

HP Best Practices

Software Version: 2.10

End-to-End Service Monitoring and Event Management

Document Release Date: December 2013



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© Copyright 2005 - 2013 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: <http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: <http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at: <http://www.hp.com/go/hpsoftwaresupport>

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

HP Software Solutions Now accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is <http://h20230.www2.hp.com/sc/solutions/index.jsp>

Contents

End-to-End Service Monitoring and Event Management	1
Contents	3
Welcome to This Guide	6
How This Guide is Organized	6
Who Should Read This Guide	6
Additional Online Resources	7
Part I: End-to-End Service Monitoring in the IT Environment	8
Chapter 1: Introduction	9
Overview	9
Use Cases	10
Chapter 2: Business Layer Monitoring – EUM	13
Overview	13
Tools	13
Installation and Configuration	15
Recommendations	16
Monitoring	16
Deployment	17
Configuration	18
Chapter 3: Software Layer Monitoring	19
Overview	19
Tools	19
Installation and Configuration	20
Recommendations	22
Monitoring	22
Deployment	23
Severity Mapping	23
Chapter 4: Infrastructure Layer Monitoring	24
Overview	24
Tools	24

Installation and Configuration	25
Recommendations	26
Monitoring	26
Deployment	26
Severity Mapping	27
Virtual Infrastructure Monitoring	28
Overview	28
Tools	28
Installation and Configuration	29
Recommendations	29
Using HP SiteScope Solution Template	29
Using HP Operations Manager Virtual Infrastructure Smart Plug-in	30
VMware	31
Microsoft Hyper-V Server	33
IBM LPAR/WPAR	34
HPVM	35
Oracle Solaris (Container)	36
Network Monitoring	37
Overview	37
Tools	37
Installation and Configuration	38
Recommendations	39
Deployment	39
Discovery	39
Monitoring	40
Configuration	40
Documentation	40
Part II: Event Management	41
Chapter 5: Introduction	42
Overview	42
What is an Event?	43

- Chapter 6: Event Management 44
 - Overview 44
 - Event Detection and Notification 49
 - Event Filtering 50
 - Overview 50
 - Configuration 51
 - Event Correlation 53
 - Overview 53
 - Health Indicators 54
 - Key Performance Indicators 55
 - Topology-based Event Correlation 56
 - Stream-based Event Correlation 57
 - Event Handling 58
 - Event Closure and Review 61
- Chapter 7: Using Event Management in the Detect to Correct Value Stream 62
 - Overview 62
 - Detect to Correct Value Stream Diagram 63
- Chapter 8: Managing Events 64
 - Overview 64

Welcome to This Guide

Welcome to the HP End-to-End Service Monitoring and Event Management Best Practices Guide. The goal of this document is to provide best practices for implementing smart end-to-end service monitoring solutions and an event management process to improve IT availability and performance. This document provides an overview of each product and the product's integrations. Additional information for each product is referenced in each section.

This chapter includes:

How This Guide is Organized	6
Who Should Read This Guide	6
Additional Online Resources	7

How This Guide is Organized

This guide contains the following parts:

Part I: End-to-End Service Monitoring in the IT Environment

Defines the end-to-end service monitoring deployment and implementation to ensure adherence to the level agreed upon by the service provider and the service consumer

Part II: Event Management

Defines the best practice for implementing an event management process

Who Should Read This Guide

This guide provides the guidelines and recommendations for planning and implementing end-to-end service monitoring and executing an event management process follow-up.

This guide is intended for:

- Customers
- Internal HP field personnel
- Partners
- Anyone responsible for planning and implementing an end-to-end service monitoring solution

The information in this guide may duplicate information available in other Best Practices documentation, but is provided here for convenience.

Additional Online Resources

Troubleshooting & Knowledge Base accesses the Troubleshooting page on the HP Software Support Web site where you can search the Self-solve knowledge base. Choose **Help > Troubleshooting & Knowledge Base**. The URL for this Web site is <http://h20230.www2.hp.com/troubleshooting.jsp>.

HP Software Support accesses the HP Software Support Web site. This site enables you to browse the Self-solve knowledge base. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help > HP Software Support**. The URL for this Web site is www.hp.com/go/hpsupport.

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:
http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport user ID, go to:
<http://h20229.www2.hp.com/passport-registration.html>.

HP Software Web site accesses the HP Software Web site. This site provides you with the most up-to-date information on HP Software products. This includes new software releases, seminars and trade shows, customer support, and more. Choose **Help > HP Software Web site**. The URL for this Web site is www.hp.com/go/software.

HP Software Solutions Now accesses the HPSW Solution and Integration Portal Web site. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this Web site is <http://support.openview.hp.com/sc/solutions/index.jsp>.

Part I: End-to-End Service Monitoring in the IT Environment

Chapter 1: Introduction

This chapter includes:

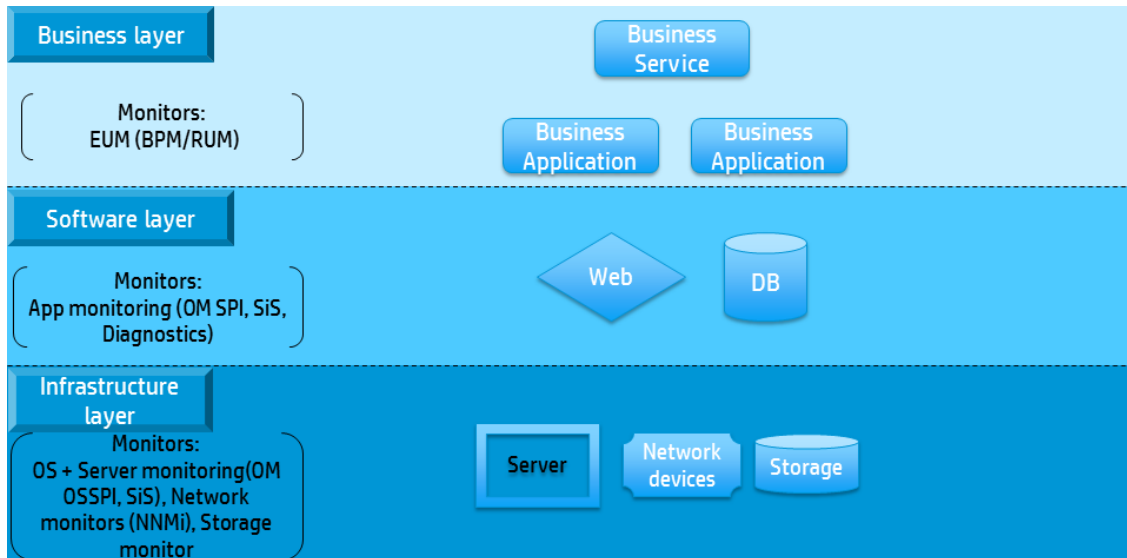
Overview	9
Use Cases	10

Overview

Part I: End-to-End Service Monitoring in the IT Environment provides our suggested best practices for deploying and implementing smart end-to-end service monitoring solutions to ensure adherence to the level agreed upon between the service provider and the service consumer. Feel free to use the entire best practice's solution, a mix of the various products, or just use a single product to address your monitoring needs.

Note: Comprehensive end-to-end service monitoring benefits the event management process, especially in the detection and correlation phases.

The following diagram illustrates how an IT services environment might look—illustrating the complexity of a contemporary business service, relying/depending on multiple infrastructure and network components, as well as with the software running on top of it. The organization responsible for this service benefits greatly when it can monitor and assess the status and performance of the components.



© Copyright 2012 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice.



A typical business service usually consists of the three layers as displayed in the diagram. Each of those layers can be monitored separately, providing insight into the status and performance of the corresponding aspect. The best results are achieved when all monitors are implemented and the

aggregated data is supplied to a central console to be accessible for further reporting and processing/analysis.

The central console is BSM OMi as part of the Service and Operations Bridge (SaOB), and is described in ["Event Management" on page 41](#) of this document.

- **Business layer.** In the Business layer, IT monitors the application itself, mainly by End-User Monitoring (EUM). It contains line of business (LOB), business services, and complex business applications—for example, an email service is a Business Service, and Microsoft Exchange Suite is a Business Application.
- **Software layer.** In the Software layer, IT monitors the software components that are installed on the servers that provide services to the application. It connects the Business layer to the Infrastructure layer, and contains all of the software components—for example, IIS software on a Client Access Server is a web application, and Microsoft SQL software on a Microsoft Exchange mailbox server is a database.
- **Infrastructure layer.** In the Infrastructure layer, IT monitors the infrastructure that is used by the Software layer—server, network, and other infrastructure services.
 - **Virtual Infrastructure Monitoring.** Virtualization enables dividing the computer resources into multiple execution environments. Virtual infrastructure is monitored differently than physical infrastructure, and this sub-chapter contains recommendations and guidelines on implementing an end-to-end service monitoring solution while working with a virtual infrastructure.
 - **Network Monitoring.** Network Monitoring is a major part of the IT infrastructure services that provides networking services to the IT environment—for example, network switch, routers, and so on. Most contemporary business services require an adequate network infrastructure to operate. This mandates special attention to the monitoring of network equipment and configuration to enable stable communications.

Each layer is divided into the following four sections:

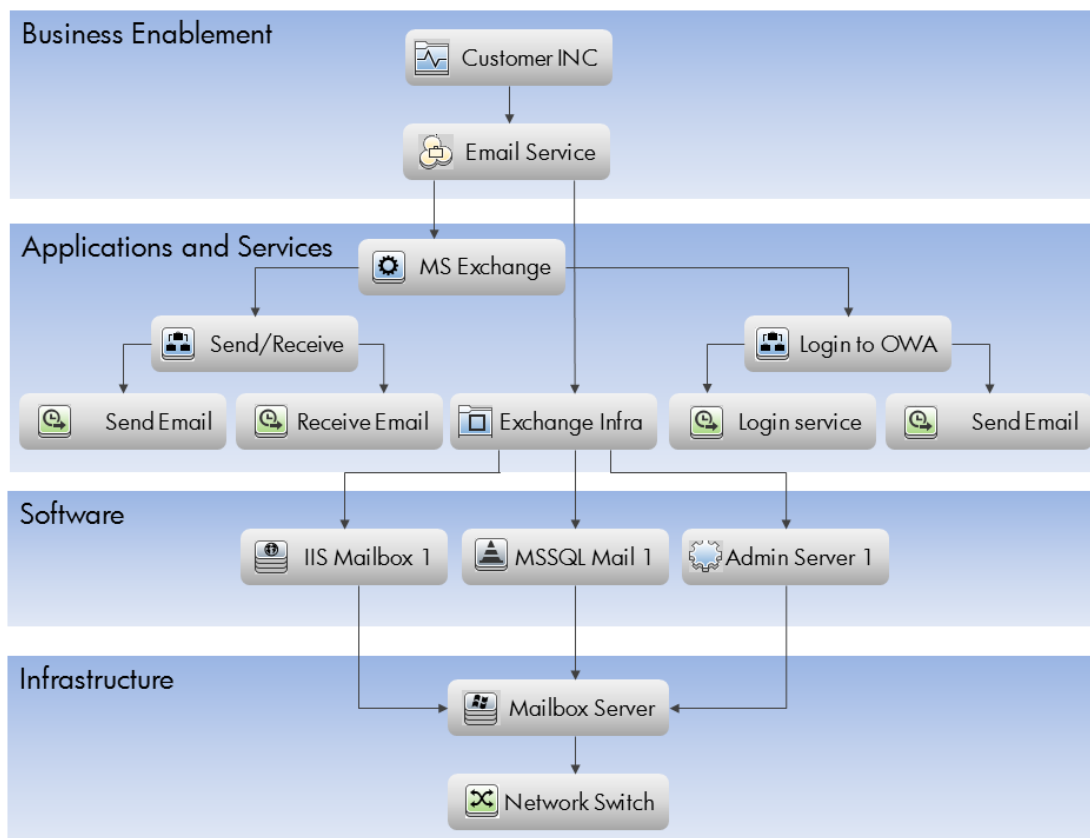
- **Overview.** Overview of what is being monitoring and why it is being monitored
- **Tools.** List of tools to be used for this type of monitoring
- **Installation and Configuration.** Flow of actions for applying the monitoring solution; including characterizing and configuring the tools and monitors
- **Recommendations.** Set of field best practice recommendations to help in effectively applying the monitoring solution

Use Cases






The IT organization provides a variety of complex services to the business. To ensure the agreed-upon service level, IT must monitor the availability and performance of these services in order to address potential errors that can affect the business.







Throughout this guide, we use the example of an email service as a common IT service in order to demonstrate the monitoring implementations being suggested in the guide. For an enterprise customer, an email service is a multi-tier, multi-component service usually spanning multiple geographies. The email service is based on the Microsoft Exchange (TM) Server, which contains several components [configuration items (CIs)]. This guide covers many aspects of an email service and applies monitoring solutions for every component.

The following CI diagram illustrates how an email service environment might look:



Configuration Items

Icon	Description
	Business function
	Business service
	Business application
	Business transaction flow
	Business transaction

Icon	Description
	Configuration item (CI) collection
	IIS web server
	SQL server
	Running software
	Windows (computer)
	Switch

The layers for the email service example consist of the following four layers:

- **Business Enablement layer.** The Business Enablement layer contains the business elements.
- **Application and Services layer.** In the Application and Services layer, IT monitors the application by end-user monitoring. In our example, IT monitors the user's ability to send and receive emails and the Outlook Web Access graphical user interface. These are critical functions of the application.
- **Software layer.** In the Software layer, IT monitors the software components that are installed on the servers and provides services to the application. In our example, we monitor the Microsoft Exchange application, IIS, and database software.
- **Infrastructure layer.** In the Infrastructure layer, IT monitors the infrastructure—server, network, and other infrastructure services of the Microsoft Exchange application. In our example, we monitor the server's file system, memory, CPU, and log files that must operate without any errors for the software to run properly on the servers. We also monitor the network traffic—availability and performance on the network switch that is connected to the Exchange mailbox servers.

Chapter 2: Business Layer Monitoring – EUM

This chapter includes:

Overview	13
Tools	13
Installation and Configuration	15
Recommendations	16

Overview

End-to-end service monitoring in the IT environment monitors the Business layer from the end-user's perspective—End-User Monitoring (EUM).

Monitoring end-user experience is the main purpose of EUM, which provides the ability to receive the up-to-date status (available/unavailable) and performance information on an application, and create reports on past problems as the user experiences it.

The starting point of EUM monitoring is defining a list of the most critical business processes and the steps within them that we want to monitor. For example, end-user monitoring of an email application confirms that the most important processes in the email service—send email, receive email, log on to the mailbox for the Outlook Web Access application, and so on—are available and perform according to the agreed-upon service level.

Tools

HP Business Service Management (BSM) is an application suite that provides a portfolio that contains a set of integrated management-layer solutions—each employing a different combination of HP products and applications.

The BSM suite contains two modules. One is used for service monitoring and is called Application Performance Management (APM). The second is used for event management and is called Operations Management (OMi). This module is discussed in "[Event Management](#)" on page 41.

The APM module includes two products which monitor the end-user experience—RUM and BPM.

- **HP Business Process Monitor (BPM)**. This is a synthetic user monitor that uses an active and location-based monitoring solution to simulate business processes for consistent, predictable measurement—identifying performance and availability problems before users experience them.

For example, when IT monitors send/receive email via BPM synthetic monitoring, they can identify when the customer is experiencing slowness while receiving emails from the Exchange servers, thus identifying a performance problem from within the emailing service.

- **HP Real User Monitor (RUM)**. RUM monitors the network traffic with the application generated by real users. The RUM product monitors application performance and availability for all users at all locations all the time based on passive listening of the network traffic.

For example, when IT monitors the action of send/receive email via RUM, they can identify when a branch in a specific geographic location is experiencing slowness while accessing their email accounts, even if the overall status of the emailing service looks okay. They can identify that they are experiencing network problems in the management floor and this is causing the performance problem.

BPM and RUM offer the following:

- **HP Business Process Monitor (BPM)**

- By running the predefined monitoring scripts on the BPM data collectors, we can monitor the application 24 hours a day/7 days a week by pretending to be a real user that is operating predefined actions in the application.
- Creates long-term trending reports to learn about the application's behavior using BSM's ability to gather data from the monitoring metrics.
- Provides proactive business application monitoring—helps find problems before the user does by monitoring important “day-to-day” actions that regular users are performing with the synthetic transaction monitors.
- Establishes a baseline performance level with reports and metric data collection.
- Provides advanced diagnostics tools—test scripts, end-user measurements, and so on.
- Provides easy to manage and define service level agreements (SLAs) based on predefined critical transactions.

BPM provides default key performance indicators (KPIs) and configures important actions in a transaction, which actions can be taken from the service level agreements, thus enabling measurement on a specific SLA use case.

- Enables users to trigger an availability event due to the predefined measurements and control monitoring environments—such as controlling locations, BPM, scripts—so that every alert has a critical priority.

It is running on the HP Software VuGen scripting technology that allows for collaboration between the development and operations groups based on this scripting technology (the same scripts and scripting tools used in HP Performance Center and UFT tools).

- **HP Real User Monitor (RUM)**

- Based on the fact that it is a passive monitor that does not affect the application being monitored, RUM can be used to monitor sensitive applications where active monitoring would be prohibited due to regulatory restrictions.
- Using RUM's ability to monitor real user data using port mirroring enables the IT operator to see the performance of each application function as the end user does—not just within your data center or selected locations.

- Does not require scripting, which allows for reduced need in resources for deployment and maintenance.
- Ensures full visibility of application usage and performance, not just what was scripted. Unlike BPM, RUM monitors each and every action performed by the real users.
- Understands the real geographic distribution of your users and the impact of that distribution on end-user experience by monitoring real user network traffic at each geographical location that has users.
- Tracks performance of your most important users, which is particularly useful in trading environments.
- Enables user analytics—understanding exactly how your application is being used.

There are more complex monitors for composite applications and transactional-based monitors (HP Diagnostics/TransactionVision) that we expect to reference in the next version of this Best Practice document.

Installation and Configuration

To install and configure the logical layer monitoring:

- Install the latest version of BSM software. For details, see the latest *HP BSM Installation Guide* on the [HP Software Product Manuals](#) web site.
- While installing BSM, confirm that the EUM model is selected and there is a license to run it.
- Plan your deployment of the data collectors according to the "[Recommendations](#)" on the next page.
- Install the relevant parts for BPM and RUM (probes and engines) and connect them to BSM. For details, see the appropriate *Business Process Monitor Deployment Guide* or *Real User Monitor Installation and Upgrade Guide* on the [HP Software Product Manuals](#) web site.
- (Highly recommended) When approaching an EUM monitoring project (or any monitoring project), create and document the monitored applications together with the application owners.

The following attributes should be defined for each application to be monitored:

- Application structure (according to the illustrated structure described in "[Configuration Items](#)" on page 11)
- Business processes and steps to be monitored
- Thresholds for those processes and steps
- Alerts

- Contacts (including clear owners for the maintenance of the overall application structure and owners for the scripting)
- Iteration times (scheduled run times of the monitors)
- Consistent down times (for example, an application that is only active from 08:00-17:00)
- Key performance indicator (KPI) configurations. For details, see the latest *HP BSM User Guide* or *BSM Application Administration Guide* "Service Health" section on the [HP Software Product Manuals](#) web site.
- Application information for the monitoring scripts (URLs, user names, business process steps, and other application information)

Note: Obtaining this information early saves a lot of time when creating the monitors in the implementation phase where all of the data is already documented.

App Name	Monitor	Thresholds	Alerts	Contacts	Iteration times	downtimes	KPI config	App info
MS exchange	Login to OWA	OK-5 sec,minor-8 sec,critical-15 sec,outlier-60 sec	Send email to application manager if the status is critical	John doe	Run every 5min	Every Friday at 15:00- 16:00 - do not monitor	Performance - worst status, availability - worst status	http://owa.com
App B	x	x	x	x	x	x	x	x

Recommendations

The recommendations for implementing end-to-end monitoring on the Business layer are described in this section. These are recommended best practices, but they are not mandatory.

This section includes:

Monitoring	16
Deployment	17
Configuration	18

Monitoring

In most cases, it is recommended to use both BPM and RUM to monitor an application since their capabilities complement each other and provide a better perspective on application availability and performance.

- Using the deployment documentation, create the monitoring scripts and configurations.
- Apply the monitors to the correct data collectors and start them according to the run times that have already been decided.
- Create alerts for the EUM monitors using the EUM section in the BSM User Guide documentation.

- Using the BSM Platform Administration documentation, configure and apply the consistent downtimes, if needed, to make sure the application is not monitored during the hours it is inactive.
- RUM monitors real user traffic and has many reports and added values provided from the user and network perspective, such as seeing and understanding the real user experience and understanding the geographical distribution and network effects on application availability and performance.
- BPM is a synthetic monitor that provides real-time data regarding the availability and performance of the application from a more neutral perspective. This includes a number of users and actions that really represent the status of important features within the application, as well as providing many reports that help troubleshoot problems and allow 24 hours a day/7 days a week monitoring.
 - Confirm that each important business step of the monitoring script is in a transaction.

For example, a script to monitor send/receive emails can contain three transactions—log on to Exchange server, send an email, receive an email. By checking the transactions, the status of each one of the actions you want to monitor is understood more clearly.
 - Confirm that the user used for monitoring has **password never expires** privileges to avoid false alarms due to user password expiration.
- Confirm all of the applications involved in the process (Load Runner, servers, operating systems, QTP) are in the supported versions. For details, see the relevant Support Matrix.
- Create a solid process for the maintenance of the BPM monitor's script since applications tend to change functionality over time and, when that happens, the scripts might not work as they did initially. Assign clear owners to this process with each business owner or application owner.

Deployment

- When running on more than one data center, the deployment of the BPM/RUM probes must be located within the data center it is monitoring. It is recommended for Enterprise customers to locate the probes in the geographic locations of the data centers. This allows more accurate user experience monitoring and allows tracking for how network issues (or lack thereof) affect the application. Plus the reports from the information help improve the application by identifying specific problematic locations and the root-cause that is causing the problems.

If you also own the HP Network Node Manager i (NNMi) tool for network monitoring, make sure you integrate RUM and NNMi (in BSM) in order to help the root cause analysis of network issues that influence the End-User Experience. For details on how to configure this integration, see the [NNMi—BSM Integration Guide version 9.22](#).

- When running EUM monitors, it is recommended to place the data collectors in places where there are a lot of users to reflect their experience of the application in the best way possible, including the dependency on networking and infrastructure. For example, you can place BPM and RUM data collectors on a specific location where there are a lot of mail users so you can

monitor the service and problem effects on a lot of its customers.

BPM monitors have the capability of monitoring many kinds of applicative protocols. Confirm you use the variety that you need.

RUM monitors have the ability to monitor generic TCP protocol traffic and report on the connections between many kinds of clients and servers.

Configuration

Alerts:

- To reduce the noise and alert storms, it is recommended in most cases to send only one alert for every triggered occurrence or no more than one alert in a set time frame, rather than to set the alert to occur for every trigger occurrence for an already known error (for instance, an alert every five minutes). It is recommended to use this option only if the monitored application is very critical and needs to be reporting errors as they happen or in a situation where the monitor has a lot of time between the run cycles.

For example, when observing BPM monitoring of sending emails from the customer's central office, and the monitor is set to run every minute, then set the alert to be **send no more than one alert every 30 minutes for each occurrence**. We would like to receive alert updates if the problem still exists after 30 minutes, but there is no need to receive an alert every minute because the person who handles the problem already knows about it.

- BSM can be configured to calculate baselines from actual performance metrics. Creating a baseline shows you the normal performance of your applications. Knowing how your site typically performs enables you to determine whether a performance problem is an isolated incident or the sign of a significant downward performance trend.

When baselining is enabled, the average response time and standard deviation are calculated for each transaction and for each location from which it runs, based on real performance data obtained from data samples. The average transaction response time and standard deviation are periodically updated with data from new samples that are received.

When configuring an application's transaction thresholds based on a baseline, specify the number of standard deviations from the average transaction response time that determine the transaction's status. We recommend using the default settings, which are three standard deviations for an **OK** status and four standard deviations for a **Critical** status.

Chapter 3: Software Layer Monitoring

This chapter includes:

Overview	19
Tools	19
Installation and Configuration	20
Recommendations	22

Overview

End-to-end service monitoring includes monitoring the applications' software components that are running on the servers themselves.

Monitoring the components of applications on the servers involves monitoring the application services, processes, log files, and additional objects on the server that are related to the application software. Application Monitoring is the first level of drilldown into components that help the root cause identification, and in some cases resolution.

For example, we can monitor the mailbox application log files and search for certain errors in the logs. We can also monitor the database on the mailbox servers, and the IIS service on the Client Access Servers that is running the Outlook Web Access application.

This and many more software components of the Microsoft Exchange application are important for the operational status of the application. Monitoring these components provides the ability to identify errors that have already occurred or receive an early warning for upcoming errors that impacts on users. For example, when we see warnings in the log file indicating that the mail queue is increasing, this could mean a problem sending emails is about to occur. Also, we could use the monitors to find the root-cause of an existing problem; for instance, the inability to connect to the mailboxes because the related database is down.

Tools

- **HP Operations Manager (HPOM)** is a tool that monitors infrastructure components. It has the ability to monitor via a monitoring agent as well as receiving messages from agent-less components. For software monitoring, it is recommended to use the HPOM out-of-the-box Smart Plug-in monitoring packages. These are monitoring scripts, conditions, and rules that are grouped together to offer better monitoring capabilities for specific software packages. For more information, see [HP Operations Smart Plug-ins](#).

For example, you can deploy the Smart Plug-in (SPI) for Microsoft Exchange in your environment. This SPI contains a set of monitoring out-of-the-box packages for monitoring Exchange servers. For more information about SPIs for Exchange, see the SPI for the *Microsoft Exchange Installation and Configuration Guide* on the [HP Software Product Manuals](#) web site.

You can also use HPOM to monitor other application components with custom-made monitors—for example, a custom-made log file monitor for errors in the mailbox server's applicative log file.

- **HP Diagnostics** software monitors application transaction health in traditional, virtualized, and cloud environments—allowing quick isolation and resolution of issues. It supplies a common tool to easily collaborate across the entire application life cycle and releases applications of higher quality. HP Diagnostics provides deep drill down into transactions from the end user through the back end. It uniquely serves as a common tool set for preproduction and production to diagnose application performance bottlenecks quickly, with improved quality.
- **HP SiteScope (SiS)** is an agent-less monitoring solution designed to ensure the availability and performance of distributed IT infrastructures. You can manage the monitors and distribute them as templates on the remote servers with predefined credentials.

In addition, there are specific monitoring capabilities which enable the ability to design and deploy custom monitors for JDBC connections, log file analysis, SNMP monitors, and so on.

For example, you can use SiteScope to monitor an applicative mailbox service on the mailbox servers, or use the **send and receive monitor** to monitor the ability of the Exchange to send mail and receive it from a third-party mail application.

There is also an Exchange monitoring package in SiteScope.

For more information, see the SiteScope documentation on the [HP Software Product Manuals](#) web site.

Installation and Configuration

To install and configure the Software layer when planning deployment and implementation of a monitoring project:

1. Install the latest version of HP Operations Manager (HPOM) software.
2. Install the latest version of HP SiteScope software.
3. Install the latest version of HP Diagnostics software.
4. Map the customer IT environment—all of the servers and applications that need to be monitored. An automated discovery tool like Universal Discovery eases this task.

Include the following in the mapping documents:

- Application names and IPs
- Application manager and other related contacts

- Application servers, such as:
 - server1: mailbox server
 - server2: hub server
 - and so on
 - Software installed on each of the servers, such as:
 - server1: MSSQL, Exchange Mailbox, IIS
 - and so on
 - Servers operating systems (for future needs), such as:
 - server1: Microsoft Windows 2008R2
 - server2: Linux
 - and so on
 - Software monitoring method for each server, such as:
 - server1-MSSQL: HPOM SPI for database
 - server1-IIS: using SiS IIS monitor
 - and so on
 - Any necessary special credentials—log file paths, application paths, such as:
 - server1: log on to database with user **SA** and password **XYZ**
 - and so on
 - Server participation in a disaster recovery/load balancer (DR/LB) deployment
- It is important to know if a server is active or passive in order to monitor it correctly. For more information, see the monitoring guide for the selected product and monitor on the [HP Software Product Manuals](#) web site.
- Downtime rules; for instance, if there are scheduled times that the application is inactive

5. After mapping, decide which tools to use to monitor each server/software.

Note: A server can be monitored both with SiteScope and HPOM, based on the monitoring needs and product capabilities.

6. Deploy the HPOM monitoring agents according to your configuration and mapping.
7. Configure all of the remote servers in SiteScope.

Note: For more information, see the Using SiteScope Guide on the [HP Software Product Manuals](#) web site.

8. Prepare the HPOM policies and SiteScope templates for a large scale deployment.
9. Deploy all monitors.
10. Install Diagnostics' agents and collectors.
11. After installing Diagnostics' agents and collectors, customize the instrumentation and control the data collection settings using a number of different configuration files located in the agents' installation. For information regarding the types of configuration files, see the [Diagnostics version 9.20 Installation and Configuration Guide](#), especially the **Custom Instrumentation for Monitoring Java, .NET Applications, Advanced Configuration of the Diagnostics Server**, and **.NET Agents** sections.

Recommendations

The recommendations for implementing end-to-end monitoring on the Software layer are described in this section. These are recommended best practices, but they are not mandatory.

This section includes:

Monitoring	22
Deployment	23
Severity Mapping	23

Monitoring

According to the user's needs, there is a choice between using monitoring application components via HPOM SPIs or SiteScope solution templates—or both. Both products monitor a variety of software components via predefined monitors. The main consideration is whether there is a monitoring solution for the specific application that it is suitable for you.

In some cases, the HPOM agent-based monitoring solution is preferable due to its queuing capabilities. An HPOM agent gathers and queues data at all times as long as the OM agent is up and running.

Note: It is common for customers to use HPOM for monitoring the critical software components since HPOM provides an in-depth view of these components, and to use SiteScope for monitoring in specific situations that usually include the need for a fast monitoring solution, an agentless monitoring solution, or a specific monitor that HPOM does

not have or is taking a very long time to implement via HPOM.

Caution: Smart Plug-ins for application monitoring are complex monitor sets with the capability to drill down to specific application functionality. In order to achieve maximum efficiency in error detection, and reduce the possible noise, it is recommended to review the appropriate SPI documentation completely.

Deployment

It is recommended to save the HPOM configuration via the **Download Configuration** option for backup and DRP purposes. If you need to backup or use another server as a disaster recovery (DR) server, it is recommended to have your configuration with all of the monitors and other configurations you have created in a backed up file.

HPOM can be configured as a MOM (Manager of Managers) to centralize message redirection into one specific GUI, thus allowing more efficient monitoring and management in your HPOM monitoring environment. By allowing all of the operation managers in the environment to forward messages to one HPOM server and allowing the managers to "talk" to each other, a monitoring environment that is configured on multiple HPOM servers from one MOM server can be managed. This is recommended in large scale environments that are managing a large number of managed nodes in HPOM or in environments that have multiple data centers with HPOM servers that need to be managed from one location.

Severity Mapping

It is very important to define the correct severity for the monitors you create.

Usually the severity of the monitoring events determines which of the events in the NOC console are addressed first. A situation where too many of the events are critical prevents the operator from assessing and handling the events in the correct order. In addition, a situation where a critical event is not addressed in a timely manner because it has the wrong severity can negatively impact the business. Those situations should be avoided.

In our email service example, the events regarding the status of the mailbox process should be defined with high severity—after taking into consideration the deployment type and the probability of a negative impact that could occur if this event is not handled. Each use case must be analyzed in order to assign the correct severity to the message.

Chapter 4: Infrastructure Layer Monitoring

This chapter includes:

Overview	24
Tools	24
Installation and Configuration	25
Recommendations	26
Virtual Infrastructure Monitoring	28
Network Monitoring	37

Overview

The Infrastructure layer (servers and hardware) that is used by the Software layer (server, network, and other infrastructure services) must also be monitored.

The applications described in Chapter 3, ["Software Layer Monitoring" on page 19](#) consume resources while operating; that is, operating system resources such as CPU cycles and memory, storage resources, network resources, and so on.

To ensure that the applications have an adequate environment to perform their function, we must monitor the consumption of those resources.

In the memory example, if the server has a memory overload, all of the applications installed on the server, including the Microsoft Exchange application, stop working. This is also the case in the event of lack of disk space that, for example, is needed for the mailbox database.

So it is very important that the servers and infrastructure services work as agreed in the service level agreement (SLA) in order for the application to run properly.

Tools

- **HP Operations Manager (HPOM)** is a tool that also monitors infrastructure components. It has the ability to monitor via a monitoring agent and also receives messages from agent-less components (SiteScope).

For server and operating system monitoring, HPOM has several Smart Plug-ins (SPIs) that monitor the operating system and hardware.

For example, the HPOM operating system SPI monitors the Exchange Client Access server's CPU.

If the CPU reaches 90%, users access to their mailboxes via the Outlook Web Access service may be delayed. There may be a performance problem on the Outlook Web Access log on transactions in EUM monitoring, but the root cause of the problem is identified more efficiently with the operating system SPI monitors.

- **HP SiteScope (SiS)** is an agent-less monitoring solution designed to ensure the availability and performance of distributed IT infrastructures. You can manage the monitors and distribute them as templates on the remote servers with predefined credentials.

For example, memory performance can be monitored on the Microsoft Exchange Mailbox servers remotely from the SiteScope machine in order to tell if the server is having memory overloads—a scenario that slows server performance and all of the applications that are running on it; including causing server overload, with users unable to connect to their mailboxes.

- **HP Network Node Management i (NNMi)** monitors and manages the network services. For more information on network monitoring, see ["Network Monitoring" on page 37](#).
- **HP Storage Essentials (SE)** is a comprehensive storage resource management solution. For more information, see [Storage Essentials](#) on HP Live Network.

Installation and Configuration

To install and configure the Infrastructure layer:

- Install the latest version of HP Operations Manager software. It is recommended to install HPOM on a UNIX/Linux server for a large scale environment.
- Install the latest version of HP SiteScope software, and plan your deployment taking into consideration that you have enough licenses for all of the monitors you need.
- Use the mapping that you have created in the Software layer to decide how to monitor the server operating system. It is recommended to use the operating system SPI according to the server operating system's version.
- Deploy the HPOM monitoring agents in the supported versions for each of the supported operating systems. For more information, see the *HPOM Support Matrix Interface* that came with the product.
- Configure the remote servers in SiteScope.
- Prepare the HPOM policies and SiteScope templates for a large scale deployment, meaning try to organize the HPOM policy group that fits the customer's needs.

For example, if you know that the customer has two types of operating systems on the servers, try to separate the servers into two **node groups** and the monitors into two **policy groups**. That way it is easier to deploy.

Also, use groups on SiteScope according to the customer's needs. For example, if you know that the customer only uses SiteScope to monitor CPU and memory on all of the servers, perhaps it is better to create two groups of monitors and to add each monitor to its group with the **server name** title.

For more information about configuring HPOM and SiteScope, see the relevant product user guides on the [HP Software Product Manuals](#) web site.

- Deploy all monitors.

Recommendations

The recommendations for implementing end-to-end monitoring on the Infrastructure layer are described in this section. These are recommended best practices, but they are not mandatory.

This section includes:

Monitoring	26
Deployment	26
Severity Mapping	27

Monitoring

According to the user's needs, there is a choice between using HPOM SPIs or SiteScope solution templates—or both—to monitor infrastructure components. Both products monitor a variety of infrastructure components via predefined monitors. The main consideration is whether there is a monitoring solution for the specific use case that it is suitable for you.

In some cases, the HPOM agent-based monitoring solution is preferable due to its queuing capabilities. An HPOM agent gathers and queues data at all times as long as the HPOM agent is up and running.

Note: It is common for customers to use HPOM for monitoring most of the infrastructure components in a large scale environment, and to use SiteScope for monitoring in specific situations that usually include the need for a fast monitoring solution, an agentless monitoring solution, or a specific monitor that HPOM does not have.

Caution: Smart Plug-ins for infrastructure monitoring are complex monitor sets with the capability to drill down to specific infrastructure functionality. In order to achieve maximum efficiency in error detection, and reduce the possible noise, it is recommended to review the appropriate SPI documentation completely.

Deployment

It is recommended to save the HPOM configuration via the **Download Configuration** option for backup and DRP purposes. If you need to backup or use another server as a disaster recovery (DR) server, it is recommended to have your configuration with all of the monitors and other configurations you have created in a backed up file.

HPOM can be configured as a MOM (Manager of Managers) to centralize message redirection into one specific GUI, thus allowing more efficient monitoring and management in your HPOM monitoring environment. By allowing all of the operation managers in the environment to forward messages to one HPOM server and allowing the managers to "talk" to each other, a monitoring environment that is configured on multiple HPOM servers from one MOM server can be managed. This is recommended in large scale environments that are managing a large number of managed nodes in HPOM or in environments that have multiple data centers with HPOM servers that need to be managed from one location.

Severity Mapping

It is very important to define the correct severity for the monitors you create.

Usually the severity of the monitoring events determines which of the events in the NOC console are addressed first. A situation where too many of the events are critical prevents the operator from assessing and handling the events in the correct order. In addition, a situation where a critical event is not addressed in a timely manner because it has the wrong severity can negatively impact the business. Those situations should be avoided.

In our email service example, the events regarding the status of the mailbox process should be defined with high severity—after taking into consideration the deployment type and the probability of a negative impact that could occur if this event is not handled. Each use case must be analyzed in order to assign the correct severity to the message.

Virtual Infrastructure Monitoring

This section includes:

Overview	28
Tools	28
Installation and Configuration	29
Recommendations	29

Overview

There are several methods to implement service monitoring on top of the virtual infrastructure. Virtualization enables dividing the computer resources into multiple execution environments. It abstracts the physical hardware layer to enhance IT resource utilization. Virtual machines are used to consolidate the workloads of several under-utilized servers to fewer machines for more effective usage of hardware. It helps to reduce costs and aids in the maintenance and administration of the infrastructure.

Virtual machines are also used to run multiple operating systems simultaneously. These operating systems can be different versions, or even entirely different systems, which can be on hot standby. Virtualization enables existing operating systems to run on shared memory multiprocessors. Since virtual machines are logical entities and are separate from the physical resources they use, the host environment is able to dynamically distribute the resources between them.

For more information, see Chapter 6, Event Management, "[Overview](#)" on page 44.

Tools

The following monitoring tools could be used to monitor virtual environments in the following ways:

- **HP SiteScope (SiS)** provides virtual infrastructure monitors and counters. For monitoring a VMware Host, SiteScope provides a solution template—a set of monitors and counters that are predefined and are HP's best practice to monitor such a host. The VMware Host Solution Template enables monitoring of virtual infrastructure such as: CPU, memory, network, state, and storage-related counters of the VMware Host Server and of guest virtual machines that reside on the host server. The VMware Host Solution Template allows you to monitor ESX hosts in two ways—via vCenter software or via the ESX host directly.

Note: The SiteScope product contains the Solution Template, for which a license is required. Instructions on how to use the SiteScope Solution Template can be found in "[Using SiteScope](#)".

- **HP Operations Manager (HPOM)** is designed to monitor infrastructure components. It is recommended to use the HPOM out-of-the-box smart plug-in monitoring packages called Virtualization Infrastructure Smart Plug-in (VI SPI) that is a part of the **Infrastructure SPI** package. This SPI contains monitoring scripts, conditions, and rules that are grouped together to offer better monitoring capabilities for virtual infrastructure (and includes events, topology, and

report metrics).

This SPI also enables relating the cross domain IT infrastructure events to relevant applications and maps them into a hierarchical service map. The map view displays the real-time status of your infrastructure environment and helps to identify the root cause of alarms reported on operating systems, associated software services and, in addition, essential hardware elements such as CPU, memory, swap space, and so on. For more information, see the [HP Operations Smart Plug-ins for Infrastructure Installation Guide](#).

Installation and Configuration

To install and configure the virtual monitoring service, see Chapter 4, Infrastructure Layer Monitoring, "[Installation and Configuration](#)" on page 25.

For additional information on Virtual Monitoring, see the [HP Operations Smart Plug-ins for Infrastructure Installation Guide](#).

Recommendations

There are two recommendations for implementing end-to-end monitoring on the virtual infrastructure layer described in this section.

This section includes:

Using HP SiteScope Solution Template	29
Using HP Operations Manager Virtual Infrastructure Smart Plug-in	30

When deciding on which type of monitoring solution to implement for your virtual infrastructure, there are several things to take into account.

Our generic recommendation is to use the HP Operations Manager Virtual Infrastructure Smart Plug-in. However, there are pros and cons for each product. For more information, see Chapter 4, Infrastructure Layer Monitoring, "[Recommendations](#)" on page 26.

Using HP SiteScope Solution Template

The VMware Host Solution Template allows you to monitor ESX hosts in two ways—via vCenter or via the ESX host directly. We recommend that you monitor ESX hosts directly to reduce the load on the vCenter machine.

We recommend deploying the solution template using a CSV file, since you can perform multiple deployments at one time without having to manually enter variable values for each deployment in the user interface. For details on deploying a CSV file, see "How to Deploy Template Using a CSV File" on page 878 in [Using SiteScope](#).

For more information about deployment of the solution template. see "How to Deploy a SiteScope Solution Template" on page 930 in [Using SiteScope](#).

Using HP Operations Manager Virtual Infrastructure Smart Plug-in

These are monitoring scripts, conditions, and rules that are grouped together to offer better monitoring capabilities for specific software packages. For more information, see [HP Operations Smart Plug-ins](#).

For example, you can deploy the Smart Plug-in (SPI) for Virtual Infrastructure in your environment. This SPI contains a set of monitoring out-of-the-box packages for monitoring virtual servers, thus confirming every object in your virtual environment is properly monitored.

There are several vendors that provide virtual infrastructure service—each with their own type of deployment. HP Smart Plug-in for Virtualization Infrastructure (VI SPI) version 11.13 has different deployment types for the following supported vendors:

VMware	31
Microsoft Hyper-V Server	33
IBM LPAR/WPAR	34
HPVM	35
Oracle Solaris (Container)	36

Policies of the VI SPI

HP Operations Manager's policy is a set of conditions and rules that create a monitoring package. SPI is a set of predefined policies to monitor specific IT services. The policy groups that the VI SPI offers are:

- **Collector policies.** This policy collects VM information and logs the data into the CODA.

For example: VI-IBMHMCDataCollector

Note: The above policy is deprecated from version 11.13.

- **Availability policies.** This policy monitors the VM state, and process or service availability.

For example: VI-HPVMStateMonitor

- **Performance policies.** Performance monitoring helps to preempt performance disruption.

For example: VI-HPVMHostCPUUtilMonitor

- **Log / Event Monitor policies.** This policy forwards all warning and error event log entries.

For example: VI-VMwareEvent Monitor and VI-MSHyperV_VMMSAdminWarnError

For more information about VI SPI use, see the [“HP Operations Smart Plug-in for Virtualization Infrastructure User Guide Version 11.13.”](#)

VMware

VMware provides ESX/ESXi servers managed by vMA or vCenter via a virtual appliance (VA).

The VI SPI on VMware allows you to monitor your virtual infrastructure in two ways:

- via vMA (policies on the VMware ESX)
- via VA (policies on the virtual appliance that monitor the vCenter)

We recommend using a virtual appliance for the following reasons:

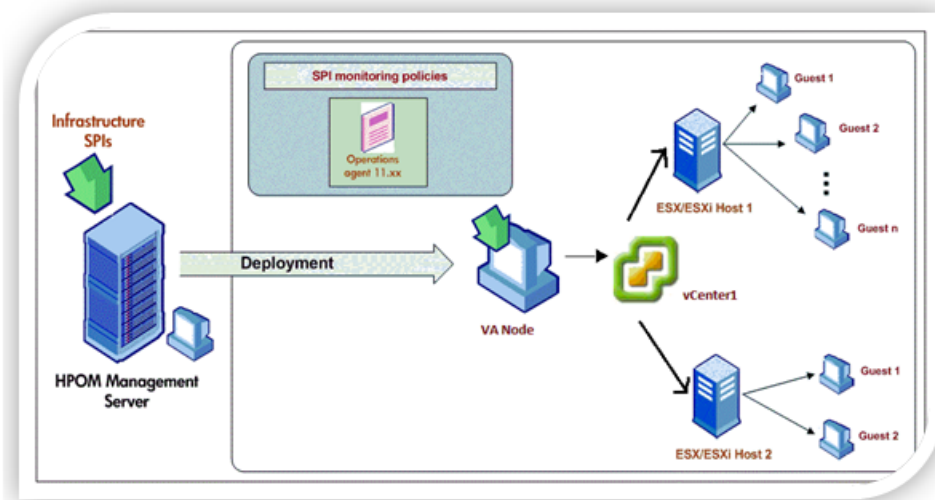
	vMA	VA
Ease of deployment and configuration	<ul style="list-style-type: none"> • No remote deployment • Configure every host to be monitored 	<ul style="list-style-type: none"> • VMware standard remote deployment • Configure vCenters to be monitored
Feature set	<ul style="list-style-type: none"> • 90 virtualization metrics • Metrics focused on server (Hosts, VMs, RPs) 	<ul style="list-style-type: none"> • 200+ virtualization metrics • Metrics across the vSphere environment (Datacenter, Clusters, Resource Pool, Host and VM, Storage, vApp) • Perl API to access collected from OA data store • Ability to monitor multiple vCenters as well as part of a vCenter from the VA • Multiple collection levels to collect subset of entity types

	vMA	VA
Scale	<ul style="list-style-type: none"> • One vMA for every 400 target managed instances • Legacy data store not designed for scale 	<ul style="list-style-type: none"> • One VA for 2000+ target managed instances – >5X scale • Scalable and robust open source RDBMS store • New access APIs provide ability to filter data at source • Access APIs allow dynamic summarization on the fly
Security	<ul style="list-style-type: none"> • Vi-fastpass (VIFP)-based security credential store allowing connection to ESX(i) or vCenter servers 	<ul style="list-style-type: none"> • Stronger and configurable encryption (default 128-bit AES) for credential store
SPI Support	<ul style="list-style-type: none"> • VI SPI for vMA-based solution 	<ul style="list-style-type: none"> • VI SPI for vCenter-based solution

For more information about VI SPI, see the [HP Operations Smart Plug-ins for Infrastructure Installation Guide](#).

The deployment of the VA includes:

- Registering the vCenter (VC) to be monitored in the VA node
- Deploying VI SPI alone on the VA—discovering Hosts, Guests, Resource Pools (RPs), Clusters, Data Centers (DCs), VCs, and DataStores
- Monitoring VA availability and performance



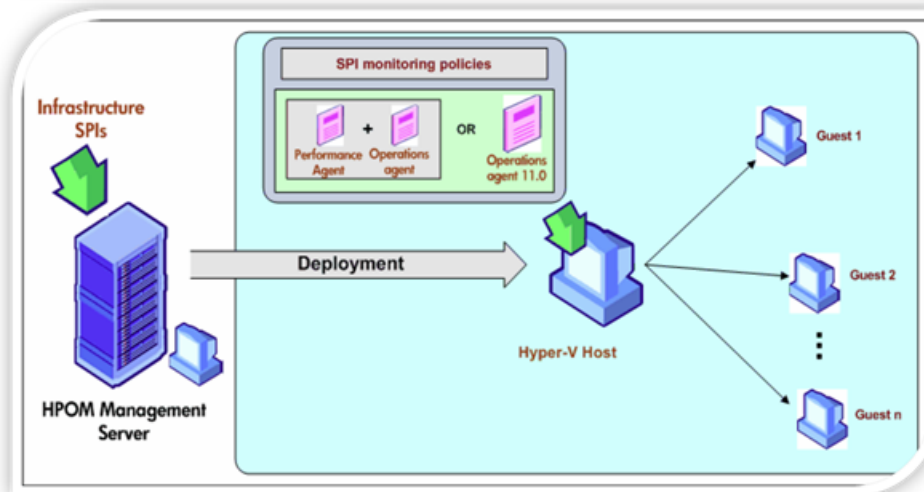
In our use case, while monitoring the email business service, all of the infrastructure of the business services is deployed on virtual infrastructure. Therefore, there are no changes in how we are monitoring our application and software layer, but there are some changes in the infrastructure.

For example, the email mailbox servers are deployed on VMware virtual servers. After deploying the VA solution in this environment, we are able to monitor all of the mailbox servers using the VI SPI. If one of the mailbox servers has a memory leak, the monitor sends an event to OMi.

Since we are dealing with a virtual server, there are several ways to resolve the memory leak. For example, if needed, the operation team can add more memory resources to the server from the VC or create an automated action. They can also choose to handle the event as a physical server memory issue and try to find the root cause. Either way, the advantages of using virtual infrastructure are available for usage via the VI SPI package.

Microsoft Hyper-V Server

Hyper-V Monitoring
Deploy VI SPI on Hyper-V Host.
Discovers host and guest machines.
Monitor the availability and performance of Hyper-V hosts and the guest machines



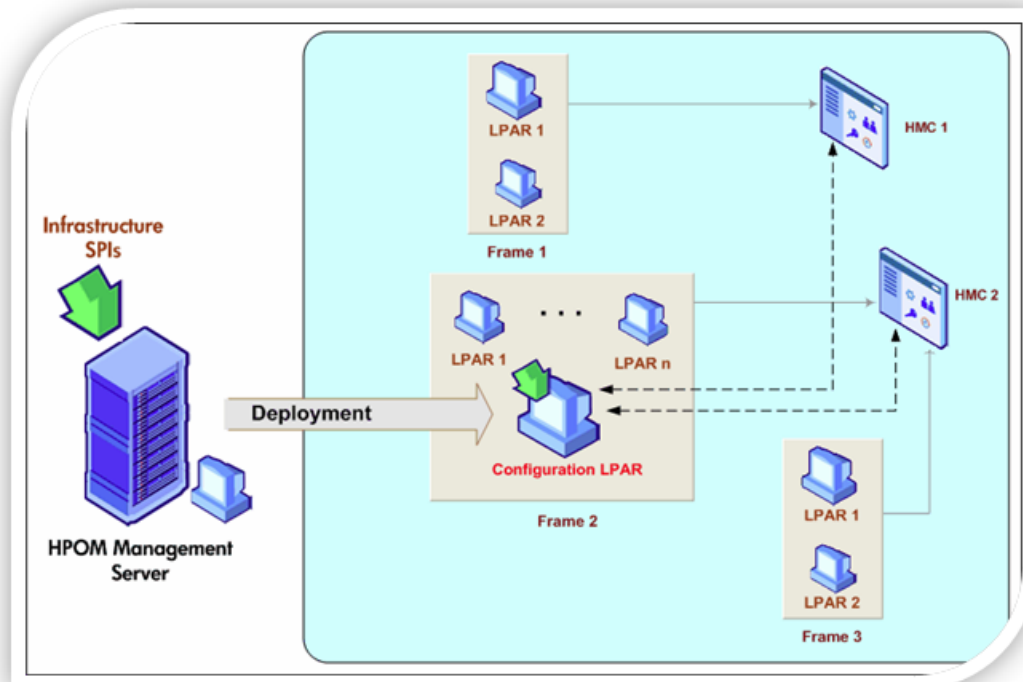
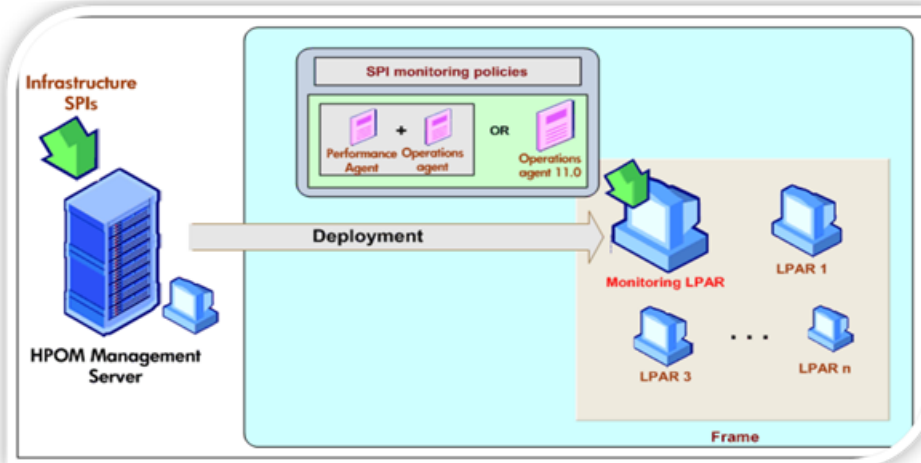
IBM LPAR/WPAR

Monitoring the LPARs, Frame, and WPARs

VI SPI, for IBM AIX LPARs, is deployed on an LPAR within a frame.

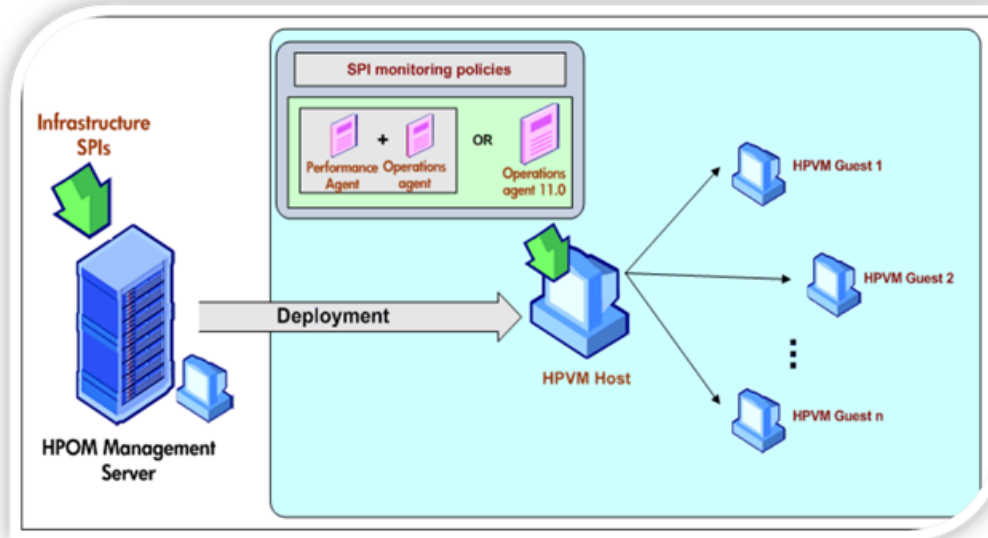
Discovers other LPARs within Frame, WPAR on the Monitoring LPAR and HMC.

Monitors Frame, all LPARs within Frame and WPAR on the Monitoring LPAR only.



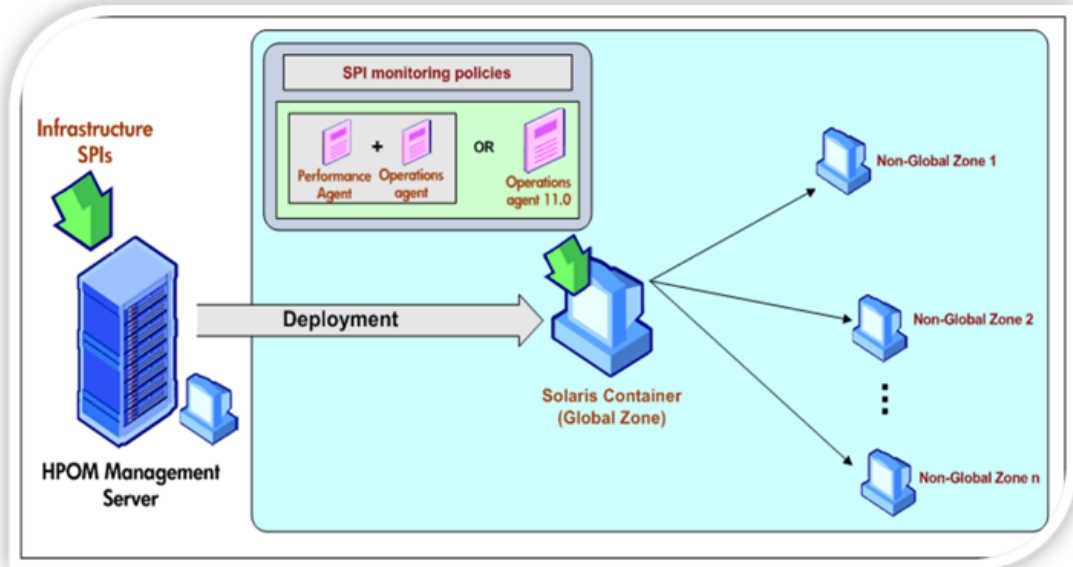
HPVM

HPVM Monitoring
Deploy VI SPI on HPVM Host.
Discovers host and guest machines.
Monitor the availability and performance of HPVM hosts and the guest machines



Oracle Solaris (Container)

Solaris Container
Deploy VI SPI, on the Solaris container which is also the global zone
Discovers global and non global zones running on the global zone.
Monitors the availability and performance



Network Monitoring

This section includes:

Overview	37
Tools	37
Installation and Configuration	38
Recommendations	39

Overview

Monitoring the network is a very important infrastructure IT service that enables the investigation of problems from every angle. The network monitoring section is separated from the Infrastructure layer monitoring step due to the complexity of network monitoring.

We need to monitor the availability and performance of our network devices so we can identify a root cause to large scale problems and solve them, instead of wasting time and resources solving problems which are the symptoms to the root cause network problem.

For example, a network switch that is connected to all of the Microsoft Exchange Environment Mailbox Servers is down. Without monitoring the network, it would appear that there is something wrong with the application since the End-User Management monitors show that the **Microsoft Exchange Send/Receive Mail** monitor is at a critical status and that the transaction flow is unavailable. When adding the network events to this scenario, we can easily fix the root cause of the problem causing the Exchange application, which is the faulted network switch, thus decreasing the application downtime.

As mentioned earlier, in order to perform their functions, applications depend on multiple resources and services. Networking services are an example of a critical component that a high percentage of applications depend on, hence the need to monitor network availability and performance very closely. Network monitoring and administration requires a specific set of tools and capabilities.

Tools

- **HP Network Node Management i (NNMi)** provides powerful capabilities to enable your network operations team to efficiently manage a network of any size, reduce the business risk of downtime, and increase network service levels. NNMi is the one solution for managing fault, availability, performance, and advanced network services for your physical, virtualized, hybrid, and cloud network environments. NNMi is one component of the HP Automated Network Management Suite that provides a holistic, automated approach across the network management domain of fault, availability, performance, and change, configuration, compliance and process automation.

For example, when monitoring the network switch that contains all of the Microsoft Exchange Client Access Servers, if the switch is down, there are a lot of unavailable Exchange servers and the whole email service is affected.

Also, by using Smart correlation rules, NNMi knows that the problem is in the switch and creates one cause event where all of the unavailable server's events are its symptoms.

- **HP Network Node Management Smart Plug-ins (NNM iSPIs)** for performance and advanced network services extend the device and protocol support of NNMi and the Automated Network Management solution to enable management of a wide range of network devices, services, and facilities. These Smart Plug-ins provide technology-specific awareness that can identify a problem more quickly and reduce mean time to repair (MTTR), making your network team more efficient.

Installation and Configuration

To install and configure the infrastructure service:

1. Install the latest NNMi software and iSPI for Performance executable files.
2. Install NNMi using the interactive installation guide at [NNMi version 9.20 Interactive Installation Guide \(Windows and UNIX\)](#).
3. Confirm that NNMi can access the monitored nodes.

Note: Access Control Lists (ACLs) can prevent NNMi from communicating with the nodes. Firewalls can also prevent communication between NNMi and the monitored nodes. It is recommended to allow all of the above so that the NNMi uses all of its monitoring capabilities.

4. It is highly recommended to configure SNMP communication.

Note: While nodes can be monitored using only ICMP, monitoring is significantly improved when SNMP is available.

5. Discover your nodes either via rule-based auto-discovery or by seeding each node individually. It is recommended to use the auto-discovery rules since it is much easier to define, instead of seeding one node at a time.
6. It is highly recommended to configure network devices to send traps to the NNMi server(s). For more information, see the [NNMi version 9.22 Deployment Reference Guide](#).

Recommendations

The recommendations for implementing end-to-end monitoring for Network Monitoring are described in this section. These are recommended best practices, but they are not mandatory.

This section includes:

Deployment	39
Discovery	39
Monitoring	40
Configuration	40
Documentation	40

Deployment

NNMi has two deployment models:

- a single NNMi station to monitor all nodes

NNMi can scale quite high even with just a single server. The advantage to this model is simplicity. The disadvantage is that all nodes are monitored from a single location and traffic across WAN links may be slower than desired.

- a distributed solution with a **global** NNMi station and multiple **regional** NNMi stations

The regional stations communicate with the global station. All polling is done by the regional stations so the load is well distributed. The advantage to this model is that polling can be done local to a region (like a data center) and may be more friendly to a DMZ or other secure environments. The disadvantage is that there is increased complexity and maintenance. It is recommended to choose whichever architecture meets your needs, taking into consideration the advantages and disadvantages of each deployment model.

Discovery

- Consider including servers in your NNMi discovery if you want better service mapping and correlation rules.
- You can have rules to both include and exclude nodes from auto-discovery. It is usually best to drive these decisions based on System OIDs. Be cautious to not include printers and other gear you do not intend to monitor. Out-of-the-box, NNMi only discovers switches and routers, which is a good starting point.

- If you discover servers, it is highly recommend that you enable SNMP on the servers. Linux servers typically also need to have additional SNMP tree access configured to include interfaces and addresses.

To enable and configure SNMP, see the [NNMi version 9.22 Deployment Reference Guide](#).

Monitoring

It is recommended to use the out-of-the-box monitoring options. These have been well tuned to meet the majority of customer needs. NNMi typically does not monitor access ports on access switches except those connecting monitored switches. This way you do not need to worry about monitoring laptops and desktops that may be disconnected and turned off.

Configuration

It is recommended for most customers to not load any MIBs into NNMi after installation. MIBs can be loaded to help define additional traps—for example, for receiving SNMP traps from a third-party monitoring application—but loading MIBs usually does not alter NNMi monitoring and may be unnecessary.

Documentation

There are a number of excellent white papers that cover example deployments and many useful scenarios available on the [HP Software Product Manuals](#) web site.

There is a very extensive Deployment Reference available. This is installed with the product and is available from the online help menu called NNMi Documentation Library. This reference is typically updated with each patch as well.

Part II: Event Management

Chapter 5: Introduction

This chapter includes:

Overview	42
What is an Event?	43

Overview

Part II: Event Management contains guidelines on how to implement event management, along with best practices to help implement the event management process using the monitors created in Part I: End-to-End Service Monitoring in the IT Environment. For more information, see ["Use Cases" on page 10](#).

ITIL v3 defines Event as "Any detectable or discernible occurrence that has significance for the management of the IT infrastructure or the delivery of IT service and evaluation of the impact a deviation might cause to the services." For more information, see the *ITIL Service Operation Guide*.

The objective of ITIL Event Management is to make sure configuration items (CIs) and services are constantly monitored throughout their life cycle. Event management aims to filter and categorize events in order to decide on appropriate actions if required.

When managing a large scale IT environment, it is essential to implement an effective event management process.

The event management process does the following:

- Detects state changes that can significantly impact CIs and services in the environment
- Provides a means to compare the actual operational status against service-level agreements (SLAs) and predefined standards
- Uses correlation and priority rules on a consolidated event console to increase availability and performance of the IT environment
- Provides information and reporting capabilities for other IT processes and activities, such as incident management, change management, and so on

The event management process is implemented using the HP BSM Operations Management (OMi) module, which is part of BSM's Service and Operations Bridge solution. For more information on the Service and Operations Bridge, see [HP Business Service Management Getting Started with BSM](#), Chapter 2, "Introducing BSM, Service and Operations Bridge" on page 12.

BSM's Service and Operations Bridge solution provides consolidated service and event management through a unified console—the OMi application—enabling you to monitor and manage the events that occur in your IT environment, helping to restore disrupted services as quickly as possible, and minimizing service disruptions.

In order to manage the events properly, you must handle the important events first. You need to correlate the events to help solve the cause for a number of events describing a service disruption, to document the events and understand the event's influence on the business, and so on.

Part II: Event Management continues to use the example of an email service as a common IT service and describes the best practice of implementing an event management process using the HP BSM OMi module in an IT environment.

What is an Event?

An event is a notification created on an IT service or any other CI by a monitoring tool once a change of state occurs that has significance for the IT service's or CI's management. The event can be detected using a variety of monitoring tools, called data collectors, such as:

- HP Operation Manager (HPOM)
- HP Business Service Management – End User Management (BSM-EUM)
- HP Network Node Manager (NNMi)
- ArcSight Logger
- Third-party tools, such as Microsoft SCOM, and so on

For additional examples, see Part I, ["End-to-End Service Monitoring in the IT Environment"](#) on page 8.

Chapter 6: Event Management

This chapter includes:

Overview	44
Event Detection and Notification	49
Event Filtering	50
Overview	50
Configuration	51
Event Correlation	53
Overview	53
Health Indicators	54
Key Performance Indicators	55
Topology-based Event Correlation	56
Stream-based Event Correlation	57
Event Handling	58
Event Closure and Review	61

Overview

Event management is the process responsible for managing events throughout their life cycle—a process that is responsible for handling all of the events in the IT environment in order to restore IT services to normal operation as quickly as possible.

Service modeling is a core prerequisite for event management. It is the process which creates relationships between layers of configuration items (CIs)—the monitored environments, business applications, and business services structure.

Service modeling is the task of connecting a model CI to its surrounding entities—using relationships. Once the model is set up, topology views can be created based on the CI relationships created from within the model.

A correct service modeling process brings a lot of added value to an event management process as it allows better understanding of the impact of events on related CIs and the business services, thus affecting the corresponding business service.

Service modeling also allows visibility of the big picture of an IT environment and how an event on one of the CIs affects the whole organization (Business Impact Analysis). With this ability, we can prioritize events, according to the business needs of the organization, correlate events, and measure business impact and SLA matrices on our business services.

There are several approaches to service modeling. Since service modeling is part of the configuration management process that supports many other processes of IT, it can be done in

either the HP Universal CMDB (UCMDB) product and/or the Run-Time Service Model (RTSM) module of HP Business Service Management (BSM).

When considering an implementation of the service modeling process for service monitoring and event management, our recommendation is to create the model in the UCMDB product (taking into account the service monitoring and event management needs).

After creating the model in UCMDB, connect the Business layer CIs to the discovered infrastructure CIs, then synchronize that model into the BSM RTSM module and enrich the model with the information coming from all of the monitoring data collectors. It is also possible (though generally not recommended) to do this the opposite way (start in the RTSM and synchronize the model to the UCMDB). Whether the modeling is done in UCMDB or in RTSM depends on many parameters.

We recommend that the following principles be observed:

- Consider the main purpose/need for service modeling before deciding on the modeling product.
- If service modeling in the UCMDB, take into consideration the needs of monitoring.
- UCMDB is expected to be the **source of truth**. If service modeling is done in RTSM, it has to be synchronized to the UCMDB.

RTSM was introduced in BSM version 9.00. It is the foundation of many of the BSM applications and services, including Operations Management (OMi) and topology-based event correlation (TBEC). In order to receive the maximum value from OMi and TBEC, be sure to have a good service model representing the CIs in your IT environment and the relationships between them.

Information about the CIs and their relationships in RTSM can be found in the following sources:

- **CMS and Universal Discovery (UD)**. If you have a Configuration Management Database (CMDB) in your IT environment (whether it is HP UCMDB or not), you might want to synchronize the service models from the CMDB into RTSM. These topologies may have been discovered by UD or in earlier versions of DDMa and/or DDMi, or modeled manually by the CMDB team.
- **BSM data collectors and stand-alone products**. BSM data collectors create CIs and relationships within RTSM based on monitored and/or discovered data. For example, when you define a CPU monitor in HP SiteScope, the remote server is a computer CI and SiteScope creates the relevant topology in RTSM. Some additional examples are:
 - HP Real User Monitor (RUM) creating a WebServer CI connected to a computer based on network traffic
 - Network Node Manager i (NNMi) discovering L2 topology and enriching RTSM with it
 - BSM Connector for SCOM reporting topology as well as events
 - HP Operations Manager (HPOM) creating service models based on its data using the TopoSync operation

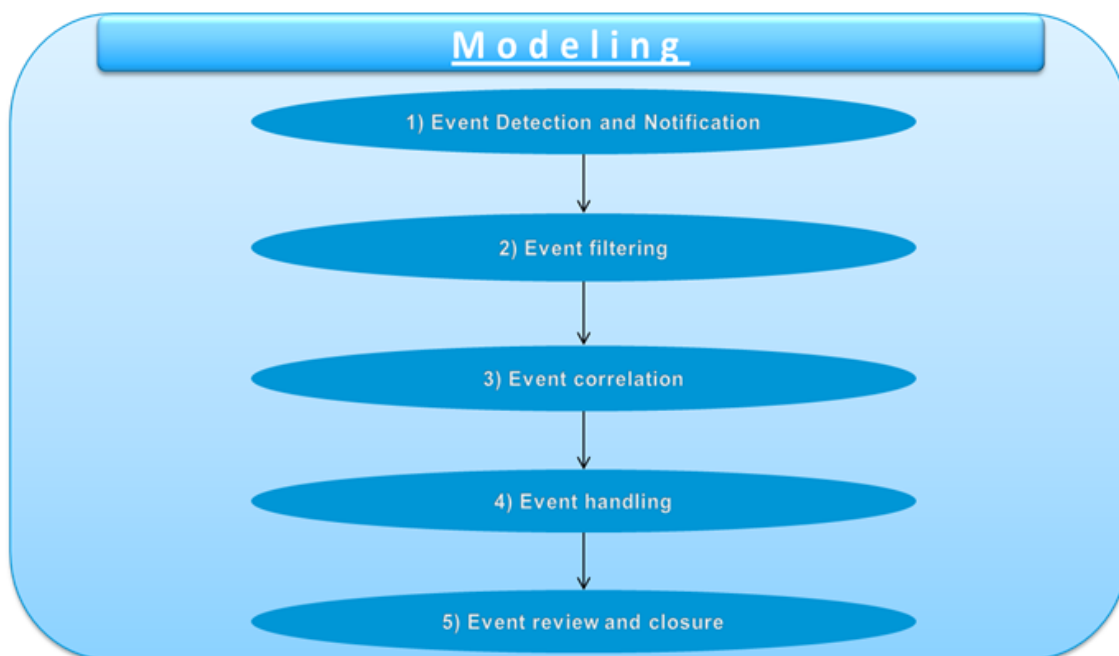
Note: For information on how to create models and Modeling Best Practices, refer to [HP Business Service Management Effective Modeling for BSM—Best Practices](#).

Note: There are several methods to implement event management in the virtual infrastructure. It is recommended to use BSM (OMi) to manage monitoring metrics, events, and topology.

Since the topology in virtual infrastructure is dynamic, the use of topology sync from the monitoring tools to the RTSM (BSM) allows us to keep our topology views up to date. For more information about virtualization, see Chapter 4, Infrastructure Layer Monitoring, Virtual Infrastructure Monitoring "[Overview](#)" on page 28.

RTSM is a dynamic deployment of UCMDB, and is capable of receiving real time topology data from various data sources. The most popular topology data sources, as described in this document, are the SiteScope solution templates and HPOM SPI's monitoring tools. They are also capable of delivering dynamic topology to UCMDB. For example, if we are monitoring our email business service, and we know that the infrastructure of the mailbox servers is deployed on a virtual infrastructure, we can use the VI SPI for both—monitoring and topology sync to UCMDB (as well as graphs). The topology is synced to the RTSM (BSM – UCMDB) in a real time sync. Every change that occurs in the virtual infrastructure will be synced to RTSM in a matter of minutes—allowing us to provide an up-to-date topology map in a dynamic IT environment while ensuring our efficient event management capabilities.

Following the prerequisite of a CI service modeling process that is already in place, good event management contains the following five steps:



1. ["Event Detection and Notification" on page 49](#)

This is the first step in the event life cycle. This section contains information about the detection of the event via monitoring tools in the organization; such as HP Operations Manager, HP SiteScope, HP End-User Management (EUM), ArcSight Logger, and third party tools such as Microsoft SCOM, NAGIOS, and so on.

After detecting a threshold/policy breach, the monitoring tool creates a notification. That notification is the event.

Event Detection and Notification is covered in Part I: ["End-to-End Service Monitoring in the IT Environment" on page 8](#), which presents recommendations and guidelines for the best way to implement a monitoring solution in a service-oriented IT environment.

2. ["Event Filtering" on page 50](#)

This section contains information on filtering the event notification according to the parameters of the events.

Not all events are critical. There is a need to filter the events in order to increase the efficiency of the operations bridge (to work on the most critical events to the business) and for them to appear correctly in the event console or be filtered out if they are not needed or not important to the organization.

This section covers the recommended methods for filtering and editing event parameters in the data collectors before sending them to the OMi operations bridge console. Doing so verifies

that all of the essential events are categorized and filtered before they arrive at the operations bridge console.

3. ["Event Correlation" on page 53](#)

This section contains recommendations, tips, techniques, and ways to correlate the events in the event management module (OMi).

Event correlation is the identification of relations between events. This is useful in the context of massive event storms and a complex monitored environment. Event correlation makes sense out of a large number of events and pinpoints the important events in this mass of information.

We can also help identify the root cause and symptoms from a large amount of events. Event correlation saves time by handling only these causal events, thereby fixing all of the events.

4. ["Event Handling" on page 58](#)

This section contains the variety of response options to an event in the HP BSM OMi operations bridge (main event console) and presents all of the recommendations regarding event handling.

It presents the response and ways to use automatic actions, semi-automatic actions, or user actions for an event, and explain the flow of necessary actions to be taken from the moment the event arrives at the operations bridge console until the beginning of event handling after selecting the correct response.

Also covered is annotating the information that has been done in the process and opening incidents in the service management application to manage all of the actions that are being performed in the event handling process.

5. ["Event Closure and Review" on page 61](#)

This section covers the management of event handling from within the operations bridge console until the closure of the event.

When you manage events in a large IT organization, you need to keep track on your events at all times and make sure you are working to solve them as quickly as possible in order to restore normal operation.

This section presents the recommended way to manage event handling using integrations between the operations bridge console (HP BSM OMi) and the service management system (HP Service Manager), and how this integration can help manage events in an efficient way from creation until closure of the event.

Event Detection and Notification

Detection and notification is the first phase of the event management process.

Event detection starts with a monitoring tool or a data collection tool that measures an IT service or a specific CI—such as an application, server, or a network component—and sends a notification on the changing of the CI status to an operational status that is not normal, meaning that there is a disruption in the IT service.

The monitoring tools and data collectors are combined with a set of tools that respond to that status change.

The first step in the event management process is to properly install and configure all of the service monitoring tools so that they generate a notification for any disruption in the services of IT. Later on, this enables the receipt of meaningful events and covers every possible scenario so that important data for a good event management process is not missed.

For example, if an email service has poor performance, and as a result of good end-to-end service monitoring there are a few events in the OMi console that are related to this issue, this is used in later steps of the event management process—for example, in correlation rules and impact rules—in order to fix the disruption in the service more efficiently.

For best practices and guidelines on the implementation of end-to-end service monitoring for the event detection phase, see Part I, "[End-to-End Service Monitoring in the IT Environment](#)" on page 8 of this guide.

Event Filtering

This section includes:

Overview	50
Configuration	51

Overview

Event filtering is the second step in the event management process. It serves the purpose of reducing the amount of events that reach the operations console, allowing better focus on those events that have significant impact on managed CIs and services.

The meaning of event filtering is to create filtering rules for events in the data collectors or in the monitoring tools.

Filtering rules are the gatekeepers for the OMi console, and they decide which one of the detected disruptions is to be sent as an event to the event management console (OMi) and which one of them needs to be filtered out due to many variables.

Event filtering refinement is an ongoing process that provides balancing between getting too many events versus missing important ones.

These rules are very dynamic and depend on a lot of variables, such as:

- **Error severity.** In some cases, it makes sense to filter out events with a lower severity. For example, if you monitor an email service and receive events on a large scale environment, you can choose to ignore some of the **informational/low severity** events, such as **user logon to his mailbox failed**, since they have no effect on the business. However, informational and low severity events, when received in particular patterns or quantities, can indicate that there might be a critical problem that causes them, and it is recommended to configure smart correlation rules on those events so that they are more meaningful when needed. For more information, see ["Event Correlation" on page 53](#).

In our example, if we receive a large amount of **user logon to his mailbox failed** events, it can indicate that there might be a problem with the connection from the mailbox servers to the directory services.

Note: A low-impact event that is not important to the IT organization should be filtered out regardless of its severity.

- **Positive event.** Does the event affect the overall status of the service?

Positive events can indicate that a certain CI has returned or is still at a normal operation status. Some positive events help in the event closure step, so it is not recommended to filter them out. However, there are events that have no effect on the business—for example, a heartbeat monitor that sends an event every minute. It makes sense to filter them out and only create an

event when the condition discovers a problem with the CI. In our example, we can send an event after the CI does not send any heartbeat messages in the last five minutes.

- **Event category.** Does the event category make the event important? We can create local view filters in the OMi by event categories. This allows us to delegate the controls on the console between operators. For example, if we are monitoring two business services, then we can set a filter that allows each one of the two operation teams to handle each one of the business services.
- **Service operation hours.** Filter out the events so they will match the planned operation hours of the service. For example, if we are receiving events from monitors on a service that is supposed to be working from only 08:00-17:00, we can filter the events from 17:00-08:00—or choose to ignore only the non-critical events at this time.

Note: This can be done using Downtime Management with either HP Operations Management or in BSM. For more information, see Chapter 17, "Downtime Management" in the [BSM 9.20 Platform Administration Guide](#).

- **Environment.** You need to take into consideration the environment of the impacted business application—development, testing or production—because the event impact is derived from the environment. For instance, in the testing environment, you can filter out all non-critical events, unless you want to validate the monitoring process itself.
- And many more...

It is recommended, when implementing an event management process, to create proper event filters and avoid unnecessary events that will cause event overload on the console and prevent us from seeing the important events.

In general, there are many rules for filtering events and many reasons to have them. Every organization has its own preferences and decision-making processes regarding event filtering and every organization provides different services using different applications and different infrastructure components.

Configuration

Almost all of the event filtering rules are defined in the monitoring creation phase—when you decide what you are monitoring and how critical it is. While creating the monitors, take into consideration that every time the monitor discovers that something is wrong, it sends an event with the severity you configured in the monitoring condition.

However, there is an additional filtering method you can use for a mass of events. This is called **grouping**. When all of the monitors are configured, you can create more universal filtering rules for a large number of events based on a common denominator.

For example, a large scale organization has decided that all of the low severity events—for example, warning messages in OMi—that are being sent to the event management main console (OMi) from an email service monitor are automatically filtered out. The decision can be based on the

lack of resources to handle the large amount of low severity events, on knowing that warning events from an email service monitor are not important to IT and to the organization, and so on.

It is important for the event to be sent with all of the correct parameters—such as severity, related CI, application, and so on—so they can be filtered more efficiently.

Filtering can also be achieved based on location. For example, HP Network Node Manager i (NNMi) can be defined to filter out all of the events on network components in the LAB data center. For more information, see "*HP Network Node Manager Online Help for Administrators*" that can be found in the [NNMi Documentation List](#) or the application's Help menu.

Sometimes there is an incentive to monitor as many aspects as possible when planning to deploy a monitor solution. Notice that this can also have disadvantages in the form of too many events that have no immediate significance to the IT. This can overload the console and the operators and cannot be responded to in a proper way. To prevent this, while planning the implementation of the monitors, concentrate on monitoring the objects that can have an impact on the CIs and services.

Remember that events with lower severity (and even positive events) can be just as important as the critical events, as they are an indication that a certain disruption has been fixed (maybe even automatically), or that can be used in correlation rules. For more information, see "[Stream-based Event Correlation](#)" on page 57.

Event Correlation

This section includes:

Overview	53
Health Indicators	54
Key Performance Indicators	55
Topology-based Event Correlation	56
Stream-based Event Correlation	57

Overview

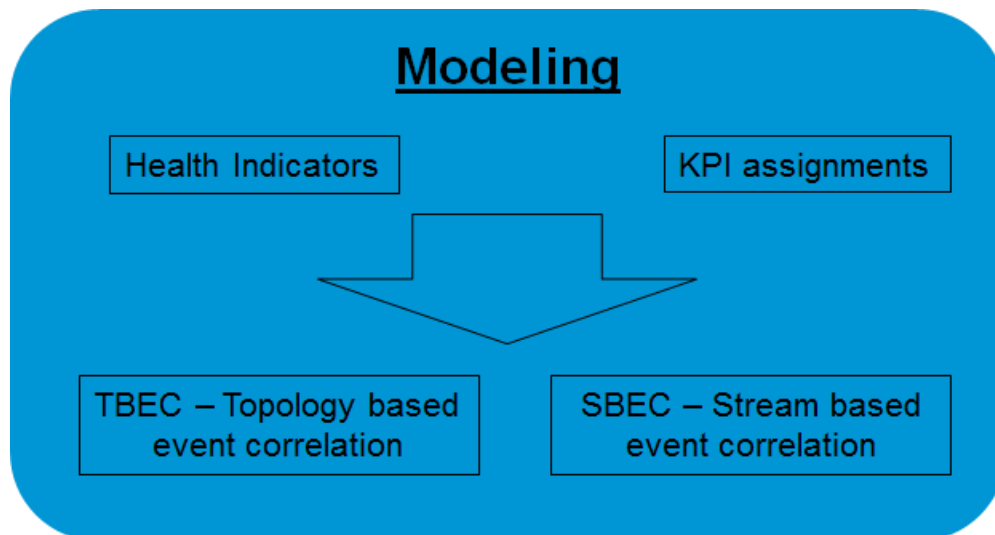
Event correlation is the ability to identify relations between events and use those relationships to create links between events.

These correlated events can be linked in either of the following two ways:

- Topology-based event correlation (TBEC)
- Stream-based event correlation (SBEC)

In order to create correlation rules in the OMi module, first create correct indicators and assign the events to them. The indicators in BSM for event correlation are:

- Health Indicators (HI).
- Key Performance Indicators (KPI).



Note: For more information on modeling, see ["Event Management" on page 44](#).

The event correlation process consists of the following steps:

- ["Health Indicators" below](#)
- ["Key Performance Indicators" on the next page](#)
- ["Topology-based Event Correlation" on page 56](#)
- ["Stream-based Event Correlation" on page 57](#)

Health Indicators

Health indicators (HIs) provide health statuses on the CIs that represent your monitored applications and business services. This means assigning an event to the correct indicator on the correct CI, thus connecting between the events and the CI.

Health indicators combined with service modeling facilitate the functionality of correlation rules.

Some HIs provide business metrics, such as backlog and volume, while others monitor various aspects of performance and availability, such as CPU load or disk space.

There are two types of data sources that affect an HI's status and value:

- events
- metrics

For example:

- **Events.** You can use data collectors, such as HP SiteScope and HP Operations Management (HPOM), to send events to OMi that will affect the health indicators—for example, **CPU load exceeded threshold**.
- **Metrics.** You can use data collectors, such as HP Real User Monitor (RUM) and HP Business Process Monitor (BPM), to send samples containing metrics—for example, **response time = 6 milliseconds**.

Another example: When monitoring a mailbox logon using BPM, we send metric samples to OMi for every monitor iteration, such as **response time = 5 seconds**, that affects the health indicator status. The HI status changes in accordance with the last sample that is received from the BPM.

For more information on managing events, see the [BSM version 9.20 OMi Concepts Guide](#).

Note: It is recommended to use metric samples from BPM and RUM monitoring to send events when a monitor is crossing a threshold, thus allowing the problem to be reflected in an event which does not only affect the KPI.

When an event is sent to OMi, it goes through the Service Health model and begins the process of the indicator's assignment.

There are two ways to assign the event to a Health Indicator:

- ETI embedded in the event
- HI (ETI) mapping rules (defined in BSM via filters on event properties)

We recommend sending the event with an event type indicator (ETI) whenever possible. The ETI includes a name and a state—for example **CPU_Load:exceeded**. Using HI definitions in the indicator repository, Service Health translates the ETI state into one of the standard Service Health statuses (Critical, Major, Minor, and so on). Also, some out-of-the-box monitors already contain an ETI. For more details, see the [BSM version 9.20 OMi Concepts Guide](#).

- We recommend not changing existing ETIs since this can influence and break impact analysis rules and stop affecting the out-of-the-box health indicators.
- When not using out-of-the-box ETIs and HIs, make sure that the HIs are as specific as possible. We recommend creating an HI for every monitor and a state of HI for every severity in the monitor conditions.

For more information on health indicators and ETI, see the latest *HP BSM User Guide* or *BSM Application Administration Guide* "Service Health" section on the [HP Software Product Manuals](#) web site.

Key Performance Indicators

Key Performance Indicators (KPIs) are high-level indicators of CI performance and availability, which apply calculation rules to the data provided by HIs to determine CI status.

Note: BSM contains many out-of-the-box KPI rules. Review them before creating a new KPI rule.

KPIs can be calculated using statuses of HIs, other KPIs, or a combination of these.

For example, you can specify a rule that sets the severity of the KPI to the worst severity status of any assigned HI, or to the average severity status of all child KPIs.

The value that results from the calculation is used to set a severity level for the KPI based on the KPI definitions. KPI severity can be normal, warning, minor, major, or critical. The resulting measurement for the KPI is translated into a color-coded status indicator displayed in Service Health, where the color represents a more desirable or less desirable condition for the KPI.

You can define a KPI to only use specific HIs that are of interest to you.

For example:

- The out-of-the-box system performance KPI (assigned to computer CI type) has many HIs; including CPU usage, disk utilization, memory, paging file, and more. If you are interested only in the CPU aspect, while measuring system performance on computers, set the KPI to only include the CPU-related HIs in the calculation.
- For a Windows server running an Exchange mailbox application, create an **Application availability** KPI and connect to it all of the HIs that are related to events that affect the application availability, such as the HI mailbox service status (**UP-normal**, **DOWN-critical**), and so on.

Note: It is important to configure the KPI with the correct calculation rule.

For more information on KPIs, see the latest *HP BSM User Guide* or *BSM Application Administration Guide* "Service Health" section in the [HP Software Product Manuals](#) web site.

Topology-based Event Correlation

Topology based event correlation (TBEC) contributes to the event management process by categorizing events using topology-based rules. Dependencies between events are analyzed to determine whether some events can be explained by other events.

In every IT operations group, there is organizational knowledge about events that cause other events. This knowledge should be used in order to create correlation rules. Within the RTSM topology views, you can also use the impact links between CIs to understand that a certain event on a CI can be the cause for another event on a linked CI (link symptom event to the cause event).

For example, you get two events—one event of a performance problem in the Outlook Web Access (OWA) user interface CI and another event of high CPU on the OWA server CI. If the CI model of the Exchange Business Service contains both CIs, we can correlate these events in the TBEC engine so that the event of high CPU on the OWA server CI is the cause for the event of the performance problem in the OWA user interface CI. In this example, we are able to work on resolving the cause event—instead of trying to address both events, leading to a waste of resources and possible conflicting actions.

This was a very simple example where we correlated two events and then were able to focus on one of them. The creation of a good event correlation process and refining it over time leads to a potential reduction of 60-70 percent of the events that the IT operations group handles on a day-to-day basis, with average correlations of five to six symptom events to one causal event. This result is not achieved overnight and requires a considerable effort of tracking down related events, but the return on investment (ROI) on this activity is positive over time.

Events can also be correlated across technical domains, such as databases, hardware, networks, and web applications. This comprehensive scope enables you to correlate events that, at first sight, might not seem to have any connection. The cross-domain functionality also increases productivity by reducing the amount of overlap between operators responsible for monitoring different technical areas. For example, by correlating events relating to database problems, network problems, and

storage problems, you can avoid the scenario of operators from the different technical areas all separately investigating different events that are the symptoms of one root cause event, or the practice of operators in different domains placing the fault on other domains which do not fall under their responsibility.

TBEC offers a number of benefits related to resolving complex events:

- Reduces the number of events displayed in the console, without ignoring or losing important data that enables users to drill down through the hierarchy of related events
- Supports event correlation across multiple domains to simplify root-cause analysis of events that generate symptom events
- Changes to topological data do not require changes to correlation rules
- Recommends the business priority for the correlated event's handling by looking at the event's severity and the event's related CIs' ties to defined SLA

Stream-based Event Correlation

Stream-based event correlation (SBEC) uses rules and filters to identify commonly occurring events or combinations of events and helps simplify the handling of such events by automatically identifying events that can be withheld, removed, or that need a new event to be generated and displayed to the operators.

The following types of SBEC rules can be configured:

- **Repetition Rules.** Frequent repetitions of the same event may indicate a problem that requires attention—for example, you can create a repetition rule about reboots of more than 10 events in the same node within two hours that create a critical event.
- **Combination Rules.** Combination of different events occurring together or in a particular order indicates an issue and requires special treatment—for example, If there are **interface down – interface up** events from NNMI in a short period of time (for example, one minute), you can choose to discard both of them using SBEC and generate a new **log only** event that a transient issue occurred.
- **Missing Recurrence Rules.** Regularly recurring event is missing—for example, a regular heartbeat event does not arrive when expected.

Event Handling

After receiving the event in HP Operations Management (OMi), after all of the configured operations (filtering, correlation, and so on) are performed, the event is left pending in the OMi console waiting for a response.

The event appears in the **Event Browser** window (OMi Event Browser) in the Operations Manager module. As a BSM user, you can define the views for using the event browser to suite your organizational needs.

You can define dashboards containing the event browser and use wiring to integrate it with other BSM modules; for example, creating a page containing the event browser wired to a topology map from RTSM, thus allowing you to see events that are related to the selected topology view. This can contribute to the investigation of the events and allow us to create a dashboard with more relevance to specific users. You can also define filters and configure predefined event filters in the OMi based on user groups (users with specific permissions for specific events), event attributes, and so on.

For more information on using the OMi console and MyBSM, see the *OMi Concept Guide* and *BSM User Guide* on the [HP Software Product Manuals](#) web site.

A response to an event is an action being performed on the affected CI based on the information provided by the event. The action can be performed automatically by scripting, semi-automatically by an operator, or fully performed by an operator (usually using a procedure to solve the event).

A response to an event can be:

- Performing an action to fix the event

For example, if there is an event for **Windows service stopped**, we perform an action to start Windows. An action can be an operator's manual action or a script.

Note: This can be done as part of the CLIP Solution using HP Business Service Management and HP Operations Orchestration. For more information, see the [CLIP version 9.30 Solution Configuration Guides](#).

- Opening an incident

For example, if there is an event causing a critical problem in the mailbox Exchange server, we open an incident and send it to the Exchange application team.

The incident can become a trigger for a change request, problem management issue, or any other process, but it stays linked to the event until the issue that caused the event is fixed.

Note: This can be done as part of the CLIP Solution using HP Business Service Management and HP Service Manager. For more information, see the [CLIP version 9.30 Solution Configuration Guides](#).

When an event appears in the OMi Event Browser, there are three ways to implement the selected response:

- **Automatic action.** When the event arrives, a script is automatically run to try and resolve the issue. This is used when fixing common events, for instance, when File System C:\ (usually the operating system's file system) is at full capacity, although it can also be used for other events as well, according to the organizational needs.

In our example, there is a minor issue with the email service—an IIS web service has stopped working on one of the Outlook Web Access servers.

In that case, an event is sent to the OMi, and an automatic action runs a script on the server trying to start the service.

Since this is a common issue, it is less time consuming to run the automatic action to try and start the service. If the automatic action fails, the event remains in the OMi console and can be handled in other ways.

In addition, if, when reviewing the day-to-day event activities, operators identify recurring actions, the operator can create an automate action to handle them.

It is worthwhile to search for patterns of common events in order to create automatic scripts/responses to resolve them.

- **Semi-automatic action.** While using the same capability as the automatic action, the action is triggered manually. This method is used while fixing common events on very critically affected CIs so that there is an operator review on what is happening. It is also common to use semi-automatic scripts to gather more information on the issue of the event.

For example, if there is an event that indicates a CPU performance issue, we can manually run a script that finds the process that consumes most of the CPU capacity and decide from a script prompt if we want to kill the process or open an incident to the correct team from within the OMi. The killing process cannot be executed automatically, as operator judgment is also a required input. Therefore, using the semi-automatic script is the best way for an operator to perform this action.

For more information, see the [HP CLIP version 9.30 Solution Configuration Guides](#).

There is also the repeatability of the results—complex actions such as **mailbox server file system clean-up** can be performed in the same way by different operators without expecting different results.

- **Operator action.** Where the operator performs a corresponding action to the event, but not from an OMi console.

For example, if there is an event on one of the mailbox server's databases that exceeds the tablespace capacity, the operator can manually open an incident and call the database administrator (DBA) to fix this. The DBA can connect to the database and allocate more storage to the tablespace.

It is possible to combine the methods, depending on the use case. For example, you can use a semi-automatic operation—running a script manually that gathers information on the problem before the operator begins to handle it.

Event Closure and Review

This step is the final step in the event's life cycle, and it refers to the event closure procedure.

After the necessary actions for fixing the issue that caused the event have been completed, you need to make sure that the problem has been solved and the event is ready to be closed.

There are several ways to close an event:

- **Automatic closure.** A positive event status indicates the status has returned to normal operation. This can be done in the filtering stage or when creating a monitor (as described in ["End-to-End Service Monitoring in the IT Environment" on page 8](#)) and it involves adding another condition to the monitor that sends an event when the status of the condition is good.
- **Semi-automatic closure.** If an incident is linked to the event via the CLIP Solution, closing the incident by the responsible team or person closes the event as well.
- **Manual closure.** The operator solves (closes) the event manually.

This action requires careful investigation of the event cause by the operator, who has to be sure that the problem has been resolved before closing the event. If an event is being closed manually and the problem still exists, the only indication for it will be the effect of the problem on users or on other CIs.

Recommendations:

- Have automatic closure as much as possible, since this is an indication the source problem of the event has been closed. However, in cases where this cannot be done (for example, when monitoring log files and there is no indication that the problem has been resolved), it is recommended to:
 - Always add an annotation of the action you performed on the event during the status check for logging purposes.
 - Check with the responsible team that the problem has been resolved.
 - Have an incident on the event. If an incident is linked to the event via the CLIP Solution, the event is closed along with the incident—giving the responsible team the ability to help in the event management process.
- When manually closing an event, make sure the event is assigned to you and you are the operator responsible for the event. This helps to avoid closing an event that is still under investigation. For more information, see ["Managing Events" on page 64](#).

Chapter 7: Using Event Management in the Detect to Correct Value Stream

This chapter includes:

Overview	62
Detect to Correct Value Stream Diagram	63

Overview

The HP Detect to Correct (D2C) Value Stream is a cross-portfolio value stream that uses several HP Products and Integrations. Service monitoring and event management are the base of the Detect to Correct Value Stream. D2C provides a framework for integrating the monitoring, management, remediation, and other operational aspects associated with realized services and/or those under construction. It also provides a comprehensive overview of the business of IT operations and the services these teams deliver.

Anchored by the service model, D2C delivers new levels of insight which help improve the understanding of the inter-dependencies among the various operational domains; including event, incident, problem, change, and configuration management. D2C also provides the business context for operational requests and new requirements. D2C is designed to accommodate a variety of sourcing methodologies across services, technologies, and functions.

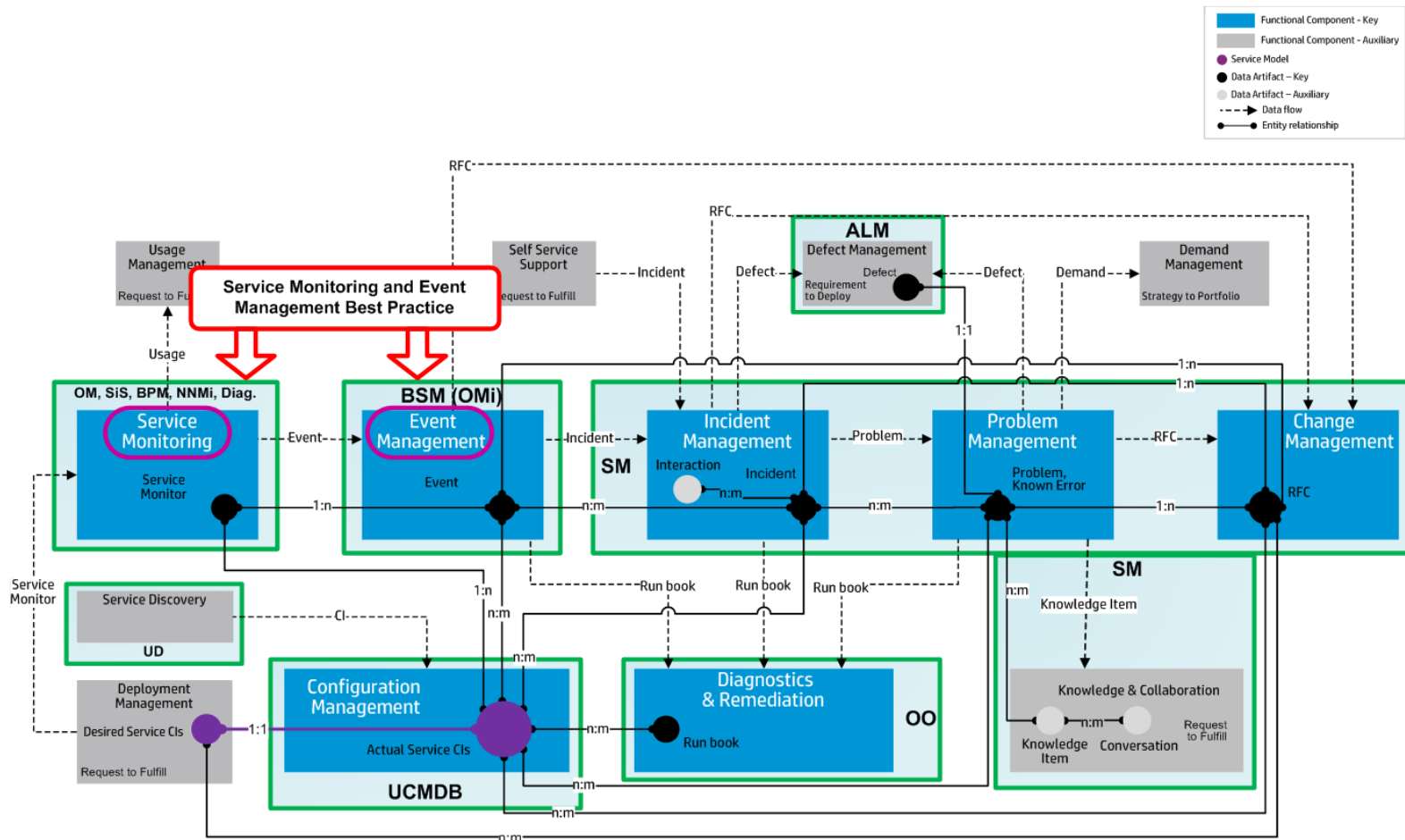
Detect to Correct connects the various functions involved in service operations to enhance results and efficiencies. Today, most teams work in isolation because they lack visibility to key artifacts and lack a common taxonomy to facilitate collaboration and sharing. When attempts are made to resolve this issue using a process-led approach, they are often too difficult and/or complex to finish or there is a technology or organization shift that invalidates the result. Using the service model and value stream, the functional components and data exchanges that comprise D2C remain consistent regardless of changes in technology, organization structure, process and/or methodologies.

The Detect to Correct Value Stream's detection phase monitors events throughout their life cycle. Therefore, it is highly recommended to configure your monitoring environment according to HP Best Practices in order to provide a solid ground for the D2C implementation.

Note: For more information regarding the Detect to Correct Value Stream, see the [Detect to Correct Value Stream Concept and Configuration Guide](#).

Detect to Correct Value Stream Diagram

The following diagram is the main use-case for the Detect to Correct Value Stream. The processes and functional components from the End-to-End Service Monitoring and Event Management Best Practice that are a part of D2C are highlighted.



Chapter 8: Managing Events

This chapter includes:

Overview	64
----------------	----

Overview

After confirming that all of the meaningful events are flowing to the main IT event console (HP BSM Operations Management (OMi)), there is a need to manage them.

Managing events means to make sure that all of the events in the console are being handled according to a predefined procedure of the most practical way to solve the events with the least effect on the business.

Every organization has its own priorities and rules regarding event management. For instance:

- Which events should we handle first?
- Do we report critical events to the operations team manager?
- Do we add annotations on every event-related action that is being performed?
- In which case do we open incidents on events?
- Should we handle low severity events?
- Plus many more

The answers to these and similar questions are the organizational guidelines/rules for event management.

Every IT organization needs to define its own event management guidelines, which usually depend on the IT service level agreements (SLAs).

Our recommendations for properly managing events are:

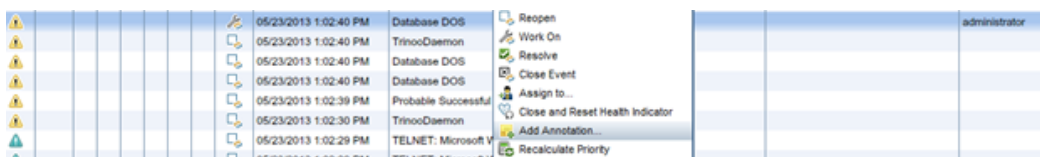
1. Have a team of operators monitor the OMi main console 24 hours a day/7 days a week. Use enough personnel to prevent messages from waiting for a response for more than a few minutes. The correct time must be configured in every organization based on parameters such as SLAs, the amount of messages per minute, and so on.
2. Use OMi properly.
 - Every operator that is using OMi must have their own user name defined in BSM with the correct permissions. For more information, see the [BSM version 9.22 User Guide](#).
 - When an operator starts handling an event, that operator must take ownership of the event in order to prevent others from working on the same event. This action is called **event assignments**.

When you start to work on an event in OMi, click the **Work on** option in order to take ownership on the event. Then the event will be assigned to you.



- Since there are shift changes between the operators and also there are investigations on events in some cases, it is very important for the operator or the person who is handling the event to write what has been done so far to try and solve the problem.

This can be done in a related incident or in **Add Annotation** in OMi.



- When handling events, open incidents on them in order to manage the event via the Incident Management process. This allows the use of all of the Incident Management features on the event, such as opening a change request, forwarding the incident to another team, and so on.

We recommend opening incidents in OMi using the CLIP Solution. (For more information, see the [HP CLIP version 9.30 Solution Guides](#).)

3. Solve events with higher priority/severity first, and then the other events. Events with higher priority/severity have more effect on the business. It is also important, when approaching a group of correlated events, to handle the cause event first, since this event is the cause for all of the other events. Solving the cause event allows us to close all of the related events. For more information on managing events, see the [BSM version 9.20 OMi Concepts Guide](#).