H2020-EINFRA-2015-1

VI-SEEM

VRE for regional Interdisciplinary communities in Southeast Europe and the Eastern Mediterranean

Deliverable D4.2

Description of the initial deployed data services

Author(s): Tamas Maray, Tamas Kazinczy (editors)

Status –Version: Final – e

Date: August 31, 2016

Distribution - Type: Public

Abstract: Deliverable D4.2 – Description of the initial set of deployed data services for the 3 user communities, listing also the potential usage scenarios and their documentation.

The VI-SEEM Consortium consists of:

GRNET	Coordinating Contractor	Greece
CYI	Contractor	Cyprus
IICT-BAS	Contractor	Bulgaria
IPB	Contractor	Serbia
NIIF	Contractor	Hungary
UVT	Contractor	Romania
UPT	Contractor	Albania
UNI BL	Contractor	Bosnia-Herzegovina
UKIM	Contractor	FYR of Macedonia
UOM	Contractor	Montenegro
RENAM	Contractor	Moldova (Republic of)
IIAP-NAS-RA	Contractor	Armenia
GRENA	Contractor	Georgia

BA	Contractor	Egypt
IUCC	Contractor	Israel
SESAME	Contractor	Jordan

The VI-SEEM project is funded by the European Commission under the Horizon 2020 e-Infrastructures grant agreement no. 675121.

Document Revision History

Date	Issue	Author/Editor/Contributor	Summary of main changes
05/07/2016	a	Tamas Maray	ToC
22/08/2016	b	Tamas Maray, Tamas Kazinczy	Draft version
28/08/2016	c	Tamas Maray, Tamas Kazinczy, Ioannis Liabotis, Vladimir Slavnic, Petar Jovanovic, Dusan Vudragovic, Emanouil Atanassov, Sofiya Ivanovska, Vladimir Dimitrov	Updated draft version
30/08/2016	d	Tamas Maray, Dusan Vudragovic	Further updates at several sections
31/08/2016	e	Ioannis Liabotis, Ognjen Prnjat, Valia Athanasaki	Final editing addressing the quality assurance comments and various other minor additions.

Preface

In the last decade, a number of initiatives were crucial for enabling high-quality research - by providing e-Infrastructure resources, application support and training - in both South East Europe (SEE) and Eastern Mediterranean (EM). They helped reduce the digital divide and brain drain in Europe, by ensuring access to regional e-Infrastructures to new member states, states on path to ascension, and states in European Neighborhood Policy area – in total 14 countries in SEE and 6 in EM.

This VI-SEEM project brings together these e-Infrastructures to build capacity and better utilize synergies, for an improved service provision within a unified Virtual Research Environment (VRE) for the inter-disciplinary scientific user communities in the combined SEE and EM regions (SEEM). The overall objective is to provide user-friendly integrated e-Infrastructure platform for regional cross-border Scientific Communities in Climatology, Life Sciences, and Cultural Heritage for the SEEM region; by linking compute, data, and visualization resources, as well as services, models, software and tools. This VRE aspires to provide the scientists and researchers with the support in full lifecycle of collaborative research: accessing and sharing relevant research data, using it with provided codes and tools to carry out new experiments and simulations on large-scale e-Infrastructures, and producing new knowledge and data - which can be stored and shared in the same VRE. Climatology and Life Science communities are directly relevant for Societal Challenges.

The driving ambition of this proposal is to maintain leadership in enabling e-Infrastructure based research and innovation in the region for the 3 strategic regional user communities: supporting multidisciplinary solutions, advancing their research, and bridging the development gap with the rest of Europe. The VI-SEEM consortium brings together e-Infrastructure operators and Scientific Communities in a common endeavor.

The overall objective is to provide user-friendly integrated e-Infrastructure platform for Scientific Communities in Climatology, Life Sciences, and Cultural Heritage for the SEEM region; by linking compute, data, and visualization resources, as well as services, software and tools.

The detailed objectives of the VI-SEEM project are:

1. Provide scientists with access to state of the art e-Infrastructure - computing, storage and connectivity resources - available in the region; and promote additional resources across the region.
2. Integrate the underlying e-Infrastructure layers with generic/standardised as well as domain-specific services for the region. The latter are leveraging on existing tools (including visualization) with additional features being co-developed and co-operated by the Scientific Communities and the e-Infrastructure providers, thus proving integrated VRE environments.
3. Promote capacity building in the region and foster interdisciplinary approaches.
4. Provide functions allowing for data management for the selected Scientific Communities, engage the full data management lifecycle, link data across the region, provide data interoperability across disciplines.

5. Provide adequate user support and training programmes for the user communities in the SEEM region.
6. Bring high level expertise in e-Infrastructure utilization to enable research activities of international standing in the selected fields of Climatology, Life Sciences and Cultural Heritage.

The VI-SEEM project kicked-off in October 2015 and is planned to be completed by September 2018. It is coordinated by GRNET with 15 contractors from Cyprus, Bulgaria, Serbia, Hungary, Romania, Albania, Bosnia-Herzegovina, FYR of Macedonia, Montenegro, Moldova (Republic of), Armenia, Georgia, Egypt, Israel, Jordan. The total budget is 3.300.000 €. The project is funded by the European Commission's Horizon 2020 Programme for Excellence in Science, e-Infrastructure.

The project plans to issue the following deliverables:

Del. no.	Deliverable name	Nature	Security	Planned Delivery
D1.1	Project management information system and “grant agreement” relationships	R	CO	M01
D1.2	3-Monthly progress report	R	CO	M03n *
D1.3a	First period progress reports	R	CO	M18
D1.3b	Final period progress reports	R	CO	M36
D2.1	Internal and external communication platform, docs repository and mailing lists	DEC	PU	M02
D2.2	Promotional package	DEC	PU	M04
D2.3	Dissemination and marketing plan	R	PU	M05
D2.4	Training plan	R	PU	M06
D2.5	Promotional package with updates	R	PU	M16
D2.6	1st Dissemination, training and marketing report	DEC	PU	M18
D2.7	2nd Dissemination, training and marketing report	R	PU	M35
D3.1	Infrastructure and services deployment plan	R	PU	M04
D3.2	Service registry, operational and service level monitoring	R	PU	M12
D3.3	Infrastructure overview, assessment and refinement plan	R	PU	M18
D3.4	VRE AAI Model and compatibility with other eInfrastructures	R	PU	M27
D3.5	Final infrastructure overview and assessment report	R	PU	M36
D4.1	Data sources and services deployment plan	R	PU	M06
D4.2	Description of the initial deployed data services	R	PU	M11
D4.3	Description of the final data platform available to	R	PU	M23

	VRE users			
D4.4	Final report on data, services, availability and usage	R	PU	M35
D5.1	Detailed technical implementation plan for VRE services and tools	R	PU	M04
D5.2	Data management plans	R	PU	M06
D5.3	User-oriented documentation and training material for VRE services	R	PU	M13
D5.4	Report on integrated services and the VRE platform	R	PU	M14
D5.5	Final report on integrated services and the VRE platform	R	PU	M36
D6.1	Framework for VRE resource and service provision	R	PU	M09
D6.2	1st Report of open calls and integration support	R	PU	M20
D6.3	Sustainability and business model	R	PU	M24
D6.4	2nd Report of open calls and integration support	R	PU	M36

Legend: R = Document, report, DEC = Websites, patent fillings, videos, etc., PU = Public, CO = Confidential, only for members of the consortium (including the Commission Services).

**n=1,2,3,…12*

Table of contents

References

[1] Project VI-SEEM-675121 - Annex I - Description of the action

[2] VI-SEEM deliverable D4.1, Data sources and services deployment plan

[3] VI-SEEM deliverable D3.1, Infrastructure and services deployment plan

[4] VI-SEEM deliverable D5.1, Detailed technical implementation plan for VRE services and tools

[5] VI-SEEM deliverable D5.2, Data management plans

[6] EUDAT, http://www.eudat.eu/

[7] CKAN, http://ckan.org

[8] OwnCloud, http://owncloud.org/

[9] GRNET HPC, http://hpc.grnet.gr/

[10] DSspace, http:/dsspace.org/

[11] iRODS, http://www.irods.org/

[12] handle.net, http://www.hdl.net/

[13] GRNET’s Handle service, ref: http://epic.grnet.gr/guides/overview/

List of Figures

List of Tables

Glossary

AAI	Authentication and Authorization Infrastructure
API	Application Programming Interface
CA	Certification Authority
CDI	Collaborative Data Infrastructure
CDMI	Cloud Data Management Interface
DSI	Data Storage Interface
DSS	Data Staging Script
EM	Eastern Mediterranean
EPIC	European Persistent Identifier Consortium
EUDAT	European Data Infrastructure
GB	Gigabyte
GPFS	General Parallel File System
GPU	Graphics Processing Unit
GRIB	GRIdded Binary
GridFTP	File Tranfer Protocol for Grid computing
HDFS	Hadoop Distributed File System
HPC	High Performance Computing
HTTP	Hypertext Transfer Protocol
iRODS	integrated Rule-Oriented Data System
iCAT	iRODS metadata catalogue
MB	Megabyte
mmCIF	macromolecular CIF (Crystallographic Information File)
MPI	Message Passing Interface
netCDF	network Common Data Form
NFS	Network File System
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
OPeNDAP	Open-source Project for a Network Data Access Protocol
pbdR	Programming with Big Data in R
PID	Persistent Identifiers
PEP	Policy Enforcement Point
PID	Persistent Identifier
PRACE	Partnership for Advanced Computing in Europe
REST	Representational State Transfer
SC	Scientific Community
SEE	South East European
SEEM	South East Europe and Eastern Mediterranean

SLA	Service Level Agreement
SPMD	Simple program, Multiple data
SQL	Structured Query Language
TB	Terabyte
UI	User Interface
UUID	Universally Unique Identifier
VAS	VI-SEEM Archival Service
VDAS	VI-SEEM Data Analysis Service
VDDS	VI-SEEM Data Discovery Service
VI-SEEM	VRE for regional Interdisciplinary communities in Southeast Europe and the Eastern Mediterranean
VLS	VI-SEEM Work Storage Space / Local Storage and Data Staging
VM	Virtual Machine
VRS	VI-SEEM Repository Service
VRE	Virtual Research Environment
VSS	VI-SEEM Simple Storage Service
WP	Work Package

Executive summary

What is the focus of this Deliverable?

D4.2 deliverable is focusing on the technical details of the initial deployment of the VI-SEEM generic data services, the plans for the final production deployment of such services, as well as on the corresponding potential usage scenarios.

What is next in the process to deliver the VI-SEEM results?

All of the VI-SEEM generic data services defined in D4.1 have been deployed at one or more VI-SEEM consortium partners’ sites, at least in the stage of “initial service deployment”, enabling initial usage, further testing, refining and tuning of the configuration and further integrating the service to the VI-SEEM ecosystem and the Virtual Research Environment (VRE). According to the development roadmap, some of the data services (e.g. the archival service and the local storage and data staging service) will be fully integrated with the AAI as the next step. The provided data services together with e-Infrastructure processing services will create an easy to use, fully interoperable, reliable data management environment for the users from different scientific communities.

The results and conclusions from this deliverable will be used by the following activities:

- WP4.2 - Data access, preservation and re-use
- WP4.3 - Data collection and provisioning
- WP4.4 - Data analysis
- WP5.3 - Development of the VRE platform
- WP5.4 - Overall integration of services

What are the deliverable contents?

The deliverable gives an overview of the VI-SEEM data services and discusses the roadmap of the deployment of each service. This is followed by a detailed technical description of the deployment of each service, namely the VI-SEEM Simple Storage Service (VSS), the VI-SEEM repository service (VRS), the VI-SEEM archival service (VAS), the VI-SEEM work storage space / local storage and data staging service (VLS), the VI-SEEM data discovery service (VDDS), and the VI-SEEM data analysis service (VDAS). The descriptions of the different services have a common structure: HW/SW information, definition of the service endpoint where the service is available, technical description and remarks about the installation, configuration and usage. The deliverable describes the supplementary persistent identification (PID) service as well as the integration of the data services with the VI-SEEM AAI infrastructure. Moreover, further development steps necessary to bring the data services to their final stage are also discussed. The deliverable also explains the potential usage scenarios of the VI-SEEM generic data services, keeping in mind the specific requirements of the VI-SEEM application areas.

Conclusions and recommendations

All of the generic data services defined in D4.1 have been already deployed at different consortium partners’ sites as “initial services”, and they are ready for usage, further testing, integration and support for the selected pilot applications.

Potential usage scenarios of the data services have been elaborated. Such scenarios and the corresponding service provided are listed below:

- The storage of simple data files that need to be exchanged among scientists. The VI-SEEM Simple Storage service will be provided to cater for this.
- The long-time storage of scientific data objects, scientific papers, videos, etc, together with their metadata allowing search, cataloguing and access to the rest of the regional and even worldwide communities. The VI-SEEM Repository service will be used for this purpose.
- The archiving of large and valuable data sets ensuring the replication of such data and their preservation in disk or tape storage. The VI-SEEM Data archiving service will be used to cater for this.
- The transfer of data between data centers for the efficient utilization of computational resources spread across the regional infrastructure. The VI-SEEM work storage / staging service will be used for this purpose.
- The analysis of big scientific data sets that require specialized software and hardware facilities in order to produce new meta data valuable for further analysis. The VI-SEEM data analysis service caters for this use case.
- The centralized search for (and access to) data sets that are stored in different repositories, or even application specific services. The VI-SEEM Data search and the issuing of PIDs for the data sets will be used.
- Finally secure and authenticated access is required to ensure accountability and auditing as well as prevention of unauthorized access to the data services. Integration of the VI-SEEM generic data services with the VI-SEEM AAI facilitates this.

The next phase of the activity will be focusing on bring together the data services and other services of the VI-SEEM ecosystem into a common platform, to further develop the user interfaces with VI-SEEM design, as well as to prepare the services for the final production phase.

1. Introduction

D4.2 provides a description of the generic VI-SEEM data services and the initial deployment work done in the context of WP4 of the project.

In Chapter 2 a short overview of our data services and the decisions made in D4.1 [2] is presented. Next, in Chapter 3 the deployment roadmap is discussed. In Chapter 4, technical details of deployment of the specific services are presented. Chapter 5 gives further information about supplementary developments, like the persistent identifier service and the integration of the data services with the VI-SEEM authentication and authorization infrastructure. Chapter 6 – discussion of potential usage scenarios – gives a hint about how to use the different data services for the benefit of the VRE users. Chapter 7 describes the further developments needed to bring the VI-SEEM data services to their full potential and complete stage in the next phase of the project, while the Annexes provide the precise list of steps to perform in order to set up the most important services.

2. VI-SEEM generic data services portfolio

In D4.1 ("Data sources and services deployment plan") [2] the decisions were made to deploy the following specific data services:

- VI-SEEM simple storage service (VSS)
- VI-SEEM repository service (VRS)
- VI-SEEM archival service (VAS)
- VI-SEEM work storage space / local storage and data staging (VLS)
- VI-SEEM data discovery service (VDDS)
- VI-SEEM data analysis service (VDAS)

The approach to deployment was also planned: a three step process, where the first step is an initial deployment at selected partner’s site(s). This should be followed by deployments of specific further capabilities at corresponding sites according to deployment scenarios defined in the first step. Finally, the third step should enable services for widespread use by scientific communities and also make specific application support available.

The following subsections provide a summary of the generic VI-SEEM data services that have been deployed.

2.1. VI-SEEM simple storage service

The VI-SEEM Simple Storage Service (VSS) is a secure data storage service provided to VI-SEEM users for storing and sharing research data as well as keeping it synchronized across different computers. Data sharing will be possible with other registered VI-SEEM users or with anyone else by using public links which can be protected with passwords if needed. The initial deployment plan - discussed in D4.1 – foresees to provide 50 GB of storage per user.

2.2. VI-SEEM repository service

The main storage service that will allow the users of the VI-SEEM VRE to deposit and share data is the VI-SEEM Repository Service (VRS). Such a repository in VI-SEEM is the main repository for hosting the "Regional Community Datasets" and therefore provide a component to host one of the main services of the VRE as specified in D5.1. It can also be used to host publications and their associated data as well as software or references to software and workflows, used to generate such data and publications.

The VRS is also the service for storing simplified data formats such as images, videos or others suitable also for the general public. The VRS is therefore the platform to host all of the types of data specified in the VI-SEEM data management plan, D5.2 [5], when users consider it suitable i.e. for sharing.

2.3. VI-SEEM archival service

Data archiving is the practice of moving data that is no longer being used or are being used on a less frequent fashion into a separate storage service. It is a single set or a collection of historical records specifically selected for longer term retention and future reference. Additionally, data archives contain data that are important for future reference or it is important to preserve them for regulatory and audit purposes. In science archived data are important for future reference and reproducibility of scientific simulations. Data archives are indexed and have search capabilities so that files and parts of files can be easily located and retrieved.

The VI-SEEM Archival Service (VAS) offers the capability to preserve data sets. It also offers safe data replication. For this - in accordance with D4.1 - a VI-SEEM iRODS Federation (VIF) will be formed with several member zones. Federations allow controlled access where the remote partner is subject to the constraints of the local zone's policies. A zone in iRODS is an administrative domain. Federated zones (i.e. zones that form a federation) in the VI-SEEM VRE will put such policies in place to implement safe data replication.

2.4. VI-SEEM work storage space / local storage and data staging

Twelve partners have offered work storage space. These work areas will hold data needed for computation. As discussed in deliverable D4.1, gridFTP will be utilized to move data around, so all these partners deployed gridFTP servers. Partners who will also provide VI-SEEM archival service need to deploy an additional gridFTP server instance too, with the iRODS DSI component that will be able to interact with iRODS. More specifically, the gridFTP server with the iRODS DSI component will only access material in the VI-SEEM iRODS namespace.

This means that local storage spaces that are not to be part of any VI-SEEM iRODS zone will be accessible only by a non-DSI gridFTP server.

2.5. VI-SEEM data discovery service

VI-SEEM data discovery service is a service provided to VI-SEEM users for flexible searching for data discovery. It is based on B2FIND technology developed as a task in the ongoing EUDAT project [6] and the open-source CKAN [7]. CKAN is a powerful data management system that provides publishing, sharing, searching and can use almost any type of data and metadata.

2.6. VI-SEEM data analysis service

Apache Hadoop was chosen as the main platform for the VI-SEEM data analysis service. The main components of the current setup are the distributed file system HDFS, the resource manager YARN and MapReduce as the analysis framework.

3. Deployment roadmap of the generic data services

As already mentioned earlier, the deployment roadmap of the generic VI-SEEM data services is outlined as a three-step process:

- Initial deployment (at selected partners’ sites)
- Complete setup (all sites running their respective capabilities)
- Final data platform (all services in place with full integration to the VRE environment)

Figure 1. VI-SEEM data services deployment phases

These steps are outlined in the following chapters.

3.1. Initial deployment

Details of the first deployment phase (initial deployment), that has been already completed, are outlined in Table 1 in a what/who fashion with remarks about a particular deployment if applicable.

Note that an administrative domain in iRODS is called a zone. Collaboration in iRODS could be managed both in the local zone and across separate (remote) zones the latter being called a federation. Federations allow controlled access where the remote partner is subject to the constraints of the local zone's policies. Regarding the VI-SEEM VRE, it is envisaged that multiple data centers will join together in a federation, and safe data replication amongst them will be implemented with a set of iRODS policies. All partners providing VAS will be part of the VI-SEEM iRODS federation, i.e. all partners are co-operating with each other.

Service	Partner	Remarks
VSS	IPB	50GB per user, including initial integration with VI-SEEM AAI
VRS	GRNET	10TB initial capacity, including full integration with VI-SEEM AAI
VAS	GRNET	separate iRODS zone; federated with IPB and NIIF
VAS	IPB	separate iRODS zone; federated with GRNET and NIIF
VAS	NIIF	separate iRODS zone; federated with GRNET and IPB
VLS	BA	
VLS	CYI	
VLS	GRNET	one additional instance required for iRODS DSI
VLS	IICT-BAS	
VLS	IIAP-NAS	
VLS	IPB	one additional instance required for iRODS DSI
VLS	IUCC	
VLS	NIIF	one additional instance required for iRODS DSI
VLS	RENAM	
VLS	SESAME	
VLS	UKIM	
VLS	UNI BL	
VLS	UoM	
VLS	UPT	
VLS	UVT	
VDDS	IICT-BAS	data source integration expected by M12
VDAS	IPB	

Table 1. Initial deployment of VI-SEEM data services

3.2. Complete setup

Details of the second deployment phase (complete setup) are outlined in table 2 in a what/who/when fashion with remarks about a particular deployment if applicable. Deployment due date is expressed in project month.

Service	Partner	Due date (project month)	Remarks
VRS	CYI/NIIF	M16	on demand (either of both)
VAS	GRNET	M16	fedarated iRODS zone with replication policy
VAS	IPB	M16	fedarated iRODS zone with replication policy
VAS	NIIF	M16	fedarated iRODS zone with replication policy
VAS	IUCC	M16	fedarated iRODS zone with replication policy
VAS	any partner to provide a separate zone	M16	separate iRODS zone + federation
VAS	any partner to join	M16	resource server

	an existing zone		
VLS	any partner to provide VAS	M16	additional instance for iRODS DSI
VDAS	IPB	M16	optional additional features
VDAS	IICT-BAS	M16	deployment + optional additional features

Table 2. Complete setup of VI-SEEM data services

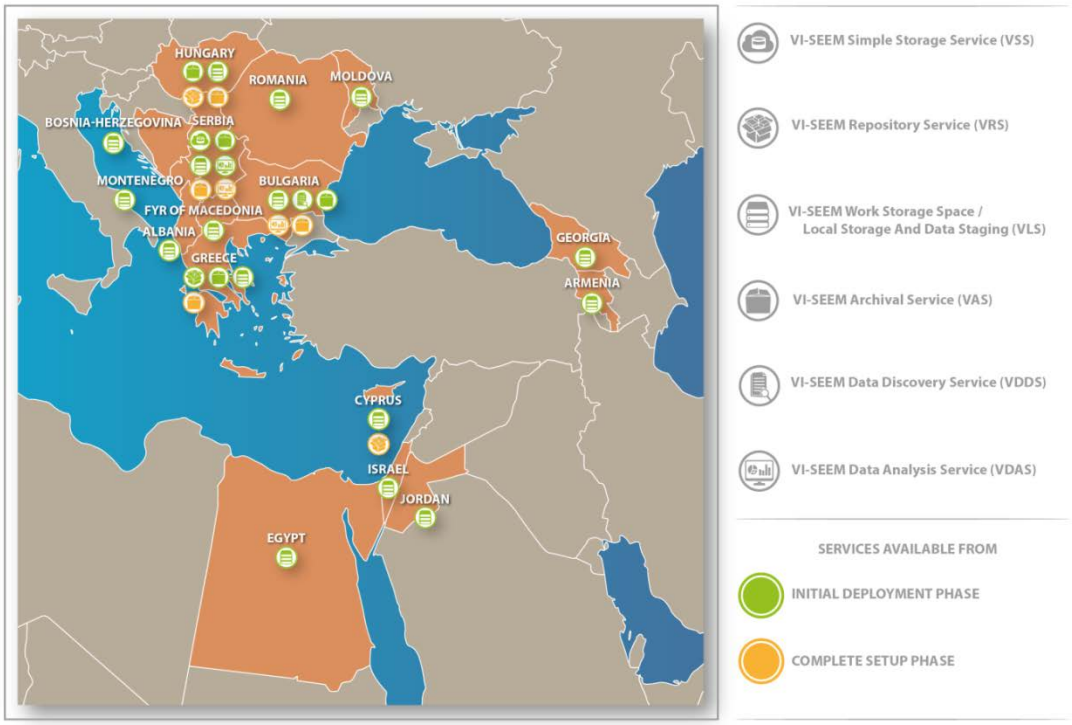

Figure 2. Geographical location and availability of VI-SEEM data services

3.3. Final data platform

The final phase of the deployment roadmap envisages bringing together above-mentioned data services under the VI-SEEM VRE umbrella and the provision of a wide range of datasets produced by the selected applications, and that adequate support is available to help the user communities to utilize them. The final data platform is scheduled to put in operation by M22-23. (Milestone M4.3, description to be available in deliverable D4.3).

4. Deployment of specific services

In this section, detailed information for each VI-SEEM data service is presented as follows:

- HW/SW information
- Service endpoint
- Detailed description
- Notable remarks about implementation

Deployment guides are found in the Annexes of this document.

4.1. VI-SEEM simple storage service

VI-SEEM simple storage service is a secure data service provided to VI-SEEM users for storing and sharing research data and keeping it synchronized across different computers. It is based on ownCloud [8] technology.

4.1.1. HW/SW information

VI-SEEM simple storage service instance is hosted at VI-SEEM partner site Institute of Physics Belgrade and is installed on a server with 2 six-core Intel E5-2620 processors (with HT enabled), 64GB of RAM and 16TB of storage space in RAID-6 array and additional 2TB in RAID-1 array. That storage space is expected to be sufficient for needs of VI-SEEM users as plan was to have 50 GB of storage per user.

4.1.2. Service endpoint

The simple storage service is hosted at the Institute of Physics Belgrade on machine ipbbox.ipb.ac.rs. The service is provided to the VI-SEEM community under the URL: https://simplestorage.vi-seem.eu

4.1.3. Detailed description

VI-SEEM simple storage service provides VI-SEEM users with the possibility to store data and share it with other registered users or with anyone by using public links that can be protected with password if needed. Interaction with the data can be done through any browser, ownCloud desktop sync client applications or through WebDAV client. Desktop sync client applications enable synchronization of data in VI-SEEM simple storage service with data on user’s computer. Sync clients provide possibility for selective syncing of folders and they are available for all major operating systems: Windows, Mac and Linux. WebDAV client can be used to mount user’s VI-SEEM simple sharing storage service folder as a drive on local computer and this way of interaction with the stored files is also available for all mentioned operating systems.

As ownCloud has a modular architecture it is possible to extend its feature with the so-called apps developed by ownCloud community. OwnCloud apps can provide different additional functionalities to users: using and sharing calendar and contacts, collaborative editing of documents in various formats, photo galleries, or even playing video and audio files, just to name a few. OwnCloud website hosts repository of already developed apps (https://apps.owncloud.com/) that are available to administrators of OwnCloud servers. Depending on the needs of VI-SEEM users, additional apps can be added to the instance of VI-SEEM simple storage service to extend its functionality.

Another standard feature of ownCloud are groups. Different groups can be defined and users can be assigned to them, which will give them access to all file shares that belong to those groups. Groups in VI-SEEM simple storage service can be defined in different ways, for example, based on specific VI-SEEM applications, or whole research communities.

4.1.4. Remarks about the implementation

The VI-SEEM simple storage service has been integrated with the VI-SEEM AAI infrastructure. Details are described in Chapter 5.2.

4.2. VI-SEEM repository service

The main storage service that will allow the users of the VI-SEEM VRE to deposit and share data is the VI-SEEM Repository Service. Such a repository in VI-SEEM is the main repository for hosting the "Regional Community Datasets" and therefore provide a component to host one of the main services of the VRE as specified in D5.1. It can also be used to host publications and their associated data as well as software or references to software and workflows, used to generate such data and publications. The VRS is also the service for storing simplified data formats such as images, videos or others suitable also for the general public. The VRS is therefore the platform to host all of the types of data specified in the VI-SEEM data management plan, D5.2, when users consider it suitable i.e. for sharing.

4.2.1. HW/SW information

GRNET has deployed the service as a VM provided by the infrastructure of GRNET’s HPC [9] service. VI-SEEM Repository is connected to the GEANT network and therefore the research communities via 2x10Gbit/s connections. The bit stream store is connected to ARIS [9] parallel file system (GPFS) that has a 1 PB of disk capacity. This storage capacity is shared with other services provided by GRNET to the project and at the national level. The available VI-SEEM repository storage capacity will depend on demand and the usage of the capacity of other storage services offered by GRNET not exceeding GRNET’s storage commitments as specified in VI-SEEM D3.1 [3], i.e. 50TB of disk space.

4.2.2. Service endpoint

The VI-SEEM Repository has been deployed by GRNET and is available for all users at https://repo.vi-seem.eu/

4.2.3. Detailed description

The VI-SEEM repository is implemented using DSpace [10]. DSpace open source software is a turnkey repository application used by more than 1000+ organizations and institutions worldwide to provide durable access to digital resources. DSpace preserves and enables easy and open access to all types of digital content including text, images, moving images, mpegs and data sets.

From the architecture point of view DSpace is organized into three layers, each of which consists of a number of components. Those three layers are:

- The application layer: The application layer contains components that communicate with the world outside of the individual DSpace installation, for example the Web user interface and the Open Archives Initiative protocol for metadata harvesting service. This layer is composed of several components such as the web user interface, OAI-PMH Data Provider, the Sub-Community Management system, etc.
- The Business Logic Layer: The business logic layer deals with managing the content of the archive, users of the archive (e-people), authorization, and workflow. This layer is composed by several important components for the operation of the repository such as the content management API, the workflow system, the administration toolkit, the authorization framework the history recorder, etc.
- The Storage Layer: The storage layer is responsible for physical storage of metadata and content. This layer is composed by a relational database for storage of metadata and the bit stream store.

4.2.4. Remarks about the implementation

PID Integration: the VI-SEEM repository will be soon integrated with the GRNET Handle service [12]. GRNET Handle service is a service dedicated to provide, resolve and mint persistent identifiers (PID). DSpace requires that a persistent identifier is assigned to each digital object (Item, Collection, Community). Because the developers wanted a solution which will work for a very long time, the identifier system had to be independent of any underlying network protocols, such as HTTP.

DSpace uses the Handle System from CNRI (Corporation for National Research Initiatives) as the persistent identifier for each digital object. Handles are resolved to actual URLs via a resolution service. The Handle resolver is an open-source system. Handles in DSpace (and elsewhere) are currently implemented as HTTP URIs, but can also be modified to work with future protocols. The Handle system is also able to support existing bibliographic identifiers such as ISBN or ISSN.

To implement support for other registration agencies, we have to develop a Java class that implements the interface DOIConnector using the DataCiteConnector as an

example. We configure the system to use our own DOIConnector when configuring the IdentifierService instead of the DataCiteConnector. For more information on the integration implementation methodology one can visit:

http://dspace.org/sites/dspace.org/files/archive/1_5_2Documentation/ch02.html#N1041B

4.3. VI-SEEM archival service

Data archiving is the practice of moving data that is no longer being used or are being used on a less frequent fashion into a separate storage device. It is a single set or a collection of historical records specifically selected for long term retention and future reference. Additionally, data archives contain data that are important for future reference or it is important to preserve them for regulatory and audit purposes. In science archived data are important for future reference and reproducibility of scientific simulations. Data archives are indexed and have search capabilities so that files and parts of files can be easily located and retrieved.

The initial deployment of the VI-SEEM archival service was planned to comprise three distinct iRODs zones at GRNET, IPB and NIIF but at the end a fourth zone is deployed at IICT-BAS as well.

4.3.1. HW/SW information

NIIF

PostgreSQL server

Hardware
- Server Type : VM
- CPU : 8 vCPU*
  - *: Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
- Memory : 16GB
- Storage : 160GB local
- Network : 10Gbit

Software
- OS : Ubuntu 14.04.5 LTS
- PostgreSQL : 9.5.4-1

iCAT server

Hardware
- Server Type : VM
- CPU : 8 vCPU*
  - *: Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
- Memory # 16GB
- Storage # 160GB local
- Network # 10Gbit

Software
- OS : Ubuntu 14.04.5 LTS
- iRODS iCAT server : 4.1.9
- iRODS Postgres Database Plugin : 1.9

iRODS - Runtime Library : 4.1.9
iRODS - Development Library : 4.1.9
iRODS GSI Auth Plugin : 1.3
Globus Toolkit : 6.0

Resource server

Hardware
Server Type : VM
CPU : 4 vCPU*
*: Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
Memory : 8GB
Storage : 80GB local + 50TB external
Network : 10Gbit

Software
OS : Ubuntu 14.04.5 LTS
iRODS resource server : 4.1.9

IICT-BAS

Hardware
Server Type : Physical
CPU : 2x Intel Xeon E5430, 4 cores, 2.6GHz, 12 MB cache
Memory : 16GB
Storage : 160GB local
Network : 1Gbit

Software
OS : Ubuntu 14.04.4 LTS, x86_64
PostgreSQL : 9.3.13
iRODS iCAT server : v4.1.9
iRODS Postgres Database Plugin : 1.9

GRNET

Hardware

Server Type : Physical
CPU : 2x Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00GHz
Memory : 128GB
Storage : 1.6TB local
Network : 10Gbit

Software
OS : Red Hat Enterprise Linux Server release 6.7 (Santiago)
PostgreSQL : 8.4.20-6
iRODS iCAT server : v4.1.8
iRODS Postgres Database Plugin : 1.8.0

IPB

Hardware
Server Type : Physical
CPU : quad-core Intel Xeon E3-1220 v3 @ at 3.1 GHz
Memory # 4GB

Storage # 500GB local (RAID-1 mirror array) + 30TB external

Software

iRODS iCAT server : 4.1.8

4.3.2. Service endpoint

VI-SEEM archival service instances deployed at 4 different sites can be reached at:

Zone	Host	Port (default:1247)	Data transfer port range (default: 20000-20199)
GRNET	irods.vi-seem.eu	default	default
IICT-BAS	icat.avitohol.acad.bg	dafault	default
IPB	Irods.ipb.ac.rs	default	default
NIIF	niificat.niif.hu	default	default

4.3.3. Detailed description

The VI-SEEM Archival Service is implemented using iRODS [11]. Each of the partners in the initially deployed federation has a separate zone configured. These zones are then connected with each other to form a federation. This allows controlled access to resources at remote partners, i.e. local policies apply for remote zone users. Creating a federation is a multi-step process. Partners have to define remote zones in their respective iCAT and then create remote users before granting them access to the resources. In practice, users should have a home zone and they would be recognized as remote users in other zones of the federation. One of the goals in VI-SEEM for the VAS is safe data replication. This will be implemented by a set of iRODS policies. In practice, when data ingest occurs, policies will assure that the data object is replicated amongst zones according to existing policies.

Policies for safe data replication may be regulated at different levels, e.g.:

- data in ZoneA replicated to some/all other zones
- data on a specific path (like '/ZoneA/home/userX/...') replicated only inside the zone where it resides

These policies have to be implemented according to SCs' needs, especially in cases where data placement is subject to restrictions, i.e. could only be preserved on resources of a specific zone.

4.3.4. Remarks about the implementation

It is essential to integrate the VI-SEEM Archival Service with the work storage space/local storage service for those sites that provide both, since a potential usage scenario is to move research data from iRODS to local storage as part of a pre-compute step or computation results back from local storage to iRODS.

This integration is done installing the iRODS DSI plugin for gridFTP server. The iRODS DSI plugin makes it possible for gridFTP users to access iRODS through gridFTP by tools such as globus-url-copy. Note that if the iRODS DSI plugin is in use then only

the iRODS namespace is accessible by that particular gridFTP server instance. Therefore, partners offering VAS should provide an additional gridFTP server instance for use with iRODS.

Several iRODS zones will form together a federation in the VI-SEEM VRE. This not only will provide the capability of safe data replication, but also allows better availability and performance for the users of the VI-SEEM VRE as each zone has its own iCAT server so it is less likely that one of them becomes a bottleneck. Also, this distributed setup has the potential of a "connect to nearest zone" approach for users, e.g. this way, computation results could be staged back to the nearest iRODS zone for best performance.

We have tested the gridFTP integration using the EUDAT’s B2STAGE plugin that enables gridFTP transfers to and from iRODS managed storage resources. To use the gridFTP access, users will have to have a valid grid certificate and an account on the iRODS which is mapped to the certificate.

4.4. VI-SEEM work storage space / local storage and data staging

4.4.1. HW/SW and service endpoint information

Details of the gridFTP server implementations at different VI-SEEM sites are shown below.

Partner	HW+SW spec	gridFTP version	service endpoint	size
NIIF	HW: 2x Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz 64391MB RAM SW: Red Hat Enterprise Linux Server release 6.6 (Santiago)	globus-gridftp-server; x86_64; v9.3	login.debrecen2.hpc.niif.hu Port: 2811 (default)	6TB
GRNET	HW: VM, 2x QEMU Virtual CPU, 4 GB RAM SW: CentOS release 6.8	globus-gridftp-server-11.1-1.el6+gt6.x86_64	gftp.aris.grnet.gr Port: 2811 (default)	1PB
CYI	HW: 2x Intel(R) Xeon(R) CPU X5650 @ 2.67GHz 48 GB RAM SW: CentOS release	GridFTP Server 9.1	login2.cytera.cyi.ac.cy Port: 2812 (default)	262TB

	6.6 (Final)			
IICT-BAS	HW: 4x Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz 15949MB RAM SW: CentosOS release 6.8	globus-gridftp-server; x86_64; v11.1	gftp.avitohol.acad.bg Port: 2811 (default)	5TB
IPB	HW: 16 x Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz 32 GB RAM SW: Scientific Linux release 6.5 (Carbon)	globus-gridftp-server; x86_64; v10.4	paradox.ipb.ac.rs Port:2811	66TB
UVT	HW: 1 x Intel Core i7 9xx @ 2GHz SW: Ubuntu 16.04.1 LTS	globus-gridftp-server; x86_64; v11.1	gridftp.viseem.hpc.uvt.ro Port: 2811 (default)	5TB
UKIM	HW: 2x Intel(R) Xeon(R) CPU L5640 @ 2.26GHz 24167MB RAM SW: CentOS Linux 6.3 (Carbon)	globus-gridftp-server; x86_64; v9.1-1	se.hpgcc.finki.ukim.mk Port: 2811 (default)	2TB
BA	HW: Cluster of 10 nodes, each with Intel i5-Intel(R) Core(TM) i5-3470S CPU @ 2.90GHz, 7880 MB RAM SW: Debian GNU/Linux 8.5 (jessie) amd64	globus-gridftp-server; x86_64; v7.11	aa112642.archive.bibalex.org Port: 2811 (default) aa112643.archive.bibalex.org Port: 2811 (default)	100 TB
RENAM	HW: 1xQuadCore Intel Xeon E5310, 1600 Mhz,14 GB RAM SW: CentOS 7	globus-gridftp-server; x86_64; v9.3	gridftp.renam.md Port: 2811 (default)	2TB
IIAP-NAS	HW: 2x Intel(R) Xeon(R)	globus-gridftp-server; x86_64;	gridgtp.grid.am Port: 2811 (default)	3 TB

	CPU E5-2620 v2 @ 2.10GHz 32 GB RAM SW: Centos 7 Server			
GRENA	HW: Virtual Machine 4x 2.8GHz Cores 8GB RAM SW: Scientific Linux release 6.8 (Carbon)	globus-gridftp-server; x86_64; v9.4	se.sg.grena.ge Port: 2811 (default)	2TB
IUCC	TBD	TBD	TBD	5TB

4.4.2. Description

As already discussed in D4.1, efficient computing at VI-SEEM partners offering grid and/or HPC facility requires quasi-local storage for short-term workloads on one hand, and a data staging capability on the other.

While the former will be provided as is - ie. existing solutions that were already implemented at the partners – the latter is expected to be provided by all sites by a separate gridFTP server for each of them.

4.4.3. Remarks about the implementation

Sites hosting VAS as well, have to deploy an additional gridFTP server instance with the iRODS DSI plugin installed. This will provide integration of VAS and VLS as it makes VAS accessible for data staging.

4.5. VI-SEEM data discovery service

VI-SEEM data discovery service is a service provided to VI-SEEM users for flexible searching for data discovery.

4.5.1. HW/SW information

The VI-SEEM data discovery service instance is deployed at IICT-BAS on a physical server with the following hardware and software characteristics.

Hardware:

2 four-cores Intel Xeon E5430 processors
16 GB of RAM, FB DDR2 with ECC

160 GB of internal storage
2 to 5 TB of external storage space
1 Gb/s Ethernet connectivity

Software configuration:

OS Ubuntu 14, x86_64, kernel 4.2.0-42-generic
Python 2.7.6
CKAN 2.5
B2FIND/ searchB2FIND.py and related scripts

4.5.2. Service endpoint

The service is available at: icat.avitohol.acad.bg and https://discovery.vi-seem.eu

4.5.3. Detailed description

The VI-SEEM data discovery service uses metadata mapped onto standardized facets and which can be collected from various research and other repositories and provides VI-SEEM users with the possibility for flexible search and browsing.

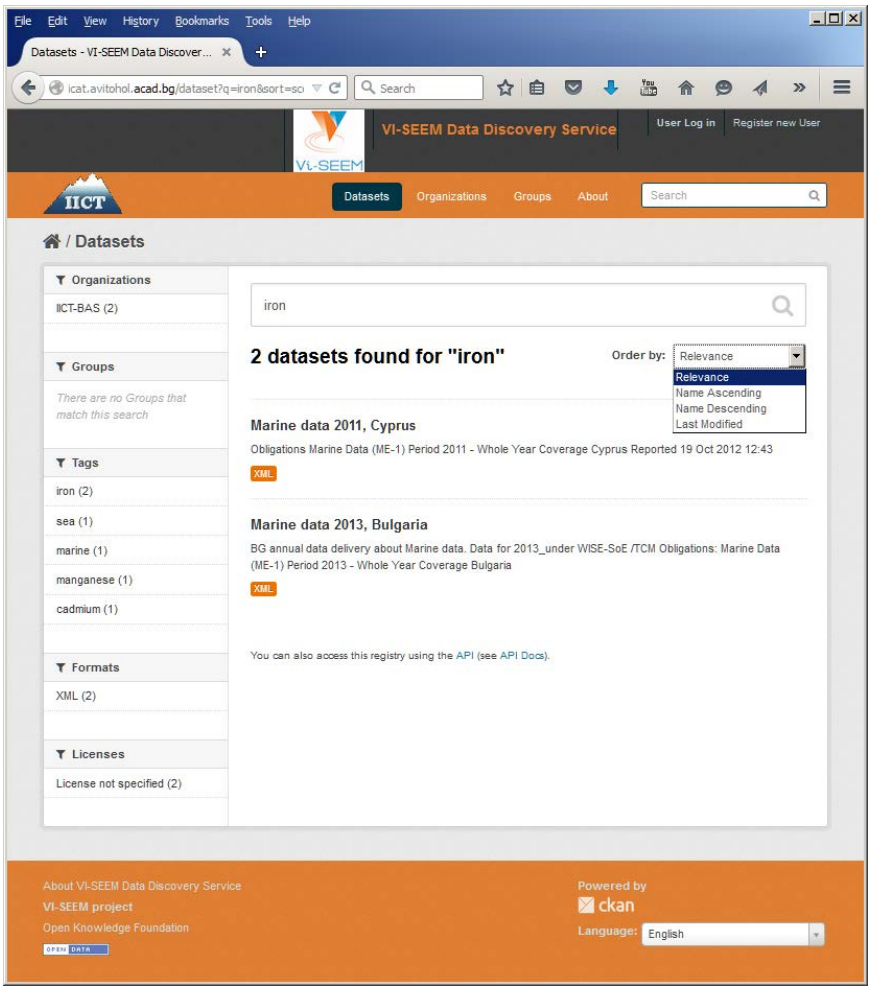

Figure 3. VI-SEEM data discovery service home page (with test datasets)

It is possible to search for keywords, partial phrases, creator, organization, publisher, time of publishing, versions, tags, research areas and communities etc. The results are presented in a user friendly form. In order to enter metadata to the service, the users need to be registered. Searching of a particular dataset is performed using easy to use command-line Python scripts or a simple web accessible form. The search task can be either different types of free-text search or so-called faceted search, concerning tags stored in the metadata accompanying the data. The users may refine their searches inside the received results.

4.5.4. Remarks about the implementation

Typically a data discovery service consumes a lot of hardware resources, especially when many users use it simultaneously with sophisticated search queries. Therefore, the above hardware configuration will be upgraded depending on the usage load.

4.6. VI-SEEM data analysis service

The VI-SEEM data analysis service will provide the capability to the VRE users to carefully and efficiently investigate and analyse even very large, unstructured datasets. The VI-SEEM data analysis service is based on Hadoop technology.

4.6.1. HW/SW information

The hardware dedicated to the service consists of a quad core Xeon E3-1220-v3 machine which acts as a HDFS name node, YARN server, and a login node, and 3 data/worker nodes each equipped with 24 (logical) core Xeon E5 2620 v3 cpus and 64 GB RAM. There is effective 7 TB of storage in the current HDFS setup. The system is configured to run up to 60 containers in parallel with 3 GB of RAM per container. If not otherwise specified in HDFS command line interface, the files are split into 128MB chunks.

Apache Hadoop was chosen as the main platform for the VI-SEEM data analysis service. The main components of the current setup are the distributed file system – HDFS, the resource manager YARN and MapReduce as the analysis framework.

4.6.2. Service endpoint

The data analysis service is hosted at the Institute of Physics Belgrade on machine hadoop.ipb.ac.rs.

4.6.3. Detailed description

The workflow that this system supports begins with a large semi structured dataset being uploaded to the distributed file system, which splits it into chunks and distributes the chunks across the data nodes in a redundant way to achieve fault

tolerance and faster access to data. The analysis procedure is defined in steps of applying a map and then a reduce function to the data. Map function is applied to every data point in the dataset and is expected to transform it into a key-value pair. Those key-value pairs are sorted and merged by key, so that all the values for the same key are collected. Such sorted list of key value pairs is given to the reduce function which processes the collected values for each key and returns the final result, also in the form of key-value pairs. The number of map functions running in parallel depends on the dataset size (i.e. number of chunks) and the user does not need to control that. A single reducer is launched by default, but the user has the control over that number when launching the MapReduce job.

To upload the data to the system, in the current setup, users will have to go through ssh on the Hadoop cluster login node on which they have command line utilities to interact with HDFS. The direct access to the HDFS is not provided at the moment of writing this document. The data is expected to be in semi structured or unstructured collection of records, such as lines of text or records with binary data.

To define the analysis in terms of map-reduce operations Java API can be used and a Maven archetype for creating such Java projects is provided. Other programming languages can use Hadoop Streaming API. The only requirement that Hadoop Streaming imposes is that map and reduce are defined as executables which will receive data on standard input and emit their results to the standard output. Streaming takes care of sorting and distributing the intermediate results from mappers to reducers.

5. Deployment of supplementary services

5.1. VI-SEEM persistent identifier service

GRNET’s Handle Service for VI-SEEM

GRNET’s Handle Service is provided to VI-SEEM for the purposes of persistently identifying digital objects across their lifecycle. Such digital objects are maintained in the VI-SEEM repository as well as other VI-SEEM generic or application specific data services. The handle service supports the management and resolution of:

- Resources: Digital objects and other internet resources.
- Part Identifiers: computes an unlimited number of handles on the fly.
- Multiple locations in a single handle: structured alternatives, e.g. multiple locations, in a single handle value.

A Persistent Identifier, also known as PID, is an identifier that is effectively permanently assigned to the object. It provides a means of connecting and distinguishing between an identifier for an object (a permanent identity) and the object's location (although it may change over time). PIDs introduce a level of indirection and complexity, since apart from managing PID a separate service needs to be used so as to resolve it.

The European Persistent Identifier Consortium (EPIC) provides persistent identifier (PID) services for European scientific and cultural heritage communities, using the Handle System (http://www.handle.net). The Handle System consists of a Global Handle Service (GHS) and Local Handle Services (LHS). It provides a resolution system consisting of a distributed infrastructure of global, local, and caching servers. GRNET provides a high availability of the PID service as a LHS. GRNET service supports the EPIC REST web service for issuing and managing PIDs.

VI-SEEM, has requested access accounts in order to be able to use the GRNET PID Service. VI-SEEM is responsible to create, maintain and update its PID collection by using the REST web service when it is necessary.

Handles are persistent identifiers for Internet resources. In the handle system the syntax of a PID handle consists of a Prefix and a Suffix.

Prefix: is used to access the service information that describes the "home" service (each organization may have one or more prefixes under its ownership).

Suffix: is a unique "local name" under the prefix. The uniqueness of a prefix and a local name under that prefix ensures that any identifier is globally unique within the context of the Handle System.

When issuing a PID, for example the VI-SEEM repository as a user, must provide some information about the resource such as the URL of the resource, and a unique prefix / suffix of its choice. As a response, the service registers the PID, which the VI-SEEM repository keeps in the metadata associated with the resource. The PID is globally unique, persistent, and resolvable by anyone. The PID can be resolved

through the Resolution Service, which will redirect the user to the registered location of the resource.

Handle service deployment

At GRNET the service provided to VI-SEEM among others, is hosted in virtual machines hosted in two different IaaS infrastructures offered by GRNET. The VMs are running version 8.1.0 of the handle.net software [ref: http://www.hdl.net/download_hnr.html] and they run several handle service instances at a primary / mirror handle service configuration. At the same VMs the relevant instances of the epic api are running i.e.

https://epic.grnet.gr/api/v2/handles/11239

https://epic.grnet.gr/api/v2/handles/11500

Replication between primary and mirror handle server is being implemented in an automatic way by the handle service. During the initial installation of a mirror server the administrator needs to manually bootstrap the mirror by using the hdl-dumpfromprimary tool. This tool downloads all the handles from the primary server. Then the mirror handle server automatically pulls the changes from the primary server at pre-defined, frequent intervals.

5.2. VI-SEEM AAI - Integration the data services with AAI

In order to provide easy to use and secure services to the VRE users, the VI-SEEM data services have to be integrated with the AAI infrastructure. The VI-SEEM AAI infrastructure and its services are described in detail in the deliverable D3.1. Some of the data services are already fully integrated with the AAI, while others (e.g. the archival service and the data staging service) will be integrated later (by M16). Further information is given in Chapter 7.

5.2.1. VI-SEEM simple storage service

In order to integrate VI-SEEM simple storage service with VI-SEEM AAI, it was extended with “user_shibboleth” ownCloud app that is available on EUDATA B2DROP Git page (https://github.com/EUDAT-B2DROP). This app allows authentication using Shibboleth Service Provider (SP). In order to enable this type of authentication, Shibboleth SP had to be deployed on the machine, apache Web server had to be appropriately configured and Shibboleth environment had to be established.

In the current configuration, VI-SEEM simple storage service supports two types of authentication when a user visits its login page in the browser: standard username/password authentication, and authentication using Shibboleth that is available by clicking on the button “Shibboleth – VI-SEEM” under the “Alternative logins” label. A click on that button will take the user to the IdP login page where he/she can be authenticated. After successful authentication the browser will open the user’s VI-SEEM simple storage service “Files” page. In the case of a new user, a new account will automatically be created.

Further integration is needed to enable desktop sync clients to work with users added through Shibboleth, as clients rely on username/password authentication. The original “user_shibboleth” app was developed for older versions of ownCloud so some compatibility issues need to be resolved as VI-SEEM simple storage service is based on a newer version. WebDAV usage still needs to be further tested and appropriately configured for this type of users.

The next step in the deployment of this service would be integration into the VI-SEEM environment by enabling authentication using the VI-SEEM Login IdP Proxy. This is described in more detail in Chapter 7 of this deliverable.

5.2.2. VI-SEEM repository service

The VI-SEEM repository is already integrated with the VI-SEEM Login service utilizing the VI-SEEM AAI infrastructure. Therefore, any VI-SEEM registered user can access the repository and based on the access rights that are provided by the Business Layer of the DSpace software. More details about the Authorization scheme and its usage are provided in the section of the deliverable that describes the usage scenario of the VI-SEEM repository service (Chapter 6).

5.2.3. VI-SEEM data discovery service

The data discovery service can use the Shibboleth service, which is the base of the VI-SEEM AAI. It is a SAML-based technology and protocol for authenticating to web services used by many “identity federations’’. The well developed software module mod_shib is provided and documented for Apache but can also be used with Tomcat.

5.2.4. VI-SEEM data analysis service

Current implementation of the VI-SEEM data Analysis Service is based on Apache Hadoop cluster installed at the Institute of Physics Belgrade. Access to the cluster is provided via SSH protocol. Also, we are investigating possibilities for deployment of web-based access through the Hue (http://gethue.com/) platform. Integration of the web-based services with VI-SEEM AAI is well-established and documented, so we expect it to be applicable on the Hue platform.

Within the D3.1 we have described options for integration of non-web-based resource with AAI. All available candidate solutions are under development, with no stable version for production deployment. One of these approaches is CILogon, and CILogon TTS - AARC Project (https://aarc-project.eu/) adopted version. It is a combination of different solutions that use SAML based credentials to provide access to non-web and x509 based resources. CILogon TTS should be able to provide end-user with the proxy certificate that can be used for interaction with various non-web services. For example, by enabling GSISSH on the login node of VI-SEEM Data Analysis Service, users would be able to login using their proxy certificates.

Another TTS service that is currently evaluated as a pilot project by AARC is LDAP Facade. It aims to provide access to non-web resources (e.g. available via SSH protocol) using existing AAIs, without need to obtain user certificates. It combines SAML logic and LDAP directory interface and appears as a local LDAP directory to the service and as a SP to the SAML federation. Another option would be GEANT Trusted Certificate Service (TCS) that releases X.509 certificates to users authenticated via AAI. Work on these solutions will be closely monitored by VI-SEEM, and depending on their outcomes the most appropriate solution will be integrated into VI-SEEM Data Analysis service.

6. Description of the potential usage scenarios

6.1. VI-SEEM simple storage service

VI-SEEM scientific communities can use simple storage service to keep and sync research data on various devices, as well as to share these data thus making it a useful tool in collaborative environment. Access is enabled via web browsers, desktop and mobile clients.

Typical usage scenario would be sharing data with other researchers working on the same VI-SEEM services or belonging to same VI-SEEM community through defined groups on the simple storage service server (one of the features of the ownCloud on which this service is based). VI-SEEM simple storage service supports versioning of the files and this feature is useful for shared data that is under development and it is often changed by collaborating users. In addition, ownCloud through its Documents applications supports collaborative editing of. odt or .doc files within the browser. Users can also post comments for each file.

In addition to sharing with other VI-SEEM Simple storage service users, researches are also able to create public shares of their files for general public and those shares can be protected by password if needed.

Supported VI-SEEM communities could store and share various kind of data and metadata: simulation and observational data for climate scientific community (e.g. raw models output and post-processed data, daily recorded meteorological parameters, rainfall records, etc.), images and text for digital cultural heritage scientific community (e.g. RTI datasets, geoelectrical tomographic data, digitized handwritten documents and books, aerial images, etc.), and images, simulation and experimental data in the case of life sciences scientific community (e.g. medical patients datasets, datasets with molecule synthesis results, MD trajectories, etc.).

6.2. VI-SEEM repository service

This section provides a high level description of the usage of the VI-SEEM repository service tailored to the needs of the VI-SEEM project and communities.

The VI-SEEM repository identifies three main user roles:

- The Submitter or Contributor: Submitters are members of the VI-SEEM community that are offering data for storage to the VI-SEEM repository. They are usually identified among the service provider teams that work in the context of VI-SEEM application’s enabling tasks of WP5. However, any interested party in the community can become contributor after getting access to the repository via the VI-SEEM open calls organized by WP6.
- The Collection Curator: This role is assigned to members of the VI-SEEM community that manage the data collections within the VI-SEEM repository. By default, the VI-SEEM repository has been configured to host 4 collections:

    - A generic VI-SEEM project collection that holds data that are of interest to the whole VI-SEEM community independent of the research field they are working on.
    - The Life Sciences community repository that stores and offers data related to the Life Sciences community of the region.
    - The Climate Sciences Community repository that stores and offers data related to the Climate Community of the region
    - and the Digital Cultural Heritage community repository that stores and offers data related to the Digital Cultural Heritage Community of the region.
- The end user: End users can be either registered to VI-SEEM or non-registered / anonymous users that have access to the open data provided in the VI-SEEM repository.

The following represents the usual workflow for the usage of the VI-SEEM repository.

The Submitter via the web interface logins to the VI-SEEM repository using the VI-SEEM Login service and can create an archival item by depositing files. The repository can handle any format from simple text files, documents, to data sets and video files. The files or organized together in related sets. Every archival item has an associated description (metadata). An item's exposed metadata is indexed for browsing and searching. Items are organized into collections of logically-related material. The end-user interface supports browsing and searching the archives. Once an item is located, Web-native formatted files can be displayed in a Web browser while other formats can be downloaded and opened with a suitable application program.

Figure 3 shows the Home page of the VI-SEEM repository and depicts the structure of the repository with the four different communities as described above.

Initially the VI-SEEM repository will host a large number of the data sets described in D5.1 of the project, while in the future it will attract more data sets from the applications that will be deployed in the context of the open calls.

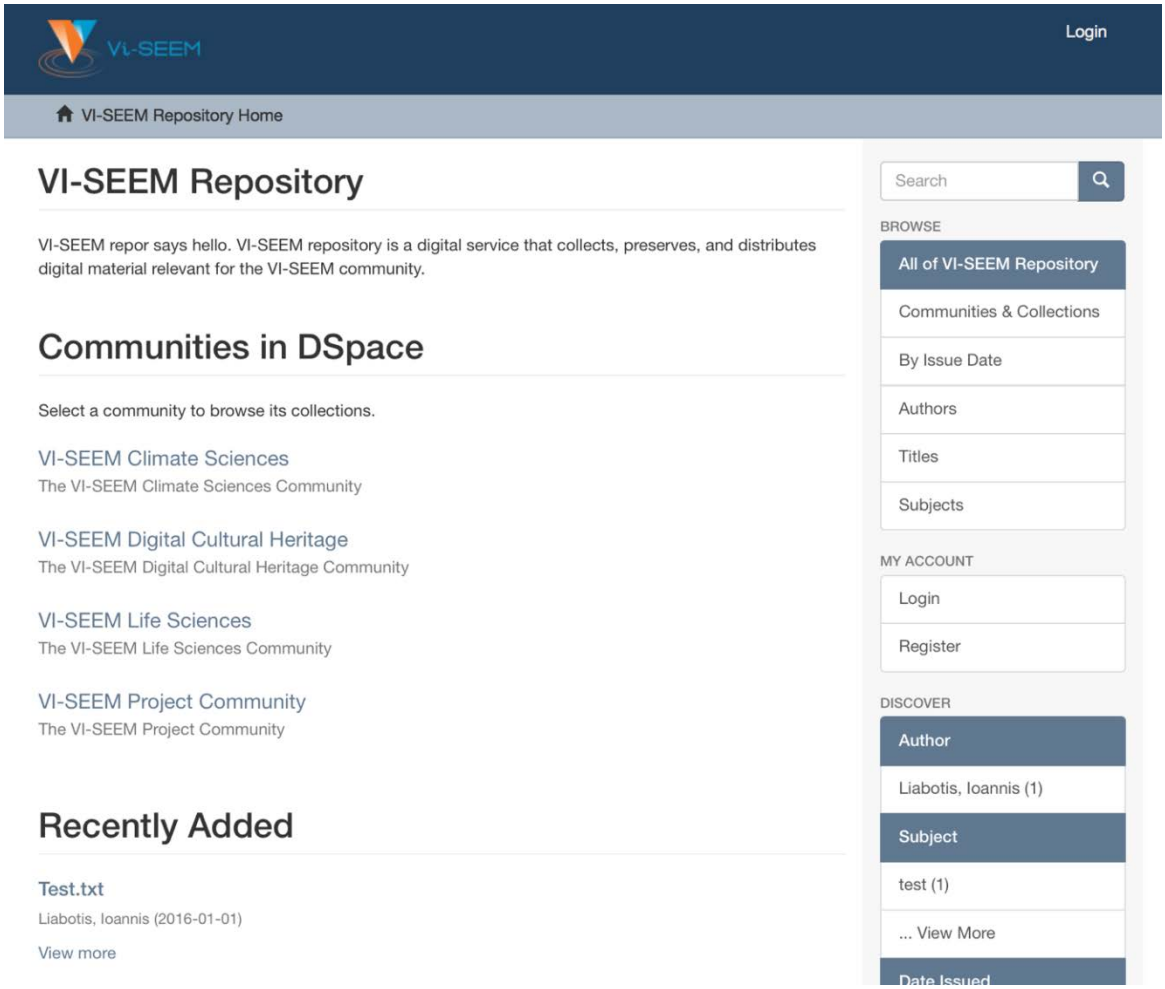

Figure 4. The VI-SEEM repository home page

The submission of a data item requires that a set of metadata is provided. The VI-SEEM repository uses the Dublin Core metadata specification [http://dublincore.org]. A detailed description of this schema is provided at: https://wiki.duraspace.org/display/DSDOC4x/Metadata+and+Bitstream+Format+Registries

6.3. VI-SEEM archival service

The archival service provides longer-term data storage for users. It also provides metadata capabilities, so users could attach metadata to their ingested data objects, search for objects by metadata or even have the metadata automatically harvested for their data objects, although the latter may require further research and development.

Coupled with the VI-SEEM work storage space/local storage and data staging service users would be able to take advantage of rather complex policies e.g.:

1) Ingest data
2) Have metadata automatically attached to ingested data (policy)
3) Have data automatically replicated to other zones in VI-SEEM (policy)
4) Transfer data to computation site (data staging, e.g. globus-url-copy)
5) Transfer computation results back to iRODS (data staging)
6) Further processing/replication could be done on results (policy)

6.4. VI-SEEM work storage space / local storage and data staging

The local storage and data staging service provides short-term data storage as well as data movement capabilities for users.

A user would typically transfer data to local storage before processing it (computation) and fetch the results or move data back home or to VI-SEEM archival service after the jobs are done.

6.5. VI-SEEM data discovery service

The Data Discovery Service is based on CKAN, providing the functionality of quick keyword search combined with data tags like the EUDAT B2FIND tags.

Following the organization in CKAN, the authorization is the primary way to control who can see, create and update datasets. Data sets that are marked as “public” are visible to everyone. Private datasets could be seen for example within the frame of the 3 different application communities, within the separate application team members, or others. The main methods of finding and accessing the data are through an easy to use web-based interface, using the command-line, but an API (CKAN’s Action API) is also available.

The core usage of the Data Discovery Service is expected to be in that applications will publish metadata to it. The metadata may be searched by users from the project or by other users, depending on the permissions.

The Internet protocol which allows data records provisioning is OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) which is a low-barrier mechanism for repository interoperability. Repositories that expose structured metadata via OAI-PMH can be harvested by Service Providers through OAI-PMH service requests.

The VI-SEEM communities will select which datasets and with what permissions will be exposed through the Data Discovery Service.

The communities themselves decide which metadata are made available and how their metadata elements are mapped.

More specifically, the Data Discovery Service will be integrated with the VI-SEEM Repository so that basic metadata about the stored datasets is synchronized

automatically, based on the trusted relationship between the two services. Rich metadata information about the datasets will be provided by the applications, depending on their usage scenarios and the needs of the respective communities. In general only keywords, tags and small documents (e.g., input files) will be stored in the Data Discovery Service, while the whole datasets or individual large files will be presented only as links (URIs).

For example, the DCH community operates with virtually any type of data: from simple arrays containing numbers that represent chemical or physical characteristics of a given artifact, plain texts, to high-resolution 2D and 3D images, video clips, audio records etc. In addition, many artifacts have unique characteristics, therefore they cannot be grouped together effectively and summarizing queries in PivotTable style cannot be performed on them. The other research communities have less types of data to process and the data formats are well documented and standardized. Hence, various DCH metadata are needed to make the datasets usable both inside and outside the project. The format of the metadata for each application should be thoroughly and carefully defined first, taking into account the needs of all potential end users and VI-SEEM applications developers. Then the uploading of datasets to the VI-SEEM repository or to any application-specific database will be coupled with the uploading of the corresponding metadata to the Data Discovery Service. The end-users will be able to search in the Data Discovery Service and obtain either the needed metadata immediately or a link to the datasets where they can obtain the parts of the data they are interested in. Thus there are two flows in this usage scenario:

- dataset metadata flow to the Data Discovery Service (DDS)
- end-user utilization of Data Discovery Service to obtain the metadata directly or the URI for the datasets.

The end-user utilization in this case is less complicated, because a user can log in to the Data Discovery service, be authenticated and authorized within the VI-SEEM community and use the detailed form to perform searching of datasets. Another option would be to use parts of the application for that, through the API. The further processing, based on the received URI, can be done by the VI-SEEM application or using other tools.

For the Climate Modelling and Weather Forecasting communities the data formats are well established and in many cases similar software tools are used, although in different ways and in combination with some tools developed by the researchers themselves. Because of this level of standardization some of the input files may be added to the Data Discovery Service together with keywords and descriptions of the types of computations being performed – e.g., validation, simulation, forecasting. The links to the software codes from the VI-SEEM code repository service shall also be provided. In this way an interested researcher should be able to either repeat the computations or use the same combination of codes on data from their own region, validating the computational approaches and evaluating their applicability for other scenarios. For each computation there should be clear description in the metadata about what kind of assumptions it is based on, in order to facilitate comparisons and increase transparency.

The Life Science community has diverse applications, which sometimes involve complicated workflows. Some computations produce large output datasets from relatively small input files. In that case these input files shall be stored together with URI for the result dataset, as well as metadata about the computation itself, e.g., how it was performed, on which machine, etc.

One popular format for life-science data is the DICOM. Since the DICOM Network will provide a web portal with online access to such datasets, we foresee that most of the searching functionality will be available in the Portal itself. However, some of the metadata shall be published in the Data Discovery Service so as to enable more generic searching through all the VI-SEEM data to be successful and to direct the end-users to the specific DICOM Portal where they can narrow the search and find the specific resources they are looking for.

6.6. VI-SEEM data analysis service

MapReduce enforces the programming pattern, but its applicability to various data analysis problems is surprisingly high. A lot of problems can be framed in that way. In the climate community, large quantities of various sensor and historical data can be processed quickly if they are stored in the form arrays such as in netCDF formats. The data from the climate community appears to be particularly good fit for analysis on Hadoop. Also in life sciences, gene sequences can be processed very efficiently. For the projects dealing with image data, the conversion of images to sequence files, where value of records is the binary image data (optionally encoded in base64), can be done to take advantage of the system.

In all cases, Hadoop will have most effectiveness on datasets with large number of data points. Based on the VI-SEEM WP5 Application Survey, we have generated list of potential users of VI-SEEM Data Analysis Service.

From the Life Sciences Community the following applications expressed interest for usage of Hadoop-based data analysis service:

- BioMoFS from Yerevan State University, Armenia
- NGS from Cyprus Institute of Neurology and Genetics, Cyprus
- THERMOGENOME from Institute of Molecular Biology, Bulgaria
- SQP-IRS from SESAME (Synchrotron-light for Experimental Science and Application in the Middle East), Jordan
- NS from Physics Department, National Research Center, Egypt
- NGS from Translational Genetics Team - Cyprus Institute of Neurology and Genetics, Cyprus
- PSOMI from Faculty of Natural Sciences, University of Montenegro, Montenegro

From the Climate Community:

- ClimStudyArmenia from Armenian State Hydrometeorological and Monitoring Service, Armenia
- EMAC from Cyprus Institute, Cyprus

- ACIQLife from National Institute of Geophysics, Geodesy and Geography, Bulgaria

From the Cultural Heritage Community:

- CH-CBIR from University of Banja Luka, Faculty of Electrical Engineering, Bosnia and Herzegovina
- IMC4CH from IICT-BAS, Department of Linguistic Modelling and Knowledge Processing; National History Museum (NHM), Bulgaria
- Dioptra from The Cyprus Institute, STARC, Cyprus

7. Further phases of deployment / integration

With the initial deployment the VI-SEEM data services can already be used for testing and supporting pilot applications. However to reach their final, production stage, further developments are needed, as well as the data services have to be brought together with the other services of the whole VI-SEEM ecosystem. This chapter describes the main related developments planned for the next phase of the project.

7.1. VI-SEEM simple storage service

Further integration is needed to enable desktop sync clients to work with users added through Shibboleth, as clients rely on username/password authentication. The original "user_shibboleth" app was developed for older versions of ownCloud so some compatibility issues need to be resolved as VI-SEEM simple storage service is based on a newer version. WebDAV usage still needs to be further tested and appropriately configured for this type of users.

In the current implementation stage of VI-SEEM simple storage service, Shibboleth authentication was successfully tested against test IdP available at http://testshib.org, which is a testing service for Shibboleth and SAML2 deployments.

Next step would be to implement authentication against VI-SEEM Login IdP Proxy according to the guide given at VI-SEEM Wiki Login integration guide for Service Providers page (http://wiki.VI-SEEM.eu/index.php/VI-SEEM_Login_integration_guide_for_Service_Providers). That will require VI-SEEM simple storage service SP metadata to be provided to VI-SEEM AAI team for inclusion in VI-SEEM Login IdP Proxy. Also, modification of "user_shibboleth" ownCloud app will be needed in order to handle attributes that are released by VI-SEEM Login IdP Proxy for authenticated VI-SEEM users (eduPersonUniqueId, mail, eduPersonEntitlement, etc.) that will be used for account creation in VI-SEEM simple storage service.

Customization and installation of additional apps that will provide added functionalities can also be done based on VI-SEEM users’ requirements.

7.2. VI-SEEM repository service

The VI-SEEM repository will be soon integrated with the GRNET Handle service [13]. GRNET Handle service is a service dedicated to provide, resolve and mint persistent identifiers (PID). DSpace requires that a persistent identifier is assigned to each digital object (Item, Collection, Community). Because the developers wanted a solution which will work for a very long time, the identifier system had to be independent of any underlying network protocols, such as HTTP.

DSpace uses the Handle System from CNRI (Corporation for National Research Initiatives) as the persistent identifier for each digital object. Handles are resolved to actual URLs via a resolution service. The Handle resolver is an open-source system. Handles in DSpace (and elsewhere) are currently implemented as HTTP URIs, but can also be modified to work with future protocols. The Handle system is also able to support existing bibliographic identifiers such as ISBN or ISSN.

To implement support for other registration agencies, we have to develop a Java class that implements the interface DOIConnector using the DataCiteConnector as an example. The we configure the system to use our own DOIConnector when configuring the IdentifierService instead of the DataCiteConnector. For more information on the integration implementation methodology one can visit http://dspace.org/sites/dspace.org/files/archive/1_5_2Documentation/ch02.html - N1041B

7.3. VI-SEEM archival service

Possible further phases could include:

- deployment of additional zones
- development of VI-SEEM wide and per-zone policies
- deployment of alternative access methods (by default access is provided through iCommands)
- integration with VI-SEEM AAI (most likely for a Web-based alternative access mode)
- integration with VI-SEEM data discovery service
- development of automated tasks such as metadata extraction for ingested data objects

7.4. VI-SEEM work storage space / local storage and data staging

A possible integration method would be to have a web UI (accessible through VI-SEEM AAI) where users may generate a token for use with VAS and/or VLS.

The token itself should be stored in LDAP and then PAM LDAP could be used for authentication. Automatic account creation would also be a useful addition to this.

7.5. VI-SEEM data discovery service

It is important to fully integrate the VI-SEEM Data Discovery Service with the other VI-SEEM services and especially with the VI-SEEM AAI and Accounting Services.

The data discovery service can use the Shibboleth service, which can be integrated with VI-SEEM AAI. It is a SAML-based technology and protocol for authenticating to

web services used by many "identity federations". The well developed software module mod_shib is provided and documented for Apache but can also be used with Tomcat.

The accounting data will be harvested with scripts and published to the Accounting Service that is also deployed at IICT-BAS, so this step is simplified. The selection of proper metadata schemes for the different communities will be worked out with the community leaders based on the feedback from application developers. Based on the levels of usage that we observe we may need to upgrade the hardware configuration of the server or to make use of VI-SEEM cloud resources for backup. The most important interactions will be with the VI-SEEM archival services (based on IRODS) where a trusted relationships should be established.

7.6. VI-SEEM data analysis service

One of the advantages of YARN is allowing for different kinds of jobs to be run, beside MapReduce. This leaves the possibility to add another framework from the Hadoop ecosystem if required by the VI-SEEM community. One of those parallel execution frameworks that can be evaluated for possible future integration is Apache Spark. It is one of the most popular alternatives to MapReduce which can be significantly (10 to 100 times) faster for some analysis workflows since it does all its processing steps in the system memory, while MapReduce writes temporary results to local disks.

8. Conclusions

In Deliverable D4.1 [2] the VI-SEEM consortium defined the generic data services of the VI-SEEM ecosystem and their deployment plans. According to these definitions and plans the consortium partners responsible for performing the tasks of WP4 deployed the initial versions of the different data services. All of the data services have been set up at one or more consortium partners, and they all are ready for initial usage and further testing, experiments and integration with other services of the VI-SEEM. The potential usage scenarios of the data services have also been elaborated, with close cooperation with the other WPs of the project. The results and experiences are summarized in this deliverable (D4.2), and the detailed technical descriptions and configurations are also uploaded to the VI-SEEM wiki knowledge base. The further development steps that are necessary to perform in the next phase of the project to bring the data services to their full potential are set in this document as well.

9. Annexes

Technical details and configuration guidelines of the different VI-SEEM data services are provided on the VI-SEEM Wiki page: http://wiki.vi-seem.eu/

You can find below a summary of the installation steps.

9.1. VI-SEEM simple storage service

VI-SEEM Simple Storage Service (https://ipbbox.ipb.ac.rs/) is based on ownCloud platform (https://owncloud.org/) that can be installed on different version of Linux operating system (Red Hat/Centos, Debian, SUSE, Ubuntu). Recommended version of installed Web server is Apache 2.4 with mod_php, PHP version should be 5.5+, with MySQL/MariaDB database (although SQLite and PostgreSQL are also supported). Recommended configuration for instance that will support up to 150 users is server with at least 2 CPU cores, 16GB RAM, and with enough amount of local storage space.

There are two installation methods: manual installation from the tar archive, or from distribution packages which are, from version 9, divided in multiple packages: `owncloud-deps` and `owncloud-files`. Package `owncloud-files` installs only ownCloud, without Apache, database, or PHP dependencies while `owncloud-deps` package install all dependencies (Apache, PHP, and MySQL) and this package is not meant to be installed by itself but pulled in by the metapackage `owncloud`. By installing metapackage `owncloud`, user will get a complete installation with all dependencies. In case of installing just `owncloud-files` package administrator will have to install LAMP stack first which allows him to create its own custom LAMP stack without dependency conflicts with the ownCloud package. Packages are available in ownCloud repositories (https://download.owncloud.org/download/repositories/): in stable repository that always tracks the current stable ownCloud version, and in specific major release repositories which usage prevents accidental upgrade. VI-SEEM Simple Storage Service at IPB is based on version 9.0 of ownCloud installed on Debian 8 OS.

After the installation of necessary packages, it is recommended that administrator improve the security of the ownCloud directories by setting the proper permissions, as strict as possible. There is a useful script for that which can be found on ownCloud documentation pages (https://doc.owncloud.org/server/9.0/admin_manual/installation/installation_wizard.html).

Next step would be to complete the installation by running the Installation Wizard by pointing the Web Browser to http://localhost/owncloud (or http://your.server/owncloud). On that page administrator needs to set username and password of ownCloud administrator, to define location of ownCloud data directory (where users’ files will actually be stored), and to choose database that will be used (MySQL/MariaDB is recommended). It is advisable to configure data directory

location during the installation as it is not simple to relocate if after. That directory must exist and it must be owned by OS HTTP user. Configuration of ownCloud can also be performed by using the command line and provided occ tool that needs to be run as HTTP user.

All URLs that are used to access ownCloud server need to be listed in ownCloud `config.php` file, under `trusted_domains` setting and both IP addresses and domain names can be used. Users will be allowed to log in to ownCloud only if they, in their browsers, use URL that is listed in the `trusted_domains` setting.

As a best practice, it is important that administrator configure its production server to use HTTPS instead of HTTP and unencrypted HTTP should never be allowed. For production instance, a browser-friendly SSL certificate that is issued by a trusted certificate authority should be used.

By default, ownCloud server will be accessible under the `/owncloud` web route (e.g. https://your.server/owncloud/) but it can be changed to https://your.server/ by changing the Apache virtual host or conf settings, as well as modifying `/var/www/owncloud/config/config.php` and `/var/www/owncloud/.htaccess` files. Example for that can be found on ownCloud documentation page https://doc.owncloud.org/server/9.0/admin_manual/installation/changing_the_web_route.html.

In order to improve performance of the ownCloud instance, it also advisable that administrator configures memcache, and ownCloud supports multiple PHP caching extensions such as APCu or Memcached.

With created ownCloud administrator user, service administrator can now login to installed ownCloud instance where it can perform additional configuration of ownCloud service. On the **Admin** page various options can be configured such as server side encryption, email notifications, sharing options and other. On the **Users** page, users and groups can be created and edited. Additional functionalities to ownCloud instance could be added by installing ownCloud applications, so called apps, on the **Apps** page. Application can be enabled or disabled by clicking on **Enable** or **Disable** button. In case where app is not part of the ownCloud installation, it will be downloaded from the ownCloud app store, installed and enabled.

In order to integrate VI-SEEM Simple Storage Service into VI-SEEM AAI, it should be configured to act as a SAML Service Provider (SP). At the IPB instance it was done by using Shibboleth software solution so that `user_shibboleth` ownCloud app which enables Shibboleth authentication for ownCloud users could be used. Necessary Shibboleth packages need to be installed (in case of Debian, as on IPB instance, it is `libapache2-mod-shib2`) and afterwards service needs to be configured by editing files in `/etc/shibboleth` folder, namely `/etc/shibboleth/shibboleth2.xml` and `/etc/shibboleth/attribute-map.xml`. First file is the main configuration file which contains data about the SP and also about Shibboleth Identity Provider (IdP, VI-SEEM Login IdP Proxy in VI-SEEM environment) while second file tells the SP how to map SAML attributes received from IdP to environment variables that can be then used in web applications.

Shibboleth will also need SSL key/certificate pair and it can be generated manually (e.g. by using command `shib-keygen` on Debian) or administrator can use already obtained browser friendly certificate and key.

In any moment Shibboleth SP setup can be tested against the TestShib web site (https://www.testshib.org/) which provides means for testing both for Shibboleth SP and IdP services.

When Shibboleth setup is completed, administrator needs to provide SP metadata to AAI team so that their SP can connect to VI-SEEM Login IdP Proxy as it is stated on VI-SEEM Wiki page dedicated to integration of Service Providers into VI-SEEM AAI infrastructure http://wiki.vi-seem.eu/index.php/VI-SEEM_Login_integration_guide_for_Service_Providers. This metadata includes `entityID` and `Metadata URL` and in the case of VI-SEEM Simple Storage Service instance at IPB these values are https://ipbbox.ipb.ac.rs/shibboleth (for `entityID`) and https://ipbbox.ipb.ac.rs/Shibboleth.sso/Metadata (for `Metadata URL`). VI-SEEM Login IdP Proxy metadata should already be present in `/etc/shibboleth/shibboleth2.xml` file.

OwnCloud app `user_shibboleth` (https://github.com/EUDAT-B2DROP/user_shibboleth) was added to ownCloud IPB instance so that it can use Shibboleth service for user authentication. App’s folder was added to ownCloud applications folder (`/var/www/owncloud/apps/`) and after that it was enabled through ownCloud Web Interface on **Apps** page and needed parameters were added to its configuration on **Admin** page as it was described in its readme file. Also, lines necessary for Shibboleth authentication were added to Apache ownCloud virtual host configuration file, together with lines needed for Shibboleth services that were added to `/etc/shibboleth/shibboleth2.xml` file. All these configuration changes can be found in app’s readme file.

Proper attributes released from VI-SEEM Login IdP Proxy upon successful authentication (e.g. `eduPersonUniqueId`, `mail`, `display name`) need to be manually enabled in app’s source files (PHP code) since this ownCloud app was originally developed for older version of ownCloud. In the current setup, after initial successful authentication to VI-SEEM Login IdP Proxy user is automatically created in VI-SEEM Simple Storage Service and logged in.

Technical details and configuration are provided on the VI-SEEM Wiki page: http://wiki.vi-seem.eu/index.php/VI-SEEM_Simple_Storage_Service_configuration_guidelines

9.2. VI-SEEM archival service

`###` iRODS deployment `###`

for Ubuntu 14.04 LTS

9.2.1. Part 1: Preparation and iCAT server installation

```
### Machine hosting PostgreSQL ###

# PostgreSQL repo

#########################################################
sudo su -

cd /etc/apt/sources.list.d/

vim ./pgdg.list

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | apt-key
add -

apt-get install postgresql-9.5 postgresql-contrib-9.5
#########################################################

# Content of pgdg.list
#########################################################
deb http://apt.postgresql.org/pub/repos/apt/ trusty-pgdg main
#########################################################

---

# PostgreSQL config

Default data directory is:
data_directory = '/var/lib/postgresql/9.5/main'

It is advisable to change this to a dir where you have plenty of space.

Also, listen_addresses = 'localhost' by default.
Change this as appropriate (comma-separated list of IP address(es) to listen on;
use '*' for all).

---

Example: dedicated storage space for PostgreSQL on XFS on LVM on /dev/vdb
- packages needed: lvm2, xfsprogs

#########################################################
sudo pvcreate /dev/vdb
sudo vgcreate pgdb_data_vg /dev/vdb
sudo lvcreate -n pgdb_data_lv_001 -l 100%VG pgdb_data_vg
sudo mkfs.xfs -L pgdb-data-fs /dev/pgdb_data_vg/pgdb_data_lv_001
#########################################################

Access/Mount/fstab:
#########################################################
sudo mkdir /path/to/mount/point
sudo chmod 700 /path/to/mount/point
sudo mount /dev/pgdb_data_vg/pgdb_data_lv_001 /path/to/mount/point
sudo chown postgres:postgres /path/to/mount/point
sudo su - postgres
```

```
cd /path/to/mount/point
umask 0022
mkdir 9.5
umask 0077
mkdir main
sudo vim /etc/fstab
########################################################

# Content of /etc/fstab:
########################################################
...
LABEL=pgdb-data-fs      /path/to/mount/point xfs defaults       0 0
########################################################

---

# Stopping PostgreSQL; modifying data_directory

########################################################
sudo -u postgres /usr/lib/postgresql/9.5/bin/pg_ctl stop \
-D /var/lib/postgresql/9.5/main -m fast
sudo vim /etc/postgresql/9.5/main/postgresql.conf
cd /usr/lib/postgresql/9.5/bin/
sudo su postgres
./pg_ctl -D /path/to/mount/point/9.5/main initdb
exit
sudo service postgresql start
########################################################

# Changed data directory in /etc/postgresql/9.5/main/postgresql.conf
########################################################
data_directory = '/path/to/mount/point/9.5/main'
########################################################

Note: It is considered best-practice to use a directory under the mount point
rather than the mount point itself.

---

# Preparation for iRODS on PostgreSQL side

########################################################
sudo su postgres

psql

CREATE USER irods WITH PASSWORD '<pwd was here>';
CREATE DATABASE "ICAT";
GRANT ALL PRIVILEGES ON DATABASE "ICAT" TO irods;
########################################################
---

# PostgreSQL auth

sudo -u postgres vim /etc/postgresql/9.5/main/pg_hba.conf

Example (if your iCAT server is in 192.168.1.0/27):
```

```
############################################################
# Allow connections from private subnet to ICAT
host    ICAT        irods       192.168.1.0/27      md5
############################################################
---

### Machine hosting iCAT ###

# iCAT installation

Required packages (you could download them at http://irods.org/download/):
 - irods-icat-4.1.8-ubuntu14-x86_64.deb
 - irods-database-plugin-postgres-1.8-ubuntu14-x86_64.deb

There may be some missing dependencies. To resolve this, do:

sudo apt-get -f install

This will get you everything you need.


Then you have to run /var/lib/irods/packaging/setup_irods.sh

For documentation purposes you may do this like:


sudo /var/lib/irods/packaging/setup_irods.sh | tee ./iRODS-setup-$(date
+%Y%m%d-%Hh%M)

...and answer all questions:
- iRODS service account name (default: irods)
- iRODS service group name (default: irods)
  (these for running iRODS)

- iRODS server's zone name (default: tempZone)
- iRODS server's port (default: 1247)
- iRODS port range (begin) (default: 20000)
- iRODS port range (end) (default: 20199)

- iRODS Vault directory (default: /var/lib/irods/iRODS/Vault)

- iRODS server's zone_key (default: TEMPORARY_zone_key)
  (This will be shared amongst federation members.)
- iRODS server's negotiation_key (default: TEMPORARY_32byte_negotiation_key)
  (This will also be shared amongst federation members.)

- Control Plane port (default: 1248)
  (This for zone-wide administrative tasks.)
- Control Plane key (default: TEMPORARY__32byte_ctrl_plane_key)

- Schema Validation Base URI (or 'off') (default:
https://schemas.irods.org/configuration)

- iRODS server's administrator username (default: rods)
- iRODS server's administrator password

- Database server's hostname or IP address
- Database server's port (default: 5432)
```

- Database name (default: ICAT; see "Preparation for iRODS on PostgreSQL side")
- Database username (default: irods; see "Preparation for iRODS on PostgreSQL side")
- Database password (as configured in step "Preparation for iRODS on PostgreSQL side")

Notes:
- zone_name --> max. 63 characters; validating regexp: "^[A-Za-z0-9_\\.]+$"
- zone_key --> max. 49 characters; validating regexp: "^[A-Za-z0-9_]+$"
- negotiation_key --> 32 characters; string (no restrictions)
- server_control_plane_port --> integer
- server_control_plane_key --> 32 characters; string (no restrictions)

From iRODS documentation:
- "zone_key should be a unique and arbitrary string ... one for your whole zone"
- "This allows the resource servers to verify the identity of the iCAT server beyond just relying on DNS."
- "Between Two Zones
  ...
  The zone_key should be a unique and arbitrary string, one for each zone.
  The negotiation_key should be a shared key only for this pairing of two zones."
- Control Plane --> "The irods-grid command is purely within the control of a data-grid administrator. For this reason we decided to secure this side-channel communication with symmetric grid-wide keys. This way the only way a grid may be paused, shutdown or queried is by an administrator with the proper credentials."

The installation will create a default resource (demoResc) for you at the directory of your choice ("iRODS Vault directory"; see above). This is not intended for production so another resource should be created (either on the iCAT server or on a separate resource server).

Hosts

Be sure that your hostnames could be resolved!
Edit your /etc/hosts (IPs you will use and localhosts as well) as appropriate.

E.g.:
127.0.0.1 localhost my-vi-seem-db

192.168.1.2 my-vi-seem-db
192.168.1.3 myresc01 vi-seem-irods-resource-01
192.168.1.4 myicat vi-seem-icat

SSL

For SSL you should get a proper cert.
You will need:
- cert chain including the root CA itself
  If you have separate cert files this could be as simple as


```
##########################################################
   cat myicat.crt myCA.crt > mycertchain.pem

##########################################################

- a Diffie-Hellman parameters file

##########################################################
   openssl dhparam -2 -out dhparams.pem 2048

##########################################################

- all (cert key, cert chain, DH params file) accessible by the iRODS service account
  (Of course the private key should NOT be readable by others!)

- to modify the service account config (~/.irods/irods_environment.json);
  params needed:
    * irods_ssl_certificate_chain_file
    * irods_ssl_certificate_key_file
    * irods_ssl_dh_params_file
  e.g.:

##########################################################
    ...
    "irods_ssl_certificate_chain_file": "/etc/irods/ssl/mycertchain.pem",
    "irods_ssl_certificate_key_file": "/etc/irods/ssl/myicat.key",
    "irods_ssl_dh_params_file": "/etc/irods/ssl/dhparams.pem"

##########################################################

If you want to force SSL, you should modify /etc/irods/core.re as well:
##########################################################
acPreConnect(*OUT) { *OUT="CS_NEG_REQUIRE"; }
##########################################################
```

9.2.2. Part 2: Resource server installation

```
Required packages (you could download them at http://irods.org/download/):
 - irods-resource-4.1.8-ubuntu14-x86_64.deb

---

Assumptions:
 - iCAT server already deployed
 - iCAT config is tweaked to have SSL in place; this includes:
   * SSL related parameters in service account config (irods_environment.json)
   * acPreConnect rule in the default Rule Base (core.re)

---

# Resource server installation

This is basically the same as in case of the iCAT server; you have to intall
the downloaded package (irods-resource-4.1.8-ubuntu14-x86_64.deb).
```

If you had unmet dependencies, do as in case of iCAT server:
sudo apt-get -f install
This will get you everything you need.

Resource server configuration

Before running /var/lib/irods/packaging/setup_irods.sh you may consider tweaking some parts of the config. That way you may not get error messages.

Anyway you have to go through the setup at some time.
Again, for documentation purposes you may do this like:
sudo /var/lib/irods/packaging/setup_irods.sh | tee ./iRODS-setup-$(date +%Y%m%d-%Hh%M)

You need to answer all questions you are already familiar with:
- iRODS service account name (default: irods)
- iRODS service group name (default: irods)
  (these for running iRODS)

- iRODS server's port (default: 1247)
- iRODS port range (begin) (default: 20000)
- iRODS port range (end) (default: 20199)

- iRODS Vault directory (default: /var/lib/irods/iRODS/Vault)

- iRODS server's zone_key (default: TEMPORARY_zone_key)
  (This will be shared amongst federation members.)
- iRODS server's negotiation_key (default: TEMPORARY_32byte_negotiation_key)
  (This will also be shared amongst federation members.)

- Control Plane port (default: 1248)
  (This for zone-wide administrative tasks.)
- Control Plane key (default: TEMPORARY__32byte_ctrl_plane_key)

- Schema Validation Base URI (or 'off') (default:
https://schemas.irods.org/configuration)

- iRODS server's administrator username (default: rods)

- iCAT server's hostname
- iCAT server's ZoneName

- iCAT server's admin username
- iCAT server's admin password

The installation will try to create a default resource (<resource server name>Resource) for you at the directory of your choice ("iRODS Vault directory"; see above).

If the asssumptions (see above) are true and you do not tweak config on the resource server before running setup_irods.sh you should see an error message in the end:

```
################################################################################
...
Step 3 of 3:  Configuring iRODS user and starting server...
   Updating iRODS user's ~/.irods/irods_environment.json...
   Starting iRODS server...
Could not start iRODS server.
   Starting iRODS server...
Validating [/var/lib/irods/.irods/irods_environment.json]... Success
Validating [/etc/irods/server_config.json]... Success
Validating [/etc/irods/hosts_config.json]... Success
Validating [/etc/irods/host_access_control_config.json]... Success
iRODS server failed to start.


Install problem:
   Cannot start iRODS server.
Found 0 processes:
      There are no iRODS servers running.

Abort.
################################################################################
```

That is because you have SSL already in place on iCAT and require its use in the default Rule Base but things are not done yet on the resource server.

This also means (as the resource server could not connect to the iCAT) that the default resource is not created for your resource server in iCAT.

What you need to do now is:

- set up SSL just like the way you did it on the iCAT server
- modify corresponding config files
- remove "default_resource_directory" and "default_resource_name" from the service account config (irods_environment.json) as they are optional and only meant for the setup phase
- prepare your actual storage and the corresponding Vault directory
- start your resource server
- create your new resource

Creating your new resource

According to iRODS documentation it is considered best practice "to use a passthru resource as the root node of the Zone's default resource. By doing this, administrative changes to disks, server names, and resources can be handled out of view of the users and without the users needing to change any configuration in their client(s)."

Example of creating your resource hierarchy would look like:

```
##########################################################
irods@myicat:~$ iadmin mkresc myResc passthru
Creating resource:
Name:              "myResc"
Type:              "passthru"
Host:              ""
Path:              ""
Context: ""
irods@myicat:~$ iadmin mkresc myxfs01 unixfilesystem
myiresc01.example.org:/path/to/my/Vault
Creating resource:
Name:              "myxfs01"
Type:              "unixfilesystem"
Host:              "myiresc01.example.org"
Path:              "/path/to/my/Vault"
Context: ""
irods@myicat:~$
irods@myicat:~$ iadmin addchildtoresc myResc myxfs01
niifirods@niificat:~$ ilsresc --tree
demoResc
myResc:passthru
â””â”€â”€ myxfs01
##########################################################
```

Checking the status of your Zone

Now that you have your new resource created on iCAT, you may check the status of your Zone (either on your iCAT or on your resource server) by:

irods-grid status --all

(This has to be done by your iRODS service account.)

Example (how it would look like):

```
##########################################################
irods@myicat:~$ irods-grid status --all
{
   "hosts": [
      {
         "agents": [
            {
               "agent_pid": 12345,
               "age": 0
            }
         ],
         "status": "server_state_running",
         "xmsg_server_pid": 0,
         "hostname": "myiresc01.example.org",
         "irods_server_pid": 13579,
         "re_server_pid": 0
      },
      {
         "hostname": "myicat.example.org",
         "re_server_pid": 9876,
         "irods_server_pid": 9875,
```



```
        "status": "server_state_running",
        "xmsg_server_pid": 0,
        "agents": [
           {
              "agent_pid": 9321,
              "age": 1
           }
        ]
     }
  ]
}
irods@myicat:~$
```


9.3. VI-SEEM data discovery service

Installing the customized CKAN server:

The easiest way to install CKAN is from packages already provided by OS. The recommended operating system is a clean minimal **Ubuntu v14.04, 64bit** as it is also recommended from the CKAN developers. The suggestion is to configure one CKAN website per physical server.

Here are the necessary steps. They must be executed with administrative permissions (as user **root)**.

1) Update of the package index:
   apt-get update

2) Installing the necessary packages for CKAN including **git** and **wget** if they are not present already:
   **apt-get install nginx **
   **apache2 libapache2-mod-wsgi **
   **libpq5 **
   **git-core **
   wget
3) Enable the Apache **wsgi** module and restart the web server:
   a2enmod wsgi
   service apache2 restart

4) Download the *Python CKAN 2.5* package:
   wget http://packaging.ckan.org/python-ckan_2.5-trusty_amd64.deb

5) Install the *CKAN* package:
   dpkg -i python-ckan_2.5-trusty_amd64.deb

6) Install **PostgreSQL** database server and Solr search engine which uses **Jetty** Java web server and Java Servlet container:
   apt-get install postgresql solr-jetty

7) Edit the config file **/etc/default/jetty** and set **NO_START** to be 0 (zero).

8) Edit the config file **/etc/default/jetty** and set **JAVA_HOME** to point to the actual JDK installation directory. In our case:
JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64

9) Start the Jetty server:
service jetty start

10) Replace the default schema file **schema.xml** with a symbolic link to the CKAN schema file:
mv /etc/solr/conf/schema.xml /etc/solr/conf/schema.xml.bak
ln -s
**/usr/lib/ckan/default/src/ckan/ckan/config/solr/schema.xml **
/etc/solr/conf/schema.xml

11) Restart Jetty server:
service jetty restart

12) Check the Solr behavior, it should be running now:
lynx http://localhost:8983/solr/
If the Lynx simple text browser is not installed then:
apt-get install lynx

13) Edit the CKAN configuration file /etc/ckan/default/production.ini and set:
solr_url = http://127.0.0.1:8983/solr

14) Check the PostgreSQL database base configuration:
sudo -u postgres psql –l
Encoding must be **UTF8**

15) Create a CKAN database user if one doesn’t already exist:
sudo -u postgres createuser -S -D -R -P ckan_default

16) Create a new CKAN database, called **ckan_default**, owned by the user **ckan_default** and set the password:
sudo -u postgres createdb -O ckan_default ckan_default -E utf-8
sudo – postgres
psql
ALTER USER ckan_default SET PASSWORD 'somepassword';
\q

17) Edit the CKAN configuration file **/etc/ckan/default/production.ini** and set (examples):
ckan.site_id = some_unique_id
ckan.site_url = http://yourckanhost.yourdomain

sqlalchemy.url =
postgresql://ckan_default:somepassword@localhost/ckan_default

18) Initialize the CKAN database:
sudo ckan db init

19) Restart Apache and Nginx servers:
sudo service apache2 restart
sudo service nginx restart

20) The CKAN sever is ready. It can be tested using a GUI browser to address as it is defined with **ckan.site_url** in the CKAN main configuration file:

http://yourckanhost.yourdomain

21) Next fallows registration of users on the CKAN site and creating organizations and datasets which will be used in response to queries. During the installation and configuration of CKAN server it is possible to run into small or large problems. Additional information and examples are given on the CKAN documents site and Wiki:
http://docs.ckan.org
https://github.com/ckan/ckan/wiki

