ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

# MEDUSA

## DELIVERABLE
## D2.1.2 - Resource Management

medusa

Medical Distributed Utilization of Services & Applications

| | |
|---|---|
| Project number: | ITEA 10004 |
| Document version no.: | 1.0 |
| Edited by: | Stephane ZENG, “09 July 2014”<br>with the contribution of the WP2 partners |

**ITEA Roadmap domains:**
Major: Content & Knowledge

**ITEA Roadmap categories:**
Major: Interaction
Minor: Network & computing

## HISTORY

| Document version # | Date | Remarks |
|---|---|---|
| | | |
| V1.0 | 9-July-2014 | Final Version (Approved by PMT) |

TABLE OF CONTENTS

# 1 Introduction

The purpose of MEDUSA is to enhance the quality of diagnosis and decision making in acute and/or critical situations of a patient's condition by enabling image exchange, dynamic ('on-the-fly') advanced image processing and collaboration between remote physicians.

Medusa is built up from many components, and integrates with many systems. This document describes the top-level architecture of the system, focusing on the computing resource needs of Medusa. Based upon this architecture, the various subsystems will be worked out in detail.

Thus, a major challenge is to provide management solutions suited to the requirements of such system. This document intends to meet this challenge, whilst being in line with the architecture specifications.

# 2 Executive summary

IT which has become the central component of modern organizations is submitted to faster changes, with tighter requirements. This statement is also applicable to Healthcare field. Technically a lot has become possible. Along with all the techniques a lot of information is generated. Doctors and other involved caregivers are becoming more and more dependent on information generated during the treatment processes.

Medusa makes this information available on any location where the doctor and the patient are.

Medusa offers the following services:

- Image processing: no longer this is only available on a dedicated workstation in a hospital, but the processing power can be used from any location where the radiologist is working
- Collaboration environment: environment comparable to the well-known GotoMeeting-like applications from which the involved doctors can discuss the status of a patient and have access to all relevant information. Security, performance and high-availability are important aspects of the offered environment.
- Decision support: protocol guided treatment is becoming more and more common practice in healthcare. Decision support is continuously gathering and monitoring all the information available from a patient, and evaluating this information against the active protocol. By doing this, the decision support systems become more and more the eyes and ears of the involved doctors.
- When relevant, alerts will be shown to assist doctors in their treatment and in the decisions they take.

From architectural view, Medusa resources are divided into three main sets which are: the Client environment, the Cloud, and Medical Equipment.
This document defines Medusa resources and how they are managed.

# 3 Medusa Resources

For a better understanding of resources management, let’s first define what Medusa is. The services provided by Medusa rest on a set of material and software that can be classified as follows:

- **The client resources:** to access Medusa services, the client can use a desktop PC or a tablet laptop. These client devices have material and software resources that are important for their smooth working. A host of tools constituting the Medusa framework is also installed, thus insuring a secure access to decentralized medical services. Even if the client administrator manages these resources, the Medusa framework and the resources it uses, are under the responsibility of the Medusa system that offers a number of tools to this effect.
- **The Cloud resources:** The Medusa Cloud is deployed on the resources of the component of infrastructures divided in three categories: the units of nodes of calculations, the network, and the storage. The users have a toolbox that is based on modern technologies for the deployment, the configuration and the monitoring of these devices.
- **The medical equipment**: Although they are not directly included within MEDUSA, the MEDUSA system is compatible with medical equipment such as PACS, and microscopes as shown by various demonstrators. The system provides a minimal management of these resources according to the needs of the different service providers.

Subsequent chapters describe the management of these resources.

# 4 Client Environment

## 4.1 Devices

Client devices are the devices that can be used to connect to Medusa, to use one or more of its services: collaboration, image analysis or decision support. A Medusa client runs in browser environment. Theoretically any device running a browser can be used, but as Medusa shows a lot of information, low end devices like a smartphone are not suitable because of their limited display size. A tablet can be used for demo purposes, but in a production environment a laptop is the system that minimally meets the requirements.

The system requirements for a client device will be defined later in the project, taking into account the results of experiments that will be obtained. These requirements could concern:

- JavaScript and MPEG enabled devices (like IE9, Google Chrome, or Firefox browsers)
- Amount of RAM
- Kind of processor

Image Viewing and processing functionality (workstations) will be virtualized in the cloud. The cloud resourcing and SLAs will ensure that the needed computing resources will be made available to the Medusa clients. The Viewing applications will be made available via the IMT provided MPEG/HTML5 based virtualization technology.

## 4.2 Networking

Medusa allows collaborative work between physicians through distributed services hosted by the Cloud. Thus, a suitable network connection managed by legacy tools is required to support this secured collaboration.

The performance requirements are tested on a LAN with minimal 100Mb/s.

The system can be used on wireless networks but performance requirements cannot be guaranteed on these networks.

## 4.3 Security

As Medusa allow exchange of sensitive and private data, the transmission between the client and services hosted by the cloud needs to be secured. Thus, the client browser needs to support at least SSL to establish a secured connection (supported by all major browsers, Internet Explorer, Google Chrome, and Firefox).

However, for some critical access, we need to have higher security requirements. In that case, the client needs to support IPsec VPN connections (supported by all major operating systems such as android, windows, Linux …).

# 5 The Cloud Resources

The Medusa Cloud enables service providers to offer on-demand computing resources, to medical stakeholders, by provisioning and managing large networks of virtual machines. Computing resources are accessible via APIs for developers building cloud applications and via web interfaces for administrators and users. The compute architecture is designed to scale horizontally on standard hardware, enabling the cloud commercial companies have come to expect.

Medusa enhances deployment of applications and services over heterogeneous systems based on the PaaS layer and the underlying IaaS. The CompatibleOne PaaS solution offers an integrated user interface for monitoring the whole cloud, leaning on COTS technologies for monitoring Network, Storage and Computing resources.

## 5.1 Deployment and Provisioning

The Medusa cloud relies on CompatibleOne PaaS that is currently capable of performing fully automated deployment of cloud resources on the market leading open source cloud provisioning IaaS systems such as OpenStack and OpenNebula. Complementary work has been accomplished to use the Amazon EC2 platform in an interoperable fashion with the other open source and third-party cloud provisioning systems, the use of the Windows Azure cloud provisioning system, and the use of the Red Hat Cloud Exchange Protocol known as DeltaCloud.

The CompatibleOne Platform performs automatic deployment and interconnection of heterogeneous resources across heterogeneous cloud systems (from IaaS to PaaS and beyond). To cope with the underlying challenges, CompatibleOne qualified and adopted the Open Cloud Computing Interface (OCCI [3]) standardized by the Open Grid Forum organization. The OCCI specifications are used as the basis for designing and building CompatibleOne platform. Each operational server component of the platform is a standalone OCCI, REST, HTTP server responsible for the management of a specific collection of categories [1]. The entire CORDS (CompatibleOne Resource Description System) is compliant with this model. The logical view of CompatibleOne deployment model is shown in the figure below.

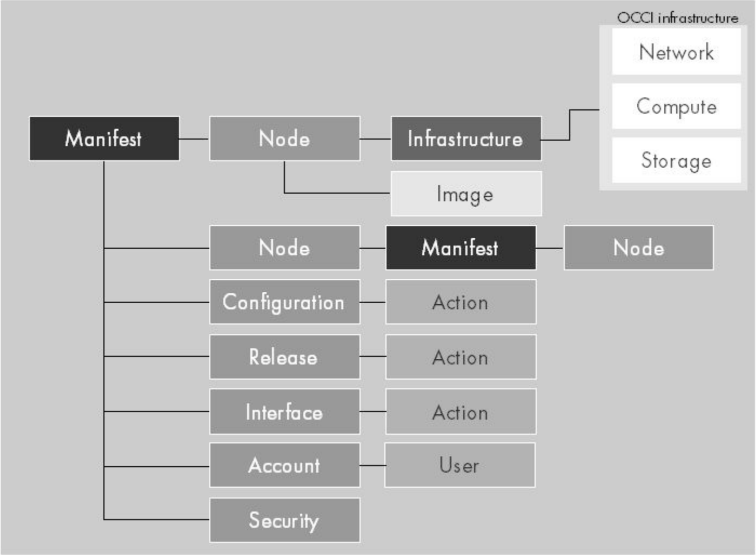

**Figure 1: CompatibleOne Deployment Model - logical view**

**Error! Reference source not found.** shows the logical view of the CORDS model, each of the boxes represents a Category and the Category is a resource. CompatibleOne's rationale is completely compliant with an OCCI interface and an OCCI specification model. In the OCCI protocol, there is a discovery capability that allows any client to discover the categories of services that these OCCI servers are capable to deliver. The illustration shows the logical descriptive view of how the CORDS Manifest is organized internally to describe required cloud resources (Node category) and their specific provisioning context (configuration, release, interface, account, security).

The Node category is used to represent the fundamental CORDS work unit. A node may be described solely in terms of its hardware infrastructure requirements and software image requirements. As the above figure shows, a node may also be described in terms of its specific manifest, in which case the infrastructure and image information will be omitted. To cope with multi-nodes provisioning, a CORDS Manifest may include as many Node elements as potentially needed.

The infrastructure category provides the description of the network, compute and storage requirements. The Image category is described in terms of its base operating system and the collection of required packages that should be installed to satisfy the desired application functionality.
The Configuration of the manifest nodes allows their interconnection as described by the associated collection of configuration actions. The manifest also requires description of the security conditions and the information concerning the account under which provisioning of the manifest may be performed. The manifest configuration section has also been extended to allow description of the way in which the provisioned resources are to be monitored during operational deployment for compliance with an eventual service level agreement.

## 5.2 Compute

The Medusa Cloud is architected to provide flexibility for designing medical services, with no proprietary hardware or software requirements and the ability to integrate with legacy systems and third party technologies. It is designed to manage and automate pools of computing resources and can work with widely available virtualization technologies, as well as bare metal and high-performance computing (HPC) configurations.

Administrators would deploy Medusa Cloud Compute using one of multiple supported hypervisors in a virtualized environment. KVM and XenServer are popular choices for hypervisor technology and recommended for most use cases. Linux container technology such as LXC is also supported for scenarios where users wish to minimize virtualization overhead and achieve greater efficiency and performance. In addition to different hypervisors, Medusa IaaS supports ARM and alternative hardware architectures.

Medusa also supports deployment on bare-metal computing nodes. The bare-metal cloud provides a way to complement or substitute virtualized cloud services with a dedicated server environment that eliminates the overhead of virtualization without sacrificing flexibility, scalability and efficiency.

Bare-metal cloud servers do not run a hypervisor, are not virtualized, but can still be delivered via a cloud-like service model.

This balances the scalability and automation of the virtualized cloud with the performance and speed of a dedicated server. The hardware is fully dedicated, including any additional storage. Bare-metal cloud instances can be provisioned and decommissioned via a web-based portal or API, providing access to high-performance dedicated servers on demand.

Also, depending on the application and use case, a single bare-metal cloud server can often support larger workloads than multiple, similarly sized VMs.

## 5.3 Network

Discovering and monitoring virtual resources, monitoring cloud operations and events and generating performance reports.

In order to face the complexity of modern IT systems, the Medusa Cloud provides a Networking component which is built on a pluggable, scalable and API-driven system for managing networks and IP addresses. Like other aspects of the cloud operating system, it can be used by administrators and users to increase the value of existing datacenter assets. Medusa Cloud Networking ensures the network will not be the bottleneck or limiting factor in a cloud deployment and gives users real self-service, even over their network configurations.

Leaning on IaaS layer, Medusa Cloud provides flexible networking models to suit the needs of different applications or user groups. Standard models include flat networks or VLANs for separation of servers and traffic.

The Networking Component manages IP addresses, allowing for dedicated static IPs or DHCP. Floating IPs allow traffic to be dynamically rerouted to any of your computing resources, which allows you to redirect traffic during maintenance or in the case of failure. Users can create their own networks, control traffic and connect servers and devices to one or more networks.

The pluggable backend architecture lets users take advantage of commodity gear or advanced networking services from supported vendors. Administrators can take advantage of software-defined networking (SDN) technology like OpenFlow to allow for high levels of multi-tenancy and massive scale.

The Networking Component has an extension framework allowing additional network services, such as intrusion detection systems (IDS), load balancing, firewalls and virtual private networks (VPN) to be deployed and managed.

## 5.4 Storage

Thanks to the IaaS layer, Medusa Cloud has support for both Object Storage and Block Storage, with many deployment options for each depending on the use case.

Object Storage is ideal for cost effective, scale-out storage. It provides a fully distributed, API-accessible storage platform that can be integrated directly into applications or used for backup, archiving and data retention. Block Storage allows block devices to be exposed and connected to compute instances for expanded storage, better performance and integration with enterprise storage platforms, such as NetApp, Nexenta and SolidFire.

**Object Storage**

Medusa Cloud provides redundant, scalable object storage using clusters of standardized servers capable of storing petabytes of data. Object Storage is not a traditional file system, but rather a distributed storage system for static data such as virtual machine images, photo storage (including DICOM images), email storage, backups and archives.

Having no central "brain" or master point of control provides greater scalability, redundancy and durability.

Objects and files are written to multiple disk drives spread throughout servers in the data center, with a software responsible for ensuring data replication and integrity across the cluster.

Storage clusters scale horizontally simply by adding new servers. If a server or hard drive fails, the content is replicated from other active nodes to new locations in the cluster. Thanks to the use of software logic to ensure data replication and distribution across different devices, inexpensive commodity hard drives and servers can be used in lieu of more expensive equipment.

**Block Storage**

Medusa Cloud also provides persistent block level storage devices for use with compute instances.

The block storage system manages the creation, attaching and detaching of the block devices to servers. Block storage volumes are fully integrated into Compute and the Dashboard components allowing for cloud users to manage their own storage needs. In addition to using simple Linux server storage, it has unified storage support for numerous storage platforms including Ceph, NetApp, Nexenta, SolidFire, and Zadara.

Block storage is appropriate for performance sensitive scenarios such as database storage, expandable file systems, or providing a server with access to raw block level storage.

Snapshot management provides powerful functionality for backing up data stored on block storage volumes. Snapshots can be restored or used to create a new block storage volume.

## 5.5 SLA Management

To handle the contractual Service Level Agreement (SLA) for MEDUSA cloud services, the CORDS [4] specification of CompatibleOne ACCORDS platform has been extended to provide the definition of the collection of categories required for the management of provisioning through service level agreements. This work concerning the SLA is based on the work performed by the Open Grid Forum organization and published in the form of the Web Service Agreement (WSA) Standard. The WSA specification was intended for use for the description of service level agreements for Web Services.

The CORDS specification aims at providing a service level agreement description and management model for Cloud services. The different constituent elements of an Agreement, as described by the WS-Agreement specification, have consequently been adapted for discrete and distributed operation as OCCI [3] category instances. The following diagram gives an overview of the structure of an Agreement.

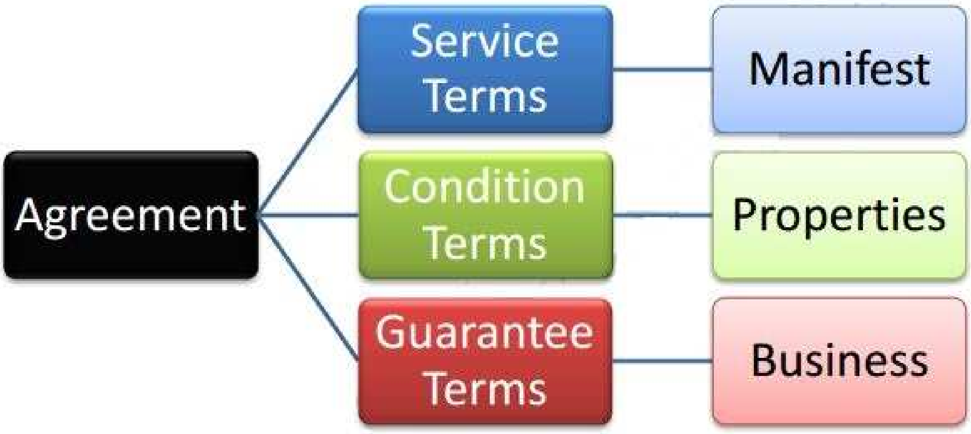

**Figure 2: Structure of ACCORDS Service level agreement**

The service level agreement describes and establishes the conditions under which the provisioning of service must occur for a particular MEDUSA service and for a particular customer of the corresponding ACCORDS platform operator. The service level agreement also describes the service to be deployed in terms of a CORDS manifest. The service conditions section allows the description of placement conditions and the guarantee section allows the definition of conditions that require monitoring and the eventual business values that they represent.

To automate a deployment of an application via ACCORDS platform, the concerned application should be well-suited for use within the cloud. From this point of view, three categories of constraints should be taken into account by the application architects:

- Compute, network and storage resources consumption: the application architect should be aware of the amount of resources the application requires.
- Application deployment process: the application to deploy and its dependencies should be designed with the capability to be installed in a completely automated manner, without requiring any user interaction.
- Application context configuration: the application architect should provide all the settings needed for the utilization of the application by the users. Following is a non exhaustive list of the requirements :

- Base operating system
- List of additional software dependencies
- Application configuration
- Pre-start application rules
- Security rules
- Account owner
- Application termination rules

The “condition terms” part of an SLA allows the description of workloads provisioning conditions. ACCORDS platform allows users to specify their preferences for the placement (localization) of their workloads.

In MEDUSA context we should consider the case where the data are located in a private data-center (hospitals or other medical organizations) and the applications would be provisioned in the cloud. The challenge here is how to provision a workload with the goal to honour the SLA contract, with the constraint that the data and the applications are in separated networks. There are two cases to consider:

- Applications that separate (or able to separate) intensive data access from data processing results output. For this kind of applications, it is recommended to run the application part that accesses the data and processes them on the data center and provision the other part of the application in the cloud. This architecture reduces drastically the needs in terms of network bandwidth.
- Applications that act as a whole, without separating data access, data processing and results output : for this kind of applications, ACCORDS platform should be able to find the most appropriate cloud provider and the most appropriate available zone offering the best conditions for executing the application, and the best network bandwidth between the data center and the application location. In many cases, the most appropriate cloud provider and geographic location are not necessary the “nearest” ones.

## 5.6 Cloud Resource Management

To orchestrate the allocation and deallocation of resources on the Cloud, a Cloud collaboration management platform (CCMP) is conceived to act as an intermediary between the Client environment and the Cloud resources. The CCMP translates Client requests into available hardware and software for application execution by invoking the CompatibleOne APIs. It also oversees the lifecycle of the deployed resources and links back the virtualized legacy applications to the Client environment.

## 5.7 Security Resources

In order to secure the cloud and built a secure context for medical services execution, MEDUSA provides a security component deployed through virtual appliances to protect medical services, using cloud network component. This security component is composed of 2 main elements: UTM and access control management system.

**Unified Threat Management (UTM)**

- Medusa provides a UTM protection to ensure a secure context for medical application, and protect cloud information resources. UTM brings many functions such as:Advanced applicative intrusion prevention system,
- Virus detection,
- Transmission protection with “easy-to-use” VPN,
- Secure access to cloud with mobile devices,
- Activity reporting for administrators,

This UTM is deployed in the cloud as a virtual machine, using the IaaS capabilities and the network component to protect MEDUSA cloud services. It also provides security report.

**Access control management system**

Medusa provides a complete access management system to allow unique authentication on all MEDUSA services. It provides 3 main functions:

- Authentication to validate a user identity on MEDUSA cloud
- Single Sign On to authenticate the user into MEDUSA services
- Access control to check user rights defined in the user management component before any access to each MEDUSA services

Those 2 elements, the Unified Threat Management and the access control management system, are fully integrated together to provide a complete security layer to the MEDUSA cloud.

# 6 Medical devices

Medical devices are not hosted in the cloud, and remain external to the Medusa system. Nevertheless, they are connected to the Medusa system.

## 6.1 Medical image storage

The Medical Image Storage in Medusa is coupled to the related Image Processing and Image Viewing applications. The related Image Storage nodes need to be able to be contacted by the hospital. E.g. a hospital PACS must at any time have the opportunity to send medical data to these storage nodes. As the PACS system is a hospital owned and managed entity inside the Medusa cloud only the connectivity to such a PACS needs to be managed. This will be part of the configuration of the storage nodes for the medical data and likely influence related SLAs on the availability of the storage nodes and the communication speed of the link between hospital PACS and Medusa storage nodes.

## 6.2 PACS integration security

Also for communication with a PACS in a hospital security is important. Such a communication can be secured via TLS and certificates. This implies that required certificates need to be installed on either the hospital PACS and on the image storage nodes handling the communication from the cloud to the hospital based PACS systems.

# 7 Conclusion

Resource management is a core function required of any man-made system. It affects the three basic criteria for system evaluation: performance, functionality and cost. Inefficient resource management has a direct negative effect on performance and cost. It can also indirectly affect system functionality. Some functions the system provides might become too expensive or ineffective due to poor performance.

Medusa Resource Management focuses on the client environment and the cloud management where Medusa services are deployed. Hence, in addition to device management, it targets the strategies for cloud resource management associated with the three cloud delivery models, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), in accordance with (Medusa-D2.1.3 2014) and all other Medusa specifications.

# 8 Glossary

| | |
|---|---|
| **API** | Application Programming Interface |
| **COTS** | Commercial Of The Shelf |
| **DICOM** | Digital imaging and communications in medicine |
| **DHCP** | Dynamic Host Configuration Protocol |
| **IaaS** | Infrastructure as a Service |
| **IE** | Internet Explorer |
| **IP** | Internet Protocol |
| **IPsec** | Internet Protocol Security |
| **KVM** | Kernel-based Virtual Machine |
| **LAN** | Local Area Network |
| **LXC** | Linux Containers |
| **MEDUSA** | Medical Distributed Utilization of Services and Applications |
| **OCCI** | Open Cloud Computing Interface |
| **OGF** | Open Grid Forum |
| **SaaS** | Software as a Service |
| **PaaS** | Platform as a Service |
| **PACS** | Picture archiving and communication system |
| **SDN** | Software-Defined Networking |
| **SLA** | Service Level Agreement |
| **TLS** | Transport Layer Security |
| **UTM** | Unified Threat Management |
| **VM** | Virtual Machine |
| **VPN** | Virtual Private Network |
| **WSA** | Web Service Agreement |

# 9 References

[1] Openstack web site: http://docs.openstack.org/

[2] CompatibleOne web site: http://www.compatibleone.org

[3] OCCI web site: http://occi-wg.org/

[4] Iain James Marshall – Cords Reference ManualV2.14, 2013

[5] MEDUSA WP1 partners – D111 deliverable: "Use Case Scenario and user requirements" – Nov. 2013.

[6] MEDUSA WP2 partners – D211/D221 deliverable: "Architecture and interface specifications and integration aspects" – Dec. 2013.

[7] MEDUSA WP2 partners – D213 deliverable: "Design for resource usage" – 2014.