OPAALS PROJECT

Contract n° FP6-034824

WP5: Integration with the Digital Ecosystem Platform

D5.3 - Re-engineered Servent

Project funded by the European Community under the "Information Society Technology" Programme

Contract Number : FP6-034824

Project Acronym : OPAALS

Deliverable N°: D5.3 Re-engineered ServENT

Due date : M29

Delivery Date : M33

Short Description : Description of the successive upgrades of the ServENT.

Author : Juanjo Aparicio, Jordi Martin, Ruben Gonzalez

Partners contributed : Techideas, ITA

Made available to: OPAALS Consortium and European Community

Versioning		
Version	**Date**	**Name, organization**
0.1	September	Juanjo Aparicio, Miguel Vidal (Techideas)
0.2	February	Juanjo Aparicio, Jordi Martin, Ruben Gonzalez (Techideas)
0.3	March	Juanjo Aparicio, Miguel Vidal (Techideas), Javier Val (ITA)

Quality check

Internal Reviewers :

Internal Reviewer: TV Prabhakar, IITK

Internal Reviewer: Ossi Nykanen, TUT

Internal Reviewer:

Dependences:

Achievements*	Technology upgrade
Work Packages	Depends on WP3
Partners	TechIdeas, ITA
Domains	Computer science
Targets	SMEs, Computer Science and Social Science domains
Publications*	
PhD Students*	
Outstanding features*	Several companies outside the OPAALS Consortium are using this technology and they requested an upgrade. This document describes such an upgrade, as well as the present state-of-the-art.
Disciplinary domains of authors*	All authors are under the Computer Science domain

The information marked with an asterisk (*) is provided in order to address Recommendation n. 4 from the Year 2 review report

Table of Contents

1 Introduction.... 6
1.1 Shifting from Java API to XMPP protocol.... 6
1.2 Ability to cross intranet boundaries.... 6
1.3 Fixing the NAT problem.... 7
1.4 Easy boot-strapping (borrowing infrastructures).... 7
2 Requirements (for a B2B integration infrastructure).... 8
2.1 Introduction.... 8
2.1.1 Background and definitions.... 8
2.1.2 Objectives.... 8
2.1.3 Structure.... 8
2.2 Methodology.... 9
2.3 Requirements list.... 9
2.4 Concluding remarks.... 15
2.5 Survey.... 16
2.5.1 SMEs involved in the survey.... 16
2.5.2 Survey done by SMEs involved in DBE test case.... 17
3 Network layer.... 28
3.1 Network layer behaviour.... 28
3.1.1 API versus protocol.... 28
3.1.2 The Protocol.... 29
3.1.2.1 JXTA.... 29
3.1.2.2 XMPP.... 31
3.1.2.3 XMPP message types.... 32
3.1.2.4 XMPP message examples:.... 33
3.1.2.5 Goals achieved using XMPP.... 34
3.1.3 Boot Strapping.... 35
3.1.4 Network Address Translation.... 35
3.1.5 Hole Punching.... 36
3.2 Socket abstraction.... 39
3.2.1 Client Socket.... 39
3.2.2 Server Socket.... 39
3.3 REST interaction.... 40
4 Services Layer.... 41
4.1 Servlet abstraction.... 41
4.2 Service example.... 42
5 Application layer (Sironta).... 45
5.1 Introduction.... 45
5.1.1 XMPP mappings.... 45
5.1.2 Registration & Login.... 45
5.2 New user interface.... 46
5.3 Improved boot strapping.... 47
5.4 Future work.... 48
5.4.1 Enhancement of the User Interface.... 48
5.4.2 Improvements in document management.... 48
5.4.3 New value added services.... 49
6 References.... 50

1 Introduction

This report describes the integration procedure followed in WP5 *("Integration with the Digital Ecosystem Platform")* during the last part of Phase II of the OPAALS Project. WP5 receives inputs from WP2, WP3 and requirements from WP10, and should support the user community of Digital Ecosystems existing outside, and beyond, the project itself.

The present deliverable has been categorized as software in the DoW. Therefore, this report essentially describes the improvement process accomplished in the Servent and complements the more than 84.000 lines of code that are the actual deliverable. It is also necessary to emphasize that the text of the present deliverable is not aimed to fully and comprehensively describe the interaction with other WP's, indeed, these are described in the deliverables of the respective WP's.

1.1 Shifting from Java API to XMPP protocol

One of the main changes in the reengineered Servent has been to move away from tightly coupled APIs for the invocation of services towards message oriented protocols. The motivation for this shift has been twofold. First of all, to allow implementations of services in other languages, platforms to interoperate with the reference Java implementation. While the Java platform is fairly ubiquitous, it is not universal. There is a fair amount of services and service consumers that will not be written in the Java language and will not run in the Java virtual machine. By decoupling the Servent from the Java platform with respect to service invocation the Servent becomes more useful. On the other hand, strict remote procedure calls (RPC) are the subject of very specific error conditions, that stem from their very nature. Loosely coupled systems seldom rely on RPC for their communications, and when they do, they present up front the tools to deal with networking errors that are hidden in a pure RPC interaction model. This change supposed the complete reimplementation of the communications stack.

1.2 Ability to cross intranet boundaries

In order to maximize the value of the Servent it is necessary to identify and attack the problems that became evident with the previous version of the Servent. The main problem is the ability for a Servent to provide services to peers outside the corporate network, when it exists. Firewalls are set up by networking administrators, and it is OK that a Servent can not be used because of the firewall policies: the were set by the networking administrator for good reason.

However, there is a different problem to deal with, and that is the Network Address Translation (NAT). To deal with the scarcity of IPv4 addresses, engineers came up with a solution to allow clients to share a single IP address. When such clients make a TCP connection to the Internet, their IP address is the same for all connections. Only the port number differs. The NAT device keeps track of what port number belongs to what internal IP address and effectively multiplexes the single shared IP address.

However, this mechanism disables the ability of the clients to receive direct connections from remote places in the network. The clients lack a globally reachable IP address. A remote peer may attempt to connect to the IP address of the NAT device, but a connection to the peers on the inside of the NAT is not generally possible.

1.3 Fixing the NAT problem

To deal with this situation hole punching techniques have been developed. These techniques involve the identification of the NAT device and its capabilities. Depending on the latter, one of various techniques will allow a remote peer to reach an peer behind the NAT device. It is estimated that these techniques work for an 87% of the commercially available networking devices which feature NAT.

1.4 Easy boot-strapping (borrowing infrastructures)

(or using already existing overlay networks: Google-Talk and Jabber)

In order to allow a peer to know about other peers some common infrastructure has to be used. It is possible to rely on the Domain Name Service, on UDDI registries, etc. In this re-engineered Servent we have opted for the usage of a widely deployed message-oriented infrastructure: XMPP servers. XMPP is an XML-based protocol that was designed for the interchange of instant messages between peers. It follows a client-multiple server model, in which peers connect to an XMPP server, and XMPP servers work in a peer-to-peer fashion in order to deliver messages, even in a cross-domain situation. Google-Talk uses the XMPP protocol and provides its customers with XMPP servers.

The objective of the following chapters of this deliverable is to provide a description of the software evolution followed during the last part of Phase II of the OPAALS Project and the successive releases of Sironta made during this reporting period.

2 Requirements (for a B2B integration infrastructure)

2.1 Introduction

In this chapter we will present a list of requirements, from the Small and Medium Enterprises point of view, for a Business to Business Integration Infrastructure.

2.1.1 Background and definitions

To achieve business objectives enterprises need to collaborate with other enterprises, so that the background is related to the following topics: B2B Integration (Interoperability) Infrastructure. We will define this terms in the context of our project:

Business-to-business (B2B) is a term commonly used to describe electronic commerce transactions between businesses, as opposed to those between businesses and other groups, such as business and individual consumers (B2C) or business and government (B2G).

Integration is the bringing together of the component subsystems into one system and ensuring that the subsystems function together as a system.

Interoperability is a property referring to the ability of diverse systems and organizations to work together (inter-operate). IEEE defines it as the ability of two or more systems or components to exchange information and to use the information that has been exchanged.

Software infrastructure in this context is a middleware that makes easier the task of connecting software modules or applications.

2.1.2 Objectives

The main objective is to retrieve requirements for a B2B integration infrastructure, based on the point of view of SMEs. We have considered the SMEs' opinion very important because the experience of Digital Business Ecosystem project showed us that the only way to guarantee infrastructure sustainability is if real business transactions could be performed with the platform.

For this purpose, we will be focused on execution infrastructure (the ExEcution environment in the DE world) instead of the Integrated Development Environments (the Studio in the DE).

These requirements will be the starting point for a re-engineered Servent to be developed into the OPAALS project and tested by SMEs in order to improve the DE community.

2.1.3 Structure

This chapter starts with this introduction, where background, definition and chapter objectives are presented. Next section is related to the methodology followed in order to retrieve requirements, which are detailed in the next section. The last section is where conclusions and references are shown. Finally the survey done by regional SMEs is shown in 2.5.

2.2 Methodology

This section shows the methodology followed in order to get infrastructure requirements. For this purpose, we have analyse three information sources:

- The Small and Medium Enterprises point of view
- Other research & development projects related to interoperability and integration of systems
- Analysis of the state of the art in technology for B2B integration

As our main purpose was the specification for a B2B integration infrastructure, from the point of view of Small and Medium, we have been in contact to all SMEs which were involved in the deployment of the Digital Business Ecosystem project in the Aragon region.

In order to know what functionalities were successfully implemented in the DBE ServENT, and what else have to be improved, we created a survey for the SMEs. This is the main information source to retrieve requirements for the infrastructure, because it represents the voice of the end users and developers. You can find the survey attached as annex to this document.

Before creating this survey, we analyse other technologies, standards and tendencies in Business to Business integration, in order to ask some questions regarding to the mature of them in their adoption by SMEs. In particular, we were focused on:

- Web Services standards and protocols
- BPEL based composition of services
- Open source Enterprise Service Bus
- Other technologies like OSGi and JBI

Finally, we have analysed other research projects and publications related to systems of systems interoperability, like:

- Service Oriented Computing Research Challenges
- Athena Integrated Project Publications
- Interop Network of Excellence Publications

2.3 Requirements list

This section shows the initial requirements lists for a Business To Business Integration Infrastructure, specially oriented for Small and Medium Enterprises. We have organized them into three categories:

- The first one is about mandatory requirements, that is, functionalities and other needs an infrastructure has to offer.
- The second one is related to suggested requirements, as a list of needs an infrastructure should offer.
- The last one is about other requirements that can't be implemented into any infrastructure but that are needed for SMEs to achieve B2B integration successfully.

Id[1]	Requirement: the infrastructure must...

R1.1 Be service oriented

Where *service* is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description (*OASIS definition*).

R1.2 Manage service deployment

It must manage the service cycle of live, that is, starting and stopping.

R1.3 Manage service searching

It must manage the searching for services deployed into it.

At the beginning, a service identifier based search can be enough, although it can be extended by a keywords based search.

The survey has shown that semantics are not mature for SMEs, so an ontology based search is not mandatory but suggested.

R1.4 Manage service consuming

It must manage the remote execution of services deployed into it, given a known interface.

Regarding to this requirement, it has to manage the transport layer in order to send and receive messages from “client” and “server”.

R1.5 Allow service composition

It must allow the composition of services deployed into it.

At the beginning, a static BPEL based service composition can be enough, although it can be extended by a keywords dynamic based service composition.

The survey has shown that dynamic service composition can be applied in restricted areas, because business transactions can't be dynamic without a legal regulatory framework.

R1.6 Manage security in the transport layer

It must use cypher algorithms to guarantee security in the message interchange.

R1.7 Manage identity

It must give an identity to any user of the infrastructure, so that in the client-server transactions, all “peers” are identified.

At the beginning, a central identity manager can be enough, although it can be

1*We will give an identifier to every requirement so that in the next chapter we can refer to each easier.*

extended to a distributed one.

The survey has shown that security is more important than wider markets, so that they prefer to reduce the number of peers in order to prevent from attacks.

R1.8 Make bootstrapping connection easier

In case the infrastructure uses peer-to-peer network technologies (suggested by SMEs), it must manage the connection to other peers without manual administration.

At the beginning, a stable community of peers can be all time running so that they guarantee the first connection for bootstrapping. It should be extended with discovering algorithms in future.

R1.9 Manage transactions in service composition

In the composition of services a transaction manager is needed, so that rollback mechanisms can be executed in case of failure.

At the beginning, a central transaction manager can be enough, although it can be extended to a distributed one.

The survey has shown that, from the SMEs' point of view, B2B integration is more related to instant transactions than time-consuming ones.

R1.10 Give support for scalability and performance

Although this is an ambiguous requirement, because there is no agreement about figures to define performance or scalability, it is under common sense that infrastructure has to be *scalable* and it has to offer g*ood performance*.

After some interviews with SMEs:

- performance is represented by response time when searching and executing a service. All interviewers agreed that ten seconds is a top value.
- scalability is represented by the challenge of adding new users or services without a high performance reduction (at least based on lineal or logarithmic reduction) and, more important, the absence of manual reconfiguration.

Of course this requirement needs to be interpreted in the context infrastructure (because it depends on the network architecture, protocols used and so on).

Id	Requirement: the infrastructure should...

R2.1 Be based on web services standards

A *web service* is defined by the World Wide Web Consortium as a software system designed to support interoperable Machine to Machine interaction over a network.

At the beginning, the use of Web Service Description Language (WSDL) for service description and Simple Object Access Protocol (SOAP) for service invocation can be enough.

R2.2 Give support for semantic management

Although semantics seems to be not very mature for SMEs, specially regarding to ontology modelling, including semantic management may be an added value for service composition.

It includes the semantic management when modelling service interfaces, when deploying service, when searching for services and specially when executing service composition.

The use of current standards (in example OWL) has to be considered in order to make easier integration to other existing infrastructures.

R2.3 Run on different software platforms

It may be run on different software platforms (Windows and Linux).

R2.4 Run on different hardware platforms

A more difficult requirement but with a higher added value would be if the infrastructure can run on different hardware platforms, specially on mobile devices.

R2.5 Be integrated to other infrastructures

If the infrastructure is based on standards like web services, integration to other infrastructures like enterprise service bus will be an added value.

The survey has shown that common open source ESBs used by SMEs are: Apache ServiceMix, open ESB and JBoss ESB.

R2.6 Be based on JBI and OSGi standards

Java Business Integration (JBI) specification (JSR 208) defines the core of a service oriented integration bus and component architecture for SOA. It standardizes the common message routing architecture, plug-in interfaces for service engines and bindings, and a mechanism to combine multiple services into a single executable and auditable unit of work.

The OSGi Alliance is a worldwide consortium of technology innovators that advances a proven and mature process to assure interoperability of applications and services based on its component integration platform.

OSGi technology is the dynamic module system for Java. The OSGi Service Platform provides functionality to Java that makes Java the premier environment for software integration and thus for development. The OSGi Service Platform provides the functions to change the composition dynamically on the device of a variety of networks, without requiring restarts. To minimize the coupling, as well as make these couplings managed, the OSGi technology provides a service-oriented architecture that enables these components to dynamically discover each other for collaboration.

R2.7 Support for asynchronous communication

Most current Web services applications are used in a connection-oriented manner so that a synchronous connection is required between the service requester and service provider. But for the mobile traveller, and for the emerging world of wireless networks, always being connected may not be possible or practical. For this audience the support for asynchronous transactions may be an added value.

R2.8 Support for extended web-services protocols

Although many of them are not mature for SMEs or event are not recognised as standards, there are some useful protocols in order to achieve more security, transactionability, coordination, reliable messaging, policy and context (WS-* protocols, like, WS-Security,
WS-Reliability, WS-Addressing ...)

R2.9 Manage distributed identity

Although at the beginning a central identity manager can be enough, a distributed one will give an added value to the platform. A reputation based mechanism is suggested to implement this feature.

R2.10 Implement some self-* properties

Behind the general idea of autonomic computing whose ultimate aim is to develop computer systems capable of self-management, any self-property will give an added value to the infrastructure in order to be used by more people. Some common properties are:

- Self-Configuration: Automatic configuration of components
- Self-Healing: Automatic discovery, and correction of faults
- Self-Optimization: Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements
- Self-Protection: Proactive identification and protection from arbitrary attacks

R2.11 Use P2P network architecture

Peer-To-Peer is suggested as network architecture so that every peer can fight in same conditions. Although, security mechanisms has to be implemented in order to prevent from attacks.

Id	Requirements that can't be implemented into the infrastructure

R3.1 Legal Regulatory Framework

If SMEs use an infrastructure to make business transactions, a legal regulatory framework is needed in order to resolve business conflicts.

Without it, an external pre-agreement based on signed contracts will be use by SMEs, adding formal legal conditions but decreasing the dynamics of the B2B integration.

R3.2 Quick Bootstrapping

For SMEs is very important to have a five-minutes step by step how-to guide in order to install, develop, deploy, search and consume a service.

The survey has shown this requirement as a key aspect in order to make easier the bootstrapping using the platform and become a critical mass.

R3.3 Have a support community

A support community is needed not only to offer help in case of problems but also to guarantee the sustainability of the infrastructure and to maintain samples and tutorials to achieve quick bootstrapping.

R3.4 Have an Integrated Development Environment

Although all requirements are related to a runtime infrastructure, to offer a design time IDE gives an added value to it.

A set of eclipse plug-ins (specially existing ones) are suggested by SMEs (in order to reduce their learning curve).

2.4 Concluding remarks

In this chapter we have presented a list of requirements, identified by SMEs, for a B2B integration infrastructure, divided into three categories: mandatory, suggested and other requirements needed to achieve B2B integration but not directly related to infrastructure.

A brief summary of them can be found as follows:

- Mandatory
  - Manage service cycle of live: deployment, searching, consuming...
  - Allow service composition
  - Manage identity
  - Make bootstrapping connection easier
- Suggested
  - Put more emphasis on standards, specially Web-Services for service description (WSDL) and invocation (SOAP)
  - Run on different software and hardware platforms (mobile devices)
  - Be integrated to other infrastructures, following JBI and OSGi standards
  - Implement some self-* properties
- Environment Requirements
  - A legal regulatory framework is needed to resolve business conflicts
  - A community is needed to give support and guarantee sustainability

2.5 Survey

2.5.1 SMEs involved in the survey

- Apser http://www.apser.es
- Assertum http://www.assertum.es
- Azierta Sistemas http://www.azierta-e.com
- Cierzo Development http://www.cierzo-development.com
- Damos Soluciones Informaticas http://www.dsi-e.com
- Dialcom http://www.dialcom.com
- Embou http://www.embou.com
- Europa Active Club http://www.europaactiveclub.eu
- Gabilos Software http://www.gabilos.com
- Iritec http://www.iritec.es
- Izanet Global Services http://www.izanet.com
- Minalink http://www.minalink.com/
- Neodoo http://www.neodoo.es
- Netfilia http://www.netfilia.com
- Net2U http://www.net2u.es
- Nitax http://www.nitax.net
- Seinteco http://www.seinteco.com
- TrackGlobe http://www.trackglobe.com
- TB-Solutions http://www.tb-solutions.com

2.5.2 Survey done by SMEs involved in DBE test case

This section shows the survey done by SMEs who where involved in the DBE test case in order to know their feedback about DBE ServENT as Business to Business Integration infrastructure.

Note: As all SMEs came from Spain, the survey was done in Spanish.

DBE

Encuesta sobre el proyecto DBE

0.- Bienvenida

0001: ¿Cuántos trabajadores tiene su empresa?

Por favor, escriba su respuesta aquí:

0002: ¿Cuántas personas de su empresa han participado en el proyecto DBE?

Por favor, escriba su respuesta aquí:

* **0003: Seleccione su rol dentro del proyecto**

Por favor, marque **todas** las que correspondan:

- ☐ Director / Gerente
- ☐ Técnico

1.- EXE

El Execution Environment es contenedor de aplicaciones o servicios del proyecto DBE

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

* **0001: Califique de 1 a 10 (siendo 1 la menor puntuación y 10 la mayor) las siguientes propiedades del Execution Environment**

Por favor, elija la respuesta apropiada para cada entrada:

	1	2	3	4	5	6	7	8	9	10
Instalación sencilla	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Actualización sencilla	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Documentación abundante	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Documentación útil	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Buen manual de usuario	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Seguridad	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Fiabilidad	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Facilidad para instalar servicios	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
Facilidad para buscar servicios	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

* **0002: ¿Conoce otros contenedores de aplicaciones?**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Sí
- ☐ No

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0002 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']

* **0003: De esta lista marque los servidores/contenedores de aplicaciones que conozca**

Por favor, marque **todas** las que correspondan:

- ☐ EAServer
- ☐ Internet Information Server
- ☐ J2EE
- ☐ JBoss
- ☐ JOnAS
- ☐ Tomcat
- ☐ WebLogic
- ☐ WebSphere
- ☐ Zope
- Otro: ☐

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0002 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']

* **0004: De esta lista marque los servidores/contenedores de aplicaciones con los que haya tabajado**

Por favor, marque **todas** las que correspondan:

- ☐ EAServer
- ☐ Internet Information Server
- ☐ J2EE
- ☐ JBoss
- ☐ JOnAS
- ☐ Tomcat
- ☐ WebLogic
- ☐ WebSphere
- ☐ Zope
- Otro: ☐

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0002 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']

* **0005: De esta lista elija el servidor/contenedor de aplicaciones que prefiera**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ EAServer
- ☐ Internet Information Server
- ☐ J2EE
- ☐ JBoss
- ☐ JOnAS
- ☐ Tomcat
- ☐ WebLogic
- ☐ WebSphere
- ☐ Zope
- ☐ Otro ☐

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0002 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']

* **0006: En relación con el servidor elegido en la pregunta anterior, DBE EXE es:**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Muchisimo peor
- ☐ Peor
- ☐ Similar
- ☐ Mejor
- ☐ Muchisimo mejor

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

*** 0007: ¿Cuál es la principal ventaja del servidor de aplicaciones DBE ExE?**

Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

*** 0008: ¿Cuál es el principal inconveniente del DBE ExE?**

Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

*** 0009: ¿Qué cree que se debe mejorar en el DBE ExE?**

Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

*** 0010: Valoración general de DBE Exe(entre 0 y 9)**

Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

0011: Escriba cualquier cosa que desee añadir sobre Dbe EXE.

Por favor, escriba su respuesta aquí:

2.- DBEstudio

DBE Studio es el entorno de desarrollo integrado para DBE

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

*** 0001: Califique de 1 a 10 (siendo 1 la menor puntuacion y 10 la mayor) las siguientes propiedades del DBE Studio**

Por favor, elija la respuesta apropiada para cada entrada:

Instalación sencilla	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6	☐ 7	☐ 8	☐ 9	☐ 10
Actualización sencilla	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6	☐ 7	☐ 8	☐ 9	☐ 10
Documentacion abundante	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6	☐ 7	☐ 8	☐ 9	☐ 10
Documentacion útil	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6	☐ 7	☐ 8	☐ 9	☐ 10
Buen manual de usuario	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6	☐ 7	☐ 8	☐ 9	☐ 10

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

*** 0002: ¿La herramienta "DBE Business Analysis" le ha resultado util para modelar los servicios?**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Sí
- ☐ No

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

*** 0003: Considera que el uso de la herramienta "DBE Business Analysis" de modelado de servicios es:**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Muy complicado
- ☐ Complicado
- ☐ Ni complicado ni sencillo
- ☐ Sencillo
- ☐ Muy sencillo

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

*** 0004: El número de ejemplos sobre modelado de servicios incluidos en DBE Studio opina que son:**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Pocos
- ☐ Adecuados
- ☐ Muchos

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

0005: ¿Como mejoraría usted la herramienta "DBE Business Analysis"?

Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

0006: ¿La herramienta "DBE Ontology Analysis" le ha resultado útil para modelar las ontologías que doten de contenido semántico a sus servicios?

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Sí
- ☐ No

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

*** 0007: Considera que el uso de la herramienta "DBE Ontology Analysis" de modelado de servicios es:**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Muy complicado
- ☐ Complicado
- ☐ Ni complicado ni sencillo
- ☐ Sencillo
- ☐ Muy sencillo

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']
* **0008: ¿Como mejoraría usted la herramienta "DBE Ontology Analysis"?**
Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']
* **0009: ¿Qué le parece la forma de desplegar los servicios?**
Por favor, elija **sólo una** de las siguientes entradas:
- ☐ Muy complicada
- ☐ Complicada
- ☐ Ni complicada ni sencilla
- ☐ Sencilla
- ☐ Muy sencilla

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']
0010: ¿Qué es lo que le ha resultado mas complicado a la hora de crear los servicios?
Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']
0011: ¿Qué es lo que le ha resultado mas complicado a la hora de desplegar los servicios?
Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']
0012: ¿Conoce la herramienta para generar la interfaz de usuario del servicio?
Por favor, elija **sólo una** de las siguientes entradas:
- ☐ Sí
- ☐ No

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0012 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']
0013: ¿Que mejoraría de la herramienta de generación de interfaces de usuario?
Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']
* **0014: Valoración general de DBE Studio (entre 0 y 9)**
Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Técnico' a la pregunta '0003 ']

0015: Escriba cualquier cosa que desee añadir sobre DBE Studio.

Por favor, escriba su respuesta aquí:

3.- ITA

Trabajo desarrollado por el Instituto Tecnológico de Aragón en el proyecto DBE

* **0001: ¿Ha acudido a agun evento (workshop, codecamp) sobre DBE organizado por ITA?**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Sí
- ☐ No

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0001 ']

* **0002: Valore de 1 a 10 las siguientes caracteristicas de los eventos DBE organizados en ITA**

Por favor, elija la respuesta apropiada para cada entrada:

Organización del evento	☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10
Condiciones Ambientales (aulas, laboratorios,...)	☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10
Nivel de los contenidos	☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10
Utilidad de los contenidos	☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10
Duración de los eventos	☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

* **0003: ¿Conoce la web de ITA sobre DBE (dbe.ita.es)?**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Sí
- ☐ No

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0003 ']

* **0004: El contenido de la web dbe.ita.es le parece**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Insuficiente
- ☐ Suficiente
- ☐ Plenamente suficiente

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0003 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']

* **0005: ¿Conoce los tutoriales disponibles en dbe.ita.es desarrollados por ITA?**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Sí
- ☐ No

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0003 ' *y* si usted respondió 'Sí' a la pregunta '0005 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']

* **0006: ¿Ha seguido algún tutorial de la web dbe.ita.es?**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Sí
- ☐ No

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0003 ' *y* si usted respondió 'Sí' a la pregunta '0005 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']

0007: ¿Le han resultado utiles estos tutoriales?

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Sí
- ☐ No

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0003 ' *y* si usted respondió 'Sí' a la pregunta '0005 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']

* **0008: ¿Que tutoriales ha utilizado?**

Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0003 ' *y* si usted respondió 'Sí' a la pregunta '0005 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']

0009: ¿Cómo mejoraría los tutoriales?

Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Sí' a la pregunta '0003 ' *y* si usted respondió 'Técnico' a la pregunta '0003 ']

0010: Califique los ejemplos disponibles en dbe.ita.es

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Variados
- ☐ Suficientes
- ☐ Complicados
- ☐ Utiles

* **0011: En general el trabajo realizado por ITA en este proyecto segun usted ha sido:**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Pesimo
- ☐ Malo
- ☐ Regular
- ☐ Bueno
- ☐ Excelente

0012: Escriba cualquier cosa que desee añadir sobre el trabajo realizado por ITA.

Por favor, escriba su respuesta aquí:

4.- Empresa

Trabajo realizado por la empresa en el proyecto DBE

[Sólo responda esta pregunta si usted respondió 'Director / Gerente ' a la pregunta '0003 ']

* **0001: ¿Cómo conoció el proyecto?**

Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Director / Gerente ' a la pregunta '0003 ']

0002: ¿Qué esperaba conseguir en este proyecto?

Por favor, escriba su respuesta aquí:

[Sólo responda esta pregunta si usted respondió 'Director / Gerente ' a la pregunta '0003 ']

* **0003: Indique el grado en el que se han cumplido sus expectativas**

Por favor, elija **sólo una** de las siguientes entradas:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

[Sólo responda esta pregunta si usted respondió 'Director / Gerente ' a la pregunta '0003 ']

* **0004: La participación en DBE para su empresa ha sido**

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Negativa
- ☐ Ni negativa ni positiva
- ☐ Positiva

[Sólo responda esta pregunta si usted respondió 'Director / Gerente ' a la pregunta '0003 ']

* **0005: ¿Qué ha aportado a su empresa este proyecto?**

Por favor, escriba su respuesta aquí:

0006: ¿Seguirá usando la infraestructura DBE tras la finalización del proyecto?

Por favor, elija **sólo una** de las siguientes entradas:

- ☐ Sí
- ☐ No

[Sólo responda esta pregunta si usted respondió 'No' a la pregunta '0006 ']

0007: ¿Por qué no seguirá usando DBE?

Por favor, marque **todas** las que correspondan:

- ☐ Tecnología poco madura
- ☐ No encuentro utilidad
- ☐ Hay cosas mejores
- ☐ No creo que se siga desarrollando

Otro:

[Sólo responda esta pregunta si usted respondió 'Director / Gerente ' a la pregunta '0003 ']

* **0008: Le gustaría participar en otros proyectos similares en el futuro**

Por favor, elija **sólo una** de las siguientes entradas:

☐ Sí

☐ No

0009: Valoración general de la participación de su empresa en el proyecto DBE (entre 0 y 9)

Por favor, escriba su respuesta aquí:

0010: Escriba cualquier cosa que desee añadir sobre el trabajo realizado por su empresa.

Por favor, escriba su respuesta aquí:

5.- Otros

0001: Escriba cualquier otro comentario que desee realizar y que no haya quedado registrado en las anteriores preguntas

Por favor, escriba su respuesta aquí:

3 Network layer

One of the main achievements has been shifting to a messaging protocol in order to open the systems to as many programming languages as possible. Other of the remarkable achievements has been facing the NAT problem. Network address translators (NATs) are something every software engineer has heard of, not to mention networking professionals. Fundamentally, a NAT device allows multiple machines to communicate with the Internet using a single globally unique IP address, effectively solving the scarce IPv4 address space problem.

Hole punching, when possible, has been the solution implemented to solve the NAT problem for P2P protocols, as discussed in the Introduction. Hole punching is a well known technique for establishing communications between two parties in separate organizations who are both behind restrictive firewalls. Using this strategy, both clients establish a connection with an unrestricted third-party server that uncovers external and internal address information for them. Since each client initiated the request to the server, the server knows their IP addresses and port numbers assigned for that session, which it shares one to the other.

3.1 Network layer behaviour

In order to sort out all the problems related with NAT a very flexible strategy has been adopted. In a first stage, a direct connection between two nodes always is tryed. In case of failure (highy probable) a work around is started, with the help of the XMPP server acting as mediator.

The concept of a rendezvous server, or mediator server, which listens on a globally routable IP address, has been introduced. Almost all peer-to-peer protocols have traditionally relied on certain supernodes. Some nodes always have acted as key players in any P2P protocol (BitTorrent tracker is an example). The rendezvous concept is nothing new in the P2P world, nor is the star model totally done away with in P2P. We used the existing XMPP infrastructure and protocols to act as rendezvous for the establishment of data streams between peers, and perforate holes in the NAT where possible, so we are able to route XMPP traffic directly between two peers. At the worst case, that is, if a hole can not be punched in the NAT, or the networking environment doesn't use NAT but stronger limitations due to network administrator policies, the existing XMPP infrastructure will allow, as long as XMPP traffic is allowed at all, any two peers to communicate by using the XMPP servers as relays.

3.1.1 API versus protocol

There are two common ways to engage a service client to a service provider:

- Using an Application Programming Interface
- Using a common protocol

An API provides a library that you must link (dynamically or statically) with to use the services. This tightly binds the client and server together. The API tends to invade all code layers and creates massive dependencies between layers. It also tends to be simple to use.

A protocol defines a standard request response layer and a common transport. Nothing other than the standard binds the client and server together. Protocols are more complex to use as they are less

direct and take a lot of serializing/deserializing/dispatching type logic. However, after the experience gathered with the previous Servent network layer implementation, we strongly argue for protocols because they provide the most flexibility. Program layers can evolve independently of each other. APIs, like CORBA, usually infect the top most layer down to the bottom most layer which create serious dependencies between layers. A change in one will require a change in the other layers making very difficult code maintenance and evolvability.

It is also possible to have *both* an API and a Protocol. Basically, any Protocol can be wrapped with proxies to form an API (for example, the Apache SOAP implementation). However, the Apache Java implementation allows an API to be layered on top of it. It doesn't matter that the server you have to contact is written in Visual Basic or Visual C++ - the Java API can still work as long as you can describe the interface. However, is something discarded due to the difficulties arisen during the design of an architecture back compatible with the previous Servent.

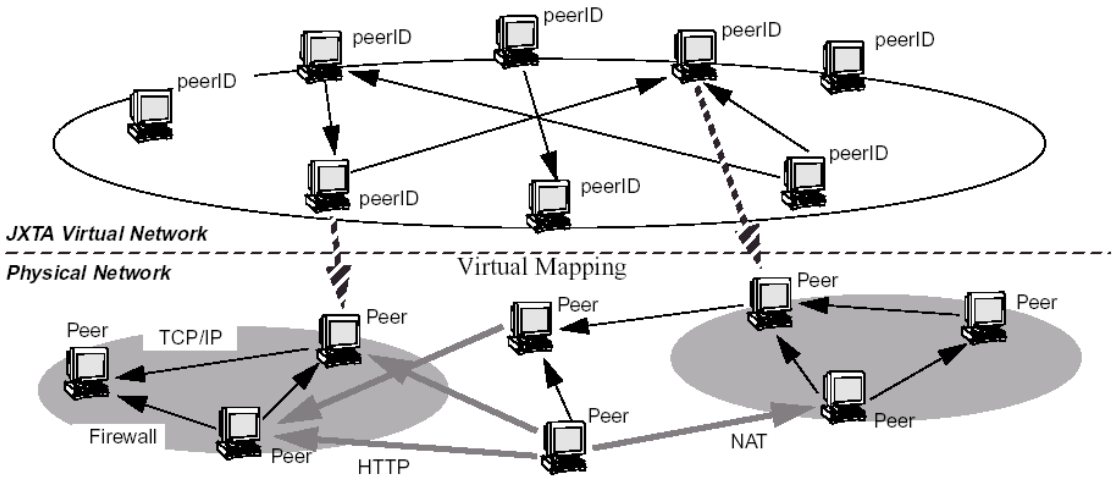
Summarizing, as an API is formally equivalent to a protocol and, a protocol multiple languages implementation and multiple devices, a standard protocol was chosen to perform Servent interactions.

3.1.2 The Protocol

One of the main tasks faced at the beginning of Phase II was to look for a standard protocol that properly satisfy all the requirements from the partners and the SME's. The next task was to look for an open standard protocol that grants decentralization, flexible enough to follow the constantly changing requirements of Digital Ecosystems and with good support of security. After a carefully analysis of the available open protocols, it was concluded that the only one protocol that satisfy all the requirements with a large number of existing implementations (Java, C++, C#, Phyton, Flex, JavaScript, etc...), wide adoption, and already deployed networks was XMPP.

3.1.2.1 JXTA

JXTA was one of the evaluated protocols. JXTA was developed by Sun Microsystems and launched at 2001 Java One, and it is a set of six protocols that enable any connected node on the network, to communicate and collaborate in a P2P manner. JXTA peers create a virtual network where any peer can interact with other peers and resources directly, even when some of the peers and resources are behind firewalls and network address translations (NAT's) or on different network transports.



Because it is based on protocols and not an API, JXTA works potentially with any language, and any network-capable device can be a JXTA peer. Because the underlying network does not have to be TCP/IP, JXTA applications can include Bluetooth-enabled mobile handsets as peers. JXTA is, in some extent, similar to Jini (reason why it was our first choice), but Jini is more about discovery and using services and require a Java Virtual Machine on every member device, due to its Java centric inception, whereas JXTA does not and is more about transport (working in a similar layer than IP).

JXTA provides the protocols for basic functions of peer-to-peer networking, such as creating, finding, joining, leaving and monitoring groups, talking to other groups and peers, and sharing content and services. The functions are performed by exchanging XML advertisements and messages between peers. The six basic protocols are:

Peer Discovery Protocol (PDP)

Peers use this protocol to discover all published JXTA resources. Since advertisements represent published resources, PDP essentially helps a peer discover an advertisement on other peers. As the lowest-level discovery protocol, PDP provides a basic mechanism for discovery. Applications might choose to use higher-level discovery mechanisms. PDP serves as a low-level protocol over which higher-level discovery mechanisms can be built.

Peer Resolver Protocol (PRP)

Often in the network, peers send queries to other peers to locate some service or content. The Peer Resolver Protocol intends to standardize these queries' formats. With this protocol, peers can send generic queries and receive responses.

Peer Information Protocol (PIP)

PIP can be used to "ping" a peer in the JXTA environment. A peer receiving a ping message has several options: It can give a simple acknowledgement, consisting only of its uptime. It can send a full response, which includes its advertisement. Or it can ignore the ping. Thus, there can be peers capable of receiving messages but not sending responses.

Peer Membership Protocol (PMP)

Peers use the Peer Membership Protocol for joining and leaving peer groups. This protocol recognizes four discrete steps used by peers and thus defines Jxta messages for each of these actions:

- *Apply:* A peer interested in entering a group can apply for a membership to the group membership authenticator. The authenticator responds by sending back an acknowledge message to the peer.
- *Join:* After an apply, the peer can choose to join the peer group.
- *Renew:* To update their membership information in the group, peers use the renew message.
- *Cancel:* Peers can choose to cancel their peer group memberships.

Pipe Binding Protocol (PBP)

In the JXTA environment, peers use pipes to access services. A peer can bind to a pipe's end at runtime and access services. The peer can create a new pipe, bind to an existing pipe, and unbind from a pipe. For those cases, the peer uses the Pipe Binding Protocol.

Endpoint Routing Protocol (ERP)

This protocol helps a peer route messages to a destination. The ERP helps peer routers query other peer routers about available routes for sending messages.

Because JXTA is based in the management of its own messages, it is needed a JXTA network infrastructure made of:

Rendezvous peers

A rendezvous peer is a special peer that stores information about other peers it knows about by caching these known peers' advertisements. Thus, a rendezvous peer can help peers discover other peers in the network. Rendezvous peers can also forward discovery requests to other rendezvous peers.

Endpoints

Endpoints are destinations on the network and can be represented by a network address. Peers don't generally use endpoints directly; they use them indirectly through pipes, which are built on top of endpoints.

Routers

Anything that moves packets around the JXTA network is called a JXTA router. Not all peers need to be routers. Peers that are not routers must find a router to route their messages.

capable of manage JXTA messages.

As a conclusion, we can say that JXTA is an excellent option for a P2P transport protocol, when we already have infrastructure (routers, relays,, etc...) and a developing language where JXTA has been ported (basically Java and C). This two factors have been a serious break to JXTA adoption, due to the difficulties to widely deploy JXTA infrastructure .

3.1.2.2 XMPP

In spite of the obvious advantages of JXTA, it was not specifically designed to give awareness about presence, events, etc... as its behaviour (the set of protocols) is designed to be a flexible transport mechanism, what means that it is a layer below what we were looking for (comparing against OSI model). Additionally we have to take into account that it has not been widely used, reason why there are no JXTA infrastructure deployed in relevant quantities (routers, rendez-vous servers, etc...)

Extensible Messaging and Presence Protocol (XMPP) is an open, XML-inspired protocol originally aimed at near-real-time, extensible instant messaging (IM) and presence information, but now expanded into the broader realm of message oriented middleware. XMPP technologies have been in use since 1999. Multiple implementations of the XMPP standards exist for clients, servers,

components, and code libraries, with the backing of large companies such as Sun Microsystems and Google.

The base specifications of the Extensible Messaging and Presence Protocol (XMPP) formalize the core protocols developed within the Jabber open-source community in 1999. They were produced by the IETF's XMPP Working Group and published as RFCs 3920 and 3921 in October 2004.

RFC	Short Name	Description
RFC3920	XMPP Core	XML streams, SASL, TLS, stringprep profiles, stanza semantics
RFC 3921	XMPP IM	XMPP extensions for basic instant messaging and presence
RFC 3922	XMPP CPIM	Mapping XMPP to the IETF's CPIM specifications
RFC 3923	XMPP E2E	End-to-end signing and object encryption for XMPP
RFC 4854	XMPP URN	A Uniform Resource Name (URN) tree for use in XMPP extensions
RFC 4979	XMPP ENUM	IANA registration of an Enumservice (see RFC 3761) for XMPP
RFC 5122	XMPP URI	A Uniform Resource Identifier (URI) scheme for XMPP (this specification corrects several errors in RFC4622)

In addition to the RFCs, there are many XMPP extensions defined in the XEP series produced by the XMPP Standards Foundation, and several XMPP-related Internet-Drafts are currently under consideration within the IETF.

3.1.2.3 XMPP message types

XMPP is based in three fundamental messages, from which any complex communication interaction can be built. This three basic building blocks are:

- Presence: Send presence data or manage presence subscriptions

```
<presence from='miguel.vidal@techideas.es'>
    <status>At Phone</status>
</presence>
```

- Message: Send data between users

```
<message to='jesus.gabaldon@techideas.es'
         from='miguel.vidal@techideas.es'>
    <body>What are you doing?</body>
</message>
```

- IQ: Exchange information and perform queries using a request / response protocol.

Request:

```
<iq to=‘date.sironta.com’ type=‘get’ id=‘2002’><query
xmlns=‘jabber:date’><timezone>GMT+1</timezone></query></iq>
```

Response:

```
<iq from=‘date.sironta.com’> type=‘result’ id=‘2002’><query
xmlns=‘jabber:date’><date>thu mar 19 21:04:48 CET 2009</date></query></iq>
```

3.1.2.4 XMPP message examples:

Example of XMPP connection message Exchange

Login process messages

Client configuration request message:

```
<?xml version='1.0'?>
<stream:stream xmlns:stream='http://etherx.jabber.org/streams'
xmlns='jabber:client' to='gmail.com' version='1.0'>
```

Server configuration response message:

```
<?xml version="1.0" encoding="UTF-8"?>
<stream:stream from="gmail.com" id="6A32046CA5D9B6BE" version="1.0"
xmlns:stream="http://etherx.jabber.org/streams" xmlns="jabber:client">
<stream:features>
<mechanisms xmlns="urn:ietf:params:xml:ns:xmpp-sasl">
<mechanism>PLAIN</mechanism>
<mechanism>X-GOOGLE-TOKEN</mechanism>
</mechanisms>
</stream:features>
```

Client authentication request message:

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl' mechanism='PLAIN'>
AG9uYXBlcGFzdGVyaXNrc2VydmVyAGlmVVBnbSoyAA==
</auth>
```

Server authentication response message:

```
<success xmlns="urn:ietf:params:xml:ns:xmpp-sasl"/>
Client connection request message:

<?xml version='1.0'?>
```

```
<stream:stream xmlns:stream='http://etherx.jabber.org/streams'
xmlns='jabber:client' to='gmail.com' version='1.0'>
```

Server connection response message (the latest login process step) :

```
<?xml version="1.0" encoding="UTF-8"?>
<stream:stream from="gmail.com" id="F6BAC003DCED3977" version="1.0"
xmlns:stream="http://etherx.jabber.org/streams"
xmlns="jabber:client"><stream:features>
<bind xmlns="urn:ietf:params:xml:ns:xmpp-bind"/>
<session xmlns="urn:ietf:params:xml:ns:xmpp-session"/>
</stream:features>
```

Session creation process messages

Binding to resource request message:

```
<iq type='set' id='aaaaa'><bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
<resource>Talk</resource></bind></iq>

<iq type='set' id='auth'><session xmlns='urn:ietf:params:xml:ns:xmpp-
session'/></iq>
```

Binding to resource response message (this request return the full qulified user id).

```
<iq id="aaaaa" type="result"><bind xmlns="urn:ietf:params:xml:ns:xmpp-bind">
<jid>techideas@gmail.com/Talk166C0B33</jid></bind></iq>
```

Presence notification message:

```
<presence type='subscribe' to='techideas@gmail.com'>
<status>Available </status>
</presence>
```

Simple chat message:

```
<message type='chat' to='techideas@gmail.com'
from='sironta@gmail.com/Talk166C0B33'>
<body>Hi Techideas</body></message>
```

Exchange information and perform queries (request / response protocol)
Request:

```
<iq to='weather.sironta.com' type='get' id='1234'><query xmlns='jabber:weather'/
><city>Barcelona</city></iq>
```

Response:

```
<iq from='weather.sironta.com' type='result' id='1234'><query
xmlns='jabber:weather'/><weather>The sun is shinning!!</weather></iq>
```

3.1.2.5 Goals achieved using XMPP

Decentralization: The architecture of the XMPP network is similar to e-mail and anyone can run its own XMPP server, reason why there is no central master server.

Open standards: The Internet Engineering Task Force has formalized XMPP as an approved instant messaging and presence technology under the name of XMPP, and the XMPP specifications have been published as RFC 3920 and RFC 3921. No royalties are required to implement support of these specifications and their development is not tied to a single vendor.

Security: XMPP servers may be isolated from the public Jabber network (e.g., on a company intranet), and robust security (via SASL and TLS) has been built into the core XMPP specifications.

Flexibility: Custom functionality can be built on top of XMPP; to maintain interoperability, common extensions are managed by the XMPP Software Foundation. XMPP applications beyond IM include network management, content syndication, collaboration tools, file sharing, gaming, and remote systems monitoring.

Ubiquity: Last but not least, the capability to interact from any standard XMPP client, or with any standard XMPP library from our own application.

3.1.3 Boot Strapping

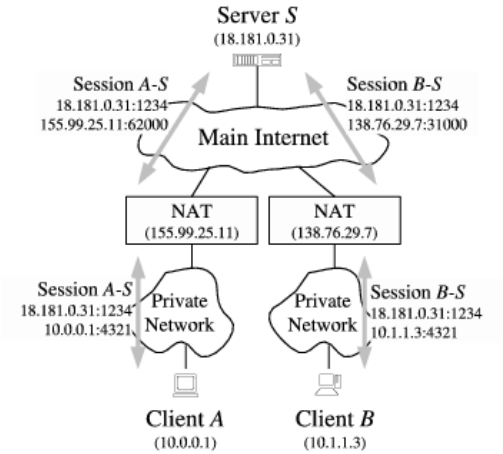
Following the XMPP standards makes the boot strapping mechanism easier, letting us to use the already existing XMPP networks, such as Jabber and Google-Talk. In this sense, there is no need of a truly bootstrapping mechanism, due to the fact that the P2P network used is already deployed by different large communities or companies.

This is not an obstacle to deploy, as a project, our own P2P XMPP overlay, but it is not incompatible with the fact that the interaction with other networks (even not supporting XMPP) is perfectly feasible. In fact, it is theoretically possible to interact with AIM, GaduGadu, ICQ, IRC, MSN, SIMPLE, Yahoo due to the fact that are available bridges from XMPP to such networks.

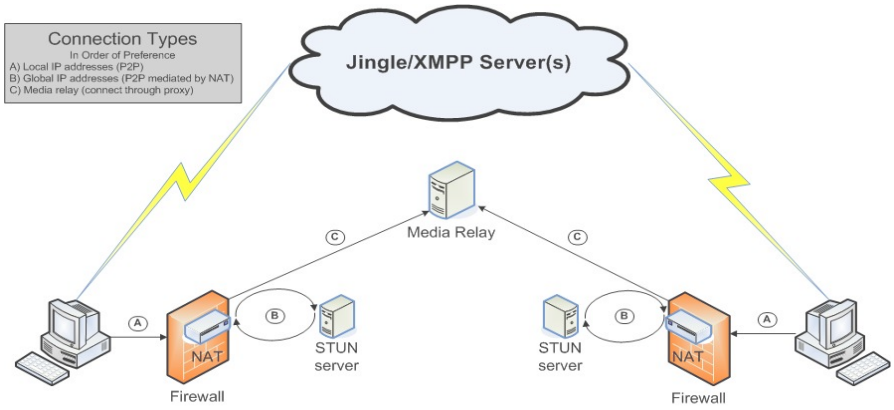
3.1.4 Network Address Translation

Network Address Translation (NAT) causes well-known difficulties for peer-to-peer (P2P) communication, since the peers involved may not be reachable at any globally valid IP address. One of the simplest but most robust and practical NAT traversal techniques is commonly known as ``hole punching.", and it has been the one chosen to be implemented in the current version of the Servent. It has been reported that about 82% of the commercial routers support hole punching for UDP, and about 64% support hole punching for TCP streams. As NAT vendors become increasingly conscious of the needs of important P2P applications such as Voice over IP and on-line gaming protocols, support for hole punching is likely to increase in the future.

Obviously, the most reliable (but least efficient) method of P2P communication across NAT is simply to make the communication look to the network like standard client/server communication, through relaying.



Relaying always works as long as both clients can connect to the server. For the current implementation, in band XMPP communications have been used using the XMPP servers as relays.



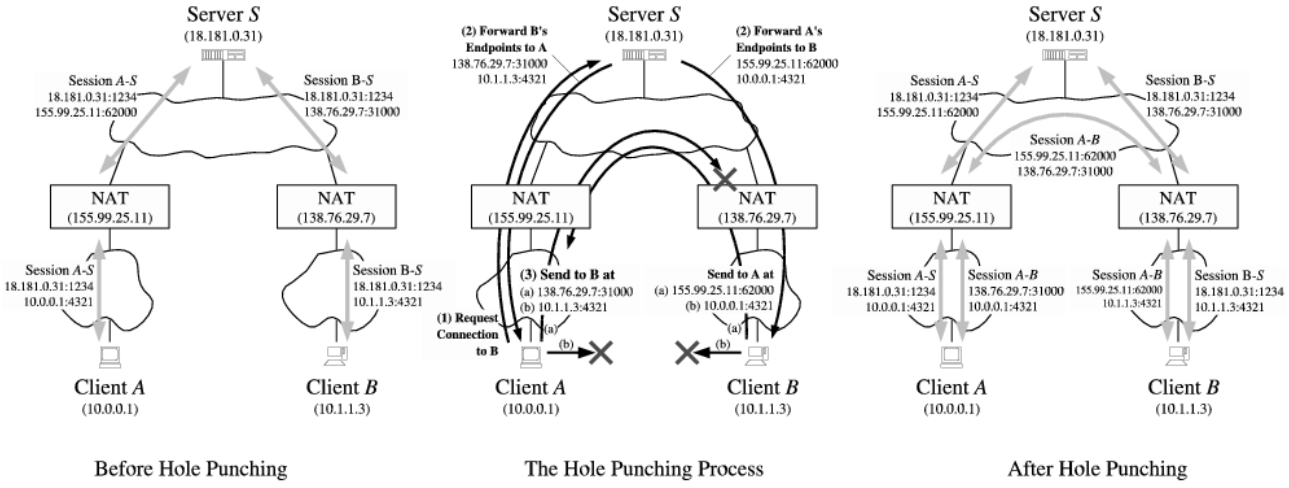
This approach is the one that grants 100% of availability in spite of a very low performance, and is the last option when the try to perform hole punching fails.

3.1.5 Hole Punching

UDP hole punching enables two clients to set up a direct peer-to-peer UDP session with the help of a well-known rendezvous server, even if the clients are both behind NATs.

Hole punching assumes that the two clients, A and B, already have active UDP sessions with a rendezvous server S (in our case any the Sironta XMPP Server). When a client registers with the XMPP server, the server records *two* endpoints for that client: the (IP address, UDP port) pair that the client *believes* itself to be using to talk with the XMPP server, and the (IP address, UDP port) pair that the server *observes* the client to be using to talk with it. We refer to the first pair as the client's *private* endpoint and the second as the client's *public* endpoint. The server might obtain the client's private endpoint from the client itself in a field in the body of the client's registration message, and obtain the client's public endpoint from the source IP address and source UDP port fields in the IP and UDP headers of that registration message. If the client is *not* behind a NAT, then its private and public endpoints should be identical.

Let's suppose that client A wants to establish a UDP session directly with client B.



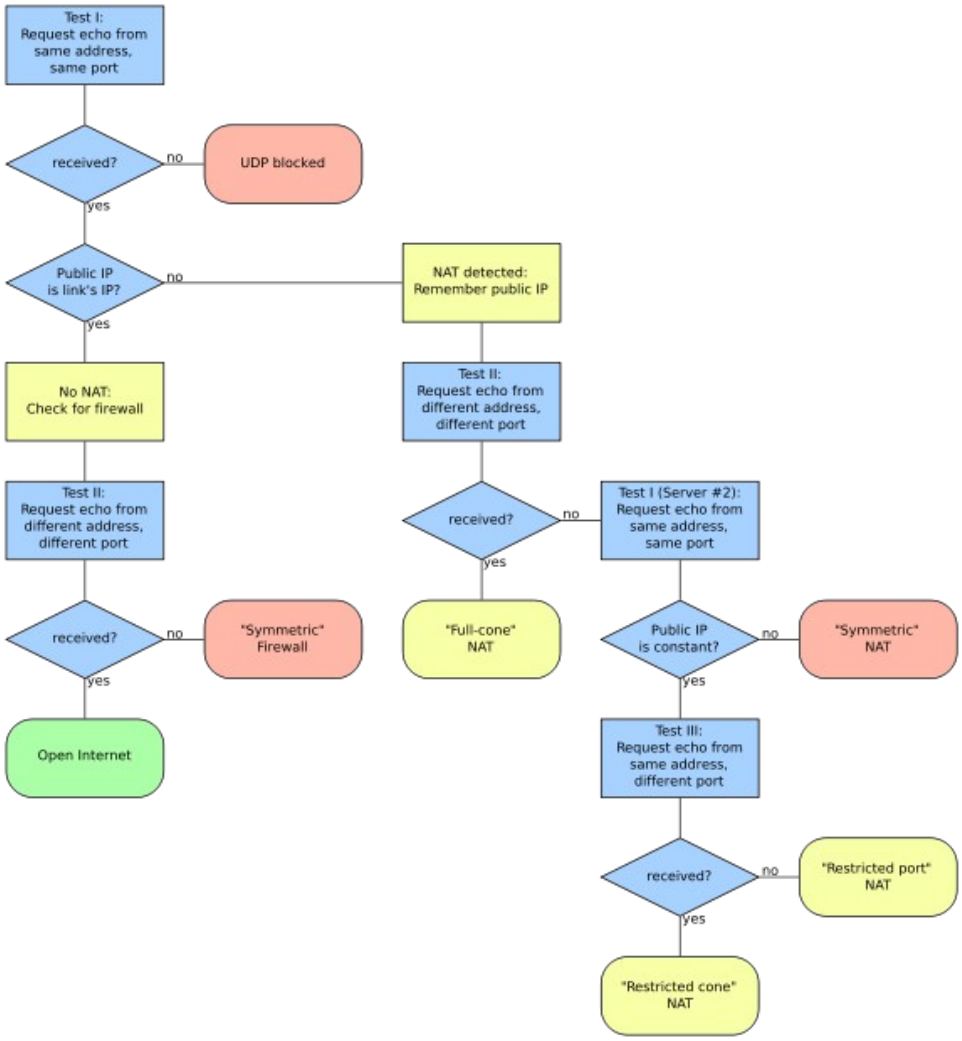
Hole punching proceeds as follows:

1. A initially does not know how to reach B, so A asks S for help establishing a UDP session with B.
2. S replies to A with a message containing B's public *and* private endpoints. At the same time, S uses its UDP session with B to send B a connection request message containing A's public and private endpoints. Once these messages are received, A and B know each other's public and private endpoints.
3. When A receives B's public and private endpoints from S, A starts sending UDP packets to *both* of these endpoints, and subsequently ``locks in" whichever endpoint first elicits a valid response from B. Similarly, when B receives A's public and private endpoints in the forwarded connection request, B starts sending UDP packets to A at each of A's known endpoints, locking in the first endpoint that works. The order and timing of these messages are not critical as long as they are asynchronous.

It is in general difficult or impossible for the application itself to determine the exact physical layout of the network, and thus to which extent hole-punching is a feasible option. Protocols such as STUN can provide some information about the NAT's present on a communication path, but this information may not always be complete or reliable, especially when multiple levels of NAT are involved.

STUN, Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) is a lightweight protocol that allows applications to discover the presence and types of NATs and firewalls between them and the public Internet. It also provides the ability for applications to determine the public Internet Protocol (IP) addresses allocated to them by the NAT.

STUN protocol is summarized in the following flux diagram:



STUN works with many existing NATs, and does not require any special behaviour from them. As a result, it allows a wide variety of applications to work through existing NAT infrastructure. Nevertheless, hole punching works automatically in all of these scenarios *without* the application having to know the specific network organization, as long as the NATs involved behave in a reasonable fashion. In our particular scenario, the STUN service provided by the XMPP server (Openfire) is the one used in order to proceed, giving encouraging results in the set of trials made up to now.

3.2 Socket abstraction

One of the main requirements of the Servent is to make easy to port already existing applications to it. The goal is to increase quickly the number of applications supported by the Servent. With this goal in mind, a Socket implementation on top of the transport layer has been implemented from scratch, allowing to plug different transport P2P protocols, with different behaviours, in a completely transparent way for the application.

3.2.1 Client Socket

```
public XMPPSocket(XMPPConnection con, String jid, String connectPort)
public void connect(XMPPConnection con, String jid, String connectPort)
public OutputStream getOutputStream()
public InputStream getInputStream()
```

3.2.2 Server Socket

```
public XMPPServerSocket(XMPPConnection con, String connectPort)
public XMPPSocket accept() throws XMPPException, InterruptedException
```

The above interface mimics the interfaces of the socket abstraction in the Java API, allowing programmers acquainted with the latter to work comfortably with the former.

3.3 REST interaction

In order to maximize the reliability of the reengineered Servent, the service layer on top of the aforementioned socket abstraction provides CRUD (Create, Retrieve, Update, Delete) interfaces for the implementation of services. CRUD interfaces are used by SQL engines and protocols like HTTP. This uniform interface is one of the traits of a REST architecture.

REST is an architectural style defined for the first time by Roy T. Fielding in his Ph.D. Thesis [1]. It is the architectural style of the Web, and one of the reasons of the Web success.

The REST acronym stands for REpresentation State Transfer. Basically, it means that interaction between parties is performed through the transfer of representations of the state of the involved entities. One corollary of the style is that no state is implicit. This allows for operations to be cancelled, paused and resumed, as long as the state of the operations can be represented, and the state representation can be transferred between the involved parties. This is very interesting when you are dealing with long-running operations which are therefore prone to failure due to networking errors.

Another trait of the REST architectural style, already mentioned, is that of the uniform interface. When the interaction between parties is performed through APIs, additional functionalities are implemented by extending the API. These extensions render older services obsolete, because they are not able to deal with the requests of newer clients. But it also makes it more difficult to deploy mediators/helpers for these requests, because the mediators/helpers have to be updated to be able to deal with the latest version of the API.

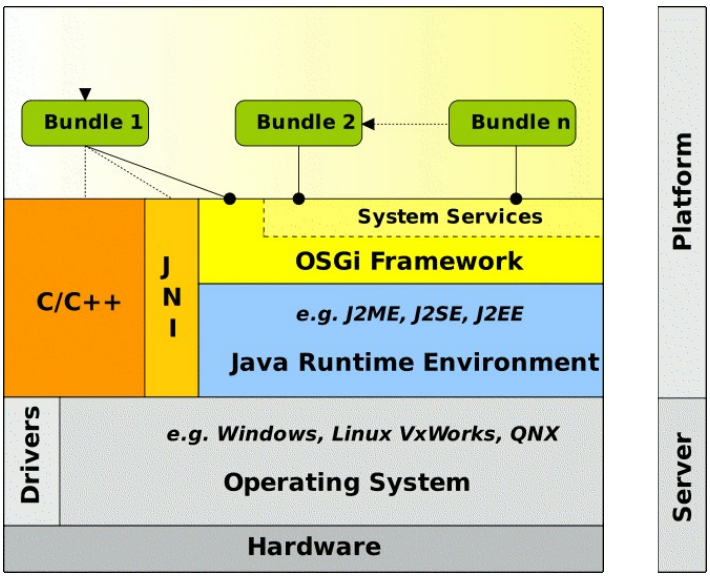
The uniform interface allows the presence of mediators/helpers like proxies and caches because they don't need to understand the effects of the operations they carry on behalf of third parties. They just need to understand (and respect) the semantics of the uniform interface. The key is to have the intelligence on the data and not on the actions.

One of the most widely deployed protocols that makes use of the CRUD interface is HTTP. J2EE programmers are used to the **doGet**, **doPost**, etc... methods on Servlets. The reengineered Servent uses a service implementor API that mimics that interface.

```
public HttpRequest(InputStream is) throws IOException
public HttpRequest(String method, String resource)
public HttpResponse(HttpRequest request, OutputStream os)
public void addHeader(String key, String value
public String getHeader(String key)
public void setBody(File temp_sigs) throws FileNotFoundException
public byte[] getBody() throws IOException
```

4 Services Layer

A tremendous effort has been made in order to provide a standard way of deployment within this version of the Servent. Now, services are bundled as OSGi services, allowing to use most of the IDE which supports this kind of bundling.



In particular, the choice for the current implementation deployment has been Eclipse. Eclipse RCP is a platform for building and deploying rich client applications. It includes *Equinox*, which is OSGi reference implementation. Beyond this, Eclipse is a component framework based on the OSGi standard, which bring us the ability to deploy native GUI applications to a variety of desktop operating systems, such as Windows, Linux and Mac OSX and an integrated update mechanism for deploying desktop applications from a central server. This features and the large active community around Eclipse made it the best choice for the project and its evolutionary capabilities.

4.1 Servlet abstraction

Following the same philosophy mentioned in the network layer, the goal in the service layer was exactly the same. Looking for a current common place for most of the service developers that were compatible with REST approach, the Servlet mimic was an easy solution.

Adopting the same semantics, allow to port to the Servent many of the applications deployed under Apache Tomcat (or any other container) without struggling the programmers.

```
public class FileProcessor extends Processor {
	@Override
	public void doGet(HttpRequest request, HttpResponse response)
	throws IOException {
		[...]
	}

	@Override
	public void doPost(HttpRequest request, HttpResponse response)
	throws IOException {
		[...]
	}
}
```

This approach should ease the migration and deployment of services already available as Servlets, and it is aligned with the programming patterns used by most of the potential users of the Servent.

4.2 Service example

As a reference, here is posted the source code of a service. This service is the one trough whis one user invite to another in order to be included in the roster.

```
package com.sironta.rooms.processors;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.URLEncoder;

import org.eclipse.core.resources.IFolder;
import org.jivesoftware.smack.XMPPConnection;
import org.jivesoftware.smack.XMPPException;

import com.sironta.rooms.model.Room;
import com.sironta.rooms.model.RoomImpl;
import com.sironta.rooms.model.RoomModel;
import com.sironta.users.model.LocalUser;
import com.sironta.users.model.LocalUserImpl;
import com.sironta.users.model.User;
import com.sironta.users.model.UserImpl;
import com.sironta.users.model.UserModel;
import com.sironta.xmpp.XMPPSocket.XMPPSocket;
import com.sironta.xmpp.http.HttpRequest;
import com.sironta.xmpp.http.HttpResponse;
import com.sironta.xmpp.http.Processor;

public class InviteUserProcessor extends Processor {
```

```
	private RoomModel roomModel;

	public InviteUserProcessor(RoomModel roomModel) {

		this.roomModel = roomModel;

	}

	@Override
	public void doPost(HttpRequest request, HttpResponse response)
			throws IOException {
		InputStream is = request.getInputStream();
		InputStreamReader isr = new InputStreamReader(is);
		BufferedReader br = new BufferedReader(isr);
		String line = br.readLine();
		LocalUser localUser = LocalUserImpl.getInstance();
		if (line.length() > 0) {
			String roomName = getRoomName(line);
			if (localUser != null
					&& !roomModel.getRooms().containsKey(roomName)) {
				RoomImpl room = getRoom(line, roomModel);
				roomModel.addRoom(room);
				room.getUserModel().addUser(localUser);

			}
		}
		response.setStatusCode(200);

	}

	private boolean sendXMPPAddUser(LocalUser localUser, Room room) {
		// send invitation to user

		StringBuffer userInfo = new StringBuffer();
		System.out.println("Sending add user");
		userInfo.append(localUser.getUsername());
		userInfo.append(";");
		userInfo.append(localUser.getDomain());
		userInfo.append(";");
		userInfo.append(localUser.getResource());
		userInfo.append(";");
		userInfo.append(";");
		userInfo.append(";");
		userInfo.append(";");
		userInfo.append("\r\n");

		if (localUser == null || !localUser.isLoggedIn()) {
			System.out.println("User not loggedin");
			return false;
		} else {
			XMPPConnection localUserConnection =
localUser.getXMPPConnection();
			// FIXME Hardcoded port
			String port = "80";
			String jid = room.getOwner().getJid();
			HttpResponse res = null;
			int response = 0;
			XMPPSocket xs;
			try {
				xs = new XMPPSocket(localUserConnection, jid, port);
				HttpRequest req = new HttpRequest("POST", "/"
						+ URLEncoder.encode(room.getName(), "UTF-
8")
						+ "/addUser/");
				InputStream is = xs.getInputStream();
				OutputStream os = xs.getOutputStream();
```

```
                                req.setBody(userInfo.toString().getBytes());
                                req.write(os);
                                res = new HttpResponse(is);
                                response = res.getStatusCode();
                                xs.close();
                        } catch (XMPPException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        } catch (InterruptedException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        } catch (IOException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
                        if (response == 200) {
                                return true;
                        } else {
                                return false;
                        }
                }
        }

        private RoomImpl getRoom(String line, RoomModel roomModel) throws IOException {
                String[] tokens = line.split(":");
                String roomName = tokens[0];
                String ownerUser = tokens[1];
                User roomOwner = createOwner(ownerUser);
                IFolder folder = roomModel.getRoomFolder().getFolder(roomName);
                String iconPath = null;
                return new RoomImpl(roomModel, roomName, folder, iconPath, roomOwner);
        }

        private String getRoomName(String line) {
                String[] tokens = line.split(":");
                String roomName = tokens[0];
                return roomName;
        }

        private UserImpl createOwner(String ownerUser) {
                String[] tokens = ownerUser.split(";", 7);
                String username = tokens[0];
                String domain = tokens[1];
                String resource = tokens[2];
                String name = tokens[3];
                String surname = tokens[4];
                String email = tokens[5];
                String avatarPath = tokens[6];
                UserImpl userImpl = new UserImpl(username, resource, domain, name,
surname, email);
                return userImpl;
        }

}
```

5 Application layer (Sironta)

As an example of application which takes advantage of all new Servent features we include Sironta. Sironta has been designed as a Collaborative Working Environment, following OPAALS users needs and enriched by Gartner Group set of requirements for such kind of applications.

5.1 Introduction

The Servent is the basic building block of the sironta application. As such, it provides facilities for the following:

- Allow an arbitrary user to log into an arbitrary XMPP server
    - Allow an arbitrary user to register into an arbitrary XMPP server
- Allow a registered user to add buddies to his roster
- Notify a logged in user of connection/disconnection of his buddies
- Allow a logged in user to export services.
- Allow a logged in user to call exported services
    - Search of services may need support on the server side

The Servent performs all of its chores by calling certain operations on the XMPP network layer.

5.1.1 XMPP mappings

The Sironta application abstractions are mapped onto XMPP entities in the following way:

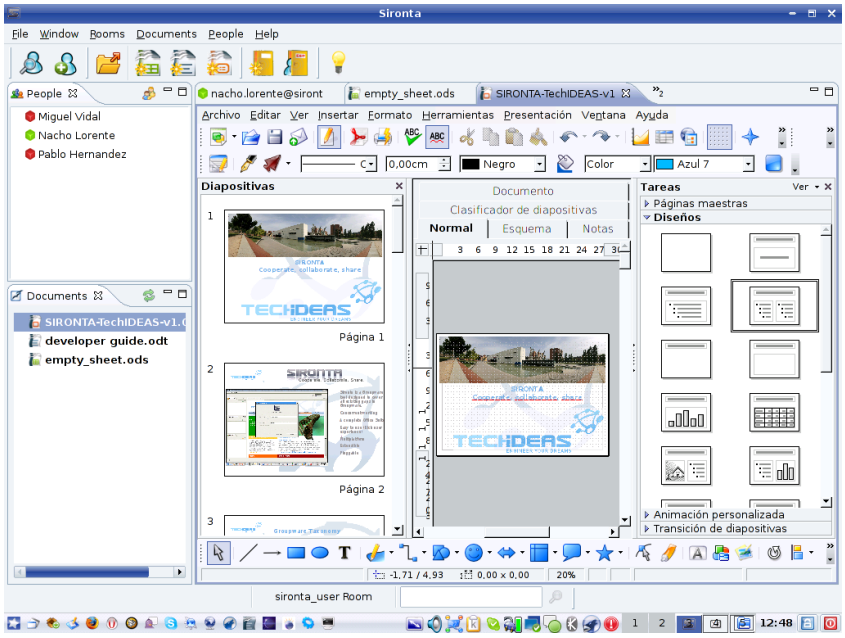
- A Sironta user is an XMPP user
- A Sironta room can be modelled in one of two ways
    - Persistent rooms in the server
        - It takes up space in the server
        - It may be easier to implement, if an XMPP room information can hold the list of users, the list of documents, etc...
    - Non-persistent rooms whose information is held by the owner/inhabitants of the room
        - No information can be recovered from the room if none of its inhabitants is on-line
    - A mixture of the two
        - Try to get the BEST of both worlds, without paying a much too high a price
- A Sironta document is just a bunch of bytes that is associated to a room.
    - It is stored in the disk of an inhabitant of a room
    - It is replicated among the inhabitants of the room
    - Information about the available documents can be held in the room information data structure. This approach lets to:
        - Store in the Server
        - Store in the disk of the room inhabitants

5.1.2 Registration & Login

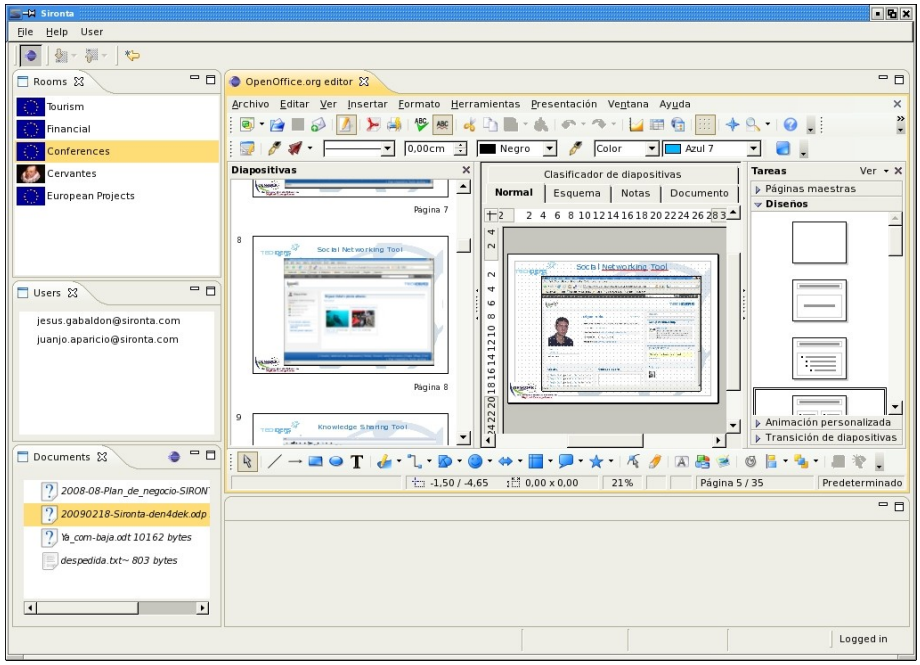
- The registration of a user into an XMPP server is performed in the usual way (XMPP wise). It is important to track the number and information about registered users, so it makes sense to allow a user to register for free, but not let its account to be active until a nonce is sent by email. Inactive accounts expire after a period (say, 24 hours)
- The login of a user into an arbitrary XMPP server is performed in the usual way (XMPP wise). The validation of the user must be synchronized with the registration process. It opens the compatibility with any client already written for XMPP and any Operating System.

5.2 New user interface

Version 0.5 user interface:



Version 0.8 user interface:



The new user interface provides better integration with the desktop (in Windows, MacOS and Linux) supporting Drag & Drop, Cut & Paste, etc... Supports a graphical representation for rooms and users avatar.

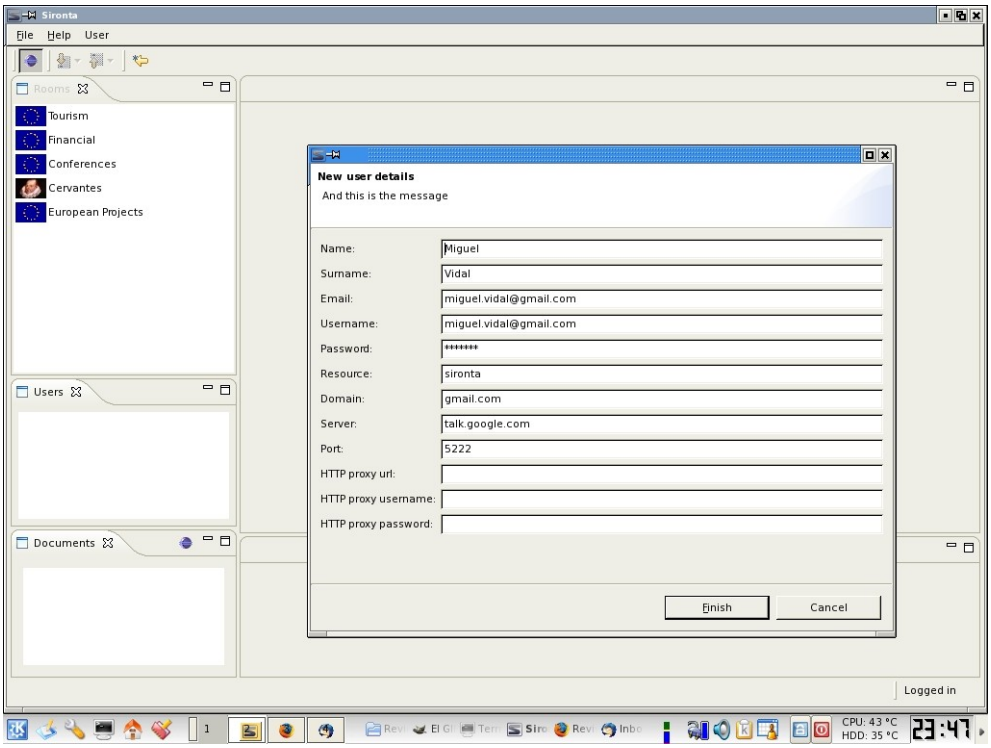
The performance of the graphical part has been dramatically improved, with a much better response time versus the previous one. To achieve this level of improvement, a complete redesign of the internal processes has been made.

5.3 Improved boot strapping

One of the main problems identified trough the interaction with SME's is the need of a easy and reliable bootstrapping mechanism. In previous projects, the strategy adopted was to deploy our (as a project) own infrastructure (i.e. network overlay). Due to the difficulties to create and maintain a huge community, the strategy adopted with this version of the Servent is to make it compatible with pre-existing network overlays.

As the open protocol adopted is XMPP, we're now in a situation in which, any XMPP account can be used from the tool. In other words, the project is in a situation in which the infrastructure from Jabber and Google can be used as P2P overlay (and any other XMPP network deployed as well).

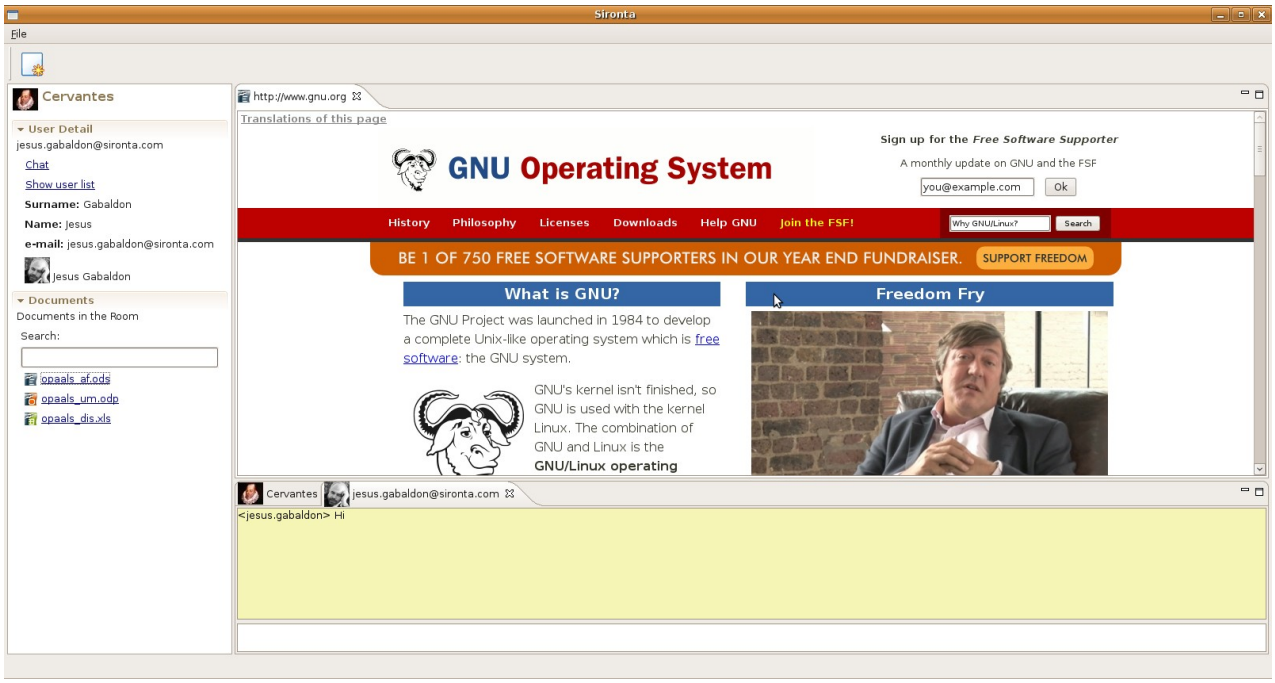
From the user interaction perspective, it is translated in the fact that now it is provided the capability of using any of the already existing XMPP user accounts (if any)



5.4 Future work

The current development of Sironta is very near of its final one, in which it would be considered fully operational and ready for a massive deployment. However it may be needed to change the implementation of some already existing components, with special emphasis in usability. In the next months roadmap these are the foreseen tasks:

5.4.1 Enhancement of the User Interface



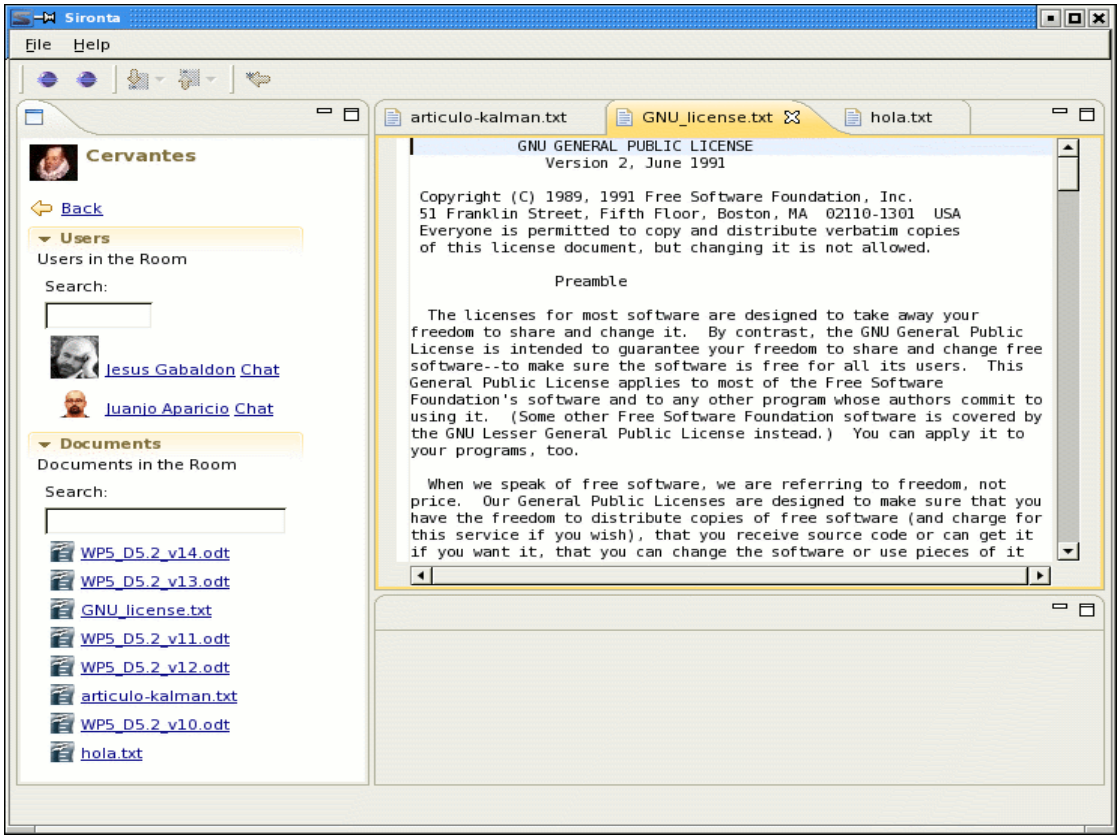
Thanks to the brand new internals of the user interface, it is now possible to perform further improvements in the user interaction with the application. Some trials have been made in this sense, just to improve the user experience and make Sironta more usable, handling events in the mouse right button, including context aware messages and behaviour, etc...

5.4.2 Improvements in document management

With Sironta the concept of folder, directory, etc... is something that should belong to the past. The interaction with the documents should be something "intelligent" and the tool have to be able to provide the right document in the right moment.

The experience teach us that a single document could belong to many projects, classifications, etc... what means that there are not a single folder where to store it. Instead, if we proper "tag" such document, multiple tags (plus its content) can characterize completely and accurately the document.

Following this design principles, Sironta should provide new methods of tagging and searching documents. Sharing the same philosophy, new ways of user searching are under research, to enrich the simple name finding.



5.4.3 New value added services

There are a set of services claimed by the early Sironta users that are in the process of being integrated, once finished the new network layer. These are:

- **Integration of VoIP** - Improving user interaction beyond what chatting offers. Voice interaction will make work cooperation more flexible and productive.
- **Videoconferencing** - Just another "turn to the screw" in better user interaction.
- **Whiteboard** - This service will provide the possibility of writing in the same sheet together one or more users to achieve a common understanding. It could be implemented as a one to many (for lecturers and teaching proposes) or many to many (to enable brain storming).
- **Remote Desktop** - For help-desk applications, is the easy approach to provide the best support to end users.

6 References

- Roy Thomas Fielding, “Architectural Styles and the Design of Network-based Software Architectures”, 2000 (http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm)
- Simple traversal of UDP over NATs (RFC3489 "http://tools.ietf.org/html/rfc3489" and RFC5389 "http://tools.ietf.org/html/rfc5389")
- XMPP Standards Foundation: http://xmpp.org/
- World Wide Web Consortium (W3C): http://www.w3.org
- Java Business Integration: http://java.sun.com/integration/
- OSGi: http://www.osgi.org/
- Eclipse: http://www.eclipse.org/
- Organization for the Advancement of Structured Information Standards (OASIS): http://www.**oasis**-open.org
- Autonomic Computing, by IMB: http://www.research.ibm.com/autonomic/
- SOA and standards readings: http://blogs.sun.com/alur
- Interoperability research for networked enterprises applications and software, Interop Network of excellence
  - http://interop-vlab.eu/
  - http://cordis.europa.eu/fetch?CALLER=PROJ_IST&ACTION=D&DOC=1&CAT=PROJ&QUERY=011ad92481eb:4fed:70b3db02&RCN=71148
  - Deliverable 6.1: practices, principles and patterns for interoperability
  - Deliverable 12.4: report on future research to support interoperability for SMEs
  - Deliverable DAP1-b: state of the art for interoperability architecture approaches. DAP (Domain Architecture and Platforms). Model driven and dynamic, federated enterprise interoperability architectures and interoperability non-funcional aspects
- Advanced technologies for interoperability of heterogeneous enterprise networks and their applications, Ahtena IP project
  - http://cordis.europa.eu/fetch?CALLER=PROJ_IST&ACTION=D&DOC=2&CAT=PROJ&QUERY=011ad94920b5:ca6f:7d0e5601&RCN=72762
  - http://www.athena-ip.org
  - Deliverable: Enterprise Interoperability - Research Roadmap
- International Conference on service oriented computing (ICSOC): http://cgi.cse.unsw.edu.au/~soc/icsoc08/
- Digital Business Ecosystem
  - Project: http://www.digital-ecosystem.org/
  - Regional: http://dbe.ita.es