Project Acronym:	**OPTIMIS**
Project Title:	**Optimized Infrastructure Services**
Project Number:	**257115**
Instrument:	**Integrated Project**
Thematic Priority:	**ICT-2009.1.2 – Internet of Services, Software and Virtualisation**

Trust Framework User Guide

Activity 3: *Service Deployment*

WP 3.3: *Trust Framework*

Due Date:		M34
Submission Date:		31/03/2013
Start Date of Project:		01/06/2010
Duration of Project:		36 months
Organisation Responsible for the Deliverable:		Atos
Version:		1.0
Status		Draft
Author(s):	Francisco Javier Nieto	Atos
Reviewer(s)		

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

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
1.0	31/02/2013	Final version	Francisco Javier Nieto

Table of Contents

Index of Figures

Index of Tables

No table of figures entries found.

1 Introduction

This document includes the user guide for the software component Trust Framework. The document presents those main functionalities provided by the component, as well as some known limitations related to these functionalities.

1.1 Glossary of Acronyms

Acronym	Definition
D	Deliverable
DRS	Document Review Sheet
EC	European Commission
HM	Holistic Manager
IP	Infrastructure Provider
PM	Project Manager
PO	Project Officer
SLA	Service Level Agreement
SP	Service Provider
TF	Trust Framework
TREC	Trust, Risk, Eco-efficiency & Cost
WP	Work Package
WS	Web Service

2 Trust Framework User Guide

2.1 Release information

Component Name	Release Number	Release Date
Trust Framework	3.0	2013-03-31

2.2 Introduction

The trust model defined in OPTIMIS has been evolving during the different releases of the component. It retrieves information from the Monitoring component about resources usage, as well as SLA monitoring data, legal aspects and previous trust values.

The result is a trust model divided in two parts: IP side (which calculates the trust of SPs deploying services) and SP side (which calculates de trust of those IPs where its services are deployed and running). More details about the aspects considered can be found in the detailed design [3].

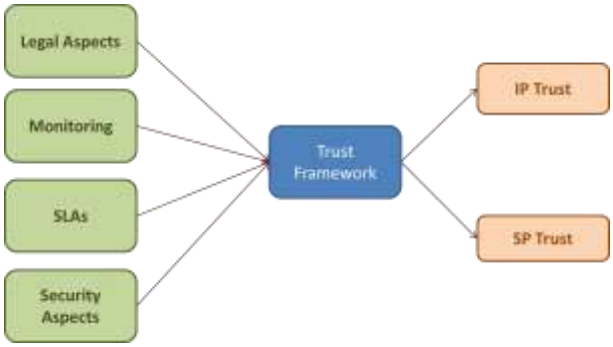

Figure 1 Y3 Trust Framework

The different aspects of each part of the model are combined by using a fuzzy model which provides a comprehensive trust result. As in models such as CMMI, there are different trust levels (from 0 to 5), each level requiring minimum values for certain aspects. Those aspects related to legal and SLA properties, together with resource usage, are considered the most important. This means that, for instance, for obtaining high trust values, it will be mandatory to have high values for legal requirements fulfillment, for SLA fulfillment and resources usage.

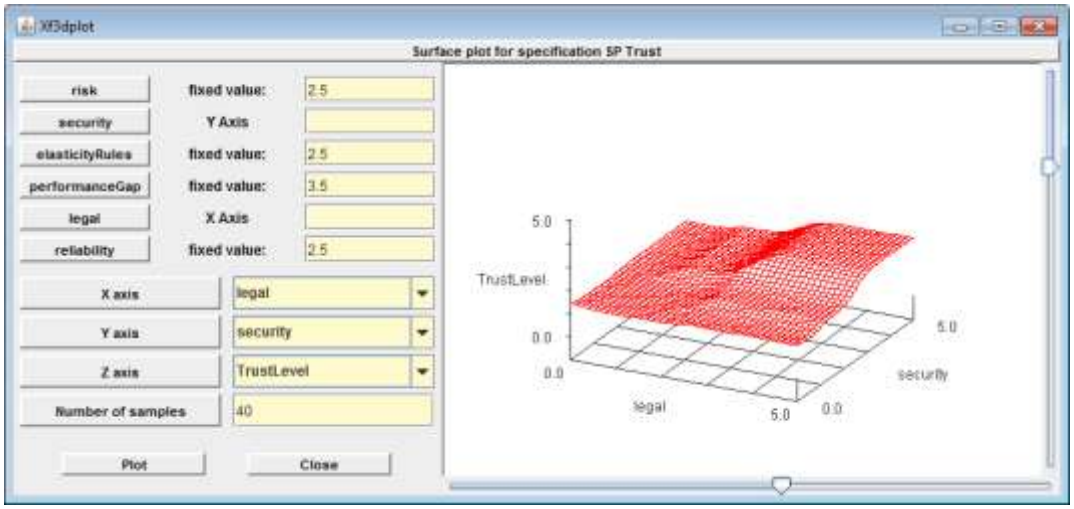

Figure 2 Simulation of the fuzzy model for aggregating SP Trust

In case that one of these aspects is low or very low, the trust obtained will be medium or low, depending on the evaluation of other aspects which could decrease the result even more. There are files containing the fuzzy rules which can be modified, in case users want to adapt the component behavior to their criteria.

It is also important to understand that, in some cases, the Trust Framework performs forecasts based on previous measurements (i.e. for resources usage forecasting). As explained in the limitations section, it is necessary to have enough data for obtaining good predictions because of the algorithm used (Holt-Winters).

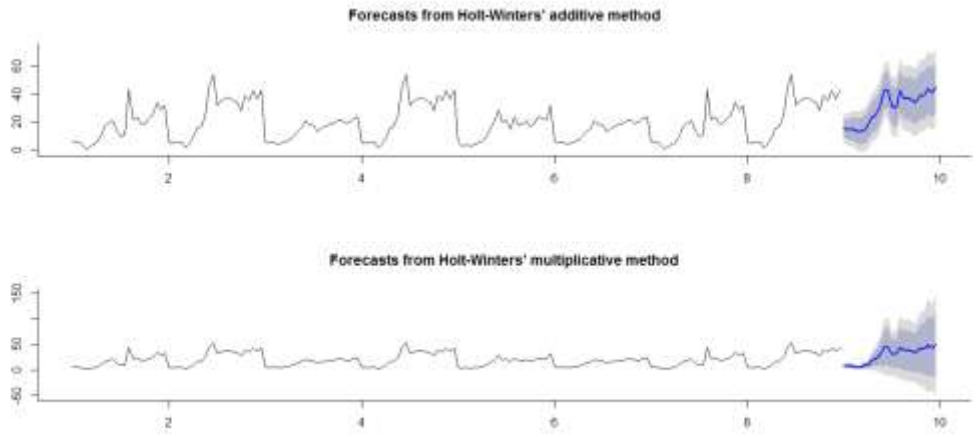

Figure 3 Example of CPU usage forecast using nine days data

Usually, even if with 48 hour data is enough, the model would provide inaccurate results during weekends or holidays, being necessary, at least, two weeks of data (this means, to have a service running during more than two weeks) for obtaining better results.

2.3 Functionalities

The main functionality of the trust framework is the provision of an accurate trust assessment value that helps the decision maker components of OPTIMIS to take the correct decisions about a service during the whole life-cycle of a service.

The component released in Y2 of the project (release version 2.0), already provided some functionalities which have been maintained and extended in the last version (release 3.0). For more information about how these functionalities, please, refer to previous users' guides [2]. In the following sub sections, we present the new functionalities added to the component.

2.3.1 Get IP Self-Assessment

As a way to anticipate problems and enable the HM to take some concrete actions on the own IP, there is a new functionality which calculates the trust for the IP which is hosting the Trust Framework and other Optimis components.

Operation	GET
Name	getSelfAssessment
URI	/trustframework/deploy/ip/ getSelfAssessment/{providerId}
Description	Provides the trust of the IP in which the TF is running
Input	Provider identifier
Output	Trust calculated for the IP
Errors	

2.3.2 Trust Proactive Behavior

One of the new features of the Trust Framework is the capability to determine proactively if a service or a provider is not performing as expected, notifying the HM component, so it can perform proper mitigation actions.

For doing so, an entity (a subscriber) needs to ask the Trust Framework to watch proactively concrete services or providers. The thread which is calculating trust periodically will check the entities and thresholds provided, raising notifications to the HM when the threshold has been exceeded.

Operation	PUT
Name	setProactiveTrustAssessor
URI	/trustframework/common/ip/setproactivetrustAssessor/{entityId}
Description	Sets up an internal alert for the provided entity id (service/provider) with the given threshold
Input	Entity identifier (service or provider) Threshold which represents the max value accepted Type (Provider=0; Service=1)
Output	Returns true if the alert was set correctly
Errors	

Once a subscriber wants to remove the alerts set for an entity (and such stopping the proactive behavior of the Trust Framework), it is necessary to invoke the TF, which will delete all the alerts for a given entity (service or provider).

Operation	DELETE
Name	stopProactiveTrust
URI	/trustframework/common/ip/stopproactivetrust/{entityId}
Description	Removes alerts set for the entity provided
Input	Entity identifier (service or provider)
Output	Returns true if the alert was removed correctly
Errors	

2.3.3 Trust Forecasting for the HM

The HM component needs that TREC components provide some forecasting information, in order to understand how some concrete actions would affect the TREC results, as an indicator of the future performance which can be expected. The TF has added some new functionalities for providing forecasting as a way to support the HM.

There will be concrete forecasts for services and IP trust, so it will be possible to know the expected value for trust in the moment requested in the future.

Operation	GET
Name	forecastServiceTrust
URI	/trustframework/common/ip/forecast/forecastServiceTrust/{serviceId}
Description	Provides the a forecast for a service trust in the given timespan
Input	Service identifier

	Timespan (how much time in the future)
Output	Trust calculated for the service
Errors	

Operation	GET
Name	forecastIPTrust
URI	/trustframework/common/ip/forecast/forecastIPTrust/{providerId}
Description	Provides the trust level forecast for the provider asked
Input	Service identifier Timespan (how much time in the future)
Output	Trust calculated for the IP
Errors	

There are also other forecasts related to the deployment of new services and the creation/removal of VMs for a concrete service, in order to understand the effect they would have in the trust calculation.

Operation	POST
Name	forecastServiceDeployment
URI	/trustframework/common/ip/forecast/forecastServiceDeployment
Description	Provides the trust level forecast for the new service with the indicated manifest
Input	The service manifest of the service to be deployed
Output	Trust forecast calculated for the service
Errors	

Operation	POST
Name	forecastServiceDeploymentIP
URI	/trustframework/common/ip/forecast/forecastServiceDeploymentIP
Description	Provides the trust level forecast for the IP if a new service is deployed with the indicated manifest
Input	The service manifest of the service to be deployed
Output	Trust forecast calculated for the IP
Errors	

Operation	GET
Name	forecastVMDeployment
URI	/trustframework/common/ip/forecast/forecastVMDeployment/{serviceId}
Description	Provides the trust level forecast for the service asked if a new VM is created
Input	The service identifier
Output	Trust forecast calculated for the service
Errors	

Operation	GET
Name	forecastVMDeploymentIP
URI	/trustframework/common/ip/forecast/forecastVMDeploymentIP/{serviceId}
Description	Provides the trust level forecast for the current IP if a new VM is created for a service.
Input	The service identifier
Output	Trust forecast calculated for the IP
Errors	

Operation	GET
Name	forecastVMCancellation
URI	/trustframework/common/ip/forecast/forecastVMCancellation/{serviceId}
Description	Provides the trust level forecast for the service asked if a VM is removed
Input	The service identifier
Output	Trust forecast calculated for the service
Errors	

Operation	GET
Name	forecastVMCancellationIP
URI	/trustframework/common/ip/forecast/forecastVMCancellationIP/{serviceId}
Description	Provides the trust level forecast for the current IP if a VM is removed for a service
Input	The service identifier
Output	Trust forecast calculated for the IP
Errors	

2.4 Known limitations

These limitations are known to exist in this release of the software:

2.4.1 Forecasts Accuracy with Few Data

In the last version of the component, we have introduced a new algorithm for calculating forecasts called Holt-Winters (also known as triple exponential smoothing). This new algorithm provides interesting capabilities on terms of identifying seasonality, which usually happens in resources consumption at IPs, providing better forecasting results than simple exponential smoothing.

The main issue is the algorithm needs enough data, so that seasonality is identified correctly. This means that, the minimum data needed for using Holt-Winters is data of a service running during, at least, 48 hours. But the services activity uses to vary also during weekends, which means that for obtaining a minimum accuracy, TF would need more than 2 weeks of data.

In those cases in which not enough data is available (i.e. a service running for less than 48 hours), the Trust Framework will apply simple exponential smoothing, assuming a certain loss of accuracy.

2.5 Getting Started

For the following sections of how to use the software, test, configuration please go to the Trust Framework Installation Guide document [1].

2.5.1 Using the Software

The component can be used by invoking the APIs through HTTP calls according to the information provided in section 2 (i.e. Poster plugin for the Firefox browser). It is also possible to use the Java client provided by the TF, importing it as a library in new projects. More information can be found in the Trust Framework Installation Guide [1] and previous users' guides [2].

2.5.2 Testing the Software

Information can be found in the Trust Framework Installation Guide [1].

2.5.3 Configuration

Information can be found in the Trust Framework Installation Guide [1].

2.6 FAQ

Information can be found in the Trust Framework Installation Guide [1].

2.7 Other information

2.7.1 Source Code Information

The source code information for the trust framework can be found at:

```
http://pandora.atosorigin.es/svn/optimis/branches/OptimisY3/TrustFramework
```

2.7.2 Directory Structure

The Trust Framework is divided in 5 different logic sections based on the capabilities and functionalities they provide:

- TrustFrameworkService/IPTrustFramework: This module contains the main logic of the component which calculates the trust for SPs, since it is installed in the IP side.
- TrustFrameworkService/SPTrustFramework: This module contains the main logic of the component which calculates the trust for IPs, since it is installed in the SP side.
- TrustFrameworkDB/iptrustdb: DAO objects for performing operations in the IP database, depending on the TREC Common database.
- TrustFrameworkDB/sptrustdb: DAO objects for performing operations in the SP database, depending on the TREC Common database.
- TrustFrameworkClients: This small component is a client which is able to access programmatically to the REST APIs provided by IPTrustFramework and SPTrustFramework.

Configuration and log files for the components are placed in a common place agreed with the rest of the Optimis components. In the case of the Trust Framework:

- /opt/optimis/etc/iptf: Contains configuration files for IPTrustFRamework (Hibernate, log4j and properties);
- /opt/optimis/etc/sptf: Contains configuration files for SPTrustFRamework (Hibernate, log4j and properties);

- /opt/optimis/var/log/TrustFramework: Contains log files for the Trust Framework. Each file contains a different detail level.

2.7.3 Contributors

The following people contributed to the implementation of the Trust Framework:

- Juan Luis Prieto
- Francisco Javier Nieto
- Pramod Pawar
- Mariam Kiran

3 References

[1] TrustFramework_Installation_Guide, D3.3.2.3 of OPTIMIS project.

[2] TrustFramework_User_Guide, D3.3.2.2 of OPTIMIS project.

[3] TrustFramework Detailed Technical Design, D3.3.1.3 of OPTIMIS project.