_User, BIVEE_Ecosystem_Admin, BIVEE_Admin**)
- KPI (**BIVEE_User, BIVEE_KPI_Admin, BIVEE_Ecosystem_Admin, BIVEE_Admin**)
- Admin (**BIVEE_Ecosystem_Admin, BIVEE_Admin**)
 - Virtual Enterprises (**BIVEE_Ecosystem_Admin, BIVEE_Admin**)
 - RDH Synchronization (**BIVEE_Admin**)

Details on each entry are given in individual sections below.

5.3 Menu entry: Home

This is the default page of the BIVEE Platform portal. It is accessible to any user. The screenshot below shows the appearance of the home page to a user holding full admin rights.



5.4 Menu entry: Mission Control Room

This menu entry links to the landing page of the Mission Control Room (MCR), which runs as a standalone application - i.e., integrated with the SSO Service but not with the Web Portal. For this reason, no screenshot is provided here: see MCR's own documentation [6] for the user manual. In the last release of the BIVEE Environment, full UI integration will be achieved.

This menu entry is accessible only to users holding any of these roles: MCR_User, BIVEE_Ecosystem_Admin, BIVEE_Admin.

5.5 Menu entry: Virtual Innovation Factory

This is the landing page of the Virtual Innovation Factory (VIF), whose UI is fully integrated with the Web Portal. This, as well as the inner pages which compose the VIF front-end (see following sub-sections), are accessible only to users holding any of these roles: VIF_User, BIVEE_Ecosystem_Admin, BIVEE_Admin.

The following is a screenshot of VIF's landing page.



A second-level menu deploys under the VIF main entry, each entry corresponding to a VIF module. Each of the sub-sections that follow provides a relevant screenshot. See VIF's own documentation [7] for the user manual.

5.5.1 Sub-menu entry: Dashboard



The dashboard features a header with the Bivee logo and navigation tabs: Mission Control Room, Virtual Innovation Factory, PIKR, KPI, and Admin. The main content area includes:

- Innovation ToDo:** A table listing tasks with columns for Activity name, Priority, and Deadline.

Activity name	Priority	Deadline
innovation task M1	medium	in 3 days
innovation task M5	who cares?	in 2 days
innovation task M3	high	in 2 days
innovation task M8	medium	in 2 days
innovation task M2	high	in 2 days
innovation task M2	who cares?	10 hours ago
- Your recent activity:** A list of recent actions with columns for Type, When, Score (x 100), Status, and Karma.
- The VIF Store:** A carousel of product images labeled 'Store Item 4', 'Store Item 6', and 'Store Item 8'.
- Open challenges:** A section for challenges with icons and descriptions.

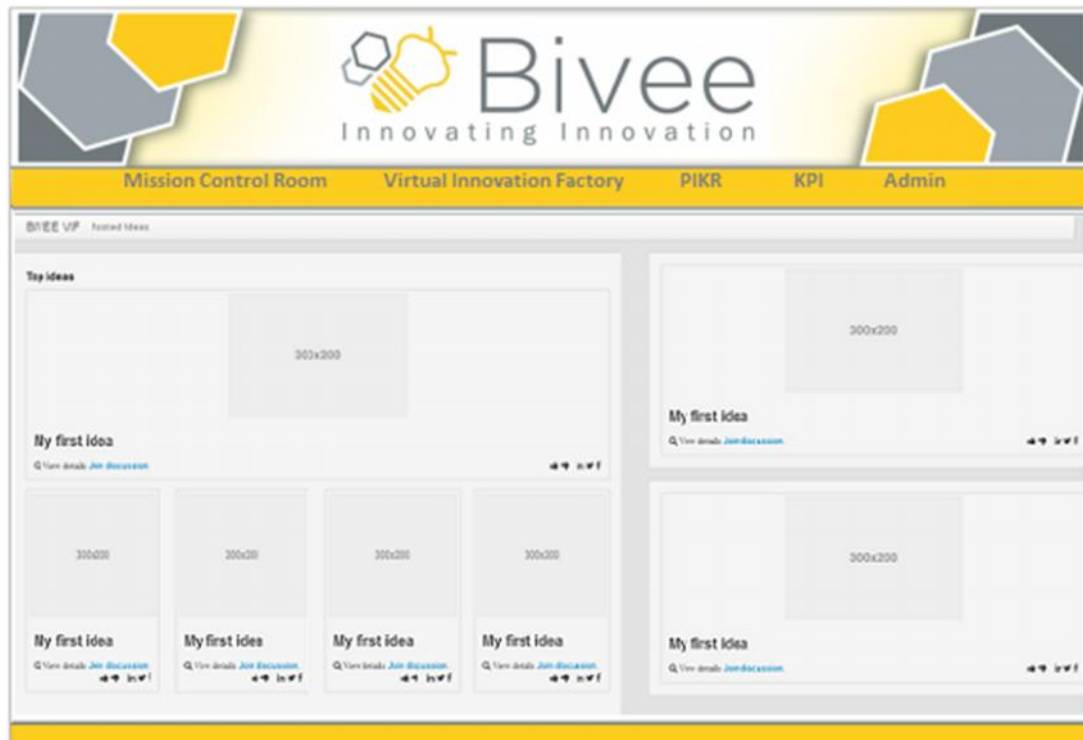
5.5.2 Sub-menu entry: New Idea



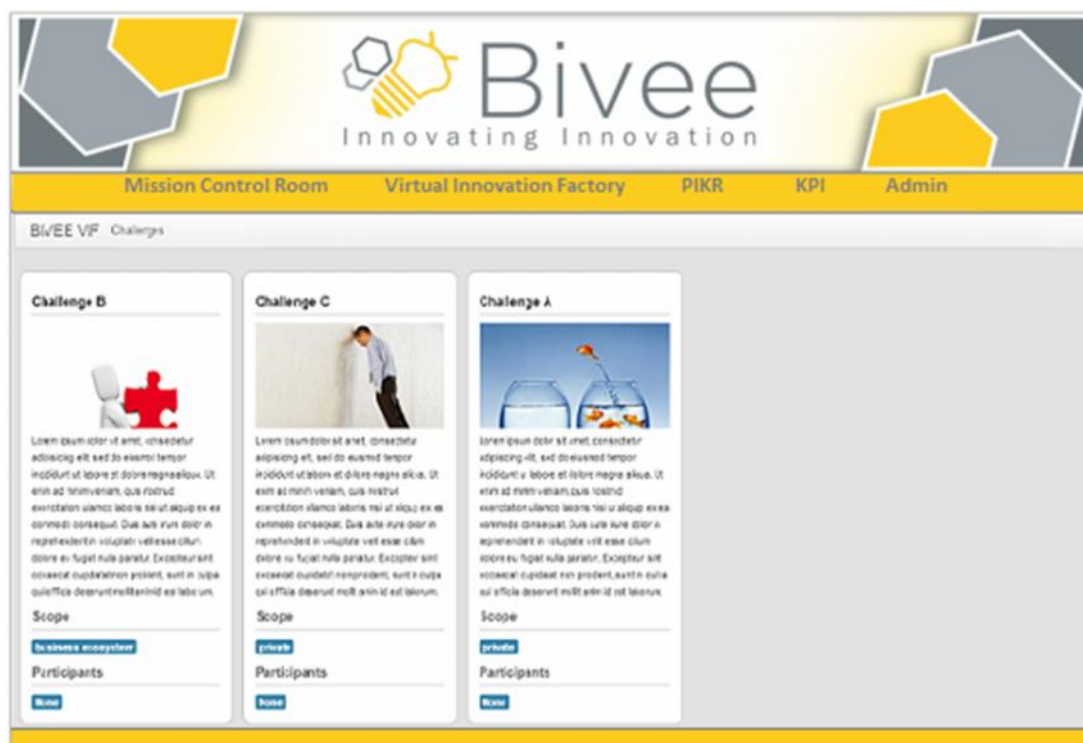
The 'New Idea' page features the Bivee header and navigation tabs. The main content area includes:

- Idea in draft status:** A section for drafting ideas.
- Where should I go with my idea?:** A central area with a stick figure icon and a text box for idea development.
- On-going ideas:** A section for ongoing ideas with a 'Click on required modules' link.
- Yes, I have a great idea!** A green button at the bottom right.

5.5.3 Sub-menu entry: Posted Ideas



5.5.4 Sub-menu entry: Challenges





Bivee

Innovating Innovation

[Mission Control Room](#)
[Virtual Innovation Factory](#)
[PIKR](#)
[KPI](#)
[Admin](#)

BIVEE VIF [store](#)

User resume



Current score: 1482

Average collector within your network

You can obtain ...

	Store Item 4 55 USD	Buy
	Store Item 5 27 USD	Buy
	Store Item 6 45 USD	Collect
	Store Item 7 34 USD	Buy
	Store Item 8 34 USD	Propose
	Store Item 9 55 USD	Propose
	Store Item 10 45 USD	Buy
	Store Item 11 55 USD	Buy
	Store Item 12 45 USD	Propose

Highlighted item



Product description

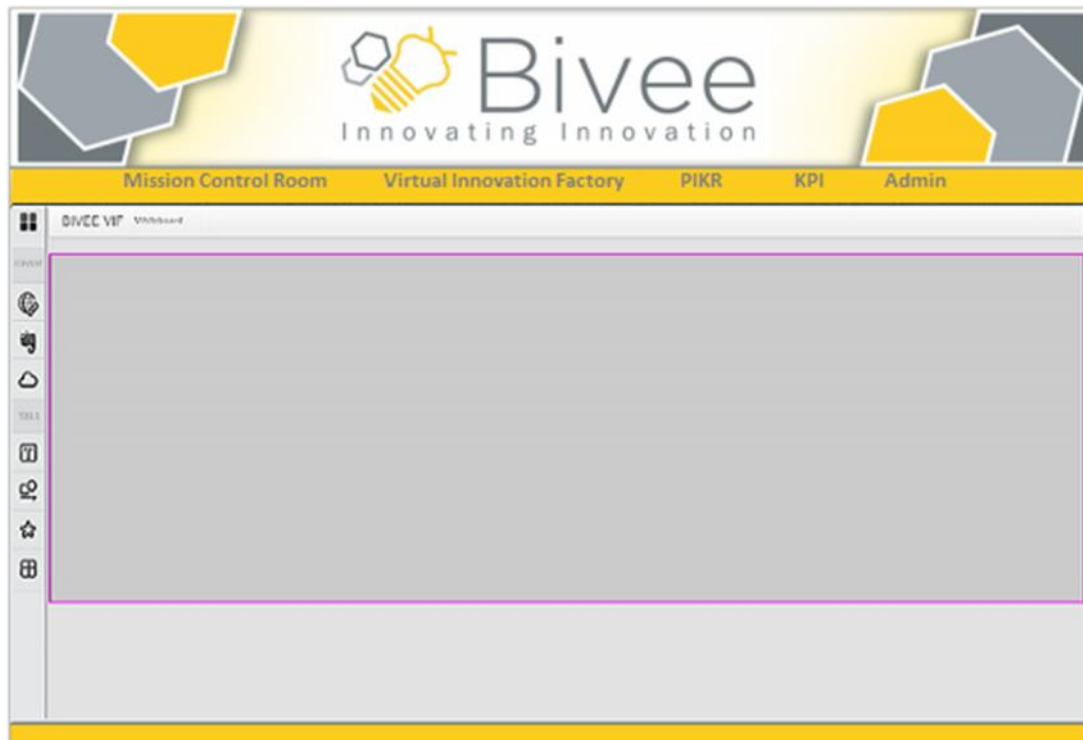
Consequence on this item



Ministar need a

Reward history

5.5.7 Sub-menu entry: Whiteboards



5.5.8 Sub-menu entry: Admin



5.6 Menu entry: PIKR

This menu entry links to the landing page of the Production and Innovation Knowledge Repository (PIKR), which runs as a standalone application - i.e., integrated with the SSO Service but not with the Web Portal. For this reason, no screenshot is provided here: see PIKR's own documentation [4] for the user manual. In the last release of the BIVEE Environment, full UI integration will be achieved.

This menu entry is accessible only to users holding any of these roles: PIKR_User, BIVEE_Ecosystem_Admin, BIVEE_Admin.

5.7 Menu entry: KPI


This is the entry point for RDH-FE - i.e., the web-based front-end of the Raw Data Handler module. RDH-FE base functionality (read) is available to authenticated users having any role, while some privileged operation (write) are enabled only for users having one of these roles: BIVEE_KPI_Admin, BIVEE_Ecosystem_Admin, BIVEE_Admin.

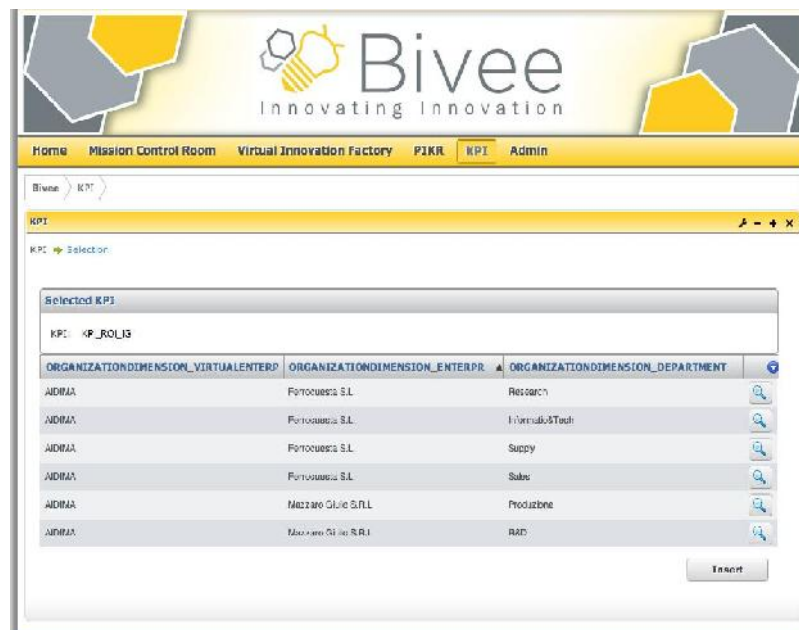
The landing page displays the list of KPI definitions for which data exists in the Data Store (RDH-DS).



Class	Name	Description	Unit Of Measure
PRODUCTION	KPI_COST	PRODUCTION COST	EURO
CONCEPTION	KPI_DISPLAYED	NUMBER OF IDEAS VIEWED	NUMBER
FINANCIAL	KPI_RETURN	RETURN ON INVESTMENT	NUMBER


Figure 25 - RDH-FE landing page

The user can browse stored data for any KPI by clicking on the  icon of the corresponding row. The page will then display the list of all the historical data sequences that are available for the given KPI.



ORGANIZATIONDIMENSION_VIRTUALENTERPR	ORGANIZATIONDIMENSION_ENTERPR	ORGANIZATIONDIMENSION_DEPARTMENT
AIDMAA	Fortocassa S.L.	Research
AIDMAA	Fortocassa S.L.	InformationTech
AIDMAA	Fortocassa S.L.	Supply
AIDMAA	Fortocassa S.L.	Sales
AIDMAA	Mezzano Olive S.R.L.	Produzione
AIDMAA	Mezzano Olive S.R.L.	R&D

Figure 26 - RDH-FE: KPI data

From here, the user can browse an historical value sequence by clicking on the  icon of the corresponding row. The page will then display the list of all values stored for the given sequence.

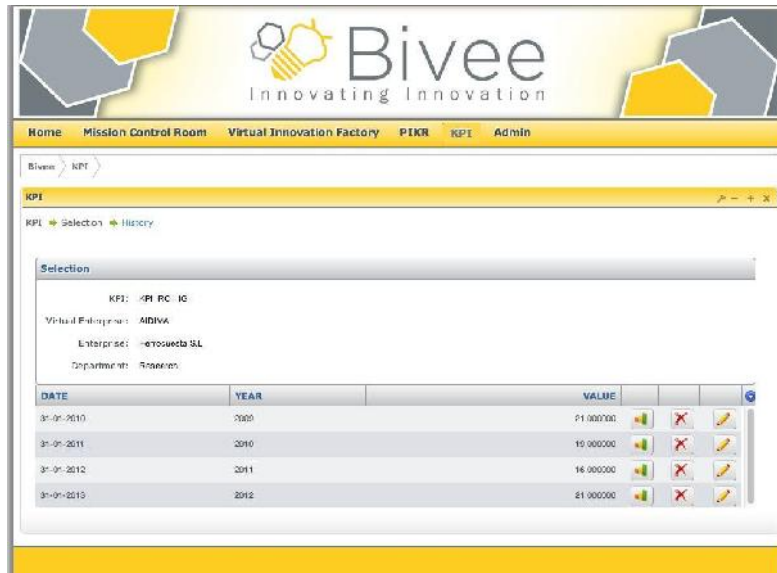




Figure 27 - RDH-FE: KPI value sequence

On each row, any user holding the appropriate rights (BIVEE_KPI_Admin, BIVEE_Ecosystem_Admin or BIVEE_Admin role) can either delete  or modify  the corresponding KPI value.

The same privileged user can, from the KPI data page, start a totally new KPI value sequence by clicking on the "Insert" button. This displays an input form where the user may determine name, attributes and value of a new record, by selecting from option lists. The list of available KPIs, of the relevant attributes (dimension levels) and of their legal values are all sourced from PIKR's KPIOnto. When done, the user commits the new KPI value to RDH-DS by clicking on the "Save" button.



Figure 28 - RDH-FE: insert KPI value

5.8 Menu entry: Admin

Administration is not an entry point by itself: it collects admin-related functionality as second-level items, as detailed in the following sub-sections.


5.8.1 Sub-menu entry: Virtual Enterprises


This is the entry point for VEM-FE - i.e., the web-based front-end of the Virtual Enterprise Management module. VEM-FE functionality is available only to users having the BIVEE_Ecosystem_Admin or the BIVEE_Admin role - i.e., to ecosystem or web portal administrators.

The landing page displays the list of Virtual Enterprises (VE) registered in the VEM.



Figure 29 - VEM-FE landing page

The user can delete any VE by clicking on the  icon of the corresponding row; the system will ask the user for confirmation. When a VE is removed from the registry, all membership information (i.e., the notion of which Enterprises are part of the given VE) is lost.

The user can navigate to the details of any VE by clicking on the  icon of the corresponding row; a dialog window is displayed:

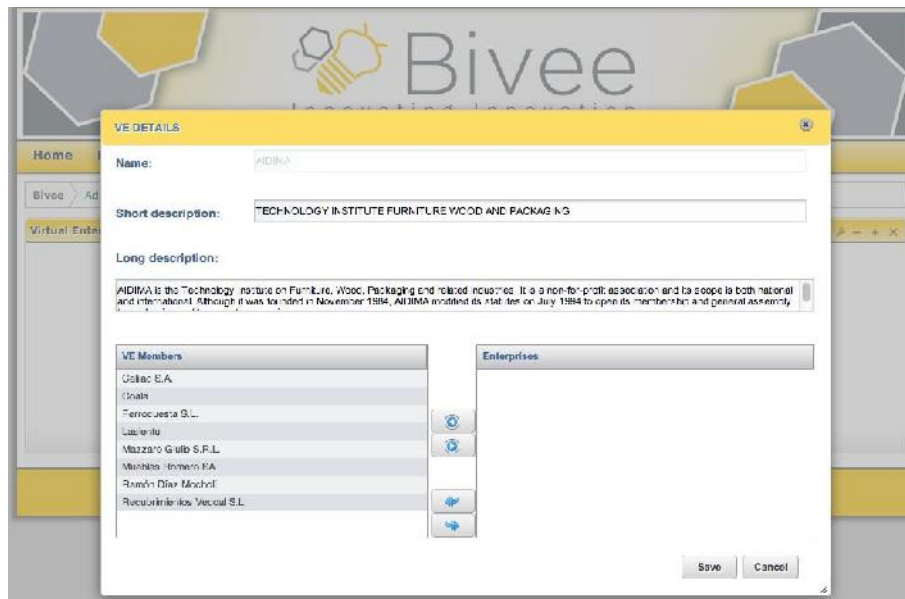






Figure 30 - VEM-FE: VE Details dialog window

The user can modify the short (max 255 characters) and long (max 1000 characters) descriptions of the VE.

The user can also edit the member list of the VE:

- add a member by selecting it in the “Enterprises” list, and by clicking on  ;
- remove a member by selecting it in the “VE Members” list, and by clicking on  ;
- remove all members by clicking on  ;
- add all non-members by clicking on .

To make changes permanent the user has to click on the “Save” button. By clicking on the “Cancel” button, all changes are discarded and the dialog window is closed.

From the main page, the user can create a new VE by clicking on the “Insert VE” button.

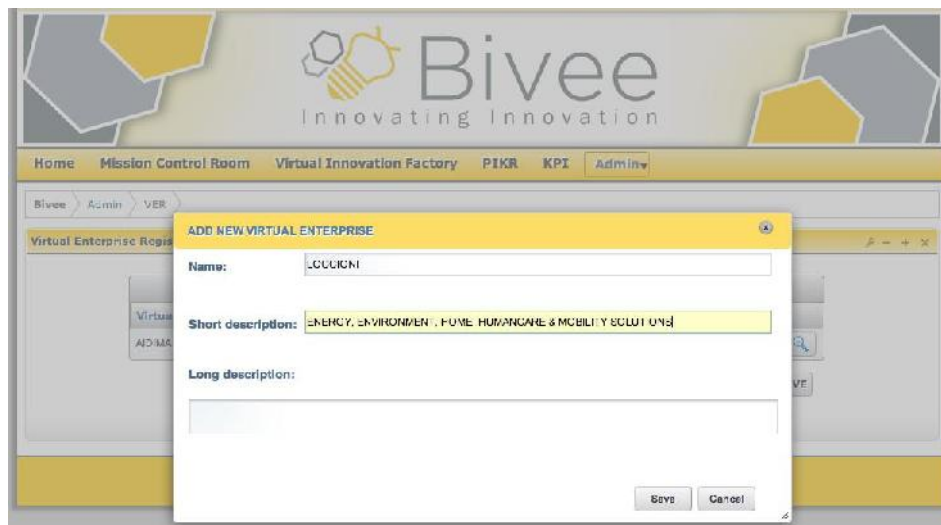


Figure 31 - VEM-FE: new VE dialog window

The user has to insert a VE name, which has to be unique and not exceeding 50 characters in length. A short (max 255 characters) and a long (max 1000 characters) description may also optionally be provided.

When done, the user has to click on the “Save” button to insert the new VE in the registry. By clicking on the “Cancel” button, the dialog window is closed without any change.

5.8.2 Sub-menu entry: RDH Synchronization

The RDH Synchronization functionality is available only to users having the BIVEE_Admin role - i.e., BIVEE Platform administrators.

When the user select this menu item, the system will ask for confirmation with a confirm / cancel dialog window. In case of confirmation, the system will execute the RDH-Sync program in the background, meaning that RDH-DS structure and (partially) contents will be synchronized with the current state of PIKR's KPIOnto ontology without any further user interaction, and providing no feedback during the process.

It is worth noting that this functionality may have adverse effects on existing KPI data stored in RDH-DS, so it is advisable to use it with caution.

5.9 User and Enterprise management

This section explains how Users and Enterprises can be managed on the BIVEE Platform. Organizations are the Web Portal abstraction that is used to represent Enterprises. In this section, we will then refer to Enterprises as Organizations.

Users with the Ecosystem Administrator or the System Administrator role can access the "Go to -> Control Panel" menu item, as shown by the following picture.



Figure 32 - Go To menu

While in the Control Panel, open the "Portal" menu located in the left column and click on the "Users and Organizations" item.

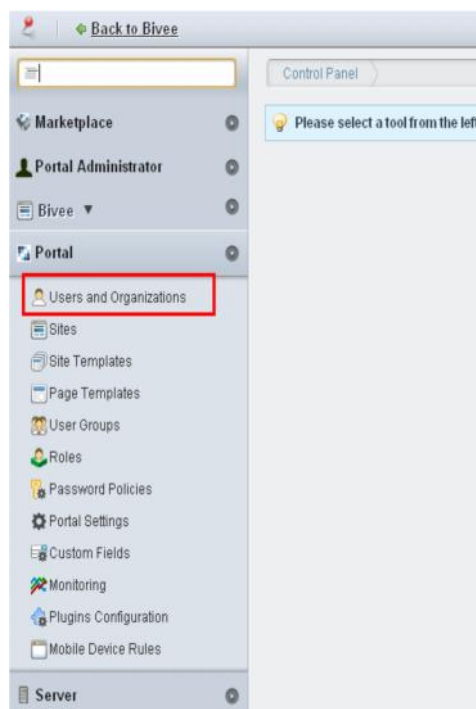
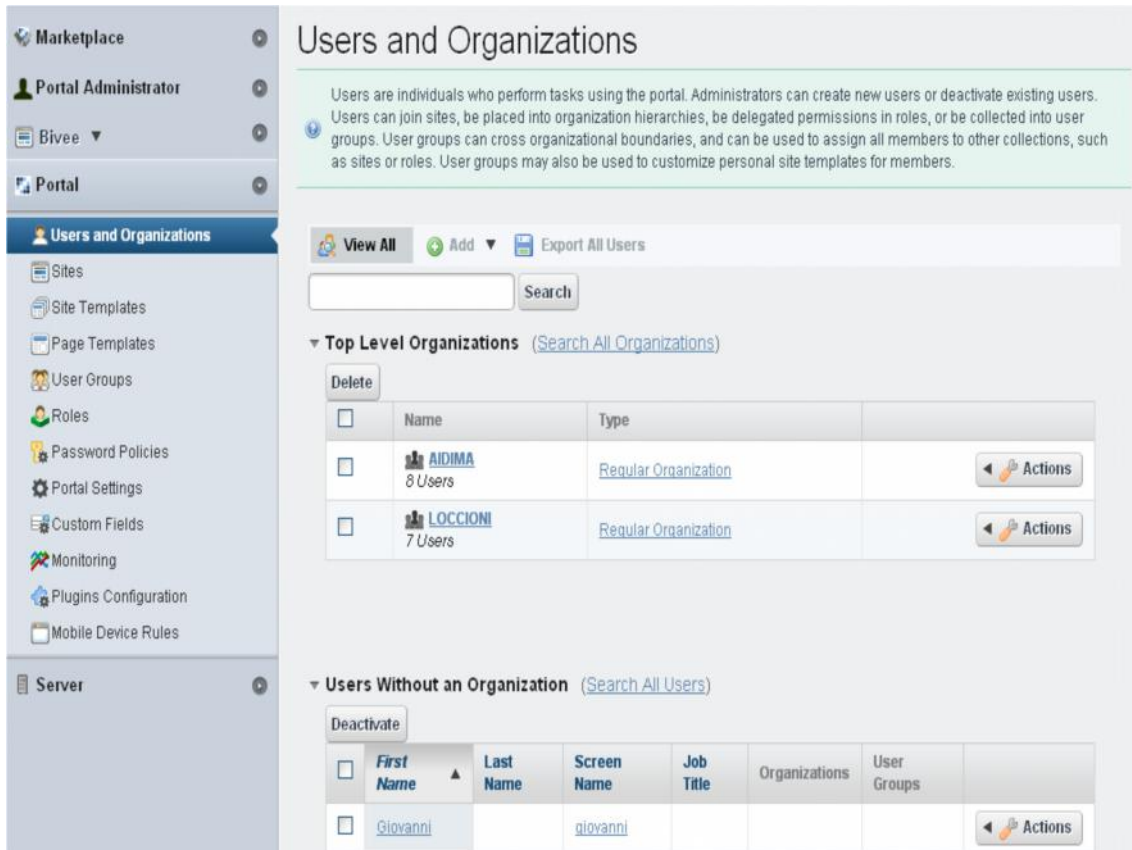


Figure 33 - Control Panel menu

Two tables are displayed:

- “Top Level Organizations”: this table lists all the Organizations registered with the Web Portal
- “Users Without an Organization”: this table contains Users not associated with any Organization



Users and Organizations

Users are individuals who perform tasks using the portal. Administrators can create new users or deactivate existing users. Users can join sites, be placed into organization hierarchies, be delegated permissions in roles, or be collected into user groups. User groups can cross organizational boundaries, and can be used to assign all members to other collections, such as sites or roles. User groups may also be used to customize personal site templates for members.

Top Level Organizations ([Search All Organizations](#))

Name	Type	Actions
AIDIMA 8 Users	Regular Organization	Actions
LOCCIONI 7 Users	Regular Organization	Actions

Users Without an Organization ([Search All Users](#))

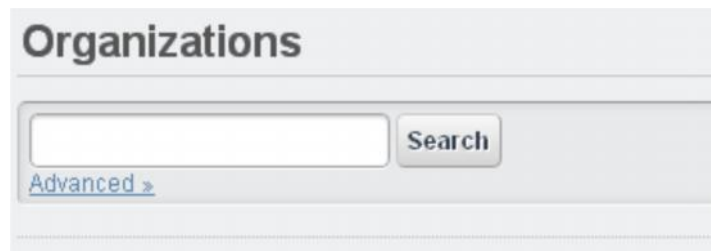
First Name	Last Name	Screen Name	Job Title	Organizations	User Groups	Actions
Giovanni		giovanni				Actions

Figure 34 - Users and Organizations page

Through the web link “Search All Organizations” it is possible to open a search box to perform the search action over all registered Organizations.

This search box is available in two flavours:

- Basic
- Advanced



The image shows a web form titled "Organizations". It features a single text input field for search criteria, a "Search" button, and a link labeled "Advanced >" below the input field.

Figure 35 - Organizations basic search



The image shows a more complex web form titled "Organizations". At the top, it says "Match ☐ All of the following fields:". Below this, there are several input fields: "Name", "Street", "City", "Type", "Country", "Region", and "Postal Code". The "Type", "Country", and "Region" fields have dropdown arrows. At the bottom, there is a "Search" button and a link labeled "< Basic".

Figure 36 - Organizations advanced search

Through the web link “Search All Users” it is possible to open a search box to perform the search action over all registered Users.

This search box is available in two flavours:

- Basic
- Advanced



The image shows a web interface titled "Users". Below the title is a search box with a "Search" button to its right. Below the search box is a link labeled "Advanced »".

Figure 37 - Users basic search



The image shows a web interface for advanced user search. At the top, it says "Match All of the following fields:". Below this are six input fields arranged in two rows: "First Name", "Middle Name", "Last Name" in the first row, and "Screen Name", "Email Address", "Status" in the second row. The "Status" field is a dropdown menu with "Active" selected. Below the input fields is a "Search" button and a link labeled "Basic".

Figure 38 - Users advanced search

To delete an Organization, select it and press the “Delete” button or click on the “Action” button and then on “Delete” item.

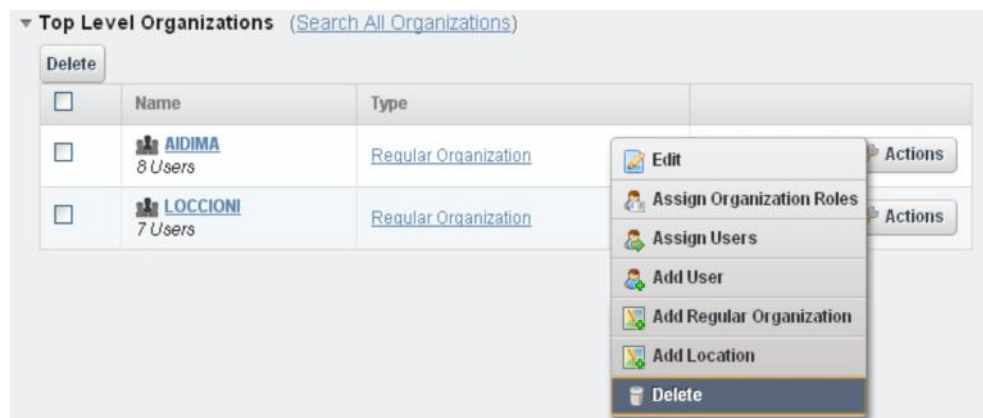


Figure 39 - Delete Organization

To delete an User it is necessary to deactivate it first. To deactivate an User, select it and press the “Deactivate” button or click on the “Action” button and then on “Deactivate” item



Figure 40 - Deactivate User

It is possible to get back a deactivated User through the search box described before.

Select the deactivated User you want to delete and press the “Delete” button or click on the “Action” button and then on “Delete” item.

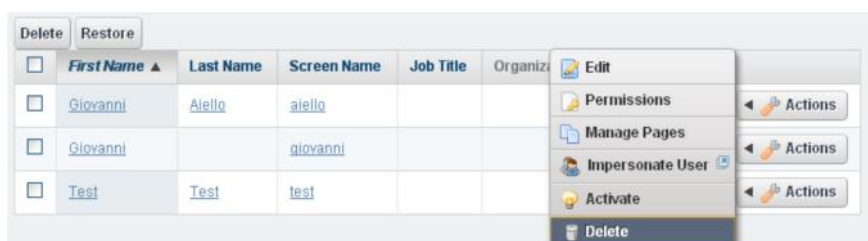


Figure 41 - Delete User

5.10 Add a new Organization

Click on the “Add” menu item and then on “Regular Organization”.

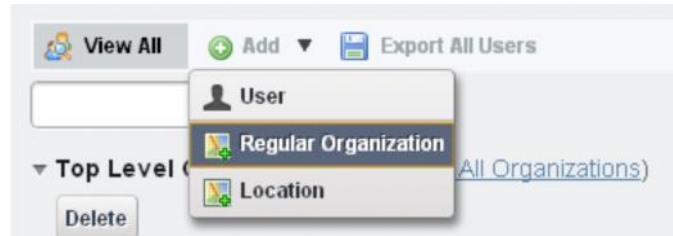


Figure 42 - Add Organization

Enter the Organization name in the form that is displayed; when done, click on the "Save" button to make changes persistent.

5.11 Add a new User

Click on the “Add” menu item and then on “User”.

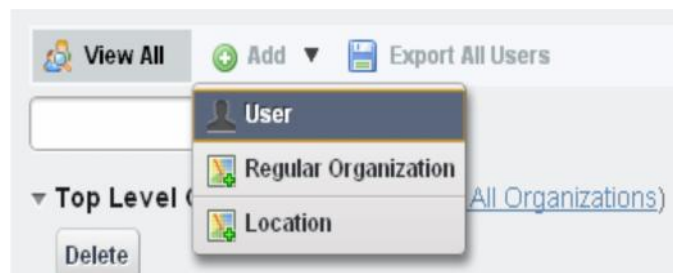
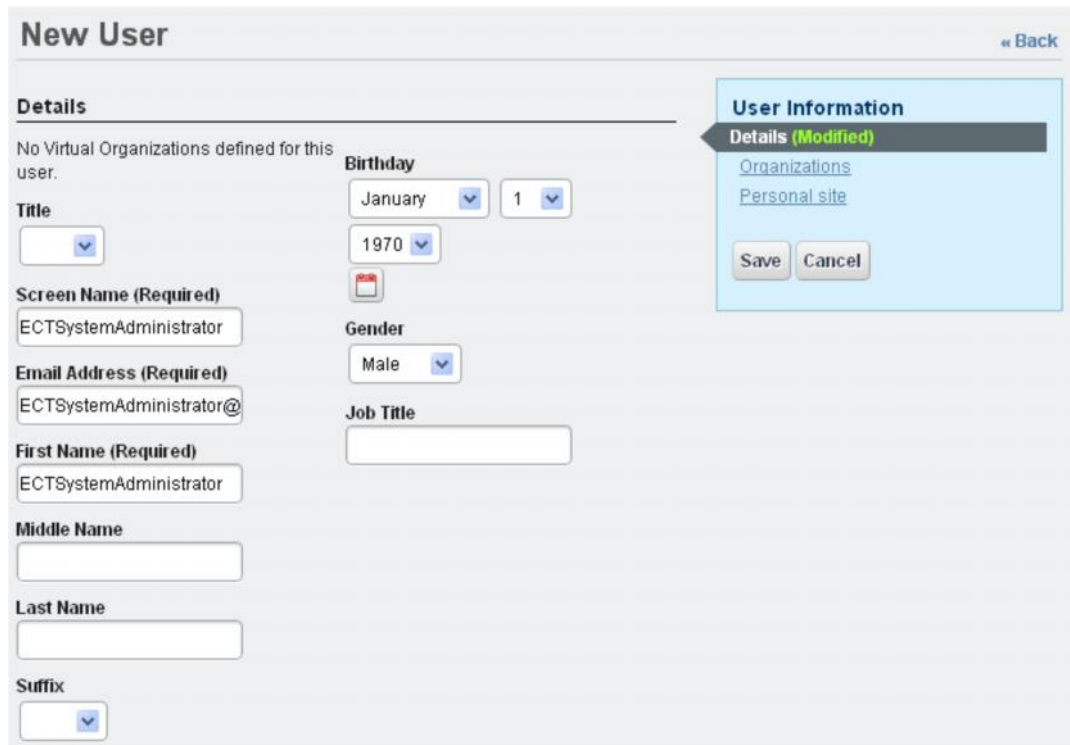


Figure 43 - Add User

It is then possible to fill the required fields with the desired values.



New User « Back

Details

No Virtual Organizations defined for this user.

Title

Screen Name (Required)

Email Address (Required)

First Name (Required)

Middle Name

Last Name

Suffix

Birthday
 January
 1970

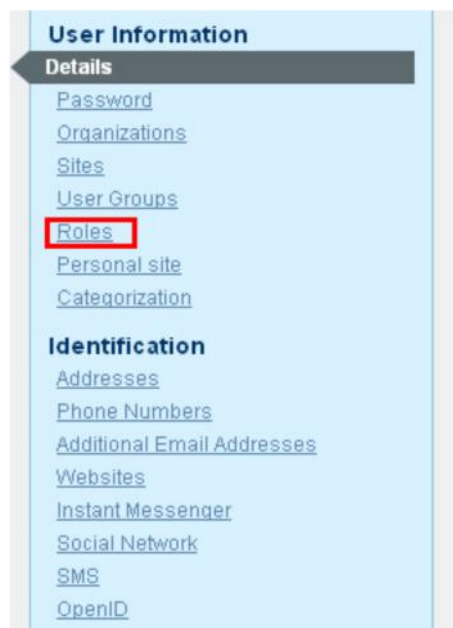
Gender
 Male

Job Title

User Information
Details (Modified)
[Organizations](#)
[Personal site](#)

Figure 44 - New User: basic data

To associate the User with one or more role she holds inside the Web Portal, click on the “Roles” item in the right column and select the desired Organizations (click on “Select” button).



User Information

Details

[Password](#)
[Organizations](#)
[Sites](#)
[User Groups](#)
[Roles](#)
[Personal site](#)
[Categorization](#)

Identification

[Addresses](#)
[Phone Numbers](#)
[Additional Email Addresses](#)
[Websites](#)
[Instant Messenger](#)
[Social Network](#)
[SMS](#)
[OpenID](#)

Figure 45 - User menu: Roles

To associate the User with the Organizations in which she is involved, click on “Organizations” item in the right column and select the desired Organizations (click on “Select” button).

When done, click on the "Save" button to make changes persistent.

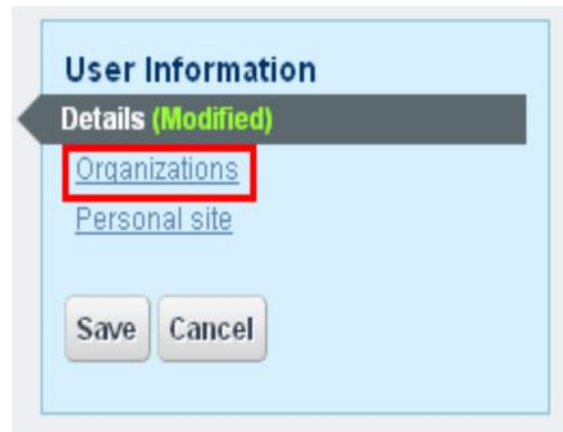


Figure 46 - User menu: Organizations



Figure 47 - New User: Organization list

6 REFERENCES

- [1] Camarinha-Matos, Afsarmanesh, Galeano, Molina, “Collaborative networked organisations - Concepts and practice in manufacturing enterprises”, Elsevier, 2008
- [2] BIVEE deliverable D6.20 Service Platform Specifications
- [3] BIVEE deliverable D6.31 BIVEE Environment prototype v1.0
- [4] BIVEE deliverable D5.1 Production and Innovation Knowledge Repository
- [5] BIVEE deliverable D4.1 State of the Art & Virtual Innovation Factory Design
- [6] BIVEE deliverable D3.21 Mission Control Room Prototype v1.0
- [7] BIVEE deliverable D4.21 Virtual Innovation Factory Prototype v1.0
- [8] BIVEE deliverable D5.3 Advanced Interoperability Services
- [9] BIVEE Description of Work document
- [10]<http://www.liferay.com/>
- [11] BIVEE deliverable D3.1 State of the Art and Mission Control Room Specification
- [12] Hrgovic, V; Karagiannis, D; Woitsch, R; “Conceptual Modeling of the Organisational Aspects for Distributed Applications: The Semantic Lifting Approach. IEEE conference COMPSAC 2013, 2013 Kyoto, Japan

ANNEX A - WEB PORTAL FACTSHEET

Prototype references

Version	2.0
Availability	http://bivee-sp.eng.it/
Source control	svn://repo.nimbus-ware.com/BIVEE/WP6/D6.32/trunk/WebPortal
Contact person	Mauro Isaja (mauro.isaja@eng.it)

Prototype technical information

Nature	Web Application
Programming Language	Java
Development Framework	Java SE Development Kit 1.6 or later
Additional libraries	Liferay Community Edition v.6
Application Server	J2EE servlet container (tested on Apache Tomcat 7)
Database	MySQL v5.1 or later

Installation pre-requisites

The prerequisites for installing the Web Portal application are:

- An operating system available, with no specific constraints (The application has been successfully installed and tested on Windows XP Enterprise Edition and Mac OS X).
- A MySQL instance version 5.1 or later (the application has been tested using MySQL Database engine, however any database with a JDBC driver available should work).

Installation instructions

The Web Portal Server application can be downloaded from

http://bivee.eng.it:8780/web/guest/documents/-/document_library/view/43585 (path : BIVEE Project Portal/Document Library/WP6 – Architecture IT Platforms federation and services/Deliverables/Final/D6.32).

The package *D6.32_BIVEE Environment prototype – M24 -WebPortal.zip* must be unzipped; it contains the folder *liferay-portal-6.1.1-ce-ga2-bivee* and the file *bivee_portal.sql*.

First of all, a new database must be created using *bivee_portal.sql*, which creates the schema with the portal configuration.

To configure database access, the file *liferay-portal-6.1.1-ce-ga2-bivee/ tomcat-7.0.27/webapps/ROOT/WEB-INF/classes/portal-ext.properties* must be edited, changing the following properties appropriately:

Property name	Value
jdbc.url	JDBC URL of MySQL instance, including hostname and port on which MySQL is installed, as well as database name in the format below: jdbc:mysql://HOSTNAME[:PORT]/DB_NAME
jdbc.username	Username of MySQL user with read/write access to database
jdbc.password	Password of MySQL user

To start the Web Portal launch the *startup* file which is in the directory *tomcat-7.0.27/bin* of the folder *liferay-portal-6.1.1-ce-ga2-bivee*.

Licensing information

All software developed is open-source and under the Apache License, Version 2.0.

All third party software is also open source, in particular:

- Liferay Portal Community Edition is distributed under the GNU Lesser General Public License version 2.1.
- MySQL is distributed under the GPL License.

ANNEX B - SINGLE-SIGN-ON FACTSHEET

Prototype references

Version	1.0
Availability	http://bivee-sp.eng.it/
Source control	svn://repo.nimbus-ware.com/BIVEE/WP6/D6.32/trunk/SSO
Contact person	Mauro Isaja (mauro.isaja@eng.it)

Prototype technical information

Nature	Web Application
Programming Language	Java
Development Framework	Java SE Development Kit 1.6 or later
Additional libraries	Jasig CAS v3.1 OpenSAML 1.1b OpenSAML 2.5 JBoss RESTEasy 2.3.5 or later
Application Server	J2EE servlet container (tested on Apache Tomcat 7)
Database	n.a.

Example of SAML response issued by LSA Service

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
    Version="2.0"
    IssueInstant="2012-12-27T16:20:22.664Z">
  <saml2:Issuer>http://lsa.mydomain.com/lsaservice</saml2:Issuer>
  <saml2:Subject>
    <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"
      NameQualifier="BIVEE">myuserid</saml2:NameID>
  </saml2:Subject>
  <saml2:Conditions>
    <saml2:OneTimeUse />
  </saml2:Conditions>
  <saml2:AttributeStatement>
    <saml2:Attribute Name="BIVEE_Name">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xs:string">John Doe</saml2:AttributeValue>
    </saml2:Attribute>
    <saml2:Attribute Name="BIVEE_Email">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xs:string">jdoe@acme.com</saml2:AttributeValue>
    </saml2:Attribute>
    <saml2:Attribute Name="BIVEE_Organization">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xs:string">BIVEE</saml2:AttributeValue>
    </saml2:Attribute>
  </saml2:AttributeStatement>
</saml2:Assertion>
```

```

                                xsi:type="xs:string">Acme Inc.</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute Name="BIVEE_VirtualOrganization">
  <saml2:AttributeValue xmlns:xs=http://www.w3.org/2001/XMLSchema
                        xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
                        xsi:type="xs:string">Acme Inc.</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute Name="Role">
  <saml2:AttributeValue xmlns:xs=http://www.w3.org/2001/XMLSchema
                        xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
                        xsi:type="xs:string">BIVEE_User</saml2:AttributeValue>
  <saml2:AttributeValue xmlns:xs=http://www.w3.org/2001/XMLSchema
                        xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
                        xsi:type="xs:string">PIKR_User</saml2:AttributeValue>
</saml2:Attribute>
</saml2:Attribute>
</saml2:AttributeStatement>
</saml2:Assertion>
```

SSO Service

Installation prerequisites

The prerequisites for installing the **SSO Service** components are:

- A J2EE servlet container version 6 or later (the prototype has been tested on Apache Tomcat 7)
- The URL of the LSA Service endpoint, and the corresponding authority name, to identify and configure the related Authentication Handler.

Installation instructions

The SSO Server web application can be downloaded from

http://bivee.eng.it:8780/web/guest/documents/-/document_library/view/43585 (path : BIVEE Project Portal/Document Library/WP6 – Architecture IT Platforms federation and services/Deliverables/Final/D6.32).

The package *D6.32_BIVEE Environment prototype – M24 -SSO.zip* contains the file *sso.war* that must be deployed on the application server.

deployerConfigContext.xml must be edited, configuring the specific *AuthenticationHandlers* invoked for authentication. Add the following beans:

- In the list of *property name="credentialsToPrincipalResolvers"*
bean class -> name of the specific *Authentication Handler class*
 - > *property name="URI"*: set the URL of the LSA Service endpoint
- In the map of *property name="authenticationHandlers"*
entry key -> name of the specific authority
 - > *bean class* = name of the specific *Authentication Handler class*
 - > *property name="URI"*: set the URL of the LSA Service endpoint.

Licensing information

All software developed is open-source and under the Apache License, Version 2.0.

All third party software is also open source, in particular:

- Jasig's CAS (Central Authentication Service): see <http://www.jasig.org/cas/license>

- OpenSAML is distributed under the Apache License, Version 2.0.
- JBoss RESTEasy is distributed under the Apache License, Version 2.0.

SSO Client

Installation pre-requisites

The prerequisites for installing the **SSO Client** are:

- a .NET framework or a PHP environment or a Java Virtual Machine.

Installation instructions

The standard version of the CAS Client can be downloaded at the address:
<http://downloads.jasig.org/cas-clients/cas-client-3.1-release.zip>.

It has to be configured according to Jasig's instructions by replacing the URL of the CAS Server with the URL of the deployed SSO Server instance.

Licensing information

All third party software is also open source, in particular:

- Jasig's CAS (Central Authentication Service): see <http://www.jasig.org/cas/license>

ANNEX C - VIRTUAL ENTERPRISE MANAGEMENT FACTSHEET

Prototype references

Version	1.0
Availability	http://bivee-sp.eng.it/
Source control	svn://repo.nimbus-ware.com/BIVEE/WP6/D6.32/trunk/VER
Contact person	Mauro Isaja (mauro.isaja@eng.it)

Prototype technical information

Nature	Web Application
Programming Language	Java
Development Framework	Java SE Development Kit 1.6 or later
Additional libraries	Liferay Community Edition v.6 Vaadin 7.1.0 MyBatis 3.1.1
Application Server	J2EE servlet container (tested on Apache Tomcat 7)
Database	MySQL v5.1 or later (other relational DBs should work)

VEM-WS API documentation

Verb	URI	Description
GET	v1/ves	retrieve list of all VEs

Normal Response Code: ok (200)

Error Response Code : error (500).

Example:

Input : `http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/ves/`

Output:

```
<VirtualEnterprises>
  <VirtualEnterprise>
    <Name>AIDIMA</Name>
    <ShortDescription>TECHNOLOGY      INSTITUTE      FURNITURE      WOOD      AND
PACKAGING</ShortDescription>
```

<LongDescription>AIDIMA is the Technology Institute on Furniture, Wood, Packaging and related industries. It is a non-for-profit association and its scope is both national and international. Although it was founded in November 1984, AIDIMA modified its statutes on July 1994 to open its membership and general assembly to packaging and transport companies.</LongDescription>

<Members>

<MemberName>Calilac S.A.</MemberName>

<MemberName>Coala</MemberName>

<MemberName>Ferrocuesta S.L.</MemberName>

<MemberName>Lasientu </MemberName>

<MemberName>Mazzaro Giulio S.R.L.</MemberName>

<MemberName>Muebles Romero SA</MemberName>

<MemberName>Recubrimientos Vecol S.L.</MemberName>

</Members>

</VirtualEnterprise>

<VirtualEnterprise>

<Name>LOCCIONI</Name>

<ShortDescription>ENERGY, ENVIRONMENT, HOME, HUMANCARE & MOBILITY SOLUTIONS</ShortDescription>

<Members></Members>

</VirtualEnterprise>

</VirtualEnterprises>

Verb	URI	Description
GET	v1/ves/{ve_name}	retrieve details of given VE

Normal Response Code: ok (200)

Error Response Code : error (500).

Example:

Input : http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/ves/AIDIMA

Output:

<VirtualEnterprises>

<VirtualEnterprise>

<Name>AIDIMA</Name>

<ShortDescription>TECHNOLOGY INSTITUTE FURNITURE WOOD AND PACKAGING</ShortDescription>

<LongDescription>AIDIMA is the Technology Institute on Furniture, Wood, Packaging and related industries. It is a non-for-profit association and its scope is both national and international. Although it was founded in November 1984, AIDIMA modified its statutes on July 1994 to open its membership and general assembly to packaging and transport companies.</LongDescription>

<Members>

<MemberName>Calilac S.A.</MemberName>

```

    <MemberName>Coala</MemberName>
    <MemberName>Ferrocuesta S.L.</MemberName>
    <MemberName>Lasientu </MemberName>
    <MemberName>Mazzaro Giulio S.R.L.</MemberName>
    <MemberName>Muebles Romero SA</MemberName>
    <MemberName>Recubrimientos Veccoal S.L.</MemberName>
  </Members>
</VirtualEnterprise>
</VirtualEnterprises>

```

Verb	URI	Description
GET	v1/members	retrieve list of all registered Enterprises

Normal Response Code: ok (200)

Error Response Code : error (500).

Example:

Input : http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/members

Output:

```

<Members>
  <EnterpriseMember>
    <MemberName>Calilac S.A.</MemberName>
    <VirtualEnterprises>
      <Name>AIDIMA</Name>
      <ShortDescription>TECHNOLOGY    INSTITUTE    FURNITURE    WOOD    AND
PACKAGING</ShortDescription>
      <LongDescription>AIDIMA is the Technology Institute on Furniture, Wood, Packaging
and related industries. It is a non-for-profit association and its scope is both national and
international. Although it was founded in November 1984, AIDIMA modified its statutes on
July 1994 to open its membership and general assembly to packaging and transport
companies.</LongDescription>
    </VirtualEnterprises>
  </EnterpriseMember>
  <EnterpriseMember>
    <MemberName>Coala</MemberName>
    <VirtualEnterprises>
      <Name>AIDIMA</Name>
      <ShortDescription>TECHNOLOGY    INSTITUTE    FURNITURE    WOOD    AND
PACKAGING</ShortDescription>
      <LongDescription>AIDIMA is the Technology Institute on Furniture, Wood, Packaging
and related industries. It is a non-for-profit association and its scope is both national and
international. Although it was founded in November 1984, AIDIMA modified its statutes on
July 1994 to open its membership and general assembly to packaging and transport
companies.</LongDescription>
    </VirtualEnterprises>
  </EnterpriseMember>

```

</VirtualEnterprises>

</Members>

Verb	URI	Description
GET	v1/members/{mb_name}/ves	retrieve list of all VE memberships for given Enterprise

Normal Response Code: ok (200)

Error Response Code : error (500).

Example:

Input :

http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/members/Lasientu/ves

Output:

<Members>

<EnterpriseMember>

<MemberName>Lasientu </MemberName>

<VirtualEnterprises>

<Name>AIDIMA</Name>

<ShortDescription>TECHNOLOGY INSTITUTE FURNITURE WOOD AND PACKAGING</ShortDescription>

<LongDescription>AIDIMA is the Technology Institute on Furniture, Wood, Packaging and related industries. It is a non-for-profit association and its scope is both national and international. Although it was founded in November 1984, AIDIMA modified its statutes on July 1994 to open its membership and general assembly to packaging and transport companies.</LongDescription>

</VirtualEnterprises>

</EnterpriseMember>

</Members>

Verb	URI	Description
GET	v1/direct/members/{mb_name}/ves?uid={username}&pwd={password}	same as above, without SSO login (user credentials registered on property file)

Normal Response Code: ok (200)

Error Response Code : error (500).

Example:

Input :

http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/members/Lasientu/ves?uid=user&pwd=password

Output:

```
<Members>
  <EnterpriseMember>
    <MemberName>Lasientu </MemberName>
    <VirtualEnterprises>
      <Name>AIDIMA</Name>
      <ShortDescription>TECHNOLOGY      INSTITUTE      FURNITURE      WOOD      AND
PACKAGING</ShortDescription>
      <LongDescription>AIDIMA is the Technology Institute on Furniture, Wood, Packaging
and related industries. It is a non-for-profit association and its scope is both national and
international. Although it was founded in November 1984, AIDIMA modified its statutes on
July 1994 to open its membership and general assembly to packaging and transport
companies.</LongDescription>
    </VirtualEnterprises>
  </EnterpriseMember>
</Members>
```

Verb	URI	Description
PUT	v1/ves/{ve_name}	create new VE

Normal Response Code: ok (200)

Error Response Code : error (500).

Example:

Input : http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/ves/LOCCIONI

```
<VirtualEnterprise>
  <ShortDescription>ENERGY,  ENVIRONMENT,  HOME,  HUMANCARE  &  MOBILITY
SOLUTIONS</ShortDescription>
  <LongDescription> </LongDescription>
  <Members></Members>
</VirtualEnterprise>
```

Verb	URI	Description
POST	v1/ves	update given VE (descriptions only):

Input : http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/ves

```
<VirtualEnterprise>
  <Name>LOCCIONI</Name>
  <ShortDescription>ENERGY, ENVIRONMENT, HOME, HUMANCARE & MOBILITY
  SOLUTIONS</ShortDescription>
  <LongDescription> </LongDescription>
</VirtualEnterprise>
```

Verb	URI	Description
DELETE	v1/ves/{ve_name}	delete given VE (cascade on members)

Normal Response Code: ok (200)

Error Response Code : error (500).

Example:

Input : http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/ves/AIDIMA

Output: n.a.

Verb	URI	Description
POST	v1/ves/{ve_name}/members	add new member to given VE

Normal Response Code: ok (200)

Error Response Code : error (500).

Example:

Input :
http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/ves/AIDIMA/members

```
<Members>
  <MemberName>Recubrimientos Vecol S.L.</MemberName>
</Members>
```


Output: n.a.

Verb	URI	Description
DELETE	v1/ves/{ve_name}/members/{mb_name}	remove given Enterprise from member list of given VE

Normal Response Code: ok (200)

Error Response Code : error (500).

Example:

Input :

http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/ves/AIDIMA/members/Coal
a

Output: n.a.

Verb	URI	Description
DELETE	v1/ves/{ve_name}/members	clear member list of given VE

Normal Response Code: ok (200)

Error Response Code : error (500).

Example:

Input :

http://HOSTNAME:[PORT]/VirtualEnterpriseManager_rest/rest/v1/ves/AIDIMA/members

Output: n.a.

Installation pre-requisites

The prerequisites for installing the VEM Web Application are:

- A Liferay portal where Vaadin version 7 is installed
- A MySQL instance version 5.1 or later (the application has been tested using MySQL Database engine, however any database with a JDBC driver available should work).

Installation instructions

The VEM web application and the relative database schema can be downloaded from:

http://bivee.eng.it:8780/web/guest/documents/-/document_library/view/43585 (path : BIVEE Project Portal/Document Library/WP6 – Architecture IT Platforms federation and services/Deliverables/Final/D6.32).

The package *D6.32_BIVEE Environment prototype – M24 -VirtualEnterpriseManager.zip* contains the files *VirtualEnterpriseManager.war* and *ver.sql*.

First of all, a new database must be created. A sample schema for MySQL is provided in *ver.sql*, which creates a basic schema and inserts some test data for users.

The file *VirtualEnterpriseManager.war* must be deployed on the Liferay portal.

To configure database access, the file *jdbc.properties* must be edited, changing the following properties appropriately:

Property name	Value
<code>jdbc.url</code>	JDBC URL of MySQL instance, including hostname and port on which MySQL is installed, as well as database name in the format below: <code>jdbc:mysql://HOSTNAME[:PORT]/DB_NAME</code>
<code>jdbc.username</code>	Username of MySQL user with read/write access to database
<code>jdbc.password</code>	Password of MySQL user

Licensing information

All software developed is open-source and under the Apache License, Version 2.0.

All third party software is also open source, in particular:

- Liferay Portal Community Edition is distributed under the GNU Lesser General Public License version 2.1.
- Vaadin is distributed under the Apache License, Version 2.0.
- MyBatis is distributed under the Apache License, Version 2.0.
- MySQL is distributed under the GPL License.

ANNEX D - RAW DATA HANDLER FACTSHEET

Prototype references

Version	1.0
Availability	http://bivee-sp.eng.it/
Source control	svn://repo.nimbus-ware.com/BIVEE/WP6/D6.32/trunk/RDH-WS
Contact person	Mauro Isaja (mauro.isaja@eng.it)

Prototype technical information

Nature	Web Application
Programming Language	Java
Development Framework	Java SE Development Kit 1.6 or later
Additional libraries	Spring Framework 4.0.0 Axis 2 1.6.2
Application Server	J2EE servlet container (tested on Apache Tomcat 7)
Database	MySQL v5.1 or later

RDH-WS API documentation

The formal WSDL definition of the RDH-WS API can be downloaded at :
bivee-sp.eng.it:8280/RDH-WS/services/RDHService?wsdl.

The RDH-WS Web Service exposes the following operations:

✓ *getAllKpi*

Input parameters :

not required

Output parameters :

array of 0-N *DscIndicator*

where *DscIndicator* is a complex structure:

DscIndicator

definedBy (string) – Name of the person that defined the KPI
hasTimeDimension (string) indicates if the KPI has the TIME dimension

indClass(string) - KPI class
indDescr(string) -KPI description
indName(string) -KPI name
rdhNormName(string) -KPI normalized name
unitOfMeasure(string) -unit of measure of the KPI.

✓ *getAllKpiValues*

Input parameters :

indicatorName (string) – normalized name of the KPI , required, must be valid (registered in RDH).

Output parameters :

array of 0-N *FactRecord*

where *FactRecord* is a complex structure:

FactRecord

factVal (decimal) - Measurement value.

factDate (date) - Date of measurement.

vEnt (string) - Virtual Enterprise.

ent (string) – Enterprise.

dimData (array of 0-N *DimElement* instances) – One entry for each reference to a dimension level.

and *DimElement* is a complex structure, too:

DimElement

levName (string) – Name of dimension level.

levRef (string) – Reference to a dimension level member.

✓ *getKpiLevels*

Input parameters :

indicatorName (string) – normalized name of the KPI , required, must be valid (registered in RDH).

Output parameters :

array of 0-N *DscIndicatorLevel*

where *DscIndicatorLevel* is a complex structure:

DscIndicatorLevel

dimNormName (string) – normalized name of the KPI dimension

dimensionClass (string) – class of the dimension
indName (string) – KPI name
isMandatory (string) – indicates if the dimension is mandatory
levelClass (string) – class of the level
levelNormName (string) – normalized name of the dimension level
levelPriority (int) - indicates the position of the level in the dimension hierarchy.

✓ *deleteKpiValue*

Input parameters :

indicatorName (string) – normalized name of the KPI for which delete a record, required, must be valid (registered in RDH).
factRecord (FactRecord) – data of the record to delete

where FactRecord is a complex structure:

FactRecord

factVal (decimal) - Measurement value, required, precision/scale 19,6.
factDate (date) - Date of measurement, required.
vEnt (string) - Virtual Enterprise, required, must be valid (registered in RDH).
ent (string) - Enterprise, optional, if provided must be valid (registered in RDH as member of *vEnt*).
dimData (array of 0-N *DimElement* instances) – One entry for each reference to a dimension level, all entries must be valid and uniquely named (*levName*)

and DimElement is a complex structure, too:

DimElement

levName (string) – Name of dimension level, required, must be valid (supported by the current KPI).
levRef (string) – Reference to a dimension level member, required, must be valid (registered in RDH).

Output parameters :

A boolean – true if the operation ended successfully, false otherwise.

✓ *putData*

Input parameters :

PutDataInput

sender (string) – identity of the sender, required, must be valid.

simLoad (string) - If value is "1", data sent with the call will go through validation but will not be committed into Data Storage.

factName (string) normalized name of the KPI for which to put data, required, must be valid (registered in RDH).

factData (array of 1-N *FactRecord* instances)

where *FactRecord* is a complex structure:

FactRecord

factVal (decimal) - Measurement value, required, precision/scale 19,6.

factDate (date) - Date of measurement, required.

vEnt (string) - Virtual Enterprise, required, must be valid (registered in RDH).

ent (string) - Enterprise, optional, if provided must be valid (registered in RDH as member of *vEnt*).

dimData (array of 0-N *DimElement* instances) – One entry for each reference to a dimension level, all entries must be valid and uniquely named (*levName*)

and *DimElement* is a complex structure, too:

DimElement

levName (string) – Name of dimension level, required, must be valid (supported by the current KPI).

levRef (string) – Reference to a dimension level member, required, must be valid (registered in RDH).

Output parameters :

A boolean – true if the operation ended successfully, false otherwise.

✓ *getLevelValues*

Input parameters :

levelName (string) – normalized name of the dimension level , required, must be valid (registered in RDH).

Output parameters :

array of 0-N *DscLevelValue*

where *DscLevelValue* is a complex structure:

DscLevelValue

name (string) – instance of dimension level

partOf(string) – indicates the parent of the member in the dimension hierarchy.

Installation pre-requisites

The prerequisites for installing the Raw Data Handler Service are:

- A Java servlet container version 6 or later (the prototype has been tested on Apache Tomcat 7)
- A MySQL instance version 5.1 or later.

Installation instructions

The Raw Data Handler Service and the relative database schema can be downloaded from http://bivee.eng.it:8780/web/guest/documents/-/document_library/view/43585 (path : BIVEE Project Portal/Document Library/WP6 – Architecture IT Platforms federation and services/Deliverables/Final/D6.32).

The package *D6.32_BIVEE Environment prototype – M24 -RawDataHandler.zip* contains the files *RDH-WS.war* and *bivee_rdh.sql*.

First of all, a new database must be created. The database schema for MySQL is provided in *bivee_rdh.sql*, which creates the schema and inserts the domains data.

To configure database access, the file *Spring-Module.xml* must be edited, changing the following properties appropriately:

Property name	Value
url	JDBC URL of MySQL instance, including hostname and port on which MySQL is installed, as well as database name in the format below: jdbc:mysql://HOSTNAME[:PORT]/DB_NAME
username	Username of MySQL user with read/write access to database
password	Password of MySQL user

The file *RDH-WS.war* must be deployed on the application server.

Licensing information

All software developed is open-source and under the Apache License, Version 2.0.

All third party software is also open source, in particular:

- Spring Framework is distributed under the Apache License, Version 2.0
- Axis 2 is distributed under the Apache License, Version 2.0.



Business Innovation in
Virtual Enterprise Environments

Bivee

Vision
Business
Customer
Performance
Ideas
Knowledge
Modelling
Originality
Improvement
Imagination
Inspiration
Process
Collaborate
Management
Mission
Semantic
KPI
Indicators
Supplier
Innovation
INNOVATION
BIVEE
Integration
Satisfaction
Environment
Enterprise
Success
Repository
Virtual
Creativity
Contribute
Ingenuity
Production
Waves
Handler
Phases
Factory
Standards
Control
Data
Inventiveness
Ontology
Organisation
Ecosystem
Resourcefulness
Solved
Problem
Advanced

