Securing Real-Time On-Line Interactive Applications in edutain@grid

J. Ferris[1], M. Surridge[1], and F. Glinka[2]

[1] IT Innovation Centre, University of Southampton, UK
{jf,ms}@it-innovation.soton.ac.uk
[2] Institute of Computer Science, University of Münster, Germany
glinkaf@uni-muenster.de

Abstract. This paper presents the analysis, design and implementation of security facilities within the edutain@grid infrastructure, to support secure hosting of Real-Time On-line Interactive Applications (ROIA). The edutain@grid project aims to develop a novel, sophisticated and service-oriented Grid infrastructure which provides a generic, scalable, reliable and secure service infrastructure for ROIA. The class of applications that comprise ROIA have requirements that present obvious challenges to security infrastructure design and implementation. In particular, the requirement to maintain real-time interactivity, particularly within the virtual world subclass of ROIA, precludes a heavyweight solution for securing ROIA. The edutain@grid project is extending 'business Grid' infrastructure that supports Service Level Agreements (SLA) for non-real-time data storage and processing. This infrastructure is based on GRIA and uses Transport Layer Security (TLS) and Web Services Security to secure web service interactions for the provision of data storage and processing. The edutain@grid project is also developing the Real-Time Framework (RTF), which provides communication and parallelisation functionality and API for application developers to create distributed ROIA that can be deployed to and hosted by instances of edutain@grid. The requirements, analysis, design and implementation of security facilities within RTF and the upper business layers of edutain@grid are presented below. We argue that the security facilities provide a suitable compromise between security and performance that will be attractive to the edutain@grid actors and stake holders.

Keywords: Real-time, Grid, Trust and Security, Business.

1 Introduction

Emerging Grid technologies [1] have the capability to substantially enhance on-line games and similar applications. Just as the World Wide Web enables people to share content over standard, open protocols, the Grid enables people and organizations to share applications, data and computing power over the Internet in order to collaborate, tackle large problems and lower the cost of computing.

The edutain@grid project [2, 3] aims to develop a novel, sophisticated and service-oriented Grid infrastructure which provides a generic, scalable, reliable and secure

E. César et al. (Eds.): Euro-Par 2008 Workshops, LNCS 5415, pp. 371–381, 2009.

service infrastructure for a new class of 'killer' applications of the Grid: Real-Time, On-Line, Interactive Applications (ROIA). ROIA include a broad sub-class of commercially important applications based on virtual environments, including massively multiplayer on-line gaming applications (MMOG), and interactive training and other e-learning applications. The edutain@grid project is aiming to provide an infrastructure to make such applications easier to develop, more economic to deploy and operate, and more capable of meeting the Quality of Experience expected and demanded by end-users.

Grid middleware systems such as Globus [4], gLite [5] and UNICORE [6] enable high-throughput applications by sharing computational resources for processing and data storage to meet the needs of individual and institutional users. ROIA such as multiplayer on-line computer games are soft real-time systems with very high interactivity between users. Large numbers of users may participate in a single ROIA instance, and are typically able to join or leave at any time. Thus ROIA typically have extremely dynamic distributed workloads, making it difficult to host them efficiently. Initiatives such as Butterfly Grid [7] and Bigworld [8] have applied Grid computing to on-line gaming with some success, enabling 'scalable' or 'elastic' terms for hosting such games. However, these 'scalable' hosting services are only as scalable as the hoster supporting them, and typically do not guarantee how far this will be. The edutain@grid project addresses these challenges using 'business Grid' developments such as GRIA [9, 10], but extending them to support scalable, multi-hosted ROIA applications, allowing scaling beyond the limits of any one hoster.

The focus for this paper is the support provided by edutain@grid for secure, real-time communications. In Section 2, we present an overview of the business actors supported by edutain@grid, between which secure, real-time communications must be established. Section 3 describes the edutain@grid architecture and discusses the requirements for secure real-time communication. Section 4 describes how this is addressed using business layer services to exchange keys and set up the required real-time communications. Section 5 describes the security features incorporated into the Real-Time Framework [11]. Finally, Section 6 presents the conclusions of the work described, and discusses possible directions for future work.

2 Business Actors and Value Chains in edutain@grid

To ensure business models for Grid-based ROIA will be economically viable, it was necessary that the edutain@grid infrastructure be generic enough to support a wide range of value chains needed to address different market conditions in these sectors. The analysis revealed an extensive hierarchy of business roles (actors). These include *providers* who host services by which the ROIA is delivered, *consumers* who access the ROIA by connecting to these services and *facilitators* who play other business roles in the creation of ROIA application software, its distribution to providers and consumers, and the operation of ROIA instances. Three important sub-classes of ROIA providers were also identified that must be supported by the project:

- *Hosters*: organisations that host (usually computationally intensive) processes that support a ROIA virtual environment including interactions of users with this environment and with each other.

- *Co-hosters*: other hosters participating in the same ROIA instance; if more than one hoster is involved in a single ROIA, each will regard the others as 'co-hosters'.
- *Coordinator*: an organisation that makes a ROIA accessible to consumers, and coordinates one or more hosters to deliver the required ROIA processes.

Today, on-line game hosters exist, but there are no 'co-hosters' or 'coordinators' because there is only one hoster per game instance. The edutain@grid project removes this limitation, enabling new business models for managing risks of ROIA hosting and delivery, and providing unlimited scalability for ROIA provision. The corresponding value chains are established via bipartite agreements, which also provide the basis for security, following the pattern used in the NextGRID project [12, 13] and with GRIA in the SIMDAT project [14]. The edutain@grid project allows for a wide range of topologies, of which a typical example is shown in Fig. 1:

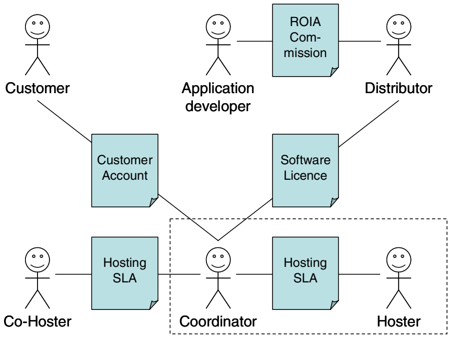

Fig. 1. A typical edutain@grid value chain

In this example, the ROIA application is commissioned by a distributor from an application developer, and operated by a coordinator who also hosts ROIA processes, as shown by the dotted line indicating that the coordinator and one of the hosters are actually the same organisation. Other co-hosters can then be brought in to handle peaks in demand, or if the ROIA becomes so popular that one organisation cannot host it all any more. For a more in-depth analysis of edutain@grid value chains and their implications for SLA terms, see [15]. The challenge for developers of ROIA is to secure real-time communications between the actors in such value chains, bearing in mind that customers will not be certified by a trusted certification authority in advance, and both customers and co-hosters can join or leave the ROIA at any time.

3 Architecture and Security Requirements

The first implementation of the edutain@grid framework was produced in early 2008, and is now being extended to incorporate the security features described in this paper. The prototype focuses on the core edutain@grid actors: the coordinator, the hoster (or co-hoster), and the customer. The framework is based on a Service Oriented Architecture, organised in four layers, as shown in Fig. 2:

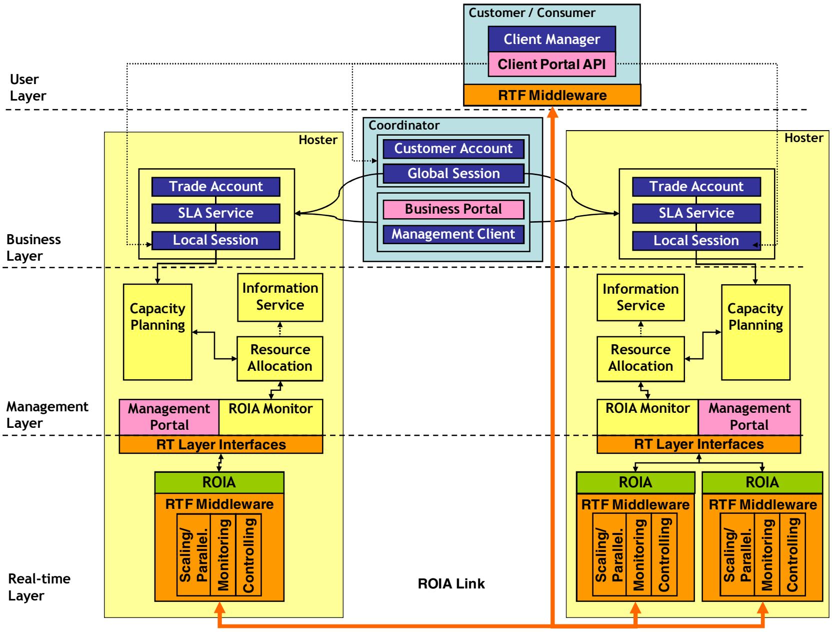

Fig. 2. Prototype edutain@grid architecture

The real-time layer provides a framework [11, 16] for ROIA developers to create scalable applications capable of running across multiple sites. The management layer handles the allocation and management of resources (and ROIA processes) by hosters. The business layer deals with the creation and enforcement of hosting SLAs and customer agreements, including dynamic updating of security policies to ensure ROIA can only be accessed under a valid agreement. There is also a client layer which provides programming interfaces to use services from the other three layers.

The focus for this paper is the establishment and use of secure ROIA links supporting real-time communications; between ROIA clients used by customers and ROIA processes (services) operated by hosters, and also between different ROIA processes even when located at different co-hosters. For more details of other aspects of the edutain@grid architecture and implementation, see [17]. The main security requirements identified are as follows:

- the underlying communication protocol should be based on UDP, since the need for real-time interactivity precludes the use of TCP packet-level handshaking over WAN communication links;
- communications should provide integrity protection at the datagram level, and support authentication of the sender's identity and other attributes, which may be used for authorisation decisions within the ROIA application; and
- communications may provide confidentiality of datagram content: this may not be necessary in on-line games where performance is more important than confidentiality, but is likely to be needed in many e-learning applications.

These requirements have to be met in the context of the business relationships in the ROIA delivery value chain. The business layer of edutain@grid handles this using web services based on management services from the GRIA 5.2 middleware [10] plus some custom ROIA services developed by the edutain@grid project and the real-time layer implementation of secure communications depends on the business layer to handle trust decisions, manage roots of trust and to 'bootstrap' security in RTF.

4 Business Layer Services

The business layer in edutain@grid is responsible for ensuring that only authorised actors can manage a ROIA instance and connect to its ROIA processes. The first step is to establish business agreements and form ROIA value chains between customers and hosters via a coordinator. There are two types of business agreements supported in the current implementation:

- Service Level Agreements (SLA) for hosting ROIA services, established between a hoster and a coordinator;
- Customer Accounts established between a customer and a coordinator to which the customer's ROIA access can be billed.

In each case, the service provider publishes the agreement terms it intends to make available. The service consumer then requests an agreement on those terms, and the service provider can then decide whether to accept or reject the agreement. At present, the decision whether to approve or reject a request is done manually via internal interfaces at the hoster, but in principle, one could automate this by using a business credit checking agency, etc. Normally, the coordinator will form an agreement with at least one hoster and then offer terms to customers, who can open accounts using a similar procedure. The edutain@grid implementation allows customers to open and manage accounts without using a ROIA (e.g. gaming software) client – they could do this using a web browser to access the coordinator's account service, for example.

These agreements provide the roots of trust for securing subsequent actions to launch and access the ROIA itself. Fig. 3 shows the main business layer services and the workflows used to do this. The coordinator first creates a ROIA Global Session resource at its own Global Session Service to hold information (including security policies) associated with each instance of the offered ROIA. The coordinator can then provision the ROIA by creating a ROIA Local Session using a hoster's Local Session Service, which handles the contribution to the ROIA by that hoster. Before creating the requested ROIA Local Session, the hoster checks the credentials of the coordinator with the hoster's own SLA Service. This ensures that only coordinators who have agreed an SLA can use the hoster's ROIA hosting facilities. The coordinator may use several ROIA Local Sessions at different hosters, depending on the scale of resources needed to support the ROIA instance when it starts up. The coordinator then sends them a token validation policy to verify tokens issued for the ROIA Global Session and by each other. All the Local Session Services then signal the edutain@grid management layer to start the ROIA processes, issue them with identification keys, set policies so they can recognise other ROIA processes in the same ROIA (Global Session), and start monitoring the ROIA.

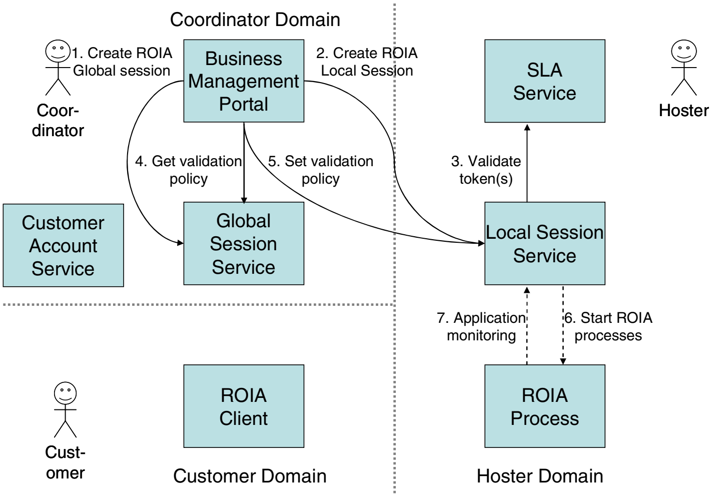

Fig. 3. ROIA Provisioning by the Coordinator and Hoster(s)

At this point, ROIA processes will be running at (potentially multiple) hoster sites, so customers can access the ROIA. The customer signs on to its account via the ROIA client application (Step 1 of Fig 4), obtaining an X509 certificate signed by the coordinator identifying him as a customer. Customers can set up additional pseudonyms at the Customer Account Service, so the certificate need not reveal their true identity, which is helpful in on-line gaming applications where customers rarely use their real names. The ROIA client can then use the certificate to get other short-lived tokens expressing its role in the Global Session (e.g. teacher, student, superhero, etc), and the address of a hoster it should connect to. The ROIA client then contacts the hoster's Local Session Service to get details of the ROIA Process it should connect to. The certificate and other tokens issued by the coordinator match the policies previously sent to the hoster, so the hoster can immediately decide if a connection request should be accepted. These tokens are also short-lived, so it isn't necessary to revoke them. Therefore, the hoster doesn't have to make call-backs to revocation lists, though it does mean the ROIA client has to renew tokens at the coordinator before they expire. By pushing validation policies to hosters in advance, and issuing short-lived tokens that don't need to be revoked, it is possible to use tokens for real-time connections without the usual overheads associated with X509 certification.

The ROIA process then reports the new connection via the hoster's management layer to the Local Session Service. This informs the Global Session Service (if necessary), and reports the usage to the SLA Service, so it can check that the total usage at this hoster remains within the terms of the SLA with the coordinator. At some point, the customer will disconnect, when he stops using the ROIA or when his actions in the ROIA means his connection should move to a different hoster. The Local Session Service is again informed, allowing continued management of the ROIA within the terms of the coordinator's SLA with hosters. It is also possible that the hoster's management layer will at some point predict that the load on its services will exceed the

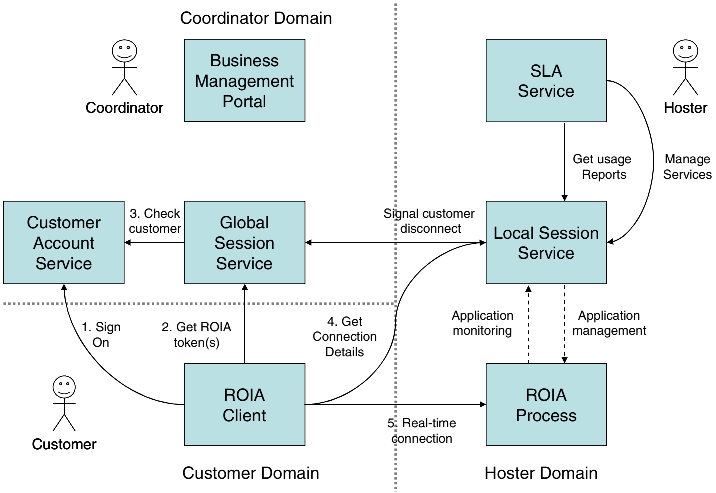

Fig. 4. Accessing The ROIA

SLA limits. That too can be reported via the Local Session Service to the coordinator, allowing 'global' management action to be taken. Actions that could be taken are the transfer of load to another hoster, or to reduce load by disabling access for some customers (if permitted under the Customer Account agreement), before the SLA is breached.

5 Real-Time Framework

The real-time layer in edutain@grid comprises Real-Time Framework (RTF) [11] that supports protocols for using tokens issued by the business layer to control application-level access. RTF is implemented as a C++-library providing parallelization and communication API for application developers to easily develop distributed ROIA.

For securing the communication between the client and ROIA Process, as well as between ROIA Processes, we chose DTLS [18] as it supports encrypted unreliable communication (unlike TLS [19]), can be easily integrated in RTF and needs no operating system support (unlike IPSec). DTLS can also handle the authentication of the communication endpoints during the connection setup.

Fig. 5 shows the main RTF components and the workflows used to establish secured connections and access control using keys and tokens from the business layer:

- the Authentication & Authorization Service (AAS) stores certificates and other security tokens used for the client authentication and authorization;
- the Communication and Distribution Services (CDS) establish RTF connections and manage the parallelization and communication of the distributed ROIA;
- the ROIA is implemented by the application developer on top of RTF CDS; and

- the Policy Decision Point (PDP) intercepts client messages before they are delivered to the ROIA – this is an optional component that can be provided by the application developer or the coordinator.

When the ROIA Process is started, the Local Session Service passes a process-unique identity certificate along with trusted keys and policies for authentication and authorization within the ROIA instance. The ROIA Process stores these for later use in its AAS. A trust chain is thus established from the Coordinator through the ROIA Local Session down to the ROIA Processes.

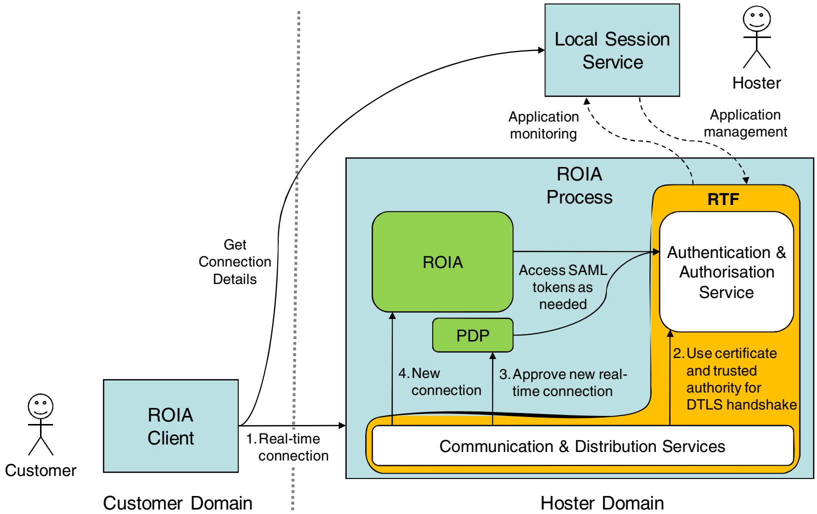

Fig. 5. Creating Real-Time Connections

If a ROIA Client now connects to the ROIA Process (Step 1, Fig. 5) using the connection details that it received from the ROIA Local Session Service, then the RTF's CDS uses the stored certificate and trusted authority for the DTLS handshake and authentication (Step 2). Thus the client and ROIA process can authenticate each other as belonging to the same ROIA instance. Once the handshake is completed, the ROIA Link is associated with the customer's distinguished name (DN) which is based on the login or pseudonym he set up with the Coordinator. If the PDP approves the new connection (Step 3), the ROIA is notified about the newly joined client (Step 4).

Clients communicate with the ROIA Process by sending messages of various, application-dependent types, e.g. to move their avatar, pick up objects, etc. Each message type is identified by a unique integer attribute. An application-specific PDP can be plugged into the RTF during the ROIA Process start-up, and used to determine whether messages should be forwarded to the ROIA. Decisions may be based on the client identity alone, but may also use other user attributes passed with the identity certificate to the ROIA Process in the form of SAML tokens issued by the business layer. These are stored in the AAS for use in subsequent decisions. For example, if a client can get a SAML token from the business layer saying it is a teacher in an e-learning ROIA, that client would be able to send administrative message types that change the structure of an ongoing e-learning lesson, while other clients could not.

If a ROIA Client wants to gain additional rights within a ROIA, it requests SAML tokens at the Coordinator (Step 1, Fig. 6). These SAML tokens are then pushed by the ROIA Client towards the ROIA Process (Step 2). The authentication and authorization service intercepts these tokens, verifies them against its trusted authority and stores them (Step 3) for later use by the PDP. This way, the client can add rights as needed during run-time by providing additional SAML tokens.

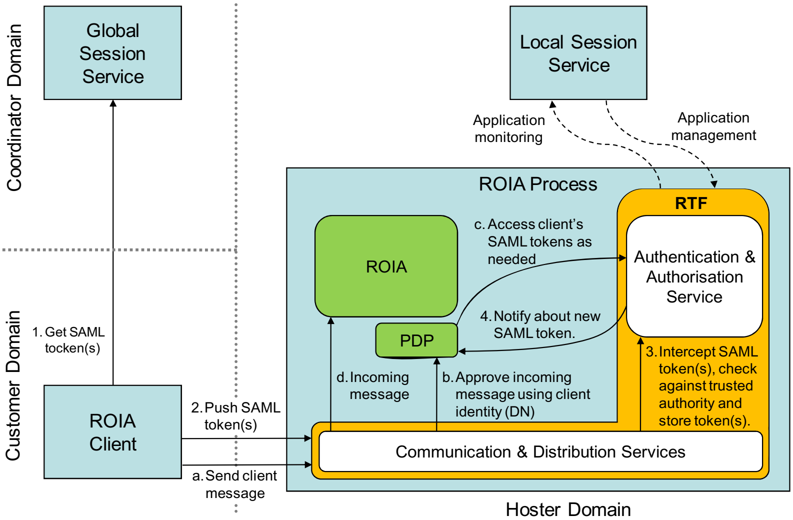
Coordinator Domain
Customer Domain
Global Session Service
Local Session Service
Application monitoring
Application management
ROIA Process
RTF
ROIA
c. Access client's SAML tokens as needed
Authentication & Authorisation Service
4. Notify about new SAML token.
PDP
1. Get SAML tocken(s)
d. Incoming message
b. Approve incoming message using client identity (DN)
3. Intercept SAML token(s), check against trusted authority and store token(s).
ROIA Client
2. Push SAML token(s)
a. Send client message
Communication & Distribution Services
Hoster Domain

Fig. 6. Application Level Access Control

6 Conclusions and Future Work

The edutain@grid project has developed a framework for grid-based multi-hosting of Real-Time Online Interactive Applications (ROIA), including scalable online gaming and virtual e-learning environments. In achieving this goal, the project has had to address the challenge of securing real-time communications in a way that is consistent with business relationships, by devising a unified architecture for business and real-time communication security. Our approach is based on three main developments:

- the use of Web Services based on GRIA to support the creation of business-level agreements, secured using a dynamic policy implementation in conjunction with Web Service message-level security specifications including WS-Security, WS-Trust, X.509 and SAML [20];
- the use of DTLS for secure real-time data-gram transport within RTF, using trusted keys provided and distributed between actors by the business-level Web Services;

- the use of a simple username-password sign-on procedure allowing users (most of whom are not experts in information security) to obtain and use X509 and SAML credentials without having to manage keys and tokens.

Evaluation studies will now be carried out: testing the performance of secure real-time protocols and the ability of policy/token management protocols to keep up, and evaluating the overall edutain@grid business processes, security, usability, quality of service, etc.

Future work is expected to include the addition of support for configurable RTF security properties (e.g. ciphers, key-lengths, whether to use encryption as well as authentication), so enabling a range of different trade-offs between security and performance requirements. We also plan to investigate options for secure multi-cast RTF communications by extending the key and policy management used to bootstrap RTF security in the business layer.

Acknowledgments. The work described in this paper is supported by the European Union through EC IST Project 034601 ‘edutain@grid’.

References

1. Foster, I., Kesselman, C. (eds.): The Grid2: Blueprint for a New Computing Infrastructure, 2nd edn. Morgan Kaufmann Publishers Inc., Elsevier, Boston (2004)
2. Fahringer, T., Anthes, C., Arragon, A., Lipaj, A., Müller-Iden, J., Rawlings, C., Prodan, R., Surridge, M.: The Edutain@Grid Project. In: Veit, D.J., Altmann, J. (eds.) GECON 2007. LNCS, vol. 4685, pp. 182–187. Springer, Heidelberg (2007)
3. See the edutain@grid, `http://www.edutaingrid.eu/index.php`
4. Foster, I., Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit. International Journal Supercomputer Applications 11(2), 115–128 (1997)
5. Czajkowski, K., Ferguson, D.F., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S., Vambenepe, W.: The WS-Resource Framework (March 2004)
6. Breuer, D., et al.: Scientific Computing with UNICORE. In: Wolf, D., Münster, G., Kremer, M. (eds.) NIC Symposium 2004. NIC Series, vol. 20, pp. 429–440 (2003)
7. IDC Case Study, Butterfly.net: Powering Next-Generation Gaming with On-Demand Computing, `http://www.ibm.com/grid/pdf/butterfly.pdf`
8. Big World Technology, http://www.bigworldtech.com/index/index_en.php
9. Surridge, M., Taylor, S., De Roure, D., Zaluska, E.: Experiences with GRIA – Industrial Applications on a Web Services Grid. In: e-Science 2005, pp. 98–105. IEEE Press, Los Alamitos (2005)
10. See the GRIA, `http://www.gria.org`
11. Glinka, F., Ploss, A., Gorlatch, S., Müller-Iden, J.: High-Level Development of Multi-Server Online Games. International Journal of Computer Game Technology (August 2008)
12. Snelling, D., Fisher, M., Basermann, A.: NextGRID Vision and Architecture White Paper. Updated periodically: at the time of writing the published version dates from (July 30, 2006)
13. Mitchell, B., Mckee, P.: SLAs A Key Commercial Tool. In: Cunningham, P., Cunningham, M. (eds.) Innovation and the Knowledge Economy: Issues, Applications, Case Studies, IOS Press, Amsterdam (2005)
14. Phillips, S.C.: GRIA SLA Service. In: Phillips, S.C. (ed.) Cracow Grid Workshop, Cracow, Poland, October 15-18 (2006)

15. Ferris, J., Surridge, M., Watkins, E.R.: Business Value Chains in Real-Time Interactive Applications. In: Altmann, J., Neumann, D., Fahringer, T. (eds.) GECON 2008. LNCS, vol. 5206, pp. 1–12. Springer, Heidelberg (2008)
16. Müller, J., Gorlatch, S.: Scaling Online Games on the Grid. In: GDTW 2006, Liverpool, UK, November 15-16 (2006)
17. Ferris, J., et al.: Edutain@Grid: A Business Grid Infrastructure for Real-Time On-line Interactive Applications. In: Altmann, J., Neumann, D., Fahringer, T. (eds.) GECON 2008. LNCS, vol. 5206, pp. 152–162. Springer, Heidelberg (2008)
18. Modadugu, N., Rescorla, E.: The Design and Implementation of Datagram TLS. In: NDSS 2004 (February 2004); See also IETF RFC 4347 on the UDP implementation
19. Transport Layer Security, See: `http://www.ietf.org/rfc/rfc4346.txt`
20. Web Services Security, See: `http://www.oasis-open.org/ committees/ download.php/ 16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf`