Interactive Realtime Multimedia Applications on Service Oriented Infrastructures

ICT FP7-214777

IRMOS Platform Architecture Whitepaper

IRMOS_Platform_Architecture_Whitepaper_v1_0

Delivery: 14.01.2011
Version 1.0

Project co-funded by the European Commission within the 7th Framework Programme		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission)	
RE	Restricted to a group specified by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	

Responsible Partner: ICCS/NTUA

Revision history:

Date	Editor	Status	Version	Changes
01.07.2010	Andreas Menychtas	Draft	0.1	Initial ToC
15.07.2010	Andreas Menychtas	Draft	0.2	Updated ToC and input in sections of platform analysis and platform architecture
22.10.2010	Andreas Menychtas	Draft	0.3	Input in application phases section, introduction and conclusions
26.10.2010	Andreas Menychtas	Draft	0.4	Input in control loops sections
18.11.2010	Andreas Menychtas	Draft	0.5	Updated figures
22.12.2010	Andreas Menychtas	Draft	0.6	Input in introduction and conclusions sections
22.12.2010	Andreas Menychtas	Draft	0.7-0.8	Minor updates on figures
10.01.2011	Andreas Menychtas	Draft	0.9	Updates following the internal review
14.01.2011	Andreas Menychtas	Final	1.0	Final

Authors

Andreas Menychtas, Spyridon Gogouvitis, Dimosthenis Kyriazis (NTUA); Eduardo Oliveros (TID); Karsten Oberle, Thomas Voith (ALUD); Michael Boniface (IT-Inn); Georgina Gallizo, Soeren Berger (USTUTT)

Internal Reviewers

Sai Narasimhamurthy (XY), Dimosthenis Kyriazis (NTUA)

Copyright

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 this document and all other public deliverables of IRMOS can be found at http://www.irmosproject.eu

Glossary of Acronyms

Acronym	Definition
CC	Client Component
FP7	Seventh Framework Programme
FS	Framework Services
IaaS	Infrastructure as a Service
ICT	Information and Communications Technology
ICT	Information and Communication Technology
IRMOS	Interactive Realtime Multimedia Applications on Service Oriented Infrastructures
ISONI	Intelligent Service Oriented Network Infrastructure
PaaS	Platform as a Service
QoS	Quality of Service
SaaS	Software as a Service
SC	Service Component
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SOI	Service Oriented Infrastructure
SPI	Software, Platform, Infrastructure
VSN	Virtual Service Network

Table of Contents

List of Figures

1. Introduction

IRMOS FP7 Project [2] aims to *"design, develop and validate a **Service Oriented Infrastructure** (SOI) which will allow the adoption of interactive real-time applications, and especially multimedia applications, enabling their rich set of attributes and their efficient integration into the infrastructure"*. This paper provides a summary of the IRMOS Platform Specification considering the overall architecture and the design of the subsystems. The *Platform Specification* [6] was developed with specific consideration of interactive real-time application requirements and how such applications can be provisioned and managed using a Service Oriented Infrastructure whilst respecting business and commercial constraints. A synopsis of the results of this work is provided while other documents and publications are referenced to enlighten the reader who is interested on particular aspects or research outcomes of the project.

1.1. Overview & Concepts

The IRMOS platform builds on service-oriented design principles and allows for the adaptation and operation of interactive applications with real-time requirements. Economically important sectors relevant to Future Internet applications have been used to drive architectural analysis including collaborative post production in the film industry, augmented virtual reality for engineering design and virtual worlds for e-Learning [7].

SOIs are now an inextricable part of ICT and Clouds facilitating a new trend for the next generation of flexible ICT systems. IRMOS combines SOIs with virtualisation technologies to manage and virtualize compute, storage and networking resources as well as to communicate with legacy systems such as wifi locators. SOIs have several advantages that IRMOS exploits in order to dynamically connect people, processes and information, and applications spanning different domains throughout all layers of the architecture. The IRMOS Platform specification advances existing service-oriented approaches by providing methodologies, tools and mechanisms in order to efficiently operate, manage and reconfigure services and resources under *real-time constraints*. The constraints are expressed as Quality of Service (QoS) terms in SLAs that are dynamically negotiated and express commitments between stakeholders in the value chain [3].

Cloud computing offers the potential to dramatically reduce the cost of software services through the commoditization of IT assets and on-demand usage patterns. However, cloud computing is such a generalized paradigm that it is impossible to consider 'the cloud' as a having a single set of Quality of Service issues. To some extent, issues with cloud computing are necessarily related to the application characteristics and purpose(s). However, the complexity of determining resource provision policies for applications in such complex environments introduces significant inefficiencies and has driven the emergence of new classes of service-based systems called Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). Today, these three main classes in the cloud services stack are generally agreed upon

(Software, Platform, Infrastructure - SPI cloud model [15]) and IRMOS follows them introducing a rich set of capabilities for each layer as presented in Figure 1:

- *Infrastructure as a service (IaaS)*: The provision of 'raw' machines (servers, storage, networking and other devices) on which the service consumers install their own software (usually as virtual machine images).
- *Platform as a service (PaaS)*: The provision of a development platform and environment providing services and storage, hosted in the cloud.
- *Software as a service (SaaS)*: The provision of a pre-defined application as a service over the Internet or distributed environment.

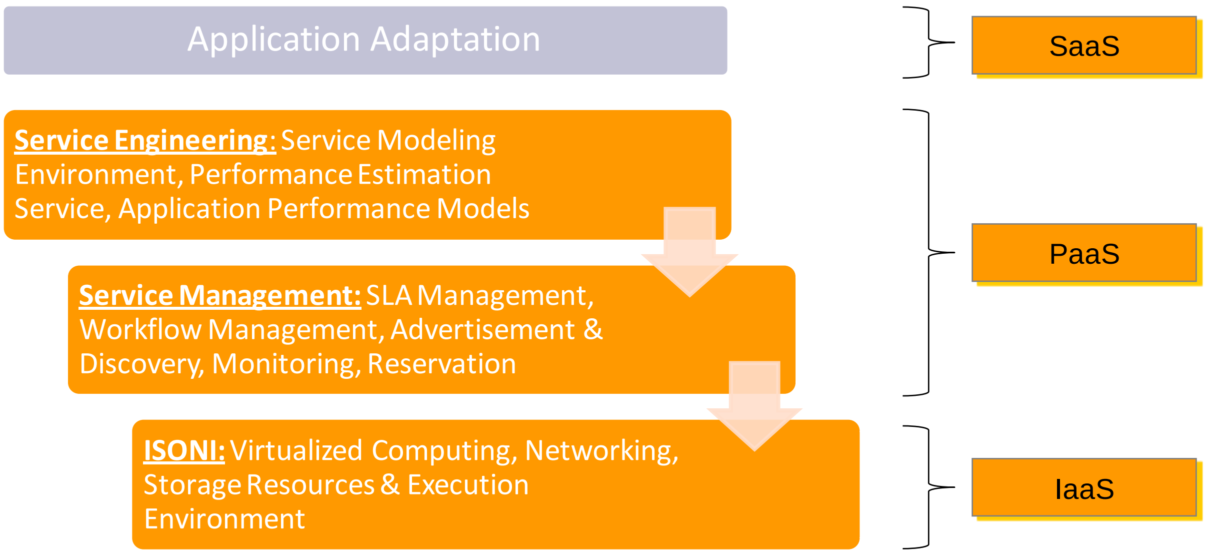

Figure 1: Mapping IRMOS to the Cloud

The capabilities presented in the previous picture are the key for provisioning applications on virtualised infrastructures with guaranteed QoS. At the core is the need to intelligently allocate and adapt resource provisioning policies based on knowledge of application, customer and infrastructure behaviour whilst monitoring actual performance and taking real-time mitigating actions for exceptional behaviour. IRMOS supports these requirements through a set of Framework services [1] that implement a QoS-oriented service engineering methodology which links lifecycle processes with novel modelling tools and autonomic management services [10]. Therefore, the IRMOS platform is not only a set of interacting services, as in the state of the art SOIs, but a real-time enabled SOI in which instances of services with real-time capabilities are deployed in the virtual environments supervising the application lifecycle and guaranteeing the agreed QoS level.

The overall management processes of the IRMOS platform as well as the individual services are designed, developed and distributed in order to support real-time and interactivity not only for the applications but also for the infrastructure itself. IRMOS solves several problems in reference to QoS provisioning, the scheduling of the service execution, the mapping of the application workflows and requirements to low level resource parameters and individual services. This is achieved through cross-layer exchange of management information between Framework Services and ISONI layers guaranteeing real-time end-to-end performance.

2. Real-time in Clouds: The Challenges

IRMOS architecture considers the full service lifecycle of both service-based systems and legacy applications deployed on cloud resources including Service Level Agreement design, resource management, and monitoring. QoS parameters at application, platform and infrastructure levels are given specific attention as the basis for dynamic QoS provisioning in real-time. The platform architecture supporting real-time service, and, human and resources interactivity implements the following key features:

- *Real-Time QoS Specification*: Specification language and associated toolkit for the specification of real-time interactive applications and application service components considering real-time QoS parameters.
- *Event Prediction*: QoS-oriented service engineering methodology and models for predicting QoS requirements contingent on application and resourcing events considering temporal profiles of application service components deployed on virtualised infrastructures.
- *Dynamic SLA Negotiation*: SLA negotiation and management services supporting the dynamic (re-) negotiation of SLAs at application level considering customer requirements, and dynamic discovery of resource providers (SLAs at infrastructure level) through automated processes of the platform.
- *On-Demand Resource Provisioning*: Provisioning services for application service components on virtualised infrastructures through the virtual service networks as well as Execution Environment (IaaS) management implementing high availability techniques such as redundancy and live migration.
- *QoS aware Event Monitoring*: Monitoring services and resources for measuring QoS at both application and infrastructure levels triggering corrective actions for runtime adaptability of resource provisioning.

Service-oriented design principles are considered as an important aspect of the architecture throughout all cloud platform layers. IRMOS platform adopts a service-oriented approach to allow services to interact dynamically and continuously even though they span between different domains, from the application layer to the layer of network resources and the Execution Environment. The challenge is to carefully design and synchronize this rich set of services so as to efficiently operate, manage and reconfigure all resources under real-time conditions, providing to the end users and to the associated applications the appropriate and required QoS. All QoS terms are dynamically negotiated and agreed in SLAs between the various actors of the value chain [3] taking into consideration the QoS guarantees from both application and resource perspectives. All platform and infrastructure capabilities are offered as on-demand services, while the service orchestrations and processes are developed so as to preserve the real-time attributes throughout the whole infrastructure layers.

A major challenge for SaaS providers wanting to exploit the benefits of cloud computing is to manage QoS commitments to customers throughout the lifecycle of a service. The PaaS offers to SaaS providers services tools for estimating resource needs in advance of execution and mechanisms for negotiating QoS and provisioning virtualised resources. The IRMOS approach considers analysis and decision support within temporal and business constraints to determine which actions are triggered offline (i.e. pre-execution) or online (i.e. during execution). Because faults are inevitably going to occur, strong fault detection and recovery mechanisms are implemented. This can have a great impact on the real-time capabilities of the platform, since intelligent fault recovery mechanisms allow timing constraints to still be met in case of a failure. The control loops between cloud layers, described in section 4, is as essential factor in ensuring that QoS guarantees are maintained.

At the IaaS layer, real-time functionality is supported by the Intelligent Networking and the Execution Environment infrastructures (storage boxes and execution nodes) applying virtualization techniques for several types of resources such as networking, storage and computational. The IaaS layer instantiates, manages and monitors the various resources (e.g. Linux Kernel, Network Routers) according to the set of services that are deployed. To this direction, the IRMOS Execution Environment considers multitasking, threads with priorities and an appropriate number of interrupt levels to achieve QoS objectives.

Another essential element of cloud computing, especially of PaaS layer, is the ability to deliver on-demand services with minimal manual configuration. In that sense, all platform subsystems can be self-managed and reconfigured in order to achieve management efficiencies, to react to QoS failures (such as for instance an SLA violation or Network link failure) in a timely way and avoid the escalation of interlayer problems.

Cloud utilisation involves several processes that span in different cloud layers and stakeholders. For example, the IRMOS supports application developers in engineering their applications for the cloud implementing standard specifications and methodologies, while other processes support application provisioning and execution through the innovative virtualised execution environment and networking infrastructure. Therefore, IRMOS platform does not only provide a set of services but also cross layer workflow mechanisms that consider the control channels and information exchanges which are required to support real-time management of interactive applications throughout the full lifecycle.

3. IRMOS Platform Phases

To achieve the aforementioned challenges for real-time functionality, IRMOS platform operation is separated in phases, the *offline,* where the application and application components are prepared (development, modelling, benchmarking, etc) and the *online,* where the resources are negotiated and reserved and the application is initialized and operational.

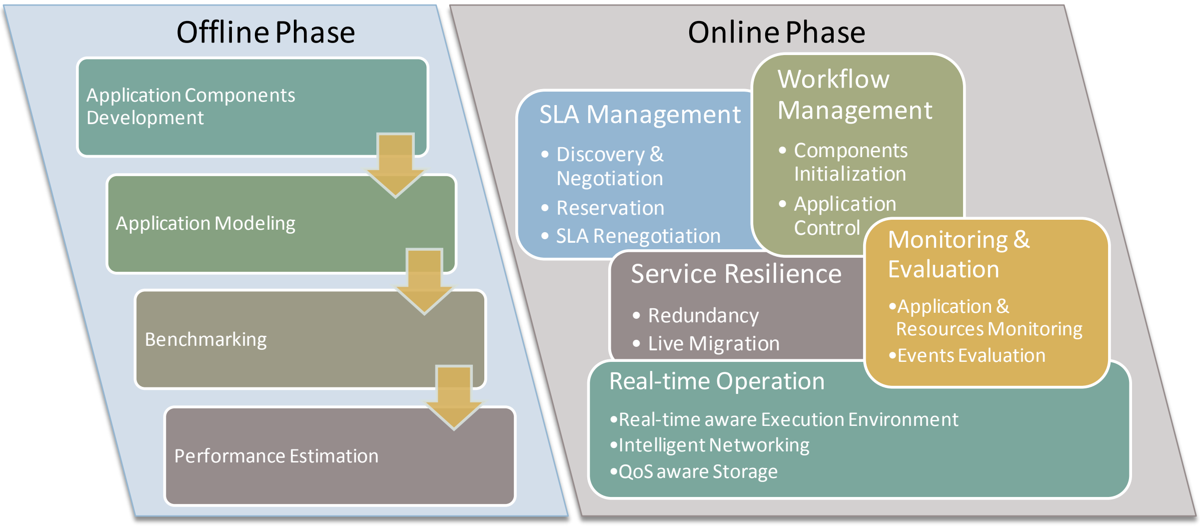

Figure 2: IRMOS Platform Phases

Offline - Design-time Service Engineering

This phase includes the processes for developing and adaptation of the application components to the IRMOS environment. This includes the creation of descriptors and documents for the application operation such as models, mapping rules, initialization scripts, SLA templates, workflows as well as provisioning and management policies based on performance estimation and benchmarking tests [8].

Online – Service Management and Real-time Execution

In the Online phase the customer submits to the platform the high level requirements for the execution of the application. On the platform side a set of processes is initiated which includes among others, mapping of the high level requirements to low level parameters of respective services and SLA negotiation for both the application requirements with the customer and operational and technical parameters with IaaS providers. As soon as the SLAs are signed, the required computational, storage and networking resources are reserved for use within the requested time interval. During the application execution the platform is responsible for orchestrating and monitoring, up to completion, the workflow execution guaranteeing the smooth operation on the agreed QoS levels. In cases of faults, such as resource outage or failures, incorrect estimations and SLA violations, respective corrective actions are triggered including renegotiation of SLAs and resilience mechanisms are put in place such as virtual machines live migration.

4. Control Loops

In order to provide QoS guarantees for interactive real-time multimedia applications, IRMOS platform provides a set of services and cross layer workflows that consider the control channels and information exchanges which are required to support real-time management during the online phase. All subsystems are self-managed and reconfigured in order to achieve management efficiencies, and to react on QoS failures (such as an SLA violation or network link failure) in a timely way. To achieve this, we introduce three control loops at infrastructure level providing the necessary functionality in order to maintain QoS metrics across the architectural levels [9]. The **IRMOS Control Loops** are the following and are depicted in Figure 3:

- *Application Control*: It deals with the relationship between users and applications required to guarantee the application QoS. This control loop is managed by the application itself in response to either user events or platform events. It is implemented with the use of models, workflows and tools that produce artifacts capturing the applications' behavior and estimating resource needs in advance of execution. During runtime it refers to application monitoring that may for example trigger events or require for changes in the provided resources.
- *Environment Control*: It deals with the relationship between applications and virtual resources in order to guarantee the platform QoS, as agreed in the SLAs. This control loop is managed by the platform services in response to application and virtualisation events. It is implemented by the framework services that support and manage the applications at run-time (e.g. actions triggered if either the application or resources do not perform as expected or need to be adjusted).
- *Virtualization Control*: It deals with the relationship between virtual and physical resources in order to guarantee the infrastructure QoS. This control loop is managed within IaaS layer called ISONI in response to platform or physical events. It is implemented by intelligent networking mechanisms as well as by the real-time enabled execution environment for computational and data storage services.

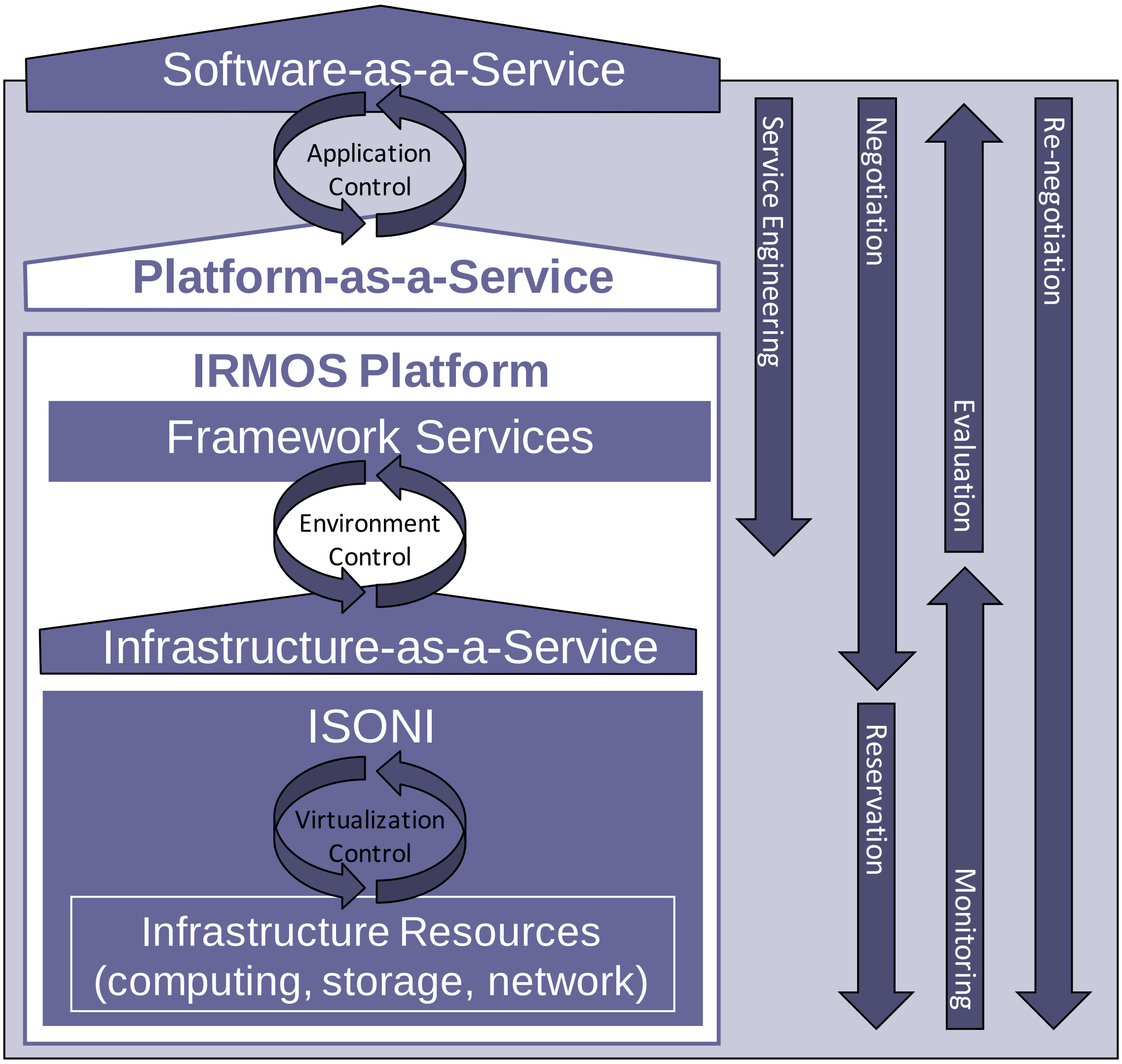

Figure 3: IRMOS Control Loops

The actual implementation of the control loops refers to tools and services used on different levels in order to monitor the applications' execution, communicate possible events and take corrective actions if needed. We identified five (5) main processes / channels implementing the control loops, which we analyze later in the document:

- Service Engineering
- Negotiation / Re-negotiation
- Reservation
- Monitoring and Evaluation

Additional information on the control loops and the exchange of information through the aforementioned channels can be found in [11].

5. IRMOS Platform Architecture

Following the analysis of the platform operation and the management approach via the control loops concept presented in sections 3 and 4 respectively, in this section we describe the overall architecture of the IRMOS platform and its main subsystems. The high-level view of the platform architecture is shown in Figure 4.

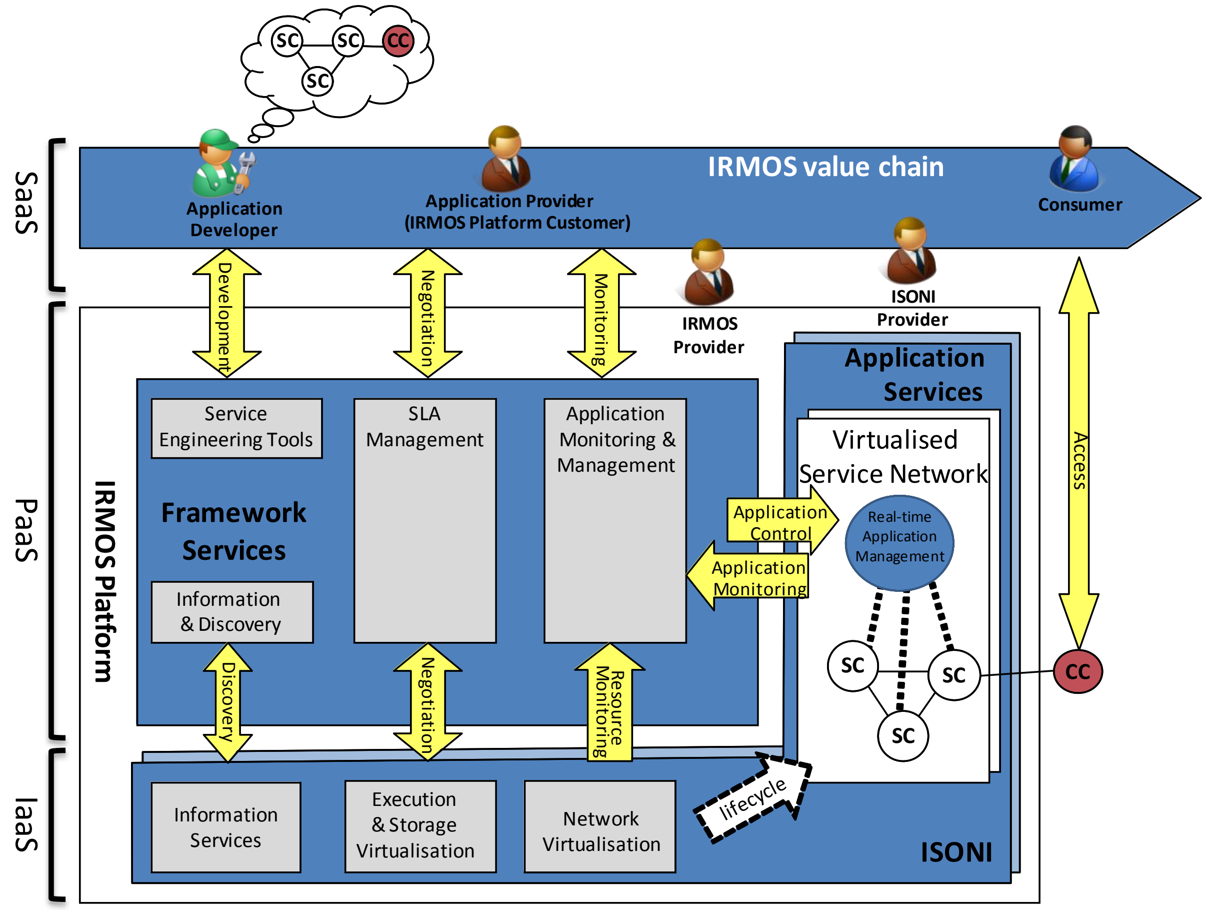

Figure 4: IRMOS Platform Overview

As indicated in the previous figure, the platform has the two main building blocks: PaaS (Framework Services) and IaaS (ISONI). During the architecture design and specification, we followed an innovative approach on how these blocks will interact. In that sense their relation is considerably different of the conventional SOA or Cloud platforms because of its real-time orientation, the virtualization capabilities and the way the management information is shared between platform and infrastructure layers to ensure end-to-end QoS. Initially, the Framework Services provide service engineering tools for the application developer and provisioning services for the IRMOS provider as the entity responsible for offering applications. End users are able to access IRMOS platform in two ways:

- A *customer* negotiates SLAs and monitors the application, consisting of Service Components (SC), through the Framework Services.

- A *consumer* accesses an application deployed in a Virtual Service Network (VSN) through Client Components (CC).

Each VSN is instantiated by ISONI and includes particular technical requirements defined by the application developer at design time and specific QoS customisation defined by the customer at runtime. These requirements are relayed to ISONI during the SLA negotiation through the Framework Services. ISONI cannot be accessed directly by end users (Customers or Consumers) and their access privileges are limited to application service components.

As already mentioned, the Framework Services communicate continuously and in various ways with ISONI. Each ISONI provider advertises its capabilities to Framework Services so as to be discovered later and as second step negotiate SLAs for an application. Additionally ISONI provides monitoring data and notification events for each VSN (at the SC level) to Framework Services that are used for both runtime (control) and design time (development and modelling). The fact that real-time functionality is required on some components of the Framework Services layer demands that instances of these components should be deployed and run in the VSN where the real-time QoS is guaranteed while the core platform services in PaaS interact with service instances running in VSNs for controlling and monitoring the application during application execution.

The Execution Environment and the Intelligent Networking subsystems are architecturally close and are expected to communicate continuously during all the processes of the platform [14]. These subsystems are wrapped in the ISONI infrastructure. The main objective of this layer is to virtualize resources, provide application services and monitor the resources [5] without the need for knowledge of the application itself. The Execution Environment subsystem, considered as an enhanced virtualization platform, includes the storage systems and is implemented so as to address the QoS and especially the real-time requirements of the application services. The network resources, provided through a VPN like approach, are classified and advertised to the Framework Services in QoS classes.

5.1. Framework Services

The Framework Services (FS) is the layer between applications and virtualized resources. As already mentioned, this layer corresponds to the PaaS layer of SPI cloud model and its architecture is shown in Figure 5. The architecture consists of two main elements: Service Engineering and Service Management.

The FS layer aims to provide and manage the execution of real-time services on request of the Application Layer using virtualized resources. These resources are offered by the IaaS providers conforming to real-time constraints as determined by the application SLAs. Apart from the execution of services that are provided to customers, FS support service engineering, fully automated SLA (re-)negotiation, mapping of high level performance parameters to low level resource parameters, and the discovery and reservation of virtualized resources needed for the execution of an application. In the execution phase of the application, FS continuously monitors and manages the application components and the resources either directly, through the application wrappers based on predefined application specific policies, or relaying the management requests to ISONI layer based on operational policies of the platform. It should be noted that instances of the FS such as Workflow Enactor and Monitoring Services are deployed within the VSN allowing for timing-constrained execution and management of real-time applications [4].

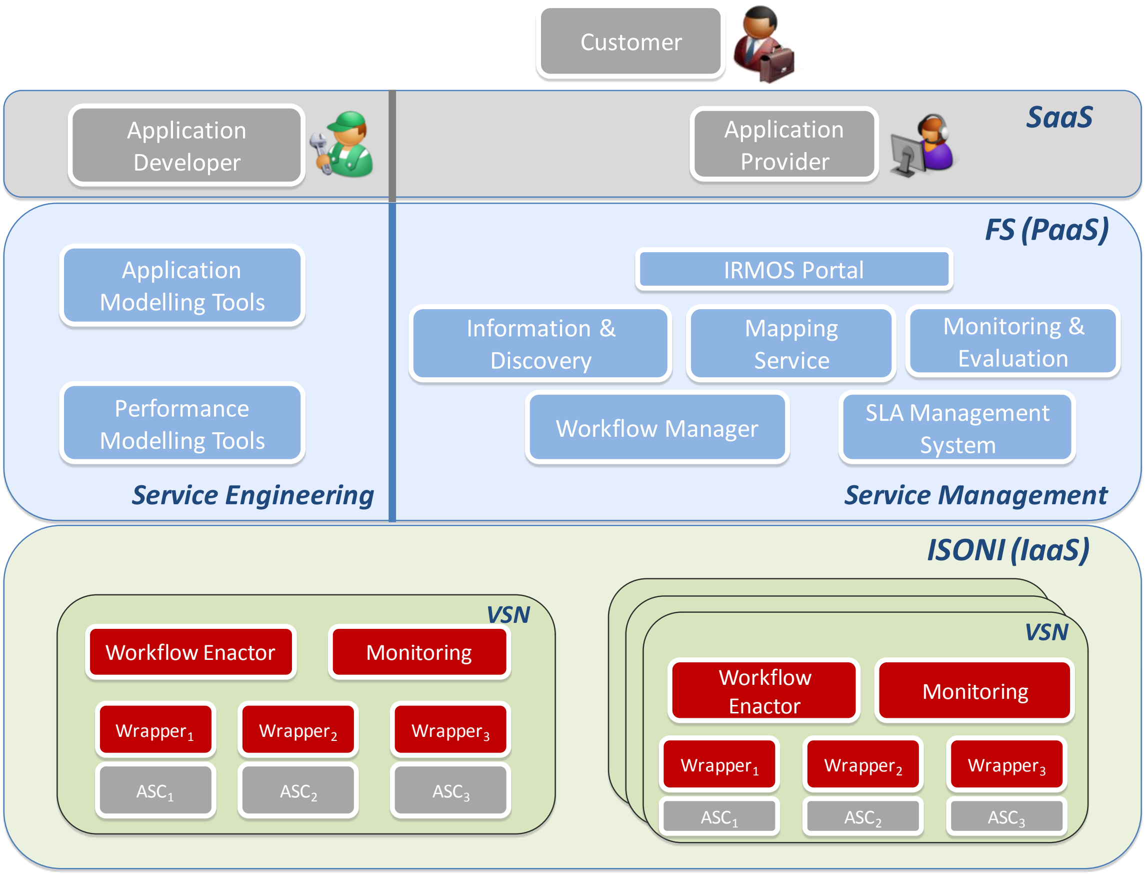

Figure 5: PaaS Architecture (along with associated IaaS and PaaS components)

For the communication with the user of the platform, the *IRMOS Portal* component has been implemented, which provides the necessary interface to enable the end-user of the application to request the SLA templates, invoke the negotiation process and reserve virtualized resources at the IaaS layer. In addition, FS functionality includes starting and stopping of an application execution relaying the requests, through the Service Management system, to the appropriate application service components running in a VSN. Additional information about the functionality and implementation of each component can be found in [6].

5.2. ISONI

ISONI (Intelligent Service Oriented Network Infrastructure) is an IaaS environment, consisting of a network of resources (e.g. CPU, storage, software, etc) managed and controlled by a middleware, which allows resource sharing among multiple services [5].

The general idea is to provide QoS capable infrastructure resources on demand for dynamically deployed services. As already described in a previous section, a service is usually composed of several smaller and simpler services, in the following called Service Components (SC). ISONI is agnostic to services, thus the decomposition of services into SCs is not its responsibility, and is accomplished by the Framework Services layer. The objective of ISONI is to provide these SCs with the best resources (Execution Environments and network links). Figure 6 depicts the main components of the IaaS layer. ISONI exposes its virtualized infrastructure in the form of VSNs [12], which can be seen as a graph whose vertexes are the SCs and whose edges are the Virtual Links. In the proposed platform the Framework Services will state their infrastructure requirements using a VSN description.

It is the role of the Framework Services layer, which is application aware, to decompose its Services and Applications into Services Components, build the VSN description and request resources from ISONI which is application unaware. The VSN description is transferred to the IaaS layer with the request to instantiate the service. Then, the ISONI has to automatically and autonomously map the highly abstracted resource request in form of the VSN description onto the network of real resources, to deploy the components in tailored execution environments on suitable resources, and to interlink them while observing QoS requirements. This instantiated VSN builds an independent Layer 3 overlay network, i.e., there is no limitation on the Layer 3 protocol stack used by the SCs.

The ISONI architecture is composed of functional blocks, where each takes on a different task for the management of the ISONI resources and deployed VSNs.

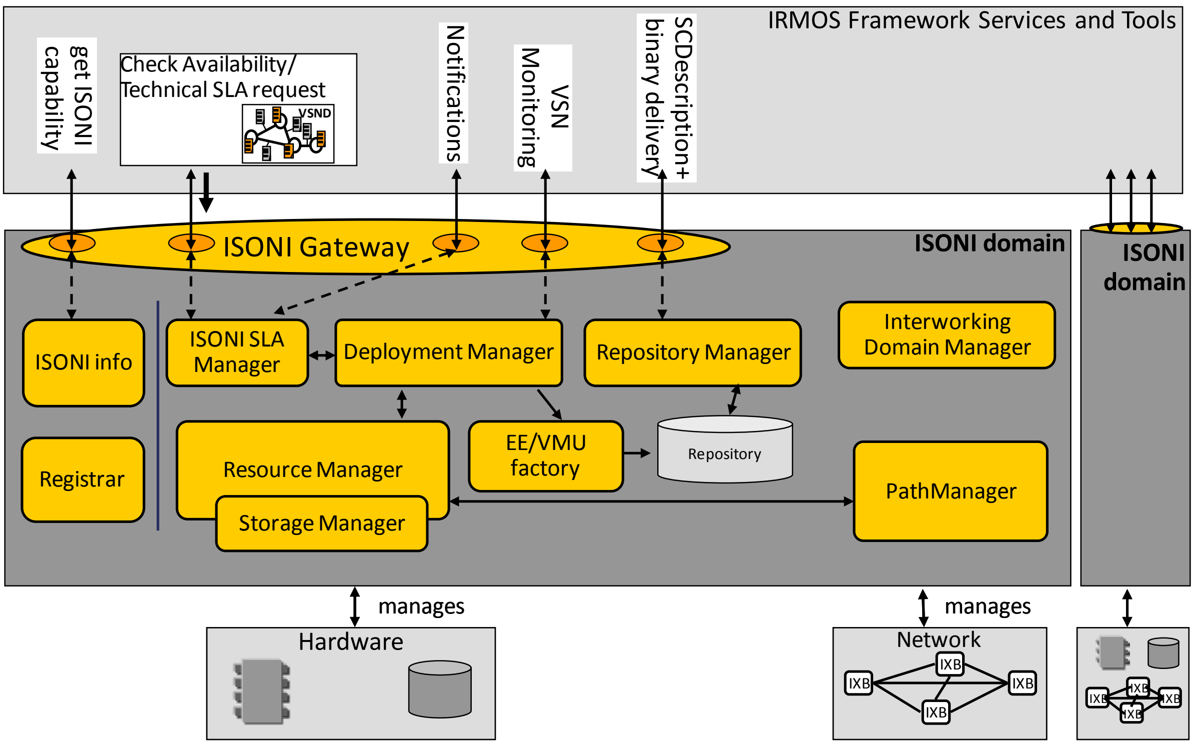

Figure 6: ISONI Internal Functional Blocks

Figure 6 shows the functional building blocks [6] of the ISONI management. The interfaces to the Framework Services and tools are indicated on top. The functional blocks Resource Manager, Storage Manager and Path Manager would be deployed in a two-level management architecture based on the composite structure, *Domain level* and *Node level* [13]. The resource responsibility lies with the Node level, i.e. the Node control and resource reservations are maintained by the middleware functional blocks running at Node level, whereas the Domain level instances coordinate the ISONI Nodes. This approach guarantees efficient management of the VSNs as well as the resource scalability, a key requirement in cloud environments.

6. Conclusions

The paper presented an innovative architectural approach that enables cloud-based environments to support the full lifecycle of applications with real-time QoS requirements. In addition to the platform design and specification we described methodologies that have been followed in the platform to effectively provision and manage application services and infrastructure resources during runtime. IRMOS platform promises to significantly advance the state-of-the-art in provisioning applications with guaranteed QoS on virtualised infrastructures and has been validated by three different application scenarios, which were also the basis for the requirements identification during the design process. Each of the scenarios highlight different capabilities of the platform and requires unique real-time characteristics: (a) A mobile eLearning scenario, which makes use of mobile devices and requires a high degree of scalability for the used resources, due to the high number of potential users; (b) A digital film postproduction scenario, which requires very high bandwidth storage and network, combined with high computing power, in order to complete workflows in real-time, and, (c) A virtual and augmented reality (VAR) scenario, used in the automotive industrial design, which poses very tight requirements on the network infrastructure in order to enable high degrees of interactivity. More details on the architecture and the implementation for the various components can be found in the public deliverables of the project that are available on-line on project web site [2]. Currently we are working on a final prototype, which will be released on January 2011, focusing on service resilience and events evaluation processes so as to advance further the QoS provisioning and real-time management capabilities of the platform.

7. References

[1] Boniface, M., Nasser, B., Papay, J., Phillips, S., Servin, A., Zlatev, Z., Yang, K. X., Katsaros, G., Konstanteli, K., Kousiouris, G., Menychtas, A., Kyriazis, D. and Gogouvitis, S., “Platform-as-a-Service Architecture for Real-time Quality of Service Management in Clouds”, Fifth International Conference on Internet and Web Applications and Services, ICIW 2010, May 2010, Barcelona

[2] EU IST IRMOS Project, http://www.irmosproject.eu

[3] Georgina Gallizo, Roland Kuebert, Karsten Oberle, Andreas Menychtas, Kleopatra Konstanteli , “Service Level Agreements in Virtualised Service Platforms”, eChallenges, Warsaw, 21-23 October 2009.

[4] Gogouvitis S, Kostanteli K, Kousiouris G, Katsaros G, Kyriazis D, Varvarigou T, “Workflow Management in Service Oriented Infrastructures”, International Conference on Utility and Cloud Computing (UCC), India, 2010

[5] IRMOS Project “ISONI Whitepaper” V2.0, ALUD and USTUTT, July 2010.

[6] IRMOS Project Deliverable “D3.1.4 Updated Final version of IRMOS Overall Architecture”, ICCS/NTUA and other partners, January 20101.

[7] IRMOS Project Deliverable “D4.1.1 Definition of the three scenarios and their real-time requirements”, TSG (GVG) and other partners, January 2009.

[8] IRMOS Project Deliverable “D5.1.1 Models of Real time Applications on Service Oriented Infrastructures”, IT Innovation and other partners, January 2009.

[9] IRMOS Project Whitepaper “Achieving Real-time on Service Oriented Infrastructures for Interactive Multimedia”, February 2010.

[10] Kyriazis D, Einhorn R, Furst L, Braitmaier M, Lamp D, Konstanteli K, Kousiouris G, Menychtas A, Oliveros E, Loughran N, Nasser B, “A Methodology for engineering real-time interactive multimedia applications on Service Oriented Infrastructures”, IADIS Applied Computing 2010, Timisora, Romania, 2010

[11] Kyriazis D, Menychtas A, Kousiouris G, Oberle K, Voith T, Boniface M, Oliveros E, Cucinotta T, Berger S, “A Real-time Service Oriented Infrastructure”, International Conference on Real-Time and Embedded Systems (RTES), Singapore, 2010

[12] Oberle, K., Voith, T., Stein, M., Gallizo, G., Kübert, R., “The Network Aspect of Infrastructure-as-a-Service”, ICIN2010, Berlin, October 2010.

[13] Oberle, L., Kessler, M., Voith, T., Stein, M., Lamp, D., Berger, S., “Network Virtualization: The missing piece”, ICIN2009, Bordeaux 26-29.10.09.

[14] T. Cucinotta, F. Checconi, G. Kousiouris, D. Kyriazis, T. Varvarigou, A. Mazzetti, Z. Zlatev, J. Papay, M. Boniface, S. Berger, D. Lamp, T. Voith, M. Stein, “Virtualized e-Learning with Real-Time Guarantees on the IRMOS Platform”, IEEE International Conference on Service-Oriented Computing and Applications SOCA2010, Perth, Australia, 13-15 December 2010.

[15] The NIST Definition of Cloud Computing, Peter Mell and Tim Grance, Version 15, http://csrc.nist.gov/groups/SNS/cloud-computing, 2009

