Interactive Realtime Multimedia Applications on Service Oriented Infrastructures

ICT FP7-214777

Guaranteeing QoS with Dynamic and Automated SLAs in real-time aware SOIs

Authors

National Technical University of Athens (NTUA)
Andreas Menychtas
Spyridon Gogouvitis
Kleopatra Konstanteli
Universität Stuttgart (USTUTT)
Georgina Gallizo
Roland Kuebert
Telefonica I+D (TID)
Jesus M. Movilla
Scuola Superiore Sant'Anna (SSSA)
Tommaso Cucinotta

Acknowledgements

The research leading to these results has received funding from the EC Seventh Framework Programme FP7/2007-2011 under grant agreement n° 214777

More information

The most recent version of the public deliverables of IRMOS can be found at http://www.irmosproject.eu

Glossary of Acronyms

Acronym	Definition
A-SLA	Application SLA
FP	Framework Programme
IRMOS	Interactive Realtime Multimedia Applications on Service Oriented Infrastructures
ISONI	Intelligent Service Oriented Network Infrastructure
MAPS	Modelling, Analysis, Planning and Specification
QoS	Quality of Service
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SOI	Service Oriented Infrastructure
T-SLA	Technical SLA

Table of Contents

List of Figures

1. Introduction

A Service Level Agreement (SLA) is a contract between the provider and the consumer of a service and specifies the function performed by the service, the obligations on both the provider and consumer of the service, the agreed bounds of performance for the service and how deviations are handled (exceptions and compensations)[2]. An SLA is made in a concrete business context and, therefore, it must include all aspects of the context related to the provided service that are relevant to all interested parties. Depending on the involved stakeholders, the service terms in which the service is agreed are of different nature (ranging from high level business requirements to hardware resources). It is also important that the SLA encloses only what is strictly necessary to formally capture the interests between provider and consumer.

When the service involves multiple consumers/providers, independent bipartite interactions may not be sufficient to cover the inclusion of all stakeholders and all requirements to guarantee the end to end service provision. The SLA management framework must therefore support an end to end SLA negotiation, considering requirements from high-level customer requirements to low-level parameters of the resources. Furthermore, it must be flexible enough to support a wide set of value chain combinations.

There are various approaches in the field of SLA management in distributed environments. In many cases, SLAs are modelled according to business objectives of both customers and service providers as discussed in [3], [4] and [5]. Authors in [6] and [7] present approaches that deal with SLA management by providing Quality of Service (QoS) guarantees at the same time.

What is regarded as of major importance refers to the provision of such guarantees for *real-time interactive* applications. Interactive real-time applications distinguish themselves from normal applications in that they require strong QoS guarantees. However, in this context the focus is on *soft* real-time applications, that are distinguished from the traditional *hard* real-time ones (typical of engine control in automotive and aerospace, or control of industrial plants), in various aspects. First, the in-place timing constraints are far from being safety-critical: failing to respect a timing constraint (e.g., a throughput or response-time constraint) as established into an SLA may cause the responsible party to incur a monetary penalty, certainly not life losses. Also, as a consequence of a few QoS constraint violations the system does not crash or become unusable, on the contrary the system is often perfectly capable of tolerating them if they occur temporarily, and the more frequently they occur, the more the provided service results degraded (graceful degradation). Second, the basic time granularity by which timing constraints are dictated (e.g., in terms of response times) is in the order of magnitude of tens or hundreds of milliseconds and not microseconds as it happens in the hard real-time world. However, in presence of SLAs that determine money losses as a consequence of possible QoS constraint violations, even for soft real-time applications the time factor is of utmost importance and needs to be given full attention, and the criticality of timing constraints needs to be properly faced with. Unfortunately, nowadays strong real-time guarantees are only achievable through dedicated hardware

and software and are, therefore, not in general provided by Service Oriented Infrastructures (SOIs).

In that framework, the ICT FP7 project IRMOS [1] will unite two, at the moment, distinct worlds: Service Oriented Infrastructures (SOIs) and real-time applications. For the provision of real-time applications over SOIs and since multiple stakeholders are expected to be involved in the value chain, a number of QoS requirements needs to be taken into account and guaranteed through the whole chain. An SLA management framework should consider these factors in order to ensure the provision of interactive real-time applications in SOIs.

2. IRMOS Description

The main objective of the IRMOS project is to build a Service Oriented Infrastructure (SOI) for interactive applications coping with real-time requirements. The realization of a challenging framework merging Service Oriented Architecture (SOA) concepts with demanding applications that have real-time interactivity requirements, needs a multi-disciplinary approach in its implementation. Although the platform designed within the IRMOS project will be independent from applications, they will be validated through different specific applications and demonstrators so as to validate the main concept of IRMOS as a general purpose SOI.

The IRMOS reference application scenarios have been selected in order to impose requirements that reflect requirements from highly demanding interactive real-time applications, namely digital film postproduction, Virtual and Augmented reality and interactive real-time E-learning. These applications demonstrate the need for a broad set of diverse tasks to be incorporated in their functionality (computational load, transcoding, streaming, etc...) as well as some attributes that have to be exposed in order to allow for an acceptable user-experience. These will be detailed in the following paragraphs.

The main structure of the IRMOS architecture [3] can be divided into three predominant layers, namely the ISONI Layer, the IRMOS Framework Services Layer and the Application Layer (see Figure 1).

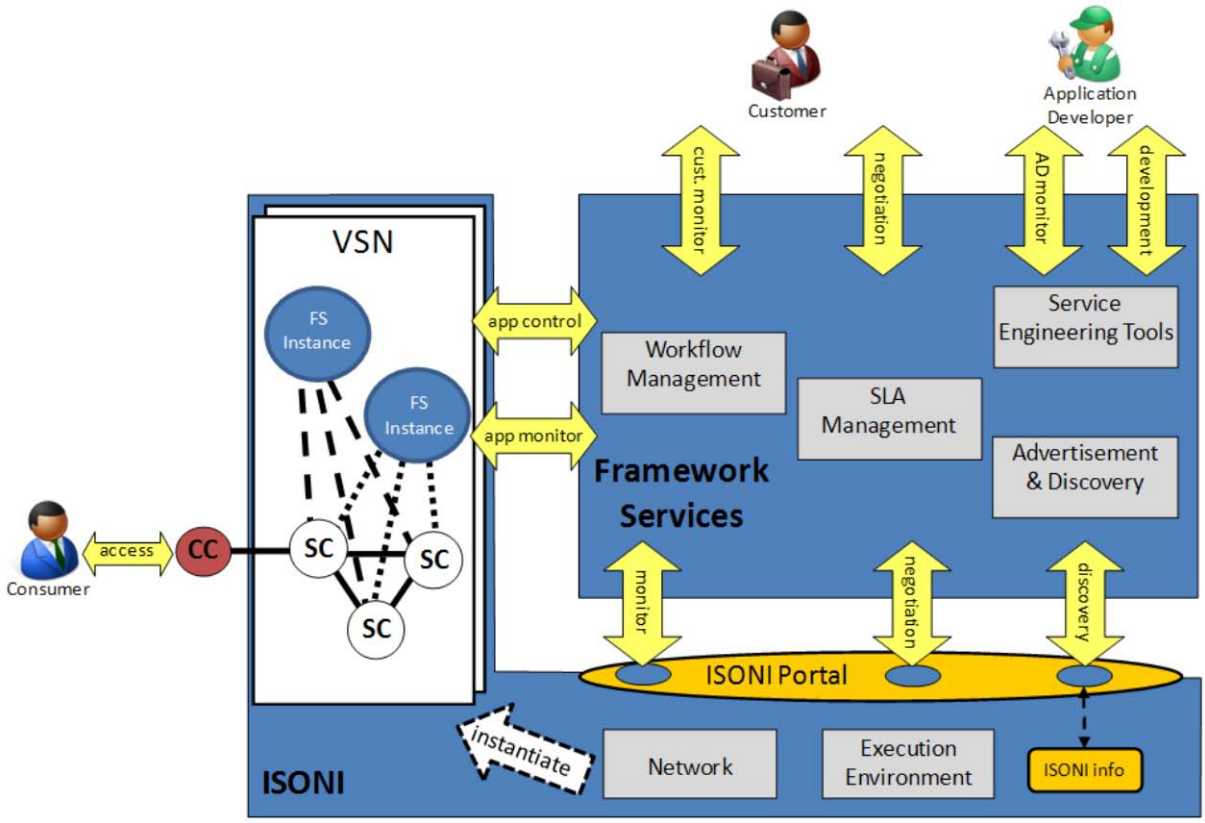

Figure 1: IRMOS Platform Overview

The Application Layer consists of the application and the corresponding models and templates that are required in order to allow the application to be adapted and run on the IRMOS real-time environment. The Framework Services Layer provides all the functionality needed for an application to run on the infrastructure, taking into consideration the real time requirements of the applications. This functionality includes modelling, benchmarking, SLA and Workflow Management, Monitoring, Advertising and Discovery. In addition, the ISONI Layer virtualizes the infrastructure and resources offered by one or multiple Operators/Providers. Consequently, the SLA Management functionality in IRMOS is placed at the Framework Services Layer as an intermediate position between the applications used by the final users and the virtualized resources.

3. The IRMOS approach

3.1. SLA Management

The SLA Management covers different phases: the publication of services' descriptions and their capabilities, the negotiation and establishment of contractual bindings (the Service Level Agreements), the provisioning of the resources and supervision, monitoring, evaluation and enforcement of the SLA terms.

The management of a service level agreement is a complex task as the overall service quality depends on several aspects such as system behaviour, network reliability, external dependencies and even unexpected events. This makes the SLA Management a

complex task that affects the way the internal resources are provisioned and managed. In case of service outage or failure that will cause an unavoidable SLA violation, the service provider should take decisions based on its business relationships with clients, the current infrastructure status and incurred penalties, in order to minimize the global cost incurred by the SLA violation.

SLA Negotiation describes the different steps leading clients and providers to agree on an SLA. The process begins with the client specifying the desired performance conditions and it involves Service Discovery mechanisms and different matching policies as well as methods for selecting and discarding the service provider offerings and renegotiation. The key point is the existence of quotes (financial offers), allowing all parties to negotiate by changing the offer provided by the other side. The client and provider have a common expression language through the use of templates. Once there is an agreement, an SLA contract is established, containing information on the agreed QoS level and price. On the other hand, the monitoring of the infrastructure and the adaptation of the running processes of tasks applying scheduling, fault recovery and migration techniques are essential for the Service Provider to perform SLA Enforcement in order to guarantee the QoS level required by the end user.

3.2. Types of SLAs and value chain aspects

Within IRMOS there are different actors placed at different levels: we have consumers/customers, IRMOS Application Providers, IRMOS Providers and ISONI Providers. The value chain that has been identified can be seen in Figure 2; the actors in the value chain are explained in detail in the following list.

- The **ISONI Provider** is the responsible for operating the **ISONI Layer**. The ISONI Provider virtualizes the infrastructure and resources offered by one or multiple Operators/Providers.
- The **IRMOS Provider** is the holder of the **Framework Services**. It provides all the services required by the Application Providers to use the IRMOS platform.
- The **IRMOS Framework Services Software Provider** is a software vendor that provides the IRMOS Framework Services layer to IRMOS Providers.
- The **Application Provider** interacts with the IRMOS Framework Services as it provides application as a service over the IRMOS Platform.
- The **Client** (both customer and consumer) uses the interfaces provided by the Application Provider to interact with the application, for defining an application SLA.

The specific requirements of these five different actors during the execution of an application within IRMOS cannot be summarised in one single SLA. The parameters typically presented in a SLA are not the same for the customer than for a resource provider (*ISONI Providers*). This is the main reason why there should exist different one-to-one agreements between the actors identified. Namely these are:

- **The Application SLA (A-SLA)**: This is an agreement established between the *Client* and the *Application Provider*. These actors represent the consumer/customer of the applications that are going to be deployed over the

IRMOS platform and the Application Provider itself. This SLA contains the high level QoS parameters of the application defined by the user.

- **The Technical SLA (T-SLA)**: This agreement is going to be negotiated between the *ISONI Provider* and the *IRMOS Provider*. Therefore, this agreement contains low level QoS parameters associated with the infrastructure. The translation of the high level parameters coming from the Application SLA to the low level parameters is performed during the Mapping process.

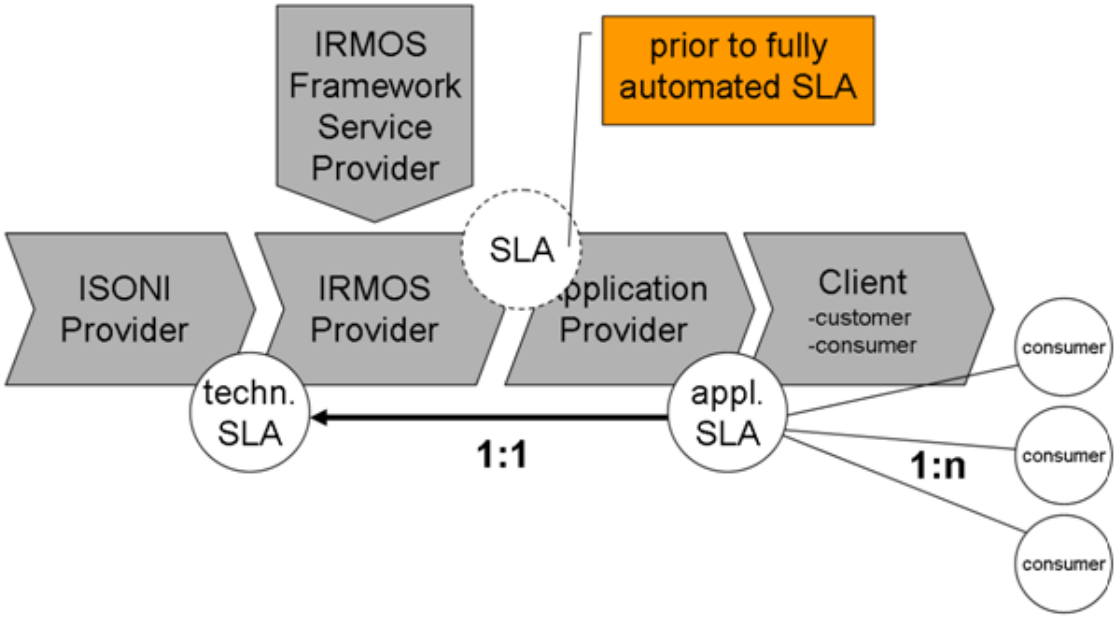

Figure 2: SLA Types between IRMOS Actors

In addition, there is another SLA established between the *Application Provider* and the *IRMOS Provider*, but this relation is a long-term relationship that is previous to the automated SLA negotiation process and that does not have to be negotiated every time a new customer uses an application over the IRMOS platform. Therefore, during the SLA negotiation phase this SLA is considered as fixed.

3.3. SLA innovative aspects

IRMOS is a real-time enabled SOI in which applications are deployed on virtual environments supervising the application lifecycle and guaranteeing the agreed QoS. To this end, various processes need to take place, which are directly mapped to specific components of the IRMOS platform [3]. IRMOS provides services that support application developers in engineering their applications for the IRMOS platform. This enables the definition of an Application SLA whose requirements can be mapped to low level parameters that can be used for the discovery and reservation of the available resources. This reservation takes the form of a Technical SLA that can be used during the execution of the application to configure the low-level mechanisms at the Operating System and Network Infrastructure levels in order to provide the QoS levels and monitor the processes execution. This is done in a fully automated way. The following figure depicts the IRMOS approach to getting from the Application to the Virtualized Resources:

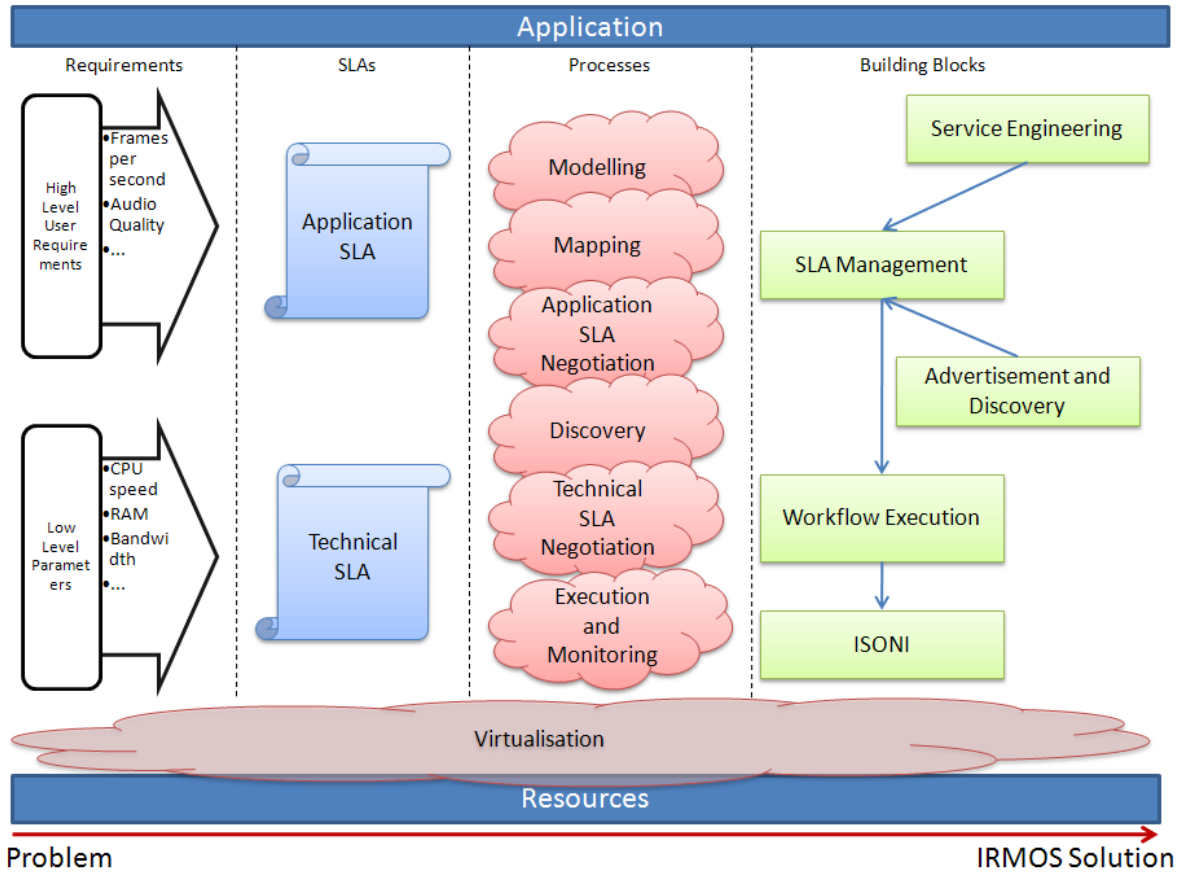

Figure 3: The IRMOS Approach

So far, most of the SLA Management procedures consider the negotiation within the SLA lifecycle to be a process that takes place only once, before the execution phase. Once the negotiation has been produced the service is monitored against the corresponding SLA established and in case there is a violation, then several actions (reflected in the SLA in most cases) are performed. However, sometimes the causes/origin of the violations could be solved by establishing again a process of negotiation. This is called SLA renegotiation or negotiation at execution time. This is one of the innovative aspects of SLA Management in IRMOS and is also known within IRMOS as **dynamic SLA**.

Another important aspect in the SLA Management within IRMOS is the nature of the negotiation process. In IRMOS, we consider the SLA negotiation process to be **automatic**. This means that the part of the IRMOS architecture involved in the process has to some extent the capacity to decide, without human intervention, about the SLA negotiation. This process will be based on policies defined by the actors involved in the negotiation.

To accomplish this task the SLA Management service relies on other services of the platform. Of key importance for the automatic SLA negotiation process are the Benchmarking and Mapping services. The Benchmarking service is responsible for "test running" an application and producing models that are able to describe the applications needs for resources depending on different inputs. These models can thereafter be used by the Mapping service, which is responsible for translating the high level requirements

defined by the user in the Application SLA to low level parameters used in the Technical SLA. The effect of the whole process is that the end user does not need to agree on an SLA (A-SLA) that contains parameters he may not have full understanding of (such as CPU cycles or network bandwidth), but rather defines his requirements in terms he is familiar with (such as frames per second or movie resolution). Moreover, the provider can be assured that he will be able to comply with the requirements posed by the user since the SLAs (T-SLAs) include all the technical parameters to set up a virtual environment as well as the low level QoS requirements for the resources that will be used and provisioned during the application execution.

3.4. Virtualized resources and QoS assurance

One key aspect of the IRMOS project is the capability to foresee the role of a Resource Provider that is capable of providing resources to the IRMOS Application Provider in a virtualized fashion. This means that, by recurring to virtualization technologies, the Resource Provider is capable of instantiating within the platform entire Virtual Machines (VMs) on the behalf of the IRMOS Application Provider, in a transparent way, i.e., without any need for understanding what actual Operating System or Applications therein are being instantiated on the platform. The only information that is relevant for the Resource Provider is: the type of resource to instantiate (e.g., computation, storage), the topology of their interconnections and the QoS requirements (i.e., Mflops, memory occupation, network throughput and latency) associated to the resources and their interconnections.

A key aspect of adopting virtualization technology is the capability for a Resource Provider to share the same physical resource (i.e., computing node) for hosting multiple virtualized resources (i.e., Operating Systems). While this is crucial from a business perspective, because it allows for scaling the costs of the offered virtualized resources, at the same time this aspect raises the challenging issue of how to guarantee the appropriate QoS guarantees to individual VMs.

As discussed in [9], the IRMOS approach to tackle this problem relies on the use of appropriate *soft real-time scheduling policies* for the management of the underlying physical resources, when shared across multiple VMs. However, this constitutes one of the key challenging issues the IRMOS Project is investigating and performing research at the levels of Virtualization Layer and Operating System kernel.

4. Conclusions

One of the main objectives of IRMOS project is to design and develop a real-time aware SOI capable to support dynamic and automated SLAs in all phases of the application lifecycle. To this direction, IRMOS consortium implements innovative approaches, such as the multi-layered SLAs and renegotiation, for addressing on one hand the requirements from the various actors -in business level- and for guaranteeing on the other hand the real-time QoS -in application and resource levels- following advanced provisioning methods.

5. Background

The IRMOS project will design, develop and validate a Service Orientated Infrastructure which will allow the adoption of interactive real-time applications, and especially multimedia applications, enabling their rich set of attributes (from time-constrained operation to dynamic service control and adaptation) and their efficient integration into the infrastructure. See reference [1] for more details.

The IRMOS partners are Xyratex (UK), Institute of Communication & Computer Systems – National Technical University of Athens (GR), Universität Stuttgart (DE), Alcatel-Lucent Deutschland AG (DE), STIFTELSEN SINTEF (NO), IT Innovation (UK), Scuola Superiore Sant'Anna (IT), Telefonica I+D (ES), Giunti Labs (IT), Grass Valley Germany GmbH (DE), Deutsche Thomson OHG (DE).

6. References

[1] IRMOS, http://www.irmosproject.eu
[2] Kleopatra Konstanteli et al., IRMOS deliverable D2.3.1 State of the Art on IRMOS technologies, August 2008, available at the URL: http://www.irmosproject.eu/Deliverables
[3] Andreas Menychtas et al., IRMOS Deliverable D3.1.2 IRMOS Overall Architecture, January 2009, available at the URL: http://www.irmosproject.eu/Deliverables
[4] Buco, M. J. Chang, R. N. Luan, L. Z. Ward, C. Wolf, J. L. Yu, P. S., Utility computing SLA management based upon business objectives, IBM Systems Journal, 2004
[5] L. M. Ching, Dr L. Sacks and P. McKee, "SLA Management and Resource Modelling for Grid Computing", Whitepaper, UCL, 2003
[6] H. Chen, H. Jin, F. Mao, H. Wu, "Q-GSM: QoS Oriented Grid Service Management", Web Technologies Research and Development - APWeb 2005, Lecture Notes in Computer Science, 2005
[7] Padgett, J., K. Djemame, and P. Dew, "Grid-based SLA Management", Lecture Notes in Computer Science, pp. 1282-1291, 2005
[8] B. Mitchell and P. McKee, "SLAs A Key Commercial Tool", Exploiting the Knowledge Economy - Issues, Applications, Case Studies, eChallenges 2006
[9] Tommaso Cucinotta et al., IRMOS Public Deliverable "D6.4.1 Initial Version of Realtime Architecture of Execution Environment", January 2009, available at the URL: http://www.irmosproject.eu/Deliverables.