EXCELERATE Deliverable 9.1

Project Title:	ELIXIR-EXCELERATE: Fast-track ELIXIR implementation and drive early user exploitation across the life sciences	
Project Acronym:	ELIXIR-EXCELERATE	
Grant agreement no.:	676559	
	H2020-INFRADEV-2014-2015/H2020-INFRADEV-1-2015-1	
Deliverable title:	Requirements analysis document	
WP No.	9	
Lead Beneficiary:	1: EMBL (EBI	
WP Title	Use Case D: ELIXIR framework for secure archiving, dissemination and analysis of human access-controlled data	
Contractual delivery date:	31 August 2016	
Actual delivery date:	31 August 2016	
WP leader:	Helen Parkinson (EMBL-EBI), Arcadi Navarro (CRG)	1: EMBL (EBI)
Partner(s) contributing to this deliverable:	NBIS, UiO, CSC, ELIXIR Hub	

Authors and contributors: Dylan Spalding (EMBL EBI – UK), Helen Parkinson (EMBL EBI – UK), Jordi Rambl (CRG – ES), Niclas Jareborg (NBIS – SW), Abdulrahman Azab (UiO – NO), Ilkka Lappalainen (CSC – FI), Jeff Almaida-King (EMBL EBI – UK), Alexander Senf (EMBL EBI – UK), Saif Ur-Rehman (EMBL EBI – UK)

Table of contents

1. Executive Summary

EXCELERATE WP9 is developing the ELIXIR framework for secure archiving, dissemination and analysis of human access-controlled data. It has three main objectives:

- make more portable data collection tools,
- enable value added services at project, regional or national level,
- extend the data transfer and access authorisation systems developed either by the European Genome-phenome Archive (EGA) or EXCELERATE WP4.

To achieve this, WP9 will leverage and extend the EGA, a controlled-access long-term archive of human identifiable -omics data. Central to delivering these objectives is the Local EGA, which:

- offers a federated model of data access and submission where the data cannot leave a particular jurisdiction for reasons of data protection,
- allows highly accessed data with permission will be mirrored to trusted ELIXIR Nodes to reduce network contention within EGA core services,
- and can host data close to the compute where data transfer times become inhibitive.

Extensive progress has been made with respect to the Local EGA, with a demonstrator already available as a virtual machine (VM) which can easily be run on a wide range of hardware, plus the source code is also available in GitHub[4]. This demonstrator has been used as part of the requirement gathering process. Training has been given on the use of the Local EGA demonstrator via webinar[3].

To facilitate value added services on EGA data, such as data analysis, this document describes the demonstrator EGA in the Cloud (Cloud). This allows users to access data and perform analyses on the data within a secure cloud environment, and along with Local EGA makes extensive use of the ELIXIR Authentication and Authorization Infrastructure[9] (ELIXIR AAI) and data mirroring infrastructure being developed by WP4.

Testing is ongoing with integrating the Resource Entitlement Management System[5] (REMS) with EGA which allows users both to apply for access to data and to be granted access from REMS directly using an ELIXIR identity.

Risk analysis has been performed determining the effects of failures of any of the federated components, and their dependencies both on systems within WP9 and systems external to WP9 on which WP9 is dependent.

Technical discussions have taken place with WP4, including participation in the WP4 F2F meeting in Helsinki. Over the next year continued collaboration with WP4 will focus on the secure transport and processing of human data, and continued development of the ELIXIR AAI to address the needs of WP9.

This document details the purpose, features, and expected interfaces for the complete WP9. It outlines the tasks the system will perform, the constraints under which it operates, and how it reacts in certain circumstances. This document is intended for stakeholders, designers and developers as well as users of the system, and derives from a joint analysis carried out with these groups. This document is a living document that details the current state of the WP9 requirements analysis and as such is subject to change as requirements are added, updated or removed.

2. Project objectives

This deliverable fulfils one of the Ethics requirements of EXCELERATE.

With this deliverable, the project has reached or the deliverable has contributed to the following objectives:

No.	Objective	Yes	No
1	To upgrade and make more portable -omics data collection and submission tools utilizing the European Genome-phenome Archive (EGA) as the core of an ELIXIR community secure data sharing network for - omics data		x
2	Enable value-added services at project specific, regional, or national resources by establishing ELIXIR-wide community facing tools that allow local resource owners and developers to add value to their systems through data and metadata services from the EGA	x	
3	Extend and generalise the system of access authorization management and high volume secure data transfer developed in the EGA project to address the secure data access needs across ELIXIR resources and open new modes of secure data access such as through public and private clouds.		x

3. Delivery and schedule

The delivery is delayed: ☐Yes ☑ No

4. Adjustments made

No adjustments were made.

5. Background information

Background information on this WP as originally indicated in the description of action (DoA) is included here for reference.

Work package number	9	Start date or starting event:	month 1
Work package title	**Use Case D: ELIXIR framework for secure archiving, dissemination and analysis of human access-controlled**		

	data
Lead	Helen Parkinson (EMBL-EBI), Arcadi Navarro (CRG)

Participant number and person months per participant

1 - EMBL (34 PM), 5 - UTARTU (16 PM), 6 - NBIC (0 PM), UMCG (5 PM), 8 - CRG (33 PM), 20 - CSC (27 PM), 24 - UiO (6 PM), 31 - LIU (6 PM)

Objectives

To upgrade and make more portable -omics data collection and submission tools utilizing the European Genome-phenome Archive (EGA) as the core of an ELIXIR community secure data sharing network for - omics data. Tools developed here will support submission of all types of -omics data from human samples consented for biomedical research from disease consortia such as International Cancer Genome Consortium (ICGC), Rare Diseases (Rd-Connect), national cohorts, and biobanks. Emphasis is given to supporting investigator and locally driven research projects with human data consented for biomedical research. To enable these projects, the data submission tool chain will be made more portable and user-friendly with the goal of distributing a common toolset “in-a-box” to enable local and national groups to collect -omics data and meta data in a distributed manner which is consistent across European groups through ELIXIR coordination.

To enable value-added services at project specific, regional, or national resources by establishing ELIXIR-wide community facing tools that allow local resource owners and developers to add value to their systems through data and metadata services from the EGA. For example, local research projects would be enabled to make their data discoverable and searchable, and linked with available -omics data from various sources, by leveraging stable unique EGA identifiers. Further, locally developed project specific data portals will be enabled through defined standard APIs using real time secure data links which allow -omics big data archived in the EGA to be presented in combination with biobanks or cohort data.

To extend and generalise the system of access authorization management and high volume secure data transfer developed in the EGA project to address the secure data access needs across ELIXIR resources and open new modes of secure data access such as through public and private clouds. For example, a trusted ELIXIR Cloud service can receive local copies of selected datasets through a secure data mirroring system and provide access to data and compute to those users that already have data access permissions available from appropriate Data Access Committees stored in the EGA system. The WP will partner first with 2-4 large resource owners to gain the required expertise, document the process in multiple ELIXIR member states and finally to propose a way to scale up these services to match wider European requirements. This WP will also be used to drive creation of the ELSI framework that supports the workflow (WP12).

Description of work and role of partners

This WP delivers the core ELIXIR workflow for long term archive and re-use of human data consented for biomedical research requiring access-control based on a data access agreement and approval process. The workflow supports data submitters and ELIXIR Node coordination on data deposition into the EGA archive in a manner that will maintain data ownership in the hands of the original research data owner, enable data release to authorised individual users from the archive and to partner with downstream secure ELIXIR data analysis platforms. This workflow ad supporting infrastructure will allow the data owners to focus on their unique areas of data generation and analysis expertise while being able to rely on EGA and the ELIXIR infrastructure for their common –omics big data storage, coordination and distribution needs under appropriate legal frameworks. The work described here will leverage the work of other ELIXIR-EXCELERATE Work Packages, for example WP10 to scale each service structure to cover all ELIXIR Nodes

and with WP4 for technical service support, and relies on WP12 to establish the necessary legal framework that supports workflows.

The Workflow can be summarized as:

1. Data preparation, validation, and submission to the EGA making use of common supporting tools and data models (e.g. through Node data Network, WP10). Focus on providing software tools and remote APIs enabling local leadership and customisation within context of specific projects, supported by common ELIXIR coordinated tools and data models.

2. Bidirectional linking and secure data streaming between -omics data archived in EGA and local repositories or data portals that hold further information about the project and samples.

3. Management of user access-rights for release of archived data to authorized researchers under Data Access Agreements using ELIXIR tools, such as the REMS (WP4), that allow resource owners to manage data access rights.

4. Expanded access through ELIXIR partner secure clouds that can host EGA datasets, requiring the provision of metadata and authorization APIs

5. Data synchronization between the main EGA archive and authorized project specific resources and access points, such as compute clouds.

Task 9.1: Enhanced secure data submission tools. (49PM)

This task will update the existing EGA submission tools and documentation to facilitate large-scale data submissions operations, emphasizing local leadership and customization within a common framework.

Partners: ES, EMBL-EBI, FI

Subtask 9.1.1: Support for large scale submission of -omics data and sample metadata to the EGA. (30PM)

Support for large-scale submission of -omics data and sample metadata to the EGA through improved online tools, automated verification, and tools for the application of standard vocabularies to phenotype collection. These tools will make use of table "spread-sheet" based views of data for submitters less comfortable with technologies such as XML. Further tools and reports supporting global EGA stable identifier mappings will allow easier integration with local identifiers, in support of federated global tracking of submitted samples and their derived -omics data.

Subtask 9.1.2: Portable submission toolkit. (21PM)

This task is composed of data format definitions and software components, a "mini-EGA in-a-box" will allow increased local control and coordination of data collection, and allow early validation of standardized data and metadata formats. This implementation provides the practical means for distributed projects to collect access-controlled human biomedical data in a manner that maintains a coordinated data model and dataset registry, enabling federated and a centralized single-point of discovery and access.

Task 9.2: Integrating centralized and distributed projects through transparent access to secure data: enabling local projects within a European wide framework. (40PM)

This task will enable local projects, such as study-specific data portals, local cohort resources, and national bioinformatics hubs by providing developer level APIs and services such that local efforts can efficiently build customized project branded solutions which make use of underlying ELIXIR and EGA tools and data archives.

Partners: ES, EMBL-EBI, FI, EE, NL, NO, SE

Subtask 9.2.1: Support secure integration of EGA data to downstream project client websites. (10PM)

Support secure integration of EGA data and metadata to downstream project client websites by providing new EGA programmatic interfaces that support standardized REST calls and provides results in ELIXIR endorsed formats (WP3 and WP6).

Subtask 9.2.2: Access management workflow support. (10PM)

Support access management workflows by data access committees through ELIXIR for EGA and other projects through developing applications of the Resource Entitlement Management Systems (REMS) expanding on an existing pilot project. This effort is focused on providing tools to delegate management to local projects and ELIXIR Nodes through new administrative roles.

Subtask 9.2.3: ELIXIR and EGA access integration. (20PM)

Specific efforts supporting controlled access -omics data infrastructure for use of partner national cohort studies in terms of submission, permissions management, and local and customized presentation of data under the cohort branding. Services will be tailored to respect the unique policy and data protection requirements of national cohorts, allowing single point of request and download from cohort branded web-pages. Support will be provided for distributed local hosting of datasets, within a common ELIXIR framework, where restrictions exist on the movement or hosting of data based on national borders.

Task 9.3: Federated authentication, large scale data management, and secure clouds in practice. (38PM)

This task is closely linked to the technically focused WP4 that provide the technical solutions required to deliver the outcomes of Task 9.1 and 9.2. In this task, technical components, including high volume secure data transfer and authentication and authorization management, are brought together to make -omics data from EGA and phenotypic data from cohort studies available for secure download, remote API access or from within public or private Cloud-based secure analysis environments. Cloud-based access to the EGA ecosystem provides a new access mode meeting a significant user need from research groups with limited local resources for compute and large-scale reference data storage.

Partners: EMBL-EBI, ES, FI, EE

Subtask 9.3.1: Large scale data mirroring support. (12PM)

Support for automated large scale data mirroring from the EGA archive to the authorized ELIXIR partner local services and cloud compute or HPC providers. This process instantiates concrete data flows based on data transfer technologies in WP4 to track domain specific files, versions of files, confirms transfer success, and tracks files available in different locations. Generic interfaces should provide transparent access to multiple underlying transfer and storage modules (e.g. gridFTP/irods/object store etc.)

Subtask 9.3.2: EGA data access authorization integration. (12PM)

Integrate EGA data access authorizations to local project data portals and Cloud access providers. This is a new service that allows authorized third-party services to programmatically check compliance with the current user data access authorizations from the ELIXIR coordinated repositories such as the EGA database each time user accesses a file in the cloud or other remote service. A first planned project using EGA data within the private, secure, cloud at CSC in Finland will provide our reference implementation.

Subtask 9.3.3: Data access APIs. (14PM)

Develop and implement standard data access APIs to be to used for inter and intra cloud communication and for secure remote REST API access in coordination with the Global Alliance for Genomics and Health (GA4GH). For tasks 1-3 we expect to list a number of updates to the submission tools while we work with the first 2-4 chosen resources. These updates will be prioritised in the scope of this WP. WP4 will provide AAI support for WP 9, and vice versa WP9 will work with WP4 to set the requirements for ELIXIR AAI services. WP9 needs to information on service component availability and this information is expected to be available from technical services registry such as cloud resource allocation, valid EGA data access authorizations, and file mirroring status if data are not yet ready to be used in the cloud. WP12 will Create a set of Legal Frameworks for ELIXIR-related operations that will be integrated within WP9 with the technical solutions devised for particular EGA needs.

Annex 1: WP9 Requirements Analysis Document

Analysis of requirements for an ELIXIR framework for secure archiving, dissemination and analysis of human access-controlled data

Authors and contributors: Dylan Spalding (EMBL EBI – UK), Helen Parkinson (EMBL EBI – UK), Jordi Rambl (CRG – ES), Niclas Jareborg (NBIS – SW), Abdulrahman Azab (UiO – NO), Ilkka Lappalainen (CSC – FI), Jeff Almaida-King (EMBL EBI – UK), Alexander Senf (EMBL EBI – UK), Saif Ur-Rehman (EMBL EBI – UK)

Table of contents

Tables and Figures

1. Executive Summary

EXCELERATE WP9 is developing the ELIXIR framework for secure archiving, dissemination and analysis of human access-controlled data. It has three main objectives:

- make more portable data collection tools,
- enable value added services at project, regional or national level,
- extend the data transfer and access authorisation systems developed either by the European Genome-phenome Archive (EGA) or EXCELERATE WP4.

To achieve this, WP9 will leverage and extend the EGA, a controlled-access long-term archive of human identifiable -omics data. Central to delivering these objectives is the Local EGA, which:

- offers a federated model of data access and submission where the data cannot leave a particular jurisdiction for reasons of data protection,
- allows highly accessed data with permission will be mirrored to trusted ELIXIR Nodes to reduce network contention within EGA core services,
- and can host data close to the compute where data transfer times become inhibitive.

Extensive progress has been made with respect to the Local EGA, with a demonstrator already available as a virtual machine (VM) which can easily be run on a wide range of hardware, plus the source code is also available in GitHub[4]. This demonstrator has been used as part of the requirement gathering process. Training has been given on the use of the Local EGA demonstrator via webinar[3].

To facilitate value added services on EGA data, such as data analysis, this document describes the demonstrator EGA in the Cloud (Cloud). This allows users to access data and perform analyses on the data within a secure cloud environment, and along with Local EGA makes extensive use of the ELIXIR Authentication and Authorization Infrastructure[9] (ELIXIR AAI) and data mirroring infrastructure being developed by WP4.

Testing is ongoing with integrating the Resource Entitlement Management System[5] (REMS) with EGA which allows users both to apply for access to data and to be granted access from REMS directly using an ELIXIR identity.

Risk analysis has been performed determining the effects of failures of any of the federated components, and their dependencies both on systems within WP9 and systems external to WP9 on which WP9 is dependent.

Technical discussions have taken place with WP4, including participation in the WP4 F2F meeting in Helsinki. Over the next year continued collaboration with WP4 will focus on the secure transport and processing of human data, and continued development of the ELIXIR AAI to address the needs of WP9.

This document details the purpose, features, and expected interfaces for the complete WP9. It outlines the tasks the system will perform, the constraints under which it operates, and how it reacts in certain circumstances. This document is intended for stakeholders, designers and developers as well as users of the system, and derives from a joint analysis carried out with these groups. This document is a living document that details the current state of the WP9 requirements analysis and as such is subject to change as requirements are added, updated or removed.

2. Introduction

Controlled access human -omics data, such as whole genome sequencing, exome sequencing and GWAS data can be consented for different uses, and is therefore subject to different data access policies and processes. WP9 delivers the core infrastructure required for long-term archiving and reuse of human data consented for biomedical research. The EGA makes controlled access data available based on data access agreements and approval by an authorised body such as a Data Access Committee (DAC)[1]. The local EGA model developed by WP9 offers a federated model of data access for the following scenarios:

- data cannot leave a country's jurisdiction for reasons of data protection
- data can be hosted close to the compute where data transfer times become inhibitive due to dataset size
- shared ELIXIR Authentication and Authorization Infrastructure (AAI) technology (developed within WP4) is available to identify and authenticate users allowing them to access data for which they have appropriate permissions

Highly accessed data with permission will be mirrored to trusted ELIXIR Nodes to reduce network contention within EGA core services.

By using the Local EGA and EGA in the Cloud, the DAC is able to release data from EGA to authorised individual users to support data analysis on secure ELIXIR data analysis platforms. An example of this is the collaboration with the TraIT project[1], where TraIT data is accessed from EGA and analysed in their private cloud[2].

This approach allows the data controllers to rely on EGA and the ELIXIR infrastructure for their common –omics storage, coordination and distribution infrastructure (which adhere to relevant legal frameworks) so they can focus on data generation and analyses.

The work described here leverages the work of other ELIXIR-EXCELERATE Work Packages, for example the effort of WP10 to scale each service structure to cover all ELIXIR Nodes, and with WP4 services such as the ELIXIR AAI and data transfer task[8] . It also relies on WP12 to establish the necessary legal framework in which each ELIXIR Node will operate a Local EGA Instance.

The EGA is comprised of the software itself, the supporting workflows to manage and process data, and the framework covering ethical, legal and societal issues (ELSI) under which the EGA operates. The EGA is already operated as a partially federated resource by EMBL-EBI (the originator of the resource) and CRG (the first federation partner). By using this technical model supporting data federation supplied with an open source (Apache 2.0 licensed) and freely accessible code base and thus creating the Local EGA, we extend the EGA network to ELIXIR Nodes as well as enhancing the tools available to the ELIXIR partners and the wider community. The existing implementation of EGA at EMBL-EBI and CRG will be referred to as Core EGA (EGAc) in this document, and to differentiate between an instance of the Local EGA and the concept of the Local EGA, an instance will be referred to as EGAl, while the concept of Local EGA will continue to be referred to as Local EGA.

We expect that, initially, each ELIXIR Node involved in WP9 will operate an EGAl. To enable each partner (Estonia, Netherlands, Spain, Finland, Norway, and Sweden) to evaluate their local IT infrastructure and the work necessary to establish an EGAl, we have developed a prototype for developers to test prior to starting collaborative code development. The distribution of this prototype was supported by a webinar[3], supporting code base[4], and interaction with the EGA development team via an online Q&A session and face to face meetings.

Section 2 describes the complete system from the APIs to the Local EGA and Cloud sub-systems, the constraints under which the complete system is expected to operate, the

[1] One or more individuals who grant, revoke, or deny access by users to data under their control

[2] Person or group who submit data to EGA for archival

[3] Subtasks 9.3.1 Large scale data mirroring support and 9.3.2 EGA data access authorization integration

users who will use the system and functions of the system and subsystems. Section 3 describes the requirements which the system must meet. These are described both from a user perspective and a technical perspective with respect to other work packages within EXCELERATE.

3. WP9 components and processes

WP9 components and processes corresponding to all tasks and objectives consist of five APIs, two portals, and two sub-systems (shown in Figure 1). Requirements gathering was performed for each of these components. The application programming interfaces (APIs) have dependencies on other work packages, for example the AAI, permissions, and data transfer API have dependencies on WP4 (Compute, Data access and exchange services), while the metadata API has dependencies on WP5 (Interoperability) and WP10 (Node Capacity). Additionally, the APIs have dependencies on each other: for example, all APIs have dependencies on the ELIXIR AAI API as a core part of subtask 9.3.2. The focus is on providing software tools and remote APIs enabling local leadership and customisation within the context of specific projects, supported by common ELIXIR coordinated tools and data models.

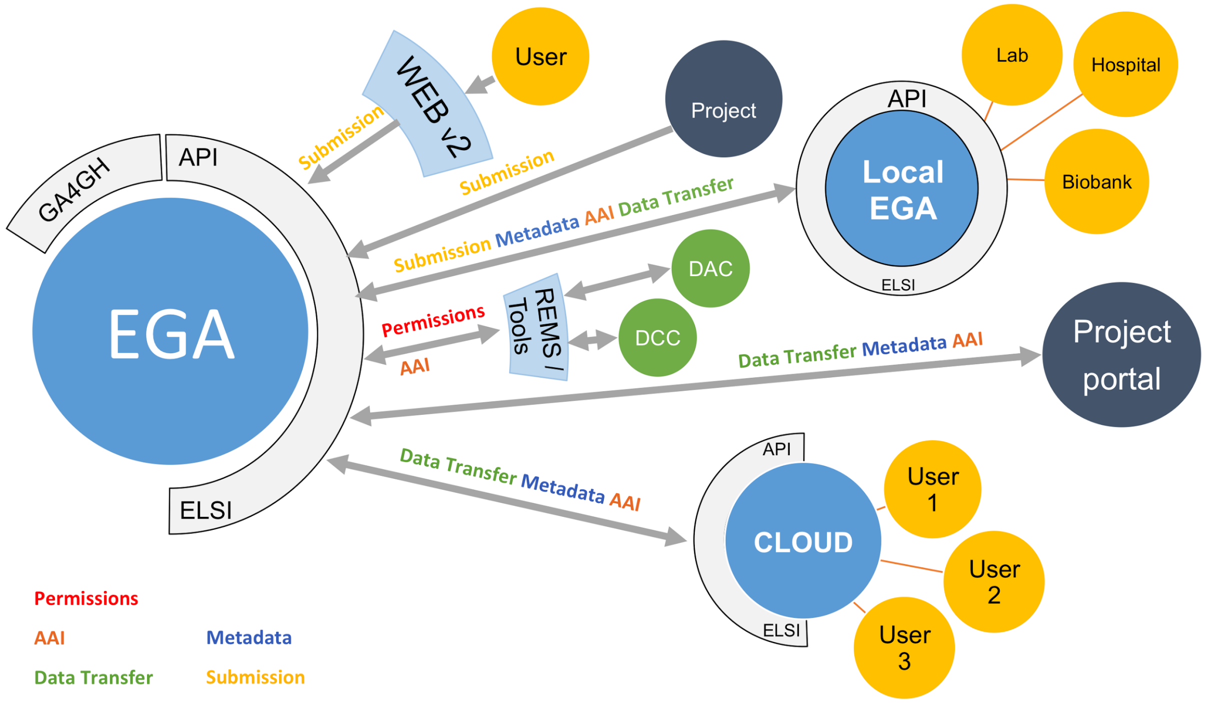

Figure 1 Overview of WP9 components and processes, including the five separate APIs, two portals - one for submissions and one for applying permissions - and two sub-systems, the Local EGA and the use of EGA data in a secure cloud environment (Cloud). The APIs have been split on purpose as they are all utilised by more than one function or use-case

3.1. Data Transfer API

The data transfer API supports the process necessary to transfer large quantities of human identifiable data securely between EGA and Local EGA or Cloud. In this document, 'data' refers either to data in general or specifically the data contained within the files that the EGA archives and which represent the controlled access data EGA distributes. The specific meaning will be clear by the context in which the term is used.

This process must be scalable to the large quantity of genomic data distributed by the EGA (petabytes per year) and adaptable to both long distance and local data transfer to ensure consistency of the API for both use cases. This API must also be able to determine presence/absence and status of a file at any Local EGA instance and will be delivered in collaboration with WP4 as it relies on the ELIXIR AAI and WP4 data mirroring infrastructure.

3.2. Metadata API

This API facilitates data discovery by the querying of metadata, and supports querying and syncing of metadata between Nodes. In this document, metadata refers to the public information describing the controlled access data or files held within EGA, such as the type of file or the high-level phenotype (such as type 2 diabetes for example) which relates to a study. These metadata describe the process by which the files were obtained and relevant information about the purpose of obtaining the associated files, while the files themselves and hence the data within the files are controlled access and identifiable. For the API, an example would be retrieving all file accessions for a particular study or dataset, or finding all datasets which relate to type 2 diabetes. As such this is the API which will support the public website and portals, as these data are not under controlled access and the data represent high level information about studies, organisations, terms of access, and publications - the metadata as previously defined. The metadata API should adhere to the WP5 recommendations on use of identifiers, ontologies and emerging best practices in API design as well as FAIR principles.

3.3. Submissions API

This is an API that allows programmatic submission of data to EGA, either via an instance of the Local EGA (EGAl) or the Core EGA (EGAc). This API must be able to support a wide range of users, from large consortia producing terabytes of data (such as the International Cancer Genome Consortium, ICGC) to single submissions from individual laboratories. Therefore, the API must be accessible and usable by programmatic submitters, such as ICGC, while also being able to support the graphical submission portal to facilitate smaller scale submissions.

3.4. Authorisation and Authentication Infrastructure

This is the mechanism through which the identity of a user is confirmed. The ELIXIR Authorisation and Authentication Infrastructure (AAI) is being developed in the context of EXCELERATE WP4 using WP9 as a use-case for access control depending on identity. For the purpose of this document, an identity is the online identity of a user (with a role of submitter, requestor, or DAC member), and this identity can be linked to an unique ELIXIR identity. An ELIXIR identity is unique and common throughout ELIXIR resources and services. It can be linked to other identities, such as Google or institutional accounts. The AAI will use the ELIXIR identity and is the infrastructure through which this identity is passed between Nodes and ELIXIR Services.

3.5. Permissions API

The Permissions API allows querying and determination of access permissions in EGA. Permissions are granted by the DAC specified for the dataset in question to the user requesting access to the dataset. The Permissions API also provides the terms of access, which may for example exclude use of the data by commercial organisations. The Resource Entitlement Management System (REMS), a portal that allows both users and

DACs to login and apply for or grant / deny access to datasets, respectively, is an example of a portal that determines many of the requirements of a permissions API (see Section 2.7.1).

3.6. Portals

The portals are graphical user interfaces (GUIs) which provide 'front end' graphical access to services provided as part of WP9. The two portals that are within the scope of this deliverable are the submissions portal and DAC admin tools, though both must interface with the permissions API and as such requirements analysis for this use case is ongoing.

- As an existing system, REMS has its own interface and technical requirements which are being implemented as part of the permissions API, and further requirements analysis is planned on the REMS system itself.
- The submissions portal is an improved graphical front end to the submissions API, which facilitates submissions from non-programmatic submitters. This will also become part of the EGAI so the user experience (UX) of data submission to the EGAI operated by any Node will be as similar as possible.

To facilitate EGA access management and integrate this with other ELIXIR resources, EGA has collaborated with CSC to deliver a solution using the Resource Entitlement Management System (REMS)[5]. The REMS supports electronic administration of access rights to resources, such as research datasets. Applicants can use their federated identity as authentication to the REMS, complete the data access application, commit to license terms, and submit the application to the appropriate resource. The REMS system circulates the application to the resource owner or denoted representative(s) in support of the data access granting process, and provides a reporting function for applications and data access rights granted. This allows the responsible DAC to administer permissions for specific datasets from within the REMS environment, allowing a single point from which to administer resource permissions.

In addition, REMS allows users to electronically apply for access to datasets across different ELIXIR resources. To facilitate this at EGA, REMS uses the first iteration of the permissions API, which indicates the datasets a user has access to, whether those permissions have been revoked, and whether those permissions were granted via REMS or the EGA's own DAC admin tools. REMS uses this information to ensure actions taken by either REMS or within EGA itself are not contradictory. To apply updated permissions, REMS will generate an encrypted and signed XML which is submitted to EGA to update the permissions as defined by the DAC.

Integrating REMS permissions management to EGA is currently in testing and, as it applies to both the DAC and the requestor once the technical implementation is complete, we will be requesting UX feedback from the DAC and users requesting access to specific datasets. Further developments here will be to design and extend the permissions API to allow the application of permissions from REMS without resorting to XML once the use-case has been fully scoped during the test phase. Dependencies here include 2-factor authentication and LoA for users to be provided by the AAI.

Once the permissions API has been developed, the DAC admin tools will be improved to take advantage of the permissions API functionality, but it is expected that the use of these tools will reduce as DACs and users migrate to REMS. However, these tools will remain available for the instances where an EGA user does not have an ELIXIR ID, and to support the transition to ELIXIR IDs. Provision will also be made for programmatic access to the permissions API for consortia who wish to manage permissions from their own portal.

3.7. Local EGA

A self-contained sub-system which facilitates the upload, archival, and distribution of controlled access data at a federated installation of EGA, for example at an ELIXIR Node. A demonstrator is provided as part of this deliverable to elicit requirements from the EGA Developer class of users.

3.8. Cloud

In the context of WP9, this is a secure cloud instance hosting EGA data with associated infrastructure to enable high throughput data access with minimal delay on up to date files. Additionally, the cloud instance provides an infrastructure through which geographically distributed users can process and access the same files on the same compute. This can improve both security and accessibility as the files do not necessarily need to be distributed to multiple destinations. The processes developed here will enable synchronisation and monitoring of data throughout a network of ELIXIR Nodes. EGA data may be hosted in a secure cloud by one or more Nodes for their local use. Benefits of cloud hosting include rapid access to data (especially when coupled with a local instance of Local EGA), reduction of I/O as data can be accessed by multiple users without repeated downloads, and access to flexible compute on a pay per use basis. Services are typically virtualised and are therefore flexible for many applications. ELSI issues related to cloud hosting will be addressed in later deliverables and tasks for WP9.

4. User profiles and design considerations

4.1. User profiles

We have identified the following user profiles, and future work will test these profiles against the requirements and implementation. User profiles allow the development team to identify tasks and skills of each group of users, to identify individuals representing these groups, to design UX testing to be conducted with each of these groups, and to test and document the system developed against each of the user groups. The EGA user profiles are described briefly below. They will be used in the next phase of the project for testing and iteration of the requirements described here as well as for the design and testing of training materials for WP9.

EGA Developer (EGAd): has technical skills; tasked with installing and running the Local EGA (for example at an ELIXIR Node)

EGA Data Provider (Bioinformatics) (DP1): - a bioinformatician depositing data at EGA. Typically with technical skills; deposits data programmatically

EGA Data Provider (Biological) (DP2): biologists, typically with biomedical research expertise and limited informatics skills; deposits data in the EGA, via a GUI

EGA Helpdesk personnel (EGAh): has biological skills and scientific research experience; responsible for answering EGA user queries, supports data loading and access tasks within EGA and supports both Biological (DP2) and Bioinformatic (DP1) EGA Data Providers[2]

EGA Data user (Bioinformatic) (DU1): a bioinformatician; applies for access to EGA and is able to use APIs/command line tools for data downloads

[2] Person or group who submit data to EGA for archival

EGA Data User (Biological) (DU2): biologists, typically with biomedical research expertise and limited informatics skills; applies for access to EGA data and requires support in accessing data via non-graphical means

EGA Scientific Stakeholder (SS): the community of EGA users who make strategic decisions related to a Node's use of, and commitment to the Local EGA (for example ELIXIR Human Data Coordinator)

EGA Data Access Committee Administrator (DAC): a user who considers requests for data access and grants access via a GUI to one or more users; may represent a committee of individuals

REMS User (REMS): A user who accesses the system via REMS, either a user requesting access or a DAC granting permissions.

EGA User (EGA): Any user of EGA services for whom EGA must utilise an external service to complete their request.

3.2 EGA Local Design Considerations

Regulatory Policies: The system must conform to the ELIXIR Ethics Policy, other European regulatory policies and best practices in the sharing of identifiable data, and allow for jurisdictional policies to be applied on a local level, in effect ensuring the Local EGA or Cloud components can conform to disparate regulation and/or policies.

Hardware Limitations: As WP9 components may be deployed on a node's existing hardware, federated components (Local EGA and Cloud) of the system must be deployable on a large range of hardware and minimal system requirements should be placed on the Node hosting the EGAl instance. The requirement for interfacing to existing hardware adds additional constraints to the federated components, with examples listed in Table 1. Each federated component must be robust in terms of maintaining data consistency in the case of communication failures with the EGAc or between the EGAc and EGAl instances. A federated component should maintain operation while communication is down and ensure data consistency is regained once communication is restored. Robustness in the event of failure of one or more EGAl instances is a requirement for a federated system and must be considered in the design, deployment and testing of the Local EGA. This can be achieved by working with EXCELERATE WP4 on utilising the European Grid Infrastructure (EGI).

Interfaces: metadata should meet the FAIR principles[7] (Findable, Accessible, Interoperable, Re-Usable) and adhere to WP5 best practices and recommendations. Therefore, a common standard supporting the FAIR principles will be applied to describe metadata throughout the system across Local EGA implementations in collaboration with WP5. This standard should be capable of interfacing with other metadata standards, for example those deployed by the Global Alliance for Genomics and Health, and be auditable by ELIXIR to ensure that the resource delivered meets the FAIR principles. This work will be supported by the activities of ELIXIR WP5 (interoperability). The AAI and data transfer components should re-use the tools developed by WP4 where possible, and the requirements from WP9 communicated to WP4 to facilitate this.

Parallel Operation: All Nodes should be able to perform functions related to the submission, archival, and distribution of data in near real-time without noticeable delays. This will exclude validation processes which require extended time to perform. These processes will be completed and reported to the submitter and Local EGA once complete.

Logging Activities: All operations performed within the system must be tracked and be subject to possible audit. For example every database operation relating to file submission, archiving, or access is logged (effectively every select, insert, delete, or create operation), and these logs will be stored in a common format. These logs include the file affected, the user performing the action, the IP address (for remote access) from which the action originated, plus history logs for all non-database operations. For federated components these logs must be within the federated component, but all operations that affect the complete system must be accessible from other Nodes within the system. For example, this would be availability of the complete Local EGA instance itself, or of files which are distributed between Local EGAs for capacity building, or attacks on a Local EGA, such as Denial of Service (DoS) attacks. A common format developed to store these logs allows the same tools to integrate the logs between different nodes. Security logs of activities affecting controlled access data that exists with a single EGAI must be stored solely within the EGAI along with the affected data. Aggregate statistics, such as total downloads for a file, Local EGA uptime, number of files by type etc., will be accessible. These requirements will be passed to the WP4 developed monitoring service.

Criticality of System: The system as a whole has a high level of criticality on failure due to damage to organizational reputation and business stoppage. In addition, similarly to EGA, data storage at individual EGAIs where the data has not been mirrored to an alternative EGAI would be regarded as the highest criticality level due to the possibility of irreversible data loss. In general, the federated nature of the complete system allows for a lower criticality level for each individual EGAI compared to a single system as some level of function will be maintained if an individual EGAI becomes unresponsive.

Safety and Security: All controlled access data must always be encrypted when stored on disk irrespective of whether these data are treated as long-term data mirrors or short term caches or scratch data. Additionally for cached or scratch data these data must be removed from the filesystem once all associated virtual machines have been removed. WP9 is also working with WP4 to define the processes critical to data security.

Table 1 List of the diverse hardware requirements for EGAI at different ELIXIR Nodes

Node	File System	Compute
EBI	Cleversafe Object store Tape backup	60 Node compute cluster running CentOS 7
Estonia		
Finland	Likely: CEPH blockstorage for computation via cloud middleware Tape for local EGA	Projects can apply for computing capacity CSC operates OpenStack clouds
Netherlands		
Norway	Hitachi NAS for the Cloud, and BeeGFS for the HPC cluster IBM BSM for archiving (Tape)	Slurm HPC cluster, 72 Nodes with 20 CPU cores each and 61.5 GiB RAM, and two Nodes with 32 cores and 1008 GiB RAM running CentOS 6.5.Jobs can be submitted from the Cloud VMs to the cluster
Spain	IBM GPFS (High Performance Storage Area Network (SAN)-based storage) IBM TSM (Tape-based Storage) IBM HSM and HDFS file-systems (not yet bought and set up)	Hardware not yet bought and set up, we cannot yet concrete number of cores or underlying technology. Same for scheduling technology and operative system, both to be determined
Sweden	To Be Determined	To Be Determined

4.2. Design Decisions and Dependencies

The system will be built upon the following assumptions:

- ELIXIR will provide and maintain the required AAI system via Compute Platform
- A suitable high performance network will be available at all ELIXIR Nodes.
- For the each EGAI all hardware will be able to run Docker containers, or communicate with hardware that runs Docker containers.

5. WP9 Requirements

Requirements below are specified on either a per-API basis, or portal / sub-system basis:

5.1. Requirements for the Permissions API

An API that returns both the permissions (access to datasets) a user has been granted and confirms the identity of the user. Currently, EGA has a permissions REST API which already implements the requirements marked with 'c'.

Table 2 User requirements for the Permissions API

Action	Response	User	Completed
Does user EGAW001 have access to dataset EGAD001	Boolean	DAC, EGAh,REMS	c
Get all permissions for user EGAW001	List of datasets	DAC, EGAh,REMS	c
Get all Users for a dataset	List of accounts	DAC, EGAh,REMS	c
Get all allowed objects (of a specific type) for user EGAW001	List of DACs/studies/ samples/...	DAC, EGAh	
Does user EGAW001 have access to object Y	Boolean	DAC, EGAh	
Remove permissions for user EGAW001 from dataset EGAD001	Boolean	DAC, EGAh	
Hold access to all objects for user EGAW001	Boolean	DAC, EGAh	
Restore access to all objects authorised immediately previously to hold said access	Boolean	DAC, EGAh	
Has user EGAW001 been revoked access to dataset EGAD001	Boolean	DAC, EGAh,REMS	c
Create group EGAG001	Boolean	DAC, EGAh	
Add dataset EGAD001 to group EGAG001	Boolean	DAC, EGAh	
Remove dataset EGAD001 from group EGAG001	Boolean	DAC, EGAh	
Remove group EGAD001	Boolean	DAC, EGAh	
Does user EGAW001 have access to group EGAG001	Boolean	DAC, EGAh	
Grant access to group EGAG001 for user EGAW001	Boolean	DAC, EGAh	
Revoke access to group EGAG001 for user EGAW001	Boolean	DAC, EGAh	

5.2. Requirements for the AAI

To integrate access throughout ELIXIR, and to allow the use of ELIXIR permissions tools such as REMS, EGAI must understand an ELIXIR ID and be able to authenticate that this identifier is valid at a specific point in time. EGAI will also want to determine the LoA of a user via an ELIXIR ID. The EGAI will be required to communicate the ELIXIR or EGA ID of the user to EGAc, where the permissions will be returned. These permissions will be returned in either binary responses to queries asking if a user has access to a specific dataset, a list of datasets to which a user has access, or a list of users who have access to a dataset. Some example queries are listed below. Note that, as the AAI is a service provided by EXCELERATE WP4, the User in these cases is EGA.

Table 3 User requirements for the AAI

Action	Response	User
Does requestor have ELIXIR ID?	ELIXIR ID	EGA
Does requestor without ELIXIR ID (i.e. just EGA or Local ID) have access to dataset EGAD001	Boolean	EGA
Does requestor with ELIXIR ID have access to dataset EGAD001	Boolean	EGA
What level of assurance does user EGAW001 have?	LoA	EGA
What level of assurance does dataset EGAD001 require for access	LoA	EGA
What level of assurance does DAC EGAC001 require to administer permissions?	LoA	EGA
Is user X a 'bona fide' researcher according to ELIXIR?	Boolean	EGA

5.3. Requirements for the Submissions API

This API relates to the submission of data to EGA (following the data model in Appendix 2, Figure 7) and queries which improve or simplify this process.

Table 4 User requirements for the Submissions API

Action	**Response**	**User**
Submit an object or group of objects	Boolean	DP1/2,EGAh
List all datasets I submitted	List of datasets	DP1/2
Map study to samples	Show all samples linked to a study	DP1/2,DU1/2
Show unlinked samples	Return samples not linked to run/analysis	DP1/2,EGAh
Show unlinked runs	Return runs not linked to a dataset	DP1/2,EGAh
Show unliked analysis	Show analysis not linked to a dataset	DP1/2,EGAh
Show unlinked DAC	Show a DAC not linked to a policy	DAC,EGAh
show unlinked policy	Show a policy not linked to a dataset	DP1/2,EGAh,DAC
Files unlinked	Files in submission table not linked to run/analysis	EGAh,EGAd
List submission contacts	Idenitify authorised submitters associated with an account	DP1,EGAh
Add additional validations in form of an XML schema	Schema has been found and imported	DP1,EGAh,EGAd
Set a list of controlled vocabularies linked to an XML	List has been validated and accepted	DP1,EGAh,EGAd

element		
Set a source for an identifier validation	Source has accepted a request from EGA	DP1,EGAh,EGAd
Link a given ontology to an element	Ontology provider localized and tested	DP1,EGAh,EGAd
(Many other endpoints to validate who have been submitting under such project and if everything is correct)		DP1,EGAh,EGAd
List all objects (of a specific type) I submitted	List of datasets/DACs/studies/...	DP1/2,EGAh,DU1/2, EGAd
Withdraw object X from distribution	Boolean	DAC,EGAh,EGAd,DP 1/2
Update object X and automatically increment the metadata version	New Version	DAC,EGAh,EGAd,DP 1/2
Remove sample EGAN001 and automatically update all associated objects, incrementing the sample version in the process. Remove access to all files referencing the sample and delete the files and all references to them excluding the original file accession.	Boolean	DAC,EGAh,EGAd,DP 1/2
Generate a report for a created dataset	Map of samples, experiments and runs or analysis, files and Policy/DAC for each generated datasets	DAC,EGAh,EGAd,DP 1/2
Hold file EGAF002 until DATE	Boolean	DAC,EGAh,EGAd,DP 1/2

5.4. Requirements for the Data Transfer API

The Data Transfer API is required to facilitate movement and tracking of the files between the individual EGAI. This API has a dependency on WP4 which will deliver the ELIXIR-wide data mirroring/transfer solution. WP9 requirements are listed in Table 5, which may be solely or partly provided by WP4. For requirements outside the scope of the WP4 data transfer/ mirroring responsibility, WP9 will develop additional complementary processes to support the requirements.

Table 5 User requirements for the Data Transfer API

Action	**Response**	**User**
Get file EGAF001	ACK	DU1
Syncronise file EGAF001	ACK	Local, Cloud, Project,EGAh,EGAd

Remove EGAF001 from distribution	Boolean	Local, Cloud, Project,EGAh,EGAd,DP1/2,DAC
Add file EGAF002 to distribution	Boolean	Local, Cloud, Project,EGAh,EGAd,DP1/2
Delete file EGAF002 from all Nodes	Boolean	Local, Cloud, Project,EGAh,EGAd,DP1/2,DAC
Hold file EGAF002 until DATE	Boolean	Local, Cloud, Project,EGAh,EGAd
Where is EGAF001 located?	Boolean	Local, Cloud, Project,EGAh,EGAd,DP1/2,DAC
What is the size of EGAF0002?	Boolean	Local, Cloud, Project,EGAh,EGAd,DP1/2,DU1/2,DAC
Is EGAF0002 available?	Boolean	Local, Cloud, Project,EGAh,EGAd,DP1/2,DU1/2,DAC
How many times has EGAF0002 been transfered?	Boolean	Local, Cloud, Project,EGAh,EGAd,DP1/2,DU1/2,DAC
Count transfers of file EGAF0002 by location	Boolean	Local, Cloud, Project,EGAh,EGAd,DP1/2,DAC
Successful, failed, paused, in progress transfers for file EGAF00002	Boolean	Local, Cloud, Project,EGAh,EGAd
Which location has the file EGAF0002 and the least data transfer load?	Boolean	Local, Cloud, Project,EGAh,EGAd,DU1/2
Which locations are responding to requests?	Boolean	Local, Cloud, Project,EGAh,EGAd

5.5. Requirements for the Metadata API

The Metadata API is a service that allows retrieval of all or some of the metadata associated with a specific object or objects at EGA. The service should return the data in a user specified format, for example XML or json, which would be an additional optional parameter specified on request for metadata. The service should accept queries to find associated or linked objects and adhere to FAIR principles. It is to be developed by WP9, with extensive input from both WP5 interoperability and WP10 (data nodes network task). The requirements are listed in Table 7.

Table 6 User requirements for the Metadata API

Action	Response	User
List all studies with sample EGAN01	List of studies	EGAh,DU1/2,DAC,DP1/2
Query study containing sample with a particular tag, e.g Cancer	List of studies	DU1/2,DP1/2
Query study with a given technology, for meta /cross platform studies	List of studies	DU1/2,DP1/2
Fetch studies by Pubmed id	List of studies	EGAh,DU1/2,DP1/2
Fetch Datasets with sample EGAN	List of datasets	EGAh,DU1/2,DAC,DP1/2

Fetch datasets with a given sequencing depth/quality score	List of datasets	DU1/2,DP1/2
Fetch samples from the same subject e.g (Disease/matched normal)	List of Samples	DU1/2,DP1/2
Fetch datasets by experiment type (transcriptomic/genomic etc)	List of datasets	DU1/2,DP1/2
Fetch all data from Consortia XYZ	JSON	DU1/2,DP1/2,DAC,EGAh
List of all files associated to a sample and vice versa	List of Samples / Files	DU1/2,DP1/2,DAC,EGAh
List files based on file-type, associated object, or not of file-type or any combination of these	List of files	DU1/2,DP1/2,DAC,EGAh
Show all released datasets	Return release status of submitted objects	DP1/2,DU1/2
Map sample to experiment	Show all samples linked to a experiment	DP1/2,DU1/2
Map samples to analysis	Show all samples linked to analysis	DP1/2,DU1/2
Map experiment to run	Show all experiments linked to run	DP1/2,DU1/2
Map run to dataset	Show all runs linked to dataset	DP1/2,DU1/2
Map analysis to dataset	Show all analysis linked to dataset	DP1/2,DU1/2
Map dataset to DAC	Show all datasets linked to DAC	DP1/2,DU1/2
Output object accessions	Return accessions for all object types	DP1/2,DU1/2
Identify analysis/run from filename	Return run or analysis using a filename	DP1/2,EGAh

5.6. Requirements for the Local EGA

Each user story for the Local EGA was ranked using the MoSCoW method, where users were given the option to rank stories using the following priorities:

M - Must Have
S - Should Have
C - Could Have
W - Would have but not this time

Four queries were performed with users, one at the 2016 ELIXIR All hands meeting in Barcelona and one each for the ELIXIR Nodes in Sweden, Finland, and Norway. The results of each were averaged over each user score and listed in Table 8. To differentiate between an instance of the Local EGA and the concept of the Local EGA, an instance will be referred to as EGAl, with the Core EGA being referred to as EGAc, while the concept of Local EGA will continue to be referred to as Local EGA.

Table 7 **MoSCow ranked stories for Local EGA requirements analysis**

ID	User Story	Mean Importance	Notes
US1	As a data owner, I want to upload raw data files, thus they are kept safe	Must	Local EGA Node would provide a file transfer service (FTP, Aspera, Globus…) that will be managed outside the domain boundaries of the Local EGA solution, although linked to it.
US2	As a data owner, I want to submit the metadata describing files, thus they are correctly annotated	Must	Initially, the metadata will follow the current EGA standard (SRA, currently). This metadata is public. Metadata would be submitted to EGAc. Further improvements would be inherited from other tasks in WP9 and follow WP5 recommendations.
US3	As a data user, I want to know which information is available at EGAl, thus I can request access to it	Must	Initially, the discovery will happen at EGAc, because metadata is submitted there
US6	As an EGAl admin, I want to allow particular users to submit data, thus I can control who is submitting	Must	In the first iteration, control of user accounts accessing the upload boxes (i.e. FTP server) must be shared with the EGAl domain, being Local, EGAc or both. (See item 10)
US7	As an EGAl admin, I want to store files in a secure way, thus I can guarantee privacy	Must	A Vault-like area must exist at each Local EGA
US8	As an EGAl admin, I want to store files in a safe way, thus I can guarantee safe storage	Must	Solution would NOT force to store at least two copies of each file, safety is a “local” responsibility.
US10	As a data steward, I want to grant or deny access to users requesting access, thus I can control access to it	Must	AutN/AutZ will initially be provided by EGAc
US16	As an EGAl admin, I want to control users accessing the system, thus I can provide the expected level of access security	Must	Access control would not be delegated to local third parties, i.e. an LDAP administrator
US11	As an EGAl admin or a data steward, I want to know which users have access to a file, and vice versa, thus I can manage them	Should	
US5	As a data user, I want to get data from the files stored, thus I can process them at a local facility	Should	Data can be distributed if a given set of conditions is matched. Such conditions depend on every country, institution or indeed project.

US9	As an EGAI admin, I want to manage the life cycle of files, thus I can optimize storage usage	Should	We are assuming that, initially, every Local EGA will provide enough storage space to do not require “guided” intelligent management
US13	As an EGAI admin, I want to QC submitted files, thus I can be sure that files are correct in the Archive	Should	At iteration 1, just file md5 checking would be provided
US15	As the EGA Core, I want to be able to get metadata from files, thus I can leverage them for discovery	Could	File metadata must be gathered at Local EGAs and sent to the EGAc
US19	Coordination between EGAI and Core on data submitters required	Could	
US20	Current EGA Core Data Access Committee model applies to all Nodes	Could	
US22	As a Requestor, I want to have one point of entry to search for, and request access to, data sets in the whole of EGA (core and local Nodes), so that I don't waste time searching for data sets at multiple EGA instances	Could	
US4	As a data user, I want to get access to the data stored in the files, thus I can browse them	Could	Refers to getting access to parts of files (“slices”) without having to download the whole data set
US12	As a data owner, I want to connect my LIMS system to EGAI, thus I can submit metadata automatically	Could	
US14	As an EGAI admin, I want to apply some management techniques to files (i.e. compression), thus I can optimize storage	Could	
US18	As a Biobank, I want to link files to their corresponding samples, thus I can provide richer information to my users	Could	Just adding a mechanism to check that the ID is valid (note: this feature would be probably part of other WP9 tasks already)
US21	As an EGAI admin, I want a (local) curation interface, so that the local submission process can be facilitated	Could	
US1	As the EGA Core, I want to	Won't	There are security regulations which not

7	know which encryption keys are applied to each file, thus I can provide this information to relevant stakeholders		allow a global party to access privacy related data of users

As part of the requirements gathering process, a Local EGA demonstrator virtual machine (VM) has been developed. This allows a user to install a functional version of a Local EGA on their own hardware using a downloadable VM image or to build a Local EGA by downloading and installing a set of Docker images hosted on the ELIXIR GitHub[3]. Swimlane diagrams of the Local EGA demonstrator for the 3 main processes - submission, archival, and distribution - are shown in Figures 2-4 below.

The aim of the Local EGA demonstrator is to allow prospective users to learn the processes EGA is required to perform in a hands-on way, and investigate issues relating to the use of Docker and VMs interfacing with their own hardware. Local EGA must be as easily installed and hardware independent as possible, hence the utilisation of both VM and Docker technology for this.

The Local EGA demonstrator is a fully functional stand-alone EGA allowing users to:

- Upload data to a secure location via FTP
- Insert metadata corresponding to the current EGA metadata model into a database and linking these data to a set of files
- Process the uploaded files into an archive:
    - Check the file encrypted md5 value
    - Decrypt the file and check the unencrypted md5 value
    - Re-encrypt the file with a Local EGA specified key
    - Move the file to the archive location
- Allow the file to be downloaded by authorised users using the EGA Downloader.

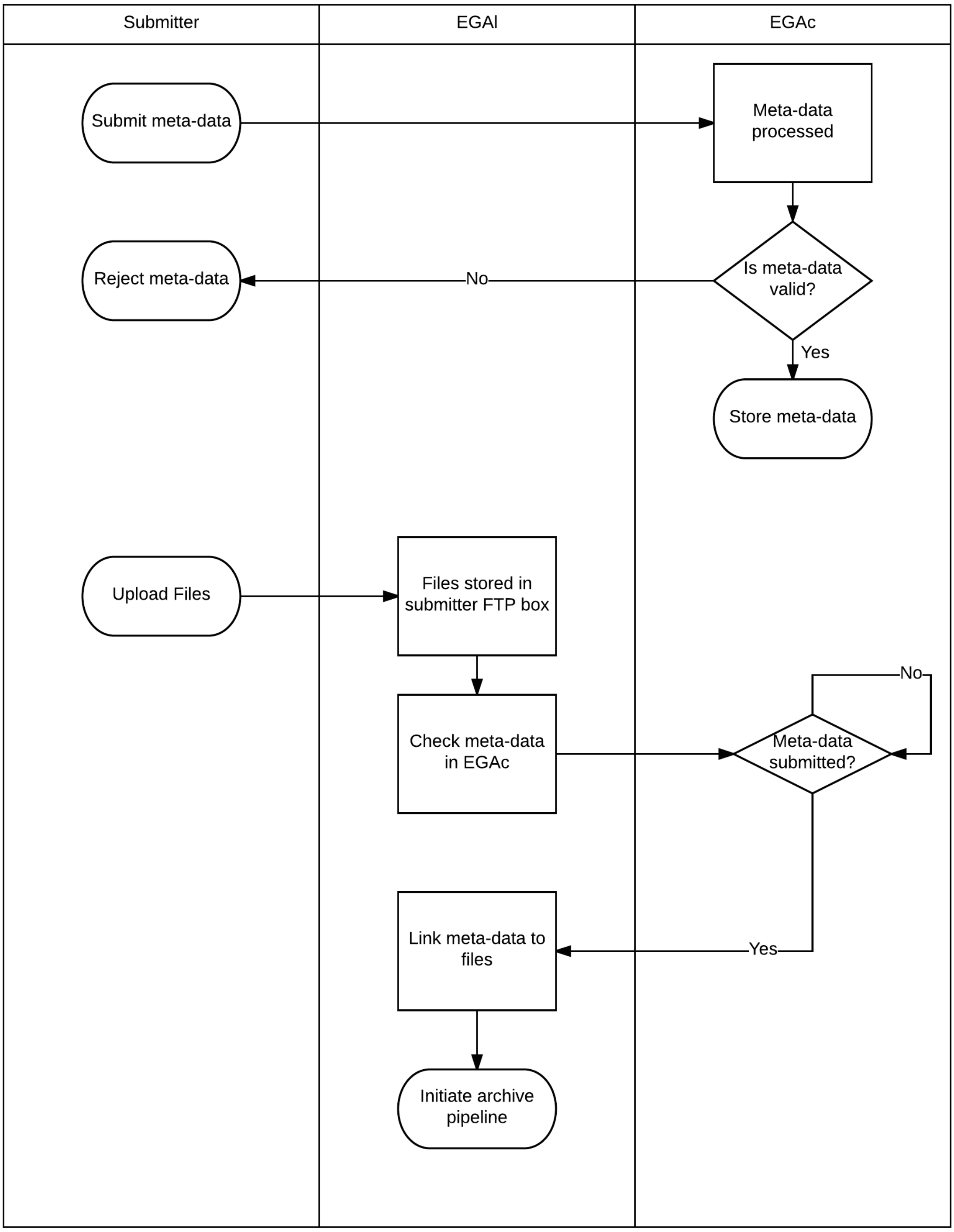

Figure 2 Swimlane diagram representing the main processes and dependencies between EGA core (EGAc) and Local EGA (EGAl) for the submission of files and metadata

The diagram indicates that the system is agnostic to having metadata submitted prior to or after file submission, although the archival process is only executed once validated metadata and the associated files are submitted. In the first iteration of EGAl, metadata will be submitted to EGAc to simplify the metadata queries at a single Node, so EGAl will query EGAc using the metadata API. To improve the ability to query controlled-access metadata, later iterations will distribute metadata to EGAl and the query will be federated across all EGAl instances. This will allow queries on controlled-access metadata to be performed.

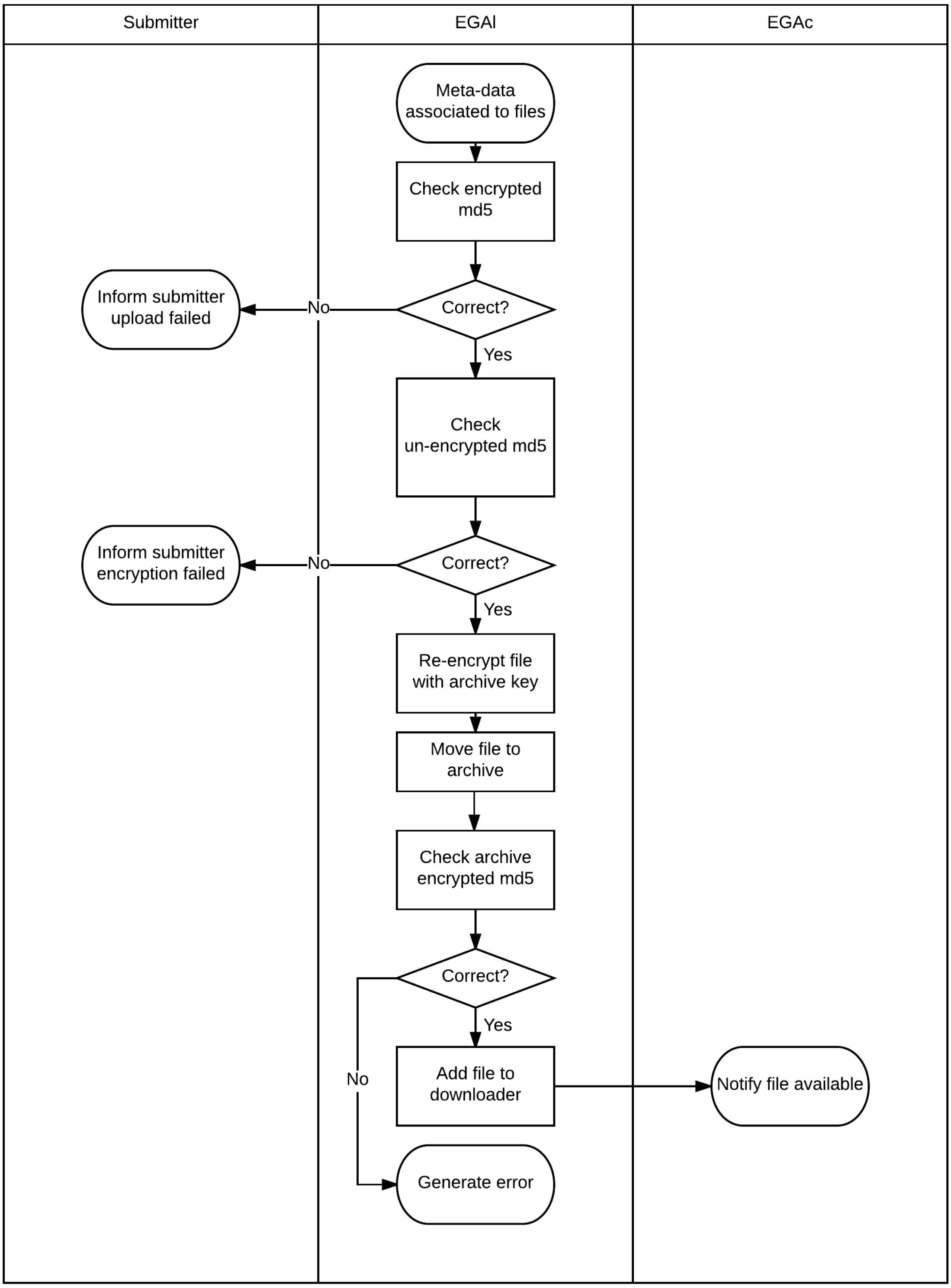

Figure 3 Swimlane diagram outlining the EGA Local (EGAl) archival process

The process is designed for maximum containment within EGAl, with the exception of notifying EGA core (EGAc) when a file is successfully archived. This notification ensures that EGAc can make the files accessible by search and hence direct users to download these from the correct locations.

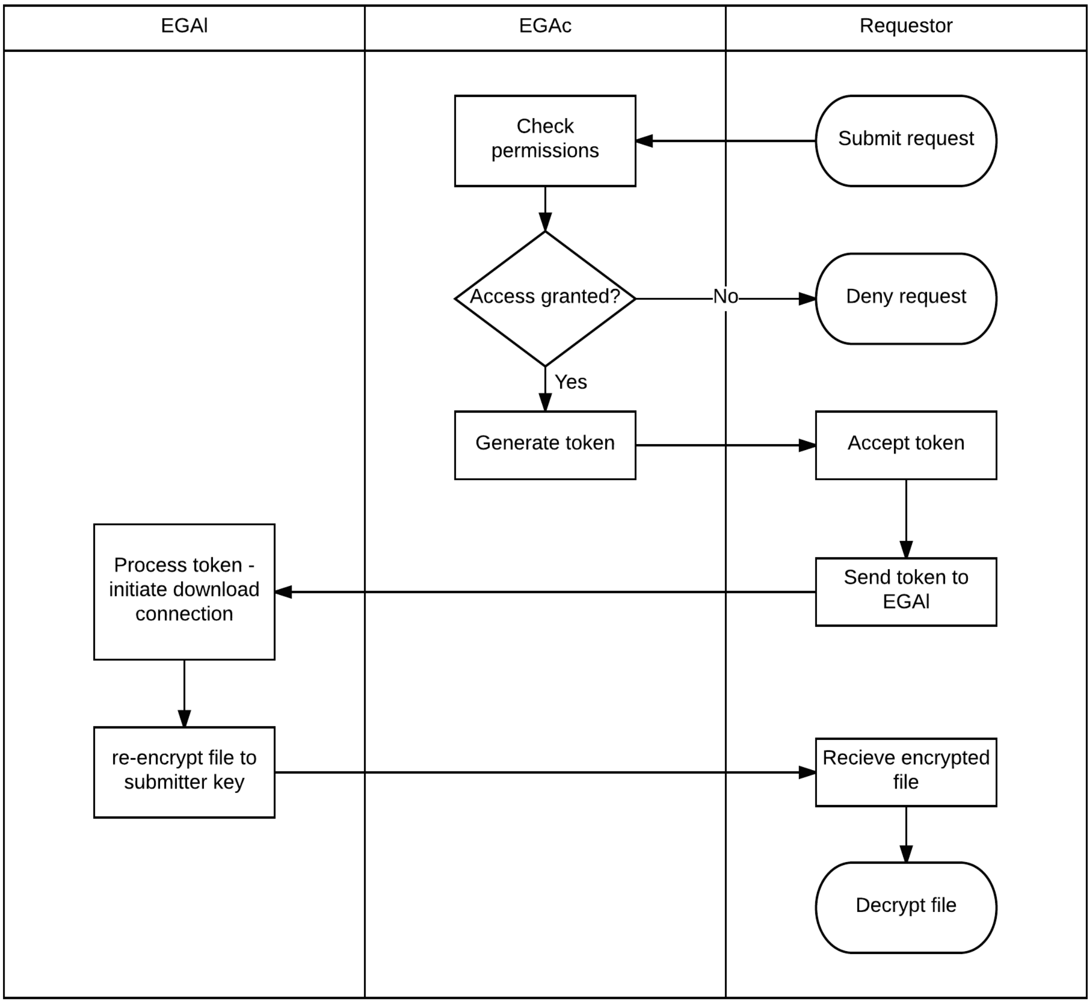

Figure 4 Swimlane diagram representing the process of requesting and downloading a file from EGA.

The representation starts with the download request and excludes the data/file discovery process. The request is processed at EGA core (EGAc), which holds the global permissions for the EGA datasets and users. Once a user has been granted access to the dataset containing the file(s) using the Permissions API (3.1 Permissions API), a token is generated which can be passed to the Local EGA (EGAl) allowing the user to initiate the download process. EGAc also logs the downloads, both successful and failed, allowing EGAc to monitor performance and identify issues early. Note that the file(s) need only be identified by accession number at EGAc, filenames and controlled file data can be held locally at EGAl.

While a majority of the technology used in the Local EGA demonstrator is expected to remain in the production version, for demonstration and speed of development purposes some technology was used which will not enter production. In the production implementation we will continue to use PostgreSQL as the metadata and tracking database as it is open-source, well supported, and has many foreign data connectors to other database technologies. Java will be the main programming language and is used for re-encryption service (RES) and downloader. These services are already in production and have been designed using the micro-service principles which the Local EGA will be developed from. This allows these services to be easily integrated within another system developed using micro-services.

Where the demonstrator differs from the expected production version is in the use of a small Perl script to run the pipeline to interface between command line scripts and the database to demonstrate the pipeline processes. For the production version this functionality is expected to be developed using micro-services and / or a combination of micro-services and pgSQL stored procedures for internal database processes. While the Local EGA demonstrator is fully functional, connections to the EGAc are yet to be developed and will be delivered using Java and the micro-service philosophy. This allows common modules and frameworks to be leveraged, while at the same time allowing new features or extended capabilities to be added as simply as possible.

Further to these requirements, the Local EGA has additional requirements such as both site level and network level monitoring of EGAls to ensure the whole federated EGA system is in a consistent state. Additional requirements comparable to those described for the Cloud (3.7 WP9 Cloud Use) are required if EGAls are allowed to distributed data for highly accessed and correctly permissioned datasets:

- **Solution to transfer data from a Local EGA to the end user via Globus and/or GridFTP**. This requires the end user to have a Globus/GridFTP end-point. It is impractical for the data to be routed via EGAc due to network bandwidth and licensing constraints. This is a requirement of the Data Transfer API and will be supported by WP4, who are in the process of implementing this.
- **Implementation of versioning at a minimum of the dataset level**. To use WP4 data transfer resources, EGA must map the concept of an EGA dataset to the object understood by WP4 as a dataset, which is broader than the definition as used by EGA as it functions as a way to track changes to files and metadata across a distributed system. EGA is collaborating with WP4 to ensure this mapping can be done, or the definition of either datasets can be adapted to ensure this functionality can be implemented.

5.7. Requirements for WP9 Cloud Use

The Cloud subsystem has two requirements closely matching the subtasks in 9.3[3], which deal with data mirroring and EGA data access integration within a cloud environment. In this case, a trusted ELIXIR Node Cloud provider would be able to provide storage and compute resources on sensitive data stored in EGA for their users. The workflow focuses on the technical solutions required to enforce the EGA data access permissions within the virtualized computing environment (access authorization). The DAC overseeing the data stored in the EGA still grants access to the data, for example using the REMS service. The datasets required for the analysis are mirrored from the EGA archive to the Node (Data Mirroring).

Requirements for Cloud Based Data Mirroring

The use-case here is for a user with access to a cloud environment to be able to access EGA data directly from within that environment. This is similar to the use-case where a EGAl instance will distribute the same data as EGAc to relieve bottlenecks in data distribution. Additionally, the requirement to securely transfer the data in an encrypted format which only the target Node can decipher requires that a process almost identical to the 2nd archival step of Local EGA be performed. This fact supports the contention that the Data Transfer API should be identical in both use-cases, as currently the use-case for

[3] Subtasks 9.3.1 Large scale data mirroring support and 9.3.2 EGA data access authorization integration

both requires that the destination is a trusted ELIXIR Node. The outline of the process is given in Figure 5 below.

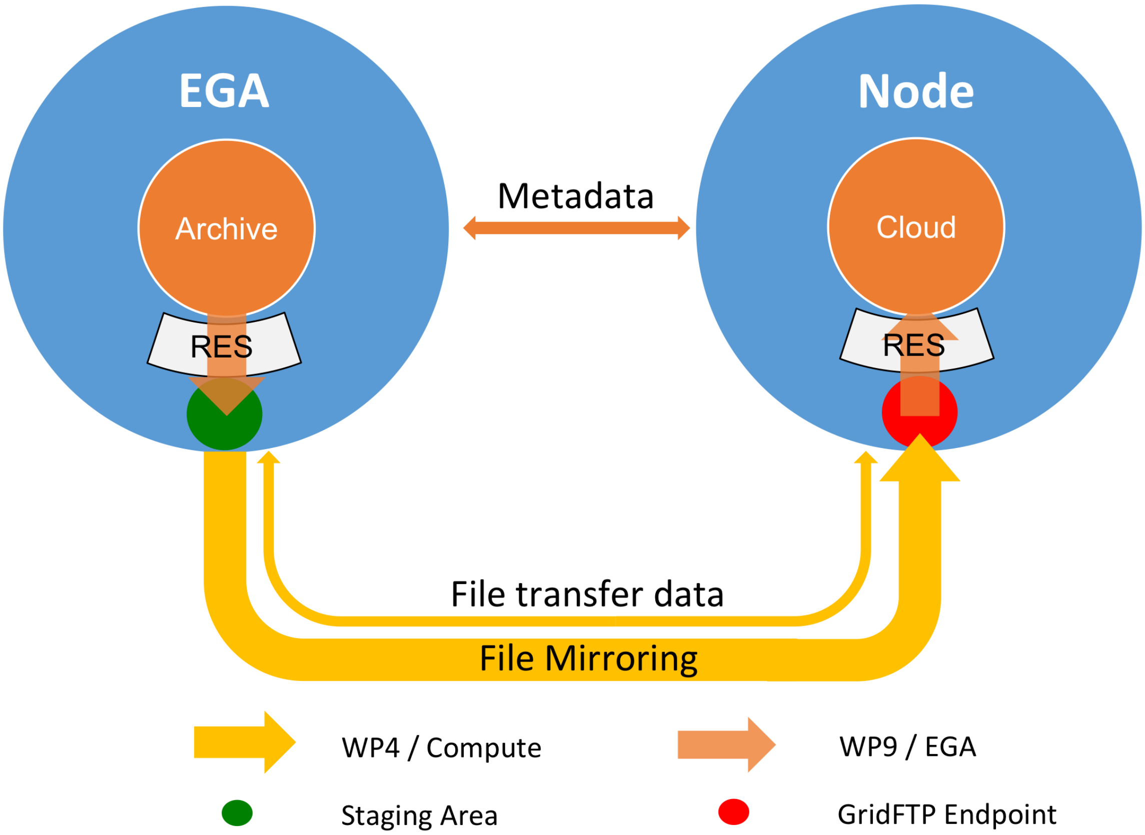

Figure 5 File transfer to a Cloud hosted VM will be unidirectional (yellow arrow) utilizing the data transfer infrastructure being developed by EXCELERATE WP4

File transfer data, such as successful transfer, destination, and transfer rate, is bi-directional. Similarly, the file metadata specific to EGA, such as file size, file type, and encryption key, will be bi-directional (orange).

The file transfer task will be administered by EGA/WP9 due to the requirement for separately transferring public keys, but may in future (as the WP4 infrastructure develops) be integrated with WP4 / Compute administered infrastructure. Encryption and decryption as well as supporting infrastructure are also WP9/EGA tasks. This requires the EGA and the Node to have a staging area and GridFTP / Globus endpoint, respectively.

To simplify the requirements where possible, the file mirroring process should be agnostic to whether a file is controlled access or not, hence the requirement for a staging area to prepare the file for mirroring at EGA and the equivalent endpoint at the Node. It is to / from these locations that the RES (re-encryption service) re-encrypts the file using the correct key and moves the file from / to the source / final location in a manner analogous to the archival step in the Local EGA. Additional requirements for data mirroring are:

- **Feedback the status of the file to EGA**. This includes RES failures in the Node analogous to the archival step monitoring in the Local EGA.
- **Transmission of the required key to / from EGA for the destination Node**. This key may change so an updated file may be transferred with another key.
- **File usage statistics**. To try and ensure correct and secure usage of the data, statistics on data access etc. should be returned to EGA on request.

- **File availability**. EGA needs to be able to withdraw a file from distribution. Similarly, a Node needs to be able to remove a file without permission from EGA in cases where the file usage is specific to the Node, analogous to EGAI withdrawing a file. Consequently, both parties need to know the current status of any given file.

Requirements for Access Authorization

The task is to provide the necessary security without loading technically complicated processes to the end user. For example, a user should be able to use authorized data files without realizing that these files are stored in an encrypted form by the service provider. It is also important to note that the virtualized environment is provided as an IaaS service. Therefore, the WP9 security requirements dictate that authentication, authorization and data access services should be deployed on hardware that is not part of the virtualized environment provided to the research project/data user and controlled by the service provider.

Similarly to the Local EGA demonstrator, a set of 3 Cloud access demonstrators are being developed to demonstrate the authorization and show proof-of-concept at CSC using their ePouta[6] cloud service (a secure Cloud instance). It is expected that the processes and procedures developed here will be able to be extended and applied to other cloud instances. The step-by-step use case workflow and proposed technical demonstrators are described below and summarized in Figure 6. The key points in the workflow are (1) use of ELIXIR AAI certificate to validate the user identity and therefore data access permissions, and (2) providing access to the data through mount points. The mount point should not be considered as a typical NFS mounted directory or Linux soft link to a particular file. Ideally, these mount points appear as normal files to the user but are in fact links that initiate a process that allow access to the data stream from the secure data storage area. Users should never have direct access to this secure storage area and the mount point to a file should only appear to authorized users.

The workflow expects that any given user belongs to user categories EGAd, DU1 or 2, or DP1 or 2, and that:

- The user has an ELIXIR identity which is used for logging into the ePouta[6] cloud service at CSC
- The user is part of a project with active resource (computing and storage) allocation from ePouta cloud service
- The project has already defined which data sets should be made available to the project members for data analysis
- The data have been mirrored to CSC (from EGA using the data transfer API described above).
- The user has been granted access to the datasets by the responsible DAC (otherwise the project requested data files are not visible to the user through the VM).

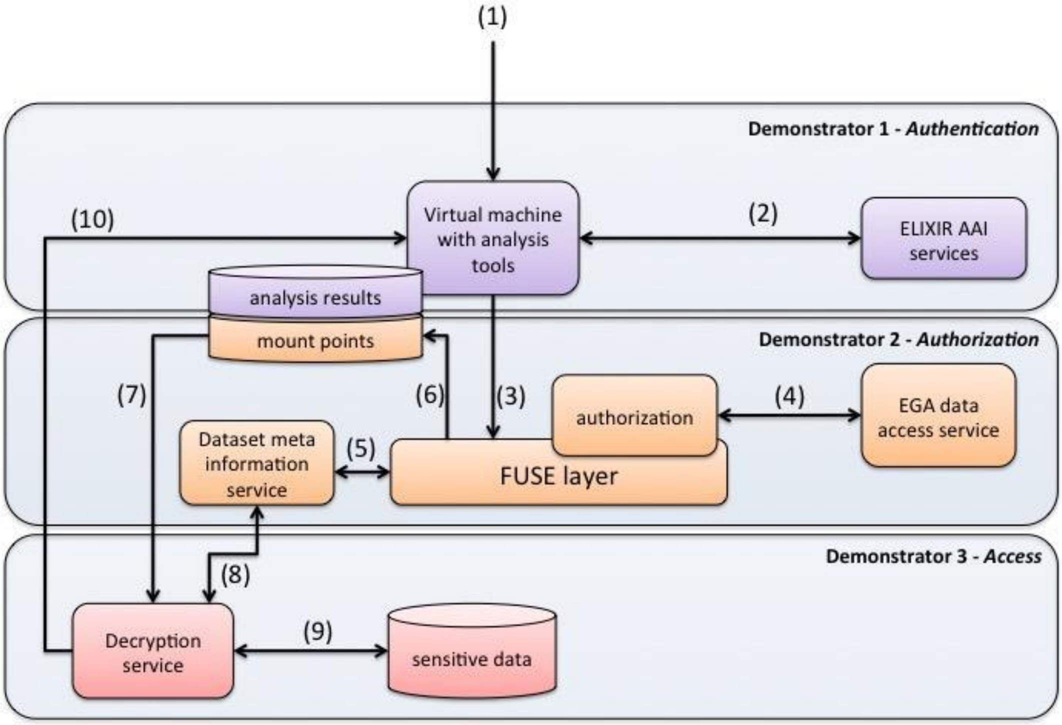
(1)
Demonstrator 1 - *Authentication*
(10)
Virtual machine with analysis tools
(2)
ELIXIR AAI services
analysis results
mount points
Demonstrator 2 - *Authorization*
(7)
(6)
(3)
authorization
(4)
EGA data access service
Dataset meta information service
(5)
FUSE layer
Demonstrator 3 - *Access*
(8)
Decryption service
(9)
sensitive data

Figure 6 Workflow for an ELIXIR user that has access approval to datasets that are made available with necessary cloud computing and storage resources from the CSC ePouta Cloud service,

The numbers on arrows refer to the workflow described as part of the use case. The data access is provided by creating mount points to EGA authorized data files dynamically for each VM user. These mount points are written into the storage linked to the VM. The user is validated using ELIXIR CILogon certificate. This figure does not yet show where these certificates should be stored.

Cloud demonstrator workflow

User authenticates to a VM on CSC ePouta using a ssh client.

The authentication uses the ELIXIR AAI services. The virtual machine connects to the FUSE[4] layer deployed on a separate server. The channel between VM and the FUSE layer is secured by the service provider. VM asks what are the dataset and file mount points that user should be able to access through the VM.

The FUSE layer queries the EGA service permissions API and requests data set permissions. The EGA returns a list of accession numbers for each authorized data set and the associated files. This information is cached for an agreed period of time.

The FUSE layer determines the mount point for the VM assuming the files have already been mirrored to CSC from the Dataset meta information service which maps the CSC locations to EGA dataset and file IDs. If the data are not yet at CSC this will be notified to

[4] Filesystem in USErspace: allows a user-defined file-system to be used. This in turn allows the user to define an encryption key so all files accessed by the FUSE layer can be encrypted and allow direct access by application via the FUSE layer.

the FUSE layer - for example data file is not accessible, being transferred, requested etc (File Transfer Data from Figure 5).

The FUSE layer returns mount point information to a storage linked to the VM.

When user attempts to list or access file(s) using e.g. an analysis tool - VM contacts Decryption service and with the ELIXIR certificate and the relevant mount point information.

Decryption service again contacts the Dataset meta information service to translate the mount point to a valid URI of the file and start to decrypt data.

Decryption service uses the URI to access the data file stored securely at the CSC. The file is decrypted using CSC private key known by the Decryption service.

Decryption service passes a decrypted stream directly to the process at the VM that requested the data.

Demonstrators

Once complete, the demonstrators aim to provide a clearly defined action of applying EGA access permissions within a distributed cloud environment. Each demonstrator is either led by WP4 or WP9. The workflow described in Figure 6 is performed by three separate demonstrators with the following requirements:

Demonstrator 1 – Authentication

- To provide access to ePouta VM(s) for project members by outsourcing authentication to the ELIXIR AAI.
- Use the ELIXIR AAI service to retrieve certificates depending on authorization. This certificate must be stored securely at the CSC.
- Work is led by WP4 (Cloud, AAI) and requires WP9 LoA requirements or clear priority for step-up authentication. CSC should be involved in defining security requirements on certificate management.

Demonstrator 2 – Authorization

- EGA data access permissions must be enforced for each VM user by using an ELIXIR certificate and EGA services. Note that allocated project resources may be shared by a number of project members, each with a different set of EGA data access permissions. Further, some users will have root access to the VM(s), which affects the security of the data accessible by the VM (See 2.3).
- Each user should only be able to access the EGA data sets they have been authorized for using the project allocated VM(s) (In Figure 6 these are shown as mount points in a storage linked to the VMs).
- Root should not be able to access AAI certificates or access directly any of the data files mounted to the VM(s).
- New EGA services supporting trusted third party service providers to query data access permissions should be provided as part of this demonstrator (“EGA data access service” in Figure 6).
- New service that supports mapping between EGA Dataset and File stable identifiers and CSC physical file location (“Dataset meta information service” in Figure 1). This service also provides the mount point format that users can access through the VM.
- The Work is lead by WP9 and forms the core part of the WP9.3.2 task.

Demonstrator 3 – Access

- Provide access to the data files mirrored to the CSC (from the central EGA archive) or made available from ELIXIR.FI resources.
- Use of existing EGA mini services for decrypting data before the data stream from CSC secure data storage area ("Sensitive data" in Figure 1) is passed to the VM. This protocol must be general enough to support also Amazon S3 etc.
- Work is led by WP4 (storage/data mirroring team) and includes members from the WP9.3.1 team and technical personnel maintaining the CSC ePouta and storage services.

Additional requirements need to be determined for the proposed workflow, such as

- What is the appropriate level of security that should be applied on ELIXIR certificate management?
- Can mount point information (Figure 1) with an ELIXIR certificate be used to provide access to the sensitive data. For example, should "Decryption service" (Figure 6) validate authorization to each data file (from "EGA data access service using x.509 certificate) before decryption process is initiated and the stream is provided to the VM?

6. Risks and Mitigation

Availability

Due to the distributed nature of the system, the risk of complete loss of service is small. EGAc is the only single point of failure for the complete system as EGAc holds a catalogue of all files, users, datasets and their permissions, and without response from EGAc the system will fail to function. This is mitigated by EGAc itself being distributed between EMBL-EBI in the UK and CRG in Spain, adding resilience to EGAc.

Dependencies

For effective data mirroring using the ELIXIR WP4 infrastructure, the concept of a dataset at EGA must be mapped to the concept of a dataset within WP4. These two concepts are distinct, as their use cases are different. A dataset at EGA is the atomic object to which permissions are approved or revoked. WP4 understand a dataset to be a collection of files which can be uniquely identified as existing at a particular point in time. WP4 also understand a file as an atomic object which cannot change. Hence EGA must implement versioning to metadata object - but not files to allow the correct mapping of an EGA dataset to a WP4 dataset. EGA are working with WP4 to try and ensure the definition of a dataset from WP4 allows the mapping of an EGA dataset to a WP4 dataset. If this is not achievable we will investigate with WP4 how to make changes to deliver a functional understanding of a dataset.

7. Conclusion and Future Work

In the first year WP9 has:

- Performed extensive requirements analysis for the Local EGA including:
  - Requirements gathering
  - Implementing and distributing a Local EGA demonstrator
- Developed concrete use cases for the ELIXIR AAI through the integration of REMS to EGA
- Started testing the first iteration of the Permissions API to allow integration of permissions application via REMS

- Worked with WP4 to develop the requirements WP9 has of WP4, and the responsibilities each WP has to the other
- Presented at, hosted, and attended meetings (Appendix 3. Related meetings) to obtain as diverse a range of use-cases as possible

Requirements analysis is ongoing with respect to the portals, with further UX testing required. This is currently being done with UX experts at EBI and CRG and members of the user community.

A workshop is being scheduled in autumn 2016 as a hackathon for developers to learn the current code and start to develop it towards the requirements outlined in this document for Local EGA using a micro-services architecture.

Another collaborative workshop between WP4 and WP9 is being planned for autumn in Hinxton to define the boundaries of responsibility between WP4 and WP9 for the Cloud task (data transfer and authenticated cloud access) along with agreed timelines and dependencies. Additionally, EGA is sending a developer to the Node in Finland to expedite this process and increase co-operation between work packages.

8. References

[1] http://www.ctmm-trait.nl/

[2] https://www.elixir-europe.org/documents/elixir-webinar-update-elixir-trait-pilot

[3] https://www.elixir-europe.org/documents/elixir-webinar-local-ega-demo-setup-june-2016

[4] https://github.com/elixir-europe/human-data-local-ega

[5] https://rems.elixir-finland.org/

[6] https://research.csc.fi/epoutahttps://research.csc.fi/epouta

[7] Wilkinson MD, Dumontier M, Aalbersberg IJJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data. 2016;3:160018;http://dx.doi.org/10.1038/sdata.2016.18

[8] The ELIXIR Compute Platform: A Technical Services Roadmap for supporting Life Science Research in Europe; http://dx.doi.org/10.5281/zenodo.60291http://dx.doi.org/10.5281/zenodo.60291

[9] ELIXIR Authentication and Authorization Infrastructure; https://docs.google.com/document/d/1cJ3mR8lqfZKRMvSFalSmPbqd1OPU-L6YcUFIRnh1rhQ/edit

[10] ELIXIR Ethics Policy: https://drive.google.com/open?id=0B7btK9HAXhx1X0NjUEtGOXdxLUE

Appendix 1. Definitions and Acronyms

Acronym	Description
DAC	Data Access Committee: One or more individuals who grant, revoke, or deny access by users to data under their control
Data Providers	Individuals or groups who submit data to EGA. To do so the data must come under control of a Data Access Committee (DAC)
EGA	European Genome-phenome Archive. Can also refer to core EGA and a set of Local EGA instances (EGAI)
EGAI	A local instance of EGA hosted on a Node
EGAc	The core EGA as currently exists as a joint service supplied by EBI / CRG.
AAI	Authentication Authorization Infrastructure
ELSI	Ethical, Legal, and Social Implications
REMS	Resource Entitlement Management System
GA4GH	Global Alliance for Genomics and Health
Vault	Highly secure firewalled file processing and storage area
Globus	High throughput data transfer technology based on GridFTP
endpoint	Logical address mapping to a GridFTP server
LoA	Level of Assurance
FUSE	Filesystem in USErspace
FAIR	Findable, Accessible, Interoperable, Re-useable

Appendix 2. EGA Data Model

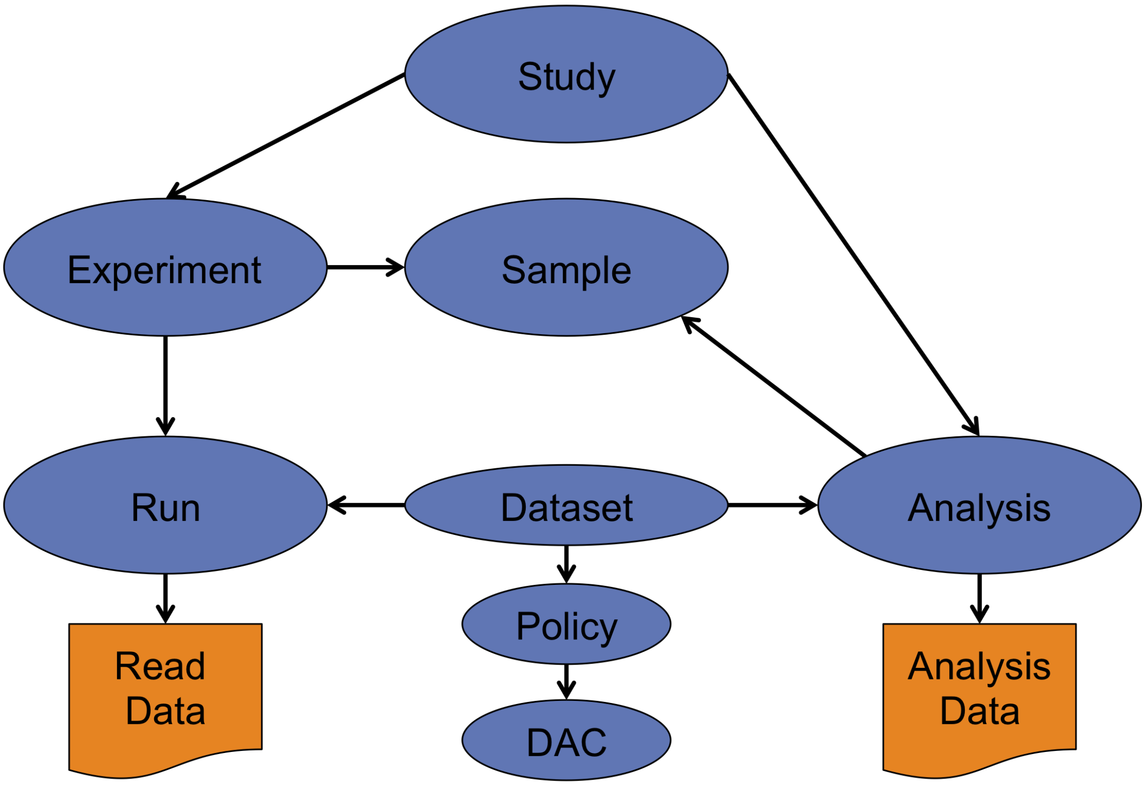

Figure 7. The EGA data model. A sequencing submission to EGA consists of a study linked to one or more Experiments and/or Analyses. Each Experiment or Analysis must reference a Sample. A Run must reference an Experiment, and both Runs and Analyses are linked to the associated file(s). Runs and Analyses are also associated to one or more Datasets, which determines the access permissions to their associated files.

Appendix 3. Training Materials

Local EGA demonstrator webinar:
https://www.elixir-europe.org/documents/elixir-webinar-local-ega-demo-setup-june-2016