Title:	*User Requirements Document*
Author:	*Work Package 2 ("Requirements Capture")*
Editor:	*Árpád Andics (MTA SZTAKI)*
Reviewers:	*John Ibbotson (IBM), László Zs. Varga (MTA SZTAKI), Steven Willmott (UPC), Frank Dannemann (DLR), Andreas Screiber (DLR), Luc Moreau (SOTON), Victor Tan (SOTON)*
Identifier:	*D2.1.1*
Type:	*Deliverable*
Version:	*1.0*
Date:	*Thursday, March 10, 2005*
Status:	*Public*

Summary

This document identifies the user requirements for the provenance architecture that is to be developed within the Provenance project. They have been obtained from the project partners as well as from external parties who filled in the User Requirements Survey (Work Package 2, Task 1). This document will form the basis for the Software Requirements Document (Work Package 2, Deliverable D2.1.2).

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme

Members of the PROVENANCE consortium:

IBM United Kingdom Limited	United Kingdom
University of Southampton	United Kingdom
University of Wales, Cardiff	United Kingdom
Deutsches Zentrum fur Luft- und Raumfahrt s.V.	Germany
Universitat Politecnica de Catalunya	Spain
Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézet	Hungary

Foreword

This document has been compiled by Árpád Andics (MTA SZTAKI) based on existing project documentation, the results of the User Requirements Survey (Work Package 2, Task 1), the results of additional discussions with project partners and the recommendations of the ESA Software Engineering Standard PSS-05-02

Table of Contents

Table of illustrations

Figures:

Tables:

List of acronyms

- DMG – DataMiningGrid (demo application scenario)
- ESA – European Space Agency
- HCI – Human-computer interface
- HLSF – Healthcare and Life Sciences Framework (demo application scenario)
- IST – Information Society Technologies
- OTM – Organ Transplant Management (demo application scenario)
- TMA – Traffic Management Application (demo application scenario)
- URS – User Requirements Survey

List of definitions

- *Actor:* An individual or an organisation that is involved in a data manipulation process.
- The *provenance of a piece of data* is the documentation of the process that produced that data.
- *Workflow:* The process by which a series of tasks are executed in a specific sequence; including the specification of how outputs of tasks are routed to the inputs of other tasks, where such action is required.
- *Workflow enactment engine:* A software program that conducts the execution of a workflow in accordance with the specification of the workflow. In distributed computational environments the workflow enactment engine is usually a service that makes use of and coordinates other services in order to execute a given workflow submitted to the engine by a client.

References

1 Applicable documents

- "ESA Software Engineering Standards" by C. Mazza, J. Fairclough, B. Melton, D. de Pablo, A. Scheffer, R. Stevens. Published by Prentice Hall 1994.

2 Reference documents

- [FKT01] Ian Foster, Carl Kesselman, and Steve Tuecke. The Anatomy of the Grid. Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications, 2001.
- [FKNT02] Ian Foster, Carl Kesselman, Jeffrey M. Nick, and Steven Tuecke. The Physiology of the Grid – An Open Grid Services Architecture for Distributed Systems Integration. Technical report, Argonne National Laboratory, 2002.
- [PASOA] Miles, S., Groth, P., Branco, M. and Moreau, L. The requirements of recording and using provenance in e-Science experiments. Technical Report, Electronics and Computer Science, University of Southampton, 2005. http://eprints.ecs.soton.ac.uk/10269/

1 Introduction

1.1 Purpose of the document

The purpose of this document is to identify and document the user requirements for the provenance architecture that is to be developed within the Provenance project. The primary source of the information presented in this document is the User Requirements Survey that was prepared as a first action in Work Package 2. This survey has been completed by project partners as well as external parties through the Web. This document has been compiled based on the evaluation of the completed surveys.

This document is based on the ESA Software Engineering Standard PSS-05-02. The document constitutes the problem definition phase of the life cycle of the Provenance project. The analysis of the ability to meet the individual requirements as well as the transformation of the abstract level user requirements into technical level requirements are the tasks of the software requirements definition phase. The Software Requirements Document, which forms the basis of the software development and testing, is to be produced by the evaluation of this document.

The document is addressed to all project partners involved in the design, implementation, testing and deployment of the provenance architecture.

1.2 Scope of the software

Provenance enables users to trace how a particular result has been arrived at by identifying the individual services and the aggregation of services that produced the result. The overarching aim of the Provenance project is to design, conceive and implement an industrial-strength open provenance architecture for grid systems, and to deploy and evaluate it in complex grid applications, namely aerospace engineering and organ transplant management.

1.3 Overview of the document

This document contains the functional requirements of the users who may use the provenance architecture being developed within this project. Users include the project partners with the demo applications (UPC, DLR), external parties whom we contacted directly (e.g. eDiamond, myGrid) and external parties who expressed interest in the topic by filling in the online questionnaire. The document contains also the constraints these users identified on a provenance architecture within the context of their systems.

The structure of this document is as follows:

- Chapter 2 gives an introduction to the topic of provenance and provides a brief overview of the Provenance project. This is followed by a brief description of the application scenarios including the demo applications as well as other potential use cases. The general capabilities and constraints, the user characteristics and the operational environment of the systems are described in the last part of this chapter.
- Chapter 3 describes the specific requirements placed on the provenance architecture by the users we have surveyed. This involves functional as well as constraint requirements. Functional requirements are divided into two groups, namely abstract level and technical level requirements. Technical level requirements are discussed following the structure of the User Requirements Survey.
- References to the collected surveys and additional scenario documents are listed in Appendix A.

- A tabular summary of the user requirements is presented in Appendix B.

Requirements presented in this document are classified in three categories:

- Abstract level requirements
- Technical level requirements and
- Constraint requirements.

Requirements are assigned the following priorities:

- *Essential:* A requirement is marked as 'essential' if any of the demo applications requires it. These requirements have high priority.
- *Desirable:* A requirement is marked as 'desirable' if it originates from a use case other than the demo applications and is considered important for the given use case(s). These requirements have normal priority.
- *Nice to have:* Requirements that are stated to be optional by the users are marked as 'nice to have'. The origin of these requirements (i.e. whether it comes from the demo applications or not) makes no difference in this case. These requirements have low priority.
- *Critical:* Requirements stated to be critical by the users are marked with this flag. It should be considered for the highest priority for the requirement.

Each requirement is flagged with one of 'essential', 'desirable' or 'nice to have' according to the above rules. The 'critical' flag is an extra flag that might be assigned to a requirement.

Requirements are label by the following pattern:

REQUIREMENT_CLASS - X - Y [- A [- Z]]

where:

- REQUIREMENT_CLASS = "AR" | "TR" | "CR" meaning abstract level, technical level and constraint requirement respectively;
- X is a number that corresponds to different sections in this document;
- Y is the ordinal number of a requirement within a section;
- A is an optional letter to distinguish different requirements imposed on the same aspect of the provenance architecture; and
- Z is an optional number to identify individual requirements within a group of requirements.

2 General Description

2.1 Product perspective

In the "Anatomy of the Grid", Foster, Kesselman and Tuecke describe the problem underlying the Grid concept as *coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organisations* [FKT01]. As part of the endeavour to define the Grid, a service-oriented approach has been adopted so as to facilitate the composition of services into more sophisticated services [FKNT02]. While the underpinning mechanisms for creating and managing such virtual organisations still remain to be understood, effort is required to allow users to place their trust in the data produced by such organisations. Understanding how a given service is likely to modify data flowing into it, and how this data has been generated, is crucial as illustrated by the following generic question:

Let us consider a set of services that belong to an open grid environment and that decide to form a virtual organisation with the aim of producing a given result; how can we determine the process that generated the result, especially after the virtual organisation has been disbanded?

Provenance is therefore important in enabling a user to trace how a particular result has been arrived at, and the sequence of steps that were involved. Specifically, we consider the specific notion of *execution provenance*, which identifies what data is passed between services, what services are available, and how results are eventually generated for particular sets of input values. Using execution provenance, a user can trace the "process" that led to the aggregation of services producing a particular output. We see provenance support as a crucial building block of a grid infrastructure that is required for users to *trust such a new paradigm*.

As [PASOA] points out there is no existing technology at the moment that provides a principled, application-independent way of recording, storing and using provenance data. The overarching aim of the Provenance project is to design, conceive and implement an industrial-strength open provenance architecture for grid systems, and to deploy and evaluate it in complex grid applications, namely aerospace engineering and organ transplant management. Industrial-strength provenance support includes a scalable and secure architecture, an open proposal for standardising the protocols and data structures, a set of tools for configuring and using the provenance architecture, an open source reference implementation, and a deployment and validation in industrial context. For the description of the planned and potential application scenarios see section 2.2 ('Application scenarios').

2.2 Application scenarios

According to the project plan, the reference implementation of the provenance architecture is to be deployed in two demonstration applications. For this reason the requirements of these applications have a high priority. The description of these application scenarios can be found in section 2.2.1 ('Demo applications').

The Provenance project aims to develop a general provenance architecture that is applicable in a wide range of use cases (beside the demo applications). In order to achieve this, requirements have been gathered from several external parties through direct contact as well as a publicly open online version of the User Requirements Survey. The description of these further use cases can be found in section 2.2.2 ('Further application scenarios explored by the User Requirements Survey').

The collected questionnaires and additional scenario documents are made available on the website of the project. For links to the individual documents refer to Appendix A.

2.2.1 Demo applications

2.2.1.1 Aerospace Engineering (the TENT system)

2.2.1.1.1 Scenario overview

The SISTEC group at DLR is involved in developing workflow based approaches for combining software subsystems and components that provide scientific simulation, data pre/post-processing and visualisation functions. Each of these involves complex software packages, some of which require specialised hardware resources to execute. Some of these packages are developed in-house, but others are obtained from a number of different vendors and consortium partners. The workflow must support both static, predefined interactions between components, and in some cases real-time interactions to support "computational steering". The TENT system at DLR is an example of such a system, which utilises distributed object technologies to connect software subsystems.

Provenance is crucially required in this context, as the need to maintain a historical record of outputs from each subsystem is an important requirement for many customers that use the end result of simulations. Associating provenance information with the workflow engine itself is also useful, as information about aircraft structures developed as a consequence of this work needs to be maintained over long time periods. For instance, aircrafts' provenance data need to be kept for up to 99 years when sold to some countries. Currently however little direct support is available for this, and involvement with this project will be useful to associate such provenance information with workflow tools.

Potential other uses of provenance in the TENT system include:

- Failure analysis within the TENT system or it's components (e.g. finding potential leaks in the architecture)
- Enhancement of the system architecture
- Finding bottlenecks in the workflows (e.g. regarding the involved network or component deployment)

2.2.1.1.2 Detailed scenario description

A) Introduction

A wide range of today's engineering applications require the numerical simulation of the underlying physical processes (e.g., fluid mechanics, structural mechanics, thermodynamics, and their coupling). Performing a complex simulation is the travers of a multistage process chain consisting of the preprocessing for the different simulation codes, the simulations themselves and their coupling, and the appropriate postprocessing. Figure is a sketch of a typical application scenario for a coupled multidisciplinary simulation.

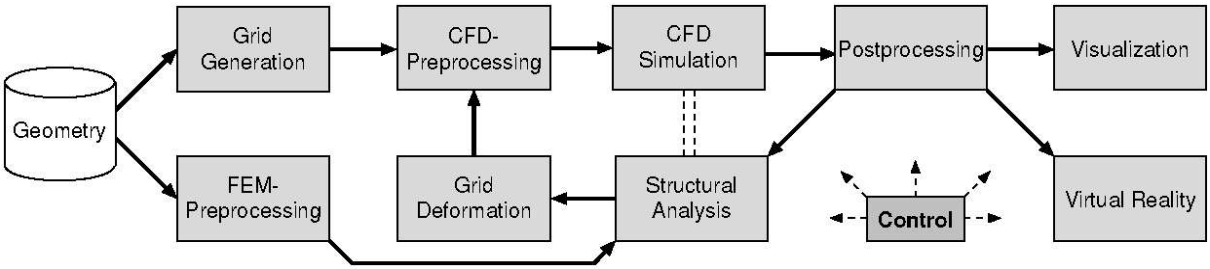

Figure 1: Typical application scenario for coupled multidisciplinary simulation

Often the performance requirements for such a complex simulation can be met only by exploiting distributed computing resources. The complexity for handling the interactions between the different elements of the simulation is enlarged by the effort for managing distributed resources. This increases the time for performing simulations and forms an upper bound for the complexity of a simulation being manageable.

In order to improve the building and managing of process chains for complex simulations, the distributed integration environment TENT has been developed, taken into account the following design goals:

- Application of component technology in distributed computing.
- Composition of process chains in a "drag and drop" manner.
- Flexible configurability of process chains.
- Easy management of user projects.
- Easy access to interactive simulations from any computer in the net.
- Simple integrability of existing applications and their supporting tools.
- Efficient data exchange between the stages in the process chain.

B) System Architecture

TENT has been realized as a CORBA based component architecture. CORBA (Common Object Request Broker Architecture) is an object-oriented communication middleware enabling the construction of distributed systems from executables running independently on different machines. With CORBA, each executable presents itself to other executables as an object, having an interface consisting of methods and parameters which are directly accessible by other objects, independent of the particular implementation language. Interfaces are described in a language and platform independent manner using the Interface Definition Language (IDL). At runtime, method invocation between objects is mediated by the Object Request Broker (ORB) performing all necessary networking and inter process communication. However, CORBA only serves as the basis for thecomponent architecture needed to build our particular integration framework. The figures below give a schematic overview of the described role of CORBA in the TENT framework.

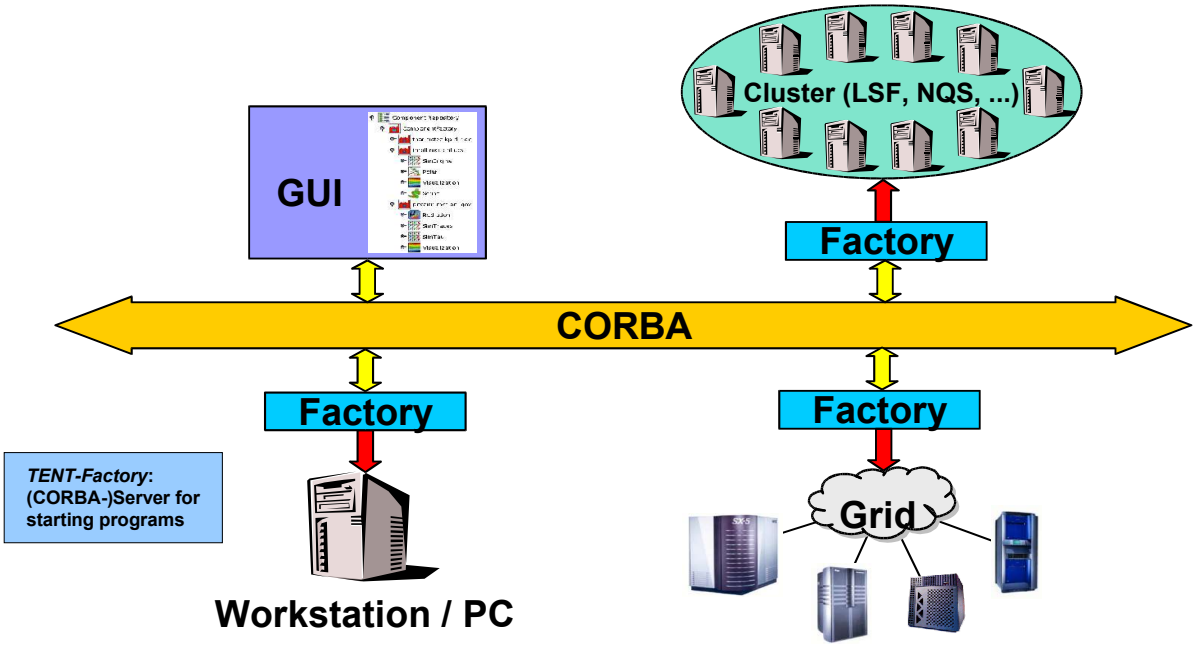

Figure 2: Role of CORBA in the TENT integration framework, illustration 1

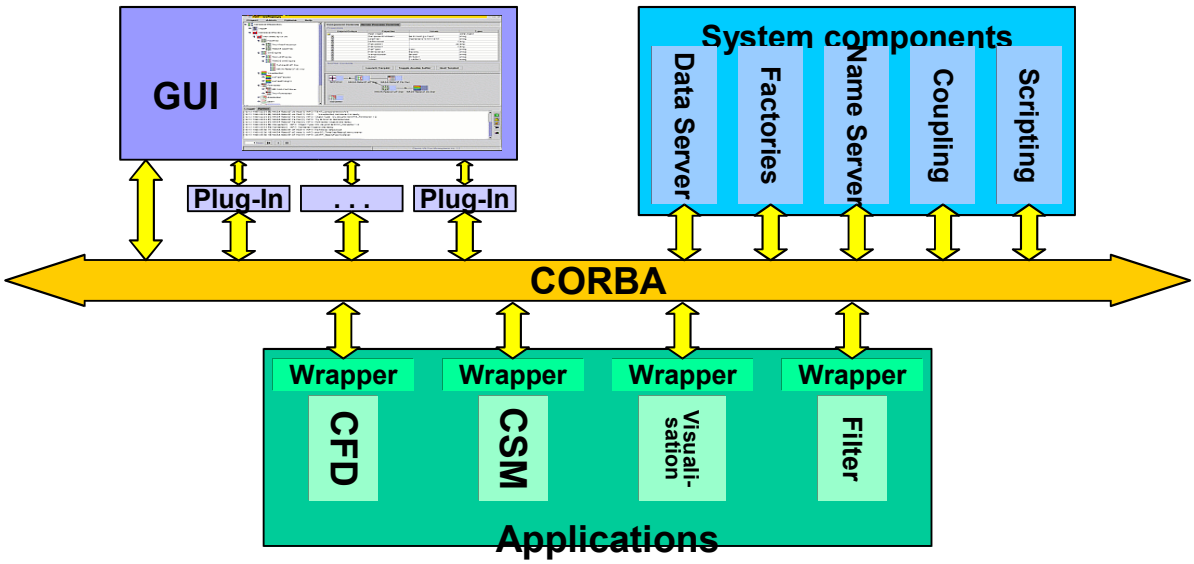

Figure 3: Role of CORBA in the TENT integration framework, illustration 2

C) Primary Actors and Roles

One can distinguish between three types of roles interacting with the TENT system:

1. *User:* A TENT user is the person who is actually executing a simulation inside the integration system. In a typical aerospace workflow this will be the engineer interested in the results of the simulation. From the users point of view the system configuration as well as the parameters which can be adapted are more or less given in terms in predefinition or restricted choices.
2. *System Designer:* The function of a system designer is to provide users with predefined system configurations. A system designer has access and can configure all parts of the TENT system. A typical system designer would be a project manager, who adapts TENT in a way that it can be used by his staff in order to perform simulations. A system designer may also interact with TENT in the role of a user.
3. *System Developer:* System developers will produce and make available new functionalities of the TENT system. Usually this are the programmers of the DLR SISTEC group. A system developer can also take over the role of a system designer or user.

Note: During the runtime of a workflow the system designer as well as system developer take over the role of an ordinary user.

D) Application of the TENT system in the SikMa project

The SikMa project (Simulation of Complex Manoeuvre) is under the direction of the DLR Institute of Aerodynamics and Flow Technology (AS). The objective of the SikMa project is to use TENT as an interactive simulation environment for the simulation of a freely flying, fully configured, elastic fighter aircraft. To implement the simulation, the aerodynamic, flight mechanical and aeroelastic equation systems will be calculated at every step, for a time-accurate coupling of aerodynamics, flight mechanics and aeroelasticity. The different simulation codes will be combined within TENT in a process chain (or workflow) as it is depicted in the figure below.

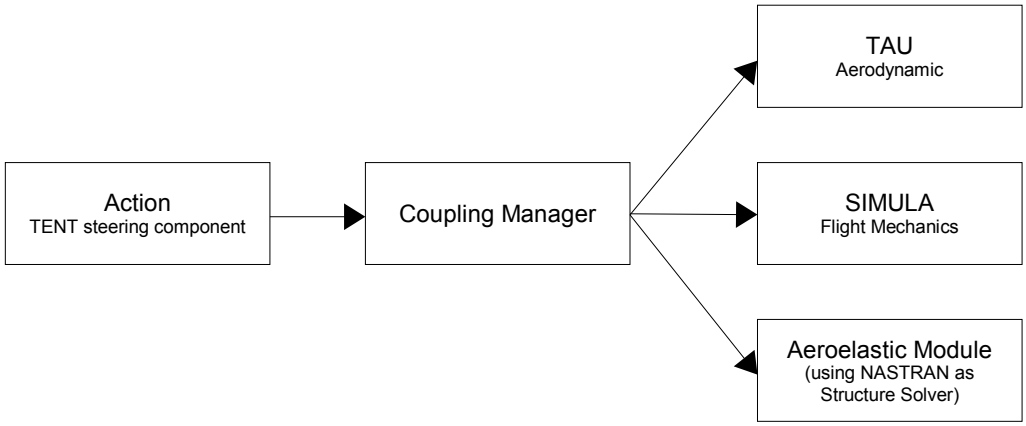

Figure 4: SikMa workflow within TENT

The envisaged transient calculation will use very large meshes as its input. Therefore the simulation makes great demands on the involved computational environment, which must be capable of:

- handling huge amount of data (in GByte range)
- distributed computation of the involved codes
- long computation times (up to two weeks)
- handling of inter-process communication of the simulation codes

Except for the Coupling Manager, which is necessary for steering the coupled scenario, all other components of the workflow expect data files as their input. The output varies from component to component:

- TAU produces several solution files (for example for field and surface), temporary files and a separate file containing all information going to stdout or stderr (like monitoring and error messages)
- SIMULA produces plot data, which is stored as a file at the end of the simulation.
- The Aeroelastic module produces plot, monitoring and restart data, whereof the first two are stored as ASCII files and the last one as a Matlab input file.

For a deeper insight into the TENT architecture as well as to see further details of provenance needs in this application scenario refer to the document entitled "Provenance data in TENT". (For reference see Appendix A.)

2.2.1.2 Organ Transplant Management

2.2.1.2.1 Scenario overview

E-Health is a major application area both for grid technology and provenance solutions. Medical information systems, databases and in particular decision support systems rely on a wide range of data sources, human input and access to patient data. In many cases, practitioners are highly regulated, must retain careful audit data and rely heavily not only on information in the system but knowledge added by doctors, surgeons and other individuals using the systems. Exemples of this are organ and tissue transplant processes that are characterised by the following constrains:

- European, national, regional and site specific rules govern how decisions are made (the application of these rules must be ensured, be auditable and may change over time);
- Patient recovery is highly dependent not only on the organ allocation choice but subsequent extraction and insertion methods as well as the care and recovery regime (while much is understood about certain types of transplants, many elements of post transplant care and the

relationship of organ/tissue acceptance rate to the match made as well as the care applied require much more detailed study);

- Patient records, organ/tissue bank databases and other information are distributed across a number of sites (In Barcelona alone, there are 4 main tissue banks and a significant number of possible transplant centres).

Current organ transplant systems are very far from grid ready (pretty much everything is still done by phone between different sites). But in the long-term, we expect such systems to:

- Link up all the tissue banks, organ recipient lists, emergency centres, etc. held at different hospitals and link the decision support systems which guide the allocation process;
- Connect allocation mechanisms across regional and state boundaries to ensure that all EU, national and regional regulations are rigorously enforced in the process;
- Maximise the efficiency in matching and recovery rate of patients.

This application will benefit from grid technologies because there are a large number of patient record sites, tissue banks and other databases in the region and in general data cannot be sent and cached (due to confidentiality and size). Also computation is very complex. Surgeons should match over about 50 dimensions, e.g. for a cornea, but tend to just use 4 because the reasoning becomes too complex and the effects are not understood.

In this application, the major provenance problems are:

- Tracking back previous decisions in any one centre to identify "whether the best match was made" (verifying/proving this and generating an explanation), who was involved in the decision, what was the context.
- Aggregating partial results from searches in different centres and applying the rules that apply between centres. Maintaining the validity of partial results.

2.2.1.2.2 Detailed scenario description

A) Introduction

Treatment of patients through the transplantation of organs or tissue is one of the most complex medical processes currently carried out. This complexity arises not only from the difficulty of the surgery itself but also from a wide range of associated processes, rules and decision making which accompany any such surgery. Depending on the country where a particular transplant is being carried out procedures and the level of electronic automation of information / decision making may vary significantly. However, it is recognized worldwide that ICT solutions which increase the speed and accuracy of decision making can have a very significant positive impact on patient care outcomes.

Electronic systems that might be implemented for transplant management can be divided into several elements:

1. Medical Record management: the storage, access and modification of medical patient care records for patients in a given geographic region. Gathering, access and modification of such data is regulated by European, national and regional laws and forms an underlying information system for any treatment process management system.
2. Transplantation Management: information systems used by medical staff during the process of a transplant incident (a single patient receiving an organ or tissue) to access existing case or background data, share it with colleagues, carry out matchmaking and/or otherwise provide decision support.
3. Transplant case post processing: long term, post incident data analysis techniques able to extract aggregate information such as general trends over large sets of previous transplant case records.

The following discussion focuses primarily on the second of these elements and touches upon the third. The organ transplant process according to the current practice will be discussed. The presentation covers major actors, workflows and decision criteria.

Before beginning, it is important to note that transplantation operations are divided into two broad classes:

1. Live organ transplants (heart, lung, intestine, liver, pancreas, kidney): In this case, the item being implanted is a live internal organ such as a heart, lung or similar. In general such organs deteriorate rapidly between when they become available and implantation (becoming useless in less than 24hours in some cases1) – creating significant time pressure on transplantation. Furthermore cases normally arise with a waiting list of patients waiting for a suitable organ and a donation being made at a given moment in time meaning that the organ then must be assigned to one of the waiting patients (or none if no good matches are found).
2. Tissue transplants: In this type of transplant the item being transplanted is a tissue such as a cornea, skin, bone or something similar. In general such transplants are carried out by matching the requirements of an incoming recipient with pieces available for transplant from large collections of relevant tissues known as "tissue banks" - making decision making a "1 recipient to one of many possible donors" matching problem. Tissues can be stored for much longer periods of time than organs and such transplants are carried out with far less urgency.

The problems are therefore significantly different in structure and challenges, furthermore the provenance issues may be somewhat different (in the organ case there is concern about whether the right recipient was chosen, in the tissue case concern whether or not the right piece for implantation was used). Now we focus on the organ transplantation case.

B) Primary Actors and Roles

A '*transplant case'* is defined as a single episode of organ transplantation (or attempted organ transplantation) including all processes from the arrival of the donor to the completion of surgery and after care of the recipient. In a given case the major actors or types of actors and their roles are:

- Donor: person donating the organ or organs in a particular case. The individual must be associated with an available medical history (otherwise transplant cannot take place).
- Recipient: person or persons being operated upon (successfully or unsuccessfully) to implant the donated organ. Associated with a particular medical history and a particular possible implantation center.
- Recipient Waiting List: ordered list of individuals who may act as potential recipients if an organ becomes available (grouped by the type of organ they required).
- Retrieval Team: medical personnel (surgeon, nurses, technicians, etc.) carrying out the retrieval of an organ from the donor. Associated with a particular retrieval site.
- Implant Team: medical personnel (surgeon, nurses, technicians, etc.) carrying out the implantation of an organ in the recipient. Associated with a particular implantation site.
- Duty Transplant Surgeon: individual physician/surgeon on duty at the retrieval or implantation center.
- Consultant Transplant Surgeons (experts): individual(s) other than the duty surgeon who may be consulted by the duty surgeon during any given case. Associated with one or more retrieval/implantation centers.
- Remote retrieval site: location where the retrieval takes place if this location is not a hospital or suitably equipped retrieval center.
- Retrieval center: hospital coordinating / carrying out the retrieval of an organ – either at the hospital itself or at a remote retrieval site.

- Implantation center: hospital carrying out the implantation of an organ.
- Post operation care center: hospital or medical center looking after the patient in post-operation care.
- Immunology center: specialist medical center performing blood and other analyzes of organs in order to determine matches in key indicators (HLA analysis and crossmatching). This step is normally skipped in the case of everything except kidney transplants since because of the extreme urgency of the transplants (the analysis of everything except blood type may not improve success rates more than a quick transplant).
- Regional Organ Transplant Authority (OTA): regulatory and oversight body for all transplants in a given region. Associated with a number of retrieval / implantation centers. The OTA center also acts as the coordinating point to find recipients if local recipients are not available.

Figure 5 illustrates the communication paths between these primary actors. In a standard incident the duty transplant physician of the retrieval center is alerted to the availability of a possible donor, this individual sets in motion processes for assessing the donor. Information on which organs may be donated is then passed to local transplant teams (in the same center) and the Organ Transplant Authority (OTA) to find potential donors. Depending on the type of organ, conditions and the protocols for the situation the duty physician and OTA mediate to find an appropriate recipient. Once a recipient has been found a two part transplant team from the potential implant center takes charged – a retrieval team is sent to the location where the donor is and an implant team is readied at the implant center, the leader of the transplant team (comprising both parts) takes charge of the proceedings.

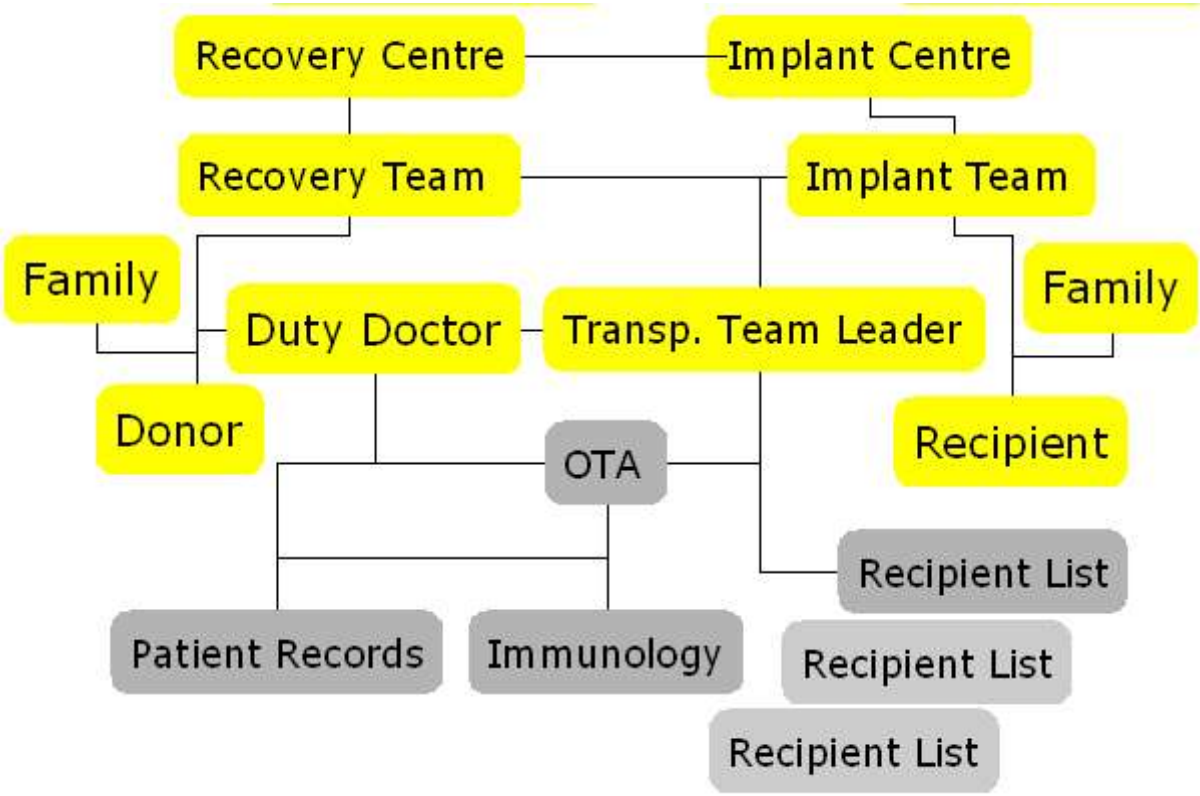

Figure 5: Direct communication during a transplant case (post-operation care center not shown)

The physical distribution of the actors shown in Figure 5 are approximately as follows (see later sections for the number of sites of each type in a typical region):

- Retrieval center and Implant center are both medical centers, each with its own physical location. In certain cases the same medical center may play both roles.
- The duty transplant physician is always located at the retrieval center site.
- If the transplant is from an accident the retrieval team may be at an arbitrary location which is not a medical center (however this case is extremely rare – almost always the donor is moved to the nearest medical center.)
- Patient records are stored at each medical center patients are registered with but can be treated as a single distinct site (data is accessible from all sites)
- The immunology center is a distinct medical center per region – it may or may not be in the same place as the retrieval or implant centers.
- Experts may be at one of the previously mentioned medical centers, at another medical center entirely or in an arbitrary place (reachable by phone). They are generally members of transplant teams currently not on duty but may have additional experience of special situations.
- The organ transplant authority (OTA in the diagram) is located at a single further geographic site an is on 24h call.
- The post-operation care center may be one of the retrieval or implantation centers or another physically located center.

C) Transplant Workflow

Figure 6 illustrates the standard workflow for a generic transplant incident as a labeled directed graph.

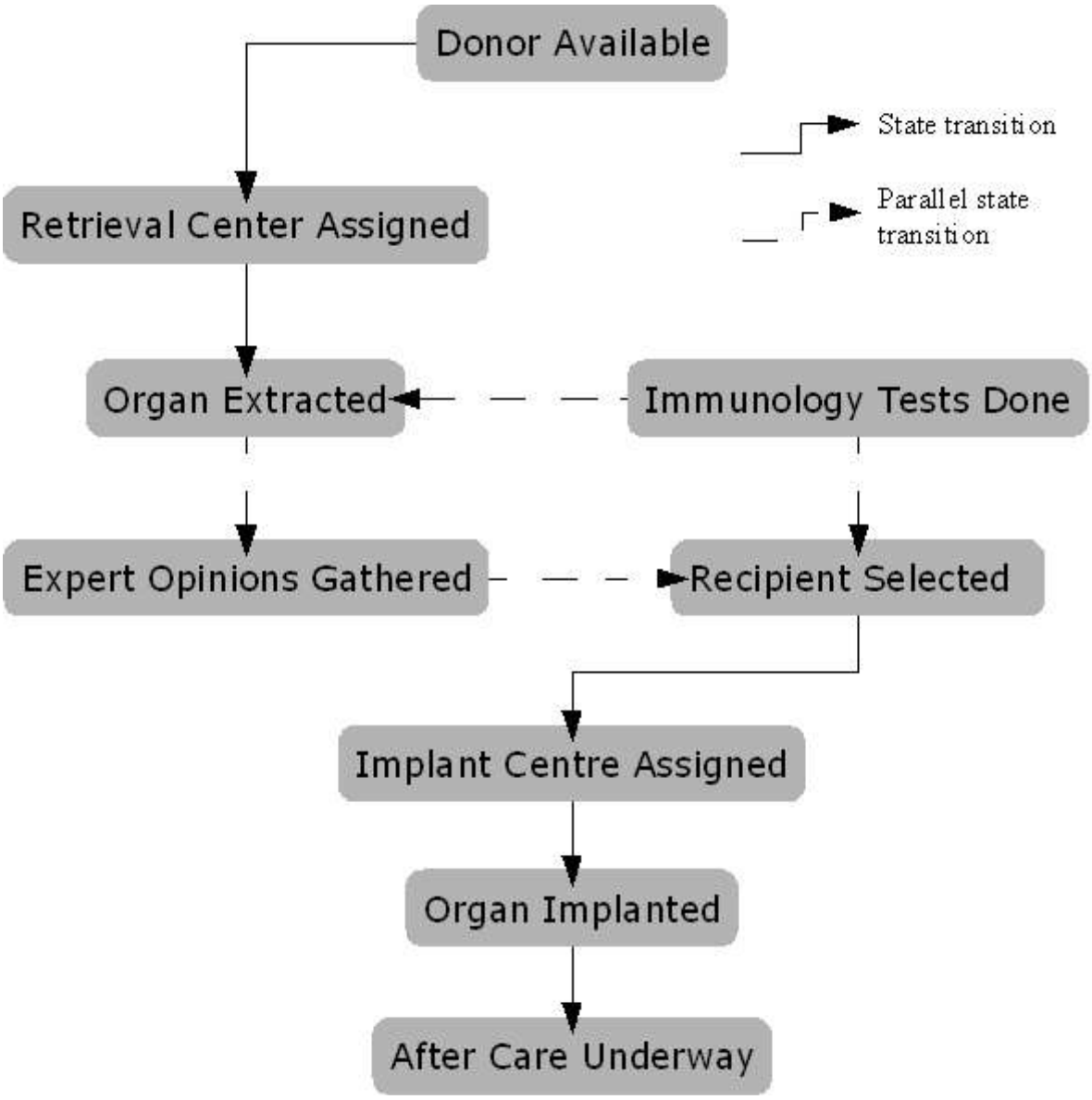

Figure 6: Approximate Transplant workflow. Boxes are states, arrows are transitions between states. Dashed arrows indicate that both paths must be followed to reach the destination state.

D) Transplant Dataflow

Figure 7 illustrates the an approximate data flow graphic for a generic transplant incident as a labeled directed graph.

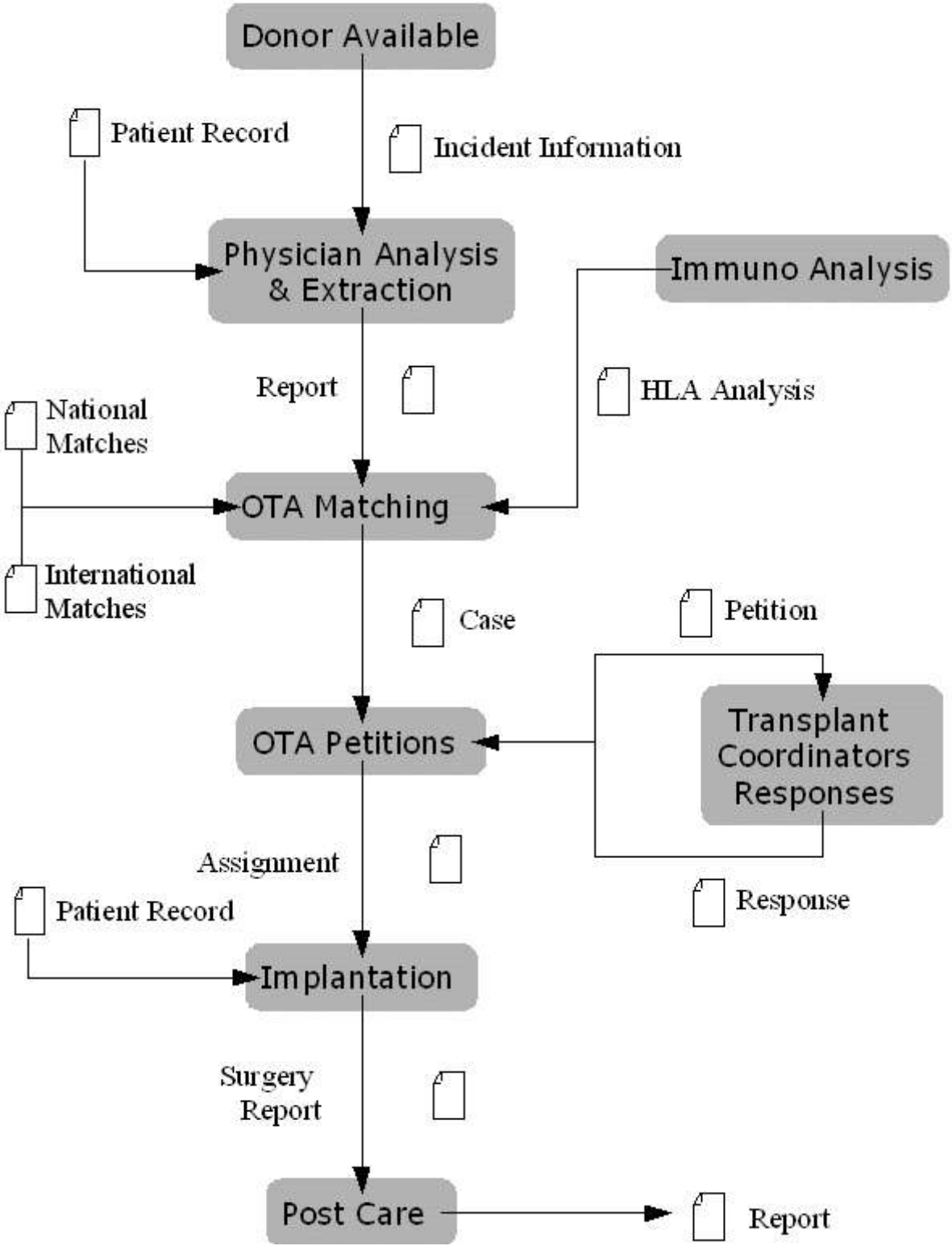

Figure 7: Approximate Transplant dataflow. Grey boxes indicate actions in the world, documents represent data of some kind.

E) Decision Criteria

As can be seen from the descriptions of actors and workflow in the previous sections a transplant case has a clear separation of responsibility for decisions (duty surgeons, external experts etc.) and

generation of information (records and for example analysis by the immunology group). The decision criteria applied to each individual case are complex and involve hard and soft constraints. Hard constraints are those which may never be violated, soft constraints are those which it is preferable not to violate but may be in certain cases and/or have a (qualitative or quantitative) measure of desirability associated with them.

The decision which effectively needs to be taken in a given transplant case is:

> *"Given an available organ **x**, which patient **y** from the set of potential recipients **Y** should be selected as the recipient?"*

Important factors which impact on this decision are:

- Which recipient has the best medical chance of successfully accepting a given organ?
  - How good is the clinical match of **x** to each **y** in **Y**? In terms of major and minor factors such as ABO blood type, age of donor and recipient etc. [full classification to be added]
  - Are there additional compatibility issues? (E.g. The donor as infected with a given virus such as HIV or Hepatitis B/C – in which case recipients also infected with this virus may be able to receive it whereas for those not carrying these viruses implantation would carry a risk of transmission.
  - Are the additional surgical, logistical etc. issues which would worsen / improve the chances for one or other of the potential recipients? (E.g. If one of the recipients is immediately available and another is not or one of the recipients is located a great distance from the donor.)
- Which recipient is in most urgent need of a transplant?
  - Is any of the potential recipients in danger of imminent death if they do not receive a transplant?
  - Which recipient's quality of life stands to improved by the greatest margin by a given organ? For example, are there familial / social circumstances (E.g. Financial hardship caused by inability to work) which constitute extenuating circumstances.
- Which recipient has been waiting for the longest period of time for a given organ?
  - I.e. Which potential recipients were added onto the waiting list earliest for a particular transplant / transplant center?
  - Where is the potential recipient registered? I.e. Is the recipient on the waiting list of the retrieval center? Of a center in the locality? Further afield?

The protocol followed for most types of organ to identify the final recipient takes into account these factors. However it passes through a series of steps which depend on the urgency codes and local rules. The general sequence is:

1. The donor becomes available and those organs which could be transplanted are identified and combined with data from the medical records of the donor (a single donor may be able to donate multiple organs in which case a process is opened for each).
2. A call is made to the OTA to check for highly urgent cases (so called Urgency Zero) which override all other priorities in the region, the regional OTA also has lists of urgency zero cases in the country – but makes a phone call to check no new cases have entered since the last update. If there are urgency zero cases with sufficient compatibility (blood group) then the assignment is made.
3. If not the OTA offers the organ back to the extraction center – at which point the transplant team head must accept or decline based on his/her list of patients.
4. If the extraction center refuses, a round robin system is used to call other possible implant systems to find a potential recipient. At each step the local team must decide whether to accept or decline based on the matching data.

5. If unsuccessful in the region, search goes further afield (inter-regional or international).
6. The informational background for these decisions is derived from the patient care record, immunology analyzes of the donor / donated organ and the physician's own knowledge (if any of the patients).

F) Outline System Architecture

This section provides a draft architecture for the CARREL trial automated organ transplant management system under development for the Catalonia region by Hospital St. Pau and UPC through FIS projects supported by the Spanish government. The CARREL system has been under development through several prototype versions with its architecture being adapted over time. The system deployment under the current project cycle is captured in the figure below.

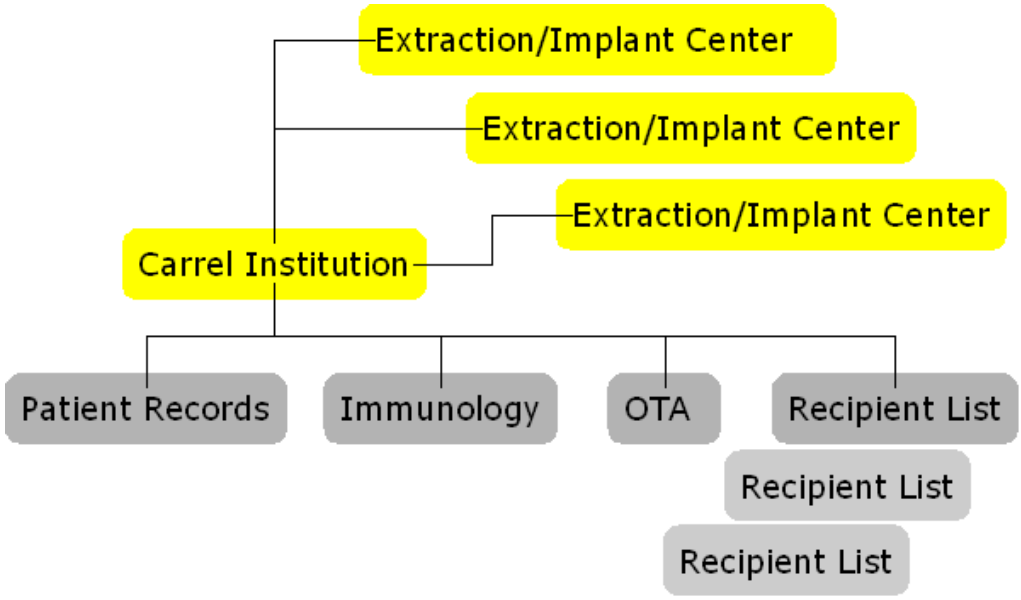

Figure 8: Transplant System Architecture (CARREL)

A full deployment for Catalonia would involve approximately the following number of services and systems:

- Hospital Sites with waiting lists (retrieval and/or implantation): 6 transplant capable sites, 10+ subsidiary medical centers associated with these sites.
- Number of patients on waiting list for a given organ type (e.g. Kidneys): 200-400
- Number of patients records of potential donors: 1-2 million
- Average demand: 2-3 per week.
- Peak demand: 2-3 demands concurrently.

The demonstration foreseen for early 2006 is expected to involve systems installed at 3-4 hospital, 1 immunology and 1 transplant coordination site, but with code scalable to at least a Catalonia wide deployment.

G) Possible provenance hooks

Figure 9 illustrates the project partner's current concept on how the CARREL architecture might be adapted to integrate provenance services. It is envisaged that each service in the system would keep a local data store (denoted by 'L' in the figure) recording mandated system records. Additionally, key services would register provenance data in application or system wide repositories (denoted by

'S1'-'S5'), which may in fact be collocated but in the general case it is assumed that they are distributed in the same way as the service components themselves.

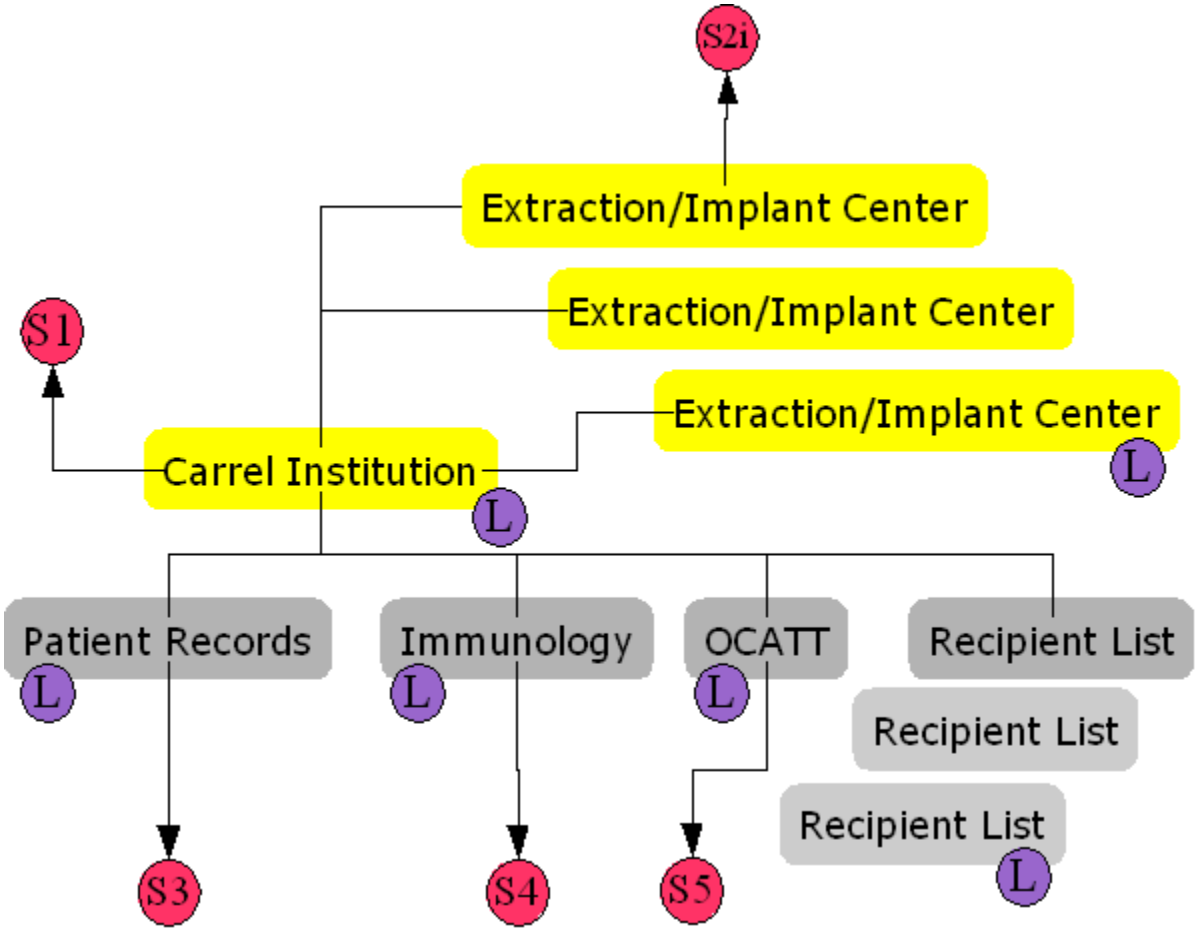

Figure 9: Possible provenance hooks in CARREL

2.2.2 Further application scenarios explored by the User Requirements Survey

2.2.2.1 eDiamond

A) Project overview

eDiamond is a research project at Oxford University that aims to build a national database of mammographic images for use in the clinical management of breast disease. This is achieved by building the necessary Grid infrastructure through a partnership between the Oxford e-Science Centre and industrial partners including IBM, working closely with clinical partners in four leading UK hospitals. Grid technology is used to develop tools that allow this database to be used in clinical diagnoses, epidemiological studies and in training and education of radiologists and clinicians.

For further details see: www.ediamond.ox.ac.uk

B) Primary actors in scenario

- Screening Radiologists
- Screening Radiographers
- Breast care unit administrators

- Student Radiologists
- Teacher Radiologists
- Student Radiographer
- Teacher Radiographer
- Epidemiologists

C) Description of the main workflows

1. *Breast Screening:*
    1. Image Capture
    2. Image upload to Grid
    3. Selection of images for clinic
    4. Radiologists read and diagnose images
    5. Upload diagnostic reports to Grid
    6. Decision made whether image is clear or patient recall needed
    7. If recall, then select images for patient clinic
    8. At patient clinic, decide whether patient clear or follow on appointment required

    Steps 3 and 7 managed by Breast care unit administrators. Exact processes dependant on individual healthcare trusts, for example, diagnosis may be by double-blind technique.
2. *Training:*
    1. Teacher prepares training roll (set of interesting images)
    2. Students work on selected images

    (Student assessment outside scope of workflow.)
3. *Epidemiology:*
    1. Epidemiologists select images with federated data

    (Anonymisation out of scope of workflow.)

D) Primary goals of using provenance

1. Protection of patient data and its usage
2. Ensure that doctors make correct diagnoses on correct data
3. Optimising the system as its distributed complexity increases

2.2.2.2 Healthcare and Life Sciences Framework

A) Project overview

The domain of Healthcare and Life Sciences is a heavily regulated industry where organizations need to prove adherence to proper procedures and best practices. This is an example application scenario provided by project partner IBM.

B) Primary actors in scenario

1. Clinical Trial Administrator
2. Laboratory Technician
3. Document Author

4. Document Approver

C) Description of the main workflows

1. A clinical trial is established.
2. A laboratory creates measurement data.
3. A report is created by analysing the measurement data.
4. The document may be modified through different drafts.
5. Once approved, the document's Provenance is linked to the clinical trial data that was used to create it.

D) Primary goals of using provenance

1. Build a trusted historical record of the events that affect a particular business process or object (such as a document).
2. Prove adherence to proper procedures and best practices.

2.2.2.3 Combechem

A) Project overview

Combechem is a UK e-Science project funded by the EPSRC. The project is working on Grid-enabled combinatorial chemistry, concentrating on crystallography and laser and surface chemistry. Another major component of the project is the development of an e-Lab, using pervasive computing technology to record detailed information on all aspects of laboratory work.

For further details see: www.combechem.org

B) Primary actors in scenario

- research scientists (PHD students, supervisors, RFs, staff)*
- publishers*
- technicians
- secretarial
- commissioners of the system
- undergraduate students
- health and safety officers
- university authorities
- arbitrary members of the public

** most heavy interaction*

C) Description of the main workflows

CombeChem experiments are a mixture of lab-based and software processes, and the group includes lab-based chemists, computer scientists and computational chemists. There are several distinct applications in the project, including: crystallography, synthetic-organic and simple-harmonic generation experiments. Experiments have typically few stages, e.g. 12 to 15 at most, but each stage may take several hours to several months.

In the crystallography application, the National Crystallography Service analyse crystals submitted to them by chemists. This is a very well-defined process of about 4 or 5 steps that determines the structure of the crystal and its comprising compound. The final results should be a data file containing refined atomic positions.

The synthetic-organic application is slightly less structured, but a rough idea of the workflow to be followed in each experiment will be known and encoded in advance (as it is required for health and safety reasons at least). At each stage of the experiment the experimenter will decide which next step to take based on the data produced at the last. This application is mostly lab-based (rather than software processes).

The simple-harmonic generation application, which analysis properties of liquids by bouncing lasers off them, is very unstructured and different processes and analysis will be attempted without a prior plan.

The computational chemists are processing result data from already performed chemistry experiments to try and determine connections between properties of materials. Some properties are easy to discover, such as the charge distribution around a molecule, while others are more difficult, such as the melting point of a molecule. Therefore, if a connection can be made between the two properties, a lot of time will be saved by discovering the easy to determine property and deriving the hard to determine one. Other experiments that the group are involved in involve simulating protein folding, protein docking and molecular dynamics.

D) Primary goals of using provenance

1. Human determination of the origin of data
2. Referencing and linking produced data
3. Recording execution of workflow that was not pre-defined
4. Third-party verification of produced data
5. Automated publication of results
6. Protection of intellectual property rights

2.2.2.4 myGrid

A) Project overview

Life science researchers traditionally chain together database searches and analytical tools, using complex scripts to overcome incompatibilities, or by manually cutting and pasting between web interfaces. These "in silico" experiments are usually undertaken without support for the scientific process of managing, sharing and reusing the results, their provenance, and the methods used to generate them. The myGrid project has developed a comprehensive loosely-coupled suite of middleware components specifically to support data intensive in silico experiments in biology. Workflows and query specifications link together third party and local resources using web service protocols. The software can be freely downloaded and has been used for building discovery workflows for investigations into Williams-Beuren Syndrome and Grave's Disease by collaborating Life Scientists.

myGrid is a UK EPSRC-funded e-Science pilot project made up of a consortium of UK Universities and institutes and supported by nine industrial partners of whom GSK and IBM are the most significant.

For further details see: www.mygrid.org.uk

B) Primary actors in scenario

Bioinformaticians and biologists.

C) Description of the main workflows

myGrid attempts to provide a useful working environment for bioinformaticians, particularly providing middleware that can be used by many parties. Experimental processes are automated or partially automated by encoding them as workflows and running them in a workflow enactment engine. This will then make calls to Web Services provided by various parties. Workflows can be shared, adapted and re-run with different data and services as desired.

myGrid has been concentrating on a few biological use cases, and two in particular that focus on determining the genetic cause of two human diseases: Graves' Disease and Williams-Bueren Syndrome. Lab-based experiments are followed by computational experiments, which may indicate further lab-based to be performed.

D) Primary goals of using provenance

1. Provide domain specific knowledge of the workflow being run, e.g. each workflow has a template that has inputs and outputs generated by each service in the workflow, establishing relationships between these inputs and outputs would be interesting in the Graves scenario.
2. Aggregate / stitch together results coming from workflows and other places, e.g. aggregating provenance from a lot of data sets pertaining to a high-level biological entity that you are trying to gather information over a period of time. This might answer questions like tracking results that a user got on a workflow relating to a genome on individual days (more like building a knowledge base on what you found in your workflow over a period of time). This should all be done in parallel with the workflow, but its probably domain specific.
3. Organizing results data. WBS produces a lot of results, provenance helps to provide some sort of context. For example visualization – overlaying the results onto the workflow diagram – helps to identify which data is the result of which service.
4. Validation of results, by comparing intermediate and final outputs from two different workflows.

2.2.2.5 GENSS (Grid-Enabled Numerical and Symbolic Services)

A) Project overview

GENSS is an EPSRC funded joint project between the University of Bath and Cardiff University.

The GENSS (Grid-Enabled Numerical and Symbolic Services) project addresses the combination of Grid computing and mathematical Web services, and their extension to deliver mathematical problem analysis, and the code and the resources to compute the answers, using a common open agent-based framework. The main research focus lies on matchmaking techniques for advertisement and discovery of mathematical services.

*For further details see:*genss.cs.bath.ac.uk

B) Primary actors in scenario

Scientists using mathematical web services to manipulate datasets.

C) Description of the main workflows

1. Client prepares a task description.
2. Description is sent to broker.
3. Broker identifies a set of applicable services and ranks them.
4. Broker returns service set to client for execution OR
5. Broker selects most applicable service and invokes it.
6. Broker returns results to client.

D) Primary goals of using provenance

1. Reproduction of results of computations.
2. Assessment of the quality of results.
3. Enabling/improving the searchability of results based on the associated metadata.

2.2.2.6 Traffic management

Traffic managers of a large town use a grid based infrastructure to find solutions for traffic problems arising in the town by analyzing simulation results generated by the system for possible traffic control interventions.

Provenance would be used for quality assurance purposes by assessing the simulation procedure. In concrete terms this means the certification of the simulation workflow selection and of the input parameter consistency to ensure reliable results.

This application scenario has been provided by Softeco Sismat SpA (Italy) being consortium member of the EU IST funded K-Wf Grid project. For further details see: www.kwfgrid.net.

2.2.2.7 DataMiningGrid

A) Project overview

EU IST funded project run by several partners including University of Ulster, Fraunhofer Institute for Autonomous intelligent Systems, DaimlerChrysler AG, Israel Institute of Technology and University of Ljubljana.

Data mining has been recognized as one of the most important information technologies for automating the process of analysing and interpreting the data in modern knowledge industries and high-tech sectors such as science and engineering. Currently there exists no coherent framework for developing and deploying data-mining applications on the Grid. The DataMiningGrid project will address this gap by developing generic and sector-independent data mining tools and services for the Grid. A test bed consisting of several applications from a diverse set of sectors will serve as platform for demonstrating and promoting the technology developed by the DataMiningGrid.

For further details see: www.datamininggrid.org

B) Primary actors in scenario

Scientists performing data-mining algorithms in grid environment.

C) Description of the main workflows

There are many use cases included in the project demonstration scenario, but the main workflow in the majority of these would be:

1. Select appropriate data service (services for manipulation – discovery, access etc. with data) from a registry of available data services.
2. Select available analysis services (services for processing of data) from a registry of available analysis services.
3. Execute preprocessing (possibly locally to the data source) on selected data source.
4. Execute processing (data-mining) algorithms (either near the data source or transfer and merge data in the user's local machine).
5. Store results.

D) Primary goals of using provenance

DataMiningGrid would use provenance to record data about processes/algorithms used in a workflow of processing raw data.

Possible other uses include:

- Security issues
- Record further data on experimental circumstances (e.g. what kind of methods/techniques, aparatures, meters were used, name of scientist and organization who performed the experiment).
- Improvement of job scheduler performance based on recorded provenance information.

2.2.2.8 DILIGENT (A DIgital Library Infrastructure on Grid ENabled Technology)

A) Project overview

DILIGENT is a project partially funded by the EU under the 2nd call of FP6 IST priority. It is coordinated scientifically by ISTI-CNR (Institute of Information Science and Technologies of the Italian National Research Council), involves 14 European partners and a number of international observers.

The main objective of DILIGENT is to create an advanced test-bed that will allow members of dynamic virtual e-Science organizations to access shared knowledge and to collaborate in a secure, coordinated, dynamic and cost-effective way. This test-bed will be built by integrating the Grid and Digital Library (DL) technologies. Merging of these different technologies will lay the foundations for a next generation e-Science knowledge infrastructure.

The DILIGENT infrastructure, which will build upon the efforts of the EGEE project (IST-2003-508833), will serve many different research and industrial applications. The test-bed will be demonstrated and validated by two complementary real-life application scenarios: one from the cultural heritage domain and one from the environmental e-Science domain. Additional objectives of the project are: i) to open up Grid technology to a broader range of research and industrial communities; ii) to broaden the diffusion of DLs by supporting a cost-effective DL operational model; iii) to promote cross-fertilization between the DL and Grid technology domains that will foster advances in both the areas. In order to achieve its objectives, DILIGENT has identified four main areas of work: 1) integration of DL services and content together with third-party applications as OGSA-compliant Grid services; 2) experimentation with real-life user communities; 3) feedback with respect to the capabilities of the Grid and the design of DL system architectures; 4) exploitation and sustainability.

For further details see: diligentproject.org.

B) Primary actors in scenario

Virtual Organisation members who access Virtual Digital Libraries.

C) Description of the main workflows

There are two demonstration scenarios in the DILIGENT project having the following goals:

- The goal of the ImpECt scenario is to improve accessibility, interoperability and usability of environmental data, models, tools, algorithms and instruments integrating the distributed data sources with specialized data handling services.
- The goal of the ARTE scenario is to stimulate collaborative, multidisciplinary scientific research, to ease multimedia artifact construction and improve support for education.

For a description of sample workflows in each scenario refer to the document “DILIGENT scenarios”. (Download location for this document is provided in Appendix A).

D) Primary goals of using provenance

Provenance would be used for process monitoring purposes in this application scenario. This involves monitoring the individual workflows carried out by digital library users as well as monitoring the services that constitute the digital library infrastructure.

2.3 General capabilities

Prior research on provenance has used several other terms including audit trail, lineage, dataset dependence and execution trace. A recent study on provenance issues by Miles, Groth, Branco and Moreau ([PASOA]) defines the term *provenance* as follows:

"We define the *provenance of a piece of data* as the documentation of the process that produced that data."

Generally speaking the provenance architecture should provide for the recording, management and querying of provenance information as well as the management of the provenance architecture itself.

2.4 General constraints

The software should be designed and implemented in a way that supports the ease of integration with existing software systems. The low costs of integration are important to make the introduction of a provenance architecture a reasonable choice versus own development of required functionality or omitting provenance related features.

The software should be designed and implemented in a way that also supports the use of existing provenance data representation standards as much as possible. This ensures interoperability with existing systems as well as future systems to be developed that adhere to these standards.

2.5 User characteristics

Section 2.2 ('Application scenarios') describes several use cases of the provenance architecture including the potential users (actors) in each scenario as well. As the use cases indicate, end users of the system may be potentially of any profession ranging from scientists through doctors to lawyers. However these end users are expected to interact with a layer of software built upon that middleware layer that is to be developed within this project. (This layered architecture is illustrated in the figure below.) Developers of the specific applications that make use of this middleware, i.e. direct users of the provenance architecture are expected to be IT professionals.

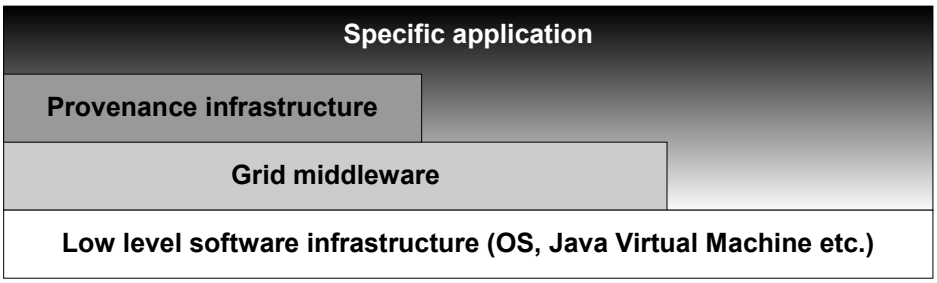

Figure 10 Place of the provenance architecture in the software stack

2.6 Operational environment

While the proposed project is to design the architecture required for provenance generation and reasoning, we sketch in the figure below some of its elements. First, provenance gathering is a collaborative process that involves multiple entities, including the workflow enactment engine, the enactment engine's client, the service directory and the invoked services. Provenance data will be submitted to one or more "provenance repositories" acting as storage for provenance data.

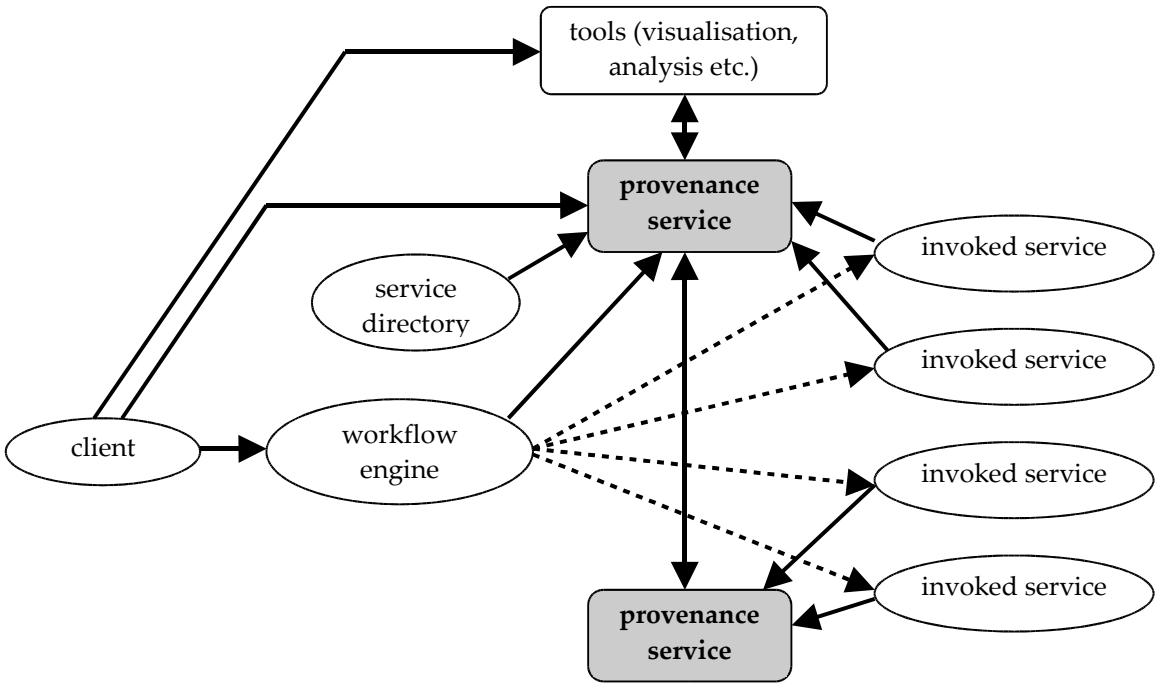

Figure 11: Operational environment of the provenance architecture

As Figure 2 indicates, the operational environment of the provenance architecture will consist mainly of software components, which can be grouped into two categories: grid middleware components and application layer components. The latter group is application specific, but the former one includes some widely used grid and other distributed technologies. Below is a statistical overview of the distributed software technologies that are used or planned to be used in the application scenarios examined as part of this requirements gathering process.

Grid middleware			
Globus Toolkit	version 2	1	10%
	version 3*	2	20%
gLite		1	10%
OGSA-DAI		1	10%
OMII		2	20%
EGEE		1	10%
Web Service related			
Tomcat/Axis*		5	50%
Taverna/FreeFluo		2	20%
Other			
CORBA*		1	10%
FIPA JADE (agent platform)*		1	10%
"Many different environments"		1	10%
**used in the demo applications*			

Table 1: Statistical overview of the usage of distributed technologies in the application scenarios

In terms of operating system and hardware platform the examined use cases cover all today's widely used OS and hardware platforms. Systems are typically heterogeneous themselves with respect to these parameters.

2.7 Assumptions and dependencies

Not applicable.

3 Specific requirements

This chapter contains the functional and constraint requirements placed by users on the provenance architecture. Requirements are described in this chapter in a narrative form, i.e. additional explanations and notes are provided to many of the requirements. For a tabular summary of the requirements refer to Appendix B.

3.1 Abstract level capability requirements

This section describes in detail what users would like to use provenance for in their application scenarios. This utilisation oriented approach can be regarded as a high or abstract level requirement specification for the provenance architecture, since eventually it has to support the operations described below.

3.1.1 Transplant application

This section lists a number of provenance questions which might be asked during or after the operation of the transplant systems. The questions and operations described below are taken as abstract level requirements on the provenance architecture as indicated in the text.

Definitions of scenario specific terms used in the requirements:

- *Case:* "In the organ transplant application a *case* is defined as the complete procedure from an organ becoming available through a donor to the completion of the transplant surgery. This is sometimes extended to include post operation care."
- *Incident report:* "An *incident report* is a document, which describes the circumstances under which a particular donation takes place."

Domain specific provenance questions:

(These questions can only be answered by processing domain specific content in recorded data.)

AR-1-1: The provenance system should support the following operation:
Check a given set of decisions in a case against the established rules to ensure that it is conformant. These rules may or may not be automatically enforced by the transplant management software – however in the general case many of them will not be. This provenance question is a post-hoc check as to whether rules were followed. (asked by Transplant Authority, Families, 3rd parties)

Flags: essential
Source: OTM

AR-1-2: The provenance system should support the following operation:
Derive a trace of the arguments, contributing factors and intermediate results which lead to a particular decision. (asked by Transplant Authority, Families, 3rd parties, Physicians)

Flags: essential
Source: OTM

AR-1-3: The provenance system should support the following operation:
Derive aggregate information across many cases such as the percentage of incidents of a certain type, success rates by center, etc. (asked by Transplant Authority, researchers, physicians)

Flags: essential
Source: OTM

AR-1-4: As an advanced feature the provenance system could support the following operation: Truth maintenance for "next best candidate" or other dynamic information. Advanced functionality: meaning that the system could be used to keep up to date pre-calculated lists of recipients ready for an incident. This is a type of result which may need to be modified as underlying data changes. (asked by transplant system itself, physicians)

Flags: nice to have
Source: OTM

Generic provenance questions:

(These questions can only be answered with derivations (reasoning) of some kind over recorded data but not using domain specific content.)

AR-1-5: The provenance system should support the following operation:
Extraction of an entire case-trace: gather all the records related to one incident into a single case-file. (asked by physicians, families, patients)

Flags: essential
Source: OTM

AR-1-6: The provenance system should support the following operation:
Identify all individual users related to an incident. (asked by physicians, Organ Transplant Authority, 3rd parties (legal challenges))

Flags: essential
Source: OTM

AR-1-7: The provenance system should support the following operation:
Provide a simulated walkthrough on service execution flow and verify this against template workflows and/or rules governing procedures (sophistication may vary). (asked by physicians, organ transplant authority, 3rd parties (legal challenges))

Flags: essential
Source: OTM

AR-1-8: As an advanced feature the provenance system could support the following operation:
Identify abstract derivation process of the result – based on some shared high level notions of the types of actions/content logged (e.g. having a standard view of what is an assertion, what is a decision etc.) and what follows what.

Flags: nice to have
Source: OTM

3.1.2 TENT

The TENT system is a framework for integrating applications to perform complex simulations in a workflow style and provenance requirements may vary depending on the actual deployment of TENT for a given purpose (project). Generally speaking the following is required for TENT from the provenance architecture:

AR-2-1: The provenance architecture should be able to store all kinds of information that is needed to trace back the preceding process of data transformation within a workflow.

Flags: essential
Source: TENT

By the examination of a concrete application of TENT in the "SikMa" project *(see appendix A for a detailed description of this application scenario)* the following additional high level requirements have been identified:

AR-2-2: Recorded provenance information should make it able to automatically restart workflows or parts of a workflow by the TENT system.

Flags: essential
Source: TENT/SikMa

Note: "The provenance architecture shall not take over the responsibility of restarting the workflow, but shall give the user the possibility to search for entry points in order to provide a restart. Appropriate queries to the provenance architecture shall reveal such entry points to TENT. Of course the TENT system itself has all necessary information to perform a restart."

AR-2-3: The provenance architecture should be able to provide a trusted historical record of user access to produced data during a workflow (including intermediate data, result data and associated metadata as well), which can be used as evidence that the given data set has been accessed only by authorised users (as specified by the initiator of the workflow).

Flags: essential
Source: TENT/SikMa

AR-2-4: The provenance architecture should make it able to identify unauthorised accesses to produced data during a workflow (including intermediate data, result data and associated metadata). Access rights are specified by the initiator of the workflow.

Flags: essential
Source: TENT/SikMa

3.1.3 eDiamond

According to the goals of using provenance in this scenario the following analysis and reasoning operations are necessary:

AR-3-1: For the protection of patient data and its usage, the provenance system should support the following operation:
Report to show that an imposed policy has (or has not) been followed.

Flags: desirable
Source: eDiamond

AR-3-2: In order to ensure that doctors make correct diagnoses on correct data, the provenance system should support the following operation:
List all occasions an image has been used and find the diagnosis produced by the processes applied to it.

Flags: desirable
Source: eDiamond

AR-3-3: For the diagnosis of process failure cases, the provenance system should support the following operation:
Perform a network analysis of paths within a process. Identify bottlenecks in the process using elapsed timing information.

Flags: desirable
Source: eDiamond

3.1.4 Healthcare and Life Sciences Framework

According to the goals of using provenance in this scenario the following analysis and reasoning operations are necessary:

AR-4-1: Proof of correct process management is required to be supported by the provenance architecture. Examples of policies and processes to be followed include the regulations defined by the Federal Drugs Administration in the US.

Flags: desirable
Source: HLSF

AR-4-2: Proof that created data has not been tampered with is required to be supported by the provenance architecture.

Flags: desirable
Source: HLSF

3.1.5 Scientific applications including Combechem, myGrid and GENSS

Use cases of provenance in the scientific applications have been summarised in [PASOA] as the following general tasks. Each one is taken as an abstract level requirement for the provenance architecture.

AR-5-1: The provenance architecture should support the following operation:
Accessing a historical record or aide memoire of work conducted.

Flags: desirable
Source: scientific applications

AR-5-2: The provenance architecture should support the following operation:
Proving that the experiment claimed to have been done was actually done.

Flags: desirable
Source: scientific applications

AR-5-3: The provenance architecture should support the following operation:
Proving that the experiment done conformed to a required standard.

Flags: desirable
Source: scientific applications

AR-5-4: The provenance architecture should support the following operation:
Checking that the experiment was performed correctly, and the services involved used correctly.

Flags: desirable
Source: scientific applications

AR-5-5: The provenance architecture should support the following operation:
Verifying that services used are working as they should be.

Flags: desirable
Source: scientific applications

AR-5-6: The provenance architecture should support the following operation:
Checking whether results were due to interesting features of the material being experimented on or nuances of the experiment performed.

Flags: desirable
Source: scientific applications

AR-5-7: The provenance architecture should support the following operation:
Linking together data and experiments by their provenance data to provide extra context to understanding those experiments.

Flags: desirable
Source: scientific applications

AR-5-8: The provenance architecture should support the following operation:
Tracing where data came from and the processes it had been through to reach its current form.

Flags: desirable
Source: scientific applications

AR-5-9: The provenance architecture should support the following operation:
Tracing which source data was used to produce given result data and vice-versa.

Flags: desirable
Source: scientific applications

AR-5-10: The provenance architecture should support the following operation:
Providing the process information required for publishing an experiment's results.

Flags: desirable
Source: scientific applications

AR-5-11: The provenance architecture should support the following operation:
Deriving the higher-level processes that have been gone through to perform an experiment, so that they can be checked and re-used.

Flags: desirable
Source: scientific applications

AR-5-12: The provenance architecture should support the following operation:
Allowing experiments to be re-enacted to check that services and/or data has not changed in a way which affects the results.

Flags: desirable
Source: scientific applications

AR-5-13: The provenance architecture should support the following operation:
Determining the probable effectiveness of similar future experiments.

Flags: desirable
Source: scientific applications

3.1.6 Traffic Management Application

AR-6-1: The provenance architecture should support the certification of the simulation workflow against a reference workflow description.

Flags: desirable
Source: TMA

AR-6-2: The provenance architecture should support the certification of input parameters against reference schemas for consistency.

Flags: desirable
Source: TMA

3.1.7 DataMiningGrid

AR-7-1: The provenance architecture should support:
Recording data about processes and/or algorithms used in a workflow of processing raw data.

Flags: desirable
Source: DMG

AR-7-2: The provenance architecture should support:
Providing a trusted historical record of user access to confidential data.

Flags: nice to have
Source: DMG

3.2 Technical level capability requirements

This section contains requirements that directly apply to the provenance architecture.

The structure of this chapter is the same as the User Requirements Survey.

3.2.1 Characteristics of provenance data

3.2.1.1 Transplant application

As a first approximation it is expected that automated logging mechanisms for the transplant application would need to record the following raw data and information:

TR-1-1-A-1: Recording of the following provenance information is required:
Service invocation: Who accessed a particular service, when, with what input parameters (or a summary thereof) and on whose authority. 'Who' can refer to either a human or a service.

Flags: essential
Source: OTM

TR-1-1-A-2: Recording of the following provenance information is required:
Service response: Who a service sent data messages to, in response to which invocation, the content of the response (or a summary thereof). 'Who' can refer to either a human or a service.

Flags: essential
Source: OTM

TR-1-1-A-3: Recording of the following provenance information would be useful:
Information state: A summary of the information state in the service at the time a particular action is taken.

Flags: nice to have
Source: OTM

TR-1-1-A-4: In addition to the logging of message based activities the provenance service also needs to capture "side effect" type actions submitted by the application (e.g. those which may not directly lead to a response message):

- Carrying out an action in the real world
- Recording a decision or fact

Flags: essential
Source: OTM

Chapter 4 of the scenario document provided for the OTM application describes what the users expect at a more detailed logging level to be recorded by the provenance architecture. For this document see Appendix A.

OTM notes:

"In general we would expect that the domain content of records themselves (both detailed records within a service) and summaries (in provenance services) would be specified by legal regulations in the field. As would the rules on who/what would later be allowed to retrieve this data (i.e. what a patient care record database must record about any data access)."

The list of applicable regulations for this application can be found in section 3.3.3 ('Legal and ethical issues').

3.2.1.2 TENT

The following requirements derive from a concrete application of the TENT system in the SikMa project.

TR-1-1-B-1: For the output of the TAU module the version information of the involved TAU code should be recorded by the provenance system.

Flags: essential
Source: TENT/SikMa

TR-1-1-B-2: For a given output of the TAU module the processed input files should be recorded by the provenance system.

Flags: essential
Source: TENT/SikMa

TR-1-1-B-3: For a given output of the Aeroelastic Module the processed input files should be recorded by the provenance system.

Flags: essential
Source: TENT/SikMa

TR-1-1-B-4: Rejection of job submission by the TENT framework to cluster batch systems should be recorded by the provenance system, so this event can be recognised by TENT and it can restart the workflow or the appropriate modules.

Flags: essential
Source: TENT/SikMa

TR-1-1-B-5: The provenance architecture shall provide a way to map TENT access rights to ensure that no misuse of provenance data will take place.

Flags: essential
Source: TENT/SikMa

3.2.1.3 eDiamond

TR-1-1-C-1: The following provenance information should be captured:
Which process were executed together with their input and output.

Flags: desirable
Source: eDiamond

TR-1-1-C-2: The following provenance information should be captured:
Who executed the process and when.

Flags: desirable
Source: eDiamond

TR-1-1-C-3: The following provenance information should be captured:
Processing elapsed times.

Flags: desirable
Source: eDiamond

TR-1-1-C-4: The following provenance information should be captured:
Location and version information.

Flags: desirable
Source: eDiamond

TR-1-1-C-5: The following provenance information should be captured:
Request and response messages.

Flags: desirable
Source: eDiamond

TR-1-1-C-6: The following provenance information should be captured:
Other context information, which may include:

- How was this service discovered?
- policies established to invoke a service
- information in SOAP message headers (security, reliability, transactionality etc.)

Flags: desirable
Source: eDiamond

3.2.1.4 Healthcare and Life Sciences Framework

TR-1-1-D-1: The following provenance information is required to be recorded:
The identity of the source of each provenance data entry.

Flags: desirable
Source: HLSF

TR-1-1-D-2: The following provenance information is required to be recorded:
The date and time that each provenance data entry is created.

Flags: desirable
Source: HLSF

TR-1-1-D-3: The provenance architecture should support storing the following information with provenance data:
Attributes for each provenance data entry that may reference other objects stored either outside or inside the provenance repository.

Flags: desirable
Source: HLSF

3.2.1.5 myGrid

TR-1-1-E-1: The following provenance information should be recorded:
Version information (algorithms, databases).

Flags: desirable
Source: myGrid

TR-1-1-E-2: The following provenance information should be recorded:
Logs of what was done.

Flags: desirable, critical
Source: myGrid

TR-1-1-E-3: The following provenance information should be recorded:
Estimates of quality of service metrics such as execution time.

Flags: desirable
Source: myGrid

3.2.1.6 Combechem

TR-1-1-F-1: The following provenance information is required to be stored:
Identity of process and version.

Flags: desirable
Source: CombeChem

TR-1-1-F-2: The following provenance information is required to be stored: Identity of operator.

Flags: desirable
Source: CombeChem

TR-1-1-F-3: The following provenance information is required to be stored: Time.

Flags: desirable
Source: CombeChem

TR-1-1-F-4: The recording of 'ambient conditions' of the experiments is required, like e.g. temperature. These parameters are known for the system during execution.

Flags: desirable
Source: CombeChem

TR-1-1-F-5: The result and intermediate data of an experiment should be available and referenceable so that it can be linked to from papers and discovered for use in other experiments.

Flags: desirable
Source: CombeChem, particularly the crystallography use case

3.2.1.7 GENSS

TR-1-1-G-1: The following provenance information is required to be stored: Calendrical information.

Flags: desirable
Source: GENSS

TR-1-1-G-2: The following provenance information is required to be stored: Algorithmic information.

Flags: desirable
Source: GENSS

TR-1-1-G-3: The following provenance information is required to be stored: Parameter information.

Flags: desirable
Source: GENSS

3.2.1.8 Traffic management application

TR-1-1-H-1: The following information should be recorded: Configuration parameters of simulation processes.

Flags: desirable
Source: TMA

TR-1-1-H-2: The following information should be recorded: Input parameters of simulation processes.

Flags: desirable
Source: TMA

3.2.1.9 DataMiningGrid

TR-1-1-i-1: The following information should be recorded: Data about processes/algorithms used in a workflow of processing raw data.

Flags: desirable
Source: DMG

3.2.1.10 Other requirements

TR-1-2: The system should provide a way for the user to annotate the provenance data.

Flags: essential
Source: TENT, eDiamond, HLSF, myGrid, Combechem

3.2.2 Export and API format of provenance data

Note:

- "API format" refers to the format of the data sent by application services to the provenance store.
- "Export format" refers to the format of the export of a provenance store contents to other applications that may analyse it.

Requirements imposed on this issue by the individual applications:

TR-2-1-A: "Format must be a non-proprietary format which can in principle be used with another tool (to be built if necessary) without violating IPR rules. An open standard would be best."

Flags: essential, critical
Source: OTM

TR-2-1-B: "The preferred API format for provenance data is the W3C Resource Description Framework (RDF). This is not mandatory, reports generated from the provenance data may be in other XML based formats and would have to conform to formats specified by external regulatory bodies. However RDF is preferred since this fits well with other standards in the Healthcare and Life Sciences arena, such as the Life Sciences Identifier (LSID)."

Flags: desirable
Source: HLSF

Note: "API format" refers to the data sent by application services to the provenance store.

TR-2-1-C: The export format of the provenance system should be XML defined by XML Schema.

Flags: desirable
Source: eDiamond

TR-2-1-D: The export format of the provenance system should be XML-based.

Flags: desirable
Source: GENSS

3.2.3 Storage and export of provenance data

TR-3-1: The system should support the multiple storage of a provenance record, i.e. the system should provide a way to store copies of a provenance record in more than one repository.

Flags: desirable
Source: eDiamond, HLSF

TR-3-2: The system should support the recording of different views on provenance information regarding to an event or an entity.

Flags: essential
Source: OTM, eDiamond, HLSF

TR-3-3: The system should support the migration of provenance data among provenance repositories.

Flags: essential
Source: OTM, eDiamond, HLSF

TR-3-4-A: On the fly recording of provenance data should be supported by the system.

Flags: essential
Source: OTM, TENT, eDiamond, HLSF, DMG

TR-3-4-B: Batch recording of provenance data should be supported by the system.

Flags: desirable
Source: eDiamond, HLSF, DMG

TR-3-5-A: The system should support the storage of recorded provenance data for a complete simulation session. Runtime for a simulation session is between 1 minute and 1 month, its typical value is a few days.

Flags: essential
Source: TENT

TR-3-5-B: The system should support the storage of recorded provenance data for an indefinite period of time.

Flags: desirable
Source: eDiamond, HLSF

TR-3-5-C: The system should support the storage of recorded provenance data for 3-4 years.

Flags: desirable
Source: myGrid

TR-3-6: The system should be able to archive recorded provenance data.

Flags: essential
Source: OTM, TENT, eDiamond, HLSF

TR-3-7: The system should be able to export recorded provenance data for external usage.

Flags: essential
Source: OTM, TENT, myGrid, CombeChem, GENSS, eDiamond, HLSF, DMG

3.2.4 Utilisation of provenance data

TR-4-1: It should be possible to query all of the data associated with a particular provenance entry, or return all of the provenance entries that have attributes matching a search criteria.

Flags: desirable
Source: HLSF

Note: The term ‘provenance entry’ refers to the information written to a provenance service.

TR-4-2: The architecture should support the dynamic processing of provenance data, i.e. recorded provenance data should be instantly queriable even if a recording session

(recording of interrelated provenance records belonging to e.g. the same workflow) is still in progress.

Flags: essential
Source: OTM, TENT, CombeChem, myGrid, eDiamond, HLSF

TR-4-3: The provenance architecture should support the storage of results of analysis and reasoning operations performed on the provenance data by tools that are not part of the generic architecture (3rd party tools on the application layer).

Flags: essential
Source: TENT, Combechem, eDiamond, HLSF

3.2.5 Operation of the provenance architecture

TR-5-1: The provenance architecture should support for the maximum automation of the provenance recording mechanism.

Flags: desirable
Source: eDiamond, HLSF, TMA, DMG

TR-5-2: Provenance handling should be policy-driven.

Flags: desirable, critical (eDiamond)
Source: eDiamond, HLSF

TR-5-3: The provenance architecture should be deployable as an integrated part of a system, as a service within the same administrative domain as the client system and as a 3rd (external) party operated service, too.

Flags: essential
Source: OTM (all), TENT (integrated), myGrid (integrated), CombeChem (integrated), GENSS (3rd party), eDiamond (same administrative domain, 3rd party), HLSF (all), TMA (integrated), DMG (integrated), DILIGENT (integrated, 3rd party)

TR-5-4: Client side components of the provenance architecture should not block workflow if provenance services are unavailable and client explicitly expresses their wish to turn off provenance recording.

Flags: desirable
Source: myGrid

3.2.6 Interface

TR-6-1: The architecture should support a rich set of generic APIs that allow analysis and reasoning tools to be built upon.

Flags: essential, critical (eDiamond)
Source: OTM, TENT, eDiamond, HLSF, TMA

TR-6-2: Human-computer interfaces presented by the system for analysis and reasoning should be designed to allow multilingual support

Flags: essential
Source: OTM

TR-6-3: Human-computer interfaces presented by the system for analysis and reasoning should be usable by a non computer scientist.

Flags: desirable
Source: GENSS

TR-6-4-A: The provenance architecture should provide a programmatic interface for the administration of the system.

Flags: essential, critical (eDiamond)
Source: TENT, eDiamond, HLSF

TR-6-4-B: The administrative interface of the provenance architecture should be able to be accessed and controlled through it's API. It has to be integratable into TENT or at least be accessible through the TENT system. Therefore some kind of user authentication may additionally be needed.

Flags: essential
Source: TENT

Question number 4.6.2 in the URS asked the users what kinds of information they would found useful to see on a HCI presented by the provenance system. A statistical overview of the answers is presented below. *Note:* there are use cases for which no HCIs are required at all (e.g. TENT).

details of all service invocations (e.g. all inputs, outputs and when)	4	40%
the services that were selected for execution	5	50%
statistics of execution of services invoked (e.g. their load, their accuracy)	5	50%
information on services invoked (e.g. algorithms they use, libraries used, version of the code used, database or external services they may rely on, the institution/person that hosts the service)	5	50%
motivational/contextual information for the execution: why this was run, by whom	6	60%
higher-level information on the execution not explained in terms of low level service description but "scientific terms" such as sequence alignment etc.	8	80%

Table 2: Answers for question 4.6.2 of the User Requirements Survey

TR-6-5-A: Provenance information should be trackable on human-computer interfaces presented by the system at set level (e.g. database table or spreadsheet).

Flags: essential
Source: OTM, HLSF, GENSS, TMA, DMG, DILIGENT

TR-6-5-B: Provenance information should be trackable on human-computer interfaces presented by the system at individual data items (e.g. record in database or cell in spreadsheet).

Flags: desirable
Source: eDiamond, HLSF, DILIGENT

TR-6-5-C: The granularity of provenance information displayed by the system on a human-computer interface should be configurable based on policies.

Flags: desirable
Source: HLSF

TR-6-6-A: Provenance information displayed by the system on a HCI should be updatable on user request.

Flags: essential
Source: OTM, eDiamond, GENSS

TR-6-6-B: HCIs presented by the provenance system for provenance monitoring should support continuous monitoring, i.e. the displayed information should be updated automatically on every change as soon as possible.

Flags: essential
Source: OTM, GENSS, DMG, DILIGENT

TR-6-6-C: Provenance information displayed by the system on a HCI should be updated on each execution session of the monitored application.

Flags: desirable
Source: TMA

TR-6-6-D: The update frequency of provenance information displayed by the system on a HCI should be configurable based on policies.

Flags: desirable
Source: HLSF

3.2.7 System documentation

TR-7-1: There should exist different levels of system documentation, including the following:

- a detailed API documentation for programmers who intend to integrate the provenance architecture into their systems,
- a detailed description of the administrative interface of the system for system administrators,
- a detailed description of other human-computer interfaces presented by the system e.g. for analysis and reasoning. Different audiences should be taken into account here including end-users as well, who want to use the provided tools as a stand-alone applications.

Flags: essential
Source: TENT

3.3 Constraint requirements

3.3.1 Performance constraints

Requirements on execution overhead due to provenance data generation and handling:

CR-1-1-A: Provenance recording should not impede a human entering data in real time.

Flags: essential
Source: OTM

CR-1-1-B: Within TENT the execution overhead due to provenance recording has the upper constraint of not affecting the interaction with the system in a significant manner. In terms of the applications used in TENT workflows: due to typical execution times of e.g. the flow solver TAU, overhead has to be kept at minimum level.

Flags: essential
Source: TENT

CR-1-1-C: Provenance recording should not slow down workflow execution by significant magnitude. (Significant not quantified.)

Flags: desirable
Source: myGrid

CR-1-1-D: Provenance recording should increase end-to-end elapsed execution time by no more that 5%.

Flags: desirable
Source: eDiamond

Requirements on storage overhead due to provenance data generation and handling:

CR-1-2-A: Recorded provenance data should not exceed 20% of overall system record data.

Flags: essential
Source: OTM

CR-1-2-B: There are the same constraints for storage overhead as for execution overhead (see CR-1-1-B), but less restricted.

Flags: essential
Source: TENT

CR-1-2-C: Recorded provenance data should be less than 100 KBytes per patient image. (Patient images in eDiamond are usually around 32 MB.)

Flags: desirable
Source: eDiamond

3.3.2 Quality of service attributes

CR-2-1: Generated provenance data must not be lost.

Flags: desirable
Source: eDiamond

CR-2-2: The provenance architecture should guarantee reliable once-and-once-only delivery as much as technically possible and up to the measure it depends on the architecture itself and not on operating conditions.

Flags: desirable
Source: eDiamond

3.3.3 Legal and ethical issues

Explored application scenarios have identified the following legal regulations that affect data handling in their systems – potentially imposing constraints on provenance data as well:

Transplant application:

The following four laws bound all activity in the area of organ/tissue transplantation:

- Law 30/79, 28th October, 1979: On the extraction and transplantation of organ.
- Orden Ministerio de Sanidad y Consumo 29th June 1987: testing for HIV in operations of procurement and implantation of human organs.
- Real Decreto 411/1996, 1st March, 1996: Regulation of activities relative to the use of human tissues.
- Real Decreto 2070/1999/30th December: regulating activities related to the procurement and clinical usage of human organs and tissues.

In addition to these activities are covered by more general medical laws – the most important of these are:

- The element of the Hippocratic Oath which states that a physician should preserve a patient's privacy
- Spanish national electronic data protection policies.

OTM states that these legal regulations change very rarely (space of 5-10 years), however they may change more rapidly in terms of local policies as electronic systems are just now being established.

eDiamond:

- UK Data Protection Activities
- Medical and Ethical rules

Healthcare and Life Sciences Framework:

- Global and national statutes for management of information

Note: Though responders identified these regulations, we have received no descriptions on the particular constraints that these regulations impose on data handling, especially on provenance data.

CombeChem:

CR-3-1: The provenance data should provide ability to ensure that appropriate regulations (such as those set by bodies like the Food and Drug Administration or the health and safety rules of a department) were adhered to.

Flags: desirable
Source: CombeChem

CR-3-2: The provenance data should provide protection for intellectual property right issues, for example through the use of digital signatures and time stamping.

Flags: desirable
Source: CombeChem

3.3.4 Security related issues

CR-4-1: The provenance architecture should have a configurable, fine-grained access control system over recorded provenance data.

Flags: essential, critical (myGrid)
Source: OTM, myGrid, eDiamond, TMA, HLSF

CR-4-2: The provenance architecture should allow both automated and manual determination of access control rights on recorded provenance data.

Flags: essential
Source: OTM, eDiamond, DILIGENT

CR-4-3: Access rights to the provenance system must be consistent with access rights to the rest of the TENT system. The provenance system should provide a way to map access rights information of TENT into its security subsystem. Access rights are stored in TENT in an LDAP server.

Flags: essential
Source: TENT

CR-4-4-A: The provenance architecture should be configurable in a way that assigns the following access rights to the given user groups:

- User: Access to provenance data directly involved in the data manipulation process of the simulation (s)he has started and configured.
- System designer: Access to secondary provenance data, which is the collection of all user provenance data and derivations from them for analyzing and reasoning purposes.

- System developer: Access to all kinds of provenance data. This especially includes data coming directly from the TENT core components. This data has to be visible only for this user group.

Flags: essential
Source: TENT

CR-4-4-B: The provenance architecture should be configurable in a way that assigns the following access rights to the given user groups:

- Digital Library user: Access to the provenance data of his/her own initiated processes.
- DILIGENT Administrator: General access.
- Virtual Organisation Manager, Digital Library Manager, DILIGENT Resource Manager: Access to his/her own local provenance data.

Flags: desirable
Source: DILIGENT

CR-4-5: The security infrastructure of the provenance architecture should have single sign-on.

Flags: desirable
Source: eDiamond

CR-4-6: The security infrastructure of the provenance architecture should be the same as the one of the application – particularly for any end-user clients.

Flags: desirable
Source: eDiamond

Note: In the case of eDiamond this infrastructure is Globus GSI integrated with OGSA-DAI. However it is not a stable version at the moment, because the National Health Service are changing their infrastructure and moving towards PKI.

CR-4-7: The provenance architecture should provide a mechanism for recording adequate provenance data in an unmodifiable way to make results non-repudiable.

Flags: desirable
Source: myGrid

3.3.5 Other constraints

Requirements in this section have been inferred from the answers provided for the following questions in the URS:

- "What criteria does a provenance architecture have to match that would make you consider integrating it into your system?" (for scenarios without existing provenance architecture)
- "What criteria does a new provenance architecture have to match that would make you consider replacing your existing provenance architecture with it?" (for applications with existing provenance architecture)

These requirements are marked with the 'critical' flag.

Requirements that were also stated in the rest of the URS or in the additional scenario documents provided by the partners – and therefore already contained in the previous sections of this document – are not repeated here.

CR-5-1: The provenance architecture should have good application fit, meaning: meet the basic logging needs and have additional potential for more complex questions outlined in the scenario description.

Flags: essential, critical
Source: OTM

CR-5-2: The provenance architecture should have the properties of cost efficiency and robustness versus an in-application hand-engineered logging system.

Flags: essential, critical
Source: OTM

CR-5-3: The provenance system should be capable of handling huge amounts of provenance data coming in very frequently from the application itself. It should not create any bottlenecks disturbing the system.

Flags: essential, critical
Source: TENT

CR-5-4: The provenance system should provide more and more detailed information about the different data and control flows taken place during workflow execution.

Flags: essential, critical
Source: TENT

CR-5-5: On top of the API of the provenance system TENT must be able to access all its functions and provide them to the users through appropriate interfaces.

Flags: essential, critical
Source: TENT

CR-5-6: The provenance architecture should be loosely coupled and independent from the application so that current system is unaffected. Provenance can depend on the application, but the application should not depend on the provenance.

Flags: desirable, critical
Source: eDiamond

CR-5-7: Tooling should be based on published APIs and not on hidden internal APIs.

Flags: desirable, critical
Source: eDiamond

CR-5-8: The provenance architecture should support transparent integration and operation with the DILIGENT infrastructure.

Flags: desirable, critical
Source: DILIGENT

CR-5-9: Provenance mechanisms should be handled at grid middleware level and/or as a third party service.

Flags: desirable, critical
Source: DILIGENT

Appendix A Source material reference

Source material for this document including the User Requirement Surveys and additional scenario documents are available on the website of the Provenance project at the following location:

http://twiki.gridprovenance.org/bin/viewauth/Restricted/CollectedSources

The additional scenario documents available are as follows:

- *Transplant Application*
  - Outline Organ Transplant Management Scenario: GRID Provenance Project
- *TENT*
  - Outline aerospace scenario
  - Provenance requirements of SikMa
  - PROVENANCE Data in TENT
- *Healthcare and Life Sciences Framework*
  - Healthcare and Life Sciences Framework – Scenarios for Provenance
- *CombeChem*
  - Provenance Requirements of CombeChem
- *myGrid*
  - Provenance Requirements of myGrid
- *DILIGENT*
  - DILIGENT architecture
  - DILIGENT scenarios

Appendix B Table of specific requirements

This appendix contains a tabular summary of the requirements described in Chapter 3.

1 Abstract level capability requirements

ID	*Textual description*	*Flags*	*Source*
Transplant application			
AR-1-1	The provenance system should support the following operation: Check a given set of decisions in a case against the established rules to ensure that it is conformant. These rules may or may not be automatically enforced by the transplant management software – however in the general case many of them will not be. This provenance question is a post-hoc check as to whether rules were followed. (asked by Transplant Authority, Families, 3rd parties)	*essential*	OTM
AR-1-2	The provenance system should support the following operation: Derive a trace of the arguments, contributing factors and intermediate results which lead to a particular decision. (asked by Transplant Authority, Families, 3rd parties, Physicians)	*essential*	OTM
AR-1-3	The provenance system should support the following operation: Derive aggregate information across many cases such as the percentage of incidents of a certain type, success rates by center, etc. (asked by Transplant Authority, researchers, physicians)	*essential*	OTM
AR-1-4	As an advanced feature the provenance system could support the following operation: Truth maintenance for "next best candidate" or other dynamic information. Advanced functionality: meaning that the system could be used to keep up to date pre-calculated lists of recipients ready for an incident. This is a type of result which may need to be modified as underlying data changes. (asked by transplant system itself, physicians)	*nice to have*	OTM
AR-1-5	The provenance system should support the following operation: Extraction of an entire case-trace: gather all the records related to one incident into a single case-file. (asked by physicians, families, patients)	*essential*	OTM

ID	*Textual description*	*Flags*	*Source*
AR-1-6	The provenance system should support the following operation: Identify all individual users related to an incident. (asked by physicians, Organ Transplant Authority, 3rd parties (legal challenges))	*essential*	OTM
AR-1-7	The provenance system should support the following operation: Provide a simulated walkthrough on service execution flow and verify this against template workflows and/or rules governing procedures (sophistication may vary). (asked by physicians, organ transplant authority, 3rd parties (legal challenges))	*essential*	OTM
AR-1-8	As an advanced feature the provenance system could support the following operation: Identify abstract derivation process of the result – based on some shared high level notions of the types of actions/content logged (e.g. having a standard view of what is an assertion, what is a decision etc.) and what follows what.	*nice to have*	OTM
TENT			
AR-2-1	The provenance architecture should be able to store all kinds of information that is needed to trace back the preceding process of data transformation within a workflow.	*essential*	TENT
AR-2-2	Recorded provenance information should make it able to automatically restart workflows or parts of a workflow by the TENT system.	*essential*	TENT
AR-2-3	The provenance architecture should be able to provide a trusted historical record of user access to produced data during a workflow (including intermediate data, result data and associated metadata as well), which can be used as evidence that the given data set has been accessed only by authorised users (as specified by the initiator of the workflow).	*essential*	TENT
AR-2-4	The provenance architecture should make it able to identify unauthorised accesses to produced data during a workflow (including intermediate data, result data and associated metadata). Access rights are specified by the initiator of the workflow.	*essential*	TENT
eDiamond			
AR-3-1	For the protection of patient data and its usage, the provenance system should support the following operation: Report to show that an imposed policy has (or has not) been followed.	*desirable*	eDiamond

ID	*Textual description*	*Flags*	*Source*
AR-3-2	In order to ensure that doctors make correct diagnoses on correct data, the provenance system should support the following operation: List all occasions an image has been used and find the diagnosis produced by the processes applied to it.	*desirable*	eDiamond
AR-3-3	For the diagnosis of process failure cases, the provenance system should support the following operation: Perform a network analysis of paths within a process. Identify bottlenecks in the process using elapsed timing information.	*desirable*	eDiamond
Healthcare and Life Scineces Framework			
AR-4-1	Proof of correct process management is required to be supported by the provenance architecture. Examples of policies and processes to be followed include the regulations defined by the Federal Drugs Administration in the US.	*desirable*	HLSF
AR-4-2	Proof that created data has not been tampered with is required to be supported by the provenance architecture.	*desir*able	HLSF
Scientific applications including Combechem, myGrid and GENSS			
AR-5-1	The provenance architecture should support the following operation: Accessing a historical record or aide memoire of work conducted.	*desirable*	scientific applications
AR-5-2	The provenance architecture should support the following operation: Proving that the experiment claimed to have been done was actually done.	*desirable*	scientific applications
AR-5-3	The provenance architecture should support the following operation: Proving that the experiment done conformed to a required standard.	*desirable*	scientific applications
AR-5-4	The provenance architecture should support the following operation: Checking that the experiment was performed correctly, and the services involved used correctly.	*desirable*	scientific applications
AR-5-5	The provenance architecture should support the following operation: Verifying that services used are working as they should be.	*desirable*	scientific applications
AR-5-6	The provenance architecture should support the following operation: Checking whether results were due to interesting features of the material being experimented on or nuances of the experiment performed.	*desirable*	scientific applications

ID	*Textual description*	*Flags*	*Source*
AR-5-7	The provenance architecture should support the following operation: Linking together data and experiments by their provenance data to provide extra context to understanding those experiments.	*desirable*	scientific applications
AR-5-8	The provenance architecture should support the following operation: Tracing where data came from and the processes it had been through to reach its current form.	*desirable*	scientific applications
AR-5-9	The provenance architecture should support the following operation: Tracing which source data was used to produce given result data and vice-versa.	*desirable*	scientific applications
AR-5-10	The provenance architecture should support the following operation: Providing the process information required for publishing an experiment's results.	*desirable*	scientific applications
AR-5-11	The provenance architecture should support the following operation: Deriving the higher-level processes that have been gone through to perform an experiment, so that they can be checked and re-used.	*desirable*	scientific applications
AR-5-12	The provenance architecture should support the following operation: Allowing experiments to be re-enacted to check that services and/or data has not changed in a way which affects the results.	*desirable*	scientific applications
AR-5-13	The provenance architecture should support the following operation: Determining the probable effectiveness of similar future experiments.	*desirable*	scientific applications
Traffic management application			
AR-6-1	The provenance architecture should support the certification of the simulation workflow against a reference workflow description.	*desirable*	TMA
AR-6-2	The provenance architecture should support the certification of input parameters against reference schemas for consistency.	*desirable*	TMA
DataMiningGrid			
AR-7-1	The provenance architecture should support: Recording data about processes and/or algorithms used in a workflow of processing raw data.	*desirable*	DMG
AR-7-2	The provenance architecture should support: Providing a trusted historical record of user access to confidential data	*nice to have*	DMG

2 Capability requirements

ID	*Textual description*	*Flags*	*Source*
Characteristics of provenance data			
TR-1-1-A-1	Recording of the following provenance information is required: Service invocation: Who accessed a particular service, when, with what input parameters (or a summary thereof) and on whose authority. 'Who' can refer to either a human or a service.	*essential*	OTM
TR-1-1-A-2	Recording of the following provenance information is required: Service response: Who a service sent data messages to, in response to which invocation, the content of the response (or a summary thereof). 'Who' can refer to either a human or a service.	*essential*	OTM
TR-1-1-A-3	Recording of the following provenance information would be useful: Information state: A summary of the information state in the service at the time a particular action is taken.	*nice to have*	OTM
TR-1-1-A-4	In addition to the logging of message based activities the provenance service also needs to capture "side effect" type actions (e.g. those which may not directly lead to a response message): • Carrying out an action in the real world • Recording a decision or fact	*essential*	OTM
TR-1-1-B-1	For the output of the TAU module the version information of the involved TAU code should be recorded by the provenance system.	*essential*	TENT/ SikMa
TR-1-1-B-2	For a given output of the TAU module the processed input files should be recorded by the provenance system.	*essential*	TENT/ SikMa
TR-1-1-B-3	For a given output of the Aeroelastic Module the processed input files should be recorded by the provenance system.	*essential*	TENT/ SikMa
TR-1-1-B-4	Rejection of job submission by the TENT framework to cluster batch systems should be recorded by the provenance system, so this event can be recognised by TENT and it can restart the workflow or the appropriate modules.	*essential*	TENT
TR-1-1-B-5	The provenance architecture shall provide a way to map TENT access rights to ensure that no misuse of provenance data will take place.	*essential*	TENT
TR-1-1-C-1	The following provenance information should be captured: Which process were executed together with their input and output.	*desirable*	eDiamond

ID	*Textual description*	*Flags*	*Source*
TR-1-1-C-2	The following provenance information should be captured: Who executed the process and when.	*desirable*	eDiamond
TR-1-1-C-3	The following provenance information should be captured:: Processing elapsed times.	*desirable*	eDiamond
TR-1-1-C-4	The following provenance information should be captured: Location and version information.	*desirable*	eDiamond
TR-1-1-C-5	The following provenance information should be captured: Request and response messages.	*desirable*	eDiamond
TR-1-1-C-6	The following provenance information should be captured: Other context information, which may include: • How was this service discovered? • policies established to invoke a service • information in SOAP message headers (security, reliability, transactionality etc.)	*desirable*	eDiamond
TR-1-1-D-1	The following provenance information is required to be recorded: The identity of the source of each provenance data entry.	*desirable*	HLSF
TR-1-1-D-2	The following provenance information is required to be recorded: The date and time that each provenance data entry is created.	*desirable*	HLSF
TR-1-1-D-3	The provenance architecture should support storing the following information with provenance data: Attributes for each provenance data entry that may reference other objects stored either outside or inside the provenance repository.	*desirable*	HLSF
TR-1-1-E-1	The following provenance information should be recorded: Version information (algorithms, databases).	*desirable*	myGrid
TR-1-1-E-2	The following provenance information should be recorded: Logs of what was done.	*desirable, critical*	myGrid
TR-1-1-E-3	The following provenance information should be recorded: Estimates of quality of service metrics such as execution time.	*desirable*	myGrid
TR-1-1-F-1	The following provenance information is required to be stored: Identity of process and version.	*desirable*	CombeChem

ID	*Textual description*	*Flags*	*Source*
TR-1-1-F-2	The following provenance information is required to be stored: Identity of operator.	*desirable*	CombeChem
TR-1-1-F-3	The following provenance information is required to be stored: Time.	*desirable*	CombeChem
TR-1-1-F-4	The recording of 'ambient conditions' of the experiments is required, like e.g. temperature. These parameters are known for the system during execution.	*desirable*	CombeChem
TR-1-1-F-5	The result and intermediate data of an experiment should be available and referenceable so that it can be linked to from papers and discovered for use in other experiments.	*desirable*	CombeChem
TR-1-1-G-1	The following provenance information is required to be stored: Calendrical information.	*desirable*	GENSS
TR-1-1-G-2	The following provenance information is required to be stored: Algorithmic information.	*desirable*	GENSS
TR-1-1-G-3	The following provenance information is required to be stored: Parameter information.	*desirable*	GENSS
TR-1-1-H-1	The following information should be recorded: Configuration parameters of simulation processes.	*desirable*	TMA
TR-1-1-H-2	The following information should be recorded: Input parameters of simulation processes.	*desirable*	TMA
TR-1-1-i-1	The following information should be recorded: Data about processes/algorithms used in a workflow of processing raw data.	*desirable*	DMG
TR-1-2	The system should provide a way for the user to annotate the provenance data.	*essential*	TENT, eDiamond, HLSF, myGrid, Combechem
Format of provenance data			
TR-2-1-A	"Format must be a non-proprietary format which can in principle be used with another tool (to be built if necessary) without violating IPR rules. An open standard would be best."	*essential, critical*	OTM

ID	*Textual description*	*Flags*	*Source*
TR-2-1-B	"The preferred API format for provenance data is the W3C Resource Description Framework (RDF). This is not mandatory, reports generated from the provenance data may be in other XML based formats and would have to conform to formats specified by external regulatory bodies. However RDF is preferred since this fits well with other standards in the Healthcare and Life Sciences arena, such as the Life Sciences Identifier (LSID)."	*desirable*	HLSF
TR-2-1-C	The export format of the provenance system should be XML defined by XML Schema.	*desirable*	eDiamond
TR-2-1-D	The export format of the provenance system should be XML-based.	*desirable*	GENSS
Storage and export of provenance data			
TR-3-1	The system should support the multiple storage of a provenance record, i.e. the system should provide a way to store copies of a provenance record in more than one repository.	*desirable*	eDiamond, HLSF
TR-3-2	The system should support the recording of different views on provenance information regarding to an event or an entity.	*essential*	OTM, eDiamond, HLSF
TR-3-3	The system should support the migration of provenance data among provenance repositories.	*essential*	OTM, eDiamond, HLSF
TR-3-4-A	On the fly recording of provenance data should be supported by the system.	*essential*	OTM, TENT, eDiamond, HLSF, DMG
TR-3-4-B	Batch recording of provenance data should be supported by the system.	*desirable*	eDiamond, HLSF, DMG
TR-3-5-A	The system should support the storage of recorded provenance data for a complete simulation session. Runtime for a simulation session is between 1 minute and 1 month, its typical value is a few days.	*essential*	TENT
TR-3-5-B	The system should support the storage of recorded provenance data for an indefinite period of time.	*desirable*	eDiamond, HLSF
TR-3-5-C	The system should support the storage of recorded provenance data for 3-4 years.	*desirable*	myGrid
TR-3-6	The system should be able to archive recorded provenance data.	*essential*	OTM, TENT, eDiamond, HLSF

ID	*Textual description*	*Flags*	*Source*
TR-3-7	The system should be able to export recorded provenance data for external usage.	*essential*	OTM, TENT, myGrid, Combechem, GENSS, eDiamond, HLSF, DMG
Utilisation of provenance data			
TR-4-1	It should be possible to query all of the data associated with a particular provenance entry, or return all of the provenance entries that have attributes matching a search criteria. *Note:* The term 'provenance entry' refers to the information written to a provenance service.	*desirable*	HLSF
TR-4-2	The architecture should support the dynamic processing of provenance data, i.e. recorded provenance data should be instantly queriable even if a recording session (recording of interrelated provenance records belonging to e.g. the same workflow) is still in progress.	*essential*	OTM, TENT, Combechem, myGrid, eDiamond, HLSF
TR-4-3	The provenance architecture should support the storage of results of analysis and reasoning operations performed on the provenance data by tools that are not part of the generic architecture (3rd party tools on the application layer).	*essential*	TENT, Combechem, eDiamond, HLSF
Operation of the provenance architecture			
TR-5-1	The provenance architecture should support for the maximum automation of the provenance recording mechanism.	*desirable*	eDiamond, HLSF, TMA, DMG
TR-5-2	Provenance handling should be policy-driven.	*desirable, critical (eDiamond)*	eDiamond, HLSF
TR-5-3	The provenance architecture should be deployable as an integrated part of a system, as a service within the same administrative domain as the client system and as a 3rd (external) party operated service, too.	*essential*	OTM, TENT, myGrid, Combechem, GENSS, eDiamond, HLSF, TMA, DMG, DILIGENT
TR-5-4	Client side components of the provenance architecture should not block workflow if provenance services are unavailable and client explicitly expresses their wish to turn off provenance recording.	*desirable*	myGrid
Interface			

ID	*Textual description*	*Flags*	*Source*
TR-6-1	The architecture should support a rich set of generic APIs that allow analysis and reasoning tools to be built upon.	*essential, critical (eDiamond)*	OTM, TENT, eDiamond, HLSF, TMA
TR-6-2	Human-computer interfaces presented by the system for analysis and reasoning should be designed to allow multilingual support.	*essential*	OTM
TR-6-3	Human-computer interfaces presented by the system for analysis and reasoning should be usable by a non computer scientist.	*desirable*	GENSS
TR-6-4-A	The provenance architecture should provide a programmatic interface for the administration of the system.	*essential, critical (eDiamond)*	TENT, eDiamond, HLSF
TR-6-4-B	The administrative interface of the provenance architecture should be able to be accessed and controlled through it's API. It has to be integratable into TENT or at least be accessible through the TENT system. Therefore some kind of user authentication may additionally be needed.	*essential*	TENT
TR-6-5-A	Provenance information should be trackable on human-computer interfaces presented by the system at set level (e.g. database table or spreadsheet).	*essential*	OTM, HLSF, GENSS, TMA, DMG, DILIGENT
TR-6-5-B	Provenance information should be trackable on human-computer interfaces presented by the system at individual data items (e.g. record in database or cell in spreadsheet).	*desirable*	eDiamond, HLSF, DILIGENT
TR-6-5-C	The granularity of provenance information displayed by the system on a human-computer interface should be configurable based on policies.	*desirable*	HLSF
TR-6-6-A	Provenance information displayed by the system on a HCI should be updatable on user request.	*essential*	OTM, eDiamond, GENSS
TR-6-6-B	HCIs presented by the provenance system for provenance monitoring should support continuous monitoring, i.e. the displayed information should be updated automatically on every change as soon as possible.	*essential*	OTM, GENSS, DMG, DILIGENT
TR-6-6-C	Provenance information displayed by the system on a HCI should be updated on each execution session of the monitored application.	*desirable*	TMA
TR-6-6-D	The update frequency of provenance information displayed by the system on a HCI should be configurable based on policies.	*desirable*	HLSF
System documentation			

ID	*Textual description*	*Flags*	*Source*
TR-7-1	There should exist different levels of system documentation, including the following: • a detailed API documentation for programmers who intend to integrate the provenance architecture into their systems, • a detailed description of the administrative interface of the system for system administrators, • a detailed description of other human-computer interfaces presented by the system e.g. for analysis and reasoning. Different audiences should be taken into account here including end-users as well, who want to use the provided tools as a stand-alone applications.	*essential*	TENT

3 Constraint requirements

ID	*Textual description*	*Flags*	*Source*
Performance constraints			
CR-1-1-A	Provenance recording should not impede a human entering data in real time.	*essential*	OTM
CR-1-1-B	Within TENT the execution overhead due to provenance recording has the upper constraint of not affecting the interaction with the system in a significant manner. In terms of the applications used in TENT workflows: due to typical execution times of e.g. the flow solver TAU, overhead has to be kept at minimum level.	*essential*	TENT
CR-1-1-C	Provenance recording should not slow down workflow execution by significant magnitude. (Significant not quantified.)	*desirable*	myGrid
CR-1-1-D	Provenance recording should increase end-to-end elapsed execution time by no more that 5%.	*desirable*	eDiamond
CR-1-2-A	Recorded provenance data should not exceed 20% of overall system record data.	*essential*	OTM
CR-1-2-B	There are the same constraints for storage overhead as for execution overhead (see CR-1-1-B), but less restricted.	*essential*	TENT
CR-1-2-C	Recorded provenance data should be less than 100 KBytes per patient image. (Patient images in eDiamond are usually around 32 MB.)	*desirable*	eDiamond
Quality of service attributes			
CR-2-1	Generated provenance data must not be lost.	*desirable*	eDiamond

ID	*Textual description*	*Flags*	*Source*
CR-2-2	The provenance architecture should guarantee reliable once-and-once-only delivery (no copies) as much as technically possible and up to the measure it depends on the architecture itself and not on operating conditions.	*desirable*	eDiamond
Legal and ethical issues			
CR-3-1	The provenance data should provide ability to ensure that appropriate regulations (such as those set by bodies like the Food and Drug Administration or the health and safety rules of a department) were adhered to.	*desirable*	CombeChem
CR-3-2	The provenance data should provide protection for intellectual property right issues, for example through the use of digital signatures and time stamping.	*desirable*	CombeChem
Security issues			
CR-4-1	The provenance architecture should have a configurable, fine-grained access control system over recorded provenance data.	*essential, critical (myGrid)*	OTM, myGrid, eDiamond, TMA, HLSF
CR-4-2	The provenance architecture should allow both automated and manual determination of access control rights on generated provenance data.	*essential*	OTM, eDiamond, DILIGENT
CR-4-3	Access rights to the provenance system must be consistent with access rights to the rest of the TENT system. The provenance system should provide a way to map access rights information of TENT into its security subsystem. Access rights are stored in TENT in an LDAP server.	*essential*	TENT
CR-4-4-A	The provenance architecture should be configurable in a way that assigns the following access rights to the given user groups: • User: Access to provenance data directly involved in the data manipulation process of the simulation (s)he has started and configured. • System designer: Access to secondary provenance data, which is the collection of all user provenance data and derivations from them for analysation and reasoning purposes. • System developer: Access to all kinds of provenance data. This especially includes data coming directly from the TENT core components. This data has to be visible only for this user group.	*essential*	TENT

ID	*Textual description*	*Flags*	*Source*
CR-4-4-B	The provenance architecture should be configurable in a way that assigns the following access rights to the given user groups: • Digital Library user: Access to the provenance data of his/her own initiated processes. • DILIGENT Administrator: General access. • Virtual Organisation Manager, Digital Library Manager, DILIGENT Resource Manager: Access to his/her own local provenance data.	*desirable*	DILIGENT
CR-4-5	The security infrastructure of the provenance architecture should have single sign-on.	*desirable*	eDiamond
CR-4-6	The security infrastructure of the provenance architecture should be the same as the one of the application – particularly for any end-user clients. *Note:* In the case of eDiamond this infrastructure is Globus GSI integrated with OGSA-DAI. However it is not a stable version at the moment, because the National Health Service are changing their infrastructure and moving towards PKI.	*desirable*	eDiamond
CR-4-7	The provenance architecture should provide a mechanism for recording adequate provenance data in an unmodifiable way to make results non-repudiable.	*desirable*	myGrid
Other constraints			
CR-5-1	The provenance architecture should have good application fit, meaning: meet the basic logging needs and have additional potential for more complex questions outlined in the scenario description.	*essential, critical*	OTM
CR-5-2	The provenance architecture should have the properties of cost efficiency and robustness versus an in-application hand-engineered logging system.	*essential, critical*	OTM
CR-5-3	The provenance system should be capable of handling huge amounts of provenance data coming in very frequently from the application itself. It should not create any bottlenecks disturbing the system.	*essential, critical*	TENT
CR-5-4	The provenance system should provide more and more detailed information about the different data and control flows taken place during workflow execution.	*essential, critical*	TENT
CR-5-5	On top of the API of the provenance system TENT must be able to access all its functions and provide them to the users through appropriate interfaces.	*essential, critical*	TENT
CR-5-6	The provenance architecture should be loosely coupled and independent from the application so that current system is unaffected. Provenance can depend on the application, but the application should not depend on the provenance.	*desirable, critical*	eDiamond

ID	*Textual description*	*Flags*	*Source*
CR-5-7	Tooling should be based on published APIs and not on hidden internal APIs	*desirable, critical*	eDiamond
CR-5-8	The provenance architecture should support transparent integration and operation with the DILIGENT infrastructure.	*desirable, critical*	DILIGENT
CR-5-9	Provenance mechanisms should be handled at grid middleware level and/or as a third party service.	*desirable, critical*	DILIGENT