Transatlantic TUmour MOdel Repositories

ID5.1.1
The TUMOR Architecture

Project Number: FP7--IST-247754
Deliverable id: ID5.1.1
Deliverable name: The TUMOR architecture
Submission Date: 10/05/2012

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	TUMOR
Project Full Name:	Transatlantic TUmour MOdel Repositories
Document id:	ID5.1.1
Document name:	The TUMOR Architecture
Document type (PU, INT, RE)	INT
Version:	0.7
Submission date:	
Editor: Organisation: Email:	Stelios Sfakianakis FORTH-ICS ssfak@ics.forth.gr

Document type PU = public, INT = internal, RE = restricted

ABSTRACT: This deliverable document describes the technical architecture of the TUMOR platform.

KEYWORD LIST: architecture, web services

MODIFICATION CONTROL			
Version	Date	Status	Author
0.3	10 May 2012	Draft	Stelios Sfakianakis
0.4	22 May 2012	Draft	Vangelis Sakkalis
0.5	22 May 2012	Draft	David Johnson
0.6	22 May 2012	Pre final	Stelios Sfakianakis
0.7	23 May 2012	Pre final	David Johnson

List of Contributors

All consortium

- Stelios Sfakianakis (FORTH)
- Vangelis Sakkalis (FORTH)
- Fay Misichroni (ICCS)
- David Johnson (UOXF.BL)
- Steve McKeever (UOXF.BL)
- Norbert Graf (USAAR)
- Thomas Taylor (INFOTECH Soft)

Table of Contents

Table of Figures

1 Executive Summary

The TUMOR project aims to build an integrated, interoperable transatlantic research environment offering the best available VPH models and tools for clinically oriented cancer modeling. It also aspires to serve as an international validation/ clinical translation platform for predictive, in silico oncology.

In order to achieve these ambitious goals, the TUMOR project has the following specific objectives:

- Design and develop a European clinically oriented, semantic layered cancer multi-scale digital model/data repository.
- Design and implement interoperable interfaces between this repository and the US semantic-layered digital model repository of CViT.org.
- Develop and provide specific tools and methods for the collection, curation, validation and customization of existing models and clinical data of EU projects and MGH (CViT) model repositories.
- Implement an integrated, interoperable transatlantic 'predictive oncology' workflow environment prototype, where the data can be accessed remotely in a secure way, the tools and the models can be transparently applied to these data, and the results can be visualized in the 'transatlantic' context.

To deliver the promised results the TUMOR's underlying technological platform needs to be defined and implemented. The purpose of this document is therefore to provide an *architectural description* of the IT infrastructure and components that comprise the TUMOR platform. In the introduction section we provide some background on the architecture definition process and the methodology that we have followed in documenting the TUMOR's architecture. Then, we describe the architecture from various views based on the selected methodology. Special emphasis is given on the functional requirements of the components and their interactions in order to implement the transatlantic scenarios.

2 Introduction

In recent years the importance of software architecture became evident. According to Bass et al [1] the software architecture of a system is, “the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.” The IEEE/ISO 42010 standard [9] defines an architecture as the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution. Software architectures are important because they represent the single abstraction for understanding the structure of a system and form the basis for a shared understanding of a system and all its stakeholders.

2.1 Architecture definition process

The architecture definition process is an important task in order to document a system’s functionality and quality attributes. But how does a system architect proceed in order to design the architecture? A proposed architecture definition process is shown in the figure below:

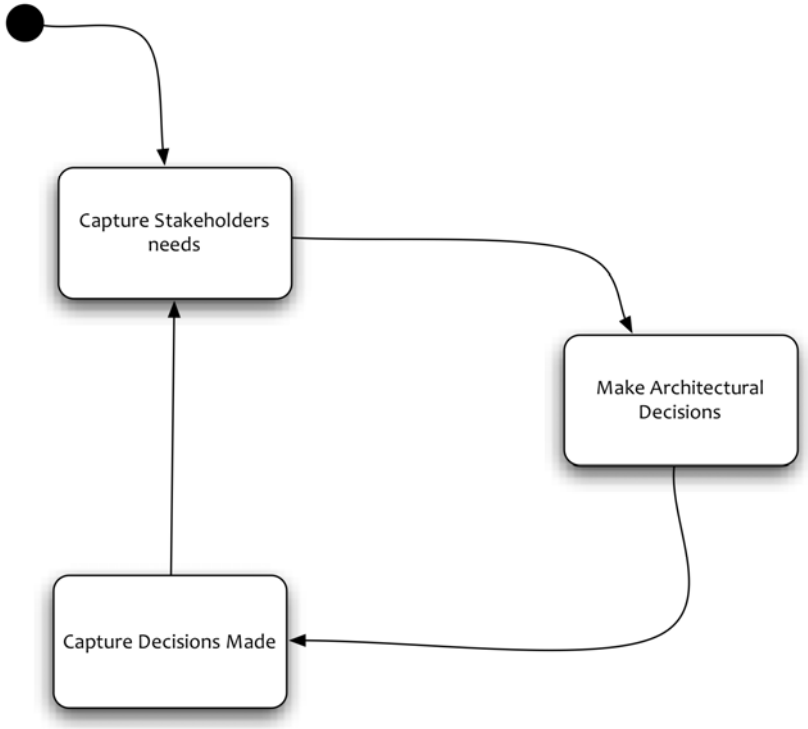

Figure 1 The architecture definition process.

According to this process, there are the following steps:

- *Capturing stakeholder needs*, that is, understanding what is important to stakeholders (possibly helping them reconcile conflicts such as functionality versus cost) and recording and agreeing on these needs.
- Making a series of *architectural design decisions* that result in a solution that meets these needs, assessing it against the stakeholder needs, and refining this solution until it is adequate.
- Capturing the architectural design decisions made in an Architectural Description.

These activities form the core of the architecture definition process and are normally performed iteratively. In order to formalize the architectural definition process we have

followed the IEEE/ISO/IEC 42010 standard that is very briefly described in the following subsection.

2.2 The IEEE 1471 standard

The IEEE 1471 standard, "Recommended practice for Architecture Description of Software-Intensive Systems," (http://www.iso-architecture.org/ieee-1471) addresses the activities of the creation, analysis, and sustainment of architectures of software-intensive systems, and the recording of such architectures in terms of architectural descriptions. This recommended practice has been also adopted since 2007 as an ISO standard, ISO/IEC 42010:2007 [9]. It illustrates the conceptual model of the architectural description, as defined in IEEE 1471.

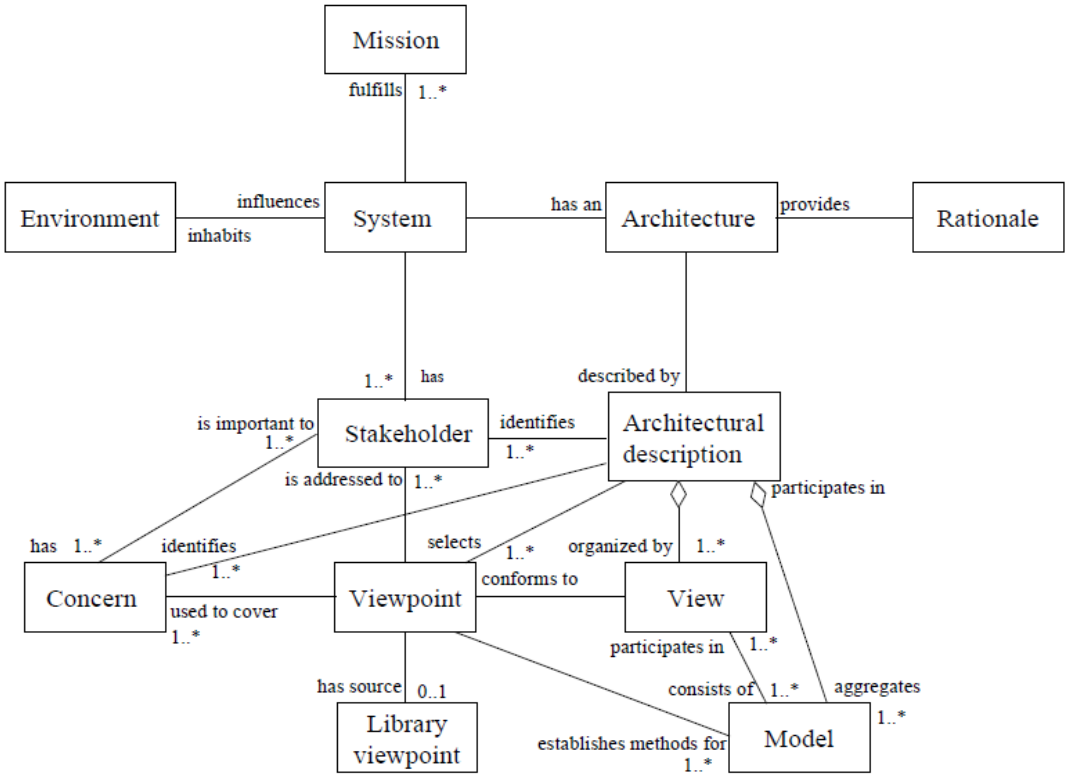

Figure 2 Conceptual model of architectural description from IEEE 1471.

According this conceptual model, a system has some architecture and this can be described in an architectural description. There is therefore a distinction between the architecture of a system, which is conceptual, from the description of this architecture, which is concrete. The architectural description (AD) is defined as, "a collection of products to document an architecture," and can be divided into one or several views. Each view covers one or more stakeholder concerns, such as usability, performance, maintainability, etc. On the other hand a View is defined as, "a representation of a whole system from the perspective of a related set of concerns." A view is created according to rules and conventions defined in a viewpoint, which is defined as, "a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis."

An architectural description selects one or more viewpoints for use and this choice depends on the concerns of the stakeholders that need to be addressed. A view may consist of one or more models and a model may participate in one or more views. Each such model is defined

according to the methods established in the corresponding viewpoint definition. The architectural description therefore aggregates the models, organized into views.

The IEEE 1471/ISO/IEC 42010:2007 standard defines a set of requirements for conforming architectural descriptions that can be summarized as:

- AD identification, version, and overview information.
- Identification of the system stakeholders and their concerns.
- Specification of each viewpoint that has been selected and the rationale for those selections.
- One or more architectural views.
- A record of all known inconsistencies among the AD's required constituents.
- A rationale for selection of the architecture.

It is evident from the discussion above that this standard is largely based on the definition of the most important viewpoints and the corresponding views but it does not provide any concrete definition of those. For this reason a number of different architectural frameworks supporting different views and viewpoints have been proposed, such as the 4+1 views model [13], the Reference Model of Open Distributed Processing (RM-ODP[1]), the Zachman framework[2], the Department of Defense Architecture Framework (DoDAF)[3], etc. In the next section of this document we present the TUMOR architecture from a number of selected views that are frequently referenced in the architectural descriptions of modern distributed systems.

[1] ITU-T Rec. X.901-X.904 / ISO/IEC 10746, http://www.rm-odp.net/

[2] http://www.zachman.com/about-the-zachman-framework

[3] http://dodcio.defense.gov/dodaf20.aspx

3 The TUMOR architecture

The TUMOR presents a number of integration, interoperability, and security related challenges. In designing the architecture we have followed the approach of views and viewpoints, which is standardized by IEEE/ISO [9]. First of all we need to identify the stakeholders and the important usage scenarios of the TUMOR platform. Next we describe the systems according to the various views of the selected methodology, putting more emphasis in the functional, logical, information, and physical views. Finally we describe some important *quality* attributes of the system under development that should be taken into account: the security, and the performance perspectives.

3.1 Stakeholders

The main stakeholders of the TUMOR platform are of course its users. As the main users of the system we can identify the VPH modelers and biomedical researchers that are focusing on the comprehension of the biological processes and interactions in nature and their simulations through computational models. The primary concerns of these users are:

- To share and reuse biological and physiological models.
- To share and reuse biomedical data.
- To plug (combine) models together to create larger, more comprehensive models without excessive demands for user input.
- To execute (run) the simulation models, trace their execution, and keep an archive of the results and the history of these executions.

In addition to the users who are the primary stakeholders of the system, there are also the system architects, engineers, developers, maintainers, etc, which comprise a different group of people that are responsible for the implementation, operation, and maintenance of the technical platform. This group of stakeholders is mostly concerned with the more technical aspects of the system construction and building and focuses more on the development and operational viewpoints of the architecture.

3.2 Functional Requirements and Constraints

The integrative TUMOR clinical workflow perspective described in the previous section can be translated to a set of functional requirements for the design of its architecture. These are the following:

- The users should be able to upload their cancer models and select/assign appropriate *metadata* in order to efficiently locate them afterwards and maintain their versioning history. Such metadata will consist of publishing information about the author, creation date, etc. They can also include access control information, which will permit or disallow their discovery by other users, licensing information to protect for intellectual property rights, and so on. It is actually the users who decide which model to share and what restrictions should be put on its use.
- The cancer models shared are accompanied by the necessary information that permits their actual execution. This information could include the code (in an executable or source format) and the necessary data to be used at the model's runtime.
- Data can also be used in more than one cancer model and therefore there is a need for supporting *data repository* in addition to the *model repository* described above. Proper metadata is provided with the uploaded data and provide hints about their purpose, type, lineage, etc.

- Different models with diverse biocomplexity levels and directions (bottom-up, top-down) are to be linked together and communicate in order to simulate cancer growth in a more *holistic* way. It is important to have an intuitive user interface to build these *hypermodels* and the paradigm of 'visual programming' where the linking of the models is done graphically is the one to follow.
- After the completion of a new workflow that connects two or more cancer models together, the users should be assisted to run the associated workflows by providing input parameters and data coming from the data repository. The execution should be transparent and leverage the metadata accompanied by each constituent cancer model in order to identify the required parameters, data, and execution environment.
- The 'transatlantic' scenarios are implemented through the workflow environment by retrieving the models both from the EU and the US model repositories.
- Intellectual properties of the users are protected while social networking facilities that are extremely popular these days are also accommodated.
- Privacy and security are built in. User authentication with 'single sign-on' ensures that the identity of the users is always available and proper access control mechanisms can be applied in every component of the platform.

An additional requirement is that according to the current legal and ethical regulations and restrictions, in both Europe and the US, even the exchange of retrospective data seems not feasible. Data needs to be stored locally in Europe or the US and not exchanged between partners. To overcome this problem tools and models need to be exchanged and shared to run simulations with locally provided data. This solution has significant implications for the type of infrastructure that TUMOR is developing.

3.3 Scenarios

Starting with the requirements and the functionality that the TUMOR platform aims to deliver, we can identify the use cases that the following picture exhibits:

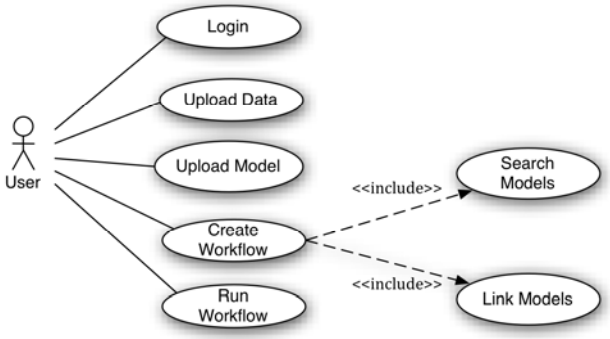

Figure 3 The main use cases.

The *login* use case represents the user authentication process that is prerequisite to any use of the platform. Here we are focusing on the domain users (modelers, researchers) that interact with the platform to implement, test, and validate their research hypotheses. Based on the supplied user credentials the user profile information can be retrieved and proper authorization decisions can later be based on. Uploading data and models is another set of important use cases where the users transfer and possibly publish and share their digital assets and artifacts. The model integration and linking is supported by workflow technologies but this requires two additional functionalities: 1) searching the model repositories based on some criteria or just browsing their contents, and 2) linking the selected models by connecting their output and input parameters. The constructed workflow can then be

executed (or 'enacted', as it is usually termed) and after its successful termination the results can be retrieved.

3.4 Logical View

Based on the scenarios and use cases described in the previous paragraph and the requirements of the project, we have identified the following software components and their responsibilities:

- The European Model and Data Repository: This is the 'main' model repository, located in Europe. In addition to storing the cancer models of the European users and their anonymized data, this repository also maintains the users profile information.
- The US Model Repository: This is the American model repository, located in the US, and operated by CViT. This is where users from the other side of the Atlantic store their models and data. It can be accessed from the European side but only the models can be transferred, due to the legal and ethical requirements.
- The Workflow Editing and Enactment environment: This is the Web-based application that allows the construction of simulation experiments through the linking of the available cancer models. In order to do so, the Workflow Environment accesses the EU and US model repositories and selectively retrieves their contents. It is hosted inside EU and therefore it has access to the data stored in the EU repository. Nevertheless since it is a web application, it has to make authorization decisions based on the users profile in order to restrict the data access mechanisms only to the European users. The execution of the workflows is taken care of a cluster of processing machines physically collocated with the workflow environment's server side.
- The Common Access Point (CAP, for short): This the main 'entrance' to platform. It is a web portal for interacting with the majority of the TUMOR services. Behind this portal there will be the EU Model and Data repositories and also the users' profile database.

A block diagram depicting these components and their interactions is shown in Figure 4. This diagram very briefly shows how the platform works from a top-level perspective.

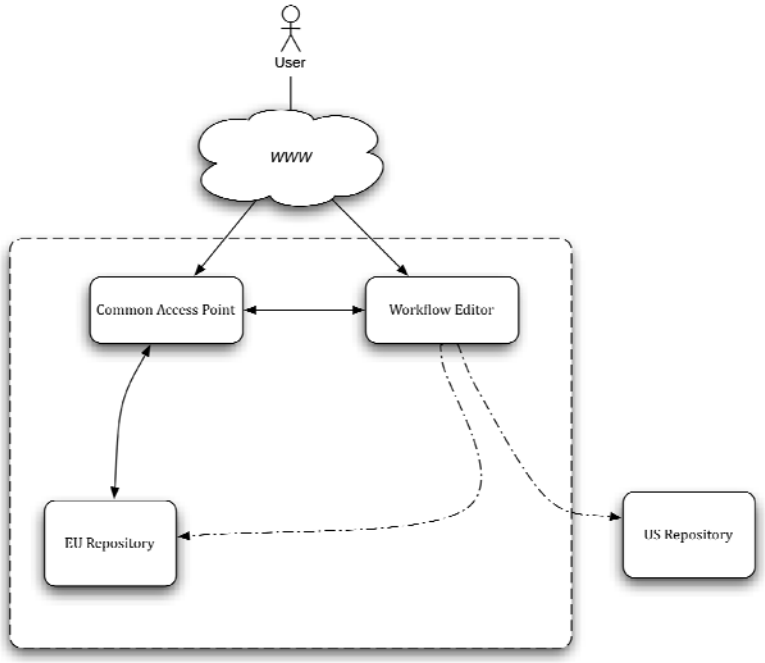

Figure 4 A logical diagram of the architecture showing its main components.

3.5 Information view

The information exchanged among the components consists of data and models. The models are described using TumorML, the new markup language (ML) for describing cancer models. The development of TumorML enables some of the key aims within the TUMOR project. Firstly, by annotating cancer models with appropriate document metadata, digital curation is facilitated in order to make publishing, search, and retrieval of cancer models easier for researchers and clinicians using the TUMOR digital repository. Second, markup will be used to describe abstract interfaces to published implementations allowing execution frameworks to run simulations using published models. Finally, TumorML markup facilitates the composition of compound models, regardless of scale and source, enabling multiscale models to be developed in a modular fashion, and models from the US CViT to be integrated with EU models in the TUMOR transatlantic scenarios.

Data sets on the other hand are managed as opaque entities (‘blobs’) at the architectural level that can be retrieved using the web services APIs of the repositories (see also the next section). Of course the repositories are expected to support a rich set of metadata to accompany the data, such as curation and provenance information (e.g. user who uploaded the data, date/time of the upload, how the data were processed, etc.).

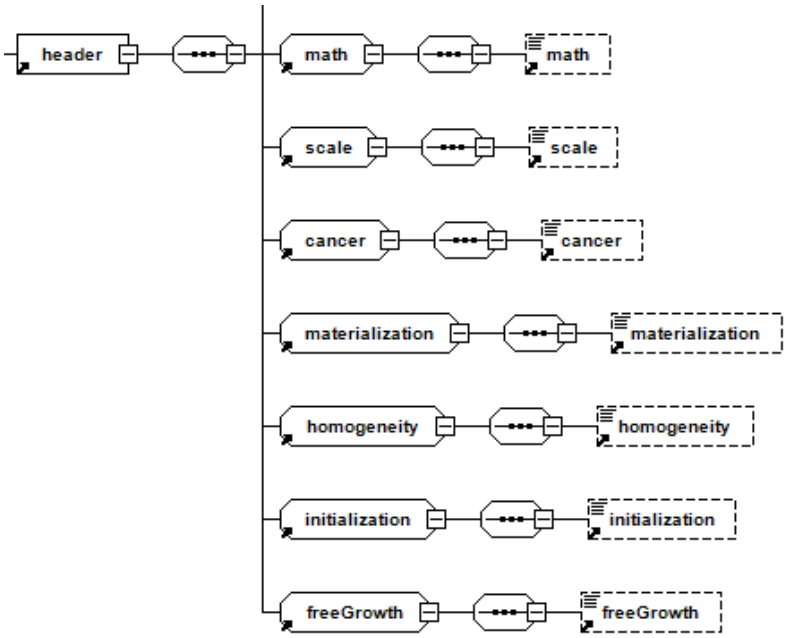

Figure 5 Part of the TumorML description.

3.6 Engineering view

The TUMOR environment is built as an online platform where its services and components are accessible over the World Wide Web. The architecture therefore is designed with 'service orientation' in mind, i.e. the software components expose a Web Service programmatic interface [10].

In essence, there is a programmatic interface for the "cross database" search and transmission of the models, so that no patient data are transmitted outside the European Union due to the lack of a legal framework and the implicated ethical and security issues. Therefore the main components that exhibit such an application programmatic interface (API) are the model repositories. Based on these APIs the Workflow Environment can browse, search, and retrieve cancer models and related data sets. There are two basic extensions to the baseline of SOAP/WSDL type of Web Services offered by the model repositories:

- Some data sets can be pretty large so the XML encoding imposed by the standard Web Services introduces a major performance tax. In these cases a more lightweight approach based on Representational State Transfer (REST) [11] is followed, i.e. the datasets are retrieved via simple HTTP(S) URLs.
- Semantic Web technologies [12] are employed in various places. The TumorML descriptions of the models are RDF compliant and therefore can be searched and retrieved using SPARQL, the query language of the Semantic Web. This approach permits the linking with more specialized domain ontologies for the model descriptions and also linking directly to the data files that are required for the model execution.

The Workflow environment is the 'epicentre' of the 'cross repository' browse, search, and retrieval of the models and data. There are various ways for this cross-repository interaction to take place:

- On the one hand the Workflow environment can always forward any browse or retrieve user request to the repositories. This has strong impact on the performance and the user perceived response time of the system since any filtering on the searchable information should be transmitted to the repositories and performed there.

It also imposes a fine-grained query functionality on the repository access web services APIs in order to support various search criteria that are important for the users. The benefit of this approach is that the search results are guaranteed to be always up-to-date and fully synchronized with the repositories contents.

- On the other hand, the workflow environment can maintain a local cache of the repositories' contents and perform all the queries on its local 'replicas'. The response time for this case is optimal at the expense of added complexity for keeping the local caches synchronized with the master copies at the repositories and the possibility of showing stale information.
- *The hybrid solution, and the one to follow, is for the workflow environment to make coarse-grained requests to the repositories, at the time the user logs in, to retrieve the user accessible models' information. Subsequent filtering and searching can then be done locally in a more expressive and detailed query language as guided by the user interface of the workflow environment.*

3.7 Security View

Security is a multifaceted task that usually involves requirements:

- **Authentication**. As the first process, authentication provides a way of identifying a user, typically by having the user enter a valid user name and valid password before access is granted.
- **Authorization**. Following authentication, a user must gain authorization for doing certain tasks. After logging into a system, for instance, the user may try to issue commands. The authorization process determines whether the user has the authority to issue such commands. Simply put, authorization is the process of enforcing policies: determining what types or qualities of activities, resources, or services a user is permitted. Usually, authorization occurs within the context of authentication. Once you have authenticated a user, they may be authorized for different types of access or activity.
- **Accounting** measures the resources a user consumes during access. This can include the amount of system time or the amount of data a user has sent and/or received during a session. Accounting is carried out by logging of session statistics and usage information and is used for authorization control, billing, trend analysis, resource utilization, and capacity planning activities.
- **Communication related security aspects** such as integrity of the messages exchanged, trust of the communicating peers, etc.

For the authentication and authorization aspects, there is the need for authenticating the users with the minimal possible distraction and also supporting authorization and access control. The workflow environment is a separate web application that stores neither user login information, nor the models and the accompanied data. So there is a need for *Single Sign On*, so that the users are not required to signup twice or to provide the same credentials twice when they access the CAP/EU Repository and the Workflow Environment. Additionally the users should be allowed to make secure and authenticated requests to the model repositories through the workflow environment. To address both of these concerns, TUMOR uses the OAuth 2.0 (Open Authorization, version 2.0) protocol that is also supported by Google, Microsoft, and Facebook in their web applications. Using OAuth the Workflow Environment can access the model repositories on the users' behalf without knowing their passwords or other authenticating information.

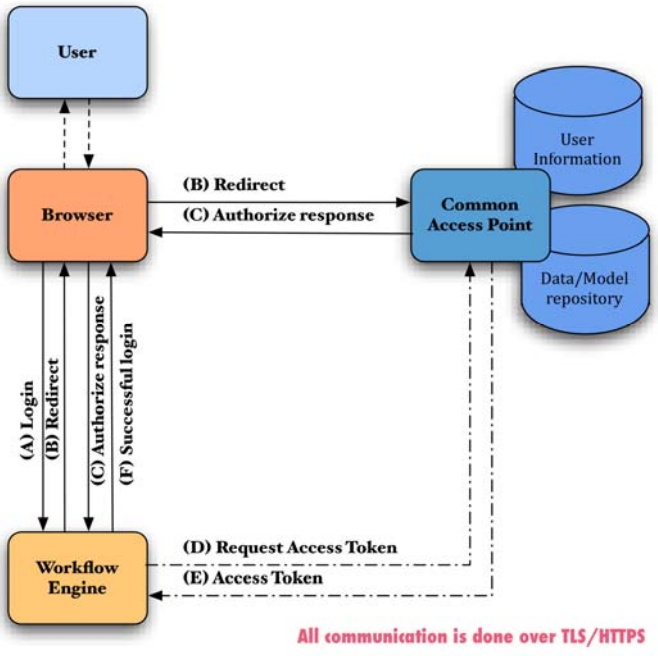

Figure 6 The flow of information and control in the oAuth based single sign on.

The flow of the information among the different components for logging into the Workflow environment is shown in Figure 6 with the various interactions (labeled as (A)-(F)) and it's as follows:

- A user through their browser visits the WF web site (A).
- The Workflow environment web site finds out that this is a new user that needs to be authenticated (logged-in) and sends a Redirect (B) to the EU Repo web site.
- The browser follows the redirect and therefore the user is presented with the login form of the EU Repo at the Common Access Point web site.
- The user fills in the username password and submits the form.
- The Common Access Point validates the credentials and redirects the user/browser back (C) to the 'Redirection URI' that was supplied in interaction A. This redirection carries an 'authorization code'.
- The Workflow environment takes the authorization code and uses it to make a 'behind the scenes' (i.e. without the user noticing it) request (D) to the Common Access Point to validate it.
- The Common Access Point responds with an 'access token' (E) that the WF stores in the user's session and can be used in subsequent communication with the CAP.
- The Workflow environment presents the welcome screen to the user (F).

All the communication is done over HTTPS, which supports the integrity and non-repudiation of the transmitted messages since it is based on the Public Key Infrastructure (PKI) and digital certificates.

Authorization is performed by the repositories, since these are the components that store the user profiles and the user access rules. All the interactions with the repositories carry (indirectly, though the OAuth access token) the user identity so that the repositories can control the user access to the data and models that the user is allowed to retrieve. Finally, accounting and auditing are also performed by the use of logging facilities but this is done also in a distributed sense, both in the Workflow environment and in the model repositories.

3.8 Physical View

The deployment architecture of TUMOR is shown in Figure 7. The EU Repository and its Access Point are hosted in the ICCS premises. The Workflow Editor and Engine alongside with its supporting execution infrastructure will be deployed in FORTH. The US Repository is hosted on the other side of the Atlantic, in the CViT.org infrastructure.

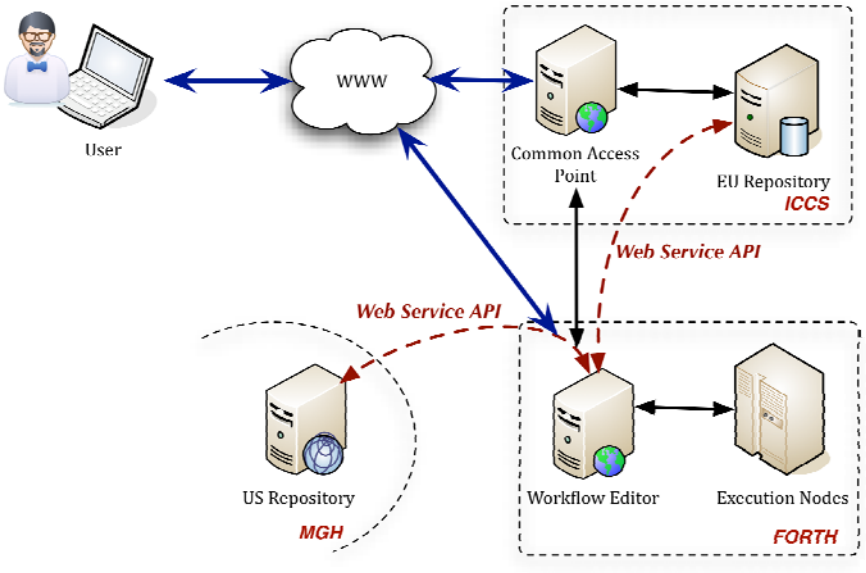

Figure 7 The deployment of the TUMOR platform.

4 References

[1] L. Bass et al, *Software Architecture in Practice*. Addison-Wesley, 1997.

[2] T.S. Deisboeck and G.S. Stamatakos (Eds), *Multiscale Cancer Modelling*. CRC Press, 2010.

[3] D.D. Dionysiou et al, “A four-dimensional simulation model of tumour response to radiotherapy in vivo: parametric validation considering radiosensitivity, genetic profile and fractionation,” in *Journal of Theoretical Biolology*, 230(1):1–20. September 2004.

[4] G.S. Stamatakos et al, “A four-dimensional computer simulation model of the in vivo response to radiotherapy of glioblastoma multiforme: studies on the effect of clonogenic cell density,” in *British Journal of Radiology*, 79(941):389–400. May 2006.

[5] R.L. Ho, “Biosimulation software is changing research,” In *Biotechnology Annual Review*, 10:297-302. 2004.

[6] T.S. Deisboeck et al, “In silico cancer modeling: is it ready for prime time?” in *Nature Clinical Practtice Oncology.* 6(1):34-42. January 2009.

[7] A. Belloum et al, “Scientific workflows,” in *Journal of Scientific Programming special issue on workflows to support large-scale science*, 14(3-4):171. 2006.

[8] G. Fox and D. Gannon, “Special Issue: Workflow in Grid Systems: Editorials,” in *Concurrency and Computation: Practice & Experience*, 18(10):1009–1019. 2006.

[9] ISO/IEC 42010:2007 *Systems and software engineering - Recommended practice for architectural description of software-intensive systems*. October 9, 2000.

[10] F. Curbera et al, “Unraveling the Web Services Web: an Introduction to SOAP, WSDL, and UDDI,” in *IEEE Internet Computing*, 6(2):86–93. 2002.

[11] R.T. Fielding and R.N. Taylor, “Principled Design of the Modern Web Architecture,” in *ACM Transactions on Internet Technology (TOIT)* 2(2):115–150. 2002.

[12] N. Shadbolt et al, “The Semantic Web Revisited,” in *IEEE Intelligent Systems*, 21(3):96–101. 2006.

[13] P. Kruchten. “Architectural Blueprints - The 4+1 View Model of Software Architecture,” in *IEEE Software*, 12(6):42–50. November 1995.

[14] D. Johnson et al., “TumorML: Concept and Requirements of an In Silico Cancer Modelling Markup Language,” in *Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE EMBS. September, 2011.