Interactive Realtime Multimedia Applications on Service Oriented Infrastructures

ICT FP7-214777

WP 7 Intelligent Networking

D7.1.1 ISONI addressing schemes

IRMOS_WP7_D7_1_1_PU_USTUTT_v1_0.doc

Scheduled Delivery:	30/11/2008
Actual Delivery:	27/11/2008
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: Universität Stuttgart -RUS (USTUTT)

Revision history:

Date	Editor	Status	Version	Changes
18.11.2008	Dominik Lamp	Release to final QA	0.9	Created from the reviewed version of the restricted edition
27.11.2008	Dominik Lamp	Final	1.0	Minor spelling corrections, updated index of figures, updated creation date and date of delivery. Version approved by QA.

Authors

Dominik Lamp, Patrick Mandic (USTUTT), Thomas Voith, Manuel Stein, Karsten Oberle (ALUD), Ralf Einhorn, Lars Fürst (DTO)

Internal Reviewers

George Kousiouris (NTUA); Mike Boniface (IT-Inn); Consolidating Reviewer: Alberto Leon (TID)

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
AC	Application Component
ACC	Application Client Component
API	Application Programming Interface
AS	Autonomous System
ASC	Application Service Component
ATCA	Advanced Telecommunications Computing Architecture
CC	Client Component
CPU	Central Processing Unit
D	Deliverable
DSP	Digital Signal Processor
eSC	external Service Component
iGW	interworking Gateway
IP	Internet Protocol
IRMOS	Interactive Realtime Multimedia Applications on Service Oriented Infrastructures
ISONI	Intelligent Service Oriented Network Infrastructure
ISP	Internet Service Provider
IXB	ISONI eXchange Box
L	(OSI-)Layer
MAN	Metropolitan Area Network
NAT	Network Address Translation
PH	Physical Host
POP	Point-of-Presence
PSTN	Public Switched Telephony Network
QoS	Quality of Service
SC	Service Component
SLA	Service Level Agreement
STP	Spanning Tree Protocol
VMU	Virtual Machine Unit
VPN	Virtual Private Network
VSN	Virtual Service Network
WAN	Wide Area Network

Table of Contents

List of Figures

1. Executive Summary

The IRMOS framework aims at enabling infrastructure providers to offer hosting for real-time-capable and highly customizable services. To cut costs, the infrastructure is shared among multiple customers, giving each exclusive access to a virtual environment.

In the context of the deliverable at hand, an addressing scheme that supports the isolation of applications that are deployed to the IRMOS framework on the level of the Intelligent Service Oriented Infrastructure (ISONI) has been developed. Basically, this is done by transparently adding namespace information to each executed application, which is described as a so-called Virtual Service Network.

To ensure (controlled) interoperability with arbitrary clients and services on the Internet as well as with other Virtual Service Networks (VSNs), an interoperability mechanism has been developed. The actual resources to use for execution of ISONI Service Components (SC) are chosen shortly before the anticipated application start time. Due to the developed addressing scheme, this process is completely hidden from the application: The addressing inside the application's VSN is done entirely on the basis of developer-defined (virtual) addresses that are transparently mapped to the concrete instance by the ISONI. As described in chapter 3 on the ISONI architecture, the actual resources used to realize a service may be changed during runtime to mitigate influences of failures and maintenance on the Quality of Service (QoS) delivered to the application. Such changes are completely hidden from the application by the developed addressing scheme, i.e., address changes are not propagated to the virtual addresses that are visible to the application.

To actually provide the connectivity between Service Components, so-called ISONI eXchange Boxes (IXB) have been designed. These building blocks are responsible for "tying" all traffic to the VSN it belongs to. The design and mediation between virtual addresses and resource addresses can be found in chapter 6.1.

The fundamental IXB functionality has been implemented to serve as building block for the ISONI part of the IRMOS integrated prototype. As a substitute for the actual Path Management Infrastructure, which is to be developed as part of D7.2.1, hardcoded scenarios have been used to verify suitability of the IXB concept. For the integration with future deliverables, an API has been made available.

2. Introduction

The document at hand is the basis for further work on the Intelligent Service Oriented Infrastructure (ISONI). It describes the underlying addressing scheme that is used to isolate multiple Virtual Service Networks (VSNs) from each other on network level and provide a framework to enable live migration of Virtual Machine Units (VMUs).

The ISONI's general concept has already been introduced in the ISONI Whitepaper [1] and D3.1.1 "Preliminary Version of IRMOS Overall Architecture" [2]. This deliverable extends the concept of an ISONI addressing scheme and The work described in this document has been carried out in coordination with the research performed on the Execution Environment and Live Migration tasks of IRMOS. As a fundamental part of the ISONI and thus IRMOS architecture, it is intended as input for all further technical work involving the network layer.

This document is structured as follows. First, a brief introduction to the ISONI concepts is given in chapter 3 for readers that are not yet familiar with the ISONI Whitepaper. In chapter 4, the requirements for the ISONI addressing scheme are given, before the actual addressing scheme is introduced. The chapter concludes with the presentation of mechanisms provided for external access to a VSN. The addressing concept is validated in chapter 5. This is done by discussing how the concept is used in the different scenarios that occur when a VSN is deployed to ISONI. The IXB as main connectivity-providing building block is detailed in chapter 6.1. Finally, Live migration and its relationship to the addressing scheme is addressed in chapter 7.

2.1. Objectives

The objective of this document is to define and describe the ISONI addressing scheme from an architectural point of view, while the addressing scheme is developed in line with requirements defined by WP3. It is also accompanied by a prototypical implementation of the IXB building block, which will be built upon in development of further prototypes.

3. Introduction into Intelligent Network Service Oriented Infrastructure (ISONI)

The basic purpose of the ISONI is to considerably reduce the complexity for service providers/developers to roll out new network-based services as it takes care of the automatic deployment of the services on best fitting resources distributed in a network. The solution strives to reduce global costs introducing a resource infrastructure provider that optimizes costs by means of virtualization techniques, whereby tailored resources can be provided for the deployment of services. Additionally, the ISONI will provide means to isolate different deployed services from each other in order to prevent unwanted crosstalk between them. To live up to that purpose the ISONI has to carry out several tasks.

The first major task of the ISONI is to completely separate the management of all kind of hardware resources distributed in a network from that of deployed services and their associated service components. By that, the actual status and distribution of resources are hidden from the application developer's view. The infrastructure provides fully virtualized resources, including the network resource. That enables an application developer to deal with a complicated network of resources in a simplified way at a level of high abstraction. This full virtualization of a network of distributed resources is the essential prerequisite to get the freedom of resource and service management we need to serve the purpose of the ISONI as described above.

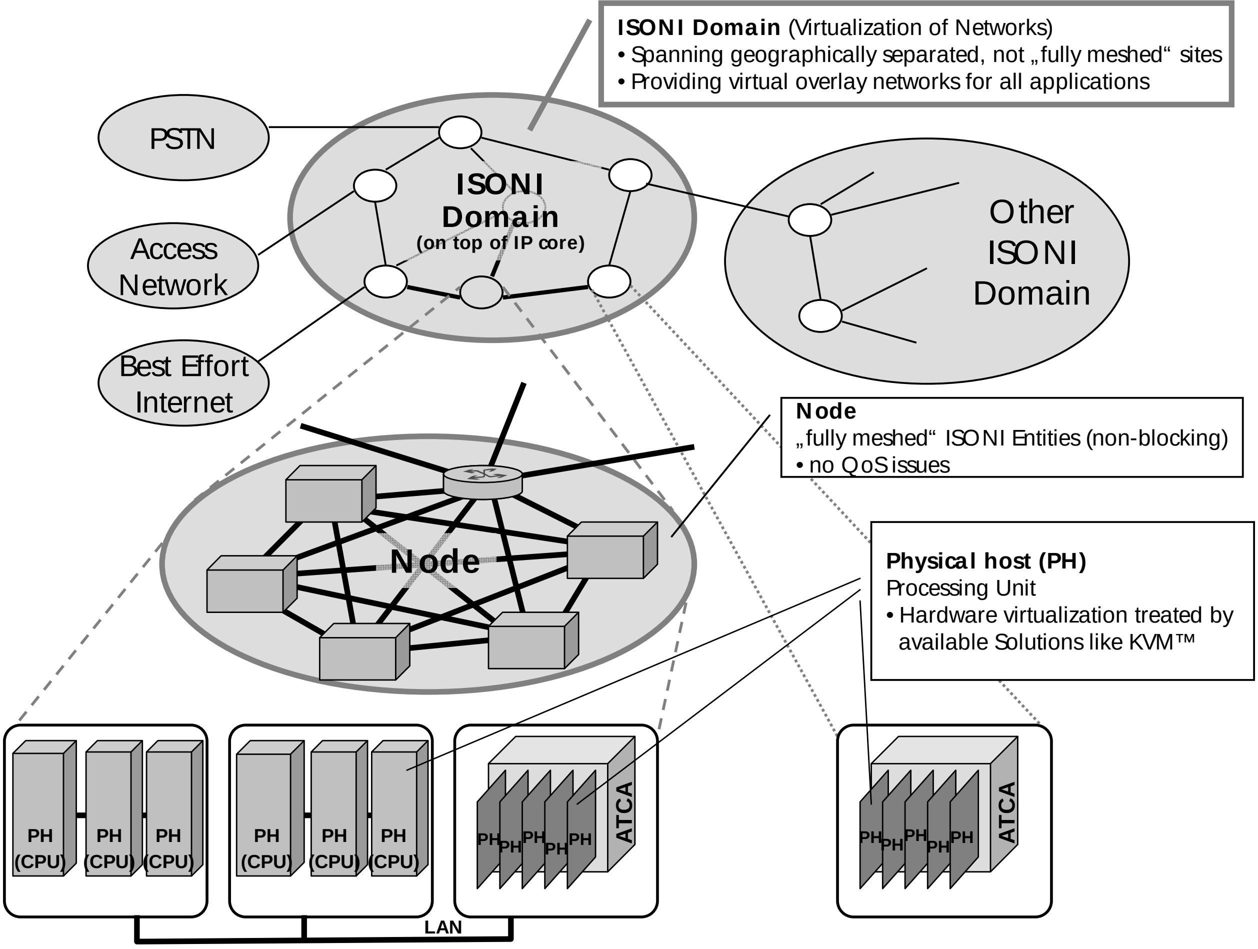
Virtualization is also used to hide outages in network or computational resources. Should network links fail or go down for maintenance, traffic can be transparently rerouted. Should a computational resource become unavailable due to (scheduled) maintenance, a replacement is made available and execution is automatically resumed.

The second major task of the ISONI is to deploy and instantiate the application developer's service on the ISONI. The ISONI will be able to accomplish this task automatically and autonomously, which is the main goal of the ISONI development. For that the ISONI needs an abstract description of all the requirements of the service on the execution environment including the description of the interconnections and their individual QoS demands. The level of abstraction of this description should be as high as possible to ease its creation, while still allowing for automatic matchmaking. In particular, the creation of the description shall not require special knowledge about the network infrastructure. This description has to be delivered to the ISONI in form of a Virtual Service Network (VSN) description.

As described in the ISONI Whitepaper [1] and chapter 6.2.2 of D3.1.1 ("Preliminary version of IRMOS Overall Architecture") [2], a network service is typically composed of multiple software modules deployed on several machines. The VSN description capsules one or more Application Service Components and their configuration in a vertex description, i.e. an ISONI Service Component description. For each ISONI Service Component description, a concrete realization is deduced. A typical realization is a Virtual Machine Unit (VMU). The ISONI management assigns resources, schedules, deploys, and interconnects these VMUs. The concept of the VSN description is explained in chapter 4.2.1 of this document. Virtual interconnections and the applied addressing scheme for service networks running on the ISONI is further detailed in chapters 4.2.2f. Chapter 4.3 covers the exposure of the services to other networks. A more detailed introduction in the ISONI concept is given in [1].

3.1. ISONI composite structure

For proper resource assignment, an accurate view of the current usage of all these resources, components, and connections is necessary. Due to scalability reasons, the resource management is hierarchically organized and decentralized wherever possible. Figure 1 shows the three hierarchical levels of ISONI, i.e., Physical Host level, Node level, and Domain level that exhibit fundamental architectural differences. The architectural levels are described from bottom to top in the following sections.



ISONI Domain (Virtualization of Networks)
• Spanning geographically separated, not „fully meshed" sites
• Providing virtual overlay networks for all applications
PSTN
Access Network
Best Effort Internet
ISONI Domain (on top of IP core)
Other ISONI Domain
Node
„fully meshed" ISONI Entities (non-blocking)
• no QoS issues
Node
Physical host (PH)
Processing Unit
• Hardware virtualization treated by available Solutions like KVM™
PH (CPU)
PH (CPU)
PH (CPU)
PH (CPU)
PH (CPU)
PH (CPU)
PH
PH
PH
PH
PH
ATCA
PH
PH
PH
PH
PH
ATCA
LAN

Figure 1 ISONI layers

3.2. Physical Host level

The individual computational units in an ISONI are called Physical Host (PH). These basic hardware building blocks are subject to ISONI management and deployment control. The computational resources and the network connections of a PH are virtualized to be able to participate in deployment and interconnection of VMUs. Generic hardware, such as x86 servers, can be used as PH. In larger installations, physical hosts might be organized in rack-mounted architectures, e.g. the Advanced Telecommunications Computing Architecture (ATCA), where individual blades are stacked in general purpose shelves. Each processing board of an ATCA shelf represents a PH in the ISONI. Processing boards can range from common off-the-shelf single-board computers to individually manufactured digital signal processor boards.

For collaboration with the ISONI, a Physical Host contains one or more network interfaces. The distinguishing network feature of an ISONI Physical Host is the IXB functionality that enables the virtualization of the network resources and controls the network interfaces according to the Physical Host’s role in the deployment of Virtual Service Networks. The general IXB concept is explained in chapter 6.1, while the functionality of an IXB on a Physical Host is illustrated in section 6.1.1 of this document.

In order to participate in deployment assignments, capacity reporting and connectivity configuration, a Physical Host requires at least connectivity to its peers in an ISONI Node - a managed group of Physical Hosts.

Physical Hosts are not only used to provide resources to a customer, but also to implement ISONI-internal functionality, e.g., the IXB functionality.

3.3. Node level

In the ISONI context, a node consists of a number of Physical Hosts. The node serves the ISONI as a group of resources for which the operation and maintenance, management and configuration is performed by superposed node control functions.

Within one ISONI Node, all Physical Hosts are interconnected by a high performance network, e.g. the backplane in an ATCA shelf or high-speed Ethernet technology. These technologies allow only short ranges but are comparatively inexpensive to deploy. The Physical Hosts share at least one fully-meshed network topology to setup direct node-internal connections for network virtualization. Due to the high-performance interconnection in a small physical area, it is assumed that network QoS is no issue within an ISONI Node.

An ISONI Node can be connected to private or public networks, e.g. managed IP networks, VPN-like resources, the Internet, etc. At the node edge, a so-called IXB_{NODE} provides the connectivity to node-external endpoints. For example, when a VSN spans multiple ISONI Nodes, the respective IXB_{NODE}s provide the connectivity across the node edges. If a VSN requires external access to, e.g. the Internet, the IXB_{NODE} mediates between the virtual address namespace and the Internet address space. All scenarios involving the IXB_{NODE} are covered in sections 5.3 to 5.6.

Ensuring the network QoS that is guaranteed towards VSNs through technical SLAs by simple overprovisioning cannot be regarded efficient beyond the Node level, so one task of the IXB_{NODE} is to employ resource admission policies, i.e., perform scheduling and traffic shaping for intra-Node traffic and traffic exchanged with the Internet.

Towards the outside world, the internals of an ISONI provider's infrastructure shall be hidden by the IXB_{NODE}. Additionally, the IXB_{NODE} needs to be able to timely reconfigure connectivity when parts of a VSN are migrated during runtime. The different tasks that an IXB_{NODE} performs are detailed in chapter 6.1.2.

3.4. Domain level

The ISONI Domain consists of several ISONI Nodes that are governed by identical policies and that are managed by one ISONI provider. Usually, these nodes are geographically distributed over a wide area network (WAN).

In contrast to the physical network infrastructure inside an ISONI Node, no fully meshed high-performance network topology amongst the ISONI Nodes is available on ISONI Domain level. Furthermore, the available links between ISONI Nodes might differ in capacity, functionality, and performance. In order to establish connectivity between two ISONI Nodes on the domain level, relay of connections or traversal of an additional node might be required while still maintaining the requirements of the VSNs.

As of the peering points towards other networks, such as other ISONI Domains, specific Autonomous Systems (AS) of the Internet, Public Switched Telephony Network (PSTN) or privately owned networks, the domain level coordinates the connectivity between a VSN and external networks. At the domain level, an (aggregated) view on the available peering points at the ISONI Domain edge is required to preselect and organize external connectivity.

3.5. Inter ISONI Domain resource collaboration

Peering with another ISONI Domain offers the opportunity of resource collaboration amongst ISONI Domains. In contrast to connectivity across the Internet and routed transit networks, resource collaboration would empower a single ISONI Domain to preserve a VSN's private address space when locating particular ISONI Service Components on ISONI Nodes of another domain. If an ISONI Service Component cannot be hosted due to the need for specialized resources which are not available within the domain of the inquired ISONI provider or due to a general resource shortage[1] in the ISONI Provider's domain, resource collaboration with other ISONI Domains is desirable to satisfy the functional requirements or resource needs of a Virtual Service Network.

As described in [2], IRMOS Framework Services are capable to discover ISONI providers and negotiate with multiple ISONI providers. The outcome of the negotiations might be that an application cannot be hosted completely by a single ISONI Domain and hence needs to be split into separate VSNs that would be independently run by different ISONI Domains. This case does not cover resource collaboration among ISONI Domains, because the ISONI providers would remain without notion of the separated requests. Consequently, the VSN would not be spanned across domains transparently, i.e., the ISONI Service Components would be in different namespaces and connectivity would be provided across other networks

[1] While non-available resources can also be considered a resource shortage, general resource shortage refers to situations where resources are available, but already in use.

where neither of the ISONI providers could guarantee network QoS. The resource collaboration instead would preserve the VSN address namespace in deployments across the domain boundaries and allow the negotiation of SLAs for all involved resources, including the network.

For resource collaboration with another ISONI Domains, dedicated gateway functionalities are deployed at each border of the related domains. As of the connectivity, this gateway has to interlink the ISONI Service Components across domain edges. Other tasks would also include brokerage of connectivity configurations, reporting of the network resource usage, supervision and so on. Resource collaboration requires a static agreement between the collaborating ISONI providers to enable usage of foreign resources. It is not created dynamically upon an individual service request, i.e. in contrast to the automated IRMOS technical SLA[2] agreement for VSN deployment, resource collaboration requires general agreements defining the coverage of collaboration between the operators. It can be considered a static expansion of an ISONI Domain. As laid out in in chapter 5.5, this kind of collaboration is not visible to an ISONI customer. The gateway's tasks are outlined in chapter 6.3 outlines the gateway tasks. A proof of concept implementation of this function in the IRMOS prototype is not foreseen.

[2] A technical SLA takes place between the ISONI provider and the IRMOS provider. It specifies the computing and communication resources offered by the former to the latter. Higher level parameters from the Application SLA are mapped down to low level and more specific parameters in the technical SLA.

4. ISONI Addressing Concept

4.1. Requirements for Addressing Scheme

As stated in chapter 3.1, there is a strict distinction between the environment that is seen by the ISONI Service Components in a VSN (and its developer) and the actual environment that is provided by ISONI.

One goal of ISONI is to ease the migration of legacy services to a Next Generation Service Oriented Infrastructure. One of the most tedious tasks in migration is to reconfigure address information in existing distributed services. Thus, the ISONI addressing scheme should pose as little restrictions on the creation of VSNs as possible, allowing the reuse of existing configurations, especially address configuration. The process of creating VSNs is performed by the *VSN Creator*. In IRMOS, the VSN Creator actually is a WP5 Framework Service process that translates the high level application description created by the Application Developer into a VSN.

As an ISONI Domain is developed to run a large amount of services concurrently, it is essential that crosstalk between services is inhibited by the platform for security reasons as well as the impact it might have on the Real-time performance of the deployed services. As legacy services are often already configured to use fixed IP addresses, the IRMOS Application Developer has to be allowed to use any address for any individual service and even if a different application developer is using the same addresses, the services will be kept separate from each other. In fact, interaction between applications is inhibited, unless the interaction is explicitly requested.

The concept of VSNs allows the VSN Creator to abstract from the concrete resource that will be used to execute an ISONI Service Component (and thus the correlating Application Service Component). Thus, the addresses used VSN-internally can be independent from the actual addresses used by the Physical Hosts. The ISONI concept also foresees that VMUs are relocated to different Physical Hosts during runtime in order to compensate host problems or to avoid resource fragmentation. Thus, the association between the addresses used in a VSN and those used on the Physical Host is not static, but can change over a VSN's lifetime. This change in association must be performed quickly enough in order not to violate the agreed SLAs that include connectivity between affected ISONI SCs.

4.2. ISONI Addressing Scheme

4.2.1. The VSN concept

In order to allow the ISONI to limit resource usage of a VSN while guarantueeing the availability of the required resources, the VSN developer needs to specify required interconnections between the ISONI Service Components in the VSN, which might lead to a fully meshed (virtual) network. The specification includes the required parameters of the interconnections. With this information, the ISONI is able to

- prevent unwanted communication ("crosstalk") between Service Components in a VSN
- prevent crosstalk between VSNs

Due to this isolation, the ISONI Service Components of a VSN can neither accidentally nor deliberately access components of another VSN. The VSN *as a whole* also cannot misbehave, i.e., consume more than the allocated resources, as both the link usage and the resource usage of computational resources are controlled by ISONI.

An example for a VSN is given in Figure 2. While the VSN is explicitly modelled by the VSN Creator, the mapping of links to connections between PHs and of ISONI SCs to concrete ISONI resources is handled by the ISONI. The mapping process is completely hidden from the VSN Creator.

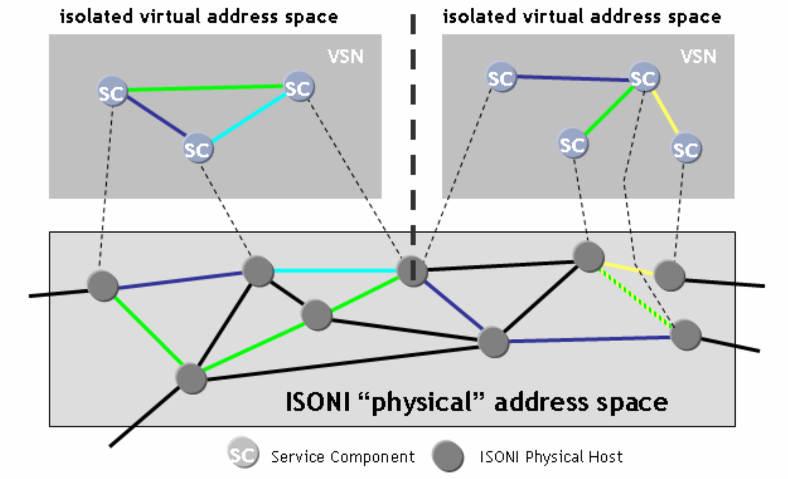

isolated virtual address space
isolated virtual address space
VSN
VSN
SC
ISONI "physical" address space
Service Component
ISONI Physical Host

Figure 2 VSN virtual address space isolation

4.2.2. Private Address space for service isolation

To achieve the goals laid out in section 4.2, the ISONI dynamically assigns a namespace to each VSN upon VSN instantiation. Although the actual namespace identifier is not known to the VSN Creator, the namespace concept creates a private address space per VSN, as the ISONI internally uses a combination of namespace identifier and VSN-internal IP address to guarantuee uniqueness and allow for deterministic routing.

4.2.3. Tri-layered addressing scheme

To fulfil the requirements laid out in the previous sections, a tri-layered addressing scheme has been developed to provide a flexible connection between the addresses assigned to the ISONI SCs and the actual hardware that is used to run that SC.

Virtual Address Layer
The address layer that is used inside the VSN is called *Virtual Address Layer*. As described earlier in this document, addresses at this layer are not required to be globally unique, as they are only considered in the context of the (unique) namespace that is assigned to the VSN by the ISONI.

From a conceptual point of view, any arbitrary layer 3 protocol could be used for ISONI SC interconnection in the VSN at the Virtual Address Layer. As the ISONI has to have a basic understanding of the addressing concept used in a VSN in order to perform routing decisions, the ISONI uses an approach specifically based on IP. The VSN Creator has to specify the virtual addresses that will be used by the ISONI SCs in the VSN description.
Confinement to a single layer 3 protocol is also crucial for creating interconnections between VSNs. The interconnection mechanism is laid out in chapter 5.

Pool Element Address Layer
For each vertex in the VSN, the ISONI prepares an appropriate ISONI SC based on the VSN description by preparing a VMU which is loaded with the customer-supplied code. For scalability reasons, it might be desirable to realize an ISONI SC decentralized, i.e., as a cluster of Service Components that are accessible through a single (virtual) address. These cluster elements are assigned a unique and immutable *Pool Element Address* within the *Pool Element Address Layer*.

The mapping from virtual addresses to Pool Element Addresses is performed by the ISONI. It is foreseen to use this mechanism to provide a framework for service pooling. Currently, the individual ISONI SCs are not aware of the Pool Element Address Layer, so that the mapping from virtual addresses to pool element addresses is a completely ISONI-internal process. As stated earlier, ISONI SCs may be migrated to a different PH during runtime. In that case, the Pool Element Address is not changed, as it is bound to the concrete ISONI SC and thus the VMU. The migration process can only be used if the error occurs below the VMU layer, i.e., to overcome problems with the network or the physical host. This process can also be used to overcome resource shortages by defragmenting resource usage. Errors on application layer, however, cannot be resolved by the migration mechanism as the error condition would also be migrated. In this case, it is foreseen to support failover on application layer by switching over to a different ISONI SC in the above-mentioned cluster. In this case, the Pool Element Address and thus the association between Virtual Address and Pool Element Address would change.

Physical Address Layer
To keep the technical SLAs negotiated with its customers, ISONI might decide to migrate running ISONI Service Components between Physical Hosts. For this reason, the Pool Element Address as introduced above is not suitable as identifier on the ISONI transport layer and ISONI assigns a *Physical Address* to each ISONI SC, i.e., the VMU that is created as its concrete realization. This address is bound to the PH the ISONI SC actually runs on.

As a result, the Pool-Element-Address-Physical-Address-binding is updated when an ISONI SC is migrated to a different PH. This mechanism enables the IXBs to correctly route packets between ISONI SCs, even when they are migrated to new PHs at runtime.

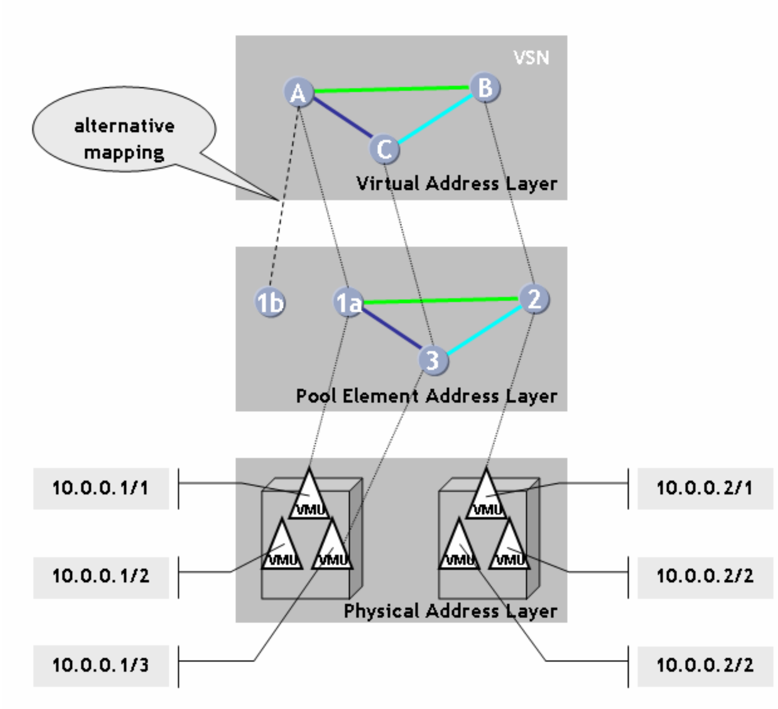


Figure 3 Mapping of address layers

An example of the described address mapping is depicted in Figure 3. All ISONI SCs A..C in the VSN are mapped to concrete realizations, i.e., a concrete VMU configuration consisting of operating system, libraries, and Application Service Components, is chosen. Alternatively, ISONI could select concrete realizations depending on dedicated hardware like DSPs. Should an application-layer error occur, ISONI could fail over to another realization. As this would require the different VMUs to be synchronized on application layer, application support is a prerequisite to this process. For execution, each VMU is deployed to a PH. The VMUs are enumerated on the PHs. This can be visualized as each PH having “execution slots” which have an address on the Physical Address Layer.

4.3. External access to a VSN

Primary goal is to make services deployed to the ISONI platform accessible from the outside world, i.e., the Internet or other VSNs. Secondary goal is to enable such services to access services that are available in the outside world. Both goals are to be achieved without losing performance compared to “flat” approaches, i.e., solutions without sophisticated ISONI features such as service isolation and service and network virtualization.
The process of adding external connectivity to a VSN must be consistent with the concept of isolated VSNs.

As the VSN concept is very flexible and allows the specification of advanced access features, such as Network Address Translation (NAT), load balancing, traffic splitting or proxying by including appropriate ISONI SCs, external access to a VSN can be reduced to a solution that provides a 1:1 connection between a publicly available endpoint and an ISONI SC.

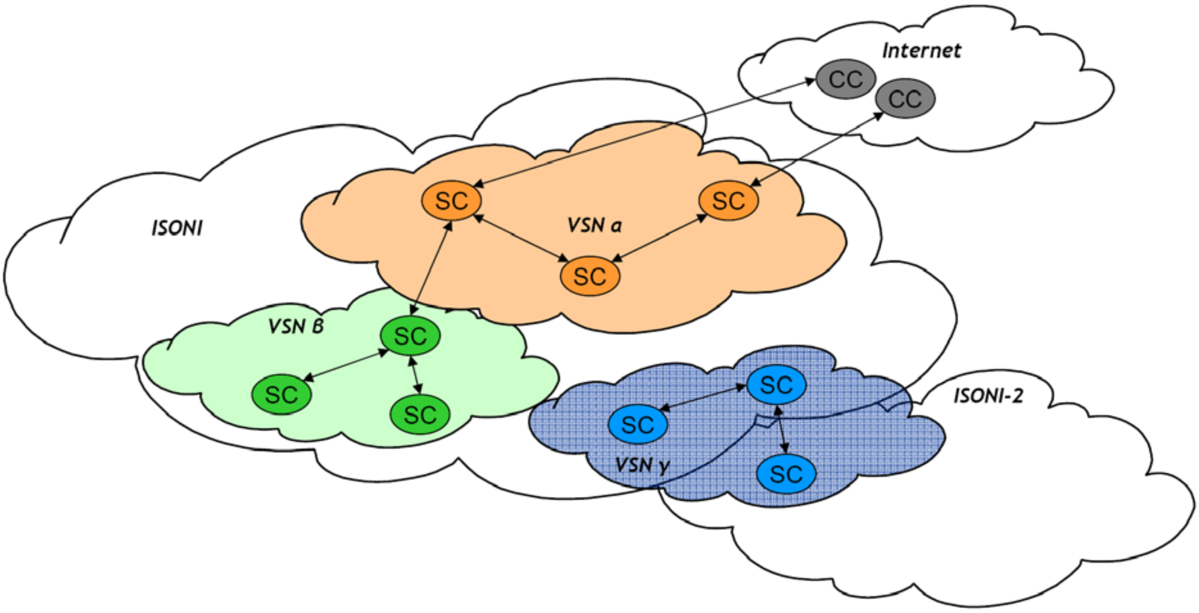


Figure 4 VSN Interaction at runtime

The runtime interaction between different VSNs and the outside world is depicted in Figure 4. Interaction is limited to the communication between well-defined ISONI SCs that proxy between two VSNs (VSN α and VSN β in the example) or a VSN (here: VSN α) and the Internet. If a single VSN spans multiple ISONI domains, like VSN γ in Figure 4, this is completely transparent to the ISONI SCs running inside of that VSN. It is also completely transparent to the ISONI customer, who makes the contracts with a single ISONI Domain operator, who has the sole responsibility of adhering to the negotiated technical SLAs. As this interoperability does not affect ISONI's interfaces towards its customer and does not add any immediate benefit, the actual development of such mechanisms might be investigated in subsequent projects.

The following sections describe how the interactions between VSNs are modelled and how ISONI derives the runtime information.

4.3.1. Access from outside ISONI to a VSN

As all ISONI SCs of a VSN run in a dedicated namespace, they are by definition isolated from the Internet, i.e., it is neither possible to access remote services on the Internet nor can customers (i.e., Application Client Components (ACC) from the application and Framework Services layers' point-of-view) connect to ISONI SCs inside the VSN by default.

If a VSN is to be accessed from the outside world, i.e., the Internet, a reference to an external IP address is added to an ISONI Service Component's definition in the VSN. The IP address might either be a placeholder for a dynamically assigned IP address or a fixed IP address in the Internet's Address Space. Effectively, the ISONI Service Component has access to both the VSN namespace and the Global Address Space, as traffic from the Global Address Space that is targeted at it is tunnelled to it by the ISONI.

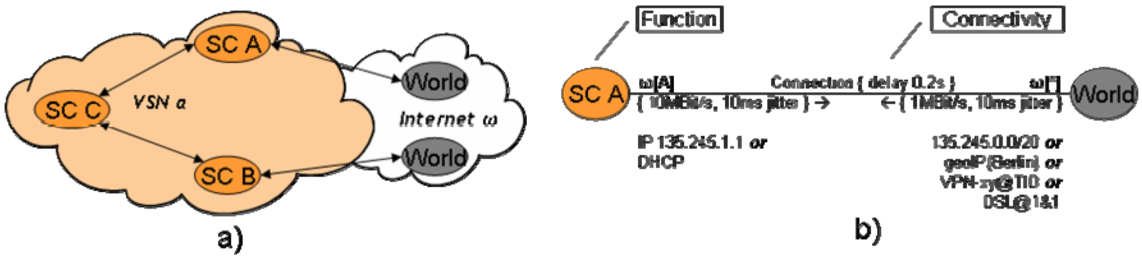


Figure 5 a) Modelling of access to external networks like the Internet b) Modelling of link between VSN and World

The modelling of access to external networks through World vertices is shown in Figure 5 a), while Figure 5 b) illustrates how the actual links between an ISONI SC and the World are modelled. Each ISONI SC that is accessible from the outside or that requires access to remote services is connected to a World vertex via a network link, whereas the World node is only needed for modelling purposes to terminate the second end of the link in the model. The link itself is basically modelled like a normal VSN link.

An ISONI Domain can only give QoS guarantees on links that it owns, i.e., links that are modelled in the VSN. In case of access to external networks, no guarantees on bandwidth, jitter, or delay can be given for networks beyond the ISONI Domain’s boundaries. Therefore, VSN links to World vertices are treated as requests how the VSN is to be connected to the ISONI provider’s backbone: For example, the VSN Creator can specify whether he wants the VSN to be connected to the backbone via 10MBit/s or 100MBit/s. This is similar to current common practice in server housing, where datacenter operators specify how they connect a server to their backbone, without giving guarantees on the QoS that actually will be delivered.

Optional attributes at the World side of the link include hints where the systems accessed via the link reside. This information is used by the ISONI to determine the proper ingress and egress locations for traffic between the VSN and systems that are located outside the ISONI Domain.

The ISONI SC that is connected to the World might be dedicated to providing the connectivity to a VSN, i.e., it acts like a proxy or performs network-level services like NAT, network-based load-balancing, i.e., by employing round-robin techniques, or Traffic Splitting. If such functionality is not required, a standard ISONI SC, that has an additional IP address in the World namespace, is connected to the World.

Multiple ISONI SCs of a VSN might be connected to the World. A single externally accessible ISONI SC might also be accessible through multiple IP addresses (but a single IP address always refers to a single ISONI SC).

To ease routing, the ISONI SC sets the source address of packets according to the virtual link the packet is to be sent over. Thus, VSN-internal traffic and traffic directed to the outside world can be easily distinguished by the $\mathrm{IXB}_{\mathrm{PH}}$.

4.3.2. Access to a VSN from another VSN

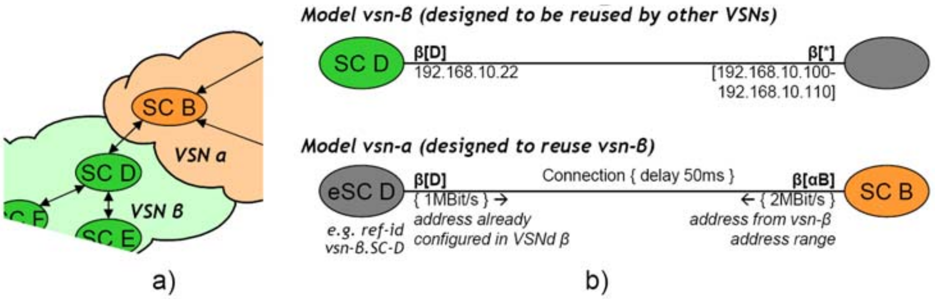


Figure 6 Access to a VSN from another VSN

Basically, the Internet can be considered as a large VSN. Thus, the mechanism presented in section 4.3.1 can be reused to access a VSN from another VSN.

A vertex that is actually provided by another VSN is called an *external Service Component (eSC)* which appears as a placeholder in the calling VSN's description. Instead of providing connectivity between an ISONI SC and the Internet, the ISONI provides a tunnel that connects the eSC to an ISONI SCs in the namespace of the included VSN.

The concept is depicted in Figure 6. In the reusable VSN, the VSN Creator connects the ISONI SC that handles external connections ("SC D" in this example) to a World vertex, thus specifying that the ISONI SC is accessible from other namespaces as shown in a). As illustrated in b), the range of addresses that is used to dynamically assign an IP address to the connecting VSN is also specified. In the including VSN, the embedded VSN is modelled as eSC which represents the externally accessible ISONI SC of the included VSN. Upon instantiation, it is replaced by the externally accessible ISONI SC. As in the VSN description one or more ISONI SCs are connected to the eSC, they can "break out" of their native namespace and access the eSC.

4.4. Summary in the context of application development

The previous sections are written from an ISONI point of view. This section aims at giving an insight at the internal processes from a bird's-eye perspective.

In the context of IRMOS, an 'application' along with its concrete parameters is finally translated into a VSN for execution within an ISONI Domain by the IRMOS Framework Services. After the mapping, including the mapping from application layer to infrastructure layer requirements ("30fps → 1GHz, 512MB"), the VSN is transferred to the ISONI Domain without further manual interaction. Therefore, the Framework Services act as the interface towards the ISONI and fulfil parts of the tasks assigned to the VSN Creator role, e.g., defining concrete network addresses for the service components.

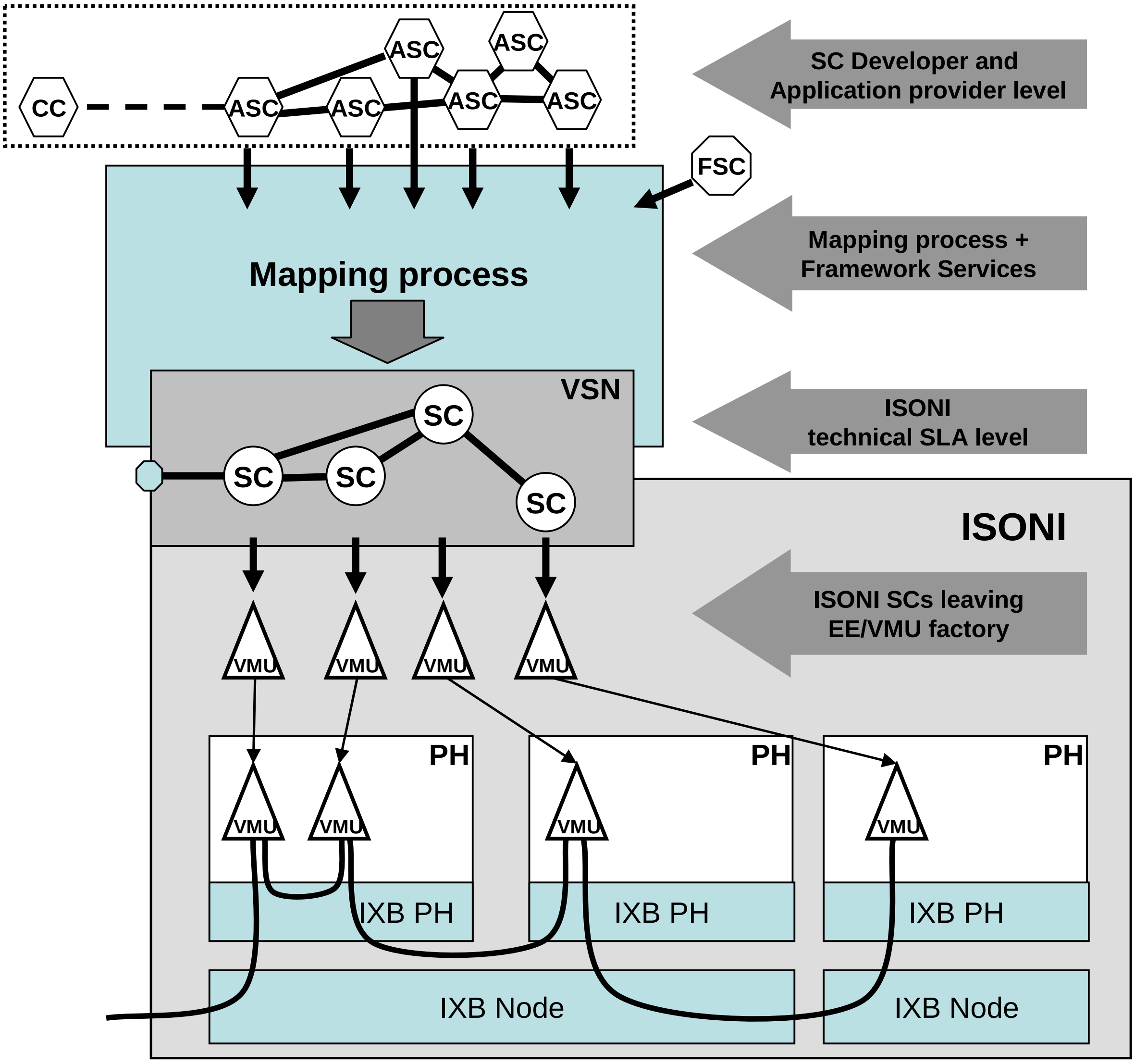
The more creative tasks were done before by the application developer who virtually designs templates for the application by interconnecting Application Components[3] (ACs) – without taking care of where the components actually will be running later on.

The application service components (ASC) are translated to ISONI SCs when the application as a whole is translated into an ISONI VSN. They need to be connected among themselves as well as to ACs not running inside ISONI, i.e., Application Client Components (ACCs) and External Application Service Components, (EASCs).

When creating these interconnections, the following items have to be considered:

- Connections between ASCs running entirely in the same VSN are completely abstracted from the underlying infrastructure so that arbitrary IP addresses may be defined. There is no need to take care of duplicate addresses as different VSNs are isolated by the ISONI. Network performance parameters must be defined and are maintained by the ISONI.
- Connections between ASCs running as ISONI SCs and components outside the ISONI (ACCs as well as EASCs; part of the World in ISONI terminology) can be realized by adding an external IP to the ISONI SC, i.e. to the VMU that is actually used to realize the ISONI SC and which contains the ASC. The connection is tunnelled by the ISONI. Network performance parameters must be defined, but of course they can only be guaranteed up to the ISONI network edge.
- If ISONI SCs are spread over *different VSNs* but within *one ISONI domain*, gateways have to be explicitly inserted in each VSN. From the inserting VSN's point of view, they behave exactly like a regular ISONI SC. The only difference is that the ISONI SC is not running natively in the VSNs namespace, but also belongs to another namespace. Therefore, such ISONI SCs are called *external Service Components (eSC).* Connections are handled as external connections, i.e. the ISONI SCs in the including VSN can only access the externally visible ISONI SC, i.e., the ISONI SC that is connected to the World vertex, of the included VSNs. In contrast to the external scenario, however, ISONI is able to guarantee QoS between the including VSN and the eSC.
- If ISONI SCs are spread over *different VSNs* that are deployed to *different ISONI Domains*, connections to the world have to be defined in both VSNs just like for connecting to an external service, i.e., a service on the Internet.
  As for connecting external ACCs, guaranteeing link QoS is not possible in this case, as the ISONI is not aware of the interaction performed over the Internet.

[3] Application Components (ACs) are the basic building blocks forming an distributed IRMOS application. While some ACs running outside ISONI (e.g. the client's GUI as an Application Client Component, ACC) the components acting as the service are called Application Service Components (ASCs). The latter will be mapped to ISONI SCs for execution.



CC
ASC
ASC
ASC
ASC
ASC
ASC
FSC
SC Developer and Application provider level
Mapping process
Mapping process + Framework Services
VSN
SC
SC
SC
SC
ISONI technical SLA level
ISONI
ISONI SCs leaving EE/VMU factory
VMU
VMU
VMU
VMU
PH
PH
PH
VMU
VMU
VMU
VMU
IXB PH
IXB PH
IXB PH
IXB Node
IXB Node

Figure 7 Correlation between application developer view and deployment on ISONI platform

An example deployment is depicted in Figure 7. On the highest level, the application developer models the application components (Application Client Components (ACCs) and Application Service Components (ASCs)). The IRMOS layer may add additional Framework Service Components (FSCs) and performs mapping downwards to a generic description of an ISONI VSN and related ISONI SC descriptions and code/binaries. Based on the ISONI SC descriptions plus code/binaries, the ISONI EE/VMU factory generates images of Virtual Machine Units (VMUs), which are ready to run on Physical Hosts (PH). Once the VMUs are deployed to a PH and started up, the IXBs have to take care of the connectivity between the VMUs, depending on the requirements as given in the VSN description.

5. Scenarios

To show that the addressing scheme presented in chapter 4 actually fulfils the requirements laid out in chapter 3.1, a representative VSN is described and the various options of deployment to Physical Hosts are derived. We will show that in each of these cases, ISONI can provide the requested connectivity by using the ISONI address concept. Without loss of generality, it is assumed that ISONI SCs in the VSN description are realized as VMUs.

5.1. VSN deployed to one physical host

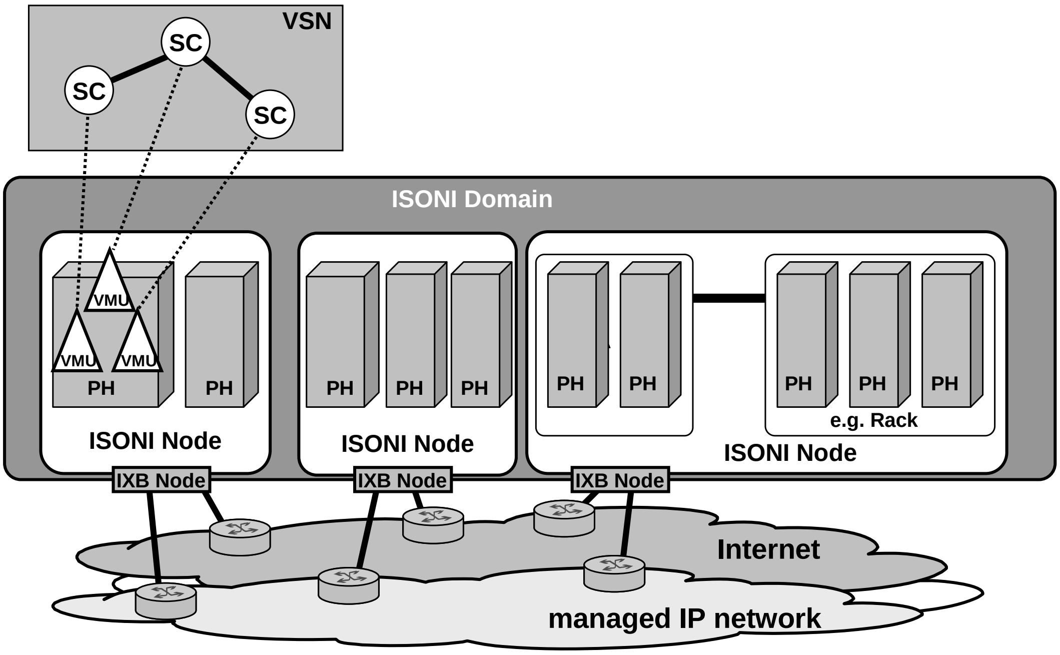


Figure 8 VSN deployment at one physical host (PH) and node

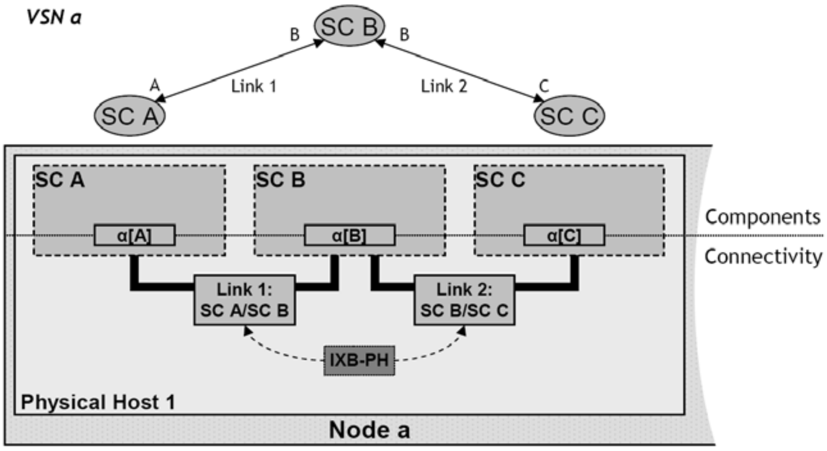


Figure 9 Correlation between links in the VSN and their realization by ISONI in intra-PH-scenario

The simplest case of a VSN deployment is to deploy all VMUs used to realize the ISONI SCs of a VSN to a single PH as depicted in Figure 8. This collocation has the advantage that network utilization is minimized, but increases the resource requirements for individual PHs.

Figure 9 shows an example VSN. Namespace α is assigned to this VSN by ISONI which is thus called VSN α. Upon instantiation, the addresses assigned to the links are associated to the corresponding interfaces of the VMUs. As depicted in the lower part of the figure, the ISONI is aware of the addresses A..C and internally always considers these addresses in the context of the VSN's namespace α. This is illustrated as α[A].. α[C].

As also shown in Figure 9, even intra-PH-traffic is handled by the IXB_{PH}. This is required to assure the ISONI isolation principles. Additionally, traffic handling by the IXB_{PH} allows a VMU to be transparently moved to a different PH, if required. Details on the migration mechanism are provided in chapter 7.

5.2. VSN deployed to one Node

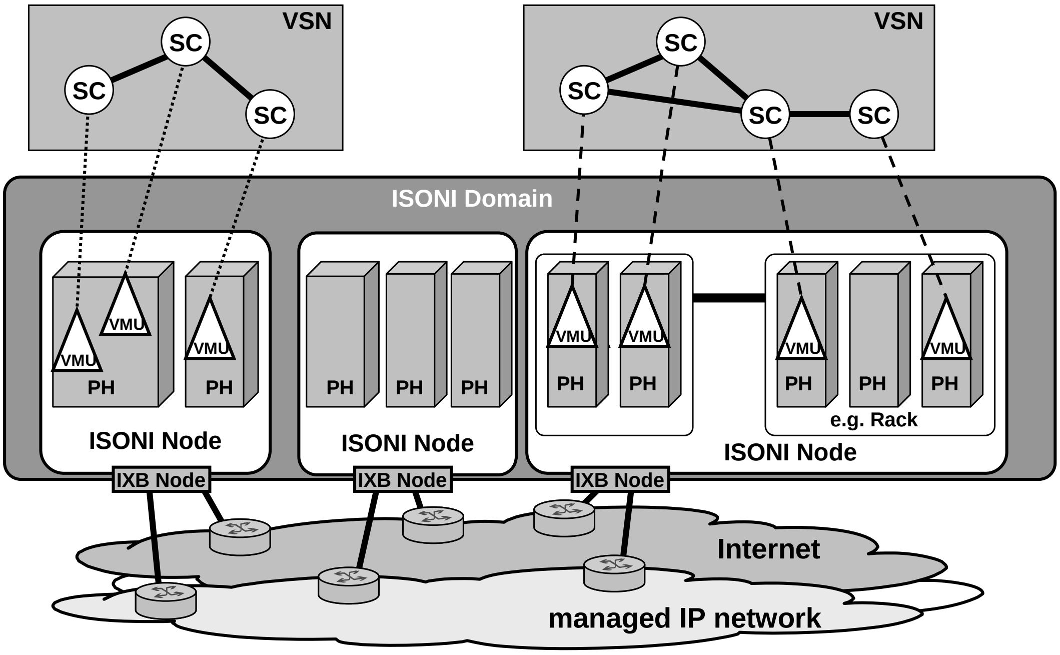


Figure 10 VSN deployed to multiple PHs inside a single Node

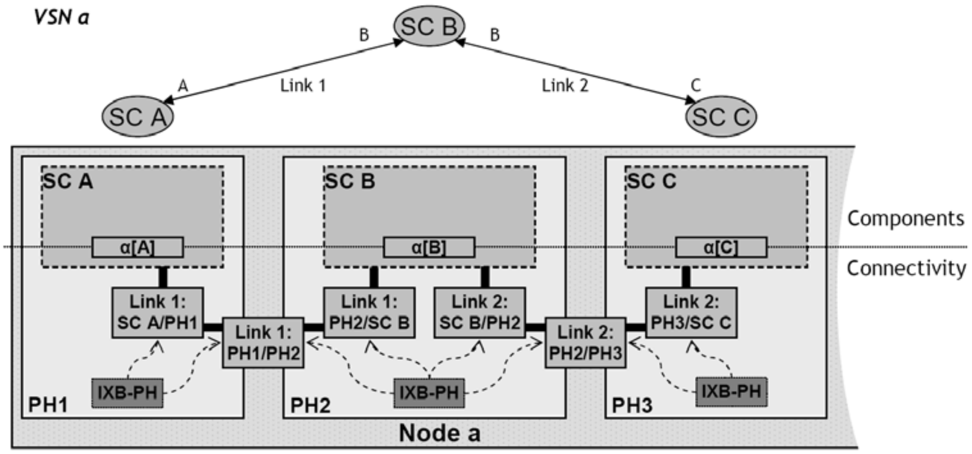


Figure 11 Correlation between links in the VSN and actual connections in intra-Node scenario

As stated in chapter 3.4, different Nodes inside an ISONI Domain are usually distributed both geographically and network wise. This results in higher delays and more expensive connectivity compared to the local network inside a Node. Unless there are special constraints, which will be addressed in the next section, ISONI prefers to deploy a VSN to a single Node. The decision whether the VSN is deployed to a single PH or a set of PH is delegated to Node level ISONI control.

An example of a VSN deployment to multiple PHs is given in Figure 10. As seen in Figure 11, connectivity between the VMUs is provided by connections between the hosting PHs that are controlled by the IXB_{PH} elements. Like in the previous section, the ISONI uses the namespace-address-tuples α[A]… α[C] for routing.

5.3. VSN deployed across Nodes, single domain

As suggested by the previous section, the ISONI can choose to distribute the ISONI SCs of a VSN among multiple Nodes. This might become necessary if the resources that are required to run a VSN are not available within a single Node or due to constraints given by the VSN on special geographical locations for certain ISONI SCs. From a global ISONI perspective, it might also be efficient to distribute ISONI SCs in order to minimize the overall bandwidth requirements. For example, if a lot of end customers are connected via the same Internet Service Provider (ISP), it might be sensible to perform initial processing close to the ingress point while the final processing, e.g., watermarking of video streams, is performed close to the egress point, for example at that ISP’s Points-of-Presence (PoPs). In this scenario, the processed stream would need to be transferred only once between ingress and egress.

The correlation between VSN and used resouces is depicted in Figure 12. As illustrated in Figure 13, intra-Node connectivity is provided analogues to 5.1 and 5.2 through IXB_{NODE} elements. While there is only a single IXB_{NODE} in this example, this is only the logical view. In the actual deployment, high availability mechanisms will be used to avoid the introduction of Single Point of Failures.

The advantage of this approach is that individual IXB_{PH}s do not need to be aware of the location of remote VMUs. Instead, they establish a connection to their Node’s IXB_{NODE}, who in turn creates a connection to the remote IXB_{NODE}. This approach is also beneficial if VMUs are migrated, as this is completely encapsulated inside the Node if a VMU is moved inside the Node and only remote IXB_{NODE}s need to be reconfigured if a VMU is moved to a new Node.

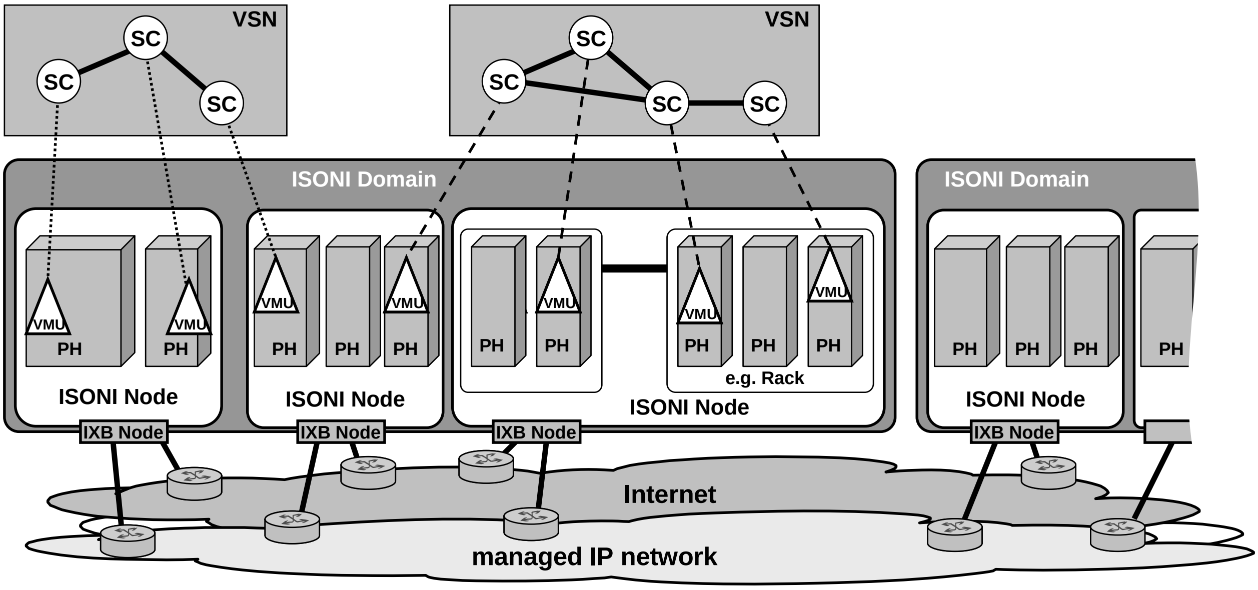


Figure 12 Node-spanning VSN deployment

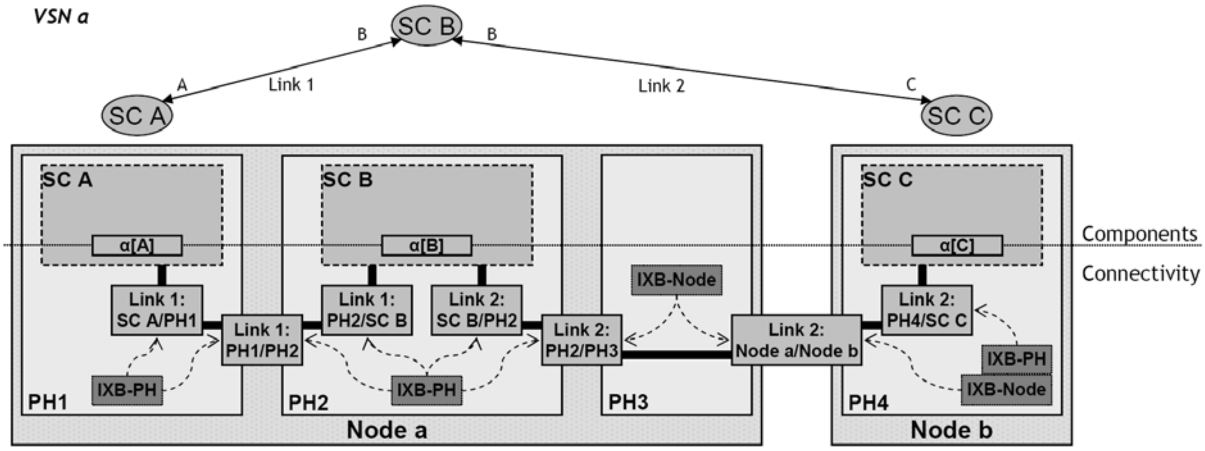


Figure 13 Correlation between VSN links and actual connections in inter-Node scenario

5.4. External access to a VSN

In chapter 4.3, the modelling of external access to a VSN has been described. Figure 14 shows how the interconnection of two VSNs and a VSN and the Internet is realized on the network level. First, internal connectivity for VSN α is established as described in the previous sections. As seen in the top part of the Figure, SC A is connected to both SC D in VSN β and the Internet ω, represented by the World vertex. SC D has address D in namespace β. To be able to access it, SC A is assigned an additional address A in that namespace. As described above, the ISONI is aware of both the virtual address of an ISONI SC and its namespace. Thus, crosstalk between VSN α and VSN β is only possible for SC A and SC D.

The World vertex represents the systems in the global namespace, i.e., the Internet. In order to be accessible from the Internet, a globally routable IP address is assigned to SC A, which is treated as ω[A] by the ISONI. If a packet directed to A is received from the Internet, ISONI forwards it to the ω[A] interface. On the other hand, packets sent through that interface by the ISONI SC are forwarded to the respective destination on the Internet.

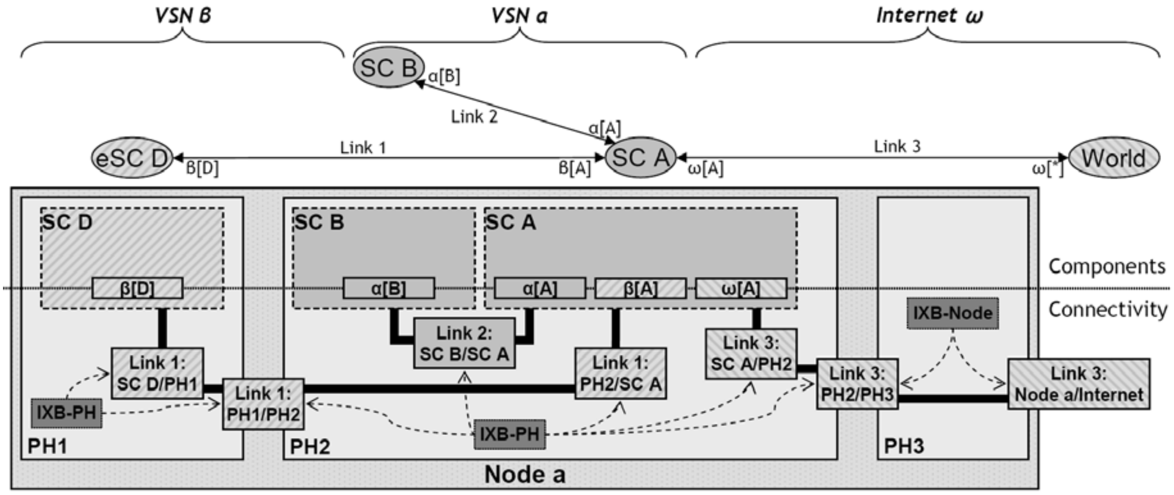


Figure 14 Cross-Namespace connectivity, realized by alien address assignment

5.5. Resource collaboration

A more general case for VSN resource allocation is the resource collaboration with other ISONI domains. In extraordinary cases like resource shortage or a request for special capabilities, an ISONI Domain may use resources from another ISONI Domain to be able to satisfy a VSN instantiation request. An interworking Gateway (iGW) on both sides ensures the controlled interworking of both ISONI Domains which is completely transparent to the ISONI Domain's customer, i.e., only one ISONI Domain appears towards the customer and therefore there is also only one partner for the technical SLA negotiation, etc.

If a customer does not want an ISONI Domain to outsource parts of the ISONI SCs, the VSN creator can specify an appropriate constraint in the VSN description.

The right hand side in Figure 15 vizualizes this interworking. As depicted in Figure 16, the VSN namespace remains coherent. As a static agreement has been signed by the involved ISONI Providers[4], the VSN behaves excactly as if it had been deployed to only a single domain. Furthermore, the process is completely transparent from a VSN developer's point of view. As stated above, all negotiation and contracting takes place with only a single ISONI Domain, who is completely responsible for keeping the negotiated technical SLAs.

[4] An ISONI provider operating multiple ISONI Domains might enact such agreements between its domains

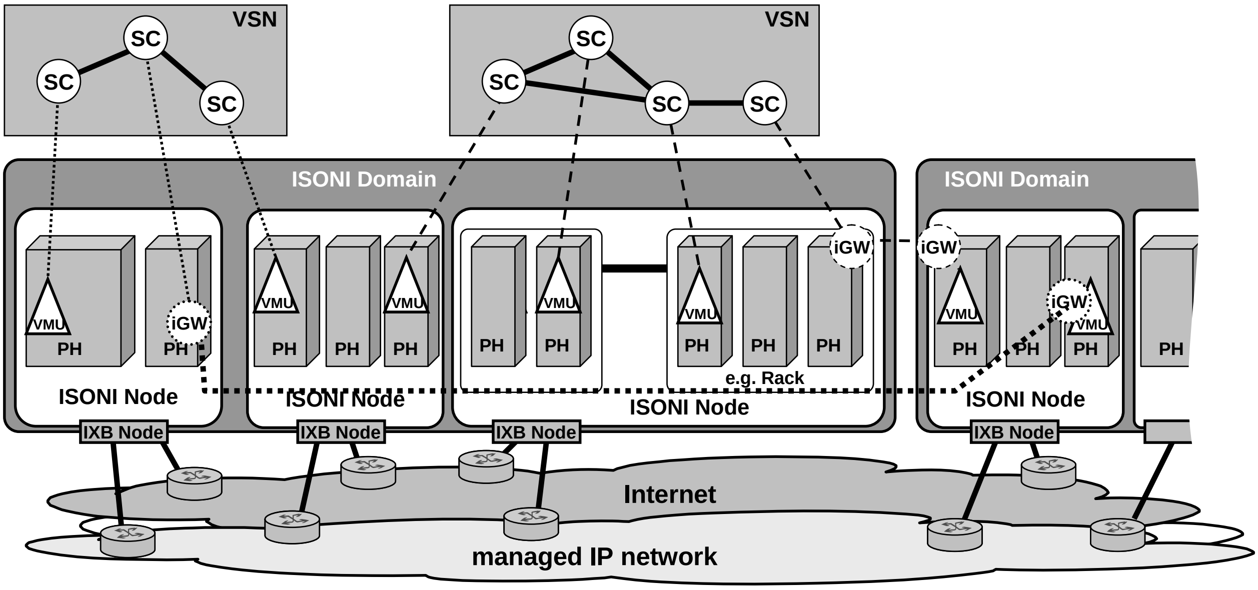


Figure 15 Resource collaboration used for VSN deployment

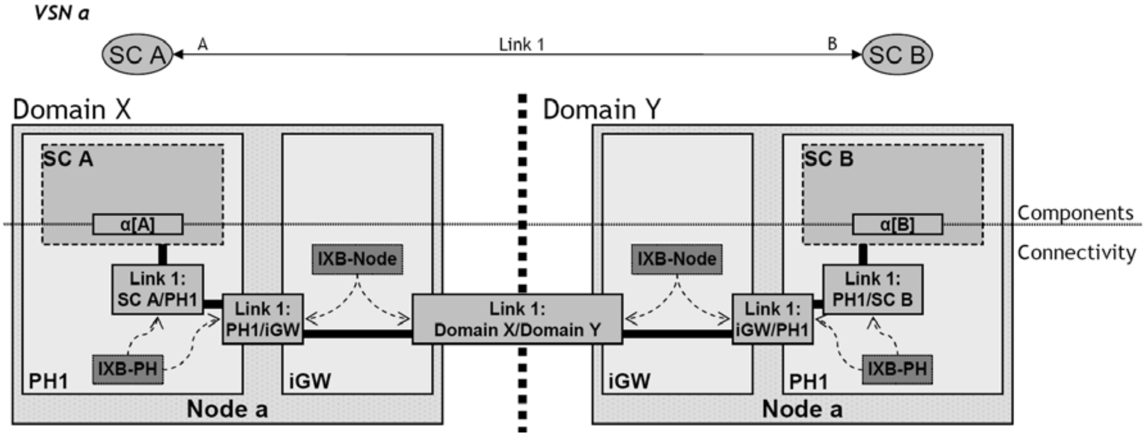


Figure 16 Actual connections for a VSN link spanning two domains with resource collaboration

5.6. Service segmented over multiple ISONI domains

As depicted in Figure 17, two distinct VSNs can be used to model interaction between WP5-level services. However, ISONI only sees the two VSNs connecting to the Internet. Although the ISONI SCs in the VSNs can access the other VSN's ISONI SC that is exposing its functionality, this process is not monitored by ISONI. As a consequence, no SLA can be offered for the connection or collaboration of these VSNs. It is also completely up to an external entity, i.e., WP5, to perform workflow management.

For this reasons, this case is out of scope and only presented for the sake of completeness. The approach is also not followed any further in future work on the ISONI layer.

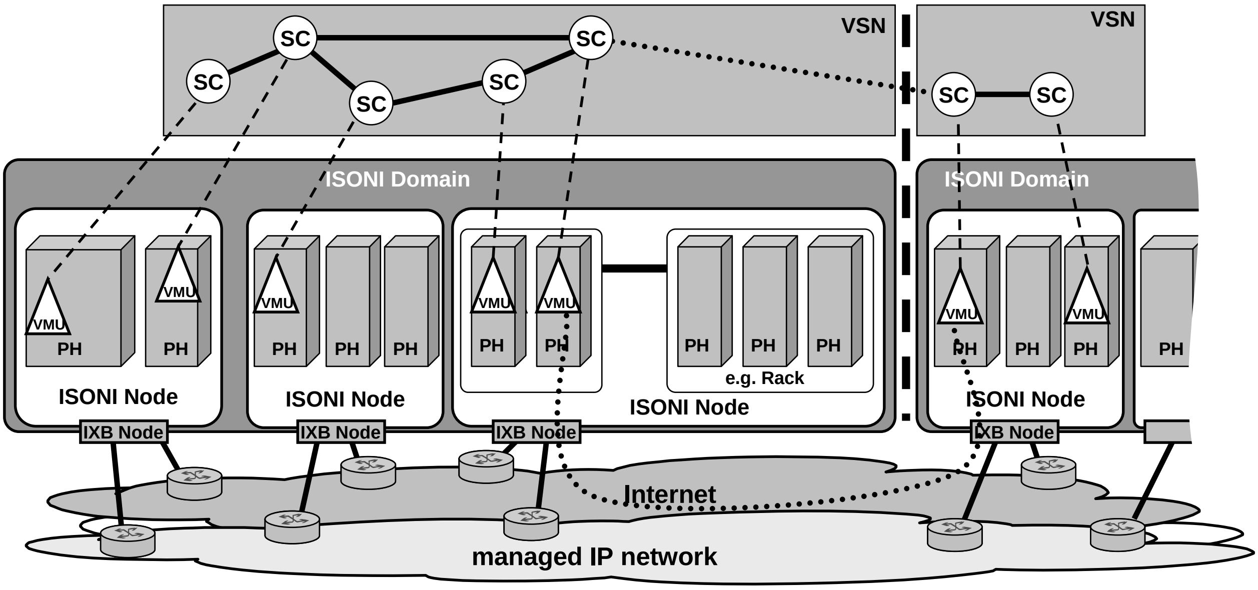


Figure 17 VSN deployment segmented over several ISONI domains

6. ISONI Addressing Architecture

6.1. ISONI eXchange Box

The VSN concept described in chapter 4 requires isolated connectivity and guaranteed QoS among the established ISONI SCs belonging to an individual VSN. The IXB is the functional element that is used to realize these requirements by creating overlays to resemble the virtual network topologies and preventing crosstalk between the overlays.
In the context of this work, ISONI SCs are assumed to always be realized as VMUs that host the appropriate software packages. Thus, the IXB interconnects running VMUs, which are the result of application service component deployment as described in chapter 4.4.

Each Physical Host (PH) is able to host one or more VMUs, depending on its performance and the VMU's resource requirements. Therefore, the IXB_{PH} is responsible to interleave the IP traffic of the running VMUs ensuring the required virtual link QoS as requested in the VSN description. The IXB as described in this deliverable is a fundamental part of the ISONI architecture. In the course of the project, it will be expanded to also perform part of the traffic management, which includes the enforcement of flow control. Measurement of link utilization and reporting on link health status will also performed by the IXB. The basic research on these topics is going to be conducted in the context of further deliverables.
This document focuses on the realization of connectivity between ISONI SCs and on connectivity to elements outside the ISONI Domain.

An IXB_{NODE} is configured by the Path Manager, and establishes VSN specific connections towards other IXB_{NODEs}. Effectively, the VSNs are spanned between the IXBs that perform routing, encapsulation, and decapsulation functions on the messages and data streams sent between the ISONI SCs. The IXB performs mapping of VSN virtual addresses of ISONI SCs to the physical addresses of the resources. Through this mapping, the physical topology of the network and the resource distribution is completely hidden from the VSN Developer as well as the services running on the ISONI. This concept also provides complete disjunction between VSNs, i.e. it prevents unauthorized information flows across VSN boundaries. It also prevents unwanted access to the ISONI SCs from outside networks, such as the Internet.

Physical Hosts are grouped together and under control of the node management. The scalability levels of domain and nodes result in an IXB_{PH} and IXB_{NODE} function.

If required by a policy, IXBs are able to encrypt traffic before it is sent to a remote IXB, for example if a path leads over un-trusted networks.

6.1.1. IXB Physical Host

The IXB Physical Host (IXB_{PH}) provides connectivity between ISONI SCs which belong to the same VSN. The VSN might be distributed over multiple PHs within a single node. For ISONI SCs co-located in the same node, the IXB_{PH} can establish this node-internal connectivity directly. Traffic which is targeted at an ISONI SC that runs on a PH that is located in a remote node is forwarded to the IXB_{NODE}.

As the network inside an ISONI Node can usually be considered secure, encryption between IXB_{PH} is usually not needed or only done to offload the IXB_{NODE}.

6.1.2. IXB Node

The IXB_{NODE} provides connectivity for node-spanning VSNs. Therefore it acts as mediation point for VSN traffic that would need to cross the Node's boundaries and is responsible for providing connectivity between Nodes.
Traffic received from an IXB_{PH} is forwarded to the remote IXB_{NODE} and vice-versa.

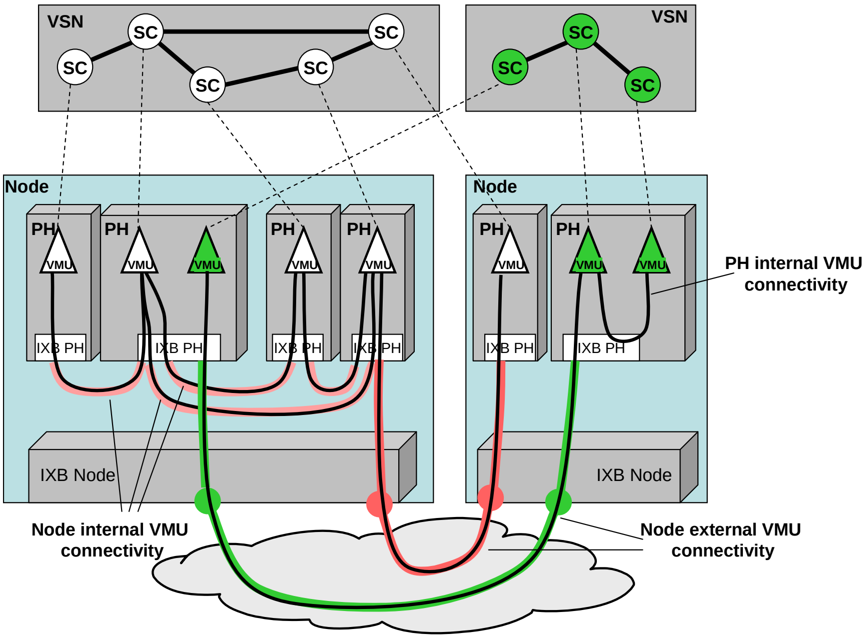

VSN
SC
VSN
Node
PH
VMU
IXB PH
IXB Node
Node
PH internal VMU connectivity
Node internal VMU connectivity
Node external VMU connectivity

Figure 18 Levels of IXB connectivity

The different levels of connectivity are depicted in Figure 18. PH-internal connectivity, intra-Node connectivity and inter-Node connectivity can coexist in the same VSN. Due to relocation of VMUs, the connectivity level might change over a VSN's lifetime.

6.2. External address binding

An ISONI Domain provider not acting as a Internet Servic Provider (ISP) is the simplest case dealing with public IP addresses for services running on ISONI. In this case, it has one or more exchange points where IP traffic is exchanged with the Internet. The public IP addresses that is assigned to an ISONI SC is "seen" at one of these exchange points from the Internet. Thus, these exchange points are called Point-of-Presence (PoP).

Another case could be that an ISONI Domain provider acts itself as an ISP for public IP addresses for hosted services. Also in this case, connection to the Internet – is provided over PoPs to the related ISONI SCs. The difference to the first case is, that the ISONI Domain provider can deal with the routing of public traffic in its domain in more flexible way, i.e., the location of the PoP can be chosen in a more flexible way.

In both cases, the PoP removes the namespace tag from packets targeted at the Internet, thus acting as egress point and adds namespace tags to incoming packets, thus acting as ingress point. Traffic between the PoP and the ISONI SC is transferred over a casual VSN link, i.e., the link connecting the SC to the World vertex.

The PoP functionality can be realized as either part of an IXB_{NODE} or it can be located on dedicated hardware acting as IXB.

Figure 19 shows the first case for two public IP addresses A and B. The PoPs for these two public IP addresses are located at one Node and encapsulate the IP packets before they forward them to the respective SCs, which are realized as VMUs in this example.

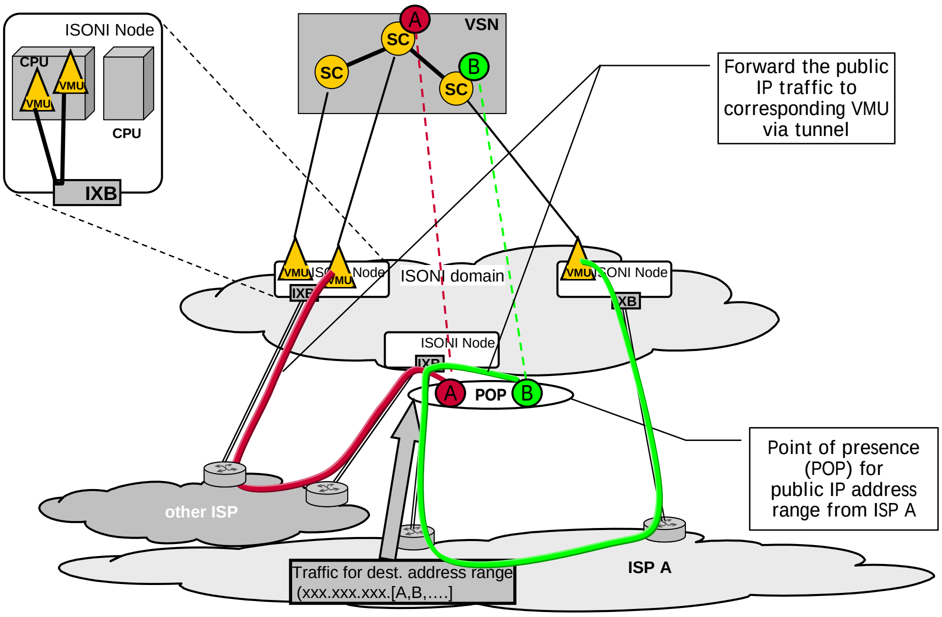

ISONI Node
CPU
VMU
VMU
CPU
IXB
VSN
A
SC
SC
B
SC
Forward the public IP traffic to corresponding VMU via tunnel
VMU
ISONI Node
VMU
IXB
ISONI domain
VMU
ISONI Node
IXB
ISONI Node
IXB
A
POP
B
Point of presence (POP) for public IP address range from ISP A
other ISP
Traffic for dest. address range (xxx.xxx.xxx.[A,B,….]
ISP A

Figure 19 Realization of public IP address binding

6.3. Inter-Domain Collaboration

The ISONI Inter-Domain Manger is responsible to manage all additional tasks that are needed to enable resource collaboration between ISONI Domains, which are

- Renting of Resources
- Resource Manager Cooperation
- Deployment Manager Cooperation

As these tasks will be covered in detail in D7.2.1, this document gives only a rough overview of the general tasks and focuses on the features related to connectivity.

Renting of complete Nodes of another ISONI Domain

Renting of complete Nodes from another Domain means the complete integration of the rented Node under management control of the renting Domain. This means that the rented Node behaves and participates in the communication as the other owned Nodes. Just the hardware is located in another Domain.

This results in a very strong coupling of the involved ISONI Domains in the sense of resource control beyond ISONI Domain borders. In this case, the other ISONI Domain acts as an equipment provider only.

Resource Manager Cooperation

As introduced [1], the ISONI has two management levels –Domain and Node level. The Resource Manager Cooperation task subordinates the rented Nodes under the Domain's managing responsibility. The Inter-Domain Managers relay the communication between Node and Domain level beyond ISONI domain borders. The relaying concerns the Node-related parts of the VSN request, resource availability, and monitoring.

During the lifetime of an instantiated VSN, interworking Gateway (iGW as introduced in 5.5) functionality is needed between related ISONI Domains to combine the different parts of a VSN so that it appears as one continous VSN address space.

Resource Manager Cooperation also results in strong coupling of ISONI Domains.

Deployment Manager Cooperation

An ISONI Domain can outsource parts of a VSN request to another ISONI Domain.

The Inter-Domain Manager initiates a technical SLA request for dedicated parts of the original VSN request towards another ISONI Domain. The collaborating ISONI Domains instantiate per-VSN Deployment Managers for the exported parts of the original VSN request.

If the exported part of the VSN request is instantiated successfully, all VSN-related Deployment Managers interact via the Inter-Domain Manager.

In contrast to the Resource Manager Cooperation, this task needs dedicated VSN Deployment Managers in the collaborating ISONI Domains.

Also in this case, an iGW is needed to combine the different parts of a VSN so that it appears as single continuous VSN address space.

The ISONI domain interworking Gateway (iGW)

The iGW is used in all collaboration cases as tie that connects the parts of a running VSN which split over different ISONI Domains. The iGW connects the continuous VSN address space transparently across ISONI Domain borders. The iGW provides a secure tunnel to the remote Domain's iGW and security features like firewalling and policing. The iGW ensures that access to the ISONI SCs of the VSN is restricted in accordance to the Virtual Link Descriptions.

If resource collaboration is performed, also ISONI Service Components that are made accessible from external networks like the Internet may be outsourced to a foreign ISONI Domain. In this case, the iGW also becomes responsible for forwarding the relevant public IP traffic between the ISONI Domains and thus guaranteeing seamless external connectivity.

7. Network support for live migration

As indicated in chapter 3, there are manifold reasons why the Physical Host that provides the resources for a Service Component ceases to provide resources. As the ISONI SCs are not run directly on bare metal hardware, but on a virtualization layer, the ISONI Service Component can be seamlessly moved to a "new" physical host. As long as this backup host is located in the same layer 2 network, providing seamless network connectivity can be considered state-of-the-art. However, cases where a whole Node or large parts of Nodes cease service also have to be considered.

The ISONI Addressing Scheme bases on providing a virtual layer 3 network to the compound service for the reasons laid out in section 4.2.3. Thus, it is sufficient to provide seamless connectivity at that layer. As shown in the previous sections, connectivity between ISONI Service Components is provided by IXBs that map the virtual links in the VSN to tunnels that are established on top of physical links. The seamless connectivity that is required for live migration is realized by real-time-tunnel adjustments.

This chapter links the ISONI Tri-Layer Addressing Scheme to the mechanism for seamless connectivity and shows that the migration of an ISONI SC does affect neither the remaining ISONI SCs in the VSN nor connections between an ISONI SC and the outside world.

In order to be transparent to the ISONI SCs, the migration process must not change the VSN-internal IP addresses, i.e., the virtual addresses. Instead, only the pool address or the physical address can be changed. The identity of the migrated ISONI SC does not change, i.e., the realizing VMU stays the same and is only moved to a new location. Thus, the ISONI SC's physical address is changed due to the migration.

As an effect of the migration, the mapping from VSN links to tunnels (which in turn are mapped onto physical links) becomes invalid. This is compensated by the Path Manager, who determines the required connections between the migrated ISONI SC and the remainder VSN and ensures that these tunnels are established between the respective IXBs. It also triggers the IXB to switch over to the tunnels to the ISONI SC's new location. To ensure a seamless transition, the new tunnels are established before the migration task is completed. A high level overview of this mechanism is depicted in Figure 20. In the case of Node-spanning VSNs, it has to be differentiated between the backup being in the same Node and the backup being in a different Node. If the backup is situated in the same Node, IXB reconfiguration is only required within that Node, as intra-node-traffic is always handled by the IXB_{NODE}. Consecutively, the IXB_{NODE}s of remote Nodes also have to be reconfigured when the backup PH is located in another Node.

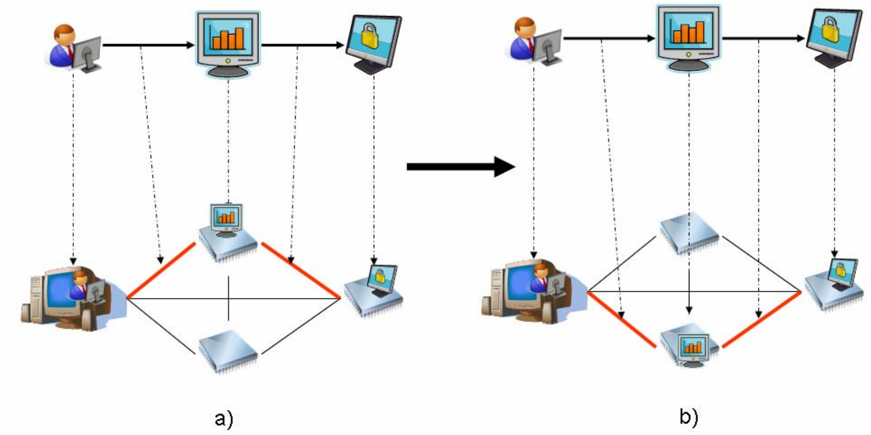


Figure 20 Principal tunnel reconfiguration process. Deployment of tunnels and VMUs a) before and b) after the migration

8. References

[1] IRMOS Project ISONI Whitepaper, ALUD and USTUTT, Sep 2008
Available on-line at http://www.irmosproject.eu/Publications/Default.aspx

[2] IRMOS Project D3.1.1 Preliminary version of IRMOS Overall Architecture, Aug 2008
Available on-line at http://www.irmosproject.eu/Deliverables/Default.aspx

[3] KVM Forum 2007 – KVM Para-Virtualized Guest Drivers, Dor Laor (Qumranet), Aug 2007
See http://kvm.qumranet.com/kvmwiki/KvmForum2007 (last accessed Nov 2008)

Annex A. ISONI management

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

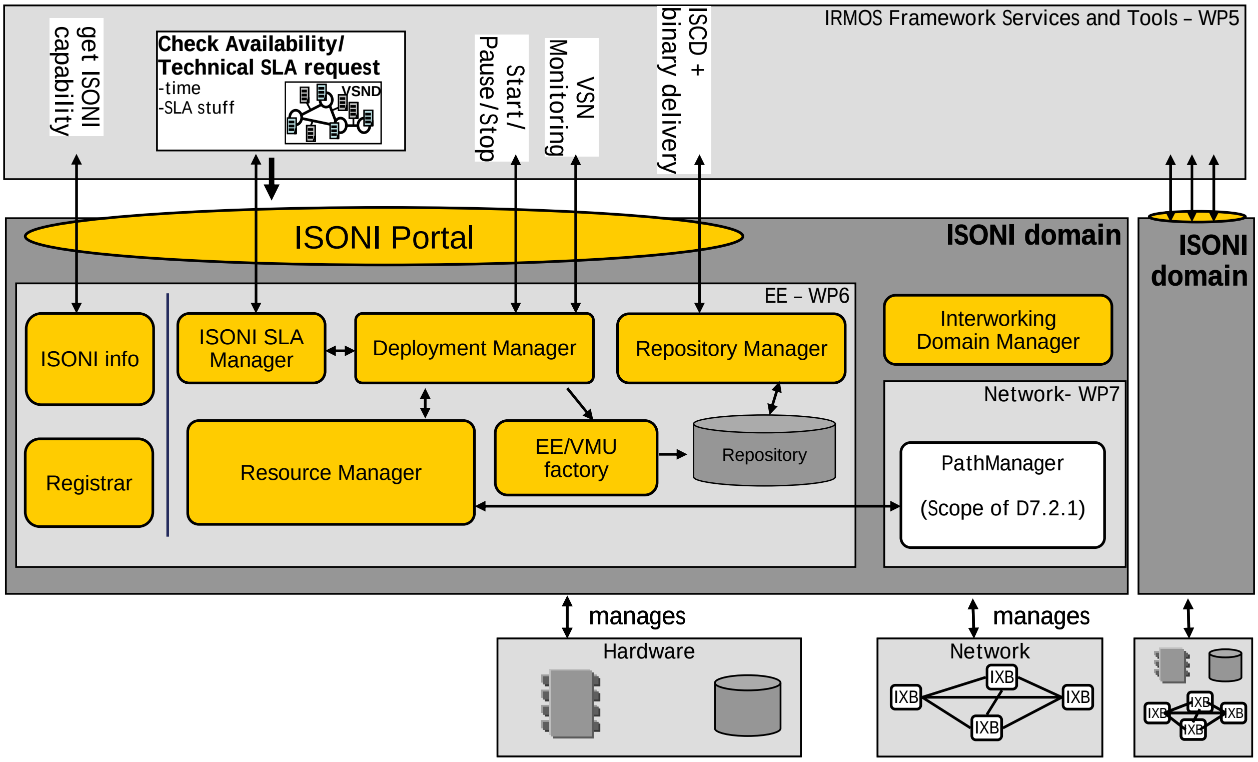


Figure 21 ISONI internal functional blocks

Figure 21 shows the functional building blocks of the ISONI management. The interfaces to the IRMOS Framework Services and Tools are indicated on top. The functional blocks Resource Manager and Path Manager would be deployed in a two-level management architecture based on the composite structure, i.e. Domain level and Node level. 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. As described in chapter 3.1 of the ISONI Whitepaper [1], the two-level structure has been developed for scalability reasons.

ISONI Info System

The ISONI info system provides information about general supported capabilities of ISONI towards the IRMOS Information Service. The reported ISONI capabilities assist the IRMOS SLA management in the ISONI provider selection process.

ISONI SLA Manager

The SLA Manager is responsible to negotiate technical SLA with the IRMOS Framework Services' technical SLA Management. If the technical SLA has been checked successfully, the SLA Manager launches the preparation and deployment by instantiating a VSN specific Deployment Manager instance.

Deployment Manager (per VSN deployment)
The Deployment Manager has the overall control to trigger resource selection and reservation, Execution Environment preparation and finally, to schedule deployment. Towards the IRMOS Framework Services, it provides execution monitoring and control.

Upon instantiation, the Deployment Manager triggers resource selection based on aggregated information on the Domain level. With the pre-selected ISONI Nodes at hand, it requests resource reservation while permuting through preferred deployment options on the selected Nodes. The control provided to the IRMOS Framework Services would cover e.g. pause/continuation and abortion of the VSN execution and monitoring controls.

Resource Manager
The Resource Manager occurs on both, domain and node level of the two-levelled ISONI management. On Domain level, the Resource Manager collects aggregated resource availability information. When resource selection is requested, it matches technical SLA requirements of the VSN description against the collected availability information and proposes ISONI Nodes, which are able to satisfy the requirements. On Node level, the Resource Manager manages the actual resources of Physical Hosts in the ISONI Node. Upon reservation request, it schedules resources reservation and deployment of the VSN sub-part that it has been assigned.

EE/VMU factory
The EE/VMU factory tailors the Execution Environment for the VSN, i.e. the factory creates the ISONI Service Components following the specification of the VSN Description and ISONI SC descriptions. The ISONI Service Components are then provided to the Nodes, ready to be deployed.

Path Manager
Like the Resource Manager, the Path Manager occurs on both, the domain and the node level of the ISONI. On domain level, the Path Manager matches network resource requirements of the VSN description against aggregated network availability information and network paths, which are able to satisfy the required network resources. On node level, the Path Manager reserves the requested network resources according to the VSN requirements.

Inter-Domain Manager
The Inter-Domain Manager is responsible to manage the additional tasks, which are needed to enable resource collaboration directly between ISONI Domains. In case of resource shortage or due to the need of special capabilities, an ISONI Domain may contemplate resource collaboration with another ISONI Domain. The Inter-Domain Manager is further described in chapter 6.3 of this document

ISONI Registrar
Any communication over the ISONI Portal and Inter-Domain Manager needs to be authenticated and authorized for any action. Therefore the Registrar keeps the information to identify external actors an ISONI system. And it keeps the related credentials like password, one-time tokens, and certificates. In respect to authorization it keeps the different levels of

privileges and/or restrictions. The scenario regarding Inter-Domain Managers is described in chapter 5.5 of this document.