# Service Level Enforcement Policy Usage Scenarios for Policy Manager

SOA | software™

## Copyright

## Trademarks

## SOA Software, Inc.

SOA Software, Inc.
12100 Wilshire Blvd, Suite 1800
Los Angeles, CA 90025
(866) SOA-9876
www.soa.com
info@soa.com

## Disclaimer

The information provided in this document is provided "AS IS" WITHOUT ANY WARRANTIES OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. SOA Software may make changes to this document at any time without notice. All comparisons, functionalities and measures as related to similar products and services offered by other vendors are based on SOA Software's internal assessment and/or publicly available information of SOA Software and other vendor product features, unless otherwise specifically stated. Reliance by you on these assessments / comparative assessments is to be made solely on your own discretion and at your own risk. The content of this document may be out of date, and SOA Software makes no commitment to update this content. This document may refer to products, programs or services that are not available in your country. Consult your local SOA Software business contact for information regarding the products, programs and services that may be available to you. Applicable law may not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

# Contents

# Service Level Enforcement Policy Usage Scenarios (Policy Manager-specific)

This document provides a list of Policy Manager-specific usage scenarios for the *Service Level Enforcement Policy*.

## Scenario 1: Block Request to Service

Block request to service for 15 minutes after 300 requests SLA has been violated. Application receives the Fault message defined in the *Service Level Enforcement Policy*.

1 Create a physical service in the *Policy Manager Management Console* using **Create Physical Service**.

2 Provide service details and **Finish** the wizard.

3 Using **Virtualize Service**, virtualize and host the physical service on Network Director (ND1), and assign a name (e.g., **Vs1**).

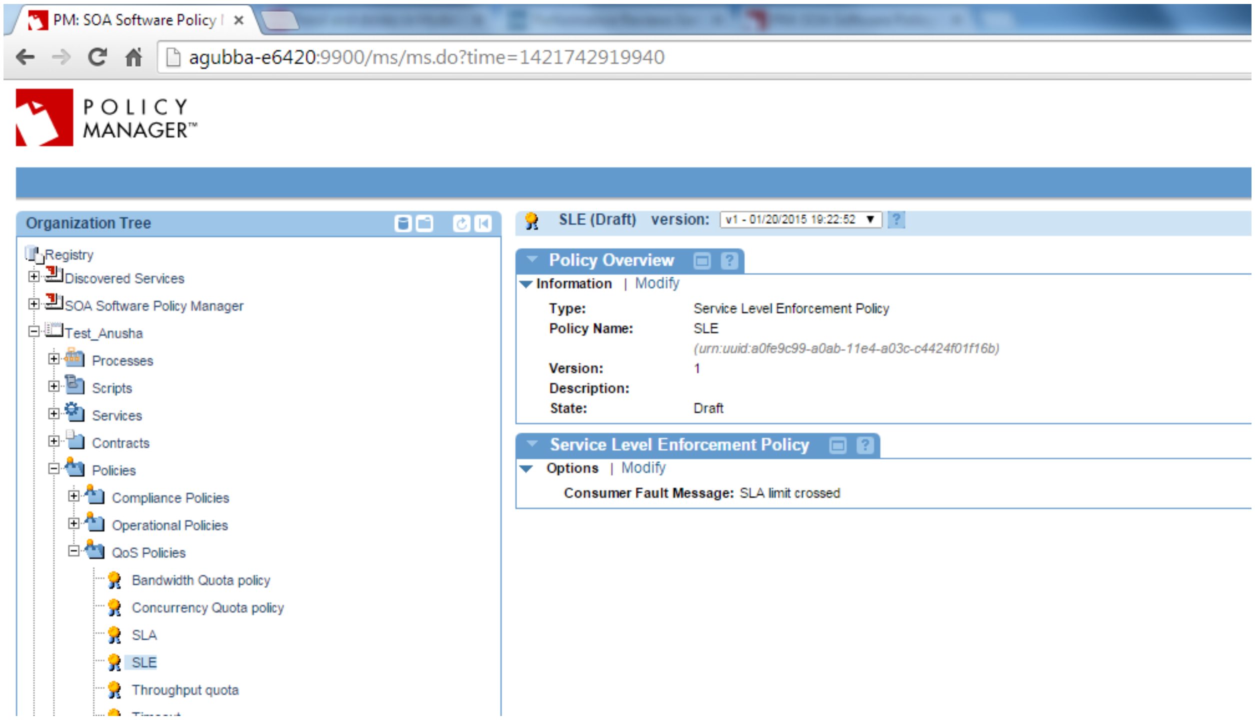
4 Navigate to *Organization > Policies > QoS Policies* and use **Add Policy** to create a *Service Level Enforcement Policy*.

5 Configure the *Service Level Enforcement Policy* and define a custom Fault message that reflects the use case scenario purpose.

6 Next, use **Add Policy** and create a *Service Level Policy*.

7 Configure this policy as per the use case with a Custom alert for **Usage Count > 300** in a 15 minute interval.

8 Attach the Service Level Enforcement policy and Service Level policy to **Vs1** service in the *Service Details > Policy Attachments > QoS* section.



SLE (Draft) version: v1 - 01/20/2015 19:22:52

Type: Service Level Enforcement Policy
Policy Name: SLE
*(urn:uuid:a0fe9c99-a0ab-11e4-a03c-c4424f01f16b)*
Version: 1
Description:
State: Draft

Consumer Fault Message: SLA limit crossed

9 Send more than 300 requests to **Vs1** from application/client.

10 The first 300 requests will be successful, and subsequent requests will fail and generate the custom Fault message defined in the *Service Level Enforcement Policy*.

# Scenario 2: SLA Clear Alert

SLA generates a clear alert and sends an email to alert the administrator(s) after the App has been denied access for 15 minutes.

1 Create a physical service in the *Policy Manager Management Console* using **Create Physical Service**.

2 Provide service details and **Finish** the wizard.

3 Using Virtualize Service, virtualize and host the physical service on Network Directory (**ND1**), and assign a name (e.g., **Vs1**).

4 Navigate to *Organization > Policies > QOS Policies* and use **Add Policy** to create a *Service Level Enforcement Policy*.

5 Configure the *Service Level Enforcement Policy* as per the use case a with a custom Fault message.

6 Navigate to *Organization > Policies > QOS Policies* and use **Add Policy** to create a *Service Level Policy*.

7 Configure the *Service Level Policy* as per the use case with a custom alert for **Usage Count > 300** in a 15 minute interval.

8 Attach the *Service Level Enforcement Policy* and *Service Level Policy* to the **Vs1** service in the *Service Details > Policy Attachments > QoS* section.

9 Send more than 300 requests to **Vs1** from application/client.

10 The first 300 requests will be successful and subsequent requests will fail and generate the custom Fault message defined in the *Service Level Enforcement Policy*.

11 After the alert is generated, if the SLA is not violated again, a clear alert will be generated.

12 If the alert is configured with an email, it will be sent to the assigned users when triggered.

## Scenario 3: SLA Reset

SLA has been reset and the App can send to the API.

1 Create a physical service in *Policy Manager Management Console* using **Create Physical Service**.

2 Provide service details and **Finish** the wizard

3 Using Virtualize Service, virtualize and host the physical service on Network Director (**ND1**), and assign a name (e.g., **Vs1**),

4 Navigate to *Organization > Policies > QOS Policies* and use **Add Policy** to create a *Service Level Enforcement Policy*.

5 Configure the *Service Level Enforcement Policy* as per the use case a custom Fault message

6 Navigate to *Organization > Policies > QOS Policies* and use **Add Policy** to create a *Service Level Policy*.

7 Configure the *Service Level Policy* as per the use case with custom alert for **Usage Count > 300** in 15 minute intervals.

8 Attach the *Service Level Enforcement Policy* and *Service Level Policy* to **Vs1** service in the *Service Details > Policy Attachments > QoS* section.

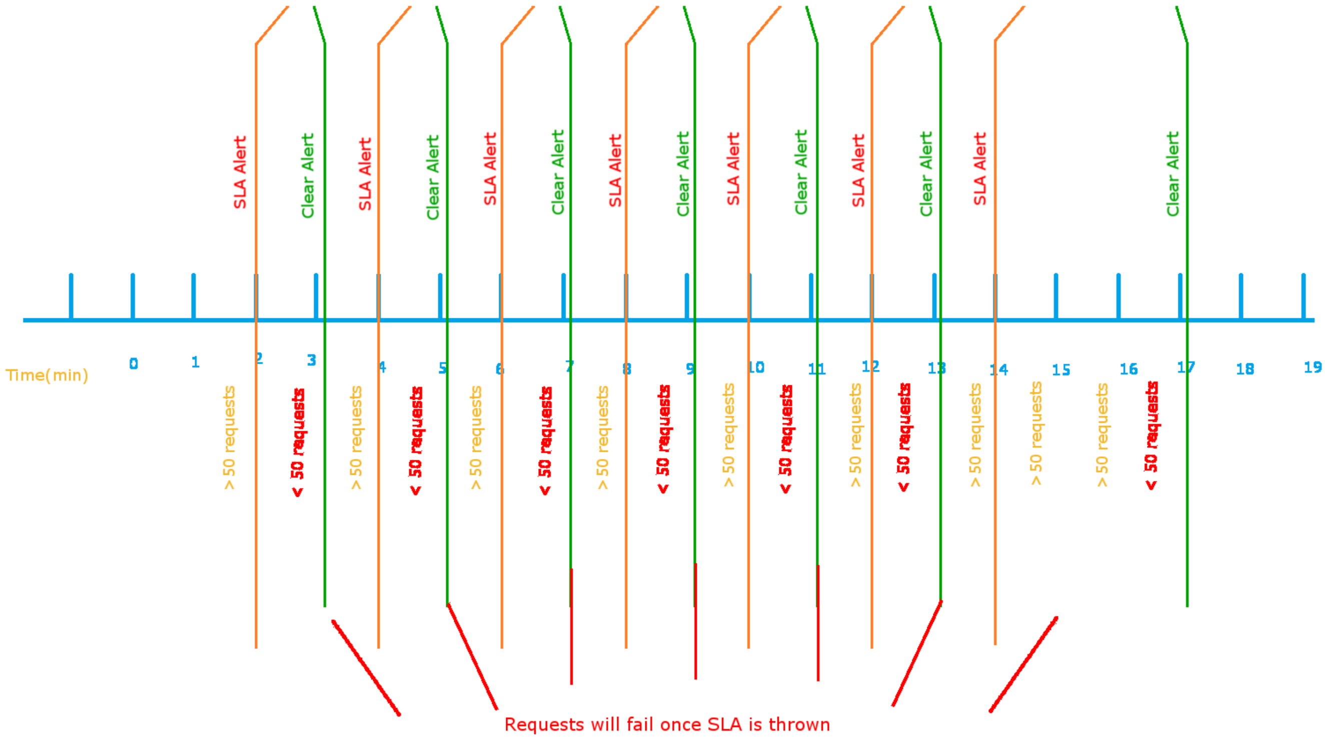
9 Send more than 300 requests to **Vs1** from application/client.

10 The first 300 requests will be successful and subsequent requests will fail  and generate the custom Fault message in *Service Level Enforcement Policy*.

11 After the alert is generated, if the SLA is not violated again, a clear alert will be generated.

12 After clear alert is generated the requests should be successful until it violates the SLA again.

# ***Timeline***


SLA Alert
Clear Alert
SLA Alert
Clear Alert
SLA Alert
Clear Alert
SLA Alert
Clear Alert
SLA Alert
Clear Alert
SLA Alert
Clear Alert
SLA Alert
Clear Alert
Time(min)
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
> 50 requests
< 50 requests
> 50 requests
< 50 requests
> 50 requests
< 50 requests
> 50 requests
< 50 requests
> 50 requests
< 50 requests
> 50 requests
< 50 requests
> 50 requests
> 50 requests
> 50 requests
< 50 requests
Requests will fail once SLA is thrown